國 立 交 通 大 學

電子工程學系　　電子研究所碩士班

碩 士 論 文

用於 UWB 設計之 Viterbi 解碼器

Viterbi Decoder Design for Ultra-Wide Band System.

研 究 生:蔡彥凱　　Yan-Kai Tsai

指導教授:溫壞岸 博士　Dr. Kuei-Ann Wen

中華民國九十五年六月

用於 UWB 設計之 Viterbi 解碼器

Viterbi Decoder Design for Ultra-Wide Band System

研 究 生:蔡彥凱　　Student：Yan-Kai Tsai

指導教授:溫瓌岸 博士　　Advisor：Dr. Kuei-Ann Wen

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 the Institute of Electronics

College of Electrical Engineering and Computer Science

National Chiao Tung University

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In

Electronic Engineering

June, 2006

HsinChu, Taiwan, Republic of China

中華民國 九十五年六月

誌　　謝

首先，第一個要感謝的是指導教授，溫瓌岸教授。感謝老師在兩年研究生涯中，不斷的給予彥凱指導與督促。溫老師的循循教誨，讓學生在學習訓練的路途上，能夠快速而正確的修正自己的研究方向，並且保持不鬆懈的心態進行研究。也感謝 TWT_LAB 在這兩年中提供的豐富研究資源，讓我在研究上無後顧之憂。

感謝實驗室的學長們的指導與照顧:彭嘉笙，溫文燊，林立協，莊源欣，周美芬，陳哲生，鄒文安。感謝兩年來一起打拚的同學:張懷仁，洪志德，賴俊憲，游振威，廖俊閔，張書瑋，卓彥宏。還有實驗室的學弟帶來的快樂時光:林義凱，莊翔琮，蘇建喻，吳家岱，梁書旗，李漢建，侯閔仁，蔡函霖。大家在生活上的互相扶持與鼓勵，讓原本辛苦煩悶的研究工作，也變的輕鬆愉快許多。同時也要感謝實驗室的助理:翁淑怡，楊怡倩，陳恩齊，陳慶宏，有妳們幫忙處理實驗室的雜務，才能讓我們能夠專心致力於研究。

最後，感謝默默支持我的母親以及哥哥。你們不斷的支持與鼓勵，讓我覺得更需要努力來回報你們。

Viterbi Decoder Design for Ultra-Wide Band System

Student: Yan-Kai Tsai Advisor: Dr. Kuei-Ann Wen

Department of Electronics Engineering Institute of Electronics

National Chiao-Tung University

Abstract

In this thesis, an IEEE 802.15.3a OFDM-based error correcting design and implementation is presented. With the newly proposed arithmetic compare-select (CS), the newly designed Viterbi decoder present good speed performance. According to IEEE 802.15.3a, the convolutional code 1/3 is the base coding rate. Through the puncture scheme, Viterbi decoder for the 802.15.3a standard can support several data rates. We analyzed the soft decision resolution and traceback-length to get the optimized solution between performance and complexity. The design flow and coding scheme is based on IP qualification. The coding style, code coverage up to 100% and other requirements are considered. Also, the macro design in CMOS.18 μm is applied with SYNOPSYS ASTRO

誌　　謝

首先，第一個要感謝的是指導教授，溫瓌岸教授。感謝老師在兩年研究生涯中，不斷的給予彥凱指導與督促。溫老師的循循教誨，讓學生在學習訓練的路途上，能夠快速而正確的修正自己的研究方向，並且保持不鬆懈的心態進行研究。也感謝 TWT_LAB 在這兩年中提供的豐富研究資源，讓我在研究上無後顧之憂。

感謝實驗室的學長們的指導與照顧:彭嘉笙，溫文燊，莊源欣，周美芬，陳哲生，鄒文安，林立協。感謝兩年來一起打拚的同學:張懷仁，洪志德，賴俊憲，游振威，廖俊閔，張書瑋，卓彥宏。還有實驗室的學弟帶來的快樂時光:林義凱，莊翔琮，蘇建喻，吳家岱，梁書旗，李漢建，侯閎仁，蔡函霖。大家在生活上的互相扶持與鼓勵，讓原本辛苦煩悶的研究工作，也變的輕鬆愉快許多。同時也要感謝實驗室的助理:翁淑怡，楊怡倩，陳恩齊，陳慶宏，有妳們幫忙處理實驗室的雜務，才能讓我們能夠專心致力於研究。

最後，感謝默默支持我的母親以及哥哥。你們不斷的支持與鼓勵，讓我覺得更需要努力來回報你們。

Contents

List of Tables

List of Figures

Figure 5.10: Verification plan……………………………………………………..56

Chapter 1.

Introduction.

1.1. Introduction to Ultra Wideband.

Ultra Wideband (UWB) is a wireless technology for transmitting digital data at very high rates over a wide spectrum of frequency bands using very low power. UWB is power efficient and suited for wireless communications, particularly short-range (generally within 10~20m) and high-speed data transmissions (53.3~480 Mb/s) for local area network applications. This technology has advantages of high speed enabling multimedia streaming in the home. [2]

1.2. Ultra Wideband physical layer (802.15.3a).

The UWB system that utilizes the unlicensed 3.1 ~ 10.6 GHz band. UWB system provides data payload communication capabilities of 53.3, 55, 80, 106.67, 110, 160, 200, 320, and 480 Mb/s, and UWB system employs orthogonal frequency division multiplexing (OFDM). The system uses a total of 122 sub-carriers that are modulated using quadrature phase shift keying (QPSK). Forward error correction coding (convolutional coding) is used with a coding rate of 1/3, 11/32, ½, 5/8, and ¾. The system also utilizes a time-frequency code (TFC) to interleave coded data over 3 frequency bands. Table 1.1 shows the rate-dependent parameters in each data rate. [1]

Table 1.1 Rate-dependent parameters. [1]

Data Rate (Mb/s)	Modulation	Coding rate (R)	Conjugate Symmetric Input to IFFT	Time Spreading Factor	Overall Spreading Gain	Coded bits per OFDM symbol (N_{CBPS})
53.3	QPSK	1/3	Yes	2	4	100
55	QPSK	11/32	Yes	2	4	100
80	QPSK	½	Yes	2	4	100
106.7	QPSK	1/3	No	2	2	200
110	QPSK	11/32	No	2	2	200
160	QPSK	½	No	2	2	200
200	QPSK	5/8	No	2	2	200
320	QPSK	½	No	1 (No spreading)	1	200
400	QPSK	5/8	No	1 (No spreading)	1	200
480	QPSK	¾	No	1 (No spreading)	1	200

In table 1.2, it lists timing-related parameters. A OFDM symbol period is T_{SYM} = T_{CP} + T_{FFT} + T_{GI} =312.5 ns. T_{CP} is the circular prefix which is used in OFDM to mitigate the effects of multipath. The parameter T_{GI} is the guard interval duration. The 128-point IFFT/FFT period is 242.42 ns.

Table 1.2 Timing-related parameters. [1]

Parameter	Value
N_{SD}: Number of data subcarriers	100
N_{SDP}: Number of defined pilot carriers	12
N_{SG}: Number of guard carriers	10
N_{ST}: Number of total subcarriers used	122 (= N_{SD} + N_{SDP} + N_{SG})
Δ_F: Subcarrier frequency spacing	4.125 MHz (= 528 MHz/128)
T_{FFT}: IFFT/FFT period	242.42 ns (1/Δ_F)
T_{CP}: Cyclic prefix duration	60.61 ns (= 32/528 MHz)
T_{GI}: Guard interval duration	9.47 ns (= 5/528 MHz)
T_{SYM}: Symbol interval	312.5 ns (T_{CP} + T_{FFT} + T_{GI})

In table1.3, the RX-to-TX turnaround time shall be pSIFSTime which is equal to 32 OFDM symbol. The pSIFSTime includes the latency of the RF, PHY and MAC. The RX-to-TX turnaround time is related to the throughput of the system. If we can reduce the latency of PHY, we can increase the throughput of the system.

Table 1.3 PHY layer timing parameters.[1]

PHY Parameter	Value
pMIFSTime	6*T_{SYM} = 1.875 µs
pSIFSTime	32*T_{SYM} = 10 µs
pCCADetectTime	15*T_{SYM} = 4.6875 µs
pChannelSwitchTime	9.0 ns

1.3. OFDM overview.

OFDM technique is widely used in wireless communication nowadays because of its high-speed data transmission and effectiveness in combating multipath fading or narrowband interference in wireless communications. Orthogonal frequency division multiplexing(OFDM) is a multicarrier transmission technique, which divides the available spectrum into many subcarriers, each one being modulated by a low data rate stream. In a single carrier system, a single fade or interferer can cause the entire link to fail, but in multi-carrier system, only a small percentage of subcarriers will be affected. Error correction coding can then be used to correct for the few erroneous subcarriers.[3]

The OFDM carriers exhibit orthogonality on a symbol interval if they are spaced in frequency exactly at the reciprocal of the symbol interval, which can be accomplished by utilizing the discrete Fourier transform (DFT). In eq.(1.1)[4] is a OFDM signal described by mathematical equation, where with N subcarriers and symbol duration is T, and notice that s(n) is the inverse Fourier Transform of the $x_i(n)$. In figure 1.1, it illustrates spectra of eq. (1.0); the spectrum of the individual carriers mutually overlap and the interference of adjacent channels is all zero.[4]

$$s(n) = \frac{A}{N}\sum_{i=0}^{N-1} x_i(n)\exp(2\pi f_i n), \quad \text{for } 0 \le n \le N; 0 \le i \le N$$

$$f_i = f_c + \frac{i}{T}, \quad i = 0,1,\ldots,N-1 \tag{1.1}$$

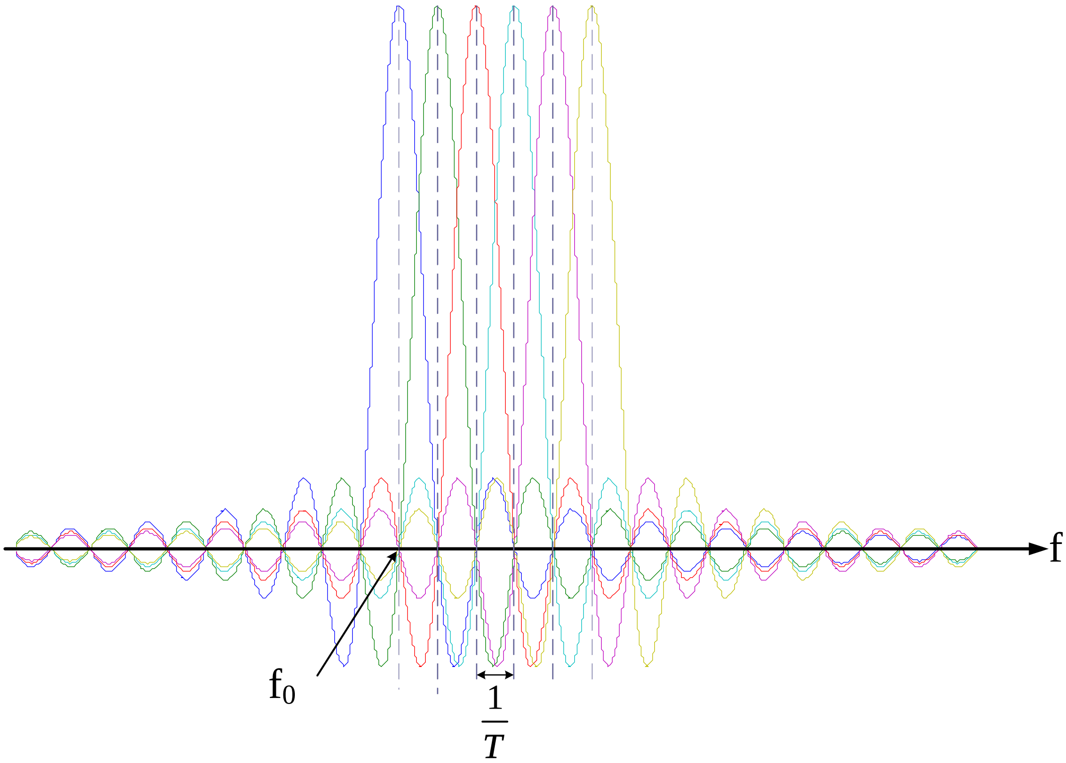

Figure 1.1Overlapping orthogonal carriers

The advantages of OFDM technique list as follows:

- Immunity to delay spread and multipath.
- OFDM is robust against narrowband interference.
- Simple equalization.
- Efficient bandwidth usage by overlapping carriers.

The disadvantages of OFDM technique are as follows:

- OFDM system is sensitive to carrier frequency offset and phase noise.
- OFDM system has relatively large peak to average power ratio.

1.4. Design and Implementation Issues

In UWB system, the high throughput reaching 480Mb/s is the major issue in hardware design. The ACS block has iterated operation, therefore we can't speed this block by pipeline technique. Here, we proposed a arithmetic compare and select (CS) circuit for speeding the critical path, and we discusses it in chapter 3.

The secondary issue is the trade off between performance and hardware complexity. The soft decision algorithm, and the traceback length of Viterbi decoder decide the performance. And we discuss the trade off in chapter 4.

1.5. Organization of this thesis.

This thesis is organized as follows: The first chapter describes a briefly introduction of UWB. In chapter 2, the specification of IEEE 802.15.3a relative to error correction coding and the system requirements will be presented. In Chapter 3, the reduction of proposed CS circuit and analysis of add-compare-select (ACS) will be described. Chapter 4 describes the design of the Viterbi decoder, including quantization scheme ,de-puncture and Viterbi decoder, respectively. And it also shows the simulation result. Chapter 5 shows the achievement of IPQ and FPGA porting. Finally, a brief conclusion and future work are presented in chapter 6.

Chapter 2

Viterbi Decoder for Ultra-Wideband

2.1 Design Requirements of Viterbi Codec for UWB

The frame format of IEEE 802.15.3a WLAN standard has preamble, header, payload, and inserted data. The header is always sent at an information data rate of 53.3 Mb/s, and the remainder of the frame is sent at the desired information data rate of 53.3, 55, 80, 106.7, 110, 160, 200, 320, 400 or 480 Mb/s [1]. The information is encoded by scrambler, convolution encoder and interleaver. Besides, the different data rate varies with different puncture scheme.

2.1.1 Scrambler

The frame synchronous scrambler uses the generator polynomial S(x) as follows, and is illustrated in Fig 2.1:

$$S(x) = 1 + D^{14} + D^{15} \tag{2.1}$$

In the receiver, we can use the same scrambler structure to descramble the received data.

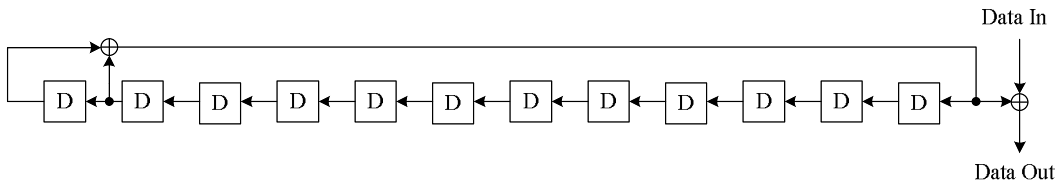

Figure 2.1: Scrambler

2.1.2 Convolution Encoder

The convolution encoder of transmitter provide coding rate r=1/3 and constraint length K=7. The generator polynomials of $G_A(D)$, $G_B(D)$ and $G_C(D)$ as follows are illustrated in Fig 2.2. Besides general coding rate above, other rates are derived from "puncturing" methodology. [1][7].

$$G_A(D)=1+D^2+D^3+D^5+D^6 \tag{2.2}$$

$$G_B(D)=1+D+D^4+D^5 \tag{2.3}$$

$$G_C(D)=1+D+D^2+D^3+D^4+D^6 \tag{2.4}$$

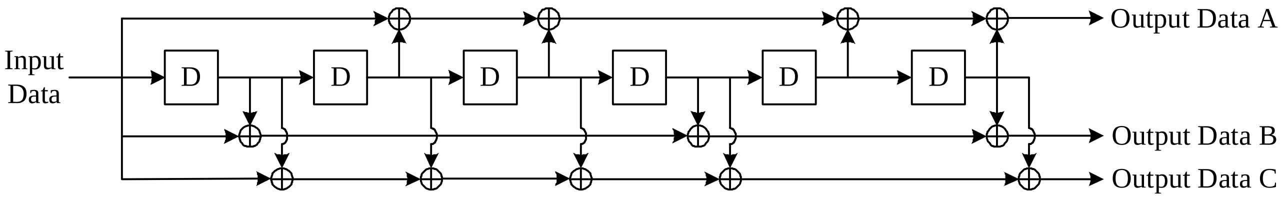

Figure 2.2: (3, 1, 7) convolution encoder

2.1.3 Puncture

Puncturing is a procedure for stealing some of the encoded bits in the transmitter. The coding rate varies with puncture scheme by stealing different transmitted bits. Figure 2.3 depicts the puncture procedure. De-puncture scheme is inserting a dummy “zero” metric instead of the deleting bit on the decoding side [7] [9]. By combining time spreading and conjugate symmetric input to IFFT and different coding rates, IEEE 802.15.3a supports ten different data rates.

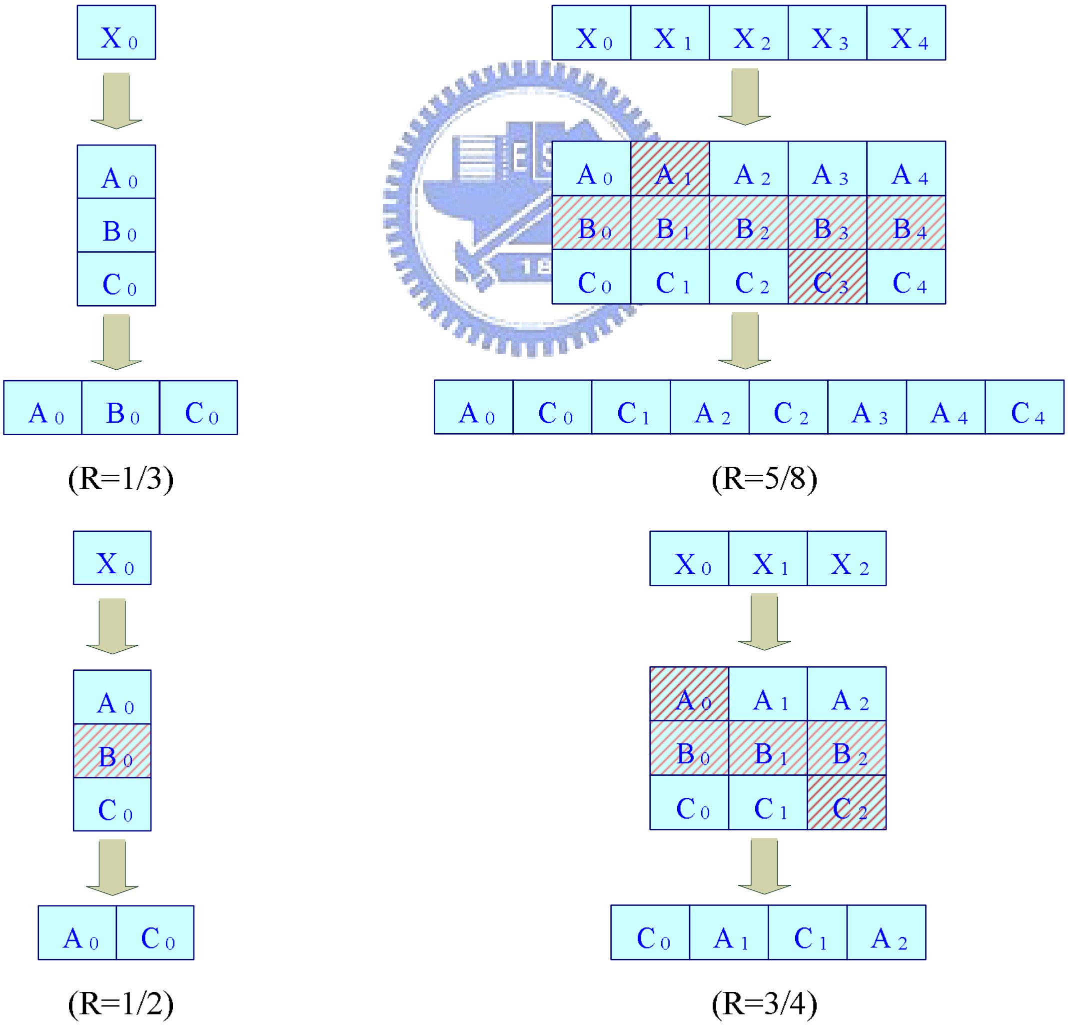

Figure 2.3: Puncture procedure

2.1.4 Interleaving

Bit interleaving provides robustness against burst errors. The bit interleaving operation is performed in two stages: symbol interleaving followed by tone interleaving. The symbol interleaver permutes the bits across OFDM symbols to exploit frequency diversity across the sub-bands, while the tone interleaver permutes the bits across the data tones within an OFDM symbol to exploit frequency diversity across tones and provide robustness against narrow-band interferers [1].

