Chapter 25 Video encoder & decoder Unit (VCODEC)

25.1 Overview

VCODEC is composed of video decoder and video encoder. VCODEC is connected to VCODEC_AHB bus through an AHB slave and VCODEC_AXI through an AXI master. The register setting is configured through the AHB slave interface and the stream data is read and written through the AXI master interface.

Video decoder and video encoder share many internal memories and they also share the bus master and slave interfaces. So it prevents video decoder and video encoder from working simultaneously. Encoding and decoding now have to time-share the memory resource on a frame by frame basis.

Video decoder and encoder share the MMU (memory management unit) and AXI bus. Video decoder can support cacheable bus operation.

25.2 Block Diagram

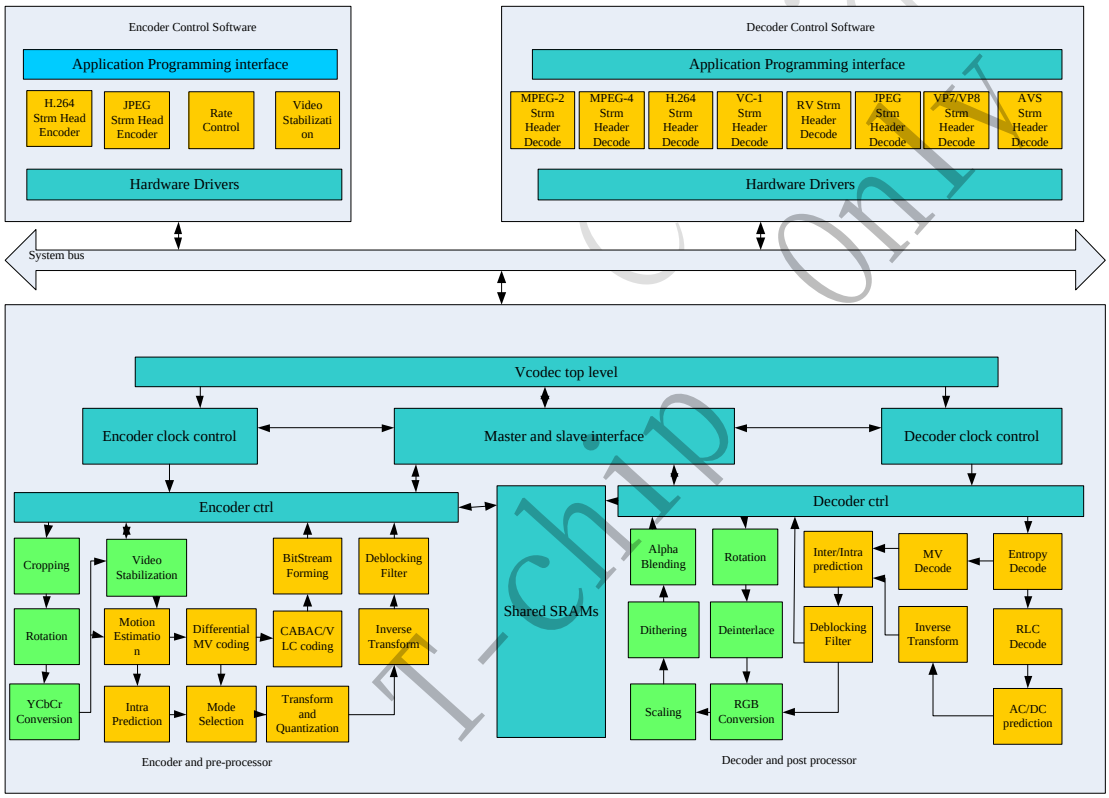

Fig. 25-1 VCODEC block diagram

The block diagram of VCODEC is shown as Fig. 25-1. The lower big box is the VCODEC hardware (HW) implementation. The left part of the lower big box is encoder and pre-processor. The green boxes in it present pre-processor part and video stabilization part while the yellow boxes in it present encoder part. Pre-processor is pipelined with the encoder and it can only be used with the encoder. Video stabilization detects and compensates undesired jitter effect on the video while the desired effects like panning are maintained. Video stabilization can be used pipelined with video encoding or in standalone mode when video encoding is disabled. Video stabilization can detect scene changes in the video sequence. Key frames are encoded when a scene change is detected. This will help improve the encoding

quality.

The right part of the lower big box is decoder and post-processor. The green boxes in it present post-processor while the yellow boxes in it present decoder. Post-processor can run in stand-alone mode or pipeline mode. In stand-alone mode, it can process image data from any external source. In pipeline mode, it can reduce bus bandwidth, because post-process can read its input data directly from the decoder output without accessing external memory.

25.3 Function Description

Decoder support multi-format stream decoding, as Table 25-1 shows. The decoder has a big embedded reference buffer, which can enhance the performance.

Table 25-1 Decoder supported standards, profiles and levels

Standard	Decoder support
H.264 profile and level	Baseline Profile, level 1-5.2 Main Profile, level 1-5.2 High Profile, level 1-5.2 Image size up to 2160p at level 5.2
SVC profile and level	Scalable Baseline Profile, base layer only Scalable High Profile, base layer only
MVC profile and level	Stereo High
MPEG-4 visual profile and level	Simple Profile, levels 0-6 Advanced Simple Profile, levels 0-5
MPEG-2 profile and level	Main Profile, low, medium and high levels
MPEG-1 profile and level	Main Profile, low, medium and high levels
H.263 profile and level	Profile 0, levels 10-70. Image size up to 720x576
Sorenson Spark profile and level	Bitstream version 0 and 1
VC-1 profile and level	Simple Profile, low, medium and high levels Main Profile, low, medium and high levels Advanced Profile, levels 0-3
JPEG profile and level	Baseline interleaved JPEG supports ROI (region of image) decode
RV profile and level	RV8 RV9 RV10
VP7 profile and level	VP7 version 0-3
VP8 profile and level	VP8 version2 (webM)
WebP profile and level	WebP
AVS profile and level	P2 Jizhun Profile, level 6.0 and 6.2
DivX profile and level	DivX Home Theater Profile Qualified DivX3 DivX4 DivX5 Divx6

The post-processer in stand-alone mode supports rotation, Deinterlace, RGB conversation, Scaling, Dithering and Alpha blending, it supports only RGB conversation, Scaling, Dithering and Alpha blending in pipeline mode.

Encoder supports H.264 and JPEG encoding, as Table 25-2 shows.

Table 25-2 Encoder supported standard, profile and level

Standard	Encoder support

H.264 Profile and level	Baseline Profile, levels 1-4.1 Main Profile, levels 1-4.1 High Profile, levels 1-4.1
JPEG profile and level	Baseline(DCT sequential)

The pre-processor supports cropping, rotation and YCbCr conversion.

As the decoder and encoder hardware share SRAM, they can't work simultaneously, but they can time-share the SRAM on a frame by frame basic. The decoder and encoder are both multi-instance capable. As the decoder and hardware has no information of previously encoded or decoded frames stores in internal memory or registers, the input data can be changed each time a frame is encoded or decoded. The format and resolution can be totally different from the previous one.

25.4 Video frame format

This chapter describes different input and output video frame formats supported by VCODEC. Each function module has its own supported video frame formats, and this chapter describes all the video frame formats.

25.4.1 YCbCr 4:2:0 Planar Format

In the planar format, each video sample component forms one memory plane. Within one plane, the data has to be stored linearly and contiguously in the memory as shown in Fig. 25-2. The luminance samples are stored in raster-scan order (Y_0Y_1 Y_2Y_3 Y_4….). The chrominance samples are stored in two planes also in raster scan order (Cb_0Cb_1 Cb_2Cb_3 Cb_4…. and Cr_0Cr_1 Cr_2Cr_3 Cr_4….). In this format each pixel takes 12 bits of memory.

