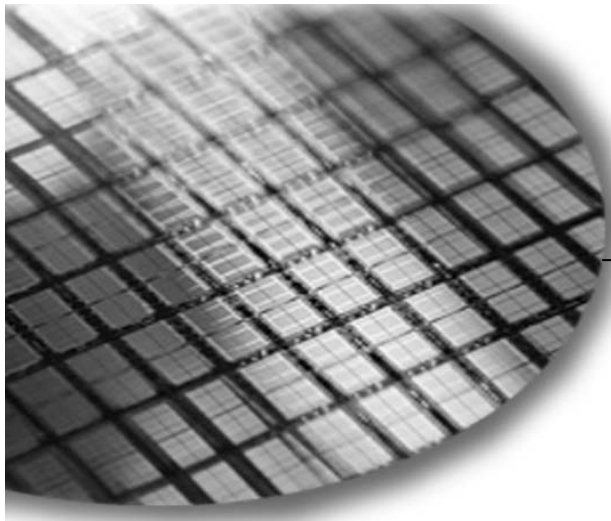
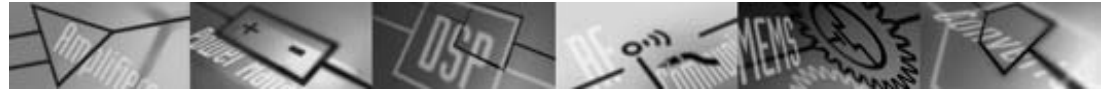




The World Leader in High-Performance Signal Processing Solutions



ADV202 Session 3

Architecture, Interfaces



Host Interface Summary

◆ SRAM Style Parallel interface

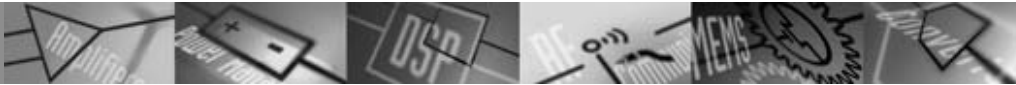
- 16-bit
- 32-bit

◆ DMA Modes

- 2 channels
- DREQ/DACK Burst & Single
- DCS Mode

◆ JDATA Mode

- 8 bit synchronous
- Must use 16-bit host interface



Parallel Interface - Summary

- ◆ **50 Mhz Max Asynchronous**
- ◆ **16/32 Bits width**
- ◆ **Control Signals (9 Total)**
 - **4 Address Pins – Must Use**
 - **CS, WR, RD – Must Use**
 - **ACK – Optional, but good idea**
 - **IRQ – Optional, but good idea**



Parallel Interface - Registers

- ◆ **4 Address bits = 16 DIRECT Registers**
- ◆ **Direct Registers**
 - Part of the host interface block, predictable timing
 - PLL_HI, PLL_LO don't require a clock, all others do
- ◆ **Indirect Registers**
 - Actually are 2 Direct registers that access internal registers
 - Much slower timing, ACK is useful



Parallel Interface – Direct Registers

- **5 Fifos**

- ◆ **PIXEL** – Uncompressed data (Host Mode) (0x0)
- ◆ **CODE** – Compressed data (0x1)
- ◆ **ATTR** – Attribute data (rarely used) (0x2)
- ◆ **ANCL** – Ancillary data (never used) (0x3)
- ◆ **CMDSTA** – Command Stack (never used) (0x4)

- **3 Interrupt Registers**

- ◆ **EIRQIE** – Interrupt enables (0x5)
- ◆ **EIRQFLG** – Interrupt flags (0x6)
- ◆ **SWFLAG** – Application ID, other info from firmware (0x7)



Parallel Interface – Direct Registers

- **2 Bus Control Registers**
 - ◆ BMODE – Sets width and mode of bus (0x8)
 - ◆ MMODE – Sets options for the bus (0x9)
- **3 Indirect Access registers**
 - ◆ STAGE – Used to “stage” upper part of 32-bit values for 16-bit bus (0xA)
 - ◆ IADDR – Internal address for indirect register (0xB)
 - ◆ IDATA – Data for indirect register (0xC)
- **Boot Mode**
 - ◆ BOOT – used to set mode for set up procedure (0xD)
- **2 PLL Registers**
 - ◆ PLL_HI – used to set multiplier for internal ARM vs JCLK
 - ◆ PLL_LO – sets internal clock rate (JCLK) vs. MCLK



