Dynamically Resizable Static CMOS Logic for Fine-Grain Leakage Reduction

Seongmoo Heo and Krste Asanović
MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, MA 02139
heomoo,krste@csail.mit.edu

Abstract

Digital circuits often have a critical path that runs through a small subset of the component subblocks, but where the path changes dynamically during operation. Dynamically resizable static CMOS (DRCMOS) logic is proposed as a fine-grain leakage reduction technique that dynamically downsizes transistors in inactive subblocks while maintaining speed in subblocks along the current critical path. A 64-entry register free list and a 64-entry pick-two arbiter are used to evaluate DRCMOS. DRCMOS is shown to give a 50% reduction in total power for equal delay in a 70 nm technology.

1 Introduction

Power has become one of the primary design constraints for modern microprocessors. Static leakage power has grown exponentially with reduced threshold voltages in deep submicron process technologies, and is now a large component of total power dissipation. Many techniques have been developed to reduce leakage power, and we can divide these into two categories [5].

The first category selects slower, lower leakage transistors on non-critical paths at design time; we refer to these as statically-selected slow transistors (SSSTs). SSST techniques include: conventional transistor sizing, lower Vdd [12], stacked gates [9], longer channels [8], higher threshold voltages [6], and thicker T_{ox}. After use of SSSTs, leakage power is often dominated by transistors in the critical paths [5].

The second category tries to reduce leakage within critical paths by placing fast transistors into a low leakage state during idle periods; we refer to these as dynamically-deactivated fast transistors (DDFTs). DDFT techniques include body biasing [7], sleep transistors [6], sleep vectors [13], and leakage biasing [5, 4]. DDFT techniques can be further categorized depending on the size of block that is deactivated. Coarse-grain DDFT techniques are common in mobile processors where the entire processor is put into a standby mode when there is no task to run. A variety of fine-grain DDFT techniques have been proposed to reduce leakage in an active processor by deactivating the idle entries, banks, or ports of critical array structures such as L1 caches and regfiles [11, 5]. For some critical logic structures such as ALUs, fine-grain power gating or leakage biasing [4] can be used to save leakage power provided the unit's activity pattern includes sufficiently long idle times to repay the often large energy cost of switching in and out of a low-leakage mode. Unfortunately, many critical logic blocks within a microprocessor are busy every cycle and so are not amenable to block-level deactivation.

Even within highly active critical blocks, however, many individual circuit paths are idle on any given cycle, though whether a path is active or inactive changes dynamically during operation. In this paper, we introduce *Dynamically Resizable CMOS* (DRCMOS) logic which exploits this phenomenon to reduce leakage. DRCMOS dynamically downsizes transistors on idle paths while maintaining speed along active critical paths.

2 Deterministic Limited Activity

The observation is that often some inputs to a large fan-in logic block remain inactive for a significant amount of time even when the block is always busy. Inactive inputs generate inactive intermediate signals and consequently inactive subblocks. Moreover, in many cases, the activity pattern can be exactly determined ahead of time based on previous input signals. We refer to this phenomenon as *Deterministic Limited Activity*. For example, many logic blocks attached to queues or arrays, which are ubiquitous in modern superscalar processors, exhibit deterministic limited activity. Figure 1 shows an example of deterministic limited activity. The top six inputs are determined to be inactive for a while. Idle subblocks are shaded. Non-shaded subblocks constitute critical paths. Only a subset of the subblocks are on the critical path and the rest remain idle for a while.

