Chapter 3

Design of Outer Receiver

3.1 Demapping

3.1.1 Metric Generation Function

In chapter 2, we have discussed BPSK, QPSK, 16-QAM, and 64-QAM constellation mapping. According to the standard, the conversion adopts Gray-coded mapping. And white Gaussian noise probability distribution can be approximated to parabolic curve in a coordinate axis. For the hardware implementation complexity consideration, a simple metric generation Equation in form of 1st order linear function is proposed.

$$M_I = S_I \times I \pm Cn \tag{3.1}$$

M_I is the output metric of demapping, S_I is the slope, I is the input of demapping, and Cn is the constant. The larger the value indicates that we have more confidence on this bit. By employing Equation (3.1), BPSK, 16-QAM, and 64-QAM metric generation function are presented in Figure 8, Figure 9, and Figure 10, respectively. The bi represents the responding code bit. One of QPSK quadrature components is the same as BPSK. The metric value represents the quantification of the region from code bit 0 to code bit 1. So we only concern about either code bit 0 or code bit 1. In my design, code bit 1 methodology is adopted.

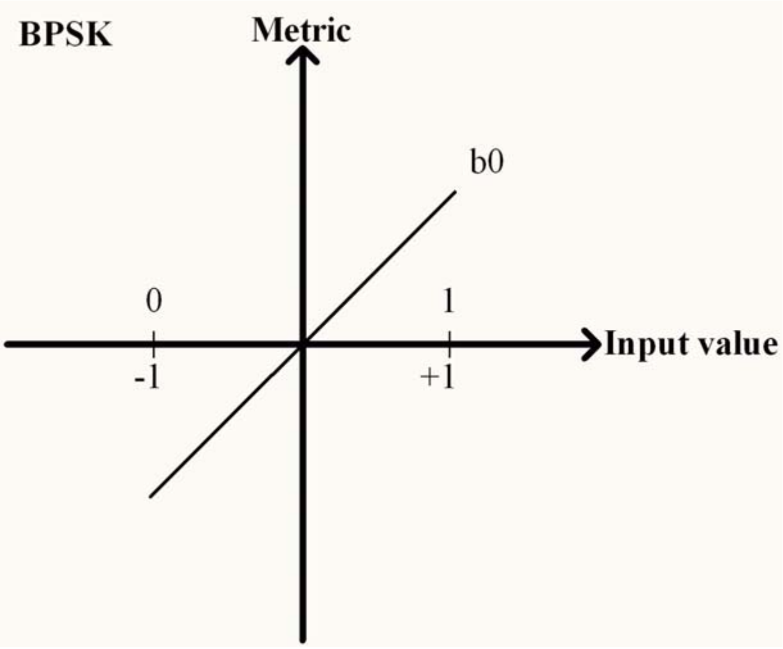


Figure 8: Metric function of BPSK

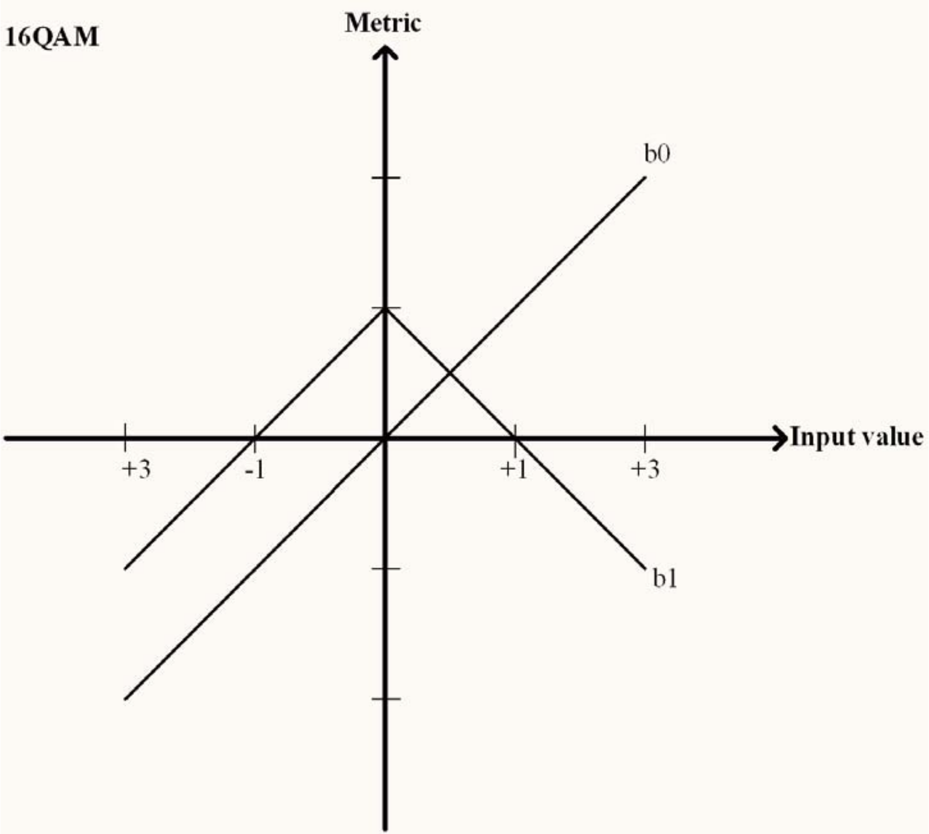


Figure 9: Two metric functions of one 16-QAM quadrature component

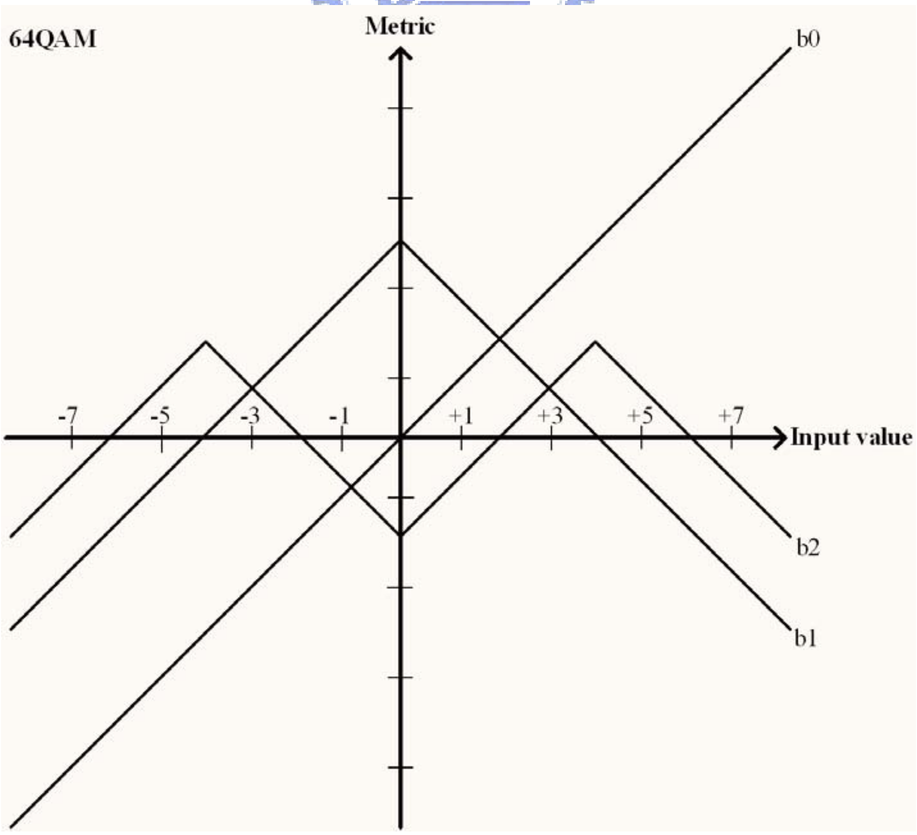


Figure 10: Three metric functions of one 64-QAM quadrature component

Trough observing Figure 8~10, it is straightforward that the metric generation function only suits with the range which code bit changes. And other range shall be set to maximum or minimum value of metric. Figure 11, 12 show the modified diagram for one quadrature component 64-QAM of code bit b0, b1, and b2, respectively.

