

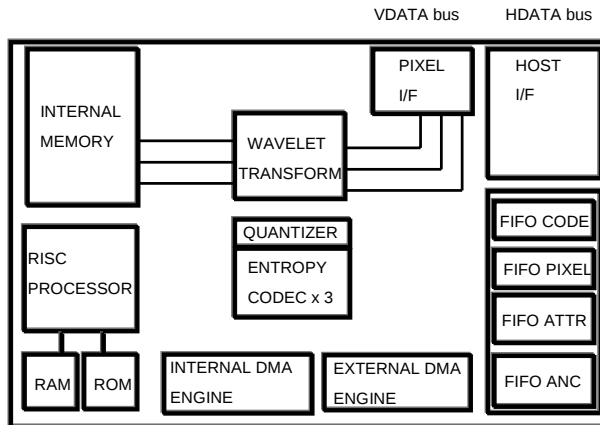
The World Leader in High-Performance Signal Processing Solutions

ADV202 HIPI mode

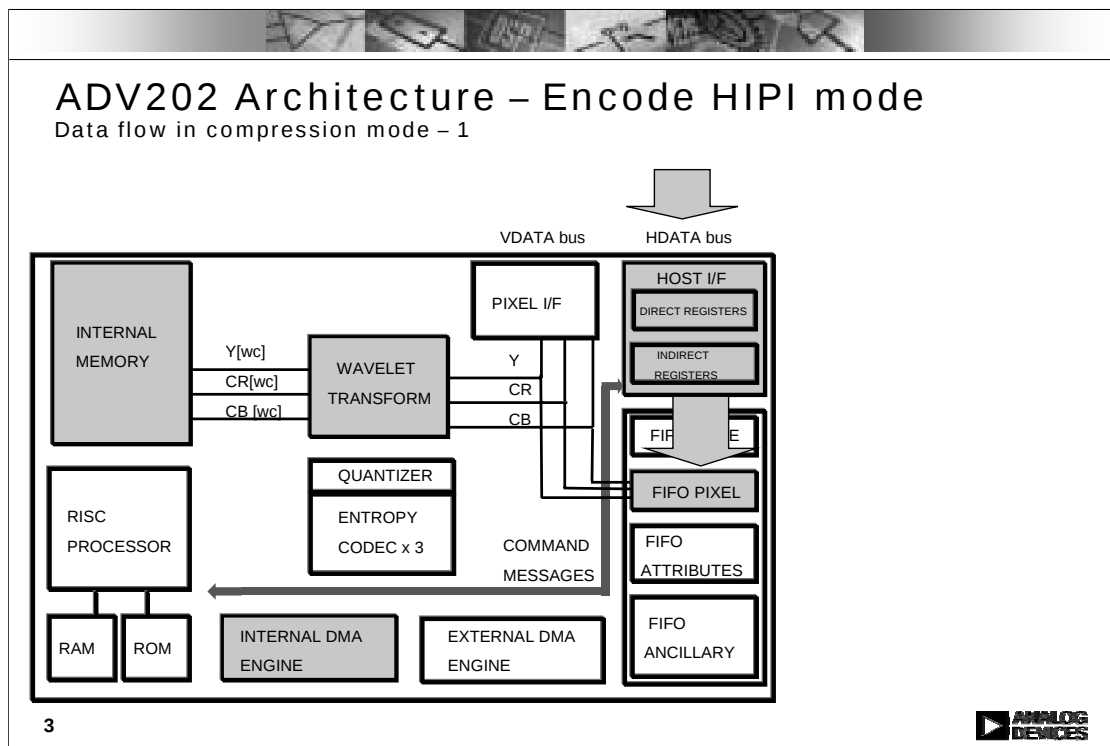
christine.bako@analog.com



ADV202 Architecture – Block diagram and functional description



- Risc processor
 - The embedded 32-bit RISC processor serves as a system controller, receiving/sending commands over the host interface and is also used for parsing and the generation of the JPEG2000 code-stream. The RISC processor includes its own ROM and RAM for both program and data memory. The processor sets up internal registers and configures functional blocks to user-defined functions
- Wavelet transform engine
 - 5/3 and 9/7 wavelet filters giving
 - up to 6 transform levels.
- 3 Entropy Codecs
 - to generate JPEG2000 code-stream: using quantization, Rate Distortion Optimization and context modeling, organize data into packets and layers
 - throughput of up to 65 Msamples/s.
- Host Interface
 - can be configured for 32-bit or 16-bit control and 8-, 16- or 32-bit data.
 - data transfers between FIFOs
 - control of 2 external DMA channels
 - access to indirect and direct register
- Pixel Interface
 - process video or pixel data transfers from/to VDATA or HDATA bus
- Internal DMA engine
 - Facilitate fast internal memory-to-memory data transfers.

