

Getting Started with the ADV202 Programming Guide

OVERVIEW 4

GENERAL ENCODE/DECODE ROUTINE	5
ENCODE/DECODE ROUTINE FOR NORMAL HOST MODE- 32BIT	9
ENCODE/DECODE ROUTINE FOR NORMAL HOST MODE-16-BIT	14
ENCODE/DECODE ROUTINE FOR DEDICATED CHIP SELECT - 32BIT	17
ENCODE/DECODE ROUTINE FOR DMA DREQ/DACK MODES - 32BIT	21
ENCODE/DECODE ROUTINE FOR JDATA MODE – 16-BIT HOST	27
CUSTOM SPECIFIC MODE ENCODE AND DECODE ROUTINES	31
CONFIGURATION EXAMPLE FOR CUSTOM SPECIFIC MODE:.....	34
RAW PIXEL MODE ENCODE AND DECODE ROUTINES.....	38
CONFIGURATION EXAMPLE FOR RAW PIXEL MODE:.....	38

APPENDIX 42

PLL SETTINGS AND INTERNAL CLOCKS.....	42
BUSMODE AND MMODE REGISTER SETTINGS.....	42
JPEG2000 PARAMETERS - ENCODE.....	43
JPEG2000 PARAMETERS - DECODE.....	49
JPEG2000 PARAMETER DESCRIPTION.....	52

VFORMAT 52
CBSIZE 52
ATTRTYPE 53
J2KPROG 56

QFACT, STEPSIZES AND LOAD_SS:	57
COD_STYLE	57
VISUAL WEIGHTING - LOAD_VW, VW_Y, VW_CB, VW_CR:	58
MULTI-LAYER LTARGET [1:16]	59
CODE EXAMPLE	60

Overview

This document applies to the ADV202 engineering samples that are branded as follows:

ADV202xx
SURF®

xxxxx 0.X or RevE
where X is equal or larger than 2.

The following document will provide a series of recommended procedures and necessary information to establish functionality of an ADV202 based target design.

It is highly recommended to consult

1. **Application Note AN790 – How to use the ADV202**
This application note gives an overview over the different interface modes mentioned in this application note.
2. **AN799 ADV202 Testmodes** contains instructions to verify correct hardware configuration and functional register access.
3. **ADV202 User Guide** contains detailed explanation of register settings mentioned in this application note.

This document contains:

1. Encode routine
2. Decode routine

For interface modes:

1. Normal Host Mode using the Threshold registers
2. Dedicated Chip Select mode
3. DMA DREQ/DACK modes
4. JDATA mode
5. Custom Specific mode
6. Raw Video mode

General encode/decode routine

ADV202 initialisation routine must follow a specific order of instructions.
The flow diagram in figure 2 and figure 3, show the essential procedures required.

ADV202 initialisation starts with a direct register access Normal Host Mode operation [datasheet, figure 4] to the PLL registers, BOOT, MMODE and BUSMODE registers.
For recommended PLL register settings refer to table 6 in the Appendix or for more detail on the PLL, refer to the *ADV202 datasheet, page 30*.

The state of the input pins must be held by the target system until the ADV202 asserts the ACK/ signal [input pins: ADDR, CS/, WR/, RD/ DACK/ and HDATA].

A write request [WR/ and CS/ asserted] or a read request [RD/ and CS/] is followed by ACK/ assertion. The HDATA should not be sampled until after ACK/ is asserted.

This does only apply to Normal Host Mode accesses and is not required for DMA modes DMA DREQ/DACK, JDATA mode, Dedicated Chip Select Mode.

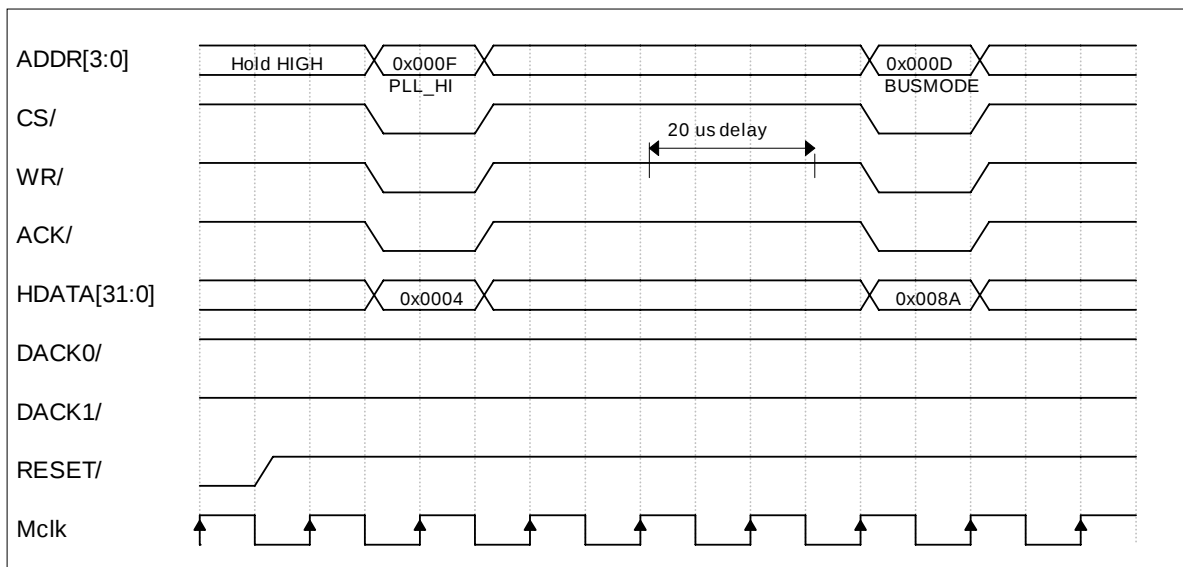


Figure 1 Direct Register Access after Power-Up - Normal Host Mode-
Writing to PLL register and BUSMODE register

The same protocol is used for indirect register accesses to load firmware, set encode parameters and if required by the application, configure DMA registers and Dimension Registers. It is important to note, that if the DMA registers require to be written to, they must stay disabled while doing so. They are enabled at a later stage in the configuration process.

Correct firmware load can be confirmed with an interrupt or poll routine of the SWFLAG register. The SWFLAG register must read FF82h for correct encode and FFA2h for correct decode firmware load.

After the firmware and the JPEG2000 parameters are loaded into ADV202 memory and application specific registers are configured, the program can be started with another direct register access to clear the SWIRQ0 interrupt flag in the EIRQFLG register. This flag is set by the ADV202 when the part is ready to start the program.

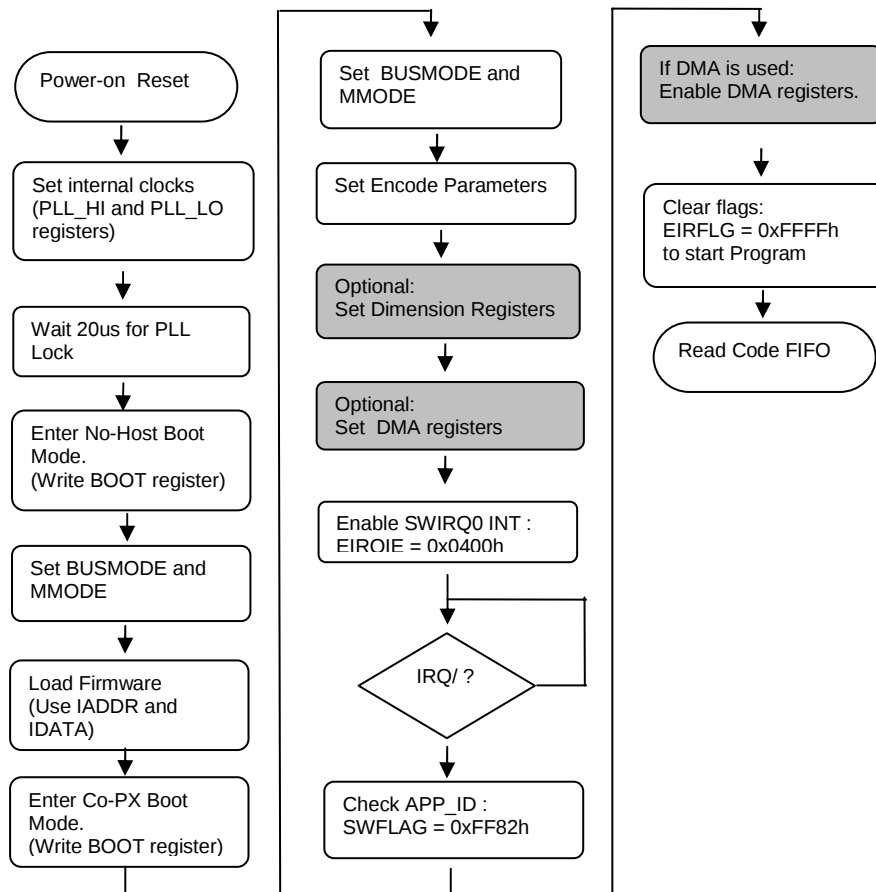


Figure 2 General Encode routine

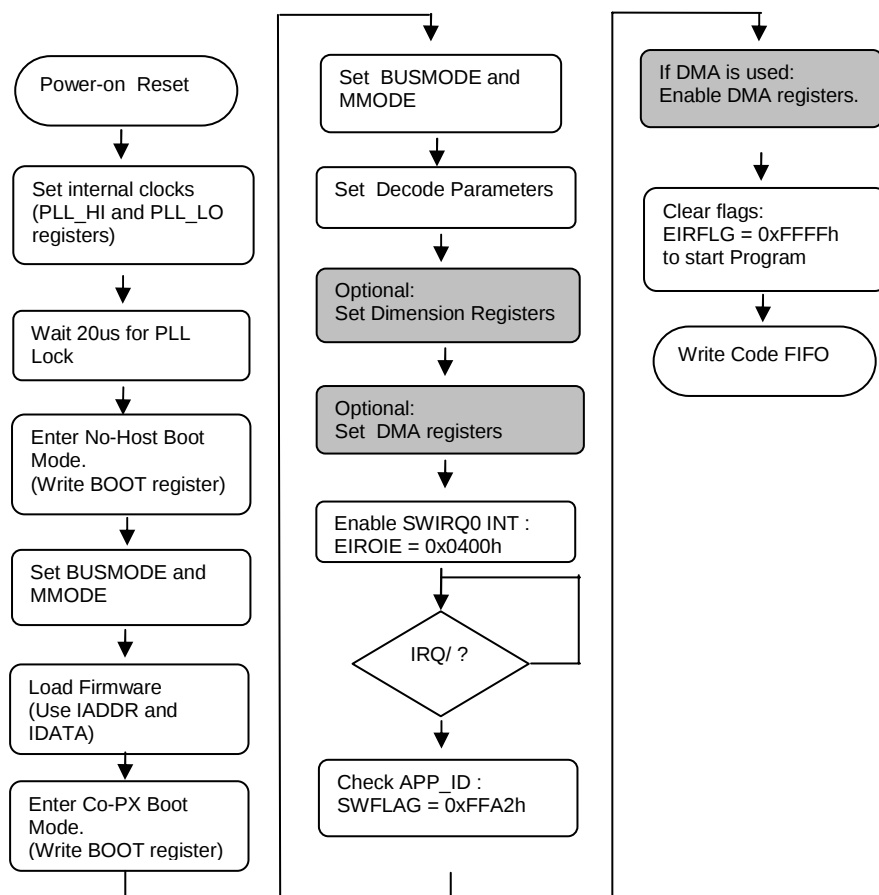


Figure 3 General Decode routine

Encode/Decode routine for Normal Host Mode- 32bit

Normal Host Mode accesses are not only used for direct register access, firmware and parameter load but can also be used for the transfer of compressed data.

The compressed data transfers are controlled with the threshold register FFTHRC and enabling a threshold specific interrupt [EIRQIE register = 0x0002h] that is asserted on the IRQ pin once the threshold conditions are met.

FFMODE register is used to set the direction of the FIFO data transfer and FIFO size. This register does not have to be written to. The firmware controls this register.

Data transfers should be initiated when the IRQ/ is asserted and EIRQFLG reads 0x0002h after SWIRQ0 has been asserted and cleared. This indicates that the CODE FIFO contains data and that the threshold conditions for the CODE FIFO have been met.

