Asynchronous 2-Phase Protocol Based on Ternary Encoding for On-Chip Interconnect

Myeong-Hoon Oh and Seongwoon Kim

Level-encoded dual-rail (LEDR) has been widely used in on-chip asynchronous interconnects supporting a 2-phase handshake protocol. However, it inevitably requires 2N wires for N-bit data transfers. Encoder and decoder circuits that perform an asynchronous 2-phase handshake protocol with only N wires for N-bit data transfers are presented for on-chip global interconnects. Their fundamentals are based on a ternary encoding scheme using current-mode multiple valued logics. Using 0.25 μm CMOS technologies, the maximum reduction ratio of the proposed circuits, compared with LEDR in terms of power-delay product, was measured as 39.5% at a wire length of 10 mm and data rate of 100 MHz.

Keywords: Asynchronous handshake protocol, ternary encoding, multiple-valued logic.

I. Introduction

As chip dimensions increase, the gap between the maximum length of wire for reliable data transmission during a single clock cycle and the chip side length increases [1]. This phenomenon can create a fundamental limit in the synchronous design technique, which depends on a global clock signal.

A globally asynchronous locally synchronous (GALS) system based on asynchronous global signaling is thought to be a design solution that addresses these issues most practically, and the international technology roadmap for semiconductors has predicted that circuits using asynchronous global signaling will account for 49% of the total design types by 2024 [2]. Asynchronous global signaling is based on asynchronous delay insensitive (DI) protocols, which ensure reliable data transfers regardless of the length of the interconnect.

The implementation of asynchronous DI protocols is classified into 2-phase signaling and 4-phase signaling. Four-phase signaling with specialized encoding schemes, such as dual-rail and 1-of-*n* [3], has been widely used because of its ease of implementation. However, 4-phase signaling requires 4 control signal transitions (request rising, acknowledgement rising, request falling, and acknowledgement falling) per data transfer. This sequence can cause a degradation in the overall system performance in a global interconnect assuming relatively long distance communications. On the other hand, since 2-phase signaling reduces the transitions by half, it is more effective in terms of performance and power consumption. For this reason, despite its design complexity, 2-phase signaling is recommended as an implementation method for asynchronous global interconnects [4].

Level-encoded dual-rail (LEDR) [5] is one of the encoding schemes for a 2-phase DI protocol and has a simple decoding operation. However, like other protocols, LEDR requires 2*N*-bit wires for transferring *N*-bit data on forwarding paths. This means that double the wire costs are needed in asynchronous global signaling, and an increase in the number of wires among blocks can lead to a deterioration in design complexity and power consumption for an on-chip global interconnect. The bus serialization method [6] is one of solutions for this overhead.

We present an asynchronous 2-phase DI protocol that transfers *N*-bit data with *N* wires for asynchronous global signaling. It employs ternary encoding and is implemented using multiple-valued logics (MVLs). The ternary encoding can represent 2-phase DI transfers with a minimum of three states, and the MVL can reduce the number of wires by symbolizing various data on a single wire. A current-mode MVL, which has a better noise margin than a voltage-mode MVL at a lower supply voltage, is used in our design.

Manuscript received Feb. 15, 2011; revised Mar. 7, 2011; accepted Mar. 19, 2011.

Myeong-Hoon Oh (phone: +82 42 860 1654, email: mhoonoh@etri.re.kr) and Seongwoon Kim (email: ksw@etri.re.kr) are with the Software Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.11.0211.0063

Thus far, there have been no implemented cases where the ternary encoding has been used in asynchronous 2-phase DI protocols. It is expected that, due to the overhead of encoding and decoding, little benefit can be achieved from our circuits under a general situation that does not assume the existence of wires. However, a reduced number of wires and the characteristics of a current-mode MVL can produce conflicting results with an on-chip global interconnect supposing relative long wires.

Our goal is to verify the effectiveness of our protocol as one of the implementations for asynchronous global signaling by comparing it with LEDR in terms of power-delay product (PDP), while considering both latency and power consumption.