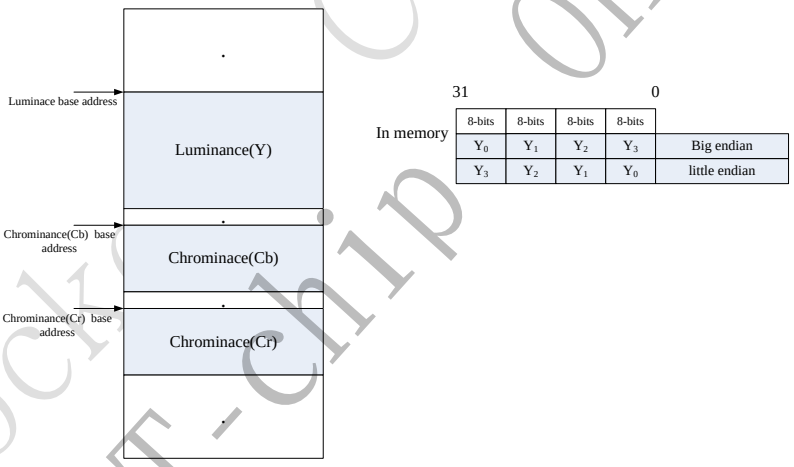

Fig. 25-2 VCODEC YCbCr 4:2:0 planar format

25.4.2 YCbCr 4:2:0 Semi-Planar format

In semi-planar YcbCr4:2:0 format the luminance samples from one plane in memory, and chrominance samples from another. Within one plane, the data has to be stored linearly and contiguously in the memory. The luminance pixels are stored in raster-scan order (Y_0Y_1 Y_2Y_3 Y_4….). The interleaved chrominance CbCr samples are stored in raster-scan order in memory as Cb_0Cr_0 Cb_1Cr_1 Cb_2 Cr_2 Cb_3Cr_3 Cb_4 Cr_4….

Semi-Planar format supports both progressive and interlaced format as presented in Fig. 25-3.The interlaced format may be alternative line or each line.

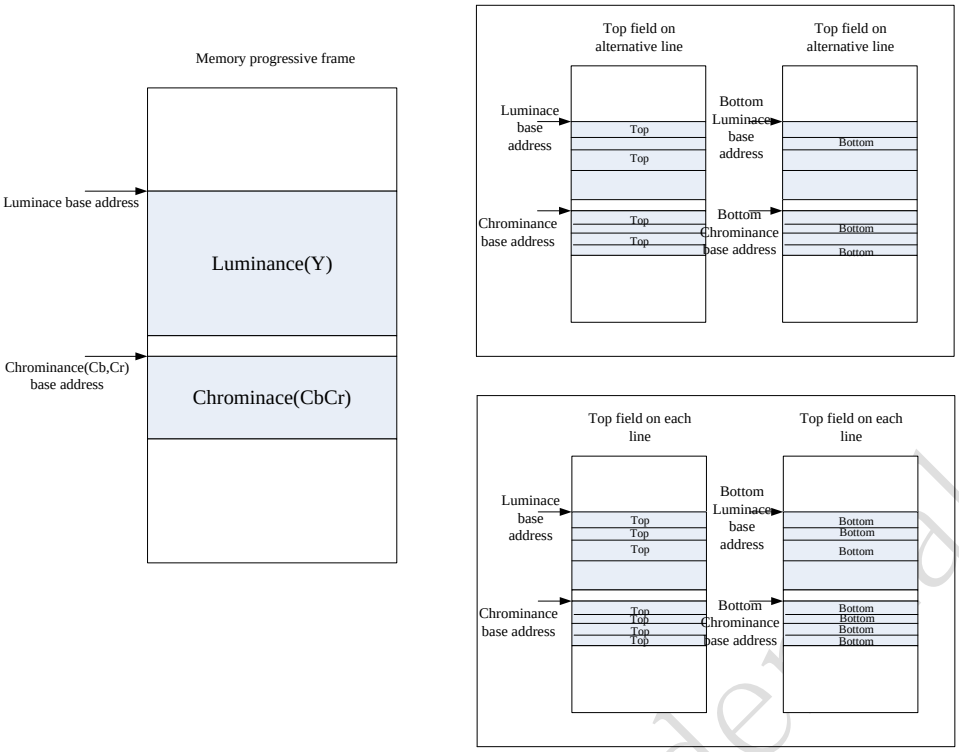

Fig. 25-3 VCODEC YCbCr 4:2:0 Semi-planar format

25.4.3 YCbCr 4:2:0 Tiled Semi-Planar Format

Like the YCbCr 4:2:0 semi-planar format, the tiled semi-planar format is also organized in the memory on two separate planes. The difference between these formats is that in tiled format the pixel samples are not anymore in raster-scan order but are stored macroblock(16x16 pixels) by macroblock. The samples of each macroblock are stored in consecutive addresses and the macroblocks are ordered from left to right and from top to down as Fig. 25-4. When this format used as input data format, it causes the lowest bus load to the system as there is minimal amount of non-sequential memory addressing required when reading the input data to the post-processor.

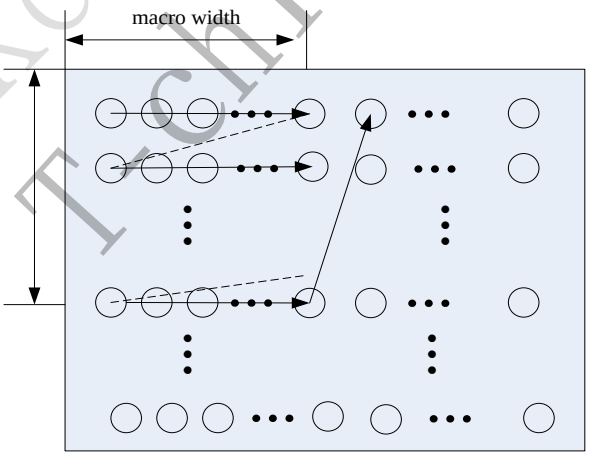

Fig. 25-4 VCODEC Tile scan mode

25.4.4 YCbCr 4:2:2 Interleaved Format

In the interleaved YCbCr 4:2:2 format the pixel samples from a single plane in which the data

has to be stored linearly and contiguously as shown in Fig. 25-5. The pixel data is in raster scan order and the chrominance samples are interleaved between the luminance samples as Y_0Cb_0 Y_1Cr_0 Y_2 Cb_1 Y_3Cr_1 Y_4 Cb_2…. YCrCb, CbYCrY and CrYCbY component orders are supported also. In this format, each pixel takes 16 bits in the memory.

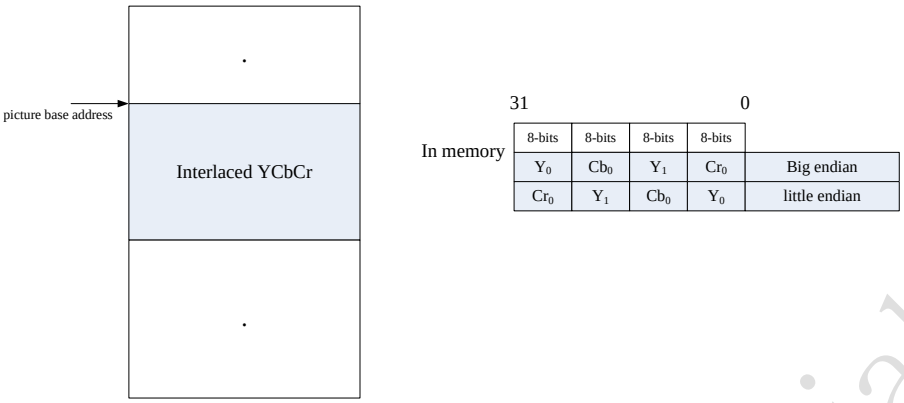

Fig. 25-5 VCODEC YCbCr4:2:2 Interleaved format

25.4.5 AYCbCr 4:4:4 Interleaved Format

In the interleaved YcbCr 4:2:2 format, the pixel samples from a single plane in which the data has to be stored linearly and contiguously as show in Fig. 25-6. The pixel data is in raster scan order and the chrominance and alpha channel samples are interleaved between the luminance samples as A_0Y_0 Cb_0Cr_0 A_1 Y_1 Cb_1Cr_1….

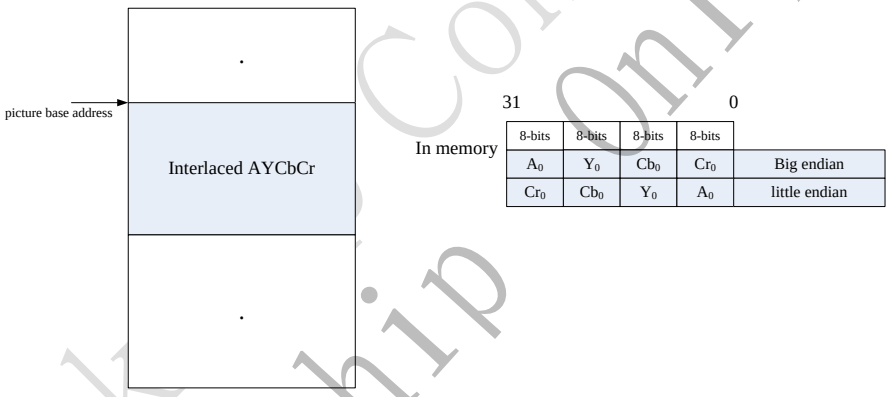

Fig. 25-6 VCODEC AYCbCr 4:4:4 Interleaved format

25.4.6 RGB 16bpp Format

In this format each pixel is represented by 16 or less bits containing the red, blue and green samples. There are several 16bpp formats which use different number of bits for each sample. For example the RGB 5-5-5 format uses 5 bits for each sample and 1 bit is left unused or can represent a transparency flag, where RGB 5-6-5 uses 6 bits for the G sample and 5 bits for R and B samples. Common for all 16bpp types is that two pixels fit into one 32-bit space.

