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

Yu-Hsuan Lee, Member, IEEE, and Cheng-Wei Pan

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 postlayout simulation in Taiwan Semiconductor Manufacturing Company (TSMC) 0.18-µm 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 x 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.

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

I. INTRODUCTION

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, intercars distance, and collision-alarm. The automobile-to-roadside

Fig. 1. System architecture of DSRC transceiver.

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. Thus, the DSRC system plays an important role in modern automobile industry. The system architecture of DSRC transceiver is shown in Fig. 1. 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 front-end transmits and receives the wireless signal through the antenna.

The DSRC standards have been established by several organizations in different countries. These DSRC standards of America, Europe, and Japan are shown in Table I. The data rate individually targets at 500 kb/s, 4 Mb/s, and 27 Mb/s with carrier frequency of 5.8 and 5.9 GHz. The modulation methods incorporate amplitude shift keying, phase shift keying, and orthogonal frequency division multiplexing. 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. 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 for downlink. The VLSI architectures of FM0 and Manchester encoders are reviewed as follows.

A. Review of VLSI Architectures for FM0 Encoder and Manchester Encoder

The literature [4] proposes a VLSI architecture of Manchester encoder for optical communications. This design adopts the CMOS inverter and the gated inverter as the switch to construct Manchester encoder. It is implemented by 0.35-μm CMOS technology and its operation frequency is 1 GHz. The literature [5] further replaces the architecture of switch in [4] by the nMOS device. It is realized in 90-nm CMOS technology, and the maximum operation frequency is as high as 5 GHz. The literature [6] develops a high-speed VLSI architecture almost fully reused with Manchester and Miller encodings for radio frequency identification (RFID) applications. This design is realized in 0.35-μm CMOS technology and the maximum operation frequency is 200 MHz. The literature [7] also proposes a Manchester encoding architecture for ultrahigh frequency (UHF) RFID tag emulator. This hardware architecture is conducted from the finite state machine (FSM) of Manchester code, and is realized into field-programmable gate array (FPGA) prototyping system. The maximum operation frequency of this design is about 256 MHz. The similar design methodology is further applied to individually construct FM0 and Miller encoders also for UHF RFID Tag emulator [8]. Its maximum operation frequency is about 192 MHz. Furthermore, [9] combines frequency shift keying (FSK) modulation and demodulation with Manchester codec in hardware realization.

B. Features of This Paper

However, the coding-diversity between both seriously limits the potential to design a VLSI architecture that can be fully reused with each other. This paper 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 22 transistors. The balance logic-operation sharing efficiently combines FM0 and Manchester encodings with the fully reused hardware architecture. With SOLS technique, this paper 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.

C. Organization

The remainder of this paper is organized as follows. Section II describes the coding principles of FM0 and Manchester codes. Section III gives a limitation analysis on hardware utilization of FM0 and Manchester encoders. This section shows the difficulty to design a fully reused VLSI architecture for FM0 and Manchester encoders. The proposed VLSI architecture design using SOLS technique is reported in Section IV. Two core methods of SOLS technique, area-compact retiming and balance logic-operation sharing, are described in this section. The experiment results and discussion are presented in Section V. This section focuses on an objective evaluation between this design and existing articles of Manchester and FM0 encoders. Finally, the conclusion is given in Section VI.

II. CODING PRINCIPLES OF FM0 CODE AND MANCHESTER CODE

In the following discussion, the clock signal and the input data are abbreviated as CLK, and X, respectively. With the above parameters, the coding principles of FM0 and Manchester codes are discussed as follows.

A. FM0 Encoding

As shown in Fig. 2, for each X, the FM0 code consists of two parts: one for former-half cycle of CLK, A, and the other one for later-half cycle of CLK, B. 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 FM0 code no matter what the X is.

A FM0 coding example is shown in Fig. 3. 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.