II. Structure

In our 2-phase ternary encoding scheme, a high level logic and a low level logic are encoded to binary logics ‘1’ and ‘0’, respectively. A middle level logic represents a binary logic ‘1’ or ‘0’ if a previous ternary logic is at a high level or low level, respectively, as in Figs. 1(a) and (b). The state assignment of ($Q1$, $Q0$) is explained in Fig. 3.

The external interface of the suggested protocol assumes a 2-phase bundled data protocol [3]. In Fig. 2, which shows the environment of the protocol for N-bit data transfers, an encoder block converts a voltage-valued request signal (*req_in*) and a data signal (*data_in*) into one of three current levels on a wire named I_{out}. The delivered current value I_{in} is recovered to the original voltage-valued request (*req_out*) and data signal (*data_out*) at the decoder block. Note that, unlike other 2-phase DI protocols, the number of wires between the encoder and decoder is not $2N$ but N.

Since the validity of transferring data is determined by the timing of both the rising and falling edges of a request signal (*req_in* in Fig. 2) in the 2-phase protocol, a finite-state machine of the ternary encoding scheme can be implemented with double-edge-triggered flip-flops. Encoded 2-bit values ($Q1$, $Q0$) of the three states are depicted in Fig. 1(a). Whenever *req_in* changes, a state transition occurs according to the data signal (*data_in* in Fig. 2). High and low levels are encoded to transfers of data ‘1’ and data ‘0’, respectively. A middle level preserves the data of the previous transfer cycle. From the state transition table of the state diagram, we obtained equations for inputs ($D0$, $D1$) of the flip-flops corresponding to $Q0$ and $Q1$. The equations are $D0 = \overline{Q0}(\overline{Q1}\;\overline{data\_in} + Q1\; data\_in)$ and $D1 = \overline{Q0}\; data\_in.$ A possible implementation based on

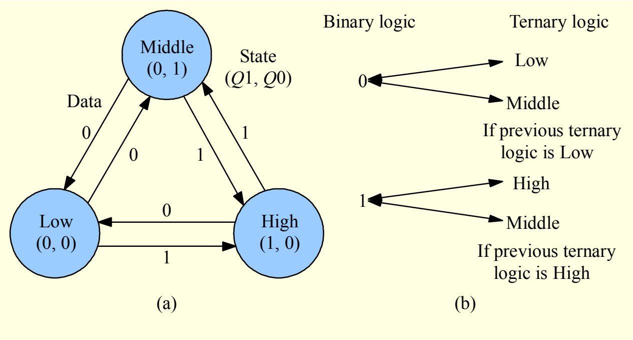

Fig. 1. Conversion between binary and ternary logics: (a) state diagram and (b) mapping relationship.

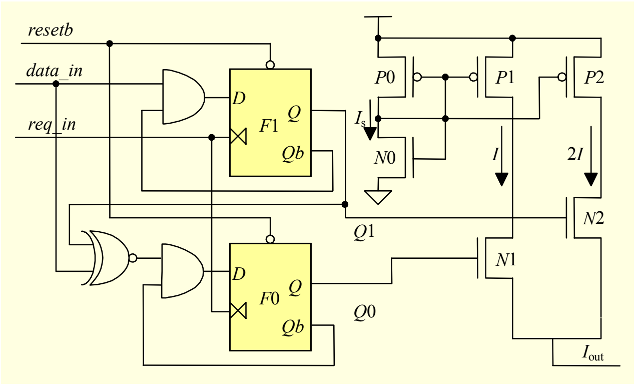

Fig. 3. Encoder circuits.

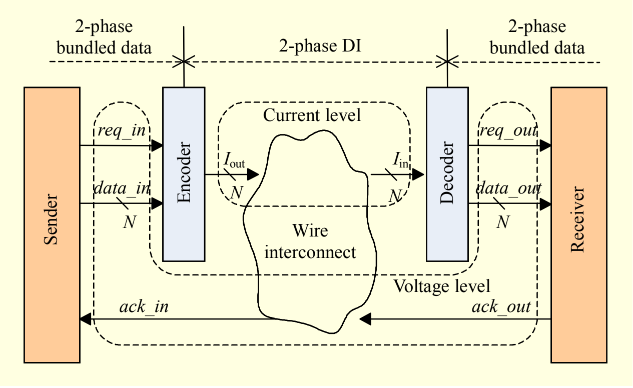

Fig. 2. Environment of suggested protocol.