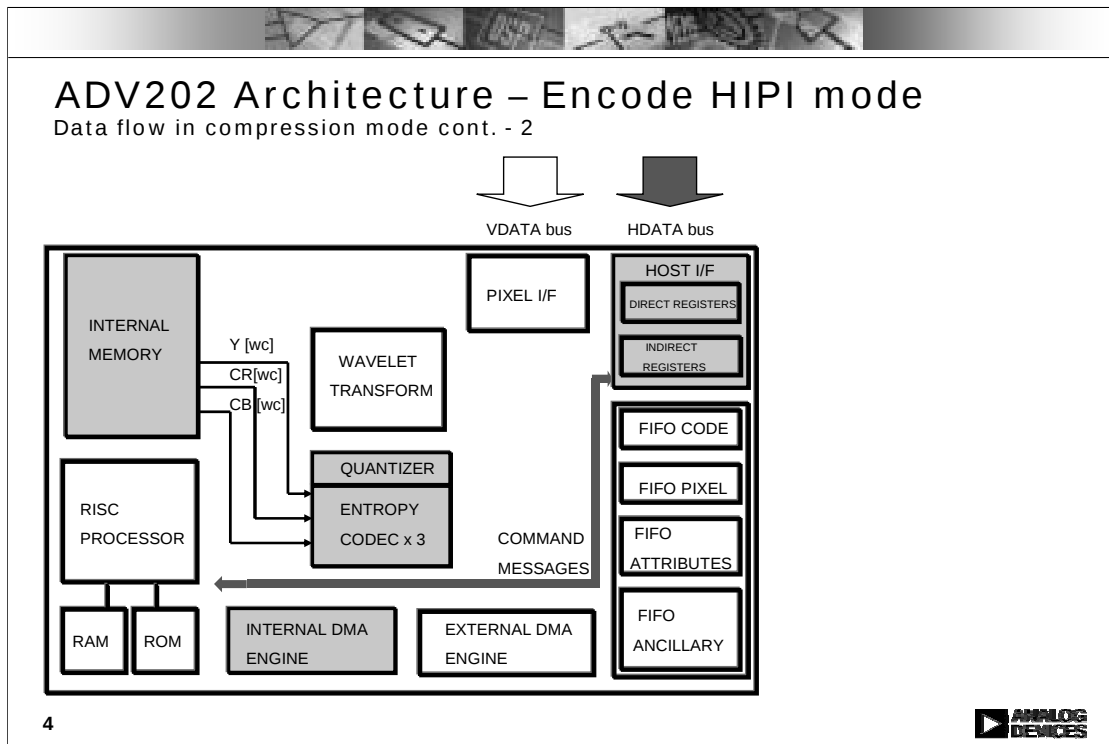


Video or pixel data can be input over the VDATA bus or alternatively pixel data can be input over the HDATA bus [HIPI mode]. In either case the video/pixel data is passed to the Pixel Interface [not shown here].

The data is then de-interleaved and passed to the wavelet transform engine. The input data, organized in tiles or frames, is then decomposed into subbands using 5/3 or 9/7 wavelet filters. The WT can perform up to 6 wavelet decomposition levels on a tile/frame. The resultant wavelet coefficients are then written to internal memory.

The ADV202 does not have any field buffers in order to perform compression. The wavelet coefficients are computed on a line-by-line basis and then the entire wavelet coefficients are stored in internal memory.

Throughout operation high bandwidth memory to memory or memory to functional block transfers are handled by the internal DMA engine. The internal DMA engine and bus are connected to every functional block in the ADV202 [not shown here for clarity purpose].



The wavelet coefficients are passed to 3 entropy codecs.

Wavelet coefficients are arranged into units of code-blocks, a code-block being essentially a bit stream of data.

Processing is done on a code-block by code-block basis.

The actual compression is achieved using:

1. Quantization: the code-block bit stream is truncated [lossy compression] or not truncated [lossless]

to reduce the amount of data.

