PLATFORM ARCHITECTURE DESIGN FOR MPEG-4 VIDEO CODING

Wei-Min Chao, Yung-Chi Chang, Chih-Wei Hsu, and Liang-Gee Chen

DSP/IC Design Lab
Graduate Institute of Electronics Engineering and Department of Electrical Engineering
National Taiwan University
1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan
{hydra,watchman,jeromn,lgchen}@video.ee.ntu.edu.tw

ABSTRACT

This paper presents a cost-effective platform architecture design for MPEG-4 video coding. A fast motion estimator architecture supporting predictive diamond search and spiral full search with halfway termination is implemented to make good compromise between compression performance and design cost. An efficient block-level scheduling for texture coding engine is employed to reduce the hardware cost. Both these key modules are integrated into an efficient platform in hardware/software co-design fashion. With high degree of optimization in both algorithm and architecture levels, a cost-efficient video encoder is implemented. It consumes 256.8mW at 40MHz and achieves real-time encoding of 30 CIF (352x288) frames per second.

1. INTRODUCTION

The emerging MPEG-4 standard becomes the main technique of the mobile devices and streaming video applications such as smart phone and handheld PDA devices. In these applications, low power, low cost, high flexibility, and high performance are four key issues to implement the video coding system for real-time specification and future applications.

Several MPEG-4 video chips have been reported in the past. To satisfy rich functionality of future multimedia, some are implemented in software [1] [2] based on the low-power DSP platform. They have highest flexibility but to achieve the real-time performance under the limited resources, the fast algorithms of motion estimation (ME) and discrete cosine transform (DCT) are applied and the compression quality degrades at the same time. Some [3] use the dedicated hardware methodology to achieve low power and low area cost. Lack of potential for future modification of advanced algorithms and higher design effort are disadvantages. Hence, some [4] [5] adopted the hybrid software/hardware co-design to compromise the performance and flexibility for complex coding flow.

In this paper, a RISC-based platform with hardware accelerators is presented to implement MPEG-4 video encoding algorithms. The optimization in both algorithm and architecture level is applied. Not only the key components but also the connection optimization are discussed in this paper. First, the coding system is divided into three main subsystems, motion, texture, and bitstream, which are optimized by observing the relationship at the algorithm and architecture level. In motion subsystem, the hybrid motion estimator supporting both predictive diamond search and spiral full search with halfway termination for real-time or high compression quality applications are proposed to reduce the dominant cost in the typical coding system. In the texture subsystem, the efficient interleaving schedule and substructure sharing technique among quantization and DC/AC prediction are proposed [6]to reduce the cost further. In the bitstream subsystem, to handle the complex bitstream syntax and avoid inefficient bit-level storage, the hardware/software co-operations scheme is applied for the bitstream generation. After the optimization described above, a compact MPEG-4 video encoder chip is implemented and occupies the 5.02x5.13 mm^2 in 4-layer-metal, 0.35 μm CMOS standard cell process. It is much smaller than any MPEG-4 video encoder previously reported and achieves the same performance. It consumes 257 mW at 40MHz operation and achieves real-time encoding of 30 CIF (352x288) frames per second.

