Odd/Even Bus Invert with Two-Phase Transfer for Buses with Coupling

Yan Zhang, John Lach, Kevin Skadron, Mircea R. Stan

University of Virginia
Charlottesville, VA 22904

{zhangyan, jlach, skadron, mircea}@virginia.edu

ABSTRACT

The coupling capacitances between on-chip bus lines become dominant in deep-submicron technologies. Coding to reduce the *switching activity* of the individual lines was enough to reduce power on buses in older technologies, but new coding techniques that reduce the *coupling activity* between lines are needed for deep-submicron buses. One such coding technique uses the simple observation that coupling capacitances are always charged and discharged by activity on neighboring bus lines, where one line has an odd number and the other has an even number (if bus lines are numbered "in-order"). We thus propose to reduce the coupling activity by independently controlling the odd and even bus lines with two separate lines, the *Odd Invert*, and *Even Invert* line, respectively. We obtain significant reductions in power simply by comparing the coupling activity for the four possible cases of the Odd and Even Invert lines (00, 01, 10, 11), and then choosing the value with the smallest coupling activity to transmit on the bus. Even after encoding, the coupling activity for a pair of bus lines is still strongly dependent on the data. In particular the toggling sequences 01→10 and 10→01 result in 4 times more coupling energy dissipation than other coupling events. We thus propose a targeted Two-Phase transfer in order to reduce total power only on the pairs of lines that carry such toggling events.

Categories and Subject Descriptors

B.7.1 [**Hardware**]: Integrated Circuits – *types and design styles.*

General Terms

Algorithms, Performance, Design.

Keywords

Coding for low-power I/O, Bus Invert, buses with coupling.

1. INTRODUCTION

In recent years low-power and power-aware design has become a driving force in the semiconductor industry. This is mainly due to the remarkable success and growth of the portable electronics industry and to the growing cost of the heat dissipation solutions for high-performance systems. As CMOS processes scale to sub-micron dimensions, power associated with system buses and the I/O accounts for a large portion of the total system power. Many encoding schemes have been presented to reduce the power

dissipation on buses. The Bus-Invert method [1] can be applied to encode buses without prior knowledge of data statistics. To encode signals with highly correlated access patterns like address buses, the T0 method [2] and working zone method [3] have been proposed. In [7], partial bus-invert was proposed to meet the requirement of application-specified systems and avoid unnecessary inversions. In deep sub-micron technologies the coupling capacitances are becoming dominant for the total energy dissipation on buses. The recently proposed Transition Pattern Coding (TPC) [4] tries to minimize inter-wire transitions, however, due to complex encoding and decoding structures, the application of this scheme to real systems may be limited. In [8], a permutation-based encoding scheme that reorders the input data sequence was proposed to reduce the coupling transitions, however this is only effective in some very special cases.

In this paper, we propose the Odd/Even Bus-Invert as a practical scheme to minimize the coupling transitions on general-purpose deep-submicron buses. Augmenting this scheme we also propose the Two-Phase transfer method to further reduce power.

2. SELF-TRANSITIONS AND COUPLING TRANSITIONS

Self-transitions are defined as transitions on the capacitance between a bus line and the substrate (ground) while *coupling transitions* are defined as transitions on the capacitance between adjacent lines [4]. Figure 1 shows a simplified bus model with coupling (ignoring all the resistances). Cs is the self-capacitance from each bus line to ground; Cc is the coupling-capacitance between two adjacent lines. There has been some confusion in the literature about the difference between power *consumption* and power *dissipation* on buses with coupling, so here we explain the difference and give several examples.