### B. Manchester Encoding

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

$$X \oplus CLK.$$  \hspace{1cm} (1)

The Manchester encoding is realized with a XOR operation for CLK and $X$. The clock always has a transition within one cycle, and so does the Manchester code no matter what the $X$ is.

### III. LIMITATION ANALYSIS ON HARDWARE UTILIZATION 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. How to construct the hardware architecture of FM0 encoding should start with the FSM of FM0 first. As shown in Fig. 5(a), 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 Fig. 2. According to the coding principle of FM0, the FSM of FM0 is shown in Fig. 5(b). Suppose the initial state is $S_1$, 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. The only one next-state that can satisfy both rules for 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 that can satisfy both rules for the $X$ of logic-1 is $S_4$. Thus, the state-transition of each state can be completely constructed.

The FSM of FM0 can also conduct the transition table of each state, as shown in Table II. $A(t)$ 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. With this transition table, the Boolean functions

<table>
<thead>
<tr>
<th>Table II</th>
<th>Transition Table of FM0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previous-state</td>
<td>Current-state</td>
</tr>
<tr>
<td>$A(t-1)$</td>
<td>$B(t-1)$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table III

HUR of FM0 and Manchester Encodings

<table>
<thead>
<tr>
<th>Coding</th>
<th>Active components (transistor count)</th>
<th>HUR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>/ Total components (transistor count)</td>
<td></td>
</tr>
<tr>
<td>FM0</td>
<td>6 (86) / 7 (98)</td>
<td>85.71%</td>
</tr>
<tr>
<td>Manchester</td>
<td>2 (26) / 7 (98)</td>
<td>28.57%</td>
</tr>
<tr>
<td>Average</td>
<td>4 (56) / 7 (98)</td>
<td>57.14%</td>
</tr>
</tbody>
</table>

of $A(t)$ and $B(t)$ are given as

$$A(t) = B(t-1)$$

$B(t) = X \oplus B(t-1)$.

(3)

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

$$CLK \ A(t) + \overline{CLK} \ B(t).$$

(4)

With (1) and (4), the hardware architectures of FM0 and Manchester encoders are shown in Fig. 6. The top part is the hardware architecture of FM0 encoder, and the bottom part is the hardware architecture of Manchester encoder. As listed in (1), 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 FM0 code. The MUX..1 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. To evaluate the hardware utilization, the hardware utilization rate (HUR) is defined as

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

(5)

The component is defined as the hardware to perform a specific logic function, such as AND, OR, NOT, and flip-flop. The active components mean 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. The HUR of FM0 and Manchester encodings is listed in Table III. For both encoding methods, the total components are 7, including MUX..2 to indicate which coding method is activated. For FM0 encoding, the active components are 6, and its HUR is 85.71%. For Manchester encoding, the active components are 2, comprising XOR..2 and MUX..2, and its HUR is as low as 28.57%. On average, this hardware architecture has a poor HUR of 57.14%, and almost half of total components are wasted.

The transistor count of the hardware architecture without SOLS technique is 98, where 86 transistors are for FM0 encoding and 26 transistors are for Manchester coding. On average, only 56 transistors can be reused, and this is consistent with its HUR.

The coding-diversity between the FM0 and Manchester codes seriously limits the potential to design a fully reused VLSI architecture.

IV. 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 encodings. The SOLS technique is classified into two parts: area-compact retiming and balance logic-operation sharing. Each part is individually described as follows. Finally, the performance evaluation of the SOLS technique is given.

A. Area-Compact Retiming

The FM0 logic in Fig. 6 is simply shown in Fig. 7(a). 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 $DFF_A$ 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 $B(t-1)$. If the $DFF_A$ is directly removed, a nonsynchronization between $A(t)$ and $B(t)$ causes the logic-fault of FM0 code. To avoid this logic-fault, the $DFF_B$ is relocated right after the MUX.. As shown in Fig. 7(b), 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 MUX.. The $Q$ of $DFF_B$ is directly updated from the logic of $B(t)$ with 1-cycle latency. In Fig. 7(b), when the CLK is logic-0, the $B(t)$ is passed through MUX.. Then, the upcoming positive-edge of CLK updates it to the $Q$ of $DFF_B$. As shown in Fig. 8, the timing diagram for the $Q$ of $DFF_B$ is consistent whether the $DFF_B$ 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.

