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

KADAVALA JYOTHI
M.Tech Student
T.K.R college of Engineering and Technology

Mr. A.INDRA KUMAR
Associate Professor
T.K.R college of Engineering and Technology

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-µm IP6M CMOS technology. The maximum operation frequency is 2 GHz and 900 MHz for Manchester and FM 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 × 30.43 µm². 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.

I. INTRODUCTION

Dedicated short-range communications are one-way or two-way short-range to medium-range wireless communication channels specifically designed for automotive use and a corresponding set of protocols and standards.

The dedicated short-range communication (DSRC) [1] is a protocol for one- or two-way medium range communication especially for intelligent transportation systems. The DSRC can be briefly classified into two categories: automobile-to-automobile and automobile-to-roadside. In automobile-to-automobile, the DSRC enables the message sending and broadcasting among automobiles for safety issues and public information announcement [2], [3]. The safety issues include blind-spot, intersection warning, inter cars distance, and collision-alarm. The automobile-to-roadside focuses on the intelligent transportation service, such as electronic toll collection (ETC) system. With ETC, the toll collecting is electrically accomplished with the contactless IC-card platform. Moreover, the ETC can be extended to the payment for parking-service, and gas-refueling. The upper and bottom parts are dedicated for transmission and receiving, respectively. This transceiver is classified into three basic modules: microprocessor, baseband processing, and RF front-end. The microprocessor interprets instructions from media access control to schedule the tasks of baseband processing and RF front-end. The baseband processing is responsible for modulation, error correction, clock synchronization, and encoding. The RF frontend transmits and receives the wireless signal through the antenna.

Communications-based active safety applications use vehicle to vehicle and vehicle to infrastructure short-range wireless communications to detect potential hazards in a vehicle’s path – even those the driver does not see. The connected vehicle provides enhanced awareness at potentially reduced cost, and offers additional functionality over autonomous sensor systems available on some vehicles today. Communications-based sensor systems could potentially be a low-cost means of enabling hazard detection capability on all vehicle classes, but requires vehicles and infrastructure to be outfitted with interoperable communications capabilities.

II. DATA ENCODING

Digital data to analog signals

A modem (modulator-demodulator) converts digital data to analog signal. There are 3 ways to modulate a digital signal on an analog carrier signal.

Amplitude shift keying (ASK): is a form of modulation which represents digital data as variations in the amplitude of a carrier wave. Two different amplitudes of carrier frequency represent ‘0’ , ‘1’.
Frequency shift keying (FSK): In Frequency Shift Keying, the change in frequency define different digits. Two different frequencies near carrier frequency represent '0', '1'.

Phase shift keying (PSK): The phase of the carrier is discretely varied in relation either to a reference phase or to the phase of the immediately preceding signal element, in accordance with data being transmitted. Phase of carrier signal is shifted to represent '0', '1'.

Digital data to digital signals
A digital signal is sequence of discrete, discontinuous voltage pulses. Each pulses a signal element. Encoding scheme is an important factor in how successfully the receiver interprets the incoming signal.

Encoding Techniques

Following are several ways to map data bits to signal elements.

Non return to zero (NRZ) NRZ codes share the property that voltage level is constant during a bit interval. High level voltage = bit 1 and Low level voltage = bit 0. A problem arises when there is a long sequence of 0's or 1's and the volatage level is maintained at the same value for a long time. This creates a problem on the receiving end because now, the clock synchronization is lost due to lack of any transitions and hence, it is difficult to determine the exact number of 0s or 1s in this sequence.

The two variations are as follows:

NRZ-Level: In NRZ-L encoding, the polarity of the signal changes only when the incoming signal changes from a 1 to a 0 or from a 0 to a 1. NRZ-L method looks just like the NRZ method, except for the first input one data bit. This is because NRZ does not consider the first data bit to be a polarity change, where NRZ-L does.

NRZ-Inverted: Transition at the beginning of bit interval = bit 1 and No Transition at beginning of bit interval = bit 0 or viceversa. This technique is known as differential encoding.

NRZ-I has an advantage over NRZ-L. Consider the situation when two data wires are wrongly connected in each other's place. In NRZ-L all bit sequences will get reversed (B'coz voltage levels get swapped). Whereas in NRZ-I since bits are recognized by transition the bits will be correctly interpreted. A disadvantage in NRZ codes is that a string of 0's or 1's will prevent synchronization of transmitter clock with receiver clock and a separate clock line need to be provided.

Biphase encoding: It has following characteristics:

1. Modulation rate twice that of NRZ and bandwidth correspondingly greater. (Modulation is the rate at which signal level is changed).
2. Because there is predictable transition during each bit time, the receiver can synchronize on that transition i.e. clock is extracted from the signal itself.
3. Since there can be transition at the beginning as well as in the middle of the bit interval the clock operates at twice the data transfer rate.

Types of Encoding -->

- **Biphase-manchester**: Transition from high to low in middle of interval = 1 and Transition from low to high in middle of interval = 0
- **Differential-manchester**: Always a transition in middle of interval. No transition at beginning of interval = 1 and Transition at beginning of interval = 0
III. VLSI ARCHITECTURE DESIGN OF FM0 ENCODER AND MANCHESTER ENCODER USING SOLS TECHNIQUE

The FM0 logic in Fig. is simply shown in Fig.. 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

![Diagram of state codes](image1)

**TABLE IV**

<table>
<thead>
<tr>
<th>Transistor Count of FM0 Encoding Architecture with Area-Compact Retiming</th>
<th>Without area-compact retiming</th>
<th>With area-compact retiming</th>
</tr>
</thead>
<tbody>
<tr>
<td>PMOS</td>
<td>36</td>
<td>25</td>
</tr>
<tr>
<td>NMOS</td>
<td>36</td>
<td>25</td>
</tr>
<tr>
<td>Total</td>
<td>72</td>
<td>50</td>
</tr>
</tbody>
</table>

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 \(B(t-1)\). Thus, the FM0 encoding just requires a single 1-bit flip-flop to store the \(B(t-1)\). 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, as shown in Fig. 7(b), where the DFFB is assumed to be positive-edge triggered. At each cycle, the FM0 code, comprising \(A(t)\) and \(B(t)\), 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 Fig. 7(a), the \(Q\) of DFFB is directly updated from the logic of \(B(t)\) with 1-cycle latency.

![Diagram of FM0 encoding architecture](image2)

In Fig. 7(b), 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. As shown in Fig. 8, the timing diagram for the \(Q\) of DFFB is consistent whether the DFFB is relocated or not. Suppose the logic components of FM0 encoder are realized with the logic-family of static CMOS, and the total transistor count is shown in Table IV. 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.

IV. BALANCE LOGIC-OPERATION SHARING

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

\[X \oplus CLK = X CLK + X CLK\] (6)

This can be realized by the multiplexer, as shown in Fig. 9(a).

![Diagram of Manchester encoding](image3)

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. 9(b), 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 shown in Fig. 10.
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. 11(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. 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. can be functionally integrated into the relocated DFFB from area-compact retiming technique, as shown in Fig. 11(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. 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. 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. 12(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 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.

V. 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-μm 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 × 30.43 μm2. 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.

VI. REFERENCES


