國 立 交 通 大 學

電子工程學系　電子研究所碩士班

碩 士 論 文

適用於 H.264/MPEG-AVC 可調式視訊編碼之快速幀內預測演算法與設計

Fast intra prediction algorithm and design for H.264/MPEG-4 AVC scalable extension

研究生：溫孟勳

指導教授：張添烜 博士

中華民國　一〇〇 年　九 月


E
S
A
1896

適用於 H.264/MPEG-AVC 可調式視訊編碼之快速幀內預測演算法與設計

Fast intra prediction algorithm and design for H.264/MPEG-4 AVC scalable extension

研 究 生：溫孟勳　　Student: Meng-Hsun Wen

指導教授：張添烜 博士　　Advisor: Dr. Tian-Sheuan Chang

國 立 交 通 大 學

電子工程學系 電子研究所碩士班

碩 士 論 文

A Thesis

Submitted to Department of Electronics Engineering & Institute of Electronics

College of Electrical and Computer Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of Master of science

in

Electronics Engineering

September 2011

Hsinchu, Taiwan, Republic of China

中華民國 一〇〇 年 九 月


E
S
A
1896

適用於 H.264/MPEG-AVC 可調式視訊編碼之快速幀內預測演算法與設計

研究生: 溫孟勳　　指導教授: 張添烜 博士

國立交通大學

電子工程學系電子研究所碩士班

摘　要

可調性視訊編碼提供高壓縮效率，並且能在一次編碼時，結合多重畫面大小、多重影像視覺品質以及多重影像播放速率，再根據不同的應用，依照系統頻寬以及終端處理能力，從相同的資料流擷取出需要的部分重建，然而，這也大幅增加設計上的複雜度，特別是針對即時性高畫質影像壓縮。

為了解決此問題，本論文提出了一種兩步驟的快速框內編碼預測演算法以及它的硬體設計，第一步先將畫面經由整數離散餘弦轉換與哈達瑪轉換，得到各個頻率值，並藉此判斷畫面平坦程度來決定畫面切割區塊大小，第二步是重複使用第一步的整數離散餘弦轉換與哈達瑪轉換來預測畫面的紋路方向，加以決定 4x4 候選模式與 16x16 候選模式，並利用 4x4 候選模式推測出 8x8 候選模式。實驗結果顯示，與 JM 的全搜索法相比，我們的做法可以節省 80％以上的候選模式，並且維持相異不大的壓縮質量(平均 BD-PSNR 差異：CIF 為-0.01dB，1080p 為+0.12dB；平均 BD-Rate 差異：CIF 為+0.01％，1080p 為- 3.13％)。另外在硬體實作上，不同的區塊大小框內預測可由共用運算單元計算，並且以交錯的方式同時處理兩個巨圖塊，可避免資料相依所導致的計算時間浪費。使用 90 奈米 CMOS 製程，則等效面積為 148k gate counts，以及需要 143k-bits 的 SRAM buffer，在

135MHz 頻率下，此編碼器可支援到的三層畫質，三層影像大小（CIF，SD 480 和 HD1080P）和高達每秒 60 張畫面的壓縮速率。

Fast intra prediction algorithm and design for H.264/MPEG-4 AVC scalable extension

Student: Meng-Hsun Wen Advisor: Tian-Sheuan Chang

Department of Electronics Engineering & Institute of Electronics

National Chiao Tung University

Abstract

H.264/MPEG-4 AVC scalable extension can provides high compression efficiency and high visual quality with spatial, temporal, quality scalabilities for diverging decoding terminals. However, this also significantly increases the design complexity, particularly for real time high definition video encoding.

This thesis proposes a fast two step intra prediction algorithm and its hardware design to meet real time demands. The first step is the intra block size decision by distinguishing the block smoothness through selected AC coefficients. The second step is a transform based mode candidates for 4x4 block and merged 4x4 mode candidates for 8x8 block. The experimental results shows that we can save more than 80% candidate modes, with similar quality (average BD-PSNR difference: -0.01 dB for CIF, +0.12dB for 1080p, average BD-Rate difference: +0.01% for CIF, -3.13% for 1080p), when compared with JM full search method. The resulted hardware design shares single prediction units for different block size computation and interleaved computes two macroblocks to avoid data dependency with embedded quality layer processing in the reconstruction loop. The implementation with 90nm CMOS process costs 148k gate counts, and 17.9k bytes SRAM buffer under 135MHz operating

frequency to support processing rate of three quality layers, three spatial layers (CIF, SD 480p and HD 1080p) and up to 60 frames per second.

誌　　謝

首先感謝我的指導教授－張添烜博士。在大學的專題研究以及研究所兩年的碩士生涯，每當我遇到問題時，總給予我相當多的建議，教導我掌握大方向並且用有效率的方式去解決，另外老師也在修課上給予我信任與尊重，讓我有充分的發揮空間。

同時也非常感謝我的口試委員們，清華大學電機系陳永昌教授與交通大學電子工程系李鎮宜教授。謝謝你們百忙之中抽空前來指導，給予我相當多寶貴的建議，讓我的論文更加充實與完備。

感謝實驗室的夥伴們，特別是國龍學長，帶領我進入影像壓縮技術的殿堂，你的包容與耐心的引導，讓我成長很多，還有佑昆學長在我大學專題的時候，耐心的指導我，宇晟學長幫我解決很多工作站相關問題，國振學長在修課上給我相當多建議，易群學長相當活潑開朗，帶給實驗室許多歡樂的氣息。謝謝之悠學長、筱珊學姐、元歆學長、博雄學長、瑩蓉學姐、奕均學長與宥辰學長教導我許多硬體方面知識。謝謝亮齊，從大學專題，研究所修課，以及 IC 設計競賽，常常一起合作，跟你共事讓我學習許多，英佑與克嘉，很謝謝你們時常給我的鼓勵與關心。感謝輔仁、兆傑、珊榕、碩文與 Daniel，你們也讓我的碩士班生涯增添不少快樂時光。

感謝我的家人們，爸爸、媽媽、以及姐姐，你們總是在後面默默的支持，給我最溫暖的依靠，讓能我專心的完成學業。

最後謝謝所有關心我的人，在這裡謹以這篇論文獻給你們。


E S A
1896

Contents

List of Figures

List of Tables


E S A
1896

1. Introduction

With the evolution of video compression technology, more and more related applications appear, such as digital television, mobile video, Blu-ray DVD, and other multimedia devices. In the past, a variety of compression methods have been proposed, such as the recent H.264/AVC standard[1][2], can provide higher and higher compression efficiency, but its compression and transmission can only provide a fixed screen size, frame rate and image quality. In other words, different conditions like network bandwidth (such as modem, cable) or terminal applications (such as mobile phones, high-definition digital television) require several encoding times and individual transmission.

Scalable video coding (SVC) [3],the extension of H.264/AVC standard, is developed by the Joint Video Team (JVT) of ISO / IEC Motion Picture Expert Group (MPEG) and ITU-T Video Coding Expert Group (VCEG) and is capable to offer flexible scalabilities in temporal, spatial and quality domains by a single bit-stream[6]. According to different applications, receivers can extract part of bit-stream and decompress for their requirements.

1.1. Motivation

With high resolution, real time face to face image transmission, 3D or Blu-ray, etc…, more and more new technology does change the world and bring the people into the different life style. These new applications not only bring the benefits of daily life but also accompanied by high-capacity data transmission and high-capacity data storage. The requirement of high data compression and fast algorithm can be expected

higher and higher.

SVC can achieve quite high data compression with different coding features. One of these features, intra prediction, can provide good compression efficiency. With rate distortion optimization process, intra prediction can detect all candidates to find the best prediction.

However, it is hard to meet the real time challenges of Rate-Distortion optimized intra prediction. Therefore, this research will provide a better efficient fast intra algorithm and its hardware design for SVC. The proposed design not only achieves timely computation, but also reduces hardware resource requirements.

1.2. Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 overviews the SVC standard, Chapter 3 shows the architecture of our SVC encoder, Chapter 4 presents fast intra prediction and comparison, and Chapter 5 shows the proposed hardware architecture and results. Finally, Chapter 6 concludes this thesis.

2. Overview of SVC Standard

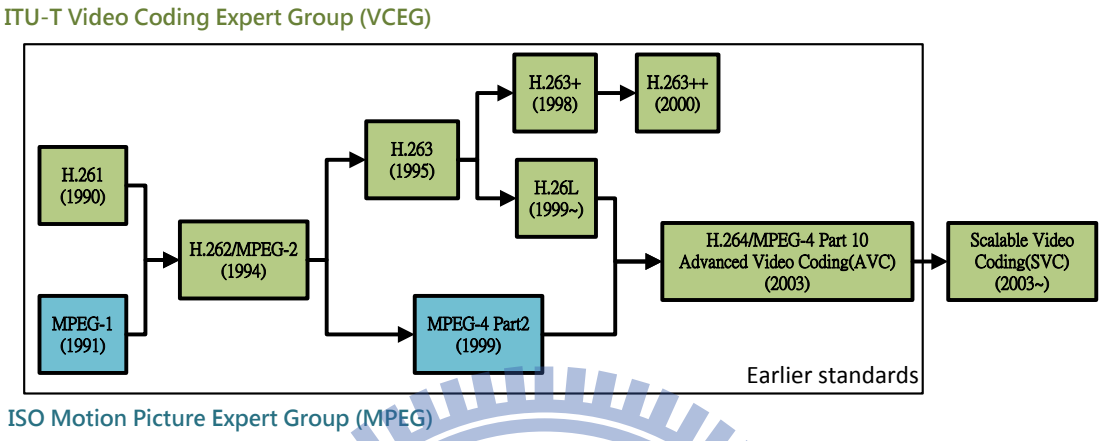


Fig. 2-1. The history of video coding standards

SVC, the extension of H.264/AVC, is developed by The Joint Video Team of the ITU-T VCEG and the ISO / IEC MPEG. Comparing with earlier compression standards (Fig. 2-1), SVC can encode multi-temporal layers (the number of frame per second), multi-spatial layers (frame size), and multi-quality layers (image visual quality) into a single bit-stream, and decode partial bit-stream depending on application constraints, such us power consumption and Bandwidth, to achieve "scalable".

For example, in Fig. 2-2, SVC encodes original input sequence with three temporal layers, three spatial layers, and three quality layers. Then, Depending on network condition of transmission environment, minor packets will be eliminated. Finally, cell phone of decoder part receives the bit-stream packet might only decode part of the bit-stream, only decode two temporal layers, one spatial layer and two quality layers due to limitation of transmission speed, low power consumption and

small screen. Similarly, HDTV can provide powerful decompression capability, and higher resolution. Therefore, if the transmission condition is allowed, HDTV can decode three temporal layers, three spatial layers, and three quality layers to achieve best performance.

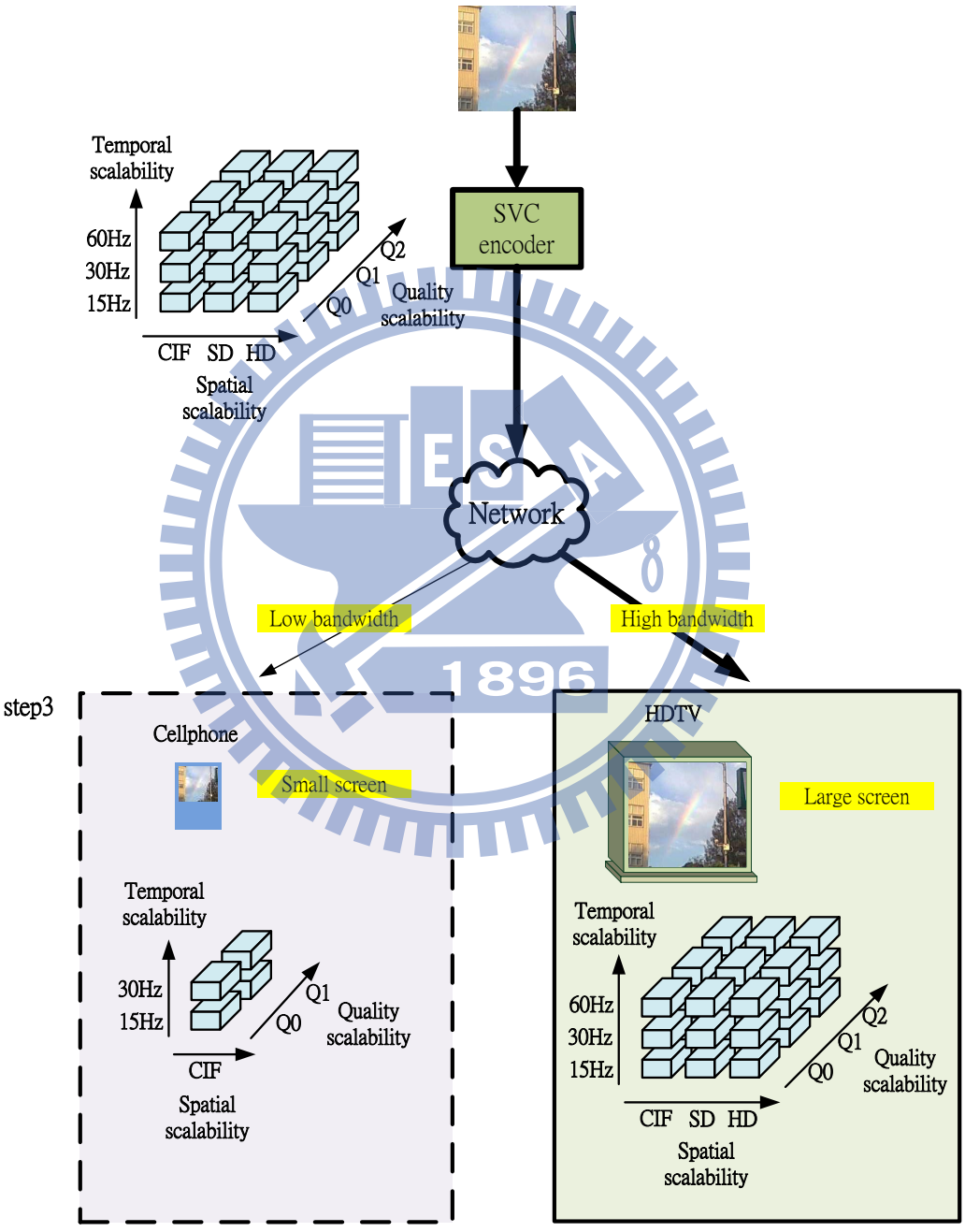


Fig. 2-2. The concept of SVC

2.1. Fundamentals of SVC

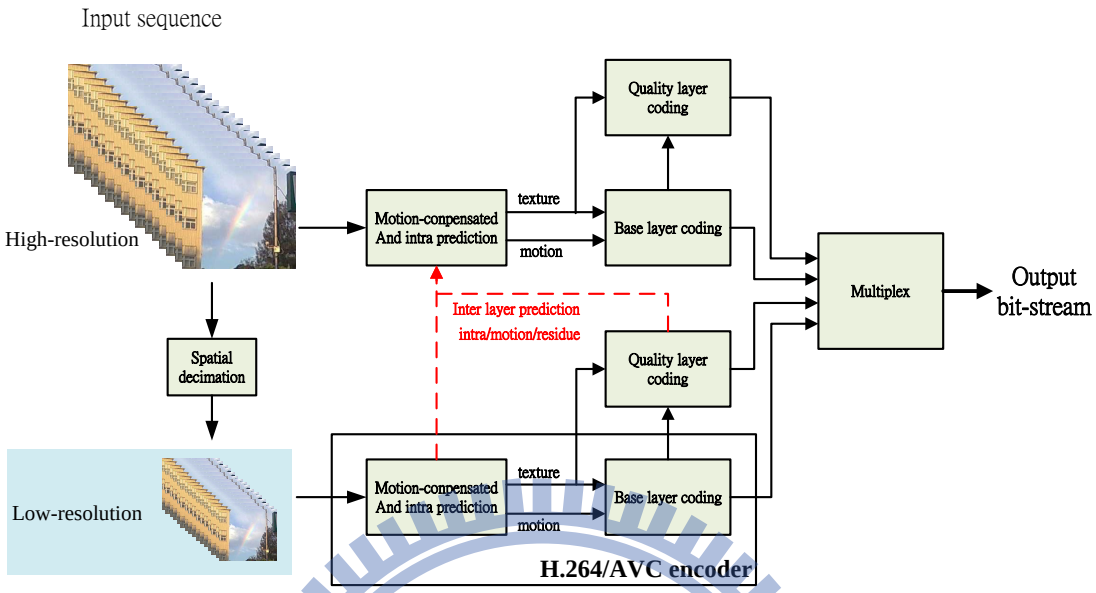


Fig. 2-3. Architecture of an SVC encoder

Fig. 2-3 shows the architecture of a SVC encoder, where portion with solid line indicates basic H.264/AVC encoder. In the beginning, the high-resolution sequence is down-sampled to obtain the low-resolution sequence (lower left corner of Fig. 2-3). One low-resolution frame will be sent into the H.264/AVC encoder to go through motion-compensation, intra prediction and base layer coding for basic H.264/AVC encoding. Then, quality layer coding will add un-delivered coding information into the bit-stream for multi-quality layers. When the low-resolution frame processing has been done, processing for high-resolution frame (as Fig. 2-3 upper of right hand side, high-resolution sequence) will be started in a similar way. These coding information like motion vectors, residuals and texture of low-resolution frame (as Fig. 2-3 dash arrow shows) can provide more efficient prediction to multi-spatial layers [8]. Finally, temporal scalability is available with hierarchical coding structure [7][9].

2.2. Components of SVC Encoder

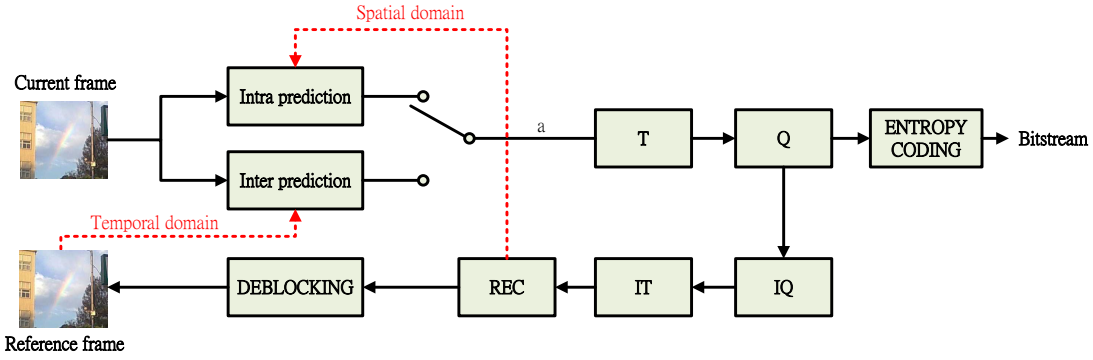


Fig. 2-4. Architecture of an H.264/AVC encoder

Due to the requirement reference data for SVC is based on H.264/AVC (in Fig. 2-3 dash line), H.264/AVC encoder will be introduced first as Fig. 2-4 shows.

First, current frame (upper left corner as Fig. 2-4 shows) will be sent to intra prediction and inter prediction to eliminate the redundant spatial and temporal data. The most cost effective prediction residue will be transformed (T) and quantized (Q) to erase minor information to achieve better compression ratio. Finally, entropy coding compresses these residues. On the other hand, quantized data is also reconstructed by inverse quantization (IQ), inverse transform (IT), and deblocking filter to be the reference for the following prediction. (Fig. 2-4 red dash line shows)

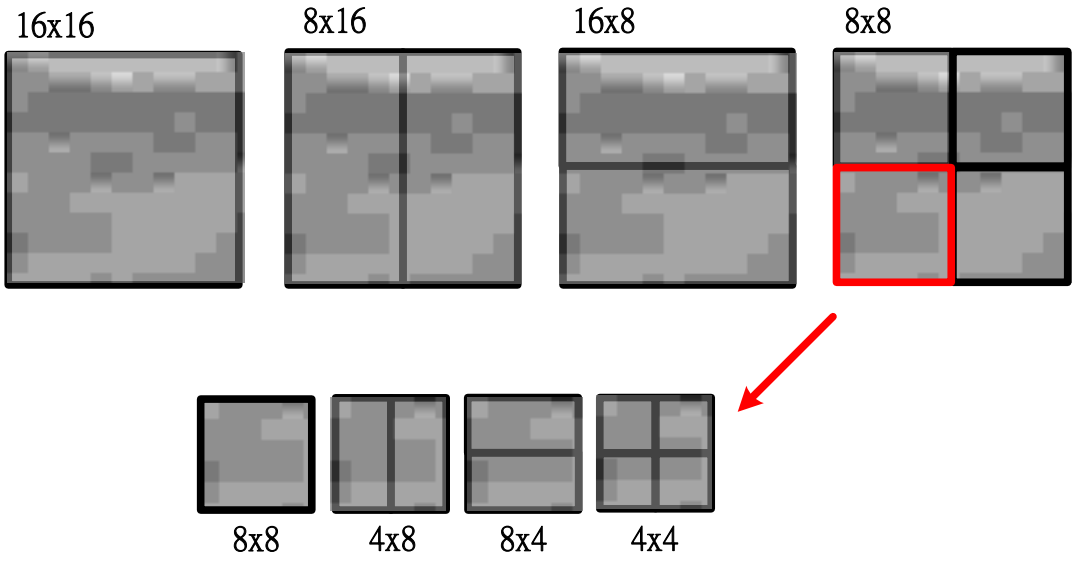


Fig. 2-5. Block segmentation of inter block

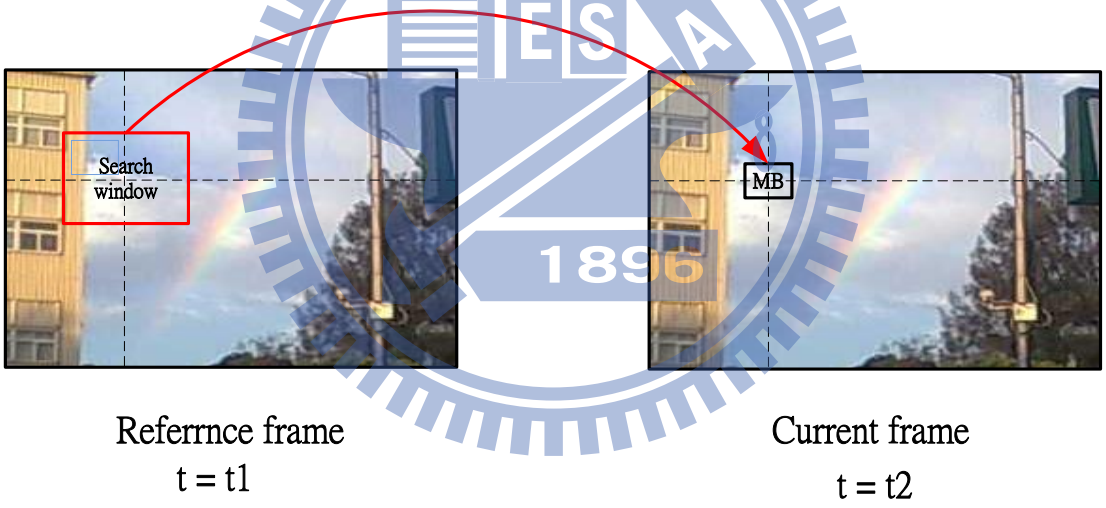


Fig. 2-6. Search window for inter prediction

For inter prediction, one MB (Macro-block) can be partitioned into 4 types of partition size (Fig. 2-5), including 16x16, 16x8, 8x16, 8x8 block types. Each 8x8 block can be further partitioned into 8x8, 8x4, 4x8, 4x4 partition sizes, there is totally 259 block partition combinations. The most cost effective partition block from 259

combinations will be the best inter prediction.

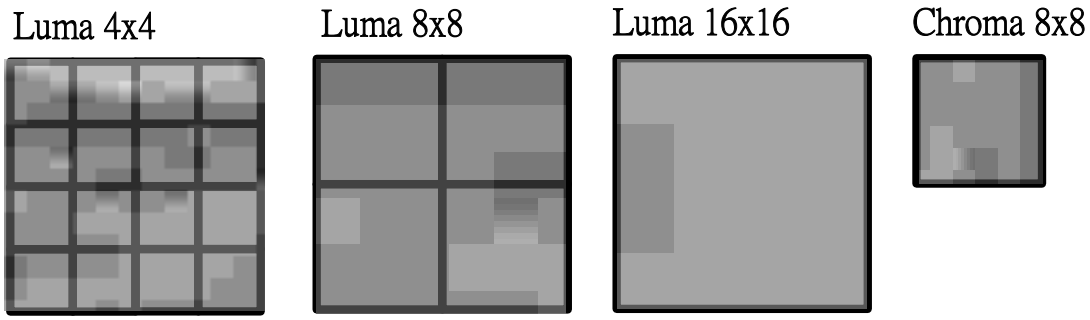


Fig. 2-7. Block segmentation of intra prediction

For intra prediction (Fig. 2-7), luma block includes three different partition sizes for luma, intra 16x16, 8x8, 4x4 and one partition size for chroma block. Intra 8x8 and 4x4 have 9 prediction modes (Fig. 2-8 a,b), and intra 16x16 and intra chroma only have 4 prediction modes. (Fig. 2-8 c,d)

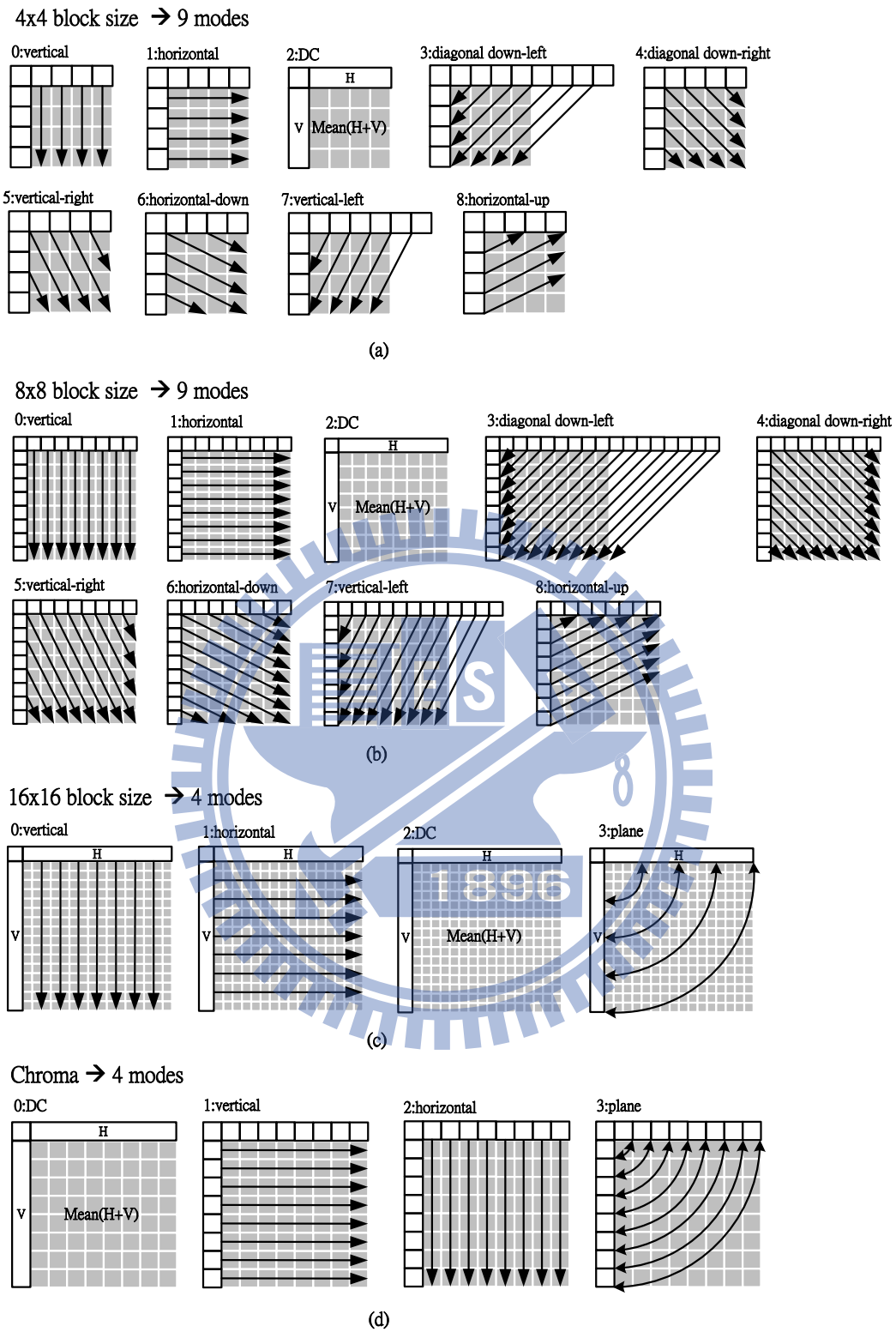


Fig. 2-8. Modes of intra prediction: (a) 4x4 block size; (b) 8x8 block size;

(c) 16x16 block size; (d) chroma block

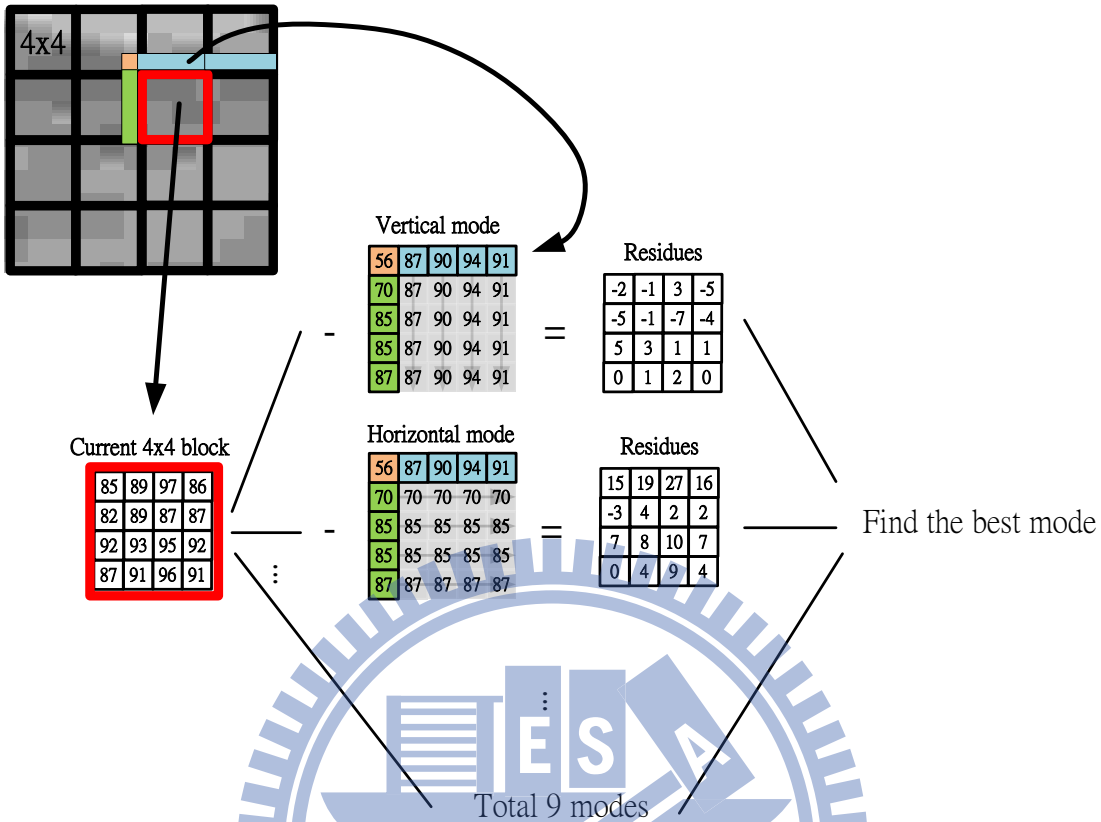


Fig. 2-9. The modes of 4x4 block size

For intra 4x4 example in Fig. 2-9, every 4x4 block will use the reconstructed pixels from left, upper left, up and upper right side for prediction. The mode with the minimum cost is the best mode of this 4x4 block. Approaches for 8x8 and 16x16 block size type are similar. Finally, we can select the most cost effective block size type from these three as the best intra prediction.

