ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

Low Power VLSI Architecture Of Encoder For Downlink Applications

[1]Mohanraj S. and [2]Dr. Sudha S.

[1]*PG Student, Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai-89, India.*
[2]*Professor, Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai-89, India.*

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised form 20 March 2015
Accepted 25 March 2015
Available online 10 April 2015

Keywords:
Dedicated Short Range Communication (DSRC), Similarity oriented Logic simplification (SOLS).

ABSTRACT

DSRC is an emerging technique to push the intelligent transportation system into our daily life. The Coding Diversity between the encoding techniques limits the potential to design VLSI architecture. To Overcome the Limitation, SOLS (Similarity oriented Logic simplification) techniques is adopted in designing encoder architecture which eliminates the limitations by two core techniques such as area-compact retiming and Balance logic operation sharing. Area-compact retiming reduces logic elements and improved Hardware utilization rate. The Balance Logic operation sharing technique combines encodings techniques to produce balanced computation time. This Project Not only develops encoder VLSI architecture but also Exhibits an efficient performance compared with existing encoder in terms of power and area. Due to the improved hardware utilization, area is compact and Power consumption gets reduced in encoding techniques. The performance is evaluated on the post layout simulation in Taiwan Semiconductor Manufacturing Company (TSMC) 0.18-µm 1P6M CMOS technology.

To Cite This Article: Mohanraj S. and Dr. Sudha S., Low Power VLSI Architecture of Encoder for Downlink Applications. ***Aust. J. Basic & Appl. Sci.,*** *9(15): 43-51, 2015*

INTRODUCTION

Dedicated Short Range Communications (DSRC) are the communication media of choice for active safety systems. It operates in a licensed frequency band where it is primarily allocated for vehicle safety applications (Ahmed-Zaid, F., 2011). Dedicated short range communication provides a secure wireless interface required by active safety applications (Benabes, P., 2003; Daniel, J., 2006). It supports high speed, low latency and also particularly known for its short-range wireless communication. Though its works in short-range communication, it can be used in high vehicle speed mobility conditions. Dedicated short range communications are designed to perform under extreme weather conditions like rain, fog, snow, etc.

Dsrc Encoding Techniques:

Communications Based active safety systems need a tightly controlled spectrum for maximized reliable transmission. DSRC communications take place over a dedicated 75 MHz spectrum band around 5.9 GHz, allocated by the US Federal Communications.

DSRC was developed with a primary goal of enabling vehicular safety application. There are many modulation techniques like Return-to-Zero (NRZ) and Return-to-Zero (RZ) which are used in communication devices.

But dedicated short Range communication uses FM0 and Manchester encoding (Hung, V., 2009) in its architecture. Generally, the waveform of transmitted signal is expected to have zero mean for robustness issue and this is also referred to as dc-balance. The transmitted signal consists of arbitrary binary sequence, which is difficult to obtain dc-balance (Kenney, B., 2011). Both FM0 and Manchester codes can provide the transmitted signal with dc-balance. For this reason, dedicated short range communication prefers FM0 and Manchester encoding techniques. Also to enhance signal reliability, both FM0/Manchester Encoding is highly preferred.

Paper Overview:

Principles of FM0 Encoding:

A transition occurs at the beginning of each clock cycle. A binary “0” is represented by an additional transition at the middle of the clock cycle and a binary “1” is represented by no transition at the middle of the clock cycle.

Corresponding Author: Mohanraj S., Department of Electronics and Communication Engineering, Easwari Engineering College, Chennai-89, India.
E-mail: Mohanrajeec@gmail.com

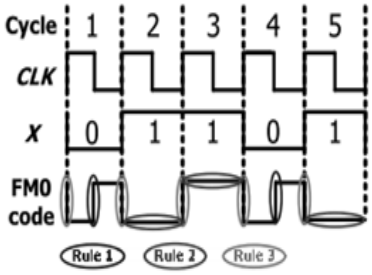


Fig. 1: Illustration of FM0 Encoding.