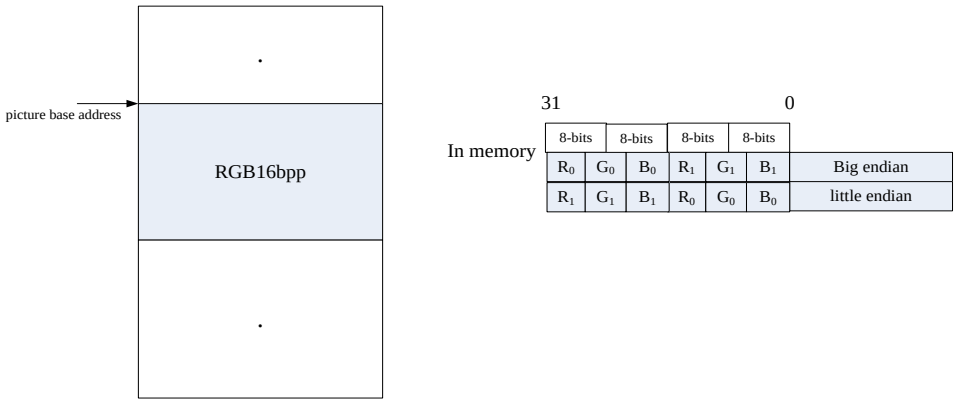

Fig. 25-7 VCODEC RGB 16bpp format

25.4.7 RGB 32bpp Format

Any RGB format that has its pixels represented by more than 16bits each is considered to be of 32bpp type. Typically in this format each pixel is represented by three bytes containing a red, blue and green sample and a 4th byte which can be empty or hold an alpha blending value. Common for all 32bpp types is that only one pixel fit into one 32-bit space. The data has to be stored linearly and contiguously in the memory.

25.5 Video Decoder

25.5.1 H.264 decoder

The H264 feature that Video decoder supports shows as Table 25-3.

Table 25-3 Video decoder H.264 feature

Feature	Decoder support
Input data format	H.264 byte or NAL unit stream /SVC stream /MVC stream
Decoding scheme	Frame by frame(or field by field) Slice by Slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 4096x2304 Step size 16 pixels
Maximum frame rate	60fps at 1920x1088 24fps at 4096x2304
Maximum bit rate	As specified by H.264 HP level 4.2
Error detection and concealment	Supported

The input of the decoder is H.264 standard bit stream in either plain NAL unit format or byte stream format. The input format in use will be automatically detected. The H.264 video encoding allows the use of multiple reference pictures, which means that the decoding order of the pictures may be different from their display order. The decoder can perform internally the display reordering of the decoded pictures or it can skip this and output all the pictures as soon as they are decoded.
The decoder has two operating modes: in the primary mode the HW performs entropy decoding, and in the secondary mode SW performs entropy decoding. Secondary mode is used in H.264 ASO or Slice Group stream decoding.

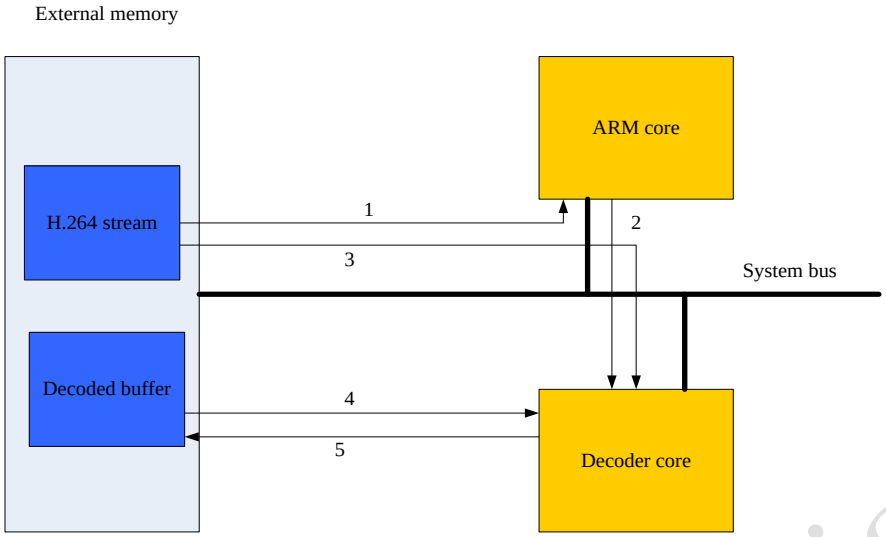

Fig. 25-8 Dataflow of HW performs entropy decoding in video decoder

The dataflow of HW performs entropy decoding is as Fig. 25-8 shown. The decoder software (SW) starts decoding the first picture by parsing the stream headers (1). Software then setups the hardware control registers (picture size, stream start address etc.) and enables the hardware (2). Hardware decodes the picture by reading stream (3) and the reference pictures (required for inter picture decoding)(4) from the external memory. Hardware writes the decoded output picture to memory one macroblock at a time (5). When the picture has been fully decoded or the hardware has run out of stream data, it gives an interrupt with a proper status flag and provides stream and address for software to continue and returns to initial state.

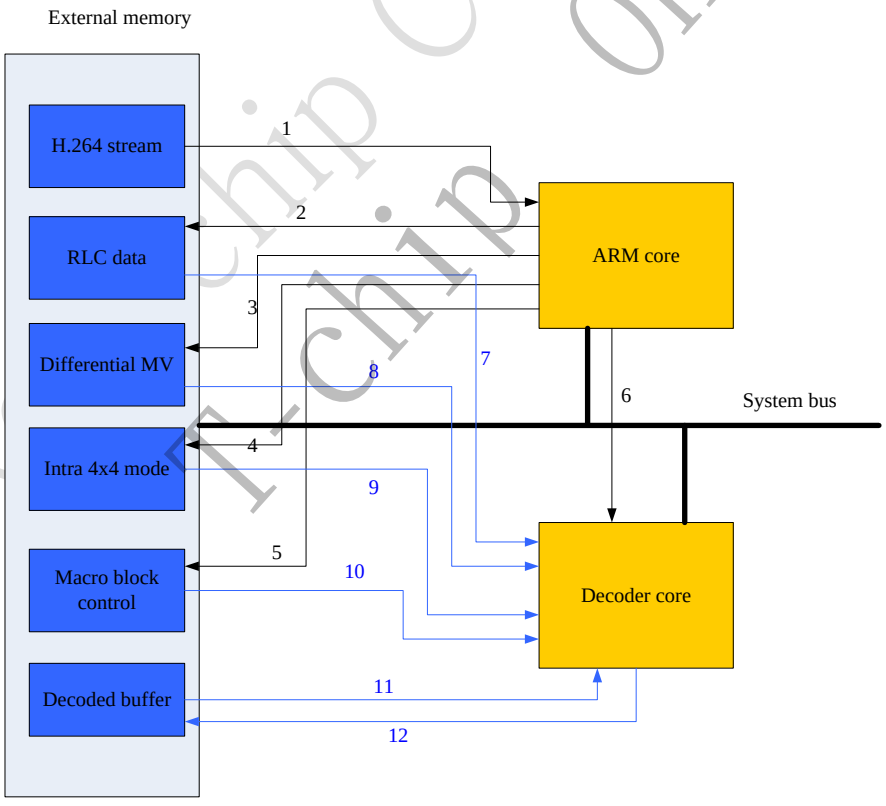

Fig. 25-9 Dataflow of SW performs entropy decoding in video decoder

SW entropy decoding mode (RLC mode) changes the input data format that is transferred

from SW to HW. The dataflow of this mode is as Fig. 25-9. In this case the decoder software starts decoding the first picture by parsing the stream headers (1), and by performing entropy decoding. Software then writes the following items to external memory:

Run-length-code (RLC) data (2)

Differential motion vectors (3)

Intra 4x4 prediction modes (4)

Macroblock control data (5)

Last step for the software is to write the hardware control registers and to enable the hardware (6).

