Architecture of RS decoder for MB-OFDM UWB

Sung-Woo Choi *, Sang-Sung Choi *, Han-Ho Lee**
Wireless Home Network Research Team, ETRI *
csw9908, sschoi@etri.re.kr*
In-ha University **
hhlee@inha.ac.kr **

Abstract - UWB is the most spotlighted wireless technology that transmits data at very high rates using low power over a wide spectrum of frequency band. UWB technology makes it possible to transmit data at rate over 100Mbps within 10 meters. To preserve important header information, MB-OFDM UWB adopts Reed-Solomon(23,17) code. In receiver, RS decoder needs high speed and low latency using efficient hardware. In this paper, we suggest the architecture of RS decoder for MB-OFDM UWB. We adopts Modified-Euclidean algorithm for key equation solver block which is most complex in area. We suggest pipelined processing cell for this block and show the detailed architecture of syndrome, Chien search and Forney algorithm block. At last, we show the hardware implementation results of RS decoder for ASIC implementation.

Keywords: UWB, Reed-Solomon, Channel code

1 Introduction

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 ideally suited for wireless communications, particularly short-range and high-speed data transmissions for local area network applications. This technology has advantage of high speed enabling multimedia streaming in the home. UWB is being standardized in IEEE 802.15.3a task group. MB-OFDM UWB and DS-UWB are competing for UWB standard. So, both sides are now producing UWB modem chip and struggling to spread the commercial use to take the initiative on the market.

This paper has a focus on MB-OFDM between these two proposals and deals with the hardware implementation of RS decoder, enabling to protect the header from impairment. The data frame of MB-OFDM UWB consists of PLCP preamble, PLCP header and PSDU. In modem receiver, PLCP preamble and header are used to perceive the signal and then starts to synchronize received signal. The PLCP header includes important information such as rate and length that are used at PHY and MAC layer. PLCP header structure is illustrated in Fig. 1 [1].

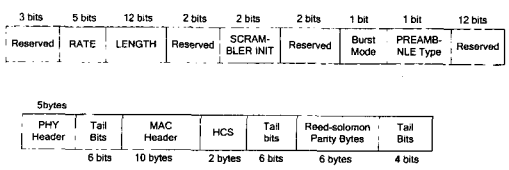

Fig. 1. PLCP Header structure

In modem transmitter, Reed-Solomon code is adopted for outer code to protect header data from impairment. In Fig. 1, PHY header, MAC header and HCS, 17 bytes, are scrambled and encoded using RS(23,17) code and residual 6 parity bytes are appended. This code enables to correct maximum 3 bytes error including parity bytes. This RS(23,17) code is shortened form of RS(255,249) code. The generator polynomial of RS(255,249) is as follows.

$$g(x)=\prod_{i=1}^{6}(x-\alpha^{i})=x^{6}+126x^{5}+4x^{4}+158x^{3}+58x^{2}+49x+117 \quad (1)$$

Transmitting codeword polynomial $c(x)$ has relation with message $m(x)$ and generator polynomial $g(x)$ as follows.

$$c(x)=x^{N-k}m(x)+\left[x^{N-k}m(x)\bmod g(x)\right]=q(x)g(x) \quad (2)$$

Because RS code is one of the systematic code, MB-OFDM uses it as mandatory code in transmitter but optional in receiver. So, in receiver, the estimation about additional hardware area and delay when it is implemented is required. In this point, this paper introduce RS decoding algorithm which is suitable to MB-OFDM UWB specification and suggest the hardware architecture.

2 Modified Euclidean Algorithm

Let GF(2^m) be the finite field of 2^m elements. Also, let N = 2^m-1 be the length of the RS(N, k) code over GF(2^m) with

minimum distance d = 2t+1 , where I = N-(d-1) denotes the number of m-bit message symbols and t denotes the number of errors that may be corrected in each codeword [2].

