One Technology Way • P.O. Box 9106 • Norwood, MA 02062-9106, U.S.A. • Tel: 781.329.4700 • Fax: 781.461.3113 • www.analog.com

ADV8005 Register Map Documentation

TABLE OF CONTENTS

REVISION HISTORY

6/14—Revision 0: Initial Version

INTRODUCTION

This document describes the I²C control registers for the ADV8005. The ADV8005 is a video signal processor (VSP) with TTL and Serial Video inputs that can de-interlace and scale input video, generate and blend a bitmap based on-screen display (OSD) and output the blended video using one or more of the part's outputs; dual HDMI transmitters and a 6-DAC encoder with SD and HD support.

The Register Tables section of this document provides detailed register tables for the ADV8005 register maps. The Signal Documentation section provides detailed signal documentation for each register.

LEGAL TERMS AND CONDITIONS

1 REGISTER TABLES

1.1 IO MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1A 00	0xFE	video in id	rw	video_in_id[7]	video_in_id[6]	video_in_id[5]	video_in_id[4]	video_in_id[3]	video_in_id[2]	video_in_id[1]	video_in_id[0]
0x1A 01	0x02	ttl output selection_1	rw	-	-	-	-	-	-	-	ttl_ps444_in
0x1A 02	0x70	ttl output selection_2	rw	ttl_op_format[3]	ttl_op_format[2]	ttl_op_format[1]	ttl_op_format[0]	ttl_vid_out_en	ttl_out_sel[2]	ttl_out_sel[1]	ttl_out_sel[0]
0x1A 03	0x00	tx input selection	rw	tx1_inp_sel[3]	tx1_inp_sel[2]	tx1_inp_sel[1]	tx1_inp_sel[0]	tx2_inp_sel[3]	tx2_inp_sel[2]	tx2_inp_sel[1]	tx2_inp_sel[0]
0x1A 04	0x00	encoder input selection	rw	hd_enc_inp_sel[3]	hd_enc_inp_sel[2]	hd_enc_inp_sel[1]	hd_enc_inp_sel[0]	sd_enc_inp_sel[3]	sd_enc_inp_sel[2]	sd_enc_inp_sel[1]	sd_enc_inp_sel[0]
0x1A 05	0x00	datapath muxing_1	rw	svsp_inp_sel[3]	svsp_inp_sel[2]	svsp_inp_sel[1]	svsp_inp_sel[0]	pvsp_inp_sel[3]	pvsp_inp_sel[2]	pvsp_inp_sel[1]	pvsp_inp_sel[0]
0x1A 06	0x00	datapath muxing_2	rw	p2i_inp_sel[3]	p2i_inp_sel[2]	p2i_inp_sel[1]	p2i_inp_sel[0]	osd_blend_inp_sel[3]	osd_blend_inp_sel[2]	osd_blend_inp_sel[1]	osd_blend_inp_sel[0]
0x1A 07	0x64	datapath muxing_3	rw	rx_proc_bypass	sde_only_gate_hde_inp	-	-	s_inp_chan_sel[1]	s_inp_chan_sel[0]	p_inp_chan_sel[1]	p_inp_chan_sel[0]
0x1A 08	0x00	datapath muxing_4	rw	aud_input_mode[1]	aud_input_mode[0]	-	-	osd_blend_inp_2_sel[3]	osd_blend_inp_2_sel[2]	osd_blend_inp_2_sel[1]	osd_blend_inp_2_sel[0]
0x1A 09	0x00	datapath muxing_5	rw	hps_inp_sel[3]	hps_inp_sel[2]	hps_inp_sel[1]	hps_inp_sel[0]	-	-	-	-
0x1A 0D	0x28	dither control_1	rw	-	-	updither_level[1]	updither_level[0]	updither_alpha[3]	updither_alpha[2]	updither_alpha[1]	updither_alpha[0]
0x1A 0E	0x00	dither control_2	rw	-	-	-	-	max_diff_pixel[11]	max_diff_pixel[10]	max_diff_pixel[9]	max_diff_pixel[8]
0x1A 0F	0x32	dither control_3	rw	max_diff_pixel[7]	max_diff_pixel[6]	max_diff_pixel[5]	max_diff_pixel[4]	max_diff_pixel[3]	max_diff_pixel[2]	max_diff_pixel[1]	max_diff_pixel[0]
0x1A 10	0x00	dither control_4	rw	max_diff_sum[15]	max_diff_sum[14]	max_diff_sum[13]	max_diff_sum[12]	max_diff_sum[11]	max_diff_sum[10]	max_diff_sum[9]	max_diff_sum[8]
0x1A 11	0x64	dither control_5	rw	max_diff_sum[7]	max_diff_sum[6]	max_diff_sum[5]	max_diff_sum[4]	max_diff_sum[3]	max_diff_sum[2]	max_diff_sum[1]	max_diff_sum[0]
0x1A 14	0x0C	spi control	rw	-	-	-	-	spi_slave_cpol	spi_slave_cpha	spi_master_cpol	spi_master_cpha
0x1A 24	0x81	video adjust_1	rw	db_vid_adj_params	saturation_en	brightness_en	contrast_en	blank_level_y[11]	blank_level_y[10]	blank_level_y[9]	blank_level_y[8]
0x1A 25	0x00	video adjust_2	rw	blank_level_y[7]	blank_level_y[6]	blank_level_y[5]	blank_level_y[4]	blank_level_y[3]	blank_level_y[2]	blank_level_y[1]	blank_level_y[0]
0x1A 26	0x80	video adjust_3	rw	blank_level_u[11]	blank_level_u[10]	blank_level_u[9]	blank_level_u[8]	blank_level_u[7]	blank_level_u[6]	blank_level_u[5]	blank_level_u[4]
0x1A 27	0x08	video adjust_4	rw	blank_level_u[3]	blank_level_u[2]	blank_level_u[1]	blank_level_u[0]	blank_level_v[11]	blank_level_v[10]	blank_level_v[9]	blank_level_v[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1A 28	0x00	video adjust_5	rw	blank_level_v[7]	blank_level_v[6]	blank_level_v[5]	blank_level_v[4]	blank_level_v[3]	blank_level_v[2]	blank_level_v[1]	blank_level_v[0]
0x1A 29	0x80	video adjust_6	rw	saturation[7]	saturation[6]	saturation[5]	saturation[4]	saturation[3]	saturation[2]	saturation[1]	saturation[0]
0x1A 2A	0x00	video adjust_7	rw	brightness[7]	brightness[6]	brightness[5]	brightness[4]	brightness[3]	brightness[2]	brightness[1]	brightness[0]
0x1A 2B	0x80	video adjust_8	rw	contrast[7]	contrast[6]	contrast[5]	contrast[4]	contrast[3]	contrast[2]	contrast[1]	contrast[0]
0x1A 2C	0x0A	spi control 2	rw	spi_filter_en	spi_filter_sel	-	-	osd_vid_drvstr[1]	osd_vid_drvstr[0]	osd_clk_drvstr[1]	osd_clk_drvstr[0]
0x1A 2D	0x00	auto contrast adjust_1	rw	-	-	-	-	ace_gamma_gain[3]	ace_gamma_gain[2]	ace_gamma_gain[1]	ace_gamma_gain[0]
0x1A 2E	0x00	auto contrast adjust_2	rw	ace_gamma_offset[7]	ace_gamma_offset[6]	ace_gamma_offset[5]	ace_gamma_offset[4]	ace_gamma_offset[3]	ace_gamma_offset[2]	ace_gamma_offset[1]	ace_gamma_offset[0]
0x1A 2F	0x00	auto contrast adjust_3	rw	ace_gamma_fixed[7]	ace_gamma_fixed[6]	ace_gamma_fixed[5]	ace_gamma_fixed[4]	ace_gamma_fixed[3]	ace_gamma_fixed[2]	ace_gamma_fixed[1]	ace_gamma_fixed[0]
0x1A 30	0x00	auto contrast adjust_4	rw	ace_enable	-	-	ace_luma_gain[4]	ace_row_offset[9]	ace_row_offset[8]	ace_col_offset[9]	ace_col_offset[8]
0x1A 31	0x0A	auto contrast adjust_5	rw	ace_row_offset[7]	ace_row_offset[6]	ace_row_offset[5]	ace_row_offset[4]	ace_row_offset[3]	ace_row_offset[2]	ace_row_offset[1]	ace_row_offset[0]
0x1A 32	0x0A	auto contrast adjust_6	rw	ace_col_offset[7]	ace_col_offset[6]	ace_col_offset[5]	ace_col_offset[4]	ace_col_offset[3]	ace_col_offset[2]	ace_col_offset[1]	ace_col_offset[0]
0x1A 33	0xDF	auto contrast adjust_7	rw	ace_luma_gain[3]	ace_luma_gain[2]	ace_luma_gain[1]	ace_luma_gain[0]	ace_blend_alpha[3]	ace_blend_alpha[2]	ace_blend_alpha[1]	ace_blend_alpha[0]
0x1A 34	0x88	auto contrast adjust_8	rw	ace_chroma_gain[3]	ace_chroma_gain[2]	ace_chroma_gain[1]	ace_chroma_gain[0]	ace_chroma_max[3]	ace_chroma_max[2]	ace_chroma_max[1]	ace_chroma_max[0]
0x1A 39	0x00	dpll control_1	rw	-	-	-	-	tt_talon_mode[3]	tt_talon_mode[2]	tt_talon_mode[1]	tt_talon_mode[0]
0x1A 3A	0x00	dpll control_2	rw	pvsp_period_force	-	-	pvsp_vid_clk_update	-	-	pvsp_vid_clk_period[33]	pvsp_vid_clk_period[32]
0x1A 3B	0xAA	dpll control_3	rw	pvsp_vid_clk_period[31]	pvsp_vid_clk_period[30]	pvsp_vid_clk_period[29]	pvsp_vid_clk_period[28]	pvsp_vid_clk_period[27]	pvsp_vid_clk_period[26]	pvsp_vid_clk_period[25]	pvsp_vid_clk_period[24]
0x1A 3C	0xA0	dpll control_4	rw	pvsp_vid_clk_period[23]	pvsp_vid_clk_period[22]	pvsp_vid_clk_period[21]	pvsp_vid_clk_period[20]	pvsp_vid_clk_period[19]	pvsp_vid_clk_period[18]	pvsp_vid_clk_period[17]	pvsp_vid_clk_period[16]
0x1A 3D	0x00	dpll control_5	rw	pvsp_vid_clk_period[15]	pvsp_vid_clk_period[14]	pvsp_vid_clk_period[13]	pvsp_vid_clk_period[12]	pvsp_vid_clk_period[11]	pvsp_vid_clk_period[10]	pvsp_vid_clk_period[9]	pvsp_vid_clk_period[8]
0x1A 3E	0x00	dpll control_6	rw	pvsp_vid_clk_period[7]	pvsp_vid_clk_period[6]	pvsp_vid_clk_period[5]	pvsp_vid_clk_period[4]	pvsp_vid_clk_period[3]	pvsp_vid_clk_period[2]	pvsp_vid_clk_period[1]	pvsp_vid_clk_period[0]
0x1A 3F	0x00	dpll control_7	rw	svsp_period_force	-	-	svsp_vid_clk_update	-	-	svsp_vid_clk_period[33]	svsp_vid_clk_period[32]
0x1A 40	0x00	dpll control_8	rw	svsp_vid_clk_period[31]	svsp_vid_clk_period[30]	svsp_vid_clk_period[29]	svsp_vid_clk_period[28]	svsp_vid_clk_period[27]	svsp_vid_clk_period[26]	svsp_vid_clk_period[25]	svsp_vid_clk_period[24]
0x1A 41	0x00	dpll control_9	rw	svsp_vid_clk_period[23]	svsp_vid_clk_period[22]	svsp_vid_clk_period[21]	svsp_vid_clk_period[20]	svsp_vid_clk_period[19]	svsp_vid_clk_period[18]	svsp_vid_clk_period[17]	svsp_vid_clk_period[16]
0x1A 42	0x00	dpll control_10	rw	svsp_vid_clk_period[15]	svsp_vid_clk_period[14]	svsp_vid_clk_period[13]	svsp_vid_clk_period[12]	svsp_vid_clk_period[11]	svsp_vid_clk_period[10]	svsp_vid_clk_period[9]	svsp_vid_clk_period[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1A 43	0x00	dpll control_11	rw	svsp_vid_clk_period[7]	svsp_vid_clk_period[6]	svsp_vid_clk_period[5]	svsp_vid_clk_period[4]	svsp_vid_clk_period[3]	svsp_vid_clk_period[2]	svsp_vid_clk_period[1]	svsp_vid_clk_period[0]
0x1A 44	0x00	dpll control_12	rw	pvsp_freq_sel	pvsp_track_en	-	-	svsp_freq_sel	svsp_track_en	-	-
0x1A 4A	0x54	ancillary extraction_2	rw	did_a[7]	did_a[6]	did_a[5]	did_a[4]	did_a[3]	did_a[2]	did_a[1]	did_a[0]
0x1A 4B	0xA8	ancillary extraction_3	rw	sdid_a[7]	sdid_a[6]	sdid_a[5]	sdid_a[4]	sdid_a[3]	sdid_a[2]	sdid_a[1]	sdid_a[0]
0x1A 4C	0x20	ancillary extraction_4	rw	vbi_src	-	-	-	ccap_odd_en	ccap_even_en	cgms_anc_en	wss_anc_en
0x1A 4D	0x18	ancillary delay	rw	-	-	-	-	-	-	anc_delay[1]	anc_delay[0]
0x1A 4E	0x00	frequency tracking control_1	rw	-	pvsp_in_vs_gate_man[2]	pvsp_in_vs_gate_man[1]	pvsp_in_vs_gate_man[0]	pvsp_err_sel	-	-	-
0x1A 4F	0x00	frequency tracking control_2	rw	-	svsp_in_vs_gate_man[2]	svsp_in_vs_gate_man[1]	svsp_in_vs_gate_man[0]	svsp_err_sel	-	-	-
0x1A 5B	0x22	ddr2 controller config_12	rw	sdram_size[3]	sdram_size[2]	sdram_size[1]	sdram_size[0]	-	-	-	-
0x1A 5C	0x35	ddr2 controller config_13	rw	word_size[3]	word_size[2]	word_size[1]	word_size[0]	-	-	-	-
0x1A 5D	0x21	ddr2 controller config_14	rw	-	-	-	-	-	-	burst_length[2]	burst_length[1]
0x1A 5E	0x00	ddr2 controller config_15	rw	burst_length[0]	-	-	-	-	-	-	-
0x1A 69	0x00	rx interrupt control_1	rw	store_unmasked_irqs	-	-	-	intrq_dur_sel[1]	intrq_dur_sel[0]	-	-
0x1A 6A	0x00	rx interrupt control_2	sc	-	-	rx_cable_det_clr	rx_tmdspll_lck_clr	rx_tmds_clk_det_clr	rx_video_3d_clr	rx_av_mute_clr	rx_hdmi_mode_clr
0x1A 6B	0x00	rx interrupt control_3	sc	rx_gen_ctl_pckt_clr	rx_gamut_mdata_pckt_clr	rx_isrc2_pckt_clr	rx_isrc1_pckt_clr	rx_vs_info_frm_clr	rx_ms_info_frm_clr	rx_spd_info_frm_clr	rx_avi_info_frm_clr
0x1A 6C	0x00	rx interrupt control_4	rw	-	-	rx_cable_det_mb2	rx_tmdspll_lck_mb2	rx_tmds_clk_mb2	rx_video_3d_mb2	rx_av_mute_mb2	rx_hdmi_mode_mb2
0x1A 6D	0x00	rx interrupt control_5	rw	rx_gen_ctl_pckt_mb2	rx_gamut_mdata_pckt_mb2	rx_isrc2_pckt_mb2	rx_isrc1_pckt_mb2	rx_vs_info_frm_mb2	rx_ms_info_frm_mb2	rx_spd_info_frm_mb2	rx_avi_info_frm_mb2
0x1A 70	0x00	rx interrupt control_8	sc	-	-	rx_vs_inf_cks_err_edge_clr	rx_ms_inf_cks_err_edge_clr	rx_spd_inf_cks_err_edge_clr	rx_avi_inf_cks_err_edge_clr	rx_deepcolor_chng_edge_clr	rx_tmds_clk_chng_edge_clr
0x1A 71	0x00	rx interrupt control_9	sc	rx_pkt_err_edge_clr	rx_gamut_mdata_pckt_edge_clr	rx_isrc2_pckt_edge_clr	rx_isrc1_pckt_edge_clr	rx_vs_info_frm_edge_clr	rx_ms_info_frm_edge_clr	rx_spd_info_frm_edge_clr	rx_avi_info_frm_edge_clr
0x1A 72	0x00	rx interrupt control_10	rw	-	-	rx_vs_inf_cks_err_edge_mb2	rx_ms_inf_cks_err_edge_mb2	rx_spd_inf_cks_err_edge_mb2	rx_avi_inf_cks_err_edge_mb2	rx_deepcolor_chng_edge_mb2	rx_tmds_clk_chng_edge_mb2
0x1A 73	0x00	rx interrupt control_11	rw	rx_pkt_err_edge_mb2	rx_gamut_mdata_pckt_edge_mb2	rx_isrc2_pckt_edge_mb2	rx_isrc1_pckt_edge_mb2	rx_vs_info_frm_edge_mb2	rx_ms_info_frm_edge_mb2	rx_spd_info_frm_edge_mb2	rx_avi_info_frm_edge_mb2
0x1A 76	0x00	interrupt control	rw	-	-	-	-	vsp_int_pol[1]	vsp_int_pol[0]	tx_int_pol[1]	tx_int_pol[0]
0x1A 78	0x02	rx hot plug assert control	rw	de_h_beg_pos[11]	de_h_beg_pos[10]	hs_beg_pos[11]	hs_beg_pos[10]	hs_end_pos[11]	hs_end_pos[10]	hpa_man_value_a	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1A 79	0xB6	vid - cea-861 timing control_1	rw	de_v_beg_e_pos[6]	de_v_beg_e_pos[5]	de_v_beg_e_pos[4]	de_v_beg_e_pos[3]	de_v_beg_e_pos[2]	de_v_beg_e_pos[1]	de_v_beg_e_pos[0]	de_v_beg_o_pos[6]
0x1A 7A	0xD0	vid - cea-861 timing control_2	rw	de_v_beg_o_pos[5]	de_v_beg_o_pos[4]	de_v_beg_o_pos[3]	de_v_beg_o_pos[2]	de_v_beg_o_pos[1]	de_v_beg_o_pos[0]	de_h_beg_pos[9]	de_h_beg_pos[8]
0x1A 7B	0xA0	vid - cea-861 timing control_3	rw	de_h_beg_pos[7]	de_h_beg_pos[6]	de_h_beg_pos[5]	de_h_beg_pos[4]	de_h_beg_pos[3]	de_h_beg_pos[2]	de_h_beg_pos[1]	de_h_beg_pos[0]
0x1A 7C	0x01	vid - cea-861 timing control_4	rw	hs_beg_pos[9]	hs_beg_pos[8]	hs_beg_pos[7]	hs_beg_pos[6]	hs_beg_pos[5]	hs_beg_pos[4]	hs_beg_pos[3]	hs_beg_pos[2]
0x1A 7D	0xC0	vid - cea-861 timing control_5	rw	hs_beg_pos[1]	hs_beg_pos[0]	hs_end_pos[9]	hs_end_pos[8]	hs_end_pos[7]	hs_end_pos[6]	hs_end_pos[5]	hs_end_pos[4]
0x1A 7E	0x00	vid - cea-861 timing control_6	rw	hs_end_pos[3]	hs_end_pos[2]	hs_end_pos[1]	hs_end_pos[0]	-	vs_h_beg_o_pos[10]	vs_h_beg_o_pos[9]	vs_h_beg_o_pos[8]
0x1A 7F	0x10	vid - cea-861 timing control_7	rw	vs_h_beg_o_pos[7]	vs_h_beg_o_pos[6]	vs_h_beg_o_pos[5]	vs_h_beg_o_pos[4]	vs_h_beg_o_pos[3]	vs_h_beg_o_pos[2]	vs_h_beg_o_pos[1]	vs_h_beg_o_pos[0]
0x1A 80	0x0E	vid - cea-861 timing control_8	rw	vs_h_beg_e_pos[10]	vs_h_beg_e_pos[9]	vs_h_beg_e_pos[8]	vs_h_beg_e_pos[7]	vs_h_beg_e_pos[6]	vs_h_beg_e_pos[5]	vs_h_beg_e_pos[4]	vs_h_beg_e_pos[3]
0x1A 81	0x00	vid - cea-861 timing control_9	rw	vs_h_beg_e_pos[2]	vs_h_beg_e_pos[1]	vs_h_beg_e_pos[0]	fld_cap_pos	vs_v_beg_pos[5]	vs_v_beg_pos[4]	vs_v_beg_pos[3]	vs_v_beg_pos[2]
0x1A 82	0x4C	vid - cea-861 timing control_10	rw	vs_v_beg_pos[1]	vs_v_beg_pos[0]	vs_v_end_pos[5]	vs_v_end_pos[4]	vs_v_end_pos[3]	vs_v_end_pos[2]	vs_v_end_pos[1]	vs_v_end_pos[0]
0x1A 85	0x85	hps control	rw	hps_power_down	-	-	hps_filt_bypass	hps_bypass_downsample	hps_phase_sel_downsample	hps_filt_mode[1]	hps_filt_mode[0]
0x1A 87	0x00	arc 1 control_1	rw	tx1_arc_powerdown	-	-	-	-	-	-	-
0x1A 88	0x00	arc 1 control_2	rw	-	-	-	-	-	-	tx1_arc_bias_hyst_adj	tx1_arc_s_end_hpd
0x1A 89	0x00	arc 2 control_1	rw	tx2_arc_powerdown	-	-	-	-	-	-	-
0x1A 8A	0x00	arc 2 control_2	rw	-	-	-	-	-	-	tx2_arc_bias_hyst_adj	tx2_arc_s_end_hpd
0x1A 94	0x00	dpll control_17	rw	-	-	-	pvsp_track_offset[20]	pvsp_track_offset[19]	pvsp_track_offset[18]	pvsp_track_offset[17]	pvsp_track_offset[16]
0x1A 95	0x00	dpll control_18	rw	pvsp_track_offset[15]	pvsp_track_offset[14]	pvsp_track_offset[13]	pvsp_track_offset[12]	pvsp_track_offset[11]	pvsp_track_offset[10]	pvsp_track_offset[9]	pvsp_track_offset[8]
0x1A 96	0x00	dpll control_19	rw	pvsp_track_offset[7]	pvsp_track_offset[6]	pvsp_track_offset[5]	pvsp_track_offset[4]	pvsp_track_offset[3]	pvsp_track_offset[2]	pvsp_track_offset[1]	pvsp_track_offset[0]
0x1A 97	0x00	dpll control_20	rw	-	-	-	svsp_track_offset[20]	svsp_track_offset[19]	svsp_track_offset[18]	svsp_track_offset[17]	svsp_track_offset[16]
0x1A 98	0x00	dpll control_21	rw	svsp_track_offset[15]	svsp_track_offset[14]	svsp_track_offset[13]	svsp_track_offset[12]	svsp_track_offset[11]	svsp_track_offset[10]	svsp_track_offset[9]	svsp_track_offset[8]
0x1A 99	0x00	dpll control_22	rw	svsp_track_offset[7]	svsp_track_offset[6]	svsp_track_offset[5]	svsp_track_offset[4]	svsp_track_offset[3]	svsp_track_offset[2]	svsp_track_offset[1]	svsp_track_offset[0]
0x1A 9D	0x00	ext mem control_1	rw	ck_drv_str[7]	ck_drv_str[6]	ck_drv_str[5]	ck_drv_str[4]	ck_drv_str[3]	ck_drv_str[2]	ck_drv_str[1]	ck_drv_str[0]
0x1A 9E	0x00	ext mem control_2	rw	dqs_drv_str[7]	dqs_drv_str[6]	dqs_drv_str[5]	dqs_drv_str[4]	dqs_drv_str[3]	dqs_drv_str[2]	dqs_drv_str[1]	dqs_drv_str[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1AA2	0x48	ddr2 pll / dll control_4	rw	-	-	plldll_sel_div[5]	plldll_sel_div[4]	plldll_sel_div[3]	plldll_sel_div[2]	plldll_sel_div[1]	plldll_sel_div[0]
0x1AA3	0x11	ddr2 pll / dll control_5	rw	-	-	-	-	plldll_pre_div[1]	plldll_pre_div[0]	-	-
0x1AA7	0x06	ext mem control_3	rw	-	bit_err_rb_en	dm_dq_drv_str[1]	dm_dq_drv_str[0]	-	-	-	mem_test_en
0x1AA8	0x00	ext mem control_4	rw	rw_ctrl_oe	ddr2_ck_oe	-	-	-	-	mem_rw_ctrl_drv_str[1]	mem_rw_ctrl_drv_str[0]
0x1AB2	0x00	ext mem control_10	rw	-	-	-	-	-	-	1x_ddr	-
0x1AB6	0x00	debug control_3	rw	-	-	spi_loop_through	-	-	-	-	-
0x1ACA	0x00	output enable control_11	rw	arc_pins_oe_man	-	-	-	-	-	-	-
0x1ACB	0x00	output enable control_12	rw	arc_pins_oe_man_en	-	-	-	-	-	-	-
0x1ACC	0x80	output enable control_13	rw	-	int_pin_oe[2]	int_pin_oe[1]	int_pin_oe[0]	-	int_pin_od_en[2]	int_pin_od_en[1]	int_pin_od_en[0]
0x1ACD	0x00	output enable control_14	rw	spi1_cs_oe_man	spi1_miso_oe_man	spi1_mosi_oe_man	spi1_sclk_oe_man	spi2_cs_oe_man	spi2_miso_oe_man	spi2_mosi_oe_man	spi2_sclk_oe_man
0x1ACE	0xFF	output enable control_15	rw	spi1_cs_oe_man_en	spi1_miso_oe_man_en	spi1_mosi_oe_man_en	spi1_sclk_oe_man_en	spi2_cs_oe_man_en	spi2_miso_oe_man_en	spi2_mosi_oe_man_en	spi2_sclk_oe_man_en
0x1AD0	0x00	chip id_1	r	rb_chip_id[15]	rb_chip_id[14]	rb_chip_id[13]	rb_chip_id[12]	rb_chip_id[11]	rb_chip_id[10]	rb_chip_id[9]	rb_chip_id[8]
0x1AD1	0x00	chip id_2	r	rb_chip_id[7]	rb_chip_id[6]	rb_chip_id[5]	rb_chip_id[4]	rb_chip_id[3]	rb_chip_id[2]	rb_chip_id[1]	rb_chip_id[0]
0x1AD3	0x00	vid_input_param_2	r	-	-	-	-	-	-	-	rb_chip_id[16]
0x1ADF	0x00	status	r	-	-	-	rb_rx_5v	rb_rx_tmds_clk_det	rb_rx_hpa_a	-	-
0x1AE1	0x00	ddr2 bist	r	-	-	-	-	-	-	lbk_test_result	lbk_test_done
0x1AE2	0x00	vs in measure_1	r	-	-	-	rb_v_in_xtal_clks[20]	rb_v_in_xtal_clks[19]	rb_v_in_xtal_clks[18]	rb_v_in_xtal_clks[17]	rb_v_in_xtal_clks[16]
0x1AE3	0x00	vs in measure_2	r	rb_v_in_xtal_clks[15]	rb_v_in_xtal_clks[14]	rb_v_in_xtal_clks[13]	rb_v_in_xtal_clks[12]	rb_v_in_xtal_clks[11]	rb_v_in_xtal_clks[10]	rb_v_in_xtal_clks[9]	rb_v_in_xtal_clks[8]
0x1AE4	0x00	vs in measure_3	r	rb_v_in_xtal_clks[7]	rb_v_in_xtal_clks[6]	rb_v_in_xtal_clks[5]	rb_v_in_xtal_clks[4]	rb_v_in_xtal_clks[3]	rb_v_in_xtal_clks[2]	rb_v_in_xtal_clks[1]	rb_v_in_xtal_clks[0]
0x1AF6	0x00	rb rx interrupt_1	r	rb_rx_intrq_raw	-	rx_cable_det_st	rx_tmdspll_lck_st	rx_tmds_clk_st	rx_video_3d_st	rx_av_mute_st	rx_hdmi_mode_st
0x1AF7	0x00	rb rx interrupt_2	r	rx_gen_ctl_pckt_st	rx_gamut_mdata_pckt_st	rx_isrc2_pckt_st	rx_isrc1_pckt_st	rx_vs_info_frm_st	rx_ms_info_frm_st	rx_spd_info_frm_st	rx_avi_info_frm_st
0x1AF8	0x00	rb rx interrupt_3	r	-	-	rx_vs_inf_cks_err_edge_st	rx_ms_inf_cks_err_edge_st	rx_spd_inf_cks_err_edge_st	rx_avi_inf_cks_err_edge_st	rx_deepcolor_chng_edge_st	rx_tmds_clk_chng_edge_st
0x1AF9	0x00	rb rx interrupt_4	r	rx_pkt_err_edge_st	rx_gamut_mdata_pckt_edge_st	rx_isrc2_pckt_edge_st	rx_isrc1_pckt_edge_st	rx_vs_info_frm_edge_st	rx_ms_info_frm_edge_st	rx_spd_info_frm_edge_st	rx_avi_info_frm_edge_st

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1A FA	0x00	rb rx interrupt_5	r	-	-	rx_vs_inf_cks_err_edge_raw	rx_ms_inf_cks_err_edge_raw	rx_spd_inf_cks_err_edge_raw	rx_avi_inf_cks_err_edge_raw	rx_deepcolor_chng_edge_raw	rx_tmds_clk_chng_edge_raw
0x1A FB	0x00	rb rx interrupt_6	r	rx_pkt_err_edge_raw	rx_gamut_mdata_pckt_edge_raw	rx_isrc2_pckt_edge_raw	rx_isrc1_pckt_edge_raw	rx_vs_info_frm_edge_raw	rx_ms_info_frm_edge_raw	rx_spd_info_frm_edge_raw	rx_avi_info_frm_edge_raw
0x1A FC	0x01	read auto inc address	rw	power_down	-	-	-	-	-	-	read_auto_inc_en
0x1A FD	0x00	resets_1	sc	svsp_reset	pvsp_reset	p2i_reset	ddr2_intf_reset	spi_reset	-	osd_reset	inp_sdr_reset
0x1A FE	0x00	resets_2	sc	rx_reset	enc_reset	tx2_reset	tx1_reset	-	dpll_reset	hps_reset	xtal_reset
0x1B 00	0x00	auto contrast adjust_9	rw	ace_frame_row_man_en	-	-	-	-	ace_frame_row_man[10]	ace_frame_row_man[9]	ace_frame_row_man[8]
0x1B 01	0x00	auto contrast adjust_10	rw	ace_frame_row_man[7]	ace_frame_row_man[6]	ace_frame_row_man[5]	ace_frame_row_man[4]	ace_frame_row_man[3]	ace_frame_row_man[2]	ace_frame_row_man[1]	ace_frame_row_man[0]
0x1B 02	0x00	auto contrast adjust_11	rw	ace_frame_col_man_en	-	-	-	-	ace_frame_col_man[10]	ace_frame_col_man[9]	ace_frame_col_man[8]
0x1B 03	0x00	auto contrast adjust_12	rw	ace_frame_col_man[7]	ace_frame_col_man[6]	ace_frame_col_man[5]	ace_frame_col_man[4]	ace_frame_col_man[3]	ace_frame_col_man[2]	ace_frame_col_man[1]	ace_frame_col_man[0]
0x1B 08	0x10	blanking area level_1	rw	vid_y_ba_lvl[7]	vid_y_ba_lvl[6]	vid_y_ba_lvl[5]	vid_y_ba_lvl[4]	vid_y_ba_lvl[3]	vid_y_ba_lvl[2]	vid_y_ba_lvl[1]	vid_y_ba_lvl[0]
0x1B 09	0x80	blanking area level_2	rw	vid_c_ba_lvl[7]	vid_c_ba_lvl[6]	vid_c_ba_lvl[5]	vid_c_ba_lvl[4]	vid_c_ba_lvl[3]	vid_c_ba_lvl[2]	vid_c_ba_lvl[1]	vid_c_ba_lvl[0]
0x1B 0A	0x10	blanking area level_3	rw	rx_y_ba_lvl[7]	rx_y_ba_lvl[6]	rx_y_ba_lvl[5]	rx_y_ba_lvl[4]	rx_y_ba_lvl[3]	rx_y_ba_lvl[2]	rx_y_ba_lvl[1]	rx_y_ba_lvl[0]
0x1B 0B	0x80	blanking area level_4	rw	rx_c_ba_lvl[7]	rx_c_ba_lvl[6]	rx_c_ba_lvl[5]	rx_c_ba_lvl[4]	rx_c_ba_lvl[3]	rx_c_ba_lvl[2]	rx_c_ba_lvl[1]	rx_c_ba_lvl[0]
0x1B 0C	0x10	blanking area level_5	rw	exosd_y_ba_lvl[7]	exosd_y_ba_lvl[6]	exosd_y_ba_lvl[5]	exosd_y_ba_lvl[4]	exosd_y_ba_lvl[3]	exosd_y_ba_lvl[2]	exosd_y_ba_lvl[1]	exosd_y_ba_lvl[0]
0x1B 0D	0x80	blanking area level_6	rw	exosd_c_ba_lvl[7]	exosd_c_ba_lvl[6]	exosd_c_ba_lvl[5]	exosd_c_ba_lvl[4]	exosd_c_ba_lvl[3]	exosd_c_ba_lvl[2]	exosd_c_ba_lvl[1]	exosd_c_ba_lvl[0]
0x1B 30	0x00	vid csc control_1	rw	vid_csc_enable	vid_csc_mode[1]	vid_csc_mode[0]	vid_a1[12]	vid_a1[11]	vid_a1[10]	vid_a1[9]	vid_a1[8]
0x1B 31	0x00	vid csc control_2	rw	vid_a1[7]	vid_a1[6]	vid_a1[5]	vid_a1[4]	vid_a1[3]	vid_a1[2]	vid_a1[1]	vid_a1[0]
0x1B 32	0x00	vid csc control_3	rw	-	-	-	vid_a2[12]	vid_a2[11]	vid_a2[10]	vid_a2[9]	vid_a2[8]
0x1B 33	0x00	vid csc control_4	rw	vid_a2[7]	vid_a2[6]	vid_a2[5]	vid_a2[4]	vid_a2[3]	vid_a2[2]	vid_a2[1]	vid_a2[0]
0x1B 34	0x00	vid csc control_5	rw	-	-	-	vid_a3[12]	vid_a3[11]	vid_a3[10]	vid_a3[9]	vid_a3[8]
0x1B 35	0x00	vid csc control_6	rw	vid_a3[7]	vid_a3[6]	vid_a3[5]	vid_a3[4]	vid_a3[3]	vid_a3[2]	vid_a3[1]	vid_a3[0]
0x1B 36	0x00	vid csc control_7	rw	-	-	-	vid_a4[12]	vid_a4[11]	vid_a4[10]	vid_a4[9]	vid_a4[8]
0x1B 37	0x00	vid csc control_8	rw	vid_a4[7]	vid_a4[6]	vid_a4[5]	vid_a4[4]	vid_a4[3]	vid_a4[2]	vid_a4[1]	vid_a4[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1B 38	0x00	vid csc control_9	rw	-	-	-	vid_b1[12]	vid_b1[11]	vid_b1[10]	vid_b1[9]	vid_b1[8]
0x1B 39	0x00	vid csc control_10	rw	vid_b1[7]	vid_b1[6]	vid_b1[5]	vid_b1[4]	vid_b1[3]	vid_b1[2]	vid_b1[1]	vid_b1[0]
0x1B 3A	0x00	vid csc control_11	rw	-	-	-	vid_b2[12]	vid_b2[11]	vid_b2[10]	vid_b2[9]	vid_b2[8]
0x1B 3B	0x00	vid csc control_12	rw	vid_b2[7]	vid_b2[6]	vid_b2[5]	vid_b2[4]	vid_b2[3]	vid_b2[2]	vid_b2[1]	vid_b2[0]
0x1B 3C	0x00	vid csc control_13	rw	-	-	-	vid_b3[12]	vid_b3[11]	vid_b3[10]	vid_b3[9]	vid_b3[8]
0x1B 3D	0x00	vid csc control_14	rw	vid_b3[7]	vid_b3[6]	vid_b3[5]	vid_b3[4]	vid_b3[3]	vid_b3[2]	vid_b3[1]	vid_b3[0]
0x1B 3E	0x00	vid csc control_15	rw	-	-	-	vid_b4[12]	vid_b4[11]	vid_b4[10]	vid_b4[9]	vid_b4[8]
0x1B 3F	0x00	vid csc control_16	rw	vid_b4[7]	vid_b4[6]	vid_b4[5]	vid_b4[4]	vid_b4[3]	vid_b4[2]	vid_b4[1]	vid_b4[0]
0x1B 40	0x00	vid csc control_17	rw	-	-	-	vid_c1[12]	vid_c1[11]	vid_c1[10]	vid_c1[9]	vid_c1[8]
0x1B 41	0x00	vid csc control_18	rw	vid_c1[7]	vid_c1[6]	vid_c1[5]	vid_c1[4]	vid_c1[3]	vid_c1[2]	vid_c1[1]	vid_c1[0]
0x1B 42	0x00	vid csc control_19	rw	-	-	-	vid_c2[12]	vid_c2[11]	vid_c2[10]	vid_c2[9]	vid_c2[8]
0x1B 43	0x00	vid csc control_20	rw	vid_c2[7]	vid_c2[6]	vid_c2[5]	vid_c2[4]	vid_c2[3]	vid_c2[2]	vid_c2[1]	vid_c2[0]
0x1B 44	0x00	vid csc control_21	rw	-	-	-	vid_c3[12]	vid_c3[11]	vid_c3[10]	vid_c3[9]	vid_c3[8]
0x1B 45	0x00	vid csc control_22	rw	vid_c3[7]	vid_c3[6]	vid_c3[5]	vid_c3[4]	vid_c3[3]	vid_c3[2]	vid_c3[1]	vid_c3[0]
0x1B 46	0x00	vid csc control_23	rw	-	-	-	vid_c4[12]	vid_c4[11]	vid_c4[10]	vid_c4[9]	vid_c4[8]
0x1B 47	0x00	vid csc control_24	rw	vid_c4[7]	vid_c4[6]	vid_c4[5]	vid_c4[4]	vid_c4[3]	vid_c4[2]	vid_c4[1]	vid_c4[0]
0x1B 48	0x08	input timing control_1	rw	vid_swap_bus_ctrl[2]	vid_swap_bus_ctrl[1]	vid_swap_bus_ctrl[0]	vid_format_sel[4]	vid_format_sel[3]	vid_format_sel[2]	vid_format_sel[1]	vid_format_sel[0]
0x1B 49	0x20	input timing control_2	rw	vid_swap_cb_cr_422	vid_ps444_r444_conv	vid_blank_blanking_area	vid_lsb_killed	vid_hs_pol	vid_vs_pol	vid_de_pol	vid_fld_pol
0x1B 4A	0x00	input control	rw	-	-	-	-	vid_ddr_edge_sel	vid_ud_bypass_man_en	vid_ud_bypass_man	vid_ddr_yc_swap
0x1B 4B	0x80	input timing control_3	rw	vid_hs_vs_mode	-	-	vid_rev_bus	vid_av_pos_sel	vid_av_split_code	vid_av_codes_rep_man_en	vid_av_codes_rep_man
0x1B 50	0x00	osd csc control_1	rw	exosd_csc_enable	exosd_csc_mode[1]	exosd_csc_mode[0]	exosd_a1[12]	exosd_a1[11]	exosd_a1[10]	exosd_a1[9]	exosd_a1[8]
0x1B 51	0x00	osd csc control_2	rw	exosd_a1[7]	exosd_a1[6]	exosd_a1[5]	exosd_a1[4]	exosd_a1[3]	exosd_a1[2]	exosd_a1[1]	exosd_a1[0]
0x1B 52	0x00	osd csc control_3	rw	-	-	-	exosd_a2[12]	exosd_a2[11]	exosd_a2[10]	exosd_a2[9]	exosd_a2[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1B 53	0x00	osd csc control_4	rw	exosd_a2[7]	exosd_a2[6]	exosd_a2[5]	exosd_a2[4]	exosd_a2[3]	exosd_a2[2]	exosd_a2[1]	exosd_a2[0]
0x1B 54	0x00	osd csc control_5	rw	-	-	-	exosd_a3[12]	exosd_a3[11]	exosd_a3[10]	exosd_a3[9]	exosd_a3[8]
0x1B 55	0x00	osd csc control_6	rw	exosd_a3[7]	exosd_a3[6]	exosd_a3[5]	exosd_a3[4]	exosd_a3[3]	exosd_a3[2]	exosd_a3[1]	exosd_a3[0]
0x1B 56	0x00	osd csc control_7	rw	-	-	-	exosd_a4[12]	exosd_a4[11]	exosd_a4[10]	exosd_a4[9]	exosd_a4[8]
0x1B 57	0x00	osd csc control_8	rw	exosd_a4[7]	exosd_a4[6]	exosd_a4[5]	exosd_a4[4]	exosd_a4[3]	exosd_a4[2]	exosd_a4[1]	exosd_a4[0]
0x1B 58	0x00	osd csc control_9	rw	-	-	-	exosd_b1[12]	exosd_b1[11]	exosd_b1[10]	exosd_b1[9]	exosd_b1[8]
0x1B 59	0x00	osd csc control_10	rw	exosd_b1[7]	exosd_b1[6]	exosd_b1[5]	exosd_b1[4]	exosd_b1[3]	exosd_b1[2]	exosd_b1[1]	exosd_b1[0]
0x1B 5A	0x00	osd csc control_11	rw	-	-	-	exosd_b2[12]	exosd_b2[11]	exosd_b2[10]	exosd_b2[9]	exosd_b2[8]
0x1B 5B	0x00	osd csc control_12	rw	exosd_b2[7]	exosd_b2[6]	exosd_b2[5]	exosd_b2[4]	exosd_b2[3]	exosd_b2[2]	exosd_b2[1]	exosd_b2[0]
0x1B 5C	0x00	osd csc control_13	rw	-	-	-	exosd_b3[12]	exosd_b3[11]	exosd_b3[10]	exosd_b3[9]	exosd_b3[8]
0x1B 5D	0x00	osd csc control_14	rw	exosd_b3[7]	exosd_b3[6]	exosd_b3[5]	exosd_b3[4]	exosd_b3[3]	exosd_b3[2]	exosd_b3[1]	exosd_b3[0]
0x1B 5E	0x00	osd csc control_15	rw	-	-	-	exosd_b4[12]	exosd_b4[11]	exosd_b4[10]	exosd_b4[9]	exosd_b4[8]
0x1B 5F	0x00	osd csc control_16	rw	exosd_b4[7]	exosd_b4[6]	exosd_b4[5]	exosd_b4[4]	exosd_b4[3]	exosd_b4[2]	exosd_b4[1]	exosd_b4[0]
0x1B 60	0x00	osd csc control_17	rw	-	-	-	exosd_c1[12]	exosd_c1[11]	exosd_c1[10]	exosd_c1[9]	exosd_c1[8]
0x1B 61	0x00	osd csc control_18	rw	exosd_c1[7]	exosd_c1[6]	exosd_c1[5]	exosd_c1[4]	exosd_c1[3]	exosd_c1[2]	exosd_c1[1]	exosd_c1[0]
0x1B 62	0x00	osd csc control_19	rw	-	-	-	exosd_c2[12]	exosd_c2[11]	exosd_c2[10]	exosd_c2[9]	exosd_c2[8]
0x1B 63	0x00	osd csc control_20	rw	exosd_c2[7]	exosd_c2[6]	exosd_c2[5]	exosd_c2[4]	exosd_c2[3]	exosd_c2[2]	exosd_c2[1]	exosd_c2[0]
0x1B 64	0x00	osd csc control_21	rw	-	-	-	exosd_c3[12]	exosd_c3[11]	exosd_c3[10]	exosd_c3[9]	exosd_c3[8]
0x1B 65	0x00	osd csc control_22	rw	exosd_c3[7]	exosd_c3[6]	exosd_c3[5]	exosd_c3[4]	exosd_c3[3]	exosd_c3[2]	exosd_c3[1]	exosd_c3[0]
0x1B 66	0x00	osd csc control_23	rw	-	-	-	exosd_c4[12]	exosd_c4[11]	exosd_c4[10]	exosd_c4[9]	exosd_c4[8]
0x1B 67	0x00	osd csc control_24	rw	exosd_c4[7]	exosd_c4[6]	exosd_c4[5]	exosd_c4[4]	exosd_c4[3]	exosd_c4[2]	exosd_c4[1]	exosd_c4[0]
0x1B 68	0x0C	osd input timing control_1	rw	exosd_swap_bus_ctrl[2]	exosd_swap_bus_ctrl[1]	exosd_swap_bus_ctrl[0]	exosd_format_sel[4]	exosd_format_sel[3]	exosd_format_sel[2]	exosd_format_sel[1]	exosd_format_sel[0]
0x1B 69	0x20	osd input timing control_2	rw	exosd_swap_cb_cr_422	exosd_ps444_r444_conv	exosd_blank_blanking_area	exosd_lsb_killed	exosd_fld_pol	exosd_de_pol	exosd_vs_pol	exosd_hs_pol

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1B 6A	0x00	osd input control	rw	-	-	-	-	exosd_ddr_edge_sel	exosd_ud_bypass_man_en	exosd_ud_bypass_man	exosd_ddr_yc_swap
0x1B 6B	0x80	osd input timing control_3	rw	exosd_hs_vs_mode	exosd_2x_1x_use_ll_man	exosd_2x_1x_use_ll_man_en	exosd_rev_bus	exosd_av_pos_sel	exosd_av_split_code	exosd_av_codes_rep_man_en	exosd_av_codes_rep_man
0x1B 6C	0xFE	osd input timing control_4	rw	exosd_in_id[7]	exosd_in_id[6]	exosd_in_id[5]	exosd_in_id[4]	exosd_in_id[3]	exosd_in_id[2]	exosd_in_id[1]	exosd_in_id[0]
0x1B 6D	0x09	osd input timing control_5	rw	-	-	-	-	exosd_alpha_fmt[3]	exosd_alpha_fmt[2]	exosd_alpha_fmt[1]	exosd_alpha_fmt[0]
0x1B 70	0x00	rx csc control_1	rw	rx_csc_enable	rx_csc_mode[1]	rx_csc_mode[0]	rx_a1[12]	rx_a1[11]	rx_a1[10]	rx_a1[9]	rx_a1[8]
0x1B 71	0x00	rx csc control_2	rw	rx_a1[7]	rx_a1[6]	rx_a1[5]	rx_a1[4]	rx_a1[3]	rx_a1[2]	rx_a1[1]	rx_a1[0]
0x1B 72	0x00	rx csc control_3	rw	-	-	-	rx_a2[12]	rx_a2[11]	rx_a2[10]	rx_a2[9]	rx_a2[8]
0x1B 73	0x00	rx csc control_4	rw	rx_a2[7]	rx_a2[6]	rx_a2[5]	rx_a2[4]	rx_a2[3]	rx_a2[2]	rx_a2[1]	rx_a2[0]
0x1B 74	0x00	rx csc control_5	rw	-	-	-	rx_a3[12]	rx_a3[11]	rx_a3[10]	rx_a3[9]	rx_a3[8]
0x1B 75	0x00	rx csc control_6	rw	rx_a3[7]	rx_a3[6]	rx_a3[5]	rx_a3[4]	rx_a3[3]	rx_a3[2]	rx_a3[1]	rx_a3[0]
0x1B 76	0x00	rx csc control_7	rw	-	-	-	rx_a4[12]	rx_a4[11]	rx_a4[10]	rx_a4[9]	rx_a4[8]
0x1B 77	0x00	rx csc control_8	rw	rx_a4[7]	rx_a4[6]	rx_a4[5]	rx_a4[4]	rx_a4[3]	rx_a4[2]	rx_a4[1]	rx_a4[0]
0x1B 78	0x00	rx csc control_9	rw	-	-	-	rx_b1[12]	rx_b1[11]	rx_b1[10]	rx_b1[9]	rx_b1[8]
0x1B 79	0x00	rx csc control_10	rw	rx_b1[7]	rx_b1[6]	rx_b1[5]	rx_b1[4]	rx_b1[3]	rx_b1[2]	rx_b1[1]	rx_b1[0]
0x1B 7A	0x00	rx csc control_11	rw	-	-	-	rx_b2[12]	rx_b2[11]	rx_b2[10]	rx_b2[9]	rx_b2[8]
0x1B 7B	0x00	rx csc control_12	rw	rx_b2[7]	rx_b2[6]	rx_b2[5]	rx_b2[4]	rx_b2[3]	rx_b2[2]	rx_b2[1]	rx_b2[0]
0x1B 7C	0x00	rx csc control_13	rw	-	-	-	rx_b3[12]	rx_b3[11]	rx_b3[10]	rx_b3[9]	rx_b3[8]
0x1B 7D	0x00	rx csc control_14	rw	rx_b3[7]	rx_b3[6]	rx_b3[5]	rx_b3[4]	rx_b3[3]	rx_b3[2]	rx_b3[1]	rx_b3[0]
0x1B 7E	0x00	rx csc control_15	rw	-	-	-	rx_b4[12]	rx_b4[11]	rx_b4[10]	rx_b4[9]	rx_b4[8]
0x1B 7F	0x00	rx csc control_16	rw	rx_b4[7]	rx_b4[6]	rx_b4[5]	rx_b4[4]	rx_b4[3]	rx_b4[2]	rx_b4[1]	rx_b4[0]
0x1B 80	0x00	rx csc control_17	rw	-	-	-	rx_c1[12]	rx_c1[11]	rx_c1[10]	rx_c1[9]	rx_c1[8]
0x1B 81	0x00	rx csc control_18	rw	rx_c1[7]	rx_c1[6]	rx_c1[5]	rx_c1[4]	rx_c1[3]	rx_c1[2]	rx_c1[1]	rx_c1[0]
0x1B 82	0x00	rx csc control_19	rw	-	-	-	rx_c2[12]	rx_c2[11]	rx_c2[10]	rx_c2[9]	rx_c2[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1B 83	0x00	rx csc control_20	rw	rx_c2[7]	rx_c2[6]	rx_c2[5]	rx_c2[4]	rx_c2[3]	rx_c2[2]	rx_c2[1]	rx_c2[0]
0x1B 84	0x00	rx csc control_21	rw	-	-	-	rx_c3[12]	rx_c3[11]	rx_c3[10]	rx_c3[9]	rx_c3[8]
0x1B 85	0x00	rx csc control_22	rw	rx_c3[7]	rx_c3[6]	rx_c3[5]	rx_c3[4]	rx_c3[3]	rx_c3[2]	rx_c3[1]	rx_c3[0]
0x1B 86	0x00	rx csc control_23	rw	-	-	-	rx_c4[12]	rx_c4[11]	rx_c4[10]	rx_c4[9]	rx_c4[8]
0x1B 87	0x00	rx csc control_24	rw	rx_c4[7]	rx_c4[6]	rx_c4[5]	rx_c4[4]	rx_c4[3]	rx_c4[2]	rx_c4[1]	rx_c4[0]
0x1B 88	0x08	rx input timing_1	rw	rx_swap_bus_ctrl[2]	rx_swap_bus_ctrl[1]	rx_swap_bus_ctrl[0]	rx_format_sel[4]	rx_format_sel[3]	rx_format_sel[2]	rx_format_sel[1]	rx_format_sel[0]
0x1B 89	0x00	rx input timing_2	rw	rx_blank_blanking_area	rx_swap_cb_cr_422	rx_ps444_r444_conv	rx_lsb_killed	rx_fld_pol	rx_de_pol	rx_vs_pol	rx_hs_pol
0x1B 8A	0x00	rx input control	rw	-	-	-	-	-	rx_ud_bypass_man_en	rx_ud_bypass_man	-
0x1B 8B	0x00	rx input timing_3	rw	de_h_beg_pos[11]	rx_2x_1x_use_ll_man	rx_2x_1x_use_ll_man_en	de_h_beg_pos[10]	hs_beg_pos[11]	hs_beg_pos[10]	hs_end_pos[11]	hs_end_pos[10]
0x1B 8C	0xB6	osd - cea-861 timing control_1	rw	de_v_beg_e_pos[6]	de_v_beg_e_pos[5]	de_v_beg_e_pos[4]	de_v_beg_e_pos[3]	de_v_beg_e_pos[2]	de_v_beg_e_pos[1]	de_v_beg_e_pos[0]	de_v_beg_o_pos[6]
0x1B 8D	0xD0	osd - cea-861 timing control_2	rw	de_v_beg_o_pos[5]	de_v_beg_o_pos[4]	de_v_beg_o_pos[3]	de_v_beg_o_pos[2]	de_v_beg_o_pos[1]	de_v_beg_o_pos[0]	de_h_beg_pos[9]	de_h_beg_pos[8]
0x1B 8E	0xA0	osd - cea-861 timing control_3	rw	de_h_beg_pos[7]	de_h_beg_pos[6]	de_h_beg_pos[5]	de_h_beg_pos[4]	de_h_beg_pos[3]	de_h_beg_pos[2]	de_h_beg_pos[1]	de_h_beg_pos[0]
0x1B 8F	0x01	osd - cea-861 timing control_4	rw	hs_beg_pos[9]	hs_beg_pos[8]	hs_beg_pos[7]	hs_beg_pos[6]	hs_beg_pos[5]	hs_beg_pos[4]	hs_beg_pos[3]	hs_beg_pos[2]
0x1B 90	0xC0	osd - cea-861 timing control_5	rw	hs_beg_pos[1]	hs_beg_pos[0]	hs_end_pos[9]	hs_end_pos[8]	hs_end_pos[7]	hs_end_pos[6]	hs_end_pos[5]	hs_end_pos[4]
0x1B 91	0x00	osd - cea-861 timing control_6	rw	hs_end_pos[3]	hs_end_pos[2]	hs_end_pos[1]	hs_end_pos[0]	-	vs_h_beg_o_pos[10]	vs_h_beg_o_pos[9]	vs_h_beg_o_pos[8]
0x1B 92	0x10	osd - cea-861 timing control_7	rw	vs_h_beg_o_pos[7]	vs_h_beg_o_pos[6]	vs_h_beg_o_pos[5]	vs_h_beg_o_pos[4]	vs_h_beg_o_pos[3]	vs_h_beg_o_pos[2]	vs_h_beg_o_pos[1]	vs_h_beg_o_pos[0]
0x1B 93	0x0E	osd - cea-861 timing control_8	rw	vs_h_beg_e_pos[10]	vs_h_beg_e_pos[9]	vs_h_beg_e_pos[8]	vs_h_beg_e_pos[7]	vs_h_beg_e_pos[6]	vs_h_beg_e_pos[5]	vs_h_beg_e_pos[4]	vs_h_beg_e_pos[3]
0x1B 94	0x00	osd - cea-861 timing control_9	rw	vs_h_beg_e_pos[2]	vs_h_beg_e_pos[1]	vs_h_beg_e_pos[0]	fld_cap_pos	vs_v_beg_pos[5]	vs_v_beg_pos[4]	vs_v_beg_pos[3]	vs_v_beg_pos[2]
0x1B 95	0x4C	osd - cea-861 timing control_10	rw	vs_v_beg_pos[1]	vs_v_beg_pos[0]	vs_v_end_pos[5]	vs_v_end_pos[4]	vs_v_end_pos[3]	vs_v_end_pos[2]	vs_v_end_pos[1]	vs_v_end_pos[0]
0x1B 96	0x00	rx input timing_4	rw	rx_in_id[7]	rx_in_id[6]	rx_in_id[5]	rx_in_id[4]	rx_in_id[3]	rx_in_id[2]	rx_in_id[1]	rx_in_id[0]
0x1B 97	0x84	pvsp frtrk ctrl	rw	-	-	-	-	-	-	mp2i_frtrk_mas_fld	pvsp_frtrk_mas_mode_en
0x1B 99	0x84	svsp frtrk ctrl	rw	-	-	-	-	-	-	sp2i_frtrk_mas_fld	svsp_frtrk_mas_mode_en
0x1B 9B	0x00	mas ckg ctrl	rw	pvsp_mas_clk_in_en	-	-	-	svsp_mas_clk_in_en	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1BA3	0x00	tx1 clean up pll control_4	rw	-	-	-	-	-	-	osd_dout_drv_str[1]	osd_dout_drv_str[0]
0x1BA7	0x00	tx2 clean up pll control_4	rw	-	-	-	-	-	-	osd_clk_drv_str[1]	osd_clk_drv_str[0]
0x1BA8	0x00	rx crc control	rw	crc_rx_en	crc_rst	crc_interlaced	-	-	crc_rb_sel[2]	crc_rb_sel[1]	crc_rb_sel[0]
0x1BA9	0x00	rx crc control 2	rw	crc_num_frames[7]	crc_num_frames[6]	crc_num_frames[5]	crc_num_frames[4]	crc_num_frames[3]	crc_num_frames[2]	crc_num_frames[1]	crc_num_frames[0]
0x1BAA	0x00	rx crc checksum_1	r	rb_rx_crc_checksum[15]	rb_rx_crc_checksum[14]	rb_rx_crc_checksum[13]	rb_rx_crc_checksum[12]	rb_rx_crc_checksum[11]	rb_rx_crc_checksum[10]	rb_rx_crc_checksum[9]	rb_rx_crc_checksum[8]
0x1BAB	0x00	rx crc checksum_2	r	rb_rx_crc_checksum[7]	rb_rx_crc_checksum[6]	rb_rx_crc_checksum[5]	rb_rx_crc_checksum[4]	rb_rx_crc_checksum[3]	rb_rx_crc_checksum[2]	rb_rx_crc_checksum[1]	rb_rx_crc_checksum[0]
0x1BAC	0x00	rx crc status	r	-	-	-	-	-	-	-	rb_rx_crc_ready
0x1BB0	0x00	ttl out csc control_1	rw	ttl_out_csc_enable	ttl_out_csc_mode[1]	ttl_out_csc_mode[0]	ttl_out_a1[12]	ttl_out_a1[11]	ttl_out_a1[10]	ttl_out_a1[9]	ttl_out_a1[8]
0x1BB1	0x00	ttl out csc control_2	rw	ttl_out_a1[7]	ttl_out_a1[6]	ttl_out_a1[5]	ttl_out_a1[4]	ttl_out_a1[3]	ttl_out_a1[2]	ttl_out_a1[1]	ttl_out_a1[0]
0x1BB2	0x00	ttl out csc control_3	rw	-	-	-	ttl_out_a2[12]	ttl_out_a2[11]	ttl_out_a2[10]	ttl_out_a2[9]	ttl_out_a2[8]
0x1BB3	0x00	ttl out csc control_4	rw	ttl_out_a2[7]	ttl_out_a2[6]	ttl_out_a2[5]	ttl_out_a2[4]	ttl_out_a2[3]	ttl_out_a2[2]	ttl_out_a2[1]	ttl_out_a2[0]
0x1BB4	0x00	ttl out csc control_5	rw	-	-	-	ttl_out_a3[12]	ttl_out_a3[11]	ttl_out_a3[10]	ttl_out_a3[9]	ttl_out_a3[8]
0x1BB5	0x00	ttl out csc control_6	rw	ttl_out_a3[7]	ttl_out_a3[6]	ttl_out_a3[5]	ttl_out_a3[4]	ttl_out_a3[3]	ttl_out_a3[2]	ttl_out_a3[1]	ttl_out_a3[0]
0x1BB6	0x00	ttl out csc control_7	rw	-	-	-	ttl_out_a4[12]	ttl_out_a4[11]	ttl_out_a4[10]	ttl_out_a4[9]	ttl_out_a4[8]
0x1BB7	0x00	ttl out csc control_8	rw	ttl_out_a4[7]	ttl_out_a4[6]	ttl_out_a4[5]	ttl_out_a4[4]	ttl_out_a4[3]	ttl_out_a4[2]	ttl_out_a4[1]	ttl_out_a4[0]
0x1BB8	0x00	ttl out csc control_9	rw	-	-	-	ttl_out_b1[12]	ttl_out_b1[11]	ttl_out_b1[10]	ttl_out_b1[9]	ttl_out_b1[8]
0x1BB9	0x00	ttl out csc control_10	rw	ttl_out_b1[7]	ttl_out_b1[6]	ttl_out_b1[5]	ttl_out_b1[4]	ttl_out_b1[3]	ttl_out_b1[2]	ttl_out_b1[1]	ttl_out_b1[0]
0x1BBA	0x00	ttl out csc control_11	rw	-	-	-	ttl_out_b2[12]	ttl_out_b2[11]	ttl_out_b2[10]	ttl_out_b2[9]	ttl_out_b2[8]
0x1BBB	0x00	ttl out csc control_12	rw	ttl_out_b2[7]	ttl_out_b2[6]	ttl_out_b2[5]	ttl_out_b2[4]	ttl_out_b2[3]	ttl_out_b2[2]	ttl_out_b2[1]	ttl_out_b2[0]
0x1BBC	0x00	ttl out csc control_13	rw	-	-	-	ttl_out_b3[12]	ttl_out_b3[11]	ttl_out_b3[10]	ttl_out_b3[9]	ttl_out_b3[8]
0x1BBD	0x00	ttl out csc control_14	rw	ttl_out_b3[7]	ttl_out_b3[6]	ttl_out_b3[5]	ttl_out_b3[4]	ttl_out_b3[3]	ttl_out_b3[2]	ttl_out_b3[1]	ttl_out_b3[0]
0x1BBE	0x00	ttl out csc control_15	rw	-	-	-	ttl_out_b4[12]	ttl_out_b4[11]	ttl_out_b4[10]	ttl_out_b4[9]	ttl_out_b4[8]
0x1BBF	0x00	ttl out csc control_16	rw	ttl_out_b4[7]	ttl_out_b4[6]	ttl_out_b4[5]	ttl_out_b4[4]	ttl_out_b4[3]	ttl_out_b4[2]	ttl_out_b4[1]	ttl_out_b4[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1BC0	0x00	ttl out csc control_17	rw	-	-	-	ttl_out_c1[12]	ttl_out_c1[11]	ttl_out_c1[10]	ttl_out_c1[9]	ttl_out_c1[8]
0x1BC1	0x00	ttl out csc control_18	rw	ttl_out_c1[7]	ttl_out_c1[6]	ttl_out_c1[5]	ttl_out_c1[4]	ttl_out_c1[3]	ttl_out_c1[2]	ttl_out_c1[1]	ttl_out_c1[0]
0x1BC2	0x00	ttl out csc control_19	rw	-	-	-	ttl_out_c2[12]	ttl_out_c2[11]	ttl_out_c2[10]	ttl_out_c2[9]	ttl_out_c2[8]
0x1BC3	0x00	ttl out csc control_20	rw	ttl_out_c2[7]	ttl_out_c2[6]	ttl_out_c2[5]	ttl_out_c2[4]	ttl_out_c2[3]	ttl_out_c2[2]	ttl_out_c2[1]	ttl_out_c2[0]
0x1BC4	0x00	ttl out csc control_21	rw	-	-	-	ttl_out_c3[12]	ttl_out_c3[11]	ttl_out_c3[10]	ttl_out_c3[9]	ttl_out_c3[8]
0x1BC5	0x00	ttl out csc control_22	rw	ttl_out_c3[7]	ttl_out_c3[6]	ttl_out_c3[5]	ttl_out_c3[4]	ttl_out_c3[3]	ttl_out_c3[2]	ttl_out_c3[1]	ttl_out_c3[0]
0x1BC6	0x00	ttl out csc control_23	rw	-	-	-	ttl_out_c4[12]	ttl_out_c4[11]	ttl_out_c4[10]	ttl_out_c4[9]	ttl_out_c4[8]
0x1BC7	0x00	ttl out csc control_24	rw	ttl_out_c4[7]	ttl_out_c4[6]	ttl_out_c4[5]	ttl_out_c4[4]	ttl_out_c4[3]	ttl_out_c4[2]	ttl_out_c4[1]	ttl_out_c4[0]
0x1BC8	0x00	input enable control_1	rw	-	-	vid_clk_ie	clk_osd_ie	pix_pins_ie[35]	pix_pins_ie[34]	pix_pins_ie[33]	pix_pins_ie[32]
0x1BC9	0x00	input enable control_2	rw	pix_pins_ie[31]	pix_pins_ie[30]	pix_pins_ie[29]	pix_pins_ie[28]	pix_pins_ie[27]	pix_pins_ie[26]	pix_pins_ie[25]	pix_pins_ie[24]
0x1BCA	0x00	input enable control_3	rw	pix_pins_ie[23]	pix_pins_ie[22]	pix_pins_ie[21]	pix_pins_ie[20]	pix_pins_ie[19]	pix_pins_ie[18]	pix_pins_ie[17]	pix_pins_ie[16]
0x1BCB	0x00	input enable control_4	rw	pix_pins_ie[15]	pix_pins_ie[14]	pix_pins_ie[13]	pix_pins_ie[12]	pix_pins_ie[11]	pix_pins_ie[10]	pix_pins_ie[9]	pix_pins_ie[8]
0x1BCC	0x00	input enable control_5	rw	pix_pins_ie[7]	pix_pins_ie[6]	pix_pins_ie[5]	pix_pins_ie[4]	pix_pins_ie[3]	pix_pins_ie[2]	pix_pins_ie[1]	pix_pins_ie[0]
0x1BCD	0x00	input enable control_6	rw	osd_pins_ie[23]	osd_pins_ie[22]	osd_pins_ie[21]	osd_pins_ie[20]	osd_pins_ie[19]	osd_pins_ie[18]	osd_pins_ie[17]	osd_pins_ie[16]
0x1BCE	0x00	input enable control_7	rw	osd_pins_ie[15]	osd_pins_ie[14]	osd_pins_ie[13]	osd_pins_ie[12]	osd_pins_ie[11]	osd_pins_ie[10]	osd_pins_ie[9]	osd_pins_ie[8]
0x1BCF	0x00	input enable control_8	rw	osd_pins_ie[7]	osd_pins_ie[6]	osd_pins_ie[5]	osd_pins_ie[4]	osd_pins_ie[3]	osd_pins_ie[2]	osd_pins_ie[1]	osd_pins_ie[0]
0x1BD0	0x00	input enable control_9	rw	hs_ie	vs_ie	de_ie	sfl_ie	-	hs_osd_ie	vs_osd_ie	de_osd_ie
0x1BD1	0x00	input enable control_10	rw	-	audio_pins_ie[6]	audio_pins_ie[5]	audio_pins_ie[4]	audio_pins_ie[3]	audio_pins_ie[2]	audio_pins_ie[1]	audio_pins_ie[0]
0x1BD2	0x00	input enable control_11	rw	arc1_pin_ie	arc2_pin_ie	int_pin_ie[2]	int_pin_ie[1]	int_pin_ie[0]	sclk_ie	mclk_ie	dsd_clk_ie
0x1BD3	0x00	input enable control_12	rw	spi1_cs_ie	spi1_miso_ie	spi1_mosi_ie	spi1_sclk_ie	spi2_cs_ie	spi2_miso_ie	spi2_mosi_ie	spi2_sclk_ie
0x1BD4	0x00	input enable control_13	rw	-	-	-	-	-	mas_clk_ie	mas_hs_ie	mas_vs_ie
0x1BE0	0x20	auto phase/position control	rw	auto_phpo_inp_sel[1]	auto_phpo_inp_sel[0]	auto_phpo_byp_csc	-	-	-	-	-
0x1BE1	0x00	auto phase control_1	rw	auto_ph_en	auto_ph_num[6]	auto_ph_num[5]	auto_ph_num[4]	auto_ph_num[3]	auto_ph_num[2]	auto_ph_num[1]	auto_ph_num[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0x1BE2	0x00	auto phase control_2	rw	-	-	auto_ph_scan[5]	auto_ph_scan[4]	auto_ph_scan[3]	auto_ph_scan[2]	auto_ph_scan[1]	auto_ph_scan[0]
0x1BE3	0x00	auto phase readback_1	r	rb_auto_ph_read_ready	-	rb_auto_ph_right_phase[5]	rb_auto_ph_right_phase[4]	rb_auto_ph_right_phase[3]	rb_auto_ph_right_phase[2]	rb_auto_ph_right_phase[1]	rb_auto_ph_right_phase[0]
0x1BE4	0x00	auto phase readback_2	r	rb_auto_ph_diff_sum_lock[23]	rb_auto_ph_diff_sum_lock[22]	rb_auto_ph_diff_sum_lock[21]	rb_auto_ph_diff_sum_lock[20]	rb_auto_ph_diff_sum_lock[19]	rb_auto_ph_diff_sum_lock[18]	rb_auto_ph_diff_sum_lock[17]	rb_auto_ph_diff_sum_lock[16]
0x1BE5	0x00	auto phase readback_3	r	rb_auto_ph_diff_sum_lock[15]	rb_auto_ph_diff_sum_lock[14]	rb_auto_ph_diff_sum_lock[13]	rb_auto_ph_diff_sum_lock[12]	rb_auto_ph_diff_sum_lock[11]	rb_auto_ph_diff_sum_lock[10]	rb_auto_ph_diff_sum_lock[9]	rb_auto_ph_diff_sum_lock[8]
0x1BE6	0x00	auto phase readback_4	r	rb_auto_ph_diff_sum_lock[7]	rb_auto_ph_diff_sum_lock[6]	rb_auto_ph_diff_sum_lock[5]	rb_auto_ph_diff_sum_lock[4]	rb_auto_ph_diff_sum_lock[3]	rb_auto_ph_diff_sum_lock[2]	rb_auto_ph_diff_sum_lock[1]	rb_auto_ph_diff_sum_lock[0]
0x1BE7	0x00	auto position control_1	rw	auto_po_en	-	-	-	-	-	auto_po_noise_thr[9]	auto_po_noise_thr[8]
0x1BE8	0x00	auto position control_2	rw	auto_po_noise_thr[7]	auto_po_noise_thr[6]	auto_po_noise_thr[5]	auto_po_noise_thr[4]	auto_po_noise_thr[3]	auto_po_noise_thr[2]	auto_po_noise_thr[1]	auto_po_noise_thr[0]
0x1BE9	0x00	auto position readback_1	r	-	-	-	-	-	-	rb_auto_po_l_edg_lock_flag	rb_auto_po_r_edg_lock_flag
0x1BEA	0x00	auto position readback_2	r	rb_auto_po_t_offset[15]	rb_auto_po_t_offset[14]	rb_auto_po_t_offset[13]	rb_auto_po_t_offset[12]	rb_auto_po_t_offset[11]	rb_auto_po_t_offset[10]	rb_auto_po_t_offset[9]	rb_auto_po_t_offset[8]
0x1BEB	0x00	auto position readback_3	r	rb_auto_po_t_offset[7]	rb_auto_po_t_offset[6]	rb_auto_po_t_offset[5]	rb_auto_po_t_offset[4]	rb_auto_po_t_offset[3]	rb_auto_po_t_offset[2]	rb_auto_po_t_offset[1]	rb_auto_po_t_offset[0]
0x1BEC	0x00	auto position readback_4	r	rb_auto_po_b_offset[15]	rb_auto_po_b_offset[14]	rb_auto_po_b_offset[13]	rb_auto_po_b_offset[12]	rb_auto_po_b_offset[11]	rb_auto_po_b_offset[10]	rb_auto_po_b_offset[9]	rb_auto_po_b_offset[8]
0x1BED	0x00	auto position readback_5	r	rb_auto_po_b_offset[7]	rb_auto_po_b_offset[6]	rb_auto_po_b_offset[5]	rb_auto_po_b_offset[4]	rb_auto_po_b_offset[3]	rb_auto_po_b_offset[2]	rb_auto_po_b_offset[1]	rb_auto_po_b_offset[0]
0x1BEE	0x00	auto position readback_6	r	rb_auto_po_l_offset[15]	rb_auto_po_l_offset[14]	rb_auto_po_l_offset[13]	rb_auto_po_l_offset[12]	rb_auto_po_l_offset[11]	rb_auto_po_l_offset[10]	rb_auto_po_l_offset[9]	rb_auto_po_l_offset[8]
0x1BEF	0x00	auto position readback_7	r	rb_auto_po_l_offset[7]	rb_auto_po_l_offset[6]	rb_auto_po_l_offset[5]	rb_auto_po_l_offset[4]	rb_auto_po_l_offset[3]	rb_auto_po_l_offset[2]	rb_auto_po_l_offset[1]	rb_auto_po_l_offset[0]
0x1BF0	0x00	auto position readback_8	r	rb_auto_po_r_offset[15]	rb_auto_po_r_offset[14]	rb_auto_po_r_offset[13]	rb_auto_po_r_offset[12]	rb_auto_po_r_offset[11]	rb_auto_po_r_offset[10]	rb_auto_po_r_offset[9]	rb_auto_po_r_offset[8]
0x1BF1	0x00	auto position readback_9	r	rb_auto_po_r_offset[7]	rb_auto_po_r_offset[6]	rb_auto_po_r_offset[5]	rb_auto_po_r_offset[4]	rb_auto_po_r_offset[3]	rb_auto_po_r_offset[2]	rb_auto_po_r_offset[1]	rb_auto_po_r_offset[0]
0x1BFF	0x00	resets_4	sc	main_reset	-	-	-	-	-	-	-

1.2 PRIMARY VSP MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE800	0x00	vsp_3d field buffer0 address_1	rw	pvsp_fieldbuffer0_addr[31]	pvsp_fieldbuffer0_addr[30]	pvsp_fieldbuffer0_addr[29]	pvsp_fieldbuffer0_addr[28]	pvsp_fieldbuffer0_addr[27]	pvsp_fieldbuffer0_addr[26]	pvsp_fieldbuffer0_addr[25]	pvsp_fieldbuffer0_addr[24]
0xE801	0x56	vsp_3d field buffer0 address_2	rw	pvsp_fieldbuffer0_addr[23]	pvsp_fieldbuffer0_addr[22]	pvsp_fieldbuffer0_addr[21]	pvsp_fieldbuffer0_addr[20]	pvsp_fieldbuffer0_addr[19]	pvsp_fieldbuffer0_addr[18]	pvsp_fieldbuffer0_addr[17]	pvsp_fieldbuffer0_addr[16]
0xE802	0x22	vsp_3d field buffer0 address_3	rw	pvsp_fieldbuffer0_addr[15]	pvsp_fieldbuffer0_addr[14]	pvsp_fieldbuffer0_addr[13]	pvsp_fieldbuffer0_addr[12]	pvsp_fieldbuffer0_addr[11]	pvsp_fieldbuffer0_addr[10]	pvsp_fieldbuffer0_addr[9]	pvsp_fieldbuffer0_addr[8]
0xE803	0x00	vsp_3d field buffer0 address_4	rw	pvsp_fieldbuffer0_addr[7]	pvsp_fieldbuffer0_addr[6]	pvsp_fieldbuffer0_addr[5]	pvsp_fieldbuffer0_addr[4]	pvsp_fieldbuffer0_addr[3]	pvsp_fieldbuffer0_addr[2]	pvsp_fieldbuffer0_addr[1]	pvsp_fieldbuffer0_addr[0]
0xE804	0x00	vsp_3d field buffer1 address_1	rw	pvsp_fieldbuffer1_addr[31]	pvsp_fieldbuffer1_addr[30]	pvsp_fieldbuffer1_addr[29]	pvsp_fieldbuffer1_addr[28]	pvsp_fieldbuffer1_addr[27]	pvsp_fieldbuffer1_addr[26]	pvsp_fieldbuffer1_addr[25]	pvsp_fieldbuffer1_addr[24]
0xE805	0xE2	vsp_3d field buffer1 address_2	rw	pvsp_fieldbuffer1_addr[23]	pvsp_fieldbuffer1_addr[22]	pvsp_fieldbuffer1_addr[21]	pvsp_fieldbuffer1_addr[20]	pvsp_fieldbuffer1_addr[19]	pvsp_fieldbuffer1_addr[18]	pvsp_fieldbuffer1_addr[17]	pvsp_fieldbuffer1_addr[16]
0xE806	0xC2	vsp_3d field buffer1 address_3	rw	pvsp_fieldbuffer1_addr[15]	pvsp_fieldbuffer1_addr[14]	pvsp_fieldbuffer1_addr[13]	pvsp_fieldbuffer1_addr[12]	pvsp_fieldbuffer1_addr[11]	pvsp_fieldbuffer1_addr[10]	pvsp_fieldbuffer1_addr[9]	pvsp_fieldbuffer1_addr[8]
0xE807	0x00	vsp_3d field buffer1 address_4	rw	pvsp_fieldbuffer1_addr[7]	pvsp_fieldbuffer1_addr[6]	pvsp_fieldbuffer1_addr[5]	pvsp_fieldbuffer1_addr[4]	pvsp_fieldbuffer1_addr[3]	pvsp_fieldbuffer1_addr[2]	pvsp_fieldbuffer1_addr[1]	pvsp_fieldbuffer1_addr[0]
0xE808	0x01	vsp_3d field buffer2 address_1	rw	pvsp_fieldbuffer2_addr[31]	pvsp_fieldbuffer2_addr[30]	pvsp_fieldbuffer2_addr[29]	pvsp_fieldbuffer2_addr[28]	pvsp_fieldbuffer2_addr[27]	pvsp_fieldbuffer2_addr[26]	pvsp_fieldbuffer2_addr[25]	pvsp_fieldbuffer2_addr[24]
0xE809	0x6F	vsp_3d field buffer2 address_2	rw	pvsp_fieldbuffer2_addr[23]	pvsp_fieldbuffer2_addr[22]	pvsp_fieldbuffer2_addr[21]	pvsp_fieldbuffer2_addr[20]	pvsp_fieldbuffer2_addr[19]	pvsp_fieldbuffer2_addr[18]	pvsp_fieldbuffer2_addr[17]	pvsp_fieldbuffer2_addr[16]
0xE80A	0x62	vsp_3d field buffer2 address_3	rw	pvsp_fieldbuffer2_addr[15]	pvsp_fieldbuffer2_addr[14]	pvsp_fieldbuffer2_addr[13]	pvsp_fieldbuffer2_addr[12]	pvsp_fieldbuffer2_addr[11]	pvsp_fieldbuffer2_addr[10]	pvsp_fieldbuffer2_addr[9]	pvsp_fieldbuffer2_addr[8]
0xE80B	0x00	vsp_3d field buffer2 address_4	rw	pvsp_fieldbuffer2_addr[7]	pvsp_fieldbuffer2_addr[6]	pvsp_fieldbuffer2_addr[5]	pvsp_fieldbuffer2_addr[4]	pvsp_fieldbuffer2_addr[3]	pvsp_fieldbuffer2_addr[2]	pvsp_fieldbuffer2_addr[1]	pvsp_fieldbuffer2_addr[0]
0xE80C	0x01	vsp_3d field buffer3 address_1	rw	pvsp_fieldbuffer3_addr[31]	pvsp_fieldbuffer3_addr[30]	pvsp_fieldbuffer3_addr[29]	pvsp_fieldbuffer3_addr[28]	pvsp_fieldbuffer3_addr[27]	pvsp_fieldbuffer3_addr[26]	pvsp_fieldbuffer3_addr[25]	pvsp_fieldbuffer3_addr[24]
0xE80D	0xFC	vsp_3d field buffer3 address_2	rw	pvsp_fieldbuffer3_addr[23]	pvsp_fieldbuffer3_addr[22]	pvsp_fieldbuffer3_addr[21]	pvsp_fieldbuffer3_addr[20]	pvsp_fieldbuffer3_addr[19]	pvsp_fieldbuffer3_addr[18]	pvsp_fieldbuffer3_addr[17]	pvsp_fieldbuffer3_addr[16]
0xE80E	0x02	vsp_3d field buffer3 address_3	rw	pvsp_fieldbuffer3_addr[15]	pvsp_fieldbuffer3_addr[14]	pvsp_fieldbuffer3_addr[13]	pvsp_fieldbuffer3_addr[12]	pvsp_fieldbuffer3_addr[11]	pvsp_fieldbuffer3_addr[10]	pvsp_fieldbuffer3_addr[9]	pvsp_fieldbuffer3_addr[8]
0xE80F	0x00	vsp_3d field buffer3 address_4	rw	pvsp_fieldbuffer3_addr[7]	pvsp_fieldbuffer3_addr[6]	pvsp_fieldbuffer3_addr[5]	pvsp_fieldbuffer3_addr[4]	pvsp_fieldbuffer3_addr[3]	pvsp_fieldbuffer3_addr[2]	pvsp_fieldbuffer3_addr[1]	pvsp_fieldbuffer3_addr[0]
0xE810	0x02	vsp_3d field buffer4 address_1	rw	pvsp_fieldbuffer4_addr[31]	pvsp_fieldbuffer4_addr[30]	pvsp_fieldbuffer4_addr[29]	pvsp_fieldbuffer4_addr[28]	pvsp_fieldbuffer4_addr[27]	pvsp_fieldbuffer4_addr[26]	pvsp_fieldbuffer4_addr[25]	pvsp_fieldbuffer4_addr[24]
0xE811	0x88	vsp_3d field buffer4 address_2	rw	pvsp_fieldbuffer4_addr[23]	pvsp_fieldbuffer4_addr[22]	pvsp_fieldbuffer4_addr[21]	pvsp_fieldbuffer4_addr[20]	pvsp_fieldbuffer4_addr[19]	pvsp_fieldbuffer4_addr[18]	pvsp_fieldbuffer4_addr[17]	pvsp_fieldbuffer4_addr[16]
0xE812	0xA2	vsp_3d field buffer4 address_3	rw	pvsp_fieldbuffer4_addr[15]	pvsp_fieldbuffer4_addr[14]	pvsp_fieldbuffer4_addr[13]	pvsp_fieldbuffer4_addr[12]	pvsp_fieldbuffer4_addr[11]	pvsp_fieldbuffer4_addr[10]	pvsp_fieldbuffer4_addr[9]	pvsp_fieldbuffer4_addr[8]
0xE813	0x00	vsp_3d field buffer4 address_4	rw	pvsp_fieldbuffer4_addr[7]	pvsp_fieldbuffer4_addr[6]	pvsp_fieldbuffer4_addr[5]	pvsp_fieldbuffer4_addr[4]	pvsp_fieldbuffer4_addr[3]	pvsp_fieldbuffer4_addr[2]	pvsp_fieldbuffer4_addr[1]	pvsp_fieldbuffer4_addr[0]
0xE814	0x03	vsp_3d field buffer5 address_1	rw	pvsp_fieldbuffer5_addr[31]	pvsp_fieldbuffer5_addr[30]	pvsp_fieldbuffer5_addr[29]	pvsp_fieldbuffer5_addr[28]	pvsp_fieldbuffer5_addr[27]	pvsp_fieldbuffer5_addr[26]	pvsp_fieldbuffer5_addr[25]	pvsp_fieldbuffer5_addr[24]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE815	0x15	vsp_3d field buffer5 address_2	rw	pvsp_fieldbuffer5_addr[23]	pvsp_fieldbuffer5_addr[22]	pvsp_fieldbuffer5_addr[21]	pvsp_fieldbuffer5_addr[20]	pvsp_fieldbuffer5_addr[19]	pvsp_fieldbuffer5_addr[18]	pvsp_fieldbuffer5_addr[17]	pvsp_fieldbuffer5_addr[16]
0xE816	0x42	vsp_3d field buffer5 address_3	rw	pvsp_fieldbuffer5_addr[15]	pvsp_fieldbuffer5_addr[14]	pvsp_fieldbuffer5_addr[13]	pvsp_fieldbuffer5_addr[12]	pvsp_fieldbuffer5_addr[11]	pvsp_fieldbuffer5_addr[10]	pvsp_fieldbuffer5_addr[9]	pvsp_fieldbuffer5_addr[8]
0xE817	0x00	vsp_3d field buffer5 address_4	rw	pvsp_fieldbuffer5_addr[7]	pvsp_fieldbuffer5_addr[6]	pvsp_fieldbuffer5_addr[5]	pvsp_fieldbuffer5_addr[4]	pvsp_fieldbuffer5_addr[3]	pvsp_fieldbuffer5_addr[2]	pvsp_fieldbuffer5_addr[1]	pvsp_fieldbuffer5_addr[0]
0xE818	0x00	vsp_3d motion buffer0 address_1	rw	pvsp_motionbuf0_addr[31]	pvsp_motionbuf0_addr[30]	pvsp_motionbuf0_addr[29]	pvsp_motionbuf0_addr[28]	pvsp_motionbuf0_addr[27]	pvsp_motionbuf0_addr[26]	pvsp_motionbuf0_addr[25]	pvsp_motionbuf0_addr[24]
0xE819	0x00	vsp_3d motion buffer0 address_2	rw	pvsp_motionbuf0_addr[23]	pvsp_motionbuf0_addr[22]	pvsp_motionbuf0_addr[21]	pvsp_motionbuf0_addr[20]	pvsp_motionbuf0_addr[19]	pvsp_motionbuf0_addr[18]	pvsp_motionbuf0_addr[17]	pvsp_motionbuf0_addr[16]
0xE81A	0x00	vsp_3d motion buffer0 address_3	rw	pvsp_motionbuf0_addr[15]	pvsp_motionbuf0_addr[14]	pvsp_motionbuf0_addr[13]	pvsp_motionbuf0_addr[12]	pvsp_motionbuf0_addr[11]	pvsp_motionbuf0_addr[10]	pvsp_motionbuf0_addr[9]	pvsp_motionbuf0_addr[8]
0xE81B	0x00	vsp_3d motion buffer0 address_4	rw	pvsp_motionbuf0_addr[7]	pvsp_motionbuf0_addr[6]	pvsp_motionbuf0_addr[5]	pvsp_motionbuf0_addr[4]	pvsp_motionbuf0_addr[3]	pvsp_motionbuf0_addr[2]	pvsp_motionbuf0_addr[1]	pvsp_motionbuf0_addr[0]
0xE81C	0x00	vsp_3d motion buffer1 address_1	rw	pvsp_motionbuf1_addr[31]	pvsp_motionbuf1_addr[30]	pvsp_motionbuf1_addr[29]	pvsp_motionbuf1_addr[28]	pvsp_motionbuf1_addr[27]	pvsp_motionbuf1_addr[26]	pvsp_motionbuf1_addr[25]	pvsp_motionbuf1_addr[24]
0xE81D	0x07	vsp_3d motion buffer1 address_2	rw	pvsp_motionbuf1_addr[23]	pvsp_motionbuf1_addr[22]	pvsp_motionbuf1_addr[21]	pvsp_motionbuf1_addr[20]	pvsp_motionbuf1_addr[19]	pvsp_motionbuf1_addr[18]	pvsp_motionbuf1_addr[17]	pvsp_motionbuf1_addr[16]
0xE81E	0xE9	vsp_3d motion buffer1 address_3	rw	pvsp_motionbuf1_addr[15]	pvsp_motionbuf1_addr[14]	pvsp_motionbuf1_addr[13]	pvsp_motionbuf1_addr[12]	pvsp_motionbuf1_addr[11]	pvsp_motionbuf1_addr[10]	pvsp_motionbuf1_addr[9]	pvsp_motionbuf1_addr[8]
0xE81F	0x00	vsp_3d motion buffer1 address_4	rw	pvsp_motionbuf1_addr[7]	pvsp_motionbuf1_addr[6]	pvsp_motionbuf1_addr[5]	pvsp_motionbuf1_addr[4]	pvsp_motionbuf1_addr[3]	pvsp_motionbuf1_addr[2]	pvsp_motionbuf1_addr[1]	pvsp_motionbuf1_addr[0]
0xE820	0x00	vsp_3d rnr buffer0 address_1	rw	pvsp_rnr_buf0_addr[31]	pvsp_rnr_buf0_addr[30]	pvsp_rnr_buf0_addr[29]	pvsp_rnr_buf0_addr[28]	pvsp_rnr_buf0_addr[27]	pvsp_rnr_buf0_addr[26]	pvsp_rnr_buf0_addr[25]	pvsp_rnr_buf0_addr[24]
0xE821	0x0F	vsp_3d rnr buffer0 address_2	rw	pvsp_rnr_buf0_addr[23]	pvsp_rnr_buf0_addr[22]	pvsp_rnr_buf0_addr[21]	pvsp_rnr_buf0_addr[20]	pvsp_rnr_buf0_addr[19]	pvsp_rnr_buf0_addr[18]	pvsp_rnr_buf0_addr[17]	pvsp_rnr_buf0_addr[16]
0xE822	0xD2	vsp_3d rnr buffer0 address_3	rw	pvsp_rnr_buf0_addr[15]	pvsp_rnr_buf0_addr[14]	pvsp_rnr_buf0_addr[13]	pvsp_rnr_buf0_addr[12]	pvsp_rnr_buf0_addr[11]	pvsp_rnr_buf0_addr[10]	pvsp_rnr_buf0_addr[9]	pvsp_rnr_buf0_addr[8]
0xE823	0x00	vsp_3d rnr buffer0 address_4	rw	pvsp_rnr_buf0_addr[7]	pvsp_rnr_buf0_addr[6]	pvsp_rnr_buf0_addr[5]	pvsp_rnr_buf0_addr[4]	pvsp_rnr_buf0_addr[3]	pvsp_rnr_buf0_addr[2]	pvsp_rnr_buf0_addr[1]	pvsp_rnr_buf0_addr[0]
0xE824	0x00	vsp_3d rnr buffer1 address_1	rw	pvsp_rnr_buf1_addr[31]	pvsp_rnr_buf1_addr[30]	pvsp_rnr_buf1_addr[29]	pvsp_rnr_buf1_addr[28]	pvsp_rnr_buf1_addr[27]	pvsp_rnr_buf1_addr[26]	pvsp_rnr_buf1_addr[25]	pvsp_rnr_buf1_addr[24]
0xE825	0x32	vsp_3d rnr buffer1 address_2	rw	pvsp_rnr_buf1_addr[23]	pvsp_rnr_buf1_addr[22]	pvsp_rnr_buf1_addr[21]	pvsp_rnr_buf1_addr[20]	pvsp_rnr_buf1_addr[19]	pvsp_rnr_buf1_addr[18]	pvsp_rnr_buf1_addr[17]	pvsp_rnr_buf1_addr[16]
0xE826	0xFA	vsp_3d rnr buffer1 address_3	rw	pvsp_rnr_buf1_addr[15]	pvsp_rnr_buf1_addr[14]	pvsp_rnr_buf1_addr[13]	pvsp_rnr_buf1_addr[12]	pvsp_rnr_buf1_addr[11]	pvsp_rnr_buf1_addr[10]	pvsp_rnr_buf1_addr[9]	pvsp_rnr_buf1_addr[8]
0xE827	0x00	vsp_3d rnr buffer1 address_4	rw	pvsp_rnr_buf1_addr[7]	pvsp_rnr_buf1_addr[6]	pvsp_rnr_buf1_addr[5]	pvsp_rnr_buf1_addr[4]	pvsp_rnr_buf1_addr[3]	pvsp_rnr_buf1_addr[2]	pvsp_rnr_buf1_addr[1]	pvsp_rnr_buf1_addr[0]
0xE828	0x10	vsp_3d control_1	rw	-	-	pvsp_is_i_to_p	pvsp_update_vom	pvsp_lock_vom	pvsp_enable_vom	pvsp_enable_vim	pvsp_enable_ffs
0xE829	0x00	vsp_3d control_2	rw	pvsp_bypass	-	-	pvsp_ex_mem_data_format[1]	pvsp_ex_mem_data_format[0]	-	-	-
0xE82E	0x00	vsp_3d input horizontal resolution_1	rw	-	-	-	-	-	pvsp_vin_h[10]	pvsp_vin_h[9]	pvsp_vin_h[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE82F	0x00	vsp_3d input horizontal resolution_2	rw	pvsp_vin_h[7]	pvsp_vin_h[6]	pvsp_vin_h[5]	pvsp_vin_h[4]	pvsp_vin_h[3]	pvsp_vin_h[2]	pvsp_vin_h[1]	pvsp_vin_h[0]
0xE830	0x00	vsp_3d input vertical resolution_1	rw	-	-	-	-	-	pvsp_vin_v[10]	pvsp_vin_v[9]	pvsp_vin_v[8]
0xE831	0x00	vsp_3d input vertical resolution_2	rw	pvsp_vin_v[7]	pvsp_vin_v[6]	pvsp_vin_v[5]	pvsp_vin_v[4]	pvsp_vin_v[3]	pvsp_vin_v[2]	pvsp_vin_v[1]	pvsp_vin_v[0]
0xE832	0x00	vsp_3d vim crop h start_1	rw	-	-	-	-	-	pvsp_vim_crop_h_start[10]	pvsp_vim_crop_h_start[9]	pvsp_vim_crop_h_start[8]
0xE833	0x00	vsp_3d vim crop h start_2	rw	pvsp_vim_crop_h_start[7]	pvsp_vim_crop_h_start[6]	pvsp_vim_crop_h_start[5]	pvsp_vim_crop_h_start[4]	pvsp_vim_crop_h_start[3]	pvsp_vim_crop_h_start[2]	pvsp_vim_crop_h_start[1]	pvsp_vim_crop_h_start[0]
0xE834	0x00	vsp_3d vim crop v start_1	rw	-	-	-	-	-	pvsp_vim_crop_v_start[10]	pvsp_vim_crop_v_start[9]	pvsp_vim_crop_v_start[8]
0xE835	0x00	vsp_3d vim crop v start_2	rw	pvsp_vim_crop_v_start[7]	pvsp_vim_crop_v_start[6]	pvsp_vim_crop_v_start[5]	pvsp_vim_crop_v_start[4]	pvsp_vim_crop_v_start[3]	pvsp_vim_crop_v_start[2]	pvsp_vim_crop_v_start[1]	pvsp_vim_crop_v_start[0]
0xE836	0x00	vsp_3d vim crop width_1	rw	-	-	-	-	-	pvsp_vim_crop_width[10]	pvsp_vim_crop_width[9]	pvsp_vim_crop_width[8]
0xE837	0x00	vsp_3d vim crop width_2	rw	pvsp_vim_crop_width[7]	pvsp_vim_crop_width[6]	pvsp_vim_crop_width[5]	pvsp_vim_crop_width[4]	pvsp_vim_crop_width[3]	pvsp_vim_crop_width[2]	pvsp_vim_crop_width[1]	pvsp_vim_crop_width[0]
0xE838	0x00	vsp_3d vim crop height_1	rw	-	-	-	-	-	pvsp_vim_crop_height[10]	pvsp_vim_crop_height[9]	pvsp_vim_crop_height[8]
0xE839	0x00	vsp_3d vim crop height_2	rw	pvsp_vim_crop_height[7]	pvsp_vim_crop_height[6]	pvsp_vim_crop_height[5]	pvsp_vim_crop_height[4]	pvsp_vim_crop_height[3]	pvsp_vim_crop_height[2]	pvsp_vim_crop_height[1]	pvsp_vim_crop_height[0]
0xE83A	0x00	vsp_3d vim scaler output width_1	rw	-	-	-	-	-	pvsp_vim_d_scal_out_width[10]	pvsp_vim_d_scal_out_width[9]	pvsp_vim_d_scal_out_width[8]
0xE83B	0x00	vsp_3d vim scaler output width_2	rw	pvsp_vim_d_scal_out_width[7]	pvsp_vim_d_scal_out_width[6]	pvsp_vim_d_scal_out_width[5]	pvsp_vim_d_scal_out_width[4]	pvsp_vim_d_scal_out_width[3]	pvsp_vim_d_scal_out_width[2]	pvsp_vim_d_scal_out_width[1]	pvsp_vim_d_scal_out_width[0]
0xE83C	0x00	vsp_3d vom crop h start_1	rw	-	-	-	-	-	pvsp_di_crop_h_start[10]	pvsp_di_crop_h_start[9]	pvsp_di_crop_h_start[8]
0xE83D	0x00	vsp_3d vom crop h start_2	rw	pvsp_di_crop_h_start[7]	pvsp_di_crop_h_start[6]	pvsp_di_crop_h_start[5]	pvsp_di_crop_h_start[4]	pvsp_di_crop_h_start[3]	pvsp_di_crop_h_start[2]	pvsp_di_crop_h_start[1]	pvsp_di_crop_h_start[0]
0xE83E	0x00	vsp_3d vom crop v start_1	rw	-	-	-	-	-	pvsp_di_crop_v_start[10]	pvsp_di_crop_v_start[9]	pvsp_di_crop_v_start[8]
0xE83F	0x00	vsp_3d vom crop v start_2	rw	pvsp_di_crop_v_start[7]	pvsp_di_crop_v_start[6]	pvsp_di_crop_v_start[5]	pvsp_di_crop_v_start[4]	pvsp_di_crop_v_start[3]	pvsp_di_crop_v_start[2]	pvsp_di_crop_v_start[1]	pvsp_di_crop_v_start[0]
0xE840	0x00	vsp_3d vom crop width_1	rw	-	-	-	-	-	pvsp_di_crop_width[10]	pvsp_di_crop_width[9]	pvsp_di_crop_width[8]
0xE841	0x00	vsp_3d vom crop width_2	rw	pvsp_di_crop_width[7]	pvsp_di_crop_width[6]	pvsp_di_crop_width[5]	pvsp_di_crop_width[4]	pvsp_di_crop_width[3]	pvsp_di_crop_width[2]	pvsp_di_crop_width[1]	pvsp_di_crop_width[0]
0xE842	0x00	vsp_3d vom crop height_1	rw	-	-	-	-	-	pvsp_di_crop_height[10]	pvsp_di_crop_height[9]	pvsp_di_crop_height[8]
0xE843	0x00	vsp_3d vom crop height_2	rw	pvsp_di_crop_height[7]	pvsp_di_crop_height[6]	pvsp_di_crop_height[5]	pvsp_di_crop_height[4]	pvsp_di_crop_height[3]	pvsp_di_crop_height[2]	pvsp_di_crop_height[1]	pvsp_di_crop_height[0]
0xE844	0x00	vsp_3d vom scaler output width_1	rw	-	-	-	pvsp_scal_out_width[12]	pvsp_scal_out_width[11]	pvsp_scal_out_width[10]	pvsp_scal_out_width[9]	pvsp_scal_out_width[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE845	0x00	vsp_3d vom scaler output width_2	rw	pvsp_scal_out_width[7]	pvsp_scal_out_width[6]	pvsp_scal_out_width[5]	pvsp_scal_out_width[4]	pvsp_scal_out_width[3]	pvsp_scal_out_width[2]	pvsp_scal_out_width[1]	pvsp_scal_out_width[0]
0xE846	0x00	vsp_3d vom scaler output height_1	rw	-	-	-	pvsp_scal_out_height[12]	pvsp_scal_out_height[11]	pvsp_scal_out_height[10]	pvsp_scal_out_height[9]	pvsp_scal_out_height[8]
0xE847	0x00	vsp_3d vom scaler output height_2	rw	pvsp_scal_out_height[7]	pvsp_scal_out_height[6]	pvsp_scal_out_height[5]	pvsp_scal_out_height[4]	pvsp_scal_out_height[3]	pvsp_scal_out_height[2]	pvsp_scal_out_height[1]	pvsp_scal_out_height[0]
0xE848	0x00	vsp_3d vom display port h start_1	rw	-	-	-	pvsp_dp_video_h_start[12]	pvsp_dp_video_h_start[11]	pvsp_dp_video_h_start[10]	pvsp_dp_video_h_start[9]	pvsp_dp_video_h_start[8]
0xE849	0x00	vsp_3d vom display port h start_2	rw	pvsp_dp_video_h_start[7]	pvsp_dp_video_h_start[6]	pvsp_dp_video_h_start[5]	pvsp_dp_video_h_start[4]	pvsp_dp_video_h_start[3]	pvsp_dp_video_h_start[2]	pvsp_dp_video_h_start[1]	pvsp_dp_video_h_start[0]
0xE84A	0x00	vsp_3d vom display port v start_1	rw	-	-	-	pvsp_dp_video_v_start[12]	pvsp_dp_video_v_start[11]	pvsp_dp_video_v_start[10]	pvsp_dp_video_v_start[9]	pvsp_dp_video_v_start[8]
0xE84B	0x00	vsp_3d vom display port v start_2	rw	pvsp_dp_video_v_start[7]	pvsp_dp_video_v_start[6]	pvsp_dp_video_v_start[5]	pvsp_dp_video_v_start[4]	pvsp_dp_video_v_start[3]	pvsp_dp_video_v_start[2]	pvsp_dp_video_v_start[1]	pvsp_dp_video_v_start[0]
0xE84C	0x0E	vsp_3d di control_1	rw	di_sharpness_enable	di_bnr_enable	di_mnr_enable	di_rnr_enable	di_ulai_enable	di_cadence_enable	-	di_intra_field_enable
0xE84D	0x1B	vsp_3d di control_2	rw	-	-	pvsp_bypass_ddr_mode	-	-	pvsp_frc_low_latency_mode	pcadence_enable	di_cue_enable
0xE84E	0x11		rw	-	-	-	pvsp_frc_change_phase_en	pvsp_data_clipping_en	pvsp_di_out_to_svsp_en	-	-
0xE84F	0x0A	vsp_3d denoise level	rw	-	-	-	-	di_mnr_level[1]	di_mnr_level[0]	di_rnr_level[1]	di_rnr_level[0]
0xE850	0x00	vsp_3d vom scaler	rw	-	-	-	-	-	-	-	m_scaler_panorama_en
0xE851	0x00	vsp_3d vom panorama position_1	rw	-	-	-	-	m_scaler_panorama_pos[11]	m_scaler_panorama_pos[10]	m_scaler_panorama_pos[9]	m_scaler_panorama_pos[8]
0xE852	0x00	vsp_3d vom panorama position_2	rw	m_scaler_panorama_pos[7]	m_scaler_panorama_pos[6]	m_scaler_panorama_pos[5]	m_scaler_panorama_pos[4]	m_scaler_panorama_pos[3]	m_scaler_panorama_pos[2]	m_scaler_panorama_pos[1]	m_scaler_panorama_pos[0]
0xE856	0x00	vsp_3d output data enable count_1	rw	-	-	-	pvsp_dp_decount[12]	pvsp_dp_decount[11]	pvsp_dp_decount[10]	pvsp_dp_decount[9]	pvsp_dp_decount[8]
0xE857	0x00	vsp_3d output data enable count_2	rw	pvsp_dp_decount[7]	pvsp_dp_decount[6]	pvsp_dp_decount[5]	pvsp_dp_decount[4]	pvsp_dp_decount[3]	pvsp_dp_decount[2]	pvsp_dp_decount[1]	pvsp_dp_decount[0]
0xE858	0x00	vsp_3d output horizontal front porch_1	rw	-	-	-	-	pvsp_dp_hfrontporch[11]	pvsp_dp_hfrontporch[10]	pvsp_dp_hfrontporch[9]	pvsp_dp_hfrontporch[8]
0xE859	0x00	vsp_3d output horizontal front porch_2	rw	pvsp_dp_hfrontporch[7]	pvsp_dp_hfrontporch[6]	pvsp_dp_hfrontporch[5]	pvsp_dp_hfrontporch[4]	pvsp_dp_hfrontporch[3]	pvsp_dp_hfrontporch[2]	pvsp_dp_hfrontporch[1]	pvsp_dp_hfrontporch[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE85A	0x00	vsp_3d output horizontal sync time_1	rw	-	-	-	-	pvsp_dp_hsyncti me[11]	pvsp_dp_hsyncti me[10]	pvsp_dp_hsyncti me[9]	pvsp_dp_hsyncti me[8]
0xE85B	0x00	vsp_3d output horizontal sync time_2	rw	pvsp_dp_hsyncti me[7]	pvsp_dp_hsyncti me[6]	pvsp_dp_hsyncti me[5]	pvsp_dp_hsyncti me[4]	pvsp_dp_hsyncti me[3]	pvsp_dp_hsyncti me[2]	pvsp_dp_hsyncti me[1]	pvsp_dp_hsyncti me[0]
0xE85C	0x00	vsp_3d output horizontal back porch_1	rw	-	-	-	-	pvsp_dp_hbackpo rch[11]	pvsp_dp_hbackpo rch[10]	pvsp_dp_hbackpo rch[9]	pvsp_dp_hbackpo rch[8]
0xE85D	0x00	vsp_3d output horizontal back porch_2	rw	pvsp_dp_hbackpo rch[7]	pvsp_dp_hbackpo rch[6]	pvsp_dp_hbackpo rch[5]	pvsp_dp_hbackpo rch[4]	pvsp_dp_hbackpo rch[3]	pvsp_dp_hbackpo rch[2]	pvsp_dp_hbackpo rch[1]	pvsp_dp_hbackpo rch[0]
0xE85E	0x00	vsp_3d output active line_1	rw	-	-	-	-	pvsp_dp_activelin e[11]	pvsp_dp_activelin e[10]	pvsp_dp_activelin e[9]	pvsp_dp_activelin e[8]
0xE85F	0x00	vsp_3d output active line_2	rw	pvsp_dp_activelin e[7]	pvsp_dp_activelin e[6]	pvsp_dp_activelin e[5]	pvsp_dp_activelin e[4]	pvsp_dp_activelin e[3]	pvsp_dp_activelin e[2]	pvsp_dp_activelin e[1]	pvsp_dp_activelin e[0]
0xE860	0x00	vsp_3d output vertical front porch_1	rw	-	-	-	-	-	-	pvsp_dp_vfrontpo rch[9]	pvsp_dp_vfrontpo rch[8]
0xE861	0x00	vsp_3d output vertical front porch_2	rw	pvsp_dp_vfrontpo rch[7]	pvsp_dp_vfrontpo rch[6]	pvsp_dp_vfrontpo rch[5]	pvsp_dp_vfrontpo rch[4]	pvsp_dp_vfrontpo rch[3]	pvsp_dp_vfrontpo rch[2]	pvsp_dp_vfrontpo rch[1]	pvsp_dp_vfrontpo rch[0]
0xE862	0x00	vsp_3d output vertical sync time_1	rw	-	-	-	-	-	-	pvsp_dp_vsyncti me[9]	pvsp_dp_vsyncti me[8]
0xE863	0x00	vsp_3d output vertical sync time_2	rw	pvsp_dp_vsyncti me[7]	pvsp_dp_vsyncti me[6]	pvsp_dp_vsyncti me[5]	pvsp_dp_vsyncti me[4]	pvsp_dp_vsyncti me[3]	pvsp_dp_vsyncti me[2]	pvsp_dp_vsyncti me[1]	pvsp_dp_vsyncti me[0]
0xE864	0x00	vsp_3d output vertical back porch_1	rw	-	-	-	-	-	-	pvsp_dp_vbackpo rch[9]	pvsp_dp_vbackpo rch[8]
0xE865	0x00	vsp_3d output vertical back porch_2	rw	pvsp_dp_vbackpo rch[7]	pvsp_dp_vbackpo rch[6]	pvsp_dp_vbackpo rch[5]	pvsp_dp_vbackpo rch[4]	pvsp_dp_vbackpo rch[3]	pvsp_dp_vbackpo rch[2]	pvsp_dp_vbackpo rch[1]	pvsp_dp_vbackpo rch[0]
0xE866	0x00	vsp_3d output margin color_1	rw	pvsp_dp_margin_ color[23]	pvsp_dp_margin_ color[22]	pvsp_dp_margin_ color[21]	pvsp_dp_margin_ color[20]	pvsp_dp_margin_ color[19]	pvsp_dp_margin_ color[18]	pvsp_dp_margin_ color[17]	pvsp_dp_margin_ color[16]
0xE867	0x80	vsp_3d output margin color_2	rw	pvsp_dp_margin_ color[15]	pvsp_dp_margin_ color[14]	pvsp_dp_margin_ color[13]	pvsp_dp_margin_ color[12]	pvsp_dp_margin_ color[11]	pvsp_dp_margin_ color[10]	pvsp_dp_margin_ color[9]	pvsp_dp_margin_ color[8]
0xE868	0x80	vsp_3d output margin color_3	rw	pvsp_dp_margin_ color[7]	pvsp_dp_margin_ color[6]	pvsp_dp_margin_ color[5]	pvsp_dp_margin_ color[4]	pvsp_dp_margin_ color[3]	pvsp_dp_margin_ color[2]	pvsp_dp_margin_ color[1]	pvsp_dp_margin_ color[0]
0xE869	0x00	vsp_3d display port control	rw	-	-	-	pvsp_dp_4kx2k_ mode_en	-	pvsp_dp_output_ blank	pvsp_dp_hpolarit y	pvsp_dp_vpolarit y
0xE86C	0x00		rw	pvsp_dp_subid[1]	pvsp_dp_subid[0]	-	-	-	-	-	-
0xE86E	0x3C		rw	pvsp_vin_fr[7]	pvsp_vin_fr[6]	pvsp_vin_fr[5]	pvsp_vin_fr[4]	pvsp_vin_fr[3]	pvsp_vin_fr[2]	pvsp_vin_fr[1]	pvsp_vin_fr[0]
0xE86F	0x3C		rw	pvsp_vout_fr[7]	pvsp_vout_fr[6]	pvsp_vout_fr[5]	pvsp_vout_fr[4]	pvsp_vout_fr[3]	pvsp_vout_fr[2]	pvsp_vout_fr[1]	pvsp_vout_fr[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE870	0x00	vsp_3d read back information	r	-	pvsp_rb_frame_latency[2]	pvsp_rb_frame_latency[1]	pvsp_rb_frame_latency[0]	-	-	-	-
0xE875	0x00	vsp_3d real time hsync latency_1	r	pvsp_rb_hsync_latency[11]	pvsp_rb_hsync_latency[10]	pvsp_rb_hsync_latency[9]	pvsp_rb_hsync_latency[8]	pvsp_rb_hsync_latency[7]	pvsp_rb_hsync_latency[6]	pvsp_rb_hsync_latency[5]	pvsp_rb_hsync_latency[4]
0xE876	0x00	vsp_3d real time hsync latency_2	r	pvsp_rb_hsync_latency[3]	pvsp_rb_hsync_latency[2]	pvsp_rb_hsync_latency[1]	pvsp_rb_hsync_latency[0]	-	-	-	-
0xE87E	0x20	vsp_3d demo window control_1	rw	pvsp_demo_window_enable	pvsp_demo_window_use_lower_screen	-	pvsp_demo_window_rnr_enable	pvsp_demo_window_mnr_enable	pvsp_demo_window_bnr_enable	pvsp_demo_window_cadence_enable	pvsp_demo_window_ulai_enable
0xE87F	0x00	vsp_3d demo window control_2	rw	-	-	pvsp_demo_window_cue_enable	pvsp_demo_window_intra_field_enable	-	-	-	-
0xE881	0x06	vsp_3d input video vid	rw	pvsp_autocfg_input_vid[7]	pvsp_autocfg_input_vid[6]	pvsp_autocfg_input_vid[5]	pvsp_autocfg_input_vid[4]	pvsp_autocfg_input_vid[3]	pvsp_autocfg_input_vid[2]	pvsp_autocfg_input_vid[1]	pvsp_autocfg_input_vid[0]
0xE882	0x10	vsp_3d output video vid	rw	pvsp_autocfg_output_vid[7]	pvsp_autocfg_output_vid[6]	pvsp_autocfg_output_vid[5]	pvsp_autocfg_output_vid[4]	pvsp_autocfg_output_vid[3]	pvsp_autocfg_output_vid[2]	pvsp_autocfg_output_vid[1]	pvsp_autocfg_output_vid[0]
0xE883	0x80	vsp_3d control register	rw	-	pvsp_vim_crop_enable	pvsp_vim_d_scal_enable	pvsp_di_crop_enable	pvsp_man_scal_out_enable	-	-	pvsp_man_dp_timing_enable
0xE884	0x00		rw	-	-	pvsp_man_input_res	di_lowpower_en	-	-	-	-
0xE889	0x03	vsp_3d field buffer6 address_1	rw	pvsp_fieldbuffer6_addr[31]	pvsp_fieldbuffer6_addr[30]	pvsp_fieldbuffer6_addr[29]	pvsp_fieldbuffer6_addr[28]	pvsp_fieldbuffer6_addr[27]	pvsp_fieldbuffer6_addr[26]	pvsp_fieldbuffer6_addr[25]	pvsp_fieldbuffer6_addr[24]
0xE88A	0xA1	vsp_3d field buffer6 address_2	rw	pvsp_fieldbuffer6_addr[23]	pvsp_fieldbuffer6_addr[22]	pvsp_fieldbuffer6_addr[21]	pvsp_fieldbuffer6_addr[20]	pvsp_fieldbuffer6_addr[19]	pvsp_fieldbuffer6_addr[18]	pvsp_fieldbuffer6_addr[17]	pvsp_fieldbuffer6_addr[16]
0xE88B	0xE2	vsp_3d field buffer6 address_3	rw	pvsp_fieldbuffer6_addr[15]	pvsp_fieldbuffer6_addr[14]	pvsp_fieldbuffer6_addr[13]	pvsp_fieldbuffer6_addr[12]	pvsp_fieldbuffer6_addr[11]	pvsp_fieldbuffer6_addr[10]	pvsp_fieldbuffer6_addr[9]	pvsp_fieldbuffer6_addr[8]
0xE88C	0x00	vsp_3d field buffer6 address_4	rw	pvsp_fieldbuffer6_addr[7]	pvsp_fieldbuffer6_addr[6]	pvsp_fieldbuffer6_addr[5]	pvsp_fieldbuffer6_addr[4]	pvsp_fieldbuffer6_addr[3]	pvsp_fieldbuffer6_addr[2]	pvsp_fieldbuffer6_addr[1]	pvsp_fieldbuffer6_addr[0]
0xE890	0x00		rw	-	-	-	pvsp_srscal_demo_mode_en	pvsp_srscal_8bit_en	-	-	pvsp_srscal_downscaling_blur
0xE891	0x00		rw	pvsp_srscal_scale_gain[11]	pvsp_srscal_scale_gain[10]	pvsp_srscal_scale_gain[9]	pvsp_srscal_scale_gain[8]	pvsp_srscal_scale_gain[7]	pvsp_srscal_scale_gain[6]	pvsp_srscal_scale_gain[5]	pvsp_srscal_scale_gain[4]
0xE892	0x00		rw	pvsp_srscal_scale_gain[3]	pvsp_srscal_scale_gain[2]	pvsp_srscal_scale_gain[1]	pvsp_srscal_scale_gain[0]	-	-	-	-
0xE894	0x00		rw	pvsp_srscal_interp_mode[1]	pvsp_srscal_interp_mode[0]	-	-	-	-	-	-
0xE8A1	0x20	mas_resync	rw	pvsp_mas_resync_en	-	-	-	-	-	-	-
0xE8E5	0xE0	vsp_3d vim scaler control	rw	pvsp_vim_scal_type[1]	pvsp_vim_scal_type[0]	pvsp_vim_scal_anti_aliasing_en	pvsp_vim_scal_separated_alpha_en	-	-	-	-
0xE8E9	0x08	vsp_3d vim scaler over shoot control_1	rw	pvsp_vim_scal_overshoot_ctrl[11]	pvsp_vim_scal_overshoot_ctrl[10]	pvsp_vim_scal_overshoot_ctrl[9]	pvsp_vim_scal_overshoot_ctrl[8]	pvsp_vim_scal_overshoot_ctrl[7]	pvsp_vim_scal_overshoot_ctrl[6]	pvsp_vim_scal_overshoot_ctrl[5]	pvsp_vim_scal_overshoot_ctrl[4]
0xE8EA	0x00	vsp_3d vim scaler over shoot control_2	rw	pvsp_vim_scal_overshoot_ctrl[3]	pvsp_vim_scal_overshoot_ctrl[2]	pvsp_vim_scal_overshoot_ctrl[1]	pvsp_vim_scal_overshoot_ctrl[0]	-	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE8 F0	0x0F	vsp_3d frame latency measuring enable	rw	-	pvsp_frc_latency_measure_en	-	-	-	-	-	-
0xE8 F2	0x00	vsp_3d max frame latency_1	r	pvsp_rb_max_latency[14]	pvsp_rb_max_latency[13]	pvsp_rb_max_latency[12]	pvsp_rb_max_latency[11]	pvsp_rb_max_latency[10]	pvsp_rb_max_latency[9]	pvsp_rb_max_latency[8]	pvsp_rb_max_latency[7]
0xE8 F3	0x00	vsp_3d max frame latency_2	r	pvsp_rb_max_latency[6]	pvsp_rb_max_latency[5]	pvsp_rb_max_latency[4]	pvsp_rb_max_latency[3]	pvsp_rb_max_latency[2]	pvsp_rb_max_latency[1]	pvsp_rb_max_latency[0]	-
0xE8 F4	0x00	vsp_3d min frame latency_1	r	pvsp_rb_min_latency[14]	pvsp_rb_min_latency[13]	pvsp_rb_min_latency[12]	pvsp_rb_min_latency[11]	pvsp_rb_min_latency[10]	pvsp_rb_min_latency[9]	pvsp_rb_min_latency[8]	pvsp_rb_min_latency[7]
0xE8 F5	0x00	vsp_3d min frame latency_2	r	pvsp_rb_min_latency[6]	pvsp_rb_min_latency[5]	pvsp_rb_min_latency[4]	pvsp_rb_min_latency[3]	pvsp_rb_min_latency[2]	pvsp_rb_min_latency[1]	pvsp_rb_min_latency[0]	-
0xE8 FA	0x00	vsp_3d disabled cadence_1	rw	di_fd_disabled_cadence[10]	di_fd_disabled_cadence[9]	di_fd_disabled_cadence[8]	di_fd_disabled_cadence[7]	di_fd_disabled_cadence[6]	di_fd_disabled_cadence[5]	di_fd_disabled_cadence[4]	di_fd_disabled_cadence[3]
0xE8 FB	0x00	vsp_3d disabled cadence_2	rw	di_fd_disabled_cadence[2]	di_fd_disabled_cadence[1]	di_fd_disabled_cadence[0]	-	-	-	-	-

1.3 PRIMARY VSP 2 MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE917	0x2A		rw	di_mnr_th_min[3]	di_mnr_th_min[2]	di_mnr_th_min[1]	di_mnr_th_min[0]	-	-	-	-
0xE935	0xC6		rw	di_fd_scaler_predict[3]	di_fd_scaler_predict[2]	di_fd_scaler_predict[1]	di_fd_scaler_predict[0]	-	-	-	-
0xE93B	0x40		rw	di_fd_sc_th[7]	di_fd_sc_th[6]	di_fd_sc_th[5]	di_fd_sc_th[4]	di_fd_sc_th[3]	di_fd_sc_th[2]	di_fd_sc_th[1]	di_fd_sc_th[0]
0xE949	0xF0		rw	di_fd_subc_th[7]	di_fd_subc_th[6]	di_fd_subc_th[5]	di_fd_subc_th[4]	di_fd_subc_th[3]	di_fd_subc_th[2]	di_fd_subc_th[1]	di_fd_subc_th[0]
0xE97C	0x20		rw	pvsp_srscal_filter_param2[4]	pvsp_srscal_filter_param2[3]	pvsp_srscal_filter_param2[2]	pvsp_srscal_filter_param2[1]	pvsp_srscal_filter_param2[0]	-	-	-
0xE987	0x78		rw	di_bnr_detect_scale_line[3]	di_bnr_detect_scale_line[2]	di_bnr_detect_scale_line[1]	di_bnr_detect_scale_line[0]	di_bnr_disable_local_detect	-	-	-
0xE988	0x80		rw	di_bnr_global_strength_gain[3]	di_bnr_global_strength_gain[2]	di_bnr_global_strength_gain[1]	di_bnr_global_strength_gain[0]	-	-	-	-
0xE98B	0xB4		rw	di_bnr_scale_global_vert[2]	di_bnr_scale_global_vert[1]	di_bnr_scale_global_vert[0]	di_bnr_scale_global_hori[2]	di_bnr_scale_global_hori[1]	di_bnr_scale_global_hori[0]	-	-
0xE98D	0x40		rw	di_bnr_edge_offset[7]	di_bnr_edge_offset[6]	di_bnr_edge_offset[5]	di_bnr_edge_offset[4]	di_bnr_edge_offset[3]	di_bnr_edge_offset[2]	di_bnr_edge_offset[1]	di_bnr_edge_offset[0]

1.4 SECONDARY VSP MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE600	0x04	vsp_2d frame buffer0 address_1	rw	svsp_fieldbuffer0_addr[31]	svsp_fieldbuffer0_addr[30]	svsp_fieldbuffer0_addr[29]	svsp_fieldbuffer0_addr[28]	svsp_fieldbuffer0_addr[27]	svsp_fieldbuffer0_addr[26]	svsp_fieldbuffer0_addr[25]	svsp_fieldbuffer0_addr[24]
0xE601	0x2E	vsp_2d frame buffer0 address_2	rw	svsp_fieldbuffer0_addr[23]	svsp_fieldbuffer0_addr[22]	svsp_fieldbuffer0_addr[21]	svsp_fieldbuffer0_addr[20]	svsp_fieldbuffer0_addr[19]	svsp_fieldbuffer0_addr[18]	svsp_fieldbuffer0_addr[17]	svsp_fieldbuffer0_addr[16]
0xE602	0x82	vsp_2d frame buffer0 address_3	rw	svsp_fieldbuffer0_addr[15]	svsp_fieldbuffer0_addr[14]	svsp_fieldbuffer0_addr[13]	svsp_fieldbuffer0_addr[12]	svsp_fieldbuffer0_addr[11]	svsp_fieldbuffer0_addr[10]	svsp_fieldbuffer0_addr[9]	svsp_fieldbuffer0_addr[8]
0xE603	0x00	vsp_2d frame buffer0 address_4	rw	svsp_fieldbuffer0_addr[7]	svsp_fieldbuffer0_addr[6]	svsp_fieldbuffer0_addr[5]	svsp_fieldbuffer0_addr[4]	svsp_fieldbuffer0_addr[3]	svsp_fieldbuffer0_addr[2]	svsp_fieldbuffer0_addr[1]	svsp_fieldbuffer0_addr[0]
0xE604	0x04	vsp_2d frame buffer1 address_1	rw	svsp_fieldbuffer1_addr[31]	svsp_fieldbuffer1_addr[30]	svsp_fieldbuffer1_addr[29]	svsp_fieldbuffer1_addr[28]	svsp_fieldbuffer1_addr[27]	svsp_fieldbuffer1_addr[26]	svsp_fieldbuffer1_addr[25]	svsp_fieldbuffer1_addr[24]
0xE605	0x8D	vsp_2d frame buffer1 address_2	rw	svsp_fieldbuffer1_addr[23]	svsp_fieldbuffer1_addr[22]	svsp_fieldbuffer1_addr[21]	svsp_fieldbuffer1_addr[20]	svsp_fieldbuffer1_addr[19]	svsp_fieldbuffer1_addr[18]	svsp_fieldbuffer1_addr[17]	svsp_fieldbuffer1_addr[16]
0xE606	0x6E	vsp_2d frame buffer1 address_3	rw	svsp_fieldbuffer1_addr[15]	svsp_fieldbuffer1_addr[14]	svsp_fieldbuffer1_addr[13]	svsp_fieldbuffer1_addr[12]	svsp_fieldbuffer1_addr[11]	svsp_fieldbuffer1_addr[10]	svsp_fieldbuffer1_addr[9]	svsp_fieldbuffer1_addr[8]
0xE607	0x00	vsp_2d frame buffer1 address_4	rw	svsp_fieldbuffer1_addr[7]	svsp_fieldbuffer1_addr[6]	svsp_fieldbuffer1_addr[5]	svsp_fieldbuffer1_addr[4]	svsp_fieldbuffer1_addr[3]	svsp_fieldbuffer1_addr[2]	svsp_fieldbuffer1_addr[1]	svsp_fieldbuffer1_addr[0]
0xE608	0x04	vsp_2d frame buffer2 address_1	rw	svsp_fieldbuffer2_addr[31]	svsp_fieldbuffer2_addr[30]	svsp_fieldbuffer2_addr[29]	svsp_fieldbuffer2_addr[28]	svsp_fieldbuffer2_addr[27]	svsp_fieldbuffer2_addr[26]	svsp_fieldbuffer2_addr[25]	svsp_fieldbuffer2_addr[24]
0xE609	0xEC	vsp_2d frame buffer2 address_2	rw	svsp_fieldbuffer2_addr[23]	svsp_fieldbuffer2_addr[22]	svsp_fieldbuffer2_addr[21]	svsp_fieldbuffer2_addr[20]	svsp_fieldbuffer2_addr[19]	svsp_fieldbuffer2_addr[18]	svsp_fieldbuffer2_addr[17]	svsp_fieldbuffer2_addr[16]
0xE60A	0x5A	vsp_2d frame buffer2 address_3	rw	svsp_fieldbuffer2_addr[15]	svsp_fieldbuffer2_addr[14]	svsp_fieldbuffer2_addr[13]	svsp_fieldbuffer2_addr[12]	svsp_fieldbuffer2_addr[11]	svsp_fieldbuffer2_addr[10]	svsp_fieldbuffer2_addr[9]	svsp_fieldbuffer2_addr[8]
0xE60B	0x00	vsp_2d frame buffer2 address_4	rw	svsp_fieldbuffer2_addr[7]	svsp_fieldbuffer2_addr[6]	svsp_fieldbuffer2_addr[5]	svsp_fieldbuffer2_addr[4]	svsp_fieldbuffer2_addr[3]	svsp_fieldbuffer2_addr[2]	svsp_fieldbuffer2_addr[1]	svsp_fieldbuffer2_addr[0]
0xE60C	0x05	vsp_2d frame buffer3 address_1	rw	svsp_fieldbuffer3_addr[31]	svsp_fieldbuffer3_addr[30]	svsp_fieldbuffer3_addr[29]	svsp_fieldbuffer3_addr[28]	svsp_fieldbuffer3_addr[27]	svsp_fieldbuffer3_addr[26]	svsp_fieldbuffer3_addr[25]	svsp_fieldbuffer3_addr[24]
0xE60D	0x4B	vsp_2d frame buffer3 address_2	rw	svsp_fieldbuffer3_addr[23]	svsp_fieldbuffer3_addr[22]	svsp_fieldbuffer3_addr[21]	svsp_fieldbuffer3_addr[20]	svsp_fieldbuffer3_addr[19]	svsp_fieldbuffer3_addr[18]	svsp_fieldbuffer3_addr[17]	svsp_fieldbuffer3_addr[16]
0xE60E	0x46	vsp_2d frame buffer3 address_3	rw	svsp_fieldbuffer3_addr[15]	svsp_fieldbuffer3_addr[14]	svsp_fieldbuffer3_addr[13]	svsp_fieldbuffer3_addr[12]	svsp_fieldbuffer3_addr[11]	svsp_fieldbuffer3_addr[10]	svsp_fieldbuffer3_addr[9]	svsp_fieldbuffer3_addr[8]
0xE60F	0x00	vsp_2d frame buffer3 address_4	rw	svsp_fieldbuffer3_addr[7]	svsp_fieldbuffer3_addr[6]	svsp_fieldbuffer3_addr[5]	svsp_fieldbuffer3_addr[4]	svsp_fieldbuffer3_addr[3]	svsp_fieldbuffer3_addr[2]	svsp_fieldbuffer3_addr[1]	svsp_fieldbuffer3_addr[0]
0xE610	0x00	vsp_2d control_1	rw	svsp_enable_ffs	svsp_enable_vim	svsp_enable_vom	svsp_lock_vom	svsp_update_vom	svsp_fieldbuf_num[2]	svsp_fieldbuf_num[1]	svsp_fieldbuf_num[0]
0xE611	0x08	vsp_2d control_2	rw	svsp_ex_mem_data_format[1]	svsp_ex_mem_data_format[0]	-	-	-	svsp_osd_mode_en	svsp_osd_exalpha_en	svsp_input_from_pvsp_di_out
0xE616	0x00	vsp_2d input horizontal resolution_1	rw	svsp_vin_h[12]	svsp_vin_h[11]	svsp_vin_h[10]	svsp_vin_h[9]	svsp_vin_h[8]	svsp_vin_h[7]	svsp_vin_h[6]	svsp_vin_h[5]
0xE617	0x00	vsp_2d input horizontal resolution_2	rw	svsp_vin_h[4]	svsp_vin_h[3]	svsp_vin_h[2]	svsp_vin_h[1]	svsp_vin_h[0]	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE618	0x00	vsp_2d input vertical resolution_1	rw	svsp_vin_v[12]	svsp_vin_v[11]	svsp_vin_v[10]	svsp_vin_v[9]	svsp_vin_v[8]	svsp_vin_v[7]	svsp_vin_v[6]	svsp_vin_v[5]
0xE619	0x00	vsp_2d input vertical resolution_2	rw	svsp_vin_v[4]	svsp_vin_v[3]	svsp_vin_v[2]	svsp_vin_v[1]	svsp_vin_v[0]	-	-	-
0xE61A	0x00	vsp_2d vim crop h start_1	rw	svsp_vim_crop_h_start[12]	svsp_vim_crop_h_start[11]	svsp_vim_crop_h_start[10]	svsp_vim_crop_h_start[9]	svsp_vim_crop_h_start[8]	svsp_vim_crop_h_start[7]	svsp_vim_crop_h_start[6]	svsp_vim_crop_h_start[5]
0xE61B	0x00	vsp_2d vim crop h start_2	rw	svsp_vim_crop_h_start[4]	svsp_vim_crop_h_start[3]	svsp_vim_crop_h_start[2]	svsp_vim_crop_h_start[1]	svsp_vim_crop_h_start[0]	-	-	-
0xE61C	0x00	vsp_2d vim crop v start_1	rw	svsp_vim_crop_v_start[12]	svsp_vim_crop_v_start[11]	svsp_vim_crop_v_start[10]	svsp_vim_crop_v_start[9]	svsp_vim_crop_v_start[8]	svsp_vim_crop_v_start[7]	svsp_vim_crop_v_start[6]	svsp_vim_crop_v_start[5]
0xE61D	0x00	vsp_2d vim crop v start_2	rw	svsp_vim_crop_v_start[4]	svsp_vim_crop_v_start[3]	svsp_vim_crop_v_start[2]	svsp_vim_crop_v_start[1]	svsp_vim_crop_v_start[0]	-	-	-
0xE61E	0x00	vsp_2d vim crop width_1	rw	svsp_vim_crop_width[12]	svsp_vim_crop_width[11]	svsp_vim_crop_width[10]	svsp_vim_crop_width[9]	svsp_vim_crop_width[8]	svsp_vim_crop_width[7]	svsp_vim_crop_width[6]	svsp_vim_crop_width[5]
0xE61F	0x00	vsp_2d vim crop width_2	rw	svsp_vim_crop_width[4]	svsp_vim_crop_width[3]	svsp_vim_crop_width[2]	svsp_vim_crop_width[1]	svsp_vim_crop_width[0]	-	-	-
0xE620	0x00	vsp_2d vim crop height_1	rw	svsp_vim_crop_height[12]	svsp_vim_crop_height[11]	svsp_vim_crop_height[10]	svsp_vim_crop_height[9]	svsp_vim_crop_height[8]	svsp_vim_crop_height[7]	svsp_vim_crop_height[6]	svsp_vim_crop_height[5]
0xE621	0x00	vsp_2d vim crop height_2	rw	svsp_vim_crop_height[4]	svsp_vim_crop_height[3]	svsp_vim_crop_height[2]	svsp_vim_crop_height[1]	svsp_vim_crop_height[0]	-	-	-
0xE622	0x00	vsp_2d vim scaler output width_1	rw	svsp_vim_scal_out_width[10]	svsp_vim_scal_out_width[9]	svsp_vim_scal_out_width[8]	svsp_vim_scal_out_width[7]	svsp_vim_scal_out_width[6]	svsp_vim_scal_out_width[5]	svsp_vim_scal_out_width[4]	svsp_vim_scal_out_width[3]
0xE623	0x00	vsp_2d vim scaler output width_2	rw	svsp_vim_scal_out_width[2]	svsp_vim_scal_out_width[1]	svsp_vim_scal_out_width[0]	-	-	-	-	-
0xE624	0x00	vsp_2d vim scaler output height_1	rw	svsp_vim_scal_out_height[10]	svsp_vim_scal_out_height[9]	svsp_vim_scal_out_height[8]	svsp_vim_scal_out_height[7]	svsp_vim_scal_out_height[6]	svsp_vim_scal_out_height[5]	svsp_vim_scal_out_height[4]	svsp_vim_scal_out_height[3]
0xE625	0x00	vsp_2d vim scaler output height_2	rw	svsp_vim_scal_out_height[2]	svsp_vim_scal_out_height[1]	svsp_vim_scal_out_height[0]	-	-	-	-	-
0xE626	0x00	vsp_2d vom crop h start_1	rw	svsp_vom_crop_h_start[10]	svsp_vom_crop_h_start[9]	svsp_vom_crop_h_start[8]	svsp_vom_crop_h_start[7]	svsp_vom_crop_h_start[6]	svsp_vom_crop_h_start[5]	svsp_vom_crop_h_start[4]	svsp_vom_crop_h_start[3]
0xE627	0x00	vsp_2d vom crop h start_2	rw	svsp_vom_crop_h_start[2]	svsp_vom_crop_h_start[1]	svsp_vom_crop_h_start[0]	-	-	-	-	-
0xE628	0x00	vsp_2d vom crop v start_1	rw	svsp_vom_crop_v_start[10]	svsp_vom_crop_v_start[9]	svsp_vom_crop_v_start[8]	svsp_vom_crop_v_start[7]	svsp_vom_crop_v_start[6]	svsp_vom_crop_v_start[5]	svsp_vom_crop_v_start[4]	svsp_vom_crop_v_start[3]
0xE629	0x00	vsp_2d vom crop v start_2	rw	svsp_vom_crop_v_start[2]	svsp_vom_crop_v_start[1]	svsp_vom_crop_v_start[0]	-	-	-	-	-
0xE62A	0x00	vsp_2d vom crop width_1	rw	svsp_vom_crop_width[10]	svsp_vom_crop_width[9]	svsp_vom_crop_width[8]	svsp_vom_crop_width[7]	svsp_vom_crop_width[6]	svsp_vom_crop_width[5]	svsp_vom_crop_width[4]	svsp_vom_crop_width[3]
0xE62B	0x00	vsp_2d vom crop width_2	rw	svsp_vom_crop_width[2]	svsp_vom_crop_width[1]	svsp_vom_crop_width[0]	-	-	-	-	-
0xE62C	0x00	vsp_2d vom crop height_1	rw	svsp_vom_crop_height[10]	svsp_vom_crop_height[9]	svsp_vom_crop_height[8]	svsp_vom_crop_height[7]	svsp_vom_crop_height[6]	svsp_vom_crop_height[5]	svsp_vom_crop_height[4]	svsp_vom_crop_height[3]
0xE62D	0x00	vsp_2d vom crop height_2	rw	svsp_vom_crop_height[2]	svsp_vom_crop_height[1]	svsp_vom_crop_height[0]	-	-	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE62E	0x00	vsp_2d vom display port h start_1	rw	svsp_dp_video_h_start[10]	svsp_dp_video_h_start[9]	svsp_dp_video_h_start[8]	svsp_dp_video_h_start[7]	svsp_dp_video_h_start[6]	svsp_dp_video_h_start[5]	svsp_dp_video_h_start[4]	svsp_dp_video_h_start[3]
0xE62F	0x00	vsp_2d vom display port h start_2	rw	svsp_dp_video_h_start[2]	svsp_dp_video_h_start[1]	svsp_dp_video_h_start[0]	-	-	-	-	-
0xE630	0x00	vsp_2d vom display port v start_1	rw	svsp_dp_video_v_start[10]	svsp_dp_video_v_start[9]	svsp_dp_video_v_start[8]	svsp_dp_video_v_start[7]	svsp_dp_video_v_start[6]	svsp_dp_video_v_start[5]	svsp_dp_video_v_start[4]	svsp_dp_video_v_start[3]
0xE631	0x00	vsp_2d vom display port v start_2	rw	svsp_dp_video_v_start[2]	svsp_dp_video_v_start[1]	svsp_dp_video_v_start[0]	-	-	-	-	-
0xE632	0x00	vsp_2d output data enable count_1	rw	svsp_dp_decount[10]	svsp_dp_decount[9]	svsp_dp_decount[8]	svsp_dp_decount[7]	svsp_dp_decount[6]	svsp_dp_decount[5]	svsp_dp_decount[4]	svsp_dp_decount[3]
0xE633	0x00	vsp_2d output data enable count_2	rw	svsp_dp_decount[2]	svsp_dp_decount[1]	svsp_dp_decount[0]	-	-	-	-	-
0xE634	0x00	vsp_2d output horizontal front porch_1	rw	svsp_dp_hfrontporch[11]	svsp_dp_hfrontporch[10]	svsp_dp_hfrontporch[9]	svsp_dp_hfrontporch[8]	svsp_dp_hfrontporch[7]	svsp_dp_hfrontporch[6]	svsp_dp_hfrontporch[5]	svsp_dp_hfrontporch[4]
0xE635	0x00	vsp_2d output horizontal front porch_2	rw	svsp_dp_hfrontporch[3]	svsp_dp_hfrontporch[2]	svsp_dp_hfrontporch[1]	svsp_dp_hfrontporch[0]	-	-	-	-
0xE636	0x00	vsp_2d output horizontal sync time_1	rw	svsp_dp_hsynctime[9]	svsp_dp_hsynctime[8]	svsp_dp_hsynctime[7]	svsp_dp_hsynctime[6]	svsp_dp_hsynctime[5]	svsp_dp_hsynctime[4]	svsp_dp_hsynctime[3]	svsp_dp_hsynctime[2]
0xE637	0x00	vsp_2d output horizontal sync time_2	rw	svsp_dp_hsynctime[1]	svsp_dp_hsynctime[0]	-	-	-	-	-	-
0xE638	0x00	vsp_2d output horizontal back porch_1	rw	svsp_dp_hbackporch[9]	svsp_dp_hbackporch[8]	svsp_dp_hbackporch[7]	svsp_dp_hbackporch[6]	svsp_dp_hbackporch[5]	svsp_dp_hbackporch[4]	svsp_dp_hbackporch[3]	svsp_dp_hbackporch[2]
0xE639	0x00	vsp_2d output horizontal back porch_2	rw	svsp_dp_hbackporch[1]	svsp_dp_hbackporch[0]	-	-	-	-	-	-
0xE63A	0x00	vsp_2d output active line_1	rw	svsp_dp_activeline[10]	svsp_dp_activeline[9]	svsp_dp_activeline[8]	svsp_dp_activeline[7]	svsp_dp_activeline[6]	svsp_dp_activeline[5]	svsp_dp_activeline[4]	svsp_dp_activeline[3]
0xE63B	0x00	vsp_2d output active line_2	rw	svsp_dp_activeline[2]	svsp_dp_activeline[1]	svsp_dp_activeline[0]	-	-	-	-	-
0xE63C	0x00	vsp_2d output vertical front porch_1	rw	svsp_dp_vfrontporch[9]	svsp_dp_vfrontporch[8]	svsp_dp_vfrontporch[7]	svsp_dp_vfrontporch[6]	svsp_dp_vfrontporch[5]	svsp_dp_vfrontporch[4]	svsp_dp_vfrontporch[3]	svsp_dp_vfrontporch[2]
0xE63D	0x00	vsp_2d output vertical front porch_2	rw	svsp_dp_vfrontporch[1]	svsp_dp_vfrontporch[0]	-	-	-	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE63E	0x00	vsp_2d output vertical sync time_1	rw	svsp_dp_vsynctime[9]	svsp_dp_vsynctime[8]	svsp_dp_vsynctime[7]	svsp_dp_vsynctime[6]	svsp_dp_vsynctime[5]	svsp_dp_vsynctime[4]	svsp_dp_vsynctime[3]	svsp_dp_vsynctime[2]
0xE63F	0x00	vsp_2d output vertical sync time_2	rw	svsp_dp_vsynctime[1]	svsp_dp_vsynctime[0]	-	-	-	-	-	-
0xE640	0x00	vsp_2d output vertical back porch_1	rw	svsp_dp_vbackporch[9]	svsp_dp_vbackporch[8]	svsp_dp_vbackporch[7]	svsp_dp_vbackporch[6]	svsp_dp_vbackporch[5]	svsp_dp_vbackporch[4]	svsp_dp_vbackporch[3]	svsp_dp_vbackporch[2]
0xE641	0x00	vsp_2d output vertical back porch_2	rw	svsp_dp_vbackporch[1]	svsp_dp_vbackporch[0]	-	-	-	-	-	-
0xE642	0x00	vsp_2d display port control	rw	svsp_dp_vpolarity	svsp_dp_hpolarity	svsp_dp_output_blank	svsp_data_clipping_en	-	-	-	-
0xE643	0x00	vsp_2d output margin color_1	rw	svsp_dp_margin_color[23]	svsp_dp_margin_color[22]	svsp_dp_margin_color[21]	svsp_dp_margin_color[20]	svsp_dp_margin_color[19]	svsp_dp_margin_color[18]	svsp_dp_margin_color[17]	svsp_dp_margin_color[16]
0xE644	0x80	vsp_2d output margin color_2	rw	svsp_dp_margin_color[15]	svsp_dp_margin_color[14]	svsp_dp_margin_color[13]	svsp_dp_margin_color[12]	svsp_dp_margin_color[11]	svsp_dp_margin_color[10]	svsp_dp_margin_color[9]	svsp_dp_margin_color[8]
0xE645	0x80	vsp_2d output margin color_3	rw	svsp_dp_margin_color[7]	svsp_dp_margin_color[6]	svsp_dp_margin_color[5]	svsp_dp_margin_color[4]	svsp_dp_margin_color[3]	svsp_dp_margin_color[2]	svsp_dp_margin_color[1]	svsp_dp_margin_color[0]
0xE646	0x00	vsp_2d vim scaler control_1	rw	svsp_vim_scal_type[1]	svsp_vim_scal_type[0]	svsp_vim_scal_sep_alpha_en	-	-	-	-	-
0xE647	0x08	vsp_2d vim scaler over shoot control_1	rw	svsp_vim_scal_overshoot_ctrl[11]	svsp_vim_scal_overshoot_ctrl[10]	svsp_vim_scal_overshoot_ctrl[9]	svsp_vim_scal_overshoot_ctrl[8]	svsp_vim_scal_overshoot_ctrl[7]	svsp_vim_scal_overshoot_ctrl[6]	svsp_vim_scal_overshoot_ctrl[5]	svsp_vim_scal_overshoot_ctrl[4]
0xE648	0x00	vsp_2d vim scaler over shoot control_2	rw	svsp_vim_scal_overshoot_ctrl[3]	svsp_vim_scal_overshoot_ctrl[2]	svsp_vim_scal_overshoot_ctrl[1]	svsp_vim_scal_overshoot_ctrl[0]	-	-	-	-
0xE649	0x00	vsp_2d control_3	rw	svsp_ddr_bypass	svsp_bypass	svsp_p2i_enable	m_p2i_enable	-	-	-	-
0xE64A	0x00	vsp_2d p2i vid	rw	svsp_p2i_vid[7]	svsp_p2i_vid[6]	svsp_p2i_vid[5]	svsp_p2i_vid[4]	svsp_p2i_vid[3]	svsp_p2i_vid[2]	svsp_p2i_vid[1]	svsp_p2i_vid[0]
0xE64B	0x00	vsp_top p2i vid	rw	m_p2i_vid[7]	m_p2i_vid[6]	m_p2i_vid[5]	m_p2i_vid[4]	m_p2i_vid[3]	m_p2i_vid[2]	m_p2i_vid[1]	m_p2i_vid[0]
0xE650	0x60	vsp_2d vim scaler control_2	rw	svsp_vim_scal_pano_en	svsp_vim_scal_anti_alias_v_en	svsp_vim_scal_anti_alias_h_en	-	-	-	-	-
0xE651	0x00	vsp_2d vim scaler control_3	rw	svsp_vim_scal_pano_pos[10]	svsp_vim_scal_pano_pos[9]	svsp_vim_scal_pano_pos[8]	svsp_vim_scal_pano_pos[7]	svsp_vim_scal_pano_pos[6]	svsp_vim_scal_pano_pos[5]	svsp_vim_scal_pano_pos[4]	svsp_vim_scal_pano_pos[3]
0xE652	0x00	vsp_2d vim scaler control_4	rw	svsp_vim_scal_pano_pos[2]	svsp_vim_scal_pano_pos[1]	svsp_vim_scal_pano_pos[0]	-	-	-	-	-
0xE658	0x3C		rw	svsp_vin_fr[7]	svsp_vin_fr[6]	svsp_vin_fr[5]	svsp_vin_fr[4]	svsp_vin_fr[3]	svsp_vin_fr[2]	svsp_vin_fr[1]	svsp_vin_fr[0]
0xE659	0x3C		rw	svsp_vout_fr[7]	svsp_vout_fr[6]	svsp_vout_fr[5]	svsp_vout_fr[4]	svsp_vout_fr[3]	svsp_vout_fr[2]	svsp_vout_fr[1]	svsp_vout_fr[0]
0xE65B	0x00		rw	m_p2i_drop_line_as_pvsp_flag	-	-	-	-	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE65D	0x00		rw	svsp_dp_subid[1]	svsp_dp_subid[0]	-	-	-	-	-	-
0xE65E	0x00		rw	s_p2i_invert_vsp2d_flag	-	-	-	-	-	-	-
0xE65F	0x00		rw	svsp_mas_resync_en	-	-	-	-	-	-	-
0xE660	0x00		rw	svsp_autocfg_input_vid[7]	svsp_autocfg_input_vid[6]	svsp_autocfg_input_vid[5]	svsp_autocfg_input_vid[4]	svsp_autocfg_input_vid[3]	svsp_autocfg_input_vid[2]	svsp_autocfg_input_vid[1]	svsp_autocfg_input_vid[0]
0xE661	0x00		rw	svsp_autocfg_output_vid[7]	svsp_autocfg_output_vid[6]	svsp_autocfg_output_vid[5]	svsp_autocfg_output_vid[4]	svsp_autocfg_output_vid[3]	svsp_autocfg_output_vid[2]	svsp_autocfg_output_vid[1]	svsp_autocfg_output_vid[0]
0xE662	0x80		rw	-	svsp_vim_crop_enable	svsp_man_scal_out_enable	svsp_man_scaler_para_enable	-	svsp_frc_latency_measure_en	svsp_vom_crop_enable	svsp_man_set_ddr_bypass
0xE663	0x00		rw	svsp_man_dp_timing_enable	-	-	svsp_man_input_res	-	-	-	-
0xE664	0x05		rw	svsp_fieldbuffer4_addr[31]	svsp_fieldbuffer4_addr[30]	svsp_fieldbuffer4_addr[29]	svsp_fieldbuffer4_addr[28]	svsp_fieldbuffer4_addr[27]	svsp_fieldbuffer4_addr[26]	svsp_fieldbuffer4_addr[25]	svsp_fieldbuffer4_addr[24]
0xE665	0xAA		rw	svsp_fieldbuffer4_addr[23]	svsp_fieldbuffer4_addr[22]	svsp_fieldbuffer4_addr[21]	svsp_fieldbuffer4_addr[20]	svsp_fieldbuffer4_addr[19]	svsp_fieldbuffer4_addr[18]	svsp_fieldbuffer4_addr[17]	svsp_fieldbuffer4_addr[16]
0xE666	0x32		rw	svsp_fieldbuffer4_addr[15]	svsp_fieldbuffer4_addr[14]	svsp_fieldbuffer4_addr[13]	svsp_fieldbuffer4_addr[12]	svsp_fieldbuffer4_addr[11]	svsp_fieldbuffer4_addr[10]	svsp_fieldbuffer4_addr[9]	svsp_fieldbuffer4_addr[8]
0xE667	0x00		rw	svsp_fieldbuffer4_addr[7]	svsp_fieldbuffer4_addr[6]	svsp_fieldbuffer4_addr[5]	svsp_fieldbuffer4_addr[4]	svsp_fieldbuffer4_addr[3]	svsp_fieldbuffer4_addr[2]	svsp_fieldbuffer4_addr[1]	svsp_fieldbuffer4_addr[0]
0xE668	0x06		rw	svsp_fieldbuffer5_addr[31]	svsp_fieldbuffer5_addr[30]	svsp_fieldbuffer5_addr[29]	svsp_fieldbuffer5_addr[28]	svsp_fieldbuffer5_addr[27]	svsp_fieldbuffer5_addr[26]	svsp_fieldbuffer5_addr[25]	svsp_fieldbuffer5_addr[24]
0xE669	0x09		rw	svsp_fieldbuffer5_addr[23]	svsp_fieldbuffer5_addr[22]	svsp_fieldbuffer5_addr[21]	svsp_fieldbuffer5_addr[20]	svsp_fieldbuffer5_addr[19]	svsp_fieldbuffer5_addr[18]	svsp_fieldbuffer5_addr[17]	svsp_fieldbuffer5_addr[16]
0xE66A	0x1E		rw	svsp_fieldbuffer5_addr[15]	svsp_fieldbuffer5_addr[14]	svsp_fieldbuffer5_addr[13]	svsp_fieldbuffer5_addr[12]	svsp_fieldbuffer5_addr[11]	svsp_fieldbuffer5_addr[10]	svsp_fieldbuffer5_addr[9]	svsp_fieldbuffer5_addr[8]
0xE66B	0x00		rw	svsp_fieldbuffer5_addr[7]	svsp_fieldbuffer5_addr[6]	svsp_fieldbuffer5_addr[5]	svsp_fieldbuffer5_addr[4]	svsp_fieldbuffer5_addr[3]	svsp_fieldbuffer5_addr[2]	svsp_fieldbuffer5_addr[1]	svsp_fieldbuffer5_addr[0]
0xE66C	0x06		rw	svsp_fieldbuffer6_addr[31]	svsp_fieldbuffer6_addr[30]	svsp_fieldbuffer6_addr[29]	svsp_fieldbuffer6_addr[28]	svsp_fieldbuffer6_addr[27]	svsp_fieldbuffer6_addr[26]	svsp_fieldbuffer6_addr[25]	svsp_fieldbuffer6_addr[24]
0xE66D	0x68		rw	svsp_fieldbuffer6_addr[23]	svsp_fieldbuffer6_addr[22]	svsp_fieldbuffer6_addr[21]	svsp_fieldbuffer6_addr[20]	svsp_fieldbuffer6_addr[19]	svsp_fieldbuffer6_addr[18]	svsp_fieldbuffer6_addr[17]	svsp_fieldbuffer6_addr[16]
0xE66E	0x0A		rw	svsp_fieldbuffer6_addr[15]	svsp_fieldbuffer6_addr[14]	svsp_fieldbuffer6_addr[13]	svsp_fieldbuffer6_addr[12]	svsp_fieldbuffer6_addr[11]	svsp_fieldbuffer6_addr[10]	svsp_fieldbuffer6_addr[9]	svsp_fieldbuffer6_addr[8]
0xE66F	0x00		rw	svsp_fieldbuffer6_addr[7]	svsp_fieldbuffer6_addr[6]	svsp_fieldbuffer6_addr[5]	svsp_fieldbuffer6_addr[4]	svsp_fieldbuffer6_addr[3]	svsp_fieldbuffer6_addr[2]	svsp_fieldbuffer6_addr[1]	svsp_fieldbuffer6_addr[0]
0xE6F2	0x00		r	svsp_rb_frame_latency[2]	svsp_rb_frame_latency[1]	svsp_rb_frame_latency[0]	-	-	-	-	-
0xE6F3	0x00		r	svsp_rb_hsync_latency[11]	svsp_rb_hsync_latency[10]	svsp_rb_hsync_latency[9]	svsp_rb_hsync_latency[8]	svsp_rb_hsync_latency[7]	svsp_rb_hsync_latency[6]	svsp_rb_hsync_latency[5]	svsp_rb_hsync_latency[4]
0xE6F4	0x00		r	svsp_rb_hsync_latency[3]	svsp_rb_hsync_latency[2]	svsp_rb_hsync_latency[1]	svsp_rb_hsync_latency[0]	-	-	-	-
0xE6F5	0x00		r	svsp_rb_max_latency[14]	svsp_rb_max_latency[13]	svsp_rb_max_latency[12]	svsp_rb_max_latency[11]	svsp_rb_max_latency[10]	svsp_rb_max_latency[9]	svsp_rb_max_latency[8]	svsp_rb_max_latency[7]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE6 F6	0x00		r	svsp_rb_max_latency[6]	svsp_rb_max_latency[5]	svsp_rb_max_latency[4]	svsp_rb_max_latency[3]	svsp_rb_max_latency[2]	svsp_rb_max_latency[1]	svsp_rb_max_latency[0]	-
0xE6 F7	0x00		r	svsp_rb_min_latency[14]	svsp_rb_min_latency[13]	svsp_rb_min_latency[12]	svsp_rb_min_latency[11]	svsp_rb_min_latency[10]	svsp_rb_min_latency[9]	svsp_rb_min_latency[8]	svsp_rb_min_latency[7]
0xE6 F8	0x00		r	svsp_rb_min_latency[6]	svsp_rb_min_latency[5]	svsp_rb_min_latency[4]	svsp_rb_min_latency[3]	svsp_rb_min_latency[2]	svsp_rb_min_latency[1]	svsp_rb_min_latency[0]	-