The input-output relationship of the first permutation shall be given by:

$$S(i) = U\left\{\mathrm{Floor}\left(\frac{i}{N_{CBPS}}\right) + 6\mathrm{Mod}(i, N_{CBPS})\right\} \tag{2.5}$$

The function floor (.) denotes the largest integer not exceeding the parameter, and the function Mod(i,N_{CBPS}) is the remainder of N_{CBPS} where N_{CBPS} is the number of coded bits per OFDM symbol. Figure 2.4 illustrates the permutation of block interleaver. The input-output relationship of the second permutation is given by:

$$T(i) = S\left\{\mathrm{Floor}\left(\frac{i}{N_{T\,\mathrm{int}}}\right) + 10\mathrm{Mod}(i, N_{Tint})\right\} \tag{2.6}$$

The value N_{Tint} is $N_{CBPS}/10$ in equation (2.6). Figure 2.5 illustrates the permutation of tone interleaver.

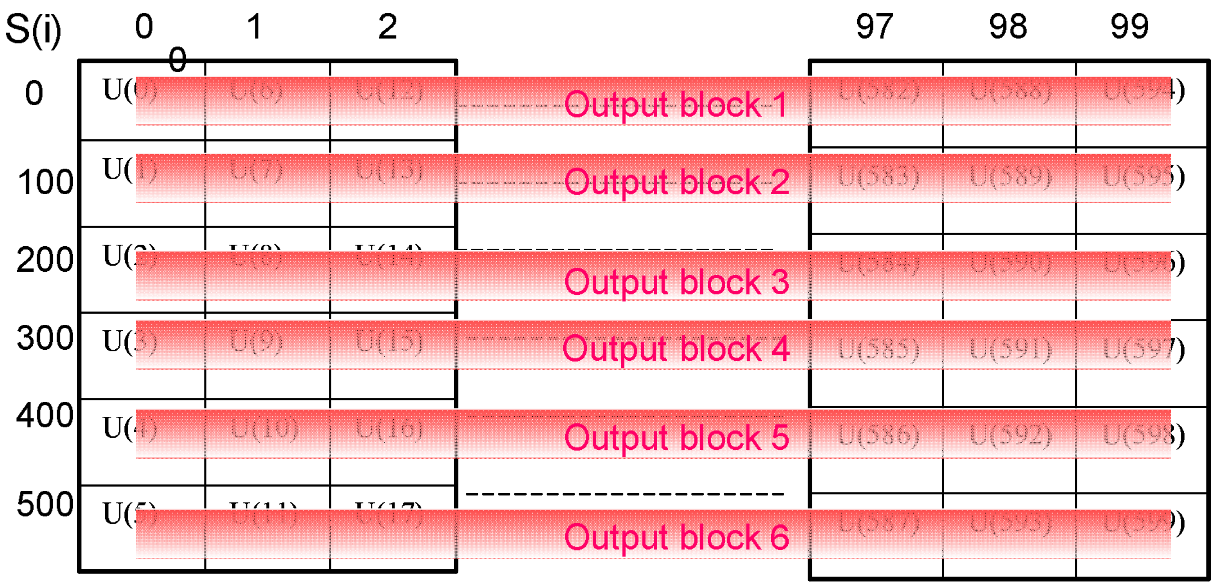

Figure 2.4: Block Interleaver

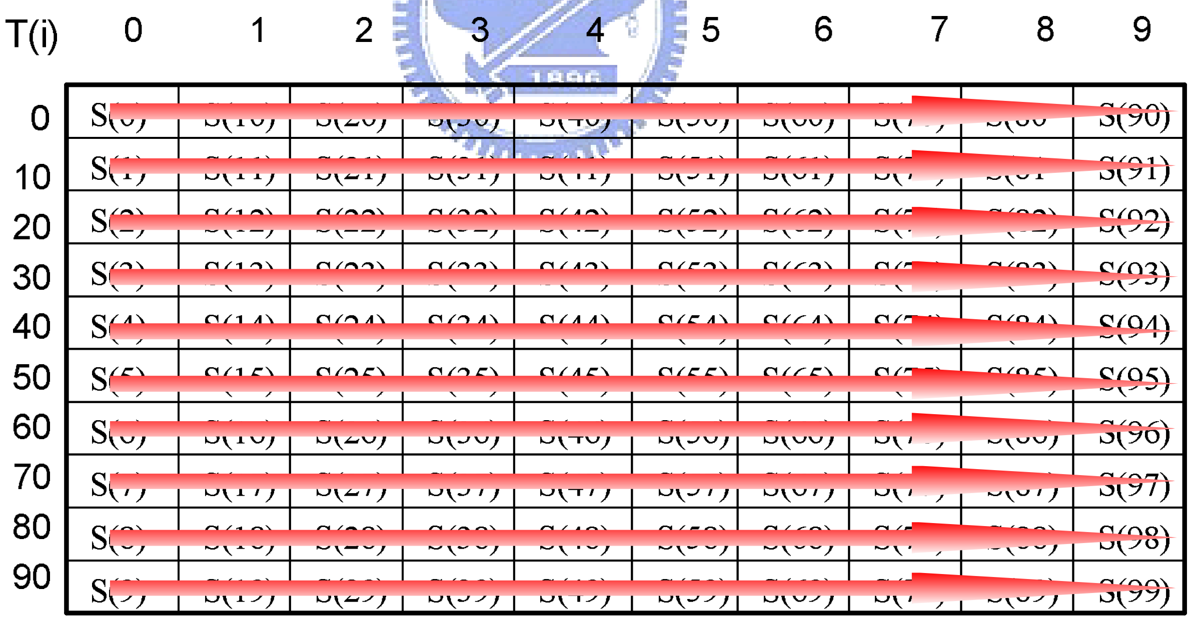

Figure 2.5: Tone Interleaver

2.2 Viterbi Decoder Algorithm

Viterbi Decoder is a maximum likelihood decoder. It finds the closest coded sequence to the received sequence by processing the sequences on an information bit-by-bit (branch of the trellis) basis. Generally, Viterbi decoder has four major decoding steps: branch metric computation, Add-Compare-Select (ACS), path memory update, and decode symbols. The example is given as Fig 2.6. The trellis (2,1,2) is shown in Fig 2.6(a) and each state has two connected path. In Fig 2.6(b), the received data "11" is shown in Fig 2.6(b) and the branch metric is calculated by comparing with the referenced metric. Each state at t_1 selects the minimum path metric and the information of survivor path is plotted with arrowheads. Fig 2.6(c) and Fig 2.6(d) continue the operations of add-compare-select at t_2 and t_3. After all the information of survivor path is found, the operation of traceback starts at the minimum path metric. In Fig 2.6(e), the minimum state at t_3 is state zero and the traceback starts at this state. Then, the survivor path is $\{S0_{t3}, S2_{t2}, S1_{t1,} S0_{t0}\}$ and the survivor path is plot with thick arrowheads. With decoding scheme, the upper path is decoded as zero and the lower path is decoded with one. Hence, the received sequence is decoded with{1,0,0}.

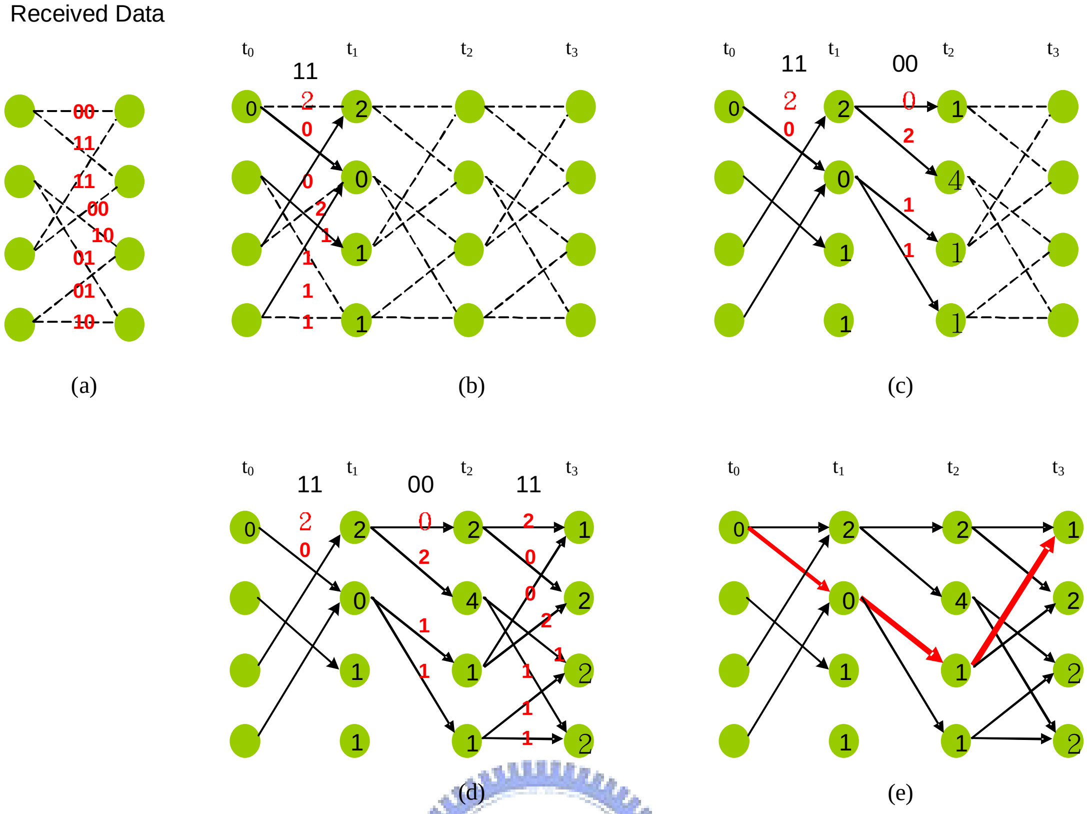

Figure 2.6: Example for Viterbi Algorithm

In this example above, the received sequence is hard-decision shown in Fig 2.7. The other case is soft-decision shown in Fig. 2.8.

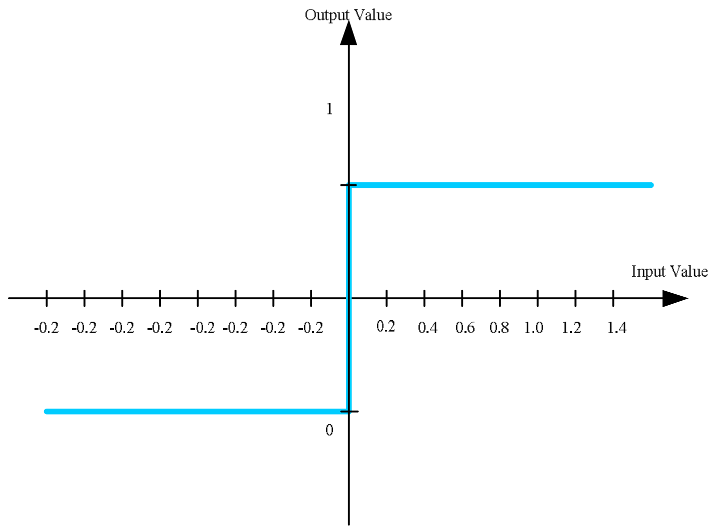

Figure 2.7: Hard Decision

The soft decision quantizes the sequence from channel and increases the error correcting capability. Figure 2.8 illustrates the soft -decision quantization. The transmitted sequence is transmitted in “0” and “1” and the input sequence added with channel ranges between -∞ and ∞. The positive value is strong one when it is bigger. On the contrary, the negative value is strong zero when it is bigger.

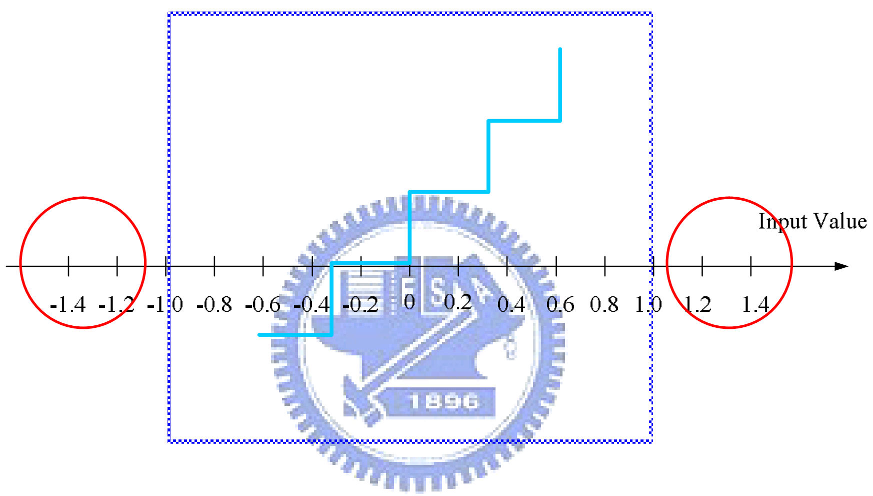

Figure 2.8: Soft Decision

Figure 2.9(a) depicts the radix-2 trellis diagram for the convolution encoder in 802.15.3a standard. It can be transformed into radix-4 trellis diagram shown as Fig. 2.9(b). The high-radix Viterbi Decoder increases the throughput by processing two stages of the constituent radix-2 trellis per iteration [10]. Furthermore, we combine two radix-4 trace-back iterations into a single radix-16 iteration for the throughput of Ultra Wideband standard.

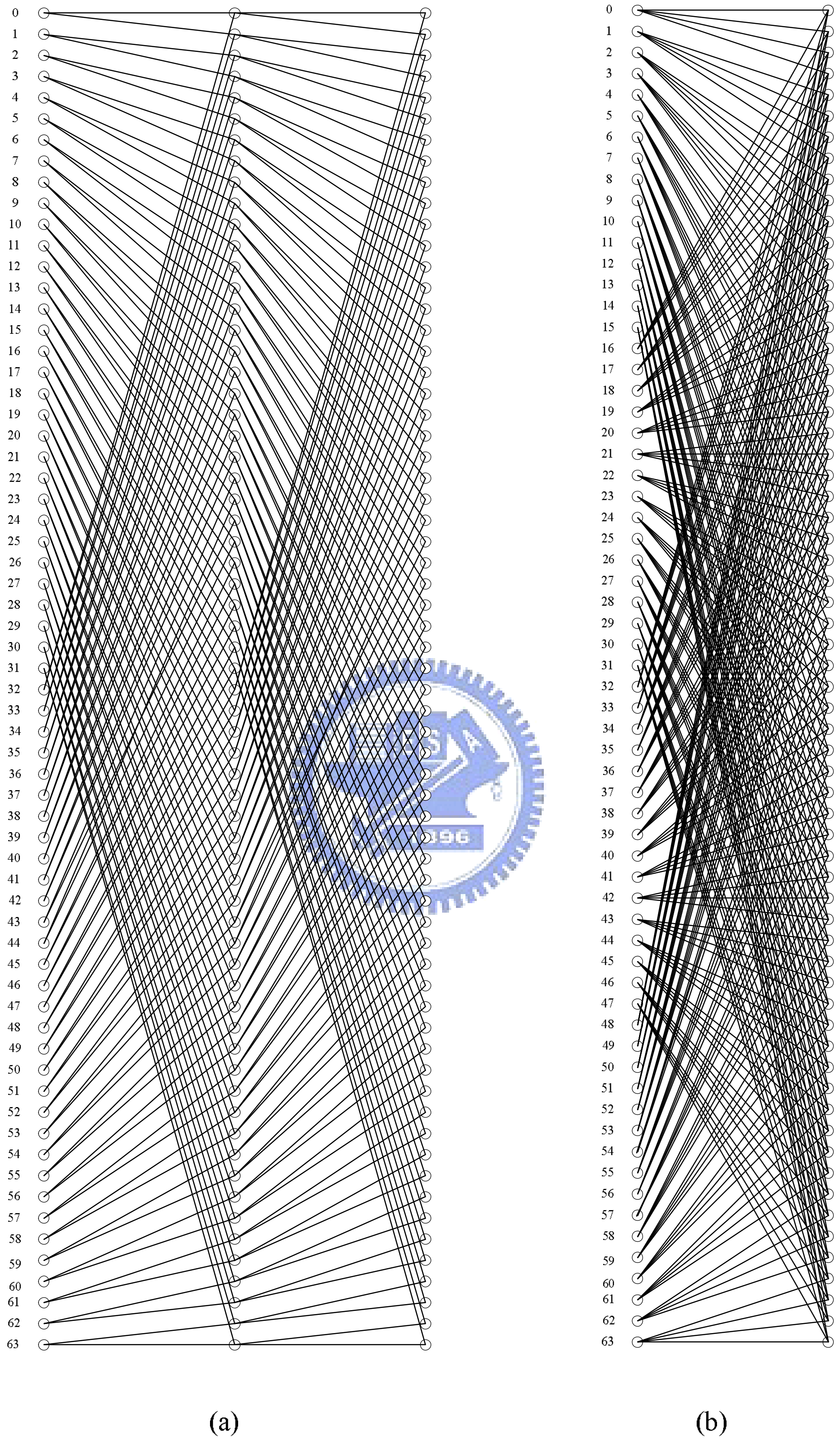

Figure 2.9: Trellis Diagram of Convolution Encoder for 802.15.3a Standard

We compute the branch metric by the specified trellis and complete the operation of ACS. Then, the survivor paths can be updated to the memory. After the memory is filled with the survivor information, the received sequences find its likelihood decoding path by trace-back method shown as Fig. 2.10 [9].

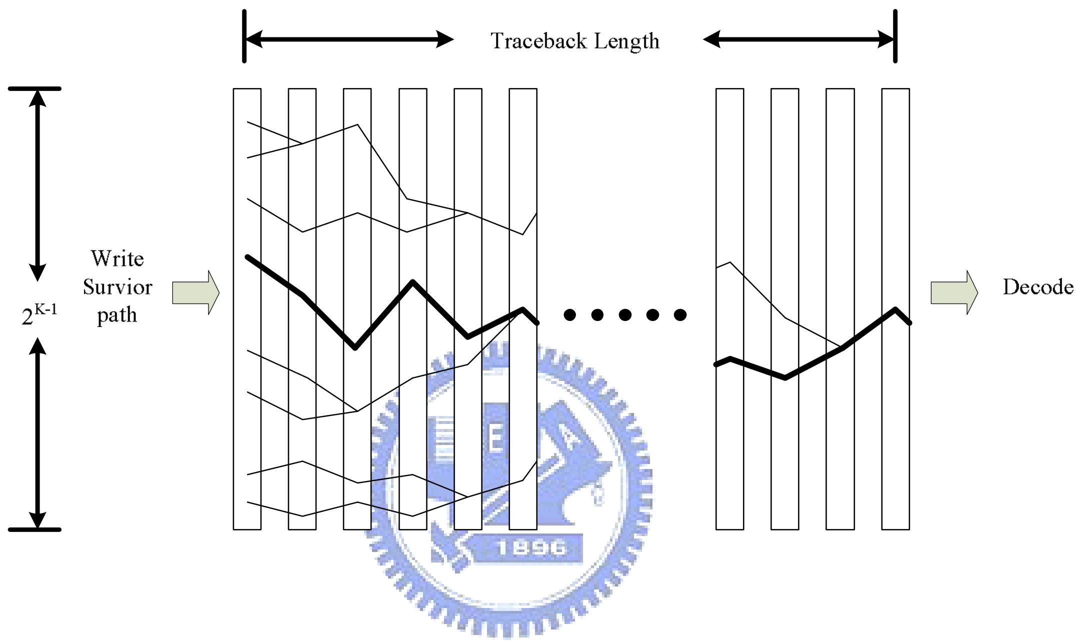

Figure 2.10: Trace-back Diagram for Finding Maximum Likelihood Path

Chapter 3

ACS module with Arithmetic CS unit

3.1 Deduction of Arithmetic CS unit

Function of compare and select is to find the maximum or minimum value of the input values. Let the input data as V={v_1,v_2,v_3…… v_m } where m is the number of input data and E={{v_1,v_2},{v_1,v_3}……{v_i,v_j}} where i≠j is a set of two-element subsets of V. The members in V are called vertices and the members in E are called edges in graph theory [5]. Generally the maximum value by this method is depicted below.

$$M_1 = \{m_1, m_2 ......m_{\frac{m}{2}}\} = \{\max\{v_1, v_2\}, \max\{v_1, v_3\}......, \max\{v_i, v_j\}\} \qquad i \neq j, \{i, j\} \in \{0,1,2......m\}$$

$$M_2 = \{m_1^2, m_2^2 ......m_{\frac{m}{4}}^2\} = \{\max\{m_1, m_2\}, \max\{m_1, m_3\}......, \max\{m_i', m_j'\}\} \qquad i \neq j, \{i, j\} \in \{0,1,2......m\}$$

$$\vdots$$

$$M_{\log_2 m} = \{m_{\max}^{\log_2 m}\} = \{\max\{m_1^{\log_2 m-1}, m_2^{\log_2 m-1}\}\} \tag{3.1}$$

In equation 3.1, the subsets M_i^j in M_i have the information of maximum value from the subsets $\max\{m_k^{j-1}, m_l^{j-1}\}$ where k≠l and the number of the maximum value in V. We can find that the maximum value of the compared data $\{v_1, v_2, v_3 \ldots\ldots v_m\}$ can be defined as $\log_2 m$ times of input and selecting in equation (3.1).