**TABLE IV**

<table>
<thead>
<tr>
<th></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>

**B. Balance Logic-Operation Sharing**

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

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

This can be realized by the multiplexer, as shown in Fig. 9(a). 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 $A(t)/\overline{X}$ is shown in Fig. 10. The $A(t)$ can be derived from an inverter of $B(t-1)$, and $\overline{X}$ is obtained by an inverter of $X$. The logic for $A(t)/\overline{X}$ 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. 11(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. 11(b)
Fig. 11. Balance logic-operation sharing of $B(t)$ and $X$. (a) Without the XOR sharing. (b) With XOR sharing. (c) Sharing of the reused DFF$_B$ from area-compact retiming technique.

The CLR is the clear signal to reset the content of DFF$_B$ 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, and the $B(t-1)$ can be derived from DFF$_B$. Hence, the multiplexer in Fig. 11(b) can be totally saved, and its function can be completely integrated into the relocated DFF$_B$. The proposed VLSI architecture of FM0/Manchester encoding using SOLS technique is shown in Fig. 12(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 $A(t)/\overline{X}$ 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 XOR with an inverter, and then this inverter is shared with that of the logic for $A(t)/\overline{X}$. 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 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.

C. Timing Analysis

The logic functions of SOLS technique can be realized by various logic families. Each logic family optimizes one or more electrical performance, such as area, power, or speed, from circuit topology perspective instead of architecture perspective. The proposed SOLS technique is developed from architecture perspective to achieve 100% HUR. Among the logic families, both static CMOS circuit and transmission-gate logic are widely applied in digital circuit owing to
their superior integration in process manufacturing. Hence, the timing analysis is given under these two kinds of logic families for a more general purpose. Although the SOLS technique enables the VLSI architecture to be fully shared for Manchester and FM0 encoders, their critical paths are not identical. For Manchester encoding, the delay time is given as

\[ T_{\text{Man}} = \max\{T_{\text{MUX}}, T_{\text{XNOR}}\} + T_{\text{MUX}} + T_{\text{INV}} \]  

where \( T_{\text{Man}} \) denotes the delay time of Manchester encoding. The \( T_{\text{MUX}}, T_{\text{XNOR}}, \) and \( T_{\text{INV}} \) represent the delay time of the multiplexer, the XNOR gate, and the inverter, respectively. The DFF\(_B\) is always kept at logic-0 in Manchester encoding; therefore, it is excluded from \( T_{\text{Man}} \). This delay path is also incorporated into that of FM0 encoding. Moreover, the FM0 encoding applies the DFF\(_B\) to store the state code, and thereby the delay time of DFF\(_B\) is further considered as

\[ T_{\text{FM0}} = T_{\text{Man}} + T_{\text{DFF}} \]  

where \( T_{\text{FM0}} \) is the delay time of FM0 encoding, and the \( T_{\text{DFF}} \) stands for the delay time of DFF\(_B\). The delay time of Manchester encoding is smaller than that of FM0 encoding. Hence, the logic components inside \( T_{\text{Man}} \) should be carefully considered in their implementation. If these logic components are totally designed with static CMOS, the \( T_{\text{Man}} \) is seriously limited owing to too many transistors in the critical path of Manchester encoding. More detail on this part is described as follows.