PIXEL Register

- The 32 bit PIXEL register is used to access the Pixel data FIFO via normal addressed accesses and generally is used for pixel data input and output on the host interface. The actual bus width when accessing this register depends on the host control data width selected
 - ◆ 16-bit mode will accept/return data on HDATA[15..0]
 - ◆ 32-bit mode will accept/return data on HDATA[31..0]
- Modes
 - ◆ Host – Used to send uncompressed data (Encode) or receive it (Decode)
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Unsued
- DATA Register, HIPI mode only



CODE Register

- The 32 bit CODE register is used to access the JPEG2000 compressed codestream FIFO via normal addressed accesses. The actual bus width when accessing this register depends on the host control data width selected [BUSMODE/HWIDTH].
 - ◆ 16-bit mode will accept/return data on HDATA[15..0]
 - ◆ 32-bit mode will accept/return data on HDATA[31..0]
- Modes
 - ◆ Host – Used to send compressed data (Decode) or receive it (Encode)
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Same as HIPI
- DATA Register, Always used



ATTR Register

- If the attribute encode parameter is non-0, this register is used to send the selected attributes. The actual bus width when accessing this register depends on the host control data width selected [BUSMODE/HWIDTH].
 - ◆ 16-bit mode will accept/return data on HDATA[15..0]
 - ◆ 32-bit mode will accept/return data on HDATA[31..0]
- Modes
 - ◆ Host mode – Used to receive attribute data (Encode)
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Same as HIPI
- DATA Register, Encode only, ATTRType > 0



ANCL Register

- This register is unused.
- Modes
 - ◆ Host mode – Unused
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Unused
- Unused



CMDSTA Register

- This register is unused. It was intended for host interface command parsing but was never implemented.
- Modes
 - ◆ Host mode – Unused
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Unused
- Unused



EIRQIE Register

- This register sets the mask by which internal interrupts are asserted on the IRQ pin.
- Setup Register, Always used

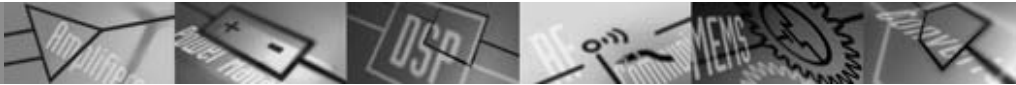
Bit	Name	Description
0	PFTH	Threshold interrupt for pixel data, used for direct reads (unusual)
1	DFTH	Threshold interrupt for code data, used for direct reads (unusual)
2	AFTH	Threshold interrupt for attribute data, used for direct reads (unusual)
3	NFTH	Threshold interrupt for ancillary data, used for direct reads (never)
4	PFERR	Read empty / Write Full error for pixel data (unusual)
5	DFERR	Read empty / Write Full error for code data (unusual)
6	AFERR	Read empty / Write Full error for attribute data (unusual)
7	NFERR	Read empty / Write Full error for ancillary data (unusual)
8	Unused	
9	Unused	
10	SWIRQ0	Always used during startup process, signals an error after startup
11	SWIRQ1	Used only for synching multiple chips, signals chip is ready to output
12	SWIRQ2	Unused, may be used for error reportinh
13	INDERR	Indirect register access error (unusual to use)
14	IHWDIRQ	Internal interrupt, ignore this, it's meaningless
15	FERR	Fatal error, can be ignored since it's unused



EIRQFLG Register

- This register indicates which interrupts are currently asserted, mirrors EIRQIE.
- Setup Register, Always used

Bit	Name	Description
0	PFTH	Threshold interrupt for pixel data, used for direct reads (unusual)
1	DFTH	Threshold interrupt for code data, used for direct reads (unusual)
2	AFTH	Threshold interrupt for attribute data, used for direct reads (unusual)
3	NFTH	Threshold interrupt for ancillary data, used for direct reads (never)
4	PFERR	Read empty / Write Full error for pixel data (unusual)
5	DFERR	Read empty / Write Full error for code data (unusual)
6	AFERR	Read empty / Write Full error for attribute data (unusual)
7	NFERR	Read empty / Write Full error for ancillary data (unusual)
8	Unused	
9	Unused	
10	SWIRQ0	Always used during startup process, signals an error after startup
11	SWIRQ1	Used only for synching multiple chips, signals chip is ready to output
12	SWIRQ2	Unused, may be used for error reportinh
13	INDERR	Indirect register access error (unusual to use)
14	IHWDIRQ	Internal interrupt, ignore this, it's meaningless
15	FERR	Fatal error, can be ignored since it's unused



