In-Order Pulsed Charge Recycling in Off-Chip Data Buses

Kimish Patel, Wonbok Lee, Massoud Pedram
University of Southern California
Department of Electrical Engineering
Los Angeles CA 90089
{kimishpa,wonbokle,pedram}@usc.edu

ABSTRACT

This paper presents in-order pulsed charge recycling to reduce energy consumption in an off-chip data bus. The proposed technique performs charge recycling by employing three steps: i) At the beginning of an off-chip data bus transaction, all bus lines which are expected to fall are connected to a common node, ii) next, one at a time and for a fixed period of time, each of the bus lines which are expected to rise are connected to the same common node to allow charge recycling, and finally, iii) regular data bus transaction is resumed by enabling the tri-state buffers to complete the remaining charging (discharging) of the rising (falling) bus lines. Experimental results in Hspice show that the proposed technique achieves 17.4% average energy savings in a 32 bit-wide data bus implemented in a 0.13μm technology with a 1.8V supply voltage.

Categories and Subject Descriptors

B.4.3 [**Interconnections (Subsystems)**]: Interfaces

General Terms

VLSI Design

Keywords

Power Dissipation, Charge Recycling, Data Buses

1. INTRODUCTION

Total power consumption in an electronic system comprises of power consumed in each system component e.g., the processor, memory, and bus drivers. Especially the power spent in off-chip communication tends to be a sizeable portion of total power consumption of the whole system due to highly capacitive nature of the off-chip buses. The effect is particularly severe in embedded systems with low power processors since these processors do not typically have large on chip caches, which subsequently results in higher off-chip memory traffic.

In the past, many researchers have focused on the off-chip bus power reduction due to large capacitive loads of these buses which tend to be orders of magnitude larger than their on-chip counterparts. This trend is likely to continue as the CMOS process technologies transition to 45nm node and below [1][2]. Some researchers have proposed charge sharing based ideas to reduce power in the off chip buses, which follow the conventional way of charge sharing, i.e., some fixed amount of charge is distributed among a fixed number of capacitive loads. Bus encoding techniques have proven quite effective in address buses because of the spatiotemporal locality of addresses that are transmitted on the bus. Alas these techniques are not as effective for data buses due to unpredictable nature of values that appear on these buses.

In this paper, we present a charge recycling technique that exploits the basic principle of charge sharing and maximizes the recycled charge in the off-chip data bus. The proposed technique does not need a priori information about the data stream in the bus, which is a must in any encoding-based techniques.

2. PRIOR WORK

Concepts of charge sharing and charge recycling are well-known and their application to energy efficient design of on- and off-chip bus architecture have been explored in the past. In [3], Khoo et al. reported the theoretically achievable energy savings of 47% for a 32-bit data bus and proposed an efficient charge-recovery technique. The authors of [4][5] extended Khoo's work and showed that a simple implementation of the charge recovery data bus is capable of reducing the average bus energy consumption by 28%. In [6], Sotiriadis et al. analyzed and implemented a charge-recycling technique for on-chip data bus. This is the closest to the idea proposed in this paper. However, these authors do not maximize the recycled charge. Moreover, on-chip bus does not have an adequate target structure for charge recycling since it tends to have repeaters which limit the scope of charge recycling only to the portion of the bus before repeaters. Analytical comparison between Sotiriadis' work, presented later, proves that more charge is recycled in our technique.

3. PULSED CHARGE-RECYCLING

3.1 Key Concept and Method

The proposed in-order *pulsed charge-recycling* technique (or PCR for short) attempts to maximize the recycled charge compared to the conventional charge recycling techniques of, say, reference [6]. From now on we will refer to the charge-recycling scheme of [6] as the *conventional charge recycling* technique (or CCR for short). Let us understand the difference between PCR and CCR with the aid of the example depicted in Figure 1.

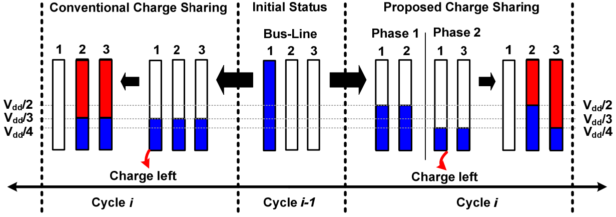

Figure 1. Comparison between CCR and PCR techniques.

