Leakage-and Crosstalk-Aware Bus Encoding for Total Power Reduction

Harmander S. Deogun, Rajeev R. Rao, Dennis Sylvester, David Blaauw
Department of EECS, University of Michigan – Ann Arbor
{hdeogun, rrrao, dmcs, blaauw}@umich.edu

ABSTRACT

Power consumption, particularly runtime leakage, in long on-chip buses has grown to an unacceptable portion of the total power budget due to heavy buffer insertion to combat RC delays. In this paper, we propose a new bus encoding algorithm and circuit scheme for on-chip buses that eliminates capacitive crosstalk while simultaneously reducing total power. We introduce a new buffer design approach with selective use of high threshold voltage transistors and couple this buffer design with a novel bus encoding scheme. The proposed encoding scheme significantly reduces total power by 26% and runtime leakage power by 42% while also eliminating capacitive crosstalk. In addition, the proposed encoding is specifically optimized to reduce the complexity of the encoding logic, allowing for a significant reduction in overhead which has not been considered in previous bus encoding work.

Categories and Subject Descriptors

B.8.2 **[Performance and Reliability]**: Performance analysis

General Terms

Algorithms, Performance, Design

Keywords

Leakage reduction, encoding, low power

1. INTRODUCTION

Continued scaling of process technologies has led to smaller device features, faster clock speeds, and rapidly shrinking interconnects. In order to maintain the performance gain associated with each technology generation, the threshold voltage (V_{th}) of the MOSFET device is aggressively scaled as well. However, lowering V_{th} has resulted in an increase in the subthreshold current of the device at 3-5X per generation [1]. It is projected that in the 90nm node subthreshold leakage power will be as much as 40% of the *total power* for high-performance processors [2]. Buffers used to manage delay and signal integrity problems on long on-chip buses contribute to a major component of this leakage power. In general, inverters or buffers contribute roughly 50% of the total device width on chips [3] and, due to the lack of stack effect, constitute a major fraction of the total leakage power. Further, it has been estimated [4] that 70% of the total cell count at the 32nm node will be due to buffers and repeaters. Consequently, it is critical to develop approaches that aim to limit this component of total power.

DAC 2004, June 7–11, 2004, San Diego, California, USA

Recently, a number of strategies have been proposed that utilize bus encoding to eliminate undesirable effects that would otherwise occur during transmission of the unencoded bits. The approaches in [5,6,7,8] seek to minimize the dynamic power and delay in buses through various encoding schemes. The work in [5] is well suited for delay reduction by elimination of crosstalk (through a "self-shield encoding") but it does not address power reduction. The authors in [6] extend the work in [5] and propose a method that eliminates crosstalk and also reduces dynamic power. However, they do not address the critical issue of static leakage power. In [7], the authors shuffle the order of the bus lines to minimize opposite-phase transitions on adjacent bus lines to reduce power due to crosstalk. The results in [8] show a reduction in both static and dynamic power using their technique of low-voltage BiCMOS and termination networks.

While the above mentioned works describe methods of eliminating crosstalk and/or reducing dynamic power, most do not tackle the rising leakage power levels in such buses. These approaches also do not attempt to minimize the complexity of the encoder and decoder (*codec*) hardware and therefore may have high power and delay overheads. In this paper, we propose a new bus encoding method that minimizes total power while simultaneously eliminating crosstalk. Our approach uses a novel bus encoding scheme coupled with a dual-V_{th} buffer design. We demonstrate that by combining a leakage-aware encoding with a dual-V_{th} bus driver design, we can reduce average runtime leakage power by 42% and average total power by 26% while concurrently eliminating crosstalk. Our approach also minimizes the codec logic complexity resulting in significantly reduced power and delay overhead.

The remainder of this paper is organized as follows. Section 2 gives an overview of our approach for leakage-aware bus encoding. Section 3 details the algorithm that is used to derive the low leakage and crosstalk eliminating encoding. Section 4 describes the experimental test setup and presents our results. Section 5 concludes the paper.