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 FM0 code.

A FM0 coding example is shown in Figure 1 .At cycle 1, the *X* is logic-0; therefore, a transition occurs on its FM0 code, according to rule 1. For simplicity, this transition is initially set from logic-0 to -1. According to rule 3, a transition is allocated among each FM0 code and thereby the logic-1 is changed to logic-0 in the beginning of cycle 2. Then, according to rule 2, this logic-level is hold without any transition in entire cycle 2 for the *X* of logic-1. Thus, the FM0 code of each cycle can be derived with these three rules mentioned earlier.

Principles Of Manchester Encoding:

Manchester encoding is a synchronous clock encoding technique used to encode the clock and data of a synchronous bit stream. It uses the rising or falling edge in the middle of each bit time to indicate a 0 or 1. Manchester encoding requires two frequencies: the base carrier and two times the carrier frequency, because it changes in the middle of the bit time.

The actual binary data to be transmitted are not sent as a sequence of logic 1s and 0s; instead, the bits are translated into a slightly different format. The data bit 1 from the level-encoded signal is represented by a full cycle of the inverted signal from the master clock, which represents 0 to 1 transmission.

To clarify, In the Figure 2 there are two bits of Manchester encoded data for each bit of original data. If the original data is a 0, the Manchester code is 0 to 1, and it shifts to an upward transition at bit center. If the original data is a 1, the Manchester code is 1 to 0 and it shifts downward at bit center.

Advantages Of Manchester Encoding:

- Manchester encoding has certain advantages over previously used encoding techniques.
- One transition occurs for every bit cycle which is used to perform clock recovery.
- Transitions detection ratio is good and it produces less failure rate.
- Unlike with NRZ Encoding, no separate clock is not required.
- There is no long strings between logic 1 and logic 0. It results in less data error during transmission.

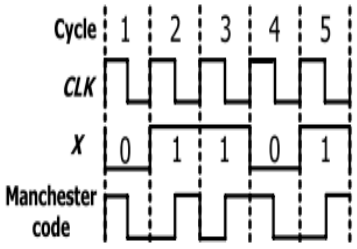


Fig. 2: Illustration of Manchester Encoding.

Limitation Analysis Of FM0 Encoder And Manchester Encoder:

To make an analysis on hardware utilization of FM0 and Manchester encoders, the hardware architectures of both are conducted first. As mentioned earlier, the hardware architecture of Manchester encoding is as simple as a XOR operation. However, the conduction of hardware architecture for FM0 is not as simple as that of Manchester.FM0 encoding architecture can be constructed with the Finite stat machine (FSM) [6] . As shown in Figure 3, the FSM of FM0 code is classified into four states.

A state code is individually assigned to each state, and each state code consists of A and B as shown in Figure 3. Suppose the initial state is S1, and its state code is 11 for *A* and *B*, respectively. If the X is logic-0, the state-transition must follow both rules1

and 3. The only one next-state that can satisfy both rules for the X of logic-0 is S. If the X is logic-1, the state-transition must follow both rules 2 and 3. The only one next-state can satisfy both rules for the X of logic-1 is S4.Thus, the state-transition of each state can be completely constructed. According to the coding principle of FM0, the FSM of FM0 is shown in Figure 4.

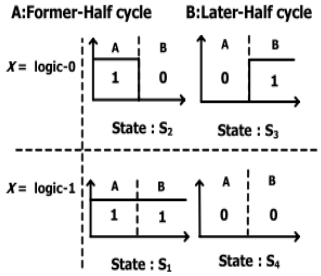


Fig. 3: Illustration of FSM for FM0 State Definition.

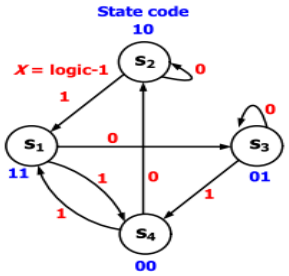


Fig. 4: FSM OF FM0.

