Closing the Gap Between ASIC and Custom: An ASIC Perspective

D. G. Chinnery and K. Keutzer

Department of Electrical Engineering and Computer Sciences

University of California at Berkeley

{chinnery,keutzer}@eecs.berkeley.edu

ABSTRACT

We investigate the differences in speed between application-specific integrated circuits and custom integrated circuits when each are implemented in the same process technology, with some examples in 0.25 micron CMOS. We first attempt to account for the elements that make the performance different and then examine ways in which tools and methodologies may close the performance gap between application-specific integrated circuits and custom circuits.

Keywords

ASIC, clock frequency, clock speed, comparison, custom.

1. INTRODUCTION

Speed of average application-specific integrated circuits (ASICs) lags that of the fastest custom circuits in the same processing geometry by factors of six to eight. There doesn't seem to be any clear consensus on the source of this performance difference, and occasionally one encounters an implicit prejudice that poor design skills in ASIC designers compounded by poor computer-aided design (CAD) tools are at fault. In this paper we aim first to develop a comprehensive rationale for the differences between the speed of custom integrated circuits (ICs) and ASICs. We then aim to constructively explore the ways in which tools and methodologies can close this gap.

We begin by giving examples of performance of both custom ICs and ASICs. We then give a top-level overview of what we feel accounts for the difference between custom ICs and ASICs. We then go through each of the factors that contribute to the difference in detail.

2. ASIC AND CUSTOM COMPARISON

To quantify the differences between ASIC and custom chip speeds, we first examine speeds of high performance designs and typical ASIC designs in 0.25um technology. When we refer to a technology, we are referring to fabrication processes with similar design rules and transistor channel lengths, and with the same interconnect (specifically, aluminum interconnect for the 0.25um technology considered).

Among the fastest 0.25um commercially produced processors is the Alpha 21264A, which runs at 750MHz, with a 2.1V supply voltage and 90W power consumption when in operation. It has an area of 2.25cm^2[1]. This processor uses dynamic logic and heavy pipelining to achieve this speed[10][18].

IBM has designed a 1.0GHz integer processor in 0.25um technology, with a 1.8V supply voltage, and 9.8mm^2 area, that consumes 6.3W of power[21].

ASIC microprocessors are not typical, because they bear more architectural similarity to custom microprocessors, but they present a good mid-point between custom design and a typical ASIC design. Tensilica has a high performance 250MHz 0.25um ASIC processor[2], with an area of about 4mm^2 (depending on the configuration). Finally, simply based on anecdotal information we postulate that average 0.25um ASICs run at between 120MHz and 150MHz, and high speed network ASICs may run at up to 200MHz in 0.25um technology. Of course, one may find ASICs that operate at slower speeds, but in these devices we presume that performance was specifically *not* a criterion. Thus, at the outset, we can see that custom ICs operate 6× to 8× faster than ASICs in the same process. At first glance this gap seems staggering. If we put the speed improvement due to one process generation (e.g. 0.35um to 0.25um) as 1.5× then this gap is equivalent to that of five process generations or nearly a decade of process improvement. In the following section we try to more precisely describe the factors that result in this significant speed differential.

3. FACTORS CONTRIBUTING TO THE DIFFERENCES

The following gives our overview of the maximum contribution of various factors to the speed differential between ASICs and custom ICs.

- ×4.00 through architecture and logic design: heavy pipelining/few logic levels between registers
- ×1.25 by good floorplanning and placement
- ×1.25 with clever sizing of transistors and wires for speed and good circuit design
- ×1.50 from use of dynamic logic on critical paths, instead of static CMOS logic
- ×1.90 due to process variation and accessibility

DAC 2000, Los Angeles, California

When designing a custom processor, the designer has a full range of choices in design style. These include architecture and micro-architecture, logic design, floorplanning and physical placement, and choice of logic family. Additionally, circuits can be optimized by hand and transistors individually sized for speed, lower power, and lower area. Thus if the speed improvements associated with each element above are approximately correct then custom circuits could run 18× faster than their average ASIC counterparts. In practice, even the best custom designs don't take full advantage of all these potential advantages over an ASIC.