There are two key concerns when exploiting determinis-

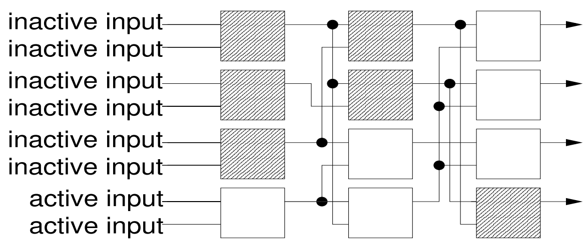

Figure 1. Deterministic limited activity.

tic limited activity to save leakage power. First, idle subblocks should maintain their output values to preserve the functionality of the entire block. Second, the critical path within a block changes dynamically, therefore soon-to-be active subblocks must be woken from a low-leakage and probably low-speed state sufficiently early to avoid a delay penalty.

3 Dynamic Resizing

Dynamic resizing (DR) is a new DDFT technique that exploits deterministic limited activity to save power by resizing each subblock dynamically according to its activity. DR determines the idle subblocks for the first stage of logic based on input patterns. For subsequent stages of logic, it is known they will be idle if all subblocks feeding their inputs will be idle. When a subblock is determined to be idle, DR downsizes the transistors in the subblock to save leakage power. To maintain speed along critical paths, DR upsizes transistors in soon-to-be active subblocks before critical paths change. Figure 2 shows an example of dynamic resizing. It is determined that the top six inputs will be inactive for subsequent cycles. The idle subblocks (shaded) are resized small to save leakage, while the active subblocks are resized large to maintain speed.

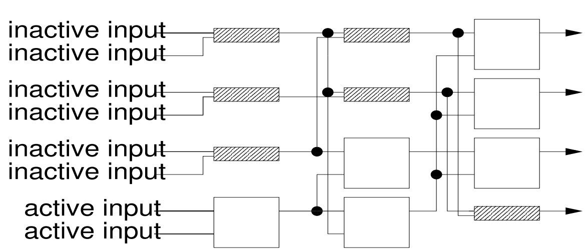

Figure 2. Dynamic resizing.

4 Dynamically Resizable Static CMOS

We propose a new static CMOS logic family, *Dynamically Resizable Static CMOS Logic (DRCMOS)*, to implement DR. Although dynamic logic has been common in the critical paths of custom microprocessors, increasing leakage currents and coupling noise are making static logic more attractive [2]. Some researchers even predict that conventional domino circuits could cease to be useful below the 70 nm generation [1].

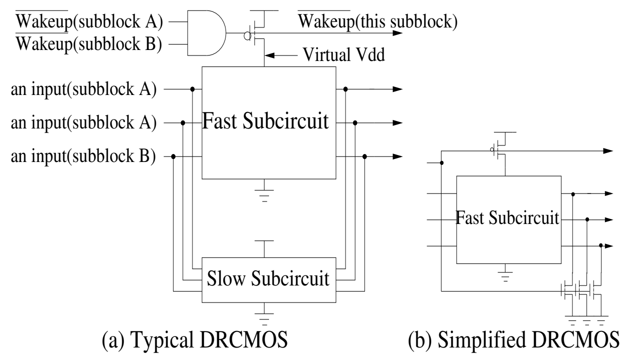

Figure 3. Dynamically resizable static CMOS logic (DRCMOS).

A DRCMOS circuit consists of a fast subcircuit and a slow subcircuit connected in parallel between inputs and outputs (Figure 3(a)). The fast subcircuit is built with large, low Vth transistors that are high speed but leaky. The slow subcircuit has the same functionality but is built using smaller, high Vth transistors to give low leakage. The slow subcircuit might also implement the logic function using more complex gates with deeper transistor stacks to reduce leakage further. When a DRCMOS circuit is active, both subcircuits are powered on and cooperate to generate output results. When the circuit is idle, the fast subcircuit is dynamically deactivated using sleep transistors to cut its subthreshold and gate leakage, while the slow subcircuit remains on to preserve output values. In effect, DRCMOS provides dynamic resizing between high-speed/high-leakage and low-speed/low-leakage modes of operation.

The slow subcircuit can be further optimized when the inactive state has a limited set of input patterns. A trivial case is where the inactive state always has a zero output in which case the slow subcircuit degenerates to a single NMOS pulldown transistor as shown in Figure 3(b).

