

# Fully Reused VLSI Architecture of FM0/Manchester Encoding Technique for Memory Application

Triveni A Patil<sup>1</sup>, Sadhana Choudhary<sup>2</sup>

<sup>1</sup>Visvesvaraya Technological University, Lingrajappa Engineering College, Bidar-585401, India

<sup>2</sup>Lingrajappa Engineering College, Visvesvaraya Technological University, Bidar-58401, India

**Abstract:** In this paper a fully reused VLSI architecture of FM0/Manchester encoding technique for memory application has been proposed. In this paper we are encoding the 1 bit data into 16 bit data and storing it into a memory of certain address location given by the linear feedback shift register (LFSR), whose input is taken from the pseudo random sequence generator (PRSG). The encoded 16 bit data is stored into memory controller; the encoded data is decoded back into 1 bit data under the condition: when MSB bit is at logic state 1. By using FM0/Manchester encoding and decoding technique, the data will be secure, this process is easy and faster to carry out. This paper develops a fully reused VLSI architecture, and also exhibits an efficient performance.

**Keywords:** FM0/ Manchester encoder, Linear feedback shift register (LFSR), Pseudo random sequence generator (PRSG), Memory controller.

## 1. Introduction

The FM0/Manchester encoding has many applications, such as Dedicated short Range communication, Digital signal processing, Memory applications etc. Firstly in DSRC application, the FM0/Manchester encoding using similarity oriented logic simplification (SOLS) technique was used. DSRC is a protocol for one or two way medium range communication. DSRC is categorized into two types

1. Automobile to Automobile.

2. Automobile to Roadside.

- In automobile to automobile, DSRC provides the means of sending the message and transmitting among the automobiles for safety issues public information announcement.
- In automobile to roadside, DSRC emphasises on the intelligent transportation service, such as electronic toll collection.

Here, in this paper we are using FM0/Manchester encoding technique for memory applications. Here the FM0 encoder, encodes the 1 bit data into 16 bit and stores in certain memory location allocated by LFSR. The encoded data is again decoded back into 1 bit, by performing XOR operation of LFSR address bits and memory controller input bits. When we get the MSB bit as logic state 1, then the encoded data of 16 bit is decoded back into 1 bit.

## 2. Literature Survey

1. The literature [1] proposes the fully reused VLSI architecture of FM0/Manchester encoding using similarity oriented logic simplification (SOLS) technique for Dedicated short range communication. The SOLS technique improves the hardware utilization rate from 57.14% to 100% for both FM0 and Manchester encodings.

2. The literature [2] proposes a VLSI architecture of Manchester encoder for optical communications. This design uses the CMOS inverter and the gated inverter as the switch to construct Manchester encoder. It is executed by 0.35- $\mu$ m CMOS technology and its operation frequency is 1 GHz.

3. The literature [3] later replaces the architecture of switch in [2] by the nMOS device. It is performed in 90-nm CMOS technology, and the maximum operation frequency is as high as 5 GHz.

4. The literature [4] evolves a high-speed VLSI architecture relatively fully reused with Manchester and Miller encodings for radio frequency identification (RFID) applications. This architecture is performed in 0.35- $\mu$ m CMOS technology and the maximum operation frequency is 200 MHz.

5. The literature [5] also proposes a Manchester encoding architecture for ultrahigh frequency (UHF) RFID tag emulator. This hardware architecture is operated by the finite state machine (FSM) of Manchester code, and is performed by field-programmable gate array (FPGA) prototyping system. The maximum operation frequency of this architecture is about 256 MHz.

6. The literature [6] proposes the identical design methodology which is applied individually to construct FM0 and Miller encoders also for UHF RFID Tag emulator.

## 3. Coding Criteria of FM0 and Manchester Code

In the following analysis, the clock signal and the input data are termed as CLK, and X, respectively. With the above variables, the coding principles of FM0 and Manchester codes are discussed as follows.

### 3.1 FM0 Encoding

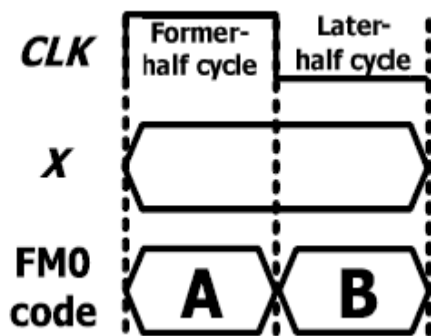


Figure 1: Code word structure of FM0

As shown in Fig. 1, for all the values of X, the FM0 code includes two parts: one for first-half cycle of CLK, A, and the other one for second-half cycle of CLK, B. The below mentioned three rules are used to describe FM0 coding criteria

- 1)The FM0 code have to perform the transition between A and B, When X is at logic-0.
- 2)There should not be any transition performed between A and B, When X is at logic-1.
- 3)Irrespective of X , The transition is performed with in each FM0 code.

