A 135 MHz 542 k Gates High Throughput H.264/AVC Scalable High Profile Decoder

Gwo-Long Li, Yu-Chen Chen, Yuan-Hsin Liao, Po-Yuan Hsu, Meng-Hsun Wen, and Tian-Sheuan Chang, *Senior Member, IEEE*

***Abstract*—To satisfy the requirement of application heterogeneities, the latest H.264/AVC based video coding standard called scalable video coding additional includes temporal, SNR, and spatial scalabilities for frame rate, quality, and frame resolution adaptation. However, these inclusions significantly increase chip design difficulties such as decoding time, memory bandwidth, and area cost. This paper presents an H.264/AVC scalable high profile decoder realization with several optimization techniques to provide high throughput video decoding. For decoding flow, this paper proposes an one-pass macroblock-based quality layer decoding flow for SNR scalability and 71% of external memory bandwidth and 66% of macroblock processing cycles can be saved. For texture padding in interlayer intra prediction, the modified padding flow can save 26% of decoding time. For interlayer predictor design, this paper proposes a centralized concept for accumulation-based calculation of corresponding spatial position, simplified poly-phase interpolator, and efficient motion vector generator to save area cost and decoding time. Furthermore, the residual reconstruction path with the parallel-pipeline architecture is also proposed to cope with the additional decoding complexity and thus leads to 54% of gate count savings compared to the traditional serial-pipeline architecture. Finally, the proposed H.264/AVC scalable high profile decoder design is implemented with 90 nm CMOS technology and it costs 542 k gate count and 39.66 Kbytes on-chip memory while is capable to decode 60 frames/s for CIF+SD480p+HD1080p resolution with three quality layers at 135 MHz operating frequency.**

***Index Terms*—Scalable video coding (SVC), SVC decoder, very large scale integration (VLSI) design.**

I. Introduction

WITH THE PROSPERITY of portable devices, digital televisions, and internet videos, video bitstreams are required to fit different video size, quality, and frame rate. To fit above needs in a unified way, an extension of H.264/AVC called scalable video coding (SVC) [1] has been standardized recently which can encode the video sources with diverse qualities into single bitstream and thus achieve the temporal, spatial, and quality scalabilities [2]. Fig. 1 shows the block diagram of an SVC decoder which includes the base layer of H.264/AVC and the scalable extension; in which the extracted bitstream with the necessary scalability layer is decoded by an entropy decoder for the following reconstruction processes. The reconstruction process uses the existing H.264/AVC standard coding tools like intra prediction and motion compensation for the base layer reconstruction and for the spatial and temporal scalability as well. The quality refinement process derives different quality layer information by means of accumulating scaled transform coefficients successively. Afterwards, the base layer information will be upsampled by the corresponding interlayer prediction (ILP). Once all predictions have been derived successfully, the pixel samples are reconstructed by adding residuals to those predicted samples. Finally, the deblocking filter is applied to remove the blocking effects.

Above advanced techniques for SVC significantly increase the design complexities beyond the intrinsic complexity of H.264/AVC. Therefore, many researches were published recently to address its real time hardware design issues. First, the memory storage issue due to additional SVC data dependency is addressed in [3] and [4]. Reference [3] analyzed the spatial layer decoding flow and concluded that the frame-based decoding flow is the most memory efficient one. Reference [4] proposed a memory architecture for SVC with focus on reduction of on-chip memory size. For interlayer prediction, [5] proposed a cost efficient residual prediction hardware architecture for encoding. However, these works only addressed parts of component design instead of overall SVC system for optimization. For the whole chip integration, designs [6]–[11] implemented and optimized H.264/AVC only decoders with focus on the power consumption, on-chip memory demand, or gate count issues. However, although these literatures can support high visual quality applications, the scalabilities specified in SVC were not supported in these literatures so that no application adaptations can be achieved. References [12] and [13] were the only two published works about the integration of a SVC decoder. However, these designs did not support combined scalabilities and only supported 30 f/s frame rate which is not enough for high visual quality video applications.

Manuscript received March 18, 2011; revised June 13, 2011 and July 26, 2011; accepted September 10, 2011. Date of publication October 10, 2011; date of current version April 2, 2012. This paper was recommended by Associate Editor R. C. Lancini.

G.-L. Li is with the Industrial Technology Research Institute, Hsinchu 31040, Taiwan (e-mail: glli@itri.org.tw).

Y.-C. Chen is with the VLSI Signal Processing Laboratory, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: ycchen@dragons.ee.nctu.edu.tw).

Y.-H. Liao and M.-H. Wen are with PixArt Imagine, Inc., Hsinchu 300, Taiwan (e-mail: yhliao@dragons.ee.nctu.edu.tw; mhwen@dragons.ee.nctu.edu.tw).

P.-Y. Hsu and T.-S. Chang are with the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: pyhsu@dra gons.ee.nctu.edu.tw; tschang@dragons.ee.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2011.2171213

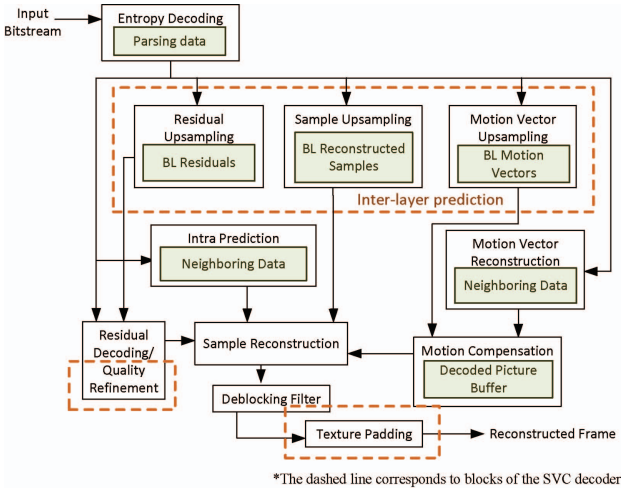

Fig. 1. Block diagram of a SVC decoder.

To achieve high performance video decoding, we propose a H.264/AVC scalable high profile decoder with several advanced techniques to reduce the hardware costs and memory requirements. The proposed decoder not only supports main features specified in scalable high profile but also attains the following processing specifications:

1) decoding of H.264/AVC scalable high profile;
2) at most three spatial layers from QCIF to HD1080p, CIF+SD480p+HD1080p at 60 f/s (equivalent to 4096×2160 at 51 f/s) for high visual quality applications;
3) at most three quality layers for any QP value setting;
4) all GOP sizes smaller than or equal to eight;
5) arbitrary spatial resolution ratios between spatial layers with extended spatial scalability (ESS).

The rest of this paper is organized as follows. Section II presents the analysis of our proposed H.264/AVC scalable high profile decoder and the overall hardware architecture design. The detailed architecture is described in Section III. Section IV shows implementation results and comparisons with other works. Finally, a conclusion is made in Section V.

