

Wire Optimization and Delay Reduction for High-Performance on-Chip Interconnection in GALS Systems

Myeong-Hoon Oh, Young Woo Kim, Hag Young Kim, Young-Kyun Kim, and Jin-Sung Kim

To address the wire complexity problem in large-scale globally asynchronous, locally synchronous systems, a current-mode ternary encoding scheme was devised for a two-phase asynchronous protocol. However, for data transmission through a very long wire, few studies have been conducted on reducing the long propagation delay in current-mode circuits. Hence, this paper proposes a current steering logic (CSL) that is able to minimize the long delay for the devised current-mode ternary encoding scheme. The CSL creates pulse signals that charge or discharge the output signal in advance for a short period of time, and as a result, helps prevent a slack in the current signals. The encoder and decoder circuits employing the CSL are implemented using 0.25- μm CMOS technology. The results of an HSPICE simulation show that the normal and optimal mode operations of the CSL achieve a delay reduction of 11.8% and 28.1%, respectively, when compared to the original scheme for a 10-mm wire. They also reduce the power-delay product by 9.6% and 22.5%, respectively, at a data rate of 100 Mb/s for the same wire length.

Keywords: High-performance interconnection, Current mode circuit, Asynchronous protocol, Delay insensitive, CSL.

I. Introduction

With the increase in demand for a multicore system-on-a-chip (SoC), the global clock distribution and power dissipation have become significant factors. To reduce or eliminate the negative effects of a global clock in a synchronous design, some asynchronous alternatives have been studied [1]. However, such asynchronous designs could cause other negative effects, such as a large area overhead. In response to this dilemma, globally asynchronous, locally synchronous (GALS) design techniques have been proposed.

Asynchronous handshake protocols with delay-insensitive (DI) characteristics are divided largely into two- and four-phase protocols depending on the number of requests and acknowledgement signal pairs. Because valid data are transmitted at the rising edges of both the request and acknowledgement signals, four-phase protocols match well with the operation of existing storage elements and are easy to implement [2]. However, four-phase protocols are inefficient owing to the long sequence of transitions, that is, rise of request, rise of acknowledgement, fall of request, and fall of acknowledgement signals, where the two falling transitions are not involved in a data transmission. In particular, for a global interconnect assuming a relatively long wire, such protocols can significantly degrade the performance of the overall system and negatively impact the power dissipation. Meanwhile, two-phase asynchronous handshake protocols such as a level encoded dual-rail (LEDR) transmit data in both the rising

Manuscript received Aug. 11, 2016; revised Apr. 25, 2017; accepted May 22, 2017.

Myeong-Hoon Oh (mhoonoh@etri.re.kr) and Young Woo Kim (bartmann@etri.re.kr) are with the SW & Contents Research Laboratory, ETRI and the Department of Computer Software, University of Science and Technology (UST), Daejeon, Rep. of Korea.

Hag Young Kim (h0kim@etri.re.kr) and Young-Kyun Kim (kimyoung@etri.re.kr) are with the SW & Contents Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Jin-Sung Kim (corresponding author, jinsungk@sunmoon.ac.kr) is with the Department of Electronic Engineering, Sun Moon University, Asan, Rep. of Korea.

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (<http://www.kogil.or.kr/news/dataView.do?dataIdx=97>).

and falling edges of the request and acknowledgement signals, and in [3] and [4] they were proven to be better than four-phase protocols in terms of their performance and power. Two-phase protocols are currently preferred for on-chip communication schemes and are widely considered to be one of the best possible solutions for a high-performance global interconnection network in large-scale SoCs. However, two-phase protocols use an excessive number of wires. Because a large number of cores are able to be integrated into a single chip owing to advances in deep sub-micron processes, the number of wires connecting them has increased dramatically [5]–[8], resulting from the excessive number of wires used by popular bus protocols. In an SoC design, an excessive number of wires, but with a limited number of metal layers, has broadened the metallization space to avoid crosstalk noise during high-speed signaling, thereby leading to an increase in the SoC die area [9], [10].

To address this issue, the authors have devised a ternary encoding scheme for a two-phase asynchronous protocol [11]. The devised scheme uses current-mode circuits [12] to present multiple-value signals in a single wire, and as a result, reduces the number of wires as much as four-phase protocols. However, for data transmission through a long wire, this scheme is very vulnerable to a signal delay. Few studies have been conducted on controlling the long signal delay in current-mode circuits, whereas the increased delay can be reduced by simply inserting buffers into the middle points of the long wire used in voltage-mode circuits, such as those applied in an LEDR. The lack of research regarding this issue is due to the fact that current-mode circuits can lead to a serious communication error in the signal transmission even if the output signal is modified slightly to reduce the delay, because the noise margin is too narrow to guarantee a certain level of current when compared to general digital logic circuits.

This paper proposes a current steering logic (CSL) that is able to minimize long delays for the ternary encoding scheme without incurring any communication errors. The CSL creates six pulses that charge or discharge the output signal in advance to decrease the slack in the current signals. Among the six signals, the widths of two signals are adjusted carefully and exquisitely to avoid a malfunction of the decoder.

The remainder of this paper is organized as follows: The following section discusses some previous related studies. Section III briefly reviews the previous work on the ternary encoding scheme, and introduces the proposed current steering logic. The role and operation mechanism of the CSL are also presented in detail. Section IV describes the implemented encoder and decoder circuits,

which include a CSL using CMOS technology; evaluates their performance; and consequentially compares the circuits with the pure ternary encoding scheme without a CSL. Finally, some concluding remarks are presented in Section V.