The average power *consumption* on the bus is given by:

$$P_{avg} = (\alpha'_s \cdot C_s + \alpha'_c \cdot C_C) \cdot Vdd^2 \cdot f \qquad (1)$$

where α'_s is the number of average self-transitions per bus cycle and α'_c is the number of average coupling-transitions per bus cycle; only the *charging* transitions that require current flow from the power supply are being counted. For example only the 0→1 transitions are counted as self-transitions for power consumption.
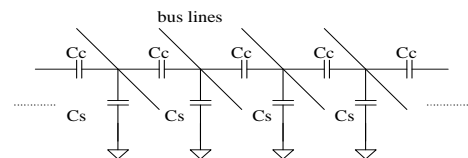

Figure 1. Bus model with self- and coupling-capacitances.

Similarly, the average power *dissipation* on the bus is given by:

$$P_{avg} = \frac{1}{2}(\alpha_s \cdot C_s + \alpha_c \cdot C_C) \cdot Vdd^2 \cdot f \qquad (2)$$

This time all transitions, charging or discharging, need to be counted. Because of this, α_s and α_c used to compute power dissipation are different than $\alpha{'}_s$ and $\alpha{'}_c$ used to compute power consumption. For example both the $0 \rightarrow 1$ and the $1 \rightarrow 0$ transitions need to be counted as self-transitions for power dissipation.

Figure 2 shows the charging and discharging of self-capacitances; Table 1 shows the power analysis for self-transitions.
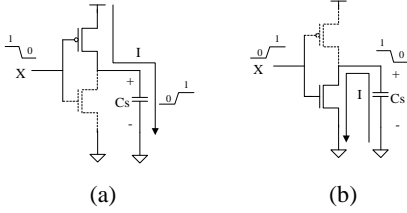

(a)　　　　　　　　(b)

Figure 2. (a) Charging (b) Discharging for self-capacitance.

Sequence of bits	Event	Initial stored energy	Final stored energy	Energy dissipated	Energy consumed
$1 \rightarrow 0$	charge	0	$C_sV^2/2$	$C_sV^2/2$	C_sV^2
$0 \rightarrow 1$	discharge	$C_sV^2/2$	0	$C_sV^2/2$	0

Table 1. Power analysis for a self-transition.

The charging and discharging of coupling capacitances require more cases. Figure 3 shows the possible cases of charging, discharging, and toggling. *Toggling* is defined as the case where adjacent lines switch simultaneously in opposite directions.
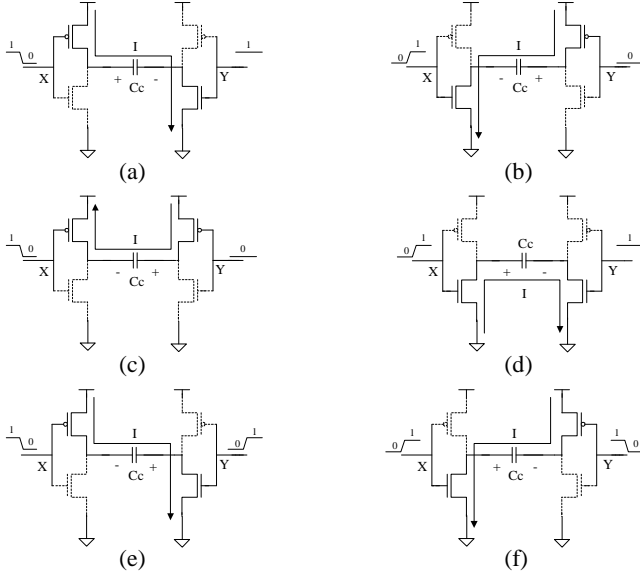

(a)　　　　　　　　(b)

(c)　　　　　　　　(d)

(e)　　　　　　　　(f)

Figure 3. (a), (b) Charging, (c), (d) Discharging, and (e), (f) Toggling for a coupling-capacitance.

As shown in Table 2, toggling is particularly expensive in terms of power. A toggling event represents twice more energy *consumption* than a charging transition, and four times more energy *dissipation* than a charging or discharging transition.