Let $c(x)$ and $r(x)$ be the codeword polynomial and the received polynomial, respectively. The transmitted polynomial can be corrupted by channel noise during transmission. There, the received polynomial can be described as $r(x) = c(x) + e(x)$, where $e(x)$ is the error polynomial.

RS decoding process has three stages, syndrome calculation, solving key equation, evaluating error location and error value. The first step in the decoding algorithm is to calculate 2t syndromes, S_i $(1 \le i \le 2t)$, which are used to correct correctable errors. S_i is computed as

$$S_j = r(\alpha^j) = \sum_{i=0}^{N-1} r_i (\alpha^j)^i \,, 1 \le j \le 2t \tag{5}$$

where α is a root of a generator polynomial. From above equation, since transmitted codeword $c(x)$ is multiple type between $g(x)$ and $q(x)$, α^j, which is the root of $g(x)$, makes $c(\alpha^j) = 0$. This leads to

$$S_j = r(\alpha^j) = c(\alpha^j) + e(\alpha^j) = e(\alpha^j) \tag{6}$$

That is, if syndrome polynomial was not zero, the received signal is corrupted. Syndromes are used to find error location and error value. Syndrome polynomial $S(x)$ has relation with the error locator polynomial $\Lambda(x)$ and the error value polynomial $\Omega(x)$ as

$$S(x) \cdot \Lambda(x) = \Omega(x) \bmod x^{2t} \tag{7}$$

which is called key equation. If there was v error in received signal, the error locator polynomial is

$$\Lambda(x) = \prod_{l=1}^{v} (1 + xX_l) = \lambda_0 + \lambda_1 x + \cdots + \lambda_v x^v \tag{8}$$

and the error value polynomial is

$$\Omega(x) = \omega_0 + \omega_1 x^1 + \cdots + \omega_{v-1} x^{v-1} \tag{9}$$

To solve key equation, there are several algorithms like Barlekamp-Massey(BM) algorithm[3], Euclidean algorithm[4]. Both algorithms require Galois field multiplication and division operations on the each iteration. Hence, they are not suitable for high-speed implementations. A Modified-Euclidean(ME) algorithm was proposed to decrease this complexity. The ME algorithm can be implemented with only inverse ROM corresponding to each symbol instead of division between polynomial. Without considering about erasure, ME algorithm is follows [5][6]:

At $i = 0$, let initial conditions be

$$\begin{aligned} R_0(x) &= x^{2t} & Q_0(x) &= S(x) \\ L_0(x) &= 0 & U_0(x) &= 1 \end{aligned} \tag{8}$$

After setting initial value, if $i \ge 1$, then compute recursively

$$\begin{aligned} R_i(x) &= \left[\sigma_{i-1} b_{i-1} R_{i-1}(x) + \bar{\sigma}_{i-1} a_{i-1} Q_{i-1}(x)\right] \\ &\quad - x^{|l_{i-1}|}\left[\sigma_{i-1} a_{i-1} Q_{i-1}(x) + \bar{\sigma}_{i-1} b_{i-1} R_{i-1}(x)\right] \\ Q_i(x) &= \sigma_{i-1} Q_{i-1}(x) + \bar{\sigma}_{i-1} R_{i-1}(x) \\ L_i(x) &= \left[\sigma_{i-1} b_{i-1} L_{i-1}(x) + \bar{\sigma}_{i-1} a_{i-1} U_{i-1}(x)\right] \\ &\quad - x^{|l_{i-1}|}\left[\sigma_{i-1} a_{i-1} U_{i-1}(x) + \bar{\sigma}_{i-1} b_{i-1} L_{i-1}(x)\right] \\ U_i(x) &= \sigma_{i-1} U_{i-1}(x) + \bar{\sigma}_{i-1} L_{i-1}(x) \end{aligned} \tag{9}$$

where,

$$\begin{aligned} l_{i-1} &= \deg(R_{i-1}(x)) - \deg(Q_{i-1}(x)) \\ \sigma_{i-1} &= 1 & l_{i-1} &\ge 0 \\ \sigma_{i-1} &= 0 & l_{i-1} &< 0 \end{aligned} \tag{10}$$

