A Two-bit Bus-Invert Coding Scheme With a Mid-level State Bus-Line for Low Power VLSI Design

Myungchul Yoon

Abstract—**A new bus-invert coding circuit, called Two-bit Bus-Invert Coding (TBIC) is presented. TBIC partitions a bus into a set of two-bit sub-buses, and applies the bus-invert (BI) algorithm to each sub-bus. Unlike ordinary BI circuits using invert-lines, TBIC does not use an invert-line, so that it sends coding information through a bus-line. To transmit 3-bit information with 2 bus-lines, TBIC allows one bus-line to have a mid-level state, called M-state. TBIC increases the performance of BI algorithm, by suppressing the generation of overhead transitions. TBIC reduces bus transitions by about 45.7%, which is 83% greater than the maximum achievable performance of ordinary BI with invert-lines.**

Index Terms—**Bus-invert coding, low-power design, low-power bus, VLSI digital circuits, performance analysis**

I. INTRODUCTION

Low-power design is one of the hottest issues in VLSI design, especially for VLSI chips for mobile devices. Although the low-power design of circuits and functional modules is essential to reduce the power consumption of VLSI chips, the design of low-power bus is not less important than that of the circuits and modules, because a substantial part of the total power is dissipated by the buses. Efforts for low power bus design are carried out, either to decrease the dynamic power per activation, or to reduce the number of bus-activations.

Bus-Invert (BI) [1] coding is one of the well-known techniques that reduce the number of bus-transitions. BI is a simple coding that reduces the transitions of bus-lines in the following algorithm; If sending a datum activates more than half of the bus-lines, BI transmits its complement. The coding information, called *inv*-bit which indicates the inversion state of the transmitted data, is transmitted simultaneously with the datum.

Because of its simplicity and usability, many enhanced BI algorithms [2-7] have been developed. Although many variations of BI algorithm have been presented, almost all of these algorithms have used an auxiliary line, called invert-line, to send an *inv*-bit. However, the invert-line has two major drawbacks. The first is the increase of bus-areas due to the additional line. The second is the performance degradation due to its transitions.

Some activation of the invert-lines or bus-lines occurs in sending the *inv.* These activations are the overhead of BI coding that should be paid to reduce the transitions of the bus-lines. However, this overhead transition (OT) is the major factor reducing the performance of BI circuits. It is known that the invert-line generates many OTs, so that it significantly degrades the performance of BI.

To remove the invert-lines in implementation of the BI circuit, Selectively Activated Flip-Driver (SAFD) [8, 9] sends the *inv* through the bus by using a special bus-driver called flip-driver. SAFD increases the performance by effectively suppressing the generation of OTs, so that it can reduce bus transitions by 35%.

According to a theoretical analysis [9, 10], BIC can reduce bus transitions by up to 50% for independent data, if no OT is generated. The ordinary BIC with invert-lines, however, can reduce transitions by a maximum of 25%

Manuscript received Mar. 1, 2014; accepted Jul. 26, 2014
The author is with the Department of Electronics Engineering, Dankook University, Cheon-An, Choong Nam, 330-714, Korea
E-mail : myoon@dankook.ac.kr

because of OTs generated by invert-lines.

A new BI implementation scheme, called Two-bit Bus-Invert Coding (TBIC) is presented in this paper. TBIC divides an *n*-bit bus into *n*/2 sub-buses of width 2, and BI coding is independently applied to each sub-bus. TBIC transmits *inv* through a bus-line to avoid the increase of bus-width. Furthermore, it is devised to generate OTs that are as small as possible. For this purpose, TBIC introduces an intermediate state, called M, between the H(1) and L(0) states. With a new state-transition rule among the three states, TBIC can send 3-bit of information through two bus-lines with few OTs. By this improvement in implementation, TBIC can reduces bus transitions by about 45.7%.

II. TWO-BIT BUS-INVERT CODING WITH A MID-LEVEL STATE BUS-LINE

According to the theoretical performance analysis [9], in general, the performance of BIC decreases with the increase of bus-width. Therefore, a bus with large bus-width needs to be partitioned into several sub-buses with narrower bus-width. For the transmission of independent data, the maximum achievable reduction ratio of BIC is 50%, when a bus is partitioned into a set of two-bit sub-buses, on the condition that the coding circuit does not generate OTs [9]. When a bus is partitioned into two-bit sub-buses, the ordinary BIC which uses the invert-line can also get its maximum performance. However, it requires 50% increase of bus-width, and can only get 25% of reduction ratio which is only half of the maximum achievable value.

