

# New Ternary Data Encoding for Delay-Insensitive Asynchronous Design

Je-Hoon Lee<sup>1\*</sup> and Won-Ki Sung<sup>1</sup>

<sup>1</sup>*Div. of Electronics, Information, and Information Eng., Kangwon National University, 1 Joongang-ro, Samcheock, Gangwon, 245-711, Rep. Of Korea*

*sungwk@kangwon.ac.kr*

## Abstract

*A conventional B-ternary logic for an asynchronous design has many drawbacks, most notably its incomplete truth table of the basic logic gates. This paper presents a new asynchronous ternary logics based on a new data-encoding scheme. The main aim of this research is to provide the flawless truth table for varying logic gates. The asynchronous circuit employing these logics uses two-rail logic for two data bits. It can reduce the number of wire by half comparing to the dual-rail or 1-of-4 data encoding. Furthermore, the proposed encoding scheme reduces switching by 25% to compare with the conventional B-ternary one.*

**Keywords:** Ternary, multi-value logic, data encoding, asynchronous

## 1. Introduction

Recently, low-power consumption and high-speed design is being the major design issue in deep sub-micron ASIC designs. Most of these ASICs are synchronous and their activities are controlled by a global clock which is triggered simultaneously. The designers focus on the data processing by assuming the existence of a global clock even though this timing assumption has questionable accuracy [1]. Furthermore, ever increasing demand for high speed clock signal that has the longest wire delay in the ASIC causes many signal integrity challenges and time closure problems [2].

An asynchronous design is free from those problems because it employs local handshaking, not global clock. It allows a large number of clock signals to be replaced by the local handshaking using two control signals, request and acknowledge between the neighboring circuitries. Asynchronous circuit is free to the clock-related problem such as clock-skew because it uses the handshake protocol instead of the system clock. In addition, because the asynchronous circuit only drives a module that activating is required, it does not waste the system power to activate the block has no work. Thus, it significantly reduces the total power consumption and greatly increases the system performance. Consequently, asynchronous design has many potential advantages over synchronous one such as no clock skew, low-power consumption, easy of global timing issues.

When we design asynchronous logic, the delay model of an element and wiring delay is very important to generate a completion signal acting like a clock signal. The delay model that adapt to the asynchronous circuit is categorized as the bounded delay model and the DI (delay insensitive) model. A DI delay model is assumed that both wire and gate delays are unbounded. In the bounded delay model, we mainly use the single-rail data encoding scheme.

---

\* J. H. Lee and W. Sung, "New ternary data encoding for asynchronous design"

To adapt this delay model, the designer has required the precise delay for each block to generate a completion signal. However, the DI (delay insensitive) model employs a dual-rail data encoding to generate a completion signal. It needs not precise simulation after executing layout. However, the dual-rail encoding makes complex circuit and requires more signal transition. Though the dual-rail circuit enables to adapt the DI delay model, a new research is required reducing the number of signal transition and the circuit area [4-7]. In particular, we deal with SI (speed-independent) delay model, which assumed that while gate delay is unbounded, each wire delay is ignored [3].

A conventional asynchronous circuit employing DI delay model transfers the data using with two or four-phase handshaking protocol [4]. Instead the bounded-delay model assumes that the data is valid when the request signal is asserted that is obtained from a matched delay, a DI asynchronous circuits do not need to know the precise delay of each module. On the other hand, they require a completion signal obtained from the data encoding method. Several data encoding schemes have been introduced for DI asynchronous circuits such as a dual-rail, 1-of-4, B-ternary data encoding as shown in Figure 1 [4]. These encoding schemes either need double data lines or require more switching activities than the bounded delay model.