II. Analysis and Architecture Overview of the Proposed H.264/AVC Scalable High Profile Decoder

The design performance and cost of a video decoder mainly depend on the adopted decoding flow and corresponding memory access. In the following, we will first analyze different decoding flows and select the most memory efficient one with the consideration of memory access. With the selected flow, we will present the proposed four-stage pipelined H.264/AVC scalable high profile decoder.

A. Analysis of Decoding Flow

For the three scalabilities of SVC, the temporal scalability adopts the hierarchical-B structure and thus its decoding flow is the same as the traditional B-slice decoding. The

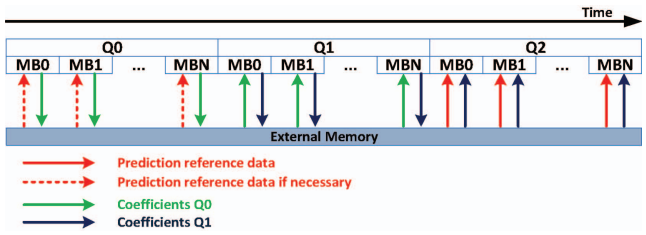

Fig. 2. Frame-based quality layer decoding flow.

spatial scalability with interlayer prediction is the most distinct part between H.264/AVC and SVC. In our previous analysis [3], the frame-based spatial decoding flow has better performance than other coding approaches like row-based or macroblock (MB)-based approach. Therefore, the frame-based spatial decoding flow will be adopted here for this design.

In SVC, quality scalability with coarse-grain scalability will sum the coefficients from the base layer and the coefficient differences in the enhancement layer decoded by the entropy decoder for reconstruction. The main design challenge is the complexity due to the increasing number of quality layers. To deal with all quality layer coefficients, the decoding flow plays an important role in both processing time and memory access aspects. Therefore, we will show the analysis for two commonly used decoding flows: frame-based and MB-based flow in the following and then present our proposed one pass quality layer decoding flow for a memory efficient design.

Fig. 2 shows frame-based quality layer decoding approach, in which enhancement layer frames are reconstructed after the frames in the base layer have been decoded. However, this flow introduces significant external memory access overheads since the required reference data must be loaded (I/P slice in top spatial layer) for generating the predictions. In this case, the external memory access for predicted reference data will be doubled when supporting multiple quality layers. In addition, the base layer coefficients also contribute the external memory access overheads since they should be stored for later reference in the enhancement layer. In summary, the external memory space of $8160 \times 384 \times 2$ bytes is required for the frame size of HD1080p.

An alternative way is the MB-based decoding approach [14]. In this method, the coefficients of different quality layers of the same MB are reconstructed in a successive order as shown in Fig. 3. Thus, it is no need to store the quality coefficients into external memory since they will be referenced immediately within the same MB decoding process. Instead, the internal memory is used to store the transform coefficients (384×2 bytes) for quality enhancement layer reconstruction. As a result, no external memory access for quality coefficients is required in the MB-based quality layer decoding approach. The only data to be accessed from external memory is the reference data for generating the predicted pixels.

Although the external memory access overheads can be saved in the MB-based decoding flow, the MB-based decoding approach still suffers from the long decoding latency problem due to entropy decoding and the residuals reconstruction of additional quality layers. Fortunately, all of the coefficients

TABLE I
MEMORY BANDWIDTH REQUIREMENT FOR DIFFERENT QUALITY DECODING FLOW

Sequence	Coding Flow	Prediction Prediction Reference	Quality Coefficients	Others**	Total
Blue-Sky	Frame-based	31.51 MB/s	160.9 MB/s	43.48 MB/s	235.89 MB/s
	MB-based	31.51 MB/s	0	43.48 MB/s	74.99 MB/s
	One-pass	24.83 MB/s	0	43.48 MB/s	68.31 MB/s
Tractor	Frame-based	33.14 MB/s	160.9 MB/s	43.53 MB/s	237.57 MB/s
	MB-based	33.14 MB/s	0	43.53 MB/s	76.67 MB/s
	One-pass	25.63 MB/s	0	43.53 MB/s	69.16 MB/s
Pedestrian-area	Frame-based	33.31 MB/s	160.9 MB/s	43.32 MB/s	237.53 MB/s
	MB-based	33.31 MB/s	0	43.32 MB/s	76.63 MB/s
	One-pass	25.59 MB/s	0	43.32 MB/s	68.91 MB/s
Average	Frame-based	32.65 MB/s	160.9 MB/s	43.44 MB/s	237 MB/s
	MB-based	32.65 MB/s	0	43.44 MB/s	76 MB/s
	One-pass	25.35 MB/s	0	43.44 MB/s	69 MB/s

*GOP: 8/frame-rate: 30 f/s/QP: 12-22-32/frame-size: CIF-480p.
**Others: the interlayer data or frame pixels to be written out.

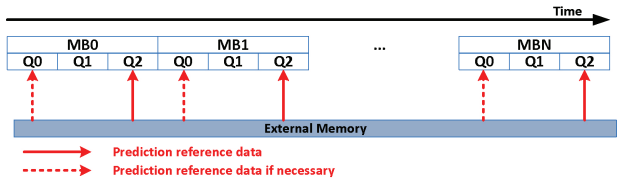

Fig. 3. MB-based quality layer decoding flow.

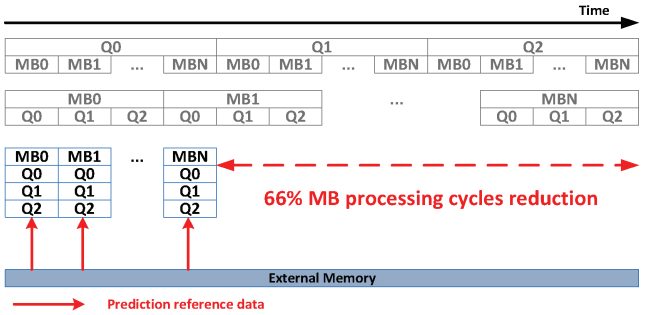

Fig. 4. Proposed one-pass quality layer decoding flow.

in each quality layer can be parsed individually since the entropy decoding process of each quality layer is independent. Therefore, the parallel entropy decoding mechanism is adopted here to achieve higher entropy decoding throughput. In addition, with the combined scalability concept [15], the residual reconstruction path can be integrated into single MB processing. With above flow, we propose an one-pass quality layer decoding, in which the quality decoding can be done in one pass for all quality layers as shown in Fig. 4. Compared to the frame-based and MB-based decoding flows, 66% (only 1/3 processing cycles are required) of cycle can be reduced to support three quality layers since our proposed one-pass quality layer decoding flow can process three quality layers at the same time but the frame-based and MB-based decoding flows only decode the quality layer one by one. Furthermore, since the quality layers are processed in parallel, the prediction generation can be processed in parallel with residual reconstruction and thus lead to the single-loop prediction generation. Therefore, the prediction reference data (i.e., reference pixels for motion compensation, base layer texture, base layer residual, and so on) are only fetched once from external memory in single MB processing.