The Two-Bit Bus-Invert Coding (TBIC) with a mid-level state bus-line scheme is developed to increase the reduction ratio as much as possible, and to get rid of the problems of the invert-line. To maximize the performance of TBIC, TBIC deals with a two-bit bus so that it also partitions a bus into a set of two-bit sub-buses, and independently applies TBIC algorithms to each sub-bus. To avoid the increase of buswidth, TBIC does not use the invert-lines. Instead, it uses one of the bus-lines to send the *inv*-bit. In addition, TBIC is intended to minimize the overhead transitions of the bus-lines in sending the coding information. For this purpose, TBIC introduces a mid-level state to the line carrying *inv*-bit.

Fig. 1 shows the structure of TBIC. An 2*k*-bit transmission bus is composed of *k* TBICs. Each TBIC transmits only two bits of 2*k*-bit data and works independently of the other. For odd-line buses, one bus-line is omitted in coding. The two lines of the bus in a TBIC work differently. One of the two lines, named N-line, works just as a normal bus-line, i.e. it transfers between H-state(V_{DD}) and L-state (0). The other line, named M-line, can have one additional state, called M-state. The voltage level of M-state is at about the middle of V_{DD} and 0, but the exact voltage level is not important. By these two bus-lines, the TBIC can send and receive the three bits of information, i.e., two data bits, and one bit of coding information (*inv*), as follows.

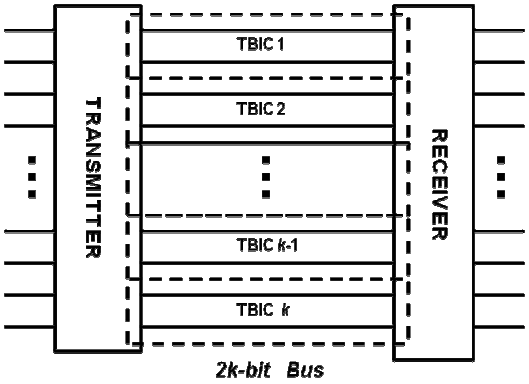


Fig. 1. The structure of TBIC.

1. Encoder Circuits

Because only two bits of a datum are involved in TBIC, the decision logic is simple: inverted transmission occurs only when both of the two bits are different from the current values of the corresponding bus-lines. The inversion information, *inv*, should be sent through the two bus-lines in the same cycle. Eight different states are required to transmit three bits, but only six stable states are possible with an N-line and an M-line. To get eight different states, TBIC uses the history (sequence) of two bus-lines.

Fig. 2(a) shows the encoder circuit of TBIC. It has a 3-bit register, and the entries of the register are named R0, R1, and RM. R0 and R1 store the values to be transmitted through the bus-lines B0 and B1 (let B0 be M-line, and B1 be N-line). RM is used to control the transmission value of M-line. If RM=1, the M-line transits to M-state, otherwise, the value in R0 is

Table 1. Truth table for TBIC encoder

RM	D0	D1	R0+	R1+	RM+	inv	B0*	B1**
0	R0	R1	R0	R1	0	0	R0	R1
0	R0	R1/	R0	R1/	0	0	R0	R1/
0	R0/	R1	R0/	R1	0	0	R0/	R1
0	R0/	R1/	R0	R1	1	1	M	R1
1	R0	R1	R0	R1	0	0	R0	R1
1	R0	R1/	R0	R1/	1	0	M	R1/
1	R0/	R1	R0/	R1	0	0	R0/	R1
1	R0/	R1/	R0	R1	1	1	M	R1

*B0: M-LINE, **B1: N-LINE

transmitted through the M-line.

