

HIGH-PERFORMANCE ALPHA MICROPROCESSOR DESIGN

Daniel W. Bailey

Compaq Computer Corporation
334 South Street, SHR3-1/S30
Shrewsbury, MA 01545 USA

Abstract

Elements of successful high-performance microprocessor design are discussed. Subjects presented include clock distribution, latches, and design methods and techniques. General design trends are mentioned where appropriate. Tradeoffs between circuit performance and electrical integrity are emphasized throughout.

Introduction

Alpha microprocessors have achieved industry-leading performance since their inception in 1992 [1-4]. This paper will discuss several of the performance techniques and tradeoffs used in these designs, as well as some expected design trends in the microprocessor industry.

The major topics discussed in this paper are clocks, latches, and design methods and techniques that have been successfully used on Alpha microprocessors. Clocks are a separate topic because they systemically affect performance, and because clock designs generally do not scale with the technology. This is in contrast to, for example, a one-cycle adder. Clock distribution becomes more difficult as devices and clock periods shrink, but die sizes do not.

Latches are an important implementation issue because they are a prevalent circuit. The choice of latch, therefore, has great influence on many areas of design methodology including performance, power consumption, clock load, and timing verification strategy. In this paper, latch design for each generation of Alpha microprocessor is shown along with an explanation of the design tradeoffs for each.

High-performance design methods and techniques are broad topics and so only a few examples can be covered here. A brief comparison of complementary static and dynamic circuits is given, with the discussion focusing on dynamic circuit hazards and how to mitigate them. Full custom design is also discussed because it is a methodology that has been instrumental in achieving Alpha microprocessor performance objectives. Finally, the design technique of getting two cache read accesses per cycle using a 6T RAM cell, or double pumping, is discussed because it doubles the bandwidth of an array.

Clocks

A. Clock Grids

The classical approach to microprocessor clock design is to first identify all clock loads, then design a network of many clocks using balanced H trees such that all clocks are driven simultaneously [5]. For a large microprocessor, however, there are variations in the clock driver network caused by process, supply voltage, temperature, and coupling variations. In practice, zero overall skew is not attained using this

method and the actual skew is difficult to characterize.

In contrast, Alpha microprocessors have used an interconnect grid that defines a single major clock node. Typically this is accomplished by a balanced H tree of increasingly larger driver stages, the final stage of which drives the grid in parallel [1]. Using a grid acknowledges that zero skew will not be attained between the clock loads that tap off the grid, but because driver variations are integrated, this skew is reliable, predictable, and tolerant of non-uniform gate load. Another advantage is that distribution network and grid design proceeds concurrent with the rest of the microprocessor circuit design, not afterwards. It is for these reasons that perhaps the most important contribution to successful high performance clock designs in Alpha microprocessors has been the decision to use major clock grids.

B. Clock Driver Placement

Fig. 1 shows final clock driver placement for the first three generations of Alpha microprocessors. For the Alpha 21064, a single bank contains all final drivers and the multi-staged H tree of pre-drivers. On the Alpha 21164, four stages of final drivers and pre-drivers are each placed in the two locations shown in Fig. 1(b). All earlier pre-drivers are in a bank (not shown) midway between the two (shown) banks. Dual final driver banks reduce the maximum distance to any clock load, and so reduce the grid skew. Although splitting the final drivers mean driver variation must be modeled and managed, it also helps diminish voltage-drop and thermal problems.

On the Alpha 21264, final global clock drivers are placed in a window pane arrangement. Not shown are additional regional clock grids driven two inverters past the global clock, plus thousands of separate buffered local clocks that tap off these grids [6]. This has the effect of dispersing clock drivers even more broadly than is shown in Fig. 1(c), and also dispersing the clock edges temporally.

Thus, clock driver placement has gotten more widely dispersed with each generation. This is likely to be a continuing trend, since the advantages are numerous and scale with the technology. With widely dispersed drivers,

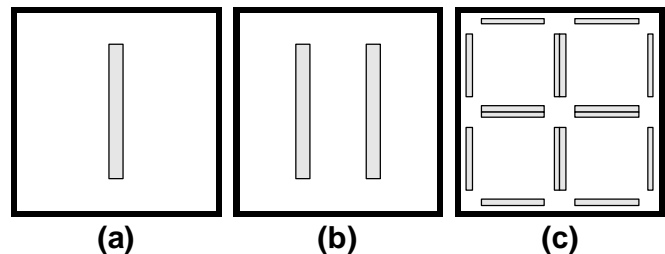


Fig. 1. Final clock driver placement for the Alpha (a) 21064, (b) 21164, and (c) 21264.

inductance on the clock node is reduced because clock drivers make for a low impedance connection to the power rails. Power and heat dissipation due to clock drivers are more uniform, and V_{DD} variation is less. Lower skew is generally easier to attain because the maximum distance of any load to the nearest driver is reduced. The primary disadvantage of widely dispersed drivers is variation caused by process and environment, but a grid ameliorates this by averaging delay variations between the clock drivers.

