The Design of a High Performance Low Power Microprocessor

Dan Dobberpuhl
Digital Semiconductor
Palo Alto, CA 94301

Introduction

The market for "Anywhere-Anytime" computing is developing rapidly as early generations of PDA's, intelligent mobile phones, and other smart handheld devices have become available to consumers. The availability of high bandwidth connections to homes is also expected to fuel the demand for interactive digital media products such as interactive TV, networked multi-player 3D video games, and inexpensive but powerful internet browsers. These applications are notable for their high performance computing requirements within a tightly constrained price and energy/power budget. Energy is obviously a factor in the hand-held arena, but power is also constrained by low-cost packaging considerations in many tethered applications as well.

Traditional embedded processors which meet the price and power objectives cannot deliver the performance required by these new applications. At the other end of the processor spectrum, incredible performance improvements have been made, but at power and price levels that are incompatible with imbedded applications. What is called for is a new class of microprocessor products which enable high performance applications within the cost and power constraints of the embedded market - the <$50, <1W supercomputer-on-chip.

In 1992, the first Alpha chip delivered the raw performance of a Cray-1 in a single device dissipating about 25W [1]. The latest generation of Alpha chips delivers more than four times that performance level at about the same power [2]. The design and fabrication technology which has made this possible, when applied within the constraints of the new embedded market, can deliver Cray-1 class performance to battery-powered and low cost tethered applications.

StrongARM 110

- **Function**
  - **Implements ARM Version 4 instruction set**
  - **Bus compatible with ARM 610, 710 and 810**
- **Performance**
  - **160MHz @ 1.5v -> 183 Dhrystone MIPS at < 0.5W**
  - **215MHz @ 2.0v -> 245 Dhrystone MIPS at < 1.0W**
  - **3.3v pin bus**
- **Process and Package Technology**
  - **2.5 million transistors fabricated in 0.35μm 3 metal CMOS with 0.35v V_T and 0.25μm L_{EFF}**
  - **Die size: 7.8mm x 6.4mm -> 50mm^2**
  - **144 pin plastic TQFP**

Table 1. Device Characteristics

The StrongARM 110^{TM} is the first example of this new generation of very high performance embedded processors [3]. Developed by UK-based ARM Ltd. approximately ten years ago, the ARM microprocessor architecture is exclusively focused on low-cost and low-power applications. It is used extensively in consumer-oriented applications such as mobile phones, PDA's, organizers, and video games .

ISLPED 1996 Monterey CA USA
0-7803-3571-8/96/$5.00©1996

Overall Approach to Reduce Operating Power

Circuit and Logic Considerations

Lower Vdd

The starting point for Vdd was defined by the 0.35µ CMOS-6 technology to be a maximum of 2.0v. This choice had already been made by considerations related to Alpha microprocessors [2] before the StrongARM design was begun. Because of the very high clock rates which are featured in the Alpha designs, it was determined that the optimum trade-off for speed vs power in 0.35µ technology could be achieved by setting a relatively low operating voltage and then maximally scaling the devices. For example, gate oxide thickness for this technology is nominally 60 Angstroms, and Vtn, |Vtp| = 0.35v.

Analysis of clock speed versus operating voltage for this design showed a loss of about 25% would result from a Vdd reduction from 2.0v to 1.5v with a corresponding decrease in power of 44% at a given frequency. Below 1.5v, clock speed falls rapidly as 1/(Vdd-Vt) begins to dominate, especially as Vt and Vdd tolerances are taken into account. Therefore our initial plan was to choose a value for Vdd within the range 1.5v < Vdd < 2.0v based on further analysis of power dissipation.

Edge-Triggered Latches / Conditional Clocking

Our previous designs of Alpha microprocessors had shown us that for high performance devices, clocking can easily dominate the dynamic power budget. For the 21064 [1], clock power represented about 65% of total power dissipation. We also knew that efficient clocking is one of the keys to high performance. We considered various approaches to reducing clock power and settled on one which we considered to be relatively low risk with reasonable speed efficiency. By changing from a flow-through latch scheme to a unique edge-triggered configuration, we estimated that the clock load could be reduced by half compared to Alpha. Another benefit, whose value was hard to estimate, was the lack of spurious transitions on the latch outputs compared to flow-through. The delay performance of this latch is quite good and comparable to the two flow-through latches it replaces. However, with this scheme we gave up the opportunity to slosh the delay budget across latch boundaries.