Sequence of bits	Event	Initial stored energy	Final stored energy	Energy dissipated	Energy consumed
$00 \rightarrow 00$	-	0	0	0	0
$00 \rightarrow 01$	charge	0	$C_cV^2/2$	$C_cV^2/2$	C_cV^2
$00 \rightarrow 10$	charge	0	$C_cV^2/2$	$C_cV^2/2$	C_cV^2
$00 \rightarrow 11$	-	0	0	0	0
$01 \rightarrow 00$	discharge	$C_cV^2/2$	0	$C_cV^2/2$	0
$01 \rightarrow 01$	-	0	0	0	0
$01 \rightarrow 10$	toggle	$C_cV^2/2$	$C_cV^2/2$	$2C_cV^2$	$2C_cV^2$
$01 \rightarrow 11$	discharge	$C_cV^2/2$	0	$C_cV^2/2$	0
$10 \rightarrow 00$	discharge	$C_cV^2/2$	0	$C_cV^2/2$	0
$10 \rightarrow 01$	toggle	$C_cV^2/2$	$C_cV^2/2$	$2C_cV^2$	$2C_cV^2$
$10 \rightarrow 10$	-	0	0	0	0
$10 \rightarrow 11$	discharge	$C_cV^2/2$	0	$C_cV^2/2$	0
$11 \rightarrow 00$	-	0	0	0	0
$11 \rightarrow 01$	charge	0	$C_cV^2/2$	$C_cV^2/2$	C_cV^2
$11 \rightarrow 10$	charge	0	$C_cV^2/2$	$C_cV^2/2$	C_cV^2
$11 \rightarrow 11$	-	0	0	0	0

Table 2. Power analysis for a coupling-transition.

In general only the power consumption or only the power dissipation needs to be calculated, as the two are equal on average, even if their instantaneous values are different. Counting only the charging transitions will result in the power consumption, while counting all transitions will result in the power dissipation.

Previous work in coding for buses with coupling [4], [6], has tried to derive codes by looking at power consumption (considering charging transitions). From previous work with the original Bus-Invert [1] we found that coding to minimize power dissipation (counting all transitions), can be more intuitive and lead to a better understanding of alternatives; it is the approach taken here.

3. ODD/EVEN BUS-INVERT

Coupling transitions happen between adjacent lines, so it is intuitive that if we can handle the odd and even lines separately we may be able to reduce the coupling transitions. We propose the Odd/Even Bus-Invert (OE-BI) to tackle the coupling problem this way. Somewhat similar to the original BI scheme [1], Odd/Even Bus-Invert will use two extra lines to indicate the inversion of the odd lines, or of the even lines, respectively. There are four possible cases with two invert-lines: no bus lines are inverted (00), only odd lines are inverted (10), only even lines are inverted (01), or all lines are inverted (11). Unlike the regular BI case, determining the optimal encoding for the two invert-lines of OE-BI is more difficult. In the first version of the OE-BI scheme, which we call the *Calculated Odd/Even Bus-Invert*, we explicitly compute the coupling transitions for all four possible cases. We can then choose the case with the minimum number of coupling transitions as the encoding pattern to transmit over the bus. Figure 4 shows a block diagram implementation for the Calculated OE-BI scheme for an 8-bit bus with two invert lines; $x(n)$ is the current value on the bus and $x(n+1)$ is the next data,

