IJAEGT
ISSN No: 2309-4893
International Journal of Advanced Engineering and Global Technology
Vol-04, Issue-01, January 2016

Fully Reused VLSI Architecture of Miller Encoding using SOLS Technique for DSRC Applications

V. Hemalatha[1], P. Srividhya[2]

1 Student, ECE Department, Sri Ramanujar Engineering College, Chennai, India.

2 Asst. Professor, ECE Department, Sri Ramanujar Engineering College, chennai, India

***Abstract*—Dedicated Short Range Communication is a simplex or duplex short range to medium range wireless communication. It is used to support Intelligent Transport System (ITS) applications such as electronic toll collection, parking lot, border crossing identification et c. It proposes a VLSI architecture design using similarity-oriented logic simplification (SOLS) technique. The SOLS consists of two core methods: area-compact retiming and balance logic-operation sharing. The area-compact retiming relocates the hardware resource to reduce 5 transistors. The balance logic-operation sharing efficiently uses miller encodings with the fully reused hardware architecture. With SOLS technique, we constructs a fully reused VLSI architecture of Manchester and FM0 encodings for DSRC applications. The experiment results reveal that this design achieves an efficient performance compared with sophisticated works. To achieve dc balance, enhancing signal reliability, FMO and Manchester codes are used in DSRC standards. The performance of this VLSI Architecture is evaluated on cadence-post layout simulation tool with 180 nm CMOS technology. The Manchester codes consumes maximum opera tion frequency is 2GHz with 1.58 mW power consumption an d 900 MHz for FMO coding with 1.14 mW power consumption. The DSRC standards with FMO and Manchester encod ing can support America, Europe, and Japan. This paper also shown that area compaction of proposed VLSI architecture compared to existing method.**

Keywords*—Dedicated Short Range Communication (DSRC), Intelligent Transport System (ITS), Similarity-Oriented Logic Simplification (SOLS), Flexible Macro -block Ordering (FMO), Manchester encoding, Media Access Control (MAC).

INTRODUCTION

The DSRC is a protocol simplex o r duplex range communication in the modern automotive industry. The DSRC mainly used for intelligent transportation systems. It can be briefly classified as vehicle to vehicle and vehicle to road-side communications. The DSRC can provide an inter communication between automobiles and road-sides for safety issues and public information announcement. These safety message include blind-spot, collision-alarm, inter cars distance etc. The vehicle to roadside communication mainly focuses on intelligent transport system such as electronic toll collection (ETC) system.

In ETC system, The On-Board Unit (OBU) referred as the vehicle and the Roadside Unit (RU) referred as toll collecting infrastructure. The communication is established between RSU, OBU, when OBU goes under the range of RSU. By using DSRC, the toll-collecting is electrically accomplished with wireless IC-Card platform. This technique eliminates the delay and latency between vehicle and gate way in the toll-collecting system. Moreover, ETC can be established to the payment for parking service, gas-refueling, freight tracking. Thus DSRC leads the major role in modern automobile industry. The transceiver Architecture consists of two parts for transmission and receiving.

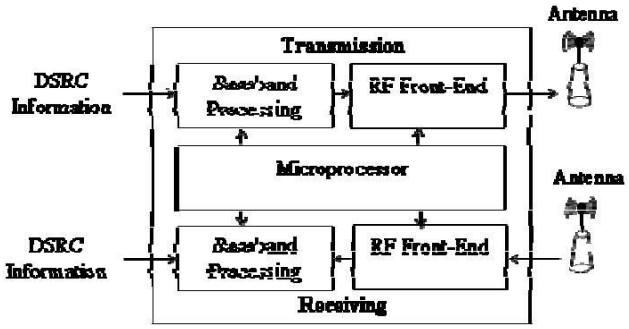


Fig. 1. Architecture of DSRC transceiver

The DSRC transceiver is classified as three basic modules (i) Microprocessor (ii) Baseband Processing (iii) RF-Front End. The Microprocessor used to interprets instructions from Media Access Control (MAC) to schedule the tasks of Baseband Processing and RF-Front End. The modulation, error correction, clock synchronization, encoding all is done by Base Band Processing. The RF-Front End transmits the wireless signal through Antenna and receives the wireless signal through Antenna (i.e. physical transmission).All three modules co -operate with each to perform DSRC protocols. Several countries have been established DSRC Standard s by various organizations. The DSRC standards of America, Europe, Japan all shown in Table.1.With carrier frequency of 5.8GHz and 5.9 GHz, the

