A Leakage-aware Bus Coding Algorithm

Sangkwon Na
Department EECS, KAIST
373-1 Guseong-Dong, Yuseong-gu,
Daejeon, Republic of Korea
82-42-869-4032
skna@vslab.kaist.ac.kr

Youngsoo Shin
Department EECS, KAIST
373-1 Guseong-Dong, Yuseong-gu,
Daejeon, Republic of Korea
youngsoo@ee.kaist.ac.kr

Chong-Min Kyoung
Department EECS, KAIST
373-1 Guseong-Dong, Yuseong-gu,
Daejeon, Republic of Korea
kyung@ee.kaist.ac.kr

ABSTRACT

In this paper, we propose a low leakage bus coding algorithm which is suitable to both the address bus and the data bus. The proposed encoding scheme converts the input bits with low leakage state of a devised buffer. We characterize the bus and find optimal encoding scheme. It uses the staggered threshold voltage buffers. When it compared to the conventional bus, the proposed bus coding scheme reduces about 71% of total power and about 3 times leakage power.

Keywords

Low leakage, bus coding, staggered threshold voltage buffer, low power

1. INTRODUCTION

As a result of technology enhancement, leakage power dissipation has emerged as a first-class design consideration in high-performance and low-power processor design. Historically, architectural innovations for improving performance relied on exploiting ever larger numbers of transistors operating at higher frequencies. To minimize the resulting switching power dissipation, recent technology generations relies on reducing the supply voltage. In order to maintain performance, however, it has required a corresponding reduction in the transistor threshold voltage. Since the MOSFET sub-threshold leakage current increases exponentially with a reduced threshold voltage, it leads huge leakage power dissipation. Besides, leakage power dissipation became a major source of overall chip power dissipation in modern deep-submicron technology (< 0.13 ㎛). Moreover, it is expected to grow by a factor of five every chip generation [1]. For processors, it is estimated that, in 0.01 ㎛ technology, leakage power will account for approximately 50% of the total chip power [2]. Figure 1 shows that comparison of active power density and sub-threshold power density according to the gate length [3]. As shown in the figure, the sub-threshold power density becomes more dominant total chip power source compared to the active switching power density. Moreover, if the gate length becomes smaller than around 65nm, the sub-threshold power density will become bigger than active power density. Therefore, careful design constraints should be considered when designing high performance chip.

In high performance SoC chips, the speed of global bus determines the total chip performance since the signal propagation time in the global bus contributes bottleneck in processing data. To improve the speed penalty of global buses in SoC chips, appropriate number of buffer (usually an inverter) should be inserted in the middle of bus wire since the bus wire is so high capacitive that it cannot meet timing constraints with sole bus driver. Table 1 shows how much leakage power is consumed in an inverter compared to the other logic component [4]. Since the inverter cannot exploit the stack effect, the average leakage current through the inverter is very large regardless of the input vector. Besides, the inverter size of the buffer used in the bus wire

Table 1. Leakage power consumption of the basic logic elements

	Input vector	type	Process technology		
			0.18um	0.10um	0.07um
NMOS	0	N	7.9	10.9	67.6
PMOS	1	P	4.0	9.7	80.4
INV	0	N	7.9	10.9	67.6
	1	P	4.0	9.7	80.4
NAND2	00	N	0.3	0.4	9.6
	01	N	7.9	10.8	46.0
	10	N	4.7	5.1	44.0
	11	P	8.1	19.4	159.5
NOR2	00	N	15.9	21.7	133.8
	01	P	3.6	5.9	45.5
	10	P	4.3	9.7	77.5
	11	P	0.9	0.7	5.9

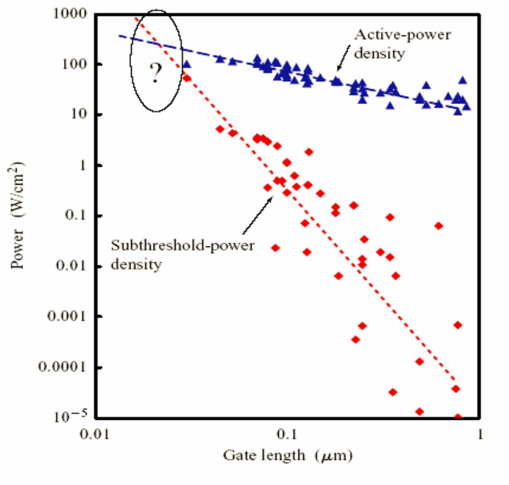


Figure 1. Active power density and sub-threshold power density according to gate length