SWFLAG Register

- This register shows the application ID on startup, and error codes later. It is a good indication that firmware has started up correctly. It's basically read only.
- Modes
 - ◆ Host mode – Used to read out software codes
 - ◆ Video modes (EAV/SAV, HVF, RAW) – Same as host mode
- DATA register, Always checked



BUSMODE Register

- This register sets the functions of the multi-use host bus pins
- Setup Register, Always used

Bit	Name	Description
0 .. 1	HWIDTH	01 = 16 bit bus, 10 = 32 bit bus 00/11 are invalid
2 .. 3	DWIDTH	DMA Width : 00 = 8 bit, 01 = 16 bit, 10 = 32 bit. Not sure why anyone would use a different DWIDTH than HWIDTH but you can.
4 .. 6	BCFG	Special host modes: 000 - Normal host mode, all pins are part of the host bus or unused 001 – JDATA mode (synchronous 8 bit interface). Bus width must be 16 010 – Dual lane. Allows dual lane luma/chroma at have the vclk. Rarely/never used All other bit combinations are invalid
7 .. 15	Unused	Write to 0



MMODE Register

- This register configures indirect register accesses
- Setup Register, Always used

Bit	Name	Description
0 .. 1	IWIDTH	01 = 16 bit access, 10 = 32 bit access 00/11 are invalid 16 bit access requires use of the STAGE register
2 .. 3	IAUTOSTP	Address auto-increment 00 – 8 bit, 01 – 16 bit, 10 – 32 bit, 11 – Disable Should either be disabled or set to IWIDTH, not sure why anyone would want it different.
4	IAUTOMOD	Controls if auto-increment is up or down 0 – Increment 1 - Decrement
5	CTLREGAM	Controls address space for indirect registers 0 – Full address space (0xFFFFXXXX) 1 – Only bottom 16 bits count, 0xFFFF is implied. Good for 16-bit host mode
6 .. 15	Unused	Set to 0



STAGE Register

- The 16 bit STAGE register is always used when using a 16 bit host. It is not required for a 32 bit host. It is used to access 32 bit words in indirect memory, indirect registers, and 32 bit direct registers. The STAGE register acts as a holding or “staging” register. For example, when using a 16 bit host the STAGE register is used in conjunction with the IADDR register to configure the 32 bit address or the IDATA register.
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



IADDR Register

- The 32 bit IADDR register is used to set the address for indirect register and indirect memory read/write accesses. The indirect address may optionally be auto-incremented/decremented by setting the IAUTOMOD and IAUTOSTP fields in the MMODE register.
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



IDATA Register

- The 32 bit IDATA direct register is used to read and write data to an indirect register or indirect memory. A write to this register will write a value to the location that is being pointed at by the IADDR register. The indirect address may optionally be auto-incremented by setting the IAUTOMOD and IAUTOSTP fields in the MMODE register.
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



BOOT Register

- The bits in the BOOT register are used to select various boot modes of the ADV202 after reset and to load specific instruction sets into memory. The boot mode can be configured via hardware [over the CFG pins] or via software [via firmware]. In a hardware configuration after a hard reset, the boot mode will be set to the values on the configuration pins CFG[2..1]. These bits are not reset on a soft reset.
- Bits 0 .. 2 Bootmode
 - ◆ 010 – Used to load firmware as part of start up procedure
 - ◆ 101 – Used to start to execute firmware
 - ◆ No other combination is used
- Bit 3 – Always write a 1
- Bits 4 .. 5 – Always write a 0
- Bit 6 – Hard reset, identical to asserting reset pin
- Bit 7 – Soft reset, like hard reset but BOOT and PLL settings remain
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



PLL_HI Register

- The 16 bit PLL_HI register is used to configure the on chip PLL. Internally, the ADV202 uses two clock domains which are generated by an on chip Phase Locked Loop from the input clock MCLK.
- This register sets the CPU Clock to either
 - ◆ 0x8 : ARM CLK = JCLK / 2
 - ◆ 0x0 : ARM CLK = JCLK
- This register doesn't require a clock.
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