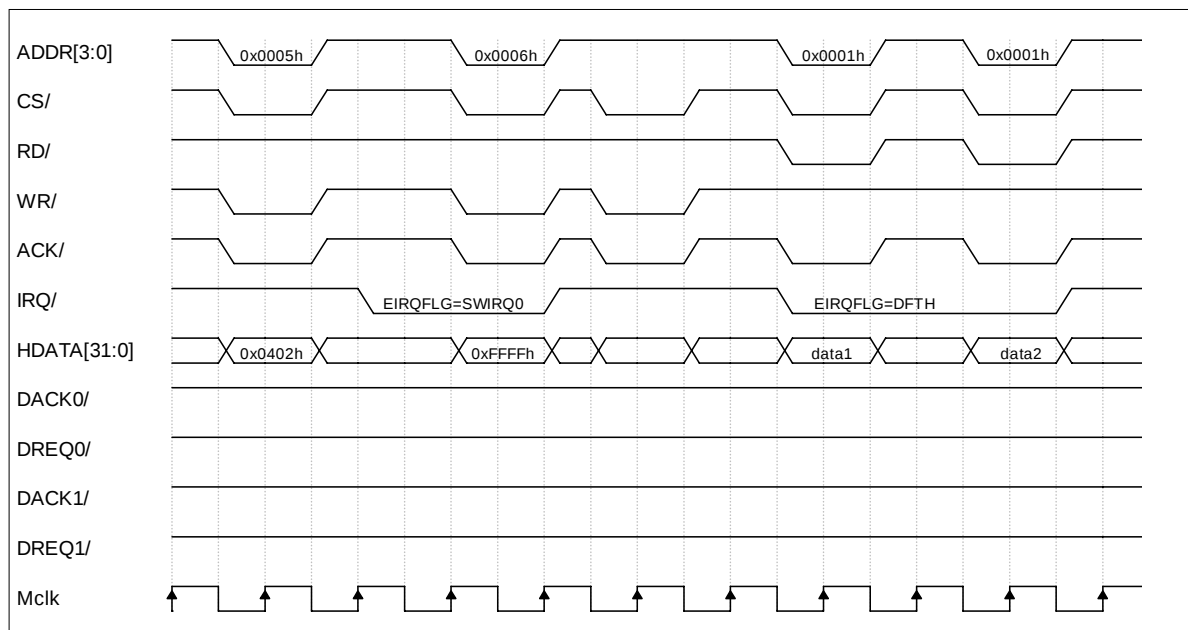


Figure 4 – Encode mode: Start program and read compressed data out of CODE FIFO

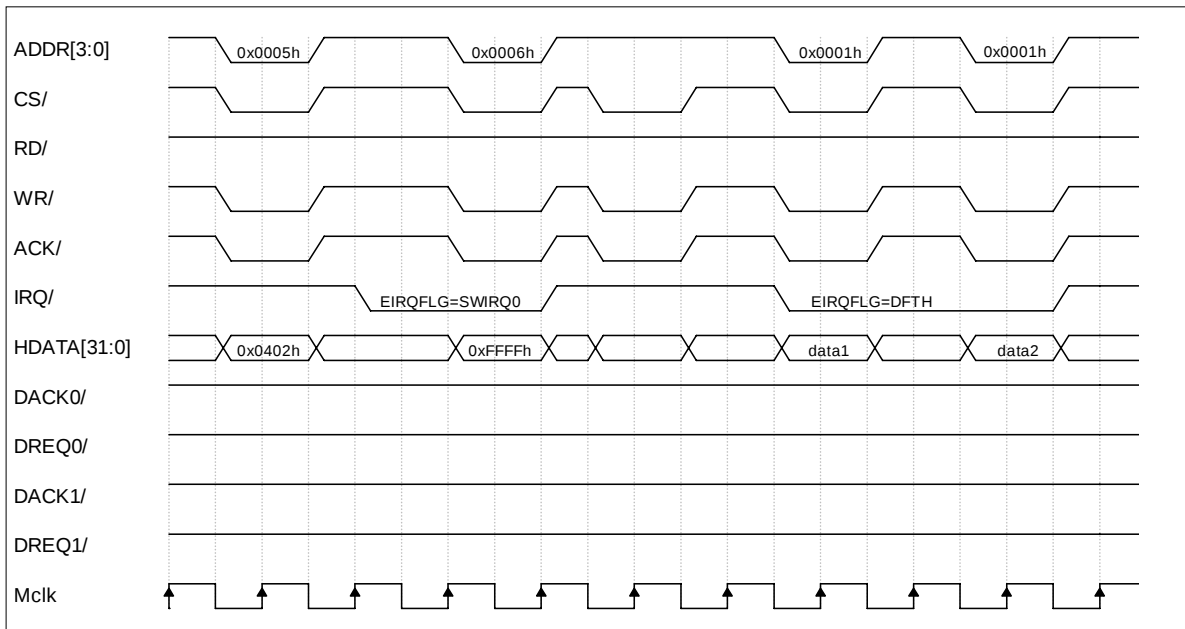


Figure 5 – Decode mode: Start program and write compressed data into CODE FIFO

In encode mode LCODE [SCOMM 4 output pin] can be used to determine when the last word of a field is read from the FIFO.

In decode mode, LCODE will be asserted as long as the ADV202 receives active video data.

When the Code Threshold register [address FFFF141C, FFTHRC] is used for compressed data in/out then :

In decode mode the IRQ/ is asserted when the number of empty locations is greater or equal the threshold value.

In encode mode, IRQ/ is asserted whenever the number of data words in the FIFO is greater or equal the threshold value.

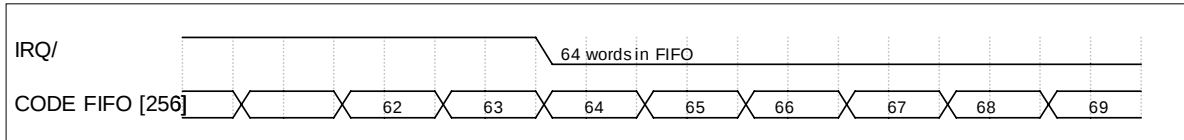


Figure 6 IRQ/ assert in encode mode if threshold is set to 64 words

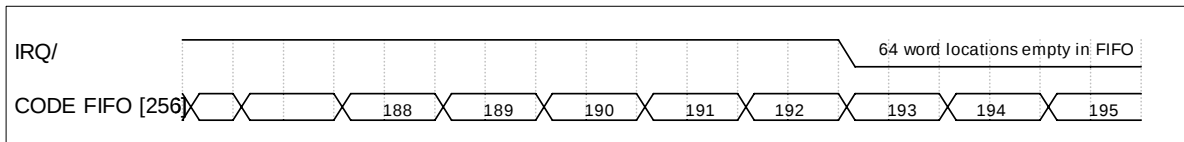


Figure 7 IRQ/ assert in decode mode if threshold is set to 64 words

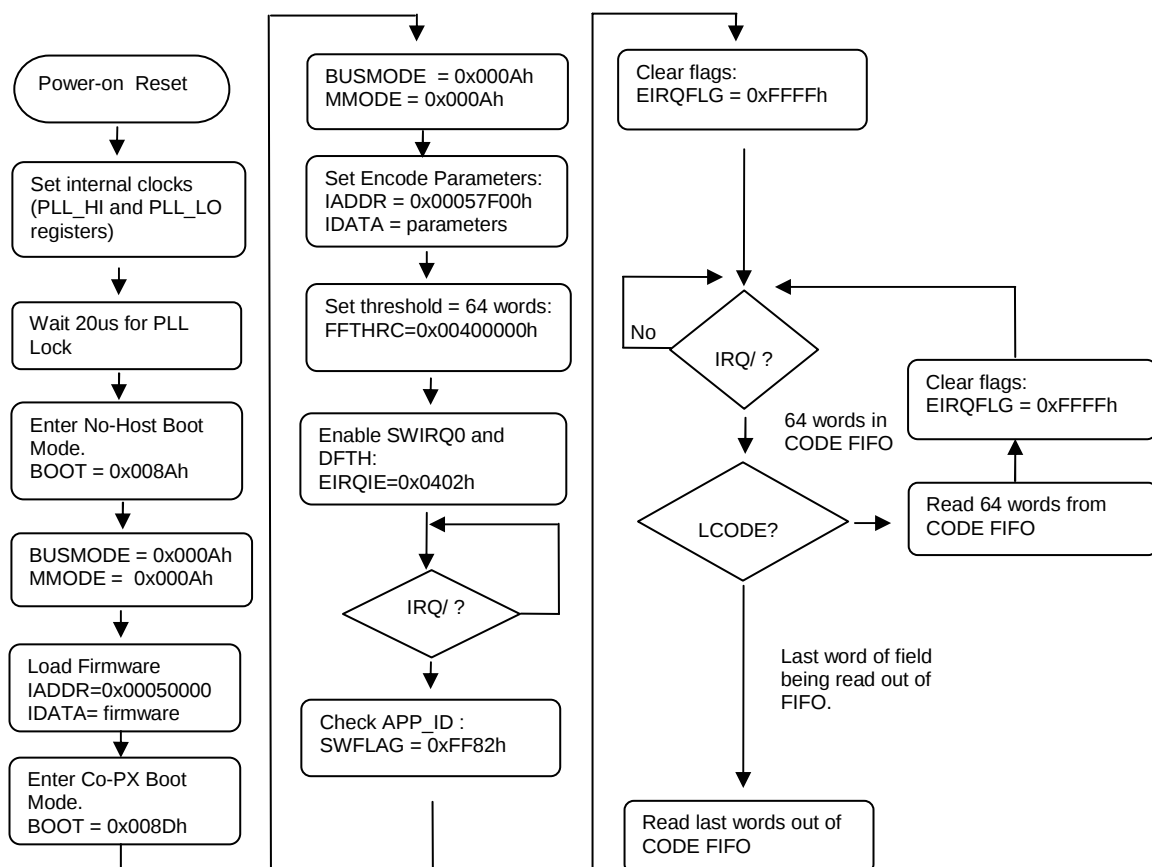


Figure 8 Encode routine for Normal Host Mode in 32-bit mode

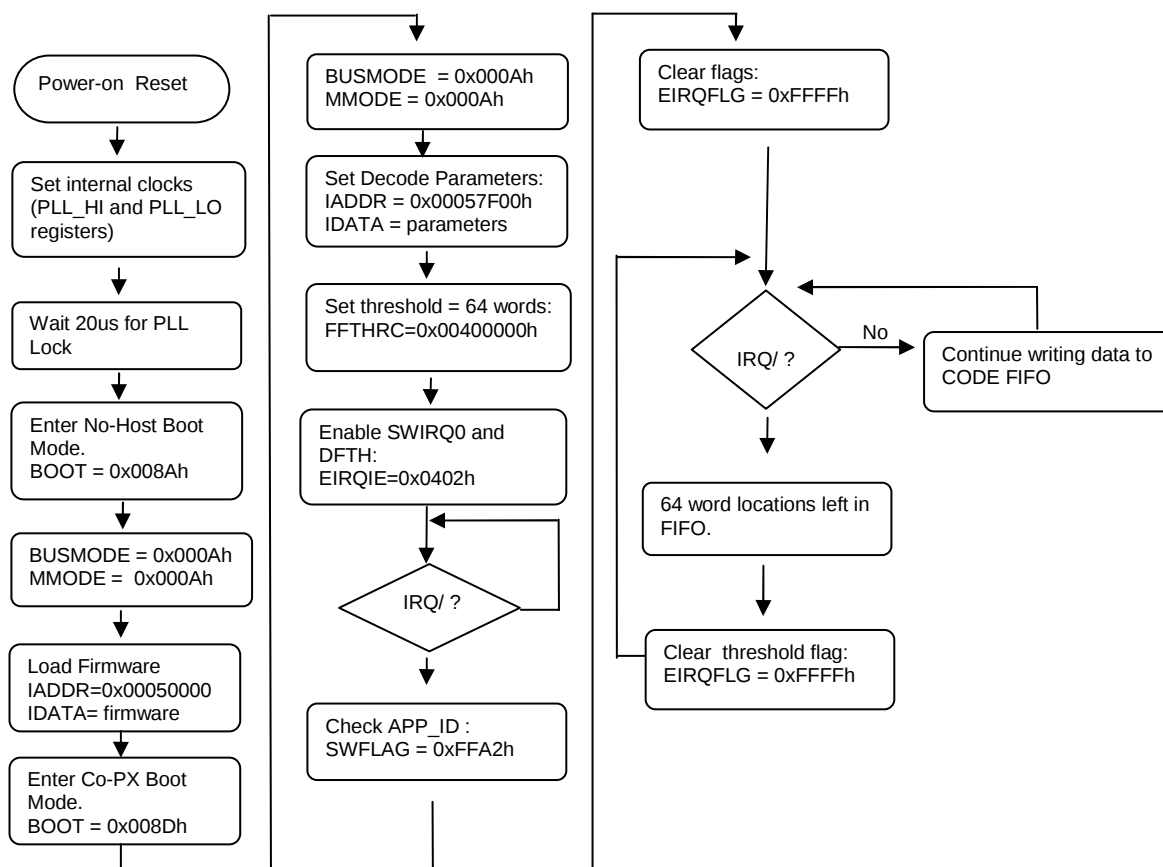


Figure 9 Decode routine for Normal Host Mode in 32-bit mode

Encode/Decode routine for Normal Host Mode-16-bit

16-bit Normal Host Mode follows the same procedures as 32-bit configurations.

For Normal Host Mode accesses in 16-bit mode the following has to be considered also:

1. The register settings for BUSMODE must reflect the 16-bit or 32-bit data accesses.

	Interface	BUSMODE	MMODE
Encode	16-bit	0005h	0005h
Encode	32-bit	000Ah	000Ah
Decode	16-bit	0005h	0005h
Decode	32-bit	000Ah	000Ah

Table 1 BUSMODE register settings for 16-bit or 32-bit accesses

2. The Stage register [address 0x0Ah] must be used.
The Stage register is used to access 32-bit words when using a 16-bit host control bus. Refer to the *ADV202 User Guide*, 4-15 for more detail.
3. In 16-bit mode, the data read-back value is halfword swapped.
The written order is Big-Endian, the read order is Little-Endian.
4. FIFO count is still in 32-bit words even if 16-bit mode is used. A threshold setting of 64 would therefore be met if 128 16-bit data words are present in the CODE FIFO [in encode mode] or 128 16-bit locations are empty in the CODE FIFO.
If the last 16-bit word does not end on a 32-bit boundary, it will be counted nevertheless.

