

# Fully Reused VLSI Architecture of FM0/Manchester Encoding Using SOLS Technique for DSRC Applications

P.KEERTHANA<sup>1</sup>, G.V.RAMANA REDDY<sup>2</sup>

<sup>1</sup>M.Tech Student in Digital Electronics And Communication Systems, NALANDA INSTITUTE OF ENGINEERING TECHNOLOGY, AP

<sup>2</sup>Associate professor, Department of Electronics and Communication Engineering, NALANDA INSTITUTE OF ENGINEERING TECHNOLOGY, AP

**ABSTRACT:** The dedicated short-range communication (DSRC) is an emerging technique to push the intelligent transportation system into our daily life. The DSRC standards generally adopt FM0 and Manchester codes to reach dc-balance, enhancing the signal reliability. Nevertheless, the coding-diversity between the FM0 and Manchester codes seriously limits the potential to design a fully reused VLSI architecture for both. In this paper, the similarity-oriented logic simplification (SOLS) technique is proposed to overcome this limitation. The SOLS technique improves the hardware utilization rate from 57.14% to 100% for both FM0 and Manchester encodings. The performance of this paper is evaluated on the post layout simulation in Taiwan Semiconductor Manufacturing Company (TSMC) 0.18- $\mu\text{m}$  1P6M CMOS technology. The maximum operation frequency is 2 GHz and 900 MHz for Manchester and FM0 encodings, respectively. The power consumption is 1.58 mW at 2 GHz for Manchester encoding and 1.14 mW at 900 MHz for FM0 encoding. The core circuit area is 65.98 $\times$ 30.43  $\mu\text{m}^2$ . The encoding capability of this paper can fully support the DSRC standards of America, Europe, and Japan. This paper not only develops a fully reused VLSI architecture, but also exhibits an efficient performance compared with the existing works.

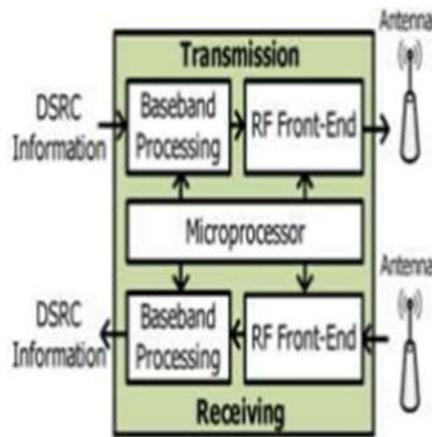
**KEYWORDS:** In DSRC, FSM, FM0, Manchester.

## 1. INTRODUCTION

DSRC communication implemented fundamentally on standards based on interoperability among devices from different

manufacturers. The dedicated short-range communication is a technique for one- or two-way medium range communication especially used for intelligent transportation systems. The DSRC can be briefly divided into two parts i.e. vehicle-to-vehicle and vehicle-to- roadside. In vehicle-to-vehicle, the DSRC activated the message sending and broadcasting among vehicle for safety issues and public information announcement. The Safety issues consist of blind-spot, intersection warning, intercars

distance, and collision-alarm. The vehicle-to- roadside focuses on the intelligent transportation service, such as automatic electronic toll collection (ETC) system. With electronic toll collection, the toll collecting is electrically or automatically accomplished with the contactless ICcard platform. Moreover, the electronic toll collection has application such as payment for parking-service, and gas-refueling. Thus, the DSRC plays an important role in automobile industry. Generally, the Waveform of transmitted signal is expected to have zero mean for robustness noise, and this is also called as dc-balance. The transmitted signal consists of arbitrary binary sequence, (1 or 0) which is difficult to achieve dc-balance. The goal of FM0 and Manchester codes can provide the transmitted signal with dc-balance. Both FM0 and Manchester codes are widely designated in encoding for downlink. The system architecture of DSRC transceiver is shown in Fig 1.



**Fig. 1: System architecture of DSRC transceiver**

The upper and bottom parts are designed for transmission and receiving, respectively. This transceiver is partitioned into three basic modules: microprocessor, baseband processing and RF front-end. The microprocessor takes and manipulates the instructions from media access control to schedule the tasks of baseband processing and RF front-end. The baseband processing is dedicated for encoding, error correction, clock synchronization, and modulation. The RF front-end transmits and receives the wireless signal from antenna for communication.