Hardware decodes the picture by buffering control data for several macroblocks at a time, and reading then appropriate amount of RLC data, differential motion vectors and intra modes depending on each macroblock type (7)-(10). For the rest of the decoding process (11)-(12), the functionality is identical to the HW entropy decoding mode. When the picture has been fully decoded, hardware can raise an interrupt and write the status bits in the status register.

25.5.2 MPEG-4/H.263/SORENSON SPARK decoder

The features that video decoder supports about MPEG-4/H.263/SORENSON SPAR shows as Fig. 25-4.

Table 25-4 MPEG-4/H.263/SORENSON SPAR feature

Feature	Decoder support
Input data format	MPEG-4/H.263/Sorenson Spark elementary video stream
Decoding scheme	Frame by frame(or field by field) Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088(MPEG-4, Sorenson Spark) 48x48 to 720x576(H.263) Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by MPEG-4 ASP level5
Error detection and concealment	Supported

The decoder of MPEG-4/H.263/Sorenson has two operating modes: in the primary mode the HW performs entropy decoding, and in the secondary mode SW performs entropy decoding. Secondary mode is used in MPEG-4 data partitioned stream decoding.

25.5.3 MPEG-2/MPEG-1 decoder

The features of MPEG-2/MPEG-1 supported by decoder are shown as Table 25-5.

Table 25-5 MPEG-2/MPEG-1 features

Feature	Decoder support
Input data format	MPEG-2/MPEG-1 elementary video stream
Decoding scheme	Frame by frame(or field by field) Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels

	MPEG-2 can support up to 3840x2160
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by MPEG-2 MP high level
Error detection and concealment	Supported

The dataflow of MPEG-2/MPEG-1 is the same of H.264 HW performs entropy decoding as Fig. 25-8.

25.5.4 VC-1 decoder

The features of VC-1 supported by decoder are shown as Table 25-6.

Table 25-6 VC-1 features

Feature	Decoder support
Input data format	VC-1
Decoding scheme	Frame by frame(or field by field) Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	30fps at 1920x1088
Maximum bit rate	As specified by VC-1 AP level3
Error detection and concealment	Supported

The VC-1 decoder has only one operating mode in which the HW performs entropy decoding.

25.5.5 RV decoder

RV features supported by decoder are as shown in Table 25-7.

Table 25-7 RV features

Feature	Decoder support
Input data format	RV8,RV9 or RV10 stream
Decoding scheme	Frame by frame Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by RV specification
Error detection and concealment	Supported

The RV decoder has only one operating mode in which the HW performs entropy decoding.

25.5.6 VP6/VP8 decoder

VP6/VP8 features supported by decoder are as shown in Table 25-8.

Table 25-8 VP6/VP8 features

Feature	Decoder support
Input data format	VP6.0/VP6.1/VP6.2/VP8 stream

Decoding scheme	Frame by frame
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels VP8 can support up to 3840x2160
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by VP6/VP8 specification
Error detection and concealment	Supported

25.5.7 AVS decoder

AVS features supported by decoder are as shown in Table 25-9.

Table 25-9 AVS features

Feature	Decoder support
Input data format	AVS stream
Decoding scheme	Frame by frame, field by field Slice by slice
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by AVS standard
Error detection and concealment	Supported

25.5.8 DIVX decoder

DIVX features supported by decoder are as shown in Table 25-10.

Table 25-10 Divx features

Feature	Decoder support
Input data format	Divx 3,4,5 or 6 stream
Decoding scheme	Frame by frame Video packet by video packet
Output data format	YCbCr 4:2:0 semi-planar
Supported image size	48x48 to 1920x1088 Step size 16 pixels
Maximum frame rate	60fps at 1920x1088
Maximum bit rate	As specified by the Divx specification
Error detection and concealment	Supported

25.6 JPEG Decoder

JPEG features supported by decoder are as shown in Table 25-11.

Table 25-11 JPEG features

Feature	Decoder support
Input data format	JFIF file format 1.02 YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4

	semi-planar
Decoding scheme	Input: buffer by buffer, from 5Kb to 8MB at a time ① Output: from 1 MB row to 16 Mpixels at a time②
Output data format	YCbCr 4:0:0, 4:2:0, 4:2:2, 4:4:0, 4:1:1 and 4:4:4 semi-planar
Supported image size	48x48 to 8176x8176(66.8 Mpixels) Step size 8 pixels③
Maximum frame rate	Up to 76 million pixels pre second
Maximum bit rate	As specified by the Divx specification
Thumbnail decoding	JPEG compressed thumbnails supported
Error detection	Supported

①Programmable buffer size for optimizing performance and memory consumption. Interrupt will be issued when buffer runs empty, and the control software will load more streams to external memory.

②Programmable output slice for optimizing performance and memory consumption. Interrupt will be issued when the requested area decoded. The control software can be used to switch the decoder output address each time.

③Non-16x16 dividable resolutions will be filled to 16 pixel boundary.

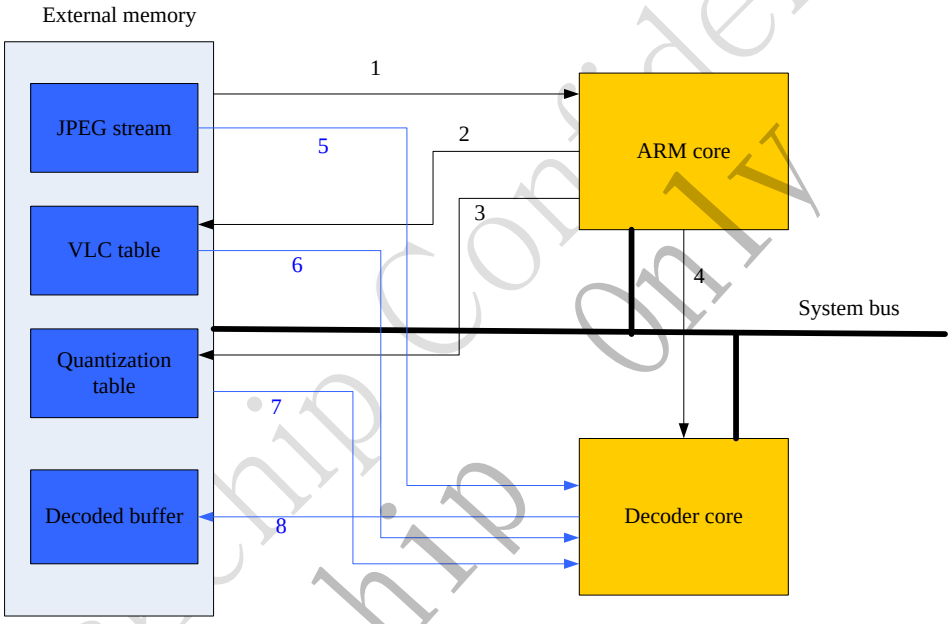

Fig. 25-10 The dataflow of JPEG decoder

The data flow of jpeg decoder is as Fig. 25-10 shown. The decoder software starts decoding the picture by parsing the stream headers (1) and then writes the following items to external memory:

VLC tables (2)

Quantization tables (3)

Last step for the software is to write the hardware control registers and to enable the hardware (4). After starting hardware, SW waits interrupt from HW.

Hardware decodes the picture by reading stream (5), VLC (6) and QP (7) tables. Hardware writes the decoded output picture memory one macroblock at a time (8). When the picture has been fully decoded, or the hardware has run out of stream data, it gives an interrupt with a proper status flag and provides stream end address for software to continue and returns to initial state.

25.7 Image Post-processor

The features supported by Post-processor are as show in Table 25-12.

