

# Analysis of Technology Trends: Making A Case for Architectural Adaptation in Custom Data-Paths

Rajesh Satapathy and Rajesh Gupta

Information and Computer Science, University of California, Irvine, CA 92697.

*Email: {rgupta, rajeshs}@ics.uci.edu*

## ABSTRACT

The paper presents an analysis of technology trends based on the data available from the recently released National Technology Roadmap for Semiconductors (NTRS 1997). This analysis shows that increasing clock rates and system diameter in clock periods will make efficient management of communication and coordination increasingly critical. Due to the decreasing cost of logic versus interconnect and the electrical necessity of signal regeneration to counter worsening effect of interconnect geometries, use of configurable logic blocks even in custom data-paths presents a unique opportunity to customize bindings, mechanisms, and policies which comprise the interaction of processing, memory, I/O and communication resources. This programming flexibility, or “customizability,” can provide the key to achieving robust high performance.

We use the results of this study to make a case for evolution of computer architectures into “Application Adaptive” (AA) architectures. These architectures exploit the capability of the underlying hardware to reconfigure logic to achieve system-level cost/performance goals by extensive analysis and profiling of application data and runtime characteristics. A key distinction made by AA architectures against traditional custom-computing machines is that architectural flexibility is used to customize architectural mechanisms and policies (instead of building additional functional resources – an approach commonly adopted by custom computing machines). Thus relatively small amounts of reconfigurable circuit blocks can be leveraged to yield high performance on a per application basis.

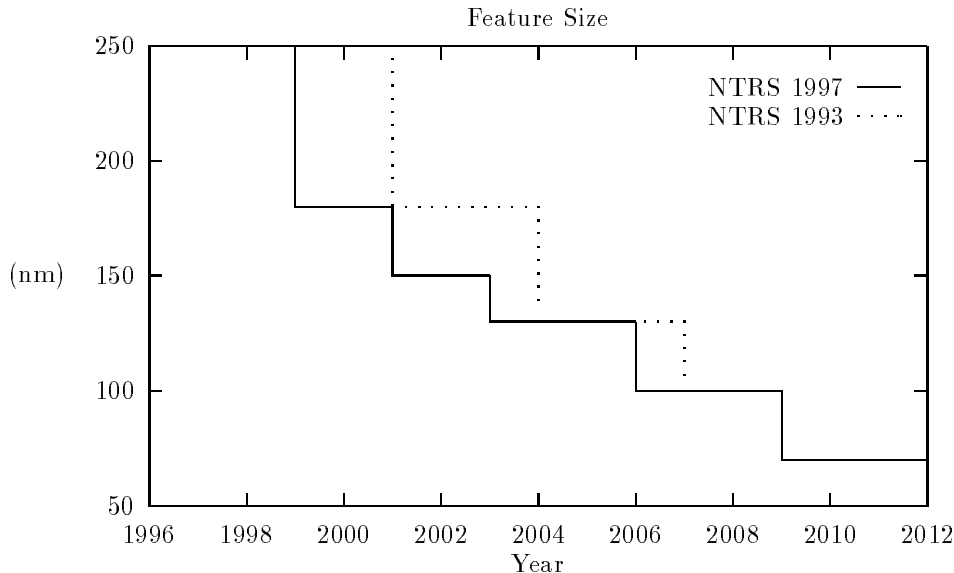
**Keywords:** Key words: NTRS 1997, technology scaling, interconnect strategies, application adaptive architectures.

## 1. INTRODUCTION

The pace of semiconductor technology evolution shows no signs of slowing down. Indeed, the projections from the National Technology Roadmap for Semiconductors (NTRS) have been revised in three years to show an acceleration of technological maturity for most deep submicron processes. Figure 1 shows a plot of (effective channel length) feature size versus the year of introduction. The solid line indicates the projections from the most recent industry survey. Between the two surveys shown, the introduction of the most advanced processes has accelerated by one to three years. For instance, the 180nm process is introduced in 1999 two year earlier than the NTRS 19993 survey (indicated by the dotted line). In the coming years, as the attention shifts to processes with channel length below 100nm, it is likely that the introduction for future processes may actually happen sooner than currently projected.

