VLSI DESIGN OF EFFICIENT ENCODING TECHNIQUE USING ROLS METHOD FOR WAVE APPLICATIONS

Banarjun S M[1], Pradeep M N[2]

[1]*M.Tech VLSI Design, Department of ECE, Dayananda Sagar College of Engineering, Bangalore, INDIA*
[2]*Assistant Professor, Department of ECE, DSCE, Bangalore, INDIA*
[1]***Email: arjun911066@gmail.com***

Abstract

The automotive industry is emerging aggressively towards the direction of advanced active safety. The automotive industry is aiming to work for the development of Dedicated Short Range Communication (DSRC) technology, for the purpose in vehicle-to-vehicle and vehicle-to-roadside communication. To transmit the vehicular information and broadcast vehicle position, DSRC communication technology is adopt as bridge. This work is devoted to develop the encoding techniques for WAVE/DSRC applications. The DSRC standards generally make use of FM0 and Manchester encoding to achieve dc balance and enhancing signal reliability. Nevertheless, the diversity of coding between FM0 and Manchester codes limits the ability to design a fully utilize hardware architecture for both. The Reuse oriented logic simplification (ROLS) technique is proposed here so that the Hardware utilization rate (HUR) will reach to 100% from 57% for both encoding techniques. The proposed architecture will have less delay, area as compared to existing architecture. In this paper the architecture analyzed to reduce the number of components. Using both encodings the area, delay, and power is reduced in DSRC. The power consumption is 22132.181nW for both encoding, and area is 193µm^2. The encoding technique in this work supports the DSRC standards used by several organizations of America, Japan and Europe.

Keywords: *DSRC (Dedicated Short Range Communication), FM0, Manchester, VLSI*

----------***----------

I. INTRODUCTION

Dedicated Short Range Communication (DSRC) [1] is an emerging standard protocol for one or two way communication especially for intelligent transportation systems. DSRC is an efficient technology for communication based vehicular safety applications. In DSRC, the automobile communication can briefly categorize into automobile to automobile and automobile to roadside. In automobile to automobile, the DSRC allows message sending and broadcasting among vehicles for safety purposes and public information [2] and [3]. The safety messages conclude blind spot, crossway warning, in vehicle warning and collision alert. The automobile to roadside concentrates on high level transportation system, such as ETC (electronic toll collection) system. The application of ETC can be furthermore extended to collection of parking service, gas refueling station, ambulance service. The DSRC transceiver system architecture is as shown in Fig. A. The upper and bottom sections are allocated for transmission and reception, respectively. The transceiver is categorized into three modules: microprocessor, baseband processing and RF front end. The baseband processing does the operation of modulation, error correction, clock synchronization and encoding. The RF front end establishes communication through the antenna.

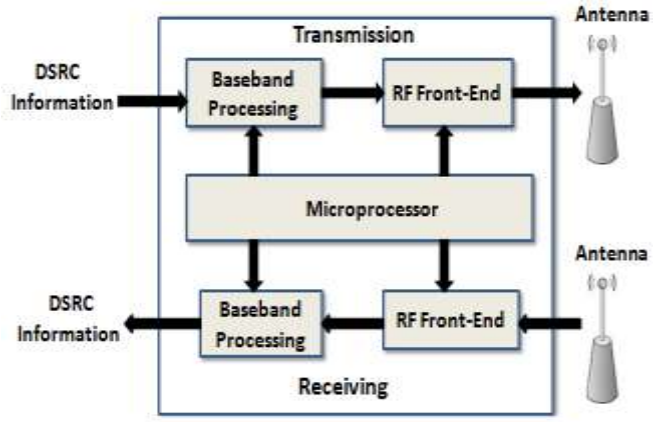


Fig. A. DSRC transeciver architecture

Many organizations in different countries started to develop the DSRC standards for automobile applications. The Table I shows the outline of DSRC standards for different countries. The DSRC standards among the above table, the data rate individually aims at 27Mbps, 4Mbps and 500Kbps and the carrier frequency is around 5.8GHz and 5.9GHz. The modulation methods used are Amplitude shift keying (ASK), phase shift keying (PSK) and orthogonal frequency division multiplexing (OFDM). The Encoding for downlink incorporates FM0 and Manchester. Usually, the wave signals of the transmitted signal is expected to have zero-mean for robustness issue and this is also called as dc-balance. The transmitted message signal contains stream of binary sequence, which is difficult to accomplish dc-

balance. The reason being of FM0 and Manchester codes are to provide the transmitted signal with dc-balance. Both FM0 and Manchester contributes dc-balance that plays a vital role in signal reliability. The dc-balance can significantly enhance the signal. The main goal of dc-balance is to maintain the number of logic 1 equal to that logic 0 for a stream of binary sequence.