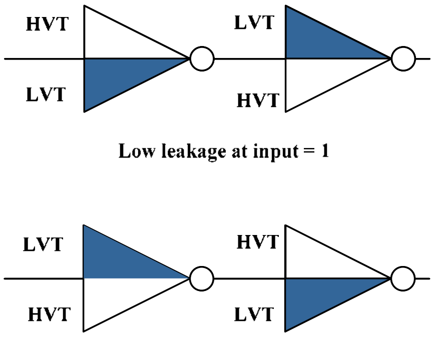
is very wide to drive high capacitive bus wire. Therefore, the leakage power reduction of the buffer used in the global bus must be required.

The remainder of this paper is organized as follows. In Section II, we present the staggered threshold voltage buffer and bus models, In Section III, we present proposed bus coding algorithm. In Section IV, the implementation of proposed algorithm and experimental results are shown. Finally we offer conclusions in Section V.

2. MODELING OF GLOBAL BUS

2.1 Staggered Threshold Voltage Buffer

Figure 2 shows that the schematic diagram of staggered threshold voltage (V_T) buffer [9]. The staggered V_T (SVT) buffer is consisted of low V_T MOS and high V_T MOS. Low V_T MOS means that the MOS transistor which has nominal threshold voltage used in the critical path of the circuit. High V_T MOS means that the MOS transistor which has higher threshold voltage than the low V_T MOS. The high V_T MOS is usually used in the non-critical path that can eventually reduce the leakage current. The shaded region of the buffer means the low V_T MOS and non-shaded region of the buffer means the high V_T MOS. Because the leakage power of the high VT MOS is much less (about 10 times less) than the low V_T MOS, the leakage power of the buffer can be significantly saved with specific input value. For example, when the input value is '1,' the first set of buffer consumes low leakage, and when the input value is '0,' the second set of buffer consumes low leakage. Because the high V_T MOS is slower than the low V_T MOS, the rising and falling propagation delay isn't same. To meet rising/falling time requirement, we use the larger high V_T MOS that the low V_T MOS. Though it increases dynamic and leakage power, its effect can be negligible.



Low leakage at input = 1

Low leakage at input = 0

Figure 2. Schematic diagram of staggered threshold voltage buffer

2.2 Buffer Insertion on Bus Wire

To minimize the propagation delay at the high capacitive global bus wire, appropriate number of buffer should be inserted in the middle of the bus wire. To obtain the optimal number of the inserted buffer and the optimal size of the buffer size, the following formula is used [5].

Table 2 shows the bus wire model which used in our experiment. Since there is lack of information for the 70nm BPTM wire model, we used the 0.13 ㎛ CMOS technology metal-5 wire model under the assumption of negligible scaling amount for the global wire.

Table 2. Bus wire model

Wire model	Metal-5 of 0.13um CMOS technology
Wire length (L)	5mm
Planar cap. of unit length wire (Ca)	0.051fF/um
Coupling cap. of unit length wire (Cc)	0.353fF/um
Fringing cap. of unit lecngh wire (Cf)	0.035fF/um
Wire resistance of unit length	0.114Ω /um
Optimal number of stage (k_{OPT})	8
Optimal size of buffer(h_{OPT})	15 times of min. size inv.

$$Dealay \approx k\left[p_1 \frac{R_{INT0}L}{k}\frac{C_{INT0}L}{k} + p_2\left(\frac{R_0}{h}hC_0 + \frac{R_0}{h}\frac{C_{INT0}L}{k}\frac{R_{INT0}L}{k}hC_0 \right)\right]$$

where h is buffer size, k is number of stages, C_0 is gate capacitance of minimum size transistor, R_0 is output resistance of minimum size transistor, C_{INT0} is capacitance of unit length wire, R_{INT0} is resistance of unit length wire. By equating following equation, we are able to obtain optimal number of stage (k_{OPT}), and optimal size of the buffer (h_{OPT}).

