High performance VLSI implementation for H.264 Inter/Intra prediction

Mythri Alle, J Biswas, S. K. Nandy
CAD Lab, Indian Institute of Science, Bangalore
Bangalore, 560 012, India
{mythri@cadl, jayanta@cadl, nandy@serc} .iisc.ernet.in

***Abstract*— We provide a hardware realization of motion compensation and reconstruction(MCR) module for H.264 baseline profile. We synthesize the MCR module using UMC library in 0.13 μ CMOS technology. Our implementation occupies an area of 94756 gates and operates at a frequency of 250MHz.**

I. Introduction

H.264 delivers high resolution video even at low bit rates. Inter and Intra prediction play an important role to achieve high compression ratios. To meet the high throughput and computation complexity of inter prediction, hardware implementations are a viable option. In this paper we propose a complete hardware solution for the motion compensation and reconstruction (MCR) module.

The overall design of our implementation is shown in Fig 1. The MCR module takes the prediction modes and its parameters as the input from the bit-stream parser and residual data from IDCT (inverse discrete cosine transform) module. Residual data is added to the predicted frame, computed by the MCR module and passed on to the deblocking filter. The communication between IDCT and MCR module is through FIFO. Parser and MCR module share two buffers in a mutually exclusive manner and access is switched for every macroblock.

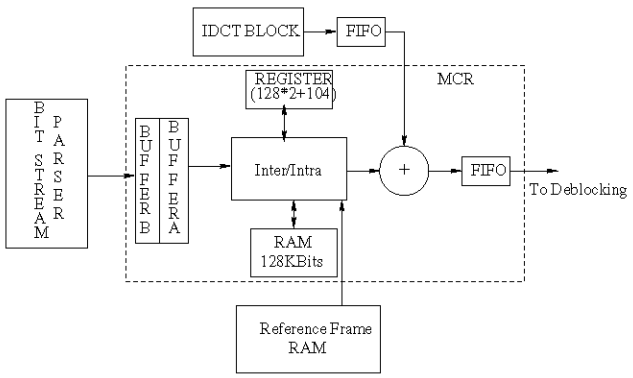

Fig. 1. Positions of Interpolation

II. Inter Prediction

H.264 allows to capture a very fine motion at quarter pixel resolution using a 6-tap FIR filter for interpolation of accurate sub-pixel values. This requires fetching of large number of pixels from the reference frame. Very often the pixels fetched for the current block can be reused for the next block.

In the proposed design inter prediction has a fetch unit, a compute unit and a cache (shown in Fig 2). These units function in parallel.

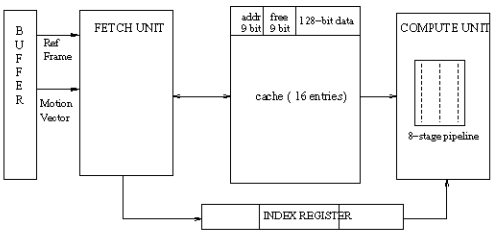

Fig. 2. High level Block Diagram of Inter prediction

A direct mapped cache of 16 entries is used to buffer the blocks of reference frame for further use. Each cache entry holds one (4 × 4) block data, 9-bit sub-macroblock (concatenation of frame number and offset of block in the frame) number and a 9-bit free entry. 4 least significant bits of sub-macroblock number are used to index cache.

In our design we assign a maximum of 9 cycles for fetch unit and 4 cycles for compute unit. Fetch unit computes macroblock addresses required for the prediction of a (4 × 4) block. This unit fetches the blocks that are not available in the cache. A block is fetched only if the cache entry is marked free, which means the block present in the cache entry is no longer required. If cache entry is not free, fetch unit idles till the entry becomes free. We do not perform out of order fetch as it complicates the control logic and benefits are not significant. Fetch unit also updates the index register with the indices, at which the data is available in the cache.

Compute unit calculates the predicted values by performing motion compensation using a eight stage pipeline for luma samples and four stage pipeline for chroma samples.

Fetch unit and compute unit synchronize at each (4 × 4) block. One (4 × 4) block requires a maximum of 9 (4 × 4) blocks from reference frame for the prediction. Since the index is 4-bit, each entry in index register is 36-bit. Index register can hold upto three such indices.

1-4244-0763-X/07/$20.00 ©2007 IEEE

III. Intra Prediction

In H.264 intra prediction is either done per macroblock (Intra_16×16) or per (4 × 4) (Intra_4×4). For chroma samples, prediction is done per macroblock.