A. Features of this Paper

The coding diversity between FM0 and Manchester codes limits the ability to design VLSI architecture that fully reused with each other. This paper proposes a VLSI architecture design using Reuse oriented logic simplification (ROLS) method.

Table I. Outline Of Dsrc Standards For America, Europe And Japan

	Europe	America	Japan
Organization	CEN [a]	ASTM [b]	ARIB [c]
Data rate	500Kbps	27Mbps	4Mbps
Carrier Frequency	5.8GHz	5.9GHz	5.8GHz
Modulation	ASK, PSK	OFDM	ASK
Encoding (Downlink)	FM0	Manchester	Manchester

[a] European Committee for Standardization
[b] American Society for Testing and Materials
[c] Association of Radio Industries and Businesses.

The ROLS consists of two origin methods: area efficient retiming and balance logic simplification. The area efficient retiming rearranges the hardware architecture which results in reduction of area. The balance logic simplification adequately combines FM0 and Manchester encodings with fully reutilize hardware architecture. With ROLS method, this paper proves the hardware architecture of FM0 and Manchester encodings for DSRC/WAVE applications is fully efficient and reusable. The experimental results show this design is fully efficient in terms of area and delay compared with existing works and implemented in Spartan 3E FPGA board.

B. Organization

The remaining sections are organized as follows. Section II explains the coding rules of FM0 and Manchester codes. Section III explains the limits of the existing hardware utilization of FM0 and Manchester encoders. It shows the difficulty to design and reutilize the fully efficient VLSI hardware architecture for both encoders. The proposed VLSI architecture using ROLS method eliminates the limitation of the existing work is reported in Section IV. Two base methods of ROLS method, area efficient retiming and balance logic simplification are explained in this section. The experimental results and discussions are described in Section V. This section explains simulation results and hardware implementation results of enhanced Manchester and FM0 encoders. Finally conclusion is discussed in section VI.

II. CODING FUNDAMENTALS OF FM0 AND MANCHESTER CODE

In this section, the input and the clock signal are expressed as X and CLK respectively. The coding fundamentals of FM0 and Manchester codes are explained as follows.

A. FM0 Encoding

The Fig. B shows the coding rule for a clock cycle of FM0 code. The FM0 contains two parts: one for positive half cycle of CLK, A, and the other one for negative half cycle of CLK, B.

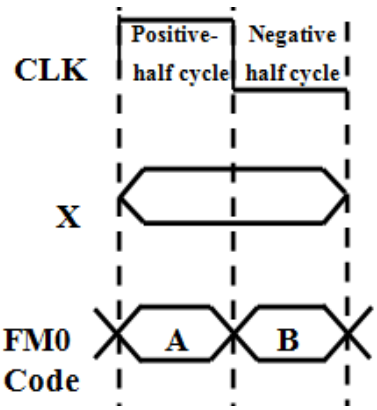


Fig. B. FM0 Codeword structure

The coding rule of FM0 is illustrated as the following three principles.

i. If the input data is logic 0, the signal must have a transition on the logic level within a clock cycle.
ii. If the input data is logic 1, no transition is allocated on logic level within a clock cycle.
iii. There is a transition among each FM0 code, irrespective what the input data is.