For simplicity, both rise and fall times of static CMOS circuit are assumed to be identical. The time period is adopted to denote the propagation delay with Elmore delay estimation. The static CMOS topologies of two-input multiplexer and two-input XNOR are shown in Fig. 13(a) and (b), respectively. The pull-down network of two-input multiplexer includes \( M_1, M_2, M_3, \) and \( M_4 \). The \( M_1 \) and \( M_2 \) are in parallel, and so are \( M_3 \) and \( M_4 \). Indeed, this connection can improve the discharging capability. However, the transistor sizing still considers the worst case where the discharging path is constructed by only two transistors in series. The propagation delay of the static CMOS two-input multiplexer is given as

\[ T_{\text{MUX}} - \text{SC} = T_{\text{INV}} + C_A R + C_B 2R \]  

(9)

where the second and third terms are predicted by Elmore delay estimation. The \( R \) stands for the equivalent resistor of each transistor in pull-down network. The \( C_A \) aggregates the junction capacitances of \( M_1, M_2, M_3, \) and \( M_4 \). The \( C_B \) gathers the junction capacitances of \( M_1, M_3, M_2, \) and \( M_4 \). Similarly, the propagation delay of static CMOS two-input XNOR is given as

\[ T_{\text{XNOR}} - \text{SC} = T_{\text{INV}} + C_C R + C_D 2R. \]  

(10)

Compared with (9), the \( C_A \) is greater than \( C_C \) since the \( C_A \) incorporates more junction area than \( C_C \), and the \( C_B \) approximates to \( C_D \). Thus, the \( T_{\text{MUX}} - \text{SC} \) is greater than \( T_{\text{XNOR}} - \text{SC} \).

In Fig. 12(b), for static CMOS, the series of MUX...1 and MUX...2 dominates Manchester encoding path and leads to a total propagation delay as

\[ 2T_{\text{MUX}} - \text{SC} = 2(T_{\text{INV}} + C_A R + C_B 2R). \]  

(11)

To further reduce the transistor count in Manchester encoding path, the transmission-gate logic is considered in the circuit designs of MUX...1, MUX...2 and XNOR. The transmission gate logic of two-input multiplexer and two-input XNOR are shown in Fig. 14(a) and (b), respectively.

The propagation delay of transmission-gate two-input multiplexer is given as

\[ T_{\text{MUX}} - \text{TG} = R'C_E \]  

(12)

where the \( C_E \) aggregates the junction capacitances of \( M_1, M_2, M_3, \) and \( M_4 \). The \( R' \) stands for the equivalent resistances of \( M_1 \) and \( M_2 \) in parallel, or \( M_3 \) and \( M_4 \) in parallel. Similarly, the propagation delay of transmission-gate two-input XNOR is given as

\[ T_{\text{XNOR}} - \text{TG} = T_{\text{INV}} + R'C_F \]  

(13)

where the \( C_F \) gathers the junction capacitances of \( M_1, M_2, M_3, \) and \( M_4 \). The \( T_{\text{XNOR}} - \text{TG} \) causes slightly...
longer propagation delay than $T_{\text{MUX-TG}}$. In Fig. 12(b), for transmission-gate logic, the series of XNOR and MUX\_2 dominates Manchester encoding path and leads to a total propagation delay as

$$T_{\text{XNOR-TG}} + T_{\text{MUX-TG}} = T_{\text{INV}} + R' C_E + R' C_F. \quad (14)$$

For a concise comparison with (11), the assumptions are given as follows. The $C_B, C_D, C_E,$ and $C_F$ consistently incorporate the junction capacitances of two nMOS and two pMOS. All these four capacitors can be assumed as $C_B = C_D = C_E = C_F$. For static CMOS circuit, both two-input multiplexer and two-input XNOR have two nMOS in series. The width of these two nMOS in series is enlarged by two times to achieve a balance fall-time, and the $R_N$ is reduced down to $R_N/2$. A transmission gate consisting of a pMOS and an nMOS in parallel has its equivalent resistor of $R_P//R_N$. To obtain a balance signal path, $R_P$ is approximated to $R_N$ by enlarging the width of pMOS by two to three times depending on the process technology. Hence, the equivalent resistor of a transmission gate is $R_P//R_N \simeq R_N//R_N = R_N/2$. The equivalent resistor of an nMOS in two-input multiplexer or two-input XNOR is identical to that of a transmission gate. Thus, the assumption is given as $R = R' = R_N/2$. With above assumptions, the propagation delay of transmission-gate logic is less than that of static CMOS. Applying the transmission-gate logic can compact the transistor count to reduce the propagation delay. Generally, the buffer is inserted every three-stage of transmission-gate logic [10], and this rule is also applied in the algorithm of buffer-insertion [11]. In Fig. 12(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.