Table 1 shows the truth table for *inv*, and the next values of the registers. The truth table is designed to minimize the transition of bus-lines. According to the table, no more than one bus-line changes at any transmission cycle. The *inv*, and the next values of the register are determined by the following logic functions:

$$inv = (D0 \oplus R0) \cdot (D1 \oplus R1)$$
$$R0+ = inv \oplus D0$$
$$R1+ = inv \oplus D1$$
$$RM+ = inv + RM \cdot (D1 \oplus R1)$$

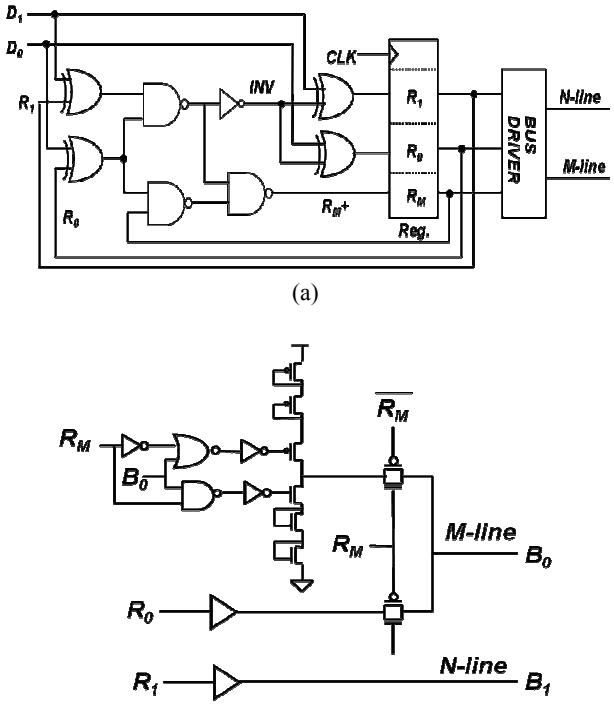
Fig. 2(b) shows a simple example circuit for the bus-driver. M-line is driven by a normal bus-driver when RM=0, while it is driven by Mid-level generator circuit when RM=1.

2. Decoder Circuits

Fig. 3(a) shows the decoder circuit for TBIC. The decoder also has a 3-bit register, of which the entries are R0, R1, and RM. Initially, the RM is reset to 0, and the values of M-line (B0) and N-line (B1) are stored at R0, and R1 of the register, respectively. The inversion state of the received data is decided by the values of the bus-lines and the register.

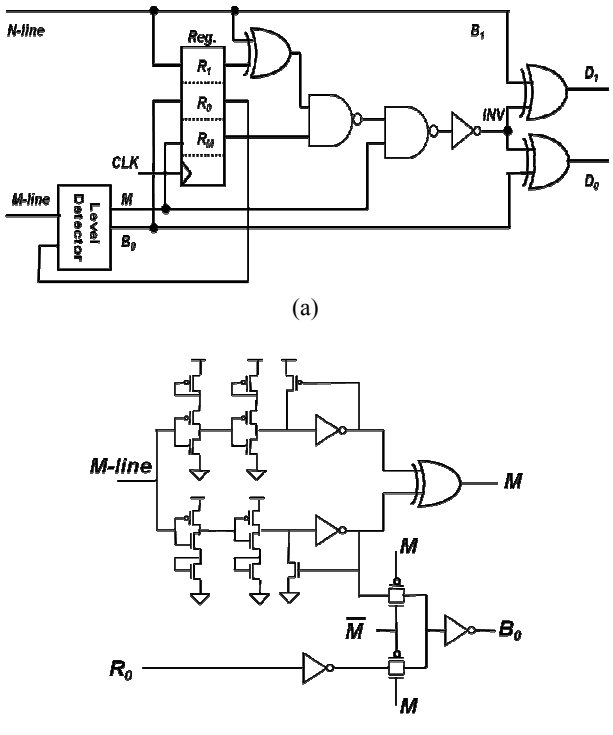
At first, the level detector checks the voltage level of M-line to determine the value of M and B0. If the M-line is at mid-level, M is set to 1, and B0 is set to the value of R0. Otherwise, M is reset to 0, and B0 becomes the value of the M-line.

The inversion state of the received bits is determined by the result of the level detector and the previous bus states stored in the register. If M=0, *inv*=0. When M=1,



(a) (b)

Fig. 2. Encoder of TBIC (a) encoder circuit, (b) an example for the bus driver circuit.



(a) (b)

Fig. 3. Decoder of TBIC (a) decoder circuit, (b) an example for level detector circuit (V_{DD}=1.2V, V_{TN}=|V_{TP}|≈1/4 V_{DD}).

inv=1 if either RM=0 or B1=R1. The logic function for *inv* is given by

$$inv = M\cdot RM' + M\cdot RM\cdot (B1 \oplus R1)' = M\cdot [RM' + (B1 \oplus R1)']$$

The results of the level detector and the value of N-line are stored in the register for decoding in the next cycle.

