Efficient Architecture Design of Motion-Compensated Temporal Filtering/Motion Compensated Prediction Engine

Yi-Hau Chen, Chih-Chi Cheng, Tzu-Der Chuang, Ching-Yeh Chen, Shao-Yi Chien, and Liang-Gee Chen, *Fellow, IEEE*

***Abstract*—Since motion-compensated temporal filtering (MCTF) becomes an important temporal prediction scheme in video coding algorithms, this paper presents an efficient temporal prediction engine which not only is the first MCTF hardware work but also supports traditional motion-compensated prediction (MCP) scheme to provide computation scalability. For the prediction stage of MCTF and MCP schemes, modified extended double current Frames is adopted to reduce the system memory bandwidth, and a frame-interleaved macroblock pipelining scheme is proposed to eliminate the induced data buffer overhead. In addition, the proposed update stage architecture with pipelined scheduling and motion estimation (ME)-like motion compensation (MC) with level C+ scheme can also save about half external memory bandwidth and eliminate irregular memory access for MC. Moreover, 76.4% hardware area of the update stage is saved by reusing the hardware resources of the prediction stage. This MCTF chip can process CIF 30 fps in real-time, and the searching range is [−32, 32) for 5/3 MCTF with four-decomposition level and also support 1/3 MCTF, hierarchical B-frames, and MCP coding schemes in JSVM and H.264/AVC. The gate count is 352-K gates with 16.8 KBytes internal memory, and the maximum operating frequency is 60 MHz.**

***Index Terms*—H.264/AVC, motion-compensated temporal filtering, motion estimation, SVC, video coding, VLSI architecture.**

I. Introduction

IN RECENT YEARS, a new temporal prediction scheme, motion-compensated temporal filtering (MCTF), is developed to achieve efficient scalable video coding instead of the traditional motion-compensated prediction (MCP) scheme, which is based on motion estimation (ME) and motion compensation (MC). The main concept of MCTF is to perform discrete wavelet transformation with MC in the temporal direction. For more details, please refer to [1] and [2]. The concept of MCTF has been introduced to scalable video coding (SVC) standard developed by MPEG since SVM 3.0 [3]. Currently, the MCTF is adopted as an alternative and non-normative coding tool in Joint Scalable Video Model (JSVM) 8.0 [4]. In MCTF, the drift problem, which results from the mismatch of reference frames between the encoder and decoder, will not induce a catastrophic error propagation because it is an open-loop video coding. Moreover, when it is combined with existed coding standards, not only the temporal scalability can be efficiently provided, but also the compression efficiency can be improved. When the single-layer H.264/AVC is cooperated with MCTF scheme without spatial and SNR scalability, it can provide up to 1-dB coding gain compared to original H.264/AVC [5].

As the IC manufacturing process has a great progress, more and more functions or hardware modules are integrated into one single chip, which is called system-on-chip (SoC). To design an SoC hardware system, not only chip area is required to be considered, but also system resources such as system memory bandwidth and battery power should be taken into consideration. It is because these resources are shared simultaneously by many different tasks, like communication, audio, video, graphic, and so on. In our previous work [6], the VLSI architecture and the usage of system resources for 5/3 and 1/3 MCTF are analyzed. It shows that MCTF requires high system memory bandwidth due to its bidirectional prediction, update stage and multilevel structure. Besides, as shown in [7], system memory access can occupy more than 60% total power of a video system. Therefore, an efficient MCTF architecture should not only be area-efficient but also require low system memory bandwidth. Previously, the data reuse schemes of temporal prediction engine is only explored within single frame [8] and it is not enough for MCTF. On the other hand, since temporal prediction engine of MCTF schemes like JSVM always contains both open-loop MCTF and closed-loop MCP schemes to maintain its coding efficiency, how to combine different coding schemes is also an issue for an MCTF hardware design.

In this paper, we will present an MCTF/ME-combined hardware accelerator which is area-efficient and require low system memory bandwidth. First, the operations of the closed-loop MCP schemes and open-loop MCTF are analyzed. Due to their operations' similarity, an MCTF/ME-combined hardware accelerator is feasible. Moreover, since these coding schemes have different compression efficiency and system resource requirements, this temporal prediction engine can also provide computation scalability in an SoC system. Second, the frame-level data reuse schemes are applied to reduce the required system memory bandwidth, and a frame-interleaved macroblock-pipelining (MB-pipelining) is proposed to save the

Manuscript received November 6, 2006; revised April 11, 2007. This work was supported in part by National Science Council, Taiwan, R.O.C., under Grant 95-2221-E-002-195-. This paper was recommended by Associate Editor A. Navarro.

Y.-H. Chen, C.-C. Cheng, T.-Z. Chuang, S.-Y. Chien, and L.-G. Chen are with the Department of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: ttchen@video.ee.ntu.edu.tw; ccc@video.ee.ntu.edu.tw; peterchuang@video.ee.ntu.edu.tw; shoayi@video.ee.ntu.edu.tw; lgchen@video.ee.ntu.edu.tw).

C.-Y. Chen is with MediaTek, Inc., HsinChu 30076, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TCSVT.2007.913759

1051-8215/$25.00 © 2008 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

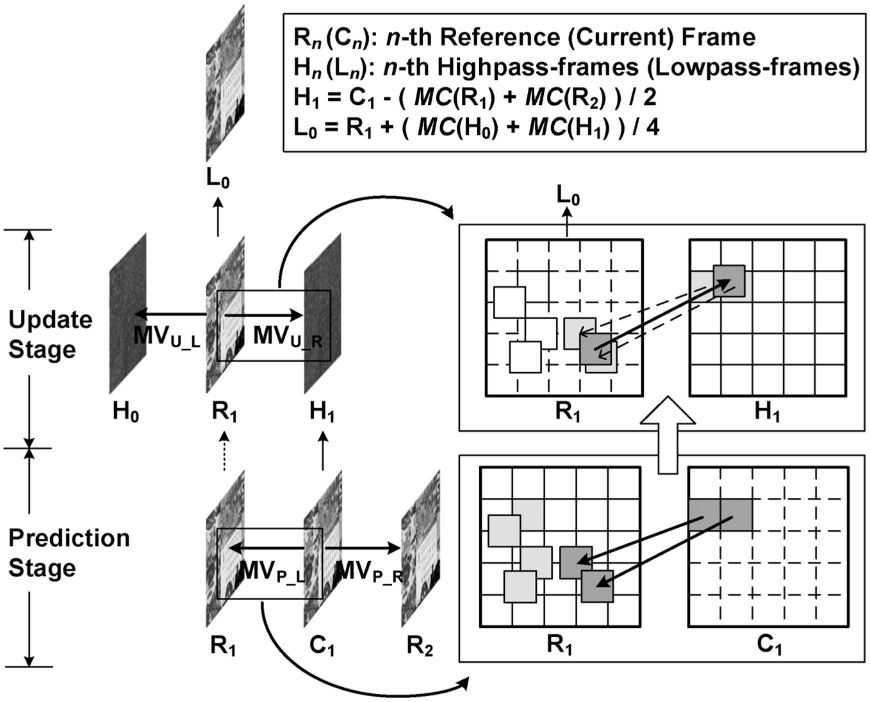

R_n (C_n): *n*-th Reference (Current) Frame
H_n (L_n): *n*-th Highpass-frames (Lowpass-frames)
$H_1 = C_1 - (MC(R_1) + MC(R_2)) / 2$
$L_0 = R_1 + (MC(H_0) + MC(H_1)) / 4$
L_0
Update Stage
MV_{U_L}
MV_{U_R}
H_0
R_1
H_1
Prediction Stage
MV_{P_L}
MV_{P_R}
R_1
C_1
R_2
L_0
R_1
H_1
R_1
C_1

Fig. 1. Scheme of 5/3 MCTF and the basic operations of prediction and update stage. Note that the derivation equations are only for biprediction mode.

induced on-chip buffer. Finally, for update stage, the system memory bandwidth is minimized by the proposed pipelined scheduling and ME-like MC with level C+ scheme, and the resource sharing strategy is discussed to reduce area cost.

This paper is organized as follows. In Section II, the basic operations of MCTF are introduced. The design challenges of MCTF and a combined temporal prediction engine with computation scalability will be discussed in Section III. Section IV presents the design of prediction stage in MCTF and MCP scheme. Section V discuss the design of the update stage. In Sections VI and VII, the experimental results and a conclusion are given, respectively.

II. Motion-Compensated Temporal Filtering in Temporal Prediction

MCTF is to perform wavelet transform in the temporal direction with MC. The coding performance depends on the adopted filter. In general, MCTF is implemented by use of 5/3, 1/3, and Haar filter with lifting scheme, which can provide the perfect reconstruction property. The 5/3 and 1/3 MCTF can be simply illustrated by Fig. 1, in which only two lifting stages are involved. The prediction stage is using reference frames to predict current frames, and the residual frames are the highpass frames (H-frames). The update stage is using the H-frames to update the reference frames, and the derived frames are the lowpass frames (L-frames). 1/3 MCTF can be performed by skipping the update stage of 5/3 MCTF and taking reference frames as L-frames directly.