data rate of individual organizations as 500 Kbps,27 mbps,4 mbps. The Amplitude Shift Keying (ASK), Phase Shift Keying (PSK),Orthogonal Frequency Division Multiplexing(OFDM) are incorporates in modulation schemes. The encoding for downlink incorporates FMO-Flexible Macro-block Ordering and Manchester. The reliability of data signal is too important in DSRC standards compared to the data rate.

TABLE I.

PROFILE OF DSRC STANDARDS FOR AMERICA,EUROPE,AND JAPAN

	Europe	**America**	**Japan**
Organization	CEN	ASTM	ARIB
Data Rate	500 kbps	27 Mbps	4 Mbps
Carrier Frequency	5.8 GHz	5.9 GHz	5.8 GHz
Modulation	ASK,PSK	OFDM	ASK
Encoding(Downlink)	FMO	Manchester	Manchester

CEN - European Committee for Standardization.
ASTM - American Society for Testing and Materials.
ARIB- Association of Radio Industries and Business.

In common, the robustness issue, the transmitted signal waveform is expected to have zero mean and this is also referred as DC-balance. But due to the arbitrary binary sequence of transmitted signal. It is difficult to achieve DC-balance. The purpose of FMO, Manchester codes can provide the transmitted signal with better DC-balance. The DC-balance is an important role in DSRC. But due to the arbitrary binary sequence of transmitted signal, it is difficult to achieve DC-balance. The purpose of FMO, Manchester codes can provide the transmitted signal with better DC-balance.

Literature Reviews

The literature [1] shows the Architectural view of Manchester encoder for optical communication, Here the Manchester encoder is designed by using CMOS inverter and the gate inverter as the switch. The operation frequency is 1 GHz and implemented by 0.35 µm CMOS technology.

The Literature [2] shows that the design of Manchester encoder by nMOS device instead of switch [1] with 90 nm CMOS technology. The maximum operation frequency is as high as 5 GHz.

The literature [3] explains high-speed VLSI Architecture of Manchester and miller encodings for RFID application with 0.35 µm CMOS technology and maximum operation frequency is 200 MHz

The literature [4] also proposes the Architecture of Manchester encoding for Ultra High Frequency (UHF) RFID tag emulator. By using FSM, the hardware Architecture is constructed and implemented in FPGA prototyping system, the maximum operational frequency is 256 MHz

Main Theme of this Paper

The diversity between FMO and Manchester coding seriously limits the potential to design a VLSI Architecture that can be fully reused with each other. The hardware utilization place a major role in DSRC applications. There are two technology used for hardware utilization in DSRC applications.

(i) Reused-Oriented Boolean Simplification (ROBS).

(ii) Similarity-Oriented Logic Simplification (SOLS).

This paper proposes the design of VLSI Architecture using SOLS technology. It consists of two major mechanisms. (i) Compact Area with retiming (ii) Sharing of Logic Operation. The Compact Area with Retiming that compact the area by reducing 22 transistors in the hardware design. The Sharing of Logic Operation that achieves fully reused hardware Architecture by combining FMO and Manchester encodings. By using above technique, the FMO and Manchester encoding Architecture are designed for DSRC applications. The result analysis shows this design achieves an efficient performance compared with previous and complex works.

Synopsis

This paper is organized as follows: Section-II explains the coding principles of FMO and Manchester encoding. Section-III Shows the VLSI Architecture design with hardware of FMO and Manchester encoder. The limitation of hardware resource is also discussed. In Section-IV, Using SOLS technique the VLSI Architecture of FMO, Manchester encodings are designed .Section-IV, The experimental result is presented on FPGA, Tanner-EDA simulation for both FMO, Manchester encodings.