Fig. 3(b) is an example of the mid-level detector circuit corresponding to the mid-level generator in Fig. 2(b). The transient point of the first inverter on the upper path is set lower than the mid-level voltage while that on the lower path is set higher than the voltage. Therefore, the outputs of the upper path and the lower path are different from each other when the M-line is at mid-level voltage, so that M becomes 1. If M-line is at 1 or 0 state, both the outputs have the same value so that M becomes 0. When M is 1, the value stored in R0 is used as the received bit.

The circuits in Figs. 2(b) and 3(b) are just example circuits to show the operation of the mid-level generator and level detector. For high speed applications, a faster mid-level generator, and/or a more sensitive level detector are required. As it will be described in the next section, the exact voltage level of M-stage is not critical in operation although it affects the power consumption of overhead transitions. Using this property, various pairs of a mid-level generator and a level-detector are possible. The mid-level generator circuit and corresponding level-detector circuit in decoder should be designed together by considering the speed, power, size, etc.

III. DYNAMIC BUS POWER WITH TBIC

Because of the existence of M-state transitions, counting the number of overhead transitions of TBIC is more complicated than that of the ordinary BIC. For convenience, let us define the bus-transition as the transitions of bus-lines required to transmit data, and the overhead transition (OT) as the transitions required to transmit coding information. Since the *inv* is transmitted through M-line, all transitions of N-line are bus-transition. The M-line can move between 0 and 1, 0 and M, and 1 and M. All transition between 0 and 1 is included in bus-transition, but it is not clear whether the transition between M and the other normal states (0 and 1) is bus-transition or OT.

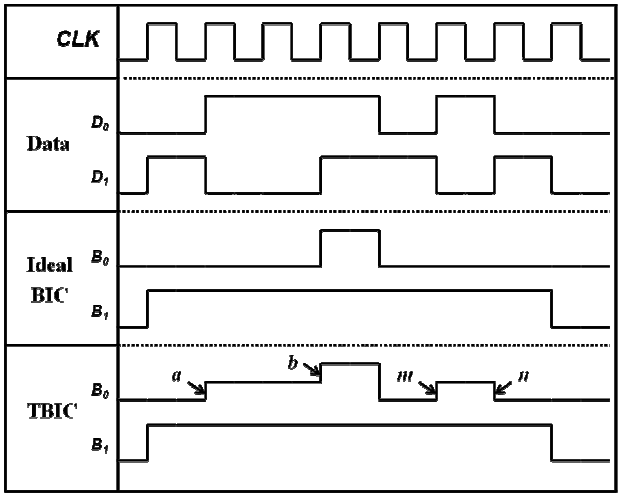


Fig. 4. Comparison of bus-line waveforms between ideal BIC circuit and TBIC.

To distinguish OT and bus-transition, let us compare the waveform of bus-lines between TBIC and the ideal BIC. Assume that no OTs happen in the ideal BIC, so that all transitions of the ideal BIC are bus-transitions. Fig. 4 shows the waveforms of bus-lines for the ideal BIC circuit and TBIC. As we can see in Fig. 4, M-line shows quite a different waveform from B0 of the ideal BIC, while N-line and B1 have the same waveform.

For convenience, the transition of M-line is classified into three patterns: direct transition (DT), via-transition (VT), and round transition (RT). A DT is a transition between 0 and 1. Both VT and RT are composed of two transitions involving M-state. The first transition of VT and RT is a transition from a normal state (0 or 1) to M. If the second transition from M goes back to its originated state, it is classified as RT. If the second transition goes to the other normal state, it is classified as VT. For example, the combined transitions *a* and *b* in Fig. 4 form a VT, while *m* and *n* become a RT.

Every DT can find its matching transition in B0 of ideal BIC. For VT, it can find a matching transition in B0 of ideal BIC at the position of the second transition. The dynamic power dissipated by a VT is the same as the power dissipated by a DT. For example, the power consumed by *a* and *b* can be calculated by

$$P_{VT} = P_a + P_b = \int_0^{V_M} C_M V dV + \int_{V_M}^{V_{DD}} C_M V dV = \int_0^{V_{DD}} C_M V dV$$

This is the same as the power dissipated in a DT. Note that the dynamic power for a VT is independent of the voltage level (V_M) of M-state. The DT and VT of M-line can be seen as bus-transition because the same amount of bus power required in the ideal BIC circuit.