In evolutionary terms, there are no surprises *per se* in each of the component area or even the process technology itself as specified by the NTRS 1997. However, the cumulative impact of continuing trends in circuit density and performance on system architectures, from small-scale embedded processors to high-performance multi-node machines, is nothing short of phenomenal. This papers outlines some of the important design technology implications of microelectronic technology advances in the context of computer architectures.

We present an examination of the technology scaling in the next section, followed by analysis of impact on VLSI design technology and computer system architectures. To make our case for dramatic changes that are likely to occur in the architectural design of future processors, we rely on an extensive body circuit simulation data using process



**Figure 1.** Trends in Device Scaling

parameters as can be reasonably expected using the projections from the NTRS 1997 survey by the Semiconductor Industry Association.\*

## 2. TECHNOLOGY EVOLUTION AND SCALING

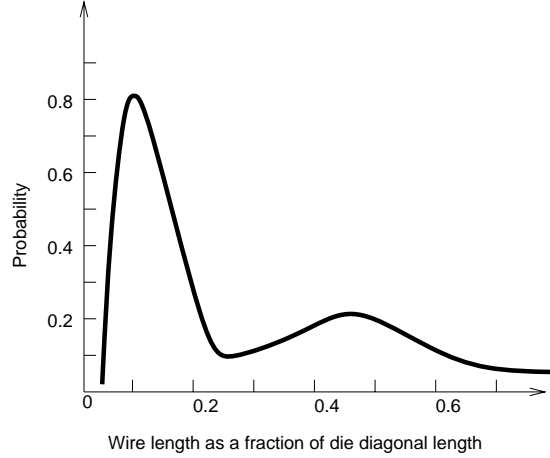
Figure 1 shows the projected trends in minimum device feature size for semiconductor devices. Assume that feature sizes are scaled by factor  $S$ . Then, the intrinsic gate delay scaling (limited by velocity saturation effects) scales by  $1/S$  for short channel devices.<sup>11</sup> This scaling is independent of the voltage scaling used.<sup>†</sup> Since the speed of a logic gate is proportional to the load capacitance, scaling of device dimensions leads to decreasing gate delays. However, the interconnect introduces a range of parasitic effects that are significantly pronounced at the deep sub-micron technology of three generations hence. This interconnect dominance has been a focus of much attention in the circuit design for deep sub-micron semiconductor processes (see<sup>7</sup> for an introduction to the problem).

To understand the nature of the interconnect for VLSI processors, consider the typical distribution of interconnect length on a chip shown in Figure 2 from.<sup>12</sup> The distribution has two peaks, one around at 10% and at 50% of the die-size. The die-size is approximately square-root of the chip area. The *average* length of the interconnect can be approximated as  $\bar{L} = \frac{\sqrt{\text{chip area}}}{3}$ .<sup>15</sup> The total interconnect can be roughly divided into two categories: local and global interconnect. The first peak in Figure 2 represents the local interconnect length, whereas the second peak is due to the global interconnect nets. With scaling, the local interconnect scales as device shrinks whereas global interconnect actually increases due to increase in die density and size.<sup>13</sup> Thus the wire-length scaling can be divided into two parts as follows:

$$S_L = \begin{cases} S & \text{local interconnect} \\ \frac{1}{S} & \text{global interconnect} \end{cases}$$

\*<http://www.sematech.org/public/roadmap/>

<sup>†</sup>Typically, keeping voltage scaling same as feature size scaling maintains electrical field patterns to the first order, thus avoiding potential secondary breakdown effects. However, it is not always possible to do so due to noise immunity and market-place reasons. Consider the scaling scenario in which all dimensions of the MOS devices scale by factor  $S$ , while the voltage levels scale by factor  $U$ . In this scenario, the current density scales by factor  $S^3/U^2$ .