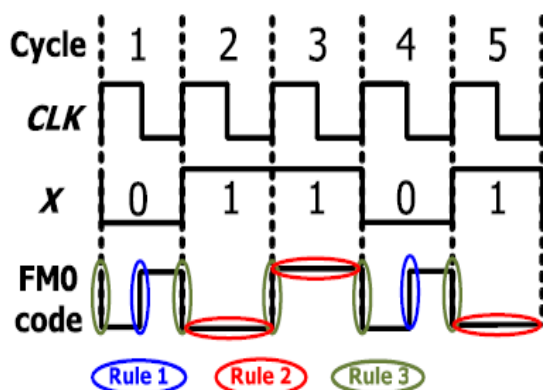


Figure 2: Illustration of FM0 coding example.

An example of FM0 coding is illustrated in Fig. 2. At cycle 1, the X is logic-0; hence, a transition occurs on its FM0 code, as per the rule 1. At the beginning for ease, the transition is set from logic-0 to -1. As per rule 3, a transition is performed with in each FM0 code, so that the logic-1 is switched to logic-0 in the beginning of cycle 2. Then, as per the rule 2, this logic-level is held without any transition in complete cycle 2 for the X of logic-1. Thus, the FM0 code of each cycle can be deduced with these three rules specified earlier.

### 3.2 Manchester Encoding

The example of a Manchester coding is illustrated in Fig. 3. The Manchester code is deduced from

$$X \oplus CLK \dots \dots \dots (1)$$

The Manchester encoding is accomplished by performing a XOR operation of CLK and X. The clock always possesses a

transition within one cycle, and so does the Manchester code irrespective to what the X is.

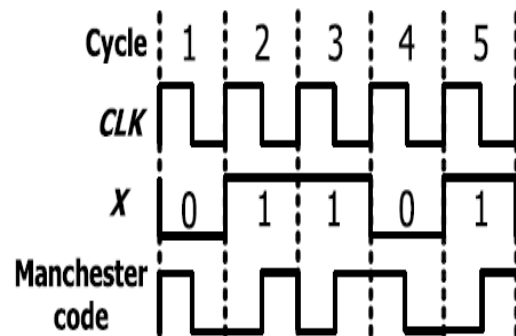


Figure 3: Illustration of Manchester coding example.

## 4. Proposed Design

These articles are implemented using HDL designer for synthesis and Model Sim for simulation. To give an intent analysis, the proposed VLSI architecture is realized with HDL design-flows.

