

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※

※ 應用於數位電視的雙模視訊解碼器 ※

※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：個別型計畫     整合型計畫

計畫編號：NSC97-2221-E-009-167

執行期間： 98 年 8 月 1 日 至 99 年 7 月 31 日

計畫主持人：李 鎮 宜 教授

共同主持人：

計畫參與人員：李曜 李韋磬 吳昱德 賴昱帆 楊均宸 林建辰 林

佳龍 李欣儒 許智翔

執行單位：國立交通大學 電子工程學系

中 華 民 國     97 年   6 月   30 日

應用於行動通訊之下世代低功耗視訊解碼器(2/3)

A Next Generation Low Power Consumption Video Decoder for Mobile Video

Application (2/3)

計畫編號: NSC97-2221-E-009-167

執行期間: 98 年 8 月 1 日 至 99 年 7 月 31 日

主持人: 李鎮宜 交通大學電子工程系教授

# INDEX

---

英文摘要.....	7
一、 計畫的緣由與自的.....	9
A. The Reduced Patterns Comparison Embedded Compressor/Decompressor (RPCC).....	9
B. The Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor .....	11
二、 研究方法及成果.....	12
A. The Reduced Patterns Comparison Embedded Compressor/Decompressor.....	12
(1) The Proposed RPCC Embedded Compression Algorithm.....	12
(2) Proposed Architecture .....	17
(3) System Integration And Verification.....	19
(4) Simulation Result.....	22
B. The Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor .....	23
(1) Proposed BTPCC Embedded Compression Algorithm .....	23
(2) Proposed BTPCC Embedded Compression Architecture .....	27
(3) System Integration .....	28
(4) Experimental Results .....	30
三、 結論與討論.....	31
四、 參考文獻.....	32
五、 計畫成果自評.....	35
(1) The Reduced Patterns Comparison Embedded Compressor/Decompressor .....	35
(2) The Bitplane Truncation with Pattern Comparison Coding Emebedded Compressor/Decompressor .....	35

## Figure Index

---

Figure 1 Compression methods of our proposed algorithm .....	13
Figure 2 Reduced patterns comparison coding concept .....	14
Figure 3 The compressed 32-bit data format .....	14
Figure 4 Four predefined bit planes .....	15
Figure 5 Distribution of PSNR loss in 4x1-based PCC algorithm .....	16
Figure 6 Hardware design for the MBPTC .....	17
Figure 7 The hardware architecture of RPCC .....	18
Figure 8 Data flow of the encoder .....	18
Figure 9 Data flow of the decoder .....	19
Figure 10 The architecture of our proposed H.264 decoder with EC capability .....	20
Figure 11 Block Diagram in CoWare simulation platform .....	21
Figure 12 Compression flow of the proposed algorithm .....	24
Figure 13 Compressed 32-bit segment format .....	24
Figure 14 Flowchart of the pixel truncation .....	25
Figure 15 Flowchart of the selective bitplane .....	25
Figure 16 Flowchart of the rounding .....	26
Figure 17 An example of partitioning 4x2 block .....	27
Figure 18 Compressor architecture .....	27
Figure 19 Decompressor architecture .....	28
Figure 20 H.264 decoder with proposed embedded compression .....	29
Figure 21 Simulation result of station sequence (HDTV) .....	31

## Table Index

---

Table 1 Ratio of error rate per position in each 4x1 bit plane.....	16
Table 2 Weights under different coding thresholds.....	16
Table 3 All cases of read access required by MC with/without EC.....	22
Table 4 Comparison of Simulation .....	23
Table 5 Three Groups of Eight Patterns .....	27
Table 6 All Cases of Read Access Requirement .....	30
Table 7 PSNR Comparison .....	30
Table 8 Comparison Among Previous Work.....	31

## 中文摘要

新一代的視訊解碼系統，除了必須滿足多標準和多模式的操作模式外，更重要的是如何降低功耗，以及隨著電源能量的多寡，提供行動視訊的最佳終端服務需求。在此三年（2008/8~2011/7）的研究計畫中，我們將延續過去三年（2005/8~2008/7）在視訊處理器的研究成果，朝低功耗、低成本、以及多模式的視訊解碼方案進行多項關鍵技術的研究。在多模式的研究議題上，主要將 H.264/SVC 的功能需求加入現有的雙模式（MPEG2 和 H.264）硬體平台上，探討新的關鍵模組的實現方案，以及從系統整體行為模式和效能的考量下，提出一更好的系統硬體架構，有助於單獨系統的效能展現和以 IP 為次系統的整合效益。在低成本的研究議題上，主要考量到如何降低解碼過程所需求的記憶容量，並採用外掛的記憶體模組，尤其當動態補償所需求的高記憶體容量和頻寬時，如何充分使用有限的資源（記憶容量和頻寬），達成符合標準解碼的運算需求。在低功耗的設計議題上，我們除了分析解碼行為的特徵和架構的相依性，藉以探討系統、個別模組、資料流等不同層級的低功耗設計方法，亦將奈米級製程所衍生的漏電流效應，一併考量，提供符合視訊解碼標準下的低功耗設計方案。此外，我們亦將建立 FPGA 的雛形系統展示平台，有利於關鍵模組和系統行為的呈現。

關鍵字：

視訊解碼系統；多模式；多標準；低功耗；低成本；行動視訊

## 英文摘要

It is well understood that research efforts, for next-generation video decoding system, have to cover not only multi-standard and multi-mode operation capability, but also less power dissipation and power awareness with optimal picture quality, especially when mobile video services are taken into account. As a result, in this 3-year (2008/8~2011/7) research project proposal, we'll further investigate several key issues related to so-called low-power, low-cost, and multi-mode video decoder solutions. Based on our previous work on a dual-mode video (2005/8~2008/7), we'll leverage the available design platform and research results to further explore new design approaches. For multi-mode task, we'll investigate the specifications defined in H.264/SVC and add those key modules into our H.264/MPEG2 decoder platform. Not only new key modules will be explored, but also system decoding behavior will be analyzed to study a better system architectural model so that a stand-alone and IP-based decoder solution can be obtained. For low-cost issue, the major problem lies in memory management and limited bus bandwidth. It is necessary to take into account available stand-alone memory modules; even SoC solutions become a must. Therefore developing a well-organized memory hierarchy and access mechanism to meet decoding requirements under limited resources (storage space and bus bandwidth) will be further explored. For low-power issue, an analysis of the decoding behavior and related hardware architecture will be conducted. Thus system exploration, module design, and data flow will be investigated to reduce power dissipation at different levels. In addition, leakage current due to nano-meter CMOS process will also be considered to provide a competitive video decoder solution. Finally an FPGA prototype will be set up to evaluate the performance of the proposed video decoder and related key modules.

Keywords:

Video Decoder, Multi-Mode, Multi-Standard, Low-Power, Low-Cost, Mobile Video



## 一、計畫的緣由與目的

### A. *The Reduced Patterns Comparison Embedded Compressor/Decompressor (RPCC)*

To improve the video coding efficiency, diminishing the data correlation of the temporal redundancy in each frame is widely used in the latest video coding standard, such as H.264/AVC [1]-[2]. However, it causes a large amount of data transmission between on-chip processing modules and external memory. In addition, the rapid and huge data access from Motion Compensation (MC) consumes the majority of system power and become serious in many portable devices. Many low power techniques have already been proposed to reduce power consumption, but data transmission still dominates huge amount of system power. Hence, reduce data access between on-chip processing modules and external memory is the critical consideration in a mobile video device. Although the mobile video devices are suffered from limited battery capability, the visual quality requirement is not as high as high resolution applications. Therefore, the embedded compression is suitable to lessen the volume of data access and the size of off-chip memory under the premise of maintaining acceptable visual quality. The mobile video devices are more and more important due to their various functions at the present time. Reducing the usage of bandwidth and the required resource of hardware in the mobile video devices is a critical topic.