Assume that we have three bus lines, 1, 2 and 3 with current (cycle *i*-1) data values '1' (represented by V_{dd}), '0' and '0',

This work was sponsored by a grant from the National Science Foundation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'08, May 4–6, 2008, Orlando, Florida, USA.

Copyright 2008 ACM 978-1-59593-999-9/08/05...$5.00.

respectively. Moreover, assume that the next set of values to be written on these lines (in cycle *i*) are ‘0’, ‘1’ and ‘1’, respectively. In the figure, the blue bar corresponds to the amount of charge which is present on the bit-line and the red bar corresponds to the amount of charge which needs to be extracted from supply voltage to bring the bus line to V_{dd}. With CCR, all three bus lines are shorted together, and thus, each of the bus lines will be charged/discharged to $V_{dd}/3$, resulting in ***66%*** of charge stored on line 1 being recycled.

Now consider the case when we allow charge recycling between bus lines 1 and 2 in a first phase, disconnect bus line 2 from 1, and subsequently enable another charge recycling between bus lines 1 and 3 in a second phase. Notice that each of these phases is long enough to allow full charge recycling and that the two phases are non-overlapping in time. When the charge recycling takes place in the first phase, bus lines 1 and 2 voltages converge to $V_{dd}/2$. When the subsequent charge recycling takes place in the second phase, remaining $V_{dd}/2$ of bus line 1 is shared with bus line 3, resulting in voltage level of $V_{dd}/4$ on both lines. As a consequence, total recycled charge is 75% of the original charge stored on line 1.

3.2 Energy Saving of PCR Compared to CCR

Consider an off-chip data bus with *N* lines, each of them with a total line to ground capacitance of *C*. Let us denote data on the bus as $X^{i-1}=[x_1^{i-1},\ x_2^{i-1},\ ...,x_n^{i-1}]$ in cycle *i-1* and as $X^{i}=[x_1^{i},\ x_2^{i},\ ...,x_n^{i}]$ in cycle *i*. Among the *N* lines, we denote the set of bus lines that will experience $1\rightarrow 0$ and $0\rightarrow 1$ transitions as $\mathcal{F}$ and $\mathcal{R}$, respectively. Furthermore, $\alpha=|\mathcal{F}|$ and $\beta=|\mathcal{R}|$.

Since the initial status of the $\mathcal{F}$ lines is V_{dd}, the amount of total charge stored on the data bus ahead of charge sharing is:

$$\alpha\cdot CV_{dd} \tag{1}$$

In the PCR scheme, charge sharing for all the $\mathcal{R}$ lines is done one at a time. In this scenario, when the first $\mathcal{R}$ line is connected to the $\mathcal{F}$ lines, it will thus receive an amount of charge equal to:

$$\frac{\alpha}{(\alpha+1)}\cdot CV_{dd} \tag{2a}$$

The charge stored on each of the $\mathcal{F}$ lines will drop from CV_{dd} to that given in Eqn. (2a). Next the first $\mathcal{R}$ line is disconnected from the $\mathcal{F}$ lines and a second $\mathcal{R}$ line is connected to the $\mathcal{F}$ lines. This second $\mathcal{R}$ line will receive an amount of charge equal to:

$$\frac{1}{\alpha+1}\cdot\left(\alpha\cdot\frac{\alpha}{\alpha+1}\cdot CV_{dd}\right)=\frac{\alpha^2}{(\alpha+1)^2}\cdot CV_{dd} \tag{2b}$$

Continuing in this manner until all $\mathcal{R}$ lines are sequentially connected for a fixed period of time to the $\mathcal{F}$ lines, the total transferred charge from the $\mathcal{F}$ lines to the $\mathcal{R}$ lines is equal to:

$$\sum_{j=1}^{\beta}\frac{\alpha^j}{(\alpha+1)^j}\cdot CV_{dd} \tag{3a}$$

During such a transaction on an off-chip data bus without charge recycling (No Charge Recycling, or NCR for short), we will have to consume $\beta\cdot CV_{dd}^{\ 2}$ of energy to raise the $\mathcal{R}$ lines from 0 to V_{dd}. In contrast, with the proposed charge recycling scheme, the total energy needed to raise the $\mathcal{R}$ lines to V_{dd} is only

$$\left(\beta\cdot CV_{dd}-\sum_{j=1}^{\beta}\frac{\alpha^j}{(\alpha+1)^j}\cdot CV_{dd}\right)\cdot V_{dd} \tag{3b}$$