### D. Performance Evaluation of the SOLS Technique

The evaluation of the SOLS technique is shown in Table V. With SOLS technique, the total components are reduced from seven down to five. Without SOLS technique, the FM0 and Manchester encodings are performed on individual hardware architecture with a poor HUR of 57.14%, as previously shown in Table III. With SOLS technique, the total transistor count...
TABLE VII
PERFORMANCE PROFILE

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding methods</td>
<td>Manchester</td>
<td>Manchester</td>
<td>Manchester</td>
<td>FM0</td>
<td>Manchester</td>
</tr>
<tr>
<td>Realization</td>
<td>0.35 um CMOS</td>
<td>90 nm CMOS</td>
<td>Xilinx</td>
<td>Xilinx</td>
<td>Xilinx</td>
</tr>
<tr>
<td>Supply voltage</td>
<td>3.3 V</td>
<td>1.2 V</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Technique</td>
<td>Gated inverter</td>
<td>CMOS switch</td>
<td>FSM based design</td>
<td>FSM based design</td>
<td>Joint MODEM/CODC</td>
</tr>
<tr>
<td>HUR</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Operation frequency</td>
<td>1 GHz [1.94 GHz]</td>
<td>5 GHz [2.5 GHz]</td>
<td>256.54 MHz</td>
<td>192.64 MHz</td>
<td>612 MHz</td>
</tr>
<tr>
<td>Power consumption</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>34 mW</td>
</tr>
<tr>
<td>Area</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Transistor count</td>
<td>54</td>
<td>26</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>FPGA resource usage</td>
<td>N/A</td>
<td>N/A</td>
<td>Slice : 1</td>
<td>Slice : 5</td>
<td>Slice : 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Flip-Flop : 2</td>
<td>Flip-Flop : 4</td>
<td>Flip-Flop : 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>LUTs : 2</td>
<td>LUTs : 10</td>
<td>LUTs : 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Bonded I/OBs : 3</td>
<td>Bonded I/OBs : 4</td>
<td>Bonded I/OBs : 3</td>
</tr>
</tbody>
</table>

* Normalized to 0.18 um CMOS process with (15) and (16).

Fig. 16. Power profiling of this paper for FM0 and Manchester encodings.

is reduced from 98 to 44, and every transistor is fully reused in either FM0 or Manchester encoding. The SOLS technique eliminates this limitation on HUR 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. The SOLS technique improves the HUR from 57.14% to 100%, whether the FM0 or Manchester code is adopted. Thus, the SOLS technique provides a fully reused VLSI architecture for FM0 and Manchester encodings with the HUR of 100%.

V. EXPERIMENT RESULTS AND DISCUSSION

A. Experiment Environment

This paper is compared with the existing articles. These articles are implemented in two kinds of design-flows. The literatures [4] and [5] are realized with full-custom and the literatures [7]–[9] are designed with FPGA. To give an objective evaluation, the proposed VLSI architecture is realized with both design-flows, as listed in Table VI. The design flow of the full-custom is Taiwan Semiconductor Manufacturing Company (TSMC) 0.18-μm 1P6M CMOS technology, and the Xilinx development board is adopted for the FPGA design-flow. The performance of full-custom design flow is from postlayout simulation, and its layout view is shown in Fig. 15. The performances of [4], [5], and [7]–[9] are listed in Table VII. Note that [6] is excluded since it further involves Miller encoding, which is out of the scope of this performance evaluation. The CMOS technologies of these works include 0.35 μm, 90 nm, which are not identical to 0.18 μm adopted in this paper.