1.5 HDMI RX MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE201	0x00	hdmi_register_01h	rw	-	-	-	-	-	-	-	term_auto
0xE204	0x00	hdmi register_04h	r	-	av_mute	-	-	-	-	tmds_pll_locked	-
0xE205	0x00	hdmi_register_05h	r	rx_hdmi_mode	-	dvi_hsync_polarity	dvi_vsync_polarity	hdmi_pixel_repetition[3]	hdmi_pixel_repetition[2]	hdmi_pixel_repetition[1]	hdmi_pixel_repetition[0]
0xE207	0x00	line width_1	r	-	audio_channel_mode	-	-	-	-	-	-
0xE20B	0x00	field1 height_1	r	deep_color_mode[1]	deep_color_mode[0]	-	-	-	-	-	-
0xE217	0x00	packets detected_1	r	-	gc_packet_det	acr_packet_det	gamut_mdata_pckt_det	isrc2_packet_det	isrc1_packet_det	acp_packet_det	vs_infoframe_det
0xE218	0x00	packets detected_2	r	ms_infoframe_det	spd_infoframe_det	audio_infoframe_det	avi_infoframe_det	hbr_audio_pckt_det	-	dsd_packet_det	audio_sample_pckt_det
0xE219	0x00	packets detected_3	r	vs_inf_cksum_err	ms_inf_cksum_err	spd_inf_cksum_err	aud_inf_cksum_err	avi_inf_cksum_err	-	-	-
0xE21B	0x18	deepcolor_fifo_debug_1	rw	-	-	-	dcfifo_reset_on_lock	dcfifo_kill_not_locked	dcfifo_kill_dis	-	-
0xE21C	0x00	deepcolor_fifo_debug_2	r	-	-	-	-	dcfifo_locked	dcfifo_level[2]	dcfifo_level[1]	dcfifo_level[0]
0xE21D	0x00	register_1dh	rw	-	pdn_pkt_processor	up_conversion_mode	-	-	-	-	-
0xE240	0x00	register_40h	rw	-	override_deep_color_mode	deep_color_mode_user[1]	deep_color_mode_user[0]	-	-	-	-
0xE241	0x40	register_41h	rw	-	-	-	derep_n_override	derep_n[3]	derep_n[2]	derep_n[1]	derep_n[0]
0xE247	0x00	register_47h	rw	-	-	-	-	-	qzero_itc_dis	qzero_rgb_full	always_store_inf
0xE248	0x00	register_48h	rw	-	dis_cable_det_rst	-	-	-	-	-	-
0xE250	0x00	hdmi_register_50	rw	-	-	-	gamut_irq_next_field	-	-	-	-
0xE253	0x00	hdmi_colorspace	r	-	-	-	-	hdmi_colorspace[3]	hdmi_colorspace[2]	hdmi_colorspace[1]	hdmi_colorspace[0]
0xE256	0x58	filt_5v_det_reg	rw	filt_5v_det_dis	filt_5v_det_timer[6]	filt_5v_det_timer[5]	filt_5v_det_timer[4]	filt_5v_det_timer[3]	filt_5v_det_timer[2]	filt_5v_det_timer[1]	filt_5v_det_timer[0]
0xE25A	0x00	register_5a	sc	-	-	-	-	-	dcfifo_recenter	-	force_n_update
0xE25B	0x00	cts_n_1	r	cts[19]	cts[18]	cts[17]	cts[16]	cts[15]	cts[14]	cts[13]	cts[12]
0xE25C	0x00	cts_n_2	r	cts[11]	cts[10]	cts[9]	cts[8]	cts[7]	cts[6]	cts[5]	cts[4]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE2 5D	0x00	cts_n_3	r	cts[3]	cts[2]	cts[1]	cts[0]	n[19]	n[18]	n[17]	n[16]
0xE2 5E	0x00	cts_n_4	r	n[15]	n[14]	n[13]	n[12]	n[11]	n[10]	n[9]	n[8]
0xE2 5F	0x00	cts_n_5	r	n[7]	n[6]	n[5]	n[4]	n[3]	n[2]	n[1]	n[0]
0xE2 83	0xFF	hdmi_register_02 h	rw	-	-	-	-	-	-	-	clock_terma_disa ble
0xE2 89	0x00	eq dynamic enable	rw	-	-	-	-	-	-	-	eq_dyn_freq2
0xE2 8A	0xA3	eq dynamic freq	rw	eq_dyn_freq2[3]	eq_dyn_freq2[2]	eq_dyn_freq2[1]	eq_dyn_freq2[0]	-	-	-	-

1.6 HDMI RX INFOFRAME MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE300	0x00	avi_inf_pb_0_1	r	avi_inf_pb[7]	avi_inf_pb[6]	avi_inf_pb[5]	avi_inf_pb[4]	avi_inf_pb[3]	avi_inf_pb[2]	avi_inf_pb[1]	avi_inf_pb[0]
0xE301	0x00	avi_inf_pb_0_2	r	avi_inf_pb[15]	avi_inf_pb[14]	avi_inf_pb[13]	avi_inf_pb[12]	avi_inf_pb[11]	avi_inf_pb[10]	avi_inf_pb[9]	avi_inf_pb[8]
0xE302	0x00	avi_inf_pb_0_3	r	avi_inf_pb[23]	avi_inf_pb[22]	avi_inf_pb[21]	avi_inf_pb[20]	avi_inf_pb[19]	avi_inf_pb[18]	avi_inf_pb[17]	avi_inf_pb[16]
0xE303	0x00	avi_inf_pb_0_4	r	avi_inf_pb[31]	avi_inf_pb[30]	avi_inf_pb[29]	avi_inf_pb[28]	avi_inf_pb[27]	avi_inf_pb[26]	avi_inf_pb[25]	avi_inf_pb[24]
0xE304	0x00	avi_inf_pb_0_5	r	avi_inf_pb[39]	avi_inf_pb[38]	avi_inf_pb[37]	avi_inf_pb[36]	avi_inf_pb[35]	avi_inf_pb[34]	avi_inf_pb[33]	avi_inf_pb[32]
0xE305	0x00	avi_inf_pb_0_6	r	avi_inf_pb[47]	avi_inf_pb[46]	avi_inf_pb[45]	avi_inf_pb[44]	avi_inf_pb[43]	avi_inf_pb[42]	avi_inf_pb[41]	avi_inf_pb[40]
0xE306	0x00	avi_inf_pb_0_7	r	avi_inf_pb[55]	avi_inf_pb[54]	avi_inf_pb[53]	avi_inf_pb[52]	avi_inf_pb[51]	avi_inf_pb[50]	avi_inf_pb[49]	avi_inf_pb[48]
0xE307	0x00	avi_inf_pb_0_8	r	avi_inf_pb[63]	avi_inf_pb[62]	avi_inf_pb[61]	avi_inf_pb[60]	avi_inf_pb[59]	avi_inf_pb[58]	avi_inf_pb[57]	avi_inf_pb[56]
0xE308	0x00	avi_inf_pb_0_9	r	avi_inf_pb[71]	avi_inf_pb[70]	avi_inf_pb[69]	avi_inf_pb[68]	avi_inf_pb[67]	avi_inf_pb[66]	avi_inf_pb[65]	avi_inf_pb[64]
0xE309	0x00	avi_inf_pb_0_10	r	avi_inf_pb[79]	avi_inf_pb[78]	avi_inf_pb[77]	avi_inf_pb[76]	avi_inf_pb[75]	avi_inf_pb[74]	avi_inf_pb[73]	avi_inf_pb[72]
0xE30A	0x00	avi_inf_pb_0_11	r	avi_inf_pb[87]	avi_inf_pb[86]	avi_inf_pb[85]	avi_inf_pb[84]	avi_inf_pb[83]	avi_inf_pb[82]	avi_inf_pb[81]	avi_inf_pb[80]
0xE30B	0x00	avi_inf_pb_0_12	r	avi_inf_pb[95]	avi_inf_pb[94]	avi_inf_pb[93]	avi_inf_pb[92]	avi_inf_pb[91]	avi_inf_pb[90]	avi_inf_pb[89]	avi_inf_pb[88]
0xE30C	0x00	avi_inf_pb_0_13	r	avi_inf_pb[103]	avi_inf_pb[102]	avi_inf_pb[101]	avi_inf_pb[100]	avi_inf_pb[99]	avi_inf_pb[98]	avi_inf_pb[97]	avi_inf_pb[96]
0xE30D	0x00	avi_inf_pb_0_14	r	avi_inf_pb[111]	avi_inf_pb[110]	avi_inf_pb[109]	avi_inf_pb[108]	avi_inf_pb[107]	avi_inf_pb[106]	avi_inf_pb[105]	avi_inf_pb[104]
0xE30E	0x00	avi_inf_pb_0_15	r	avi_inf_pb[119]	avi_inf_pb[118]	avi_inf_pb[117]	avi_inf_pb[116]	avi_inf_pb[115]	avi_inf_pb[114]	avi_inf_pb[113]	avi_inf_pb[112]
0xE30F	0x00	avi_inf_pb_0_16	r	avi_inf_pb[127]	avi_inf_pb[126]	avi_inf_pb[125]	avi_inf_pb[124]	avi_inf_pb[123]	avi_inf_pb[122]	avi_inf_pb[121]	avi_inf_pb[120]
0xE310	0x00	avi_inf_pb_0_17	r	avi_inf_pb[135]	avi_inf_pb[134]	avi_inf_pb[133]	avi_inf_pb[132]	avi_inf_pb[131]	avi_inf_pb[130]	avi_inf_pb[129]	avi_inf_pb[128]
0xE311	0x00	avi_inf_pb_0_18	r	avi_inf_pb[143]	avi_inf_pb[142]	avi_inf_pb[141]	avi_inf_pb[140]	avi_inf_pb[139]	avi_inf_pb[138]	avi_inf_pb[137]	avi_inf_pb[136]
0xE312	0x00	avi_inf_pb_0_19	r	avi_inf_pb[151]	avi_inf_pb[150]	avi_inf_pb[149]	avi_inf_pb[148]	avi_inf_pb[147]	avi_inf_pb[146]	avi_inf_pb[145]	avi_inf_pb[144]
0xE313	0x00	avi_inf_pb_0_20	r	avi_inf_pb[159]	avi_inf_pb[158]	avi_inf_pb[157]	avi_inf_pb[156]	avi_inf_pb[155]	avi_inf_pb[154]	avi_inf_pb[153]	avi_inf_pb[152]
0xE314	0x00	avi_inf_pb_0_21	r	avi_inf_pb[167]	avi_inf_pb[166]	avi_inf_pb[165]	avi_inf_pb[164]	avi_inf_pb[163]	avi_inf_pb[162]	avi_inf_pb[161]	avi_inf_pb[160]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE3 15	0x00	avi_inf_pb_0_22	r	avi_inf_pb[175]	avi_inf_pb[174]	avi_inf_pb[173]	avi_inf_pb[172]	avi_inf_pb[171]	avi_inf_pb[170]	avi_inf_pb[169]	avi_inf_pb[168]
0xE3 16	0x00	avi_inf_pb_0_23	r	avi_inf_pb[183]	avi_inf_pb[182]	avi_inf_pb[181]	avi_inf_pb[180]	avi_inf_pb[179]	avi_inf_pb[178]	avi_inf_pb[177]	avi_inf_pb[176]
0xE3 17	0x00	avi_inf_pb_0_24	r	avi_inf_pb[191]	avi_inf_pb[190]	avi_inf_pb[189]	avi_inf_pb[188]	avi_inf_pb[187]	avi_inf_pb[186]	avi_inf_pb[185]	avi_inf_pb[184]
0xE3 18	0x00	avi_inf_pb_0_25	r	avi_inf_pb[199]	avi_inf_pb[198]	avi_inf_pb[197]	avi_inf_pb[196]	avi_inf_pb[195]	avi_inf_pb[194]	avi_inf_pb[193]	avi_inf_pb[192]
0xE3 19	0x00	avi_inf_pb_0_26	r	avi_inf_pb[207]	avi_inf_pb[206]	avi_inf_pb[205]	avi_inf_pb[204]	avi_inf_pb[203]	avi_inf_pb[202]	avi_inf_pb[201]	avi_inf_pb[200]
0xE3 1A	0x00	avi_inf_pb_0_27	r	avi_inf_pb[215]	avi_inf_pb[214]	avi_inf_pb[213]	avi_inf_pb[212]	avi_inf_pb[211]	avi_inf_pb[210]	avi_inf_pb[209]	avi_inf_pb[208]
0xE3 1B	0x00	avi_inf_pb_0_28	r	avi_inf_pb[223]	avi_inf_pb[222]	avi_inf_pb[221]	avi_inf_pb[220]	avi_inf_pb[219]	avi_inf_pb[218]	avi_inf_pb[217]	avi_inf_pb[216]
0xE3 1C	0x00	aud_inf_pb_0_1	r	aud_inf_pb[7]	aud_inf_pb[6]	aud_inf_pb[5]	aud_inf_pb[4]	aud_inf_pb[3]	aud_inf_pb[2]	aud_inf_pb[1]	aud_inf_pb[0]
0xE3 1D	0x00	aud_inf_pb_0_2	r	aud_inf_pb[15]	aud_inf_pb[14]	aud_inf_pb[13]	aud_inf_pb[12]	aud_inf_pb[11]	aud_inf_pb[10]	aud_inf_pb[9]	aud_inf_pb[8]
0xE3 1E	0x00	aud_inf_pb_0_3	r	aud_inf_pb[23]	aud_inf_pb[22]	aud_inf_pb[21]	aud_inf_pb[20]	aud_inf_pb[19]	aud_inf_pb[18]	aud_inf_pb[17]	aud_inf_pb[16]
0xE3 1F	0x00	aud_inf_pb_0_4	r	aud_inf_pb[31]	aud_inf_pb[30]	aud_inf_pb[29]	aud_inf_pb[28]	aud_inf_pb[27]	aud_inf_pb[26]	aud_inf_pb[25]	aud_inf_pb[24]
0xE3 20	0x00	aud_inf_pb_0_5	r	aud_inf_pb[39]	aud_inf_pb[38]	aud_inf_pb[37]	aud_inf_pb[36]	aud_inf_pb[35]	aud_inf_pb[34]	aud_inf_pb[33]	aud_inf_pb[32]
0xE3 21	0x00	aud_inf_pb_0_6	r	aud_inf_pb[47]	aud_inf_pb[46]	aud_inf_pb[45]	aud_inf_pb[44]	aud_inf_pb[43]	aud_inf_pb[42]	aud_inf_pb[41]	aud_inf_pb[40]
0xE3 22	0x00	aud_inf_pb_0_7	r	aud_inf_pb[55]	aud_inf_pb[54]	aud_inf_pb[53]	aud_inf_pb[52]	aud_inf_pb[51]	aud_inf_pb[50]	aud_inf_pb[49]	aud_inf_pb[48]
0xE3 23	0x00	aud_inf_pb_0_8	r	aud_inf_pb[63]	aud_inf_pb[62]	aud_inf_pb[61]	aud_inf_pb[60]	aud_inf_pb[59]	aud_inf_pb[58]	aud_inf_pb[57]	aud_inf_pb[56]
0xE3 24	0x00	aud_inf_pb_0_9	r	aud_inf_pb[71]	aud_inf_pb[70]	aud_inf_pb[69]	aud_inf_pb[68]	aud_inf_pb[67]	aud_inf_pb[66]	aud_inf_pb[65]	aud_inf_pb[64]
0xE3 25	0x00	aud_inf_pb_0_10	r	aud_inf_pb[79]	aud_inf_pb[78]	aud_inf_pb[77]	aud_inf_pb[76]	aud_inf_pb[75]	aud_inf_pb[74]	aud_inf_pb[73]	aud_inf_pb[72]
0xE3 26	0x00	aud_inf_pb_0_11	r	aud_inf_pb[87]	aud_inf_pb[86]	aud_inf_pb[85]	aud_inf_pb[84]	aud_inf_pb[83]	aud_inf_pb[82]	aud_inf_pb[81]	aud_inf_pb[80]
0xE3 27	0x00	aud_inf_pb_0_12	r	aud_inf_pb[95]	aud_inf_pb[94]	aud_inf_pb[93]	aud_inf_pb[92]	aud_inf_pb[91]	aud_inf_pb[90]	aud_inf_pb[89]	aud_inf_pb[88]
0xE3 28	0x00	aud_inf_pb_0_13	r	aud_inf_pb[103]	aud_inf_pb[102]	aud_inf_pb[101]	aud_inf_pb[100]	aud_inf_pb[99]	aud_inf_pb[98]	aud_inf_pb[97]	aud_inf_pb[96]
0xE3 29	0x00	aud_inf_pb_0_14	r	aud_inf_pb[111]	aud_inf_pb[110]	aud_inf_pb[109]	aud_inf_pb[108]	aud_inf_pb[107]	aud_inf_pb[106]	aud_inf_pb[105]	aud_inf_pb[104]
0xE3 2A	0x00	spd_inf_pb_0_1	r	spd_inf_pb[7]	spd_inf_pb[6]	spd_inf_pb[5]	spd_inf_pb[4]	spd_inf_pb[3]	spd_inf_pb[2]	spd_inf_pb[1]	spd_inf_pb[0]
0xE3 2B	0x00	spd_inf_pb_0_2	r	spd_inf_pb[15]	spd_inf_pb[14]	spd_inf_pb[13]	spd_inf_pb[12]	spd_inf_pb[11]	spd_inf_pb[10]	spd_inf_pb[9]	spd_inf_pb[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE32C	0x00	spd_inf_pb_0_3	r	spd_inf_pb[23]	spd_inf_pb[22]	spd_inf_pb[21]	spd_inf_pb[20]	spd_inf_pb[19]	spd_inf_pb[18]	spd_inf_pb[17]	spd_inf_pb[16]
0xE32D	0x00	spd_inf_pb_0_4	r	spd_inf_pb[31]	spd_inf_pb[30]	spd_inf_pb[29]	spd_inf_pb[28]	spd_inf_pb[27]	spd_inf_pb[26]	spd_inf_pb[25]	spd_inf_pb[24]
0xE32E	0x00	spd_inf_pb_0_5	r	spd_inf_pb[39]	spd_inf_pb[38]	spd_inf_pb[37]	spd_inf_pb[36]	spd_inf_pb[35]	spd_inf_pb[34]	spd_inf_pb[33]	spd_inf_pb[32]
0xE32F	0x00	spd_inf_pb_0_6	r	spd_inf_pb[47]	spd_inf_pb[46]	spd_inf_pb[45]	spd_inf_pb[44]	spd_inf_pb[43]	spd_inf_pb[42]	spd_inf_pb[41]	spd_inf_pb[40]
0xE330	0x00	spd_inf_pb_0_7	r	spd_inf_pb[55]	spd_inf_pb[54]	spd_inf_pb[53]	spd_inf_pb[52]	spd_inf_pb[51]	spd_inf_pb[50]	spd_inf_pb[49]	spd_inf_pb[48]
0xE331	0x00	spd_inf_pb_0_8	r	spd_inf_pb[63]	spd_inf_pb[62]	spd_inf_pb[61]	spd_inf_pb[60]	spd_inf_pb[59]	spd_inf_pb[58]	spd_inf_pb[57]	spd_inf_pb[56]
0xE332	0x00	spd_inf_pb_0_9	r	spd_inf_pb[71]	spd_inf_pb[70]	spd_inf_pb[69]	spd_inf_pb[68]	spd_inf_pb[67]	spd_inf_pb[66]	spd_inf_pb[65]	spd_inf_pb[64]
0xE333	0x00	spd_inf_pb_0_10	r	spd_inf_pb[79]	spd_inf_pb[78]	spd_inf_pb[77]	spd_inf_pb[76]	spd_inf_pb[75]	spd_inf_pb[74]	spd_inf_pb[73]	spd_inf_pb[72]
0xE334	0x00	spd_inf_pb_0_11	r	spd_inf_pb[87]	spd_inf_pb[86]	spd_inf_pb[85]	spd_inf_pb[84]	spd_inf_pb[83]	spd_inf_pb[82]	spd_inf_pb[81]	spd_inf_pb[80]
0xE335	0x00	spd_inf_pb_0_12	r	spd_inf_pb[95]	spd_inf_pb[94]	spd_inf_pb[93]	spd_inf_pb[92]	spd_inf_pb[91]	spd_inf_pb[90]	spd_inf_pb[89]	spd_inf_pb[88]
0xE336	0x00	spd_inf_pb_0_13	r	spd_inf_pb[103]	spd_inf_pb[102]	spd_inf_pb[101]	spd_inf_pb[100]	spd_inf_pb[99]	spd_inf_pb[98]	spd_inf_pb[97]	spd_inf_pb[96]
0xE337	0x00	spd_inf_pb_0_14	r	spd_inf_pb[111]	spd_inf_pb[110]	spd_inf_pb[109]	spd_inf_pb[108]	spd_inf_pb[107]	spd_inf_pb[106]	spd_inf_pb[105]	spd_inf_pb[104]
0xE338	0x00	spd_inf_pb_0_15	r	spd_inf_pb[119]	spd_inf_pb[118]	spd_inf_pb[117]	spd_inf_pb[116]	spd_inf_pb[115]	spd_inf_pb[114]	spd_inf_pb[113]	spd_inf_pb[112]
0xE339	0x00	spd_inf_pb_0_16	r	spd_inf_pb[127]	spd_inf_pb[126]	spd_inf_pb[125]	spd_inf_pb[124]	spd_inf_pb[123]	spd_inf_pb[122]	spd_inf_pb[121]	spd_inf_pb[120]
0xE33A	0x00	spd_inf_pb_0_17	r	spd_inf_pb[135]	spd_inf_pb[134]	spd_inf_pb[133]	spd_inf_pb[132]	spd_inf_pb[131]	spd_inf_pb[130]	spd_inf_pb[129]	spd_inf_pb[128]
0xE33B	0x00	spd_inf_pb_0_18	r	spd_inf_pb[143]	spd_inf_pb[142]	spd_inf_pb[141]	spd_inf_pb[140]	spd_inf_pb[139]	spd_inf_pb[138]	spd_inf_pb[137]	spd_inf_pb[136]
0xE33C	0x00	spd_inf_pb_0_19	r	spd_inf_pb[151]	spd_inf_pb[150]	spd_inf_pb[149]	spd_inf_pb[148]	spd_inf_pb[147]	spd_inf_pb[146]	spd_inf_pb[145]	spd_inf_pb[144]
0xE33D	0x00	spd_inf_pb_0_20	r	spd_inf_pb[159]	spd_inf_pb[158]	spd_inf_pb[157]	spd_inf_pb[156]	spd_inf_pb[155]	spd_inf_pb[154]	spd_inf_pb[153]	spd_inf_pb[152]
0xE33E	0x00	spd_inf_pb_0_21	r	spd_inf_pb[167]	spd_inf_pb[166]	spd_inf_pb[165]	spd_inf_pb[164]	spd_inf_pb[163]	spd_inf_pb[162]	spd_inf_pb[161]	spd_inf_pb[160]
0xE33F	0x00	spd_inf_pb_0_22	r	spd_inf_pb[175]	spd_inf_pb[174]	spd_inf_pb[173]	spd_inf_pb[172]	spd_inf_pb[171]	spd_inf_pb[170]	spd_inf_pb[169]	spd_inf_pb[168]
0xE340	0x00	spd_inf_pb_0_23	r	spd_inf_pb[183]	spd_inf_pb[182]	spd_inf_pb[181]	spd_inf_pb[180]	spd_inf_pb[179]	spd_inf_pb[178]	spd_inf_pb[177]	spd_inf_pb[176]
0xE341	0x00	spd_inf_pb_0_24	r	spd_inf_pb[191]	spd_inf_pb[190]	spd_inf_pb[189]	spd_inf_pb[188]	spd_inf_pb[187]	spd_inf_pb[186]	spd_inf_pb[185]	spd_inf_pb[184]
0xE342	0x00	spd_inf_pb_0_25	r	spd_inf_pb[199]	spd_inf_pb[198]	spd_inf_pb[197]	spd_inf_pb[196]	spd_inf_pb[195]	spd_inf_pb[194]	spd_inf_pb[193]	spd_inf_pb[192]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE343	0x00	spd_inf_pb_0_26	r	spd_inf_pb[207]	spd_inf_pb[206]	spd_inf_pb[205]	spd_inf_pb[204]	spd_inf_pb[203]	spd_inf_pb[202]	spd_inf_pb[201]	spd_inf_pb[200]
0xE344	0x00	spd_inf_pb_0_27	r	spd_inf_pb[215]	spd_inf_pb[214]	spd_inf_pb[213]	spd_inf_pb[212]	spd_inf_pb[211]	spd_inf_pb[210]	spd_inf_pb[209]	spd_inf_pb[208]
0xE345	0x00	spd_inf_pb_0_28	r	spd_inf_pb[223]	spd_inf_pb[222]	spd_inf_pb[221]	spd_inf_pb[220]	spd_inf_pb[219]	spd_inf_pb[218]	spd_inf_pb[217]	spd_inf_pb[216]
0xE346	0x00	ms_inf_pb_0_1	r	ms_inf_pb[7]	ms_inf_pb[6]	ms_inf_pb[5]	ms_inf_pb[4]	ms_inf_pb[3]	ms_inf_pb[2]	ms_inf_pb[1]	ms_inf_pb[0]
0xE347	0x00	ms_inf_pb_0_2	r	ms_inf_pb[15]	ms_inf_pb[14]	ms_inf_pb[13]	ms_inf_pb[12]	ms_inf_pb[11]	ms_inf_pb[10]	ms_inf_pb[9]	ms_inf_pb[8]
0xE348	0x00	ms_inf_pb_0_3	r	ms_inf_pb[23]	ms_inf_pb[22]	ms_inf_pb[21]	ms_inf_pb[20]	ms_inf_pb[19]	ms_inf_pb[18]	ms_inf_pb[17]	ms_inf_pb[16]
0xE349	0x00	ms_inf_pb_0_4	r	ms_inf_pb[31]	ms_inf_pb[30]	ms_inf_pb[29]	ms_inf_pb[28]	ms_inf_pb[27]	ms_inf_pb[26]	ms_inf_pb[25]	ms_inf_pb[24]
0xE34A	0x00	ms_inf_pb_0_5	r	ms_inf_pb[39]	ms_inf_pb[38]	ms_inf_pb[37]	ms_inf_pb[36]	ms_inf_pb[35]	ms_inf_pb[34]	ms_inf_pb[33]	ms_inf_pb[32]
0xE34B	0x00	ms_inf_pb_0_6	r	ms_inf_pb[47]	ms_inf_pb[46]	ms_inf_pb[45]	ms_inf_pb[44]	ms_inf_pb[43]	ms_inf_pb[42]	ms_inf_pb[41]	ms_inf_pb[40]
0xE34C	0x00	ms_inf_pb_0_7	r	ms_inf_pb[55]	ms_inf_pb[54]	ms_inf_pb[53]	ms_inf_pb[52]	ms_inf_pb[51]	ms_inf_pb[50]	ms_inf_pb[49]	ms_inf_pb[48]
0xE34D	0x00	ms_inf_pb_0_8	r	ms_inf_pb[63]	ms_inf_pb[62]	ms_inf_pb[61]	ms_inf_pb[60]	ms_inf_pb[59]	ms_inf_pb[58]	ms_inf_pb[57]	ms_inf_pb[56]
0xE34E	0x00	ms_inf_pb_0_9	r	ms_inf_pb[71]	ms_inf_pb[70]	ms_inf_pb[69]	ms_inf_pb[68]	ms_inf_pb[67]	ms_inf_pb[66]	ms_inf_pb[65]	ms_inf_pb[64]
0xE34F	0x00	ms_inf_pb_0_10	r	ms_inf_pb[79]	ms_inf_pb[78]	ms_inf_pb[77]	ms_inf_pb[76]	ms_inf_pb[75]	ms_inf_pb[74]	ms_inf_pb[73]	ms_inf_pb[72]
0xE350	0x00	ms_inf_pb_0_11	r	ms_inf_pb[87]	ms_inf_pb[86]	ms_inf_pb[85]	ms_inf_pb[84]	ms_inf_pb[83]	ms_inf_pb[82]	ms_inf_pb[81]	ms_inf_pb[80]
0xE351	0x00	ms_inf_pb_0_12	r	ms_inf_pb[95]	ms_inf_pb[94]	ms_inf_pb[93]	ms_inf_pb[92]	ms_inf_pb[91]	ms_inf_pb[90]	ms_inf_pb[89]	ms_inf_pb[88]
0xE352	0x00	ms_inf_pb_0_13	r	ms_inf_pb[103]	ms_inf_pb[102]	ms_inf_pb[101]	ms_inf_pb[100]	ms_inf_pb[99]	ms_inf_pb[98]	ms_inf_pb[97]	ms_inf_pb[96]
0xE353	0x00	ms_inf_pb_0_14	r	ms_inf_pb[111]	ms_inf_pb[110]	ms_inf_pb[109]	ms_inf_pb[108]	ms_inf_pb[107]	ms_inf_pb[106]	ms_inf_pb[105]	ms_inf_pb[104]
0xE354	0x00	vs_inf_pb_0_1	r	vs_inf_pb[7]	vs_inf_pb[6]	vs_inf_pb[5]	vs_inf_pb[4]	vs_inf_pb[3]	vs_inf_pb[2]	vs_inf_pb[1]	vs_inf_pb[0]
0xE355	0x00	vs_inf_pb_0_2	r	vs_inf_pb[15]	vs_inf_pb[14]	vs_inf_pb[13]	vs_inf_pb[12]	vs_inf_pb[11]	vs_inf_pb[10]	vs_inf_pb[9]	vs_inf_pb[8]
0xE356	0x00	vs_inf_pb_0_3	r	vs_inf_pb[23]	vs_inf_pb[22]	vs_inf_pb[21]	vs_inf_pb[20]	vs_inf_pb[19]	vs_inf_pb[18]	vs_inf_pb[17]	vs_inf_pb[16]
0xE357	0x00	vs_inf_pb_0_4	r	vs_inf_pb[31]	vs_inf_pb[30]	vs_inf_pb[29]	vs_inf_pb[28]	vs_inf_pb[27]	vs_inf_pb[26]	vs_inf_pb[25]	vs_inf_pb[24]
0xE358	0x00	vs_inf_pb_0_5	r	vs_inf_pb[39]	vs_inf_pb[38]	vs_inf_pb[37]	vs_inf_pb[36]	vs_inf_pb[35]	vs_inf_pb[34]	vs_inf_pb[33]	vs_inf_pb[32]
0xE359	0x00	vs_inf_pb_0_6	r	vs_inf_pb[47]	vs_inf_pb[46]	vs_inf_pb[45]	vs_inf_pb[44]	vs_inf_pb[43]	vs_inf_pb[42]	vs_inf_pb[41]	vs_inf_pb[40]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE35A	0x00	vs_inf_pb_0_7	r	vs_inf_pb[55]	vs_inf_pb[54]	vs_inf_pb[53]	vs_inf_pb[52]	vs_inf_pb[51]	vs_inf_pb[50]	vs_inf_pb[49]	vs_inf_pb[48]
0xE35B	0x00	vs_inf_pb_0_8	r	vs_inf_pb[63]	vs_inf_pb[62]	vs_inf_pb[61]	vs_inf_pb[60]	vs_inf_pb[59]	vs_inf_pb[58]	vs_inf_pb[57]	vs_inf_pb[56]
0xE35C	0x00	vs_inf_pb_0_9	r	vs_inf_pb[71]	vs_inf_pb[70]	vs_inf_pb[69]	vs_inf_pb[68]	vs_inf_pb[67]	vs_inf_pb[66]	vs_inf_pb[65]	vs_inf_pb[64]
0xE35D	0x00	vs_inf_pb_0_10	r	vs_inf_pb[79]	vs_inf_pb[78]	vs_inf_pb[77]	vs_inf_pb[76]	vs_inf_pb[75]	vs_inf_pb[74]	vs_inf_pb[73]	vs_inf_pb[72]
0xE35E	0x00	vs_inf_pb_0_11	r	vs_inf_pb[87]	vs_inf_pb[86]	vs_inf_pb[85]	vs_inf_pb[84]	vs_inf_pb[83]	vs_inf_pb[82]	vs_inf_pb[81]	vs_inf_pb[80]
0xE35F	0x00	vs_inf_pb_0_12	r	vs_inf_pb[95]	vs_inf_pb[94]	vs_inf_pb[93]	vs_inf_pb[92]	vs_inf_pb[91]	vs_inf_pb[90]	vs_inf_pb[89]	vs_inf_pb[88]
0xE360	0x00	vs_inf_pb_0_13	r	vs_inf_pb[103]	vs_inf_pb[102]	vs_inf_pb[101]	vs_inf_pb[100]	vs_inf_pb[99]	vs_inf_pb[98]	vs_inf_pb[97]	vs_inf_pb[96]
0xE361	0x00	vs_inf_pb_0_14	r	vs_inf_pb[111]	vs_inf_pb[110]	vs_inf_pb[109]	vs_inf_pb[108]	vs_inf_pb[107]	vs_inf_pb[106]	vs_inf_pb[105]	vs_inf_pb[104]
0xE362	0x00	vs_inf_pb_0_15	r	vs_inf_pb[119]	vs_inf_pb[118]	vs_inf_pb[117]	vs_inf_pb[116]	vs_inf_pb[115]	vs_inf_pb[114]	vs_inf_pb[113]	vs_inf_pb[112]
0xE363	0x00	vs_inf_pb_0_16	r	vs_inf_pb[127]	vs_inf_pb[126]	vs_inf_pb[125]	vs_inf_pb[124]	vs_inf_pb[123]	vs_inf_pb[122]	vs_inf_pb[121]	vs_inf_pb[120]
0xE364	0x00	vs_inf_pb_0_17	r	vs_inf_pb[135]	vs_inf_pb[134]	vs_inf_pb[133]	vs_inf_pb[132]	vs_inf_pb[131]	vs_inf_pb[130]	vs_inf_pb[129]	vs_inf_pb[128]
0xE365	0x00	vs_inf_pb_0_18	r	vs_inf_pb[143]	vs_inf_pb[142]	vs_inf_pb[141]	vs_inf_pb[140]	vs_inf_pb[139]	vs_inf_pb[138]	vs_inf_pb[137]	vs_inf_pb[136]
0xE366	0x00	vs_inf_pb_0_19	r	vs_inf_pb[151]	vs_inf_pb[150]	vs_inf_pb[149]	vs_inf_pb[148]	vs_inf_pb[147]	vs_inf_pb[146]	vs_inf_pb[145]	vs_inf_pb[144]
0xE367	0x00	vs_inf_pb_0_20	r	vs_inf_pb[159]	vs_inf_pb[158]	vs_inf_pb[157]	vs_inf_pb[156]	vs_inf_pb[155]	vs_inf_pb[154]	vs_inf_pb[153]	vs_inf_pb[152]
0xE368	0x00	vs_inf_pb_0_21	r	vs_inf_pb[167]	vs_inf_pb[166]	vs_inf_pb[165]	vs_inf_pb[164]	vs_inf_pb[163]	vs_inf_pb[162]	vs_inf_pb[161]	vs_inf_pb[160]
0xE369	0x00	vs_inf_pb_0_22	r	vs_inf_pb[175]	vs_inf_pb[174]	vs_inf_pb[173]	vs_inf_pb[172]	vs_inf_pb[171]	vs_inf_pb[170]	vs_inf_pb[169]	vs_inf_pb[168]
0xE36A	0x00	vs_inf_pb_0_23	r	vs_inf_pb[183]	vs_inf_pb[182]	vs_inf_pb[181]	vs_inf_pb[180]	vs_inf_pb[179]	vs_inf_pb[178]	vs_inf_pb[177]	vs_inf_pb[176]
0xE36B	0x00	vs_inf_pb_0_24	r	vs_inf_pb[191]	vs_inf_pb[190]	vs_inf_pb[189]	vs_inf_pb[188]	vs_inf_pb[187]	vs_inf_pb[186]	vs_inf_pb[185]	vs_inf_pb[184]
0xE36C	0x00	vs_inf_pb_0_25	r	vs_inf_pb[199]	vs_inf_pb[198]	vs_inf_pb[197]	vs_inf_pb[196]	vs_inf_pb[195]	vs_inf_pb[194]	vs_inf_pb[193]	vs_inf_pb[192]
0xE36D	0x00	vs_inf_pb_0_26	r	vs_inf_pb[207]	vs_inf_pb[206]	vs_inf_pb[205]	vs_inf_pb[204]	vs_inf_pb[203]	vs_inf_pb[202]	vs_inf_pb[201]	vs_inf_pb[200]
0xE36E	0x00	vs_inf_pb_0_27	r	vs_inf_pb[215]	vs_inf_pb[214]	vs_inf_pb[213]	vs_inf_pb[212]	vs_inf_pb[211]	vs_inf_pb[210]	vs_inf_pb[209]	vs_inf_pb[208]
0xE36F	0x00	vs_inf_pb_0_28	r	vs_inf_pb[223]	vs_inf_pb[222]	vs_inf_pb[221]	vs_inf_pb[220]	vs_inf_pb[219]	vs_inf_pb[218]	vs_inf_pb[217]	vs_inf_pb[216]
0xE370	0x00	acp_pb_0_1	r	acp_pb[7]	acp_pb[6]	acp_pb[5]	acp_pb[4]	acp_pb[3]	acp_pb[2]	acp_pb[1]	acp_pb[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE371	0x00	acp_pb_0_2	r	acp_pb[15]	acp_pb[14]	acp_pb[13]	acp_pb[12]	acp_pb[11]	acp_pb[10]	acp_pb[9]	acp_pb[8]
0xE372	0x00	acp_pb_0_3	r	acp_pb[23]	acp_pb[22]	acp_pb[21]	acp_pb[20]	acp_pb[19]	acp_pb[18]	acp_pb[17]	acp_pb[16]
0xE373	0x00	acp_pb_0_4	r	acp_pb[31]	acp_pb[30]	acp_pb[29]	acp_pb[28]	acp_pb[27]	acp_pb[26]	acp_pb[25]	acp_pb[24]
0xE374	0x00	acp_pb_0_5	r	acp_pb[39]	acp_pb[38]	acp_pb[37]	acp_pb[36]	acp_pb[35]	acp_pb[34]	acp_pb[33]	acp_pb[32]
0xE375	0x00	acp_pb_0_6	r	acp_pb[47]	acp_pb[46]	acp_pb[45]	acp_pb[44]	acp_pb[43]	acp_pb[42]	acp_pb[41]	acp_pb[40]
0xE376	0x00	acp_pb_0_7	r	acp_pb[55]	acp_pb[54]	acp_pb[53]	acp_pb[52]	acp_pb[51]	acp_pb[50]	acp_pb[49]	acp_pb[48]
0xE377	0x00	acp_pb_0_8	r	acp_pb[63]	acp_pb[62]	acp_pb[61]	acp_pb[60]	acp_pb[59]	acp_pb[58]	acp_pb[57]	acp_pb[56]
0xE378	0x00	acp_pb_0_9	r	acp_pb[71]	acp_pb[70]	acp_pb[69]	acp_pb[68]	acp_pb[67]	acp_pb[66]	acp_pb[65]	acp_pb[64]
0xE379	0x00	acp_pb_0_10	r	acp_pb[79]	acp_pb[78]	acp_pb[77]	acp_pb[76]	acp_pb[75]	acp_pb[74]	acp_pb[73]	acp_pb[72]
0xE37A	0x00	acp_pb_0_11	r	acp_pb[87]	acp_pb[86]	acp_pb[85]	acp_pb[84]	acp_pb[83]	acp_pb[82]	acp_pb[81]	acp_pb[80]
0xE37B	0x00	acp_pb_0_12	r	acp_pb[95]	acp_pb[94]	acp_pb[93]	acp_pb[92]	acp_pb[91]	acp_pb[90]	acp_pb[89]	acp_pb[88]
0xE37C	0x00	acp_pb_0_13	r	acp_pb[103]	acp_pb[102]	acp_pb[101]	acp_pb[100]	acp_pb[99]	acp_pb[98]	acp_pb[97]	acp_pb[96]
0xE37D	0x00	acp_pb_0_14	r	acp_pb[111]	acp_pb[110]	acp_pb[109]	acp_pb[108]	acp_pb[107]	acp_pb[106]	acp_pb[105]	acp_pb[104]
0xE37E	0x00	acp_pb_0_15	r	acp_pb[119]	acp_pb[118]	acp_pb[117]	acp_pb[116]	acp_pb[115]	acp_pb[114]	acp_pb[113]	acp_pb[112]
0xE37F	0x00	acp_pb_0_16	r	acp_pb[127]	acp_pb[126]	acp_pb[125]	acp_pb[124]	acp_pb[123]	acp_pb[122]	acp_pb[121]	acp_pb[120]
0xE380	0x00	acp_pb_0_17	r	acp_pb[135]	acp_pb[134]	acp_pb[133]	acp_pb[132]	acp_pb[131]	acp_pb[130]	acp_pb[129]	acp_pb[128]
0xE381	0x00	acp_pb_0_18	r	acp_pb[143]	acp_pb[142]	acp_pb[141]	acp_pb[140]	acp_pb[139]	acp_pb[138]	acp_pb[137]	acp_pb[136]
0xE382	0x00	acp_pb_0_19	r	acp_pb[151]	acp_pb[150]	acp_pb[149]	acp_pb[148]	acp_pb[147]	acp_pb[146]	acp_pb[145]	acp_pb[144]
0xE383	0x00	acp_pb_0_20	r	acp_pb[159]	acp_pb[158]	acp_pb[157]	acp_pb[156]	acp_pb[155]	acp_pb[154]	acp_pb[153]	acp_pb[152]
0xE384	0x00	acp_pb_0_21	r	acp_pb[167]	acp_pb[166]	acp_pb[165]	acp_pb[164]	acp_pb[163]	acp_pb[162]	acp_pb[161]	acp_pb[160]
0xE385	0x00	acp_pb_0_22	r	acp_pb[175]	acp_pb[174]	acp_pb[173]	acp_pb[172]	acp_pb[171]	acp_pb[170]	acp_pb[169]	acp_pb[168]
0xE386	0x00	acp_pb_0_23	r	acp_pb[183]	acp_pb[182]	acp_pb[181]	acp_pb[180]	acp_pb[179]	acp_pb[178]	acp_pb[177]	acp_pb[176]
0xE387	0x00	acp_pb_0_24	r	acp_pb[191]	acp_pb[190]	acp_pb[189]	acp_pb[188]	acp_pb[187]	acp_pb[186]	acp_pb[185]	acp_pb[184]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE388	0x00	acp_pb_0_25	r	acp_pb[199]	acp_pb[198]	acp_pb[197]	acp_pb[196]	acp_pb[195]	acp_pb[194]	acp_pb[193]	acp_pb[192]
0xE389	0x00	acp_pb_0_26	r	acp_pb[207]	acp_pb[206]	acp_pb[205]	acp_pb[204]	acp_pb[203]	acp_pb[202]	acp_pb[201]	acp_pb[200]
0xE38A	0x00	acp_pb_0_27	r	acp_pb[215]	acp_pb[214]	acp_pb[213]	acp_pb[212]	acp_pb[211]	acp_pb[210]	acp_pb[209]	acp_pb[208]
0xE38B	0x00	acp_pb_0_28	r	acp_pb[223]	acp_pb[222]	acp_pb[221]	acp_pb[220]	acp_pb[219]	acp_pb[218]	acp_pb[217]	acp_pb[216]
0xE38C	0x00	isrc1_pb_0_1	r	isrc1_pb[7]	isrc1_pb[6]	isrc1_pb[5]	isrc1_pb[4]	isrc1_pb[3]	isrc1_pb[2]	isrc1_pb[1]	isrc1_pb[0]
0xE38D	0x00	isrc1_pb_0_2	r	isrc1_pb[15]	isrc1_pb[14]	isrc1_pb[13]	isrc1_pb[12]	isrc1_pb[11]	isrc1_pb[10]	isrc1_pb[9]	isrc1_pb[8]
0xE38E	0x00	isrc1_pb_0_3	r	isrc1_pb[23]	isrc1_pb[22]	isrc1_pb[21]	isrc1_pb[20]	isrc1_pb[19]	isrc1_pb[18]	isrc1_pb[17]	isrc1_pb[16]
0xE38F	0x00	isrc1_pb_0_4	r	isrc1_pb[31]	isrc1_pb[30]	isrc1_pb[29]	isrc1_pb[28]	isrc1_pb[27]	isrc1_pb[26]	isrc1_pb[25]	isrc1_pb[24]
0xE390	0x00	isrc1_pb_0_5	r	isrc1_pb[39]	isrc1_pb[38]	isrc1_pb[37]	isrc1_pb[36]	isrc1_pb[35]	isrc1_pb[34]	isrc1_pb[33]	isrc1_pb[32]
0xE391	0x00	isrc1_pb_0_6	r	isrc1_pb[47]	isrc1_pb[46]	isrc1_pb[45]	isrc1_pb[44]	isrc1_pb[43]	isrc1_pb[42]	isrc1_pb[41]	isrc1_pb[40]
0xE392	0x00	isrc1_pb_0_7	r	isrc1_pb[55]	isrc1_pb[54]	isrc1_pb[53]	isrc1_pb[52]	isrc1_pb[51]	isrc1_pb[50]	isrc1_pb[49]	isrc1_pb[48]
0xE393	0x00	isrc1_pb_0_8	r	isrc1_pb[63]	isrc1_pb[62]	isrc1_pb[61]	isrc1_pb[60]	isrc1_pb[59]	isrc1_pb[58]	isrc1_pb[57]	isrc1_pb[56]
0xE394	0x00	isrc1_pb_0_9	r	isrc1_pb[71]	isrc1_pb[70]	isrc1_pb[69]	isrc1_pb[68]	isrc1_pb[67]	isrc1_pb[66]	isrc1_pb[65]	isrc1_pb[64]
0xE395	0x00	isrc1_pb_0_10	r	isrc1_pb[79]	isrc1_pb[78]	isrc1_pb[77]	isrc1_pb[76]	isrc1_pb[75]	isrc1_pb[74]	isrc1_pb[73]	isrc1_pb[72]
0xE396	0x00	isrc1_pb_0_11	r	isrc1_pb[87]	isrc1_pb[86]	isrc1_pb[85]	isrc1_pb[84]	isrc1_pb[83]	isrc1_pb[82]	isrc1_pb[81]	isrc1_pb[80]
0xE397	0x00	isrc1_pb_0_12	r	isrc1_pb[95]	isrc1_pb[94]	isrc1_pb[93]	isrc1_pb[92]	isrc1_pb[91]	isrc1_pb[90]	isrc1_pb[89]	isrc1_pb[88]
0xE398	0x00	isrc1_pb_0_13	r	isrc1_pb[103]	isrc1_pb[102]	isrc1_pb[101]	isrc1_pb[100]	isrc1_pb[99]	isrc1_pb[98]	isrc1_pb[97]	isrc1_pb[96]
0xE399	0x00	isrc1_pb_0_14	r	isrc1_pb[111]	isrc1_pb[110]	isrc1_pb[109]	isrc1_pb[108]	isrc1_pb[107]	isrc1_pb[106]	isrc1_pb[105]	isrc1_pb[104]
0xE39A	0x00	isrc1_pb_0_15	r	isrc1_pb[119]	isrc1_pb[118]	isrc1_pb[117]	isrc1_pb[116]	isrc1_pb[115]	isrc1_pb[114]	isrc1_pb[113]	isrc1_pb[112]
0xE39B	0x00	isrc1_pb_0_16	r	isrc1_pb[127]	isrc1_pb[126]	isrc1_pb[125]	isrc1_pb[124]	isrc1_pb[123]	isrc1_pb[122]	isrc1_pb[121]	isrc1_pb[120]
0xE39C	0x00	isrc1_pb_0_17	r	isrc1_pb[135]	isrc1_pb[134]	isrc1_pb[133]	isrc1_pb[132]	isrc1_pb[131]	isrc1_pb[130]	isrc1_pb[129]	isrc1_pb[128]
0xE39D	0x00	isrc1_pb_0_18	r	isrc1_pb[143]	isrc1_pb[142]	isrc1_pb[141]	isrc1_pb[140]	isrc1_pb[139]	isrc1_pb[138]	isrc1_pb[137]	isrc1_pb[136]
0xE39E	0x00	isrc1_pb_0_19	r	isrc1_pb[151]	isrc1_pb[150]	isrc1_pb[149]	isrc1_pb[148]	isrc1_pb[147]	isrc1_pb[146]	isrc1_pb[145]	isrc1_pb[144]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE39F	0x00	isrc1_pb_0_20	r	isrc1_pb[159]	isrc1_pb[158]	isrc1_pb[157]	isrc1_pb[156]	isrc1_pb[155]	isrc1_pb[154]	isrc1_pb[153]	isrc1_pb[152]
0xE3A0	0x00	isrc1_pb_0_21	r	isrc1_pb[167]	isrc1_pb[166]	isrc1_pb[165]	isrc1_pb[164]	isrc1_pb[163]	isrc1_pb[162]	isrc1_pb[161]	isrc1_pb[160]
0xE3A1	0x00	isrc1_pb_0_22	r	isrc1_pb[175]	isrc1_pb[174]	isrc1_pb[173]	isrc1_pb[172]	isrc1_pb[171]	isrc1_pb[170]	isrc1_pb[169]	isrc1_pb[168]
0xE3A2	0x00	isrc1_pb_0_23	r	isrc1_pb[183]	isrc1_pb[182]	isrc1_pb[181]	isrc1_pb[180]	isrc1_pb[179]	isrc1_pb[178]	isrc1_pb[177]	isrc1_pb[176]
0xE3A3	0x00	isrc1_pb_0_24	r	isrc1_pb[191]	isrc1_pb[190]	isrc1_pb[189]	isrc1_pb[188]	isrc1_pb[187]	isrc1_pb[186]	isrc1_pb[185]	isrc1_pb[184]
0xE3A4	0x00	isrc1_pb_0_25	r	isrc1_pb[199]	isrc1_pb[198]	isrc1_pb[197]	isrc1_pb[196]	isrc1_pb[195]	isrc1_pb[194]	isrc1_pb[193]	isrc1_pb[192]
0xE3A5	0x00	isrc1_pb_0_26	r	isrc1_pb[207]	isrc1_pb[206]	isrc1_pb[205]	isrc1_pb[204]	isrc1_pb[203]	isrc1_pb[202]	isrc1_pb[201]	isrc1_pb[200]
0xE3A6	0x00	isrc1_pb_0_27	r	isrc1_pb[215]	isrc1_pb[214]	isrc1_pb[213]	isrc1_pb[212]	isrc1_pb[211]	isrc1_pb[210]	isrc1_pb[209]	isrc1_pb[208]
0xE3A7	0x00	isrc1_pb_0_28	r	isrc1_pb[223]	isrc1_pb[222]	isrc1_pb[221]	isrc1_pb[220]	isrc1_pb[219]	isrc1_pb[218]	isrc1_pb[217]	isrc1_pb[216]
0xE3A8	0x00	isrc2_pb_0_1	r	isrc2_pb[7]	isrc2_pb[6]	isrc2_pb[5]	isrc2_pb[4]	isrc2_pb[3]	isrc2_pb[2]	isrc2_pb[1]	isrc2_pb[0]
0xE3A9	0x00	isrc2_pb_0_2	r	isrc2_pb[15]	isrc2_pb[14]	isrc2_pb[13]	isrc2_pb[12]	isrc2_pb[11]	isrc2_pb[10]	isrc2_pb[9]	isrc2_pb[8]
0xE3AA	0x00	isrc2_pb_0_3	r	isrc2_pb[23]	isrc2_pb[22]	isrc2_pb[21]	isrc2_pb[20]	isrc2_pb[19]	isrc2_pb[18]	isrc2_pb[17]	isrc2_pb[16]
0xE3AB	0x00	isrc2_pb_0_4	r	isrc2_pb[31]	isrc2_pb[30]	isrc2_pb[29]	isrc2_pb[28]	isrc2_pb[27]	isrc2_pb[26]	isrc2_pb[25]	isrc2_pb[24]
0xE3AC	0x00	isrc2_pb_0_5	r	isrc2_pb[39]	isrc2_pb[38]	isrc2_pb[37]	isrc2_pb[36]	isrc2_pb[35]	isrc2_pb[34]	isrc2_pb[33]	isrc2_pb[32]
0xE3AD	0x00	isrc2_pb_0_6	r	isrc2_pb[47]	isrc2_pb[46]	isrc2_pb[45]	isrc2_pb[44]	isrc2_pb[43]	isrc2_pb[42]	isrc2_pb[41]	isrc2_pb[40]
0xE3AE	0x00	isrc2_pb_0_7	r	isrc2_pb[55]	isrc2_pb[54]	isrc2_pb[53]	isrc2_pb[52]	isrc2_pb[51]	isrc2_pb[50]	isrc2_pb[49]	isrc2_pb[48]
0xE3AF	0x00	isrc2_pb_0_8	r	isrc2_pb[63]	isrc2_pb[62]	isrc2_pb[61]	isrc2_pb[60]	isrc2_pb[59]	isrc2_pb[58]	isrc2_pb[57]	isrc2_pb[56]
0xE3B0	0x00	isrc2_pb_0_9	r	isrc2_pb[71]	isrc2_pb[70]	isrc2_pb[69]	isrc2_pb[68]	isrc2_pb[67]	isrc2_pb[66]	isrc2_pb[65]	isrc2_pb[64]
0xE3B1	0x00	isrc2_pb_0_10	r	isrc2_pb[79]	isrc2_pb[78]	isrc2_pb[77]	isrc2_pb[76]	isrc2_pb[75]	isrc2_pb[74]	isrc2_pb[73]	isrc2_pb[72]
0xE3B2	0x00	isrc2_pb_0_11	r	isrc2_pb[87]	isrc2_pb[86]	isrc2_pb[85]	isrc2_pb[84]	isrc2_pb[83]	isrc2_pb[82]	isrc2_pb[81]	isrc2_pb[80]
0xE3B3	0x00	isrc2_pb_0_12	r	isrc2_pb[95]	isrc2_pb[94]	isrc2_pb[93]	isrc2_pb[92]	isrc2_pb[91]	isrc2_pb[90]	isrc2_pb[89]	isrc2_pb[88]
0xE3B4	0x00	isrc2_pb_0_13	r	isrc2_pb[103]	isrc2_pb[102]	isrc2_pb[101]	isrc2_pb[100]	isrc2_pb[99]	isrc2_pb[98]	isrc2_pb[97]	isrc2_pb[96]
0xE3B5	0x00	isrc2_pb_0_14	r	isrc2_pb[111]	isrc2_pb[110]	isrc2_pb[109]	isrc2_pb[108]	isrc2_pb[107]	isrc2_pb[106]	isrc2_pb[105]	isrc2_pb[104]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE3 B6	0x00	isrc2_pb_0_15	r	isrc2_pb[119]	isrc2_pb[118]	isrc2_pb[117]	isrc2_pb[116]	isrc2_pb[115]	isrc2_pb[114]	isrc2_pb[113]	isrc2_pb[112]
0xE3 B7	0x00	isrc2_pb_0_16	r	isrc2_pb[127]	isrc2_pb[126]	isrc2_pb[125]	isrc2_pb[124]	isrc2_pb[123]	isrc2_pb[122]	isrc2_pb[121]	isrc2_pb[120]
0xE3 B8	0x00	isrc2_pb_0_17	r	isrc2_pb[135]	isrc2_pb[134]	isrc2_pb[133]	isrc2_pb[132]	isrc2_pb[131]	isrc2_pb[130]	isrc2_pb[129]	isrc2_pb[128]
0xE3 B9	0x00	isrc2_pb_0_18	r	isrc2_pb[143]	isrc2_pb[142]	isrc2_pb[141]	isrc2_pb[140]	isrc2_pb[139]	isrc2_pb[138]	isrc2_pb[137]	isrc2_pb[136]
0xE3 BA	0x00	isrc2_pb_0_19	r	isrc2_pb[151]	isrc2_pb[150]	isrc2_pb[149]	isrc2_pb[148]	isrc2_pb[147]	isrc2_pb[146]	isrc2_pb[145]	isrc2_pb[144]
0xE3 BB	0x00	isrc2_pb_0_20	r	isrc2_pb[159]	isrc2_pb[158]	isrc2_pb[157]	isrc2_pb[156]	isrc2_pb[155]	isrc2_pb[154]	isrc2_pb[153]	isrc2_pb[152]
0xE3 BC	0x00	isrc2_pb_0_21	r	isrc2_pb[167]	isrc2_pb[166]	isrc2_pb[165]	isrc2_pb[164]	isrc2_pb[163]	isrc2_pb[162]	isrc2_pb[161]	isrc2_pb[160]
0xE3 BD	0x00	isrc2_pb_0_22	r	isrc2_pb[175]	isrc2_pb[174]	isrc2_pb[173]	isrc2_pb[172]	isrc2_pb[171]	isrc2_pb[170]	isrc2_pb[169]	isrc2_pb[168]
0xE3 BE	0x00	isrc2_pb_0_23	r	isrc2_pb[183]	isrc2_pb[182]	isrc2_pb[181]	isrc2_pb[180]	isrc2_pb[179]	isrc2_pb[178]	isrc2_pb[177]	isrc2_pb[176]
0xE3 BF	0x00	isrc2_pb_0_24	r	isrc2_pb[191]	isrc2_pb[190]	isrc2_pb[189]	isrc2_pb[188]	isrc2_pb[187]	isrc2_pb[186]	isrc2_pb[185]	isrc2_pb[184]
0xE3 C0	0x00	isrc2_pb_0_25	r	isrc2_pb[199]	isrc2_pb[198]	isrc2_pb[197]	isrc2_pb[196]	isrc2_pb[195]	isrc2_pb[194]	isrc2_pb[193]	isrc2_pb[192]
0xE3 C1	0x00	isrc2_pb_0_26	r	isrc2_pb[207]	isrc2_pb[206]	isrc2_pb[205]	isrc2_pb[204]	isrc2_pb[203]	isrc2_pb[202]	isrc2_pb[201]	isrc2_pb[200]
0xE3 C2	0x00	isrc2_pb_0_27	r	isrc2_pb[215]	isrc2_pb[214]	isrc2_pb[213]	isrc2_pb[212]	isrc2_pb[211]	isrc2_pb[210]	isrc2_pb[209]	isrc2_pb[208]
0xE3 C3	0x00	isrc2_pb_0_28	r	isrc2_pb[223]	isrc2_pb[222]	isrc2_pb[221]	isrc2_pb[220]	isrc2_pb[219]	isrc2_pb[218]	isrc2_pb[217]	isrc2_pb[216]
0xE3 C4	0x00	gamut_mdata_pb_0_1	r	gbd[7]	gbd[6]	gbd[5]	gbd[4]	gbd[3]	gbd[2]	gbd[1]	gbd[0]
0xE3 C5	0x00	gamut_mdata_pb_0_2	r	gbd[15]	gbd[14]	gbd[13]	gbd[12]	gbd[11]	gbd[10]	gbd[9]	gbd[8]
0xE3 C6	0x00	gamut_mdata_pb_0_3	r	gbd[23]	gbd[22]	gbd[21]	gbd[20]	gbd[19]	gbd[18]	gbd[17]	gbd[16]
0xE3 C7	0x00	gamut_mdata_pb_0_4	r	gbd[31]	gbd[30]	gbd[29]	gbd[28]	gbd[27]	gbd[26]	gbd[25]	gbd[24]
0xE3 C8	0x00	gamut_mdata_pb_0_5	r	gbd[39]	gbd[38]	gbd[37]	gbd[36]	gbd[35]	gbd[34]	gbd[33]	gbd[32]
0xE3 C9	0x00	gamut_mdata_pb_0_6	r	gbd[47]	gbd[46]	gbd[45]	gbd[44]	gbd[43]	gbd[42]	gbd[41]	gbd[40]
0xE3 CA	0x00	gamut_mdata_pb_0_7	r	gbd[55]	gbd[54]	gbd[53]	gbd[52]	gbd[51]	gbd[50]	gbd[49]	gbd[48]
0xE3 CB	0x00	gamut_mdata_pb_0_8	r	gbd[63]	gbd[62]	gbd[61]	gbd[60]	gbd[59]	gbd[58]	gbd[57]	gbd[56]
0xE3 CC	0x00	gamut_mdata_pb_0_9	r	gbd[71]	gbd[70]	gbd[69]	gbd[68]	gbd[67]	gbd[66]	gbd[65]	gbd[64]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE3 CD	0x00	gamut_mdata_pb _0_10	r	gbd[79]	gbd[78]	gbd[77]	gbd[76]	gbd[75]	gbd[74]	gbd[73]	gbd[72]
0xE3 CE	0x00	gamut_mdata_pb _0_11	r	gbd[87]	gbd[86]	gbd[85]	gbd[84]	gbd[83]	gbd[82]	gbd[81]	gbd[80]
0xE3 CF	0x00	gamut_mdata_pb _0_12	r	gbd[95]	gbd[94]	gbd[93]	gbd[92]	gbd[91]	gbd[90]	gbd[89]	gbd[88]
0xE3 D0	0x00	gamut_mdata_pb _0_13	r	gbd[103]	gbd[102]	gbd[101]	gbd[100]	gbd[99]	gbd[98]	gbd[97]	gbd[96]
0xE3 D1	0x00	gamut_mdata_pb _0_14	r	gbd[111]	gbd[110]	gbd[109]	gbd[108]	gbd[107]	gbd[106]	gbd[105]	gbd[104]
0xE3 D2	0x00	gamut_mdata_pb _0_15	r	gbd[119]	gbd[118]	gbd[117]	gbd[116]	gbd[115]	gbd[114]	gbd[113]	gbd[112]
0xE3 D3	0x00	gamut_mdata_pb _0_16	r	gbd[127]	gbd[126]	gbd[125]	gbd[124]	gbd[123]	gbd[122]	gbd[121]	gbd[120]
0xE3 D4	0x00	gamut_mdata_pb _0_17	r	gbd[135]	gbd[134]	gbd[133]	gbd[132]	gbd[131]	gbd[130]	gbd[129]	gbd[128]
0xE3 D5	0x00	gamut_mdata_pb _0_18	r	gbd[143]	gbd[142]	gbd[141]	gbd[140]	gbd[139]	gbd[138]	gbd[137]	gbd[136]
0xE3 D6	0x00	gamut_mdata_pb _0_19	r	gbd[151]	gbd[150]	gbd[149]	gbd[148]	gbd[147]	gbd[146]	gbd[145]	gbd[144]
0xE3 D7	0x00	gamut_mdata_pb _0_20	r	gbd[159]	gbd[158]	gbd[157]	gbd[156]	gbd[155]	gbd[154]	gbd[153]	gbd[152]
0xE3 D8	0x00	gamut_mdata_pb _0_21	r	gbd[167]	gbd[166]	gbd[165]	gbd[164]	gbd[163]	gbd[162]	gbd[161]	gbd[160]
0xE3 D9	0x00	gamut_mdata_pb _0_22	r	gbd[175]	gbd[174]	gbd[173]	gbd[172]	gbd[171]	gbd[170]	gbd[169]	gbd[168]
0xE3 DA	0x00	gamut_mdata_pb _0_23	r	gbd[183]	gbd[182]	gbd[181]	gbd[180]	gbd[179]	gbd[178]	gbd[177]	gbd[176]
0xE3 DB	0x00	gamut_mdata_pb _0_24	r	gbd[191]	gbd[190]	gbd[189]	gbd[188]	gbd[187]	gbd[186]	gbd[185]	gbd[184]
0xE3 DC	0x00	gamut_mdata_pb _0_25	r	gbd[199]	gbd[198]	gbd[197]	gbd[196]	gbd[195]	gbd[194]	gbd[193]	gbd[192]
0xE3 DD	0x00	gamut_mdata_pb _0_26	r	gbd[207]	gbd[206]	gbd[205]	gbd[204]	gbd[203]	gbd[202]	gbd[201]	gbd[200]
0xE3 DE	0x00	gamut_mdata_pb _0_27	r	gbd[215]	gbd[214]	gbd[213]	gbd[212]	gbd[211]	gbd[210]	gbd[209]	gbd[208]
0xE3 DF	0x00	gamut_mdata_pb _0_28	r	gbd[223]	gbd[222]	gbd[221]	gbd[220]	gbd[219]	gbd[218]	gbd[217]	gbd[216]
0xE3 E0	0x82	avi_packet_id	rw	avi_packet_id[7]	avi_packet_id[6]	avi_packet_id[5]	avi_packet_id[4]	avi_packet_id[3]	avi_packet_id[2]	avi_packet_id[1]	avi_packet_id[0]
0xE3 E1	0x00	avi_inf_vers	r	avi_inf_vers[7]	avi_inf_vers[6]	avi_inf_vers[5]	avi_inf_vers[4]	avi_inf_vers[3]	avi_inf_vers[2]	avi_inf_vers[1]	avi_inf_vers[0]
0xE3 E2	0x00	avi_inf_len	r	avi_inf_len[7]	avi_inf_len[6]	avi_inf_len[5]	avi_inf_len[4]	avi_inf_len[3]	avi_inf_len[2]	avi_inf_len[1]	avi_inf_len[0]
0xE3 E3	0x84	aud_packet_id	rw	aud_packet_id[7]	aud_packet_id[6]	aud_packet_id[5]	aud_packet_id[4]	aud_packet_id[3]	aud_packet_id[2]	aud_packet_id[1]	aud_packet_id[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE3 E4	0x00	aud_inf_vers	r	aud_inf_vers[7]	aud_inf_vers[6]	aud_inf_vers[5]	aud_inf_vers[4]	aud_inf_vers[3]	aud_inf_vers[2]	aud_inf_vers[1]	aud_inf_vers[0]
0xE3 E5	0x00	aud_inf_len	r	aud_inf_len[7]	aud_inf_len[6]	aud_inf_len[5]	aud_inf_len[4]	aud_inf_len[3]	aud_inf_len[2]	aud_inf_len[1]	aud_inf_len[0]
0xE3 E6	0x83	spd_packet_id	rw	spd_packet_id[7]	spd_packet_id[6]	spd_packet_id[5]	spd_packet_id[4]	spd_packet_id[3]	spd_packet_id[2]	spd_packet_id[1]	spd_packet_id[0]
0xE3 E7	0x00	spd_inf_vers	r	spd_inf_vers[7]	spd_inf_vers[6]	spd_inf_vers[5]	spd_inf_vers[4]	spd_inf_vers[3]	spd_inf_vers[2]	spd_inf_vers[1]	spd_inf_vers[0]
0xE3 E8	0x00	spd_inf_len	r	spd_inf_len[7]	spd_inf_len[6]	spd_inf_len[5]	spd_inf_len[4]	spd_inf_len[3]	spd_inf_len[2]	spd_inf_len[1]	spd_inf_len[0]
0xE3 E9	0x85	ms_packet_id	rw	ms_packet_id[7]	ms_packet_id[6]	ms_packet_id[5]	ms_packet_id[4]	ms_packet_id[3]	ms_packet_id[2]	ms_packet_id[1]	ms_packet_id[0]
0xE3 EA	0x00	ms_inf_vers	r	ms_inf_vers[7]	ms_inf_vers[6]	ms_inf_vers[5]	ms_inf_vers[4]	ms_inf_vers[3]	ms_inf_vers[2]	ms_inf_vers[1]	ms_inf_vers[0]
0xE3 EB	0x00	ms_inf_len	r	ms_inf_len[7]	ms_inf_len[6]	ms_inf_len[5]	ms_inf_len[4]	ms_inf_len[3]	ms_inf_len[2]	ms_inf_len[1]	ms_inf_len[0]
0xE3 EC	0x81	vs_packet_id	rw	vs_packet_id[7]	vs_packet_id[6]	vs_packet_id[5]	vs_packet_id[4]	vs_packet_id[3]	vs_packet_id[2]	vs_packet_id[1]	vs_packet_id[0]
0xE3 ED	0x00	vs_inf_vers	r	vs_inf_vers[7]	vs_inf_vers[6]	vs_inf_vers[5]	vs_inf_vers[4]	vs_inf_vers[3]	vs_inf_vers[2]	vs_inf_vers[1]	vs_inf_vers[0]
0xE3 EE	0x00	vs_inf_len	r	vs_inf_len[7]	vs_inf_len[6]	vs_inf_len[5]	vs_inf_len[4]	vs_inf_len[3]	vs_inf_len[2]	vs_inf_len[1]	vs_inf_len[0]
0xE3 EF	0x04	acp_packet_id	rw	acp_packet_id[7]	acp_packet_id[6]	acp_packet_id[5]	acp_packet_id[4]	acp_packet_id[3]	acp_packet_id[2]	acp_packet_id[1]	acp_packet_id[0]
0xE3 F0	0x00	acp_type	r	acp_type[7]	acp_type[6]	acp_type[5]	acp_type[4]	acp_type[3]	acp_type[2]	acp_type[1]	acp_type[0]
0xE3 F1	0x00	acp_header2	r	acp_header2[7]	acp_header2[6]	acp_header2[5]	acp_header2[4]	acp_header2[3]	acp_header2[2]	acp_header2[1]	acp_header2[0]
0xE3 F2	0x05	isrc1_packet_id	rw	isrc1_packet_id[7]	isrc1_packet_id[6]	isrc1_packet_id[5]	isrc1_packet_id[4]	isrc1_packet_id[3]	isrc1_packet_id[2]	isrc1_packet_id[1]	isrc1_packet_id[0]
0xE3 F3	0x00	isrc1_header1	r	isrc1_header1[7]	isrc1_header1[6]	isrc1_header1[5]	isrc1_header1[4]	isrc1_header1[3]	isrc1_header1[2]	isrc1_header1[1]	isrc1_header1[0]
0xE3 F4	0x00	isrc1_header2	r	isrc1_header2[7]	isrc1_header2[6]	isrc1_header2[5]	isrc1_header2[4]	isrc1_header2[3]	isrc1_header2[2]	isrc1_header2[1]	isrc1_header2[0]
0xE3 F5	0x06	isrc2_packet_id	rw	isrc2_packet_id[7]	isrc2_packet_id[6]	isrc2_packet_id[5]	isrc2_packet_id[4]	isrc2_packet_id[3]	isrc2_packet_id[2]	isrc2_packet_id[1]	isrc2_packet_id[0]
0xE3 F6	0x00	isrc2_header1	r	isrc2_header1[7]	isrc2_header1[6]	isrc2_header1[5]	isrc2_header1[4]	isrc2_header1[3]	isrc2_header1[2]	isrc2_header1[1]	isrc2_header1[0]
0xE3 F7	0x00	isrc2_header2	r	isrc2_header2[7]	isrc2_header2[6]	isrc2_header2[5]	isrc2_header2[4]	isrc2_header2[3]	isrc2_header2[2]	isrc2_header2[1]	isrc2_header2[0]
0xE3 F8	0x0A	gamut_packet_id	rw	gamut_packet_id[7]	gamut_packet_id[6]	gamut_packet_id[5]	gamut_packet_id[4]	gamut_packet_id[3]	gamut_packet_id[2]	gamut_packet_id[1]	gamut_packet_id[0]
0xE3 F9	0x00	gamut_header1	r	gamut_header1[7]	gamut_header1[6]	gamut_header1[5]	gamut_header1[4]	gamut_header1[3]	gamut_header1[2]	gamut_header1[1]	gamut_header1[0]
0xE3 FA	0x00	gamut_header2	r	gamut_header2[7]	gamut_header2[6]	gamut_header2[5]	gamut_header2[4]	gamut_header2[3]	gamut_header2[2]	gamut_header2[1]	gamut_header2[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE3 FD	0x81		rw	pkt_cnt_id[7]	pkt_cnt_id[6]	pkt_cnt_id[5]	pkt_cnt_id[4]	pkt_cnt_id[3]	pkt_cnt_id[2]	pkt_cnt_id[1]	pkt_cnt_id[0]
0xE3 FE	0x00		rw	-	-	-	en_pkt_cnt_sel	pkt_cnt_sel[3]	pkt_cnt_sel[2]	pkt_cnt_sel[1]	pkt_cnt_sel[0]
0xE3 FF	0x00		r	-	-	-	-	rb_pkt_cnt[3]	rb_pkt_cnt[2]	rb_pkt_cnt[1]	rb_pkt_cnt[0]

1.7 TX1 MAIN MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 01	0x00	main_reg_ec01	rw	-	-	-	-	n[19]	n[18]	n[17]	n[16]
0xEC 02	0x00	main_reg_ec02	rw	n[15]	n[14]	n[13]	n[12]	n[11]	n[10]	n[9]	n[8]
0xEC 03	0x00	main_reg_ec03	rw	n[7]	n[6]	n[5]	n[4]	n[3]	n[2]	n[1]	n[0]
0xEC 04	0x00	main_reg_ec04	r	spdif_sf[3]	spdif_sf[2]	spdif_sf[1]	spdif_sf[0]	cts_internal[19]	cts_internal[18]	cts_internal[17]	cts_internal[16]
0xEC 05	0x00	main_reg_ec05	r	cts_internal[15]	cts_internal[14]	cts_internal[13]	cts_internal[12]	cts_internal[11]	cts_internal[10]	cts_internal[9]	cts_internal[8]
0xEC 06	0x00	main_reg_ec06	r	cts_internal[7]	cts_internal[6]	cts_internal[5]	cts_internal[4]	cts_internal[3]	cts_internal[2]	cts_internal[1]	cts_internal[0]
0xEC 07	0x00	main_reg_ec07	rw	-	-	-	-	cts_manual[19]	cts_manual[18]	cts_manual[17]	cts_manual[16]
0xEC 08	0x00	main_reg_ec08	rw	cts_manual[15]	cts_manual[14]	cts_manual[13]	cts_manual[12]	cts_manual[11]	cts_manual[10]	cts_manual[9]	cts_manual[8]
0xEC 09	0x00	main_reg_ec09	rw	cts_manual[7]	cts_manual[6]	cts_manual[5]	cts_manual[4]	cts_manual[3]	cts_manual[2]	cts_manual[1]	cts_manual[0]
0xEC 0A	0x01	main_reg_ec0a	rw	cts_sel	audio_input_sel[2]	audio_input_sel[1]	audio_input_sel[0]	audio_mode[1]	audio_mode[0]	mclk_ratio[1]	mclk_ratio[0]
0xEC 0B	0x0E	main_reg_ec0b	rw	spdif_en	mclk_pol	mclk_en	-	-	-	-	rx_aud_packet_sel
0xEC 0C	0xBC	main_reg_ec0c	rw	audio_sampling_freq_sel	cs_bit_override	i2s_en[3]	i2s_en[2]	i2s_en[1]	i2s_en[0]	i2s_format[1]	i2s_format[0]
0xEC 0D	0x18	main_reg_ec0d	rw	-	-	-	i2s_bit_width[4]	i2s_bit_width[3]	i2s_bit_width[2]	i2s_bit_width[1]	i2s_bit_width[0]
0xEC 0E	0x01	main_reg_ec0e	rw	-	-	subpkt0_l_src[2]	subpkt0_l_src[1]	subpkt0_l_src[0]	subpkt0_r_src[2]	subpkt0_r_src[1]	subpkt0_r_src[0]
0xEC 0F	0x13	main_reg_ec0f	rw	-	-	subpkt1_l_src[2]	subpkt1_l_src[1]	subpkt1_l_src[0]	subpkt1_r_src[2]	subpkt1_r_src[1]	subpkt1_r_src[0]
0xEC 10	0x25	main_reg_ec10	rw	-	-	subpkt2_l_src[2]	subpkt2_l_src[1]	subpkt2_l_src[0]	subpkt2_r_src[2]	subpkt2_r_src[1]	subpkt2_r_src[0]
0xEC 11	0x37	main_reg_ec11	rw	-	-	subpkt3_l_src[2]	subpkt3_l_src[1]	subpkt3_l_src[0]	subpkt3_r_src[2]	subpkt3_r_src[1]	subpkt3_r_src[0]
0xEC 12	0x00	main_reg_ec12	rw	channel_status[1]	channel_status[0]	cr_bit	a_info[2]	a_info[1]	a_info[0]	clk_acc[1]	clk_acc[0]
0xEC 13	0x00	main_reg_ec13	rw	category_code[7]	category_code[6]	category_code[5]	category_code[4]	category_code[3]	category_code[2]	category_code[1]	category_code[0]
0xEC 14	0x00	main_reg_ec14	rw	source_number[3]	source_number[2]	source_number[1]	source_number[0]	word_length[3]	word_length[2]	word_length[1]	word_length[0]
0xEC 15	0x00	main_reg_ec15	rw	i2s_sf[3]	i2s_sf[2]	i2s_sf[1]	i2s_sf[0]	vfe_input_id[3]	vfe_input_id[2]	vfe_input_id[1]	vfe_input_id[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 16	0x00	main_reg_ec16	rw	vfe_output_format[1]	vfe_output_format[0]	-	-	-	-	-	vfe_input_cs
0xEC 17	0x00	main_reg_ec17	rw	-	-	-	-	-	gen_444_en	aspect_ratio	-
0xEC 18	0x46	main_reg_ec18	rw	csc_en	csc_scaling_factor[1]	csc_scaling_factor[0]	csc_a1[12]	csc_a1[11]	csc_a1[10]	csc_a1[9]	csc_a1[8]
0xEC 19	0x62	main_reg_ec19	rw	csc_a1[7]	csc_a1[6]	csc_a1[5]	csc_a1[4]	csc_a1[3]	csc_a1[2]	csc_a1[1]	csc_a1[0]
0xEC 1A	0x04	main_reg_ec1a	rw	-	-	-	csc_a2[12]	csc_a2[11]	csc_a2[10]	csc_a2[9]	csc_a2[8]
0xEC 1B	0xA8	main_reg_ec1b	rw	csc_a2[7]	csc_a2[6]	csc_a2[5]	csc_a2[4]	csc_a2[3]	csc_a2[2]	csc_a2[1]	csc_a2[0]
0xEC 1C	0x00	main_reg_ec1c	rw	-	-	-	csc_a3[12]	csc_a3[11]	csc_a3[10]	csc_a3[9]	csc_a3[8]
0xEC 1D	0x00	main_reg_ec1d	rw	csc_a3[7]	csc_a3[6]	csc_a3[5]	csc_a3[4]	csc_a3[3]	csc_a3[2]	csc_a3[1]	csc_a3[0]
0xEC 1E	0x1C	main_reg_ec1e	rw	-	-	-	csc_a4[12]	csc_a4[11]	csc_a4[10]	csc_a4[9]	csc_a4[8]
0xEC 1F	0x84	main_reg_ec1f	rw	csc_a4[7]	csc_a4[6]	csc_a4[5]	csc_a4[4]	csc_a4[3]	csc_a4[2]	csc_a4[1]	csc_a4[0]
0xEC 20	0x1C	main_reg_ec20	rw	-	-	-	csc_b1[12]	csc_b1[11]	csc_b1[10]	csc_b1[9]	csc_b1[8]
0xEC 21	0xBF	main_reg_ec21	rw	csc_b1[7]	csc_b1[6]	csc_b1[5]	csc_b1[4]	csc_b1[3]	csc_b1[2]	csc_b1[1]	csc_b1[0]
0xEC 22	0x04	main_reg_ec22	rw	-	-	-	csc_b2[12]	csc_b2[11]	csc_b2[10]	csc_b2[9]	csc_b2[8]
0xEC 23	0xAB	main_reg_ec23	rw	csc_b2[7]	csc_b2[6]	csc_b2[5]	csc_b2[4]	csc_b2[3]	csc_b2[2]	csc_b2[1]	csc_b2[0]
0xEC 24	0x1E	main_reg_ec24	rw	-	-	-	csc_b3[12]	csc_b3[11]	csc_b3[10]	csc_b3[9]	csc_b3[8]
0xEC 25	0x70	main_reg_ec25	rw	csc_b3[7]	csc_b3[6]	csc_b3[5]	csc_b3[4]	csc_b3[3]	csc_b3[2]	csc_b3[1]	csc_b3[0]
0xEC 26	0x02	main_reg_ec26	rw	-	-	-	csc_b4[12]	csc_b4[11]	csc_b4[10]	csc_b4[9]	csc_b4[8]
0xEC 27	0x1E	main_reg_ec27	rw	csc_b4[7]	csc_b4[6]	csc_b4[5]	csc_b4[4]	csc_b4[3]	csc_b4[2]	csc_b4[1]	csc_b4[0]
0xEC 28	0x00	main_reg_ec28	rw	-	-	-	csc_c1[12]	csc_c1[11]	csc_c1[10]	csc_c1[9]	csc_c1[8]
0xEC 29	0x00	main_reg_ec29	rw	csc_c1[7]	csc_c1[6]	csc_c1[5]	csc_c1[4]	csc_c1[3]	csc_c1[2]	csc_c1[1]	csc_c1[0]
0xEC 2A	0x04	main_reg_ec2a	rw	-	-	-	csc_c2[12]	csc_c2[11]	csc_c2[10]	csc_c2[9]	csc_c2[8]
0xEC 2B	0xA8	main_reg_ec2b	rw	csc_c2[7]	csc_c2[6]	csc_c2[5]	csc_c2[4]	csc_c2[3]	csc_c2[2]	csc_c2[1]	csc_c2[0]
0xEC 2C	0x08	main_reg_ec2c	rw	-	-	-	csc_c3[12]	csc_c3[11]	csc_c3[10]	csc_c3[9]	csc_c3[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 2D	0x12	main_reg_ec2d	rw	csc_c3[7]	csc_c3[6]	csc_c3[5]	csc_c3[4]	csc_c3[3]	csc_c3[2]	csc_c3[1]	csc_c3[0]
0xEC 2E	0x1B	main_reg_ec2e	rw	-	-	-	csc_c4[12]	csc_c4[11]	csc_c4[10]	csc_c4[9]	csc_c4[8]
0xEC 2F	0xAC	main_reg_ec2f	rw	csc_c4[7]	csc_c4[6]	csc_c4[5]	csc_c4[4]	csc_c4[3]	csc_c4[2]	csc_c4[1]	csc_c4[0]
0xEC 3B	0x80	main_reg_ec3b	rw	-	pr_mode[1]	pr_mode[0]	pr_pll_manual[1]	pr_pll_manual[0]	pr_value_manual[1]	pr_value_manual[0]	-
0xEC 3C	0x00	main_reg_ec3c	rw	-	-	vic_manual[5]	vic_manual[4]	vic_manual[3]	vic_manual[2]	vic_manual[1]	vic_manual[0]
0xEC 3D	0x00	main_reg_ec3d	r	pr_to_rx[1]	pr_to_rx[0]	vic_to_rx[5]	vic_to_rx[4]	vic_to_rx[3]	vic_to_rx[2]	vic_to_rx[1]	vic_to_rx[0]
0xEC 3E	0x00	main_reg_ec3e	r	vic_detected[5]	vic_detected[4]	vic_detected[3]	vic_detected[2]	vic_detected[1]	vic_detected[0]	-	-
0xEC 3F	0x00	main_reg_ec3f	r	aux_vic_detected[2]	aux_vic_detected[1]	aux_vic_detected[0]	progressive_mode_info[1]	progressive_mode_info[0]	-	-	-
0xEC 40	0x00	main_reg_ec40	rw	gc_pkt_en	spd_pkt_en	mpeg_pkt_en	acp_pkt_en	isrc_pkt_en	gm_pkt_en	spare_pkt1_en	spare_pkt0_en
0xEC 41	0x50	main_reg_ec41	rw	-	system_pd	-	-	-	-	-	-
0xEC 42	0x90	main_reg_ec42	r	-	hpd_state	rx_sense_state	-	i2s_32bit_mode	-	-	-
0xEC 44	0x79	main_reg_ec44	rw	-	n_cts_pkt_en	audio_sample_pkt_en	aviif_pkt_en	audioif_pkt_en	-	-	-
0xEC 45	0x70	main_reg_ec45	rw	packet_memory_address[7]	packet_memory_address[6]	packet_memory_address[5]	packet_memory_address[4]	packet_memory_address[3]	packet_memory_address[2]	packet_memory_address[1]	packet_memory_address[0]
0xEC 46	0x00	main_reg_ec46	rw	dsd_en[7]	dsd_en[6]	dsd_en[5]	dsd_en[4]	dsd_en[3]	dsd_en[2]	dsd_en[1]	dsd_en[0]
0xEC 47	0x01	main_reg_ec47	rw	dsd_mux_en	papb_sync	sample_invalid[3]	sample_invalid[2]	sample_invalid[1]	sample_invalid[0]	-	arc_eff_tran_en
0xEC 49	0x54	main_reg_ec49	rw	-	dither_mode[5]	dither_mode[4]	dither_mode[3]	dither_mode[2]	dither_mode[1]	dither_mode[0]	-
0xEC 4A	0x80	main_reg_ec4a	rw	auto_checksum_en	avi_update	audio_update	gcp_update	man_layout_en	man_layout_sel	-	-
0xEC 4B	0x00	main_reg_ec4b	rw	clear_avmute	set_avmute	-	-	-	-	-	-
0xEC 4C	0x00	main_reg_ec4c	rw	-	-	-	-	gc_cd[3]	gc_cd[2]	gc_cd[1]	gc_cd[0]
0xEC 4D	0x00	main_reg_ec4d	rw	gc_byte2[7]	gc_byte2[6]	gc_byte2[5]	gc_byte2[4]	gc_byte2[3]	gc_byte2[2]	gc_byte2[1]	gc_byte2[0]
0xEC 4E	0x00	main_reg_ec4e	rw	gc_byte3[7]	gc_byte3[6]	gc_byte3[5]	gc_byte3[4]	gc_byte3[3]	gc_byte3[2]	gc_byte3[1]	gc_byte3[0]
0xEC 4F	0x00	main_reg_ec4f	rw	gc_byte4[7]	gc_byte4[6]	gc_byte4[5]	gc_byte4[4]	gc_byte4[3]	gc_byte4[2]	gc_byte4[1]	gc_byte4[0]
0xEC 50	0x00	main_reg_ec50	rw	gc_byte5[7]	gc_byte5[6]	gc_byte5[5]	gc_byte5[4]	gc_byte5[3]	gc_byte5[2]	gc_byte5[1]	gc_byte5[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 51	0x00	main_reg_ec51	rw	gc_byte6[7]	gc_byte6[6]	gc_byte6[5]	gc_byte6[4]	gc_byte6[3]	gc_byte6[2]	gc_byte6[1]	gc_byte6[0]
0xEC 52	0x02	main_reg_ec52	rw	-	-	-	-	-	avi_version[2]	avi_version[1]	avi_version[0]
0xEC 53	0x0D	main_reg_ec53	rw	-	-	-	avi_length[4]	avi_length[3]	avi_length[2]	avi_length[1]	avi_length[0]
0xEC 54	0x00	main_reg_ec54	rw	avi_checksum[7]	avi_checksum[6]	avi_checksum[5]	avi_checksum[4]	avi_checksum[3]	avi_checksum[2]	avi_checksum[1]	avi_checksum[0]
0xEC 55	0x00	main_reg_ec55	rw	avi_byte1_7	y1y0[1]	y1y0[0]	a0	b1b0[1]	b1b0[0]	s1s0[1]	s1s0[0]
0xEC 56	0x00	main_reg_ec56	rw	c1c0[1]	c1c0[0]	m1m0[1]	m1m0[0]	r[3]	r[2]	r[1]	r[0]
0xEC 57	0x00	main_reg_ec57	rw	itc	ec[2]	ec[1]	ec[0]	q1q0[1]	q1q0[0]	sc[1]	sc[0]
0xEC 58	0x00	main_reg_ec58	rw	avi_byte4_7	-	-	-	-	-	-	-
0xEC 59	0x00	main_reg_ec59	rw	avi_byte5_7_4[3]	avi_byte5_7_4[2]	avi_byte5_7_4[1]	avi_byte5_7_4[0]	-	-	-	-
0xEC 5A	0x00	main_reg_ec5a	rw	avi_byte6[7]	avi_byte6[6]	avi_byte6[5]	avi_byte6[4]	avi_byte6[3]	avi_byte6[2]	avi_byte6[1]	avi_byte6[0]
0xEC 5B	0x00	main_reg_ec5b	rw	avi_byte7[7]	avi_byte7[6]	avi_byte7[5]	avi_byte7[4]	avi_byte7[3]	avi_byte7[2]	avi_byte7[1]	avi_byte7[0]
0xEC 5C	0x00	main_reg_ec5c	rw	avi_byte8[7]	avi_byte8[6]	avi_byte8[5]	avi_byte8[4]	avi_byte8[3]	avi_byte8[2]	avi_byte8[1]	avi_byte8[0]
0xEC 5D	0x00	main_reg_ec5d	rw	avi_byte9[7]	avi_byte9[6]	avi_byte9[5]	avi_byte9[4]	avi_byte9[3]	avi_byte9[2]	avi_byte9[1]	avi_byte9[0]
0xEC 5E	0x00	main_reg_ec5e	rw	avi_byte10[7]	avi_byte10[6]	avi_byte10[5]	avi_byte10[4]	avi_byte10[3]	avi_byte10[2]	avi_byte10[1]	avi_byte10[0]
0xEC 5F	0x00	main_reg_ec5f	rw	avi_byte11[7]	avi_byte11[6]	avi_byte11[5]	avi_byte11[4]	avi_byte11[3]	avi_byte11[2]	avi_byte11[1]	avi_byte11[0]
0xEC 60	0x00	main_reg_ec60	rw	avi_byte12[7]	avi_byte12[6]	avi_byte12[5]	avi_byte12[4]	avi_byte12[3]	avi_byte12[2]	avi_byte12[1]	avi_byte12[0]
0xEC 61	0x00	main_reg_ec61	rw	avi_byte13[7]	avi_byte13[6]	avi_byte13[5]	avi_byte13[4]	avi_byte13[3]	avi_byte13[2]	avi_byte13[1]	avi_byte13[0]
0xEC 62	0x00	main_reg_ec62	rw	avi_byte14[7]	avi_byte14[6]	avi_byte14[5]	avi_byte14[4]	avi_byte14[3]	avi_byte14[2]	avi_byte14[1]	avi_byte14[0]
0xEC 63	0x00	main_reg_ec63	rw	avi_byte15[7]	avi_byte15[6]	avi_byte15[5]	avi_byte15[4]	avi_byte15[3]	avi_byte15[2]	avi_byte15[1]	avi_byte15[0]
0xEC 64	0x00	main_reg_ec64	rw	avi_byte16[7]	avi_byte16[6]	avi_byte16[5]	avi_byte16[4]	avi_byte16[3]	avi_byte16[2]	avi_byte16[1]	avi_byte16[0]
0xEC 65	0x00	main_reg_ec65	rw	avi_byte17[7]	avi_byte17[6]	avi_byte17[5]	avi_byte17[4]	avi_byte17[3]	avi_byte17[2]	avi_byte17[1]	avi_byte17[0]
0xEC 66	0x00	main_reg_ec66	rw	avi_byte18[7]	avi_byte18[6]	avi_byte18[5]	avi_byte18[4]	avi_byte18[3]	avi_byte18[2]	avi_byte18[1]	avi_byte18[0]
0xEC 67	0x00	main_reg_ec67	rw	avi_byte19[7]	avi_byte19[6]	avi_byte19[5]	avi_byte19[4]	avi_byte19[3]	avi_byte19[2]	avi_byte19[1]	avi_byte19[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 68	0x00	main_reg_ec68	rw	avi_byte20[7]	avi_byte20[6]	avi_byte20[5]	avi_byte20[4]	avi_byte20[3]	avi_byte20[2]	avi_byte20[1]	avi_byte20[0]
0xEC 69	0x00	main_reg_ec69	rw	avi_byte21[7]	avi_byte21[6]	avi_byte21[5]	avi_byte21[4]	avi_byte21[3]	avi_byte21[2]	avi_byte21[1]	avi_byte21[0]
0xEC 6A	0x00	main_reg_ec6a	rw	avi_byte22[7]	avi_byte22[6]	avi_byte22[5]	avi_byte22[4]	avi_byte22[3]	avi_byte22[2]	avi_byte22[1]	avi_byte22[0]
0xEC 6B	0x00	main_reg_ec6b	rw	avi_byte23[7]	avi_byte23[6]	avi_byte23[5]	avi_byte23[4]	avi_byte23[3]	avi_byte23[2]	avi_byte23[1]	avi_byte23[0]
0xEC 6C	0x00	main_reg_ec6c	rw	avi_byte24[7]	avi_byte24[6]	avi_byte24[5]	avi_byte24[4]	avi_byte24[3]	avi_byte24[2]	avi_byte24[1]	avi_byte24[0]
0xEC 6D	0x00	main_reg_ec6d	rw	avi_byte25[7]	avi_byte25[6]	avi_byte25[5]	avi_byte25[4]	avi_byte25[3]	avi_byte25[2]	avi_byte25[1]	avi_byte25[0]
0xEC 6E	0x00	main_reg_ec6e	rw	avi_byte26[7]	avi_byte26[6]	avi_byte26[5]	avi_byte26[4]	avi_byte26[3]	avi_byte26[2]	avi_byte26[1]	avi_byte26[0]
0xEC 6F	0x00	main_reg_ec6f	rw	avi_byte27[7]	avi_byte27[6]	avi_byte27[5]	avi_byte27[4]	avi_byte27[3]	avi_byte27[2]	avi_byte27[1]	avi_byte27[0]
0xEC 70	0x01	main_reg_ec70	rw	-	-	-	-	-	audioif_version[2]	audioif_version[1]	audioif_version[0]
0xEC 71	0x0A	main_reg_ec71	rw	-	-	-	audioif_length[4]	audioif_length[3]	audioif_length[2]	audioif_length[1]	audioif_length[0]
0xEC 72	0x00	main_reg_ec72	rw	audioif_checksum [7]	audioif_checksum [6]	audioif_checksum [5]	audioif_checksum [4]	audioif_checksum [3]	audioif_checksum [2]	audioif_checksum [1]	audioif_checksum [0]
0xEC 73	0x00	main_reg_ec73	rw	audioif_ct[3]	audioif_ct[2]	audioif_ct[1]	audioif_ct[0]	audioif_byte1_3	audioif_cc[2]	audioif_cc[1]	audioif_cc[0]
0xEC 74	0x00	main_reg_ec74	rw	audioif_byte2_7_ 5[2]	audioif_byte2_7_ 5[1]	audioif_byte2_7_ 5[0]	audioif_sf[2]	audioif_sf[1]	audioif_sf[0]	audioif_ss[1]	audioif_ss[0]
0xEC 75	0x00	main_reg_ec75	rw	audioif_byte3[7]	audioif_byte3[6]	audioif_byte3[5]	audioif_byte3[4]	audioif_byte3[3]	audioif_byte3[2]	audioif_byte3[1]	audioif_byte3[0]
0xEC 76	0x00	main_reg_ec76	rw	audioif_ca[7]	audioif_ca[6]	audioif_ca[5]	audioif_ca[4]	audioif_ca[3]	audioif_ca[2]	audioif_ca[1]	audioif_ca[0]
0xEC 77	0x00	main_reg_ec77	rw	audioif_dm_inh	audioif_lsv[3]	audioif_lsv[2]	audioif_lsv[1]	audioif_lsv[0]	audioif_byte5_2_ 0[2]	audioif_byte5_2_ 0[1]	audioif_byte5_2_ 0[0]
0xEC 78	0x00	main_reg_ec78	rw	audioif_byte6[7]	audioif_byte6[6]	audioif_byte6[5]	audioif_byte6[4]	audioif_byte6[3]	audioif_byte6[2]	audioif_byte6[1]	audioif_byte6[0]
0xEC 79	0x00	main_reg_ec79	rw	audioif_byte7[7]	audioif_byte7[6]	audioif_byte7[5]	audioif_byte7[4]	audioif_byte7[3]	audioif_byte7[2]	audioif_byte7[1]	audioif_byte7[0]
0xEC 7A	0x00	main_reg_ec7a	rw	audioif_byte8[7]	audioif_byte8[6]	audioif_byte8[5]	audioif_byte8[4]	audioif_byte8[3]	audioif_byte8[2]	audioif_byte8[1]	audioif_byte8[0]
0xEC 7B	0x00	main_reg_ec7b	rw	audioif_byte9[7]	audioif_byte9[6]	audioif_byte9[5]	audioif_byte9[4]	audioif_byte9[3]	audioif_byte9[2]	audioif_byte9[1]	audioif_byte9[0]
0xEC 7C	0x00	main_reg_ec7c	rw	audioif_byte10[7]	audioif_byte10[6]	audioif_byte10[5]	audioif_byte10[4]	audioif_byte10[3]	audioif_byte10[2]	audioif_byte10[1]	audioif_byte10[0]
0xEC 80	0x7F	main_reg_ec80	rw	-	-	-	-	pre_en_ch0	pre_en_ch1	pre_en_ch2	pre_en_clk
0xEC 81	0x88	main_reg_ec81	rw	chg_inj_ch0[3]	chg_inj_ch0[2]	chg_inj_ch0[1]	chg_inj_ch0[0]	chg_inj_ch1[3]	chg_inj_ch1[2]	chg_inj_ch1[1]	chg_inj_ch1[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC 82	0x88	main_reg_ec82	rw	chg_inj_ch2[3]	chg_inj_ch2[2]	chg_inj_ch2[1]	chg_inj_ch2[0]	chg_inj_clk[3]	chg_inj_clk[2]	chg_inj_clk[1]	chg_inj_clk[0]
0xEC 83	0x00	main_reg_ec83	rw	ch0_sterm_disable	-	-	-	-	-	-	-
0xEC 84	0x00	main_reg_ec84	rw	ch1_sterm_disable	-	-	-	-	-	-	-
0xEC 85	0x00	main_reg_ec85	rw	ch2_sterm_disable	-	-	-	-	-	-	-
0xEC 86	0x00	main_reg_ec86	rw	clk_sterm_disable	-	-	-	-	-	-	-
0xEC 94	0xC0	main_reg_ec94	rw	hpd_int_en	rx_sense_int_en	vsync_int_en	-	-	edid_ready_int_en	hdcp_authenticated_int_en	ri_ready_int_en
0xEC 95	0x00	main_reg_ec95	rw	hdcp_error_int_en	bksv_flag_int_en	-	-	-	-	-	-
0xEC 96	0x00	main_reg_ec96	rw	hpd_int	rx_sense_int	vsync_int	-	-	edid_ready_int	hdcp_authenticated_int	ri_ready_int
0xEC 97	0x00	main_reg_ec97	rw	hdcp_error_int	bksv_flag_int	-	cec_tx_arbitration_lost_int	cec_tx_retry_timeout_int	-	-	-
0xEC 98	0x00	main_reg_ec98	rw	-	-	-	cec_pd	-	-	-	-
0xEC 9E	0x00	main_reg_ec9e	rw	high_freq_video[1]	high_freq_video[0]	video_offset_ctrl[1]	video_offset_ctrl[0]	-	-	-	-
0xEC 9F	0x00	main_reg_ec9f	rw	-	-	hpd_override[1]	hpd_override[0]	-	-	-	-
0xEC AB	0x20	main_reg_ecab	rw	-	-	hdcp_start_delay[2]	hdcp_start_delay[1]	hdcp_start_delay[0]	-	-	-
0xEC AE	0x00	main_reg_ecae	rw	-	hdcp_1p1_dis	r0_wait_time[1]	r0_wait_time[0]	hdcp_repeater_timeout[1]	hdcp_repeater_timeout[0]	-	-
0xEC AF	0x14	main_reg_ecaf	rw	hdcp_desired	-	-	frame_enc	-	hdmi_dvi_sel_en	hdmi_dvi_sel	-
0xEC B0	0x00	main_reg_ecb0	r	an[7]	an[6]	an[5]	an[4]	an[3]	an[2]	an[1]	an[0]
0xEC B1	0x00	main_reg_ecb1	r	an[15]	an[14]	an[13]	an[12]	an[11]	an[10]	an[9]	an[8]
0xEC B2	0x00	main_reg_ecb2	r	an[23]	an[22]	an[21]	an[20]	an[19]	an[18]	an[17]	an[16]
0xEC B3	0x00	main_reg_ecb3	r	an[31]	an[30]	an[29]	an[28]	an[27]	an[26]	an[25]	an[24]
0xEC B4	0x00	main_reg_ecb4	r	an[39]	an[38]	an[37]	an[36]	an[35]	an[34]	an[33]	an[32]
0xEC B5	0x00	main_reg_ecb5	r	an[47]	an[46]	an[45]	an[44]	an[43]	an[42]	an[41]	an[40]
0xEC B6	0x00	main_reg_ecb6	r	an[55]	an[54]	an[53]	an[52]	an[51]	an[50]	an[49]	an[48]
0xEC B7	0x00	main_reg_ecb7	r	an[63]	an[62]	an[61]	an[60]	an[59]	an[58]	an[57]	an[56]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEC B8	0x00	main_reg_ecb8	r	-	enc_on	-	keys_read_error	-	-	-	-
0xEC BE	0x00	main_reg_ecbe	r	bcaps[7]	bcaps[6]	bcaps[5]	bcaps[4]	bcaps[3]	bcaps[2]	bcaps[1]	bcaps[0]
0xEC BF	0x00	main_reg_ecbf	r	bksv[7]	bksv[6]	bksv[5]	bksv[4]	bksv[3]	bksv[2]	bksv[1]	bksv[0]
0xEC C0	0x00	main_reg_ecc0	r	bksv[15]	bksv[14]	bksv[13]	bksv[12]	bksv[11]	bksv[10]	bksv[9]	bksv[8]
0xEC C1	0x00	main_reg_ecc1	r	bksv[23]	bksv[22]	bksv[21]	bksv[20]	bksv[19]	bksv[18]	bksv[17]	bksv[16]
0xEC C2	0x00	main_reg_ecc2	r	bksv[31]	bksv[30]	bksv[29]	bksv[28]	bksv[27]	bksv[26]	bksv[25]	bksv[24]
0xEC C3	0x00	main_reg_ecc3	r	bksv[39]	bksv[38]	bksv[37]	bksv[36]	bksv[35]	bksv[34]	bksv[33]	bksv[32]
0xEC C4	0x00	main_reg_ecc4	rw	edid_segment[7]	edid_segment[6]	edid_segment[5]	edid_segment[4]	edid_segment[3]	edid_segment[2]	edid_segment[1]	edid_segment[0]
0xEC C5	0x00	main_reg_ecc5	r	-	an_stop	-	-	-	ri_flag	bksv_update_flag	pj_flag
0xEC C6	0x00	main_reg_ecc6	r	-	-	-	hdmi_mode	hdcp_requested	rx_sense	-	tmds_output_en
0xEC C7	0x00	main_reg_ecc7	rw	bksv_flag	bksv_count[6]	bksv_count[5]	bksv_count[4]	bksv_count[3]	bksv_count[2]	bksv_count[1]	bksv_count[0]
0xEC C8	0x00	main_reg_ecc8	r	hdcp_controller_error[3]	hdcp_controller_error[2]	hdcp_controller_error[1]	hdcp_controller_error[0]	hdcp_controller_state[3]	hdcp_controller_state[2]	hdcp_controller_state[1]	hdcp_controller_state[0]
0xEC C9	0x03	main_reg_ecc9	rw	-	-	-	edid_reread	edid_tries[3]	edid_tries[2]	edid_tries[1]	edid_tries[0]
0xEC CA	0x00	main_reg_ecca	r	hdcp_bstatus[15]	hdcp_bstatus[14]	hdcp_bstatus[13]	hdcp_bstatus[12]	hdcp_bstatus[11]	hdcp_bstatus[10]	hdcp_bstatus[9]	hdcp_bstatus[8]
0xEC CB	0x00	main_reg_eccb	r	hdcp_bstatus[7]	hdcp_bstatus[6]	hdcp_bstatus[5]	hdcp_bstatus[4]	hdcp_bstatus[3]	hdcp_bstatus[2]	hdcp_bstatus[1]	hdcp_bstatus[0]
0xEC E4	0x00	main_reg_ece4	r	pll_lock_status	-	-	-	-	-	-	-
0xEC E6	0x00	main_reg_ece6	rw	-	-	rx_sense_pd	-	-	-	-	-
0xEC EA	0x84	main_reg_ecea	rw	-	-	-	-	-	-	tmds_clk_invert	cci_controls