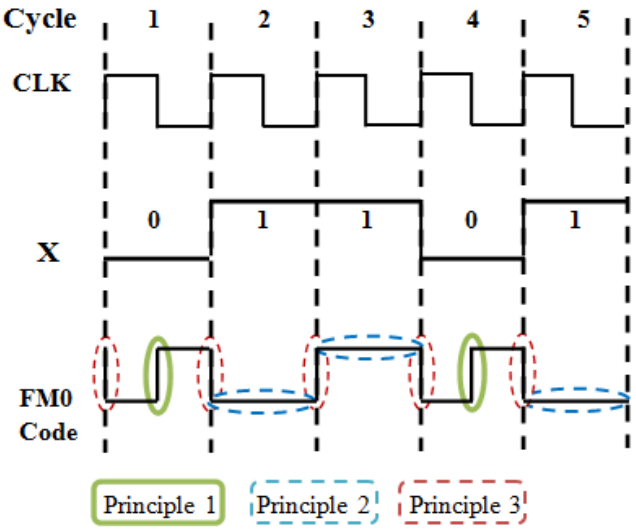


Fig. C. Illustration of FM0 coding for specified input

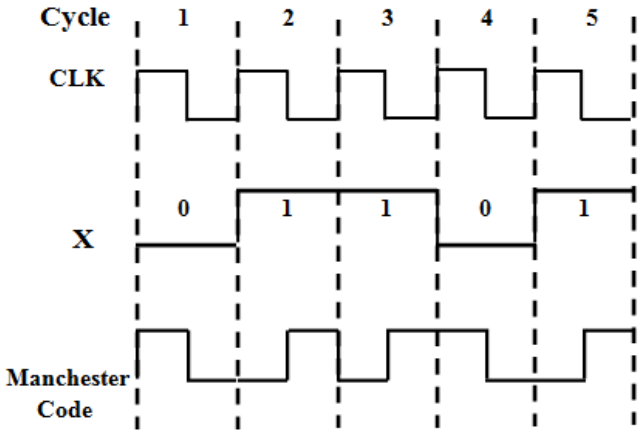


Fig. D. Illustration of Manchester coding for specified input

The illustration for FM0 coding is as shown in Fig. C. According to principle 1, if input data is logic 0, therefore, a transition exists on its FM0 code. To understand, this transition initially starts from logic 0 to logic 1. A transition is allocated among each FM0 code, according to principle 3, and therefore the logic 1 is changed to logic 0 in the beginning of cycle 2. According to principle 2, FM0 exhibits no transition on logic level within a clock cycle.

B. Manchester Encoding

The Manchester coding illustration is as shown in Fig. D. The Manchester code can be realized from

X CLK. $\oplus$ (1)

The Manchester encoding is derived with an XOR function for CLK and input X. There is a transition for both input data logic 0 and logic 1. The clock always exhibits a transition within a clock cycle, and so do the Manchester code irrespective what the input X is.

III. CIRCUMSCRIPTION REPORTS ON HARDWARE UTILIZATION OF MANCHESTER AND FM0 ENCODER

To analyze and make a report on hardware utilization of FM0 and Manchester encoders, the hardware architectures for both are performed. As explained in earlier section, the hardware architecture of Manchester encoding is as simple as XOR function. However, the realization for hardware architecture of FM0 is not as simple as that of Manchester. To construct the hardware architecture of FM0 encoding we need to start with the FSM of FM0. As shown in Fig. E (i), the FSM of FM0 code is of four states. A state code is individually allocated to each state, and each state code consists of A and B, as shown in Fig. B. As per coding fundamental of FM0, the FSM of FM0 is as shown in Fig. E (ii). If the initial state is S_1, and its state code is 11 for A and B, respectively. If the input X is logic 0, the state transition will follow both principles 1 and 3. The only next state that can meet the both principles for the input X of logic 0 is S_3. If the input X is logic 1, the state transition will follow both principles 2 and 3. The only next state that can meet both principles for the input X of logic 1 is S_4. Therefore, the state transition of every state can be constructed completely.

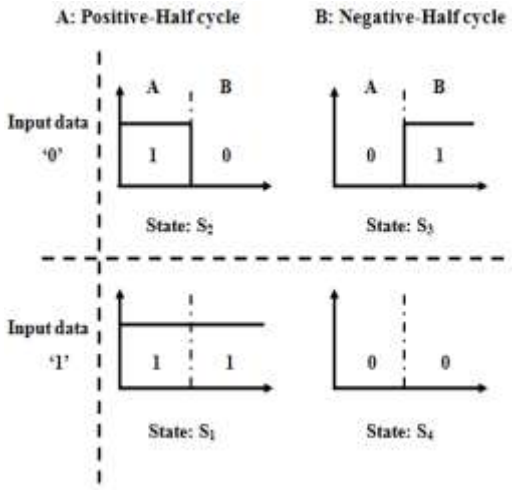
The FSM of FM0 can also lead to the transition table of each state, as shown in Table II. A (t) and B (t) denotes the discrete time state code of current state at time t. Their previous states are represented as the A (t-1) and B (t-1), respectively. By the transition table, the logical functions of A (t) and B (t) are represented as,