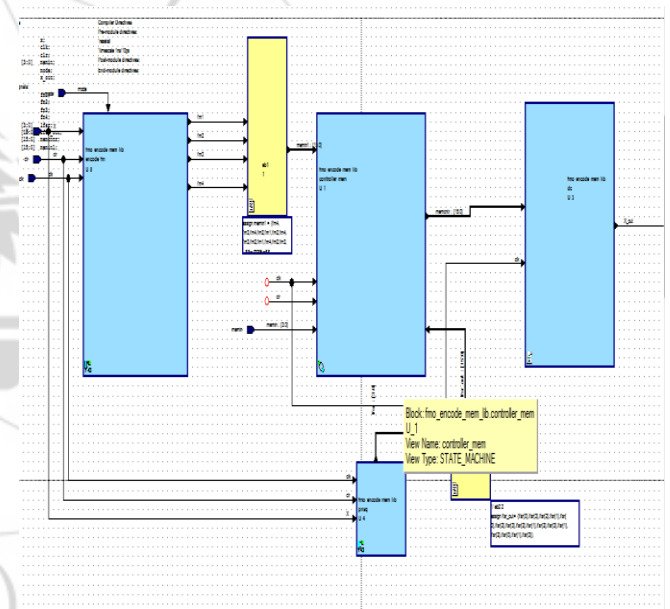


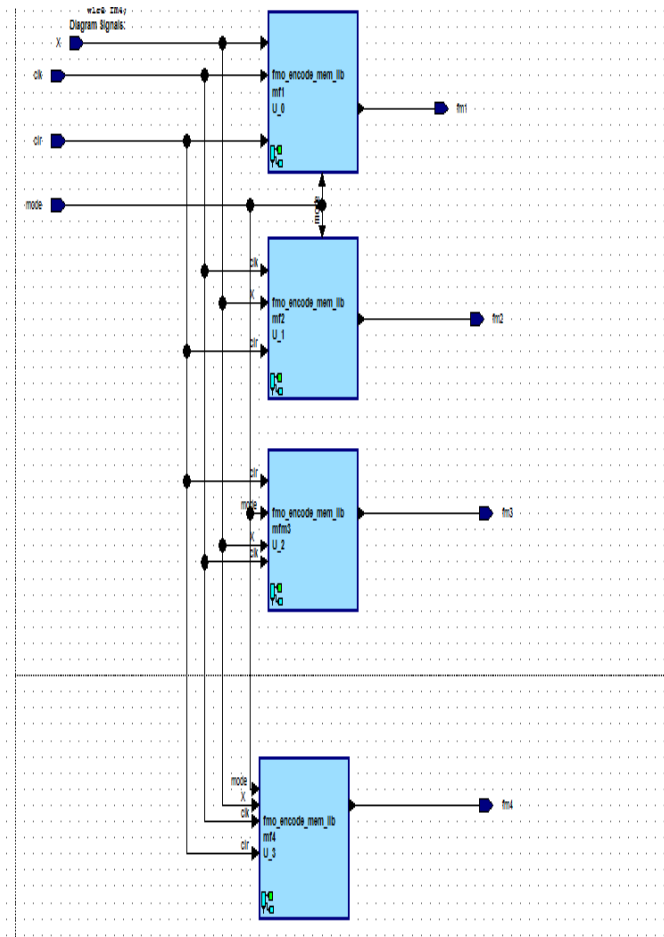
Figure 4: Block diagram

The above block diagram consists of 5 main blocks.

1. Encoding block.
2. Pseudo random sequence generator (PRSG).
3. Linear feedback shift register (LFSR) .
4. Memory controller block.
5. Decoding block.

### 4.1 Functions of these Blocks

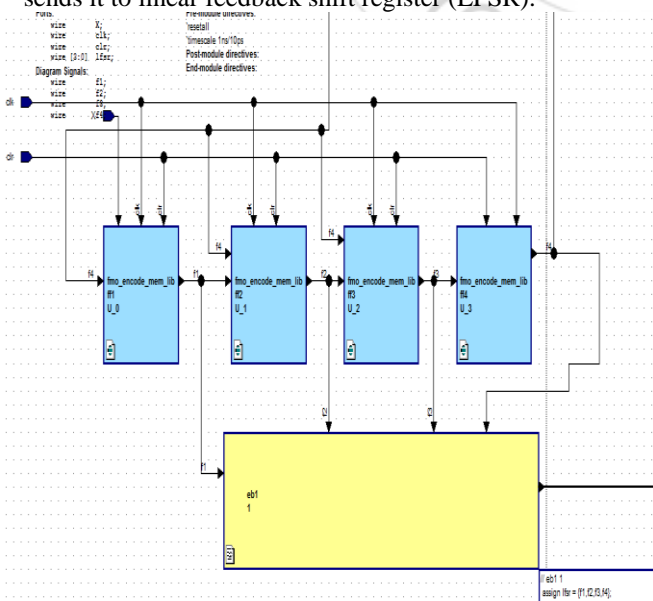
1. Encoding block : Here in this block it consists of 4 modules of FM0 encoder, X bit is the input given to all the 4 modules. Inputs to this block are X bit, clk, clr, and mode.



**Figure 5: FM0 encoder modules**

Here the 1 bit input data is encoded into the 16 bit data and sent to the memory controller block

2. Pseudo Random Sequence Generator: It is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random number. Here it generates sum random 4 bit numbers and sends it to linear feedback shift register (LFSR).



**Figure 6:** Block diagram of PRSG

3. **Linear Feedback Shift Register (LFSR)** : LFSR gives the address location where the 16 bit encoded data has to be stored in memory controller. The PRSG sends the 4 bit input i.e. nothing but the address location to the LFSR, where encoded data has to be stored.

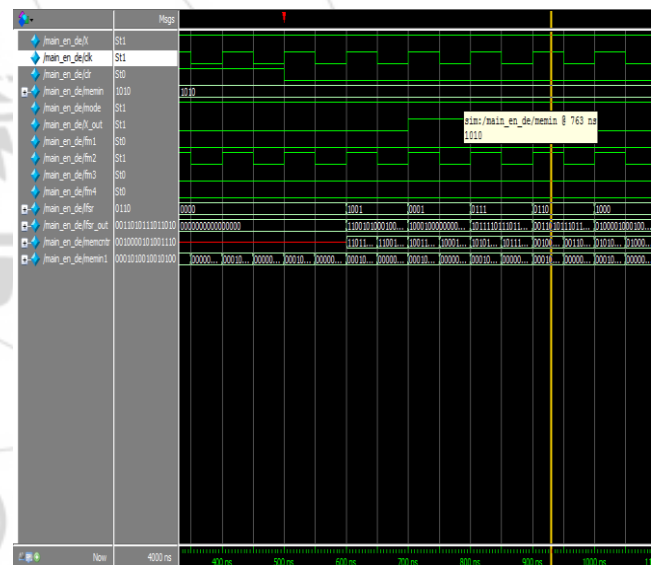
4. Memory Controller : In memory controller , firstly the 4 bit encoded data from 4 modules of FM0 encoder are concatenated into 16 bit encoded data. This 16 bit encoded data is stored into a particular address location sent by LFSR.