$x(n)(00)$ means none of the bus lines are inverted, $x(n)(10)$ means only the odd bus lines are inverted, $x(n)(01)$ means only the even bus lines are inverted, $x(n)(11)$ means all bus lines are inverted.
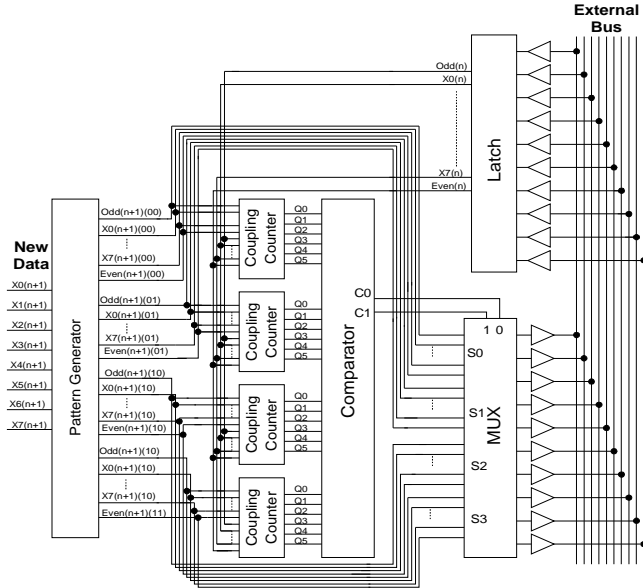

Figure 4. Calculated OE-BI structure

The coupling transitions for the four different cases are computed with four different counters. A coupling counter consists of a coupling estimator and an adder as shown in Figure 5. The coupling estimator has three possible outputs: "00" for no coupling transitions, "01" for 1 coupling transition (charging or discharging), and "10" for 4 coupling transitions (toggling). The four adders then combine the outputs from the corresponding coupling estimators. Finally, the comparator chooses then the lowest case to be transmitted.
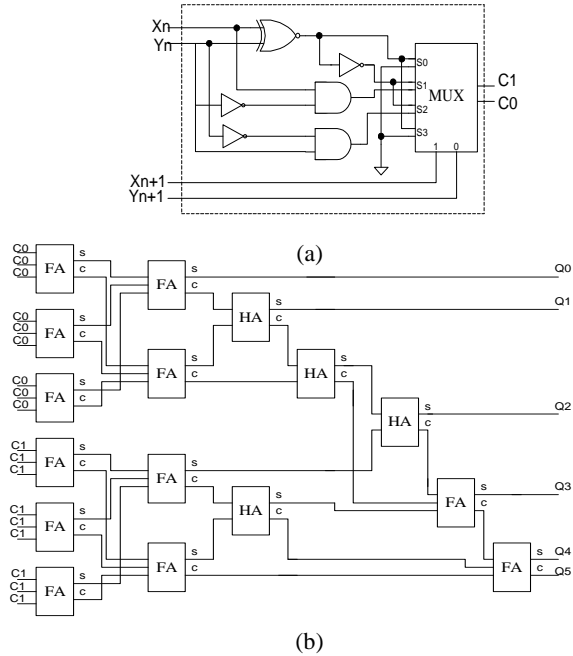

(a)

(b)

Figure 5. Coupling Counter: (a) Coupling estimator, (b) Adder

A simpler solution than the Calculated OE-BI is a direct extension of the regular Bus-Invert with two invert-lines, which we call *Simple OE-BI*. Unlike the Calculated OE-BI, which counts coupling transitions, the Simple OE-BI only counts self-transitions. Then, if the number of self-transitions on the odd lines is more than half, the odd lines are inverted, and if the number of self-transitions on the even lines is more than half, the even lines are inverted. The simple OE-BI is a less accurate encoding scheme and as it does not take the actual coupling transitions into account, but has very little hardware overhead. The scheme is similar to the Partial BI [7] and the statistics are the same as for a regular BI scheme with the bus partitioned into two halves.

4. TWO-PHASE TRANSFER METHOD

Observing the cost of coupling transitions, we can see that there are only three possible values: 0, 1 and 4. The two "toggling" cases, $10 \rightarrow 01$ and $01 \rightarrow 10$, have a very high cost (4) in power dissipation. The idea for the *Two-Phase Transfer Method* (TPTM) is then to eliminate such toggling events at the expense of an extra delay. Figure 6 shows a block diagram implementation of TPTM.
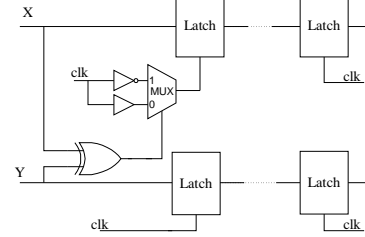