Before reviewing each of the factors that contribute to speed differentials it will be useful to review how the speed of an integrated circuit is determined. The speed of a circuit is determined by the delay of its longest critical path, and the length of the critical path is a function of gate delays, wiring delays, set-up and hold-times, clock-to-Q (the delay from when a clock signal arrives at a latch to when the latch output stabilizes), and clock skew[26]. To improve the speed of an integrated circuit requires reducing the delay of one or more of these elements. How these elements of the critical-path delay of a circuit are reduced by factors such as micro-architecture and pipelining will be detailed in each of the following sections.

4. MICRO-ARCHITECTURE AND HARDWARE IMPLEMENTATION: PIPELINING AND LOGIC LEVELS, AND LOGIC DESIGN

Pipelines place additional latches or registers in long chains of logic, reducing the length of the critical path, and allowing time stealing between pipeline stages with multi-phase clocking.

The 1.0GHz IBM PowerPC chip has a single-issue pipeline with four stages[22]. The Alpha 21264A processor has seven pipeline stages, but it has out-of-order and speculative execution. Similarly, the Tensilica ASIC processor has a single-issue five stage pipeline[9], whereas typical ASIC designs may have no pipelining and significantly longer critical paths.

A metric for expressing the number of logic levels in a design is in terms of the number of fanout-of-four (FO4) inverter delays (an inverter driving four times its input capacitance)[15]. There are 15 FO4 delays in the Alpha 21264[12][15], and 13 FO4 delays in the 1.0 GHz IBM PowerPC[1]. An ASIC typically has significantly more levels of logic on the critical path; Tensilica's Xtensa processor is estimated to have about 44 FO4 delays[2].

Estimating the pipelining overheads, such as clock skew and latch overheads, as about 30% for an ASIC design, the Tensilica pipelined ASIC processor with five stages is about 3.8 times faster due to pipelining. Estimating the clock skew and latch overheads as about 20% for a custom design, the IBM PowerPC processor with four pipeline stages is about 3.4 times faster with pipelining.

4.1 What's the problem?

For pipelining to be of value, multiple tasks must be able to be initiated in parallel, and branches in execution will diminish performance. Many designs, such as bus interfaces, have a tight interaction with their environment in which each execution cycle depends on new primary inputs and branches are common. In such cases, it is not clear how an ASIC may be reorganized to allow pipelining. Simply increasing the clock speed by adding latches would only increase latency due to the additional latch setup and hold times.

ASIC tools have problems with complicated multi-phase clocking schemes that would allow time borrowing between pipeline stages to increase speed. While there are level-sensitive latches in some ASIC libraries, typically only one or two clock phases are used.

Pipelining ASICs is also limited by the speed of registers in the pipeline, and greater clock skew than carefully designed custom ICs. There is typically 10% clock skew or more for ASICs, compared with about 5% clock skew for a high quality custom design of clocking trees. The 600MHz Alpha 21264 has 75ps global clock skew, or about 5%[12]. Comparing the absolute differences in clock skews, there is about a 10% increase in speed due to custom quality clock skew alone.

Registers and latches in ASICs have additional overheads as they have to be more tolerant to clock skew, and require a far larger absolute segment of the clock cycle, whereas custom designs can include some logic within the latch to reduce the overhead. At high speeds in custom designs, latches still take a significant component of the cycle time, 15% in the Alpha 21264 processor[12].

Custom designs may also show superior logic-level design of regular structures such as adders, multipliers, and other datapath elements. They achieve fewer levels of logic on the critical path with more compact, complex logic cells and by combining logic with the latches. In a custom processor, careful design can balance the logic in pipeline stages *after placement*, ensuring that the delays in each stage are close, whereas an ASIC may have unbalanced pipeline stages resulting in more levels of logic on the critical path.

Additional processing speed can be achieved by issuing multiple instructions, but this requires speculative execution with additional complex hardware logic (such as forwarding and branch prediction) and more pipeline stages, unless there is a high degree of parallelism in instructions. There is a trade-off between issuing more instructions simultaneously and the penalties for branch misprediction and data hazards, which reduce the performance, and additional hardware and design cost[16]. The Alpha 21264 can issue up to six instructions per cycle, and has four integer execution units and two floating-point execution units[18], giving it significantly faster performance when instruction parallelism can be exploited.