The FSM of FM0 (Khan, M.A., 2009) can also conduct the transition table of each state, as shown in Table 1. From the Table, A(t) and B(t) represent the discrete-time state code of current-state at time instant(t).Their previous-states are denoted as the A(t - 1) and the B(t - 1) respectively. Previous state stores the values which are stored in the past.

Table 1: Transition Table For Fm0.

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

With this transition table, the Boolean functions of A (t) and B (t) are given as

$$A(t) = (t- 1) \quad (1)$$

$$B(t) = X \oplus B(t - 1) \quad (2)$$

With both A (t) and B (t), the Boolean function of FM0 code is denoted as

$$CLK\ A(t) + CLK\ B(t) \quad (3)$$

With (1) and (2), the hardware architectures of FM0 and Manchester encoders are shown in Figure 5. The top part is the hardware architecture of FM0 encoder and the bottom part is the hardware architecture of Manchester encoder.

As listed earlier, the Manchester encoder is as simple as a XOR operation for X and CLK. Nevertheless, the FM0 encoding depends not only on the X but also on the previous-state of the FM0 code. The DFF*A* and DFF*B* store the state code of the FM0code. The MUX1 is to switch A(t) and B(t) through the selection of CLK signal. Both A (t) and B(t) are realized by (2) and (3) respectively. The determination of which coding is adopted depends on the Mode selection of the MUX_2, where the Mode = 0 is for FM0 code and the Mode = 1 is for Manchester code. The component is defined as the hardware to perform a specific logic function, such as AND, OR, NOT, and flipflop.

Proposed Encoding Techniques:

In addition to FM0 and Manchester Encoding techniques, FM1 and Differential Manchester Encoding techniques is proposed in this paper. Both the encoding techniques performs different operations and it is used to improve the performance in terms of area and Power. The Proposed Encoding architecture not only improves area and Power but it also improves the hardware utilization rate.

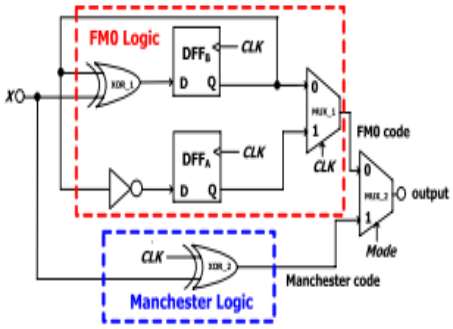
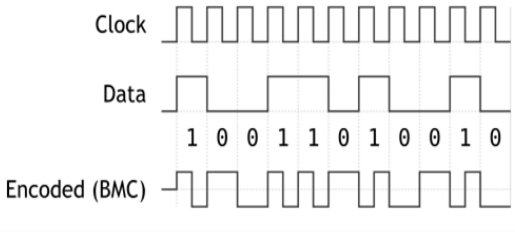


Fig. 5: Hardware architecture of FM0 and Manchester Encoding.

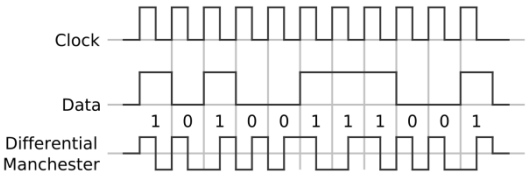
Principles of FM1 Encoding:



The FM1 Encoding performs inverse transition of FM0 Encoding. When Data =1, then half-cycle transition occurs and when Data=0, then full-cycle transition occurs.

Principles of Differential Manchester Encoding:

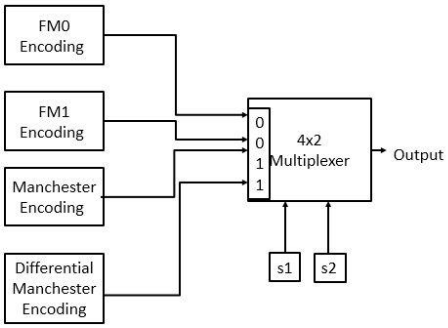
The Differential Manchester Encoding comes with some variations from Manchester Encoding.