C. Clock Domains and Active Skew Management

Multiple clock domains are also an emerging trend [7,8]. In this case, different clocks have different regions of responsibility which nominally do not overlap. Dovetailing with this method is the design technique of active skew management of the different clock domains, either by DLLs or PLLs. Active skew management attempts to eliminate variation effects, but the downside is that new dynamic timing variations are introduced with DLLs and PLLs, including jitter and offset latencies. These timing variations, however, can be managed with careful circuit techniques [9,10].

Fig. 2 shows both of these approaches in a block diagram of the next-generation Alpha microprocessor. An external system clock provides reference to an on-chip PLL. This PLL provides the incipient clock that eventually drives the clock grid for the CPU core. This final clock node, GCLK, provides the feedback clock for the PLL as well as the reference for the other three clock domains: NCLK for the memory controller, L2L_CLK for the left L2 cache, and L2R_CLK for the right L2 cache. These three satellite clock domains are synchronized by DLLs.

The PLL output is a multiple of the SYSCLK frequency. By inputting a low-frequency clock source, package signal attenuation issues are eliminated. Since frequency synthesis is unneeded beyond the PLL, more stable and simpler DLLs are used to perform frequency synchronization for the other clock domains [10].

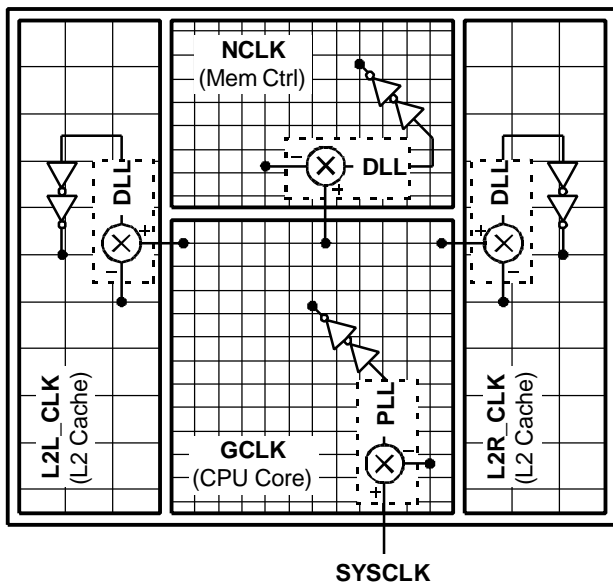


Fig. 2. Clock domains and active skew management on the next-generation Alpha microprocessor.

Latches

A. Alpha 21064 Latches

Pseudo-inverters as shown in Fig. 3 are the primary family of latches on the first-generation Alpha microprocessor [1]. It is a level-sensitive latch that is transparent when clock, CLK, is high. When CLK goes low, the second pseudo-inverter is tri-stated and data is capacitively held on node N2. As long as CLK falls quickly, the latch is race-free by construction. It also poses relatively light clock load because it is NFET dominant, and can easily accommodate functionality in the first stage. It is, however, subject to loss of state on N1 and N2 due to sub-threshold leakage, and therefore requires a minimum cycle time, which is calculated to be less than 10% of the nominal operating frequency. It is also susceptible to data corruption from coupling events on N1 and N2. Another aspect of this design is that it has three gate delays, and two latches are needed per cycle path.

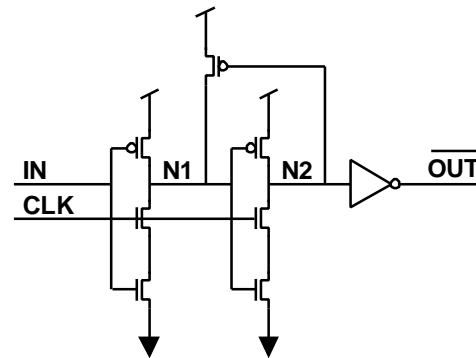


Fig. 3. Latch family of the Alpha 21064.

