



General Description

This core implements a full motion estimation and compensation processor suitable for hardware implementation of modern video compression algorithms such as MPEG-1, MPEG-2, MPEG-4, H263+. The core accepts video in YCbCr 4:2:0 raster as input and outputs motion vectors and transformed, quantized prediction error. Simple, fully synchronous design with low gate count.

Applications

- ◆ Video compression systems.
- ◆ Video wireless devices.
- ◆ Video surveillance systems.

Features

- ◆ Motion vector up to $-16.0/+15.5$ pixels with single and bi-directional search (I, P & B pictures).
- ◆ Glueless interface to SDRAM for frame storage (a single 16 Mbit or 64 Mbit SDRAM chip with 16 bit data bus is sufficient for most applications).
- ◆ The processor includes IEEE-1180 compliant DCT/IDCT and quantizer/dequantizer.
- ◆ Supports YCbCr 4:2:0 raster video input
- ◆ The core outputs motion vectors and quantized DCT of the prediction error.
- ◆ Min Clock speed = 8 x the raw pixel clock speed
- ◆ Very low operational frequency : from ~3 MHz for QCIF @ 15 fps.
- ◆ Simple, fully synchronous design.
- ◆ Available as fully functional and synthesizable VHDL or Verilog soft-core.

Functional Description

Motion estimation and compensation is at the heart of all standard video compression algorithms. This technique is used to exploit the temporal redundancy present in natural video sequences. All the pictures processed by the core are assumed to be divided into macroblocks (normally a block of 16x16 pixels).

Rather than exploiting spatial redundancy only (as in JPEG), temporal redundancy is exploited by transmitting the difference between a macroblock in the current picture and the best matching macroblock in a reference picture.

This is known as unidirectional prediction and is shown in the picture below. The difference between the macroblocks is known as prediction error. The best matching macroblock is defined as the one that minimizes the prediction error within a search area.

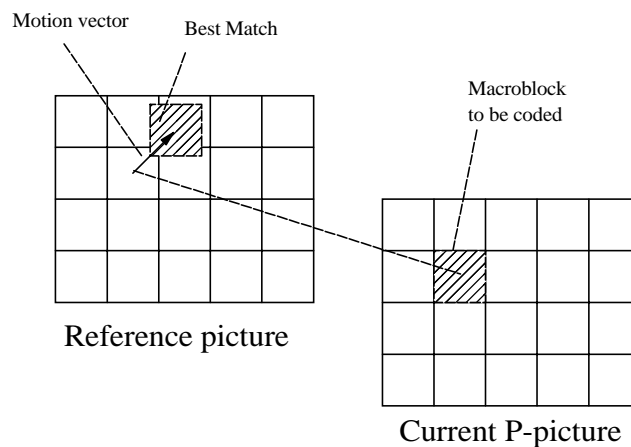


Figure 1 Unidirectional prediction.

The best matching macroblock can be individuated in the reference picture by a displacement vector known as motion vector.

Thus predicted pictures (P-pictures) are pictures that can be reconstructed from a reference picture using the motion vector and the prediction error.

A reference picture can be a previous P-picture or a picture that was compressed using Intraframe coding techniques (I-picture). An I-picture is encoded with a JPEG-like technique which exploits spatial redundancy only (hence the name Intraframe as opposed to Interframe for techniques exploiting temporal redundancy).

Another type of Interframe prediction technique is known as bidirectional prediction and it is shown in the picture below. The prediction error can be the difference between the current macroblock and either of the best matching macroblocks in the reference pictures.

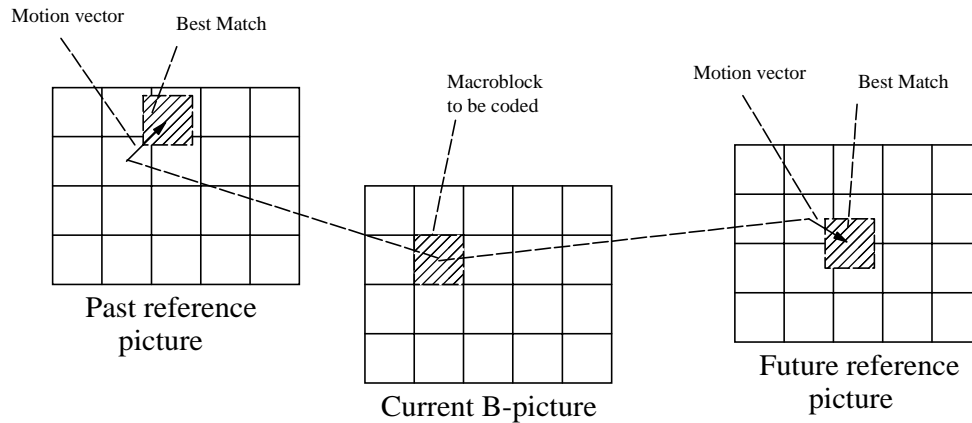


Figure 2 Bidirectional prediction.

Thus bidirectional pictures (B-pictures) are pictures that can be reconstructed from reference pictures using the prediction error and the forward and backward motion vectors.