In order to align the objects in different frames, both lifting stages need motion vectors (MVs). The block-based translational motion model is usually adopted. The major operations of prediction stage are ME and MC, as shown in the right-bottom part of Fig. 1. For every block in the current frames, ME can be performed to find the best MVs, MV_{P_L} and MV_{P_R}. Next, MC is performed to generate the H-frame by subtracting motion compensated frame from C_1 frame. In JSVM, because it is extended from H.264/AVC, variable block size ME and rate-distortion optimized mode decision are also adopted. Moreover, not only the biprediction but also the uniprediction are allowed in 5/3 or 1/3 MCTF. That is, Haar filter is also supported in 5/3 or 1/3 MCTF. If so, the equations in Fig. 1 will be replaced by uniprediction.

As for the MVs of update stage, MV_{U_L} and MV_{U_R}, they are usually derived from the MV of prediction stage, MV_{P_L} and MV_{P_R}, for saving the MV bitrate. Therefore, the major operations of update stage are deriving inverse MV and MC, as shown in the right-middle part of Fig. 1. For each current block in R_1 frame, the MV, MV_{P_L} or MV_{P_R}, which has the largest overlapped area between the reference block and current block will be selected, and its inverse MV will be assigned to be the MV of current block. The detailed operations will be further illustrated in Section V.

Fig. 1 only shows the one-level MCTF scheme. The multilevel MCTF scheme can be derived by recursively performing one-level MCTF on the derived L-frames in a bottom-up order, that is, from higher to lower frame rates. Furthermore, in JSVM, a coding scheme called hierarchical B-frames (HB) is also introduced to provide H.264/AVC compatible scalable coding bitstreams. HB is to perform multilevel temporal prediction structure in a top-bottom way. HB can be classified into two types, closed-loop HB and open-loop HB, according to its reference frames. In general, the coding performance of closed-loop HB is somewhat better than that of 5/3 MCTF [9] while the open-loop HB is almost the same as 1/3 MCTF, but the coding characteristics of both HB schemes are quite similar to 1/3 MCTF.

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

TABLE I
REQUIRED OPERATIONS OF VARIOUS CODING SCHEMES AND THEIR CORRESPONDING SYSTEM RESOURCE REQUIREMENT IN HARDWARE

	Required Operations			ME Complexity	Memory Bandwidth [10]
Coding Scheme	ME	MC	Update	search candidates/sec	MBytes/s
JSVM 4-level 5/3 MCTF	Y	Y	Y	9.49×10^7	71.62†
JSVM 4-level 1/3 MCTF	Y	Y	N	9.49×10^7	40.9
JSVM 4-level HB	Y	Y	N	9.49×10^7	40.9
H.264/AVC IPPP with 1-ref.	Y	Y	N	4.87×10^7	24.05
H.264/AVC IPPP with 2-ref.	Y	Y	N	9.73×10^7	42.02
H.264/AVC IBPBP with 2-ref.	Y	Y	N	9.73×10^7	42.02
H.264/AVC IBBP with 2-ref.	Y	Y	N	9.73×10^7	42.02

CIF 30 fps, Searching Range: [−32, 32), full-search blocking matching algorithm and Level C scheme [8].
† The external memory bandwidth of update stage is also derived by DRF scheme in [10].

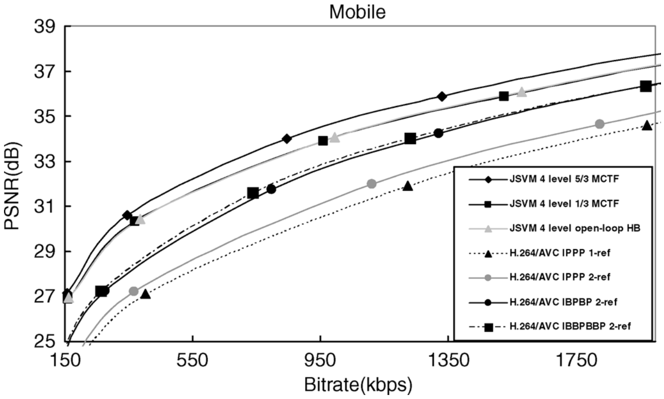


Fig. 2. PSNR comparison of JSVM 2.0 and H.264/AVC main profile for sequence "Mobile" in CIF 30 fps with searching range: [−32, 32). JSVM-coded schemes represent the single layer dedicated coding points without spatial and SNR scalability.

III. System Analysis and Design Challenges of MCTF

In this section, the performance and system analyses of MCTF and MCP schemes are given, and the computation scalability of the proposed MCTF/ME-combined hardware accelerator will be introduced. Then, the design challenges of prediction stage and update stage of MCTF are discussed, respectively.

A. *System Analysis and Computation-Scalability Property*

Our target is to design a combined temporal prediction engine, which can support MCTF and MCP schemes. The specification is CIF 30 fps, and the searching range is [−32, 32) in the horizontal and vertical directions. For MCTF, 5/3 MCTF, 1/3 MCTF, and HB from one level to four level decompositions are supported. For MCP, P-frame and B-frame are supported. There are two reference frames (2-ref) and two B-frames at most. For example, IPPP with 1-ref, IPPP with 2-ref, IBP with 2-ref, and IBBP with 2-ref are supported in our specification. Table I demonstrates the required operations' type and the hardware resource requirements of these supported coding schemes. In Table I, we take the number of search candidates per seconds as the threshold to compare "ME Complexity." In general, it can be formulated as

$$\begin{aligned}\#\text{ of search candidates/sec} &= \mathrm{SR}_H \times \mathrm{SR}_V \\ &\times(\#\text{ of Macroblock/frames}) \times (\#\text{ of frames/sec}) \times N\end{aligned} \quad (1)$$

where SR_H and SR_V are searching range in horizontal and vertical directions. N is the parameter decided by frame type. When P-frame with one reference frame is processed, N is equal to 1. For simplicity, no iteration refinement is executed for B-frame. Thus, when B-frame or P-frame with two reference frames is processed, N is equal to 2. The external memory bandwidth in Table I is referred from double reference frames scheme (DRF) in [10] with Level C data reuse scheme [8].

From Table I, all of these schemes have the same operations, ME and MC. Moreover, when two reference frames are adopted in MCP schemes, the computation complexities of these schemes are quite similar. In general, the similarity of operation types can increase the reuse ratio of processing element and the hardware cost overhead will be small, and the similar computation complexities can avoid the degradation of hardware utilization while performing different schemes. Upon above reasons, the similarity of required operations and computation complexity implies that it is feasible and efficient to design an unified hardware accelerator which consists of ME and MC functions and can be configured to perform different coding schemes.

Fig. 2 shows the coding performances of these supported coding schemes. Although they have the same basic operations and computation complexity, the coding performances are quite different due to various temporal prediction methods and coding flows. The simulation software is JSVM 2.0 for MCTF and open-loop HB prediction schemes and JM 9.8 for H.264/AVC main profile. The test sequence is "Mobile," which is in CIF 30 fps, and the searching range is [−32, 32). Noted that all the MCTF schemes in Fig. 2 are single-layer encoded without spatial and SNR scalability. In Fig. 2, four level 5/3 MCTF outperforms 1/3 MCTF and open-loop HB because of update stage, and its coding gain is 0.7 dB or so. The performances of SVC are better than those of H.264/AVC. The coding gain of 5/3 MCTF is 1.5 dB, compared to H.264/AVC IBBP with 2-ref. For more sequences' coding performance comparison, please refer to [5], [11]. On the other hand, these schemes' required system resources, like system memory bandwidth, are also very different as shown in Table I. Therefore, by adopting different coding schemes, the combined hardware accelerator can fit different system source constraints, such as memory bandwidth and power, while providing various coding performances. The above analysis shows the computation scalability for the MCTF/ME-combined hardware.