$$A(t) = \overline{B(t-1)} \quad (2)$$

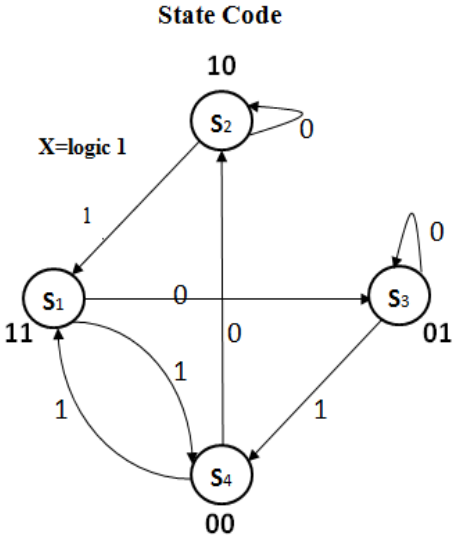
$$B(t) = X \oplus B(t-1). \quad (3)$$

For both A (t) and B (t), the logical functions of FM0 code is represented as

CLK A $\overline{(t)}$ + CLK B (t). (4)



(i)



(ii)

Fig. E. Illustration of FSM of FM0.
(i) States definition
(ii) State diagram of FM0

Table II. State Transition Table Of Fm0

Previous State		Current State			
A(t-1)	B(t-1)	A(t)		B(t)	
		X=0	X=1	X=0	X=1
1	1	0	0	1	0
1	0	1	1	0	1
0	1	0	0	1	0
0	0	1	1	0	1

From (1) and (4), the VLSI hardware architectures of FM0 and Manchester encoders is as shown in Fig. F. This VLSI hardware architecture consists of 7 logic components. The upper part is the hardware architecture of FM0 encoder, which contains 5 logic components. The lower part is the Manchester encoder hardware architecture which contains 1 logic component. From (1), the Manchester encoder is as simple as XOR function for CLK and input X. However, the FM0 encoding depends not only on the input X but also depends on the previous state of FM0 code. The D flip-flops store the state code of FM0 code. The Multiplexer_1 is to switch A (t) and B (t) through the select signal Clock. The Multiplexer_2 decides among both the encoders based upon the select signal Mode. Whenever Mode=0, FM0 code is selected and Mode=1, for Manchester code. To calculate the hardware utilized, HUR (hardware utilization rate) is defined and is given by,

$$\text{HUR} = \frac{Active\ Components}{Total\ Components} \text{ x } 100\% \quad (5)$$

The component is termed as the hardware to operate as a specific logic function, such as AND, NOT, OR and flip-flop. The active components represent the components that work for FM0 or Manchester encoding. The total components are the number of components in the complete hardware architecture irrespective what the encoding being selected.

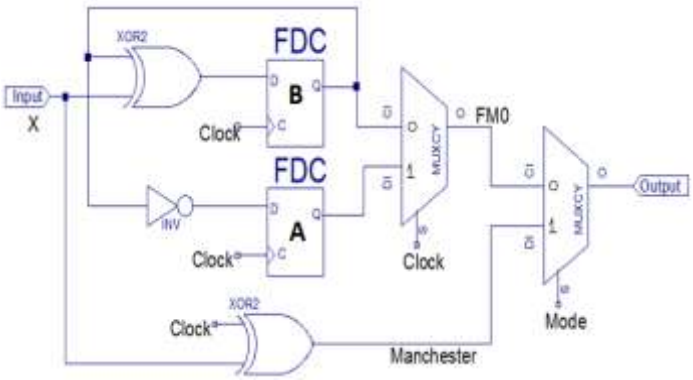


Fig. F. Hardware architecture of FM0 and Manchester encoding

Table III. Hur of FM0 and Manchester Encodings

Coding	Active Components (transistor count) / Total Components (transistor count)	HUR
FM0	6 (86) / 7 (98)	85.7%
Manchester	2 (26) / 7 (98)	28.6%
Average	4 (56) / 7 (98)	57.1%

