Journal of VLSI Signal Processing 42, 7–19, 2006
© 2005 Springer Science + Business Media, Inc. Manufactured in The Netherlands.
DOI: 10.1007/s11265-005-4159-8

Platform-Based MPEG-4 SOC Design for Video Communications

YUNG-CHI CHANG, WEI-MIN CHAO, CHIH-WEI HSU AND LIANG-GEE CHEN
DSP/IC Design Lab, Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Received March 31, 2004; Revised August 10, 2004; Accepted September 1, 2004

Published online: 29 January 2006

Abstract. An MPEG-4 video coding SOC design is presented in this paper. We adopt platform-based architecture with an embedded RISC core and efficient memory organization. A motion estimator supporting predictive diamond search and spiral full search is implemented for compromise between compression performance and design cost. The proposed data reuse scheme reduces required memory access bandwidth. For texture coding path, an interleaving DCT/IDCT scheduling with substructure sharing technique is proposed. Several key modules are integrated into an efficient platform in hardware/software co-design fashion. The cost-efficient video encoder SOC consumes 256.8 mW at 40 MHz and achieves real-time encoding of 30 CIF (352×288) frames per second.

Keywords: MPEG-4 video, video encoder, platform-based architecture

1. Introduction

MPEG-4 standard is becoming the main technique of the mobile devices and streaming video applications such as smart phone and handheld PDA devices. The improved coding efficiency and advanced functionalities of MPEG-4 come with much higher computational complexity compared with previous standards. Several MPEG-4 video chips have been reported. To satisfy rich functionality of future multimedia, some are implemented in software [3] based on the low-power DSP platform. They have highest flexibility but degraded quality due to the fast algorithms of ME and DCT. Some [4] 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 [5, 6] adopted the hybrid software/hardware co-design to compromise the performance and flexibility.

According to the computational complexity analysis reported in [1] and [2], the dominating computation-intensive tasks in MPEG-4 core profile coding are motion estimation(ME) and shape encoding, which together contribute more than 90% of the overall complexity. For simple profile without shape coding tools, ME becomes the most significant one. It belongs to highly regular low-level task, and a huge amount of data access through frame buffer is also required. So, dedicated architectures and local buffers are heavily relied for efficient implementations and data access reduction. For other coding tasks, including DCT/IDCT, Q/IQ, and MC, dedicated architectures can be adopted for these highly regular tasks. Programmable architectures are suitable for the other less-demanding but high-level task, such as system control.

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 and memory organization are discussed in this paper. The whole system is divided into three main subsystems. 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 texture subsystem, the efficient interleaving schedule and substructure sharing technique among quantization and DC/AC prediction are proposed [13] to reduce the cost further. In 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. By applying these optimization approaches, a low cost and high performance MPEG-4 video encoder SOC is implemented.

This paper is organized as follows. In Section 2, the whole system architecture will be explored. The system memory organization will be discussed in Section 3. In Section 4, we will present the algorithm, architecture, and performance of the motion estimator. In Section 9, we will present the architecture design of texture block engine. The implementation result will be shown in Section 6. In Section 7, a brief conclusion will be given.

2. System Architecture

Figure 1 depicts the proposed platform-based MPEG-4 video coding system. RISC takes responsibility for MB 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

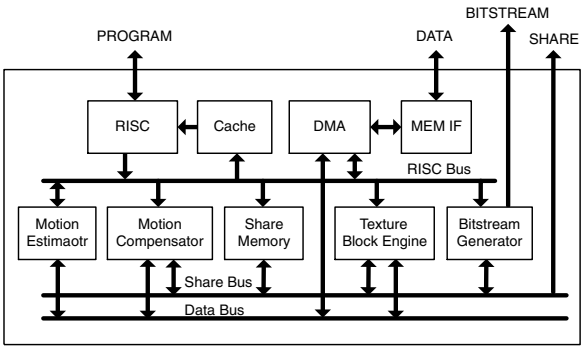

Figure 1. System architecture.

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. DMA involved in dedicated commands efficiently generates the proper addresses issued by RISC. 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 for MB coding mode decision at RISC. 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 the data automatically according 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.

The RISC core contains four stages pipeline with separated program and data memory. Its instruction set is 21 bits. The special 2-operand MAX and MIN instruction is included for the median operations for MV predictor decision. Besides, a hardwired datapath for multiplication and division is also provided. We also propose an immediate store instruction (SWI) to send a specified data to memory. Compared to traditional approach, which requires one instruction to move data into a register and then the other one to store it from the register to the memory, it results in significant reduction in the code size. To achieve cycle-accurately controlling, an inner-timer and polling technique are introduced. A special instruction, WAIT, is used to support this functionality. While the RISC encounters the WAIT instructions, it waits until the next trigger events.

3. Memory Organization

We have off-chip memory and several on-chip memory blocks. Off-chip memory contains source frames, reconstructed frames, and AC/DC information.

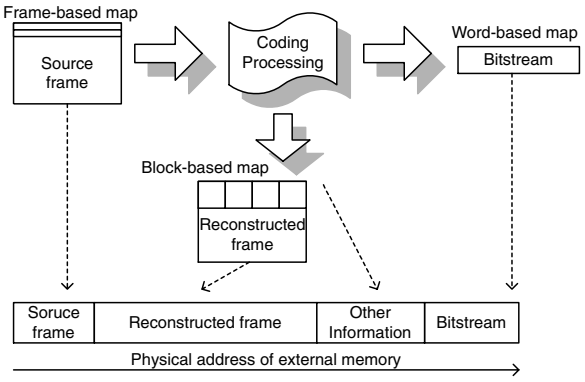

Figure 2. Heterogeneous memory organizations.