In the CCR scheme whereby the $\mathcal{R}$ lines are connected to the $\mathcal{F}$ lines in one step (alternatively, the previously-connected $\mathcal{R}$ lines are not disconnected before the current $\mathcal{R}$ line is connected to the $\mathcal{F}$ lines), the total transferred charge from the $\mathcal{F}$ lines to the $\mathcal{R}$ lines is equal to:

$$\frac{\alpha\beta}{(\alpha+\beta)}\cdot CV_{dd} \tag{4a}$$

Therefore, the total energy needed to raise the $\mathcal{R}$ lines to V_{dd} in a conventional charge sharing scheme is

$$\left(\beta\cdot CV_{dd}-\frac{\alpha\beta}{(\alpha+\beta)}\cdot CV_{dd}\right)\cdot V_{dd} \tag{4b}$$

Notice that in general,

$$\sum_{j=1}^{\beta}\frac{\alpha^j}{(\alpha+1)^j}\geq\frac{\alpha\beta}{(\alpha+\beta)}$$

which indicates that the PCR is more effective than the CCR in achieving higher energy saving through the charge recycling idea. In particular PCR is superior to CCR for $\beta\geq 2$ and $\alpha\geq 1$.

3.3 Pulsed Charge Recycling Implementation

Consider an off-chip data bus where some bus lines are expected to undergo rising or falling transitions from cycle *i*-1 to *i*. The proposed technique targets $\mathcal{R}$ and $\mathcal{F}$ lines and consists of three steps: i) connect all $\mathcal{F}$ lines to a common node, ii) connect for a fixed period of time and subsequently disconnect each of the $\mathcal{R}$ lines to the same common node, one at a time, to enable charge sharing with the $\mathcal{F}$ lines, and finally, iii) resume regular data bus transaction by enabling the tri-state buffers to complete the remaining charging (discharging) of the $\mathcal{R}$ ($\mathcal{F}$) bus lines.

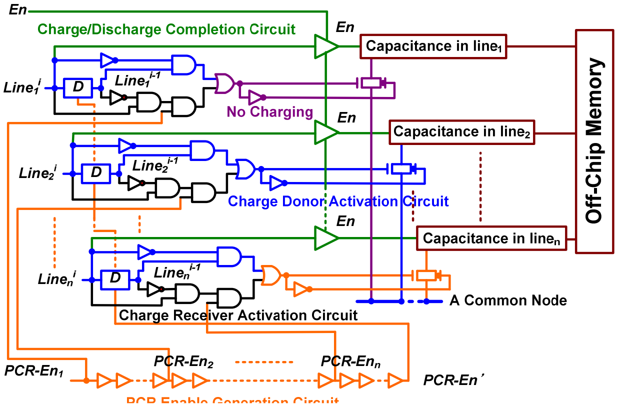

Figure 2. Proposed charge sharing structure for a data bus.

Figure 2 shows the circuit diagram for the proposed idea. For brevity, we show three lines in the data bus corresponding to each transition type: *$line_1^{\,i}$* which experiences no transition*,* *$line_2^{\,i}$* that is undergoing a $1\rightarrow 0$ transition, and *$line_n^{\,i}$* that is undergoing a $0\rightarrow 1$ transition. In accordance with three steps in the proposed technique, we add three functional blocks: 1) charge donor activation circuit, 2) pulsed charge recycling circuit, and 3) charge/discharge completion circuit.

3.4 Charge Donor Activation Circuit

During the first step, all $\mathcal{F}$ lines are connected to the common node. The logic block that performs this operation is shown in blue color in Figure 2. The circuit basically compares the previous data, from the D-latch, with the current data to be written on the corresponding bus line to detect a falling transition. Upon such detection, it turns on the transmission gate (TG) which connects these all $\mathcal{F}$ lines to the common node. Note that this operation does not need to wait for the arrival of the *PCR_En*$_1$ signal. Moreover, this operation does not perform any charge sharing by itself; instead it simply provides a path from the stored charge on the $\mathcal{F}$ lines to a common node from which a potential receiver could collect the charge.