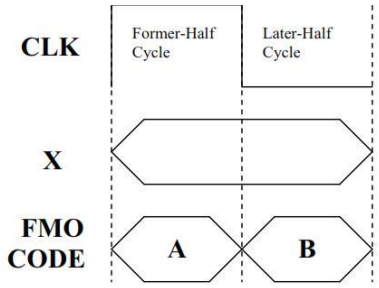


Fig. 2. Coding of FMO

FMO AND MANCHESTER CODING PRINCIPLES

Here, the clock signal is abbreviated as CLK, Data signal is abbreviated as X.

Coding Conditions of FMO

The FMO code consists of two former-half cycle for each X. It is represented by A, B Respectively as shown in

The .Three main conditions are followed i n FMO as listed below (i) The FMO code must allow the tra nsition between A and B, when X is the logic-0.(ii)There is no transition is allowed between A,B, when X is logic-1.(iii)There must be a transition between two consecutive input datum.

The example of FMO coding is at first CLK cycle is logic -0, there must be a transition between A and B according to the condition-1.According to the condition-2, there is no transition in entire CLK cycle of 2, when X is logic-1.Then, According to the c ondition-3, there must be a transition between two consecutive input datum when the logic-1 is changed to logic-0 in the beginning of cycle2.

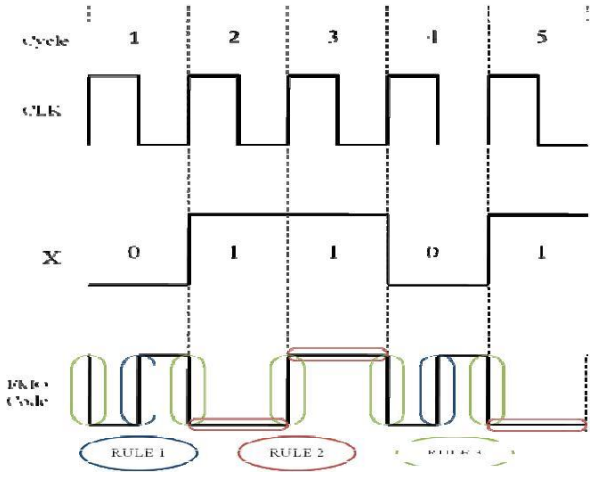


Fig. 3. Example of FMO coding

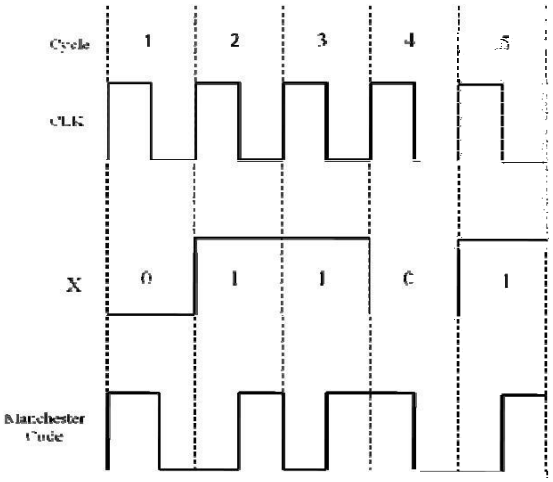
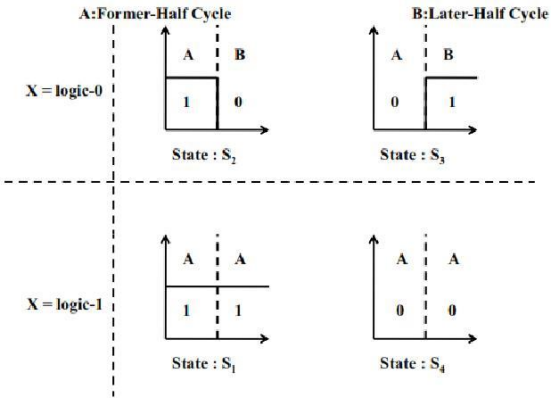
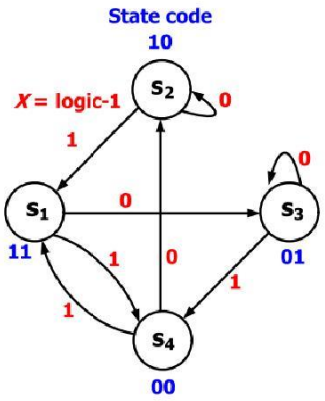