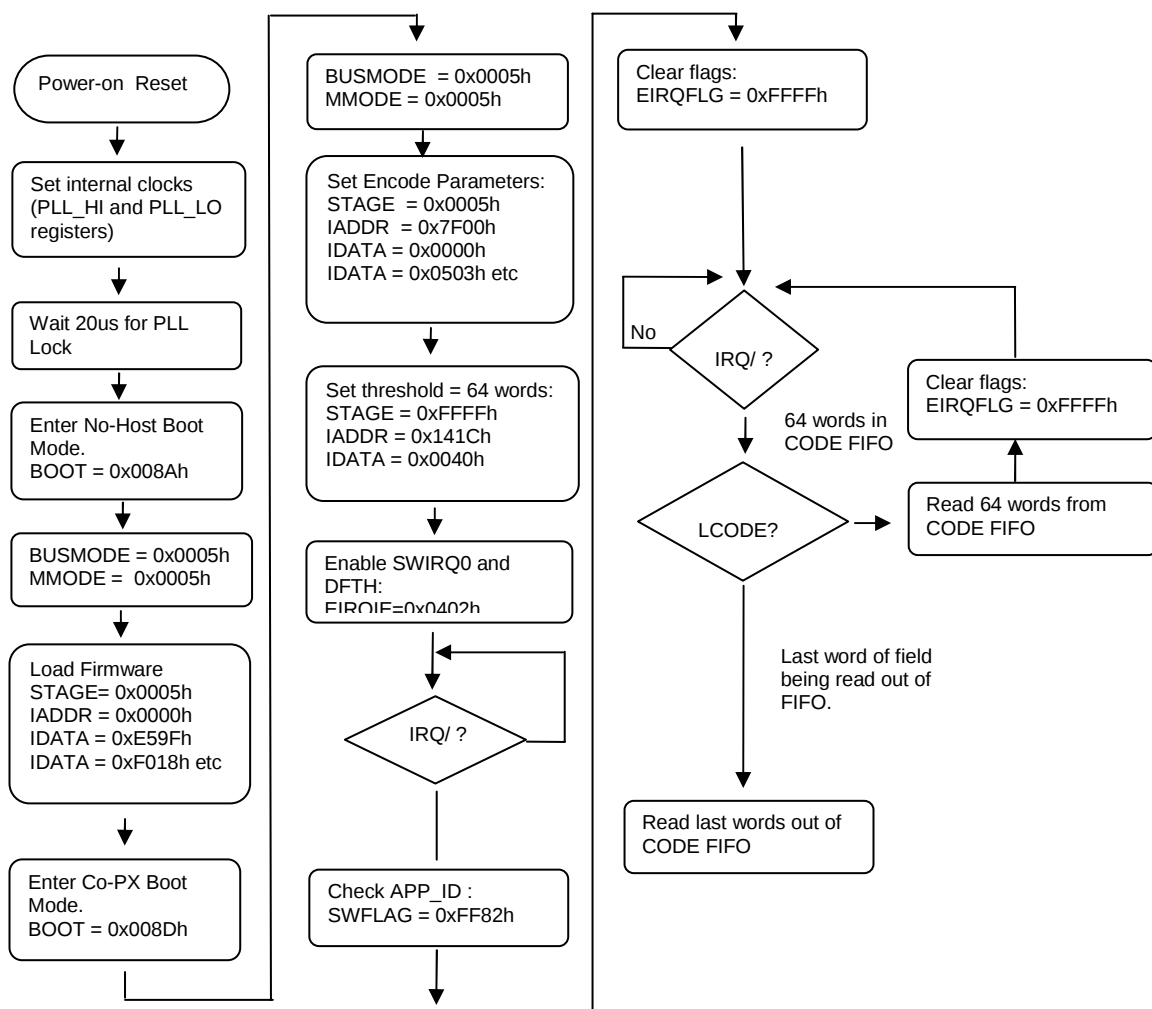


Figure 10 Encode routine for Normal Host Mode in 16-bit mode

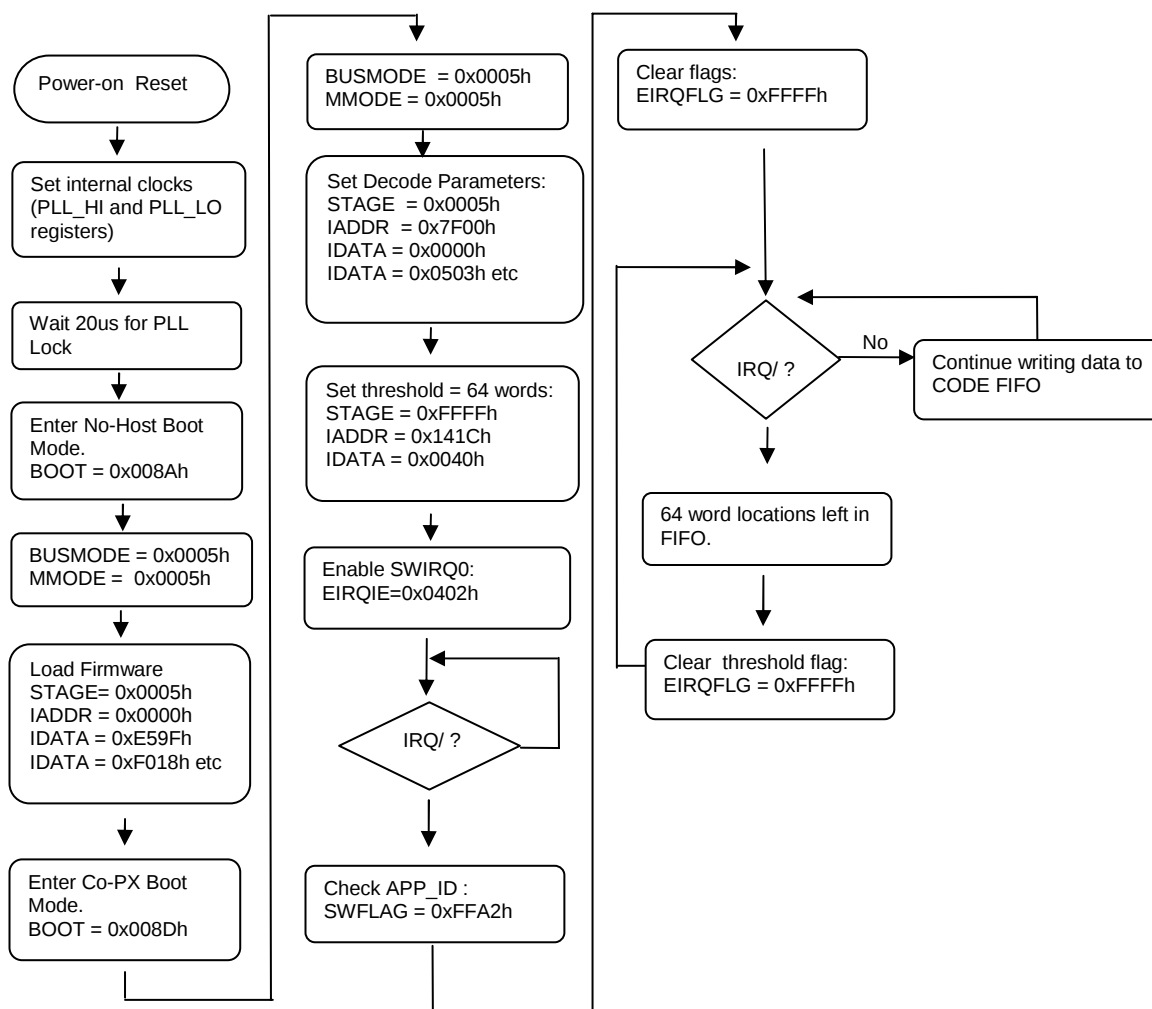


Figure 11 Decode routine for Normal Host Mode in 16-bit mode

Encode/Decode routine for Dedicated Chip Select - 32bit

In DCS mode direct register configuration, firmware and parameter load is the same as before using Normal Host Mode. This mode differs in the compressed data transfer interface since it requires the DACK/ and DREQ/ pins to be connected to the host but on the other hand threshold registers do not have to be configured.

The compressed data transfers are controlled internally by the ADV202 when DCS mode is enabled in the EDMOD registers.

In encode mode DREQ/ is asserted, as long as data is in the CODE FIFO.

In decode mode DREQ/ is asserted, as long as there is enough space in the CODE FIFO to take in data.

Threshold registers do not have to be configured but it requires to configure EDMOD registers. EDMOD registers must stay disabled while writing to these registers. They are enabled at a later stage in the configuration process.

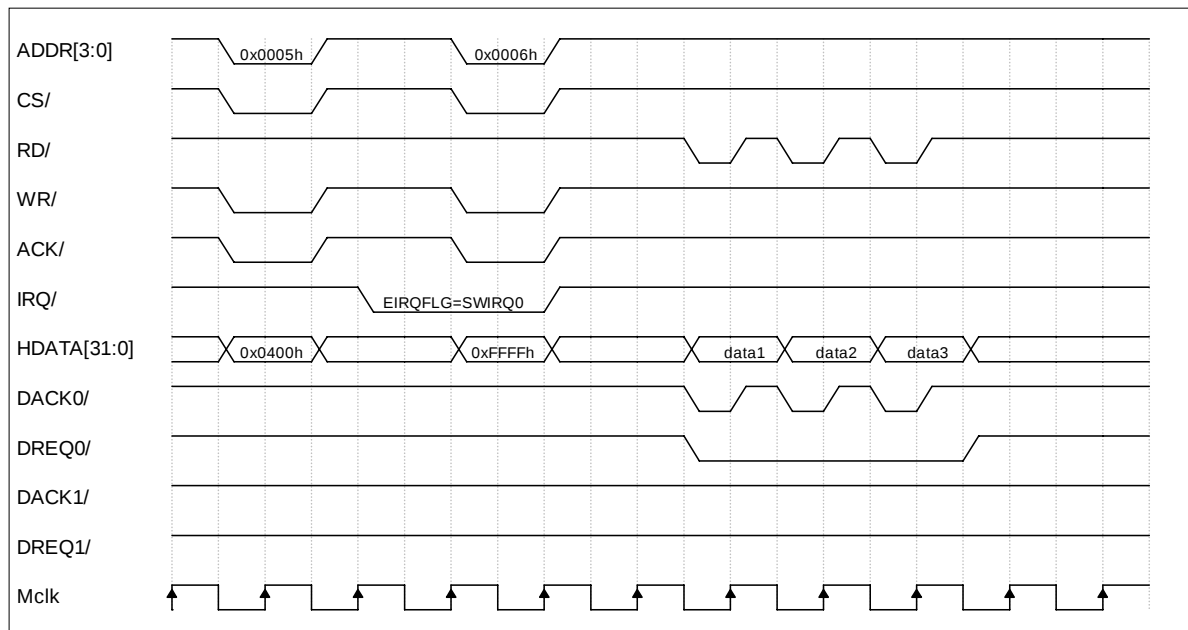


Figure 12 DCS Encode mode – Start program and read data from CODE FIFO

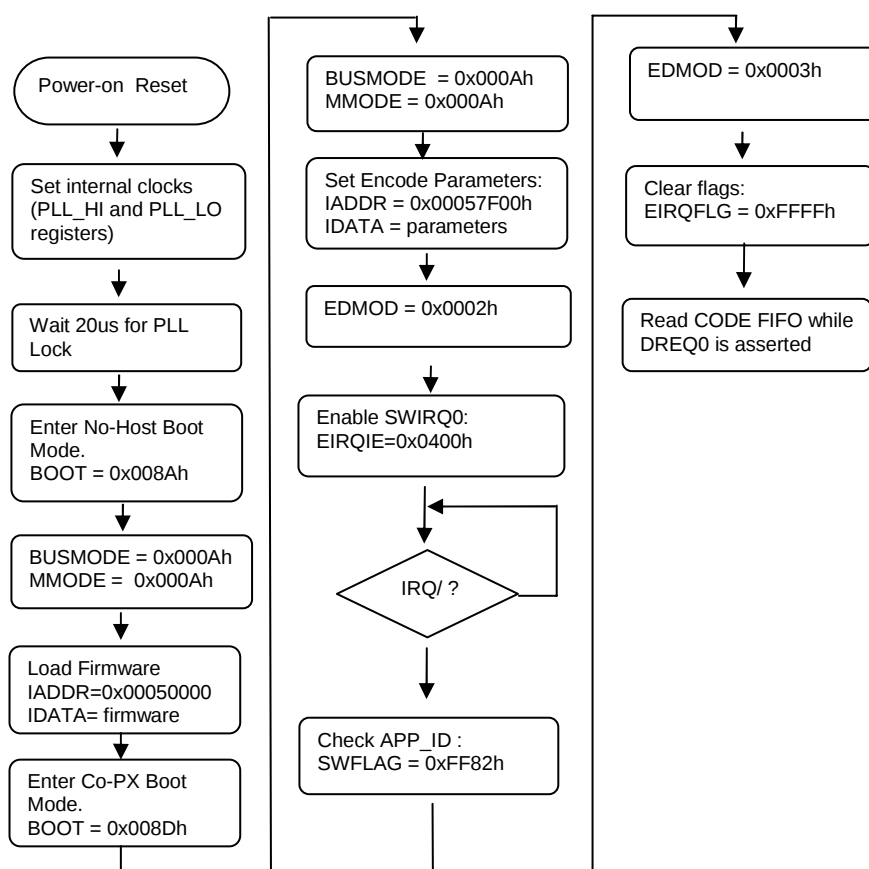


Figure 13 Encode Routine for DCS mode

Getting Started with the ADV202

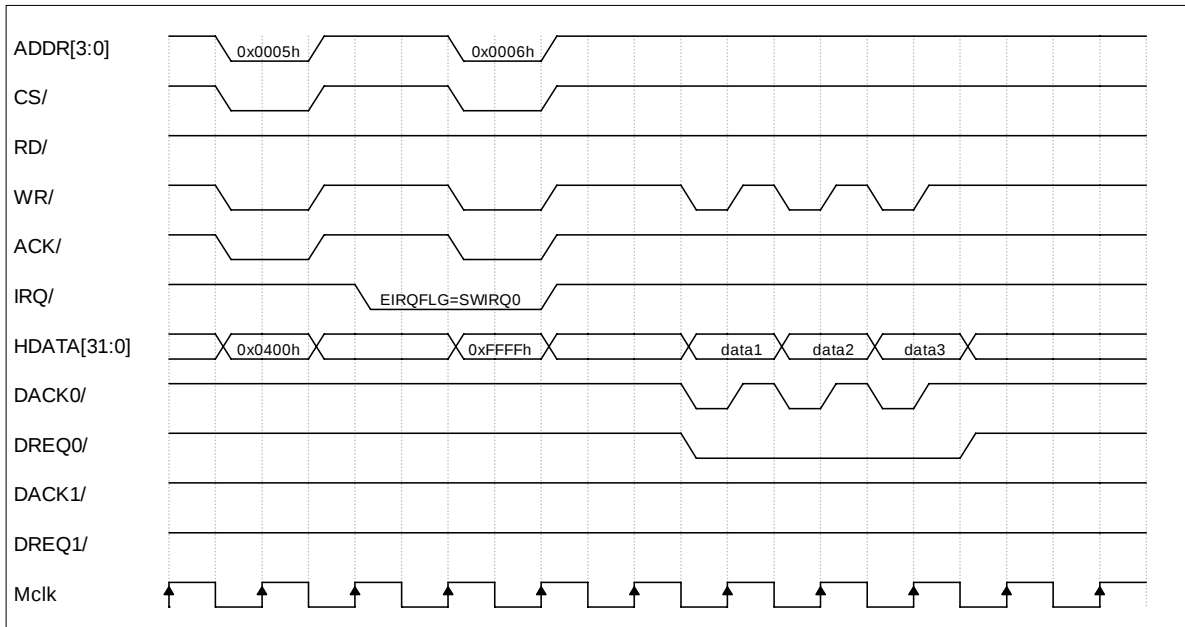


Figure 14 Decode DCS mode – Start program and write data to CODE FIFO

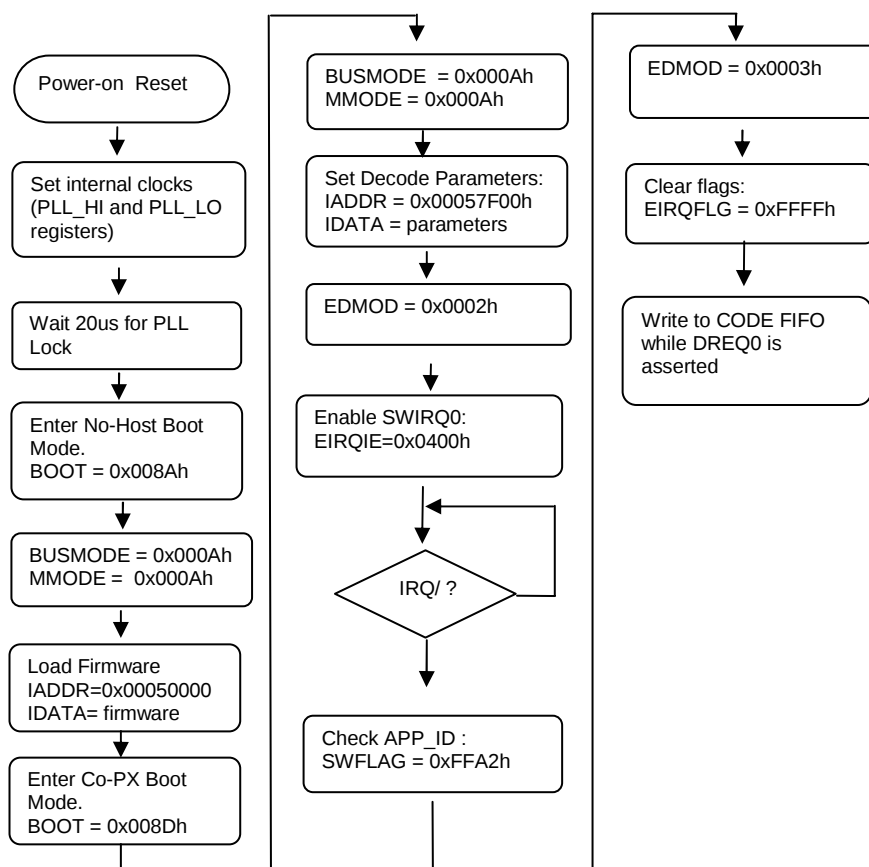


Figure 15 Decode Routine for DCS mode

Encode/Decode routine for DMA DREQ/DACK Modes - 32bit

DMA mode can be used in burst or single access. Direct register configuration, firmware and parameter load and indirect register configuration is the same as before using Normal Host Mode. Compressed data in/output is controlled by the ADV202's DMA interface.