3.5 Pulsed Charge Sharing Circuit

To maximize the charge recycled, each of the $\mathcal{R}$ lines should in order receive some charge from the common node. To facilitate this operation in the second phase, we perform two operations: 1) Detection of a rising transition. This is done by a logic block similar to charge donor activation circuit, called 'charge receiver activation circuit', drawn in black for each bus line in Figure 2. 2) Generation of the PCR enable signals (*PCR_En*$_i$) which connect each of the $\mathcal{R}$ lines to the common node to enable charge recycling. This is done by a logic block named 'PCR enable generation circuit', drawn in orange for each bus line in Figure 2.

To generate the enable signals for each bus line, we use a buffer chain as depicted in Figure 2. The buffer chain receives the *PCR_En*$_1$ signal as its input and shifts the signal such that no two enable signals (*PCR_En*$_i$ and *PCR_En*$_{i-1}$) corresponding to two different bus lines intersect one another. For a fixed size of TG, notice that the amount of time required to carry out full charge sharing is variable and depends on the number of donors. To limit the complexity, we decided to use a fixed period for charge sharing independent of the number of donors.

Notice that charge donation and reception occur only for the bus lines that are undergoing some transition from cycle i to $i+1$. The remaining bus lines, which experience no transition from the current to next cycle, have their charge sharing switches (TG) turned-off, and hence, the charge on these bus lines remains intact.

3.6 Charge/Discharge Completion Circuit

When the charge-recycling step is completed, to avoid shorting the bus lines, every charge sharing switch, i.e., TG, needs to be turned-off before the tri-state buffers are enabled. This is in turn achieved by applying the *PCR_En'* signal to the clock input of the D-latches. This essentially turns-off TGs for all $\mathcal{F}$ lines, by overwriting the previous data stored in D-latch with the current data. When this has been done, the delayed enable signal (*En*) to the tri-state buffer of each bus line activates the buffer to perform the remaining charging/discharging operation. Note that during these operations tri-state buffers on the receiver side are OFF.

3.7 Bus Line Grouping

The design presented in the previous section enables the charge receiver circuit of each bus line one after another in some order requiring exactly 32 charge sharing cycles for a 32-bit wide bus. To reduce this overhead, one may group the bus lines. We experimented with a group of 8 bus lines (8-line group) and a group of 4 bus lines (4-line group). In the case of 8-line groups (there are 4 such groups in a 32-bit bus), we enable charge receiver activation circuits for bus lines in the same bit position of the 4 different groups at the same time. For example *bus line*$_1$, *bus line*$_9$, *bus line*$_{17}$ and *bus line*$_{25}$ are enabled simultaneously by using the same exact PCR enable signal for all of them. (Notice however that only the ones that are in the $\mathcal{R}$ set will actually connect to the common node.) Since each group has 8 bus lines, our buffer chain has to generate 8 such PCR enable signals, each of which drives charge-receiver activation circuit of the corresponding 4 bus lines. Similarly, in the case of 4-line groups, buffer chain produces 4 such PCR enable signals.

The notion of group exists only for charge reception, i.e., only for the $\mathcal{R}$ lines. All the $\mathcal{F}$ lines bus lines are connected to common node regardless of the group they belong to. This enables us to receive charge even from the donors belonging to a different group. It is worthwhile mentioning that the way the bus lines are grouped can have noticeable impact on charge sharing. For example, if only 1 of 4 $\mathcal{R}$ lines is enabled for charge reception, at a given instance of time, than the recycled charge will be higher than the case when all 4 $\mathcal{R}$ lines are enabled simultaneously for charge reception. As a result, application-specific grouping based on the profiled data could result in larger savings.

4. EXPERIMENTAL RESULTS

The in-order pulsed charge sharing technique was applied to a 32 bit-wide data bus and implemented in a 0.13µm CMOS process with a 1.8V supply voltage. The power dissipation was measured with HSpice. Each line in the off-chip data bus was modeled to have 20pF of capacitance and 100Ω of resistance values referenced from [7]. This bus structure modeled in HSpice was configured (i.e., its drivers were appropriately sized) to run at 100MHz. Any delay penalty due to the PCR technique was calculated with respect to this baseline 10ns bus transaction delay. The PCR technique implemented corresponded to the eight 4-line group architecture explained above.