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

B. Design Challenges of MCTF

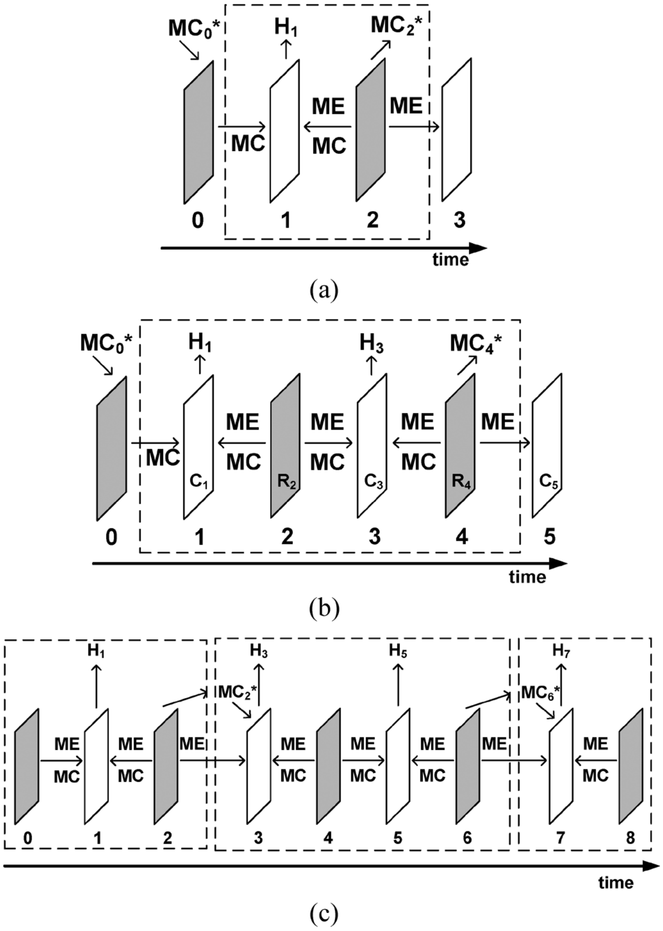
The design challenges of MCTF can be divided into two parts. One is for the prediction stage, and the other is for the update stage. In the prediction stage, the major computations are ME and MC, where the processing element (PE) design and memory bandwidth reduction with data reuse are always important issues. Since the prediction modes in JSVM are similar to those in H.264, the design of PEs for H.264/AVC [12] can be directly applied in the design of MCTF. Therefore, the major design challenge of prediction stage becomes data reuse. In the past, the data reuse scheme within one single frame has been explored [8]. For MCTF, frame-level data reuse is possible because of the nature of an open-loop video coding. In our previous work [10], [13], [14], frame-level data reuse is discussed, and there are many tradeoffs between system memory bandwidth and memory storage with various frame-level data reuse schemes. However, when frame-level data reuse schemes are adopted to reduce the memory bandwidth, the related hardware overhead and utilization should be considered. This is because different frame-level data reuse schemes may require different on-chip data buffer size in the conventional H.264/AVC design [15]. How to construct these frame-level data reuse schemes on hardware without on-chip data buffer overhead becomes the problem of prediction stage.

The major computations of update stage in previous JSVM 2.0 [16] are deriving inverse MV and MC. The former is a new operation, and the latter has many reference designs [12], [17]. Their computation complexities are small compared to ME, but they require large system memory bandwidth. Besides, unlike video encoders always store searching range data in on-chip buffer for random access of ME, video decoder has no on-chip buffer to store whole search range data under cost consideration. Therefore, MC has to access data directly from system memory. It makes that the MC-accessed location on system memory become unpredictable and discontinuous because the decoded MVs point to different regions on reference frames. Such irregular memory access will decrease the access efficiency of system memory and easily induce bubble cyclesdue to the property of DRAM [18], [19]. Hence, for deriving inverse MV, a new architecture is required, and a suitable schedule with the proper data reuse scheme is demanded to eliminate irregular memory access and reduce the required memory bandwidth for MC in update stage. On the other hand, as shown in Table I, three main operations, ME, MC, and update stage, are necessary for building an MCTF/ME-combined hardware accelerator. However, only 5/3 MCTF scheme requires the update stage, so it is inefficient to build a dedicated module to perform the update stage. Therefore, how to reuse the existed on-chip memory and PE resources is the key to achieve an efficient update stage design.

IV. Design Strategies of Prediction Stage

A. Proposed Frame-Level Data Reuse for MCTF and MCP Schemes

In [10], [13], and [14], the frame-level data reuse schemes are explored for MCTF and MCP schemes. For MCTF, the extended double current frames scheme (EDCF) which can minimize the number of input reference frames and share the searching range data between two successive current frames, has an apparent memory bandwidth reduction with two or more searching range buffers. Considering the MCP scheme like IBP with 2-ref or IPPP with 2-ref, two reference frames are supported, so it is reasonable that two searching range buffers are used in our specification.



(a)

(b)

(c)

Fig. 3. Frame-level data reuse scheme (white frame represents current frame; gray frame represents reference frame.). (a) Modified double current frame. (b) Modified extended double current frame. (c) Example of using three adaptive frame-level data reuse schemes for one GOP of 8 frames. The region surrounded by the dotted line represents one frame-level data reuse stage.

However, when we adopt EDCF for MCTF with two searching range buffer, the performance is degraded due to the overhead of memory bandwidth for MC, especially for variable block size MC. For example, for a 4 × 4 block in H.264, 81 pixels are required for fractional MC. Besides, the memory access of MC is irregular, and it induces a low efficiency of memory access as described in Section III-B. In order to solve this problem, we propose a modified EDCF (mEDCF), which is similar to the modified double current frames (mDCF) scheme in [13]. The concept of mDCF is shown in Fig. 3(a). The left reference frame can be replaced by the pre-interpolated MC frame (MC_0^*), and then the intermediate MC frame (MC_2^*) for the right current frame is interpolated after ME. Then, for a 4 × 4 block, only 32 pixels are required to be stored into and loaded from external memory to do MC. Moreover, the irregular

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

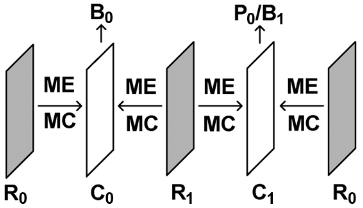
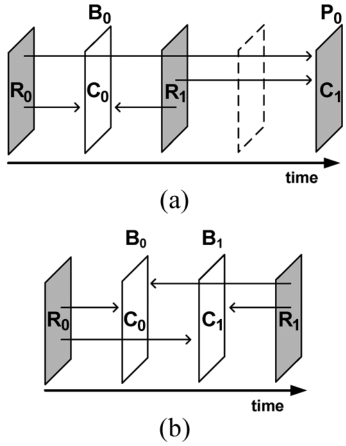


Fig. 4. Data dependency and resource allocation of MCFI-S scheme for P-/B-frame coding in MCP scheme. (C: current frame; R: reference frame).



(a)

(b)

Fig. 5. Definition of MCFI-S scheme for different MCP scheme (white frame represents B-frame; gray frame represents P-frame). (a) IBPBP coding scheme encodes one P-frame and one B-frame simultaneously. (b) IBBP coding scheme encodes two B-frames simultaneously.

memory access of MC is eliminated. We use the concept of mDCF to modify EDCF, so the final adopted scheme is shown in Fig. 3(b), which shares two reference frames between three current frames and interpolates the MC frame in advance for the next stage. As shown in Fig. 3(b), four ME of three current frames can be processed in one stage, and the number of input reference frames in the same temporal level can be minimized.

In practical MCTF coding scheme in JSVM [16], the frame-level data reuse schemes are limited by the GOP boundary. If the above mEDCF with two reference frames is directly applied for all frames, the bandwidth reduction may be degraded. Therefore, the structure of mEDCF is adaptively modified to fit the frame number in current GOP. As shown in Fig. 3(c), one mEDCF and two mEDCF's subsets are applied to construct the frame-level data reuse scheme for one 8-frame GOP for the best bandwidth reduction. Note that the frame-level data reuse scheme is applied for the frames in the same temporal layer in this paper.

For MCP scheme, our hardware engine can change the operation order of mEDCF to support the frame-level data reuse scheme, which is called minimum current frame inputs—P/B simultaneously (MCFI-S), as shown in Fig. 4. The concept is that the P-frame and B-frame can be computed at the same stage to halve the number of input reference frames, and Fig. 5(a) and (b) represent the definition of each frame in MCFI-S scheme for IBPBP and IBBP coding schemes, respectively. In Fig. 5, the notation shown in the frames in are the same as those of Fig. 4. The MCFI-S scheme can reduce the system memory bandwidth while B-frame is used in MCP coding scheme. As for IPPP with 1-ref or 2-ref, the traditional scheme [12] is adopted.

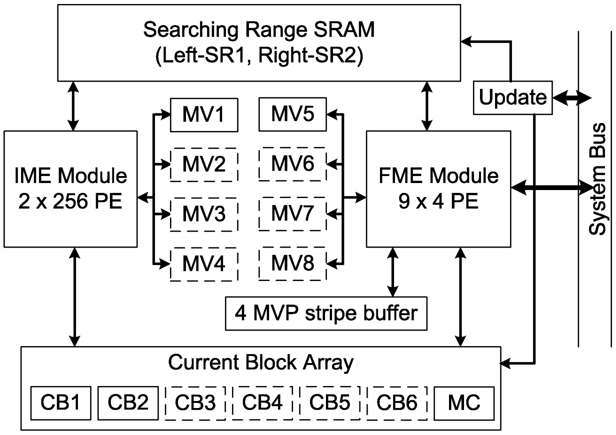


Fig. 6. Block diagram of an MCTF/ME-combined hardware accelerator based on proposed frame-interleaved MB pipelining scheme, the region surrounded by dotted lines are the extra buffer while applying MB pipelining scheme (SR: searching range; CB: current block; MC: motion compensated; MV: motion vector buffer).

