Efficient Power Reduction Techniques for Time Multiplexed Address Buses [*]

Mahesh Mamidipaka
Center for Embedded
Computer Systems
Univ. of California, Irvine, USA
maheshmn@cecs.uci.edu

Nikil Dutt
Center for Embedded
Computer Systems
Univ. of California, Irvine, USA
dutt@cecs.uci.edu

Dan Hirschberg
Information and Computer
Science
Univ. of California, Irvine, USA
dan@ics.uci.edu

ABSTRACT

We address the problem of reducing power dissipation on the time multiplexed address buses employed by contemporary DRAMs in SOC designs. We propose address encoding techniques to reduce the transition activity on the time-multiplexed address buses and hence reduce power dissipation. The reduction in transition activity is achieved by exploiting the principle of locality in address streams in addition to its sequential nature. We consider a realistic processor-memory architecture and apply the proposed techniques on the address streams derived from time-multiplexed DRAM addresses. Although the techniques by themselves are not new, we show that a judicious combination of the existing techniques yield significant gains in power reductions. Experiments on SPEC95 benchmark programs show that our encoding techniques yield as much as 82% in transition activity compared to binary encoding. We show that these reductions amount to as much 60% reduction in the off-chip address bus power. Also since the encoder/decoder add some power overhead, we calculate the minimum off-chip bus capacitance to the internal node capacitance ratio needed to achieve power reductions.

Categories and Subject Descriptors

B.6.1 [**Hardware**]: Design Styles—*Memory control and access*

General Terms

Algorithms, Design

Keywords

Low power, address encoding techniques, time-multiplexed addressing

1. INTRODUCTION

Off-chip buses have been identified as major contributors to total chip power. It is estimated that power dissipated on the I/O pads of an IC ranges from 10% to 80% of the total power dissipation with a typical value of 50% for circuits optimized for low power[10]. Various techniques have been proposed in the literature that encode the data before transmission on the off-chip buses so as to reduce the average and peak number of transitions.

Time multiplexed addressing is often used in SoCs for off-chip communication because of the reduced pin count in this scheme also reduces the effective chip cost. Contemporary DRAMs typically employ time-multiplexed addressing for communication. With the availability of cheaper DRAMs that employ advanced access features such as page mode, burst mode and Extended Data Out (EDO), DRAMs are highly used as main memories. Also because of their higher density, increased speed, and fewer address pins for communication, DRAMs will continue to be used in future system designs. Since off-chip address buses are one of the main contributors to power dissipation, it is necessary to employ encoding techniques that reduce transition activity in the addressing of DRAMs. While the techniques we propose can be used for any time multiplexed address buses, this paper focuses on the DRAM time-multiplexed address bus because of its wide use. In this paper, the term time-multiplexed DRAM addressing and time-multiplexed addressing are used interchangeably.

Although there are many encoding techniques in the literature that reduce transition activity on the off-chip address buses, they are intended for non-multiplexed address buses and hence cannot be applied directly to time-multiplexed address buses. Pyramid coding[4] was recently proposed for time-multiplexed DRAM addressing, that exploits the sequential nature in time multiplexed addresses. However, the DRAM addresses, in addition to the sequential nature, also exhibit the principle of locality property. We believe that a significant reduction in transition activity could be achieved if all the characteristics of the address stream are effectively exploited. Furthermore, our work is the first to evaluate the effectiveness of existing encoding techniques on realistic contemporary processor-memory architectures that employ DRAMs accessed through multi-level cache hierarchies.

We exploit the characteristics of the time-multiplexed address buses by applying a different encoding technique on each time multiplexed section of the complete address. In case of DRAMs (or any time-multiplexed addressing scheme),

[*] This work was supported by grants from DARPA(F33615-00-C-1632)

ISSS'02, October 2–4, 2002, Kyoto, Japan.
Copyright 2002 ACM 1-58113-576-9/02/0010 ...$5.00.

since the address is split into row and column addresses and sent over two cycles, we employ a different encoding technique that is most suitable for row and column addresses. It is important to note that contemporary DRAM based architectures use a cache hierarchy (with DRAMs at higher levels in memory hierarchy); in the presence of cache hierarchies, existing encoding schemes yield small off-chip transition reductions. In this paper, our proposed techniques for time-multiplexed addressing are evaluated on contemporary processor-memory architectures. While the individual encoding techniques themselves are not new, the contribution of this paper is presenting an approach that applies a judicious combination of the encoding techniques. We show that such an approach yields a significant reduction in address bus power dissipation, while incurring minimal area, delay, and power overhead. To the best of our knowledge, we have not seen any work which proposes heuristics based on a combination of encoding techniques on time multiplexed address buses for reducing power dissipation.