Table I shows the external memory bandwidth requirements for above quality decoding flows and the results are derived by calculating the required data amount from the simulation of reference software JSVM9.14. The proposed one-pass quality decoding flow can, respectively, achieve 77.5% and 66% of memory bandwidth and processing cycle savings on average when compared to the frame-based quality decoding flow, as shown in Fig. 5.

In summary, the most memory efficient decoding flow will be the frame-based spatial layer decoding with one-pass quality layer decoding, which will be adopted in our architecture design.

B. Overview of Proposed Architecture

With above proposed decoding flow, Fig. 6 shows the overall four-stage pipeline architecture of our high throughput H.264/AVC scalable high profile decoder, which will be briefly described as follows.

The first stage is parallelized entropy decoding components and a syntax parser, in which at most three DCT coefficients from three quality layers are parsed in this stage to support one-pass quality layer decoding. However, by adopting our previous proposed high throughput entropy decoding [16], [17], we need only two sets of context-based adaptive variable length coding (CAVLC)/context-based adaptive binary arithmetic coding (CABAC) decoders, which result in less hardware cost. Besides, the motion vector difference (MVD) is also derived in this stage and will be passed to the next stage along with parsed syntax parameters.

The second stage is organized by the residual reconstruction path, interlayer predictor, motion vector generator, and other reconstruction elements. In this stage, the coefficients received from the first stage are fed into the reconstruction path, such as the inverse quantization, coefficients refinement, inverse transform, and residual accumulation pipeline chain,

TABLE II
COMPARISON OF SERIAL- AND PARALLEL-PIPELINE STRATEGY

Item		Serial-Pipeline	Parallel-Pipeline
Inverse quantization		9327	4724 × 3
Coefficients refinement		314	155 × 3
Inverse transform		23 124	12 137 × 3
Residual accumulation		314	155 × 2
Residual/texture Interpolation	Input selector	5901 × 20	5901 × 16
	Basic interpolator	Ver. 1281 × 20	Ver. 1281 × 16
		Hor. 1816 × 8	Hor. 1816 × 4
Total		191 247	173 379
Memory requirement		768 Bytes	0

*Synthesized by 90 nm CMOS process at 135 MHz operating frequency.

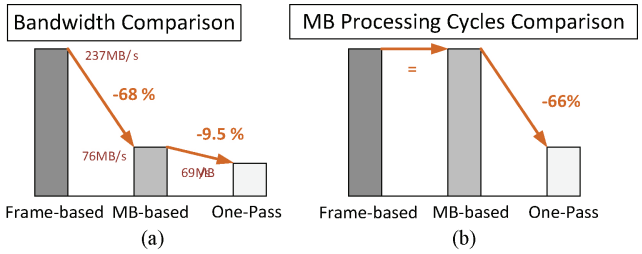

Fig. 5. Comparison of quality decoding flows in (a) memory bandwidth and (b) MB processing cycles.

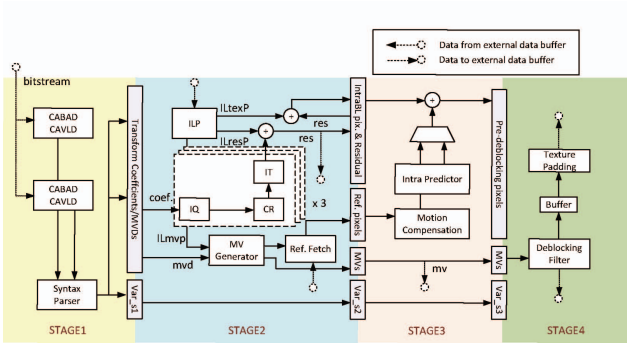

Fig. 6. Architecture of proposed decoder.

to derive residual information. In this path, triple sets of reconstruction pipeline chains are used to produce different quality layer residuals in parallel for one pass quality layer decoding. The ILP module generates interlayer predictions by upsampling the information of the base layer, including motion vectors, reconstructed pixels, and residuals. The ILP interpolator will be involved in the pipeline chain to obtain the final residuals if the residual prediction mode is applied to current MB. Besides, MV generator generates the motion vectors for the bi-directional reference of current MB. With the derived motion vectors and partition sizes, reference pixels for motion compensation can be fetched at once.

The third stage will generate the predicted pixels of both intercoded and intracoded modes. The produced predictions are then added with residuals from the second stage to form the predeblocking samples. In the fourth stage these samples are filtered by the deblocking filter, and padded by the texture padding module for interlayer upsampling. In this design, each pipeline stage is separated by pipelined ping-pong buffers for interleaved read or write.

To support high visual quality application with reasonable operating frequency, we will discuss the cycle limit for designing our system. Instead of designing our system with the highest operating frequency possible, we finally select the operating frequency of 135 MHz for our system since it is multiple of 27 MHz (a commonly used base frequency in current video system design) and is also easily achievable with the modern 0.13 μm or 90 nm CMOS technology. With above designs and the target specification, the cycle limit for each pipeline stage will be

$$227 = (135\times10^{6})/[396 + 1350 + 8160 \times 60] \quad (1)$$

Frequency total MBs in one frame frame rate

cycles for single MB decoding at the selected 135 MHz operation frequency. Under the available clock cycle counts constraint, the optimization techniques are hence proposed with suitable parallelism and pipelining.

III. DETAILED ARCHITECTURES OF THE PROPOSED DECODER

A. First Stage Design

In our decoder design, the first stage is composed by entropy decoding including syntax parser, CAVLC, and CABAC decoder. Fig. 7 shows the system level architecture of the proposed entropy decoder, in which dual hardware units of our previous works [16], [17] are adopted here to support high decoding throughput. In addition, to support the extra adopted scalabilities in SVC, we have made some modifications for the overall entropy decoding process.

First, we adopt a bitstream scanner to quickly detect the start points of quality enhancement layers and nonquality enhancement layers by recognizing the start code 0×00000001 to make two entropy decoding engines work in parallel. Afterwards, the addresses are transmitted to the bitstream fetcher to access the distinguished quality enhancement layer bitstream from the nonquality enhancement layer bitstream.

Second, to reduce hardware cost overhead, one CABAC decoder for the quality enhancement layer is simplified by removing the unused context models in quality enhancement layers since the MB information is inherited from the quality base layer when decoding the quality enhancement layers. As

TABLE III
SYMMETRY OF COEFFICIENT TABLE: (A) BI-LINEAR FILTER, (B) 4-TAP FILTER

(a)

Phase	Coefficients	
	C_0	C_1
0	32	0
8	16	16
1	30	2
15	30	2
2	28	4
14	28	4
3	26	6
13	26	6
4	24	8
12	24	8
5	22	10
11	22	10
6	20	12
10	20	12
7	18	14
9	18	14

(b)

Phase	Interpolation Filter Coefficients			
	C_{-1}	C_0	C_1	C_2
0	0	32	0	0
8	−3	19	19	−3
1	−1	32	2	−1
15	−1	2	32	−1
2	−2	31	4	−1
14	−1	4	31	−2
3	−3	30	6	−1
13	−1	6	30	−3
4	−3	28	8	−1
12	−1	8	28	−3
5	−4	26	11	−1
11	−1	11	26	−4
6	−4	24	14	−2
10	−2	14	24	−4
7	−3	22	16	−3
9	−3	16	22	−3