## 2. LITERATURE SURVEY

In last few years VLSI architecture of Manchester encoder is used in optical communications [1]. A new Manchester code generator designed at transistor level is represented. This Manchester code generator uses 32 transistors and has the same complexity as a standard D flip-flop. The VLSI architecture of Manchester encoder [2] further replaces the architecture of switch in [1] by the NMOS device. It is realized in 90-nm CMOS technology, and its maximum clock frequency is as high as 5 GHz. The high-speed VLSI architecture also fully reused with Manchester and Miller encodings [3] for radio frequency identification (RFID) applications is implemented. This design is realized in 0.35- $\mu\text{m}$  CMOS technology and has the maximum operation frequency is 200 MHz. This design uses concept of parallel operation to improve

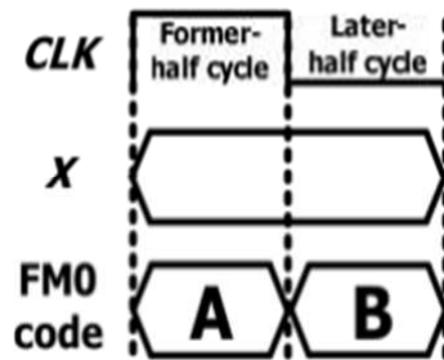
data throughput. In addition, the technique of hardware sharing is improved in this design to reduce the number of transistors. This design uses TSMC CMOS 0.35- $\mu\text{m}$  2P4M technology. A Manchester encoding architecture for ultrahigh frequency (UHF) RFID tag emulator [4] is designed. This hardware architecture is constructed from the finite state machine (FSM) of Manchester code, and is implemented into field-programmable gate array (FPGA) prototyping system. The similar design methodology is further applied to individually construct FM0 and Miller encoders also for UHF RFID Tag emulator [5]. Its maximum operating frequency is about 192 MHz. Furthermore, [6] combines frequency shift keying (FSK) modulation and demodulation with Manchester codec in hardware realization.

## 3. RELATED WORK:

In this part, the clock signal and the input data are denoted as CLK and X. with these parameters the coding principles of FM0 and Manchester codes are explained as follows.

### (i).FM0 Encoding

As given in Fig 2, for each X, the FM0 code structure consists of two parts: one for former-half cycle of CLK, A, and the other one for later-half cycle of CLK, B. Fig.2: Code word structure of FM0.

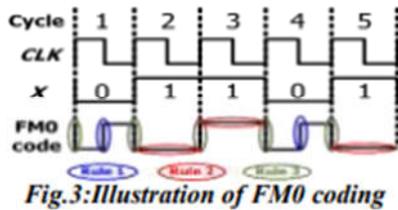


**Fig.2: Codeword structure of FM0.**

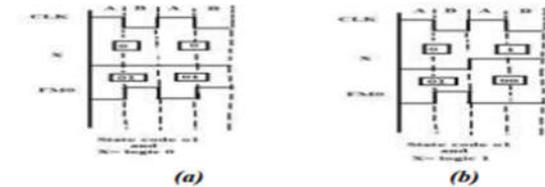
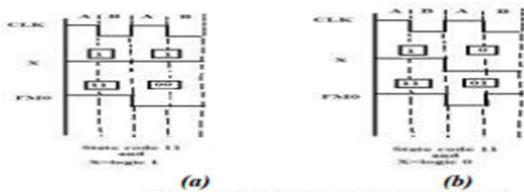
The coding principle of FM0 is listed as the following three rules.

- 1) If X is the logic-0, the FM0 code must exhibit a transition between A and B
- 2) If X is the logic-1, no transition is allowed between A and B.

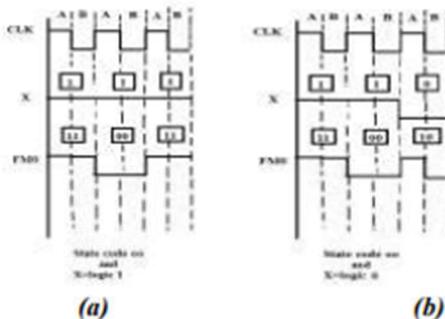
3) The transition is allocated among each FM0Code no matter what the X is. Fig.3 shows the waveform of FM0 encoding.



As shown in fig 4, for each X, FM0 has two parts. if state code is 11 and X=logic 1 then next state for FM0 is 00 and if state code is 11 and X=logic 0 then next state for FM0 is 01.