In general, the compression methods are classified into two categories: lossless compression and lossy compression. It is obvious that lossless compression methods [3] completely reserve the information while truncating the size of data, so there has no quality loss. However, some problems of lossless compression are so fatal that it's not suitable for system integration application. The lossless compression suffers from variable length of lossless compressed data that we cannot regularly control the compression ratio, frame memory size and bandwidth requirement. These

disadvantages are also attributed to the needs of memory to prepare for the worst case of data access and the unknown size of data. Therefore, there exists an important characteristic of lossy compression methods [4]-[7] which differs them from lossless compression methods. The characteristic of fixed compression ratio allows us to improve the disadvantages of lossless compression methods mentioned previously. Although lossy compression algorithm will sacrifice tolerable visual quality, the reduced power consumption, memory size and bandwidth requirement are more attractive for mobile video devices.

Several lossy compression schemes have been proposed in [4]-[7]. The transform-based compression methods can convert the signal from time domain to frequency domain and move the energy to up-left corner. In human visual system, the lower frequency component is more important than the higher frequency component whose feature can be exploited to efficiently compress the amount of data, such as in [4]-[5]. In [4], both Modified Hadamard Transform (MHT) and quantization of Golomb-Rice Coding (GRC) are employed. To improve the quality loss of [4], [5] adopts Discrete Cosine Transform (DCT) and Modified Bit Plane Zonal Coding (MBPZC) instead of MHT and GRC. Although transform-based schemes provide good compressed quality, MHT and DCT are too complicated to suit for being embedded with H.264 mobile video devices. Another kind of algorithms is pattern-based [6]-[7]. [6] adopts 64 patterns to improve Bit Plane Truncation (BTC) algorithm and [7] increases extra acceptable quality loss to reduce the number of compared patterns from [6]. Both [6] and [7] are limited by BTC algorithm; the coding latency is still too long to be well- embedded into the target H.264 system. However, through [6] and [7], we find a way to utilize the patterns to reduce the coding latency and the amount of data.

In this paper, we propose a pattern-based lossy embedded compression method

which adopts 4x2 pixels as coding unit and CR is fixed as two.

### *B. The Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor*

A video coding standard achieves high compression efficiency such as H.264 [7] and so forth. For H.264, at least one previous frame is stored in frame memory to generate a predicted frame. Obviously, Motion Compensation (MC) demands a huge amount of data accesses between off-chip memory devices and the video decoder chip. However, data transfer consumes a lot of power. For mobile video devices, one major issue is the limited power supply from battery. Therefore, reducing the bandwidth requirement and size of frame memory is greatly demanded while maintaining acceptable visual quality.

In general, embedded compression methods can be categorized into two fundamental groups: lossless and lossy. Lossless compression algorithms [9] have no error propagation problem. Lossy compression algorithms, comparing with lossless compression algorithms, accomplish the fixed compression ratio (CR). Several lossy compression algorithms have been proposed such as Modified Hadamard Transform (MHT) plus quantization of Colomb-Rice Coding [4], DCT plus Modified Bit Plane Zonal Coding [5], and et al. [6] exploits forty-six patterns to improve Block Truncation Coding and [7] increases extra acceptable quality loss to reduce the number of compared patterns from [6].

Lossless compression can guarantee no quality loss, but variable length of the compressed data caused irreducible frame memory size. Therefore, existing lossless algorithms are not suitable for frame compression because their primary purpose is high coding efficiency rather than low latency, computation complexity, and high random accessibility. On the contrary, lossy compression algorithm with the fixed CR

can guarantee the reduction of frame memory size. Consequently, it is important to design a lossy algorithm with the following features: 1) Low distortion visual quality, 2) Low complexity, 3) Low bandwidth requirement, and 4) Low power consumption.

## 二、 研究方法及成果

### A. *The Reduced Patterns Comparison Embedded Compressor/Decompressor*

#### (1) *The Proposed RPCC Embedded Compression Algorithm*

##### (i) *Algorithm*

The proposed compression scheme adopts pattern-based and 4x2 block-grid. The CR is fixed as two and each 4x2 unit (64 bits) will be compressed into 32-bit data package. Because fixed CR results in regular amount of coded data, the EC assures the ability of random access without extra memory to register the segment address of coded data. In addition, the 4x4 block unit is the basic coding unit in H.264 standard, we partition each 4x4 block into two 4x2 blocks. Thus, 4x2-based block-grid lessens the coding latency and makes the data access more efficient.

The proposed algorithm is shown in Figure 1. There are three parts in the overall EC method: 1) MBPTC, 2) RPCC and 3) average coding. In MBPTC algorithm, we partition a 4x2 matrix into eight bit planes and search the Start Plane (SP) in four continuous layers which are close to MSB with 2 bits as the first compression step

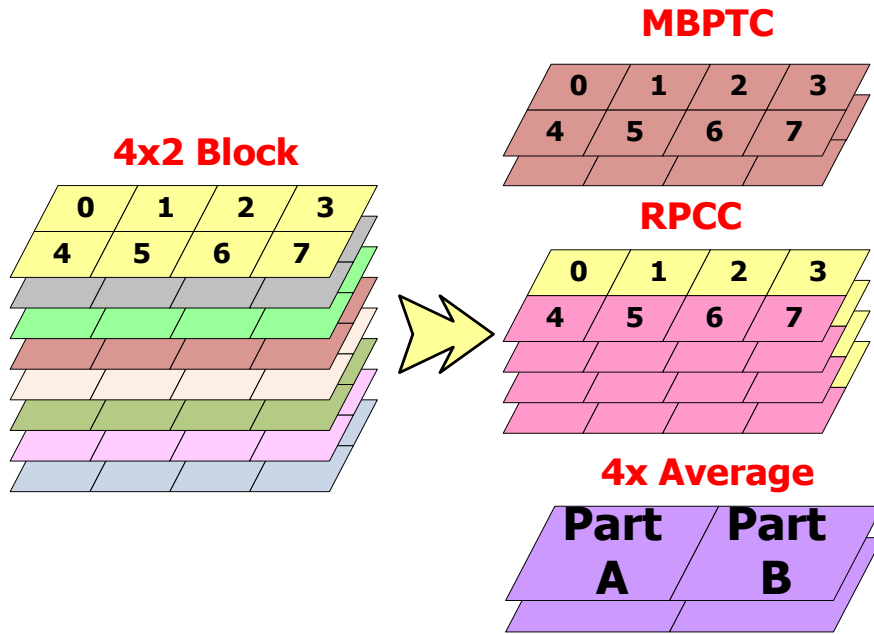


Figure 1 Compression methods of our proposed algorithm

The second step is to deal with remaining bit planes after MBPTC by RPCC. As shown in Figure 2, we partition a 4x1 section with 4 bits into four 4x1 layers. According to the coding threshold adopted, the RPCC will select the left or right strategy. While we set the coding threshold to level 2, RPCC compares layer 1 and layer 2 with eight 4x1-based patterns at the same time. If there is no error in layer 1 and layer 2, RPCC adopts the left strategy to compress the 4x1 section. Otherwise, RPCC adopts right strategy. According to the simulation result with different thresholds, while the right strategy is adopted, the right strategy is often in worse case. We exploit this feature to improve the drawback in 4x1-based PCC algorithm.