2. OVERVIEW OF ENCODING

In general, low threshold voltage (LVT) buffers are used in on-chip memory buses to achieve high performance requirements. However, LVT devices are unsuitable from a power perspective due to their very high leakage power. A simple way to reduce subthreshold leakage current is by raising V_{th}, which is accomplished by replacing the LVT buffers with high threshold voltage (HVT) buffers. Using HVT instead of LVT devices typically provides a leakage savings of 10X for the same size device. Note that there are

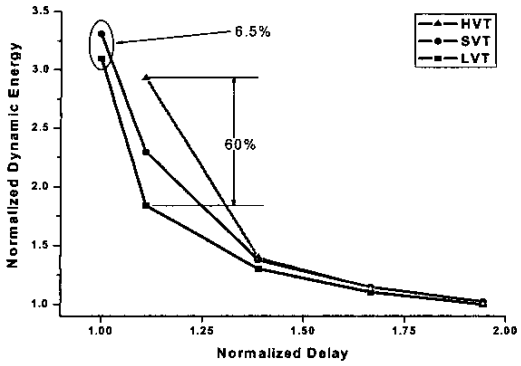

Figure 1: Normalized worst-case delay and dynamic energy for different bus line types.

other known techniques to reduce leakage during standby mode but in this paper we focus on *runtime* leakage reduction which is a more difficult and pressing problem. Currently dual-V_{th} is the only practical approach to achieving substantial runtime leakage reduction [9].

However, using HVT buffers leads to a large degradation in performance. Figure 1 was generated using HSPICE simulations of an on-chip global bus topology. The initial specified delay point is normalized to 1.00 on this plot. Different delay targets were set and the buffers were sized optimally to meet these new targets. It can be seen that a bus using HVT buffers is not able to meet a stringent delay constraint even with device sizing as a variable. The HVT buffers are only able to meet a delay target that is about 13% slower than the LVT buffers while incurring a substantial penalty (60%) in dynamic energy due to the aggressive sizing requirements. Thus, HVT buffers can greatly reduce leakage current but only by incurring significant penalties in delay and dynamic energy. In many high-performance applications, a penalty in delay or dynamic energy cannot be tolerated – this leads to a difficult tradeoff between meeting delay while trying to maintain power at manageable levels. Furthermore, it is known that delay becomes more sensitive to V_{th} in sub-1V technologies and the corresponding delay penalty associated with using high-V_{th} devices in these processes will grow [10].

To resolve this problem, and also address leakage issues, we propose the use of staggered threshold voltage (SVT) buffers. These buffers are constructed by combining LVT and HVT transistors in a staggered fashion as show in Figure 2. We note that the idea of dual-V_{th} inverters has been known for some time [11], however we propose a novel construction method in SVT devices. SVT devices enable the design of high-performance buses with a

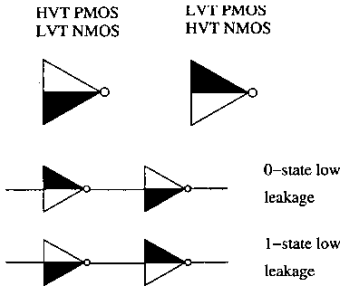

Figure 2: Staggered threshold voltage buffers.

much reduced penalty in dynamic energy. In Figure 1, the initial specified delay constraint which was not met using HVT devices is now attainable using SVT buffers. Further, the dynamic energy penalty has been reduced by nearly 10X to only 6.5% at the fastest achievable design point of SVT. This dynamic penalty is due to the slightly larger device sizes that must be used to ensure that the delay numbers of SVT and LVT are the same.

In SVT buffers, if the active devices are LVT and the off-state devices are HVT then we achieve the optimal trade-off between leakage and power since the LVT devices ensure shorter propagation delays while the HVT devices result in lower leakage power. Thus, if the input values to the bus line are known, then the SVT buffers can be designed to achieve the optimal power-delay trade-off. It has been pointed out recently that on-chip caches store primarily zeroes which indicates that data buses out of such caches may have a high probability of carrying *0*s rather than *1*s [12]. This type of application would benefit greatly from SVT buffers. However, assuming this sort of imbalance in input probabilities does not exist, we turn our attention to the use of bus encoding to *enforce* the input states that will result in lowest leakage.

To ensure that the HVT device in a given inverter is usually off (to reduce subthreshold leakage), an encoding scheme is developed to skew the data bits of the bus to either the *0* or *1* state. The stagger configuration for the SVT bus is then chosen appropriately (Figure 2) such that each bus line can be designated as a 0-state or 1-state low leakage bus line. In this way, the bus line can spend, on average, most of the time in the designated low leakage state. This is the leakage-aware portion of the encoding.