As shown in fig 5, for each X, FM0 has two parts. if state code is 01 and X=logic 0 then next state for FM0 is 01 and if state code is 01 and X=logic 1 then next state for FM0 is 00. As shown in fig 6, for each X, FM0 has two parts. if state code is 00 and X=logic 1 then next state for FM0 is 11 and if state code is 00 and X=logic 0 then next state for FM0 is 10.



As shown in fig 7, for each X, FM0 has two parts. if state code is 10 and X=logic 0 then next state for FM0 is 10 and if state code is 10 and X=logic 1 then next state for FM0 is 11.

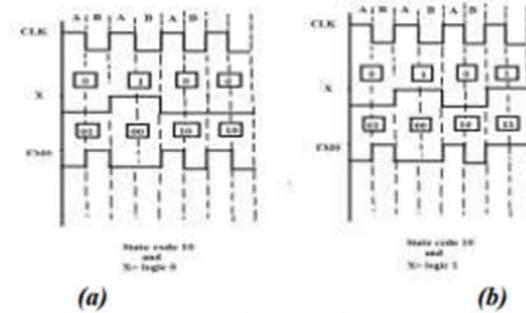


Fig 7: Waveform for State code 10  
**(ii).Manchester Encoding**

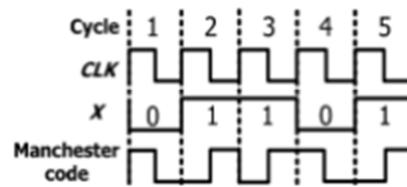


Fig 8: Illustration of Manchester coding example The Manchester coding example is shown in Fig. 8. The Manchester code is obtained from  $X \oplus CLK$ . (1) The Manchester encoding is designed with a XOR operation for X and CLK. The clock always has always transition within one cycle, and so does the Manchester code no matter what the X is.

**4. HARDWARE ARCHITECTURE OF FM0 AND MANCHESTER ENCODERS**

This is the hardware architecture of the fm0/Manchester code. the top part is denoted the fm0 code and then the bottom part is denoted as the Manchester code. in fm0 code the DFFA and DFFB are used to store the state code of the fm0 code and also mux\_1 and not gate is used in the fm0 code. when the mode=0 is for the fm0 code. the Manchester code is developed only using the XOR gate and when the mode=1 is for the Manchester code. the hardware utilization rate is defined as the following

$$HUR = \frac{\text{Active components}}{\text{total components}} \times 100$$

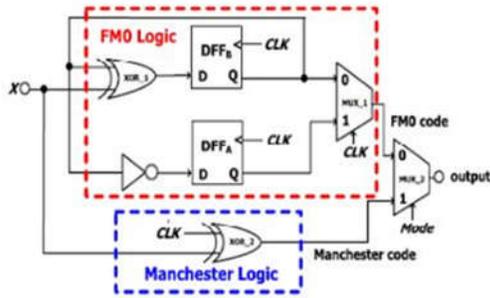


Fig 9. Hardware architecture

The active components means the components are work in the both fm0 and Manchester code. The total components means the number of the components are present in the hole circuit. the HUR rate is given below the following sect

TABLE-II

Coding	Active components(transistor count)/ total components (transistor count)	HUR
FM0	6(86)/7(98)	85.71%
MANCHESTER	2(26)/7(98)	28.57%
AVERAGE	4(56)/7(98)	57.14%

For both the encoding methods the total components is 7. for the fm0 code the total component is 7 and then the active component is 6. in Manchester code the total component is 7 the active component is 2. in both coding having 98 transistors are used without SOLS. The fm0 having 86 transistor, and then the Manchester having the 26 transistor. the average for both coding is 56 transistors. In proposed work reduce the total components from 7 to 6 and reduce the transistor counts. In this paper two multiplexer is used in proposed work reduce two multiplexer from one multiplexer, when reduce the multiplexer the total components are reduced the area and then the power consumption also reduced.