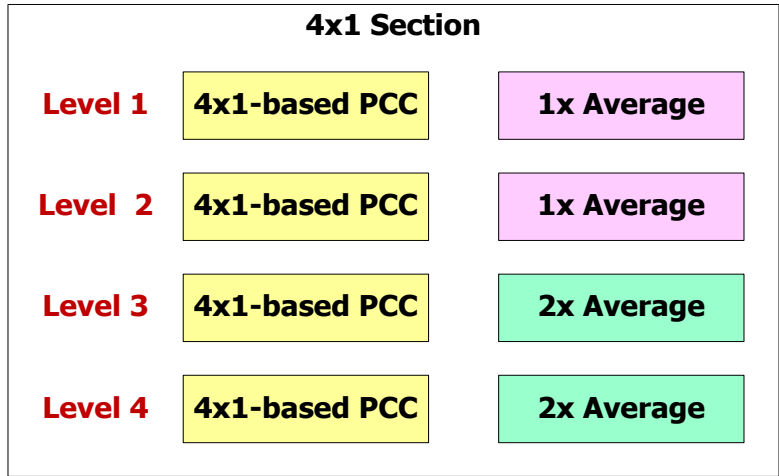


Figure 2 Reduced patterns comparison coding concept

The third part is the average coding scheme which deals with the two residual continuous bit planes after RPCC. We partition these bit planes into two 2x2-based parts and calculate the average value in each 2x2-based part. The coded data format is shown in Figure 3.

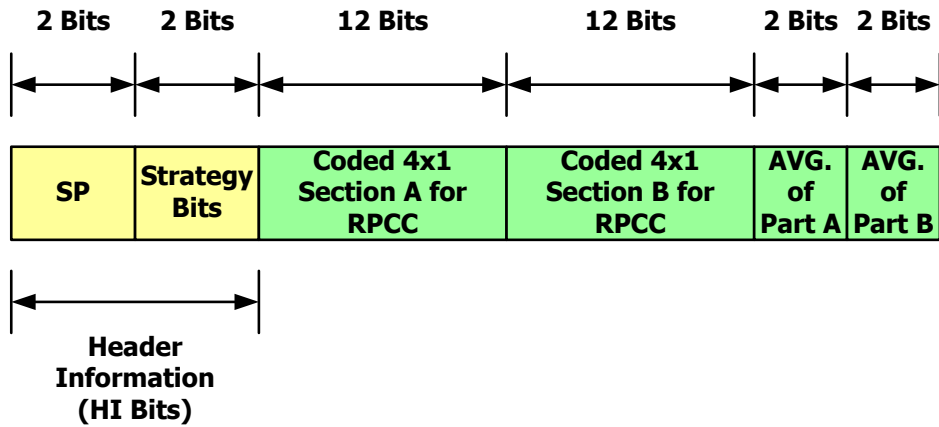


Figure 3 The compressed 32-bit data format

(ii) *Design of Patterns*

For a 4x2 block, the bit plane consists of 8 bits, leading to 28 (= 256) possible number of bit planes. However, most of bit planes do not often appear in an image

and contribute the less visual quality of decoded image. In addition, some different bit planes can provide the proximate visual quality. Thus, we focus on the design of a small set of visually sensitive predefined bit planes as shown in Figure 4. By inverting the polarization (0s and 1s) of predefined bit planes, eight patterns representing edges and lines are generated. Each pattern is represented by a 3-bit index as the number of patterns is eight.

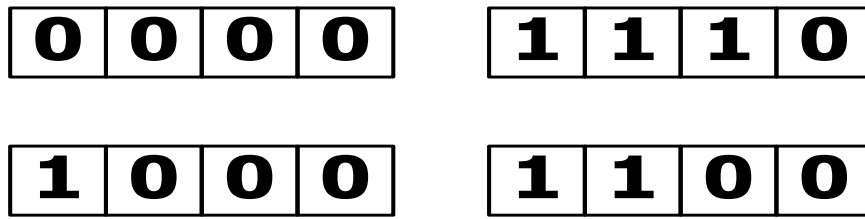


Figure 4 Four predefined bit planes

(iii) *Formula*

We derive the formula (1) from the simulation result. It is about the PSNR loss of 4x1-based PCC algorithm.  $i$  is the number of 4x1 error bit plane.  $P_m$  is the error rate and  $P_n$  is the ratio of error rate per position in each 4x1 bit plane as described in Table 1. As described in the previous section, we can setup the different coding thresholds (Level 0~4) in 4x1-based PCC algorithm to obtain corresponding weight ( $W_i$ ) as described in Table 2. We can exploit the formula to estimate for the PSNR loss in 4x1-based PCC algorithm while the previous parameters are modified.

$$\text{PSNR Loss (4x1 - based)} = \sum_{i=0}^4 \left[ C_i^4 (P_m \cdot P_n) \cdot (1 - P_m \cdot P_n)^{4-i} \right] \cdot W_i \quad (1)$$

Table 1 Ratio of error rate per position in each 4x1 bit plane

$W_i$	Level 0	Level 1	Level 2	Level 3	Level 4
$W_0$	0	0	0	0	0
$W_1$	240	112	48	16	0
$W_2$	720	224	48	0	0
$W_3$	720	112	0	0	0
$W_4$	240	0	0	0	0

Table 2 Weights under different coding thresholds

$P_n$	Error Rate (%)	Total Ratio (%)
$P_0$	8.68	32.54
$P_1$	4.65	17.43
$P_2$	4.65	17.43
$P_3$	8.70	32.60

Figure 5 shows the distribution of PSNR loss in all thresholds. It helps improving the coding performance of the algorithm.

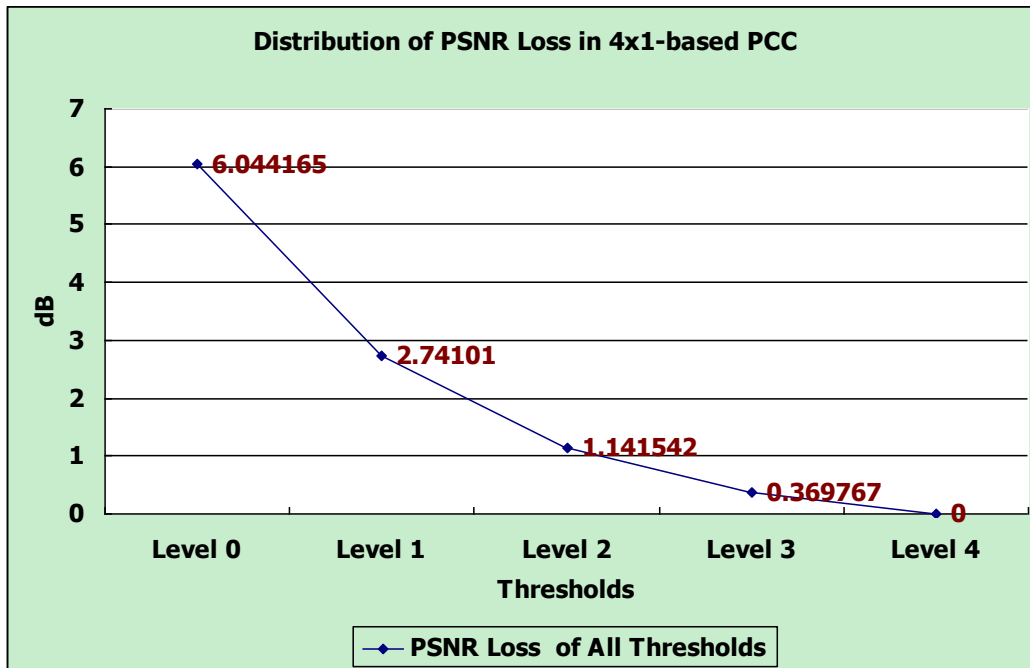


Figure 5 Distribution of PSNR loss in 4x1-based PCC algorithm



(2) *Proposed Architecture*

(i) *Modified Bit Plane Truncation Coding*

The hardware design of MBPTC is improved from original BPTC. It is a combinational block to deal with 4x2 pixels to obtain Start Plane (SP) and 4x2-plane component for each 4x2 array. In Figure 6, we employ three 8-input OR gates as thresholds to control the value of SP. The bits of layer 1, 2 and 3 are used to be input of 8-input OR gate individually.