II. Related Studies

Multi-valued logic circuits are used for transmitting data with a small number of wires while maintaining the DI characteristics. In general, current-mode multi-valued logic circuits that use a current to express multiple values are preferred over voltage-mode multi-valued logic circuits that express multiple values using voltage levels because the noise margin characteristic of the voltage-mode circuits is very unstable owing to the supply power, which is gradually decreased during operation. Thus, the methods applied in [11] and [13] through [15] use current-mode multi-valued logic as a DI transmission mechanism. In fact, unlike voltage-mode based circuits, a DI transmission based on current-mode multi-valued logic does not use a repeater or pipeline register to improve the data throughput, and is thus more advantageous in terms of cost and power efficiency compared to an interconnect employing voltage-mode circuits. A repeater or pipeline is not required because the completion of a transmission is detected only once at the receiver [12]. However, the global interconnect of such a current-mode circuit with no repeater can lead to significant limitations in performance when the scale of the chip increases. In other words, if the scale of the chip increases, the length of the transmission wire increases and its resistance is enlarged, eventually causing any transmission delay through the wire to significantly increase. Therefore, the long-wire delay problem should be resolved for large-scale SoCs.

Meanwhile, in [16], the design and implementation of another ternary encoder are described. However, this encoder uses a four-phase protocol instead of a two-phase protocol, and its performance enhancement may therefore be limited, as discussed in Section I. It also uses voltage-mode circuits instead of current-mode circuits for the ternary encoding, that is 0, $1/2V_{dd}$, and V_{dd} . The main contribution of the encoder is simply a minimization of the state transitions for an energy reduction, and the authors did not deal with a long-wire delay, which is our major concern.

III. Current Steering Logic

This section briefly reviews the previous work on the ternary encoding scheme, as well as the proposed CSL and its detailed operations.

1. Ternary Coding Scheme

Figure 1 shows the devised scheme. In the encoder, voltage-mode data and handshaking signals are encoded in three levels of current, and sent to the decoder in a DI

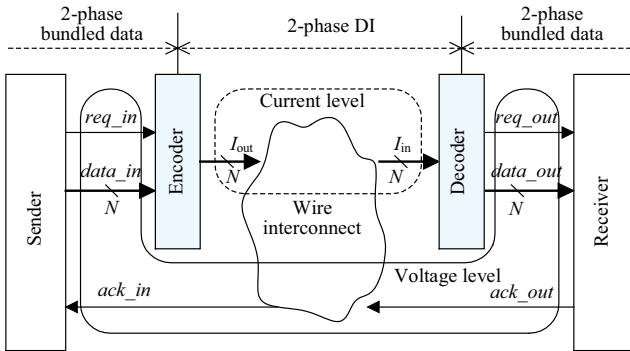


Fig. 1. Transmission environment.

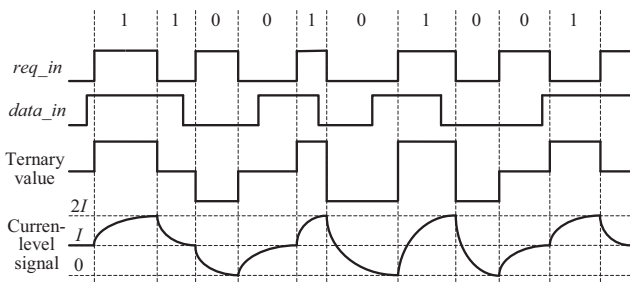


Fig. 2. Example data transmission using ternary encoding.

format. The encoded current-mode values are restored to the original voltage-mode data and protocol signals in the decoder.

Figure 2 illustrates an example of a data transmission using this scheme. The two input signals *req_in* and *data_in* express two-phase bundled data. Basically, the transmission of data 1 and 0 are represented as a high current value ($2I$), and a low current value (0), respectively. If the data are the same as those of the previous transmission, the transmission signal is represented as a middle current value (I). In this way, the number of wires required between the encoder and decoder is reduced to half of those ($2N$) for an LEDR to transfer N -bits of data. However, if the data are transmitted through a very long wire, a lengthy amount of time is consumed for charging or discharging the output current (I_{out} in Fig. 1) to a certain level owing to its resistance and capacitance. Figure 3 shows that it takes a significant amount of time to restore the original request signal (*req_in*) at *req_out* in the decoder if the signal is transmitted through a 10-mm long wire. The waveforms shown in the figure are the results of an HSPICE simulation using 0.25- μm CMOS technology.

To reduce this amount of time, the authors employ a CSL that creates pulse signals to charge or discharge the output signal in advance for a short period of time, and as a result, help prevent a slack in the current signals. A conceptual block diagram of the ternary encoder using the proposed CSL is provided in Fig. 4.

