MPEG-4 AVC – H.264 Deblocking loop filter IP Datasheet

SUMMARY

1 Introduction

1.1 H.264 Overview

The H.264 is also known as MPEG-4 ISO/IEC14496-10 or MPEG-4 / AVC. This standard has been co-developed by JVT group composed by MPEG-ISO/IEC members and VCEG-ITU-T members.

Three profiles have first been defined, each with several levels. A High Profile has been added, and standardisation work is going on.
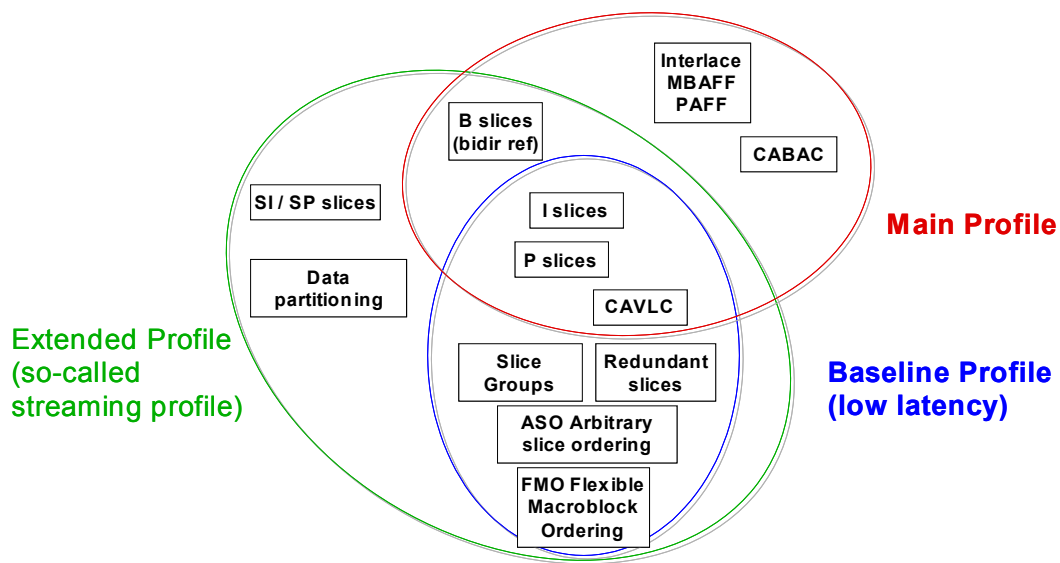
Figure 1- Profiles of MPEG-4 part 10 / AVC H.264

Ateme has developed a full Main Profile Level 1-4 implementation of H.264 encoder and decoder, ensuring Baseline, Extended and High Profiles partial compatibility.

The encoder is available as:
- – Software library for x86 (PC),
- – VHDL firmware, for embedded equipements.

The decoder is available as software library for x86 and for DSP.