The Table III shows the HUR of FM0 and Manchester encodings. The total components are 7 for both encoding techniques, including Multiplexer_2 to decide which coding method is selected. The active components are 6 for FM0 encoding; hence the HUR is 85.7%. In case of Manchester encoding, the active components are 2, comprising XOR2 and Multiplexer_2, and hence its HUR is 28.5%. Upon calculating the average, this hardware architecture has poor HUR of 57.1%, and half of total components are wasted.

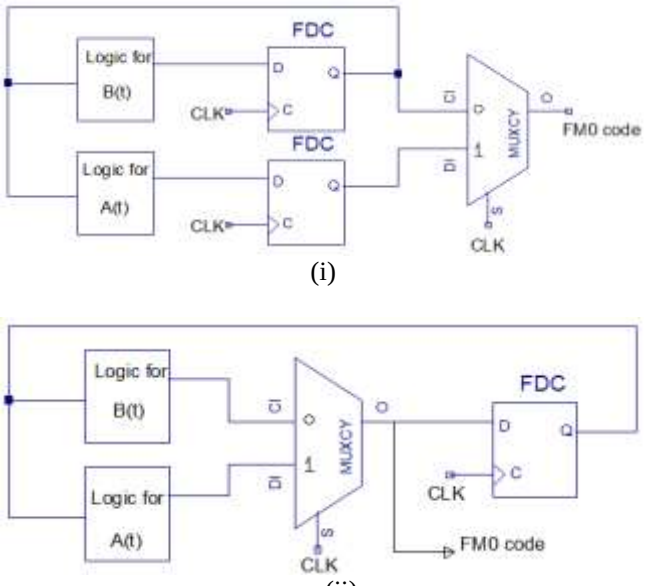


Fig. G. Illustration of area efficient retiming on FM0 encoding architecture. (i) FM0 encoding without area efficient retiming. (ii) FM0 encoding with area efficient retiming.

IV. DESIGN OF VLSI ARCHITECTURE FOR FM0 ENCODER AND MANCHESTER ENCODER USING ROLS METHOD

The goal of ROLS method is to design a fully efficient reusable VLSI architecture for both encoders. The ROLS method is categorized into two core parts: area efficient retiming and balance logic simplification.

A. Area Efficient retiming

The FM0 logic in Fig. F is shown in Fig. G (i). The logic for A (t) and B (t) are the functions to express A (t) and B (t), where the input X is neglected for a concise representation. In FM0, the state code of each state is stored into D flip-

flops. From (2) and (3), the transition of state code only depends on B (t-1) instead of A (t-1) and B (t-1). Hence the FM0 encoding requires a single 1 bit flip-flop to store B (t-1). Since both A (t) and B (t) depends on B (t-1), the flip-flop B can be eliminated. If it is directly removed there will be no synchronization between A (t) and B (t). To avoid this logic error, the flip-flop B is replaced after the multiplexer as shown in Fig. G (ii). This results an architecture which is as shown in Fig. G (ii). The resulting architecture is area compact hence the name area efficient retiming. On each cycle, the FM0 code, comprising A and B, is realized from the logic of A (t) and logic of B (t), respectively. The FM0 code is switched alternatively between A (t) and B (t) through multiplexer by select signal CLK.

B. Balance logic Simplification

As earlier explained, the Manchester encoding can be realized from⊕ CLK, and is expressed as

$$X\,\overline{CLK} + \overline{X}\,CLK \qquad (6)$$

This can be shown in Fig. H (i). It is similar to logical function of FM0 encoding in (4). From (4) and (6), the FM0 and Manchester logics have multiplexer as a common point with the select signal as CLK. In Fig. H (ii), the idea of balance