B. Alpha 21164 Latches

The Alpha 21164 uses the dynamic CMOS transmission gate latch family shown in Fig. 4 [2]. This is also a level-sensitive latch that is transparent when CLK is high. When CLK goes low, data is capacitively stored on node N2 and blocked by the pass gate. Because it is dynamic, this latch also has a minimum operating frequency. To incorporate logic, the first inverter can simply be converted into a logic gate. These features are similar to the Alpha 21064 latches.

Compared to the pseudo-inverter latch, however, the transmission gate latch has several advantages. First, latency is smaller at two logic gates plus a pass gate. A second important performance advantage is that the fan-in is smaller because CLK is not in the input stack. With the adoption of this latch, however, comes an increased susceptibility to race through because of the inverter gated by CLK. To counter this, the design methodology and CAD tool timing verification on the Alpha 21164 required at least one gate between driver and receiver latches.

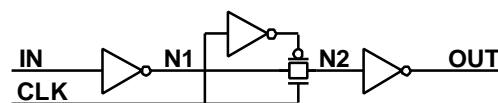


Fig. 4. Latch family of the Alpha 21164.

C. Alpha 21264 Latches

The edge-triggered latch (or flip flop) used on the third generation Alpha microprocessor is shown in Fig. 5 [3]. This CMOS differential sense amplifier latch is static for pre-charged inputs, or can be made static by the addition of weak pull-down devices within the latch. When CLK is low, all internal nodes are pre-charged. As CLK rises, the IN data is sampled and the NFET stack gated by the higher input voltage discharges. The cross-coupled NFETs and PFETs lock the latch into the evaluated state until clock goes low and the nodes gets pre-charged again.

There are several noteworthy features of this type of latch. Similar to the other latch families, the differential sense amplifier latch can include embedded logic in form of NFETs in the input stacks. The hold time for this type of flip flop can be reduced if the common source transistor is enlarged at the expense of clock load. Since only three transistors are gated by clock, on the other hand, it can also be adapted to low power design [11]. Because it is edge-triggered, it only requires one latch per clock period. The disadvantage to using flip flops is that design flexibility is lost in the adjustment of critical paths. On the Alpha 21264, this loss is compensated by the ability to use local clocks to solve timing problems.

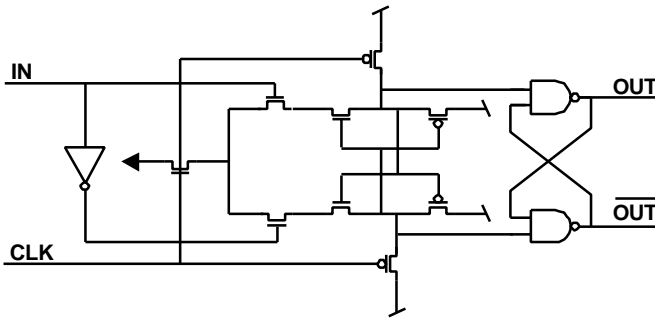


Fig. 5. Latch family of the Alpha 21264.

D. Latch Trends

Latch trends are difficult to identify because many latch families are appropriate for high-performance design. For instance, double-edge-triggered flip flops are promising because data is latched on both clock edges [12]. Pulsed flip flops have many virtues and have been adopted in several recent microprocessor designs [13,14]. The differential sense amplifier latch family shown in Fig. 5 is also obviously well-suited for high-performance design and has been chosen for the next-generation Alpha microprocessor.

Design Methods and Techniques

A. Complementary Static Logic vs Dynamic Logic

An important design method decision is where and when to use complementary static and dynamic logic. The advantages to complementary static logic are numerous and well known: it is robust, compliant to CAD tool analysis, tractable for synthesis, and can be low power. Alpha designs use this style extensively in control schematics and non-critical performance circuitry.

Dynamic circuits, on the other hand, are more susceptible to data corruption due to noise, charge sharing, sub-threshold

leakage, charge injection, and soft error upset. Dynamic circuits, however, are also faster. On Alpha designs, dynamic circuits are used where performance is critical and the extra verification is justified, such as in data paths [4].

Fig. 6(a) shows a complementary static implementation of $\overline{X+Y+Z_1Z_2}$ and Fig. 6(b) shows a counterpart version using pseudo-dynamic domino logic that outputs $X+Y+Z_1Z_2$. The dynamic circuit inputs have lower fan-in load because they do not drive PFETs too, which can be a huge advantage for a many-term OR gate. The dynamic circuit has a lower switching point at V_{TN} , which is a performance advantage, but a noise margin disadvantage. These two attributes are the primary reasons dynamic circuits are generally faster.