$$\frac{\partial Delay}{\partial h} = 0 \quad \rightarrow \quad h_{OPT} = \sqrt{\frac{C_{INT0}R_0}{R_{INT0}C_0}}$$

$$\frac{\partial Delay}{\partial k} = 0 \quad \rightarrow \quad k_{OPT} = L\sqrt{\frac{p_1}{p_2}}\sqrt{\frac{R_{INT0}C_{INT0}}{R_0C_0}}$$

3. PROPOSED ALGORITHM

3.1 Characterization of Bus

Most SoCs have more than one processor, such as RISC processor and/or DSP. These processor-based SoCs use a shared bus to communicate with other hardware IPs. According to the characteristics of data, we can generate coding elements variously. For example, the addresses from load/store of the processor are sequential and sparse. It means that address bus's utilization is very low, less than 15%. When the bus isn't used, they can be fixed to a specific value, such as all ones or all zeros. From this fact, coding scheme becomes very simple.

We choose our processor as LSI Logic's ZSP500 [6]. ZSP500 has an eight stage pipeline and is RISC-based four-way superscalar architecture. Using instruction set simulator (ISS), we extract the list of the load/store and calculate the utilization of the address bus. As shown in Section IV, when using the characteristics of the address bus, we can get large gain.

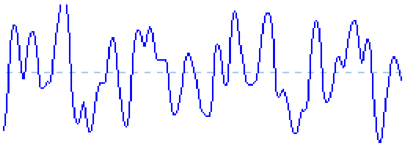


Figure 3. Voice waveform

Secondly, audio data, especially voice data, have the continuity characteristics. As shown in Figure 3, voice pulse changes across zero point, and its width is narrow. Therefore the MSBs of voice data are mostly zero or one consecutively. In other words, most MSBs of voice data have big correlativity.

3.2 Encoding Algorithm

Among various types of interconnect buffers, we determine to use staggered threshold voltage buffer from leakage power perspective. In order to match input bits, which are sent through interconnect buffers, with low leakage state of SVT buffer, we introduce coding scheme using the characterization of bus. At first, we consider the following two features.

(1) Minimize overhead from additional coding elements, such as an encoder and a decoder.

(2) Minimize leakage power as matching coded data with low leakage state of staggered threshold voltage (SVT) buffer.

At the beginning, we arrange the SVT buffer's low leakage state as 101010…1010 according to the normalized switching activity. In other words, at MSB, we place the SVT buffer which shows lower leakage current when the input is one, and the adjacent SVT buffer shows lower leakage current when the input is zero. For the first feature addressed above, we use the characteristics of the address bus and the data bus to minimize the complexity of coding logics.

According to the utilization of the address bus, we focus on a period when the address bus isn't used. While idle period, the address bus drives all zeros. Therefore, a complex coding scheme is not required. Fixed coding scheme can be applied to the address bus. The encoder simply matches the address bus to the low leakage state of the SVT buffer.

Figure 4 shows an encoder for address bus. Each processing element (PE) changes an input bit according to the position of it. PE is arranged according to a bit position's low leakage state of SVT buffer.

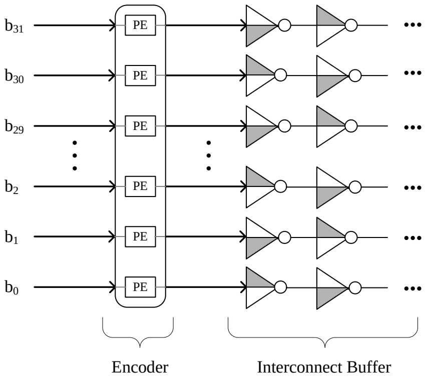


Figure 4. An encoder for the address bus

For another coding scheme targeted on the data bus, we assume a system which deals with voice data. Using voice data's characteristics, we partition the data bus into a unit of coding process. Typical bus coding algorithm using masks needs additional bits to transfer the index of masks [7]. Because additional bits also switch and consume dynamic power, it is possible that they can set off the benefit from encoding. To prevent this side effect, we use the data bit itself as the index of mask. In other words, among a part of coding process unit, a first bit is used as the index of mask.