The synoptic for a full encoder is shown below, with the Deblocking loop filter highlighted.
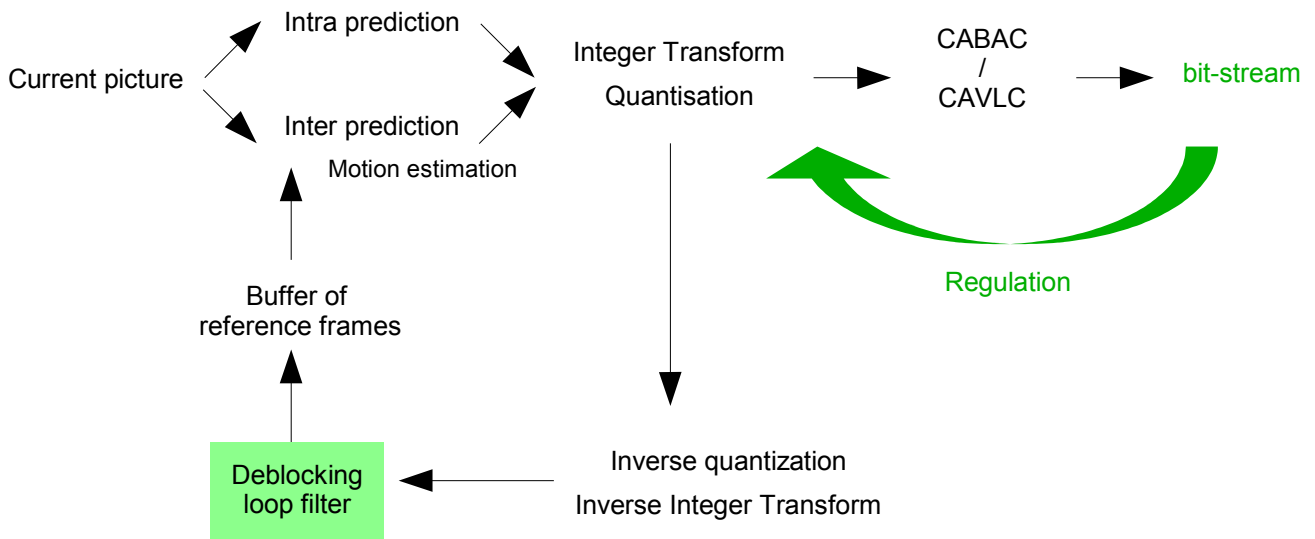

Figure 2- synoptic of MPEG-4 part 10 / AVC H.264 encoder

A video sequence is composed of images. In the H.264 process, an image is split in macroblocks (MB). A macroblock is a 16x16 pixels luminance block and two 8x8 pixels chrominance blocks.

A slice is a collection of macroblocks belonging to a single image.

The H.264 bitstream is structured in NAL units. Each NAL unit contains the information of one slice.

1.2 IP overview

Ateme's H.264 deblocking Filter IP is an hardware module able to filter the macroblocks information compliant with ISO/IEC14496-10 specification.

The IP has two inputs data path. One data path is used to receive control words and the second one is used to receive current macroblock pixels to process.

The neighborhood information necessary for filtering is autonomously managed.

The module is designed to compute full size video sequences. It can be adapted to any format.

1.3 Commercial References

The IP is available under the following references:

	SD – levels 1-3	*HD – levels 1-4*
H264_SD-LOOPFILT	X	
H264_HD-LOOPFILT	X	X

SD stands for Standard Definition, or Full D1, or 720x480 30 / 29.97 fps in NTSC and 720x576 25 fps in PAL.
This corresponds to a throughput of max 60750 Macroblocks per second.
Level 3 is up to 10 Mbps.

HD stands for High Definition, with 720p 50 / 60, 1080 i 50 / 60, 1080 p 25 / 30.
This corresponds to a throughput of max 367200 Macroblocks per second.
- HD 720p/25-30fps is level 3.1 - up to 14 Mbps
- HD 720p/50-60fps is level 3.2 - up to 20 Mbps
- HD 1080 i&p/25/30fps is level 4 up to 25 Mbps

Other products are available from Ateme:
- CABAC/CAVLC Bitstream encoding IP
- Motion Estimation IP
- Full SD main profile Level 3 encoder IP,
  including video acquisition, decision, regulation, memory management, PCI, etc...

2 Features

- Platform independent design (written in VHDL)
- Full support of Main and Baseline profiles
- MBAFF (MB Adaptive Frame and Field) support
- Non adjacent macroblocks support
- Neighborhood context autonomously managed
- PAL and NTSC support
- HD ready

3 Performances

High throughput performances are obtained with low clock frequency.

The performances are given for :
- A progressive and / or field encoded picture (720x480)
- MBAFF mode active or not.

The resulting performances are :
- Time to process a MB in frame mode: ~ 230 clock cycles
- Time to process two Mb in MBAFF mode: ~ 430 clock cycles
- Time to process a 720 x 480 picture in frame mode: ~ 3 ms with a 100 MHz clock frequency

4 Resources

The IP has been optimized, thus for information on footprint and size, Logic Element number, and memory usage, please consult us for your specific target.