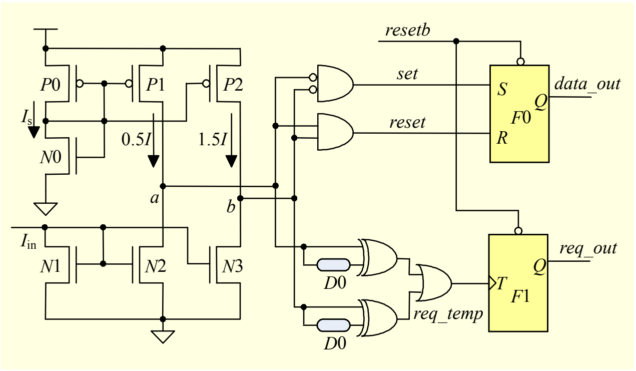

Fig. 4. Decoder circuit.

Table 1. Recovered data and mapping relation between node (a, b) values and input current.

Input current	Value of (a, b)	Data
0	(1, 1)	0
I	(0, 1)	Previous data
$2I$	(0, 0)	1

the state diagram is illustrated on the left side of Fig. 3.

In the current-mode circuitry on the right side of Fig. 3, constant current I_s from current source ($P0$, $N0$) is duplicated at the drains of $P1$ and $P2$, which are working as a current mirror. Two binary logic signals, $Q0$ and $Q1$, determine one of the encoded current levels, 0 (low), I (middle), or $2I$ (high), by switching pass transistors $N1$ and $N2$. Following the current mapping in Fig. 1(a), the current through $N2$ should be set to a size roughly twice as large as that flowing through $N1$ by varying the size of $P2$ and $P1$.

A schematic of the decoder is illustrated in Fig. 4. The three-valued input current I_{in} is copied to the drains of $N2$ and $N3$, which comprise a current comparator jointly coupling with $P1$ and $P2$. The current mirror with $P1$ and $P2$ generates the threshold current, $0.5I$ and $1.5I$, and a differential current between the threshold current and I_{in} is generated in the current comparator. Depending on the differential current, nodes labeled a and b at the $N2$ and $N3$ drains, respectively, take a logical '0' or '1' value. Table 1 lists the logical voltage values at node (a, b), along with the data that would be recovered, according to each current level.

The original data signal (*data_out*) is reconstructed easily using an SR latch ($F0$) and a combination of a and b as depicted in Fig. 4. To recover the request signal (*req_out*) of the 2-phase protocol, the signal's rising and falling edges should alternate whenever a new input current arrives. To deal with this, the decoder generates a pulse signal (*req_temp*), the width of which is as wide as the time of delay element ($D0$). The pulse signal always goes to logic '1' when any changes in a and b are detected, and finally, toggles the signal *req_out* through a T-flip-flop labeled $F1$ in Fig. 4.

When the input current varies from 0 to $2I$ and vice versa, there may be a very small time difference between the changes of a and b in real circuits. This can cause the signal *req_temp* to be asserted twice, and consequently, the signal *req_out* creates a malfunction of the handshake protocols. Another roll of delay element, $D0$, is preventing the occurrence of this situation by filtering the time difference.

III. Implementation and Simulation

The implementation of the encoder and decoder circuits was realized using 0.25 μm CMOS technology. The supply voltage was 2.5 V and 2 V for standard cells and current-mode circuits, respectively, to save power consumption. Reference current I was set to 96 μA through the optimization method described in [7]. The wire for the simulation is based on a distributed RC model, and the parameters are referenced from the third metal layer in ANAM 0.25 μm technology.

