

Low-Power Design of Page-Based Intelligent Memory

Mark Oskin, Frederic T. Chong, Aamir Farooqui, Timothy Sherwood, and Justin Hensley
University of California at Davis

Abstract

Advances in DRAM technology have led many researchers to integrate computational logic on DRAM chips to improve performance and reduce power dissipated across chip boundaries. The density, packaging, and storage characteristics of these intelligent memory chips, however, present new challenges in power management.

We introduce *Active Pages*, a promising architecture for intelligent memory based upon pages of data and simple functions associated with that data [OCS98]. We evaluate the power consumption of three design alternatives for supporting Active-Page functions in DRAM: reconfigurable logic, a simple processing element, and a hybrid combination of reconfigurable logic and a processing element. Additionally, we discuss operating system techniques to manage power consumption by limiting the number of Active Pages computing simultaneously on a chip.

1 Introduction

As processor performance grows, both memory bandwidth and power consumption become serious issues. At the same time, commodity DRAM densities are increasing dramatically. Many researchers have proposed integrating computation with DRAM to increase memory bandwidth and decrease power consumption. The operating and packaging characteristics of DRAM chips, however, present new constraints on the power consumption of these systems.

We introduce *Active Pages*, a promising architecture for intelligent memory with substantial performance benefits for data-intensive applications [OCS98]. An Active Page consists of a page of data and a set of associated functions that operate on that data.

To use Active Pages, computation for an application must be divided, or *partitioned*, between processor and memory. For example, we use Active-Page functions to gather operands for a sparse-matrix multiply and pass those operands on to the processor for multiplication. To perform such a computation, the matrix data and gathering functions must first be loaded into a memory system that supports Active Pages. The processor then, through a series of memory-mapped writes, starts the gather functions in the memory system. As the operands are gathered, the processor reads them from user-defined output areas in each page, multiplies them, and writes the results back to the array datastructures in memory. Simulation results show up to a 1000X speedup on such applications running on systems that use Active-Page memory over conventional memory.

This paper focuses upon ongoing work which evaluates the power consumption of Active-Page implementations. First, we

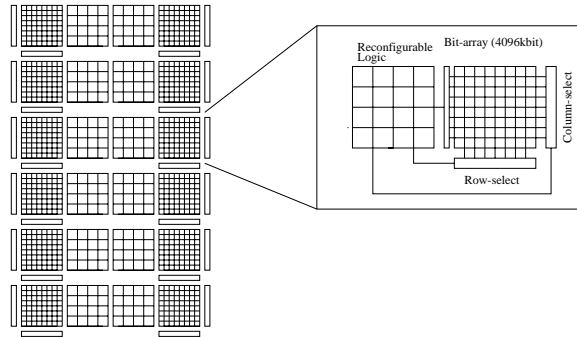


Figure 1: The RADram System

describe three alternative designs based upon reconfigurable logic, small processors, and a combination of the two. Then we present some power models and estimates for these designs. We also discuss software techniques for reducing power consumption. Finally, we conclude with some future directions for this work.

2 Design Alternatives

In this section we briefly present three architectures for Active Page memory systems. These architectures consist of RADram, or reconfigurable logic in DRAM [OCS98], an architecture consisting of only a small RISC or MISC [Jon98] like processing element near each page of data, and a hybrid architecture which provides a small processing element with reconfigurable logic processing capability.

Currently, within all of our designs we adopt a *processor-mediated* approach to inter-page communication which assumes infrequent communication. When an Active-Page function reaches a memory reference that can not be satisfied by its local page, it blocks and raises a processor interrupt. The processor satisfies the request by reading and writing to the appropriate pages. The processor-mediated approach reduces power by not requiring extensive inter-page communication networks, and global clocking.