The diagram below puts I, P and B pictures in context by showing an example of picture sequence.

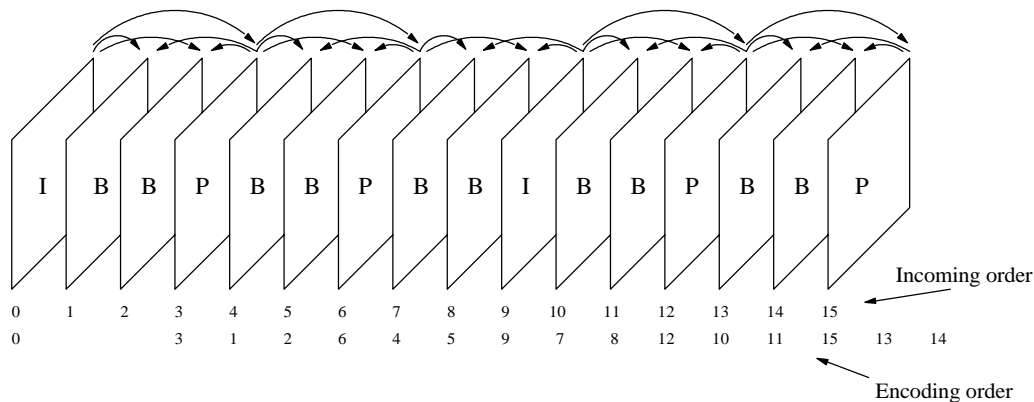


Figure 3 Incoming versus encoding order.

Note that the order of incoming pictures is not the same as the encoding order.

The core supports I, P and B pictures and it is capable of searching for best matching macroblocks within a $-16.0/+15.5$ window at half pixel resolution.

The core is also capable of making various decisions regarding the macroblock it is currently processing:

- Deciding whether to use forward or backward motion compensation (B-pictures only).
- Deciding whether to set the motion vector to zero (P-pictures only). In many cases the prediction error for the zero vector is close enough to the best prediction error. Since non zero motion vectors require additional coding bits, it can be more efficient to code a macroblock using a zero valued motion vector.
- Deciding whether a macroblock is to be coded as intra or inter type macroblock. Sometimes, in case of high temporal activity, intra type of encoding can take fewer bits.

All these capabilities are summarized in the picture below.

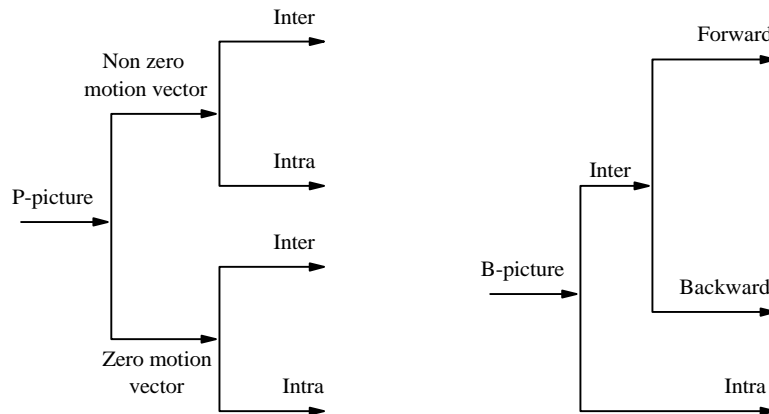


Figure 4 P and B frames block type decision tree.

The quantization matrix can also be changed by a scaling factor (type 1 quantizer), under the control of an external device, in order to achieve bit rate control.

Structural Description

The motion processor is a fully synchronous design clocked by a single clock. The core can be asynchronously reset with an active low reset signal.

Lowering the enable signal can synchronously stall the whole circuit.

The core can be controlled through a processor interface that allows the selection of various options such as image resolution and type of frames.

The operating clock frequency of the core must be at least 8 times the raw pixel clock rate. The latter is calculated multiplying the number of pixel per frame by the number of frames per second. For example, if the video input is 640x480 pixels at 30 frames/s, then the raw pixel clock rate is $640 \times 480 \times 30 = 9.216$ Mpixel/s. The core minimum clock frequency will then be 73.728 MHz.

The table below summarizes the relation between some common video resolutions and frame rates and the clock frequency of the cores.

Resolution	QCIF @ 15 fps	QCIF @ 30 fps	CIF @ 30fps	VGA @ 30 fps	4CIF @ 30 fps
Core freq.	~3 MHz	~6.1 MHz	~24.3 MHz	~73.7 MHz	~97.3 MHz

Table 1 Core frequency versus video resolution and frame rate.

The core requires blocks of 256 pixels at the time. At the same time a macroblock (16x16 pixels) is processed.

Pixels are input in raster sequence in packets of 256 with no blanks or gaps. A simple handshaking mechanism allows to synchronize the core with incoming pixels.

The incoming stream of pixels is stored directly in the SDRAM, ready for subsequent processing.

The core block diagram is shown below.