2. Rate Distortion Optimization:

The amount of quantization depends on the output rate requirements or output quality requirements.

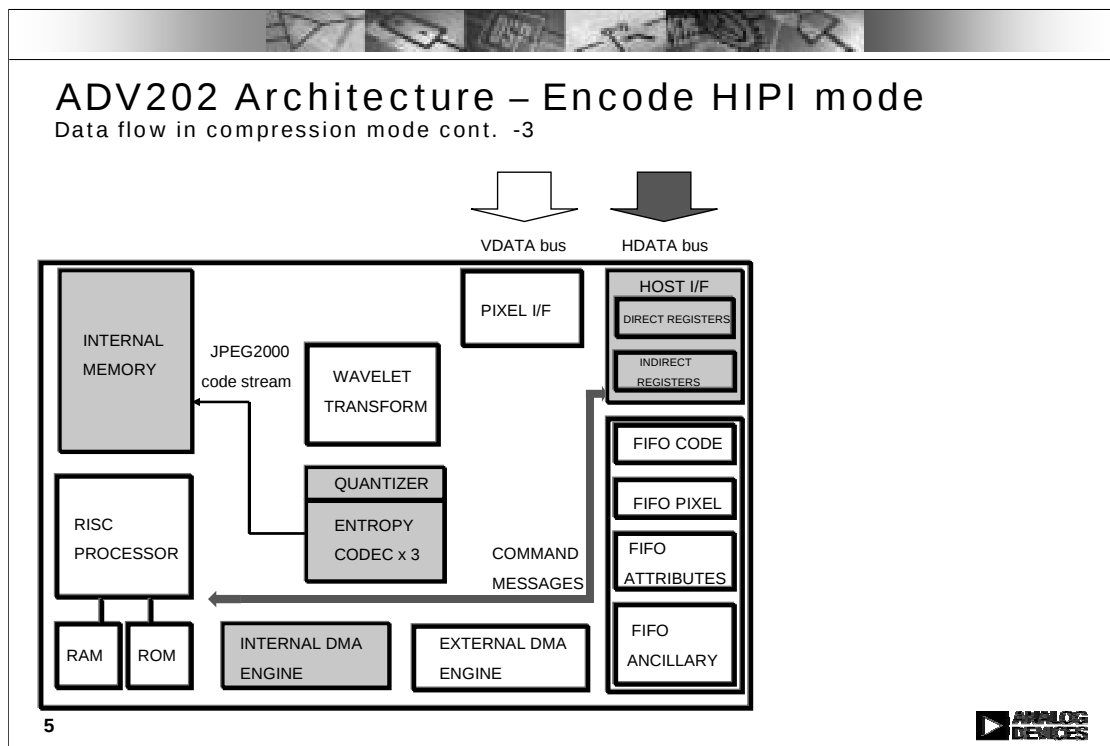
The ECs will perform distortion metric calculations to find the optimal rate/distortion performance.

3. Context Modeling:

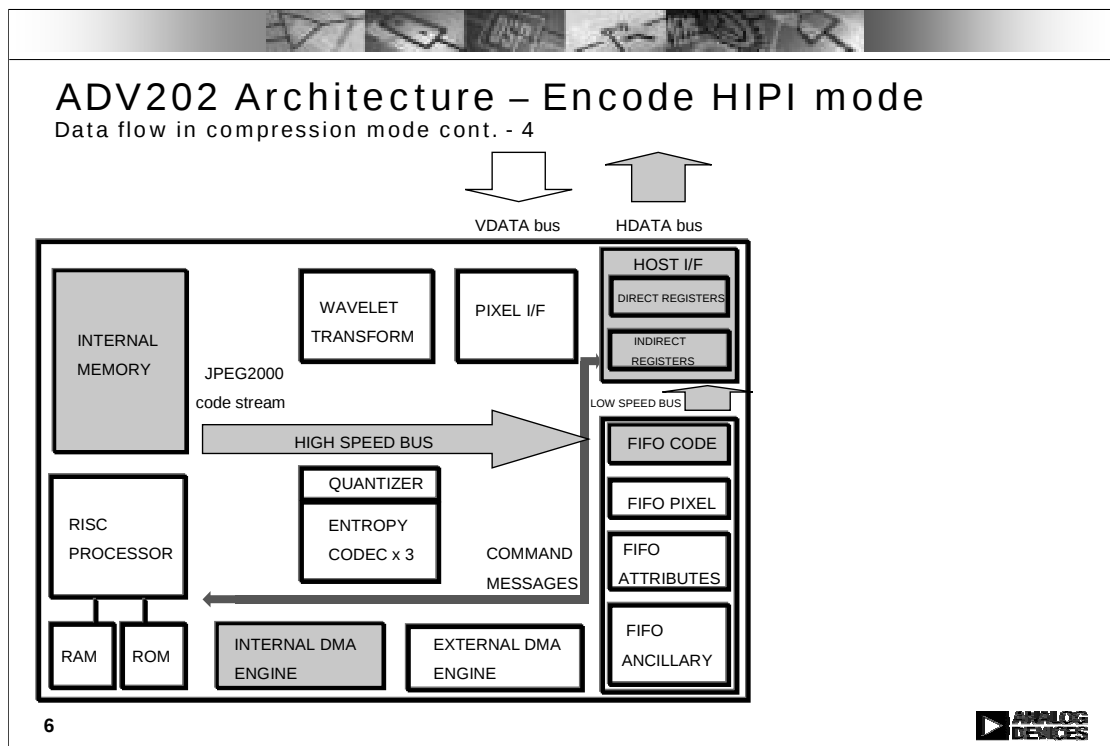
Is a process which assigns information about the significance of each individual coefficient. This allows