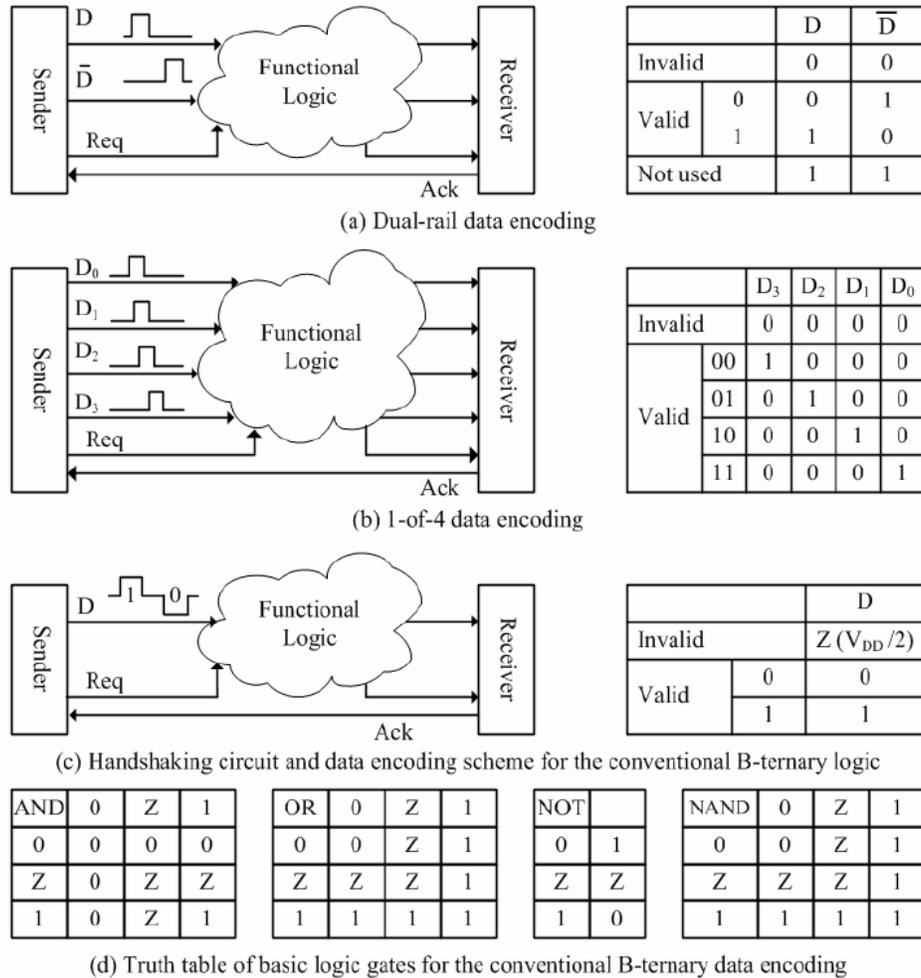
In particular, a B-ternary encoding employs multi-valued logic so as to reduce the number of data lines, as shown in Figure 1(c) [5]. This B-ternary data encoding exploits 0 and 1 for single data bit, and Z that represents  $V_{dd}/2$  for spacer as shown in Figure 1(c). Thus, it can reduce the number of data lines by half comparing to the other data encoding methods. However, a B-ternary data encoding has two major problems such as the number of switching activities and the propagation delay owing to the incomplete truth table for the basic logic gates as shown in Figure 1(d).

In this paper, we propose new ternary data encoding scheme so as to solve these problems. Furthermore, we provide the complete truth table for the basic logic gates such as AND, OR, NAND, and NOT to prevent the propagation delay caused by the incomplete truth table of B-ternary data encoding.

## 2. The Proposed Ternary Data Encoding

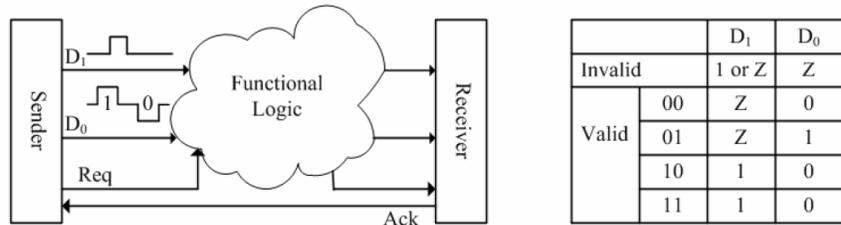
The proposed ternary data encoding uses multi-value logic so as to provide an efficient way related with the number of wires because it employs multi-value logics. A binary logic uses two different logical values such as 0 and 1. However, a ternary logic uses three distinct logical values that consist of 0, Z, and 1. Both conventional B-ternary and the proposed ternary data encoding use this multi-value logic. As shown in Figure 2, the dual-rail and 1-of-4 encoding scheme need  $2n$ -wires to move  $n$ -bit information even though 1-of-4 encoding can save 50% of switching activities comparing to the dual-rail data encoding. However, the B-ternary and proposed ternary encoding scheme require  $n$  wires for  $n$ -bit data transferring.