There are two bits of Differential Manchester encoded data for each bit of original data. If the original data is a 0, the Differential Manchester code is 1 to 0, and it shifts to an upward transition at bit center. If Data is from 0 to 1 or 1 to 1 then transition is from previous half cycle.

Encoding architecture:

The Proposed Encoding architecture consists of four different encoding techniques i.e. FM0 Encoding, FM1 Encoding, Manchester Encoding and Differential Manchester Encoding. The architecture can be drawn as



The purpose of SOLS (Similarity oriented logic simplification) technique is to design a fully reused VLSI architecture for FM0 and Manchester encodings. The SOLS technique is classified into two parts: area-compact retiming and balance logic-operation sharing. Each part is individually described as follows.

Encoding Architecture Using Area-Compact Retiming:

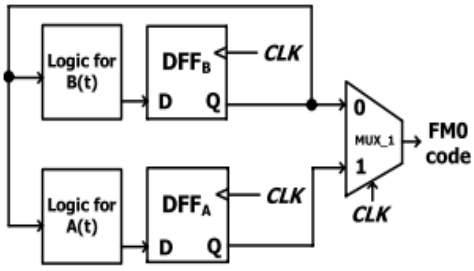


Fig. 6: FM0 Encoding without area-compact retiming.

The FM0 logic in Figure 1 is simply shown in Figure 6. The logic for A (t) and the logic for B (t) are the Boolean functions to derive A (t) and B(t),where the X is omitted for a concise representation. For FM0, the state code of each state is stored into DFFA and DFF B.

According to (2) and (3), the transition of state code only depends on B(t - 1) instead of both A(t - 1) and B(t - 1). Thus, the FM0 encoding just requires a single 1-bit flip-flop to store the B(t -1).

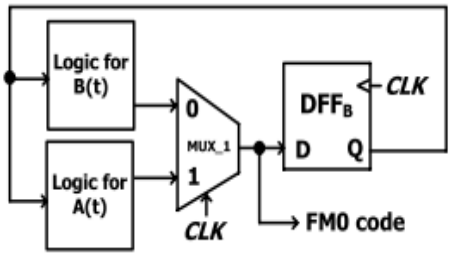


Fig. 7: FM0 Encoding with area-compact retiming.

If the DFF 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 DFF*A* is relocated right after the MUX1, as shown in Figure 7, where the DFF*B* is assumed to be positive-edge triggered. 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 MUX1 by the control signal of the CLK. In Figure 7, the *Q* of DFF*B* is directly updated from the logic of B(t) with 1-cycle latency. In Figure 7 when the CLK is logic-0, the B(t) is passed through MUX-1 to the *D* of DFF.

Encoding Architecture Using Balance Logic-Operation Sharing:

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

$$X \oplus \text{CLK} = X\,\text{CLK} + X\,\text{CLK} \quad (4)$$

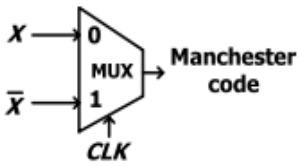


Fig. 8: Manchester Encoding in Multiplexer.

This can be realized by the multiplexer, as shown in Figure 8. It is quite similar to the Boolean function of FM0 encoding in (3). By comparing with (3) and (4), the FM0 and Manchester logics have a common point of the multiplexer like logic with the selection of CLK.

As shown in Figure 9, the concept of balance logic operation sharing is to integrate the X into A (t) and X into B(t) respectively. The logic for A(t)/X is showing Figure 10.The A (t) can be derived from an inverter of B(t - 1) and X is obtained by an inverter of X. The Mode indicates either FM0 or Manchester encoding is adopted.

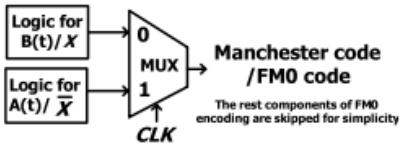


Fig. 9: Logic operations of Manchester and FM0 encodings.