As can be seen in Fig. 4, however, there are no matching transitions in B0 of ideal BIC for RT. Therefore, RTs are OTs of TBIC. Let us denote

$$\Delta V_{MH} = V_{DD} - V_M$$
$$\Delta V_{ML} = V_M - 0 = V_M$$

The dynamic bus power for a RT-0 such as *m* and *n* in Fig. 4 is

$$\begin{aligned} P_{RT-0} &= P_m + P_n \\ &= \frac{1}{2} C_M (\Delta V_{ML})^2 + \frac{1}{2} C_M (\Delta V_{ML})^2 \\ &= C_M (\Delta V_{ML})^2 \end{aligned}$$

Similarly, for 1→M→1 transition

$$P_{RT-1} = C_M (\Delta V_{MH})^2$$

The power for a RT depends on V_M. If 1→M→1 transitions and 0→M→0 transitions happen in equal rate, the average power of RT is minimum when V_M=V_{DD}/2, and then $\Delta V_{MH} = \Delta V_{ML}$=$V_{DD}$/2, and

$$P_{RT} = \frac{1}{4} C_M V_{DD}^2 = \frac{1}{2} P_{DT}$$

The relation shows that the power dissipated in a RT is 1/2 of that of DT, which means that the effective number of OT is half of the number of RTs. This helps to increase the performance of TBIC.

IV. EXPERIMENTS

Fig. 5 shows the simulation result of the M-line driver circuit. The simulation is performed by HSPICE with IBM's "1.2V-0.13μm 8RF-LM" model parameters [11]. The mid-level voltage is in the range of 0.5~0.7V. Although it may slightly affect the power dissipation of OTs, it is not a serious problem in operation.

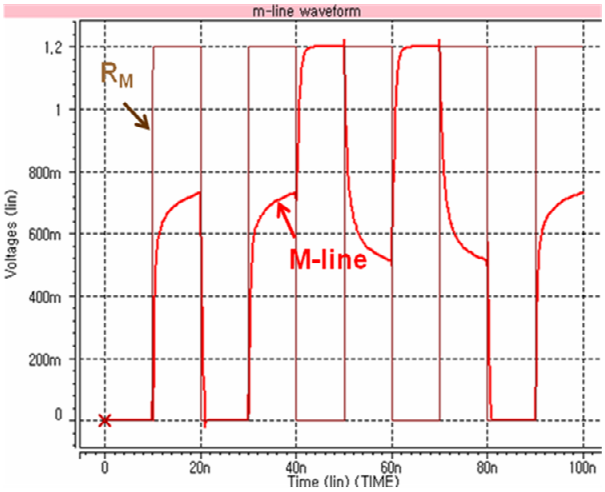


Fig. 5. Simulation waveforms of M-line.

Simulations are performed to estimate the performance of TBIC and to measure the amount of OTs occurring in TBIC. For the application to multimedia VLSI chips, some audio and video files are used in the experiments. The experiments are carried out with 9 different files. Three of them are random binary files generated by a random number generator. Other three are music files of the MP3 format: "For Elise", "Under the Sea", and "The Cup of Life". The other three are movie trailers of the MOV format: "Kung Fu Panda 2", Transformers 3", and "Water for Elephants".

The simulations count the number of transitions of bus-lines during the transfer of each file through 16-bit, 32-bit, 64-bit and 128-bit buses. The performance of TBIC is compared to that of the ordinary partitioned BI (PBI) scheme which uses invert-lines. Two PBIs are used; the first PBI (PBI-1) is partitioned into 8-bit sub-buses which is usually used to get moderate performance with small increase of bus-width, and the second PBI (PBI-2) is partitioned into 2-bit sub-buses which can provide the maximum performance.

For these three BIC schemes, the number of transitions of bus lines, and the number of overhead transitions are obtained by simulations. Through the simulation, the following numbers are obtained.