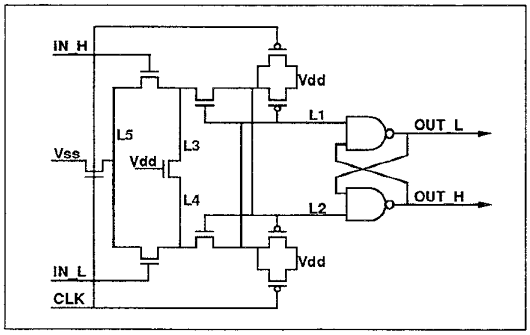

Figure 1. Edge-Triggered Latch

The latches used in the majority of the design are differential edge-triggered latches (figure 1). The circuit structure is a precharged differential sense amp followed by a pair of cross-coupled NAND gates. The sense amp need not be particularly well balanced because the inputs to the latch are full CMOS levels. The NMOS shorting device between nodes L3 and L4 provides a DC path to ground for nodes L1 and L2 in case the inputs to the latch switch after the latch evaluates. This device is required for the latch to be static. In addition, at least one of the two data inputs to the latch must be high during the full phase that the clock to the latch is high, not just the period while the latch is loading. One of the disadvantages of this latch is the fact that the internal nodes precharge and discharge each cycle that the clock is active, independent of whether the data in the latch changes.

Conditional clocking was also included in the methodology. In the Alpha designs, the chips were dynamic, so conditional clocking implied a significant penalty. In StrongARM, the applications require fully static operation, allowing the clock to be stopped indefinitely. Given that requirement, conditional clocking was thought to have minimal cost with considerable but unquantified power savings potential.

Conditional clock buffers are simple NAND/invert structures with an integral latch on the condition. The buffers must be matched to their load to minimize skew. Since adding dummy clock loads is contrary to the low-power design philosophy, we created scaled clock buffers which would produce matched clocks for a wide range of loads and only needed to add dummy clock loads for a small number of very lightly loaded clock nodes. The task of matching the clock buffers to the load was greatly simplified by the fact the clock load presented by our standard latches is largely data-independent. While the use of conditional clock buffers is central to the design method used on StrongARM, it should be noted that the critical paths to generate the condition input to these buffers represent some of the most difficult design problems in the chip. In this case, we decided that the power saving associated with the conditional clocking was worth the additional design effort and possible performance reduction.