The paper is organized as follows: Section 2 reviews related work and Section 3 describes contemporary processor-memory architecture and identifies the characteristics of the addresses at memory hierarchies where DRAMs are typically employed. Section 4 presents our address encoding schemes based on these time multiplexed address characteristics. Section 5 shows the reduction in transition activity obtained by applying our techniques on several large benchmark programs. These results are compared with existing techniques to demonstrate the efficacy of our techniques in reducing the transition activity with minimal overhead in terms of area and delay. The actual power reductions obtained using these techniques are calculated and also the minimum bus capacitance to internal node capacitance ratio necessary for achieving power reductions are shown in Section 5. Finally, conclusions and future work are presented in Section 6.

2. RELATED WORK

There is a fairly large body of address encoding techniques proposed in the literature. Stan and Burleson proposed Bus-Invert (BI) coding[10] for reducing the number of transitions on an off-chip bus. In this scheme, if the transition count between the current data and previous data on the bus is more than half the bus width, the data is inverted before being transmitted on the bus. To signal the inversion, an extra bit line is used. To achieve further reductions, encoding techniques were proposed that focus on the sequentiality of addresses in instruction address bus. Gray code[11] ensures that, when the data values are sequential, there is only one transition between two consecutive data words. T0 coding[3] achieves transition activity reduction in sequential addresses by using an extra bit line along with the address bus. The extra bit line is set when the addresses on the bus are sequential, in which case the data on the address bus are not altered. When the addresses are not sequential, the actual address is put on the address bus. An enhancement of the T0 coding, called T0-C coding[1], was proposed by Aghaghiri et al. in which the redundancy of the extra bit line is eliminated using extra logic. Ramprasad et al. proposed INC-XOR coding[9], which reduces the transitions on the instruction address bus better than any other existing technique.

While these techniques solve the problem for instruction addresses, most system designs have unified off-chip address busses (instruction and data addresses sharing the same bus). Due to sharing, the sequentiality of addresses is reduced. To enhance the reduction in transition activity in such cases, techniques have been proposed that exploit the principle of locality of addresses. Mussol et al. proposed a Working Zone Encoding (WZE) technique[8], based on the principle of locality of the addresses on the bus. Self-organizing lists[6] based adaptive encoding techniques, Move-To-Front (MTF) and TRanspose (TR), were proposed by Mahesh et al.[7] to consistently reduce the transition activity without adding any redundancy in space or time. Also, adaptive encoding techniqueswere proposed by Ramprasad et al.[9] and Benini et al.[2] that can be applied to data exhibiting significant data correlation.

However, none of these encoding techniques can be applied directly to time-multiplexed addresses because the characteristics of time-multiplexed addresses are entirely different from non-multiplexed addresses. Recently, Cheng et al. proposed a Pyramid coding technique for optimizing switching activity on a multiplexed DRAM address bus[4]. Similar to Gray coding, Pyramid coding is an irredundant coding scheme (a one-to-one mapping between actual data and encoded data) that minimizes the switching activity primarily in the time multiplexed sequential addresses. The Pyramid technique also has significant delay overhead when the encoding is applied to larger address spaces.

Thus there is a need for techniques that reduce the transition activity for time-multiplexed address buses. In this paper, we address this issue of encoding schemes for time-multiplexed address buses with minimal power/delay overheads. Although the techniques are generic and can be applied to any time-multiplexed addressing scheme, the focus of this paper is on time-multiplexed DRAM addressing in the context of contemporary processor-memory architectures that employ DRAMs at different levels in the memory hierarchy. We propose schemes that exploit the principle of locality of addresses in addition to the sequential nature of addresses for further reduction in transition activity.