This mode requires the DACK/ and DREQ/ pins to be connected to the host for DMA configuration.

The compressed data transfers are controlled internally by the ADV202.

DREQ/ is asserted, as soon as the ADV202 is ready to transmit/receive data and as long as sufficient data or space is available in the CODE FIFO to complete a burst or single DMA access.

Threshold registers do not have to be configured but it requires to configure EDMOD registers.

EDMOD registers must stay disabled while writing to them, they are enabled at a later stage in the configuration process. Generally only one DMA channel is used to either input compressed data in decode mode or to output compressed data in encode mode. Therefore only DMA channel 0 is used and only EDMOD0 has to be configured.

Only in HIPI mode both EDMOD registers are used and must be configured. In this mode, pixel data is in-/output on DMA channel 0, while compressed data is in-/output on DMA channel 1.

Refer to:

ftp://ftp.analog.com/pub/Digital_Imaging/ADV202_ApplicationNotes/ADV202_HIPImode_Still_Image_Application/

for more information on HIPI mode.

In single DREQ/DACK mode, the ADV202 will assert DREQ/ when there is at least one data transfer available in the assigned FIFO in encode mode. In decode, DREQ/ will be asserted when the ADV202 is ready to receive at least one data transfer to the assigned FIFO.

The same applies to DREQ/DACK burst mode, except that the ADV202 asserts DREQ/ as soon as enough data, as set by the burst size, is present in the assigned FIFO.

DMA DREQ/DACK mode allows several possible configurations :

1. Single DMA DREQ/DACK mode
DREQ/ is asserted for every data access to the data FIFO.
Refer to datasheet, figures 5,6,9,10

2. Burst DMA DREQ/DACK mode
DREQ/ is asserted once at the beginning of the burst.
Refer to datasheet, figures 13, 14, 17, 18.
3. DREQ/ pulse width configuration – controlled by DACK/ and RD/ or WR/.
DREQ/ is asserted until the host responds with a DACK/ and RD/ or WR/ assert.
Refer to datasheet, figure 14.
4. DREQ/ pulse width configuration – DREQ/ pulse width is programmable and is fixed. This parameter can be programmed in the EDMOD0/1 registers.
Refer to datasheet, figure 13.

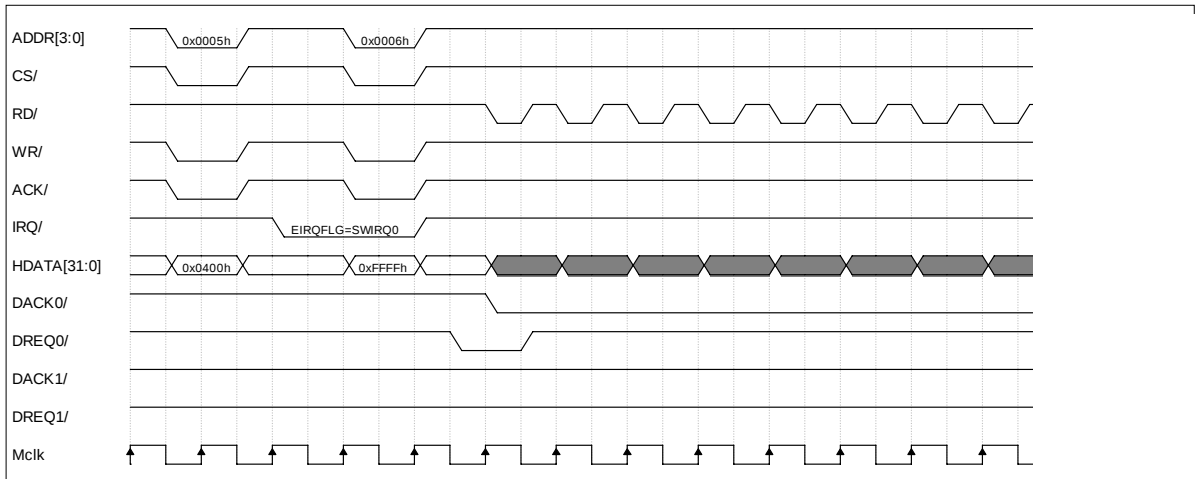
Refer to datasheet for detailed timing diagram of all of these modes.

The table below shows examples of EDMOD register settings for the different DREQ/DACK DMA modes and configurations.

	DREQ/ pulse configuration *	DMA Mode	Burst Cycles	EDMOD0	EDMOD1
Encode	controlled by DACK/ and RD/ or WR/	Single	n/a	0Ah	0802h
Encode	fixed pulse width to 4 JCLK cycles	Single	n/a	200Ah	0802h
Decode	controlled by DACK/	Single	n/a	0Ah	0802h
Decode	fixed pulse width to 4 JCLK cycles	Single	n/a	200Ah	0802h
Encode	controlled by DACK/ and RD/ or WR/	Burst	16	52h	0802h
Encode	fixed pulse width to 4 JCLK cycles	Burst	16	2052h	0802h
Decode	controlled by DACK/	Burst	16	52h	0802h
Decode	fixed pulse width to 4 JCLK cycles	Burst	16	2052h	0802h

* Refer to datasheet, figure 24 for Jclk definition. For maximum Jclk frequency of 150Mhz, Jclk cycle is approx. 6.7 ns.

Table 2 DMA configuration parameters and their register settings



*Figure 16 DMA DREQ/DACK Burst encode –
Start program and read first burst out of CODE FIFO -
EDMOD0 = 0x0012h*

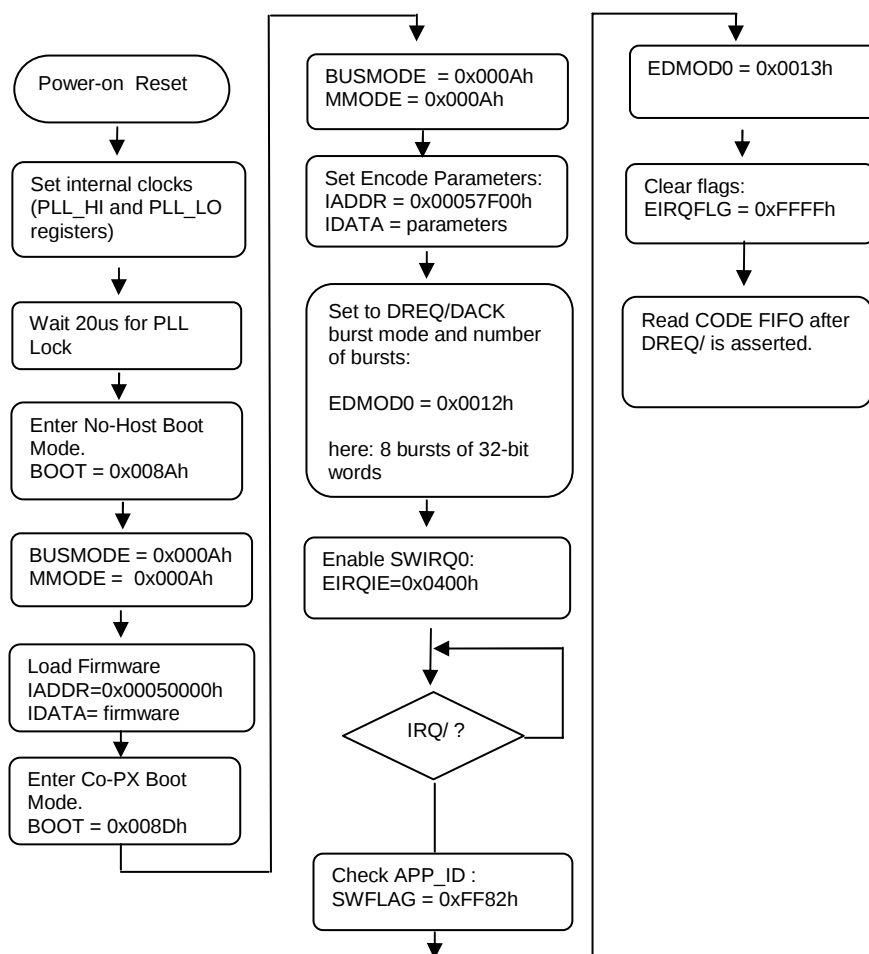
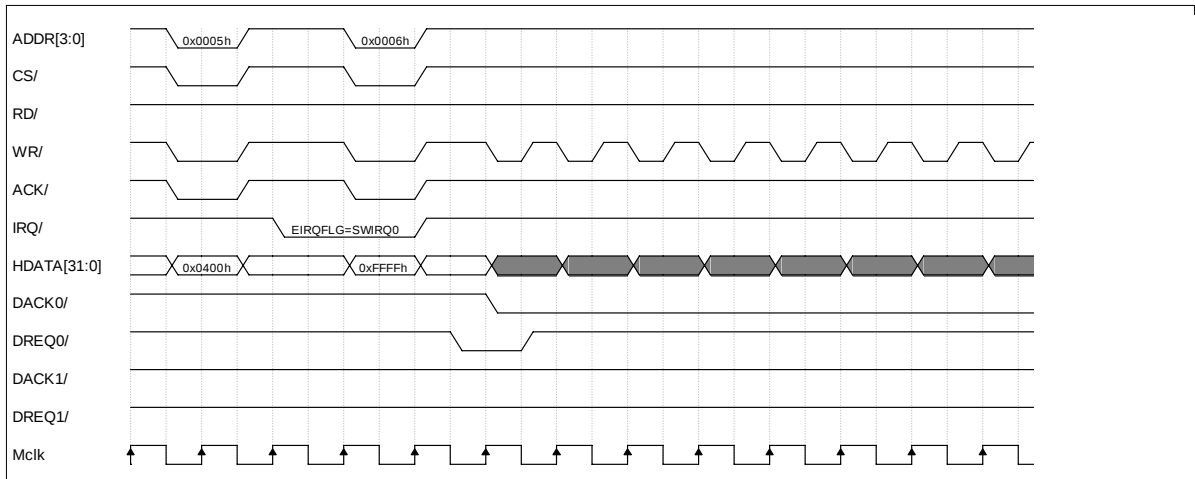


Figure 17 Encode Routine - DMA DREQ/DACK Burst mode



*Figure 18 DMA DREQ/DACK Burst Decode –
Start program and write first burst to CODE FIFO -
EDMOD0 = 0x0012h*

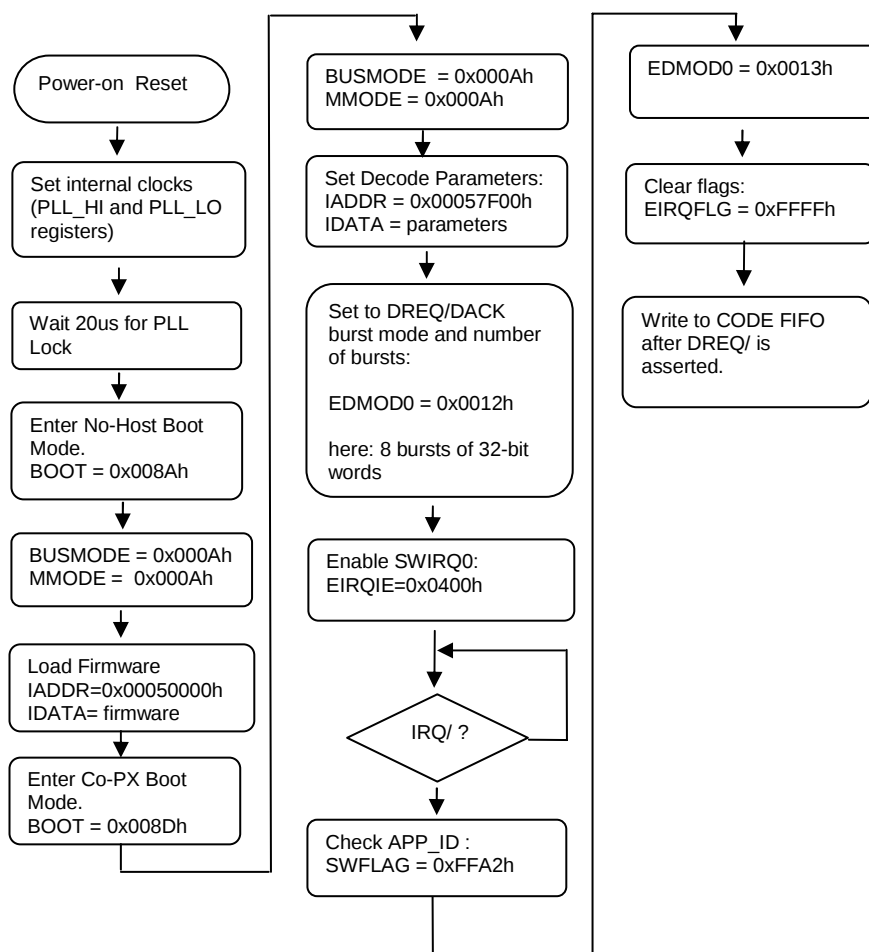


Figure 19 Decode routine - DREQ/DACK DMA burst mode

Encode/Decode routine for JDATA mode – 16-bit Host

JDATA mode is a synchronous mode. Compressed data transfers are synched to MCLK. As all modes it requires configuration of direct registers, firmware load, parameter load and indirect register configuration using the Normal Host protocol as mentioned earlier in the document.

But since the 8-bit JDATA codestream requires HDATA pins [31-24], only 16-bit host mode is available with JDATA.

There are also some additional restrictions on the PLL and clock requirements in JDATA mode:

$JCLK \geq 4 \times MCLK$

This means for 141-pin parts, the max. Mclk can be 37.125 MHz. And this is approx. the compressed data rate also.

In encode mode, data transfer occurs when VALID is asserted by the ADV202 whenever data is available and the host should respond with a HOLD de-assert.