For high-performance dynamic circuit design, therefore, maintaining electrical integrity is the issue. When in the pre-charged high state or the evaluated low state, DYN in Fig. 6(b) is static. When PRECH is low, the latch is in a pseudo-dynamic state and is vulnerable. Glitches on the input nodes X, Y, or Z_1 and Z_2 can be catastrophic because the original data is not recovered as for a complementary static gate. Another functional issue is coupling to DYN that can cause a long lasting glitch on OUT. A partial solution is to install a weak half-latch (i.e., the PFET feedback restoring device shown in Fig. 6(b)) which also guards against sub-threshold leakage. Another fix is to set limits on the maximum allowable coupling and the ranges for the input and output beta ratios to ensure acceptable noise margin. Shielding and isolating dynamic nodes also helps. Adding fixed capacitance to DYN helps increase immunity to coupling, charge injection, and soft error upset, although too much capacitance can degrade performance.

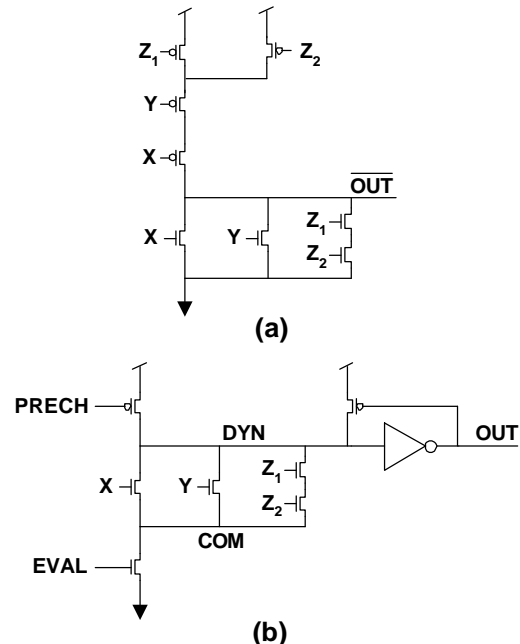


Fig. 6. Examples of (a) static logic, and (b) pseudo-dynamic (domino) logic.

B. Full Custom Design

As a methodology, full custom design has been key to achieving high performance on Alpha microprocessors [1-4]. Thorough, manual, transistor-level design and layout have

enabled critical circuits to be optimized for both performance and area. Full custom design, however, is not as time-consuming and difficult as the name implies.

First, full custom in practice does not generally mean starting from scratch. As much as possible is shrunk from previous designs. Second, although Alpha microprocessors are not built from standard cell macros, complementary static gates and the most common latches are built from a layout library. Generally, though, full custom is labor intensive and Alpha microprocessor designs have relied on the performance achievable by manual design and layout at the transistor level.

C. Double Pumping

Double pumping is a specific design technique that uses the design methods of dynamic circuits and full custom. It is used on the data cache of the Alpha 21264 to obtain two bit line references per cycle using a conventional single-ported 6T RAM cell. The crux of the method is aggressive word line operation coupled with a static PFET bit line pre-charging so that pre-charging and establishing differential are accomplished in a single phase [3].

On the Alpha 21264, a two-way set associative, double-pumped, 64kB data cache supports two unrestricted quadword (8B each) loads per cycle, or one octaword (16B) store or fill per cycle. The alternatives to double pumping are using a dual-ported SRAM cell, or using two independent, single-ported cache arrays [2]. Both these approaches consume much more area than double pumping, and dual-ported cells have the additional complexity and latency associated with bank conflicts.

Fig. 7 shows the SPICE simulation using this method for back-to-back loads on the same bit lines for separate SRAM cells. In the first phase of Stage 6, the indices are sensed when the RD_IDX clock fires, and the address decodes to enable the Port-0 word line, WL0. Bit line differential evolves and S0 asserts early in the next phase firing the bank sense amplifiers. With the Port-0 data sampled, WL0 falls and the Port-1 word line WL1 asserts, inverting the bit line differential with the aid of the static PFET pre-charge. In Stage 7, S1 rises sampling the Port-1 data. Port-0 data is driven onto the appropriate global bit lines with Port-1 data following a phase later. When the Port-1 access is complete and the Port-0 data has been pipelined an additional phase, both results are simultaneously sensed and driven to the execution units.