3. DRAM BASED CONTEMPORARY PROCESSOR MEMORY ARCHITECTURES

In this section, we show a contemporary DRAM based processor memory architecture and the characteristics exhibited by the DRAM addresses. All the experiments and results shown in the later part of the paper are based on this target architecture. Most contemporary processors (Eg: PowerPC 750, Intel Pentium III, R10000, AMD Athlon, Sun SparcIIi) employ a similar cache hierarchy as shown in Figure 1, though with different cache configurations. Typically these architectures have separate instruction and data caches at Level 1 (L1) and a unified instruction and data cache at Level 2 (L2) which communicates with the main memory. Because of their higher latencies, DRAMs are typically not used for cache memories. However, with increasing bandwidth and support for modes with higher throughputs, DRAMs are dominantly being used for main memories. In addition to the cache hierarchy, contemporary processors have virtual memory support (not shown in the figure) for larger address space. In this paper, we assume that the address space is sufficiently large to hold all the application data and program without any virtual memory support.

To evaluate the characteristics of addresses on the bus between the L2 cache and main memory, we ran several different application programs and SPEC benchmarks on the SHADE[5] instruction set simulator for different cache configurations. The cache configurations were based on several existing processors in the market. The following characteristics were observed on the address stream:

- The addresses between the L2 cache and main memory are considerably sequential even after applying burst mode for transfer of data between them whenever applicable. For example, a miss at an L2 cache level requires a number of sequential accesses to the main memory to refill the cache line depending on the L2 cache line size and the data bus width between the L2 cache and DRAM. In such cases, we assume that the memory controller initiates a burst mode, thereby requiring only the start address to be sent on the address bus to the DRAM. However, the percentage sequentiality of these addresses is dependent on the application program and also on the cache configurations at both level 1 and level 2. On average, for various cache configurations, several application programs, and including the burst mode feature in DRAMs, 14% of these addresses were seen to be sequential.

- These addresses still follow the principle of locality, both spatially and temporally. To determine whether the addresses still exhibit the locality principle between the L2 cache and main memory, the DRAM based main memory was replaced by another level of cache in the simulation setup. When application programs were run on this setup, we observed that the cache hits in the extra level of cache were significantly higher (ranging from 27% to 99.3% with a mean of 71% of total accesses) than the misses at this level.

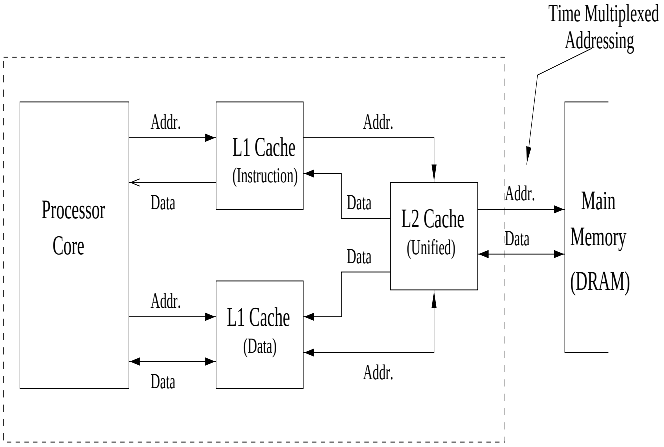

Figure 1: Block diagram showing memory architecture of contemporary processors

Although we used the SHADE instruction set simulator for simulations, we believe that similar address characteristics would appear when using other processors because the control and data flow of instructions for any application would be the same regardless of the target processor. Based on these characteristics, in the following section we propose techniques that significantly reduce transition activity on the time-multiplexed address bus.

4. PROPOSED ENCODING SCHEMES

In time-multiplexed addressing of DRAMs, the row address (Most Significant Half) is sent first and subsequently the column address (Least Significant Half) is transmitted over the bus. Since the characteristics exhibited by row and column addresses are different, the main heuristic behind the proposed encoding techniques is to apply different encoding to row and column addresses for achieving the reduction in transition activity. However, this does not entirely solve the problem. Since even when the encoded row and column addresses are time multiplexed, it could happen that the encoded addresses might yield a greater number of transitions than the unencoded time multiplexed addresses. To overcome this, the encoding techniques are chosen with a common objective of reducing the total number of ones in the encoded address so that the transition activity of the overall time multiplexed address is reduced.

