Data Encryption and Decryption using Reed-Muller Techniques

Upputuri Neelima¹, Fazal Noorbasha²

¹,²Department of ECE, KL University
Green Fields, Vaddeswaram, Guntur -522502, A.P, India
¹upputurineelima@gmail.com
² fazalnoorbasha@kluniversity.in

Abstract – Reed-Muller codes play an important role in communication. In communication, security and error free data transmission are two major problems. In this paper, we propose an eight bit original data is encoded using distinct Reed-Muller techniques such as positive polarity Reed-Muller (PPRM), negative polarity Reed-Muller (NPRM), fixed polarity Reed-Muller (FPRM) for secure data communication and also we can compare these techniques in terms of cost. The eight bit encoded data which is obtained from these Reed-Muller techniques are encoded again using hamming code for error free communication. It is found that among all these techniques fixed polarity Reed-Muller is the best technique which gives less cost. We can also observe that secure and error free communication is possible between transmitter and receiver. The data encryption and decryption process has been simulated using Isim simulator and the code is written in Verilog HDL.

Keyword – Reed-Muller, Hamming code, Encoder, Decoder, Verilog HDL.

I. INTRODUCTION

Reed-Muller expansion plays an important role in logic synthesis and circuit design. AND-EXOR circuits are used for representing Reed-Muller expressions. These circuits requires less AND gates than AND-OR circuits [1]. It produces the Boolean functions with highly testable. It creates the unique representations of a Boolean function. It mainly used in arithmetic and telecommunication applications. In communication, secure and error free data transmission from transmitter to receiver are the major issues. There are different techniques for secure and error free data communication. In [2], the eight bit data was encoded using positive polarity Reed-Muller (PPRM), negative polarity Reed-Muller (NPRM) techniques. There are two problems raises due to these techniques. First problem is that when we encode the data using PPRM, NPRM techniques, it gives more number of product terms leading to high power dissipation (high cost). In order to overcome this problem we use fixed polarity Reed-Muller (FPRM) technique. Here FPRM gives less or equal number of product terms than PPRM. Second problem is that when the encoded is transmitted through the channel, there might be a chance for occurrence of error if the channel is noisy. So, there is no possibility to detect and then correct the error. In this paper we propose a method which provides secure and error free data encryption and decryption process. The original eight bit data is encoded using PPRM, NPRM and FPRM techniques for obtaining secure communication. Again the data which is obtained from these techniques are encoded using hamming code for error free communication. In this paper, section II describes the proposed encryption and decryption process. Section III gives the simulation results and then section IV concludes the paper.

II. PROPOSED DATA ENCRYPTION AND DECRYPTION PROCESS

The original eight bit data is encoded using distinct types of Reed-Muller techniques such as positive polarity Reed-Muller (PPRM), negative polarity Reed-Muller (NPRM), fixed polarity Reed-Muller (FPRM). The encoded data which is obtained from these Reed-Muller techniques are encoded again using hamming code for error detection and correction. This encoded hamming data is send through the channel and then receiver receives the data and then detects and corrects the data if the channel is noisy. This noise free data is then decoded using...
hamming code and again decoded by distinct Reed–Muller techniques and then finally got the original eight bit data. Let us consider the 3-variable function.

\[ g(y_1, y_2, y_3) = \sum(0, 1, 3, 4, 5, 6, 7) \]  

(1)

Truth table for the function (1) is shown in the Table I.

**TABLE I. Truth table**

<table>
<thead>
<tr>
<th>y₁, y₂, y₃</th>
<th>g(y₁, y₂, y₃)</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1</td>
</tr>
<tr>
<td>001</td>
<td>1</td>
</tr>
<tr>
<td>010</td>
<td>0</td>
</tr>
<tr>
<td>011</td>
<td>1</td>
</tr>
<tr>
<td>100</td>
<td>1</td>
</tr>
<tr>
<td>101</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>1</td>
</tr>
<tr>
<td>111</td>
<td>1</td>
</tr>
</tbody>
</table>

A. **Data Encryption Process**

Encryption is the process of encoding a message so that its meaning is not obvious.