5 Architecture

5.1 Principle

The IP uses a generic bus interface for registers and streams.
It is delivered encapsulated with two AHB agents:

      1.AHB Slave for registers access

      2.AHB Master to access directly data in external memory
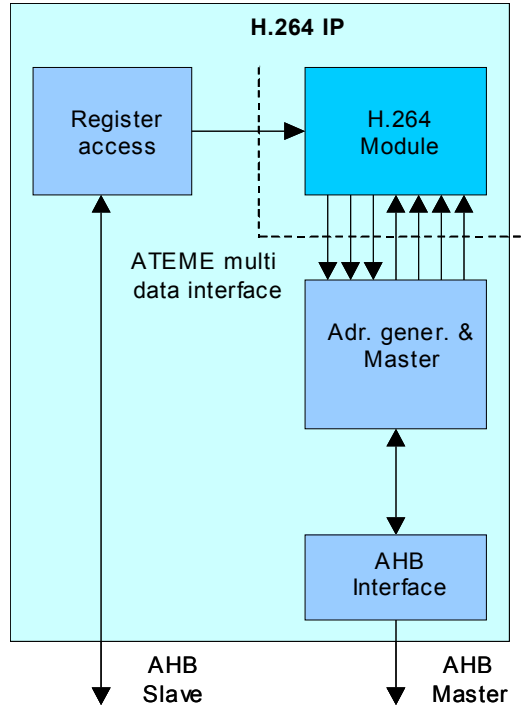
Figure 3- CPU or AHB bus interfaces

The three associated modules:

    – Register access,

    – Address generation and management of data accesses,

    – Master AHB interface,

are delivered in source code with the H.264 module.
Thus the IP can be used either directly, for example by a CPU,
or with all associated modules in a SoC design.
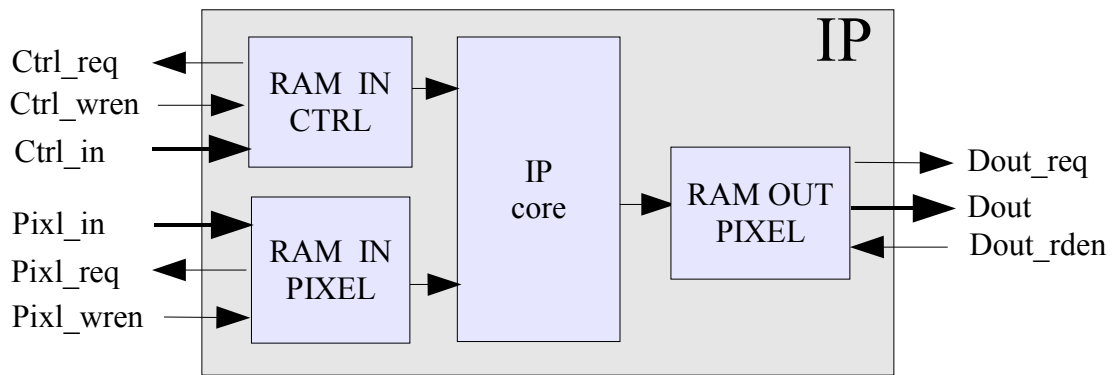
5.2 Hardware interface

Figure 4- Hardware Interfaces

The IP uses a synchronous RAM at the input and output.

The input CTRL RAM receives information about current Mb
  (SbDeblockInfo defined in software interface).

The input PIXEL RAM receives the current macroblock to filter
  (SbDeblockData defined in software interface).

Each input and output memory can be considered as FIFO memory.

When an input buffer request is emitted, a buffer must be filled (*see software interface*). When a macroblock is processed, an output buffer request is asserted, to inform that output data are available.

The above table lists the IP ports.