2.1 Reconfigurable Logic

Our initial implementation of Active Pages focused upon the integration of DRAM and reconfigurable logic. For gigabit DRAMs, a reasonable sub-array size to minimize power and latency is 512Kbytes [I⁺97]. The RADram (Reconfigurable Architecture DRAM) system, shown in Figure 1, associates 256 LEs (Logic Elements, a standard block of logic in FPGAs which is based upon a 4-element Look Up Table or 4-LUT) to each of these sub-arrays. This allows efficient support for Active-Page sizes which are multiples of 512Kbytes.

Each LE requires approximately 1k transistors of area on a logic chip. The Semiconductor Industry Association (SIA) roadmap [Sem97] projects mass production of 1-gigabit DRAM

Acknowledgments: Thanks to Venkatesh Akella for his helpful comments. This work is supported in part by an NSF CAREER award to Fred Chong, by Altera, and by grants from the UC Davis Academic Senate. More info at <http://arch.cs.ucdavis.edu/RAD>

chips by the year 2001. If we devote half of the area of such a chip to logic, we expect the DRAM process to support approximately 32M transistors, which is enough to provide 256 LEs to each 512Kbytes sub-array of the remaining 0.5-gigabits of memory on the chip. DeHon [DeH96] gives several estimates of FPGA area, and existing prototype merged DRAM/reconfigurable logic designs demonstrate this to be a realistic design partitioning [M⁺97].

2.2 Processing Elements

An alternative architecture for Active Page implementations is that of a small RISC or MISC [Jon98] type processing element used to execute user level process functions in memory. Several low-power processing cores currently exist and Active Page designs can leverage the existing and future work in this area. However, a processing engine designed for Active Pages must satisfy two constraints: power and size. While high-performance low-power designs exist, such as the StrongARM [San96], their size makes them prohibitive for use in Active Page memory. For this study we assume a slightly modified Toshiba TLCS-R3900 series [NTM⁺95] processor. The current processor contains a 4k direct mapped instruction cache. Such a cache is not needed for the limited number of instructions an Active Page memory function contains. The calculations in this paper assume a 1k instruction cache. The remainder of the TLCS-R3900 processing engine is relatively standard, with a 1k data cache, 32 bit data-path, 32 general purpose registers and MIPS-like instruction set.

2.3 Hybrids

An alternative architecture from RADram or a small processing element, is a hybrid, which mixes a RISC-like processor with reconfigurable logic. Such an architecture is presented in [HW97]. Here we extend this notion by simplifying the processing element core, and shrinking the reconfigurable logic array. The hybrid architecture would consist of a simplified processing element described above in Section 2.2 and a smaller reconfigurable logic array than that presented for RADram in Section 2.1. The configurable logic array would be on the order of 128 LEs in size.

3 Power Consumption

In order to analyze the power consumption of the three alternative Active Page implementations described in Section 2, an analysis of each was performed based on analytical models and previously published results. In this section we present analytical models for power consumption for DRAM sub-arrays and reconfigurable logic arrays. Furthermore, we present estimations for the power consumed by an Active Page memory which utilizes off-the-shelf MIPS-like processing cores as page based functional units. Finally we estimate the power of a combined processor / reconfigurable logic array design. Throughout these analyses the target has been a 1 gigabit size DRAM array. Each design compromises a certain percentage of DRAM storage for functional logic. Each design has its own size and here we also estimate the relative percentage of logic to DRAM and thus the overall size of the resulting memory chip. While these designs are based on a 1 gigabit size DRAM, it should be kept in mind that the actual number of pages will most likely be a power of two, and thus either an

increased or decreased chip size will ultimately be necessary for non-power-of-two size designs.

3.1 DRAM model for Power Analysis

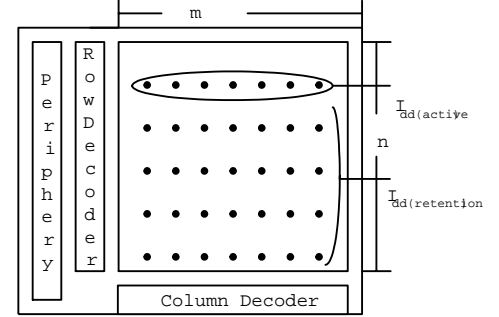


Figure 2: DRAM Model used in the analysis.