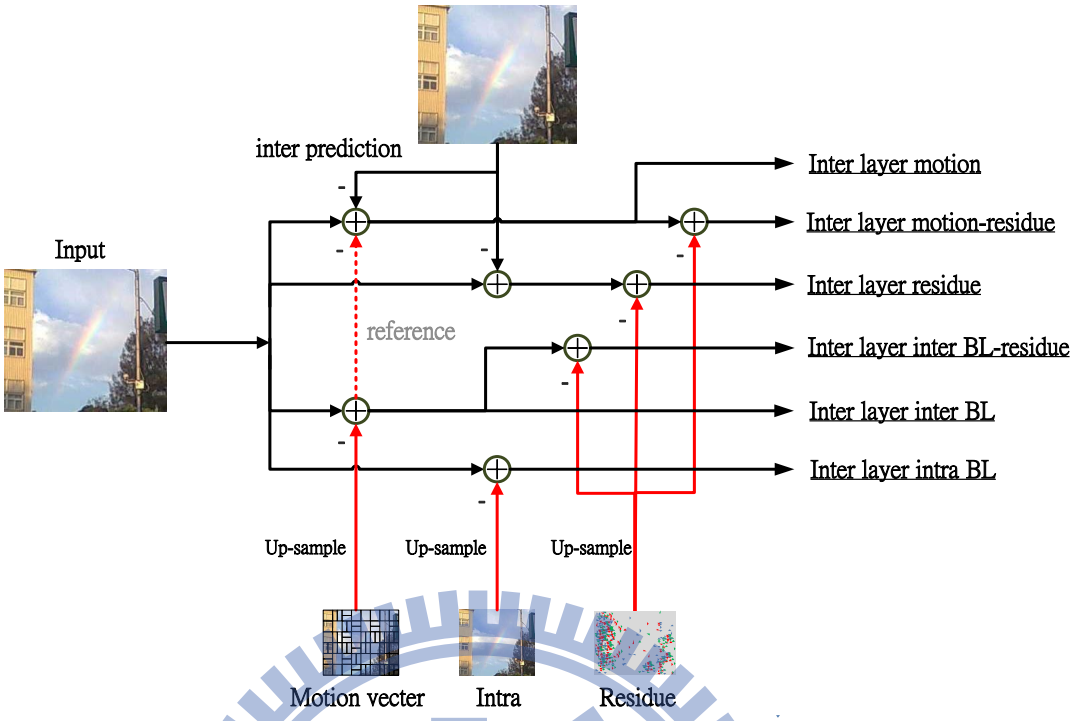


Fig. 2-10. Prediction modes of inter-layer prediction

To provide multi-spatial layers, SVC encoder added six inter-layer prediction modes (Fig. 2-10) compared with H.264/AVC. Higher spatial layer will employ the up-sampled motion vectors, residues, texture from lower spatial layer to support the prediction (red line) and erase the redundant information.

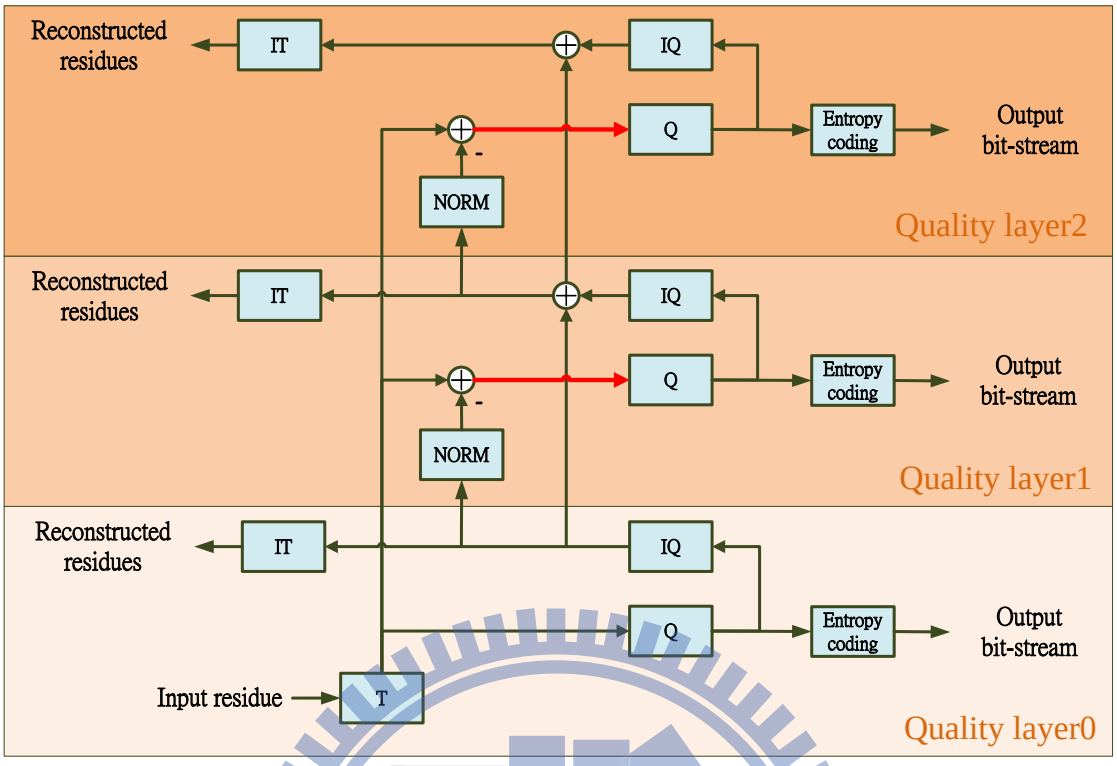


Fig. 2-11. Block diagram of quality enhancement coding in SVC

SVC also provides multi-quality layers (Fig. 2-11). Due to quantization procedure, some information is missed without encoding into bit-stream. In order to enhance visual quality, SVC encoder proceeds (as Fig. 2-11 quality layer 1, 2) the rest of transformed coefficients with smaller quantization parameter to encode more detailed information into bit-stream to achieve multi-quality layers.

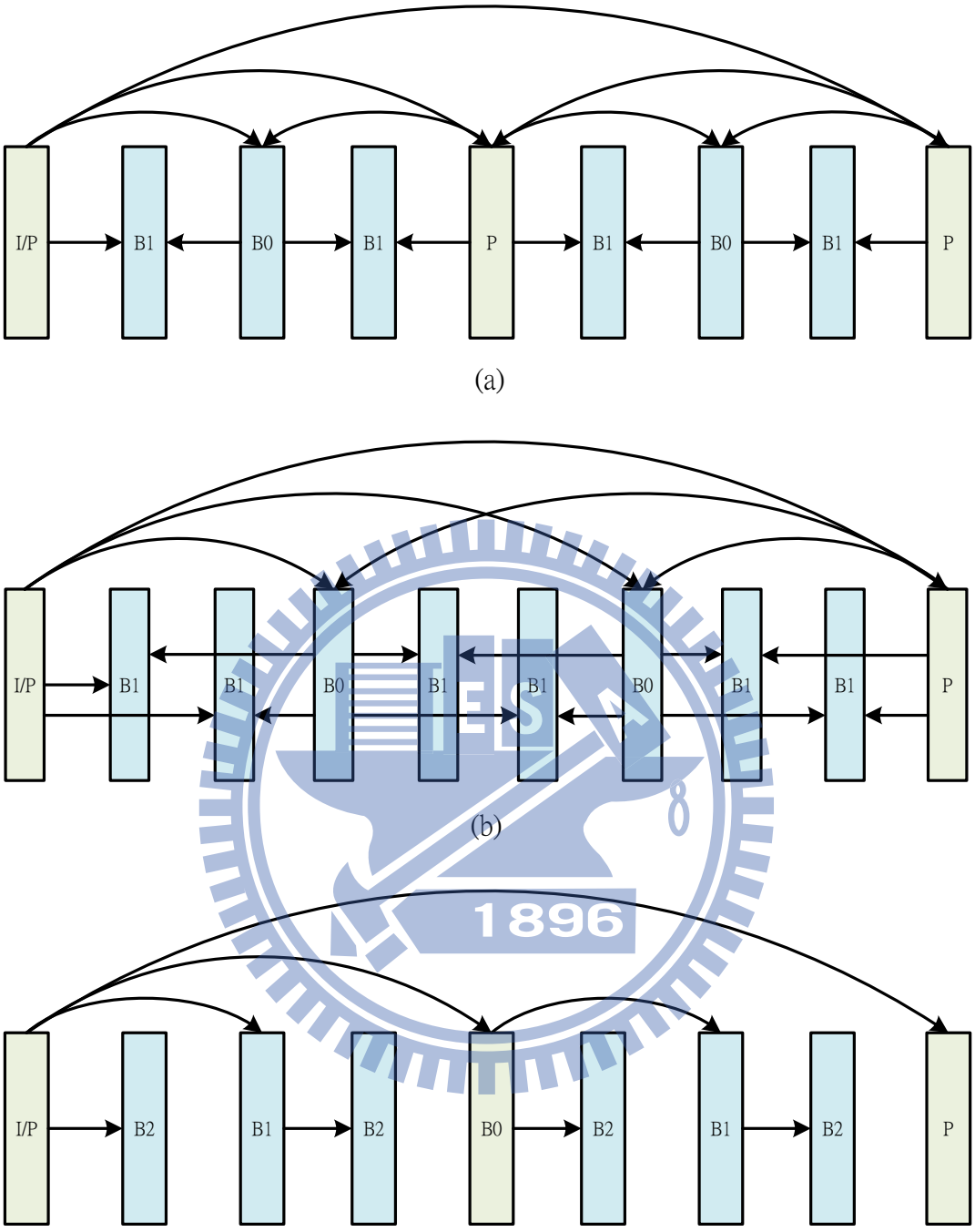


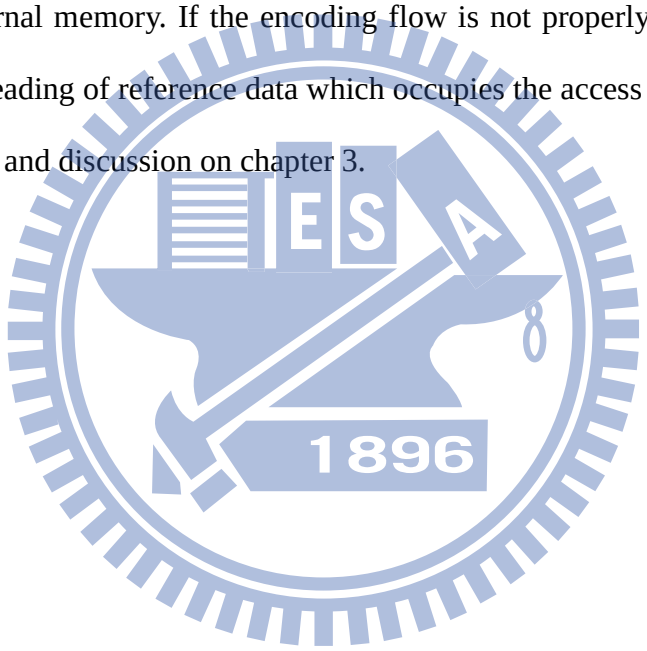
Fig. 2-12. Hierarchical coding structure for temporal scalability:

(a) hierarchical-B structure; (b) Nondyadic hierarchical prediction structure.; (c) Hierarchical coding structure with delay of zero

About multi-temporal layers of SVC, there are different designs by the different limitation and requirement (Fig. 2-12)[6][7][9]. The three most popular designs are (a) hierarchical-B structure. (b) Nondyadic hierarchical prediction structure. (c) Hierarchical coding structure with delay of zero.

2.3. Summary

SVC needs high data computation due to a lot of prediction modes. Especially, when we proceeds inter prediction, it needs to access the pixel data of reference frames from external memory. If the encoding flow is not properly design, this may induce repeated reading of reference data which occupies the access bandwidth. There is further analysis and discussion on chapter 3.



3. Architecture Of Our SVC Encoder

The target of our SVC encoder is 3 quality layers, 3 spatial layers (CIF, SD 480p, and HD 1080p), and up to 60 frames per second.

3.1. Coding Methodology

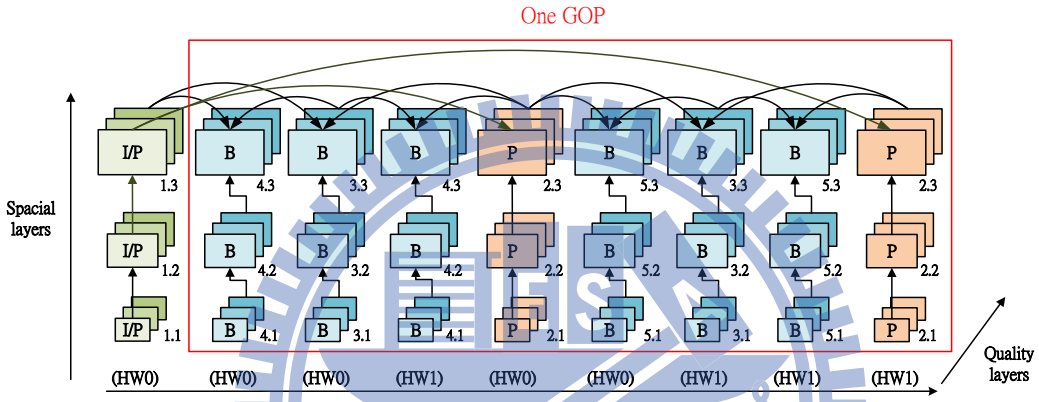


Fig. 3-1. Coding flow chart

There are some memory analyses for H.264/AVC scalable extension encoder [4][5], and we adopt “frame parallel encoding scheme [10]”. Under the structure of frame-parallel encoding scheme, when two P or two B frames proceed inter prediction at the same time, search window can be shared to decrease the access of the external memory and reduce the internal memory.

Fig. 3-1 shows coding flow of one GOP (group of pictures) under frame parallel encoding scheme. It starts with I frame, from CIF, SD 480p to HD 1080p frames (1.1 →1.2→1.3). Then we encode two P frames at the same time (2.1 →2.2→2.3). Finally, we encode triple times two B frames (3.1 →3.2→3.3→4.1 →4.2→4.3→5.1

→5.2→5.3).

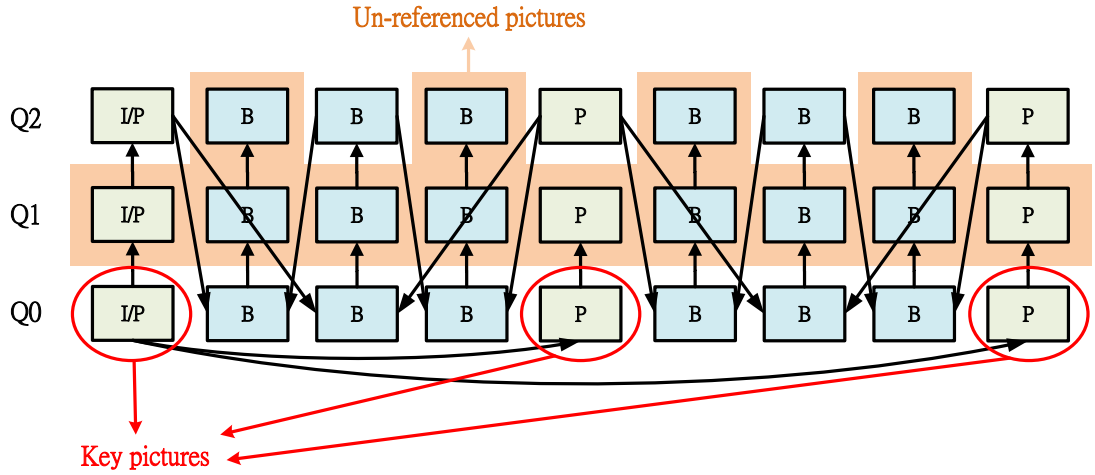


Fig. 3-2. Key picture concept

About reference frames, the Key Picture Concept [11] is adopted as Fig. 3-2 shows. I and P frames are considered to be “key pictures” and use base layer as reference frames in their encoding. But B frames are considered as “non-key pictures” and use highest quality layers as reference frames. Because some frames will not be referenced, there is no requirement to full reconstruct and the following procedures like deblocking can be eliminated. We only need to calculate and store temporary reconstructed data in internal memory for the use of intra prediction.

3.2. Hardware Architecture and Encoding Schedule

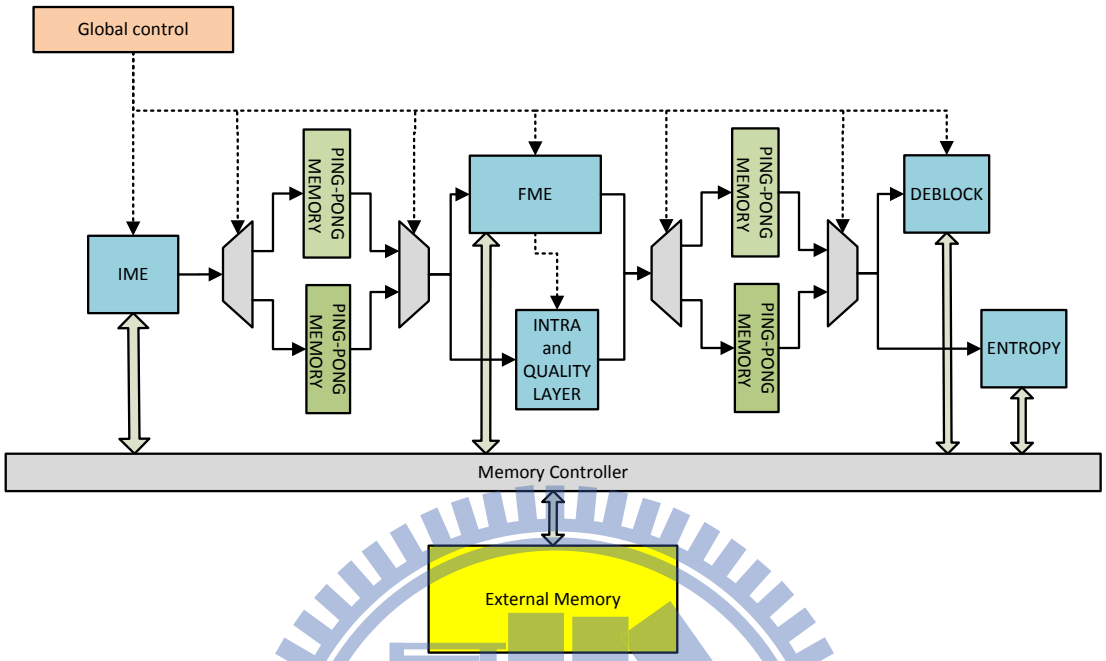


Fig. 3-3. Hardware architecture of our SVC encoder

Fig. 3-3 shows our SVC hardware structure. The first pipeline stage contains IME module. The second pipeline stage contains FME module and INTRA module. And the last pipeline stage contains deblocking filter module and entropy coding module. The Ping-Pong memories are adopted between each stage. And each stage can proceed two MBs data at the same time. (Please refer 5.1 for further discussion)

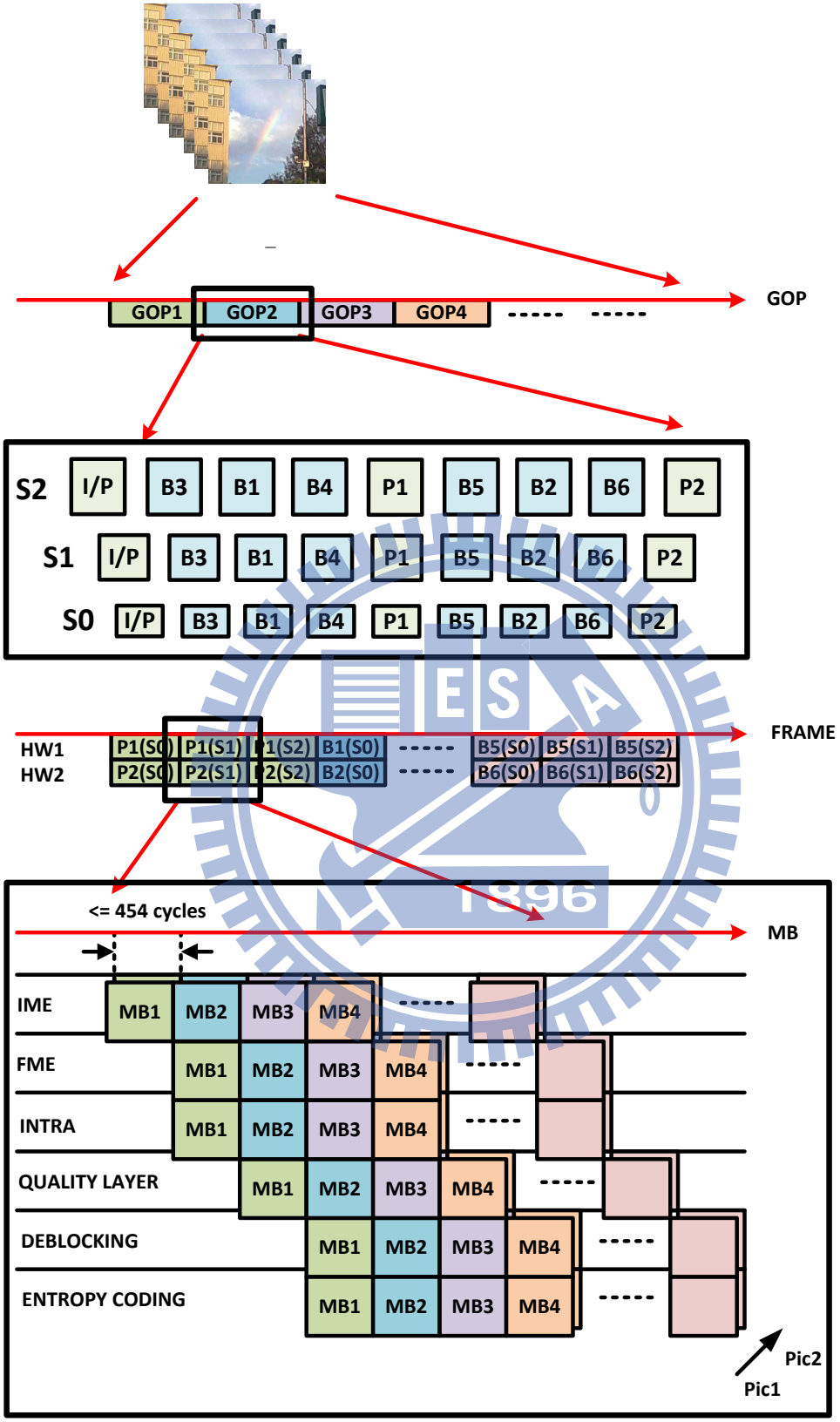


Fig. 3-4. Encoding schedule of our SVC encoder

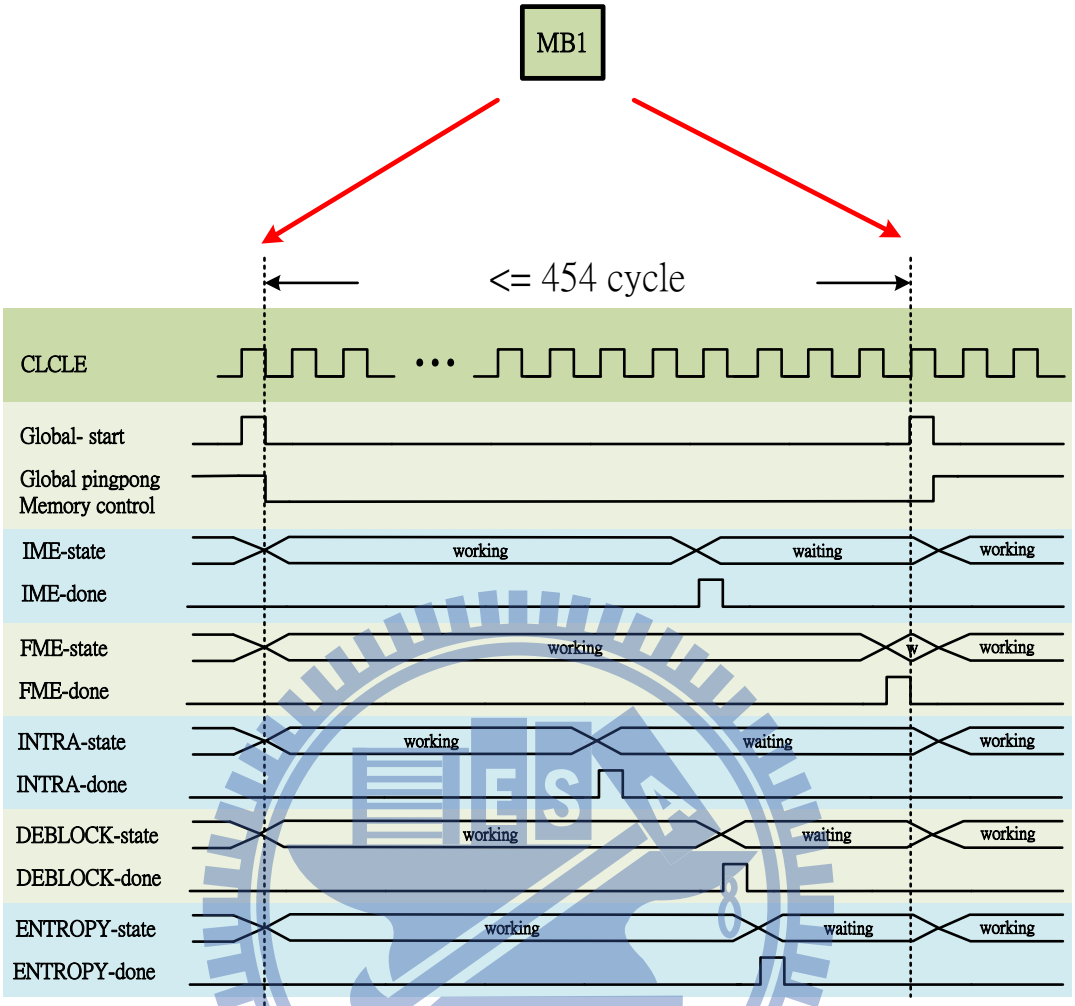


Fig. 3-5. Global control

Fig. 3-4 shows how to proceed the pipeline MB data in hardware. In the first, GOP-axis shows the sequence will be divided into GOPs. Every GOP includes 24 frames. The Frame-axis shows all 24 frames of one GOP. Two frames will be encoded at the same time. The MB-axis shows one frame will be divided into group of MBs according to pipelined encoding flow. At the same time period. Therefore, INTER IME, INTER FME, INTRA PREDICTION, DEBLOCKING and ENTROPY CODING modules will be calculated at the same time period.

In the beginning, the global control sends start signal to each pipeline module as Fig. 3-5 shows, and the status of each module will be switched from waiting to working. Every module extracts the data from the external memory or the Ping pong memories of previous stage to proceed the algorithm and storage the result into the ping-pong memories of next stage. When the algorithm has been done, the status of each module will be switched to waiting and send the ending signal to the global control. After all modules have sent ending signal, the global control will send next start signal to repeat the same procedure again.

3.3. Summary

Although we have introduced the frame-parallel encoding scheme to reduce the data access of external memory and the key picture concept to decrease the amount of data reconstruction in this chapter. However, due to the frame-parallel encoding scheme demand, the intra prediction module needs to parallel proceed two MBs. To simplify hardware design, next chapter will focus on the fast intra prediction algorithm.

4. Fast Intra Prediction For SVC

4.1. Introduction of Reference Work

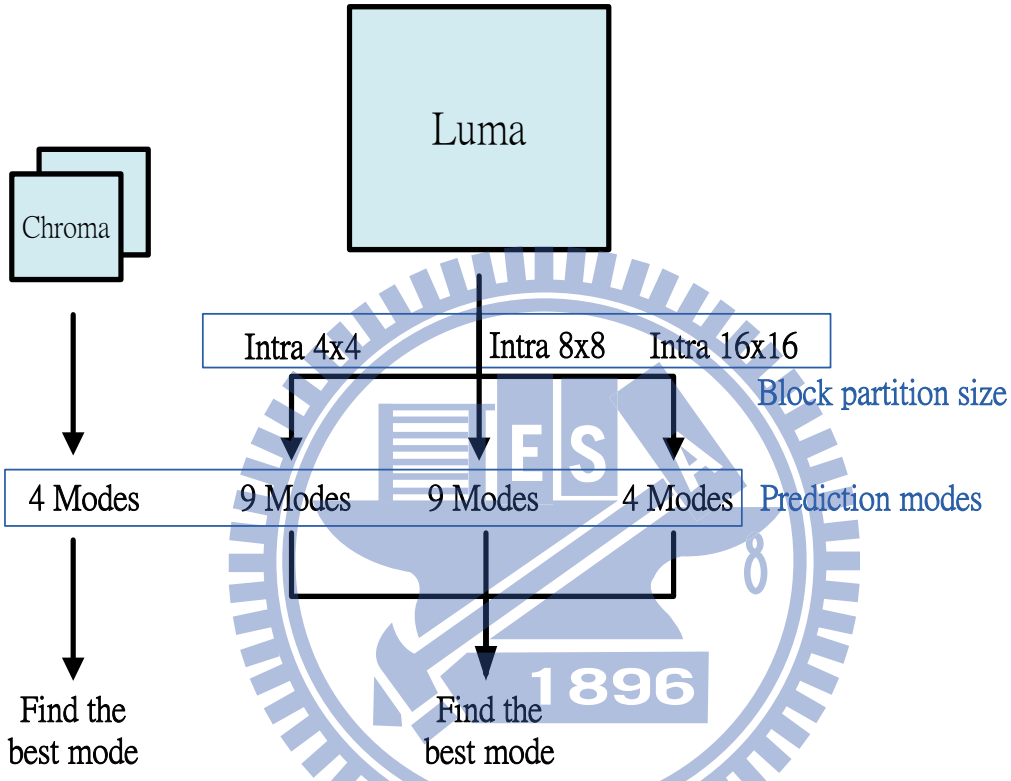


Fig. 4-1. Block partition sizes and prediction modes of intra Prediction

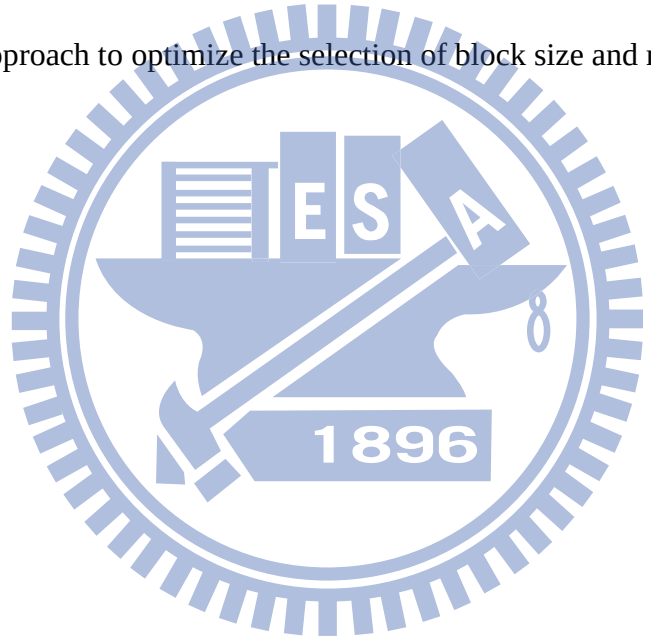
There are a lot of researches proposed intra fast algorithms. These proposals can be divided into two types.