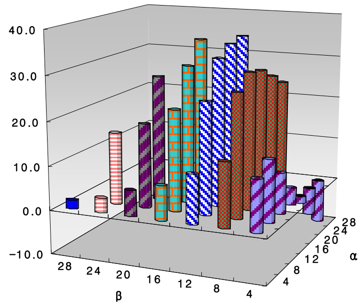

Figure 3. Energy savings with different numbers of rising and falling transitions.

4.1 Energy Saving Analysis

Figure 3 reports the energy savings achieved by the PCR technique compared to the bus architecture with no charge sharing (NCR). In each measurement, we used different combinations of α and β values. Note that a maximum of 32 bus lines can undergo transitions in any cycle, although most of the time the number of bit transitions is small and limited to the lower bits (explained later). Through HSpice measurements, which fully accounted for the power dissipation due to the added circuitry of the PCR architecture, we obtained an average of 17.4% energy savings of PCR over NCR. Note that higher energy savings were achieved when the two types of transitions were balanced.

Figure 4 shows the comparison between the proposed and the conventional charge sharing technique [6] using different values of α and β. As shown in the figure, the PCR technique outperforms the CCR technique in both 32 and 16 transitions.

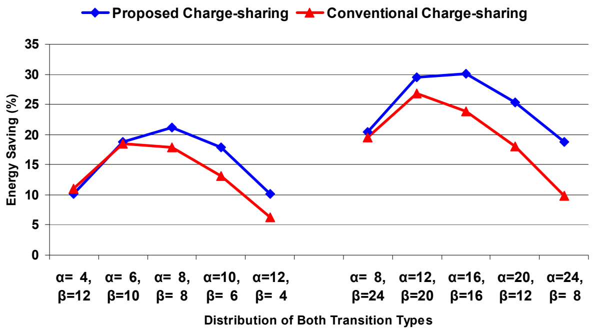

Figure 4. Energy savings comparison between the PCR and CCR.

Figure 5 shows voltage and current waveforms for some data bus transaction for both the NCR and PCR designs. In both designs, 32 bit-wide data bus has 16 falling transitions on bus lines 0 to 15, and 16 rising transitions on bus lines 16 to 31. Note that the PCR design corresponds to that of eight 4-line group charge sharing architecture, i.e., eight of the bus lines are enabled simultaneously for charge sharing. Let G_j denote a group of bus lines that are enabled together,

$$G_j = \{b_i \mid b_i \bmod 4 = j\} \quad \text{for} \quad i = 0,\dots,31 \ \text{and} \ j = 0,\dots,3 \qquad (5)$$

The NCR design takes 10ns while the PCR design takes 15ns to complete the same bus transaction, giving rise to 50% delay penalty. In contrast, the CCR design has 20% delay penalty compared to the NCR design. The current measurements demonstrate that the PCR (CCR) design consumes 35.4% (23.2%) less energy compared to the NCR design.

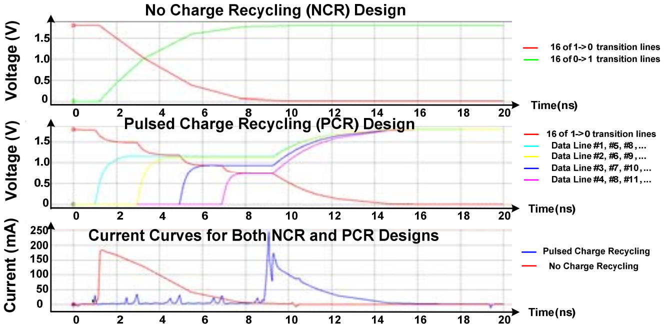

Figure 5. Electrical waveforms for various signals.