On-chip memory is used as local buffers to reduce the bus bandwidth. Due to the penalty of irregular accessing to and from off-chip RAM, we access off-chip RAM more successively by using random access on-chip RAM. For MPEG-4 video coding, block-based memory organization is efficient to burst reading a block of data for video processing. However, the common video input/output devices usually adopt the raster scan direction. It makes addressing more regular if frame data is arranged in frame-based scheme. Therefore, we use heterogeneous memory organization for off-chip RAM as shown in Fig. 2. The source frames are stored in the frame-based way, while the reconstructed frames are store in the block-based way for processing in the future. The bitstream data and AC/DC information is arranged as traditional 1-D addressing. After this arrangement, the data access to/from off-chip will more consecutive.

Figure 3 shows four types off-chip memory access. Each store unit (word) consists of four pixels. In case (a) the search window (SW) data of 48 $\times$ 48 pixels for ME are loaded from the previous reconstructed frame. In case (b) the 9 $\times$ 9 reference blocks are required for half-pixel MC. Reading in vertical direction can reduce the frequency of crossing neighboring blocks. In case (c) data are read out from source frames in frame-based organization. In case (d) the 8 $\times$ 8 reconstructed block is burst written to the block-based organization region of external memory without any segmentation.

The input video source, reconstructed frames, and transformed coefficients for AC/DC prediction are stored in the external memory. Direct Memory Access (DMA) plays a role to control memory interface (MIF) to read data from or write data to the external memory in a specified sequence after being initialized by RISC. For this kind of data-intensive applications, DMA always have a heavy load to handle the traffic through the data bus. Therefore, three special functions are involved in DMA to reduce addressing overhead and to provide pixel data more efficiently. It not only can improve the data access but also decrease the complexity of address generation in other hardware modules. First, the addressing generation combines the conversion process of 2-D to 1-D address. Second, the advanced prediction mode allows motion vectors to point out of the VOP and the data is padded from the boundary pixels in this situation. DMA handles this problem of boundary data for ME and MC units that can focus on the current processing MB. Third, special addressing for half-pixel precision compensation is supported. Due to the half-pixel precision for motion compensation, the compensated block is read out in 9 by 9 pixels and may occupy the four blocks in the block-based memory organization. This kind of fixed addressing is designed in the control unit of the DMA to improve the performance.

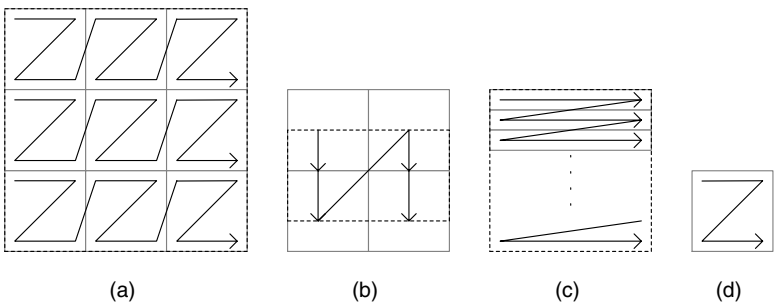

Figure 3. Memory access scheme. (a) SW for ME. (b) Block for MC. (c) Block to TBE. (d) Block from TBE.

4. Motion Estimator Design

4.1. Algorithm

To meet the requirement of various applications under the acceptable cost, we adopt two kinds of algorithms for the motion estimation of 16 × 16 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 MBs (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 8 × 8 pixels blocks in a MB around +2 to −2 positions of the previous best motion vector. The half-pixel refine-