Signal name	DIR.	Active	Description
Resetn	I	0	Reset signal
Clock	I	↑	IP internal clock
Ctrl_clk	I	↑	Input CTRL RAM clock
Ctrl_wren	I	1	Input CTRL RAM write enable
Ctrl_in	I	-	Input CTRL RAM
Ctrl_req	O	1	Request for CTRL data
Ctrl_DMA_length	O	-	DMA length for CTRL input
Pixl_clk	I	↑	Input PIXEL RAM clock
Pixl_wren	I	1	Input PIXEL RAM write enable
Pixl_in	I	-	Input RAM data bus
Pixl_req	O	1	Request for a new macroblock
Pixl_DMA_length	O	-	DMA length for pixels input
Pixl_DMA_adrs			DMA start address for pixels input
Mbaff	I	1	Mbaff input mode (coming from external register)
Start_Line	I	1	Synchronization flag active when a new line is started
End_Line	O	1	Prevent CPU that a line is finished
End_Picture	O	1	Prevent CPU that a picture is finished
Mbaff	I	1	Mbaff input mode (from external register)
ImageWidth	I	-	Gives the picture width (in pixels, modulo 16, from external register)
ImageHeight	I	-	Gives the picture height (in pixels, modulo 16, from external register)
Dout_clk	I	1	Output RAM clock
Dout_req	O	1	Prevent that filtered Mb is available in Output RAM
Dout_rden	I	1	Output RAM read enable
Dout	O	-	Output RAM bus
Dout_DMA_length	O	-	DMA length for Dout output
Dout_DMA_adrs			DMA start address for Dout output

5.3 Software interface

Little endian byte arrangement must be used.
Three structures of data are required :

- – SbDeblockInfo : Input structure describing the current Mb ,
- – SbDeblockData : Input structure containing pixels of current Mb to process,
- – OutputData : Output structure containing a part of filtered pixels of a Mb.

// Input structures

```
/typedef struct SbDeblockInfo
{
  Uint8 bFlags;               // 0000 000x : Mb type (1 : Field / 0 : Frame )
  Uint8 bQpY;                 // Luminance quantization scales for Mb
  Uint8 bQpC;                 // Chroma quantization scales for Mb
  Uint8 bReserved[3];
  Int8  sbFilterOffsetA;      // Mb Filter offset A
  Int8  sbFilterOffsetB;      // Mb Filter offset B
  Uint8 bBSVertical[16];      // Vertical bS of all 4x4 blocks
  Uint8 bBSHorizontal[16];    // Horizontal bS of all 4x4 blocks
} SbDeblockInfo;

/typedef struct SbDeblockData
{
  Uint8 bCoeffL[256];         // Luminance pixels
  Uint8 bCoeffCr[64];         // Cr pixels
  Uint8 bCoeffCb[64];         // Cb pixels

} SbDeblockData;
```

// Output structure

```
typedef struct OutputData
{
  Uint32 dwxY;                // X coordinate of luma buffer in the picture
  Uint32 dwyY;                // Y coordinate of luma buffer in the picture
  Uint8  blY;                 // Luminance buffer width;
  Uint8  bhY;                 // Luminance buffer height;
  Uint32 dwxC;                // X coordinate of chroma buffers in the picture
  Uint32 dwyC;                // Y coordinate of chroma buffers in the picture
  Uint8  blC;                 // Chrominance buffer width;
  Uint8  bhC;                 // Chrominance buffer height;
  Uint8  bCoeffL[800];        // Luminance pixels : 40 lines * 20 columns max
  Uint8  bCoeffCr[288];       // Cr pixels : 24 lines * 12 columns max
  Uint8  bCoeffCb[288];       // Cb pixels : 24 lines * 12 columns max
} OutputData;
```

In SbDeblockInfo, structure BSVertical[16] is a special case with the following arrangement:
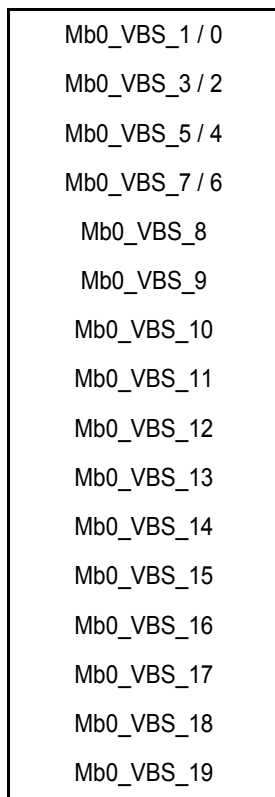
Mb0_VBS_1 / 0
Mb0_VBS_3 / 2
Mb0_VBS_5 / 4
Mb0_VBS_7 / 6
Mb0_VBS_8
Mb0_VBS_9
Mb0_VBS_10
Mb0_VBS_11
Mb0_VBS_12
Mb0_VBS_13
Mb0_VBS_14
Mb0_VBS_15
Mb0_VBS_16
Mb0_VBS_17
Mb0_VBS_18
Mb0_VBS_19

