Hardware-Only Compression to Reduce Cost and Improve Utilization of Address Buses*

JIANGJIANG LIU, NIHAR R. MAHAPATRA, and KRISHNAN SUNDARESAN†
Email: {jliu3, mahapatr, ks48}@cse.buffalo.edu

Abstract

Communication components (address, instruction, and data buses and associated hardware like I/O pins, pads, and buffers) are contributing increasingly to the area/cost and power consumption of microprocessor systems. To decrease costs due to address buses, we propose to use narrow widths for underutilized buses (hardware-only compression) to transmit information in multiple cycles. We analyze performance and power consumption overheads of hardware-only compression and investigate the use of "address concatenation" to mitigate performance loss and address offsets and XORs to reduce power consumption overheads.

1 Introduction

Higher instruction issue and clock rates in modern processors necessitate more *communication components* (address, instruction, and data buses and associated hardware like I/O buffers, pads, and pins), which contribute to increased area/cost and power consumption. On-chip buses scale relatively poorly in size compared to logic and this results in more area, which causes more individual wire capacitance and hence power consumption. Further, due to tighter spacing between higher aspect-ratio wires of buses in deep submicron regime, interwire capacitance effects become pronounced, resulting in even more power consumption. Increasing the number of pins and off-chip buses is difficult because it is limited by the surface area of the chip, whereas the amount of logic circuitry grows as the volume of the die. Also, off-chip buses have orders of magnitude more capacitance than on-chip circuit nodes and this exacerbates the power dissipation problem. Therefore, it is important to minimize area/cost and power consumption due to communication components.

This paper focuses on the area/cost and power consumption of address buses. Previous work in reducing cost relies on using narrow buses to transmit compressed addresses [4, 2]. Bus encoding schemes, on the other hand, strive to reduce power consumption by transmitting encoded (uncompressed) addresses that cause fewer transitions [1]. In earlier work, we presented a comprehensive analysis of the compression potential and power consumption of addresses at various levels of the memory system [5, 6]. In this paper, we propose *hardware-only compression* of underutilized address buses in which address information is transmitted on a narrow bus over multiple cycles to reduce area/cost and improve bus utilization, and possibly also lower power consumption, and study its impact on performance. Due to its simplicity, hardware-only compression is expected to have lower area and power consumption overheads at the sending and receiving ends compared to address compression and encoding methods. The rest of the paper is organized as follows. Section 2 describes the benefits and feasibility of using compressed address buses. Section 3 presents results of our simulations. Finally, Section 4 concludes the paper.

*Research supported by US National Science Foundation Grant # 0102830.

†J. Liu and N.R. Mahapatra are with the Dept. of Comp. Sci. & Eng. and K. Sundaresan with the Dept. of Electrical Eng., University at Buffalo, The State Univ. of New York, Buffalo, NY.

2 Hardware-Only Compression

We define *bus utilization ratio (BUR)* for a bus as the fraction of total cycles during which the bus is busy. BUR decreases as we move to buses away from the processor towards main memory because address traffic is generated only when misses occur at a previous memory level. In hardware-only compression, so called because only the bus hardware, but not the information transmitted on the bus, is compressed, a narrow bus is used to transmit information over multiple cycles. When an n-bit bus is compressed to a narrow bus of width $\lceil n \times \mathrm{BUR} \rceil$, utilization becomes almost 100% assuming uniformly-spaced addresses. Using a narrower bus will make it a performance bottleneck. A limited form of hardware-only compression to reduce the number of pins on DRAM is used by multiplexing row and column addresses.

Hardware-only compression of on-chip buses reduces their area and when it is applied to off-chip buses, the number of I/O pads and pins decreases, thus leading to lower die and packaging costs. Due to the smaller area, capacitance and thus power consumption may reduce. Further, by using area no more than a normal, uncompressed bus, a narrow bus can: (1) use greater spacing between bus lines, which will reduce interwire capacitance and hence delay, power consumption, and cross talk; and/or (2) use wider wires to reduce resistance and hence delay.