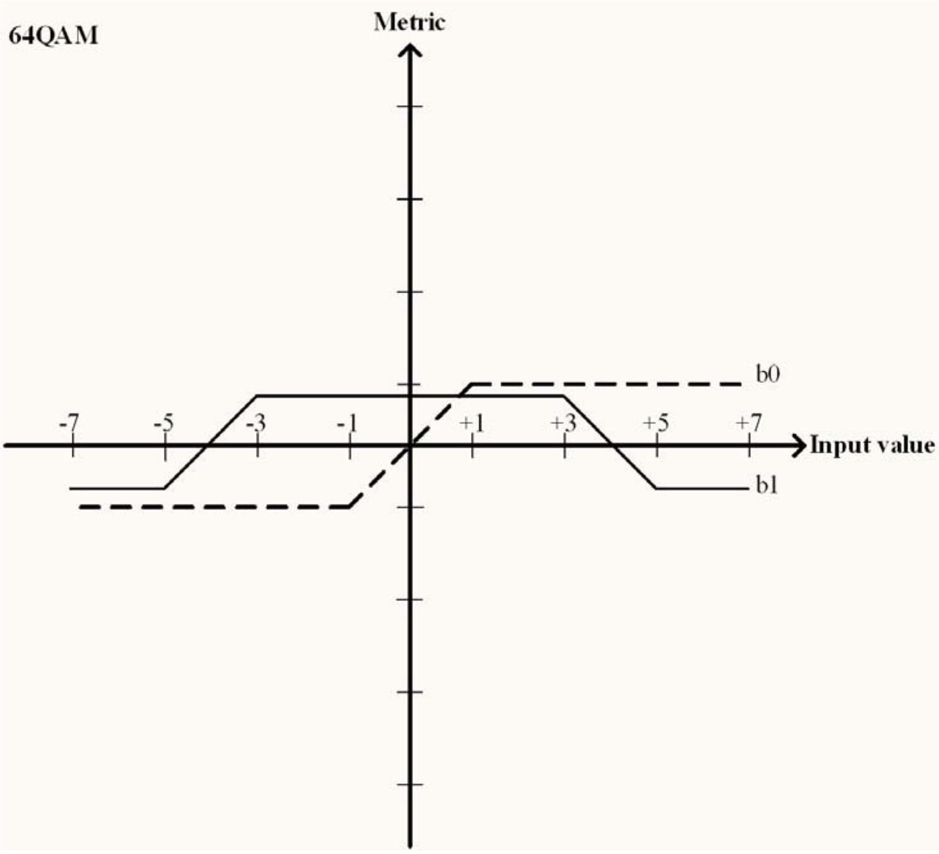


Figure 11: Two modified metric functions of one 64-QAM quadrature component

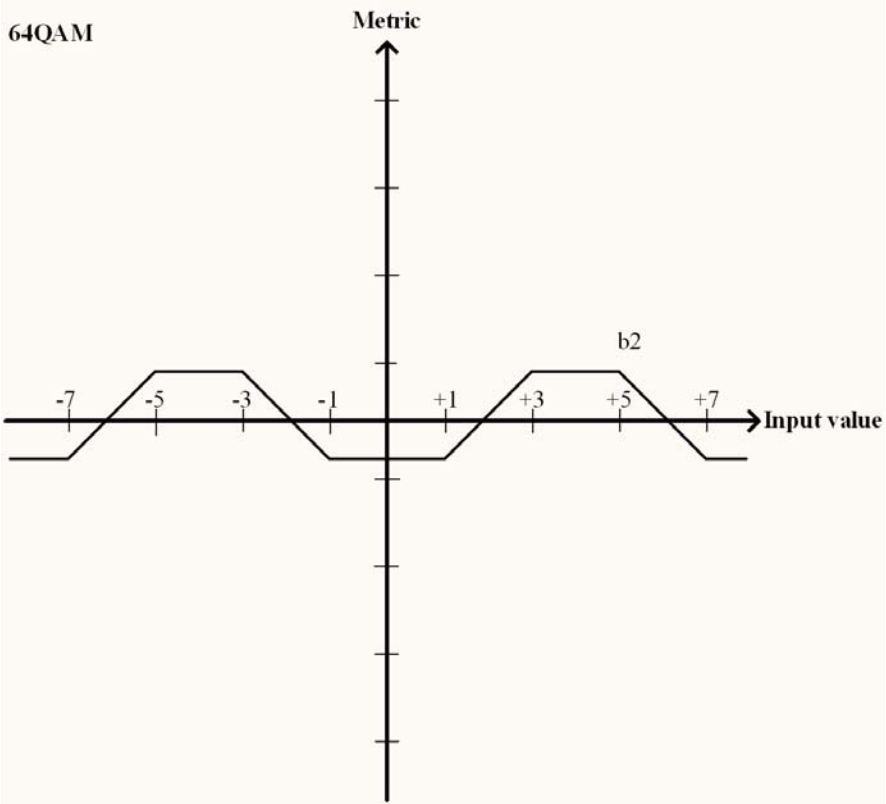


Figure 12: One modified metric functions of one 64-QAM quadrature component

3.1.2 Soft Decision

For high QAM technology applications, soft decision algorithm is usually adopted in order to get higher performance. Figure 13 shows the SNR vs BER compared with hard decision, soft decision 3-bit, 4-bit, 5-bit, and 6-bit. As shown in Figure 13 under 64-QAM and coding rate 3/4, the coding gain improvement between hard decision and soft decision are 3.2dB, 3.6dB, 3.75dB, 3.8dB, respectively. From 3-bit to 4- bit soft decision, the performance gets higher improvement, and the improvement of 4-bit to 6-bit soft decision is not larger than 3-bit to 4-bit. Therefore, 4-bit soft decision shall be adopted in this thesis. As shown in Figure 14, Figure 15, Figure 16, and Figure 17, we apply 4-bit soft decision to the demapping functions.

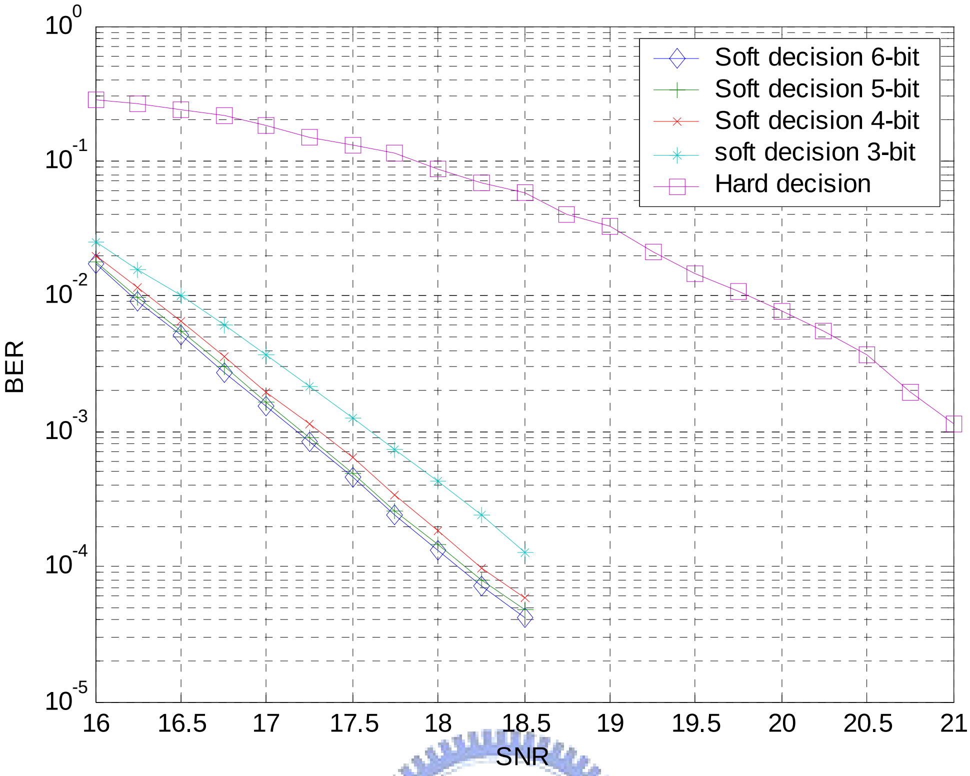


Figure 13: BER versus SNR for 3-bit, 4-bit, 5-bit, 6-bit soft decision, and hard decision