**FM0 AND MANCHESTER ENCODER USING SOLS TECHNIQUE:**

The SOLS technique is classified into two parts area compact retiming and balance logic operation sharing

**A. area compact retiming**

For fm0 the state code of the each state is stored into DFFA and DFFB .the transition of the state code is only depends on the previous state of B(t-1) instead of the both A(t-1) and B(t-1)

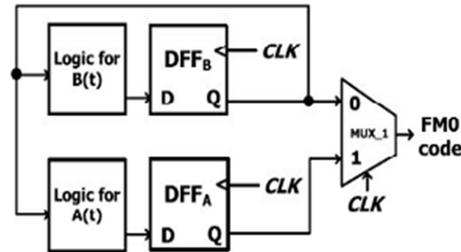


Fig 10. Area compact retiming

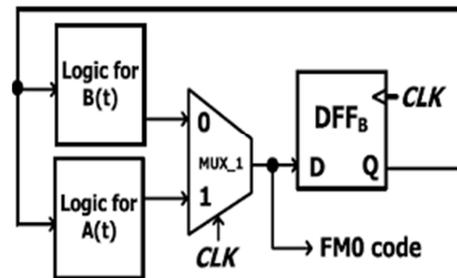


Fig11. FM0 encoding without area compact retiming.

The previous state is denoted as the A(t-1) and then the B(t-1). and then the current state is denoted as the A(t) and then the B(t).

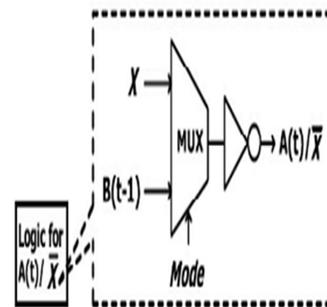


Fig 12. FM0 encoding with area compact retiming.

Thus, the FM0 encoding just requires a single 1-bit flip-flop to store the previous value B(t-1). If the DFFA is directly removed, a non synchronization between A(t) and B(t) causes the

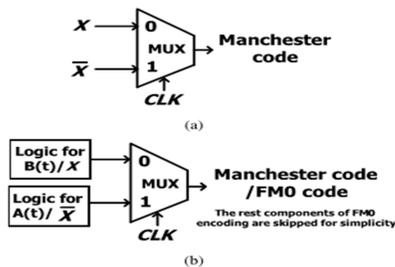
logic fault of FM0 code. To avoid this logic-fault, the DFFB is relocated right after the MUX-1, where the DFFB is assumed be positive-edge triggered flip flop. At each cycle, the FM0 code, comprising A and B, is derived from the logic of A(t) and the logic of B(t), respectively. The FM0 code is alternatively switched between A(t) and B(t) through the MUX-1 by the control signal of the CLK. In the Q of DFFB is directly updated from the logic of B(t) with 1-cycle latency. when the CLK is logic-0, the B(t) is passed through MUX-1 to the D of DFFB. Then, the upcoming positive-edge of CLK updates it to the Q of DFFB. the timing diagram for the Q of DFFB is consistent whether the DFFB is relocated or not. the B(t) is passed through MUX-1 to the D of DFFB. Then, the upcoming positive-edge of CLK updates it to the Q of DFFB. the timing diagram for the Q of DFFB is consistent whether the DFFB is relocated or not. The transistor count of the FM0 encoding architecture without area-compact retiming is 72, and that with area-compact retiming is 50. The area-compact retiming technique reduces 22 transistors.

**B. Balance logic operation sharing**

As mentioned previously, the Manchester encoding can be derived from  $X \oplus \text{CLK}$ , and it is also equivalent to

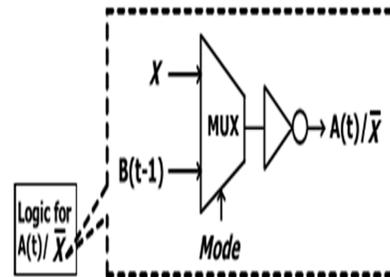
$$X \oplus \text{CLK} = X \text{CLK} + \bar{X} \text{CLK}. \quad (6)$$

This can be realized by the multiplexer, as shown in Fig. 13(a).



**Fig13.(a),(b)**

It is quite similar to the Boolean function of FM0 encoding in (4). By comparing with (4) and (6), the FM0 and Manchester logics have a common point of the multiplexer like logic with the selection of CLK. As shown in Fig. 13(b), the concept of balance logic-operation sharing is to integrate the  $X$  into  $A(t)$  and  $X$  into  $B(t)$ , respectively.