a_{i-1} and b_{i-1} are the leading coefficients of $R_{i-1}(x)$ and $Q_{i-1}(x)$, respectively. The algorithm stops when $\deg(R_{i-1}(x)) < t$. Then, the error locator polynomial $\Lambda(x)$ and the error value polynomial $\Omega(x)$ is

$$\Lambda(x) = L_i(x) \,, \quad \Omega(x) = R_i(x) \tag{11}$$

After solving key equation, the error locator polynomial $\Lambda(x)$ is fed into the Chien search algorithm block, which calculates the roots of the error locator polynomial. From input α^{i_l} for $0 \le i_l \le t-1$, if $\Lambda(\alpha^{i_l}) = 0$, then $X_l = \alpha^{-i_l}$ is the location of error. Error value is calculated by the Forney algorithm. Using the Forney algorithm, error value is

$$Y_l = \left.\frac{\Omega(x)}{\Lambda'(x)}\right| \text{ at } x = \alpha^{i_l} \tag{12}$$

where $\Lambda'(x)$ is the derivative of $\Lambda(x)$.

The Forney algorithm works in parallel with the Chien search algorithm to calculate the magnitude of the error symbol at each error location. Fig. 2 is block diagram of RS decoder using ME algorithm.

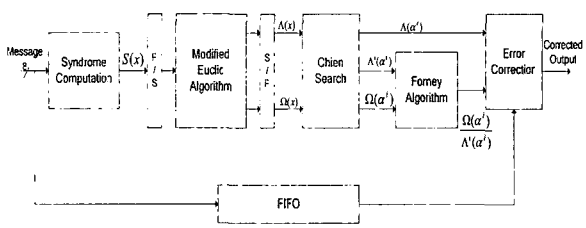

Fig. 2. RS decoder architecture

3 Reed-Solomon Decoder Architecture

3.1 Syndrome calculation

Let the input signal of the RS(23,17) decoder be $v(x)$.

$$v(x) = v_{22}x^{22} + v_{21}x^{21} + v_{20}x^{20} + \cdots + v_1 x + v_0 \qquad (13)$$

Syndrome is computed by insert α^i (i=1, 2, ... , $2t$) which are the root of generator polynomial to $v(x)$.

$$\begin{aligned} S_1 &= v(\alpha) \\ S_2 &= v(\alpha^2) \\ &\cdots\cdots \\ S_6 &= v(\alpha^6) \end{aligned} \qquad (14)$$

Fig. 3 is syndrome calculation cell and Fig. 4 is the architecture of syndrome calculation block using this cell.

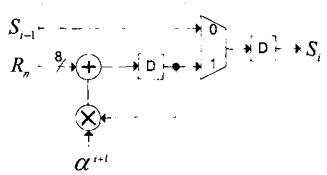

Fig. 3. Syndrome calculation cell

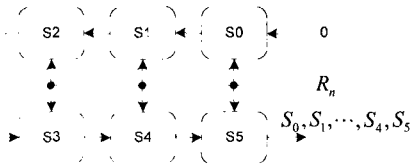

Fig. 4. Syndrome calculation block

Each cell which is composed of GF(2^8) constant multiplier and adder, multiplexer and register, computes syndromes for 23 bytes of input symbol and then outputs syndromes from the cell S5 to the ME calculation block.

3.2 ME Calculation Block

In the case of other RS decoder with higher error correcting capability, KES block consumes 60~70% of hardware of all blocks. But, this system uses (23, 17) RS code correct up to 3 errors. So, KES block of this system consumes much less hardware than other system. So, in-out latency is the major factor to resolve the excellence of a RS decoder.

For ME algorithm block, we choose systolic array architecture of linked algorithm processing cell which is easily pipelined, therefore, it is more suitable to high speed operation. Because t=3 , total 6 ME processing cell are used. Fig. 5 is the block diagram of ME calculation block.