An asynchronous circuit employing the proposed ternary data encoding represents the pair of two data bits as shown in Figure 2(a). The proposed encoding scheme expresses two data bits 00, 01, 10, and 11 as Z0, Z1, 10, and 11, respectively. In addition, ZZ and 1Z are used for spacer that represents the invalid data. We can reduce the switching activities by restricting the MSB to 1 and Z without 0. Each module communicates with other modules on a promised 2 or 4-phase handshaking protocol.

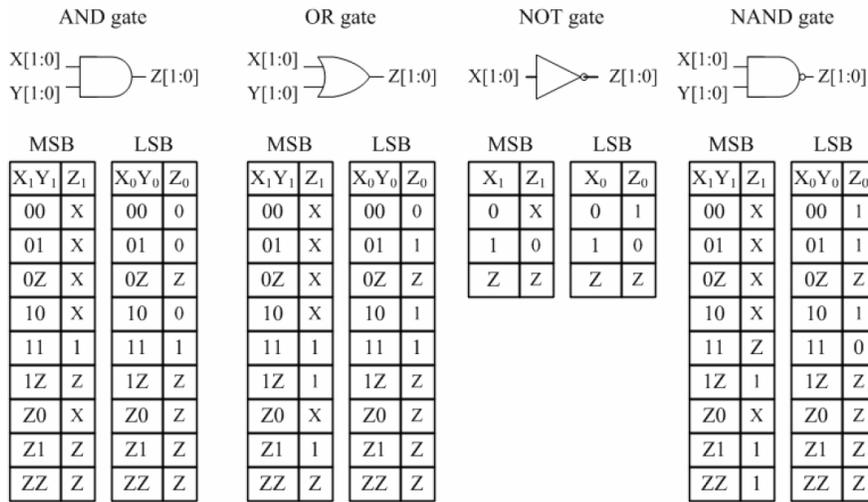


**Figure 1. Comparison of the typical data encodings such as (a) dual-rail, (b) 1-of-4, (c) B-ternary data encoding, and (d) example of truth table of basic logic gates for B-ternary data encoding**

A handshaking model for the proposed ternary logic is shown in Figure 2(a). The asynchronous circuits employing both data encodings use request signal, *Req*, and acknowledge signal, *Ack*, for handshaking protocol as shown in Figure 2(a). The handshaking protocol is performed as follows. First, the sender issues a valid codeword based on the proposed ternary data encoding, and then it transmits the request signal to high. Second, the receiver confirms all transferring codeword is valid, and sets acknowledge signal high. Third, the sender responds by transferring invalid data, spacer, and transmits request signal to low. Finally, the receiver acknowledges this by transiting acknowledge signal to low.



(a) Handshaking circuit and data encoding scheme for the proposed ternary logic



(b) Truth table of MSB and LSB for basic logic gates in the proposed ternary data encoding

<b>AND</b>	Z0	Z1	10	11	ZZ	<b>OR</b>	Z0	Z1	10	11	ZZ	<b>NOT</b>		<b>NAND</b>	Z0	Z1	10	11	ZZ			
	Z0	Z0	Z0	Z0	ZZ		Z0	Z0	Z1	10	11	ZZ		Z0	11	Z0	11	11	11	ZZ		
	Z1	Z0	Z1	ZZ	Z1	ZZ		Z1	Z1	Z1	11	11	ZZ		Z1	10	Z1	11	10	ZZ		
	10	Z0	ZZ	10	10	ZZ		10	10	11	10	11	ZZ		10	11	11	Z1	Z1	ZZ		
	11	Z0	Z1	10	11	ZZ		11	11	11	11	11	ZZ		11	Z0	11	11	10	Z1	Z0	ZZ
	ZZ	ZZ	ZZ	ZZ	ZZ	ZZ		ZZ	ZZ	ZZ	ZZ	ZZ	ZZ		ZZ							