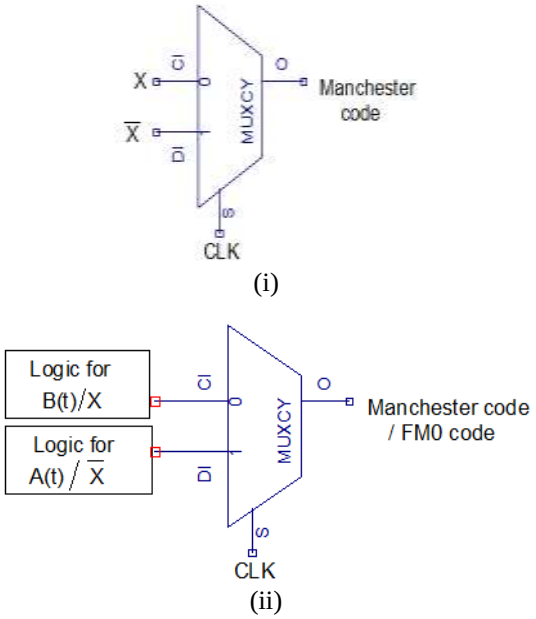


Fig. H. Concept of balance logic simplification for FM0 and Manchester encodings. (i) Manchester encoding using multiplexer. (b) Combines the logic functions of FM0 and Manchester encodings.

logic simplification is to integrate the $\overline{X}$ into A (t) and X into B (t), respectively. The Fig. I show the logic for A (t)/X. The inverter of B (t-1) is used to derive A (t), and $\overline{X}$ is obtained by an inverter of X. The logic for A (t)/$\overline{X}$ can use the same inverter, and then mux is connected before the inverter to switch the operands of B (t-1) and X. The select signal mode decides either FM0 or Manchester encoding is selected. The similar way is adopted for the logic B (t)/X., as shown in Fig. J.

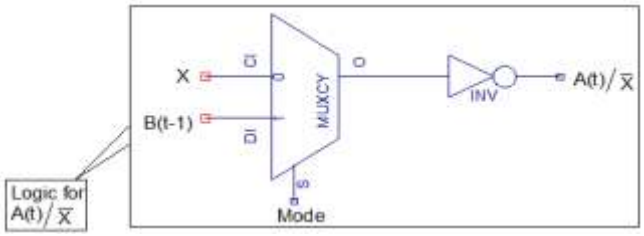


Fig. I. Balance Logic Simplification For A (T) And X

Ho⊕ever, this results in drawback as the architecture having XOR is only devoted for FM0 encoding and it is not shared with Manchester encoding. Hence the HUR for this architecture is limited. The X can also be interpreted as the X 0, and hence making the XOR operation shared with FM0 and Manchester encodings. The resulting fig. of logic for B (t) is as shown in Fig. J (ii), where the mux responsible to switch the operands of B (t-1) and logic 0. This architecture shares XOR for X and B (t), and results in increased HUR. The mux in Fig. J (ii) can furthermore replaced by flip-flop B from area efficient retiming technique, as shown in Fig. J (iii). The resulting VLSI architecture of FM0/Manchester encoding using ROLS method is as shown in Fig. K (i). From the resulting architecture we analyze that the timing computation of mux with select signal as CLK is almost identical to that of XOR.