It is important to determine coding unit size corresponding to the correlated range of data. The enlargement of unit size achieves the simplification of coding logic. For example, for 32-bit bus, a four-bit unit coding needs totally eight coding elements. On the other hand, a sixteen-bit unit coding just needs two coding elements.

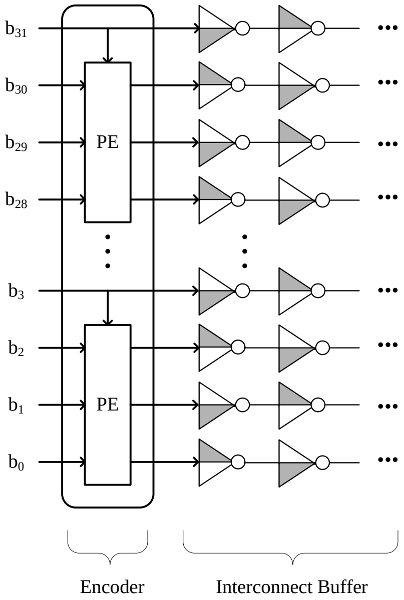


Figure 5. An encoder for the data bus

Figure 5 shows an encoder of four-bit unit coding. The first bit of coding unit is used to encode and decode remainder bits and is bypassed.

4. EXPERIMENTAL RESULTS

4.1 Address bus

We generate testbenches with the extracted data from ISS of LSI Logic's ZSP. The extracted data are captured from VLD and IDCT function of JPEG decoder. We design the encoder and the decoder using BPTM 70nm process technology [8]. The encoder and the decoder, targeted on the address bus, are designed simply as an inverter in order to change it into low leakage input of SVT buffer.

Therefore, the portion of power consumption of the encoder and the decoder is considerable little and negligible.

Figure 6 shows HSPICE simulation results. The number of the address bus line is 32-bit. Total simulation period is 1880 cycles. The first bar of the Figure shows simulation results of VLD. In case of VLD, when coding algorithm applied, 71% of total power is reduced. The second bar of the Figure shows simulation results of IDCT. In case of IDCT, we can get 69% of total power reduction using coding algorithm.

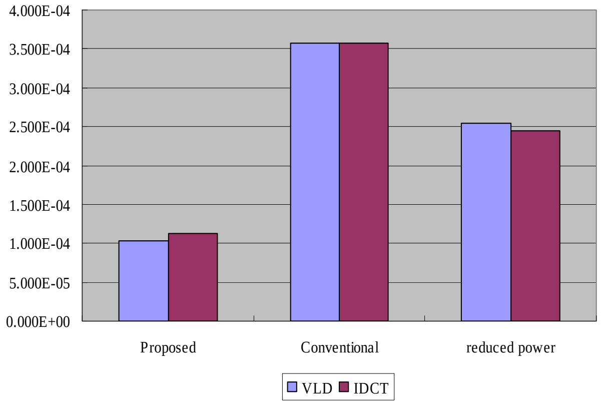


Figure 6. Power consumption of the address bus

4.2 Data bus

For the data bus, we use voice data at simulation. These data are male's voice, which pronounces 'test'. We make three experiments according to the coding unit (4-bit, 8-bit, and 16-bit coding unit).