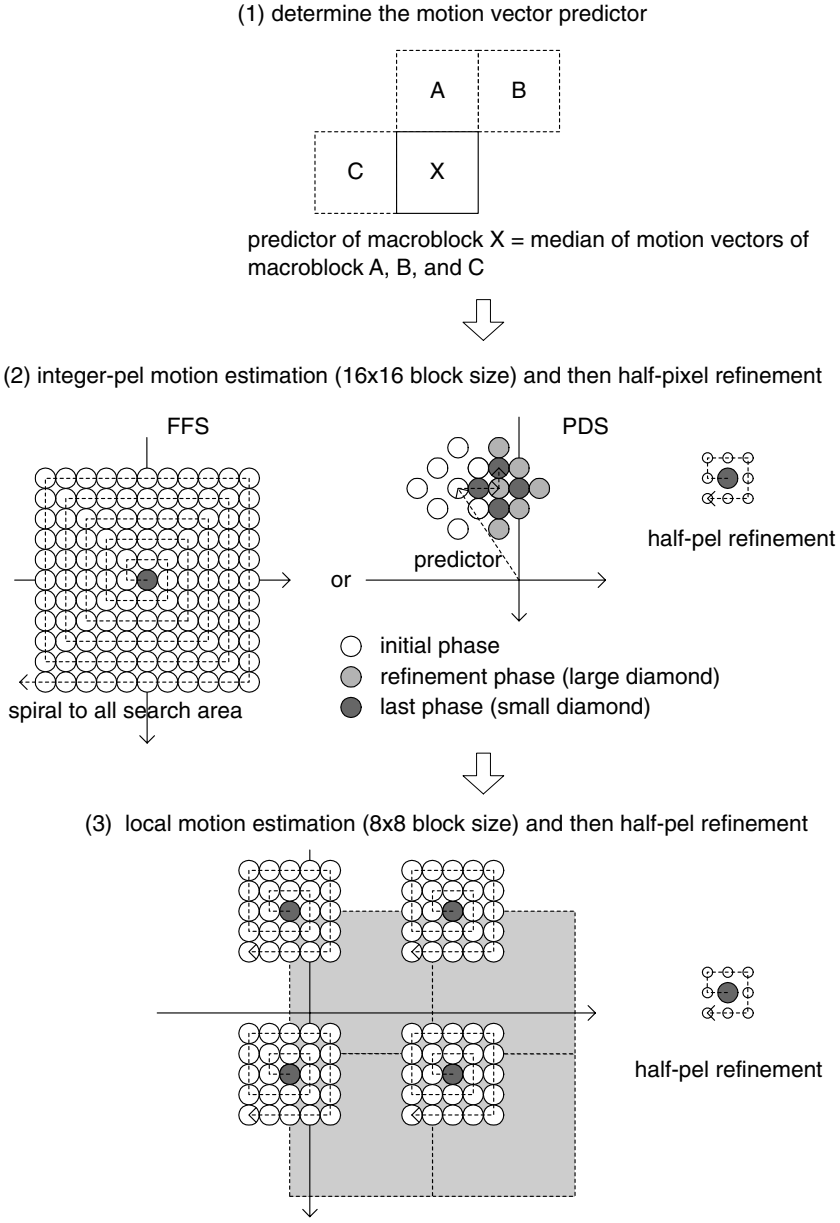

Figure 4. Algorithms of motion estimation.

ment is also applied for all found integer-pixel motion vectors. The whole stages of motion estimation is described as follows. The predictor is determined from neighboring MBs. 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 8×8 pixel blocks in a MB, 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.

Figure 4 depicts the whole stages of motion estimation and describes as follows. The predictor is determined from neighboring MBs. 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 8×8 pixel blocks in a MB, 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.

4.2. *Architecture*

Figure 5 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 reduce the computational complexity.

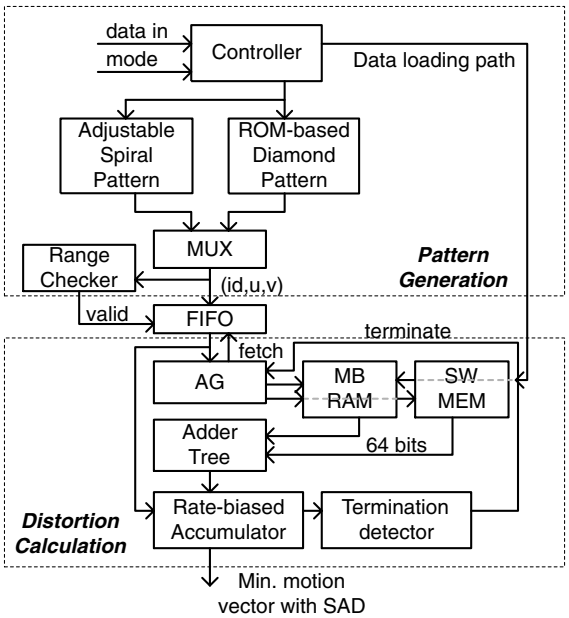

Figure 5. Architecture of motion estimator.

4.3. *Data Reuse Scheme*

The eight-way interleaved memory organization is used to dynamically fetch eight pixels in one cycle without reading collision in the same memory bank. Figure 6 depicts this organization with the search range -16 to $+15$ pixels. Before performing motion estimation for each MB labeled as A, B, or C, it needs to load colocated 48 by 48 pixel size of search windows in the reference frame to search window memory (SWMEM) which is divided into three strips and each one contains exclusive sixteen pixels. Under the leftmost MB of the frame, all data located in these three strips are required to be loaded. However, the left MBs in the same row can reuse two-thrid of search window of the immediately previous MB.

With rotation and modulation operations for addressing, this column-by-column data reuse scheme is applied to this motion estimation architecture. The bus traffic for loading search window is then reduced from 26.10 to 9.49 Mbytes per second for CIF format with the search range of -16 to $+15$ pixel. In each strip, eight horizontal neighboring pixels (a half row) are stored into eight separated memory with the linear addressing. While reading a half-row of pixels randomly, two consecutive addresses are calculated first from the two-dimension coordinates. Then the proper circular rotative operations are applied to the data read out from the memory banks.

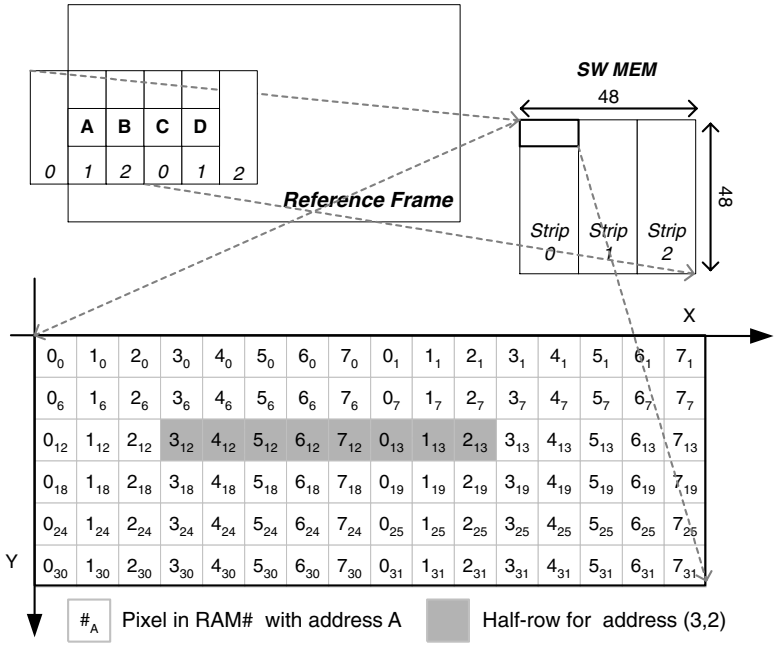

Figure 6. Memory organization of motion estimator.

