OL_H264MCE
Multi-channel HDTV
H.264/AVC Video
Encoder
Rev 1.2

General Description

The OL_H264e core is a hardware implementation of the H.264 video compression algorithm. The core accepts up to the highest resolution HDTV video stream as input and outputs the encoded bitstream. Simple, fully synchronous design with low gate count.

Applications

♦ Digital video recorders.
♦ Video wireless devices.
♦ Video surveillance systems.
♦ Hand held HDTV video cameras.

Features

♦ Fully compatible with the ITU-T H.264 specification.
♦ Proven in FPGA : VGA (640x480) at 30 fps or 720p @ 15 fps in Virtex4-10 demo board with video streamed to Ethernet.
♦ Profile level 4.1, can be decoded by Baseline, Main or Hi Profile decoder.
♦ Supports up to the highest HDTV video resolution (1920x1080 @ 30 fps progressive or 60 fields/s interlaced).
♦ Direct support for both progressive and interlaced video.
♦ Very low operational frequency : from ~1.5 MHz for QCIF @ 15 fps to ~250 MHz for 1920x1080 @ 30 fps.
♦ Single core HDTV support in FPGA : 720p (1280x720) at 30 fps in high end FPGAs (Virtex4, StratixII) . 4 CIF (704x576) at 30 fps in low end FPGAs (Spartan3-4, slowest speed grade).
♦ Supports up to 32 video channel simultaneously.
♦ No CPU required for encoding.
♦ Variable Bit Rate (VBR) and Constant Bit Rate (CBR).
♦ Very low latency (~1.1 ms for VGA @ 30 fps).
♦ Motion vector up to −16.00/+15.75 pixels around the predicted motion vector (-24.00/+23.75 around the origin), down to quarter pixel.
♦ Support for most of intra4x4 and all intra16x16 modes.
♦ Multiple slices support for better error resilience.
♦ Block skipping logic for lower bitrate.
♦ Deblocking filter for better quality.
♦ External memory interface tolerant of high latencies and delays, ideal in a SoC system or in a shared bus with a CPU. The memory interface can be clocked at a different frequency from the core for easier integration.
♦ Supports YUV 4:2:0 video input.
♦ Min Clock speed = 4 x the raw pixel clock speed.
♦ Low gate count : from 145K gates + 133 Kbits of RAM for real time VGA encoding to 195 Kgates + 133 Kbits of RAM for real time 1080p encoding.
♦ Simple, fully synchronous design.
♦ Available as fully functional and synthesizable VHDL or Verilog soft-core.

Functional Description

The OL_H264e core is a hardware implementation of the H.264 video compression algorithm designed to process HDTV progressive video up to 1920x1080 at 30 fps.

Each block of 16x16 pixels is processed in just 1024 cycles. This means that each pixel is processed in just 4 cycles. Consequently, given an uncompressed video stream of resolution X by Y, and frame rate fps, the minimum clock frequency to process a such video stream is :

$$F = 4*X*Y*fps$$

This allows the core to process the video stream at relatively low clock frequencies. For example, HDTV video of 1920x1080 @ 30 fps requires ~250 MHz, whereas VGA video of 640x480 @ 30 fps requires ~37 MHz.

The table below summarizes the relationship between some possible video resolutions and frame rates and the clock frequency of the core.

Resolution	QCIF @ 15 fps	CIF @ 30fps	VGA @ 30fps	1280x720 @ 30fps	1920x1080 @ 30fps
Core freq.	~1.5 MHz	~12.1 MHz	~36.8 MHz	~110.5 MHz	~250.6 MHz

Table 1 Core frequency versus video resolution and frame rate.

A block diagram of the core is shown in Figure 1.

For each block, the intra prediction unit generates a suitable prediction. The intra prediction unit supports most of intra4x4 and intra16x16 modes.

In case of P-frames, the motion estimation unit generates a prediction as well. It examines an area of 32x32 pixels down to the quarter pixel (motion vector from −16.00 to +15.75). Quarter pixel prediction is generated using the tap filters described in the ITU-T specification.