Fig. 4. Example of Manchester coding



(a)



(b)

Fig. 5. FSM for FMO. (a) Definition of states (b) FSM of FMO

TABLE II. S TATE TRANSITION OF FMO

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

Coding conditions of Manchester

The operation of Manchester code is divided from

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

It is realized with XOR operation for C LK and X and it is given above.

HARDWARE ARCHITECTURE OF FMO AND MANCHESTER ENCODER

The hardware Architecture is mainly used to analyze the hardware utilization of both F MO, Manchester encoder. The hardware Architecture of Manchester encoding is simple XOR operation. But it is very difficult to construct hardware Architecture of FMO compare with Manchester. To construct the FMO hardware Architecture should start with FSM of FMO.

TABLE III. HUR OF FMO AND MANCHESTER ENCODINGS

Coding	Active Components(transistor count) / Total Components(transistor count)	HUR
FMO	6 (86) / 7 (98)	85.71%
Manchester	2 (26) / 7 (98)	28.57%
Average	4 (56) / 7 (98)	57.14%

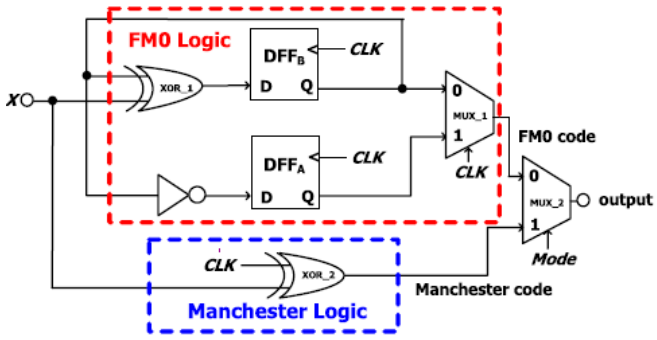
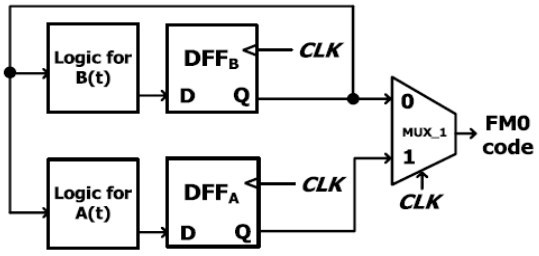


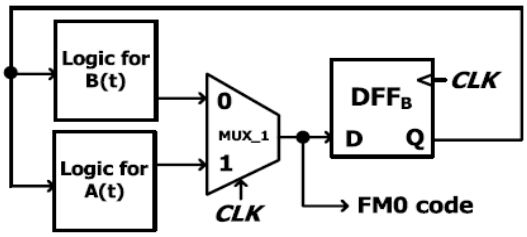
Fig. 6. VLSI Hardware architecture of FMO, Manc hester encodings.

FSM of FMO encoder

The FSM of FMO code is classified into 4 states as shown in Figure. Each state code contains A ,B as shown in The FSM of FMO is shown in Figure. Assume the starting state is S1and its state code is 11 for A,B. If the value of X is logic-0 can satisfy both conditions 1,3 and the state transition goes to S3.if the X is logic-1,can satisfy both conditions 2,3 and the state-transition goes to S4.Thus the state-transition can be constructed completely.