In this section we present a DRAM model for power estimation. Using this model we estimate the power consumption of a sub-array size used for gigabit size DRAMs. A DRAM block of m columns and n rows can be represented as shown in Figure 2[⁺95]. This figure shows the current flowing in the DRAM block. There are two major sources of power consumption in DRAMs:

1. Active current, which flows during the read operation.
2. Data retention current, which flows during re-charging the DRAM cells.

The active current is expressed by:

$$I_{dd} = (m \cdot C_d \cdot V_w + C_{pt} \cdot V_{int}) \cdot f + I_{dcp} \quad (1)$$

where C_d is the data line capacitance, V_w is the voltage swing at a particular data line, C_{pt} is the total capacitance of the CMOS logic and driving circuitry in the periphery, V_{int} is the internal supply voltage, f is the operating frequency, and I_{dcp} is the DC static current.

The data retention current is given by:

$$I_{dd} = (m \cdot C_d \cdot V_w + C_{pt} \cdot V_{int}) \cdot (n/t_{ref}) + I_{dcp} \quad (2)$$

where t_{ref} is the refresh time of cells in retention mode.

C_d	0.2pf
V_w	0.5V
m	4096
n	1024
f	20Mhz
t_{ref}	20ms
V_{dd}	1.65V

Table 1: DRAM parameters for 512Kbyte block sub-arrays.

Based on Equations 1 and 2, and technological parameters summarized in Table 1, it is possible to estimate power consumption for a single 512 kilobyte page of memory. Using

this base power per-page value we will extrapolate the power consumption of various Active Page implementations.

With mixed logic and memory designs, the effects of the C_{pt} term are negligible compared to the power consumption of the attached logic, and thus the term is assumed to be zero. Furthermore, at 20Mhz operation the DC static current I_{dcp} is negligible. Thus, the active current required to access a word in memory is given by:

$$I_{active} = (4096 \cdot 0.2pf \cdot 0.5V) \cdot (20 \times 10^6) = 8.1mA \quad (3)$$

The current due to data retention is given by:

$$I_{retention} = (4096 \cdot 0.2pf \cdot 0.5V) \cdot (1024/20ms) = 20\mu A \quad (4)$$

Thus, for a 1.65V power supply, the total power dissipated by a 512Kbyte DRAM sub-array is:

$$P_{total} = P_{active} + P_{retention} \quad (5)$$

$$P_{total} = (I_{active} + I_{retention}) \cdot V_{dd} \quad (6)$$

$$P_{total} = (8.1mA + 20\mu A) \cdot 1.65V = 13.4mW \quad (7)$$

The estimated power per 512Kbyte sub-array is slightly below existing DRAM power consumption[Tos98]. This is due to the reduced supply and swing voltages used in the calculation, and is expected in future DRAM memory designs. Using this calculation we can deduce that a storage-only gigabit size DRAM chip will consume $13.4mW \cdot 256 = 3.4W$ of power. The 1 gigabit DRAM power consumption, while greater than power consumed by existing DRAM designs, is well below the maximum power ratings predicted by the SIA Technology Roadmap [Sem97] for high-performance package technology.

3.2 Power for Reconfigurable Logic

In order to estimate the power consumption of the RADram architecture, we use the power estimation guide provided by Altera [Alt96] for the FLEX architecture of devices. We model power for all devices in an operational state. It is assumed that appropriate software control by the Operating System can be used to shutdown portions of the Active Page memory device during reconfiguration. The Altera power estimation guide predicts power for operational devices as:

$$Power = P_{int} + P_{io} \quad (8)$$

$$P_{int} = (I_{CCStandby} + I_{CCActive}) \times V_{dd} \quad (9)$$

$$I_{CCStandby} = I_{CC0} \quad (10)$$

$$I_{CCActive} = (K \cdot f_{max} \cdot N \cdot toggle_{avg})\mu A \quad (11)$$

where P_{int} is the power consumed internally by the FPGA, P_{io} is the power consumed driving external loads, $I_{CCStandby}$ is the current drain while idle, $I_{CCActive}$ is the current drain while active, K is a technology parameter which is dependent on the FPGA design, f_{max} is the operating frequency in megahertz, N is the number of logic elements, and $toggle_{avg}$ is the average percent of cells toggling at each clock.