PLL_LO Register

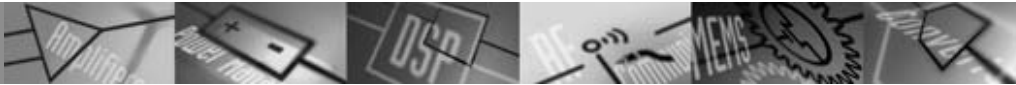
- The 16 bit PLL_LO Register sets the relationship between JCLK and VCLK.
- Bits 0 .. 4 – Multiplier of VCLK
- Bits 5 .. 6 – Always write a 0
- Bits 7 – A 1 here will do a divide by 2 after the multiplier
- This register doesn't require a clock.
- Modes
 - ◆ There is no modality for this register, it always behaves the same
- SETUP register, Always used.



EDMODE0 & EDMODE1 Registers

- These INDIRECT registers set up DMA
- Setup Register, Always used

Bit	Name	Description
0	DMEN	0 – Disable DMA, 1 – Enable DMA
1 .. 2	DMSEL0	Fifo select 00 – Pixel, 01 – Code, 10 – Attribute, 11 - Ancillary
3 .. 5	DMMOD0	DMA Channel 0 mode 000 - Dedicated Chip Select DMA mode 001 - Single transfer DREQ/DACK DMA mode 010 - Burst transfer DREQ/DACK DMA mode 011 - JDATA mode 100 - reserved 101 - Single transfer Fly-by DMA mode 110 - Burst transfer Fly-by DMA mode 111 - reserved
6 .. 8	DMBL0	DMA Burst Length in words of BUSWIDTH 000 – 8, 001 – 16, 010 – 32, 011 – 64, 100 – 128
9	DR0POL	Polarity of request signal. 0 – Active Low, 1 – Active High
10	DA0POL	Polarity of acknowledge signal. 0 – Active Low, 1 – Active High
11 .. 14	DR0PULS	DREQ pulse length in JCLKS, if 0 then it will stay active until DACK is asserted



Video Interface Summary

◆ Embedded Syncs mode

- EAV/SAV
- Single or multiple components

◆ HVF Mode

- Standard H Sync, V Sync, Field

◆ RAW Pixel Mode

- Simple handshaking mode
- No Blanking

◆ Host mode

- Pixel data comes in on host interface



PMODE1 Register

- This INDIRECT register sets up the video packing and precision
- Setup Register, Custom mode only

Bit	Name	Description
0 .. 4	PFMT	This sets the video format. Important modes are: 0x4 – Video interface, Single Component 0x5 – Video Interface, CbYCrY 0x7 – Video Interface, CbCr 0x14 – 0x1B – Host interface, many packing options
5 .. 7	Reserved	Always write 0
8 .. 10	Precision	Sample Precision: 0x0 8-bit precision 0x1 10-bit precision 0x2 12-bit precision 0x3 14-bit precision 0x4 16-bit precision
11 .. 15	Reserved	Always Write 0



PMODE2 Register

- This INDIRECT register sets the interface polarity and sample range
- Setup Register, Custom mode only

Bit	Name	Description
0	VCLK_POL	VCLK Active Edge 0 – Falling Edge, 1 – Rising Edge
1	VSYNC_POL	VSYNC Active Edge 0 – Falling Edge, 1 – Rising Edge
2	HSYNC_POL	HSYNC Active Edge 0 – Falling Edge, 1 – Rising Edge
3	FIELD_POL	FIELD Active Edge 0 – Falling Edge, 1 – Rising Edge
4	YUNI	Range of Luma Samples 1 – Unipolar (0 to 255 for 8 bit), 0 –Bipolar (-127 to 127 for 8-bit) This is nearly always a 1
5	CUNI	Range of Chroma Samples 1 – Unipolar (0 to 255 for 8 bit), 0 –Bipolar (-127 to 127 for 8-bit) This is nearly always a 1
6 .. 15	Reserved	Always write a 0