**Fig. 14. The logic for A(t)/X is shown in**

The  $A(t)$  can be derived from an inverter of  $B(t - 1)$ , and  $X$  is obtained by an inverter of  $X$ . The logic for  $A(t)/X$  can share the same inverter, and then a multiplexer is placed before the inverter to switch the operands of  $B(t - 1)$  and  $X$ . The Mode indicates either FM0 or Manchester encoding is adopted. The similar concept can be also applied to the logic for  $B(t)/X$ , as shown in Fig. 15(a). Nevertheless, this architecture exhibits a drawback that the XOR is only dedicated for FM0 encoding, and is not shared with Manchester encoding. Therefore, the HUR of this architecture is certainly limited. The  $X$  can be also interpreted as the  $X \oplus 0$ , and thereby the XOR operation can be shared with Manchester and FM0 encodings. As a result, the logic for  $B(t)/X$  is shown in Fig. 15(b), where the multiplexer is responsible to switch the operands of  $B(t-1)$  and logic-0. This architecture shares the XOR for both

$B(t)$  and  $X$ , and thereby increases the HUR. Furthermore, the multiplexer in Fig. 15(b)

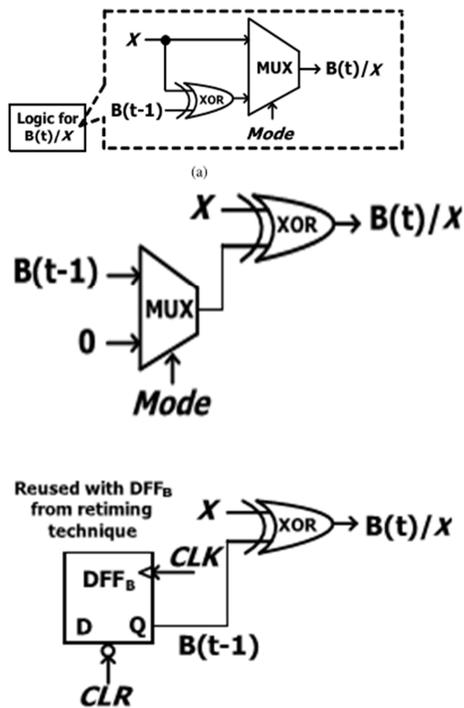


Fig. 15. Balance logic-operation sharing of  $B(t)$  and  $X$ . (a) Without the XOR sharing. (b) With XOR sharing. (c) Sharing of the reused DFFB from area-compact retiming technique.

can be functionally integrated into the relocated DFFB from area-compact retiming technique, as shown in Fig. 15(c). The CLR is the clear signal to reset the content of DFFB to logic-0. The DFFB can be set to zero by activating CLR for Manchester encoding. When the FM0 code is adopted, the CLR is disabled, and the  $B(t - 1)$  can be derived from DFFB. Hence, the multiplexer in Fig. 15(b) can be totally saved, and its function can be completely integrated into the relocated DFFB. The proposed VLSI architecture of FM0/Manchester encoding using SOLS technique is shown in Fig. 16(a). The logic for  $A(t)/X$  includes the MUX-2 and an inverter. Instead, the logic for  $B(t)/X$  just incorporates a XOR gate. In the logic for  $A(t)/X$ , the computation time of MUX-2 is almost identical to that of XOR in the logic for  $B(t)/X$ . However, the logic for  $A(t)/X$  further incorporates an inverter in the series of MUX -2. This unbalance computation time between  $A(t)/X$  and

$B(t)/X$  results in the glitch to MUX-1, possibly causing the logic-fault on coding. To alleviate this unbalance computation time, the architecture of the balance computation time between  $A(t)/X$  and  $B(t)/X$  is shown in Fig. 16(b). The XOR in the logic for  $B(t)/X$  is translated into the XNOR with an inverter, and then this inverter is shared with that of the logic for  $A(t)/X$ . This shared inverter is relocated backward to the output of MUX-1. Thus, the logic computation time between  $A(t)/X$  and  $B(t)/X$  is more balance to each other. The adoption of FM0 or Manchester code depends on Mode and CLR. In addition, the CLR further has another individual function of a hardware initialization. If the CLR is simply derived by inverting Mode without assigning an individual CLR control signal, this leads to a

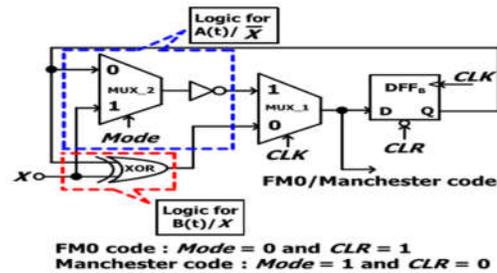


Fig 16(a). Unbalance computation time between  $A(t)/X$  and  $B(t)/X$ .

conflict between the coding mode selection and the hardware initialization. To avoid this conflict, both Mode and CLR are assumed to be separately allocated to this design from a system controller. Whether FM0 or Manchester code is adopted, no logic component of the proposed VLSI architecture is wasted. Every component is active in both FM0 and Manchester encodings. Therefore, the HUR of the proposed VLSI architecture is greatly improved.