Area overhead in terms of the number of transistors exists in the implementation of our circuits. A total of 176 and 90 transistors are used for ternary circuits and LEDR, respectively. However, this difference is ignorable, considering that a locally synchronous module or an IP normally comprises millions of transistors. Instead, the benefit can be achieved by reducing the

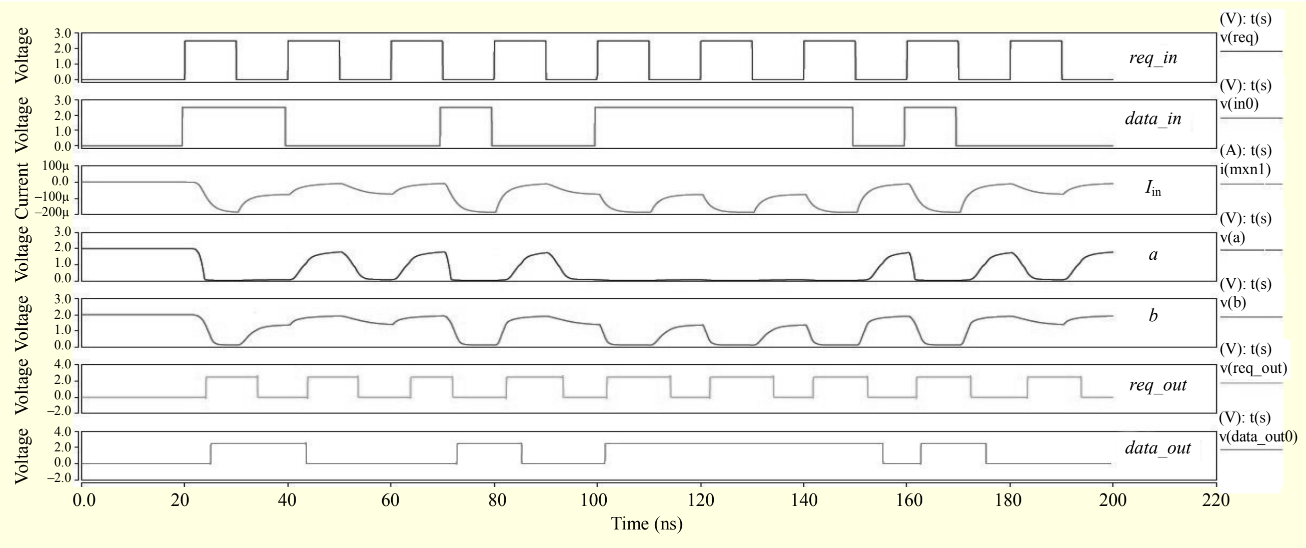

Fig. 5. Example of waveform of the HSPICE simulation with wire length of 10 mm.

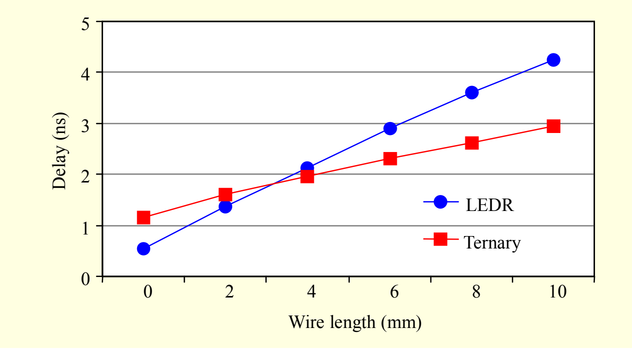

Fig. 6. Delay measurement results.

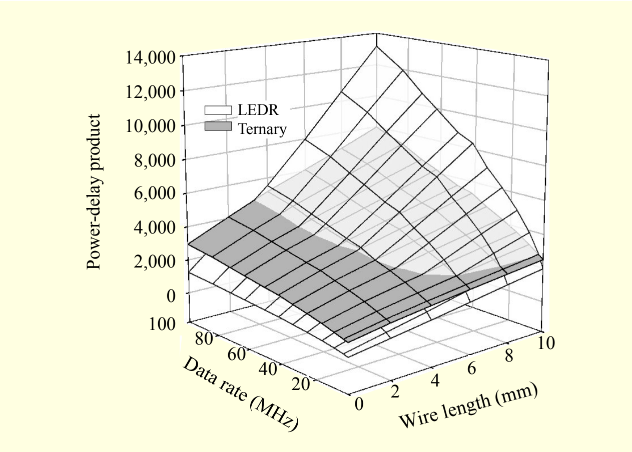

Fig. 7. Comparison of power-delay products.

number of required wires.