(a)



(b)

Fig. 7. (a) FMO encoding without area-compact retiming (b) FMO encoding with area-compact retiming

Transition Table of FMO

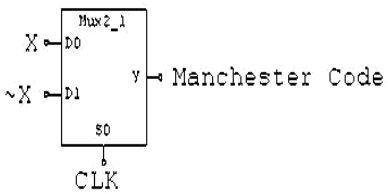
The FSM of FMO can also constructed by transition Table of each state, as show n in Table II.A(t),B(t) denotes present state of state code at time instant t and their previous state denoted as A(t-1),B(t -1).By using this, the Boolean function are

$$A(t) = B\,t \quad 1 \qquad (2)$$

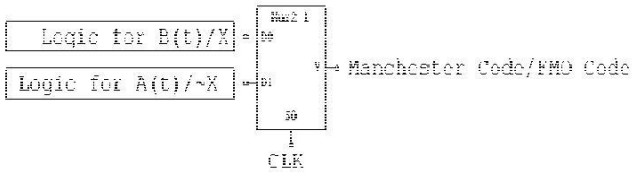
$$B(t) = X \oplus B(t\text{-}1) \qquad (3)$$

Using A(t),B(t) the FMO code would be CLK

$$CLK\ A(t) + CLK\ B(t) \qquad (4)$$



(a)



(b)

Fig. 8. Working of Sharing of Logic Operati on (a) Manchester encodings in Multiplexer.(b) Combination of FM O and Manchester encodings

The hardware Architecture of FMO, Manchester encoder is from that the FMO encoding depends on X, previous state of FMO code. The DFF's are used to store the stated code of FMO code, the MUX1 is to select A(t), B(t) with CLK signal. The MUX2 is to switch the FMO code and Manchester code through mode=D0, mode=D1 respectively. The Hardware Utilization Rate (HUR) is used to determine the hardware utilization.

$$HUR = \frac{A_C}{T_C} \times 100\% \quad (5)$$

TABLE IV. TRANSISTOR COUNT OF FM O ENCODING ARCHITECTURE

	Without area-compact retiming	With area-compact retiming
PMOS	36	25
NMOS	36	25
Total	72	50

The component represent hardware elem ents like AND, OR, NOT, FF and the component that wo rk for FMO of Manchester encoding. The total component represents the hardware component in entire Architectur e. The HUR of FMO, Manchester encoding is listed above.

PROPOSED SYSTEM

The SOLS technique mainly used to design a fully reused VLSI Architecture for both encoding s and consist of two major parts Compact Area with Retiming and Sharing of logic operation.