The first type is to determine block partition size through the smoothness of MB before any mode prediction. A smoother MB will be partitioned into larger block partition size. There are some related studies [16]. For examples, the gradient vector at each pixel [12], the entropy of brightness of the MB pixels [13], the total pixel

difference of inner and outer of MB boundary [14], or the DC/AC energy ratio from the 16x16 DCT of MB [15] can help us to determine the MB is flat or not.

The second type is to quickly select mode candidates from all prediction modes. The principle is to determine the directional tendency of MB, and modes with adjacent directions usually has higher probability to be chosen. There are also a lot of researches for this prediction type, like [17]-[21].

However, none of these approaches have optimized the two issues together to attain better quality while reduce the hardware cost significantly. Our study will present a new approach to optimize the selection of block size and mode together.



4.2. Overall Fast Intra Prediction

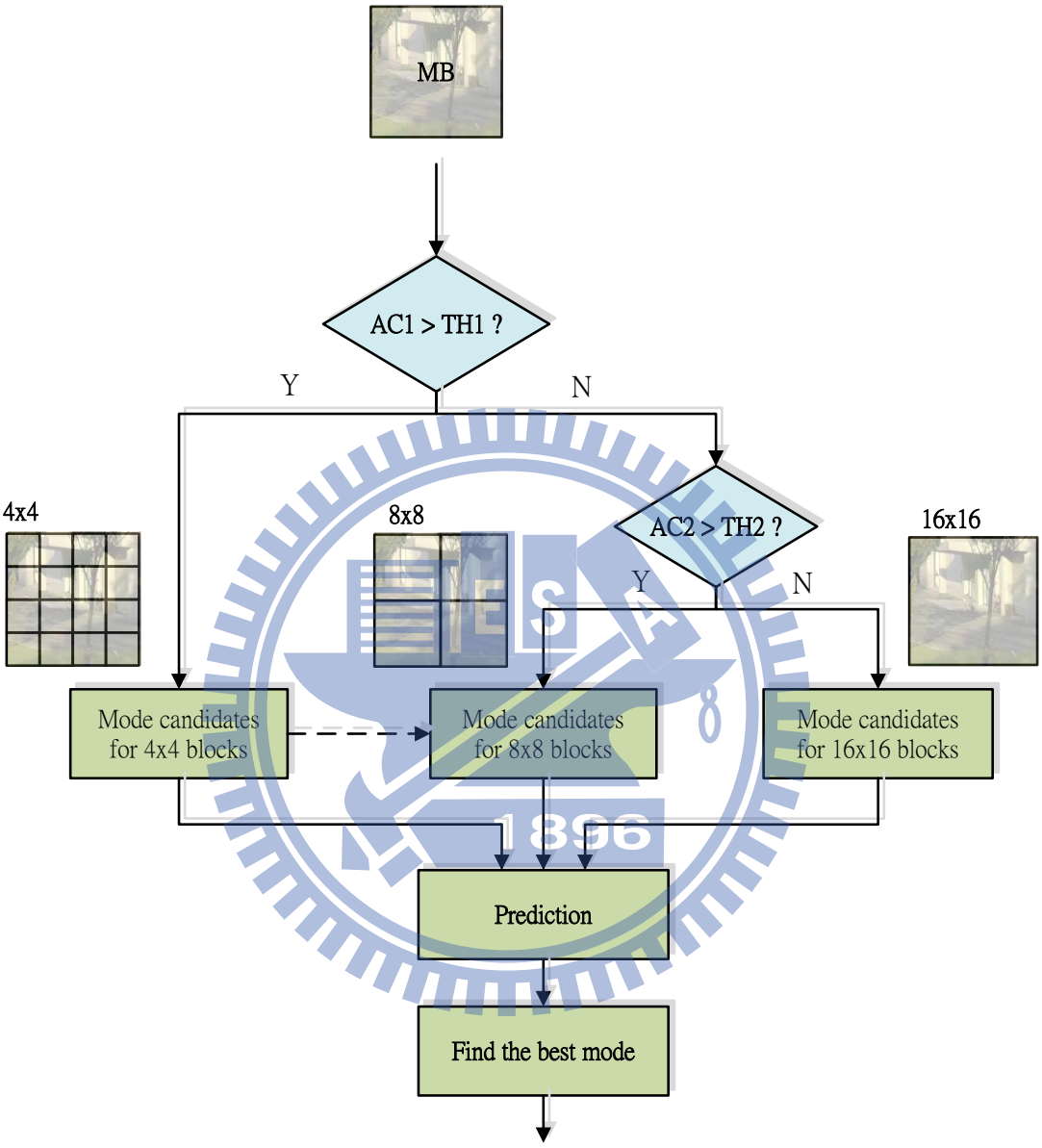


Fig. 4-2. Overall fast intra prediction diagram

Fig. 4-2 shows the proposed algorithm. We first decide the possible block type through the smoothness test of the MB. Then, we use transform domain based approach to decide the best mode prediction of the selected block type.

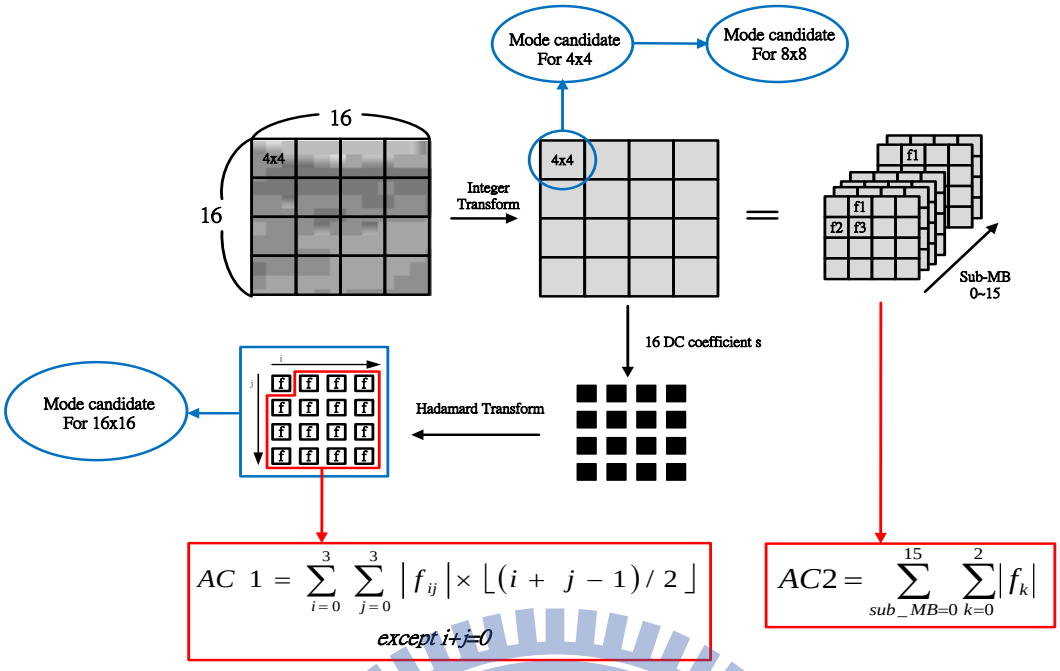


Fig. 4-3. Calculation process for mode candidates and block size parameters

4.3. Fast Intra Block Size Prediction Algorithm

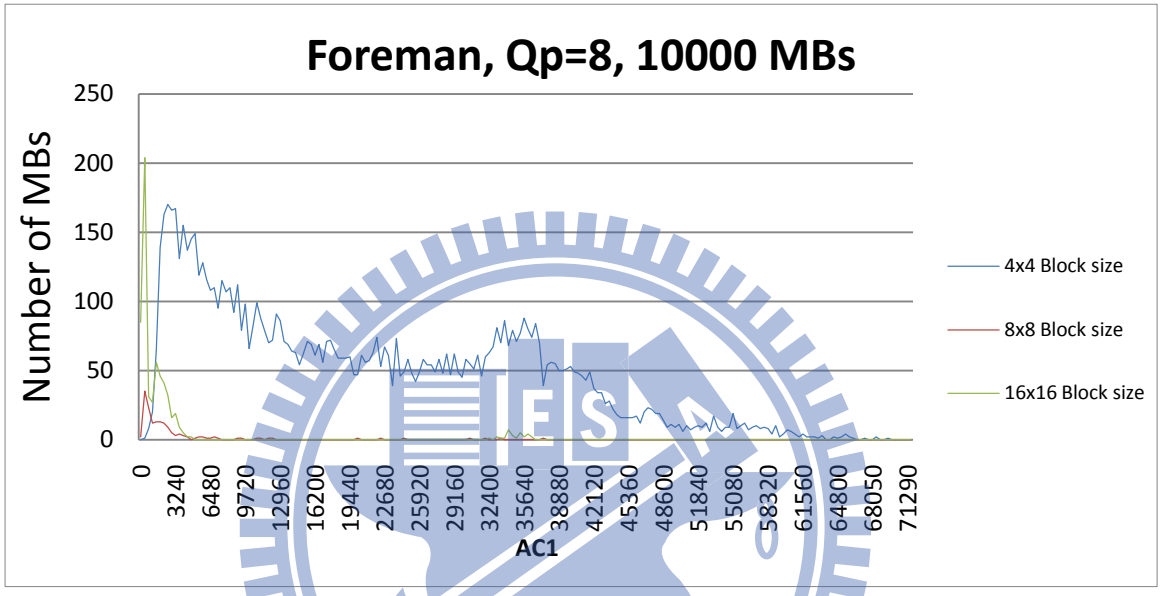
4.3.1. Determination of 4x4 Block Size

The concept of the block type decision exploits the smoothness characteristic of a MB. In general, a smoother region will tend to select the larger block size. Previous approaches used the AC/DC ratio to calculate the flatness. However, it requires complex computation. Instead, our first flatness test uses the sum of AC components in a 4x4 Hadamard transformed block as an index, AC1, which is defined as

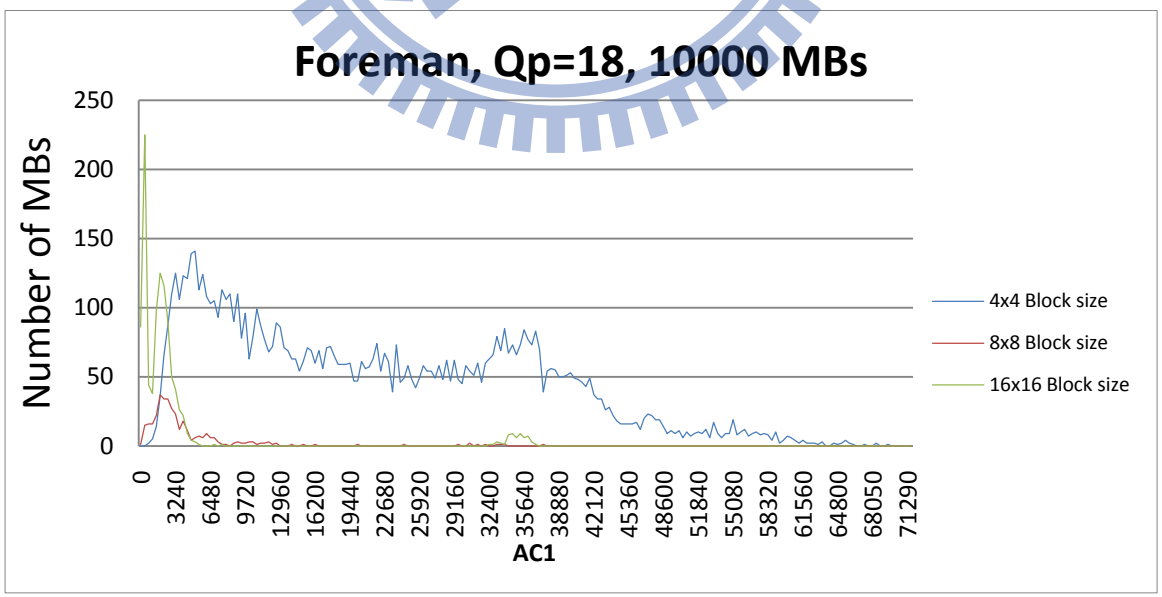
$$AC\ 1 = \sum_{i=0}^{3}\sum_{j=0}^{3} |f_{ij}| \times \lfloor (i + j - 1)/2 \rfloor \qquad \text{Eq. 4-1}$$

This Hadamard transform is the same as the Hadamard transform used in intra 16x16 blocks, whose input are the DC components of the 16 4x4 integer transformed

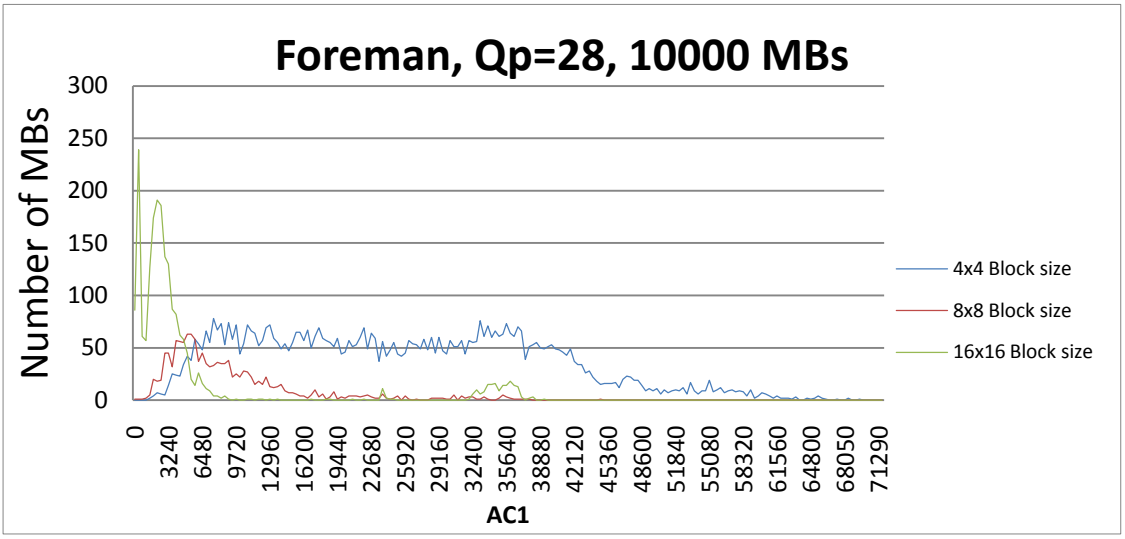
block. With this, we can distinguish the intra 4x4 block type with other block sizes. Fig. 4-4 shows the number of MBs for different block types and corresponding AC1 values. We can find that AC1 value for the 4x4 block type is tended to be small for all tested sequences and quantization parameters, since 4x4 block partition will be rougher and has lower low frequency components.



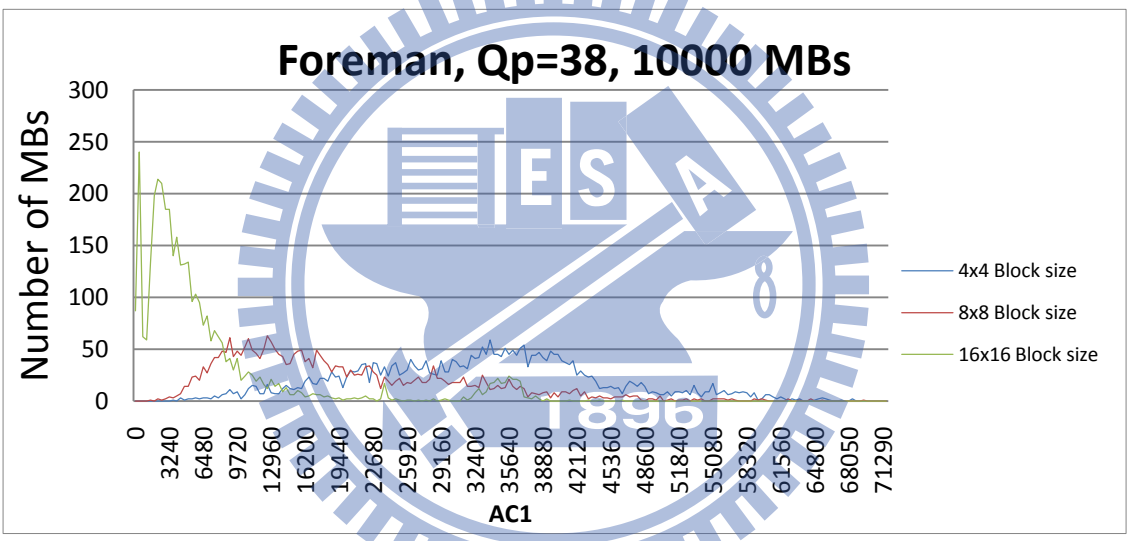
(a)



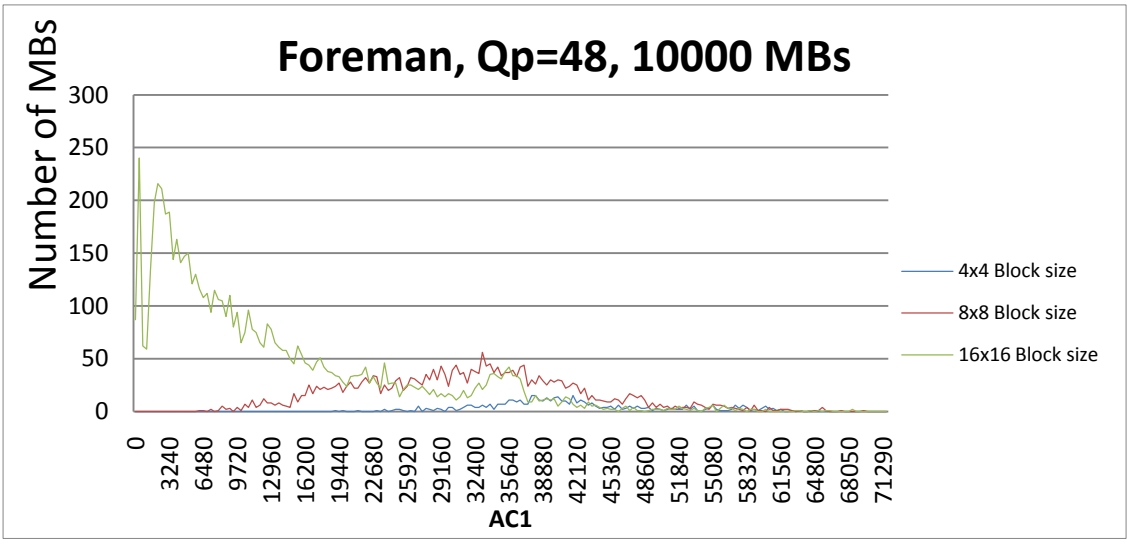
(b)


Foreman, Qp=28, 10000 MBs
Number of MBs
300
250
200
150
100
50
0
0
3240
6480
9720
12960
16200
19440
22680
25920
29160
32400
35640
38880
42120
45360
48600
51840
55080
58320
61560
64800
68050
71290
AC1
4x4 Block size
8x8 Block size
16x16 Block size

(c)


Foreman, Qp=38, 10000 MBs
Number of MBs
300
250
200
150
100
50
0
0
3240
6480
9720
12960
16200
19440
22680
25920
29160
32400
35640
38880
42120
45360
48600
51840
55080
58320
61560
64800
68050
71290
AC1
4x4 Block size
8x8 Block size
16x16 Block size

(d)


Foreman, Qp=48, 10000 MBs
Number of MBs
300
250
200
150
100
50
0
0
3240
6480
9720
12960
16200
19440
22680
25920
29160
32400
35640
38880
42120
45360
48600
51840
55080
58320
61560
64800
68050
71290
AC1
4x4 Block size
8x8 Block size
16x16 Block size

(e)

Fig. 4-4. The number of MBs for different block types and corresponding AC1 values

Thus, we can define the following condition:

$$\begin{cases} AC1 \geq TH1 & \text{choose 4x4 block size} \\ AC1 < TH1 & \text{choose 8x8 block size or 16x16 block size.} \end{cases} \quad \text{Eq. 4-2}$$

The threshold value, TH1, is highly dependent on Qp size, as shown in Fig. 4-4. Thus, we test foreman, moble, akiyo and weather sequences under Qp at 8, 18, 28, 38, 48 to find the proper TH1 values to minimize the RD performance loss. Fig. 4-5 shows the corresponding TH1 values, which can be fitted as

$$TH1 = 2571.4Qp^2 - 1228.6Qp + 1000. \quad \text{Eq. 4-3}$$

Thus, the threshold value, TH1, will adapt as the Qp is changed.

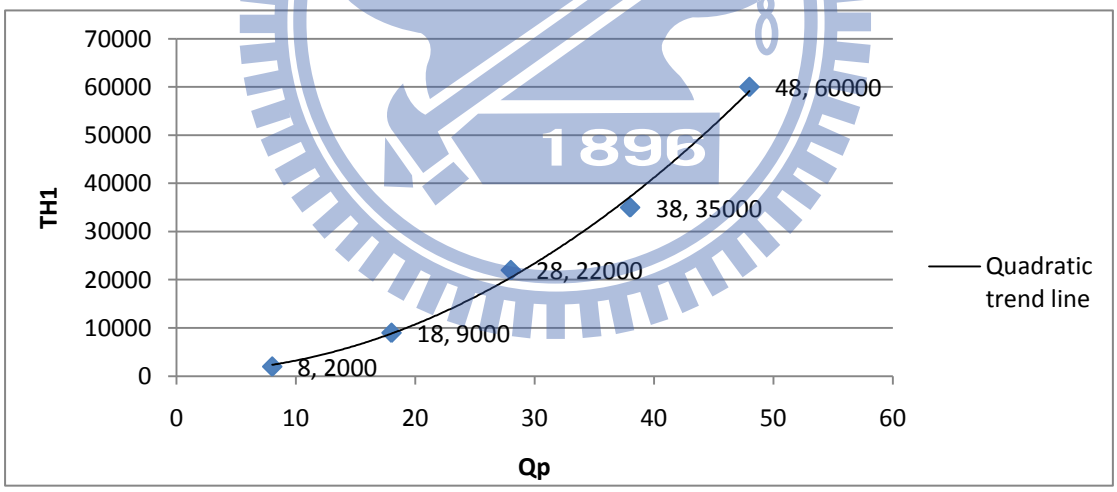


Fig. 4-5. Corresponding TH1 values with different Qps

4.3.2. Determination of 16x16 Block Size

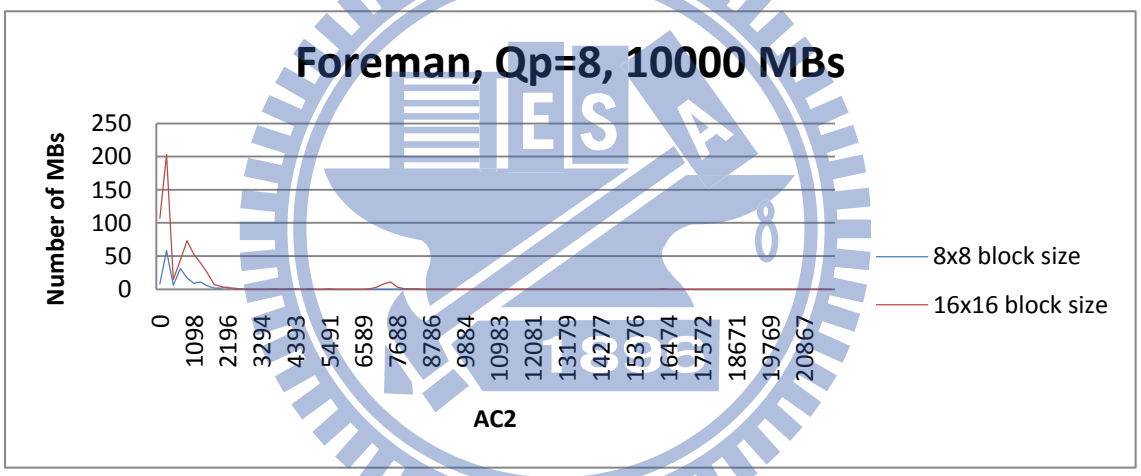
Above condition cannot distinguish the intra8x8 and intra 16x16 well. Thus, we propose the second smoothness test, AC2, by summing the first three AC components

of the 16 4x4 integer transformed blocks, which is defined as

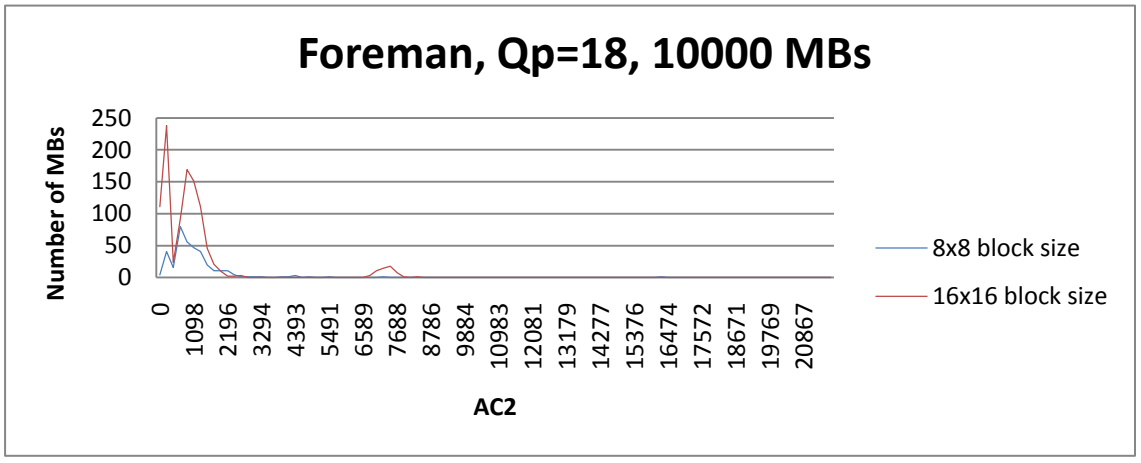
$$AC\ 2 = \sum_{sub\ \_MB\ =0}^{15} \sum_{k=0}^{2} \left| f_k \right| \qquad \text{Eq. 4-4}$$

Fig. 4-6 shows AC2 distribution for intra8x8 and intra16x16. We can find that the AC2 value of intra8x8 is tended to be larger since 8x8 block partition will be rougher and has higher high frequency component. Thus, we can define the following condition to distinguish intra 8x8 and intra 16x16

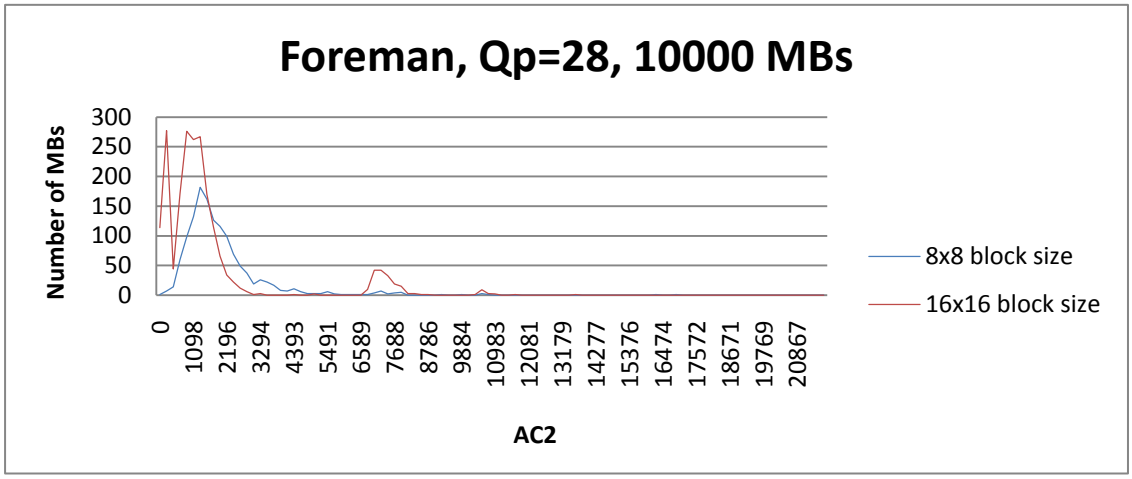
$$\begin{cases} \text{AC2} \geq \text{TH2} & \text{choose 8x8 block size} \\ \text{AC2} < TH2 & \text{choose 16x16 block size} \end{cases} \qquad \text{Eq. 4-5}$$



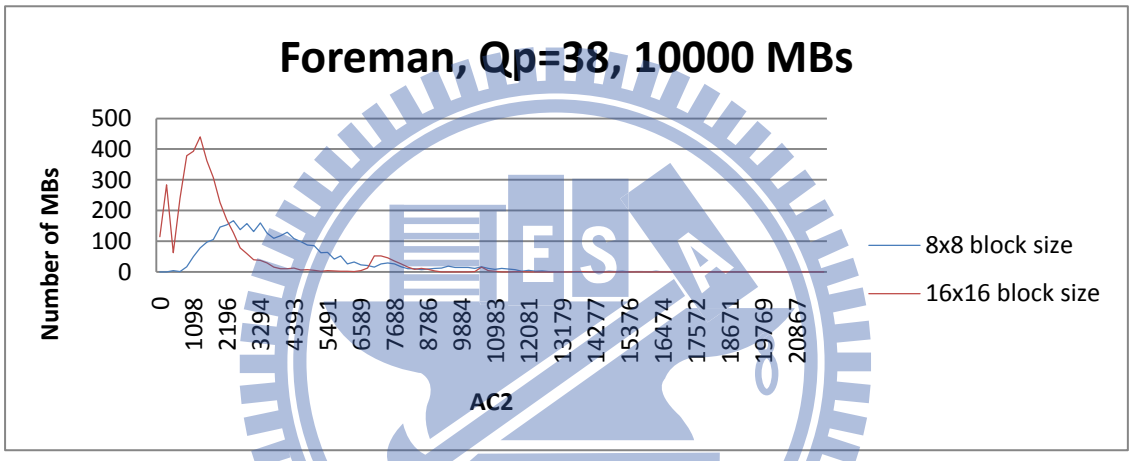
(a)



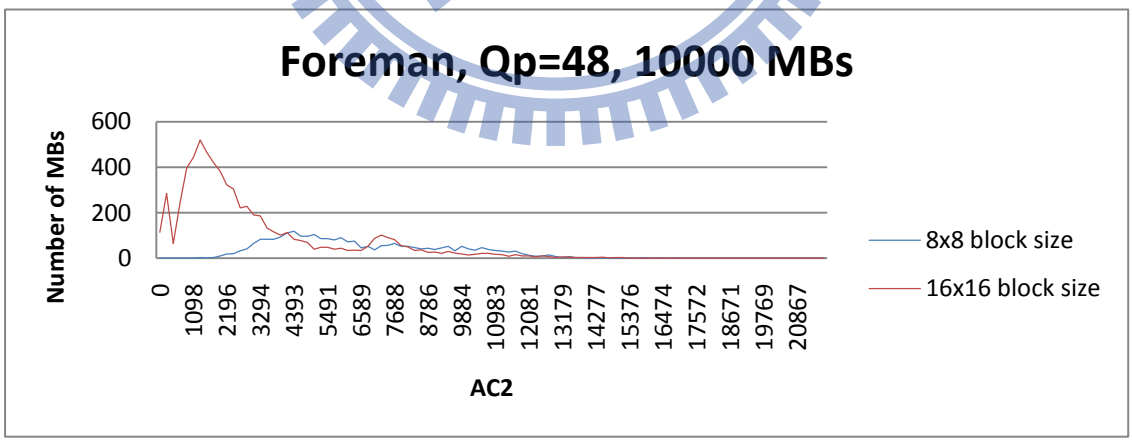
(b)



(c)



(d)



(e)

Fig. 4-6. The number of MBs for different block types and corresponding AC2 values

Similarly, the threshold value, TH2, is also highly dependent on the Qp values.