While the proposed heuristic is generic and can be applied to any time multiplexed address bus this section we propose two schemes, the XOR-INCXOR and the MTF-INCXOR technique, specific to time-multiplexed DRAM addressing. The XOR-INCXOR technique focuses on maximizing the transition activity reduction in sequential and spatial locality based time multiplexed addresses and the MTF-INCXOR technique tries to exploit both the temporal and spatial locality in these addresses in addition to the sequential nature of addresses. It is important to note that our proposed encoding schemes do not add any redundancy in space or time (no extra bit lines or time cycles).

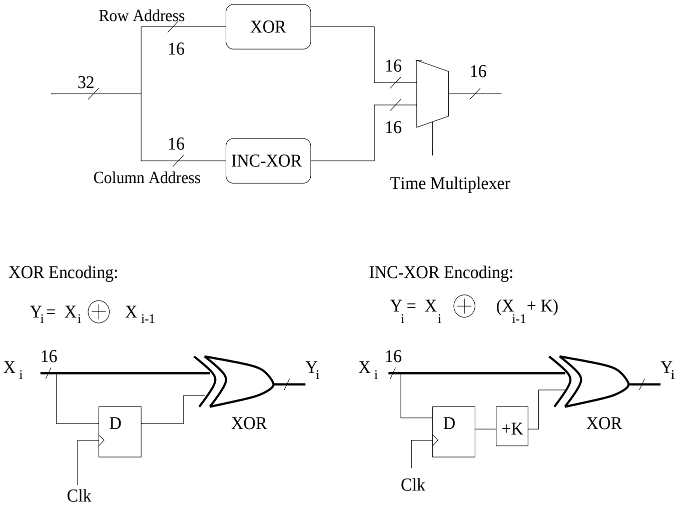

Figure 2: Block diagram showing the implementation of XOR-INCXOR encoding technique

4.1 XOR-INCXOR technique

As was observed in the previous section, a considerable number of DRAM addresses are sequential and exhibit the principle of locality. It can be noted that when the complete addresses are sequential or follow spatial locality, the column addresses would be within a small range of the previous address, but the row address would remain constant. The total transition activity is minimized when each encoding minimizes the number of ones in the encoded addresses. To achieve this, XOR encoding is applied to row addresses while the INCXOR technique is applied to column addresses. The block diagram of the implementation of the XOR-INCXOR technique is shown in Figure 2.

4.2 MTF-INCXOR technique

To exploit the principle of locality (temporal and spatial) in addresses for reducing the transition activity, Mahesh et al. proposed self-organizing based techniques - MTF and TR[7]. Move-To-Front (MTF) is an adaptive encoding technique in which encodings corresponding to each address space are maintained in a Look-Up Table (LUT) and the LUT is organized based on a heuristic for every new address. In this scheme MTF is applied to the row addresses since they exhibit the locality principle and INCXOR is applied to column addresses to minimize transitions due to sequential addresses. The block diagram of this encoding scheme is shown in Figure 3. More details on the MTF encoder implementation can be found in [7].