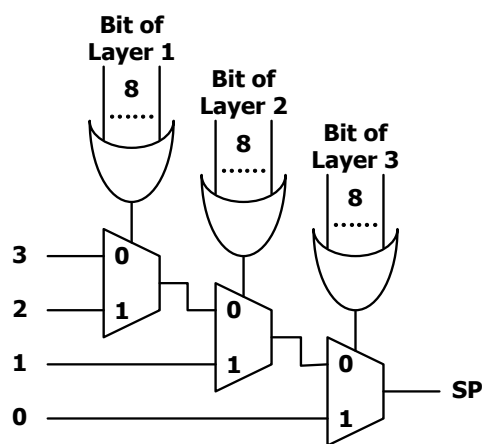


Figure 6 Hardware design for the MBPTC

(ii) *Reduced Patterns Comparison Coding*

RCP is a combinational block to deal with coded data by MBPTC. As shown in Figure 7, SP selects four layers to be compressed and threshold is exploited to choose the strategy to be adopted. The SP is produced by MBPTC and the threshold is defined by users with different levels as described in Table 1. (Here we adopt Level 2)

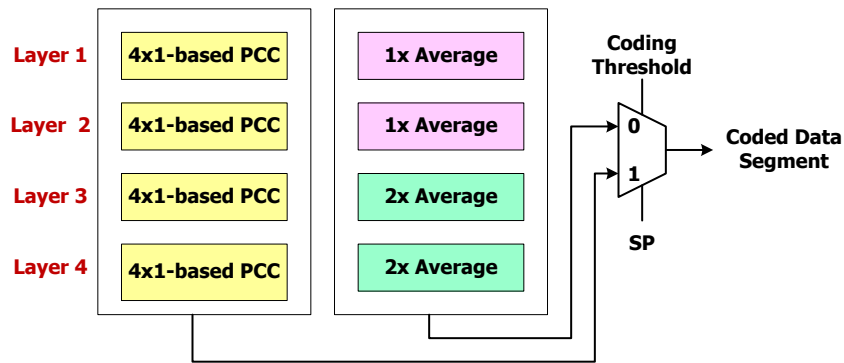


Figure 7 The hardware architecture of RPCC

(iii) *Data Rearrange*

Data unpacking is a simple reverse process of encoding. The decoder focuses on putting the data on proper positions. According to the coded data format, the SP selects the initial bit plane of decoding. The continuous four layers are then placed on corresponding positions depending on strategy bits. Afterward the average of part A and B is placed on the continuous two bit planes after the four layers.

(iv) *Overall Design of Encoder and Decoder*

The overall compressor design is shown in Figure 8. It takes one cycle to deal with 4x2 block. Here each MB takes 16 cycles to be encoded.

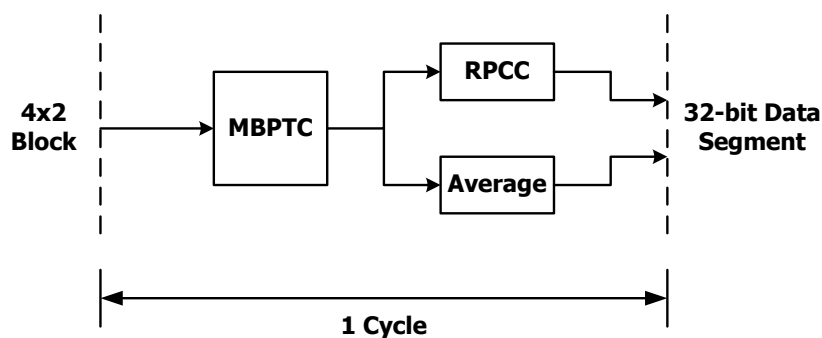


Figure 8 Data flow of the encoder

For providing data to MC, the decompressor needs to support higher throughput.

The actual architecture of decompressor design is shown in Figure 9. A 4x2 block takes one cycle to be decoded. Under the design, each MB takes 16 cycles to be decoded.

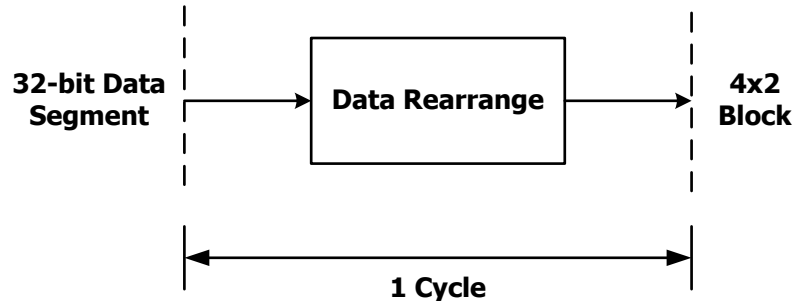


Figure 9 Data flow of the decoder

### (3) *System Integration And Verification*

Figure 10 shows the overall block diagram of this system. The adopted H.264 decoder works at 5, 100 and 150 MHz respectively to perform CIF, HD 1080 AVC, HD 1080/720 SVC at 30 frames/per second (FPS). The embedded compressor compresses the data from deblocking filter into 64-bit data segment which is stored in external memory. The embedded decompressor decompresses the coded data segment from off-chip memory into 4x2-sized block which is sent to Motion Compensation. The bandwidth of system bus is 32 bits and the external memory is 32 bits per entry.