DRCMOS requires additional control logic to generate the wakeup signals. The wakeup signal must be generated early enough (typically at least one clock cycle earlier) to ensure each subcircuit will be activated in time to propagate a critical transition at full speed. The wakeup signals for the first stage subblocks are generated from external logic associated with the inputs. Subsequent stage subblocks generate their wakeup signals by OR-ing the wakeup signals from their input subblocks.

A sneak leakage path can occur when the fast subcircuit

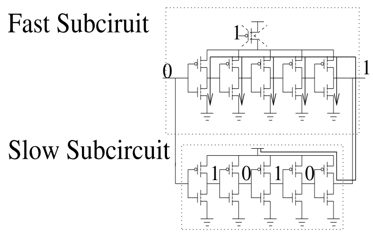

Figure 4. Sneak leakage path problem.

is deactivated as shown in Figure 4. To prevent this, the fast subcircuit must use both NMOS and PMOS sleep transistors, or its output stage must have separate power gating.

5 Evaluation Methodology

To evaluate DRCMOS logic, two key blocks of a modern superscalar processor were chosen: a static 64-entry register free list slice (Figure 5), and a static 64-entry pick-two arbiter (Figure 6).

The register free list slice is a FIFO containing a list of currently unassigned 9-bit physical register numbers (Figure 5). The FIFO is implemented as a small circular RAM with two pointers giving the head and tail of the list. A complete free list would use multiple slices to allow parallel access to multiple distinct registers. The free list uses a static mux tree to provide the read port. The read mux trees exhibit deterministic limited activity and so DR can be applied. Only one input to the mux tree, the input from the entry pointed to by the read pointer, is active, thus many muxes in the tree remain idle for multiple cycles. Also, the location of the next active input to the tree is predetermined to be circularly sequential from the current input owing to the circular FIFO structure. In the DRCMOS register free list design, only the muxes in the first stage of the mux tree where the read pointer is currently pointing or will point next cycle are upsized. In the second stage, any mux which has any upsized children is also upsized. The root mux is always on the critical path and is not resized.

The arbiter selects instructions for execution from the pool of ready instructions in the issue window. The inputs to the arbiter are request signals from ready instructions and the outputs are issue grant signals. Our arbiter selects the two oldest ready instructions (Figure 6). The issue window contains full and empty regions delimited by the read and write pointers (Figure 7). To simplify control, we treat the empty region as inactive inputs to the arbiter and the entire full area as active, even though some entries in the full area will be inactive as they are not ready for issue. The arbiter shows deterministic limited activity as the borders between the full and empty areas move sequentially as instructions are fetched and retired. When idle, the arbiter cell has zero

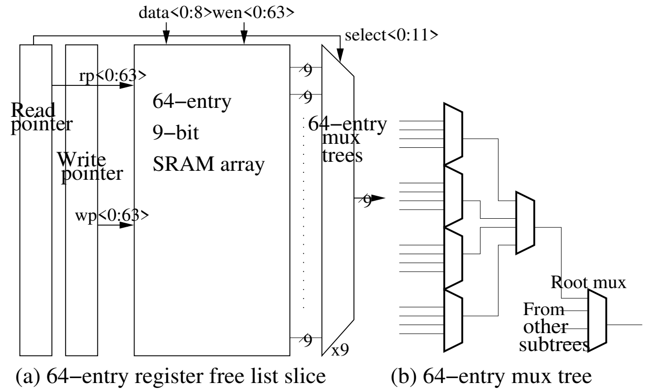

Figure 5. A static 64-entry register free list slice.

outputs, and so the DRCMOS slow subcircuit was simplified as shown in Figure 3(b).