Table 25-12 Post-processor features

Feature	Post-processor support
Input data format	Any format generated by the decoder in combined mode YCbCr 4:2:0 semi-planar YCbCr 4:2:0 planar YCbYCr 4:2:2 YCrYCb 4:2:2 CbYCrY 4:2:2 CrYCbY 4:2:2
Post-processor scheme	Frame by frame. Post-processor handles the image macroblock by macroblock, also in standalone mode.
Input image source	Internal source(combined mode) External source(standalone mode): e.g. a software decoder or camera interface
Output data format	YCbCr 4:2:0 semi-planar YCbCr 4:2:2 YCrYCb 4:2:2 CbYCrY 4:2:2 CrYCbY 4:2:2 Fully configurable ARGB channel lengths and locations inside 32 bits, e.g. ARGB 32-bit (8-8-8-8), RGB 16-bit(5-6-5), ARGB 16-ibt(4-4-4-4).
Input image size (combined mode)	48x48 to 8176x8176(66.8 Mpixels) Step size 16 pixels
Input image size (stand-alone mode)	Width from 48 to 8176 Height from 48 to 8176 Maximum size limited to 66.8 Mpixels Step size 16 pixels
Output image size	16x16 to 2560x4088 (when there is no scale, it can support 4088x4088) Horizontal step size 8 Vertical step size 2
Image up-scaling①	Bicubic polynomial interpolation with a four-tap horizontal kernel and a two-tap vertical kernel Arbitrary, non-integer scaling ratio, separately for both dimensions. Maximum output width is 3x the input width(within the maximum output image size limit) Maximum output height is 3x the input height -2 pixels (within the maximum output image size limit) Maximum output height is 2.5x the input height – 2 pixels (within the maximum output image size limit) when running RealVideo, VP8 format in pipeline
Image down-scaling①	Proprietary averaging filter Arbitrary, non-integer scaling ratio separately for both dimensions Unlimited down-scaling ratio
YCbCr to RGB color conversion	BT.601-5 compliant BT.709 compliant User definable conversion coefficient
Dithering	allego dithering for 4,5 and 6 bit RGB channel precision

RGB image contrast adjustment	Segmented linear
RGB image brightness adjustment	Linear
RGB image color saturation adjustment	Linear
De-blocking filter for MPEG-4 simple profile /H.263 /Sorenson	Using a modified H.264 in-loop filter as a post-processing filter. Filtering has to be performed in combined mode
Image cropping / digital zoom	User definable start position, height and width. Can be used with scaling to perform digital zoom. Usable only for JPEG or stand-alone mode.
Picture in picture	Output image can be written to any location inside video memory. Up to 2560x4088 sized displays supported. (when there is no scale, the size can up to 4088x4088)
Image rotation	Rotation 90,180, or 270 degrees Horizontal flip Vertical flip

①It is not allowed to perform horizontal up-scaling and vertical down-scaling (or vice versa) at the same. If needed, this kind of operation can be performed in two phases.

The PP has two modes: standalone mode and pipe-line mode. In standalone mode, picture processing is performed to any external source. The processing is done independently and asynchronously from the video decoder. The dataflow block gram is as Fig. 25-11 shows.

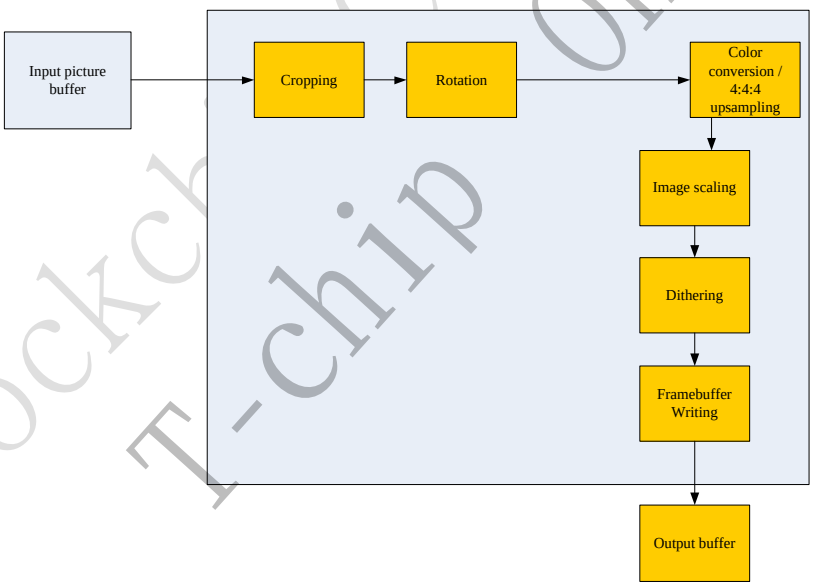

Fig. 25-11 Post-process standalone dataflow

In pipe-line mode, the post-processor works together with the multi-format decoder. The PP will take its input directly from the decoder. The post-processor doesn't have cropping function in pipe-line mode other than combined with jpeg decoder. The dataflow is as Fig. 25-12 show. In the pipe-line mode, most decoder will also put the data to the decoder out buffer other than JPEG decoder. So, JPEG decoder with pipe-line mode will save bus bandwidth when it crops the input picture to a smaller picture.

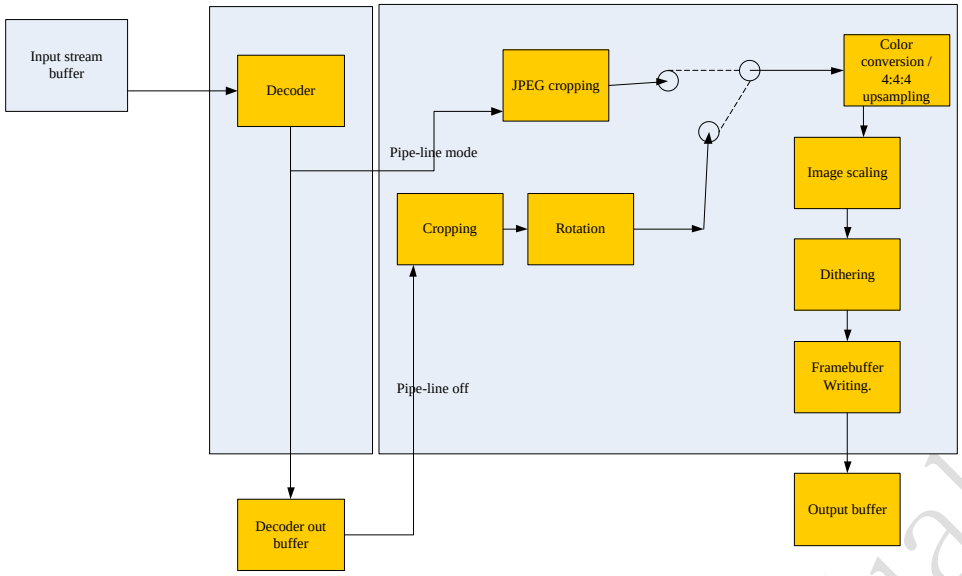

Fig. 25-12 Post-process Pipe-line Mode Dataflow

The post-processor has some restrictions in the input and output picture size. Table 25-13 presents the divisibility requirements for all the post-processor parameters.

Table 25-13 Requirements for post-processor

Output format parameters	YCbCr 4:2:0	YCbCr 4:2:2	RGB16bpp	RGB32bpp
Input picture width and height	16	16	16	16
Cropped picture width and height	8	8	8	8
Cropping start coordinates(x,y)	16	16	16	16
Output picture width	8	8	8	8
Output picture height	2	2	2	2
Masks width and origin X	8	4	4	2
Masks width and origin Y	2	1	1	1
Frame buffer width and origin X	8	4	4	2
Frame buffer height and origin Y	2	1	1	1

25.8 Image Pre-processor

Pre-processor is pipelined with the encoder and it can be used only with the encoder. Pre-processor features are presented in Table 25-14.

Table 25-14 Post-processor features

Feature	Encoder support
RGB to YCbCr 4:2:0 color space	BT.601, BT.709 or user defined

conversion	coefficients conversion for RGB: ● RGB444 and BGR444 ● RGB555 and BRG555 ● RGB565 and BGR565 ● RGB888 and BRG888 ● RGB101010 and BRG101010
YCbCr 4:2:2 to YCbCr 4:2:0 color space conversion	YCbCr formats: ● YCbCr 4:2:0 planar ● YCbCr 4:2:0 semi-planar ● YCbYCr 4:2:2 ● CbYCrY 4:2:2 interleaved
Cropping	Video – from 8192x8192 to any supported encoding size
Rotation	90 or 270 degrees

25.9 H.264 Encoder

The H.264 features supported by encoder are as shown in Table 25-15 .