With the similar approach, we can find the relationship of TH2 and Qp as

$$TH2 = 228.57Qp^2 - 891.43Qp + 1220 \quad \text{Eq. 4-6}$$

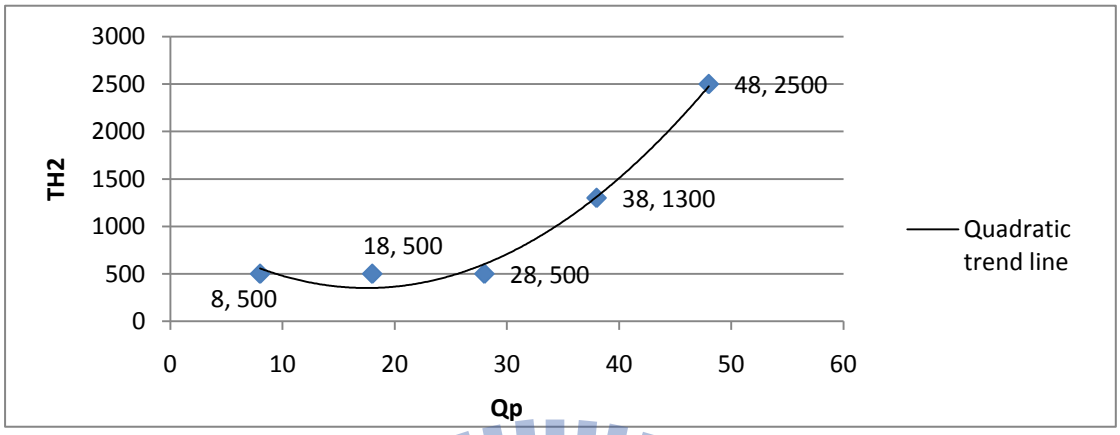


Fig. 4-7. Corresponding TH2 values with different Qps

4.4. Fast Intra Mode Prediction Algorithm

4.4.1. Intra 4x4 Mode Prediction, Intra 16x16 Mode Prediction and Intra chroma Mode Prediction

With fast block type selection, we will predict the possible modes in the selected block type. Our approach for 4x4 and 16x16 block is adapted from the transform domain fast intra prediction algorithm [25] with modification of enforced DC mode selection and weighted cost for most probability mode as that in JM [22]. However, [25] uses independent mode prediction for 4x4 and 8x8 blocks, which implies two hardware units for high throughput demands. Chroma prediction is also adapted from [25] with enforced plane mode selection.

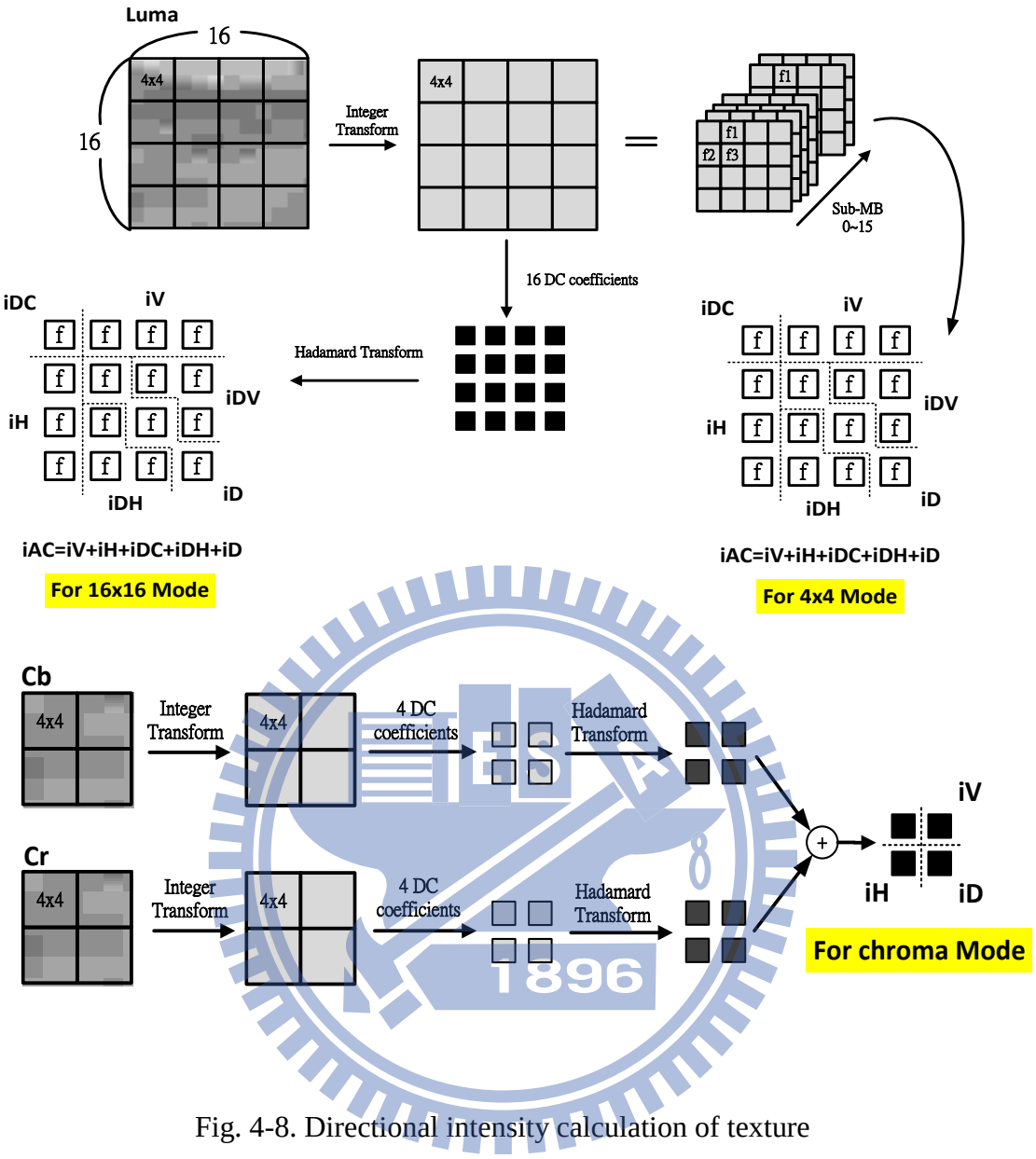


Fig. 4-8. Directional intensity calculation of texture

The original intra 4x4 mode prediction method [25] is:

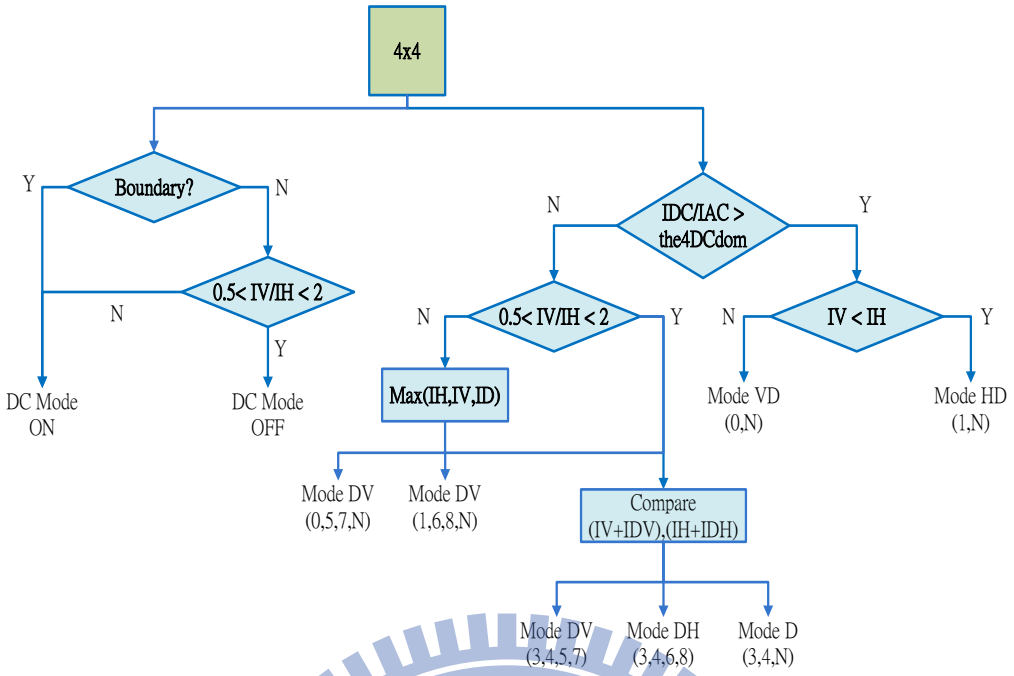


Fig. 4-9. Intra 4x4 mode decision of [25]

The modified intra 4x4 mode prediction method is:

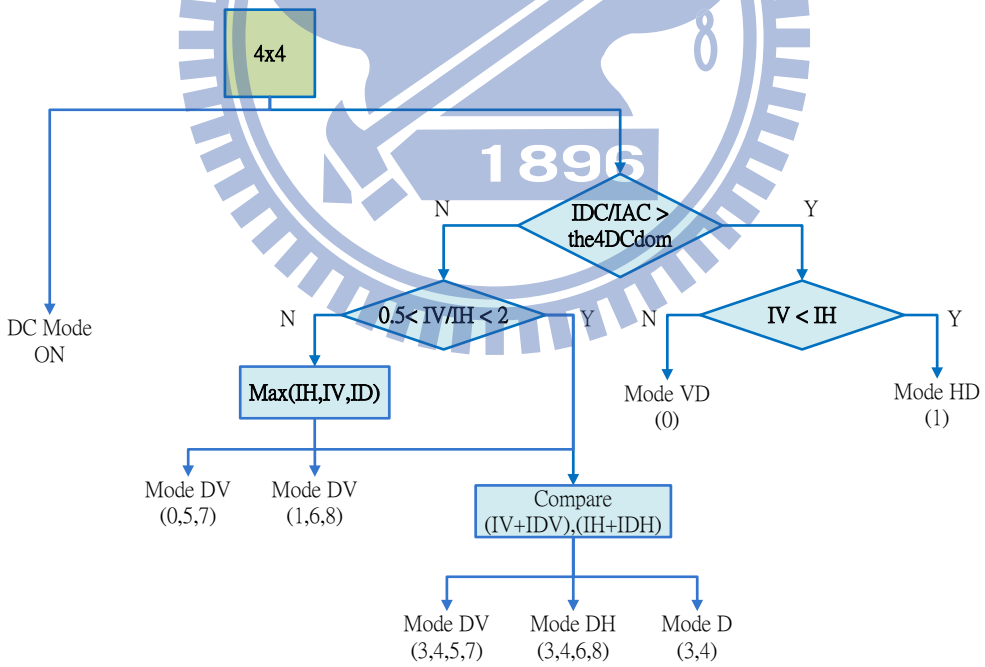


Fig. 4-10. Intra 4x4 mode decision of modified algorithm

The original intra 16x16 mode prediction method [25] is:

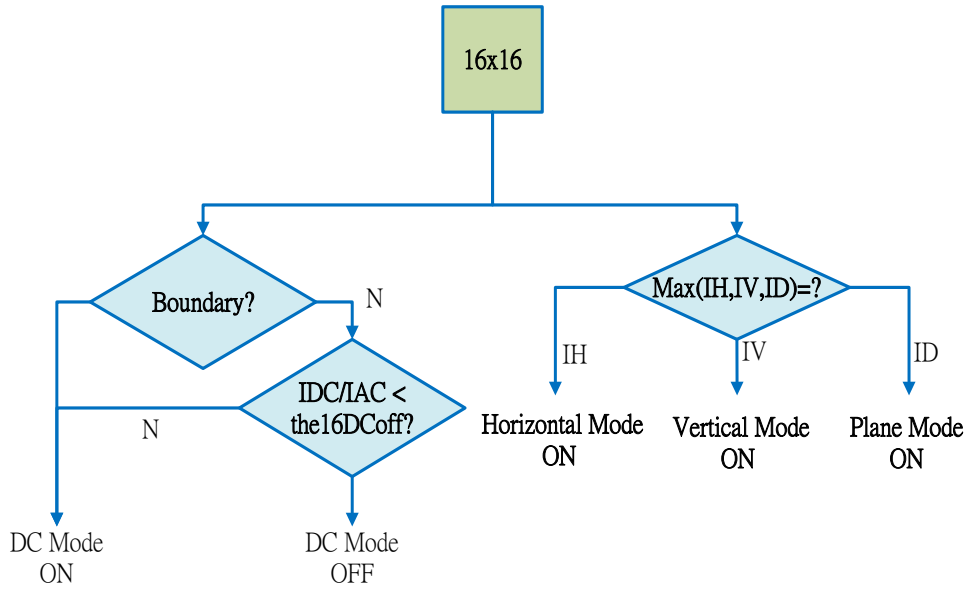


Fig. 4-11. Intra 16x16 Mode Decision of [25]

The modified intra 16x16 mode prediction method is:

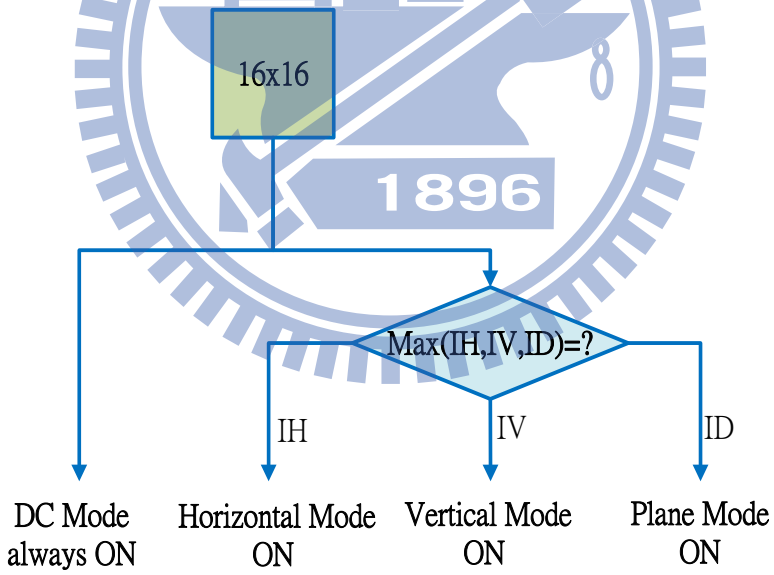


Fig. 4-12. Intra 16x16 mode decision of modified algorithm

The original intra chroma mode prediction method [25] is:

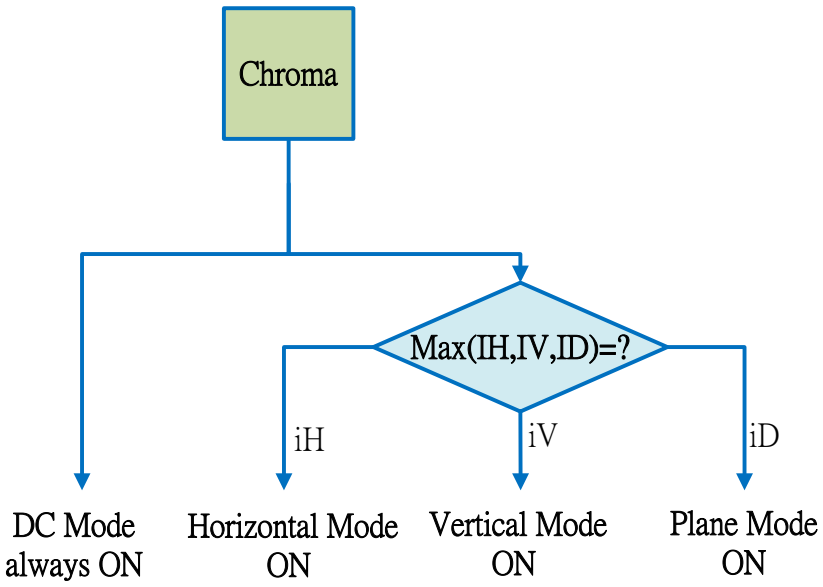


Fig. 4-13. Intra chroma mode decision of [25]

The modified intra chroma mode prediction method is:

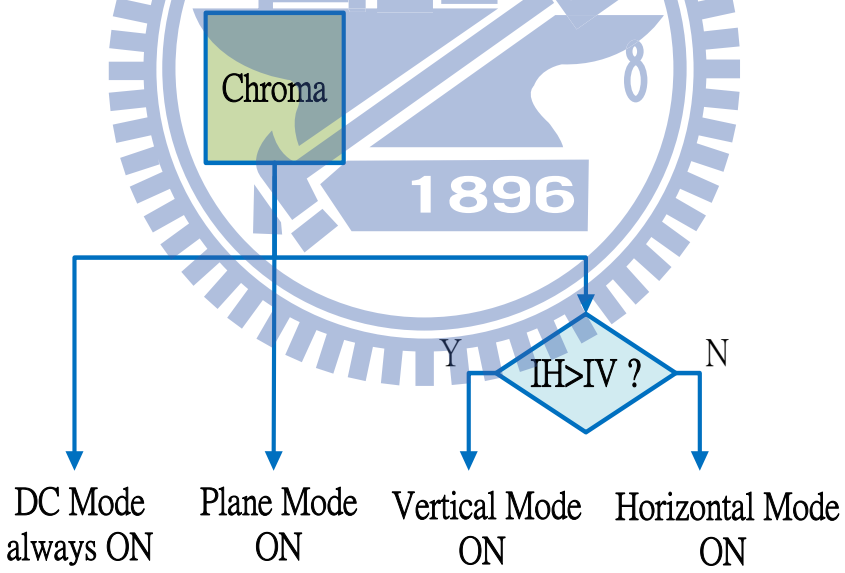


Fig. 4-14. Intra chroma mode decision of modified algorithm

4.4.2. Proposed Intra 8x8 Mode Prediction

Our concept for fast intra 8x8 is to reuse the information of 4x4 blocks by

merging the candidates of 4x4 to generate the ones of 8x8. With this approach, we can avoid extra computation and transform hardware to predict 8x8 candidates. Table 4-1 shows the correlation of 4x4 and 8x8 prediction modes for the same MB. We can find the correlation is very high for the prediction modes of 4x4 and 8x8. Most of 8x8 best modes will appear in the 4x4 candidate modes.

	Best mode	Best mode distribution (%)	Hit rate=8x8 best mode in the 4x4 candidate modes (%)
mode 8x8 (Full mode)	0	10.98	87.78
	1	13.09	93.79
	2	23.20	100.00
	3	3.97	86.96
	4	15.67	96.89
	5	7.15	83.03
	6	9.57	96.47
	7	4.93	79.64
	8	11.44	95.25

Table 4-1. The correlation of 4x4 and 8x8 prediction modes for "Foreman" sequence with 100 frames

Fig. 4-15 shows an example to generate 8x8 prediction mode candidates from 4x4 mode candidates. First, we sum up the number of 4x4 mode candidates, S_n, within the same 8x8 block. Then we can use the following rules to determine 8x8 mode candidates, where n is the mode number.

$$
\begin{aligned}
&\textit{For all } 8x8 \textit{ prediction mode } n = 0 \textit{ to } 8 \\
&\quad \text{if} \quad ((n < 2) \text{ and } (Sn > 0)) \quad \text{assign mode}_n \text{ to 8x8 mode candidates} \\
&\quad \text{else if}\big((n > 2) \text{ and } (Sn > 1)\big) \quad \text{assign mode}_n \text{ to 8x8 mode candidates} \\
&\quad \text{else} \quad (n = 2) \quad \text{assign DC mode to 8x8 mode candidates}
\end{aligned}
$$

Eq. 4-7

To further reduce the 8x8 mode candidates, we discard larger mode number till the total number of predictions is less than 5.

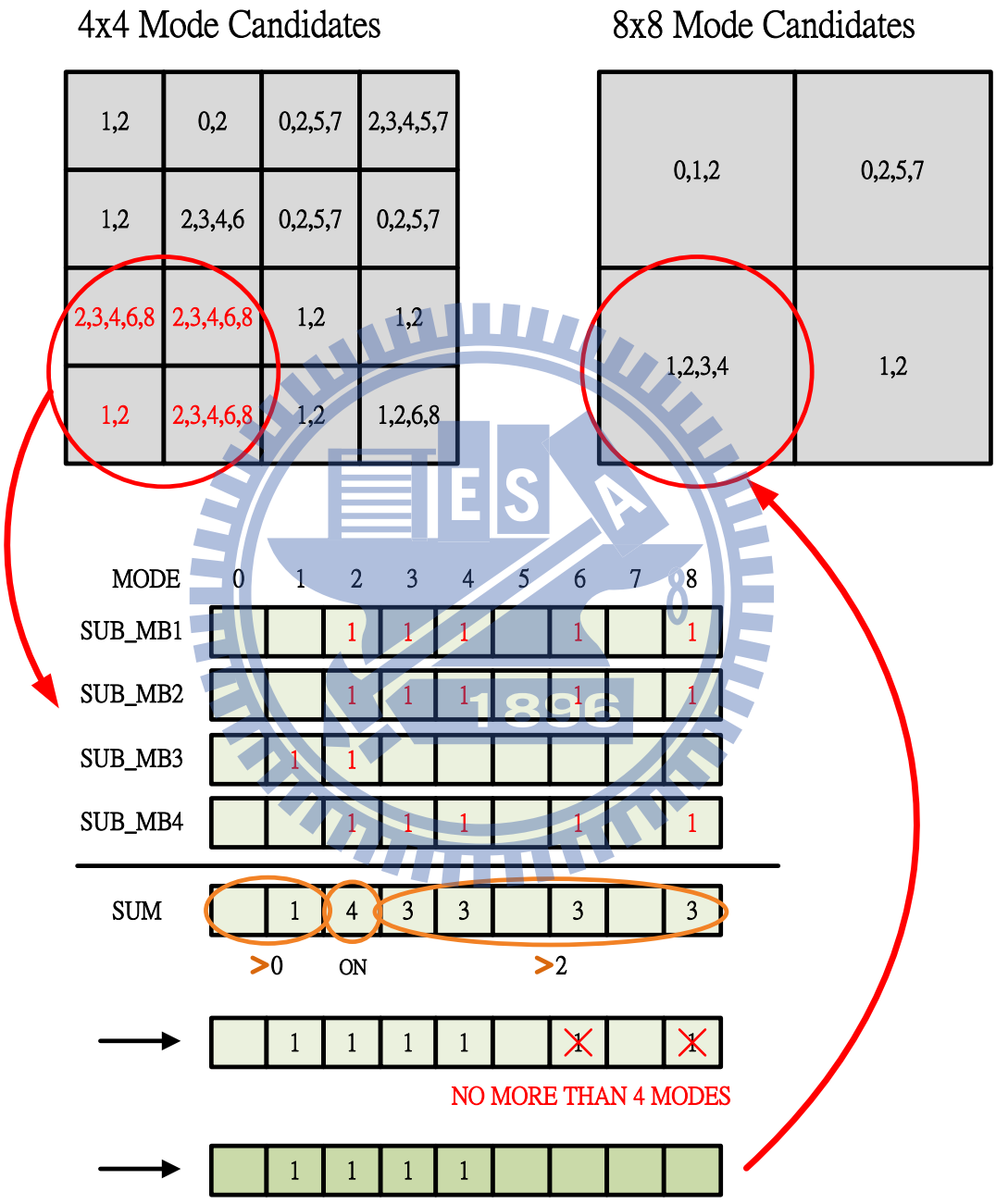


Fig. 4-15. 8x8 mode decision with merged 4x4 candidates

4.5. Simulation Results

In the following simulation, 300 I-frames in 16 sequences with CIF and 1080p resolutions are under tested. The comparison of bit-rate (BR) and PSNR between original result in JM 12.4 [22] (with the low-complexity mode) and proposed algorithm is listed in Table 4-2 and Table 4-3, and the average mode candidates for each sequence is summarized in Table 4-4.

Table 4-2. Rate-distortion comparison between JM 12.4 and proposed algorithm for 300 I-frames

		QP=8		QP=12		QP=16		QP=20	
		ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)
CIF	*Akiyo*	0.01	0.85	-0.03	0.94	0	2.41	0.02	-0.68
	Foreman	0	0.97	-0.04	0.94	-0.01	1.48	-0.05	1.66
	News	-0.02	0.31	-0.01	0.21	0.02	1.61	0.01	0.75
	Table	-0.01	1.47	-0.02	-0.80	0.07	-1.30	0.08	-1.24
	Mobile	0	1.01	0	1.33	-0.01	1.46	-0.02	0.85
	Silent	0	0.70	-0.01	0.84	0.01	1.05	0.01	1.27
	Mother_daughter	0.02	0.66	-0.01	0.81	0.03	2.51	0.05	0.98
	Stefan	0	1.29	-0.01	1.84	0	1.45	0.01	1.40
	Coastguard	0.01	1.19	0.01	0.30	0.06	-0.24	0.1	-1.17
Average		**0.00**	**0.94**	**-0.01**	**0.71**	**0.02**	**1.16**	**0.02**	**0.42**
1080p	*Tractor*	0.02	0.85	0	0.74	0.02	0.88	0.04	-0.32
	Sunflower	0.04	0.97	0	1.07	0.02	1.32	0.03	0.05
	Station2	0.01	0.31	0	0.73	0.07	1.09	0.06	-0.26
	Pedestrian_area	-0.01	1.47	-0.02	0.46	0.04	1.72	0.04	0.85
	Rush_hour	-0.03	1.01	-0.02	0.62	0.04	2.53	0.04	0.74
	*Riverbed**	0.02	0.70	0.02	0.73	0.03	0.79	0.04	-0.59
	*Blue_sky***	0.01	0.66	-0.02	0.36	0.02	0.90	0.01	0.50
Average		**0.01**	**0.85**	**-0.01**	**0.67**	**0.03**	**1.32**	**0.04**	**0.14**

(a)

		QP=24		QP=28		QP=32		QP=36	
		ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)	ΔPSNR(dB)	ΔBR(%)
CIF	*Akiyo*	0.06	-1.36	-0.02	-3.67	-0.06	-6.74	-0.08	-8.91
	Foreman	-0.04	1.90	-0.07	0.29	-0.13	-2.08	-0.1	-3.16
	News	0.01	1.06	0	-0.47	-0.06	-2.28	-0.06	-3.80
	Table	0.12	0.50	0.07	1.67	0.01	-0.43	0	-4.08
	Mobile	-0.02	1.11	-0.05	1.09	-0.11	0.51	-0.11	0.47
	Silent	0.03	2.32	-0.03	1.62	-0.09	-0.70	-0.07	-3.39
	Mother_daughter	0.08	1.59	0.03	0.96	-0.1	-3.32	-0.05	-6.84
	Stefan	0.01	1.53	0.01	1.38	0.01	0.93	0.05	0.61
	Coastguard	0.19	-0.70	0.16	-0.76	0.07	-0.99	0.09	-1.03
Average		**0.05**	**0.88**	**0.01**	**0.23**	**-0.05**	**-1.68**	**-0.04**	**-3.35**
1080p	*Tractor*	0.15	-1.29	0.16	-2.79	0.08	-3.55	0.06	-4.65
	Sunflower	0.05	-0.87	0.01	-2.03	-0.05	-2.33	-0.09	-1.87
	Station2	0.1	0.58	0.11	-1.17	0.08	-5.11	0.1	-7.82
	Pedestrian_area	0.06	0.24	0.06	-3.39	0.01	-6.35	-0.01	-6.41
	Rush_hour	0.04	-1.59	0.03	-1.81	-0.01	-3.22	-0.05	-2.91
	*Riverbed**	0.13	-1.40	0.16	-3.11	0.11	-4.22	0.14	-5.08
	*Blue_sky***	-0.01	0.37	-0.03	-0.60	-0.06	-1.77	-0.05	-2.99
Average		**0.07**	**-0.57**	**0.07**	**-2.13**	**0.02**	**-3.79**	**0.01**	**-4.53**

(b)

Note:

Entropy coding method: CABAC

Measurement unit: ΔPSNR (dB), ΔBR (%)

**:250 I-frames*

***:217 I-frames*

Table 4-3. Delta BD-RATE and BD-PSNR between JM12.4 and proposed algorithm

	sequence	Qp = 20,24,28,32	
		BD-PSNR	BD-RATE
CIF	Akiyo	0.19	-3.09
	Foreman	-0.12	1.86
	News	0.00	0.01
	Table	0.03	-0.53
	Mobile	-0.17	1.37
	Silent	-0.11	1.81
	Mother_daughter	-0.01	0.10
	Stefan	-0.13	1.28
	Coastguard	0.21	-2.70
Average		-0.01	0.01
1080p	Tractor	0.22	-4.89
	Sunflower	0.08	-1.97
	Station2	0.13	-3.32
	Pedestrian_area	0.11	-3.53
	Rush_hour	0.07	-3.14
	Riverbed*	0.22	-5.11
	Blue_sky**	0.00	0.05
Average		0.12	-3.13

Note:

Entropy coding method: CABAC

Measurement unit: ΔPSNR (dB), ΔBR (%)

**:250 I-frames*

***:217 I-frames*

Table 4-4. Average mode candidates in encoding

		JM 12.4 (Full search)				Proposed				Proposed			
		Qps				Qp=8				Qp=12			
		4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}
CIF	Akiyo	8.86	8.72	3.80	21.38	2.76	0.31	0.34	3.41	2.30	0.99	0.18	3.47
	Foreman	8.86	8.72	3.80	21.38	3.67	0.10	0.12	3.89	3.35	0.42	0.11	3.87
	News	8.86	8.72	3.80	21.38	3.18	0.36	0.23	3.77	2.70	1.08	0.05	3.83
	Table	8.86	8.72	3.80	21.38	3.84	0.31	0.01	4.16	2.51	1.42	0.00	3.93
	Mobile	8.86	8.72	3.80	21.38	4.13	0.02	0.02	4.18	4.10	0.06	0.01	4.18
	Silent	8.86	8.72	3.80	21.38	4.11	0.06	0.04	4.20	3.89	0.26	0.03	4.18
	Mother_daughter	8.86	8.72	3.80	21.38	2.86	0.09	0.45	3.40	2.69	0.41	0.33	3.43
	Stefan	8.86	8.72	3.80	21.38	3.37	0.19	0.19	3.75	3.30	0.46	0.05	3.81
	Coastguard	8.86	8.72	3.80	21.38	4.16	0.06	0.00	4.22	3.73	0.45	0.00	4.19
Average		**8.86**	**8.72**	**3.80**	**21.38**	**3.56**	**0.17**	**0.16**	**3.89**	**3.17**	**0.62**	**0.08**	**3.87**
Savings (%)						**59.77**	**98.1**	**95.91**	**81.83**	**64.17**	**92.94**	**97.79**	**81.88**
1080p	Tractor	8.97	8.94	3.95	21.86	3.86	0.06	0.13	4.05	3.54	0.39	0.10	4.03
	Sunflower	8.97	8.94	3.95	21.86	3.97	0.11	0.10	4.18	3.32	0.77	0.04	4.13
	Station2	8.97	8.94	3.95	21.86	3.66	0.14	0.26	4.06	3.05	0.88	0.10	4.03
	Pedestrian_area	8.97	8.94	3.95	21.86	2.99	0.58	0.23	3.80	2.09	1.46	0.17	3.72
	Rush_hour	8.97	8.94	3.95	21.86	2.56	1.12	0.17	3.85	1.82	2.00	0.02	3.84
	Riverbed*	8.97	8.94	3.95	21.86	4.35	0.01	0.00	4.36	4.12	0.21	0.00	4.33
	Blue_sky**	8.97	8.94	3.95	21.86	2.78	0.13	0.65	3.56	2.53	0.60	0.47	3.60
Average		**8.97**	**8.94**	**3.95**	**21.86**	**3.45**	**0.31**	**0.22**	**3.98**	**2.92**	**0.90**	**0.13**	**3.96**
Savings (%)						**61.49**	**96.55**	**94.47**	**81.79**	**67.39**	**89.9**	**96.76**	**81.90**