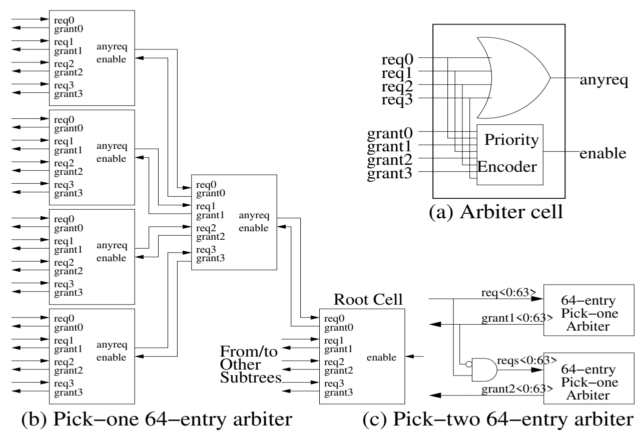

Figure 6. A static 64-entry pick-two arbiter.

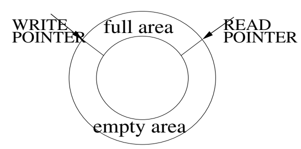

Figure 7. Logical structure of issue window.

In addition to comparing DRCMOS to baseline designs, architectural pipelining was also evaluated as another way to lower active and leakage power for the free list and arbiter. A pipelined structure has relaxed performance demands and so can use slower and lower-power transistors. However, pipelining adds clock and switching power. Also, the increased area and number of transistors can lead to more leakage power. Although pipelining lowers local cycle time, it adds global latency and so can impact the overall clocks-per-instruction performance of the processor by adding more hazards. When pipelining the register free list, timing elements are inserted between the mux select control logic (read pointer) and the mux tree, that is, select<0:11> in Figure 5(a) is pipelined. For the pipelined arbiter, timing elements are inserted between the AND gates and the second pick-one arbiter, that is, reqs<0:63> in Figure 6 is pipelined. All flip-flops had appropriate clock gating.

6 Results

The circuits were designed for a projected 70 nm process obtained from the BPTM project [3]. All simulations used HSPICE and the results for the DR scheme include the power overhead of the wakeup control logic and of switching the sleep transistors.

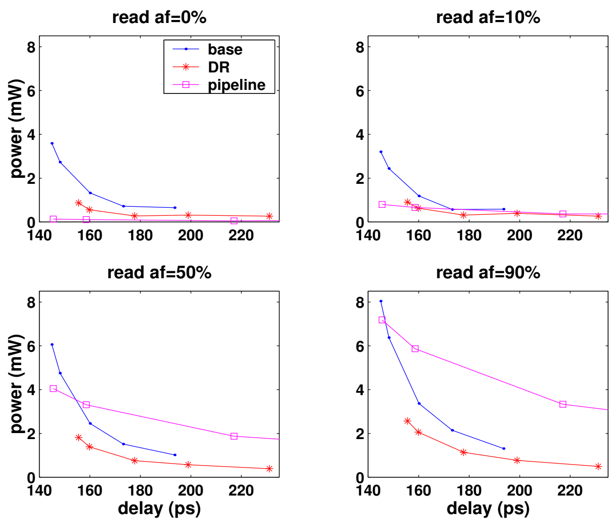

Figure 8. PD curves for register free list using supply voltage scaling.

Figure 8 shows power-delay (PD) curves for the baseline, DRCMOS, and pipelined versions of the static 64-entry register free list under supply voltage scaling. Supply voltage was varied from 0.7 V to 1.35 V. Temperature was set at 100 ^{o}C. The SRAM array was assumed to contain 50% zero bits. The read activity factor is the rate at which entries are read and was varied from 0% to 90%. The graph clearly shows that DR gives the best PD curve except for the 0% read activity factor case. Even when the read activity factor is high, DR upsizes only a small subset of the muxes keeping others small. At 90% read activity, DRCMOS gives around 10% delay reduction for equal power or 1.5× total power reduction at equal delay. The pipelined version suffers from flip-flop switching power overhead and only performs well at low activity factors where the advantage of small low-leakage transistors becomes apparent.