2. MPEG-4 VIDEO ENCODER ARCHITECTURE

Fig.1 depicts the proposed platform-based system with hardware accelerators to achieve a MPEG-4 video coding functionalities. RISC takes responsibility for macroblock level hardware scheduling, coding mode decision, motion vector coding, and other high level procedures. Other hardware accelerators improve the system performance by parallel processing according to the parallelism of algorithms. Motion estimator (ME) carries out motion estimation with the search range -16.0 to +15.5 pixel unit. Motion compensator (MC) interpolates pixels in reference frames into compensated blocks by specified motion vectors. Texture block engine (TBE) carries out discrete cosine transform (DCT), inverse cosine transform (IDCT), quantization (Q), inverse quantization (IQ), and AC/DC prediction on texture pixels in block unit. Bitstream generator (BTS) produces headers, motion information, and texture information in the format of variable length codes. In addition, share memory builds the direct channels from MC to TBE and BE to BTS to decrease the traffic of the data bus. Sequencer (SEQ) handles the pixel by pixel scheduling of these share memory without bothering RISC. DMA involved in dedicated commands efficiently generates the proper addresses issued by RISC or SEQ. Four global bus channels are used in this system. First, RISC bus broadcasts controlling information to each hardware modules. After applying operations issued by RISC, hardware modules respond processed side information on which RISC depends to decide the coding modes for macroblocks. At the same time source, reference, and reconstructed frames required by hardware modules are passed through DMA and then provided by DATA bus. Hardware modules efficiently access this data automatically according

0-7803-7750-8/03/$17.00 ©2003 IEEE

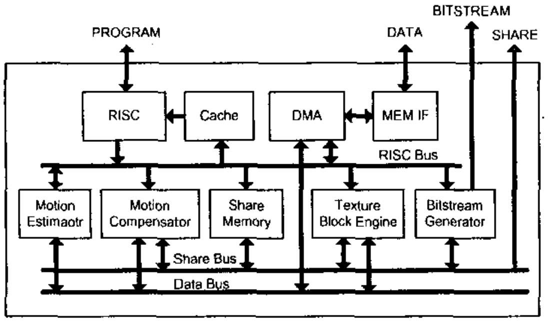

Fig. 1. System Architecture

to pre-determined scheduling. These parts are integrated into a single chip with the firmware stored outside for programmability through PROGRAM bus after taped out. SHARE bus can transfer DCT coefficients, quantized coefficients, or other immediate information in the testing mode. The developing time and effort can be reduced through this information.

3. MOTION ESTIMATION

3.1. Algorithm

Motion estimation is the key technique of video coding and can reduce the temporal redundancies of sequences to make compression efficient. In all algorithms of motion estimation, full search block matching (FSBM) algorithm is well known and commonly used in the video coding system because of its good performance and regularity. However, the huge computational power is required to meet the real-time application. Dedicated hardware is usually employed through the parallel processing and it causes a large cost design. Besides, the encoder should decide the optimal prediction blocks among the various block sizes and in the finer pixel precision in the MPEG-4 standard. It makes the system difficult to handle these operations under acceptable cost and maintain the same compression quality. To meet the requirement of various applications under the acceptable cost, we adopt two kinds of algorithms for the motion estimation of 16x16 block size at integer-pixel precision. One is the spiral full search with halfway termination (called fast full search, FFS) which can achieve the same compression efficiency as the full search algorithm. The other is the diamond search starting from the predictor derived from neighboring macroblocks (called predictive diamond search, PDS) and it meets the real-time specification under the visual quality degradation. Afterwards, the hierarchy scheme is applied for the motion estimation for four 8x8 pixels blocks in a macroblock around +2 to -2 positions of the previous best motion vector. The half-pixel refinement is also applied for all found integer-pixel motion vectors. Fig.2 depicts the whole stages of motion estimation and describes as follows. The predictor is determined from neighboring macroblocks. The PDS mode or FFS mode is employed to find the integer pixel motion vectors. The half-pixel refinement is applied around the motion vector found in the phase 2. For four 8x8 pixel blocks in a macroblock, the spiral search around -2 to +2 is applied to obtain four optimal motion vectors. Four times of half-pixel refinement is applied around the motion vectors found in the previous phases.