Figure 5- BSVertical[16] structure

BSVertical is 16 BS depth but the first 4 BS are merged two by two, then :
Vertical BS 1 is merged with Vertical BS 0 etc,etc.
No special arrangement for BSHorizontal.

A superblock is composed of two macroblocks.
When MBAFF is negated (0), the structure required by the IP is the macroblock.
One SbDeblockInfo and one SbDeblockData structures are required.

When MBAFF is asserted, the structure required by the IP is a superblock and 2 SbDeblockInfo and 2 SbDeblockData structures are required by the IP.

5.4 System integration

5.4.1 Integration with CPU bus

A typical multi-master system integration of the IP is represented on the figure below.
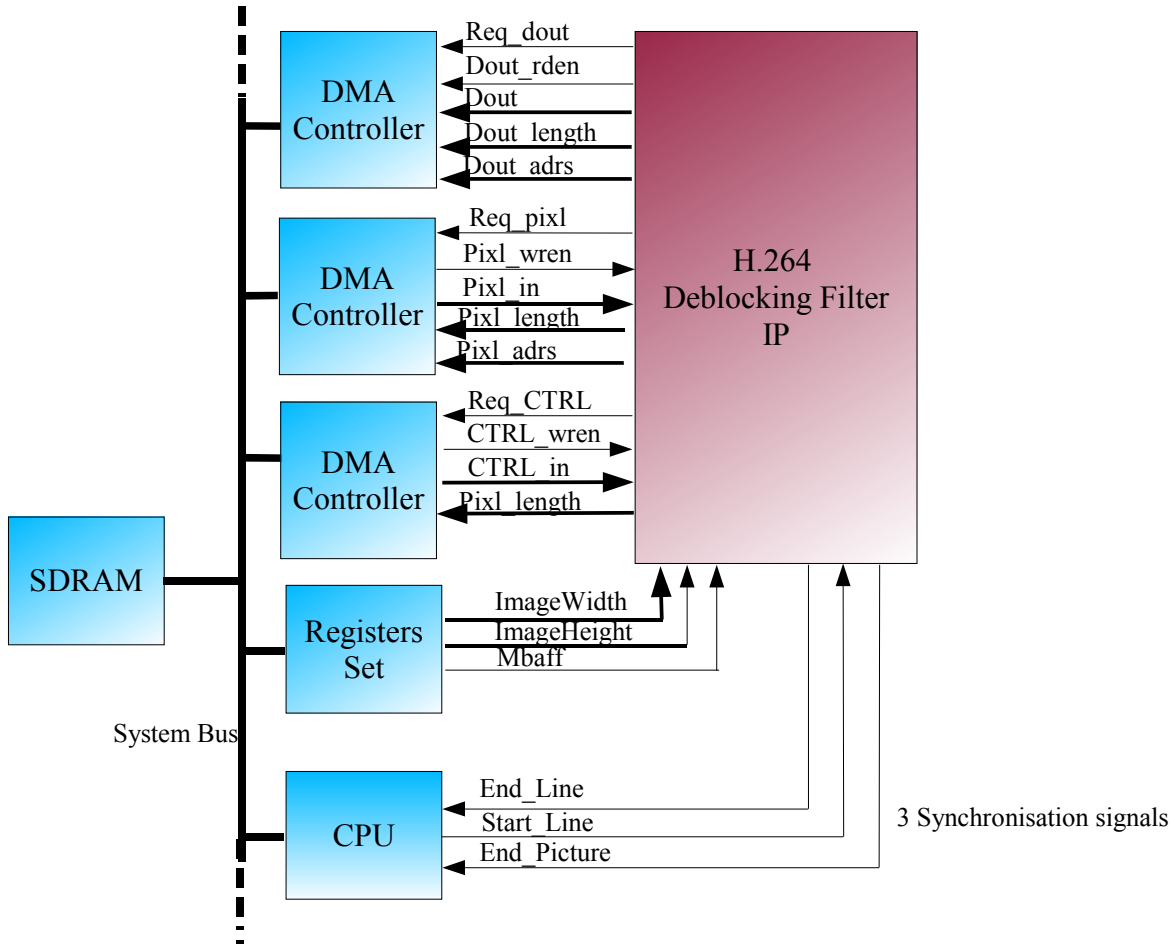