We define an ordered pair (v_i, v_j) where $i \neq j$ and it means v_i is greater than v_j. For each member of E we define an ordered pair, and we let R be the set of all such ordered pairs, and R is indicated in equation 3.2.

$$R=\{r_1, r_2, r_3 \ldots\ldots, r_{\frac{m}{2}}\}=\{(v_1,v_2),(v_1,v_3)\ldots\ldots(v_i,v_j)\} \qquad where \quad i \neq j \tag{3.2}$$

In this deduction of the proposed method, we list the combinations of elements and find the maximum value in different combinations. Finally, we can conclude a logical equation by Boolean.

For example, we give a V with four elements and V= $\{v_1, v_2, v_3, v_4\}$. Therefore, it has $C_2^4 = 6$ edges and the edges E={$\{v_1,v_2\}$,$\{v_1,v_3\}$,$\{v_1,v_4\}$,$\{v_1,v_2\}$,$\{v_1,v_3\}$},$\{v_1,v_4\}$} are shown in Fig. 3.1.

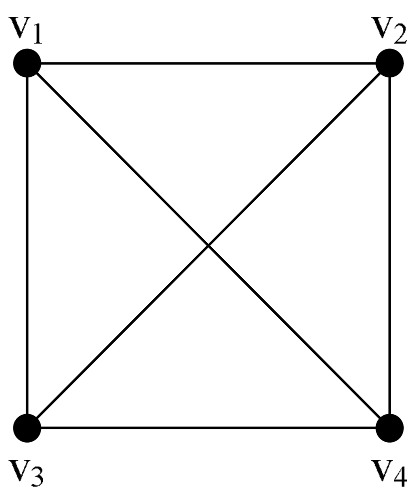

Figure 3.1: Complete Graph on 4 Vertices

Table 3.1: Relations Definition

Representation	Relation	Representation	Relation
$r_1=1$	(v_1,v_2)	$r_1=0$	(v_2,v_1)
$r_2=1$	(v_1,v_3)	$r_2=0$	(v_3,v_1)
$r_3=1$	(v_1,v_4)	$r_3=0$	(v_4,v_1)
$r_4=1$	(v_2,v_3)	$r_4=0$	(v_3,v_2)
$r_5=1$	(v_2,v_4)	$r_5=0$	(v_4,v_2)
$r_6=1$	(v_3,v_4)	$r_6=0$	(v_4,v_3)

In table 3.1 above, we use relations of R where $R=\{r_1,r_2,r_3,r_4,r_5,r_6\}$ to represent the ordered pairs. When v_1 is greater than v_2, r_1 is equal to zero and v_2 is greater than v_1 when r_1 is equal to one. Therefore we have total 2^6 kinds of the vector $R=\{r_1, r_2\ldots\ldots, r_6\}$.

In this deduction of the proposed method, we have two conditions. One is the maximum value existing, and the other is not.

For example, the conditions are $R=\{r_1,r_2,r_3,r_4,r_5,r_6\}=\{1,1,1,1,1,1\}$ in Fig 3.2. We can find that all of the arrowheads come from v_1. It means v_1 is greater than v_2 and v_3 and v_4. Therefore the maximum value is v_1 by logical reasoning.

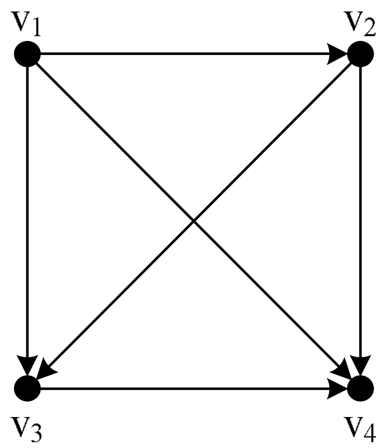

Figure 3.2: Directed Graph with Maximum Value

The second condition can be illustrated with the directed graph in Fig 3.3, and the relation is $R=\{r_1,r_2,r_3,r_4,r_5,r_6\}=\{1,1,0,1,1,1\}$. We infer that in case of the maximum value being not existing, it forms non radiated vertices. Furthermore, there is a clockwise loop in $\{v_1, v_2, v_3\}$. The vertices in loops will be an unlogical case with no being appeared maximum value.

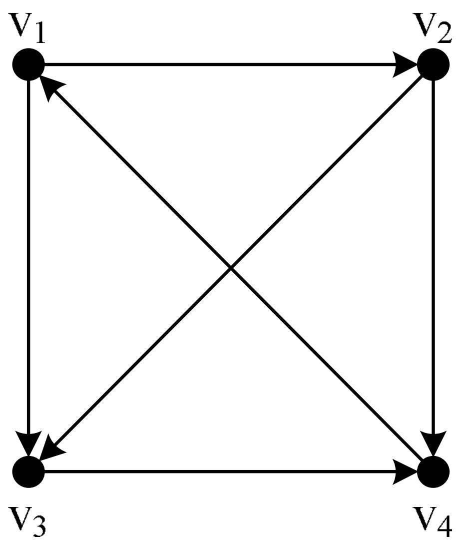

Figure 3.3: Directed Graph without Maximum Value

After we analyzed all the 2^6 conditions, we can list a table as follows. The symbols of v_i corresponds to the 6-bit R is exactly the maximum value among {v1, v2, v3, v4} and symbol 'x' denotes that there is no maximum value can be identified.

Table 3.2: All relations with four input data

r_1 r_2 r_3 r_4 \ r_5 r_6	**00**	**01**	**10**	**11**
0 0 0 0	v_4	v_3	v_4	v_3
0 0 0 1	x	v_3	x	v_3
0 0 1 0	v_4	x	v_4	x
0 0 1 1	v_2	v_2	v_2	v_2
0 1 0 0	x	v_3	x	v_3
0 1 0 1	v_3	v_3	v_3	v_3
0 1 1 0	x	x	x	x
0 1 1 1	v_2	v_2	x	x
1 0 0 0	v_4	x	v_4	x
1 0 0 1	x	x	x	x
1 0 1 0	v_4	v_4	v_4	v_4
1 0 1 1	v_2	v_2	x	x
1 1 0 0	v_1	v_1	v_1	v_1
1 1 1 0	x	x	v_1	v_1
1 1 0 1	x	x	v_1	v_1
1 1 1 1	v_2	v_2	v_1	v_1

We can directly select the maximum value with the known relations from r_1 to r_6 by using table 3.2.

Finally we can use Boolean reduction on the logical information in table 3.3 and derived the logical equation as equation 3.3 and equation 3.4. These equations mean we can get the result for selecting the maximum value by knowing r_1 to r_6.

$$SEL[0] = \overline{r_2}\,\overline{r_4}\,\overline{r_6} + r_1\overline{r_2}\,\overline{r_4} + \overline{r_2}r_3 + \overline{r_1}r_3r_4 + r_1r_3r_4\overline{r_5} \tag{3.3}$$

$$SEL[1] = \overline{r_2}\,\overline{r_4} + \overline{r_1}\,\overline{r_3} \tag{3.4}$$

The maximum value is v_4 when SEL[0] and SEL[1] are equal to zeros, and the maximum value is v_3 when SEL[0] is equal to one and SEL[1] are equal to zero. The maximum value is v_2 when SEL[0] is equal to zero and SEL[1] are equal to one, and so on. If we want to change the order, we just exchange the orders of the input data.

Table 3.3: Conditions of the Maximum Value

Max vaule $= v_1$					
r_1	r_2	r_3	r_4	r_5	r_6
1	1	0	0	x	x
1	1	0	1	x	x
1	1	1	0	x	x
1	1	1	1	1	x

Max vaule $= v_2$					
r_1	r_2	r_3	r_4	r_5	r_6
0	0	1	1	x	x
0	1	1	1	x	x
1	0	1	1	x	x
1	1	1	1	0	x

Max vaule $= v_3$					
r_1	r_2	r_3	r_4	r_5	r_6
0	0	0	0	x	x
0	0	0	1	x	x
0	1	0	0	x	x
0	1	0	1	x	x

Max vaule $= v_4$					
r_1	r_2	r_3	r_4	r_5	r_6
0	0	0	0	x	x
0	0	1	0	x	x
1	0	0	0	x	x
1	0	1	0	x	x

3.2 Arithmetic CS Circuit Analysis

We apply the proposed arithmetic CS circuit to Viterbi decoder. In Viterbi decoder we can not use pipeline technique to speed up the critical block because of iterated operations. Timing requirement is the main problem of high throughput of 480 Mb/s specified for Ultra-Wideband standard. Hence increasing the throughput or reducing the delay of critical path will be the key issue of the design.

Firstly, we compare the path delay of CS circuit compared with traditional comparator. For example, with the number of input data be four and the resolution is seven. The path delay simulation is done with design compiler (synopsys) and use UMC 0.18 library for timing analysis.

Figure 3.4 shows the traditional comparator. It has three adders and three multiplexers, and the critical path goes through two adders and two multiplexers.

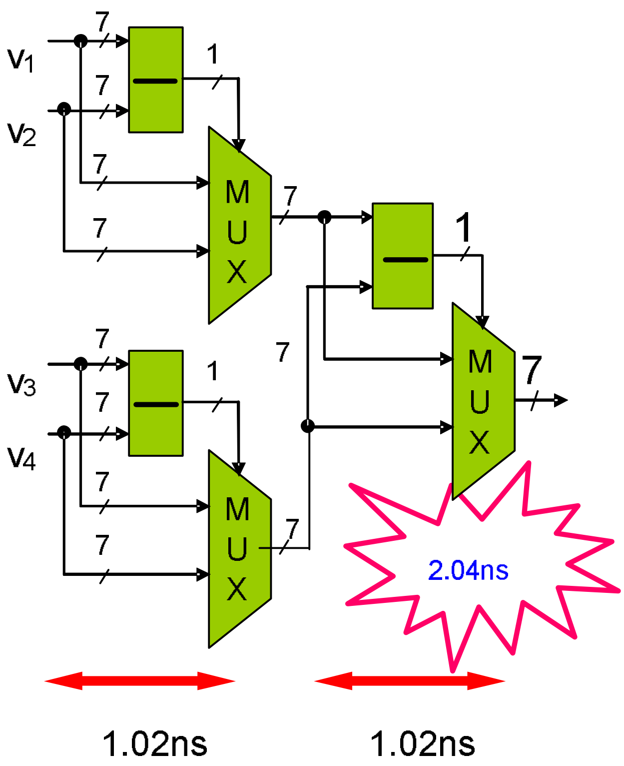

Figure 3.4: Path Delay of General CS unit

Figure 3.5 shows the architecture of proposed CS unit. The critical path goes through one adder and one multiplexer and one combinational block with fixed path delay. As the input resolution increases the path delay of adder and multiplexer increase, but the combinational block still has the fixed path delay. Therefore, the benefits to the path delay in the proposed CS unit increases as the circuit resolution increase. We can see the trend in Fig 3.6.

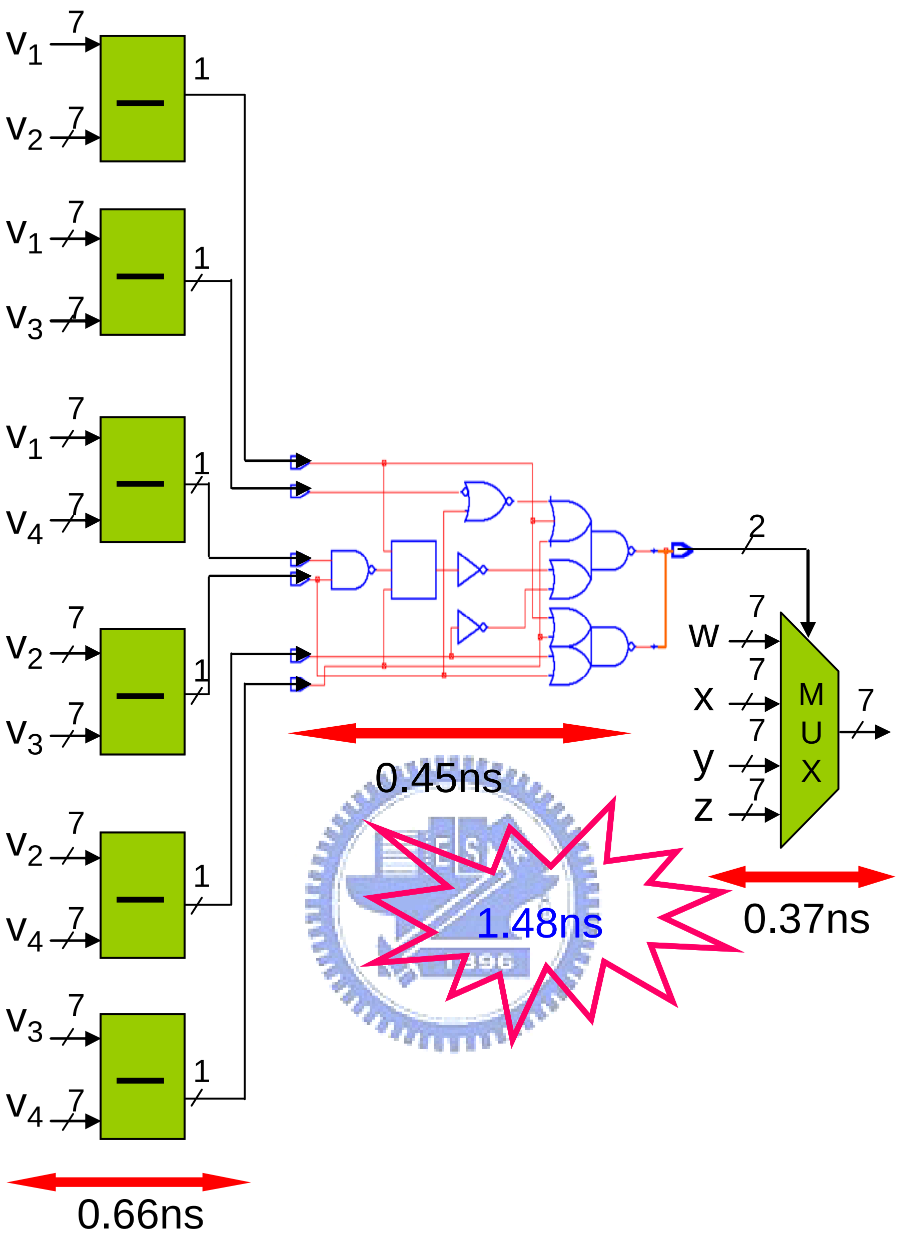

Figure 3.5: Path Delay of Proposed CS unit

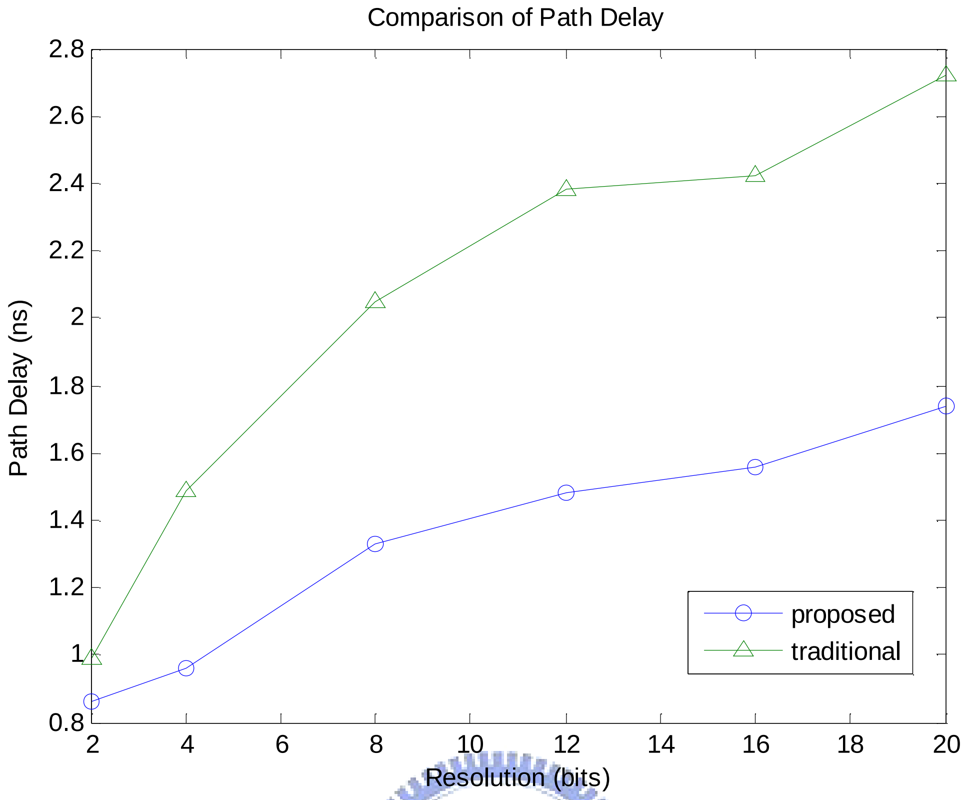

Figure 3.6: Comparison of Path Delay between the proposed CS and traditional CS

Simultaneously, we analyze the drawback of the input architecture. The complexity of the proposed CS circuit increases as the number of compared data increases. The numbers of adders increase with C_2^n where n is the resolution bits and table 4 shows the complexity.

Table 3.4: Complexity of CS unit

Number of input Data	2	4	8	16
Tradition	1 adder 1 mux	3 adders 3 2-to-1muxs	7 adders 7 2-to-1muxs	15 adders 15 2-to-1muxs
Arithmetic	1 adder 1 mux	6 adders 1 4-to-1muxs	28 adders 1 8-1muxs	120 adders 1 16-to-1muxs

In the table 3.4, the four-input arithmetic CS has the optimized architecture by comparing with higher input architecture. The optimized architecture is applied to the radix-4 Viterbi codec design and the speed issue is the first consideration. Furthermore, we analyze the speed, area and power with the arithmetic CS and general CS. The comparison of gate counts is listed in Fig. 3.7. This trend of arithmetic CS becomes bigger than general CS because the gate count of adders increases as the resolution increased and the number of arithmetic CS has three more than general CS.

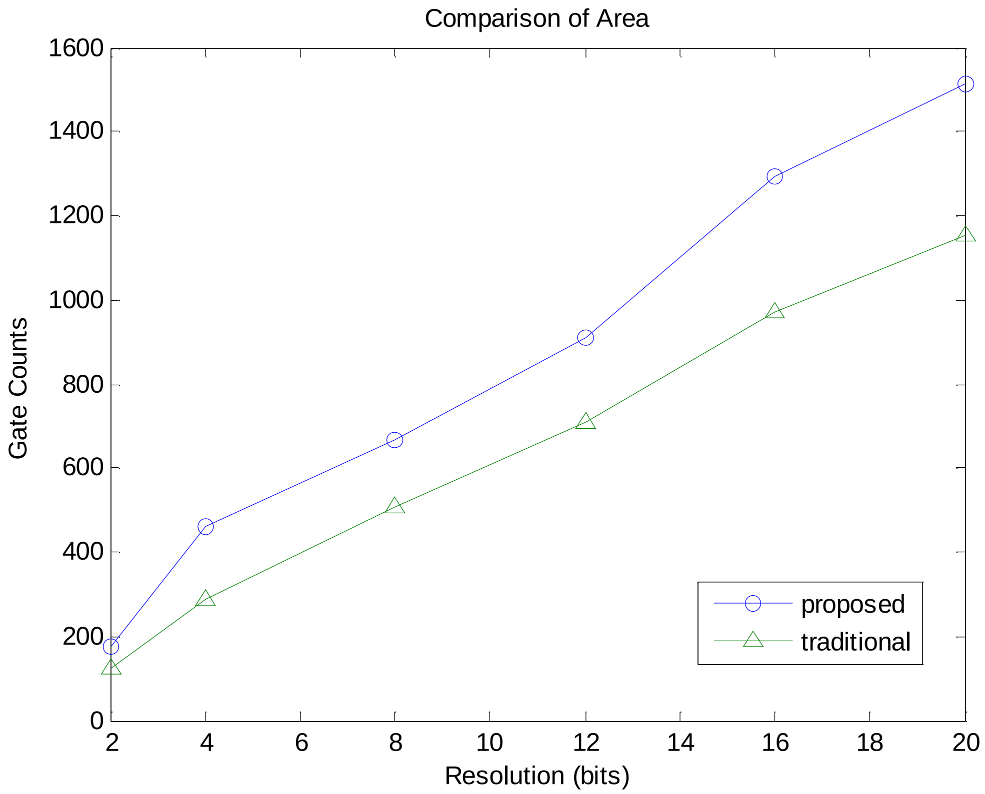

Figure 3.7: Comparison of Area between the proposed CS and the traditional CS

Figure 3.8 illustrates the comparison between the arithmetic CS and the traditional CS. The trend of power generally can reference from the trend of area.