Using hardware-only compression entails performance, area, and power consumption overheads. Performance overheads can occur due to two reasons. First, since addresses occur nonuniformly over time, buffering will be required at the sending end and even then these buffers may fill up due to an address burst, necessitating pipeline stalls. Second, since addresses arrive at the receiving end later, cache/memory access is delayed and this delayed fetch may cause a dependent instruction to stall the pipeline. However, modern processors using dynamic scheduling can minimize the occurrence of such stalls by executing instructions out of order. Performance overheads can be mitigated in three ways. First, more buffering can be done at the sending end to avoid buffer-full related stalls. Second, addresses can be transmitted in w-bit groups from the high- to low-order end so that address tag and index fields are received quickly to allow early hit detection in the cache/memory at the receiving end. Third, when bus width is a non-integral fraction of the address width, *address concatenation* can be used, i.e., during the last cycle of transmission of an address, if there are unused bus lines and if the next address is available, a part of this next address (starting from the high-order end) can be transmitted. To indicate the presence or absence of a concatenated address in this last cycle, a *concatenation bit* can be used.

There will be area and power consumption overheads due to the additional logic required to transmit and receive addresses in parts, for extra address buffers at the sending end, and for supporting address concatenation if used. But

these overheads will not be much compared to the savings obtained by compressing long on-chip buses and off-chip buses. Address partitioning is not expected to take any additional cycles (although transmission of the address itself will take multiple cycles depending upon bus width). Due to the misaligned transfer of addresses in hardware-only compression, there will be some extra transitions and power consumption. To mitigate this problem, address offsets (offset w.r.t. previous address: mostly small magnitude) and address XORs (XOR w.r.t. previous address: mostly zeros) can be used. Of course, in these cases, the previous address needs to be saved at the receiving end in order to determine the new address. Finally, since logic scales better in size, speed, and power consumption than long bus wires, with technology improvements, the logic overhead to perform hardware-only compression and decompression will reduce relative to the savings in bus lines.

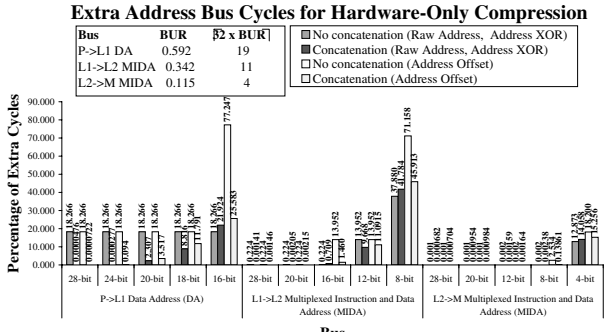

Figure 1: **Bus utilization and performance overheads for hardware-only address bus compression.** Extra cycle overheads for different buses compressed to different extents for raw, XOR, and offset addresses with and without concatenation.

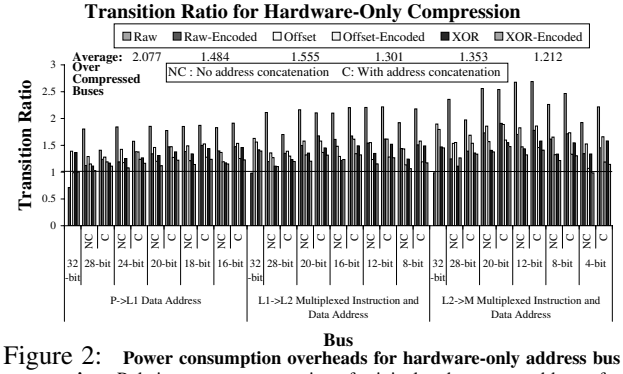

Figure 2: **Power consumption overheads for hardware-only address bus compression.** Relative power consumption of original and compressed buses for raw, offset, and XOR addresses with and without concatenation and bus-invert encoding [7].