Figure 6. Two-phase transfer method

If the two adjacent lines x and y have opposite transitions, x is delayed by half cycle so that the transitions are not simultaneous (no toggling). Thus TPTM transforms a toggling event into a charging followed by a discharging event (or vice-versa), reducing the coupling cost from 4 to 2. For example, assuming that the current value of xy is 01, and that the next data of xy is 10, the number of coupling transitions would be 4. Applying TPTM, the x line is delayed by half cycle. The sequence of data now becomes $01 \rightarrow 00$ followed by $00 \rightarrow 10$, and the number of coupling transitions is reduced to 2, for 50% savings. To simplify the implementation, we always delay the x line when x and y have different values, even if there is no toggling event.

5. EXPERIMENTS AND CALCULATED RESULTS

We implemented the different schemes as mixed RTL/transistor level circuit schematics with explicit bus capacitances. We then ran a combination of simulations to obtain power savings estimations, and theoretical calculations to validate the results.

For theoretical validation we use information theory results [4]. The Time Averaged Expected Energy consumption (TAEE) [4] is given by the equation:

$$E_a = \frac{1}{2^m} b_0^T \cdot (T \bullet C) \cdot \underline{1} \tag{3}$$

Assuming the input data is independent and uniformly distributed, E_a is the average number of coupling transitions per bus cycle and m is the number of bus lines. As described in [4] several matrices are necessary. C is the cost matrix of the scheme where a matrix

entry C_{ij} equals $\alpha \blacksquare \lambda$, where α is the coupling transition between bit-patterns i and j, and λ is the ratio of C_c and C_s. T is the transition matrix. If the encoding rule allows a transition between bit-patterns i and j, T_{ij} equals 1, otherwise T_{ij} equals 0. P is the probability matrix which equals $(1/2^m)T$. The operator ($\bullet$) is the Hadamard product and b_0 is the left eigenvector of matrix P corresponding to the eigenvalue 1 [4].

	Self-transitions		Coupling-transitions		TPTM
	Calc.	Exp.	Calc.	Exp.	
Uncoded	4	4.001	7.0	7.046	5.775
Bus-Invert	3.3	3.272	5.8125	5.832	5.219
Simple OE-BI	3.1	3.126	5.625	5.624	4.743
Calculated OE-BI	3.7	3.697	4.45	4.451	4.074

Table 3. : Calculated and experimental results for an 8-bit bus with uniform data

Table 3 shows the calculated (TAEE) and experimental (simulation) results for OE-BI on an 8-bit bus (plus the Odd and Even invert lines). The input data sequence is independent and uniformly distributed. The reduction in coupling transitions is 36.8% for Calculated OE-BI, 20.2% for Simple OE-BI, and only 17.2% for the original BI. The Simple OE-BI benefits the most from the two-phase transfer method as the reduction in coupling transitions for the Simple OE-BI becomes 32.7% with TPTM.

	BI		Simple OE-BI			Calculated OE-BI	
	Ts	Tc	Ts	Tc	TPTM	Ts	Tc
1.jpg	3.24	5.82	3.11	5.58	4.71	3.68	4.43
2.jpg	3.23	5.67	3.08	5.54	4.67	3.67	4.36
1.mp3	2.96	5.33	2.83	5.11	4.32	3.36	4.02
2.mp3	3.11	5.54	2.97	5.37	4.55	3.52	4.27
1.mpeg	3.15	5.23	3.00	5.08	4.57	3.63	4.17
2.mpeg	3.08	5.10	2.94	5.00	4.49	3.56	4.09
Avg. red. (%)	18	15	22	18	29	7	34

Table 4. Simulation results of 8-bit bus with multimedia files (Ts: self-transitions; Tc: coupling transitions per bus cycle)