The prediction of each unit is costed using Lagrange multipliers and the best is selected for encoding.

The residual information is calculated from the difference between the current block and the prediction. It is then transformed and quantized to be encoded by the lossless encoding unit.

The transformed, quantized residual is also used to reconstruct a reference frame, which will be used during the encoding of future P-frames. This is achieved by inverse quantization and transform of the residual, that is then added back to the prediction.

Finally, the reconstructed frame is filtered before being stored back in the memory.
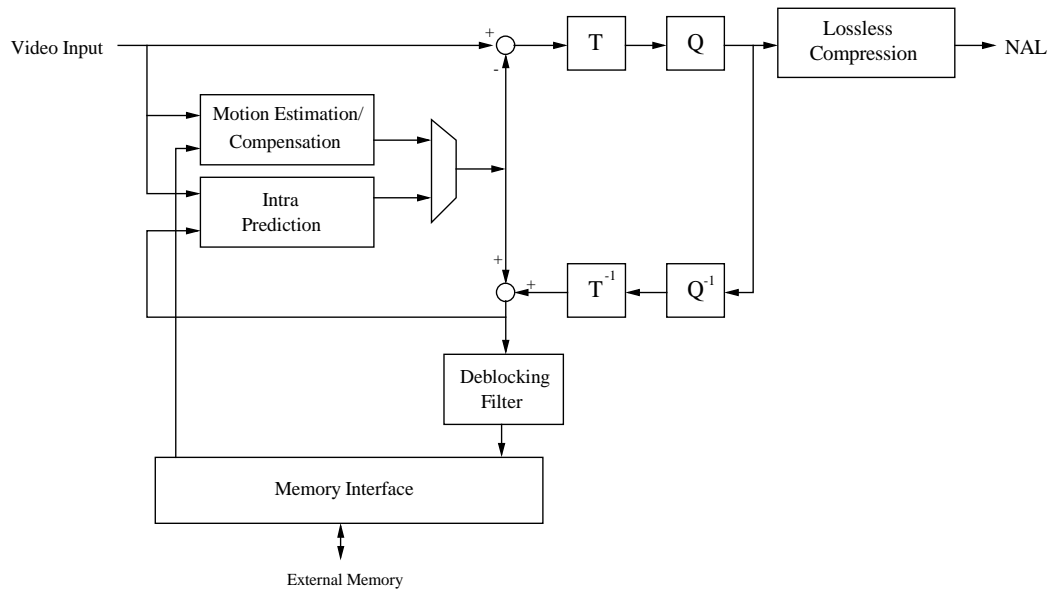
Figure 1 The OL_H264MCE block diagram.

Advantages of the core

Some of the key advantages of the core are discussed further below:

HDTV support

The core is designed to support up to the highest HDTV resolution, 1920x1080 @ 30 fps progressive. This opens a whole new range of applications from high-end video camcorders to high-resolution video surveillance at very low cost.

Low gate count

As it can be seen in the **Performance** section below, the core is designed very efficiently, with a low gate count. This allows 4CIF (704x576) video @ 30 fps to be processed by low end FPGAs at the slowest speed grade as well as 720p (1280x720) video @ 30 fps in high end FPGAs. Thus HDTV 720p real time encoding is possible in FPGAs without multiple core instantiation.

Multi-channel support

The pixel processing capability of the core can be shared among multiple video channels (up to 32). Each video channel can have its own resolution and switching from one channel to the other will happen on the frame boundary. For example, at 250 MHz, up to 6 D1 (720x480) channels can be encoded simultaneously at 30 fps or up to 20 CIF (352x288) channels at 30 fps or a combination of both.
This allows for a very flexible encoding environment where multiple channels with different resolutions and frame rates are encoded by the same small core.

Progressive and interlaced video support

The core can support both progressive and interlaced video, for maximum flexibility.

When encoding progressive video, the output can be compatible with Baseline, Main and Hi profile decoders. When encoding interlaced video, the output is compatible Main and Hi profile decoders.