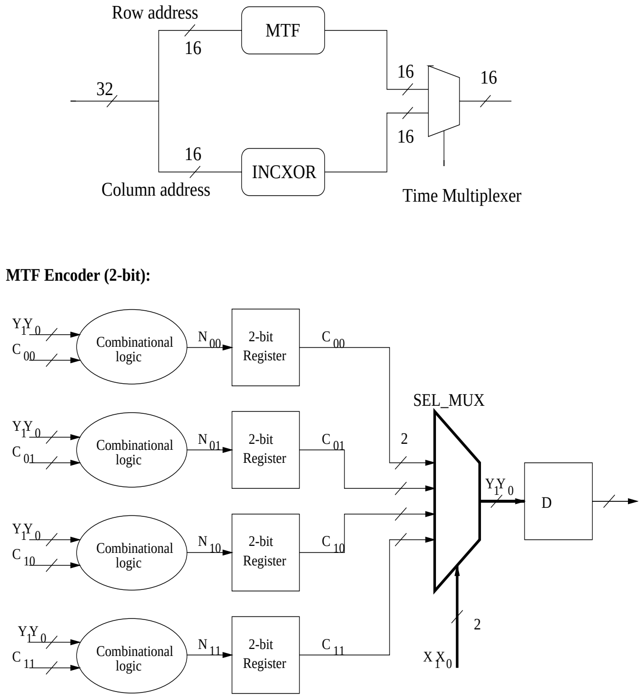

Figure 3: Block diagram showing the implementation of MTF-INCXOR encoding technique

Since the techniques are based on reducing the number of ones in the address stream, it is imperative to use transition signaling ($Y_i = Y_{i-1} \;\; xor \;\; X_i$, where Y is the outgoing bit stream and X is the incoming bit stream) on the encoded bit stream for further reduction in transition activity. The bold lines in the encoder implementations shown in Figure 2 and Figure 3 represent the delay overhead incurred on the critical path of the address lines because of the encoding techniques. It is important to note that this delay overhead is minimal in all the encoders. In the case of INCXOR and XOR, the delay overhead is the 2-input XOR delay and, in the case of the MTF encoder, the delay overhead is the delay of a 4-1 multiplexer from the select input. In the following section, the proposed techniques, with and without transition signaling, are compared with existing techniques for reduction in transition activity for the two different configurations of the target architecture described in Section 3. Also the encoding techniques are compared for overhead incurred in terms of the area, delay, and power.

5. EXPERIMENTAL SETUP AND RESULTS

We now present the results of experiments based on the proposed encoding schemes applied to the SPEC95 benchmark based address streams. The memory architecture shown in Figure 1 is used for generating the address streams. A time-multiplexed address scheme is used between the L2 cache and main memory. The cache configurations are based on two contemporary processors - PowerPC 750 and SparcIIi. Figure 4 shows the memory configuration of these processors. The cache configuration is given in terms of cache size, line size and associativity.

	L1 Cache		L2 Cache
	Instr. Cache	Data Cache	Unified
PowerPC 750	32KB, 32B, 8	32KB, 32B, 8	256KB, 128B(2 Sectors), 2
Sparc IIi	16KB, 32B, 2	16KB, 16B, direct	256KB, 64B(2 Sectors), direct

Figure 4: Memory configuration of processors

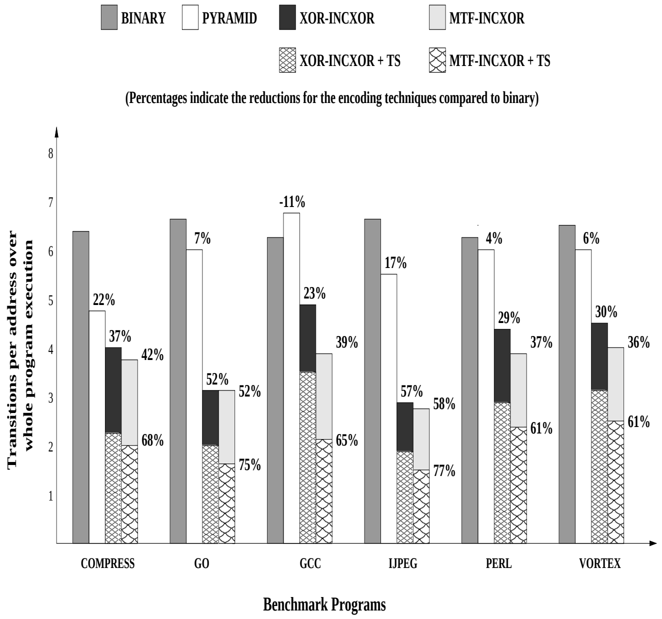

Figure 5: Plot showing average transitions per address of various encoding techniques for PowerPC configuration