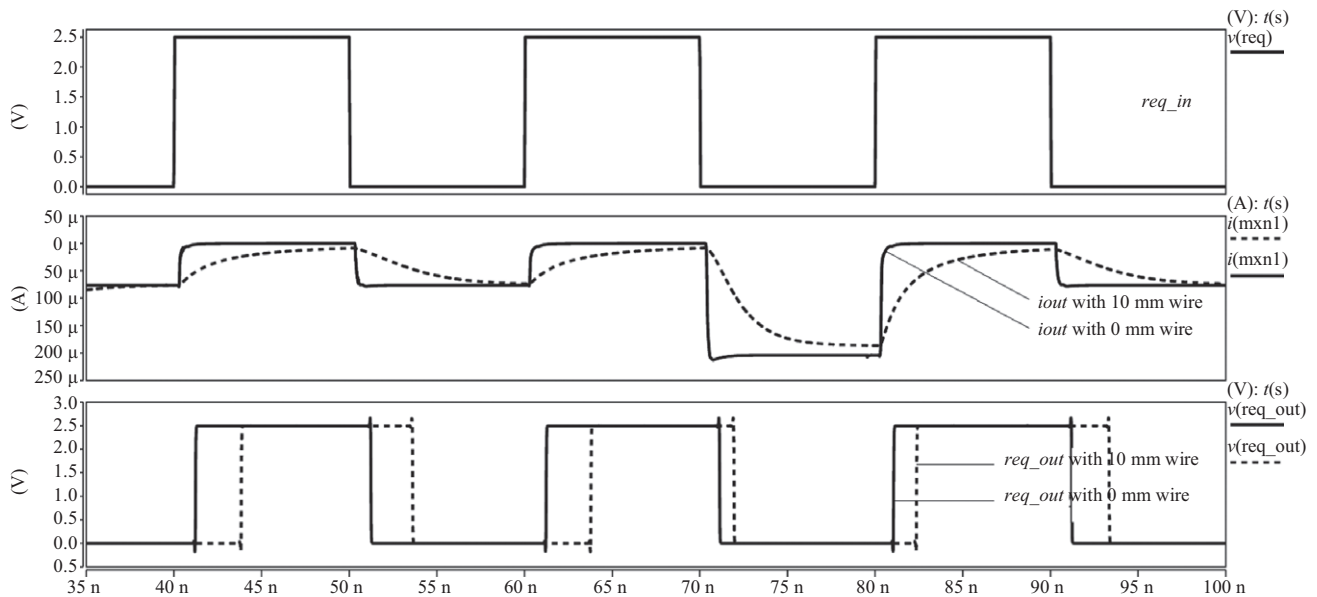


Fig. 3. Performance degradation caused by wire length.

2. Current Steering Logic

A detailed circuit diagram for the encoder and CSL is shown in Fig. 5. For the falling and rising edges of the *req_in* signal, two double-edge-triggered flip-flops, shown in Fig. 5, change their state (*Q1*, *Q0*) according to the input signal *data_in*. The two transistors *P1* and *P2* on the right composing a current mirror produce the reference current *I* by changing the constant current (*I_s*) in a current

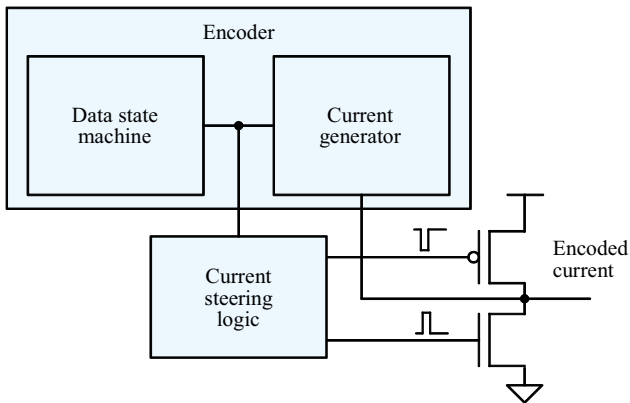


Fig. 4. Conceptual block diagram of encoder and current steering logic.

source with transistors *P0* and *N0*. Two other transistors *N1* and *N2* make up three types of current depending on the logic values of *Q1* and *Q0*. The CSL described below also works depending on the values, and generates six short pulses on the *charge_a*, *charge_b*, *charge_c*, *discharge_a*, *discharge_b*, and *discharge_c* signals to decrease the slack in output current *I_{out}*. Table 1 shows the pulse signals that should be generated for the state transitions in the two flip-flops.

To generate a *charge_a* signal, the CSL should detect a change in only *Q1* when both *Q1* and *Q0* become logic 1. Likewise, to generate a *discharge_a* signal, the CSL should

Table 1. Pulse types generated from CSL.

Transition in <i>I_{out}</i>	<i>Q1</i>	<i>Q0</i>	Pulse signals
$I \rightarrow 2I$	0 → 1	1	<i>charge_a</i>
$I \rightarrow 0$	0	1 → 0	<i>discharge_a</i>
$0 \rightarrow 2I$	0 → 1	0 → 1	<i>charge_b</i>
$2I \rightarrow 0$	1 → 0	1 → 0	<i>discharge_b</i>
$0 \rightarrow I$	0	0 → 1	<i>charge_c</i>
$2I \rightarrow I$	1 → 0	1	<i>discharge_c</i>