B. Proposed Frame-Interleaved MB Pipelining Scheme

Since the prediction modes in MCTF are very similar to those in H.264/AVC, the architecture design of processing elements in our previous work [12] is adopted with minor modifications for the proposed temporal prediction engine. Because of the high complexity of variable block size ME (VBSME) and rate-distortion optimized mode decision in H.264/AVC, the computations of ME are divided into two parts, integer ME (IME) and fractional ME (FME) [15], and each part is executed by its corresponding module, as shown in Fig. 6.

However, there are many buffers for current blocks and partial results, because three current frames and four ME are processed at the same stage in the proposed mEDCF scheme. Fig. 7 shows the schedule of four ME for three current frames in one EDCF stage, where the notations are the same as those of Fig. 3(b). If the original MB-pipelining scheme [15] is adopted, FME will start to process MB_n only after the computations of IME for MB_n in different current frames are finished as shown in Fig. 7(a). Therefore, a larger buffer, six current MBs and eight MV sets, are required for four ME of three current MBs for IME and FME, respectively, as shown in Fig. 6. Moreover, these buffers also lower the hardware utilization, because not all of them are required in other coding schemes.

In order to reduce these data buffer and improve the hardware utilization, a frame-interleaved MB pipelining scheme is proposed. The ME schedule of our proposed frame-interleaved MB pipelining scheme is shown in Fig. 7(b). In the proposed scheme, once the unidirectional IME and FME of each current block is finished, IME and FME enter the next pipeline stage and the related partial results are also propagated to the next processing module. Therefore, the data lifetime of current MBs and MV data are minimized, and only two current MB buffers and two MV buffers are required for IME and FME. The buffer surrounded by dotted region in Fig. 6 can be removed. That is, the buffer overhead induced by the frame-level data reuse scheme is eliminated.

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

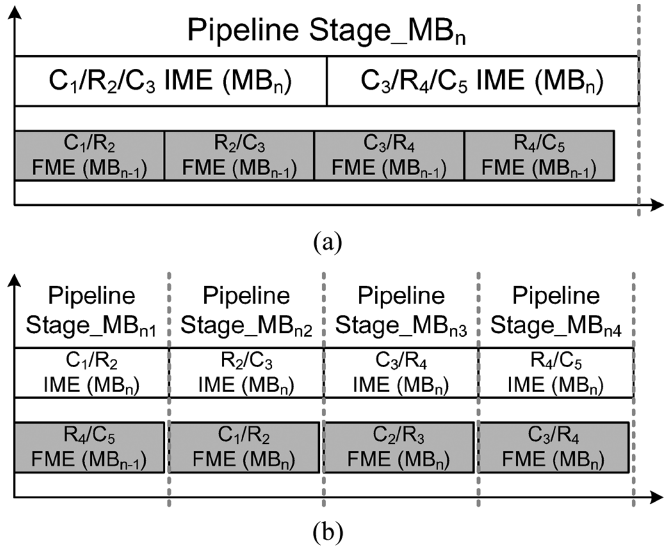


Fig. 7. ME schedule compatible to extended m-DCF (C: current frame; R: reference frame). (a) ME schedule of MB pipelining scheme. (b) ME schedule of the proposed frame-interleaved MB pipelining scheme.

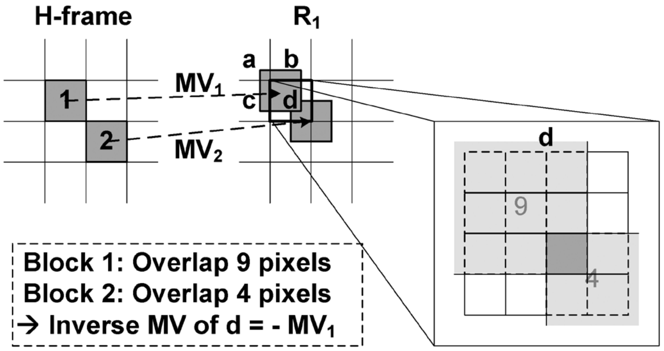

H-frame
R1
a
b
c
d
MV1
MV2
1
2
d
9
4
Block 1: Overlap 9 pixels
Block 2: Overlap 4 pixels
→ Inverse MV of d = - MV1

Fig. 8. Operations of deriving inverse MV. Two 4 × 4 blocks, block 1, and block 2, overlap block d with 9 and 4 pixels, respectively. The inverse MV of block d is therefore $-\mathrm{MV}_1$.

V. Design Strategies of Update Stage

In the beginning of this section, we will introduce the detailed operations and several schemes to derive inverse MV in the update stage. Next, hardware resource reuse of the prediction stage to perform MC in the update stage is discussed. In the last part, the proposed architecture of update stage is presented.

A. Proposed Pipelined Scheduling for Deriving Inverse MV

The detailed operations of deriving inverse MVs in the update stage of SVC are shown in Fig. 8. Each 4 × 4 block in H-frames (ex. block 1) has a MV and a corresponding reference block, which overlaps 4 blocks, block a, b, c, and d, in the reference frames, R_1. For each block in the reference frames, the overlapped areas of different MVs will be calculated and compared to each other. When more than two MVs point to the same 4 × 4 block in the reference frame, only the MV with the larger overlapped area will be kept. The derived MV of this block will be equal to the inverse of the MV with the largest overlapped area as shown in Fig. 8. Once no MV points to the 4 × 4 block in the reference frame, the overlapped area of this 4 × 4 block will be set to zero. It means the 4 × 4 block will not perform MC in this direction as shown in (3) and (4).

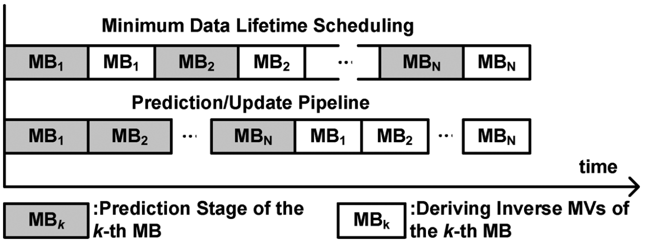

Minimum Data Lifetime Scheduling
MB1
MB1
MB2
MB2
...
MBN
MBN
Prediction/Update Pipeline
MB1
MB2
...
MBN
MB1
MB2
...
MBN
time
MBk :Prediction Stage of the k-th MB
MBk :Deriving Inverse MVs of the k-th MB

Fig. 9. Two different task scheduling between prediction stage and deriving inverse MV in a frame (assume there are N MBs in a frame).

For each block in the reference frame, if deriving inverse MVs is processed MB by MB (also block by block) of reference frame, it is required to examine all blocks in its corresponding searching range in H-frame to compute the overlapped area. It induces a high computation complexity because the number of examined blocks is always large. To avoid high computation complexity, the preferred scheme is to perform the operations from the viewpoint of H-frames. Hence, deriving inverse MVs MB by MB of H-frames is first proposed in SVM 3.0 [3]. In this scheme, motion information of update stage, including currently selected inverse MV and its corresponding overlapped area, is required to be buffered for each 4 × 4 block of reference frame. For example, for block 1 in Fig. 8, motion information of block a, b, c, and d is read from an inverse MV buffer. If the overlapped area induced by currently processed MV is larger than the stored one, the corresponding motion information is updated and written back to inverse MV buffer.

Since both the prediction stage and the deriving inverse MVs are processed MB-by-MB of H-frames, it is possible to interleave the processing of prediction and deriving inverse MVs. Therefore, there are two possible scheduling. The first one is the interleaved scheduling to minimize data lifetime of the MVs in the prediction stage, and the second one is the proposed pipelined scheduling to perform the prediction and update stages, respectively. We will discuss these two schedulings in the following.

1) The Interleaved Scheduling: The interleaved scheduling is to minimize the data lifetime of the MVs in the prediction stage, which means that when a 4 × 4 block in H-frames is generated, its MV has to be used for deriving inverse MVs, as shown in the top part of Fig. 9. This is because if this MV is not used out immediately, it has to be stored in external memory for the update stage and induces additional external memory access. The interleaved scheduling is to use out the MV as soon as it is generated in order to reduce the external memory access and storage.

Once a 4 × 4 block finishes its prediction stage, motion information of the overlapped four 4 × 4 blocks in the reference frame has to be read from external memory. The motion information of the four overlapped blocks is then updated and written back to external memory. Therefore, the required external memory access of processing one 4 × 4 block in H-frame is $[2 \times 4 \times 2 \times (\#\text{ of Bits for Motion Info.})]$, where the first 2 means the forward and backward directions, 4 is the number of overlapped blocks, the second 2 is the memory access of read and write operations, and # of Bits for Motion Info. is the amount of data to represent MV and overlapped area of update stage. The overlapped area ranges from 0 pixels to 16 pixels, and 5 bits are required. Assume a MV occupies 16 bits, then the external memory access for one 4 × 4 block of H-frames is 336 bits.