4.4. Performance

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) [10]. 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

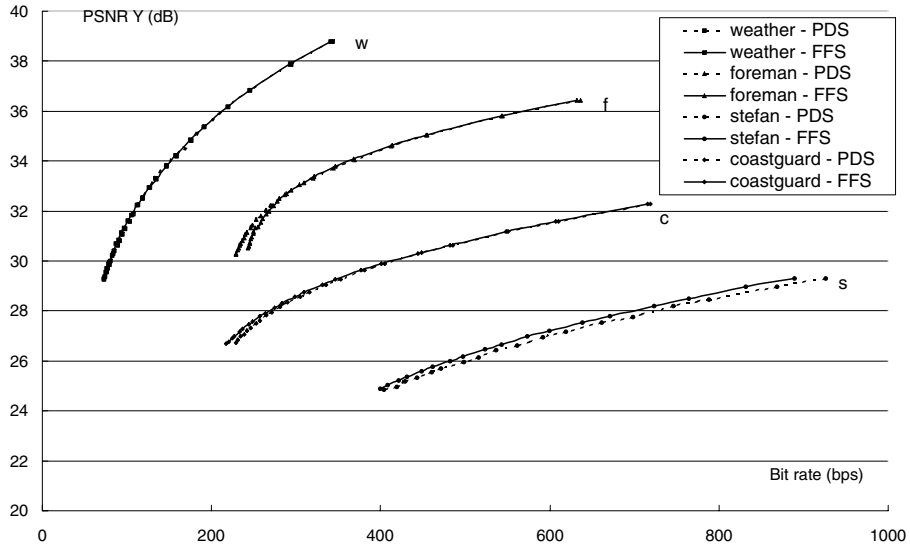

Figure 7. RD curves with PDS and FFS modes.

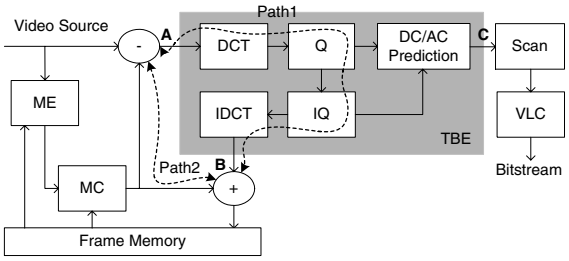

Figure 8. Texture block engine in MPEG-4 video encoding flow.

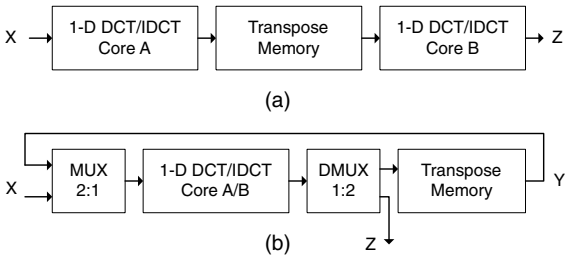

Figure 9. 2-D DCT architecture with row-column decomposition.

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. 7.

5. Texture Block Engine

The architecture of TBE is depicted in this section. It performs all operations of transform coding on a block basis, including DCT, Q, IQ, IDCT, and DC/AC prediction. Hence, a high degree of optimization have been carried out in previous DCT/IDCT designs (VM) [11]. However, without the consideration with other coding units in the same coding loop, it involves large buffer and cost while DCT/IDCT units are integrated into the coding system. The MPEG-4 encoding flow is shown in Fig. 8. The shaded region is TBE with one input (A) and two outputs (B, C). The point A is the source pixels compensated with the previous encoded pixels in the reference frame. The point B is the reconstructed pixels to be added with the reference pixels. The point C is the quantization coefficients in transform domain and will be reordered in the scan buffer and sent into the variable length coder (VLC). PATH 1, which traverses the coding loop, is longer than the PATH 2. Therefore, an additional buffer is required to be inserted into PATH 2 for the same latency and throughput with the other. If the latency of the PATH 2 is shorter, the size of the buffer will be smaller and it makes cost down. The shortest path will be achieved by using the two separated pipelined DCT and IDCT architectures. However, it causes a dominant cost in TBE after involving two suite of hardware design. Under the cost-effective consideration, one suite of hardware architecture with interleaving DCT/IDCT scheduling (IDIS) technique is exploited.

5.1. *DCT/IDCT Architecture*

2-D DCT transformation is typically separated into 1-D DCT transformation twice to reduce

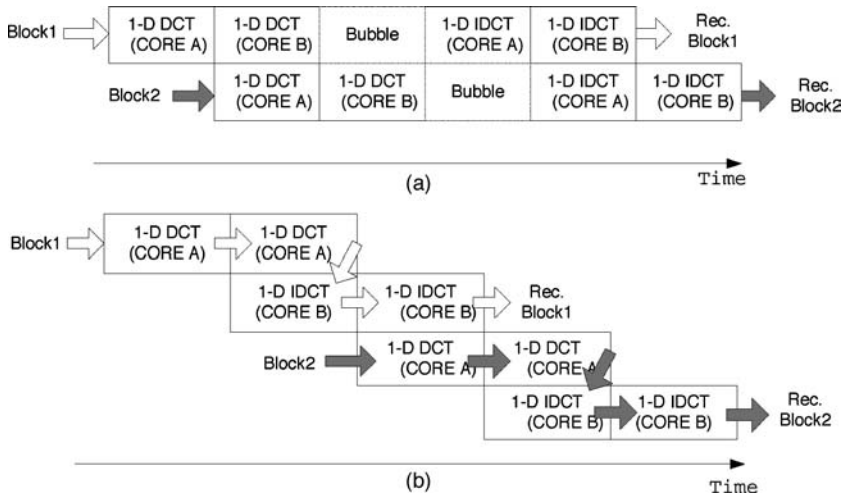

Figure 10. Configuration of core A and core B (a) the general schedule (b) proposed schedule.