In the proposed implementation intra prediction is divided into three phases. First phase is the init phase, where the data required for the intra prediction is obtained. Second phase called the compute phase uses the data fetched by the first phase and performs the computation. In the final phase data is passed on to deblocking filter. Final phase of intra prediction retains the required pixels that are not passed through deblocking filter for prediction of other blocks.

Intra_16×16 prediction requires last row of top , top left and top right macroblocks (B, C, D in Fig 3) and last column of left macroblock (E in Fig 3). In the init phase this data is fetched from RAM. *Intra_4×4* predicts one (4 × 4) block at a time. Each (4 × 4) block requires last row of top, top right and top left of (4 × 4) block (B1,C1,D1 in Fig 3) and last column of left (4 × 4) block (E1 in Fig 3). of (4 × 4) block. This data is available in register, which is stored by the write phase. Chroma prediction needs last row of top macroblock and last column of left macroblock which are fetched from RAM.

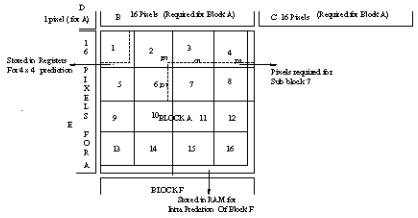

Fig. 3. Blocks required for Intra prediction

Compute phase, performs computation on data depending on the prediction mode. In the write phase the pixel values are stored into a FIFO, which is read by the deblocking filter. The FIFO is 128-bit wide, and holds one row of the macroblock (16 pixels). The blocks are processed in raster scan order as shown in Fig 3. The data is buffered till an entire row of the (4 × 4) block is decoded and is then written into FIFO.

IV. Results

The proposed design is synthesized using synopsys design vision. We use UMC [3] library in 0.13μ technology. Synopsys design vision does not report number of gates. We calculate approximate gate count by normalizing the area with average area of a two input basic gate.

MCR requires 95597 gates at 250 MHz. The gates reported includes the gates required by the cache, FIFOs and the buffer shown in Figure 1. RAM should be able to store pixels of one row of the frame. We use RAM of 128 Kbits which allows us to process very large frame sizes. The split up of number of gates for each unit is shown in Table I.

The average number of clock cycles required for inter prediction of various sequences are tabulated in Table II. The table also gives the percentage of *reuse* of the (4 × 4) blocks using the cache. On an average we reuse about 50% of the blocks. The average number of cycles without cache (shown in Table II) is the number of cycles required to fetch the blocks from RAM without taking into account any idle cycles. Average cycles reported for the proposed design include all idle and computation cycles.

In the intra prediction number of cycles taken per macroblock depend on the prediction mode. The maximum number of cycles required per macroblock is 160 and minimum is 100 cycles. The average number of cycles taken by various sequences is tabulated in Table III. The number of cycles required per each macroblock, is sufficient to meet requirements of high resolution large size frames like 1080HD.

TABLE I
Inter prediction area

Total Gates	Inter	Intra	FIFO (Deblock)	FIFO (IDCT)	Cache
94756	50866	26034	3072	768	14016

TABLE II
Average number of cycles for Inter prediction

Sequence Name	Avg cycles (proposed)	Avg fetch cycles (No Cache)	reuse of blocks %
AUD_MW_E [5]	143	159	57.1
CAPM3_SONY _D [5]	144	154	51
SVA_BA1_B [5]	131	113	55

V. Conclusion

This paper proposes hardware implementation for MCR module of H.264. MCR module is computationally complex and also memory bandwidth required is high. If these issues are not addressed properly it is not possible to meet throughput requirements of high resolution video. The proposed implementation addresses these issues and the area occupied is affordable for practical implementations. MCR module occupies 94756 gates and requires 128Kbits of RAM and operates at a clock frequency of 250MHz. The throughput offered by the architectures is sufficient to operate at level 5 of H.264 [4].

References

[1] J. Lee, S. Moon and W. sung, " H.264 Decoder optimization using SIMD instructions" IEEE Asia Pacifi c Conference on Circuits ans systems.

[2] K. Sato and Y.Yagasaaki "Adaptive MC interpolation for memory access reduction in JVT video coding" Proc of Seventh Intl. Sym on Signal processing and its applications.

[3] UMC 0.13 library *http://umc.com*

[4] Joint Video Team(JVT) of ISP/IEC MPEG and ITU-T VCEG, doc JVT-G050.

[5] *http://ftp3.itu.ch/av-arch/jvt_site/draft_conformance*

TABLE III
Number of cycles for Intra prediction

seqname	Avg cycles	Max Cycles
Foreman [5]	155	280
NL1_sony_D [5]	142	254
BA1_Sony_D [5]	144	268
SVA_BA1_B [5]	120	188