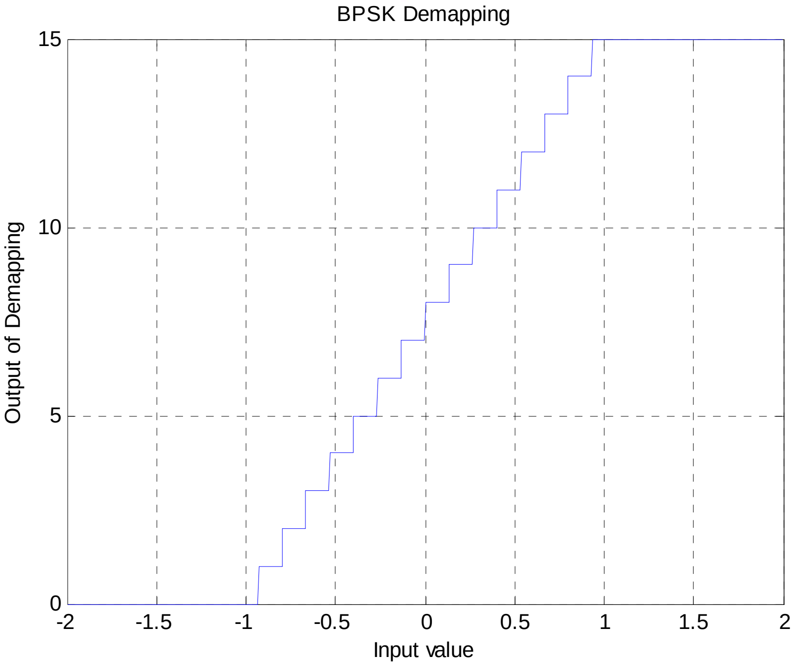


Figure 14: The demapping functions of BPSK for 4-bit soft decision

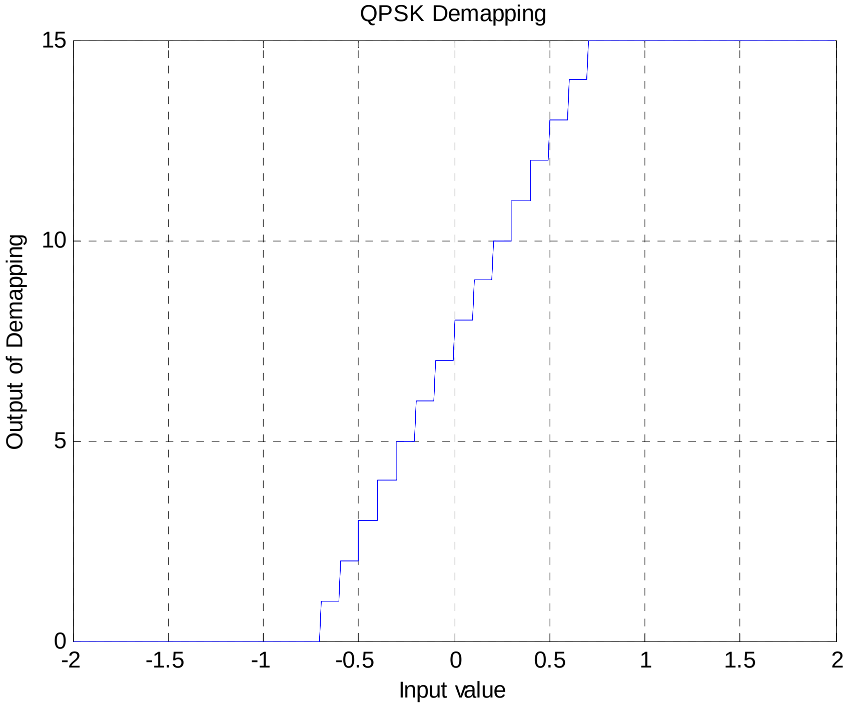


Figure 15: The demapping functions of QPSK for 4-bit soft decision

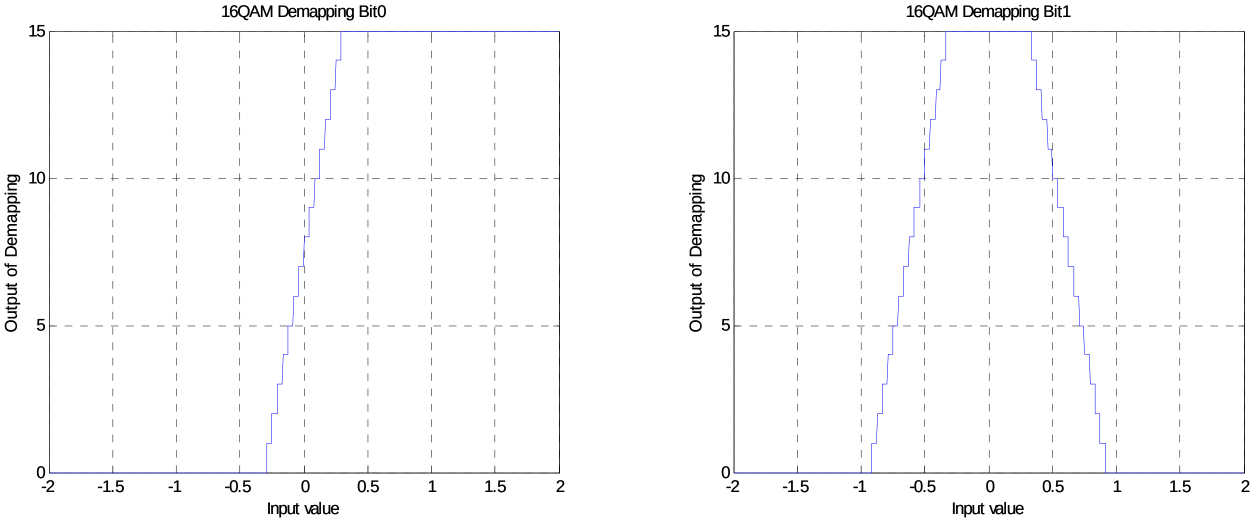


Figure 16: The demapping functions of 16-QAM for soft decision 4 bits

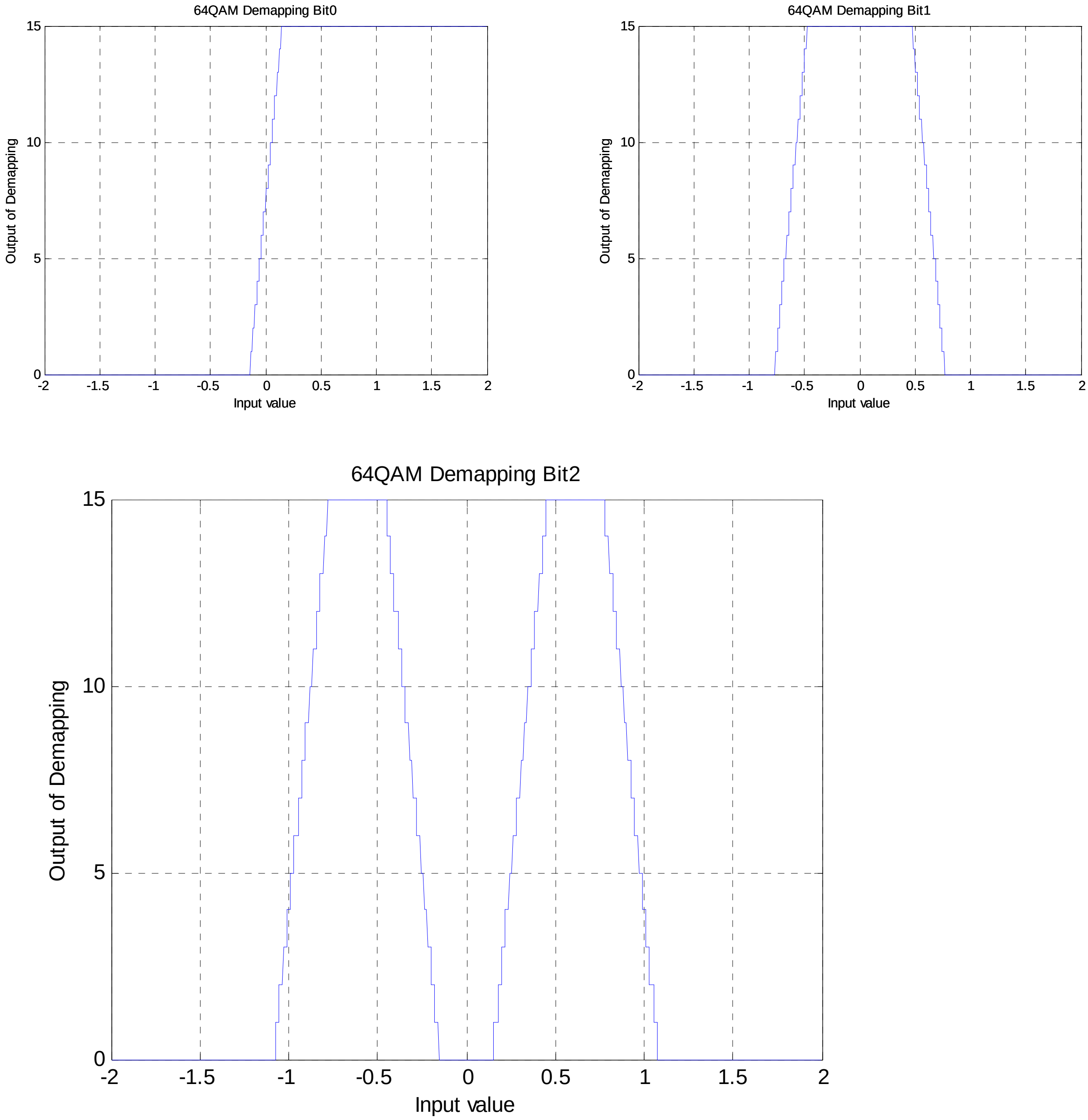
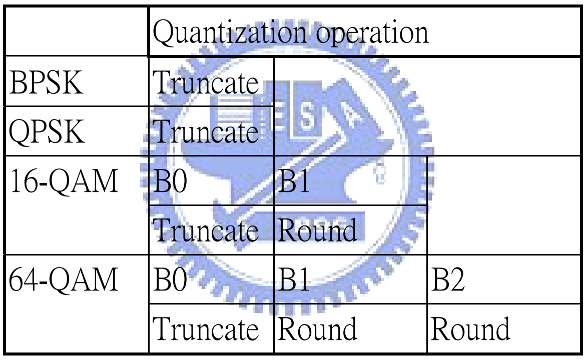


Figure 17: The demapping functions of 64-QAM for 4-bit soft decision

3.1.3 Quantization Effect

Another issue of demapping design is the problem produced by floor and round operations. Since the soft decision is adopted, the demapping function will be quantified as a stepladder form. The problem is that the middle point will shift left or right when we truncate or round the demapping function. The mismatch produces the unbalance of metric, and then we will lose some performance. Therefore, we shall choose proper quantization operation for demapping functions to eliminate the mismatch effect. Figure 18 shows the mismatch case of 16-QAM, and the results of quantization operation is presented in Table 4.

Table 4: The results of quantization operation



	Quantization operation		
BPSK	Truncate		
QPSK	Truncate		
16-QAM	B0	B1	
	Truncate	Round	