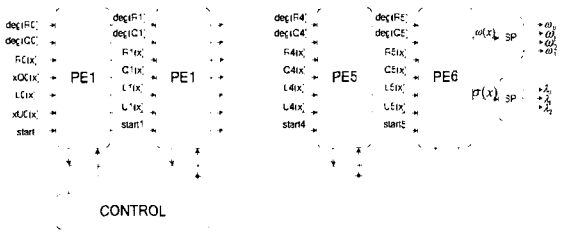

Fig. 5. ME calculation block

We proposed pipelined, low latency ME calculation cell for ME algorithm. This cell is composed of degree computation block and processing arithmetic block [7]. The degree computation block processes several operations such as the degree computation, the degree update using subtraction and assert the STOP signal to next cell when stop condition is satisfied.

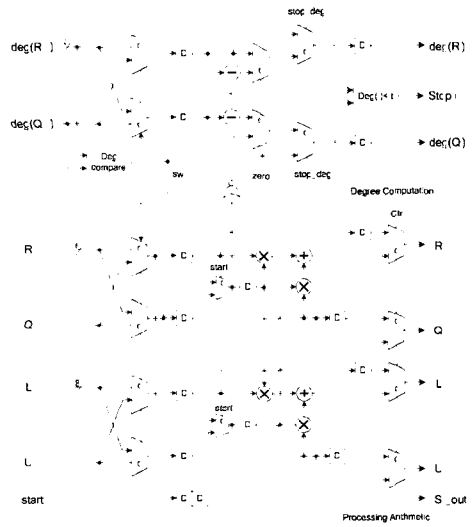

Fig. 6. ME calculation cell

In the ME calculation cell, the input data is computed and transferred to output after 2 delays. So, total output delay of the ME calculation block is 12. In fig. 5, the control block examine the STOP signals from all cells and decide when the output buffer outputs the error locator polynomial and the error value polynomial.

3.3 Chien search and Forney block

The Chien search block sets initial coefficients using error locator polynomial and inputs α^n, ($233 \le n \le 255$) to the polynomial and calculates $\Lambda(\alpha^n)$. If $\Lambda(\alpha^n) = 0$, then α^{-n} is the error location. Because α^{-n} satisfies the relation such as $\alpha^{-n} = \alpha^{(N-n) \bmod N}$ in Galois field (2^m), if $\Lambda(\alpha^{233}) = 0$, then the first symbol v_{22} is the error location.

Also, the Chien search block calculates $\Lambda'(\alpha^i)$ for the use of Forney block. In Galois field, the derivative of even term is zero. If let be $\Lambda(x)=\Lambda_{even}(x)+\Lambda_{odd}(x)$ then, it satisfies $\Lambda_{odd}(x)=x\cdot\Lambda'(x)$. So, as described in Fig. 8, $\Lambda_{even}(x)$ and $\Lambda_{odd}(x)$ are calculated in each path and combined to make $\Lambda(x)$. This can eliminate the circuit of calculating $\Lambda(\alpha^i)$ [6]. The circuit of i-th Chien search cell is shown in Fig. 7. Fig. 8 is the shows the block diagram of the Chien search block with three Chien search cells.

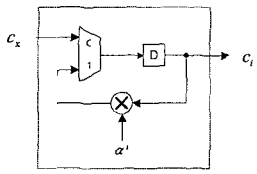

Fig. 7. Chien search cell

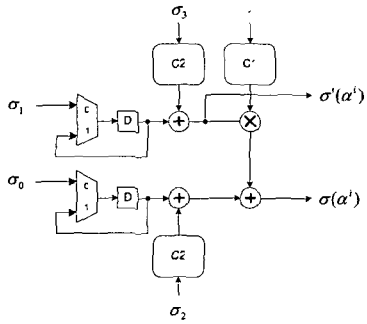

Fig. 8. Chien search 블록

The Forney block computes $\Omega(\alpha^i)$ using the error value polynomial from the ME block. The Forney block contains the same cell with the Chien search cell. The Forney block generates error value due to formula (12). Fig. 9 is the block diagram of the Forney block and the error-correction block. From the Fig. 9, the division is composed of inverse ROM of the input value and a multiplication. This ROM is composed of 255 words of 8 bits which contains inverse value of the input in GF(2^8). The error value is connected to a AND gate which operates as a switch which is ON when $\Lambda(\alpha^n)=0$. The output of FIFO is corrected by XOR operation.