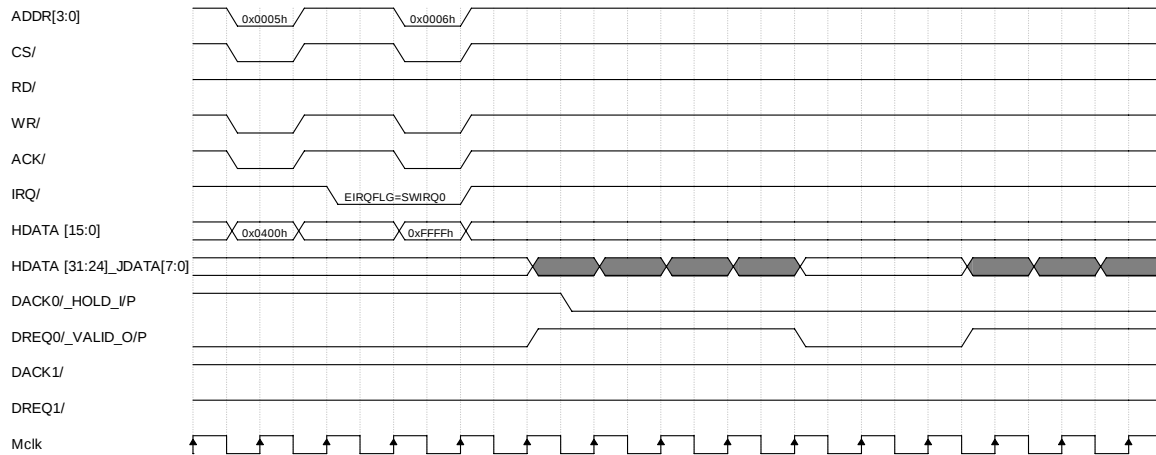
In decode mode, data has to be transferred to the ADV202 as soon as VALID is asserted by the ADV202.

VALID is always an output and HOLD is always an input.

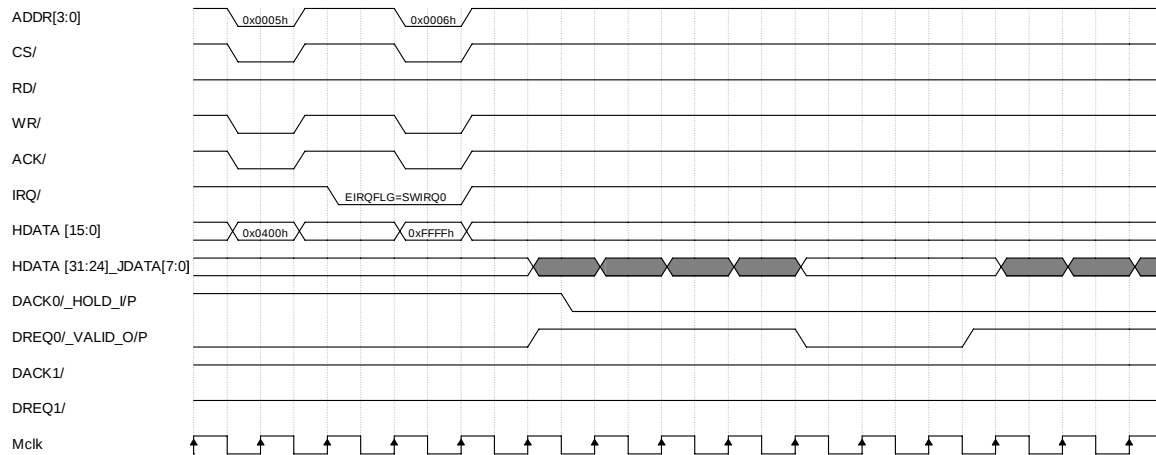
If HOLD and VALID are set to active high polarity, the following truth table will apply on each rising edge of MCLK:

HOLD	VALID	JDATA
0	0	no valid data transfer
1	0	no valid data transfer
0	1	valid data transfer
1	1	no valid data transfer

Note: on ES3 samples, after enabling the JDATA interface HOLD must be pulled to its inactive state in order to initiate a VALID assertion.



*Figure 21 JDATA Mode Encode [ADV202 ES5] –
VALID and HOLD are programmed for active high polarity*



*Figure 22 JDATA Mode Decode [ADV202 ES5] –
VALID and HOLD are programmed for active high polarity*

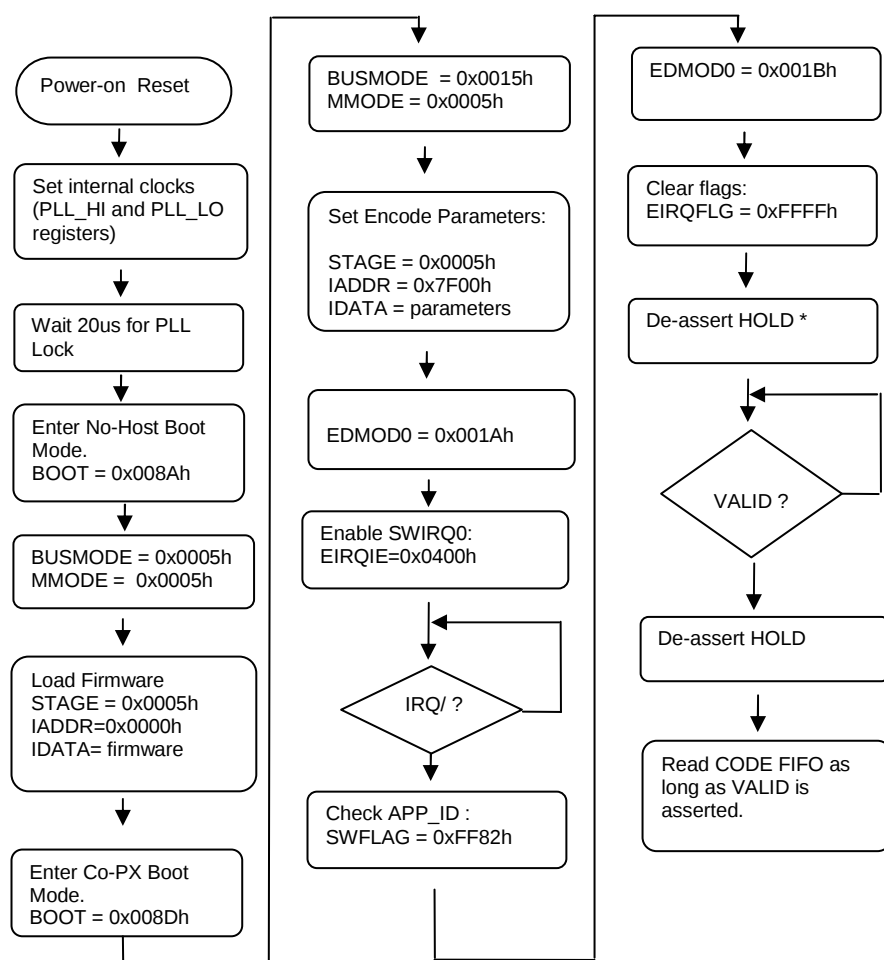
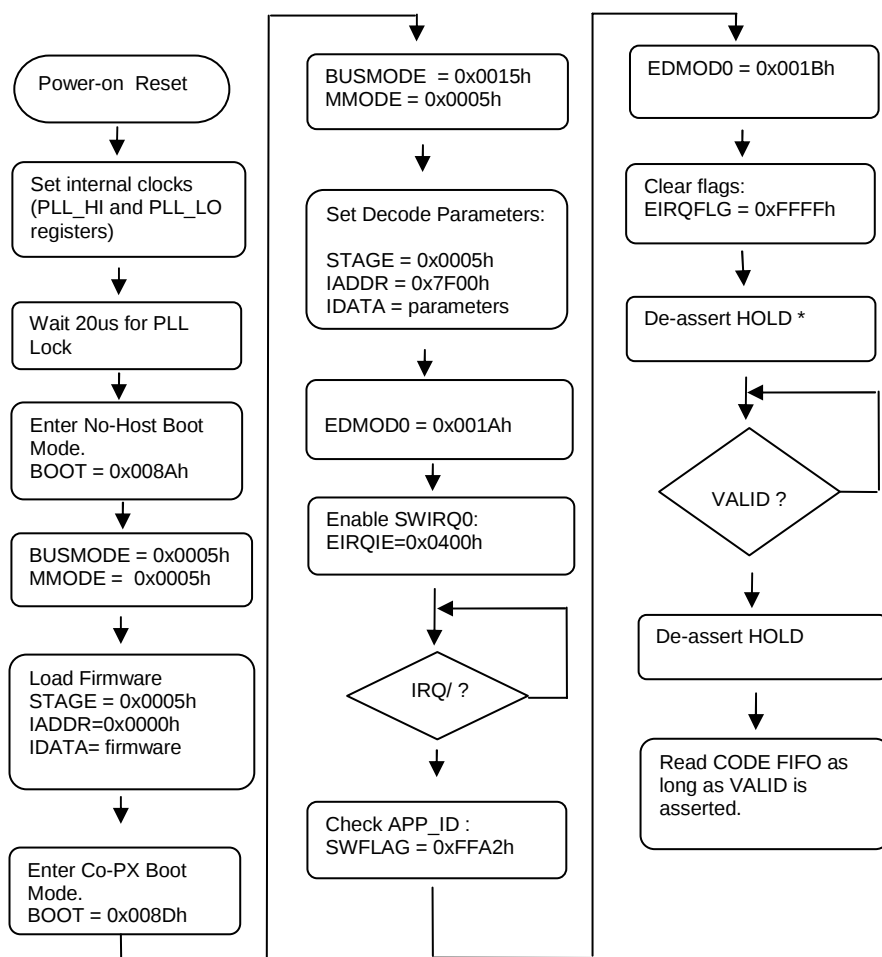


Figure 23 JDATA encode routine

*Only for ADV202- ES3. On ADV202 ES3 samples HOLD must be de-asserted after JDATA is enabled in order to initiate a VALID assert.



*Only for ADV202- ES3. On ADV202 ES3 samples HOLD must be de-asserted after JDATA is enabled in order to initiate a VALID assert.

Figure 24 JDATA Mode Decode

Custom Specific Mode Encode and Decode routines

For any input format that does not comply to the standards listed under VFORMAT [JPEG2000 parameters, table 8], Custom Specific mode must be used.

There are certain input limitations in Custom Specific mode:

1. Maximum Pixel Samples per image : 1 048 576 Msamples/image [Luma and Chroma]
2. Maximum Number of codeblocks per image : 610
Number of codeblocks/ image can be controlled with the cbsize JPEG2000 parameter, see table 8.
Refer to TechNote *TechNote_codeblock_1.pdf* for more information on estimating number of codeblocks per image:

ftp://ftp.analog.com/pub/Digital_Imaging/ADV202_ApplicationNotes/ADV202_Getting_Started

In Custom Specific Mode, it is required by the user to configure dimension registers for the non-standard video input.

The following registers must be configured in Custom Specific mode:

Indirect Register Address	Name	Description
0xFFFF0400	PMODE1	Pixel/Video Format
0xFFFF040C	XTOT	Total Samples per line
0xFFFF0410	YTOT	Total lines per frame/field
0xFFFF0414	F0_START	Start Line of Field 0
0xFFFF0418	F1_START	Start Line of Field 1
0xFFFF041C	V0_START	Start of active video Field 0
0xFFFF0420	V1_START	Start of active video Field 1
0xFFFF0424	V0_END	End of active video Field 0
0xFFFF0428	V1_END	End of active video Field 1
0xFFFF042C	PIXEL_START	Horizontal start of active video
0xFFFF0430	PIXEL_END	Horizontal end of active video

Getting Started with the ADV202

0xFFFF0448	PMODE2	Pixel Mode 2
0xFFFF044C	VMODE	Video Mode

Table 3 ADV202 Dimension Registers

For a detailed bit description of each register, refer to the *ADV202 User's Guide, Chapter 4*.

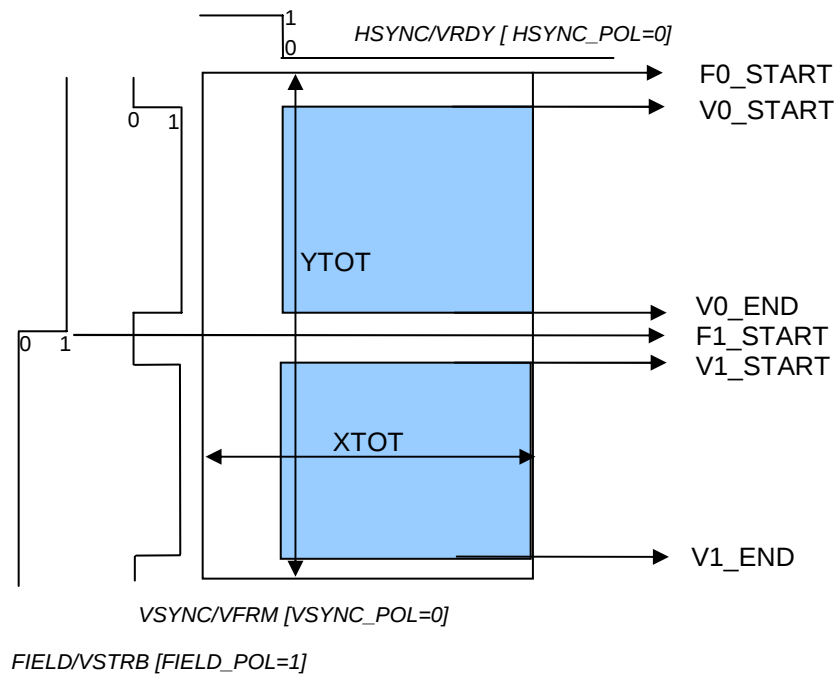


Figure 25 Dimension Registers settings and HVF timing relationship

In Custom Specific mode the video is expected to be input on VDATA. The input can either be input with HVF syncs or with EAV/SAV codes.

Polarities for H, V, F will be programmed with the JPEG2000 parameters [PICFG parameter] or in the PMODE2 and VMODE registers.

For non-interlaced formats or frame based input formats, such as still images, register settings for F1_START, V1_START, V1_END are not required.

If HVF syncs are used:

Getting Started with the ADV202

Only the first active HSYNC transition at the start of active video is recognized and starts the horizontal counters, this means there is no HSYNC pulsewidth requirement. HSYNC must be asserted at the start of each active line.

For vertical synchronisation, the first in-active edge of VSYNC will initiate synchronisation. The ADV202 will then synchronize according to the dimension register settings and the internal counters.

F0_START, F1_START is only used in decode mode to identify at which line number the Field bit will transition from 0 to 1. In decode master, the Field output transition is determined by the Field dimension register settings.

In decode slave the Field transition is determined by the input signal on the FIELD input pin.

Encode mode

In encode mode the part is always in slave configuration. Input data can be accompanied by separate H,V,F signals or embedded timing codes. In both cases the dimension register settings must reflect the video standard of the input. The part synchronizes itself to the incoming sync signals.

Using the value of registers Vx_START, Vx_END, PIXEL_START, PIXEL_END the active video region to be processed is calculated. This does apply to all input modes using the VDATA bus, in HIPI mode only values for XTOT and YTOT need to be programmed, all other dimension register values are ignored in HIPI mode [refer to "ADV202_HIPI_mode" application note].

Decode slave

In decode slave mode, video output is synced to the incoming HVF signals. The dimension registers settings must match the output format as set by the dimension registers.

Decode master

VSYNC, HSYNC, FIELD or EAV/SAV codes are generated according to the register settings of: XTOT, YTOT, F0_START, F1_START, F0_END, F1_END, V0_START, V1_START, V0_END, V1_END, PIXEL_START, PIXEL_END. To enable the generation of these timing signals in decode mode, VMODE register must be programmed to decode master mode.