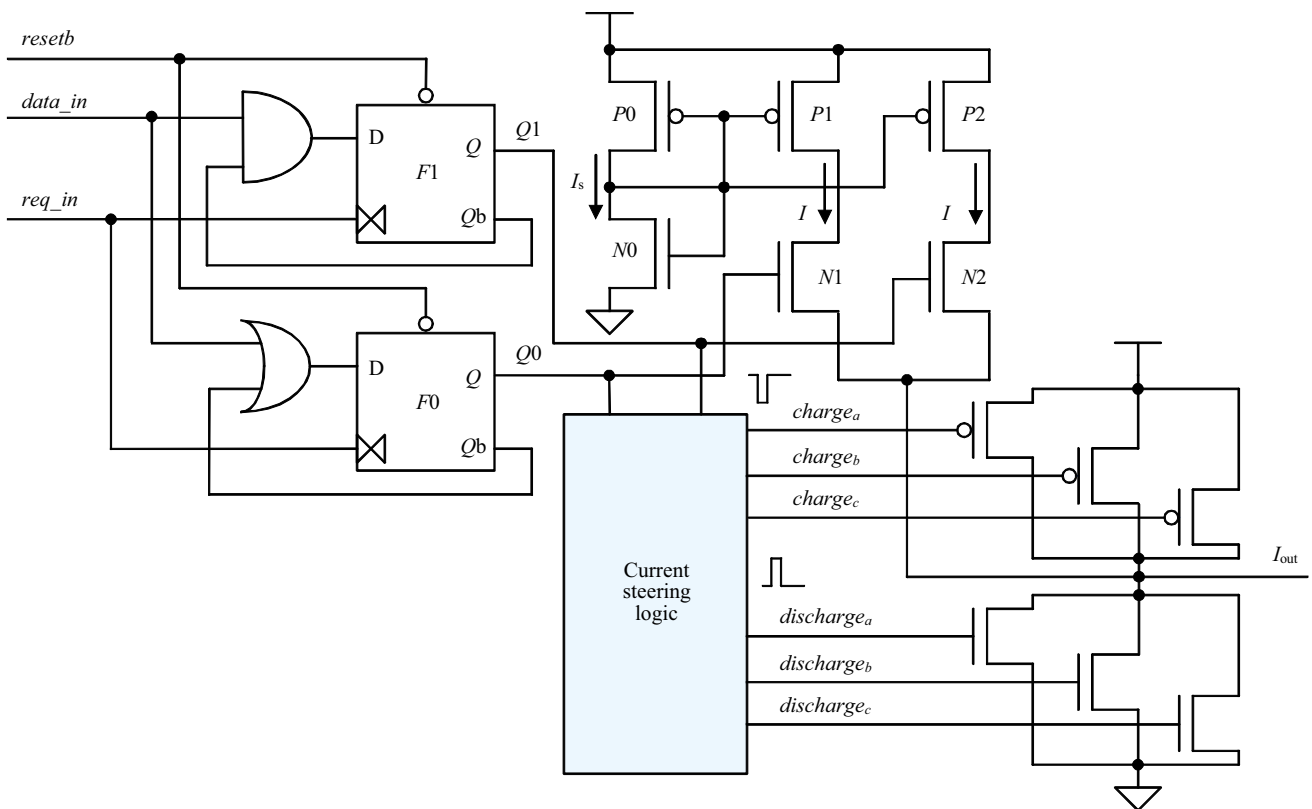


Fig. 5. Circuit diagram for encoder and current steering logic.

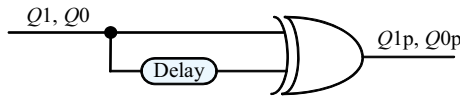


Fig. 6. Change detection logic in $Q1, Q0$ signals.

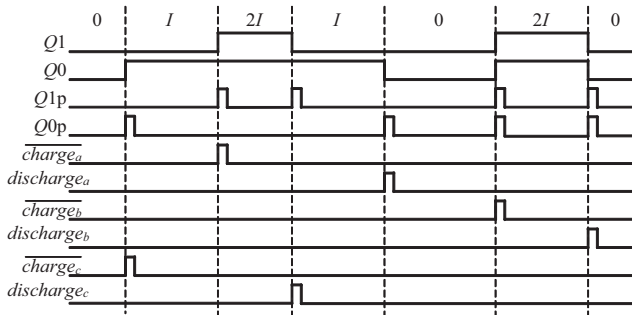


Fig. 7. Waveforms of pulse signals generated from CSL.

detect a change in only $Q0$ when both $Q1$ and $Q0$ become logic 0. The signal changes in $Q1$ and $Q0$ can be detected using a delay element and an XOR gate, as shown in Fig. 6. They produce pulse signals ($Q1p$ and $Q0p$) that have the same width as the delay element at the changing time of $Q1$ and $Q0$, respectively. Other pulse signals, that is, $charge_b$, $discharge_b$, $charge_c$, and $discharge_c$, are also generated using the pulse singles for $Q1$ and $Q2$, that is, $Q1p$ and $Q0p$, respectively.

Figure 7 shows waveforms that help us understand how the six pulse signals of the CSL are produced, and based upon which, each pulse signal can be expressed through the following conditional equations. For convenience, the $charge_a$, $charge_b$, and $charge_c$ signals are marked as inverted signals.

$$\begin{aligned} \overline{charge_a} &= Q1p \cdot \overline{Q0p} \cdot Q1 \cdot Q0, \\ \overline{discharge_a} &= \overline{Q1p} \cdot Q0p \cdot \overline{Q1} \cdot \overline{Q0}, \\ \overline{charge_b} &= Q1p \cdot Q0p \cdot Q1 \cdot Q0, \\ \overline{discharge_b} &= Q1p \cdot Q0p \cdot \overline{Q1} \cdot \overline{Q0}, \\ \overline{charge_c} &= \overline{Q1p} \cdot Q0p \cdot \overline{Q1} \cdot Q0, \\ \overline{discharge_c} &= Q1p \cdot \overline{Q0p} \cdot \overline{Q1} \cdot Q0, \end{aligned}$$

As shown in Fig. 5, although it is possible to compose the pre-charging (or pre-discharging) circuits using only a single pair, they are composed of three pairs of P/N transistors to save the logic circuit area because OR gates are required, and thus the gate counts increase if each of the three $charge_x$ and $discharge_x$ signals are merged into one signal inside the CSL.

In the case of a change to high ($2I$) or low (0), a control miss of the pulse width does not significantly affect the logical decision of the decoder. However, in the case of a change to the middle (I), the decoder may consider it as

Table 2. Number of NAND gates for delay element.

Pulse signals	Normal mode	Optimal mode
$charge_a$	8	8
$discharge_a$	8	8
$charge_b$	8	8
$discharge_b$	8	8
$charge_c$	Not used	2 to 12 depending on wire length
$discharge_c$	Not used	2 to 12 depending on wire length

high or low momentarily if the current is too charged over I or discharged under I , respectively. In addition, considering the RC delay at a different wire length, a precise adjustment of the pulse width is necessary.

In fact, the delay element shown in Fig. 6, which determines the width of each pulse signal, is basically implemented using eight NAND gates connected in-series. However, for the $charge_c$ and $discharge_c$ signals in the case of a change to the middle (I), two to twelve NAND gates are used instead of the fixed eight so as to modulate their pulse widths. The CSL works on two different modes: normal and optimal. Normal mode does not activate the $charge_c$ and $discharge_c$ signals, and thus a control miss of the pulse width can be fundamentally avoided. Optimal mode uses all of the six pulse signals and exquisitely adjusts the pulse widths of the $charge_c$ and $discharge_c$ signals, thereby compensating for the RC delay caused by the length of the transmission wire, and maintains the current under or over I for each charge or discharge case, respectively.