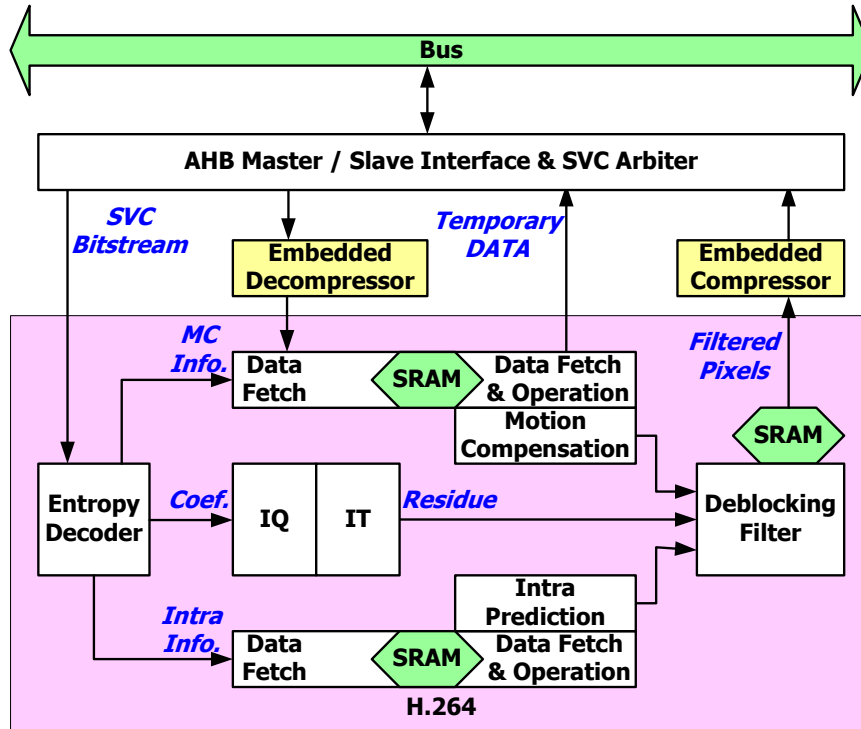


Figure 10 The architecture of our proposed H.264 decoder with EC capability

The related accesses of EC are partitioned into write accesses and read accesses. Write accesses from deblocking filter write the data to external memory and read accesses read the data from external memory to MC. Many methods have been proposed to improve embedded compression and all of them aim to improve the performance of embedded compression. However most of performance measured by these methods is fragmental, lacking verification from system level. In addition, we expect to precisely estimate the amount of read/write accesses on system view point. Thus, we employ “CoWare” to deal with the complicated problems. As shown in Figure 11, “CoWare” provides many functions to simulate a complete system and the user-defined means user’s design. It makes more efficient that we can change the user-defined field relied on our demands. We add the proposed design and H.264 system into user-defined field. The AMBA interface between “CoWare” and user-defined is coded in System C and it provides a protocol to commutate each other.

Furthermore, user-defined means all designs in this file need to be coded in Verilog.

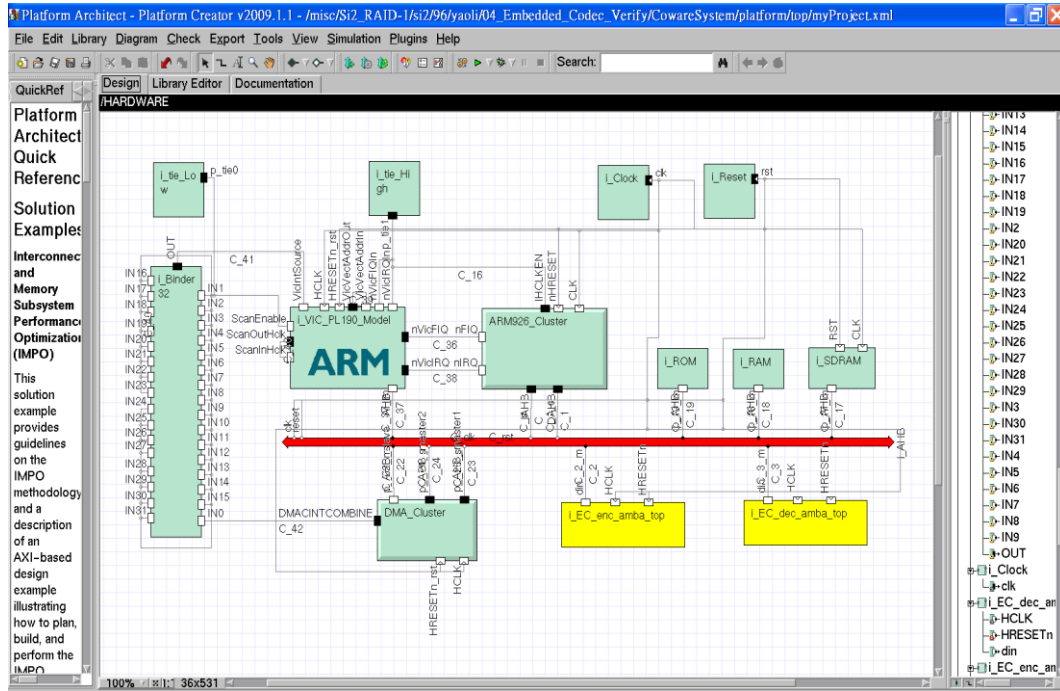


Figure 11 Block Diagram in CoWare simulation platform

Because the proposed algorithm provides fixed CR as two, the write access times after adding the EC are always half of original system. The reduction ratio of write access is 50%. The embedded decompressor decompresses the data from external memory to MC. Because the bandwidth of system bus is 32 bits and the external memory is 32 bits per entry, the original system takes 4x1 pixels as access unit. The read access behavior of MC with/without EC is analyzed as Table 2. The worst condition is the sub-pixel case. The 4x4 block needs a 9x9 block to complete the motion compensation. Thus, while original system needs 27 cycles to deal with this case, embedded compressor takes 15 cycles to do that. If the required 4x4 blocks of MC are aligned with the coded 4x4 blocks, original system with/without embedded compressor needs 2/4 cycles to deal with the best case. The special cases are included to (Align, Not Align), (Not Align, Not Align) and (Sub, Not Align). If required data of MC is not fit for 4x2 block-grids, it may increase extra access.

Table 3 All cases of read access required by MC with/without EC

Case of MV (x , y)	Access Cycles for System Without EC	Access Cycles for System With proposed EC	Reduction of Access Cycles Without EC (%)
( Align , Align )	4	2	50
( Align , Not Align )	4	2 / 3	50 / 25
( Align , Sub )	9	5	44.4
( Not Align , Align )	8	4	50
( Not Align , Not Align )	8	4 / 6	50 / 25
( Not Align , Sub )	18	10	44.4
( Sub , Align )	12	6	50
( Sub , Not Align )	12	6 / 9	50 / 25
( Sub , Sub )	27	15	44.4
<b>AVG.</b>	13.2	6.8 ~ 6.9	49.1 ~ 48.3

#### (4) Simulation Result

Software implementation of the proposed algorithm is integrated with JM 16.1. The reference frames are compressed by the proposed algorithm and then compared with those results from [3] and [4] respectively. The test sequences are akiyo, flower, football, foreman, mobile calendar, carphone, canoa, coastguard, waterfall and tempete in CIF format. For each sequence, computing the average PSNR value refers to the original sequence with 100 frames.

Table 4 shows the comparison results. It can be found that our proposed solution based on non-transform compression scheme provides lower complexity and less gate count with acceptable PSNR loss and visual quality.

Table 4 Comparison of Simulation

	MHT + GRC [4]	DCT + MBPZC [5]	Proposed	
<b>Technology</b>	UMC 90 nm	UMC 90 nm	UMC 90 nm	
<b>System</b>	MPEG-2 Decoder	H.264 Decoder	H.264 / SVC	
<b>Processing Data Unit</b>	8x1 Array ( 8 Pixels )	4x4 Array ( 16 Pixels )	4x2 Array ( 8 Pixels )	
<b>Working Frequency</b>	100 MHz	100 MHz	100 MHz	
<b>Total Gate Count</b>	20 K	30K	3.1 K	
<b>Cycle Count (Cycle)</b>	<i>Encoder</i>	2 for first 1x8, 1 for Pipeline Stage	12 for first 4x4, 4 for Pipeline Stage	2 for each 4x4
	<i>Decoder</i>	2 for first 1x8, 1 for Pipeline Stage	4 for first 4x4, 2 for Pipeline Stage	2 for each 4x4
<b>For a MB (Encoder/Decoder)</b>	33 Cycles/33 Cycles	72 Cycles/34 Cycles	33 Cycles/32 Cycles	
<b>PSNR Loss</b>	11.81 dB~14.57 dB	3.22 dB~8.39 dB	4.42 dB~7.10 dB ( Average: 5.98 )	
<b>Power Consumption</b>	N/A	2.78 mW / 1.66 mW	228 uW / 130 uW	

*B. The Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor*

*(1) Proposed BTPCC Embedded Compression Algorithm*

*(i) Algorithm*

The proposed algorithm compresses a 4x2 block (64-bit) from the output of the deblocking filter. The CR is fixed at 2. After compressing, a 4x2 block will become a 32-bit segment. With fixed CR, the amount of the coded data is constant. Therefore, this compression can guarantee access times. Besides, in H.264 standard, a 4x4 block which is a basic coding unit can be partitioned into two 4x2 blocks.