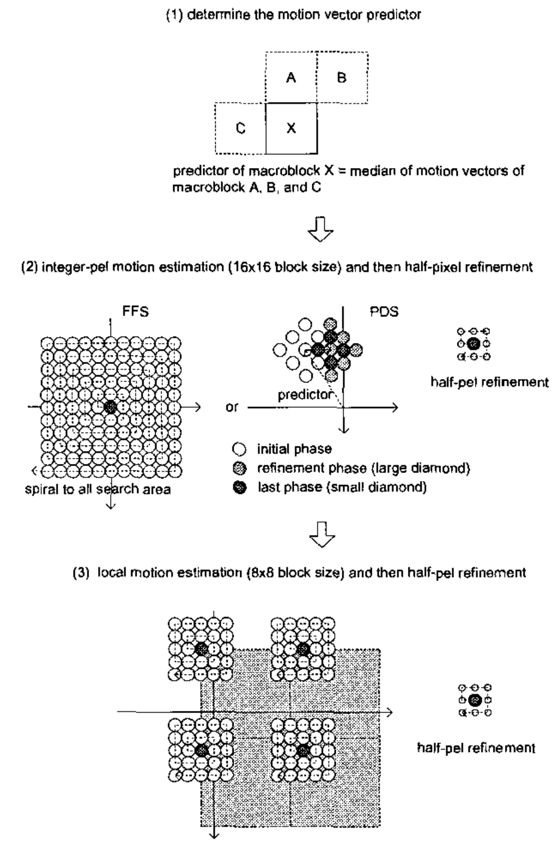
(1) determine the motion vector predictor
A
B
C
X
predictor of macroblock X = median of motion vectors of macroblock A, B, and C
(2) integer-pel motion estimation (16x16 block size) and then half-pixel refinement
FFS
PDS
or
predictor
half-pel refinement
spiral to all search area
○ initial phase
◍ refinement phase (large diamond)
● last phase (small diamond)
(3) local motion estimation (8x8 block size) and then half-pel refinement
half-pel refinement

Fig. 2. Algorithms of motion estimation

3.2. Architecture

Fig.3 depicts the hardware architecture of the motion estimator supporting PDS and FFS. This architecture mainly includes three processing stages and two buffers to store current MB and the search window. Before performing motion estimation, the video coding system transfers data from external memory into these buffers to eliminate the bus bandwidth for calculating of sum of absolute difference in the following. Meanwhile, the adder tree accumulates the sum of the pixels in the current MB to save it into a register for the mode decision in the future. To speed up the data loading and reduce the bus traffic, the search window buffer can be loaded using column-by-column data-reuse scheme. After motion estimation starts, the pattern generation (PG) stage generates the valid candidate positions. Then these positions are passed through the FIFO stage and fetched by the distortion calculation (DC) stage. The DC stage is responsible for calculating SAD of candidate positions and finds the minimum one. The accumulation comparison elimination (ACE) unit performs the PDE algorithm to

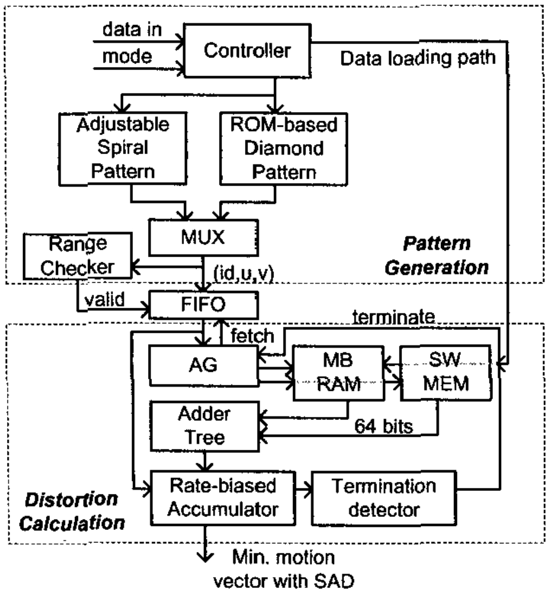

Fig. 3. Architecture of motion estimator

reduce the computational complexity.