To normalize the process parameters due to different CMOS technologies, the CMOS process scaling of the constant field theorem is applied. The normalization equations are given as follows [12]:

\[
\begin{align*}
    f_{0.18 \mu m} &= f_{0.35 \mu m/90 \mu m} \times \alpha \\
    P_{0.18 \mu m} &= P_{0.35 \mu m/90 \mu m} \times \frac{1}{\alpha} \\
    A_{0.18 \mu m} &= A_{0.35 \mu m/90 \mu m} \times \frac{1}{\alpha^2}
\end{align*}
\]

where the \( f_{0.18 \mu m} \), \( P_{0.18 \mu m} \), and \( A_{0.18 \mu m} \) represent the normalized operation frequency, power, and area for 0.18-μm CMOS process. The \( \alpha \) is a scaling factor defined as

\[
\alpha = \frac{0.35 \mu m}{0.18 \mu m} \quad \text{from 0.35 \mu m to 0.18 \mu m}
\]

\[
\alpha = \frac{0.18 \mu m}{90 \mu m} \quad \text{from 90 \mu m to 0.18 \mu m}
\]

B. Comparison With Sophisticated Articles

This paper adopts the proposed SOLS technique to construct a fully reused VLSI architecture for both FM0 and Manchester
encodings. Every logic component of this design is not only used in FM0 encoding, but also in Manchester encoding. None of them is wasted in either encoding function; therefore, the HUR of the proposed VLSI architecture is as high as 100%. The performance evaluation classifies the electrical characteristics into the operation frequency, the power consumption, and the area. They are all normalized to 0.18 \( \mu \text{m} \) for an objective evaluation by (15).

For Manchester encoding, the operation frequency of this paper is comparable with that of [4], but slower than that of [5]. The literature [5] is dedicated to optimize the signal path by reducing the number of transistors, and thereby has faster operation frequency. Instead, this paper integrates the signal paths of both FM0 and Manchester encodings together. This causes the overhead on the operation frequency of Manchester encoding. Suppose that both FM0 and Manchester encoders are operated at the same \( f \). The impact of \( \gamma \) on power consumption is described as follows.

In Fig. 12(b), the Manchester encoding sets the Mode = 1 and CLR = 0. In other words, the \( Q \) of DFF is kept at logic-0 in Manchester encoding. The transistor count of DFF is 44% of total transistor count. It is equivalent that the DFF almost dominates 44% of all switching-activity. The Manchester encoding saves the switching-activity of DFF, and thereby exhibits lower switching-activity. Instead, the FM0 encoding activates all logic components, and certainly leads to a higher switching-activity compared with that of Manchester encoding. The power profiling of both FM0 and Manchester encodings is shown in Fig. 16. The operation frequency ranges from 27 to 900 MHz for FM0 encoding and to 2 GHz for Manchester encoding, respectively. Since the switching-activity of FM0 encoding is higher than that of encoding, FM0 encoding has more power consumption than Manchester encoding at the same operation frequency.

The SOLS technique integrates Manchester and FM0 encodings into a fully reused hardware architecture. Obviously, the coding procedure of FM0 is more complex than that of Manchester. The data path of Manchester encoding is

![Table VIII](image-url)