Getting Started with the ADV202

Configuration Example for Custom Specific Mode:

VGA, 10-bit YCbCr data with 800x525@60Hz total and 640x480@60Hz active resolution.

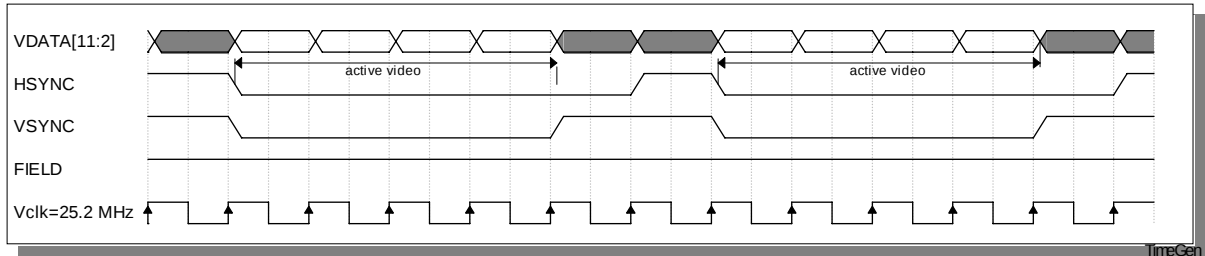


Figure 26 Custom Specific Example – encode timing

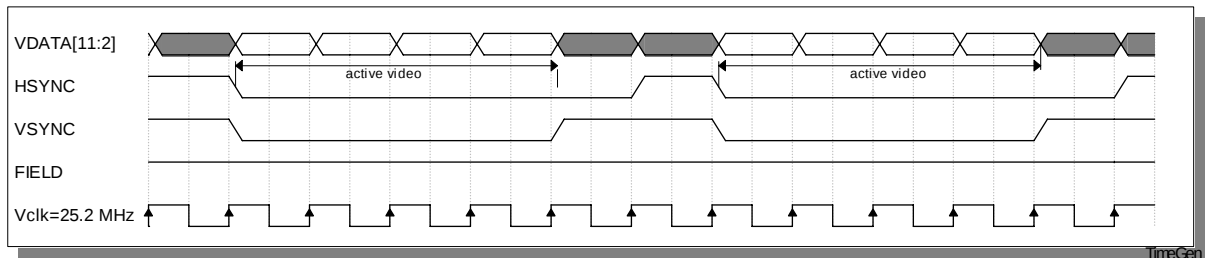


Figure 27 Custom Specific Example – decode master timing

Indirect Register Address	Name	Register Setting for Encode	Register Settings for Decode
0xFFFF0400	PMODE1	0x0005h	0x0005h
0xFFFF040C	XTOT	0x0640h [1600]	0x0640h
0xFFFF0410	YTOT	0x020Dh [525]	0x020Dh
0xFFFF0414	F0_START	0x0001h [1]	0x0001h
0xFFFF0418	F1_START	0x0000h [0]	0x0000h
0xFFFF041C	V0_START	0x002Eh [46]	0x002Eh
0xFFFF0420	V1_START	0x0000h [0]	0x0000h

0xFFFF0424	V0_END	0x020Dh [525]	0x020Dh
0xFFFF0428	V1_END	0x0000h [0]	0x0000h
0xFFFF042C	PIXEL_START	0x0001h [1]	0x0001h
0xFFFF0430	PIXEL_END	0x0500h [1280]	0x0500h
0xFFFF0448	PMODE2	0x0003xh for polarity as in figure 26	0x0003h
0xFFFF044C	VMODE	0x0086h HVF	0x0084h

Table 4 Custom Specific example for 640x480x60Hz – Dimension register settings

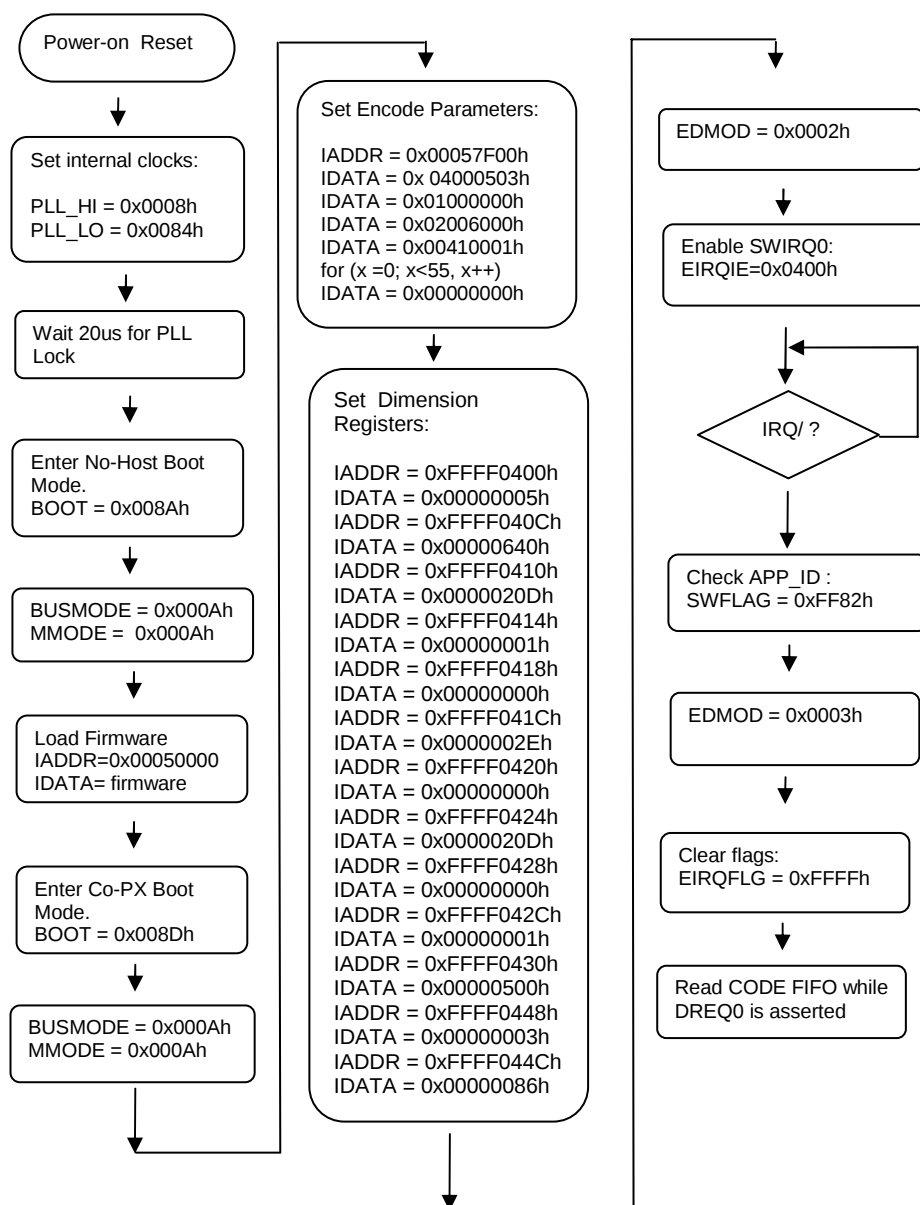


Figure 28 Custom Specific Example – Encode programming sequence in DCS mode

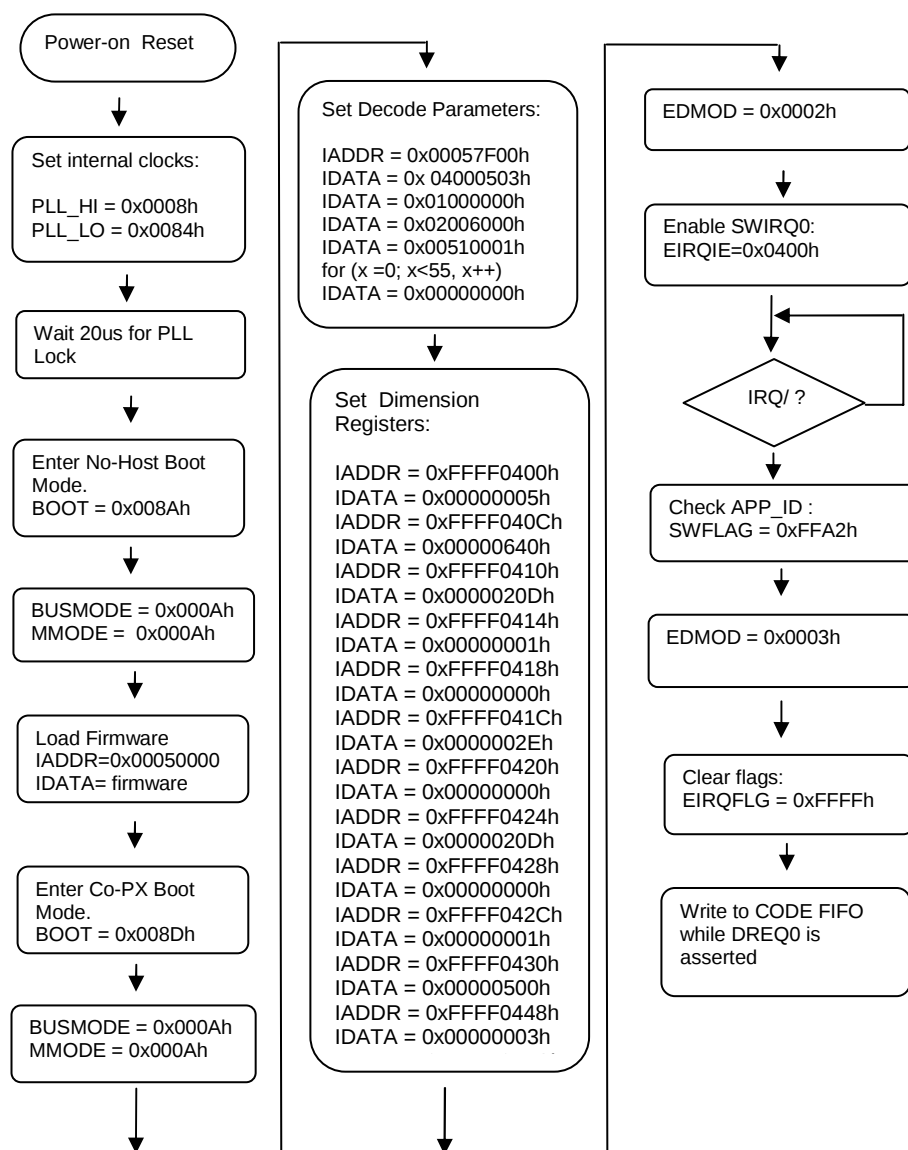


Figure 29 Custom Specific Mode – Decode programming sequence in DCS mode

Raw Pixel Mode Encode and Decode routines

In Raw Pixel or Raw Video mode pixel data is again input on the VDATA bus. The input data is expected to have no timing information or blanking region associated with it. The active region to be captured and frame transitions are determined by the Hsync, Vsync, Field inputs and the settings for XTOT and YTOT.

For Raw Pixel mode the encode parameter VFORMAT must be set to Custom Specific mode.

Raw Pixel mode supports single component input and component input in YCbYCr format [note: this is not the conventional format used for video, which is CbYCrY]. JDATA mode is not available in this mode.

Pixel data should be clocked into the part with 1 Vclk cycle per input sample.

Configuration Example for Raw Pixel Mode:

VGA, single component 8-bit Y data with 640x480@60Hz active resolution.

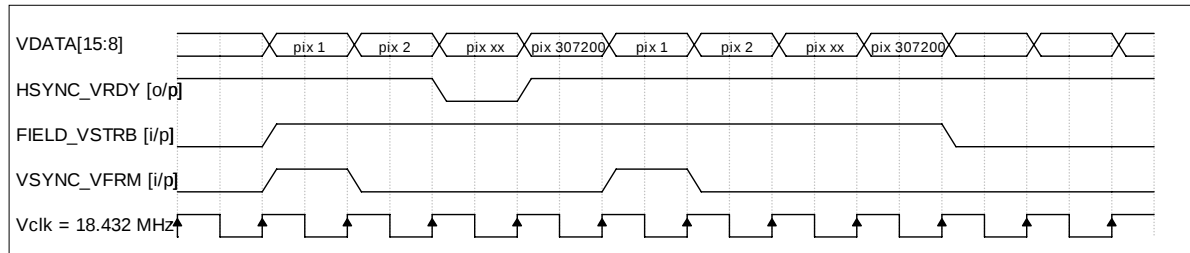


Figure 30 Raw Pixel Mode – Encode timing – Sync signals are programmed for active high polarity

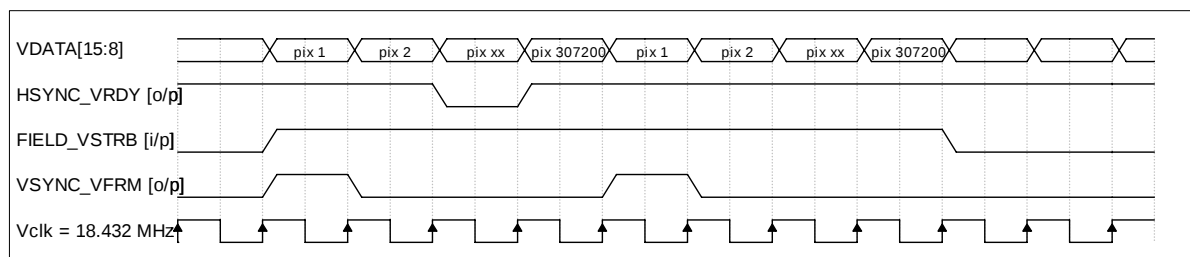


Figure 31 Raw Pixel Mode – Decode timing – Sync signals are programmed for active high polarity

The figure above shows H, V, F to be programmed to active high polarity in both encode and decode mode.

HSYNC_VRDY Output : When programmed to active high polarity the ADV202 is ready to take in pixel data as long as this signal is HIGH.

Getting Started with the ADV202

FIELD_VSTRB Input : When programmed to active high, this signal must be asserted HIGH before the first pixel and stay asserted HIGH during the entire transfer.

VSYNC_VFRM Input : Input in encode mode and output in decode mode. When programmed to active high, this signal must be asserted on every first pixel of the frame.