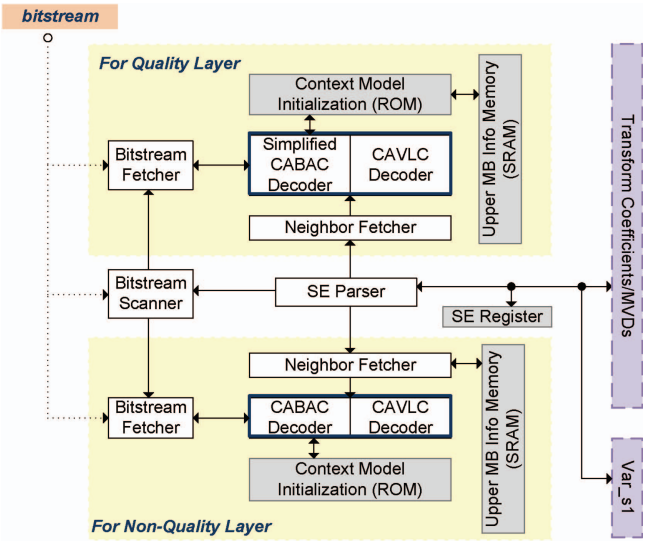

Fig. 7. Detailed framework of proposed entropy decoder.

a result, 120×99 bits memory space can be saved for storing upper MB information such as *mvd* and *mb_type*. As to the CAVLC decoder engine, no simplification can be applied since it is designed for decoding residual block information.

B. Second Stage Design

In this stage, the interlayer prediction and related prediction modules are realized to derive the prediction information and residuals for the following reconstruction usage. Interlayer prediction includes interlayer texture prediction, interlayer residual prediction, and interlayer motion prediction. Moreover, to support nondyadic spatial scalability, a more complex mechanism, ESS, [18] has been adopted to achieve nondyadic interlayer prediction. The interlayer prediction process with ESS includes two steps. The first step is the operation called "calculation of corresponding spatial positions (CCSP)" which

TABLE IV
INPUT SELECTION AND CLASSIFICATION FOR ADDERS

Mode	Phase	Adder A			Adder B			Adder C		Adder D		
		in0	in1	in2	in0	in1	in2	in0	in1	in0	in1	in2
0	0	16b	16b	0	0	0	0	0	0	A	B	C
0	1	16b	16b	0	2c	0	0	−a	−d	A	B	C
0	2	16b	16b	−b	0	4c	0	−2a	−d	A	B	C
0	3	16b	16b	−2b	2c	4c	−d	−a	−2a	A	B	C
0	4	16b	16b	−4b	0	8c	−d	−a	−2a	A	B	C
0	5	16b	16b	2b	2c	8c	c	−4a	−d	A	B	C
0	6	16b	16b	0	2c	8c	4c	−4a	−2d	A	B	C
0	7	16b	16b	0	2d	16c	−d	−a	−2a	A	B	C
0	8	C	2C	16B	0	b	c	−a	−d	A	B	2B
1	0	16b	16b	0	0	0	0	0	0	A	B	C
1	1	16b	16b	−2b	0	0	2c	0	0	A	B	C
1	2	16b	8b	4b	4c	0	0	0	0	A	B	C
1	3	16b	8b	2b	4c	0	2c	0	0	A	B	C
1	4	16b	8b	0	0	8c	0	0	0	A	B	C
1	5	16b	2b	4b	0	8c	2c	0	0	A	B	C
1	6	16b	0	4b	4c	8c	0	0	0	A	B	C
1	7	16b	2b	0	4c	8c	2c	0	0	A	B	C
1	8	8b	0	0	0	8c	0	0	0	A	B	C
		Set A			*Set B*			*Set C*				

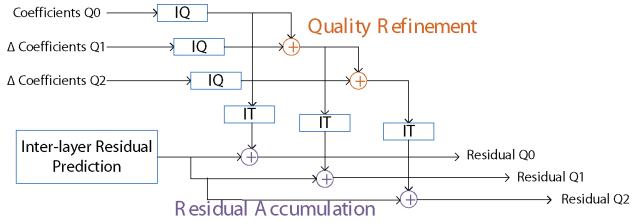

Fig. 8. Residual reconstruction flow for SNR scalability.

computes the corresponding position in the base layer (BL). The second step executes the interlayer intra or residual prediction. In the following subsections, the detail design principles of our proposed ILP module will be described.

1) *Interlayer Residual Prediction Module (ILresP):* Fig. 8 shows the typical residual reconstruction flow for SNR scalability. Coefficients in the enhancement layer are reconstructed by summing the delta coefficients from the previous quality layers. Then, these coefficients are summed with the interlayer residuals to derive the residuals of different quality layers. This multilevel coefficient summation path makes the video quality scalable for different requirements but at the cost of complex reconstruction path.

To speed up above reconstruction, a straightforward design is to use pipelining, as shown in Fig. 9. Each pipeline stage deals with a block of samples and then passes the results to next stage in every cycle. After first four cycles, the residuals of a block from the same MB are generated in successive cycles.

A straightforward pipelined design called the serial-pipeline strategy is to reconstruct quality refinement coefficients from different layers in serial order within different timing intervals. By reusing the computations in every quality layer, only one set of pipeline processing unit is required. However, a size of 384×2 bytes coefficient refinement buffer is required to restore the coefficients of quality base layer. In addition, to

TABLE V
LIST OF GATE COUNT OF PROPOSED DECODER

Module	Gate Counts
Entropy decoder + syntax parser	213 638
Motion compensation	107 080
Deblocking filter	24 573
Interlayer prediction	87 178
- Centralized CCSP	14 312
- Texture/residual upsample	43 032
- MV upsample	6703
- External data buffer	23 131
Residual reconstruction	56 184
- Inverse DCT and Hadamard transform	36 117
- Inverse quantization	15 972
- Reconstruction + control	4095
Intraprediction	28 326
-Prediction generator + control	20 836
- Neighboring pixel buffer	7490
Texture padding	9870
- Padding unit	4398
- Neighboring pixel buffer	5472
Memory controller	12 678
System control	2001
Total	541 528

*Synthesized by UMC90 at 135 MHz.

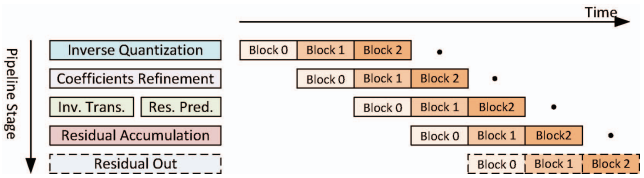

Fig. 9. Pipeline chain of residual reconstruction.

meet 227 cycles timing constraint, the processing throughput has to be eight in serial-pipeline strategy.