1.8 TX1 PACKET MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF200	0x00	packetmemory_reg_f200	rw	spd_header_byte_0[7]	spd_header_byte_0[6]	spd_header_byte_0[5]	spd_header_byte_0[4]	spd_header_byte_0[3]	spd_header_byte_0[2]	spd_header_byte_0[1]	spd_header_byte_0[0]
0xF201	0x00	packetmemory_reg_f201	rw	spd_header_byte_1[7]	spd_header_byte_1[6]	spd_header_byte_1[5]	spd_header_byte_1[4]	spd_header_byte_1[3]	spd_header_byte_1[2]	spd_header_byte_1[1]	spd_header_byte_1[0]
0xF202	0x00	packetmemory_reg_f202	rw	spd_header_byte_2[7]	spd_header_byte_2[6]	spd_header_byte_2[5]	spd_header_byte_2[4]	spd_header_byte_2[3]	spd_header_byte_2[2]	spd_header_byte_2[1]	spd_header_byte_2[0]
0xF203	0x00	packetmemory_reg_f203	rw	spd_packet_byte_0[7]	spd_packet_byte_0[6]	spd_packet_byte_0[5]	spd_packet_byte_0[4]	spd_packet_byte_0[3]	spd_packet_byte_0[2]	spd_packet_byte_0[1]	spd_packet_byte_0[0]
0xF204	0x00	packetmemory_reg_f204	rw	spd_packet_byte_1[7]	spd_packet_byte_1[6]	spd_packet_byte_1[5]	spd_packet_byte_1[4]	spd_packet_byte_1[3]	spd_packet_byte_1[2]	spd_packet_byte_1[1]	spd_packet_byte_1[0]
0xF205	0x00	packetmemory_reg_f205	rw	spd_packet_byte_2[7]	spd_packet_byte_2[6]	spd_packet_byte_2[5]	spd_packet_byte_2[4]	spd_packet_byte_2[3]	spd_packet_byte_2[2]	spd_packet_byte_2[1]	spd_packet_byte_2[0]
0xF206	0x00	packetmemory_reg_f206	rw	spd_packet_byte_3[7]	spd_packet_byte_3[6]	spd_packet_byte_3[5]	spd_packet_byte_3[4]	spd_packet_byte_3[3]	spd_packet_byte_3[2]	spd_packet_byte_3[1]	spd_packet_byte_3[0]
0xF207	0x00	packetmemory_reg_f207	rw	spd_packet_byte_4[7]	spd_packet_byte_4[6]	spd_packet_byte_4[5]	spd_packet_byte_4[4]	spd_packet_byte_4[3]	spd_packet_byte_4[2]	spd_packet_byte_4[1]	spd_packet_byte_4[0]
0xF208	0x00	packetmemory_reg_f208	rw	spd_packet_byte_5[7]	spd_packet_byte_5[6]	spd_packet_byte_5[5]	spd_packet_byte_5[4]	spd_packet_byte_5[3]	spd_packet_byte_5[2]	spd_packet_byte_5[1]	spd_packet_byte_5[0]
0xF209	0x00	packetmemory_reg_f209	rw	spd_packet_byte_6[7]	spd_packet_byte_6[6]	spd_packet_byte_6[5]	spd_packet_byte_6[4]	spd_packet_byte_6[3]	spd_packet_byte_6[2]	spd_packet_byte_6[1]	spd_packet_byte_6[0]
0xF20A	0x00	packetmemory_reg_f20a	rw	spd_packet_byte_7[7]	spd_packet_byte_7[6]	spd_packet_byte_7[5]	spd_packet_byte_7[4]	spd_packet_byte_7[3]	spd_packet_byte_7[2]	spd_packet_byte_7[1]	spd_packet_byte_7[0]
0xF20B	0x00	packetmemory_reg_f20b	rw	spd_packet_byte_8[7]	spd_packet_byte_8[6]	spd_packet_byte_8[5]	spd_packet_byte_8[4]	spd_packet_byte_8[3]	spd_packet_byte_8[2]	spd_packet_byte_8[1]	spd_packet_byte_8[0]
0xF20C	0x00	packetmemory_reg_f20c	rw	spd_packet_byte_9[7]	spd_packet_byte_9[6]	spd_packet_byte_9[5]	spd_packet_byte_9[4]	spd_packet_byte_9[3]	spd_packet_byte_9[2]	spd_packet_byte_9[1]	spd_packet_byte_9[0]
0xF20D	0x00	packetmemory_reg_f20d	rw	spd_packet_byte_10[7]	spd_packet_byte_10[6]	spd_packet_byte_10[5]	spd_packet_byte_10[4]	spd_packet_byte_10[3]	spd_packet_byte_10[2]	spd_packet_byte_10[1]	spd_packet_byte_10[0]
0xF20E	0x00	packetmemory_reg_f20e	rw	spd_packet_byte_11[7]	spd_packet_byte_11[6]	spd_packet_byte_11[5]	spd_packet_byte_11[4]	spd_packet_byte_11[3]	spd_packet_byte_11[2]	spd_packet_byte_11[1]	spd_packet_byte_11[0]
0xF20F	0x00	packetmemory_reg_f20f	rw	spd_packet_byte_12[7]	spd_packet_byte_12[6]	spd_packet_byte_12[5]	spd_packet_byte_12[4]	spd_packet_byte_12[3]	spd_packet_byte_12[2]	spd_packet_byte_12[1]	spd_packet_byte_12[0]
0xF210	0x00	packetmemory_reg_f210	rw	spd_packet_byte_13[7]	spd_packet_byte_13[6]	spd_packet_byte_13[5]	spd_packet_byte_13[4]	spd_packet_byte_13[3]	spd_packet_byte_13[2]	spd_packet_byte_13[1]	spd_packet_byte_13[0]
0xF211	0x00	packetmemory_reg_f211	rw	spd_packet_byte_14[7]	spd_packet_byte_14[6]	spd_packet_byte_14[5]	spd_packet_byte_14[4]	spd_packet_byte_14[3]	spd_packet_byte_14[2]	spd_packet_byte_14[1]	spd_packet_byte_14[0]
0xF212	0x00	packetmemory_reg_f212	rw	spd_packet_byte_15[7]	spd_packet_byte_15[6]	spd_packet_byte_15[5]	spd_packet_byte_15[4]	spd_packet_byte_15[3]	spd_packet_byte_15[2]	spd_packet_byte_15[1]	spd_packet_byte_15[0]
0xF213	0x00	packetmemory_reg_f213	rw	spd_packet_byte_16[7]	spd_packet_byte_16[6]	spd_packet_byte_16[5]	spd_packet_byte_16[4]	spd_packet_byte_16[3]	spd_packet_byte_16[2]	spd_packet_byte_16[1]	spd_packet_byte_16[0]
0xF214	0x00	packetmemory_reg_f214	rw	spd_packet_byte_17[7]	spd_packet_byte_17[6]	spd_packet_byte_17[5]	spd_packet_byte_17[4]	spd_packet_byte_17[3]	spd_packet_byte_17[2]	spd_packet_byte_17[1]	spd_packet_byte_17[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF215	0x00	packetmemory_reg_f215	rw	spd_packet_byte_18[7]	spd_packet_byte_18[6]	spd_packet_byte_18[5]	spd_packet_byte_18[4]	spd_packet_byte_18[3]	spd_packet_byte_18[2]	spd_packet_byte_18[1]	spd_packet_byte_18[0]
0xF216	0x00	packetmemory_reg_f216	rw	spd_packet_byte_19[7]	spd_packet_byte_19[6]	spd_packet_byte_19[5]	spd_packet_byte_19[4]	spd_packet_byte_19[3]	spd_packet_byte_19[2]	spd_packet_byte_19[1]	spd_packet_byte_19[0]
0xF217	0x00	packetmemory_reg_f217	rw	spd_packet_byte_20[7]	spd_packet_byte_20[6]	spd_packet_byte_20[5]	spd_packet_byte_20[4]	spd_packet_byte_20[3]	spd_packet_byte_20[2]	spd_packet_byte_20[1]	spd_packet_byte_20[0]
0xF218	0x00	packetmemory_reg_f218	rw	spd_packet_byte_21[7]	spd_packet_byte_21[6]	spd_packet_byte_21[5]	spd_packet_byte_21[4]	spd_packet_byte_21[3]	spd_packet_byte_21[2]	spd_packet_byte_21[1]	spd_packet_byte_21[0]
0xF219	0x00	packetmemory_reg_f219	rw	spd_packet_byte_22[7]	spd_packet_byte_22[6]	spd_packet_byte_22[5]	spd_packet_byte_22[4]	spd_packet_byte_22[3]	spd_packet_byte_22[2]	spd_packet_byte_22[1]	spd_packet_byte_22[0]
0xF21A	0x00	packetmemory_reg_f21a	rw	spd_packet_byte_23[7]	spd_packet_byte_23[6]	spd_packet_byte_23[5]	spd_packet_byte_23[4]	spd_packet_byte_23[3]	spd_packet_byte_23[2]	spd_packet_byte_23[1]	spd_packet_byte_23[0]
0xF21B	0x00	packetmemory_reg_f21b	rw	spd_packet_byte_24[7]	spd_packet_byte_24[6]	spd_packet_byte_24[5]	spd_packet_byte_24[4]	spd_packet_byte_24[3]	spd_packet_byte_24[2]	spd_packet_byte_24[1]	spd_packet_byte_24[0]
0xF21C	0x00	packetmemory_reg_f21c	rw	spd_packet_byte_25[7]	spd_packet_byte_25[6]	spd_packet_byte_25[5]	spd_packet_byte_25[4]	spd_packet_byte_25[3]	spd_packet_byte_25[2]	spd_packet_byte_25[1]	spd_packet_byte_25[0]
0xF21D	0x00	packetmemory_reg_f21d	rw	spd_packet_byte_26[7]	spd_packet_byte_26[6]	spd_packet_byte_26[5]	spd_packet_byte_26[4]	spd_packet_byte_26[3]	spd_packet_byte_26[2]	spd_packet_byte_26[1]	spd_packet_byte_26[0]
0xF21E	0x00	packetmemory_reg_f21e	rw	spd_packet_byte_27[7]	spd_packet_byte_27[6]	spd_packet_byte_27[5]	spd_packet_byte_27[4]	spd_packet_byte_27[3]	spd_packet_byte_27[2]	spd_packet_byte_27[1]	spd_packet_byte_27[0]
0xF21F	0x00	packetmemory_reg_f21f	rw	spd_update	-	-	-	-	-	-	-
0xF220	0x00	packetmemory_reg_f220	rw	mpeg_header_byte_0[7]	mpeg_header_byte_0[6]	mpeg_header_byte_0[5]	mpeg_header_byte_0[4]	mpeg_header_byte_0[3]	mpeg_header_byte_0[2]	mpeg_header_byte_0[1]	mpeg_header_byte_0[0]
0xF221	0x00	packetmemory_reg_f221	rw	mpeg_header_byte_1[7]	mpeg_header_byte_1[6]	mpeg_header_byte_1[5]	mpeg_header_byte_1[4]	mpeg_header_byte_1[3]	mpeg_header_byte_1[2]	mpeg_header_byte_1[1]	mpeg_header_byte_1[0]
0xF222	0x00	packetmemory_reg_f222	rw	mpeg_header_byte_2[7]	mpeg_header_byte_2[6]	mpeg_header_byte_2[5]	mpeg_header_byte_2[4]	mpeg_header_byte_2[3]	mpeg_header_byte_2[2]	mpeg_header_byte_2[1]	mpeg_header_byte_2[0]
0xF223	0x00	packetmemory_reg_f223	rw	mpeg_packet_byte_0[7]	mpeg_packet_byte_0[6]	mpeg_packet_byte_0[5]	mpeg_packet_byte_0[4]	mpeg_packet_byte_0[3]	mpeg_packet_byte_0[2]	mpeg_packet_byte_0[1]	mpeg_packet_byte_0[0]
0xF224	0x00	packetmemory_reg_f224	rw	mpeg_packet_byte_1[7]	mpeg_packet_byte_1[6]	mpeg_packet_byte_1[5]	mpeg_packet_byte_1[4]	mpeg_packet_byte_1[3]	mpeg_packet_byte_1[2]	mpeg_packet_byte_1[1]	mpeg_packet_byte_1[0]
0xF225	0x00	packetmemory_reg_f225	rw	mpeg_packet_byte_2[7]	mpeg_packet_byte_2[6]	mpeg_packet_byte_2[5]	mpeg_packet_byte_2[4]	mpeg_packet_byte_2[3]	mpeg_packet_byte_2[2]	mpeg_packet_byte_2[1]	mpeg_packet_byte_2[0]
0xF226	0x00	packetmemory_reg_f226	rw	mpeg_packet_byte_3[7]	mpeg_packet_byte_3[6]	mpeg_packet_byte_3[5]	mpeg_packet_byte_3[4]	mpeg_packet_byte_3[3]	mpeg_packet_byte_3[2]	mpeg_packet_byte_3[1]	mpeg_packet_byte_3[0]
0xF227	0x00	packetmemory_reg_f227	rw	mpeg_packet_byte_4[7]	mpeg_packet_byte_4[6]	mpeg_packet_byte_4[5]	mpeg_packet_byte_4[4]	mpeg_packet_byte_4[3]	mpeg_packet_byte_4[2]	mpeg_packet_byte_4[1]	mpeg_packet_byte_4[0]
0xF228	0x00	packetmemory_reg_f228	rw	mpeg_packet_byte_5[7]	mpeg_packet_byte_5[6]	mpeg_packet_byte_5[5]	mpeg_packet_byte_5[4]	mpeg_packet_byte_5[3]	mpeg_packet_byte_5[2]	mpeg_packet_byte_5[1]	mpeg_packet_byte_5[0]
0xF229	0x00	packetmemory_reg_f229	rw	mpeg_packet_byte_6[7]	mpeg_packet_byte_6[6]	mpeg_packet_byte_6[5]	mpeg_packet_byte_6[4]	mpeg_packet_byte_6[3]	mpeg_packet_byte_6[2]	mpeg_packet_byte_6[1]	mpeg_packet_byte_6[0]
0xF22A	0x00	packetmemory_reg_f22a	rw	mpeg_packet_byte_7[7]	mpeg_packet_byte_7[6]	mpeg_packet_byte_7[5]	mpeg_packet_byte_7[4]	mpeg_packet_byte_7[3]	mpeg_packet_byte_7[2]	mpeg_packet_byte_7[1]	mpeg_packet_byte_7[0]
0xF22B	0x00	packetmemory_reg_f22b	rw	mpeg_packet_byte_8[7]	mpeg_packet_byte_8[6]	mpeg_packet_byte_8[5]	mpeg_packet_byte_8[4]	mpeg_packet_byte_8[3]	mpeg_packet_byte_8[2]	mpeg_packet_byte_8[1]	mpeg_packet_byte_8[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2 2C	0x00	packetmemory_reg_f22c	rw	mpeg_packet_byte_9[7]	mpeg_packet_byte_9[6]	mpeg_packet_byte_9[5]	mpeg_packet_byte_9[4]	mpeg_packet_byte_9[3]	mpeg_packet_byte_9[2]	mpeg_packet_byte_9[1]	mpeg_packet_byte_9[0]
0xF2 2D	0x00	packetmemory_reg_f22d	rw	mpeg_packet_byte_10[7]	mpeg_packet_byte_10[6]	mpeg_packet_byte_10[5]	mpeg_packet_byte_10[4]	mpeg_packet_byte_10[3]	mpeg_packet_byte_10[2]	mpeg_packet_byte_10[1]	mpeg_packet_byte_10[0]
0xF2 2E	0x00	packetmemory_reg_f22e	rw	mpeg_packet_byte_11[7]	mpeg_packet_byte_11[6]	mpeg_packet_byte_11[5]	mpeg_packet_byte_11[4]	mpeg_packet_byte_11[3]	mpeg_packet_byte_11[2]	mpeg_packet_byte_11[1]	mpeg_packet_byte_11[0]
0xF2 2F	0x00	packetmemory_reg_f22f	rw	mpeg_packet_byte_12[7]	mpeg_packet_byte_12[6]	mpeg_packet_byte_12[5]	mpeg_packet_byte_12[4]	mpeg_packet_byte_12[3]	mpeg_packet_byte_12[2]	mpeg_packet_byte_12[1]	mpeg_packet_byte_12[0]
0xF2 30	0x00	packetmemory_reg_f230	rw	mpeg_packet_byte_13[7]	mpeg_packet_byte_13[6]	mpeg_packet_byte_13[5]	mpeg_packet_byte_13[4]	mpeg_packet_byte_13[3]	mpeg_packet_byte_13[2]	mpeg_packet_byte_13[1]	mpeg_packet_byte_13[0]
0xF2 31	0x00	packetmemory_reg_f231	rw	mpeg_packet_byte_14[7]	mpeg_packet_byte_14[6]	mpeg_packet_byte_14[5]	mpeg_packet_byte_14[4]	mpeg_packet_byte_14[3]	mpeg_packet_byte_14[2]	mpeg_packet_byte_14[1]	mpeg_packet_byte_14[0]
0xF2 32	0x00	packetmemory_reg_f232	rw	mpeg_packet_byte_15[7]	mpeg_packet_byte_15[6]	mpeg_packet_byte_15[5]	mpeg_packet_byte_15[4]	mpeg_packet_byte_15[3]	mpeg_packet_byte_15[2]	mpeg_packet_byte_15[1]	mpeg_packet_byte_15[0]
0xF2 33	0x00	packetmemory_reg_f233	rw	mpeg_packet_byte_16[7]	mpeg_packet_byte_16[6]	mpeg_packet_byte_16[5]	mpeg_packet_byte_16[4]	mpeg_packet_byte_16[3]	mpeg_packet_byte_16[2]	mpeg_packet_byte_16[1]	mpeg_packet_byte_16[0]
0xF2 34	0x00	packetmemory_reg_f234	rw	mpeg_packet_byte_17[7]	mpeg_packet_byte_17[6]	mpeg_packet_byte_17[5]	mpeg_packet_byte_17[4]	mpeg_packet_byte_17[3]	mpeg_packet_byte_17[2]	mpeg_packet_byte_17[1]	mpeg_packet_byte_17[0]
0xF2 35	0x00	packetmemory_reg_f235	rw	mpeg_packet_byte_18[7]	mpeg_packet_byte_18[6]	mpeg_packet_byte_18[5]	mpeg_packet_byte_18[4]	mpeg_packet_byte_18[3]	mpeg_packet_byte_18[2]	mpeg_packet_byte_18[1]	mpeg_packet_byte_18[0]
0xF2 36	0x00	packetmemory_reg_f236	rw	mpeg_packet_byte_19[7]	mpeg_packet_byte_19[6]	mpeg_packet_byte_19[5]	mpeg_packet_byte_19[4]	mpeg_packet_byte_19[3]	mpeg_packet_byte_19[2]	mpeg_packet_byte_19[1]	mpeg_packet_byte_19[0]
0xF2 37	0x00	packetmemory_reg_f237	rw	mpeg_packet_byte_20[7]	mpeg_packet_byte_20[6]	mpeg_packet_byte_20[5]	mpeg_packet_byte_20[4]	mpeg_packet_byte_20[3]	mpeg_packet_byte_20[2]	mpeg_packet_byte_20[1]	mpeg_packet_byte_20[0]
0xF2 38	0x00	packetmemory_reg_f238	rw	mpeg_packet_byte_21[7]	mpeg_packet_byte_21[6]	mpeg_packet_byte_21[5]	mpeg_packet_byte_21[4]	mpeg_packet_byte_21[3]	mpeg_packet_byte_21[2]	mpeg_packet_byte_21[1]	mpeg_packet_byte_21[0]
0xF2 39	0x00	packetmemory_reg_f239	rw	mpeg_packet_byte_22[7]	mpeg_packet_byte_22[6]	mpeg_packet_byte_22[5]	mpeg_packet_byte_22[4]	mpeg_packet_byte_22[3]	mpeg_packet_byte_22[2]	mpeg_packet_byte_22[1]	mpeg_packet_byte_22[0]
0xF2 3A	0x00	packetmemory_reg_f23a	rw	mpeg_packet_byte_23[7]	mpeg_packet_byte_23[6]	mpeg_packet_byte_23[5]	mpeg_packet_byte_23[4]	mpeg_packet_byte_23[3]	mpeg_packet_byte_23[2]	mpeg_packet_byte_23[1]	mpeg_packet_byte_23[0]
0xF2 3B	0x00	packetmemory_reg_f23b	rw	mpeg_packet_byte_24[7]	mpeg_packet_byte_24[6]	mpeg_packet_byte_24[5]	mpeg_packet_byte_24[4]	mpeg_packet_byte_24[3]	mpeg_packet_byte_24[2]	mpeg_packet_byte_24[1]	mpeg_packet_byte_24[0]
0xF2 3C	0x00	packetmemory_reg_f23c	rw	mpeg_packet_byte_25[7]	mpeg_packet_byte_25[6]	mpeg_packet_byte_25[5]	mpeg_packet_byte_25[4]	mpeg_packet_byte_25[3]	mpeg_packet_byte_25[2]	mpeg_packet_byte_25[1]	mpeg_packet_byte_25[0]
0xF2 3D	0x00	packetmemory_reg_f23d	rw	mpeg_packet_byte_26[7]	mpeg_packet_byte_26[6]	mpeg_packet_byte_26[5]	mpeg_packet_byte_26[4]	mpeg_packet_byte_26[3]	mpeg_packet_byte_26[2]	mpeg_packet_byte_26[1]	mpeg_packet_byte_26[0]
0xF2 3E	0x00	packetmemory_reg_f23e	rw	mpeg_packet_byte_27[7]	mpeg_packet_byte_27[6]	mpeg_packet_byte_27[5]	mpeg_packet_byte_27[4]	mpeg_packet_byte_27[3]	mpeg_packet_byte_27[2]	mpeg_packet_byte_27[1]	mpeg_packet_byte_27[0]
0xF2 3F	0x00	packetmemory_reg_f23f	rw	mpeg_update	-	-	-	-	-	-	-
0xF2 40	0x00	packetmemory_reg_f240	rw	acp_header_byte_0[7]	acp_header_byte_0[6]	acp_header_byte_0[5]	acp_header_byte_0[4]	acp_header_byte_0[3]	acp_header_byte_0[2]	acp_header_byte_0[1]	acp_header_byte_0[0]
0xF2 41	0x00	packetmemory_reg_f241	rw	acp_header_byte_1[7]	acp_header_byte_1[6]	acp_header_byte_1[5]	acp_header_byte_1[4]	acp_header_byte_1[3]	acp_header_byte_1[2]	acp_header_byte_1[1]	acp_header_byte_1[0]
0xF2 42	0x00	packetmemory_reg_f242	rw	acp_header_byte_2[7]	acp_header_byte_2[6]	acp_header_byte_2[5]	acp_header_byte_2[4]	acp_header_byte_2[3]	acp_header_byte_2[2]	acp_header_byte_2[1]	acp_header_byte_2[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF243	0x00	packetmemory_reg_f243	rw	acp_packet_byte_0[7]	acp_packet_byte_0[6]	acp_packet_byte_0[5]	acp_packet_byte_0[4]	acp_packet_byte_0[3]	acp_packet_byte_0[2]	acp_packet_byte_0[1]	acp_packet_byte_0[0]
0xF244	0x00	packetmemory_reg_f244	rw	acp_packet_byte_1[7]	acp_packet_byte_1[6]	acp_packet_byte_1[5]	acp_packet_byte_1[4]	acp_packet_byte_1[3]	acp_packet_byte_1[2]	acp_packet_byte_1[1]	acp_packet_byte_1[0]
0xF245	0x00	packetmemory_reg_f245	rw	acp_packet_byte_2[7]	acp_packet_byte_2[6]	acp_packet_byte_2[5]	acp_packet_byte_2[4]	acp_packet_byte_2[3]	acp_packet_byte_2[2]	acp_packet_byte_2[1]	acp_packet_byte_2[0]
0xF246	0x00	packetmemory_reg_f246	rw	acp_packet_byte_3[7]	acp_packet_byte_3[6]	acp_packet_byte_3[5]	acp_packet_byte_3[4]	acp_packet_byte_3[3]	acp_packet_byte_3[2]	acp_packet_byte_3[1]	acp_packet_byte_3[0]
0xF247	0x00	packetmemory_reg_f247	rw	acp_packet_byte_4[7]	acp_packet_byte_4[6]	acp_packet_byte_4[5]	acp_packet_byte_4[4]	acp_packet_byte_4[3]	acp_packet_byte_4[2]	acp_packet_byte_4[1]	acp_packet_byte_4[0]
0xF248	0x00	packetmemory_reg_f248	rw	acp_packet_byte_5[7]	acp_packet_byte_5[6]	acp_packet_byte_5[5]	acp_packet_byte_5[4]	acp_packet_byte_5[3]	acp_packet_byte_5[2]	acp_packet_byte_5[1]	acp_packet_byte_5[0]
0xF249	0x00	packetmemory_reg_f249	rw	acp_packet_byte_6[7]	acp_packet_byte_6[6]	acp_packet_byte_6[5]	acp_packet_byte_6[4]	acp_packet_byte_6[3]	acp_packet_byte_6[2]	acp_packet_byte_6[1]	acp_packet_byte_6[0]
0xF24A	0x00	packetmemory_reg_f24a	rw	acp_packet_byte_7[7]	acp_packet_byte_7[6]	acp_packet_byte_7[5]	acp_packet_byte_7[4]	acp_packet_byte_7[3]	acp_packet_byte_7[2]	acp_packet_byte_7[1]	acp_packet_byte_7[0]
0xF24B	0x00	packetmemory_reg_f24b	rw	acp_packet_byte_8[7]	acp_packet_byte_8[6]	acp_packet_byte_8[5]	acp_packet_byte_8[4]	acp_packet_byte_8[3]	acp_packet_byte_8[2]	acp_packet_byte_8[1]	acp_packet_byte_8[0]
0xF24C	0x00	packetmemory_reg_f24c	rw	acp_packet_byte_9[7]	acp_packet_byte_9[6]	acp_packet_byte_9[5]	acp_packet_byte_9[4]	acp_packet_byte_9[3]	acp_packet_byte_9[2]	acp_packet_byte_9[1]	acp_packet_byte_9[0]
0xF24D	0x00	packetmemory_reg_f24d	rw	acp_packet_byte_10[7]	acp_packet_byte_10[6]	acp_packet_byte_10[5]	acp_packet_byte_10[4]	acp_packet_byte_10[3]	acp_packet_byte_10[2]	acp_packet_byte_10[1]	acp_packet_byte_10[0]
0xF24E	0x00	packetmemory_reg_f24e	rw	acp_packet_byte_11[7]	acp_packet_byte_11[6]	acp_packet_byte_11[5]	acp_packet_byte_11[4]	acp_packet_byte_11[3]	acp_packet_byte_11[2]	acp_packet_byte_11[1]	acp_packet_byte_11[0]
0xF24F	0x00	packetmemory_reg_f24f	rw	acp_packet_byte_12[7]	acp_packet_byte_12[6]	acp_packet_byte_12[5]	acp_packet_byte_12[4]	acp_packet_byte_12[3]	acp_packet_byte_12[2]	acp_packet_byte_12[1]	acp_packet_byte_12[0]
0xF250	0x00	packetmemory_reg_f250	rw	acp_packet_byte_13[7]	acp_packet_byte_13[6]	acp_packet_byte_13[5]	acp_packet_byte_13[4]	acp_packet_byte_13[3]	acp_packet_byte_13[2]	acp_packet_byte_13[1]	acp_packet_byte_13[0]
0xF251	0x00	packetmemory_reg_f251	rw	acp_packet_byte_14[7]	acp_packet_byte_14[6]	acp_packet_byte_14[5]	acp_packet_byte_14[4]	acp_packet_byte_14[3]	acp_packet_byte_14[2]	acp_packet_byte_14[1]	acp_packet_byte_14[0]
0xF252	0x00	packetmemory_reg_f252	rw	acp_packet_byte_15[7]	acp_packet_byte_15[6]	acp_packet_byte_15[5]	acp_packet_byte_15[4]	acp_packet_byte_15[3]	acp_packet_byte_15[2]	acp_packet_byte_15[1]	acp_packet_byte_15[0]
0xF253	0x00	packetmemory_reg_f253	rw	acp_packet_byte_16[7]	acp_packet_byte_16[6]	acp_packet_byte_16[5]	acp_packet_byte_16[4]	acp_packet_byte_16[3]	acp_packet_byte_16[2]	acp_packet_byte_16[1]	acp_packet_byte_16[0]
0xF254	0x00	packetmemory_reg_f254	rw	acp_packet_byte_17[7]	acp_packet_byte_17[6]	acp_packet_byte_17[5]	acp_packet_byte_17[4]	acp_packet_byte_17[3]	acp_packet_byte_17[2]	acp_packet_byte_17[1]	acp_packet_byte_17[0]
0xF255	0x00	packetmemory_reg_f255	rw	acp_packet_byte_18[7]	acp_packet_byte_18[6]	acp_packet_byte_18[5]	acp_packet_byte_18[4]	acp_packet_byte_18[3]	acp_packet_byte_18[2]	acp_packet_byte_18[1]	acp_packet_byte_18[0]
0xF256	0x00	packetmemory_reg_f256	rw	acp_packet_byte_19[7]	acp_packet_byte_19[6]	acp_packet_byte_19[5]	acp_packet_byte_19[4]	acp_packet_byte_19[3]	acp_packet_byte_19[2]	acp_packet_byte_19[1]	acp_packet_byte_19[0]
0xF257	0x00	packetmemory_reg_f257	rw	acp_packet_byte_20[7]	acp_packet_byte_20[6]	acp_packet_byte_20[5]	acp_packet_byte_20[4]	acp_packet_byte_20[3]	acp_packet_byte_20[2]	acp_packet_byte_20[1]	acp_packet_byte_20[0]
0xF258	0x00	packetmemory_reg_f258	rw	acp_packet_byte_21[7]	acp_packet_byte_21[6]	acp_packet_byte_21[5]	acp_packet_byte_21[4]	acp_packet_byte_21[3]	acp_packet_byte_21[2]	acp_packet_byte_21[1]	acp_packet_byte_21[0]
0xF259	0x00	packetmemory_reg_f259	rw	acp_packet_byte_22[7]	acp_packet_byte_22[6]	acp_packet_byte_22[5]	acp_packet_byte_22[4]	acp_packet_byte_22[3]	acp_packet_byte_22[2]	acp_packet_byte_22[1]	acp_packet_byte_22[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF25A	0x00	packetmemory_reg_f25a	rw	acp_packet_byte_23[7]	acp_packet_byte_23[6]	acp_packet_byte_23[5]	acp_packet_byte_23[4]	acp_packet_byte_23[3]	acp_packet_byte_23[2]	acp_packet_byte_23[1]	acp_packet_byte_23[0]
0xF25B	0x00	packetmemory_reg_f25b	rw	acp_packet_byte_24[7]	acp_packet_byte_24[6]	acp_packet_byte_24[5]	acp_packet_byte_24[4]	acp_packet_byte_24[3]	acp_packet_byte_24[2]	acp_packet_byte_24[1]	acp_packet_byte_24[0]
0xF25C	0x00	packetmemory_reg_f25c	rw	acp_packet_byte_25[7]	acp_packet_byte_25[6]	acp_packet_byte_25[5]	acp_packet_byte_25[4]	acp_packet_byte_25[3]	acp_packet_byte_25[2]	acp_packet_byte_25[1]	acp_packet_byte_25[0]
0xF25D	0x00	packetmemory_reg_f25d	rw	acp_packet_byte_26[7]	acp_packet_byte_26[6]	acp_packet_byte_26[5]	acp_packet_byte_26[4]	acp_packet_byte_26[3]	acp_packet_byte_26[2]	acp_packet_byte_26[1]	acp_packet_byte_26[0]
0xF25E	0x00	packetmemory_reg_f25e	rw	acp_packet_byte_27[7]	acp_packet_byte_27[6]	acp_packet_byte_27[5]	acp_packet_byte_27[4]	acp_packet_byte_27[3]	acp_packet_byte_27[2]	acp_packet_byte_27[1]	acp_packet_byte_27[0]
0xF25F	0x00	packetmemory_reg_f25f	rw	acp_update	-	-	-	-	-	-	-
0xF260	0x00	packetmemory_reg_f260	rw	isrc1_header_byte_0[7]	isrc1_header_byte_0[6]	isrc1_header_byte_0[5]	isrc1_header_byte_0[4]	isrc1_header_byte_0[3]	isrc1_header_byte_0[2]	isrc1_header_byte_0[1]	isrc1_header_byte_0[0]
0xF261	0x00	packetmemory_reg_f261	rw	isrc1_header_byte_1[7]	isrc1_header_byte_1[6]	isrc1_header_byte_1[5]	isrc1_header_byte_1[4]	isrc1_header_byte_1[3]	isrc1_header_byte_1[2]	isrc1_header_byte_1[1]	isrc1_header_byte_1[0]
0xF262	0x00	packetmemory_reg_f262	rw	isrc1_header_byte_2[7]	isrc1_header_byte_2[6]	isrc1_header_byte_2[5]	isrc1_header_byte_2[4]	isrc1_header_byte_2[3]	isrc1_header_byte_2[2]	isrc1_header_byte_2[1]	isrc1_header_byte_2[0]
0xF263	0x00	packetmemory_reg_f263	rw	isrc1_packet_byte_0[7]	isrc1_packet_byte_0[6]	isrc1_packet_byte_0[5]	isrc1_packet_byte_0[4]	isrc1_packet_byte_0[3]	isrc1_packet_byte_0[2]	isrc1_packet_byte_0[1]	isrc1_packet_byte_0[0]
0xF264	0x00	packetmemory_reg_f264	rw	isrc1_packet_byte_1[7]	isrc1_packet_byte_1[6]	isrc1_packet_byte_1[5]	isrc1_packet_byte_1[4]	isrc1_packet_byte_1[3]	isrc1_packet_byte_1[2]	isrc1_packet_byte_1[1]	isrc1_packet_byte_1[0]
0xF265	0x00	packetmemory_reg_f265	rw	isrc1_packet_byte_2[7]	isrc1_packet_byte_2[6]	isrc1_packet_byte_2[5]	isrc1_packet_byte_2[4]	isrc1_packet_byte_2[3]	isrc1_packet_byte_2[2]	isrc1_packet_byte_2[1]	isrc1_packet_byte_2[0]
0xF266	0x00	packetmemory_reg_f266	rw	isrc1_packet_byte_3[7]	isrc1_packet_byte_3[6]	isrc1_packet_byte_3[5]	isrc1_packet_byte_3[4]	isrc1_packet_byte_3[3]	isrc1_packet_byte_3[2]	isrc1_packet_byte_3[1]	isrc1_packet_byte_3[0]
0xF267	0x00	packetmemory_reg_f267	rw	isrc1_packet_byte_4[7]	isrc1_packet_byte_4[6]	isrc1_packet_byte_4[5]	isrc1_packet_byte_4[4]	isrc1_packet_byte_4[3]	isrc1_packet_byte_4[2]	isrc1_packet_byte_4[1]	isrc1_packet_byte_4[0]
0xF268	0x00	packetmemory_reg_f268	rw	isrc1_packet_byte_5[7]	isrc1_packet_byte_5[6]	isrc1_packet_byte_5[5]	isrc1_packet_byte_5[4]	isrc1_packet_byte_5[3]	isrc1_packet_byte_5[2]	isrc1_packet_byte_5[1]	isrc1_packet_byte_5[0]
0xF269	0x00	packetmemory_reg_f269	rw	isrc1_packet_byte_6[7]	isrc1_packet_byte_6[6]	isrc1_packet_byte_6[5]	isrc1_packet_byte_6[4]	isrc1_packet_byte_6[3]	isrc1_packet_byte_6[2]	isrc1_packet_byte_6[1]	isrc1_packet_byte_6[0]
0xF26A	0x00	packetmemory_reg_f26a	rw	isrc1_packet_byte_7[7]	isrc1_packet_byte_7[6]	isrc1_packet_byte_7[5]	isrc1_packet_byte_7[4]	isrc1_packet_byte_7[3]	isrc1_packet_byte_7[2]	isrc1_packet_byte_7[1]	isrc1_packet_byte_7[0]
0xF26B	0x00	packetmemory_reg_f26b	rw	isrc1_packet_byte_8[7]	isrc1_packet_byte_8[6]	isrc1_packet_byte_8[5]	isrc1_packet_byte_8[4]	isrc1_packet_byte_8[3]	isrc1_packet_byte_8[2]	isrc1_packet_byte_8[1]	isrc1_packet_byte_8[0]
0xF26C	0x00	packetmemory_reg_f26c	rw	isrc1_packet_byte_9[7]	isrc1_packet_byte_9[6]	isrc1_packet_byte_9[5]	isrc1_packet_byte_9[4]	isrc1_packet_byte_9[3]	isrc1_packet_byte_9[2]	isrc1_packet_byte_9[1]	isrc1_packet_byte_9[0]
0xF26D	0x00	packetmemory_reg_f26d	rw	isrc1_packet_byte_10[7]	isrc1_packet_byte_10[6]	isrc1_packet_byte_10[5]	isrc1_packet_byte_10[4]	isrc1_packet_byte_10[3]	isrc1_packet_byte_10[2]	isrc1_packet_byte_10[1]	isrc1_packet_byte_10[0]
0xF26E	0x00	packetmemory_reg_f26e	rw	isrc1_packet_byte_11[7]	isrc1_packet_byte_11[6]	isrc1_packet_byte_11[5]	isrc1_packet_byte_11[4]	isrc1_packet_byte_11[3]	isrc1_packet_byte_11[2]	isrc1_packet_byte_11[1]	isrc1_packet_byte_11[0]
0xF26F	0x00	packetmemory_reg_f26f	rw	isrc1_packet_byte_12[7]	isrc1_packet_byte_12[6]	isrc1_packet_byte_12[5]	isrc1_packet_byte_12[4]	isrc1_packet_byte_12[3]	isrc1_packet_byte_12[2]	isrc1_packet_byte_12[1]	isrc1_packet_byte_12[0]
0xF270	0x00	packetmemory_reg_f270	r	isrc1_packet_byte_13[7]	isrc1_packet_byte_13[6]	isrc1_packet_byte_13[5]	isrc1_packet_byte_13[4]	isrc1_packet_byte_13[3]	isrc1_packet_byte_13[2]	isrc1_packet_byte_13[1]	isrc1_packet_byte_13[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF271	0x00	packetmemory_reg_f271	r	isrc1_packet_byte_14[7]	isrc1_packet_byte_14[6]	isrc1_packet_byte_14[5]	isrc1_packet_byte_14[4]	isrc1_packet_byte_14[3]	isrc1_packet_byte_14[2]	isrc1_packet_byte_14[1]	isrc1_packet_byte_14[0]
0xF272	0x00	packetmemory_reg_f272	r	isrc1_packet_byte_15[7]	isrc1_packet_byte_15[6]	isrc1_packet_byte_15[5]	isrc1_packet_byte_15[4]	isrc1_packet_byte_15[3]	isrc1_packet_byte_15[2]	isrc1_packet_byte_15[1]	isrc1_packet_byte_15[0]
0xF273	0x00	packetmemory_reg_f273	r	isrc1_packet_byte_16[7]	isrc1_packet_byte_16[6]	isrc1_packet_byte_16[5]	isrc1_packet_byte_16[4]	isrc1_packet_byte_16[3]	isrc1_packet_byte_16[2]	isrc1_packet_byte_16[1]	isrc1_packet_byte_16[0]
0xF274	0x00	packetmemory_reg_f274	r	isrc1_packet_byte_17[7]	isrc1_packet_byte_17[6]	isrc1_packet_byte_17[5]	isrc1_packet_byte_17[4]	isrc1_packet_byte_17[3]	isrc1_packet_byte_17[2]	isrc1_packet_byte_17[1]	isrc1_packet_byte_17[0]
0xF275	0x00	packetmemory_reg_f275	rw	isrc1_packet_byte_18[7]	isrc1_packet_byte_18[6]	isrc1_packet_byte_18[5]	isrc1_packet_byte_18[4]	isrc1_packet_byte_18[3]	isrc1_packet_byte_18[2]	isrc1_packet_byte_18[1]	isrc1_packet_byte_18[0]
0xF276	0x00	packetmemory_reg_f276	rw	isrc1_packet_byte_19[7]	isrc1_packet_byte_19[6]	isrc1_packet_byte_19[5]	isrc1_packet_byte_19[4]	isrc1_packet_byte_19[3]	isrc1_packet_byte_19[2]	isrc1_packet_byte_19[1]	isrc1_packet_byte_19[0]
0xF277	0x00	packetmemory_reg_f277	rw	isrc1_packet_byte_20[7]	isrc1_packet_byte_20[6]	isrc1_packet_byte_20[5]	isrc1_packet_byte_20[4]	isrc1_packet_byte_20[3]	isrc1_packet_byte_20[2]	isrc1_packet_byte_20[1]	isrc1_packet_byte_20[0]
0xF278	0x00	packetmemory_reg_f278	rw	isrc1_packet_byte_21[7]	isrc1_packet_byte_21[6]	isrc1_packet_byte_21[5]	isrc1_packet_byte_21[4]	isrc1_packet_byte_21[3]	isrc1_packet_byte_21[2]	isrc1_packet_byte_21[1]	isrc1_packet_byte_21[0]
0xF279	0x00	packetmemory_reg_f279	rw	isrc1_packet_byte_22[7]	isrc1_packet_byte_22[6]	isrc1_packet_byte_22[5]	isrc1_packet_byte_22[4]	isrc1_packet_byte_22[3]	isrc1_packet_byte_22[2]	isrc1_packet_byte_22[1]	isrc1_packet_byte_22[0]
0xF27A	0x00	packetmemory_reg_f27a	rw	isrc1_packet_byte_23[7]	isrc1_packet_byte_23[6]	isrc1_packet_byte_23[5]	isrc1_packet_byte_23[4]	isrc1_packet_byte_23[3]	isrc1_packet_byte_23[2]	isrc1_packet_byte_23[1]	isrc1_packet_byte_23[0]
0xF27B	0x00	packetmemory_reg_f27b	rw	isrc1_packet_byte_24[7]	isrc1_packet_byte_24[6]	isrc1_packet_byte_24[5]	isrc1_packet_byte_24[4]	isrc1_packet_byte_24[3]	isrc1_packet_byte_24[2]	isrc1_packet_byte_24[1]	isrc1_packet_byte_24[0]
0xF27C	0x00	packetmemory_reg_f27c	rw	isrc1_packet_byte_25[7]	isrc1_packet_byte_25[6]	isrc1_packet_byte_25[5]	isrc1_packet_byte_25[4]	isrc1_packet_byte_25[3]	isrc1_packet_byte_25[2]	isrc1_packet_byte_25[1]	isrc1_packet_byte_25[0]
0xF27D	0x00	packetmemory_reg_f27d	rw	isrc1_packet_byte_26[7]	isrc1_packet_byte_26[6]	isrc1_packet_byte_26[5]	isrc1_packet_byte_26[4]	isrc1_packet_byte_26[3]	isrc1_packet_byte_26[2]	isrc1_packet_byte_26[1]	isrc1_packet_byte_26[0]
0xF27E	0x00	packetmemory_reg_f27e	rw	isrc1_packet_byte_27[7]	isrc1_packet_byte_27[6]	isrc1_packet_byte_27[5]	isrc1_packet_byte_27[4]	isrc1_packet_byte_27[3]	isrc1_packet_byte_27[2]	isrc1_packet_byte_27[1]	isrc1_packet_byte_27[0]
0xF27F	0x00	packetmemory_reg_f27f	rw	isrc1_update	-	-	-	-	-	-	-
0xF280	0x00	packetmemory_reg_f280	rw	isrc2_header_byte_0[7]	isrc2_header_byte_0[6]	isrc2_header_byte_0[5]	isrc2_header_byte_0[4]	isrc2_header_byte_0[3]	isrc2_header_byte_0[2]	isrc2_header_byte_0[1]	isrc2_header_byte_0[0]
0xF281	0x00	packetmemory_reg_f281	rw	isrc2_header_byte_1[7]	isrc2_header_byte_1[6]	isrc2_header_byte_1[5]	isrc2_header_byte_1[4]	isrc2_header_byte_1[3]	isrc2_header_byte_1[2]	isrc2_header_byte_1[1]	isrc2_header_byte_1[0]
0xF282	0x00	packetmemory_reg_f282	rw	isrc2_header_byte_2[7]	isrc2_header_byte_2[6]	isrc2_header_byte_2[5]	isrc2_header_byte_2[4]	isrc2_header_byte_2[3]	isrc2_header_byte_2[2]	isrc2_header_byte_2[1]	isrc2_header_byte_2[0]
0xF283	0x00	packetmemory_reg_f283	rw	isrc2_packet_byte_0[7]	isrc2_packet_byte_0[6]	isrc2_packet_byte_0[5]	isrc2_packet_byte_0[4]	isrc2_packet_byte_0[3]	isrc2_packet_byte_0[2]	isrc2_packet_byte_0[1]	isrc2_packet_byte_0[0]
0xF284	0x00	packetmemory_reg_f284	rw	isrc2_packet_byte_1[7]	isrc2_packet_byte_1[6]	isrc2_packet_byte_1[5]	isrc2_packet_byte_1[4]	isrc2_packet_byte_1[3]	isrc2_packet_byte_1[2]	isrc2_packet_byte_1[1]	isrc2_packet_byte_1[0]
0xF285	0x00	packetmemory_reg_f285	rw	isrc2_packet_byte_2[7]	isrc2_packet_byte_2[6]	isrc2_packet_byte_2[5]	isrc2_packet_byte_2[4]	isrc2_packet_byte_2[3]	isrc2_packet_byte_2[2]	isrc2_packet_byte_2[1]	isrc2_packet_byte_2[0]
0xF286	0x00	packetmemory_reg_f286	rw	isrc2_packet_byte_3[7]	isrc2_packet_byte_3[6]	isrc2_packet_byte_3[5]	isrc2_packet_byte_3[4]	isrc2_packet_byte_3[3]	isrc2_packet_byte_3[2]	isrc2_packet_byte_3[1]	isrc2_packet_byte_3[0]
0xF287	0x00	packetmemory_reg_f287	rw	isrc2_packet_byte_4[7]	isrc2_packet_byte_4[6]	isrc2_packet_byte_4[5]	isrc2_packet_byte_4[4]	isrc2_packet_byte_4[3]	isrc2_packet_byte_4[2]	isrc2_packet_byte_4[1]	isrc2_packet_byte_4[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF288	0x00	packetmemory_reg_f288	rw	isrc2_packet_byte_5[7]	isrc2_packet_byte_5[6]	isrc2_packet_byte_5[5]	isrc2_packet_byte_5[4]	isrc2_packet_byte_5[3]	isrc2_packet_byte_5[2]	isrc2_packet_byte_5[1]	isrc2_packet_byte_5[0]
0xF289	0x00	packetmemory_reg_f289	rw	isrc2_packet_byte_6[7]	isrc2_packet_byte_6[6]	isrc2_packet_byte_6[5]	isrc2_packet_byte_6[4]	isrc2_packet_byte_6[3]	isrc2_packet_byte_6[2]	isrc2_packet_byte_6[1]	isrc2_packet_byte_6[0]
0xF28A	0x00	packetmemory_reg_f28a	rw	isrc2_packet_byte_7[7]	isrc2_packet_byte_7[6]	isrc2_packet_byte_7[5]	isrc2_packet_byte_7[4]	isrc2_packet_byte_7[3]	isrc2_packet_byte_7[2]	isrc2_packet_byte_7[1]	isrc2_packet_byte_7[0]
0xF28B	0x00	packetmemory_reg_f28b	rw	isrc2_packet_byte_8[7]	isrc2_packet_byte_8[6]	isrc2_packet_byte_8[5]	isrc2_packet_byte_8[4]	isrc2_packet_byte_8[3]	isrc2_packet_byte_8[2]	isrc2_packet_byte_8[1]	isrc2_packet_byte_8[0]
0xF28C	0x00	packetmemory_reg_f28c	rw	isrc2_packet_byte_9[7]	isrc2_packet_byte_9[6]	isrc2_packet_byte_9[5]	isrc2_packet_byte_9[4]	isrc2_packet_byte_9[3]	isrc2_packet_byte_9[2]	isrc2_packet_byte_9[1]	isrc2_packet_byte_9[0]
0xF28D	0x00	packetmemory_reg_f28d	rw	isrc2_packet_byte_10[7]	isrc2_packet_byte_10[6]	isrc2_packet_byte_10[5]	isrc2_packet_byte_10[4]	isrc2_packet_byte_10[3]	isrc2_packet_byte_10[2]	isrc2_packet_byte_10[1]	isrc2_packet_byte_10[0]
0xF28E	0x00	packetmemory_reg_f28e	rw	isrc2_packet_byte_11[7]	isrc2_packet_byte_11[6]	isrc2_packet_byte_11[5]	isrc2_packet_byte_11[4]	isrc2_packet_byte_11[3]	isrc2_packet_byte_11[2]	isrc2_packet_byte_11[1]	isrc2_packet_byte_11[0]
0xF28F	0x00	packetmemory_reg_f28f	rw	isrc2_packet_byte_12[7]	isrc2_packet_byte_12[6]	isrc2_packet_byte_12[5]	isrc2_packet_byte_12[4]	isrc2_packet_byte_12[3]	isrc2_packet_byte_12[2]	isrc2_packet_byte_12[1]	isrc2_packet_byte_12[0]
0xF290	0x00	packetmemory_reg_f290	rw	isrc2_packet_byte_13[7]	isrc2_packet_byte_13[6]	isrc2_packet_byte_13[5]	isrc2_packet_byte_13[4]	isrc2_packet_byte_13[3]	isrc2_packet_byte_13[2]	isrc2_packet_byte_13[1]	isrc2_packet_byte_13[0]
0xF291	0x00	packetmemory_reg_f291	rw	isrc2_packet_byte_14[7]	isrc2_packet_byte_14[6]	isrc2_packet_byte_14[5]	isrc2_packet_byte_14[4]	isrc2_packet_byte_14[3]	isrc2_packet_byte_14[2]	isrc2_packet_byte_14[1]	isrc2_packet_byte_14[0]
0xF292	0x00	packetmemory_reg_f292	rw	isrc2_packet_byte_15[7]	isrc2_packet_byte_15[6]	isrc2_packet_byte_15[5]	isrc2_packet_byte_15[4]	isrc2_packet_byte_15[3]	isrc2_packet_byte_15[2]	isrc2_packet_byte_15[1]	isrc2_packet_byte_15[0]
0xF293	0x00	packetmemory_reg_f293	rw	isrc2_packet_byte_16[7]	isrc2_packet_byte_16[6]	isrc2_packet_byte_16[5]	isrc2_packet_byte_16[4]	isrc2_packet_byte_16[3]	isrc2_packet_byte_16[2]	isrc2_packet_byte_16[1]	isrc2_packet_byte_16[0]
0xF294	0x00	packetmemory_reg_f294	rw	isrc2_packet_byte_17[7]	isrc2_packet_byte_17[6]	isrc2_packet_byte_17[5]	isrc2_packet_byte_17[4]	isrc2_packet_byte_17[3]	isrc2_packet_byte_17[2]	isrc2_packet_byte_17[1]	isrc2_packet_byte_17[0]
0xF295	0x00	packetmemory_reg_f295	rw	isrc2_packet_byte_18[7]	isrc2_packet_byte_18[6]	isrc2_packet_byte_18[5]	isrc2_packet_byte_18[4]	isrc2_packet_byte_18[3]	isrc2_packet_byte_18[2]	isrc2_packet_byte_18[1]	isrc2_packet_byte_18[0]
0xF296	0x00	packetmemory_reg_f296	rw	isrc2_packet_byte_19[7]	isrc2_packet_byte_19[6]	isrc2_packet_byte_19[5]	isrc2_packet_byte_19[4]	isrc2_packet_byte_19[3]	isrc2_packet_byte_19[2]	isrc2_packet_byte_19[1]	isrc2_packet_byte_19[0]
0xF297	0x00	packetmemory_reg_f297	rw	isrc2_packet_byte_20[7]	isrc2_packet_byte_20[6]	isrc2_packet_byte_20[5]	isrc2_packet_byte_20[4]	isrc2_packet_byte_20[3]	isrc2_packet_byte_20[2]	isrc2_packet_byte_20[1]	isrc2_packet_byte_20[0]
0xF298	0x00	packetmemory_reg_f298	rw	isrc2_packet_byte_21[7]	isrc2_packet_byte_21[6]	isrc2_packet_byte_21[5]	isrc2_packet_byte_21[4]	isrc2_packet_byte_21[3]	isrc2_packet_byte_21[2]	isrc2_packet_byte_21[1]	isrc2_packet_byte_21[0]
0xF299	0x00	packetmemory_reg_f299	rw	isrc2_packet_byte_22[7]	isrc2_packet_byte_22[6]	isrc2_packet_byte_22[5]	isrc2_packet_byte_22[4]	isrc2_packet_byte_22[3]	isrc2_packet_byte_22[2]	isrc2_packet_byte_22[1]	isrc2_packet_byte_22[0]
0xF29A	0x00	packetmemory_reg_f29a	rw	isrc2_packet_byte_23[7]	isrc2_packet_byte_23[6]	isrc2_packet_byte_23[5]	isrc2_packet_byte_23[4]	isrc2_packet_byte_23[3]	isrc2_packet_byte_23[2]	isrc2_packet_byte_23[1]	isrc2_packet_byte_23[0]
0xF29B	0x00	packetmemory_reg_f29b	rw	isrc2_packet_byte_24[7]	isrc2_packet_byte_24[6]	isrc2_packet_byte_24[5]	isrc2_packet_byte_24[4]	isrc2_packet_byte_24[3]	isrc2_packet_byte_24[2]	isrc2_packet_byte_24[1]	isrc2_packet_byte_24[0]
0xF29C	0x00	packetmemory_reg_f29c	rw	isrc2_packet_byte_25[7]	isrc2_packet_byte_25[6]	isrc2_packet_byte_25[5]	isrc2_packet_byte_25[4]	isrc2_packet_byte_25[3]	isrc2_packet_byte_25[2]	isrc2_packet_byte_25[1]	isrc2_packet_byte_25[0]
0xF29D	0x00	packetmemory_reg_f29d	rw	isrc2_packet_byte_26[7]	isrc2_packet_byte_26[6]	isrc2_packet_byte_26[5]	isrc2_packet_byte_26[4]	isrc2_packet_byte_26[3]	isrc2_packet_byte_26[2]	isrc2_packet_byte_26[1]	isrc2_packet_byte_26[0]
0xF29E	0x00	packetmemory_reg_f29e	rw	isrc2_packet_byte_27[7]	isrc2_packet_byte_27[6]	isrc2_packet_byte_27[5]	isrc2_packet_byte_27[4]	isrc2_packet_byte_27[3]	isrc2_packet_byte_27[2]	isrc2_packet_byte_27[1]	isrc2_packet_byte_27[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF29F	0x00	packetmemory_reg_f29f	rw	isrc2_update	-	-	-	-	-	-	-
0xF2A0	0x00	packetmemory_reg_f2a0	rw	gm_header_byte_0[7]	gm_header_byte_0[6]	gm_header_byte_0[5]	gm_header_byte_0[4]	gm_header_byte_0[3]	gm_header_byte_0[2]	gm_header_byte_0[1]	gm_header_byte_0[0]
0xF2A1	0x00	packetmemory_reg_f2a1	rw	gm_header_byte_1[7]	gm_header_byte_1[6]	gm_header_byte_1[5]	gm_header_byte_1[4]	gm_header_byte_1[3]	gm_header_byte_1[2]	gm_header_byte_1[1]	gm_header_byte_1[0]
0xF2A2	0x00	packetmemory_reg_f2a2	rw	gm_header_byte_2[7]	gm_header_byte_2[6]	gm_header_byte_2[5]	gm_header_byte_2[4]	gm_header_byte_2[3]	gm_header_byte_2[2]	gm_header_byte_2[1]	gm_header_byte_2[0]
0xF2A3	0x00	packetmemory_reg_f2a3	rw	gm_packet_byte_0[7]	gm_packet_byte_0[6]	gm_packet_byte_0[5]	gm_packet_byte_0[4]	gm_packet_byte_0[3]	gm_packet_byte_0[2]	gm_packet_byte_0[1]	gm_packet_byte_0[0]
0xF2A4	0x00	packetmemory_reg_f2a4	rw	gm_packet_byte_1[7]	gm_packet_byte_1[6]	gm_packet_byte_1[5]	gm_packet_byte_1[4]	gm_packet_byte_1[3]	gm_packet_byte_1[2]	gm_packet_byte_1[1]	gm_packet_byte_1[0]
0xF2A5	0x00	packetmemory_reg_f2a5	rw	gm_packet_byte_2[7]	gm_packet_byte_2[6]	gm_packet_byte_2[5]	gm_packet_byte_2[4]	gm_packet_byte_2[3]	gm_packet_byte_2[2]	gm_packet_byte_2[1]	gm_packet_byte_2[0]
0xF2A6	0x00	packetmemory_reg_f2a6	rw	gm_packet_byte_3[7]	gm_packet_byte_3[6]	gm_packet_byte_3[5]	gm_packet_byte_3[4]	gm_packet_byte_3[3]	gm_packet_byte_3[2]	gm_packet_byte_3[1]	gm_packet_byte_3[0]
0xF2A7	0x00	packetmemory_reg_f2a7	rw	gm_packet_byte_4[7]	gm_packet_byte_4[6]	gm_packet_byte_4[5]	gm_packet_byte_4[4]	gm_packet_byte_4[3]	gm_packet_byte_4[2]	gm_packet_byte_4[1]	gm_packet_byte_4[0]
0xF2A8	0x00	packetmemory_reg_f2a8	rw	gm_packet_byte_5[7]	gm_packet_byte_5[6]	gm_packet_byte_5[5]	gm_packet_byte_5[4]	gm_packet_byte_5[3]	gm_packet_byte_5[2]	gm_packet_byte_5[1]	gm_packet_byte_5[0]
0xF2A9	0x00	packetmemory_reg_f2a9	rw	gm_packet_byte_6[7]	gm_packet_byte_6[6]	gm_packet_byte_6[5]	gm_packet_byte_6[4]	gm_packet_byte_6[3]	gm_packet_byte_6[2]	gm_packet_byte_6[1]	gm_packet_byte_6[0]
0xF2AA	0x00	packetmemory_reg_f2aa	rw	gm_packet_byte_7[7]	gm_packet_byte_7[6]	gm_packet_byte_7[5]	gm_packet_byte_7[4]	gm_packet_byte_7[3]	gm_packet_byte_7[2]	gm_packet_byte_7[1]	gm_packet_byte_7[0]
0xF2AB	0x00	packetmemory_reg_f2ab	rw	gm_packet_byte_8[7]	gm_packet_byte_8[6]	gm_packet_byte_8[5]	gm_packet_byte_8[4]	gm_packet_byte_8[3]	gm_packet_byte_8[2]	gm_packet_byte_8[1]	gm_packet_byte_8[0]
0xF2AC	0x00	packetmemory_reg_f2ac	rw	gm_packet_byte_9[7]	gm_packet_byte_9[6]	gm_packet_byte_9[5]	gm_packet_byte_9[4]	gm_packet_byte_9[3]	gm_packet_byte_9[2]	gm_packet_byte_9[1]	gm_packet_byte_9[0]
0xF2AD	0x00	packetmemory_reg_f2ad	rw	gm_packet_byte_10[7]	gm_packet_byte_10[6]	gm_packet_byte_10[5]	gm_packet_byte_10[4]	gm_packet_byte_10[3]	gm_packet_byte_10[2]	gm_packet_byte_10[1]	gm_packet_byte_10[0]
0xF2AE	0x00	packetmemory_reg_f2ae	rw	gm_packet_byte_11[7]	gm_packet_byte_11[6]	gm_packet_byte_11[5]	gm_packet_byte_11[4]	gm_packet_byte_11[3]	gm_packet_byte_11[2]	gm_packet_byte_11[1]	gm_packet_byte_11[0]
0xF2AF	0x00	packetmemory_reg_f2af	rw	gm_packet_byte_12[7]	gm_packet_byte_12[6]	gm_packet_byte_12[5]	gm_packet_byte_12[4]	gm_packet_byte_12[3]	gm_packet_byte_12[2]	gm_packet_byte_12[1]	gm_packet_byte_12[0]
0xF2B0	0x00	packetmemory_reg_f2b0	rw	gm_packet_byte_13[7]	gm_packet_byte_13[6]	gm_packet_byte_13[5]	gm_packet_byte_13[4]	gm_packet_byte_13[3]	gm_packet_byte_13[2]	gm_packet_byte_13[1]	gm_packet_byte_13[0]
0xF2B1	0x00	packetmemory_reg_f2b1	rw	gm_packet_byte_14[7]	gm_packet_byte_14[6]	gm_packet_byte_14[5]	gm_packet_byte_14[4]	gm_packet_byte_14[3]	gm_packet_byte_14[2]	gm_packet_byte_14[1]	gm_packet_byte_14[0]
0xF2B2	0x00	packetmemory_reg_f2b2	rw	gm_packet_byte_15[7]	gm_packet_byte_15[6]	gm_packet_byte_15[5]	gm_packet_byte_15[4]	gm_packet_byte_15[3]	gm_packet_byte_15[2]	gm_packet_byte_15[1]	gm_packet_byte_15[0]
0xF2B3	0x00	packetmemory_reg_f2b3	rw	gm_packet_byte_16[7]	gm_packet_byte_16[6]	gm_packet_byte_16[5]	gm_packet_byte_16[4]	gm_packet_byte_16[3]	gm_packet_byte_16[2]	gm_packet_byte_16[1]	gm_packet_byte_16[0]
0xF2B4	0x00	packetmemory_reg_f2b4	rw	gm_packet_byte_17[7]	gm_packet_byte_17[6]	gm_packet_byte_17[5]	gm_packet_byte_17[4]	gm_packet_byte_17[3]	gm_packet_byte_17[2]	gm_packet_byte_17[1]	gm_packet_byte_17[0]
0xF2B5	0x00	packetmemory_reg_f2b5	rw	gm_packet_byte_18[7]	gm_packet_byte_18[6]	gm_packet_byte_18[5]	gm_packet_byte_18[4]	gm_packet_byte_18[3]	gm_packet_byte_18[2]	gm_packet_byte_18[1]	gm_packet_byte_18[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2B6	0x00	packetmemory_reg_f2b6	rw	gm_packet_byte_19[7]	gm_packet_byte_19[6]	gm_packet_byte_19[5]	gm_packet_byte_19[4]	gm_packet_byte_19[3]	gm_packet_byte_19[2]	gm_packet_byte_19[1]	gm_packet_byte_19[0]
0xF2B7	0x00	packetmemory_reg_f2b7	rw	gm_packet_byte_20[7]	gm_packet_byte_20[6]	gm_packet_byte_20[5]	gm_packet_byte_20[4]	gm_packet_byte_20[3]	gm_packet_byte_20[2]	gm_packet_byte_20[1]	gm_packet_byte_20[0]
0xF2B8	0x00	packetmemory_reg_f2b8	rw	gm_packet_byte_21[7]	gm_packet_byte_21[6]	gm_packet_byte_21[5]	gm_packet_byte_21[4]	gm_packet_byte_21[3]	gm_packet_byte_21[2]	gm_packet_byte_21[1]	gm_packet_byte_21[0]
0xF2B9	0x00	packetmemory_reg_f2b9	rw	gm_packet_byte_22[7]	gm_packet_byte_22[6]	gm_packet_byte_22[5]	gm_packet_byte_22[4]	gm_packet_byte_22[3]	gm_packet_byte_22[2]	gm_packet_byte_22[1]	gm_packet_byte_22[0]
0xF2BA	0x00	packetmemory_reg_f2ba	rw	gm_packet_byte_23[7]	gm_packet_byte_23[6]	gm_packet_byte_23[5]	gm_packet_byte_23[4]	gm_packet_byte_23[3]	gm_packet_byte_23[2]	gm_packet_byte_23[1]	gm_packet_byte_23[0]
0xF2BB	0x00	packetmemory_reg_f2bb	rw	gm_packet_byte_24[7]	gm_packet_byte_24[6]	gm_packet_byte_24[5]	gm_packet_byte_24[4]	gm_packet_byte_24[3]	gm_packet_byte_24[2]	gm_packet_byte_24[1]	gm_packet_byte_24[0]
0xF2BC	0x00	packetmemory_reg_f2bc	rw	gm_packet_byte_25[7]	gm_packet_byte_25[6]	gm_packet_byte_25[5]	gm_packet_byte_25[4]	gm_packet_byte_25[3]	gm_packet_byte_25[2]	gm_packet_byte_25[1]	gm_packet_byte_25[0]
0xF2BD	0x00	packetmemory_reg_f2bd	rw	gm_packet_byte_26[7]	gm_packet_byte_26[6]	gm_packet_byte_26[5]	gm_packet_byte_26[4]	gm_packet_byte_26[3]	gm_packet_byte_26[2]	gm_packet_byte_26[1]	gm_packet_byte_26[0]
0xF2BE	0x00	packetmemory_reg_f2be	rw	gm_packet_byte_27[7]	gm_packet_byte_27[6]	gm_packet_byte_27[5]	gm_packet_byte_27[4]	gm_packet_byte_27[3]	gm_packet_byte_27[2]	gm_packet_byte_27[1]	gm_packet_byte_27[0]
0xF2BF	0x00	packetmemory_reg_f2bf	rw	gm_update	-	-	-	-	-	-	-
0xF2C0	0x00	packetmemory_reg_f2c0	rw	spare_packet_1_header_byte_0[7]	spare_packet_1_header_byte_0[6]	spare_packet_1_header_byte_0[5]	spare_packet_1_header_byte_0[4]	spare_packet_1_header_byte_0[3]	spare_packet_1_header_byte_0[2]	spare_packet_1_header_byte_0[1]	spare_packet_1_header_byte_0[0]
0xF2C1	0x00	packetmemory_reg_f2c1	rw	spare_packet_1_header_byte_1[7]	spare_packet_1_header_byte_1[6]	spare_packet_1_header_byte_1[5]	spare_packet_1_header_byte_1[4]	spare_packet_1_header_byte_1[3]	spare_packet_1_header_byte_1[2]	spare_packet_1_header_byte_1[1]	spare_packet_1_header_byte_1[0]
0xF2C2	0x00	packetmemory_reg_f2c2	rw	spare_packet_1_header_byte_2[7]	spare_packet_1_header_byte_2[6]	spare_packet_1_header_byte_2[5]	spare_packet_1_header_byte_2[4]	spare_packet_1_header_byte_2[3]	spare_packet_1_header_byte_2[2]	spare_packet_1_header_byte_2[1]	spare_packet_1_header_byte_2[0]
0xF2C3	0x00	packetmemory_reg_f2c3	rw	spare_packet_1_packet_byte_0[7]	spare_packet_1_packet_byte_0[6]	spare_packet_1_packet_byte_0[5]	spare_packet_1_packet_byte_0[4]	spare_packet_1_packet_byte_0[3]	spare_packet_1_packet_byte_0[2]	spare_packet_1_packet_byte_0[1]	spare_packet_1_packet_byte_0[0]
0xF2C4	0x00	packetmemory_reg_f2c4	rw	spare_packet_1_packet_byte_1[7]	spare_packet_1_packet_byte_1[6]	spare_packet_1_packet_byte_1[5]	spare_packet_1_packet_byte_1[4]	spare_packet_1_packet_byte_1[3]	spare_packet_1_packet_byte_1[2]	spare_packet_1_packet_byte_1[1]	spare_packet_1_packet_byte_1[0]
0xF2C5	0x00	packetmemory_reg_f2c5	rw	spare_packet_1_packet_byte_2[7]	spare_packet_1_packet_byte_2[6]	spare_packet_1_packet_byte_2[5]	spare_packet_1_packet_byte_2[4]	spare_packet_1_packet_byte_2[3]	spare_packet_1_packet_byte_2[2]	spare_packet_1_packet_byte_2[1]	spare_packet_1_packet_byte_2[0]
0xF2C6	0x00	packetmemory_reg_f2c6	rw	spare_packet_1_packet_byte_3[7]	spare_packet_1_packet_byte_3[6]	spare_packet_1_packet_byte_3[5]	spare_packet_1_packet_byte_3[4]	spare_packet_1_packet_byte_3[3]	spare_packet_1_packet_byte_3[2]	spare_packet_1_packet_byte_3[1]	spare_packet_1_packet_byte_3[0]
0xF2C7	0x00	packetmemory_reg_f2c7	rw	spare_packet_1_packet_byte_4[7]	spare_packet_1_packet_byte_4[6]	spare_packet_1_packet_byte_4[5]	spare_packet_1_packet_byte_4[4]	spare_packet_1_packet_byte_4[3]	spare_packet_1_packet_byte_4[2]	spare_packet_1_packet_byte_4[1]	spare_packet_1_packet_byte_4[0]
0xF2C8	0x00	packetmemory_reg_f2c8	rw	spare_packet_1_packet_byte_5[7]	spare_packet_1_packet_byte_5[6]	spare_packet_1_packet_byte_5[5]	spare_packet_1_packet_byte_5[4]	spare_packet_1_packet_byte_5[3]	spare_packet_1_packet_byte_5[2]	spare_packet_1_packet_byte_5[1]	spare_packet_1_packet_byte_5[0]
0xF2C9	0x00	packetmemory_reg_f2c9	rw	spare_packet_1_packet_byte_6[7]	spare_packet_1_packet_byte_6[6]	spare_packet_1_packet_byte_6[5]	spare_packet_1_packet_byte_6[4]	spare_packet_1_packet_byte_6[3]	spare_packet_1_packet_byte_6[2]	spare_packet_1_packet_byte_6[1]	spare_packet_1_packet_byte_6[0]
0xF2CA	0x00	packetmemory_reg_f2ca	rw	spare_packet_1_packet_byte_7[7]	spare_packet_1_packet_byte_7[6]	spare_packet_1_packet_byte_7[5]	spare_packet_1_packet_byte_7[4]	spare_packet_1_packet_byte_7[3]	spare_packet_1_packet_byte_7[2]	spare_packet_1_packet_byte_7[1]	spare_packet_1_packet_byte_7[0]
0xF2CB	0x00	packetmemory_reg_f2cb	rw	spare_packet_1_packet_byte_8[7]	spare_packet_1_packet_byte_8[6]	spare_packet_1_packet_byte_8[5]	spare_packet_1_packet_byte_8[4]	spare_packet_1_packet_byte_8[3]	spare_packet_1_packet_byte_8[2]	spare_packet_1_packet_byte_8[1]	spare_packet_1_packet_byte_8[0]
0xF2CC	0x00	packetmemory_reg_f2cc	rw	spare_packet_1_packet_byte_9[7]	spare_packet_1_packet_byte_9[6]	spare_packet_1_packet_byte_9[5]	spare_packet_1_packet_byte_9[4]	spare_packet_1_packet_byte_9[3]	spare_packet_1_packet_byte_9[2]	spare_packet_1_packet_byte_9[1]	spare_packet_1_packet_byte_9[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2CD	0x00	packetmemory_reg_f2cd	rw	spare_packet_1_packet_byte_10[7]	spare_packet_1_packet_byte_10[6]	spare_packet_1_packet_byte_10[5]	spare_packet_1_packet_byte_10[4]	spare_packet_1_packet_byte_10[3]	spare_packet_1_packet_byte_10[2]	spare_packet_1_packet_byte_10[1]	spare_packet_1_packet_byte_10[0]
0xF2CE	0x00	packetmemory_reg_f2ce	rw	spare_packet_1_packet_byte_11[7]	spare_packet_1_packet_byte_11[6]	spare_packet_1_packet_byte_11[5]	spare_packet_1_packet_byte_11[4]	spare_packet_1_packet_byte_11[3]	spare_packet_1_packet_byte_11[2]	spare_packet_1_packet_byte_11[1]	spare_packet_1_packet_byte_11[0]
0xF2CF	0x00	packetmemory_reg_f2cf	rw	spare_packet_1_packet_byte_12[7]	spare_packet_1_packet_byte_12[6]	spare_packet_1_packet_byte_12[5]	spare_packet_1_packet_byte_12[4]	spare_packet_1_packet_byte_12[3]	spare_packet_1_packet_byte_12[2]	spare_packet_1_packet_byte_12[1]	spare_packet_1_packet_byte_12[0]
0xF2D0	0x00	packetmemory_reg_f2d0	rw	spare_packet_1_packet_byte_13[7]	spare_packet_1_packet_byte_13[6]	spare_packet_1_packet_byte_13[5]	spare_packet_1_packet_byte_13[4]	spare_packet_1_packet_byte_13[3]	spare_packet_1_packet_byte_13[2]	spare_packet_1_packet_byte_13[1]	spare_packet_1_packet_byte_13[0]
0xF2D1	0x00	packetmemory_reg_f2d1	rw	spare_packet_1_packet_byte_14[7]	spare_packet_1_packet_byte_14[6]	spare_packet_1_packet_byte_14[5]	spare_packet_1_packet_byte_14[4]	spare_packet_1_packet_byte_14[3]	spare_packet_1_packet_byte_14[2]	spare_packet_1_packet_byte_14[1]	spare_packet_1_packet_byte_14[0]
0xF2D2	0x00	packetmemory_reg_f2d2	rw	spare_packet_1_packet_byte_15[7]	spare_packet_1_packet_byte_15[6]	spare_packet_1_packet_byte_15[5]	spare_packet_1_packet_byte_15[4]	spare_packet_1_packet_byte_15[3]	spare_packet_1_packet_byte_15[2]	spare_packet_1_packet_byte_15[1]	spare_packet_1_packet_byte_15[0]
0xF2D3	0x00	packetmemory_reg_f2d3	rw	spare_packet_1_packet_byte_16[7]	spare_packet_1_packet_byte_16[6]	spare_packet_1_packet_byte_16[5]	spare_packet_1_packet_byte_16[4]	spare_packet_1_packet_byte_16[3]	spare_packet_1_packet_byte_16[2]	spare_packet_1_packet_byte_16[1]	spare_packet_1_packet_byte_16[0]
0xF2D4	0x00	packetmemory_reg_f2d4	rw	spare_packet_1_packet_byte_17[7]	spare_packet_1_packet_byte_17[6]	spare_packet_1_packet_byte_17[5]	spare_packet_1_packet_byte_17[4]	spare_packet_1_packet_byte_17[3]	spare_packet_1_packet_byte_17[2]	spare_packet_1_packet_byte_17[1]	spare_packet_1_packet_byte_17[0]
0xF2D5	0x00	packetmemory_reg_f2d5	rw	spare_packet_1_packet_byte_18[7]	spare_packet_1_packet_byte_18[6]	spare_packet_1_packet_byte_18[5]	spare_packet_1_packet_byte_18[4]	spare_packet_1_packet_byte_18[3]	spare_packet_1_packet_byte_18[2]	spare_packet_1_packet_byte_18[1]	spare_packet_1_packet_byte_18[0]
0xF2D6	0x00	packetmemory_reg_f2d6	rw	spare_packet_1_packet_byte_19[7]	spare_packet_1_packet_byte_19[6]	spare_packet_1_packet_byte_19[5]	spare_packet_1_packet_byte_19[4]	spare_packet_1_packet_byte_19[3]	spare_packet_1_packet_byte_19[2]	spare_packet_1_packet_byte_19[1]	spare_packet_1_packet_byte_19[0]
0xF2D7	0x00	packetmemory_reg_f2d7	rw	spare_packet_1_packet_byte_20[7]	spare_packet_1_packet_byte_20[6]	spare_packet_1_packet_byte_20[5]	spare_packet_1_packet_byte_20[4]	spare_packet_1_packet_byte_20[3]	spare_packet_1_packet_byte_20[2]	spare_packet_1_packet_byte_20[1]	spare_packet_1_packet_byte_20[0]
0xF2D8	0x00	packetmemory_reg_f2d8	rw	spare_packet_1_packet_byte_21[7]	spare_packet_1_packet_byte_21[6]	spare_packet_1_packet_byte_21[5]	spare_packet_1_packet_byte_21[4]	spare_packet_1_packet_byte_21[3]	spare_packet_1_packet_byte_21[2]	spare_packet_1_packet_byte_21[1]	spare_packet_1_packet_byte_21[0]
0xF2D9	0x00	packetmemory_reg_f2d9	rw	spare_packet_1_packet_byte_22[7]	spare_packet_1_packet_byte_22[6]	spare_packet_1_packet_byte_22[5]	spare_packet_1_packet_byte_22[4]	spare_packet_1_packet_byte_22[3]	spare_packet_1_packet_byte_22[2]	spare_packet_1_packet_byte_22[1]	spare_packet_1_packet_byte_22[0]
0xF2DA	0x00	packetmemory_reg_f2da	rw	spare_packet_1_packet_byte_23[7]	spare_packet_1_packet_byte_23[6]	spare_packet_1_packet_byte_23[5]	spare_packet_1_packet_byte_23[4]	spare_packet_1_packet_byte_23[3]	spare_packet_1_packet_byte_23[2]	spare_packet_1_packet_byte_23[1]	spare_packet_1_packet_byte_23[0]
0xF2DB	0x00	packetmemory_reg_f2db	rw	spare_packet_1_packet_byte_24[7]	spare_packet_1_packet_byte_24[6]	spare_packet_1_packet_byte_24[5]	spare_packet_1_packet_byte_24[4]	spare_packet_1_packet_byte_24[3]	spare_packet_1_packet_byte_24[2]	spare_packet_1_packet_byte_24[1]	spare_packet_1_packet_byte_24[0]
0xF2DC	0x00	packetmemory_reg_f2dc	rw	spare_packet_1_packet_byte_25[7]	spare_packet_1_packet_byte_25[6]	spare_packet_1_packet_byte_25[5]	spare_packet_1_packet_byte_25[4]	spare_packet_1_packet_byte_25[3]	spare_packet_1_packet_byte_25[2]	spare_packet_1_packet_byte_25[1]	spare_packet_1_packet_byte_25[0]
0xF2DD	0x00	packetmemory_reg_f2dd	rw	spare_packet_1_packet_byte_26[7]	spare_packet_1_packet_byte_26[6]	spare_packet_1_packet_byte_26[5]	spare_packet_1_packet_byte_26[4]	spare_packet_1_packet_byte_26[3]	spare_packet_1_packet_byte_26[2]	spare_packet_1_packet_byte_26[1]	spare_packet_1_packet_byte_26[0]
0xF2DE	0x00	packetmemory_reg_f2de	rw	spare_packet_1_packet_byte_27[7]	spare_packet_1_packet_byte_27[6]	spare_packet_1_packet_byte_27[5]	spare_packet_1_packet_byte_27[4]	spare_packet_1_packet_byte_27[3]	spare_packet_1_packet_byte_27[2]	spare_packet_1_packet_byte_27[1]	spare_packet_1_packet_byte_27[0]
0xF2DF	0x00	packetmemory_reg_f2df	rw	spare1_update	-	-	-	-	-	-	-
0xF2E0	0x00	packetmemory_reg_f2e0	rw	spare_packet_2_header_byte_0[7]	spare_packet_2_header_byte_0[6]	spare_packet_2_header_byte_0[5]	spare_packet_2_header_byte_0[4]	spare_packet_2_header_byte_0[3]	spare_packet_2_header_byte_0[2]	spare_packet_2_header_byte_0[1]	spare_packet_2_header_byte_0[0]
0xF2E1	0x00	packetmemory_reg_f2e1	rw	spare_packet_2_header_byte_1[7]	spare_packet_2_header_byte_1[6]	spare_packet_2_header_byte_1[5]	spare_packet_2_header_byte_1[4]	spare_packet_2_header_byte_1[3]	spare_packet_2_header_byte_1[2]	spare_packet_2_header_byte_1[1]	spare_packet_2_header_byte_1[0]
0xF2E2	0x00	packetmemory_reg_f2e2	rw	spare_packet_2_header_byte_2[7]	spare_packet_2_header_byte_2[6]	spare_packet_2_header_byte_2[5]	spare_packet_2_header_byte_2[4]	spare_packet_2_header_byte_2[3]	spare_packet_2_header_byte_2[2]	spare_packet_2_header_byte_2[1]	spare_packet_2_header_byte_2[0]
0xF2E3	0x00	packetmemory_reg_f2e3	rw	spare_packet_2_packet_byte_0[7]	spare_packet_2_packet_byte_0[6]	spare_packet_2_packet_byte_0[5]	spare_packet_2_packet_byte_0[4]	spare_packet_2_packet_byte_0[3]	spare_packet_2_packet_byte_0[2]	spare_packet_2_packet_byte_0[1]	spare_packet_2_packet_byte_0[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2 E4	0x00	packetmemory_reg_f2e4	rw	spare_packet_2_packet_byte_1[7]	spare_packet_2_packet_byte_1[6]	spare_packet_2_packet_byte_1[5]	spare_packet_2_packet_byte_1[4]	spare_packet_2_packet_byte_1[3]	spare_packet_2_packet_byte_1[2]	spare_packet_2_packet_byte_1[1]	spare_packet_2_packet_byte_1[0]
0xF2 E5	0x00	packetmemory_reg_f2e5	rw	spare_packet_2_packet_byte_2[7]	spare_packet_2_packet_byte_2[6]	spare_packet_2_packet_byte_2[5]	spare_packet_2_packet_byte_2[4]	spare_packet_2_packet_byte_2[3]	spare_packet_2_packet_byte_2[2]	spare_packet_2_packet_byte_2[1]	spare_packet_2_packet_byte_2[0]
0xF2 E6	0x00	packetmemory_reg_f2e6	rw	spare_packet_2_packet_byte_3[7]	spare_packet_2_packet_byte_3[6]	spare_packet_2_packet_byte_3[5]	spare_packet_2_packet_byte_3[4]	spare_packet_2_packet_byte_3[3]	spare_packet_2_packet_byte_3[2]	spare_packet_2_packet_byte_3[1]	spare_packet_2_packet_byte_3[0]
0xF2 E7	0x00	packetmemory_reg_f2e7	rw	spare_packet_2_packet_byte_4[7]	spare_packet_2_packet_byte_4[6]	spare_packet_2_packet_byte_4[5]	spare_packet_2_packet_byte_4[4]	spare_packet_2_packet_byte_4[3]	spare_packet_2_packet_byte_4[2]	spare_packet_2_packet_byte_4[1]	spare_packet_2_packet_byte_4[0]
0xF2 E8	0x00	packetmemory_reg_f2e8	rw	spare_packet_2_packet_byte_5[7]	spare_packet_2_packet_byte_5[6]	spare_packet_2_packet_byte_5[5]	spare_packet_2_packet_byte_5[4]	spare_packet_2_packet_byte_5[3]	spare_packet_2_packet_byte_5[2]	spare_packet_2_packet_byte_5[1]	spare_packet_2_packet_byte_5[0]
0xF2 E9	0x00	packetmemory_reg_f2e9	rw	spare_packet_2_packet_byte_6[7]	spare_packet_2_packet_byte_6[6]	spare_packet_2_packet_byte_6[5]	spare_packet_2_packet_byte_6[4]	spare_packet_2_packet_byte_6[3]	spare_packet_2_packet_byte_6[2]	spare_packet_2_packet_byte_6[1]	spare_packet_2_packet_byte_6[0]
0xF2 EA	0x00	packetmemory_reg_f2ea	rw	spare_packet_2_packet_byte_7[7]	spare_packet_2_packet_byte_7[6]	spare_packet_2_packet_byte_7[5]	spare_packet_2_packet_byte_7[4]	spare_packet_2_packet_byte_7[3]	spare_packet_2_packet_byte_7[2]	spare_packet_2_packet_byte_7[1]	spare_packet_2_packet_byte_7[0]
0xF2 EB	0x00	packetmemory_reg_f2eb	rw	spare_packet_2_packet_byte_8[7]	spare_packet_2_packet_byte_8[6]	spare_packet_2_packet_byte_8[5]	spare_packet_2_packet_byte_8[4]	spare_packet_2_packet_byte_8[3]	spare_packet_2_packet_byte_8[2]	spare_packet_2_packet_byte_8[1]	spare_packet_2_packet_byte_8[0]
0xF2 EC	0x00	packetmemory_reg_f2ec	rw	spare_packet_2_packet_byte_9[7]	spare_packet_2_packet_byte_9[6]	spare_packet_2_packet_byte_9[5]	spare_packet_2_packet_byte_9[4]	spare_packet_2_packet_byte_9[3]	spare_packet_2_packet_byte_9[2]	spare_packet_2_packet_byte_9[1]	spare_packet_2_packet_byte_9[0]
0xF2 ED	0x00	packetmemory_reg_f2ed	rw	spare_packet_2_packet_byte_10[7]	spare_packet_2_packet_byte_10[6]	spare_packet_2_packet_byte_10[5]	spare_packet_2_packet_byte_10[4]	spare_packet_2_packet_byte_10[3]	spare_packet_2_packet_byte_10[2]	spare_packet_2_packet_byte_10[1]	spare_packet_2_packet_byte_10[0]
0xF2 EE	0x00	packetmemory_reg_f2ee	rw	spare_packet_2_packet_byte_11[7]	spare_packet_2_packet_byte_11[6]	spare_packet_2_packet_byte_11[5]	spare_packet_2_packet_byte_11[4]	spare_packet_2_packet_byte_11[3]	spare_packet_2_packet_byte_11[2]	spare_packet_2_packet_byte_11[1]	spare_packet_2_packet_byte_11[0]
0xF2 EF	0x00	packetmemory_reg_f2ef	rw	spare_packet_2_packet_byte_12[7]	spare_packet_2_packet_byte_12[6]	spare_packet_2_packet_byte_12[5]	spare_packet_2_packet_byte_12[4]	spare_packet_2_packet_byte_12[3]	spare_packet_2_packet_byte_12[2]	spare_packet_2_packet_byte_12[1]	spare_packet_2_packet_byte_12[0]
0xF2 F0	0x00	packetmemory_reg_f2f0	rw	spare_packet_2_packet_byte_13[7]	spare_packet_2_packet_byte_13[6]	spare_packet_2_packet_byte_13[5]	spare_packet_2_packet_byte_13[4]	spare_packet_2_packet_byte_13[3]	spare_packet_2_packet_byte_13[2]	spare_packet_2_packet_byte_13[1]	spare_packet_2_packet_byte_13[0]
0xF2 F1	0x00	packetmemory_reg_f2f1	rw	spare_packet_2_packet_byte_14[7]	spare_packet_2_packet_byte_14[6]	spare_packet_2_packet_byte_14[5]	spare_packet_2_packet_byte_14[4]	spare_packet_2_packet_byte_14[3]	spare_packet_2_packet_byte_14[2]	spare_packet_2_packet_byte_14[1]	spare_packet_2_packet_byte_14[0]
0xF2 F2	0x00	packetmemory_reg_f2f2	rw	spare_packet_2_packet_byte_15[7]	spare_packet_2_packet_byte_15[6]	spare_packet_2_packet_byte_15[5]	spare_packet_2_packet_byte_15[4]	spare_packet_2_packet_byte_15[3]	spare_packet_2_packet_byte_15[2]	spare_packet_2_packet_byte_15[1]	spare_packet_2_packet_byte_15[0]
0xF2 F3	0x00	packetmemory_reg_f2f3	rw	spare_packet_2_packet_byte_16[7]	spare_packet_2_packet_byte_16[6]	spare_packet_2_packet_byte_16[5]	spare_packet_2_packet_byte_16[4]	spare_packet_2_packet_byte_16[3]	spare_packet_2_packet_byte_16[2]	spare_packet_2_packet_byte_16[1]	spare_packet_2_packet_byte_16[0]
0xF2 F4	0x00	packetmemory_reg_f2f4	rw	spare_packet_2_packet_byte_17[7]	spare_packet_2_packet_byte_17[6]	spare_packet_2_packet_byte_17[5]	spare_packet_2_packet_byte_17[4]	spare_packet_2_packet_byte_17[3]	spare_packet_2_packet_byte_17[2]	spare_packet_2_packet_byte_17[1]	spare_packet_2_packet_byte_17[0]
0xF2 F5	0x00	packetmemory_reg_f2f5	rw	spare_packet_2_packet_byte_18[7]	spare_packet_2_packet_byte_18[6]	spare_packet_2_packet_byte_18[5]	spare_packet_2_packet_byte_18[4]	spare_packet_2_packet_byte_18[3]	spare_packet_2_packet_byte_18[2]	spare_packet_2_packet_byte_18[1]	spare_packet_2_packet_byte_18[0]
0xF2 F6	0x00	packetmemory_reg_f2f6	rw	spare_packet_2_packet_byte_19[7]	spare_packet_2_packet_byte_19[6]	spare_packet_2_packet_byte_19[5]	spare_packet_2_packet_byte_19[4]	spare_packet_2_packet_byte_19[3]	spare_packet_2_packet_byte_19[2]	spare_packet_2_packet_byte_19[1]	spare_packet_2_packet_byte_19[0]
0xF2 F7	0x00	packetmemory_reg_f2f7	rw	spare_packet_2_packet_byte_20[7]	spare_packet_2_packet_byte_20[6]	spare_packet_2_packet_byte_20[5]	spare_packet_2_packet_byte_20[4]	spare_packet_2_packet_byte_20[3]	spare_packet_2_packet_byte_20[2]	spare_packet_2_packet_byte_20[1]	spare_packet_2_packet_byte_20[0]
0xF2 F8	0x00	packetmemory_reg_f2f8	rw	spare_packet_2_packet_byte_21[7]	spare_packet_2_packet_byte_21[6]	spare_packet_2_packet_byte_21[5]	spare_packet_2_packet_byte_21[4]	spare_packet_2_packet_byte_21[3]	spare_packet_2_packet_byte_21[2]	spare_packet_2_packet_byte_21[1]	spare_packet_2_packet_byte_21[0]
0xF2 F9	0x00	packetmemory_reg_f2f9	rw	spare_packet_2_packet_byte_22[7]	spare_packet_2_packet_byte_22[6]	spare_packet_2_packet_byte_22[5]	spare_packet_2_packet_byte_22[4]	spare_packet_2_packet_byte_22[3]	spare_packet_2_packet_byte_22[2]	spare_packet_2_packet_byte_22[1]	spare_packet_2_packet_byte_22[0]
0xF2 FA	0x00	packetmemory_reg_f2fa	rw	spare_packet_2_packet_byte_23[7]	spare_packet_2_packet_byte_23[6]	spare_packet_2_packet_byte_23[5]	spare_packet_2_packet_byte_23[4]	spare_packet_2_packet_byte_23[3]	spare_packet_2_packet_byte_23[2]	spare_packet_2_packet_byte_23[1]	spare_packet_2_packet_byte_23[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF2 FB	0x00	packetmemory_reg_f2fb	rw	spare_packet_2_packet_byte_24[7]	spare_packet_2_packet_byte_24[6]	spare_packet_2_packet_byte_24[5]	spare_packet_2_packet_byte_24[4]	spare_packet_2_packet_byte_24[3]	spare_packet_2_packet_byte_24[2]	spare_packet_2_packet_byte_24[1]	spare_packet_2_packet_byte_24[0]
0xF2 FC	0x00	packetmemory_reg_f2fc	rw	spare_packet_2_packet_byte_25[7]	spare_packet_2_packet_byte_25[6]	spare_packet_2_packet_byte_25[5]	spare_packet_2_packet_byte_25[4]	spare_packet_2_packet_byte_25[3]	spare_packet_2_packet_byte_25[2]	spare_packet_2_packet_byte_25[1]	spare_packet_2_packet_byte_25[0]
0xF2 FD	0x00	packetmemory_reg_f2fd	rw	spare_packet_2_packet_byte_26[7]	spare_packet_2_packet_byte_26[6]	spare_packet_2_packet_byte_26[5]	spare_packet_2_packet_byte_26[4]	spare_packet_2_packet_byte_26[3]	spare_packet_2_packet_byte_26[2]	spare_packet_2_packet_byte_26[1]	spare_packet_2_packet_byte_26[0]
0xF2 FE	0x00	packetmemory_reg_f2fe	rw	spare_packet_2_packet_byte_27[7]	spare_packet_2_packet_byte_27[6]	spare_packet_2_packet_byte_27[5]	spare_packet_2_packet_byte_27[4]	spare_packet_2_packet_byte_27[3]	spare_packet_2_packet_byte_27[2]	spare_packet_2_packet_byte_27[1]	spare_packet_2_packet_byte_27[0]
0xF2 FF	0x00	packetmemory_reg_f2ff	rw	spare2_update	-	-	-	-	-	-	-

1.9 TX1 EDID MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 00	0x00	edidmemory_reg_ee00	r	edid_0[7]	edid_0[6]	edid_0[5]	edid_0[4]	edid_0[3]	edid_0[2]	edid_0[1]	edid_0[0]
0xEE 01	0x00	edidmemory_reg_ee01	r	edid_1[7]	edid_1[6]	edid_1[5]	edid_1[4]	edid_1[3]	edid_1[2]	edid_1[1]	edid_1[0]
0xEE 02	0x00	edidmemory_reg_ee02	r	edid_2[7]	edid_2[6]	edid_2[5]	edid_2[4]	edid_2[3]	edid_2[2]	edid_2[1]	edid_2[0]
0xEE 03	0x00	edidmemory_reg_ee03	r	edid_3[7]	edid_3[6]	edid_3[5]	edid_3[4]	edid_3[3]	edid_3[2]	edid_3[1]	edid_3[0]
0xEE 04	0x00	edidmemory_reg_ee04	r	edid_4[7]	edid_4[6]	edid_4[5]	edid_4[4]	edid_4[3]	edid_4[2]	edid_4[1]	edid_4[0]
0xEE 05	0x00	edidmemory_reg_ee05	r	edid_5[7]	edid_5[6]	edid_5[5]	edid_5[4]	edid_5[3]	edid_5[2]	edid_5[1]	edid_5[0]
0xEE 06	0x00	edidmemory_reg_ee06	r	edid_6[7]	edid_6[6]	edid_6[5]	edid_6[4]	edid_6[3]	edid_6[2]	edid_6[1]	edid_6[0]
0xEE 07	0x00	edidmemory_reg_ee07	r	edid_7[7]	edid_7[6]	edid_7[5]	edid_7[4]	edid_7[3]	edid_7[2]	edid_7[1]	edid_7[0]
0xEE 08	0x00	edidmemory_reg_ee08	r	edid_8[7]	edid_8[6]	edid_8[5]	edid_8[4]	edid_8[3]	edid_8[2]	edid_8[1]	edid_8[0]
0xEE 09	0x00	edidmemory_reg_ee09	r	edid_9[7]	edid_9[6]	edid_9[5]	edid_9[4]	edid_9[3]	edid_9[2]	edid_9[1]	edid_9[0]
0xEE 0A	0x00	edidmemory_reg_ee0a	r	edid_10[7]	edid_10[6]	edid_10[5]	edid_10[4]	edid_10[3]	edid_10[2]	edid_10[1]	edid_10[0]
0xEE 0B	0x00	edidmemory_reg_ee0b	r	edid_11[7]	edid_11[6]	edid_11[5]	edid_11[4]	edid_11[3]	edid_11[2]	edid_11[1]	edid_11[0]
0xEE 0C	0x00	edidmemory_reg_ee0c	r	edid_12[7]	edid_12[6]	edid_12[5]	edid_12[4]	edid_12[3]	edid_12[2]	edid_12[1]	edid_12[0]
0xEE 0D	0x00	edidmemory_reg_ee0d	r	edid_13[7]	edid_13[6]	edid_13[5]	edid_13[4]	edid_13[3]	edid_13[2]	edid_13[1]	edid_13[0]
0xEE 0E	0x00	edidmemory_reg_ee0e	r	edid_14[7]	edid_14[6]	edid_14[5]	edid_14[4]	edid_14[3]	edid_14[2]	edid_14[1]	edid_14[0]
0xEE 0F	0x00	edidmemory_reg_ee0f	r	edid_15[7]	edid_15[6]	edid_15[5]	edid_15[4]	edid_15[3]	edid_15[2]	edid_15[1]	edid_15[0]
0xEE 10	0x00	edidmemory_reg_ee10	r	edid_16[7]	edid_16[6]	edid_16[5]	edid_16[4]	edid_16[3]	edid_16[2]	edid_16[1]	edid_16[0]
0xEE 11	0x00	edidmemory_reg_ee11	r	edid_17[7]	edid_17[6]	edid_17[5]	edid_17[4]	edid_17[3]	edid_17[2]	edid_17[1]	edid_17[0]
0xEE 12	0x00	edidmemory_reg_ee12	r	edid_18[7]	edid_18[6]	edid_18[5]	edid_18[4]	edid_18[3]	edid_18[2]	edid_18[1]	edid_18[0]
0xEE 13	0x00	edidmemory_reg_ee13	r	edid_19[7]	edid_19[6]	edid_19[5]	edid_19[4]	edid_19[3]	edid_19[2]	edid_19[1]	edid_19[0]
0xEE 14	0x00	edidmemory_reg_ee14	r	edid_20[7]	edid_20[6]	edid_20[5]	edid_20[4]	edid_20[3]	edid_20[2]	edid_20[1]	edid_20[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 15	0x00	edidmemory_reg_ee15	r	edid_21[7]	edid_21[6]	edid_21[5]	edid_21[4]	edid_21[3]	edid_21[2]	edid_21[1]	edid_21[0]
0xEE 16	0x00	edidmemory_reg_ee16	r	edid_22[7]	edid_22[6]	edid_22[5]	edid_22[4]	edid_22[3]	edid_22[2]	edid_22[1]	edid_22[0]
0xEE 17	0x00	edidmemory_reg_ee17	r	edid_23[7]	edid_23[6]	edid_23[5]	edid_23[4]	edid_23[3]	edid_23[2]	edid_23[1]	edid_23[0]
0xEE 18	0x00	edidmemory_reg_ee18	r	edid_24[7]	edid_24[6]	edid_24[5]	edid_24[4]	edid_24[3]	edid_24[2]	edid_24[1]	edid_24[0]
0xEE 19	0x00	edidmemory_reg_ee19	r	edid_25[7]	edid_25[6]	edid_25[5]	edid_25[4]	edid_25[3]	edid_25[2]	edid_25[1]	edid_25[0]
0xEE 1A	0x00	edidmemory_reg_ee1a	r	edid_26[7]	edid_26[6]	edid_26[5]	edid_26[4]	edid_26[3]	edid_26[2]	edid_26[1]	edid_26[0]
0xEE 1B	0x00	edidmemory_reg_ee1b	r	edid_27[7]	edid_27[6]	edid_27[5]	edid_27[4]	edid_27[3]	edid_27[2]	edid_27[1]	edid_27[0]
0xEE 1C	0x00	edidmemory_reg_ee1c	r	edid_28[7]	edid_28[6]	edid_28[5]	edid_28[4]	edid_28[3]	edid_28[2]	edid_28[1]	edid_28[0]
0xEE 1D	0x00	edidmemory_reg_ee1d	r	edid_29[7]	edid_29[6]	edid_29[5]	edid_29[4]	edid_29[3]	edid_29[2]	edid_29[1]	edid_29[0]
0xEE 1E	0x00	edidmemory_reg_ee1e	r	edid_30[7]	edid_30[6]	edid_30[5]	edid_30[4]	edid_30[3]	edid_30[2]	edid_30[1]	edid_30[0]
0xEE 1F	0x00	edidmemory_reg_ee1f	r	edid_31[7]	edid_31[6]	edid_31[5]	edid_31[4]	edid_31[3]	edid_31[2]	edid_31[1]	edid_31[0]
0xEE 20	0x00	edidmemory_reg_ee20	r	edid_32[7]	edid_32[6]	edid_32[5]	edid_32[4]	edid_32[3]	edid_32[2]	edid_32[1]	edid_32[0]
0xEE 21	0x00	edidmemory_reg_ee21	r	edid_33[7]	edid_33[6]	edid_33[5]	edid_33[4]	edid_33[3]	edid_33[2]	edid_33[1]	edid_33[0]
0xEE 22	0x00	edidmemory_reg_ee22	r	edid_34[7]	edid_34[6]	edid_34[5]	edid_34[4]	edid_34[3]	edid_34[2]	edid_34[1]	edid_34[0]
0xEE 23	0x00	edidmemory_reg_ee23	r	edid_35[7]	edid_35[6]	edid_35[5]	edid_35[4]	edid_35[3]	edid_35[2]	edid_35[1]	edid_35[0]
0xEE 24	0x00	edidmemory_reg_ee24	r	edid_36[7]	edid_36[6]	edid_36[5]	edid_36[4]	edid_36[3]	edid_36[2]	edid_36[1]	edid_36[0]
0xEE 25	0x00	edidmemory_reg_ee25	r	edid_37[7]	edid_37[6]	edid_37[5]	edid_37[4]	edid_37[3]	edid_37[2]	edid_37[1]	edid_37[0]
0xEE 26	0x00	edidmemory_reg_ee26	r	edid_38[7]	edid_38[6]	edid_38[5]	edid_38[4]	edid_38[3]	edid_38[2]	edid_38[1]	edid_38[0]
0xEE 27	0x00	edidmemory_reg_ee27	r	edid_39[7]	edid_39[6]	edid_39[5]	edid_39[4]	edid_39[3]	edid_39[2]	edid_39[1]	edid_39[0]
0xEE 28	0x00	edidmemory_reg_ee28	r	edid_40[7]	edid_40[6]	edid_40[5]	edid_40[4]	edid_40[3]	edid_40[2]	edid_40[1]	edid_40[0]
0xEE 29	0x00	edidmemory_reg_ee29	r	edid_41[7]	edid_41[6]	edid_41[5]	edid_41[4]	edid_41[3]	edid_41[2]	edid_41[1]	edid_41[0]
0xEE 2A	0x00	edidmemory_reg_ee2a	r	edid_42[7]	edid_42[6]	edid_42[5]	edid_42[4]	edid_42[3]	edid_42[2]	edid_42[1]	edid_42[0]
0xEE 2B	0x00	edidmemory_reg_ee2b	r	edid_43[7]	edid_43[6]	edid_43[5]	edid_43[4]	edid_43[3]	edid_43[2]	edid_43[1]	edid_43[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 2C	0x00	edidmemory_reg_ee2c	r	edid_44[7]	edid_44[6]	edid_44[5]	edid_44[4]	edid_44[3]	edid_44[2]	edid_44[1]	edid_44[0]
0xEE 2D	0x00	edidmemory_reg_ee2d	r	edid_45[7]	edid_45[6]	edid_45[5]	edid_45[4]	edid_45[3]	edid_45[2]	edid_45[1]	edid_45[0]
0xEE 2E	0x00	edidmemory_reg_ee2e	r	edid_46[7]	edid_46[6]	edid_46[5]	edid_46[4]	edid_46[3]	edid_46[2]	edid_46[1]	edid_46[0]
0xEE 2F	0x00	edidmemory_reg_ee2f	r	edid_47[7]	edid_47[6]	edid_47[5]	edid_47[4]	edid_47[3]	edid_47[2]	edid_47[1]	edid_47[0]
0xEE 30	0x00	edidmemory_reg_ee30	r	edid_48[7]	edid_48[6]	edid_48[5]	edid_48[4]	edid_48[3]	edid_48[2]	edid_48[1]	edid_48[0]
0xEE 31	0x00	edidmemory_reg_ee31	r	edid_49[7]	edid_49[6]	edid_49[5]	edid_49[4]	edid_49[3]	edid_49[2]	edid_49[1]	edid_49[0]
0xEE 32	0x00	edidmemory_reg_ee32	r	edid_50[7]	edid_50[6]	edid_50[5]	edid_50[4]	edid_50[3]	edid_50[2]	edid_50[1]	edid_50[0]
0xEE 33	0x00	edidmemory_reg_ee33	r	edid_51[7]	edid_51[6]	edid_51[5]	edid_51[4]	edid_51[3]	edid_51[2]	edid_51[1]	edid_51[0]
0xEE 34	0x00	edidmemory_reg_ee34	r	edid_52[7]	edid_52[6]	edid_52[5]	edid_52[4]	edid_52[3]	edid_52[2]	edid_52[1]	edid_52[0]
0xEE 35	0x00	edidmemory_reg_ee35	r	edid_53[7]	edid_53[6]	edid_53[5]	edid_53[4]	edid_53[3]	edid_53[2]	edid_53[1]	edid_53[0]
0xEE 36	0x00	edidmemory_reg_ee36	r	edid_54[7]	edid_54[6]	edid_54[5]	edid_54[4]	edid_54[3]	edid_54[2]	edid_54[1]	edid_54[0]
0xEE 37	0x00	edidmemory_reg_ee37	r	edid_55[7]	edid_55[6]	edid_55[5]	edid_55[4]	edid_55[3]	edid_55[2]	edid_55[1]	edid_55[0]
0xEE 38	0x00	edidmemory_reg_ee38	r	edid_56[7]	edid_56[6]	edid_56[5]	edid_56[4]	edid_56[3]	edid_56[2]	edid_56[1]	edid_56[0]
0xEE 39	0x00	edidmemory_reg_ee39	r	edid_57[7]	edid_57[6]	edid_57[5]	edid_57[4]	edid_57[3]	edid_57[2]	edid_57[1]	edid_57[0]
0xEE 3A	0x00	edidmemory_reg_ee3a	r	edid_58[7]	edid_58[6]	edid_58[5]	edid_58[4]	edid_58[3]	edid_58[2]	edid_58[1]	edid_58[0]
0xEE 3B	0x00	edidmemory_reg_ee3b	r	edid_59[7]	edid_59[6]	edid_59[5]	edid_59[4]	edid_59[3]	edid_59[2]	edid_59[1]	edid_59[0]
0xEE 3C	0x00	edidmemory_reg_ee3c	r	edid_60[7]	edid_60[6]	edid_60[5]	edid_60[4]	edid_60[3]	edid_60[2]	edid_60[1]	edid_60[0]
0xEE 3D	0x00	edidmemory_reg_ee3d	r	edid_61[7]	edid_61[6]	edid_61[5]	edid_61[4]	edid_61[3]	edid_61[2]	edid_61[1]	edid_61[0]
0xEE 3E	0x00	edidmemory_reg_ee3e	r	edid_62[7]	edid_62[6]	edid_62[5]	edid_62[4]	edid_62[3]	edid_62[2]	edid_62[1]	edid_62[0]
0xEE 3F	0x00	edidmemory_reg_ee3f	r	edid_63[7]	edid_63[6]	edid_63[5]	edid_63[4]	edid_63[3]	edid_63[2]	edid_63[1]	edid_63[0]
0xEE 40	0x00	edidmemory_reg_ee40	r	edid_64[7]	edid_64[6]	edid_64[5]	edid_64[4]	edid_64[3]	edid_64[2]	edid_64[1]	edid_64[0]
0xEE 41	0x00	edidmemory_reg_ee41	r	edid_65[7]	edid_65[6]	edid_65[5]	edid_65[4]	edid_65[3]	edid_65[2]	edid_65[1]	edid_65[0]
0xEE 42	0x00	edidmemory_reg_ee42	r	edid_66[7]	edid_66[6]	edid_66[5]	edid_66[4]	edid_66[3]	edid_66[2]	edid_66[1]	edid_66[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 43	0x00	edidmemory_reg_ee43	r	edid_67[7]	edid_67[6]	edid_67[5]	edid_67[4]	edid_67[3]	edid_67[2]	edid_67[1]	edid_67[0]
0xEE 44	0x00	edidmemory_reg_ee44	r	edid_68[7]	edid_68[6]	edid_68[5]	edid_68[4]	edid_68[3]	edid_68[2]	edid_68[1]	edid_68[0]
0xEE 45	0x00	edidmemory_reg_ee45	r	edid_69[7]	edid_69[6]	edid_69[5]	edid_69[4]	edid_69[3]	edid_69[2]	edid_69[1]	edid_69[0]
0xEE 46	0x00	edidmemory_reg_ee46	r	edid_70[7]	edid_70[6]	edid_70[5]	edid_70[4]	edid_70[3]	edid_70[2]	edid_70[1]	edid_70[0]
0xEE 47	0x00	edidmemory_reg_ee47	r	edid_71[7]	edid_71[6]	edid_71[5]	edid_71[4]	edid_71[3]	edid_71[2]	edid_71[1]	edid_71[0]
0xEE 48	0x00	edidmemory_reg_ee48	r	edid_72[7]	edid_72[6]	edid_72[5]	edid_72[4]	edid_72[3]	edid_72[2]	edid_72[1]	edid_72[0]
0xEE 49	0x00	edidmemory_reg_ee49	r	edid_73[7]	edid_73[6]	edid_73[5]	edid_73[4]	edid_73[3]	edid_73[2]	edid_73[1]	edid_73[0]
0xEE 4A	0x00	edidmemory_reg_ee4a	r	edid_74[7]	edid_74[6]	edid_74[5]	edid_74[4]	edid_74[3]	edid_74[2]	edid_74[1]	edid_74[0]
0xEE 4B	0x00	edidmemory_reg_ee4b	r	edid_75[7]	edid_75[6]	edid_75[5]	edid_75[4]	edid_75[3]	edid_75[2]	edid_75[1]	edid_75[0]
0xEE 4C	0x00	edidmemory_reg_ee4c	r	edid_76[7]	edid_76[6]	edid_76[5]	edid_76[4]	edid_76[3]	edid_76[2]	edid_76[1]	edid_76[0]
0xEE 4D	0x00	edidmemory_reg_ee4d	r	edid_77[7]	edid_77[6]	edid_77[5]	edid_77[4]	edid_77[3]	edid_77[2]	edid_77[1]	edid_77[0]
0xEE 4E	0x00	edidmemory_reg_ee4e	r	edid_78[7]	edid_78[6]	edid_78[5]	edid_78[4]	edid_78[3]	edid_78[2]	edid_78[1]	edid_78[0]
0xEE 4F	0x00	edidmemory_reg_ee4f	r	edid_79[7]	edid_79[6]	edid_79[5]	edid_79[4]	edid_79[3]	edid_79[2]	edid_79[1]	edid_79[0]
0xEE 50	0x00	edidmemory_reg_ee50	r	edid_80[7]	edid_80[6]	edid_80[5]	edid_80[4]	edid_80[3]	edid_80[2]	edid_80[1]	edid_80[0]
0xEE 51	0x00	edidmemory_reg_ee51	r	edid_81[7]	edid_81[6]	edid_81[5]	edid_81[4]	edid_81[3]	edid_81[2]	edid_81[1]	edid_81[0]
0xEE 52	0x00	edidmemory_reg_ee52	r	edid_82[7]	edid_82[6]	edid_82[5]	edid_82[4]	edid_82[3]	edid_82[2]	edid_82[1]	edid_82[0]
0xEE 53	0x00	edidmemory_reg_ee53	r	edid_83[7]	edid_83[6]	edid_83[5]	edid_83[4]	edid_83[3]	edid_83[2]	edid_83[1]	edid_83[0]
0xEE 54	0x00	edidmemory_reg_ee54	r	edid_84[7]	edid_84[6]	edid_84[5]	edid_84[4]	edid_84[3]	edid_84[2]	edid_84[1]	edid_84[0]
0xEE 55	0x00	edidmemory_reg_ee55	r	edid_85[7]	edid_85[6]	edid_85[5]	edid_85[4]	edid_85[3]	edid_85[2]	edid_85[1]	edid_85[0]
0xEE 56	0x00	edidmemory_reg_ee56	r	edid_86[7]	edid_86[6]	edid_86[5]	edid_86[4]	edid_86[3]	edid_86[2]	edid_86[1]	edid_86[0]
0xEE 57	0x00	edidmemory_reg_ee57	r	edid_87[7]	edid_87[6]	edid_87[5]	edid_87[4]	edid_87[3]	edid_87[2]	edid_87[1]	edid_87[0]
0xEE 58	0x00	edidmemory_reg_ee58	r	edid_88[7]	edid_88[6]	edid_88[5]	edid_88[4]	edid_88[3]	edid_88[2]	edid_88[1]	edid_88[0]
0xEE 59	0x00	edidmemory_reg_ee59	r	edid_89[7]	edid_89[6]	edid_89[5]	edid_89[4]	edid_89[3]	edid_89[2]	edid_89[1]	edid_89[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 5A	0x00	edidmemory_reg_ee5a	r	edid_90[7]	edid_90[6]	edid_90[5]	edid_90[4]	edid_90[3]	edid_90[2]	edid_90[1]	edid_90[0]
0xEE 5B	0x00	edidmemory_reg_ee5b	r	edid_91[7]	edid_91[6]	edid_91[5]	edid_91[4]	edid_91[3]	edid_91[2]	edid_91[1]	edid_91[0]
0xEE 5C	0x00	edidmemory_reg_ee5c	r	edid_92[7]	edid_92[6]	edid_92[5]	edid_92[4]	edid_92[3]	edid_92[2]	edid_92[1]	edid_92[0]
0xEE 5D	0x00	edidmemory_reg_ee5d	r	edid_93[7]	edid_93[6]	edid_93[5]	edid_93[4]	edid_93[3]	edid_93[2]	edid_93[1]	edid_93[0]
0xEE 5E	0x00	edidmemory_reg_ee5e	r	edid_94[7]	edid_94[6]	edid_94[5]	edid_94[4]	edid_94[3]	edid_94[2]	edid_94[1]	edid_94[0]
0xEE 5F	0x00	edidmemory_reg_ee5f	r	edid_95[7]	edid_95[6]	edid_95[5]	edid_95[4]	edid_95[3]	edid_95[2]	edid_95[1]	edid_95[0]
0xEE 60	0x00	edidmemory_reg_ee60	r	edid_96[7]	edid_96[6]	edid_96[5]	edid_96[4]	edid_96[3]	edid_96[2]	edid_96[1]	edid_96[0]
0xEE 61	0x00	edidmemory_reg_ee61	r	edid_97[7]	edid_97[6]	edid_97[5]	edid_97[4]	edid_97[3]	edid_97[2]	edid_97[1]	edid_97[0]
0xEE 62	0x00	edidmemory_reg_ee62	r	edid_98[7]	edid_98[6]	edid_98[5]	edid_98[4]	edid_98[3]	edid_98[2]	edid_98[1]	edid_98[0]
0xEE 63	0x00	edidmemory_reg_ee63	r	edid_99[7]	edid_99[6]	edid_99[5]	edid_99[4]	edid_99[3]	edid_99[2]	edid_99[1]	edid_99[0]
0xEE 64	0x00	edidmemory_reg_ee64	r	edid_100[7]	edid_100[6]	edid_100[5]	edid_100[4]	edid_100[3]	edid_100[2]	edid_100[1]	edid_100[0]
0xEE 65	0x00	edidmemory_reg_ee65	r	edid_101[7]	edid_101[6]	edid_101[5]	edid_101[4]	edid_101[3]	edid_101[2]	edid_101[1]	edid_101[0]
0xEE 66	0x00	edidmemory_reg_ee66	r	edid_102[7]	edid_102[6]	edid_102[5]	edid_102[4]	edid_102[3]	edid_102[2]	edid_102[1]	edid_102[0]
0xEE 67	0x00	edidmemory_reg_ee67	r	edid_103[7]	edid_103[6]	edid_103[5]	edid_103[4]	edid_103[3]	edid_103[2]	edid_103[1]	edid_103[0]
0xEE 68	0x00	edidmemory_reg_ee68	r	edid_104[7]	edid_104[6]	edid_104[5]	edid_104[4]	edid_104[3]	edid_104[2]	edid_104[1]	edid_104[0]
0xEE 69	0x00	edidmemory_reg_ee69	r	edid_105[7]	edid_105[6]	edid_105[5]	edid_105[4]	edid_105[3]	edid_105[2]	edid_105[1]	edid_105[0]
0xEE 6A	0x00	edidmemory_reg_ee6a	r	edid_106[7]	edid_106[6]	edid_106[5]	edid_106[4]	edid_106[3]	edid_106[2]	edid_106[1]	edid_106[0]
0xEE 6B	0x00	edidmemory_reg_ee6b	r	edid_107[7]	edid_107[6]	edid_107[5]	edid_107[4]	edid_107[3]	edid_107[2]	edid_107[1]	edid_107[0]
0xEE 6C	0x00	edidmemory_reg_ee6c	r	edid_108[7]	edid_108[6]	edid_108[5]	edid_108[4]	edid_108[3]	edid_108[2]	edid_108[1]	edid_108[0]
0xEE 6D	0x00	edidmemory_reg_ee6d	r	edid_109[7]	edid_109[6]	edid_109[5]	edid_109[4]	edid_109[3]	edid_109[2]	edid_109[1]	edid_109[0]
0xEE 6E	0x00	edidmemory_reg_ee6e	r	edid_110[7]	edid_110[6]	edid_110[5]	edid_110[4]	edid_110[3]	edid_110[2]	edid_110[1]	edid_110[0]
0xEE 6F	0x00	edidmemory_reg_ee6f	r	edid_111[7]	edid_111[6]	edid_111[5]	edid_111[4]	edid_111[3]	edid_111[2]	edid_111[1]	edid_111[0]
0xEE 70	0x00	edidmemory_reg_ee70	r	edid_112[7]	edid_112[6]	edid_112[5]	edid_112[4]	edid_112[3]	edid_112[2]	edid_112[1]	edid_112[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 71	0x00	edidmemory_reg_ee71	r	edid_113[7]	edid_113[6]	edid_113[5]	edid_113[4]	edid_113[3]	edid_113[2]	edid_113[1]	edid_113[0]
0xEE 72	0x00	edidmemory_reg_ee72	r	edid_114[7]	edid_114[6]	edid_114[5]	edid_114[4]	edid_114[3]	edid_114[2]	edid_114[1]	edid_114[0]
0xEE 73	0x00	edidmemory_reg_ee73	r	edid_115[7]	edid_115[6]	edid_115[5]	edid_115[4]	edid_115[3]	edid_115[2]	edid_115[1]	edid_115[0]
0xEE 74	0x00	edidmemory_reg_ee74	r	edid_116[7]	edid_116[6]	edid_116[5]	edid_116[4]	edid_116[3]	edid_116[2]	edid_116[1]	edid_116[0]
0xEE 75	0x00	edidmemory_reg_ee75	r	edid_117[7]	edid_117[6]	edid_117[5]	edid_117[4]	edid_117[3]	edid_117[2]	edid_117[1]	edid_117[0]
0xEE 76	0x00	edidmemory_reg_ee76	r	edid_118[7]	edid_118[6]	edid_118[5]	edid_118[4]	edid_118[3]	edid_118[2]	edid_118[1]	edid_118[0]
0xEE 77	0x00	edidmemory_reg_ee77	r	edid_119[7]	edid_119[6]	edid_119[5]	edid_119[4]	edid_119[3]	edid_119[2]	edid_119[1]	edid_119[0]
0xEE 78	0x00	edidmemory_reg_ee78	r	edid_120[7]	edid_120[6]	edid_120[5]	edid_120[4]	edid_120[3]	edid_120[2]	edid_120[1]	edid_120[0]
0xEE 79	0x00	edidmemory_reg_ee79	r	edid_121[7]	edid_121[6]	edid_121[5]	edid_121[4]	edid_121[3]	edid_121[2]	edid_121[1]	edid_121[0]
0xEE 7A	0x00	edidmemory_reg_ee7a	r	edid_122[7]	edid_122[6]	edid_122[5]	edid_122[4]	edid_122[3]	edid_122[2]	edid_122[1]	edid_122[0]
0xEE 7B	0x00	edidmemory_reg_ee7b	r	edid_123[7]	edid_123[6]	edid_123[5]	edid_123[4]	edid_123[3]	edid_123[2]	edid_123[1]	edid_123[0]
0xEE 7C	0x00	edidmemory_reg_ee7c	r	edid_124[7]	edid_124[6]	edid_124[5]	edid_124[4]	edid_124[3]	edid_124[2]	edid_124[1]	edid_124[0]
0xEE 7D	0x00	edidmemory_reg_ee7d	r	edid_125[7]	edid_125[6]	edid_125[5]	edid_125[4]	edid_125[3]	edid_125[2]	edid_125[1]	edid_125[0]
0xEE 7E	0x00	edidmemory_reg_ee7e	r	edid_126[7]	edid_126[6]	edid_126[5]	edid_126[4]	edid_126[3]	edid_126[2]	edid_126[1]	edid_126[0]
0xEE 7F	0x00	edidmemory_reg_ee7f	r	edid_127[7]	edid_127[6]	edid_127[5]	edid_127[4]	edid_127[3]	edid_127[2]	edid_127[1]	edid_127[0]
0xEE 80	0x00	edidmemory_reg_ee80	r	edid_128[7]	edid_128[6]	edid_128[5]	edid_128[4]	edid_128[3]	edid_128[2]	edid_128[1]	edid_128[0]
0xEE 81	0x00	edidmemory_reg_ee81	r	edid_129[7]	edid_129[6]	edid_129[5]	edid_129[4]	edid_129[3]	edid_129[2]	edid_129[1]	edid_129[0]
0xEE 82	0x00	edidmemory_reg_ee82	r	edid_130[7]	edid_130[6]	edid_130[5]	edid_130[4]	edid_130[3]	edid_130[2]	edid_130[1]	edid_130[0]
0xEE 83	0x00	edidmemory_reg_ee83	r	edid_131[7]	edid_131[6]	edid_131[5]	edid_131[4]	edid_131[3]	edid_131[2]	edid_131[1]	edid_131[0]
0xEE 84	0x00	edidmemory_reg_ee84	r	edid_132[7]	edid_132[6]	edid_132[5]	edid_132[4]	edid_132[3]	edid_132[2]	edid_132[1]	edid_132[0]
0xEE 85	0x00	edidmemory_reg_ee85	r	edid_133[7]	edid_133[6]	edid_133[5]	edid_133[4]	edid_133[3]	edid_133[2]	edid_133[1]	edid_133[0]
0xEE 86	0x00	edidmemory_reg_ee86	r	edid_134[7]	edid_134[6]	edid_134[5]	edid_134[4]	edid_134[3]	edid_134[2]	edid_134[1]	edid_134[0]
0xEE 87	0x00	edidmemory_reg_ee87	r	edid_135[7]	edid_135[6]	edid_135[5]	edid_135[4]	edid_135[3]	edid_135[2]	edid_135[1]	edid_135[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 88	0x00	edidmemory_reg_ee88	r	edid_136[7]	edid_136[6]	edid_136[5]	edid_136[4]	edid_136[3]	edid_136[2]	edid_136[1]	edid_136[0]
0xEE 89	0x00	edidmemory_reg_ee89	r	edid_137[7]	edid_137[6]	edid_137[5]	edid_137[4]	edid_137[3]	edid_137[2]	edid_137[1]	edid_137[0]
0xEE 8A	0x00	edidmemory_reg_ee8a	r	edid_138[7]	edid_138[6]	edid_138[5]	edid_138[4]	edid_138[3]	edid_138[2]	edid_138[1]	edid_138[0]
0xEE 8B	0x00	edidmemory_reg_ee8b	r	edid_139[7]	edid_139[6]	edid_139[5]	edid_139[4]	edid_139[3]	edid_139[2]	edid_139[1]	edid_139[0]
0xEE 8C	0x00	edidmemory_reg_ee8c	r	edid_140[7]	edid_140[6]	edid_140[5]	edid_140[4]	edid_140[3]	edid_140[2]	edid_140[1]	edid_140[0]
0xEE 8D	0x00	edidmemory_reg_ee8d	r	edid_141[7]	edid_141[6]	edid_141[5]	edid_141[4]	edid_141[3]	edid_141[2]	edid_141[1]	edid_141[0]
0xEE 8E	0x00	edidmemory_reg_ee8e	r	edid_142[7]	edid_142[6]	edid_142[5]	edid_142[4]	edid_142[3]	edid_142[2]	edid_142[1]	edid_142[0]
0xEE 8F	0x00	edidmemory_reg_ee8f	r	edid_143[7]	edid_143[6]	edid_143[5]	edid_143[4]	edid_143[3]	edid_143[2]	edid_143[1]	edid_143[0]
0xEE 90	0x00	edidmemory_reg_ee90	r	edid_144[7]	edid_144[6]	edid_144[5]	edid_144[4]	edid_144[3]	edid_144[2]	edid_144[1]	edid_144[0]
0xEE 91	0x00	edidmemory_reg_ee91	r	edid_145[7]	edid_145[6]	edid_145[5]	edid_145[4]	edid_145[3]	edid_145[2]	edid_145[1]	edid_145[0]
0xEE 92	0x00	edidmemory_reg_ee92	r	edid_146[7]	edid_146[6]	edid_146[5]	edid_146[4]	edid_146[3]	edid_146[2]	edid_146[1]	edid_146[0]
0xEE 93	0x00	edidmemory_reg_ee93	r	edid_147[7]	edid_147[6]	edid_147[5]	edid_147[4]	edid_147[3]	edid_147[2]	edid_147[1]	edid_147[0]
0xEE 94	0x00	edidmemory_reg_ee94	r	edid_148[7]	edid_148[6]	edid_148[5]	edid_148[4]	edid_148[3]	edid_148[2]	edid_148[1]	edid_148[0]
0xEE 95	0x00	edidmemory_reg_ee95	r	edid_149[7]	edid_149[6]	edid_149[5]	edid_149[4]	edid_149[3]	edid_149[2]	edid_149[1]	edid_149[0]
0xEE 96	0x00	edidmemory_reg_ee96	r	edid_150[7]	edid_150[6]	edid_150[5]	edid_150[4]	edid_150[3]	edid_150[2]	edid_150[1]	edid_150[0]
0xEE 97	0x00	edidmemory_reg_ee97	r	edid_151[7]	edid_151[6]	edid_151[5]	edid_151[4]	edid_151[3]	edid_151[2]	edid_151[1]	edid_151[0]
0xEE 98	0x00	edidmemory_reg_ee98	r	edid_152[7]	edid_152[6]	edid_152[5]	edid_152[4]	edid_152[3]	edid_152[2]	edid_152[1]	edid_152[0]
0xEE 99	0x00	edidmemory_reg_ee99	r	edid_153[7]	edid_153[6]	edid_153[5]	edid_153[4]	edid_153[3]	edid_153[2]	edid_153[1]	edid_153[0]
0xEE 9A	0x00	edidmemory_reg_ee9a	r	edid_154[7]	edid_154[6]	edid_154[5]	edid_154[4]	edid_154[3]	edid_154[2]	edid_154[1]	edid_154[0]
0xEE 9B	0x00	edidmemory_reg_ee9b	r	edid_155[7]	edid_155[6]	edid_155[5]	edid_155[4]	edid_155[3]	edid_155[2]	edid_155[1]	edid_155[0]
0xEE 9C	0x00	edidmemory_reg_ee9c	r	edid_156[7]	edid_156[6]	edid_156[5]	edid_156[4]	edid_156[3]	edid_156[2]	edid_156[1]	edid_156[0]
0xEE 9D	0x00	edidmemory_reg_ee9d	r	edid_157[7]	edid_157[6]	edid_157[5]	edid_157[4]	edid_157[3]	edid_157[2]	edid_157[1]	edid_157[0]
0xEE 9E	0x00	edidmemory_reg_ee9e	r	edid_158[7]	edid_158[6]	edid_158[5]	edid_158[4]	edid_158[3]	edid_158[2]	edid_158[1]	edid_158[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE 9F	0x00	edidmemory_reg_ee9f	r	edid_159[7]	edid_159[6]	edid_159[5]	edid_159[4]	edid_159[3]	edid_159[2]	edid_159[1]	edid_159[0]
0xEE A0	0x00	edidmemory_reg_eea0	r	edid_160[7]	edid_160[6]	edid_160[5]	edid_160[4]	edid_160[3]	edid_160[2]	edid_160[1]	edid_160[0]
0xEE A1	0x00	edidmemory_reg_eea1	r	edid_161[7]	edid_161[6]	edid_161[5]	edid_161[4]	edid_161[3]	edid_161[2]	edid_161[1]	edid_161[0]
0xEE A2	0x00	edidmemory_reg_eea2	r	edid_162[7]	edid_162[6]	edid_162[5]	edid_162[4]	edid_162[3]	edid_162[2]	edid_162[1]	edid_162[0]
0xEE A3	0x00	edidmemory_reg_eea3	r	edid_163[7]	edid_163[6]	edid_163[5]	edid_163[4]	edid_163[3]	edid_163[2]	edid_163[1]	edid_163[0]
0xEE A4	0x00	edidmemory_reg_eea4	r	edid_164[7]	edid_164[6]	edid_164[5]	edid_164[4]	edid_164[3]	edid_164[2]	edid_164[1]	edid_164[0]
0xEE A5	0x00	edidmemory_reg_eea5	r	edid_165[7]	edid_165[6]	edid_165[5]	edid_165[4]	edid_165[3]	edid_165[2]	edid_165[1]	edid_165[0]
0xEE A6	0x00	edidmemory_reg_eea6	r	edid_166[7]	edid_166[6]	edid_166[5]	edid_166[4]	edid_166[3]	edid_166[2]	edid_166[1]	edid_166[0]
0xEE A7	0x00	edidmemory_reg_eea7	r	edid_167[7]	edid_167[6]	edid_167[5]	edid_167[4]	edid_167[3]	edid_167[2]	edid_167[1]	edid_167[0]
0xEE A8	0x00	edidmemory_reg_eea8	r	edid_168[7]	edid_168[6]	edid_168[5]	edid_168[4]	edid_168[3]	edid_168[2]	edid_168[1]	edid_168[0]
0xEE A9	0x00	edidmemory_reg_eea9	r	edid_169[7]	edid_169[6]	edid_169[5]	edid_169[4]	edid_169[3]	edid_169[2]	edid_169[1]	edid_169[0]
0xEE AA	0x00	edidmemory_reg_eeaa	r	edid_170[7]	edid_170[6]	edid_170[5]	edid_170[4]	edid_170[3]	edid_170[2]	edid_170[1]	edid_170[0]
0xEE AB	0x00	edidmemory_reg_eeab	r	edid_171[7]	edid_171[6]	edid_171[5]	edid_171[4]	edid_171[3]	edid_171[2]	edid_171[1]	edid_171[0]
0xEE AC	0x00	edidmemory_reg_eeac	r	edid_172[7]	edid_172[6]	edid_172[5]	edid_172[4]	edid_172[3]	edid_172[2]	edid_172[1]	edid_172[0]
0xEE AD	0x00	edidmemory_reg_eead	r	edid_173[7]	edid_173[6]	edid_173[5]	edid_173[4]	edid_173[3]	edid_173[2]	edid_173[1]	edid_173[0]
0xEE AE	0x00	edidmemory_reg_eeae	r	edid_174[7]	edid_174[6]	edid_174[5]	edid_174[4]	edid_174[3]	edid_174[2]	edid_174[1]	edid_174[0]
0xEE AF	0x00	edidmemory_reg_eeaf	r	edid_175[7]	edid_175[6]	edid_175[5]	edid_175[4]	edid_175[3]	edid_175[2]	edid_175[1]	edid_175[0]
0xEE B0	0x00	edidmemory_reg_eeb0	r	edid_176[7]	edid_176[6]	edid_176[5]	edid_176[4]	edid_176[3]	edid_176[2]	edid_176[1]	edid_176[0]
0xEE B1	0x00	edidmemory_reg_eeb1	r	edid_177[7]	edid_177[6]	edid_177[5]	edid_177[4]	edid_177[3]	edid_177[2]	edid_177[1]	edid_177[0]
0xEE B2	0x00	edidmemory_reg_eeb2	r	edid_178[7]	edid_178[6]	edid_178[5]	edid_178[4]	edid_178[3]	edid_178[2]	edid_178[1]	edid_178[0]
0xEE B3	0x00	edidmemory_reg_eeb3	r	edid_179[7]	edid_179[6]	edid_179[5]	edid_179[4]	edid_179[3]	edid_179[2]	edid_179[1]	edid_179[0]
0xEE B4	0x00	edidmemory_reg_eeb4	r	edid_180[7]	edid_180[6]	edid_180[5]	edid_180[4]	edid_180[3]	edid_180[2]	edid_180[1]	edid_180[0]
0xEE B5	0x00	edidmemory_reg_eeb5	r	edid_181[7]	edid_181[6]	edid_181[5]	edid_181[4]	edid_181[3]	edid_181[2]	edid_181[1]	edid_181[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE B6	0x00	edidmemory_reg_eeb6	r	edid_182[7]	edid_182[6]	edid_182[5]	edid_182[4]	edid_182[3]	edid_182[2]	edid_182[1]	edid_182[0]
0xEE B7	0x00	edidmemory_reg_eeb7	r	edid_183[7]	edid_183[6]	edid_183[5]	edid_183[4]	edid_183[3]	edid_183[2]	edid_183[1]	edid_183[0]
0xEE B8	0x00	edidmemory_reg_eeb8	r	edid_184[7]	edid_184[6]	edid_184[5]	edid_184[4]	edid_184[3]	edid_184[2]	edid_184[1]	edid_184[0]
0xEE B9	0x00	edidmemory_reg_eeb9	r	edid_185[7]	edid_185[6]	edid_185[5]	edid_185[4]	edid_185[3]	edid_185[2]	edid_185[1]	edid_185[0]
0xEE BA	0x00	edidmemory_reg_eeba	r	edid_186[7]	edid_186[6]	edid_186[5]	edid_186[4]	edid_186[3]	edid_186[2]	edid_186[1]	edid_186[0]
0xEE BB	0x00	edidmemory_reg_eebb	r	edid_187[7]	edid_187[6]	edid_187[5]	edid_187[4]	edid_187[3]	edid_187[2]	edid_187[1]	edid_187[0]
0xEE BC	0x00	edidmemory_reg_eebc	r	edid_188[7]	edid_188[6]	edid_188[5]	edid_188[4]	edid_188[3]	edid_188[2]	edid_188[1]	edid_188[0]
0xEE BD	0x00	edidmemory_reg_eebd	r	edid_189[7]	edid_189[6]	edid_189[5]	edid_189[4]	edid_189[3]	edid_189[2]	edid_189[1]	edid_189[0]
0xEE BE	0x00	edidmemory_reg_eebe	r	edid_190[7]	edid_190[6]	edid_190[5]	edid_190[4]	edid_190[3]	edid_190[2]	edid_190[1]	edid_190[0]
0xEE BF	0x00	edidmemory_reg_eebf	r	edid_191[7]	edid_191[6]	edid_191[5]	edid_191[4]	edid_191[3]	edid_191[2]	edid_191[1]	edid_191[0]
0xEE C0	0x00	edidmemory_reg_eec0	r	edid_192[7]	edid_192[6]	edid_192[5]	edid_192[4]	edid_192[3]	edid_192[2]	edid_192[1]	edid_192[0]
0xEE C1	0x00	edidmemory_reg_eec1	r	edid_193[7]	edid_193[6]	edid_193[5]	edid_193[4]	edid_193[3]	edid_193[2]	edid_193[1]	edid_193[0]
0xEE C2	0x00	edidmemory_reg_eec2	r	edid_194[7]	edid_194[6]	edid_194[5]	edid_194[4]	edid_194[3]	edid_194[2]	edid_194[1]	edid_194[0]
0xEE C3	0x00	edidmemory_reg_eec3	r	edid_195[7]	edid_195[6]	edid_195[5]	edid_195[4]	edid_195[3]	edid_195[2]	edid_195[1]	edid_195[0]
0xEE C4	0x00	edidmemory_reg_eec4	r	edid_196[7]	edid_196[6]	edid_196[5]	edid_196[4]	edid_196[3]	edid_196[2]	edid_196[1]	edid_196[0]
0xEE C5	0x00	edidmemory_reg_eec5	r	edid_197[7]	edid_197[6]	edid_197[5]	edid_197[4]	edid_197[3]	edid_197[2]	edid_197[1]	edid_197[0]
0xEE C6	0x00	edidmemory_reg_eec6	r	edid_198[7]	edid_198[6]	edid_198[5]	edid_198[4]	edid_198[3]	edid_198[2]	edid_198[1]	edid_198[0]
0xEE C7	0x00	edidmemory_reg_eec7	r	edid_199[7]	edid_199[6]	edid_199[5]	edid_199[4]	edid_199[3]	edid_199[2]	edid_199[1]	edid_199[0]
0xEE C8	0x00	edidmemory_reg_eec8	r	edid_200[7]	edid_200[6]	edid_200[5]	edid_200[4]	edid_200[3]	edid_200[2]	edid_200[1]	edid_200[0]
0xEE C9	0x00	edidmemory_reg_eec9	r	edid_201[7]	edid_201[6]	edid_201[5]	edid_201[4]	edid_201[3]	edid_201[2]	edid_201[1]	edid_201[0]
0xEE CA	0x00	edidmemory_reg_eeca	r	edid_202[7]	edid_202[6]	edid_202[5]	edid_202[4]	edid_202[3]	edid_202[2]	edid_202[1]	edid_202[0]
0xEE CB	0x00	edidmemory_reg_eecb	r	edid_203[7]	edid_203[6]	edid_203[5]	edid_203[4]	edid_203[3]	edid_203[2]	edid_203[1]	edid_203[0]
0xEE CC	0x00	edidmemory_reg_eecc	r	edid_204[7]	edid_204[6]	edid_204[5]	edid_204[4]	edid_204[3]	edid_204[2]	edid_204[1]	edid_204[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE CD	0x00	edidmemory_reg_eecd	r	edid_205[7]	edid_205[6]	edid_205[5]	edid_205[4]	edid_205[3]	edid_205[2]	edid_205[1]	edid_205[0]
0xEE CE	0x00	edidmemory_reg_eece	r	edid_206[7]	edid_206[6]	edid_206[5]	edid_206[4]	edid_206[3]	edid_206[2]	edid_206[1]	edid_206[0]
0xEE CF	0x00	edidmemory_reg_eecf	r	edid_207[7]	edid_207[6]	edid_207[5]	edid_207[4]	edid_207[3]	edid_207[2]	edid_207[1]	edid_207[0]
0xEE D0	0x00	edidmemory_reg_eed0	r	edid_208[7]	edid_208[6]	edid_208[5]	edid_208[4]	edid_208[3]	edid_208[2]	edid_208[1]	edid_208[0]
0xEE D1	0x00	edidmemory_reg_eed1	r	edid_209[7]	edid_209[6]	edid_209[5]	edid_209[4]	edid_209[3]	edid_209[2]	edid_209[1]	edid_209[0]
0xEE D2	0x00	edidmemory_reg_eed2	r	edid_210[7]	edid_210[6]	edid_210[5]	edid_210[4]	edid_210[3]	edid_210[2]	edid_210[1]	edid_210[0]
0xEE D3	0x00	edidmemory_reg_eed3	r	edid_211[7]	edid_211[6]	edid_211[5]	edid_211[4]	edid_211[3]	edid_211[2]	edid_211[1]	edid_211[0]
0xEE D4	0x00	edidmemory_reg_eed4	r	edid_212[7]	edid_212[6]	edid_212[5]	edid_212[4]	edid_212[3]	edid_212[2]	edid_212[1]	edid_212[0]
0xEE D5	0x00	edidmemory_reg_eed5	r	edid_213[7]	edid_213[6]	edid_213[5]	edid_213[4]	edid_213[3]	edid_213[2]	edid_213[1]	edid_213[0]
0xEE D6	0x00	edidmemory_reg_eed6	r	edid_214[7]	edid_214[6]	edid_214[5]	edid_214[4]	edid_214[3]	edid_214[2]	edid_214[1]	edid_214[0]
0xEE D7	0x00	edidmemory_reg_eed7	r	edid_215[7]	edid_215[6]	edid_215[5]	edid_215[4]	edid_215[3]	edid_215[2]	edid_215[1]	edid_215[0]
0xEE D8	0x00	edidmemory_reg_eed8	r	edid_216[7]	edid_216[6]	edid_216[5]	edid_216[4]	edid_216[3]	edid_216[2]	edid_216[1]	edid_216[0]
0xEE D9	0x00	edidmemory_reg_eed9	r	edid_217[7]	edid_217[6]	edid_217[5]	edid_217[4]	edid_217[3]	edid_217[2]	edid_217[1]	edid_217[0]
0xEE DA	0x00	edidmemory_reg_eeda	r	edid_218[7]	edid_218[6]	edid_218[5]	edid_218[4]	edid_218[3]	edid_218[2]	edid_218[1]	edid_218[0]
0xEE DB	0x00	edidmemory_reg_eedb	r	edid_219[7]	edid_219[6]	edid_219[5]	edid_219[4]	edid_219[3]	edid_219[2]	edid_219[1]	edid_219[0]
0xEE DC	0x00	edidmemory_reg_eedc	r	edid_220[7]	edid_220[6]	edid_220[5]	edid_220[4]	edid_220[3]	edid_220[2]	edid_220[1]	edid_220[0]
0xEE DD	0x00	edidmemory_reg_eedd	r	edid_221[7]	edid_221[6]	edid_221[5]	edid_221[4]	edid_221[3]	edid_221[2]	edid_221[1]	edid_221[0]
0xEE DE	0x00	edidmemory_reg_eede	r	edid_222[7]	edid_222[6]	edid_222[5]	edid_222[4]	edid_222[3]	edid_222[2]	edid_222[1]	edid_222[0]
0xEE DF	0x00	edidmemory_reg_eedf	r	edid_223[7]	edid_223[6]	edid_223[5]	edid_223[4]	edid_223[3]	edid_223[2]	edid_223[1]	edid_223[0]
0xEE E0	0x00	edidmemory_reg_eee0	r	edid_224[7]	edid_224[6]	edid_224[5]	edid_224[4]	edid_224[3]	edid_224[2]	edid_224[1]	edid_224[0]
0xEE E1	0x00	edidmemory_reg_eee1	r	edid_225[7]	edid_225[6]	edid_225[5]	edid_225[4]	edid_225[3]	edid_225[2]	edid_225[1]	edid_225[0]
0xEE E2	0x00	edidmemory_reg_eee2	r	edid_226[7]	edid_226[6]	edid_226[5]	edid_226[4]	edid_226[3]	edid_226[2]	edid_226[1]	edid_226[0]
0xEE E3	0x00	edidmemory_reg_eee3	r	edid_227[7]	edid_227[6]	edid_227[5]	edid_227[4]	edid_227[3]	edid_227[2]	edid_227[1]	edid_227[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE E4	0x00	edidmemory_reg_eee4	r	edid_228[7]	edid_228[6]	edid_228[5]	edid_228[4]	edid_228[3]	edid_228[2]	edid_228[1]	edid_228[0]
0xEE E5	0x00	edidmemory_reg_eee5	r	edid_229[7]	edid_229[6]	edid_229[5]	edid_229[4]	edid_229[3]	edid_229[2]	edid_229[1]	edid_229[0]
0xEE E6	0x00	edidmemory_reg_eee6	r	edid_230[7]	edid_230[6]	edid_230[5]	edid_230[4]	edid_230[3]	edid_230[2]	edid_230[1]	edid_230[0]
0xEE E7	0x00	edidmemory_reg_eee7	r	edid_231[7]	edid_231[6]	edid_231[5]	edid_231[4]	edid_231[3]	edid_231[2]	edid_231[1]	edid_231[0]
0xEE E8	0x00	edidmemory_reg_eee8	r	edid_232[7]	edid_232[6]	edid_232[5]	edid_232[4]	edid_232[3]	edid_232[2]	edid_232[1]	edid_232[0]
0xEE E9	0x00	edidmemory_reg_eee9	r	edid_233[7]	edid_233[6]	edid_233[5]	edid_233[4]	edid_233[3]	edid_233[2]	edid_233[1]	edid_233[0]
0xEE EA	0x00	edidmemory_reg_eeea	r	edid_234[7]	edid_234[6]	edid_234[5]	edid_234[4]	edid_234[3]	edid_234[2]	edid_234[1]	edid_234[0]
0xEE EB	0x00	edidmemory_reg_eeeb	r	edid_235[7]	edid_235[6]	edid_235[5]	edid_235[4]	edid_235[3]	edid_235[2]	edid_235[1]	edid_235[0]
0xEE EC	0x00	edidmemory_reg_eeec	r	edid_236[7]	edid_236[6]	edid_236[5]	edid_236[4]	edid_236[3]	edid_236[2]	edid_236[1]	edid_236[0]
0xEE ED	0x00	edidmemory_reg_eeed	r	edid_237[7]	edid_237[6]	edid_237[5]	edid_237[4]	edid_237[3]	edid_237[2]	edid_237[1]	edid_237[0]
0xEE EE	0x00	edidmemory_reg_eeee	r	edid_238[7]	edid_238[6]	edid_238[5]	edid_238[4]	edid_238[3]	edid_238[2]	edid_238[1]	edid_238[0]
0xEE EF	0x00	edidmemory_reg_eeef	r	edid_239[7]	edid_239[6]	edid_239[5]	edid_239[4]	edid_239[3]	edid_239[2]	edid_239[1]	edid_239[0]
0xEE F0	0x00	edidmemory_reg_eef0	r	edid_240[7]	edid_240[6]	edid_240[5]	edid_240[4]	edid_240[3]	edid_240[2]	edid_240[1]	edid_240[0]
0xEE F1	0x00	edidmemory_reg_eef1	r	edid_241[7]	edid_241[6]	edid_241[5]	edid_241[4]	edid_241[3]	edid_241[2]	edid_241[1]	edid_241[0]
0xEE F2	0x00	edidmemory_reg_eef2	r	edid_242[7]	edid_242[6]	edid_242[5]	edid_242[4]	edid_242[3]	edid_242[2]	edid_242[1]	edid_242[0]
0xEE F3	0x00	edidmemory_reg_eef3	r	edid_243[7]	edid_243[6]	edid_243[5]	edid_243[4]	edid_243[3]	edid_243[2]	edid_243[1]	edid_243[0]
0xEE F4	0x00	edidmemory_reg_eef4	r	edid_244[7]	edid_244[6]	edid_244[5]	edid_244[4]	edid_244[3]	edid_244[2]	edid_244[1]	edid_244[0]
0xEE F5	0x00	edidmemory_reg_eef5	r	edid_245[7]	edid_245[6]	edid_245[5]	edid_245[4]	edid_245[3]	edid_245[2]	edid_245[1]	edid_245[0]
0xEE F6	0x00	edidmemory_reg_eef6	r	edid_246[7]	edid_246[6]	edid_246[5]	edid_246[4]	edid_246[3]	edid_246[2]	edid_246[1]	edid_246[0]
0xEE F7	0x00	edidmemory_reg_eef7	r	edid_247[7]	edid_247[6]	edid_247[5]	edid_247[4]	edid_247[3]	edid_247[2]	edid_247[1]	edid_247[0]
0xEE F8	0x00	edidmemory_reg_eef8	r	edid_248[7]	edid_248[6]	edid_248[5]	edid_248[4]	edid_248[3]	edid_248[2]	edid_248[1]	edid_248[0]
0xEE F9	0x00	edidmemory_reg_eef9	r	edid_249[7]	edid_249[6]	edid_249[5]	edid_249[4]	edid_249[3]	edid_249[2]	edid_249[1]	edid_249[0]
0xEEF A	0x00	edidmemory_reg_eefa	r	edid_250[7]	edid_250[6]	edid_250[5]	edid_250[4]	edid_250[3]	edid_250[2]	edid_250[1]	edid_250[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xEE FB	0x00	edidmemory_reg_eefb	r	edid_251[7]	edid_251[6]	edid_251[5]	edid_251[4]	edid_251[3]	edid_251[2]	edid_251[1]	edid_251[0]
0xEE FC	0x00	edidmemory_reg_eefc	r	edid_252[7]	edid_252[6]	edid_252[5]	edid_252[4]	edid_252[3]	edid_252[2]	edid_252[1]	edid_252[0]
0xEE FD	0x00	edidmemory_reg_eefd	r	edid_253[7]	edid_253[6]	edid_253[5]	edid_253[4]	edid_253[3]	edid_253[2]	edid_253[1]	edid_253[0]
0xEE FE	0x00	edidmemory_reg_eefe	r	edid_254[7]	edid_254[6]	edid_254[5]	edid_254[4]	edid_254[3]	edid_254[2]	edid_254[1]	edid_254[0]
0xEE FF	0x00	edidmemory_reg_eeff	r	edid_255[7]	edid_255[6]	edid_255[5]	edid_255[4]	edid_255[3]	edid_255[2]	edid_255[1]	edid_255[0]