---

[1] Calculated from the effective transistor channel length of 0.15um[21], using the rule of thumb that FO4 delay is $0.5L_{eff}$ in nanoseconds, the FO4 delay is 75ps. This gives 13 FO4 delays in a clock cycle. (Calculation courtesy of Andrew Chang.)

[2] Assuming 0.18um effective transistor channel length in a typical 0.25um ASIC process. (Calculation courtesy of Andrew Chang and Ricardo Gonzalez.)

4.2 What can we do about it?

If processing the data is interdependent, there is little that can be done to pipeline ASIC designs. If data can be processed in parallel, it should be possible to pipeline circuitry performing the calculations or have parallel processing units, increasing the speed significantly, especially if large amounts of data can be processed in parallel.

Fast datapath designs, such as carry-lookahead and carry-select adders and other regular elements, do exist in pre-designed libraries, but are not automatically invoked in register-transfer level logic synthesis of ASICs. Use of these predefined macro cells for an ASIC can significantly improve the resulting design, by reducing the number of logic levels for implementing complex logic functions and reducing the area taken up by logic[5].

5. FLOORPLANNING, PLACEMENT, AND ROUTING

Wire-delays associated with "global" wires between physical modules can be a dominant portion of the total path delay. The delay associated with wires depends on the length of the wire, the width and aspect ratios of the wire, and on proper driving of the wire. Proper driving of a wire depends on sizing of drivers and insertion of repeaters, but the primary factor in wire delay is wire length. Wire length is obviously dependent on placement, which in turn depends on floorplanning, but is also influenced by the quality of routing.

5.1 What's the problem?

We compared localizing critical paths to within a module (emulating careful floorplanning) to a critical path distributed across a 100mm chip. Based on our simulations[3], using careful floorplanning and placement to minimize wire lengths may increase circuit speed by up to 25%.

5.2 What can we do about it?

Custom ICs are typically manually floorplanned. A number of tools are now reaching the ASIC market to facilitate chip-level floorplanning. These should diminish the gap between ASIC and custom designs due to floorplanning.

In addition, tools with the capacity to identify similar structures that may be abutted or placed next to each other appropriately will reduce area, reducing wire lengths and increasing performance. A bit slice may be laid out automatically then tiled, rather than the circuitry being placed without considering that it may be abutted[5]. Regular data paths can be best laid out by hand or tiling slices for abutment, but custom design is not as effective with irregular structures.

6. CIRCUITS, TRANSISTOR AND WIRE SIZING

In an ideal design, each circuit is optimally crafted from transistors and each transistor is individually sized to meet the drive requirements of the capacitive load it faces, subject to timing constraints. Also wires may be widened to reduce the delays (proportional to the product of resistance and capacitance) by reducing the resistance. Additional buffers may be included to drive large capacitive loads that would be charged and discharged too slowly otherwise. Only in a custom design methodology can this ideal be realized. Any current ASIC methodology requires cell selection from a fixed library, where transistor sizes and drive strengths are determined by the choices in the library, and wire sizes are fixed.

6.1 What's the problem?

One element of the performance degradation of ASIC designs, relative to custom, is due to the limits of the ability of ASIC libraries to approximate custom designs in their transistor-level design and in their transistor sizing. ASIC cells typically include design guard banding, such as buffering flip-flops, which introduce overhead. More fundamentally, the discrete transistor sizes of a library only approximate the continuous transistor sizing of a custom design. With a rich library of sizes the performance impact of discrete sizes may be 2% to 7% or less[13][11]. However, many ASICs still do not use good standard cell libraries with varied drive strengths and both polarities of gates. A cell library with only two drive strengths may be 25% slower than an ASIC library with a rich selection of drive strengths and buffer sizes, as well as dual polarities for functions (gates with and without negated output) [23]. A richer library also reduces circuit area[19].

6.2 What can we do about it?

ASIC designs should be using standard cell libraries with dual gate polarities and several drive sizes of each gate. If appropriate libraries are used, then ASIC designs are not lagging behind custom designs in this area, when the logic design is optimal.

Initial logic synthesis may choose drive strengths using estimations for wire lengths and the net load a gate has to drive, but this will differ from that in the final layout. After layout, transistors can be resized accounting for the drive strengths required to send signals across the circuit, and buffers can be inserted or removed as necessary. Sizing transistors minimally to reduce power consumption, except on critical paths where they are optimally sized to meet speed requirements, can make a speed difference of 20% or more[7].

