ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Design of FM0 and Manchester Encoder and Decoder for DSRC Application Using SOLS Technique

G.Anudeep
M.Tech (VLSI),
CMR Institute of Technology,
Hyderabad, India.

Mohd.Shahbaz Khan
Associate Professor,
CMR Institute of Technology,
Hyderabad, India.

B.S.Priyanka Kumari
Assistant Professor,
CMR Institute of Technology,
Hyderabad, India.

Abstract

Dedicated short-range communication (DSRC) is used in intelligent transportation system. The DSRC standards generally adopt FM0 and Manchester encodes and decodes to reach dc-balance, enhancing the signal reliability but the coding-diversity between the FM0 and Manchester codes seriously limits the potential to design a fully reused VLSI architecture for both so the similarity-oriented logic simplification (SOLS) technique is proposed. Two blocks are added, serial to parallel data converter, parallel to serial data converter. The extension of this project is fmo and Manchester decoding. The SOLS technique improves the hardware utilization rate for both FM0 and Manchester encoding and decoding.

Index Terms— ***Dedicated short-range communication (DSRC), FM0, Manchester.***

I. INTRODUCTION

The dedicated short-range communication (DSRC) is used for small and medium range communication. The DSRC is classified into two types: automobile-to-automobile and automobile-to-roadside. In automobile-to-automobile, the DSRC makes the information sending and broadcasting among automobiles for safety purposes and public information announcement. The automobile-to-roadside mainly targets on the intelligent transportation service, such as electronic toll collection (ETC) system. The ETC can be forwarded to the payment for parking-service, and gas-refueling. Thus, the DSRC system plays an important role in modern vehicle industry.

II. ENCODING PRINCIPLES OF FM0 AND MANCHESTER.

The block diagram is divided in to two parts, transmission, receiving parts.

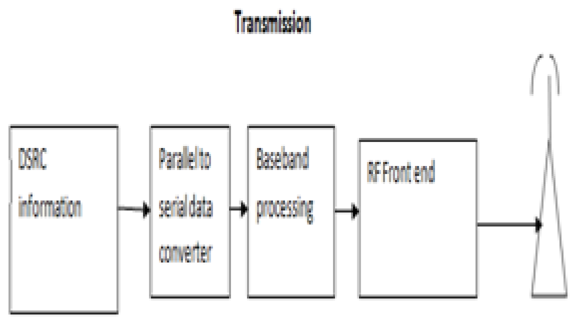


Fig 1: Transmission

Transmission has DSRC information, parallel to serial data converter, baseband processing and RF frontend.

The baseband processing is responsible for modulation, error correction, clock synchronization, and encoding. The RF frontend is used as amplifier it receives the wireless signal from antenna. Dsrc information is converted into serial data with parallel to serial data converter .we used a parallel to serial data converter for convert the data in to serial because fm0 and Manchester allows only serial data. The purposes of FM0 and Manchester codes can provide the transmitted signal with dc-balance. Both FM0 and Manchester codes are widely adopted in encoding and decoding for automobile industry.

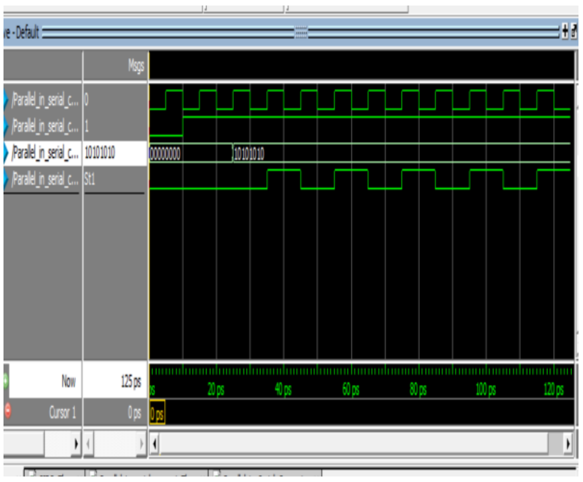


Fig 2. Output of parallel to serial data converter

Parallel to serial data converter converts the parallel data into serial and it is given to the baseband processing where encoding is performed.
The clock signal and the input data are abbreviated as CLK, and X, respectively.