Table 25-15 Video encoder H.264 feature

Feature	Encoder support
Input data format	● YCbCr formats: YCbCr 4:2:0 planar YCbCr 4:2:0 semi-planar YCbYCr 4:2:2① CbYCrY 4:2:2 Interleaved① ● RGB formats:① RGB444 to BGR444 RGB555 to BGR555 RGB565 to BGR565 RGB888 to BRG888 RGB101010 and BRG 101010
Output data format	H.264: Byte unit stream NAL unit stream
Supported image size	96x96 to 1920x1080(Full HD) Step size 4 pixels
Maximum frame rate	30 fps at 1920 x1080
Bit rate	Maximum 20Mbps Minimum 10kbps

①internally encoder handles image only in 4:2:0 format

Figure Fig. 25-13 illustrates the encoder data flow in H.264 encoding mode. The numbers present the following transactions:

1. Memory-mapped register writes and reads
2. Input image read
3. Reference image write
4. Reference image read
5. NAL sizes write from HW
6. NAL sizes read to SW
7. Output byte or NAL unit stream write from HW

8. Output byte or NAL unit stream headers write from SW

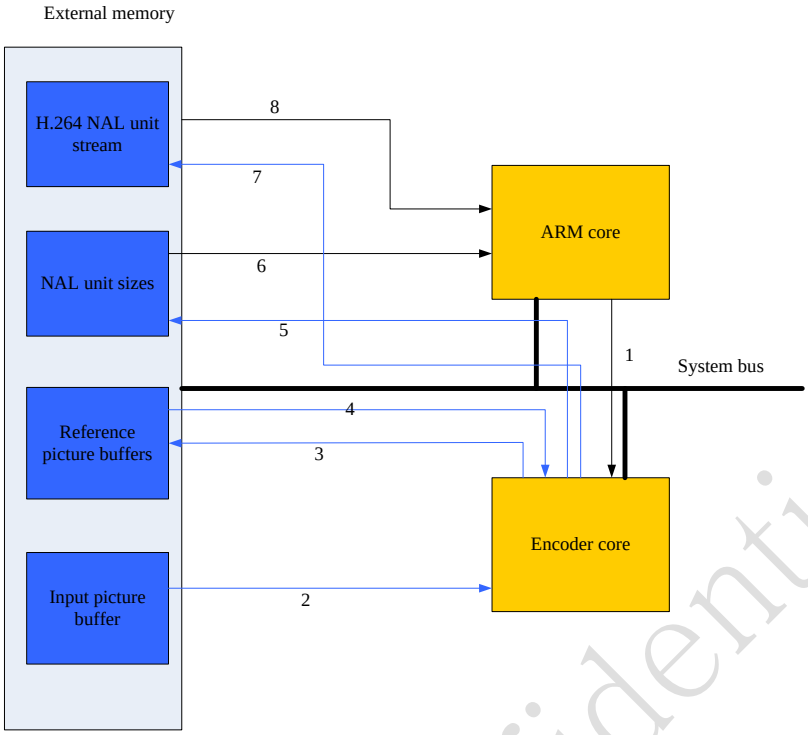

Fig. 25-13 Video Encoder Dataflow

The encoder software starts encoding the first picture by initializing hardware and writing the stream headers. After HW has encoded the image, SW calculates new quantization values for HW, and initializes HW again.

25.10 JPEG Encoder

The JPEG features supported by the encoder are as shown in Table 25-16.

Table 25-16 JPGE features

Feature	Encoder support
Input data format	● YCbCr formats: YCbCr 4:2:0 planar YCbCr 4:2:0 semi-planar YCbCr 4:2:2① CbYCrY 4:2:2 Interleaved ① ● RGB formats:① RGB444 and BGR444 RGB555 and BGR555 RGB565 and BGR565 RGB888 and BRG888 RGB101010 and BRG101010
Output data format	JFIF ifle format 1.02 Non-progressive JPEG
Supported image size	96x32 to 8192x8192(64 million pixels) Step size 4 pixels
Maximum data rate	Up to 90 million pixels per second
Thumbnail insertion	RGB 8-bits, RGB 24-bits and JPEG compressed thumbnails supported

①internally encoder handles images only in 4:2:0 format

25.11 MMU

The MMU divides memory into 4KB pages, where each page can be individually configured. For each page the following parameters are specified:

- Address translation of virtual memory, this enables the processor to work using address that differ from the physical address in the memory system.
- The permitted types of accesses to that page. Each page can permit read, write, both, or none.

The MMU use 2-level page table structure:

1. The first level, the page directory consists of 1024 directory table entries(DTEs), each pointing to a page table.

2. The second level, the page table consists of 1024 page table entries(PTEs), each pointing to a page in memory.

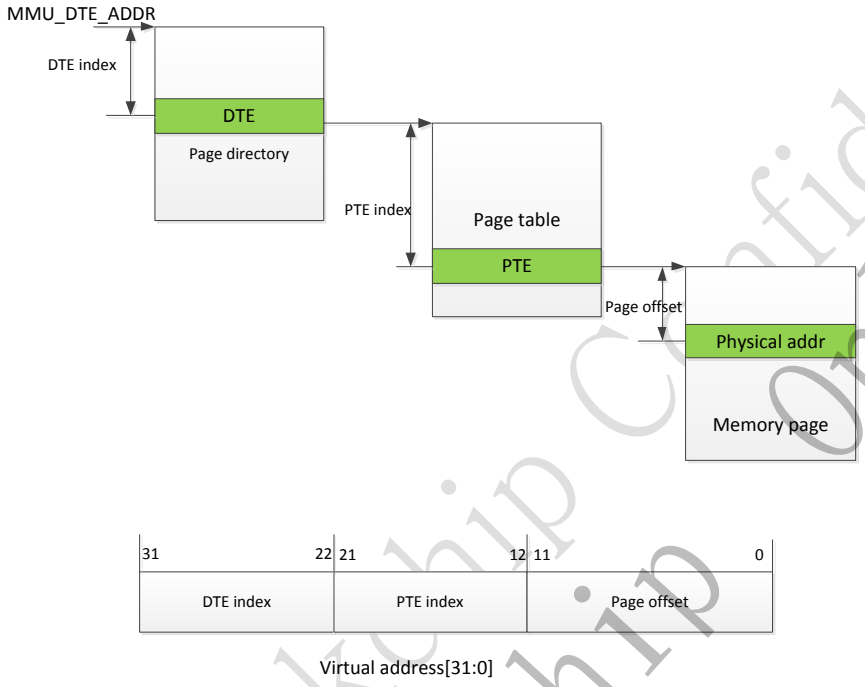

Fig. 25-14 structure of two-level page table

25.12 Register Description

This section describes the control/status registers of the design.

The VEPU base address is 0xff9a0000, the VDPU base address is 0xff9a0400.

The MMU base address is 0xff9a0800, the VDPU cache control base address is 0xff0a0c00.

Please refer to the document VDPU_SWReg_Map.pdf and VEPU_SWReg_Map.pdf.

The table below is the add new feature on the VDPU_SWReg_Map.pdf

25.12.1 VDPU new feature detail register description

SWREG15
Address: Operational Base + offset (0x03c)

bit	Attr	Reset Value	Description
19	RW	0x0	Sw_jpegroi_in_endian 0 = big endian (0-1-2-3 order) 1 = little endian (3-2-1-0 order)
18	RW	0x0	Sw_jpegroi_in_swap32 '0' = no swapping of 32 bit words '1' = 32bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1 byte order (also little endian should be enabled)
17:16	RW	0x0	Sw_roi_sample_size[1:0] ROI MB num sample each time 00: 1 01: 8 10:16 11: 8
15:12	RW	0x0	Sw_roi_distance[3:0] The distance between the sample MB and ROI start MB
11:10	RW	0x0	Sw_roi_out_sel[1:0] 00: output control 01: output picture 10: output control and picture 11: output control
9	RW	0x0	Sw_roi_decode 1'b0: build offset/dc table 1'b1: ROI decode
8	RW	0x0	Sw_roi_en 1'b0: normal jpeg decode mode 1'b1: jpeg roi mode

SWREG35

Address: Operational Base + offset (0x08c)

bit	Attr	Reset Value	Description
31:2	RW	0x0	Sw_jpegdcoff_base

SWREG36

Address: Operational Base + offset (0x090)

bit	Attr	Reset Value	Description
16:0	RW	0x0	Sw_jpegdcoff_len, the number of 64bit jpegdcoff, it can be used both when sw_roi_decode is 1'b0 or 1'b1

SWREG57

Address: Operational Base + offset (0x0E4)

bit	Attr	Reset Value	Description
14:8	R	0x0	Debug_service[6:0] Debug_service[6:0] = {service_wr[2:0], service_rd[3:0]} Now, it is only for paral axi bus version
7	RW	0x0	sw_cache_en when sw_cache_en = 1'b1 and sw_pref_sigchan = 1'b1, the prefetch cache is enable
6	RW	0x0	sw_pref_sigchan when it is set to 1'b1, the prefetch data is a single channel

5	RW	0x0	Sw_axird_sel:default is 1'b0 1'b0: auto sel encoder axi signals and decoder axi signals 1'b1: sel decoder axi signals(it only use to set bu_dec_e to 1'b0 in the middle of a frame
4	RW	0x1	Sw_paral_bus: when it is set 1'b1, the axi support read and write service parallel; when it is set 1'b0, the axi only support read and write serial
3	RW	0x0	sw_intra_dbl3t: in chroma dc intra prediction, when this bit is enable, there will 3 cycle enhance for every block
2	RW	0x0	Sw_intra_dblspeed: intra double speed enable
1	RW	0x0	Sw_inter_dblspeed: inter double speed enable
0	RW	0x0	Sw_stream_len_hi: the extension bit of sw_stream_len