1.10 TX1 TEST MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF3 BF	0x00	test_reg_f3bf	rw	-	-	-	-	-	spare_pkt3_en	spare_pkt4_en	-
0xF3 C0	0x00	test_reg_f3c0	rw	spare3_header0[7]	spare3_header0[6]	spare3_header0[5]	spare3_header0[4]	spare3_header0[3]	spare3_header0[2]	spare3_header0[1]	spare3_header0[0]
0xF3 C1	0x00	test_reg_f3c1	rw	spare3_header1[7]	spare3_header1[6]	spare3_header1[5]	spare3_header1[4]	spare3_header1[3]	spare3_header1[2]	spare3_header1[1]	spare3_header1[0]
0xF3 C2	0x00	test_reg_f3c2	rw	spare3_header2[7]	spare3_header2[6]	spare3_header2[5]	spare3_header2[4]	spare3_header2[3]	spare3_header2[2]	spare3_header2[1]	spare3_header2[0]
0xF3 C3	0x00	test_reg_f3c3	rw	spare3_byte0[7]	spare3_byte0[6]	spare3_byte0[5]	spare3_byte0[4]	spare3_byte0[3]	spare3_byte0[2]	spare3_byte0[1]	spare3_byte0[0]
0xF3 C4	0x00	test_reg_f3c4	rw	spare3_byte1[7]	spare3_byte1[6]	spare3_byte1[5]	spare3_byte1[4]	spare3_byte1[3]	spare3_byte1[2]	spare3_byte1[1]	spare3_byte1[0]
0xF3 C5	0x00	test_reg_f3c5	rw	spare3_byte2[7]	spare3_byte2[6]	spare3_byte2[5]	spare3_byte2[4]	spare3_byte2[3]	spare3_byte2[2]	spare3_byte2[1]	spare3_byte2[0]
0xF3 C6	0x00	test_reg_f3c6	rw	spare3_byte3[7]	spare3_byte3[6]	spare3_byte3[5]	spare3_byte3[4]	spare3_byte3[3]	spare3_byte3[2]	spare3_byte3[1]	spare3_byte3[0]
0xF3 C7	0x00	test_reg_f3c7	rw	spare3_byte4[7]	spare3_byte4[6]	spare3_byte4[5]	spare3_byte4[4]	spare3_byte4[3]	spare3_byte4[2]	spare3_byte4[1]	spare3_byte4[0]
0xF3 C8	0x00	test_reg_f3c8	rw	spare3_byte5[7]	spare3_byte5[6]	spare3_byte5[5]	spare3_byte5[4]	spare3_byte5[3]	spare3_byte5[2]	spare3_byte5[1]	spare3_byte5[0]
0xF3 C9	0x00	test_reg_f3c9	rw	spare3_byte6[7]	spare3_byte6[6]	spare3_byte6[5]	spare3_byte6[4]	spare3_byte6[3]	spare3_byte6[2]	spare3_byte6[1]	spare3_byte6[0]
0xF3 CA	0x00	test_reg_f3ca	rw	spare3_byte7[7]	spare3_byte7[6]	spare3_byte7[5]	spare3_byte7[4]	spare3_byte7[3]	spare3_byte7[2]	spare3_byte7[1]	spare3_byte7[0]
0xF3 CB	0x00	test_reg_f3cb	rw	spare3_byte8[7]	spare3_byte8[6]	spare3_byte8[5]	spare3_byte8[4]	spare3_byte8[3]	spare3_byte8[2]	spare3_byte8[1]	spare3_byte8[0]
0xF3 CC	0x00	test_reg_f3cc	rw	spare3_byte9[7]	spare3_byte9[6]	spare3_byte9[5]	spare3_byte9[4]	spare3_byte9[3]	spare3_byte9[2]	spare3_byte9[1]	spare3_byte9[0]
0xF3 CD	0x00	test_reg_f3cd	rw	spare3_byte10[7]	spare3_byte10[6]	spare3_byte10[5]	spare3_byte10[4]	spare3_byte10[3]	spare3_byte10[2]	spare3_byte10[1]	spare3_byte10[0]
0xF3 CE	0x00	test_reg_f3ce	rw	spare3_byte11[7]	spare3_byte11[6]	spare3_byte11[5]	spare3_byte11[4]	spare3_byte11[3]	spare3_byte11[2]	spare3_byte11[1]	spare3_byte11[0]
0xF3 CF	0x00	test_reg_f3cf	rw	spare3_byte12[7]	spare3_byte12[6]	spare3_byte12[5]	spare3_byte12[4]	spare3_byte12[3]	spare3_byte12[2]	spare3_byte12[1]	spare3_byte12[0]
0xF3 D0	0x00	test_reg_f3d0	rw	spare3_byte13[7]	spare3_byte13[6]	spare3_byte13[5]	spare3_byte13[4]	spare3_byte13[3]	spare3_byte13[2]	spare3_byte13[1]	spare3_byte13[0]
0xF3 D1	0x00	test_reg_f3d1	rw	spare3_byte14[7]	spare3_byte14[6]	spare3_byte14[5]	spare3_byte14[4]	spare3_byte14[3]	spare3_byte14[2]	spare3_byte14[1]	spare3_byte14[0]
0xF3 D2	0x00	test_reg_f3d2	rw	spare3_byte15[7]	spare3_byte15[6]	spare3_byte15[5]	spare3_byte15[4]	spare3_byte15[3]	spare3_byte15[2]	spare3_byte15[1]	spare3_byte15[0]
0xF3 D3	0x00	test_reg_f3d3	rw	spare3_byte16[7]	spare3_byte16[6]	spare3_byte16[5]	spare3_byte16[4]	spare3_byte16[3]	spare3_byte16[2]	spare3_byte16[1]	spare3_byte16[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF3 D4	0x00	test_reg_f3d4	rw	spare3_byte17[7]	spare3_byte17[6]	spare3_byte17[5]	spare3_byte17[4]	spare3_byte17[3]	spare3_byte17[2]	spare3_byte17[1]	spare3_byte17[0]
0xF3 D5	0x00	test_reg_f3d5	rw	spare3_byte18[7]	spare3_byte18[6]	spare3_byte18[5]	spare3_byte18[4]	spare3_byte18[3]	spare3_byte18[2]	spare3_byte18[1]	spare3_byte18[0]
0xF3 D6	0x00	test_reg_f3d6	rw	spare3_byte19[7]	spare3_byte19[6]	spare3_byte19[5]	spare3_byte19[4]	spare3_byte19[3]	spare3_byte19[2]	spare3_byte19[1]	spare3_byte19[0]
0xF3 D7	0x00	test_reg_f3d7	rw	spare3_byte20[7]	spare3_byte20[6]	spare3_byte20[5]	spare3_byte20[4]	spare3_byte20[3]	spare3_byte20[2]	spare3_byte20[1]	spare3_byte20[0]
0xF3 D8	0x00	test_reg_f3d8	rw	spare3_byte21[7]	spare3_byte21[6]	spare3_byte21[5]	spare3_byte21[4]	spare3_byte21[3]	spare3_byte21[2]	spare3_byte21[1]	spare3_byte21[0]
0xF3 D9	0x00	test_reg_f3d9	rw	spare3_byte22[7]	spare3_byte22[6]	spare3_byte22[5]	spare3_byte22[4]	spare3_byte22[3]	spare3_byte22[2]	spare3_byte22[1]	spare3_byte22[0]
0xF3 DA	0x00	test_reg_f3da	rw	spare3_byte23[7]	spare3_byte23[6]	spare3_byte23[5]	spare3_byte23[4]	spare3_byte23[3]	spare3_byte23[2]	spare3_byte23[1]	spare3_byte23[0]
0xF3 DB	0x00	test_reg_f3db	rw	spare3_byte24[7]	spare3_byte24[6]	spare3_byte24[5]	spare3_byte24[4]	spare3_byte24[3]	spare3_byte24[2]	spare3_byte24[1]	spare3_byte24[0]
0xF3 DC	0x00	test_reg_f3dc	rw	spare3_byte25[7]	spare3_byte25[6]	spare3_byte25[5]	spare3_byte25[4]	spare3_byte25[3]	spare3_byte25[2]	spare3_byte25[1]	spare3_byte25[0]
0xF3 DD	0x00	test_reg_f3dd	rw	spare3_byte26[7]	spare3_byte26[6]	spare3_byte26[5]	spare3_byte26[4]	spare3_byte26[3]	spare3_byte26[2]	spare3_byte26[1]	spare3_byte26[0]
0xF3 DE	0x00	test_reg_f3de	rw	spare3_byte27[7]	spare3_byte27[6]	spare3_byte27[5]	spare3_byte27[4]	spare3_byte27[3]	spare3_byte27[2]	spare3_byte27[1]	spare3_byte27[0]
0xF3 DF	0x00	test_reg_f3df	rw	spare3_update	-	-	-	-	-	-	-
0xF3 E0	0x00	test_reg_f3e0	rw	spare4_header0[7]	spare4_header0[6]	spare4_header0[5]	spare4_header0[4]	spare4_header0[3]	spare4_header0[2]	spare4_header0[1]	spare4_header0[0]
0xF3 E1	0x00	test_reg_f3e1	rw	spare4_header1[7]	spare4_header1[6]	spare4_header1[5]	spare4_header1[4]	spare4_header1[3]	spare4_header1[2]	spare4_header1[1]	spare4_header1[0]
0xF3 E2	0x00	test_reg_f3e2	rw	spare4_header2[7]	spare4_header2[6]	spare4_header2[5]	spare4_header2[4]	spare4_header2[3]	spare4_header2[2]	spare4_header2[1]	spare4_header2[0]
0xF3 E3	0x00	test_reg_f3e3	rw	spare4_byte0[7]	spare4_byte0[6]	spare4_byte0[5]	spare4_byte0[4]	spare4_byte0[3]	spare4_byte0[2]	spare4_byte0[1]	spare4_byte0[0]
0xF3 E4	0x00	test_reg_f3e4	rw	spare4_byte1[7]	spare4_byte1[6]	spare4_byte1[5]	spare4_byte1[4]	spare4_byte1[3]	spare4_byte1[2]	spare4_byte1[1]	spare4_byte1[0]
0xF3 E5	0x00	test_reg_f3e5	rw	spare4_byte2[7]	spare4_byte2[6]	spare4_byte2[5]	spare4_byte2[4]	spare4_byte2[3]	spare4_byte2[2]	spare4_byte2[1]	spare4_byte2[0]
0xF3 E6	0x00	test_reg_f3e6	rw	spare4_byte3[7]	spare4_byte3[6]	spare4_byte3[5]	spare4_byte3[4]	spare4_byte3[3]	spare4_byte3[2]	spare4_byte3[1]	spare4_byte3[0]
0xF3 E7	0x00	test_reg_f3e7	rw	spare4_byte4[7]	spare4_byte4[6]	spare4_byte4[5]	spare4_byte4[4]	spare4_byte4[3]	spare4_byte4[2]	spare4_byte4[1]	spare4_byte4[0]
0xF3 E8	0x00	test_reg_f3e8	rw	spare4_byte5[7]	spare4_byte5[6]	spare4_byte5[5]	spare4_byte5[4]	spare4_byte5[3]	spare4_byte5[2]	spare4_byte5[1]	spare4_byte5[0]
0xF3 E9	0x00	test_reg_f3e9	rw	spare4_byte6[7]	spare4_byte6[6]	spare4_byte6[5]	spare4_byte6[4]	spare4_byte6[3]	spare4_byte6[2]	spare4_byte6[1]	spare4_byte6[0]
0xF3 EA	0x00	test_reg_f3ea	rw	spare4_byte7[7]	spare4_byte7[6]	spare4_byte7[5]	spare4_byte7[4]	spare4_byte7[3]	spare4_byte7[2]	spare4_byte7[1]	spare4_byte7[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF3 EB	0x00	test_reg_f3eb	rw	spare4_byte8[7]	spare4_byte8[6]	spare4_byte8[5]	spare4_byte8[4]	spare4_byte8[3]	spare4_byte8[2]	spare4_byte8[1]	spare4_byte8[0]
0xF3 EC	0x00	test_reg_f3ec	rw	spare4_byte9[7]	spare4_byte9[6]	spare4_byte9[5]	spare4_byte9[4]	spare4_byte9[3]	spare4_byte9[2]	spare4_byte9[1]	spare4_byte9[0]
0xF3 ED	0x00	test_reg_f3ed	rw	spare4_byte10[7]	spare4_byte10[6]	spare4_byte10[5]	spare4_byte10[4]	spare4_byte10[3]	spare4_byte10[2]	spare4_byte10[1]	spare4_byte10[0]
0xF3 EE	0x00	test_reg_f3ee	rw	spare4_byte11[7]	spare4_byte11[6]	spare4_byte11[5]	spare4_byte11[4]	spare4_byte11[3]	spare4_byte11[2]	spare4_byte11[1]	spare4_byte11[0]
0xF3 EF	0x00	test_reg_f3ef	rw	spare4_byte12[7]	spare4_byte12[6]	spare4_byte12[5]	spare4_byte12[4]	spare4_byte12[3]	spare4_byte12[2]	spare4_byte12[1]	spare4_byte12[0]
0xF3 F0	0x00	test_reg_f3f0	rw	spare4_byte13[7]	spare4_byte13[6]	spare4_byte13[5]	spare4_byte13[4]	spare4_byte13[3]	spare4_byte13[2]	spare4_byte13[1]	spare4_byte13[0]
0xF3 F1	0x00	test_reg_f3f1	rw	spare4_byte14[7]	spare4_byte14[6]	spare4_byte14[5]	spare4_byte14[4]	spare4_byte14[3]	spare4_byte14[2]	spare4_byte14[1]	spare4_byte14[0]
0xF3 F2	0x00	test_reg_f3f2	rw	spare4_byte15[7]	spare4_byte15[6]	spare4_byte15[5]	spare4_byte15[4]	spare4_byte15[3]	spare4_byte15[2]	spare4_byte15[1]	spare4_byte15[0]
0xF3 F3	0x00	test_reg_f3f3	rw	spare4_byte16[7]	spare4_byte16[6]	spare4_byte16[5]	spare4_byte16[4]	spare4_byte16[3]	spare4_byte16[2]	spare4_byte16[1]	spare4_byte16[0]
0xF3 F4	0x00	test_reg_f3f4	rw	spare4_byte17[7]	spare4_byte17[6]	spare4_byte17[5]	spare4_byte17[4]	spare4_byte17[3]	spare4_byte17[2]	spare4_byte17[1]	spare4_byte17[0]
0xF3 F5	0x00	test_reg_f3f5	rw	spare4_byte18[7]	spare4_byte18[6]	spare4_byte18[5]	spare4_byte18[4]	spare4_byte18[3]	spare4_byte18[2]	spare4_byte18[1]	spare4_byte18[0]
0xF3 F6	0x00	test_reg_f3f6	rw	spare4_byte19[7]	spare4_byte19[6]	spare4_byte19[5]	spare4_byte19[4]	spare4_byte19[3]	spare4_byte19[2]	spare4_byte19[1]	spare4_byte19[0]
0xF3 F7	0x00	test_reg_f3f7	rw	spare4_byte20[7]	spare4_byte20[6]	spare4_byte20[5]	spare4_byte20[4]	spare4_byte20[3]	spare4_byte20[2]	spare4_byte20[1]	spare4_byte20[0]
0xF3 F8	0x00	test_reg_f3f8	rw	spare4_byte21[7]	spare4_byte21[6]	spare4_byte21[5]	spare4_byte21[4]	spare4_byte21[3]	spare4_byte21[2]	spare4_byte21[1]	spare4_byte21[0]
0xF3 F9	0x00	test_reg_f3f9	rw	spare4_byte22[7]	spare4_byte22[6]	spare4_byte22[5]	spare4_byte22[4]	spare4_byte22[3]	spare4_byte22[2]	spare4_byte22[1]	spare4_byte22[0]
0xF3 FA	0x00	test_reg_f3fa	rw	spare4_byte23[7]	spare4_byte23[6]	spare4_byte23[5]	spare4_byte23[4]	spare4_byte23[3]	spare4_byte23[2]	spare4_byte23[1]	spare4_byte23[0]
0xF3 FB	0x00	test_reg_f3fb	rw	spare4_byte24[7]	spare4_byte24[6]	spare4_byte24[5]	spare4_byte24[4]	spare4_byte24[3]	spare4_byte24[2]	spare4_byte24[1]	spare4_byte24[0]
0xF3 FC	0x00	test_reg_f3fc	rw	spare4_byte25[7]	spare4_byte25[6]	spare4_byte25[5]	spare4_byte25[4]	spare4_byte25[3]	spare4_byte25[2]	spare4_byte25[1]	spare4_byte25[0]
0xF3 FD	0x00	test_reg_f3fd	rw	spare4_byte26[7]	spare4_byte26[6]	spare4_byte26[5]	spare4_byte26[4]	spare4_byte26[3]	spare4_byte26[2]	spare4_byte26[1]	spare4_byte26[0]
0xF3 FE	0x00	test_reg_f3fe	rw	spare4_byte27[7]	spare4_byte27[6]	spare4_byte27[5]	spare4_byte27[4]	spare4_byte27[3]	spare4_byte27[2]	spare4_byte27[1]	spare4_byte27[0]
0xF3 FF	0x00	test_reg_f3ff	rw	spare4_update	-	-	-	-	-	-	-

1.11 TX2 MAIN MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF401	0x00	main_reg_f401	rw	-	-	-	-	n[19]	n[18]	n[17]	n[16]
0xF402	0x00	main_reg_f402	rw	n[15]	n[14]	n[13]	n[12]	n[11]	n[10]	n[9]	n[8]
0xF403	0x00	main_reg_f403	rw	n[7]	n[6]	n[5]	n[4]	n[3]	n[2]	n[1]	n[0]
0xF404	0x00	main_reg_f404	r	spdif_sf[3]	spdif_sf[2]	spdif_sf[1]	spdif_sf[0]	cts_internal[19]	cts_internal[18]	cts_internal[17]	cts_internal[16]
0xF405	0x00	main_reg_f405	r	cts_internal[15]	cts_internal[14]	cts_internal[13]	cts_internal[12]	cts_internal[11]	cts_internal[10]	cts_internal[9]	cts_internal[8]
0xF406	0x00	main_reg_f406	r	cts_internal[7]	cts_internal[6]	cts_internal[5]	cts_internal[4]	cts_internal[3]	cts_internal[2]	cts_internal[1]	cts_internal[0]
0xF407	0x00	main_reg_f407	rw	-	-	-	-	cts_manual[19]	cts_manual[18]	cts_manual[17]	cts_manual[16]
0xF408	0x00	main_reg_f408	rw	cts_manual[15]	cts_manual[14]	cts_manual[13]	cts_manual[12]	cts_manual[11]	cts_manual[10]	cts_manual[9]	cts_manual[8]
0xF409	0x00	main_reg_f409	rw	cts_manual[7]	cts_manual[6]	cts_manual[5]	cts_manual[4]	cts_manual[3]	cts_manual[2]	cts_manual[1]	cts_manual[0]
0xF40A	0x01	main_reg_f40a	rw	cts_sel	audio_input_sel[2]	audio_input_sel[1]	audio_input_sel[0]	audio_mode[1]	audio_mode[0]	mclk_ratio[1]	mclk_ratio[0]
0xF40B	0x0E	main_reg_f40b	rw	spdif_en	mclk_pol	mclk_en	-	-	-	-	rx_aud_packet_sel
0xF40C	0xBC	main_reg_f40c	rw	audio_sampling_freq_sel	cs_bit_override	i2s_en[3]	i2s_en[2]	i2s_en[1]	i2s_en[0]	i2s_format[1]	i2s_format[0]
0xF40D	0x18	main_reg_f40d	rw	-	-	-	i2s_bit_width[4]	i2s_bit_width[3]	i2s_bit_width[2]	i2s_bit_width[1]	i2s_bit_width[0]
0xF40E	0x01	main_reg_f40e	rw	-	-	subpkt0_l_src[2]	subpkt0_l_src[1]	subpkt0_l_src[0]	subpkt0_r_src[2]	subpkt0_r_src[1]	subpkt0_r_src[0]
0xF40F	0x13	main_reg_f40f	rw	-	-	subpkt1_l_src[2]	subpkt1_l_src[1]	subpkt1_l_src[0]	subpkt1_r_src[2]	subpkt1_r_src[1]	subpkt1_r_src[0]
0xF410	0x25	main_reg_f410	rw	-	-	subpkt2_l_src[2]	subpkt2_l_src[1]	subpkt2_l_src[0]	subpkt2_r_src[2]	subpkt2_r_src[1]	subpkt2_r_src[0]
0xF411	0x37	main_reg_f411	rw	-	-	subpkt3_l_src[2]	subpkt3_l_src[1]	subpkt3_l_src[0]	subpkt3_r_src[2]	subpkt3_r_src[1]	subpkt3_r_src[0]
0xF412	0x00	main_reg_f412	rw	channel_status[1]	channel_status[0]	cr_bit	a_info[2]	a_info[1]	a_info[0]	clk_acc[1]	clk_acc[0]
0xF413	0x00	main_reg_f413	rw	category_code[7]	category_code[6]	category_code[5]	category_code[4]	category_code[3]	category_code[2]	category_code[1]	category_code[0]
0xF414	0x00	main_reg_f414	rw	source_number[3]	source_number[2]	source_number[1]	source_number[0]	word_length[3]	word_length[2]	word_length[1]	word_length[0]
0xF415	0x00	main_reg_f415	rw	i2s_sf[3]	i2s_sf[2]	i2s_sf[1]	i2s_sf[0]	vfe_input_id[3]	vfe_input_id[2]	vfe_input_id[1]	vfe_input_id[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF416	0x00	main_reg_f416	rw	vfe_output_format[1]	vfe_output_format[0]	-	-	-	-	-	vfe_input_cs
0xF417	0x00	main_reg_f417	rw	-	-	-	-	-	gen_444_en	aspect_ratio	-
0xF418	0x46	main_reg_f418	rw	csc_en	csc_scaling_factor[1]	csc_scaling_factor[0]	csc_a1[12]	csc_a1[11]	csc_a1[10]	csc_a1[9]	csc_a1[8]
0xF419	0x62	main_reg_f419	rw	csc_a1[7]	csc_a1[6]	csc_a1[5]	csc_a1[4]	csc_a1[3]	csc_a1[2]	csc_a1[1]	csc_a1[0]
0xF41A	0x04	main_reg_f41a	rw	-	-	-	csc_a2[12]	csc_a2[11]	csc_a2[10]	csc_a2[9]	csc_a2[8]
0xF41B	0xA8	main_reg_f41b	rw	csc_a2[7]	csc_a2[6]	csc_a2[5]	csc_a2[4]	csc_a2[3]	csc_a2[2]	csc_a2[1]	csc_a2[0]
0xF41C	0x00	main_reg_f41c	rw	-	-	-	csc_a3[12]	csc_a3[11]	csc_a3[10]	csc_a3[9]	csc_a3[8]
0xF41D	0x00	main_reg_f41d	rw	csc_a3[7]	csc_a3[6]	csc_a3[5]	csc_a3[4]	csc_a3[3]	csc_a3[2]	csc_a3[1]	csc_a3[0]
0xF41E	0x1C	main_reg_f41e	rw	-	-	-	csc_a4[12]	csc_a4[11]	csc_a4[10]	csc_a4[9]	csc_a4[8]
0xF41F	0x84	main_reg_f41f	rw	csc_a4[7]	csc_a4[6]	csc_a4[5]	csc_a4[4]	csc_a4[3]	csc_a4[2]	csc_a4[1]	csc_a4[0]
0xF420	0x1C	main_reg_f420	rw	-	-	-	csc_b1[12]	csc_b1[11]	csc_b1[10]	csc_b1[9]	csc_b1[8]
0xF421	0xBF	main_reg_f421	rw	csc_b1[7]	csc_b1[6]	csc_b1[5]	csc_b1[4]	csc_b1[3]	csc_b1[2]	csc_b1[1]	csc_b1[0]
0xF422	0x04	main_reg_f422	rw	-	-	-	csc_b2[12]	csc_b2[11]	csc_b2[10]	csc_b2[9]	csc_b2[8]
0xF423	0xAB	main_reg_f423	rw	csc_b2[7]	csc_b2[6]	csc_b2[5]	csc_b2[4]	csc_b2[3]	csc_b2[2]	csc_b2[1]	csc_b2[0]
0xF424	0x1E	main_reg_f424	rw	-	-	-	csc_b3[12]	csc_b3[11]	csc_b3[10]	csc_b3[9]	csc_b3[8]
0xF425	0x70	main_reg_f425	rw	csc_b3[7]	csc_b3[6]	csc_b3[5]	csc_b3[4]	csc_b3[3]	csc_b3[2]	csc_b3[1]	csc_b3[0]
0xF426	0x02	main_reg_f426	rw	-	-	-	csc_b4[12]	csc_b4[11]	csc_b4[10]	csc_b4[9]	csc_b4[8]
0xF427	0x1E	main_reg_f427	rw	csc_b4[7]	csc_b4[6]	csc_b4[5]	csc_b4[4]	csc_b4[3]	csc_b4[2]	csc_b4[1]	csc_b4[0]
0xF428	0x00	main_reg_f428	rw	-	-	-	csc_c1[12]	csc_c1[11]	csc_c1[10]	csc_c1[9]	csc_c1[8]
0xF429	0x00	main_reg_f429	rw	csc_c1[7]	csc_c1[6]	csc_c1[5]	csc_c1[4]	csc_c1[3]	csc_c1[2]	csc_c1[1]	csc_c1[0]
0xF42A	0x04	main_reg_f42a	rw	-	-	-	csc_c2[12]	csc_c2[11]	csc_c2[10]	csc_c2[9]	csc_c2[8]
0xF42B	0xA8	main_reg_f42b	rw	csc_c2[7]	csc_c2[6]	csc_c2[5]	csc_c2[4]	csc_c2[3]	csc_c2[2]	csc_c2[1]	csc_c2[0]
0xF42C	0x08	main_reg_f42c	rw	-	-	-	csc_c3[12]	csc_c3[11]	csc_c3[10]	csc_c3[9]	csc_c3[8]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF4 2D	0x12	main_reg_f42d	rw	csc_c3[7]	csc_c3[6]	csc_c3[5]	csc_c3[4]	csc_c3[3]	csc_c3[2]	csc_c3[1]	csc_c3[0]
0xF4 2E	0x1B	main_reg_f42e	rw	-	-	-	csc_c4[12]	csc_c4[11]	csc_c4[10]	csc_c4[9]	csc_c4[8]
0xF4 2F	0xAC	main_reg_f42f	rw	csc_c4[7]	csc_c4[6]	csc_c4[5]	csc_c4[4]	csc_c4[3]	csc_c4[2]	csc_c4[1]	csc_c4[0]
0xF4 3B	0x80	main_reg_f43b	rw	-	pr_mode[1]	pr_mode[0]	pr_pll_manual[1]	pr_pll_manual[0]	pr_value_manual[1]	pr_value_manual[0]	-
0xF4 3C	0x00	main_reg_f43c	rw	-	-	vic_manual[5]	vic_manual[4]	vic_manual[3]	vic_manual[2]	vic_manual[1]	vic_manual[0]
0xF4 3D	0x00	main_reg_f43d	r	pr_to_rx[1]	pr_to_rx[0]	vic_to_rx[5]	vic_to_rx[4]	vic_to_rx[3]	vic_to_rx[2]	vic_to_rx[1]	vic_to_rx[0]
0xF4 3E	0x00	main_reg_f43e	r	vic_detected[5]	vic_detected[4]	vic_detected[3]	vic_detected[2]	vic_detected[1]	vic_detected[0]	-	-
0xF4 3F	0x00	main_reg_f43f	r	aux_vic_detected[2]	aux_vic_detected[1]	aux_vic_detected[0]	progressive_mode _info[1]	progressive_mode _info[0]	-	-	-
0xF4 40	0x00	main_reg_f440	rw	gc_pkt_en	spd_pkt_en	mpeg_pkt_en	acp_pkt_en	isrc_pkt_en	gm_pkt_en	spare_pkt1_en	spare_pkt0_en
0xF4 41	0x50	main_reg_f441	rw	-	system_pd	-	-	-	-	-	-
0xF4 42	0x90	main_reg_f442	r	-	hpd_state	rx_sense_state	-	i2s_32bit_mode	-	-	-
0xF4 44	0x79	main_reg_f444	rw	-	n_cts_pkt_en	audio_sample_pkt _en	aviif_pkt_en	audioif_pkt_en	-	-	-
0xF4 45	0x70	main_reg_f445	rw	packet_memory_ address[7]	packet_memory_ address[6]	packet_memory_ address[5]	packet_memory_ address[4]	packet_memory_ address[3]	packet_memory_ address[2]	packet_memory_ address[1]	packet_memory_ address[0]
0xF4 46	0x00	main_reg_f446	rw	dsd_en[7]	dsd_en[6]	dsd_en[5]	dsd_en[4]	dsd_en[3]	dsd_en[2]	dsd_en[1]	dsd_en[0]
0xF4 47	0x01	main_reg_f447	rw	dsd_mux_en	papb_sync	sample_invalid[3]	sample_invalid[2]	sample_invalid[1]	sample_invalid[0]	-	arc_eff_tran_en
0xF4 49	0x54	main_reg_f449	rw	-	dither_mode[5]	dither_mode[4]	dither_mode[3]	dither_mode[2]	dither_mode[1]	dither_mode[0]	-
0xF4 4A	0x80	main_reg_f44a	rw	auto_checksum_e n	avi_update	audio_update	gcp_update	man_layout_en	man_layout_sel	-	-
0xF4 4B	0x00	main_reg_f44b	rw	clear_avmute	set_avmute	-	-	-	-	-	-
0xF4 4C	0x00	main_reg_f44c	rw	-	-	-	-	gc_cd[3]	gc_cd[2]	gc_cd[1]	gc_cd[0]
0xF4 4D	0x00	main_reg_f44d	rw	gc_byte2[7]	gc_byte2[6]	gc_byte2[5]	gc_byte2[4]	gc_byte2[3]	gc_byte2[2]	gc_byte2[1]	gc_byte2[0]
0xF4 4E	0x00	main_reg_f44e	rw	gc_byte3[7]	gc_byte3[6]	gc_byte3[5]	gc_byte3[4]	gc_byte3[3]	gc_byte3[2]	gc_byte3[1]	gc_byte3[0]
0xF4 4F	0x00	main_reg_f44f	rw	gc_byte4[7]	gc_byte4[6]	gc_byte4[5]	gc_byte4[4]	gc_byte4[3]	gc_byte4[2]	gc_byte4[1]	gc_byte4[0]
0xF4 50	0x00	main_reg_f450	rw	gc_byte5[7]	gc_byte5[6]	gc_byte5[5]	gc_byte5[4]	gc_byte5[3]	gc_byte5[2]	gc_byte5[1]	gc_byte5[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF451	0x00	main_reg_f451	rw	gc_byte6[7]	gc_byte6[6]	gc_byte6[5]	gc_byte6[4]	gc_byte6[3]	gc_byte6[2]	gc_byte6[1]	gc_byte6[0]
0xF452	0x02	main_reg_f452	rw	-	-	-	-	-	avi_version[2]	avi_version[1]	avi_version[0]
0xF453	0x0D	main_reg_f453	rw	-	-	-	avi_length[4]	avi_length[3]	avi_length[2]	avi_length[1]	avi_length[0]
0xF454	0x00	main_reg_f454	rw	avi_checksum[7]	avi_checksum[6]	avi_checksum[5]	avi_checksum[4]	avi_checksum[3]	avi_checksum[2]	avi_checksum[1]	avi_checksum[0]
0xF455	0x00	main_reg_f455	rw	avi_byte1_7	y1y0[1]	y1y0[0]	a0	b1b0[1]	b1b0[0]	s1s0[1]	s1s0[0]
0xF456	0x00	main_reg_f456	rw	c1c0[1]	c1c0[0]	m1m0[1]	m1m0[0]	r[3]	r[2]	r[1]	r[0]
0xF457	0x00	main_reg_f457	rw	itc	ec[2]	ec[1]	ec[0]	q1q0[1]	q1q0[0]	sc[1]	sc[0]
0xF458	0x00	main_reg_f458	rw	avi_byte4_7	-	-	-	-	-	-	-
0xF459	0x00	main_reg_f459	rw	avi_byte5_7_4[3]	avi_byte5_7_4[2]	avi_byte5_7_4[1]	avi_byte5_7_4[0]	-	-	-	-
0xF45A	0x00	main_reg_f45a	rw	avi_byte6[7]	avi_byte6[6]	avi_byte6[5]	avi_byte6[4]	avi_byte6[3]	avi_byte6[2]	avi_byte6[1]	avi_byte6[0]
0xF45B	0x00	main_reg_f45b	rw	avi_byte7[7]	avi_byte7[6]	avi_byte7[5]	avi_byte7[4]	avi_byte7[3]	avi_byte7[2]	avi_byte7[1]	avi_byte7[0]
0xF45C	0x00	main_reg_f45c	rw	avi_byte8[7]	avi_byte8[6]	avi_byte8[5]	avi_byte8[4]	avi_byte8[3]	avi_byte8[2]	avi_byte8[1]	avi_byte8[0]
0xF45D	0x00	main_reg_f45d	rw	avi_byte9[7]	avi_byte9[6]	avi_byte9[5]	avi_byte9[4]	avi_byte9[3]	avi_byte9[2]	avi_byte9[1]	avi_byte9[0]
0xF45E	0x00	main_reg_f45e	rw	avi_byte10[7]	avi_byte10[6]	avi_byte10[5]	avi_byte10[4]	avi_byte10[3]	avi_byte10[2]	avi_byte10[1]	avi_byte10[0]
0xF45F	0x00	main_reg_f45f	rw	avi_byte11[7]	avi_byte11[6]	avi_byte11[5]	avi_byte11[4]	avi_byte11[3]	avi_byte11[2]	avi_byte11[1]	avi_byte11[0]
0xF460	0x00	main_reg_f460	rw	avi_byte12[7]	avi_byte12[6]	avi_byte12[5]	avi_byte12[4]	avi_byte12[3]	avi_byte12[2]	avi_byte12[1]	avi_byte12[0]
0xF461	0x00	main_reg_f461	rw	avi_byte13[7]	avi_byte13[6]	avi_byte13[5]	avi_byte13[4]	avi_byte13[3]	avi_byte13[2]	avi_byte13[1]	avi_byte13[0]
0xF462	0x00	main_reg_f462	rw	avi_byte14[7]	avi_byte14[6]	avi_byte14[5]	avi_byte14[4]	avi_byte14[3]	avi_byte14[2]	avi_byte14[1]	avi_byte14[0]
0xF463	0x00	main_reg_f463	rw	avi_byte15[7]	avi_byte15[6]	avi_byte15[5]	avi_byte15[4]	avi_byte15[3]	avi_byte15[2]	avi_byte15[1]	avi_byte15[0]
0xF464	0x00	main_reg_f464	rw	avi_byte16[7]	avi_byte16[6]	avi_byte16[5]	avi_byte16[4]	avi_byte16[3]	avi_byte16[2]	avi_byte16[1]	avi_byte16[0]
0xF465	0x00	main_reg_f465	rw	avi_byte17[7]	avi_byte17[6]	avi_byte17[5]	avi_byte17[4]	avi_byte17[3]	avi_byte17[2]	avi_byte17[1]	avi_byte17[0]
0xF466	0x00	main_reg_f466	rw	avi_byte18[7]	avi_byte18[6]	avi_byte18[5]	avi_byte18[4]	avi_byte18[3]	avi_byte18[2]	avi_byte18[1]	avi_byte18[0]
0xF467	0x00	main_reg_f467	rw	avi_byte19[7]	avi_byte19[6]	avi_byte19[5]	avi_byte19[4]	avi_byte19[3]	avi_byte19[2]	avi_byte19[1]	avi_byte19[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF4 68	0x00	main_reg_f468	rw	avi_byte20[7]	avi_byte20[6]	avi_byte20[5]	avi_byte20[4]	avi_byte20[3]	avi_byte20[2]	avi_byte20[1]	avi_byte20[0]
0xF4 69	0x00	main_reg_f469	rw	avi_byte21[7]	avi_byte21[6]	avi_byte21[5]	avi_byte21[4]	avi_byte21[3]	avi_byte21[2]	avi_byte21[1]	avi_byte21[0]
0xF4 6A	0x00	main_reg_f46a	rw	avi_byte22[7]	avi_byte22[6]	avi_byte22[5]	avi_byte22[4]	avi_byte22[3]	avi_byte22[2]	avi_byte22[1]	avi_byte22[0]
0xF4 6B	0x00	main_reg_f46b	rw	avi_byte23[7]	avi_byte23[6]	avi_byte23[5]	avi_byte23[4]	avi_byte23[3]	avi_byte23[2]	avi_byte23[1]	avi_byte23[0]
0xF4 6C	0x00	main_reg_f46c	rw	avi_byte24[7]	avi_byte24[6]	avi_byte24[5]	avi_byte24[4]	avi_byte24[3]	avi_byte24[2]	avi_byte24[1]	avi_byte24[0]
0xF4 6D	0x00	main_reg_f46d	rw	avi_byte25[7]	avi_byte25[6]	avi_byte25[5]	avi_byte25[4]	avi_byte25[3]	avi_byte25[2]	avi_byte25[1]	avi_byte25[0]
0xF4 6E	0x00	main_reg_f46e	rw	avi_byte26[7]	avi_byte26[6]	avi_byte26[5]	avi_byte26[4]	avi_byte26[3]	avi_byte26[2]	avi_byte26[1]	avi_byte26[0]
0xF4 6F	0x00	main_reg_f46f	rw	avi_byte27[7]	avi_byte27[6]	avi_byte27[5]	avi_byte27[4]	avi_byte27[3]	avi_byte27[2]	avi_byte27[1]	avi_byte27[0]
0xF4 70	0x01	main_reg_f470	rw	-	-	-	-	-	audioif_version[2]	audioif_version[1]	audioif_version[0]
0xF4 71	0x0A	main_reg_f471	rw	-	-	-	audioif_length[4]	audioif_length[3]	audioif_length[2]	audioif_length[1]	audioif_length[0]
0xF4 72	0x00	main_reg_f472	rw	audioif_checksum [7]	audioif_checksum [6]	audioif_checksum [5]	audioif_checksum [4]	audioif_checksum [3]	audioif_checksum [2]	audioif_checksum [1]	audioif_checksum [0]
0xF4 73	0x00	main_reg_f473	rw	audioif_ct[3]	audioif_ct[2]	audioif_ct[1]	audioif_ct[0]	audioif_byte1_3	audioif_cc[2]	audioif_cc[1]	audioif_cc[0]
0xF4 74	0x00	main_reg_f474	rw	audioif_byte2_7_ 5[2]	audioif_byte2_7_ 5[1]	audioif_byte2_7_ 5[0]	audioif_sf[2]	audioif_sf[1]	audioif_sf[0]	audioif_ss[1]	audioif_ss[0]
0xF4 75	0x00	main_reg_f475	rw	audioif_byte3[7]	audioif_byte3[6]	audioif_byte3[5]	audioif_byte3[4]	audioif_byte3[3]	audioif_byte3[2]	audioif_byte3[1]	audioif_byte3[0]
0xF4 76	0x00	main_reg_f476	rw	audioif_ca[7]	audioif_ca[6]	audioif_ca[5]	audioif_ca[4]	audioif_ca[3]	audioif_ca[2]	audioif_ca[1]	audioif_ca[0]
0xF4 77	0x00	main_reg_f477	rw	audioif_dm_inh	audioif_lsv[3]	audioif_lsv[2]	audioif_lsv[1]	audioif_lsv[0]	audioif_byte5_2_ 0[2]	audioif_byte5_2_ 0[1]	audioif_byte5_2_ 0[0]
0xF4 78	0x00	main_reg_f478	rw	audioif_byte6[7]	audioif_byte6[6]	audioif_byte6[5]	audioif_byte6[4]	audioif_byte6[3]	audioif_byte6[2]	audioif_byte6[1]	audioif_byte6[0]
0xF4 79	0x00	main_reg_f479	rw	audioif_byte7[7]	audioif_byte7[6]	audioif_byte7[5]	audioif_byte7[4]	audioif_byte7[3]	audioif_byte7[2]	audioif_byte7[1]	audioif_byte7[0]
0xF4 7A	0x00	main_reg_f47a	rw	audioif_byte8[7]	audioif_byte8[6]	audioif_byte8[5]	audioif_byte8[4]	audioif_byte8[3]	audioif_byte8[2]	audioif_byte8[1]	audioif_byte8[0]
0xF4 7B	0x00	main_reg_f47b	rw	audioif_byte9[7]	audioif_byte9[6]	audioif_byte9[5]	audioif_byte9[4]	audioif_byte9[3]	audioif_byte9[2]	audioif_byte9[1]	audioif_byte9[0]
0xF4 7C	0x00	main_reg_f47c	rw	audioif_byte10[7]	audioif_byte10[6]	audioif_byte10[5]	audioif_byte10[4]	audioif_byte10[3]	audioif_byte10[2]	audioif_byte10[1]	audioif_byte10[0]
0xF4 80	0x7F	main_reg_f480	rw	-	-	-	-	pre_en_ch0	pre_en_ch1	pre_en_ch2	pre_en_clk
0xF4 81	0x88	main_reg_f481	rw	chg_inj_ch0[3]	chg_inj_ch0[2]	chg_inj_ch0[1]	chg_inj_ch0[0]	chg_inj_ch1[3]	chg_inj_ch1[2]	chg_inj_ch1[1]	chg_inj_ch1[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF482	0x88	main_reg_f482	rw	chg_inj_ch2[3]	chg_inj_ch2[2]	chg_inj_ch2[1]	chg_inj_ch2[0]	chg_inj_clk[3]	chg_inj_clk[2]	chg_inj_clk[1]	chg_inj_clk[0]
0xF483	0x00	main_reg_f483	rw	ch0_sterm_disable	-	-	-	-	-	-	-
0xF484	0x00	main_reg_f484	rw	ch1_sterm_disable	-	-	-	-	-	-	-
0xF485	0x00	main_reg_f485	rw	ch2_sterm_disable	-	-	-	-	-	-	-
0xF486	0x00	main_reg_f486	rw	clk_sterm_disable	-	-	-	-	-	-	-
0xF494	0xC0	main_reg_f494	rw	hpd_int_en	rx_sense_int_en	vsync_int_en	-	-	edid_ready_int_en	hdcp_authenticated_int_en	ri_ready_int_en
0xF495	0x00	main_reg_f495	rw	hdcp_error_int_en	bksv_flag_int_en	-	-	-	-	-	-
0xF496	0x00	main_reg_f496	rw	hpd_int	rx_sense_int	vsync_int	-	-	edid_ready_int	hdcp_authenticated_int	ri_ready_int
0xF497	0x00	main_reg_f497	rw	hdcp_error_int	bksv_flag_int	-	cec_tx_arbitration_lost_int	cec_tx_retry_timeout_int	-	-	-
0xF498	0x00	main_reg_f498	rw	-	-	-	cec_pd	-	-	-	-
0xF49E	0x00	main_reg_f49e	rw	high_freq_video[1]	high_freq_video[0]	video_offset_ctrl[1]	video_offset_ctrl[0]	-	-	-	-
0xF49F	0x00	main_reg_f49f	rw	-	-	hpd_override[1]	hpd_override[0]	-	-	-	-
0xF4AB	0x20	main_reg_f4ab	rw	-	-	hdcp_start_delay[2]	hdcp_start_delay[1]	hdcp_start_delay[0]	-	-	-
0xF4AE	0x00	main_reg_f4ae	rw	-	hdcp_1p1_dis	r0_wait_time[1]	r0_wait_time[0]	hdcp_repeater_timeout[1]	hdcp_repeater_timeout[0]	-	-
0xF4AF	0x14	main_reg_f4af	rw	hdcp_desired	-	-	frame_enc	-	hdmi_dvi_sel_en	hdmi_dvi_sel	-
0xF4B0	0x00	main_reg_f4b0	r	an[7]	an[6]	an[5]	an[4]	an[3]	an[2]	an[1]	an[0]
0xF4B1	0x00	main_reg_f4b1	r	an[15]	an[14]	an[13]	an[12]	an[11]	an[10]	an[9]	an[8]
0xF4B2	0x00	main_reg_f4b2	r	an[23]	an[22]	an[21]	an[20]	an[19]	an[18]	an[17]	an[16]
0xF4B3	0x00	main_reg_f4b3	r	an[31]	an[30]	an[29]	an[28]	an[27]	an[26]	an[25]	an[24]
0xF4B4	0x00	main_reg_f4b4	r	an[39]	an[38]	an[37]	an[36]	an[35]	an[34]	an[33]	an[32]
0xF4B5	0x00	main_reg_f4b5	r	an[47]	an[46]	an[45]	an[44]	an[43]	an[42]	an[41]	an[40]
0xF4B6	0x00	main_reg_f4b6	r	an[55]	an[54]	an[53]	an[52]	an[51]	an[50]	an[49]	an[48]
0xF4B7	0x00	main_reg_f4b7	r	an[63]	an[62]	an[61]	an[60]	an[59]	an[58]	an[57]	an[56]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF4 B8	0x00	main_reg_f4b8	r	-	enc_on	-	keys_read_error	-	-	-	-
0xF4 BE	0x00	main_reg_f4be	r	bcaps[7]	bcaps[6]	bcaps[5]	bcaps[4]	bcaps[3]	bcaps[2]	bcaps[1]	bcaps[0]
0xF4 BF	0x00	main_reg_f4bf	r	bksv[7]	bksv[6]	bksv[5]	bksv[4]	bksv[3]	bksv[2]	bksv[1]	bksv[0]
0xF4 C0	0x00	main_reg_f4c0	r	bksv[15]	bksv[14]	bksv[13]	bksv[12]	bksv[11]	bksv[10]	bksv[9]	bksv[8]
0xF4 C1	0x00	main_reg_f4c1	r	bksv[23]	bksv[22]	bksv[21]	bksv[20]	bksv[19]	bksv[18]	bksv[17]	bksv[16]
0xF4 C2	0x00	main_reg_f4c2	r	bksv[31]	bksv[30]	bksv[29]	bksv[28]	bksv[27]	bksv[26]	bksv[25]	bksv[24]
0xF4 C3	0x00	main_reg_f4c3	r	bksv[39]	bksv[38]	bksv[37]	bksv[36]	bksv[35]	bksv[34]	bksv[33]	bksv[32]
0xF4 C4	0x00	main_reg_f4c4	rw	edid_segment[7]	edid_segment[6]	edid_segment[5]	edid_segment[4]	edid_segment[3]	edid_segment[2]	edid_segment[1]	edid_segment[0]
0xF4 C5	0x00	main_reg_f4c5	r	-	an_stop	-	-	-	ri_flag	bksv_update_flag	pj_flag
0xF4 C6	0x00	main_reg_f4c6	r	-	-	-	hdmi_mode	hdcp_requested	rx_sense	-	tmds_output_en
0xF4 C7	0x00	main_reg_f4c7	rw	bksv_flag	bksv_count[6]	bksv_count[5]	bksv_count[4]	bksv_count[3]	bksv_count[2]	bksv_count[1]	bksv_count[0]
0xF4 C8	0x00	main_reg_f4c8	r	hdcp_controller_error[3]	hdcp_controller_error[2]	hdcp_controller_error[1]	hdcp_controller_error[0]	hdcp_controller_state[3]	hdcp_controller_state[2]	hdcp_controller_state[1]	hdcp_controller_state[0]
0xF4 C9	0x03	main_reg_f4c9	rw	-	-	-	edid_reread	edid_tries[3]	edid_tries[2]	edid_tries[1]	edid_tries[0]
0xF4 CA	0x00	main_reg_f4ca	r	hdcp_bstatus[15]	hdcp_bstatus[14]	hdcp_bstatus[13]	hdcp_bstatus[12]	hdcp_bstatus[11]	hdcp_bstatus[10]	hdcp_bstatus[9]	hdcp_bstatus[8]
0xF4 CB	0x00	main_reg_f4cb	r	hdcp_bstatus[7]	hdcp_bstatus[6]	hdcp_bstatus[5]	hdcp_bstatus[4]	hdcp_bstatus[3]	hdcp_bstatus[2]	hdcp_bstatus[1]	hdcp_bstatus[0]
0xF4 E4	0x00	main_reg_f4e4	r	pll_lock_status	-	-	-	-	-	-	-
0xF4 E6	0x00	main_reg_f4e6	rw	-	-	rx_sense_pd	-	-	-	-	-
0xF4 EA	0x84	main_reg_f4ea	rw	-	-	-	-	-	-	tmds_clk_invert	cci_controls

1.12 TX2 PACKET MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 00	0x00	packetmemory_reg_fa00	rw	spd_header_byte_0[7]	spd_header_byte_0[6]	spd_header_byte_0[5]	spd_header_byte_0[4]	spd_header_byte_0[3]	spd_header_byte_0[2]	spd_header_byte_0[1]	spd_header_byte_0[0]
0xFA 01	0x00	packetmemory_reg_fa01	rw	spd_header_byte_1[7]	spd_header_byte_1[6]	spd_header_byte_1[5]	spd_header_byte_1[4]	spd_header_byte_1[3]	spd_header_byte_1[2]	spd_header_byte_1[1]	spd_header_byte_1[0]
0xFA 02	0x00	packetmemory_reg_fa02	rw	spd_header_byte_2[7]	spd_header_byte_2[6]	spd_header_byte_2[5]	spd_header_byte_2[4]	spd_header_byte_2[3]	spd_header_byte_2[2]	spd_header_byte_2[1]	spd_header_byte_2[0]
0xFA 03	0x00	packetmemory_reg_fa03	rw	spd_packet_byte_0[7]	spd_packet_byte_0[6]	spd_packet_byte_0[5]	spd_packet_byte_0[4]	spd_packet_byte_0[3]	spd_packet_byte_0[2]	spd_packet_byte_0[1]	spd_packet_byte_0[0]
0xFA 04	0x00	packetmemory_reg_fa04	rw	spd_packet_byte_1[7]	spd_packet_byte_1[6]	spd_packet_byte_1[5]	spd_packet_byte_1[4]	spd_packet_byte_1[3]	spd_packet_byte_1[2]	spd_packet_byte_1[1]	spd_packet_byte_1[0]
0xFA 05	0x00	packetmemory_reg_fa05	rw	spd_packet_byte_2[7]	spd_packet_byte_2[6]	spd_packet_byte_2[5]	spd_packet_byte_2[4]	spd_packet_byte_2[3]	spd_packet_byte_2[2]	spd_packet_byte_2[1]	spd_packet_byte_2[0]
0xFA 06	0x00	packetmemory_reg_fa06	rw	spd_packet_byte_3[7]	spd_packet_byte_3[6]	spd_packet_byte_3[5]	spd_packet_byte_3[4]	spd_packet_byte_3[3]	spd_packet_byte_3[2]	spd_packet_byte_3[1]	spd_packet_byte_3[0]
0xFA 07	0x00	packetmemory_reg_fa07	rw	spd_packet_byte_4[7]	spd_packet_byte_4[6]	spd_packet_byte_4[5]	spd_packet_byte_4[4]	spd_packet_byte_4[3]	spd_packet_byte_4[2]	spd_packet_byte_4[1]	spd_packet_byte_4[0]
0xFA 08	0x00	packetmemory_reg_fa08	rw	spd_packet_byte_5[7]	spd_packet_byte_5[6]	spd_packet_byte_5[5]	spd_packet_byte_5[4]	spd_packet_byte_5[3]	spd_packet_byte_5[2]	spd_packet_byte_5[1]	spd_packet_byte_5[0]
0xFA 09	0x00	packetmemory_reg_fa09	rw	spd_packet_byte_6[7]	spd_packet_byte_6[6]	spd_packet_byte_6[5]	spd_packet_byte_6[4]	spd_packet_byte_6[3]	spd_packet_byte_6[2]	spd_packet_byte_6[1]	spd_packet_byte_6[0]
0xFA 0A	0x00	packetmemory_reg_fa0a	rw	spd_packet_byte_7[7]	spd_packet_byte_7[6]	spd_packet_byte_7[5]	spd_packet_byte_7[4]	spd_packet_byte_7[3]	spd_packet_byte_7[2]	spd_packet_byte_7[1]	spd_packet_byte_7[0]
0xFA 0B	0x00	packetmemory_reg_fa0b	rw	spd_packet_byte_8[7]	spd_packet_byte_8[6]	spd_packet_byte_8[5]	spd_packet_byte_8[4]	spd_packet_byte_8[3]	spd_packet_byte_8[2]	spd_packet_byte_8[1]	spd_packet_byte_8[0]
0xFA 0C	0x00	packetmemory_reg_fa0c	rw	spd_packet_byte_9[7]	spd_packet_byte_9[6]	spd_packet_byte_9[5]	spd_packet_byte_9[4]	spd_packet_byte_9[3]	spd_packet_byte_9[2]	spd_packet_byte_9[1]	spd_packet_byte_9[0]
0xFA 0D	0x00	packetmemory_reg_fa0d	rw	spd_packet_byte_10[7]	spd_packet_byte_10[6]	spd_packet_byte_10[5]	spd_packet_byte_10[4]	spd_packet_byte_10[3]	spd_packet_byte_10[2]	spd_packet_byte_10[1]	spd_packet_byte_10[0]
0xFA 0E	0x00	packetmemory_reg_fa0e	rw	spd_packet_byte_11[7]	spd_packet_byte_11[6]	spd_packet_byte_11[5]	spd_packet_byte_11[4]	spd_packet_byte_11[3]	spd_packet_byte_11[2]	spd_packet_byte_11[1]	spd_packet_byte_11[0]
0xFA 0F	0x00	packetmemory_reg_fa0f	rw	spd_packet_byte_12[7]	spd_packet_byte_12[6]	spd_packet_byte_12[5]	spd_packet_byte_12[4]	spd_packet_byte_12[3]	spd_packet_byte_12[2]	spd_packet_byte_12[1]	spd_packet_byte_12[0]
0xFA 10	0x00	packetmemory_reg_fa10	rw	spd_packet_byte_13[7]	spd_packet_byte_13[6]	spd_packet_byte_13[5]	spd_packet_byte_13[4]	spd_packet_byte_13[3]	spd_packet_byte_13[2]	spd_packet_byte_13[1]	spd_packet_byte_13[0]
0xFA 11	0x00	packetmemory_reg_fa11	rw	spd_packet_byte_14[7]	spd_packet_byte_14[6]	spd_packet_byte_14[5]	spd_packet_byte_14[4]	spd_packet_byte_14[3]	spd_packet_byte_14[2]	spd_packet_byte_14[1]	spd_packet_byte_14[0]
0xFA 12	0x00	packetmemory_reg_fa12	rw	spd_packet_byte_15[7]	spd_packet_byte_15[6]	spd_packet_byte_15[5]	spd_packet_byte_15[4]	spd_packet_byte_15[3]	spd_packet_byte_15[2]	spd_packet_byte_15[1]	spd_packet_byte_15[0]
0xFA 13	0x00	packetmemory_reg_fa13	rw	spd_packet_byte_16[7]	spd_packet_byte_16[6]	spd_packet_byte_16[5]	spd_packet_byte_16[4]	spd_packet_byte_16[3]	spd_packet_byte_16[2]	spd_packet_byte_16[1]	spd_packet_byte_16[0]
0xFA 14	0x00	packetmemory_reg_fa14	rw	spd_packet_byte_17[7]	spd_packet_byte_17[6]	spd_packet_byte_17[5]	spd_packet_byte_17[4]	spd_packet_byte_17[3]	spd_packet_byte_17[2]	spd_packet_byte_17[1]	spd_packet_byte_17[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 15	0x00	packetmemory_reg_fa15	rw	spd_packet_byte_18[7]	spd_packet_byte_18[6]	spd_packet_byte_18[5]	spd_packet_byte_18[4]	spd_packet_byte_18[3]	spd_packet_byte_18[2]	spd_packet_byte_18[1]	spd_packet_byte_18[0]
0xFA 16	0x00	packetmemory_reg_fa16	rw	spd_packet_byte_19[7]	spd_packet_byte_19[6]	spd_packet_byte_19[5]	spd_packet_byte_19[4]	spd_packet_byte_19[3]	spd_packet_byte_19[2]	spd_packet_byte_19[1]	spd_packet_byte_19[0]
0xFA 17	0x00	packetmemory_reg_fa17	rw	spd_packet_byte_20[7]	spd_packet_byte_20[6]	spd_packet_byte_20[5]	spd_packet_byte_20[4]	spd_packet_byte_20[3]	spd_packet_byte_20[2]	spd_packet_byte_20[1]	spd_packet_byte_20[0]
0xFA 18	0x00	packetmemory_reg_fa18	rw	spd_packet_byte_21[7]	spd_packet_byte_21[6]	spd_packet_byte_21[5]	spd_packet_byte_21[4]	spd_packet_byte_21[3]	spd_packet_byte_21[2]	spd_packet_byte_21[1]	spd_packet_byte_21[0]
0xFA 19	0x00	packetmemory_reg_fa19	rw	spd_packet_byte_22[7]	spd_packet_byte_22[6]	spd_packet_byte_22[5]	spd_packet_byte_22[4]	spd_packet_byte_22[3]	spd_packet_byte_22[2]	spd_packet_byte_22[1]	spd_packet_byte_22[0]
0xFA 1A	0x00	packetmemory_reg_fa1a	rw	spd_packet_byte_23[7]	spd_packet_byte_23[6]	spd_packet_byte_23[5]	spd_packet_byte_23[4]	spd_packet_byte_23[3]	spd_packet_byte_23[2]	spd_packet_byte_23[1]	spd_packet_byte_23[0]
0xFA 1B	0x00	packetmemory_reg_fa1b	rw	spd_packet_byte_24[7]	spd_packet_byte_24[6]	spd_packet_byte_24[5]	spd_packet_byte_24[4]	spd_packet_byte_24[3]	spd_packet_byte_24[2]	spd_packet_byte_24[1]	spd_packet_byte_24[0]
0xFA 1C	0x00	packetmemory_reg_fa1c	rw	spd_packet_byte_25[7]	spd_packet_byte_25[6]	spd_packet_byte_25[5]	spd_packet_byte_25[4]	spd_packet_byte_25[3]	spd_packet_byte_25[2]	spd_packet_byte_25[1]	spd_packet_byte_25[0]
0xFA 1D	0x00	packetmemory_reg_fa1d	rw	spd_packet_byte_26[7]	spd_packet_byte_26[6]	spd_packet_byte_26[5]	spd_packet_byte_26[4]	spd_packet_byte_26[3]	spd_packet_byte_26[2]	spd_packet_byte_26[1]	spd_packet_byte_26[0]
0xFA 1E	0x00	packetmemory_reg_fa1e	rw	spd_packet_byte_27[7]	spd_packet_byte_27[6]	spd_packet_byte_27[5]	spd_packet_byte_27[4]	spd_packet_byte_27[3]	spd_packet_byte_27[2]	spd_packet_byte_27[1]	spd_packet_byte_27[0]
0xFA 1F	0x00	packetmemory_reg_fa1f	rw	spd_update	-	-	-	-	-	-	-
0xFA 20	0x00	packetmemory_reg_fa20	rw	mpeg_header_byte_0[7]	mpeg_header_byte_0[6]	mpeg_header_byte_0[5]	mpeg_header_byte_0[4]	mpeg_header_byte_0[3]	mpeg_header_byte_0[2]	mpeg_header_byte_0[1]	mpeg_header_byte_0[0]
0xFA 21	0x00	packetmemory_reg_fa21	rw	mpeg_header_byte_1[7]	mpeg_header_byte_1[6]	mpeg_header_byte_1[5]	mpeg_header_byte_1[4]	mpeg_header_byte_1[3]	mpeg_header_byte_1[2]	mpeg_header_byte_1[1]	mpeg_header_byte_1[0]
0xFA 22	0x00	packetmemory_reg_fa22	rw	mpeg_header_byte_2[7]	mpeg_header_byte_2[6]	mpeg_header_byte_2[5]	mpeg_header_byte_2[4]	mpeg_header_byte_2[3]	mpeg_header_byte_2[2]	mpeg_header_byte_2[1]	mpeg_header_byte_2[0]
0xFA 23	0x00	packetmemory_reg_fa23	rw	mpeg_packet_byte_0[7]	mpeg_packet_byte_0[6]	mpeg_packet_byte_0[5]	mpeg_packet_byte_0[4]	mpeg_packet_byte_0[3]	mpeg_packet_byte_0[2]	mpeg_packet_byte_0[1]	mpeg_packet_byte_0[0]
0xFA 24	0x00	packetmemory_reg_fa24	rw	mpeg_packet_byte_1[7]	mpeg_packet_byte_1[6]	mpeg_packet_byte_1[5]	mpeg_packet_byte_1[4]	mpeg_packet_byte_1[3]	mpeg_packet_byte_1[2]	mpeg_packet_byte_1[1]	mpeg_packet_byte_1[0]
0xFA 25	0x00	packetmemory_reg_fa25	rw	mpeg_packet_byte_2[7]	mpeg_packet_byte_2[6]	mpeg_packet_byte_2[5]	mpeg_packet_byte_2[4]	mpeg_packet_byte_2[3]	mpeg_packet_byte_2[2]	mpeg_packet_byte_2[1]	mpeg_packet_byte_2[0]
0xFA 26	0x00	packetmemory_reg_fa26	rw	mpeg_packet_byte_3[7]	mpeg_packet_byte_3[6]	mpeg_packet_byte_3[5]	mpeg_packet_byte_3[4]	mpeg_packet_byte_3[3]	mpeg_packet_byte_3[2]	mpeg_packet_byte_3[1]	mpeg_packet_byte_3[0]
0xFA 27	0x00	packetmemory_reg_fa27	rw	mpeg_packet_byte_4[7]	mpeg_packet_byte_4[6]	mpeg_packet_byte_4[5]	mpeg_packet_byte_4[4]	mpeg_packet_byte_4[3]	mpeg_packet_byte_4[2]	mpeg_packet_byte_4[1]	mpeg_packet_byte_4[0]
0xFA 28	0x00	packetmemory_reg_fa28	rw	mpeg_packet_byte_5[7]	mpeg_packet_byte_5[6]	mpeg_packet_byte_5[5]	mpeg_packet_byte_5[4]	mpeg_packet_byte_5[3]	mpeg_packet_byte_5[2]	mpeg_packet_byte_5[1]	mpeg_packet_byte_5[0]
0xFA 29	0x00	packetmemory_reg_fa29	rw	mpeg_packet_byte_6[7]	mpeg_packet_byte_6[6]	mpeg_packet_byte_6[5]	mpeg_packet_byte_6[4]	mpeg_packet_byte_6[3]	mpeg_packet_byte_6[2]	mpeg_packet_byte_6[1]	mpeg_packet_byte_6[0]
0xFA 2A	0x00	packetmemory_reg_fa2a	rw	mpeg_packet_byte_7[7]	mpeg_packet_byte_7[6]	mpeg_packet_byte_7[5]	mpeg_packet_byte_7[4]	mpeg_packet_byte_7[3]	mpeg_packet_byte_7[2]	mpeg_packet_byte_7[1]	mpeg_packet_byte_7[0]
0xFA 2B	0x00	packetmemory_reg_fa2b	rw	mpeg_packet_byte_8[7]	mpeg_packet_byte_8[6]	mpeg_packet_byte_8[5]	mpeg_packet_byte_8[4]	mpeg_packet_byte_8[3]	mpeg_packet_byte_8[2]	mpeg_packet_byte_8[1]	mpeg_packet_byte_8[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 2C	0x00	packetmemory_reg_fa2c	rw	mpeg_packet_byte_9[7]	mpeg_packet_byte_9[6]	mpeg_packet_byte_9[5]	mpeg_packet_byte_9[4]	mpeg_packet_byte_9[3]	mpeg_packet_byte_9[2]	mpeg_packet_byte_9[1]	mpeg_packet_byte_9[0]
0xFA 2D	0x00	packetmemory_reg_fa2d	rw	mpeg_packet_byte_10[7]	mpeg_packet_byte_10[6]	mpeg_packet_byte_10[5]	mpeg_packet_byte_10[4]	mpeg_packet_byte_10[3]	mpeg_packet_byte_10[2]	mpeg_packet_byte_10[1]	mpeg_packet_byte_10[0]
0xFA 2E	0x00	packetmemory_reg_fa2e	rw	mpeg_packet_byte_11[7]	mpeg_packet_byte_11[6]	mpeg_packet_byte_11[5]	mpeg_packet_byte_11[4]	mpeg_packet_byte_11[3]	mpeg_packet_byte_11[2]	mpeg_packet_byte_11[1]	mpeg_packet_byte_11[0]
0xFA 2F	0x00	packetmemory_reg_fa2f	rw	mpeg_packet_byte_12[7]	mpeg_packet_byte_12[6]	mpeg_packet_byte_12[5]	mpeg_packet_byte_12[4]	mpeg_packet_byte_12[3]	mpeg_packet_byte_12[2]	mpeg_packet_byte_12[1]	mpeg_packet_byte_12[0]
0xFA 30	0x00	packetmemory_reg_fa30	rw	mpeg_packet_byte_13[7]	mpeg_packet_byte_13[6]	mpeg_packet_byte_13[5]	mpeg_packet_byte_13[4]	mpeg_packet_byte_13[3]	mpeg_packet_byte_13[2]	mpeg_packet_byte_13[1]	mpeg_packet_byte_13[0]
0xFA 31	0x00	packetmemory_reg_fa31	rw	mpeg_packet_byte_14[7]	mpeg_packet_byte_14[6]	mpeg_packet_byte_14[5]	mpeg_packet_byte_14[4]	mpeg_packet_byte_14[3]	mpeg_packet_byte_14[2]	mpeg_packet_byte_14[1]	mpeg_packet_byte_14[0]
0xFA 32	0x00	packetmemory_reg_fa32	rw	mpeg_packet_byte_15[7]	mpeg_packet_byte_15[6]	mpeg_packet_byte_15[5]	mpeg_packet_byte_15[4]	mpeg_packet_byte_15[3]	mpeg_packet_byte_15[2]	mpeg_packet_byte_15[1]	mpeg_packet_byte_15[0]
0xFA 33	0x00	packetmemory_reg_fa33	rw	mpeg_packet_byte_16[7]	mpeg_packet_byte_16[6]	mpeg_packet_byte_16[5]	mpeg_packet_byte_16[4]	mpeg_packet_byte_16[3]	mpeg_packet_byte_16[2]	mpeg_packet_byte_16[1]	mpeg_packet_byte_16[0]
0xFA 34	0x00	packetmemory_reg_fa34	rw	mpeg_packet_byte_17[7]	mpeg_packet_byte_17[6]	mpeg_packet_byte_17[5]	mpeg_packet_byte_17[4]	mpeg_packet_byte_17[3]	mpeg_packet_byte_17[2]	mpeg_packet_byte_17[1]	mpeg_packet_byte_17[0]
0xFA 35	0x00	packetmemory_reg_fa35	rw	mpeg_packet_byte_18[7]	mpeg_packet_byte_18[6]	mpeg_packet_byte_18[5]	mpeg_packet_byte_18[4]	mpeg_packet_byte_18[3]	mpeg_packet_byte_18[2]	mpeg_packet_byte_18[1]	mpeg_packet_byte_18[0]
0xFA 36	0x00	packetmemory_reg_fa36	rw	mpeg_packet_byte_19[7]	mpeg_packet_byte_19[6]	mpeg_packet_byte_19[5]	mpeg_packet_byte_19[4]	mpeg_packet_byte_19[3]	mpeg_packet_byte_19[2]	mpeg_packet_byte_19[1]	mpeg_packet_byte_19[0]
0xFA 37	0x00	packetmemory_reg_fa37	rw	mpeg_packet_byte_20[7]	mpeg_packet_byte_20[6]	mpeg_packet_byte_20[5]	mpeg_packet_byte_20[4]	mpeg_packet_byte_20[3]	mpeg_packet_byte_20[2]	mpeg_packet_byte_20[1]	mpeg_packet_byte_20[0]
0xFA 38	0x00	packetmemory_reg_fa38	rw	mpeg_packet_byte_21[7]	mpeg_packet_byte_21[6]	mpeg_packet_byte_21[5]	mpeg_packet_byte_21[4]	mpeg_packet_byte_21[3]	mpeg_packet_byte_21[2]	mpeg_packet_byte_21[1]	mpeg_packet_byte_21[0]
0xFA 39	0x00	packetmemory_reg_fa39	rw	mpeg_packet_byte_22[7]	mpeg_packet_byte_22[6]	mpeg_packet_byte_22[5]	mpeg_packet_byte_22[4]	mpeg_packet_byte_22[3]	mpeg_packet_byte_22[2]	mpeg_packet_byte_22[1]	mpeg_packet_byte_22[0]
0xFA 3A	0x00	packetmemory_reg_fa3a	rw	mpeg_packet_byte_23[7]	mpeg_packet_byte_23[6]	mpeg_packet_byte_23[5]	mpeg_packet_byte_23[4]	mpeg_packet_byte_23[3]	mpeg_packet_byte_23[2]	mpeg_packet_byte_23[1]	mpeg_packet_byte_23[0]
0xFA 3B	0x00	packetmemory_reg_fa3b	rw	mpeg_packet_byte_24[7]	mpeg_packet_byte_24[6]	mpeg_packet_byte_24[5]	mpeg_packet_byte_24[4]	mpeg_packet_byte_24[3]	mpeg_packet_byte_24[2]	mpeg_packet_byte_24[1]	mpeg_packet_byte_24[0]
0xFA 3C	0x00	packetmemory_reg_fa3c	rw	mpeg_packet_byte_25[7]	mpeg_packet_byte_25[6]	mpeg_packet_byte_25[5]	mpeg_packet_byte_25[4]	mpeg_packet_byte_25[3]	mpeg_packet_byte_25[2]	mpeg_packet_byte_25[1]	mpeg_packet_byte_25[0]
0xFA 3D	0x00	packetmemory_reg_fa3d	rw	mpeg_packet_byte_26[7]	mpeg_packet_byte_26[6]	mpeg_packet_byte_26[5]	mpeg_packet_byte_26[4]	mpeg_packet_byte_26[3]	mpeg_packet_byte_26[2]	mpeg_packet_byte_26[1]	mpeg_packet_byte_26[0]
0xFA 3E	0x00	packetmemory_reg_fa3e	rw	mpeg_packet_byte_27[7]	mpeg_packet_byte_27[6]	mpeg_packet_byte_27[5]	mpeg_packet_byte_27[4]	mpeg_packet_byte_27[3]	mpeg_packet_byte_27[2]	mpeg_packet_byte_27[1]	mpeg_packet_byte_27[0]
0xFA 3F	0x00	packetmemory_reg_fa3f	rw	mpeg_update	-	-	-	-	-	-	-
0xFA 40	0x00	packetmemory_reg_fa40	rw	acp_header_byte_0[7]	acp_header_byte_0[6]	acp_header_byte_0[5]	acp_header_byte_0[4]	acp_header_byte_0[3]	acp_header_byte_0[2]	acp_header_byte_0[1]	acp_header_byte_0[0]
0xFA 41	0x00	packetmemory_reg_fa41	rw	acp_header_byte_1[7]	acp_header_byte_1[6]	acp_header_byte_1[5]	acp_header_byte_1[4]	acp_header_byte_1[3]	acp_header_byte_1[2]	acp_header_byte_1[1]	acp_header_byte_1[0]
0xFA 42	0x00	packetmemory_reg_fa42	rw	acp_header_byte_2[7]	acp_header_byte_2[6]	acp_header_byte_2[5]	acp_header_byte_2[4]	acp_header_byte_2[3]	acp_header_byte_2[2]	acp_header_byte_2[1]	acp_header_byte_2[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 43	0x00	packetmemory_reg_fa43	rw	acp_packet_byte_0[7]	acp_packet_byte_0[6]	acp_packet_byte_0[5]	acp_packet_byte_0[4]	acp_packet_byte_0[3]	acp_packet_byte_0[2]	acp_packet_byte_0[1]	acp_packet_byte_0[0]
0xFA 44	0x00	packetmemory_reg_fa44	rw	acp_packet_byte_1[7]	acp_packet_byte_1[6]	acp_packet_byte_1[5]	acp_packet_byte_1[4]	acp_packet_byte_1[3]	acp_packet_byte_1[2]	acp_packet_byte_1[1]	acp_packet_byte_1[0]
0xFA 45	0x00	packetmemory_reg_fa45	rw	acp_packet_byte_2[7]	acp_packet_byte_2[6]	acp_packet_byte_2[5]	acp_packet_byte_2[4]	acp_packet_byte_2[3]	acp_packet_byte_2[2]	acp_packet_byte_2[1]	acp_packet_byte_2[0]
0xFA 46	0x00	packetmemory_reg_fa46	rw	acp_packet_byte_3[7]	acp_packet_byte_3[6]	acp_packet_byte_3[5]	acp_packet_byte_3[4]	acp_packet_byte_3[3]	acp_packet_byte_3[2]	acp_packet_byte_3[1]	acp_packet_byte_3[0]
0xFA 47	0x00	packetmemory_reg_fa47	rw	acp_packet_byte_4[7]	acp_packet_byte_4[6]	acp_packet_byte_4[5]	acp_packet_byte_4[4]	acp_packet_byte_4[3]	acp_packet_byte_4[2]	acp_packet_byte_4[1]	acp_packet_byte_4[0]
0xFA 48	0x00	packetmemory_reg_fa48	rw	acp_packet_byte_5[7]	acp_packet_byte_5[6]	acp_packet_byte_5[5]	acp_packet_byte_5[4]	acp_packet_byte_5[3]	acp_packet_byte_5[2]	acp_packet_byte_5[1]	acp_packet_byte_5[0]
0xFA 49	0x00	packetmemory_reg_fa49	rw	acp_packet_byte_6[7]	acp_packet_byte_6[6]	acp_packet_byte_6[5]	acp_packet_byte_6[4]	acp_packet_byte_6[3]	acp_packet_byte_6[2]	acp_packet_byte_6[1]	acp_packet_byte_6[0]
0xFA 4A	0x00	packetmemory_reg_fa4a	rw	acp_packet_byte_7[7]	acp_packet_byte_7[6]	acp_packet_byte_7[5]	acp_packet_byte_7[4]	acp_packet_byte_7[3]	acp_packet_byte_7[2]	acp_packet_byte_7[1]	acp_packet_byte_7[0]
0xFA 4B	0x00	packetmemory_reg_fa4b	rw	acp_packet_byte_8[7]	acp_packet_byte_8[6]	acp_packet_byte_8[5]	acp_packet_byte_8[4]	acp_packet_byte_8[3]	acp_packet_byte_8[2]	acp_packet_byte_8[1]	acp_packet_byte_8[0]
0xFA 4C	0x00	packetmemory_reg_fa4c	rw	acp_packet_byte_9[7]	acp_packet_byte_9[6]	acp_packet_byte_9[5]	acp_packet_byte_9[4]	acp_packet_byte_9[3]	acp_packet_byte_9[2]	acp_packet_byte_9[1]	acp_packet_byte_9[0]
0xFA 4D	0x00	packetmemory_reg_fa4d	rw	acp_packet_byte_10[7]	acp_packet_byte_10[6]	acp_packet_byte_10[5]	acp_packet_byte_10[4]	acp_packet_byte_10[3]	acp_packet_byte_10[2]	acp_packet_byte_10[1]	acp_packet_byte_10[0]
0xFA 4E	0x00	packetmemory_reg_fa4e	rw	acp_packet_byte_11[7]	acp_packet_byte_11[6]	acp_packet_byte_11[5]	acp_packet_byte_11[4]	acp_packet_byte_11[3]	acp_packet_byte_11[2]	acp_packet_byte_11[1]	acp_packet_byte_11[0]
0xFA 4F	0x00	packetmemory_reg_fa4f	rw	acp_packet_byte_12[7]	acp_packet_byte_12[6]	acp_packet_byte_12[5]	acp_packet_byte_12[4]	acp_packet_byte_12[3]	acp_packet_byte_12[2]	acp_packet_byte_12[1]	acp_packet_byte_12[0]
0xFA 50	0x00	packetmemory_reg_fa50	rw	acp_packet_byte_13[7]	acp_packet_byte_13[6]	acp_packet_byte_13[5]	acp_packet_byte_13[4]	acp_packet_byte_13[3]	acp_packet_byte_13[2]	acp_packet_byte_13[1]	acp_packet_byte_13[0]
0xFA 51	0x00	packetmemory_reg_fa51	rw	acp_packet_byte_14[7]	acp_packet_byte_14[6]	acp_packet_byte_14[5]	acp_packet_byte_14[4]	acp_packet_byte_14[3]	acp_packet_byte_14[2]	acp_packet_byte_14[1]	acp_packet_byte_14[0]
0xFA 52	0x00	packetmemory_reg_fa52	rw	acp_packet_byte_15[7]	acp_packet_byte_15[6]	acp_packet_byte_15[5]	acp_packet_byte_15[4]	acp_packet_byte_15[3]	acp_packet_byte_15[2]	acp_packet_byte_15[1]	acp_packet_byte_15[0]
0xFA 53	0x00	packetmemory_reg_fa53	rw	acp_packet_byte_16[7]	acp_packet_byte_16[6]	acp_packet_byte_16[5]	acp_packet_byte_16[4]	acp_packet_byte_16[3]	acp_packet_byte_16[2]	acp_packet_byte_16[1]	acp_packet_byte_16[0]
0xFA 54	0x00	packetmemory_reg_fa54	rw	acp_packet_byte_17[7]	acp_packet_byte_17[6]	acp_packet_byte_17[5]	acp_packet_byte_17[4]	acp_packet_byte_17[3]	acp_packet_byte_17[2]	acp_packet_byte_17[1]	acp_packet_byte_17[0]
0xFA 55	0x00	packetmemory_reg_fa55	rw	acp_packet_byte_18[7]	acp_packet_byte_18[6]	acp_packet_byte_18[5]	acp_packet_byte_18[4]	acp_packet_byte_18[3]	acp_packet_byte_18[2]	acp_packet_byte_18[1]	acp_packet_byte_18[0]
0xFA 56	0x00	packetmemory_reg_fa56	rw	acp_packet_byte_19[7]	acp_packet_byte_19[6]	acp_packet_byte_19[5]	acp_packet_byte_19[4]	acp_packet_byte_19[3]	acp_packet_byte_19[2]	acp_packet_byte_19[1]	acp_packet_byte_19[0]
0xFA 57	0x00	packetmemory_reg_fa57	rw	acp_packet_byte_20[7]	acp_packet_byte_20[6]	acp_packet_byte_20[5]	acp_packet_byte_20[4]	acp_packet_byte_20[3]	acp_packet_byte_20[2]	acp_packet_byte_20[1]	acp_packet_byte_20[0]
0xFA 58	0x00	packetmemory_reg_fa58	rw	acp_packet_byte_21[7]	acp_packet_byte_21[6]	acp_packet_byte_21[5]	acp_packet_byte_21[4]	acp_packet_byte_21[3]	acp_packet_byte_21[2]	acp_packet_byte_21[1]	acp_packet_byte_21[0]
0xFA 59	0x00	packetmemory_reg_fa59	rw	acp_packet_byte_22[7]	acp_packet_byte_22[6]	acp_packet_byte_22[5]	acp_packet_byte_22[4]	acp_packet_byte_22[3]	acp_packet_byte_22[2]	acp_packet_byte_22[1]	acp_packet_byte_22[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 5A	0x00	packetmemory_reg_fa5a	rw	acp_packet_byte_23[7]	acp_packet_byte_23[6]	acp_packet_byte_23[5]	acp_packet_byte_23[4]	acp_packet_byte_23[3]	acp_packet_byte_23[2]	acp_packet_byte_23[1]	acp_packet_byte_23[0]
0xFA 5B	0x00	packetmemory_reg_fa5b	rw	acp_packet_byte_24[7]	acp_packet_byte_24[6]	acp_packet_byte_24[5]	acp_packet_byte_24[4]	acp_packet_byte_24[3]	acp_packet_byte_24[2]	acp_packet_byte_24[1]	acp_packet_byte_24[0]
0xFA 5C	0x00	packetmemory_reg_fa5c	rw	acp_packet_byte_25[7]	acp_packet_byte_25[6]	acp_packet_byte_25[5]	acp_packet_byte_25[4]	acp_packet_byte_25[3]	acp_packet_byte_25[2]	acp_packet_byte_25[1]	acp_packet_byte_25[0]
0xFA 5D	0x00	packetmemory_reg_fa5d	rw	acp_packet_byte_26[7]	acp_packet_byte_26[6]	acp_packet_byte_26[5]	acp_packet_byte_26[4]	acp_packet_byte_26[3]	acp_packet_byte_26[2]	acp_packet_byte_26[1]	acp_packet_byte_26[0]
0xFA 5E	0x00	packetmemory_reg_fa5e	rw	acp_packet_byte_27[7]	acp_packet_byte_27[6]	acp_packet_byte_27[5]	acp_packet_byte_27[4]	acp_packet_byte_27[3]	acp_packet_byte_27[2]	acp_packet_byte_27[1]	acp_packet_byte_27[0]
0xFA 5F	0x00	packetmemory_reg_fa5f	rw	acp_update	-	-	-	-	-	-	-
0xFA 60	0x00	packetmemory_reg_fa60	rw	isrc1_header_byte_0[7]	isrc1_header_byte_0[6]	isrc1_header_byte_0[5]	isrc1_header_byte_0[4]	isrc1_header_byte_0[3]	isrc1_header_byte_0[2]	isrc1_header_byte_0[1]	isrc1_header_byte_0[0]
0xFA 61	0x00	packetmemory_reg_fa61	rw	isrc1_header_byte_1[7]	isrc1_header_byte_1[6]	isrc1_header_byte_1[5]	isrc1_header_byte_1[4]	isrc1_header_byte_1[3]	isrc1_header_byte_1[2]	isrc1_header_byte_1[1]	isrc1_header_byte_1[0]
0xFA 62	0x00	packetmemory_reg_fa62	rw	isrc1_header_byte_2[7]	isrc1_header_byte_2[6]	isrc1_header_byte_2[5]	isrc1_header_byte_2[4]	isrc1_header_byte_2[3]	isrc1_header_byte_2[2]	isrc1_header_byte_2[1]	isrc1_header_byte_2[0]
0xFA 63	0x00	packetmemory_reg_fa63	rw	isrc1_packet_byte_0[7]	isrc1_packet_byte_0[6]	isrc1_packet_byte_0[5]	isrc1_packet_byte_0[4]	isrc1_packet_byte_0[3]	isrc1_packet_byte_0[2]	isrc1_packet_byte_0[1]	isrc1_packet_byte_0[0]
0xFA 64	0x00	packetmemory_reg_fa64	rw	isrc1_packet_byte_1[7]	isrc1_packet_byte_1[6]	isrc1_packet_byte_1[5]	isrc1_packet_byte_1[4]	isrc1_packet_byte_1[3]	isrc1_packet_byte_1[2]	isrc1_packet_byte_1[1]	isrc1_packet_byte_1[0]
0xFA 65	0x00	packetmemory_reg_fa65	rw	isrc1_packet_byte_2[7]	isrc1_packet_byte_2[6]	isrc1_packet_byte_2[5]	isrc1_packet_byte_2[4]	isrc1_packet_byte_2[3]	isrc1_packet_byte_2[2]	isrc1_packet_byte_2[1]	isrc1_packet_byte_2[0]
0xFA 66	0x00	packetmemory_reg_fa66	rw	isrc1_packet_byte_3[7]	isrc1_packet_byte_3[6]	isrc1_packet_byte_3[5]	isrc1_packet_byte_3[4]	isrc1_packet_byte_3[3]	isrc1_packet_byte_3[2]	isrc1_packet_byte_3[1]	isrc1_packet_byte_3[0]
0xFA 67	0x00	packetmemory_reg_fa67	rw	isrc1_packet_byte_4[7]	isrc1_packet_byte_4[6]	isrc1_packet_byte_4[5]	isrc1_packet_byte_4[4]	isrc1_packet_byte_4[3]	isrc1_packet_byte_4[2]	isrc1_packet_byte_4[1]	isrc1_packet_byte_4[0]
0xFA 68	0x00	packetmemory_reg_fa68	rw	isrc1_packet_byte_5[7]	isrc1_packet_byte_5[6]	isrc1_packet_byte_5[5]	isrc1_packet_byte_5[4]	isrc1_packet_byte_5[3]	isrc1_packet_byte_5[2]	isrc1_packet_byte_5[1]	isrc1_packet_byte_5[0]
0xFA 69	0x00	packetmemory_reg_fa69	rw	isrc1_packet_byte_6[7]	isrc1_packet_byte_6[6]	isrc1_packet_byte_6[5]	isrc1_packet_byte_6[4]	isrc1_packet_byte_6[3]	isrc1_packet_byte_6[2]	isrc1_packet_byte_6[1]	isrc1_packet_byte_6[0]
0xFA 6A	0x00	packetmemory_reg_fa6a	rw	isrc1_packet_byte_7[7]	isrc1_packet_byte_7[6]	isrc1_packet_byte_7[5]	isrc1_packet_byte_7[4]	isrc1_packet_byte_7[3]	isrc1_packet_byte_7[2]	isrc1_packet_byte_7[1]	isrc1_packet_byte_7[0]
0xFA 6B	0x00	packetmemory_reg_fa6b	rw	isrc1_packet_byte_8[7]	isrc1_packet_byte_8[6]	isrc1_packet_byte_8[5]	isrc1_packet_byte_8[4]	isrc1_packet_byte_8[3]	isrc1_packet_byte_8[2]	isrc1_packet_byte_8[1]	isrc1_packet_byte_8[0]
0xFA 6C	0x00	packetmemory_reg_fa6c	rw	isrc1_packet_byte_9[7]	isrc1_packet_byte_9[6]	isrc1_packet_byte_9[5]	isrc1_packet_byte_9[4]	isrc1_packet_byte_9[3]	isrc1_packet_byte_9[2]	isrc1_packet_byte_9[1]	isrc1_packet_byte_9[0]
0xFA 6D	0x00	packetmemory_reg_fa6d	rw	isrc1_packet_byte_10[7]	isrc1_packet_byte_10[6]	isrc1_packet_byte_10[5]	isrc1_packet_byte_10[4]	isrc1_packet_byte_10[3]	isrc1_packet_byte_10[2]	isrc1_packet_byte_10[1]	isrc1_packet_byte_10[0]
0xFA 6E	0x00	packetmemory_reg_fa6e	rw	isrc1_packet_byte_11[7]	isrc1_packet_byte_11[6]	isrc1_packet_byte_11[5]	isrc1_packet_byte_11[4]	isrc1_packet_byte_11[3]	isrc1_packet_byte_11[2]	isrc1_packet_byte_11[1]	isrc1_packet_byte_11[0]
0xFA 6F	0x00	packetmemory_reg_fa6f	rw	isrc1_packet_byte_12[7]	isrc1_packet_byte_12[6]	isrc1_packet_byte_12[5]	isrc1_packet_byte_12[4]	isrc1_packet_byte_12[3]	isrc1_packet_byte_12[2]	isrc1_packet_byte_12[1]	isrc1_packet_byte_12[0]
0xFA 70	0x00	packetmemory_reg_fa70	r	isrc1_packet_byte_13[7]	isrc1_packet_byte_13[6]	isrc1_packet_byte_13[5]	isrc1_packet_byte_13[4]	isrc1_packet_byte_13[3]	isrc1_packet_byte_13[2]	isrc1_packet_byte_13[1]	isrc1_packet_byte_13[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 71	0x00	packetmemory_reg_fa71	r	isrc1_packet_byte_14[7]	isrc1_packet_byte_14[6]	isrc1_packet_byte_14[5]	isrc1_packet_byte_14[4]	isrc1_packet_byte_14[3]	isrc1_packet_byte_14[2]	isrc1_packet_byte_14[1]	isrc1_packet_byte_14[0]
0xFA 72	0x00	packetmemory_reg_fa72	r	isrc1_packet_byte_15[7]	isrc1_packet_byte_15[6]	isrc1_packet_byte_15[5]	isrc1_packet_byte_15[4]	isrc1_packet_byte_15[3]	isrc1_packet_byte_15[2]	isrc1_packet_byte_15[1]	isrc1_packet_byte_15[0]
0xFA 73	0x00	packetmemory_reg_fa73	r	isrc1_packet_byte_16[7]	isrc1_packet_byte_16[6]	isrc1_packet_byte_16[5]	isrc1_packet_byte_16[4]	isrc1_packet_byte_16[3]	isrc1_packet_byte_16[2]	isrc1_packet_byte_16[1]	isrc1_packet_byte_16[0]
0xFA 74	0x00	packetmemory_reg_fa74	r	isrc1_packet_byte_17[7]	isrc1_packet_byte_17[6]	isrc1_packet_byte_17[5]	isrc1_packet_byte_17[4]	isrc1_packet_byte_17[3]	isrc1_packet_byte_17[2]	isrc1_packet_byte_17[1]	isrc1_packet_byte_17[0]
0xFA 75	0x00	packetmemory_reg_fa75	rw	isrc1_packet_byte_18[7]	isrc1_packet_byte_18[6]	isrc1_packet_byte_18[5]	isrc1_packet_byte_18[4]	isrc1_packet_byte_18[3]	isrc1_packet_byte_18[2]	isrc1_packet_byte_18[1]	isrc1_packet_byte_18[0]
0xFA 76	0x00	packetmemory_reg_fa76	rw	isrc1_packet_byte_19[7]	isrc1_packet_byte_19[6]	isrc1_packet_byte_19[5]	isrc1_packet_byte_19[4]	isrc1_packet_byte_19[3]	isrc1_packet_byte_19[2]	isrc1_packet_byte_19[1]	isrc1_packet_byte_19[0]
0xFA 77	0x00	packetmemory_reg_fa77	rw	isrc1_packet_byte_20[7]	isrc1_packet_byte_20[6]	isrc1_packet_byte_20[5]	isrc1_packet_byte_20[4]	isrc1_packet_byte_20[3]	isrc1_packet_byte_20[2]	isrc1_packet_byte_20[1]	isrc1_packet_byte_20[0]
0xFA 78	0x00	packetmemory_reg_fa78	rw	isrc1_packet_byte_21[7]	isrc1_packet_byte_21[6]	isrc1_packet_byte_21[5]	isrc1_packet_byte_21[4]	isrc1_packet_byte_21[3]	isrc1_packet_byte_21[2]	isrc1_packet_byte_21[1]	isrc1_packet_byte_21[0]
0xFA 79	0x00	packetmemory_reg_fa79	rw	isrc1_packet_byte_22[7]	isrc1_packet_byte_22[6]	isrc1_packet_byte_22[5]	isrc1_packet_byte_22[4]	isrc1_packet_byte_22[3]	isrc1_packet_byte_22[2]	isrc1_packet_byte_22[1]	isrc1_packet_byte_22[0]
0xFA 7A	0x00	packetmemory_reg_fa7a	rw	isrc1_packet_byte_23[7]	isrc1_packet_byte_23[6]	isrc1_packet_byte_23[5]	isrc1_packet_byte_23[4]	isrc1_packet_byte_23[3]	isrc1_packet_byte_23[2]	isrc1_packet_byte_23[1]	isrc1_packet_byte_23[0]
0xFA 7B	0x00	packetmemory_reg_fa7b	rw	isrc1_packet_byte_24[7]	isrc1_packet_byte_24[6]	isrc1_packet_byte_24[5]	isrc1_packet_byte_24[4]	isrc1_packet_byte_24[3]	isrc1_packet_byte_24[2]	isrc1_packet_byte_24[1]	isrc1_packet_byte_24[0]
0xFA 7C	0x00	packetmemory_reg_fa7c	rw	isrc1_packet_byte_25[7]	isrc1_packet_byte_25[6]	isrc1_packet_byte_25[5]	isrc1_packet_byte_25[4]	isrc1_packet_byte_25[3]	isrc1_packet_byte_25[2]	isrc1_packet_byte_25[1]	isrc1_packet_byte_25[0]
0xFA 7D	0x00	packetmemory_reg_fa7d	rw	isrc1_packet_byte_26[7]	isrc1_packet_byte_26[6]	isrc1_packet_byte_26[5]	isrc1_packet_byte_26[4]	isrc1_packet_byte_26[3]	isrc1_packet_byte_26[2]	isrc1_packet_byte_26[1]	isrc1_packet_byte_26[0]
0xFA 7E	0x00	packetmemory_reg_fa7e	rw	isrc1_packet_byte_27[7]	isrc1_packet_byte_27[6]	isrc1_packet_byte_27[5]	isrc1_packet_byte_27[4]	isrc1_packet_byte_27[3]	isrc1_packet_byte_27[2]	isrc1_packet_byte_27[1]	isrc1_packet_byte_27[0]
0xFA 7F	0x00	packetmemory_reg_fa7f	rw	isrc1_update	-	-	-	-	-	-	-
0xFA 80	0x00	packetmemory_reg_fa80	rw	isrc2_header_byte_0[7]	isrc2_header_byte_0[6]	isrc2_header_byte_0[5]	isrc2_header_byte_0[4]	isrc2_header_byte_0[3]	isrc2_header_byte_0[2]	isrc2_header_byte_0[1]	isrc2_header_byte_0[0]
0xFA 81	0x00	packetmemory_reg_fa81	rw	isrc2_header_byte_1[7]	isrc2_header_byte_1[6]	isrc2_header_byte_1[5]	isrc2_header_byte_1[4]	isrc2_header_byte_1[3]	isrc2_header_byte_1[2]	isrc2_header_byte_1[1]	isrc2_header_byte_1[0]
0xFA 82	0x00	packetmemory_reg_fa82	rw	isrc2_header_byte_2[7]	isrc2_header_byte_2[6]	isrc2_header_byte_2[5]	isrc2_header_byte_2[4]	isrc2_header_byte_2[3]	isrc2_header_byte_2[2]	isrc2_header_byte_2[1]	isrc2_header_byte_2[0]
0xFA 83	0x00	packetmemory_reg_fa83	rw	isrc2_packet_byte_0[7]	isrc2_packet_byte_0[6]	isrc2_packet_byte_0[5]	isrc2_packet_byte_0[4]	isrc2_packet_byte_0[3]	isrc2_packet_byte_0[2]	isrc2_packet_byte_0[1]	isrc2_packet_byte_0[0]
0xFA 84	0x00	packetmemory_reg_fa84	rw	isrc2_packet_byte_1[7]	isrc2_packet_byte_1[6]	isrc2_packet_byte_1[5]	isrc2_packet_byte_1[4]	isrc2_packet_byte_1[3]	isrc2_packet_byte_1[2]	isrc2_packet_byte_1[1]	isrc2_packet_byte_1[0]
0xFA 85	0x00	packetmemory_reg_fa85	rw	isrc2_packet_byte_2[7]	isrc2_packet_byte_2[6]	isrc2_packet_byte_2[5]	isrc2_packet_byte_2[4]	isrc2_packet_byte_2[3]	isrc2_packet_byte_2[2]	isrc2_packet_byte_2[1]	isrc2_packet_byte_2[0]
0xFA 86	0x00	packetmemory_reg_fa86	rw	isrc2_packet_byte_3[7]	isrc2_packet_byte_3[6]	isrc2_packet_byte_3[5]	isrc2_packet_byte_3[4]	isrc2_packet_byte_3[3]	isrc2_packet_byte_3[2]	isrc2_packet_byte_3[1]	isrc2_packet_byte_3[0]
0xFA 87	0x00	packetmemory_reg_fa87	rw	isrc2_packet_byte_4[7]	isrc2_packet_byte_4[6]	isrc2_packet_byte_4[5]	isrc2_packet_byte_4[4]	isrc2_packet_byte_4[3]	isrc2_packet_byte_4[2]	isrc2_packet_byte_4[1]	isrc2_packet_byte_4[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 88	0x00	packetmemory_reg_fa88	rw	isrc2_packet_byte_5[7]	isrc2_packet_byte_5[6]	isrc2_packet_byte_5[5]	isrc2_packet_byte_5[4]	isrc2_packet_byte_5[3]	isrc2_packet_byte_5[2]	isrc2_packet_byte_5[1]	isrc2_packet_byte_5[0]
0xFA 89	0x00	packetmemory_reg_fa89	rw	isrc2_packet_byte_6[7]	isrc2_packet_byte_6[6]	isrc2_packet_byte_6[5]	isrc2_packet_byte_6[4]	isrc2_packet_byte_6[3]	isrc2_packet_byte_6[2]	isrc2_packet_byte_6[1]	isrc2_packet_byte_6[0]
0xFA 8A	0x00	packetmemory_reg_fa8a	rw	isrc2_packet_byte_7[7]	isrc2_packet_byte_7[6]	isrc2_packet_byte_7[5]	isrc2_packet_byte_7[4]	isrc2_packet_byte_7[3]	isrc2_packet_byte_7[2]	isrc2_packet_byte_7[1]	isrc2_packet_byte_7[0]
0xFA 8B	0x00	packetmemory_reg_fa8b	rw	isrc2_packet_byte_8[7]	isrc2_packet_byte_8[6]	isrc2_packet_byte_8[5]	isrc2_packet_byte_8[4]	isrc2_packet_byte_8[3]	isrc2_packet_byte_8[2]	isrc2_packet_byte_8[1]	isrc2_packet_byte_8[0]
0xFA 8C	0x00	packetmemory_reg_fa8c	rw	isrc2_packet_byte_9[7]	isrc2_packet_byte_9[6]	isrc2_packet_byte_9[5]	isrc2_packet_byte_9[4]	isrc2_packet_byte_9[3]	isrc2_packet_byte_9[2]	isrc2_packet_byte_9[1]	isrc2_packet_byte_9[0]
0xFA 8D	0x00	packetmemory_reg_fa8d	rw	isrc2_packet_byte_10[7]	isrc2_packet_byte_10[6]	isrc2_packet_byte_10[5]	isrc2_packet_byte_10[4]	isrc2_packet_byte_10[3]	isrc2_packet_byte_10[2]	isrc2_packet_byte_10[1]	isrc2_packet_byte_10[0]
0xFA 8E	0x00	packetmemory_reg_fa8e	rw	isrc2_packet_byte_11[7]	isrc2_packet_byte_11[6]	isrc2_packet_byte_11[5]	isrc2_packet_byte_11[4]	isrc2_packet_byte_11[3]	isrc2_packet_byte_11[2]	isrc2_packet_byte_11[1]	isrc2_packet_byte_11[0]
0xFA 8F	0x00	packetmemory_reg_fa8f	rw	isrc2_packet_byte_12[7]	isrc2_packet_byte_12[6]	isrc2_packet_byte_12[5]	isrc2_packet_byte_12[4]	isrc2_packet_byte_12[3]	isrc2_packet_byte_12[2]	isrc2_packet_byte_12[1]	isrc2_packet_byte_12[0]
0xFA 90	0x00	packetmemory_reg_fa90	rw	isrc2_packet_byte_13[7]	isrc2_packet_byte_13[6]	isrc2_packet_byte_13[5]	isrc2_packet_byte_13[4]	isrc2_packet_byte_13[3]	isrc2_packet_byte_13[2]	isrc2_packet_byte_13[1]	isrc2_packet_byte_13[0]
0xFA 91	0x00	packetmemory_reg_fa91	rw	isrc2_packet_byte_14[7]	isrc2_packet_byte_14[6]	isrc2_packet_byte_14[5]	isrc2_packet_byte_14[4]	isrc2_packet_byte_14[3]	isrc2_packet_byte_14[2]	isrc2_packet_byte_14[1]	isrc2_packet_byte_14[0]
0xFA 92	0x00	packetmemory_reg_fa92	rw	isrc2_packet_byte_15[7]	isrc2_packet_byte_15[6]	isrc2_packet_byte_15[5]	isrc2_packet_byte_15[4]	isrc2_packet_byte_15[3]	isrc2_packet_byte_15[2]	isrc2_packet_byte_15[1]	isrc2_packet_byte_15[0]
0xFA 93	0x00	packetmemory_reg_fa93	rw	isrc2_packet_byte_16[7]	isrc2_packet_byte_16[6]	isrc2_packet_byte_16[5]	isrc2_packet_byte_16[4]	isrc2_packet_byte_16[3]	isrc2_packet_byte_16[2]	isrc2_packet_byte_16[1]	isrc2_packet_byte_16[0]
0xFA 94	0x00	packetmemory_reg_fa94	rw	isrc2_packet_byte_17[7]	isrc2_packet_byte_17[6]	isrc2_packet_byte_17[5]	isrc2_packet_byte_17[4]	isrc2_packet_byte_17[3]	isrc2_packet_byte_17[2]	isrc2_packet_byte_17[1]	isrc2_packet_byte_17[0]
0xFA 95	0x00	packetmemory_reg_fa95	rw	isrc2_packet_byte_18[7]	isrc2_packet_byte_18[6]	isrc2_packet_byte_18[5]	isrc2_packet_byte_18[4]	isrc2_packet_byte_18[3]	isrc2_packet_byte_18[2]	isrc2_packet_byte_18[1]	isrc2_packet_byte_18[0]
0xFA 96	0x00	packetmemory_reg_fa96	rw	isrc2_packet_byte_19[7]	isrc2_packet_byte_19[6]	isrc2_packet_byte_19[5]	isrc2_packet_byte_19[4]	isrc2_packet_byte_19[3]	isrc2_packet_byte_19[2]	isrc2_packet_byte_19[1]	isrc2_packet_byte_19[0]
0xFA 97	0x00	packetmemory_reg_fa97	rw	isrc2_packet_byte_20[7]	isrc2_packet_byte_20[6]	isrc2_packet_byte_20[5]	isrc2_packet_byte_20[4]	isrc2_packet_byte_20[3]	isrc2_packet_byte_20[2]	isrc2_packet_byte_20[1]	isrc2_packet_byte_20[0]
0xFA 98	0x00	packetmemory_reg_fa98	rw	isrc2_packet_byte_21[7]	isrc2_packet_byte_21[6]	isrc2_packet_byte_21[5]	isrc2_packet_byte_21[4]	isrc2_packet_byte_21[3]	isrc2_packet_byte_21[2]	isrc2_packet_byte_21[1]	isrc2_packet_byte_21[0]
0xFA 99	0x00	packetmemory_reg_fa99	rw	isrc2_packet_byte_22[7]	isrc2_packet_byte_22[6]	isrc2_packet_byte_22[5]	isrc2_packet_byte_22[4]	isrc2_packet_byte_22[3]	isrc2_packet_byte_22[2]	isrc2_packet_byte_22[1]	isrc2_packet_byte_22[0]
0xFA 9A	0x00	packetmemory_reg_fa9a	rw	isrc2_packet_byte_23[7]	isrc2_packet_byte_23[6]	isrc2_packet_byte_23[5]	isrc2_packet_byte_23[4]	isrc2_packet_byte_23[3]	isrc2_packet_byte_23[2]	isrc2_packet_byte_23[1]	isrc2_packet_byte_23[0]
0xFA 9B	0x00	packetmemory_reg_fa9b	rw	isrc2_packet_byte_24[7]	isrc2_packet_byte_24[6]	isrc2_packet_byte_24[5]	isrc2_packet_byte_24[4]	isrc2_packet_byte_24[3]	isrc2_packet_byte_24[2]	isrc2_packet_byte_24[1]	isrc2_packet_byte_24[0]
0xFA 9C	0x00	packetmemory_reg_fa9c	rw	isrc2_packet_byte_25[7]	isrc2_packet_byte_25[6]	isrc2_packet_byte_25[5]	isrc2_packet_byte_25[4]	isrc2_packet_byte_25[3]	isrc2_packet_byte_25[2]	isrc2_packet_byte_25[1]	isrc2_packet_byte_25[0]
0xFA 9D	0x00	packetmemory_reg_fa9d	rw	isrc2_packet_byte_26[7]	isrc2_packet_byte_26[6]	isrc2_packet_byte_26[5]	isrc2_packet_byte_26[4]	isrc2_packet_byte_26[3]	isrc2_packet_byte_26[2]	isrc2_packet_byte_26[1]	isrc2_packet_byte_26[0]
0xFA 9E	0x00	packetmemory_reg_fa9e	rw	isrc2_packet_byte_27[7]	isrc2_packet_byte_27[6]	isrc2_packet_byte_27[5]	isrc2_packet_byte_27[4]	isrc2_packet_byte_27[3]	isrc2_packet_byte_27[2]	isrc2_packet_byte_27[1]	isrc2_packet_byte_27[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFA 9F	0x00	packetmemory_reg_fa9f	rw	isrc2_update	-	-	-	-	-	-	-
0xFA A0	0x00	packetmemory_reg_faa0	rw	gm_header_byte_0[7]	gm_header_byte_0[6]	gm_header_byte_0[5]	gm_header_byte_0[4]	gm_header_byte_0[3]	gm_header_byte_0[2]	gm_header_byte_0[1]	gm_header_byte_0[0]
0xFA A1	0x00	packetmemory_reg_faa1	rw	gm_header_byte_1[7]	gm_header_byte_1[6]	gm_header_byte_1[5]	gm_header_byte_1[4]	gm_header_byte_1[3]	gm_header_byte_1[2]	gm_header_byte_1[1]	gm_header_byte_1[0]
0xFA A2	0x00	packetmemory_reg_faa2	rw	gm_header_byte_2[7]	gm_header_byte_2[6]	gm_header_byte_2[5]	gm_header_byte_2[4]	gm_header_byte_2[3]	gm_header_byte_2[2]	gm_header_byte_2[1]	gm_header_byte_2[0]
0xFA A3	0x00	packetmemory_reg_faa3	rw	gm_packet_byte_0[7]	gm_packet_byte_0[6]	gm_packet_byte_0[5]	gm_packet_byte_0[4]	gm_packet_byte_0[3]	gm_packet_byte_0[2]	gm_packet_byte_0[1]	gm_packet_byte_0[0]
0xFA A4	0x00	packetmemory_reg_faa4	rw	gm_packet_byte_1[7]	gm_packet_byte_1[6]	gm_packet_byte_1[5]	gm_packet_byte_1[4]	gm_packet_byte_1[3]	gm_packet_byte_1[2]	gm_packet_byte_1[1]	gm_packet_byte_1[0]
0xFA A5	0x00	packetmemory_reg_faa5	rw	gm_packet_byte_2[7]	gm_packet_byte_2[6]	gm_packet_byte_2[5]	gm_packet_byte_2[4]	gm_packet_byte_2[3]	gm_packet_byte_2[2]	gm_packet_byte_2[1]	gm_packet_byte_2[0]
0xFA A6	0x00	packetmemory_reg_faa6	rw	gm_packet_byte_3[7]	gm_packet_byte_3[6]	gm_packet_byte_3[5]	gm_packet_byte_3[4]	gm_packet_byte_3[3]	gm_packet_byte_3[2]	gm_packet_byte_3[1]	gm_packet_byte_3[0]
0xFA A7	0x00	packetmemory_reg_faa7	rw	gm_packet_byte_4[7]	gm_packet_byte_4[6]	gm_packet_byte_4[5]	gm_packet_byte_4[4]	gm_packet_byte_4[3]	gm_packet_byte_4[2]	gm_packet_byte_4[1]	gm_packet_byte_4[0]
0xFA A8	0x00	packetmemory_reg_faa8	rw	gm_packet_byte_5[7]	gm_packet_byte_5[6]	gm_packet_byte_5[5]	gm_packet_byte_5[4]	gm_packet_byte_5[3]	gm_packet_byte_5[2]	gm_packet_byte_5[1]	gm_packet_byte_5[0]
0xFA A9	0x00	packetmemory_reg_faa9	rw	gm_packet_byte_6[7]	gm_packet_byte_6[6]	gm_packet_byte_6[5]	gm_packet_byte_6[4]	gm_packet_byte_6[3]	gm_packet_byte_6[2]	gm_packet_byte_6[1]	gm_packet_byte_6[0]
0xFA AA	0x00	packetmemory_reg_faaa	rw	gm_packet_byte_7[7]	gm_packet_byte_7[6]	gm_packet_byte_7[5]	gm_packet_byte_7[4]	gm_packet_byte_7[3]	gm_packet_byte_7[2]	gm_packet_byte_7[1]	gm_packet_byte_7[0]
0xFA AB	0x00	packetmemory_reg_faab	rw	gm_packet_byte_8[7]	gm_packet_byte_8[6]	gm_packet_byte_8[5]	gm_packet_byte_8[4]	gm_packet_byte_8[3]	gm_packet_byte_8[2]	gm_packet_byte_8[1]	gm_packet_byte_8[0]
0xFA AC	0x00	packetmemory_reg_faac	rw	gm_packet_byte_9[7]	gm_packet_byte_9[6]	gm_packet_byte_9[5]	gm_packet_byte_9[4]	gm_packet_byte_9[3]	gm_packet_byte_9[2]	gm_packet_byte_9[1]	gm_packet_byte_9[0]
0xFA AD	0x00	packetmemory_reg_faad	rw	gm_packet_byte_10[7]	gm_packet_byte_10[6]	gm_packet_byte_10[5]	gm_packet_byte_10[4]	gm_packet_byte_10[3]	gm_packet_byte_10[2]	gm_packet_byte_10[1]	gm_packet_byte_10[0]
0xFA AE	0x00	packetmemory_reg_faae	rw	gm_packet_byte_11[7]	gm_packet_byte_11[6]	gm_packet_byte_11[5]	gm_packet_byte_11[4]	gm_packet_byte_11[3]	gm_packet_byte_11[2]	gm_packet_byte_11[1]	gm_packet_byte_11[0]
0xFA AF	0x00	packetmemory_reg_faaf	rw	gm_packet_byte_12[7]	gm_packet_byte_12[6]	gm_packet_byte_12[5]	gm_packet_byte_12[4]	gm_packet_byte_12[3]	gm_packet_byte_12[2]	gm_packet_byte_12[1]	gm_packet_byte_12[0]
0xFA B0	0x00	packetmemory_reg_fab0	rw	gm_packet_byte_13[7]	gm_packet_byte_13[6]	gm_packet_byte_13[5]	gm_packet_byte_13[4]	gm_packet_byte_13[3]	gm_packet_byte_13[2]	gm_packet_byte_13[1]	gm_packet_byte_13[0]
0xFA B1	0x00	packetmemory_reg_fab1	rw	gm_packet_byte_14[7]	gm_packet_byte_14[6]	gm_packet_byte_14[5]	gm_packet_byte_14[4]	gm_packet_byte_14[3]	gm_packet_byte_14[2]	gm_packet_byte_14[1]	gm_packet_byte_14[0]
0xFA B2	0x00	packetmemory_reg_fab2	rw	gm_packet_byte_15[7]	gm_packet_byte_15[6]	gm_packet_byte_15[5]	gm_packet_byte_15[4]	gm_packet_byte_15[3]	gm_packet_byte_15[2]	gm_packet_byte_15[1]	gm_packet_byte_15[0]
0xFA B3	0x00	packetmemory_reg_fab3	rw	gm_packet_byte_16[7]	gm_packet_byte_16[6]	gm_packet_byte_16[5]	gm_packet_byte_16[4]	gm_packet_byte_16[3]	gm_packet_byte_16[2]	gm_packet_byte_16[1]	gm_packet_byte_16[0]
0xFA B4	0x00	packetmemory_reg_fab4	rw	gm_packet_byte_17[7]	gm_packet_byte_17[6]	gm_packet_byte_17[5]	gm_packet_byte_17[4]	gm_packet_byte_17[3]	gm_packet_byte_17[2]	gm_packet_byte_17[1]	gm_packet_byte_17[0]
0xFA B5	0x00	packetmemory_reg_fab5	rw	gm_packet_byte_18[7]	gm_packet_byte_18[6]	gm_packet_byte_18[5]	gm_packet_byte_18[4]	gm_packet_byte_18[3]	gm_packet_byte_18[2]	gm_packet_byte_18[1]	gm_packet_byte_18[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFAB6	0x00	packetmemory_reg_fab6	rw	gm_packet_byte_19[7]	gm_packet_byte_19[6]	gm_packet_byte_19[5]	gm_packet_byte_19[4]	gm_packet_byte_19[3]	gm_packet_byte_19[2]	gm_packet_byte_19[1]	gm_packet_byte_19[0]
0xFAB7	0x00	packetmemory_reg_fab7	rw	gm_packet_byte_20[7]	gm_packet_byte_20[6]	gm_packet_byte_20[5]	gm_packet_byte_20[4]	gm_packet_byte_20[3]	gm_packet_byte_20[2]	gm_packet_byte_20[1]	gm_packet_byte_20[0]
0xFAB8	0x00	packetmemory_reg_fab8	rw	gm_packet_byte_21[7]	gm_packet_byte_21[6]	gm_packet_byte_21[5]	gm_packet_byte_21[4]	gm_packet_byte_21[3]	gm_packet_byte_21[2]	gm_packet_byte_21[1]	gm_packet_byte_21[0]
0xFAB9	0x00	packetmemory_reg_fab9	rw	gm_packet_byte_22[7]	gm_packet_byte_22[6]	gm_packet_byte_22[5]	gm_packet_byte_22[4]	gm_packet_byte_22[3]	gm_packet_byte_22[2]	gm_packet_byte_22[1]	gm_packet_byte_22[0]
0xFABA	0x00	packetmemory_reg_faba	rw	gm_packet_byte_23[7]	gm_packet_byte_23[6]	gm_packet_byte_23[5]	gm_packet_byte_23[4]	gm_packet_byte_23[3]	gm_packet_byte_23[2]	gm_packet_byte_23[1]	gm_packet_byte_23[0]
0xFABB	0x00	packetmemory_reg_fabb	rw	gm_packet_byte_24[7]	gm_packet_byte_24[6]	gm_packet_byte_24[5]	gm_packet_byte_24[4]	gm_packet_byte_24[3]	gm_packet_byte_24[2]	gm_packet_byte_24[1]	gm_packet_byte_24[0]
0xFABC	0x00	packetmemory_reg_fabc	rw	gm_packet_byte_25[7]	gm_packet_byte_25[6]	gm_packet_byte_25[5]	gm_packet_byte_25[4]	gm_packet_byte_25[3]	gm_packet_byte_25[2]	gm_packet_byte_25[1]	gm_packet_byte_25[0]
0xFABD	0x00	packetmemory_reg_fabd	rw	gm_packet_byte_26[7]	gm_packet_byte_26[6]	gm_packet_byte_26[5]	gm_packet_byte_26[4]	gm_packet_byte_26[3]	gm_packet_byte_26[2]	gm_packet_byte_26[1]	gm_packet_byte_26[0]
0xFABE	0x00	packetmemory_reg_fabe	rw	gm_packet_byte_27[7]	gm_packet_byte_27[6]	gm_packet_byte_27[5]	gm_packet_byte_27[4]	gm_packet_byte_27[3]	gm_packet_byte_27[2]	gm_packet_byte_27[1]	gm_packet_byte_27[0]
0xFABF	0x00	packetmemory_reg_fabf	rw	gm_update	-	-	-	-	-	-	-
0xFAC0	0x00	packetmemory_reg_fac0	rw	spare_packet_1_header_byte_0[7]	spare_packet_1_header_byte_0[6]	spare_packet_1_header_byte_0[5]	spare_packet_1_header_byte_0[4]	spare_packet_1_header_byte_0[3]	spare_packet_1_header_byte_0[2]	spare_packet_1_header_byte_0[1]	spare_packet_1_header_byte_0[0]
0xFAC1	0x00	packetmemory_reg_fac1	rw	spare_packet_1_header_byte_1[7]	spare_packet_1_header_byte_1[6]	spare_packet_1_header_byte_1[5]	spare_packet_1_header_byte_1[4]	spare_packet_1_header_byte_1[3]	spare_packet_1_header_byte_1[2]	spare_packet_1_header_byte_1[1]	spare_packet_1_header_byte_1[0]
0xFAC2	0x00	packetmemory_reg_fac2	rw	spare_packet_1_header_byte_2[7]	spare_packet_1_header_byte_2[6]	spare_packet_1_header_byte_2[5]	spare_packet_1_header_byte_2[4]	spare_packet_1_header_byte_2[3]	spare_packet_1_header_byte_2[2]	spare_packet_1_header_byte_2[1]	spare_packet_1_header_byte_2[0]
0xFAC3	0x00	packetmemory_reg_fac3	rw	spare_packet_1_packet_byte_0[7]	spare_packet_1_packet_byte_0[6]	spare_packet_1_packet_byte_0[5]	spare_packet_1_packet_byte_0[4]	spare_packet_1_packet_byte_0[3]	spare_packet_1_packet_byte_0[2]	spare_packet_1_packet_byte_0[1]	spare_packet_1_packet_byte_0[0]
0xFAC4	0x00	packetmemory_reg_fac4	rw	spare_packet_1_packet_byte_1[7]	spare_packet_1_packet_byte_1[6]	spare_packet_1_packet_byte_1[5]	spare_packet_1_packet_byte_1[4]	spare_packet_1_packet_byte_1[3]	spare_packet_1_packet_byte_1[2]	spare_packet_1_packet_byte_1[1]	spare_packet_1_packet_byte_1[0]
0xFAC5	0x00	packetmemory_reg_fac5	rw	spare_packet_1_packet_byte_2[7]	spare_packet_1_packet_byte_2[6]	spare_packet_1_packet_byte_2[5]	spare_packet_1_packet_byte_2[4]	spare_packet_1_packet_byte_2[3]	spare_packet_1_packet_byte_2[2]	spare_packet_1_packet_byte_2[1]	spare_packet_1_packet_byte_2[0]
0xFAC6	0x00	packetmemory_reg_fac6	rw	spare_packet_1_packet_byte_3[7]	spare_packet_1_packet_byte_3[6]	spare_packet_1_packet_byte_3[5]	spare_packet_1_packet_byte_3[4]	spare_packet_1_packet_byte_3[3]	spare_packet_1_packet_byte_3[2]	spare_packet_1_packet_byte_3[1]	spare_packet_1_packet_byte_3[0]
0xFAC7	0x00	packetmemory_reg_fac7	rw	spare_packet_1_packet_byte_4[7]	spare_packet_1_packet_byte_4[6]	spare_packet_1_packet_byte_4[5]	spare_packet_1_packet_byte_4[4]	spare_packet_1_packet_byte_4[3]	spare_packet_1_packet_byte_4[2]	spare_packet_1_packet_byte_4[1]	spare_packet_1_packet_byte_4[0]
0xFAC8	0x00	packetmemory_reg_fac8	rw	spare_packet_1_packet_byte_5[7]	spare_packet_1_packet_byte_5[6]	spare_packet_1_packet_byte_5[5]	spare_packet_1_packet_byte_5[4]	spare_packet_1_packet_byte_5[3]	spare_packet_1_packet_byte_5[2]	spare_packet_1_packet_byte_5[1]	spare_packet_1_packet_byte_5[0]
0xFAC9	0x00	packetmemory_reg_fac9	rw	spare_packet_1_packet_byte_6[7]	spare_packet_1_packet_byte_6[6]	spare_packet_1_packet_byte_6[5]	spare_packet_1_packet_byte_6[4]	spare_packet_1_packet_byte_6[3]	spare_packet_1_packet_byte_6[2]	spare_packet_1_packet_byte_6[1]	spare_packet_1_packet_byte_6[0]
0xFACA	0x00	packetmemory_reg_faca	rw	spare_packet_1_packet_byte_7[7]	spare_packet_1_packet_byte_7[6]	spare_packet_1_packet_byte_7[5]	spare_packet_1_packet_byte_7[4]	spare_packet_1_packet_byte_7[3]	spare_packet_1_packet_byte_7[2]	spare_packet_1_packet_byte_7[1]	spare_packet_1_packet_byte_7[0]
0xFACB	0x00	packetmemory_reg_facb	rw	spare_packet_1_packet_byte_8[7]	spare_packet_1_packet_byte_8[6]	spare_packet_1_packet_byte_8[5]	spare_packet_1_packet_byte_8[4]	spare_packet_1_packet_byte_8[3]	spare_packet_1_packet_byte_8[2]	spare_packet_1_packet_byte_8[1]	spare_packet_1_packet_byte_8[0]
0xFACC	0x00	packetmemory_reg_facc	rw	spare_packet_1_packet_byte_9[7]	spare_packet_1_packet_byte_9[6]	spare_packet_1_packet_byte_9[5]	spare_packet_1_packet_byte_9[4]	spare_packet_1_packet_byte_9[3]	spare_packet_1_packet_byte_9[2]	spare_packet_1_packet_byte_9[1]	spare_packet_1_packet_byte_9[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFACD	0x00	packetmemory_reg_facd	rw	spare_packet_1_packet_byte_10[7]	spare_packet_1_packet_byte_10[6]	spare_packet_1_packet_byte_10[5]	spare_packet_1_packet_byte_10[4]	spare_packet_1_packet_byte_10[3]	spare_packet_1_packet_byte_10[2]	spare_packet_1_packet_byte_10[1]	spare_packet_1_packet_byte_10[0]
0xFACE	0x00	packetmemory_reg_face	rw	spare_packet_1_packet_byte_11[7]	spare_packet_1_packet_byte_11[6]	spare_packet_1_packet_byte_11[5]	spare_packet_1_packet_byte_11[4]	spare_packet_1_packet_byte_11[3]	spare_packet_1_packet_byte_11[2]	spare_packet_1_packet_byte_11[1]	spare_packet_1_packet_byte_11[0]
0xFACF	0x00	packetmemory_reg_facf	rw	spare_packet_1_packet_byte_12[7]	spare_packet_1_packet_byte_12[6]	spare_packet_1_packet_byte_12[5]	spare_packet_1_packet_byte_12[4]	spare_packet_1_packet_byte_12[3]	spare_packet_1_packet_byte_12[2]	spare_packet_1_packet_byte_12[1]	spare_packet_1_packet_byte_12[0]
0xFAD0	0x00	packetmemory_reg_fad0	rw	spare_packet_1_packet_byte_13[7]	spare_packet_1_packet_byte_13[6]	spare_packet_1_packet_byte_13[5]	spare_packet_1_packet_byte_13[4]	spare_packet_1_packet_byte_13[3]	spare_packet_1_packet_byte_13[2]	spare_packet_1_packet_byte_13[1]	spare_packet_1_packet_byte_13[0]
0xFAD1	0x00	packetmemory_reg_fad1	rw	spare_packet_1_packet_byte_14[7]	spare_packet_1_packet_byte_14[6]	spare_packet_1_packet_byte_14[5]	spare_packet_1_packet_byte_14[4]	spare_packet_1_packet_byte_14[3]	spare_packet_1_packet_byte_14[2]	spare_packet_1_packet_byte_14[1]	spare_packet_1_packet_byte_14[0]
0xFAD2	0x00	packetmemory_reg_fad2	rw	spare_packet_1_packet_byte_15[7]	spare_packet_1_packet_byte_15[6]	spare_packet_1_packet_byte_15[5]	spare_packet_1_packet_byte_15[4]	spare_packet_1_packet_byte_15[3]	spare_packet_1_packet_byte_15[2]	spare_packet_1_packet_byte_15[1]	spare_packet_1_packet_byte_15[0]
0xFAD3	0x00	packetmemory_reg_fad3	rw	spare_packet_1_packet_byte_16[7]	spare_packet_1_packet_byte_16[6]	spare_packet_1_packet_byte_16[5]	spare_packet_1_packet_byte_16[4]	spare_packet_1_packet_byte_16[3]	spare_packet_1_packet_byte_16[2]	spare_packet_1_packet_byte_16[1]	spare_packet_1_packet_byte_16[0]
0xFAD4	0x00	packetmemory_reg_fad4	rw	spare_packet_1_packet_byte_17[7]	spare_packet_1_packet_byte_17[6]	spare_packet_1_packet_byte_17[5]	spare_packet_1_packet_byte_17[4]	spare_packet_1_packet_byte_17[3]	spare_packet_1_packet_byte_17[2]	spare_packet_1_packet_byte_17[1]	spare_packet_1_packet_byte_17[0]
0xFAD5	0x00	packetmemory_reg_fad5	rw	spare_packet_1_packet_byte_18[7]	spare_packet_1_packet_byte_18[6]	spare_packet_1_packet_byte_18[5]	spare_packet_1_packet_byte_18[4]	spare_packet_1_packet_byte_18[3]	spare_packet_1_packet_byte_18[2]	spare_packet_1_packet_byte_18[1]	spare_packet_1_packet_byte_18[0]
0xFAD6	0x00	packetmemory_reg_fad6	rw	spare_packet_1_packet_byte_19[7]	spare_packet_1_packet_byte_19[6]	spare_packet_1_packet_byte_19[5]	spare_packet_1_packet_byte_19[4]	spare_packet_1_packet_byte_19[3]	spare_packet_1_packet_byte_19[2]	spare_packet_1_packet_byte_19[1]	spare_packet_1_packet_byte_19[0]
0xFAD7	0x00	packetmemory_reg_fad7	rw	spare_packet_1_packet_byte_20[7]	spare_packet_1_packet_byte_20[6]	spare_packet_1_packet_byte_20[5]	spare_packet_1_packet_byte_20[4]	spare_packet_1_packet_byte_20[3]	spare_packet_1_packet_byte_20[2]	spare_packet_1_packet_byte_20[1]	spare_packet_1_packet_byte_20[0]
0xFAD8	0x00	packetmemory_reg_fad8	rw	spare_packet_1_packet_byte_21[7]	spare_packet_1_packet_byte_21[6]	spare_packet_1_packet_byte_21[5]	spare_packet_1_packet_byte_21[4]	spare_packet_1_packet_byte_21[3]	spare_packet_1_packet_byte_21[2]	spare_packet_1_packet_byte_21[1]	spare_packet_1_packet_byte_21[0]
0xFAD9	0x00	packetmemory_reg_fad9	rw	spare_packet_1_packet_byte_22[7]	spare_packet_1_packet_byte_22[6]	spare_packet_1_packet_byte_22[5]	spare_packet_1_packet_byte_22[4]	spare_packet_1_packet_byte_22[3]	spare_packet_1_packet_byte_22[2]	spare_packet_1_packet_byte_22[1]	spare_packet_1_packet_byte_22[0]
0xFADA	0x00	packetmemory_reg_fada	rw	spare_packet_1_packet_byte_23[7]	spare_packet_1_packet_byte_23[6]	spare_packet_1_packet_byte_23[5]	spare_packet_1_packet_byte_23[4]	spare_packet_1_packet_byte_23[3]	spare_packet_1_packet_byte_23[2]	spare_packet_1_packet_byte_23[1]	spare_packet_1_packet_byte_23[0]
0xFADB	0x00	packetmemory_reg_fadb	rw	spare_packet_1_packet_byte_24[7]	spare_packet_1_packet_byte_24[6]	spare_packet_1_packet_byte_24[5]	spare_packet_1_packet_byte_24[4]	spare_packet_1_packet_byte_24[3]	spare_packet_1_packet_byte_24[2]	spare_packet_1_packet_byte_24[1]	spare_packet_1_packet_byte_24[0]
0xFADC	0x00	packetmemory_reg_fadc	rw	spare_packet_1_packet_byte_25[7]	spare_packet_1_packet_byte_25[6]	spare_packet_1_packet_byte_25[5]	spare_packet_1_packet_byte_25[4]	spare_packet_1_packet_byte_25[3]	spare_packet_1_packet_byte_25[2]	spare_packet_1_packet_byte_25[1]	spare_packet_1_packet_byte_25[0]
0xFADD	0x00	packetmemory_reg_fadd	rw	spare_packet_1_packet_byte_26[7]	spare_packet_1_packet_byte_26[6]	spare_packet_1_packet_byte_26[5]	spare_packet_1_packet_byte_26[4]	spare_packet_1_packet_byte_26[3]	spare_packet_1_packet_byte_26[2]	spare_packet_1_packet_byte_26[1]	spare_packet_1_packet_byte_26[0]
0xFADE	0x00	packetmemory_reg_fade	rw	spare_packet_1_packet_byte_27[7]	spare_packet_1_packet_byte_27[6]	spare_packet_1_packet_byte_27[5]	spare_packet_1_packet_byte_27[4]	spare_packet_1_packet_byte_27[3]	spare_packet_1_packet_byte_27[2]	spare_packet_1_packet_byte_27[1]	spare_packet_1_packet_byte_27[0]
0xFADF	0x00	packetmemory_reg_fadf	rw	spare1_update	-	-	-	-	-	-	-
0xFAE0	0x00	packetmemory_reg_fae0	rw	spare_packet_2_header_byte_0[7]	spare_packet_2_header_byte_0[6]	spare_packet_2_header_byte_0[5]	spare_packet_2_header_byte_0[4]	spare_packet_2_header_byte_0[3]	spare_packet_2_header_byte_0[2]	spare_packet_2_header_byte_0[1]	spare_packet_2_header_byte_0[0]
0xFAE1	0x00	packetmemory_reg_fae1	rw	spare_packet_2_header_byte_1[7]	spare_packet_2_header_byte_1[6]	spare_packet_2_header_byte_1[5]	spare_packet_2_header_byte_1[4]	spare_packet_2_header_byte_1[3]	spare_packet_2_header_byte_1[2]	spare_packet_2_header_byte_1[1]	spare_packet_2_header_byte_1[0]
0xFAE2	0x00	packetmemory_reg_fae2	rw	spare_packet_2_header_byte_2[7]	spare_packet_2_header_byte_2[6]	spare_packet_2_header_byte_2[5]	spare_packet_2_header_byte_2[4]	spare_packet_2_header_byte_2[3]	spare_packet_2_header_byte_2[2]	spare_packet_2_header_byte_2[1]	spare_packet_2_header_byte_2[0]
0xFAE3	0x00	packetmemory_reg_fae3	rw	spare_packet_2_packet_byte_0[7]	spare_packet_2_packet_byte_0[6]	spare_packet_2_packet_byte_0[5]	spare_packet_2_packet_byte_0[4]	spare_packet_2_packet_byte_0[3]	spare_packet_2_packet_byte_0[2]	spare_packet_2_packet_byte_0[1]	spare_packet_2_packet_byte_0[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFAE4	0x00	packetmemory_reg_fae4	rw	spare_packet_2_packet_byte_1[7]	spare_packet_2_packet_byte_1[6]	spare_packet_2_packet_byte_1[5]	spare_packet_2_packet_byte_1[4]	spare_packet_2_packet_byte_1[3]	spare_packet_2_packet_byte_1[2]	spare_packet_2_packet_byte_1[1]	spare_packet_2_packet_byte_1[0]
0xFAE5	0x00	packetmemory_reg_fae5	rw	spare_packet_2_packet_byte_2[7]	spare_packet_2_packet_byte_2[6]	spare_packet_2_packet_byte_2[5]	spare_packet_2_packet_byte_2[4]	spare_packet_2_packet_byte_2[3]	spare_packet_2_packet_byte_2[2]	spare_packet_2_packet_byte_2[1]	spare_packet_2_packet_byte_2[0]
0xFAE6	0x00	packetmemory_reg_fae6	rw	spare_packet_2_packet_byte_3[7]	spare_packet_2_packet_byte_3[6]	spare_packet_2_packet_byte_3[5]	spare_packet_2_packet_byte_3[4]	spare_packet_2_packet_byte_3[3]	spare_packet_2_packet_byte_3[2]	spare_packet_2_packet_byte_3[1]	spare_packet_2_packet_byte_3[0]
0xFAE7	0x00	packetmemory_reg_fae7	rw	spare_packet_2_packet_byte_4[7]	spare_packet_2_packet_byte_4[6]	spare_packet_2_packet_byte_4[5]	spare_packet_2_packet_byte_4[4]	spare_packet_2_packet_byte_4[3]	spare_packet_2_packet_byte_4[2]	spare_packet_2_packet_byte_4[1]	spare_packet_2_packet_byte_4[0]
0xFAE8	0x00	packetmemory_reg_fae8	rw	spare_packet_2_packet_byte_5[7]	spare_packet_2_packet_byte_5[6]	spare_packet_2_packet_byte_5[5]	spare_packet_2_packet_byte_5[4]	spare_packet_2_packet_byte_5[3]	spare_packet_2_packet_byte_5[2]	spare_packet_2_packet_byte_5[1]	spare_packet_2_packet_byte_5[0]
0xFAE9	0x00	packetmemory_reg_fae9	rw	spare_packet_2_packet_byte_6[7]	spare_packet_2_packet_byte_6[6]	spare_packet_2_packet_byte_6[5]	spare_packet_2_packet_byte_6[4]	spare_packet_2_packet_byte_6[3]	spare_packet_2_packet_byte_6[2]	spare_packet_2_packet_byte_6[1]	spare_packet_2_packet_byte_6[0]
0xFAEA	0x00	packetmemory_reg_faea	rw	spare_packet_2_packet_byte_7[7]	spare_packet_2_packet_byte_7[6]	spare_packet_2_packet_byte_7[5]	spare_packet_2_packet_byte_7[4]	spare_packet_2_packet_byte_7[3]	spare_packet_2_packet_byte_7[2]	spare_packet_2_packet_byte_7[1]	spare_packet_2_packet_byte_7[0]
0xFAEB	0x00	packetmemory_reg_faeb	rw	spare_packet_2_packet_byte_8[7]	spare_packet_2_packet_byte_8[6]	spare_packet_2_packet_byte_8[5]	spare_packet_2_packet_byte_8[4]	spare_packet_2_packet_byte_8[3]	spare_packet_2_packet_byte_8[2]	spare_packet_2_packet_byte_8[1]	spare_packet_2_packet_byte_8[0]
0xFAEC	0x00	packetmemory_reg_faec	rw	spare_packet_2_packet_byte_9[7]	spare_packet_2_packet_byte_9[6]	spare_packet_2_packet_byte_9[5]	spare_packet_2_packet_byte_9[4]	spare_packet_2_packet_byte_9[3]	spare_packet_2_packet_byte_9[2]	spare_packet_2_packet_byte_9[1]	spare_packet_2_packet_byte_9[0]
0xFAED	0x00	packetmemory_reg_faed	rw	spare_packet_2_packet_byte_10[7]	spare_packet_2_packet_byte_10[6]	spare_packet_2_packet_byte_10[5]	spare_packet_2_packet_byte_10[4]	spare_packet_2_packet_byte_10[3]	spare_packet_2_packet_byte_10[2]	spare_packet_2_packet_byte_10[1]	spare_packet_2_packet_byte_10[0]
0xFAEE	0x00	packetmemory_reg_faee	rw	spare_packet_2_packet_byte_11[7]	spare_packet_2_packet_byte_11[6]	spare_packet_2_packet_byte_11[5]	spare_packet_2_packet_byte_11[4]	spare_packet_2_packet_byte_11[3]	spare_packet_2_packet_byte_11[2]	spare_packet_2_packet_byte_11[1]	spare_packet_2_packet_byte_11[0]
0xFAEF	0x00	packetmemory_reg_faef	rw	spare_packet_2_packet_byte_12[7]	spare_packet_2_packet_byte_12[6]	spare_packet_2_packet_byte_12[5]	spare_packet_2_packet_byte_12[4]	spare_packet_2_packet_byte_12[3]	spare_packet_2_packet_byte_12[2]	spare_packet_2_packet_byte_12[1]	spare_packet_2_packet_byte_12[0]
0xFAF0	0x00	packetmemory_reg_faf0	rw	spare_packet_2_packet_byte_13[7]	spare_packet_2_packet_byte_13[6]	spare_packet_2_packet_byte_13[5]	spare_packet_2_packet_byte_13[4]	spare_packet_2_packet_byte_13[3]	spare_packet_2_packet_byte_13[2]	spare_packet_2_packet_byte_13[1]	spare_packet_2_packet_byte_13[0]
0xFAF1	0x00	packetmemory_reg_faf1	rw	spare_packet_2_packet_byte_14[7]	spare_packet_2_packet_byte_14[6]	spare_packet_2_packet_byte_14[5]	spare_packet_2_packet_byte_14[4]	spare_packet_2_packet_byte_14[3]	spare_packet_2_packet_byte_14[2]	spare_packet_2_packet_byte_14[1]	spare_packet_2_packet_byte_14[0]
0xFAF2	0x00	packetmemory_reg_faf2	rw	spare_packet_2_packet_byte_15[7]	spare_packet_2_packet_byte_15[6]	spare_packet_2_packet_byte_15[5]	spare_packet_2_packet_byte_15[4]	spare_packet_2_packet_byte_15[3]	spare_packet_2_packet_byte_15[2]	spare_packet_2_packet_byte_15[1]	spare_packet_2_packet_byte_15[0]
0xFAF3	0x00	packetmemory_reg_faf3	rw	spare_packet_2_packet_byte_16[7]	spare_packet_2_packet_byte_16[6]	spare_packet_2_packet_byte_16[5]	spare_packet_2_packet_byte_16[4]	spare_packet_2_packet_byte_16[3]	spare_packet_2_packet_byte_16[2]	spare_packet_2_packet_byte_16[1]	spare_packet_2_packet_byte_16[0]
0xFAF4	0x00	packetmemory_reg_faf4	rw	spare_packet_2_packet_byte_17[7]	spare_packet_2_packet_byte_17[6]	spare_packet_2_packet_byte_17[5]	spare_packet_2_packet_byte_17[4]	spare_packet_2_packet_byte_17[3]	spare_packet_2_packet_byte_17[2]	spare_packet_2_packet_byte_17[1]	spare_packet_2_packet_byte_17[0]
0xFAF5	0x00	packetmemory_reg_faf5	rw	spare_packet_2_packet_byte_18[7]	spare_packet_2_packet_byte_18[6]	spare_packet_2_packet_byte_18[5]	spare_packet_2_packet_byte_18[4]	spare_packet_2_packet_byte_18[3]	spare_packet_2_packet_byte_18[2]	spare_packet_2_packet_byte_18[1]	spare_packet_2_packet_byte_18[0]
0xFAF6	0x00	packetmemory_reg_faf6	rw	spare_packet_2_packet_byte_19[7]	spare_packet_2_packet_byte_19[6]	spare_packet_2_packet_byte_19[5]	spare_packet_2_packet_byte_19[4]	spare_packet_2_packet_byte_19[3]	spare_packet_2_packet_byte_19[2]	spare_packet_2_packet_byte_19[1]	spare_packet_2_packet_byte_19[0]
0xFAF7	0x00	packetmemory_reg_faf7	rw	spare_packet_2_packet_byte_20[7]	spare_packet_2_packet_byte_20[6]	spare_packet_2_packet_byte_20[5]	spare_packet_2_packet_byte_20[4]	spare_packet_2_packet_byte_20[3]	spare_packet_2_packet_byte_20[2]	spare_packet_2_packet_byte_20[1]	spare_packet_2_packet_byte_20[0]
0xFAF8	0x00	packetmemory_reg_faf8	rw	spare_packet_2_packet_byte_21[7]	spare_packet_2_packet_byte_21[6]	spare_packet_2_packet_byte_21[5]	spare_packet_2_packet_byte_21[4]	spare_packet_2_packet_byte_21[3]	spare_packet_2_packet_byte_21[2]	spare_packet_2_packet_byte_21[1]	spare_packet_2_packet_byte_21[0]
0xFAF9	0x00	packetmemory_reg_faf9	rw	spare_packet_2_packet_byte_22[7]	spare_packet_2_packet_byte_22[6]	spare_packet_2_packet_byte_22[5]	spare_packet_2_packet_byte_22[4]	spare_packet_2_packet_byte_22[3]	spare_packet_2_packet_byte_22[2]	spare_packet_2_packet_byte_22[1]	spare_packet_2_packet_byte_22[0]
0xFAFA	0x00	packetmemory_reg_fafa	rw	spare_packet_2_packet_byte_23[7]	spare_packet_2_packet_byte_23[6]	spare_packet_2_packet_byte_23[5]	spare_packet_2_packet_byte_23[4]	spare_packet_2_packet_byte_23[3]	spare_packet_2_packet_byte_23[2]	spare_packet_2_packet_byte_23[1]	spare_packet_2_packet_byte_23[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFAFB	0x00	packetmemory_reg_fafb	rw	spare_packet_2_packet_byte_24[7]	spare_packet_2_packet_byte_24[6]	spare_packet_2_packet_byte_24[5]	spare_packet_2_packet_byte_24[4]	spare_packet_2_packet_byte_24[3]	spare_packet_2_packet_byte_24[2]	spare_packet_2_packet_byte_24[1]	spare_packet_2_packet_byte_24[0]
0xFAFC	0x00	packetmemory_reg_fafc	rw	spare_packet_2_packet_byte_25[7]	spare_packet_2_packet_byte_25[6]	spare_packet_2_packet_byte_25[5]	spare_packet_2_packet_byte_25[4]	spare_packet_2_packet_byte_25[3]	spare_packet_2_packet_byte_25[2]	spare_packet_2_packet_byte_25[1]	spare_packet_2_packet_byte_25[0]
0xFAFD	0x00	packetmemory_reg_fafd	rw	spare_packet_2_packet_byte_26[7]	spare_packet_2_packet_byte_26[6]	spare_packet_2_packet_byte_26[5]	spare_packet_2_packet_byte_26[4]	spare_packet_2_packet_byte_26[3]	spare_packet_2_packet_byte_26[2]	spare_packet_2_packet_byte_26[1]	spare_packet_2_packet_byte_26[0]
0xFAFE	0x00	packetmemory_reg_fafe	rw	spare_packet_2_packet_byte_27[7]	spare_packet_2_packet_byte_27[6]	spare_packet_2_packet_byte_27[5]	spare_packet_2_packet_byte_27[4]	spare_packet_2_packet_byte_27[3]	spare_packet_2_packet_byte_27[2]	spare_packet_2_packet_byte_27[1]	spare_packet_2_packet_byte_27[0]
0xFAFF	0x00	packetmemory_reg_faff	rw	spare2_update	-	-	-	-	-	-	-