A. FM0.

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 done between A and B.
3) The transition is allocated among each FM0 code no matter what the X is.

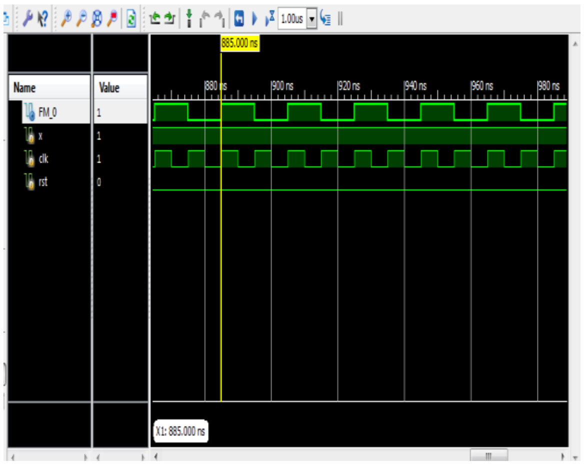


Fig 3. Fmo encoding

Fm0 depends on the input and clk. If the input is logic 0 it has transition between A and B and irrespective of input there should be transition between each fm0 code.

B. Manchester

The Manchester coding example is shown in Fig. 4. The Manchester code is derived from

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

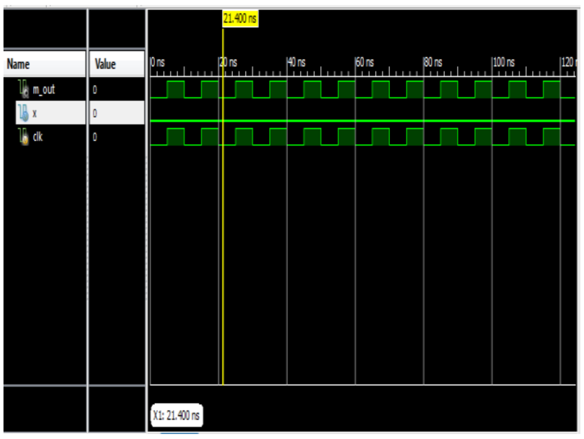


Fig 4. Encoding of Manchester

The Manchester encoding is xor operation between clock and input.. The clock always has a transition within one cycle, and so does the Manchester code no matter what the X .

The Manchester encoding is easy a XOR operation. However, the conduction of hardware architecture for FM0 is not as simple as that of Manchester. 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. According to the encoding principle of FM0. 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 rules 1 and 3.

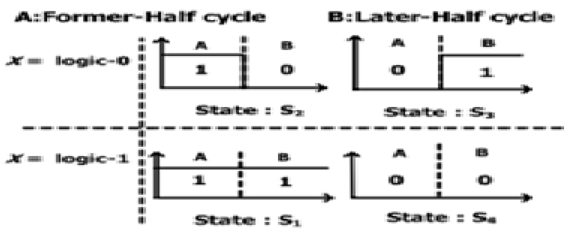


Fig 5: Illustration of FSM for FM0. (a) States definition.

The X of logic-0 is S_3.If the X is logic-1, the state-transition must follow both rules 2 and 3. The only one next-state is s4 .A(t) and B(t) represent the discrete-time state code of current-state at time instant t .The previous-states are denoted as the A(t-1) and the B(t -1). The Boolean functions of are given in below form

$$\begin{cases} A(t) = \overline{B(t-1)} & (2) \\ B(t) = X \oplus B(t-1). & (3) \end{cases}$$

FM0 code is denoted as

CLK A (t) +CLK B (t) (4)

With (1) and (4), the hardware architectures of FM0 and Manchester encoders are shown.

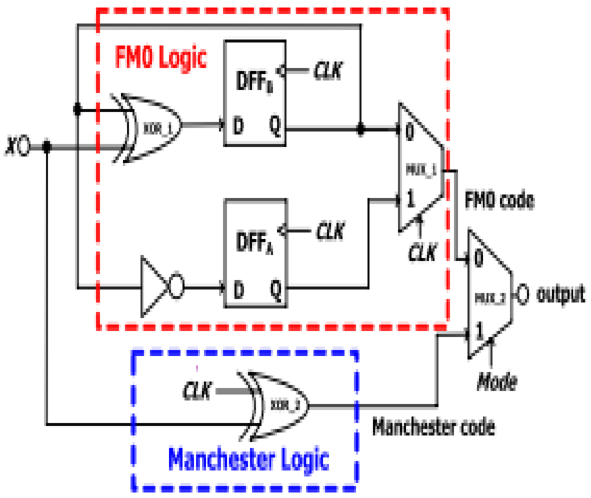


Fig.6. Hardware architecture of FM0 and Manchester encodings.