Figure 12 shows the flowchart of the proposed compression algorithm. We divide the algorithm into four parts: 1) Pixel Truncation, 2) Selective Bitplane, 3) Rounding, and 4) Pattern Comparison. These parts will be described in the following paragraphs. The compressed 32-bit segment format is shown in Figure 13. The representation format consists of 2-bit Mode, 2-bit Start Plane (SP), 2-bit Decision L, 2-bit Decision

R, 12-bit Coded Data L, and 12-bit Coded Data R.

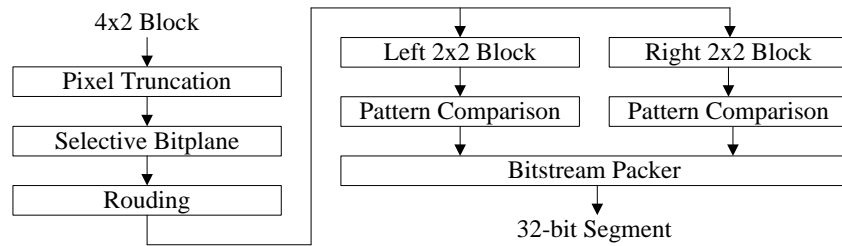


Figure 12 Compression flow of the proposed algorithm

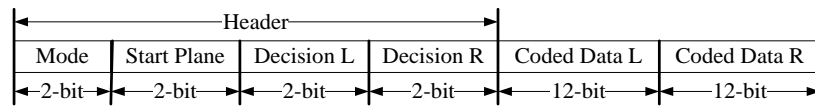


Figure 13 Compressed 32-bit segment format

#### A. Pixel Truncation

Figure 14 shows the flowchart of the pixel truncation. First, we calculate the average value (Avg.) of the 4x2 block and the difference value (Diff.) between maximum pixel and minimum pixel of the 4x2 block. Second, according to the average and the difference, we classify those 4x2 sub-blocks into five types as the following: 1) Avg. from 0 to 63 and Diff. less than 32, 2) Avg. from 64 to 127 and Diff. less than 64, 3) Avg. from 128 to 191 and Diff. less than 64, 4) Avg. from 192 to 255 and Diff. less than 32, and 5) no change. In type 1, if each pixel is larger than or equal to 64, we force the pixel to be 63. In type 2, if each pixel is less than 64, we force the pixel to be 64; if each pixel is larger than or equal to 128, we force the pixel to be 127. Types 3 and 4 are processed like types 2 and 1 respectively. In type 5, the original pixel value remains unchanged.



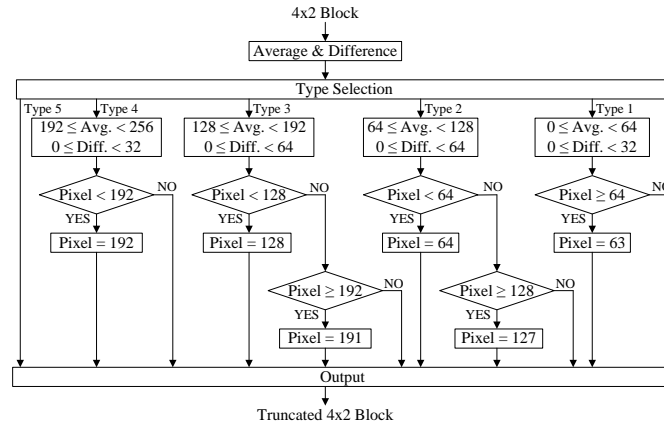


Figure 14 Flowchart of the pixel truncation

### B. Selective Bitplane

Figure 15 shows the flowchart of the selective bitplane. Bitplane coding is a well-known method. We exploit bitplane as a basic unit to a group numbers, instead of pixel-wised basic unit. First, we consider a 4x2 block in which each pixel value is represented by 8-bit. A bitplane can be formed by selecting a single bit from the same position in the binary representation of each pixel.

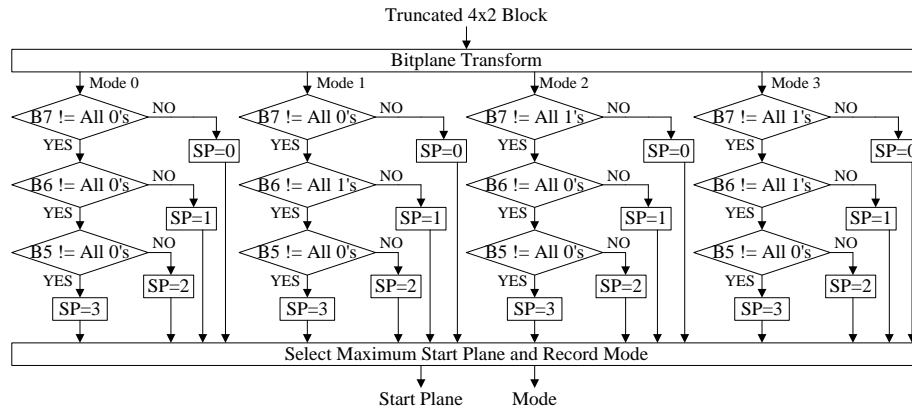


Figure 15 Flowchart of the selective bitplane

We define that B7 represents the MSB plane while B0 represents the LSB plane. Second, the start plane (SP) is searched for four successive bitplanes from the MSB bitplane with four modes as follows: 1) from B7 to B5 are all-0, 2) B6 is all-1; B7 and B5 are all-0, 3) B7 are all-1; B6 and B5 are all-0, and 4) B7 and B6 are all-1; B5 is

all-0. In the first mode, if both B7 and B6 are all-0 and B5 is not all-0, then SP is equal to 1. Similarly, the other modes like as the first mode. Finally, the maximum start plane of four modes is selected to record the mode and start plane.

### C. Rounding

Since lower bitplanes are truncated due to the limited budget, a simple rounding is applied here. The rounding is applied when the significant bit of the truncated bits is nonzero and the coded bits are not all 1's. In Figure 16(a), the simple idea is shown. This idea leads to a satisfied quality improvement. Two rounding modes are proposed because the pattern comparison has two data compressed formats. As shown in Figure 16(b), the first one is the compressed code rounding and the other is the uncompressed rounding. For pattern comparison, the first rounding method is applied to the first three types and the second rounding method is only for the final type.