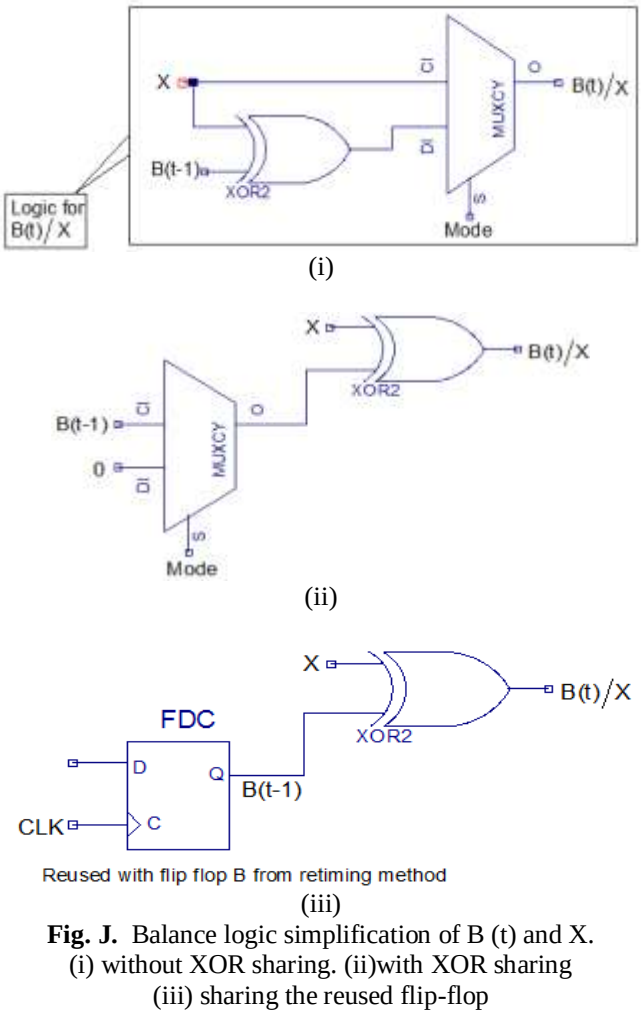


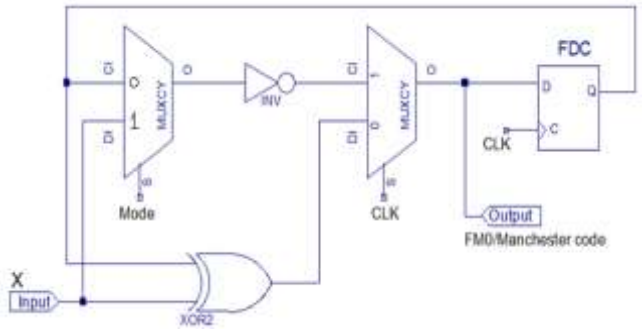
Fig. J. Balance logic simplification of B (t) and X. (i) without XOR sharing. (ii)with XOR sharing (iii) sharing the reused flip-flop

This incorporates an inverter in series with that mux. This unbalance computation results in the glitch to that mux, resulting in logical fault on coding. To avoid this unbalance computation, the XOR is replaced by XNOR and the inverter is placed at the output of mux with CLK as select signal. Hence the logic computation is more balance each other.

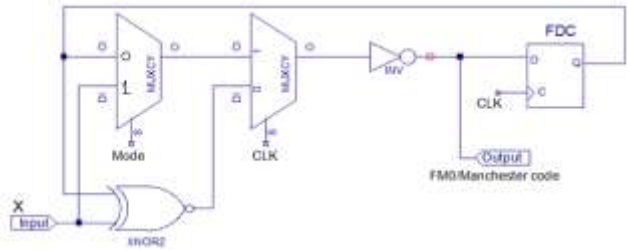
Table IV. Performance Analysis of Rols Method

Coding	Active Components (transistor count) / Total Components (transistor count)	HUR
FM0	5 (44) / 5 (44)	100%
Manchester	5 (44) / 5 (44)	100%
Average	5 (44) / 5 (44)	100%

The proposed and enhanced VLSI architecture containing only four logic components, lesser compared to architecture proposed by Yu-Hsuan et al which is as shown in Fig. L. The HUR is greatly improved. In the proposed VLSI architecture if Mode, clock=0, then X (XNOR) Q will be selected through mux4_1 and fed to inverter. If mode=0, clock=1, then Q will be selected. If Mode=1 and clock=0, then X (XNOR) Q selected. If Mode, clock=1 then X is passed through mux4_1.



FM0 code: Mode = 0; Manchester code: Mode = 1
(i)



FM0 code: Mode = 0; Manchester code: Mode = 1
(ii)

Fig. K. VLSI architecture for FM0/Manchester encoding (i) Unbalance timing computation. (ii) Balance timing computation

Proposed Architecture

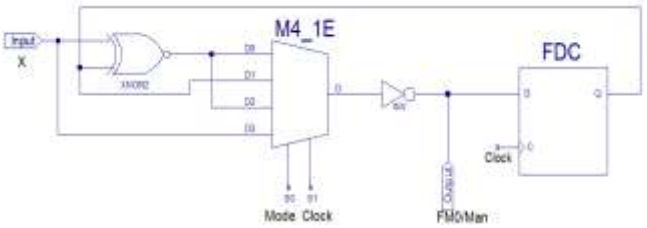


Fig. L. VLSI architecture of FM0/Manchester encoding

Table V. Performance Evaluation of Proposed Architecture

Coding	Active Components / Total Components	UR
FM0	4/ 4	100%
Manchester	4 / 4	100%
Average	4 / 4	100%