We assume the presence of a simple controller to handle the address and data transfers between the L2 cache and DRAM. Because of the limited data bus widths, an L2 cache miss requires more than one transfer between the L2 cache and main memory. In such cases, we assume that the controller uses the DRAM burst mode to fetch the whole cache line, thus requiring only the beginning address to be sent to the DRAM. Note that, in the absence of burst mode, the techniques would be much more effective, since the sequentiality of the addresses increases tremendously. The address traces between the L2 cache and main memory for each memory configuration are collected using the SHADE[5] instruction set simulator on a SUN ultra-5 workstation. The comparison is made for the SPEC95 integer benchmark programs: compress, go, gcc, ijpeg, perl, and vortex. The experiments are done for the encoding

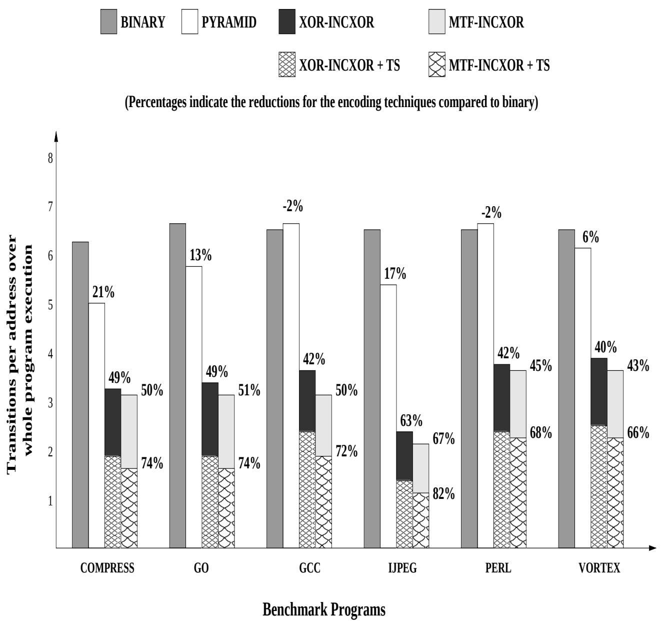

Figure 6: Plot showing average transitions per address of various encoding techniques for SparcIIi configuration

techniques: Pyramid, XOR-INCXOR, XOR-INCXOR+TS, MTF-INCXOR, and MTF-INCXOR+TS. '+TS' in XOR-INCXOR+TS and MTF-INCXOR+TS indicates that transition signaling is used on top of the corresponding encoding schemes. The address stream lengths of the benchmark programs over which the encodings were applied varied from 0.4 million to 32 million depending on the benchmark program. Figures 5 and 6 show the results of transition activity reductions obtained by applying the proposed techniques on these programs. The transitions per address, for each benchmark program and for each encoding technique, was calculated by dividing the total number of transitions with the total address stream length on the time multiplexed bus over the whole program execution. For MTF, the encoding was applied over widths of 2-bits (W=2 is used because the reductions are only marginally greater for higher bit width encodings).

From Figures 5 and 6, we infer that both the XOR-INCXOR and MTF-INCXOR techniques yield consistently significant reductions in transition activity and that these reductions compare favorably to those obtained from Pyramid coding. The reductions due to Pyramid coding were inconsistent (negative for gcc in both configurations and for perl in SparcIIi configuration). This could be because Pyramid coding focusing only on sequential addresses for reduction in transition activity. The percentage sequentiality of addresses in these benchmark programs varied from 4% to 28% with an average of 16%. However, the reductions for Pyramid coding were not proportional to the percentage sequentiality for the corresponding application. A possible reason could be that, the depending on the program, reductions obtained during the sequential accesses might have been nullified by the excessive transitions due to the encodings for the non-sequential addresses.

As expected, use of transition signaling on top of the proposed encoding techniques has yielded further reductions in transition activity. Among all the benchmark programs, ijpeg yielded the best reduction in transition activity, by 82% for the SparcIIi configuration. The average reductions in transition activity across all experimental programs for Pyramid, XOR-INCXOR, MTF-INCXOR, XOR-INCXOR+TS, and MTF-INCXOR+TS were 8%, 43%, 47%, 64%, and 70.5% respectively. Interestingly, the variation of transitions per address across different benchmark programs across both configurations was minimal (ranging from 6.3 to 6.7). The reductions obtained using MTF-INCXOR which exploits temporal locality of the address streams in addition to sequential and spatial locality was not significant. The average reduction obtained using the MTF-INCXOR+TS was only 6.5% more than that of the XOR-INCXOR+TS technique. We conjecture that this occurred because of the lack of significant temporal locality in the addresses of those application programs at post-L2 level.