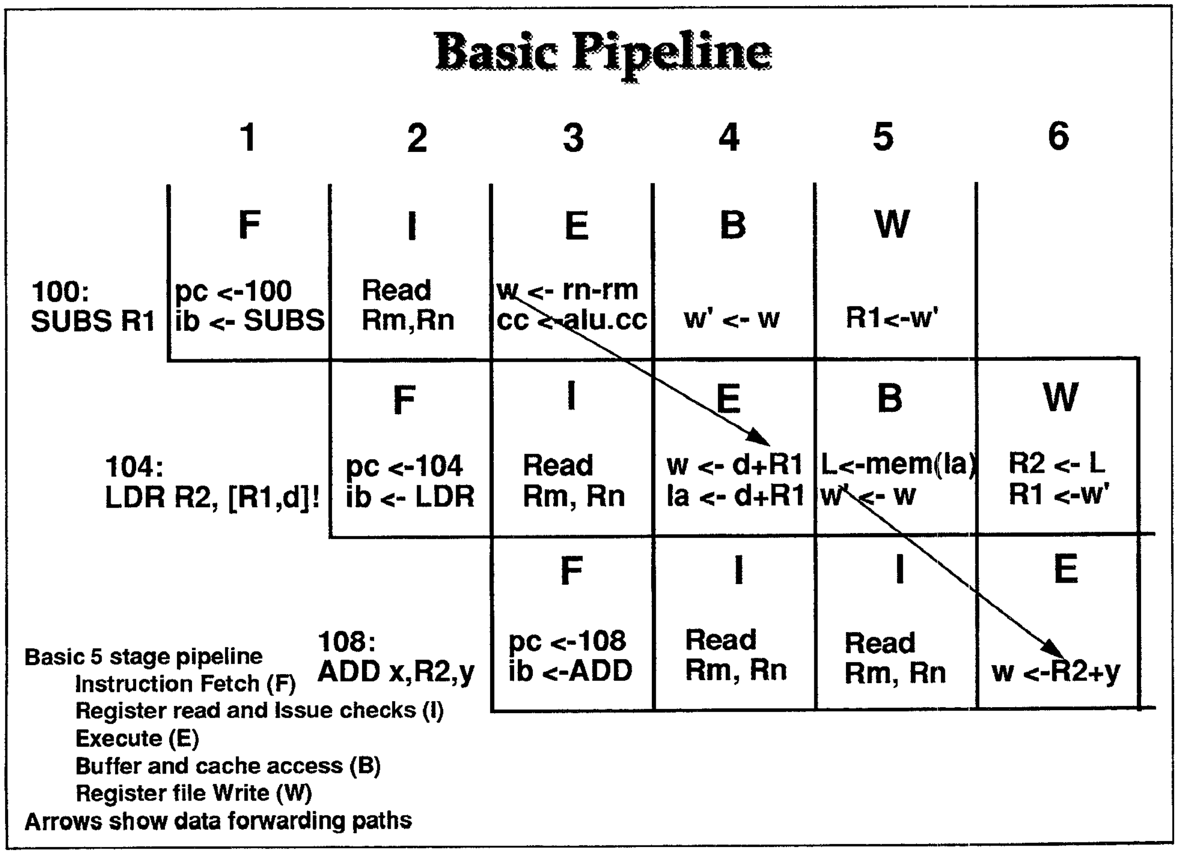

Figure 2. Pipeline Diagram

Micro-architectural Considerations

Long-tick Model

Digital's recent microprocessors [1,2,4] utilize short tick micro-architectures, e.g., relatively few gate delays per cycle. Alpha designs have typically been in the range of 15-18 including latch delays. For StrongARM we ended up with relatively long pipe stages - approximately 25 gate delays. A deeper pipeline would have given faster cycle times but we would have lost some low level implementation optimizations and not met our power goals.

At the other extreme, we opted not to include multi-issue. A superscalar design would have increased control complexity and therefore increased power and design time as well as required higher per cycle memory interface performance. Even without multi-issue, it is possible to saturate the available memory bandwidth that can be provided with the 32b 66MHz bus interface. The chip utilizes a simple five stage pipeline as shown in Figure 2.

Clock Switching

To provide compatibility with previous ARM designs, the chip can be operated from two separate clocks for I/O and internal logic. As shown in Figure 3, an internal clock, called DCLK, is usually generated by an on-chip PLL [5] and runs nominally between 88 and 287MHz. The second clock is a bus clock, known as MCLK which operates up to 66mhz.

It can be supplied by an external source or it can be supplied by the chip based on a division of the PLL clock signal. In normal operation, the bus interface unit or BIU and part of the write buffer are clocked by MCLK. The internal sections of the chip are clocked by DCLK. Both clocks are distributed by buffered H-trees to conditional clock buffers in the various sections of the chip.

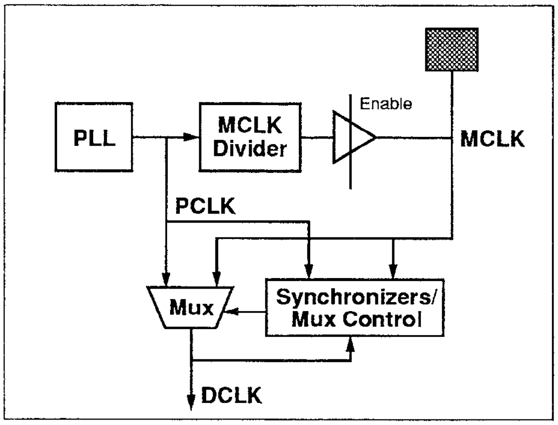

Figure 3. Clock Generation

For chip output operations, data is passed through the Write Buffer which provides the necessary clock synchronization easily since the function is basically that of a FIFO. On the other hand, reading data into the chip, as in a cache miss, is a demand situation where the latency must be minimized. After considering several schemes, we decided that the optimum solution for handling this situation would be to clock the entire chip synchronously from the external clock source during all processor read events. Because the cache is blocking, it is unlikely that much forward progress could be made while waiting for a cache fill. Thus, the performance penalty for clocking the core at the slower bus clock rate during fills is minimal. However, power dissipation is reduced considerably because the core is being clocked at the slower rate (typically 1/3 factor) of the bus clock. The net result is that for many application environments (a function of the external bus loading and the clock frequency ratios) the StrongARM chip actually utilizes less power when the chip is encountering a high cache miss rate.

Circuit/Logic Techniques Considered but Rejected

Several additional circuit techniques were evaluated for possible use in StrongARM including charge recycling and dual edge-triggered latching. Charge recycling in the output buffers, did appear to offer some potential gains. However, I/O power turned out to not be much of an issue for this design due to the unique clocking factor explained above, so it was not utilized.

Dual edge-triggered latches (DET) were considered and rejected because of the loss of intermediate timing resolution for synchronous circuitry like precharges. Also, with the special latch described above, the number of clocked transistors is only 3. Standard DET's require 12 clocked transistors for static implementations, resulting in little potential for power savings despite the halved clock frequency.