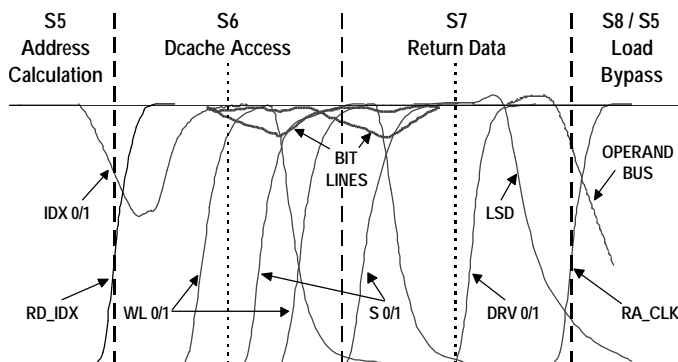


Fig. 7. SPICE simulation of double-pumped data cache accesses.

Double pumping increases the read bandwidth of the data cache by a factor of two with a minimal increase in area. This technique is also applicable to other pre-charged circuits and arrays, not just the data cache.

Conclusions

There are many ways to successfully design a high-performance CMOS microprocessor. Work begins with the prudent design of critical, broad-based circuits such as a clock distribution network and latches. An aggressive design methodology and a set of design techniques are also needed early in the design process. Most important perhaps is a willingness to trade risk for performance, and then a willingness to work to reduce the risk as much as possible.

Acknowledgements

The methods shown here have been presented on behalf of the Alpha Development Group. Special thanks for help in the preparation of this manuscript to Paul Gronowski, Vincent von Kaenel, Mike Gowan, Bill Bowhill, and Anil Jain.

References

- [1] D. Dobberpuhl *et al.*, "A 200 MHz 64 b dual issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 27, no. 11, pp. 1555-1567, Nov. 1992.
- [2] B. Benschneider *et al.*, "A 300-MHz 64-b dual-issue CMOS microprocessor," *IEEE J. Solid-State Circuits*, vol. 30, no. 11, pp. 1203-1214, Nov. 1995.
- [3] B. A. Gieseke *et al.*, "A 600MHz superscalar RISC microprocessor with out-of-order execution," in *ISSCC 1997 Dig. Tech. Paper*, Feb. 1997, pp. 176-177.
- [4] P. E. Gronowski *et al.*, "High-performance microprocessor design," *IEEE J. Solid-State Circuits*, vol. 33, no. 5, pp. 676-686, May 1998.
- [5] E. G. Friedman, "Introduction" in *Clock Distribution Networks in VLSI Circuits and Systems*, E. G. Friedman, Ed. New York: IEEE Press, 1995, pp. 1-36.
- [6] D. W. Bailey and B. J. Benschneider, "Clocking design and analysis for a 600-MHz Alpha microprocessor," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1627-1633, Nov. 1998.
- [7] G. Geannopoulos and X. Dai, "An adaptive digital deskewing circuit for clock distribution networks," in *ISSCC 1998 Dig. Tech. Paper*, Feb. 1998, pp. 400-401.
- [8] J. Alvarez *et al.*, "450MHz PowerPC™ microprocessor with enhanced instruction set and copper interconnect," in *ISSCC 1999 Dig. Tech. Paper*, Feb. 1999, pp. 96-97.
- [9] V. R. von Kaenel, "A high-speed, low-power clock generator for a microprocessor application," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1634-1639, Nov. 1998.
- [10] S. Sidiropoulos and M. Horowitz, "A semidigital dual delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 32, no. 11, pp. 1683-1692, Nov. 1997.
- [11] J. Montanaro *et al.*, "A 160-MHz, 32-b, 0.5-W CMOS RISC microprocessor," *IEEE J. Solid-State Circuits*, vol. 31, no. 11, pp. 1703-1714, Nov. 1996.
- [12] G. M. Blair, "Low-power double-edge triggered flipflop," *Electron. Lett.*, vol. 33, no. 10, pp. 845-847, May 1997.
- [13] L. Sigal *et al.*, "Circuit design techniques for the high-performance CMOS IBM S/390 parallel enterprise server G4 microprocessor," *IBM J. Res. Develop.*, vol. 41, no. 4/5, pp. 489-501, Jul./Sep. 1997.
- [14] S. Hesley *et al.*, "A 7th-generation x86 microprocessor," in *ISSCC 1999 Dig. Tech. Paper*, Feb. 1999, pp. 92-93