5. Decoding block: Now the 16 bit encoded data stored in memory is XOR with the address location of LFSR. If the MSB bit of the XOR output is at logic 1, then in only that condition the data will be decoded back into 1 bit, Because that is the condition we have designed for.

## 5. Experimental Results

Here the block diagram shown in the fig 4 is simulated using Modelsim and by giving the following data as input  
X= 1, Clock = clock, Clear=1, Mode = 1, Memory input= 1010

We obtain the following results shown in below figures



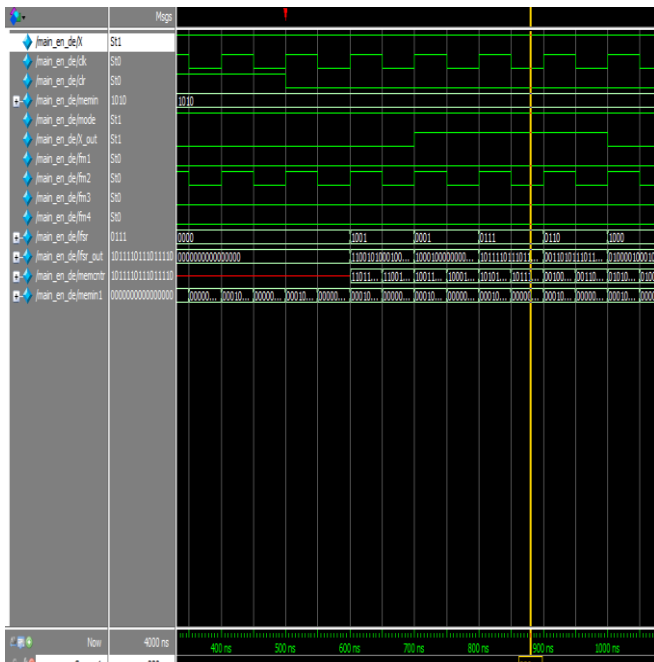
### Figure 7: Experimental Results 1

Here from the above figure 7, by analysing the data

LFSR out = 0011010111011010 and

Memory input= 0001010010010100.

Now by performing the XOR operation we get the output of Memory controller = 0010000101001110. As the MSB bit of this 16 bit data is 0, the encoded data is not decoded.



**Figure 8: Experimental Results 2**

Here from the above fig 8, by analysing the data

LFSR out=101110111011110 and

Memory input= 0000000000000000.

Now by performing the XOR operation we get the output of memory controller= 101110111011110. As here the MSB bit of this 16 bit encoded data is “1” the encoded data is decoded back into 1 bit.

## 6. Conclusion

The fully reused VLSI Architecture of FM0/Manchester encoder for memory applications is effective and powerful in securing the data, as compared to the other encoding techniques. FM0/Manchester encoder are easy to perform operations and faster. Hence these encoding techniques are efficient.

## References

- [1] Fully Reused VLSI Architecture of FM0/Manchester Encoding Using SOLS Technique for DSRC Applications Yu-Hsuan Lee, *Member, IEEE*, and Cheng-Wei Pan
- [2] P. Benabes, A. Gauthier, and J. Oksman, “A Manchester code generator running at 1 GHz,” in *Proc. IEEE, Int. Conf. Electron., Circuits Syst.*, vol. 3. Dec. 2003, pp. 1156–1159.
- [3] A. Karagounis, A. Polyzos, B. Kotsos, and N. Assimakis, “A 90nm Manchester code generator with CMOS switches running at 2.4 GHz and 5 GHz,” in *Proc. 16th Int. Conf. Syst., Signals Image Process.*, Jun. 2009, pp. 1–4.
- [4] Y.-C. Hung, M.-M. Kuo, C.-K. Tung, and S.-H. Shieh, “High-speed CMOS chip design for Manchester and Miller encoder,” in *Proc. Intell. Inf. Hiding Multimedia Signal Process.*, Sep. 2009, pp. 538–541.
- [5] M. A. Khan, M. Sharma, and P. R. Brahmanandha, “FSM based Manchester encoder for UHF RFID tag emulator,”

in *Proc. Int. Conf. Comput., Commun. Netw.*, Dec. 2008, pp. 1–6.

- [6] M. A. Khan, M. Sharma, and P. R. Brahmanandha, “FSM based FM0 and Miller encoder for UHF RFID tag emulator,” in *Proc. IEEE Adv. Comput. Conf.*, Mar. 2009, pp. 1317–1322.

## Author Profile



**Triveni A Patil** was born in May 1991, completed her Bachelor of Engineering Degree in Electrical & Electronics, currently pursuing M. Tech VLSI Design & Embedded System in Lingaraj Appa Engineering College.