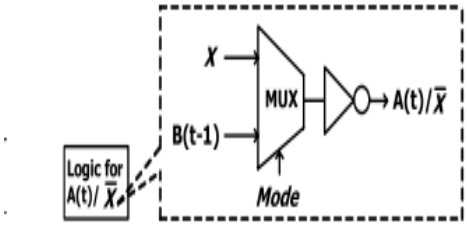


Fig. 10: Balance logic-operation sharing of A(t) and X.

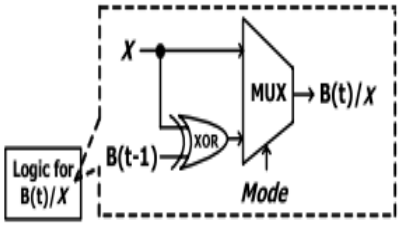


Fig. 11: Balance Logic-operation sharing of B(t)/x Without Xor sharing.

The similar concept can be also applied to the logic for B (t)/X, as shown in Figure 11. 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.

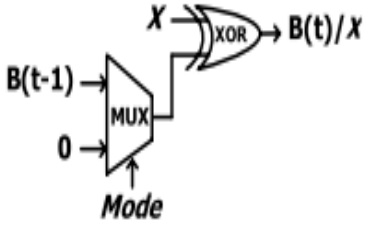


Fig. 12: Balance Logic-operation sharing of B(t)/x With Xor Sharing.

As a result, the logic for B(t)/X is shown in Figure 12, 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 thereby increases the HUR. Furthermore, the multiplexer in Figure 12 can be functionally integrated into the relocated DFF from area-compact retiming technique, as shown in Figure 13.

The CLR is the clear signal to reset the content of DFF to logic-0. The DFF 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 Figure 12 can be totally saved and its function can be completely integrated into the relocated DFF.

The proposed VLSI architecture of FM0/Manchester encoding using SOLS technique is shown in Figure 14. 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.

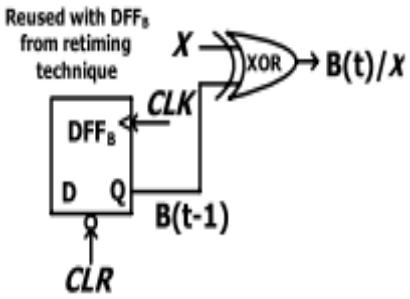


Fig. 13: Sharing of the reused DFFB from area-compact retiming technique.

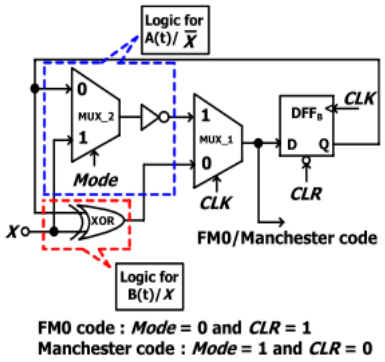


Fig. 14: Unbalance computation time between A(t)/x and B(t)/x.

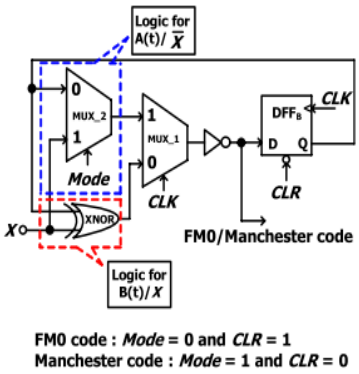


Fig. 15: Balance computation time between A(t)/x and B(t)/x.

To alleviate this unbalance computation time, the architecture of the balance computation time between A (t)/X and B(t)/ X is shown in Figure 15. 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. 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 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, Performance of the VLSI architecture is greatly improved.

Results:

Power consumption in FM0 and Manchester Encoding is calculated by two types. They are without area-compact retiming and with area-compact retiming.