3.3. Performance

MPEG-4 standard only defines the decoder and left how to implement the encoder an open problem. Many different algorithms can be adopted alternatively under the different conditions of the cost, bit-rate, and picture quality. In our point of view, we use a novel motion estimator to support PDE or FFS algorithms to compromise the compression performance and the design cost. The PDS mode can satisfy the real-time specification while the FFS mode can achieve the same compression quality as MPEG-4 software verified model (VM)[7]. To explore the degradation in the PDS mode, four sequences with different features are used as test patterns. The average difference between PDS and VM in PSNR is only 0.136 dB and the maximum PSNR drop through the testing sequences is only 0.618 dB. Even in the frames whose the difference in PSNR are maximum, it is still indistinguishable between these two in subject view. While encoding in the FFS mode, the PSNR and bit-rate of the reconstructed frames are almost the same as that encoded by VM. The average PSNR are even better than 0.00625 dB. The general R-D curves for testing sequence are simulated and shown in Fig.4.

4. CONFIGURABLE PLATFORM PROTOTYPING

A configurable platform is used to verify the functionality of our architecture design. This prototyping board is connected through the PCI interface to the host computer. Four separated memory with DMA modules are used to handle PROGRAM, DATA, SHARE, and BITSTREAM bus from our design. An arbiter is responsible for the memory access through PCI and memory. The MPEG-4 video encoder design is synthesized and placed on the FPGA chip. The program to run in RISC processor is compiled to machine codes by the host computer and then sent to the program memory. Raw image data is transferred from the host computer to the frame memory on the prototyping board. Video encoding is processed concurrently. Afterwards, bitstream data are stored in the bitstream memory and then read from the host computer. Besides, the share memory can record the immediate information for debugging in the testing mode.

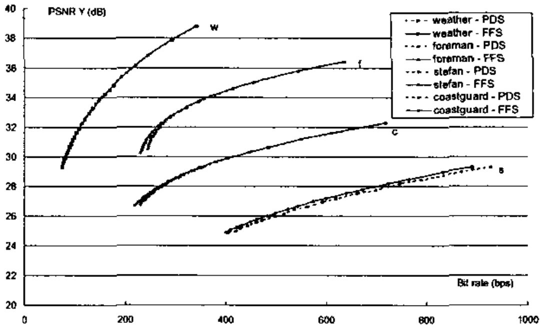

Fig. 4. RD curves with PDS and FFS modes

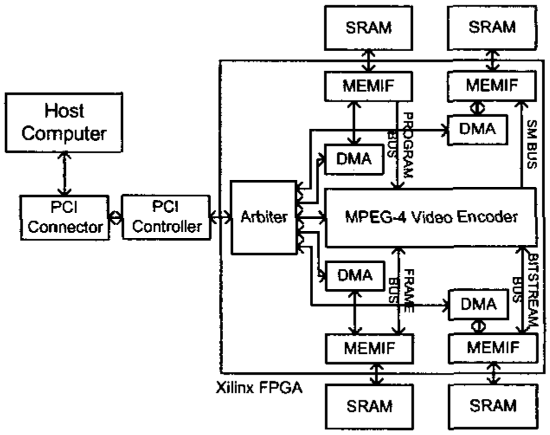

Fig. 5. Reconfigurable platform

5. IMPLEMENTATION

Fig.6 shows a micrograph of the encoder LSI and Table 1 depicts its characteristics. The LSI contains 828K transistors and is fabricated on a 5.02 x 5.13 mm^2 with 0.35 μ m and single-poly quadruple-metal CMOS process. The chip is tested and works successfully. The supply voltage is 3.3V and consumes 256.8mW at 40MHz working frequency. Table 2 shows the number of transistors, the area, and the size ratio to the LSI of each unit.