2) The Proposed Pipelined Scheduling: The proposed pipelined scheduling is to separate the operations of prediction and update stages, as shown in the bottom part of Fig. 9. In this case, the MVs of H-frames have to be stored in the external memory. This will induce external memory access of one read and one write of each MV in H-frames.

However, since the prediction stage is finished when update stage is processed, the on-chip memory, which is used as the searching range buffer for ME in the prediction stage, is available to buffer the motion information of the update stage. In general, the searching range buffer is large and is sufficient for using Level D data reuse scheme [8] to buffer the motion information of update stage. Level D date reuse scheme can achieve minimum external memory access of reference frame. After finishing processing one MB row, the motion information of the top MB row in the buffer will be the final inverse MVs.

Therefore, for one 4 × 4 block in H-frames, the external memory access is $[2 \times (2 \times (\#\text{ of Bits for MV}) + (\#\text{ of Bits for Motion Info.}))]$, where the first 2 are the bidirections, the second 2 is the memory access of reading and writing MVs of H-frames, and the final part is the output of final inverse MVs. If a MV occupies 16 bits and the overlapped area (from 0 to 16 pixels) requires five bits, this number will be 106 bits. Compared with the interleaved scheduling, the proposed pipelined scheduling has a 68.5% reduction of memory bandwidth.

B. Proposed ME-Like MC With Level C+ Data Reuse Scheme

The major operations of MC in the update stage of JSVM can be shown as the following equations:

$$E_{L,R} = \left(\sum_{(L,R)\in \text{Block}_{L,R}} I_{L,R}(i,j)^2 + 128 \right) \gg 8 \quad (2)$$

$$W_{L,R} = \max(0, \text{Area}_{L,R} - 8) \times \max(0, \min(16, 20 - E_{L,R})) \gg 7 \quad (3)$$

$$L(i,j) = \text{Ori.}(i,j) + (W_L \times I_L(i,j) + W_R \times I_R(i,j) + 1) \gg 2 \quad (4)$$

where $\text{Block}_{L,R}$ is the left or right interpolated block, $\text{Area}_{L,R}$ is the overlapped area of the derived inverse MV, $E_{L,R}$ represents the energy sum of selected interpolated block, $W_{L,R}$ is the weighting factor, and $I_{L,R}(i,j)$, $\text{Ori.}(i,j)$, and $L(i,j)$ are the pixel values in the interpolated, original, and updated blocks, respectively. An L-frame is generated, after a reference frame is added by the bidirectional MC in the update stage. The weighting factor $W_{L,R}$ which depends on the H-frame energy and overlapped area, is used to reduce the ghost effect, which occurs if the derived MV is wrong. This method is called adaptive weighting update stage. Note that because of the weighting factor, the word length of H-frame for update stage can be truncated from 9 to 8 bits, which is beneficial for memory access and storage, while the coding performance of update stage is almost the same [20]. In the following, we will first discuss the conventional MC scheme, which is adopted in most current H.264/AVC decoder design [17], [21]. Then, the proposed ME-like MC with Level C+ scheme will be introduced.

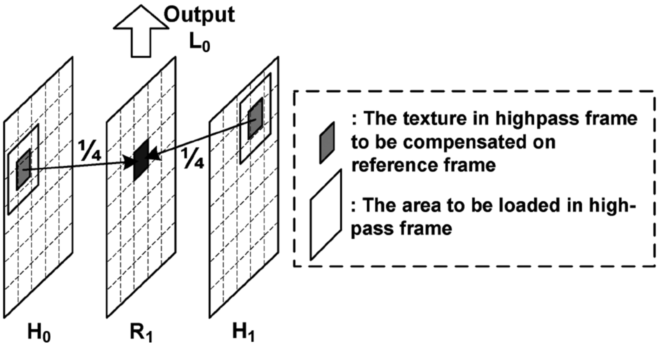

Output
L_0
¼
¼
H_0
R_1
H_1
: The texture in highpass frame to be compensated on reference frame
: The area to be loaded in highpass frame

Fig. 10. Illustration of double reference frames scheme of MC (H: highpass frame; R: reference frame).

1) The Double Reference Frames Scheme: The direct implementation of MC in the update stage is to adopt the DRF, as shown in Fig. 10. Both sides of MC are executed at the same time, and the required data for the interpolation of bidirectional MC are directly loaded from the external memory. In JSVM 2.0, the interpolation filter with 6 taps is utilized. 9 × 9 pixels are required for a 4 × 4 block. Then the required memory access of this scheme for a 4 × 4 block becomes

$$[2 \times 4 \times 4 + 2 \times 9 \times 9] \times 8 \text{ bits} \quad (5)$$

where the first part is the input and output of current block to be performed MC, the second part is the input data for MC, and 8 bits is the number of bits for each pixel. The total required memory access for one 4 × 4 block is 1552 bits. Moreover, the memory access is irregular, since the inverse MV can point to anywhere within search range.

2) The Proposed ME-Like MC With Level C+ Scheme: When performing the update stage, the hardware resources of the prediction stage are idle, so they can be used in the update stage to reduce the hardware cost of update stage. There are two major hardware resources, processing elements and on-chip memory. The processing elements of FME module can be reused to perform MC in the update stage. The on-chip memory is usually a large buffer when full search ME with Level C data reuse scheme is adopted, and it can be used as the data buffer for MC. The basic idea of the proposed ME-like MC is to preload the searching range of current MB into the on-chip memory. Moreover, the double current frames scheme (DCF) [10] and Level C+ data reuse scheme [22] are also applied to save the required external memory access. We will introduce them in the following, respectively. By these methods, not only the required memory access can be reduced but also the irregular external memory access can be eliminated.

The concept of DCF is applied to reuse the searching range in ME-like MC, as shown in Fig. 11. The loaded searching range of H-frames can be used to compensate two neighboring reference

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

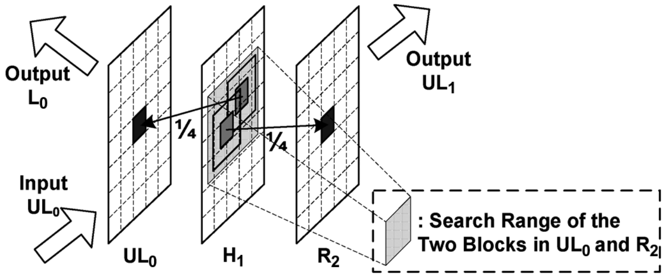


Fig. 11. Illustration of the proposed ME-like MC with Level C+ scheme (H: highpass frame; R: reference frame; UL: partial result of reference frame; L: lowpass frame).

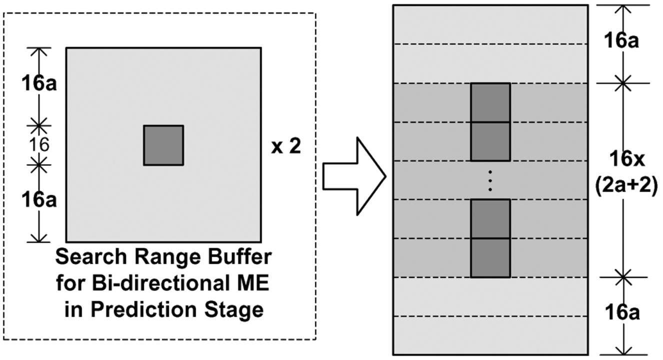

16a
16
16a
x 2
Search Range Buffer for Bi-directional ME in Prediction Stage
16a
16x (2a+2)
16a

Fig. 12. Data reuse scheme for the proposed ME-like MC with Level C+. If the search range of one MB is $[-16a, 16a)$, the data in this integrated searching range is sufficient for $2a+2$ successive MBs in the vertical direction. The MB width and height are 16 in this figure.

frames, so the external memory access of this part can be halved. Similar to the discussion of DCF in Section IV-A, the overhead is to read and write the partial results of reference frame for the left-side MC (UL_0).

Since there are two searching range buffers in the prediction stage and no data dependency exists between different blocks in the MC of update stage, Level C+ data reuse scheme with stripe scan can be utilized without overhead [22]. Fig. 12 shows the vertical data reuse scheme for the proposed ME-like MC with Level C+. We assume that the searching range is $[-16a, 16a)$. If we cascade two searching ranges in the vertical direction, the data in this integrated searching range is sufficient for $2a+2$ successive MBs in the vertical direction. Therefore, the vertical data reuse of $2a+2$ successive MBs is achieved in this scheme. With the proposed scheme, each pixel in the searching range has to be averagely loaded only $1+(2a/(2a+2))$ times.