data to be arranged according to its significance.

After processing the data is arranged into packets and layers defining the JPEG2000 code stream.



The output of the entropy codecs is passed back to internal memory.



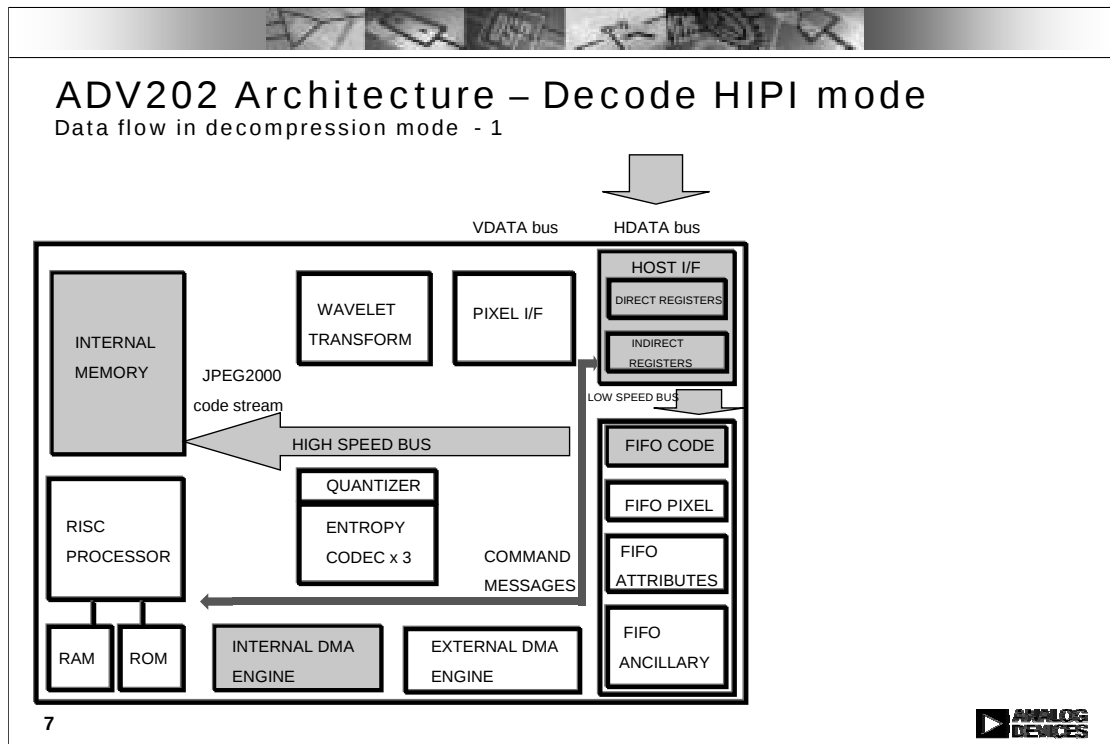
The JPEG2000 code-stream is send over a high speed bus to the Code FIFO in order to buffer internal high speed bus and low speed host interface.