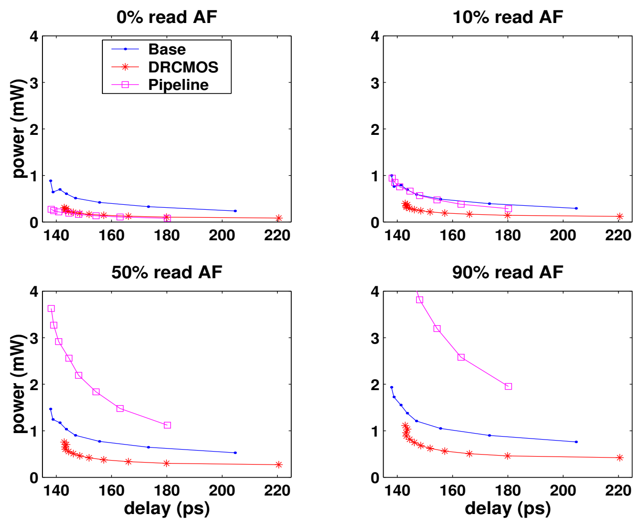

Figure 9. PD curves for register free list using transistor sizing.

Figure 9 shows alternate PD curves of the baseline, DRCMOS, and pipelined free lists when conventional design-time transistor sizing is used to vary delay with supply voltage fixed at 0.9 V. The graph clearly shows that DRCMOS gives the best PD curve. At 90% activity factor, DRCMOS gives at least 10% delay reduction for equal power or around 50% total power reduction at equal delay.

Figure 10 shows power-delay curves of the alternate designs of the static 64-entry pick-two arbiter using supply voltage scaling, and varying the number of the entries in full area from 0 to 32. DR performs better when the instruction window is emptier as more arbiter cells remain small reducing total leakage power. Typical instruction window occupancies vary greatly during program execution depending on application program characteristics. With only 6 entries in the full area, around 10% delay reduction for equal power or 2× total power reduction for equal delay can be achieved with DR. On the other hand, when half the entries in the issue window are ready, the DRCMOS curve is close to the baseline at shorter delays. The pipelined version supports lower delays at low power, but at looser delay constraints, the flip-flop power overhead overwhelms the power saving from the use of small and high Vth transistors.

Figure 11 shows PD curves of the arbiter with transistor sizing at a 0.9 V supply. When there are 16 ready entries, around 3% delay reduction for equal power or 50% total power reduction for equal delay can be achieved by using DRCMOS logic, compared to the baseline. Pipelining always gives a better PD curve than baseline as transistors

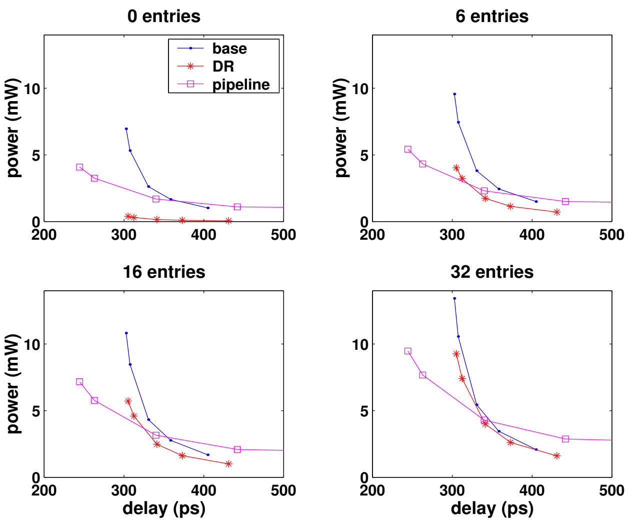

Figure 10. PD curves for register free list using transistor sizing.

can be downsized to take advantage of the pipelining.