The top part is the hardware architecture of FM0 encoder, and the bottom part is the hardware architecture of Manchester encoder. The Manchester encoder is as simple as a XOR operation for X and CLK. But the FM0 encoding depends not only on the X but also on the previous-state of the FM0code. The DFFA and DFFB store the state code of the FM0code. The MUX-B1 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). 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. If mode is logic 0 fmo will be the output and if mode is logic 1 Manchester is the output. To evaluate the hardware utilization, the hardware utilization rate (HUR) is defined as

$$\text{HUR} = \frac{\text{Active components}}{\text{Total components}} \times 100\%.$$

The active component means the components that work for FM0 or Manchester encoding. The total components are the number of components in the entire hardware architecture no matter what encoding method is adopted. For FM0 encoding, the active components are 6, and its HUR is 85.71%. For Manchester encoding, the active components are two its HUR is as 28.57%. On average, this hardware architecture has a poor HUR of 57.14%. The coding-diversity between the FM0 and Manchester codes seriously limits the potential to design a fully reused VLSI architecture.

III. VLSI ARCHITECTURE DESIGN OF FM0 ENCODER AND MANCHESTER ENCODER USING SOLS TECHNIQUE

The purpose of SOLS technique is to design a fully reused VLSI architecture for FM0 and Manchester encoding and decoding. The SOLS technique is classified into two parts: area-compact retiming and balance logic-operation sharing.

A. Area-Compact Retiming

The logic for A(t) and the logic for B(t) are the Boolean functions to derive A(t) and B(t). For FM0, the state code of each state is stored into DFFA and DFFB. 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). 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 MUX1. 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

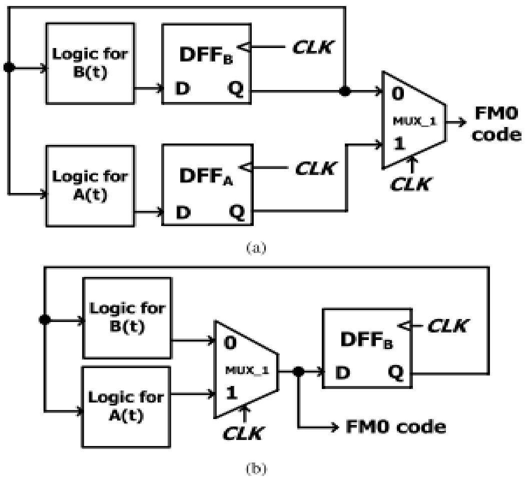


Fig. 7. Illustration of area-compact retiming on FM0 encoding architecture. (a) FM0 encoding without area-compact retiming. (b) FM0 encoding with area-compact retiming.

B. Balance Logic-Operation Sharing

The Manchester encoding can be derived from X ⊕ CLK, and it is also equivalent to

X ⊕ CLK = X CLK + X CLK. (6)

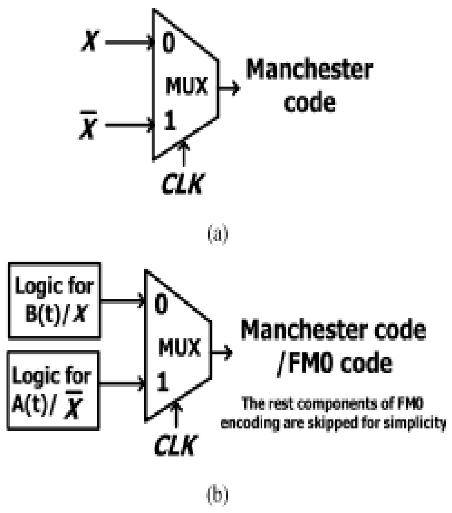

X
0
MUX
1
X̄
Manchester code
CLK
(a)
Logic for B(t)/X
Logic for A(t)/ X̄
0
MUX
1
CLK
Manchester code /FM0 code
The rest components of FM0 encoding are skipped for simplicity
(b)

Fig.8. Concept of balance logic-operation sharing for FM0 and Manchester encodings.