Power Comparison to Alpha

It is interesting to compare the power dissipation of StrongARM to an Alpha CPU to see how the savings is achieved. As shown in Table 2, we relate the power of the Alpha 21064 [1] to StrongARM as a function of the first order differences.

Start with Alpha 21064: 200MHz @ 3.45v Power= 26W

Vdd reduction:	**Power reduction = 5.3x**	**-> 4.9W**
Reduce functions:	**Power reduction = 3x**	**-> 1.6W**
Scale process:	**Power reduction = 2x**	**-> 0.8W**
Clock load:	**Power reduction = 1.3x**	**-> 0.6W**
Clock rate:	**Power reduction = 1.25x**	**-> 0.5W**

Table 2. Alpha --> StrongARM Power Dissipation

Starting with a 200MHz 21064 in 0.75μ technology, factoring in Vdd, functionality, process scaling, clock loading, and clock frequency, we end up with a power dissipation close to the realized value of 450mW.

Device Leakage Issues

Leakage vs channel length

As noted earlier, a low voltage process is key to the design of a microprocessor which will run at 160MHz while burning only 500mW. However, the same low device thresholds which allow the reduction of Vdd also result in significant device leakage. While this leakage is not large enough to cause a problem for normal operation, it does pose problems for standby current, especially if the process skews toward short channel devices.

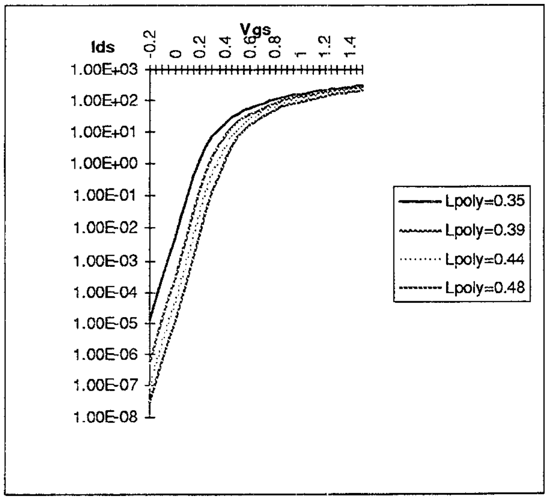

Figure 4. Leakage (uA/u) vs Vgs as a Function of Drawn Channel Length.

Our initial analysis indicated that the StrongARM chip might burn as much as 80mW in Idle mode with the clocks stopped. To reduce this leakage, devices in the cache arrays, the pad drivers, and certain other areas were lengthened by 0.045μ or 0.09μ.This brought the leakage power to below the 20mW spec in the fastest process corner.

Vddi Power Down

The standby power requirement in Sleep mode is about three orders of magnitude lower than the Idle power. To meet the power limit in Sleep, we considered a variety of options including integrated power supply switches and substrate biasing schemes before choosing the simple approach of turning off the external supply. This approach is reasonable for this design since it likely to have a dedicated 1.5v supply. As more parts of the system shift to the low voltage supply, this may not be acceptable. The conflicting requirements of high performance at low voltage and low standby current promise to create interesting challenges in future designs.

Acknowledgments

StrongARM 110™ was designed by an exceptionally dedicated and talented group of engineers: Jim Montanaro, Rich Witek, Krishna Anne, Andy Black, Beth Cooper, Paul Donahue, Jim Eno, Alex Farell, Greg Hoeppner, David Kruckemyer, Tom Lee, Peter Lin, Liam Madden, Dan Murray, Mark Pearce, Sri Santhanam, Kathy Snyder, Ray Stephany, and Steve Thierauf.

References

[1] Dobberpuhl, D., et. al., "A 200MHz 64b Dual-Issue CMOS Microprocessor," IEEE Journal of Solid State Circuits, vol. 27, no. 11, Nov., 1992.

[2] Gronowski, P., et. al., "A 433MHz 64b Quad-Issue CMOS RISC Microprocessor," ISSCC Digest of Technical Papers, pp. 222-223, Feb., 1996.

[3] Montanaro, J., et. al., "A 160MHz 32b 0.5W CMOS RISC Microprocessor," ISSCC Digest of Technical Papers, pp. 214-215, Feb., 1996.

[4] Bowhill, W., et. al., "A 300MHz 64b Quad-Issue CMOS RISC Microprocessor," ISSCC Digest of Technical Papers, pp. 182-183, Feb., 1995.

[5] von Kaenel, V., et. al., "A 320MHz, 1.5mW CMOS PLL for Microprocessor Clock Generation," ISSCC Digest of Technical Papers, pp. 132-133, Feb., 1996.