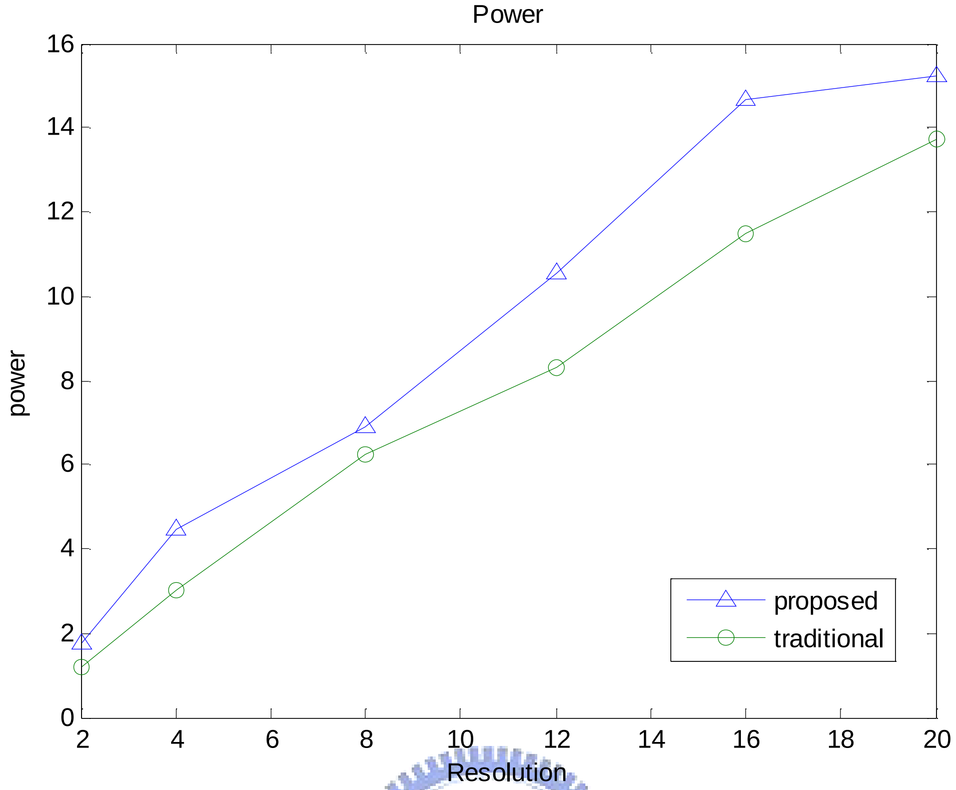

Figure 3.8: Comparison of Power between the proposed CS and the traditional CS

Figure 3.9 illustrated the ratio of figure of merit defined by $\frac{1}{AT^2}$ which A is area and T is the path delay. All the area and timing analysis is run by SYNOPSYS design compile with UMC.18 μ m library. The curve below depicts the proposed design is roughly 1.8 times better than the general design. Besides, the FoM goes down in lower resolution bits because the benefit from the path delay of adder decreases.

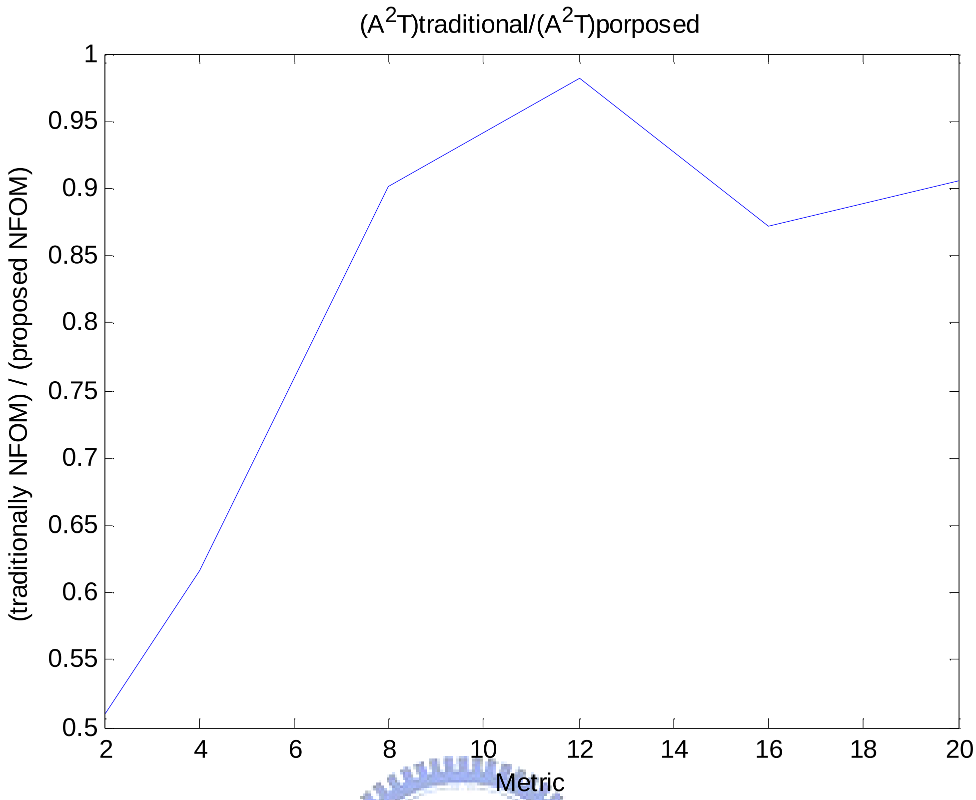

Figure 3.9: Ratio of Figure of merit defined by $\dfrac{AT^2{}_{trad}}{AT^2{}_{proposed}}$

3.3 ACS module with Arithmetic CS Circuit

For Ultra-Wideband standard, the high-throughput issue is the main constraint.

Viterbi decoder's critical path is in Add Compare Select (ACS) block. The timing limitation comes from feed-back loop. It causes a limit of pipelining data process. Therefore, how to speed up the add-compare-select block is what we will discuss.[2]

The basic function block of ACS block is called the radix-2 ACS unit. We take the trellis diagram of four states as an example in Fig 3.10. $\Gamma_{t-1,s0}$ is the previous survival metric, and $l_{t,s0->s0}$ is the branch metric from state 0 to state 0. $\Gamma_{t,s0}$ is the current metric of the ACS unit.

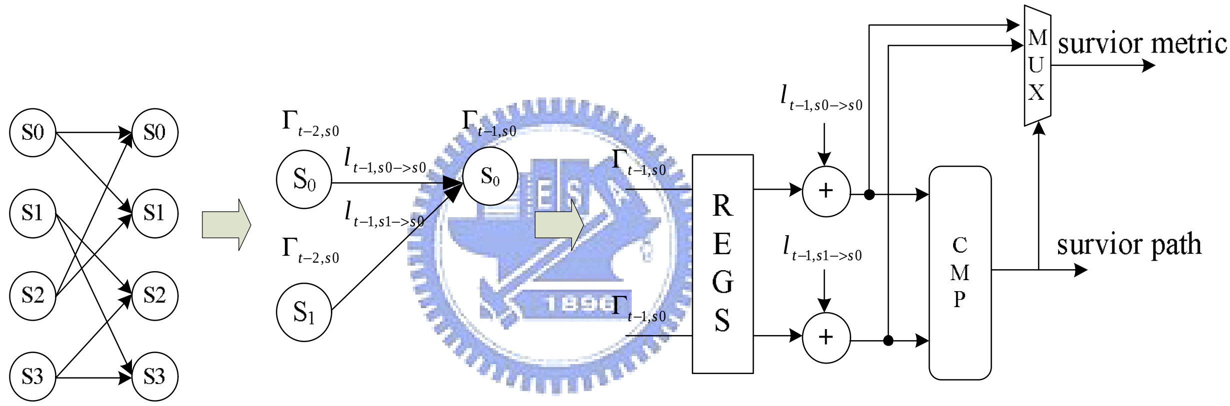

Figure 3.10: Radix-2 ACS trellis diagram and its function unit

The main consideration of ACS architecture design is the trade off between the decoder throughput and the number of ACS stage. Several kinds of the ACS architecture are proposed to achieve the different applications. For low throughput applications, we can use serial architecture completing the same decoding operations with more clock cycles instead. But for high data rate applications, we increase the throughput with the cost of high complexity hardware. It is well known that high radix ACS unit is proposed to improve the decoding throughput [6] [7]. Actually, the high radix ACS concept is to decode two or more symbols each time.

Figure 3.11 depicts the conversion from two-stage radix-2 ACS unit to one-stage radix-4 ACS unit. The radix-4 architecture completes two-stage ACS operations instead of one-stage ACS operation.

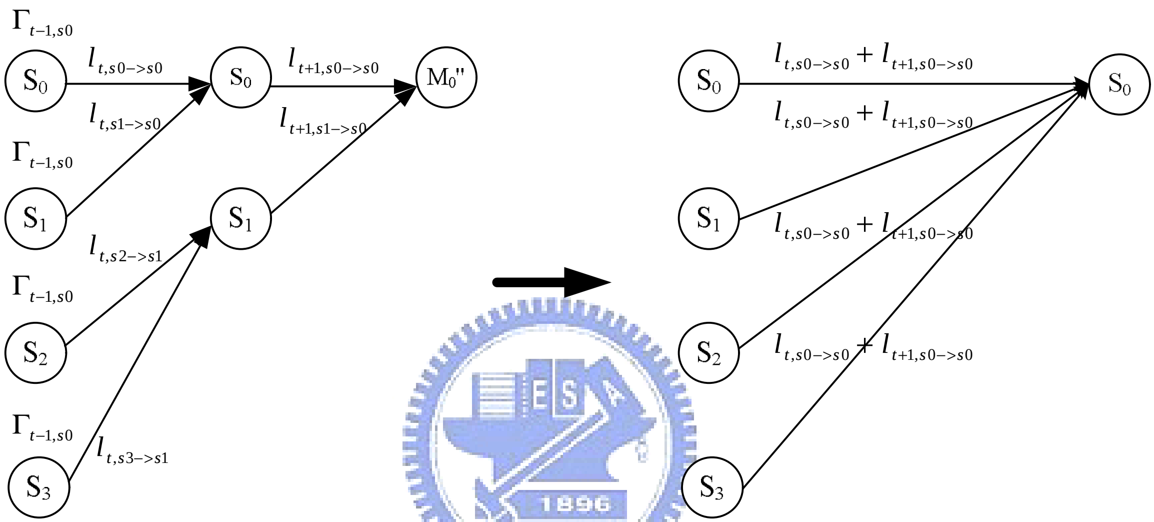

Figure 3.11: The Conversion from radix-2 to radix-4

Table 3.5 illustrates the complexity with different number of radix. We select radix-4 as the basic ACS unit because of some improved techniques and acceptable cost.

Table 3.5: Complexity of ACS unit

	General comparator		Arithmetic CS	
Radix Complexity	Add operations for branch metric	Comparator operations	Add operations for branch metric	Comparator operations
2	2	1	2	1
4	8	3	8	6
8	24	7	24	28
16	64	15	64	120

In radix-4 ACS unit, we can move the operations of $l_{t-1} + l_t$ to the block of branch metric. Therefore, ACS block reduces the path delay of one adder in the critical path and doubles the throughput by high radix architecture.

Figure 3.12 illustrates the conversion from the radix-4 ACS unit with two adders to the radix-4 ACS unit with one adder in the critical path.

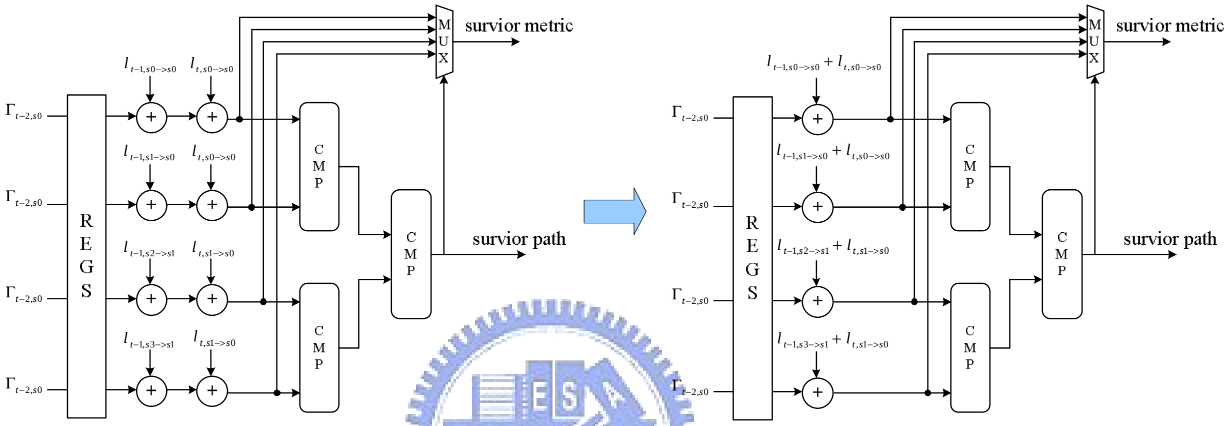

Figure 3.12: The Conversion from the architecture with eight adders to the architecture with four adders

Furthermore, we additionally add the Arithmetic CS circuit to the radix-4 ACS unit and the modified radix-4 ACS is illustrated in Fig. 3.13. In this step, the costs only come from the property of the proposed CS unit.

Finally, we reduce four times of clock rate for implementation. With consideration of the trade off between throughput and complexity, we adopt two-stage radix-4 ACS architecture instead of radix-16 ACS block for Ultra-Wideband standard. The conversion of four-stage radix-2 trellis diagram to two-stage radix-4 trellis diagram is illustrated in Fig 3.14.

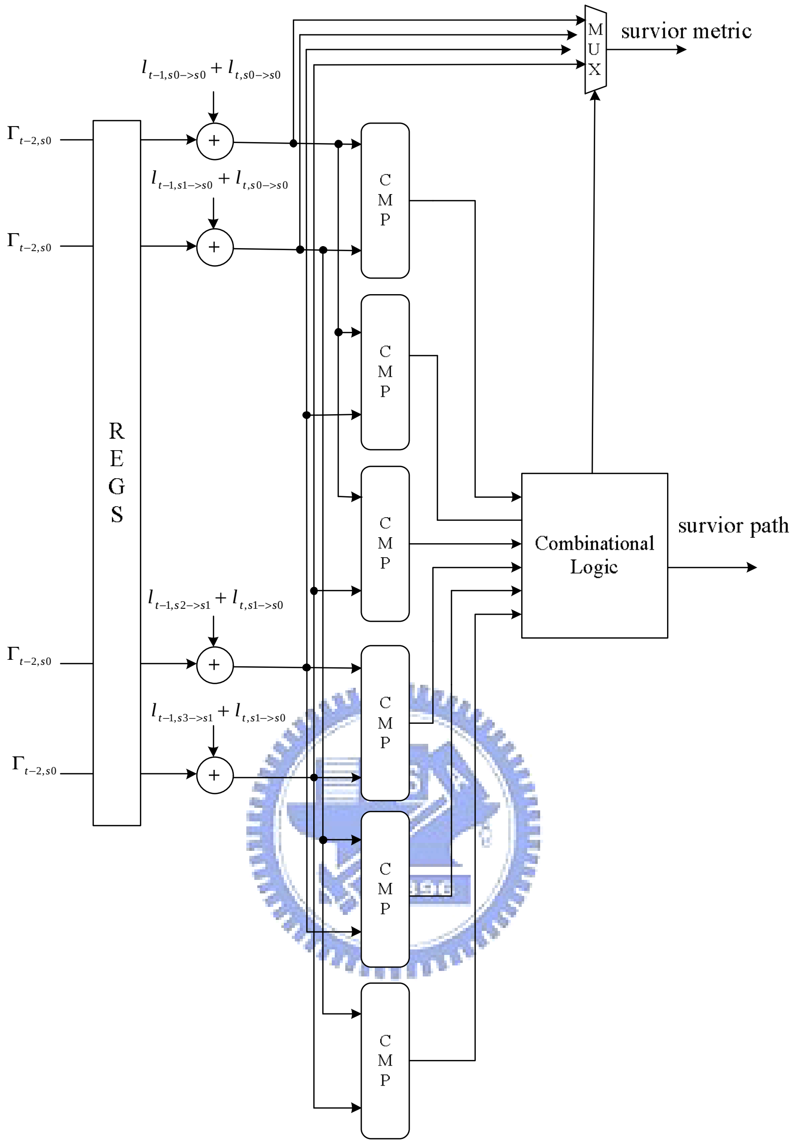

Figure 3.13: Modified Radix-4 ACS

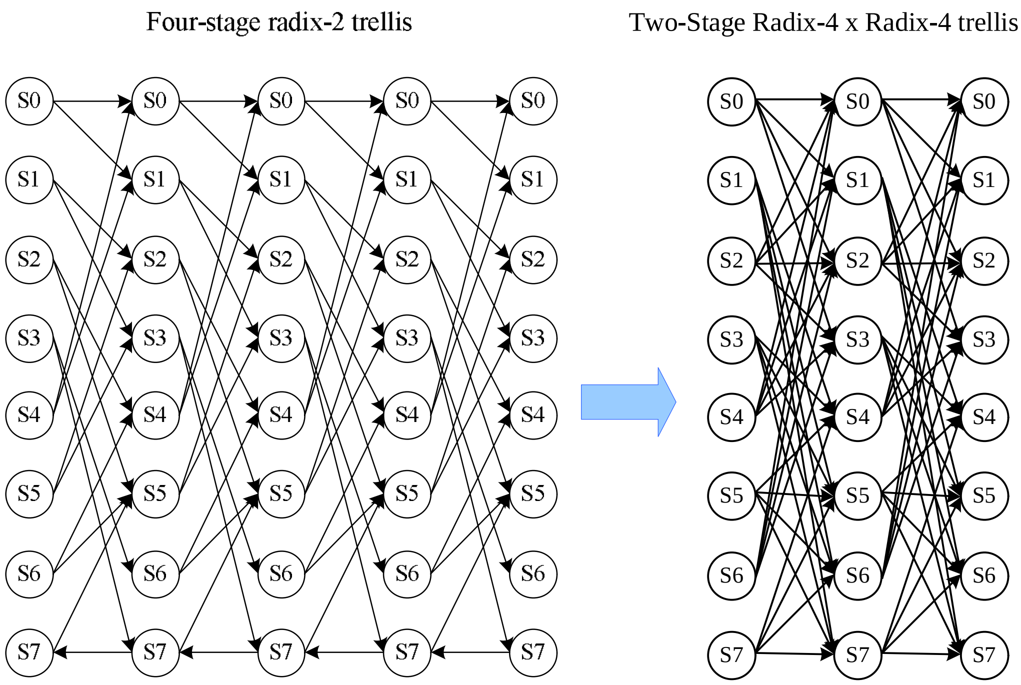

Figure 3.14: Conversion from four-stage radix-2 trellis
to two-stage radix-4 x radix-4 trellis

Chapter 4

Architecture of Viterbi Decoder

By using the ACS scheme as described in chapter 3, we will discuss the implementation of outer receiver. Figure 4.1 depicts the function blocks of the processed Viterbi decoder. The implementation result, including core size, pin assignment, and timing will be analyzed in this chapter.

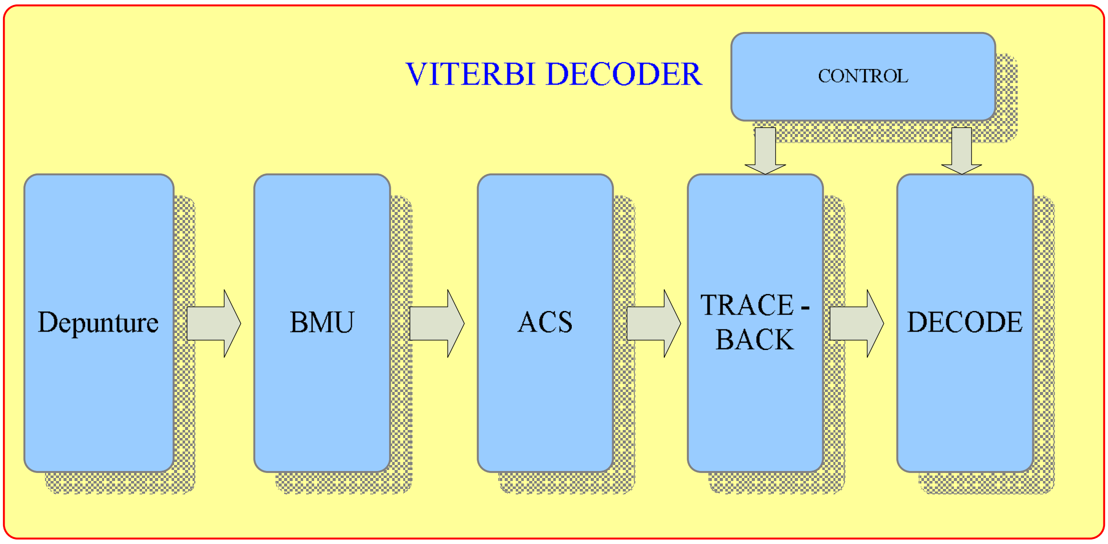

Figure 4.1: Function blocks of Viterbi decoder

4.1 Depuncture Module

In Viterbi decoding, the stolen bits are not sent in their position of the puncture scheme. The stolen bits are taken as dummy bits in depuncture task. In the design of depuncture module, we combine it with branch metric computation (BMC) module. The following sections in this chapter will discuss the decoding mechanism of each modulation type, respectively. Notice although we use (3, 1, 3) convolution code to depict these patterns, the same situations also apply for (3, 1, 7) convolution code in the following discussions. As depicted in Fig. 4.2, Fig 4.3, Fig 4.4 and Fig 4.5, the patterns of four kinds of coding rate, 1/3 ,11/32 ,1/2 ,3/4 & 5/8 are shown, respectively.