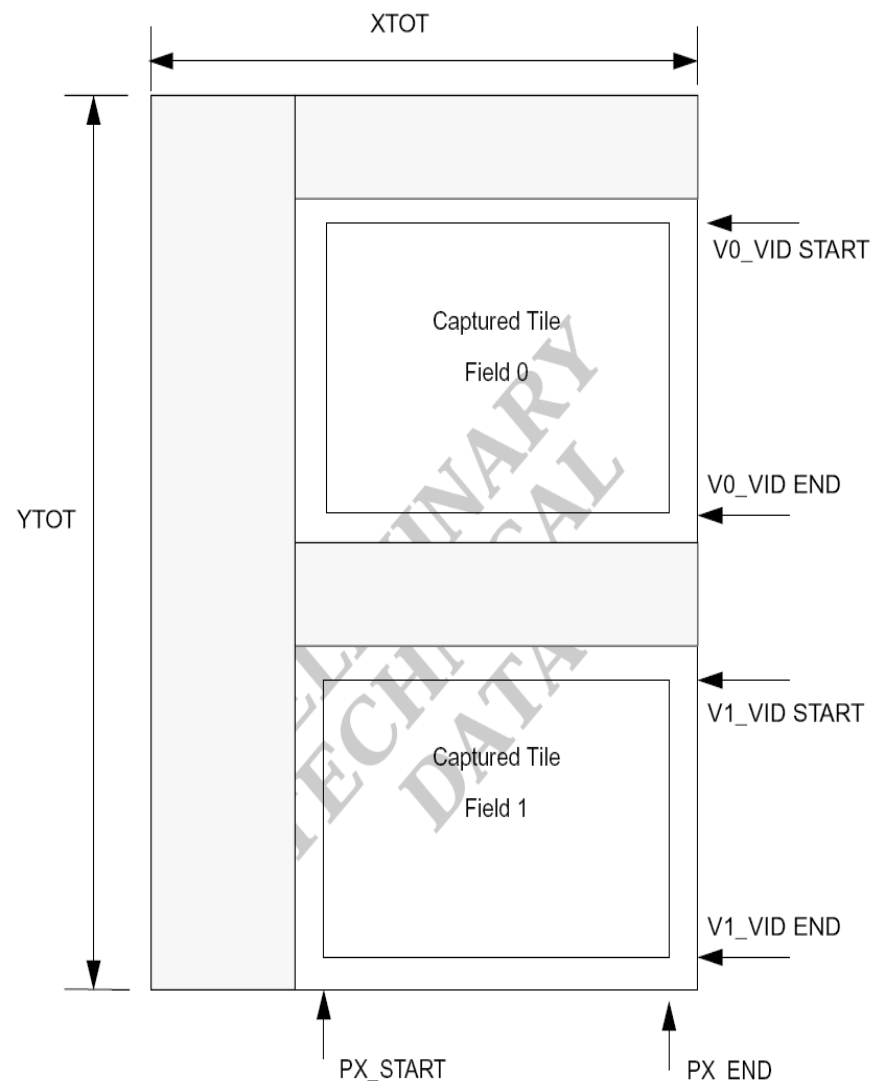
VMODE Register

- This INDIRECT register sets up the video mode
- Setup Register, Custom mode only

Bit	Name	Description
0	MAS_SLV	Master or Slave mode: 1 – Master, generate clocks in decode 0 – Slave, accept clocks in decode, always slave in encode
1	ENC_DEC	1 – Encode, 0 - Decode
2	MP_656	Input timing, 0 – EAV/SAV 1 – HVF Mode
3	DUAL_LANE	Dual lane mode. 1 – dual lane mode on, 0 – dual lane mode off (normal)
4	HOST_MODE	1 – Host mode on, 0 – Video mode on
5	RAW_MODE	1 – RAW Pixel mode on, 0 – RAW Pixel mode off
6	Reserved	Always write a 0
7	PGRSV_SCN	1 – non-interlaced source, 0 – interlaced source
8 .. 15	Reserved	Always write a 0

Video Interface Summary

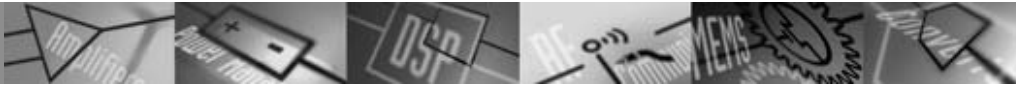
- ◆ This graphic shows the relationship between the video registers and the incoming image.
- ◆ These parameters **ONLY** need to be set in **CUSTOM** mode.