64-QAM	B0	B1	B2
	Truncate	Round	Round

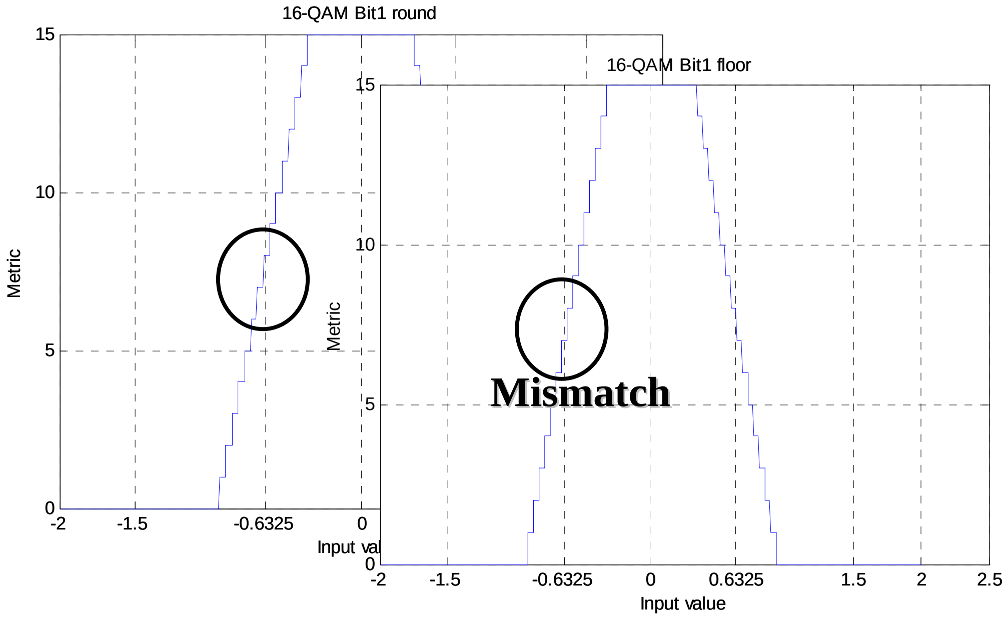


Figure 18: The mismatch of floor and round operation

3.1.4 Generalized Metric Generation Function

From Equation (3.1), we can derive the generalized metric generation function for other soft decision resolutions. We define *M* to be the resolution of soft decision.

$$0 \le M_I \le 2^M - 1 \tag{3.2}$$

The minimum value of metric is zero, and the maximum value is $2^M - 1$. And the threshold values of code bit range are I_{t0} and I_{t1}, respectively. Therefore, we set the slope value to be $S_I = \dfrac{2^M - 1}{I_{t1} - I_{t0}}$. Rewrite Equation (3.1) by replacing S_I, we obtain Equation (3.3).

$$M_I = \frac{2^M - 1}{I_{t1} - I_{t0}} \times I \pm C_n, I_{t0} \le I \le I_{t1} \tag{3.3}$$

3.2 Deinterleaver

From Table 3, the number of coded bits per OFDM symbol for different type modulations is 48, 96, 192, and 288, respectively. For the data capacity of different modulation types, it is sure that the deinterleaver employs the maximum value as the buffer size. Another role of the deinterleaver in the proposed outer receiver is set to be a control unit. The deinterleaver shall enable the following modules for 8 different data rate in different time interval.