N_{RAW} : the total number of transitions without applying any BI algorithm.
N_B : the total number of bus-transitions
N_{OT} : the total number of overhead transitions
N_T : the effective total number of transitions

Table 2. Simulation results (Unit: %)

Bus Width (bit)			16			32			64			128		
File Format			.bin	.mp3	.mov	.bin	.mp3	.mov	.bin	.mp3	.mov	.bin	.mp3	.mov
PBI-1	P_B		49.92	50.28	50.25	73.09	73.39	72.77	72.74	72.79	72.84	72.32	72.72	72.81
	P_{OT}		25.04	24.86	24.87	9.05	8.84	9.03	9.09	9.01	9.01	9.12	9.04	9.03
	P_N		81.70	82.28	82.04	82.13	82.23	81.81	81.83	81.79	81.86	81.44	81.76	81.84
	R		18.30	17.72	17.96	17.87	17.77	18.19	18.17	18.21	18.14	18.56	18.24	18.16
PBI-2	P_B		49.92	50.28	50.25	50.40	50.68	50.02	49.99	50.15	50.07	49.70	50.09	50.05
	P_{OT}		25.04	24.86	24.87	24.80	24.66	24.99	25.01	24.93	24.97	25.15	24.96	24.97
	P_T		74.96	75.14	75.13	75.20	75.34	75.01	75.00	75.08	75.04	74.85	75.05	75.02
	R		25.04	24.86	24.87	24.80	24.66	24.99	25.00	24.92	24.96	25.15	24.95	24.98
TBIC	P_B	0↔1	41.65	42.13	41.99	42.10	42.41	41.75	41.82	41.84	41.79	41.29	41.76	41.76
		0→M→1	4.18	3.99	4.06	4.24	4.08	4.08	4.15	4.10	4.08	4.34	4.12	4.09
		1→M→0	4.09	4.15	4.20	4.09	4.19	4.19	4.05	4.21	4.19	4.18	4.20	4.20
		total	49.92	50.27	50.25	50.43	50.68	50.02	50.02	50.15	50.06	49.81	50.08	50.05
	P_{OT}	0→M→0	4.12	4.17	4.28	4.14	4.14	4.31	4.26	4.17	4.31	4.14	4.19	4.30
		1→M→1	4.27	4.25	4.06	4.10	4.10	4.07	4.22	4.17	4.06	4.09	4.13	4.06
		total	8.39	8.42	8.34	8.24	8.24	8.38	8.48	8.34	8.37	8.23	8.32	8.36
	P_T		54.12	54.48	54.42	54.55	54.80	54.21	54.26	54.32	54.25	53.93	54.24	54.23
	R		45.89	45.52	45.58	45.45	45.20	45.79	45.74	45.68	45.76	46.08	45.76	45.77

For PBI-1 and PBI-2, the transitions of bus-lines are counted as N_B, and every transition of invert-lines is counted as N_{OT}. For TBIC, N_B includes all transitions of N-line and all DTs and VTs of M-line. N_{OT} of TBIC is the number of RTs. The effective total number of transition (N_T) is obtained by the relation

$$N_T = N_B + N_{OT} \quad \text{(for PBI-1, PBI-2)}$$
$$N_T = N_B + N_{OT}/2 \quad \text{(for TBIC)}$$

The simulations results are shown in Table 2. The averaged values are used to simplify the table; the values in the columns of the bin, mp3, and mov represent the average of the three files of the same format for random binary, mp3, and mov, respectively.

P_B, P_{OT}, and P_T in Table 2 are the percentage of N_B N_{OT} and N_T against N_{RAW}, respectively. The reduction ratio R (=100-P_T) represents the percentage reduction of N_{RAW} by the applied BIC scheme, so that it can be used as the performance of the scheme.

As we can see in Table 2, there is no significant performance difference among the three data formats. PBI-1 reduces transitions by around 18%, while PBI-2 reduces transitions by around 25%. As expected, the performance of PBI-1 is poorer than that of PBI-2. Partitioning with the smaller bus-width can provide a higher reduction ratio. PBI-2 can reduce bus transitions by 25% at the expense of a 50% increase of bus width. The reduction ratio of TBIC is about 45.7%, which is almost triple that of PBI-1, and 83% greater than that of PBI-2.

The discrepancy of the reduction ratio comes from the difference of the P_{OT}. Note that the P_B of TBIC and PBI-2 are almost the same. Both schemes reduce bus-transitions by about 50%. However, the P_{OT} of PBI-2 is about 25% of N_{RAW}, which is about half the number of the reduced transitions. Therefore, in PBI-2, about 50% of the performance is lost by the transitions of the invert-lines.