(a)

		Proposed				Proposed				Proposed			
		Qp=16				Qp=20				Qp=24			
		4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}
CIF	Akiyo	1.75	1.69	0.05	3.49	1.29	2.10	0.06	3.45	1.03	2.07	0.24	3.35
	Foreman	2.97	0.77	0.09	3.83	2.56	1.12	0.10	3.77	2.05	1.52	0.11	3.69
	News	2.38	1.42	0.01	3.82	1.91	1.83	0.02	3.76	1.55	2.02	0.09	3.66
	Table	1.77	2.02	0.00	3.80	1.48	2.27	0.00	3.75	1.27	2.45	0.00	3.72
	Mobile	4.01	0.15	0.01	4.16	3.79	0.32	0.01	4.13	3.51	0.55	0.02	4.08
	Silent	3.42	0.68	0.01	4.11	2.68	1.31	0.01	3.99	1.96	1.87	0.03	3.87
	Mother_daughter	2.24	1.02	0.17	3.44	1.55	1.59	0.20	3.34	0.95	1.88	0.37	3.20
	Stefan	3.17	0.64	0.01	3.82	2.95	0.83	0.01	3.80	2.74	0.95	0.07	3.75
	Coastguard	3.13	1.01	0.00	4.14	2.35	1.70	0.00	4.05	1.71	2.27	0.00	3.98
Average		**2.76**	**1.04**	**0.04**	**3.85**	**2.28**	**1.45**	**0.05**	**3.78**	**1.86**	**1.73**	**0.10**	**3.70**
Savings (%)		**68.84**	**88.03**	**98.94**	**82.02**	**74.22**	**83.35**	**98.78**	**82.31**	**78.98**	**80.14**	**97.25**	**82.70**
1080p	Tractor	2.83	1.06	0.09	3.97	1.79	1.99	0.09	3.88	0.91	2.76	0.11	3.78
	Sunflower	2.20	1.75	0.02	3.97	1.31	2.50	0.02	3.84	0.78	2.92	0.05	3.74
	Station2	1.97	1.89	0.03	3.88	1.06	2.66	0.04	3.75	0.51	2.98	0.14	3.63
	Pedestrian_area	1.32	2.17	0.15	3.64	0.78	2.64	0.15	3.56	0.41	2.91	0.17	3.49
	Rush_hour	1.19	2.56	0.01	3.76	0.70	2.98	0.01	3.70	0.39	3.23	0.03	3.65
	Riverbed*	3.24	0.99	0.00	4.23	1.95	2.12	0.00	4.07	0.99	2.97	0.00	3.96
	Blue_sky**	2.27	1.05	0.31	3.63	2.03	1.22	0.34	3.58	2.03	1.22	0.34	3.58
Average		**2.14**	**1.64**	**0.09**	**3.87**	**1.37**	**2.30**	**0.09**	**3.77**	**0.86**	**2.71**	**0.12**	**3.69**
Savings (%)		**76.09**	**81.66**	**97.82**	**82.30**	**84.67**	**74.24**	**97.67**	**82.76**	**90.4**	**69.64**	**96.97**	**83.10**

(b)

		Proposed				Proposed				Proposed			
		Qp=28				Qp=32				Qp=36			
		4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}	4×4	8×8	16×16	AN_{4x4}
CIF	Akiyo	0.77	2.08	0.37	3.22	0.47	2.08	0.54	3.09	0.26	1.92	0.75	2.93
	Foreman	1.55	1.92	0.13	3.60	1.09	2.25	0.17	3.51	0.69	2.46	0.25	3.41
	News	1.27	1.96	0.28	3.51	0.98	1.87	0.48	3.33	0.70	1.88	0.61	3.20
	Table	1.04	2.62	0.02	3.68	0.82	2.69	0.11	3.62	0.60	2.70	0.23	3.53
	Mobile	3.07	0.92	0.02	4.01	2.47	1.40	0.03	3.90	1.83	1.92	0.04	3.79
	Silent	1.34	2.36	0.05	3.76	0.76	2.84	0.06	3.66	0.38	3.10	0.10	3.58
	Mother_daughter	0.48	2.17	0.47	3.12	0.20	2.34	0.53	3.06	0.06	2.38	0.58	3.01
	Stefan	2.40	1.03	0.23	3.66	1.98	1.28	0.32	3.58	1.47	1.67	0.35	3.49
	Coastguard	1.22	2.70	0.00	3.92	0.82	3.05	0.00	3.87	0.45	3.37	0.01	3.83
Average		**1.46**	**1.97**	**0.17**	**3.61**	**1.07**	**2.20**	**0.25**	**3.51**	**0.71**	**2.38**	**0.32**	**3.42**
Savings (%)		**83.51**	**77.37**	**95.42**	**83.13**	**87.96**	**74.77**	**93.48**	**83.56**	**91.93**	**72.71**	**91.47**	**84.01**
1080p	Tractor	0.44	3.11	0.15	3.70	0.22	3.18	0.23	3.63	0.11	3.07	0.34	3.52
	Sunflower	0.48	3.00	0.16	3.63	0.30	2.75	0.38	3.43	0.19	2.23	0.71	3.13
	Station2	0.22	3.03	0.28	3.53	0.07	3.08	0.33	3.48	0.01	2.92	0.44	3.38
	Pedestrian_area	0.20	2.95	0.26	3.41	0.09	2.67	0.48	3.24	0.04	2.19	0.77	3.00
	Rush_hour	0.21	2.99	0.27	3.47	0.10	2.10	0.81	3.00	0.04	1.52	1.15	2.71
	Riverbed*	0.46	3.45	0.00	3.91	0.18	3.69	0.01	3.87	0.05	3.72	0.05	3.82
	Blue_sky**	1.79	1.13	0.54	3.46	1.56	1.10	0.68	3.34	1.33	1.15	0.77	3.25
Average		**0.54**	**2.81**	**0.24**	**3.59**	**0.36**	**2.65**	**0.42**	**3.43**	**0.25**	**2.40**	**0.61**	**3.26**
Savings (%)		**93.96**	**68.55**	**94.04**	**83.59**	**95.98**	**70.32**	**89.48**	**84.31**	**97.18**	**73.13**	**84.69**	**85.09**

(c)

Note:

**:250 I-frames*

***:217 I-frames*

From the above results, the BD-RATE is increased by 0.01% for CIF sequences and decreased by 3.13% for 1080p sequences in average, and the BD-PSNR is decreased by 0.01dB for CIF sequences and increased by 0.12dB for 1080p sequences. In most cases, the performance of the proposed algorithm pretty approaches or the performance of full search algorithm in JM 12.4. And in high Qp points, especially

with 1080p resolution, the performance of the proposed algorithm is even better. The main reason is that we use Integer transform instead of Hadamard transform for SATD-Cost computation.

The comparison result with [23], [25] and proposed algorithm is listed in Table 4-5 with four different QP values (from 8 to 32). In high Qp points, the proposed algorithm outperforms.

Table 4-5. Comparison between [23]*, [25]** and our proposed algorithm*** for ΔPSNR and ΔBR

		QP = 8			QP = 16			QP = 24			QP = 32		
		[23]	**[25]**	**Proposed**	**[23]**	**[25]**	**Proposed**	**[23]**	**[25]**	**Proposed**	**[23]**	**[25]**	**Proposed**
Akiyo	**ΔPSNR**	-0.16	-0.02	-0.01	-0.07	-0.02	-0.05	-0.04	-0.04	-0.02	-0.03	-0.18	-0.15
(QCIF)	**ΔBR**	0.27	1.42	0.94	0.76	1.64	1.67	0.75	1.50	0.02	1.02	1.19	-4.50
Foreman	**ΔPSNR**	-0.15	0.02	0.00	-0.07	0.03	-0.01	-0.04	0.06	0.00	-0.03	0.03	-0.08
(QCIF)	**ΔBR**	0.15	0.26	0.95	0.43	0.94	0.88	0.74	1.65	1.41	0.88	2.53	-0.04
Mobile	**ΔPSNR**	-0.20	0.03	0.00	-0.09	0.05	-0.01	-0.07	0.11	-0.02	-0.08	0.10	-0.11
(CIF)	**ΔBR**	0.36	0.51	1.01	0.58	0.77	1.46	0.74	1.21	1.11	0.89	2.14	0.51
Stefan	**ΔPSNR**	-0.18	0.02	0.00	-0.07	0.03	0.00	-0.07	0.09	0.01	-0.08	0.05	0.01
(CIF)	**ΔBR**	0.30	0.55	1.29	0.61	1.03	1.45	0.81	1.56	1.53	1.10	0.99	0.93
Shields	**ΔPSNR**	-0.22	0.05	-0.02	-0.10	0.06	0.01	-0.06	-0.03	0.03	-0.06	-0.13	-0.02
(720p)	**ΔBR**	0.29	0.70	0.66	0.50	1.13	0.60	0.90	1.93	1.63	1.23	3.93	-4.11
Stockholm	**ΔPSNR**	-0.23	0.06	-0.08	-0.12	0.07	0.01	-0.05	-0.06	0.06	-0.04	-0.20	-0.05
(720p)	**ΔBR**	0.20	0.80	0.61	0.36	1.38	0.86	0.95	1.25	2.45	1.37	4.15	-0.47
Tractor	**ΔPSNR**	-0.16	0.01	0.02	-0.06	0.04	0.02	-0.04	0.02	0.15	-0.02	-0.41	0.08
(1080p)	**ΔBR**	0.04	0.32	0.80	0.54	1.03	0.88	1.13	1.08	-1.29	1.31	3.01	-3.55
Average	**ΔPSNR**	**-0.19**	**0.03**	**-0.01**	**-0.08**	**0.04**	**0.00**	**-0.05**	**0.02**	**0.03**	**-0.05**	**-0.11**	**-0.05**
	ΔBR	**0.23**	**0.65**	**0.90**	**0.54**	**1.13**	**1.11**	**0.86**	**1.45**	**0.98**	**1.11**	**2.56**	**-1.60**

Note: measurement unit: ΔPSNR (dB), ΔBR (%)

**: implemented in JM 12.0, intra 8x8 mode is not included, 100 I-frames, CAVLC, and compared with JM 12.0 in RDO mode*

***: implemented in self-developed program, intra 8x8 mode is included, 100 I-frames, CAVLC, and compare with JM 12.4 in low complexity mode*

****: implemented in JM 12.4, intra 8x8 mode is included, 300 I-frames, CABAC, and compare with JM 12.4 in low complexity mode*

On the other hand, the comparison of average mode candidates between [23], [24] and proposed algorithm is summarized in Table 4-6. The "AN_{4x4}" means the equivalent average mode computation per 4x4 block, which can be defined as

$$AN4x4 = \text{mode cadidate number for 4x4} + \text{mode cadidate number for 8x8} + \text{mode cadidate number for 16x16} \quad \text{Eq. 4-8}$$

For example, for full intra prediction search, its $AN_{4x4} = 9 + 9 + 4 = 22$. Due to the fast intra block size algorithm, we can reduce more than 1/2 prediction modes. Besides, by the help of the 8x8 mode decision with merged 4x4 candidates method, we select less than 5 modes for intra 8x8. Therefore, the proposed algorithm further reduces about 66 % of mode candidates corresponding to [24].

Table 4-6. Comparison between [23], [25] and our proposed algorithm for average mode candidates

		[23]		[25]				Proposed				Proposed			
		Qps		Qps				Qp=8				Qp=12			
		4x4	16x16	4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4
QCIF	Coastguard	4.11	2	3.87	3.62	1.69	9.18	3.98	0.11	0.00	4.09	3.53	0.54	0.00	4.07
	Container	4.33	2	3.52	3.75	1.73	9	3.14	0.47	0.07	3.68	2.53	1.03	0.05	3.61
	Foreman	4.67	2	3.74	4.02	1.83	9.59	3.82	0.03	0.03	3.92	3.72	0.14	0.06	3.92
	News	4.4	2	3.82	3.81	1.55	9.18	3.69	0.17	0.10	3.95	3.58	0.28	0.08	3.94
	Silent	4.66	2	4.1	4.05	1.8	9.95	4.18	0.01	0.00	4.20	4.13	0.06	0.00	4.19
	Average	**4.43**	**2**	**3.81**	**3.85**	**1.72**	**9.38**	**3.76**	**0.16**	**0.04**	**3.97**	**3.50**	**0.41**	**0.04**	**3.94**
CIF	Mobile	4.43	2	3.7	3.83	1.62	9.15	4.13	0.02	0.02	4.18	4.10	0.06	0.01	4.18
	Paris	4.33	2	3.85	3.8	1.66	9.31	3.99	0.07	0.02	4.09	3.57	0.47	0.01	4.05
	Stefan	4.57	2	3.48	3.86	1.83	9.17	3.37	0.19	0.19	3.75	3.30	0.46	0.05	3.81
	Tempete	4.37	2	3.83	3.7	1.41	8.94	3.85	0.04	0.18	4.06	3.79	0.14	0.15	4.07
	Average	**4.43**	**2**	**3.72**	**3.8**	**1.63**	**9.15**	**3.84**	**0.08**	**0.10**	**4.02**	**3.69**	**0.28**	**0.05**	**4.03**
Maximum Number		6	2	5	5	2	**12**	**5**	**4**	**2**	**5**	**5**	**4**	**2**	**5**

(a)

		Proposed				Proposed				Proposed			
		Qp=16				Qp=20				Qp=24			
		4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4
QCIF	Coastguard	3.05	0.98	0.00	4.03	2.54	1.44	0.00	3.98	1.93	1.97	0.00	3.90
	Container	2.19	1.37	0.03	3.59	2.07	1.47	0.03	3.57	1.78	1.73	0.05	3.55
	Foreman	3.52	0.33	0.05	3.90	3.32	0.51	0.05	3.87	2.91	0.83	0.06	3.80
	News	3.35	0.48	0.08	3.91	2.76	0.99	0.08	3.83	2.36	1.34	0.08	3.78
	Silent	3.86	0.29	0.00	4.15	3.36	0.70	0.00	4.06	2.74	1.22	0.00	3.95
	Average	**3.19**	**0.69**	**0.03**	**3.92**	**2.81**	**1.02**	**0.03**	**3.86**	**2.34**	**1.42**	**0.04**	**3.80**
CIF	Mobile	4.01	0.15	0.01	4.16	3.79	0.32	0.01	4.13	3.51	0.55	0.02	4.08
	Paris	3.26	0.75	0.00	4.01	2.93	1.04	0.00	3.98	2.59	1.33	0.01	3.93
	Stefan	3.17	0.64	0.01	3.82	2.95	0.83	0.01	3.80	2.74	0.95	0.07	3.75
	Tempete	3.70	0.27	0.10	4.08	3.51	0.42	0.11	4.05	3.13	0.69	0.16	3.98
	Average	**3.53**	**0.45**	**0.03**	**4.02**	**3.30**	**0.66**	**0.03**	**3.99**	**2.99**	**0.88**	**0.06**	**3.93**
Maximum Number		**5**	**4**	**2**	**5**	**5**	**4**	**2**	**5**	**5**	**4**	**2**	**5**

(b)

		Proposed				Proposed				Proposed			
		Qp=28				Qp=32				Qp=36			
		4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4	4x4	8x8	16x16	AN4x4
QCIF	Coastguard	1.31	2.50	0.00	3.82	0.74	2.99	0.01	3.74	0.38	3.26	0.04	3.68
	Container	1.39	2.03	0.09	3.50	1.13	2.01	0.25	3.39	0.80	2.02	0.41	3.23
	Foreman	2.31	1.32	0.07	3.70	1.65	1.85	0.09	3.59	1.01	2.34	0.12	3.47
	News	1.84	1.77	0.10	3.72	1.41	2.05	0.15	3.61	1.03	2.31	0.19	3.53
	Silent	1.74	2.02	0.00	3.76	0.92	2.70	0.01	3.62	0.54	3.01	0.01	3.56
	Average	**1.72**	**1.93**	**0.05**	**3.70**	**1.17**	**2.32**	**0.10**	**3.59**	**0.75**	**2.59**	**0.15**	**3.49**
CIF	Mobile	3.07	0.92	0.02	4.01	2.47	1.40	0.03	3.90	1.83	1.92	0.04	3.79
	Paris	2.14	1.67	0.03	3.85	1.54	2.13	0.08	3.74	1.02	2.45	0.15	3.62
	Stefan	2.40	1.03	0.23	3.66	1.98	1.28	0.32	3.58	1.47	1.67	0.35	3.49
	Tempete	2.46	1.22	0.19	3.87	1.59	1.94	0.21	3.74	0.80	2.58	0.23	3.60
	Average	**2.52**	**1.21**	**0.12**	**3.85**	**1.90**	**1.69**	**0.16**	**3.74**	**1.28**	**2.16**	**0.19**	**3.63**
Maximum Number		**5**	**4**	**2**	**5**	**5**	**4**	**2**	**5**	**5**	**4**	**2**	**5**

(c)

4.6. Summary

By the proposed fast intra algorithms, during intra prediction, more than 80% intra luma modes (4x4, 8x8, and 16x16) can be skipped with similar quality (average BD-PSNR difference: -0.01 dB for CIF, +0.12dB for 1080p, average BD-Rate difference: +0.01% for CIF, -3.13% for 1080p).

5. Hardware Design for SVC Intra Encoder

This chapter will go through the demand of our SVC with fast algorithm and final implementation of hardware design. We will start from system analysis (memory usage and throughput). Next is to introduce the architecture of hardware, corresponding encoding schedule and the specific hardware module design. Finally, we will compare the results with previous studies.

5.1. System Analysis and its Solutions

With spec defined intra prediction procedure, it takes more resources to support strong dependencies between sub-MBs.

For example, our target spec of hardware design is 3 quality layers, 3 spatial layers (CIF, SD 480p, and HD 1080p), and up to 60 frames per second.

For hardware, the processing number of MB/per second is:

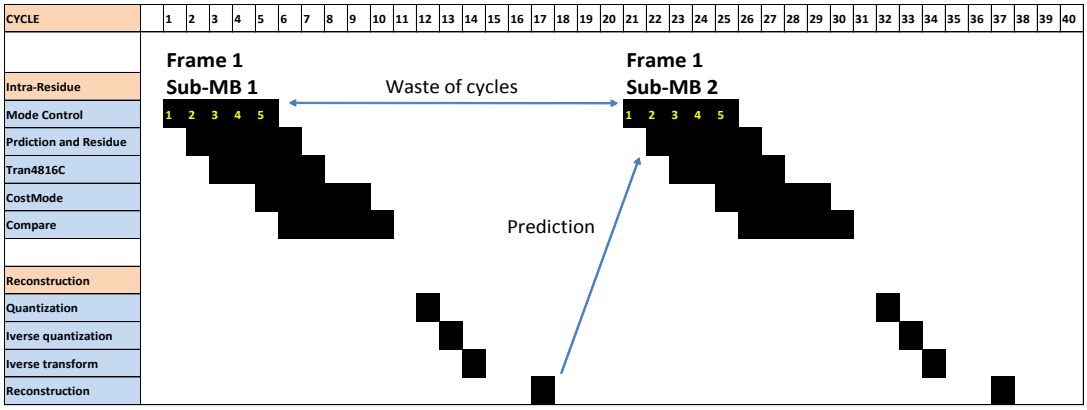
*(CIF_MB_ Number + SD_MB_ Number + HD_MB_ Number) * Frame_Rate*

*= (396 + 1,350 + 8,160)*60=594,360 MBs*

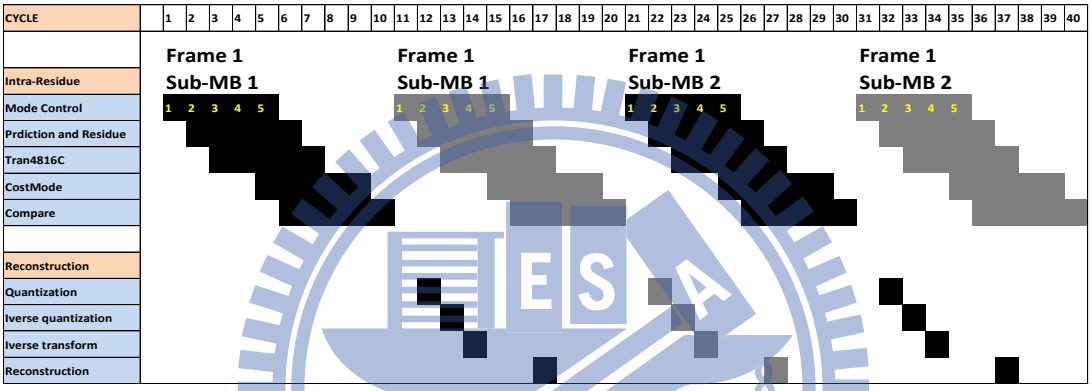
With the limitation of 135MHz working frequency, the usable cycles to proceed one MB are:

(Frequency / MB_Number)

= (135M / 594,360) = 227 cycles



(a)



(b)

Fig. 5-1. Coding schedule

(a) non-interleaved coding schedule; (b) interleaved coding schedule;

As Fig. 5-1-a shows, when 4x4 sub-MB of upper left corner is processing, 4x4 sub-MBs of its right and bottom site need to wait till it finishes the reconstruction value to predict. During the wait, there are a lot of arithmetical units will stand by, and it is hard to finish sixteen 4x4 sub-MB within 227 cycles. In order to fix the standby of arithmetical units and limited cycles issue, we adopt the frame-parallel encoding scheduling as shown in Fig. 5-1-b. Due to there is no any dependence between each MB of different frames during the algorithm of intra prediction, hardware can process MBs more efficient.

A. Intra-Residue with 16-pixel Throughput

With fast intra prediction algorithm, the required computation of every MB can be split into two parts.

The first part is the extra computation to determine the best block size and the candidates of prediction mode.

Luma_pixels + Cb_pixels + Cr_pixels

*=16 * 16 + 8 * 8 + 8 * 8 = 384 pixels*

The second part is the computation to find the best prediction.

Luma_pixels mode_candidates + Cb_pixels* mode_candidates + Cr_pixels * mode_candidates=*

*16 * 16 * 5 + 8 * 8 * 3 + 8 * 8 * 3 = 1,792 pixels (4x4 block type) or*

*16 * 16 * 4 + 8 * 8 * 3 + 8 * 8 * 3 = 1,536 pixels (8x8 block type) or*

*16 * 16 * 2 + 8 * 8 * 3 + 8 * 8 * 3 = 1,024 pixels (16x16 block type)*

Therefore, to combine these two parts and consider processing two MBs of different frames at the same period, the bottom neck is:

*(Max_Processing_Pixels_For_Fast_Mode_Decision+ Processing_Pixels_For_Fast_Block Size_Decision) * MB_numbers*

*=(1,792 + 384)*2=4352 pixels*

The pixel number of every cycle needs to handle is

(Total_Processing_Pixels)/(Avaliable_Processing_Cycles)

*=4352/(227*2)=9.59 pixels/cycle => 16 pixels/cycle*

So, the throughput is set to 16 pixels.

B. Intra-Reconstruction and Quality-Refine with 8-pixel Throughput

During proceeding quantization, reconstruction and quality layers, due to the computation is fewer, the throughput can be lower to reduce the area of hardware. The analysis shows as following.

First, the pixel number of every MB per quality layer is

Luma_pixels + Cb_pixels + Cr_pixels

*=16 * 16 + 8 * 8 + 8 * 8 = 384 pixels*

And the pixel number of processing two MB with 3 quality layers is

*Total_Pixels_per_MB * Number_of_MB * Number_Of_Quality_Layers=*

*384 * 2 * 3 = 2304 pixels*

Then the pixel number of every cycle needs to handle is

(Total_Processing_Pixels)/(Avaliable_Processing_Cycles)=

2304 / 454 = 4.48 pixels => 8 pixels

Therefore, the throughput is set to 8 pixels.

According to these analyses, we can adopt frame-parallel with interleaved coding flow and separate hardware architecture into two parts. The first part is intra–residue of 16 pixels throughput. The second part is intra reconstruction and quality refine of 8 pixels throughput; then we can achieve the tough system requirement by one set of hardware.

5.2. Architecture and Encoding Schedule

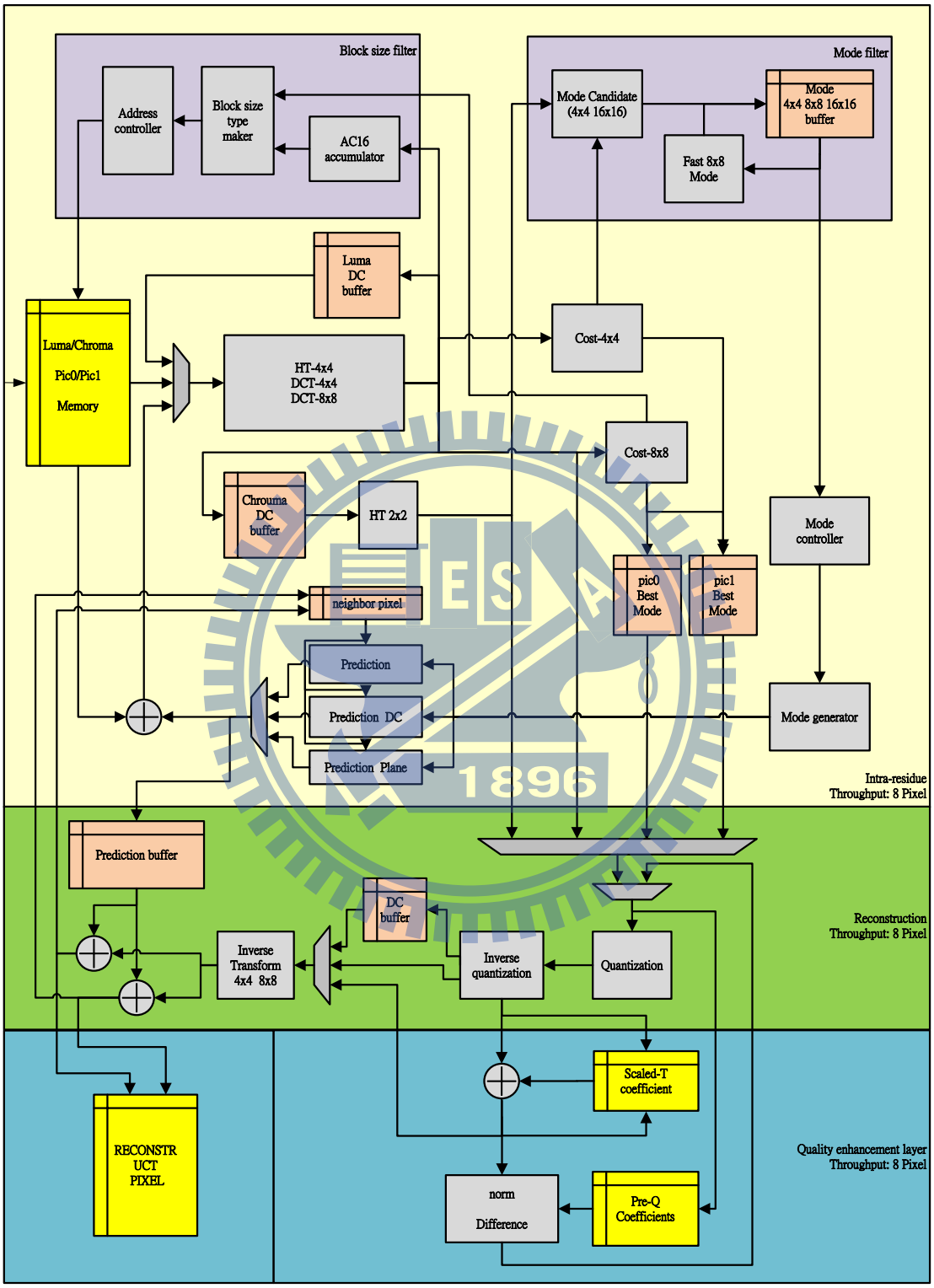


Fig. 5-2. Hardware architecture of fast intra prediction

Fig. 5-2 is the overall hardware architecture of this work. It can be divided into 3

blocks. The top block is to compute Intra–residue prediction with 16 pixels throughput. The center block is to compute quantization and reconstruction with 8 pixels throughput. And the bottom block is to compute quality layer with 8 pixels throughput.

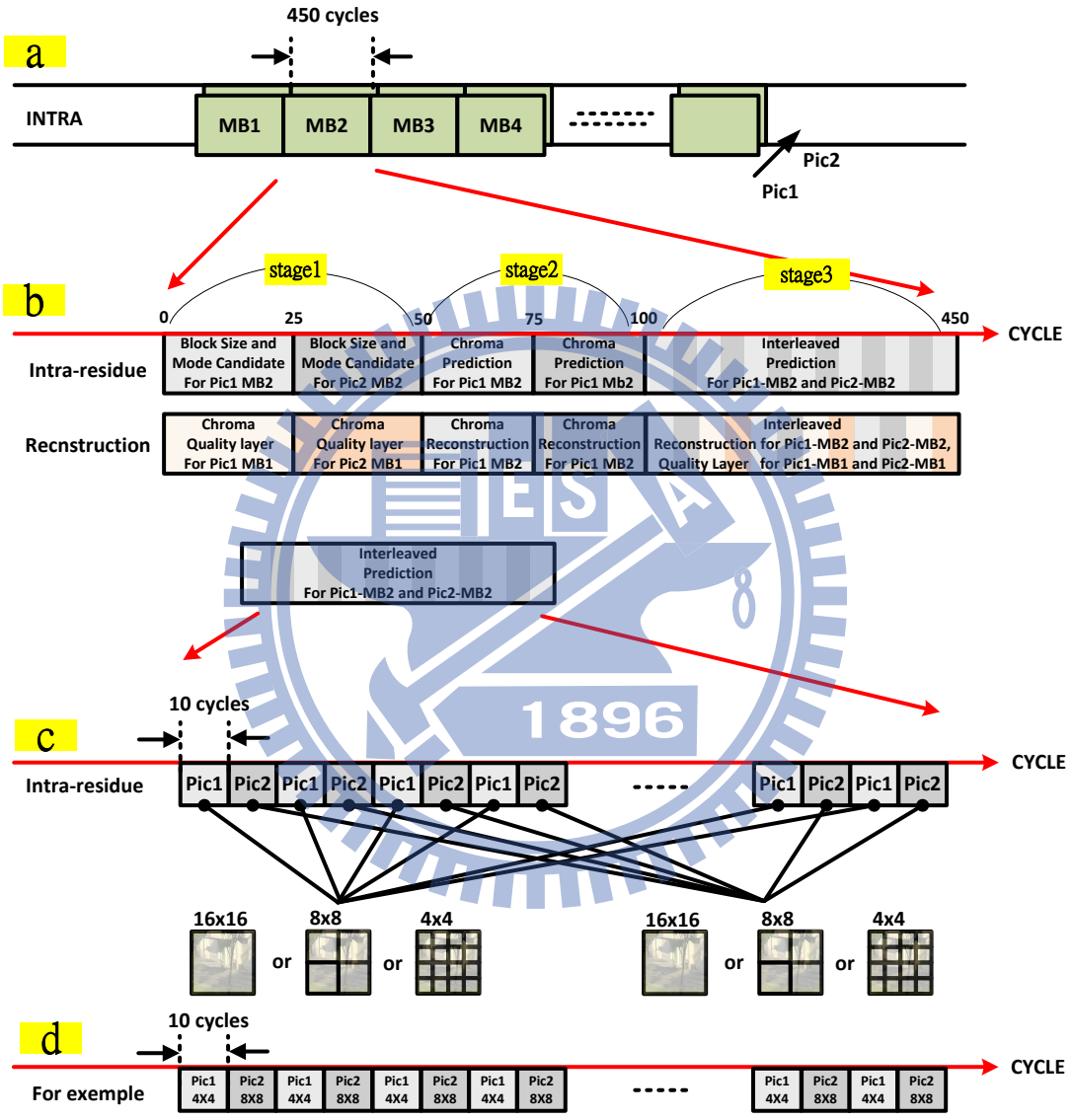


Fig. 5-3. Encoding schedule of fast intra prediction

Fig. 5-3 shows the corresponding encoding schedule of hardware. As Fig. 5-3-b shows, block size type and mode candidates of current two MBs would be calculated

by Intra-Residue modules. At the same time, reconstruction modules proceed quality enhancement layers of chroma block of previous two MBs. Then, Intra-Residue and reconstruction modules deal with base layer of current chroma blocks. Finally, the hardware interleaved computes two current MBs to avoid data dependency with embedded quality layer processing of previous MBs in the reconstruction loop.