3.3 Depuncture

3.3.1 Traditional Method

The depuncture module shall insert a dummy “zero” metric into the convolutional decoder on the receive side in place of the stolen bit. Because we use unsigned number to represent the metric value, we shall insert a middle value between maximum and minimum value in order to replace the dummy “zero” metric. As shown in Figure 19, the dummy metric value is described.

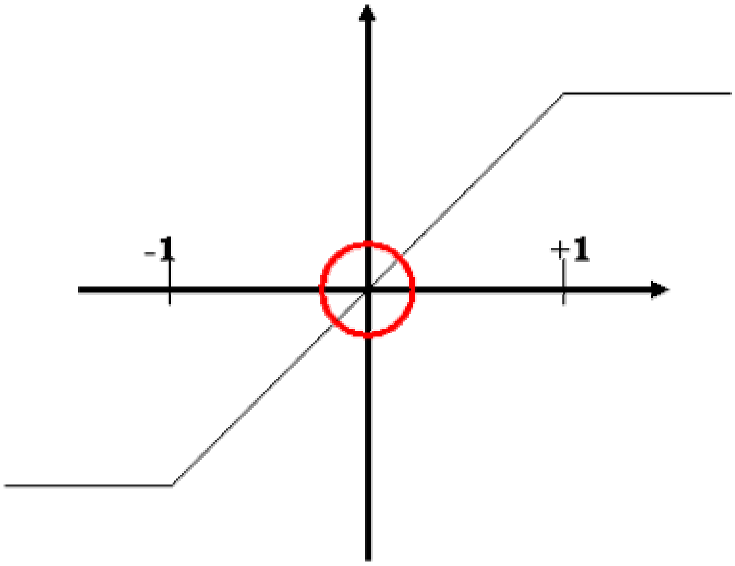


Figure 19: Dummy metric value

3.3.2 Modified Method

Through the quantization of metric, there is one problem about unbalance of metric given by dummy metric value. For coding rate 3/4 or 2/3 case, there are always dummy metric terms of depuncture for the input of Viterbi decoder. The effect of dummy metric terms is equivalent to produce error to Viterbi decoder. For example, the coding rate 3/4 depuncture pattern responding to trellis diagram is shown in Figure 20. According to Figure 20, the branches metric calculations are listed in Equation (3.4). In Equation (3.5), the depuncture metric terms inserted.

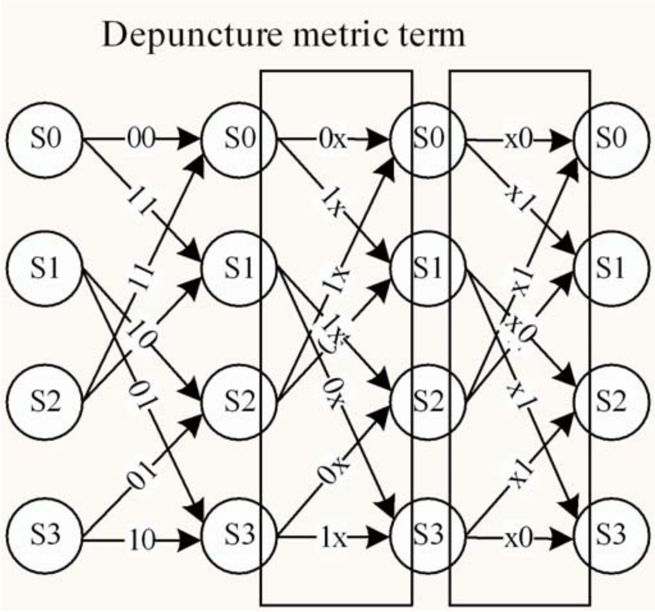


Figure 20: Trellis diagram for depuncture metric term

$$\begin{cases} M_{0,0} = (X_I - X_{r,0})^2 + (Y_I - Y_{r,0})^2 + \Gamma_0 \\ M_{1,1} = (X_I - X_{r,1})^2 + (Y_I - Y_{r,1})^2 + \Gamma_2 \end{cases} \tag{3.4}$$

$$\begin{cases} M_{0,x} = (X_I - X_{r,0})^2 + (Y_{dep} - Y_{r,0})^2 + \Gamma_0 \\ M_{1,x} = (X_I - X_{r,1})^2 + (Y_{dep} - Y_{r,1})^2 + \Gamma_2 \end{cases} \tag{3.5}$$

Γ_i is the previous stage survival metric of state i. $M_{i,j}$ is the metric of the branch of (i,j). $X_{r,i}$ and $Y_{r,i}$ are the reference metric of code bit i. Ideal depuncture dummy metric shall give the middle point for the balance of metric, and then the square term shall be equal to other branch square term. Since Viterbi algorithm includes comparison with two metrics, the two square terms will be eliminated. In depuncture trellis diagram, all branches exist the same square term. So we find that the dummy metric term shall not be inserted into Viterbi decoder and it shall be replaced by zero metric.

3.4 Viterbi Decoder

An implementation of the Viterbi algorithm, referred to as Viterbi decoder, consists of several functional blocks. A branch metric calculation (BMC) unit, which is responsible for calculating the set of branch metrics for each time instant, an add-compare-select (ACS) unit, which is responsible for generating the decisions (survivors) made by the decoder and the metrics for next stage. There are two methods which extract decoded bits known as register-exchange (RE) and trace-back (TB). In the literature, the RE technique is suitable for trellises with only a small number of states, whereas the TB approach is acceptable for trellises with a large number of states. We choose the TB method instead of RE method because RE consumes more power than TB does [5] [6].

3.4.1 Branch Metric Calculation

3.4.1.1 Metric of Euclidean Distance

Since the soft decision algorithm has been discussed in section 3.1, the traditional metric is Euclidean distance. Each instant time the data sequences are received by Viterbi decoder, and the data sequences will be divided into real part and imaginary part (X , Y). The branch metric equation is described as

$$M_{t0\to tN} = \sum_{i=0}^{N}(X_i - X_r)^2 + (Y_i - Y_r)^2 \propto \sum_{i=0}^{N}\sqrt{(X_i - X_r)^2 + (Y_i - Y_r)^2} \quad (3.6)$$

(X_r, Y_r) is the reference symbol. Bit 1 means metric value being $2^m - 1$, and Bit 0 means metric value being zero. Although the Euclidean distance should be calculated in Equation (3.6), but the hardware implementation of Euclidean distance requests an extra ROM to achieve the square root function. We calculate the fixed point operation of positive-relationship of Euclidean distance as the right term. We may obtain 0.2dB performance better than the right term. Figure 21 shows the SNR versus BER of BMC applying square root function or not under 4-bit soft decision, 64-QAM, and coding rate 3/4.