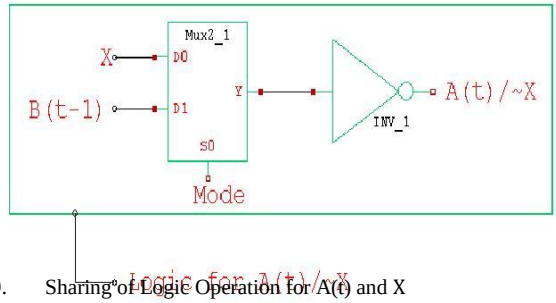


Fig. 9. Sharing of Logic Operation for A(t) and X

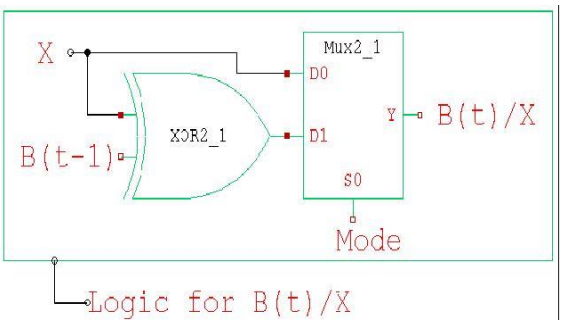
Sharing of Logic Operation

The Manchester encoding denoted as X $\oplus$ CLK and it is also equal to CLK

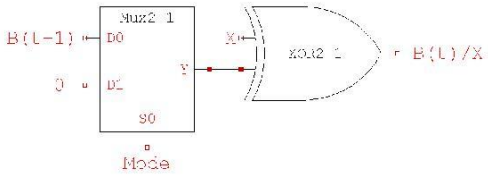
Compact of Area with Retiming

The logic representation of hardware Architecture of FMO, Manchester encoding is shown below. X is removed for a concise representation. Each state of FMO code is stored in DFF1,DF F2 and the state codes depends only B(t-1).So, Only one Flip-Flop is used to store B(t-1).If the DFF1 is directly remove d, a non synchronization effect causes between A(t),B(t) and gives fault in the output.

To overcome the above demerit, the DFF2 is relocated right side of the MUX1 is shown below. Here the DFF2 is assumed to positive-edge triggered. With each cycle, the FMO code comprising A, B as derived from A (t),B(t). The output D of D FF2 is directly updated from the logic of B (t) with 1 cycle latency.

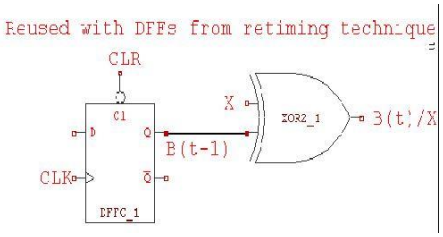


(a)



(b)

The logic-components of FMO encoder are realized with the logic-family of static C MOS, and the total transistor count is shown in Table IV.



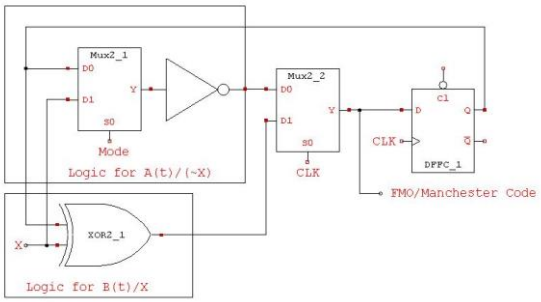
(c)

Fig. 10. Logic operation Sharing of B(t),X.(a) Without XOR sharing.(b) With XOR sharing.(c) Reused DFF 1

$$X\ CLK + X\ CLK \quad (6)$$

Using multiplexer, the Boolean function of FMO, Manchester encoding are quite similar. Both having the

common point of the MUX with the selection of CLK. By integrate X into A(t) and X into B(t),the sharing of logic operation shows the logic for A(t)/ X .From (2) A(t) is the inverter of B(t-1) and X is the inverter of X from (3). The main drawback of this Architecture shows that the XOR is only dedicated for FMO coding and not for Manchester encoding. So, the HUR is limited of this Architecture while X can be derived by X $\oplus$ 0, the XOR operation can be shared by both encodings shown in Fig.10 (b).



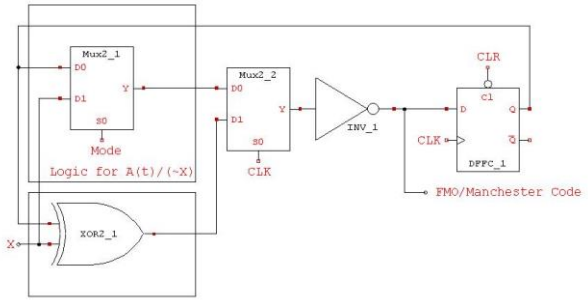
FMO code: Mode = 0; CLR = 1

Manchester code:Mode = 1: CLR = 0

Fig. 11. Unbalance computation time between A(t)/ X,B(t)/X

But this Architecture increases the HUR, So the DFF2 is relocated to achieve compact of area with retiming. The CLR is the clear signal to make the DFF2 as logic-0 for Manchester. The proposed VLSI Architecture of FMO/Manchester encoding using SOLS technique is shown above. When the logic for B(t)/X is enable, the computation time of MUX2 is almost identical and also an inverter is connected in the series of MUX2.