In this paper, the searching range is $[-32, 32)$ $(a=2)$, for a 4×4 block, the required external memory access is

$$\left[2+2+\left(1+\frac{4}{6}\right)\right] \times 4 \times 4 \times 8 \text{ bits} \qquad (6)$$

where the first 2 is the input of R_2 and the output of L_0, the second 2 is the reading and writing of partial results (UL_0 and UL_1), and the third part is the amount of data for MC. The total required memory access is 725.33 bits in average for a 4×4 block, which is a 53% reduction compared to the direct implementation. Moreover, the irregular off-chip memory access for MC is turned into regular memory access because the scheme of preloading whole searching range data is static and regular.

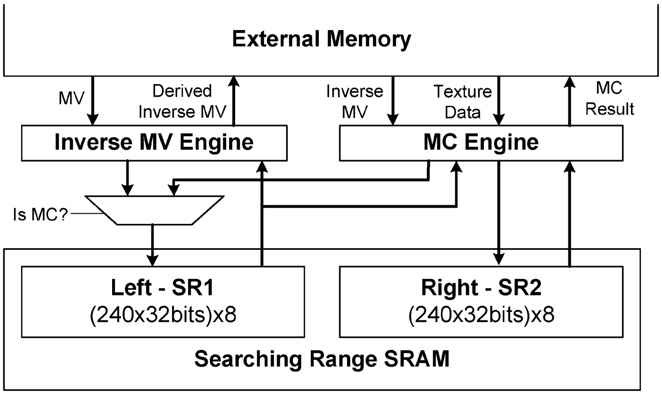


Fig. 13. Proposed hardware architecture of update stage (SR: searching range buffer).

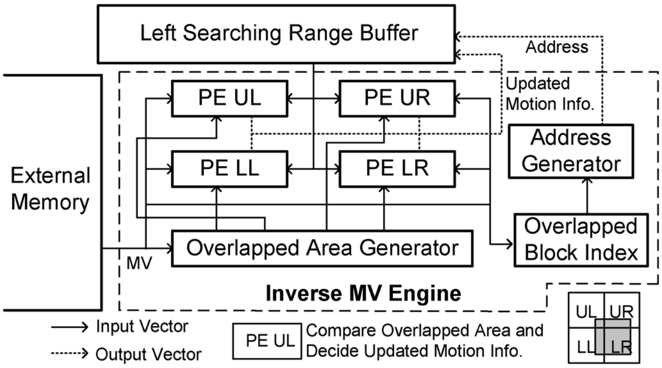


Fig. 14. Proposed hardware architecture of inverse MV engine.

C. Proposed Hardware Architecture of Update Stage

Fig. 13 shows the proposed hardware architecture of the update stage. The target specification is CIF 30 fps, and the searching range is [−32, 32) in both vertical and horizontal directions. When deriving inverse MVs, the MVs of H-frames are input MB-by-MB. The motion information of update stage is stored in *Left Searching Range Buffer* with Level D data reuse scheme [8]. After processing one MB row, the derived inverse MVs of one MB row are output to the external memory for the MC in the update stage. When the operations of deriving inverse MVs are finished, the MC engine performs the proposed ME-like MC with Level C+ scheme by both two *Searching Range Buffers*.

Not all hardware resources are necessarily dedicated for the update stage. As discussed in Section V-A and V-B, the searching range buffers and processing elements of FME module, which are designed for the prediction stage, can be reused for the computation of the update stage. In the following, we introduce the detailed architecture of the update stage.

1) Hardware Architecture of Inverse MV Engine: The MC engine can reuse the MC module needed in FME of prediction stage. Therefore, the dedicated hardware resource for update stage is only the *Inverse MV Engine*. Fig. 14 shows the proposed hardware architecture of *Inverse MV Engine*. At each cycle, one 4×4 block in H-frames (current block) is processed. The MV of current block is loaded from external memory. In *Overlapped Block Index*, the logical address of the overlapped four 4×4

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

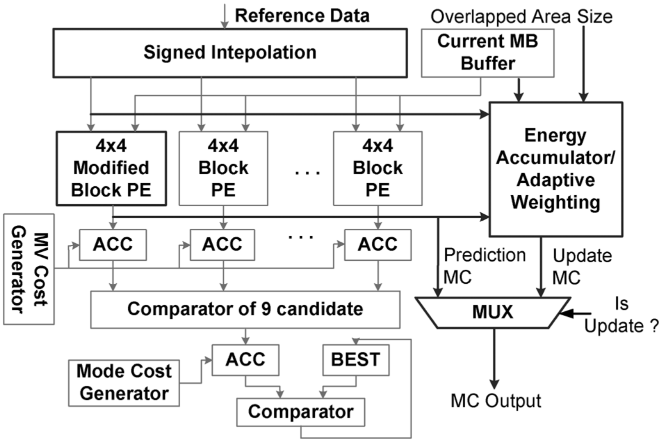


Fig. 15. Modified FME architecture which can also support the update stage. The thick-lined parts are additionally designed for update stage.

TABLE II
EXTERNAL MEMORY BANDWIDTH COMPARISON FOR PREDICTION STAGE

Coding Scheme	DRF [10] (MByte/s)	Proposed (MByte/s)	Reduction
Prediction Stage in MCTF			
1 Level Decomposition	33.04	33.04	0.0%
2 Level Decomposition	37.53	35.32	5.9%
3 Level Decomposition	39.78	34.21	14.0%
4 Level Decomposition	40.90	32.54	20.4%
MCP schemes			
IPPP with 1-ref	24.05	24.05	0.0%
IPPP with 2-ref	42.02	42.02	0.0%
IBP with 2-ref	42.02	24.05	42.8%
IBBP with 2-ref	42.02	30.04	28.5%

CIF 30 fps, Searching Range: $[-32, 32)$, two searching range buffers are included. Level C scheme [8] is applied for both schemes' MB-level data reuse.

blocks in reference frame will be generated. The *Address Generator* maps the logical address into physical memory address and reads the motion information of the four overlapped blocks from the search range buffer.

The overlapped area of four overlapped blocks caused by current block is calculated in *Overlapped Area Generator*. Four PEs are utilized to compare the overlapped areas of the current block and that of the previously processed blocks. The updated results of the four overlapped blocks are written back into the searching range buffer. The organization of *Left Searching Range Buffer* is 8 banks of 240 $(= 1920)$ words, and each word is 32 bits [12]. If one 32-bit word is used to buffer motion information of one 4×4 block, $\lceil 352/4 \rceil \times \lceil (31+16-(-32))/4 \rceil =$ 1760 words are required for deriving inverse MVs with Level D data reuse scheme. Therefore, the size of searching range buffer in one frame is large enough, and it can greatly reduce the total system power and system memory bandwidth as described in Section V-A.2.

2) Hardware Architecture Reuse of FME Module: Fig. 15 shows the modified architecture of the FME module [23] to support the MC operations in the update stage. First, the interpolation circuit is modified to support from the unsigned interpolation to the signed interpolation since the input pixel value in update stage may be negative. *Energy Accumulator/Adaptive Weighting* is responsible for calculating the weighting factor and weighting the interpolated data to perform the computation of the update stage, as shown in (2), (3), and (4). One of 4×4 *Modified Block PE* is modified from the original PE to bypass the interpolated data and used as the data buffer of interpolated residual pixels until their corresponding energy weighting is derived. Finally, a multiplexer is required to select the MC result for the prediction or update stage.

VI. EXPERIMENTAL RESULTS

In this section, we will show the comparisons between our proposed scheme and the direct implementation for the prediction and update stages, respectively. Our target specification and supported coding schemes have been shown in Section III. Full search with VBSME and rate-distortion optimized mode decision are also supported. Note that because HB is the reordered 1/3 MCTF, they have the same requirements and then the related data of HB is not shown in the following discussion. Finally, the VLSI implementation result of this hardware accelerator will be given, and the whole system diagram is the same as Fig. 6.

A. Memory Bandwidth of Prediction Stage

Table II shows the memory bandwidth comparison of our proposed and the conventional schemes for the prediction stage in MCTF and MCP schemes. During our comparison, we assume that the Level C scheme [8] is applied for both schemes' MB-level data reuse within one frame so that the system memory bandwidth reduction by proposed frame-level data reuse can be demonstrated. Since Level C scheme is originally designed for P-frame with one reference frame, we take DRF scheme [10], which is the direct implementation for B-frame without any data reuse among reference frames, as the conventional scheme. For the proposed scheme, the techniques in the Section IV-A are used. The memory bandwidth of the prediction stage includes loading the searching range, current MB, and intermediate MC block, and outputting residual and MC block. From Table II, we can see that the reduction ratio of the proposed scheme in MCTF increases as the number of decomposition level increases. At four level decomposition, the proposed scheme can provide 20.4% memory bandwidth reduction. This is because fewer GOP boundaries exist, and the mEDCF scheme can be performed with fewer bandwidth overhead from inputs and outputs of MC blocks.