Table 4 presents simulation results for several multimedia applications. The Calculated OE-BI results in an average 34% reduction of coupling transitions. By applying the TPTM, the Simple OE-BI shows reductions from 18% to 29%.
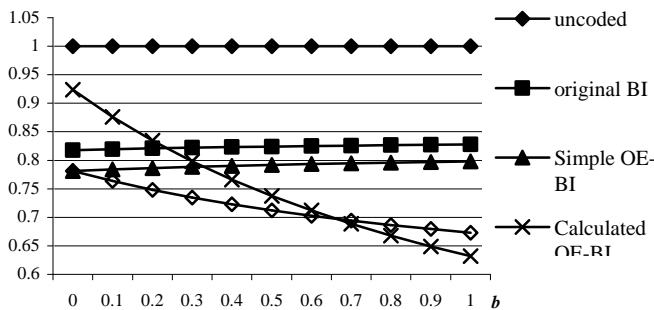

Figure 7. Normalized effective transitions for an 8-bit bus

Figure 7 shows the combined effect of self- and coupling-transitions on an 8-bit bus. It takes into account both the self and the coupling transitions, as well as different ratio of coupling to self-capacitances for the bus. Figure 7 plots the value of "effective transition" which equals $(1-b)\cdot\alpha_s + b\cdot\alpha_c$ and normalize it to the uncoded data. Here α_s is the average number of self-transitions, α_c is the average number of coupling transitions per bus cycle and b is such that the capacitance ratio $\lambda = b/(1-b)$. When b varies from 0 to 1, the capacitance ratio λ varies from 0 to ∞. As expected, the figure confirms that the Calculated OE-BI outperforms all other schemes for large λ (b close to 1).

We also implemented the calculated BI at the transistor level in a 0.18μm CMOS technology. At 300K and 1.8V supply voltage, we obtain 30% percent savings in coupling energy with only 5% overhead for the encoder circuit. The input sequence is pseudorandom data generated with a linear feedback shift register. Since the data is not strictly uniformly distributed, the power reduction only approximately matches the theoretical predictions.

6. Summary
We proposed the Odd/Even Bus-Invert to reduce the power dissipation by decreasing the coupling transitions on the bus. We also proposed the Two-Phase Transfer Method to further reduce the penalty for toggling events. The experimental results show that the Calculated OE-BI can reduce the coupling transitions by 36% compared to only 17% for the original BI. By applying TPTM, the Simple OE-BI can achieve a 32% reduction in coupling transitions with very little hardware overhead.

7. REFERENCES

[1] M. R. Stan and W. P. Burleson, "Bus-invert coding for low-power I/O," *IEEE Transactions on VLSI systems*, 1995.

[2] L. Benini, G. De Micheli, E. Macii and C. Silvano, "Asymptotic zero-tranistion activity encoding for address busses in low-power microprocessor-based system," *The Great Lakes Symposium on VLSI*, 1997.

[3] E. Musoll, T. Lang and J. Cortadella, "Working-zone method for reducing the energy in microprocessor address busses," *IEEE Transactions on VLSI systems*, 1998.

[4] P. P. Sotiriadis and A. Chandrakasan, "Low power bus coding techniques considering inter-wire capacitances," *Custom Integrated Circuits Conference*, 2000.

[5] S. Ramprasad, N. R. Shanbhag and I. N. Hajj, "Information-theoretic bounds on average signal transition activity," *IEEE Transactions on VLSI systems*, 1999.

[6] K. W. Kim, et al. "Coupling-driven signal encoding scheme for low-power interface design," *IEEE/ACM International Conference on CAD*, 2000.

[7] Y. Shin, S. Chae, and K.Choi, "Partial bus-invert coding for power optimization of application-specific systems," *IEEE Transactions on VLSI systems*, 2001.

[8] L.Macchiarulo, E.Macii, M.Poncino, "Low-energy encoding for deep-submicron address buses," *International Symposium on Low Power Electronics and Design*, 2001.