(c) Truth table of basic logic gates for the proposed ternary data encoding

**Figure 2. The proposed ternary data encoding: (a) handshaking circuit and data encoding scheme, (b) Truth tables for MSB and LSB according to the kind of logic gates, (c) the operation results for AND, OR, NOT, and NAND operation**

The proposed ternary data encoding uses a group of two wires to transmit two bits of information of symbol concurrently. There is one timing restriction during data transmission. A sender is in spacer by transferring ZZ should transfer 1Z prior to 11 and 10 when it wants to transfer valid data 10 and 11, respectively. Since the symbol of ZZ and 1Z represent an invalid data, spacer, for handshaking protocol, the proposed ternary encoding meets the timing assumption of SI delay model. Instead the logic value of LSB of the proposed ternary data encoding consists of 0, 1, and Z, the logic value of MSB consists of just 1 and Z without 0. This can reduce the number of switching activities in MSB by half during data transferring and processing in asynchronous circuit. For example, it requires only one half-swing when the valid data, Z0 or Z1 are transferred and returned to spacer, ZZ. In addition, it requires two half-swings when the valid data, 10 or 11 are transferred and returned to spacer, ZZ.

In B-ternary logic, 0 and 1 represent valid data bit 0 and 1, respectively. In addition, Z represents the spacer that is invalid data bit. Thus, a B-ternary data encoding need 2 half

swings to move 2-bit information as shown in Figure 1(c) since it transfers the valid data bits and returns to spacer. On the other hand, the proposed ternary data encoding need 1 or 2 half swings to move 2-bit information since MSB consists of 1 and Z as shown in Figure 2(a). This shows the reduced switching activity during state change, from valid to invalid state. Thus, the proposed encoding scheme reduces switching by 25% comparing with the conventional B-ternary one.

### 3. Truth Table for Implementing the Proposed Ternary Gates

The dynamic power dissipation to drive the data line with a full swing is given

$$P_{dynamic} = C_L \cdot V_{DD} \cdot V_{swing} \cdot f_c \quad (1)$$

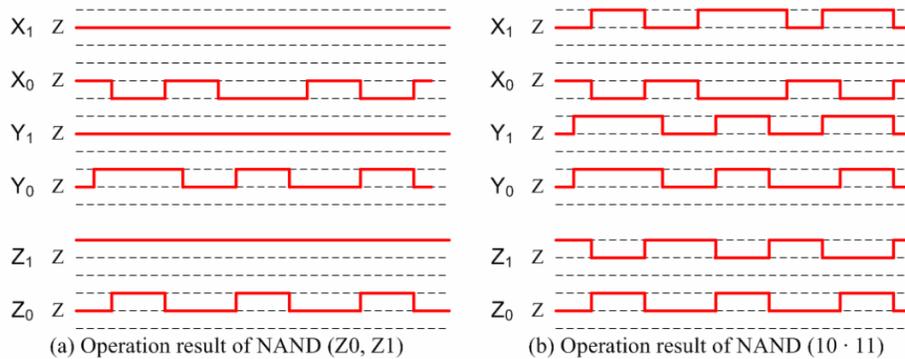
where  $C_L$  is a load capacitance and  $f_c$  is the frequency for switching. This means that the potential power saving of the ternary encoding scheme over the dual-rail encoding scheme is 50%. In addition, it certainly reduces the required data lines by half. Furthermore, the potential power saving of the proposed ternary encoding scheme over the conventional ternary encoding scheme is about 25% when the sender transfers the 8 bit data.

The other advantage of the proposed ternary data encoding provides the flawless truth table. A conventional B-ternary logic uses the comparators for two data inputs, X and Y so as to implement basic logic gates such as AND, OR, NAND, and NOT as shown in Figure 1(d). In B-ternary logic, binomial operation AND, OR and unary operation NOT are defined as:  $X \cdot Y = \min(X, Y)$ ,  $X + Y = \max(X, Y)$ , and  $\sim(X) = 1 - X$ , respectively. Even though these definitions are efficient to implement logic gates without area overhead, there are uncertain results in the operation between the valid data and space. In example, the result of 0 AND Z is 0 but the result of 1 AND Z is Z. This fact can induce both the inaccurate operation and the propagation delay according to increasing the length of datapath.