**TABLE VIII**

**IMPLEMENTATION RESULT OF FM0 AND MANCHESTER LOGICS IN FIG. 6**

<table>
<thead>
<tr>
<th>Realization</th>
<th>FM0 logic</th>
<th>Manchester logic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supply voltage</td>
<td>1.8 V</td>
<td>3.3 V</td>
</tr>
<tr>
<td>Operation frequency</td>
<td>1 GHz</td>
<td>188 MHz</td>
</tr>
<tr>
<td>Power consumption</td>
<td>5.87 mW</td>
<td>22.9 mW</td>
</tr>
<tr>
<td>Transistor count</td>
<td>72</td>
<td>N/A</td>
</tr>
<tr>
<td>FPGA resource usage</td>
<td>Slice : 2</td>
<td>Slice : 1</td>
</tr>
<tr>
<td></td>
<td>Flip-Flop : 2</td>
<td>Flip-Flop : 0</td>
</tr>
<tr>
<td></td>
<td>LUTs : 2</td>
<td>LUTs : 1</td>
</tr>
<tr>
<td></td>
<td>Bonded IOBs : 3</td>
<td>Bonded IOBs : 3</td>
</tr>
<tr>
<td>Combination of article</td>
<td>[4], [5]</td>
<td>[7], [9]</td>
</tr>
</tbody>
</table>

![Table IX](image-url)

**TABLE IX**

**PERFORMANCE EVALUATION ON FM0 ENCODING**

<table>
<thead>
<tr>
<th>2009 [8]</th>
<th>This work</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design-flow</td>
<td>FPGA</td>
</tr>
<tr>
<td>( FOM_1 ) (Mbps/device)</td>
<td>6.88</td>
</tr>
<tr>
<td>( FOM_2 ) (device/coding mode)</td>
<td>14</td>
</tr>
<tr>
<td>HUR (%)</td>
<td>82.14</td>
</tr>
</tbody>
</table>

at the clock rate of 900 MHz, which is identical to that of FM0 encoding, this power consumption is estimated as 0.71 mW at 900 MHz. It is about 62.28% of that for FM0 encoding, which is 1.14 mW at 900 MHz. The reason why FM0 encoding dissipates more power than Manchester encoding is explained as follows: The dynamic power can be given as

\[
P = \gamma f C V^2
\]  

(17)

where \( \gamma \), \( f \), \( C \), and \( V \) denote the switching-activity, operation frequency, equivalent capacitance of a circuitry node, and supply voltage, respectively. The circuitry node in a CMOS circuit generally means one of three terminals of a CMOS transistor: source, drain, and gate. Note that the body is connected to power supply or ground. Every circuitry node is connected to each other, and total parasitic capacitances of a circuitry node is lumped into \( C \). Both FM0 and Manchester encodings are performed on same VLSI architecture, and their \( C \) and \( V \) are identical to each other. Suppose that both FM0 and Manchester encoders are operated at the same \( f \). The impact of \( \gamma \) on power consumption is described as follows.

In Fig. 12(b), the Manchester encoding sets the Mode = 1 and CLR = 0. In other words, the \( Q \) of DFF is kept at logic-0 in Manchester encoding. The transistor count of DFF is 44% of total transistor count. It is equivalent that the DFF almost dominates 44% of all switching-activity. The Manchester encoding saves the switching-activity of DFF, and thereby exhibits lower switching-activity. Instead, the FM0 encoding activates all logic components, and certainly leads to a higher switching-activity compared with that of Manchester encoding. The power profiling of both FM0 and Manchester encodings is shown in Fig. 16. The operation frequency ranges from 27 to 900 MHz for FM0 encoding and to 2 GHz for Manchester encoding, respectively. Since the switching-activity of FM0 encoding is higher than that of encoding, FM0 encoding has more power consumption than Manchester encoding at the same operation frequency.
restricted to that of FM0 encoding. Then, the operation frequency of Manchester encoding is also limited by that of FM0 encoding. Our work targets at an efficient integration of hardware devices for Manchester encoding and FM0 encoding instead of operation frequency and power consumption. Generally, more coding methods a hardware architecture can support, more hardware devices it requires. A performance evaluation is given under identical conditions for a more objective evaluation. A building block that can perform FM0 and Manchester encodings is considered an evaluation platform. The literatures [4], [5], and [7]–[9] are dedicated for either FM0 or Manchester encoding. To make these literatures fit the evaluation platform, the FM0 logic of Fig. 6 is implemented in full-custom to combine [4] and [5] and in FPGA to combine [7] and [9]. The Manchester logic of Fig. 6 is realized in FPGA to combine [8]. Table VIII shows the implementation result of FM0 and Manchester logics.