About interleaved scheduling, as Fig. 5-3-c shows, MB1 and Mb2 alternately share hardware with each 10 cycles. And with fast algorithm of previous chapter, MB1 and MB2 will be one of 4x4, 8x8 and 16x16 block sizes.

5.3. Specific Components

According to the discussion of Section 5.1 and 5.2, we hope that the SVC intra encoder can share single prediction units for different block size computation. However, the calculation procedure is different for each block partition size. It is necessary to adjust some prediction units to combine different purposes. For example, 4x4 DCT calculates 4 pixels length rows and columns, but 8x8 DCT calculates 8 pixel lengths rows and columns.

Hence, in this section, we will introduce specific components with shared hardware. Finally, there is a simple summary from Table 5-1.

TYPE / Module	Luma prediction			Chroma prediction	Quality layer
	4x4	8x8	16x16		
Intra_Residue					
BlockSize		v			
CostMode	v	v	v	v	
Mode8x8		v			
ModeChroma				v	
Trans4816DC	v	v	v		
Hadamard2x2				v	
Prdiction_48DC	v	v	v	v	
Prdiction_DC	v	v	v	v	
Prdiction_Pls			v	v	
Prdiction_Plane			v	v	
Residue	v	v	v	v	
Reconstruction					
Quantization	v	v	v	v	v
IQuantization	v	v	v	v	v
ITrans48DC	v	v	v	v	v
Reconstruction416C	v		v	v	v
Reconstruction8x8		v			v
NormMinus					v

Table 5-1. The table of components with shared hardware

5.3.1. “CostMode” Module

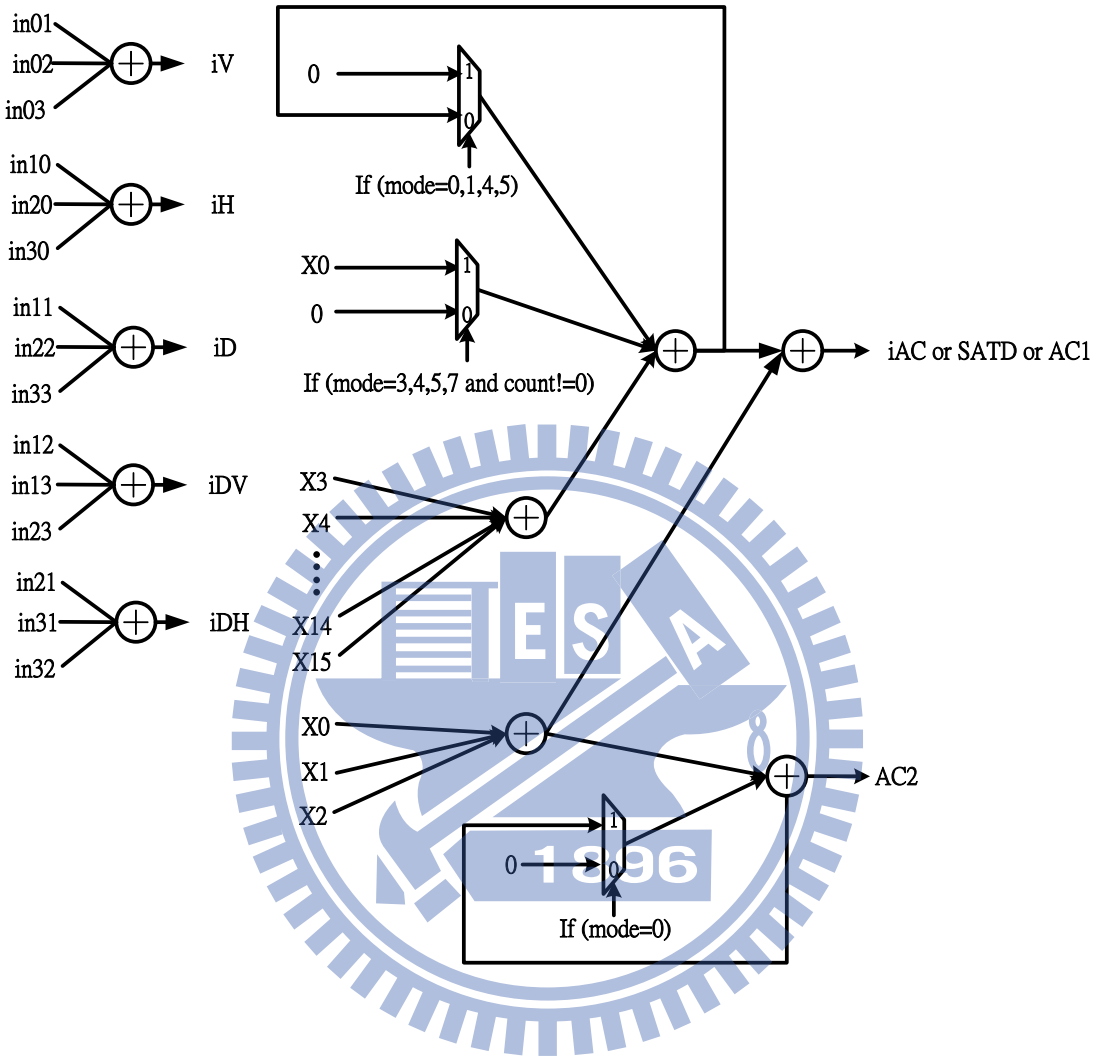


Fig. 5-4. Architecture of “CostMode”

MODE \ INPUT		X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
0	Mode4x4 and AC2	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
1	Mode16x16 and AC1_1	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
2	SATD 4X4	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
3	SATD 8X8	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
4	SATD 16X16	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
5	SATD 16X16	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33
6	AC1_2	in03	in12	in21	in30	in13	in22	in31	in23	in32	in31	in23	in32	in31	in23	in32	in31
7	SATD chrom	in00	in01	in01	in11	in02	in03	in12	in13	in20	in21	in22	in23	in30	in31	in32	in33

Table 5-2. Input selection of “CostMode”

The “CostMode” module not only provides mode candidates for 4x4 blocks and 16x16 blocks and the SATD for residue, but also calculates the AC1 and AC2 values for block size decision.

Fig. 5-4 shows the architecture of “CostMode”, the left hand side in the figure can calculate the directional tendency of texture (IV, IH etc.) for mode candidates. On the other hand, the right hand side in the figure provides the SATD, AC1 and AC2 with accumulation procedure. And the further description of the inputs is shown in the Table 5-2.

5.3.2. Prediction Modules

About mode prediction modules, we adopt the hardware designs proposed by [25]. The module “Prediction” can calculate different prediction modes with different block size types. Because it is not resource-efficient to complete DC or plane mode prediction within one cycle, the DC mode is calculated in module “Prediction DC” and the plane mode is calculated in module “Prediction Plane”.

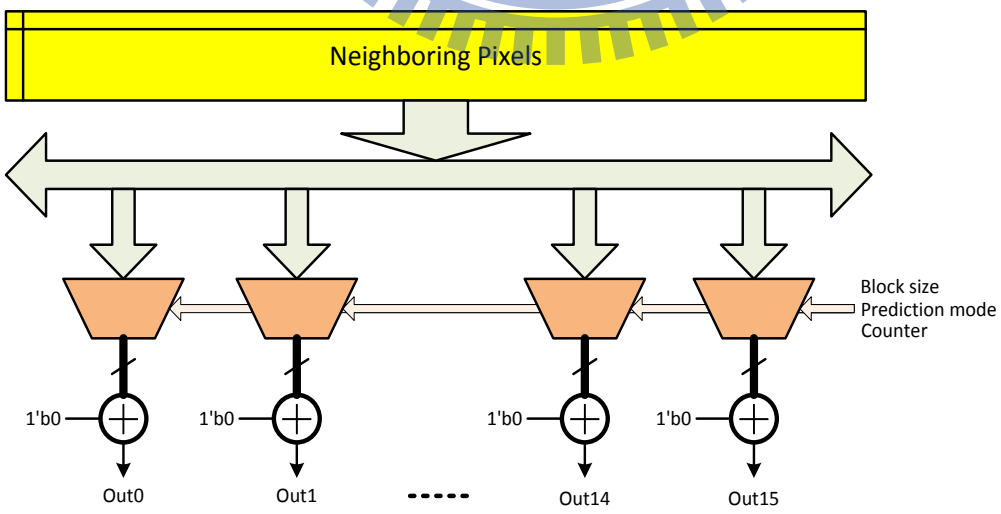


Fig. 5-5. Architecture of module “Prediction”

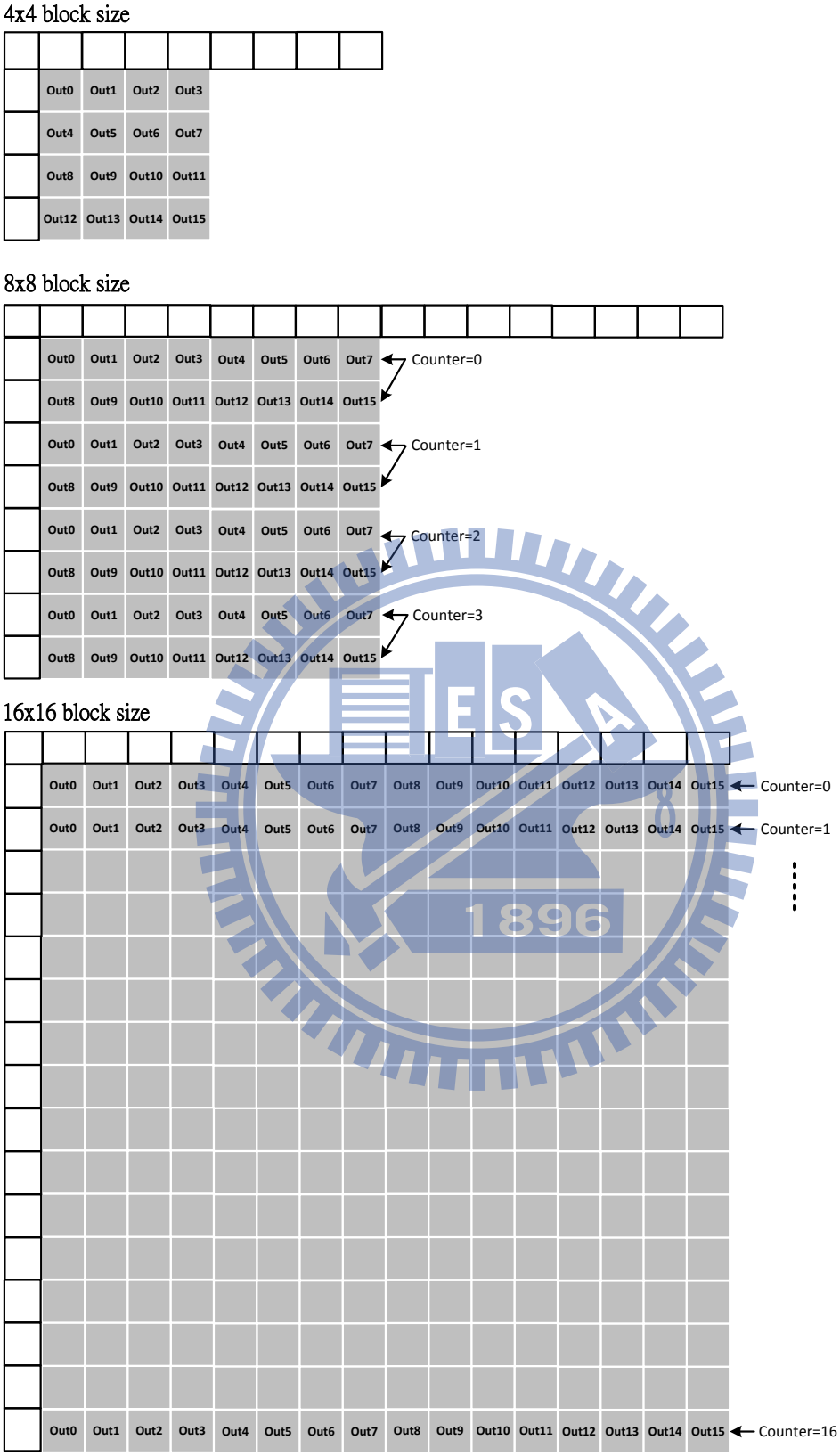


Fig. 5-6. Output assignment for module “Prediction”

Fig. 5-5 shows the architecture of “Prediction”, which can provide the 4x4 prediction modes within one cycle, the 8x8 prediction modes within four cycles, the 16x16 prediction modes within sixteen cycles and the chroma prediction modes within four cycles. From the spec, every output pixel of any prediction mode is calculated from maximum 3 neighboring pixels, so this architecture can provide each prediction mode by proper input assignment. And the output assignment is drawn in Fig. 5-6.

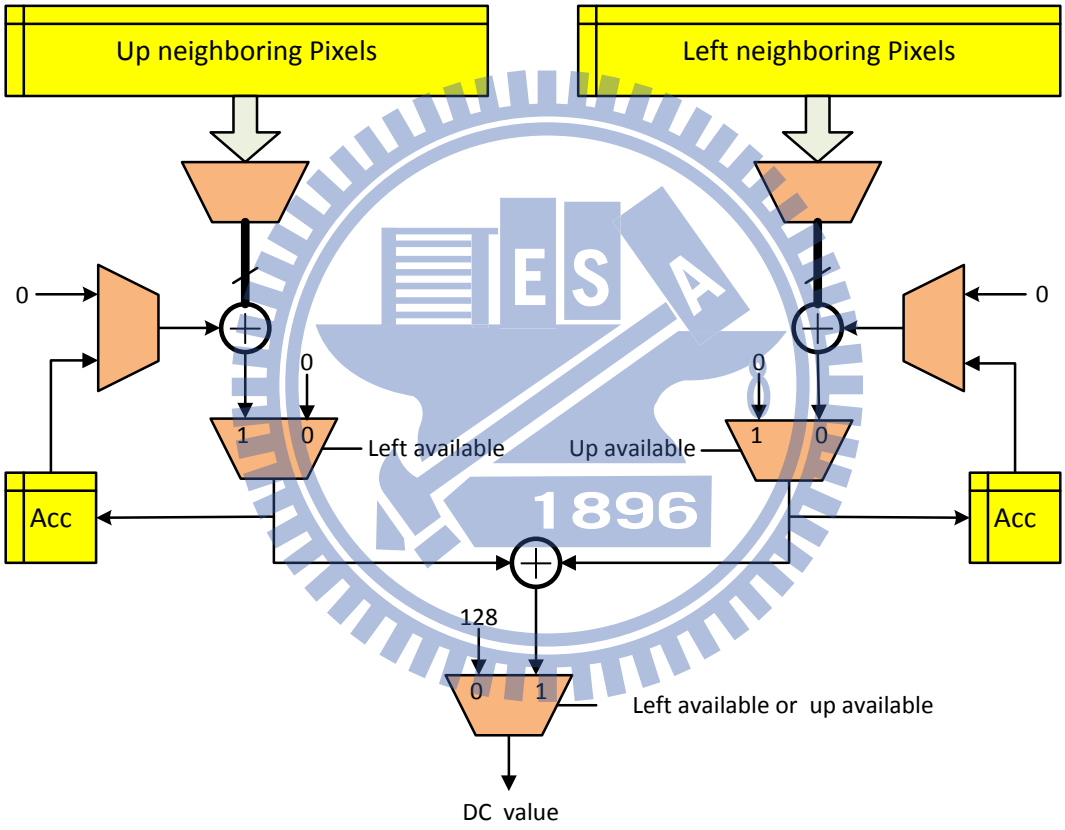


Fig. 5-7. Architecture of module “Prediction DC”

Fig. 5-7 shows the architecture of “Prediction DC”, which can provide the DC modes for the 4x4, 8x8, 16x16 and chroma block types. The 8x8 DC mode and the 16x16 DC mode is calculated by accumulation process.

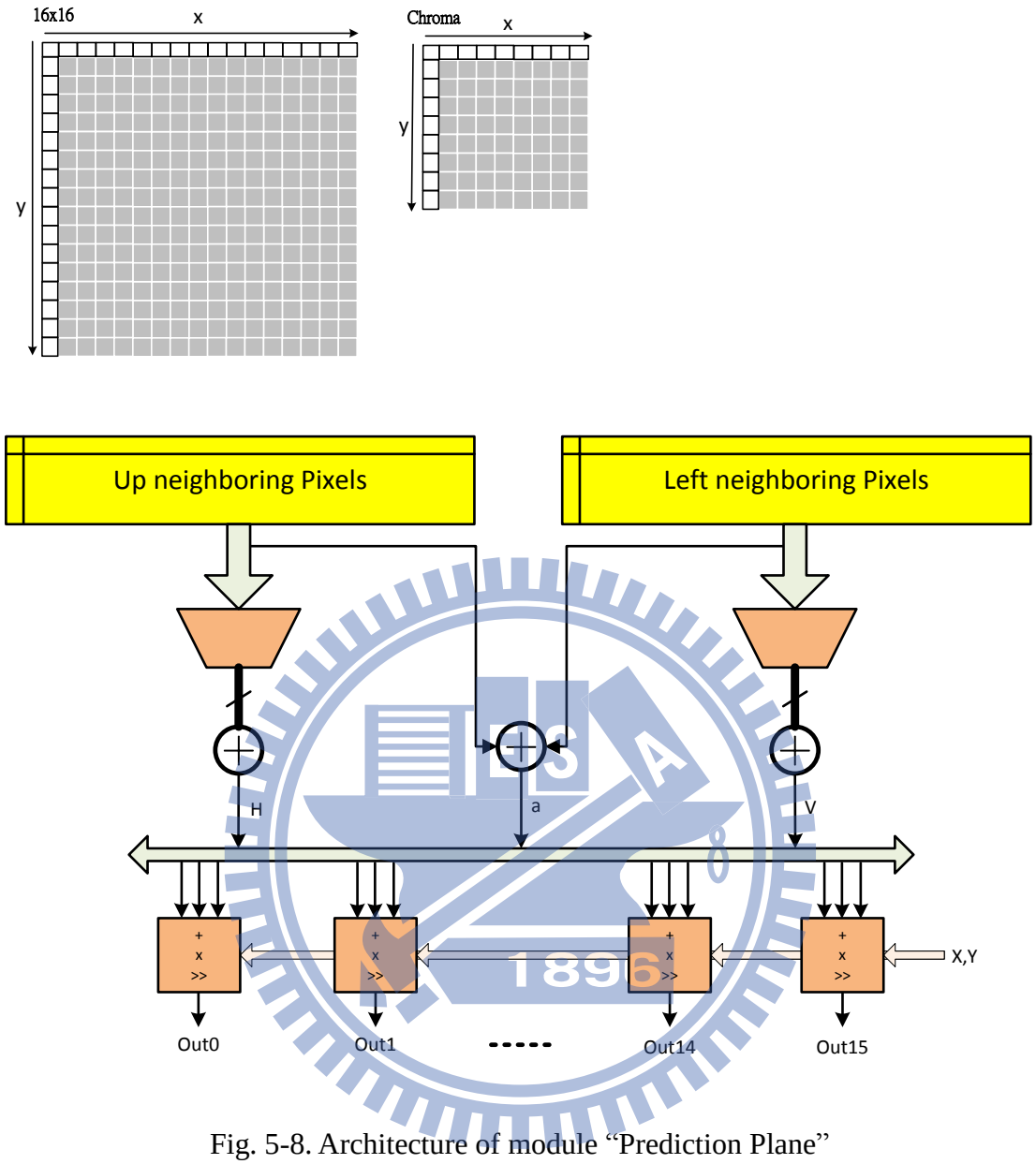


Fig. 5-8. Architecture of module "Prediction Plane"

Finally, Fig. 5-8 shows the architecture of "Prediction Plane", which can provide 16x16 plane mode and chroma plane mode.

The equations for 16x16 Plane mode are:

$$H=\sum x * [p(7+i,-1) - p(7-i,-1)], \text{for } i = 1\sim 8$$

$$V=\sum y * [p(-1,7+i) - p(-1,7-i)], \text{for } i = 1\sim 8$$

$$a=16 * [p(-1,15) - p(15,-1)]$$

$b=(5 * H + 32) \gg 6$

$c=(5 * V + 32) \gg 6$

$Pred(x,y)= [a + b * (x - 7) + c * (y - 7) + 16] \gg 5$

The equations for chroma Plane mode are:

$H=\sum x * [p(4 + i, -1) - p(4 - i, -1)], \text{for } i = 1 \sim 4$

$V=\sum y * [p(-1,4 + i) - p(-1,4 - i)], \text{for } i = 1 \sim 4$

$a=16 * [p(-1,7) - p(7, -1)]$

$b=(17 * H + 16) \gg 6$

$c=(17 * V + 16) \gg 6$

$Pred(x,y)= [a + b * (x - 3) + c * (y - 3) + 16] \gg 5$

5.3.3. “Trans48DC” Module

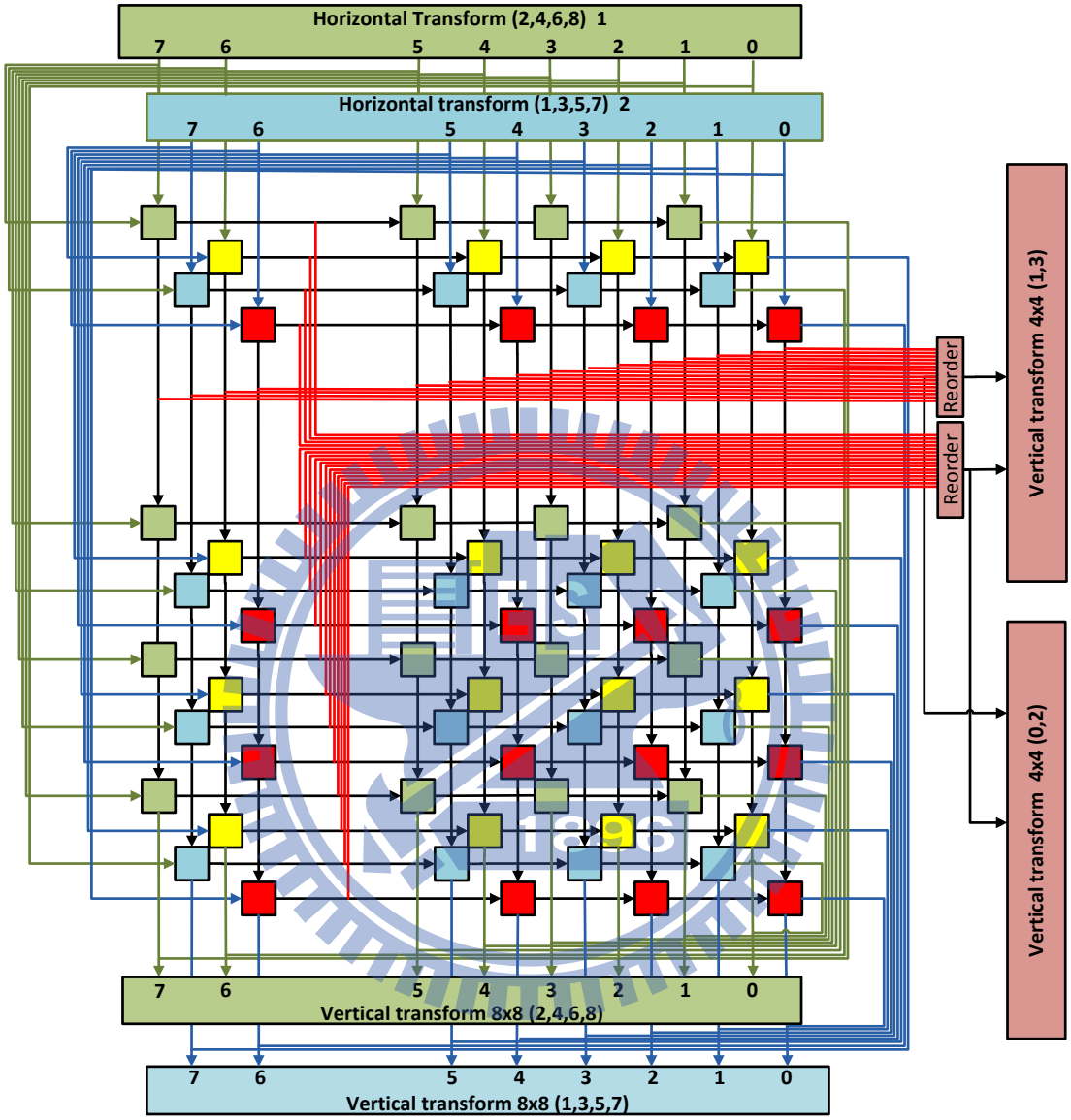
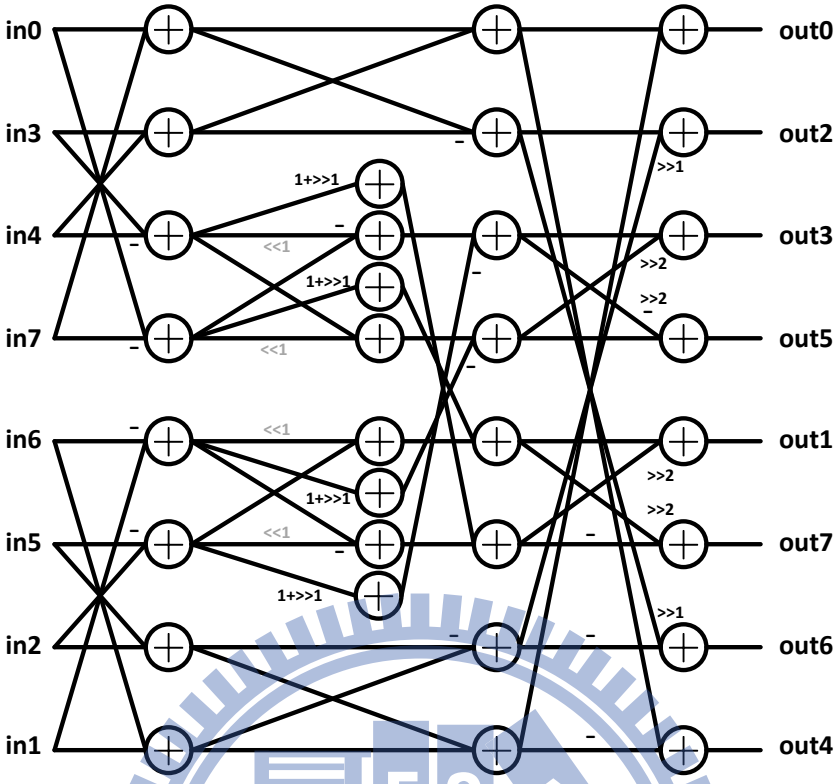
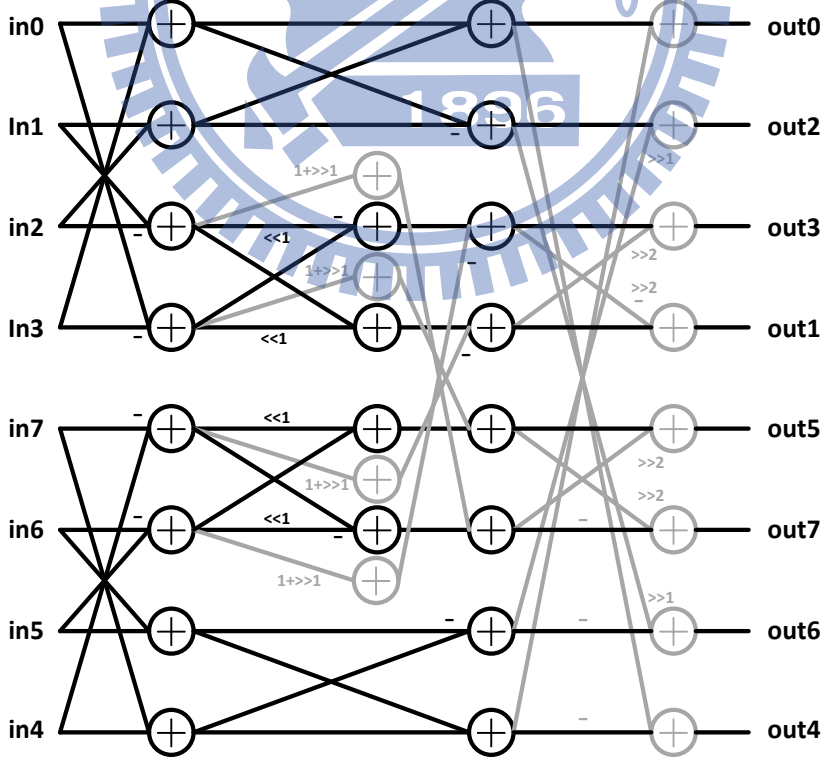


Fig. 5-9. Hardware architecture of module “Trans48DC”

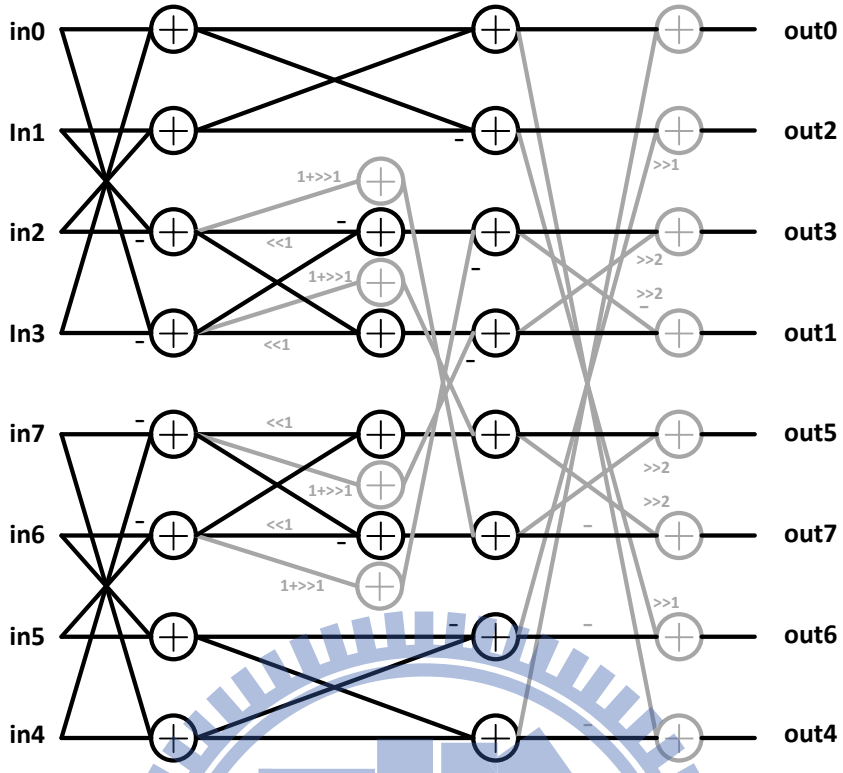
The architecture of “Trans48DC” is shown in Fig. 5-9, which can provide 4x4 Integer transform, 8x8 Integer transform, and 4x4 Hadamard transform.


in0
in3
in4
in7
in6
in5
in2
in1
out0
out2
out3
out5
out1
out7
out6
out4
1+>>1
<<1
>>2
>>1

(a)


in0
In1
in2
In3
in7
in6
in5
in4
out0
out2
out3
out1
out5
out7
out6
out4
1+>>1
<<1
>>2
>>1

(b)



(c)

Fig. 5-10. Data path for forward horizontal transform

Fig. 5-10 shows the architecture of "Horizontal transform", and (a) ~ (c) represent the data flow of 4x4 Integer transform, 8x8 Integer transform, and 4x4 Hadamard transform under the shared hardware.