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. The concept of balance logic-operation sharing is to join the X into A (t) and X into B (t) 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.

This architecture exhibits imitation that the XOR is only dedicated for FM0 encoding, and is not shared with Manchester encoding. Therefore, the HUR of this is less. This shares the XOR for both B (t) and X, and thereby increases the HUR.

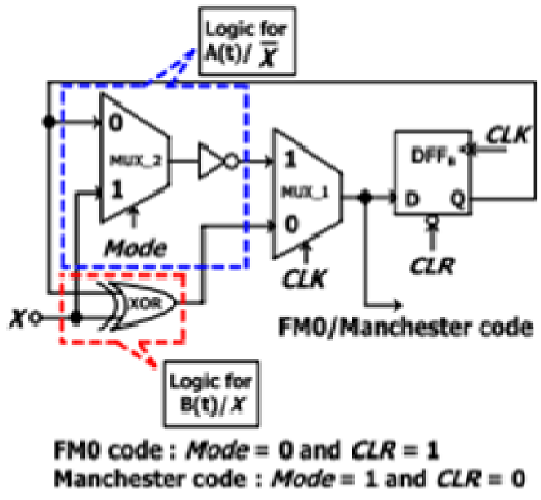


Fig. 9VLSI architecture of FM0 and Manchester encodings using SOLS technique. (a) Unbalance computation time between A (t)/X and B (t)/ X (b)

The CLR is the clear signal to reset the content of DFFB to logic-0. The DFF_B can be set to zero by activating CLR for Manchester encoding. When the FM0 code is adopted, the CLR is disabled. The proposed VLSI architecture of FM0/Manchester coding using SOLS technique The logic for A(t)/X includes the MUXB 2 and an inverter. Instead, the logic for B(t)/ X just incorporates a XOR gate.

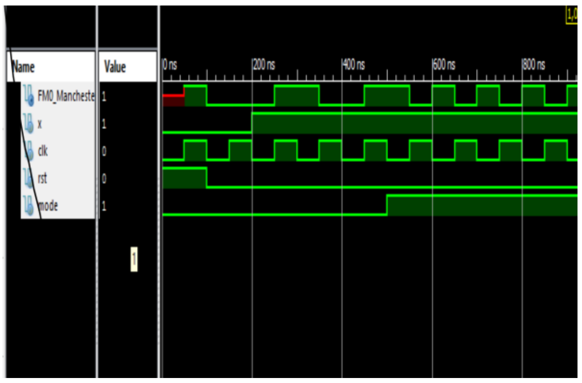


Fig.10 unbalance computation

From the figure to reduce the unbalance computation time which 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 time between A(t)/X and B(t)/ X .The XOR in the logic for B(t)/ X is changed to XNOR with an inverter, and it is shared with that of the logic for A(t)/ X. This shared inverter1 is relocated to the output of mux_1. Thus, the logic computation time between A (t)/X and B (t)/ X is balanced to each other.

The FM0 or Manchester code depends on Mode and CLR. To avoid the conflict between coding mode selection and hardware initialization CLK AND MODE are given individually..

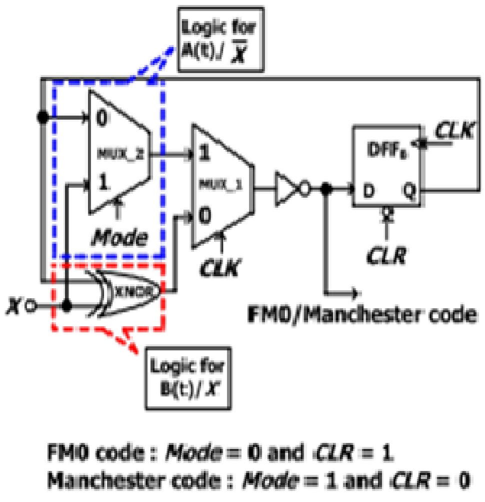


Fig.11. VLSI architecture of FM0 and Manchester encodings using SOLS technique Balance computation time between A(t)/X and B (t)/ X.