For the configuration of IBP with 2-ref, MCFI-S can provide 42.8% memory bandwidth reduction. For IBBP with 2-ref, sharing two reference frames between two B-frames has a 28.5% memory bandwidth reduction. Note that the reduction ratio between our proposed and original schemes increases, as the searching range is enlarged. That is, the larger the searching range is, the larger reduction ratio can be achieved.

B. Memory Bandwidth of Update Stage

Table III shows the memory bandwidth comparison of our proposed and the original schemes for the update stage in MCTF. The original scheme is the interleaved scheduling

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

TABLE III
EXTERNAL MEMORY BANDWIDTH COMPARISON FOR THE UPDATE STAGE

	1 Level	2 Level	3 Level	4 Level
Deriving MV				
Interleaved Scheduling (MBytes/s)	2.00	4.00	5.49	6.48
Proposed (MBytes/s)	0.63	1.26	1.73	2.05
Reduction	68.5%	68.5%	68.5%	68.5%
MC				
DRF Scheme (MBytes/s)	10.74	19.96	26.49	30.72
Proposed (MBytes/s)	6.04	10.58	13.61	15.50
Reduction	43.8%	47.0%	48.6%	49.5%
Total				
Conventional† (MBytes/s)	12.74	23.96	31.98	37.20
Proposed (MBytes/s)	6.67	11.84	15.34	17.55
Reduction	47.6%	50.6%	52.0%	52.8%

CIF 30 fps, and Searching Range: [−32, 32).
†: The combination of interleaved scheduling and DRF scheme.

TABLE IV
EXTERNAL MEMORY BANDWIDTH SUMMARY OF 5/3 MCTF

Coding Scheme	Original	Proposed	Reduction
1 Level (MBytes/s)	45.78	39.71	13.3%
2 Level (MBytes/s)	61.49	47.16	23.3%
3 Level (MBytes/s)	71.76	49.55	31.0%
4 Level (MBytes/s)	78.10	50.09	35.9%

and MC with DRF, and the proposed scheme is the pipelined scheduling and ME-like MC with Level C+.

In the deriving MV part, the proposed scheme can provide 68.5% memory bandwidth reduction. In the MC part, the reduction ratio of our proposed scheme increases as the number of decomposition level increases. This is also due to the effect of GOP boundary. The reduction ratio of our proposed ME-like MC with Level C+ scheme is 49.5% at four decomposition level. Totally, the proposed scheme for the update stage can achieve 52.8% reduction ratio at four decomposition level.

The memory bandwidth of 5/3 MCTF is summarized in Table IV. As the number of decomposition level increases, the proposed scheme has a better performance. In the direct implementation, the total memory bandwidth is 78.1 MBytes/s for four level decomposition. In our proposed scheme, only 50.09 MBytes/s are required and the reduction ratio is 35.9%. Moreover, the irregular memory access of update stage for deriving inverse MV and MC is eliminated so that the efficiency of accessing system memory can be largely improved.

C. Implementation Results and Summary

Table V summarized the required frequency and memory bandwidth for different coding schemes in our proposed hardware accelerator. The range of operating frequency is from 29.09 MHz for IPPP with 1-ref to 59.7 MHz for 5/3 MCTF with four level decomposition. The range of memory bandwidth is from 24.05 MBytes/s for IPPP with 1-ref to 50.09 MBytes/s for 5/3 MCTF with four level decomposition. The coding performance difference between the worst and best coding schemes is 4 dB, as shown in Fig. 2.

With system constraints like available memory bandwidth or operating frequency, the coding scheme which not only fits the constraints but also has a better coding performance can be chosen to perform in this hardware accelerator. For example, if the available memory bandwidth and working frequency is sufficient, 5/3 MCTF with four level decomposition can be performed to achieve the best coding performance. However, if the memory bandwidth is low, 1/3 MCTF can be selected to fit the constraint. When the available working frequency is hard limited, IPPP with 1-ref can be executed to sustain the normal function of video encoding.

TABLE V
SUMMARY OF WORKING FREQUENCY AND EXTERNAL MEMORY BANDWIDTH IN OUR PROPOSED COMPUTATION-AWARE TEMPORAL PREDICTION ENGINE

Coding Scheme	Memory Bandwidth (MBytes/s)	Required Frequency (MHz)
5/3 MCTF		
4 Level	50.09	59.70
3 Level	49.55	57.79
2 Level	47.16	53.52
1 Level	39.71	44.05
1/3 MCTF or HB		
4 Level	32.54	54.21
3 Level	34.21	52.94
2 Level	35.32	49.70
1 Level	33.04	41.82
MCP		
IBBP with 2-ref	30.04	52.78
IBP with 2-ref	24.05	52.80
IPPP with 2-ref	42.02	52.73
IPPP with 1-ref	24.05	29.09

TABLE VI
SPECIFICATION OF PROPOSED HARDWARE ACCELERATOR

Technology	TSMC 0.18um with Artisan Library
Package	208CQFP
Die Size	4.9398 mm x 4.9109 mm
Core Size	3.8240 mm x 3.5683 mm
Gate Count	352,405
On-Chip Memory	4 88x16b single port Register file : MVP buffer 16 240x32b dual ports SRAM : SR buffer 6144b register : CB and MC buffer
Working Frequency	60 MHz
Processing Ability	CIF 30fps, SR: [−32,32) @ 60 MHz, Support: 1–4 Level 5/3, 1/3 MCTF, HB IPPP, IBP, IBBP, with 2-ref
Power Consumption	469 mW @ 60 MHz, 1.8V

Table VI shows the implementation results of the proposed hardware architecture. TSMC 0.18-μm process with Artisan cell library is adopted, and the power consumption is estimated during chip testing on Agilent 93000 SOC Test System. The data listed in Table V are validated by both NC-Verilog simulator and porting our design on Virtex2 XC2V8000. The total gate count is 352 K gates, and the memory is 30 KBytes. The chip size is 3.8240 × 3.5683 mm^2. The maximum operating frequency is 60 MHz for 5/3 MCTF with four level decomposition. Fig. 16 shows the distribution of gate counts in the proposed hardware architecture. By use of the hardware reuse scheme, the hardware dedicated to the updated stage is of only 18.5 K gates. According to the implemental results in [17], we assume that the MC module itself occupies more than 60 K gates. It means about 76.4% hardware cost of update stage is saved. Fig. 17 shows the die photo of this hardware accelerator.

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

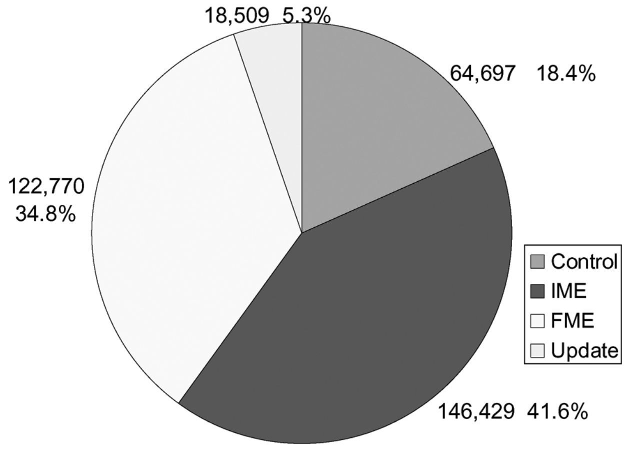


Fig. 16. Distribution of gate count in the proposed hardware accelerator.

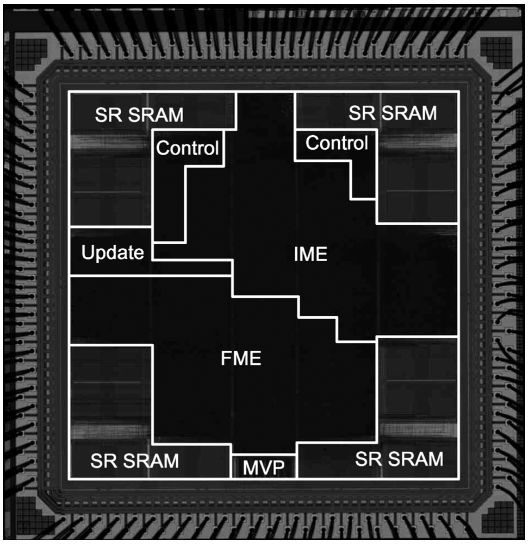


Fig. 17. Die photo of our proposed computation-aware temporal prediction engine.