Another important issue that needs to be considered is the area, delay and power overheads of different encoding techniques. Table 1 shows the area and delay overheads of different encoding techniques. The designs were synthesized using Synopsys Design Compiler on a 0.6μm LSI_10K library and the synthesis was done for a 4-bit address bus. The area overheads are given in terms of the number of library cells. The library cells include only the basic logic gates (e.g. NOR2, NAND2, XOR2, Flipflops etc.) and the macro cells (e.g. adders, comparators, etc.) are expressed in terms of these basic gates to obtain the final cell count for a specific implementation. Although the delay values might be slightly misleading because of the technology library used in the synthesis, the purpose is to at least make relative evaluations of various implementations.

The area overhead of MTF is more than that of any other encoding technique. The delay overhead for XOR and INCXOR techniques is minimal compared to those of other techniques. Note that these area overheads are for a 4-bit address bus. For 32-bit address buses, while the area will increase proportionally, the delay overhead for MTF, XOR, and INCXOR will remain the same. But for Pyramid coding, the delay overhead would also increase with the address width.

Table 1: Area and Critical Path Delay overheads for different Encoding Schemes

	Area		Delay	
	(# of lib. cells)		(ns)	
	Enc	Dec	Enc	Dec
MTF	84	110	2.3	2.3
INCXOR	30	30	0.96	0.96
XOR	8	8	0.96	0.96
Pyramid	36	48	2.1	2.6

Figure 7 shows the net power reductions obtained on a time-multiplexed address bus using the proposed schemes. The transition activity over the bus was calculated as the average of all the transition activities across all application programs and configurations over which experiments were conducted. The power consumption over the bus was calculated using the simple equation:

$$P_{bus} = 0.5 * C_{bus} * V^2 * f * T_{avg} \qquad (1)$$

where P_{bus} is the power dissipated on the address bus, C_{bus} is the bus capacitance, T_{bus} is the average number of transitions on the address bus per clock cycle, f is the frequency of operation and V the operating voltage. While the power was plotted for varying bus capacitance, the voltage and frequency of operation were assumed to be 1.35 V and 50 MHz respectively. To calculate the power overhead due to the encoder and decoder, the synthesized gate-level netlists were simulated using Synopsys over a large set of address traces and the average transition activity and average node capacitances in each encoder and decoder are used to calculate its corresponding dynamic power consumption. To account for leakage and short circuit power, the dynamic power was increased by 10% (leakage power is typically 10% of dynamic power) in the total power overhead calculation.

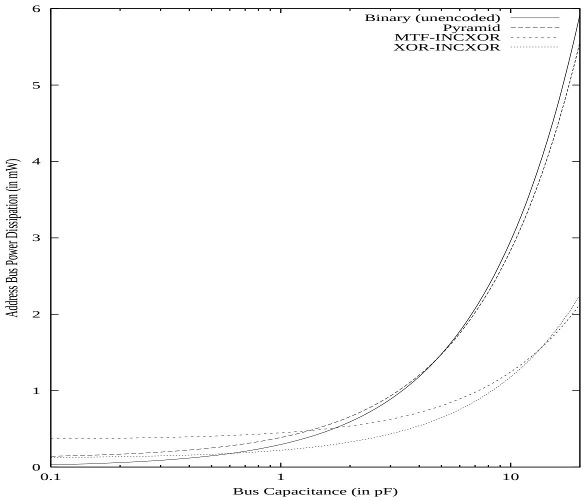

Figure 7: Plot of power reductions for various encoding techniques

As can be seen from Figure 7, the XOR-INCXOR technique yields power reductions when the bus capacitance is greater than 0.6pF. Similarly MTF-INCXOR gives reduction in power for bus capacitances greater than 1.7pF. The minimum bus capacitance needed for power gains for MTF-INCXOR is greater than that of XOR-INCXOR because of the higher transition activity and hence larger power overhead due to the encoder and decoder corresponding to MTF-INCXOR. Note that both XOR-INCXOR and MTF-INCXOR give similar reduction in transition activity on the time-multiplexed bus. However, when the bus capacitance is more than 13.8pF, MTF-INCXOR yields a better reduction in power than XOR-INCXOR techniques. Because of the significant overhead in Pyramid coding and smaller transition activity reductions the overall reductions in power due to Pyramid coding is minimal. For a typical off-chip capacitance of 10pF, the reduction in power is as much as 60% (MTF-INCXOR and XOR-INCXOR techniques yield 57% and 60% respectively).