The dynamic power expended by a set of bus lines can be reduced drastically by eliminating crosstalk effects between them. When a pair of adjacent wires transition in opposite directions, it results in worst-case conditions for both delay and power [5]. The magnitude of coupling capacitance between adjacent wires is normally greater than the ground capacitance due to the increased aspect ratio of modern interconnects [13], therefore the delay can be nearly twice as large as with just one wire switching next to a quiet wire. The crosstalk-aware portion of the encoding focuses on developing a coding mechanism that skews the states on the bus such that the possibility of the worst-case transition between a pair of adjacent wires is eliminated. The self-shield encoding method presented in [5] uses encoder/decoder logic and some additional wires to implement such a mechanism. Since these codec stages are unavoidable, we require them to be as small a percentage of the total bus delay as possible to minimize the overhead. The reduction in total power, along with elimination of crosstalk, far outweighs this incremental delay penalty. Moreover, as devices become faster and areas shrink with each generation, the overhead on the bus line grows smaller. Thus, the tradeoff between extra logic and reduction in power and elimination of crosstalk is reasonable.

We now utilize the SVT buffer technique in our encoding algorithm that uses an enhanced self-shield mechanism to eliminate crosstalk while simultaneously minimizing the total power. Additionally, we minimize the delay overhead by optimizing the codec logic.

3. PROPOSED ENCODING ALGORITHM

In the enhanced self-shield encoding scheme, the input bits are processed by the encoder architecture to produce a set of *codewords*. The mapping between the input bits and codewords is called a *codebook*. The *Hamming Distance* (HD) between a pair of

codewords is given by the number of 1s in the bitwise *XOR* between them. The HD term describes the number of bit differences between a pair of codewords.

As motivated in the previous section, we require an encoding scheme that has the following three features (*F1-F3*):

[*F1*] Eliminate crosstalk between adjacent bus lines
[*F2*] Minimize leakage by skewing the probability of the bits
[*F3*] Minimize overhead due to the encoding and decoding logic

We first address *F3*. When encoding b bits of input as e-bit codewords, it is essential to pick the smallest possible values for b and e to minimize the codec logic. We pick $(b,e) = (3,4)$ since this encoding leads to fairly simple encoder/decoder circuits (we also note that there does not exist a codeword mapping for (b,e) = (4,5) and for larger values of (b,e) the complexity of the codec logic increases considerably). For a 32-bit bus, we split the full bus into sets of 3-bits each and then encode each set individually. A shield (dedicated ground or V_{dd} wire) separates each set. Shield insertion is common in high-speed buses to suppress inductive effects and our use of one shield for every 3 or 4 wires is a typical method [14,15]. Thus, for each set of 3 input bits we use 4-bit codewords and the size of the codebook is $2^3 = 8$.

To address *F2*, it is essential to pick an encoding method where the leakage incurred by the eight codewords is minimized. Since the SVT technique skews the buffers on each bus line such that there exists an ideal leakage state for each line, there exists only one *ideal* 4-bit codeword that corresponds to the minimum leakage state simultaneously for a set of 4 bus lines. From our codebook (of size eight), we need seven additional codewords that are as close as possible to the ideal leakage state. To accomplish this, we choose codewords that have the least HD to the ideal leakage state. For any 4-bit ideal codeword, there are ${}_4C_1 = 4$ codewords within HD = 1 and ${}_4C_2 = 6$ codewords within HD = 2.

Conceptually, any of the 16 4-bit codewords can be chosen as the ideal codeword since a bus line consisting of SVT buffers can be tailored towards having either 0 or 1 as its low leakage state. However, for our encoding scheme the selection of the ideal codeword is dictated by *F1*. We first prove the following lemma.

Lemma: The ideal codeword in a codebook that satisfies *F1-F3* does not contain two adjacent bits that are the same.

Proof: Let $b_1b_2b_3b_4$ be the ideal 4-bit codeword. Suppose $b_1 = b_2$. Since we need to pick all codes that are HD = 1 (to satisfy *F2*), $b_1'b_1b_3b_4$, $b_1b_1'b_3b_4$ need to both be part of the codebook. However, a transition between these two states violates the self-shield coding condition since two adjacent bits are switching in opposite directions. Hence, it is impossible to have an ideal codeword with two adjacent bits that are the same. □

Using this lemma we can identify the ideal 4-bit codewords as *0101* and *1010*. Since these codewords are analogous, we only consider *0101*. Among the 4+6 HD=1,2 codewords we eliminate three HD=2 codewords since they violate the self-shield coding requirement. Thus, our codebook of size eight is given by *0101*, *0100*, *0111*, *0001*, *1101*, *0000*, *1100* and *1111*. With the codebook determined, we now assign the eight possible input states to the codewords; this assignment determines the overall performance of the encoding scheme and complexity of the codec logic.