VII. Conclusion

MCTF is a new temporal prediction scheme, which can provide 1 dB coding gain at least, compared to traditional MCP schemes. In this paper, an MCTF/ME-combined temporal prediction engine which can support both MCTF and MCP schemes is presented. For the prediction stage in MCTF, with the proposed frame-level data reuse schemes, the system memory bandwidth of prediction stage of MCTF and MCP can be reduced 20.4% and 42.8%, respectively. A frame-interleaved MB pipelining scheme is also proposed to eliminate the data buffer overhead induced by frame-level data reuse. As for the update stage, the proposed pipelined scheme for deriving inverse MVs and the proposed ME-like MC with Level C+ for MC can reduce 52.8% bandwidth, and the proposed hardware reuse strategy saves about 76.4% area of update stage. The implementation result shows that 352-K gates is required with 16.8 KBytes memory, where TSMC 0.18-μm with Artisan Library is employed, and the maximum operating frequency is 60 MHz. The processing ability is CIF 30 fps, and the searching range is $[-32, 32)$. The supported coding schemes include 5/3 MCTF, 1/3 MCTF, and HB frames with one-four decomposition level, IPPP with 1-ref, IPPP with 2-ref, IBP with 2-ref, and IBBP with 2-ref. By performing different coding schemes, this hardware accelerator can provide a computation scalability for various applications.

References

[1] D. Taubman, "Successive refinement of video: Fundamental issues, past efforts and new directions," in *Proc. Int. Symp. Vis. Commun. Image Process.*, 2003, pp. 791–805.

[2] L. Luo *et al.*, "Advanced motion threading for 3-D wavelet video coding," *Signal Process.: Image Commun.*, vol. 19, no. 7, pp. 601–616, Aug. 2004.

[3] *Scalable Video Model 3.0*, ISO/IEC JTC1/WG11 Doc. N6716, ISO/IEC JTC1, Oct. 2004.

[4] *Joint Scalable Video Model 8.0*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-U202, ISO/IEC JTC1, Oct. 2006.

[5] H. Schwarz, D. Marpe, and T. Wiegand, "MCTF and scalability extension of H.264/AVC," presented at the Picture Coding Symp., Dec. 2004.

[6] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, S.-Y. Chien, and L.-G. Chen, "System analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 4004–4014, Oct. 2006.

[7] T. Nishikawa *et al.*, "A 60 MHz 240 mW MPEG-4 video-phone LSI with 16 Mb embedded DRAM," in *Proc. IEEE Int. Solid-State Circuits Conference*, 2000, pp. 230–231.

[8] J.-C. Tuan, T.-S. Chang, and C.-W. Jen, "On the data reuse and memory bandwidth analysis for full-search block-matching VLSI architecture," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 1, pp. 61–72, Jan. 2002.

[9] *Comparison of MCTF and Closed-Loop Hierarchical b Pictures*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. JVT-P059, ISO/IEC JTC1, Jul. 2005.

[10] C.-T. Huang, C.-Y. Chen, Y.-H. Chen, and L.-G. Chen, "Memory analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2005, pp. V-93–V-96.

[11] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B pictures and MCTF," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2006, pp. 1929–1932.

[12] Y.-W. Huang, T.-C. Chen, C.-H. Tsai, C.-Y. Chen, T.-W. Chen, C.-S. Chen, C.-F. Shen, S.-Y. Ma, T.-C. Wang, B.-Y. Hsieh, H.-C. Fang, and L.-G. Chen, "A 1.3 tops H.264/AVC single-chip encoder for HDTV applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2005, pp. 128–588.

[13] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, C.-J. Lian, and L.-G. Chen, "System analysis of VLSI architecture for motion-compensated temporal filtering," in *Proc. IEEE Int. Conf. Image Process.*, 2005, pp. 992–995.

[14] C.-Y. Chen, Y.-H. Chen, C.-C. Cheng, and L.-G. Chen, "Frame-level data reuse for motion-compensated temporal filtering," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2006, pp. 5571–5574.

[15] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, "Analysis and design of macroblock pipelining for H.264/AVC VLSI architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2004, pp. 273–276.

[16] *Joint Scalable Video Model 2.0*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, Doc. N7084, ISO/IEC JTC1, Apr. 2005.

[17] T.-W. Chen, Y.-W. Huang, T.-C. Chen, Y.-H. Chen, C.-Y. Tsai, and L.-G. Chen, "Architecture design of H.264/AVC decoder with hybrid task pipelining for high definition videos," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, pp. 2931–2934.

[18] H. Kim and I.-C. Park, "High-performance and low-power memory-interface architecture for video processing applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 11, pp. 1160–1170, Nov. 2001.

[19] T. Takizawa and M. Hirasawa, "An efficient memory arbitration algorithm for a single chip MPEG2 AV decoder," *IEEE Trans. Consum. Electron.*, vol. 47, no. 3, pp. 660–665, Aug. 2001.

[20] *CE01: Reduced Memory MCTF*, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 Doc. JVT-P037, Joint Video Team of ISO/IEC MPEG and ITU-T VCEG, Jul. 2005.

[21] T.-M. Liu, T.-A. Lin, S.-Z. Wang, W.-P. Lee, K.-C. Hou, J.-Y. Yang, and C.-Y. Lee, "A 125 uW, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, 2006, pp. 1576–1585.

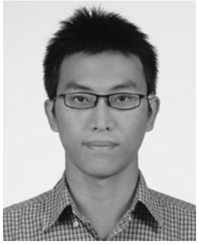
Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.

[22] C.-Y. Chen, C.-T. Huang, Y.-H. Chen, and L.-G. Chen, “Level C+ data reuse scheme for motion estimation with corresponding coding orders,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 16, no. 4, pp. 553–558, Apr. 2006.

[23] T.-C. Chen, Y.-W. Huang, and L.-G. Chen, “Fully utilized and reusable architecture for fractional motion estimation of H.264/AVC,” in *Proc. 2004 IEEE Int. Conf. Acoustics, Speech, Signal Process.*, May 2004, pp. V-9–V-12.

Yi-Hau Chen was born in Taipei, Taiwan, R.O.C., in 1981. He received the B.S.E.E degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 2003, where he is working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering.

His major research interests include the algorithm and related VLSI architectures of global/local motion estimation, H.264/AVC, and scalable video coding.



Chih-Chi Cheng was born in Taipei, Taiwan, R.O.C., in 1982. He received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 2004. where he is currently pursuing the Ph.D. degree at the Graduate Institute of Electronics Engineering.

His research interests include the algorithms and VLSI architectures of intelligent video signal processing and image/video coding.

Tzu-Der Chuang was born in Taipei, Taiwan, R.O.C., in 1983. He received the B.S.E.E degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., in 2005, where he is working toward the Ph.D. degree in the Graduate Institute of Electronics Engineering.

His major research interests include the algorithm and related VLSI architectures of H.264/AVC, scalable video coding.

Ching-Yeh Chen was born in Taipei, Taiwan, R.O.C., in 1980. He received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, R.O.C. in 2002, and 2006, respectively.

He joined MediaTek, Inc., Hsinchu, Taiwan, R.O.C., in 2006. His research interests include intelligent video signal processing, global/local motion estimation, scalable video coding, and associated VLSI architectures.

Shao-Yi Chien was born in Taipei, Taiwan, R.O.C., in 1977. He received the B.S. and Ph.D. degrees from the Department of Electrical Engineering, National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 1999 and 2003, respectively.

During 2003 to 2004, he was on the research staff in Quanta Research Institute, Tao Yuan Shien, Taiwan, R.O.C. In 2004, he joined the Graduate Institute of Electronics Engineering and Department of Electrical Engineering, National Taiwan University, as an Assistant Professor. His research interests include video segmentation algorithm, intelligent video coding technology, image processing, computer graphics, and associated VLSI architectures.

Liang-Gee Chen (S’84–M’86–SM’94–F’01) was born in Yun-Lin, Taiwan, R.O.C., 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 (NTU). During 1993 to 1994, he was a Visiting Consultant with the DSP Research Department, AT&T Bell Labs, Murray Hill, NJ. In 1997, he was a Visiting Scholar in 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), NTU. Currently, he is a Professor in the Department of Electrical Engineering and GIEE, NTU, Taipei, Taiwan. He is also Director of the Electronics Research and Service Organization, Industrial Technology Research Institute, Hsinchu, Taiwan. His current research interests are digital signal-processing architecture design, video processor design, and video coding systems. He has been 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.*

Dr. Chen is a member of Phi Tan Phi. He has been an Associate Editor of the IEEE Transactions on Circuits and Systems for Video Technology since 1996, as Associate Editor of the IEEE Transactions on VLSI Systems since 1999, and an Associate Editor of the IEEE Transactions on Circuits and Systems—II: Analog and Digital Signal Processing since 2000. He is also an Associate Editor of the Proceedings of the IEEE. He was 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 the Taipei Chapter of the IEEE Circuits and Systems (CAS) Society. He 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 was 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. In 2000, the he received the Dragon Excellence Award from Acer.

Authorized licensed use limited to: National Taiwan University. Downloaded on January 16, 2009 at 00:10 from IEEE Xplore. Restrictions apply.