It is worth noting that the minimum bus capacitance values indicated for power reductions, depends on the internal node capacitance (C_{node}) of the encoder and decoder. Since the node capacitance changes according to technology, we derive the metric (C_{bus}/C_{node}) to determine the applicability of the encoding techniques. Table 2 show these ratios corresponding to each encoding technique. The values indicate the minimum bus to node capacitance ratios needed to give power reductions. In the case of the INC-XOR technique, the minimum C_{bus}/C_{node} corresponding to architecture and technology needed to achieve power reductions is 40.4.

	Pyramid	MTF-INCXOR	XOR-INCXOR
C_{bus}/C_{node}	323.1	115.8	40.4

Table 2: Minimum C_{bus}/C_{node} ratio for power reductions for various encoding techniques

6. CONCLUSIONS AND FUTURE WORK

We present effective encoding schemes for time-multiplexed address buses, specifically in the context of DRAMs employed in contemporary processor-memory architectures accessed through multiple levels of cache hierarchy. Our experimental results demonstrate significant (up to 82%) reduction in address bus transition activity for the DRAMs. We present several options for encoding and analyzed their efficacy in this paper. Although the MTF-INCXOR+TS technique has higher area overhead, it gives the best average reductions (70.5%) over several benchmark programs. On the other hand, for area critical designs, the XOR-INCXOR+TS technique can be used. The average reductions obtained using the XOR-INCXOR+TS technique on various benchmark programs is 64%. The delay and power overhead for these schemes are smaller than that of previous approaches. We show the ratio of bus capacitance to the internal node capacitance should be at least 40.4 to achieve reductions in power. For typical off-chip capacitances (10pF), as much as 60% reduction in off-chip address bus power can be achieved using these techniques. Future work will involve better characterization of addresses on these multiplexed bus for specific applications and development of the framework that determines the best encoding techniques for those address characteristics. We also plan to develop efficient encoding techniques for other DRAM based architectures.

7. REFERENCES

[1] Y. Aghaghiri et al. Irredundant address bus coding for low power. In *ISLPED* , Huntington, CA, 2001.

[2] L. Benini et al. Architectures and synthesis algorithms for power-efficient bus interfaces. *IEEE Trans. on CAD of Circuits and Systems*, 19, 2000.

[3] L. Benini et al. Asymptotic zero-transition activity encoding for address buses in low-power microprocessor-based systems. In *GLSVLSI*, 1997.

[4] W.-C. Cheng et al. Low power techniques for address encoding and memory allocation. In *ASPDAC*, 2001.

[5] R. F. Cmelik and D. Keppel. Shade: A fast instruction-set simulator for execution profiling. Technical Report UW-CSE-93-06-06, 1993.

[6] J. Hester and D. S. Hirschberg. Self-organizing linear search. *Computing surveys*, 17:295–311, 1985.

[7] M. N. Mahesh et al. Low power address encoding using self-organizing lists. In *ISLPED*, Huntington, CA, 2001.

[8] E. Musoll et al. Working-zone encoding for reducing the energy in microprocessor address buses. *IEEE Trans. on VLSI Systems*, 6, 1998.

[9] S. Ramprasad, et al. A coding framework for low power address and data busses. *IEEE Trans. on VLSI Systems*, 7:212–221, 1999.

[10] M. R. Stan and W. P. Burleson. Bus-invert coding for low-power I/O. *IEEE Trans. on VLSI Systems*, 1995.

[11] C. L. Su et al. Saving power in the control path of embedded processors. *IEEE Design and Test of computers*, 1994.