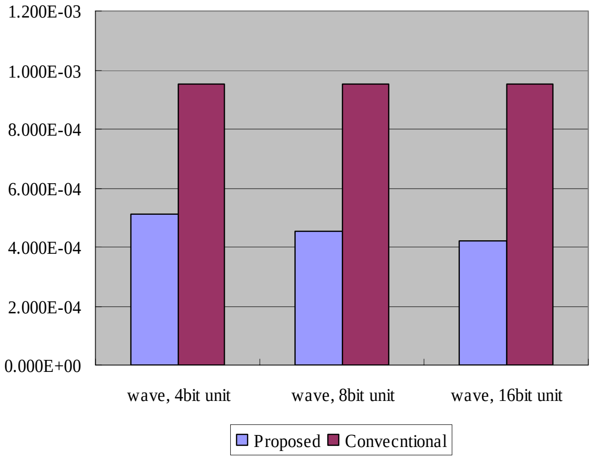


Figure 7. Power consumption of the data bus

Figure 7 shows HSPICE simulation results. The number of the data bus line is 32-bit. Total simulations are executed during 2000 cycles. The column of the Figure means the coding unit, respectively. At the data bus which transfers highly self-correlated data, more than 50% of power consumption can be saved.

Figure 8 shows that the larger unit reduces more power. Its tendency differs according to the data pattern. Testbenches we use are highly self-correlated data, and its range is large. Besides, the enlargement of unit size simplifies coding logic.

Finally, we can get about 70% of total power savings at the address bus and 58% of total power savings at the data bus with proposed coding algorithm.

5. CONCLUSION

A low-leakage bus coding algorithm which is suitable to both the address bus and the data bus is proposed. At first, we characterize the bus and analysis coding scheme. We generate coding elements using as possible as minimum logics. The encoder converts input data with coded words which show minimum leakage power consumption at SVT buffers.

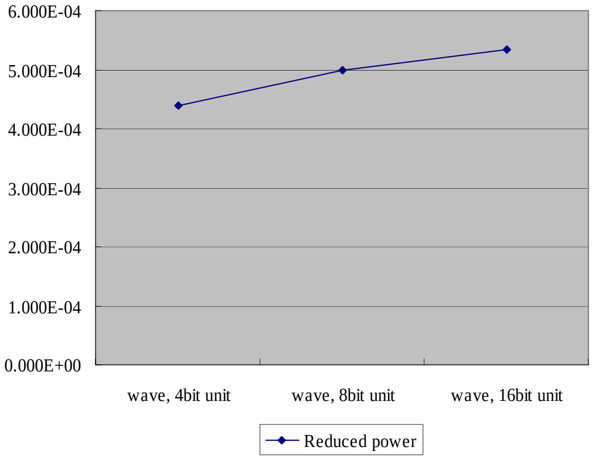


Figure 8. Reduced power according to the coding unit

For the address bus, the proposed coding algorithm saves about 71% of total power. Its benefit is that 1) easy application and 2) target for general-purpose processor. For the data bus, we can get more than 50% of total power consumption using the proposed method.

If the portion of leakage power consumption among total power consumption gets larger, the savings of power consumption will become larger.

6. REFERENCES

[1] S. Borkar, "*Design challenges of technology scaling,*" IEEE Micro, vol. 19, pp. 23-29, July-Aug, 1999.

[2] T. Kam, S. Rawat, D. Kirkpatrick, R. Roy, G. S. Spirakis, N. Sherwani, and C. Peterson, "*EDA chanllenges facing future microprocessor design*," IEEE Trans. Computer-Aided Design, vol. 19, pp. 1498-1506, Dec. 2000.

[3] Edward Nowak, "*Maintaining the Benefits of CMOS Scaling when Scaling Bogs Down*," IBM J. Res. & Dev., vol. 46, no. 2/3, March/May 2002.

[4] Xuning Chen and Li-Shiuan Peh, "*Leakage Power Modeling and Optimization in Interconnect Networks,*" ISLPED, 2003, August.

[5] T. Sakurai, "Superconnect Technology, "IEICE Trans. On Electron., vol. E84-C, no. 12, December, 2001.

[6] http://www.lsil.com

[7] P. P. Sotiriadis, A. Chandrakasan, "*Low Power Bus Coding Techniques Considering Inter-wire Capacitances,*" IEEE CICC, pp. 507-510, May, 2000.

[8] http://www-device.eecs.berkeley.edu/~ptm/

[9] Harmander S. Deogun, Rajeev R. Rao, Dennis Sylvester, David Blaauw, "Leakage-and Crosstalk-aware Bus Encoding for Total Power Reduction," DAC, pp. 779-782, June, 2004.