For estimation purposes, we assume P_{io} to be negligible, and the standby current is equal to a constant $I_{CC0} = 0.5mA$. Furthermore, current technology has K values between 32 and 95. We assume a conservative view of future technology, with a K of 30. Using the RADram configuration presented in

Section 2.1, we let N equal 128 and the frequency of operation equal 100Mhz. Furthermore, we assume an average toggling percentage of 12.5 percent. Thus the active current is given by:

$$I_{CCActive} = 30 \cdot 100Mhz \cdot 256LEs \cdot 0.125 = 96mA \quad (12)$$

Therefore, power consumption of the reconfigurable logic for a single page in the RADram architecture is:

$$P = (.5mA + 96mA) \cdot 1.65V = 159mW \quad (13)$$

The RADram architecture trades DRAM memory storage for reconfigurable logic. This dedicates fifty percent of the chip space to DRAM and fifty percent to logic. Thus, a gigabit size DRAM actually stores only 0.5 gigabits of data (128 512 Kbyte pages). Using the power estimation presented in Section 3.1 we can estimate the power consumed by a 0.5 gigabit RADram memory chip to be:

$$P_{radram} = 128 \cdot (159mW + 13.4mW) = 17.1W \quad (14)$$

3.3 Power for Processing Elements

The current TLCS-R3900 series processors operate at 74Mhz and 3.3V and have a typical operating current of 110mA. We project the power consumption of the modified TLCS-R3900 series processor running at 100Mhz and 1.65V. Currently, roughly 20% of the power consumption is used by the instruction cache. By reducing the size of this cache by a factor of four, it is expected the power and area of the processing engine will be reduced. We expect the smaller instruction cache to consume one-third the power of the existing cache. Thus, the expected operating current will be 95mA. However, the faster clock will increase the current used to roughly 126mA. Finally, operating at 1.65V we expect the processor core to consume 208mW of power.

Despite this power consumption, the TLCS-R3900 series processing engine is quite large. Nearly 440k transistors are required to implement the base TLCS-R3900 core. It thus becomes impossible to continue with the 50% DRAM and 50% logic ratio when a processing element is used. In order to maintain the 512Kbyte sub-array size, a roughly 63% logic / 37% DRAM ratio is required. This means that only 92 512Kbyte pages will be available on our hypothetical Active Page memory chip. Therefore, total power consumption will be:

$$P_{pe} = 92 \cdot (208mW + 13.4mW) = 20.3W \quad (15)$$

3.4 Power for Hybrid PE/LE Design

Although the true cost of power consumption by a mixed design is not available without an actual design, we can estimate the power by mixing a TLCS-R3900 series processing element with a limited number of reconfigurable logic LEs. Here we have chosen a hybrid design with a limited number (128) LEs along side a fully functional processing element. Additional power will be required to interconnect these two processing engines, but here we assume this interconnection power to be negligible. Thus, each hybrid processing element consumes roughly one-half the power of the reconfigurable logic-only design added to the power consumed by the processing element. Combined these consume 288mW of power at 1.65V.

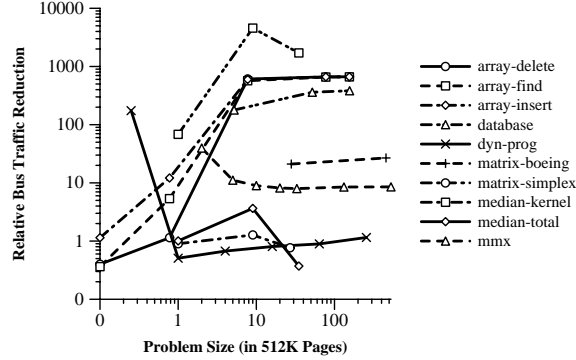


Figure 3: Experimental Results of RADram Relative Reduction in Processor / Memory Bus Traffic