In this paper, we adopt the 4-pixel/cycle throughput parallel-pipeline strategy that triples the pipeline chains to separately reconstruct the residuals of different quality layers as shown in Fig. 10. Table II lists the synthesis gate counts and memory requirements of major components in both strategies. From Table II, we can observe that the gate counts of the serial-parallel strategy are larger than that of the parallel-pipeline strategy. This situation can be analyzed as follows. First, to meet a 227-cycle timing constraint in our design, 8-pixels/cycle throughput has to be supported in the serial-pipeline strategy, but only 4-pixels/cycle for the parallel-pipeline strategy. Therefore, the hardware cost of the serial-pipeline strategy is much higher than that of the parallel-pipeline strategy from the throughput perspective. Besides, although components such as inverse quantization and inverse transform are tripled in different quality layers, the largest area component, the interlayer prediction module, is not tripled in the parallel-pipeline strategy. That is because all quality layers use the same contents as their prediction with the combined scalability. Thus, only one set of interpolator is required for the parallel reconstruction for different quality layers. In addition, by simplifying the most complex part in the interlayer interpolator, total gate counts in the interpolation

TABLE VI
LIST OF SRAM REQUIREMENT OF PROPOSED DECODER (UNIT: KBYTE)

Module	SRAM Requirement	
	Single Port	Dual/Two Port
Entropy decoder + syntax parser	2.039	3.411
Motion compensation	2.461	
Deblocking filter	7.461	1.218
- Neighboring pixels (Q0+Qmax)	7.374	
- Others	0.087	1.218
Interlayer prediction	4	
Intraprediction	4.116	
- Luma neighboring data	2.179	
- Chroma neighboring pixels	1.937	
Texture padding	0.937	
Pipeline ping-pong buffer	12.893	1.125
- Stage1	4.269	
-_$Coefficients(Q0+Q1+Q2)$	4.113	
- MVDs	0.156	
- Stage2	6.968	1.125
- Residuals (Q0+Qmax)	0.562	1.125
-MC reference pixels	6.250	
- MVs	0.156	
- Stage3	1.656	
-_$Reconstructed_Pels(Q0+Qmax)$	1.50	
- MVs	0.156	
Total	33.907	5.754
Total (single port + two/dual port)	39.661	

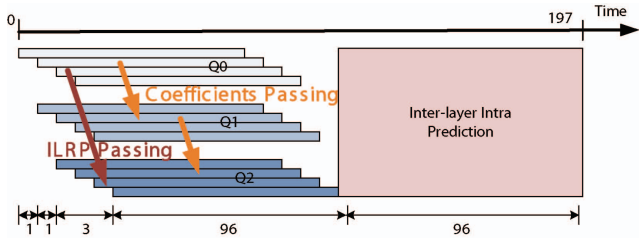

Fig. 10. Parallel-pipeline chain for quality layer processing.

are significantly reduced in spite of the increasing gate counts of pipelined components. Furthermore, the memory usage is saved in the parallel-pipeline strategy via the coefficient passing technique. As a result, the parallel-pipeline method is adopted in this paper.

2) *Proposed Centralized Accumulation-Based CCSP Engine:* Compared to the dyadic spatial scalability, the nondyadic one suffers from the issue of indirect position mapping between two spatial layers. ESS adopts the CCSP scheme to map the corresponding samples between two successive layers. This paper adopts the simplification form of the CCSP operation from [18]. As a result, the area cost of the CCSP can be reduced to be simple accumulators, which is a more efficient way in hardware architecture design.

Second, the repeated CCSP operations can be eliminated to save the computation complexity. The CCSP operation is to calculate the corresponding position of current MB in the BL by generating some spatial information such as interpolation phase and corresponding MB partition. In the traditional interlayer prediction flow, the CCSP would be executed several times to generate the corresponding spatial

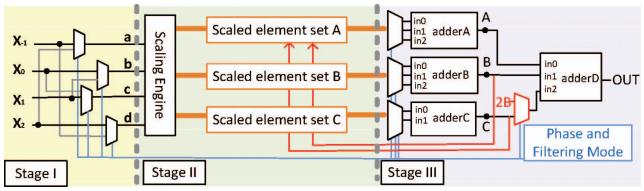

Fig. 11. Architecture of the proposed interpolator.

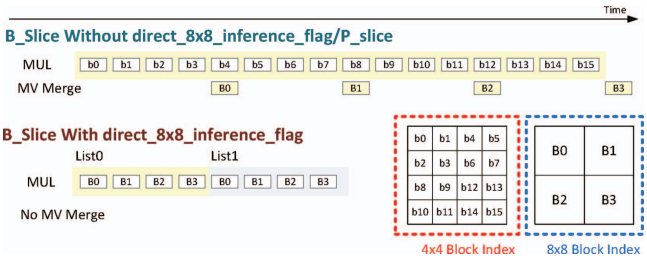

Fig. 12. Timing schedule of MV generator.

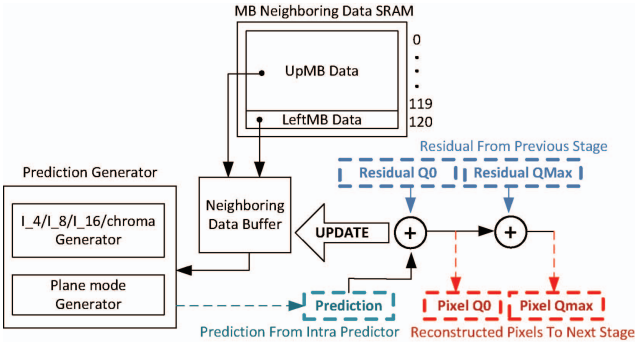

Fig. 13. Data path of intra prediction.

parameters for various interlayer prediction modes. However, since the spatial parameters mentioned above are all the same for all interlayer prediction modes when decoding a MB, these spatial parameters can be stored temporarily for the following decoding usage and thus the repeated computations for deriving the spatial parameters can be eliminated. Therefore, we propose a centralized CCSP approach to remove these repeated computations. In the proposed strategy, the accumulation-based CCSP is only executed once to generate spatial parameters including interpolation phase and the corresponding MB partition, and the generated spatial parameters are stored internally in the accumulation-based CCSP engine for the further reuse purpose. Afterwards, for all interlayer prediction modes, the stored spatial parameters can be freely used without recalculation and thus result in the flexible prediction flow and decoding time improvement.

3) *Shared Interpolation Components for Interlayer Intra and Residual Prediction:* Both of interlayer intra and interlayer residual prediction use the poly-phase 4-tap filter and bilinear filter to generate the prediction samples for luma and chroma signals, respectively. These filter coefficients have the symmetrical property as listed in Table III, in which the Phase indicates the index of interpolation coefficient set used in the following interpolation operation. Thus, half of contents in the table can be reduced by exchanging inputs with symmetric coefficient columns in the table for hardware realization.

Fig. 11 shows our proposed hybrid interpolation module with three pipeline stages to implement 4-tap and bilinear interpolation in a shared adder tree. In Stage I, reference samples are rearranged according to the interpolation phase and filtering mode as listed in the above table. In Stage II, the scaling engine produces scaled elements and classifies them to three sets. The classifying strategy is listed in Table IV. Finally, Stage III uses a simple two-level adder-tree architecture to compute interpolation values.