With "liquid cells" or resynthesis, later arriving signals can be routed closer to the gate output and transistors moved to maximize the adjacent drains and sources for diffusion sharing[17]. Iterative transistor resizing and resynthesis can improve speeds by 20%[8]. Tools for wire sizing along with transistor sizing may be available in the future (e.g. [6]).

7. DYNAMIC LOGIC

Dynamic logic can be used to speed up critical paths within the circuit, by reducing gate delays. It is significantly faster than static CMOS logic and smaller area, but requires careful design to ensure no glitching of input signals. Static CMOS logic has far less sensitivity to noise and consumes less power.

Both the IBM PowerPC integer processor and the Alpha 21264 make use of dynamic logic for increased circuit speed[10][22]. Whereas an ASIC design, such as the Tensilica processor, is mapped to a static CMOS library, which does not take advantage of faster speeds achievable with dynamic logic.

[3] Using BACPAC
http://www-device.EECS.Berkeley.EDU/~dennis/bacpac/

7.1 What's the problem?

Dynamic logic libraries are not available for ASIC design, because of the difficulties and care required with dynamic logic. Design of dynamic logic requires careful consideration of noise and power consumption. Dynamic logic is particularly susceptible to noise, as any glitches on input voltages may cause a discharge of the charge stored, which should only occur when the logic function evaluates to false. Additionally, dynamic logic has higher power consumption, requiring careful design of power distribution, and clock distribution as well; the clock determines when precharging occurs, and inputs must not glitch during or after the precharge. These problems become more pronounced with deeper submicron technologies.

Dynamic logic functions used in the IBM 1.0 GHz design are 50% to 100% faster than static CMOS combinational logic with the same functionality[21][Nowka, personal communication]. This implies that sequential circuitry using dynamic logic will be about 50% faster.

7.2 What can we do about it?

There has been some progress in dynamic logic circuit synthesis[25], but it has yet to produce commercially available libraries. It is used as an aid to in-house custom design. It seems unlikely that the methodological obstacles described above will be overcome, to enable dynamic logic synthesis for ASIC designs.

ASIC designs can take advantage of custom logic if high-speed custom macro cells are provided for particular functions such as barrel shifters, adders and multipliers.

8. PROCESS VARIATION AND ACCESSIBILITY

Typical ASIC chips fabricated on a typical process may be 60% to 70% faster than the worst case speeds quoted by ASIC library estimates for the worse fabrication plants[4][2]. In addition, the fastest speeds produced in a plant may be 20% to 40% faster[5], but without sufficient yield for low cost ASIC use. Overall, the highest speed custom chips fabricated may be 90% faster than an equivalent ASIC design, running at worst case speeds produced at a slower fabrication plant in the same technology, due to process variation and accessibility.

8.1.1 Variation in speeds from within a plant

There are several types of process variations that can occur within a plant: line-to-line; wafer-to-wafer; die-to-die, and intra-die. These process variations cause the delays of wires and gates within a chip to vary, and chips are produced with a range of working speeds. Some chips with minor imperfections will only operate correctly at slower speeds.

It appears that when Intel and AMD start using a new technology, the variation is about 30% to 40%[6]. This variation decreases as the process matures, but additional improvements to the process or the design of the custom ICs are possible. In Intel's 0.25um 856 process, a shrink of 5% was achieved, giving a speed improvement of 18%[4].

Thus within a technology generation, a 50% to 60% range in produced clock speeds of the identical custom IC design may be expected from the start to the end due to process variation. However, there is often some room for improvement in design and changes in the custom IC design, and down-binning of chips with higher clock frequency to meet demand (when stores of slower versions are depleted, evidenced by the ease of over-clocking many chips), which extend the range of clock speeds typically seen within a technology generation.

8.1.2 Variation between fabrication plants

Additionally, in the same technology, the speed of identical ASIC designs (but with different standard cell libraries and resulting synthesized circuitry for the different foundries) may vary by 20% to 25% between fabrication plants of different companies[2].

Within a company, there are standards to ensure the same yields and quality at different fabrication plants with the same technology[20], and the difference in speeds is likely to be minimal.