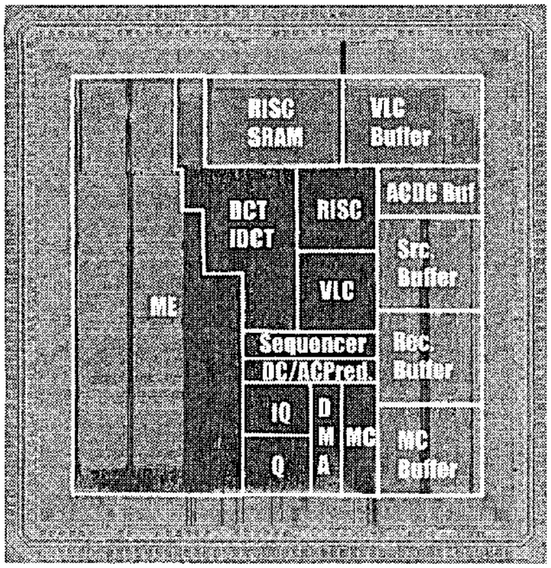

Fig. 6. Micrograph of this encoder

Table 1. Characteristics of the encoder chip

Technology	TSMC 0.35 μm 1P4M CMOS
Die Size	5.02 x 5.13 mm^2
Transistor count	828,692 trans.
On-chip memory	39,080 bits
Off-chip memory	2,027,527 bits
Clock frequency	40 MHz
Voltage	3.3V
Power consumption	256.8mW
Package	208 CQFP
Function	MPEG-4 SP@L3 video encoder
Motion estimation algorithm	Predictive diamond search & Search range -16.0 to +15.5 & Advanced prediction mode
Encoding complexity	352 x 288 at 30 fps

Table 2. Cost distribution

	Trans.	Area	Size ratio
	(k)	(mm^2)	(%)
ME	288	5.8	22.6
MC	53	0.3	1.2
DCT/IDCT in TBE	126	1.6	6.2
Q/IQ in TBE	64	0.7	2.9
ACDCP in TBE	22	0.8	3.0
RISC	112	1.8	7.0
DMA	19	0.3	1.2
VLC	95	0.7	2.7
Share MEM	68	2.8	10.9
Others (PAD etc.)	49	10.9	42.3
Total	829	25.8	100.0

6. CONCLUSION

In this paper, an efficient platform architecture design with hardware accelerators for MPEG-4 Simple Profile@Level 3 video encoder is proposed. The hardware module is written in Verilog and verified in modular fashion while the firmware is written in assembly. The co-design and co-simulation is employed to reduce the development time. Also, the efficient reconfigurable FPGA prototyping system is exploited to verify the functionality. With cost-effective hybird motion estimation and interleaving DCT/IDCT hardware modules, the system are implemented into 5.03x5.13 mm^2 die size with 0.35 μm CMOS technology process. It works at 40MHz and consumes 256.8mW to meet the real-time encoding specification.

7. REFERENCES

[1] A.Hatabu and et al., "QVGA/CIF Resolution MPEG-4 Video Codec Based on a Low-Power and General Purpose DSP," *SIPS*, vol. 23, pp. 27–49, 2002.

[2] T.Kumura and et al., "VLSI DSP for Mobile Applications," *IEEE Signal Processing Magazine*, vol. 23, pp. 27–49, 2002.

[3] M. Takahashi and et al., "An MPEG-4 Video LSI with an Error-Resilient Codec Core Based on a Fast Motion Estimation Algorithm," *IEEE International Solid-State Circuits Conference*, vol. 35, pp. 1713–1721, Feb 2002.

[4] M. Takahashi and et al., "A 60-MHz 240-mW MPEG-4 Videophone LSI with 16-Mb Embedded DRAM," *IEEE Journal of Solid-State Circuit*, vol. 35, pp. 1713–1721, Nov 2000.

[5] J.H. Park and et al., "MPEG-4 Video Codec on an ARM core and AMBA," *MPEG-4 Proceedings of Workshop and Exhibition*, vol. 35, pp. 95–98, June 2001.

[6] C.W. Hsu, W.M. Chao, Y.C. Chang, and L.G. Chen, "Cost-Effective Scheduling Of Texture Coding For MPEG-4 Video," *IEEE International Conference on Multimedia and Expo(ICME'02)*, Aug 2002.

[7] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, pp. 19–31, Feb 1997.