The DRAM to logic ratio of this design is roughly 70% to 30%, thus yielding an Active Page device with only 78 pages of memory. Therefore total power consumption will be:

$$P_{hybrid} = 78 \cdot (288mW + 13.4mW) = 23.5W \quad (16)$$

3.5 Clock Distribution

In high-frequency, large-area designs the power consumed by the clock distribution and generation network cannot be neglected. The Active Page memory model permits an efficient solution to this problem. Active Page designs do not include hardware support for inter-page communication do not require a globally synchronized clock. An external clock can be used and a large amount of skew can be tolerated. Hence, the internal clock generation network need only consist of regeneration buffers, without an extensive and difficult to design network of clock distribution nodes.

Furthermore, reconfigurable logic designs usually are operated at the highest possible frequency permitted by the design. This suggests that the clock of each Active Page will be distinct, or else operate at some nominal baseline. Furthermore, software techniques discussed in Section 4.1 require variable clock speeds for all Active Page designs. For this reason, we propose that all clocks be generated locally for each page, using internal oscillator techniques. A global clock may be necessary to provide a baseline reference with which to self-time the locally generated clocks.

Given these design constraints, hardware inter-page communication facilities can still be constructed, if desired, using asynchronous logic techniques. Relevant techniques can be found in asynchronous processor cores [FGT⁺97], caches, and reconfigurable logic array designs [BEHB] [MA98].

3.6 Reduced Memory I/O

As is pointed out by Stan and Burleson [SB97] typically 10-15% of the power used by desktop systems, and 50% of the power in low power systems comes directly from external I/O. This is due largely to the fact that the busses used for off chip communications use higher voltages and have two to three orders of magnitude more capacitance. Substantial amounts of power are consumed every time the voltage on a bus line must be changed.

The number of transitions on the bus can be reduced in two ways: by encoding the bus to minimize the number of

transitions per transaction [SB97] [SB95] [WCF⁺94], or by reducing the total number of transactions on the bus. The use of Active Pages reduces I/O power by addressing the former.

In a traditional system, memory intensive sections of code produce poor cache hit rates which in turn results in large amounts of memory traffic. However in an Active Page system, memory intensive sections of an application are executed inside the memory chip thereby reducing the total number of external bus transactions.

Figure 3 shows relative reduction in bus-traffic for several applications executing on the RADram Active Page memory system. Each application is described in [OCS98], but here we note that a significant reduction in the total number of memory bus accesses can be achieved by the use of Active Pages. The actual reduction in bus traffic varies from application to application, however the general trend of larger problems sizes yielding more efficient bus use can be seen. The tailing off of power efficiency for very large problem sizes is due to the fact that at very large problem sizes the system becomes saturated, and the processor remains active throughout the computation.

3.7 Power Summary

Our estimates show that our initial RADram design using reconfigurable logic has the lowest power consumption of our three alternatives. A further advantage for RADram is increased chip yield due to fault-tolerant designs. The complexity and number of Active-Page functions, however, is limited by the amount of reconfigurable logic available for each page. The processing element and hybrid designs have more computational power. Furthermore, we expect to refine both of the latter designs to use less power. Refinements such as clock-gating, and asynchronous processor designs [FGT⁺97] will help reduce the power consumed by processor, and hybrid processor / reconfigurable logic processor element designs.

4 Software Power Management

Active Pages presents new design challenges under low-power constraints. One potential solution is to use software mechanisms which trade off performance for decreased power consumption. Most of these mechanisms can be implemented directly in the operating system and can be made to operate across all Active-Page applications, however, specific compiler techniques can be employed which reduce overall power consumption on a per-application basis. For example, Active Page memory devices could enable an operating system to decrease the clock frequency of inactive Active Pages with minimal effects on performance. Furthermore, for extremely low-power environments, an operating system can schedule fewer computations per clock cycle to keep under power budgets. Finally, for reduced power, application partitioning techniques can take into account power consumption and thus balance power consumption versus performance.