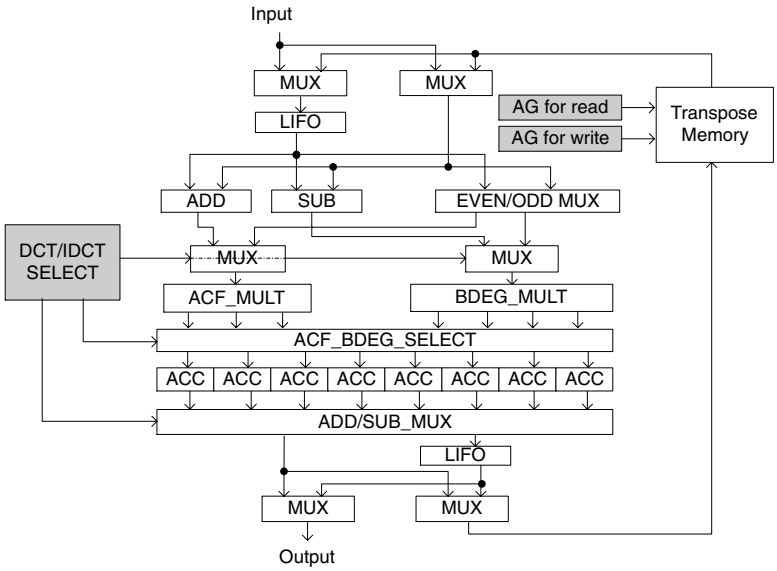

Figure 11. Architecture of DCT/IDCT.

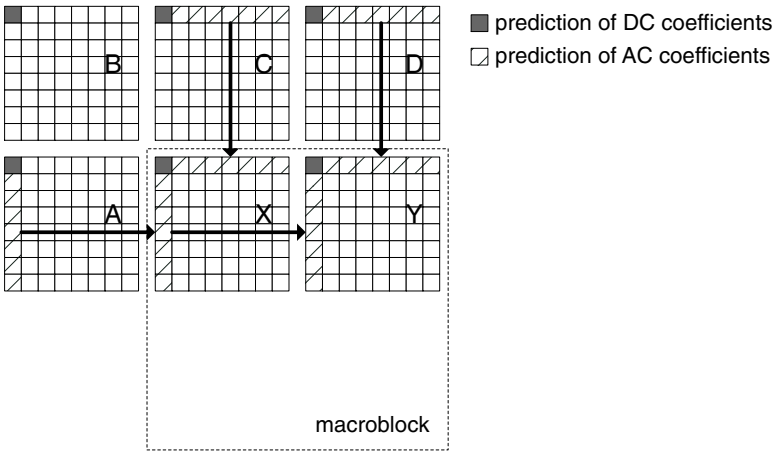

Figure 12. Previous neighboring blocks and coefficients used in improved ACDC prediction.

the complexity and hardware cost as depicted in Fig. 9(a). The transpose memory is required to reorder the coefficients from column-by-column to row-by-row directions. The parallel processing core with double throughput is used instead of two different cores due to the parallelism of 8 × 8 pixels block shown in Fig. 9(b). In generation consideration, the unit A and unit B performs the transformation in the same direction for the coefficients in the same block. However, the DCT/IDCT architecture has to perform six successive blocks for a MB in MPEG standards. It results in bubble cycles while changing the direction of the transformation and block buffers to hold the immediate block data. After changing the core A and core B to handle forward or inverse discrete cosine transformation at the same block, the block buffer is eliminated and reduces the bubble cycles in Fig. 10. The DCT/IDCT architecture in [11] is adopted and modified to achieve IDIS by simply adding a switching circuit that select between the DCT/IDCT operations. However, consider the intermediate Q and IQ operations between consecutive DCT and IDCT operations, latency will be introduced for pipelining to meet the timing specification. Therefore, the first 1-D IDCT operations cannot start and multiplex with the second 1-D DCT operations due to the latency mentioned above.

Separate address generators are used to cope with this timing problem to generate two independent addresses for reading and writing at the same time. The overall architecture of DCT/IDCT module is as shown in Fig. 11 and the necessary modifications are shown in shaded region. The AG units are dedicated designed and optimized for compacting the required working cycles.

5.2. *Substructure Sharing of Q/IQ and AC/DC Prediction*

AC/DC prediction is a new tool adopted by MPEG-4 to improve intra coding significantly as compared with MPEG-1 and MPEG-2 due to a strong correlation between the AC/DC coefficients of neighboring blocks. Figure 12 shows the concepts of AC/DC prediction. DC and AC coefficients are predicted adaptively from either the values of the immediately previous block or those of the block immediately above it. The prediction direction depends on the gradients of DC values of horizontal and vertical blocks. The prediction coefficients should be scaled according to the various quantization steps of blocks. A multiplication and then a division are required for each coefficient before predicting. Table 1 summaries the similarity among the operations of Q/IQ and AC/DC prediction. LEVEL means the quantized coefficients and is just the QAC value in the case of processing AC terms. Hence, the sharing of immediate data with AC/DC prediction and IQ is promising. Also, the division is the core operations for both Q and the second operation of ACDCP.