XTOT Register

- This register sets the total number of horizontal samples, including blanking. Note that this is samples, not pixels. For a 4:2:2 image 640 pixels wide, this would be 1280. For a single component image it would only be 640
- Modes
 - ◆ This register is used in all video modes, including host mode
- SETUP register, Used in CUSTOM mode only



YTOT Register

- This register sets the total number of lines, including blanking.
- Modes
 - ◆ This register is used in all video modes, including host mode
- SETUP register, Used in CUSTOM mode only



F0_START Register

- Start of Field 0. Only used in Decode mode to identify at which line number the Field bit will transition from Field 1 to Field 0.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode, interlaced video
- SETUP register, Used in CUSTOM mode only, DECODE



F1_START Register

- Start of field 1. Only used in Decode mode to identify at which line number the Field bit will transition from Field 0 to Field 1.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode, interlaced video
- SETUP register, Used in CUSTOM mode only, DECODE



V0_START Register

- The 16 bit V0_START register is used to set the first active video line in field 0 / the frame that will be captured by the ADV202 in Encode. It specifies the first active line in field 0 / the frame to output in Decode. Line count starts with line 1.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode
- SETUP register, Used in CUSTOM mode only



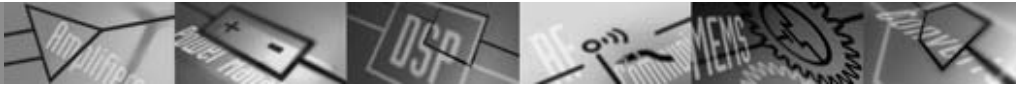
V1_START Register

- The 16 bit V1_START register is used to set the first active video line in field 1 that will be captured by the ADV202 in Encode. It specifies the first active line to output in Decode for field 1. Line count starts with line 1.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode, interlaced video
- SETUP register, Used in CUSTOM mode only



V1_END Register

- The 16 bit V1_END register is used to set the vertical end of the active video in field . This register should hold the value of the last active line number. Line count starts with line 1.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode, interlaced video
- SETUP register, Used in CUSTOM mode only



V0_END Register

- The 16 bit V0_END register is used to set the vertical end of the active video in field 0 / the frame. This register should hold the value of the last active line number. Line count starts with line 1.
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode
- SETUP register, Used in CUSTOM mode only



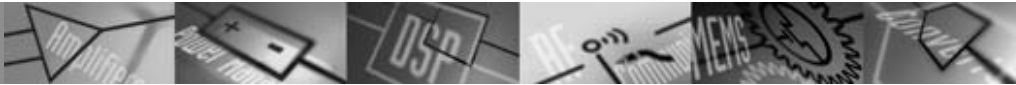
PIXEL_START Register

- The 16 bit PIXEL_START register is used to set the horizontal start (i.e. first active sample in the line) of the active video in units of samples (horizontal counter starts at sample 1).
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode
- SETUP register, Used in CUSTOM mode only



PIXEL_END Register

- The 16 bit PIXEL_END register is used to set the horizontal end (i.e. last active sample in the line) of the active video in units of samples (horizontal counter starts at sample 1).
- Modes
 - ◆ This register is only used in HVF or EAV/SAV mode
- SETUP register, Used in CUSTOM mode only



Host Mode

- ◆ Pixel data comes in on host interface (PIXEL Fifo)
- ◆ Compressed data goes out on host interface (CODE Fifo)
- ◆ Usually 1 DMA channel is used for each
- ◆ There is no blanking in host mode, only active pixels
- ◆ Only XTOT and YTOT need to be programmed, aside from setup registers



RAW Pixel Mode

◆ Simple Handshaking protocol

- VFRM (VSYNC) I/O is used to signal the beginning of a new frame. In Encode, it's an input and in Decode it's an output
 - VRDY (HSYNC) O is used to signal the outside that the ADV202 is ready to accept / output data. Any data clocked in when this isn't asserted will be ignored (as will a VFRM strobe)
 - VSTRB (FIELD) I is used from the outside to signal the ADV202 that the outside system is ready to accept / output data. If this is not asserted, then any data clocked in (or VFRM strobes) will be ignored.
- ◆ RAW mode uses no blanking, only XTOT and YTOT need to be programmed aside from setup registers.