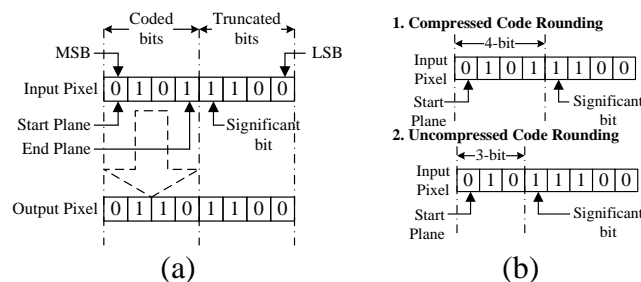


Figure 16 Flowchart of the rounding

### D. Pattern Comparison

The final step encodes the preserving bitplanes. First, the truncated 4x2 block is partitioned into two 2x2 blocks that are called the left 2x2 block and the right 2x2 block as shown in Figure 17(a). In Figure 17(b), both the left 2x2 block and the right 2x2 block exploited the equal SP and compressed individually. Second, four types for a 2x2 block is classified as follows: 1) Group A, 2) Group B, 3) Group C, and 4) Uncompression. The first three types exploit a group of the eight patterns to

compare with four successive bitplanes from SP and select one type which can hit three successive bitplanes. The three groups of the eight patterns are shown in Table I. If the first three types cannot hit larger than or equal to three bitplanes, the type 4 is chosen and three successive bitplanes from SP are stored.

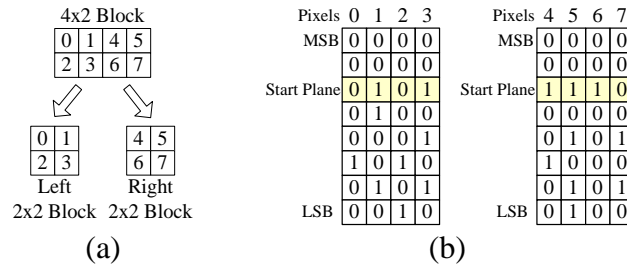


Figure 17 An example of partitioning 4x2 block

Table 5 Three Groups of Eight Patterns

Pattern No.	1	2	3	4	5	6	7	8
Group A	0000	1111	1110	0111	0011	1100	0001	1000
Group B	0000	1111	1110	0111	1010	1001	0110	0101
Group C	0000	1111	1110	0111	1101	1011	0010	0100

(2) Proposed BTPCC Embedded Compression Architecture

(i) Compressor Design

Figure 18 shows the pipeline architecture of compressor design. We use two pipeline stages and each stage requires one cycle. The first stage is the pixel truncation. The second stage is composed of selective start plane, rounding, selective pattern comparison, and packer. This compressor encodes a 4x2 block in 2 cycles.

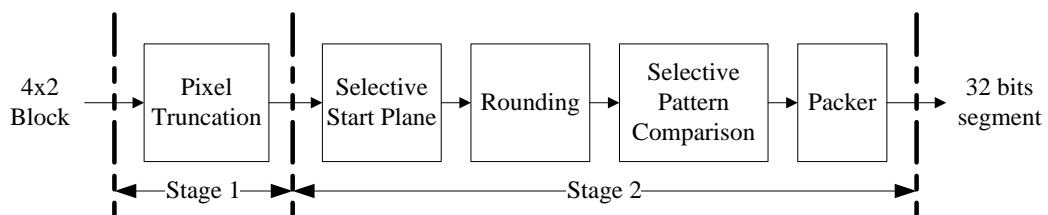


Figure 18 Compressor architecture

(ii) *Decompressor Design*

Figure 19 shows the pipeline architecture of decompressor. The decompressor only needs one stage with one cycle, including parser, start plane decoding, and pattern decoding. This decompressor reaches a higher throughput; therefore we can provide a higher random accessibility.

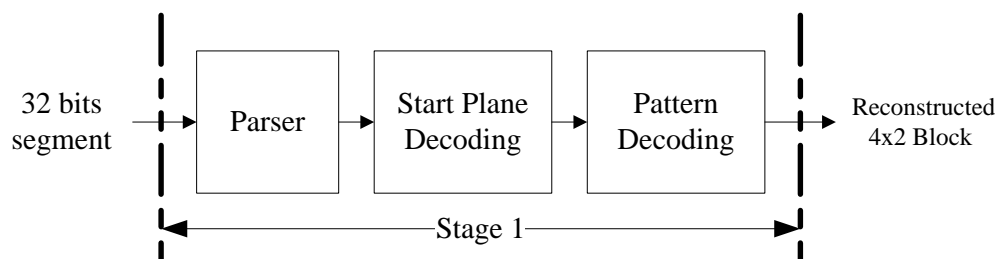


Figure 19 Decompressor architecture

(3) *System Integration*

The overall H.264 decoder [10] with the embedded compression codec is shown in Figure 20. The embedded compressor works between the deblocking filter and the external memory. The embedded decompressor works between the external memory and the motion compensation. To design address controller of EC is very simple since our compression ratio is fixed at two. Our system bus is 32 bits and the external memory is 32 bits per entry.

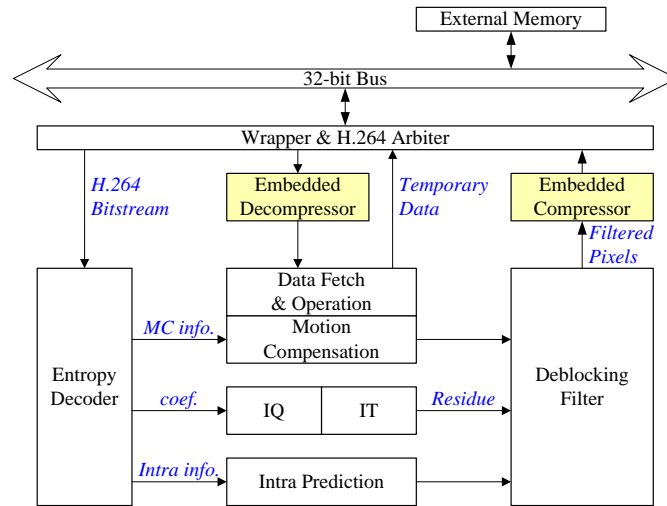


Figure 20 H.264 decoder with proposed embedded compression

The compatible H.264 decoder specification is HD1080+HD720@30fps and works at 150MHz. The compressor converts a 4x2 block from the deblocking filter into a 32 bits segment which is stored into the external memory. Comparing the data access times of the external memory for the system without EC, the data access times of our system is half. The decompressor converts a 32 bits segment into a 4x2 block which is sent to the motion compensation. Since our system bus is 32 bits and the external memory is 32 bits per entry, the system accesses once a data as four pixels. In Table II, we analyze the read times of the motion compensation with/without EC. The worst case is the (Sub, Sub) case. To finish the motion compensation, a 4x4 block needs a 9x9 block. Therefore, the system with/without proposed embedded compressor takes 15/27 cycles. The best case is the (Align, Align) case. Original system with/without embedded compressor needs 2/4 cycles to finish the best case. For the other cases when the required data of motion compensation are not fit for 4x2 block-grids, the access times become increased.

Table 6 All Cases of Read Access Requirement

Case of MV (x, y)	Access Cycles for System without EC	Access Cycles for System with Proposed EC	Reduction of Access Cycles without EC
(Align, Align)	4	2	50
(Align, Not Align)	4	2/3	50/25
(Align, Sub)	9	5	44.4
(Not Align, Align)	8	4	50
(Not Align, Not Align)	8	4/6	50/25
(Not Align, Sub)	18	10	44.4
(Sub, Align)	12	6	50
(Sub, Not Align)	12	6/9	50/25
(Sub, Sub)	27	15	44.4
<b>Average</b>	13.2	6.8~6.9	49.1~48.3

(4) *Experimental Results*

Table 7 PSNR Comparison shows the software result of the proposed algorithm which is integrated with JM16.2. The test sequences are Akiyo, Forman, Mobile, Stefan, and Station. Each test sequence executes 100 frames. And then the average PSNR value is calculated. Results show that the PSNR loss of the proposed algorithm is from 1.27 to 3.94dB.