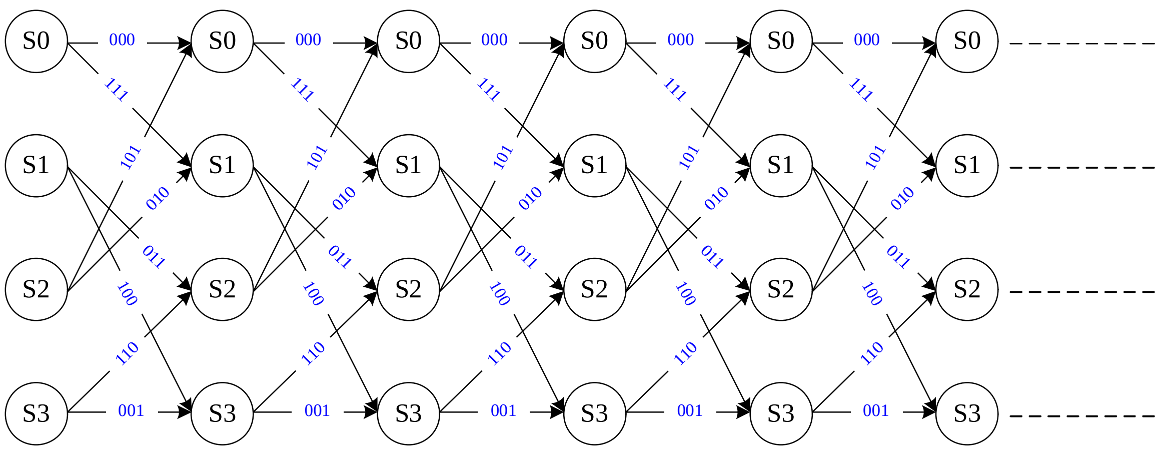

Figure 4.2: The pattern of coding rate 1/3

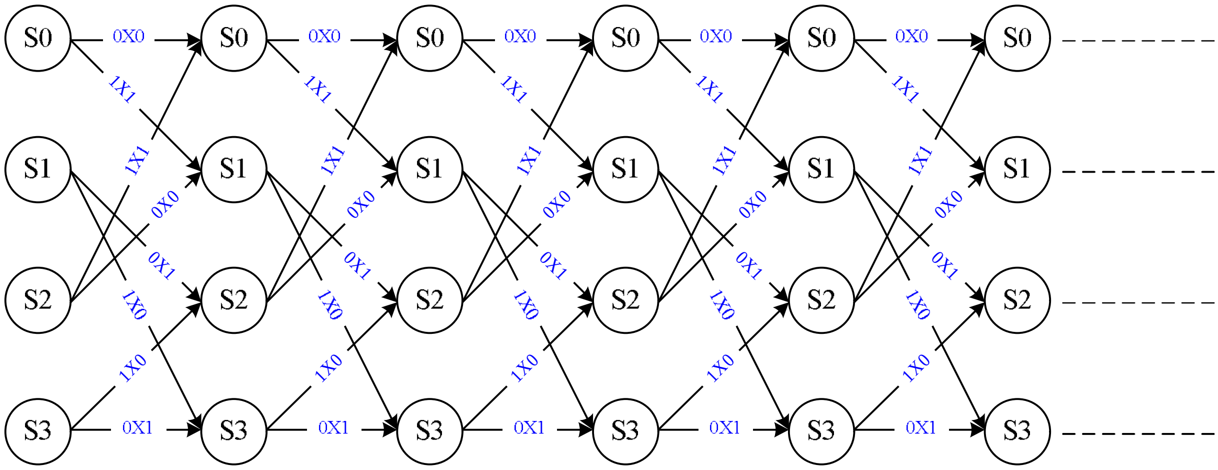

Figure 4.3: The pattern of coding rate 1/2

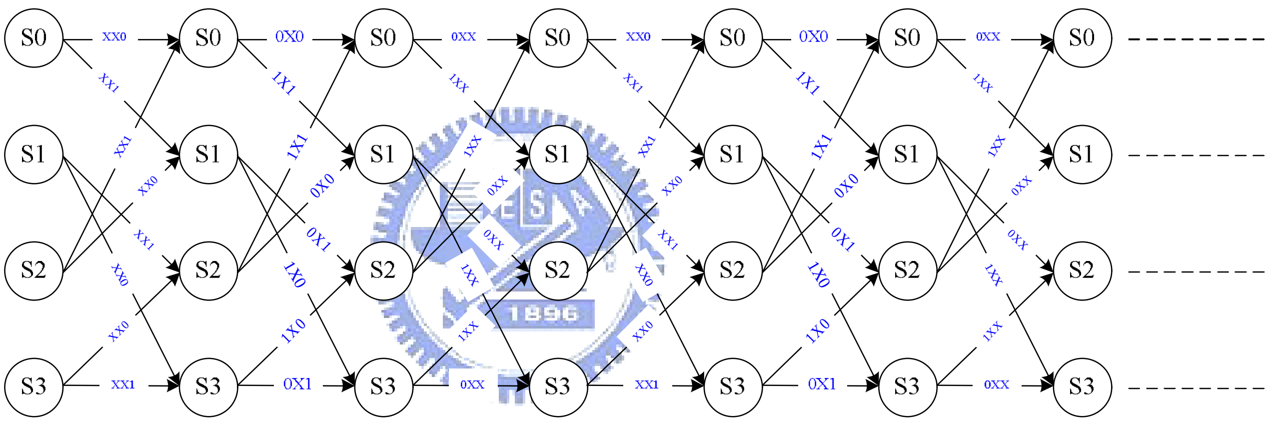

Figure 4.4: The pattern of coding rate 3/4

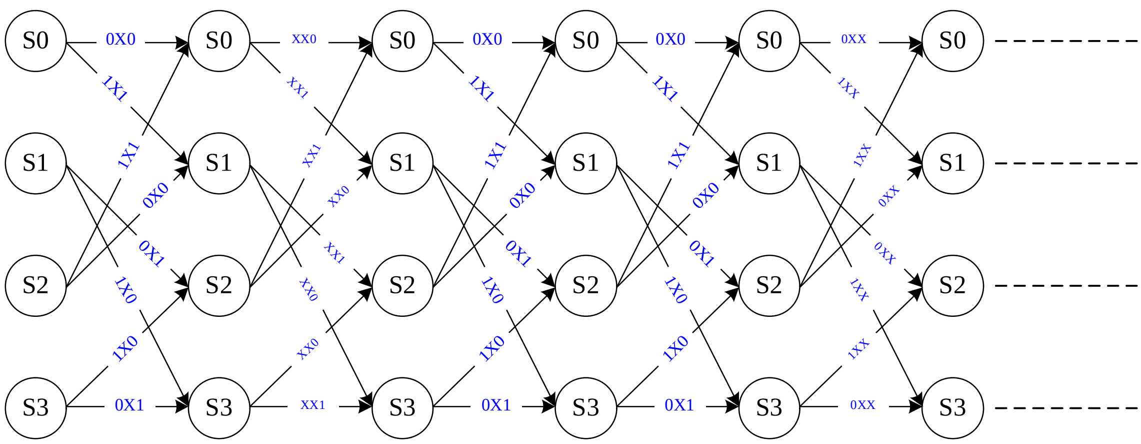

Figure 4.5: The pattern of coding rate 5/8

4.2 Viterbi Decoder Module

Generally, the parameters of Viterbi decoder contain the resolution bits of soft decision and traceback length. Number of resolution brings the trade off between performance and complexity. Besides, the resolution bits of soft decision influences the path delay and the complexity of ACS module. Fig 4.6 depicts the quantization of soft four. For the requirement of Ultra-Wideband standard, the coding gain needs above 5dB. For the property of Viterbi decoder, the performance approximates the ideal case with 4 bits resolution. But the performance of three bits resolution is similar to the performance of 4 bits resolution.

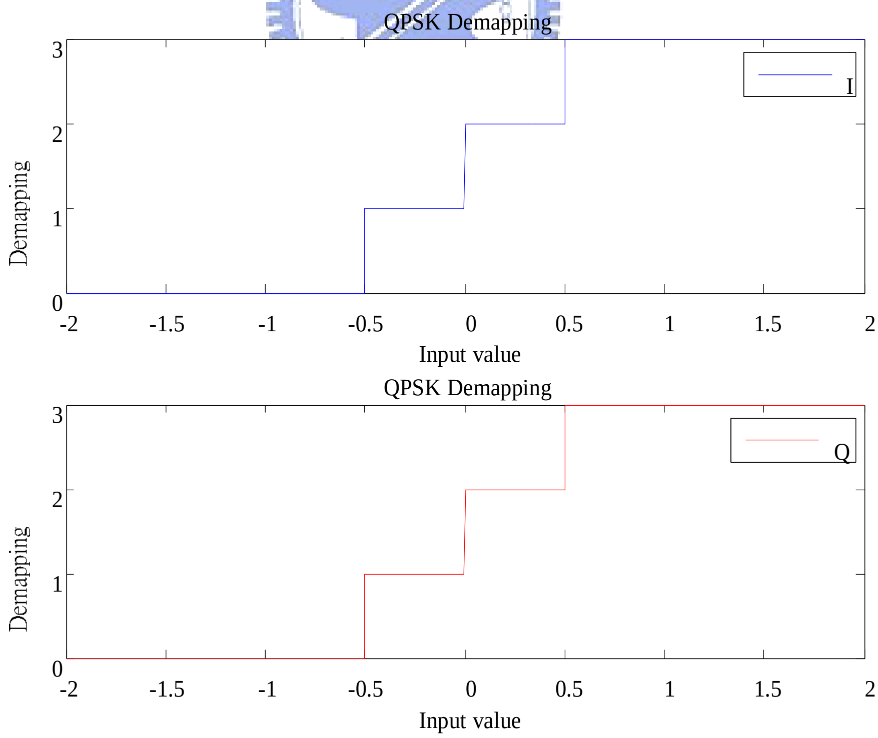

Figure 4.6: Quantization of soft decision 4

Hence soft-decision eight illustrated in Fig. 4.7 is selected for satisfying the requirement of UWB specification by our simulation illustrated in Fig 4.8.

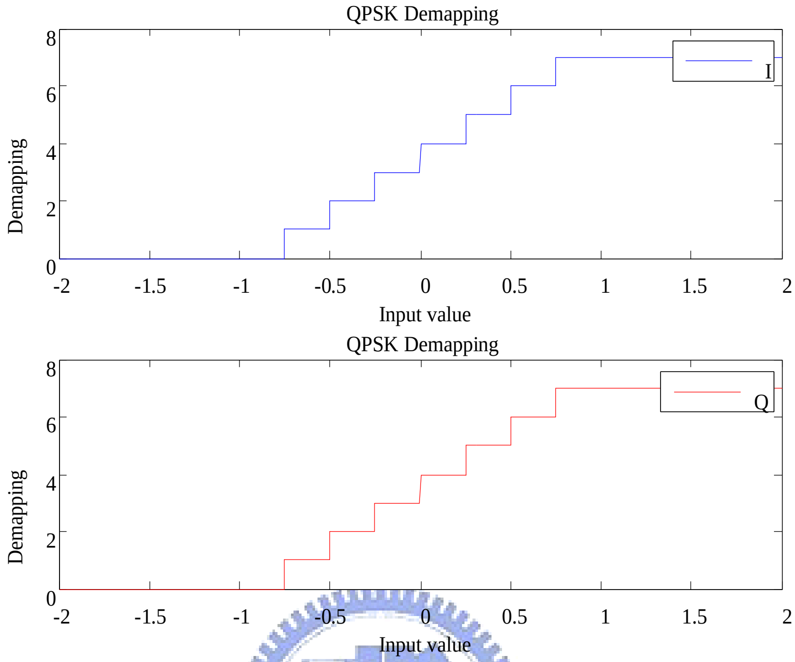

Figure 4.7: Quantization of soft decision 8

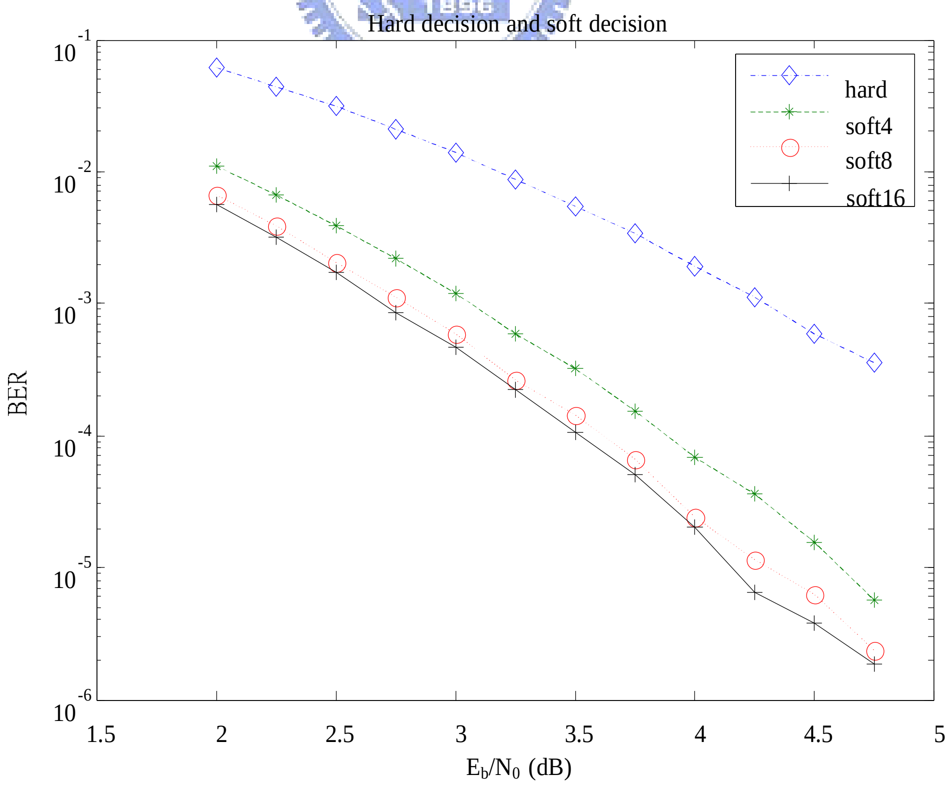

Figure 4.8: Fixed-point simulation of hard decision and soft decision

4.2.1 BMC Module

The caculation of branch metric in Euclidean distance is listed in equation (4.1). The values of X_I , Y_I and Z_I are the received metric and the values of X_I ,Y_I and Z_I are the reference metric from the derived trellis.

$$BM = (X_I - X_{r,i})^2 + (Y_I - Y_{r,i})^2 + (Z_I - Z_{r,i})^2 \tag{4.1}$$

The square value is not desirable for hardware implementation. Though, the metric of correlation derived form equation (4.1) is used in the calculation of branch metric [7]. The modified branch metric calculation equation is represented as:

$$BM' = M_{X,i} + M_{Y,i} + M_{Z,i} \tag{4.2}$$

The values of $M_{X,i}$, $M_{Y,i}$ and $M_{Z,i}$ are the modified metric obtained from table 4.1. Table 4.1 shows the conversion of X_I under bit 1 and bit 0, respectively. The metric is the received value when the referenced bit is one. On the contrary, the metric is calculated by subtracting 7 from the received value.

For example, the decoder receives (0,3,7) symbol and the referenced bits are (0,1,0). The value of ($M_{X,i}$,$M_{Y,I}$,$M_{Z,i}$) will be (7,3,0) and the branch metric will be equal to 10.

Table 4.1: The mapping table of metrics by the correlation algorithm [7]

Soft Decision 3 bits		
input	Ref. bit=1	Ref. bit=0
0	0	7
1	1	6
2	2	5
3	3	4
4	4	3
5	5	2
6	6	1
7	7	0

Because of radix-4 ACS architecture, the BM needs the summation of six received metric. With soft-decision 8, the maximum value of BM is 42 and it needs six bits and it influences the resolution of ACS. The BM is limited in five bits by subtracting those values exceeding 31. Fig 4.9 illustrates the offset for subtraction. Figure 4.10 illustrates the architecture of radix-4 branch metric unit.

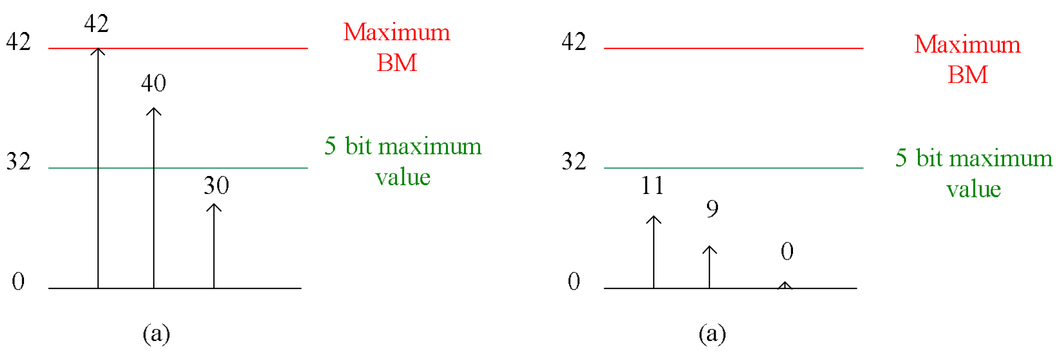

Figure 4.9: (a) Received branch metric (b) Offset for reduction resolution

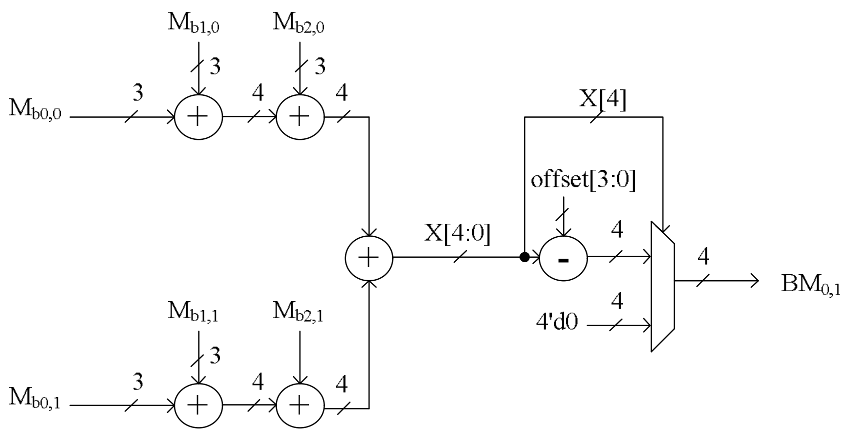

Figure 4.10: The radix-4 branch metric element

4.2.2 ACS Module

4.2.2.1 Implementation Issues

Depending on chapter 3 we have described, the two-stage radix-4 ACS architecture used in the literature. Table 4.2 lists the complexity and design respects of different ACS units. With consideration of backend margin, we use two-stage radix-4 with arithmetic CS.

Table 4.2: Analysis of ACS architecture

	Throughput	Min. required clock(Mhz)	Hardware Complexity	ACS unit Min. Delay(ns)	ACS unit Power (mW)	ACS unit Area(gate counts)
Radix-2	480Mb/s	480 Mhz	2 adders, 1 comparator	1.83 ns (target=2.08ns)	4.29	362.5
Radix-4	480Mb/s	240 Mhz	8 adders,3 comparators	3.48 (target=4.16ns)	15.19	1066.1
Modified Radix-4	480Mb/s	240 Mhz	4 adders,3 comparators	2.75 ns(target=4.16ns)	10.3	782
Radix-4 with arithmetic CS	480Mb/s	240 Mhz	8 adders,6 comparators	2.94 (target=4.16ns)	16.39	1194.5
Modified Radix-4 with arithmetic CS	480Mb/s	240 Mhz	4 adders,6 comparators	2.26 (target=4.16ns)	12.81	935.7
two-stage modified Radix-4 with arithmetic CS	480Mb/s	120 Mhz	8 adders,12 comparators	4.43 (target=8.33ns)	23.2	1762.3

4.2

.2.2 ACS Overflow Prevention

The operation of ACS module is recursive and its word-length is finite. Therefore, if we do not prevent the overflow from appearing, the results of survivors will go wrong.