For a given mapping from the input data bits to the codewords, we first define a power function, $P = D+L$. Here, D represents the dynamic power and L represents the leakage power expended by the encoded data bits. The dynamic power is dependent on the transition characteristics of the input data bits while the leakage power is dependent on their state characteristics. Based on the simulation profile of a memory bus we generate both the state and transition probabilities for sets of 3-bit data bits. Our objective is to generate a mapping from the 3-bit input to the 4-bit codewords that minimizes this power function P in addition to satisfying *F1-F3* and minimizing the logic complexity.

It is evident that to minimize P we need to assign the lowest probability values in both the state and transition probability tables to the codewords that consume the greatest amount of power. Since the codewords are known *a priori* and the codebook size of 8 is fairly small, we search the entire sample space of 8! mappings of symbols to codewords to determine the minimum value (P_{min}) of the power function. However, the mapping corresponding to P_{min} may potentially require complicated codec logic that would result in unacceptably large overhead. To avoid this, we set a tolerance limit T on P_{min} and examine the logic complexity of all mappings that have $P \leq P_{min}(1+T)$. Among these mappings, we choose the one with the smallest overhead (the overhead is quantified using Espresso [16] to determine the total number of gates required to construct the encoder and decoder for each mapping). Thus, we have obtained a mapping that consumes a sufficiently small amount of power while minimizing the logic overhead. A tolerance limit of approximately 5% typically captures the minimal number of gates. Note that T=0 corresponds to selecting the power-optimal mapping regardless of encode/decode overhead, making this a special case of our approach. Figure 3 is a summary of our proposed algorithm, called *BuffPower*, and has as inputs M, the memory trace of a program, and T, the tolerance limit set on P_{min}.

4. POWER AND PERFORMANCE ANALYSIS

The encoding algorithm described in Section 3 was implemented using industrial 0.13μm device models at a temperature of 105°C. A bus line length of 8 mm was assumed with an inverting repeater inserted every 800 μm. There were ten inverters such that the total bus line remained non-inverted. We obtained a large number of traces of a 64-bit memory bus for nine different benchmarks (from the Spec CINT 2000 suite [17]) running on an Alpha architecture-based microprocessor. For each benchmark we first constructed the state and transition probability tables. The static and dynamic power values were then scaled by the numbers in these tables.

In Figure 4, the base case (striped column) corresponds to an un-encoded set of bus lines driven using only LVT buffers. In addition

Algorithm BuffPower (M, T)
1. Construct state (S) and transition (R) probability tables from the memory trace M
2. E={set of all 8! mappings from 3-bit I/P to 4-bit codes}
3. **for each** (mapping ∈ E)
   Calculate P(mapping)
4. Sort mappings according to the Ps
5. Set E_{trunc}={set of all mappings with $P \leq P_{min}(1+T)$}
6. **for each** (mapping ∈ E_{trunc})
   Calculate delay(mapping)
7. Sort mappings according to the delays
8. **return** (mapping of min delay)

Figure 3: Summary of the proposed algorithm.

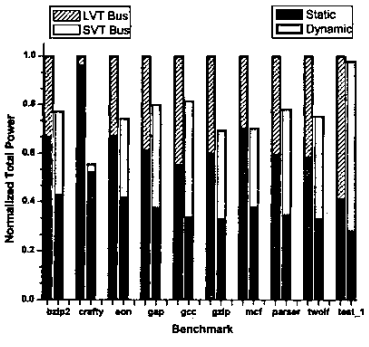

Figure 4: Total power reduction on various benchmarks using SVT buffer scheme and enhanced self-shield encoding.

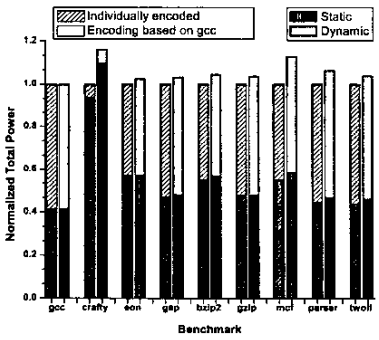

Figure 5: Total power comparison when using a single application-independent encoding.