Figure 6- CPU bus system integration

A CPU manages the system. It is in charge of memory allocation, DMA controllers programming, and computation enabling.

A memory base address is loaded by the CPU into the DMA controllers.
A synchronization system used by CPU allows to keep synchronization between the others IP.
Synchronization occurs when end of line occurs.

5.4.2 Integration with AHB bus

The IP is simply configured by registers, then it works in an autonomous way, with Master data acces in external memory, for input and output streams.

5.5 Operating mode

The macroblocks order must be respected, according to the MBAFF mode. For each SbDeblockData structure, an SbDeblockInfo is attached. Length of these structures depends on MBAFF mode.
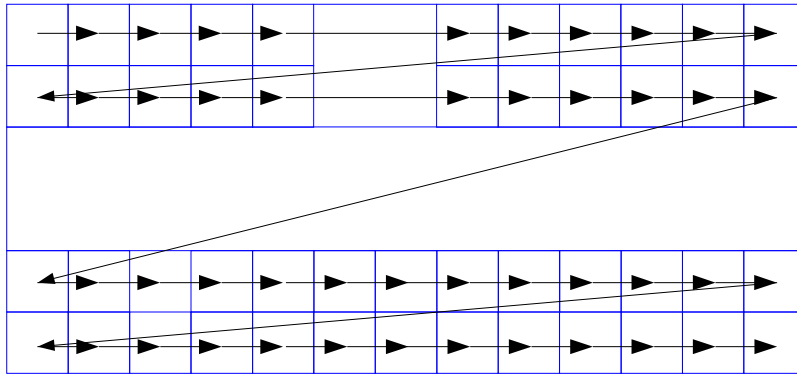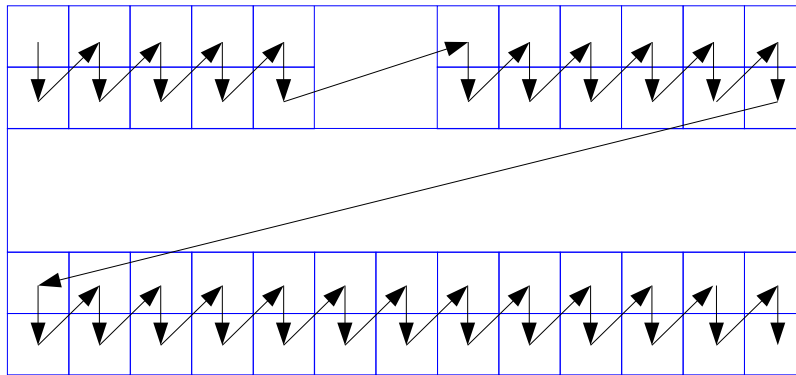

Figure 7- Mb read acces when.Mbaff = 0

Figure 8- Super block read access when.Mbaff = 1

6 Deliverables

The deliverables includes:
- – Loop deblocking filter IP VHDL source code
- – User's manual
- – Simulation files
- – Reference software on PC (windows)

The reference software encodes a video file, and delivers output files, corresponding to the input and output of the IP. This enables to insert known data into the IP, and to compare the IP output with the software output.

An evaluation version of the IP is available.

As option, a PCI board with Altera Stratix II FPGA will soon be available, to ease real-time benches.