We use the common method. First, we set the overflow threshold based on resolution 7 bits. The path metrics are subtracted from a truncated threshold at each state when the overflow happens. And those path metrics below the truncated threshold are set zero. When the path goes through 4-stage trellis diagram, the new path metrics will replace the original. Therefore, the overflow problem would never happen [12] [13] .

For example, the path metrics are set as (70, 40, 33, 10) in Fig.4.11. With the overflow path metric appeared, the new path metrics are (38, 8, 1 ,0) after overflow prevention.

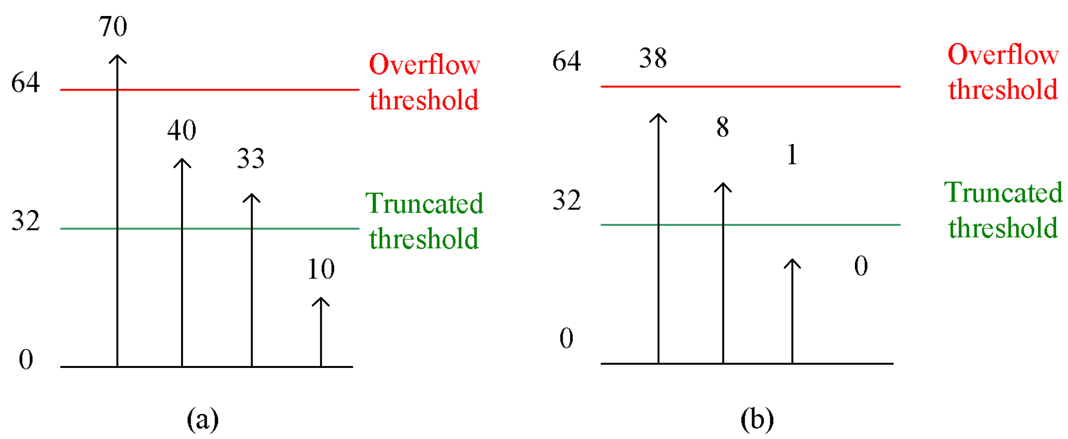

Figure 11: (a) Path metrics with overflow appeared

(b) Path metrics after overflow prevention

Figure 4.12 depicts the architecture of overflow prevention unit.

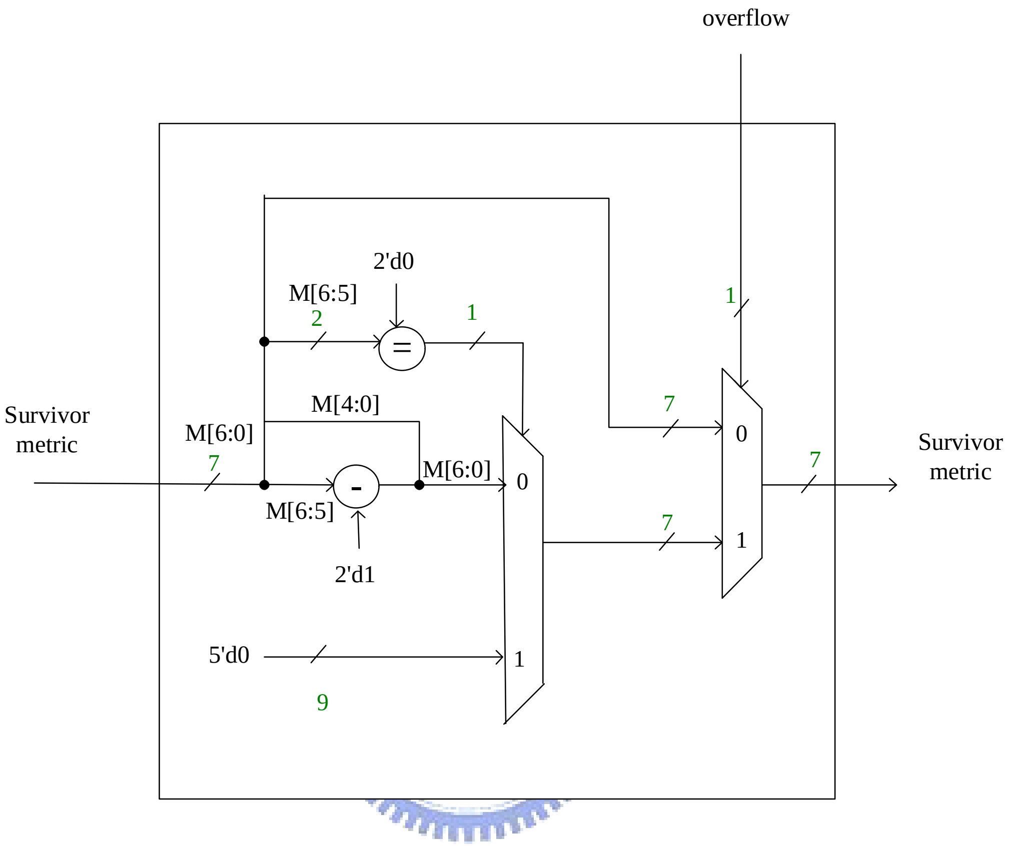

Figure 4.12: Overflow prevention element

4.2.3 Traceback Module

In the literature, the traceback algorithm is adopted for decoding mechanism. We proposed the radix-4 traceback element (TE) depicted in Figure 11. The survivor path is selected as “00” while the output survival path is notated as “Survivor Path 0” and “01” as “Survivor Path 1”, and so on. If the survival path exists in this state, one of the input path will be asserted and one of the output survivor path will be passed..

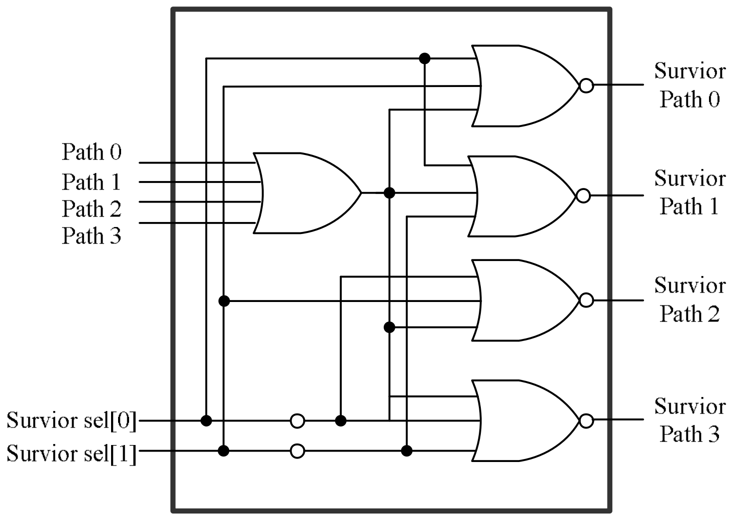

Figure 4.13: The radix-4 traceback element

The traceback architecture is a combinational circuit shown in Fig. 4.13, including 48x40 traceback elements. The starting state starts at zero state in this design.

The decoding mechanism is according to the trellis structure. Table 4.3 where i ranges from 0 to 15 depicts the decoding table for our trellis structure. In this table, the traceback path goes into the decoded states. The decoded bits can be decoded by its survivor states. Figure 4.14 illustrates the traceback architecture.

Table 4.3: Decoding mapping table

Survivor state	Decoded bits
4xi	00
4xi+1	01
4xi+2	10
4xi+3	11

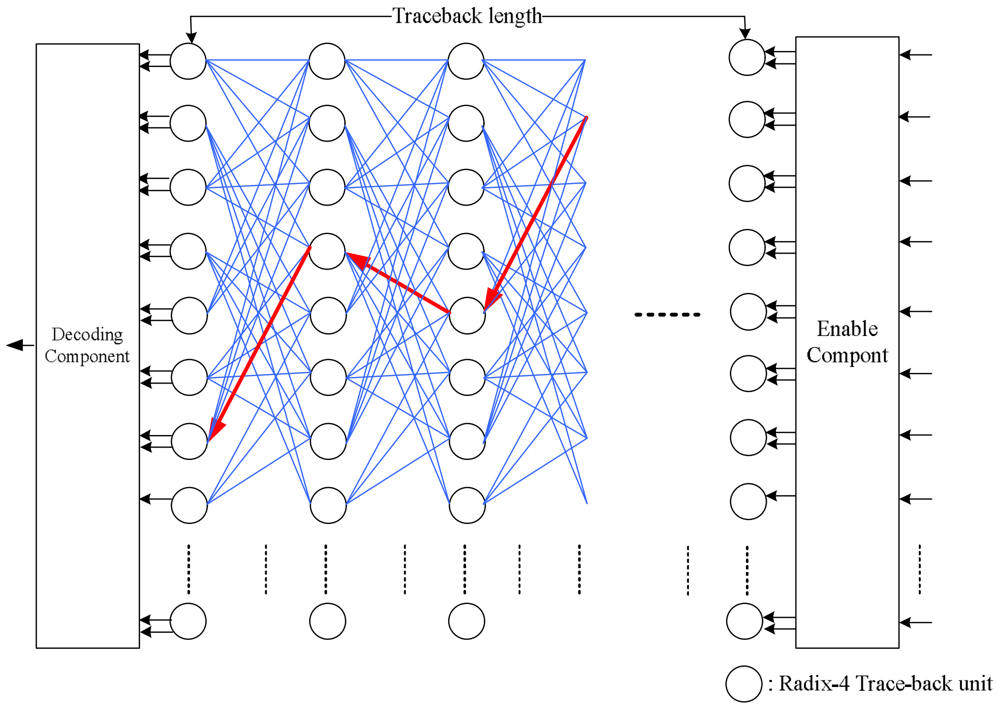

Figure 4.14: The traceback architecture

4.2.4 Discussion between different Traceback Length

For the demand of real-time decoding mechanism, the information of survivors must be transmitted to the traceback architecture parallel from ACS block. Therefore, registers are used as storage in the high speed design.

Generally, the traceback length is five times more than the constraint length [9]. Consideration of implementation, the path delay in the combinational circuit can not be too long for completing the operation because it will cause to be rather long the traceback length. We select traceback length to be 40 and it is based on the simulation of Fig. 4.15. Besides, the path delay is still too long to be finished in a clock cycle. Therefore, the property of path merge can be used for multi-cycle design [14].

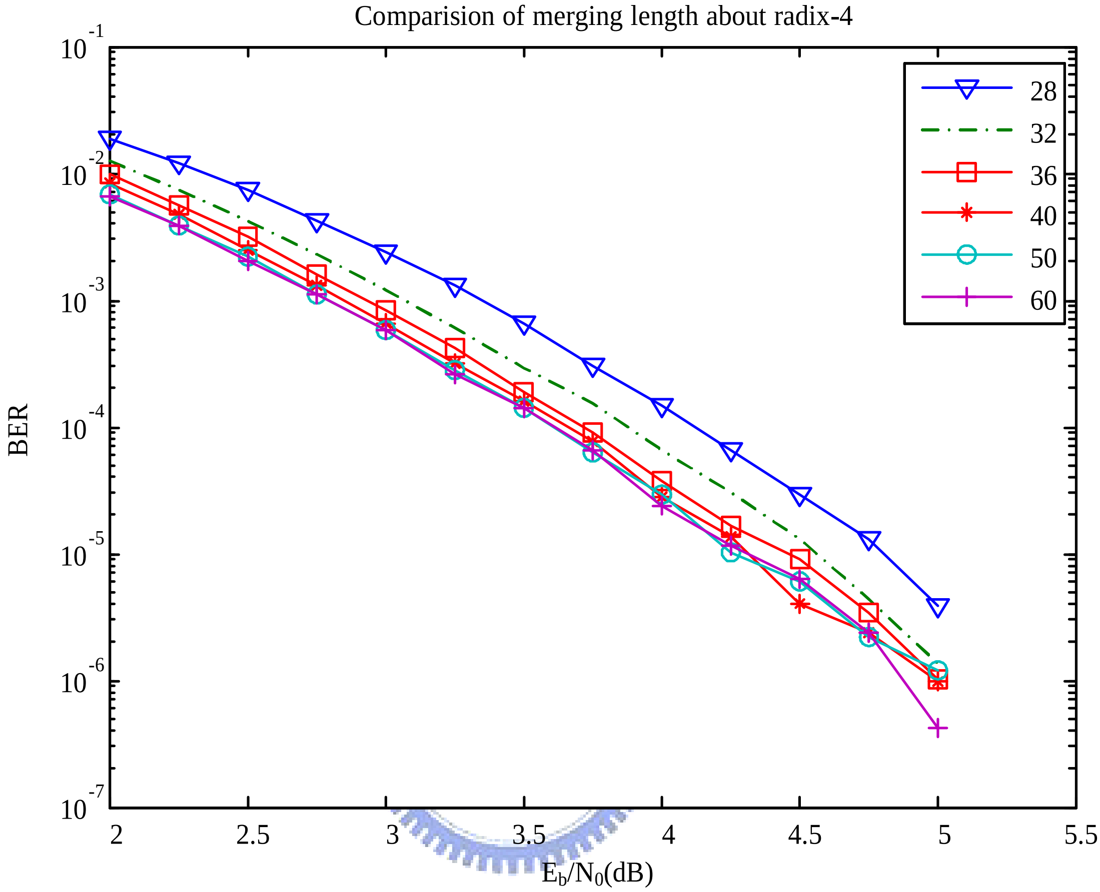

Figure 4.15 Performance between different traceback Length

In the design, the path delay of the tracebacck module is roughly 13ns. Which can not be completed during one clock cycle and it takes two clock cycle for completing the operation of trackback, and the decoding length is eight. This design also decreases the power consumption because of the reduction of registers switches. The total registers of the traceback module are 56x64 bits, and the extra 8x64 bits are for the buffer. Fig 16 depicts the decoded length and traceback length.

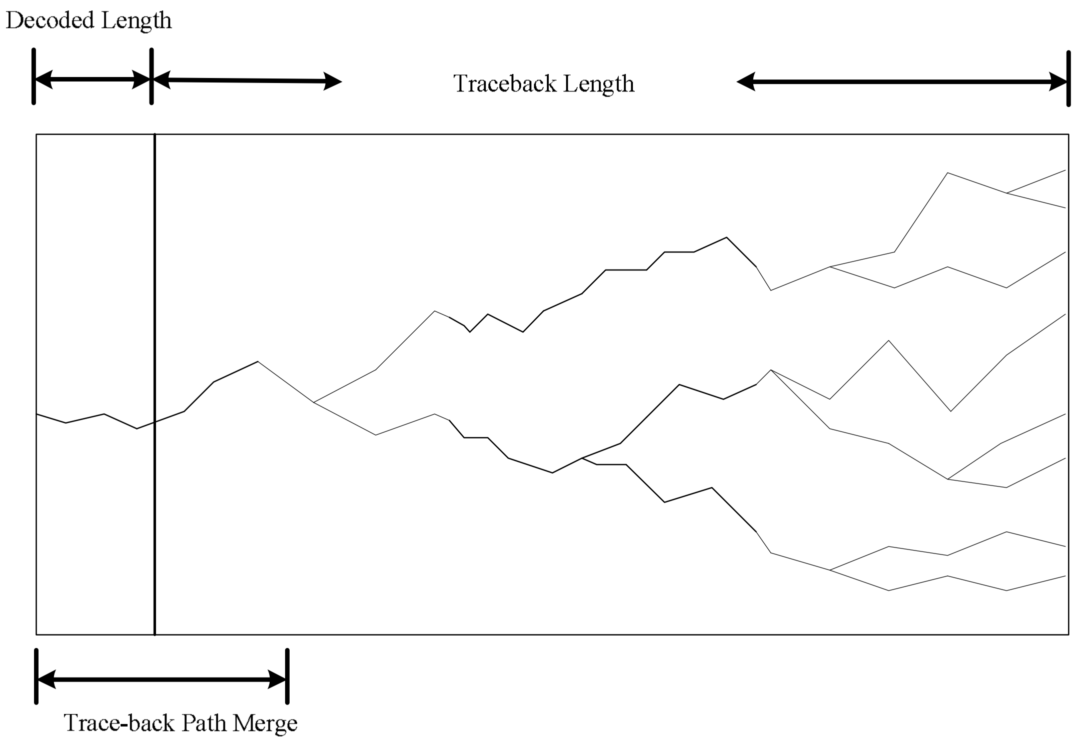

Figure 4.16: The property of path merge in traceback

Chapter 5
Implementation and Verification

5.1 Introduction

In this chapter we discuss the design flow, verification plan, IP qualification and co-simulation for the proposed design. In this study, behavior model is built by C which is bit accurate and a MATLAB model for system co-simulation with RF. The design flow is illustrated in Fig. 5.1, and this is a kind of waterfall models which works well up to 100k gate count design. It is a serial flow from specification survey to post layout and there integrate a verification flow to verify the design [15]. In this design flow, the RTL module is verified by accurate C model and the system co-simulates with MATLAB model. Why do we use two behavior models? The reason is that C program has much higher processing speed than MATLAB and for MAC link. For example, the simulation time by MATLAB is too long when the amount of simulation is up to 10^6 and not to mention applying more information. Besides, with the behavior model the baseband and RF co-simulation platform can be built in Agilent ADS tool. Therefore, the platform can be applied to RF model simulation.

After RTL code is development and verified, there are two ways for implementing design, one is ASIC, and the other is FPGA prototyping. FPGA prototyping is for verifying hardware design in general, because FPGA can simulate the work in real world and some situations which we don't concern may appear. If we want to produce ASIC or IP, we will go through synthesis and Place & Route. First, we synthesis the design to gate-level netlist by reasonable design constrains. After checking the timing, area and power, we will run Place & Route. After timing, area, power and design rule are all conformed, the design is done [15].

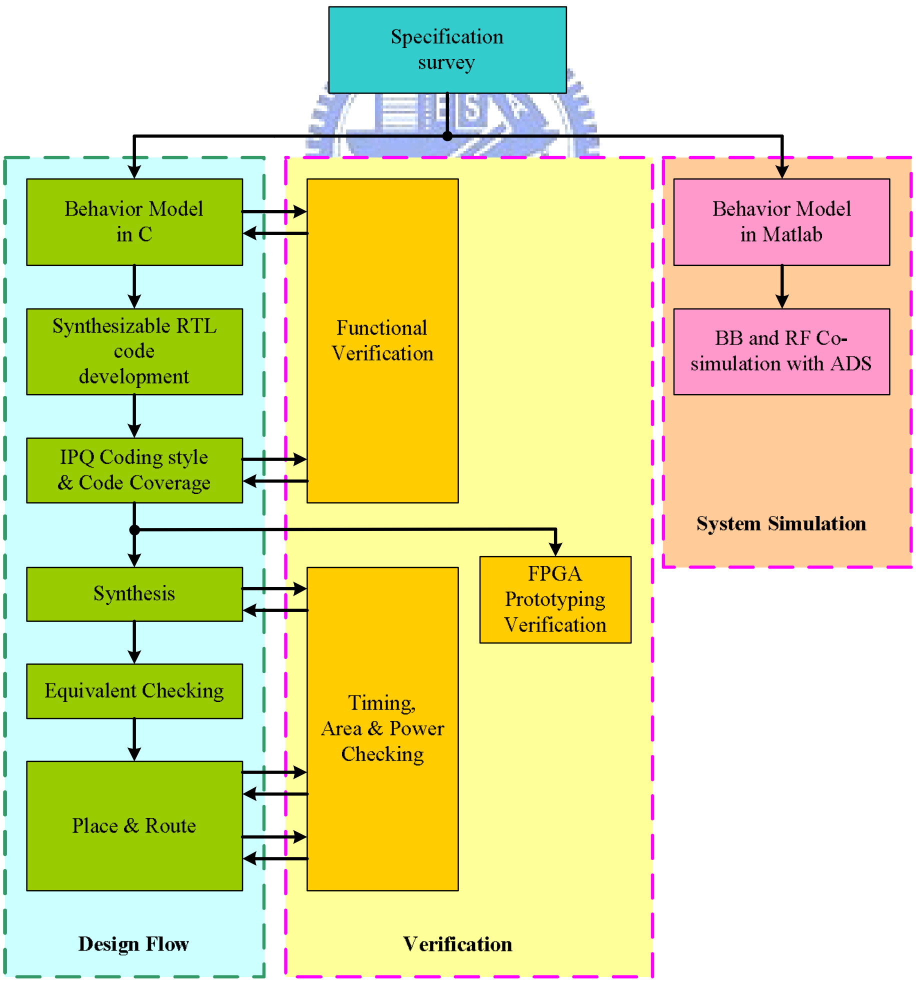

Figure 5.1 Design & Verification flow

5.2 System co-simulation

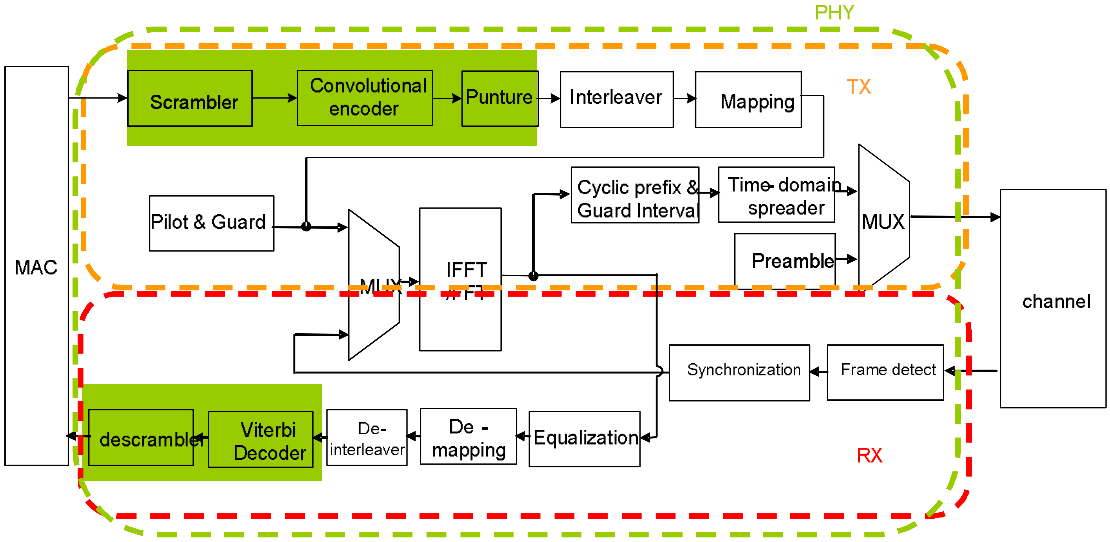

Figure 5.2 Design & Verification flow

Our system is illustrated in Fig. 5.2. The platform by MATLAB is based on 802.15.3a standard. The pattern is transmitted in ten rates with the varieties of puncture scheme, conjugate and spreading. Figure 5.3 depicts the pack error rate at 480Mb/.