This causes unbalance computation time between A(t)/ X ,B(t)/X.As the result, there is glitches in MUX1 and causing logic fault on coding. To alleviate above, the Architecture of the balance computation time between A(t)/ X ,B(t)/X is shown above. Here the inverter is shared by logic for A(t)/ X ,B(t)/X and it is relocated right after the MUX1.Thus the balance logic computation time is achieved between A(t)/ X,B(t)/X.



FMO code: Mode = 0; CLR = 1

Manchester code: Mode = 1: CLR = 0

Fig. 12. (b) Balance computation time between A(t)/ X,B(t)/X

The selection of FMO or Manchester code depends on the control of mode, CLR. The CLR has another separate function of hardware initialization. Without assigning an individual CLR control signal, the inverting mode of CLR leads conflict between selection of mode and initialization of hardware. By using system monitor, the design of both modes, CLR is constructed. This leads to avoid the conflict between selections of mode and also none of the logic component is wasted in proposed VLSI Architecture.

Evaluation of Performance of the SOLS Technique

The performance evaluation is shown in Table. V. The total numbers of components are reduced from 7 to 5 using SOLS technique. Without SOLS technique, the individual hardware Architecture of both coding with a poor HUR of 57.14%.

Result Analysis

By comparing this paper with previous papers this paper is implemented in two different types of design flows.(i)full custom design flow. (ii)FPGA design flow. The full custom is realized in the literature [1],[2] and The FPGA design flow is designed in literature [4],[6].

Full Custom Design Flow

With 0.18µm 1P6M CMOS technology the design of full custom design flow dine by Taiwan Semiconductor Manufacturing Company (TSMC).the full custom design flow is simulated by Tanner EDA simulation tool.

FPGA Design Flow

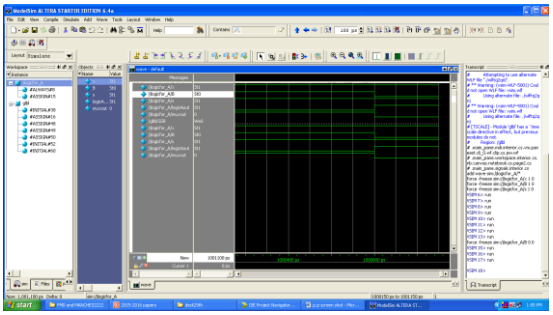
The design flow of FPGA is realized by XILINX development board is shown in Fig.12.The signal transition is not aligned to positive and negative edge trigger of both encodings. The synchronization of FPGA is achieved by two sets of clock signals are used as CLK_{EXT},CLK_{INT} .the frequency of CLK_{EXT} is more efficient than CLK_{INT} .Because the frequency of CLK_{EXT} is twice fast as CLK_{INT}. The synchronization of every signal inside FPGA is done by CLK_{EXT} and The manipulation of FMO, Manchester code is done by CLK_{INT}.

Conclusion

The hardware utilization is limited by the diversity between both FMO and Manchester encodings in VLSI Architectural design. This limitation is analyzed in detail. This paper shows the encoding technique of FMO and Manchester with SOLS technique eliminates the limitation of hardware utilization by two core techniques (a)Compact of Area Retiming (b)Sharing of Logic Operation.

Using compact of area retiming, the number of transistor is reduce to 22 transistors. The sharing of logic operation combines FMO and Manchester encodings. The maximum operating frequency of Manchester, FMO encodings are 2 GHz, 900MHz with consumption of power 1.58mW, 1.14mW respectively. The encoding technique used in DSRC standards is fully supported in America, Europe and Japan.

SIMULATION RESULT



ACKNOWLEDGEMENT

The authors would like to thank prof.Yu-Husuan Lee and Mr.Cheng-Wei Pun for their valuable comments and also like to thank each reviewer for their suggestions.

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] I.-M. Liu, T.-H. Liu, H. Zhou, and A. Aziz, "Simultaneous PTL buffer insertion and sizing for minimizing Elmore delay," in Proc. Int.Workshop Logic Synth., May 1998, pp. 162–168.