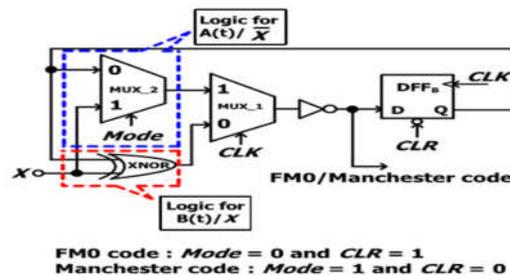


Fig 16(b).Balance computation time between  $A(t)/X$  and  $B(t)/X$

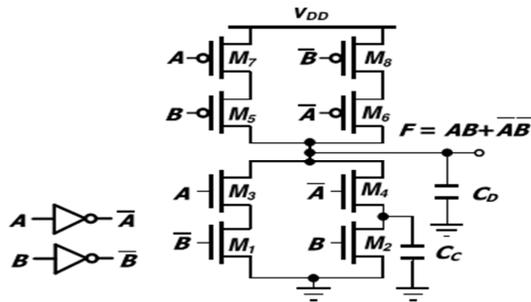
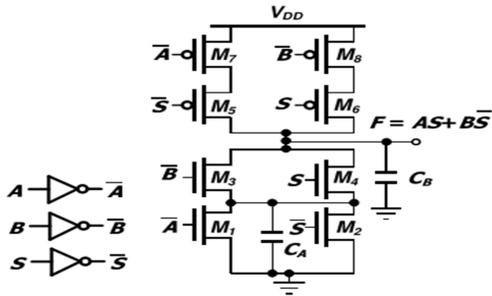


Fig.17.Static CMOS topologies of multiplexer and XNOR. (a) Two-input multiplexer. (b) Two-input XNOR.

In Fig. 17(b), for static CMOS, the series of MUX-1 and MUX-2 dominates Manchester encoding path and leads to a total propagation delay as

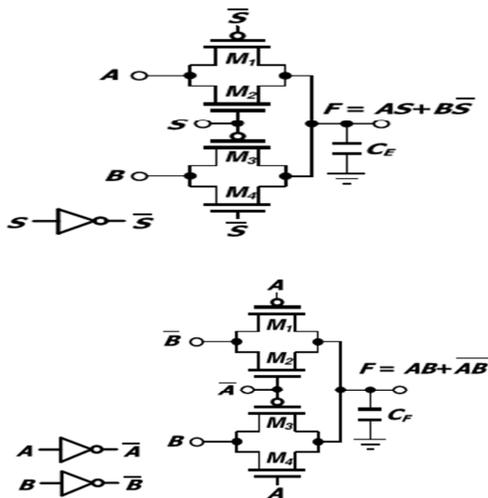


Fig. 18. Transmission-gate multiplexer and XNOR. (a) Two-input multiplexer.(b) Two-input XNOR.

In Fig. 18(b), accordingly, the circuit architectures of MUX-1, MUX-2, and XNOR are designed with transmission-gate logic. Hence, only two-stage of transmission-gate logic is adopted, and no extra buffer-insertion is required in this design.

5. RESULTS

The proposed and existing design is simulated using Model Sim6.4b Synthesis: Xilinx ISE. The below figures shows the RTL schematic of existing and proposed model. with clk, mode, x as input to existing design and out as output ,and clk,mode,rst,x0 as inputs to the proposed design and fm and out are the output of proposed design

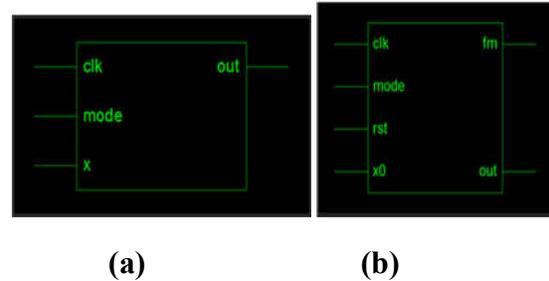


Fig19.(a)existing design RTL schematic,(b)proposed design RTL schematic.