Table 7 PSNR Comparison

Sequence	Format	H.264 (dB)	Proposed (dB)	PSNR loss
<b>Akiyo</b>	CIF	43.72	41.16	2.56
<b>Forman</b>	CIF	41.23	39.20	2.03
<b>Mobile</b>	CIF	37.61	34.14	3.47
<b>Stefan</b>	CIF	38.82	34.88	3.94
<b>Station</b>	HDTV	39.12	37.84	1.27

Table IV shows the comparison among previous work. It can be found that our proposed hardware provides less hardware complexity and better visual quality. Especially, the proposed decoder just requires one cycle with higher random accessibility for embedded compression without degrading overall system performance. The power consumption of the proposed hardware is better than Lee's [4] and Wu's [10]. Figure 21 shows the Station sequence result of the original system

with EC in HDTV format. The propagation of quality loss is unavoidable but video quality remains acceptable.

Table 8 Comparison Among Previous Work

		Lee's [3]	Wu's [4]	This Work
<b>Technology</b>		CMOS 0.25um	UMC 90nm	UMC 90nm
<b>System</b>		MPEG-2 Decoder	H.264 Decoder	H.264/SVC Decoder
<b>Working Frequency</b>		100MHz	100MHz	150MHz
<b>Processing Data Unit</b>		8x1 Block	4x4 Block	4x2 Block
<b>Total Gate Count</b>		20k	30k	4.9k
<b>Cycle Count for 4x2 Block</b>	<b>Encoder</b>	2 cycles	N/A	2 cycles
	<b>Decoder</b>	2 cycles	N/A	1 cycle
<b>Cycle Count for a MB</b>	<b>Encoder</b>	33 cycles	72 cycles	33 cycles
	<b>Decoder</b>	33 cycles	34 cycles	32 cycles
<b>PSNR Loss</b>		6.08dB~ 10.65dB	1.31dB ~ 4.48dB	1.27dB ~ 3.94dB
<b>Power Consumption</b>	<b>Encoder</b>	N/A	2.78mW	158uW
	<b>Decoder</b>	N/A	1.66mW	86uW

\*The N/A is because of the processing data unit is 4x4 block in MBPZC

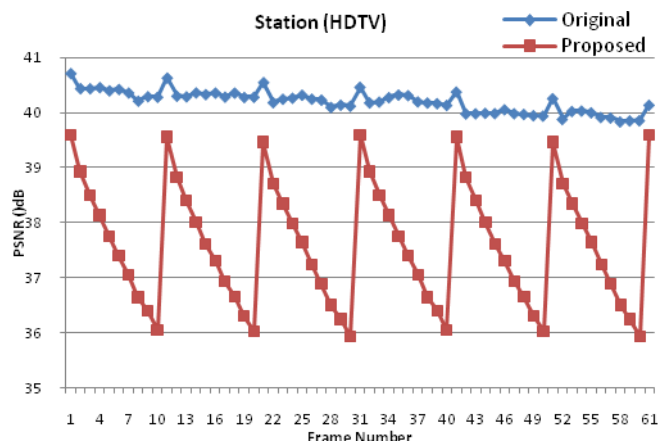


Figure 21 Simulation result of station sequence (HDTV)

### 三、 結論與討論

The objective of this project contains two topics: (1) the Reduced Patterns Comparison Embedded Compressor/Decompressor, and (2) the Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor. We described as below:

First, we have proposed a new embedded compression algorithm for mobile video applications. With these advantages of the proposed EC engine, we can lessen the size of external memory and bandwidth utilization to achieve the goal of power saving. Due to the fixed Compression Ratio, the proposed function is easy to be integrated with an H.264 system. The proposed architecture is synthesized with 90-nm CMOS standard-cell library and the gate counts of the proposed algorithm for embedded compressor/decompressor are 1.8K/3.1K respectively. The average PSNR loss of proposed algorithm is 5.98 dB. The working frequencies are 5 (CIF), 100 (HD 720) and 150 (HD 1080 + HD720) MHz depending on different operation modes.

Second, we have proposed a new embedded compression algorithm for mobile video applications. With these advantages of the proposed EC algorithm, we can lessen the size of external memory and bandwidth utilization to achieve power saving. The pipelined architecture of the proposed decompressor requires 1 cycle, thus the random accessibility becomes better. Due to the fixed CR, the proposed EC algorithm is easier to be integrated with H.264 decoder. From the experimental results, the PSNR loss of the proposed EC algorithm is from 1.27 to 3.94dB. The proposed architecture is synthesized with 90-nm CMOS standard-cell library and the gate counts of the proposed algorithm for compressor/decompressor are 4.0k/0.9k respectively. The working frequency is up to 150MHz@HD1080/720. For power consumption, the compressor is 158uW and the decompressor is 86uW.

#### 四、 參考文獻



- [1] "ITU-T Recommendation H.264 and ISO/IEC 14496-10, Advanced Video Coding for Generic Audiovisual Services", May 2003.
- [2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, July 2003.
- [3] R. Manniesing, R. Kleihorst<sup>1</sup>, R. V. Vleuten<sup>1</sup>, and E. Hendriks, "Implementation of lossless coding for embedded compression," *IEEE ProRISC*, 1998.
- [4] T. Y. Lee, "A New Frame-Recompression Algorithm and its Hardware Design for MPEG-2 Video Decoders," *IEEE Trans. CSVT*, vol. 13, no.6, pp. 529-534, June 2003.
- [5] Y. D. Wu, Y. Li, and C. Y. Lee, "A Novel Embedded Bandwidth-Aware Frame Compressor for Mobile Video Applications," in *Proc. IEEE Intelligent Signal Processing and Communication Syst. (ISPACS)*, pp. 1-4, Feb 2009.
- [6] C. K. Yang and W. H. Tsai, "Improving block truncation coding by line and edge information and adaptive bit plane selection for gray-scale image compression," *Pattern Recognition Letter*, vol. 16, pp. 67-75, 1995.
- [7] T. M. Amarunnishad, V. K. Govindan, and T. M. Abraham, "Block Truncation Coding Using a Set of Predefined Bit Planes," in *Proc. IEEE Int. Conf. Computational Intelligence and Multimedia Applications (ICCIMA)*, vol. 3, pp. 73-78, Dec 2007.
- [8] "Draft ITU-T recommendation and final draft international standard of Joint Video Specification (ITU-T Rec. H264-ISO/IEC 14496-10:2005 AVC)," *JVT G050*, 2005.
- [9] J. Kim, and C. M. Kyung, "A Lossless Embedded Compression Using Significant Bit Truncation for HD Video Coding," *IEEE Trans. CSVT*, accepted, 2010.

- [10] T. M. Liu, and et al., “A 125/spl mu/w, fully scalable MPEG-2 and H.264/AVC video decoder for mobile applications,” in *Proc. IEEE Int. Solid-State Circuits Conference (ISSCC)*, pp. 1576–1585, 2006.

## 五、 計畫成果自評

在此計畫執行第二年中，我們提供兩個可應用於行動通訊之下世代低功耗視訊解碼器元件，其中分別為：

- (1) *The Reduced Patterns Comparison Embedded Compressor/Decompressor*
- (2) *The Bitplane Truncation with Pattern Comparison Coding Embedded Compressor/Decompressor*