Extremely low latency

Encoding latency within a single channel is only 16 lines of video. This means that the decoder starts outputting a valid bitstream almost immediately after video is fed in. This feature is vital for a variety of applications such as videoconferencing and mobile videophones. The table below lists the encoding latency for some resolutions and frame rates.

Resolution	QCIF @ 15 fps	CIF @ 30fps	VGA @ 30fps	1280x720 @ 30fps	1920x1080 @ 30fps
Latency	~7.4 ms	~1.85 ms	~1.1 ms	~0.75 ms	~0.5 ms

Table 2 Core encoding latency versus video resolution and frame rate.

No external CPU required

The core can encode video independently, without the support of an additional CPU. This represents a large cost saving compared to solutions that require an external CPU.

Both VBR and CBR supported

The core supports both VBR (Variable Bit Rate) and CBR (Constant Bit Rate). This allows maximum flexibility for the designer.

Flexible memory interface

The core requires access to an external memory via a 32-bit data bus. About 50% of the whole theoretical bandwidth of the memory is actually used by the core.
This interface is designed to be independent from the memory used (i.e. DDR, SDRAM, SRAM, etc.).
More importantly, the memory interface is designed to tolerate high and unpredictable latencies and delays that are typical of a shared memory (i.e. AMBA and/or SoC where the bus is shared with a CPU or other cores).
In addition to this the memory interface can run at a different clock speed from the rest of the core. This simplifies the integration process and can save gates by not forcing the core to be synthesised to a much higher frequency just to be synchronous with the local bus.
This allows, for example, a core running at 37 MHz (encoding VGA @ 30 fps) to be easily integrated in a SoC sharing a 200 MHz bus with a processor.

Error resilience

The core supports multiple slices. This is useful in environments prone to data transmission problems (i.e. mobile phone or other wireless applications) in order to restrict the damage inflicted to the image by transmission errors to a particular slice.

Low data rate features

The core supports two important features for low data rates: deblocking filter and macroblock skipping.
The deblocking filter especially improves the visual quality of the decoded image at low bitrates where the high quantisation noise produces unappealing "blockiness" in an image.

Macroblock skipping greatly reduces the bitrate with minimal effect on the visual quality of the decoded image.

Performances

Performance figures of the OL_H264e core implemented with some particular technologies are shown in the table below. All the features listed above are included in the gate count.

Technology	Approx Area	Speed	Video Throughput
0.13 u LV 0.9V, 125 C	195 Kgates + 133 Kbits RAM Optimised for speed	~ 250 MHz	1920x1080 (1080p) @ 30 fps
0.18 u slow process	145 Kgates + 133 Kbits RAM Optimised for area	~50 MHz	4 CIF (704x576) @ 30 fps
StratixII-C4	13,950 ALUTs + 5 M512 + 57 M4K + 5 DSPs	~125 MHz	1280x720 (720p) @ 34 fps
CycloneII-C6	19,190 Les + 74 M4K + 9 DSPs	~90 MHz	4 CIF (704x576) @ 55 fps
Virtex4-12	13000 slices + 5 multipliers + 33 RAM blocks	~110 MHz	1280x720 (720p) @ 30 fps
Spartan3-4	13000 slices + 5 multipliers + 33 RAM blocks	~50 MHz	4 CIF (704x576) @ 30 fps

Table 3 Performance of the OL_H264e core.

Summary

The combination of low gate count, low operating frequency, and full HDTV resolution support makes this core an application-enabling technology.

The applications of this core range from low power wireless application at relatively low resolution such as mobile phones to HDTV handheld recorders and video surveillance cameras.

Furthermore, the multi-channel capability of the core allows multiple video streams at different resolutions and frame rate to be encoded simultaneously by the same small core. This is ideal in a video surveillance environment where different cameras can be encoded at the same time.

The very small area of this core also allows novel applications such as its direct integration on to a CMOS sensor. This would create an extremely compact intelligent sensor accepting light directly at its input and outputting an H.264 bitstream.

Deliverables

Synthesizable VHDL or Verilog RTL.
Bit accurate C model.
Complete HDL testbench.
Complete data sheet.