1.13 TX2 EDID MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF600	0x00	edidmemory_reg_f600	r	edid_0[7]	edid_0[6]	edid_0[5]	edid_0[4]	edid_0[3]	edid_0[2]	edid_0[1]	edid_0[0]
0xF601	0x00	edidmemory_reg_f601	r	edid_1[7]	edid_1[6]	edid_1[5]	edid_1[4]	edid_1[3]	edid_1[2]	edid_1[1]	edid_1[0]
0xF602	0x00	edidmemory_reg_f602	r	edid_2[7]	edid_2[6]	edid_2[5]	edid_2[4]	edid_2[3]	edid_2[2]	edid_2[1]	edid_2[0]
0xF603	0x00	edidmemory_reg_f603	r	edid_3[7]	edid_3[6]	edid_3[5]	edid_3[4]	edid_3[3]	edid_3[2]	edid_3[1]	edid_3[0]
0xF604	0x00	edidmemory_reg_f604	r	edid_4[7]	edid_4[6]	edid_4[5]	edid_4[4]	edid_4[3]	edid_4[2]	edid_4[1]	edid_4[0]
0xF605	0x00	edidmemory_reg_f605	r	edid_5[7]	edid_5[6]	edid_5[5]	edid_5[4]	edid_5[3]	edid_5[2]	edid_5[1]	edid_5[0]
0xF606	0x00	edidmemory_reg_f606	r	edid_6[7]	edid_6[6]	edid_6[5]	edid_6[4]	edid_6[3]	edid_6[2]	edid_6[1]	edid_6[0]
0xF607	0x00	edidmemory_reg_f607	r	edid_7[7]	edid_7[6]	edid_7[5]	edid_7[4]	edid_7[3]	edid_7[2]	edid_7[1]	edid_7[0]
0xF608	0x00	edidmemory_reg_f608	r	edid_8[7]	edid_8[6]	edid_8[5]	edid_8[4]	edid_8[3]	edid_8[2]	edid_8[1]	edid_8[0]
0xF609	0x00	edidmemory_reg_f609	r	edid_9[7]	edid_9[6]	edid_9[5]	edid_9[4]	edid_9[3]	edid_9[2]	edid_9[1]	edid_9[0]
0xF60A	0x00	edidmemory_reg_f60a	r	edid_10[7]	edid_10[6]	edid_10[5]	edid_10[4]	edid_10[3]	edid_10[2]	edid_10[1]	edid_10[0]
0xF60B	0x00	edidmemory_reg_f60b	r	edid_11[7]	edid_11[6]	edid_11[5]	edid_11[4]	edid_11[3]	edid_11[2]	edid_11[1]	edid_11[0]
0xF60C	0x00	edidmemory_reg_f60c	r	edid_12[7]	edid_12[6]	edid_12[5]	edid_12[4]	edid_12[3]	edid_12[2]	edid_12[1]	edid_12[0]
0xF60D	0x00	edidmemory_reg_f60d	r	edid_13[7]	edid_13[6]	edid_13[5]	edid_13[4]	edid_13[3]	edid_13[2]	edid_13[1]	edid_13[0]
0xF60E	0x00	edidmemory_reg_f60e	r	edid_14[7]	edid_14[6]	edid_14[5]	edid_14[4]	edid_14[3]	edid_14[2]	edid_14[1]	edid_14[0]
0xF60F	0x00	edidmemory_reg_f60f	r	edid_15[7]	edid_15[6]	edid_15[5]	edid_15[4]	edid_15[3]	edid_15[2]	edid_15[1]	edid_15[0]
0xF610	0x00	edidmemory_reg_f610	r	edid_16[7]	edid_16[6]	edid_16[5]	edid_16[4]	edid_16[3]	edid_16[2]	edid_16[1]	edid_16[0]
0xF611	0x00	edidmemory_reg_f611	r	edid_17[7]	edid_17[6]	edid_17[5]	edid_17[4]	edid_17[3]	edid_17[2]	edid_17[1]	edid_17[0]
0xF612	0x00	edidmemory_reg_f612	r	edid_18[7]	edid_18[6]	edid_18[5]	edid_18[4]	edid_18[3]	edid_18[2]	edid_18[1]	edid_18[0]
0xF613	0x00	edidmemory_reg_f613	r	edid_19[7]	edid_19[6]	edid_19[5]	edid_19[4]	edid_19[3]	edid_19[2]	edid_19[1]	edid_19[0]
0xF614	0x00	edidmemory_reg_f614	r	edid_20[7]	edid_20[6]	edid_20[5]	edid_20[4]	edid_20[3]	edid_20[2]	edid_20[1]	edid_20[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF615	0x00	edidmemory_reg_f615	r	edid_21[7]	edid_21[6]	edid_21[5]	edid_21[4]	edid_21[3]	edid_21[2]	edid_21[1]	edid_21[0]
0xF616	0x00	edidmemory_reg_f616	r	edid_22[7]	edid_22[6]	edid_22[5]	edid_22[4]	edid_22[3]	edid_22[2]	edid_22[1]	edid_22[0]
0xF617	0x00	edidmemory_reg_f617	r	edid_23[7]	edid_23[6]	edid_23[5]	edid_23[4]	edid_23[3]	edid_23[2]	edid_23[1]	edid_23[0]
0xF618	0x00	edidmemory_reg_f618	r	edid_24[7]	edid_24[6]	edid_24[5]	edid_24[4]	edid_24[3]	edid_24[2]	edid_24[1]	edid_24[0]
0xF619	0x00	edidmemory_reg_f619	r	edid_25[7]	edid_25[6]	edid_25[5]	edid_25[4]	edid_25[3]	edid_25[2]	edid_25[1]	edid_25[0]
0xF61A	0x00	edidmemory_reg_f61a	r	edid_26[7]	edid_26[6]	edid_26[5]	edid_26[4]	edid_26[3]	edid_26[2]	edid_26[1]	edid_26[0]
0xF61B	0x00	edidmemory_reg_f61b	r	edid_27[7]	edid_27[6]	edid_27[5]	edid_27[4]	edid_27[3]	edid_27[2]	edid_27[1]	edid_27[0]
0xF61C	0x00	edidmemory_reg_f61c	r	edid_28[7]	edid_28[6]	edid_28[5]	edid_28[4]	edid_28[3]	edid_28[2]	edid_28[1]	edid_28[0]
0xF61D	0x00	edidmemory_reg_f61d	r	edid_29[7]	edid_29[6]	edid_29[5]	edid_29[4]	edid_29[3]	edid_29[2]	edid_29[1]	edid_29[0]
0xF61E	0x00	edidmemory_reg_f61e	r	edid_30[7]	edid_30[6]	edid_30[5]	edid_30[4]	edid_30[3]	edid_30[2]	edid_30[1]	edid_30[0]
0xF61F	0x00	edidmemory_reg_f61f	r	edid_31[7]	edid_31[6]	edid_31[5]	edid_31[4]	edid_31[3]	edid_31[2]	edid_31[1]	edid_31[0]
0xF620	0x00	edidmemory_reg_f620	r	edid_32[7]	edid_32[6]	edid_32[5]	edid_32[4]	edid_32[3]	edid_32[2]	edid_32[1]	edid_32[0]
0xF621	0x00	edidmemory_reg_f621	r	edid_33[7]	edid_33[6]	edid_33[5]	edid_33[4]	edid_33[3]	edid_33[2]	edid_33[1]	edid_33[0]
0xF622	0x00	edidmemory_reg_f622	r	edid_34[7]	edid_34[6]	edid_34[5]	edid_34[4]	edid_34[3]	edid_34[2]	edid_34[1]	edid_34[0]
0xF623	0x00	edidmemory_reg_f623	r	edid_35[7]	edid_35[6]	edid_35[5]	edid_35[4]	edid_35[3]	edid_35[2]	edid_35[1]	edid_35[0]
0xF624	0x00	edidmemory_reg_f624	r	edid_36[7]	edid_36[6]	edid_36[5]	edid_36[4]	edid_36[3]	edid_36[2]	edid_36[1]	edid_36[0]
0xF625	0x00	edidmemory_reg_f625	r	edid_37[7]	edid_37[6]	edid_37[5]	edid_37[4]	edid_37[3]	edid_37[2]	edid_37[1]	edid_37[0]
0xF626	0x00	edidmemory_reg_f626	r	edid_38[7]	edid_38[6]	edid_38[5]	edid_38[4]	edid_38[3]	edid_38[2]	edid_38[1]	edid_38[0]
0xF627	0x00	edidmemory_reg_f627	r	edid_39[7]	edid_39[6]	edid_39[5]	edid_39[4]	edid_39[3]	edid_39[2]	edid_39[1]	edid_39[0]
0xF628	0x00	edidmemory_reg_f628	r	edid_40[7]	edid_40[6]	edid_40[5]	edid_40[4]	edid_40[3]	edid_40[2]	edid_40[1]	edid_40[0]
0xF629	0x00	edidmemory_reg_f629	r	edid_41[7]	edid_41[6]	edid_41[5]	edid_41[4]	edid_41[3]	edid_41[2]	edid_41[1]	edid_41[0]
0xF62A	0x00	edidmemory_reg_f62a	r	edid_42[7]	edid_42[6]	edid_42[5]	edid_42[4]	edid_42[3]	edid_42[2]	edid_42[1]	edid_42[0]
0xF62B	0x00	edidmemory_reg_f62b	r	edid_43[7]	edid_43[6]	edid_43[5]	edid_43[4]	edid_43[3]	edid_43[2]	edid_43[1]	edid_43[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF62C	0x00	edidmemory_reg_f62c	r	edid_44[7]	edid_44[6]	edid_44[5]	edid_44[4]	edid_44[3]	edid_44[2]	edid_44[1]	edid_44[0]
0xF62D	0x00	edidmemory_reg_f62d	r	edid_45[7]	edid_45[6]	edid_45[5]	edid_45[4]	edid_45[3]	edid_45[2]	edid_45[1]	edid_45[0]
0xF62E	0x00	edidmemory_reg_f62e	r	edid_46[7]	edid_46[6]	edid_46[5]	edid_46[4]	edid_46[3]	edid_46[2]	edid_46[1]	edid_46[0]
0xF62F	0x00	edidmemory_reg_f62f	r	edid_47[7]	edid_47[6]	edid_47[5]	edid_47[4]	edid_47[3]	edid_47[2]	edid_47[1]	edid_47[0]
0xF630	0x00	edidmemory_reg_f630	r	edid_48[7]	edid_48[6]	edid_48[5]	edid_48[4]	edid_48[3]	edid_48[2]	edid_48[1]	edid_48[0]
0xF631	0x00	edidmemory_reg_f631	r	edid_49[7]	edid_49[6]	edid_49[5]	edid_49[4]	edid_49[3]	edid_49[2]	edid_49[1]	edid_49[0]
0xF632	0x00	edidmemory_reg_f632	r	edid_50[7]	edid_50[6]	edid_50[5]	edid_50[4]	edid_50[3]	edid_50[2]	edid_50[1]	edid_50[0]
0xF633	0x00	edidmemory_reg_f633	r	edid_51[7]	edid_51[6]	edid_51[5]	edid_51[4]	edid_51[3]	edid_51[2]	edid_51[1]	edid_51[0]
0xF634	0x00	edidmemory_reg_f634	r	edid_52[7]	edid_52[6]	edid_52[5]	edid_52[4]	edid_52[3]	edid_52[2]	edid_52[1]	edid_52[0]
0xF635	0x00	edidmemory_reg_f635	r	edid_53[7]	edid_53[6]	edid_53[5]	edid_53[4]	edid_53[3]	edid_53[2]	edid_53[1]	edid_53[0]
0xF636	0x00	edidmemory_reg_f636	r	edid_54[7]	edid_54[6]	edid_54[5]	edid_54[4]	edid_54[3]	edid_54[2]	edid_54[1]	edid_54[0]
0xF637	0x00	edidmemory_reg_f637	r	edid_55[7]	edid_55[6]	edid_55[5]	edid_55[4]	edid_55[3]	edid_55[2]	edid_55[1]	edid_55[0]
0xF638	0x00	edidmemory_reg_f638	r	edid_56[7]	edid_56[6]	edid_56[5]	edid_56[4]	edid_56[3]	edid_56[2]	edid_56[1]	edid_56[0]
0xF639	0x00	edidmemory_reg_f639	r	edid_57[7]	edid_57[6]	edid_57[5]	edid_57[4]	edid_57[3]	edid_57[2]	edid_57[1]	edid_57[0]
0xF63A	0x00	edidmemory_reg_f63a	r	edid_58[7]	edid_58[6]	edid_58[5]	edid_58[4]	edid_58[3]	edid_58[2]	edid_58[1]	edid_58[0]
0xF63B	0x00	edidmemory_reg_f63b	r	edid_59[7]	edid_59[6]	edid_59[5]	edid_59[4]	edid_59[3]	edid_59[2]	edid_59[1]	edid_59[0]
0xF63C	0x00	edidmemory_reg_f63c	r	edid_60[7]	edid_60[6]	edid_60[5]	edid_60[4]	edid_60[3]	edid_60[2]	edid_60[1]	edid_60[0]
0xF63D	0x00	edidmemory_reg_f63d	r	edid_61[7]	edid_61[6]	edid_61[5]	edid_61[4]	edid_61[3]	edid_61[2]	edid_61[1]	edid_61[0]
0xF63E	0x00	edidmemory_reg_f63e	r	edid_62[7]	edid_62[6]	edid_62[5]	edid_62[4]	edid_62[3]	edid_62[2]	edid_62[1]	edid_62[0]
0xF63F	0x00	edidmemory_reg_f63f	r	edid_63[7]	edid_63[6]	edid_63[5]	edid_63[4]	edid_63[3]	edid_63[2]	edid_63[1]	edid_63[0]
0xF640	0x00	edidmemory_reg_f640	r	edid_64[7]	edid_64[6]	edid_64[5]	edid_64[4]	edid_64[3]	edid_64[2]	edid_64[1]	edid_64[0]
0xF641	0x00	edidmemory_reg_f641	r	edid_65[7]	edid_65[6]	edid_65[5]	edid_65[4]	edid_65[3]	edid_65[2]	edid_65[1]	edid_65[0]
0xF642	0x00	edidmemory_reg_f642	r	edid_66[7]	edid_66[6]	edid_66[5]	edid_66[4]	edid_66[3]	edid_66[2]	edid_66[1]	edid_66[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF643	0x00	edidmemory_reg_f643	r	edid_67[7]	edid_67[6]	edid_67[5]	edid_67[4]	edid_67[3]	edid_67[2]	edid_67[1]	edid_67[0]
0xF644	0x00	edidmemory_reg_f644	r	edid_68[7]	edid_68[6]	edid_68[5]	edid_68[4]	edid_68[3]	edid_68[2]	edid_68[1]	edid_68[0]
0xF645	0x00	edidmemory_reg_f645	r	edid_69[7]	edid_69[6]	edid_69[5]	edid_69[4]	edid_69[3]	edid_69[2]	edid_69[1]	edid_69[0]
0xF646	0x00	edidmemory_reg_f646	r	edid_70[7]	edid_70[6]	edid_70[5]	edid_70[4]	edid_70[3]	edid_70[2]	edid_70[1]	edid_70[0]
0xF647	0x00	edidmemory_reg_f647	r	edid_71[7]	edid_71[6]	edid_71[5]	edid_71[4]	edid_71[3]	edid_71[2]	edid_71[1]	edid_71[0]
0xF648	0x00	edidmemory_reg_f648	r	edid_72[7]	edid_72[6]	edid_72[5]	edid_72[4]	edid_72[3]	edid_72[2]	edid_72[1]	edid_72[0]
0xF649	0x00	edidmemory_reg_f649	r	edid_73[7]	edid_73[6]	edid_73[5]	edid_73[4]	edid_73[3]	edid_73[2]	edid_73[1]	edid_73[0]
0xF64A	0x00	edidmemory_reg_f64a	r	edid_74[7]	edid_74[6]	edid_74[5]	edid_74[4]	edid_74[3]	edid_74[2]	edid_74[1]	edid_74[0]
0xF64B	0x00	edidmemory_reg_f64b	r	edid_75[7]	edid_75[6]	edid_75[5]	edid_75[4]	edid_75[3]	edid_75[2]	edid_75[1]	edid_75[0]
0xF64C	0x00	edidmemory_reg_f64c	r	edid_76[7]	edid_76[6]	edid_76[5]	edid_76[4]	edid_76[3]	edid_76[2]	edid_76[1]	edid_76[0]
0xF64D	0x00	edidmemory_reg_f64d	r	edid_77[7]	edid_77[6]	edid_77[5]	edid_77[4]	edid_77[3]	edid_77[2]	edid_77[1]	edid_77[0]
0xF64E	0x00	edidmemory_reg_f64e	r	edid_78[7]	edid_78[6]	edid_78[5]	edid_78[4]	edid_78[3]	edid_78[2]	edid_78[1]	edid_78[0]
0xF64F	0x00	edidmemory_reg_f64f	r	edid_79[7]	edid_79[6]	edid_79[5]	edid_79[4]	edid_79[3]	edid_79[2]	edid_79[1]	edid_79[0]
0xF650	0x00	edidmemory_reg_f650	r	edid_80[7]	edid_80[6]	edid_80[5]	edid_80[4]	edid_80[3]	edid_80[2]	edid_80[1]	edid_80[0]
0xF651	0x00	edidmemory_reg_f651	r	edid_81[7]	edid_81[6]	edid_81[5]	edid_81[4]	edid_81[3]	edid_81[2]	edid_81[1]	edid_81[0]
0xF652	0x00	edidmemory_reg_f652	r	edid_82[7]	edid_82[6]	edid_82[5]	edid_82[4]	edid_82[3]	edid_82[2]	edid_82[1]	edid_82[0]
0xF653	0x00	edidmemory_reg_f653	r	edid_83[7]	edid_83[6]	edid_83[5]	edid_83[4]	edid_83[3]	edid_83[2]	edid_83[1]	edid_83[0]
0xF654	0x00	edidmemory_reg_f654	r	edid_84[7]	edid_84[6]	edid_84[5]	edid_84[4]	edid_84[3]	edid_84[2]	edid_84[1]	edid_84[0]
0xF655	0x00	edidmemory_reg_f655	r	edid_85[7]	edid_85[6]	edid_85[5]	edid_85[4]	edid_85[3]	edid_85[2]	edid_85[1]	edid_85[0]
0xF656	0x00	edidmemory_reg_f656	r	edid_86[7]	edid_86[6]	edid_86[5]	edid_86[4]	edid_86[3]	edid_86[2]	edid_86[1]	edid_86[0]
0xF657	0x00	edidmemory_reg_f657	r	edid_87[7]	edid_87[6]	edid_87[5]	edid_87[4]	edid_87[3]	edid_87[2]	edid_87[1]	edid_87[0]
0xF658	0x00	edidmemory_reg_f658	r	edid_88[7]	edid_88[6]	edid_88[5]	edid_88[4]	edid_88[3]	edid_88[2]	edid_88[1]	edid_88[0]
0xF659	0x00	edidmemory_reg_f659	r	edid_89[7]	edid_89[6]	edid_89[5]	edid_89[4]	edid_89[3]	edid_89[2]	edid_89[1]	edid_89[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF65A	0x00	edidmemory_reg_f65a	r	edid_90[7]	edid_90[6]	edid_90[5]	edid_90[4]	edid_90[3]	edid_90[2]	edid_90[1]	edid_90[0]
0xF65B	0x00	edidmemory_reg_f65b	r	edid_91[7]	edid_91[6]	edid_91[5]	edid_91[4]	edid_91[3]	edid_91[2]	edid_91[1]	edid_91[0]
0xF65C	0x00	edidmemory_reg_f65c	r	edid_92[7]	edid_92[6]	edid_92[5]	edid_92[4]	edid_92[3]	edid_92[2]	edid_92[1]	edid_92[0]
0xF65D	0x00	edidmemory_reg_f65d	r	edid_93[7]	edid_93[6]	edid_93[5]	edid_93[4]	edid_93[3]	edid_93[2]	edid_93[1]	edid_93[0]
0xF65E	0x00	edidmemory_reg_f65e	r	edid_94[7]	edid_94[6]	edid_94[5]	edid_94[4]	edid_94[3]	edid_94[2]	edid_94[1]	edid_94[0]
0xF65F	0x00	edidmemory_reg_f65f	r	edid_95[7]	edid_95[6]	edid_95[5]	edid_95[4]	edid_95[3]	edid_95[2]	edid_95[1]	edid_95[0]
0xF660	0x00	edidmemory_reg_f660	r	edid_96[7]	edid_96[6]	edid_96[5]	edid_96[4]	edid_96[3]	edid_96[2]	edid_96[1]	edid_96[0]
0xF661	0x00	edidmemory_reg_f661	r	edid_97[7]	edid_97[6]	edid_97[5]	edid_97[4]	edid_97[3]	edid_97[2]	edid_97[1]	edid_97[0]
0xF662	0x00	edidmemory_reg_f662	r	edid_98[7]	edid_98[6]	edid_98[5]	edid_98[4]	edid_98[3]	edid_98[2]	edid_98[1]	edid_98[0]
0xF663	0x00	edidmemory_reg_f663	r	edid_99[7]	edid_99[6]	edid_99[5]	edid_99[4]	edid_99[3]	edid_99[2]	edid_99[1]	edid_99[0]
0xF664	0x00	edidmemory_reg_f664	r	edid_100[7]	edid_100[6]	edid_100[5]	edid_100[4]	edid_100[3]	edid_100[2]	edid_100[1]	edid_100[0]
0xF665	0x00	edidmemory_reg_f665	r	edid_101[7]	edid_101[6]	edid_101[5]	edid_101[4]	edid_101[3]	edid_101[2]	edid_101[1]	edid_101[0]
0xF666	0x00	edidmemory_reg_f666	r	edid_102[7]	edid_102[6]	edid_102[5]	edid_102[4]	edid_102[3]	edid_102[2]	edid_102[1]	edid_102[0]
0xF667	0x00	edidmemory_reg_f667	r	edid_103[7]	edid_103[6]	edid_103[5]	edid_103[4]	edid_103[3]	edid_103[2]	edid_103[1]	edid_103[0]
0xF668	0x00	edidmemory_reg_f668	r	edid_104[7]	edid_104[6]	edid_104[5]	edid_104[4]	edid_104[3]	edid_104[2]	edid_104[1]	edid_104[0]
0xF669	0x00	edidmemory_reg_f669	r	edid_105[7]	edid_105[6]	edid_105[5]	edid_105[4]	edid_105[3]	edid_105[2]	edid_105[1]	edid_105[0]
0xF66A	0x00	edidmemory_reg_f66a	r	edid_106[7]	edid_106[6]	edid_106[5]	edid_106[4]	edid_106[3]	edid_106[2]	edid_106[1]	edid_106[0]
0xF66B	0x00	edidmemory_reg_f66b	r	edid_107[7]	edid_107[6]	edid_107[5]	edid_107[4]	edid_107[3]	edid_107[2]	edid_107[1]	edid_107[0]
0xF66C	0x00	edidmemory_reg_f66c	r	edid_108[7]	edid_108[6]	edid_108[5]	edid_108[4]	edid_108[3]	edid_108[2]	edid_108[1]	edid_108[0]
0xF66D	0x00	edidmemory_reg_f66d	r	edid_109[7]	edid_109[6]	edid_109[5]	edid_109[4]	edid_109[3]	edid_109[2]	edid_109[1]	edid_109[0]
0xF66E	0x00	edidmemory_reg_f66e	r	edid_110[7]	edid_110[6]	edid_110[5]	edid_110[4]	edid_110[3]	edid_110[2]	edid_110[1]	edid_110[0]
0xF66F	0x00	edidmemory_reg_f66f	r	edid_111[7]	edid_111[6]	edid_111[5]	edid_111[4]	edid_111[3]	edid_111[2]	edid_111[1]	edid_111[0]
0xF670	0x00	edidmemory_reg_f670	r	edid_112[7]	edid_112[6]	edid_112[5]	edid_112[4]	edid_112[3]	edid_112[2]	edid_112[1]	edid_112[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF671	0x00	edidmemory_reg_f671	r	edid_113[7]	edid_113[6]	edid_113[5]	edid_113[4]	edid_113[3]	edid_113[2]	edid_113[1]	edid_113[0]
0xF672	0x00	edidmemory_reg_f672	r	edid_114[7]	edid_114[6]	edid_114[5]	edid_114[4]	edid_114[3]	edid_114[2]	edid_114[1]	edid_114[0]
0xF673	0x00	edidmemory_reg_f673	r	edid_115[7]	edid_115[6]	edid_115[5]	edid_115[4]	edid_115[3]	edid_115[2]	edid_115[1]	edid_115[0]
0xF674	0x00	edidmemory_reg_f674	r	edid_116[7]	edid_116[6]	edid_116[5]	edid_116[4]	edid_116[3]	edid_116[2]	edid_116[1]	edid_116[0]
0xF675	0x00	edidmemory_reg_f675	r	edid_117[7]	edid_117[6]	edid_117[5]	edid_117[4]	edid_117[3]	edid_117[2]	edid_117[1]	edid_117[0]
0xF676	0x00	edidmemory_reg_f676	r	edid_118[7]	edid_118[6]	edid_118[5]	edid_118[4]	edid_118[3]	edid_118[2]	edid_118[1]	edid_118[0]
0xF677	0x00	edidmemory_reg_f677	r	edid_119[7]	edid_119[6]	edid_119[5]	edid_119[4]	edid_119[3]	edid_119[2]	edid_119[1]	edid_119[0]
0xF678	0x00	edidmemory_reg_f678	r	edid_120[7]	edid_120[6]	edid_120[5]	edid_120[4]	edid_120[3]	edid_120[2]	edid_120[1]	edid_120[0]
0xF679	0x00	edidmemory_reg_f679	r	edid_121[7]	edid_121[6]	edid_121[5]	edid_121[4]	edid_121[3]	edid_121[2]	edid_121[1]	edid_121[0]
0xF67A	0x00	edidmemory_reg_f67a	r	edid_122[7]	edid_122[6]	edid_122[5]	edid_122[4]	edid_122[3]	edid_122[2]	edid_122[1]	edid_122[0]
0xF67B	0x00	edidmemory_reg_f67b	r	edid_123[7]	edid_123[6]	edid_123[5]	edid_123[4]	edid_123[3]	edid_123[2]	edid_123[1]	edid_123[0]
0xF67C	0x00	edidmemory_reg_f67c	r	edid_124[7]	edid_124[6]	edid_124[5]	edid_124[4]	edid_124[3]	edid_124[2]	edid_124[1]	edid_124[0]
0xF67D	0x00	edidmemory_reg_f67d	r	edid_125[7]	edid_125[6]	edid_125[5]	edid_125[4]	edid_125[3]	edid_125[2]	edid_125[1]	edid_125[0]
0xF67E	0x00	edidmemory_reg_f67e	r	edid_126[7]	edid_126[6]	edid_126[5]	edid_126[4]	edid_126[3]	edid_126[2]	edid_126[1]	edid_126[0]
0xF67F	0x00	edidmemory_reg_f67f	r	edid_127[7]	edid_127[6]	edid_127[5]	edid_127[4]	edid_127[3]	edid_127[2]	edid_127[1]	edid_127[0]
0xF680	0x00	edidmemory_reg_f680	r	edid_128[7]	edid_128[6]	edid_128[5]	edid_128[4]	edid_128[3]	edid_128[2]	edid_128[1]	edid_128[0]
0xF681	0x00	edidmemory_reg_f681	r	edid_129[7]	edid_129[6]	edid_129[5]	edid_129[4]	edid_129[3]	edid_129[2]	edid_129[1]	edid_129[0]
0xF682	0x00	edidmemory_reg_f682	r	edid_130[7]	edid_130[6]	edid_130[5]	edid_130[4]	edid_130[3]	edid_130[2]	edid_130[1]	edid_130[0]
0xF683	0x00	edidmemory_reg_f683	r	edid_131[7]	edid_131[6]	edid_131[5]	edid_131[4]	edid_131[3]	edid_131[2]	edid_131[1]	edid_131[0]
0xF684	0x00	edidmemory_reg_f684	r	edid_132[7]	edid_132[6]	edid_132[5]	edid_132[4]	edid_132[3]	edid_132[2]	edid_132[1]	edid_132[0]
0xF685	0x00	edidmemory_reg_f685	r	edid_133[7]	edid_133[6]	edid_133[5]	edid_133[4]	edid_133[3]	edid_133[2]	edid_133[1]	edid_133[0]
0xF686	0x00	edidmemory_reg_f686	r	edid_134[7]	edid_134[6]	edid_134[5]	edid_134[4]	edid_134[3]	edid_134[2]	edid_134[1]	edid_134[0]
0xF687	0x00	edidmemory_reg_f687	r	edid_135[7]	edid_135[6]	edid_135[5]	edid_135[4]	edid_135[3]	edid_135[2]	edid_135[1]	edid_135[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF688	0x00	edidmemory_reg_f688	r	edid_136[7]	edid_136[6]	edid_136[5]	edid_136[4]	edid_136[3]	edid_136[2]	edid_136[1]	edid_136[0]
0xF689	0x00	edidmemory_reg_f689	r	edid_137[7]	edid_137[6]	edid_137[5]	edid_137[4]	edid_137[3]	edid_137[2]	edid_137[1]	edid_137[0]
0xF68A	0x00	edidmemory_reg_f68a	r	edid_138[7]	edid_138[6]	edid_138[5]	edid_138[4]	edid_138[3]	edid_138[2]	edid_138[1]	edid_138[0]
0xF68B	0x00	edidmemory_reg_f68b	r	edid_139[7]	edid_139[6]	edid_139[5]	edid_139[4]	edid_139[3]	edid_139[2]	edid_139[1]	edid_139[0]
0xF68C	0x00	edidmemory_reg_f68c	r	edid_140[7]	edid_140[6]	edid_140[5]	edid_140[4]	edid_140[3]	edid_140[2]	edid_140[1]	edid_140[0]
0xF68D	0x00	edidmemory_reg_f68d	r	edid_141[7]	edid_141[6]	edid_141[5]	edid_141[4]	edid_141[3]	edid_141[2]	edid_141[1]	edid_141[0]
0xF68E	0x00	edidmemory_reg_f68e	r	edid_142[7]	edid_142[6]	edid_142[5]	edid_142[4]	edid_142[3]	edid_142[2]	edid_142[1]	edid_142[0]
0xF68F	0x00	edidmemory_reg_f68f	r	edid_143[7]	edid_143[6]	edid_143[5]	edid_143[4]	edid_143[3]	edid_143[2]	edid_143[1]	edid_143[0]
0xF690	0x00	edidmemory_reg_f690	r	edid_144[7]	edid_144[6]	edid_144[5]	edid_144[4]	edid_144[3]	edid_144[2]	edid_144[1]	edid_144[0]
0xF691	0x00	edidmemory_reg_f691	r	edid_145[7]	edid_145[6]	edid_145[5]	edid_145[4]	edid_145[3]	edid_145[2]	edid_145[1]	edid_145[0]
0xF692	0x00	edidmemory_reg_f692	r	edid_146[7]	edid_146[6]	edid_146[5]	edid_146[4]	edid_146[3]	edid_146[2]	edid_146[1]	edid_146[0]
0xF693	0x00	edidmemory_reg_f693	r	edid_147[7]	edid_147[6]	edid_147[5]	edid_147[4]	edid_147[3]	edid_147[2]	edid_147[1]	edid_147[0]
0xF694	0x00	edidmemory_reg_f694	r	edid_148[7]	edid_148[6]	edid_148[5]	edid_148[4]	edid_148[3]	edid_148[2]	edid_148[1]	edid_148[0]
0xF695	0x00	edidmemory_reg_f695	r	edid_149[7]	edid_149[6]	edid_149[5]	edid_149[4]	edid_149[3]	edid_149[2]	edid_149[1]	edid_149[0]
0xF696	0x00	edidmemory_reg_f696	r	edid_150[7]	edid_150[6]	edid_150[5]	edid_150[4]	edid_150[3]	edid_150[2]	edid_150[1]	edid_150[0]
0xF697	0x00	edidmemory_reg_f697	r	edid_151[7]	edid_151[6]	edid_151[5]	edid_151[4]	edid_151[3]	edid_151[2]	edid_151[1]	edid_151[0]
0xF698	0x00	edidmemory_reg_f698	r	edid_152[7]	edid_152[6]	edid_152[5]	edid_152[4]	edid_152[3]	edid_152[2]	edid_152[1]	edid_152[0]
0xF699	0x00	edidmemory_reg_f699	r	edid_153[7]	edid_153[6]	edid_153[5]	edid_153[4]	edid_153[3]	edid_153[2]	edid_153[1]	edid_153[0]
0xF69A	0x00	edidmemory_reg_f69a	r	edid_154[7]	edid_154[6]	edid_154[5]	edid_154[4]	edid_154[3]	edid_154[2]	edid_154[1]	edid_154[0]
0xF69B	0x00	edidmemory_reg_f69b	r	edid_155[7]	edid_155[6]	edid_155[5]	edid_155[4]	edid_155[3]	edid_155[2]	edid_155[1]	edid_155[0]
0xF69C	0x00	edidmemory_reg_f69c	r	edid_156[7]	edid_156[6]	edid_156[5]	edid_156[4]	edid_156[3]	edid_156[2]	edid_156[1]	edid_156[0]
0xF69D	0x00	edidmemory_reg_f69d	r	edid_157[7]	edid_157[6]	edid_157[5]	edid_157[4]	edid_157[3]	edid_157[2]	edid_157[1]	edid_157[0]
0xF69E	0x00	edidmemory_reg_f69e	r	edid_158[7]	edid_158[6]	edid_158[5]	edid_158[4]	edid_158[3]	edid_158[2]	edid_158[1]	edid_158[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF69F	0x00	edidmemory_reg_f69f	r	edid_159[7]	edid_159[6]	edid_159[5]	edid_159[4]	edid_159[3]	edid_159[2]	edid_159[1]	edid_159[0]
0xF6A0	0x00	edidmemory_reg_f6a0	r	edid_160[7]	edid_160[6]	edid_160[5]	edid_160[4]	edid_160[3]	edid_160[2]	edid_160[1]	edid_160[0]
0xF6A1	0x00	edidmemory_reg_f6a1	r	edid_161[7]	edid_161[6]	edid_161[5]	edid_161[4]	edid_161[3]	edid_161[2]	edid_161[1]	edid_161[0]
0xF6A2	0x00	edidmemory_reg_f6a2	r	edid_162[7]	edid_162[6]	edid_162[5]	edid_162[4]	edid_162[3]	edid_162[2]	edid_162[1]	edid_162[0]
0xF6A3	0x00	edidmemory_reg_f6a3	r	edid_163[7]	edid_163[6]	edid_163[5]	edid_163[4]	edid_163[3]	edid_163[2]	edid_163[1]	edid_163[0]
0xF6A4	0x00	edidmemory_reg_f6a4	r	edid_164[7]	edid_164[6]	edid_164[5]	edid_164[4]	edid_164[3]	edid_164[2]	edid_164[1]	edid_164[0]
0xF6A5	0x00	edidmemory_reg_f6a5	r	edid_165[7]	edid_165[6]	edid_165[5]	edid_165[4]	edid_165[3]	edid_165[2]	edid_165[1]	edid_165[0]
0xF6A6	0x00	edidmemory_reg_f6a6	r	edid_166[7]	edid_166[6]	edid_166[5]	edid_166[4]	edid_166[3]	edid_166[2]	edid_166[1]	edid_166[0]
0xF6A7	0x00	edidmemory_reg_f6a7	r	edid_167[7]	edid_167[6]	edid_167[5]	edid_167[4]	edid_167[3]	edid_167[2]	edid_167[1]	edid_167[0]
0xF6A8	0x00	edidmemory_reg_f6a8	r	edid_168[7]	edid_168[6]	edid_168[5]	edid_168[4]	edid_168[3]	edid_168[2]	edid_168[1]	edid_168[0]
0xF6A9	0x00	edidmemory_reg_f6a9	r	edid_169[7]	edid_169[6]	edid_169[5]	edid_169[4]	edid_169[3]	edid_169[2]	edid_169[1]	edid_169[0]
0xF6AA	0x00	edidmemory_reg_f6aa	r	edid_170[7]	edid_170[6]	edid_170[5]	edid_170[4]	edid_170[3]	edid_170[2]	edid_170[1]	edid_170[0]
0xF6AB	0x00	edidmemory_reg_f6ab	r	edid_171[7]	edid_171[6]	edid_171[5]	edid_171[4]	edid_171[3]	edid_171[2]	edid_171[1]	edid_171[0]
0xF6AC	0x00	edidmemory_reg_f6ac	r	edid_172[7]	edid_172[6]	edid_172[5]	edid_172[4]	edid_172[3]	edid_172[2]	edid_172[1]	edid_172[0]
0xF6AD	0x00	edidmemory_reg_f6ad	r	edid_173[7]	edid_173[6]	edid_173[5]	edid_173[4]	edid_173[3]	edid_173[2]	edid_173[1]	edid_173[0]
0xF6AE	0x00	edidmemory_reg_f6ae	r	edid_174[7]	edid_174[6]	edid_174[5]	edid_174[4]	edid_174[3]	edid_174[2]	edid_174[1]	edid_174[0]
0xF6AF	0x00	edidmemory_reg_f6af	r	edid_175[7]	edid_175[6]	edid_175[5]	edid_175[4]	edid_175[3]	edid_175[2]	edid_175[1]	edid_175[0]
0xF6B0	0x00	edidmemory_reg_f6b0	r	edid_176[7]	edid_176[6]	edid_176[5]	edid_176[4]	edid_176[3]	edid_176[2]	edid_176[1]	edid_176[0]
0xF6B1	0x00	edidmemory_reg_f6b1	r	edid_177[7]	edid_177[6]	edid_177[5]	edid_177[4]	edid_177[3]	edid_177[2]	edid_177[1]	edid_177[0]
0xF6B2	0x00	edidmemory_reg_f6b2	r	edid_178[7]	edid_178[6]	edid_178[5]	edid_178[4]	edid_178[3]	edid_178[2]	edid_178[1]	edid_178[0]
0xF6B3	0x00	edidmemory_reg_f6b3	r	edid_179[7]	edid_179[6]	edid_179[5]	edid_179[4]	edid_179[3]	edid_179[2]	edid_179[1]	edid_179[0]
0xF6B4	0x00	edidmemory_reg_f6b4	r	edid_180[7]	edid_180[6]	edid_180[5]	edid_180[4]	edid_180[3]	edid_180[2]	edid_180[1]	edid_180[0]
0xF6B5	0x00	edidmemory_reg_f6b5	r	edid_181[7]	edid_181[6]	edid_181[5]	edid_181[4]	edid_181[3]	edid_181[2]	edid_181[1]	edid_181[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF6 B6	0x00	edidmemory_reg_ f6b6	r	edid_182[7]	edid_182[6]	edid_182[5]	edid_182[4]	edid_182[3]	edid_182[2]	edid_182[1]	edid_182[0]
0xF6 B7	0x00	edidmemory_reg_ f6b7	r	edid_183[7]	edid_183[6]	edid_183[5]	edid_183[4]	edid_183[3]	edid_183[2]	edid_183[1]	edid_183[0]
0xF6 B8	0x00	edidmemory_reg_ f6b8	r	edid_184[7]	edid_184[6]	edid_184[5]	edid_184[4]	edid_184[3]	edid_184[2]	edid_184[1]	edid_184[0]
0xF6 B9	0x00	edidmemory_reg_ f6b9	r	edid_185[7]	edid_185[6]	edid_185[5]	edid_185[4]	edid_185[3]	edid_185[2]	edid_185[1]	edid_185[0]
0xF6 BA	0x00	edidmemory_reg_ f6ba	r	edid_186[7]	edid_186[6]	edid_186[5]	edid_186[4]	edid_186[3]	edid_186[2]	edid_186[1]	edid_186[0]
0xF6 BB	0x00	edidmemory_reg_ f6bb	r	edid_187[7]	edid_187[6]	edid_187[5]	edid_187[4]	edid_187[3]	edid_187[2]	edid_187[1]	edid_187[0]
0xF6 BC	0x00	edidmemory_reg_ f6bc	r	edid_188[7]	edid_188[6]	edid_188[5]	edid_188[4]	edid_188[3]	edid_188[2]	edid_188[1]	edid_188[0]
0xF6 BD	0x00	edidmemory_reg_ f6bd	r	edid_189[7]	edid_189[6]	edid_189[5]	edid_189[4]	edid_189[3]	edid_189[2]	edid_189[1]	edid_189[0]
0xF6 BE	0x00	edidmemory_reg_ f6be	r	edid_190[7]	edid_190[6]	edid_190[5]	edid_190[4]	edid_190[3]	edid_190[2]	edid_190[1]	edid_190[0]
0xF6 BF	0x00	edidmemory_reg_ f6bf	r	edid_191[7]	edid_191[6]	edid_191[5]	edid_191[4]	edid_191[3]	edid_191[2]	edid_191[1]	edid_191[0]
0xF6 C0	0x00	edidmemory_reg_ f6c0	r	edid_192[7]	edid_192[6]	edid_192[5]	edid_192[4]	edid_192[3]	edid_192[2]	edid_192[1]	edid_192[0]
0xF6 C1	0x00	edidmemory_reg_ f6c1	r	edid_193[7]	edid_193[6]	edid_193[5]	edid_193[4]	edid_193[3]	edid_193[2]	edid_193[1]	edid_193[0]
0xF6 C2	0x00	edidmemory_reg_ f6c2	r	edid_194[7]	edid_194[6]	edid_194[5]	edid_194[4]	edid_194[3]	edid_194[2]	edid_194[1]	edid_194[0]
0xF6 C3	0x00	edidmemory_reg_ f6c3	r	edid_195[7]	edid_195[6]	edid_195[5]	edid_195[4]	edid_195[3]	edid_195[2]	edid_195[1]	edid_195[0]
0xF6 C4	0x00	edidmemory_reg_ f6c4	r	edid_196[7]	edid_196[6]	edid_196[5]	edid_196[4]	edid_196[3]	edid_196[2]	edid_196[1]	edid_196[0]
0xF6 C5	0x00	edidmemory_reg_ f6c5	r	edid_197[7]	edid_197[6]	edid_197[5]	edid_197[4]	edid_197[3]	edid_197[2]	edid_197[1]	edid_197[0]
0xF6 C6	0x00	edidmemory_reg_ f6c6	r	edid_198[7]	edid_198[6]	edid_198[5]	edid_198[4]	edid_198[3]	edid_198[2]	edid_198[1]	edid_198[0]
0xF6 C7	0x00	edidmemory_reg_ f6c7	r	edid_199[7]	edid_199[6]	edid_199[5]	edid_199[4]	edid_199[3]	edid_199[2]	edid_199[1]	edid_199[0]
0xF6 C8	0x00	edidmemory_reg_ f6c8	r	edid_200[7]	edid_200[6]	edid_200[5]	edid_200[4]	edid_200[3]	edid_200[2]	edid_200[1]	edid_200[0]
0xF6 C9	0x00	edidmemory_reg_ f6c9	r	edid_201[7]	edid_201[6]	edid_201[5]	edid_201[4]	edid_201[3]	edid_201[2]	edid_201[1]	edid_201[0]
0xF6 CA	0x00	edidmemory_reg_ f6ca	r	edid_202[7]	edid_202[6]	edid_202[5]	edid_202[4]	edid_202[3]	edid_202[2]	edid_202[1]	edid_202[0]
0xF6 CB	0x00	edidmemory_reg_ f6cb	r	edid_203[7]	edid_203[6]	edid_203[5]	edid_203[4]	edid_203[3]	edid_203[2]	edid_203[1]	edid_203[0]
0xF6 CC	0x00	edidmemory_reg_ f6cc	r	edid_204[7]	edid_204[6]	edid_204[5]	edid_204[4]	edid_204[3]	edid_204[2]	edid_204[1]	edid_204[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF6 CD	0x00	edidmemory_reg_ f6cd	r	edid_205[7]	edid_205[6]	edid_205[5]	edid_205[4]	edid_205[3]	edid_205[2]	edid_205[1]	edid_205[0]
0xF6 CE	0x00	edidmemory_reg_ f6ce	r	edid_206[7]	edid_206[6]	edid_206[5]	edid_206[4]	edid_206[3]	edid_206[2]	edid_206[1]	edid_206[0]
0xF6 CF	0x00	edidmemory_reg_ f6cf	r	edid_207[7]	edid_207[6]	edid_207[5]	edid_207[4]	edid_207[3]	edid_207[2]	edid_207[1]	edid_207[0]
0xF6 D0	0x00	edidmemory_reg_ f6d0	r	edid_208[7]	edid_208[6]	edid_208[5]	edid_208[4]	edid_208[3]	edid_208[2]	edid_208[1]	edid_208[0]
0xF6 D1	0x00	edidmemory_reg_ f6d1	r	edid_209[7]	edid_209[6]	edid_209[5]	edid_209[4]	edid_209[3]	edid_209[2]	edid_209[1]	edid_209[0]
0xF6 D2	0x00	edidmemory_reg_ f6d2	r	edid_210[7]	edid_210[6]	edid_210[5]	edid_210[4]	edid_210[3]	edid_210[2]	edid_210[1]	edid_210[0]
0xF6 D3	0x00	edidmemory_reg_ f6d3	r	edid_211[7]	edid_211[6]	edid_211[5]	edid_211[4]	edid_211[3]	edid_211[2]	edid_211[1]	edid_211[0]
0xF6 D4	0x00	edidmemory_reg_ f6d4	r	edid_212[7]	edid_212[6]	edid_212[5]	edid_212[4]	edid_212[3]	edid_212[2]	edid_212[1]	edid_212[0]
0xF6 D5	0x00	edidmemory_reg_ f6d5	r	edid_213[7]	edid_213[6]	edid_213[5]	edid_213[4]	edid_213[3]	edid_213[2]	edid_213[1]	edid_213[0]
0xF6 D6	0x00	edidmemory_reg_ f6d6	r	edid_214[7]	edid_214[6]	edid_214[5]	edid_214[4]	edid_214[3]	edid_214[2]	edid_214[1]	edid_214[0]
0xF6 D7	0x00	edidmemory_reg_ f6d7	r	edid_215[7]	edid_215[6]	edid_215[5]	edid_215[4]	edid_215[3]	edid_215[2]	edid_215[1]	edid_215[0]
0xF6 D8	0x00	edidmemory_reg_ f6d8	r	edid_216[7]	edid_216[6]	edid_216[5]	edid_216[4]	edid_216[3]	edid_216[2]	edid_216[1]	edid_216[0]
0xF6 D9	0x00	edidmemory_reg_ f6d9	r	edid_217[7]	edid_217[6]	edid_217[5]	edid_217[4]	edid_217[3]	edid_217[2]	edid_217[1]	edid_217[0]
0xF6 DA	0x00	edidmemory_reg_ f6da	r	edid_218[7]	edid_218[6]	edid_218[5]	edid_218[4]	edid_218[3]	edid_218[2]	edid_218[1]	edid_218[0]
0xF6 DB	0x00	edidmemory_reg_ f6db	r	edid_219[7]	edid_219[6]	edid_219[5]	edid_219[4]	edid_219[3]	edid_219[2]	edid_219[1]	edid_219[0]
0xF6 DC	0x00	edidmemory_reg_ f6dc	r	edid_220[7]	edid_220[6]	edid_220[5]	edid_220[4]	edid_220[3]	edid_220[2]	edid_220[1]	edid_220[0]
0xF6 DD	0x00	edidmemory_reg_ f6dd	r	edid_221[7]	edid_221[6]	edid_221[5]	edid_221[4]	edid_221[3]	edid_221[2]	edid_221[1]	edid_221[0]
0xF6 DE	0x00	edidmemory_reg_ f6de	r	edid_222[7]	edid_222[6]	edid_222[5]	edid_222[4]	edid_222[3]	edid_222[2]	edid_222[1]	edid_222[0]
0xF6 DF	0x00	edidmemory_reg_ f6df	r	edid_223[7]	edid_223[6]	edid_223[5]	edid_223[4]	edid_223[3]	edid_223[2]	edid_223[1]	edid_223[0]
0xF6 E0	0x00	edidmemory_reg_ f6e0	r	edid_224[7]	edid_224[6]	edid_224[5]	edid_224[4]	edid_224[3]	edid_224[2]	edid_224[1]	edid_224[0]
0xF6 E1	0x00	edidmemory_reg_ f6e1	r	edid_225[7]	edid_225[6]	edid_225[5]	edid_225[4]	edid_225[3]	edid_225[2]	edid_225[1]	edid_225[0]
0xF6 E2	0x00	edidmemory_reg_ f6e2	r	edid_226[7]	edid_226[6]	edid_226[5]	edid_226[4]	edid_226[3]	edid_226[2]	edid_226[1]	edid_226[0]
0xF6 E3	0x00	edidmemory_reg_ f6e3	r	edid_227[7]	edid_227[6]	edid_227[5]	edid_227[4]	edid_227[3]	edid_227[2]	edid_227[1]	edid_227[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF6 E4	0x00	edidmemory_reg_ f6e4	r	edid_228[7]	edid_228[6]	edid_228[5]	edid_228[4]	edid_228[3]	edid_228[2]	edid_228[1]	edid_228[0]
0xF6 E5	0x00	edidmemory_reg_ f6e5	r	edid_229[7]	edid_229[6]	edid_229[5]	edid_229[4]	edid_229[3]	edid_229[2]	edid_229[1]	edid_229[0]
0xF6 E6	0x00	edidmemory_reg_ f6e6	r	edid_230[7]	edid_230[6]	edid_230[5]	edid_230[4]	edid_230[3]	edid_230[2]	edid_230[1]	edid_230[0]
0xF6 E7	0x00	edidmemory_reg_ f6e7	r	edid_231[7]	edid_231[6]	edid_231[5]	edid_231[4]	edid_231[3]	edid_231[2]	edid_231[1]	edid_231[0]
0xF6 E8	0x00	edidmemory_reg_ f6e8	r	edid_232[7]	edid_232[6]	edid_232[5]	edid_232[4]	edid_232[3]	edid_232[2]	edid_232[1]	edid_232[0]
0xF6 E9	0x00	edidmemory_reg_ f6e9	r	edid_233[7]	edid_233[6]	edid_233[5]	edid_233[4]	edid_233[3]	edid_233[2]	edid_233[1]	edid_233[0]
0xF6 EA	0x00	edidmemory_reg_ f6ea	r	edid_234[7]	edid_234[6]	edid_234[5]	edid_234[4]	edid_234[3]	edid_234[2]	edid_234[1]	edid_234[0]
0xF6 EB	0x00	edidmemory_reg_ f6eb	r	edid_235[7]	edid_235[6]	edid_235[5]	edid_235[4]	edid_235[3]	edid_235[2]	edid_235[1]	edid_235[0]
0xF6 EC	0x00	edidmemory_reg_ f6ec	r	edid_236[7]	edid_236[6]	edid_236[5]	edid_236[4]	edid_236[3]	edid_236[2]	edid_236[1]	edid_236[0]
0xF6 ED	0x00	edidmemory_reg_ f6ed	r	edid_237[7]	edid_237[6]	edid_237[5]	edid_237[4]	edid_237[3]	edid_237[2]	edid_237[1]	edid_237[0]
0xF6 EE	0x00	edidmemory_reg_ f6ee	r	edid_238[7]	edid_238[6]	edid_238[5]	edid_238[4]	edid_238[3]	edid_238[2]	edid_238[1]	edid_238[0]
0xF6 EF	0x00	edidmemory_reg_ f6ef	r	edid_239[7]	edid_239[6]	edid_239[5]	edid_239[4]	edid_239[3]	edid_239[2]	edid_239[1]	edid_239[0]
0xF6 F0	0x00	edidmemory_reg_ f6f0	r	edid_240[7]	edid_240[6]	edid_240[5]	edid_240[4]	edid_240[3]	edid_240[2]	edid_240[1]	edid_240[0]
0xF6 F1	0x00	edidmemory_reg_ f6f1	r	edid_241[7]	edid_241[6]	edid_241[5]	edid_241[4]	edid_241[3]	edid_241[2]	edid_241[1]	edid_241[0]
0xF6 F2	0x00	edidmemory_reg_ f6f2	r	edid_242[7]	edid_242[6]	edid_242[5]	edid_242[4]	edid_242[3]	edid_242[2]	edid_242[1]	edid_242[0]
0xF6 F3	0x00	edidmemory_reg_ f6f3	r	edid_243[7]	edid_243[6]	edid_243[5]	edid_243[4]	edid_243[3]	edid_243[2]	edid_243[1]	edid_243[0]
0xF6 F4	0x00	edidmemory_reg_ f6f4	r	edid_244[7]	edid_244[6]	edid_244[5]	edid_244[4]	edid_244[3]	edid_244[2]	edid_244[1]	edid_244[0]
0xF6 F5	0x00	edidmemory_reg_ f6f5	r	edid_245[7]	edid_245[6]	edid_245[5]	edid_245[4]	edid_245[3]	edid_245[2]	edid_245[1]	edid_245[0]
0xF6 F6	0x00	edidmemory_reg_ f6f6	r	edid_246[7]	edid_246[6]	edid_246[5]	edid_246[4]	edid_246[3]	edid_246[2]	edid_246[1]	edid_246[0]
0xF6 F7	0x00	edidmemory_reg_ f6f7	r	edid_247[7]	edid_247[6]	edid_247[5]	edid_247[4]	edid_247[3]	edid_247[2]	edid_247[1]	edid_247[0]
0xF6 F8	0x00	edidmemory_reg_ f6f8	r	edid_248[7]	edid_248[6]	edid_248[5]	edid_248[4]	edid_248[3]	edid_248[2]	edid_248[1]	edid_248[0]
0xF6 F9	0x00	edidmemory_reg_ f6f9	r	edid_249[7]	edid_249[6]	edid_249[5]	edid_249[4]	edid_249[3]	edid_249[2]	edid_249[1]	edid_249[0]
0xF6 FA	0x00	edidmemory_reg_ f6fa	r	edid_250[7]	edid_250[6]	edid_250[5]	edid_250[4]	edid_250[3]	edid_250[2]	edid_250[1]	edid_250[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xF6 FB	0x00	edidmemory_reg_f6fb	r	edid_251[7]	edid_251[6]	edid_251[5]	edid_251[4]	edid_251[3]	edid_251[2]	edid_251[1]	edid_251[0]
0xF6 FC	0x00	edidmemory_reg_f6fc	r	edid_252[7]	edid_252[6]	edid_252[5]	edid_252[4]	edid_252[3]	edid_252[2]	edid_252[1]	edid_252[0]
0xF6 FD	0x00	edidmemory_reg_f6fd	r	edid_253[7]	edid_253[6]	edid_253[5]	edid_253[4]	edid_253[3]	edid_253[2]	edid_253[1]	edid_253[0]
0xF6 FE	0x00	edidmemory_reg_f6fe	r	edid_254[7]	edid_254[6]	edid_254[5]	edid_254[4]	edid_254[3]	edid_254[2]	edid_254[1]	edid_254[0]
0xF6 FF	0x00	edidmemory_reg_f6ff	r	edid_255[7]	edid_255[6]	edid_255[5]	edid_255[4]	edid_255[3]	edid_255[2]	edid_255[1]	edid_255[0]

1.14 TX2 TEST MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFBBF	0x00	test_reg_fbbf	rw	-	-	-	-	-	spare_pkt3_en	spare_pkt4_en	-
0xFBC0	0x00	test_reg_fbc0	rw	spare3_header0[7]	spare3_header0[6]	spare3_header0[5]	spare3_header0[4]	spare3_header0[3]	spare3_header0[2]	spare3_header0[1]	spare3_header0[0]
0xFBC1	0x00	test_reg_fbc1	rw	spare3_header1[7]	spare3_header1[6]	spare3_header1[5]	spare3_header1[4]	spare3_header1[3]	spare3_header1[2]	spare3_header1[1]	spare3_header1[0]
0xFBC2	0x00	test_reg_fbc2	rw	spare3_header2[7]	spare3_header2[6]	spare3_header2[5]	spare3_header2[4]	spare3_header2[3]	spare3_header2[2]	spare3_header2[1]	spare3_header2[0]
0xFBC3	0x00	test_reg_fbc3	rw	spare3_byte0[7]	spare3_byte0[6]	spare3_byte0[5]	spare3_byte0[4]	spare3_byte0[3]	spare3_byte0[2]	spare3_byte0[1]	spare3_byte0[0]
0xFBC4	0x00	test_reg_fbc4	rw	spare3_byte1[7]	spare3_byte1[6]	spare3_byte1[5]	spare3_byte1[4]	spare3_byte1[3]	spare3_byte1[2]	spare3_byte1[1]	spare3_byte1[0]
0xFBC5	0x00	test_reg_fbc5	rw	spare3_byte2[7]	spare3_byte2[6]	spare3_byte2[5]	spare3_byte2[4]	spare3_byte2[3]	spare3_byte2[2]	spare3_byte2[1]	spare3_byte2[0]
0xFBC6	0x00	test_reg_fbc6	rw	spare3_byte3[7]	spare3_byte3[6]	spare3_byte3[5]	spare3_byte3[4]	spare3_byte3[3]	spare3_byte3[2]	spare3_byte3[1]	spare3_byte3[0]
0xFBC7	0x00	test_reg_fbc7	rw	spare3_byte4[7]	spare3_byte4[6]	spare3_byte4[5]	spare3_byte4[4]	spare3_byte4[3]	spare3_byte4[2]	spare3_byte4[1]	spare3_byte4[0]
0xFBC8	0x00	test_reg_fbc8	rw	spare3_byte5[7]	spare3_byte5[6]	spare3_byte5[5]	spare3_byte5[4]	spare3_byte5[3]	spare3_byte5[2]	spare3_byte5[1]	spare3_byte5[0]
0xFBC9	0x00	test_reg_fbc9	rw	spare3_byte6[7]	spare3_byte6[6]	spare3_byte6[5]	spare3_byte6[4]	spare3_byte6[3]	spare3_byte6[2]	spare3_byte6[1]	spare3_byte6[0]
0xFBCA	0x00	test_reg_fbca	rw	spare3_byte7[7]	spare3_byte7[6]	spare3_byte7[5]	spare3_byte7[4]	spare3_byte7[3]	spare3_byte7[2]	spare3_byte7[1]	spare3_byte7[0]
0xFBCB	0x00	test_reg_fbcb	rw	spare3_byte8[7]	spare3_byte8[6]	spare3_byte8[5]	spare3_byte8[4]	spare3_byte8[3]	spare3_byte8[2]	spare3_byte8[1]	spare3_byte8[0]
0xFBCC	0x00	test_reg_fbcc	rw	spare3_byte9[7]	spare3_byte9[6]	spare3_byte9[5]	spare3_byte9[4]	spare3_byte9[3]	spare3_byte9[2]	spare3_byte9[1]	spare3_byte9[0]
0xFBCD	0x00	test_reg_fbcd	rw	spare3_byte10[7]	spare3_byte10[6]	spare3_byte10[5]	spare3_byte10[4]	spare3_byte10[3]	spare3_byte10[2]	spare3_byte10[1]	spare3_byte10[0]
0xFBCE	0x00	test_reg_fbce	rw	spare3_byte11[7]	spare3_byte11[6]	spare3_byte11[5]	spare3_byte11[4]	spare3_byte11[3]	spare3_byte11[2]	spare3_byte11[1]	spare3_byte11[0]
0xFBCF	0x00	test_reg_fbcf	rw	spare3_byte12[7]	spare3_byte12[6]	spare3_byte12[5]	spare3_byte12[4]	spare3_byte12[3]	spare3_byte12[2]	spare3_byte12[1]	spare3_byte12[0]
0xFBD0	0x00	test_reg_fbd0	rw	spare3_byte13[7]	spare3_byte13[6]	spare3_byte13[5]	spare3_byte13[4]	spare3_byte13[3]	spare3_byte13[2]	spare3_byte13[1]	spare3_byte13[0]
0xFBD1	0x00	test_reg_fbd1	rw	spare3_byte14[7]	spare3_byte14[6]	spare3_byte14[5]	spare3_byte14[4]	spare3_byte14[3]	spare3_byte14[2]	spare3_byte14[1]	spare3_byte14[0]
0xFBD2	0x00	test_reg_fbd2	rw	spare3_byte15[7]	spare3_byte15[6]	spare3_byte15[5]	spare3_byte15[4]	spare3_byte15[3]	spare3_byte15[2]	spare3_byte15[1]	spare3_byte15[0]
0xFBD3	0x00	test_reg_fbd3	rw	spare3_byte16[7]	spare3_byte16[6]	spare3_byte16[5]	spare3_byte16[4]	spare3_byte16[3]	spare3_byte16[2]	spare3_byte16[1]	spare3_byte16[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFBD4	0x00	test_reg_fbd4	rw	spare3_byte17[7]	spare3_byte17[6]	spare3_byte17[5]	spare3_byte17[4]	spare3_byte17[3]	spare3_byte17[2]	spare3_byte17[1]	spare3_byte17[0]
0xFBD5	0x00	test_reg_fbd5	rw	spare3_byte18[7]	spare3_byte18[6]	spare3_byte18[5]	spare3_byte18[4]	spare3_byte18[3]	spare3_byte18[2]	spare3_byte18[1]	spare3_byte18[0]
0xFBD6	0x00	test_reg_fbd6	rw	spare3_byte19[7]	spare3_byte19[6]	spare3_byte19[5]	spare3_byte19[4]	spare3_byte19[3]	spare3_byte19[2]	spare3_byte19[1]	spare3_byte19[0]
0xFBD7	0x00	test_reg_fbd7	rw	spare3_byte20[7]	spare3_byte20[6]	spare3_byte20[5]	spare3_byte20[4]	spare3_byte20[3]	spare3_byte20[2]	spare3_byte20[1]	spare3_byte20[0]
0xFBD8	0x00	test_reg_fbd8	rw	spare3_byte21[7]	spare3_byte21[6]	spare3_byte21[5]	spare3_byte21[4]	spare3_byte21[3]	spare3_byte21[2]	spare3_byte21[1]	spare3_byte21[0]
0xFBD9	0x00	test_reg_fbd9	rw	spare3_byte22[7]	spare3_byte22[6]	spare3_byte22[5]	spare3_byte22[4]	spare3_byte22[3]	spare3_byte22[2]	spare3_byte22[1]	spare3_byte22[0]
0xFBDA	0x00	test_reg_fbda	rw	spare3_byte23[7]	spare3_byte23[6]	spare3_byte23[5]	spare3_byte23[4]	spare3_byte23[3]	spare3_byte23[2]	spare3_byte23[1]	spare3_byte23[0]
0xFBDB	0x00	test_reg_fbdb	rw	spare3_byte24[7]	spare3_byte24[6]	spare3_byte24[5]	spare3_byte24[4]	spare3_byte24[3]	spare3_byte24[2]	spare3_byte24[1]	spare3_byte24[0]
0xFBDC	0x00	test_reg_fbdc	rw	spare3_byte25[7]	spare3_byte25[6]	spare3_byte25[5]	spare3_byte25[4]	spare3_byte25[3]	spare3_byte25[2]	spare3_byte25[1]	spare3_byte25[0]
0xFBDD	0x00	test_reg_fbdd	rw	spare3_byte26[7]	spare3_byte26[6]	spare3_byte26[5]	spare3_byte26[4]	spare3_byte26[3]	spare3_byte26[2]	spare3_byte26[1]	spare3_byte26[0]
0xFBDE	0x00	test_reg_fbde	rw	spare3_byte27[7]	spare3_byte27[6]	spare3_byte27[5]	spare3_byte27[4]	spare3_byte27[3]	spare3_byte27[2]	spare3_byte27[1]	spare3_byte27[0]
0xFBDF	0x00	test_reg_fbdf	rw	spare3_update	-	-	-	-	-	-	-
0xFBE0	0x00	test_reg_fbe0	rw	spare4_header0[7]	spare4_header0[6]	spare4_header0[5]	spare4_header0[4]	spare4_header0[3]	spare4_header0[2]	spare4_header0[1]	spare4_header0[0]
0xFBE1	0x00	test_reg_fbe1	rw	spare4_header1[7]	spare4_header1[6]	spare4_header1[5]	spare4_header1[4]	spare4_header1[3]	spare4_header1[2]	spare4_header1[1]	spare4_header1[0]
0xFBE2	0x00	test_reg_fbe2	rw	spare4_header2[7]	spare4_header2[6]	spare4_header2[5]	spare4_header2[4]	spare4_header2[3]	spare4_header2[2]	spare4_header2[1]	spare4_header2[0]
0xFBE3	0x00	test_reg_fbe3	rw	spare4_byte0[7]	spare4_byte0[6]	spare4_byte0[5]	spare4_byte0[4]	spare4_byte0[3]	spare4_byte0[2]	spare4_byte0[1]	spare4_byte0[0]
0xFBE4	0x00	test_reg_fbe4	rw	spare4_byte1[7]	spare4_byte1[6]	spare4_byte1[5]	spare4_byte1[4]	spare4_byte1[3]	spare4_byte1[2]	spare4_byte1[1]	spare4_byte1[0]
0xFBE5	0x00	test_reg_fbe5	rw	spare4_byte2[7]	spare4_byte2[6]	spare4_byte2[5]	spare4_byte2[4]	spare4_byte2[3]	spare4_byte2[2]	spare4_byte2[1]	spare4_byte2[0]
0xFBE6	0x00	test_reg_fbe6	rw	spare4_byte3[7]	spare4_byte3[6]	spare4_byte3[5]	spare4_byte3[4]	spare4_byte3[3]	spare4_byte3[2]	spare4_byte3[1]	spare4_byte3[0]
0xFBE7	0x00	test_reg_fbe7	rw	spare4_byte4[7]	spare4_byte4[6]	spare4_byte4[5]	spare4_byte4[4]	spare4_byte4[3]	spare4_byte4[2]	spare4_byte4[1]	spare4_byte4[0]
0xFBE8	0x00	test_reg_fbe8	rw	spare4_byte5[7]	spare4_byte5[6]	spare4_byte5[5]	spare4_byte5[4]	spare4_byte5[3]	spare4_byte5[2]	spare4_byte5[1]	spare4_byte5[0]
0xFBE9	0x00	test_reg_fbe9	rw	spare4_byte6[7]	spare4_byte6[6]	spare4_byte6[5]	spare4_byte6[4]	spare4_byte6[3]	spare4_byte6[2]	spare4_byte6[1]	spare4_byte6[0]
0xFBEA	0x00	test_reg_fbea	rw	spare4_byte7[7]	spare4_byte7[6]	spare4_byte7[5]	spare4_byte7[4]	spare4_byte7[3]	spare4_byte7[2]	spare4_byte7[1]	spare4_byte7[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xFB EB	0x00	test_reg_fbeb	rw	spare4_byte8[7]	spare4_byte8[6]	spare4_byte8[5]	spare4_byte8[4]	spare4_byte8[3]	spare4_byte8[2]	spare4_byte8[1]	spare4_byte8[0]
0xFB EC	0x00	test_reg_fbec	rw	spare4_byte9[7]	spare4_byte9[6]	spare4_byte9[5]	spare4_byte9[4]	spare4_byte9[3]	spare4_byte9[2]	spare4_byte9[1]	spare4_byte9[0]
0xFB ED	0x00	test_reg_fbed	rw	spare4_byte10[7]	spare4_byte10[6]	spare4_byte10[5]	spare4_byte10[4]	spare4_byte10[3]	spare4_byte10[2]	spare4_byte10[1]	spare4_byte10[0]
0xFB EE	0x00	test_reg_fbee	rw	spare4_byte11[7]	spare4_byte11[6]	spare4_byte11[5]	spare4_byte11[4]	spare4_byte11[3]	spare4_byte11[2]	spare4_byte11[1]	spare4_byte11[0]
0xFB EF	0x00	test_reg_fbef	rw	spare4_byte12[7]	spare4_byte12[6]	spare4_byte12[5]	spare4_byte12[4]	spare4_byte12[3]	spare4_byte12[2]	spare4_byte12[1]	spare4_byte12[0]
0xFB F0	0x00	test_reg_fbf0	rw	spare4_byte13[7]	spare4_byte13[6]	spare4_byte13[5]	spare4_byte13[4]	spare4_byte13[3]	spare4_byte13[2]	spare4_byte13[1]	spare4_byte13[0]
0xFB F1	0x00	test_reg_fbf1	rw	spare4_byte14[7]	spare4_byte14[6]	spare4_byte14[5]	spare4_byte14[4]	spare4_byte14[3]	spare4_byte14[2]	spare4_byte14[1]	spare4_byte14[0]
0xFB F2	0x00	test_reg_fbf2	rw	spare4_byte15[7]	spare4_byte15[6]	spare4_byte15[5]	spare4_byte15[4]	spare4_byte15[3]	spare4_byte15[2]	spare4_byte15[1]	spare4_byte15[0]
0xFB F3	0x00	test_reg_fbf3	rw	spare4_byte16[7]	spare4_byte16[6]	spare4_byte16[5]	spare4_byte16[4]	spare4_byte16[3]	spare4_byte16[2]	spare4_byte16[1]	spare4_byte16[0]
0xFB F4	0x00	test_reg_fbf4	rw	spare4_byte17[7]	spare4_byte17[6]	spare4_byte17[5]	spare4_byte17[4]	spare4_byte17[3]	spare4_byte17[2]	spare4_byte17[1]	spare4_byte17[0]
0xFB F5	0x00	test_reg_fbf5	rw	spare4_byte18[7]	spare4_byte18[6]	spare4_byte18[5]	spare4_byte18[4]	spare4_byte18[3]	spare4_byte18[2]	spare4_byte18[1]	spare4_byte18[0]
0xFB F6	0x00	test_reg_fbf6	rw	spare4_byte19[7]	spare4_byte19[6]	spare4_byte19[5]	spare4_byte19[4]	spare4_byte19[3]	spare4_byte19[2]	spare4_byte19[1]	spare4_byte19[0]
0xFB F7	0x00	test_reg_fbf7	rw	spare4_byte20[7]	spare4_byte20[6]	spare4_byte20[5]	spare4_byte20[4]	spare4_byte20[3]	spare4_byte20[2]	spare4_byte20[1]	spare4_byte20[0]
0xFB F8	0x00	test_reg_fbf8	rw	spare4_byte21[7]	spare4_byte21[6]	spare4_byte21[5]	spare4_byte21[4]	spare4_byte21[3]	spare4_byte21[2]	spare4_byte21[1]	spare4_byte21[0]
0xFB F9	0x00	test_reg_fbf9	rw	spare4_byte22[7]	spare4_byte22[6]	spare4_byte22[5]	spare4_byte22[4]	spare4_byte22[3]	spare4_byte22[2]	spare4_byte22[1]	spare4_byte22[0]
0xFB FA	0x00	test_reg_fbfa	rw	spare4_byte23[7]	spare4_byte23[6]	spare4_byte23[5]	spare4_byte23[4]	spare4_byte23[3]	spare4_byte23[2]	spare4_byte23[1]	spare4_byte23[0]
0xFB FB	0x00	test_reg_fbfb	rw	spare4_byte24[7]	spare4_byte24[6]	spare4_byte24[5]	spare4_byte24[4]	spare4_byte24[3]	spare4_byte24[2]	spare4_byte24[1]	spare4_byte24[0]
0xFB FC	0x00	test_reg_fbfc	rw	spare4_byte25[7]	spare4_byte25[6]	spare4_byte25[5]	spare4_byte25[4]	spare4_byte25[3]	spare4_byte25[2]	spare4_byte25[1]	spare4_byte25[0]
0xFB FD	0x00	test_reg_fbfd	rw	spare4_byte26[7]	spare4_byte26[6]	spare4_byte26[5]	spare4_byte26[4]	spare4_byte26[3]	spare4_byte26[2]	spare4_byte26[1]	spare4_byte26[0]
0xFB FE	0x00	test_reg_fbfe	rw	spare4_byte27[7]	spare4_byte27[6]	spare4_byte27[5]	spare4_byte27[4]	spare4_byte27[3]	spare4_byte27[2]	spare4_byte27[1]	spare4_byte27[0]
0xFB FF	0x00	test_reg_fbff	rw	spare4_update	-	-	-	-	-	-	-

1.15 ENCODER MAP

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE4 00	0x12	power mode	rw	dac4_enable	dac5_enable	dac6_enable	dac1_enable	dac2_enable	dac3_enable	pll_pdn	sleep_mode
0xE4 01	0x00	mode select	rw	-	func_mode[2]	func_mode[1]	func_mode[0]	-	-	-	-
0xE4 02	0x20	mode register 0	rw	-	-	yuv_out	rgb_sync_en	matrix_prog_en	crt_blk_en	-	-
0xE4 03	0x03	ed/hd csc matrix 0	rw	-	-	-	-	-	-	matrix_gy[1]	matrix_gy[0]
0xE4 04	0xF0	ed/hd csc matrix 1	rw	matrix_gu[1]	matrix_gu[0]	matrix_gv[1]	matrix_gv[0]	matrix_bu[1]	matrix_bu[0]	matrix_rv[1]	matrix_rv[0]
0xE4 05	0x4E	ed/hd csc matrix 2	rw	matrix_gy[9]	matrix_gy[8]	matrix_gy[7]	matrix_gy[6]	matrix_gy[5]	matrix_gy[4]	matrix_gy[3]	matrix_gy[2]
0xE4 06	0x0E	ed/hd csc matrix 3	rw	matrix_gu[9]	matrix_gu[8]	matrix_gu[7]	matrix_gu[6]	matrix_gu[5]	matrix_gu[4]	matrix_gu[3]	matrix_gu[2]
0xE4 07	0x24	ed/hd csc matrix 4	rw	matrix_gv[9]	matrix_gv[8]	matrix_gv[7]	matrix_gv[6]	matrix_gv[5]	matrix_gv[4]	matrix_gv[3]	matrix_gv[2]
0xE4 08	0x92	ed/hd csc matrix 5	rw	matrix_bu[9]	matrix_bu[8]	matrix_bu[7]	matrix_bu[6]	matrix_bu[5]	matrix_bu[4]	matrix_bu[3]	matrix_bu[2]
0xE4 09	0x7C	ed/hd csc matrix 6	rw	matrix_rv[9]	matrix_rv[8]	matrix_rv[7]	matrix_rv[6]	matrix_rv[5]	matrix_rv[4]	matrix_rv[3]	matrix_rv[2]
0xE4 0A	0x00	dac 4, dac 5, dac 6 output levels	rw	dac4to6_tuning[7]	dac4to6_tuning[6]	dac4to6_tuning[5]	dac4to6_tuning[4]	dac4to6_tuning[3]	dac4to6_tuning[2]	dac4to6_tuning[1]	dac4to6_tuning[0]
0xE4 0B	0x00	dac 1, dac 2, dac 3 output levels	rw	dac1to3_tuning[7]	dac1to3_tuning[6]	dac1to3_tuning[5]	dac1to3_tuning[4]	dac1to3_tuning[3]	dac1to3_tuning[2]	dac1to3_tuning[1]	dac1to3_tuning[0]
0xE4 0D	0x00	dac power mode	rw	-	-	-	-	-	lp_en_f	lp_en_e	lp_en_d
0xE4 18	0x00	vbi data control	rw	vbi_from_anc_in	-	-	-	-	-	-	-
0xE4 29	0x01	dac_4_5_sel	rw	-	dac1_sel[2]	dac1_sel[1]	dac1_sel[0]	-	dac2_sel[2]	dac2_sel[1]	dac2_sel[0]
0xE4 2A	0x23	dac_6_1_sel	rw	-	dac3_sel[2]	dac3_sel[1]	dac3_sel[0]	-	dac4_sel[2]	dac4_sel[1]	dac4_sel[0]
0xE4 2B	0x45	dac_2_3_sel	rw	-	dac5_sel[2]	dac5_sel[1]	dac5_sel[0]	-	dac6_sel[2]	dac6_sel[1]	dac6_sel[0]
0xE4 30	0x00	ed/hd mode register 1	rw	hd_enc_ip_mode[4]	hd_enc_ip_mode[3]	hd_enc_ip_mode[2]	hd_enc_ip_mode[1]	hd_enc_ip_mode[0]	-	encode_mode_hdtv[1]	encode_mode_hdtv[0]
0xE4 31	0x00	ed/hd mode register 2	rw	sharp_en	limit_sel_hdtv[1]	limit_sel_hdtv[0]	vbi_data_en	hdtv_flat_tp	hdtv_tp_en	-	pixel_data_en
0xE4 32	0x00	ed/hd mode register 3	rw	cgms_crc_hdtv	cgms_en_hdtv	uv_del_hdtv[2]	uv_del_hdtv[1]	uv_del_hdtv[0]	y_del_hdtv[2]	y_del_hdtv[1]	y_del_hdtv[0]
0xE4 33	0x68	ed/hd mode register 4	rw	db_en_hdtv	-	ssaf_422	-	sinc_filt_en	-	-	-

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE434	0x48	ed/hd mode register 5	rw	no_wraparound	-	-	-	-	-	-	-
0xE435	0x00	ed/hd mode register 6	rw	adapt_en	adapt_bc	gamma_en_hdtv	gamma_curve_b_hdtv	-	sync_on_prpb	-	-
0xE436	0xA0	ed/hd y level	rw	y_colour[7]	y_colour[6]	y_colour[5]	y_colour[4]	y_colour[3]	y_colour[2]	y_colour[1]	y_colour[0]
0xE437	0x80	ed/hd cr level	rw	cr_colour[7]	cr_colour[6]	cr_colour[5]	cr_colour[4]	cr_colour[3]	cr_colour[2]	cr_colour[1]	cr_colour[0]
0xE438	0x80	ed/hd cb level	rw	cb_colour[7]	cb_colour[6]	cb_colour[5]	cb_colour[4]	cb_colour[3]	cb_colour[2]	cb_colour[1]	cb_colour[0]
0xE439	0x40	ed/hd mode register 7	rw	-	-	eia_cea_861_mode_hd	-	-	-	-	-
0xE440	0x00	ed/hd sharpness filter gain	rw	kb[3]	kb[2]	kb[1]	kb[0]	ka[3]	ka[2]	ka[1]	ka[0]
0xE441	0x00	ed/hd c gms data_1	rw	-	-	-	-	cgms_hdtv[19]	cgms_hdtv[18]	cgms_hdtv[17]	cgms_hdtv[16]
0xE442	0x00	ed/hd c gms data_2	rw	cgms_hdtv[15]	cgms_hdtv[14]	cgms_hdtv[13]	cgms_hdtv[12]	cgms_hdtv[11]	cgms_hdtv[10]	cgms_hdtv[9]	cgms_hdtv[8]
0xE443	0x00	ed/hd c gms data_3	rw	cgms_hdtv[7]	cgms_hdtv[6]	cgms_hdtv[5]	cgms_hdtv[4]	cgms_hdtv[3]	cgms_hdtv[2]	cgms_hdtv[1]	cgms_hdtv[0]
0xE444	0x00	ed/hd gamma a0	rw	gamma_a0_hdtv[7]	gamma_a0_hdtv[6]	gamma_a0_hdtv[5]	gamma_a0_hdtv[4]	gamma_a0_hdtv[3]	gamma_a0_hdtv[2]	gamma_a0_hdtv[1]	gamma_a0_hdtv[0]
0xE445	0x00	ed/hd gamma a1	rw	gamma_a1_hdtv[7]	gamma_a1_hdtv[6]	gamma_a1_hdtv[5]	gamma_a1_hdtv[4]	gamma_a1_hdtv[3]	gamma_a1_hdtv[2]	gamma_a1_hdtv[1]	gamma_a1_hdtv[0]
0xE446	0x00	ed/hd gamma a2	rw	gamma_a2_hdtv[7]	gamma_a2_hdtv[6]	gamma_a2_hdtv[5]	gamma_a2_hdtv[4]	gamma_a2_hdtv[3]	gamma_a2_hdtv[2]	gamma_a2_hdtv[1]	gamma_a2_hdtv[0]
0xE447	0x00	ed/hd gamma a3	rw	gamma_a3_hdtv[7]	gamma_a3_hdtv[6]	gamma_a3_hdtv[5]	gamma_a3_hdtv[4]	gamma_a3_hdtv[3]	gamma_a3_hdtv[2]	gamma_a3_hdtv[1]	gamma_a3_hdtv[0]
0xE448	0x00	ed/hd gamma a4	rw	gamma_a4_hdtv[7]	gamma_a4_hdtv[6]	gamma_a4_hdtv[5]	gamma_a4_hdtv[4]	gamma_a4_hdtv[3]	gamma_a4_hdtv[2]	gamma_a4_hdtv[1]	gamma_a4_hdtv[0]
0xE449	0x00	ed/hd gamma a5	rw	gamma_a5_hdtv[7]	gamma_a5_hdtv[6]	gamma_a5_hdtv[5]	gamma_a5_hdtv[4]	gamma_a5_hdtv[3]	gamma_a5_hdtv[2]	gamma_a5_hdtv[1]	gamma_a5_hdtv[0]
0xE44A	0x00	ed/hd gamma a6	rw	gamma_a6_hdtv[7]	gamma_a6_hdtv[6]	gamma_a6_hdtv[5]	gamma_a6_hdtv[4]	gamma_a6_hdtv[3]	gamma_a6_hdtv[2]	gamma_a6_hdtv[1]	gamma_a6_hdtv[0]
0xE44B	0x00	ed/hd gamma a7	rw	gamma_a7_hdtv[7]	gamma_a7_hdtv[6]	gamma_a7_hdtv[5]	gamma_a7_hdtv[4]	gamma_a7_hdtv[3]	gamma_a7_hdtv[2]	gamma_a7_hdtv[1]	gamma_a7_hdtv[0]
0xE44C	0x00	ed/hd gamma a8	rw	gamma_a8_hdtv[7]	gamma_a8_hdtv[6]	gamma_a8_hdtv[5]	gamma_a8_hdtv[4]	gamma_a8_hdtv[3]	gamma_a8_hdtv[2]	gamma_a8_hdtv[1]	gamma_a8_hdtv[0]
0xE44D	0x00	ed/hd gamma a9	rw	gamma_a9_hdtv[7]	gamma_a9_hdtv[6]	gamma_a9_hdtv[5]	gamma_a9_hdtv[4]	gamma_a9_hdtv[3]	gamma_a9_hdtv[2]	gamma_a9_hdtv[1]	gamma_a9_hdtv[0]
0xE44E	0x00	ed/hd gamma b0	rw	gamma_b0_hdtv[7]	gamma_b0_hdtv[6]	gamma_b0_hdtv[5]	gamma_b0_hdtv[4]	gamma_b0_hdtv[3]	gamma_b0_hdtv[2]	gamma_b0_hdtv[1]	gamma_b0_hdtv[0]
0xE44F	0x00	ed/hd gamma b1	rw	gamma_b1_hdtv[7]	gamma_b1_hdtv[6]	gamma_b1_hdtv[5]	gamma_b1_hdtv[4]	gamma_b1_hdtv[3]	gamma_b1_hdtv[2]	gamma_b1_hdtv[1]	gamma_b1_hdtv[0]
0xE450	0x00	ed/hd gamma b2	rw	gamma_b2_hdtv[7]	gamma_b2_hdtv[6]	gamma_b2_hdtv[5]	gamma_b2_hdtv[4]	gamma_b2_hdtv[3]	gamma_b2_hdtv[2]	gamma_b2_hdtv[1]	gamma_b2_hdtv[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE451	0x00	ed/hd gamma b3	rw	gamma_b3_hdtv[7]	gamma_b3_hdtv[6]	gamma_b3_hdtv[5]	gamma_b3_hdtv[4]	gamma_b3_hdtv[3]	gamma_b3_hdtv[2]	gamma_b3_hdtv[1]	gamma_b3_hdtv[0]
0xE452	0x00	ed/hd gamma b4	rw	gamma_b4_hdtv[7]	gamma_b4_hdtv[6]	gamma_b4_hdtv[5]	gamma_b4_hdtv[4]	gamma_b4_hdtv[3]	gamma_b4_hdtv[2]	gamma_b4_hdtv[1]	gamma_b4_hdtv[0]
0xE453	0x00	ed/hd gamma b5	rw	gamma_b5_hdtv[7]	gamma_b5_hdtv[6]	gamma_b5_hdtv[5]	gamma_b5_hdtv[4]	gamma_b5_hdtv[3]	gamma_b5_hdtv[2]	gamma_b5_hdtv[1]	gamma_b5_hdtv[0]
0xE454	0x00	ed/hd gamma b6	rw	gamma_b6_hdtv[7]	gamma_b6_hdtv[6]	gamma_b6_hdtv[5]	gamma_b6_hdtv[4]	gamma_b6_hdtv[3]	gamma_b6_hdtv[2]	gamma_b6_hdtv[1]	gamma_b6_hdtv[0]
0xE455	0x00	ed/hd gamma b7	rw	gamma_b7_hdtv[7]	gamma_b7_hdtv[6]	gamma_b7_hdtv[5]	gamma_b7_hdtv[4]	gamma_b7_hdtv[3]	gamma_b7_hdtv[2]	gamma_b7_hdtv[1]	gamma_b7_hdtv[0]
0xE456	0x00	ed/hd gamma b8	rw	gamma_b8_hdtv[7]	gamma_b8_hdtv[6]	gamma_b8_hdtv[5]	gamma_b8_hdtv[4]	gamma_b8_hdtv[3]	gamma_b8_hdtv[2]	gamma_b8_hdtv[1]	gamma_b8_hdtv[0]
0xE457	0x00	ed/hd gamma b9	rw	gamma_b9_hdtv[7]	gamma_b9_hdtv[6]	gamma_b9_hdtv[5]	gamma_b9_hdtv[4]	gamma_b9_hdtv[3]	gamma_b9_hdtv[2]	gamma_b9_hdtv[1]	gamma_b9_hdtv[0]
0xE458	0x00	ed/hd adaptive filter gain 1	rw	fil_resp_ab[3]	fil_resp_ab[2]	fil_resp_ab[1]	fil_resp_ab[0]	fil_resp_aa[3]	fil_resp_aa[2]	fil_resp_aa[1]	fil_resp_aa[0]
0xE459	0x00	ed/hd adaptive filter gain 2	rw	fil_resp_bb[3]	fil_resp_bb[2]	fil_resp_bb[1]	fil_resp_bb[0]	fil_resp_ba[3]	fil_resp_ba[2]	fil_resp_ba[1]	fil_resp_ba[0]
0xE45A	0x00	ed/hd adaptive filter gain 3	rw	fil_resp_cb[3]	fil_resp_cb[2]	fil_resp_cb[1]	fil_resp_cb[0]	fil_resp_ca[3]	fil_resp_ca[2]	fil_resp_ca[1]	fil_resp_ca[0]
0xE45B	0x00	ed/hd adaptive filter threshold a	rw	thold_a[7]	thold_a[6]	thold_a[5]	thold_a[4]	thold_a[3]	thold_a[2]	thold_a[1]	thold_a[0]
0xE45C	0x00	ed/hd adaptive filter threshold b	rw	thold_b[7]	thold_b[6]	thold_b[5]	thold_b[4]	thold_b[3]	thold_b[2]	thold_b[1]	thold_b[0]
0xE45D	0x00	ed/hd adaptive filter threshold c	rw	thold_c[7]	thold_c[6]	thold_c[5]	thold_c[4]	thold_c[3]	thold_c[2]	thold_c[1]	thold_c[0]
0xE45E	0x00	ed/hd cgms type b register 0	rw	cgmsb_0[5]	cgmsb_0[4]	cgmsb_0[3]	cgmsb_0[2]	cgmsb_0[1]	cgmsb_0[0]	internal_crc_enable	cgms_b_enable
0xE45F	0x00	ed/hd cgms type b register 1	rw	cgmsb_1[7]	cgmsb_1[6]	cgmsb_1[5]	cgmsb_1[4]	cgmsb_1[3]	cgmsb_1[2]	cgmsb_1[1]	cgmsb_1[0]
0xE460	0x00	ed/hd cgms type b register 2	rw	cgmsb_2[7]	cgmsb_2[6]	cgmsb_2[5]	cgmsb_2[4]	cgmsb_2[3]	cgmsb_2[2]	cgmsb_2[1]	cgmsb_2[0]
0xE461	0x00	ed/hd cgms type b register 3	rw	cgmsb_3[7]	cgmsb_3[6]	cgmsb_3[5]	cgmsb_3[4]	cgmsb_3[3]	cgmsb_3[2]	cgmsb_3[1]	cgmsb_3[0]
0xE462	0x00	ed/hd cgms type b register 4	rw	cgmsb_4[7]	cgmsb_4[6]	cgmsb_4[5]	cgmsb_4[4]	cgmsb_4[3]	cgmsb_4[2]	cgmsb_4[1]	cgmsb_4[0]
0xE463	0x00	ed/hd cgms type b register 5	rw	cgmsb_5[7]	cgmsb_5[6]	cgmsb_5[5]	cgmsb_5[4]	cgmsb_5[3]	cgmsb_5[2]	cgmsb_5[1]	cgmsb_5[0]
0xE464	0x00	ed/hd cgms type b register 6	rw	cgmsb_6[7]	cgmsb_6[6]	cgmsb_6[5]	cgmsb_6[4]	cgmsb_6[3]	cgmsb_6[2]	cgmsb_6[1]	cgmsb_6[0]
0xE465	0x00	ed/hd cgms type b register 7	rw	cgmsb_7[7]	cgmsb_7[6]	cgmsb_7[5]	cgmsb_7[4]	cgmsb_7[3]	cgmsb_7[2]	cgmsb_7[1]	cgmsb_7[0]
0xE466	0x00	ed/hd cgms type b register 8	rw	cgmsb_8[7]	cgmsb_8[6]	cgmsb_8[5]	cgmsb_8[4]	cgmsb_8[3]	cgmsb_8[2]	cgmsb_8[1]	cgmsb_8[0]
0xE467	0x00	ed/hd cgms type b register 9	rw	cgmsb_9[7]	cgmsb_9[6]	cgmsb_9[5]	cgmsb_9[4]	cgmsb_9[3]	cgmsb_9[2]	cgmsb_9[1]	cgmsb_9[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE468	0x00	ed/hd cgms type b register 10	rw	cgmsb_10[7]	cgmsb_10[6]	cgmsb_10[5]	cgmsb_10[4]	cgmsb_10[3]	cgmsb_10[2]	cgmsb_10[1]	cgmsb_10[0]
0xE469	0x00	ed/hd cgms type b register 11	rw	cgmsb_11[7]	cgmsb_11[6]	cgmsb_11[5]	cgmsb_11[4]	cgmsb_11[3]	cgmsb_11[2]	cgmsb_11[1]	cgmsb_11[0]
0xE46A	0x00	ed/hd cgms type b register 12	rw	cgmsb_12[7]	cgmsb_12[6]	cgmsb_12[5]	cgmsb_12[4]	cgmsb_12[3]	cgmsb_12[2]	cgmsb_12[1]	cgmsb_12[0]
0xE46B	0x00	ed/hd cgms type b register 13	rw	cgmsb_13[7]	cgmsb_13[6]	cgmsb_13[5]	cgmsb_13[4]	cgmsb_13[3]	cgmsb_13[2]	cgmsb_13[1]	cgmsb_13[0]
0xE46C	0x00	ed/hd cgms type b register 14	rw	cgmsb_14[7]	cgmsb_14[6]	cgmsb_14[5]	cgmsb_14[4]	cgmsb_14[3]	cgmsb_14[2]	cgmsb_14[1]	cgmsb_14[0]
0xE46D	0x00	ed/hd cgms type b register 15	rw	cgmsb_15[7]	cgmsb_15[6]	cgmsb_15[5]	cgmsb_15[4]	cgmsb_15[3]	cgmsb_15[2]	cgmsb_15[1]	cgmsb_15[0]
0xE46E	0x00	ed/hd cgms type b register 16	rw	cgmsb_16[7]	cgmsb_16[6]	cgmsb_16[5]	cgmsb_16[4]	cgmsb_16[3]	cgmsb_16[2]	cgmsb_16[1]	cgmsb_16[0]
0xE480	0x10	sd mode register 0	rw	chroma_filter_sel[2]	chroma_filter_sel[1]	chroma_filter_sel[0]	luma_filter_sel[2]	luma_filter_sel[1]	luma_filter_sel[0]	sd_enc_ip_mode[1]	sd_enc_ip_mode[0]
0xE481	0x01	sd mode register 1	rw	dnr_en	-	-	-	-	-	-	-
0xE482	0x09	sd mode register 2	rw	slope_en	pixel_data_valid	dvd_r	-	pedestal	-	-	wide_uv_filt
0xE483	0x04	sd mode register 3	rw	-	close_cap_even	close_cap_odd	vbi_data_en	uv_ctrl[1]	uv_ctrl[0]	betacam	ped_en_yuv
0xE484	0x00	sd mode register 4	rw	-	color_bar_control	burst_control	chroma_control	ccir601_ccir624	rtcen[1]	rtcen[0]	-
0xE486	0x02	sd mode register 5	rw	-	no_wraparound_sd	-	-	eia_cea_mode	-	ntsc_c_brst_adj[1]	ntsc_c_brst_adj[0]
0xE487	0x00	sd mode register 6	rw	sd_rgb_ip_en	-	sd_autodetect_en	peak_en	setup_en	hue_en	saturate_luma	scale_ycbcr_en
0xE488	0x00	sd mode register 7	rw	gamma_curve_b	gamma_en	-	-	-	db_en	sd_non_interlaced	-
0xE489	0x00	sd mode register 8	rw	-	-	chroma_del[1]	chroma_del[0]	blk_burst_luma	-	sd_under_limiter[1]	sd_under_limiter[0]
0xE48A	0x08	sd timing register 0	rw	-	sd_y_min_value	ydel[1]	ydel[0]	-	-	-	-
0xE48C	0x1F		rw	fsc[7]	fsc[6]	fsc[5]	fsc[4]	fsc[3]	fsc[2]	fsc[1]	fsc[0]
0xE48D	0x7C		rw	fsc[15]	fsc[14]	fsc[13]	fsc[12]	fsc[11]	fsc[10]	fsc[9]	fsc[8]
0xE48E	0xF0		rw	fsc[23]	fsc[22]	fsc[21]	fsc[20]	fsc[19]	fsc[18]	fsc[17]	fsc[16]
0xE48F	0x21		rw	fsc[31]	fsc[30]	fsc[29]	fsc[28]	fsc[27]	fsc[26]	fsc[25]	fsc[24]
0xE490	0x00	sd fsc phase	rw	sub_carrier_phase[9]	sub_carrier_phase[8]	sub_carrier_phase[7]	sub_carrier_phase[6]	sub_carrier_phase[5]	sub_carrier_phase[4]	sub_carrier_phase[3]	sub_carrier_phase[2]
0xE491	0x00	sd closed captioning_1	rw	vbi_ext[7]	vbi_ext[6]	vbi_ext[5]	vbi_ext[4]	vbi_ext[3]	vbi_ext[2]	vbi_ext[1]	vbi_ext[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE492	0x00	sd closed captioning_2	rw	vbi_ext[15]	vbi_ext[14]	vbi_ext[13]	vbi_ext[12]	vbi_ext[11]	vbi_ext[10]	vbi_ext[9]	vbi_ext[8]
0xE493	0x00	sd closed captioning_3	rw	vbi_cap[7]	vbi_cap[6]	vbi_cap[5]	vbi_cap[4]	vbi_cap[3]	vbi_cap[2]	vbi_cap[1]	vbi_cap[0]
0xE494	0x00	sd closed captioning_4	rw	vbi_cap[15]	vbi_cap[14]	vbi_cap[13]	vbi_cap[12]	vbi_cap[11]	vbi_cap[10]	vbi_cap[9]	vbi_cap[8]
0xE495	0x00	sd pedestal register 0	rw	ntsc_ped0[7]	ntsc_ped0[6]	ntsc_ped0[5]	ntsc_ped0[4]	ntsc_ped0[3]	ntsc_ped0[2]	ntsc_ped0[1]	ntsc_ped0[0]
0xE496	0x00	sd pedestal register 1	rw	ntsc_ped1[7]	ntsc_ped1[6]	ntsc_ped1[5]	ntsc_ped1[4]	ntsc_ped1[3]	ntsc_ped1[2]	ntsc_ped1[1]	ntsc_ped1[0]
0xE497	0x00	sd pedestal register 2	rw	ntsc_ped2[7]	ntsc_ped2[6]	ntsc_ped2[5]	ntsc_ped2[4]	ntsc_ped2[3]	ntsc_ped2[2]	ntsc_ped2[1]	ntsc_ped2[0]
0xE498	0x00	sd pedestal register 3	rw	ntsc_ped3[7]	ntsc_ped3[6]	ntsc_ped3[5]	ntsc_ped3[4]	ntsc_ped3[3]	ntsc_ped3[2]	ntsc_ped3[1]	ntsc_ped3[0]
0xE499	0x00	sd cgms/wss_1	rw	wss_en	cgms_even	cgms_odd	cgms_poly	cgms_wss[19]	cgms_wss[18]	cgms_wss[17]	cgms_wss[16]
0xE49A	0x00	sd cgms/wss_2	rw	cgms_wss[15]	cgms_wss[14]	cgms_wss[13]	cgms_wss[12]	cgms_wss[11]	cgms_wss[10]	cgms_wss[9]	cgms_wss[8]
0xE49B	0x00	sd cgms/wss_3	rw	cgms_wss[7]	cgms_wss[6]	cgms_wss[5]	cgms_wss[4]	cgms_wss[3]	cgms_wss[2]	cgms_wss[1]	cgms_wss[0]
0xE49C	0x00	sd scale lsb	rw	sub_carrier_phase[1]	sub_carrier_phase[0]	cr_scale[1]	cr_scale[0]	cb_scale[1]	cb_scale[0]	contrast[1]	contrast[0]
0xE49D	0x00	sd y scale	rw	contrast[9]	contrast[8]	contrast[7]	contrast[6]	contrast[5]	contrast[4]	contrast[3]	contrast[2]
0xE49E	0x00	sd cb scale	rw	cb_scale[9]	cb_scale[8]	cb_scale[7]	cb_scale[6]	cb_scale[5]	cb_scale[4]	cb_scale[3]	cb_scale[2]
0xE49F	0x00	sd cr scale	rw	cr_scale[9]	cr_scale[8]	cr_scale[7]	cr_scale[6]	cr_scale[5]	cr_scale[4]	cr_scale[3]	cr_scale[2]
0xE4A0	0x00	sd hue adjust	rw	hue[7]	hue[6]	hue[5]	hue[4]	hue[3]	hue[2]	hue[1]	hue[0]
0xE4A1	0x00	sd brightness/wss	rw	blank23	setup[6]	setup[5]	setup[4]	setup[3]	setup[2]	setup[1]	setup[0]
0xE4A2	0x00	sd luma ssaf	rw	-	-	-	-	peak[3]	peak[2]	peak[1]	peak[0]
0xE4A3	0x00	sd dnr 0	rw	dnr_coring_gain_a[3]	dnr_coring_gain_a[2]	dnr_coring_gain_a[1]	dnr_coring_gain_a[0]	dnr_coring_gain_b[3]	dnr_coring_gain_b[2]	dnr_coring_gain_b[1]	dnr_coring_gain_b[0]
0xE4A4	0x00	sd dnr 1	rw	dnr_mpeg_1	blk_border_2	dnr_threshold[5]	dnr_threshold[4]	dnr_threshold[3]	dnr_threshold[2]	dnr_threshold[1]	dnr_threshold[0]
0xE4A5	0x00	sd dnr 2	rw	blk_offset[3]	blk_offset[2]	blk_offset[1]	blk_offset[0]	dnr_enable_sharpness	dnr_fmode_control[2]	dnr_fmode_control[1]	dnr_fmode_control[0]
0xE4A6	0x00	sd gamma a0	rw	gamma_a_0[7]	gamma_a_0[6]	gamma_a_0[5]	gamma_a_0[4]	gamma_a_0[3]	gamma_a_0[2]	gamma_a_0[1]	gamma_a_0[0]
0xE4A7	0x00	sd gamma a1	rw	gamma_a_1[7]	gamma_a_1[6]	gamma_a_1[5]	gamma_a_1[4]	gamma_a_1[3]	gamma_a_1[2]	gamma_a_1[1]	gamma_a_1[0]
0xE4A8	0x00	sd gamma a2	rw	gamma_a_2[7]	gamma_a_2[6]	gamma_a_2[5]	gamma_a_2[4]	gamma_a_2[3]	gamma_a_2[2]	gamma_a_2[1]	gamma_a_2[0]

ADD	DEF	REGISTER NAME	ACC	7	6	5	4	3	2	1	0
0xE4 A9	0x00	sd gamma a3	rw	gamma_a_3[7]	gamma_a_3[6]	gamma_a_3[5]	gamma_a_3[4]	gamma_a_3[3]	gamma_a_3[2]	gamma_a_3[1]	gamma_a_3[0]
0xE4 AA	0x00	sd gamma a4	rw	gamma_a_4[7]	gamma_a_4[6]	gamma_a_4[5]	gamma_a_4[4]	gamma_a_4[3]	gamma_a_4[2]	gamma_a_4[1]	gamma_a_4[0]
0xE4 AB	0x00	sd gamma a5	rw	gamma_a_5[7]	gamma_a_5[6]	gamma_a_5[5]	gamma_a_5[4]	gamma_a_5[3]	gamma_a_5[2]	gamma_a_5[1]	gamma_a_5[0]
0xE4 AC	0x00	sd gamma a6	rw	gamma_a_6[7]	gamma_a_6[6]	gamma_a_6[5]	gamma_a_6[4]	gamma_a_6[3]	gamma_a_6[2]	gamma_a_6[1]	gamma_a_6[0]
0xE4 AD	0x00	sd gamma a7	rw	gamma_a_7[7]	gamma_a_7[6]	gamma_a_7[5]	gamma_a_7[4]	gamma_a_7[3]	gamma_a_7[2]	gamma_a_7[1]	gamma_a_7[0]
0xE4 AE	0x00	sd gamma a8	rw	gamma_a_8[7]	gamma_a_8[6]	gamma_a_8[5]	gamma_a_8[4]	gamma_a_8[3]	gamma_a_8[2]	gamma_a_8[1]	gamma_a_8[0]
0xE4 AF	0x00	sd gamma a9	rw	gamma_a_9[7]	gamma_a_9[6]	gamma_a_9[5]	gamma_a_9[4]	gamma_a_9[3]	gamma_a_9[2]	gamma_a_9[1]	gamma_a_9[0]
0xE4 B0	0x00	sd gamma b0	rw	gamma_b_0[7]	gamma_b_0[6]	gamma_b_0[5]	gamma_b_0[4]	gamma_b_0[3]	gamma_b_0[2]	gamma_b_0[1]	gamma_b_0[0]
0xE4 B1	0x00	sd gamma b1	rw	gamma_b_1[7]	gamma_b_1[6]	gamma_b_1[5]	gamma_b_1[4]	gamma_b_1[3]	gamma_b_1[2]	gamma_b_1[1]	gamma_b_1[0]
0xE4 B2	0x00	sd gamma b2	rw	gamma_b_2[7]	gamma_b_2[6]	gamma_b_2[5]	gamma_b_2[4]	gamma_b_2[3]	gamma_b_2[2]	gamma_b_2[1]	gamma_b_2[0]
0xE4 B3	0x00	sd gamma b3	rw	gamma_b_3[7]	gamma_b_3[6]	gamma_b_3[5]	gamma_b_3[4]	gamma_b_3[3]	gamma_b_3[2]	gamma_b_3[1]	gamma_b_3[0]
0xE4 B4	0x00	sd gamma b4	rw	gamma_b_4[7]	gamma_b_4[6]	gamma_b_4[5]	gamma_b_4[4]	gamma_b_4[3]	gamma_b_4[2]	gamma_b_4[1]	gamma_b_4[0]
0xE4 B5	0x00	sd gamma b5	rw	gamma_b_5[7]	gamma_b_5[6]	gamma_b_5[5]	gamma_b_5[4]	gamma_b_5[3]	gamma_b_5[2]	gamma_b_5[1]	gamma_b_5[0]
0xE4 B6	0x00	sd gamma b6	rw	gamma_b_6[7]	gamma_b_6[6]	gamma_b_6[5]	gamma_b_6[4]	gamma_b_6[3]	gamma_b_6[2]	gamma_b_6[1]	gamma_b_6[0]
0xE4 B7	0x00	sd gamma b7	rw	gamma_b_7[7]	gamma_b_7[6]	gamma_b_7[5]	gamma_b_7[4]	gamma_b_7[3]	gamma_b_7[2]	gamma_b_7[1]	gamma_b_7[0]
0xE4 B8	0x00	sd gamma b8	rw	gamma_b_8[7]	gamma_b_8[6]	gamma_b_8[5]	gamma_b_8[4]	gamma_b_8[3]	gamma_b_8[2]	gamma_b_8[1]	gamma_b_8[0]
0xE4 B9	0x00	sd gamma b9	rw	gamma_b_9[7]	gamma_b_9[6]	gamma_b_9[5]	gamma_b_9[4]	gamma_b_9[3]	gamma_b_9[2]	gamma_b_9[1]	gamma_b_9[0]
0xE4 BA	0x00	sd brightness detect	r	bright_detect_val[7]	bright_detect_val[6]	bright_detect_val[5]	bright_detect_val[4]	bright_detect_val[3]	bright_detect_val[2]	bright_detect_val[1]	bright_detect_val[0]
0xE4 BB	0x40	field count	r	-	-	-	-	-	fcount[2]	fcount[1]	fcount[0]

2 SIGNAL DOCUMENTATION

2.1 IO MAP

Reg	Bits	Description	
VIDEO_IN_ID[7:0]			R/W
0x1A00	11111110	This register is used to set the output clock frequencies from the input video formatting block used by both the Serial Video RX and Video TTL input ports. 0x01 - 640x480p@60Hz 0x03 - 720x480p@60Hz 0x04 - 1280x720p@60Hz 0x05 - 1920x1080i@60Hz 0x07 - 720(1440)x480i@60Hz 0x09 - 720(1440)x240p@60Hz 0x0B - (2880)x480i@60Hz 0x0D - (2880)x240p@60Hz 0x0F - 1440x480p@60Hz 0x10 - 1920x1080p@60Hz 0x12 - 720x576p@50Hz 0x13 - 1280x720p@50Hz 0x14 - 1920x1080i@50Hz 0x16 - 720(1440)x576i@50Hz 0x18 - 720(1440)x288p@50Hz 0x1A - (2880)x576i@50Hz 0x1C - (2880)x288p@50Hz 0x1E - 1440x576p@50Hz 0x1F - 1920x1080p@50Hz 0x20 - 1920x1080p@24Hz 0x21 - 1920x1080p@25Hz 0x22 - 1920x1080p@30Hz 0x24 - 2880x480p@60Hz 0x26 - 2880x576p@50Hz 0x80 - 640x350@85hz 0x81 - 640x400@85hz 0x82 - 720x400@85hz 0x83 - 640x480@60hz 0x84 - 640x480@72hz 0x85 - 640x480@75hz 0x86 - 640x480@85hz 0x87 - 800x600@56hz 0x88 - 800x600@60hz 0x89 - 800x600@72hz 0x8A - 800x600@75hz 0x8B - 800x600@85hz 0x8D - 1024x768@60hz 0x8E - 1024x768@70hz 0x8F - 1024x768@75hz 0x90 - 1024x768@85hz 0xFC - 720x288p@50Hz 0xFD - 720x240p@60Hz 0xFE - 720x480i@60Hz 0xFF - 720x576i@50Hz	
TTL_PS444_IN			R/W
0x1A01	0000001<u>0</u>	This bit is used to select the video type sent to the TTL output format block. 0 - Input to TTL output block is real 4:4:4 1 - Input to TTL output block is pseudo 4:4:4	
TTL_OP_FORMAT[3:0]			R/W
0x1A02	0111<u>0000</u>	This signal is used to specify the TTL output format. 0011 - 2 x 8-bit buses, SDR 4:2:2 0100 - 2 x 10-bit buses, SDR 4:2:2 0101 - 2 x 12-bit buses, SDR 4:2:2 0110 - 3 x 8-bit buses, SDR 4:4:4 0111 - 3 x 10-bit buses, SDR 4:4:4 1000 - 3 x 12-bit buses, SDR 4:4:4	

Reg	Bits	Description	
TTL_VID_OUT_EN			R/W
0x1A02	01110000	This bit is used to enable the TTL video output. 0 - Disable TTL output 1 - Enable TTL output	
TTL_OUT_SEL[2:0]			R/W
0x1A02	01110000	This signal is used to select the video source for the TTL video output. 0x00 - From Primary Input Channel 0x01 - From Primary VSP 0x02 - From PtoI Converter 0x03 - From Internal OSD Blend 1 0x04 - From Secondary VSP/PtoI Converter 0x05 - From Secondary Input Channel 0x06 - From RX Input 0x07 - From Internal OSD Blend 2	
TX1_INP_SEL[3:0]			R/W
0x1A03	00000000	This signal is used to select the video source for the HDMI Tx1. 0x00 - From Primary Input Channel 0x01 - From Primary VSP 0x02 - From PtoI Converter 0x03 - From Internal OSD Blend 1 0x04 - From Secondary VSP/PtoI Converter 0x05 - From Secondary Input Channel 0x06 - From RX Input 0x07 - From Internal OSD Blend 2	
TX2_INP_SEL[3:0]			R/W
0x1A03	00000000	This signal is used to select the video source for the HDMI Tx2. 0x00 - From Primary Input Channel 0x01 - From Primary VSP 0x02 - From PtoI Converter 0x03 - From Internal OSD Blend 1 0x04 - From Secondary VSP/PtoI Converter 0x05 - From Secondary Input Channel 0x06 - From RX Input 0x07 - From Internal OSD Blend 2	
HD_ENC_INP_SEL[3:0]			R/W
0x1A04	00000000	This signal is used to select the video source for the HD encoder. When using the encoder in SD only mode, this signal must be set to the same value as sd_enc_inp_sel. 0x00 - From Primary Input Channel 0x01 - From Primary VSP 0x02 - From PtoI Converter 0x03 - From Internal OSD Blend 1 0x04 - From Secondary VSP/PtoI Converter 0x05 - From Secondary Input Channel 0x06 - From RX Input 0x07 - From Internal OSD Blend 2	
SD_ENC_INP_SEL[3:0]			R/W
0x1A04	00000000	This signal is used to select the video source for the SD encoder. When using the encoder in SD only mode, hd_enc_inp_sel must be set to the same value as this signal. 0x00 - From Primary Input Channel 0x01 - From Primary VSP 0x02 - From PtoI Converter 0x03 - From Internal OSD Blend 1 0x04 - From Secondary VSP/PtoI Converter 0x05 - From Secondary Input Channel 0x06 - From RX Input 0x07 - From Internal OSD Blend 2	

Reg	Bits	Description	
SVSP_INP_SEL[3:0]			R/W
0x1A05	0000 0000	This signal is used to select the video source for the Secondary VSP. 0x00 - From Primary Input Channel 0x01 - From Internal OSD Blend 1 0x02 - From Primary VSP 0x03 - From Internal OSD (OSD only, no blend) 0x04 - From Secondary Input Channel 0x05 - From RX Input 0x06 - From Horizontal Prescaler	
PVSP_INP_SEL[3:0]			R/W
0x1A05	0000 0000	This signal is used to select the video source for the Primary VSP. 0x00 - From Primary Input Channel 0x01 - From Internal OSD Blend 1 0x02 - From Secondary Input Channel 0x03 - From RX Input 0x04 - From Secondary VSP 0x05 - From Horizontal Pre-scaler	
P2I_INP_SEL[3:0]			R/W
0x1A06	0000 0000	This signal is used to select the video source for the Progressive to Interlaced converter. 0x00 - From Primary VSP 0x01 - From Internal OSD Blend 1 0x02 - From Secondary Input Channel 0x03 - From RX Input 0x04 - From Primary Input Channel	
OSD_BLEND_INP_SEL[3:0]			R/W
0x1A06	0000 0000	This signal is used to select the video source to the OSD Blend block. 0x00 - From Primary Input Channel 0x01 - From Secondary VSP/Ptol Converter 0x02 - From Primary VSP 0x03 - From Secondary Input Channel 0x04 - From RX Input	
RX_PROC_BYPASS			R/W
0x1A07	0 1100100	This bit is used to bypass the RX input processing for high speed inputs. 0 - Enable RX input processing 1 - Bypass RX input processing	
SDE_ONLY_GATE_HDE_INP			R/W
0x1A07	0 1100100	This bit muxes the SD Encoder data onto the HD Encoder path in SD only modes. This avoids the user having to set the sd and hd enc inp sel values for SD only modes. 0 - Do not gate HD Encoder inputs for SD only mode 1 - use SD inputs for HD Encoder in SD only mode	
S_INP_CHAN_SEL[1:0]			R/W
0x1A07	0110 0100	This signal is used to select the input for the Secondary Input Channel. 00 - Video TTL input (P[35:0]) 01 - EXOSD TTL Input (OSD_IN[23:0]) 10 - RX video 11 - N/A	
P_INP_CHAN_SEL[1:0]			R/W
0x1A07	0110 0100	This signal is used to select the input for the Primary Input Channel. 00 - Video TTL input (P[35:0]) 01 - EXOSD TTL Input (OSD_IN[23:0]) 10 - 48-bit TTL input (OSD_IN[11:0] and P[35:0]) for 3GHz interleaved TTL 11 - Reserved	
AUD_INPUT_MODE[1:0]			R/W
0x1A08	00 000000	This signal is used to select the audio input mode. 00 - Single mode. 01 - Dual mode; TX1 with I2S stream, TX2 with SPDIF stream. 10 - Dual mode; TX1 with SPDIF stream, TX2 with I2S stream. 11 - Dual mode; TX1 with SPDIF stream 1, TX2 with SPDIF stream 2.	

Reg	Bits	Description	
OSD_BLEND_INP_2_SEL[3:0]			R/W
0x1A08	00000000	This signal is used to select the video to be blended on OSD channel 2. 0x00 - From Primary Input Channel 0x01 - From Secondary VSP/Ptol Converter 0x02 - From Primary VSP 0x03 - From Secondary Input Channel 0x04 - From RX Input	
HPS_INP_SEL[3:0]			R/W
0x1A09	00000000	This signal is used to select the video source for the Horizontal pre-scaler (HPS) block 0x00 - From Primary Input Channel 0x01 - From Secondary Input Channel 0x02 - From RX Input 0x03 - From Internal OSD Blend 1	
UPDITHER_LEVEL[1:0]			R/W
0x1A0D	00101000	This signal is used to set the sharpness of the updither block's HPF processing of the video data. When this signal is set to low the characteristic of the dither block's HPF gives smoother output video. When this signal is set to high, the characteristic of the dither block's HPF gives sharper output video. 00 - Low updither 11 - High updither	
UPDITHER_ALPHA[3:0]			R/W
0x1A0D	00101000	This signal is used to set the sharpness gain for the dither block's HPF. The high frequency component from the HPF is multiplied by this value and added back.	
MAX_DIFF_PIXEL[11:0]			R/W
0x1A0E 0x1A0F	00000000 00110010	This signal is used to set the detection threshold for separation of dither input into high and low frequency components.	
MAX_DIFF_SUM[15:0]			R/W
0x1A10 0x1A11	00000000 01100100	This signal is used to set the detection threshold for separation of dither input into high and low frequency components.	
SPI_SLAVE_CPOL			R/W
0x1A14	00001100	This bit is used to select the SPI slave clock polarity. 0 - Idle state, clock is low 1 - Idle state, clock is high	
SPI_SLAVE_CPHA			R/W
0x1A14	00001100	This bit is used to select the SPI slave clock phase. 0 - Negedge used 1 - Posedge used	
SPI_MASTER_CPOL			R/W
0x1A14	00001100	This bit is used to select the SPI master clock polarity. 0 - Idle state, clock is low 1 - Idle state, clock is high	
SPI_MASTER_CPHA			R/W
0x1A14	00001100	This bit is used to select the SPI master clock phase. 0 - Negedge used 1 - Posedge used	
DB_VID_ADJ_PARAMS			R/W
0x1A24	10000001	This bit is used to update video adjustment parameters during the vsync period. 0 - Update vid adjust parameters once they are written to 1 - Update vid adjust parameters only in VBI region	
SATURATION_EN			R/W
0x1A24	10000001	This bit is used to enable the saturation control. 0 - Disable Saturation 1 - Enable Saturation	

Reg	Bits	Description	
BRIGHTNESS_EN			R/W
0x1A24	10000001	This bit is used to enable the brightness control. 0 - Disable Brightness 1 - Enable Brightness	
CONTRAST_EN			R/W
0x1A24	10000001	This bit is used to enable the contrast control. 0 - Disable Contrast 1 - Enable Contrast	
BLANK_LEVEL_Y[11:0]			R/W
0x1A24 0x1A25	10000001 00000000	This signal is used to adjust the blank level of y input to the vid adjust block. 0x000 - y blank level sits at code 0 0x100 - y blank level sits at code 256 decimal	
BLANK_LEVEL_U[11:0]			R/W
0x1A26 0x1A27	10000000 00001000	This signal is used to adjust the blank level of u input to the vid adjust block. 0x000 - u blank level sits at code 0 0x800 - u blank level sits at code 2048 decimal	
BLANK_LEVEL_V[11:0]			R/W
0x1A27 0x1A28	00001000 00000000	This signal is used to adjust the blank level of v input to the vid adjust block 0x000 - v blank level sits at code 0 0x800 - v blank level sits at code 2048 decimal	
SATURATION[7:0]			R/W
0x1A29	10000000	This register is used to adjust the saturation value for U/V channels. The register uses 1.7 notation. 0x00 - Gain of 0 0x80 - Unity Gain 0xFF - Gain of 2	
BRIGHTNESS[7:0]			R/W
0x1A2A	00000000	This register is used to adjust the brightness value for Y channel. The register uses s1.6 notation. 0x7F - (+127) * 8 0x00 - (No adjustment) * 8 0xFF - (-1) * 8	
CONTRAST[7:0]			R/W
0x1A2B	10000000	This register is used to adjust the contrast value for Y channel. This register uses 1.7 notation. 0x00 - Gain of 0 0x80 - Unity gain 0xFF - Gain of 2	
SPI_FILTER_EN			R/W
0x1A2C	00001010	This bit is used to enable the SPI anti glitch filter. 0 - Anti glitch filter disable 1 - Anti glitch filter enable	
SPI_FILTER_SEL			R/W
0x1A2C	00001010	This bit is used to select the response of the SPI anti glitch filter. 0 - 2ns glitch rejection 1 - 5ns glitch rejection	
OSD_VID_DRVSTR[1:0]			R/W
0x1A2C	00001010	This signal is used to set the drive strength for the data output to the OSD and VID pins. 0 - min drive strength 3 - max drive strength	
OSD_CLK_DRVSTR[1:0]			R/W
0x1A2C	00001010	This signal is used to set the drive strength for the clock output to the OSD_clk pin. 0 - min drive strength 3 - max drive strength	
ACE_GAMMA_GAIN[3:0]			R/W
0x1A2D	00000000		

Reg	Bits	Description	
ACE_GAMMA_OFFSET[7:0]			R/W
0x1A2E	00000000		
ACE_GAMMA_FIXED[7:0]			R/W
0x1A2F	00000000		
ACE_ENABLE			R/W
0x1A30	0000000	This bit is used to enable the automatic contrast enhancement (ACE) block. 0 - Bypass A.C.E. 1 - Enable A.C.E.	
ACE_LUMA_GAIN[4:0]			R/W
0x1A30 0x1A33	00000000 11011111	This bit is the MSB of the ace_luma_gain signal. This signal sets the auto-contrast strength level for the luma channel gain. These bits are the LSBs of the ace_luma_gain signal. This signal sets the auto-contrast strength level for the luma channel gain.	
ACE_ROW_OFFSET[9:0]			R/W
0x1A30 0x1A31	00000000 00001010	These bits are the MSBs of the ace_row_offset signal. This signal sets the ACE vertical analysis window - a larger value reduces the vertical window from the last active line upwards. These bits are the LSBs of the ace_row_offset signal. This signal sets the ACE vertical analysis window - a larger value reduces the vertical window from the last active line upwards.	
ACE_COL_OFFSET[9:0]			R/W
0x1A30 0x1A32	00000000 00001010	These bits are the MSBs of the ace_col_offset signal. This signal sets the ACE horizontal analysis window - a larger value reduces the horizontal window from the last active pixel backwards. These bits are the LSBs of the ace_col_offset signal. This signal sets the ACE horizontal analysis window - a larger value reduces the horizontal window from the last active pixel backwards.	
ACE_BLEND_ALPHA[3:0]			R/W
0x1A33	11011111	This signal is used to set the delay for the auto-contrast to take effect.	
ACE_CHROMA_GAIN[3:0]			R/W
0x1A34	10001000	This signal is used to set the auto saturation level for the colour channels.	
ACE_CHROMA_MAX[3:0]			R/W
0x1A34	10001000	This signal is used to set the max auto-saturation for the colour channels.	
TT_TALON_MODE[3:0]			R/W
0x1A39	00000000	This register is used to set the mode for which the DPLL is to configure the clocks. This should be programmed dependant on the section of the device for which the clocks are being configured. 0000 - ADC mode 0001 - HDMI mode 0010 - SD2X mode 0011 - VFM mode 0100 - EXT1X mode 0101 - EXT2X mode 0110 - EXT4X mode 0111 - SD4X mode 1000 - SD8x mode 1001 - Audio mode 1010 - Scaler mode	
PVSP_PERIOD_FORCE			R/W
0x1A3A	0000000	This bit is used to enable a forced value onto the DPLL in open loop operation. 0 - Use auto open_loop_period for Primary VSP 1 - Use forced open_loop_period value for Primary VSP	
PVSP_VID_CLK_UPDATE			R/W
0x1A3A	00000000	This bit is used to trigger the open loop period to be captured in the DPLL. A low to high transition triggers the action. 0 - Do not update open_loop_period in DPLL 1 - Update open_loop_period in DPLL	

Reg	Bits	Description
PVSP_VID_CLK_PERIOD[33:0]		R/W
0x1A3A 0x1A3B 0x1A3C 0x1A3D 0x1A3E	000000<u>00</u> <u>10101010</u> <u>10100000</u> <u>00000000</u> <u>00000000</u>	This register is used to set the open_loop_period of the DPLL section. This should be programmed based on the value calculated from the given equations.
SVSP_PERIOD_FORCE		R/W
0x1A3F	<u>0</u>0000000	This bit is used to enable a forced value onto the DPLL in open loop operation. 0 - Use auto open_loop_period value for secondary VSP 1 - Use forced open_loop_period value for secondary VSP
SVSP_VID_CLK_UPDATE		R/W
0x1A3F	000<u>0</u>0000	This bit is used to trigger the open loop period to be captured in the DPLL. A low to high transition triggers the action. 0 - Do not update open_loop_period in DPLL 1 - Update open_loop_period in DPLL
SVSP_VID_CLK_PERIOD[33:0]		R/W
0x1A3F 0x1A40 0x1A41 0x1A42 0x1A43	000000<u>00</u> <u>00000000</u> <u>00000000</u> <u>00000000</u> <u>00000000</u>	This signal is used to set the open_loop_period of the DPLL section. This should be programmed based on the value calculated from the given equations.
PVSP_FREQ_SEL		R/W
0x1A44	<u>0</u>0000000	This bit is used to manually configure the vertical frequency for the Primary VSP. 0 - 59.94Hz or 23.9Hz 1 - 60Hz or 24Hz
PVSP_TRACK_EN		R/W
0x1A44	0<u>0</u>000000	This bit is used to enable tracking of the frequency error to reduce the number of dropped/repeated frames for the Primary VSP. 0 - Do not adjust for frequency difference between input and output vertical sync 1 - Adjust for frequency difference between input and output vertical sync
SVSP_FREQ_SEL		R/W
0x1A44	0000<u>0</u>000	This bit is used to manually select the vertical frequency for the Secondary VSP. 0 - 59.94Hz or 23.9Hz 1 - 60Hz or 24Hz
SVSP_TRACK_EN		R/W
0x1A44	00000<u>0</u>00	This bit is used to enable tracking of the frequency error to reduce the number of dropped/repeated frames for the Secondary VSP. 0 - Do not adjust for frequency difference between input and output vertical sync 1 - Adjust for frequency difference between input and output vertical sync
DID_A[7:0]		R/W
0x1A44A	<u>01010100</u>	This register is used to specify the value of the DID sent in the ancillary stream with VBI decoded data.
SDID_A[7:0]		R/W
0x1A44B	<u>10101000</u>	This register is used to specify the value of the SDID sent in the ancillary stream with VBI decoded data.
VBI_SRC		R/W
0x1A44C	<u>0</u>0100000	This bit is used to choose the source of the VBI data. 0 - VBI data from ancillary input 1 - VBI data from SPI input

Reg	Bits	Description
CCAP_ODD_EN		R/W
0x1A44 C	00100000	This bit is used to enable/disable closed caption data extraction on the odd field. 0 - Disable closed caption data extraction on odd field 1 - Enable closed caption data extraction on odd field
CCAP_EVEN_EN		R/W
0x1A44 C	00100000	This bit is used to enable/disable closed caption data extraction on the even field. 0 - Disable closed caption data extraction on even field 1 - Enable closed caption data extraction on even field
CGMS_ANC_EN		R/W
0x1A44 C	00100000	This bit is used to enable/disable CGMS data extraction on the even field. 0 - Disable CGMS data extraction on even field 1 - Enable CGMS data extraction on even field
WSS_ANC_EN		R/W
0x1A44 C	00100000	This bit is used to enable/disable WSS data extraction on the even field. 0 - Disable WSS data extraction on even field 1 - Enable WSS data extraction on even field
ANC_DELAY[1:0]		R/W
0x1A44 D	00011000	This bit is used to set the delay on ancillary data in vsyncs. The interlaced input delay will be in fields and the progressive delay will be in frames. Decoded data is firstly transferred onto input vsync and then output vsync, this will be the base delay with a setting of 0. Every increment above this adds one input vsync delay.
PVSP_IN_VS_GATE_MAN[2:0]		R/W
0x1A44 E	00000000	This signal is used to specify the number of vsync pulses gated before sending information to the Primary VSP tracking block. This is necessary in modes where frame rate conversion is to be performed. 000 - 0 001 - 1 010 - 2 011 - 3 100 - 4 101 - 5 110 - 6 111 - 7
PVSP_ERR_SEL		R/W
0x1A44 E	00000000	This bit is used to choose between phase locked loop and frequency locked loop for the Primary VSP frame tracking mode. 0 - Phase error 1 - Frequency error
SVSP_IN_VS_GATE_MAN[2:0]		R/W
0x1A44 F	00000000	This signal is used to specify the number of vsync pulses gated before sending information to the Secondary VSP tracking block. This is necessary in modes where frame rate conversion is to be performed. 000 - 0 001 - 1 010 - 2 011 - 3 100 - 4 101 - 5 110 - 6 111 - 7
SVSP_ERR_SEL		R/W
0x1A44 F	00000000	This bit is used to choose between phase locked loop and frequency locked loop for the Secondary VSP frame tracking mode. 0 - Phase error 1 - Frequency error
SDRAM_SIZE[3:0]		R/W
0x1A5 B	00100010	This signal is used to specify the SDRAM size. All values other than those specified here are reserved. 0001 - individual SDRAM is 256Mbit 0010 - individual SDRAM is 512Mbit 0011 - individual SDRAM is 1Gbit 0100 - individual SDRAM is 2Gbit

Reg	Bits	Description
WORD_SIZE[3:0]		R/W
0x1A5C	0011[0101]	This signal is used to specify the word size on the user interface. The data width to the SDRAM is half of this value. All other values are reserved 0010 - 32 bits 0011 - 64 bits
BURST_LENGTH[2:0]		R/W
0x1A5D 0x1A5E	00100001 00000000	This signal is used to indicate the burst length of the read/write transaction. 010 - Burst of 4 011 - Burst of 8.
STORE_UNMASKED_IRQS		R/W
0x1A69	00000000	This bit is used to specify whether the HDMI status flags for any HDMI interrupt should be triggered regardless of whether the mask bits are set. This bit allows an HDMI interrupt to trigger and allows this interrupt to be read back through the corresponding status bit without triggering an interrupt on the interrupt pin. The status is stored until the clear bit is used to clear the status register and allows another interrupt to occur. 0 - Do not store triggered interrupts 1 - Store triggered interrupts
INTRQ_DUR_SEL[1:0]		R/W
0x1A69	00000000	This signal is used to set the interrupt signal duration for the Serial Video RX interrupts output on pin INT2. 00 - 4 Xtal periods 01 - 16 Xtal periods 10 - 64 Xtal periods 11 - Active until cleared
RX_CABLE_DET_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive cable detect interrupt. 0 - Do not clear the cable detect interrupt 1 - Clear the cable detect interrupt
RX_TMDSPLL_LCK_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive TMDS PLL lock interrupt. 0 - Do not clear the TMDS PLL lock interrupt 1 - Clear the TMDS PLL lock interrupt
RX_TMDS_CLK_DET_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive TMDS clock detect interrupt. 0 - Do not clear the TMDS clock detect interrupt 1 - Clear the TMDS clock detect interrupt
RX_VIDEO_3D_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive 3D video interrupt. 0 - Do not clear the 3D video detect interrupt 1 - Clear the 3D video detect interrupt
RX_AV_MUTE_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive AV mute interrupt. 0 - Do not clear the AV mute interrupt 1 - Clear the AV mute interrupt
RX_HDMI_MODE_CLR		SC
0x1A6A	00000000	This bit is used to clear the level sensitive HDMI mode interrupt. 0 - Do not clear the HDMI mode interrupt 1 - Clear the HDMI mode interrupt
RX_GEN_CTL_PCKT_CLR		SC
0x1A6B	00000000	This bit is used to clear the level sensitive general control packet interrupt. 0 - Do not clear the general control packet interrupt 1 - Clear the general control packet detect interrupt
RX_GAMUT_MDATA_PCKT_CLR		SC
0x1A6B	00000000	This bit is used to clear the level sensitive gamut meta data packet interrupt. 0 - Do not clear the gamut meta-data packet detect interrupt 1 - Clear the gamut meta-data packet detect interrupt

Reg	Bits	Description	
RX_ISRC2_PCKT_CLR			SC
0x1A6B	00<u>0</u>00000	This bit is used to clear the level sensitive ISRC2 interrupt. 0 - Do not clear the ISRC2 packet detect interrupt 1 - Clear theISRC2 packet detect interrupt	
RX_ISRC1_PCKT_CLR			SC
0x1A6B	000<u>0</u>0000	This bit is used to clear the level sensitive ISRC1 interrupt. 0 - Do not clear the ISRC1 packet detect interrupt 1 - Clear theISRC1 packet detect interrupt	
RX_VS_INFO_FRM_CLR			SC
0x1A6B	0000<u>0</u>000	This signal is used to clear the level sensitive VS info-frame interrupt. 0 - Do not clear the VS info-frame detect interrupt 1 - Clear the VS info-frame detect interrupt	
RX_MS_INFO_FRM_CLR			SC
0x1A6B	00000<u>0</u>00	This bit is used to clear the level sensitive MS info-frame interrupt. 0 - Do not clear the MS info-frame detect interrupt 1 - Clear the MS info-frame detect interrupt	
RX_SPD_INFO_FRM_CLR			SC
0x1A6B	000000<u>0</u>0	This bit is used to clear the level sensitive SPD info-frame interrupt. 0 - Do not clear the SPD info-frame detect interrupt 1 - Clear the SPD info-frame detect interrupt	
RX_AVI_INFO_FRM_CLR			SC
0x1A6B	0000000<u>0</u>	This bit is used to clear the level sensitive AVI info-frame interrupt. 0 - Do not clear the AVI info-frame detect interrupt 1 - Clear the AVI info-frame detect interrupt	
RX_CABLE_DET_MB2			R/W
0x1A6C	00<u>0</u>00000	This bit is used to enable the cable detect level sensitive interrupt on the INT2 pin. 0 - Disable the cable detect interrupt for the INT2 pin 1 - Enable the cable detect interrupt for the INT2 pin	
RX_TMDSPLL_LCK_MB2			R/W
0x1A6C	000<u>0</u>0000	This bit is used to enable the TMDS PLL lock level sensitive interrupt on the INT2 pin. 0 - Disable the TMDS PLL LCK interrupt for the INT 2 pin. 1 - Enable TMDS PLL LCK interrupt for the INT2 pin.	
RX_TMDS_CLK_MB2			R/W
0x1A6C	0000<u>0</u>000	This bit is used to enable the TMDS clock detect level sensitive interrupt on the INT2 pin. 0 - Disable the TMDS clock detect interrupt for the INT2 pin 1 - Enable the TMDS clock detect interrupt for the INT2 pin	
RX_VIDEO_3D_MB2			R/W
0x1A6C	00000<u>0</u>00	This bit is used to enable the 3D video detect level sensitive interrupt on the INT2 pin. 0 - Disable the 3D Video detect interrupt for the INT2 pin 1 - Enable the 3D Video detect interrupt for the INT2 pin	
RX_AV_MUTE_MB2			R/W
0x1A6C	000000<u>0</u>0	This bit is used to enable the AV mute level sensitive interrupt on the INT2 pin. 0 - Disable the AV mute interrupt for the INT2 pin 1 - Enable the AV mute interrupt for the INT2 pin	
RX_HDMI_MODE_MB2			R/W
0x1A6C	0000000<u>0</u>	This bit is used to enable the HDMI mode level sensitive interrupt on the INT2 pin. 0 - Disable the HDMI mode interrupt for the INT2 pin 1 - Enable the HDMI mode interrupt for the INT2 pin	
RX_GEN_CTL_PCKT_MB2			R/W
0x1A6D	<u>0</u>0000000	This bit is used to enable the general control packet detect level sensitive interrupt on the INT2 pin. 0 - Disable the general control packet detect interrupt for the INT2 pin 1 - Enable the general control packet detect interrupt for the INT2 pin	
RX_GAMUT_MDATA_PCKT_MB2			R/W
0x1A6D	0<u>0</u>000000	This bit is used to enable the gamut meta-data packet detect level sensitive interrupt on the INT2 pin. 0 - Disable the gamut meta-data packet detect interrupt for the INT2 pin 1 - Enable the gamut meta-data packet detect interrupt for the INT2 pin	

Reg	Bits	Description	
RX_ISRC2_PCKT_MB2			R/W
0x1A6D	00<u>0</u>00000	This bit is used to enable the ISRC2 packet detect level sensitive interrupt on the INT2 pin. 0 - Disable the ISRC2 packet detect interrupt for the INT2 pin 1 - Enable the ISRC2 control packet detect interrupt for the INT2 pin	
RX_ISRC1_PCKT_MB2			R/W
0x1A6D	000<u>0</u>0000	This bit is used to enable the ISRC1 packet detect level sensitive interrupt on the INT2 pin. 0 - Disable the ISRC1 packet detect interrupt for the INT2 pin 1 - Enable the ISRC1 packet detect interrupt for the INT2 pin	
RX_VS_INFO_FRM_MB2			R/W
0x1A6D	0000<u>0</u>000	This bit is used to enable the VS info-frame detect level sensitive interrupt on the INT2 pin. 0 - Disable the VS info-frame detect interrupt for the INT2 pin 1 - Enable the VS info-frame detect interrupt for the INT2 pin	
RX_MS_INFO_FRM_MB2			R/W
0x1A6D	00000<u>0</u>00	This bit is used to enable the MS info-frame detect level sensitive interrupt on the INT2 pin. 0 - Disable the MS info-frame detect interrupt for the INT2 pin 1 - Enable the MS info-frame interrupt for the INT2 pin	
RX_SPD_INFO_FRM_MB2			R/W
0x1A6D	000000<u>0</u>0	This bit is used to enable the SPD info-frame detect level sensitive interrupt on the INT2 pin. 0 - Disable the SPD info-frame detect interrupt for the INT2 pin 1 - Enable the SPD info-frame detect interrupt for the INT2 pin	
RX_AVI_INFO_FRM_MB2			R/W
0x1A6D	0000000<u>0</u>	This bit is used to enable the AVI info-frame detect level sensitive interrupt on the INT2 pin. 0 - Disable the AVI info-frame detect interrupt for the INT2 pin 1 - Enable the AVI info-frame detect interrupt for the INT2 pin	
RX_VS_INF_CKS_ERR_EDGE_CLR			SC
0x1A70	00<u>0</u>00000	This bit is used to clear the VS info-frame detect edge sensitive status interrupt. 0 - Do not clear the VS info-frame check sum error interrupt 1 - Clear the VS info-frame check sum error interrupt	
RX_MS_INF_CKS_ERR_EDGE_CLR			SC
0x1A70	000<u>0</u>0000	This bit is used to clear the MS info-frame edge sensitive status interrupt. 0 - Do not clear the MS info-frame check sum error interrupt 1 - Clear the MS info-frame check sum error interrupt	
RX_SPD_INF_CKS_ERR_EDGE_CLR			SC
0x1A70	0000<u>0</u>000	This bit is used to clear the SPD info-frame edge sensitive status interrupt. 0 - Do not clear the SPD info-frame check sum error interrupt 1 - Clear the SPD info-frame check sum error interrupt	
RX_AVI_INF_CKS_ERR_EDGE_CLR			SC
0x1A70	00000<u>0</u>00	This bit is used to clear the AVI info-frame checksum error edge sensitive status interrupt. 0 - Do not clear the AVI check sum error interrupt 1 - Clear the AVI check sum error interrupt	
RX_DEEPCOLOR_CHNG_EDGE_CLR			SC
0x1A70	000000<u>0</u>0	This bit is used to clear the deep colour change edge sensitive status interrupt. 0 - Do not clear the deep color change interrupt 1 - Clear the deep color change interrupt	
RX_TMDS_CLK_CHNG_EDGE_CLR			SC
0x1A70	0000000<u>0</u>	This bit is used to clear the TMDS clock change edge sensitive status interrupt. 0 - Do not clear the TMDS clock change interrupt 1 - Clear the TMDS clock change interrupt	
RX_PKT_ERR_EDGE_CLR			SC
0x1A71	<u>0</u>0000000	This bit is used to clear the packet error edge sensitive status interrupt. 0 - Do not clear the packet error detect interrupt 1 - Clear the packet error detect interrupt	
RX_GAMUT_MDATA_PCKT_EDGE_CLR			SC
0x1A71	0<u>0</u>000000	This bit is used to clear the gamut meta-data packet edge sensitive status interrupt. 0 - Do not clear the gamut meta-data edge detect interrupt 1 - Clear the gamut meta-data edge detect interrupt	