In system co-simulation, we firstly know that the information of RF simulation is viewed as timed sequence. Hence, the sequences calculated by MATLAB should be packed and transformed into timed sequence. Then, RF team can check their parameter settings and performance, such as TX EVM, TX power spectrum, RX sensitivity and PER etc. Figure 5.4 depicts the co-simulation platform.

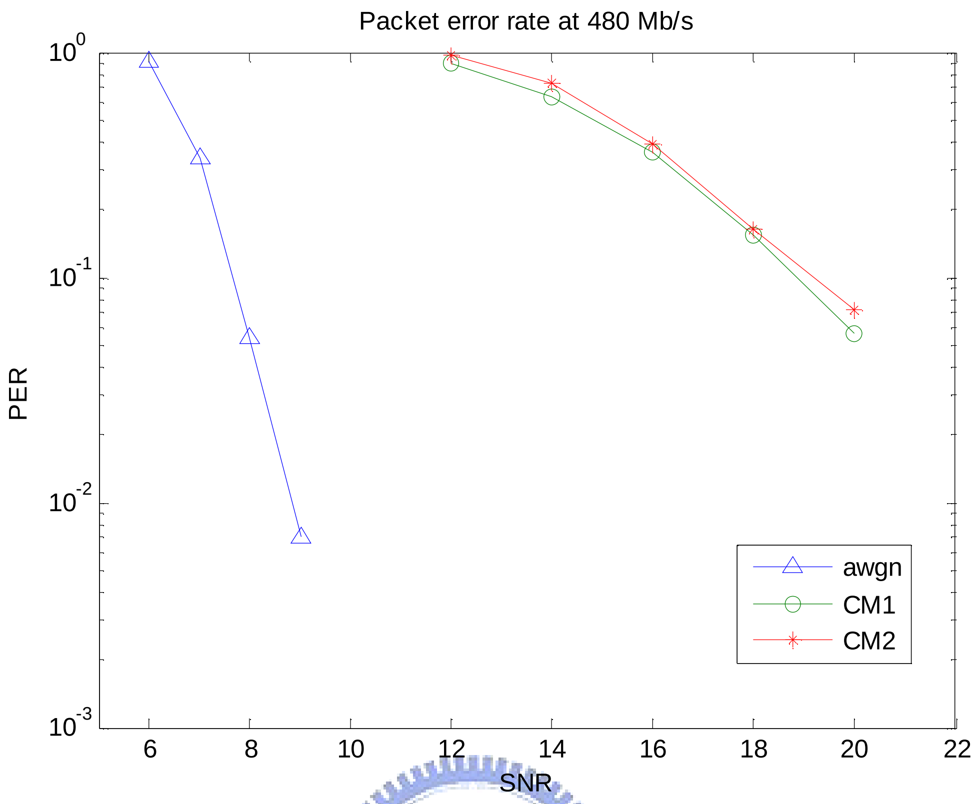

Figure 5.3 Pack error rate at 480Mb/s

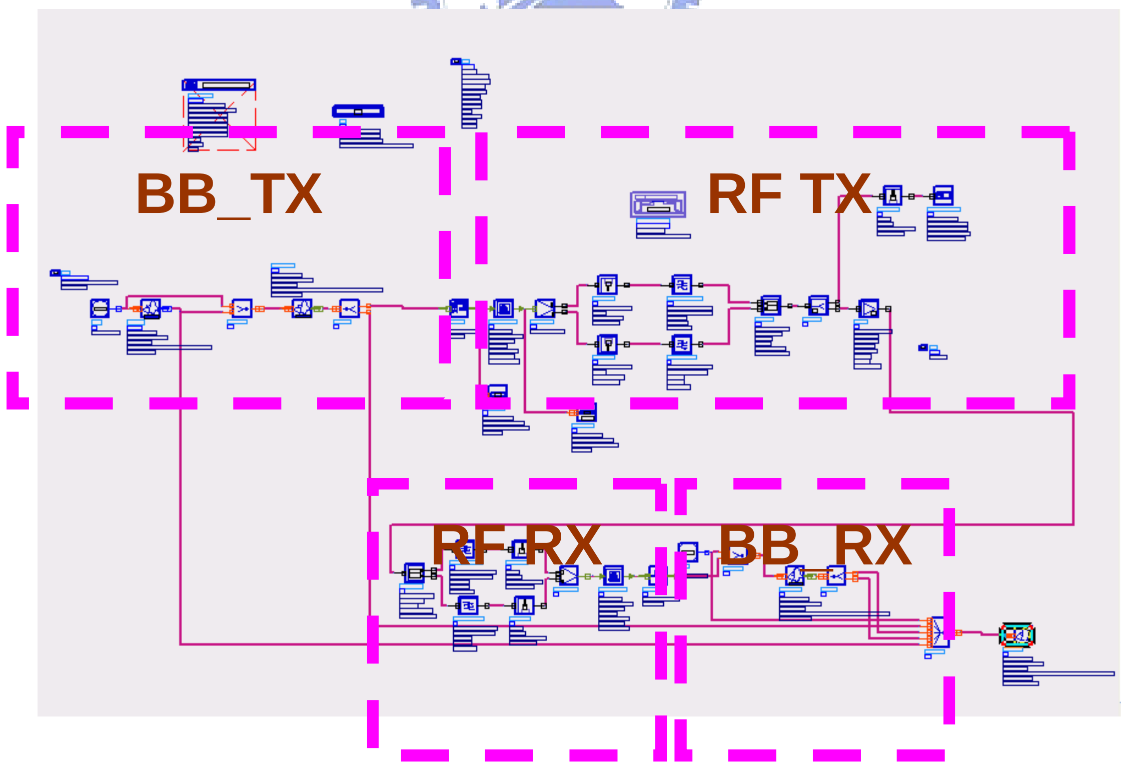

Figure 5.4 Co-simulation platform

5.3RTL Design and soft IP Qualification

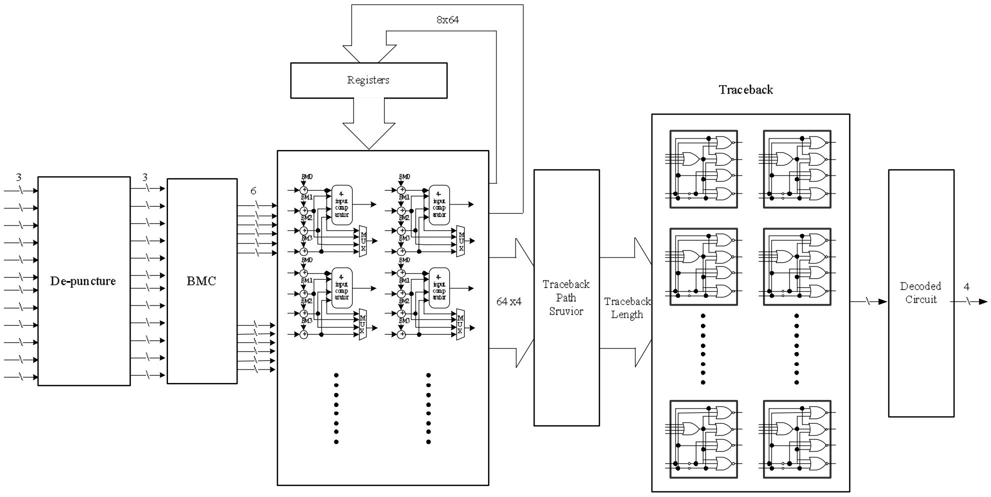

Figure 5.5: The proposed Viterbi Decoder architecture

The synthesizable RTL is desired according to the architecture discussed in chapter 3 and the architecture is illustrated in Fig 5.5. In this study, we also discuss the soft IP qualification (IPQ). IPQ has its defined coding style [16]. Table 5.1 and Fig 5.6 depict the number of coding rules fitting IPQ in our design. In this table, two warnings come from the architecture of feed-back circuit because of overflow prevention and the others are header warnings.

Table 5.1 Number of coding rules fits IPQ

	Before Modification	After Modification
ERROR	1076 Errors	0
M1 & M2	2132 Warnings	25

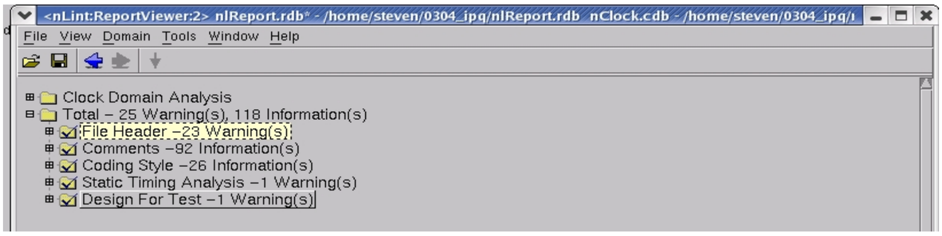

Figure 5.6 Design & Verification flow

The IPQ needs reasonable test patterns for function verification. The code coverage means that the percentage of the verified design is checked in different verifying methodology. The code coverage of statement, condition and toggle coverage are almost up to 100% in our design. The results are illustrated in Fig 5.7, Fig 5.8 and Fig 5.9. And we list the met soft IP qualification in table 5.2.

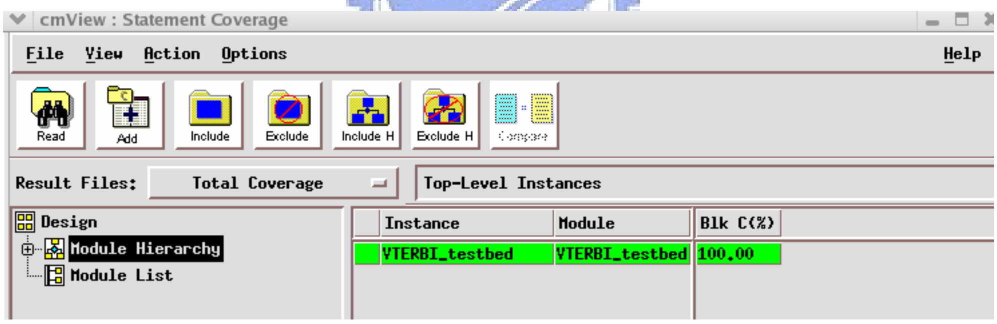

Figure 5.7 Statement coverage

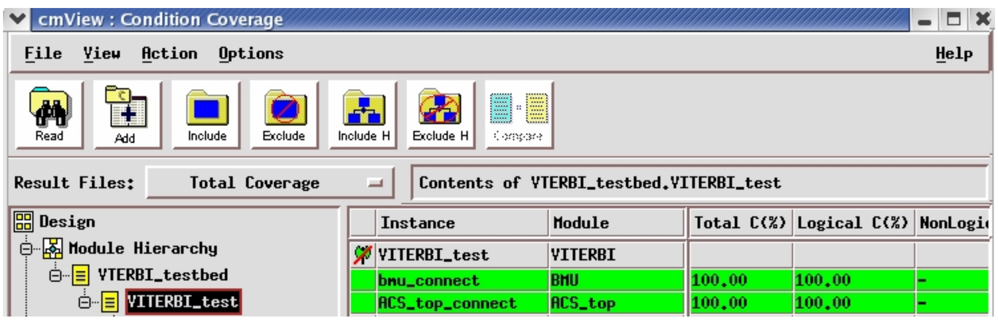

Figure 5.8 Condition coverage

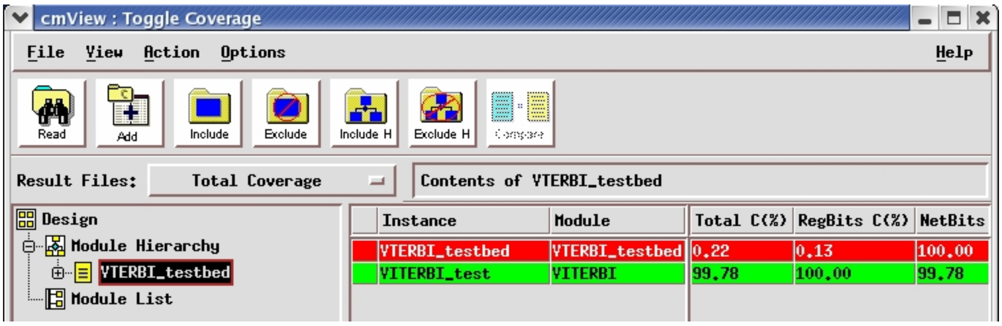

Figure 5.9 Toggle coverage

Table 5.2 The meeting list of soft IP qualification [16]

	Soft IP	IP Designer Self-assessment	File Name and Section
Verification			
[S.VG.1] Verification plan	M1	y	
[S.VG.1.1] RTL dynamic simulation	M1	y	
[S.VG.1.2] All timing exceptions must be identified	M2	y	
[S.VG.2] Code coverage must be conducted and the coverage metrics must be documented and delivered.	M2	y	
[S.VG.3] All response checking must be done automatically.	M2	n	
The rules of writing testbench			
[S.TB.1] Testbenches must begin with a header.	M2	y	
[S.TB.2] Testbenches must be written with comments.	R	N	
[S.TB.3] Keep line length within 72 characters in testbench codes	R	y	
[S.TB.4] Testbenches should be partitioned into behavioral and synthesizable sections.	R	y	
IP prototyping			
[S.PT.1] Soft IP prototyping.	M2	y	

5.4 Function Verification.

Functional verification and debugging usually cost about double time more than develop a RTL code. First, a bit accurate C model based on the decided architecture is built. The BER of C program simulation is compared with ideal Viterbi decoder and satisfied with the requirement of Ultra-Wideband specification. Second, we decide the interface and write the testbed for RTL simulation. Figure 5.10 illustrates the verification plan. The patterns are added with noise and decoded with C model. The testbed is fed with the pattern generated form C program. After finishing the RTL simulation, the BER of C model and BER of RTL code are compared for analyzing the consistency. The gate-level simulation is depicted in Fig 5.11.

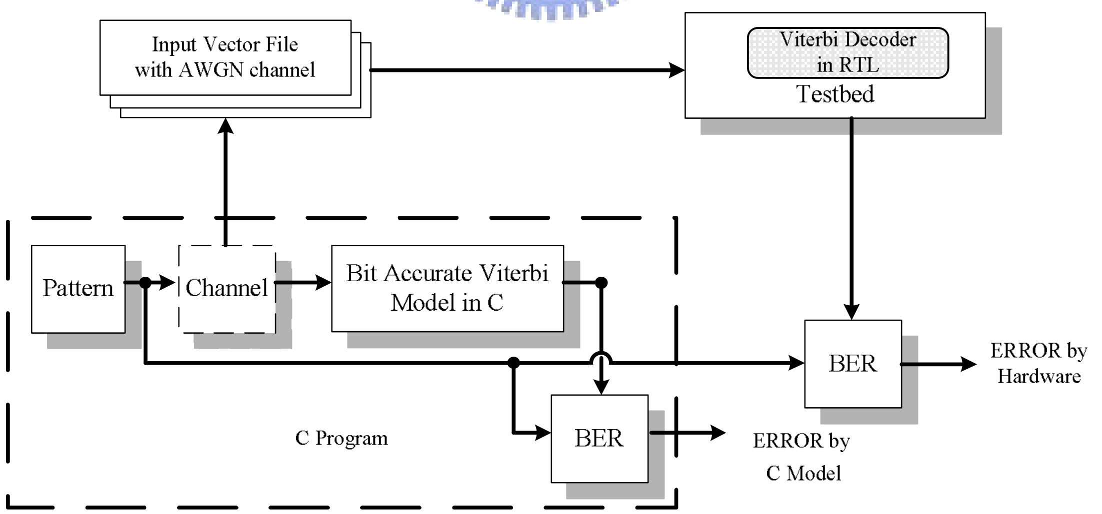

Figure 5.10 Verification plan

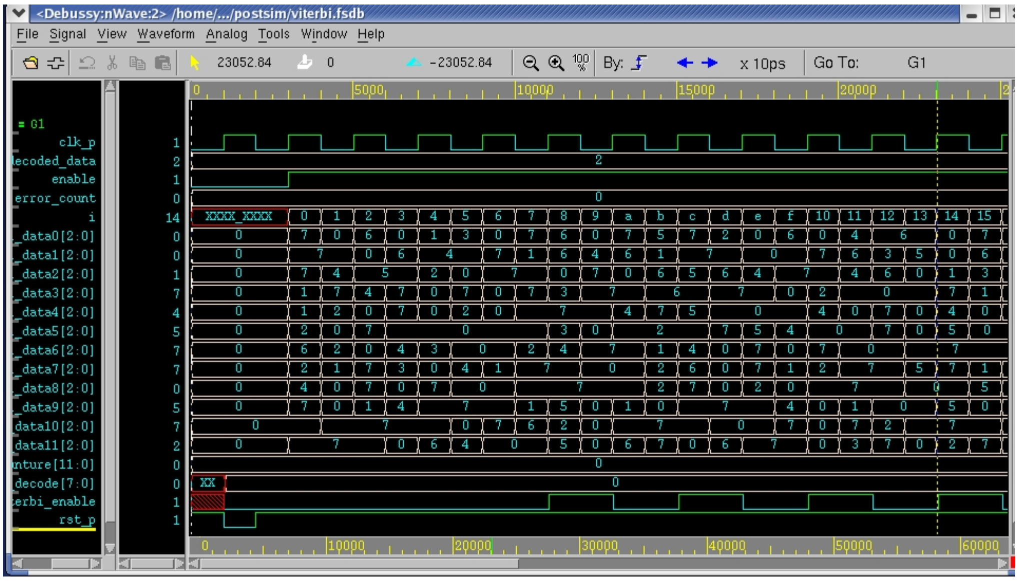

Figure 5.11 Gate level simulation

5.5 Timing and Area analysis

We use SYNOPSYS design compiler to synthesize the register-level Verilog file with UMC0.18 slow library. And the parameters of the Viterbi Decoder are: 3-bit soft decision and traceback-length 40. The gate counts and the critical path delay of each module is shown in Table 5.3, respectively.