4) *MV Generator for Interlayer Motion Prediction:* To efficiently exploit the relationship between interlayers, SVC adopts the interlayer motion prediction mechanism by upsampling motion vector if the spatial resolution changes. The whole process first finds the corresponding motion vectors in the base layer, and upsamples the motion vector as follows:

$$MVEL = (MVBL \times Dmv + 32\,768) \gg 16 \qquad (2)$$

where *MVBL* is the reference motion vector in the base layer, and *Dmv* is the spatial parameter which represents the ratio between spatial layers. *MVEL* indicates the target derived motion vector in the current MB.

To implement motion vector upsampling, we use two multipliers to realize (2) for the *MVx* and *MVy* in the proposed MV generator, which takes 16 cycles to scale the motion vectors in a reference list. Afterwards, the scaled motion vectors and the reference indexes are refined and merged by profiling their similarities to decide new partitions and MVs and thus avoiding inconsistency. Furthermore, the MV upsampling can be accelerated by the identification of $direct\_8\times8\_inference\_flag$. This optional flag is used to signal whether the other three motion vectors within the same 8×8 block should be set to the motion vector of corner one in bi-direction slice type frame. Therefore, by using $direct\_8\times8\_inference\_flag$, only eight motion vectors are required to be upsampled and the MV merge step is skipped due to the motion vectors are already integrated. Fig. 12 shows the timing schedule of the proposed MV generator determined by the slice type and $direct\_8 \times 8\_inference\_flag$. With flag identification, the processing time is thus saved by the adaptive scheduling.

C. Third Stage Design

This stage reconstructs the sample pixels by using the decoded residual and prediction data from previous stages through inverse intra prediction and motion compensation as described as follows.

1) *Intra Prediction:* Fig. 13 shows the architecture of the proposed intra predictor in which four-pixel parallelism is adopted to achieve best tradeoff between hardware cost and the cycle budget. With the required neighboring data, prediction of current MB can be generated in a sequential order. The prediction data would be added with residuals to form the reconstructed pixels. During the reconstruction process, the reconstructed data of the lowest quality layer will be updated to the neighboring buffer for the un-processed blocks. Besides, with these reconstructed samples, the highest quality layer residuals can be formed by a simple accumulation. The resid-

TABLE VII
COMPARISON WITH OTHER STATE-OF-THE-ART VIDEO DECODERS

	[7]	[8]	[11]	[12]	[13]	Proposed
Technology	0.18 μm	0.18 μm	0.09 μm	0.03 μm	0.09 μm	0.09 μm
Max clock rate	100 MHz	120 MHz	175 MHz	163 MHz	210 MHz	135 MHz
Profile	MPEG-2 SP, H.264 L4	H.264 BP/MP	H.264 High	MPEG-2 MP, H.264 HP, SVC SBP	SVC HP MVC HP	SVC HP at L5
Max spec. (H.264)	1920×1088 at 30 f/s	1920×1088 at 30 f/s	4096×2160 at 60 f/s	1920×1088 at 30 f/s	4096×2160 at 24 f/s	1920×1088 at 60 f/s
Max spec. (SVC)	N/A	N/A	N/A	N/A	[1]SpatialScalability [2]SNRScalability	[3]CombinedScalability
Gate count	303.78 K	160 K	662 K	439 K	414.28 K	541.52 K
Internal memory	22.75 Kbtyes	4.5 Kbtyes	59.6 Kbytes	10.9 Kbytes	8.99 Kbtyes	39.66 Kbtyes
Max throughput	244 800 MB/s	244 800 MB/s	2 073 600 MB/s	244 800 MB/s	979 200 MB/s	1 783 080 MB/s
Gate efficiency	805.84 MB/Kgates-s	1530 MB/Kgates-s	3132.33 MB/Kgates-s	557.63 MB/Kgates-s	2363.62 MB/Kgates-s	3292.68 MB/Kgates-s

Max throughput = frame rate × processing MBs in (spatial + quality) layers.
[1]SpatialScalability: (1920×1088)+(1280×720) at 30 f/s.
[2]SNRScalability: (1920×1088)×4 quality layers at 30 f/s.
[3]CombinedScalability: [(1920×1088)+(720×480)+(352×288)]×3 quality layers at 60 f/s.

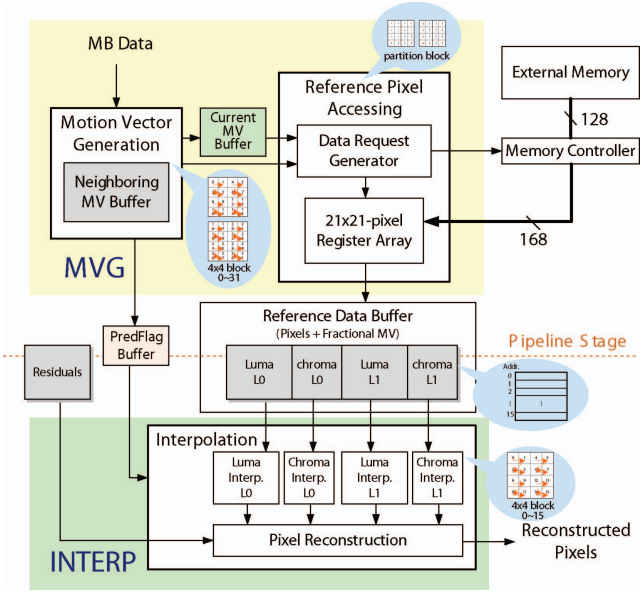

Fig. 14. Proposed pipeline architecture of motion compensation design.

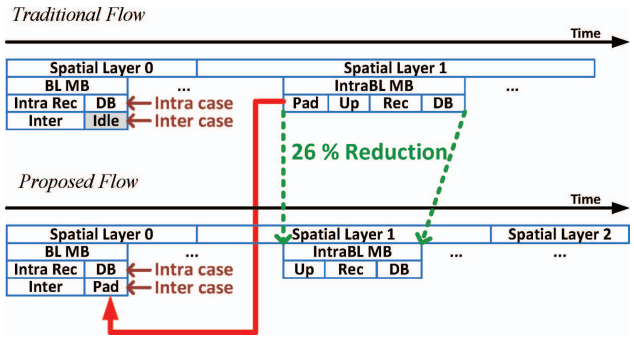

Fig. 15. Proposed combined BL-level padding and deblocking flow.

uals with different quality layers are accessed in parallel in this paper and thus the reconstruction of two different quality layers can be processed within the same cycle.

2) *Motion Compensation:* The high memory bandwidth requirement in motion compensation is also the bottleneck in a video decoder design. Therefore, reusing the overlapped data inside a partitioned block [19]–[23] can solve this problem. To efficiently reduce memory bandwidth with less hardware complexity and cost, this paper adopts block size based data request approach [20]–[22] with direct data accessing mechanism to acquire reference data for motion compensation. In other words, the reference data is only fetched from external memory in necessary without any aids of cache buffers or complex addresses generators so that the hardware costs and implementation complexity can be reduced significantly. Simulation results show that about 62%–74% [20]–[22] of the data bandwidth can be reduced.