4.1 Variables Clock Frequencies

There are two situations where a slower clock frequency can be advantageous for low-power Active Page memory designs. The first is when an Active Page is allocated, but not computing. In this case the clock frequency of the Active Page processing unit can be decreased with minimal impact on application performance. The second is when an Active Page is computing but the application is limited by the processor.

Active Pages which are not computing generally remain in an idle loop checking a synchronization variable. A decrease in clock frequency of the Active Page functional unit can be viewed as an increase in the startup latency. Generally, the computation time is on the order of several thousand clock cycles. Thus, increasing the startup latency from one clock cycle to ten or even a hundred would not seriously affect performance. There would be considerable power savings by such ten or hundred fold decrease in clock frequency.

Active-Page applications often require a balance between processor and intelligent memory. If Active Pages are producing results too quickly for the processor to consume, then their clock frequency may be reduced without affecting application performance. Application characteristics vary widely, however, and can make developing general operating system policies difficult.

4.2 Replacement and Activation Control

At a coarser grain, the operating system can further decrease current draw by limiting the number of Active Pages computing. The operating system must already manage page replacement and schedule Active-Page computation under the constraints of the amount of physical memory available. If the number of Active Pages that fit in a single chip exceeds the power that can be dissipated, then the number that are allowed to compute simultaneously must be limited. Once again, if the application is processor-limited, then performance may not be affected. Furthermore, deactivation of the associated logic does not restrict ordinary memory I/O operations on the memory super-page.

4.3 Compiler Techniques

Future work will address partitioning entire applications. Existing technology exists for such system level partitioning in the form of exact solutions obtained by Integer Linear Programming (ILP), and approximate solutions obtained by genetic and simulated annealing algorithms. Such algorithms balance communication and synchronization costs against potential parallel execution in an attempt to find an efficient overall system partitioning. The same algorithms can also be employed to consider power consumption. Power thus becomes another variable in the minimization goal, and can be balanced by the programmer at compilation time against performance.

5 Future Work

Our previous work [OCS98] has shown that Active Pages are a promising architecture for intelligent memory with substantial performance benefits. Our current work focuses upon the power consumption of Active Page implementations. Estimates show that several design alternatives exist, but Active Page DRAMs will benefit from advances in low-cost packaging technologies. Future work will pursue both architectural and software approaches to reducing power consumption. This work will include detailed chip-level synthesized VHDL models. Furthermore, we will explore the software techniques discussed in this paper using cycle-by-cycle simulation of the processor and memory subsystem.

An increasing amount of research is being done in the area of asynchronous logic, due to the fact that clock skew and distribution are serious issues that a logic designer must face.

Clock speed has an effect on both the area and power consumed by a design. As the clock speed increases, more complex structures are needed to distribute the clock accurately throughout the chip. The increased clock speed has two disadvantages. First, the larger clock drivers consume more power. Second, overall power dissipation increases linearly with clock frequency.

Recent developments have made it possible to use asynchronous logic in more complex logic designs. The AMULET2e processor [FGT⁺97] is an asynchronous ARM[Fur96] 32-bit processor that consumes 30% of the power and takes 54% of the area of the ARM810[Fur96]. An asynchronous RISC core provides a low-power means of implementing an Active Page processing element.

Another possibility is the use of an asynchronous FPGA. Commercial FPGAs are currently targeted to synchronous logic and do not lend themselves to the implementation of asynchronous logic[HBBE94][HBBE92]. Various new FPGA designs have been developed solely for the purpose of implementing self-timed logic[HBBE94][MA98]. Work done at the University of Washington has shown that it is possible to implement real circuits in an asynchronous FPGA[BEHB]. Asynchronous FPGAs may prove to be a viable architecture for Active Page memory system processing elements.