The data can then be output over the host interface using a common read/write access protocol [CS/, RD/, WR/, ACK/, ADDR] or over the external DMA engine in conjunction with an external DMA controller using a DREQ/DACK protocol.

The FIFOs have no other function as to buffer internal high speed bus and the low speed host interface.

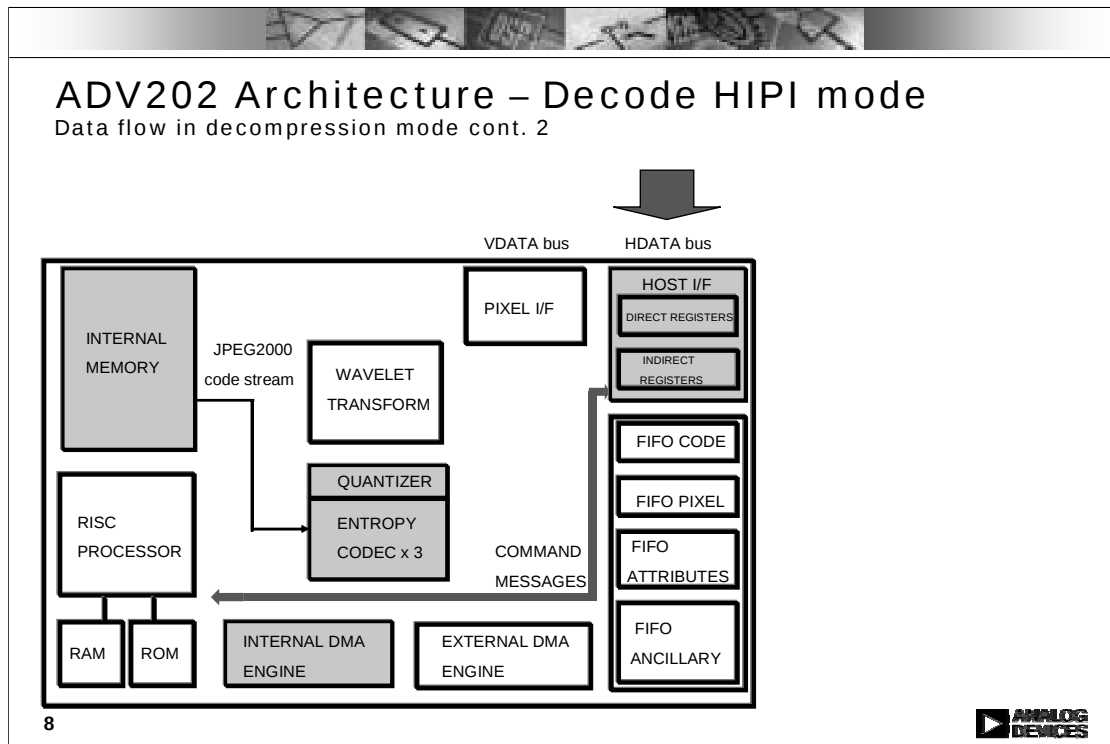
The CODE FIFO is used for compressed data,

As an option, the ATTR FIFO is used for attribute data such as distortion metrics, distortion slopes or byte length, ANCL FIFOs can be used for any other miscellaneous data.