Fig. 14 shows the block diagram of proposed motion compensation architecture in two pipeline stages, data access and interpolation. The first stage consists of motion vector generation and reference pixel accessing modules to generate MVs of current MB and its data request to memory controller for accessing reference pixels from external memory. The returned reference pixels are collected in a 21 × 21-pixel register array, and then written to a reference data buffer for next MB use. The data rearrangement for the 21 × 21-pixel register array is briefly stated below. At the beginning, the reference data are requested from external memory according to the partition size. If the partition size is 16 × 16 for current MB, 21 × 21 pixels will be acquired from external memory and stored into the 21 × 21-pixel register array. However, for other partition size smaller than 16 × 16, we will fetch and store the reference data for one partition at a time. For example, if the partition size is 16 × 8, the 21 × 13 reference data of one block would be stored into the 21 × 21-pixel register array first. Once this block has finished the operations of motion compensation, the 21 × 21-pixel register array would be refreshed immediately and the new 21 × 13 reference data of another block would be fetched and stored. The same data rearranging strategy has also been applied for other partition sizes. The second stage consists of the interpolation module which is used to interpolate fractional pixels from reference data and then reconstruct pixels by adding the interpolated pixels and residuals together.

D. Fourth Stage Design

This stage includes the deblocking filter and texture padding modules. The deblocking filter module adopts the designs from [24] and [25] for high throughput requirement. The texture padding module will generate the un-available regions in the spatial base layer for texture upsampling used in the interlayer intra prediction of the enhancement layer. To reduce the padding complexity, we adopt a BL-level padding flow [14] in our design. In the BL-level padding flow, the padding procedure is moved from the enhancement layer to the base layer. In other words, the MB pre-padding is executed in the base layer to extend the reconstruction border and fill up the un-available inter coded regions so that no extension procedure will be executed in the enhancement layer and thus save the cycle time.

It is worth mentioning that the deblocking filter would not be applied in these MBs since the inter coded MBs in the spatial base layer would not be reconstructed to derive the pixel samples. Thus, pre-padding for un-available region instead of deblocking can be processed during the intercoded MBs decoding process. Therefore, we combine the padding and deblocking in the same stage in our design as shown in Fig. 15 so that the padding cycle can be hidden without additional penalty since the deblocking filter component is commonly designed as an independent pipeline stage in a video decoder. From simulation results, 26% of decoding time can be saved for interlayer intra coded MBs on average with the combined padding and deblocking flow.

IV. Implementation Results

The proposed architecture is implemented in Verilog HDL, and synthesized by Synopsys Design Compiler with UMC 90 nm 1P9M CMOS Technology Library in the worse case setting. The decoded videos of our decoder are verified to be the same as the decoded videos of reference software JSVM9.14 [26] with various test conditions (for example, the video sequences of *Bluesky, Pedestrain_area*, and *Tractor* with three spatial, temporal, quality layers as specified in Section I). Table V shows the detailed gate count of each component. The result shows that the total gate count of this paper is about 542 k for the target 135 MHz operating frequency; in which, the entropy decoder occupies the most of the area due to parallel designs and two types of decoding scheme.

Table VI lists the internal memory requirements among components and pipeline stages. In summary, the total internal memory requirement in this design is 39.66 Kbytes. The major buffer cost is due to buffering all required key picture data of two reconstructed quality layers and pipeline data for the parallel reconstruction.

The comparison of this paper and other state-of-the-art video decoders is listed in Table VII. Since only two SVC literatures [12], [13] have been published so far, H.264/AVC HD decoders [7], [8], and [11] are adopted here for comparison. Generally, the gate count costs of SVC decoder are larger than that of H.264/AVC decoder due to the additional scalabilities. It majorly comes from the interlayer prediction which introduces high arithmetic complexity and numerous external data buffers. Also, SVC applications have more external memory requirements as well. Compared to other SVC decoders, [13] can support two spatial layers with only one quality layer each, and four quality layers inside single spatial layer. However, our design can provide three spatial layers with three quality scalabilities in each spatial layer. In addition, this paper can provide superior Max throughput for multiple scalabilities in which the Max throughput represents the processing capability for combined spatial and quality scalability as defined in Table VII. To normalize the performance, the Gate efficiency is calculated by means of computing the max throughput per kilo gates. The results show that this paper has better performance in Gate efficiency when compared to other designs.

V. Conclusion

This paper presented a complete H.264/AVC scalable high profile decoder with optimizations from decoding flow analysis to module implementation with focus on the design of three scalabilities. For decoding flow, we proposed an one-pass quality decoding method to save 71% of memory bandwidth and at most 66% of MB processing time. For interlayer prediction, we proposed the combined decoding and padding flow, centralized accumulation-based CCSP concept, simplified poly-phase interpolator, and efficient motion vector upsampling to save the area cost and decoding time. Furthermore, the proposed parallel-pipeline architecture for the residual reconstruction path can achieve 54% of gate count savings compared to the traditional serial-pipeline architecture. Implementation results show that the proposed decoder can simultaneously decode 60 frames/s for CIF+SD480p+HD1080p resolution along with three quality layers under 135 MHz operating frequency with 541.52 k gate counts and 39.66 Kbytes internal memory consumption.

References

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.

[2] *Joint Draft 11 of SVC Amendment*, Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Oct. 2007.

[3] P.-Y. Hsu, G.-L. Li, and T.-S. Chang, "Memory analysis for H.264/AVC scalable extension decoder," in *Proc. APSIPA Annu. Summit Conf.*, Oct. 2009, pp. 299–302.

[4] N. D. Narvekar, B. Konnanath, S. Mehta, S. Chintalapati, I. Alkamal, C. Chakrabarti, and L. J. Karam, "An H.264/SVC memory architecture supporting spatial and course-grained quality scalabilities," in *Proc. IEEE Conf. Image Process.*, Nov. 2009, pp. 2661–2664.

[5] Y. H. Chen, T. D. Chuang, C. Y. Tsai, Y. J. Chen, and L. G. Chen, "A cost-efficient residual prediction VLSI architecture for H.264/AVC scalable extension," in *Proc. Picture Coding Symp.*, Nov. 2007.

[6] D. Zhou, Z. You, J. Zhu, J. Kong, Y. Hong, X. Chen, X. He, C. Xu, H. Zhang, J. Zhou, N. Deng, P. Liu, and S. Goto, "A 1080p@60fps multistandard video decoder chip designed for power and cost efficiency in a system perspective," in *Proc. Symp. VLSI Circuit*, Jun. 2009, pp. 262–263.

[7] T. M. Liu, T. A. Lin, S. Z. Wang, W. P. Lee, J. Y. Yang, K. C. Hou, and C. Y. Lee, "A 125 μW, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications," *IEEE J. Solid-State Circuits*, vol. 42, no. 1, pp. 161–169, Jan. 2007.