Although pipelining works well for the arbiter, it introduces an architectural hazard that prevents dependent instructions from issuing in consecutive cycles, which causes a significant global processor performance penalty [10] and so would usually be avoided in a high performance design.

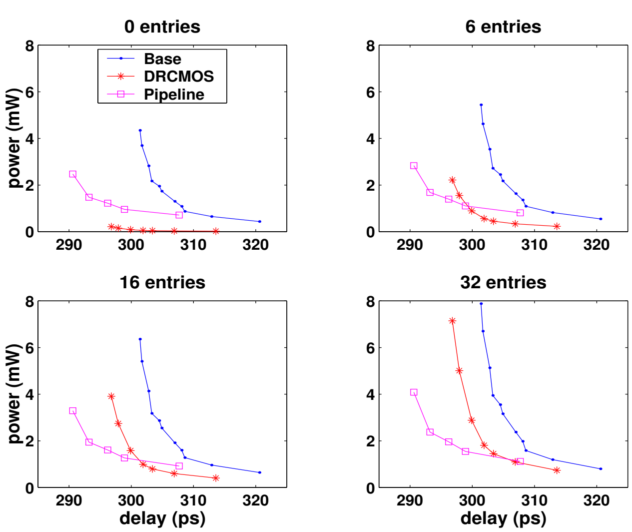

Figure 11. PD curves for register free list using transistor sizing.

7 Conclusion

DRCMOS reduces leakage power of critical path transistors in active microprocessors. DRCMOS exploits the regular predictable patterns of activity within microarchitectural blocks to downsize transistors that are known to be off the critical path in the next cycle. DRCMOS can be used at a very fine-grain within blocks that are active every cycle, but where many subblocks will be idle. Dynamically resizable CMOS is shown to reduce power consumption by up to 50% at equal delay in critical components of a modern superscalar processor implemented in a 70 nm technology.

References

[1] M. Anders et al. Robustness of sub-70nm dynamic circuits: analytical techniques and scaling trends. In *Symp. on VLSI Circuits*, pages 23–24, 2001.

[2] H. Ando et al. A 1.3-ghz fifth-generation SPARC64 microprocessor. *IEEE JSSC*, 38(11):1896–1905, November 2003.

[3] Device Group at UC Berkeley. Predictive technology model. Technical report, PTM, 2001.

[4] S. Heo and K. Asanovic. Leakage-biased domino circuits for dynamic fine-grain leakage reduction. In *Symp. on VLSI Circuits*, pages 316–319, 2002.

[5] S. Heo et al. Dynamic fine-grain leakage reduction using leakage-biased bitlines. In *ISCA*, pages 137–147, 2002.

[6] J. T. Kao and A. P. Chandrakasan. Dual-threshold voltage techniques for low-power digital circuits. *IEEE JSSC*, 35(7):1009–1018, July 2000.

[7] H. Makino et al. An auto-backgate-controlled MT-CMOS circuit. In *Symp. on VLSI Circuits*, pages 42–43, 1998.

[8] J. Montanaro et al. A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor. *IEEE JSSC*, 31(11):1703–1714, November 1996.

[9] S. Narendra et al. Scaling of stack effect and its application for leakage reduction. In *ISLPED*, pages 195–200, August 2001.

[10] S. Palacharla et al. Complexity-effective superscalar processors. In *ISCA*, pages 206–218, 1997.

[11] M. Powell et al. Gated Vdd: A circuit technique to reduce leakage in deep-submicron cache memories. In *ISLPED*, 2000.

[12] M. Takahasi et al. A 60-mW MPEG4 video codec using clustered voltage scaling with variable supply-voltage scheme. *IEEE JSSC*, 33(11):1772–1778, November 1998.

[13] Y. Ye, S. Borkar, and V. De. A technique for standby leakage reduction in high-performance circuits. In *Symp. on VLSI Circuits*, pages 40–41, 1998.