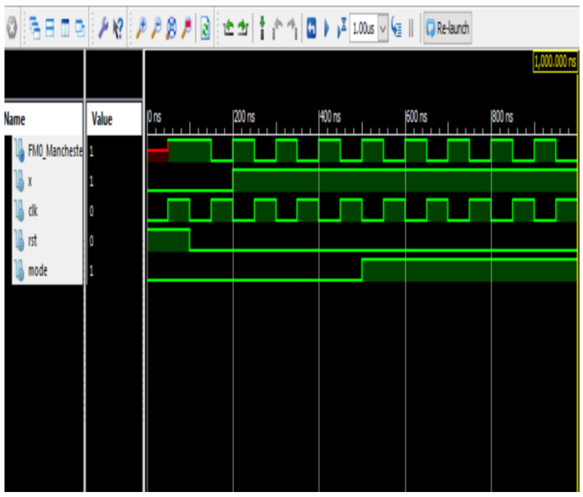


Fig 12 balance computation time between A(t)/X and B (t)/ X.

To alleviate this unbalance computation time, the architecture of the balance computation time between A(t)/X and B(t)/ X .The XOR in the logic for B(t)/ X is translated into the XNOR with an inverter.

Depending on the mode we can choose either fmo or Manchester, if mode is logic 0 and CLR is high it goes for fmo coding. , if mode is logic 1 and CLR is low it goes for Manchester coding. 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.

VI DECODER.

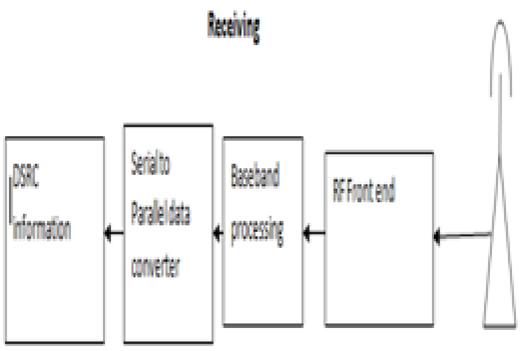
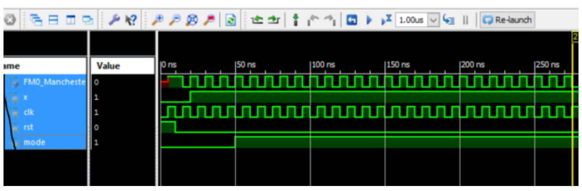


FIG 13: RECEIVING PART (DECODING)

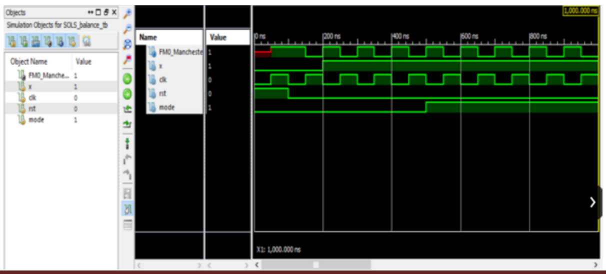
Decoders are simply a collection of logic gates which are arranged in a specific way so as to breakdown any combination of inputs. The other part of the project deals with the receiving part. It has baseband processor, RF front part, serial to parallel data converter and antenna and dsrc information. The fmo and Manchester decoding is used to convert the information to the original form from the antenna. Serial to parallel data converter is used to get the data in parallel .the output will be the dsrc information. The serial data is converted to parallel data with serial to parallel data converter which gives the dsrc information. The serial to parallel converter. The output of encoding and decoding is same which is obtained from parallel to serial converter. The output of baseband processing after decoding is given to serial to parallel data converter which gives the dsrc information. Fmo and Manchester decoding techniques are used to translate received message to actual message. The example taken here is 10101010 .it is received from the rf frontend and it undergoes decoding using mfsr technique and then it is given to the serial in parallel out to get the desired output, which is equal to the actual input .The received signal 10101010 is decoded using xor operation . xor operation is performed on input to get the unique result which gives the actual result.