The implementations of the delay length for each mode are summarized in Table 2.

The delay element in Fig. 6 might be able to be changed through the variation in process, voltage, and temperature (PVT), that is, the widths of the six output signals from the CSL might be changed if affected by such variation. However, the circuits in normal mode will work fine because the $charge_c$ and $discharge_c$ signals that steer the current to change to intermediate level I are deactivated.

For optional mode, although a longer delay element for the $charge_c$ and $discharge_c$ signals can guarantee the operation, a shorter-delay element is used as the maximum delay element (12 NAND gates). This means that our circuit is less sensitive to a current change by the variations in PVT in terms of its functionality. A detailed experiment conducted to check whether the circuits can work well based on the length of the delay in optional mode is described in Section IV.

3. Decoder

The encoded levels of a current are transformed into the previously defined voltage levels in a decoder. A schematic of a decoder is shown in Fig. 8. Transistors $N0$, $P0$, $P1$, and $P2$ work as a current source and a current mirror in the same manner as the encoder. A three-valued input current I_{in} is applied to $N1$. The transistor $N1$ then copies I_{in} to the drains of $N2$ and $N3$, which jointly act as a current comparator through the coupling with $P1$ and $P2$. A current mirror with $P1$ and $P2$ should create threshold currents $0.5I$ and $1.5I$ in order to generate differential currents according to the input current I_{in} . Nodes labeled a and b at the drains of $N1$ and $N2$, respectively, take logic 1 as long as I_{in} is 0 because $N1$ and $N2$ do not pull any current. When input current I is applied, node a takes the voltage of logic 0 because $N2$ consumes all of the threshold current $0.5I$, whereas node b still remains at the voltage level of logic 1 owing to the remaining differential current ($1.5I - I = 0.5I$) at node b . For a similar reason, both nodes a and b have a voltage level of logic 0 for the input current $2I$. These voltage value combinations are used to reconstruct the original voltage-mode input signals with standard CMOS logic gates. Table 3 lists the logical voltage values at nodes a and b according to each current level.

According to Table 3, the original data signal ($data_out$) is easily restored using an SR latch ($F0$) and a combination of a and b , as shown in Fig. 8. To restore the request signal (req_out), the rising and falling edges of the signal should alternate with each other whenever a new

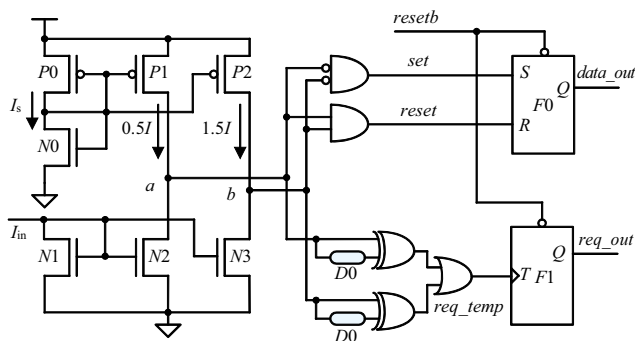


Fig. 8. Circuit diagram for decoder.

Table 3. Logical values of nodes a and b .

Input current	Values of (a , b)	$data_out$
0	(1, 1)	0
I	(0, 1)	Previous data
$2I$	(0, 0)	1

input current signal arrives. The decoder generates a pulse signal (req_temp) whose width is similar to the time delay of the delay element ($D0$). The pulse signal always goes to logic 1 whenever any changes in a or b are detected, and eventually toggles the signal req_out in the T-flip-flop ($F1$).

When the input current varies from 0 to $2I$, or vice versa, a very small time difference may occur between the changes in both a and b when they are implemented in read circuits. This can make a req_temp signal be asserted twice, and consequently a req_out signal can cause a malfunction of the handshake protocols. The delay element $D0$ helps prevent this from occurring by filtering the time difference.

IV. Implementation and Simulation

The encoder and decoder circuits are implemented using $0.25\text{-}\mu\text{m}$ five-metal CMOS technology (ANAM) at the transistor level. A supply voltage of 2.5 V is used to operate the circuits, and the reference current I is set to $96\ \mu\text{A}$. The simulation model for the wires is based on a distributed RC model composed of five sections, and the third metal layer of the technology is used as the reference for setting its parameters [17]. Based on these parameters, specific values for R and C are used for each wire length, as shown in Table 4.

In fact, our simulation was conducted using a pre-layout stage because we designed our circuits at the transistor level. However, by applying the specific parameters from the technology and using a commercial simulation tool, HSPICE, the performance impact from the RC delay at the pre-layout stage can be maintained at the post-layout level or actual implementation level.

Figure 9 shows the HSPICE simulation results for the original scheme without the CSL (req_out_org), normal-mode operation with the CSL (req_out_CSL), and its optimal-mode operation ($req_out_CSL_opt$) when a req_in signal is transmitted through a 10-mm long wire. The propagation delay is significantly reduced in both modes of the CSL when compared to the original scheme.

Table 5 shows the time delay estimated with a gradual change in wire length. For a short distance, the original scheme (org) has a shorter delay than the others.

Table 4. R and C values for each wire length.