Reg	Bits	Description	
RX_ISRC2_PCKT_EDGE_CLR			SC
0x1A71	00<u>0</u>00000	This bit is used to clear the ISRC2 packet detect edge sensitive status interrupt. 0 - Do not clear the ISRC2 packet detect interrupt 1 - Clear theISRC2 packet detect interrupt	
RX_ISRC1_PCKT_EDGE_CLR			SC
0x1A71	000<u>0</u>0000	This bit is used to clear the ISRC1 packet detect edge sensitive status interrupt. 0 - Do not clear the ISRC1 packet_edge detect interrupt 1 - Clear theISRC1 packet_edge detect interrupt	
RX_VS_INFO_FRM_EDGE_CLR			SC
0x1A71	0000<u>0</u>000	This bit is used to clear the VS info-frame detect edge sensitive status interrupt. 0 - Do not clear the VS info-frame detect interrupt 1 - Clear the VS info-frame detect interrupt	
RX_MS_INFO_FRM_EDGE_CLR			SC
0x1A71	00000<u>0</u>00	This bit is used to clear the MS info-frame detect edge sensitive status interrupt. 0 - Do not clear the MS info-frame detect interrupt 1 - Clear the MS info-frame detect interrupt	
RX_SPD_INFO_FRM_EDGE_CLR			SC
0x1A71	000000<u>0</u>0	This bit is used to clear the SPD info-frame detect edge sensitive status interrupt. 0 - Do not clear the SPD info-frame detect interrupt 1 - Clear the SPD info-frame detect interrupt	
RX_AVI_INFO_FRM_EDGE_CLR			SC
0x1A71	0000000<u>0</u>	This bit is used to clear the AVI info-frame detect edge sensitive status interrupt. 0 - Do not clear the AVI info-frame detect interrupt 1 - Clear the AVI info-frame detect interrupt	
RX_VS_INF_CKS_ERR_EDGE_MB2			R/W
0x1A72	00<u>0</u>00000	This bit is used to enable the VS info-frame check sum error edge sensitive interrupt on the INT2 pin. 0 - Disable the VS info-frame check sum error interrupt for the INT2 pin 1 - Enable the VS info-frame check sum error interrupt for the INT2 pin	
RX_MS_INF_CKS_ERR_EDGE_MB2			R/W
0x1A72	000<u>0</u>0000	This bit is used to enable the MS info-frame check sum error edge sensitive interrupt on the INT2 pin. 0 - Disable the MS info-frame check sum error interrupt for the INT2 pin 1 - Enable the MS info-frame check sum error interrupt for the INT2 pin	
RX_SPD_INF_CKS_ERR_EDGE_MB2			R/W
0x1A72	0000<u>0</u>000	This bit is used to enable the SPD info-frame check sum error edge sensitive interrupt on the INT2 pin. 0 - Disable the SPD info-frame check sum error interrupt for the INT2 pin 1 - Enable the SPD info-frame check sum error interrupt for the INT2 pin	
RX_AVI_INF_CKS_ERR_EDGE_MB2			R/W
0x1A72	00000<u>0</u>00	This bit is used to enable the AVI check sum error edge sensitive interrupt on the INT2 pin. 0 - Disable the AVI check sum error interrupt for the INT2 pin 1 - Enable the AVI check sum error interrupt for the INT2 pin	
RX_DEEPCOLOR_CHNG_EDGE_MB2			R/W
0x1A72	000000<u>0</u>0	This bit is used to enable the deep colorchange edge sensitive interrupt on the INT2 pin. 0 - Disable the deep colorchange interrupt for the INT2 pin 1 - Enable the deep color change interrupt for the INT2 pin	
RX_TMDS_CLK_CHNG_EDGE_MB2			R/W
0x1A72	0000000<u>0</u>	This bit is used to enable the TMDS clock change edge sensitive interrupt on the INT2 pin. 0 - Disable the TMDS clock change interrupt for the INT2 pin 1 - Enable the TMDS clock change interrupt for the INT2 pin	
RX_PKT_ERR_EDGE_MB2			R/W
0x1A73	<u>0</u>0000000	This bit is used to enable the packet error detect edge sensitive interrupt on the INT2 pin. 0 - Disable the packet error detect interrupt for the INT2 pin 1 - Enable the packet error detect interrupt for the INT2 pin.	
RX_GAMUT_MDATA_PCKT_EDGE_MB2			R/W
0x1A73	0<u>0</u>000000	This bit is used to enable the gamut meta-data packet detect edge sensitive interrupt on the INT2 pin. 0 - Disable the gamut meta-data packet detect interrupt for the INT2 pin 1 - Enable the gamut meta-data packet detect interrupt for the INT2 pin	

Reg	Bits	Description	
RX_ISRC2_PCKT_EDGE_MB2			R/W
0x1A73	00<u>0</u>00000	This bit is used to enable the ISRC2 packet detect edge sensitive interrupt on the INT2 pin. 0 - Disable the ISRC2 packet detect interrupt for the INT2 pin 1 - Enable the ISRC2 packet detect interrupt for the INT2 pin	
RX_ISRC1_PCKT_EDGE_MB2			R/W
0x1A73	000<u>0</u>0000	This bit is used to enable the ISRC1 packet detect edge sensitive interrupt on the INT2 pin. 0 - Disable the ISRC1 packet detect interrupt for the INT2 pin 1 - Enable the ISRC1 packet detect interrupt for the INT2 pin	
RX_VS_INFO_FRM_EDGE_MB2			R/W
0x1A73	0000<u>0</u>000	This bit is used to enable the VS info-frame detect edge sensitive interrupt on the INT2 pin. 0 - Disable the VS info-frame detect interrupt for the INT2 pin 1 - Enable the VS info-frame detect interrupt for the INT2 pin	
RX_MS_INFO_FRM_EDGE_MB2			R/W
0x1A73	00000<u>0</u>00	This bit is used to enable the MS info-frame detect edge sensitive interrupt on the INT2 pin. 0 - Disable the MS info-frame detect interrupt for the INT2 pin 1 - Enable the MS info-frame detect interrupt for the INT2 pin	
RX_SPD_INFO_FRM_EDGE_MB2			R/W
0x1A73	000000<u>0</u>0	This bit is used to enable the SPD info-frame detect edge sensitive interrupt on the INT2 pin. 0 - Disable the SPD info-frame detect interrupt for the INT2 pin 1 - Enable the SPD info-frame detect interrupt for the INT2 pin	
RX_AVI_INFO_FRM_EDGE_MB2			R/W
0x1A73	0000000<u>0</u>	This bit is used to enable the AVI info-frame detect edge sensitive interrupt on the INT2 pin. 0 - Disable the AVI info-frame detect interrupt for the INT2 pin 1 - Enable the AVI info-frame detect interrupt for the INT2 pin	
VSP_INT_POL[1:0]			R/W
0x1A76	0000<u>00</u>00	This signal is used to control the VSP interrupt polarity. 00 - VSP interrupt is logical AND of VSP/OSD interrupts 01 - VSP interrupt is inverted logical AND of VSP/OSD interrupts 10 - VSP interrupt is logical OR of VSP/OSD interrupts 11 - VSP interrupt is inverted logical OR of VSP/OSD interrupts	
TX_INT_POL[1:0]			R/W
0x1A76	000000<u>00</u>	This signal is used to control the TX interrupt polarity. 00 - Tx interrupt is logical AND of Tx1/Tx2 interrupts 01 - Tx interrupt is inverted logical AND of Tx1/Tx2 interrupts 10 - Tx interrupt is logical OR of Tx1/Tx2 interrupts 11 - Tx interrupt is inverted logical OR of Tx1/Tx2 interrupts	
DE_H_BEG_POS[11:10]			R/W
0x1A78	<u>00</u>000010	This signal is used to select the DE horizontal beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX	
HS_BEG_POS[11:10]			R/W
0x1A78	00<u>00</u>0010	This signal is used to select the HS beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX	
HS_END_POS[11:10]			R/W
0x1A78	0000<u>00</u>10	This signal is used to select the HS ending position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX	
HPA_MAN_VALUE_A			R/W
0x1A78	000000<u>1</u>0	This bit is used to control the TMDS receiver HPA override. 0 - HPA normal operation (auto-mode) 1 - Assert HPA manually	
DE_V_BEG_E_POS[6:0]			R/W
0x1A79	<u>1011011</u>0	This signal is used to select the DE vertical beginning position for even fields if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when lcount reaches 0xXX on even fields	

Reg	Bits	Description
DE_V_BEG_O_POS[6:0]		R/W
0x1A79 0x1A7A	10110110 11010000	This signal is used to select the DE vertical beginning position for odd fields if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when lcount reaches 0xXX on even fields
DE_H_BEG_POS[9:0]		R/W
0x1A7A 0x1A7B	11010000 10100000	This signal is used to select the DE horizontal beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX
HS_BEG_POS[9:0]		R/W
0x1A7C 0x1A7D	00000001 11000000	This signal is used to select the HS beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert hs when hcount reaches 0xXX
HS_END_POS[9:0]		R/W
0x1A7D 0x1A7E	11000000 00000000	This signal is used to select the HS ending position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - release hs when hcount reaches 0xXX
VS_H_BEG_O_POS[10:0]		R/W
0x1A7E 0x1A7F	00000000 00010000	This signal is used to select the VS horizontal beginning position on odd fields, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert vs when hcount reaches 0xXX on odd fields
VS_H_BEG_E_POS[10:0]		R/W
0x1A80 0x1A81	00001110 00000000	This signal is used to select the VS horizontal beginning position on even fields, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert vs when hcount reaches 0xXX on even fields
FLD_CAP_POS		R/W
0x1A81	00000000	This signal is used to select the capture position of the fld signal. The captured fld signal will be used to generate the right VS timing when in interlace format. 0x0 - fld value is captured when de_in goes low 0x1 - fld value is capture in middle of active line.
VS_V_BEG_POS[5:0]		R/W
0x1A81 0x1A82	00000000 01001100	This signal is used to select the VS vertical beginning position, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert vs when lcount reaches 0xXX
VS_V_END_POS[5:0]		R/W
0x1A82	01001100	This signal is used to select the VS vertical ending position, if CEA 861 timing generation is enabled and manual values selected. 0xXX - release vs when lcount reaches 0xXX
HPS_POWER_DOWN		R/W
0x1A85	10000101	Powers down the horizontal pre-scaler block (HPS). Powered down by default to save power 0 - HPS is active 1 - HPS Block is powered down
HPS_FILT_BYPASS		R/W
0x1A85	10000101	This bit bypasses filtering done before downsampling. Aliasing may occur if this filtering is not done 0 - Do not bypass 1 - Bypass
HPS_BYPASS_DOWNSAMPLE		R/W
0x1A85	10000101	This bit bypasses data downsampling. Use this control to just filter but not downsample video data 0 - Do not bypass 1 - Bypass

Reg	Bits	Description	
HPS_PHASE_SEL_DOWNSAMPLE			R/W
0x1A85	10000101	This bit selects whether the downsampling should start by keeping or dropping the first pixel when in a 2 - 1 downsampling. ' 0 - Start by keeping the first Pixel 1 - Start by dropping the first Pixel	
HPS_FILT_MODE[1:0]			R/W
0x1A85	10000101	The filter has 2 operating modes. Mode 0 has higher bandpass but less aliasing rejection. 0 - Filter mode 0 1 - Filer mode 1 2 - Unused 3 - Unused	
TX1_ARC_POWERDOWN			R/W
0x1A87	00000000	This bit is used to powerdown the TX1 ARC block. 0 - Power up ARC 1 - Power down ARC	
TX1_ARC_BIAS_HYST_ADJ			R/W
0x1A88	00000000	This bit is used to control the addition of hysteresis to the TX1 ARC. 0 - Normal 1 - ADD hysteresis	
TX1_ARC_S_END_HPD			R/W
0x1A88	00000000	This bit is used to control the TX1 ARC input. 0 - Common-mode input 1 - Single-ended mode input	
TX2_ARC_POWERDOWN			R/W
0x1A89	00000000	This bit is used to powerdown the TX2 ARC block. 0 - Power up ARC 1 - Power down ARC	
TX2_ARC_BIAS_HYST_ADJ			R/W
0x1A8A	00000000	This bit is used to control the addition of hysteresis to the TX2 ARC. 0 - Normal 1 - ADD hysteresis	
TX2_ARC_S_END_HPD			R/W
0x1A8A	00000000	This bit is used to control the TX2 ARC input. 0 - Common-mode input 1 - Single-ended mode input	
PVSP_TRACK_OFFSET[20:0]			R/W
0x1A94 0x1A95 0x1A96	00000000 00000000 00000000	This signal is used to program the delay on the output timing of vsyncs from the Primary VSP. 0 - input and output vsyncs are coincident 1 - 1 xltal clk between input and output vsync	
SVSP_TRACK_OFFSET[20:0]			R/W
0x1A97 0x1A98 0x1A99	00000000 00000000 00000000	This signal is used to program the delay on the output timing of vsyncs from the Secondary VSP. 0 - input and output vsync coincident 1 - 1 xltal clk between input nad output vsync	
CK_DRV_STR[7:0]			R/W
0x1A9D	00000000	This signal is used to set the drive strength for the clock output to the DDR2 memory. 0x00 - Min drive strength 0xFF - Max drive strength	
DQS_DRV_STR[7:0]			R/W
0x1A9E	00000000	This signal is used to set the drive strength for the DQS outputs to the DDR2 memory. 0x00 - Min drive strength 0xFF - Max drive strength	

Reg	Bits	Description
PLLDLL_SEL_DIV[5:0]		R/W
0x1AA2	01001000	This signal is used to control the DDR2 PLL loop divider. The DDR2 clock frequency is given by: fxtal * i2c_plldll_sel_div / i2c_plldll_pre_div.
PLLDLL_PRE_DIV[1:0]		R/W
0x1AA3	00010001	This signal is used to control the DDR2 PLL pre divider.
BIT_ERR_RB_EN		R/W
0x1AA7	00000110	This bit is used to enable the readback of the bit error bus from the loopback memory test. 0 - Read back values from vih/vil testing 1 - Read back bit error bus from loopback test
DM_DQ_DRV_STR[1:0]		R/W
0x1AA7	00000110	This signal is used to control the drive strength setting for the clock output to the DDR2 memory. 00 - Min drive strength 11 - Max drive strength
MEM_TEST_EN		R/W
0x1AA7	00000110	This bit is used to enable the DDR2 memory BIST to be run just after initialisation and the results to be reported back. 0 - Disable ddr memory bist 1 - Enable ddr memory bist
RW_CTRL_OE		R/W
0x1AA8	00000000	This bit is used to control the output enable for external memory read/write signals (ras, cas, clock, address…). 0 - Input 1 - Output
DDR2_CK_OE		R/W
0x1AA8	00000000	This bit is used to control the output enable for external memory clock signal. 0 - Input 1 - Output
MEM_RW_CTRL_DRV_STR[1:0]		R/W
0x1AA8	00000000	This signal is used to control the drive strength setting for the read/write control signals to the DDR2 memory (ras, cas, wr, cke, a[12:0]…). 00 - Min drive strength 11 - Max drive strength
1X_DDR		R/W
0x1AB2	00000000	This bit is used to select between single or dual DDR2 memories. 0 - 2 x DDR2 memories available 1 - 1 x DDR2 memory available
SPI_LOOP_THROUGH		R/W
0x1AB6	00000000	This bit is used to enable SPI loop through mode. In loop through mode, Serial Port 1 (SCK1, MOSI1, MISO1 and CS1) is connected to the Serial Port 2 (SCK2, MOSI2, MISO2, CS2). 0 - Regular SPI mode 1 - SPI slave clock routed to SPI master clock output
ARC_PINS_OE_MAN		R/W
0x1ACA	00000000	This bit is used to control the output enable for ARC outputs. 0 - Input 1 - Output
ARC_PINS_OE_MAN_EN		R/W
0x1ACB	00000000	This bit is used to control the manual override for ARC outputs. 0 - Auto 1 - Manual override
INT_PIN_OE[2:0]		R/W
0x1ACC	10000000	This signal is used to enable the INT0, INT1 and INT2 interrupt pins. INT0 is linked to the OSD interrupts, INT1 is linked to the HDMI TX interrupts and INT2 is linked to the Serial Video RX interrupts. 000 - All interrupts tristated 001 - INT0 interrupt enabled 010 - INT1 interrupt enabled 100 - INT2 interrupt enabled 111 - All interrupts enabled

Reg	Bits	Description
INT_PIN_OD_EN[2:0]		R/W
0x1ACC	10000000	This signal is used to select whether the interrupt pins are configured as TTL or as open drain. INT0 is linked to the OSD interrupts, INT1 is linked to the HDMI TX interrupts and INT2 is linked to the Serial Video RX interrupts. 000 - All interrupts TTL 001 - INT0 open drain 010 - INT1 open drain 100 - INT2 open drain 111 - All interrupts open drain
SPI1_CS_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi1 chip select. 0 - Input 1 - Output
SPI1_MISO_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi1 'master in slave out'. 0 - Input 1 - Output
SPI1_MOSI_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi1 'master out slave in'. 0 - Input 1 - Output
SPI1_SCLK_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi1 serial clock. 0 - Input 1 - Output
SPI2_CS_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi2 chip select. 0 - Input 1 - Output
SPI2_MISO_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi2 'master in slave out'. 0 - Input 1 - Output
SPI2_MOSI_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi2 'master out slave in'. 0 - Input 1 - Output
SPI2_SCLK_OE_MAN		R/W
0x1ACD	00000000	This bit is used to control the output enable for spi2 serial clock. 0 - Input 1 - Output
SPI1_CS_OE_MAN_EN		R/W
0x1ACE	11111111	This bit is used to control the output enable manual override for spi1_cs. 0 - Auto 1 - manual override
SPI1_MISO_OE_MAN_EN		R/W
0x1ACE	11111111	This bit is used to control the output enable manual override for spi1_miso. 0 - Auto 1 - Manual override
SPI1_MOSI_OE_MAN_EN		R/W
0x1ACE	11111111	This bit is used to control the output enable manual override for spi1_mosi. 0 - Auto 1 - Manual override
SPI1_SCLK_OE_MAN_EN		R/W
0x1ACE	11111111	This bit is used to control the output enable manual override for spi1_sclk. 0 - Auto 1 - Manual override

Reg	Bits	Description	
SPI2_CS_OE_MAN_EN			R/W
0x1ACE	1111<u>1</u>111	This bit is used to control the output enable manual override for spi2_cs. 0 - Auto 1 - Manual override	
SPI2_MISO_OE_MAN_EN			R/W
0x1ACE	11111<u>1</u>11	This bit is used to control the output enable manual override for spi2_miso. 0 - Auto 1 - Manual override	
SPI2_MOSI_OE_MAN_EN			R/W
0x1ACE	111111<u>1</u>1	This bit is used to control the output enable manual override for spi2_mosi. 0 - Auto 1 - Manual override	
SPI2_SCLK_OE_MAN_EN			R/W
0x1ACE	1111111<u>1</u>	This bit is used to control the output enable manual override for spi2_sclk. 0 - Auto 1 - Manual override	
RB_CHIP_ID[15:0]			R
0x1AD0 0x1AD1	<u>00000000</u> <u>00000000</u>	This signal is used as a readback of the chip ID. 0x6080 - ES1 0x6081 - ES2	
RB_CHIP_ID[16]			R
0x1AD3	0000000<u>0</u>	Readback of Macrovision enabled / disabled 0 - Rovi Enabled 1 - Rovi Disabled	
RB_RX_5V			R
0x1ADF	0000<u>0</u>000	This bit is used to indicate that the Serial Video RX has detected 5V on its input. 0 - No 5V detected by the Serial Video RX 1 - 5V detected by the Serial Video RX	
RB_RX_TMDS_CLK_DET			R
0x1ADF	0000<u>0</u>000	This bit is used to indicate if there is a clock on the Serial Video RX input lines. 0 - No TMDS clock detected on the Serial Video RX input lines 1 - TMDS clock detected on Serial Video RX input lines	
RB_RX_HPA_A			R
0x1ADF	00000<u>0</u>00	This bit is used to readback the Serial Video RX hot plug assert state. 0 - Hot plug assert inactive (low) 1 - Hot plug assert active (high)	
LBK_TEST_RESULT			R
0x1AE1	000000<u>0</u>0	This bit is used to readback the DDR2 loopback test error result. 0 - No error detected 1 - Errors detected	
LBK_TEST_DONE			R
0x1AE1	0000000<u>0</u>	This bit is used to readback the DDR2 loopback test has completed. 0 - Test not complete 1 - Loopback test finished	
RB_V_IN_XTAL_CLKS[20:0]			R
0x1AE2 0x1AE3 0x1AE4	000<u>00000</u> <u>00000000</u> <u>00000000</u>	This signal is used to readback the number of XTAL clocks counted within one input vsync period. This is a measure of the PVSP input Vsync and does not work with the SVSP. 0 - 0 xtal clocks 1 - 1 xtal clocks etc	
RB_RX_INTRQ_RAW			R
0x1AF6	<u>0</u>0000000	This bit is set high if: 1) Any of the level or edge sensitive interrupt status flags are set high AND 2) Their corresponding mask bits are set 0 - Raw bit is low 1 - Raw bit is high	

Reg	Bits	Description	
RX_CABLE_DET_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the cable detect interrupt. 0 - The cable detect status has not changed, interrupt not generated 1 - The cable detect status has changed, and an interrupt was generated	
RX_TMDSPLL_LCK_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the TMDS PLL lock interrupt. 0 - The TMDS PLL lock status has not changed, interrupt not generated 1 - The TMDS PLL lock status has changed, and an interrupt was generated	
RX_TMDS_CLK_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the TMDS clock detection interrupt. 0 - The TMDS PLL clock status has not changed, interrupt not generated 1 - The TMDS PLL clock status has changed, and an interrupt was generated	
RX_VIDEO_3D_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the 3D Video interrupt. 0 - The 3D Video interrupt status has not changed, interrupt not generated 1 - The 3D Video interrupt status has changed, and an interrupt was generated	
RX_AV_MUTE_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the AV mute interrupt. 0 - The AV mute status has not changed, interrupt not generated 1 - The AV mute status has changed, and an interrupt was generated	
RX_HDMI_MODE_ST			R
0x1AF6	00000000	This bit is used to indicate the latched status of the HDMI mode interrupt. 0 - The HDMI mode status has not changed, interrupt not generated 1 - The HDMI mode status has changed, and an interrupt was generated	
RX_GEN_CTL_PCKT_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the general control packet interrupt. 0 - The general control packet status has not changed, interrupt not generated 1 - The general control packet status has changed, and an interrupt was generated	
RX_GAMUT_MDATA_PCKT_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the gamut meta-data packet interrupt. 0 - The gamut meta-data packet status has not changed, interrupt not generated 1 - The gamut meta-data packet status has changed, and an interrupt was generated	
RX_ISRC2_PCKT_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the ISRC2 packet interrupt. 0 - The ISRC2 packet status has not changed, interrupt not generated 1 - The ISRC2 packet status has changed, and an interrupt was generated	
RX_ISRC1_PCKT_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the ISRC1 packet interrupt. 0 - The ISRC1 packet status has not changed, interrupt not generated 1 - The ISRC1 packet status has changed, and an interrupt was generated	
RX_VS_INFO_FRM_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the VS info-frame detect interrupt. 0 - The VS info-frame status has not changed, interrupt not generated 1 - The VS info-frame status has changed, and an interrupt was generated	
RX_MS_INFO_FRM_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the MS info-frame detect interrupt. 0 - The MS info-frame status has not changed, interrupt not generated 1 - The MS info-frame status has changed, and an interrupt was generated	
RX_SPD_INFO_FRM_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the SPD info-frame detect interrupt. 0 - The SPD info-frame status has not changed, interrupt not generated 1 - The SPD info-frame status has changed, and an interrupt was generated	
RX_AVI_INFO_FRM_ST			R
0x1AF7	00000000	This bit is used to indicate the latched status of the AVI info-frame detect interrupt. 0 - The AVI info-frame status has not changed, interrupt not generated 1 - The AVI info-frame status has changed, and an interrupt was generated	

Reg	Bits	Description	
RX_VS_INF_CKS_ERR_EDGE_ST			R
0x1AF8	00<u>0</u>00000	This bit is used to indicate the latched status of the VS info-frame checksum error interrupt. 0 - No change in the VS info-frame checksum error, no interrupt generated 1 - An error in the VS info-frame checksum error triggered this interrupt.	
RX_MS_INF_CKS_ERR_EDGE_ST			R
0x1AF8	000<u>0</u>0000	This bit is used to indicate the latched status of the MS info-frame checksum error interrupt. 0 - No change in the MS info-frame checksum error, no interrupt generated 1 - An error in the MS info-frame checksum error triggered this interrupt.	
RX_SPD_INF_CKS_ERR_EDGE_ST			R
0x1AF8	0000<u>0</u>000	This bit is used to indicate the latched status of the SPD info-frame checksum error interrupt. 0 - No change in the SPD info-frame checksum error, no interrupt generated 1 - An error in the SPD info-frame checksum error triggered this interrupt.	
RX_AVI_INF_CKS_ERR_EDGE_ST			R
0x1AF8	00000<u>0</u>00	This bit is used to indicate the latched status of the AVI info-frame checksum error interrupt. 0 - No change in the AVI info-frame checksum error, no interrupt generated 1 - An error in the AVI info-frame checksum error triggered this interrupt.	
RX_DEEPCOLOR_CHNG_EDGE_ST			R
0x1AF8	000000<u>0</u>0	This bit is used to indicate the latched status of the deep colour change interrupt. 0 - The deep color status has not changed, interrupt not generated 1 - The deep color status has changed, and an interrupt was generated	
RX_TMDS_CLK_CHNG_EDGE_ST			R
0x1AF8	0000000<u>0</u>	This bit is used to indicate the latched status of the TMDS clock change interrupt. 0 - The TMDS clock detection status has not changed, interrupt not generated 1 - The TMDS clock detection status has changed, and an interrupt was generated	
RX_PKT_ERR_EDGE_ST			R
0x1AF9	<u>0</u>0000000	This bit is used to indicate the latched status of the packet error interrupt. 0 - No uncorrectable error detected in a packet header, interrupt not generated 1 - An uncorrectable error was detected in an unknown packet (packet header), and an interrupt was generated	
RX_GAMUT_MDATA_PCKT_EDGE_ST			R
0x1AF9	0<u>0</u>000000	This bit is used to indicate the latched status of the gamut meta-data packet interrupt. 0 - The gamut meta-data packet status has not changed, interrupt not generated 1 - The gamut meta-data packet status has changed, and an interrupt was generated	
RX_ISRC2_PCKT_EDGE_ST			R
0x1AF9	00<u>0</u>00000	This bit is used to indicate the latched status of the ISRC2 packet interrupt. 0 - The ISRC2 packet status has not changed, interrupt not generated 1 - The ISRC2 packet status has changed, and an interrupt was generated	
RX_ISRC1_PCKT_EDGE_ST			R
0x1AF9	000<u>0</u>0000	This bit is used to indicate the latched status of the ISRC1 packet interrupt. 0 - The ISRC1 packet status has not changed, interrupt not generated 1 - The ISRC1 packet status has changed, and an interrupt was generated	
RX_VS_INFO_FRM_EDGE_ST			R
0x1AF9	0000<u>0</u>000	This bit is used to indicate the latched status of the VS info-frame detect interrupt. 0 - The Vendor Specific info-frame status has not changed, interrupt not generated 1 - The Vendor Specific info-frame status has changed, and an interrupt was generated	
RX_MS_INFO_FRM_EDGE_ST			R
0x1AF9	00000<u>0</u>00	This bit is used to indicate the latched status of the MS info-frame detect interrupt. 0 - The MPEG Source info-frame status has not changed, interrupt not generated 1 - The MPEG Source info-frame status has changed, and an interrupt was generated	
RX_SPD_INFO_FRM_EDGE_ST			R
0x1AF9	000000<u>0</u>0	This bit is used to indicate the latched status of the SPD info-frame detect interrupt. 0 - The SPD info-frame status has not changed, interrupt not generated 1 - The SPD info-frame status has changed, and an interrupt was generated	
RX_AVI_INFO_FRM_EDGE_ST			R
0x1AF9	0000000<u>0</u>	This bit is used to indicate the latched status of the AVI info-frame detect interrupt. 0 - The AVI info-frame status has not changed, interrupt not generated 1 - The AVI info-frame status has changed, and an interrupt was generated	

Reg	Bits	Description
RX_VS_INF_CKS_ERR_EDGE_RAW		R
0x1AFA	00<u>0</u>00000	This readback indicates the raw status of the VS info-frame check sum error. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No change in the VS info-frame checksum error raw status 1 - There was an error in the VS info-frame checksum .
RX_MS_INF_CKS_ERR_EDGE_RAW		R
0x1AFA	000<u>0</u>0000	This readback indicates the raw status of the MS info-frame check sum error. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No change in the MS info-frame checksum error raw status 1 - There was an error in the MS info-frame checksum .
RX_SPD_INF_CKS_ERR_EDGE_RAW		R
0x1AFA	0000<u>0</u>000	This readback indicates the raw status of the SPD info-frame check sum error. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No change in the SPD info-frame checksum error raw status 1 - There was an error in the SPD info-frame checksum .
RX_AVI_INF_CKS_ERR_EDGE_RAW		R
0x1AFA	00000<u>0</u>00	This readback indicates the raw status of the AVI info-frame check sum error. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No change in the AVI info-frame checksum error raw status 1 - There was an error in the AVI info-frame checksum .
RX_DEEPCOLOR_CHNG_EDGE_RAW		R
0x1AFA	000000<u>0</u>0	This readback indicates the raw status of the deep color mode. Once set this bit remains high until cleared via the corresponding clear bit. 0 - The deep color mode has not changed 1 - The deep color mode has changed
RX_TMDS_CLK_CHNG_EDGE_RAW		R
0x1AFA	0000000<u>0</u>	This readback indicates the raw status of the TMDS clock detected signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No change in TMDS clock detected 1 - A change in the TMDS clock has been detected
RX_PKT_ERR_EDGE_RAW		R
0x1AFB	<u>0</u>0000000	This readback indicates the raw status of the packet error signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No uncorrectable error detected in a packet header 1 - An uncorrectable error was detected in an unknown packet (packet header)
RX_GAMUT_MDATA_PCKT_EDGE_RAW		R
0x1AFB	0<u>0</u>000000	This readback indicates the raw status of the Gamut Meta Data packet received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No Gamut Meta-data packet received in the last video frame or since last HDMI packet detection reset. 1 - The Gamut Meta-data packet received in the last video frame or since last HDMI packet detection reset.
RX_ISRC2_PCKT_EDGE_RAW		R
0x1AFB	00<u>0</u>00000	This readback indicates the raw status of the ISRC2 packet received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new ISRC2 packet received 1 - ISRC2 packet with new content received
RX_ISRC1_PCKT_EDGE_RAW		R
0x1AFB	000<u>0</u>0000	This readback indicates the raw status of the ISRC1 packet received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new ISRC1 packet received 1 - ISRC1 packet with new content received
RX_VS_INFO_FRM_EDGE_RAW		R
0x1AFB	0000<u>0</u>000	This readback indicates the raw status of the new VS info-frame received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new VS packet received 1 - VS packet with new content received
RX_MS_INFO_FRM_EDGE_RAW		R
0x1AFB	00000<u>0</u>00	This readback indicates the raw status of the MS info-frame received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new MS packet received 1 - MS packet with new content received

Reg	Bits	Description	
RX_SPD_INFO_FRM_EDGE_RAW			R
0x1AFB	0000000<u>0</u>	This readback indicates the raw status of the new SPD info-frame received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new SPD info-frame received 1 - SPD info-frame with new content received	
RX_AVI_INFO_FRM_EDGE_RAW			R
0x1AFB	0000000<u>0</u>	This readback indicates the raw status of the AVI info-frame received signal. Once set this bit remains high until cleared via the corresponding clear bit. 0 - No new AVI info-frame received 1 - AVI info-frame with new content received	
POWER_DOWN			R/W
0x1AFC	<u>0</u>0000001	Software powerdown to mimic the behaviour of hardware powerdown 0 - Software powerdown inactive 1 - Software powerdown active	
READ_AUTO_INC_EN			R/W
0x1AFC	0000000<u>1</u>	This register is used to auto increment I2C addresses in the device for consecutive reads. 0 - No auto increment of I2C address for consecutive reads 1 - Auto increment of I2C address for consecutive reads	
SVSP_RESET			SC
0x1AFD	<u>0</u>0000000	This bit is used to reset the Secondary VSP. 0 - Default 1 - Reset	
PVSP_RESET			SC
0x1AFD	0<u>0</u>000000	This bit is used to reset the Primary VSP. 0 - Default 1 - Reset	
P2I_RESET			SC
0x1AFD	00<u>0</u>00000	This bit is used to reset the Progressive to Interlaced core. 0 - Default 1 - Reset	
DDR2_INTF_RESET			SC
0x1AFD	000<u>0</u>0000	This bit is used to reset the external DDR memory interface core. 0 - Default 1 - Reset	
SPI_RESET			SC
0x1AFD	0000<u>0</u>000	This bit is used to reset the SPI hardware, both master and slave. 0 - Default 1 - Reset	
OSD_RESET			SC
0x1AFD	0000000<u>0</u>	This bit is used to reset the OSD core and the secondary input channel. 0 - Default 1 - Reset	
INP_SDR_RESET			SC
0x1AFD	0000000<u>0</u>	This bit is used to reset the input capture and formatting logic for the primary input channel. 0 - Default 1 - Reset	
RX_RESET			SC
0x1AFE	<u>0</u>0000000	This bit is used to reset the Serial Video RX core and the RX input channel. 0 - Default 1 - Reset	
ENC_RESET			SC
0x1AFE	0<u>0</u>000000	This bit is used to reset the HD and SD encoders. 0 - Default 1 - Reset	

Reg	Bits	Description	
TX2_RESET			SC
0x1AFE	00<u>0</u>00000	This bit is used to reset the HDMI TX2. 0 - Default 1 - Reset	
TX1_RESET			SC
0x1AFE	000<u>0</u>0000	This bit is used to reset the HDMI TX1. 0 - Default 1 - Reset	
DPLL_RESET			SC
0x1AFE	0000<u>0</u>000	This bit is used to reset the DPLL clock generator. 0 - Default 1 - Reset	
HPS_RESET			SC
0x1AFE	00000<u>0</u>00	This bit is used to reset the HPS block 0 - Default 1 - Reset	
XTAL_RESET			SC
0x1AFE	0000000<u>0</u>	This bit is used to reset all the clocks in the device and peripheral logic in the core including some CEC logic, the interrupt generator and the automatic clock selection. 0 - Default 1 - Reset	
ACE_FRAME_ROW_MAN_EN			R/W
0x1B00	<u>0</u>0000000	0 - Auto select 1 - Manual select	
ACE_FRAME_ROW_MAN[10:0]			R/W
0x1B00 0x1B01	00000<u>000</u> <u>00000000</u>		
ACE_FRAME_COL_MAN_EN			R/W
0x1B02	<u>0</u>0000000	0 - Auto select 1 - Manual select	
ACE_FRAME_COL_MAN[10:0]			R/W
0x1B02 0x1B03	00000<u>000</u> <u>00000000</u>		
VID_Y_BA_LVL[7:0]			R/W
0x1B08	<u>00010000</u>	This signal sets the blanking area MSB level for the luma vid input.	
VID_C_BA_LVL[7:0]			R/W
0x1B09	<u>10000000</u>	This signal sets the blanking area MSB level for the color vid input.	
RX_Y_BA_LVL[7:0]			R/W
0x1B0A	<u>00010000</u>	This signal sets the blanking area MSB level for the luma rx input.	
RX_C_BA_LVL[7:0]			R/W
0x1B0B	<u>10000000</u>	This signal sets the blanking area MSB level for the color rx input.	
EXOSD_Y_BA_LVL[7:0]			R/W
0x1B0C	<u>00010000</u>	This signal sets the blanking area MSB level for the luma osd input.	
EXOSD_C_BA_LVL[7:0]			R/W
0x1B0D	<u>10000000</u>	This signal sets the blanking area MSB level for the color osd input.	

Reg	Bits	Description	
VID_CSC_ENABLE			R/W
0x1B30	0000000	This bit is used to control the Primary Input Channel CSC. 0 - CSC disable 1 - CSC enable	
VID_CSC_MODE[1:0]			R/W
0x1B30	00000000	This signal is used to specify the CSC mode for the Primary Input Channel CSC. The CSC mode sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380	
VID_A1[12:0]			R/W
0x1B30 0x1B31	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient A1.	
VID_A2[12:0]			R/W
0x1B32 0x1B33	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient A2.	
VID_A3[12:0]			R/W
0x1B34 0x1B35	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient A3.	
VID_A4[12:0]			R/W
0x1B36 0x1B37	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient A4.	
VID_B1[12:0]			R/W
0x1B38 0x1B39	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient B1.	
VID_B2[12:0]			R/W
0x1B3A 0x1B3B	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient B2.	
VID_B3[12:0]			R/W
0x1B3C 0x1B3D	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient B3.	
VID_B4[12:0]			R/W
0x1B3E 0x1B3F	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient B4.	
VID_C1[12:0]			R/W
0x1B40 0x1B41	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient C1.	
VID_C2[12:0]			R/W
0x1B42 0x1B43	0000000000000	This signal is used to specify the Primary Input Channel CSC coefficient C2.	

Reg	Bits	Description	
VID_C3[12:0]			R/W
0x1B44 0x1B45	00000000 00000000	This signal is used to specify the Primary Input Channel CSC coefficient C3.	
VID_C4[12:0]			R/W
0x1B46 0x1B47	00000000 00000000	This signal is used to specify the Primary Input Channel CSC coefficient C4.	
VID_SWAP_BUS_CTRL[2:0]			R/W
0x1B48	00001000	This signal is used to control the video input pixel bus. The input pixel bus is 36 bits wide and is divided into three data channels: Top = D[35:24], Middle = D[23:12] and Bottom = D[11:0]. This register allows the user to swap the order of these three data channels. 000 - D[35:24] D[23:12] D[11:0] 001 - D[35:24] D[11:0] D[23:12] 010 - D[35:24] D[23:12] D[11:0] 011 - D[23:12] D[35:24] D[11:0] 100 - D[11:0] D[35:24] D[23:12] 101 - D[11:0] D[23:12] D[35:24] 110 - D[23:12] D[11:0] D[35:24] 111 - D[35:24] D[23:12] D[11:0]	
VID_FORMAT_SEL[4:0]			R/W
0x1B48	00001000	This signal is used to select the input format for the video data. 0x00 - 1 x 8-bit bus, SDR 4:2:2 0x01 - 1 x 10-bit bus, SDR 4:2:2 0x02 - 1 x 12-bit bus, SDR 4:2:2 0x03 - 2 x 8-bit buses, SDR 4:2:2 0x04 - 2 x 10-bit buses, SDR 4:2:2 0x05 - 2 x 12-bit buses, SDR 4:2:2 0x06 - 3 x 8-bit buses, SDR 4:4:4 (P[35:28], P[23:16], P[11:4]) 0x07 - 3 x 10-bit buses, SDR 4:4:4 (P[35:26], P[23:14], P[11:2]) 0x08 - 3 x 12-bit buses, SDR 4:4:4 0x09 - 1 x 8-bit bus, DDR 4:2:2 0x0A - 1 x 10-bit bus DDR 4:2:2 0x0B - 1 x 12 bit bus, DDR 4:2:2 0x0C - 3 x 8 bit buses, SDR 4:4:4 (P[23:0]) 0x0D - 2 x 3 x 8-bit interleaved buses, SDR 4:4:4 0x0E - 2 x 2 x 8-bit interleaved buses, SDR 4:2:2 0x0F - 2 x 2 x 10-bit interleaved buses, SDR 4:2:2 0x10 - 2 x 2 x 12-bit interleaved buses, SDR 4:2:2 0x11 - 3 x 10-bit buses, SDR 4:4:4 (P[29:0]) 0x12 - 3 x 7-bit buses, SDR 4:4:4 (for external alpha blend) 0x13 - 3 x 10-bit buses, SDR 4:4:4 (OSD_IN[23:0] and P[35:30])	
VID_SWAP_CB_CR_422			R/W
0x1B49	00100000	This bit is used to swap the order of the C data when decoding 4:2:2 data. 0 - Cb/Cr decoding 1 - Cr/Cb decoding	
VID_PS444_R444_CONV			R/W
0x1B49	00100000	This bit is used to convert 4:2:2 data to pseudo 444 or to real 444. 0 - Nothing done 1 - Pseudo 444 to Real 444 conversion	
VID_BLANK_BLANKING_AREA			R/W
0x1B49	00100000	This bit is used to specify the blanking area that is blanked to avoid the filters mistakenly interpreting data in the blanking area. 1 - Blanking area is blanked. 0 - Blanking area data passes through.	
VID_LSB_KILLED			R/W
0x1B49	00100000	This bit is used to ignore the unused LSB in the case that the part is set in 8 or 10 bits mode (i2c_input_format). 0 - Nothing done to LSB 1 - Unused LSB are ignored	

Reg	Bits	Description
VID_HS_POL		R/W
0x1B49	00100000	This bit is used to set the polarity of the input HS timing signal. 0 - Input HS polarity does not change 1 - Input HS polarity gets inverted
VID_VS_POL		R/W
0x1B49	00100000	This bit is used to set the polarity of the input VS timing signal. 0 - Input VS polarity does not change 1 - Input VS polarity gets inverted
VID_DE_POL		R/W
0x1B49	00100000	This bit is used to set the polarity of the input DE enable signal. 0 - Input DE polarity does not change 1 - Input DE polarity gets inverted
VID_FLD_POL		R/W
0x1B49	00100000	This bit is used to set the polarity of the input Field (FLD) timing signal. 0 - Input FLD polarity does not change 1 - Input FLD polarity gets inverted
VID_DDR_EDGE_SEL		R/W
0x1BA	00000000	This bit is used to select which edge the first sample of DDR data is latched on. 0 - Posedge data first 1 - Negedge data first
VID_UD_BYPASS_MAN_EN		R/W
0x1BA	00000000	This bit is used to enable the manual bypass for the up dither. Setting this bit enables the bypass to be used. 0 - Manual bypass disable 1 - Manual bypass enable
VID_UD_BYPASS_MAN		R/W
0x1BA	00000000	This bit is used to bypass the up dither block. 0 - Disable bypass 1 - Enable bypass
VID_DDR_YC_SWAP		R/W
0x1BA	00000000	This bit is used to swap the Luma (Y) and Chroma (C) data in DDR modes. By default, Y is expected on the rising edge of the clock. 0 - Y on rising edge of clock 1 - C on rising edge of clock
VID_HS_VS_MODE		R/W
0x1BB	10000000	This bit is used to select the method of input timing. 0 - Use embedded SAV/EAV codes 1 - Use external HS/VS synchronization signals
VID_REV_BUS		R/W
0x1BB	10000000	This bit is used to reverse the input video bus, i.e. D[35:0] -> D[0:35]. 0 - Normal operation 1 - Reverse video input bus
VID_AV_POS_SEL		R/W
0x1BB	10000000	This bit is used to select if the HS generated is consistent with EIA 861 timing or dependant on the embedded timing codes. 0 - Generate HS coincident with EAV code 1 - Generate HS/VS based on 861 timing
VID_AV_SPLIT_CODE		R/W
0x1BB	10000000	This bit is used to control how AV codes are decoded - replicated on or split across all channels. 0 - Decodes AV codes which are replicated on all channels 1 - Decodes AV codes which are split across all channels
VID_AV_CODES_REP_MAN_EN		R/W
0x1BB	10000000	This bit is used to control the enable for AV source codes. AV_codes_rep_man is used instead of the auto based on the input video format. 0 - AV codes replicated based on internal flag 1 - Use i2c bit

Reg	Bits	Description
VID_AV_CODES_REP_MAN		R/W
0x1B4B	1000000<u>0</u>	This bit is used to specify if the AV_codes are replicated or not. Codes replicated (4:4:4) = FF,FF,FF,00,00,00,00,00,00,AV,AV,AV. Codes not replicated = FF,00,00,AV. 1 - AV codes are replicated. 0 - AV codes are not replicated.
EXOSD_CSC_ENABLE		R/W
0x1B50	<u>0</u>0000000	This bit is used to enable the Secondary Input Channel CSC. 0 - CSC disable 1 - CSC enable
EXOSD_CSC_MODE[1:0]		R/W
0x1B50	0<u>00</u>00000	This signal is used to specify the CSC mode for the Secondary Input Channel CSC. The CSC mode sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380
EXOSD_A1[12:0]		R/W
0x1B50 0x1B51	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient A1.
EXOSD_A2[12:0]		R/W
0x1B52 0x1B53	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient A2.
EXOSD_A3[12:0]		R/W
0x1B54 0x1B55	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient A3.
EXOSD_A4[12:0]		R/W
0x1B56 0x1B57	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient A4.
EXOSD_B1[12:0]		R/W
0x1B58 0x1B59	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient B1.
EXOSD_B2[12:0]		R/W
0x1B5A 0x1B5B	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient B2.
EXOSD_B3[12:0]		R/W
0x1B5C 0x1B5D	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient B3.
EXOSD_B4[12:0]		R/W
0x1B5E 0x1B5F	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient B4.
EXOSD_C1[12:0]		R/W
0x1B60 0x1B61	000<u>00000</u> <u>00000000</u>	This signal is used to specify the Secondary Input Channel CSC coefficient C1.

Reg	Bits	Description
EXOSD_C2[12:0]		R/W
0x1B62 0x1B63	00000000 00000000	This signal is used to specify the Secondary Input Channel CSC coefficient C2.
EXOSD_C3[12:0]		R/W
0x1B64 0x1B65	00000000 00000000	This signal is used to specify the Secondary Input Channel CSC coefficient C3.
EXOSD_C4[12:0]		R/W
0x1B66 0x1B67	00000000 00000000	This signal is used to specify the Secondary Input Channel CSC coefficient C4.
EXOSD_SWAP_BUS_CTRL[2:0]		R/W
0x1B68	00001100	This signal is used to control the external OSD input pixel bus. The input pixel bus is 24 bits wide and is divided into three data channels: Top = D[23:16], Middle = D[15:8] and Bottom = D[7:0]. This register allows the user to swap the order of these three data channels. 000 - D[23:16] D[15:8] D[7:0] 001 - D[23:16] D[7:0] D[15:8] 010 - D[23:16] D[15:8] D[7:0] 011 - D[15:8] D[23:16] D[7:0] 100 - D[7:0] D[23:16] D[15:8] 101 - D[7:0] D[15:8] D[23:16] 110 - D[15:8] D[7:0] D[23:16] 111 - D[23:16] D[15:8] D[7:0]
EXOSD_FORMAT_SEL[4:0]		R/W
0x1B68	00001100	This signal is used to select the input format for the video data. 0x00 - 1 x 8 bit bus 4:2:2 0x01 - 1 x 10 bit bus 4:2:2 0x02 - 1 x 12 bit bus 4:2:2 0x03 - 2 x 8 bit buses 4:2:2 0x04 - 2 x 10 bit buses 4:2:2 0x05 - 2 x 12 bit buses 4:2:2 0x06 - 3 x 8-bit buses, SDR 4:4:4 0x07 - 3 x 10-bit buses, SDR 4:4:4 0x08 - 3 x 12-bit buses, SDR 4:4:4 0x09 - 1 x 8 bit DDR bus 4:2:2 0x0A - 1 x 10 bit DDR bus 4:2:2 0x0B - 1 x 12 bit DDR bus 4:2:2 0x0C - 3 x 8 bit buses 4:4:4
EXOSD_SWAP_CB_CR_422		R/W
0x1B69	00100000	This bit is used to swap the order of the C data when decoding 4:2:2 data. 0 - Cb/Cr decoding 1 - Cr/Cb decoding
EXOSD_PS444_R444_CONV		R/W
0x1B69	00100000	This bit is used to convert 4:2:2 data to pseudo 444 or to real 444. 0 - Nothing done. 1 - Pseudo444 to Real 444 conversion.
EXOSD_BLANK_BLANKING_AREA		R/W
0x1B69	00100000	This bit is used to specify the blanking area that is blanked to avoid the filters mistakenly interpreting data in the blanking area. 1 - Blanking area is blanked. 0 - Blanking area data goes through.
EXOSD_LSB_KILLED		R/W
0x1B69	00100000	This bit is used to ignore the unused LSB in the case that the part is set in 8 or 10 bits mode (i2c_input_format). 0 - Nothing done to LSB 1 - Unused LSB are ignored

Reg	Bits	Description	
EXOSD_FLD_POL			R/W
0x1B69	00100000	This bit is used to set the polarity of the input External OSD FLD timing signal. 0 - Input FLD polarity doesn't change. 1 - Input FLD polarity gets inverted.	
EXOSD_DE_POL			R/W
0x1B69	00100000	This bit is used to set the polarity of the input External OSD DE timing signal. 0 - Input DE polarity doesn't change. 1 - Input DE polarity gets inverted.	
EXOSD_VS_POL			R/W
0x1B69	00100000	This bit is used to set the polarity of the input External OSD VS timing signal. 0 - Input VS polarity doesn't change. 1 - Input VS polarity gets inverted.	
EXOSD_HS_POL			R/W
0x1B69	00100000	This bit is used to set the polarity of the input External OSD HS timing signal. 0 - Input HS polarity doesn't change. 1 - Input HS polarity gets inverted.	
EXOSD_DDR_EDGE_SEL			R/W
0x1B6A	00000000	This bit is used to select which edge the first sample of DDR data is latched on. 0 - Posedge data first 1 - Negedge data first	
EXOSD_UD_BYPASS_MAN_EN			R/W
0x1B6A	00000000	This bit is used to enable the manual bypass for the up dither. Setting this bit enables the bypass to be used. 0 - Manual bypass disable 1 - Manual bypass enable	
EXOSD_UD_BYPASS_MAN			R/W
0x1B6A	00000000	This bit is used to bypass the up dither block. 0 - Disable bypass 1 - Enable bypass	
EXOSD_DDR_YC_SWAP			R/W
0x1B6A	00000000	This bit is used to swap the Luma (Y) and Chroma (C) data in DDR modes. By default, Y is expected on the rising edge of the clock. 0 - Y on rising edge of clock 1 - C on rising edge of clock	
EXOSD_HS_VS_MODE			R/W
0x1B6B	10000000	This bit is used to select the method of input timing. 0 - Embedded timing codes 1 - VS/DE mode	
EXOSD_2X_1X_USE_LL_MAN			R/W
0x1B6B	10000000	This bit is used to control the enable for the lockup latch for the 2x to 1x transfer. This allows for safe transfer of data to a 1x clock when the input clock is in pixel repetition mode. 0 - Do a transfer from 2x to 1x clock 1 - Use lockup latches to transfer the data	
EXOSD_2X_1X_USE_LL_MAN_EN			R/W
0x1B6B	10000000	This bit is used to control the enable for the lockup latch for the 2x to 1x transfer. 0 - Lockup latch used based on internal flag 1 - Use i2c bit	
EXOSD_REV_BUS			R/W
0x1B6B	10000000	This bit is used to reverse the input video bus, i.e. D[23:0] -> D[0:23]. 0 - Reverse the pin mapping on the OSD bus 1 - Use the OSD bus as it comes from the pins	
EXOSD_AV_POS_SEL			R/W
0x1B6B	10000000	This bit is used to select if the HS generated is consistent with EIA 861 timing or dependant on the embedded timing codes. 0 - Generate hs coincident with eav code 1 - Generate hs/vs based on 861 timing	

Reg	Bits	Description	
EXOSD_AV_SPLIT_CODE			R/W
0x1B6B	10000000	This bit is used to control how AV codes are decoded - replicated on or split across all channels. 0 - Replicated av codes on all channels 1 - AV codes split across all buses	
EXOSD_AV_CODES_REP_MAN_EN			R/W
0x1B6B	10000000	This bit is used to control the enable for AV source codes. AV_codes_rep_man is used instead of the auto based on the input video format. 0 - AV codes replicated based on internal flag 1 - Use i2c bit	
EXOSD_AV_CODES_REP_MAN			R/W
0x1B6B	10000000	This bit is used to specify if the AV_codes are replicated or not. Codes replicated (4:4:4) = FF,FF,FF,00,00,00,00,00, 00,AV,AV,AV. Codes not replicated = FF,00,00,AV. 1 - AV codes are replicated. 0 - AV codes are not replicated.	
EXOSD_IN_ID[7:0]			R/W
0x1B6C	11111110	This register is used to specify the video_id relative to CEA 861. 0x01 - CEA 861 VIC 1 (480p_60 640) 0x02 - CEA 861 VIC 2 (480p_60) 0x03 - CEA 861 VIC 3 (480p_60) 0x04 - CEA 861 VIC 4 (720p_60) 0x05 - CEA 861 VIC 5 (1080i_60) 0x06 - CEA 861 VIC 6 (480i_60) 0x07 - CEA 861 VIC 7 (480i_60) 0x08 - CEA 861 VIC 8 (240p_60) 0x09 - CEA 861 VIC 9 (240p_60) 0x10 - CEA 861 VIC 16 (1080p_60) 0x11 - CEA 861 VIC 17 (576p_50) 0x12 - CEA 861 VIC 18 (576p_50) 0x13 - CEA 861 VIC 19 (720p_50) 0x14 - CEA 861 VIC 20 (1080i_50) 0x15 - CEA 861 VIC 21 (576i_50) 0x16 - CEA 861 VIC 22 (576i_50) 0x17 - CEA 861 VIC 23 (288p_50) 0x18 - CEA 861 VIC 24 (288p_50) 0x1F - CEA 861 VIC 31 (1080p_50) 0xFC - CEA 861 VIC 252 (288p_50) 0xFD - CEA 861 VIC 253 (240p_60) 0xFE - CEA 861 VIC 254 (480i_60) 0xFF - CEA 861 VIC 255 (576i_50)	
EXOSD_ALPHA_FMT[3:0]			R/W
0x1B6D	00001001	This signal is used to select the inputs used to receive the external OSD alpha value. 0x0 - External alpha input on P[3:0] and P[15:12] 0x1 - External alpha input on P[3:0] and P[27:24] 0x2 - External alpha input on P[15:12] and P[3:0] 0x3 - External alpha input on P[15:12] and P[27:24] 0x4 - External alpha input on P[27:24] and P[3:0] 0x5 - External alpha input on P[27:24] and P[15:12] 0x6 - External alpha input on P[35:28] 0x7 - External alpha input on P[23:16] 0x8 - External alpha input on P[11:4] 0x9 - External alpha input on OSD_IN[16], OSD_IN[8], OSD_IN[0] and 00000 0xA - External alpha input on OSD_IN[13:12], OSD_IN[1:0] and 0000 0xB - External alpha input on OSD_IN[15:12] and OSD_IN[3:0] 0xC - External alpha input on OSD_IN[23:16] 0xD - External alpha input on OSD_IN[15:8] 0xE - External alpha input on OSD_IN[7:0]	
RX_CSC_ENABLE			R/W
0x1B70	00000000	This bit is used to enable the RX input channel CSC. 0 - CSC disable 1 - CSC enable	

Reg	Bits	Description	
RX_CSC_MODE[1:0]			R/W
0x1B70	00000000	This signal is used to specify the CSC mode for the RX input channel CSC. The CSC mode sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380	
RX_A1[12:0]			R/W
0x1B70 0x1B71	0000000000000000	This signal is used to specify the RX input channel CSC coefficient A1.	
RX_A2[12:0]			R/W
0x1B72 0x1B73	0000000000000000	This signal is used to specify the RX input channel CSC coefficient A2.	
RX_A3[12:0]			R/W
0x1B74 0x1B75	0000000000000000	This signal is used to specify the RX input channel CSC coefficient A3.	
RX_A4[12:0]			R/W
0x1B76 0x1B77	0000000000000000	This signal is used to specify the RX input channel CSC coefficient A4.	
RX_B1[12:0]			R/W
0x1B78 0x1B79	0000000000000000	This signal is used to specify the RX input channel CSC coefficient B1.	
RX_B2[12:0]			R/W
0x1B7A 0x1B7B	0000000000000000	This signal is used to specify the RX input channel CSC coefficient B2.	
RX_B3[12:0]			R/W
0x1B7C 0x1B7D	0000000000000000	This signal is used to specify the RX input channel CSC coefficient B3.	
RX_B4[12:0]			R/W
0x1B7E 0x1B7F	0000000000000000	This signal is used to specify the RX input channel CSC coefficient B4.	
RX_C1[12:0]			R/W
0x1B80 0x1B81	0000000000000000	This signal is used to specify the RX input channel CSC coefficient C1.	
RX_C2[12:0]			R/W
0x1B82 0x1B83	0000000000000000	This signal is used to specify the RX input channel CSC coefficient C2.	

Reg	Bits	Description	
RX_C3[12:0]			R/W
0x1B84 0x1B85	00000000 00000000	This signal is used to specify the RX input channel CSC coefficient C3.	
RX_C4[12:0]			R/W
0x1B86 0x1B87	00000000 00000000	This signal is used to specify the RX input channel CSC coefficient C4.	
RX_SWAP_BUS_CTRL[2:0]			R/W
0x1B88	00001000	This signal is used to configure the order of the input video bus. 000 - D[35:24] D[23:12] D[11:0] 001 - D[35:24] D[11:0] D[23:12] 010 - D[35:24] D[23:12] D[11:0] 011 - D[23:12] D[35:24] D[11:0] 100 - D[11:0] D[35:24] D[23:12] 101 - D[11:0] D[23:12] D[35:24] 110 - D[23:12] D[11:0] D[35:24] 111 - D[35:24] D[23:12] D[11:0]	
RX_FORMAT_SEL[4:0]			R/W
0x1B88	00001000	This signal is used to select the input format for the RX input data. 0x06 - 3 x 8 bit buses 4:4:4 0x07 - 3 x 10 bit buses 4:4:4 0x08 - 3 x 12 bit buses 4:4:4	
RX_BLANK_BLANKING_AREA			R/W
0x1B89	00000000	This bit is used to specify the blanking area that is blanked to avoid the filters mistakenly interpreting data in the blanking area. 1 - Blanking area is blanked. 0 - Blanking area data passes through.	
RX_SWAP_CB_CR_422			R/W
0x1B89	00000000	This bit is used to swap the order of the C data when decoding 4:2:2 data. 0 - Cb/Cr decoding 1 - Cr/Cb decoding	
RX_PS444_R444_CONV			R/W
0x1B89	00000000	This bit is used to convert 4:2:2 data to pseudo 444 or to real 444. 0 - Nothing done 1 - Pseudo 444 to Real 444 conversion	
RX_LSB_KILLED			R/W
0x1B89	00000000	This bit is used to ignore the unused LSB in the case that the part is set in 8 or 10 bits mode (i2c_input_format). 0 - Nothing done to LSB 1 - Unused LSB are ignored	
RX_FLD_POL			R/W
0x1B89	00000000	This bit is used to set the polarity of the RX FLD timing signal. 0 - Input FLD polarity doesn't change. 1 - Input FLD polarity gets inverted.	
RX_DE_POL			R/W
0x1B89	00000000	This bit is used to set the polarity of the RX DE timing signal. 0 - Input DE polarity doesn't change. 1 - Input DE polarity gets inverted.	
RX_VS_POL			R/W
0x1B89	00000000	This bit is used to set the polarity of the RX VS timing signal. 0 - Input VS polarity doesn't change. 1 - Input VS polarity gets inverted.	
RX_HS_POL			R/W
0x1B89	00000000	This bit is used to set the polarity of the RX HS timing signal. 0 - Input HS polarity doesn't change. 1 - Input HS polarity gets inverted.	

Reg	Bits	Description
RX_UD_BYPASS_MAN_EN		R/W
0x1B8A	00000000	This bit is used to enable the manual bypass for the up dither. Setting this bit enables the bypass to be used. 0 - Manual bypass disable 1 - Manual bypass enable
RX_UD_BYPASS_MAN		R/W
0x1B8A	00000000	This bit is used to bypass the up dither block. 0 - Disable bypass 1 - Enable bypass
DE_H_BEG_POS[11]		R/W
0x1B8B	00000000	This signal is used to select the DE horizontal beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX
RX_2X_1X_USE_LL_MAN		R/W
0x1B8B	00000000	This bit is used to control the enable for the lockup latch for the 2x to 1x transfer. This allows for safe transfer of data to a 1x clock when the input clock is in pixel repetition mode. 0 - Do a transfer from 2x to 1x clock 1 - Use lockup latches to transfer the data
RX_2X_1X_USE_LL_MAN_EN		R/W
0x1B8B	00000000	This bit is used to control the enable for the lockup latch for the 2x to 1x transfer. 0 - Lockup latch used based on internal flag 1 - Use i2c bit
DE_H_BEG_POS[10]		R/W
0x1B8B	00000000	This signal is used to select the DE horizontal beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX
HS_BEG_POS[11:10]		R/W
0x1B8B	00000000	This signal is used to select the HS beginning position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX
HS_END_POS[11:10]		R/W
0x1B8B	00000000	This signal is used to select the HS ending position, counting from the EAV, if CEA 861 timing generation is enabled and manual values selected. 0xXX - assert de when hcount reaches 0xXX
DE_V_BEG_E_POS[6:0]		R/W
0x1B8C	10110110	This signal is used to specify the DE vertical beginning position for even fields, if CEA 861 timing generation is enable and manual values selected. 0xXX - assert de when lcount reaches 0xXX on even fields
DE_V_BEG_O_POS[6:0]		R/W
0x1B8C 0x1B8D	10110110 11010000	This signal is used to specify the DE vertical beginning position for odd fields, if CEA 861 timing generation is enable and manual values selected. 0xXX - assert de when lcount reaches 0xXX on even fields
DE_H_BEG_POS[9:0]		R/W
0x1B8D 0x1B8E	11010000 10100000	This signal is used to specify the DE horizontal beginning position, counting from the EAV, if CEA 861 timing generation is enable and manual values selected. 0xXX - assert de when hcount reaches 0xXX
HS_BEG_POS[9:0]		R/W
0x1B8F 0x1B90	00000001 11000000	This signal is used to specify the HS beginning position, counting from the EAV, if CEA 861 timing generation is enable and manual values selected. 0xXX - assert hs when hcount reaches 0xXX
HS_END_POS[9:0]		R/W
0x1B90 0x1B91	11000000 00000000	This signal is used to specify the HS ending position, counting from the EAV, if CEA 861 timing generation is enable and manual values selected. 0xXX - release hs when hcount reaches 0xXX

Reg	Bits	Description
VS_H_BEG_O_POS[10:0]		R/W
0x1B91 0x1B92	00000000 00010000	This signal is used to specify the horizontal beginning position of VS for odd fields (counting from the EAV), if CEA 861 timing generation is enable and manual values selected. 0xXX - assert vs when hcount reaches 0xXX on odd fields
VS_H_BEG_E_POS[10:0]		R/W
0x1B93 0x1B94	00001110 00000000	This signal is used to specify the horizontal beginning position of VS for even fields (counting from the EAV), if CEA 861 timing generation is enable and manual values selected. 0xXX - assert vs when hcount reaches 0xXX on even fields
FLD_CAP_POS		R/W
0x1B94	00000000	This signal is used to select the capture position of the fld signal. The captured fld signal will be used to generate the right VS timing when in interlace format. 0x0 - fld value is captured when de_in goes low 0x1 - fld value is capture in middle of active line.
VS_V_BEG_POS[5:0]		R/W
0x1B94 0x1B95	00000000 01001100	This signal is used to specify the vertical beginning position of VS, if CEA 861 timing generation is enable and manual values selected. 0xXX - assert vs when lcount reaches 0xXX
VS_V_END_POS[5:0]		R/W
0x1B95	01001100	This signal is used to specify the vertical ending position of VS, if CEA 861 timing generation is enable and manual values selected. 0xXX - release vs when lcount reaches 0xXX
RX_IN_ID[7:0]		R/W
0x1B96	00000000	This register is used to specify the VIC relative to CEA 861. 0x06 - CEA861 VIC 6 (480i60 2x) 0x07 - CEA861 VIC 7 (480i60 2x) 0x08 - CEA861 VIC 8 (240p60 2x) 0x09 - CEA861 VIC 9 (240p60 2x) 0x15 - CEA861 VIC 21 (576i50 2x) 0x16 - CEA861 VIC 22 (576i50 2x) 0x17 - CEA861 VIC 23 (288p50 2x) 0x18 - CEA861 VIC 24 (288p50 2x)
MP2I_FRTRK_MAS_FLD		R/W
0x1B97	10000100	This bit select whether the input field information from the mas_vs and mas_hs is tracked by the mp2i block or not. The control signal pvsp_frtrk_mas_mode_en, must also be enabled for this bit to take effect. 0 - Disable tracking of input master field 1 - Enable tracking of input master field
PVSP_FRTRK_MAS_MODE_EN		R/W
0x1B97	10000100	This bit enables the use of external master hs and vs for frame tracking 0 - Frame track input 1 - Frame track external master hs/vs
SP2I_FRTRK_MAS_FLD		R/W
0x1B99	10000100	This bit selects whether the input field information from the mas_vs and mas_hs is tracked by the mp2i block or not. The control signal svsp_frtrk_mas_mode_en, must also be enabled for this bit to take effect. 0 - Disable tracking of input master field 1 - Enable tracking of input master field
SVSP_FRTRK_MAS_MODE_EN		R/W
0x1B99	10000100	This bit enables the use of external master hs and vs for frame tracking 0 - Frame track input 1 - Frame track external master hs/vs
PVSP_MAS_CLK_IN_EN		R/W
0x1B9B	00000000	This bit enables the use of an external master clock for frame tracking mode 0 - Don't use input master clock from pin 1 - Use input master clock from pin
SVSP_MAS_CLK_IN_EN		R/W
0x1B9B	00000000	Activates the use of an external master clock for frame tracking mode 0 - Don't use input master clock from pin 1 - Use input master clock from pin

Reg	Bits	Description	
OSD_DOUT_DRV_STR[1:0]			R/W
0x1BA3	000000<u>00</u>	This signal is used to control the drive strength for the video output data and sync signals. 00 - Minimum 01 - Medium low (x2) 10 - Medium high (x3) 11 - Maximum (x4)	
OSD_CLK_DRV_STR[1:0]			R/W
0x1BA7	000000<u>00</u>	This signal is used to control the drive strength for the video output clock signal. 00 - Minimum 01 - Medium low (x2) 10 - Medium high (x3) 11 - Maximum (x4)	
CRC_RX_EN			R/W
0x1BA8	<u>0</u>0000000	This bit enables or disables the CRC module	
CRC_RST			R/W
0x1BA8	0<u>0</u>000000	This bit resets the CRC module	
CRC_INTERLACED			R/W
0x1BA8	00<u>0</u>00000	This bit is used to set whether the format is interleaved or not.	
CRC_RB_SEL[2:0]			R/W
0x1BA8	00000<u>000</u>	The crc algoritm generates 5 checksums of 16bits each. This bit selects which one to read.	
CRC_NUM_FRAMES[7:0]			R/W
0x1BA9	<u>00000000</u>	This signal indicates the number of frames to use for the CRC selection	
RB_RX_CRC_CHECKSUM[15:0]			R
0x1BAA 0x1BAB	<u>00000000</u> <u>00000000</u>	This signal provides a readback of the CRC calculation	
RB_RX_CRC_READY			R
0x1BAC	0000000<u>0</u>	This is a readback of the CRC ready bit.	
TTL_OUT_CSC_ENABLE			R/W
0x1BB0	<u>0</u>0000000	This bit is used to enable the ttl output channel CSC. 0 - CSC disable 1 - CSC enable	
TTL_OUT_CSC_MODE[1:0]			R/W
0x1BB0	0<u>00</u>00000	This signal is used to specify the CSC mode for the ttl output channel CSC. The CSC mode sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380	
TTL_OUT_A1[12:0]			R/W
0x1BB0 0x1BB1	000<u>00000</u> <u>00000000</u>	This signal is used to specify the ttl out channel CSC coefficient A1.	
TTL_OUT_A2[12:0]			R/W
0x1BB2 0x1BB3	000<u>00000</u> <u>00000000</u>	This signal is used to specify the ttl out channel CSC coefficient A2.	

Reg	Bits	Description	
TTL_OUT_A3[12:0]			R/W
0x1BB4 0x1BB5	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient A3.	
TTL_OUT_A4[12:0]			R/W
0x1BB6 0x1BB7	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient A4.	
TTL_OUT_B1[12:0]			R/W
0x1BB8 0x1BB9	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient B1.	
TTL_OUT_B2[12:0]			R/W
0x1BBA 0x1BBB	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient B2.	
TTL_OUT_B3[12:0]			R/W
0x1BBC 0x1BBD	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient B3.	
TTL_OUT_B4[12:0]			R/W
0x1BBE 0x1BBF	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient B4.	
TTL_OUT_C1[12:0]			R/W
0x1BC0 0x1BC1	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient C1.	
TTL_OUT_C2[12:0]			R/W
0x1BC2 0x1BC3	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient C2.	
TTL_OUT_C3[12:0]			R/W
0x1BC4 0x1BC5	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient C3.	
TTL_OUT_C4[12:0]			R/W
0x1BC6 0x1BC7	00000 00000000	This signal is used to specify the ttl out channel CSC coefficient C4.	
VID_CLK_IE			R/W
0x1BC8	0	This bit is used to control the input path enable for the VID CLK pin. 0 - input path disable 1 - input path enable	
CLK_OSD_IE			R/W
0x1BC8	0	This bit is used to control the input path enable for the osd clk pin. 0 - input path disable 1 - input path enable	

Reg	Bits	Description	
PIX_PINS_IE[35:32]			R/W
0x1BC8	0000<u>0000</u>	This bit is used to control the input path enable for the pixel pins. 0 - input path disable 1 - input path enable	
PIX_PINS_IE[31:0]			R/W
0x1BC9 0x1BCA 0x1BCB 0x1BCC	<u>00000000</u> <u>00000000</u> <u>00000000</u> <u>00000000</u>	This bit is used to control the input path enable for the pixel pins. 0 - input path disable 1 - input path enable	
OSD_PINS_IE[23:0]			R/W
0x1BCD 0x1BCE 0x1BCF	<u>00000000</u> <u>00000000</u> <u>00000000</u>	This bit is used to control the input path enable for the osd pins. 0 - input path disable 1 - input path enable	
HS_IE			R/W
0x1BD0	<u>0</u>0000000	This bit is used to control the input path enable for the HS pin. 0 - input path disable 1 - input path enable	
VS_IE			R/W
0x1BD0	0<u>0</u>000000	This bit is used to control the input path enable for the VS pin. 0 - input path disable 1 - input path enable	
DE_IE			R/W
0x1BD0	00<u>0</u>00000	This bit is used to control the input path enable for the DE pin. 0 - input path disable 1 - input path enable	
SFL_IE			R/W
0x1BD0	000<u>0</u>0000	This bit is used to control the input path enable for the SFL pin. 0 - input path disable 1 - input path enable	
HS_OSD_IE			R/W
0x1BD0	00000<u>0</u>00	This bit is used to control the input path enable for the osd HS pin. 0 - input path disable 1 - input path enable	
VS_OSD_IE			R/W
0x1BD0	000000<u>0</u>0	This bit is used to control the input path enable for the osd VS pin. 0 - input path disable 1 - input path enable	
DE_OSD_IE			R/W
0x1BD0	0000000<u>0</u>	This bit is used to control the input path enable for the osd DE pin. 0 - input path disable 1 - input path enable	
AUDIO_PINS_IE[6:0]			R/W
0x1BD1	0<u>0000000</u>	This bit is used to control the input path enable for the audio pins. 0 - input path disable 1 - input path enable	
ARC1_PIN_IE			R/W
0x1BD2	<u>0</u>0000000	This bit is used to control the input path enable for the ARC 1 pin. 0 - input path disable 1 - input path enable	

Reg	Bits	Description	
ARC2_PIN_IE			R/W
0x1BD2	0000000	This bit is used to control the input path enable for the ARC 2 pin. 0 - input path disable 1 - input path enable	
INT_PIN_IE[2:0]			R/W
0x1BD2	000000	This bit is used to control the input path enable for the INT pins. 0 - input path disable 1 - input path enable	
SCLK_IE			R/W
0x1BD2	00000	This bit is used to control the input path enable for the audio SCLK pin. 0 - input path disable 1 - input path enable	
MCLK_IE			R/W
0x1BD2	0000000	This bit is used to control the input path enable for the audio MCLK pin. 0 - input path disable 1 - input path enable	
DSD_CLK_IE			R/W
0x1BD2	0000000	This bit is used to control the input path enable for the audio DSD CLK pin. 0 - input path disable 1 - input path enable	
SPI1_CS_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi1 CS pin. 0 - input path disable 1 - input path enable	
SPI1_MISO_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi1 MISO pin. 0 - input path disable 1 - input path enable	
SPI1_MOSI_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi1 MOSI pin. 0 - input path disable 1 - input path enable	
SPI1_SCLK_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi1 SCLK pin. 0 - input path disable 1 - input path enable	
SPI2_CS_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi1 CS pin. 0 - input path disable 1 - input path enable	
SPI2_MISO_IE			R/W
0x1BD3	00000000	This bit is used to control the input path enable for the spi2 ,OSP pin. 0 - input path disable 1 - input path enable	
SPI2_MOSI_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi2 MOSI pin. 0 - input path disable 1 - input path enable	
SPI2_SCLK_IE			R/W
0x1BD3	0000000	This bit is used to control the input path enable for the spi2 SCLK pin. 0 - input path disable 1 - input path enable	
MAS_CLK_IE			R/W
0x1BD4	00000000	This bit is used to control the input path enable for the master CLK pin. 0 - input path disable 1 - input path enable	

Reg	Bits	Description
MAS_HS_IE		R/W
0x1BD4	0000000	This bit is used to control the input path enable for the master HS pin. 0 - input path disable 1 - input path enable
MAS_VS_IE		R/W
0x1BD4	0000000	This bit is used to control the input path enable for the master VS pin. 0 - input path disable 1 - input path enable
AUTO_PHPO_INP_SEL[1:0]		R/W
0x1BE0	00100000	This control signal is used to select which input is routed to the auto position and auto phase blocks 00 - VID TTL 01 - OSD TTL 10 - RX 11 - N/A
AUTO_PHPO_BYP_CSC		R/W
0x1BE0	00100000	This bit is used to bypass the CSC or not before routing to the auto Phase and auto Position detection blocks 0 - CSC output used for auto PHPO 1 - CSC input used for auto PHPO
AUTO_PH_EN		R/W
0x1BE1	0000000	This bit is used to enable auto phase detection block 0 - Disabled 1 - Enabled
AUTO_PH_NUM[6:0]		R/W
0x1BE1	0000000	This control signal sets the total number of phases available on the front end part, e.g. 8, 16, 32, etc
AUTO_PH_SCAN[5:0]		R/W
0x1BE2	000000	This control signal sets the scan phase number being tested. When the scan value changes, a new scan is triggered to start.
RB_AUTO_PH_READ_READY		R
0x1BE3	0000000	This bit is used to indicate rb_auto_ph_diff_sum_lock is valid, a HIGH means it is valid to read the value in auto_ph_diff_sum_lock.
RB_AUTO_PH_RIGHT_PHASE[5:0]		R
0x1BE3	000000	This signal is used to indicate the correct phase after i2c_auto_ph_scan has been indexed through all of the phases.
RB_AUTO_PH_DIFF_SUM_LOCK[23:0]		R
0x1BE4 0x1BE5 0x1BE6	00000000 00000000 00000000	This signal is used to indicate the statistical result for the phase in AUTO_PH_SCAN (Auto Phase Scan Number). This signal is valid when AUTO_PH_READ_READY is HIGH.
AUTO_PO_EN		R/W
0x1BE7	0000000	This bit is used to enable the auto position detection block 0 - Disabled 1 - Enabled
AUTO_PO_NOISE_THR[9:0]		R/W
0x1BE7 0x1BE8	000000 00000000	This signal sets the noise threshold (minimum value) for the sum of the three channels R, G and B to differentiate the active pixels from the blank pixels. For example, if blank value for RGB is 16, the noise threshold should be larger than 48.
RB_AUTO_PO_L_EDG_LOCK_FLAG		R
0x1BE9	0000000	This bit indicates if the algorithm has locked to the left edge of the input video. If this bit is high, it has locked to the left edge, a low indicates it has not locked to it.
RB_AUTO_PO_R_EDG_LOCK_FLAG		R
0x1BE9	0000000	This bit indicates if the algorithm has locked to the right edge of the input video. If this bit is high, it has locked to the right edge, a low indicates it has not locked to it.

Reg	Bits	Description	
RB_AUTO_PO_T_OFFSET[15:0]			R
0x1BEA 0x1BEB	00000000 00000000	This readback signal returns the top offset, the number of blank lines before the start of active video. This offset excludes the vertical blanking area.	
RB_AUTO_PO_B_OFFSET[15:0]			R
0x1BEC 0x1BED	00000000 00000000	This readback signal returns the bottom offset, the number of blank lines after active video. This offset excludes the vertical blanking area.	
RB_AUTO_PO_L_OFFSET[15:0]			R
0x1BEE 0x1BEF	00000000 00000000	This readback signal returns the left offset, the number of blank Pixels before the start of active video. This offset excludes the horizontal blanking area.	
RB_AUTO_PO_R_OFFSET[15:0]			R
0x1BF0 0x1BF1	00000000 00000000	This readback signal returns the right offset, the number of blank Pixels after active video. This offset excludes the horizontal blanking area.	
MAIN_RESET			SC
0x1BFF	00000000	This bit is used to initiate a global reset for the device. 0 - Default 1 - Reset	

2.2 PRIMARY VSP MAP

Reg	Bits	Description	
PVSP_FIELDBUFFER0_ADDR[31:0]			R/W
0xE800 0xE801 0xE802 0xE803	00000000 01010110 00100010 00000000	This signal is used to set the start address of field/frame buffer 0. Software should arrange memory space properly, avoiding conflict between different buffers. 0x004F1A00 - Default 0xXXXXXXXX - Start address of field/frame buffer 0	
PVSP_FIELDBUFFER1_ADDR[31:0]			R/W
0xE804 0xE805 0xE806 0xE807	00000000 11100010 11000010 00000000	This signal is used to set the start address of field/frame buffer 1. Software should arrange memory space properly, avoiding conflict between different buffers. 0x00CDAA00 - Default 0xXXXXXXXX - Start address of field/frame buffer 1	
PVSP_FIELDBUFFER2_ADDR[31:0]			R/W
0xE808 0xE809 0xE80A 0xE80B	00000001 01101111 01100010 00000000	This signal is used to set the start address of field/frame buffer 2. Software should arrange memory space properly, avoiding conflict between different buffers. 0x014C3A00 - Default 0xXXXXXXXX - Start address of field/frame buffer 2	
PVSP_FIELDBUFFER3_ADDR[31:0]			R/W
0xE80C 0xE80D 0xE80E 0xE80F	00000001 11111100 00000010 00000000	This signal is used to set the start address of field/frame buffer 3. Software should arrange memory space properly, avoiding conflict between different buffers. 0x01CACA00 - Default 0xXXXXXXXX - Start address of field/frame buffer 3	
PVSP_FIELDBUFFER4_ADDR[31:0]			R/W
0xE810 0xE811 0xE812 0xE813	00000010 10001000 10100010 00000000	This signal is used to set the start address of field/frame buffer 4. Software should arrange memory space properly, avoiding conflict between different buffers. 0x02495A00 - Default 0xXXXXXXXX - Start address of field/frame buffer 4	
PVSP_FIELDBUFFER5_ADDR[31:0]			R/W
0xE814 0xE815 0xE816 0xE817	00000011 00010101 01000010 00000000	This signal is used to set the start address of field/frame buffer 5. Software should arrange memory space properly, avoiding conflict between different buffers. 0x02C7EA00 - Default 0xXXXXXXXX - Start address of field/frame buffer 5	

Reg	Bits	Description
PVSP_MOTIONBUF0_ADDR[31:0]		R/W
0xE818 0xE819 0xE81A 0xE81B	00000000 00000000 00000000 00000000	This signal is used to set the start address of motion information buffer 0. Motion buffers are needed only when motion adaptive deinterlacing is enabled for interlaced input. 0x00000000 - Default 0xXXXXXXXX - Start address of motion buffer 0
PVSP_MOTIONBUF1_ADDR[31:0]		R/W
0xE81C 0xE81D 0xE81E 0xE81F	00000000 00000111 11101001 00000000	This signal is used to set the start address of motion information buffer 1. Motion buffers are needed only when motion adaptive deinterlacing is enabled for interlaced input. 0x0007E900 - Default 0xXXXXXXXX - Start address of motion buffer 1
PVSP_RNR_BUF0_ADDR[31:0]		R/W
0xE820 0xE821 0xE822 0xE823	00000000 00001111 11010010 00000000	This signal is used to set the start address of random noise reduction information buffer 0. RNR buffers are needed only when random noise reduction is enabled. 0x000FD200 - Default 0xXXXXXXXX - Start address of RNR buffer 0
PVSP_RNR_BUF1_ADDR[31:0]		R/W
0xE824 0xE825 0xE826 0xE827	00000000 00110010 11111010 00000000	This signal is used to set the start address of random noise reduction information buffer 1. RNR buffers are needed only when random noise reduction is enabled. 0x002F7600 - Default 0xXXXXXXXX - Start address of RNR buffer 1
PVSP_IS_I_TO_P		R/W
0xE828	000**0**10000	This bit is used to set the input video format. If the input video is interlaced, this bit should be set to 1. If the input video is progressive, it should be set to 0. This register's value will be used while pvsp_autocfg_input_vid is 0. 0 - Input video is progressive 1 - Input video is interlaced
PVSP_UPDATE_VOM		R/W
0xE828	000**1**10000	This bit is used to control the updating of the VOM. Registers in the VOM can be updated only when pvsp_update_vom is asserted. To modify registers in the VOM, pvsp_update_vom should be de-asserted. The registers can then be modified. pvsp_update_vom should then be asserted to let the VOM use the updated register value in the next frame. This procedure will guarantee the correctness of the VOM configuration. 0 - Do not update VOM 1 - Update VOM
PVSP_LOCK_VOM		R/W
0xE828	0001**0**000	This bit is used to lock the Video Output Module (VOM). If the Primary VSP is running and this bit is set to 1, the VOM will be locked to a current register setting to display the last frame. The Primary VSP registers can be configured safely in this state. All new register settings will be updated after this bit is set back to 0. 0 - Unlock VOM 1 - Lock VOM
PVSP_ENABLE_VOM		R/W
0xE828	00010**0**00	This bit is used to control the Video Output Module (VOM). If this bit is set to 1, the VOM is enabled to read video data from external memory, process it and then output it. 0 - Disable VOM 1 - Enable VOM

Reg	Bits	Description
PVSP_ENABLE_VIM		R/W
0xE828	0001000<u>0</u>	This bit is used to control the Video Input Module (VIM). If this bit is set to 1, the VIM is enabled to write packed input video data into a defined external field/frame buffer. While the Primary VSP is running, if this bit is set to 0, the output video stream will be frozen. 0 - Disable VIM 1 - Enable VIM
PVSP_ENABLE_FFS		R/W
0xE828	0001000<u>0</u>	This bit is used to control the Field Frame Scheduler (FFS). If this bit is set to 1, the FFS is enabled and the VIM and VOM are scheduled by the FFS, which means the Primary VSP is in operating mode. If this bit is set to 0, the Primary VSP is in idle mode. 0 - Disable FFS/FRC 1 - Enable FFS/FRC
PVSP_BYPASS		R/W
0xE829	<u>0</u>0000000	This bit is used to bypass the Primary VSP. If this bit is set to 1, the input video to the Primary VSP will be directly bypassed to the output port. 0 - Not bypass Primary VSP 1 - Bypass Primary VSP
PVSP_EX_MEM_DATA_FORMAT[1:0]		R/W
0xE829	000<u>00</u>000	This signal is used to set the data format in external memory. 00 - YCbCr-12b-10b-10b 01 - YCbCr-8b-8b-8b 10 - YCbCr-4:2:4-12b 11 - YCbCr-4:2:2-8b
PVSP_VIN_H[10:0]		R/W
0xE82E 0xE82F	00000<u>000</u> <u>00000000</u>	This signal is used to set the horizontal resolution of the input video. This register's value will be used while pvsp_man_input_res is 1 or pvsp_autocfg_input_vid is 0. 0x000 - Default 0xXXX - Horizontal resolution of input video
PVSP_VIN_V[10:0]		R/W
0xE830 0xE831	00000<u>000</u> <u>00000000</u>	This signal is used to set the vertical resolution of the input video. This register's value will be used while pvsp_man_input_res is 1 or pvsp_autocfg_input_vid is 0. 0x000 - Default 0xXXX - Vertical resolution of input video
PVSP_VIM_CROP_H_START[10:0]		R/W
0xE832 0xE833	00000<u>000</u> <u>00000000</u>	This signal is used to set the horizontal start position of the VIM cropper. 0x000 - Default 0xXXX - Horizontal start position of VIM cropper input
PVSP_VIM_CROP_V_START[10:0]		R/W
0xE834 0xE835	00000<u>000</u> <u>00000000</u>	This signal is used to set the vertical start position of the VIM cropper. 0x000 - Default 0xXXX - Vertical start position of VIM cropper input
PVSP_VIM_CROP_WIDTH[10:0]		R/W
0xE836 0xE837	00000<u>000</u> <u>00000000</u>	This signal is used to set the input width of the VIM cropper. 0x000 - Default 0xXXX - Width of VIM cropper input
PVSP_VIM_CROP_HEIGHT[10:0]		R/W
0xE838 0xE839	00000<u>000</u> <u>00000000</u>	This signal is used to set the input height of the VIM cropper. 0x000 - Default 0xXXX - Height of VIM cropper input
PVSP_VIM_D_SCAL_OUT_WIDTH[10:0]		R/W
0xE83A 0xE83B	00000<u>000</u> <u>00000000</u>	This signal is used to set the output video width of the down-scaling scaler in the VIM. The input video width is set by register pvsp_vim_crop_width. If VIM crop is not enabled, pvsp_vim_crop_width is auto configured by pvsp_autocfg_input_vid, which is the same with input video's horizontal resolution. 0x000 - Default 0xXXX - Output width of VIM scalar

Reg	Bits	Description	
PVSP_DI_CROP_H_START[10:0]			R/W
0xE83C 0xE83D	00000000 00000000	This signal is used to set the horizontal start position of the VOM cropper. 0x000 - Default 0xXXX - Horizontal start position of VOM cropper input	
PVSP_DI_CROP_V_START[10:0]			R/W
0xE83E 0xE83F	00000000 00000000	This signal is used to set the vertical start position of the VOM cropper. 0x000 - Default 0xXXX - Vertical start position of VOM cropper input	
PVSP_DI_CROP_WIDTH[10:0]			R/W
0xE840 0xE841	00000000 00000000	This signal is used to set the width of the VOM cropper. 0x000 - Default 0xXXX - Width of VOM cropper input	
PVSP_DI_CROP_HEIGHT[10:0]			R/W
0xE842 0xE843	00000000 00000000	This signal is used to set the height of the VOM cropper. 0x000 - Default 0xXXX - Height of VOM cropper input	
PVSP_SCAL_OUT_WIDTH[12:0]			R/W
0xE844 0xE845	00000000 00000000	This signal is used to set the output horizontal resolution of scaler in the VOM. 0x000 - Default 0xXXX - Output width of VOM scaler	
PVSP_SCAL_OUT_HEIGHT[12:0]			R/W
0xE846 0xE847	00000000 00000000	This signal is used to set the output vertical resolution of scaler in the VOM. 0x000 - Default 0xXXX - Output height of VOM scaler	
PVSP_DP_VIDEO_H_START[12:0]			R/W
0xE848 0xE849	00000000 00000000	This signal is used to set the horizontal start position where the output video of the scaler is placed. 0x000 - Default 0xXXX - Horizontal start position of VOM output	
PVSP_DP_VIDEO_V_START[12:0]			R/W
0xE84A 0xE84B	00000000 00000000	This signal is used to set the vertical start position where the output video of scaler is placed. 0x000 - Default 0xXXX - Vertical start position of VOM output	
DI_SHARPNESS_ENABLE			R/W
0xE84C	00001110	This bit is used to enable sharpness control. 0 - Disable sharpness 1 - Enable sharpness	
DI_BNR_ENABLE			R/W
0xE84C	00001110	This bit is used to enable block noise reduction (BNR). 0 - Disable BNR 1 - Enable BNR	
DI_MNR_ENABLE			R/W
0xE84C	00001110	This bit is used to enable mosquito noise reduction (MNR). 0 - Disable MNR 1 - Enable MNR	
DI_RNR_ENABLE			R/W
0xE84C	00001110	This bit is used to enable random noise reduction (RNR). 0 - Disable RNR 1 - Enable RNR	

Reg	Bits	Description
DI_ULAI_ENABLE		R/W
0xE84C	00001110	This bit is used to enable the ultra low angle de-interlacing algorithm (ULAI). 0 - Disable ULAI 1 - Enable ULAI
DI_CADENCE_ENABLE		R/W
0xE84C	00001110	This bit is used to enable cadence detection. 0 - Disable cadence detection 1 - Enable cadence detection
DI_INTRA_FIELD_ENABLE		R/W
0xE84C	00001110	This bit is used to enable intra-field interpolation for interlaced inputs. For progressive input, this register should be set to zero. If this bit is asserted, motion adaptive deinterlacing, RNR, MNR, BNR, cadence detection, chroma processing, CUE correction will all be disabled automatically. This bit should be asserted in low latency mode. 0 - Disable intra field interpolation 1 - Enable intra field interpolation
PVSP_BYPASS_DDR_MODE		R/W
0xE84D	00011011	This bit is used to enable game mode for the Primary VSP. 0 - Normal mode 1 - Game mode
PVSP_FRC_LOW_LATENCY_MODE		R/W
0xE84D	00011011	This bit is used to enable low latency mode. 0 - Disable low latency mode 1 - Enable low latency mode
PCADENCE_ENABLE		R/W
0xE84D	00011011	This bit is used to enable progressive cadence detection. 0 - Disable progressive cadence detection 1 - Enable progressive cadence detection
DI_CUE_ENABLE		R/W
0xE84D	00011011	This bit is used to enable CUE correction. 0 - Disable CUE correction 1 - Enable CUE correction
PVSP_FRC_CHANGE_PHASE_EN		R/W
0xE84E	00010001	This bit is used to lock the phase change for cadence detection. 0 - Disable 1 - Enable
PVSP_DATA_CLIPPING_EN		R/W
0xE84E	00010001	This bit is used to limit the output data within range of 16~235. 0 - Not limit output data. 1 - Limit output data
PVSP_DI_OUT_TO_SVSP_EN		R/W
0xE84E	00010001	This bit is used to enable the output of the deinterlacer to SVSP. 0 - Disable deinterlacer output to SVSP 1 - Enable deinterlacer output to SVSP
DI_MNR_LEVEL[1:0]		R/W
0xE84F	00001010	This signal sets the MNR level. 00 - N/A 01 - Low 10 - Middle 11 - High
DI_RNR_LEVEL[1:0]		R/W
0xE84F	00001010	This signal sets the RNR level. 00 - N/A 01 - Low 10 - Middle 11 - High
M_SCALER_PANORAMA_EN		R/W
0xE850	00000000	This bit enables panorama scaling for the VOM scaler. 0 - Disable VOM panorama 1 - Enable VOM panorama

Reg	Bits	Description
M_SCALER_PANORAMA_POS[11:0]		R/W
0xE851 0xE852	0000<u>0000</u> <u>00000000</u>	This signal is used to define the width of the output video frame which is not stretched when panorama mode is enabled but, rather, is scaled properly. The maximum value of this register is set by: pvsp_di_crop_width * (pvsp_scal_out_width/pvsp_di_crop_height) - pvsp_scal_out_width/2. This register sets half the width of the output frame which is to be scaled normally. By default, this register is set to 0 which means that all the input frame will be stretched. It is, therefore, recommended that this register is set by the user before enabling the panorama function. 0x000 - Default 0xXXX - Width of not-stretched image
PVSP_DP_DECOUNT[12:0]		R/W
0xE856 0xE857	000<u>00000</u> <u>00000000</u>	This signal is used to set the DE duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Data enable count of output timing
PVSP_DP_HFRONTPORCH[11:0]		R/W
0xE858 0xE859	0000<u>0000</u> <u>00000000</u>	This signal is used to set the horizontal front porch duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Horizontal front porch of output timing
PVSP_DP_HSYNCTIME[11:0]		R/W
0xE85A 0xE85B	0000<u>0000</u> <u>00000000</u>	This signal sets the Hsync duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Hsync width of output timing
PVSP_DP_HBACKPORCH[11:0]		R/W
0xE85C 0xE85D	0000<u>0000</u> <u>00000000</u>	This signal is used to set the horizontal back porch duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Horizontal back porch of output timing
PVSP_DP_ACTIVELINE[11:0]		R/W
0xE85E 0xE85F	0000<u>0000</u> <u>00000000</u>	This signal is used to set the active line number of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Active lines of output timing
PVSP_DP_VFRONTPORCH[9:0]		R/W
0xE860 0xE861	000000<u>00</u> <u>00000000</u>	This signal is used to set the vertical front porch duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vertical front porch of output timing
PVSP_DP_VSYNCTIME[9:0]		R/W
0xE862 0xE863	000000<u>00</u> <u>00000000</u>	This signal is used to set the vertical synchronous time duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vsync width of output timing
PVSP_DP_VBACKPORCH[9:0]		R/W
0xE864 0xE865	000000<u>00</u> <u>00000000</u>	This signal is used to set the vertical back porch duration of output timing. This register's value will be used while pvsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vertical back porch of output timing
PVSP_DP_MARGIN_COLOR[23:0]		R/W
0xE866 0xE867 0xE868	<u>00000000</u> <u>10000000</u> <u>10000000</u>	This signal is used to set the default color in output video in YUV colorspace. 0x000000 - Default 0xXXXXXX - Default color in YUV colorspace
PVSP_DP_4KX2K_MODE_EN		R/W
0xE869	000<u>0</u>0000	This bit is used to make the VOM display module work in 4K x 2K mode. This register's value will be used while pvsp_autocfg_output_vid is 0. 0 - Not in 4K x 2K mode 1 - In 4K x 2K mode

Reg	Bits	Description
PVSP_DP_OUTPUT_BLANK		R/W
0xE869	0000000	This bit is used to force the colour output of the Primary VSP. This If this bit is set to 1, the output of Primary VSP is forced to the user defined color in pvsp_dp_margin_color. 0 - Not output default color 1 - Output default Color
PVSP_DP_HPOLARITY		R/W
0xE869	000000	This bit is used to set the polarity of output Hsync. This register's value will be used while pvsp_autocfg_output_vid is 0. 0 - Low 1 - High
PVSP_DP_VPOLARITY		R/W
0xE869	00000000	This bit is used to set the polarity of output Vsync. This register's value will be used while pvsp_autocfg_output_vid is 0. 0 - Low 1 - High
PVSP_DP_SUBID[1:0]		R/W
0xE86C	00000000	This signal is used while pvsp_autocfg_output_vid is 8/9/12/13/23/24/27/28. If pvsp_autocfg_output_vid is 8 or 9 or 12 or 13, output timing vfrontporch is 4 while pvsp_dp_subid is 0, output timing vfrontporch is 5 while pvsp_dp_subid is 1. If pvsp_autocfg_output_vid is 23 or 24 or 27 or 28, output timing vfrontporch is 2 while pvsp_dp_subid is 0, output timing vfrontporch is 3 while pvsp_dp_subid is1.
PVSP_VIN_FR[7:0]		R/W
0xE86E	00111100	This register is used to specify the input video frame rate. This register's value will be used while pvsp_autocfg_input_vid is 0.
PVSP_VOUT_FR[7:0]		R/W
0xE86F	00111100	This register is used to specify the output video frame rate. This register's value will be used while pvsp_autocfg_output_vid is 0.
PVSP_RB_FRAME_LATENCY[2:0]		R
0xE870	00000000	This signal is used to indicate the real time vsync latency. 0xXXX - number of frame latency
PVSP_RB_HSYNC_LATENCY[11:0]		R
0xE875 0xE876	00000000 00000000	This signal is used to indicate the real time Hsync latency. 0xXXX - number of hsync latency
PVSP_DEMO_WINDOW_ENABLE		R/W
0xE87E	00100000	Enables demo window. 0 - Disable demo window 1 - Enable demo window
PVSP_DEMO_WINDOW_USE_LOWER_SCREEN		R/W
0xE87E	00100000	This bit is used to enable a demo mode on the lower half of the screen. If this bit is set to 1, the lower half display window will be used for certain processing functions, otherwise the upper half display window will be used. 0 - Use upper half screen as demo window 1 - Use lower half screen as demo window
PVSP_DEMO_WINDOW_RNR_ENABLE		R/W
0xE87E	00100000	This bit is used to enable the RNR in the demo window. 0 - Disable RNR in demo window 1 - Enable RNR in demo window
PVSP_DEMO_WINDOW_MNR_ENABLE		R/W
0xE87E	00100000	This bit is used to enable the MNR in the demo window. 0 - Disable MNR in demo window 1 - Enable MNR in demo window
PVSP_DEMO_WINDOW_BNR_ENABLE		R/W
0xE87E	00100000	This bit is used to enable the BNR in the demo window. 0 - Disable BNR in demo window 1 - Enable BNR in demo window
PVSP_DEMO_WINDOW_CADENCE_ENABLE		R/W
0xE87E	00100000	This bit is used to enable the cadence detection in the demo window. 0 - Disable Cadence detection in demo window 1 - Enable Cadence detection in demo window

Reg	Bits	Description	
PVSP_DEMO_WINDOW_ULAI_ENABLE			R/W
0xE87E	0010000<u>0</u>	This bit is used to enable the ULAI in the demo window. 0 - Disable ULAI in demo window 1 - Enable ULAI in demo window	
PVSP_DEMO_WINDOW_CUE_ENABLE			R/W
0xE87F	00<u>0</u>00000	This bit is used to enable CUE correction in the demo window. 0 - Disable CUE in demo window 1 - Enable CUE in demo window'	
PVSP_DEMO_WINDOW_INTRA_FIELD_ENABLE			R/W
0xE87F	000<u>0</u>0000	This bit is used to enable the intra field interpolation in the demo window. 0 - Disable intra field interpolation in demo window 1 - Enable intra field interpolation in demo window	
PVSP_AUTOCFG_INPUT_VID[7:0]			R/W
0xE881	<u>00000110</u>	This register is used to set the input timing VIC. If this register is 0, PVSP will use values in registers of pvsp_vin_h, pvsp_vin_v, pvsp_is_i_to_p and pvsp_vin_fr to set input video. 0x06 - Default: 480i@60 0xXX - Input timing VID	
PVSP_AUTOCFG_OUTPUT_VID[7:0]			R/W
0xE882	<u>00010000</u>	This register is used to set the output timing VIC. If this register is 0, PVSP will use values in registers of pvsp_dp_decount, pvsp_dp_hfrontporch, pvsp_dp_hsynctime, pvsp_dp_hbackporch, pvsp_dp_activeline, pvsp_dp_vfrontporch, pvsp_dp_vsynctime, pvsp_dp_vbackporch, pvsp_dp_hpolarity, pvsp_dp_vpolarity, pvsp_vout_fr and pvsp_dp_4kx2k_mode_en to set output video. 0x10 - Default: 1080p@60 0xXX - Output timing VID	
PVSP_VIM_CROP_ENABLE			R/W
0xE883	1<u>0</u>000000	This bit is used to enable the VIM crop. 0 - Disable VIM Crop 1 - Enable VIM Crop	
PVSP_VIM_D_SCAL_ENABLE			R/W
0xE883	10<u>0</u>00000	This bit is used to enable the VIM down scaler. 0 - Disable VIM down scaler 1 - Enable VIM down scaler	
PVSP_DI_CROP_ENABLE			R/W
0xE883	100<u>0</u>0000	This bit is used to enable the VOM crop. 0 - Disable VOM Crop 1 - Enable VOM Crop	
PVSP_MAN_SCAL_OUT_ENABLE			R/W
0xE883	1000<u>0</u>000	This bit is used to enable the manual setting of pvsp_scal_out_width and pvsp_scal_out_height. 0 - Disable manually setting M_Scaler output resolution 1 - Enable manually setting M_Scaler output resolution	
PVSP_MAN_DP_TIMING_ENABLE			R/W
0xE883	1000000<u>0</u>	This bit is used to enable the manual setting of the display port's timing. 0 - Disable manually setting output timing 1 - Enable manually setting output timing	
PVSP_MAN_INPUT_RES			R/W
0xE884	00<u>0</u>00000	This bit is used to enable the manual configuration of the input resolution. 0 - Disable manual configuration of input resolution 1 - Enable manual configuration of input resolution	
DI_LOWPOWER_EN			R/W
0xE884	000<u>0</u>0000	This bit is used to set the deinterlacer into low power mode. This bit should be set when output timing is 4K x 2K. 0 - Deinterlacer not in low power mode 1 - Deinterlacer in low power mode	

Reg	Bits	Description	
PVSP_FIELDBUFFER6_ADDR[31:0]			R/W
0xE889 0xE88A 0xE88B 0xE88C	00000011 10100001 11100010 00000000	This signal is used to set the start address of field/frame buffer 6. Software should arrange memory space properly, avoiding conflict between different buffers. 0x03073200 - Default 0xXXXXXXXX - Start address of field/frame buffer 6	
PVSP_SRSCAL_DEMO_MODE_EN			R/W
0xE890	0000**0**000	This bit is used to enable scaler demo mode. 0 - Scaler not in demo mode 1 - Scaler in demo mode	
PVSP_SRSCAL_8BIT_EN			R/W
0xE890	0000**0**000	This bit is used to set the scaler into 8-bit mode. This bit should be set when output 4K x 2K timing. 0 - Scaler not in 8 bit mode 1 - Scaler in 8 bit mode	
PVSP_SRSCAL_DOWNSCALING_BLUR			R/W
0xE890	0000000**0**	This bit is used to control down-scaling blur. This bit should be configured as 1 when down-scaling. 0 - Disable down scaling blur 1 - Enable down scaling blur	
PVSP_SRSCAL_SCALE_GAIN[11:0]			R/W
0xE891 0xE892	00000000 0000**0000**	This signal is used to control the sharpness level. 0x000 - Sharpness level	
PVSP_SRSCAL_INTERP_MODE[1:0]			R/W
0xE894	**00**000000	This signal is used to select the scaler algorithm employed. 00 - Automatic scaler algorithm selection 01 - Contour-based interpolation scaler (2nd gen scaling algorithm with 4k x 2k support) 10 - Frequency-adaptive scaler (1st gen scaling algorithm) 11 - Bilinear scaler	
PVSP_MAS_RESYNC_EN			R/W
0xE8A1	**0**0100000	This bit enables direct timing generation reset via external sync for the PVSP. This is for modes 2 and 3 only.	
PVSP_VIM_SCAL_TYPE[1:0]			R/W
0xE8E5	**11**100000	This signal is used to set the VIM scaling algorithm. For up-scaling, the proprietary ADI algorithm is recommended; whereas for down-scaling, the sharp setting is recommended. 00 - Proprietary ADI Algorithm 01 - Sharp 10 - Smooth 11 - Bilinear	
PVSP_VIM_SCAL_ANTI_ALIASING_EN			R/W
0xE8E5	111**1**0000	This bit is used to improve the performance of the down-scaling. By setting this bit to 1, high frequency aliasing can be reduced to weaken the moiré effect. 0 - Disable 1 - Enable	
PVSP_VIM_SCAL_SEPARATED_ALPHA_EN			R/W
0xE8E5	1110**0**000	This bit is used to control the interpolation parameter selection for the VIM. This bit should be set to 1 for RGB input and set to 0 for YCbCr input. 0 - YCbCr input 1 - RGB input	
PVSP_VIM_SCAL_OVERSHOOT_CTRL[11:0]			R/W
0xE8E9 0xE8EA	00001000 0000**0000**	This bit is used to control the overshoot in the scaling of input video. If set to a value larger than the default setting, more overshoot is allowed. 0x080 - Default	

Reg	Bits	Description	
PVSP_FRC_LATENCY_MEASURE_EN			R/W
0xE8F0	00001111	This bit is used to enable frame latency measuring. The results are recorded in pvsp_rb_max_latency and pvsp_rb_min_latency. 0 - Disable frame latency measuring 1 - Enable frame latency measuring	
PVSP_RB_MAX_LATENCY[14:0]			R
0xE8F2 0xE8F3	00000000 0000000	This signal is used to record the maximum frame latency. 0xXXX - Maximum of frame latency	
PVSP_RB_MIN_LATENCY[14:0]			R
0xE8F4 0xE8F5	00000000 0000000	This is signal is used to record the minimum frame latency. 0xXXX - Minimum of frame latency	
DI_FD_DISABLED_CADENCE[10:0]			R/W
0xE8FA 0xE8FB	00000000 00000000	This signal is used to disable corresponding cadence detection. 0x000 - Default	

2.3 PRIMARY VSP 2 MAP

Reg	Bits	Description	
DI_MNR_TH_MIN[3:0]			R/W
0xE917	00101010	This signal is used to set the strength of the mosquito noise reduction (MNR). The larger the value, the stronger the MNR noise reduction. 0010 - Normal strength MNR 0110 - High strength MNR	
DI_FD_SCALER_PREDICT[3:0]			R/W
0xE935	11000110	This signal is used to set the locking threshold for cadence detection. Smaller values mean that cadence detection will be locked very easily. 0100 - 480i/576i to 1080p24 conversion 1100 - All other conversions	
DI_FD_SC_TH[7:0]			R/W
0xE93B	01000000	This register is used to control the threshold for scene change detection.	
DI_FD_SUBC_TH[7:0]			R/W
0xE949	11110000	This register is used to set the threshold for subtitle detection.	
PVSP_SRSCAL_FILTER_PARAM2[4:0]			R/W
0xE97C	00100000	This signal is used to set the anti-aliasing filter configuration for down-scaling. 00000 - Low level (for 1080p to 720p or 720p to 480p) 00011 - Mid level 00100 - High level (all other downscaling conversions) 00101 - 11111 - Reserved	
DI_BNR_DETECT_SCALE_LINE[3:0]			R/W
0xE987	01111000	This signal is used to configure the BNR processing ability. 0111 - Recommended setting for low/mid level BNR 1001 - Recommended setting for high level BNR	
DI_BNR_DISABLE_LOCAL_DETECT			R/W
0xE987	01111000	This signal is used to configure the BNR processing ability. 0 - Recommended setting for high level BNR 1 - Recommended setting for low/mid level BNR	
DI_BNR_GLOBAL_STRENGTH_GAIN[3:0]			R/W
0xE988	10000000	This signal is used to configure the BNR processing ability. 1000 - Recommended setting for low/mid level BNR 1100 - Recommended setting for high level BNR	
DI_BNR_SCALE_GLOBAL_VERT[2:0]			R/W
0xE98B	10110100	This signal is used to configure the BNR processing ability. 0101 - Recommended setting for low/mid level BNR 0110 - Recommended setting for high level BNR	
DI_BNR_SCALE_GLOBAL_HORI[2:0]			R/W
0xE98B	10110100	This signal is used to configure the BNR processing ability. 0101 - Recommended setting for low/mid level BNR 0110 - Recommended setting for high level BNR	
DI_BNR_EDGE_OFFSET[7:0]			R/W
0xE98D	01000000	This signal is used to configure the BNR processing ability. 0x32 - Recommended setting for low level BNR 0x64 - Recommended value for mid level BNR 0x96 - Recommended value for high level BNR	

2.4 SECONDARY VSP MAP

Reg	Bits	Description
SVSP_FIELDBUFFER0_ADDR[31:0]		R/W
0xE600 0xE601 0xE602 0xE603	00000100 00101110 10000010 00000000	This signal is used to set the start address of frame buffer 0. Software should arrange memory space properly, avoiding conflict between different buffers. 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 0
SVSP_FIELDBUFFER1_ADDR[31:0]		R/W
0xE604 0xE605 0xE606 0xE607	00000100 10001101 01101110 00000000	This signal is used to set the start address of frame buffer 1. Software should arrange memory space properly, avoiding conflict between different buffers. 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 1
SVSP_FIELDBUFFER2_ADDR[31:0]		R/W
0xE608 0xE609 0xE60A 0xE60B	00000100 11101100 01011010 00000000	This signal is used to set the start address of frame buffer 2. Software should arrange memory space properly, avoiding conflict between different buffers. 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 2
SVSP_FIELDBUFFER3_ADDR[31:0]		R/W
0xE60C 0xE60D 0xE60E 0xE60F	00000101 01001011 01000110 00000000	This signal is used to set the start address of frame buffer 3. Software should arrange memory space properly, avoiding conflict between different buffers. 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 3
SVSP_ENABLE_FFS		R/W
0xE610	0000000	This bit is used to control the Field Frame Scheduler (FFS). If this bit is set to 1, the FFS is enabled and the VIM and VOM are scheduled by the FFS, which means the Secondary VSP is in work mode. If this bit is set to 0, the Secondary VSP is in idle mode. 0 - Disable FFS/FRC 1 - Enable FFS/FRC
SVSP_ENABLE_VIM		R/W
0xE610	0000000	This bit is used to control the Video Input Module (VIM). If this bit is set to 1, the VIM is enabled to write packed input video data into the defined external frame buffer. While the Secondary VSP is running, if this bit is set to 0, the output video stream will be frozen. 0 - Disable VIM 1 - Enable VIM
SVSP_ENABLE_VOM		R/W
0xE610	0000000	This bit is used to control the Video Output Module (VOM). If this bit is set to 1, the VOM is enabled to read video data from external memory, process it and then output it. 0 - Disable VOM 1 - Enable VOM
SVSP_LOCK_VOM		R/W
0xE610	0000000	'This bit is used to lock the Video Output Module (VOM). If the Secondary VSP is running and this bit is set to 1, the VOM will be locked to the current register setting to display the last frame. The Secondary VSP registers can be configured safely in this state. All new register settings will be updated after this bit is set back to 0. 0 - Unlock VOM 1 - Lock VOM

Reg	Bits	Description	
SVSP_UPDATE_VOM			R/W
0xE610	00000000	Registers related to the VOM can be updated only when this bit is set to 0. All new register settings will be updated by VOM in next frame after this bit is set back to 1. 0 - Do not update VOM 1 - Update VOM	
SVSP_FIELDBUF_NUM[2:0]			R/W
0xE610	00000000	This signal is used to set the number of field/frame buffers. This signal needs to be configured while svsp_osd_mode_en is 1. 000 - Default XXX - Number of field/frame buffers	
SVSP_EX_MEM_DATA_FORMAT[1:0]			R/W
0xE611	00001000	This signal is used to set the data format in external memory. 01 - YCbCr-8b-8b-8b 11 - YCbCr-4:2:2-8b	
SVSP_OSD_MODE_EN			R/W
0xE611	00001000	This bit is used to enable the SVSP VIM external OSD mode. 0 - VIM works in normal mode 1 - VIM works in EXOSD mode	
SVSP_OSD_EXALPHA_EN			R/W
0xE611	00001000	This bit is used to enable the use of external alpha. 0 - Not use external alpha 1 - Use external alpha	
SVSP_INPUT_FROM_PVSP_DI_OUT			R/W
0xE611	00001000	This bit is used to select an input from the PVSP's deinterlacer's output. 0 - Input not from PVSP's deinterlacer 1 - Input from PVSP's deinterlacer	
SVSP_VIN_H[12:0]			R/W
0xE616 0xE617	00000000 00000000	This signal is used to set the horizontal resolution of the input video. This register's value will be used while svsp_man_input_res is 1 or svsp_autocfg_input_vid is 1. 0x000 - Default 0xXXX - Horizontal resolution of input video	
SVSP_VIN_V[12:0]			R/W
0xE618 0xE619	00000000 00000000	This signal is used to set the vertical resolution of the input video. This register's value will be used while svsp_man_input_res is 1 or svsp_autocfg_input_vid is 1. 0x000 - Default 0xXXX - Vertical resolution of input video	
SVSP_VIM_CROP_H_START[12:0]			R/W
0xE61A 0xE61B	00000000 00000000	Sets the horizontal start position of the VIM cropper. 0x000 - Default 0xXXX - Horizontal start position of VIM cropper input	
SVSP_VIM_CROP_V_START[12:0]			R/W
0xE61C 0xE61D	00000000 00000000	This signal is used to set the horizontal start position of the VIM cropper. 0x000 - Default 0xXXX - Vertical start position of VIM cropper input	
SVSP_VIM_CROP_WIDTH[12:0]			R/W
0xE61E 0xE61F	00000000 00000000	This signal is used to set the input width of the VIM cropper. 0x000 - Default 0xXXX - Width of VIM cropper input	
SVSP_VIM_CROP_HEIGHT[12:0]			R/W
0xE620 0xE621	00000000 00000000	This signal is used to set the input height of the VIM cropper. 0x000 - Default 0xXXX - Height of VIM cropper input	

Reg	Bits	Description	
SVSP_VIM_SCAL_OUT_WIDTH[10:0]			R/W
0xE622 0xE623	00000000 00000000	This signal is used to set the output horizontal resolution of scaler in the VIM. 0x000 - Default 0xXXX - Output width of VIM scaler	
SVSP_VIM_SCAL_OUT_HEIGHT[10:0]			R/W
0xE624 0xE625	00000000 00000000	This signal is used to set the output vertical resolution of scaler in the VIM. 0x000 - Default 0xXXX - Output height of VIM scaler	
SVSP_VOM_CROP_H_START[10:0]			R/W
0xE626 0xE627	00000000 00000000	This signal is used to set the horizontal start position of the VOM cropper. 0x000 - Default 0xXXX - Horizontal start position of VOM cropper	
SVSP_VOM_CROP_V_START[10:0]			R/W
0xE628 0xE629	00000000 00000000	This signal is used to set the vertical start position of the VOM cropper. 0x000 - Default 0xXXX - Vertical start position of VOM cropper	
SVSP_VOM_CROP_WIDTH[10:0]			R/W
0xE62A 0xE62B	00000000 00000000	This signal is used to set the width of the VOM cropper. 0x000 - Default 0xXXX - Width of VOM cropper input	
SVSP_VOM_CROP_HEIGHT[10:0]			R/W
0xE62C 0xE62D	00000000 00000000	This signal is used to set the height of the VOM cropper. 0x000 - Default 0xXXX - Height of VOM cropper input	
SVSP_DP_VIDEO_H_START[10:0]			R/W
0xE62E 0xE62F	00000000 00000000	This signal is used to set the horizontal start position where the output video of scaler is placed. 0x000 - Default 0xXXX - Horizontal start position of output port	
SVSP_DP_VIDEO_V_START[10:0]			R/W
0xE630 0xE631	00000000 00000000	This signal is used to set the vertical start position where the output video of scaler is placed. 0x000 - Default 0xXXX - Vertical start position of output port	
SVSP_DP_DECOUNT[10:0]			R/W
0xE632 0xE633	00000000 00000000	This signal is used to set the DE duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Data enable count of output timing	
SVSP_DP_HFRONTPORCH[11:0]			R/W
0xE634 0xE635	00000000 00000000	This signal is used to set the horizontal front porch duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Horizontal front porch of output timing	
SVSP_DP_HSYNCTIME[9:0]			R/W
0xE636 0xE637	00000000 00000000	This signal is used to set the Hsync duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Hsync width of output timing	
SVSP_DP_HBACKPORCH[9:0]			R/W
0xE638 0xE639	00000000 00000000	This signal is used to set the horizontal back porch duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Horizontal back porch of output timing	

Reg	Bits	Description
SVSP_DP_ACTIVELINE[10:0]		R/W
0xE63A 0xE63B	00000000 00000000	This signal is used to set the active line number of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Active lines of output timing
SVSP_DP_VFRONTPORCH[9:0]		R/W
0xE63C 0xE63D	00000000 00000000	This signal is used to set the vertical front porch duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vertical front porch of output timing
SVSP_DP_VSYNCTIME[9:0]		R/W
0xE63E 0xE63F	00000000 00000000	This signal is used to set the vertical synchronous time of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vsync width of output timing
SVSP_DP_VBACKPORCH[9:0]		R/W
0xE640 0xE641	00000000 00000000	This signal is used to set the vertical back porch duration of output timing. This register's value will be used while svsp_autocfg_output_vid is 0. 0x000 - Default 0xXXX - Vertical back porch of output timing
SVSP_DP_VPOLARITY		R/W
0xE642	00000000	This signal is used to set the polarity of output Vsync. This register's value will be used while svsp_autocfg_output_vid is 0. 0 - Low 1 - High
SVSP_DP_HPOLARITY		R/W
0xE642	00000000	This signal is used to set the polarity of output Hsync. This register's value will be used while svsp_autocfg_output_vid is 0. 0 - Low 1 - High
SVSP_DP_OUTPUT_BLANK		R/W
0xE642	00000000	'This bit is used to force the colour output of the Secondary VSP. If this register is set to 1, the output of the Secondary VSP is forced to the used defined color in svsp_dp_margin_color. 0 - Not Output default Color 1 - Output default Color
SVSP_DATA_CLIPPING_EN		R/W
0xE642	00000000	This bit is used to limit the output data within range of 16~235. 0 - Not limit output data range 1 - Limit output data range to 16~235
SVSP_DP_MARGIN_COLOR[23:0]		R/W
0xE643 0xE644 0xE645	00000000 10000000 10000000	This signal is used to set the default color in output video in YUV colorspace. 0x000000 - Default 0xXXXXXX - Default color in YUV colorspace
SVSP_VIM_SCAL_TYPE[1:0]		R/W
0xE646	00000000	This signal is used to set the VIM scaling algorithm. In most cases, the scaler type should be left at the default setting. 00 - Proprietary ADI Algorithm 01 - Sharp 10 - Smooth 11 - Bilinear
SVSP_VIM_SCAL_SEP_ALPHA_EN		R/W
0xE646	00000000	This signal is used to control the interpolation parameter selection. Set to 1 for RGB input and set to 0 for YCbCr input. 0 - Three channel use middle channel parameter 1 - Three channel use separated parameter

Reg	Bits	Description
SVSP_VIM_SCAL_OVERSHOOT_CTRL[11:0]		R/W
0xE647 0xE648	00001000 00000000	This signal is used to control the overshoot in the scaling of input video. If set to a value larger than the default setting, more overshoot is allowed. 0x080 - Default
SVSP_DDR_BYPASS		R/W
0xE649	00000000	This bit is used to bypass external memory. This register's value will be used while svsp_man_set_ddr_bypass is 1. 0 - Not bypass external memory 1 - Bypass external memory
SVSP_BYPASS		R/W
0xE649	00000000	This bit is used to bypass the Secondary VSP. 0 - Not bypass Secondary VSP 1 - Bypass Secondary VSP
SVSP_P2I_ENABLE		R/W
0xE649	00000000	This bit is used to enable the PtoI in Secondary VSP. 0 - Disable 1 - Enable
M_P2I_ENABLE		R/W
0xE649	00000000	This bit is used to enable the PtoI In VSP_top. 0 - Disable 1 - Enable
SVSP_P2I_VID[7:0]		R/W
0xE64A	00000000	'This register is used to set the VIC of the PtoI in Secondary VSP. 0x00 - Default
M_P2I_VID[7:0]		R/W
0xE64B	00000000	'This register is used to set the VIC of the PtoI in VSP_top. 0x00 - Default
SVSP_VIM_SCAL_PANO_EN		R/W
0xE650	01100000	This bit is used to enable panorama scaling for the Secondary VSP. 0 - Disable panorama 1 - Enable panorama
SVSP_VIM_SCAL_ANTI_ALIAS_V_EN		R/W
0xE650	01100000	This bit is used to enable anti-aliasing filter for vertical direction. 0 - Disable 1 - Enable
SVSP_VIM_SCAL_ANTI_ALIAS_H_EN		R/W
0xE650	01100000	This bit is used to enable the anti-aliasing filter for horizontal direction. 0 - Disable 1 - Enable
SVSP_VIM_SCAL_PANO_POS[10:0]		R/W
0xE651 0xE652	00000000 00000000	This signal is used to define the width of the output video frame which is not stretched when panorama mode is enabled but rather scaled properly. The maximum value of this register is set by: svsp_vim_crop_width * (svsp_vim_scal_out_height/svsp_vim_crop_height) - svsp_vim_scal_out_width/2. This register sets half the width of the output frame which is to be scaled normally. By default, this register is set to 0 which means that all the input frame will be stretched. It is, therefore, recommended that this register is set by the user before enabling the panorama function. 0x000 - Default 0xXXX - Width of not-stretched image
SVSP_VIN_FR[7:0]		R/W
0xE658	00111100	This register is used to set the input video frame rate. This register's value will be used while svsp_autocfg_input_vid is 0.
SVSP_VOUT_FR[7:0]		R/W
0xE659	00111100	This register is used to set the output video frame rate. This register's value will be used while svsp_autocfg_output_vid is 0.
M_P2I_DROP_LINE_AS_PVSP_FLAG		R/W
0xE65B	00000000	In Game Mode, this bit is used to select an interlaced mode. If the PVSP works in game mode and the PVSP's input is interlaced, this bit should be set to 1 for the P2I block to drop interpolated lines. Otherwise, this bit should be set to 0. In external sync mode, this bit enables field tracking. When this bit is set low, it uses the internally generated field instead of the master one provided.

Reg	Bits	Description
SVSP_DP_SUBID[1:0]		R/W
0xE65D	00000000	This signal is used while svsp_autocfg_output_vid is 8/9/12/13/23/24/27/28. If svsp_autocfg_output_vid is 8 or 9 or 12 or 13, output timing vfrontporch is 4 while svsp_dp_subid is 0, output timing vfrontporch is 5 while svsp_dp_subid is 1. If svsp_autocfg_output_vid is 23 or 24 or 27 or 28, output timing vfrontporch is 2 while svsp_dp_subid is 0, output timing vfrontporch is 3 while svsp_dp_subid is1.
S_P2I_INVERT_VSP2D_FLAG		R/W
0xE65E	00000000	This bit is used to invert the field information being sent to the secondary P2I block.
SVSP_MAS_RESYNC_EN		R/W
0xE65F	00000000	This bit enables direct timing generation reset via external sync for the SVSP. This is for modes 2 and 3 only.
SVSP_AUTOCFG_INPUT_VID[7:0]		R/W
0xE660	00000000	This register is used to set the input timing VIC. If this register is 0, SVSP will use values in registers of svsp_vin_h, svsp_vin_v and svsp_vin_fr to set input video. 0x00 - Custom input video; 0xXX - Input timing VIC
SVSP_AUTOCFG_OUTPUT_VID[7:0]		R/W
0xE661	00000000	This register is used to set the output timing VIC. If this register is 0, SVSP will use values in registers of svsp_dp_decount, svsp_dp_hfrontporch, svsp_dp_hsynctime, svsp_dp_hbackporch, svsp_dp_activeline, svsp_dp_vfrontporch, svsp_dp_vsynctime, svsp_dp_vbackporch, svsp_dp_hpolarity, svsp_dp_vpolarity and svsp_vout_fr to set output video. 0x00 - Custom output video; 0xXX - Output timing VIC
SVSP_VIM_CROP_ENABLE		R/W
0xE662	10000000	This bit is used to enables the VIM crop. 0 - Disable 1 - Enable
SVSP_MAN_SCAL_OUT_ENABLE		R/W
0xE662	01000000	This bit is used to enable manually setting scaler output resolution. 0 - Disable 1 - Enable
SVSP_MAN_SCALER_PARA_ENABLE		R/W
0xE662	00100000	This bit is used to enable manually setting scaler parameters. 0 - Disable 1 - Enable
SVSP_FRC_LATENCY_MEASURE_EN		R/W
0xE662	00010000	This bit is used to enable measuring frame/Hsync latency. 0 - Disable 1 - Enable
SVSP_VOM_CROP_ENABLE		R/W
0xE662	00001000	This bit is used to enable the VOM crop. 0 - Disable 1 - Enable
SVSP_MAN_SET_DDR_BYPASS		R/W
0xE662	00000100	This bit is used to enable manually setting DDR bypass. If this bit is set to 1, SVSP will bypass DDR while svsp_ddr_bypass is 1, or not bypass DDR while svsp_ddr_bypass is 0. 0 - Disable 1 - Enable
SVSP_MAN_DP_TIMING_ENABLE		R/W
0xE663	00000000	This bit is used to enable manually setting output timing. 0 - Disable 1 - Enable
SVSP_MAN_INPUT_RES		R/W
0xE663	00000000	This bit is used to enable manual configuration of input resolution. 0 - Disable manual configuration of input resolution 1 - Enable manual configuration of input resolution

Reg	Bits	Description	
SVSP_FIELDBUFFER4_ADDR[31:0]			R/W
0xE664 0xE665 0xE666 0xE667	00000101 10101010 00110010 00000000	This signal is used to set the start address of field/frame buffer 4. Software should arrange memory space properly, avoiding conflict between different buffers.' 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 4	
SVSP_FIELDBUFFER5_ADDR[31:0]			R/W
0xE668 0xE669 0xE66A 0xE66B	00000110 00001001 00011110 00000000	This signal is used to set the start address of field/frame buffer 5. Software should arrange memory space properly, avoiding conflict between different buffers.' '0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 5	
SVSP_FIELDBUFFER6_ADDR[31:0]			R/W
0xE66C 0xE66D 0xE66E 0xE66F	00000110 01101000 00001010 00000000	This signal is used to set the start address of field/frame buffer 6. Software should arrange memory space properly, avoiding conflict between different buffers.' 0x00000000 - Default 0xXXXXXXXX - Start address of frame buffer 6	
SVSP_RB_FRAME_LATENCY[2:0]			R
0xE6F2	000 00000	This signal is used to readback the realtime frame latency. 0xXXX - Frame latency	
SVSP_RB_HSYNC_LATENCY[11:0]			R
0xE6F3 0xE6F4	00000000 0000 0000	This signal is used to readback the realtime Hsync latency. 0xXXX - HSync latency	
SVSP_RB_MAX_LATENCY[14:0]			R
0xE6F5 0xE6F6	00000000 0000000 0	This signal is used to readback the maximum frame/Hsync latency. Upper 3 bit is VS latency, Lower 12 bit HS latency. 0xXXX - Maximum of frame latency	
SVSP_RB_MIN_LATENCY[14:0]			R
0xE6F7 0xE6F8	00000000 0000000 0	This signal is used to readback the minimum frame/Hsync latency. Upper 3 bit is VS latency, Lower 12 bit HS latency. 0xXXX - Minimum of frame latency	

2.5 HDMI RX MAP

Reg	Bits	Description	
TERM_AUTO			R/W
0xE201	0000000**0**	This bit allows the user to select automatic or manual control of clock termination. 0 - Disable Termination automatic control 1 - Enable Termination automatic control	
AV_MUTE			R
0xE204	0**0**000000	This bit is a readback of AVMUTE status received in the last General Control packet received. 0 - AVMUTE not set 1 - AVMUTE set	
TMDS_PLL_LOCKED			R
0xE204	000000**0**0	This bit is a readback to indicate if the TMDS PLL is locked to the TMDS clock input of the selected HDMI port. 0 - The TMDS PLL is not locked 1 - The TMDS PLL is locked to the TMDS clock input of the selected HDMI port.	
RX_HDMI_MODE			R
0xE205	**0**0000000	This bit is a readback to indicate whether the stream processed by the HDMI core is a DVI or an HDMI stream. 0 - DVI Mode Detected 1 - HDMI Mode Detected	
DVI_HSYNC_POLARITY			R
0xE205	00**0**00000	This bit is a readback to indicate the polarity of the HSync encoded in the input stream 0 - The HSync is active low 1 - The HSync is active high	
DVI_VSYNC_POLARITY			R
0xE205	000**0**0000	This bit is a readback to indicate the polarity of the VSync encoded in the input stream 0 - The VSync is active low 1 - The VSync is active high	
HDMI_PIXEL_REPETITION[3:0]			R
0xE205	0000**0000**	This signal is a readback to provide the current HDMI pixel repetition value decoded from the AVI Infoframe received. The HDMI receiver automatically discards repeated pixel data and divides the pixel clock frequency appropriately as per the pixel repetition value. 0000 - 1x 0001 - 2x 0010 - 3x 0011 - 4x 0100 - 5x 0101 - 6x 0110 - 7x 0111 - 8x 1000 - 9x 1001 - 10x 1010 - 1111 - Reserved	
AUDIO_CHANNEL_MODE			R
0xE207	0**0**000000	This bit is a readback to indicate whether stereo or multichannel audio packets are being received. Note stereo packets may carry compressed multi-channel audio. 0 - Stereo Audio (may be compressed multichannel) 1 - Multichannel uncompressed audio detected (3-8 channels).	
DEEP_COLOR_MODE[1:0]			R
0xE20B	**00**000000	This control is a readback indicating the deep color mode information extracted from the general control packet. 00 - 8-bits per channel 01 - 10-bits per channel 10 - 12-bits per channel 11 - 16-bits per channel (not supported)	
GC_PACKET_DET			R
0xE217	0**0**000000	Packet Flag - GC Packet Detected	
ACR_PACKET_DET			R
0xE217	00**0**00000	Packet Flag - ACR Packet Detected	

Reg	Bits	Description	
GAMUT_MDATA_PCKT_DET			R
0xE217	0000**0**000	Packet Flag - Gamut Metadata Packet Detected	
ISRC2_PACKET_DET			R
0xE217	0000**0**000	Packet Flag - ISRC2 Packet Detected	
ISRC1_PACKET_DET			R
0xE217	00000**0**00	Packet Flag - ISRC1 Packet Detected	
ACP_PACKET_DET			R
0xE217	000000**0**0	Packet Flag - ACP Packet Detected	
VS_INFOFRAME_DET			R
0xE217	0000000**0**	Infoframe Flag - Vendor Specific Infoframe Detected	
MS_INFOFRAME_DET			R
0xE218	**0**0000000	Infoframe Flag - MS Infoframe Detected	
SPD_INFOFRAME_DET			R
0xE218	0**0**000000	Infoframe Flag - SPD Infoframe Detected	
AUDIO_INFOFRAME_DET			R
0xE218	00**0**00000	Infoframe Flag - Audio Infoframe Detected	
AVI_INFOFRAME_DET			R
0xE218	000**0**0000	Infoframe Flag - AVI Infoframe Detected	
HBR_AUDIO_PCKT_DET			R
0xE218	0000**0**000	This bit indicates if HBR Packets have been detected. It resets to zero on the 11th HSync leading edge following an HBR packet if a subsequent HBR packet has not been detected. It also resets if an Audio, DSD or DST packet sample packet has been received and after an HDMI reset condition. 0 - No HBR audio packet received within the last 10 HSyncs. 1 - HBR audio packet received within the last 10 HSyncs.	
DSD_PACKET_DET			R
0xE218	000000**0**0	This bit indicates if DSD Audio Packets have been detected. This bit resets to zero on the 11th HSync leading edge following a DSD packet or if an Audio, DST or HBR packet sample packet has been received or after an HDMI reset condition. 0 - No DSD packet received within the last 10 HSynsc. 1 - DSD packet received within the last 10 HSyncs.	
AUDIO_SAMPLE_PCKT_DET			R
0xE218	0000000**0**	This bit indicates if Audio Sample Packets have been detected. It resets to zero on the 11th HSync leading edge following an Audio packet if a subsequent audio sample packet has not been received or if a DSD, DST or HBR Audio packet sample packet has been received. 0 - No L_PCM or IEC 61937 compressed audio sample packet received within the last 10 HSyncs. 1 - L_PCM or IEC 61937 compressed audio sample packet received within the last 10 HSyncs.	
VS_INF_CKSUM_ERR			R
0xE219	**0**0000000	Flags when the latest received Vendor Specific infoframe had checksum error. This is redundant, there is an interrupt flag in IO map 0 - Vendor Specific Infoframe checksum error not detected. 1 - Vendor Specific Infoframe checksum error detected.	
MS_INF_CKSUM_ERR			R
0xE219	0**0**000000	Flags when the latest received MPEG Source infoframe had checksum error. This is redundant, there is an interrupt flag in IO map 0 - MS Infoframe checksum error not detected. 1 - MS Infoframe checksum error detected.	

Reg	Bits	Description
SPD_INF_CKSUM_ERR		R
0xE219	0<u>0</u>000000	Flags when the latest received Source Product Descriptor infoframe had checksum error. This is redundant, there is an interrupt flag in IO map 0 - SPD Infoframe checksum error not detected. 1 - SPD Infoframe checksum error detected.
AUD_INF_CKSUM_ERR		R
0xE219	000<u>0</u>0000	Flags when the latest received Audio infoframe had checksum error. This is to be used in conjunction with the interrupt InfoFrameError 0 - Audio Infoframe checksum error not detected. 1 - Audio Infoframe checksum error detected.
AVI_INF_CKSUM_ERR		R
0xE219	0000<u>0</u>000	Flags when the latest received AVI infoframe had checksum error. This is to be used in conjunction with the interrupt InfoFrameError 0 - AVI Infoframe checksum error not detected. 1 - AVI Infoframe checksum error detected.
DCFIFO_RESET_ON_LOCK		R/W
0xE21B	0001<u>1</u>000	This bit is used to enable the reset/re-centering of video FIFO on video PLL unlock 0 - Do not reset on video PLL lock 1 - Reset FIFO on video PLL lock
DCFIFO_KILL_NOT_LOCKED		R/W
0xE21B	0001<u>1</u>000	This bit control is used to control whether or not the output of the Video FIFO is set to zero when the video PLL is unlocked. 0 - FIFO data is output regardless of video PLL lock status 1 - FIFO output is zeroed if video PLL is unlocked
DCFIFO_KILL_DIS		R/W
0xE21B	00011<u>0</u>00	This bit is used to control whether or not the Video FIFO output is zeroed if there is more than one resynchronization of the pointers within 2 FIFO cycles. This behavior can be disabled with this bit. 0 - FIFO output set to zero if more than one resynchronization is necessary during two FIFO cycles 1 - FIFO output never set to zero regardless of how many resynchronizations occur
DCFIFO_LOCKED		R
0xE21C	0000<u>0</u>000	This bit is a readback to indicate if the Video FIFO is locked. 0 - Video FIFO is not locked. Video FIFO had to resynchronize between previous two Vsyncs 1 - Video FIFO is locked. Video FIFO did not have to resynchronize between previous two Vsyncs
DCFIFO_LEVEL[2:0]		R
0xE21C	00000<u>000</u>	This signal is a readback to indicate the distance between the read and write pointers. Overflow and underflow will read as level 0. The ideal centered functionality will read as 0b100. 000 - FIFO has underflowed or overflowed 001 - FIFO is about to overflow 010 - FIFO has some margin 011 - FIFO has some margin 100 - FIFO perfectly balanced 101 - FIFO has some margin. 110 - FIFO has some margin. 111 - FIFO is about to underflow
PDN_PKT_PROCESSOR		R/W
0xE21D	<u>0</u>0000000	This bit is used to enable a power saving feature that disables the clocking of the tmds rate section of the packet processor. Note that the audio clocking can be stopped separately, from the digital PLL (clock generator). 0 - Packet processor is active 1 - Packet processor is stopped, (i.e. powererd down)
UP_CONVERSION_MODE		R/W
0xE21D	0<u>0</u>000000	This bit is used to select linear or interpolated 4:2:2 to 4:4:4 conversion. A 4:2:2 incoming stream is always upconverted to a 4:4:4 stream before being sent to the CP. 0 - Cr and Cb samples are repeated in their respective channel. 1 - Interpolate Cr and Cb values.
OVERRIDE_DEEP_COLOR_MODE		R/W
0xE240	0<u>0</u>000000	This bit is used to override the Deep Color mode. 0 - The HDMI section unpacks the video data according to the deep-color information extracted from the General Control packets. (Normal operation) 1 - Override the deep color mode extracted from the General Control Packet. The HDMI section unpacks the video data according to the Deep Color mode set in DEEP_COLOR_MODE_USER[1:0].