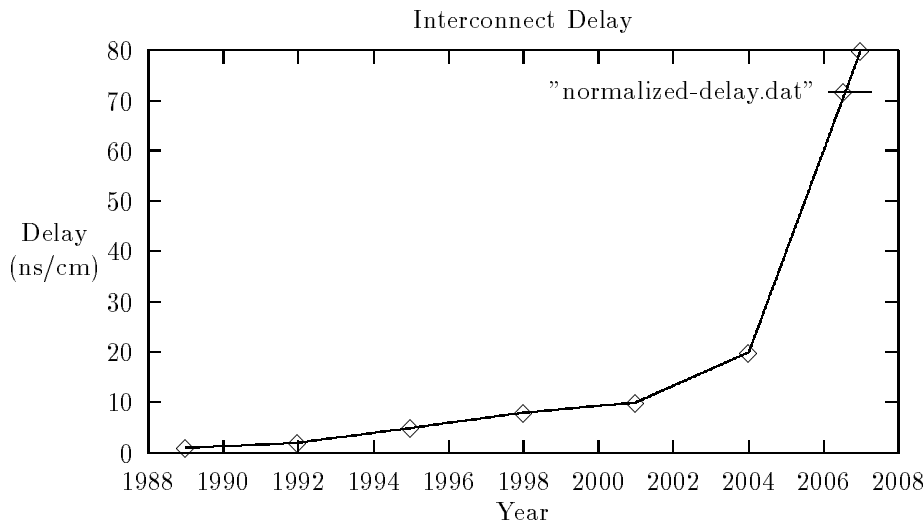


**Figure 2.** Distribution of Interconnect Lengths in Logic Devices

The resistance scaling is  $S_R = \frac{S_L}{S^2}$  whereas the RC delay scaling is given by

$$S_{RC} = S_C \times S_R = \frac{S \cdot S_L}{S} \times S_R = \frac{S_L^2}{S^2}$$

Thus RC is constant for local interconnect while gate delay is decreasing. RC grows as  $S^4$  for the global interconnect. Let us now consider, in absolute terms, the characteristics of circuits and devices that would be available in the year 2007. Figure 3 plots the normalized delay of the interconnect (per unit length) which is about 80 ns/mm. Figure 4 shows the increasing role of interconnect in absolute terms. For complex microprocessors, the on-chip cycle time is limited to 1 nanosecond primarily to combat the increasing parasitic (inductive) effects of metal interconnect, while the unit gate delay (inverter with fanout of two) scales down to 20 pico-seconds. Thus modern day control logic consisting of 7-8 logic stages per cycle would form less than 20% of the total cycle time. This clearly challenges the fundamental design trade-off today that tries to simplify the amount of logic per stage in the interest of reducing



**Figure 3.** Normalized Delay of Interconnect Metal

the cycle time.<sup>10</sup> In addition, this points to a sharply reduced *marginal* cost of per stage logic complexity on the circuit-level performance.

### 3. CIRCUIT AND ARCHITECTURAL IMPLICATIONS OF SCALING

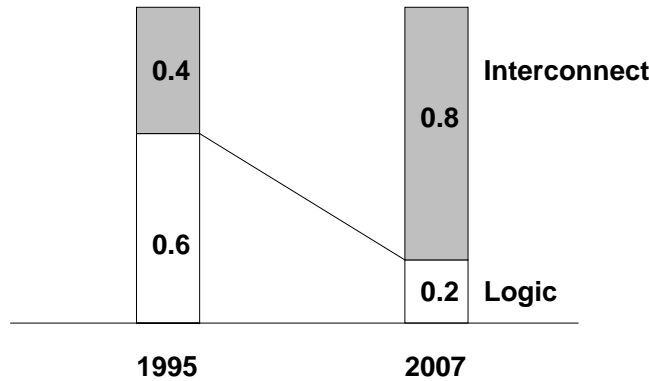
The impact of a technology scaling that affects gate versus interconnect delays differently, goes far beyond the number of logical stages that can be packed into a given cycle period. For instance, the circuit structures and system architectures can be chosen that rely on extensive *local logic* to improve circuit/system performance. The local logic can consist of state machines that direct the flow of data or modify control based on immediate application needs. Thus an important aspect of chip-level system design that will be qualitatively challenged by these technology trends is the role of reprogrammable logic on chip designs (even for custom/semi-custom designs). This is best understood by examining the effect of technology scaling on the the length of wire, or  $L_{crit}$ , that is equivalent to a unit gate delay. Roughly speaking, it is a length of wire between two logic stages beyond which the marginal delay of additional logic stage is less than the cost of the interconnect. It can be shown that  $L_{crit}$  is fairly independent of the actual sizes of the transistors in the logic stages.<sup>14</sup> The *critical length* can be computed as:

$$L_{crit} = \sqrt{\frac{t_{pgate}}{0.38rc}}$$

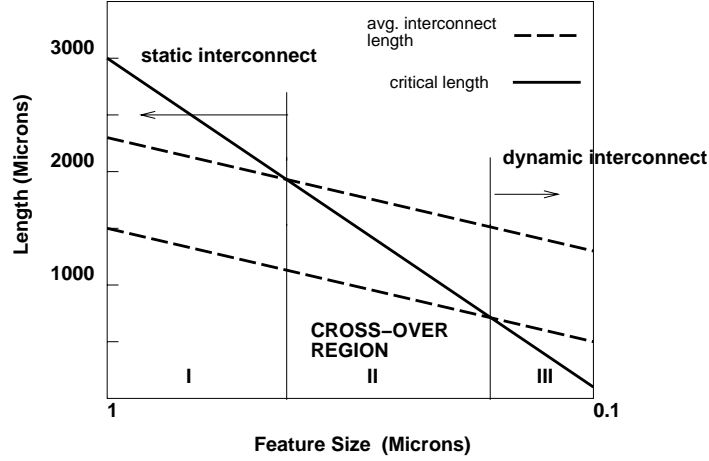
where  $t_{pgate}$  is the unit gate delay that scales by  $S$ .  $r$  and  $c$  are unit length resistance and capacitance respectively. The scaling of  $rc$  delay is given as (using local interconnect scaling)

$$S_{rc} = \frac{S_R}{S_L} \cdot \frac{S_C}{S_L} = \frac{S_{RC}}{S_L^2} = \frac{1}{S^2}$$

Thus the critical length scales as  $S^{3/2}$ . This scaling corresponds to a reduction by a factor of 32 for a 10X reduction in feature size. An estimate of the absolute value of the interconnect lengths is provided by considering the metal pitch in sub-micron devices. Pitch for the finest interconnect is projected at 0.4-0.6  $\mu$ . On logic devices, average interconnect length,  $\bar{L}$ , is approximately 1000X to 10,000X the pitch. In view of the interconnect distributions in Figure 2, this means that local interconnect is 300X to 3000X pitch. This scaling trends is shown in Figure 5. Within three generations of evolutions in process technology we can expect a **cross-over point where the critical length is larger than the average inter-connect length between logic blocks that are located in the same cycle time**. Depending upon the average length of local interconnect this cross-over could occur within the Region II indicated as cross-over region in Figure 5. Once past this cross-over region, the delay due to the **average**



**Figure 4.** Fraction of cycle time devoted to gate and interconnect delays

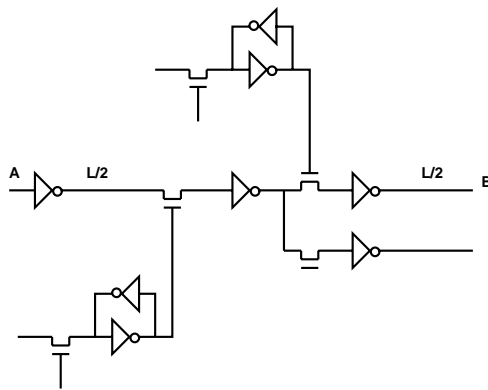


**Figure 5.** Inter-Connect Critical Length: Trends show a cross-over point beyond which the wire critical length is longer than the average interconnect length.

interconnect length would exceed switching delays. For a wire of length  $L$ , the total inter-connect delay with  $M$  “repeating buffers” is given as