1) **Data Encoding using Positive Polarity Reed-Muller:** In an PPRM expression of a given function \( g(y_1, y_2, ..., y_n) \), every variable appears in an uncomplimented form. In PPRM expression, we apply positive davio expansion on each variable of a function. The output vector obtained from Table I is taken as original eight bit data. It is encoded by using positive davio (pD) expansion as shown in Fig.2.

If we apply positive davio expansion on the variable \( y_1 \), then \( g_0 \) goes to the left child of the root which is obtained from equation (2), \( g_1 \) is obtained from equation (3),\( g_2 \) goes to the right child of the root which is obtained by performing EX-OR operation between \( g_0 \) and \( g_1 \) as in equation (4).

\[
\begin{align*}
g_0 &= g(y_1, \ldots, y_{j-1}, 0, y_{j+1}, \ldots, y_n) \\
g_1 &= g(y_1, \ldots, y_{j-1}, 1, y_{j+1}, \ldots, y_n) \\
g_2 &= g_0 \oplus g_1
\end{align*}
\]

2) **Data Encoding using Negative Polarity Reed-Muller:** In an NPRM expression of a given function \( g(y_1, y_2, ..., y_n) \), every variable appears in an complimented form. In NPRM expression, we apply negative davio(nD) expansion on each variable of a function. The output vector obtained from Table I is taken as original data and it is encoded by using negative davio expansion as shown in Fig.3.
If we apply negative Davio expansion on the variable $y_1$, then $g_1$ goes to the left child of the root which is obtained from equation (3), and $g_2$ goes to the right child of the root which is obtained by performing ex-or operation between $g_0$ and $g_1$ as in equation (4). Similarly, we perform negative Davio expansion on the variables $y_2$ and $y_3$. The resulting expression is obtained by considering the ones of the leaves of the tree and their corresponding input combination [2]. The resulting NPRM expression for the function (1) is

$$g(y_1, y_2, y_3) = 1 \oplus y_1' \cdot y_3' \oplus y_1' \cdot y_2' \cdot y_3'$$

3) Data Encoding using Fixed Polarity Reed-Muller: In an FPRM expression of a given function $g(y_1, y_2, ..., y_n)$, every variable appears either uncomplemented or complemented form but never exists in both forms. In an FPRM expression, we apply combination of both positive Davio and negative Davio expansions on variables of a function. In FPRM, there are $2^n$ different polarity vectors for $n$-variables. Different polarity vectors give different FPRM expressions with different cost [4],[5]. The output vector of function (1) is taken as original data and then it is encoded for the polarity vector $P = (101)$ as shown in Fig.4.

In the polarity vector $P = (101)$, we apply negative Davio expansion on the variables $y_1$ and $y_3$ and then apply positive Davio expansion on the variable $y_2$. The resulting FPRM expression for equation (1) is

$$g(y_1, y_2, y_3) = 1 \oplus y_1' \cdot y_2'$$

4) Data Encoding using Hamming Code for Distinct Reed-Muller Encoded Data: Hamming code is a linear block code. It can be used to identify single and two bit errors and to correct single bit error. The number of redundancy bits are calculated by using the formula is given as

$$2^r \geq n + r + 1$$

where $n$ is the number of data bits, $r$ is the number of redundancy bits.
Here $n+r$ represents the total number of bits which are to be transmitted. Hence $2^r$ must be greater than or equal to $n+r+1$. The number of redundancy bits $r$ can be calculated by placing the value of $n[6,7]$. If $n$ is 8, then the number of redundancy bits are 4 which is obtained from equation (5). Hence total number of transmitted bits is 12. The Data which is obtained from Fig.2 is taken as Data bits for calculating redundancy bits. The redundant bit $r_1$ is the combination of data bits are $d_0, d_1, d_3, d_4, d_6$ as shown in Fig.5. The redundant bit $r_2$ is the combination of data bits are $d_0, d_2, d_3, d_5, d_6$ as shown in Fig.5. The redundant bit $r_3$ is the combination of data bits are $d_1, d_2, d_3, d_7$ as shown in Fig.5. The redundant bit $r_4$ is the combination of data bits are $d_4, d_5, d_6, d_7$ as shown in Fig.5. Hence the encoded Hamming code word is 101101100011 which is obtained from Fig.5. Similarly, The Data which is obtained from Fig.3, Fig.4 are taken as Data bits for calculating redundancy bits and then their corresponding encoded hamming code word is 101100000101, 111100010001 as shown in Fig.6. These Encoded Hamming code words are send through the channel.