6 Conclusion

This paper evaluates three different Active Page processing-element architectures, and explores the power requirements of each. Due to the many projections required to estimate our power consumption in future technologies, we suspect up to a 30-50% error in our power models. However, these rough calculations suggest that Active Pages will be in line with the power budgets of the commodity packaging technologies of the future [Sem97]. Furthermore, software mechanisms show strong potential for reducing peak and overall power consumption.

References

- [Alt96] Altera Corporation. *The Altera Data Book*, 1996.
- [BEHB] G. Borriello, C. Ebeling, S. Hauck, and S. Burns. The triptych fpga architecture. Univ. of Washington, Seattle WA 98195.
- [DeH96] André DeHon. *Reconfigurable Architectures for General-Purpose Computing*. PhD thesis, MIT, 1996.
- [FGT⁺97] S.B. Furber, J.D. Garside, S. Temple, J. Liu, P. Day, and N.C. Paver. Amulet2e: An asynchronous embedded controller. In *AsynC '97*, 1997.
- [Fur96] S.B. Furber. *ARM System Architecture*. Addison Wesley Longman, 1996.
- [HBBE92] S. Hauck, G. Borriello, S. Burns, and C. Ebeling. Montage: An fpga for synchronous and asynchronous circuits. In *2nd International Workshop on Field-Programmable Gate Arrays*. FPL, August 1992.
- [HBBE94] S. Hauck, S. Burns, G. Borriello, and C. Ebeling. An fpga for implementing asynchronous circuits. *IEEE Design and Test of Computers*, 11(3), Fall 1994.
- [HW97] J. Hauser and J. Wawrzyniek. Garp – a MIPS processor with a reconfigurable coprocessor. In *Symp on FPGAs for Custom Computing Machines*, Napa Valley, CA, April 1997.
- [I⁺95] K. Itoh et al. Trends in low-power RAM circuit technologies. *Proceedings of the IEEE*, 83(4), April 1995.
- [I⁺97] K. Itoh et al. Limitations and challenges of multigigabit DRAM chip design. *IEEE Journal of Solid-State Circuits*, 32(5):624–634, 1997.

- [Jon98] D. Jones. The ultimate risc. *ACM Computer Architecture News*, 16(3):48–55, 1998.
- [M⁺97] Masato Motomura et al. An embedded DRAM-FPGA chip with instantaneous logic reconfiguration. In *1997 Symposium on VLSI Circuits*, 1997.
- [MA98] K. Maheswaran and V. Akella. Pga-stc: Programmable gate array for implementing self-timed circuits. *International Journal of Electronics*, 84(3), 1998.
- [NTM⁺95] Masato Nagamatsu, H. Tago, T. Miyamori, et al. Ta 6.6: A 150mips/w cmos risc processor for pda applications. In *Proc of the International Solid-State Circuits Conference*. IEEE, 1995.
- [OCS98] Mark Oskin, Frederic T. Chong, and Timothy Sherwood. Active pages: A computation model for intelligent memory. In *Proceedings of the 25th Annual International Symposium on Computer Architecture (ISCA'98)*, Barcelona, Spain, 1998. To Appear.
- [San96] S. Santhanam. Strongarm sa110: A 160mhz 32b 0.5w cmos arm processor. *HotChips*, VIII:119–130, 1996.
- [SB95] M. R. Stan and W. P. Burleson. Bus-invert coding for low-power i/o. *Transactions on VLSI Systems*, 3(1), March 1995.
- [SB97] M. R. Stan and W. P. Burleson. Low-power encodings for global communication in cmos vlsi. *Transactions on VLSI Systems*, 5(4), December 1997.
- [Sem97] Semiconductor Industry Association. The national technology roadmap for semiconductors. <http://www.sematech.org/public/roadmap/>, 1997.
- [Tos98] Toshiba. *TOSHIBA TC59S6416 Datasheet*, January 1998.
- [WCF⁺94] S. Wuytack, F. Catthoor, F. Franssen, L. Nachtergaele, and H. D. Man. Global communication and memory optimizing transformations for low power systems. In *Proc. Int. Workshop Low Power Design*, Napa Valley, California, April 1994.