[8] C. C. Lin, J. I. Guo, H. C. Chang, Y. C. Yang, J. W. Chen, M. C. Tsai, and J. S. Wang, "A 160 kGate 4.5 kB SRAM H.264 video decoder for HDTV applications," in *Proc. IEEE Int. Solid-State Circuits Conf.*, Feb. 2006, pp. 406–407.

[9] C. D. Chien, C. A. Chien, J. C. Chu, J. I. Guo, and C. H. Cheng, "A 252 Kgates/4.9 Kbytes SRAM/71 mW multistandard video decoder for high definition video applications," *ACM Trans. Design Autom. Electron. Syst.*, vol. 14, no. 1, p. 17:17, Jan. 2009.
[10] V. Sze, D. F. Finchelstein, M. E. Sinangil, and A. P. Chandrakasan, "A 0.7 V 1.8-mW H.264/AVC 720p video decoder," *IEEE J. Solid-State Circuits*, vol. 4, no. 11, pp. 2943–2956, Nov. 2009.
[11] D. Zhou, J. Zhou, X. He, J. Zhu, J. Kong, P. Liu, and S. Goto, "A 530 Mpixels/s 4096×2160@60fps H.264/AVC high profile video decoder chip," *IEEE J. Solid-State Circuits*, vol. 46, no. 4, pp. 777–788, Apr. 2011.
[12] C. A. Chien, Y. C. Yang, H. C. Chang, J. W. Chen, C. Y. Chang, J. I. Guo, J. S. Wang, and C. W. Cheng, "A H.264/MPEG-2 dual mode video decoder chip supporting temporal/spatial scalable video," in *Proc. Asia South Pac. Des. Autom. Conf.*, Jan. 2011, pp. 73–74.
[13] T. D. Chuang, P. K. Tsung, P. C. Lin, L. M. Chang, T. C. Ma, Y. H. Chen, Y. H. Chen, C. Y. Tsai, and L. G. Chen, "A 59.5 mW scalable/multi-view video decoder chip for quad/3-D full HDTV and video streaming applications," in *Proc. IEEE Solid-State Circuits Conf.*, Feb. 2010, pp. 330–331.
[14] T. D. Chuang, P. K. Tsung, P. C. Lin, L. M. Chang, T. C. Ma, Y. H. Chen, and L. G. Chen, "Low bandwidth decoder framework for H.264/AVC scalable extension," in *Proc. IEEE Symp. Circuits Syst.*, May 2010, pp. 2960–2963.
[15] H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, "Combined scalability support for the scalable extension of H.264/AVC," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 446–449.
[16] Y. H. Liao, G. L. Li, and T. S. Chang, "A 385 MHz 13.54 K gates CAVLD decoder for level 5.1 H.264/AVC video," *IEEE Trans. Circuits Syst. Video Technol.*, to be published.
[17] Y. H. Liao, G. L. Li, and T. S. Chang, "A high throughput VLSI design with hybrid memory architecture for H.264/AVC CABAD decoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2010, pp. 2007–2010.
[18] C. A. Segall and G. J. Sullivan, "Spatial scalability within the H.264/AVC scalable video coding extension," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1121–1135, Sep. 2007.
[19] C. Y. Tsai, T. C. Chen, T. W. Chen, and L. G. Chen, "Bandwidth optimized motion compensation hardware design for H.264/AVC HDTV decoder," in *Proc. IEEE Int. Midwest Symp. Circuit Syst.*, vol. 2. Aug. 2005, pp. 1199–1202.
[20] T. D. Chuang, L. M. Chang, T. W. Chiu, Y. H. Chen, and L. G. Chen, "Bandwidth-efficient cache-based motion compensation architecture with DRAM-friendly data access control," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2009, pp. 2003–2013.
[21] P. Chao and Y. L. Lin, "A motion compensation system with a high efficiency reference frame pre-fetch scheme for QFHD H.264/AVC decoding," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 256–259.
[22] P. Chao and Y. L. Lin, "Reference frame access optimization for ultrahigh resolution H.264/AVC decoding," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jun. 2008, pp. 1441–1444.
[23] P. Chao and Y. L. Lin, "An elastic software cache with fast prefetching for motion compensation in video decoding," in *Proc. IEEE/ACM Int. Conf. Hardw.-Softw. Codesign Syst. Synthesis*, Oct. 2010, pp. 23–32.
[24] F. Tobajas, G. M. Callico, P. A. Perez, V. de Armas, and R. Sarmiento, "An efficient double-filter hardware architecture for H.264/AVC deblocking filtering," *IEEE Trans. Consumer Electron.*, vol. 54, no. 1, pp. 131–139, Feb. 2008.
[25] K. Xu and C.-S. Choy, "A five-stage pipeline, 204 cycles/MB, single-port SRAM-based deblocking filter for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 363–374, Mar. 2008.
[26] *JSVM Software Version JSVM 9.14*, ITU-T, I. JTC1.

Gwo-Long Li received the B.S. degree from the Department of Computer Science and Information Engineering, Shu-Te University, Kaohsiung, Taiwan, in 2004, the M.S. degree from the Department of Electrical Engineering, National Dong-Hwa University, Hualien, Taiwan, in 2006, and the Ph.D. degree from the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, in 2011.

He is currently an Engineer with the Industrial Technology Research Institute, Hsinchu. His current research interests include the video signal processing and its very large scale integration architecture design.

Dr. Li received the Excellent Master Thesis Award from the Institute of Information and Computer Machinery in 2006.

Yu-Chen Chen received the M.S. degree from the Department of Electronics Engineering, National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 2010.

In 2008, he joined the VLSI Signal Processing Laboratory, NCTU. His current research interests include signal processing, scalable video coding, as well as very large scale integration architecture design of video decoder.

Yuan-Hsin Liao received the B.S. and M.S. degrees in electronics engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 2008 and 2010, respectively.

In 2010, he joined PixArt Imagine, Inc., Hsinchu. His current research interests include video processing, computer vision, IP, and system-on-chip design.

Po-Yuan Hsu received the B.S. and M.S. degrees from the Department of Electronics Engineering, National Chiao-Tung University, Hsinchu, Taiwan, in 2007 and 2009, respectively.

His current research interests include digital signal processing, scalable video coding, and associated very large scale integration architectures.

Meng-Hsun Wen received the B.S. and M.S. degrees in electrical engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 2009 and 2011, respectively.

After graduation, he joined PixArt Imagine, Inc., Hsinchu. His major research interests include H.264/AVC video coding and associated very large scale integration architecture design.

Tian-Sheuan Chang (S'93–M'06–SM'07) received the B.S., M.S., and Ph.D. degrees in electronic engineering from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, in 1993, 1995, and 1999, respectively.

He is currently an Associate Professor with the Department of Electronics Engineering, NCTU. From 2000 to 2004, he was a Deputy Manager with Global Unichip Corporation, Hsinchu. His current research interests include (silicon) intellectual property and system-on-a-chip design, very large scale integration signal processing, and computer architecture.