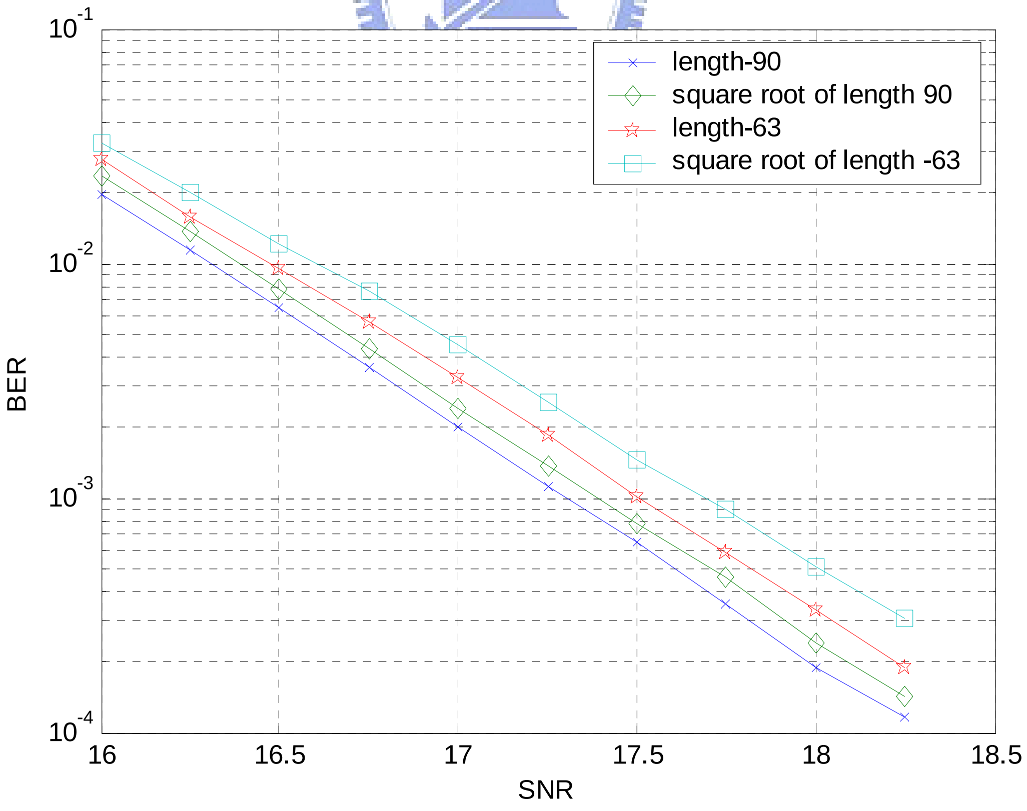


Figure 21: SNR versus BER for square-root case and normal case

3.4.1.2 Metric of Correlation

Besides applying Euclidean distance for metric unit, another scheme can be derived from Euclidean distance equations. It is described as follows:

$$\begin{cases} M_{up} = (X_I - X_{r,0})^2 + (Y_I - Y_{r,0})^2 \\ M_{down} = (X_I - X_{r,1})^2 + (Y_I - Y_{r,1})^2 \end{cases} \tag{3.7}$$

Derive Equation (3.8) from expanding Equation (3.7).

$$\begin{cases} M_{up} = (X_I{}^2 - 2X_I X_{r,0} + X_{r,0}{}^2) + (Y_I{}^2 - 2Y_I Y_{r,0} + Y_{r,0}{}^2) \\ M_{down} = (X_I{}^2 - 2X_I X_{r,1} + X_{r,1}{}^2) + (Y_I{}^2 - 2Y_I Y_{r,1} + Y_{r,1}{}^2) \end{cases} \tag{3.8}$$

By eliminating the same term, Equation (3.9) is derived.

$$\begin{cases} M'_{up} = (-2X_I X_{r,0} + X_{r,0}{}^2) + (-2Y_I Y_{r,0} + Y_{r,0}{}^2) \\ M'_{down} = (-2X_I X_{r,1} + X_{r,1}{}^2) + (-2Y_I Y_{r,1} + Y_{r,1}{}^2) \end{cases} \tag{3.9}$$

The reference metrics $X_{r,0}$, $X_{r,1}$, $Y_{r,0}$, and $Y_{r,1}$ can be taken as constant. And X_I and Y_I are discrete random variables. So the calculation of BMC can be viewed as correlator. The metric terms consist of real part and imaginary part, and can be calculated independently. The branch metric calculation equations are represented as:

$$\begin{cases} M'_{up} = M_{b0,0} + M_{b1,0} \\ M'_{down} = M_{b0,1} + M_{b1,1} \end{cases} \tag{3.10}$$

Table 5 shows the conversion of X_I under bit 1 and bit 0, respectively. For example, the case which the decoder receives (0,3) symbol is shown in Figure 22.

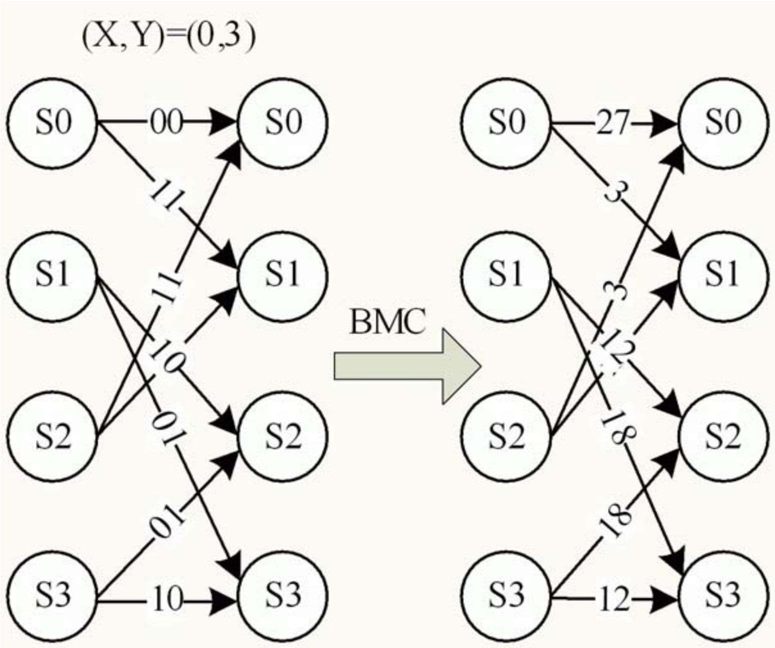
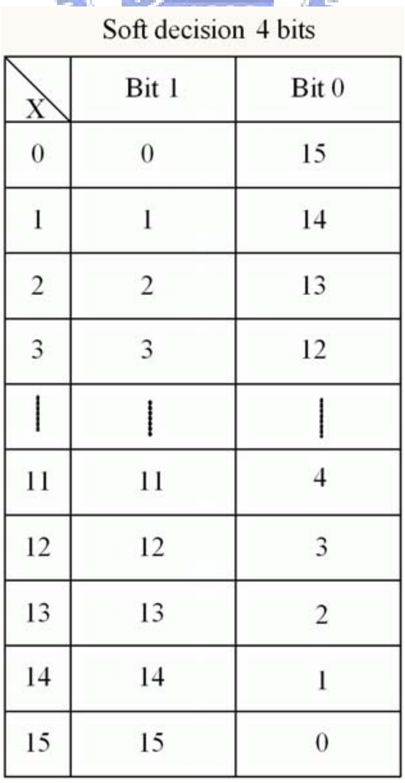


Figure 22: The example of the received symbol (0,3)

Table 5: The mapping of metrics



Soft decision 4 bits

X	Bit 1	Bit 0
0	0	15
1	1	14
2	2	13
3	3	12
⋮	⋮	⋮
11	11	4
12	12	3
13	13	2
14	14	1
15	15	0

The proposed correlator algorithm for BMC focuses on that the larger value of metric, the more likely path in the trellis. In the literature, it is the original definition of the correlation. It is different from the traditional algorithm of Euclidean distance. It is sure that we also can employ the reverse operation to find the minimum metric of the paths.

3.4.2 Add-Compare-Select

IEEE 802.11a uses (2, 1, 7) convolutional code as its channel encoder, and the trellis diagram of Viterbi decoder can be decomposed into basic function blocks. The basic function block is called the radix-2 ACS unit. We take the trellis diagram of four states as an example in Figure 23. $\Gamma_{t-1,s0}$ is the previous survival metric, $\lambda_{t,s0->s0}$ is the branch metric from state 0 to state 0, and $\Gamma_{t,s0}$ is the instant metric of the ACS unit. For the application with the large number of trellis state, the fully parallel architecture of ACS causes huge gate count in hardware implementation.