Register Settings for Raw Pixel Mode example:

Indirect Register Address	Name	Register Setting for Encode	Register Settings for Decode
0xFFFF0400	PMODE1	0x0004h	0x0004h
0xFFFF040C	XTOT	0x0280h [640]	0x0280h [640]
0xFFFF0410	YTOT	0x01E0h [480]	0x01E0h [480]
0xFFFF0414	F0_START	0x0001h	0x0001h
0xFFFF0418	F1_START	0x0000h	0x0000h
0xFFFF041C	V0_START	0x0001h	0x0001h
0xFFFF0420	V1_START	0x0000h	0x0000h
0xFFFF0424	V0_END	0x01E0h	0x01E0h
0xFFFF0428	V1_END	0x0000h	0x0000h
0xFFFF042C	PIXEL_START	0x0001h	0x0001h
0xFFFF0430	PIXEL_END	0x0280h	0x0280h
0xFFFF0448	PMODE2	0x003Fh for polarity as in figure 30	0x003Fh for polarity as in figure 31
0xFFFF044C	VMODE	0x00A6h - HVF	0x00A5h

Table 5 – Dimension register settings for Raw Pixel mode example

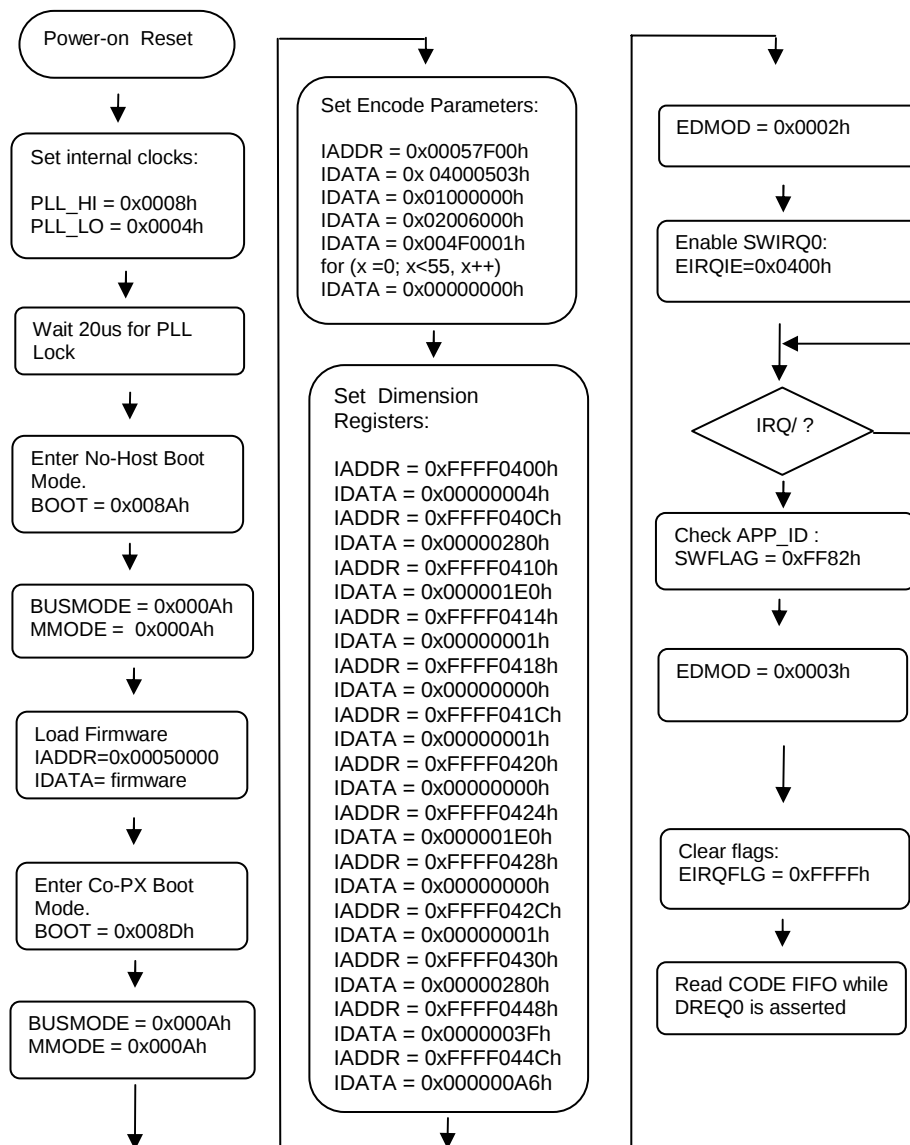


Figure 32 Raw Pixel Mode – Encode programming sequence using DCS mode for compressed data output

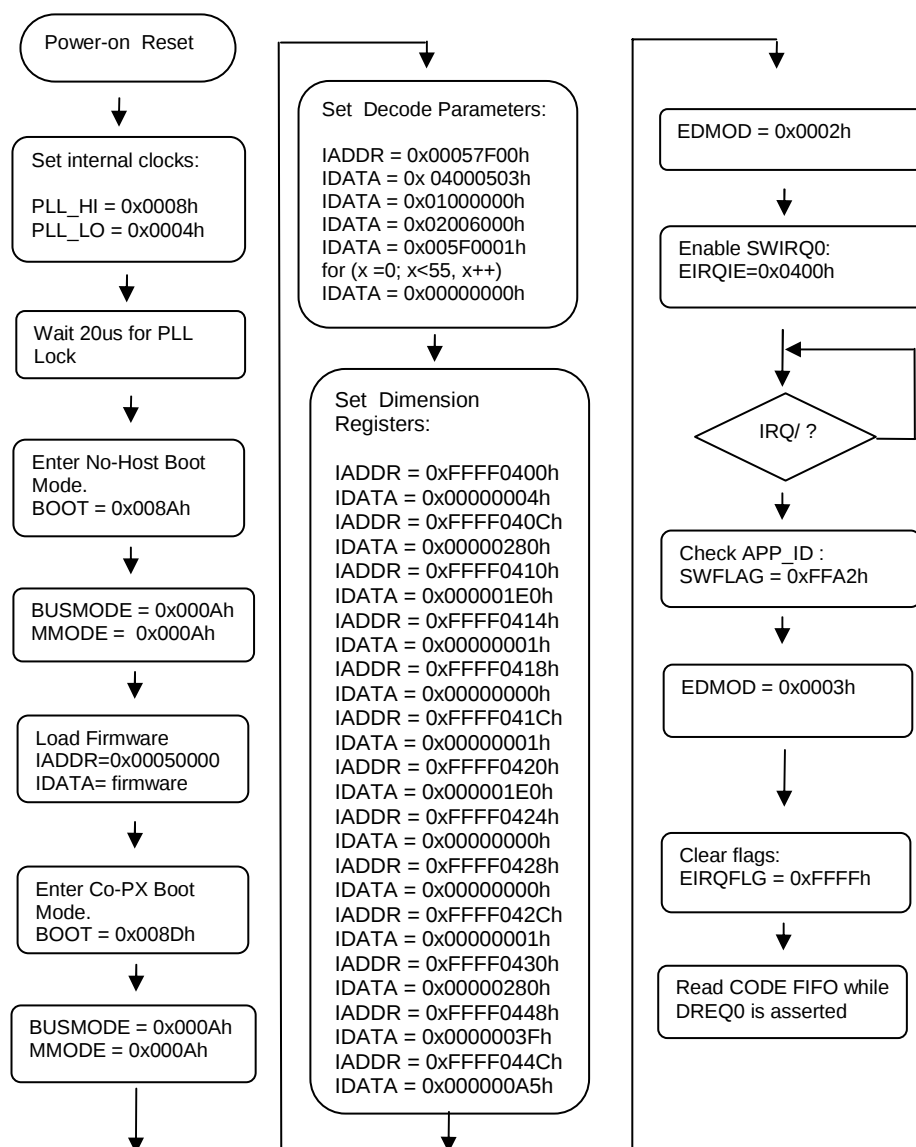


Figure 33 Raw Pixel Mode – Decode programming sequence using DCS mode for compressed data input

Appendix

PLL settings and internal clocks

Please refer to the datasheet for a detailed description of the ADV202 internal clock and PLL register settings.

The table below show recommended register settings for most common MCLK frequencies used.

Video Standard	MCLK input [MHz]	PLL_HI address=0xEh	PLL_LO address=0xFh
ITU.R.BT656	27	0x0008	0x0004
1080i SMPTE274M	74.25	0x0008	0x0084
1080i SMPTE274M	37.125	0x0008	0x0004
1080i SMPTE274M	25	0x0008	0x0006
525p SMPTE293M	27	0x0008	0x0004
625p ITU.R.BT1358	27	0x0008	0x0004

Table 6 Recommended PLL register settings

BUSMODE and MMODE register settings

	BUSMODE address=0x8h	MMODE address=0x9h
32-bit interface	0x000Ah	0x000Ah
16-bit interface	0x0005h	0x0005h
JDATA mode	0x0015h	0x0005h

Table 7 BUSMODE and MMODE register settings for 16-bit and 32-bit interfaces

Getting Started with the ADV202

JPEG2000 Parameters - Encode

Big Endian Byte offset of parameter	Number of bytes in parameter	Parameter Name	Description
0x0	1	VFORMAT	Video Standard 0 = NTSC 4:2:2 1 = PAL 4:2:2 2 = 1080i/60 Luminance 3 = 1080i/60 Chrominance 4 = Custom 5 = 720/60p Luminance 6 = 720/60p Chrominance 7 = Reserved 8 = NTSC de-interlaced 9 = PAL de-interlaced
0x1	1	PREC	Precision 0 = 8bit 1 = 10bit 2 = 12bit
0x2	1	XFORMLE V	Numer of Wavelet Transform Level 1 = 1 level of transform 2 = 2 levels of transform 3 = 3 levels of transform 4 = 4 levels of transform 5 = 5 levels of transform 6 = 6 levels of transform
0x3	1	UNI	Component polarity 0 = Bipolar C, Bipolar Y 1 = Unipolar C, Bipolar Y 2 = Bipolar C, Unipolar Y 3 = Unipolar C, Unipolar Y

Getting Started with the ADV202

0x4	1	CBSIZE	Codeblock Dimensions 0 = 32x32 [will not work for 1080i or 720p] 1 = 64X32 2 = 64X64 3 = 128X32
0X5	1	WKERNEL /QUANT	Wavelet Kernel and Quantization factor 0 = Irreversible 9x7 using fixed table [lossy] 1 = Reversible 5x3 [lossless] 2 = Irreversible 5x3 [lossy]
0X6	1	STALLPAR	Skip Fields in Encode mode 0 = capture all fields 1-63= skip this number of fields 255 = stop capture
0X7	1	ATTRYPE	Attribute Data output format : 0 = Do not output attribute data. 1 = Use ADV-JP2000 attribute data format 2 = Log2 Convex Hull without Adjustments 3 = Log2 Convex Hull with Quantization Adjust 4 = ADV202 field header output to ATTR FIFO 5 = ADV202 field header and packet byte length output to ATTR FIFO
0x8	1	RCTYPE	Rate Control Algorithm 0 = None (no truncation) 1 = Target Size per video field/image within 5% of RCVAL. [bytes/image] 2 = Target Quality per video field/image. 3 = Multi-layered Target Size per video field/image. Requires RCVAL and LTARGET parameters also. 4 = Multi-layered Target Quality per video field/image. Requires RCVAL and LTARGET parameters also.



TechNote

Getting Started with the ADV202

0x9	3	RCVAL	Target Size or Quality factor. If RCTYPE = 0 then RCVAL = n/a If RCTYPE = 1 then RCVAL = Size of field/image in bytes. If RCTYPE = 2 then RCVAL = Quality factor. If RCTYPE >= 3 then RCVAL = Number of layers.																		
0xC	1	J2KPROG	JPEG2000 Progression Style 0 = LRCP 1 = RLCP 2 = RPCL 3 = PCRL 4 = CPRL 8 = Use Complex Attribute Data format Mode 0/1 9 = Use Complex Attribute Data format Mode 2 10 = Use Complex Attribute Data format Mode 3																		
0xD	1	PICFG	Pixel Interface Configuration Settings: <table><tr><td>BIT</td><td>Description</td></tr><tr><td>7</td><td>reserved (always write 0)</td></tr><tr><td>6</td><td>HVF or EAV/SAV 1 = use external HVF pins 0 = use embedded SAV/EAV</td></tr><tr><td>5</td><td>reserved (write 0)</td></tr><tr><td>4</td><td>reserved (write 0)</td></tr><tr><td>3</td><td>Field Polarity (1=pos 0=neg)</td></tr><tr><td>2</td><td>HSYNC Polarity (1=pos 0=neg)</td></tr><tr><td>1</td><td>VSYNC Polarity (1=pos 0=neg)</td></tr><tr><td>0</td><td>VCLK Polarity (1=pos 0=neg)</td></tr></table>	BIT	Description	7	reserved (always write 0)	6	HVF or EAV/SAV 1 = use external HVF pins 0 = use embedded SAV/EAV	5	reserved (write 0)	4	reserved (write 0)	3	Field Polarity (1=pos 0=neg)	2	HSYNC Polarity (1=pos 0=neg)	1	VSYNC Polarity (1=pos 0=neg)	0	VCLK Polarity (1=pos 0=neg)
BIT	Description																				
7	reserved (always write 0)																				
6	HVF or EAV/SAV 1 = use external HVF pins 0 = use embedded SAV/EAV																				
5	reserved (write 0)																				
4	reserved (write 0)																				
3	Field Polarity (1=pos 0=neg)																				
2	HSYNC Polarity (1=pos 0=neg)																				
1	VSYNC Polarity (1=pos 0=neg)																				
0	VCLK Polarity (1=pos 0=neg)																				

Getting Started with the ADV202

0xE	1	QFACT	Quantization factor 255 = 255/256 factor 255 = 255/256 factor 254 = 254/256 factor . . 2 = 2/256 factor 1 = 1/256 factor 0 = 256/256 factor (1x)
0xF	1	COD_STY LE	Output Code Stream Format: COD_STYLE [2:0] : 0 = ADV202 Raw Format 1 = J2C Format 2 = JP2 Format COD_STYLE [3] : Reserved COD_STYLE [7:4] applies to J2C or JP2 Formats only. COD_STYLE [7]: 0 = Packet headers with packet body 1 = use PPT COD_STYLE [6]: 0 = no PLT 1 = include PLT COD_STYLE [5]: 0 = no SOP 1 = include SOP COD_STYLE [4] : 0 = no EPH 1 = include EPH

Getting Started with the ADV202

0x10	2*19	STEPSIZE S [18:0]	Quantization Stepsizes using 5 exponent bits plus 11 mantissa bits For more detail refer to the ISO/IEC15444-1 specification, section Annex E. There is one step size assigned per subband (note there is only one set of stepsizes defined which all components use). The order of subband stepsizes for this parameter list is as follows for 6 transform levels: LL 1HL (for > 0 transform levels) 1LH (for > 0 transform levels) 1HH (for > 0 transform levels) 2HL (for > 1 transform levels) 2LH (for > 1 transform levels) 2HH (for > 1 transform levels) 3HL (for > 2 transform levels) 3LH (for > 2 transform levels) 3HH (for > 2 transform levels) 4HL (for > 3 transform levels) 4LH (for > 3 transform levels) 4HH (for > 3 transform levels) 5HL (for > 4 transform levels) 5LH (for > 4 transform levels) 5HH (for > 4 transform levels) 6HL (for > 5 transform levels) 6LH (for > 5 transform levels) 6HH (for > 5 transform levels)
------	------	-------------------------	--