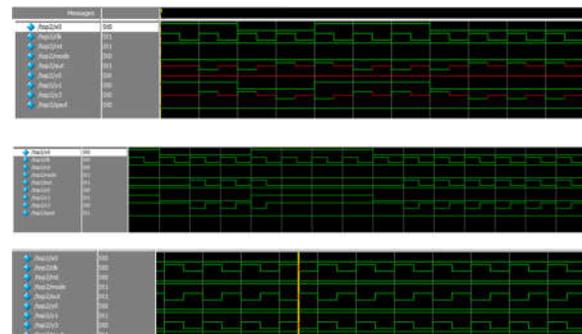


Fig 20. Simulation outputs of existing design

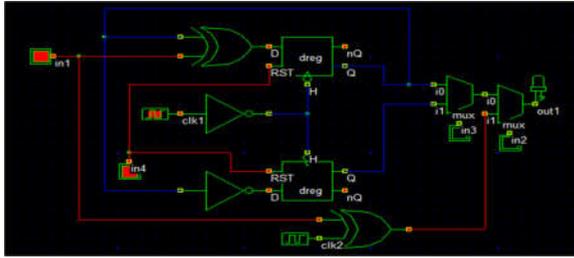


Fig 21. Digital schematic of proposed design

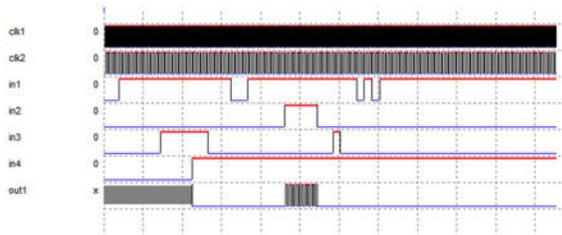


Fig 22. Wave form of proposed design.

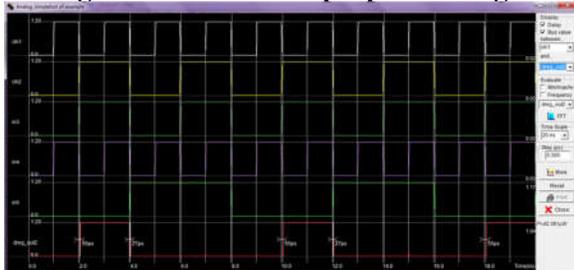


Fig 23. Simulation result of proposed design.

## 6. CONCLUSION

The coding-diversity between FM0 and Manchester encodings causes the limitation on hardware utilization of VLSI architecture design. A limitation analysis on hardware utilization of FM0 and Manchester encodings is discussed in detail. In this paper, the fully reused VLSI architecture using SOLS technique for both FM0 and Manchester encodings is proposed. The SOLS technique eliminates the limitation on hardware utilization by two core techniques: area compact retiming and balance logic-operation sharing. The area-compact retiming relocates the hardware resource to reduce 22 transistors. The balance logic-operation sharing efficiently combines FM0 and Manchester encodings with the identical logic components. This paper is realized in

TSMC0.18- $\mu\text{m}$  1P6M CMOS technology with an outstanding device efficiency. The maximum operation frequency is 2 GHz and 900 MHz for Manchester and FM0 encodings, respectively. The power consumption is 1.58 mW at 2 GHz for Manchester encoding and 1.14 mW at 900 MHz for FM0 encoding. The core circuit area is  $65.98 \times 30.43 \mu\text{m}^2$ . The encoding capability of this paper can fully support the DSRC standards of America, Europe, and Japan. This paper not only develops a fully reused VLSI architecture, but also exhibits a competitive performance compared with the existing works.

## REFERENCES

- [1] 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.
- [2] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, vol. 99, no. 7, pp. 1162–1182, Jul. 2011.
- [3] 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.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [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.
- [9] J.-H. Deng, F.-C. Hsiao, and Y.-H. Lin, "Top down design of joint MODEM and CODEC detection schemes for DSRC coded-FSK systems over high mobility fading channels," in *Proc. Adv. Commun. Technol.*, Jan. 2013, pp. 98–103.
- [10] I.-M. Liu, T.-H. Liu, H. Zhou, and A. Aziz, "Simultaneous PTL buffer insertion and sizing for minimizing Elmore delay," in