SWREG58

Address: Operational Base + offset (0x0E8)

bit	Attr	Reset Value	Description
30	R	0x0	Mvst_mv_req signal value
29	R	0x0	Debug_rlc_req: scst_rlc_req signal value
28	R	0x0	Debug_res_y_req: prtr_res_y_req signal value
27	R	0x0	Debug_res_c_req: prtr_res_c_req signal value
26	R	0x0	Debug_strm_da_e: strm_da_e signal value
25	R	0x0	Debug_framerdy: dfbu_framerdy signal value
24	R	0x0	Debug_filter_req: dfbu_req_e signal value
23	R	0x0	Debug_referreq0: prbu_referreq0 signal value
22	R	0x0	Debug_referreq1: Prbu_referreq1 signal value
20:0	R	0x0	Debug_dec_mb_counter: HW internal MB counter value , it is now only reset by dec_e pulse

SWREG98

Address: Operational Base + offset (0x18C)

bit	Attr	Reset Value	Description
1	RW	0x0	Sw_pp_out_h_ext
0	RW	0x0	Sw_pp_out_w_ext

SWREG101

Address: Operational Base + offset (0x194)

bit	Attr	Reset Value	Description
0	W	0x0	Soft reset signals When write to 1'b1, it will reset VDPU, VEPU and pp when write to 1'b0, no use

The shared MMU register is as below

25.12.2 MMU Register Summary

Name	Offset	Size	Reset Value	Description
VCODEC_MMU_DTE_ADDR	0x0000	W	0x00000000	MMU current page Table address
VCODEC_MMU_STATUS	0x0004	W	0x00000000	MMU status register
VCODEC_MMU_COMMAND	0x0008	W	0x00000000	MMU command register
VCODEC_MMU_PAGE_FAULT_ADDR	0x000c	W	0x00000000	MMU logical address of last page fault
VCODEC_MMU_ZAP_ONE_LINE	0x0010	W	0x00000000	MMU Zap cache line register
VCODEC_MMU_INT_RAWSTAT	0x0014	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_CLEAR	0x0018	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_MASK	0x001c	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_INT_STATUS	0x0020	W	0x00000000	MMU raw interrupt status register
VCODEC_MMU_AUTO_GATING	0x0024	W	0x00000001	mmu auto gating

*Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access*

25.12.3 MMU Detail Register Description

VCODEC_MMU_DTE_ADDR

Address: Operational Base + offset (0x0000)

MMU current page Table address

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	MMU_DTE_ADDR MMU_DTE_ADDR MMU current page Table address

VCODEC_MMU_STATUS

Address: Operational Base + offset (0x0004)

MMU status register

Bit	Attr	Reset Value	Description
31:11	RO	0x0	reserved
10:6	RO	0x00	PAGE_FAULT_BUS_ID page fault bus id Index of master reponsible for last page fault
5	RO	0x0	PAGE_FAULT_IS_WRITE page fault acess The direction of access for last page fault: 0 = Read 1 = Write
4	RO	0x0	REPLAY_BUFFER_EMPTY replay buffer empty The MMU replay buffer is empty
3	RO	0x0	MMU_IDLE mmu idle The MMU is idle when accesses are being translated and there are no unfinished translated accesses.

Bit	Attr	Reset Value	Description
2	RO	0x0	STAIL_ACTIVE stall active MMU stall mode currently enabled. The mode is enabled by command
1	RO	0x0	PAGE_FAULT_ACTIVE page fault active MMU page fault mode currently enabled . The mode is enabled by command.
0	RO	0x0	PAGING_ENABLED Paging is enabled Paging is enabled

VCODEC_MMU_COMMAND
Address: Operational Base + offset (0x0008)
MMU command register

Bit	Attr	Reset Value	Description
31:3	RO	0x0	reserved
2:0	WO	0x0	MMU_CMD mmu cmd MMU_CMD. This can be: 0: MMU_ENABLE_PAGING 1: MMU_DISABLE_PAGING 2: MMU_ENABLE_STALL 3: MMU_DISABLE_STALL 4: MMU_ZAP_CACHE 5: MMU_PAGE_FAULT_DONE 6: MMU_FORCE_RESET

VCODEC_MMU_PAGE_FAULT_ADDR
Address: Operational Base + offset (0x000c)
MMU logical address of last page fault

Bit	Attr	Reset Value	Description
31:0	RO	0x00000000	PAGE_FAULT_ADDR page fault addr address of last page fault

VCODEC_MMU_ZAP_ONE_LINE
Address: Operational Base + offset (0x0010)
MMU Zap cache line register

Bit	Attr	Reset Value	Description
31:0	WO	0x00000000	MMU_ZAP_ONE_LINE zap one line address to be invalidated from the page table cache

VCODEC_MMU_INT_RAWSTAT
Address: Operational Base + offset (0x0014)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error read bus error status
0	RW	0x0	PAGE_FAULT page fault page fault status

VCODEC_MMU_INT_CLEAR
Address: Operational Base + offset (0x0018)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	WO	0x0	READ_BUS_ERROR read bus error write 1 to clear read bus error
0	WO	0x0	PAGE_FAULT page fault clear write 1 to page fault clear

VCODEC_MMU_INT_MASK
Address: Operational Base + offset (0x001c)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	READ_BUS_ERROR read bus error enable the read bus interrupt souce when this bit is set to 1'b1
0	RW	0x0	PAGE_FAULT page fault mask enable the page fault interrupt souce when this bit is set to 1'b1

VCODEC_MMU_INT_STATUS
Address: Operational Base + offset (0x0020)
MMU raw interrupt status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RO	0x0	READ_BUS_ERROR read bus error read bus error status

Bit	Attr	Reset Value	Description
0	RO	0x0	PAGE_FAULT page fault page fault status

VCODEC_MMU_AUTO_GATING
Address: Operational Base + offset (0x0024)
mmu auto gating

Bit	Attr	Reset Value	Description
31:1	RO	0x0	reserved
0	RW	0x1	mmu_auto_clkgating mmu_auto_clkgating when it is 1'b1, the mmu will auto gating it self

25.12.4 VDPU pref_cache Register Summary

Name	Offset	Size	Reset Value	Description
pref_cache_VERSION	0x0000	W	0xcac20101	VERSION register
pref_cache_SIZE	0x0004	W	0x06170206	L2 cache SIZE
pref_cache_STATUS	0x0008	W	0x00000000	Status register
pref_cache_COMMAND	0x0010	W	0x00000000	Command setting register
pref_cache_CLEAR_PAGE	0x0014	W	0x00000000	clear page register
pref_cache_MAX_READS	0x0018	W	0x0000001c	maximum read register
pref_cache_PERFCNT_SRC0	0x0020	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL0	0x0024	W	0x00000000	performance counter 0 value register
pref_cache_PERFCNT_SRC1	0x0028	W	0x00000000	performance counter 0 source register
pref_cache_PERFCNT_VAL1	0x002c	W	0x00000000	performance counter 1 value register

*Notes: Size : **B** - Byte (8 bits) access, **HW** - Half WORD (16 bits) access, **W** -WORD (32 bits) access*

25.12.5 VDPU pref_cache Detail Register Description

pref_cache_VERSION
Address: Operational Base + offset (0x0000)
VERSION register

Bit	Attr	Reset Value	Description
31:16	RO	0xcac2	PRODUCT_ID
15:8	RO	0x01	VERSION_MAJOR
7:0	RO	0x01	VERSION_MINOR

pref_cache_SIZE
Address: Operational Base + offset (0x0004)
L2 cache SIZE

Bit	Attr	Reset Value	Description

Bit	Attr	Reset Value	Description
31:24	RO	0x06	External_bus_width Log2 external bus width in bits
23:16	RO	0x17	CACHE_SIZE Log2 cache size in bytes
15:8	RO	0x02	ASSOCIATIVITY Log2 associativity
7:0	RO	0x06	LINE_SIZE Log2 line size in bytes

pref_cache_STATUS
Address: Operational Base + offset (0x0008)
Status register

Bit	Attr	Reset Value	Description
31:2	RO	0x0	reserved
1	RW	0x0	DATA_BUSY set when the cache is busy handling data
0	RW	0x0	CMD_BUSY set when the cache is busy handling commands

pref_cache_COMMAND
Address: Operational Base + offset (0x0010)
Command setting register