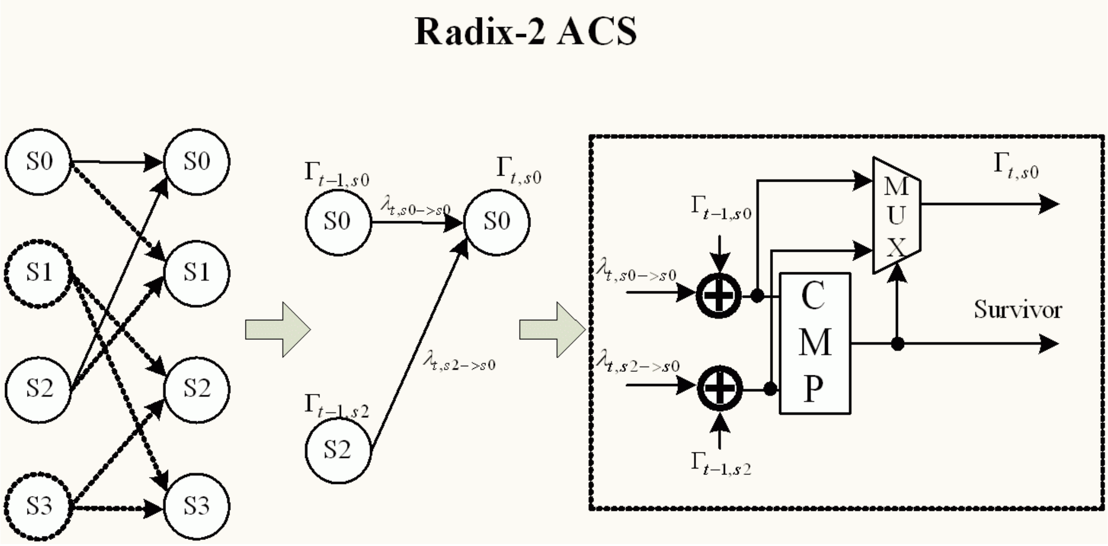


Figure 23: Radix-2 ACS function block

The main consideration of ACS architecture design is the trade off between the number of ACS stage and the decoder throughput. Several kinds of the ACS unit are proposed to achieve the different applications. For low data rate applications, we even can complete the same decoding work by parts of the fully parallel architecture of ACS. But for high data rate applications, we have to afford the cost of high complexity hardware. It is well known that high radix ACS unit is proposed to improve the decoding throughput. Actually, the high radix ACS concept is to decode several received symbols each instant time. Figure 24 depicts the conversion from two-stage radix-2 ACS to one-stage radix-4 ACS. Figure 25 depicts the conversion from three-stage radix-2 ACS to one-stage radix-8 ACS for 8 states trellis. Because we use 3-stage radix-2 ACS architecture, the word-length optimization of ACS-stage registers can be completed.

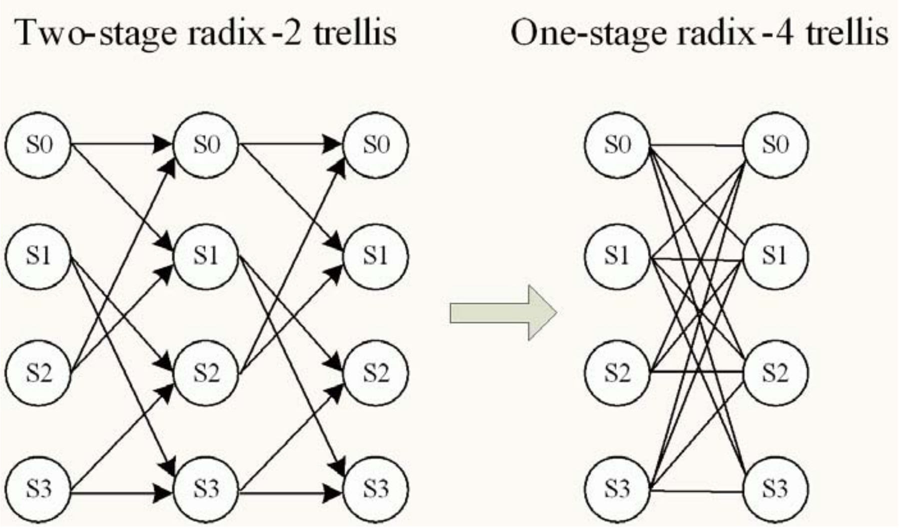


Figure 24: Two-stage radix-2 trellis to one-stage radix-4 trellis

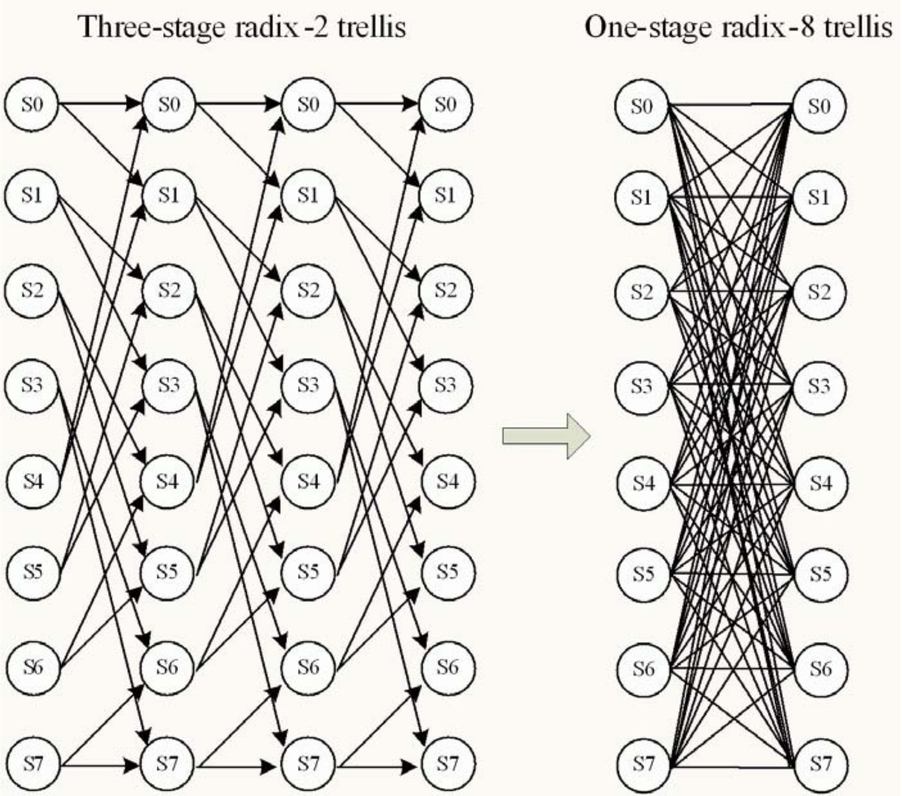


Figure 25: Three-stage radix-2 trellis to one-stage radix-8 trellis

3.4.3 Traceback

For the decoding scheme, we adopt the well-known traceback method, which is described as follows [7] [8]:

By using ACS methodology, the survivors can be obtained. The survivor means that which way the path goes from the previous state to the present state. So we usually store the survivors from ACS in memory unit called SMU. For example, the received symbol is [00 00 11 01 00 01]. When we receive the specified number of symbols, we start to trace the path which has the minimum distance. We depend on the survivors to decide the next path we trace back. Figure 26 shows the traceback operation.