Table 1. Datapath of Q/IQ and AC/DC prediction.

ACDCP	$QAC = (QAC_{AorC} * QP_{AorC})/QP_X$
Q	$LEVEL = \|Coefficient\|/(2 * QP_X)$
IQ	$\|Coefficient\| = 2 * QP_X * LEVEL$

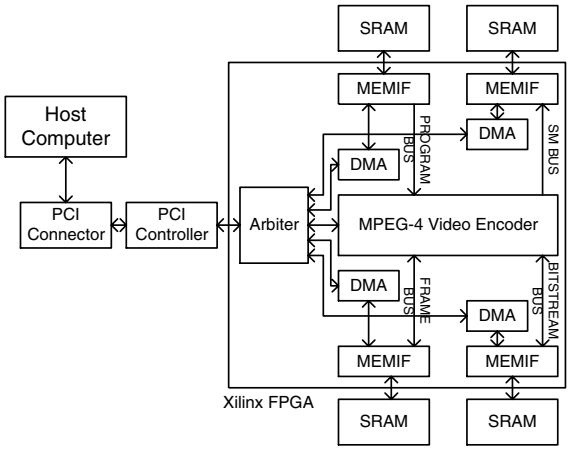

Figure 14. Configurable platform.

The datapath can be shared between them to reduce the cost.

The timing of TBE is shown in Fig. 13. We use the idle data path in the Q and IQ hardware unit to perform scaling operations in AC/DC prediction. For Y1 block, QAC * QP is first calculated in the IQ unit and then stored in the local prediction memory. For Y2 block in the case of horizontal AC prediction, it obtains QAC * QP (represented by M) of the Y1 block from the local prediction memory and use the divider

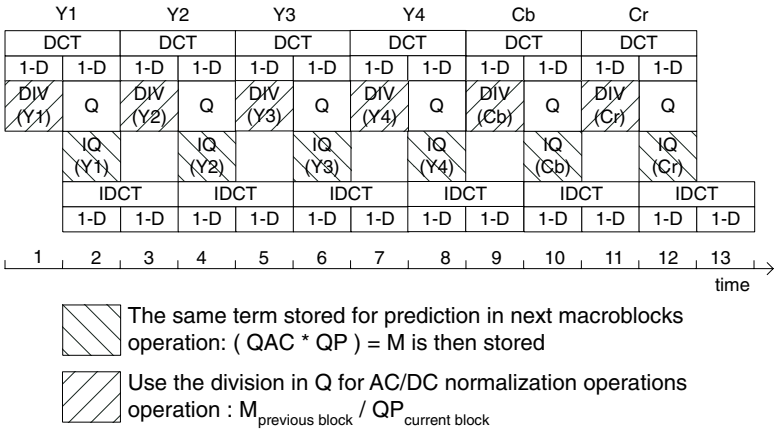

Figure 13. Timing diagram of substructure sharing.

Table 2. Comparison of texture block engine.

	Proposed 963 cycles/MB		K.Suh's [12] 1064 cycles/MB	
Module	Logic (Gate)	RAM (Bit)	Logic (Gate)	RAM (Bit)
DCT	16,511	64×16	7,005	64×16
IDCT			8,091	64×16
Q	5,000	128×8	3,514	0
IQ	3,403	0		
Buffer	0	384×12		384×12
DCACP	2,133	76×12	17,939	740×12
SCAN	1,736	0	2,814	0
IF	2,308	0	2,790	0
Total	31,091	7,568	42,180	15,536

Table 3. Characteristics of the encoder chip.

Technology	TSMC 0.35 μM 1P4M CMOS
Die size	5.02 × 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	3.3 V
Power consumption	256.8 mW
Package	208 CQFP
ME algorithm	PDS/FFS, 4 MV mode
	Search range −16.0 to +15.5
Encoding complexity	352 × 288 at 30 fps

in the Q unit, which is no operation for quantization in that time slot, to derive M / QP. By using this substructure sharing technique, the AC/DC prediction design is implemented under a little cost without dedicated multiplier and divider.

Table 2 shows the implementation and comparison of TBE. Compared to previous work, the coding loop contains no buffers and least latency, which in turn makes the number of buffer for MC a minimum of two. By the characteristics of IDIS, we apply sub-structure sharing technique for DC/AC prediction with Q and IQ to reduce hardware cost further.

6. Implementation

A configurable platform shown in Fig. 14 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 RISC program 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.

Figure 15 shows a micrograph of the encoder and Table 3 depicts its characteristics. It contains 828 K transistors and is fabricated on a 5.02 × 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 3.3 V and consumes 256.8 mW at 40 MHz working frequency. Table 4 shows the number of transistors, the area, and the size ratio to the chip of each unit.

Table 5 gives a comparison of some MPEG-4 video codec proposed before. In [4], it is a full dedicated hardware video codec design. It uses MVFAST for ME with search range −16∼+15.5. In [5], it is a platform-based video/speech codec design. It uses

Table 4. Cost distribution.

	Trans. (k)	Area (mm^2)	Size ratio (%)
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

Table 5 Architectures comparison.