Reg	Bits	Description
DEEP_COLOR_MODE_USER[1:0]		R/W
0xE24 0	0000**00**00	This control is used to manually set the Deep Color mode. The value set in this register is effective when override_deep_color_mode is set to 1. 00 - 8 bits per channel 01 - 10 bits per channel 10 - 12 bits per channel 11 - 16 bits per channel (not supported)
DEREP_N_OVERRIDE		R/W
0xE24 1	0100**0**000	This bit is used to allow the user to override the pixel repetition factor. derep_n is then used instead of hdmi_pixel_repetition[3:0] to discard video pixel data from the incoming HDMI stream. 0 - Automatic detection and processing of pixel repeated modes using the AVI infoframe information. 1 - Enables manual setting of the pixel repetition factor as per DEREP_N[3:0].
DEREP_N[3:0]		R/W
0xE24 1	0100**0000**	This signal is used to set the derepetition value if derep_n_override is set to 1. 0000 - DEREP_N+1 indicates the pixel and clock discard factor xxxx - DEREP_N+1 indicates the pixel and clock discard factor
QZERO_ITC_DIS		R/W
0xE24 7	00000**0**00	This bit is used to select manual control of the RGB colorimetry when the AVI infoframe field Q[1:0]=00. To be used in conjunction with qzero_rgb_full 0 - AVI InfoFrame ITC bit decides RGB-full or limited range in case Q[1:0]=00 1 - Manual RGB range as per QZERO_RGB_FULL.
QZERO_RGB_FULL		R/W
0xE24 7	000000**0**0	This signal is used to manually select the HDMI colorimetry when the AVI infoframe field Q[1:0]=00. It is valid only when qzero_itc_dis is set to 1. 0 - RGB-limited range when Q[1:0]=00 1 - RGB-full when Q[1:0]=00
ALWAYS_STORE_INF		R/W
0xE24 7	0000000**0**	This bit is used to force InfoFrames with checksum errors to be stored. 0 - Stores data from received InfoFrames only if their checksum is correct 1 - Always store the data from received InfoFrame regardless of their checksum
DIS_CABLE_DET_RST		R/W
0xE24 8	0**0**000000	This bit is used to disable the reset effects of cable detection. It should be set to 1 if the +5 V pins are unused and left unconnected. 0 - Resets the HDMI section if the 5 V input pin is inactive 1 - Do not use the 5 V input pins as reset signal for the HDMI section
GAMUT_IRQ_NEXT_FIELD		R/W
0xE25 0	000**0**0000	This bit is used to set the NEW_GAMUT_MDATA_RAW interrupt to detect when the new contents are applicable to next field or to indicate that the Gamut packet is new. This is done using header information of the gamut packet. 0 - Interrupt flag indicates that Gamut packet is new 1 - Interrupt flag indicates that Gamut packet is to be applied next field
HDMI_COLORSPACE[3:0]		R
0xE25 3	0000**0000**	This signal is used to provide a readback of the HDMI input colorspace decoded from the AVI infoframe. 0000 - RGB_LIMITED 0001 - RGB_FULL 0010 - YUV_601 0011 - YUV_709 0100 - XVYCC_601 0101 - XVYCC_709 0110 - YUV_601_FULL 0111 - YUV_709_FULL 1000 - sYCC 601 1001 - Adobe YCC 601 1010 - Adobe RGB
FILT_5V_DET_DIS		R/W
0xE25 6	**0**1011000	This bit is used to disable the digital glitch filter on the HDMI 5V detect signals. The filtered signals are used as interrupt flags, and also used to reset the HDMI section. The filter works from an internal ring oscillator clock and is therefore available in power-down mode. The clock frequency of the ring oscillator is 42MHz +/-10%. Note: If the 5 V pins are not used and left unconnected, the 5 V detect circuitry should be disconnected from the HDMI reset signal by setting dis_cable_det_rst to 1. This avoids holding the HDMI section in reset. 0 - Enabled 1 - Disabled

Reg	Bits	Description
FILT_5V_DET_TIMER[6:0]		R/W
0xE256	01011000	This bit is used to set the timer for the digital glitch filter on the HDMI +5 V detect inputs. The unit of this parameter is 2 clock cycles of the ring oscillator (~ 47ns). The input must be constantly high for the duration of the timer, otherwise the filter output remains low. The output of the filter returns low as soon as any change in the +5 V power signal is detected. 1011000 - Approximately 4.2us xxxxxxx - Time duration of +5 V deglitch filter. The unit of this parameter is 2 clock cycles of the ring oscillator (~ 47ns)
DCFIFO_RECENTER		SC
0xE25A	0000000	This bit is used as a reset to recenter the Video FIFO. This is a self clearing bit. 0 - Video FIFO normal operation. 1 - Video FIFO to re-centre.
FORCE_N_UPDATE		SC
0xE25A	0000000	This signal is used to force an N and CTS value update 0 - No effect 1 - Forces an update on the N and CTS values for audio clock regeneration
CTS[19:0]		R
0xE25B 0xE25C 0xE25D	00000000 00000000 0000	This control is used to provide a readback for the CTS value received in the HDMI datastream. xxxxxxxxxxxxxxxxxxxx - CTS value readback from HDMI stream
N[19:0]		R
0xE25D 0xE25E 0xE25F	0000 00000000 00000000	This control is used to provide a readback for the N value received in the HDMI datastream xxxxxxxxxxxxxxxxxxxx - N value readback from HDMI stream
CLOCK_TERMA_DISABLE		R/W
0xE283	1111111	This control is used to disable clock termination on port A. It can be used when term_auto is set to 0. 0 - Enable Termination port A 1 - Disable Termination port A
EQ_DYN_FREQ2		R/W
0xE289	0000000	This bit is used to set the HDMI equalizer mode for Port A. 0 - Disables equalizer dynamic mode. The equalizer is configured in static mode. This configuration is not recommended. 1 - Enables equalizer dynamic mode. This configuration is recommended.
EQ_DYN_FREQ2[3:0]		R/W
0xE28A	1010	This control is used to set the upper limit, limit 2, for the HDMI equalizer dynamic control frequency range. The frequency must be specified in MHz divided by 16. 0000 - Reserved. Do not use. 1010 - Default dynamic equalizer frequency limit 2. The default value corresponds to 160 MHz. xxxx - Frequency for limit 2.

2.6 HDMI RX INFOFRAME MAP

Reg	Bits	Description	
AVI_INF_PB[223:0]			R
0xE300	00000000	This readback is used to indicate the AVI InfoFrame Data	
0xE301	00000000		
0xE302	00000000		
0xE303	00000000		
0xE304	00000000		
0xE305	00000000		
0xE306	00000000		
0xE307	00000000		
0xE308	00000000		
0xE309	00000000		
0xE30A	00000000		
0xE30B	00000000		
0xE30C	00000000		
0xE30D	00000000		
0xE30E			
0xE30F			
0xE310			
0xE311			
0xE312			
0xE313			
0xE314			
0xE315			
0xE316			
0xE317			
0xE318			
0xE319			
0xE31A			
0xE31B			

Reg	Bits	Description	
AUD_INF_PB[111:0]			R
0xE31C 0xE31D 0xE31E 0xE31F 0xE320 0xE321 0xE322 0xE323 0xE324 0xE325 0xE326 0xE327 0xE328 0xE329	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the Audio InfoFrame Data	

Reg	Bits	Description	
SPD_INF_PB[223:0]			R
0xE32A	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Data	
0xE32B	00000000		
0xE32C	00000000		
0xE32D	00000000		
0xE32E	00000000		
0xE32F	00000000		
0xE330	00000000		
0xE331	00000000		
0xE332	00000000		
0xE333	00000000		
0xE334	00000000		
0xE335	00000000		
0xE336	00000000		
0xE337	00000000		
0xE338			
0xE339			
0xE33A			
0xE33B			
0xE33C			
0xE33D			
0xE33E			
0xE33F			
0xE340			
0xE341			
0xE342			
0xE343			
0xE344			
0xE345			

Reg	Bits	Description	
MS_INF_PB[111:0]			R
0xE346 0xE347 0xE348 0xE349 0xE34A 0xE34B 0xE34C 0xE34D 0xE34E 0xE34F 0xE350 0xE351 0xE352 0xE353	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the Moving Picture Expert Group (MPEG) Source InfoFrame Data	

Reg	Bits	Description
VS_INF_PB[223:0]		R
0xE354	00000000	This readback is used to indicate the Vendor Specific InfoFrame Data
0xE355	00000000	
0xE356	00000000	
0xE357	00000000	
0xE358	00000000	
0xE359	00000000	
0xE35A	00000000	
0xE35B	00000000	
0xE35C	00000000	
0xE35D	00000000	
0xE35E	00000000	
0xE35F	00000000	
0xE360	00000000	
0xE361	00000000	
0xE362		
0xE363		
0xE364		
0xE365		
0xE366		
0xE367		
0xE368		
0xE369		
0xE36A		
0xE36B		
0xE36C		
0xE36D		
0xE36E		
0xE36F		

Reg	Bits	Description	
ACP_PB[223:0]			R
0xE370	00000000	This readback is used to indicate the ACP InfoFrame Data	
0xE371	00000000		
0xE372	00000000		
0xE373	00000000		
0xE374	00000000		
0xE375	00000000		
0xE376	00000000		
0xE377	00000000		
0xE378	00000000		
0xE379	00000000		
0xE37A	00000000		
0xE37B	00000000		
0xE37C	00000000		
0xE37D	00000000		
0xE37E			
0xE37F			
0xE380			
0xE381			
0xE382			
0xE383			
0xE384			
0xE385			
0xE386			
0xE387			
0xE388			
0xE389			
0xE38A			
0xE38B			

Reg	Bits	Description	
ISRC1_PB[223:0]			R
0xE38C 0xE38D 0xE38E 0xE38F 0xE390 0xE391 0xE392 0xE393 0xE394 0xE395 0xE396 0xE397 0xE398 0xE399 0xE39A 0xE39B 0xE39C 0xE39D 0xE39E 0xE39F 0xE3A0 0xE3A1 0xE3A2 0xE3A3 0xE3A4 0xE3A5 0xE3A6 0xE3A7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the ISRC 1 InfoFrame Data	

Reg	Bits	Description	
ISRC2_PB[223:0]			R
0xE3A8 0xE3A9 0xE3AA 0xE3AB 0xE3AC 0xE3AD 0xE3AE 0xE3AF 0xE3B0 0xE3B1 0xE3B2 0xE3B3 0xE3B4 0xE3B5 0xE3B6 0xE3B7 0xE3B8 0xE3B9 0xE3BA 0xE3BB 0xE3BC 0xE3BD 0xE3BE 0xE3BF 0xE3C0 0xE3C1 0xE3C2 0xE3C3	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the ISRC 2 InfoFrame Data	

Reg	Bits	Description
GBD[223:0]		R
0xE3C4 0xE3C5 0xE3C6 0xE3C7 0xE3C8 0xE3C9 0xE3CA 0xE3CB 0xE3CC 0xE3CD 0xE3CE 0xE3CF 0xE3D0 0xE3D1 0xE3D2 0xE3D3 0xE3D4 0xE3D5 0xE3D6 0xE3D7 0xE3D8 0xE3D9 0xE3DA 0xE3DB 0xE3DC 0xE3DD 0xE3DE 0xE3DF	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This readback is used to indicate the Gamut InfoFrame Data
AVI_PACKET_ID[7:0]		R/W
0xE3E0	10000010	This control is used to set the AVI InfoFrame ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x00 to 0x1B 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x00 to 0x1B
AVI_INF_VERS[7:0]		R
0xE3E1	00000000	This readback is used to indicate the AVI InfoFrame Version
AVI_INF_LEN[7:0]		R
0xE3E2	00000000	This readback is used to indicate the AVI InfoFrame Length

Reg	Bits	Description	
AUD_PACKET_ID[7:0]			R/W
0xE3E3	10000100	This control is used to set the Audio InfoFrame ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x1C to 0x29 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x1C to 0x29	
AUD_INF_VERS[7:0]			R
0xE3E4	00000000	This readback is used to indicate the Audio InfoFrame Version	
AUD_INF_LEN[7:0]			R
0xE3E5	00000000	This readback is used to indicate the Audio InfoFrame Length	
SPD_PACKET_ID[7:0]			R/W
0xE3E6	10000011	This control is used to set the Source Product Descriptor InfoFrame ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x2A to 0x45 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x2A to 0x45	
SPD_INF_VERS[7:0]			R
0xE3E7	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Version	
SPD_INF_LEN[7:0]			R
0xE3E8	00000000	This readback is used to indicate the Source Product Descriptor InfoFrame Length	
MS_PACKET_ID[7:0]			R/W
0xE3E9	10000101	This control is used to set the MPEG Source InfoFrame ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x46 to 0x53 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x46 to 0x53	
MS_INF_VERS[7:0]			R
0xE3EA	00000000	This readback is used to indicate the MPEG Source InfoFrame Version	
MS_INF_LEN[7:0]			R
0xE3EB	00000000	This readback is used to indicate the MPEG Source InfoFrame Length	
VS_PACKET_ID[7:0]			R/W
0xE3EC	10000001	This control is used to set the Vendor Specific InfoFrame ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x54 to 0x6F 1xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x54 to 0x6F	
VS_INF_VERS[7:0]			R
0xE3ED	00000000	This readback is used to indicate the Vendor Specific InfoFrame Version	
VS_INF_LEN[7:0]			R
0xE3EE	00000000	This readback is used to indicate the Vendor Specific InfoFrame Length	
ACP_PACKET_ID[7:0]			R/W
0xE3EF	00000100	This control is used to set the ACP Packet ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x70 to 0x8B 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x70 to 0x8B	
ACP_TYPE[7:0]			R
0xE3F0	00000000	This readback is used to indicate the ACP Type	
ACP_HEADER2[7:0]			R
0xE3F1	00000000	This readback is used to indicate the ACP Header 2	
ISRC1_PACKET_ID[7:0]			R/W
0xE3F2	00000101	This control is used to set the ISRC1 Packet ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0x8C to 0xA7 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0x8C to 0xA7	

Reg	Bits	Description	
ISRC1_HEADER1[7:0]			R
0xE3F3	00000000	This readback is used to indicate the ISRC1 Header 1	
ISRC1_HEADER2[7:0]			R
0xE3F4	00000000	This readback is used to indicate the ISRC1 Header 2	
ISRC2_PACKET_ID[7:0]			R/W
0xE3F5	00000110	This control is used to set the ISRC2 Packet ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0xA8 to 0xC3 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0xA8 to 0xC3	
ISRC2_HEADER1[7:0]			R
0xE3F6	00000000	This readback is used to indicate the ISRC2 Header 1	
ISRC2_HEADER2[7:0]			R
0xE3F7	00000000	This readback is used to indicate the ISRC2 Header 2	
GAMUT_PACKET_ID[7:0]			R/W
0xE3F8	00001010	This control is used to set the Gamut Metadata Packet ID 0xxxxxxx - Packet type value of packet stored in InfoFrame Map, Address 0xC4 to 0xDF 1xxxxxxx - Packet type value of InfoFrame stored in InfoFrame Map, Address 0xC4 to 0xDF	
GAMUT_HEADER1[7:0]			R
0xE3F9	00000000	This readback is used to indicate the Gamut Metadata Header 1	
GAMUT_HEADER2[7:0]			R
0xE3FA	00000000	This readback is used to indicate the Gamut Metadata Header 2	
PKT_CNT_ID[7:0]			R/W
0xE3FD	10000001	Select which type of packet is going to be counted during each frame. Any packet header ID is supported to detect the count, but only collectable packets can be stored. Default is 0x81 for Vendor Specific Infoframe.	
EN_PKT_CNT_SEL			R/W
0xE3FE	0000**0**0000	This control is used to enable the feature whereby one can choose which of the multiple received packets per frame of type pkt_cnt_id is to be collected. This feature is only relevant for packet types that can be stored in the Infoframe map. 0 - Disabled. Collect every packet as it is received 1 - Enable selectively collecting one of multiple packets per frame of the same type	
PKT_CNT_SEL[3:0]			R/W
0xE3FE	0000**0000**	This control is used to select which one of the multplile received packets per frame of type pkt_cnt_id is to be collected. Must be enabled with en_pkt_cnt_sel. It should be manually changed after receiving the corresponding new packet detect flag 0000 - select the 1st packet after Vsync rising edge xxxx - 1110 - Select the 15th packet after Vsync rising edge 1111 - undefined	
RB_PKT_CNT[3:0]			R
0xE3FF	0000**0000**	This readback is used to indicate the count of number of packets of type pkt_cnt_id per frame, as detected between Vsync rising edges. The count readback is supported for any pkt_cnt_id, even those that can not be stored in the Infoframe map (e.g. General control packets, and even custom ID packets). The count will saturate to the 15 maximum, even if more packets are received. A count of zero represents that no packet were detected.	

2.7 TX1 MAIN MAP

Reg	Bits	Description	
N[19:0]			R/W
0xEC01 0xEC02 0xEC03	00000000 00000000 00000000	This signal is used to specifies the audio clock regeneration parameter N. This parameter is used with CTS to regenerate the audio clock in the receiver.	
SPDIF_SF[3:0]			R
0xEC04	00000000	This signal is used to readback the audio sampling frequency from the SPDIF channel. 0000 - 44.1kHz 0001 - NA 0010 - 48 kHz 0011 - 32kHz 0100 - NA 0101 - NA 0110 - NA 0111 - NA 1000 - 88.2kHz 1001 - NA 1010 - 96kHz 1011 - NA 1100 - 176.4kHz 1101 - NA 1110 - 192kHz 1111 - NA	
CTS_INTERNAL[19:0]			R
0xEC04 0xEC05 0xEC06	00000000 00000000 00000000	This signal is used to readback the automatically generated Cycle Time Stamp (CTS) parameter. This parameter is used with the N parameter to regenerate the audio clock in the receiver.	
CTS_MANUAL[19:0]			R/W
0xEC07 0xEC08 0xEC09	00000000 00000000 00000000	This signal is used to manually set the Cycle Time Stamp (CTS). This parameter is used with the N parameter to regenerate the audio clock in the receiver.	
CTS_SEL			R/W
0xEC0A	00000001	This bit is used to specify whether CTS is automatically or manually set. 0 - Automatic CTS. Use the internally generated CTS value 1 - Manual CTS. Use the CTS programmed via CTS_MANUAL[19:0]	
AUDIO_INPUT_SEL[2:0]			R/W
0xEC0A	00000001	This signal is used to specify the audio mode when the input format of the audio is specified. 000 - I2S 001 - SPDIF 010 - One Bit Audio (DSD) 011 - High Bit Rate (HBR) Audio 100 - Reserved	

Reg	Bits	Description
AUDIO_MODE[1:0]		R/W
0xEC0A	0000<u>00</u>01	This signal is used to specify the exact audio mode when the input format of the audio is specified. Case 1: DSD (audio_input_select = 0b010): 0x = DSD raw mode; 1x = SDIF-3 mode Case 2: HBR (audio_input_select = 0b011): 00 = 4 stream, with Bi-Phase Mark (BPM) encoding; 01 = 4 stream, without BPM encoding; 10 = 1 stream, with BPM encoding; 11 = 1 stream, without BPM encoding Case 3: DST (audio_input_select = 0b100): x0 = normal mode; 01 = DST 2x clock; 10 = DST 1x clock (DDR) Case 1 - DSD (AUDIO_INPUT_SELECT = 0b010) 0x - DSD raw mode 1x - SDIF-3 mode Case 2 - HBR (AUDIO_INPUT_SELECT = 0b011) 00 - 4 stream, with Bi-Phase Mark (BPM) encoding 01 - 4 stream, without BPM encoding 10 - 1 stream, with BPM encoding 11 - 1 stream, without BPM encoding
MCLK_RATIO[1:0]		R/W
0xEC0A	000000<u>01</u>	This signal is used to specify the ratio between the audio sampling frequency and the clock described using the N and CTS values. 00 - 128*fs 01 - 256*fs 10 - 384*fs 11 - 512*fs
SPDIF_EN		R/W
0xEC0B	<u>0</u>0001110	This bit is used to enable the SPDIF receiver. The SPDIF receiver is only needed when MCLK is internally generated. 0 - SPDIF receiver disabled 1 - SPDIF receiver enabled
MCLK_POL		R/W
0xEC0B	0<u>0</u>001110	This bit is used to specify the clock polarity for the input MCLK and SCLK for I2S and DSD. The clock polarity indicates the clock edge when input data is latched. 0 - Data latched on rising edge 1 - Data latched on falling edge
MCLK_EN		R/W
0xEC0B	000<u>0</u>1110	This bit is used to select the audio master clock that is used by the audio block. 0 - Use internally generated MCLK 1 - Use external MCLK
RX_AUD_PACKET_SEL		R/W
0xEC0B	0000111<u>0</u>	This bit is used to select the source of audio packet data routed into the HDMI transmitter. 0 - Get audio packet from external audio pins 1 - Get audio packet from internal audio receiver
AUDIO_SAMPLING_FREQ_SEL		R/W
0xEC0C	<u>1</u>0111100	This bit is used to select whether the audio sampling frequency is set automatically or manually (via I2C). 0 - Use sampling frequency from I2S stream, for SPDIF stream 1 - Use sampling frequency from I2C registers
CS_BIT_OVERRIDE		R/W
0xEC0C	1<u>0</u>111100	This bit is used to select the source of channel status bits when using I2S Mode 4. 0 - Use channel status bits from I2S stream 1 - Use channel status bits programmed in I2C registers
I2S_EN[3:0]		R/W
0xEC0C	1<u>1111</u>00	This signal is used to enable the I2S pins. 0000 - All I2S disabled 1111 - All I2S enabled
I2S_FORMAT[1:0]		R/W
0xEC0C	101111<u>00</u>	This signal is used to set the format of the I2S audio stream input to the part. 00 - I2S 01 - Right justified 10 - Left justified 11 - AES3 direct mode

Reg	Bits	Description
I2S_BIT_WIDTH[4:0]		R/W
0xEC0D	00011000	This signal is used to set the I2S bit width. This setting is for right justified audio only. It is not valid for widths greater than 24. 00000 - I2S[0], left channel 00001 - I2S[0], right channel 00010 - I2S[1], left channel 00011 - I2S[1], right channel 00100 - I2S[2], left channel 00101 - I2S[2], right channel 00110 - I2S[3], left channel 00111 - I2S[3], right channel
SUBPKT0_L_SRC[2:0]		R/W
0xEC0E	00000001	This signal is used to specify the source of sub packet 0, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT0_R_SRC[2:0]		R/W
0xEC0E	00000001	This signal is used to specify the source of sub packet 0, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT1_L_SRC[2:0]		R/W
0xEC0F	00010011	This signal is used to specify the source of sub packet 1, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT1_R_SRC[2:0]		R/W
0xEC0F	00010011	This signal is used to specify the source of sub packet 1, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT2_L_SRC[2:0]		R/W
0xEC10	00100101	This signal is used to specify the source of sub packet 2, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel

Reg	Bits	Description	
SUBPKT2_R_SRC[2:0]			R/W
0xEC10	00100101	This signal is used to specify the source of sub packet 2, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
SUBPKT3_L_SRC[2:0]			R/W
0xEC11	00110111	This signal is used to specify the source of sub packet 3, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
SUBPKT3_R_SRC[2:0]			R/W
0xEC11	00110111	This signal is used to specify the source of sub packet 3, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
CHANNEL_STATUS[1:0]			R/W
0xEC12	00000000	This signal is used to set the Channel Status bits [1:0]. Set to 0b00 as specified in IEC60958-3. Refer to IEC60958-3 specification. xx - Channel status bits 0 and 1	
CR_BIT			R/W
0xEC12	00000000	This bit is used to set the Channel Status Copyright Information. Refer to the IEC 60958-3 specification. 0 - Copyright asserted 1 - Copyright not asserted	
A_INFO[2:0]			R/W
0xEC12	00000000	This signal is used to set the Channel Status Emphasis information. Refer to the IEC 60958-3 specification. 000 - 2 audio channels without pre-emphasis 001 - 2 audio channels with 50/15uS pre-emphasis 010 - Reserved (for 2 audio channels with pre-emphasis) 011 - Reserved (for 2 audio channels with pre-emphasis) 100-111 - Reserved	
CLK_ACC[1:0]			R/W
0xEC12	00000000	This signal is used to set the Channel Status Clock Accuracy information. Refer to the IEC 60958-3 specification. 00 - level II - normal accuracy +/-1000 x 10^-6 01 - level I - high accuracy +/- 50 x 10^-6 10 - level III - variable pitch shifted clock 11 - Reserved	
CATEGORY_CODE[7:0]			R/W
0xEC13	00000000	This register is used to set the Channel Status Category Code. Refer to the IEC 60958-3 specification. 00000000 - Default value xxxxxxxx - Channel Status category code	
SOURCE_NUMBER[3:0]			R/W
0xEC14	00000000	This signal is used to set the Channel Status source number. 0000 - Default value xxxx - Channel Status source number	

Reg	Bits	Description	
WORD_LENGTH[3:0]			R/W
0xEC14	00000000	This signal is used to set the Channel Status Audio Word Length. Refer to the IEC 60958-3 specification. 0000 - Not specified 0001 - Not specified 0010 - 16 bits 0011 - 20 bits 0100 - 18 bits 0101 - 22 bits 0110 - Reserved 0111 - Reserved 1000 - 19 bits 1001 - 23 bits 1010 - 20 bits 1011 - 24 bits 1100 - 17 bits 1101 - 21 bits 1110 - Reserved 1111 - Reserved	
I2S_SF[3:0]			R/W
0xEC15	00000000	This signal is used to set the Sampling frequency for I2S audio. This information is used both by the audio Rx and the pixel rep. Other values reserved. 0000 - 44.1kHz 0001 - Do not use 0010 - 48kHz 0011 - 32kHz 0100 - Do not use 0101 - Do not use 0110 - Do not use 0111 - Do not use 1000 - 88.2kHz 1001 - Do not use 1010 - 96kHz 1011 - Do not use 1100 - 176.4kHz 1101 - Do not use 1110 - 192kHz 1111 - Do not use	
VFE_INPUT_ID[3:0]			R/W
0xEC15	00000000	This signal is used to specify the video input format. 0000 - RGB 444 or YCbCr 444 0001 - YCbCr 422 0101 - Pseudo 422 YCbCr All Other Values - Reserved	
VFE_OUTPUT_FORMAT[1:0]			R/W
0xEC16	00000000	This signal is used to set the output format. 10 - Reserved	
VFE_INPUT_CS			R/W
0xEC16	00000000	This bit is used to specify the video input colour space. 0 - RGB 1 - YCbCr	
GEN_444_EN			R/W
0xEC17	00000000	This bit is used to enable 4:2:2 to 4:4:4 up-conversion. 0 - Disable 1 - Enable	
ASPECT_RATIO			R/W
0xEC17	00000000	This bit is used to set the aspect ratio of input video. This bit is used to distinguish between CEA-861D video timing codes where aspect ratio is the only difference. 0 - 4:3 1 - 16:9	
CSC_EN			R/W
0xEC18	01000110	This bit is used to enable the colour space converter. 0 - CSC Disabled 1 - CSC Enabled	

Reg	Bits	Description
CSC_SCALING_FACTOR[1:0]		R/W
0xEC18	01000110	This signal is used to specify the CSC scaling factor. The CSC scaling factor sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380
CSC_A1[12:0]		R/W
0xEC18 0xEC19	0100011001100010	This signal is used to specify the CSC coefficient for a1.
CSC_A2[12:0]		R/W
0xEC1A 0xEC1B	0000010010101000	This signal is used to specify the CSC coefficient for a2.
CSC_A3[12:0]		R/W
0xEC1C 0xEC1D	0000000000000000	This signal is used to specify the CSC coefficient for a3.
CSC_A4[12:0]		R/W
0xEC1E 0xEC1F	0001110010000100	This signal is used to specify the CSC coefficient for a4.
CSC_B1[12:0]		R/W
0xEC20 0xEC21	0001110010111111	This signal is used to specify the CSC coefficient for b1.
CSC_B2[12:0]		R/W
0xEC22 0xEC23	0000010010101011	This signal is used to specify the CSC coefficient for b2.
CSC_B3[12:0]		R/W
0xEC24 0xEC25	0001111001110000	This signal is used to specify the CSC coefficient for b3.
CSC_B4[12:0]		R/W
0xEC26 0xEC27	0000001000011110	This signal is used to specify the CSC coefficient for b4.
CSC_C1[12:0]		R/W
0xEC28 0xEC29	0000000000000000	This signal is used to specify the CSC coefficient for c1.
CSC_C2[12:0]		R/W
0xEC2A 0xEC2B	0000010010101000	This signal is used to specify the CSC coefficient for c2.

Reg	Bits	Description	
CSC_C3[12:0]			R/W
0xEC2C 0xEC2D	00001000 00010010	This signal is used to specify the CSC coefficient for c3.	
CSC_C4[12:0]			R/W
0xEC2E 0xEC2F	00011011 10101100	This signal is used to specify the CSC coefficient for c4.	
PR_MODE[1:0]			R/W
0xEC3B	10000000	This signal is used to specify the pixel repetition mode selection. This should be set to 00 unless a non CEA-861 standard video resolution must be supported. 00 - auto mode 01 - max mode 10 - manual mode 11 - manual mode	
PR_PLL_MANUAL[1:0]			R/W
0xEC3B	10000000	This signal is used to specify the ratio between the input pixel clock and the TMDS output clock when manual pixel repetition is enabled. 00 - x1 01 - x2 10 - x4 11 - x4	
PR_VALUE_MANUAL[1:0]			R/W
0xEC3B	10000000	This signal is used to specify the user programmed pixel repetition sent to the downstream sink. This field is used in manual pixel repetition. 00 - x1 01 - x2 10 - x4 11 - x4	
VIC_MANUAL[5:0]			R/W
0xEC3C	00000000	This signal is used to specify the video code (VIC) to send to the downstream sink. Refer to the CEA-861 specification. xxxxxxx - VIC to send to the downstream sink.	
PR_TO_RX[1:0]			R
0xEC3D	00000000	This signal is used to readback the pixel repetition value sent to the downstream sink. 00 - x1 01 - x2 10 - x4 11 - x4	
VIC_TO_RX[5:0]			R
0xEC3D	00000000	This signal is used to set the AVI InfoFrame video code (VIC) to send to the downstream sink. xxxxxx - VIC sent to the downstream sink	
VIC_DETECTED[5:0]			R
0xEC3E	00000000	This signal is used to readback the input video code (VIC) detected (refer to the CEA-861 specification).	
AUX_VIC_DETECTED[2:0]			R
0xEC3F	00000000	This register returns the format of video inputs that have a resolution not defined in the CEA 861 specification. 000 - Set by Register 3E 001 - 240p Not Active 010 - 576i not active 011 - 288p not active 100 - 480i active 101 - 240p active 110 - 576i active 111 - 288p active	

Reg	Bits	Description	
PROGRESSIVE_MODE_INFO[1:0]			R
0xEC3F	000<u>00</u>000	This bit is used to specify additional information for 240p or 288p input formats. 00 - Reserved 01 - 262 total lines per frame for 240p and 312 total lines per frame for 288p 10 - 263 total lines per frame for 240p and 313 total lines per frame for 288p 11 - Reserved for 240p and 314 total lines per frame for 288p	
GC_PKT_EN			R/W
0xEC40	<u>0</u>0000000	This bit is used to enable the General Control Packet. 0 - Disabled 1 - Enabled	
SPD_PKT_EN			R/W
0xEC40	0<u>0</u>000000	This bit is used to enable the Source Product Descriptor InfoFrame. 0 - Disabled 1 - Enabled	
MPEG_PKT_EN			R/W
0xEC40	00<u>0</u>00000	This bit is used to enable the MPEG Packet. 0 - Disabled 1 - Enabled	
ACP_PKT_EN			R/W
0xEC40	000<u>0</u>0000	This bit is used to enable the ACP Packet. 0 - Disabled 1 - Enabled	
ISRC_PKT_EN			R/W
0xEC40	0000<u>0</u>000	This bit is used to enable the ISRC Packet. 0 - Disabled 1 - Enabled	
GM_PKT_EN			R/W
0xEC40	00000<u>0</u>00	This bit is used to enable the Gamut Metadata Packet. 0 - Disabled 1 - Enabled	
SPARE_PKT1_EN			R/W
0xEC40	000000<u>0</u>0	This bit is used to enable the Spare Packet 2. 0 - Disabled 1 - Enabled	
SPARE_PKT0_EN			R/W
0xEC40	0000000<u>0</u>	This bit is used to enable the Spare Packet 1. 0 - Disabled 1 - Enabled	
SYSTEM_PD			R/W
0xEC41	0<u>1</u>010000	This bit is used to power down the TX. 0 - Normal operation 1 - Power down TX	
HPD_STATE			R
0xEC42	1<u>0</u>010000	This bit is used to readback the state of the hot plug detect. 0 - Hot Plug Detect inactive (low) 1 - Hot Plug active (high)	
RX_SENSE_STATE			R
0xEC42	10<u>0</u>10000	This bit is used to readback the state of the Rx sense. 0 - HDMI clock termination not detected 1 - HDMI clock termination detected	
I2S_32BIT_MODE			R
0xEC42	1001<u>0</u>000	This bit is used to readback the I2S mode detection. It shows the number of SCLK periods per LRCLK period. 0 - I2S 32 bit mode detected 1 - I2S 64 bit mode detected	

Reg	Bits	Description	
N_CTS_PKT_EN			R/W
0xEC44	01111001	This bit is used to enable the N and CTS packets. 0 - Disable N_CTS packet 1 - Enable N_CTS packet	
AUDIO_SAMPLE_PKT_EN			R/W
0xEC44	01111001	This bit is used to enable the Audio Sample Packet. 0 - Disable audio sample packet 1 - Enable audio sample packet	
AVIIF_PKT_EN			R/W
0xEC44	01111001	This bit is used to enable the AVI InfoFrame Packet. 0 - Disable AVI InfoFrame 1 - Enable AVI InfoFrame	
AUDIOIF_PKT_EN			R/W
0xEC44	01111001	This bit is used to enable the Audio InfoFrame. 0 - Disable audio InfoFrame 1 - Enable audio InfoFrame	
PACKET_MEMORY_ADDRESS[7:0]			R/W
0xEC45	01110000	This register is used to set the I2C address for the packet memory.	
DSD_EN[7:0]			R/W
0xEC46	00000000	This register is used to enable the DSD data channel.	
DSD_MUX_EN			R/W
0xEC47	00000001	This bit is used to enable DSD mode. 0 - Disabled 1 - Enabled	
PAPB_SYNC			R/W
0xEC47	00000001	This bit is used to synchronize the Pa and Pb syncwords with subpacket 0 for HBR audio. 0 - No function 1 - Synchronize Pa and Pb syncwords with subpacket 0	
SAMPLE_INVALID[3:0]			R/W
0xEC47	00000001	This signal is used to indicate that the subpackets have valid data. xxx0 - subpackets 0 have valid data xxx1 - subpackets 0 do not have valid data xx0x - subpackets 1 have valid data xx1x - subpackets 1 do not have valid data x0xx - subpackets 2 have valid data x1xx - subpackets 2 do not have valid data 0xxx - subpackets 3 have valid data 1xxx - subpackets 3 do not have valid data	
ARC_EFF_TRAN_EN			R/W
0xEC47	00000001	When enable it ensures more efficient transmission of ARC packets and audio samples in 176.4kHz and 192kHz modes. This ensures ACR packets can get sent at the right rate (~1ms). 0 - ARC packet efficient transmision disable. 1 - ARC packet efficient transmition enable.	
DITHER_MODE[5:0]			R/W
0xEC49	01010100	This signal is used to select the ADI proprietary downdither method. The downdither module has 3 stages, each two bits for one stage. [5:4] for 14 bits to 12 bits, [3:2] for 12 bits to 10 bits, [1:0] for 10 bits to 8 bits. 000000 - Active dither 101010 - Truncate	
AUTO_CHECKSUM_EN			R/W
0xEC4A	10000000	This bit is used to enable autocheck sum generation for infoframes. 0 - Disabled 1 - Use the auto check sum generated by the chip for all IF packets	
AVI_UPDATE			R/W
0xEC4A	10000000	This bit is used to trigger an update of the AVI infoframe. 0 - Disabled 1 - Enabled	

Reg	Bits	Description	
AUDIO_UPDATE			R/W
0xEC4A	1000000	This bit is used to trigger an update of the Audio infoframe. 0 - Disabled 1 - Enabled	
GCP_UPDATE			R/W
0xEC4A	10000000	This bit is used to trigger an update of the GCP infoframe. 0 - Disabled 1 - Enabled	
MAN_LAYOUT_EN			R/W
0xEC4A	10000000	This bit is used to manually set whether audio layout 0 or 1 are used. When transmitting HDMI audio sample packets with compressed audio the layout is automatically set to 1 (multichannel). When enabled, this bit in conjunction with Man_layout_sel allow users to select layout 0. 0 - The Audio sample packet layout is set automatically 1 - The audio sample packet layout value is set by Man_layout_sel	
MAN_LAYOUT_SEL			R/W
0xEC4A	10000000	This bit is used to select the audio layout value. 0 - Audio Layout 0 is used if Man_layout_en = 1 1 - Audio Layout 1 is used if Man_layout_en = 1	
CLEAR_AVMUTE			R/W
0xEC4B	00000000	This bit is used to control the CLEAR_AVMUTE signal. 0 - Set CLEAR_AVMUTE to 0 1 - Set CLEAR_AVMUTE to 1	
SET_AVMUTE			R/W
0xEC4B	00000000	This bit is used to control the SET_AVMUTE signal. 0 - Set SET_AVMUTE to 0 1 - Set SET_AVMUTE to 1	
GC_CD[3:0]			R/W
0xEC4C	00000000	This signal is used to set the colour depth of the video send to downstream sink. 0000 - colour depth not indicated 0100 - 24 bits per pixel 0101 - 30 bits per pixel 0110 - 36 bits per pixel 0111-111 - Reserved	
GC_BYTE2[7:0]			R/W
0xEC4D	00000000	This register is used to set byte 2 of the General Control subpacket.	
GC_BYTE3[7:0]			R/W
0xEC4E	00000000	This register is used to set byte 3 of the General Control subpacket.	
GC_BYTE4[7:0]			R/W
0xEC4F	00000000	This register is used to set byte 4 of the General Control subpacket.	
GC_BYTE5[7:0]			R/W
0xEC50	00000000	This register is used to set byte 5 of the General Control subpacket.	
GC_BYTE6[7:0]			R/W
0xEC51	00000000	This register is used to set byte 6 of the General Control subpacket.	
AVI_VERSION[2:0]			R/W
0xEC52	00000010	This signal is used to specify the AVI InfoFrame version.	
AVI_LENGTH[4:0]			R/W
0xEC53	00001101	This signal is used to specify the AVI InfoFrame length. This is the number of valid bytes in the AVI InfoFrame packet body excluding the checksum.	
AVI_CHECKSUM[7:0]			R/W
0xEC54	00000000	This register is used to specify the AVI InfoFrame checksum. The checksum should only be used in manual checksum mode (See AUTO_CHECKSUM_EN).	

Reg	Bits	Description	
AVI_BYTE1_7			R/W
0xEC55	0000000	This bit is used to set bit 7 of AVI InfoFrame Data Byte 1. Set to 0 as per the CEA 861 specification.	
Y1Y0[1:0]			R/W
0xEC55	00000000	This signal is used to set the Output Format (AVI InfoFrame). 00 - RGB 01 - YCbCr 422 10 - YCbCr 444 11 - reserved	
A0			R/W
0xEC55	00000000	This bit is used to set the Active Format Information status (AVI InfoFrame). 0 - No data 1 - Active format Information valid	
B1B0[1:0]			R/W
0xEC55	00000000	This signal is used to set the Bar Information (AVI InfoFrame). 00 - No bar information 01 - Vertical bar information valid 10 - Horizontal bar information valid 11 - Horizontal and vertical bar information valid	
S1S0[1:0]			R/W
0xEC55	00000000	This signal is used to set the Scan Information (AVI InfoFrame). 00 - No information 01 - Overscanned (TV) 10 - Underscanned (PC) 11 - Undefined	
C1C0[1:0]			R/W
0xEC56	00000000	This signal is used to set the Colorimetry (AVI InfoFrame). 00 - No data 01 - SMPTE 170M, ITU601 10 - ITU709 11 - Undefined	
M1M0[1:0]			R/W
0xEC56	00000000	This signal is used to set the Picture Aspect Ratio (AVI InfoFrame). 00 - No data 01 - 4:3 10 - 16:9 11 - Undefined	
R[3:0]			R/W
0xEC56	00000000	This signal is used to set the Active Format Aspect Ratio (AVI InfoFrame). 1000 - Same as aspect ratio 1001 - 4:3 (centre) 1010 - 16:9 (centre) 1011 - 14:9 (centre)	
ITC			R/W
0xEC57	00000000	This bit is used to set the IT Content (AVI InfoFrame). 0 - Not available 1 - Available	
EC[2:0]			R/W
0xEC57	00000000	This signal is used to set the Extended Colorimetry (AVI InfoFrame). 000 - XVYCC 601 001 - XVUCC 709	
Q1Q0[1:0]			R/W
0xEC57	00000000	This signal is used to set the RGB Quantization Range (AVI InfoFrame). 00 - Default range 01 - Limited range 10 - Full range 11 - Fixed	

Reg	Bits	Description	
SC[1:0]			R/W
0xEC57	000000<u>00</u>	This signal is used to set the Non-Uniform Picture Scaling information (AVI InfoFrame). 00 - No known non-uniform scaling 01 - Picture has been scaled horizontally 10 - Picture has been scaled vertically 11 - Picture has been scaled horizontally and vertically	
AVI_BYTE4_7			R/W
0xEC58	<u>0</u>0000000	This bit is used to set bit 7 of the AVI InfoFrame Data, byte 4. Set to 0 as per the CEA-861 specification.	
AVI_BYTE5_7_4[3:0]			R/W
0xEC59	<u>0000</u>0000	This signal is used to set bits [7:4] of the AVI InfoFrame Data, byte 5.	
AVI_BYTE6[7:0]			R/W
0xEC5A	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 6.	
AVI_BYTE7[7:0]			R/W
0xEC5B	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 7.	
AVI_BYTE8[7:0]			R/W
0xEC5C	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 8.	
AVI_BYTE9[7:0]			R/W
0xEC5D	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 9.	
AVI_BYTE10[7:0]			R/W
0xEC5E	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 10.	
AVI_BYTE11[7:0]			R/W
0xEC5F	<u>00000000</u>	'This register is used to specify the AVI Infoframe, byte 11.	
AVI_BYTE12[7:0]			R/W
0xEC60	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 12.	
AVI_BYTE13[7:0]			R/W
0xEC61	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 13.	
AVI_BYTE14[7:0]			R/W
0xEC62	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 14.	
AVI_BYTE15[7:0]			R/W
0xEC63	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 15.	
AVI_BYTE16[7:0]			R/W
0xEC64	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 16.	
AVI_BYTE17[7:0]			R/W
0xEC65	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 17.	
AVI_BYTE18[7:0]			R/W
0xEC66	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 18.	
AVI_BYTE19[7:0]			R/W
0xEC67	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 19.	

Reg	Bits	Description	
AVI_BYTE20[7:0]			R/W
0xEC68	00000000	This register is used to specify the AVI Infoframe, byte 20.	
AVI_BYTE21[7:0]			R/W
0xEC69	00000000	This register is used to specify the AVI Infoframe, byte 21.	
AVI_BYTE22[7:0]			R/W
0xEC6A	00000000	This register is used to specify the AVI Infoframe, byte 22.	
AVI_BYTE23[7:0]			R/W
0xEC6B	00000000	This register is used to specify the AVI Infoframe, byte 23.	
AVI_BYTE24[7:0]			R/W
0xEC6C	00000000	This register is used to specify the AVI Infoframe, byte 24.	
AVI_BYTE25[7:0]			R/W
0xEC6D	00000000	This register is used to specify the AVI Infoframe, byte 25.	
AVI_BYTE26[7:0]			R/W
0xEC6E	00000000	This register is used to specify the AVI Infoframe, byte 26.	
AVI_BYTE27[7:0]			R/W
0xEC6F	00000000	This register is used to specify the AVI Infoframe, byte 27.	
AUDIOIF_VERSION[2:0]			R/W
0xEC70	00000001	This signal is used to specify the Audio InfoFrame Version.	
AUDIOIF_LENGTH[4:0]			R/W
0xEC71	00001010	This signal is used to set the Audio InfoFrame length. This is the number of valid bytes in the Audio InfoFrame packet body excluding the checksum.	
AUDIOIF_CHECKSUM[7:0]			R/W
0xEC72	00000000	This register is used to set the Audio InfoFrame Checksum. This should only be used in manual checksum mode (See AUTO_CHECKSUM_EN).	
AUDIOIF_CT[3:0]			R/W
0xEC73	0000 0000	This signal is used to set the Audio Coding Type (Audio InfoFrame).	
AUDIOIF_BYTE1_3			R/W
0xEC73	0000 0 000	This bit is used to set bit 3 of Audio InfoFrame Data, byte 1. Set to 0 as per the CEA-861 specification.	
AUDIOIF_CC[2:0]			R/W
0xEC73	00000 000	This signal is used to set the Audio Channel Count (Audio InfoFrame). 000 - Refer to Stream Header 001 - 2 channels 010 - 3 channels 011 - 4 channels 100 - 5 channels 101 - 6 channels 110 - 7 channels 111 - 8 channels	
AUDIOIF_BYTE2_7_5[2:0]			R/W
0xEC74	000 00000	This signal is used to set bits [7:5] of Audio InfoFrame Data, byte 2. Set to 000 as per the CEA-861 specification.	

Reg	Bits	Description	
AUDIOIF_SF[2:0]			R/W
0xEC74	000000	This signal is used to specify the Audio Sampling Frequency in the Audio InfoFrame. 000 - Case 1: Not DSD audio or Case 2: DSD audio (AUDIO_INPUT_SEL = 0b010) 001 - 64 x 32 kHz 010 - 64 x 44.1 kHz 011 - 64 x 48 kHz 100 - 64 x 88.2 kHz 101 - 64 x 96 kHz 110 - 64 x 176.4 kHz 111 - 64 x 192 kHz	
AUDIOIF_SS[1:0]			R/W
0xEC74	00000000	This signal is used to set the Sample Size (Audio InfoFrame).	
AUDIOIF_BYTE3[7:0]			R/W
0xEC75	00000000	This register is used to set the Data Byte 3 (Audio InfoFrame). Set to 0x0 as per the CEA-861 specification.	
AUDIOIF_CA[7:0]			R/W
0xEC76	00000000	This register is used to set the Speaker Mapping or placement (Audio InfoFrame). 00000000 - Default value xxxxxxxx - Speaker mapping	
AUDIOIF_DM_INH			R/W
0xEC77	00000000	This bit is used to set the Down-mix Information (Audio InfoFrame). 0 - Permitted or no information about this 1 - Prohibited	
AUDIOIF_LSV[3:0]			R/W
0xEC77	00000000	This signal is used to set the Audio Level Shift Value (Audio InfoFrame). 0000 - 0dB attenuation 0001 - 1dB attenuation 0010 - 2dB attenuation 0011 - 3dB attenuation 0100 - 4dB attenuation 0101 - 5dB attenuation 0110 - 6dB attenuation 0111 - 7dB attenuation 1000 - 8dB attenuation 1001 - 9dB attenuation 1010 - 10dB attenuation 1011 - 11dB attenuation 1100 - 12dB attenuation 1101 - 13dB attenuation 1110 - 14dB attenuation 1111 - 15dB attenuation	
AUDIOIF_BYTE5_2_0[2:0]			R/W
0xEC77	00000000	This signal is used to set bits [2:0] of Audio InfoFrame Data, byte 5. Set to 000 as per the CEA-861 specification.	
AUDIOIF_BYTE6[7:0]			R/W
0xEC78	00000000	This register is used to set Data Byte 6 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE7[7:0]			R/W
0xEC79	00000000	This register is used to set Data Byte 7 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE8[7:0]			R/W
0xEC7A	00000000	This register is used to set Data Byte 8 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE9[7:0]			R/W
0xEC7B	00000000	This register is used to set Data Byte 9 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	

Reg	Bits	Description	
AUDIOIF_BYTE10[7:0]			R/W
0xEC7C	00000000	This register is used to set Data Byte 10 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
PRE_EN_CH0			R/W
0xEC80	0111**1**111	Enable data channel 0 1 - Enabled 0 = Disabled	
PRE_EN_CH1			R/W
0xEC80	01111**1**11	Enable data channel 1 1 - Enabled 0 = Disabled	
PRE_EN_CH2			R/W
0xEC80	011111**1**1	Enable data channel 2 1 - Enabled 0 = Disabled	
PRE_EN_CLK			R/W
0xEC80	0111111**1**	Enable clock channel 1 - Enabled 0 = Disabled	
CHG_INJ_CH0[3:0]			R/W
0xEC81	**1000**1000	Binary control of charge injection for data channel 0 with LSB cap value of 77fF	
CHG_INJ_CH1[3:0]			R/W
0xEC81	1000**1000**	Binary control of charge injection for data channel 1 with LSB cap value of 77fF	
CHG_INJ_CH2[3:0]			R/W
0xEC82	**1000**1000	Binary control of charge injection for data channel 2 with LSB cap value of 77fF	
CHG_INJ_CLK[3:0]			R/W
0xEC82	1000**1000**	Binary control of charge injection for clock channel with LSB cap value of 77fF	
CH0_STERM_DISABLE			R/W
0xEC83	**0**0000000	This bit is used to disable source termination on channel 0. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CH1_STERM_DISABLE			R/W
0xEC84	**0**0000000	This bit is used to disable source termination on channel 1. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CH2_STERM_DISABLE			R/W
0xEC85	**0**0000000	This bit is used to disable source termination on channel 2. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CLK_STERM_DISABLE			R/W
0xEC86	**0**0000000	This bit is used to disable source termination on the clk channel. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
HPD_INT_EN			R/W
0xEC94	**1**1000000	This bit is used to enable the HPD interrupt. 0 - Enabled 1 - Disabled	
RX_SENSE_INT_EN			R/W
0xEC94	1**1**000000	This bit is used to enable the RX Sense interrupt. 0 - Enabled 1 - Disabled	
VSYNC_INT_EN			R/W
0xEC94	11**0**00000	This bit is used to enable the Vsync Edge interrupt. 0 - Enabled 1 - Disabled	

Reg	Bits	Description	
EDID_READY_INT_EN			R/W
0xEC94	1100000	This bit is used to enable the EDID Ready interrupt. 0 - Enabled 1 - Disabled	
HDCP_AUTHENTICATED_INT_EN			R/W
0xEC94	110000	This bit is used to enable the HDCP Authenticated interrupt. 0 - Enabled 1 - Disabled	
RI_READY_INT_EN			R/W
0xEC94	11000000	This bit is used to enable the Ri Ready interrupt. 0 - Enabled 1 - Disabled	
HDCP_ERROR_INT_EN			R/W
0xEC95	0000000	This bit is used to enable the HDCP Controller Error interrupt. 0 - Enabled 1 - Disabled	
BKSV_FLAG_INT_EN			R/W
0xEC95	000000	This bit is used to enable the BKSV Flag interrupt. 0 - Enabled 1 - Disabled	
HPD_INT			R/W
0xEC96	0000000	This bit is used to readback and control the HPD interrupt. 0 - Interrupt not active. 1 - Interrupt active. A transition for high to low or low to high has been detected on the input HPD signal	
RX_SENSE_INT			R/W
0xEC96	000000	This bit is used to readback and control the Rx Sense interrupt. 0 - Interrupt Not Active 1 - Interrupt Active. The TMDS clock lines voltage has crossed 1.8V from high to low or low to high	
VSYNC_INT			R/W
0xEC96	00000	This bit is used to readback and control the Vsync edge interrupt. 0 - Interrupt not active 1 - Interrupt active. A leading edge on the input Vsync has been detected.	
EDID_READY_INT			R/W
0xEC96	0000000	This bit is used to readback and control the EDID Ready interrupt. 0 - no interrupt detected 1 - interrupt detected	
HDCP_AUTHENTICATED_INT			R/W
0xEC96	000000	This bit is used to readback and control the HDCP Authenticated interrupt. 0 - no interrupt detected 1 - interrupt detected	
RI_READY_INT			R/W
0xEC96	00000000	This bit is used to readback and control the Ri Ready interrupt. 0 - Interrupt not active 1 - interrupt active	
HDCP_ERROR_INT			R/W
0xEC97	0000000	This bit is used to readback and control the HDCP/EDID Controller Error interrupt. Refer to HDCP_CONTROLLER_ERROR for error codes. 0 - Interrupt not active 1 - Interrupt active. The HDCP/EDID controller has reported an error. This error is available in HDCP_CONTROLLER_ERROR	
BKSV_FLAG_INT			R/W
0xEC97	000000	This bit is used to readback and control the BKSV Flag interrupt. 0 - Interrupt not active 1 - Interrupt active. The KSVs from the downstream sink have been read and available in the Memory Map	

Reg	Bits	Description
CEC_TX_ARBITRATION_LOST_INT		R/W
0xEC97	0000<u>0</u>0000	This bit is used to readback and control the CEC TX Arbitration Lost interrupt. 0 - Interrupt not active 1 - Interrupt active. The CEC controller is indicating that the TX has lost arbitration
CEC_TX_RETRY_TIMEOUT_INT		R/W
0xEC97	0000<u>0</u>000	This bit is used to readback and control the CEC TX Retry Timeout interrupt. 0 - Interrupt not active 1 - Interrupt active. The CEC controller is indicating that the TX retry timeout has expired
CEC_PD		R/W
0xEC98	0000<u>0</u>000	This bit is used to powerdown the CEC controller. 0 - CEC enabled 1 - CEC power down and CEC map reset
HIGH_FREQ_VIDEO[1:0]		R/W
0xEC9E	<u>00</u>000000	This signal is used to set the high frequency video enable. This control can be used to configure HDMI TX for 2x and 4x refresh rates. 00 - "normal" refresh rates 01 - 2x refresh rates 10 - 4x refresh rates 11 - "normal" refresh rates
VIDEO_OFFSET_CTRL[1:0]		R/W
0xEC9E	00<u>00</u>0000	This signal is used to control the video offset control. Bit 1 is the control to offset the U and V channels. Bit 0 is the black image control to force black video data out of the TX. 00 - No video offset & black image disabled 01 - No video offset & black image enabled 10 - Video offset & black image disabled 11 - Video offset & black image enabled
HPD_OVERRIDE[1:0]		R/W
0xEC9F	00<u>00</u>0000	This signal is used to select the source of the internal HPD signal. 00 - HPD from HPD pin and CDC HPD 01 - HPD from CDC HPD 10 - HPD from HPD pin 11 - HPD set to 1
HDCP_START_DELAY[2:0]		R/W
0xECAB	00<u>100</u>000	This signal is used to control the delay programmability between HDCP Enable and read BKSV. 000 - no delay 001 - 1ms 010 - 2ms 011 - 5ms 100 - 10ms 101 - 20ms 110 - 40ms 111 - 200ms
HDCP_1P1_DIS		R/W
0xECAE	0<u>0</u>000000	This bit is used to force the part into HDCP 1.0 mode - even if the RX side supports HDCP 1.1. 0 - enabled 1 - disabled
R0_WAIT_TIME[1:0]		R/W
0xECAE	00<u>00</u>0000	This signal is used to control the programmable HDCP Ro' read delay. 00 - 100ms 01 - 110ms 10 - 150ms 11 - 200ms
HDCP_REPEATER_TIMEOUT[1:0]		R/W
0xECAE	0000<u>00</u>00	This signal is used to configure the HDMI Tx repeater Ready bit watchdog timer. 00 - 5s 01 - 6.5s 10 - 8s 11 - 10s

Reg	Bits	Description	
HDCP_DESIRED			R/W
0xECAF	00010100	This bit is used to request HDCP encryption. 0 - Input audio and video content not to be encrypted 1 - The input audio and video content should be encrypted	
FRAME_ENC			R/W
0xECAF	00010100	This bit is used to request HDCP frame encryption. 0 - The current video frame should not be encrypted 1 - The current video frame should be encrypted	
HDMI_DVI_SEL_EN			R/W
0xECAF	00010100	This bit is used to enable the output mode control. 0 - Automatic 1 - Output mode set by hdmi_dvi_sel	
HDMI_DVI_SEL			R/W
0xECAF	00010100	This bit is used to control the output mode - DVI or HDMI. 0 - DVI 1 - HDMI	
AN[63:0]			R
0xECB0 0xECB1 0xECB2 0xECB3 0xECB4 0xECB5 0xECB6 0xECB7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This register is used to readback byte 0 of An/AKSV. This register is used to readback byte 1 of An/AKSV. This register is used to readback byte 2 of An/AKSV. This register is used to readback byte 3 of An/AKSV. This register is used to readback byte 4 of An/AKSV. This register is used to readback byte 5 of An/AKSV. This register is used to readback byte 6 of An/AKSV. This register is used to readback byte 7 of An/AKSV.	
ENC_ON			R
0xECB8	00000000	This bit is used to readback the HDCP encryption status. 0 - The audio and video content is not being encrypted 1 - The audio and video content is being encrypted	
KEYS_READ_ERROR			R
0xECB8	00000000	This bit is used to readback the HDCP Key Reading Error status. 0 - The HDCP key has been read successfully 1 - The HDCP key was not read successfully	
BCAPS[7:0]			R
0xECBE	00000000	This register is used to readback the BCAPS value. BCAPS[6] indicates Repeater, BCAPS[5] indicates BKSV FIFO ready, BCAPS[4] indicates Fast DDC bus, BCAPS[1] indicates HDMI 1.1, BCAPS[0] indicates Fast Re-authentication, See the HDCP specification.	
BKSV[7:0]			R
0xECBF	00000000	This register is used to readback the BKSV Byte 0 read from the downstream receiver by the HDCP controller.	
BKSV[15:8]			R
0xECC0	00000000	This register is used to readback the BKSV Byte 1 read from the downstream receiver by the HDCP controller.	
BKSV[23:16]			R
0xECC1	00000000	This register is used to readback the BKSV Byte 2 read from the downstream receiver by the HDCP controller.	
BKSV[31:24]			R
0xECC2	00000000	This register is used to readback the BKSV Byte 3 read from the downstream receiver by the HDCP controller.	

Reg	Bits	Description	
BKSV[39:32]			R
0xECC3	00000000	This register is used to readback the BKSV Byte 4 read from the downstream receiver by the HDCP controller.	
EDID_SEGMENT[7:0]			R/W
0xECC4	00000000	This register is used to set the segment of the EDID read from the downstream receiver. xxxxxxxx - User programmed EDID segment value	
AN_STOP			R
0xECC5	0000000	This bit is set high to get An. Set low again if authentication is restarted.	
RI_FLAG			R
0xECC5	000000	This bit is set high on positive edge of ri_update from HDCP engine. Reset by the firmware.	
BKSV_UPDATE_FLAG			R
0xECC5	00000	This bit indicates to the HDCP engine that a new BKSV has been received.	
PJ_FLAG			R
0xECC5	0000000	This bit is set high on positive edge of pj_update from HDCP engine. Reset by the firmware. 0 - No flag 1 - Flag	
HDMI_MODE			R
0xECC6	0000	This bit is used to readback the HDMI mode status. 0 - DVI 1 - HDMI	
HDCP_REQUESTED			R
0xECC6	0000	This bit is used to readback the HDCP Request status. This bit shows if a HDCP request has been accepted. 0 - No HDCP request 1 - HDCP has been requested	
RX_SENSE			R
0xECC6	000000	This bit is used to readback the Rx sense status. 0 - TMDS clock pull-up not present 1 - TMDS clock pull-up present	
TMDS_OUTPUT_EN			R
0xECC6	0000000	This bit is used to enable TMDS output. 0 - Enable 1 - Disable	
BKSV_FLAG			R/W
0xECC7	0000000	This bit is used to indicate that KSVs from a downstream device have been read. 0 - Not detected 1 - Detected	
BKSV_COUNT[6:0]			R
0xECC7	0000000	This signal is used to specify the total number of downstream HDCP devices. xxxxxxx - Total number of downstream HDCP devices	
HDCP_CONTROLLER_ERROR[3:0]			R
0xECC8	0000	This signal is used to readback the error code when the HDCP controller error interrupt HDCP_ERROR_INT is 1. 0000 - No error 0001 - Bad receiver BKSV 0010 - Ri Mismatch 0011 - Pj Mismatch 0100 - I2C error (usually no acknowledge) 0101 - Timed out waiting for downstream repeater 0110 - Maximum cascade of repeaters exceeded 0111 - SHA-1 Hash check of KSV list fail	

Reg	Bits	Description	
HDCP_CONTROLLER_STATE[3:0]			R
0xECC8	00000000	This signal is used to readback the state of the EDID/HDCP controller. 0000 - In Reset (No Hot Plug Detected) 0001 - Reading EDID 0010 - In Idle state (Waiting for HDCP Request) 0011 - Initializing HDCP 0100 - HDCP enabled 0101 - Initializing HDCP Repeater 0110 - 1111 - Reserved	
EDID_REREAD			R/W
0xECC9	00000011	This bit is used to request a the EDID controller to reread the current segment if toggled from 0 to 1 for 10 times consecutively. 0 - No action 1 - Request the EDID/HDCP controller to read the EDID	
EDID_TRIES[3:0]			R/W
0xECC9	00000011	This signal is used to control the number of times that the EDID read will be attempted if unsuccessful. xxxx - Number of time the EDID/HDCP controller attempts to read the EDID	
HDCP_BSTATUS[15:0]			R
0xECCA 0xECCB	00000000 00000000	This signal is used to readback the BSTATUS information for HDCP.	
PLL_LOCK_STATUS			R
0xECE4	00000000	This bit is used to readback the video PLL lock status. 0 - PLL Not Locked 1 - PLL Locked	
RX_SENSE_PD			R/W
0xECE6	00000000	This bit is used to enable the termination sense power down. 0 - Termination Sense Monitoring Enabled 1 - Termination Sense Monitoring Disabled	
TMDS_CLK_INVERT			R/W
0xECEA	10000100	This bit is used to control the inversion of the TMDS clock. 0 - Normal TMDS Clock 1 - Inverted TMDS Clock	
CCI_CONTROLS			R/W
0xECEA	10000100	This bit is used to control the cross-coupled inverters used to correct for duty cycle distortion. 0 - Disable cross-coupled inverters 1 - Enable cross-coupled inverters for duty cycle	

2.8 TX1 PACKET MAP

Reg	Bits	Description	
SPD_HEADER_BYTE_0[7:0]			R/W
0xF200	00000000	This register is used to specify SPD Infoframe Header, byte 0.	
SPD_HEADER_BYTE_1[7:0]			R/W
0xF201	00000000	This register is used to specify SPD Infoframe Header, byte 1.	
SPD_HEADER_BYTE_2[7:0]			R/W
0xF202	00000000	This register is used to specify SPD Infoframe Header, byte 2.	
SPD_PACKET_BYTE_0[7:0]			R/W
0xF203	00000000	This register is used to specify SPD Infoframe Packet, byte 0.	
SPD_PACKET_BYTE_1[7:0]			R/W
0xF204	00000000	This register is used to specify SPD Infoframe Packet, byte 1.	
SPD_PACKET_BYTE_2[7:0]			R/W
0xF205	00000000	This register is used to specify SPD Infoframe Packet, byte 2.	
SPD_PACKET_BYTE_3[7:0]			R/W
0xF206	00000000	This register is used to specify SPD Infoframe Packet, byte 3.	
SPD_PACKET_BYTE_4[7:0]			R/W
0xF207	00000000	This register is used to specify SPD Infoframe Packet, byte 4.	
SPD_PACKET_BYTE_5[7:0]			R/W
0xF208	00000000	This register is used to specify SPD Infoframe Packet, byte 5.	
SPD_PACKET_BYTE_6[7:0]			R/W
0xF209	00000000	This register is used to specify SPD Infoframe Packet, byte 6.	
SPD_PACKET_BYTE_7[7:0]			R/W
0xF20A	00000000	This register is used to specify SPD Infoframe Packet, byte 7.	
SPD_PACKET_BYTE_8[7:0]			R/W
0xF20B	00000000	This register is used to specify SPD Infoframe Packet, byte 8.	
SPD_PACKET_BYTE_9[7:0]			R/W
0xF20C	00000000	This register is used to specify SPD Infoframe Packet, byte 9.	
SPD_PACKET_BYTE_10[7:0]			R/W
0xF20D	00000000	This register is used to specify SPD Infoframe Packet, byte 10.	
SPD_PACKET_BYTE_11[7:0]			R/W
0xF20E	00000000	This register is used to specify SPD Infoframe Packet, byte 11.	
SPD_PACKET_BYTE_12[7:0]			R/W
0xF20F	00000000	This register is used to specify SPD Infoframe Packet, byte 12.	
SPD_PACKET_BYTE_13[7:0]			R/W
0xF210	00000000	This register is used to specify SPD Infoframe Packet, byte 13.	

Reg	Bits	Description	
SPD_PACKET_BYTE_14[7:0]			R/W
0xF211	00000000	This register is used to specify SPD Infoframe Packet, byte 14.	
SPD_PACKET_BYTE_15[7:0]			R/W
0xF212	00000000	This register is used to specify SPD Infoframe Packet, byte 15.	
SPD_PACKET_BYTE_16[7:0]			R/W
0xF213	00000000	This register is used to specify SPD Infoframe Packet, byte 16.	
SPD_PACKET_BYTE_17[7:0]			R/W
0xF214	00000000	This register is used to specify SPD Infoframe Packet, byte 17.	
SPD_PACKET_BYTE_18[7:0]			R/W
0xF215	00000000	This register is used to specify SPD Infoframe Packet, byte 18.	
SPD_PACKET_BYTE_19[7:0]			R/W
0xF216	00000000	This register is used to specify SPD Infoframe Packet, byte 19.	
SPD_PACKET_BYTE_20[7:0]			R/W
0xF217	00000000	This register is used to specify SPD Infoframe Packet, byte 20.	
SPD_PACKET_BYTE_21[7:0]			R/W
0xF218	00000000	This register is used to specify SPD Infoframe Packet, byte 21.	
SPD_PACKET_BYTE_22[7:0]			R/W
0xF219	00000000	This register is used to specify SPD Infoframe Packet, byte 22.	
SPD_PACKET_BYTE_23[7:0]			R/W
0xF21A	00000000	This register is used to specify SPD Infoframe Packet, byte 23.	
SPD_PACKET_BYTE_24[7:0]			R/W
0xF21B	00000000	This register is used to specify SPD Infoframe Packet, byte 24.	
SPD_PACKET_BYTE_25[7:0]			R/W
0xF21C	00000000	This register is used to specify SPD Infoframe Packet, byte 25.	
SPD_PACKET_BYTE_26[7:0]			R/W
0xF21D	00000000	This register is used to specify SPD Infoframe Packet, byte 26.	
SPD_PACKET_BYTE_27[7:0]			R/W
0xF21E	00000000	This register is used to specify SPD Infoframe Packet, byte 27.	
SPD_UPDATE			R/W
0xF21F	00000000	This bit is used to control the SPD Infoframe protection feature. 0 - Set this bit to 0 when the update of the SDP InfoFrame via I2C is complete. 1 - Set this bit to 1 before updating the SDP InfoFrame data via I2C. If this bit is set high while the Tx is transmitting a SDP InfoFrame during the vsync region, this SDP InfoFrame data transmitted will not be affected by the SDP InfoFrame data programmed via I2C. The SDP InfoFrame data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPD_UPDATE being set back to 0.	
MPEG_HEADER_BYTE_0[7:0]			R/W
0xF220	00000000	This register is used to specify MPEG Infoframe Header, byte 0.	
MPEG_HEADER_BYTE_1[7:0]			R/W
0xF221	00000000	This register is used to specify MPEG Infoframe Header, byte 1.	

Reg	Bits	Description	
MPEG_HEADER_BYTE_2[7:0]			R/W
0xF222	00000000	This register is used to specify MPEG Infoframe Header, byte 2.	
MPEG_PACKET_BYTE_0[7:0]			R/W
0xF223	00000000	This register is used to specify MPEG Infoframe Packet, byte 0.	
MPEG_PACKET_BYTE_1[7:0]			R/W
0xF224	00000000	This register is used to specify MPEG Infoframe Packet, byte 1.	
MPEG_PACKET_BYTE_2[7:0]			R/W
0xF225	00000000	This register is used to specify MPEG Infoframe Packet, byte 2.	
MPEG_PACKET_BYTE_3[7:0]			R/W
0xF226	00000000	This register is used to specify MPEG Infoframe Packet, byte 3.	
MPEG_PACKET_BYTE_4[7:0]			R/W
0xF227	00000000	This register is used to specify MPEG Infoframe Packet, byte 4.	
MPEG_PACKET_BYTE_5[7:0]			R/W
0xF228	00000000	This register is used to specify MPEG Infoframe Packet, byte 5.	
MPEG_PACKET_BYTE_6[7:0]			R/W
0xF229	00000000	This register is used to specify MPEG Infoframe Packet, byte 6.	
MPEG_PACKET_BYTE_7[7:0]			R/W
0xF22A	00000000	This register is used to specify MPEG Infoframe Packet, byte 7.	
MPEG_PACKET_BYTE_8[7:0]			R/W
0xF22B	00000000	This register is used to specify MPEG Infoframe Packet, byte 8.	
MPEG_PACKET_BYTE_9[7:0]			R/W
0xF22C	00000000	This register is used to specify MPEG Infoframe Packet, byte 9.	
MPEG_PACKET_BYTE_10[7:0]			R/W
0xF22D	00000000	This register is used to specify MPEG Infoframe Packet, byte 10.	
MPEG_PACKET_BYTE_11[7:0]			R/W
0xF22E	00000000	This register is used to specify MPEG Infoframe Packet, byte 11.	
MPEG_PACKET_BYTE_12[7:0]			R/W
0xF22F	00000000	This register is used to specify MPEG Infoframe Packet, byte 12.	
MPEG_PACKET_BYTE_13[7:0]			R/W
0xF230	00000000	This register is used to specify MPEG Infoframe Packet, byte 13.	
MPEG_PACKET_BYTE_14[7:0]			R/W
0xF231	00000000	This register is used to specify MPEG Infoframe Packet, byte 14.	
MPEG_PACKET_BYTE_15[7:0]			R/W
0xF232	00000000	This register is used to specify MPEG Infoframe Packet, byte 15.	
MPEG_PACKET_BYTE_16[7:0]			R/W
0xF233	00000000	This register is used to specify MPEG Infoframe Packet, byte 16.	

Reg	Bits	Description	
MPEG_PACKET_BYTE_17[7:0]			R/W
0xF234	00000000	This register is used to specify MPEG Infoframe Packet, byte 17.	
MPEG_PACKET_BYTE_18[7:0]			R/W
0xF235	00000000	This register is used to specify MPEG Infoframe Packet, byte 18.	
MPEG_PACKET_BYTE_19[7:0]			R/W
0xF236	00000000	This register is used to specify MPEG Infoframe Packet, byte 19.	
MPEG_PACKET_BYTE_20[7:0]			R/W
0xF237	00000000	This register is used to specify MPEG Infoframe Packet, byte 20.	
MPEG_PACKET_BYTE_21[7:0]			R/W
0xF238	00000000	This register is used to specify MPEG Infoframe Packet, byte 21.	
MPEG_PACKET_BYTE_22[7:0]			R/W
0xF239	00000000	This register is used to specify MPEG Infoframe Packet, byte 22.	
MPEG_PACKET_BYTE_23[7:0]			R/W
0xF23A	00000000	This register is used to specify MPEG Infoframe Packet, byte 23.	
MPEG_PACKET_BYTE_24[7:0]			R/W
0xF23B	00000000	This register is used to specify MPEG Infoframe Packet, byte 24.	
MPEG_PACKET_BYTE_25[7:0]			R/W
0xF23C	00000000	This register is used to specify MPEG Infoframe Packet, byte 25.	
MPEG_PACKET_BYTE_26[7:0]			R/W
0xF23D	00000000	This register is used to specify MPEG Infoframe Packet, byte 26.	
MPEG_PACKET_BYTE_27[7:0]			R/W
0xF23E	00000000	This register is used to specify MPEG Infoframe Packet, byte 27.	
MPEG_UPDATE			R/W
0xF23F	00000000	This bit is used to control the MPEG Infoframe protection feature. 0 - Set this bit to 0 when the update of the MPEG InfoFrame via I2C is complete. 1 - Set this bit to 1 before updating the MPEG InfoFrame data via I2C. If this bit is set high while the Tx is transmitting a MPEG InfoFrame during the vsync region, this MPEG InfoFrame data transmitted will not be affected by the MPEG InfoFrame data programmed via I2C. The MPEG InfoFrame data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following MPEG_UPDATE being set back to 0.	
ACP_HEADER_BYTE_0[7:0]			R/W
0xF240	00000000	This register is used to specify ACP Infoframe Header, byte 0.	
ACP_HEADER_BYTE_1[7:0]			R/W
0xF241	00000000	This register is used to specify ACP Infoframe Header, byte 1.	
ACP_HEADER_BYTE_2[7:0]			R/W
0xF242	00000000	This register is used to specify ACP Infoframe Header, byte 2.	
ACP_PACKET_BYTE_0[7:0]			R/W
0xF243	00000000	This register is used to specify ACP Infoframe Packet, byte 0.	
ACP_PACKET_BYTE_1[7:0]			R/W
0xF244	00000000	This register is used to specify ACP Infoframe Packet, byte 1.	

Reg	Bits	Description	
ACP_PACKET_BYTE_2[7:0]			R/W
0xF245	00000000	This register is used to specify ACP Infoframe Packet, byte 2.	
ACP_PACKET_BYTE_3[7:0]			R/W
0xF246	00000000	This register is used to specify ACP Infoframe Packet, byte 3.	
ACP_PACKET_BYTE_4[7:0]			R/W
0xF247	00000000	This register is used to specify ACP Infoframe Packet, byte 4.	
ACP_PACKET_BYTE_5[7:0]			R/W
0xF248	00000000	This register is used to specify ACP Infoframe Packet, byte 5.	
ACP_PACKET_BYTE_6[7:0]			R/W
0xF249	00000000	This register is used to specify ACP Infoframe Packet, byte 6.	
ACP_PACKET_BYTE_7[7:0]			R/W
0xF24A	00000000	This register is used to specify ACP Infoframe Packet, byte 7.	
ACP_PACKET_BYTE_8[7:0]			R/W
0xF24B	00000000	This register is used to specify ACP Infoframe Packet, byte 8.	
ACP_PACKET_BYTE_9[7:0]			R/W
0xF24C	00000000	This register is used to specify ACP Infoframe Packet, byte 9.	
ACP_PACKET_BYTE_10[7:0]			R/W
0xF24D	00000000	This register is used to specify ACP Infoframe Packet, byte 10.	
ACP_PACKET_BYTE_11[7:0]			R/W
0xF24E	00000000	This register is used to specify ACP Infoframe Packet, byte 11.	
ACP_PACKET_BYTE_12[7:0]			R/W
0xF24F	00000000	This register is used to specify ACP Infoframe Packet, byte 12.	
ACP_PACKET_BYTE_13[7:0]			R/W
0xF250	00000000	This register is used to specify ACP Infoframe Packet, byte 13.	
ACP_PACKET_BYTE_14[7:0]			R/W
0xF251	00000000	This register is used to specify ACP Infoframe Packet, byte 14.	
ACP_PACKET_BYTE_15[7:0]			R/W
0xF252	00000000	This register is used to specify ACP Infoframe Packet, byte 15.	
ACP_PACKET_BYTE_16[7:0]			R/W
0xF253	00000000	This register is used to specify ACP Infoframe Packet, byte 16.	
ACP_PACKET_BYTE_17[7:0]			R/W
0xF254	00000000	This register is used to specify ACP Infoframe Packet, byte 17.	
ACP_PACKET_BYTE_18[7:0]			R/W
0xF255	00000000	This register is used to specify ACP Infoframe Packet, byte 18.	
ACP_PACKET_BYTE_19[7:0]			R/W
0xF256	00000000	This register is used to specify ACP Infoframe Packet, byte 19.	

Reg	Bits	Description	
ACP_PACKET_BYTE_20[7:0]			R/W
0xF257	00000000	This register is used to specify ACP Infoframe Packet, byte 20.	
ACP_PACKET_BYTE_21[7:0]			R/W
0xF258	00000000	This register is used to specify ACP Infoframe Packet, byte 21.	
ACP_PACKET_BYTE_22[7:0]			R/W
0xF259	00000000	This register is used to specify ACP Infoframe Packet, byte 22.	
ACP_PACKET_BYTE_23[7:0]			R/W
0xF25A	00000000	This register is used to specify ACP Infoframe Packet, byte 23.	
ACP_PACKET_BYTE_24[7:0]			R/W
0xF25B	00000000	This register is used to specify ACP Infoframe Packet, byte 24.	
ACP_PACKET_BYTE_25[7:0]			R/W
0xF25C	00000000	This register is used to specify ACP Infoframe Packet, byte 25.	
ACP_PACKET_BYTE_26[7:0]			R/W
0xF25D	00000000	This register is used to specify ACP Infoframe Packet, byte 26.	
ACP_PACKET_BYTE_27[7:0]			R/W
0xF25E	00000000	This register is used to specify ACP Infoframe Packet, byte 27.	
ACP_UPDATE			R/W
0xF25F	0	This bit is used to control the ACP Infoframe protection feature. 0 - Set this bit to 0 when the update of the ACP packet via I2C is complete. 1 - Set this bit to 1 before updating the ACP packet data via I2C. If this bit is set high while the Tx is transmitting an ACP packet during the vsync region, this ACP packet data transmitted will not be affected by the ACP packet data programmed via I2C. The ACP packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ACP_UPDATE being set back to 0	
ISRC1_HEADER_BYTE_0[7:0]			R/W
0xF260	00000000	This register is used to specify ISRC1 Infoframe Header, byte 0.	
ISRC1_HEADER_BYTE_1[7:0]			R/W
0xF261	00000000	This register is used to specify ISRC1 Infoframe Header, byte 1.	
ISRC1_HEADER_BYTE_2[7:0]			R/W
0xF262	00000000	This register is used to specify ISRC1 Infoframe Header, byte 2.	
ISRC1_PACKET_BYTE_0[7:0]			R/W
0xF263	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 0.	
ISRC1_PACKET_BYTE_1[7:0]			R/W
0xF264	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 1.	
ISRC1_PACKET_BYTE_2[7:0]			R/W
0xF265	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 2.	
ISRC1_PACKET_BYTE_3[7:0]			R/W
0xF266	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 3.	
ISRC1_PACKET_BYTE_4[7:0]			R/W
0xF267	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 4.	

Reg	Bits	Description	
ISRC1_PACKET_BYTE_5[7:0]			R/W
0xF268	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 5.	
ISRC1_PACKET_BYTE_6[7:0]			R/W
0xF269	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 6.	
ISRC1_PACKET_BYTE_7[7:0]			R/W
0xF26A	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 7.	
ISRC1_PACKET_BYTE_8[7:0]			R/W
0xF26B	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 8.	
ISRC1_PACKET_BYTE_9[7:0]			R/W
0xF26C	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 9.	
ISRC1_PACKET_BYTE_10[7:0]			R/W
0xF26D	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 10.	
ISRC1_PACKET_BYTE_11[7:0]			R/W
0xF26E	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 11.	
ISRC1_PACKET_BYTE_12[7:0]			R/W
0xF26F	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 12.	
ISRC1_PACKET_BYTE_13[7:0]			R
0xF270	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 13.	
ISRC1_PACKET_BYTE_14[7:0]			R
0xF271	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 14.	
ISRC1_PACKET_BYTE_15[7:0]			R
0xF272	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 15.	
ISRC1_PACKET_BYTE_16[7:0]			R
0xF273	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 16.	
ISRC1_PACKET_BYTE_17[7:0]			R
0xF274	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 17.	
ISRC1_PACKET_BYTE_18[7:0]			R/W
0xF275	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 18.	
ISRC1_PACKET_BYTE_19[7:0]			R/W
0xF276	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 19.	
ISRC1_PACKET_BYTE_20[7:0]			R/W
0xF277	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 20.	
ISRC1_PACKET_BYTE_21[7:0]			R/W
0xF278	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 21.	
ISRC1_PACKET_BYTE_22[7:0]			R/W
0xF279	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 22.	

Reg	Bits	Description	
ISRC1_PACKET_BYTE_23[7:0]			R/W
0xF27A	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 23.	
ISRC1_PACKET_BYTE_24[7:0]			R/W
0xF27B	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 24.	
ISRC1_PACKET_BYTE_25[7:0]			R/W
0xF27C	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 25.	
ISRC1_PACKET_BYTE_26[7:0]			R/W
0xF27D	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 26.	
ISRC1_PACKET_BYTE_27[7:0]			R/W
0xF27E	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 27.	
ISRC1_UPDATE			R/W
0xF27F	00000000	This bit is used to control the ISRC1 Infoframe protection feature. 0 - Set this bit to 0 when the update of the ISRC1 packet via I2C is complete. 1 - Set this bit to 1 before updating the ISRC1 packet data via I2C. If this bit is set high while the Tx is transmitting an ISRC1 packet during the vsync region, this ISRC1 packet data transmitted will not be affected by the ISRC1 packet data programmed via I2C. The ISRC1 packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ISRC1_UPDATE being set back to 0	
ISRC2_HEADER_BYTE_0[7:0]			R/W
0xF280	00000000	This register is used to specify ISRC2 Infoframe Header, byte 0.	
ISRC2_HEADER_BYTE_1[7:0]			R/W
0xF281	00000000	This register is used to specify ISRC2 Infoframe Header, byte 1.	
ISRC2_HEADER_BYTE_2[7:0]			R/W
0xF282	00000000	This register is used to specify ISRC2 Infoframe Header, byte 2.	
ISRC2_PACKET_BYTE_0[7:0]			R/W
0xF283	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 0.	
ISRC2_PACKET_BYTE_1[7:0]			R/W
0xF284	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 1.	
ISRC2_PACKET_BYTE_2[7:0]			R/W
0xF285	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 2.	
ISRC2_PACKET_BYTE_3[7:0]			R/W
0xF286	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 3.	
ISRC2_PACKET_BYTE_4[7:0]			R/W
0xF287	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 4.	
ISRC2_PACKET_BYTE_5[7:0]			R/W
0xF288	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 5.	
ISRC2_PACKET_BYTE_6[7:0]			R/W
0xF289	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 6.	
ISRC2_PACKET_BYTE_7[7:0]			R/W
0xF28A	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 7.	

Reg	Bits	Description	
ISRC2_PACKET_BYTE_8[7:0]			R/W
0xF28B	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 8.	
ISRC2_PACKET_BYTE_9[7:0]			R/W
0xF28C	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 9.	
ISRC2_PACKET_BYTE_10[7:0]			R/W
0xF28D	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 10.	
ISRC2_PACKET_BYTE_11[7:0]			R/W
0xF28E	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 11.	
ISRC2_PACKET_BYTE_12[7:0]			R/W
0xF28F	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 12.	
ISRC2_PACKET_BYTE_13[7:0]			R/W
0xF290	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 13.	
ISRC2_PACKET_BYTE_14[7:0]			R/W
0xF291	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 14.	
ISRC2_PACKET_BYTE_15[7:0]			R/W
0xF292	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 15.	
ISRC2_PACKET_BYTE_16[7:0]			R/W
0xF293	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 16.	
ISRC2_PACKET_BYTE_17[7:0]			R/W
0xF294	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 17.	
ISRC2_PACKET_BYTE_18[7:0]			R/W
0xF295	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 18.	
ISRC2_PACKET_BYTE_19[7:0]			R/W
0xF296	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 19.	
ISRC2_PACKET_BYTE_20[7:0]			R/W
0xF297	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 20.	
ISRC2_PACKET_BYTE_21[7:0]			R/W
0xF298	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 21.	
ISRC2_PACKET_BYTE_22[7:0]			R/W
0xF299	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 22.	
ISRC2_PACKET_BYTE_23[7:0]			R/W
0xF29A	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 23.	
ISRC2_PACKET_BYTE_24[7:0]			R/W
0xF29B	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 24.	
ISRC2_PACKET_BYTE_25[7:0]			R/W
0xF29C	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 25.	

Reg	Bits	Description	
ISRC2_PACKET_BYTE_26[7:0]			R/W
0xF29D	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 26.	
ISRC2_PACKET_BYTE_27[7:0]			R/W
0xF29E	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 27.	
ISRC2_UPDATE			R/W
0xF29F	00000000	This bit is used to control the ISRC2 Infoframe protection feature. 0 - Set this bit to 0 when the update of the ISRC2 packet via I2C is complete. 1 - Set this bit to 1 before updating the ISRC2 packet data via I2C. If this bit is set high while the Tx is transmitting an ISRC2 packet during the vsync region, this ISRC2 packet data transmitted will not be affected by the ISRC2 packet data programmed via I2C. The ISRC2 packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ISRC2_UPDATE being set back to 0	
GM_HEADER_BYTE_0[7:0]			R/W
0xF2A0	00000000	This register is used to specify GM Infoframe Header, byte 0.	
GM_HEADER_BYTE_1[7:0]			R/W
0xF2A1	00000000	This register is used to specify GM Infoframe Header, byte 1.	
GM_HEADER_BYTE_2[7:0]			R/W
0xF2A2	00000000	This register is used to specify GM Infoframe Header, byte 2.	
GM_PACKET_BYTE_0[7:0]			R/W
0xF2A3	00000000	This register is used to specify GM Infoframe Packet, byte 0.	
GM_PACKET_BYTE_1[7:0]			R/W
0xF2A4	00000000	This register is used to specify GM Infoframe Packet, byte 1.	
GM_PACKET_BYTE_2[7:0]			R/W
0xF2A5	00000000	This register is used to specify GM Infoframe Packet, byte 2.	
GM_PACKET_BYTE_3[7:0]			R/W
0xF2A6	00000000	This register is used to specify GM Infoframe Packet, byte 3.	
GM_PACKET_BYTE_4[7:0]			R/W
0xF2A7	00000000	This register is used to specify GM Infoframe Packet, byte 4.	
GM_PACKET_BYTE_5[7:0]			R/W
0xF2A8	00000000	This register is used to specify GM Infoframe Packet, byte 5.	
GM_PACKET_BYTE_6[7:0]			R/W
0xF2A9	00000000	This register is used to specify GM Infoframe Packet, byte 6.	
GM_PACKET_BYTE_7[7:0]			R/W
0xF2AA	00000000	This register is used to specify GM Infoframe Packet, byte 7.	
GM_PACKET_BYTE_8[7:0]			R/W
0xF2AB	00000000	This register is used to specify GM Infoframe Packet, byte 8.	
GM_PACKET_BYTE_9[7:0]			R/W
0xF2AC	00000000	This register is used to specify GM Infoframe Packet, byte 9.	
GM_PACKET_BYTE_10[7:0]			R/W
0xF2AD	00000000	This register is used to specify GM Infoframe Packet, byte 10.	

Reg	Bits	Description	
GM_PACKET_BYTE_11[7:0]			R/W
0xF2AE	00000000	This register is used to specify GM Infoframe Packet, byte 11.	
GM_PACKET_BYTE_12[7:0]			R/W
0xF2AF	00000000	This register is used to specify GM Infoframe Packet, byte 12.	
GM_PACKET_BYTE_13[7:0]			R/W
0xF2B0	00000000	This register is used to specify GM Infoframe Packet, byte 13.	
GM_PACKET_BYTE_14[7:0]			R/W
0xF2B1	00000000	This register is used to specify GM Infoframe Packet, byte 14.	
GM_PACKET_BYTE_15[7:0]			R/W
0xF2B2	00000000	This register is used to specify GM Infoframe Packet, byte 15.	
GM_PACKET_BYTE_16[7:0]			R/W
0xF2B3	00000000	This register is used to specify GM Infoframe Packet, byte 16.	
GM_PACKET_BYTE_17[7:0]			R/W
0xF2B4	00000000	This register is used to specify GM Infoframe Packet, byte 17.	
GM_PACKET_BYTE_18[7:0]			R/W
0xF2B5	00000000	This register is used to specify GM Infoframe Packet, byte 18.	
GM_PACKET_BYTE_19[7:0]			R/W
0xF2B6	00000000	This register is used to specify GM Infoframe Packet, byte 19.	
GM_PACKET_BYTE_20[7:0]			R/W
0xF2B7	00000000	This register is used to specify GM Infoframe Packet, byte 20.	
GM_PACKET_BYTE_21[7:0]			R/W
0xF2B8	00000000	This register is used to specify GM Infoframe Packet, byte 21.	
GM_PACKET_BYTE_22[7:0]			R/W
0xF2B9	00000000	This register is used to specify GM Infoframe Packet, byte 22.	
GM_PACKET_BYTE_23[7:0]			R/W
0xF2BA	00000000	This register is used to specify GM Infoframe Packet, byte 23.	
GM_PACKET_BYTE_24[7:0]			R/W
0xF2BB	00000000	This register is used to specify GM Infoframe Packet, byte 24.	
GM_PACKET_BYTE_25[7:0]			R/W
0xF2BC	00000000	This register is used to specify GM Infoframe Packet, byte 25.	
GM_PACKET_BYTE_26[7:0]			R/W
0xF2BD	00000000	This register is used to specify GM Infoframe Packet, byte 26.	
GM_PACKET_BYTE_27[7:0]			R/W
0xF2BE	00000000	This register is used to specify GM Infoframe Packet, byte 27.	

Reg	Bits	Description	
GM_UPDATE			R/W
0xF2BF	00000000	This bit is used to control the GM Infoframe protection feature. 0 - Set this bit to 0 when the update of the Gamut Metadata packet via I2C is complete. 1 - Set this bit to 1 before updating the Gamut Metadata packet data via I2C. If this bit is set high while the Tx is transmitting a Gamut Metadata packet during the vsync region, this Gamut Metadata packet data transmitted will not be affected by the Gamut Metadata packet data programmed via I2C. The Gamut Metadata packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following GMP_UPDATE being set back to 0	
SPARE_PACKET_1_HEADER_BYTE_0[7:0]			R/W
0xF2C0	00000000	This register is used to specify Spare1 Infoframe Header, byte 0.	
SPARE_PACKET_1_HEADER_BYTE_1[7:0]			R/W
0xF2C1	00000000	This register is used to specify Spare1 Infoframe Header, byte 1.	
SPARE_PACKET_1_HEADER_BYTE_2[7:0]			R/W
0xF2C2	00000000	This register is used to specify Spare1 Infoframe Header, byte 2.	
SPARE_PACKET_1_PACKET_BYTE_0[7:0]			R/W
0xF2C3	00000000	This register is used to specify Spare1 Infoframe Packet, byte 0.	
SPARE_PACKET_1_PACKET_BYTE_1[7:0]			R/W
0xF2C4	00000000	This register is used to specify Spare1 Infoframe Packet, byte 1.	
SPARE_PACKET_1_PACKET_BYTE_2[7:0]			R/W
0xF2C5	00000000	This register is used to specify Spare1 Infoframe Packet, byte 2.	
SPARE_PACKET_1_PACKET_BYTE_3[7:0]			R/W
0xF2C6	00000000	This register is used to specify Spare1 Infoframe Packet, byte 3.	
SPARE_PACKET_1_PACKET_BYTE_4[7:0]			R/W
0xF2C7	00000000	This register is used to specify Spare1 Infoframe Packet, byte 4.	
SPARE_PACKET_1_PACKET_BYTE_5[7:0]			R/W
0xF2C8	00000000	This register is used to specify Spare1 Infoframe Packet, byte 5.	
SPARE_PACKET_1_PACKET_BYTE_6[7:0]			R/W
0xF2C9	00000000	This register is used to specify Spare1 Infoframe Packet, byte 6.	
SPARE_PACKET_1_PACKET_BYTE_7[7:0]			R/W
0xF2CA	00000000	This register is used to specify Spare1 Infoframe Packet, byte 7.	
SPARE_PACKET_1_PACKET_BYTE_8[7:0]			R/W
0xF2CB	00000000	This register is used to specify Spare1 Infoframe Packet, byte 8.	
SPARE_PACKET_1_PACKET_BYTE_9[7:0]			R/W
0xF2CC	00000000	This register is used to specify Spare1 Infoframe Packet, byte 9.	
SPARE_PACKET_1_PACKET_BYTE_10[7:0]			R/W
0xF2CD	00000000	This register is used to specify Spare1 Infoframe Packet, byte 10.	
SPARE_PACKET_1_PACKET_BYTE_11[7:0]			R/W
0xF2CE	00000000	This register is used to specify Spare1 Infoframe Packet, byte 11.	

Reg	Bits	Description	
SPARE_PACKET_1_PACKET_BYTE_12[7:0]			R/W
0xF2CF	00000000	This register is used to specify Spare1 Infoframe Packet, byte 12.	
SPARE_PACKET_1_PACKET_BYTE_13[7:0]			R/W
0xF2D0	00000000	This register is used to specify Spare1 Infoframe Packet, byte 13.	
SPARE_PACKET_1_PACKET_BYTE_14[7:0]			R/W
0xF2D1	00000000	This register is used to specify Spare1 Infoframe Packet, byte 14.	
SPARE_PACKET_1_PACKET_BYTE_15[7:0]			R/W
0xF2D2	00000000	This register is used to specify Spare1 Infoframe Packet, byte 15.	
SPARE_PACKET_1_PACKET_BYTE_16[7:0]			R/W
0xF2D3	00000000	This register is used to specify Spare1 Infoframe Packet, byte 16.	
SPARE_PACKET_1_PACKET_BYTE_17[7:0]			R/W
0xF2D4	00000000	This register is used to specify Spare1 Infoframe Packet, byte 17.	
SPARE_PACKET_1_PACKET_BYTE_18[7:0]			R/W
0xF2D5	00000000	This register is used to specify Spare1 Infoframe Packet, byte 18.	
SPARE_PACKET_1_PACKET_BYTE_19[7:0]			R/W
0xF2D6	00000000	This register is used to specify Spare1 Infoframe Packet, byte 19.	
SPARE_PACKET_1_PACKET_BYTE_20[7:0]			R/W
0xF2D7	00000000	This register is used to specify Spare1 Infoframe Packet, byte 20.	
SPARE_PACKET_1_PACKET_BYTE_21[7:0]			R/W
0xF2D8	00000000	This register is used to specify Spare1 Infoframe Packet, byte 21.	
SPARE_PACKET_1_PACKET_BYTE_22[7:0]			R/W
0xF2D9	00000000	This register is used to specify Spare1 Infoframe Packet, byte 22.	
SPARE_PACKET_1_PACKET_BYTE_23[7:0]			R/W
0xF2DA	00000000	This register is used to specify Spare1 Infoframe Packet, byte 23.	
SPARE_PACKET_1_PACKET_BYTE_24[7:0]			R/W
0xF2DB	00000000	This register is used to specify Spare1 Infoframe Packet, byte 24.	
SPARE_PACKET_1_PACKET_BYTE_25[7:0]			R/W
0xF2DC	00000000	This register is used to specify Spare1 Infoframe Packet, byte 25.	
SPARE_PACKET_1_PACKET_BYTE_26[7:0]			R/W
0xF2DD	00000000	This register is used to specify Spare1 Infoframe Packet, byte 26.	
SPARE_PACKET_1_PACKET_BYTE_27[7:0]			R/W
0xF2DE	00000000	This register is used to specify Spare1 Infoframe Packet, byte 27.	
SPARE1_UPDATE			R/W
0xF2DF	00000000	This bit is used to control the Spare1 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 1 via I2C is complete. 1 - Set this bit to 1 before updating the spare packet 1 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 1 packet during the vsync region, this spare packet 1 data transmitted will not be affected by the spare packet 1 data programmed via I2C. The spare packet 1 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE1_UPDATE being set back to 0	

Reg	Bits	Description	
SPARE_PACKET_2_HEADER_BYTE_0[7:0]			R/W
0xF2E0	00000000	This register is used to specify Spare2 Infoframe Header, byte 0.	
SPARE_PACKET_2_HEADER_BYTE_1[7:0]			R/W
0xF2E1	00000000	This register is used to specify Spare2 Infoframe Header, byte 1.	
SPARE_PACKET_2_HEADER_BYTE_2[7:0]			R/W
0xF2E2	00000000	This register is used to specify Spare2 Infoframe Header, byte 2.	
SPARE_PACKET_2_PACKET_BYTE_0[7:0]			R/W
0xF2E3	00000000	This register is used to specify Spare2 Infoframe Packet, byte 0.	
SPARE_PACKET_2_PACKET_BYTE_1[7:0]			R/W
0xF2E4	00000000	This register is used to specify Spare2 Infoframe Packet, byte 1.	
SPARE_PACKET_2_PACKET_BYTE_2[7:0]			R/W
0xF2E5	00000000	This register is used to specify Spare2 Infoframe Packet, byte 2.	
SPARE_PACKET_2_PACKET_BYTE_3[7:0]			R/W
0xF2E6	00000000	This register is used to specify Spare2 Infoframe Packet, byte 3.	
SPARE_PACKET_2_PACKET_BYTE_4[7:0]			R/W
0xF2E7	00000000	This register is used to specify Spare2 Infoframe Packet, byte 4.	
SPARE_PACKET_2_PACKET_BYTE_5[7:0]			R/W
0xF2E8	00000000	This register is used to specify Spare2 Infoframe Packet, byte 5.	
SPARE_PACKET_2_PACKET_BYTE_6[7:0]			R/W
0xF2E9	00000000	This register is used to specify Spare2 Infoframe Packet, byte 6.	
SPARE_PACKET_2_PACKET_BYTE_7[7:0]			R/W
0xF2EA	00000000	This register is used to specify Spare2 Infoframe Packet, byte 7.	
SPARE_PACKET_2_PACKET_BYTE_8[7:0]			R/W
0xF2EB	00000000	This register is used to specify Spare2 Infoframe Packet, byte 8.	
SPARE_PACKET_2_PACKET_BYTE_9[7:0]			R/W
0xF2EC	00000000	This register is used to specify Spare2 Infoframe Packet, byte 9.	
SPARE_PACKET_2_PACKET_BYTE_10[7:0]			R/W
0xF2ED	00000000	This register is used to specify Spare2 Infoframe Packet, byte 10.	
SPARE_PACKET_2_PACKET_BYTE_11[7:0]			R/W
0xF2EE	00000000	This register is used to specify Spare2 Infoframe Packet, byte 11.	
SPARE_PACKET_2_PACKET_BYTE_12[7:0]			R/W
0xF2EF	00000000	This register is used to specify Spare2 Infoframe Packet, byte 12.	
SPARE_PACKET_2_PACKET_BYTE_13[7:0]			R/W
0xF2F0	00000000	This register is used to specify Spare2 Infoframe Packet, byte 13.	
SPARE_PACKET_2_PACKET_BYTE_14[7:0]			R/W
0xF2F1	00000000	This register is used to specify Spare2 Infoframe Packet, byte 14.	

Reg	Bits	Description	
SPARE_PACKET_2_PACKET_BYTE_15[7:0]			R/W
0xF2F2	00000000	This register is used to specify Spare2 Infoframe Packet, byte 15.	
SPARE_PACKET_2_PACKET_BYTE_16[7:0]			R/W
0xF2F3	00000000	This register is used to specify Spare2 Infoframe Packet, byte 16.	
SPARE_PACKET_2_PACKET_BYTE_17[7:0]			R/W
0xF2F4	00000000	This register is used to specify Spare2 Infoframe Packet, byte 17.	
SPARE_PACKET_2_PACKET_BYTE_18[7:0]			R/W
0xF2F5	00000000	This register is used to specify Spare2 Infoframe Packet, byte 18.	
SPARE_PACKET_2_PACKET_BYTE_19[7:0]			R/W
0xF2F6	00000000	This register is used to specify Spare2 Infoframe Packet, byte 19.	
SPARE_PACKET_2_PACKET_BYTE_20[7:0]			R/W
0xF2F7	00000000	This register is used to specify Spare2 Infoframe Packet, byte 20.	
SPARE_PACKET_2_PACKET_BYTE_21[7:0]			R/W
0xF2F8	00000000	This register is used to specify Spare2 Infoframe Packet, byte 21.	
SPARE_PACKET_2_PACKET_BYTE_22[7:0]			R/W
0xF2F9	00000000	This register is used to specify Spare2 Infoframe Packet, byte 22.	
SPARE_PACKET_2_PACKET_BYTE_23[7:0]			R/W
0xF2FA	00000000	This register is used to specify Spare2 Infoframe Packet, byte 23.	
SPARE_PACKET_2_PACKET_BYTE_24[7:0]			R/W
0xF2FB	00000000	This register is used to specify Spare2 Infoframe Packet, byte 24.	
SPARE_PACKET_2_PACKET_BYTE_25[7:0]			R/W
0xF2FC	00000000	This register is used to specify Spare2 Infoframe Packet, byte 25.	
SPARE_PACKET_2_PACKET_BYTE_26[7:0]			R/W
0xF2FD	00000000	This register is used to specify Spare2 Infoframe Packet, byte 26.	
SPARE_PACKET_2_PACKET_BYTE_27[7:0]			R/W
0xF2FE	00000000	This register is used to specify Spare2 Infoframe Packet, byte 27.	
SPARE2_UPDATE			R/W
0xF2FF	00000000	This bit is used to control the Spare2 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 2 via I2C is complete. 1 - Set this bit to 1 before updating the spare packet 2 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 2 packet during the vsync region, this spare packet 2 data transmitted will not be affected by the spare packet 2 data programmed via I2C. The spare packet 2 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE2_UPDATE being set back to 0	

2.9 TX1 EDID MAP

Reg	Bits	Description	
EDID_0[7:0]			R
0xEE00	00000000		
EDID_1[7:0]			R
0xEE01	00000000		
EDID_2[7:0]			R
0xEE02	00000000		
EDID_3[7:0]			R
0xEE03	00000000		
EDID_4[7:0]			R
0xEE04	00000000		
EDID_5[7:0]			R
0xEE05	00000000		
EDID_6[7:0]			R
0xEE06	00000000		
EDID_7[7:0]			R
0xEE07	00000000		
EDID_8[7:0]			R
0xEE08	00000000		
EDID_9[7:0]			R
0xEE09	00000000		
EDID_10[7:0]			R
0xEE0A	00000000		
EDID_11[7:0]			R
0xEE0B	00000000		
EDID_12[7:0]			R
0xEE0C	00000000		
EDID_13[7:0]			R
0xEE0D	00000000		
EDID_14[7:0]			R
0xEE0E	00000000		
EDID_15[7:0]			R
0xEE0F	00000000		
EDID_16[7:0]			R
0xEE10	00000000		

Reg	Bits	Description	
EDID_17[7:0]			R
0xEE11	00000000		
EDID_18[7:0]			R
0xEE12	00000000		
EDID_19[7:0]			R
0xEE13	00000000		
EDID_20[7:0]			R
0xEE14	00000000		
EDID_21[7:0]			R
0xEE15	00000000		
EDID_22[7:0]			R
0xEE16	00000000		
EDID_23[7:0]			R
0xEE17	00000000		
EDID_24[7:0]			R
0xEE18	00000000		
EDID_25[7:0]			R
0xEE19	00000000		
EDID_26[7:0]			R
0xEE1A	00000000		
EDID_27[7:0]			R
0xEE1B	00000000		
EDID_28[7:0]			R
0xEE1C	00000000		
EDID_29[7:0]			R
0xEE1D	00000000		
EDID_30[7:0]			R
0xEE1E	00000000		
EDID_31[7:0]			R
0xEE1F	00000000		
EDID_32[7:0]			R
0xEE20	00000000		
EDID_33[7:0]			R
0xEE21	00000000		
EDID_34[7:0]			R
0xEE22	00000000		

Reg	Bits	Description	
EDID_35[7:0]			R
0xEE23	00000000		
EDID_36[7:0]			R
0xEE24	00000000		
EDID_37[7:0]			R
0xEE25	00000000		
EDID_38[7:0]			R
0xEE26	00000000		
EDID_39[7:0]			R
0xEE27	00000000		
EDID_40[7:0]			R
0xEE28	00000000		
EDID_41[7:0]			R
0xEE29	00000000		
EDID_42[7:0]			R
0xEE2A	00000000		
EDID_43[7:0]			R
0xEE2B	00000000		
EDID_44[7:0]			R
0xEE2C	00000000		
EDID_45[7:0]			R
0xEE2D	00000000		
EDID_46[7:0]			R
0xEE2E	00000000		
EDID_47[7:0]			R
0xEE2F	00000000		
EDID_48[7:0]			R
0xEE30	00000000		
EDID_49[7:0]			R
0xEE31	00000000		
EDID_50[7:0]			R
0xEE32	00000000		
EDID_51[7:0]			R
0xEE33	00000000		
EDID_52[7:0]			R
0xEE34	00000000		

Reg	Bits	Description	
EDID_53[7:0]			R
0xEE35	00000000		
EDID_54[7:0]			R
0xEE36	00000000		
EDID_55[7:0]			R
0xEE37	00000000		
EDID_56[7:0]			R
0xEE38	00000000		
EDID_57[7:0]			R
0xEE39	00000000		
EDID_58[7:0]			R
0xEE3A	00000000		
EDID_59[7:0]			R
0xEE3B	00000000		
EDID_60[7:0]			R
0xEE3C	00000000		
EDID_61[7:0]			R
0xEE3D	00000000		
EDID_62[7:0]			R
0xEE3E	00000000		
EDID_63[7:0]			R
0xEE3F	00000000		
EDID_64[7:0]			R
0xEE40	00000000		
EDID_65[7:0]			R
0xEE41	00000000		
EDID_66[7:0]			R
0xEE42	00000000		
EDID_67[7:0]			R
0xEE43	00000000		
EDID_68[7:0]			R
0xEE44	00000000		
EDID_69[7:0]			R
0xEE45	00000000		
EDID_70[7:0]			R
0xEE46	00000000		

Reg	Bits	Description	
EDID_71[7:0]			R
0xEE47	00000000		
EDID_72[7:0]			R
0xEE48	00000000		
EDID_73[7:0]			R
0xEE49	00000000		
EDID_74[7:0]			R
0xEE4A	00000000		
EDID_75[7:0]			R
0xEE4B	00000000		
EDID_76[7:0]			R
0xEE4C	00000000		
EDID_77[7:0]			R
0xEE4D	00000000		
EDID_78[7:0]			R
0xEE4E	00000000		
EDID_79[7:0]			R
0xEE4F	00000000		
EDID_80[7:0]			R
0xEE50	00000000		
EDID_81[7:0]			R
0xEE51	00000000		
EDID_82[7:0]			R
0xEE52	00000000		
EDID_83[7:0]			R
0xEE53	00000000		
EDID_84[7:0]			R
0xEE54	00000000		
EDID_85[7:0]			R
0xEE55	00000000		
EDID_86[7:0]			R
0xEE56	00000000		
EDID_87[7:0]			R
0xEE57	00000000		
EDID_88[7:0]			R
0xEE58	00000000		

Reg	Bits	Description	
EDID_89[7:0]			R
0xEE59	00000000		
EDID_90[7:0]			R
0xEE5A	00000000		
EDID_91[7:0]			R
0xEE5B	00000000		
EDID_92[7:0]			R
0xEE5C	00000000		
EDID_93[7:0]			R
0xEE5D	00000000		
EDID_94[7:0]			R
0xEE5E	00000000		
EDID_95[7:0]			R
0xEE5F	00000000		
EDID_96[7:0]			R
0xEE60	00000000		
EDID_97[7:0]			R
0xEE61	00000000		
EDID_98[7:0]			R
0xEE62	00000000		
EDID_99[7:0]			R
0xEE63	00000000		
EDID_100[7:0]			R
0xEE64	00000000		
EDID_101[7:0]			R
0xEE65	00000000		
EDID_102[7:0]			R
0xEE66	00000000		
EDID_103[7:0]			R
0xEE67	00000000		
EDID_104[7:0]			R
0xEE68	00000000		
EDID_105[7:0]			R
0xEE69	00000000		
EDID_106[7:0]			R
0xEE6A	00000000		

Reg	Bits	Description	
EDID_107[7:0]			R
0xEE6B	00000000		
EDID_108[7:0]			R
0xEE6C	00000000		
EDID_109[7:0]			R
0xEE6D	00000000		
EDID_110[7:0]			R
0xEE6E	00000000		
EDID_111[7:0]			R
0xEE6F	00000000		
EDID_112[7:0]			R
0xEE70	00000000		
EDID_113[7:0]			R
0xEE71	00000000		
EDID_114[7:0]			R
0xEE72	00000000		
EDID_115[7:0]			R
0xEE73	00000000		
EDID_116[7:0]			R
0xEE74	00000000		
EDID_117[7:0]			R
0xEE75	00000000		
EDID_118[7:0]			R
0xEE76	00000000		
EDID_119[7:0]			R
0xEE77	00000000		
EDID_120[7:0]			R
0xEE78	00000000		
EDID_121[7:0]			R
0xEE79	00000000		
EDID_122[7:0]			R
0xEE7A	00000000		
EDID_123[7:0]			R
0xEE7B	00000000		
EDID_124[7:0]			R
0xEE7C	00000000		

Reg	Bits	Description	
EDID_125[7:0]			R
0xEE7D	00000000		
EDID_126[7:0]			R
0xEE7E	00000000		
EDID_127[7:0]			R
0xEE7F	00000000		
EDID_128[7:0]			R
0xEE80	00000000		
EDID_129[7:0]			R
0xEE81	00000000		
EDID_130[7:0]			R
0xEE82	00000000		
EDID_131[7:0]			R
0xEE83	00000000		
EDID_132[7:0]			R
0xEE84	00000000		
EDID_133[7:0]			R
0xEE85	00000000		
EDID_134[7:0]			R
0xEE86	00000000		
EDID_135[7:0]			R
0xEE87	00000000		
EDID_136[7:0]			R
0xEE88	00000000		
EDID_137[7:0]			R
0xEE89	00000000		
EDID_138[7:0]			R
0xEE8A	00000000		
EDID_139[7:0]			R
0xEE8B	00000000		
EDID_140[7:0]			R
0xEE8C	00000000		
EDID_141[7:0]			R
0xEE8D	00000000		
EDID_142[7:0]			R
0xEE8E	00000000		

Reg	Bits	Description	
EDID_143[7:0]			R
0xEE8F	00000000		
EDID_144[7:0]			R
0xEE90	00000000		
EDID_145[7:0]			R
0xEE91	00000000		
EDID_146[7:0]			R
0xEE92	00000000		
EDID_147[7:0]			R
0xEE93	00000000		
EDID_148[7:0]			R
0xEE94	00000000		
EDID_149[7:0]			R
0xEE95	00000000		
EDID_150[7:0]			R
0xEE96	00000000		
EDID_151[7:0]			R
0xEE97	00000000		
EDID_152[7:0]			R
0xEE98	00000000		
EDID_153[7:0]			R
0xEE99	00000000		
EDID_154[7:0]			R
0xEE9A	00000000		
EDID_155[7:0]			R
0xEE9B	00000000		
EDID_156[7:0]			R
0xEE9C	00000000		
EDID_157[7:0]			R
0xEE9D	00000000		
EDID_158[7:0]			R
0xEE9E	00000000		
EDID_159[7:0]			R
0xEE9F	00000000		
EDID_160[7:0]			R
0xEEA0	00000000		

Reg	Bits	Description	
EDID_161[7:0]			R
0xEEA1	00000000		
EDID_162[7:0]			R
0xEEA2	00000000		
EDID_163[7:0]			R
0xEEA3	00000000		
EDID_164[7:0]			R
0xEEA4	00000000		
EDID_165[7:0]			R
0xEEA5	00000000		
EDID_166[7:0]			R
0xEEA6	00000000		
EDID_167[7:0]			R
0xEEA7	00000000		
EDID_168[7:0]			R
0xEEA8	00000000		
EDID_169[7:0]			R
0xEEA9	00000000		
EDID_170[7:0]			R
0xEEAA	00000000		
EDID_171[7:0]			R
0xEEAB	00000000		
EDID_172[7:0]			R
0xEEAC	00000000		
EDID_173[7:0]			R
0xEEAD	00000000		
EDID_174[7:0]			R
0xEEAE	00000000		
EDID_175[7:0]			R
0xEEAF	00000000		
EDID_176[7:0]			R
0xEEB0	00000000		
EDID_177[7:0]			R
0xEEB1	00000000		
EDID_178[7:0]			R
0xEEB2	00000000		

Reg	Bits	Description	
EDID_179[7:0]			R
0xEEB3	00000000		
EDID_180[7:0]			R
0xEEB4	00000000		
EDID_181[7:0]			R
0xEEB5	00000000		
EDID_182[7:0]			R
0xEEB6	00000000		
EDID_183[7:0]			R
0xEEB7	00000000		
EDID_184[7:0]			R
0xEEB8	00000000		
EDID_185[7:0]			R
0xEEB9	00000000		
EDID_186[7:0]			R
0xEEBA	00000000		
EDID_187[7:0]			R
0xEEBB	00000000		
EDID_188[7:0]			R
0xEEBC	00000000		
EDID_189[7:0]			R
0xEEBD	00000000		
EDID_190[7:0]			R
0xEEBE	00000000		
EDID_191[7:0]			R
0xEEBF	00000000		
EDID_192[7:0]			R
0xEEC0	00000000		
EDID_193[7:0]			R
0xEEC1	00000000		
EDID_194[7:0]			R
0xEEC2	00000000		
EDID_195[7:0]			R
0xEEC3	00000000		
EDID_196[7:0]			R
0xEEC4	00000000		

Reg	Bits	Description	
EDID_197[7:0]			R
0xEEC5	00000000		
EDID_198[7:0]			R
0xEEC6	00000000		
EDID_199[7:0]			R
0xEEC7	00000000		
EDID_200[7:0]			R
0xEEC8	00000000		
EDID_201[7:0]			R
0xEEC9	00000000		
EDID_202[7:0]			R
0xEECA	00000000		
EDID_203[7:0]			R
0xEECB	00000000		
EDID_204[7:0]			R
0xEECC	00000000		
EDID_205[7:0]			R
0xEECD	00000000		
EDID_206[7:0]			R
0xEECE	00000000		
EDID_207[7:0]			R
0xEECF	00000000		
EDID_208[7:0]			R
0xEED0	00000000		
EDID_209[7:0]			R
0xEED1	00000000		
EDID_210[7:0]			R
0xEED2	00000000		
EDID_211[7:0]			R
0xEED3	00000000		
EDID_212[7:0]			R
0xEED4	00000000		
EDID_213[7:0]			R
0xEED5	00000000		
EDID_214[7:0]			R
0xEED6	00000000		

Reg	Bits	Description	
EDID_215[7:0]			R
0xEED 7	00000000		
EDID_216[7:0]			R
0xEED 8	00000000		
EDID_217[7:0]			R
0xEED 9	00000000		
EDID_218[7:0]			R
0xEED A	00000000		
EDID_219[7:0]			R
0xEED B	00000000		
EDID_220[7:0]			R
0xEED C	00000000		
EDID_221[7:0]			R
0xEED D	00000000		
EDID_222[7:0]			R
0xEED E	00000000		
EDID_223[7:0]			R
0xEED F	00000000		
EDID_224[7:0]			R
0xEEE 0	00000000		
EDID_225[7:0]			R
0xEEE 1	00000000		
EDID_226[7:0]			R
0xEEE 2	00000000		
EDID_227[7:0]			R
0xEEE 3	00000000		
EDID_228[7:0]			R
0xEEE 4	00000000		
EDID_229[7:0]			R
0xEEE 5	00000000		
EDID_230[7:0]			R
0xEEE 6	00000000		
EDID_231[7:0]			R
0xEEE 7	00000000		
EDID_232[7:0]			R
0xEEE 8	00000000		

Reg	Bits	Description	
EDID_233[7:0]			R
0xEEE9	00000000		
EDID_234[7:0]			R
0xEEEA	00000000		
EDID_235[7:0]			R
0xEEEB	00000000		
EDID_236[7:0]			R
0xEEEC	00000000		
EDID_237[7:0]			R
0xEEED	00000000		
EDID_238[7:0]			R
0xEEEE	00000000		
EDID_239[7:0]			R
0xEEEF	00000000		
EDID_240[7:0]			R
0xEEF0	00000000		
EDID_241[7:0]			R
0xEEF1	00000000		
EDID_242[7:0]			R
0xEEF2	00000000		
EDID_243[7:0]			R
0xEEF3	00000000		
EDID_244[7:0]			R
0xEEF4	00000000		
EDID_245[7:0]			R
0xEEF5	00000000		
EDID_246[7:0]			R
0xEEF6	00000000		
EDID_247[7:0]			R
0xEEF7	00000000		
EDID_248[7:0]			R
0xEEF8	00000000		
EDID_249[7:0]			R
0xEEF9	00000000		
EDID_250[7:0]			R
0xEEFA	00000000		
EDID_251[7:0]			R
0xEEFB	00000000		

Reg	Bits	Description	
EDID_252[7:0]			R
0xEEFC	00000000		
EDID_253[7:0]			R
0xEEFD	00000000		
EDID_254[7:0]			R
0xEEFE	00000000		
EDID_255[7:0]			R
0xEEFF	00000000		

2.10 TX1 TEST MAP

Reg	Bits	Description	
SPARE_PKT3_EN			R/W
0xF3BF	0000000	This bit is used to enable the Spare Packet 3. 0 - Disabled 1 - Enabled	
SPARE_PKT4_EN			R/W
0xF3BF	0000000	This bit is used to enable the Spare Packet 4. 0 - Disabled 1 - Enabled	
SPARE3_HEADER0[7:0]			R/W
0xF3C0	00000000	This register is used to specify Spare3 Infoframe Header, byte 0.	
SPARE3_HEADER1[7:0]			R/W
0xF3C1	00000000	This register is used to specify Spare3 Infoframe Header, byte 1.	
SPARE3_HEADER2[7:0]			R/W
0xF3C2	00000000	This register is used to specify Spare3 Infoframe Header, byte 2.	
SPARE3_BYTE0[7:0]			R/W
0xF3C3	00000000	This register is used to specify Spare3 Infoframe Packet, byte 0.	
SPARE3_BYTE1[7:0]			R/W
0xF3C4	00000000	This register is used to specify Spare3 Infoframe Packet, byte 1.	
SPARE3_BYTE2[7:0]			R/W
0xF3C5	00000000	This register is used to specify Spare3 Infoframe Packet, byte 2.	
SPARE3_BYTE3[7:0]			R/W
0xF3C6	00000000	This register is used to specify Spare3 Infoframe Packet, byte 3.	
SPARE3_BYTE4[7:0]			R/W
0xF3C7	00000000	This register is used to specify Spare3 Infoframe Packet, byte 4.	
SPARE3_BYTE5[7:0]			R/W
0xF3C8	00000000	This register is used to specify Spare3 Infoframe Packet, byte 5.	
SPARE3_BYTE6[7:0]			R/W
0xF3C9	00000000	This register is used to specify Spare3 Infoframe Packet, byte 6.	
SPARE3_BYTE7[7:0]			R/W
0xF3CA	00000000	This register is used to specify Spare3 Infoframe Packet, byte 7.	
SPARE3_BYTE8[7:0]			R/W
0xF3CB	00000000	This register is used to specify Spare3 Infoframe Packet, byte 8.	
SPARE3_BYTE9[7:0]			R/W
0xF3CC	00000000	This register is used to specify Spare3 Infoframe Packet, byte 9.	
SPARE3_BYTE10[7:0]			R/W
0xF3CD	00000000	This register is used to specify Spare3 Infoframe Packet, byte 10.	

Reg	Bits	Description	
SPARE3_BYTE11[7:0]			R/W
0xF3CE	00000000	This register is used to specify Spare3 Infoframe Packet, byte 11.	
SPARE3_BYTE12[7:0]			R/W
0xF3CF	00000000	This register is used to specify Spare3 Infoframe Packet, byte 12.	
SPARE3_BYTE13[7:0]			R/W
0xF3D0	00000000	This register is used to specify Spare3 Infoframe Packet, byte 13.	
SPARE3_BYTE14[7:0]			R/W
0xF3D1	00000000	This register is used to specify Spare3 Infoframe Packet, byte 14.	
SPARE3_BYTE15[7:0]			R/W
0xF3D2	00000000	This register is used to specify Spare3 Infoframe Packet, byte 15.	
SPARE3_BYTE16[7:0]			R/W
0xF3D3	00000000	This register is used to specify Spare3 Infoframe Packet, byte 16.	
SPARE3_BYTE17[7:0]			R/W
0xF3D4	00000000	This register is used to specify Spare3 Infoframe Packet, byte 17.	
SPARE3_BYTE18[7:0]			R/W
0xF3D5	00000000	This register is used to specify Spare3 Infoframe Packet, byte 18.	
SPARE3_BYTE19[7:0]			R/W
0xF3D6	00000000	This register is used to specify Spare3 Infoframe Packet, byte 19.	
SPARE3_BYTE20[7:0]			R/W
0xF3D7	00000000	This register is used to specify Spare3 Infoframe Packet, byte 20.	
SPARE3_BYTE21[7:0]			R/W
0xF3D8	00000000	This register is used to specify Spare3 Infoframe Packet, byte 21.	
SPARE3_BYTE22[7:0]			R/W
0xF3D9	00000000	This register is used to specify Spare3 Infoframe Packet, byte 22.	
SPARE3_BYTE23[7:0]			R/W
0xF3DA	00000000	This register is used to specify Spare3 Infoframe Packet, byte 23.	
SPARE3_BYTE24[7:0]			R/W
0xF3DB	00000000	This register is used to specify Spare3 Infoframe Packet, byte 24.	
SPARE3_BYTE25[7:0]			R/W
0xF3DC	00000000	This register is used to specify Spare3 Infoframe Packet, byte 25.	
SPARE3_BYTE26[7:0]			R/W
0xF3DD	00000000	This register is used to specify Spare3 Infoframe Packet, byte 26.	
SPARE3_BYTE27[7:0]			R/W
0xF3DE	00000000	This register is used to specify Spare3 Infoframe Packet, byte 27.	

Reg	Bits	Description	
SPARE3_UPDATE			R/W
0xF3DF	00000000	This bit is used to control the Spare3 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 3 via I2C is complete 1 - Set this bit to 1 before updating the spare packet 3 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 3 packet during the vsync region, this spare packet 3 data transmitted will not be affected by the spare packet 3 data programmed via I2C. The spare packet 3 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE3_UPDATE being set back to 0	
SPARE4_HEADER0[7:0]			R/W
0xF3E0	00000000	This register is used to specify Spare4 Infoframe Header, byte 0.	
SPARE4_HEADER1[7:0]			R/W
0xF3E1	00000000	This register is used to specify Spare4 Infoframe Header, byte 1.	
SPARE4_HEADER2[7:0]			R/W
0xF3E2	00000000	This register is used to specify Spare4 Infoframe Header, byte 2.	
SPARE4_BYTE0[7:0]			R/W
0xF3E3	00000000	This register is used to specify Spare4 Infoframe Packet, byte 0.	
SPARE4_BYTE1[7:0]			R/W
0xF3E4	00000000	This register is used to specify Spare4 Infoframe Packet, byte 1.	
SPARE4_BYTE2[7:0]			R/W
0xF3E5	00000000	This register is used to specify Spare4 Infoframe Packet, byte 2.	
SPARE4_BYTE3[7:0]			R/W
0xF3E6	00000000	This register is used to specify Spare4 Infoframe Packet, byte 3.	
SPARE4_BYTE4[7:0]			R/W
0xF3E7	00000000	This register is used to specify Spare4 Infoframe Packet, byte 4.	
SPARE4_BYTE5[7:0]			R/W
0xF3E8	00000000	This register is used to specify Spare4 Infoframe Packet, byte 5.	
SPARE4_BYTE6[7:0]			R/W
0xF3E9	00000000	This register is used to specify Spare4 Infoframe Packet, byte 6.	
SPARE4_BYTE7[7:0]			R/W
0xF3EA	00000000	This register is used to specify Spare4 Infoframe Packet, byte 7.	
SPARE4_BYTE8[7:0]			R/W
0xF3EB	00000000	This register is used to specify Spare4 Infoframe Packet, byte 8.	
SPARE4_BYTE9[7:0]			R/W
0xF3EC	00000000	This register is used to specify Spare4 Infoframe Packet, byte 9.	
SPARE4_BYTE10[7:0]			R/W
0xF3ED	00000000	This register is used to specify Spare4 Infoframe Packet, byte 10.	
SPARE4_BYTE11[7:0]			R/W
0xF3EE	00000000	This register is used to specify Spare4 Infoframe Packet, byte 11.	
SPARE4_BYTE12[7:0]			R/W
0xF3EF	00000000	This register is used to specify Spare4 Infoframe Packet, byte 12.	

Reg	Bits	Description	
SPARE4_BYTE13[7:0]			R/W
0xF3F0	00000000	This register is used to specify Spare4 Infoframe Packet, byte 13.	
SPARE4_BYTE14[7:0]			R/W
0xF3F1	00000000	This register is used to specify Spare4 Infoframe Packet, byte 14.	
SPARE4_BYTE15[7:0]			R/W
0xF3F2	00000000	This register is used to specify Spare4 Infoframe Packet, byte 15.	
SPARE4_BYTE16[7:0]			R/W
0xF3F3	00000000	This register is used to specify Spare4 Infoframe Packet, byte 16.	
SPARE4_BYTE17[7:0]			R/W
0xF3F4	00000000	This register is used to specify Spare4 Infoframe Packet, byte 17.	
SPARE4_BYTE18[7:0]			R/W
0xF3F5	00000000	This register is used to specify Spare4 Infoframe Packet, byte 18.	
SPARE4_BYTE19[7:0]			R/W
0xF3F6	00000000	This register is used to specify Spare4 Infoframe Packet, byte 19.	
SPARE4_BYTE20[7:0]			R/W
0xF3F7	00000000	This register is used to specify Spare4 Infoframe Packet, byte 20.	
SPARE4_BYTE21[7:0]			R/W
0xF3F8	00000000	This register is used to specify Spare4 Infoframe Packet, byte 21.	
SPARE4_BYTE22[7:0]			R/W
0xF3F9	00000000	This register is used to specify Spare4 Infoframe Packet, byte 22.	
SPARE4_BYTE23[7:0]			R/W
0xF3FA	00000000	This register is used to specify Spare4 Infoframe Packet, byte 23.	
SPARE4_BYTE24[7:0]			R/W
0xF3FB	00000000	This register is used to specify Spare4 Infoframe Packet, byte 24.	
SPARE4_BYTE25[7:0]			R/W
0xF3FC	00000000	This register is used to specify Spare4 Infoframe Packet, byte 25.	
SPARE4_BYTE26[7:0]			R/W
0xF3FD	00000000	This register is used to specify Spare4 Infoframe Packet, byte 26.	
SPARE4_BYTE27[7:0]			R/W
0xF3FE	00000000	This register is used to specify Spare4 Infoframe Packet, byte 27.	
SPARE4_UPDATE			R/W
0xF3FF	00000000	This bit is used to control the Spare4 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 4 via I2C is complete 1 - Set this bit to 1 before updating the spare packet 4 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 4 packet during the vsync region, this spare packet 4 data transmitted will not be affected by the spare packet 4 data programmed via I2C. The spare packet 4 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE4_UPDATE being set back to 0	

2.11 TX2 MAIN MAP

Reg	Bits	Description	
N[19:0]			R/W
0xF401 0xF402 0xF403	00000000 00000000 00000000	This signal is used to specifies the audio clock regeneration parameter N. This parameter is used with CTS to regenerate the audio clock in the receiver.	
SPDIF_SF[3:0]			R
0xF404	00000000	This signal is used to readback the audio sampling frequency from the SPDIF channel. 0000 - 44.1kHz 0001 - NA 0010 - 48 kHz 0011 - 32kHz 0100 - NA 0101 - NA 0110 - NA 0111 - NA 1000 - 88.2kHz 1001 - NA 1010 - 96kHz 1011 - NA 1100 - 176.4kHz 1101 - NA 1110 - 192kHz 1111 - NA	
CTS_INTERNAL[19:0]			R
0xF404 0xF405 0xF406	00000000 00000000 00000000	This signal is used to readback the automatically generated Cycle Time Stamp (CTS) parameter. This parameter is used with the N parameter to regenerate the audio clock in the receiver.	
CTS_MANUAL[19:0]			R/W
0xF407 0xF408 0xF409	00000000 00000000 00000000	This signal is used to manually set the Cycle Time Stamp (CTS). This parameter is used with the N parameter to regenerate the audio clock in the receiver.	
CTS_SEL			R/W
0xF40A	00000001	This bit is used to specify whether CTS is automatically or manually set. 0 - Automatic CTS. Use the internally generated CTS value 1 - Manual CTS. Use the CTS programmed via CTS_MANUAL[19:0]	
AUDIO_INPUT_SEL[2:0]			R/W
0xF40A	00000001	This signal is used to specify the audio mode when the input format of the audio is specified. 000 - I2S 001 - SPDIF 010 - One Bit Audio (DSD) 011 - High Bit Rate (HBR) Audio 100 - Reserved	

Reg	Bits	Description	
AUDIO_MODE[1:0]			R/W
0xF40A	000000<u>00</u>01	This signal is used to specify the exact audio mode when the input format of the audio is specified. Case 1: DSD (audio_input_select = 0b010): 0x = DSD raw mode; 1x = SDIF-3 mode Case 2: HBR (audio_input_select = 0b011): 00 = 4 stream, with Bi-Phase Mark (BPM) encoding; 01 = 4 stream, without BPM encoding; 10 = 1 stream, with BPM encoding; 11 = 1 stream, without BPM encoding Case 3: DST (audio_input_select = 0b100): x0 = normal mode; 01 = DST 2x clock; 10 = DST 1x clock (DDR) Case 1 - DSD (AUDIO_INPUT_SELECT = 0b010) 0x - DSD raw mode 1x - SDIF-3 mode Case 2 - HBR (AUDIO_INPUT_SELECT = 0b011) 00 - 4 stream, with Bi-Phase Mark (BPM) encoding 01 - 4 stream, without BPM encoding 10 - 1 stream, with BPM encoding 11 - 1 stream, without BPM encoding	
MCLK_RATIO[1:0]			R/W
0xF40A	00000<u>01</u>	This signal is used to specify the ratio between the audio sampling frequency and the clock described using the N and CTS values. 00 - 128*fs 01 - 256*fs 10 - 384*fs 11 - 512*fs	
SPDIF_EN			R/W
0xF40B	<u>0</u>0001110	This bit is used to enable the SPDIF receiver. The SPDIF receiver is only needed when MCLK is internally generated. 0 - SPDIF receiver disabled 1 - SPDIF receiver enabled	
MCLK_POL			R/W
0xF40B	<u>0</u>0001110	This bit is used to specify the clock polarity for the input MCLK and SCLK for I2S and DSD. The clock polarity indicates the clock edge when input data is latched. 0 - Data latched on rising edge 1 - Data latched on falling edge	
MCLK_EN			R/W
0xF40B	000<u>0</u>1110	This bit is used to select the audio master clock that is used by the audio block. 0 - Use internally generated MCLK 1 - Use external MCLK	
RX_AUD_PACKET_SEL			R/W
0xF40B	0000111<u>0</u>	This bit is used to select the source of audio packet data routed into the HDMI transmitter. 0 - Get audio packet from external audio pins 1 - Get audio packet from internal audio receiver	
AUDIO_SAMPLING_FREQ_SEL			R/W
0xF40C	<u>1</u>0111100	This bit is used to select whether the audio sampling frequency is set automatically or manually (via I2C). 0 - Use sampling frequency from I2S stream, for SPDIF stream 1 - Use sampling frequency from I2C registers	
CS_BIT_OVERRIDE			R/W
0xF40C	1<u>0</u>111100	This bit is used to select the source of channel status bits when using I2S Mode 4. 0 - Use channel status bits from I2S stream 1 - Use channel status bits programmed in I2C registers	
I2S_EN[3:0]			R/W
0xF40C	10<u>1111</u>00	This signal is used to enable the I2S pins. 0000 - All I2S disabled 1111 - All I2S enabled	
I2S_FORMAT[1:0]			R/W
0xF40C	101111<u>00</u>	This signal is used to set the format of the I2S audio stream input to the part. 00 - I2S 01 - Right justified 10 - Left justified 11 - AES3 direct mode	

Reg	Bits	Description
I2S_BIT_WIDTH[4:0]		R/W
0xF40D	00011000	This signal is used to set the I2S bit width. This setting is for right justified audio only. It is not valid for widths greater than 24. 00000 - I2S[0], left channel 00001 - I2S[0], right channel 00010 - I2S[1], left channel 00011 - I2S[1], right channel 00100 - I2S[2], left channel 00101 - I2S[2], right channel 00110 - I2S[3], left channel 00111 - I2S[3], right channel
SUBPKT0_L_SRC[2:0]		R/W
0xF40E	00000001	This signal is used to specify the source of sub packet 0, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT0_R_SRC[2:0]		R/W
0xF40E	00000001	This signal is used to specify the source of sub packet 0, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT1_L_SRC[2:0]		R/W
0xF40F	00010011	This signal is used to specify the source of sub packet 1, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT1_R_SRC[2:0]		R/W
0xF40F	00010011	This signal is used to specify the source of sub packet 1, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel
SUBPKT2_L_SRC[2:0]		R/W
0xF410	00100101	This signal is used to specify the source of sub packet 2, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel

Reg	Bits	Description	
SUBPKT2_R_SRC[2:0]			R/W
0xF410	00100101	This signal is used to specify the source of sub packet 2, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
SUBPKT3_L_SRC[2:0]			R/W
0xF411	00110111	This signal is used to specify the source of sub packet 3, left channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
SUBPKT3_R_SRC[2:0]			R/W
0xF411	00110111	This signal is used to specify the source of sub packet 3, right channel. 000 - I2S[0], left channel 001 - I2S[0], right channel 010 - I2S[1], left channel 011 - I2S[1], right channel 100 - I2S[2], left channel 101 - I2S[2], right channel 110 - I2S[3], left channel 111 - I2S[3], right channel	
CHANNEL_STATUS[1:0]			R/W
0xF412	00000000	This signal is used to set the Channel Status bits [1:0]. Set to 0b00 as specified in IEC60958-3. Refer to IEC60958-3 specification. xx - Channel status bits 0 and 1	
CR_BIT			R/W
0xF412	00000000	This bit is used to set the Channel Status Copyright Information. Refer to the IEC 60958-3 specification. 0 - Copyright asserted 1 - Copyright not asserted	
A_INFO[2:0]			R/W
0xF412	00000000	This signal is used to set the Channel Status Emphasis information. Refer to the IEC 60958-3 specification. 000 - 2 audio channels without pre-emphasis 001 - 2 audio channels with 50/15uS pre-emphasis 010 - Reserved (for 2 audio channels with pre-emphasis) 011 - Reserved (for 2 audio channels with pre-emphasis) 100-111 - Reserved	
CLK_ACC[1:0]			R/W
0xF412	00000000	This signal is used to set the Channel Status Clock Accuracy information. Refer to the IEC 60958-3 specification. 00 - level II - normal accuracy +/-1000 x 10^-6 01 - level I - high accuracy +/- 50 x 10^-6 10 - level III - variable pitch shifted clock 11 - Reserved	
CATEGORY_CODE[7:0]			R/W
0xF413	00000000	This register is used to set the Channel Status Category Code. Refer to the IEC 60958-3 specification. 00000000 - Default value xxxxxxxx - Channel Status category code	
SOURCE_NUMBER[3:0]			R/W
0xF414	00000000	This signal is used to set the Channel Status source number. 0000 - Default value xxxx - Channel Status source number	

Reg	Bits	Description	
WORD_LENGTH[3:0]			R/W
0xF414	0000<u>0000</u>	This signal is used to set the Channel Status Audio Word Length. Refer to the IEC 60958-3 specification. 0000 - Not specified 0001 - Not specified 0010 - 16 bits 0011 - 20 bits 0100 - 18 bits 0101 - 22 bits 0110 - Reserved 0111 - Reserved 1000 - 19 bits 1001 - 23 bits 1010 - 20 bits 1011 - 24 bits 1100 - 17 bits 1101 - 21 bits 1110 - Reserved 1111 - Reserved	
I2S_SF[3:0]			R/W
0xF415	<u>0000</u>0000	This signal is used to set the Sampling frequency for I2S audio. This information is used both by the audio Rx and the pixel rep. Other values reserved. 0000 - 44.1kHz 0001 - Do not use 0010 - 48kHz 0011 - 32kHz 0100 - Do not use 0101 - Do not use 0110 - Do not use 0111 - Do not use 1000 - 88.2kHz 1001 - Do not use 1010 - 96kHz 1011 - Do not use 1100 - 176.4kHz 1101 - Do not use 1110 - 192kHz 1111 - Do not use	
VFE_INPUT_ID[3:0]			R/W
0xF415	0000<u>0000</u>	This signal is used to specify the video input format. 0000 - RGB 444 or YCbCr 444 0001 - YCbCr 422 0101 - Pseudo 422 YCbCr All Other Values - Reserved	
VFE_OUTPUT_FORMAT[1:0]			R/W
0xF416	<u>00</u>000000	This signal is used to set the output format. 10 - Reserved	
VFE_INPUT_CS			R/W
0xF416	0000000<u>0</u>	This bit is used to specify the video input colour space. 0 - RGB 1 - YCbCr	
GEN_444_EN			R/W
0xF417	00000<u>0</u>00	This bit is used to enable 4:2:2 to 4:4:4 up-conversion. 0 - Disable 1 - Enable	
ASPECT_RATIO			R/W
0xF417	000000<u>0</u>0	This bit is used to set the aspect ratio of input video. This bit is used to distinguish between CEA-861D video timing codes where aspect ratio is the only difference. 0 - 4:3 1 - 16:9	
CSC_EN			R/W
0xF418	<u>0</u>1000110	This bit is used to enable the colour space converter. 0 - CSC Disabled 1 - CSC Enabled	

Reg	Bits	Description	
CSC_SCALING_FACTOR[1:0]			R/W
0xF418	0100 0110	This signal is used to specify the CSC scaling factor. The CSC scaling factor sets the fixed point position of the CSC coefficients, including a4, b4, c4 and offsets. 00 - +/- 1.0, -4096 to 4095 01 - +/-2.0, -8192 to 8190 10 - +/- 4.0, -16384 to 16380 11 - +/- 4.0, -16384 to 16380	
CSC_A1[12:0]			R/W
0xF418 0xF419	0100 0110 0110 0010	This signal is used to specify the CSC coefficient for a1.	
CSC_A2[12:0]			R/W
0xF41A 0xF41B	0000 0100 1010 1000	This signal is used to specify the CSC coefficient for a2.	
CSC_A3[12:0]			R/W
0xF41C 0xF41D	0000 0000 0000 0000	This signal is used to specify the CSC coefficient for a3.	
CSC_A4[12:0]			R/W
0xF41E 0xF41F	0001 1100 1000 0100	This signal is used to specify the CSC coefficient for a4.	
CSC_B1[12:0]			R/W
0xF420 0xF421	0001 1100 1011 1111	This signal is used to specify the CSC coefficient for b1.	
CSC_B2[12:0]			R/W
0xF422 0xF423	0000 0100 1010 1011	This signal is used to specify the CSC coefficient for b2.	
CSC_B3[12:0]			R/W
0xF424 0xF425	0001 1110 0111 0000	This signal is used to specify the CSC coefficient for b3.	
CSC_B4[12:0]			R/W
0xF426 0xF427	0000 0010 0001 1110	This signal is used to specify the CSC coefficient for b4.	
CSC_C1[12:0]			R/W
0xF428 0xF429	0000 0000 0000 0000	This signal is used to specify the CSC coefficient for c1.	
CSC_C2[12:0]			R/W
0xF42A 0xF42B	0000 0100 1010 1000	This signal is used to specify the CSC coefficient for c2.	