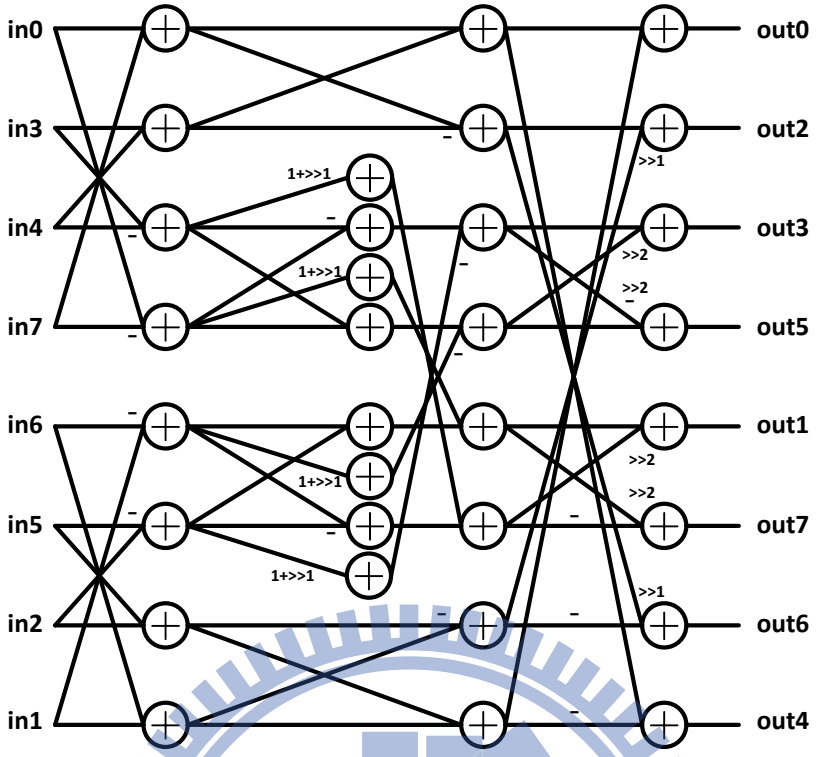
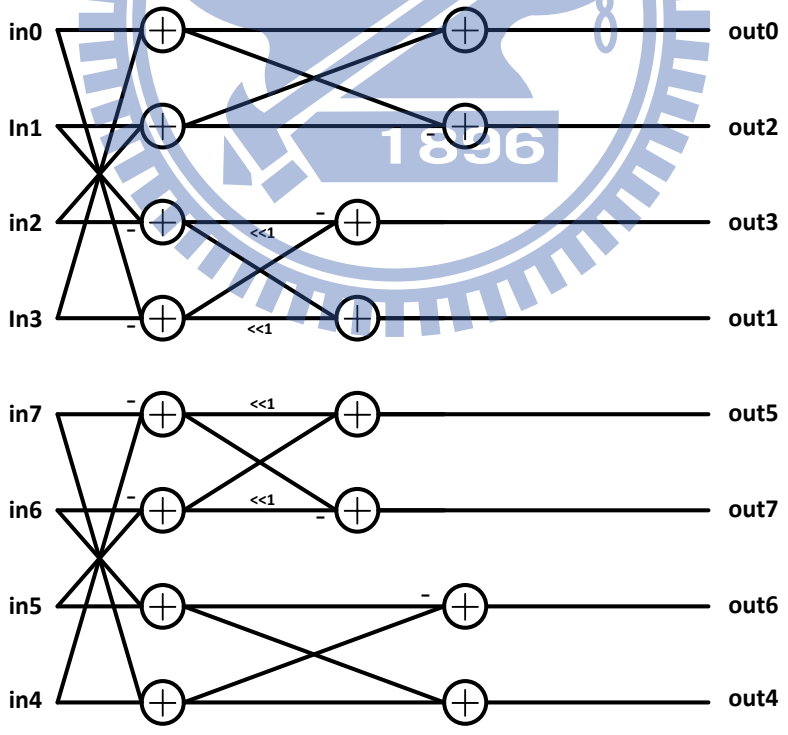
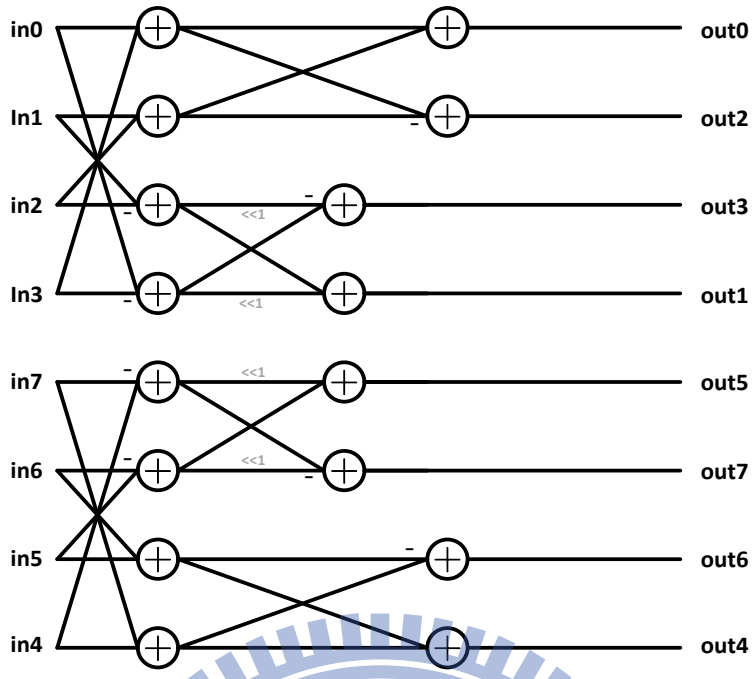


Fig. 5-11. Data path for forward 8x8 vertical transform



(a)

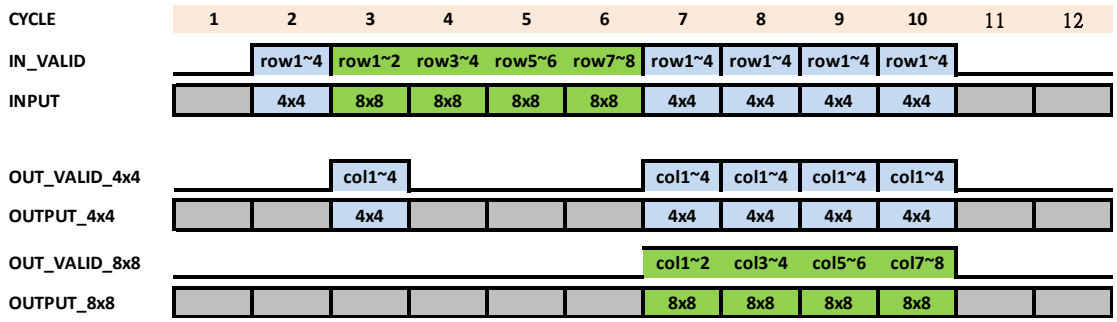


(b)

Fig. 5-12. Data path for forward 4x4 vertical transform

In addition, Fig. 5-11 shows the architecture of "Vertical transform 4x4",and (a) ~ (b) represent the 4x4 Integer transform, 4x4 Hadamard transform under the shared hardware. Fig. 5-12 shows the architecture of "Vertical transform 8x8".

(a)



(b)

Fig. 5-13. Timing diagram of “Trans48DC”

For all transform types, the calculation process can be divided into two steps. First, to calculate several row data by ”Horizontal transform”, and the result is temporarily stored in transpose buffers. Then the re-arranged data will be put into ”Vertical transform 4x4” or ”Vertical transform 8x8” for the final result.

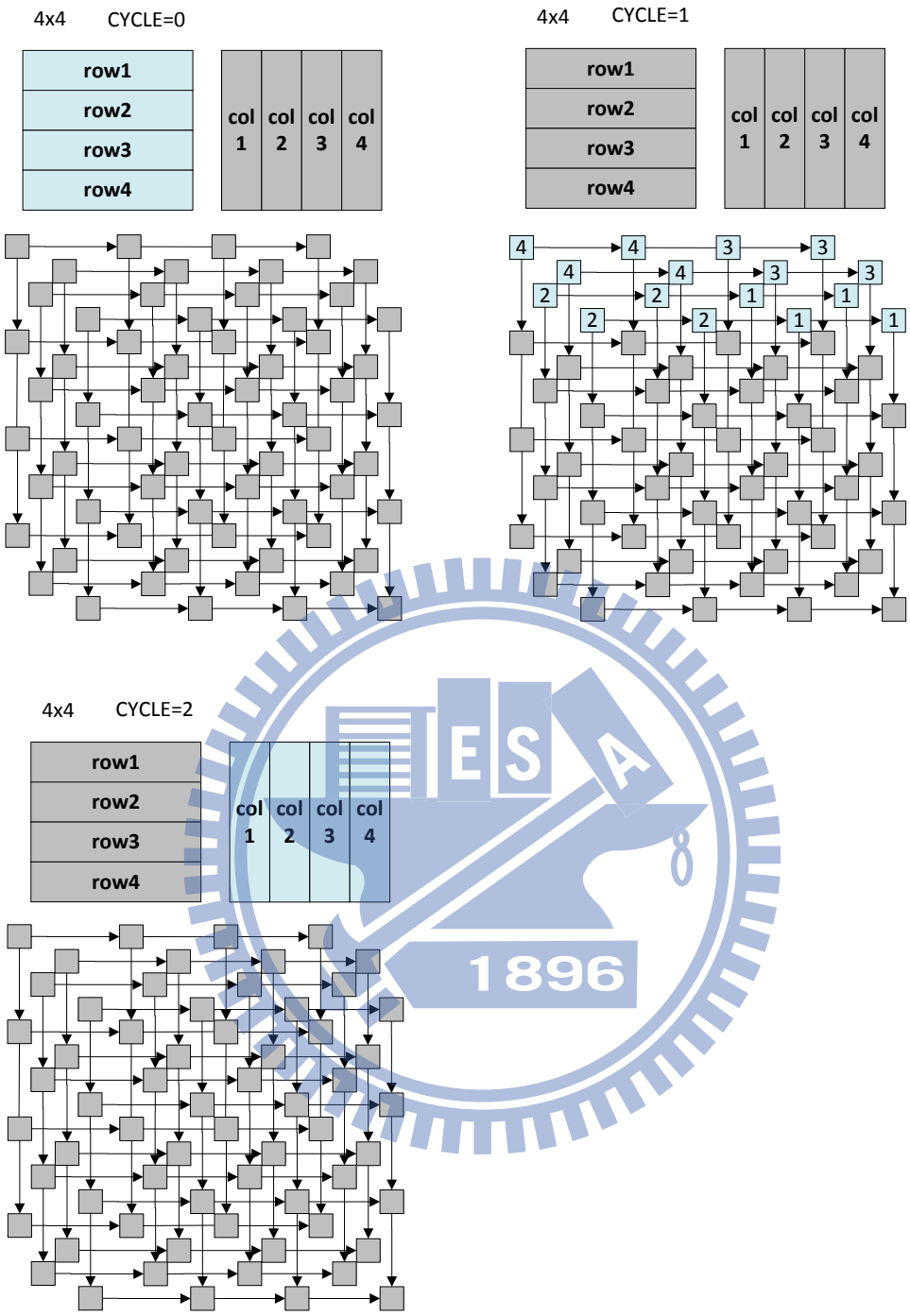


Fig. 5-14. Data flow for 4x4 transform

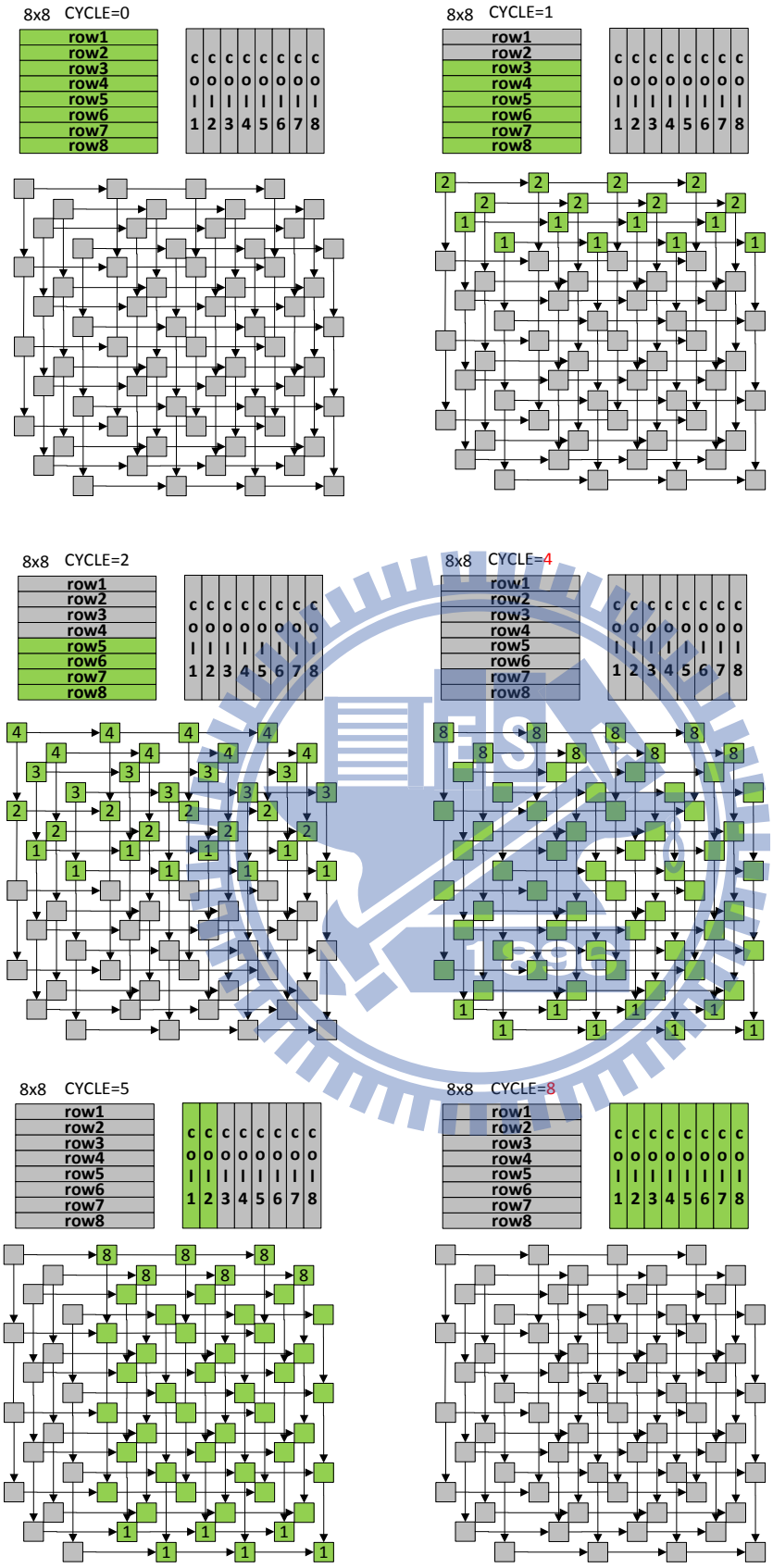


Fig. 5-15. Data flow for 8x8 transform

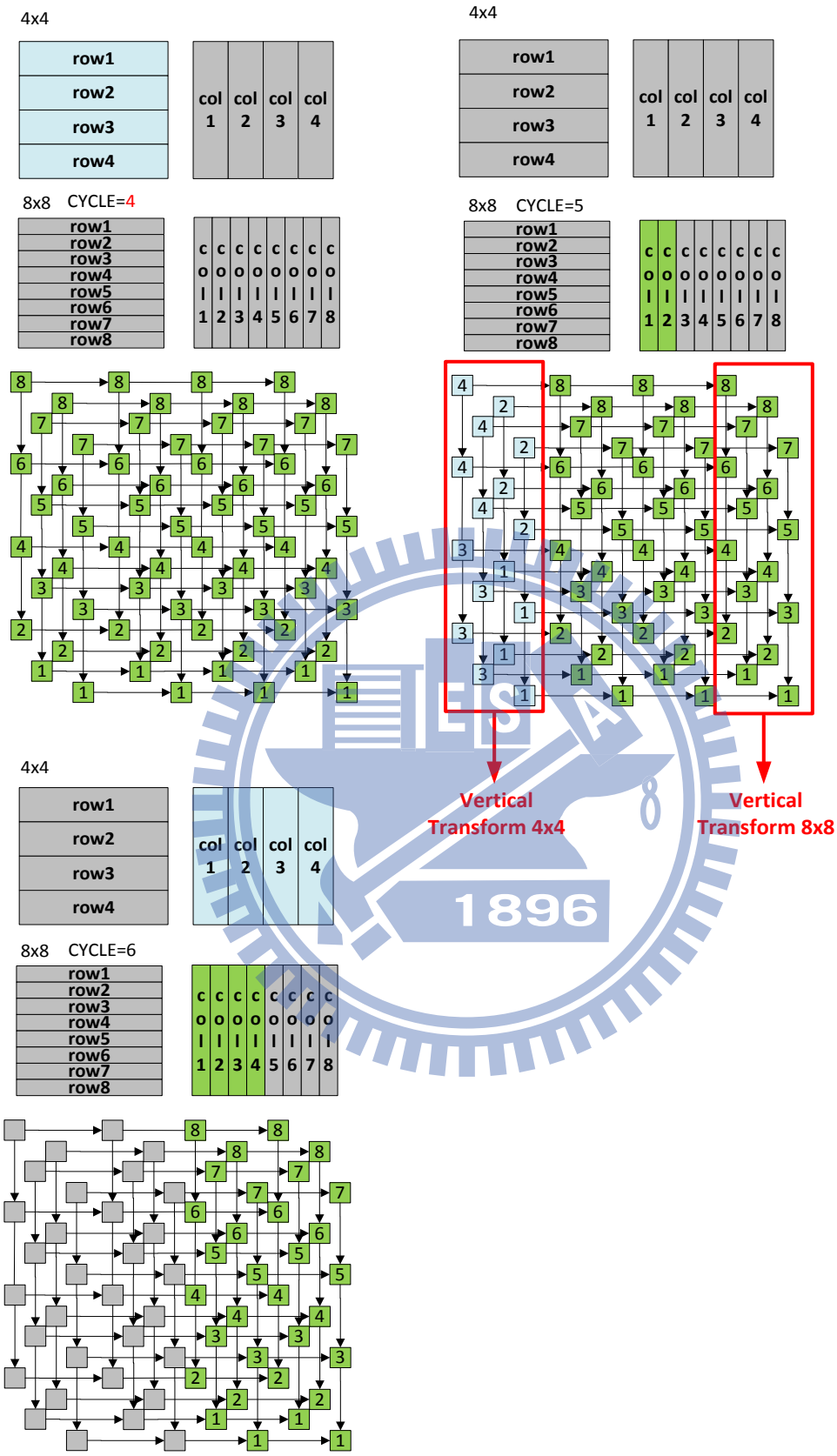
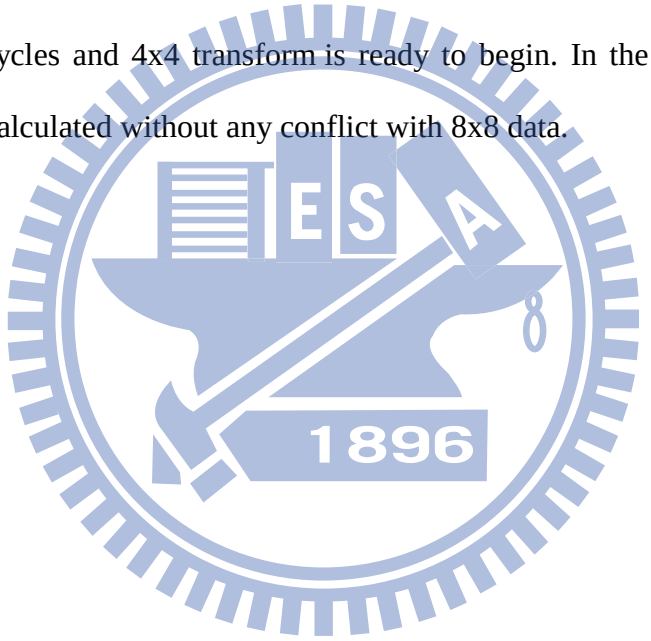


Fig. 5-16. Data flow for 4x4 transform and 8x8 transform

Fig. 5-14 shows how to calculate 4x4 Integer transform and 4X4 Hadamard transform, and Fig. 5-15 shows how to calculate 8x8 Integer transform.

For example, if we want to calculate 4x4 Integer transform, in the first cycle, four rows are calculated by “Horizontal transform” and the result is stored in transpose buffers. At the second cycle, the re-arranged data is calculated by “Vertical transform 4x4” for the final result.

Now we consider the 4x4 transform and 8x8 transform are calculate and the same time. As upper left corner in Fig. 5-16 shows, we assume that 8x8 transform has been processed by 4 cycles and 4x4 transform is ready to begin. In the next two cycles, 4x4 data will be calculated without any conflict with 8x8 data.



5.3.4. “ITrans48DC” Module

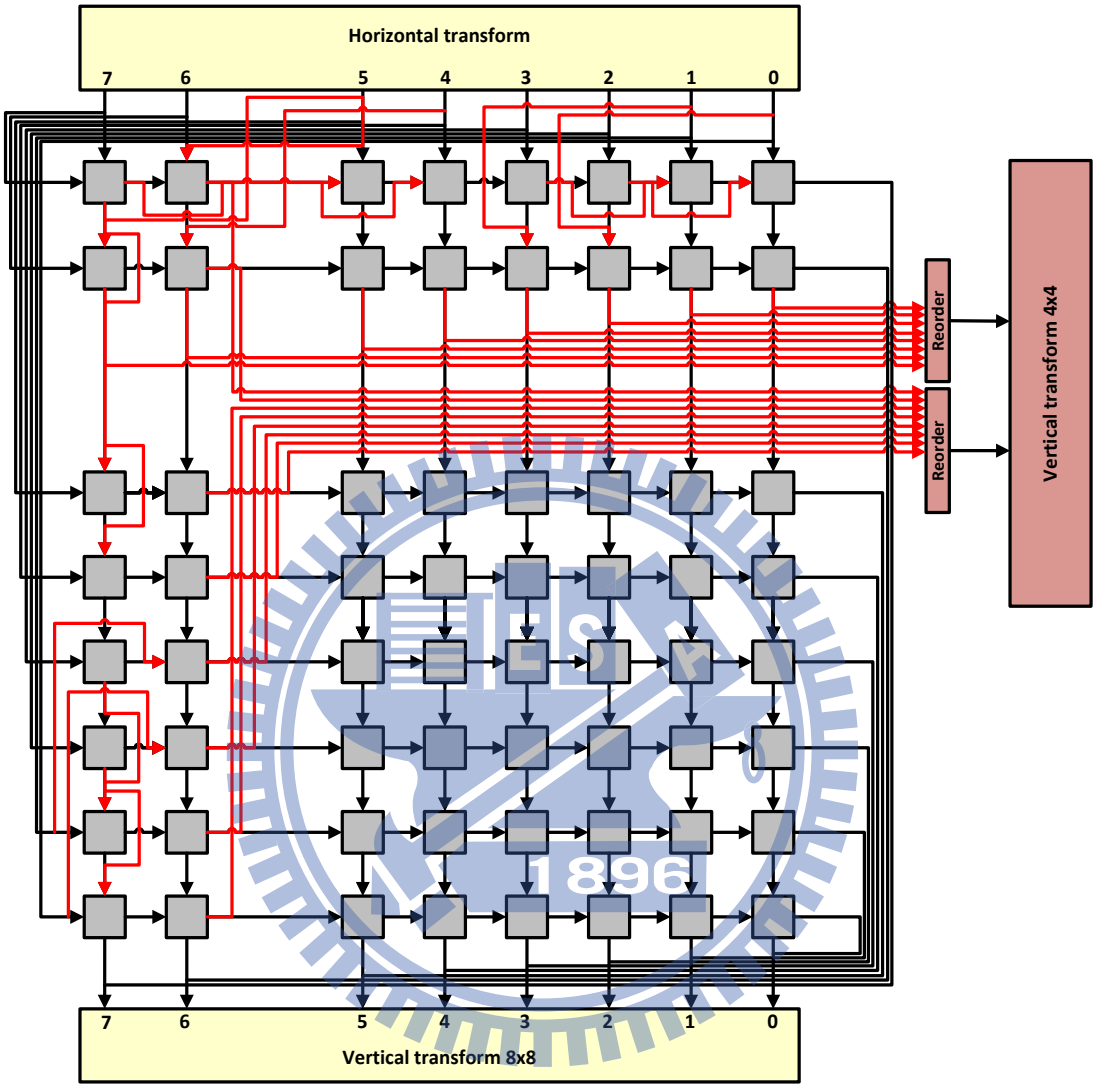
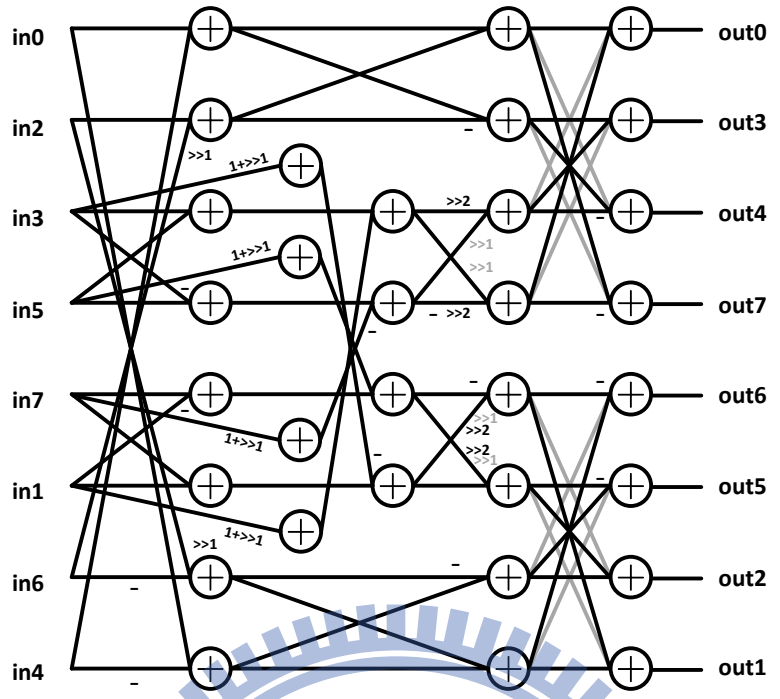
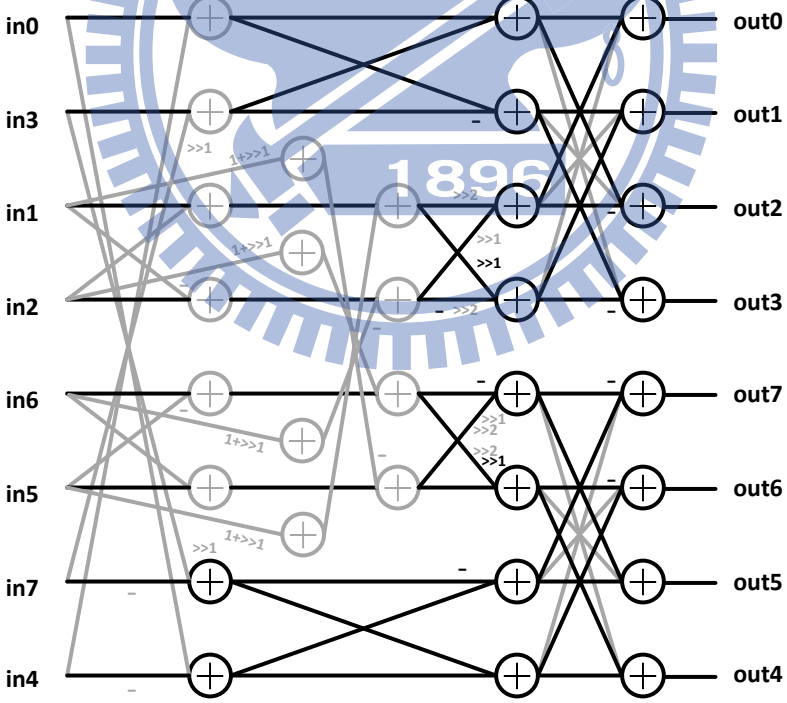


Fig. 5-17. Hardware architecture of module “ITrans48DC”

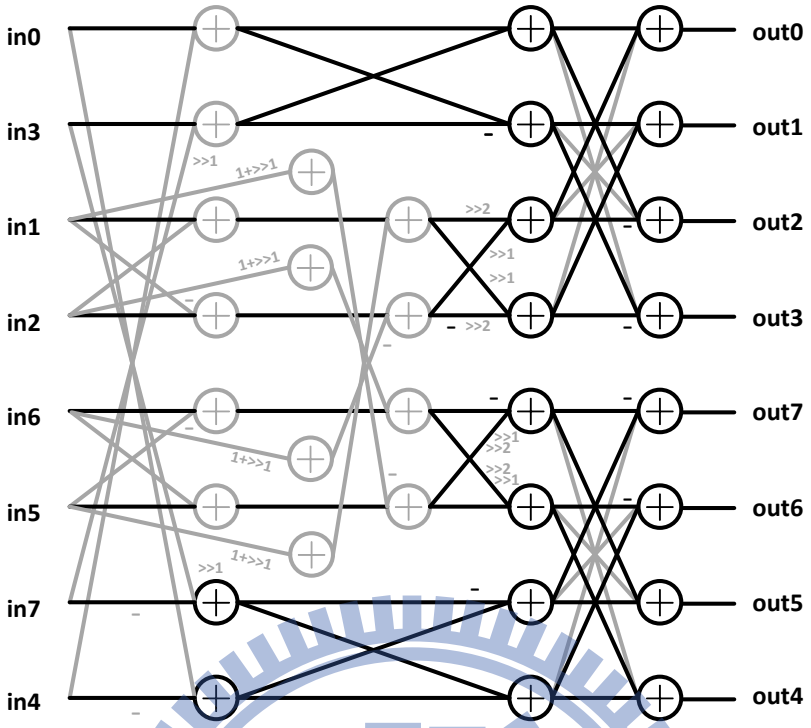
The architecture of “ITrans48DC” is shown in Fig. 5-17, which can provide 4x4 inverse Integer transform, 8x8 inverse Integer transform, and 4x4 inverse Hadamard transform.


in0
in2
in3
in5
in7
in1
in6
in4
out0
out3
out4
out7
out6
out5
out2
out1
>>1
1+>>1
>>2

(a)


in0
in3
in1
in2
in6
in5
in7
in4
out0
out1
out2
out3
out7
out6
out5
out4
>>1
1+>>1
>>2

(b)



(c)

Fig. 5-18. Data path for inverse horizontal transform

Fig. 5-18 shows the architecture of "Horizontal transform", and (a) ~ (c) represent the data flow of 4x4 inverse Integer transform, 8x8 inverse Integer transform, and 4x4 inverse Hadamard transform under the shared hardware.