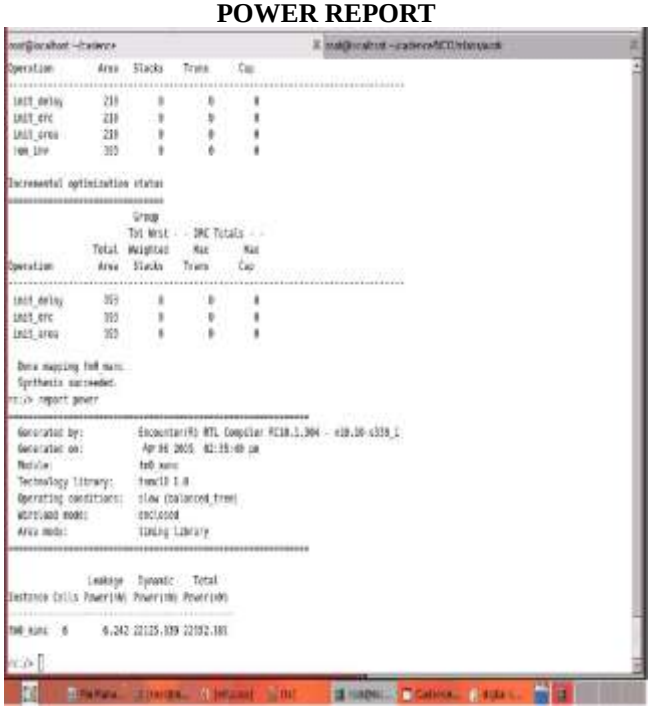
V. EXPERIMENTAL RESULTS

This paper is compared with the existing works. These works are performed in two kinds of design flows. The literatures [3] and [4] are performed with full custom and the literature [5] and [6] are implemented in FPGA. The proposed VLSI architecture is performed in both design flows: FPGA and cadence 180nm technology. As it is said in 180nm technology, the power consumption is 22132.181nW and area is 193μm^2. The FPGA implementation is also performed not only as objective but for functional prototyping.

Table VI. Performance Analysis of Proposed Architecture

	2014	This work
Realization	Xilinx FPGA Spartan 2	Xilinx FPGA Spartan 3
Supply voltage	3.3 V	3.3 V
Coding methods	Manchester FM0	Manchester FM0
Operation frequency	562 MHz	562 MHz
Power consumption	78mW	52 mW
HUR	100%	100%
Delay	5.94ns	5.77ns
FPGA resource usage	Slice : 1 Flip-flop : 1 LUTs : 1 Bonded IOBs : 5	Slice : 1 Flip-flop : 1 LUTs : 1 Bonded IOBs : 6

POWER REPORT



VI. CONCLUSION

The diversity of coding between Manchester and FM0 limits the ability to design and implement fully efficient and reusable VLSI hardware architecture. In this paper, the proposed architecture uses ROLS method to overcome the drawback of the existing architecture. The ROLS method uses two base techniques: area efficient retiming and balance logic simplification. The area efficient technique reduces the area and the balance logic simplification shares and combines both encodings with identical logic components. This paper is implemented in Xilinx FPGA Spartan 3 and high performance result is obtained. This paper is realized in 180nm technology with an device efficiency. The power consumption is 22132.181nW for FM0 and Manchester encoding. The power report is as shown above.

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, 1063-8210, 2014.

[2] F. Ahmed-Zaid, F. Bai, S. Bai, C. Basnayake, B. Bellur, S. Brovold, *et al.*, "Vehicle safety communications—Applications (VSC-A) final report," U.S. Dept. Trans., Nat. Highway Traffic Safety Admin., Washington, DC, USA, Rep. DOT HS 810 591, Sep. 2011.

[3] J. B. Kenney, "Dedicated short-range communications (DSRC) Standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162– 1182, Jul. 2011.

[4] J. Daniel, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich, "Design of 5.9 GHz DSRC-based vehicular safety communication," *IEEE Wireless Commun. Mag.*, vol. 13, no. 5, pp. 36–43, Oct. 2006.

[5] 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.

[6] 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.

[7] 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.

[8] 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.