Wire length	2 mm	4 mm	6 mm	8 mm	10 mm
R (Ω)	32.5	65	97.5	130	162.5
C (fF)	4	8	12	16	20

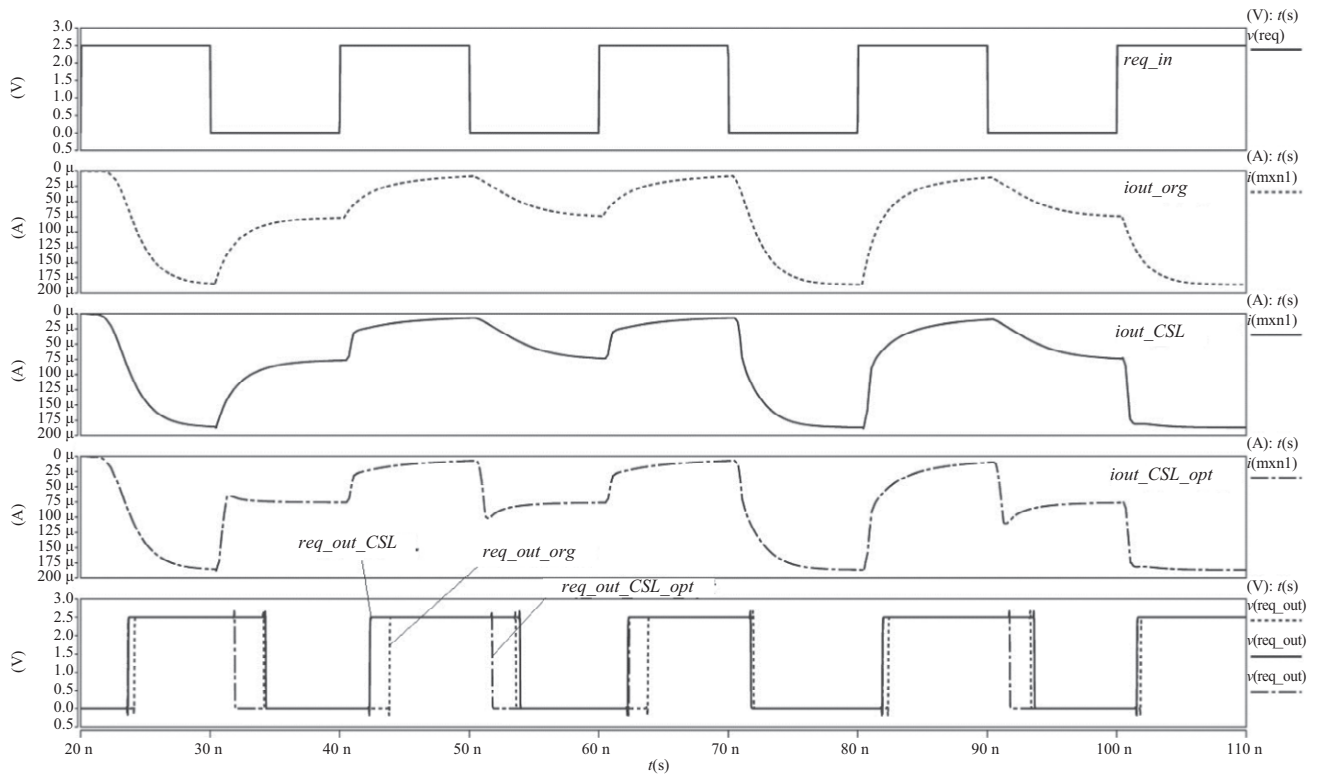


Fig. 9. HSPICE simulation results.

Table 5. Propagation delay (ns) estimation with change in wire length.

Wire length (mm)	org	CSL	CSL_opt
0	1.156	1.295	1.320
2	1.603	1.690	1.471
4	1.968	1.914	1.556
6	2.306	2.143	1.643
8	2.627	2.356	1.740
10	2.939	2.592	1.864

However, the opposite is true for a length of greater than 2 mm. For a 10-mm wire, the normal (CSL) and optimal (CSL_opt) modes achieve a delay reduction of 11.8% and 28.1%, respectively, compared to the original scheme.

As mentioned in Section III-2, we use the $charge_c$ and $discharge_c$ signals in optimal mode, and control the pulse width of each signal using a delay element that connects up to 12 NAND gates in-series. Based on an experiment increasing the pulse width, it was shown that our circuits can operate well with up to 16 NAND gates, although their operation is not guaranteed beyond that. In other words, even if the delay is increased by the variation in PVT, and thus the current is also increased, the circuits

designed with a maximum of 12 NAND gates guarantee operation of up to 33.3% $((16 - 12)/12 * 100)$ of the delay increment.

To compare the three schemes in terms of power consumption and propagation delay, the power-delay product (PDP) is estimated using gradual changes in the data rate and wire length, as shown in Table 6. As the data rate or wire length increases, the PDP values also increase for all schemes. However, the increasing rates of both operation modes of the CSL are lower than those of the original scheme. At a data rate of 100 Mb/s in a 10-mm long wire, normal and optimal modes achieve a PDP reduction of 9.6% and 22.5%, respectively, when compared to the original scheme.

In the PDP-based evaluation of the proposed scheme, we take only dynamic power into account, excluding the leakage power, because the leakage power is negligible in the technology used in our simulation. Considering recent technologies in which the portion of leakage power is significant, the two additional logic parts, that is, the CSL and the current sourcing circuits, may increase the leakage power.

However, because we assume the GALS system which uses a global interconnection, the encoder containing the proposed CSL will be applied to only a few special ports

Table 6. Average power delay product with changes in data rate and wire length: (a) 0 mm, (b) 2 mm, (c) 4 mm, (d) 6 mm, (e) 8 mm, and (f) 10 mm wire lengths.