8.2 What's the problem?

The design rules for an ASIC process must be fixed for standard cell library design. If there are process improvements, then the library must be redesigned to take advantage of these, and if it is not then potentially as much as a 20% possible improvement in speed is lost.

Fabrication plants won't offer ASIC customers the top chip speed off the production line, as they cannot guarantee a sufficiently high yield for this to be profitable. The fabrication plant guarantees that they can produce an ASIC chip with a certain speed. This speed is limited by the worst speeds off the production line, but chips capable of faster speeds are produced.

ASIC designers may not have access to the best fabrication plants in a particular technology generation for production of their chips.

8.3 What can we do about it?

ASIC designs are typically easy to migrate between technology generations, as they are retargetable to different processes, and thus can easily switch to use the best fabrication plants available for ASIC production. Whereas custom designs cannot simply be mapped to a new gate library for the next technology generation, but must have transistors resized and circuits altered to account for design rules, voltage, current and power considerations not scaling linearly. For high volume custom designs where high speeds sell, it is profitable to have a design team making changes to take best advantage of a process.

---

[4] Calculated from speeds quoted for the Tensilica Xtensa processor on page 9 of the Overview Handbook[2].

[5] Estimate based on communications with Spanos, Hu and Orshansky (calculated speed variation from process variations), and frequencies of new processors produced in a new technology where down binning is unlikely.

[6] For example, comparing the range of speeds, from 533 to 733MHz, that Intel was initially producing in 0.18um technology from October 25, 1999[14].

ASIC designs are not necessarily limited by the worst speeds off the production line. If the designers can afford to test produced chips and verify correct operation at higher speeds, then they can use them at greater speeds. This may allow a 30% to 40% improvement in speed over worst-case speeds[2].

Fabrication companies do provide new ASIC libraries throughout a technology generation as their technology matures. They update libraries and worst-case speeds with the higher speeds for processes that have shorter effective transistor channel lengths for speed. These libraries are close in speed to contemporary high-speed custom processes. For example, IBM's CMOS7SF 0.18um SA-27E process has 0.11um effective transistor lengths and copper interconnect[3], and IBM's CMOS7S 0.18um process with 0.12um effective transistor lengths has FO4 delay of 55ps[24].

9. SUMMARY AND CONCLUSIONS

We have given examples of performance of both custom ICs and ASICs. We then gave a top-level overview of what we feel accounts for the principal differences in performance between custom ICs and ASICs. We then examined in detail each of the factors of performance and attempted to justify the numeric percentage of our factors.

Based on our analysis we believe that the influence of the factors of floorplanning and circuit design, while significant, are probably overstated in their importance in the performance gap between ASIC and custom ICs. From our analysis the two most significant factors are pipelining and process variation. It appears to us that these two factors alone account for all except a factor of about 2 to 3×. The use of dynamic-logic families is a third significant influence resulting in about 1.5×. Adding this factor to pipelining and process variation accounts for all but a factor of about 1.6×.

One key point in interpreting our results is that we have focused on *entire IC designs*. It is certainly true that if one restricts the focus to a single circuit element, such as a barrel-shifter for example, the influence of custom circuit design techniques may appear much more significant than we have indicated. However, when such elements are integrated into an entire path, such as in an ALU their individual significance is naturally reduced.

Another important caveat is that because of space restrictions we have focused exclusively on speed differences between ASIC and custom ICs and not on area or power differences. Viewed from the standpoint of area our results and conclusions would be significantly different.

These results may be viewed either optimistically or pessimistically. Optimistically these results point out that ASIC design methodologies are not as inefficient as has been presumed. Pessimistically they do imply that even with tool and library improvements the performance gap between ASIC and custom ICs is likely to remain a large one.

10. ACKNOWLEDGMENTS

Thanks to Ricardo Gonzalez, Earl Killian and Albert Wang of Tensilica; Desmond Kirkpatrick of Intel; Dennis Sylvester and Michael Keating of Synopsys; Brock Barton of Texas Instruments; and Randy Allmon of Compaq. Thanks also to Tom Burd, Mikhail Orshansky, Prof. Chenming Hu and Prof. Costas Spanos of UC Berkeley; and Andrew Chang and Prof. William Dally of Stanford University.

11. REFERENCES

[1] API Products: 21264A. http://www.alpha-processor.com/products/21264A-processor.asp.