B. Data Decryption Process

1) Error Detection and Correction Process: The receiver receives the data which is transmitted through the channel. If the channel is noisy, then there is a corruption of data occurs. In order to find the location of the error, the redundant bits are recalculated by using the data bits which is used by the sender plus their relevant redundant bit [6,7]. The redundant bit $r_1$ is calculated by using the bits $r_1, d_0, d_1, d_3, d_4, d_6$. The redundant bit $r_2$ is calculated by using the bits $r_2, d_0, d_2, d_3, d_5, d_6$.
The redundant bit $r_3$ is calculated by using the bits $r_3, d_1, d_2, d_3, d_7$. The redundant bit $r_4$ is calculated by using the bits $r_4, d_4, d_5, d_6, d_7$ as shown in Fig.7. Hence the error bit position is calculated by $r_4 r_3 r_2 r_1$. Once the error position is identified, then that bit value in that error position is complimented. The Encoded Hamming code word which is obtained from Fig.5, Fig.6 is send through the channel. Then the receiver receives the data, if there is any error occurs, it can be corrected by complimenting those bit value and then extract the data bits as shown in Fig.7, Fig.8.

2) **Data Decoding using Positive Polarity Reed-Muller:** The extracted data bits 10110011 which are obtained from Fig.7 are taken as input. If we perform positive davenio expansion on $y_1$, then $b_0, b_2, b_4, b_6$ are remain unchanged and $b_0, b_2, b_4, b_6$ are perform bitwise ex-or operation with the corresponding bits $b_1, b_3, b_5, b_7$. Similarly same procedure is applied on the variables $y_2$ and $y_3$ as shown in Fig.9.
3) **Data Decoding using Negative Polarity Reed-Muller**: The extracted data bits 10000101 which are obtained from Fig.8 are taken as input. If we perform negative davio expansion on $y_1$, then $b_0, b_2, b_4, b_6$ are perform bitwise ex-or operation with the corresponding bits $b_1, b_3, b_5, b_7$ and then $b_0, b_2, b_4, b_6$ are remain unchanged. Similarly same procedure is applied on the variables $y_2$ and $y_3$ as shown in Fig.10.

4) **Data Decoding using Fixed Polarity Reed-Muller**: The extracted data bits 10000001 which are obtained from Fig.8 are taken as input and Negative davio expansion is performed on the variables $y_1$ and $y_3$, then $b_0, b_2, b_4, b_6$ are perform bitwise ex-or operation with the corresponding bits $b_1, b_3, b_5, b_7$ and then $b_0, b_2, b_4, b_6$ are remain unchanged. Similarly, positive davio expansion is performed on the variable $y_2$ for polarity vector $P = (101)$ as shown in Fig.11.
III. SIMULATION RESULTS

In this section, we compared the distinct types of Reed-Muller techniques such as positive polarity Reed-Muller, Negative polarity Reed-Muller, Fixed polarity Reed-Muller technique in terms of cost. Different techniques will give different cost. The cost is determined by finding number of ones in the vector. Among those Reed-Muller techniques Fixed polarity Reed-Muller gives less cost as shown in Table II.

TABLE II. Comparison Between Different Reed-Muller Techniques

<table>
<thead>
<tr>
<th>Data</th>
<th>NPRM cost</th>
<th>PPRM cost</th>
<th>FPRM Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>00001101</td>
<td>6</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>00010011</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>00010101</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>00011011</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

A. Data Encoding using Positive Polarity Reed-Muller

The original data which is to be transmitted is 11011111. The encoded hamming code word is 10110100011 which is obtained from Fig.5 for the encoded positive polarity reed-muller data is 10110011 as shown in Fig.2. These 12 bit encoded hamming code word is send through the channel as shown in Fig.12.