Data rate (MHz)	2.5	5	10	16.7	25	33.3	41.6	50	62.5	71.4	83.3	100
(a)												
org	872.76	899.99	971.85	1,078.86	1,207.23	1,331.12	1,517.74	1,574.47	1,752.08	1,890.12	2,097.10	2,304.08
CSL	981.17	1,014.37	1,098.92	1,224.51	1,376.08	1,522.31	1,743.89	1,812.09	2,025.42	2,186.31	2,433.42	2,680.52
CSL_opt	1,004.12	1,042.05	1,135.79	1,274.51	1,441.44	1,604.14	1,842.88	1,924.14	2,161.02	2,339.32	2,612.85	2,886.38
(b)												
org	1,213.65	1,255.61	1,360.24	1,513.41	1,696.98	1,873.92	2,127.36	2,219.43	2,471.25	2,664.49	2,954.11	3,243.73
CSL	1,280.21	1,323.83	1,435.47	1,598.25	1,793.53	1,984.43	2,269.06	2,357.52	2,703.14	2,845.98	3,164.08	3,482.18
CSL_opt	1,122.70	1,168.46	1,280.45	1,444.30	1,640.65	1,832.03	2,112.21	2,210.65	2,492.33	2,701.95	3,025.75	3,349.56
(c)												
org	1,490.15	1,544.66	1,677.46	1,868.52	2,096.00	2,315.94	2,621.57	2,741.64	3,058.78	3,296.79	3,657.55	4,018.30
CSL	1,450.75	1,503.85	1,632.87	1,820.81	2,046.31	2,263.50	2,578.35	2,692.54	3,011.16	3,246.79	3,608.52	3,970.25
CSL_opt	1,190.95	1,244.21	1,369.81	1,549.39	1,767.10	1,977.55	2,277.07	2,398.04	2,708.86	2,937.88	3,292.33	3,646.78
(d)												
org	1,745.20	1,810.90	1,971.91	2,200.64	2,472.33	2,733.12	3,088.98	3,248.90	3,616.71	3,891.81	4,306.73	4,721.65
CSL	1,624.69	1,686.58	1,837.02	2,053.10	2,310.26	2,556.34	2,901.17	3,044.47	3,407.58	3,672.03	4,082.45	4,492.86
CSL_opt	1,258.23	1,319.51	1,460.94	1,660.53	1,895.18	2,125.47	2,449.47	2,588.22	2,931.93	3,184.48	3,570.98	3,957.48
(e)												
org	1,985.46	2,066.56	2,256.41	2,517.82	2,833.11	3,138.40	3,531.40	3,720.75	4,147.93	4,474.04	4,959.33	5,444.62
CSL	1,783.94	1,857.14	2,028.59	2,274.46	2,561.96	2,838.65	3,217.31	3,384.30	3,784.42	4,083.28	4,529.69	4,976.11
CSL_opt	1,334.51	1,403.57	1,559.16	1,777.51	2,039.75	2,295.08	2,645.01	2,802.67	3,177.48	3,452.06	3,881.74	4,311.42
(f)												
org	2,222.97	2,315.81	2,530.68	2,834.52	3,192.67	3,530.97	3,954.01	4,191.78	4,669.48	5,012.58	5,568.42	6,124.26
CSL	1,964.58	2,049.00	2,241.92	2,513.67	2,839.98	3,145.86	3,556.51	3,751.63	4,204.59	4,532.45	5,035.39	5,538.33
CSL_opt	1,435.22	1,512.26	1,685.37	1,929.50	2,219.89	2,505.85	2,886.61	3,062.29	3,483.59	3,789.94	4,269.24	4,748.54

(for global interconnections), and the sub-modules of the GALS system will have a much smaller number of global ports than local ports. As a result, the portion occupied by the CSL is expected to be relatively small.

If the proposed scheme is applied to the latest technologies, we can expect certain benefits. In terms of the supply voltage, the voltage-mode multi-valued logics reduce the margins between voltage levels, and are thus vulnerable to signal decoding because semiconductors manufactured using the latest technology operate at lower supply voltages. On the other hand, because the current-mode multi-valued logic proposed in this paper controls the amount of current to represent the corresponding values, the signal robustness can be retained even in a relatively low supply-voltage environment.

With GALS implemented using the latest technology, data signals physically pass through far longer wires when they are transmitted globally. Therefore, a solution is required to improve the performance as well as provide

stability for data transmissions over relatively long wires. If the proposed CSL is used for such purposes, we can expect certain benefits to be derived. Our results also indicate that the gain increases with an increase in wire length.

However, in terms of design complexity, the data rate will be much higher in semiconductors implemented using the latest technology, and it is therefore necessary to control the amount of current to minimize the power consumption while maintaining the data rate. Consequently, precisely controlling the amount of current will increase the design complexity because more preliminary experimentation and tuning will be required.

V. Conclusion

For this study, we revised a current-mode ternary encoding scheme and proposed a current steering logic (CSL) to minimize the performance degradation when transmitting data through a very long interconnection in a

large-scale GALS system. Six pulse signals are created that charge or discharge the output signal in advance so as to decrease the slack in the current signals. Among these six signals, the *charge_c* and *discharge_c* signals need a precise adjustment of their widths to avoid a malfunction of the decoder. Two different modes are suggested depending on whether these two signals are used or not. The normal-mode operation of the CSL does not activate either of the two signals. The optimal-mode operation uses all six pulse signals, and exquisitely adjusts the pulse widths of the two signals. HSPICE simulation results indicate that both modes surpass the original ternary encoding scheme without the use of the CSL. In addition, an optimal-mode operation was proved to be better than a normal-mode operation in terms of the delay and power.

The potential engineering applications of this novel circuit include relatively long interconnections for high-performance data communications between large cores that have suffered from wire complexity problems, such as capacitive crosstalk noise during high-speed signaling.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (NO.B0101-16-0548, Low-power and High-density Micro Server System Development for Cloud Infrastructure).