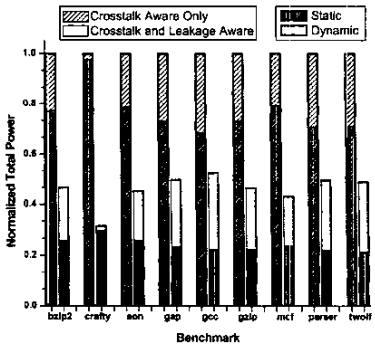

Figure 6: Power savings for the described crosstalk and leakage-aware encoding vs. crosstalk-only encoding [5].

to the nine standard applications shown, a TEST_1 application with a very high switching activity rate was artificially constructed. Figure 4 shows that for every application, the total power is reduced in the SVT bus case. On average, our method provides a savings of 26% in total power and in the best case about 44%. There was an average leakage savings of 42% with a small increase (<5%) in dynamic power, due to the additional bus line. For the TEST_1 application, although the dynamic power in the 4-bit encoded bus increases significantly, there was still enough savings in static power such that the total power was reduced slightly. This shows that our proposed encoding scheme is robust, even for cases where leakage power is a small portion of the total power.

In addition to finding the dynamic and static power reduction for each memory bus trace with respect to its own optimal encoding, we calculated the average state and transition probability table across all memory traces. It has been observed that on-the-fly encoding of the bus for each specific application would incur substantial overhead [18]. Instead, we used the memory traces of all applications and constructed the average state and transition probability tables. We observed that the tables obtained from such an averaging were almost identical to the tables corresponding to gcc. In Figure 5, we use the encoding corresponding to gcc (tolerance T = 5%) and calculate the power for each application. The increase in power when using a generic encoding compared to an application-specific encoding is about 10-15% in two cases and is essentially zero for the remaining applications.

In Figure 6, we compare our new technique with a crosstalk-aware only approach similar to [5] in terms of total power. We see that although the dynamic power is approximately the same, using the SVT buffer scheme reduces the leakage power significantly. On average, our technique yields about 54% savings in total power over all the benchmarks.

5. CONCLUSIONS

To address the growing issue of runtime leakage power consumed by on-chip buffers, we propose an enhanced self-shield encoding algorithm combined with a novel staggered threshold voltage (SVT) buffering technique for crosstalk-aware low-power bus encoding. We achieve considerable reductions in total power consumption while simultaneously eliminating crosstalk-inducing transitions and reducing the encode/decode logic overhead. On average, we achieved a total power savings of about 26% with a best case savings of 44%.

6. ACKNOWLEDGEMENTS

This work was supported by the MARCO/DARPA GSRC, the NSF, and by equipment donations from Intel and Sun.

7. REFERENCES

[1] S. Borkar. "Design Challenges of Technology Scaling," *IEEE Micro*, July/Aug 1999.

[2] J. Kao, S. Narendra, A. Chandrakasan. "Subthreshold Leakage Modeling and Reduction Techniques," *ICCAD 2002*.

[3] K. Bernstein, et al. "Design and CAD Challenges in sub-90nm CMOS Technologies," *Embedded tutorial, ICCAD 2003*.

[4] P. Saxena, et al. "The scaling challenge: can correct-by-construction design help?" *ISPD 2003*.

[5] B. Victor, K. Keutzer. "Bus encoding to prevent crosstalk delay," *ICCAD 2001*.

[6] Chun-Gi Lyuh, Taewhan Kim. "Low power bus encoding with crosstalk delay elimination," *ASIC/SOC Conference, 2002*.

[7] Youngsoo Shin, Takayasu Sakurai. "Coupling-driven bus design for low-power application-specific systems," *DAC, 2001*.

[8] Naehyuck Chang, et al, "Bus encoding for low-power high-performance memory systems," *DAC, 2000*.

[9] S. Tyagi, et al. "A 130nm generation logic technology featuring 70nm transistors, dual-Vt transistors and 6 layers of Cu interconnects," *Proc. IEDM*, Dec. 2000.

[10] D. Sylvester, H. Kaul. "Future performance challenges in nanometer design," *DAC 2001*.

[11] Qi Wang, S.B.K. Vrudhula. "An investigation of power delay trade-offs for dual V_t CMOS circuits," ICCD 1999.

[12] N. Azizi, et al, "Low-leakage asymmetric-cell SRAM," *ISLPED 2002*.

[13] C.K. Cheng, et al, *Interconnect Analysis and Synthesis*, Wiley and Sons, New York, 2000.

[14] S. Morton. "Inductance: implications and solutions for high-speed digital circuits - on-chip signaling," *ISSCC 2002*.

[15] K. Lepak, et al, "Simultaneous shield insertion and net ordering under explicit RLC noise constraint," *DAC 2001*.

[16] R. Rudell, A. Vincentelli. "Multiple valued minimization for PLA optimization," *IEEE CAD*, Sep. 1987.

[17] http://www.specbench.org/osg/cpu2000/CINT2000/. Spec CINT 2000 Benchmarks.

[18] L. Li, et al, "Adaptive error protection for energy efficiency," *ICCAD 2003*.