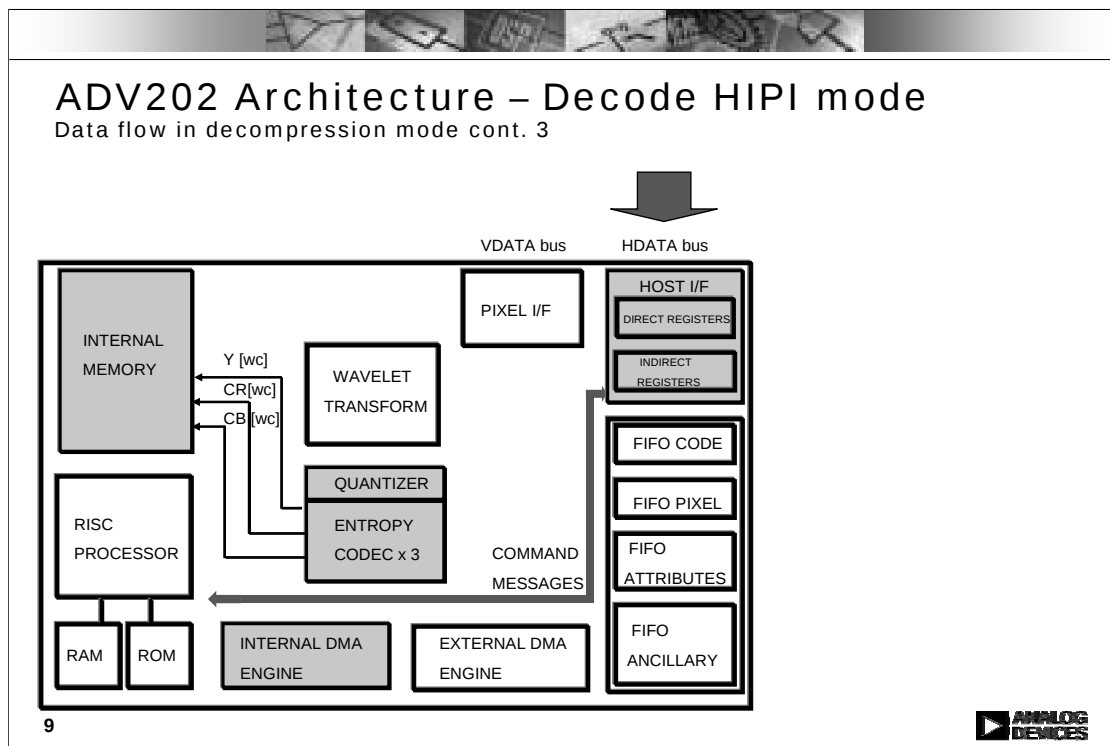


Compressed data is input over the host interface using the same protocols as in Encode mode and passed on to the CODE FIFO from which it is sent to the internal memory.

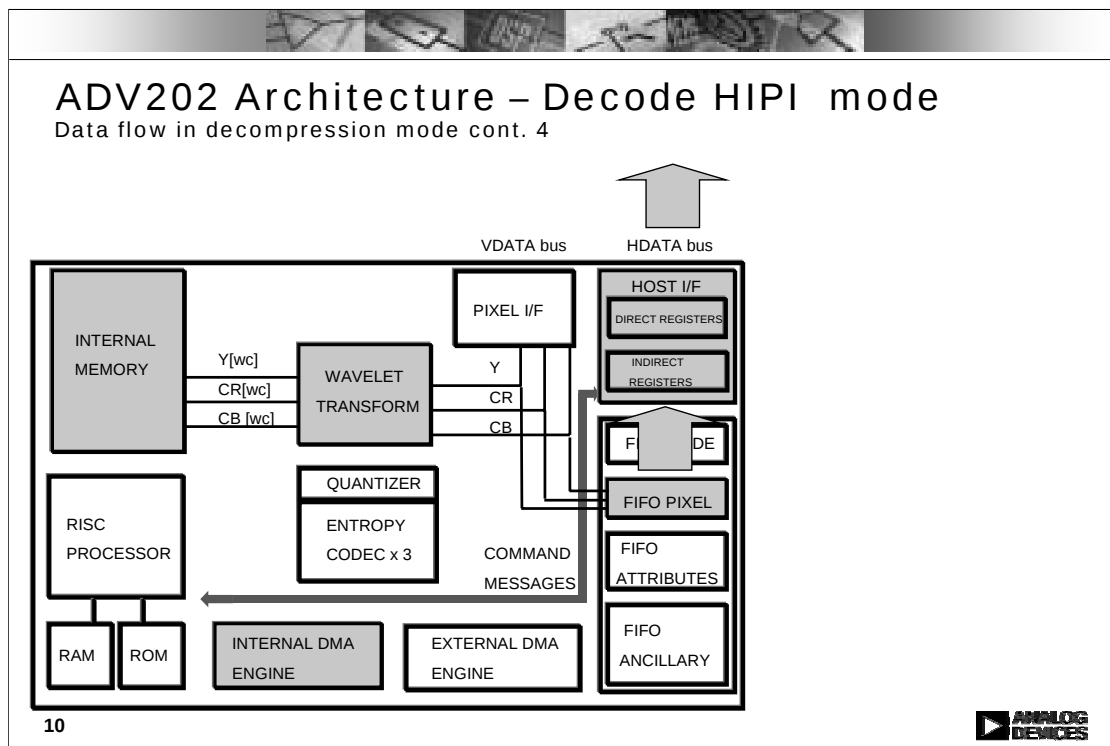
500KB are available for internal storage [uncompressed NTSC field = 345KB].