Reg	Bits	Description	
CSC_C3[12:0]			R/W
0xF42C 0xF42D	00001000 00010010	This signal is used to specify the CSC coefficient for c3.	
CSC_C4[12:0]			R/W
0xF42E 0xF42F	00011011 10101100	This signal is used to specify the CSC coefficient for c4.	
PR_MODE[1:0]			R/W
0xF43B	00000000	This signal is used to specify the pixel repetition mode selection. This should be set to 00 unless a non CEA-861 standard video resolution must be supported. 00 - auto mode 01 - max mode 10 - manual mode 11 - manual mode	
PR_PLL_MANUAL[1:0]			R/W
0xF43B	00000000	This signal is used to specify the ratio between the input pixel clock and the TMDS output clock when manual pixel repetition is enabled. 00 - x1 01 - x2 10 - x4 11 - x4	
PR_VALUE_MANUAL[1:0]			R/W
0xF43B	00000000	This signal is used to specify the user programmed pixel repetition sent to the downstream sink. This field is used in manual pixel repetition. 00 - x1 01 - x2 10 - x4 11 - x4	
VIC_MANUAL[5:0]			R/W
0xF43C	00000000	This signal is used to specify the video code (VIC) to send to the downstream sink. Refer to the CEA-861 specification. xxxxxxx - VIC to send to the downstream sink.	
PR_TO_RX[1:0]			R
0xF43D	00000000	This signal is used to readback the pixel repetition value sent to the downstream sink. 00 - x1 01 - x2 10 - x4 11 - x4	
VIC_TO_RX[5:0]			R
0xF43D	00000000	This signal is used to set the AVI InfoFrame video code (VIC) to send to the downstream sink. xxxxxx - VIC sent to the downstream sink	
VIC_DETECTED[5:0]			R
0xF43E	00000000	This signal is used to readback the input video code (VIC) detected (refer to the CEA-861 specification).	
AUX_VIC_DETECTED[2:0]			R
0xF43F	00000000	This register returns the format of video inputs that have a resolution not defined in the CEA 861 specification. 000 - Set by Register 3E 001 - 240p Not Active 010 - 576i not active 011 - 288p not active 100 - 480i active 101 - 240p active 110 - 576i active 111 - 288p active	

Reg	Bits	Description	
PROGRESSIVE_MODE_INFO[1:0]			R
0xF43F	000<u>00</u>000	This bit is used to specify additional information for 240p or 288p input formats. 00 - Reserved 01 - 262 total lines per frame for 240p and 312 total lines per frame for 288p 10 - 263 total lines per frame for 240p and 313 total lines per frame for 288p 11 - Reserved for 240p and 314 total lines per frame for 288p	
GC_PKT_EN			R/W
0xF440	<u>0</u>0000000	This bit is used to enable the General Control Packet. 0 - Disabled 1 - Enabled	
SPD_PKT_EN			R/W
0xF440	0<u>0</u>000000	This bit is used to enable the Source Product Descriptor InfoFrame. 0 - Disabled 1 - Enabled	
MPEG_PKT_EN			R/W
0xF440	00<u>0</u>00000	This bit is used to enable the MPEG Packet. 0 - Disabled 1 - Enabled	
ACP_PKT_EN			R/W
0xF440	000<u>0</u>0000	This bit is used to enable the ACP Packet. 0 - Disabled 1 - Enabled	
ISRC_PKT_EN			R/W
0xF440	0000<u>0</u>000	This bit is used to enable the ISRC Packet. 0 - Disabled 1 - Enabled	
GM_PKT_EN			R/W
0xF440	00000<u>0</u>00	This bit is used to enable the Gamut Metadata Packet. 0 - Disabled 1 - Enabled	
SPARE_PKT1_EN			R/W
0xF440	000000<u>0</u>0	This bit is used to enable the Spare Packet 2. 0 - Disabled 1 - Enabled	
SPARE_PKT0_EN			R/W
0xF440	0000000<u>0</u>	This bit is used to enable the Spare Packet 1. 0 - Disabled 1 - Enabled	
SYSTEM_PD			R/W
0xF441	0<u>1</u>010000	This bit is used to power down the TX. 0 - Normal operation 1 - Power down TX	
HPD_STATE			R
0xF442	1<u>0</u>010000	This bit is used to readback the state of the hot plug detect. 0 - Hot Plug Detect inactive (low) 1 - Hot Plug active (high)	
RX_SENSE_STATE			R
0xF442	10<u>0</u>10000	This bit is used to readback the state of the Rx sense. 0 - HDMI clock termination not detected 1 - HDMI clock termination detected	
I2S_32BIT_MODE			R
0xF442	1001<u>0</u>000	This bit is used to readback the I2S mode detection. It shows the number of SCLK periods per LRCLK period. 0 - I2S 32 bit mode detected 1 - I2S 64 bit mode detected	

Reg	Bits	Description
N_CTS_PKT_EN		R/W
0xF444	01111001	This bit is used to enable the N and CTS packets. 0 - Disable N_CTS packet 1 - Enable N_CTS packet
AUDIO_SAMPLE_PKT_EN		R/W
0xF444	01111001	This bit is used to enable the Audio Sample Packet. 0 - Disable audio sample packet 1 - Enable audio sample packet
AVIIF_PKT_EN		R/W
0xF444	01111001	This bit is used to enable the AVI InfoFrame Packet. 0 - Disable AVI InfoFrame 1 - Enable AVI InfoFrame
AUDIOIF_PKT_EN		R/W
0xF444	01111001	This bit is used to enable the Audio InfoFrame. 0 - Disable audio InfoFrame 1 - Enable audio InfoFrame
PACKET_MEMORY_ADDRESS[7:0]		R/W
0xF445	01110000	This register is used to set the I2C address for the packet memory.
DSD_EN[7:0]		R/W
0xF446	00000000	This register is used to enable the DSD data channel.
DSD_MUX_EN		R/W
0xF447	00000001	This bit is used to enable DSD mode. 0 - Disabled 1 - Enabled
PAPB_SYNC		R/W
0xF447	00000001	This bit is used to synchronize the Pa and Pb syncwords with subpacket 0 for HBR audio. 0 - No function 1 - Synchronize Pa and Pb syncwords with subpacket 0
SAMPLE_INVALID[3:0]		R/W
0xF447	00000001	This signal is used to indicate that the subpackets have valid data. xxx0 - subpackets 0 have valid data xxx1 - subpackets 0 do not have valid data xx0x - subpackets 1 have valid data xx1x - subpackets 1 do not have valid data x0xx - subpackets 2 have valid data x1xx - subpackets 2 do not have valid data 0xxx - subpackets 3 have valid data 1xxx - subpackets 3 do not have valid data
ARC_EFF_TRAN_EN		R/W
0xF447	00000001	When enable it ensures more efficient transmission of ARC packets and audio samples in 176.4kHz and 192kHz modes. This ensures ACR packets can get sent at the right rate (~1ms). 0 - ARC packet efficient transmision disable. 1 - ARC packet efficient transmition enable.
DITHER_MODE[5:0]		R/W
0xF449	01010100	This signal is used to select the ADI proprietary downdither method. The downdither module has 3 stages, each two bits for one stage. [5:4] for 14 bits to 12 bits, [3:2] for 12 bits to 10 bits, [1:0] for 10 bits to 8 bits. 000000 - Active dither 101010 - Truncate
AUTO_CHECKSUM_EN		R/W
0xF44A	10000000	This bit is used to enable autocheck sum generation for infoframes. 0 - Disabled 1 - Use the auto check sum generated by the chip for all IF packets
AVI_UPDATE		R/W
0xF44A	10000000	This bit is used to trigger an update of the AVI infoframe. 0 - Disabled 1 - Enabled

Reg	Bits	Description	
AUDIO_UPDATE			R/W
0xF44A	10000000	This bit is used to trigger an update of the Audio infoframe. 0 - Disabled 1 - Enabled	
GCP_UPDATE			R/W
0xF44A	10000000	This bit is used to trigger an update of the GCP infoframe. 0 - Disabled 1 - Enabled	
MAN_LAYOUT_EN			R/W
0xF44A	10000000	This bit is used to manually set whether audio layout 0 or 1 are used. When transmitting HDMI audio sample packets with compressed audio the layout is automatically set to 1 (multichannel). When enabled, this bit in conjunction with Man_layout_sel allow users to select layout 0. 0 - The Audio sample packet layout is set automatically 1 - The audio sample packet layout value is set by Man_layout_sel	
MAN_LAYOUT_SEL			R/W
0xF44A	10000000	This bit is used to select the audio layout value. 0 - Audio Layout 0 is used if Man_layout_en = 1 1 - Audio Layout 1 is used if Man_layout_en = 1	
CLEAR_AVMUTE			R/W
0xF44B	00000000	This bit is used to control the CLEAR_AVMUTE signal. 0 - Set CLEAR_AVMUTE to 0 1 - Set CLEAR_AVMUTE to 1	
SET_AVMUTE			R/W
0xF44B	00000000	This bit is used to control the SET_AVMUTE signal. 0 - Set SET_AVMUTE to 0 1 - Set SET_AVMUTE to 1	
GC_CD[3:0]			R/W
0xF44C	00000000	This signal is used to set the colour depth of the video send to downstream sink. 0000 - colour depth not indicated 0100 - 24 bits per pixel 0101 - 30 bits per pixel 0110 - 36 bits per pixel 0111-111 - Reserved	
GC_BYTE2[7:0]			R/W
0xF44D	00000000	This register is used to set byte 2 of the General Control subpacket.	
GC_BYTE3[7:0]			R/W
0xF44E	00000000	This register is used to set byte 3 of the General Control subpacket.	
GC_BYTE4[7:0]			R/W
0xF44F	00000000	This register is used to set byte 4 of the General Control subpacket.	
GC_BYTE5[7:0]			R/W
0xF450	00000000	This register is used to set byte 5 of the General Control subpacket.	
GC_BYTE6[7:0]			R/W
0xF451	00000000	This register is used to set byte 6 of the General Control subpacket.	
AVI_VERSION[2:0]			R/W
0xF452	00000010	This signal is used to specify the AVI InfoFrame version.	
AVI_LENGTH[4:0]			R/W
0xF453	00001101	This signal is used to specify the AVI InfoFrame length. This is the number of valid bytes in the AVI InfoFrame packet body excluding the checksum.	
AVI_CHECKSUM[7:0]			R/W
0xF454	00000000	This register is used to specify the AVI InfoFrame checksum. The checksum should only be used in manual checksum mode (See AUTO_CHECKSUM_EN).	

Reg	Bits	Description	
AVI_BYTE1_7			R/W
0xF455	0000000	This bit is used to set bit 7 of AVI InfoFrame Data Byte 1. Set to 0 as per the CEA 861 specification.	
Y1Y0[1:0]			R/W
0xF455	00000000	This signal is used to set the Output Format (AVI InfoFrame). 00 - RGB 01 - YCbCr 422 10 - YCbCr 444 11 - reserved	
A0			R/W
0xF455	00000000	This bit is used to set the Active Format Information status (AVI InfoFrame). 0 - No data 1 - Active format Information valid	
B1B0[1:0]			R/W
0xF455	00000000	This signal is used to set the Bar Information (AVI InfoFrame). 00 - No bar information 01 - Vertical bar information valid 10 - Horizontal bar information valid 11 - Horizontal and vertical bar information valid	
S1S0[1:0]			R/W
0xF455	00000000	This signal is used to set the Scan Information (AVI InfoFrame). 00 - No information 01 - Overscanned (TV) 10 - Underscanned (PC) 11 - Undefined	
C1C0[1:0]			R/W
0xF456	00000000	This signal is used to set the Colorimetry (AVI InfoFrame). 00 - No data 01 - SMPTE 170M, ITU601 10 - ITU709 11 - Undefined	
M1M0[1:0]			R/W
0xF456	00000000	This signal is used to set the Picture Aspect Ratio (AVI InfoFrame). 00 - No data 01 - 4:3 10 - 16:9 11 - Undefined	
R[3:0]			R/W
0xF456	00000000	This signal is used to set the Active Format Aspect Ratio (AVI InfoFrame). 1000 - Same as aspect ratio 1001 - 4:3 (centre) 1010 - 16:9 (centre) 1011 - 14:9 (centre)	
ITC			R/W
0xF457	00000000	This bit is used to set the IT Content (AVI InfoFrame). 0 - Not available 1 - Available	
EC[2:0]			R/W
0xF457	00000000	This signal is used to set the Extended Colorimetry (AVI InfoFrame). 000 - XVYCC 601 001 - XVUCC 709	
Q1Q0[1:0]			R/W
0xF457	00000000	This signal is used to set the RGB Quantization Range (AVI InfoFrame). 00 - Default range 01 - Limited range 10 - Full range 11 - Fixed	

Reg	Bits	Description	
SC[1:0]			R/W
0xF457	000000<u>00</u>	This signal is used to set the Non-Uniform Picture Scaling information (AVI InfoFrame). 00 - No known non-uniform scaling 01 - Picture has been scaled horizontally 10 - Picture has been scaled vertically 11 - Picture has been scaled horizontally and vertically	
AVI_BYTE4_7			R/W
0xF458	<u>0</u>0000000	This bit is used to set bit 7 of the AVI InfoFrame Data, byte 4. Set to 0 as per the CEA-861 specification.	
AVI_BYTE5_7_4[3:0]			R/W
0xF459	<u>0000</u>0000	This signal is used to set bits [7:4] of the AVI InfoFrame Data, byte 5.	
AVI_BYTE6[7:0]			R/W
0xF45A	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 6.	
AVI_BYTE7[7:0]			R/W
0xF45B	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 7.	
AVI_BYTE8[7:0]			R/W
0xF45C	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 8.	
AVI_BYTE9[7:0]			R/W
0xF45D	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 9.	
AVI_BYTE10[7:0]			R/W
0xF45E	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 10.	
AVI_BYTE11[7:0]			R/W
0xF45F	<u>00000000</u>	'This register is used to specify the AVI Infoframe, byte 11.	
AVI_BYTE12[7:0]			R/W
0xF460	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 12.	
AVI_BYTE13[7:0]			R/W
0xF461	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 13.	
AVI_BYTE14[7:0]			R/W
0xF462	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 14.	
AVI_BYTE15[7:0]			R/W
0xF463	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 15.	
AVI_BYTE16[7:0]			R/W
0xF464	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 16.	
AVI_BYTE17[7:0]			R/W
0xF465	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 17.	
AVI_BYTE18[7:0]			R/W
0xF466	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 18.	
AVI_BYTE19[7:0]			R/W
0xF467	<u>00000000</u>	This register is used to specify the AVI Infoframe, byte 19.	

Reg	Bits	Description	
AVI_BYTE20[7:0]			R/W
0xF468	00000000	This register is used to specify the AVI Infoframe, byte 20.	
AVI_BYTE21[7:0]			R/W
0xF469	00000000	This register is used to specify the AVI Infoframe, byte 21.	
AVI_BYTE22[7:0]			R/W
0xF46A	00000000	This register is used to specify the AVI Infoframe, byte 22.	
AVI_BYTE23[7:0]			R/W
0xF46B	00000000	This register is used to specify the AVI Infoframe, byte 23.	
AVI_BYTE24[7:0]			R/W
0xF46C	00000000	This register is used to specify the AVI Infoframe, byte 24.	
AVI_BYTE25[7:0]			R/W
0xF46D	00000000	This register is used to specify the AVI Infoframe, byte 25.	
AVI_BYTE26[7:0]			R/W
0xF46E	00000000	This register is used to specify the AVI Infoframe, byte 26.	
AVI_BYTE27[7:0]			R/W
0xF46F	00000000	This register is used to specify the AVI Infoframe, byte 27.	
AUDIOIF_VERSION[2:0]			R/W
0xF470	00000001	This signal is used to specify the Audio InfoFrame Version.	
AUDIOIF_LENGTH[4:0]			R/W
0xF471	00001010	This signal is used to set the Audio InfoFrame length. This is the number of valid bytes in the Audio InfoFrame packet body excluding the checksum.	
AUDIOIF_CHECKSUM[7:0]			R/W
0xF472	00000000	This register is used to set the Audio InfoFrame Checksum. This should only be used in manual checksum mode (See AUTO_CHECKSUM_EN).	
AUDIOIF_CT[3:0]			R/W
0xF473	0000 0000	This signal is used to set the Audio Coding Type (Audio InfoFrame).	
AUDIOIF_BYTE1_3			R/W
0xF473	0000 0 000	This bit is used to set bit 3 of Audio InfoFrame Data, byte 1. Set to 0 as per the CEA-861 specification.	
AUDIOIF_CC[2:0]			R/W
0xF473	00000 000	This signal is used to set the Audio Channel Count (Audio InfoFrame). 000 - Refer to Stream Header 001 - 2 channels 010 - 3 channels 011 - 4 channels 100 - 5 channels 101 - 6 channels 110 - 7 channels 111 - 8 channels	
AUDIOIF_BYTE2_7_5[2:0]			R/W
0xF474	000 00000	This signal is used to set bits [7:5] of Audio InfoFrame Data, byte 2. Set to 000 as per the CEA-861 specification.	

Reg	Bits	Description	
AUDIOIF_SF[2:0]			R/W
0xF474	000000	This signal is used to specify the Audio Sampling Frequency in the Audio InfoFrame. 000 - Case 1: Not DSD audio or Case 2: DSD audio (AUDIO_INPUT_SEL = 0b010) 001 - 64 x 32 kHz 010 - 64 x 44.1 kHz 011 - 64 x 48 kHz 100 - 64 x 88.2 kHz 101 - 64 x 96 kHz 110 - 64 x 176.4 kHz 111 - 64 x 192 kHz	
AUDIOIF_SS[1:0]			R/W
0xF474	00000000	This signal is used to set the Sample Size (Audio InfoFrame).	
AUDIOIF_BYTE3[7:0]			R/W
0xF475	00000000	This register is used to set the Data Byte 3 (Audio InfoFrame). Set to 0x0 as per the CEA-861 specification.	
AUDIOIF_CA[7:0]			R/W
0xF476	00000000	This register is used to set the Speaker Mapping or placement (Audio InfoFrame). 00000000 - Default value xxxxxxxx - Speaker mapping	
AUDIOIF_DM_INH			R/W
0xF477	0	This bit is used to set the Down-mix Information (Audio InfoFrame). 0 - Permitted or no information about this 1 - Prohibited	
AUDIOIF_LSV[3:0]			R/W
0xF477	0000	This signal is used to set the Audio Level Shift Value (Audio InfoFrame). 0000 - 0dB attenuation 0001 - 1dB attenuation 0010 - 2dB attenuation 0011 - 3dB attenuation 0100 - 4dB attenuation 0101 - 5dB attenuation 0110 - 6dB attenuation 0111 - 7dB attenuation 1000 - 8dB attenuation 1001 - 9dB attenuation 1010 - 10dB attenuation 1011 - 11dB attenuation 1100 - 12dB attenuation 1101 - 13dB attenuation 1110 - 14dB attenuation 1111 - 15dB attenuation	
AUDIOIF_BYTE5_2_0[2:0]			R/W
0xF477	00000000	This signal is used to set bits [2:0] of Audio InfoFrame Data, byte 5. Set to 000 as per the CEA-861 specification.	
AUDIOIF_BYTE6[7:0]			R/W
0xF478	00000000	This register is used to set Data Byte 6 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE7[7:0]			R/W
0xF479	00000000	This register is used to set Data Byte 7 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE8[7:0]			R/W
0xF47A	00000000	This register is used to set Data Byte 8 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
AUDIOIF_BYTE9[7:0]			R/W
0xF47B	00000000	This register is used to set Data Byte 9 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	

Reg	Bits	Description	
AUDIOIF_BYTE10[7:0]			R/W
0xF47C	00000000	This register is used to set Data Byte 10 (Audio InfoFrame). Set to 0x00 as per the CEA-861 specification.	
PRE_EN_CH0			R/W
0xF480	0111**1**111	Enable data channel 0 1 - Enabled 0 = Disabled	
PRE_EN_CH1			R/W
0xF480	01111**1**11	Enable data channel 1 1 - Enabled 0 = Disabled	
PRE_EN_CH2			R/W
0xF480	011111**1**1	Enable data channel 2 1 - Enabled 0 = Disabled	
PRE_EN_CLK			R/W
0xF480	0111111**1**	Enable clock channel 1 - Enabled 0 = Disabled	
CHG_INJ_CH0[3:0]			R/W
0xF481	**1000**1000	Binary control of charge injection for data channel 0 with LSB cap value of 77fF	
CHG_INJ_CH1[3:0]			R/W
0xF481	1000**1000**	Binary control of charge injection for data channel 1 with LSB cap value of 77fF	
CHG_INJ_CH2[3:0]			R/W
0xF482	**1000**1000	Binary control of charge injection for data channel 2 with LSB cap value of 77fF	
CHG_INJ_CLK[3:0]			R/W
0xF482	1000**1000**	Binary control of charge injection for clock channel with LSB cap value of 77fF	
CH0_STERM_DISABLE			R/W
0xF483	**0**0000000	This bit is used to disable source termination on channel 0. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CH1_STERM_DISABLE			R/W
0xF484	**0**0000000	This bit is used to disable source termination on channel 1. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CH2_STERM_DISABLE			R/W
0xF485	**0**0000000	This bit is used to disable source termination on channel 2. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
CLK_STERM_DISABLE			R/W
0xF486	**0**0000000	This bit is used to disable source termination on the clk channel. Bits 6:2 in this register must always be set to 0. 0 - Source Termination Enabled 1 - Source Termination Disabled	
HPD_INT_EN			R/W
0xF494	**1**1000000	This bit is used to enable the HPD interrupt. 0 - Enabled 1 - Disabled	
RX_SENSE_INT_EN			R/W
0xF494	1**1**000000	This bit is used to enable the RX Sense interrupt. 0 - Enabled 1 - Disabled	
VSYNC_INT_EN			R/W
0xF494	11**0**00000	This bit is used to enable the Vsync Edge interrupt. 0 - Enabled 1 - Disabled	

Reg	Bits	Description	
EDID_READY_INT_EN			R/W
0xF494	1100000	This bit is used to enable the EDID Ready interrupt. 0 - Enabled 1 - Disabled	
HDCP_AUTHENTICATED_INT_EN			R/W
0xF494	1100000	This bit is used to enable the HDCP Authenticated interrupt. 0 - Enabled 1 - Disabled	
RI_READY_INT_EN			R/W
0xF494	1100000	This bit is used to enable the Ri Ready interrupt. 0 - Enabled 1 - Disabled	
HDCP_ERROR_INT_EN			R/W
0xF495	0000000	This bit is used to enable the HDCP Controller Error interrupt. 0 - Enabled 1 - Disabled	
BKSV_FLAG_INT_EN			R/W
0xF495	0000000	This bit is used to enable the BKSV Flag interrupt. 0 - Enabled 1 - Disabled	
HPD_INT			R/W
0xF496	0000000	This bit is used to readback and control the HPD interrupt. 0 - Interrupt not active. 1 - Interrupt active. A transition for high to low or low to high has been detected on the input HPD signal	
RX_SENSE_INT			R/W
0xF496	0000000	This bit is used to readback and control the Rx Sense interrupt. 0 - Interrupt Not Active 1 - Interrupt Active. The TMDS clock lines voltage has crossed 1.8V from high to low or low to high	
VSYNC_INT			R/W
0xF496	0000000	This bit is used to readback and control the Vsync edge interrupt. 0 - Interrupt not active 1 - Interrupt active. A leading edge on the input Vsync has been detected.	
EDID_READY_INT			R/W
0xF496	0000000	This bit is used to readback and control the EDID Ready interrupt. 0 - no interrupt detected 1 - interrupt detected	
HDCP_AUTHENTICATED_INT			R/W
0xF496	0000000	This bit is used to readback and control the HDCP Authenticated interrupt. 0 - no interrupt detected 1 - interrupt detected	
RI_READY_INT			R/W
0xF496	0000000	This bit is used to readback and control the Ri Ready interrupt. 0 - Interrupt not active 1 - interrupt active	
HDCP_ERROR_INT			R/W
0xF497	0000000	This bit is used to readback and control the HDCP/EDID Controller Error interrupt. Refer to HDCP_CONTROLLER_ERROR for error codes. 0 - Interrupt not active 1 - Interrupt active. The HDCP/EDID controller has reported an error. This error is available in HDCP_CONTROLLER_ERROR	
BKSV_FLAG_INT			R/W
0xF497	0000000	This bit is used to readback and control the BKSV Flag interrupt. 0 - Interrupt not active 1 - Interrupt active. The KSVs from the downstream sink have been read and available in the Memory Map	

Reg	Bits	Description	
CEC_TX_ARBITRATION_LOST_INT			R/W
0xF497	00000000	This bit is used to readback and control the CEC TX Arbitration Lost interrupt. 0 - Interrupt not active 1 - Interrupt active. The CEC controller is indicating that the TX has lost arbitration	
CEC_TX_RETRY_TIMEOUT_INT			R/W
0xF497	00000000	This bit is used to readback and control the CEC TX Retry Timeout interrupt. 0 - Interrupt not active 1 - Interrupt active. The CEC controller is indicating that the TX retry timeout has expired	
CEC_PD			R/W
0xF498	00000000	This bit is used to powerdown the CEC controller. 0 - CEC enabled 1 - CEC power down and CEC map reset	
HIGH_FREQ_VIDEO[1:0]			R/W
0xF49E	00000000	This signal is used to set the high frequency video enable. This control can be used to configure HDMI TX for 2x and 4x refresh rates. 00 - "normal" refresh rates 01 - 2x refresh rates 10 - 4x refresh rates 11 - "normal" refresh rates	
VIDEO_OFFSET_CTRL[1:0]			R/W
0xF49E	00000000	This signal is used to control the video offset control. Bit 1 is the control to offset the U and V channels. Bit 0 is the black image control to force black video data out of the TX. 00 - No video offset & black image disabled 01 - No video offset & black image enabled 10 - Video offset & black image disabled 11 - Video offset & black image enabled	
HPD_OVERRIDE[1:0]			R/W
0xF49F	00000000	This signal is used to select the source of the internal HPD signal. 00 - HPD from HPD pin and CDC HPD 01 - HPD from CDC HPD 10 - HPD from HPD pin 11 - HPD set to 1	
HDCP_START_DELAY[2:0]			R/W
0xF4AB	00100000	This signal is used to control the delay programmability between HDCP Enable and read BKSV. 000 - no delay 001 - 1ms 010 - 2ms 011 - 5ms 100 - 10ms 101 - 20ms 110 - 40ms 111 - 200ms	
HDCP_1P1_DIS			R/W
0xF4AE	00000000	This bit is used to force the part into HDCP 1.0 mode - even if the RX side supports HDCP 1.1. 0 - enabled 1 - disabled	
R0_WAIT_TIME[1:0]			R/W
0xF4AE	00000000	This signal is used to control the programmable HDCP Ro' read delay. 00 - 100ms 01 - 110ms 10 - 150ms 11 - 200ms	
HDCP_REPEATER_TIMEOUT[1:0]			R/W
0xF4AE	00000000	This signal is used to configure the HDMI Tx repeater Ready bit watchdog timer. 00 - 5s 01 - 6.5s 10 - 8s 11 - 10s	

Reg	Bits	Description	
HDCP_DESIRED			R/W
0xF4AF	00010100	This bit is used to request HDCP encryption. 0 - Input audio and video content not to be encrypted 1 - The input audio and video content should be encrypted	
FRAME_ENC			R/W
0xF4AF	00010100	This bit is used to request HDCP frame encryption. 0 - The current video frame should not be encrypted 1 - The current video frame should be encrypted	
HDMI_DVI_SEL_EN			R/W
0xF4AF	00010100	This bit is used to enable the output mode control. 0 - Automatic 1 - Output mode set by hdmi_dvi_sel	
HDMI_DVI_SEL			R/W
0xF4AF	00010100	This bit is used to control the output mode - DVI or HDMI. 0 - DVI 1 - HDMI	
AN[63:0]			R
0xF4B0 0xF4B1 0xF4B2 0xF4B3 0xF4B4 0xF4B5 0xF4B6 0xF4B7	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	This register is used to readback byte 0 of An/AKSV. This register is used to readback byte 1 of An/AKSV. This register is used to readback byte 2 of An/AKSV. This register is used to readback byte 3 of An/AKSV. This register is used to readback byte 4 of An/AKSV. This register is used to readback byte 5 of An/AKSV. This register is used to readback byte 6 of An/AKSV. This register is used to readback byte 7 of An/AKSV.	
ENC_ON			R
0xF4B8	00000000	This bit is used to readback the HDCP encryption status. 0 - The audio and video content is not being encrypted 1 - The audio and video content is being encrypted	
KEYS_READ_ERROR			R
0xF4B8	00000000	This bit is used to readback the HDCP Key Reading Error status. 0 - The HDCP key has been read successfully 1 - The HDCP key was not read successfully	
BCAPS[7:0]			R
0xF4BE	00000000	This register is used to readback the BCAPS value. BCAPS[6] indicates Repeater, BCAPS[5] indicates BKSV FIFO ready, BCAPS[4] indicates Fast DDC bus, BCAPS[1] indicates HDMI 1.1, BCAPS[0] indicates Fast Re-authentication, See the HDCP specification.	
BKSV[7:0]			R
0xF4BF	00000000	This register is used to readback the BKSV Byte 0 read from the downstream receiver by the HDCP controller.	
BKSV[15:8]			R
0xF4C0	00000000	This register is used to readback the BKSV Byte 1 read from the downstream receiver by the HDCP controller.	
BKSV[23:16]			R
0xF4C1	00000000	This register is used to readback the BKSV Byte 2 read from the downstream receiver by the HDCP controller.	
BKSV[31:24]			R
0xF4C2	00000000	This register is used to readback the BKSV Byte 3 read from the downstream receiver by the HDCP controller.	

Reg	Bits	Description	
BKSV[39:32]			R
0xF4C3	00000000	This register is used to readback the BKSV Byte 4 read from the downstream receiver by the HDCP controller.	
EDID_SEGMENT[7:0]			R/W
0xF4C4	00000000	This register is used to set the segment of the EDID read from the downstream receiver. xxxxxxxx - User programmed EDID segment value	
AN_STOP			R
0xF4C5	0000000	This bit is set high to get An. Set low again if authentication is restarted.	
RI_FLAG			R
0xF4C5	000000	This bit is set high on positive edge of ri_update from HDCP engine. Reset by the firmware.	
BKSV_UPDATE_FLAG			R
0xF4C5	00000	This bit indicates to the HDCP engine that a new BKSV has been received.	
PJ_FLAG			R
0xF4C5	0000000	This bit is set high on positive edge of pj_update from HDCP engine. Reset by the firmware. 0 - No flag 1 - Flag	
HDMI_MODE			R
0xF4C6	0000	This bit is used to readback the HDMI mode status. 0 - DVI 1 - HDMI	
HDCP_REQUESTED			R
0xF4C6	0000	This bit is used to readback the HDCP Request status. This bit shows if a HDCP request has been accepted. 0 - No HDCP request 1 - HDCP has been requested	
RX_SENSE			R
0xF4C6	000000	This bit is used to readback the Rx sense status. 0 - TMDS clock pull-up not present 1 - TMDS clock pull-up present	
TMDS_OUTPUT_EN			R
0xF4C6	0000000	This bit is used to enable TMDS output. 0 - Enable 1 - Disable	
BKSV_FLAG			R/W
0xF4C7	0000000	This bit is used to indicate that KSVs from a downstream device have been read. 0 - Not detected 1 - Detected	
BKSV_COUNT[6:0]			R
0xF4C7	0000000	This signal is used to specify the total number of downstream HDCP devices. xxxxxxx - Total number of downstream HDCP devices	
HDCP_CONTROLLER_ERROR[3:0]			R
0xF4C8	0000	This signal is used to readback the error code when the HDCP controller error interrupt HDCP_ERROR_INT is 1. 0000 - No error 0001 - Bad receiver BKSV 0010 - Ri Mismatch 0011 - Pj Mismatch 0100 - I2C error (usually no acknowledge) 0101 - Timed out waiting for downstream repeater 0110 - Maximum cascade of repeaters exceeded 0111 - SHA-1 Hash check of KSV list fail	

Reg	Bits	Description	
HDCP_CONTROLLER_STATE[3:0]			R
0xF4C8	00000000	This signal is used to readback the state of the EDID/HDCP controller. 0000 - In Reset (No Hot Plug Detected) 0001 - Reading EDID 0010 - In Idle state (Waiting for HDCP Request) 0011 - Initializing HDCP 0100 - HDCP enabled 0101 - Initializing HDCP Repeater 0110 - 1111 - Reserved	
EDID_REREAD			R/W
0xF4C9	00000011	This bit is used to request a the EDID controller to reread the current segment if toggled from 0 to 1 for 10 times consecutively. 0 - No action 1 - Request the EDID/HDCP controller to read the EDID	
EDID_TRIES[3:0]			R/W
0xF4C9	00000011	This signal is used to control the number of times that the EDID read will be attempted if unsuccessful. xxxx - Number of time the EDID/HDCP controller attempts to read the EDID	
HDCP_BSTATUS[15:0]			R
0xF4CA 0xF4CB	00000000 00000000	This signal is used to readback the BSTATUS information for HDCP.	
PLL_LOCK_STATUS			R
0xF4E4	00000000	This bit is used to readback the video PLL lock status. 0 - PLL Not Locked 1 - PLL Locked	
RX_SENSE_PD			R/W
0xF4E6	00000000	This bit is used to enable the termination sense power down. 0 - Termination Sense Monitoring Enabled 1 - Termination Sense Monitoring Disabled	
TMDS_CLK_INVERT			R/W
0xF4EA	10000100	This bit is used to control the inversion of the TMDS clock. 0 - Normal TMDS Clock 1 - Inverted TMDS Clock	
CCI_CONTROLS			R/W
0xF4EA	10000100	This bit is used to control the cross-coupled inverters used to correct for duty cycle distortion. 0 - Disable cross-coupled inverters 1 - Enable cross-coupled inverters for duty cycle	

2.12 TX2 PACKET MAP

Reg	Bits	Description	
SPD_HEADER_BYTE_0[7:0]			R/W
0xFA00	00000000	This register is used to specify SPD Infoframe Header, byte 0.	
SPD_HEADER_BYTE_1[7:0]			R/W
0xFA01	00000000	This register is used to specify SPD Infoframe Header, byte 1.	
SPD_HEADER_BYTE_2[7:0]			R/W
0xFA02	00000000	This register is used to specify SPD Infoframe Header, byte 2.	
SPD_PACKET_BYTE_0[7:0]			R/W
0xFA03	00000000	This register is used to specify SPD Infoframe Packet, byte 0.	
SPD_PACKET_BYTE_1[7:0]			R/W
0xFA04	00000000	This register is used to specify SPD Infoframe Packet, byte 1.	
SPD_PACKET_BYTE_2[7:0]			R/W
0xFA05	00000000	This register is used to specify SPD Infoframe Packet, byte 2.	
SPD_PACKET_BYTE_3[7:0]			R/W
0xFA06	00000000	This register is used to specify SPD Infoframe Packet, byte 3.	
SPD_PACKET_BYTE_4[7:0]			R/W
0xFA07	00000000	This register is used to specify SPD Infoframe Packet, byte 4.	
SPD_PACKET_BYTE_5[7:0]			R/W
0xFA08	00000000	This register is used to specify SPD Infoframe Packet, byte 5.	
SPD_PACKET_BYTE_6[7:0]			R/W
0xFA09	00000000	This register is used to specify SPD Infoframe Packet, byte 6.	
SPD_PACKET_BYTE_7[7:0]			R/W
0xFA0A	00000000	This register is used to specify SPD Infoframe Packet, byte 7.	
SPD_PACKET_BYTE_8[7:0]			R/W
0xFA0B	00000000	This register is used to specify SPD Infoframe Packet, byte 8.	
SPD_PACKET_BYTE_9[7:0]			R/W
0xFA0C	00000000	This register is used to specify SPD Infoframe Packet, byte 9.	
SPD_PACKET_BYTE_10[7:0]			R/W
0xFA0D	00000000	This register is used to specify SPD Infoframe Packet, byte 10.	
SPD_PACKET_BYTE_11[7:0]			R/W
0xFA0E	00000000	This register is used to specify SPD Infoframe Packet, byte 11.	
SPD_PACKET_BYTE_12[7:0]			R/W
0xFA0F	00000000	This register is used to specify SPD Infoframe Packet, byte 12.	
SPD_PACKET_BYTE_13[7:0]			R/W
0xFA10	00000000	This register is used to specify SPD Infoframe Packet, byte 13.	

Reg	Bits	Description	
SPD_PACKET_BYTE_14[7:0]			R/W
0xFA11	00000000	This register is used to specify SPD Infoframe Packet, byte 14.	
SPD_PACKET_BYTE_15[7:0]			R/W
0xFA12	00000000	This register is used to specify SPD Infoframe Packet, byte 15.	
SPD_PACKET_BYTE_16[7:0]			R/W
0xFA13	00000000	This register is used to specify SPD Infoframe Packet, byte 16.	
SPD_PACKET_BYTE_17[7:0]			R/W
0xFA14	00000000	This register is used to specify SPD Infoframe Packet, byte 17.	
SPD_PACKET_BYTE_18[7:0]			R/W
0xFA15	00000000	This register is used to specify SPD Infoframe Packet, byte 18.	
SPD_PACKET_BYTE_19[7:0]			R/W
0xFA16	00000000	This register is used to specify SPD Infoframe Packet, byte 19.	
SPD_PACKET_BYTE_20[7:0]			R/W
0xFA17	00000000	This register is used to specify SPD Infoframe Packet, byte 20.	
SPD_PACKET_BYTE_21[7:0]			R/W
0xFA18	00000000	This register is used to specify SPD Infoframe Packet, byte 21.	
SPD_PACKET_BYTE_22[7:0]			R/W
0xFA19	00000000	This register is used to specify SPD Infoframe Packet, byte 22.	
SPD_PACKET_BYTE_23[7:0]			R/W
0xFA1A	00000000	This register is used to specify SPD Infoframe Packet, byte 23.	
SPD_PACKET_BYTE_24[7:0]			R/W
0xFA1B	00000000	This register is used to specify SPD Infoframe Packet, byte 24.	
SPD_PACKET_BYTE_25[7:0]			R/W
0xFA1C	00000000	This register is used to specify SPD Infoframe Packet, byte 25.	
SPD_PACKET_BYTE_26[7:0]			R/W
0xFA1D	00000000	This register is used to specify SPD Infoframe Packet, byte 26.	
SPD_PACKET_BYTE_27[7:0]			R/W
0xFA1E	00000000	This register is used to specify SPD Infoframe Packet, byte 27.	
SPD_UPDATE			R/W
0xFA1F	00000000	This bit is used to control the SPD Infoframe protection feature. 0 - Set this bit to 0 when the update of the SDP InfoFrame via I2C is complete. 1 - Set this bit to 1 before updating the SDP InfoFrame data via I2C. If this bit is set high while the Tx is transmitting a SDP InfoFrame during the vsync region, this SDP InfoFrame data transmitted will not be affected by the SDP InfoFrame data programmed via I2C. The SDP InfoFrame data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPD_UPDATE being set back to 0.	
MPEG_HEADER_BYTE_0[7:0]			R/W
0xFA20	00000000	This register is used to specify MPEG Infoframe Header, byte 0.	
MPEG_HEADER_BYTE_1[7:0]			R/W
0xFA21	00000000	This register is used to specify MPEG Infoframe Header, byte 1.	

Reg	Bits	Description	
MPEG_HEADER_BYTE_2[7:0]			R/W
0xFA22	00000000	This register is used to specify MPEG Infoframe Header, byte 2.	
MPEG_PACKET_BYTE_0[7:0]			R/W
0xFA23	00000000	This register is used to specify MPEG Infoframe Packet, byte 0.	
MPEG_PACKET_BYTE_1[7:0]			R/W
0xFA24	00000000	This register is used to specify MPEG Infoframe Packet, byte 1.	
MPEG_PACKET_BYTE_2[7:0]			R/W
0xFA25	00000000	This register is used to specify MPEG Infoframe Packet, byte 2.	
MPEG_PACKET_BYTE_3[7:0]			R/W
0xFA26	00000000	This register is used to specify MPEG Infoframe Packet, byte 3.	
MPEG_PACKET_BYTE_4[7:0]			R/W
0xFA27	00000000	This register is used to specify MPEG Infoframe Packet, byte 4.	
MPEG_PACKET_BYTE_5[7:0]			R/W
0xFA28	00000000	This register is used to specify MPEG Infoframe Packet, byte 5.	
MPEG_PACKET_BYTE_6[7:0]			R/W
0xFA29	00000000	This register is used to specify MPEG Infoframe Packet, byte 6.	
MPEG_PACKET_BYTE_7[7:0]			R/W
0xFA2A	00000000	This register is used to specify MPEG Infoframe Packet, byte 7.	
MPEG_PACKET_BYTE_8[7:0]			R/W
0xFA2B	00000000	This register is used to specify MPEG Infoframe Packet, byte 8.	
MPEG_PACKET_BYTE_9[7:0]			R/W
0xFA2C	00000000	This register is used to specify MPEG Infoframe Packet, byte 9.	
MPEG_PACKET_BYTE_10[7:0]			R/W
0xFA2D	00000000	This register is used to specify MPEG Infoframe Packet, byte 10.	
MPEG_PACKET_BYTE_11[7:0]			R/W
0xFA2E	00000000	This register is used to specify MPEG Infoframe Packet, byte 11.	
MPEG_PACKET_BYTE_12[7:0]			R/W
0xFA2F	00000000	This register is used to specify MPEG Infoframe Packet, byte 12.	
MPEG_PACKET_BYTE_13[7:0]			R/W
0xFA30	00000000	This register is used to specify MPEG Infoframe Packet, byte 13.	
MPEG_PACKET_BYTE_14[7:0]			R/W
0xFA31	00000000	This register is used to specify MPEG Infoframe Packet, byte 14.	
MPEG_PACKET_BYTE_15[7:0]			R/W
0xFA32	00000000	This register is used to specify MPEG Infoframe Packet, byte 15.	
MPEG_PACKET_BYTE_16[7:0]			R/W
0xFA33	00000000	This register is used to specify MPEG Infoframe Packet, byte 16.	

Reg	Bits	Description	
MPEG_PACKET_BYTE_17[7:0]			R/W
0xFA34	00000000	This register is used to specify MPEG Infoframe Packet, byte 17.	
MPEG_PACKET_BYTE_18[7:0]			R/W
0xFA35	00000000	This register is used to specify MPEG Infoframe Packet, byte 18.	
MPEG_PACKET_BYTE_19[7:0]			R/W
0xFA36	00000000	This register is used to specify MPEG Infoframe Packet, byte 19.	
MPEG_PACKET_BYTE_20[7:0]			R/W
0xFA37	00000000	This register is used to specify MPEG Infoframe Packet, byte 20.	
MPEG_PACKET_BYTE_21[7:0]			R/W
0xFA38	00000000	This register is used to specify MPEG Infoframe Packet, byte 21.	
MPEG_PACKET_BYTE_22[7:0]			R/W
0xFA39	00000000	This register is used to specify MPEG Infoframe Packet, byte 22.	
MPEG_PACKET_BYTE_23[7:0]			R/W
0xFA3A	00000000	This register is used to specify MPEG Infoframe Packet, byte 23.	
MPEG_PACKET_BYTE_24[7:0]			R/W
0xFA3B	00000000	This register is used to specify MPEG Infoframe Packet, byte 24.	
MPEG_PACKET_BYTE_25[7:0]			R/W
0xFA3C	00000000	This register is used to specify MPEG Infoframe Packet, byte 25.	
MPEG_PACKET_BYTE_26[7:0]			R/W
0xFA3D	00000000	This register is used to specify MPEG Infoframe Packet, byte 26.	
MPEG_PACKET_BYTE_27[7:0]			R/W
0xFA3E	00000000	This register is used to specify MPEG Infoframe Packet, byte 27.	
MPEG_UPDATE			R/W
0xFA3F	00000000	This bit is used to control the MPEG Infoframe protection feature. 0 - Set this bit to 0 when the update of the MPEG InfoFrame via I2C is complete. 1 - Set this bit to 1 before updating the MPEG InfoFrame data via I2C. If this bit is set high while the Tx is transmitting a MPEG InfoFrame during the vsync region, this MPEG InfoFrame data transmitted will not be affected by the MPEG InfoFrame data programmed via I2C. The MPEG InfoFrame data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following MPEG_UPDATE being set back to 0.	
ACP_HEADER_BYTE_0[7:0]			R/W
0xFA40	00000000	This register is used to specify ACP Infoframe Header, byte 0.	
ACP_HEADER_BYTE_1[7:0]			R/W
0xFA41	00000000	This register is used to specify ACP Infoframe Header, byte 1.	
ACP_HEADER_BYTE_2[7:0]			R/W
0xFA42	00000000	This register is used to specify ACP Infoframe Header, byte 2.	
ACP_PACKET_BYTE_0[7:0]			R/W
0xFA43	00000000	This register is used to specify ACP Infoframe Packet, byte 0.	
ACP_PACKET_BYTE_1[7:0]			R/W
0xFA44	00000000	This register is used to specify ACP Infoframe Packet, byte 1.	

Reg	Bits	Description	
ACP_PACKET_BYTE_2[7:0]			R/W
0xFA45	00000000	This register is used to specify ACP Infoframe Packet, byte 2.	
ACP_PACKET_BYTE_3[7:0]			R/W
0xFA46	00000000	This register is used to specify ACP Infoframe Packet, byte 3.	
ACP_PACKET_BYTE_4[7:0]			R/W
0xFA47	00000000	This register is used to specify ACP Infoframe Packet, byte 4.	
ACP_PACKET_BYTE_5[7:0]			R/W
0xFA48	00000000	This register is used to specify ACP Infoframe Packet, byte 5.	
ACP_PACKET_BYTE_6[7:0]			R/W
0xFA49	00000000	This register is used to specify ACP Infoframe Packet, byte 6.	
ACP_PACKET_BYTE_7[7:0]			R/W
0xFA4A	00000000	This register is used to specify ACP Infoframe Packet, byte 7.	
ACP_PACKET_BYTE_8[7:0]			R/W
0xFA4B	00000000	This register is used to specify ACP Infoframe Packet, byte 8.	
ACP_PACKET_BYTE_9[7:0]			R/W
0xFA4C	00000000	This register is used to specify ACP Infoframe Packet, byte 9.	
ACP_PACKET_BYTE_10[7:0]			R/W
0xFA4D	00000000	This register is used to specify ACP Infoframe Packet, byte 10.	
ACP_PACKET_BYTE_11[7:0]			R/W
0xFA4E	00000000	This register is used to specify ACP Infoframe Packet, byte 11.	
ACP_PACKET_BYTE_12[7:0]			R/W
0xFA4F	00000000	This register is used to specify ACP Infoframe Packet, byte 12.	
ACP_PACKET_BYTE_13[7:0]			R/W
0xFA50	00000000	This register is used to specify ACP Infoframe Packet, byte 13.	
ACP_PACKET_BYTE_14[7:0]			R/W
0xFA51	00000000	This register is used to specify ACP Infoframe Packet, byte 14.	
ACP_PACKET_BYTE_15[7:0]			R/W
0xFA52	00000000	This register is used to specify ACP Infoframe Packet, byte 15.	
ACP_PACKET_BYTE_16[7:0]			R/W
0xFA53	00000000	This register is used to specify ACP Infoframe Packet, byte 16.	
ACP_PACKET_BYTE_17[7:0]			R/W
0xFA54	00000000	This register is used to specify ACP Infoframe Packet, byte 17.	
ACP_PACKET_BYTE_18[7:0]			R/W
0xFA55	00000000	This register is used to specify ACP Infoframe Packet, byte 18.	
ACP_PACKET_BYTE_19[7:0]			R/W
0xFA56	00000000	This register is used to specify ACP Infoframe Packet, byte 19.	

Reg	Bits	Description	
ACP_PACKET_BYTE_20[7:0]			R/W
0xFA57	00000000	This register is used to specify ACP Infoframe Packet, byte 20.	
ACP_PACKET_BYTE_21[7:0]			R/W
0xFA58	00000000	This register is used to specify ACP Infoframe Packet, byte 21.	
ACP_PACKET_BYTE_22[7:0]			R/W
0xFA59	00000000	This register is used to specify ACP Infoframe Packet, byte 22.	
ACP_PACKET_BYTE_23[7:0]			R/W
0xFA5A	00000000	This register is used to specify ACP Infoframe Packet, byte 23.	
ACP_PACKET_BYTE_24[7:0]			R/W
0xFA5B	00000000	This register is used to specify ACP Infoframe Packet, byte 24.	
ACP_PACKET_BYTE_25[7:0]			R/W
0xFA5C	00000000	This register is used to specify ACP Infoframe Packet, byte 25.	
ACP_PACKET_BYTE_26[7:0]			R/W
0xFA5D	00000000	This register is used to specify ACP Infoframe Packet, byte 26.	
ACP_PACKET_BYTE_27[7:0]			R/W
0xFA5E	00000000	This register is used to specify ACP Infoframe Packet, byte 27.	
ACP_UPDATE			R/W
0xFA5F	0	This bit is used to control the ACP Infoframe protection feature. 0 - Set this bit to 0 when the update of the ACP packet via I2C is complete. 1 - Set this bit to 1 before updating the ACP packet data via I2C. If this bit is set high while the Tx is transmitting an ACP packet during the vsync region, this ACP packet data transmitted will not be affected by the ACP packet data programmed via I2C. The ACP packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ACP_UPDATE being set back to 0	
ISRC1_HEADER_BYTE_0[7:0]			R/W
0xFA60	00000000	This register is used to specify ISRC1 Infoframe Header, byte 0.	
ISRC1_HEADER_BYTE_1[7:0]			R/W
0xFA61	00000000	This register is used to specify ISRC1 Infoframe Header, byte 1.	
ISRC1_HEADER_BYTE_2[7:0]			R/W
0xFA62	00000000	This register is used to specify ISRC1 Infoframe Header, byte 2.	
ISRC1_PACKET_BYTE_0[7:0]			R/W
0xFA63	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 0.	
ISRC1_PACKET_BYTE_1[7:0]			R/W
0xFA64	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 1.	
ISRC1_PACKET_BYTE_2[7:0]			R/W
0xFA65	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 2.	
ISRC1_PACKET_BYTE_3[7:0]			R/W
0xFA66	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 3.	
ISRC1_PACKET_BYTE_4[7:0]			R/W
0xFA67	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 4.	

Reg	Bits	Description	
ISRC1_PACKET_BYTE_5[7:0]			R/W
0xFA68	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 5.	
ISRC1_PACKET_BYTE_6[7:0]			R/W
0xFA69	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 6.	
ISRC1_PACKET_BYTE_7[7:0]			R/W
0xFA6A	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 7.	
ISRC1_PACKET_BYTE_8[7:0]			R/W
0xFA6B	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 8.	
ISRC1_PACKET_BYTE_9[7:0]			R/W
0xFA6C	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 9.	
ISRC1_PACKET_BYTE_10[7:0]			R/W
0xFA6D	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 10.	
ISRC1_PACKET_BYTE_11[7:0]			R/W
0xFA6E	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 11.	
ISRC1_PACKET_BYTE_12[7:0]			R/W
0xFA6F	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 12.	
ISRC1_PACKET_BYTE_13[7:0]			R
0xFA70	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 13.	
ISRC1_PACKET_BYTE_14[7:0]			R
0xFA71	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 14.	
ISRC1_PACKET_BYTE_15[7:0]			R
0xFA72	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 15.	
ISRC1_PACKET_BYTE_16[7:0]			R
0xFA73	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 16.	
ISRC1_PACKET_BYTE_17[7:0]			R
0xFA74	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 17.	
ISRC1_PACKET_BYTE_18[7:0]			R/W
0xFA75	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 18.	
ISRC1_PACKET_BYTE_19[7:0]			R/W
0xFA76	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 19.	
ISRC1_PACKET_BYTE_20[7:0]			R/W
0xFA77	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 20.	
ISRC1_PACKET_BYTE_21[7:0]			R/W
0xFA78	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 21.	
ISRC1_PACKET_BYTE_22[7:0]			R/W
0xFA79	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 22.	

Reg	Bits	Description	
ISRC1_PACKET_BYTE_23[7:0]			R/W
0xFA7A	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 23.	
ISRC1_PACKET_BYTE_24[7:0]			R/W
0xFA7B	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 24.	
ISRC1_PACKET_BYTE_25[7:0]			R/W
0xFA7C	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 25.	
ISRC1_PACKET_BYTE_26[7:0]			R/W
0xFA7D	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 26.	
ISRC1_PACKET_BYTE_27[7:0]			R/W
0xFA7E	00000000	This register is used to specify ISRC1 Infoframe Packet, byte 27.	
ISRC1_UPDATE			R/W
0xFA7F	00000000	This bit is used to control the ISRC1 Infoframe protection feature. 0 - Set this bit to 0 when the update of the ISRC1 packet via I2C is complete. 1 - Set this bit to 1 before updating the ISRC1 packet data via I2C. If this bit is set high while the Tx is transmitting an ISRC1 packet during the vsync region, this ISRC1 packet data transmitted will not be affected by the ISRC1 packet data programmed via I2C. The ISRC1 packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ISRC1_UPDATE being set back to 0	
ISRC2_HEADER_BYTE_0[7:0]			R/W
0xFA80	00000000	This register is used to specify ISRC2 Infoframe Header, byte 0.	
ISRC2_HEADER_BYTE_1[7:0]			R/W
0xFA81	00000000	This register is used to specify ISRC2 Infoframe Header, byte 1.	
ISRC2_HEADER_BYTE_2[7:0]			R/W
0xFA82	00000000	This register is used to specify ISRC2 Infoframe Header, byte 2.	
ISRC2_PACKET_BYTE_0[7:0]			R/W
0xFA83	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 0.	
ISRC2_PACKET_BYTE_1[7:0]			R/W
0xFA84	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 1.	
ISRC2_PACKET_BYTE_2[7:0]			R/W
0xFA85	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 2.	
ISRC2_PACKET_BYTE_3[7:0]			R/W
0xFA86	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 3.	
ISRC2_PACKET_BYTE_4[7:0]			R/W
0xFA87	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 4.	
ISRC2_PACKET_BYTE_5[7:0]			R/W
0xFA88	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 5.	
ISRC2_PACKET_BYTE_6[7:0]			R/W
0xFA89	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 6.	
ISRC2_PACKET_BYTE_7[7:0]			R/W
0xFA8A	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 7.	

Reg	Bits	Description	
ISRC2_PACKET_BYTE_8[7:0]			R/W
0xFA8B	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 8.	
ISRC2_PACKET_BYTE_9[7:0]			R/W
0xFA8C	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 9.	
ISRC2_PACKET_BYTE_10[7:0]			R/W
0xFA8D	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 10.	
ISRC2_PACKET_BYTE_11[7:0]			R/W
0xFA8E	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 11.	
ISRC2_PACKET_BYTE_12[7:0]			R/W
0xFA8F	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 12.	
ISRC2_PACKET_BYTE_13[7:0]			R/W
0xFA90	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 13.	
ISRC2_PACKET_BYTE_14[7:0]			R/W
0xFA91	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 14.	
ISRC2_PACKET_BYTE_15[7:0]			R/W
0xFA92	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 15.	
ISRC2_PACKET_BYTE_16[7:0]			R/W
0xFA93	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 16.	
ISRC2_PACKET_BYTE_17[7:0]			R/W
0xFA94	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 17.	
ISRC2_PACKET_BYTE_18[7:0]			R/W
0xFA95	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 18.	
ISRC2_PACKET_BYTE_19[7:0]			R/W
0xFA96	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 19.	
ISRC2_PACKET_BYTE_20[7:0]			R/W
0xFA97	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 20.	
ISRC2_PACKET_BYTE_21[7:0]			R/W
0xFA98	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 21.	
ISRC2_PACKET_BYTE_22[7:0]			R/W
0xFA99	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 22.	
ISRC2_PACKET_BYTE_23[7:0]			R/W
0xFA9A	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 23.	
ISRC2_PACKET_BYTE_24[7:0]			R/W
0xFA9B	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 24.	
ISRC2_PACKET_BYTE_25[7:0]			R/W
0xFA9C	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 25.	

Reg	Bits	Description	
ISRC2_PACKET_BYTE_26[7:0]			R/W
0xFA9D	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 26.	
ISRC2_PACKET_BYTE_27[7:0]			R/W
0xFA9E	00000000	This register is used to specify ISRC2 Infoframe Packet, byte 27.	
ISRC2_UPDATE			R/W
0xFA9F	00000000	This bit is used to control the ISRC2 Infoframe protection feature. 0 - Set this bit to 0 when the update of the ISRC2 packet via I2C is complete. 1 - Set this bit to 1 before updating the ISRC2 packet data via I2C. If this bit is set high while the Tx is transmitting an ISRC2 packet during the vsync region, this ISRC2 packet data transmitted will not be affected by the ISRC2 packet data programmed via I2C. The ISRC2 packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following ISRC2_UPDATE being set back to 0	
GM_HEADER_BYTE_0[7:0]			R/W
0xFAA0	00000000	This register is used to specify GM Infoframe Header, byte 0.	
GM_HEADER_BYTE_1[7:0]			R/W
0xFAA1	00000000	This register is used to specify GM Infoframe Header, byte 1.	
GM_HEADER_BYTE_2[7:0]			R/W
0xFAA2	00000000	This register is used to specify GM Infoframe Header, byte 2.	
GM_PACKET_BYTE_0[7:0]			R/W
0xFAA3	00000000	This register is used to specify GM Infoframe Packet, byte 0.	
GM_PACKET_BYTE_1[7:0]			R/W
0xFAA4	00000000	This register is used to specify GM Infoframe Packet, byte 1.	
GM_PACKET_BYTE_2[7:0]			R/W
0xFAA5	00000000	This register is used to specify GM Infoframe Packet, byte 2.	
GM_PACKET_BYTE_3[7:0]			R/W
0xFAA6	00000000	This register is used to specify GM Infoframe Packet, byte 3.	
GM_PACKET_BYTE_4[7:0]			R/W
0xFAA7	00000000	This register is used to specify GM Infoframe Packet, byte 4.	
GM_PACKET_BYTE_5[7:0]			R/W
0xFAA8	00000000	This register is used to specify GM Infoframe Packet, byte 5.	
GM_PACKET_BYTE_6[7:0]			R/W
0xFAA9	00000000	This register is used to specify GM Infoframe Packet, byte 6.	
GM_PACKET_BYTE_7[7:0]			R/W
0xFAAA	00000000	This register is used to specify GM Infoframe Packet, byte 7.	
GM_PACKET_BYTE_8[7:0]			R/W
0xFAAB	00000000	This register is used to specify GM Infoframe Packet, byte 8.	
GM_PACKET_BYTE_9[7:0]			R/W
0xFAAC	00000000	This register is used to specify GM Infoframe Packet, byte 9.	
GM_PACKET_BYTE_10[7:0]			R/W
0xFAAD	00000000	This register is used to specify GM Infoframe Packet, byte 10.	

Reg	Bits	Description	
GM_PACKET_BYTE_11[7:0]			R/W
0xFAAE	00000000	This register is used to specify GM Infoframe Packet, byte 11.	
GM_PACKET_BYTE_12[7:0]			R/W
0xFAAF	00000000	This register is used to specify GM Infoframe Packet, byte 12.	
GM_PACKET_BYTE_13[7:0]			R/W
0xFAB0	00000000	This register is used to specify GM Infoframe Packet, byte 13.	
GM_PACKET_BYTE_14[7:0]			R/W
0xFAB1	00000000	This register is used to specify GM Infoframe Packet, byte 14.	
GM_PACKET_BYTE_15[7:0]			R/W
0xFAB2	00000000	This register is used to specify GM Infoframe Packet, byte 15.	
GM_PACKET_BYTE_16[7:0]			R/W
0xFAB3	00000000	This register is used to specify GM Infoframe Packet, byte 16.	
GM_PACKET_BYTE_17[7:0]			R/W
0xFAB4	00000000	This register is used to specify GM Infoframe Packet, byte 17.	
GM_PACKET_BYTE_18[7:0]			R/W
0xFAB5	00000000	This register is used to specify GM Infoframe Packet, byte 18.	
GM_PACKET_BYTE_19[7:0]			R/W
0xFAB6	00000000	This register is used to specify GM Infoframe Packet, byte 19.	
GM_PACKET_BYTE_20[7:0]			R/W
0xFAB7	00000000	This register is used to specify GM Infoframe Packet, byte 20.	
GM_PACKET_BYTE_21[7:0]			R/W
0xFAB8	00000000	This register is used to specify GM Infoframe Packet, byte 21.	
GM_PACKET_BYTE_22[7:0]			R/W
0xFAB9	00000000	This register is used to specify GM Infoframe Packet, byte 22.	
GM_PACKET_BYTE_23[7:0]			R/W
0xFABA	00000000	This register is used to specify GM Infoframe Packet, byte 23.	
GM_PACKET_BYTE_24[7:0]			R/W
0xFABB	00000000	This register is used to specify GM Infoframe Packet, byte 24.	
GM_PACKET_BYTE_25[7:0]			R/W
0xFABC	00000000	This register is used to specify GM Infoframe Packet, byte 25.	
GM_PACKET_BYTE_26[7:0]			R/W
0xFABD	00000000	This register is used to specify GM Infoframe Packet, byte 26.	
GM_PACKET_BYTE_27[7:0]			R/W
0xFABE	00000000	This register is used to specify GM Infoframe Packet, byte 27.	

Reg	Bits	Description	
GM_UPDATE			R/W
0xFABF	0000000	This bit is used to control the GM Infoframe protection feature. 0 - Set this bit to 0 when the update of the Gamut Metadata packet via I2C is complete. 1 - Set this bit to 1 before updating the Gamut Metadata packet data via I2C. If this bit is set high while the Tx is transmitting a Gamut Metadata packet during the vsync region, this Gamut Metadata packet data transmitted will not be affected by the Gamut Metadata packet data programmed via I2C. The Gamut Metadata packet data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following GMP_UPDATE being set back to 0	
SPARE_PACKET_1_HEADER_BYTE_0[7:0]			R/W
0xFAC0	00000000	This register is used to specify Spare1 Infoframe Header, byte 0.	
SPARE_PACKET_1_HEADER_BYTE_1[7:0]			R/W
0xFAC1	00000000	This register is used to specify Spare1 Infoframe Header, byte 1.	
SPARE_PACKET_1_HEADER_BYTE_2[7:0]			R/W
0xFAC2	00000000	This register is used to specify Spare1 Infoframe Header, byte 2.	
SPARE_PACKET_1_PACKET_BYTE_0[7:0]			R/W
0xFAC3	00000000	This register is used to specify Spare1 Infoframe Packet, byte 0.	
SPARE_PACKET_1_PACKET_BYTE_1[7:0]			R/W
0xFAC4	00000000	This register is used to specify Spare1 Infoframe Packet, byte 1.	
SPARE_PACKET_1_PACKET_BYTE_2[7:0]			R/W
0xFAC5	00000000	This register is used to specify Spare1 Infoframe Packet, byte 2.	
SPARE_PACKET_1_PACKET_BYTE_3[7:0]			R/W
0xFAC6	00000000	This register is used to specify Spare1 Infoframe Packet, byte 3.	
SPARE_PACKET_1_PACKET_BYTE_4[7:0]			R/W
0xFAC7	00000000	This register is used to specify Spare1 Infoframe Packet, byte 4.	
SPARE_PACKET_1_PACKET_BYTE_5[7:0]			R/W
0xFAC8	00000000	This register is used to specify Spare1 Infoframe Packet, byte 5.	
SPARE_PACKET_1_PACKET_BYTE_6[7:0]			R/W
0xFAC9	00000000	This register is used to specify Spare1 Infoframe Packet, byte 6.	
SPARE_PACKET_1_PACKET_BYTE_7[7:0]			R/W
0xFACA	00000000	This register is used to specify Spare1 Infoframe Packet, byte 7.	
SPARE_PACKET_1_PACKET_BYTE_8[7:0]			R/W
0xFACB	00000000	This register is used to specify Spare1 Infoframe Packet, byte 8.	
SPARE_PACKET_1_PACKET_BYTE_9[7:0]			R/W
0xFACC	00000000	This register is used to specify Spare1 Infoframe Packet, byte 9.	
SPARE_PACKET_1_PACKET_BYTE_10[7:0]			R/W
0xFACD	00000000	This register is used to specify Spare1 Infoframe Packet, byte 10.	
SPARE_PACKET_1_PACKET_BYTE_11[7:0]			R/W
0xFACE	00000000	This register is used to specify Spare1 Infoframe Packet, byte 11.	

Reg	Bits	Description	
SPARE_PACKET_1_PACKET_BYTE_12[7:0]			R/W
0xFACF	00000000	This register is used to specify Spare1 Infoframe Packet, byte 12.	
SPARE_PACKET_1_PACKET_BYTE_13[7:0]			R/W
0xFAD0	00000000	This register is used to specify Spare1 Infoframe Packet, byte 13.	
SPARE_PACKET_1_PACKET_BYTE_14[7:0]			R/W
0xFAD1	00000000	This register is used to specify Spare1 Infoframe Packet, byte 14.	
SPARE_PACKET_1_PACKET_BYTE_15[7:0]			R/W
0xFAD2	00000000	This register is used to specify Spare1 Infoframe Packet, byte 15.	
SPARE_PACKET_1_PACKET_BYTE_16[7:0]			R/W
0xFAD3	00000000	This register is used to specify Spare1 Infoframe Packet, byte 16.	
SPARE_PACKET_1_PACKET_BYTE_17[7:0]			R/W
0xFAD4	00000000	This register is used to specify Spare1 Infoframe Packet, byte 17.	
SPARE_PACKET_1_PACKET_BYTE_18[7:0]			R/W
0xFAD5	00000000	This register is used to specify Spare1 Infoframe Packet, byte 18.	
SPARE_PACKET_1_PACKET_BYTE_19[7:0]			R/W
0xFAD6	00000000	This register is used to specify Spare1 Infoframe Packet, byte 19.	
SPARE_PACKET_1_PACKET_BYTE_20[7:0]			R/W
0xFAD7	00000000	This register is used to specify Spare1 Infoframe Packet, byte 20.	
SPARE_PACKET_1_PACKET_BYTE_21[7:0]			R/W
0xFAD8	00000000	This register is used to specify Spare1 Infoframe Packet, byte 21.	
SPARE_PACKET_1_PACKET_BYTE_22[7:0]			R/W
0xFAD9	00000000	This register is used to specify Spare1 Infoframe Packet, byte 22.	
SPARE_PACKET_1_PACKET_BYTE_23[7:0]			R/W
0xFADA	00000000	This register is used to specify Spare1 Infoframe Packet, byte 23.	
SPARE_PACKET_1_PACKET_BYTE_24[7:0]			R/W
0xFADB	00000000	This register is used to specify Spare1 Infoframe Packet, byte 24.	
SPARE_PACKET_1_PACKET_BYTE_25[7:0]			R/W
0xFADC	00000000	This register is used to specify Spare1 Infoframe Packet, byte 25.	
SPARE_PACKET_1_PACKET_BYTE_26[7:0]			R/W
0xFADD	00000000	This register is used to specify Spare1 Infoframe Packet, byte 26.	
SPARE_PACKET_1_PACKET_BYTE_27[7:0]			R/W
0xFADE	00000000	This register is used to specify Spare1 Infoframe Packet, byte 27.	
SPARE1_UPDATE			R/W
0xFADF	00000000	This bit is used to control the Spare1 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 1 via I2C is complete. 1 - Set this bit to 1 before updating the spare packet 1 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 1 packet during the vsync region, this spare packet 1 data transmitted will not be affected by the spare packet 1 data programmed via I2C. The spare packet 1 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE1_UPDATE being set back to 0	

Reg	Bits	Description	
SPARE_PACKET_2_HEADER_BYTE_0[7:0]			R/W
0xFAE 0	00000000	This register is used to specify Spare2 Infoframe Header, byte 0.	
SPARE_PACKET_2_HEADER_BYTE_1[7:0]			R/W
0xFAE 1	00000000	This register is used to specify Spare2 Infoframe Header, byte 1.	
SPARE_PACKET_2_HEADER_BYTE_2[7:0]			R/W
0xFAE 2	00000000	This register is used to specify Spare2 Infoframe Header, byte 2.	
SPARE_PACKET_2_PACKET_BYTE_0[7:0]			R/W
0xFAE 3	00000000	This register is used to specify Spare2 Infoframe Packet, byte 0.	
SPARE_PACKET_2_PACKET_BYTE_1[7:0]			R/W
0xFAE 4	00000000	This register is used to specify Spare2 Infoframe Packet, byte 1.	
SPARE_PACKET_2_PACKET_BYTE_2[7:0]			R/W
0xFAE 5	00000000	This register is used to specify Spare2 Infoframe Packet, byte 2.	
SPARE_PACKET_2_PACKET_BYTE_3[7:0]			R/W
0xFAE 6	00000000	This register is used to specify Spare2 Infoframe Packet, byte 3.	
SPARE_PACKET_2_PACKET_BYTE_4[7:0]			R/W
0xFAE 7	00000000	This register is used to specify Spare2 Infoframe Packet, byte 4.	
SPARE_PACKET_2_PACKET_BYTE_5[7:0]			R/W
0xFAE 8	00000000	This register is used to specify Spare2 Infoframe Packet, byte 5.	
SPARE_PACKET_2_PACKET_BYTE_6[7:0]			R/W
0xFAE 9	00000000	This register is used to specify Spare2 Infoframe Packet, byte 6.	
SPARE_PACKET_2_PACKET_BYTE_7[7:0]			R/W
0xFAE A	00000000	This register is used to specify Spare2 Infoframe Packet, byte 7.	
SPARE_PACKET_2_PACKET_BYTE_8[7:0]			R/W
0xFAE B	00000000	This register is used to specify Spare2 Infoframe Packet, byte 8.	
SPARE_PACKET_2_PACKET_BYTE_9[7:0]			R/W
0xFAE C	00000000	This register is used to specify Spare2 Infoframe Packet, byte 9.	
SPARE_PACKET_2_PACKET_BYTE_10[7:0]			R/W
0xFAE D	00000000	This register is used to specify Spare2 Infoframe Packet, byte 10.	
SPARE_PACKET_2_PACKET_BYTE_11[7:0]			R/W
0xFAE E	00000000	This register is used to specify Spare2 Infoframe Packet, byte 11.	
SPARE_PACKET_2_PACKET_BYTE_12[7:0]			R/W
0xFAE F	00000000	This register is used to specify Spare2 Infoframe Packet, byte 12.	
SPARE_PACKET_2_PACKET_BYTE_13[7:0]			R/W
0xFAF 0	00000000	This register is used to specify Spare2 Infoframe Packet, byte 13.	
SPARE_PACKET_2_PACKET_BYTE_14[7:0]			R/W
0xFAF 1	00000000	This register is used to specify Spare2 Infoframe Packet, byte 14.	

Reg	Bits	Description	
SPARE_PACKET_2_PACKET_BYTE_15[7:0]			R/W
0xFAF2	00000000	This register is used to specify Spare2 Infoframe Packet, byte 15.	
SPARE_PACKET_2_PACKET_BYTE_16[7:0]			R/W
0xFAF3	00000000	This register is used to specify Spare2 Infoframe Packet, byte 16.	
SPARE_PACKET_2_PACKET_BYTE_17[7:0]			R/W
0xFAF4	00000000	This register is used to specify Spare2 Infoframe Packet, byte 17.	
SPARE_PACKET_2_PACKET_BYTE_18[7:0]			R/W
0xFAF5	00000000	This register is used to specify Spare2 Infoframe Packet, byte 18.	
SPARE_PACKET_2_PACKET_BYTE_19[7:0]			R/W
0xFAF6	00000000	This register is used to specify Spare2 Infoframe Packet, byte 19.	
SPARE_PACKET_2_PACKET_BYTE_20[7:0]			R/W
0xFAF7	00000000	This register is used to specify Spare2 Infoframe Packet, byte 20.	
SPARE_PACKET_2_PACKET_BYTE_21[7:0]			R/W
0xFAF8	00000000	This register is used to specify Spare2 Infoframe Packet, byte 21.	
SPARE_PACKET_2_PACKET_BYTE_22[7:0]			R/W
0xFAF9	00000000	This register is used to specify Spare2 Infoframe Packet, byte 22.	
SPARE_PACKET_2_PACKET_BYTE_23[7:0]			R/W
0xFAFA	00000000	This register is used to specify Spare2 Infoframe Packet, byte 23.	
SPARE_PACKET_2_PACKET_BYTE_24[7:0]			R/W
0xFAFB	00000000	This register is used to specify Spare2 Infoframe Packet, byte 24.	
SPARE_PACKET_2_PACKET_BYTE_25[7:0]			R/W
0xFAFC	00000000	This register is used to specify Spare2 Infoframe Packet, byte 25.	
SPARE_PACKET_2_PACKET_BYTE_26[7:0]			R/W
0xFAFD	00000000	This register is used to specify Spare2 Infoframe Packet, byte 26.	
SPARE_PACKET_2_PACKET_BYTE_27[7:0]			R/W
0xFAFE	00000000	This register is used to specify Spare2 Infoframe Packet, byte 27.	
SPARE2_UPDATE			R/W
0xFAFF	00000000	This bit is used to control the Spare2 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 2 via I2C is complete. 1 - Set this bit to 1 before updating the spare packet 2 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 2 packet during the vsync region, this spare packet 2 data transmitted will not be affected by the spare packet 2 data programmed via I2C. The spare packet 2 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE2_UPDATE being set back to 0	

2.13 TX2 EDID MAP

Reg	Bits	Description	
EDID_0[7:0]			R
0xF600	00000000		
EDID_1[7:0]			R
0xF601	00000000		
EDID_2[7:0]			R
0xF602	00000000		
EDID_3[7:0]			R
0xF603	00000000		
EDID_4[7:0]			R
0xF604	00000000		
EDID_5[7:0]			R
0xF605	00000000		
EDID_6[7:0]			R
0xF606	00000000		
EDID_7[7:0]			R
0xF607	00000000		
EDID_8[7:0]			R
0xF608	00000000		
EDID_9[7:0]			R
0xF609	00000000		
EDID_10[7:0]			R
0xF60A	00000000		
EDID_11[7:0]			R
0xF60B	00000000		
EDID_12[7:0]			R
0xF60C	00000000		
EDID_13[7:0]			R
0xF60D	00000000		
EDID_14[7:0]			R
0xF60E	00000000		
EDID_15[7:0]			R
0xF60F	00000000		
EDID_16[7:0]			R
0xF610	00000000		

Reg	Bits	Description	
EDID_17[7:0]			R
0xF611	00000000		
EDID_18[7:0]			R
0xF612	00000000		
EDID_19[7:0]			R
0xF613	00000000		
EDID_20[7:0]			R
0xF614	00000000		
EDID_21[7:0]			R
0xF615	00000000		
EDID_22[7:0]			R
0xF616	00000000		
EDID_23[7:0]			R
0xF617	00000000		
EDID_24[7:0]			R
0xF618	00000000		
EDID_25[7:0]			R
0xF619	00000000		
EDID_26[7:0]			R
0xF61A	00000000		
EDID_27[7:0]			R
0xF61B	00000000		
EDID_28[7:0]			R
0xF61C	00000000		
EDID_29[7:0]			R
0xF61D	00000000		
EDID_30[7:0]			R
0xF61E	00000000		
EDID_31[7:0]			R
0xF61F	00000000		
EDID_32[7:0]			R
0xF620	00000000		
EDID_33[7:0]			R
0xF621	00000000		
EDID_34[7:0]			R
0xF622	00000000		

Reg	Bits	Description	
EDID_35[7:0]			R
0xF623	00000000		
EDID_36[7:0]			R
0xF624	00000000		
EDID_37[7:0]			R
0xF625	00000000		
EDID_38[7:0]			R
0xF626	00000000		
EDID_39[7:0]			R
0xF627	00000000		
EDID_40[7:0]			R
0xF628	00000000		
EDID_41[7:0]			R
0xF629	00000000		
EDID_42[7:0]			R
0xF62A	00000000		
EDID_43[7:0]			R
0xF62B	00000000		
EDID_44[7:0]			R
0xF62C	00000000		
EDID_45[7:0]			R
0xF62D	00000000		
EDID_46[7:0]			R
0xF62E	00000000		
EDID_47[7:0]			R
0xF62F	00000000		
EDID_48[7:0]			R
0xF630	00000000		
EDID_49[7:0]			R
0xF631	00000000		
EDID_50[7:0]			R
0xF632	00000000		
EDID_51[7:0]			R
0xF633	00000000		
EDID_52[7:0]			R
0xF634	00000000		

Reg	Bits	Description	
EDID_53[7:0]			R
0xF635	00000000		
EDID_54[7:0]			R
0xF636	00000000		
EDID_55[7:0]			R
0xF637	00000000		
EDID_56[7:0]			R
0xF638	00000000		
EDID_57[7:0]			R
0xF639	00000000		
EDID_58[7:0]			R
0xF63A	00000000		
EDID_59[7:0]			R
0xF63B	00000000		
EDID_60[7:0]			R
0xF63C	00000000		
EDID_61[7:0]			R
0xF63D	00000000		
EDID_62[7:0]			R
0xF63E	00000000		
EDID_63[7:0]			R
0xF63F	00000000		
EDID_64[7:0]			R
0xF640	00000000		
EDID_65[7:0]			R
0xF641	00000000		
EDID_66[7:0]			R
0xF642	00000000		
EDID_67[7:0]			R
0xF643	00000000		
EDID_68[7:0]			R
0xF644	00000000		
EDID_69[7:0]			R
0xF645	00000000		
EDID_70[7:0]			R
0xF646	00000000		

Reg	Bits	Description	
EDID_71[7:0]			R
0xF647	00000000		
EDID_72[7:0]			R
0xF648	00000000		
EDID_73[7:0]			R
0xF649	00000000		
EDID_74[7:0]			R
0xF64A	00000000		
EDID_75[7:0]			R
0xF64B	00000000		
EDID_76[7:0]			R
0xF64C	00000000		
EDID_77[7:0]			R
0xF64D	00000000		
EDID_78[7:0]			R
0xF64E	00000000		
EDID_79[7:0]			R
0xF64F	00000000		
EDID_80[7:0]			R
0xF650	00000000		
EDID_81[7:0]			R
0xF651	00000000		
EDID_82[7:0]			R
0xF652	00000000		
EDID_83[7:0]			R
0xF653	00000000		
EDID_84[7:0]			R
0xF654	00000000		
EDID_85[7:0]			R
0xF655	00000000		
EDID_86[7:0]			R
0xF656	00000000		
EDID_87[7:0]			R
0xF657	00000000		
EDID_88[7:0]			R
0xF658	00000000		

Reg	Bits	Description	
EDID_89[7:0]			R
0xF659	00000000		
EDID_90[7:0]			R
0xF65A	00000000		
EDID_91[7:0]			R
0xF65B	00000000		
EDID_92[7:0]			R
0xF65C	00000000		
EDID_93[7:0]			R
0xF65D	00000000		
EDID_94[7:0]			R
0xF65E	00000000		
EDID_95[7:0]			R
0xF65F	00000000		
EDID_96[7:0]			R
0xF660	00000000		
EDID_97[7:0]			R
0xF661	00000000		
EDID_98[7:0]			R
0xF662	00000000		
EDID_99[7:0]			R
0xF663	00000000		
EDID_100[7:0]			R
0xF664	00000000		
EDID_101[7:0]			R
0xF665	00000000		
EDID_102[7:0]			R
0xF666	00000000		
EDID_103[7:0]			R
0xF667	00000000		
EDID_104[7:0]			R
0xF668	00000000		
EDID_105[7:0]			R
0xF669	00000000		
EDID_106[7:0]			R
0xF66A	00000000		

Reg	Bits	Description	
EDID_107[7:0]			R
0xF66B	00000000		
EDID_108[7:0]			R
0xF66C	00000000		
EDID_109[7:0]			R
0xF66D	00000000		
EDID_110[7:0]			R
0xF66E	00000000		
EDID_111[7:0]			R
0xF66F	00000000		
EDID_112[7:0]			R
0xF670	00000000		
EDID_113[7:0]			R
0xF671	00000000		
EDID_114[7:0]			R
0xF672	00000000		
EDID_115[7:0]			R
0xF673	00000000		
EDID_116[7:0]			R
0xF674	00000000		
EDID_117[7:0]			R
0xF675	00000000		
EDID_118[7:0]			R
0xF676	00000000		
EDID_119[7:0]			R
0xF677	00000000		
EDID_120[7:0]			R
0xF678	00000000		
EDID_121[7:0]			R
0xF679	00000000		
EDID_122[7:0]			R
0xF67A	00000000		
EDID_123[7:0]			R
0xF67B	00000000		
EDID_124[7:0]			R
0xF67C	00000000		

Reg	Bits	Description	
EDID_125[7:0]			R
0xF67D	00000000		
EDID_126[7:0]			R
0xF67E	00000000		
EDID_127[7:0]			R
0xF67F	00000000		
EDID_128[7:0]			R
0xF680	00000000		
EDID_129[7:0]			R
0xF681	00000000		
EDID_130[7:0]			R
0xF682	00000000		
EDID_131[7:0]			R
0xF683	00000000		
EDID_132[7:0]			R
0xF684	00000000		
EDID_133[7:0]			R
0xF685	00000000		
EDID_134[7:0]			R
0xF686	00000000		
EDID_135[7:0]			R
0xF687	00000000		
EDID_136[7:0]			R
0xF688	00000000		
EDID_137[7:0]			R
0xF689	00000000		
EDID_138[7:0]			R
0xF68A	00000000		
EDID_139[7:0]			R
0xF68B	00000000		
EDID_140[7:0]			R
0xF68C	00000000		
EDID_141[7:0]			R
0xF68D	00000000		
EDID_142[7:0]			R
0xF68E	00000000		

Reg	Bits	Description	
EDID_143[7:0]			R
0xF68F	00000000		
EDID_144[7:0]			R
0xF690	00000000		
EDID_145[7:0]			R
0xF691	00000000		
EDID_146[7:0]			R
0xF692	00000000		
EDID_147[7:0]			R
0xF693	00000000		
EDID_148[7:0]			R
0xF694	00000000		
EDID_149[7:0]			R
0xF695	00000000		
EDID_150[7:0]			R
0xF696	00000000		
EDID_151[7:0]			R
0xF697	00000000		
EDID_152[7:0]			R
0xF698	00000000		
EDID_153[7:0]			R
0xF699	00000000		
EDID_154[7:0]			R
0xF69A	00000000		
EDID_155[7:0]			R
0xF69B	00000000		
EDID_156[7:0]			R
0xF69C	00000000		
EDID_157[7:0]			R
0xF69D	00000000		
EDID_158[7:0]			R
0xF69E	00000000		
EDID_159[7:0]			R
0xF69F	00000000		
EDID_160[7:0]			R
0xF6A0	00000000		

Reg	Bits	Description	
EDID_161[7:0]			R
0xF6A1	00000000		
EDID_162[7:0]			R
0xF6A2	00000000		
EDID_163[7:0]			R
0xF6A3	00000000		
EDID_164[7:0]			R
0xF6A4	00000000		
EDID_165[7:0]			R
0xF6A5	00000000		
EDID_166[7:0]			R
0xF6A6	00000000		
EDID_167[7:0]			R
0xF6A7	00000000		
EDID_168[7:0]			R
0xF6A8	00000000		
EDID_169[7:0]			R
0xF6A9	00000000		
EDID_170[7:0]			R
0xF6AA	00000000		
EDID_171[7:0]			R
0xF6AB	00000000		
EDID_172[7:0]			R
0xF6AC	00000000		
EDID_173[7:0]			R
0xF6AD	00000000		
EDID_174[7:0]			R
0xF6AE	00000000		
EDID_175[7:0]			R
0xF6AF	00000000		
EDID_176[7:0]			R
0xF6B0	00000000		
EDID_177[7:0]			R
0xF6B1	00000000		
EDID_178[7:0]			R
0xF6B2	00000000		

Reg	Bits	Description	
EDID_179[7:0]			R
0xF6B3	00000000		
EDID_180[7:0]			R
0xF6B4	00000000		
EDID_181[7:0]			R
0xF6B5	00000000		
EDID_182[7:0]			R
0xF6B6	00000000		
EDID_183[7:0]			R
0xF6B7	00000000		
EDID_184[7:0]			R
0xF6B8	00000000		
EDID_185[7:0]			R
0xF6B9	00000000		
EDID_186[7:0]			R
0xF6BA	00000000		
EDID_187[7:0]			R
0xF6BB	00000000		
EDID_188[7:0]			R
0xF6BC	00000000		
EDID_189[7:0]			R
0xF6BD	00000000		
EDID_190[7:0]			R
0xF6BE	00000000		
EDID_191[7:0]			R
0xF6BF	00000000		
EDID_192[7:0]			R
0xF6C0	00000000		
EDID_193[7:0]			R
0xF6C1	00000000		
EDID_194[7:0]			R
0xF6C2	00000000		
EDID_195[7:0]			R
0xF6C3	00000000		
EDID_196[7:0]			R
0xF6C4	00000000		

Reg	Bits	Description	
EDID_197[7:0]			R
0xF6C5	00000000		
EDID_198[7:0]			R
0xF6C6	00000000		
EDID_199[7:0]			R
0xF6C7	00000000		
EDID_200[7:0]			R
0xF6C8	00000000		
EDID_201[7:0]			R
0xF6C9	00000000		
EDID_202[7:0]			R
0xF6CA	00000000		
EDID_203[7:0]			R
0xF6CB	00000000		
EDID_204[7:0]			R
0xF6CC	00000000		
EDID_205[7:0]			R
0xF6CD	00000000		
EDID_206[7:0]			R
0xF6CE	00000000		
EDID_207[7:0]			R
0xF6CF	00000000		
EDID_208[7:0]			R
0xF6D0	00000000		
EDID_209[7:0]			R
0xF6D1	00000000		
EDID_210[7:0]			R
0xF6D2	00000000		
EDID_211[7:0]			R
0xF6D3	00000000		
EDID_212[7:0]			R
0xF6D4	00000000		
EDID_213[7:0]			R
0xF6D5	00000000		
EDID_214[7:0]			R
0xF6D6	00000000		

Reg	Bits	Description	
EDID_215[7:0]			R
0xF6D7	00000000		
EDID_216[7:0]			R
0xF6D8	00000000		
EDID_217[7:0]			R
0xF6D9	00000000		
EDID_218[7:0]			R
0xF6DA	00000000		
EDID_219[7:0]			R
0xF6DB	00000000		
EDID_220[7:0]			R
0xF6DC	00000000		
EDID_221[7:0]			R
0xF6DD	00000000		
EDID_222[7:0]			R
0xF6DE	00000000		
EDID_223[7:0]			R
0xF6DF	00000000		
EDID_224[7:0]			R
0xF6E0	00000000		
EDID_225[7:0]			R
0xF6E1	00000000		
EDID_226[7:0]			R
0xF6E2	00000000		
EDID_227[7:0]			R
0xF6E3	00000000		
EDID_228[7:0]			R
0xF6E4	00000000		
EDID_229[7:0]			R
0xF6E5	00000000		
EDID_230[7:0]			R
0xF6E6	00000000		
EDID_231[7:0]			R
0xF6E7	00000000		
EDID_232[7:0]			R
0xF6E8	00000000		

Reg	Bits	Description	
EDID_233[7:0]			R
0xF6E9	00000000		
EDID_234[7:0]			R
0xF6EA	00000000		
EDID_235[7:0]			R
0xF6EB	00000000		
EDID_236[7:0]			R
0xF6EC	00000000		
EDID_237[7:0]			R
0xF6ED	00000000		
EDID_238[7:0]			R
0xF6EE	00000000		
EDID_239[7:0]			R
0xF6EF	00000000		
EDID_240[7:0]			R
0xF6F0	00000000		
EDID_241[7:0]			R
0xF6F1	00000000		
EDID_242[7:0]			R
0xF6F2	00000000		
EDID_243[7:0]			R
0xF6F3	00000000		
EDID_244[7:0]			R
0xF6F4	00000000		
EDID_245[7:0]			R
0xF6F5	00000000		
EDID_246[7:0]			R
0xF6F6	00000000		
EDID_247[7:0]			R
0xF6F7	00000000		
EDID_248[7:0]			R
0xF6F8	00000000		
EDID_249[7:0]			R
0xF6F9	00000000		
EDID_250[7:0]			R
0xF6FA	00000000		

Reg	Bits	Description	
EDID_251[7:0]			R
0xF6FB	00000000		
EDID_252[7:0]			R
0xF6FC	00000000		
EDID_253[7:0]			R
0xF6FD	00000000		
EDID_254[7:0]			R
0xF6FE	00000000		
EDID_255[7:0]			R
0xF6FF	00000000		

2.14 TX2 TEST MAP

Reg	Bits	Description	
SPARE_PKT3_EN			R/W
0xFBBF	0000000	This bit is used to enable the Spare Packet 3. 0 - Disabled 1 - Enabled	
SPARE_PKT4_EN			R/W
0xFBBF	0000000	This bit is used to enable the Spare Packet 4. 0 - Disabled 1 - Enabled	
SPARE3_HEADER0[7:0]			R/W
0xFBC0	00000000	This register is used to specify Spare3 Infoframe Header, byte 0.	
SPARE3_HEADER1[7:0]			R/W
0xFBC1	00000000	This register is used to specify Spare3 Infoframe Header, byte 1.	
SPARE3_HEADER2[7:0]			R/W
0xFBC2	00000000	This register is used to specify Spare3 Infoframe Header, byte 2.	
SPARE3_BYTE0[7:0]			R/W
0xFBC3	00000000	This register is used to specify Spare3 Infoframe Packet, byte 0.	
SPARE3_BYTE1[7:0]			R/W
0xFBC4	00000000	This register is used to specify Spare3 Infoframe Packet, byte 1.	
SPARE3_BYTE2[7:0]			R/W
0xFBC5	00000000	This register is used to specify Spare3 Infoframe Packet, byte 2.	
SPARE3_BYTE3[7:0]			R/W
0xFBC6	00000000	This register is used to specify Spare3 Infoframe Packet, byte 3.	
SPARE3_BYTE4[7:0]			R/W
0xFBC7	00000000	This register is used to specify Spare3 Infoframe Packet, byte 4.	
SPARE3_BYTE5[7:0]			R/W
0xFBC8	00000000	This register is used to specify Spare3 Infoframe Packet, byte 5.	
SPARE3_BYTE6[7:0]			R/W
0xFBC9	00000000	This register is used to specify Spare3 Infoframe Packet, byte 6.	
SPARE3_BYTE7[7:0]			R/W
0xFBCA	00000000	This register is used to specify Spare3 Infoframe Packet, byte 7.	
SPARE3_BYTE8[7:0]			R/W
0xFBCB	00000000	This register is used to specify Spare3 Infoframe Packet, byte 8.	
SPARE3_BYTE9[7:0]			R/W
0xFBCC	00000000	This register is used to specify Spare3 Infoframe Packet, byte 9.	
SPARE3_BYTE10[7:0]			R/W
0xFBCD	00000000	This register is used to specify Spare3 Infoframe Packet, byte 10.	

Reg	Bits	Description	
SPARE3_BYTE11[7:0]			R/W
0xFBCE	00000000	This register is used to specify Spare3 Infoframe Packet, byte 11.	
SPARE3_BYTE12[7:0]			R/W
0xFBCF	00000000	This register is used to specify Spare3 Infoframe Packet, byte 12.	
SPARE3_BYTE13[7:0]			R/W
0xFBD0	00000000	This register is used to specify Spare3 Infoframe Packet, byte 13.	
SPARE3_BYTE14[7:0]			R/W
0xFBD1	00000000	This register is used to specify Spare3 Infoframe Packet, byte 14.	
SPARE3_BYTE15[7:0]			R/W
0xFBD2	00000000	This register is used to specify Spare3 Infoframe Packet, byte 15.	
SPARE3_BYTE16[7:0]			R/W
0xFBD3	00000000	This register is used to specify Spare3 Infoframe Packet, byte 16.	
SPARE3_BYTE17[7:0]			R/W
0xFBD4	00000000	This register is used to specify Spare3 Infoframe Packet, byte 17.	
SPARE3_BYTE18[7:0]			R/W
0xFBD5	00000000	This register is used to specify Spare3 Infoframe Packet, byte 18.	
SPARE3_BYTE19[7:0]			R/W
0xFBD6	00000000	This register is used to specify Spare3 Infoframe Packet, byte 19.	
SPARE3_BYTE20[7:0]			R/W
0xFBD7	00000000	This register is used to specify Spare3 Infoframe Packet, byte 20.	
SPARE3_BYTE21[7:0]			R/W
0xFBD8	00000000	This register is used to specify Spare3 Infoframe Packet, byte 21.	
SPARE3_BYTE22[7:0]			R/W
0xFBD9	00000000	This register is used to specify Spare3 Infoframe Packet, byte 22.	
SPARE3_BYTE23[7:0]			R/W
0xFBDA	00000000	This register is used to specify Spare3 Infoframe Packet, byte 23.	
SPARE3_BYTE24[7:0]			R/W
0xFBDB	00000000	This register is used to specify Spare3 Infoframe Packet, byte 24.	
SPARE3_BYTE25[7:0]			R/W
0xFBDC	00000000	This register is used to specify Spare3 Infoframe Packet, byte 25.	
SPARE3_BYTE26[7:0]			R/W
0xFBDD	00000000	This register is used to specify Spare3 Infoframe Packet, byte 26.	
SPARE3_BYTE27[7:0]			R/W
0xFBDE	00000000	This register is used to specify Spare3 Infoframe Packet, byte 27.	

Reg	Bits	Description
SPARE3_UPDATE		R/W
0xFBDF	00000000	This bit is used to control the Spare3 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 3 via I2C is complete 1 - Set this bit to 1 before updating the spare packet 3 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 3 packet during the vsync region, this spare packet 3 data transmitted will not be affected by the spare packet 3 data programmed via I2C. The spare packet 3 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE3_UPDATE being set back to 0
SPARE4_HEADER0[7:0]		R/W
0xFBE0	00000000	This register is used to specify Spare4 Infoframe Header, byte 0.
SPARE4_HEADER1[7:0]		R/W
0xFBE1	00000000	This register is used to specify Spare4 Infoframe Header, byte 1.
SPARE4_HEADER2[7:0]		R/W
0xFBE2	00000000	This register is used to specify Spare4 Infoframe Header, byte 2.
SPARE4_BYTE0[7:0]		R/W
0xFBE3	00000000	This register is used to specify Spare4 Infoframe Packet, byte 0.
SPARE4_BYTE1[7:0]		R/W
0xFBE4	00000000	This register is used to specify Spare4 Infoframe Packet, byte 1.
SPARE4_BYTE2[7:0]		R/W
0xFBE5	00000000	This register is used to specify Spare4 Infoframe Packet, byte 2.
SPARE4_BYTE3[7:0]		R/W
0xFBE6	00000000	This register is used to specify Spare4 Infoframe Packet, byte 3.
SPARE4_BYTE4[7:0]		R/W
0xFBE7	00000000	This register is used to specify Spare4 Infoframe Packet, byte 4.
SPARE4_BYTE5[7:0]		R/W
0xFBE8	00000000	This register is used to specify Spare4 Infoframe Packet, byte 5.
SPARE4_BYTE6[7:0]		R/W
0xFBE9	00000000	This register is used to specify Spare4 Infoframe Packet, byte 6.
SPARE4_BYTE7[7:0]		R/W
0xFBEA	00000000	This register is used to specify Spare4 Infoframe Packet, byte 7.
SPARE4_BYTE8[7:0]		R/W
0xFBEB	00000000	This register is used to specify Spare4 Infoframe Packet, byte 8.
SPARE4_BYTE9[7:0]		R/W
0xFBEC	00000000	This register is used to specify Spare4 Infoframe Packet, byte 9.
SPARE4_BYTE10[7:0]		R/W
0xFBED	00000000	This register is used to specify Spare4 Infoframe Packet, byte 10.
SPARE4_BYTE11[7:0]		R/W
0xFBEE	00000000	This register is used to specify Spare4 Infoframe Packet, byte 11.
SPARE4_BYTE12[7:0]		R/W
0xFBEF	00000000	This register is used to specify Spare4 Infoframe Packet, byte 12.

Reg	Bits	Description	
SPARE4_BYTE13[7:0]			R/W
0xFBF 0	00000000	This register is used to specify Spare4 Infoframe Packet, byte 13.	
SPARE4_BYTE14[7:0]			R/W
0xFBF 1	00000000	This register is used to specify Spare4 Infoframe Packet, byte 14.	
SPARE4_BYTE15[7:0]			R/W
0xFBF 2	00000000	This register is used to specify Spare4 Infoframe Packet, byte 15.	
SPARE4_BYTE16[7:0]			R/W
0xFBF 3	00000000	This register is used to specify Spare4 Infoframe Packet, byte 16.	
SPARE4_BYTE17[7:0]			R/W
0xFBF 4	00000000	This register is used to specify Spare4 Infoframe Packet, byte 17.	
SPARE4_BYTE18[7:0]			R/W
0xFBF 5	00000000	This register is used to specify Spare4 Infoframe Packet, byte 18.	
SPARE4_BYTE19[7:0]			R/W
0xFBF 6	00000000	This register is used to specify Spare4 Infoframe Packet, byte 19.	
SPARE4_BYTE20[7:0]			R/W
0xFBF 7	00000000	This register is used to specify Spare4 Infoframe Packet, byte 20.	
SPARE4_BYTE21[7:0]			R/W
0xFBF 8	00000000	This register is used to specify Spare4 Infoframe Packet, byte 21.	
SPARE4_BYTE22[7:0]			R/W
0xFBF 9	00000000	This register is used to specify Spare4 Infoframe Packet, byte 22.	
SPARE4_BYTE23[7:0]			R/W
0xFBF A	00000000	This register is used to specify Spare4 Infoframe Packet, byte 23.	
SPARE4_BYTE24[7:0]			R/W
0xFBF B	00000000	This register is used to specify Spare4 Infoframe Packet, byte 24.	
SPARE4_BYTE25[7:0]			R/W
0xFBF C	00000000	This register is used to specify Spare4 Infoframe Packet, byte 25.	
SPARE4_BYTE26[7:0]			R/W
0xFBF D	00000000	This register is used to specify Spare4 Infoframe Packet, byte 26.	
SPARE4_BYTE27[7:0]			R/W
0xFBF E	00000000	This register is used to specify Spare4 Infoframe Packet, byte 27.	
SPARE4_UPDATE			R/W
0xFBF F	00000000	This bit is used to control the Spare4 Infoframe protection feature. 0 - Set this bit to 0 when the update of the spare packet 4 via I2C is complete 1 - Set this bit to 1 before updating the spare packet 4 data via I2C. If this bit is set high while the Tx is transmitting a spare packet 4 packet during the vsync region, this spare packet 4 data transmitted will not be affected by the spare packet 4 data programmed via I2C. The spare packet 4 data transmitted by the Tx will be updated with the newly programmed I2C data in the vsync period following SPARE4_UPDATE being set back to 0	

2.15 ENCODER MAP

Reg	Bits	Description	
DAC4_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 4. 0 - DAC 4 off 1 - DAC 4 on	
DAC5_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 5. 0 - DAC 5 off 1 - DAC 5 on	
DAC6_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 6. 0 - DAC 6 off 1 - DAC 6 on	
DAC1_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 1. 0 - DAC 1 off 1 - DAC 1 on	
DAC2_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 2. 0 - DAC 2 off 1 - DAC 2 on	
DAC3_ENABLE			R/W
0xE400	00010010	This bit is used to power on/off DAC 3. 0 - DAC 3 off 1 - DAC 3 on	
PLL_PDN			R/W
0xE400	00010010	This bit is used to control the PLL and oversampling. This control allows the internal PLL 1 circuit to be powered down and the oversampling feature to be switched off. By default this is disabled, setting this bit to 0 enables this feature. 0 - PLL On 1 - PLL Off	
SLEEP_MODE			R/W
0xE400	00010010	This bit is used to enable sleep mode. With this control enabled, the current consumption in the encoder is reduced to µA level. All DACs and the internal PLL circuits are disabled. Registers can be read from and written to in sleep mode. 0 - Sleep Mode Off 1 - Sleep Mode On	
FUNC_MODE[2:0]			R/W
0xE401	00000000	This signal is used to select the input mode to the encoder. 000 - SD Input Only 001 - ED/HD-SDR input only 010 - Reserved 011 - Simultaneous SD and ED/HD-SDR 100 - Reserved 101 - Reserved 110 - Reserved 111 - Reserved	
YUV_OUT			R/W
0xE402	00100000	This bit is used to select the output colour space for the encoder. 0 - RGB component outputs. 1 - YPrPb component outputs.	
RGB_SYNC_EN			R/W
0xE402	00100000	This bit is used to control whether sync signals are sent on the RGB channels. 0 - No sync on any of the RGB outputs 1 - Sync on all RGB outputs.	

Reg	Bits	Description
MATRIX_PROG_EN		R/W
0xE402	00100000	This bit is used to enable the manual mode for the ED/HD colour space converter. 0 - Automatic Mode 1 - Manual Mode
CRT_BLK_EN		R/W
0xE402	00100000	This bit is used to enable the black bar test pattern. 0 - Disabled 1 - Enabled
MATRIX_GY[9:0]		R/W
0xE403 0xE405	00000011 01001110	This signal is used to set the ED/HD Manual CSC Matrix Adjust Feature: GY value. 00 - None 01 - None 10 - None 11 - None
MATRIX_GU[9:0]		R/W
0xE404 0xE406	11110000 00001110	This signal is used to set the ED/HD Manual CSC Matrix Adjust Feature: GU value. 00 - None 01 - None 10 - None 11 - None
MATRIX_GV[9:0]		R/W
0xE404 0xE407	11110000 00100100	This signal is used to set the ED/HD Manual CSC Matrix Adjust Feature: GV value. 00 - None 01 - None 10 - None 11 - None
MATRIX_BU[9:0]		R/W
0xE404 0xE408	11110000 10010010	This signal is used to set the ED/HD Manual CSC Matrix Adjust Feature: BU value. 00 - None 01 - None 10 - None 11 - None
MATRIX_RV[9:0]		R/W
0xE404 0xE409	11110000 01111100	This signal is used to set the ED/HD Manual CSC Matrix Adjust Feature: RV value. 00 - None 01 - None 10 - None 11 - None
DAC4TO6_TUNING[7:0]		R/W
0xE40A	00000000	This register is used to set the gain for DACs 4-6 output voltage. 11000000 - -7.5% 11000001 - -7.382% 11000010 - -7.364% - ... 11111111 - -0.018% 00000000 - 0% 00000001 - 0.018% 00000010 - 0.036% - ... 00111111 - +7.382% 01000000 - +7.5%

Reg	Bits	Description	
DAC1TO3_TUNING[7:0]			R/W
0xE40B	00000000	This register is used to set the gain for DACs 1-3 output voltage. 11000000 - -7.5% 11000001 - -7.382% 11000010 - -7.364% - ... 11111111 - -0.018% 00000000 - 0% 00000001 - 0.018% 00000010 - 0.036% - ... 00111111 - +7.382% 01000000 - +7.5%	
LP_EN_F			R/W
0xE40D	0000000	This bit is used to enable the low power mode on DAC3. By default this is set to 0 which means low power mode is disabled. 0 - DAC 3 Low Power Mode Disabled 1 - DAC 3 Low Power Mode Enabled	
LP_EN_E			R/W
0xE40D	000000	This bit is used to enable the low power mode on DAC2. By default this is set to 0 which means low power mode is disabled. 0 - DAC 2 Low Power Mode Disabled 1 - DAC 2 Low Power Mode Enabled	
LP_EN_D			R/W
0xE40D	00000000	This bit is used to enable the low power mode on DAC1. By default this is set to 0 which means low power mode is disabled. 0 - DAC 1 Low Power Mode Disabled 1 - DAC 1 Low Power Mode Enabled	
VBI_FROM_ANC_IN			R/W
0xE418	00000000	0 - Use CGMS/WSS/CCAP data from i2c 1 - Use CGMS/WSS/CCAP data provided externally	
DAC1_SEL[2:0]			R/W
0xE429	00000001	This signal selects the data that is supplied to DAC 1. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R	
DAC2_SEL[2:0]			R/W
0xE429	00000001	This signal selects the data that is supplied to DAC 2. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R	
DAC3_SEL[2:0]			R/W
0xE42A	00100011	This signal selects the data that is supplied to DAC 3. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R	

Reg	Bits	Description
DAC4_SEL[2:0]		R/W
0xE42A	00100011	This signal selects the data that is supplied to DAC 4. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R
DAC5_SEL[2:0]		R/W
0xE42B	01000101	This signal selects the data that is supplied to DAC 5. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R
DAC6_SEL[2:0]		R/W
0xE42B	01000101	This signal selects the data that is supplied to DAC 6. 0 - CVBS or Black Burst 1 - Luma 2 - Chroma 3 - Y/G 4 - Pb/B 5 - Pr/R
HD_ENC_IP_MODE[4:0]		R/W
0xE430	00000000	This signal is used to select the ED/HD output standard. 00000 - SMPTE293M-1996 483P 60/1.001 OR ITU-R BT.1358 483P 60/1.001 00010 - BTA T-1004 EDTV2 483P 60/1.001 OR ITU-R BT.1362 483P 60/1.001 00011 - ITU-R BT.1358 576P 50 00100 - ITU-R BT.1362 576P 50 00101 - SMPTE296M-2001(1) 720P 60 OR SMPTE296M-2001(2) 720P 60/1.001 00110 - SMPTE296M-2001(3) 720P 50 00111 - SMPTE296M-2001(4) 720P 30 OR SMPTE296M-2001(5) 720P 30/1.001 01000 - SMPTE296M-2001(6) 720P 25 OR 01001 - SMPTE296M-2001(7) 720P 24 OR SMPTE296M-2001(8) 720P 24/1.001 01010 - SMPTE240M-1999 1035I 30 OR SMPTE240M-1999 1035I 30/1.001 01011 - SMPTE274-1998(1) 1080P 60 OR SMPTE274-1998(2) 1080P 60/1.001 01100 - SMPTE274-1998(3) 1080P 50 01101 - SMPTE274-1998(4) 1080I 30 OR SMPTE274-1998(5) 1080I 30/1.001 01110 - SMPTE274-1998(6) 1080I 25 01111 - SMPTE274-1998(7) 1080P 30 OR SMPTE274-1998(8) 1080P 30/1.001 10000 - SMPTE274-1998(9) 1080P 25 10001 - SMPTE274-1998(10) 1080P 24 OR SMPTE274-1998(11) 1080P 24/1.001 10011 - SMPTE295-1997 1080I 25 10100 - SMPTE295-1997 1080P 50 10110 - ITU-R BT.709-5 1152I 50
ENCODE_MODE_HDTV[1:0]		R/W
0xE430	00000000	This signal is used to set the ED/HD output standard. 00 - EIA770.2 output, EIA770.3 output (ED, HD) 01 - EIA770.1 output 10 - Output levels for full input range 11 - Reserved
SHARP_EN		R/W
0xE431	00000000	This bit is used to enable the ED/HD sharpness filter on the luma data. By default this is set to 0 which means the filter is disabled. 0 - Disabled 1 - Enabled
LIMIT_SEL_HDTV[1:0]		R/W
0xE431	00000000	This bit is used to configure the ED/HD undershoot limiter. 00 - Disabled 01 - -11IRE 10 - -6IRE 11 - -1.5IRE

Reg	Bits	Description
VBI_DATA_EN		R/W
0xE431	0000**0**0000	This bit is used to enable data on the Vertical Blanking Interval (VBI) to be accepted as valid data. This is valid for ED/HD video data only. 0 - Disabled 1 - Enabled
HDTV_FLAT_TP		R/W
0xE431	0000**0**000	This bit is used to select the pattern used by the internal test pattern generator. 0 - Hatch 1 - Field/frame
HDTV_TP_EN		R/W
0xE431	00000**0**00	This bit is used to enable the ED/HD test pattern generator. 0 - ED/HD test pattern off 1 - ED/HD test pattern on.
PIXEL_DATA_EN		R/W
0xE431	0000000**0**	This bit is used to select whether ED/HD pixel data is valid. 0 - Pixel data valid off 1 - Pixel data valid on
CGMS_CRC_HDTV		R/W
0xE432	**0**0000000	This bit is used to enable the ED/HD CGMS cyclic redundancy check (CRC). 0 - Disabled 1 - Enabled
CGMS_EN_HDTV		R/W
0xE432	0**0**000000	This bit is used to enable the ED/HD CGMS. 0 - Disabled 1 - Enabled
UV_DEL_HDTV[2:0]		R/W
0xE432	00**000**000	This signal is used to set the ED/HD color delay with respect to the falling edge of HSYNC. 000 - 0 Clock Cycles 001 - 1 Clock Cycles 010 - 2 Clock Cycles 011 - 3 Clock Cycles 100 - 4 Clock Cycles
Y_DEL_HDTV[2:0]		R/W
0xE432	00000**000**	This signal is used to set the ED/HD Y delay with respect to the falling edge of HSYNC. 000 - 0 Clock Cycles 001 - 1 Clock Cycles 010 - 2 Clock Cycles 011 - 3 Clock Cycles 100 - 4 Clock Cycles
DB_EN_HDTV		R/W
0xE433	**0**1101000	This bit is used to enable the double buffering on the appropriate ED/HD registers. 0 - Cb after falling edge of HSYNC 1 - Cr after falling edge of HSYNC
SSAF_422		R/W
0xE433	01**1**01000	This bit is used to enable the ED/HD chroma SSAF. 0 - Disabled 1 - Enabled
SINC_FILT_EN		R/W
0xE433	0110**1**000	This bit is used to disable the sinc compensation filter on DAC1, DAC2 and DAC3. 0 - Disabled 1 - Enabled
NO_WRAPAROUND		R/W
0xE434	**0**1001000	This bit is used to configure the horizontal/vertical counters. 0 - Update field/line counter 1 - Field/line counter free running

Reg	Bits	Description	
ADAPT_EN			R/W
0xE435	00000000	This bit is used to enable the ED/HD adaptive filter. 0 - Disabled 1 - Enabled	
ADAPT_BC			R/W
0xE435	00000000	This bit is used to select the adaptive filter mode. 0 - Mode A 1 - Mode B	
GAMMA_EN_HDTV			R/W
0xE435	00000000	This bit is used to enable the gamma correction curves for ED/HD video data. 0 - Disabled 1 - Enabled	
GAMMA_CURVE_B_HDTV			R/W
0xE435	00000000	This bit is used to select the gamma correction curves for ED/HD video data. 0 - Gamma Correction Curve A 1 - Gamma Correction Curve B	
SYNC_ON_PRPB			R/W
0xE435	00000000	This bit is used to enable the ED/HD sync on PrPb. 0 - Disabled 1 - Enabled	
Y_COLOUR[7:0]			R/W
0xE436	10100000	This register is used to control the ED/HD test pattern, Y level.	
CR_COLOUR[7:0]			R/W
0xE437	10000000	This register is used to control the ED/HD test pattern, Cr level.	
CB_COLOUR[7:0]			R/W
0xE438	10000000	This register is used to control the ED/HD test pattern, Cb level.	
EIA_CEA_861_MODE_HD			R/W
0xE439	00000000	This bit is used to enable the ED/HD EIA/CEA-861 synchronization compliance. 0 - Disabled 1 - Enabled	
KB[3:0]			R/W
0xE440	00000000	This signal is used to configure the ED/HD sharpness filter gain, value B. 0000 - Gain B 0 0001 - Gain B +1 - ... 0111 - Gain B +7 1000 - Gain B -8 - ... 1110 - Gain B -2 1111 - Gain B -1	
KA[3:0]			R/W
0xE440	00000000	This signal is used to configure the ED/HD sharpness filter gain, value A. 0000 - Gain A 0 0001 - Gain A +1 - ... 0111 - Gain A +7 1000 - Gain A -8 - ... 1110 - Gain A -2 1111 - Gain A -1	

Reg	Bits	Description	
CGMS_HDTV[19:0]			R/W
0xE441 0xE442 0xE443	00000000 00000000 00000000	This signal is used to set the ED/HD CGMS data bits (CGMS C19 to C0).	
GAMMA_A0_HDTV[7:0]			R/W
0xE444	00000000	This register is used to configure the ED/HD gamma curve A (A0 - Point 24).	
GAMMA_A1_HDTV[7:0]			R/W
0xE445	00000000	This register is used to configure the ED/HD gamma curve A (A1 - Point 32).	
GAMMA_A2_HDTV[7:0]			R/W
0xE446	00000000	This register is used to configure the ED/HD gamma curve A (A2 - Point 48).	
GAMMA_A3_HDTV[7:0]			R/W
0xE447	00000000	This register is used to configure the ED/HD gamma curve A (A3 - Point 64).	
GAMMA_A4_HDTV[7:0]			R/W
0xE448	00000000	This register is used to configure the ED/HD gamma curve A (A4 - Point 80).	
GAMMA_A5_HDTV[7:0]			R/W
0xE449	00000000	This register is used to configure the ED/HD gamma curve A (A5 - Point 96).	
GAMMA_A6_HDTV[7:0]			R/W
0xE44A	00000000	This register is used to configure the ED/HD gamma curve A (A6 - Point 128).	
GAMMA_A7_HDTV[7:0]			R/W
0xE44B	00000000	This register is used to configure the ED/HD gamma curve A (A7 - Point 160).	
GAMMA_A8_HDTV[7:0]			R/W
0xE44C	00000000	This register is used to configure the ED/HD gamma curve A (A8 - Point 192).	
GAMMA_A9_HDTV[7:0]			R/W
0xE44D	00000000	This register is used to configure the ED/HD gamma curve A (A9 - Point 224).	
GAMMA_B0_HDTV[7:0]			R/W
0xE44E	00000000	This register is used to configure the ED/HD gamma curve B (B0 - Point 24).	
GAMMA_B1_HDTV[7:0]			R/W
0xE44F	00000000	This register is used to configure the ED/HD gamma curve B (B1 - Point 32).	
GAMMA_B2_HDTV[7:0]			R/W
0xE450	00000000	This register is used to configure the ED/HD gamma curve B (B2 - Point 48).	
GAMMA_B3_HDTV[7:0]			R/W
0xE451	00000000	This register is used to configure the ED/HD gamma curve B (B3 - Point 64).	
GAMMA_B4_HDTV[7:0]			R/W
0xE452	00000000	This register is used to configure the ED/HD gamma curve B (B4 - Point 80).	
GAMMA_B5_HDTV[7:0]			R/W
0xE453	00000000	This register is used to configure the ED/HD gamma curve B (B5 - Point 96).	

Reg	Bits	Description	
GAMMA_B6_HDTV[7:0]			R/W
0xE454	00000000	This register is used to configure the ED/HD gamma curve B (B6 - Point 128).	
GAMMA_B7_HDTV[7:0]			R/W
0xE455	00000000	This register is used to configure the ED/HD gamma curve B (B7 - Point 160).	
GAMMA_B8_HDTV[7:0]			R/W
0xE456	00000000	This register is used to configure the ED/HD gamma curve B (B8 - Point 192).	
GAMMA_B9_HDTV[7:0]			R/W
0xE457	00000000	This register is used to configure the ED/HD gamma curve B (B9 - Point 224).	
FIL_RESP_AB[3:0]			R/W
0xE458	00000000	This signal is used to set the adaptive filter gain 1 for the ED/HD standard. This is value B. 0000 - Gain B 0 0001 - Gain B +1 - ... 0111 - Gain B +7 1000 - Gain B -8 - ... 1110 - Gain B -2 1111 - Gain B -1	
FIL_RESP_AA[3:0]			R/W
0xE458	00000000	This signal is used to set the adaptive filter gain 1 for the ED/HD standard. This is value A. 0000 - Gain A 0 0001 - Gain A +1 - ... 0111 - Gain A +7 1000 - Gain A -8 - ... 1110 - Gain A -2 1111 - Gain A -1	
FIL_RESP_BB[3:0]			R/W
0xE459	00000000	This signal is used to set the adaptive filter gain 2 for the ED/HD standard. This is value B. 0000 - Gain B 0 0001 - Gain B +1 - ... 0111 - Gain B +7 1000 - Gain B -8 - ... 1110 - Gain B -2 1111 - Gain B -1	
FIL_RESP_BA[3:0]			R/W
0xE459	00000000	This signal is used to set the adaptive filter gain 2 for the ED/HD standard. This is value A. 0000 - Gain A 0 0001 - Gain A +1 - ... 0111 - Gain A +7 1000 - Gain A -8 - ... 1110 - Gain A -2 1111 - Gain A -1	
FIL_RESP_CB[3:0]			R/W
0xE45A	00000000	This signal is used to set the adaptive filter gain 3 for the ED/HD standard. This is value B. 0000 - Gain B 0 0001 - Gain B +1 - ... 0111 - Gain B +7 1000 - Gain B -8 - ... 1110 - Gain B -2 1111 - Gain B -1	

Reg	Bits	Description	
FIL_RESP_CA[3:0]			R/W
0xE45A	0000<u>0000</u>	This signal is used to set the adaptive filter gain 3 for the ED/HD standard. This is value A. 0000 - Gain A 0 0001 - Gain A +1 - ... 0111 - Gain A +7 1000 - Gain A -8 - ... 1110 - Gain A -2 1111 - Gain A -1	
THOLD_A[7:0]			R/W
0xE45B	<u>00000000</u>	This register is used to set the ED/HD adaptive filter threshold A.	
THOLD_B[7:0]			R/W
0xE45C	<u>00000000</u>	This register is used to set the ED/HD adaptive filter threshold B.	
THOLD_C[7:0]			R/W
0xE45D	<u>00000000</u>	This register is used to set the ED/HD adaptive filter threshold C.	
CGMSB_0[5:0]			R/W
0xE45E	<u>000000</u>00	This signal is used to set the ED/HD CGMS Type B header bits (H5 to H0).	
INTERNAL_CRC_ENABLE			R/W
0xE45E	000000<u>0</u>0	This bit is used to enable the ED/HD CGMS Type B cyclic redundancy check (CRC). 1 - Enabled 0 - Disabled	
CGMS_B_ENABLE			R/W
0xE45E	0000000<u>0</u>	This bit is used to enable the transmission of ED/HD CGMS Type B. 1 - Enabled 0 - Disabled	
CGMSB_1[7:0]			R/W
0xE45F	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P7 to P0).	
CGMSB_2[7:0]			R/W
0xE460	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P15 to P8).	
CGMSB_3[7:0]			R/W
0xE461	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P23 to P16).	
CGMSB_4[7:0]			R/W
0xE462	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P31 to P24).	
CGMSB_5[7:0]			R/W
0xE463	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P39 to P32).	
CGMSB_6[7:0]			R/W
0xE464	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P47 to P40).	
CGMSB_7[7:0]			R/W
0xE465	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P55 to P48).	
CGMSB_8[7:0]			R/W
0xE466	<u>00000000</u>	This register is used to set the ED/HD CGMS Type B data bits (P63 to P56).	

Reg	Bits	Description	
CGMSB_9[7:0]			R/W
0xE467	00000000	This register is used to set the ED/HD CGMS Type B data bits (P71 to P64).	
CGMSB_10[7:0]			R/W
0xE468	00000000	This register is used to set the ED/HD CGMS Type B data bits (P79 to P72).	
CGMSB_11[7:0]			R/W
0xE469	00000000	This register is used to set the ED/HD CGMS Type B data bits (P87 to P80).	
CGMSB_12[7:0]			R/W
0xE46A	00000000	This register is used to set the ED/HD CGMS Type B data bits (P95 to P88).	
CGMSB_13[7:0]			R/W
0xE46B	00000000	This register is used to set the ED/HD CGMS Type B data bits (P103 to P96).	
CGMSB_14[7:0]			R/W
0xE46C	00000000	This register is used to set the ED/HD CGMS Type B data bits (P111 to P104).	
CGMSB_15[7:0]			R/W
0xE46D	00000000	This register is used to set the ED/HD CGMS Type B data bits (P119 to P112).	
CGMSB_16[7:0]			R/W
0xE46E	00000000	This register is used to set the ED/HD CGMS Type B data bits (P127 to P120).	
CHROMA_FILTER_SEL[2:0]			R/W
0xE480	00010000	This signal is used to configure the chroma filters for SD data. 000 - 1.3MHz 001 - 0.65MHz 010 - 1MHz 011 - 2MHz 100 - Reserved 101 - Chroma CIF 110 - Chroma QCIF 111 - 3MHz	
LUMA_FILTER_SEL[2:0]			R/W
0xE480	00010000	This signal is used to configure the luma filters for SD data. 000 - LPF NTSC 001 - LPF PAL 010 - Notch NTSC 011 - Notch PAL 100 - SSAF Luma 101 - Luma CIF 110 - Luma QCIF 111 - Reserved	
SD_ENC_IP_MODE[1:0]			R/W
0xE480	00010000	This signal is used to select the SD standard. 00 - NTSC 01 - PAL B/D/G/H/I 10 - PAL M 11 - PAL N	
DNR_EN			R/W
0xE481	00000001	This bit is used to enable the SD Digital Noise Reduction (DNR) function. 1 - Enabled 0 - Disabled	
SLOPE_EN			R/W
0xE482	00001001	This bit is used to enable the SD active video edge control. 1 - Enabled 0 - Disabled	

Reg	Bits	Description	
PIXEL_DATA_VALID			R/W
0xE482	00001001	This bit is used to enable SD pixel data valid. 1 - Enabled 0 - Disabled	
DVD_R			R/W
0xE482	00001001	This bit is used to enable the SD VCR FF/RW sync feature. 1 - Enabled 0 - Disabled	
PEDESTAL			R/W
0xE482	00001001	This bit is used to enable the SD pedestal. 1 - Enabled 0 - Disabled	
WIDE_UV_FILT			R/W
0xE482	00001001	This bit is used to enable the SSAF filter for PrPb SD data. 1 - Enabled 0 - Disabled	
CLOSE_CAP_EVEN			R/W
0xE483	00000100	This bit is used to enable SD closed captioning on even fields. 1 - Enabled 0 - Disabled	
CLOSE_CAP_ODD			R/W
0xE483	00000100	This bit is used to enable SD closed captioning on odd fields. 1 - Enabled 0 - Disabled	
VBI_DATA_EN			R/W
0xE483	00000100	This bit is used to enable data on the Vertical Blanking Interval (VBI) to be accepted as valid data. This is valid for SD video data only. 1 - Enabled 0 - Disabled	
UV_CTRL[1:0]			R/W
0xE483	00000100	This signal is used to configure the SD output levels for PrPb. 00 - 700 mV p-p (PAL), 1000 mV p-p (NTSC) 01 - 700 mV p-p 10 - 1000 mV p-p 11 - 648 mV p-p	
BETACAM			R/W
0xE483	00000100	This bit is used to configure the SD output levels for Y. 0 - 700mV/300mV 1 - 714mV/286mV	
PED_EN_YUV			R/W
0xE483	00000100	This bit is used to configure the SD pedestal on YPrPb output. 0 - No pedestal on YPrPb 1 - 7.5 IRE pedestal on YPrPb	
COLOR_BAR_CONTROL			R/W
0xE484	00000000	This bit is used to enable the SD color bars test pattern. 1 - Disabled 0 - Enabled	
BURST_CONTROL			R/W
0xE484	00000000	This bit is used to enable the SD burst test pattern. 1 - Disabled 0 - Enabled	
CHROMA_CONTROL			R/W
0xE484	00000000	This bit is used to control SD chroma. 0 - Chroma enabled. 1 - Chroma disabled.	

Reg	Bits	Description	
CCIR601_CCIR624			R/W
0xE484	0000<u>0</u>000	This bit is used to adjust the SD active video length. 0 - 720 pixels 1 - 710 (NTSC), 702 (PAL)	
RTCEN[1:0]			R/W
0xE484	000000<u>00</u>	This signal is used to select the Sub-carrier Frequency Lock mode. The value of these register bits along with the status of the SFL pin determine the operation. 00 - Disabled. 11 - SFL mode enabled.	
NO_WRAPAROUND_SD			R/W
0xE486	0<u>0</u>000010	This bit is used to select the SD horizontal/vertical counter mode. 0 - Update field/line counter. 1 - Field/line counter free running	
EIA_CEA_MODE			R/W
0xE486	0000<u>0</u>010	This bit is used to enable SD EIA/CEA-861 synchronization compliance. 1 - Enabled 0 - Disabled	
NTSC_C_BRST_ADJ[1:0]			R/W
0xE486	000000<u>10</u>	This signal is used to adjust the NTSC color subcarrier position. It sets the delay from the falling edge of the output HSYNC pulse to the start of color burst. 00 - 5.17 µs. 01 - 5.31 µs. 10 - 5.59 µs (must be set for Macrovision compliance). 11 - Reserved.	
SD_RGB_IP_EN			R/W
0xE487	<u>0</u>0000000	This bit is used to enable the SD CSC for RGB inputs. 0 - SD RGB processing disabled 1 - SD RGB processing enabled	
SD_AUTODETECT_EN			R/W
0xE487	00<u>0</u>00000	This bit is used to enable the encoder section to auto-detect the input standard. 1 - Enabled 0 - Disabled	
PEAK_EN			R/W
0xE487	0000<u>0</u>000	This bit is used to enable the SD SSAF filter gain. 1 - Enabled 0 - Disabled	
SETUP_EN			R/W
0xE487	0000<u>0</u>000	This bit is used to enable the SD brightness control feature. 1 - Enabled 0 - Disabled	
HUE_EN			R/W
0xE487	00000<u>0</u>00	This bit is used to enable the hue adjust function. 1 - Enabled 0 - Disabled	
SATURATE_LUMA			R/W
0xE487	000000<u>0</u>0	This bit is used to enable the SD luma scale saturation. 1 - Enabled 0 - Disabled	
SCALE_YCBCR_EN			R/W
0xE487	0000000<u>0</u>	This bit is used to enable the SD luma and colour scale control feature. 1 - Enabled 0 - Disabled	
GAMMA_CURVE_B			R/W
0xE488	<u>0</u>0000000	This bit is used to select the gamma correction curves for SD video data. 0 - Gamma Correction Curve A 1 - Gamma Correction Curve B	

Reg	Bits	Description	
GAMMA_EN			R/W
0xE488	0<u>0</u>000000	This bit is used to enable the gamma correction curves for SD video data. 1 - Enabled 0 - Disabled	
DB_EN			R/W
0xE488	000000<u>0</u>0	This bit is used to enable double buffering on the appropriate SD registers. 1 - Enabled 0 - Disabled	
SD_NON_INTERLACED			R/W
0xE488	00000<u>0</u>00	This bit is used to enable the support of SD non-interlaced modes. 1 - Enabled 0 - Disabled	
CHROMA_DEL[1:0]			R/W
0xE489	00<u>00</u>0000	This signal is used to configure the SD chroma delay. 00 - No Delay 01 - Four clock cycles 10 - Eight clock cycles 11 - Reserved	
BLK_BURST_LUMA			R/W
0xE489	0000<u>0</u>000	This bit is used to enable the SD black burst output on DAC luma. 1 - Enabled 0 - Disabled	
SD_UNDER_LIMITER[1:0]			R/W
0xE489	000000<u>00</u>	This signal is used to configure the SD undershoot limiter. 00 - Disabled 01 - -11IRE 10 - -6IRE 11 - -1.5IRE	
SD_Y_MIN_VALUE			R/W
0xE48A	0<u>0</u>001000	This bit is used to configure the SD minimum luma value. 0 - -40IRE 1 - -7.5IRE	
YDEL[1:0]			R/W
0xE48A	00<u>00</u>1000	This signal is used to configure the SD luma delay. 00 - No delay 01 - Two clock cycles 10 - Four clock cycles 11 - Six clock cycles	
FSC[31:0]			R/W
0xE48C 0xE48D 0xE48E 0xE48F	<u>00011111</u> <u>01111100</u> <u>11110000</u> <u>00100001</u>	This register is used to set the subcarrier frequency value.	
SUB_CARRIER_PHASE[9:0]			R/W
0xE490 0xE49C	<u>00000000</u> <u>00</u>000000	This register is used to configure the SD subcarrier phase.	
VBI_EXT[15:0]			R/W
0xE491 0xE492	<u>00000000</u> <u>00000000</u>	This register is used to configure the extended data on even fields.	

Reg	Bits	Description	
VBI_CAP[15:0]			R/W
0xE493 0xE494	00000000 00000000	This register is used to configure the extended data on odd fields.	
NTSC_PED0[7:0]			R/W
0xE495	00000000	This register is used to enable pedestal on line number 17 of the odd field. 1 - Enabled 0 - Disabled	
NTSC_PED1[7:0]			R/W
0xE496	00000000	This register is used to enable pedestal on line number 25 of the odd field. 1 - Enabled 0 - Disabled	
NTSC_PED2[7:0]			R/W
0xE497	00000000	This register is used to enable pedestal on line number 17 of the even field. 1 - Enabled 0 - Disabled	
NTSC_PED3[7:0]			R/W
0xE498	00000000	This register is used to enable pedestal on line number 25 of the even field. 1 - Enabled 0 - Disabled	
WSS_EN			R/W
0xE499	00000000	This bit is used to enable SD WSS. 1 - Enabled 0 - Disabled	
CGMS_EVEN			R/W
0xE499	00000000	This bit is used to enable SD CGMS on the even field. 1 - Enabled 0 - Disabled	
CGMS_ODD			R/W
0xE499	00000000	This bit is used to enable SD CGMS on the odd field. 1 - Enabled 0 - Disabled	
CGMS_POLY			R/W
0xE499	00000000	This bit is used to enable SD CGMS cyclic redundancy check (CRC). 1 - Enabled 0 - Disabled	
CGMS_WSS[19:0]			R/W
0xE499 0xE49A 0xE49B	00000000 00000000 00000000	This signal is used to specify the SD CGMS [20-bits]/WSS data[14-bits]. 0x00000 - CGMS Data Bits or WSS Data Bits	
CR_SCALE[9:0]			R/W
0xE49C 0xE49F	00000000 00000000	This signal is used to set the SD Cr scale value.	
CB_SCALE[9:0]			R/W
0xE49C 0xE49E	00000000 00000000	This signal is used to set the SD Cb scale value.	
CONTRAST[9:0]			R/W
0xE49C 0xE49D	00000000 00000000	This signal is used to set the SD Y scale value.	

Reg	Bits	Description
HUE[7:0]		R/W
0xE4A0	00000000	This register is used to set the SD hue adjust value. 0x00 - SD Hue Value
BLANK23		R/W
0xE4A1	00000000	This bit is used to enable the SD blank WSS data. 1 - Enabled 0 - Disabled
SETUP[6:0]		R/W
0xE4A1	00000000	This signal is used to specify the SD brightness value.
PEAK[3:0]		R/W
0xE4A2	00000000	This signal is used to configure the SD luma SSAF gain/attenuation (only applicable if subaddress 0x87, Bit 4 = 1). 0000 - -4dB - ... 0100 - 0dB - ... 1000 - +4dB
DNR_CORING_GAIN_A[3:0]		R/W
0xE4A3	00000000	This signal is used to configure the coring gain border (in Digital Noise Reduction (DNR) mode, the values in brackets apply). 0000 - No gain 0001 - +1/16 [–1/8] 0010 - +2/16 [–2/8] 0011 - +3/16 [–3/8] 0100 - +4/16 [–4/8] 0101 - +5/16 [–5/8] 0110 - +6/16 [–6/8] 0111 - +7/16 [–7/8] 1000 - +8/16 [–1]
DNR_CORING_GAIN_B[3:0]		R/W
0xE4A3	00000000	This signal is used to configure the coring gain data (in Digital Noise Reduction (DNR) mode, the values in brackets apply). 0000 - No gain 0001 - +1/16 [–1/8] 0010 - +2/16 [–2/8] 0011 - +3/16 [–3/8] 0100 - +4/16 [–4/8] 0101 - +5/16 [–5/8] 0110 - +6/16 [–6/8] 0111 - +7/16 [–7/8] 1000 - +8/16 [–1]
DNR_MPEG_1		R/W
0xE4A4	00000000	This bit is used to select the Digital Noise Reduction (DNR) block size. 1 - 16 pixels 0 - 8 pixels
BLK_BORDER_2		R/W
0xE4A4	00000000	This bit is used to select the Digital Noise Reduction (DNR) border area. 0 - 2 pixels 1 - 4 pixels
DNR_THRESHOLD[5:0]		R/W
0xE4A4	00000000	This signal is used to configure the Digital Noise Reduction (DNR) threshold. 000000 - 0 000001 - 1 - ... 111110 - 62 111111 - 63

Reg	Bits	Description	
BLK_OFFSET[3:0]			R/W
0xE4A5	0000 0000	This signal is used to configure the Digital Noise Reduction (DNR) block offset. 0000 - 0 pixel offset 0001 - One pixel offset - ... 1110 - 14 pixel offset 1111 - 15 pixel offset	
DNR_ENABLE_SHARPNESS			R/W
0xE4A5	0000 0000	This bit is used to select the Digital Noise Reduction (DNR) mode. 0 - DNR mode 1 - DNR sharpness mode	
DNR_FMODE_CONTROL[2:0]			R/W
0xE4A5	0000 0000	This signal is used to configure the Digital Noise Reduction (DNR) input filter. 001 - Filter A 010 - Filter B 011 - Filter C 100 - Filter D	
GAMMA_A_0[7:0]			R/W
0xE4A6	00000000	This register is used to configure the SD gamma curve A (A0 - Point 24).	
GAMMA_A_1[7:0]			R/W
0xE4A7	00000000	This register is used to configure the SD gamma curve A (A1 - Point 32).	
GAMMA_A_2[7:0]			R/W
0xE4A8	00000000	This register is used to configure the SD gamma curve A (A2 - Point 48).	
GAMMA_A_3[7:0]			R/W
0xE4A9	00000000	This register is used to configure the SD gamma curve A (A3 - Point 64).	
GAMMA_A_4[7:0]			R/W
0xE4AA	00000000	This register is used to configure the SD gamma curve A (A4 - Point 80).	
GAMMA_A_5[7:0]			R/W
0xE4AB	00000000	This register is used to configure the SD gamma curve A (A5 - Point 96).	
GAMMA_A_6[7:0]			R/W
0xE4AC	00000000	This register is used to configure the SD gamma curve A (A6 - Point 128).	
GAMMA_A_7[7:0]			R/W
0xE4AD	00000000	This register is used to configure the SD gamma curve A (A7 - Point 160).	
GAMMA_A_8[7:0]			R/W
0xE4AE	00000000	This register is used to configure the SD gamma curve A (A8 - Point 192).	
GAMMA_A_9[7:0]			R/W
0xE4AF	00000000	This register is used to configure the SD gamma curve A (A9 - Point 224).	
GAMMA_B_0[7:0]			R/W
0xE4B0	00000000	This register is used to configure the SD gamma curve B (B0 - Point 24).	
GAMMA_B_1[7:0]			R/W
0xE4B1	00000000	This register is used to configure the SD gamma curve B (B1 - Point 32).	
GAMMA_B_2[7:0]			R/W
0xE4B2	00000000	This register is used to configure the SD gamma curve B (B2 - Point 48).	

Reg	Bits	Description	
GAMMA_B_3[7:0]			R/W
0xE4B3	00000000	This register is used to configure the SD gamma curve B (B3 - Point 64).	
GAMMA_B_4[7:0]			R/W
0xE4B4	00000000	This register is used to configure the SD gamma curve B (B4 - Point 80).	
GAMMA_B_5[7:0]			R/W
0xE4B5	00000000	This register is used to configure the SD gamma curve B (B5 - Point 96).	
GAMMA_B_6[7:0]			R/W
0xE4B6	00000000	This register is used to configure the SD gamma curve B (B6 - Point 128).	
GAMMA_B_7[7:0]			R/W
0xE4B7	00000000	This register is used to configure the SD gamma curve B (B7 - Point 160).	
GAMMA_B_8[7:0]			R/W
0xE4B8	00000000	This register is used to configure the SD gamma curve B (B8 - Point 192).	
GAMMA_B_9[7:0]			R/W
0xE4B9	00000000	This register is used to configure the SD gamma curve B (B9 - Point 224).	
BRIGHT_DETECT_VAL[7:0]			R
0xE4BA	00000000	This register is used to adjust the SD brightness value. 0xXX - (Larger settings results in a brighter output)	
FCOUNT[2:0]			R
0xE4BB	01000000	This signal is used to readback the SD field count. The field count register can be used to identify the number of the active field.	

3 INDEX

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

ESD Caution

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

UG12393-0-6/14(0)

www.analog.com