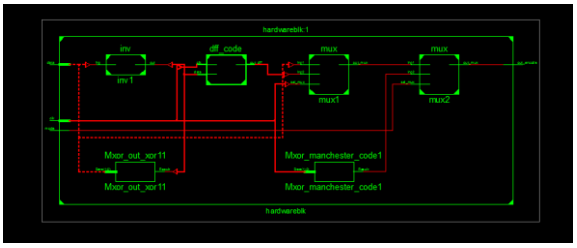


Fig. 15: RTL View of FM0 and Manchester Encoding.

Table 2: Power Consumption Comparsion.

	Without Area-compact retiming	With area-compact retiming
Power consumption	2280.72 nW	1223.26 nW
	Unbalance computation time	Balance computation time
	2362.02 nW	1533.98 nW

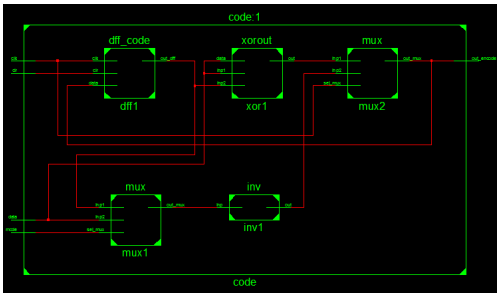


Fig. 16: RTL View of Unbalance computation time.

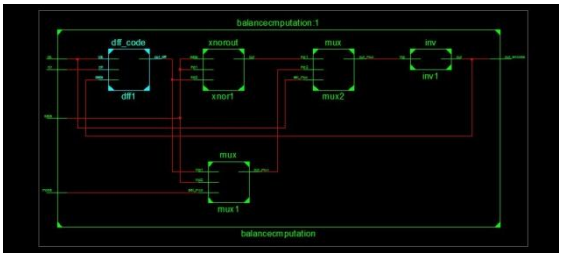


Fig. 17: RTL View of Balance Computation time.

Conclusion:

The coding-diversity between FM0 and Manchester encodings causes the limitation on designing VLSI architecture. The limitation analysis of FM0 and Manchester encodings is discussed in detail. Hence, the fully reused VLSI architecture using SOLS (Similarity oriented logic simplification) technique for both FM0 and Manchester encodings is adapted. The SOLS technique eliminates the limitation by two core techniques: area-compact retiming and balance logic-operation sharing. Area-compact retiming relocates the logic elements effectively. The Balance Logic operation sharing technique combines FM0 and Manchester encodings to produce balanced computation time. Results shows the improved hardware utilization rate through which power consumption is reduced. This project not only develops VLSI architecture for FM0 and Manchester Encoding but also improves the performance of both.

ACKNOWLEDGMENT

We would like to extend sincere appreciation to Easwari Engineering College and to all members of Research Group at ECE Department, Anna University, for all the supports and encouragement.

REFERENCES

Ahmed-Zaid, F., F. Bai, S. Bai, C. Basnayake, B. Bellur, S. Brovold, 2011. 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.

Benabes, P., A. Gauthier and J. Oksman, 2003. A Manchester code generator running at 1 GHz, in *Proc. IEEE, Int. Conf. Electron., Circuits Syst.*, 3: 1156–1159.

Daniel, J., V. Taliwal, A. Meier, W. Holfelder and R. Herrtwich, 2006. Design of 5.9 GHz DSRC-based vehicular safety communication.*IEEE Wireless Commun. Mag.*, 13(5): 36–43.

Hung, V., M.M. Kuo, C.K. Tung and S.H. Shieh 2009. High-speed CMOS chip design for Manchester and Miller encoder, in *Proc. Intell. Inf. Hiding Multimedia Signal Process.*, pp: 538–541.

Kenney, B., 2011. Dedicated short-range communications (DSRC) standards in the United States," *Proc. IEEE*, 99(7): 1162–1182.

Khan, M.A., M. Sharma and P.R. Brahmanandha, 2008. FSM based Manchester encoder for UHF RFID tag emulator.in *Proc. Int. Conf. Comput., Commun. Netw.*, 1–6.

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