3 Experiments and Results

We performed a trace-driven simulation of a memory system consisting of separate L1 instruction and data caches (16KB each, 32B block size, associativity four, write through, LRU replacement), a unified L2 cache (256KB, 64B block size, associativity four, write back, LRU replacement), and main memory using physical address traces obtained from runs of the first 10 million instructions of 4 integer (gcc, gzip, vortex, parser) and 4 floating-point (mgrid, lucas, mesa, sixtrack) SPEC CPU2000 benchmarks on a Pentium-III system with Redhat Linux (6.2) operating system [3]. We studied bus utilization and performance and power consumption overheads when hardware-only compression was applied to the data address bus between processor and L1 (P→L1) data cache and the multiplexed instruction and data address buses between L1 and L2 (L1→L2) and L2 and main memory (L2→M). The P→L1 instruction address bus is quite busy and hence should not be compressed.

A simple upper bound on the performance penalty due to compression of a bus is obtained by determining the number of extra cycles taken to transmit addresses as they occur compared to an original uncompressed bus. Fig. 1 shows the BUR values and extra cycles for the three address buses mentioned above for 32-bit raw and XOR and 33-bit offset (1 bit for sign and 32 for magnitude) addresses, both with and without address concatenation. As expected, BUR decreases as we move away from the processor—all buses were assumed to operate at the processor core frequency; a bus operating at, say, half the frequency will have twice the BUR shown. An n-bit address ($n = 32$ for raw and XOR and $n = 33$ for offset addresses) on a compressed bus of width w will take $\lceil n/w \rceil$ cycles for transmission without concatenation. That is the reason why the performance penalty without concatenation remains unchanged for, say, bus widths between 31 and 17. We find that address concatenation significantly reduces the number of extra cycles taken.

Next, to study power consumption overheads, in Fig. 2, we plot the *average transition ratio* (ratio of the total number of transitions on all bit lines of a compressed bus for all benchmarks to that for an original uncompressed bus carrying raw addresses) for different buses using varying amounts of compression, both with and without concatenation. We also provide results after bus-invert encoding was applied [7]. On compressed buses, offset and XOR addresses, particularly XOR, help in reducing power consumption overhead. If we use bus encoding, the power consumption overhead w.r.t. the original bus reduces even further. This gap can be reduced even more by using bus encoding schemes developed for, say, DRAM buses, such as the Pyramid code [1]. Typically, address concatenation results in little less power consumption compared to that without concatenation. Power consumption overheads increase somewhat as we move to buses away from the processor. There is no clear relationship between bus width for a given bus and power consumption overhead.

4 Conclusions

We proposed and analyzed the overheads (extra cycles required and power consumption) of a hardware-only compression scheme to reduce costs of underutilized address buses in the memory system by using narrow bus widths. We showed that address concatenation can mitigate performance loss due to hardware-only compression significantly. Finally, we also evaluated the effectiveness of address offset and address XOR streams instead of original addresses to reduce address bus power consumption overheads in narrow address buses.

References

[1] W.-C. Cheng and M. Pedram. Memory Bus Encoding for Low-power: A Tutorial. In *Proceedings of International Symposium on Quality of Electronics Design*, March 2001.

[2] D. Citron and L. Rudolph. Creating a Wider Bus using Caching Techniques. In *Proceedings of International Symposium on High Performance Computer Architecture*, pages 90–99, January 1995.

[3] Brigham Young University Trace distribution center. http://tds.cs.byu.edu/tds/.

[4] M. Farrens and A. Park. Dynamic Base Register Caching: A technique for Reducing Address Bus Width. In *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation*, May 1991.

[5] J. Liu, N.R. Mahapatra, K. Sundaresan, S. Dangeti, and B.V. Venkatrao. Memory System Compression and Its Benefits. In *Proceedings of the 15th Annual IEEE International ASIC/SOC Conference*, September 2002.

[6] N.R. Mahapatra, J. Liu, K. Sundaresan, S. Dangeti, and B.V. Venkatrao. The Potential of Compression to Improve Memory System Performance, Power Consumption, and Cost. In *Proc. of IEEE Performance, Computing and Communications Conference, Phoenix, AZ, USA*, April 2003. To appear.

[7] M.R. Stan and W.P. Burleson. Bus-invert Coding for Low-power I/O. *IEEE Transactions on VLSI Systems*, 3:49–58, March 1995.