B. Data Decoding using Positive Polarity Reed-Muller

The encoded 12 bit hamming code word is sent through the channel as shown in Fig.12 but the receiver receives the data as 101101100011. Hence, the receiver detect and then correct the data if there is an error occurs and then extract the data. Again this extracted data is further decoded using positive polarity reed-muller technique as shown in Fig.9. Finally, we got the original data as shown in Fig.13.
C. Data Encoding using Negative Polarity Reed-Muller

The original data which is to be transmitted is 11011111. If we perform negative polarity reed-Muller technique, then the resulting encoded data is 1000101 as in Fig.3. The encoded hamming code word is 101100000101 which is obtained from Fig.6 which is send through the channel as shown in Fig.14.

![Fig.14.Data encoding using NPRM](image)

D. Data Decoding using Negative Polarity Reed-Muller

The receiver receives the data which is sent through the channel as shown in Fig.14. If the channel is noisy, then there is a chance to error occurs at the reception. The receiver can detect and correct the data and then extract the data bits from the hamming code word as shown in Fig.8. Again this extracted data is further decoded using negative polarity reed-muller technique as shown in Fig.10. Finally, we got the original data as shown in Fig.15.

![Fig.15.Data decoding using NPRM](image)

E. Data Encoding using Fixed Polarity Reed-Muller

The original data which is to be transmitted is 11011111. The encoded hamming code word is 111100010001 which is obtained from Fig.6 for the encoded fixed polarity reed-muller data is 10000001 as shown in Fig.4. These 12 bit encoded hamming code word is send through the channel as shown in Fig.16.

![Fig.16.Data encoding using FPRM](image)

F. Data Decoding using Fixed Polarity Reed-Muller

The receiver receives the data which is sent through the channel as shown in Fig.16. If the channel is noisy, then there is a chance to error occurs at the reception. The receiver can detect and correct the data and then extract the data bits from the hamming code word as shown in Fig.8. Again this extracted data is further decoded using fixed polarity reed-muller technique as shown in Fig.11. Finally, we got the original data as shown in Fig.17.

![Fig.17.Data decoding using FPRM](image)

IV. CONCLUSION

Secure and error free data transmission from transmitter to receiver is two important factors in the field of communication. In this paper, we proposed a technique which provides both secure and error free data communication. Secure data transmission obtained by encoding original data using distinct types of Reed-Muller techniques such as PPRM, NPRM, FPRM and also we compared these techniques in terms of cost. Among those Fixed polarity Reed-Muller technique gave less cost. Error free data communication is obtained by again encoding the encoded data which is obtained from distinct Reed-Muller techniques using hamming code. Single error is corrected but multiple errors cannot be corrected using hamming code. To overcome this we can use LDPC, Reed-solmon codes.
REFERENCES


AUTHOR PROFILE

Upputuri Neelima was born on 25th March 1992. She received her B.Tech Degree in Electronics and Communication From Chalapathi Institute and Technology, Mothadaka, Guntur, Andhra Pradesh,INDIA in 2013.Presently she is Pursuing M.Tech in VLSI Specialization at KL University, Guntur, Andhra Pradesh, India. Her main interest in Low power VLSI, Testing of VLSI circuits.

Dr. Fazal Noorbasha was born on 29th April 1982. He received his M.Tech Degree in VLSI Technology From the North Maharashtra University, Jalgaon, Maharashtra, INDIA in 2008,and Ph.D Degree in VLSI from Department of Physics and Electronics, Dr. Harisingh Gour Central University, Sagar, Madhya Pradesh, INDIA in 2011. Presently he is working as a Associate Professor, Department of Electronics and Communication Engineering, KL University, Guntur, Andhra Pradesh, India, where he has been engaged in teaching, research and development of Low-power, High-speed CMOS VLSI Soc, Memory Processors LSI's, Fault Testing in VLSI, Embedded Systems and Nanotechnology.