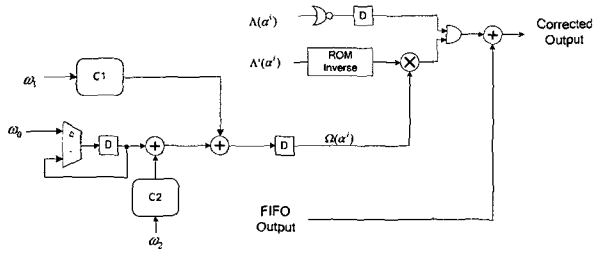

Fig. 9. Forney block and error correction block

4 Hardware implementation result

The proposed RS decoder was modeled in Verilog hardware description language(HDL) and simulated to verify its functionality. After that, it was synthesized using TSMC 0.18um library. The latency is 46 symbol clocks and the FIFO of RS decoder is made by dual port RAM of the same size to the latency. From synthesis result, the area estimation is 27k gates and it can operate at clock frequency of 232MHz, which means it has a data processing rate of 1.6Gbps. The result shows high speed RS decoder can be implemented using small area.

Also, we expanded the RS code to the data field using a RS(255,239) code and estimated the hardware. This expansion can be easily done because the proposed architecture has modular architecture. The estimated area increased to 78 gates. The ME calculation block has more area than the architecture of using recursive cell but has less latency. Compared to high pipelined ME block, it has less speed but has less latency and area.

5 Conclusion

In this paper, we researched the RS(23,17) code of MB-OFDM UWB. We showed the appropriate decoding algorithm and suggested the architecture with small area and low latency. From synthesis result, the area estimation is 27k gates and the latency is 46 symbol clocks. The proposed architecture adopts Modified-Euclidean algorithm for key equation solver block which consumes most area in decoder. For ME block, we proposed pipelined ME calculation cell with 2 delays. Because the ME calculation cell is systolic array architecture, it is easily expanded to higher RS code. Using this merit, we expanded the RS code to data field of MB-OFDM UWB and simulated architecture and synthesized the architecture. The synthesis result shows that the hardware increased to about 3 times.

References

[1] ftp://ftp.802wirelessworld.com/15/04/15-04-0268-03-003a-Multiband-OFDM-CFP-Doc

[2] S. B. Wicker, Error Control Coding : Fundamentals and Applications. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[3] H. M. Hsu, C. L. Wang, "An area-efficien pipelined VLSI architecture for decoding of Reed-Solomon codes based on a time-domain algorithm", IEEE Trans. Ciucuits Syst. Video Technol., vol. 7, pp. 864-871, Dec. 1997

[4] T. K. Matsushima, T. Masuchima, and S. Hirasawa, "Parallel architecture for high-speed Reed-Solomon decoder," IEEE Trans. Comput., vol. C-34, pp. 393-403, May 1985

[5] H. M. Shao, T. K. Truong, L. J. Deutsch, J. H. Yuen, and I. S. Reed, "A VLSI design of a pipelined Reed-Solomon decoder", IEEE Trans. Comput., vol. C-34, pp. 393-403, May 1985

[6] H. M. Shao, I. S. Reed, "On the VLSI Design of Pipelined Reed-Solomon Decoder Using Systolic Arrays", IEEE Trans. Comput., vol 37, pp. 1273-1280, Oct. 1998

[7] H. Lee, "High-speed VLSI architecture for Parallel Reed-Solomon Decoder", IEEE Trans. Very Large Scale (VLSI) Integr. Syst., vol. 11, no.2, pp. 288-294, Apr. 2003