The number of RTs in TBIC is about 8.3% of N_{RAW}, and the effective number of OTs is about 4.2% when $V_M = 1/2V_{DD}$. Theoretically, the maximum reduction ratio achievable by the BI algorithm is 50%. The reduction ratio of TBIC is about 45.7%, which is only 4.3% smaller than the theoretical maximum reduction ratio.

This shows that implementation that prevents generation of OT is very important in improving performance of algorithm.

V. CONCLUSIONS

A new bus-invert coding circuit called TBIC (Two-bit Bus-Invert Coding with a mid-level state bus-line) is presented in this paper. TBIC intends to remove the problems of the invert-line and to approach the maximum performance of the BI algorithm. To avoid the increase of bandwidth, TBIC removes the invert-line by transmitting the coding information through a bus-line.

To send 3-bit by two buslines in a cycle, a mid-level state called M-state is added to the normal 1 and 0 states. The encoding and decoding logic of TBIC is developed based on the transition among the 0, 1, and M states.

The result of simulations shows that TBIC can reduce bus-transitions by 45.7%. This reduction ratio is 83% greater than the maximum reduction ratio achievable by ordinary BIC with invert-lines. The large performance improvement of TBIC comes from the effective suppression of OTs. The number of OTs generated in TBIC is only 17% of the OTs generated in invert-lines.

REFERENCES

[1] M. R. Stan and W. P. Burleson, “Bus-invert coding for low-power I/O”, *IEEE Trans. VLSI Syst.*, vol. 3, pp. 49–58, Mar. 1995.

[2] Y. Shin, S. I. Chae, and K. Choi, “Partial bus-invert coding for power optimization of application-specific systems,” *IEEE Trans. VLSI Syst*., vol. 9, pp. 377-383, Apr. 2001.

[3] Y. Shin, K. Choi, and Y. Chang, “Narrow bus encoding for low-power DSP systems,” *IEEE Trans. VLSI Syst*., vol. 9, pp. 656-660, Oct. 2001.

[4] R. B. Lin and C. M. Tsai, “Weight-based bus-invert coding for low-power applications”, in *Proc. ASP-DAC/VLSI Design*, pp. 121-125. Jan. 2002.

[5] J. Natesan and D. Radhakrishnan, “Shift invert coding (SINV) for low power VLSI”, *2004. Euromicro Symp. on Digital system Design*, pp. 190-194, Sept. 2004.

[6] Y. Zhang, J. Lach, K. Skadron, and M. R. Stan, **“**Odd/Even bus invert with two-phase transfer for buses with coupling”, in *Proc. 2002 Int. Symp. on Low Power Electronics and Design,* pp. 80-83, 2002.

[7] U. Narayanan, K. S. Chung, and T. Kim, “Enhanced bus invert encodings for low-power”, *IEEE Int. Symp. on Circuits and Systems,* vol. 5, pp. 25-28, May 2002.

[8] M. Yoon and B. Roh, "A novel low-power bus design for bus-invert coding," *IEICE Trans. Electronics*, vol. E90-C, no. 4, pp. 731-734, 2007.

[9] M. Yoon "Achieving maximum performance for bus-invert coding with time-splitting transmitter circuit," *IEICE Trans. Fundamentals*, vol. E95-A, no. 12, pp. 2357-2363, 2012.

[10] S. Ramprasad, N. R. Shanbhag, and I. N. Hajj, “Informaiton-Theoretic Bounds on Average Signal Transition Activity”, IEEE Trans. VLSI Systems, vol. 7, pp. 359-368, 1999.

[11] The MOSIS Service, http://www.mosis.com/ [Online]

Myungchul Yoon received the BS and MS degrees in electronics engineering from Seoul National University, Korea, in 1986 and 1988 respectively, and the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin in 1998. From 1988 to 2002, he was with Hynix Inc. Icheon, Korea as a technical research staff at Semiconductor R&D Lab. And Mobile Communication R&D Lab. From 2005 to 2006, he was with DGIST, Korea as a technical staff at the Information Technology R&D Division. Since 2006, he has been with the Department of Electronics Engineering, Dankook University, Cheonan, Korea, where he is a professor. His research interests are in low-power VLSI design, embedded systems, mobile communication, and wireless personal area networks (WPAN).