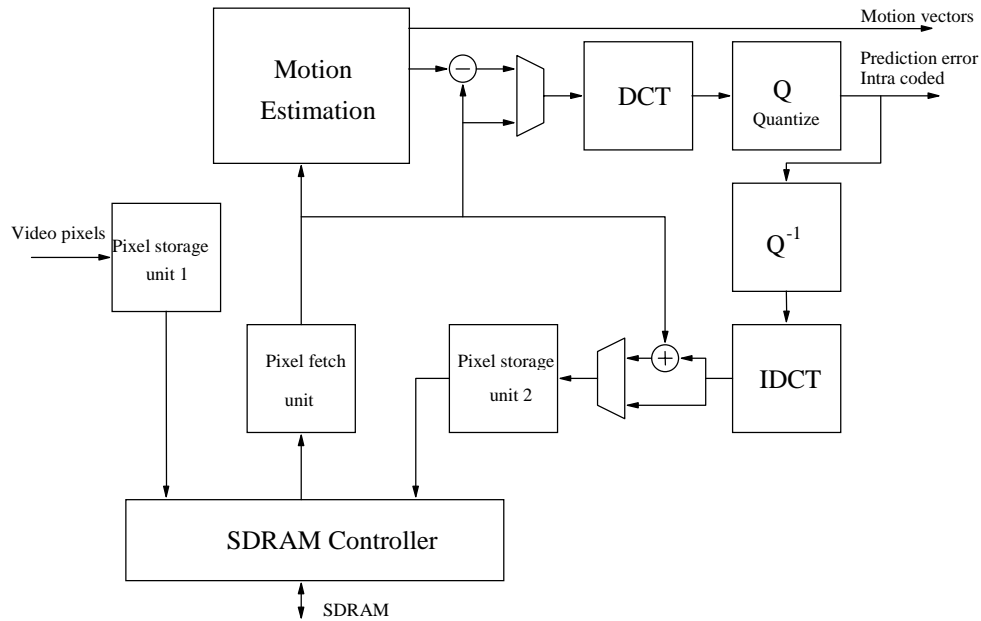


Figure 5 OL_Motion block diagram.

All the submodules in the core share access to the SDRAM through the SDRAM controller. The SDRAM acts as a frame buffer.

A single 16 or 64 Mbit SDRAM chip with 16 bit wide data bus is sufficient as frame buffer.

Operation and data flow

The pixel storage unit 1 works continuously, storing incoming pixels in the SDRAM buffer.

During the processing of I-pictures macroblocks are fetched, DCT transformed and quantized. The result is output by the core. This information is also used to reconstruct a reference picture for subsequent motion estimation.

This is done by emulating the behaviour of a decoder. The transformed, quantized coefficients are dequantized and the IDCT is applied. The reconstructed macroblock is then stored back to the SDRAM.

The emulation of the decoder behaviour ensures that both encoder and decoder work on the same reference pictures.

Currently the OL_Motion supports the MPEG-4 type 1 quantizer. Support for type 2 quantizer (basically a subset of type 1 quantizer) will be added soon.

During the processing of P-pictures, macroblocks are fetched and searched for the best matching macroblock within the past reference picture. The prediction error or the intra-coded macroblock is then transformed and quantized and the result output together with the motion vector.

As for the case of I-pictures, the macroblock is reconstructed by reversing the previous steps. If the macroblock was not intra coded, the reference macroblock is added to the reconstructed prediction error. The resulting macroblock is then stored as part of the reference picture.

During the processing of B-pictures, macroblocks are fetched and a bi-directional search for the best matching macroblock is performed within the past and future reference picture. The

OL_Motion Motion processor

prediction error or the intra-coded macroblock is then transformed and quantized and the result output together with the motion vectors.

Since B-pictures do not form reference pictures, the macroblocks are not stored back to SDRAM.

Output of the core

The motion processor outputs blocks of pixels or prediction error already DCT transformed and quantized. It also outputs a forward or backward motion vector.

All this pre-processed information enormously simplifies the design of real time video compressors in a variety of standards such as MPEG-1, MPEG-2, MPEG-4, H263+.

It is estimated that this core can provide up to 90% of the computational complexity requirement for real time video encoding with one of the algorithms listed above.

The user needs only to implement the lossless stage of the chosen video compression algorithm (run length encoding, Huffman encoding, etc.). This represents only a small fraction of the computational requirement and can in many cases be handled in software by an embedded processor.

Performance

Preliminary performance figures of the core implemented with some particular technologies are shown in the table below.

Technology	Approx Area	Speed	Video Throughput
ASIC 0.18 μ	33 Kbytes+17.5Kbits RAM	~ 100 MHz	704x576 (4CIF) @ 30 fps
Virtex II	~2900 slices + 10 Multipliers + 12 RAM Blocks	> 70 MHz	640 x 480 @ 28 fps

Table 2 Performance of the OL_Motion core.

Deliverables

Synthesizable VHDL or Verilog RTL.

Bit accurate C model.

Complete HDL testbench.

Complete data sheet.

Ocean Logic Pty Ltd

PO BOX 768 - Manly NSW 1655 - Australia

Tel: +61-2-99054152 Fax: +61-2-99050921

E-Mail: info@ocean-logic.com URL : <http://www.ocean-logic.com/>