Getting Started with the ADV202

0x36	1	LOAD_SS	Bit 0 of this byte MUST BE '0' prior to loading the stepsizes and/or QFACT. After writing the stepsizes and/or QFACT, the user must set Bit 0 to '1' to tell the firmware that the step-sizes/QFACT are valid and ready for use. Once the firmware loads these new stepsizes/QFACT, this bit will again be reset to '0'.
0x37	1	LOAD_VW	Visual Weighting Options: 0 = No visual weighting or use previously loaded visual weights. 1 = Load custom visual weights using VW_Y, VW_CB and VW_CR parameters. 2 = Load visual weights as recommended in ISO/IEC 15444-3 standard for Motion Jpeg2000
0x38	2*19	VW_Y	Visual Weighting Factors for Luminance.
0x5E	2*19	VW_CB	Visual Weighting Factors for Cb
0x84	2*19	VW_CR	Visual Weighting Factors for Cr
0xAA	6	RESERVE D	Reserved

Getting Started with the ADV202

0xB0	4*16	LTARGET [1:16]	Target Size or Target Quality for each layer in Multi-layer mode
------	------	-------------------	--

Table 8 – JPEG2000 parameters : Encode

JPEG2000 Parameters - Decode

Big Endian Byte offset of parameter	Number of bytes in paramete r	Parameter Name	Description
0x0	1	VFORMAT	Video Standard 0 = NTSC 4:2:2 1 = PAL 4:2:2 2 = 1080i Luminance 3 = 1080i Chrominance 4 = Custom 5 = 720/60p Luminance 6 = 720/60p Chrominance 7 = Reserved 8 = NTSC de-interlaced 9 = PAL de-interlaced
0x1	1	PREC	Precision 0 = 8bit 1 = 10bit 2 = 12bit
0x2	1	Reserved	
0x3	1	UNI	Component polarity 0 = Bipolar C, Bipolar Y 1 = Unipolar C, Bipolar Y 2 = Bipolar C, Unipolar Y 3 = Unipolar C, Unipolar Y
0x4	1	Reserved	

Getting Started with the ADV202

0x5	1	Reserved	
0x6	1	Reserved	
0x7	1	Reserved	
0x8	1	Reserved	
0x9	1	PICFG	Pixel Interface Configuration Settings: BIT Description: 7 reserved (always write 0) 6 Use HVF pins 1 = use external HVF pins 0 = use embedded SAV/EAV 5 reserved (write 0) 4 0 = Decode Slave 1 = Decode Master 3 Field Polarity (1=pos 0=neg) 2 HSYNC Polarity (1=pos 0=neg) 1 VSYNC Polarity (1=pos 0=neg) 0 VCLK Polarity (1=pos 0=neg)
0xA	1	DRES	Decode Resolution settings: BIT Description: 7 Scale Image Size Enable: 0 = keep image size the same 1 = scale image down to desired resolution size. Refer to bit 2:0 for settings. 6:3 reserved (write 0) 2:0 Highest Resolution Level to Decode (up to maximum of XFORMLEV parameter): 0 = Decode all resolutions 1 = Decode to ¼ resolution 2 = Decode to 1/16 resolution 3 = Decode to 1/64 resolution 4 = Decode to 1/256 resolution 5 = Decode to 1/1024 resolution 6 = Decode to 1/4096 resolution

0xB	1	COD_STYLE	Output Code Stream Format: COD_STYLE [2:0] : 0 = ADV202 Raw Format 1 = J2C Format 2 = JP2 Format
-----	---	-----------	---

Table 9 – JPEG2000 Parameters – Decode

JPEG2000 Parameter Description

VFORMAT

NTSC 4:2:2

This setting should be used only for a resolution of 720x243@60 fields/sec.

PAL 4:2:2

This setting should be used only for a resolution of 720x288@50 fields/sec.

1080i Luminance

This setting should be used only for a resolution of 1920x540@60 fields/sec.

1080i Chrominance

This setting should be used only for a resolution of 1920x540@360 fields/sec.

Custom specific

This setting must be used for all settings that do not conform to VFORMAT = 0-9. Refer to the previous section for programming instructions for Custom Specific mode.

720/60p Luminance

This setting should be used only for a resolution of 1280x720@60 frames/sec.

720/60p Chrominance

This setting should be used only for a resolution of 1280x720@60 frames/sec.

NTSC de-interlaced

This setting should be used only for NTSC standard definition at a resolution of 720x243@60 fields/sec. The input is in interlaced format but the compression is on a frame by frame basis. The ADV202 will compress an odd and even field together. This can improve compression efficiency of up to 40-50%.

PAL de-interlaced

This setting should be used only for PAL standard definition at a resolution of 720x288@50 fields/sec. The input is in interlaced format but the compression is on a frame by frame basis. The ADV202 will compress an odd and even field together. This can improve compression efficiency of up to 40-50%.

CBSIZE

Codestream generation is facilitated with large codeblocks but code block height should be small to minimize memory usage on the ADV202. Therefore a cbsize of 32x32 will not work for 1080i or 720p modes.

The most robust for 1080i and 720p modes is a cbsize of 128x32. This cbsize will also provide lowest latency.

The smaller the codeblocks, the larger the number of codeblocks generated. The MIPs required to perform Rate Control and J2K generation are directly related to the number of codeblocks present.

For recommendation on how to estimate the number of codeblocks per image, refer to the technical note *TechNote_Codeblocks_1.pdf*.

ATTRTYPE

If the ATTRTYPE parameter is set to 1, 2, or 3, then the ADV202 will be configured to output codeblock attribute data via the ATTR FIFO. Also, the host must enable both DMA channels when using this feature, one assigned to the CODE FIFO, and the other to the ATTR FIFO.

If ATTRTYPE is set to 4, then the 16-byte ADV202 compressed image header will be output to the ATTR FIFO instead of being placed at the top of each compressed image. This will allow systems to obtain this header information without the need to parse through the code stream data.

If ATTRTYPE is set to 5, then a more detailed byte-length summary will be output to the ATTR FIFO to support external merging of HD Luminance and HD Chrominance streams from two different ADV202s.

Refer to technical note *TechNote_ADV202_Compressed_Data_Format_1.pdf*.

RCTYPE and RCVAL

At the time of writing this version of the firmware has 5 options for performing the rate-control algorithm:

No Truncation:

Each compressed codeblock will be assembled into the codestream without being truncated, thus providing the maximum possible quality. This will also result in large variations in the number of bytes generated per compressed image. The compressed codeblocks will be truncated in the other modes described next.

Target Size per video field/image:

The number of bytes per compressed field/image is provided in the RCVAL Parameter. If Target Size is chosen as parameter the data output rate is to be within 5% of the programmed value.

For example, to achieve 12 Mbits per second (assuming NTSC 60 fields/sec), RCVAL should be set to $(12,000,000 / (8 * 60)) = 25,000$ bytes/field = 0x0061A8h. The rate-control algorithm will try to adjust the quality of successive images to achieve this target, but variations from field-to-field will occur.

Target Quality per video field/image:

The target quality value is provided in the RCVAL Parameter.

As this value increases the quality [and field/image size] decreases.

If Target Quality is chosen as parameter, the picture quality will be held at a constant while the data output rate will change.

Assuming the default Quantization Stepsizes are being used, Highest Quality is achieved at RCVAL=0x000100 or below, and Lowest Quality is achieved at RCVAL=0x000A00 and above.

Multi-layer Target Size:

As (2) above but this option will support SNR scalable code streams. The number of layers should be programmed into the RCVAL parameter. The target byte length for each layer must also be programmed. This is done in setting the LTARGET value.

RCVAL is programmed for the number of layers. RCVAL can be programmed from a minimum value of 1 up to a maximum of 16.

Multi-layer Target Quality: RCVAL is programmed for the number of layers. RCVAL can be programmed from a minimum value of 1 up to a maximum of 16.

Getting Started with the ADV202

Video	Horizontal Pixels per Field	Lines per Field	Field Rate	Data output rate Mbits/ sec	Bytes per Field	RCVAL
NTSC	720	243	59.94	20	41708	0x00A2E2
NTSC	720	243	59.94	15	31281	0x007A31
NTSC	720	243	59.94	10	20854	0x005176
NTSC	720	243	59.94	9	18750	0x004950
NTSC	720	243	59.94	8	16683	0x00412B
NTSC	720	243	59.94	7	14598	0x003905
NTSC	720	243	59.94	6	12513	0x0030E0
NTSC	720	243	59.94	5	10417	0x0028BB
NTSC	720	243	59.94	4	8342	0x002095
NTSC	720	243	59.94	3	6256	0x001870
NTSC	720	243	59.94	2	4167	0x00104A
NTSC	720	243	59.94	1	2085	0x000825
PAL	720	288	50	20	50000	0x00C350
PAL	720	288	50	15	37500	0x00927C
PAL	720	288	50	10	25000	0x0061A8
PAL	720	288	50	9	22500	0x0057E4
PAL	720	288	50	8	20000	0x004E20
PAL	720	288	50	7	17500	0x00445C
PAL	720	288	50	6	15000	0x003A98

Getting Started with the ADV202

PAL	720	288	50	5	12500	0x0030D4
PAL	720	288	50	4	10000	0x002710
PAL	720	288	50	3	7500	0x001D4C
PAL	720	288	50	2	5000	0x001388
PAL	720	288	50	1	2500	0x0009C4

Table 10 RCVal settings for various data output rates for Target Size

RCVal	Description
0x0100	Near mathematically lossless
0x0300	Visually lossless
0x0500	Noticeable artifacts for busy images, visually lossless for soft images
0x0700	Noticeable artifacts but details of image are still present
0x0900	Very noticeable artifacts; used for low bit rate applications

Table 11 RCVal settings for Target Quality

J2KPROG

This value selects one of the 5 progression styles defined in the ISO/IEC15444-1 standard. If J2KPROG is programmed to LRCP then first the layer information, then the resolution, then the component and finally the position will be decoded from the JPEG2000 stream.

LRCP : Layer-Resolution-Component-Position.

RLCP : Resolution-Layer-Component-Position.

RPCL : Resolution-Position-Component-Layer

PCRL : Position-Component-Resolution-Layer

CPRL : Component-Position-Resolution-Layer

PCRL provides optimal latency of less than 1 field. This style is required for tiles with more than 1.2 Msamples [lossy mode] or 0.6 Msamples [lossless mode].

QFACT, STEPSIZES and LOAD_SS:

Initially, a default set of stepsizes will be assigned to each subband. After the Host Boot procedure, the LD_SS parameter is checked, and if it is 0, then these default stepsizes will be written to the STEPSIZES parameters automatically.

The QFACT parameter provides a simple method to scale the quantization factors (inverse stepsizes) by a fraction determined by dividing the QFACT parameter by 256 (if QFACT=0, then the quantization factor is set to 1). If custom stepsizes are not to be used at startup (LD_SS=0 after Initialization), then the QFACT parameter will be applied to the default stepsizes after they are written to the STEPSIZES parameters as mentioned earlier. Future changes to QFACT will require the same procedure as changing the STEPSIZES parameters.

To use custom stepsizes at startup, the following procedure should be used:

- During the Pre-Initialization routine, write the new stepsizes to the STEPSIZES parameters using the standard JPEG2000 step size format.
- Write 0x0001 to the LOAD_SS parameter.

To change the STEPSIZES or QFACT parameters during the encode capture process, the following procedure should be used:

- Poll the LOAD_SS parameter (indirect address 0x00057f36) until it reads as 0x0000.
- Write the new stepsizes to the STEPSIZES parameters using the standard JPEG2000 step size format.
- Write the new quantization factor to the QFACT parameter.
- Write 0x0001 to the LOAD_SS parameter.

Just before the start of the next input video field, the program will load these new stepsizes and/or QFACT, and then reset the LOAD_SS parameter to 0.

If custom stepsizes are to be used at startup, then these parameters must be provided and the LD_SS parameter must be written to 1.

COD_STYLE

COD_STYLE [2:0]:

ADV202 Raw Format is described in the technical note

TechNote_ADV202_Compressed_Data_Format_1.pdf.

JP2 Format is only needed for YCbCr 4:2:2 inputs.

COD_STYLE [7:4] :

PPT = Packed packet headers for a tile.

For more detail refer to the ISO/IEC15444-1 specification, section A.7.5 and table A-2.

PLT = Packet length header for a tile.

For more detail refer to the ISO/IEC15444-1 specification, section A.7.3 and table A-2.

SOP = Start of Packet header.

For more detail refer to the ISO/IEC15444-1 specification, section A.8.1 and table A-2.

EPH = End of Packet header.

For more detail refer to the ISO/IEC15444-1 specification, section A.8.2 and table A-2.

See TechNote – ADV202 Output Format. for more detailed information.

Visual Weighting - LOAD_VW, VW_Y, VW_CB, VW_CR:

Initially all wavelet subbands are weighted only by their contribution to the Mean-Square-Error [MSE] of the decoded image, this is also called the L2Norm of the subband. However, it is possible to reduce image compression artifacts by applying an additional weighting factor to each subband based on that subband's contribution to visual perception.

The weighting factor would normally be a floating-point number between 0 and 1, but can be larger than 1 as well. However, the ADV202 requires a signed twos-complement 16-bit log2 representation of this value with 9,7 precision. A factor that is less than 1 will result in a negative log2 value, while a factor that is larger than 1 will result in a positive log2 value. Table 12 illustrates some examples of this log2 representation.

Upon startup, the LOAD_VW parameter is used to load the appropriate visual weighting table, or to use no weighting if this parameter is set to 0. At any time, the user can either have the firmware load a pre-defined visual weighting table, or load a custom table for the VW_Y, VW_CB, VW_CR parameters. This loading procedure is the same as for the Step size loading:

Poll the LOAD_VW parameter [indirect register address 0x000057f37] until it reads 0x0000.

Write the visual weight to the VW_Y, VW_CB_VW_CR parameters using the format described above.

To load the custom visual weights, write 0x0001 to the LOWAD_VW parameter, or to load the pre-defined visual weight recommended in the ISO/IEC15444-3 standard form Motion JPEG2000, write 0x0002 to the LOAD_VW parameter.

Once the firmware loads the visual weights, LOAD_VW will be reset to '0' again.

Floating Point Weighting Factor	Log2 Representation for VW_Y, VW_CB, VW_CR parameter
1. 20	0x0022
1.0	0x0000
0.75	0xFFCB
0.3	0xFF22
0.1	0xFE57

Table 12 Log2 Representation of Weighting Factor examples

Multi-Layer LTARGET [1:16]

These 16 words [32-bit] contain the desired target for each layer in the code stream and are used with multi-layer Rate Control programmed with RCTYPE.

LTARGET [1] represents a 32-bit word for layer 1, LTARGET [2] represents a 32-bit word for layer 2 etc.

RCTYPE = 3 :

LTARGET represents the target byte length of the layer. This includes bytes in all layers up to and including the specified layer.

RCTYPE = 4 :

LTARGET represents the target quality for each layer.

Code example

Source code example for Standard Definition NTSC/PAL encode/decode application can be found at :

ftp://ftp.analog.com/pub/Digital_Imaging/ADV202_Eval_P160SD_FPGA_SYSTEM/P160SD_system/projects/Spartan3SLC400/video_pipe/edk71/

using the latest revision.

The main function is contained in the <TestApp.c> file.

ADV202 configuration and initialization is contained in the <ADV202.c> file.