An example waveform of the HSPICE simulation with a wire length of 10 mm is depicted in Fig. 5. The original input signals of *req_in* and *data_in* are safely recovered into *req_out* and *data_out* by converting between the voltage and current levels.

Figure 6 shows the HSPICE simulation latency results for 2-bit data transfers from the time before a conversion to the time after reconstruction. Our circuits meet the DI characteristics by transferring data safely regardless of the wire length. As the graph indicates, the slope degree for the delay in the ternary case is lower because the current-mode circuit, which generates a comparatively constant amount of current, is less affected by the wire length. This is mainly due to the much larger resistance of the current mirror of the encoder in the current-mode circuit, as compared to the equivalent resistance of the encoder in the voltage-mode circuit. Consequently, the suggested circuits perform faster transfers with a wire length of over 3 mm.

To evaluate the effectiveness of our design over LDER in terms of both power consumption and transmission speed, we used the metric figure of the PDP value. Figure 7 depicts the measurement of PDP of two methods for 2-bit data transfers with various wire lengths and data rates of 2.5 MHz to 100 MHz considering the turnaround time of the simulation. We used two patterns composed of 1,000 randomly generated 1's and 0's and assigned each pattern to *data_in*[0 bit] and *data_in*[1 bit] in Fig. 2. Finally, we simulated our circuits using five sets of newly generated patterns and obtained the average results. The ternary case consumes more PDP value with a lower data rate and shorter wire length, but is superior to LEDR over a wire length of 3 mm and data rate of 62.5 MHz. The longer the wire becomes, the more the PDP value is affected by the wire itself compared to the encoder and the decoder circuit. Therefore, the proposed method, which uses half the amount of wires, can save more PDP value than LEDR with a relatively longer wire. In the simulation, the maximum ratio of the PDP value reduction of the proposed ternary circuits to the LEDR is 39.5% at a wire length of 10 mm and data rate of 100 MHz.

IV. Conclusion

We suggested a 2-phase DI protocol for asynchronous global signaling based on ternary encodings and implemented it using current-mode multiple-valued logics. The protocol can reduce the number of wires by half in comparison with a conventional 2-phase DI protocol such as LEDR. With the simulation using 0.25 μm CMOS technology, the results indicated that both the latency and power-delay product value of the suggested protocol were lower than those of LEDR over a wire length of 3 mm and data rate of 62.5 MHz.

References

[1] D. Sylverster and K. Keutzer, "A Global Wiring Paradigm for Deep Submicron Design," *IEEE Trans. CAD Integr. Circuits Syst.*, vol. 19, no. 2, Feb. 2000, pp. 242-252.

[2] *International Technology Roadmap for Semiconductors*, Semiconductor Industry Association, 2009.

[3] J. Sparsø and S.B. Furber, *Principles of Asynchronous Circuit Design: A System Perspective*, Kluwer Academic Publishers, 2001.

[4] 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.

[5] M.E. Dean, T.E. Williams, and D.L. Dill, "Efficient Self-Timing with Level-Encoded 2-Phase Dual-Rail (LEDR)," *Adv. Research VLSI*, UC Santa Cruz, 1991, pp. 55-70.

[6] J.S. Lee, "On-Chip Bus Serialization Method for Low-Power Communications," *ETRI J.*, vol. 32, no. 4, Aug. 2010, pp. 540-547.

[7] M.H. Oh and D.S. Har, "Low Delay-Power Product Current-Mode Multiple Valued Logic for Delay-Insensitive Data Transfer Mechanism," *IEICE Trans. Fundamentals*, vol. E88-A, no. 5, May 2005, pp. 1379-1383.