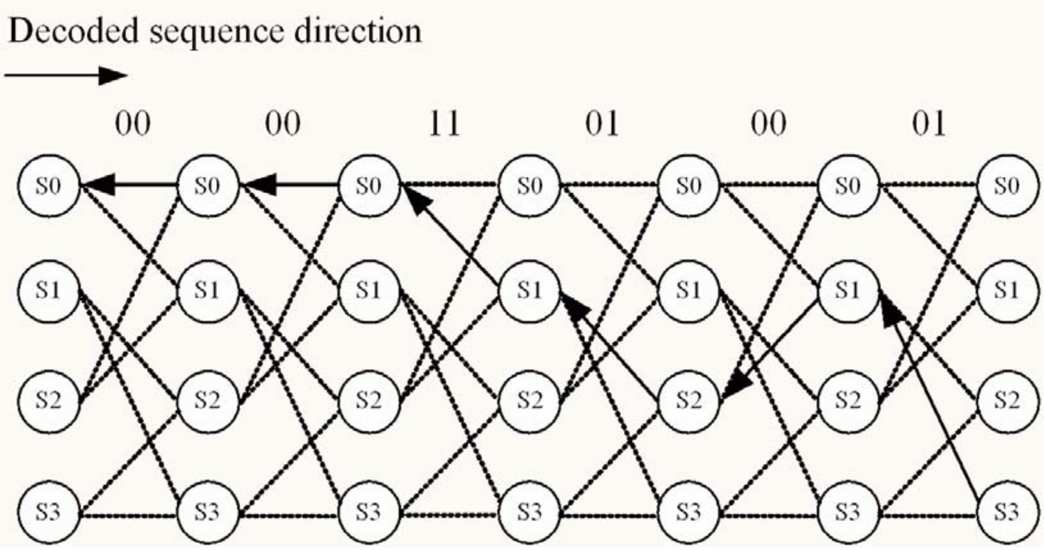


Figure 26: The traceback operation

In this example, we pay attention on that we start to trace back the path at 7th symbol. The traceback parameter is usually called traceback-length or traceback-spread. It means how many symbols we received, and we start to trace back the path. Traceback length is an important parameter when we implement Viterbi decoder. So we simulate the performance BER versus SNR with different traceback length in AWGN as follows.

3.5 Simulations

3.5.1 Simulation Environment

The simulation environment is set up with MATLAB. The soft decision resolution adopts 16-level (4bits). Because the 64QAM is the most critical case, we simulate the case about BER versus SNR and PER versus SNR. The PER simulation is done at a length of 1024 bytes.

3.5.2 Simulation Result

64-QAM Case

As depicted in Figure 27, BER versus SNR simulation for different traceback-length is presented. It is clear that the case of from traceback-length-51 to traceback-length 63 has more coding gain improvement than other cases. Because 64-QAM is the most critical case, we adopt traceback-length 63 for the proposed Viterbi decoder. Figure 28 shows the PER versus SNR for different traceback-length.

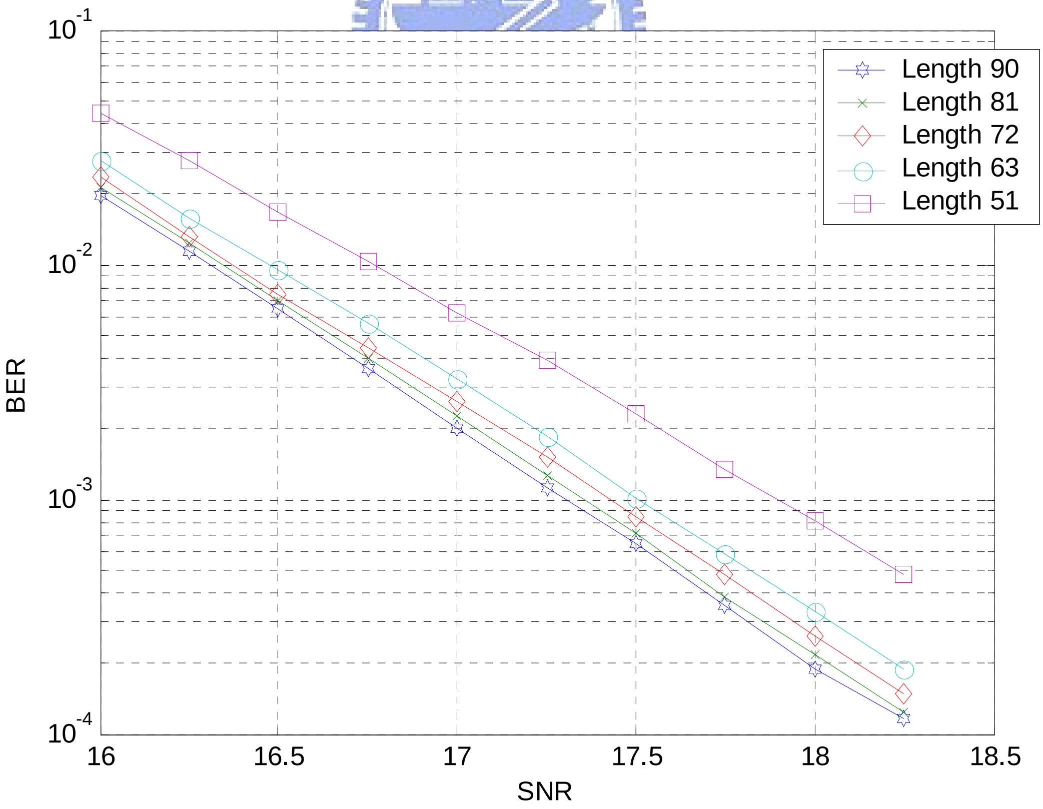


Figure 27: BER versus SNR for different traceback-length

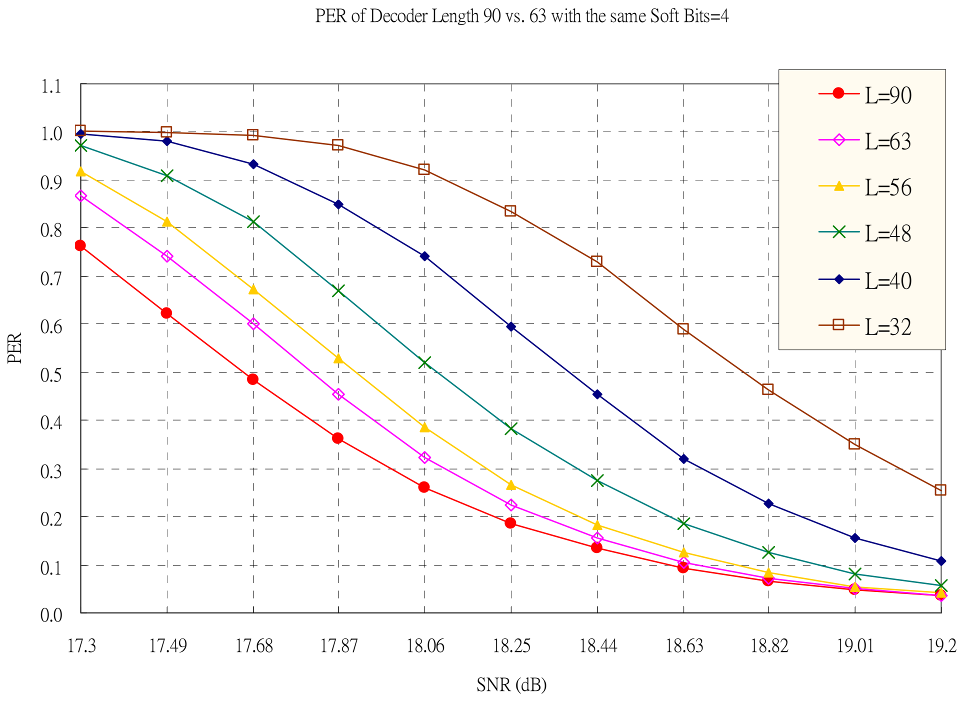


Figure 28: PER versus SNR for different traceback-length