Bit	Attr	Reset Value	Description
31:6	RO	0x0	reserved
5:4	RW	0x0	sw_addrb_sel 2'b00: to sel b[14:6] 2'b01: to sel b[15:9], b[7:6] 2'b10: to sel b[16:10], b[7:6] 2'b11: to sel b[17:11], b[7:6]
3	RO	0x0	reserved
2:0	RW	0x0	COMMAND The possible command is 1 = Clear entire cache

pref_cache_CLEAR_PAGE
Address: Operational Base + offset (0x0014)
clear page register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	CLEAR_PAGE writing an address, invlidates all lines in that page from the cache

pref_cache_MAX_READS
Address: Operational Base + offset (0x0018)
maximum read register

Bit	Attr	Reset Value	Description
31:5	RO	0x0	reserved
4:0	RW	0x1c	MAX_READS Limit the number of outstanding read transactions to this amount

pref_cache_PERFCNT_SRC0
Address: Operational Base + offset (0x0020)
Performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved
3:0	RW	0x0	PERFCNT_SRC0 This register holds all the possible source values for Performance Counter 0 0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave

pref_cache_PERFCNT_VAL0
Address: Operational Base + offset (0x0024)
performance counter 0 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERFCNT_VAL0 Performance counter 0 value

pref_cache_PERFCNT_SRC1
Address: Operational Base + offset (0x0028)
performance counter 0 source register

Bit	Attr	Reset Value	Description
31:4	RO	0x0	reserved

Bit	Attr	Reset Value	Description
3:0	RW	0x0	PERFCNT_SRC1 This register holds all the possible source values for Performance Counter 1 0: total clock cycles 1: active clock cycles 2: read transactions, master 3: word reads, master 4: read transactions, slave 5: word reads, slave 6: read hit, slave 7: read misses, slave 8: read invalidates, slave 9: cacheable read transactions, slave 10: bad hit nmber, slave

pref_cache_PERFCNT_VAL1
Address: Operational Base + offset (0x002c)
Performance counter 1 value register

Bit	Attr	Reset Value	Description
31:0	RW	0x00000000	PERFCNT_VAL1 Performance counter 1 value

25.13 Timing Diagram

Fig. 25-15 illustrates the internal clock structure of VCODEC. VCODEC has two clocks input, which are aclk_vcodec and hclk_vcodec. They can be different frequency, but they should be in the same clock domains.

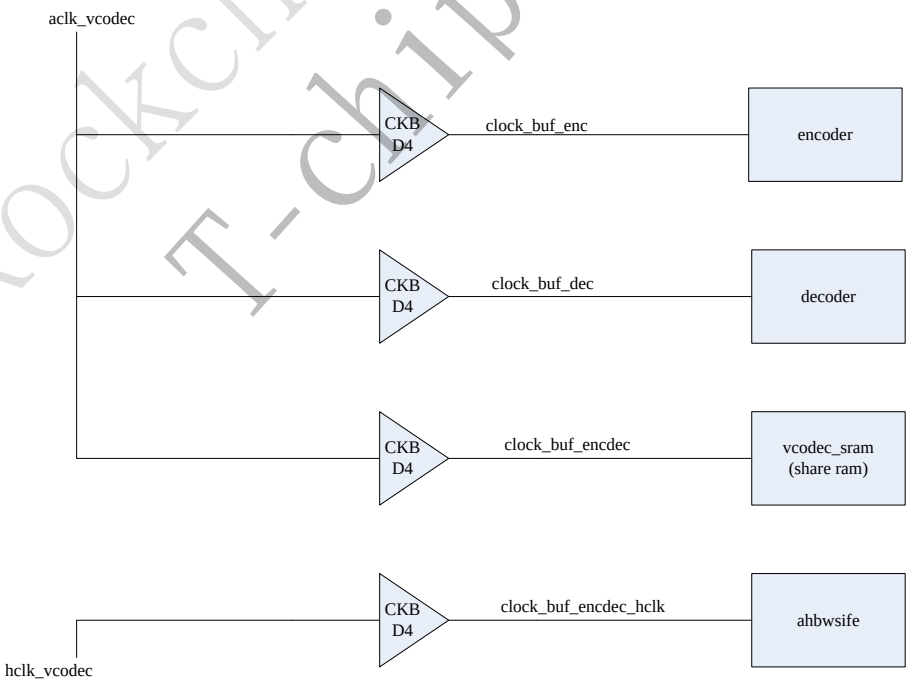

Fig. 25-15 VCODEC clock structure

The Fig. 25-16 shows the aclk_vcodec and hclk_vcodec architecture in the CRU module. Most signals are from the CRU register CRU_CLKSEL32_CON, while only one signal is from the GRF register GRF_SOC_CON0.

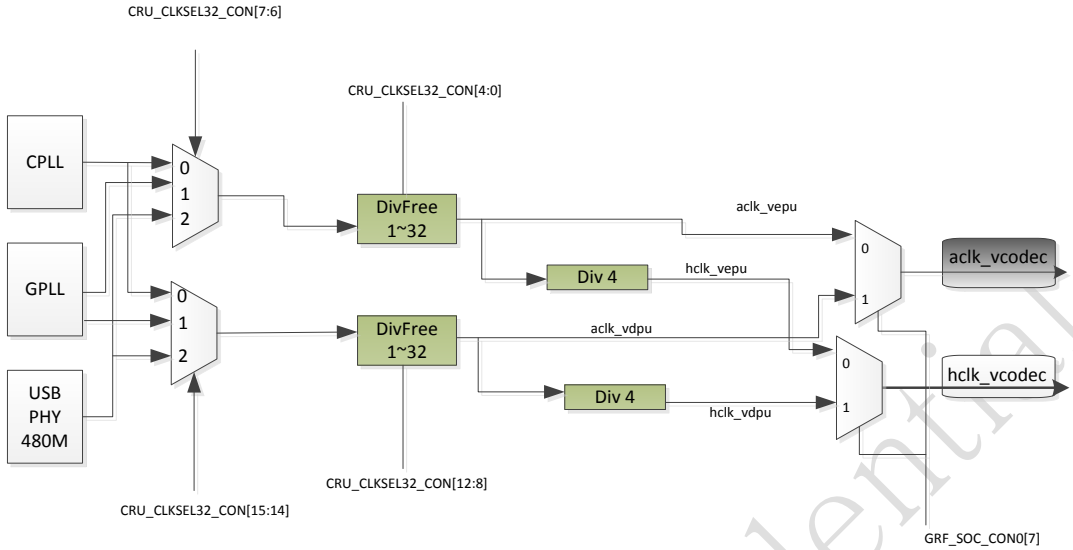

Fig. 25-16 Aclk_vcodec and Hclk_vcodec Architecture

When the encoder is working, the maximum frequency of aclk_vcodec is 400MHz@worst case. The maximum frequency of aclk_vcodec is also 400MHz@worst case when the decoder is working.

25.14 Interface Description

VCODEC supports writing and reading its internal registers through AHB bus and it just supports single 32bits read and write.

VCODEC reads the input data and write the output data through AXI bus. The VCODEC AXI master supports up to 32 outstanding bursts (in read) in most cases, while 24 bursts are the maximum in write. There are multiple bursts issued as outstanding bursts, they have the same ID. When the ID would change, the previous ID transactions are first completed. So the VCODEC AXI master doesn't support out of order.

VCODEC has three interrupt output signals vdpu_intr and vepu_intr and irq_mmu, which are high valid.

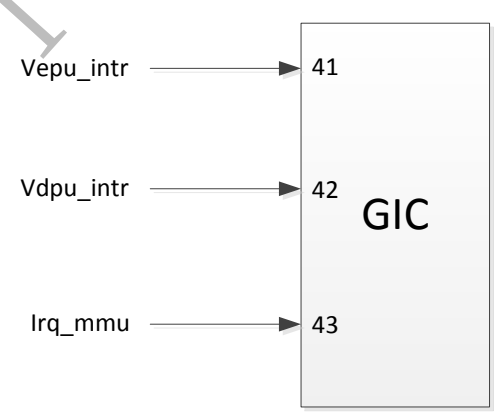

Fig. 25-17 The interrupt interface of vcodec

25.15 Application Notes

- In decoder other than JPEG decoder, the input stream buffer should at least contain a slice or a frame data, otherwise the decoder will produce an interrupt and show error and then reset itself.
- In encoder, we can configure the registers to control the input picture data format (such as endian and swap), but some input data format are fixed, such as cabac_table data.
- The register VEPU_SWREG64~95 are JPEG quantization registers. They are write only registers. When you want to write these registers, you should first set VEPU_SWREG14[0] to 1'b0 and VEPU_SWREG14[2:1] to 2'b10(select JPEG mode).
- The decoder can support ref buffer mode or cacheable mode, but they can't be both enabled. We can config the swreg57[6],swreg57[7] to enable cache and config the swreg51 to control the ref buffer.