References

- [1] M.H. Oh et al., "Architectural Design Issues on a Clockless 32-bit Processor Using an Asynchronous HDL," *ETRI J.*, vol. 35, no. 3, June 2013, pp. 480–490.
- [2] J. Sparsø and S.B. Furber, *Principles of Asynchronous Circuit Design: A System Perspective*, Dordrecht, Netherlands: Kluwer Academic Publishers, 2001.
- [3] W.F. McLaughlin, A. Mitra, and S.M. Nowick, "Asynchronous Protocol Converters for Two-Phase Delay-Insensitive Global Communication," *IEEE Trans. VLSI Syst.*, vol. 17, no. 7, July 2009, pp. 923–928.
- [4] M.E. Dean, T.E. Williams, and D.L. Dill, "Efficient Self-Timing with Level-Encoded 2-Phase Dual-Rail (LEDR)," *Proc. Univ. California/Santa Cruz Conf. Adv. Res. VLSI*, Santa Cruz, CA, USA, 1991, pp. 55–70.
- [5] J. Kim, K. Choi, and G. Loh, "Exploiting New Interconnect Technologies in On-chip Communication," *IEEE J. Emerg. Select. Topics Circuits Syst.*, vol. 2, no. 2, June 2012, pp. 124–136.
- [6] C.A. Zeferino et al., "Models for Communication Trade-offs on System-on-Chip," in *Proc. Intell. Workshop IP-Based SoC Des.*, Oct. 2002, p. 394.
- [7] C.T. Hsieh and M. Pedram, "Architectural Energy Optimization by Bus Splitting," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 21, no. 4, Apr. 2002, pp. 408–414.
- [8] T. Seceleanu, J. Plosila, and P. Liljeberg, "On-chip Segmented Bus: A Self-Timed Approach," in *Annu. IEEE Int. ASIC/SOC Conf.*, Rochester, NY, USA, Sept. 25–27, 2002, pp. 216–221.
- [9] J. Lee and H.-J. Lee, "Wire Optimization for Multimedia SoC and SiP Designs," *IEEE Trans. Circuits Syst. I: Regular Paper*, vol. 55, no. 8, Sept. 2008, pp. 2202–2215.
- [10] J. Lee, H.-J. Lee, and C. Lee, "A Phase-Based Approach for On-chip Bus Architecture Optimization," *Comput. J.*, vol. 52, no. 6, Aug. 2009, pp. 626–645.
- [11] M.H. Oh and S.W. Kim, "Asynchronous 2-Phase Protocol Based on Ternary Encoding for On-chip Interconnect," *ETRI J.*, vol. 33, no. 5, Oct. 2011, pp. 822–825.
- [12] E. Nigussie, J. Plosila, and J. Isoaho, "High-Speed Completion Detection for Current Sensing On-chip Interconnects," *Electron. Lett.*, vol. 45, no. 11, May 2009, pp. 547–548.
- [13] E. Nigussie, J. Plosila, and J. Isoaho, "Area Efficient Delay-Insensitive and Differential Current Sensing On-chip Interconnect," *IEEE Int. SoC Conf.*, Newport Beach, CA, USA, Sept. 17–20, 2008, pp. 143–146.
- [14] E.-J. Choi, K.-R. Cho, and J.-H. Lee, "New Data Encoding Method with a Multi-value Logic for Low Power Asynchronous Circuit Design," *Int. Symp. Multiple-Valued Logic (ISMVL'06)*, Singapore, May 2006, pp. 4.
- [15] E. Nigussie et al., "High Performance Long NoC Link Using Delay-Insensitive Current-Mode Signaling," *J. VLSI Des.*, 2007, p. 13.
- [16] T. Takahashi and T. Hanyu, "Implementation of a High-Speed Asynchronous Data-Transfer Chip Based on Multiple-valued Current Signal Multiplexing," *IEICE Trans. Electron.*, vol. E89-C, no. 11, Nov. 2006, pp. 1598–1604.
- [17] CNU-IDEC cell library data book, IDEC Chungnam National University, 1999.



Myeong-Hoon Oh received his PhD in information and communications engineering from Gwangju Institute of Science and Technology (GIST), Gwangju, Rep. of Korea in 2005. He has been with the ETRI, Daejeon, Rep. of Korea since 2005 as a principal engineer. Since 2006, he has been

an associate professor at the University of Science and Technology (UST), Daejeon, Rep. of Korea. His current research focuses on digital circuit design, embedded systems, cloud computing infrastructure, and standardization. He has also been an editor on the development of recommendations for cloud computing in ITU-T SG13.



Young Woo Kim received his BS and MS degrees and his PhD in electronics engineering from Korea University, Seoul, Rep. of Korea, in 1994, 1996, and 2001, respectively. He was an associate professor at the UST, Daejeon, Rep. of Korea. since 2009. In 2001, He joined ETRI, Daejeon,

Rep. of Korea. He has been researching asynchronous processors and working on computer systems development. His current research interests include high-speed networking and supercomputing system architectures.



Hag Young Kim received his BS and MS degrees in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea in 1983, and 1985, respectively, and his PhD in computer engineering from Chungnam National University, Daejeon, Rep. of Korea, in

2003. He joined ETRI, Daejeon, Rep. of Korea. in 1988. His current research interests include micro-servers and high-performance computing system architectures.



Young-Kyun Kim received his PhD in computer science from Chonnam National University, Gwangju, Rep. of Korea in 1995. He has been with ETRI, Daejeon, Reop of Korea, since 1995 as a principal researcher and is a managing director of the high performance computing research

group. His current research interests include high-performance computing, cloud computing, and Exabyte-scale storage systems.



Jin-Sung Kim received his BS and MS degrees and his PhD degrees in electrical engineering and computer science from Seoul National University, Rep. of Korea, in 1996, 1998, and 2009, respectively. From 1998 to 2004, and from 2009 to 2010, he was with Samsung SDI Ltd.,

Cheonan, Rep. of Korea, as a senior researcher, where he was involved in driver circuits and discharge waveform research. From 2010 to 2011, he was a post-doctoral researcher at Seoul National University. In 2011, he joined the Department of Electronic Engineering, Sun Moon University, Asan, Rep. of Korea, where he is currently an Associate Professor. His current research interests include algorithms and architectures for video compression, computer vision, and display driving systems.