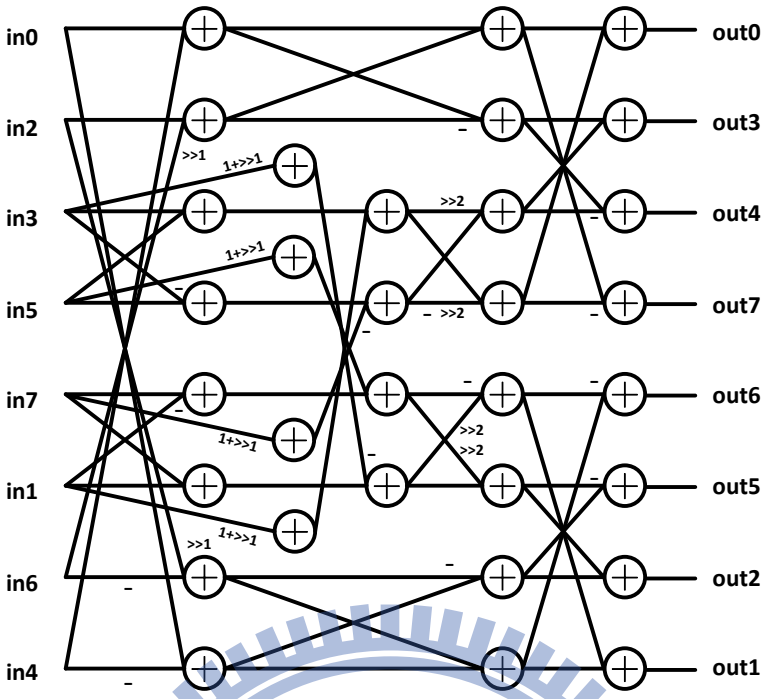
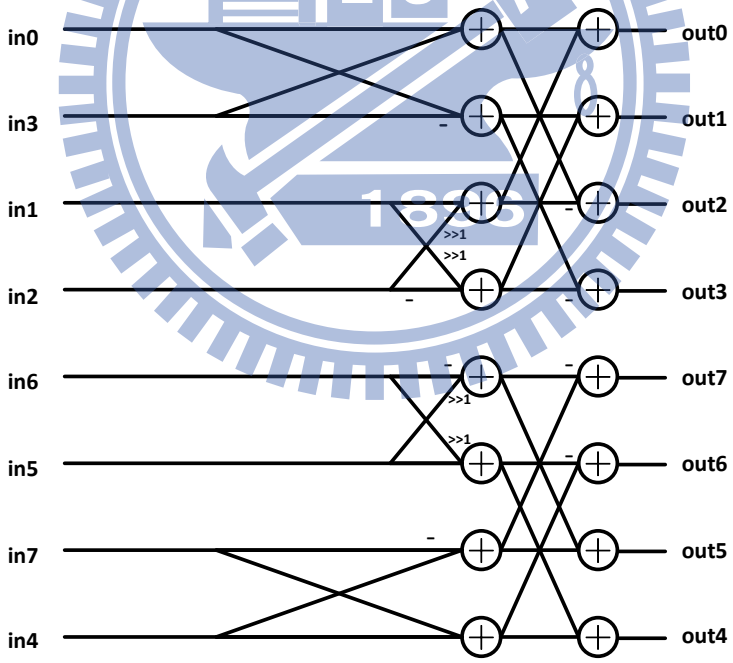
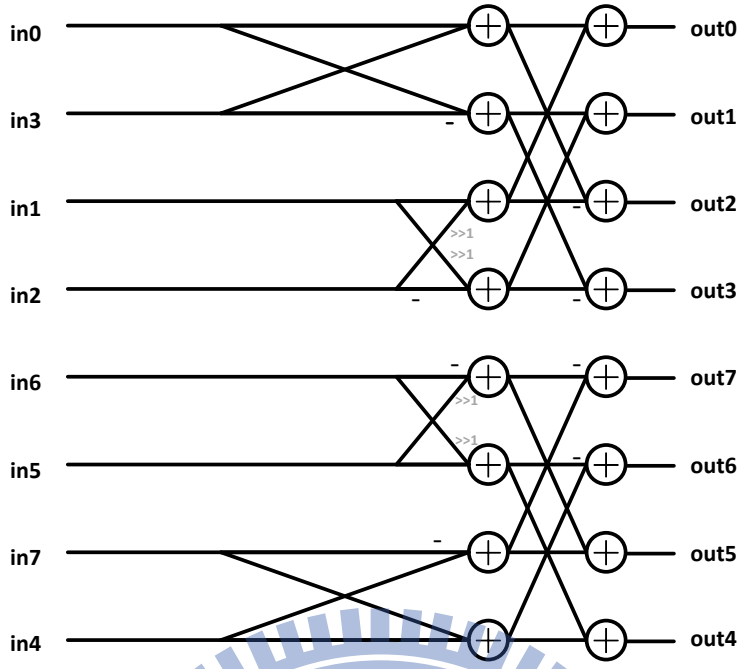


Fig. 5-19. Data path for 8x8 inverse vertical transform



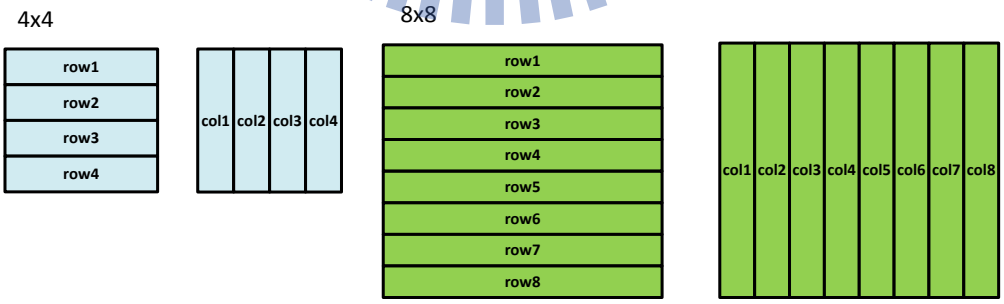
(a)



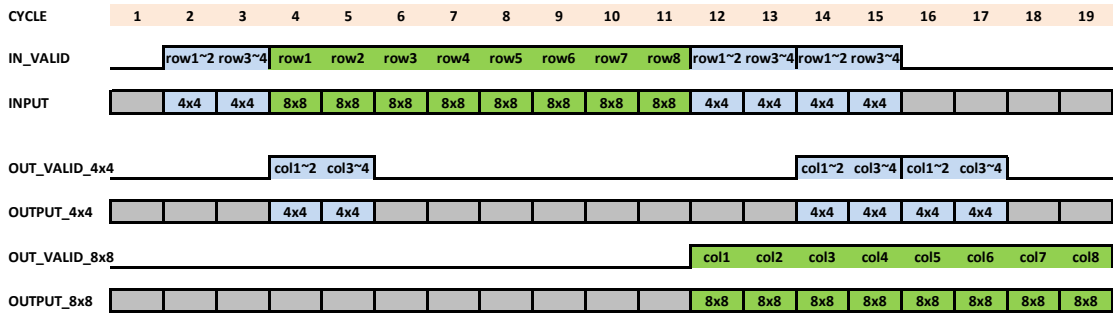
(b)

Fig. 5-20. Data path for 4x4 inverse vertical transform

In addition, Fig. 5-19 shows the architecture of "Vertical transform 4x4",and (a) ~ (b) represent the 4x4 inverse Integer transform, 4x4 inverse Hadamard transform under the shared hardware. Fig. 5-20 shows the architecture of "Vertical transform 8x8".



(a)



(b)

Fig. 5-21. Timing diagram of “ITrans48DC”

For all transform types, the calculation process can be divided into two steps. First, to calculate several row data by ”Horizontal transform”, and the result is temporarily stored in transpose buffers. Then the re-arranged data will be put into ”Vertical transform 4x4” or ”Vertical transform 8x8” for the final result.

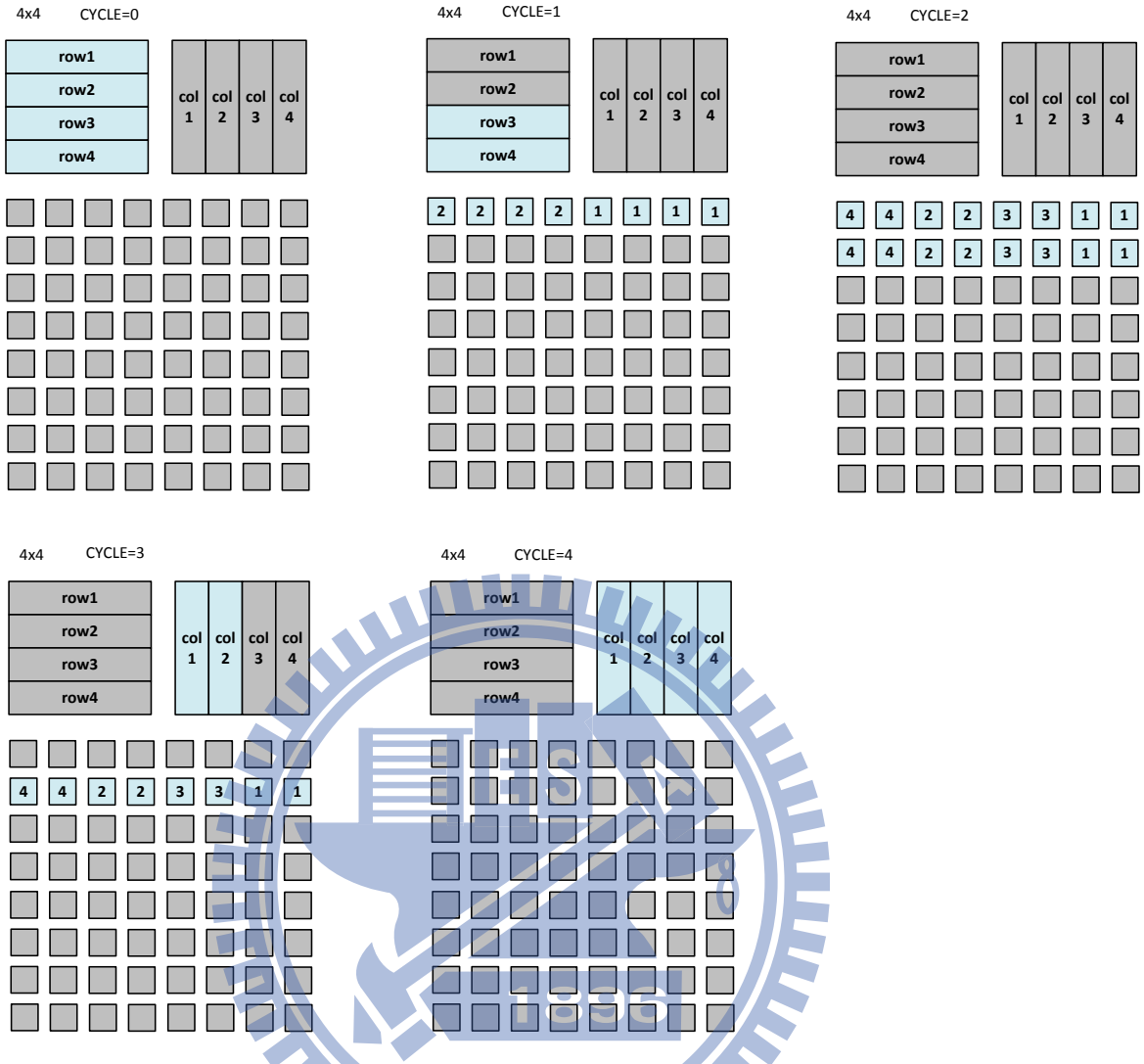


Fig. 5-22. Data flow for 4x4 inverse transform

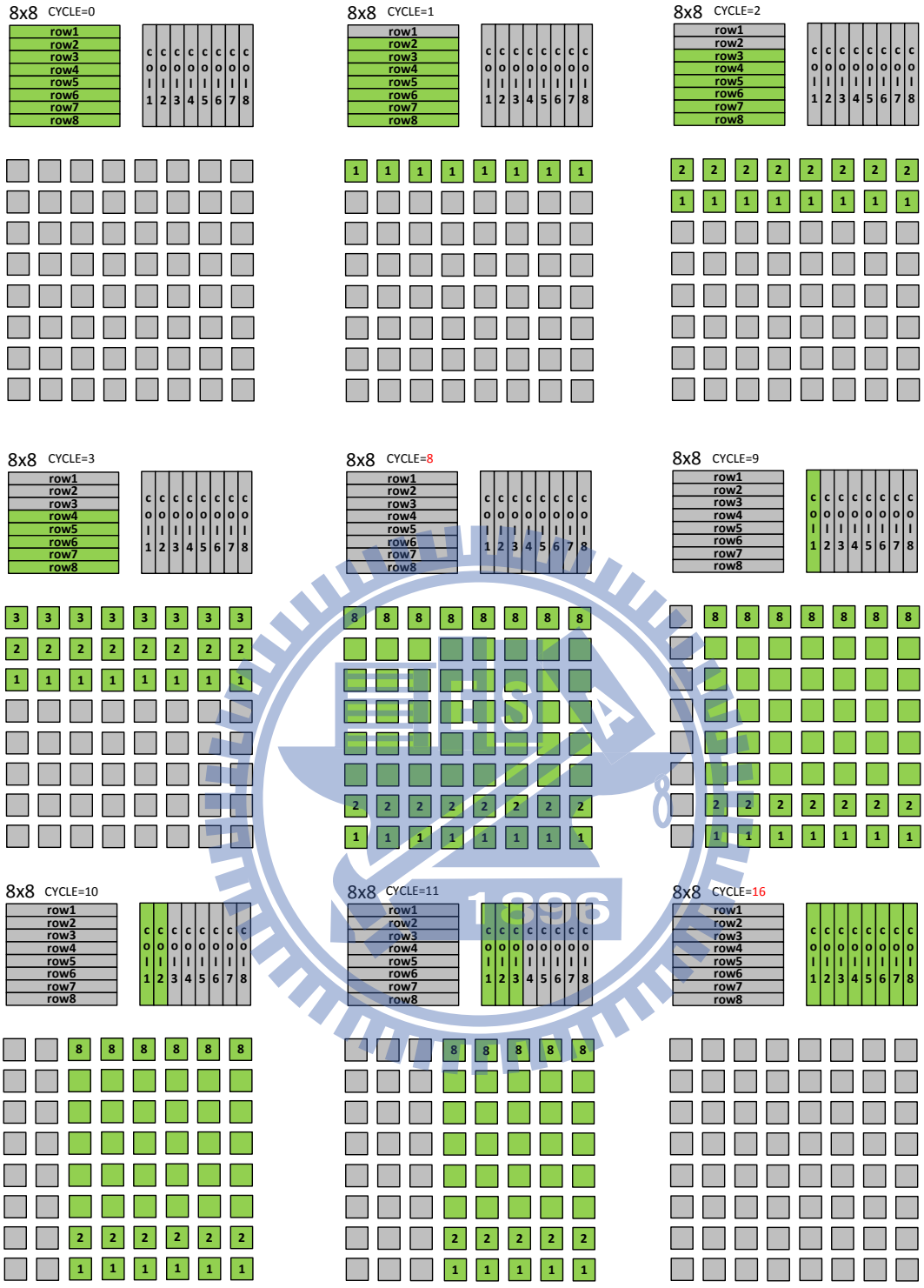


Fig. 5-23. Data flow for 8x8 inverse transform

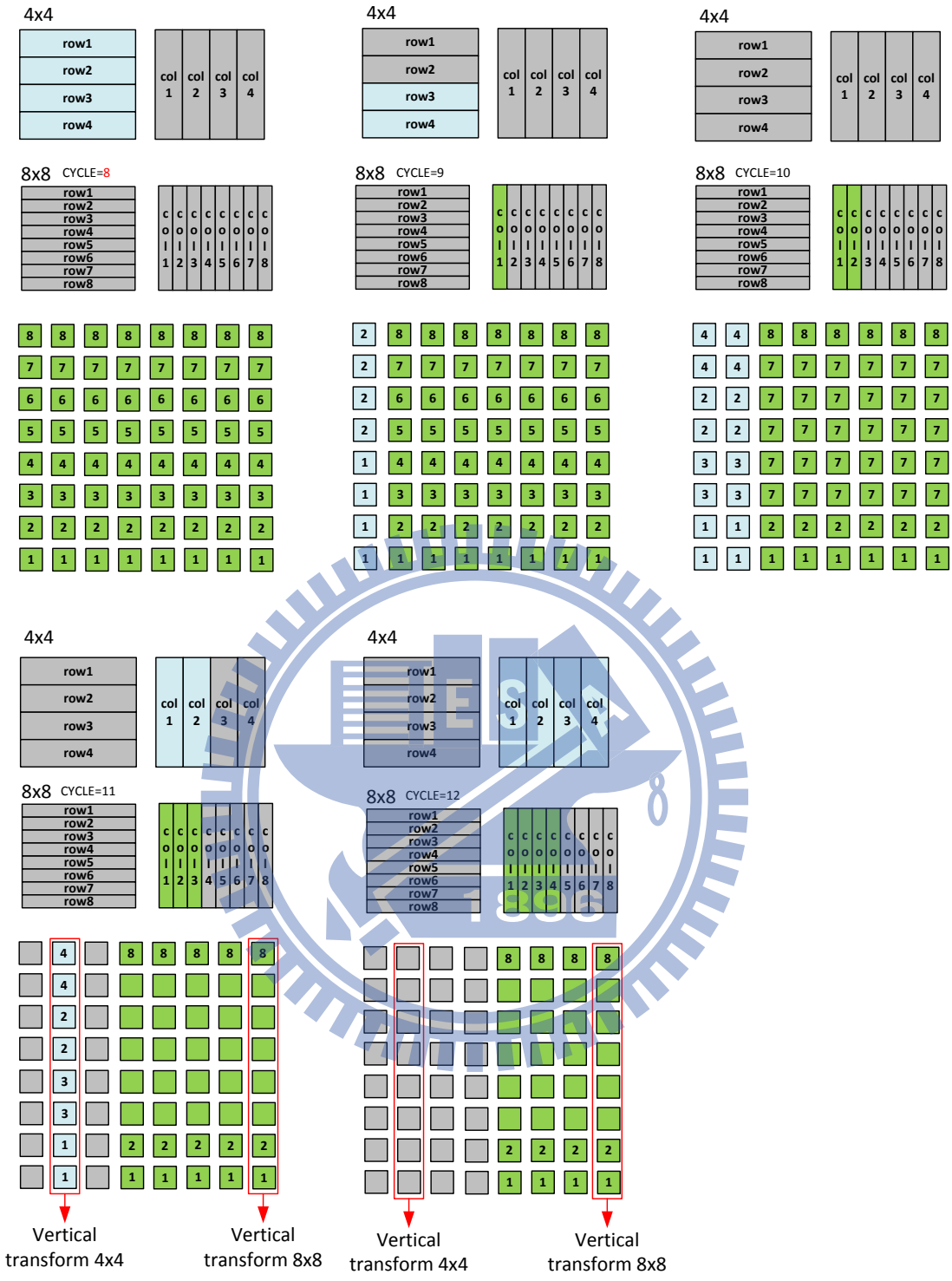


Fig. 5-24. Data flow for 4x4 inverse transform and 8x8 inverse transform

Fig. 5-22 shows how to calculate 4x4 inverse Integer transform and 4x4 inverse Hadamard transform, and Fig. 5-23 shows how to calculate 8x8 inverse Integer transform.

For example, if we want to calculate the 4x4 inverse Integer transform, in the first two cycles, four rows are calculated by “Horizontal transform” and the result is stored in transpose buffers. At the third and fourth cycles, the re-arranged data is calculated by “Vertical transform 4x4” for the final result.

Now we consider the 4x4 inverse transform and the 8x8 inverse transform are calculate and the same time. As upper left corner in Fig. 5-24 shows, we assume that the 8x8 inverse transform has been processed by 8 cycles and the 4x4 inverse transform is ready to begin. In the next four cycles, the 4x4 data will be calculated without any conflict with 8x8 data.

5.3.5. Memory

Table 5-3 shows all memories in this design, compared to an H.264/AVC encoder, an SVC encoder needs to store more information for quality enhancement coding.

Name	Type	Entry	Width	Number
Intra-Residue				
Current MB	Dual port	96 (2 MBs)	64	1
Neighboring pixels	Single port	960 (2 rows)	64	1
Neighboring modes	Single port	240 (2 rows)	16	1
Best coefficients	Dual port	24	76	4
Reconstruction				
Reconstructed pixels for BL	Single port	96 (2 MBs)	64	1
Quality Layer				
Pre-quantized coefficients	Two port	96 (2 MBs)	136	2
Scaled transform coefficients	Two port	96 (2 MBs)	136	2
Reconstructed pixels for EL	Single port	96 (2 MBs)	64	1

Table 5-3. Internal memory storage for proposed design

5.4. Implementation Results

This proposed design is implemented by Verilog and synthesized under the 90-nm CMOS technology process.

5.4.1. Gate Count

Table 5-4. List of gate count for [23], [24], [25] and proposed design

Module	[23]*	[24]**	[25]***	Proposed
Block size decision, Cost calculation and mode decision	11,934	12,923	24,779	8,262
Transform	9,392	19,868	42,783	25,573
Prediction and residual generator	17,449	6,646	31,362	11,896
Quantization and reconstructed path	29,378	70,104	86,344	43,344
Global control	20,846	24,534	21,137	52,415
Quality enhancement	Not Support	Not Support	9,024	6,151
Total	88,999	134,075	215,429	147,641

**: synthesizing in 100 MHz, UMC 0.18μm technology*

***: synthesizing in 145 MHz, UMC 0.13μm technology, and not support plane mode*

****: synthesizing in 135 MHz, UMC 90nm technology*

As Table 5-4 shows, the total gate count of the proposed intra prediction module excluded internal memory is about 148 k.

Compared to [25], the gate count of proposed design is effectively decreased by 30%. The main reason is that [25] uses double hardware resources (double transform modules, double SATD cost modules… etc.) to support intra prediction and this work

only uses one hardware resource.

Besides, if we compare the proposed design with [24], the gate count increment of this work is about 10%. The reason is that our proposed design needs extra components for plane mode and quality enhancement layer which are not support in [24].

5.4.2. Comparison

Table 5-5. Comparison between [23], [24], [25] and this work

Design feature	[23]	[24]	[25]	This work
CMOS technology	UMC 0.18μm	UMC 0.13μm	UMC 90nm	UMC 90nm
System pipeline	MB-based	MB-based	MB-based	MB-based
Pixel parallelism	8 pixels	8 pixels	16 pixels / 8 pixels	16 pixels / 8 pixels
Max operation frequency	100 MHz	145 MHz	135 MHz	135 MHz
Gate count	89 k	134 k	215 k	148 k
On-chip memory usage	N/A	(dual) 48 x 64 (x 1) (single) 16 x 112 (x 2) 8 x 96 (x 2)	(dual) 96 x 64 (x 1) 128 x 152 (x 1) (single) 960 x 64 (x 1) 240 x 16 (x 1) 128 x 128 (x 1) 128 x 136 (x 1) 128 x 64 (x 1) 96 x 64 (x 1) 96 x 136 (x 2)	(dual) 96 x 64 (x 1) 24 x 76 (x 4) (Two) 96 x 136 (x 4) (single) 96 x 64 (x 1) 240 x 16 (x 1) 960 x 64 (x 1)
Standard	H.264/AVC	H.264/AVC	SVC	SVC
Number of quality layers	1	1	3	3
Max target resolution	16 CIF (1408 x 960)	HD 1080p (1920 x 1080)	HD 1080p (1920 x 1080) + SD 480p (720 x 480) + CIF (352 x 288)	HD 1080p (1920 x 1080) +SD 480p (720 x 480) +CIF (352 x 288)
Frame rate	30 fps	30 fps	60 fps	60 fps

Throughput (MB / sec)	158,400	244,800	594,360	594,360
Hardware efficiency (throughput / gate count)	1.780	1.826	2.759	4.016
Processing cycles/MB	620	600	454	454
Cost function	SAITD	Enhanced DCT-based SATD	DCT-based SATD	Integer transform -based SATD
Plane mode	Y	N	Y	Y

Table 5-5 shows the comparison between [23], [24], [25] and this work. Compared to [24], this work uses more internal memories for quality enhancement coding in order to support the SVC standard. In conclusion, the hardware efficiency of this work is higher than the previous works.

5.5. Summary

In this chapter, we proposed an efficient hardware design for SVC intra encoder. With 135 MHz working frequency and 90 nm CMOS technology, the gate count of this work is about 148 k. And this encoder can support 3 quality layers, 3 spatial layers (CIF, SD 480p, and HD 1080p), and 3 temporal layers (up to 60 Hz frame rate).


E S A
1896

6. Conclusion and Future Work

In summary, the main contribution of this thesis can be divided into two parts:

First, we propose a fast two step intra prediction algorithm in Chapter 4. We observe the transformed patterns to determine the flatness and the directional tendency of texture. Then we use the result to choose the suitable block partition size and select few prediction modes. Compared to the reference software, JM, this proposed algorithm skips over 80% of candidate modes, with similar quality (average BD-PSNR difference: -0.01 dB for CIF, +0.12dB for 1080p, average BD-Rate difference: +0.01% for CIF, -3.13% for 1080p).

Second, in Chapter5, the implementation of a SVC intra encoder is introduced. Through the application of fast intra prediction algorithm and interleaved scheduling with embedded quality layer processing in the reconstruction loop, this encoder can support three quality layers, three spatial layers (CIF, SD 480p and HD 1080p) and up to 60 frames per second.

For the future work, the proposed design can be further integrated with inter prediction and inter-layer prediction for more powerful coding efficiency.


E S A
1896

Reference

[1] *Advanced Video Coding for Generic Audiovisual Services,* ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), Joint Video Team, 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, pp. 560–576, July 2003.

[3] T. Wiegand, G. Sullivan, J. Reichel, H. Schwarz and M. Wien, ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6: JVT-X201 'Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10/Amd.3 Scalable video coding,' 24th Meeting, Geneva, Switzerland, Jun.29-Jul.5, 2007.

[4] C. -T. Huang, C. -Y. Chen, Y. -H. Chen, and L. -G. Chen, "Memory analysis of VLSI architecture for 5/3 and 1/3 motion-compensated temporal filtering," *in proceeding of IEEE International Conference on Acoustic Speech, and Signal Processing*, vol. 5, pp. v/93-v/96, Mar. 2005.

[5] Y. -H. Chen, T. -D. Chuang, Y. -J. Chen, and L. -G. Chen, "Bandwidth-efficient encoder framework for H.264/AVC scalable extension," *in proceeding of IEEE International Symposium on Multimedia Workshop,* pp. 401-406, Dec. 2007.

[6] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 17, no. 9, pp. 1103-1120, Sep. 2007.

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Hierarchical B Pictures," Joint Video

Team, Doc. JVT-P014, Jul. 2005.

[8] A. Segall and G. 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.

[9] H. Schwarz, D. Marpe, and T. Wiegand, "Analysis of hierarchical B-pictures and MCTF," *in proceeding of IEEE International Conference on Mutimedia and Expo*, pp. 1929-1932, Jul. 2006.

[10] Y. -H. Chen, T. -D. Chuang, Y. -H. Chen, C. -H. Tsai, and L. -G. Chen, "Frame-parallel design strategy for high definition B-frame H.264/AVC encoder," *in proceeding of IEEE International Symposium on Circuits and Systems*, pp. 29-32, May. 2008

[11] H. Schwarz, T. Hinz, H. Kirchihoffer, D. Marpe, and T. Wiegard, *"Technical Description of the HHI Proposal for SVC CE1,"* ISO/IEC, JTC 1/SC 29/WG 11 Doc. M11244, Oct. 2004.

[12] Z. Kun, Y. Chun, L. Qiang, and Z. Yuzhou, "A Fast Block Type Decision Method for H.264/AVC Intra Prediction," *the 9th International Conference on Advanced Communication Technology*, vol.1. pp.673-676, Feb.2007.

[13] G. Tian, T. Zhang, X. Wei and S. Goto, "A block type decision algorithm for H.264/AVC intra prediction based on entropy feature," *in proceedings of IEEE asia pacific conference on circuits and systems, Macao, China*, pp. 1348-1351, Nov. 2008-Dec. 2008.

[14] S. Kim, T. Lee and K. Chung, "An Efficient and Fast Block Size Decision Exploiting Boundary Information on Inner Block for H.264/AVC Intra

Prediction," *in proceedings of the 2008 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 101-104, 2008.

[15] T. Zhang, G. Tian, T. Ikenaga, etc. "A novel fast block type decision algorithm for intra prediction in H.264/AVC high profile," *in proceedings of the 23rd international technical conference on circuits/systems, computers and communications*, 2008.

[16] X. Jing, W.-C. Siu, L.-P. Chau, and A.G. Constantinides, "Fast intra mode decision algorithm for H.263 to H.264/AVC transcoding," *in Proc. Int. Conf. Neural Networks Signal Process*, ICNNSP'08, pp. 666-670, Jun. 2008.

[17] C. -H. Hsia, J. -S. Chiang, Y. -H. Wang, and T. -Y. Teng, "Fast intra prediction mode decision algorithm for H.264/AVC video coding standard," *in proceeding of IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, vol. 2, pp. 535-538, Nov. 2007.

[18] J. -C. Wang, J. -F. Wang, J. -F. Yang, and J. -T. Chen, "A fast mode decision algorithm and its VLSI design for H.264/AVC intra-prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1412-1422, Oct. 2007.

[19] Y. -C. Wei and J. -F. Yang, "Transformed-domain intra mode decision in H.264/AVC encoder," *in proceeding of IEEE Region 10 Conference*, pp. 1-4, Nov. 2006.

[20] D. Quan and Y. –S. Ho, "Categorization for fast intra prediction mode decision in H.264/AVC," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 1049-1056, May 2010.

[21] B. La, M. Eom and Y. Choe, “Fast Mode Decision for Intra Prediction in H.264/AVC Encoder,” *IEEE International Conference on Image Processing*, pp. 321-324, Sep. 2007 - Oct. 2007.

[22] H.264/MPEG-4 AVC *reference software*, JM 12.4.

[23] H. -Y. Lin, K. -H. Wu, B. -D. Liu, and J. -F. Yang, “An efficient VLSI architecture for transform-based intra prediction in H.264/AVC,” *IEEE Transactions on Circuits and Systems for Video Technology.* Vol.20, no.6, pp.894 -906, Jun. 2010.

[24] Y. -K. Lin, D. -W. Li, C. -C. Lin, T. -Y. Kuo, S. -J. Wu, W. -C. Tai, W. -C. Chang, and T. -S. Chang, “A 242mW 10mm^2 1080p H.264/AVC high-profile encoder chip,” *IEEE International Solid-State Circuits Conference Digest of Technical Papers*, pp. 314-615, Feb. 2008.

[25] T. -Y. Chen, “Analysis and Intra Encoder Design for H.264/AVC Scalable Extension,” *Master Thesis of National Chiao-Tung University*, Taiwan, Aug. 2009.

Biographical Notes

姓名：溫孟勳

學歷：

國立交通大學電子所系統組	碩士	(民國 98 年 09 月 ~ 民國 100 年 09 月)
國立交通大學電子工程學系	學士	(民國 94 年 09 月 ~ 民國 98 年 06 月)
國立新竹高級中學		(民國 91 年 09 月 ~ 民國 94 年 06 月)

得獎事蹟：

- ✓ 97 學年度大專院校積體電路設計競賽大學部標準元件數位電路設計組「特優」
- ✓ 98 學年度大專院校積體電路設計競賽研究所標準元件數位電路設計組「設計完整獎」