Designer	[4]	[5]	[6]	Proposed
Encoding complexity	CIF, 15 fps	QCIF, 15 fps	CIF, 15 fps	CIF, 30 fps
Frequency (MHz)	13.5	60	27	40
Power mW	29	240	500	256.8
Transistor (K)	3,150	20,500 (DRAM)	1,700	829
Process (μ m)	0.18	0.25	0.35	0.35
Chip area (mm^2)	28.048	117.506	110.25	25.801

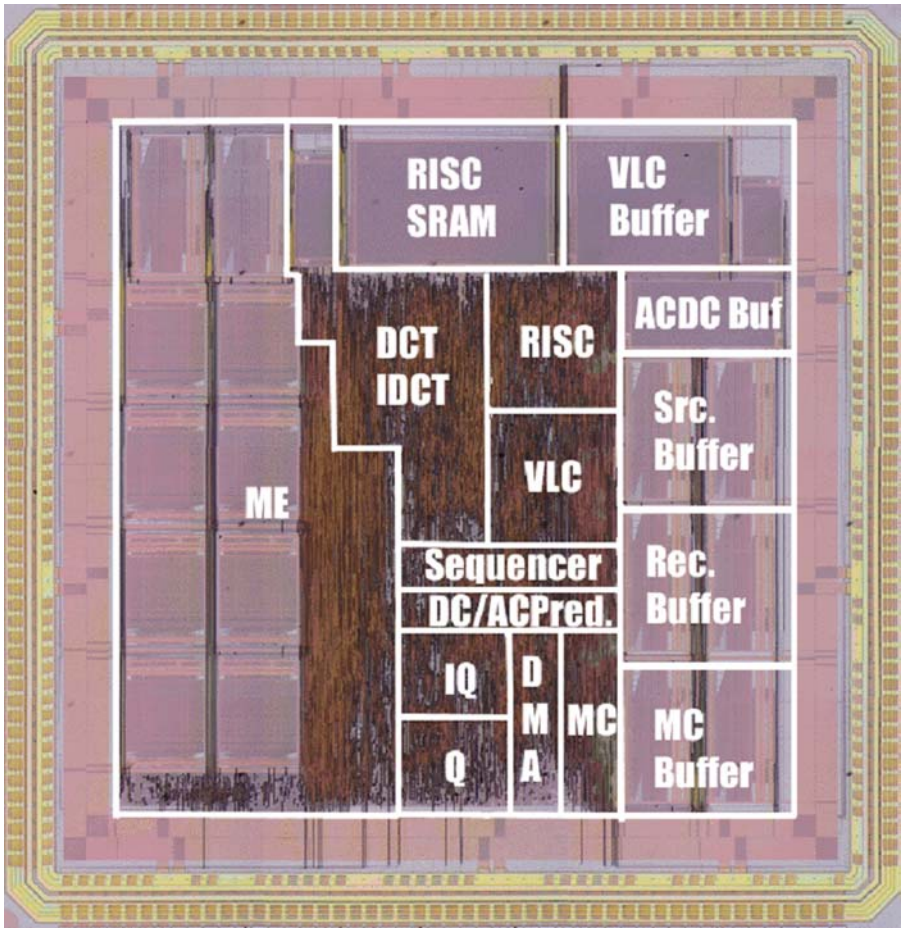

Figure 15. Micrograph of this encoder.

3-step hierarchical search for ME with search range $-32\sim+31.5$. In [6], it is a platform-based video codec design with ARM/AMBA. It uses a coarse ME with search range $-8\sim+7.5$. All chip designs adopts fast algorithms for motion estimation. In the viewpoint of video encoder parts, our work has highest encoding complexity and the lowest cost meanwhile.

7. Conclusion

An efficient platform architecture design with hardware accelerators for MPEG-4 video encoder SOC is proposed in this paper. With the proposed hybrid motion estimator and efficient texture block engine design, a real-time MPEG-4 Simple Profile@Level

3 video encoder is implemented. The proposed SOC adopts platform-based architecture with an embedded RISC core and efficient memory organization. The system is implemented with 0.35 μm CMOS technology. It works at 40 MHz and consumes 256.8 mW with 5.03×5.13 mm^2 die size to meet the real-time encoding specification. The proposed design achieves high performance with low design cost, which proves that a cost-effective MPEG-4 coding system implementation is realized.

References

1. P.M. Kuhn and W. Stechele, "Complexity Analysis of the Emerging MPEG-4 Standard as a Basis for VLSI Implementation," *International Conference on Visual Communications and Image Processing*, 1998.
2. H.C. Chang, L.G. Chen, M.Y. Hsu, and Y.C. Chang, "Performance Analysis and Architecture Evaluation of MPEG-4 Video Codec System," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2, 2000, pp. 449–452.
3. A. Hatabu, T. Miyazaki, and I. Kuroda, "QVGA/CIF Resolution MPEG-4 Video Codec Based on a Low-Power and General-Purpose DSP," in *IEEE Workshop on Signal Processing Systems (SiPS)*, 2002, pp. 15–20.
4. H. Nakayama, T. Yoshitake, H. Komazaki, Y. Watanabe, H. Araki, K. Morioka, J. Li, L. Peilin, S. Lee, H. Kubosawa, and Y. Otobe, "An MPEG-4 Video LSI with an Error-Resilient Codec Core Based on a Fast Motion Estimation Algorithm," *IEEE International Solid-State Circuits Conference (ISSCC)*, vol. 1, 2002, pp. 368–474.
5. M. Takahashi, T. Nishikawa, M. Hamada, T. Takayanagi, H. Arakida, N. Machida, H. Yamamoto, T. Fujiyoshi, Y. Ohashi, O. Yamagishi, T. Samata, A. Asano, T. Terazawa, K. Ohmori, Y. Watanabe, H. Nakamura, S. Minami, T. Kuroda, and T. Furuyama, "A 60-MHz 240-mW MPEG-4 Videophone LSI with 16-Mb Embedded DRAM," *IEEE Journal of Solid-State Circuit*, vol. 35, no. 11, 2000, pp. 1713–1721.
6. J.H. Park, I.K. Kim, S.M. Kim, S.M. Park, B.T. Koo, K.S. Shin, K.B. Seo, and J.J. Cha, "MPEG-4 Video Codec on an ARM core and AMBA," *Workshop and Exhibition on MPEG-4*, 2001, pp. 95–98.
7. W.M. Chao, C.W. Hsu, Y.C. Chang, and L.G. Chen, "A Novel Hybrid Motion Estimator Supporting Diamond Search and Fast Full Search," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 2, 2002, pp. 492–495.
8. S. Zhu and K.K. Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation," *IEEE Trans. on Image Processing*, vol. 9, no. 2, 2000, pp. 287–290.
9. P. Kuhn, *Algorithms, Complexity Analysis, and VLSI Architectures for MPEG-4 Motion Estimation*, Kluwer Academic Publications, 1999.
10. T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 1, 1997, pp. 19–31.
11. A. Madisetti and A.N. Willson, "A 100 MHz 2-D 8 × 8DCT/IDCT Processor for HDTV Applications," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 5, no. 2,1995, pp. 158–165.
12. K. Suh, S. Park, S. Kim, B. Koo, I. Kim, K. Kim, and H. Cho, "An Efficient Architecture of DCTQ Module in MPEG-4 Video Codec," *IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 1, 2002, pp. 777–780.
13. 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)*, 2002.