SIMULATION RESULTS

ENCODER:



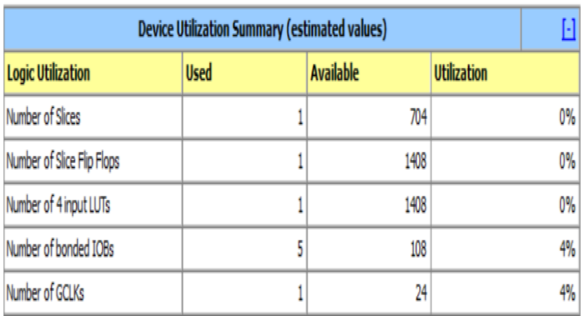
DECODER:



SYNTHESIS REPORT

ENCODER

Design summary:

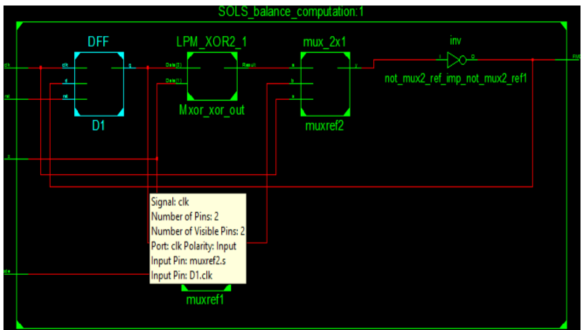


Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	1	704	0%
Number of Slice Flip Flops	1	1408	0%
Number of 4 input LUTs	1	1408	0%
Number of bonded IOBs	5	108	4%
Number of GCLKs	1	24	4%

Timing Report:

```
                    Gate    Net
Cell: in->out   fan out  Delay   Logical Name (Net Name)
-------------------------------------        |-----------
FDR:C->Q          1  0.591  0.423  D1/q (D1/q)
LUT4:I3->O        2  0.648  0.000  not_mux2_ref1 (FM0_Manchester_out_OBUF)
FDR:D              0.252       D1/q
-------------------------------------
Total              1.914ns (1.491ns logic, 0.423ns route)
```

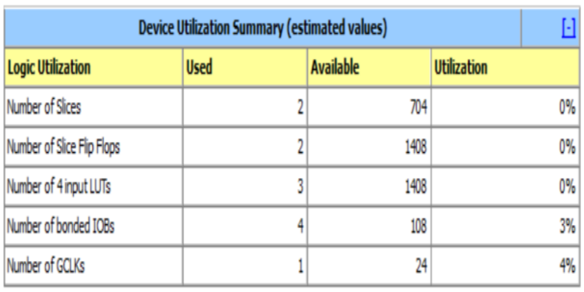
RTL schematic:



Signal: clk
Number of Pins: 2
Number of Visible Pins: 2
Port: clk Polarity: Input
Input Pin: muxref2.s
Input Pin: D1.clk

Decoder

Design summary:

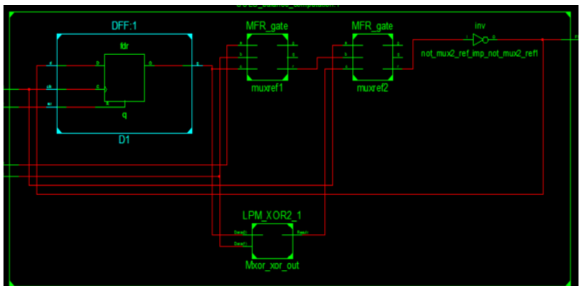


Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	2	704	0%
Number of Slice Flip Flops	2	1408	0%
Number of 4 input LUTs	3	1408	0%
Number of bonded IOBs	4	108	3%
Number of GCLKs	1	24	4%

Timing Report:

```
                         Gate     Net
Cell: in->out    fan out   Delay    Logical Name (Net Name)
----------------------------------------     ------------
 FDR:C->Q            2  1.122  0.423  D1/q (D1/q)
 LUT4:I3->O          2   1.648   0.000   not_mux2_ref1 (FM0_Manchester_out_OBUF)
----------------------------------------
Total                 2.770ns (1.491ns logic, 0.423ns route)
```

RTL schematic:



Encoder

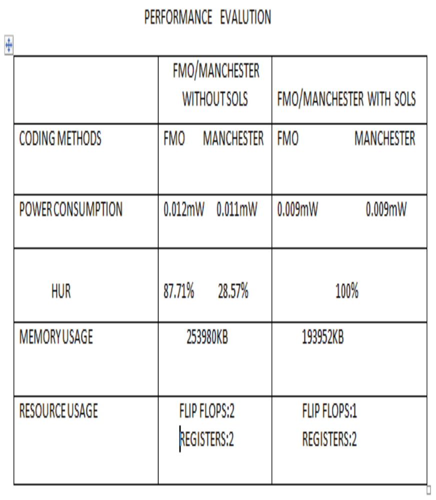
```
========================================================================
*                        Final Report                          *
========================================================================
Final Results
RTL Top Level Output File Name      : SOLS_balance_computation.ngr
Top Level Output File Name        : SOLS_balance _computation
Output Format                    : NGC
Optimization Goal                 : Speed
Keep Hierarchy                   : No
Design Statistics
# IOs                        : 5
Cell Usage:
# BELS                        : 1
#      LUT4                   : 1
# FlipFlops/Latches            : 1
#      FDR                    : 1
# Clock Buffers                : 1
#      BUFG                   : 1
# IO Buffers                   : 5
#      IBUF                   : 4
#      OBUF                   : 1
========================================================================
```

Decoder

```
========================================================================
*                        Final Report                          *
========================================================================
Final Results
RTL Top Level Output File Name      : SOLS_balance_computation_decoder.ngr
Top Level Output File Name        : SOLS_ balance _computation
Output Format                    : NGC
Optimization Goal                 : Speed
Keep Hierarchy                   : No
Design Statistics
# IOs                        : 6
Cell Usage:
# BELS                        : 1
#      LUT4                   : 2
# Flip Flops/Latches           : 1
#      FDR                    : 2
# Clock Buffers                : 1
#      BUFG                   : 1
# IO Buffers                   : 3
#      IBUF                   : 4
#      OBUF                   : 1
========================================================================
```

PERFORMANCE EVALUTION

It compares the power consumption, memory usage, HUR for fmo and Manchester encoding and decoding with sols technique and without sols technique.



PERFORMANCE EVALUTION

	FMO/MANCHESTER WITHOUT SOLS		FMO/MANCHESTER WITH SOLS	
CODING METHODS	FMO	MANCHESTER	FMO	MANCHESTER
POWER CONSUMPTION	0.012mW	0.011mW	0.009mW	0.009mW
HUR	87.71%	28.57%	100%	
MEMORY USAGE	253980KB		193952KB	
RESOURCE USAGE	FLIP FLOPS:2 REGISTERS:2		FLIP FLOPS:1 REGISTERS:2	

COMPARISON TABLES

	ENCODER WITHOUT PARALLEL TO SERIAL DATA CONVERTER	DECODER WITHOUT SERIAL TO PARALLEL DATA CONVERTER
Number of slices	2	2
No. Of slice flip-flops	1	2
No. Of 4 input LUTs	1	3
No. Of bonded IOB'S	3	4
No. Of GCLK'S	1	1

	ENCODER WITH PARALLEL TO SERIAL DATA CONVERTER	DECODER WITH SERIAL TO PARALLEL DATA CONVERTER
Number of slices	4	3
No. Of slice flip-flops	1	2
No. Of 4 input LUTs	1	2
No. Of bonded IOB'S	3	5
No. Of GCLK'S	2	1

VI. CONCLUSION

The coding-diversity between FM0 and Manchester encodings and decoding causes the limitation on hardware utilization of VLSI architecture design. The fully reused VLSI architecture using SOLS technique for both FM0 and Manchester encoding and decoding is proposed. The SOLS technique eliminates the limitation on hardware utilization by two core techniques: area- compact retiming and logic-operation sharing. The area-compact retiming relocates the hardware resource.

The balance logic-operation sharing efficiently joined FM0 and Manchester encoding and decoding with the identical logic components. It improves HUR rate.

REFERENCES

[1] F. Ahmed-Zaid, F. Bai, S. Bai, C. Basnayake, B. Bellur, S. Brovold,et al., “Vehicle safety communications—Applications (VSC-A) finalreport,” 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) standardsin 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..