[2] Archide, R. (ed.). Xtensa Application Specific Microprocessor Solutions Overview Handbook, a Summary of the Xtensa Data Sheet. February 2000. http://www.tensilica.com/dl/handbook.pdf

[3] ASIC SA-27E Databook http://www.chips.ibm.com/products/asics/document/databook/220802_1.pdf

[4] Brand, A., et al. Intel's 0.25 Micron, 2.0 Volts Logic Process Technology, Intel Technology Journal, Q3'98. http://developer.intel.com/technology/itj/q31998/pdf/p856.pdf.

[5] Chang, A. VLSI Datapath Choices: Cell-Based Versus Full-Custom", SM Thesis, Massachusetts Institute of Technology, February 1998. ftp://cva.stanford.edu/pub/publications/achang_masterworks980427.pdf

[6] Chen, C., Chu, C., and Wong, D. Fast and Exact Simultaneous Gate and Wire Sizing by Lagrangian Relaxation, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, vol. 18-7, July 1999, 1014-1025.

[7] Fishburn, J., and Dunlop, A. TILOS: A Posynomial Programming Approach to Transistor Sizing. Proc. of ICCAD '85, 326-328.

[8] Gavrilov, S., et al. Library-Less Synthesis for Static CMOS Combinational Logic Circuits, Proc. of ICCAD '97, 658-663.

[9] Gonzalez, R. Configurable and Extensible Processors Change System Design. Hot Chips 11, 1999. ftp://www.hotchips.org/pub/hotc7to11cd/hc99/hc11_pdf/hc99.s4.3.Gonzalez.pdf

[10] Gowan, M., Biro, L., and Jackson, D. Power Considerations in the Design of the Alpha 21264 Microprocessor. Proc. of DAC '98, 726-731.

[11] Grodstein, J., et al. A Delay Model for Logic Synthesis of Continuously-Sized Networks, Proc. of ICCAD '95, 458-462.

[12] Gronowski, P., et al. High-Performance Microprocessor Design. IEEE Journal of Solid-State Circuits, vol. 33-5, May 1998, 676-686.

[13] Haddad, R., van Ginneken, L., and Shenoy, N. Discrete Drive Selection for Continuous Sizing, Proc. of ICCD '97, 110-115.

[14] Hare, C. 586/686 Processors Chart. http://users.erols.com/chare/586.htm.

[15] Harris, D., and Horowitz, M. Skew-Tolerant Domino Circuits, IEEE Journal of Solid-State Circuits, vol. 32-11, November 1997, 1702-1711.

[16] Hennessy, J., and Patterson, D. Computer Architecture: A Quantitative Approach, 2nd Ed. Morgan Kaufmann, San Francisco CA, 1996.

[17] Hill, D. Sc2: A Hybrid Automatic Layout System. Proc. of ICCAD '85, 172-174.

[18] Kessler, R., McLellan, E., and Webb, D. The Alpha 21264 Microprocessor Architecture. Proc. ICCD '98, 90-95.

[19] Keutzer, K., Kolwicz, K., and Lega, M. Impact of Library Size on the Quality of Automated Synthesis. Proc. of ICCAD '87, 120-123.

[20] McDonald, C. The Evolution of Intel's Copy Exactly! Technology Transfer Method, Intel Technology Journal, Q4'98. http://developer.intel.com/technology/itj/q41998/pdf/copyexactly.pdf.

[21] Nowka, K., and Galambos, T. Circuit Design Techniques for a Gigahertz Integer Microprocessor, Proc. Of ICCD '98, 11-16.

[22] Posluszny, S., et al. Design Methodology of a 1.0 GHz Microprocessor. Proc. of ICCD '98, 17-23.

[23] Scott, K., and Keutzer, K. Improving Cell Libraries for Synthesis. Proc. of CICC '94, 128-131.

[24] Stasiak, D., et al. A 2nd Generation 440ps SOI 64b Adder. Proc. of ISSCC 2000, 288-289 and 465.

[25] Thorp, T., Yee, G., and Sechen, C. Domino Logic Synthesis Using Complex Static Gates. Proc. of ICCAD'98, 242-247.

[26] Weste, N., and Eshraghian, K. Principles of CMOS VLSI Design. Addison-Wesley, 1992.