The evaluation platform consists of two coding methods, and how to efficiently allocate hardware devices to perform them is a critical design issue for performance evaluation. To measure the device-efficiency of each work, two figure of merits (FOMs) are defined as follows:

\[
\begin{align*}
FOM_1 &= \frac{\text{Data rate}}{\text{Total devices}}; \text{(Mb/s/device)} \\
FOM_2 &= \frac{\text{Total devices}}{\text{Total coding modes}}; \text{(device/coding mode)}
\end{align*}
\]

where the data rate is either for FM0 or Manchester encoding. Total coding modes denote how many coding methods a hardware architecture can support. In this evaluation platform, total coding modes are set to 2, representing FM0 and Manchester encodings. The total devices mean the total transistor count and total FPGA resource for full-custom and FPGA designs, respectively. FPGA resource incorporates Slice, Flop-Flip, LUTs, and Bonded IOBs. The total FPGA resource is a summation of the FPAG resource used in both Manchester and FM0 encodings. Similarly, the total data rate is a summation of the data rates of both FM0 and Manchester encodings. The data rate of each work is identical to the operation frequency of each work. The operation frequencies designed with non-0.18 μm are normalized by (15) and (16), as previously shown in Table VII. Hence, the compensation of technology-dependent differences on data rate is considered in FOM1. The FOM1 means how high the data rate a single device can contribute on average is. The FOM2 represents how many devices are required for a single coding mode on average. This performance evaluation is classified into two categories: FM0 and Manchester encodings in Tables IX and X, respectively. For FM0 encoding, this paper exhibits outstanding FOM1, FOM2, and HUR in comparison with [8]. For Manchester encoding, the FOM1 of this paper is less than that of [9] by 15.35%. However, this paper has almost half the FOM2 of [12] and 2.8 times the HUR of [12]. In full-custom design flow, this paper still has higher performance on FOM1, FOM2, and HUR compared with [4] and [5]. Therefore, the SOLS technique presents a quite competitive performance on device-efficiency.

C. FPGA Prototyping

This paper is also implemented with FPGA not only for an objective comparison but also for the functional prototyping, as shown in Fig. 17. However, the signal-transition of FM0 and Manchester codes is not aligned to the positive- or
negative-edge trigger. To be compatible with the synchronization of FPGA, two sets of clock, CLK EXT and CLK INT, are adopted in this FPGA prototyping system. The frequency of CLK EXT is twice as fast as that of CLK INT. The faster CLK EXT is used to synchronize every signal inside FPGA. The slower CLK INT is for the encoding manipulation of FM0 and Manchester codes. The waveforms of the functional verification are shown in Fig. 18, where the FM0 and Manchester encodings are shown in Fig. 18(a) and (b), respectively.

VI. 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 TSMC 0.18-μm IP6M 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.

ACKNOWLEDGMENT

The authors would like to thank each reviewer for their precious comments and suggestions, and Prof. J.-H. Deng for his support in the experiment.

REFERENCES


Yu-Hsuan Lee (M’11) was born in Taipei, Taiwan, in 1979. He received the B.S. degree from the Department of Electronics Engineering, Southern Taiwan University, Tainan, Taiwan, and the M.S. and Ph.D. degrees from the Graduate Institute of Electrical Engineering, National Central University, Taoyuan, Taiwan, in 2003 and 2010, respectively. He joined the Department of Electrical Engineering, Yuan-Ze University, Taoyuan, in 2010, where he is currently an Assistant Professor. His current research interests include video/image processing, coding, and VLSI architecture design. Prof. Lee is a member of Phi Tau Phi.

Cheng-Wei Pan was born in Hsinchu, Taiwan, in 1987. He received the B.E. degree from the Department of Industrial Engineering and Management, National Chiao-Tung University, Hsinchu, in 2009. He is currently pursuing the M.S. degree with the Department of Electrical Engineering, Yuan-Ze University, Taoyuan, Taiwan.