Table 5.3 Synthesis reports for each module

Module Name	Gate Count	Max. Path Delay (ns)
BMC (depuncture)	21146	3.72
ACS	83865	6.84
TB	18988	13.52
TB_control	38322	1.59

5.6 FPGA prototyping.

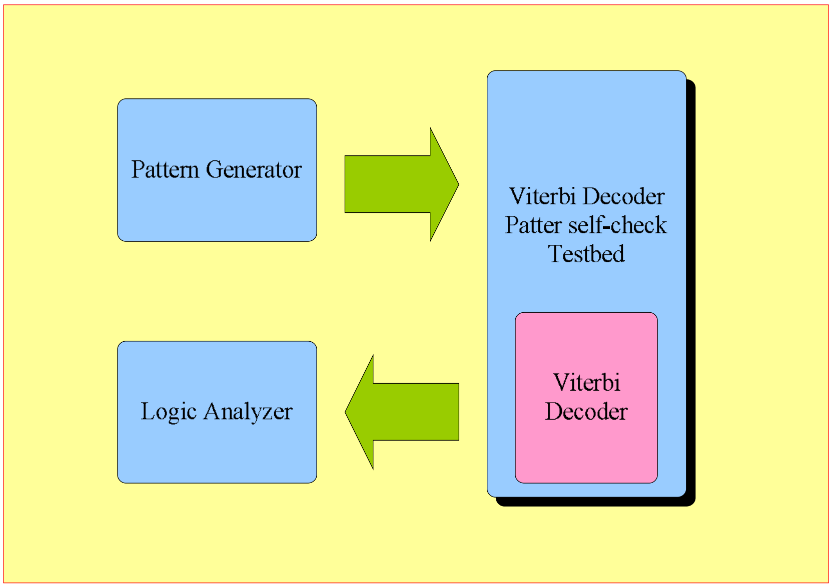

Figure 5.12: The FPGA verification plan

The input pattern is saved in pattern generator and sent to FPGA. Then, we check the result by waveform or dump the result file for checking. In this study, we build another synthesizable built-in testbed and test pattern in FPGA. The self-check testbed has synthesizable verification pattern and self-check circuit for cycle accurate error checking. The FPGA verification plan is shown in Fig. 12. Figure 13 depicts the verifying situation.

Table 5.4 Xilinx FPGA synthesis report.

Target Device	xcv2000e-bg560-6
Slices	12252
Slices Flip Flops	6419
Gate count	21835
Timing	52.125ns (15.606ns logic, 36.520ns route) = 19.18 MHz

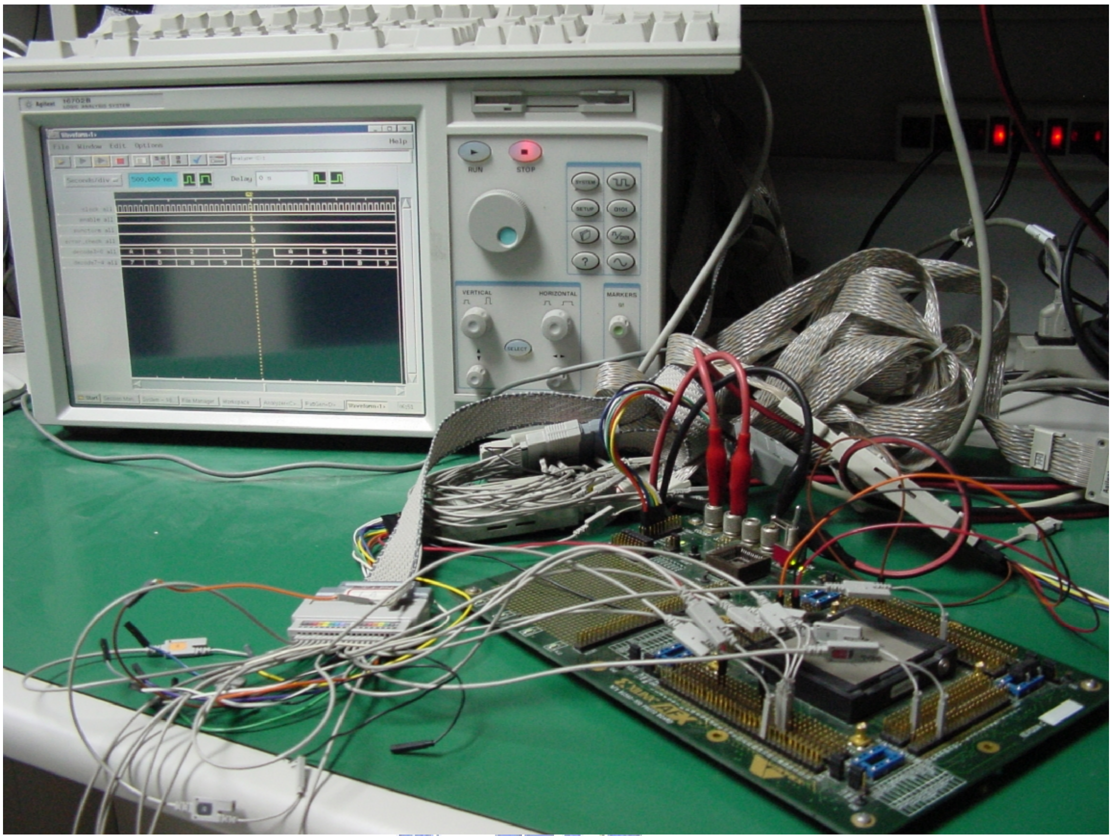

Figure 5.13: Pattern Generator, Logic Analyzer and Xilinx xcv2000e6bg560 FPGA.

5.7 Implementation Results

The macro is implemented by cell-based design flow, and fabricated in 0.18 CMOS process. We use SYNOPSYS Design Compiler to synthesize the gate-level Verilog file. And the parameters of the Viterbi Decoder are: 3-bit soft decision and traceback-length 40. The pins of Viterbi module are shown in Fig 14.

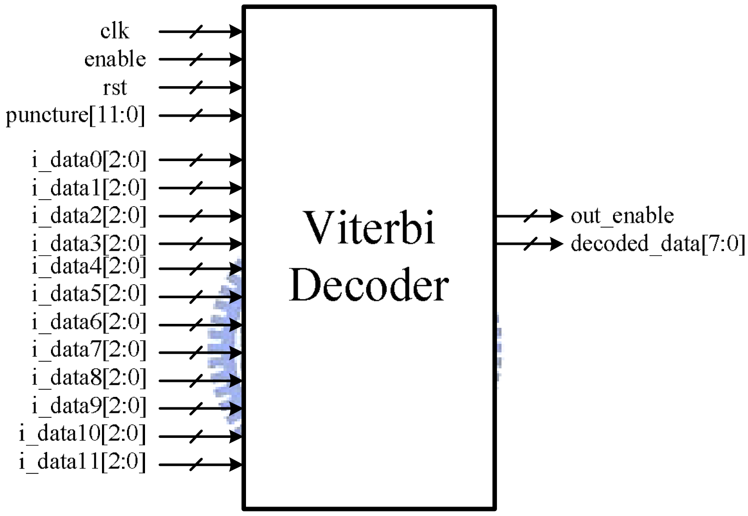

Figure 5.14: Viterbi Interface

The timing diagram is illustrated in Fig 5.15. Because of two-stage radix-4 architecture, the design uses the quarter clock. The first decoded data is calculated after sixteen clock cycles when the Viterbi decoder starts decoding. In the first sixteen clock cycles, the data passes through the blocks of BM and ACS and fills the traceback memory.

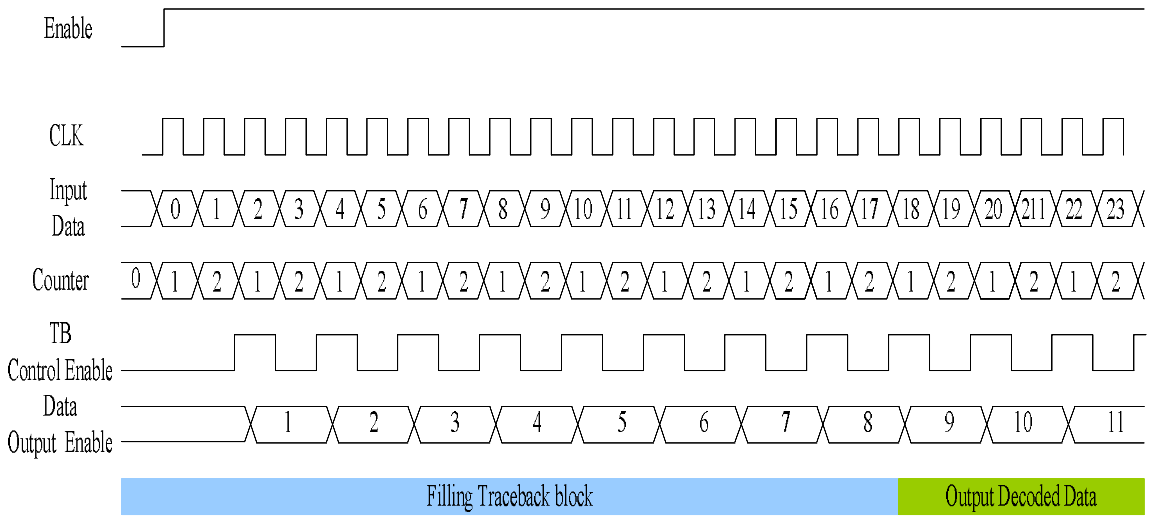

Figure 5.15 Timing Diagram

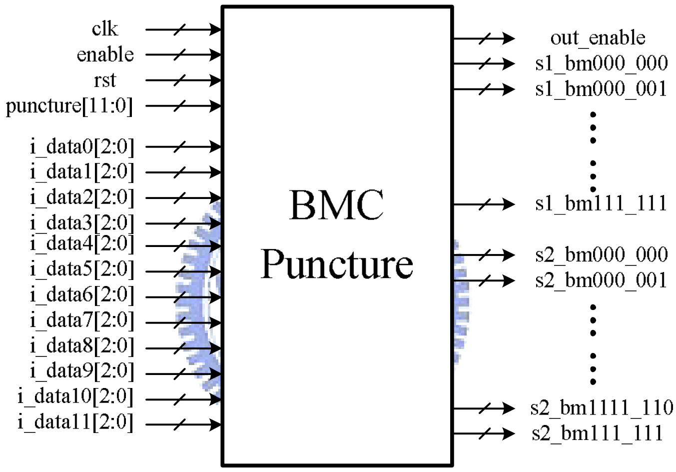

Figure 5.16 BMC Module

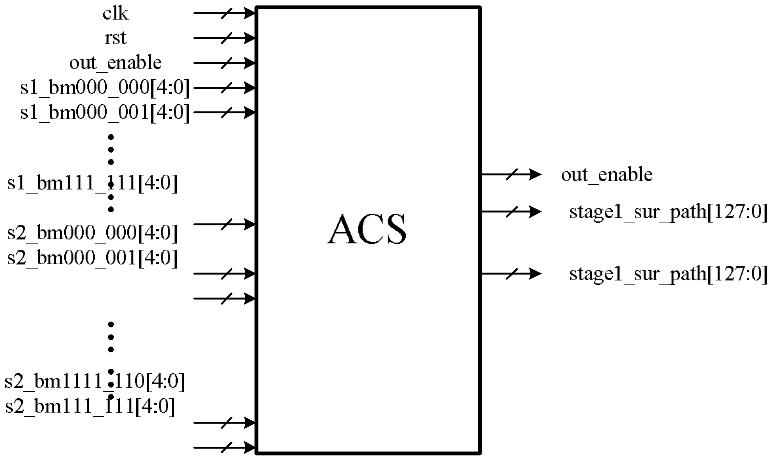

Figure 5.17 ACS Module

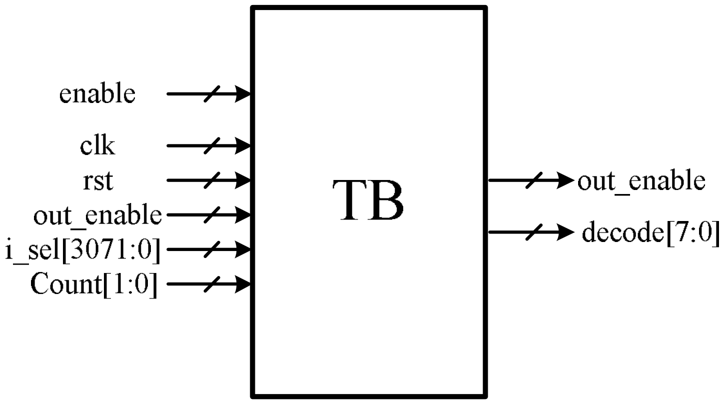

Figure 5.18 TB Module

The detail pins assignment of sub modules are depicted in Fig 5.16, Fig. 5.17 and Fig. 5.18.

Table 5.5: The layout area of the proposed design

Module Name	Viterbi Decoder
Gate Count	176K
Macro Size	2037x2017
Max. Throughput	480 Mbps
Power dissipation	78.85mW

The maximum operation frequency is 120MHz and the throughput is 480Mb/s. The PAR process of layout is applied with SYNOPSYS ASTRO by UMC 0.18μm. The area of the layout is shown in Table 5.5. Figure 5.19 depicts the macro layout view of Viterbi Decoder.

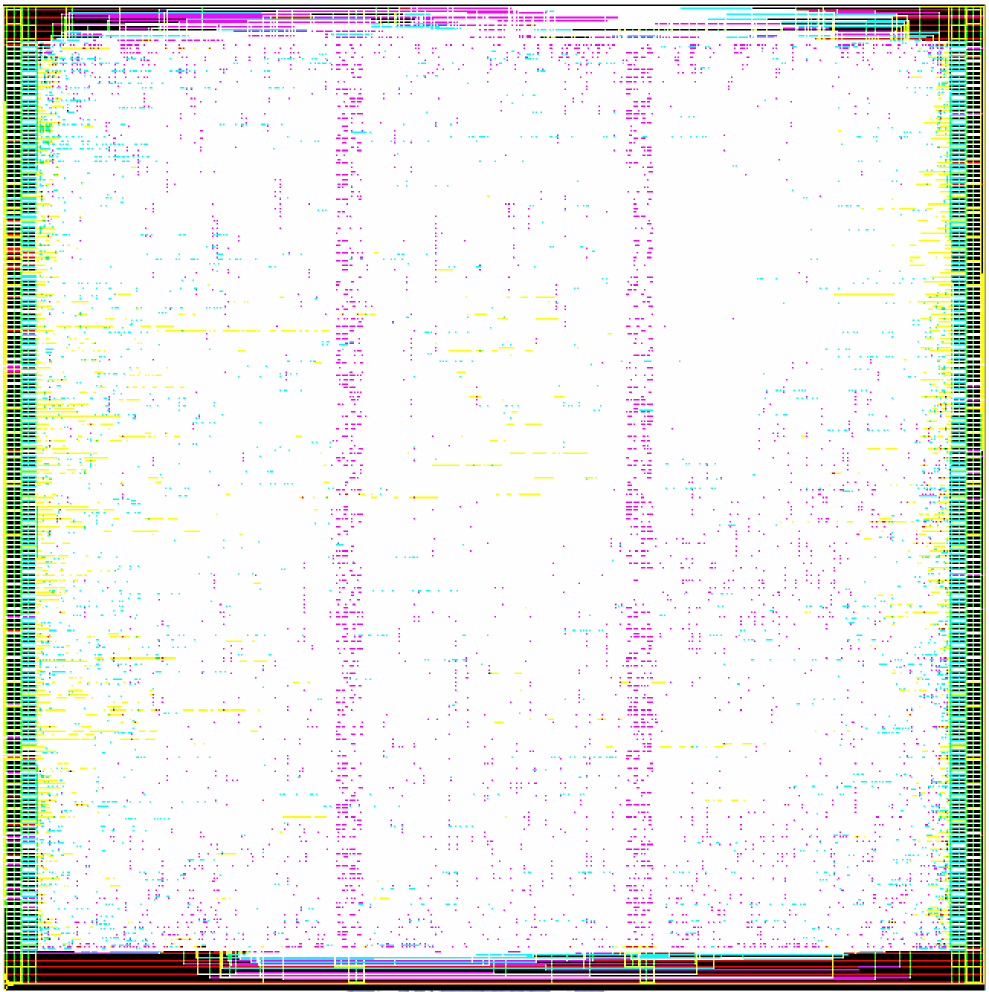

Figure 5.19: Macro Layout View

5.8 Performance Analysis

Table 5.6 Comparison of Viterbi Decoder

	This design	VLSI design and implementation of high-speed Viterbi decoder [12]	200Mbps Viterbi Decoder for UWB[2]
Synthesis speed & synthesis library	6.84ns with UMC018 slow library	N/A	N/A
Synthesis gate count	176K	50K	87K
Throughput	480Mb/s @120Mhz	200Mb/s @100Mhz	200Mb/s @100Mhz
Latency	4+12=16 clock cycle (at quarter clock rate)	N/A	N/A
P&R speed	8.13 ns	N/A	N/A
P&R core area	2037 x 1564.64 um^2	3.88 mm^2 @.25CMOS	N/A
resolution	Soft-decision 8	N/A	Soft-decision 8
Architecture	Two-stage radix-4	Radix-4	Radix-4
Parameters	(3,1,7) TB length=40	(2,1,7) TB length=32	(3,1,7) TB length=42

We take some published Viterbi decoders which are listed below [2] [12] as comparison with the proposed Viterbi decoder in Table 5.6. The proposed Viterbi decoder for Ultra-Wideband standard has higher throughput than others listed in this table. And the latency contain 24 clock cycles from filling traceback module and the other four come from the buffer between different modules.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

As the SOC trend becomes popular, the qualification of IP is more important. In this thesis, we consider the soft IP qualification and process the macro design with P&R. Besides, we propose a high performance and high throughput Viterbi decoder for WLAN IEEE 802.15.3a. For soft decision resolution issue, we apply 3-bit soft decision for demapping design. For traceback-length issue, we employ traceback-length 40 in Viterbi decoder. For ACS module, we apply many techniques to improve the critical path such as arithmetic CS and branch metric limitation.

6.2 Future Work

In the proposed outer receiver architecture, we adopt a 480Mb/s at 120Mhz. In system integration, the module could use different clock source from outer clock. Hence, it increases the complexity of integration. Therefore, the higher radix architecture or better P&R techniques could be applied. We can optimize the ACS module in P&R view. Besides timing issue, the high speed ram based design and dynamic traceback length design can be considered for low power issue,

Bibliography

[1] A. Batra, et al., "MultiBand OFDM Physical Layer Proposal for IEEE 802.15 Task Group 3a," http://www.multibandofdm.org, September 2004.

[2] Sung-Woo Choi and Sang-Sung Choi, "*200Mbps Viterbi decoder for UWB,*" ICACT, 2005.

[3] Richard van Nee and Ramjee Parsad, "*OFDM Wireless Multimedia Communications,*" Artech House, 2000.

[4] J. Heiskala and J. Terry, "*OFDM Wireless LANs: A Theoretical and Practical Guide,*" Sams, 2002.

[5] Fletcher, Hoyle, Patty, "*Foudations of Disiscrete Mathematics,*" PWS-KENT.

[6] A. K. Yeung, and I.M. Rabaey, "*A 210Mb/s Radix-4 Bit-level Pipelined Viterbi Decoder*", IEEE Int. Solid-State Circuit Conf., pp. 88-90, 1995.

[7] Chia-Hsin Lin "Design and Implementation of 802.11a/g OFDM-Based Outer Receiver". Thesis, National Chiao Tung University, Taiwan.

[8] Chien-Ching Lin, Chia-Cho Wu, and Chen-Yi Lee, "A Low Power and High Speed Viterbi Decoder Chip for WLAN Applications" Solid-State Circuits Conference, 2003. ESSCIRC '03. Proceedings of the 29th European

[9] Robert H. Morelos-Zaragoza, The Art of Error Correcting Coding, New York: John Wiley & Sons, 2002.

[10] Black, P.J.; Meng, T.H., “A 140-Mb/s, 32-state, radix-4 Viterbi decoder”, IEEE JOURNAL OF SOLID-STATE CIRCUITS.

[11] Lin Costello, “Error Control Coding 2/e”, Prentice Hall PEARSON, 2004.

[12] Y. Y. Xin, W. J. Xiang, L. F. Chang and Y. Y. Zheng, “VLSI design and implementation of high-speed Viterbi decoder”, Communications, Circuits and Systems and West Sino Expositions, IEEE 2002 International Conference on, Volume: 1, 29 June-1 July 2002.

[13] A. K. Yeung, and I.M. Rabaey, “A 210Mb/s Radix-4 Bit-level Pipelined Viterbi Decoder”, IEEE Int. Solid-State Circuit Conf., pp. 88-90, 1995.

[14] Suk-Jin Jung, Myeong-Hwan Lee#, and hyung-Jin Choi, “A New Survivor Memory Management Method In Viterbi Decoders: Trace-Delete Method and Its Implementation”, IEEE Int. Global Telecommunications Conf., pp. 3284-3286, 1996.

[15] Ko-Hui Lin “Design of FFT/IFFT module for Ultra Wideband System”. Thesis, National Chiao Tung University, Taiwan.

[16] IP Qualification 標準制定聯盟 ,”IP Qualification Guidelines” http://www.taiwanipgateway.org/IPQ/index.jsp

簡　　　　　　　　歷

姓名 ： 蔡彥凱

性別 ： 男

籍貫 ： 桃園縣

生日 ： 民國七十年六月二十號

地址 ：桃園縣中壢市龍岡里龍門街 161 巷 2 號

學歷 ： 國立交通大學電子工程研究所碩士班	93/09~95/06
國立成功大學機械工程學系	88/09~93/06
省立武陵高級中學	85/09~88/06

論文題目 ： Viterbi Decoder Design for Ultra-Wide band System

用於 UWB 設計之 Viterbi 解碼器