The proposed ternary logic gates such as AND, OR, NOT, and NAND are to be designed which perform the given operation with two 2-bit data as shown in Figure 2(c). All of them provide the separate operation for MSB and LSB of two input data. In MSB operation, the input combinations 00, 01, 0Z, and Z0 do not represent valid ternary data and will never occur, thus, X in the output, Z1 is don't care condition for these combinations. The logic for MSB operation can be implemented without comparators for detecting three different logic values since it has only two values, 1 and Z. On the other hand, the logic for LSB operation requires two comparators to identify three logic values for the input data.

The proposed ternary encoding expresses 2-bit data symbol 00 as Z0, 01 as Z1, 10 as 10, and 11 as 11. As shown in Figure 2(c), all logic gates for the proposed ternary data encoding such as AND, OR, NOT, and NAND can be implemented based on truth table of MSB and LSB operation as shown in Figure 2(b). Note that all outputs of these gates meet the definition of the proposed ternary data encoding.

The timing simulation for the proposed NAND gate is shown in Figure 3. Two input symbols have two-bit data, X and Y that are fed into NAND gate and output the symbol, Z. There are two examples; Z0 and Z1 and 10 and 11 with different wire delay as shown in Figure 3(a) and Figure 3(b), respectively. These results show that the ternary NAND gate can output the results that are encoded by the proposed ternary encoding scheme with regardless of the input delays. The completion signal can be obtained from the output of NAND gate. Even though this NAND gate need 2 additional comparators for LSB operation comparing to the typical B-ternary NAND gate, it provides the flawless truth table and requires small number of switching activities. Thus, it is applicable for asynchronous circuit employing SI delay model for low-energy consumption.



**Figure 3. Example of timing simulation of the proposed 2-input ternary NAND gate**

Table 1 shows the features of several kinds of encoding scheme such as conventional dual-rail, 1-of-4, conventional ternary, and the proposed RZ/NRZ mixture ternary, *etc.* The proposed ternary data encoding can use one wire for transferring one valid data and it is same with that of conventional ternary data encoding. However, the proposed ternary data encoding method can reduce the number of transitions and it can save the 25% signal transition on average compared to the conventional ternary one.

**Table 1. Comparison of DI encoding methods**

	Number of wire per bit	Energy (Transition / bit)
Conventional Dual-Rail	2	2
Dual-Rail (RZ/NRZ)	2	1.5 (average)
1-of-4	2	1
Conventional Ternary	1	0.5
Proposed Ternary	1	0.375 (average)

#### 4. Conclusions

This paper proposes a new ternary data encoding scheme for asynchronous circuit employing SI delay model. It requires N wires to transfer N bits of information as same as the conventional B-ternary encoding. The proposed ternary encoding can reduce the number of switching activities in asynchronous circuit by 25% comparing to B-ternary encoding by restricting the logic value of MSB for each symbol to 1 and Z without 0. In addition, it provides the flawless truth table for designing basic logic gates so as to avoid the propagation delay of B-ternary logic. It helps to reduce power consumption of asynchronous circuits without significant area overhead. From the simulation result, the completion of a computation is successfully detected by detecting spacer. Design of basic gate meets all mandatory timing requirements for SI delay model. Thus, it is applicable for asynchronous circuit with respect to the circuit area and power consumption.

## References

- [1] J. Spaso and S. Furber, "Principles of Asynchronous Circuit Design: A System Perspective", (2001).
- [2] J. H. Lee, Y. H. Kim and K. R. Cho, "Design of a fast asynchronous embedded CISC microprocessor, A8051", IEICE Trans. on Electronics, vol. E87-C, no. 4, (2004), pp. 527-534.
- [3] Y. Nagata and M. Mukaidono, "B-Ternary asynchronous digital system under relativity delay", IEICE Trans. on Information and Systems, vol. E86-D, no. 5, (2003), pp. 910-919.
- [4] J. Bainbridge and S. Furber, "Delay insensitive system-on-chip interconnect using 1-of-4 data encoding", in Proc. of ASYNC 2001, (2001), pp. 118-126.
- [5] R. Mariani, R. Roncella, R. Saletti and P. Terreni, "On the realization of delay-insensitive asynchronous circuits with CMOS ternary logic", in Proc. of ASYNC 1997, (1997), pp. 54-62.