The JPEG2000 code-stream is passed to the entropy codecs which decompose the code-stream back into wavelet coefficients.



The wavelet coefficients are passed back into internal memory.



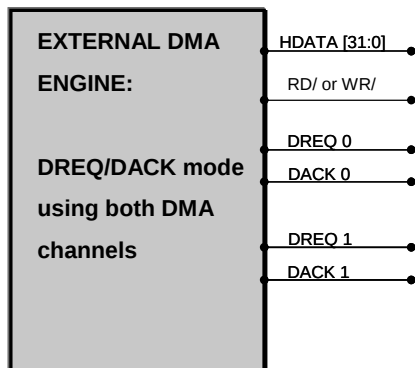
The wavelet coefficients are read from internal memory and are recomposed into uncompressed component samples and are passed to the pixel interface [not shown here] where the components are interleaved into a video stream and output over the VDATA or HDATA [HIPI mode] bus.

In decode, the ADV202 works 'ahead':

It waits until 1+ field is present before starting to decode and to output data.

ADV202 Interfaces –

External DMA Engine – DMA modes



◆ There are 3 DMA modes:

- Dedicated Chip Select
- DREQ/DACK DMA mode
- Fly-By DMA mode

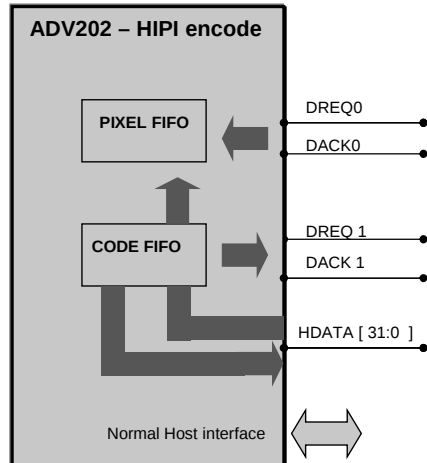
- DMA modes are used to read or write video/pixel data from/ to the ADV202.

◆ The ADV202 has 2 DMA channels

- single DMA channel applications use a configuration similar to JDATA mode, where uncompressed data is input over VDATA while compressed data is output over one of the DMA channels [encode mode].
- dual DMA channel applications input pixel data over DMA channel 0 while DMA channel 1 is used to output compressed data [encode mode]. This mode is called HIPI mode.
- For more information on HIPI mode refer to the <ADV202_HIPI_mode.pdf> application note.

Video and Still Image Input formats

- Still Images



- For still image data in YCbCr 4:2:2 or single component format, the VDATA bus can be used in <Raw Video> mode as the interface to in/output pixel data.
- In this mode, the region to be captured by the ADV202 are defined by the Hsync and Vsync inputs.
[Refer to <Getting started with the ADV202> app note for more information on this mode described in the <custom specific> chapter].
- More commonly the HDATA bus is used for still image applications to input [encode] or output [decode] pixel data. This mode is called <HIPI mode>.
- When using this mode still image data has to be input in raw format, i.e. raw pixel data without timing information.
- The region of the image to be captured is defined by internal registers and not by incoming Hsync or Vsync signals.
- The HDATA bus is shared between uncompressed data input and compressed data output. Access is controlled by DREQ0/DACK0 and DREQ1/DACK1 signals.
- HIPI mode is explained in more detail in the <ADV202_HIPI mode.pdf> application note.

ADV202 References

- ♦ ftp://ftp.analog.com/pub/Digital_Imaging contains
 - ADV202 datasheet
 - Application Notes
 - Application Software
 - Video samples recorded with the ADV202 evaluation board
 - ADV202 evaluation board documentation
 - Errata sheets
- ♦ ADV202 updates are sent via a mailing list.
To be added to this list please contact:
christine.bako@analog.com
- ♦ For application questions please contact:
 - christine.bako@analog.com
- ♦ For release dates and price information please contact:
 - yuzo.shida@analog.com
- ♦ The JPEG2000 standard [ISO/IEC 15444-1] can be downloaded from:
<http://www.jpeg.org/jpeg2000/CDs15444.html>
The final published document can be obtained from:
<http://www.iso.ch>
- ♦ Other useful information:
<http://www.jpeg.org>