4.2 Off-Chip Bus Traffic Analysis

The aforementioned analysis does not account for the characteristics of the off-chip traffic in different applications. We profiled programs in terms of off-chip bus traffic from SPEC2000INT [8] and MediaBench [9] benchmarks using the eight 4-line group charge sharing architecture (cf. Eqn. 5). In Figure 6 we report the distribution of both transition types for a different 8-bit grouping of the bus lines:

$$G_j = \left\{b_i \mid 8(j) \le b_i < 8(j+1)\right\} \qquad (6)$$
$$\text{for} \quad i = 0,\dots,31 \ \text{and} \ j = 0,\dots,3$$

For each group, we report the number of falling and rising transitions per group of bits per application program. More precisely, for each group of bits, we provide two bar graphs: the first bar graph corresponds to the percentage of falling transitions in that group whereas the second bar graph corresponds to the percentage of rising transitions in the group. From this figure, we observe that most of the transitions (around 60 to 70%) occur in the first two groups, i.e., in lower 16 bus lines.

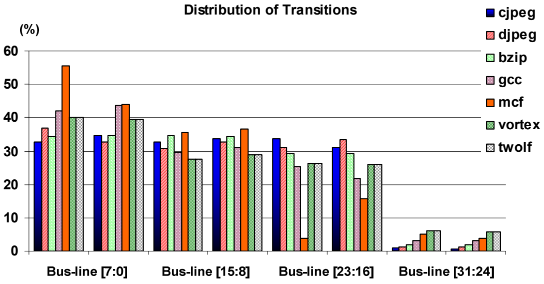

Figure 6. Distribution of Transitions.

Based on the data reported in Figure 6, we propose an improved PCR design where we only apply charge recycling to the lower 16 bus lines. Furthermore we change the grouping strategy within these 16 bits in order to reduce the delay penalty. The new charge sharing architecture uses eight 2-line groups, i.e. bus lines 1, 3, 5, …, 15 are connected to the common node in the first charge sharing cycle and bus lines 2, 4, 6,…, 16 are connected to the common node in the second charge sharing cycle:

$$G_j = \{b_i \mid b_i \bmod 2 = j\} \quad \text{for} \quad i = 0,\dots,15 \ \text{and} \ j = 0,1 \qquad (7)$$

The new energy savings for this case with four falling transitions in the lower half of the bus lines and four rising transitions in the upper half of the bus lines is 26.4% compared to NCR design. Furthermore delay penalty is reduced from 5ns (corresponding to 4 charge sharing cycles, cf. Figure 6) to 2.5ns (for 2 such cycles), resulting in only 25% delay penalty with respect to the NCR design (which takes 10ns). In contrast, compared to the NCR design, the CCR design produces 16.8% energy saving at the cost of 20% delay increase for the same case.

5. CONCLUSION

We presented a novel in-order pulsed charge recycling technique for off-chip buses. Our simulation shows that the proposed charge recycling technique achieves, on average, 17.4% and 5.4% energy savings compared to the NCR design and the CCR designs, respectively, while paying 50% delay penalty, compared to the 20% delay penalty of CCR, with respect to the NCR. Furthermore half-width PCR design achieves 26.4% and 16.8% energy savings compared to the NCR and CCR designs, respectively, while resulting in 25% delay penalty with respect to the NCR design.

6. REFERENCES

[1] N. Weste and D. Harris "CMOS VLSI Design: A Circuits and Systems Perspective," 3rd Edition, *Addison Wesley*, 2003.

[2] *Int'l Technology Roadmap for Semiconductors*, 2007.

[3] K.-Y. Khoo et al., "Charge Recovery on a Data bus," *Proc. of Int'l Symp. on Low Power Electronics and Design*, 1995.

[4] V. Lyuboslavsky et al., "Design of Data bus Charge Recovery Mechanism," *Int'l Conf. on ASIC/SOC*, 2000.

[5] B. Bishop et al., "Design Considerations for Data bus Charge Recovery," *IEEE Trans. on Very Large Scale Integration Systems*, Vol. 9, No. 1, Feb. 2001.

[6] P. P. Sotiriadis et al., "Analysis and Implementation of Charge Recycling for Deep Sub-micron Buses," *Int'l Symp. on Low Power Electronics and Design*, 2001.

[7] H-J. Shim et al., "Low-Energy Off-Chip SDRAM Memory Systems for Embedded Applications," *ACM Trans. On Embedded Computing Systems*, Vol. 2, No. 1, Feb. 2003.

[8] SPEC2000 at http://www.spec.org .

[9] MediaBench at: http://euler.slu.edu/~fritts/mediabench.