Yung-Chi Chang was born in Kaohsiung, Taiwan, R.O.C., in 1975. He received the B.S. and M.S. degrees from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 1998 and 2000, respectively, where he is currently pursuing the Ph.D. degree in the Graduate Institute of Electrical Engineering. His research interests include video coding algorithms and VLSI architectures for image/video processing.
watchman@video.ee.ntu.edu.tw

Wei-Min Chao was born in Taoyuan, Taiwan, R.O.C., in 1977. He received the B.S. and M.S. degrees from the Department of Electronics Engineering, National Taiwan University in 2000 and 2002 separately. His research interests include video coding algorithms and VLSI architecture for image and video processing.
hydra@video.ee.ntu.edu.tw

Chih-Wei Hsu was born in Taipei, Taiwan, in 1979. He received the B.S.E.E and M.S.E.E degrees from National Taiwan University (NTU), Taipei, in 2001 and 2003, respectively. He joined MediaTek, Inc., Hsinchu, Taiwan, in 2003, where he develops integrated circuits related to multimedia systems and optical storage devices. His research interests include object tracking, video coding, baseband signal processing, and VLSI design.
jeromn@video.ee.ntu.edu.tw

Liang-Gee Chen was born in Yun-Lin, Taiwan, in 1956. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from National Cheng Kung University, Tainan, Taiwan, in 1979, 1981, and 1986, respectively. He was an Instructor (1981-1986), and an Associate Professor (1986-1988) in the Department of Electrical Engineering, National Cheng Kung University. In the military service during 1987 to 1988, he was an Associate Professor in the Institute of Resource Management, Defense Management College. In 1988, he joined the Department of Electrical Engineering, National Taiwan University. During 1993 to 1994 he was a Visiting Consultant of DSP Research Department, AT & T Bell Lab, Murray Hill. In 1997, he was a visiting scholar of the Department of Electrical Engineering, University of Washington, Seattle. During 2001 to 2004, he was the first director of the Graduate Institute of Electronics Engineering (GIEE) in National Taiwan University (NTU). Currently, he is a Professor of the Department of Electrical Engineering and GIEE in NTU, Taipei, Taiwan. He is also the director of the Electronics Research and Service Organization in Industrial Technology Research Institute, Hsinchu, Taiwan. His current research interests are DSP architecture design, video processor design, and video coding systems.

Dr. Chen has served as an Associate Editor of IEEE Transactions on Circuits and Systems for Video Technology since 1996, as Associate Editor of IEEE Transactions on VLSI Systems since 1999, and as Associate Editor of IEEE Transactions on Circuits and Systems II since 2000. He has been the Associate Editor of the Journal of Circuits, Systems, and Signal Processing since 1999, and a Guest Editor for the Journal of Video Signal Processing Systems. He is also the Associate Editor of the Proceedings of the IEEE. He was the General Chairman of the 7th VLSI Design/CAD Symposium in 1995 and of the 1999 IEEE Workshop on Signal Processing Systems: Design and Implementation. He is the Past-Chair of Taipei Chapter of IEEE Circuits and Systems (CAS) Society, and is a member of the IEEE CAS Technical Committee of VLSI Systems and Applications, the Technical Committee of Visual Signal Processing and Communications, and the IEEE Signal Processing Technical Committee of Design and Implementation of SP Systems. He is the Chair-Elect of the IEEE CAS Technical Committee on Multimedia Systems and Applications. During 2001–2002, he served as a Distinguished Lecturer of the IEEE CAS Society. He received the Best Paper Award from the R.O.C. Computer Society in 1990 and 1994. Annually from 1991 to 1999, he received Long-Term (Acer) Paper Awards. In 1992, he received the Best Paper Award of the 1992 Asia-Pacific Conference on circuits and systems in the VLSI design track. In 1993, he received the Annual Paper Award of the Chinese Engineer Society. In 1996 and 2000, he received the Outstanding Research Award from the National Science Council, and in 2000, the Dragon Excellence Award from Acer. He is a member of Phi Tan Phi.
lgchen@video.ee.ntu.edu.tw