$$t = 0.38rc \left( \frac{L}{M} \right)^2 M + (M - 1)t_{pgate}$$

where  $t_{pgate}$  is the propagation delay of the buffer. Optimal buffering stages can be found by setting  $\partial t / \partial M$  to zero. A rule of thumb in circuit designs is provide electrical buffering for signal integrity at least per unit critical length of wire. This implies that due to purely electrical reasons it would be preferred to include one or more inter-connect buffers within a cycle time to mitigate the effect of delay due to average local interconnect. Such a buffer gate when combined with a weak-feedback device would form core of a storage element that presents less than 50% switching delay overhead (from 20ps to 30ps). Figure 6 shows a schematic diagram of the interconnect between two blocks that is made programmable using reprogrammable devices. For wirelengths,  $L$ , larger than the critical lengths, such a circuit would present a shorter delay from  $A$  to  $B$  than a direct wire connection. Indeed, SPICE simulations show that as channel length scales down to below 100 nm, the wirelength  $L$  for which a reprogrammable circuit provides shorter delay far interconnect lengths as low as a couple of hundred microns. Referring back to Figure 5, the effect of critical length cross-over means that in the cross-over region II device, circuit and CAD strategies may be effective in mitigating the effect of interconnect dominance. However, once past this region, **dynamic interconnect**, this is,



**Figure 6.** Circuit model using programmable interconnect between functional blocks

interconnect with logic would be the preferred choice.

While technologically inconceivable today, such incorporation of reprogrammable logic gates would be consistent with the earlier observation that the cost of local decision making on cycle time would be significantly less than the cost of sending information leading to increased logic level per cycle. Since the logical complexity per cycle is related to system microarchitectures, most immediate use of inter-connect buffering gates would be in reprogrammable interconnects to improve versatility of the hardware blocks (perhaps sold as core elements by semiconductor vendors<sup>9</sup>).

In the face of these technological changes, the challenge for the system architect is to find ways that future architectures can effectively exploit the opportunity presented by the dynamic interconnect. In<sup>16</sup> the authors present a machine architecture that uses relatively small amounts of reconfigurable logic embedded in the CPU-memory interconnection fabric to build architectural assists (such as prefetch, gather/scatter hardware). These assists have been shown to improve the system performance by several orders of magnitude on a range of applications by adapting to application needs for memory latency and bandwidth. We believe that this would be increasingly an attractive way of exploiting an interconnected-dominated process technology. Further, as the marginal cost of logic decreases well below the cost of additional interconnect, there is no fundamental reason why reconfigurable logic could not be used in building data-paths that respond to application needs by providing, for instance, multiple variable precision functional units.

#### 4. CONCLUSIONS

Our preliminary analysis of technology trends indicates that the conventional wisdom – that reprogrammability in high performance processor designs is expensive – is already beginning to be challenged. As gate switching continues to scale well below 100 ps range, local decision making would cost significantly less than the cost of sending information.

Coupled with the advances in semiconductor technology, the advances in CAD tools and algorithms are beginning to have an impact on how designs are done today. With the emphasis on system models in hardware description languages (HDLs) such as Verilog and VHDL, the process of hardware design is increasingly a language-level activity, supported by compilation and synthesis tools.<sup>8</sup> These tools are beginning to support a variety of design constraints, on performance, size, power, and even the pin-outs. Locking I/O-maps to ensure that physical design remains unchanged while logical connections are modified based on applications will soon be a common feature to allow programmable logic to be embedded in key modules of a system and provide on-line programmability to change hardware functionality. Tools for distributed hardware control synthesis to allow dynamic binding of hardware resources,<sup>6</sup> and synthesis of protocols to low latency hardware<sup>1,4,5</sup> have been successfully demonstrated. With these CAD and synthesis capabilities, embedded programmable logic can be inserted into the key parts of systems, and used to alter behavior dramatically with modest performance overhead.

Such changes will pave the way for a new class of system architectures that can exploit flexibility to deliver robust, high performance to applications. Indeed this has been the basis of most of current work on customizable computing machines.<sup>2,3</sup>

#### Acknowledgments

The work was sponsored by support from National Science Foundation CAREER award number MIP 95-01615, NSF/DARPA ASC-96-34947 and from DARPA DABT63-98-C-0045.

## REFERENCES

1. BORRIELLO, G., AND KATZ, R. Synthesis and Optimization of Interface Transducer Logic. In *Proceedings of the IEEE International Conference on Computer-Aided Design* (Nov. 1987), pp. 274–277.
2. BUELL, D. A., ARNOLD, J. M., AND KLEINFELDER, W. J. *Splash 2: FPGA in a Custom Computing Machine*. IEEE Computer Society Press, 1996.
3. CHIEN, A. A., AND GUPTA, R. K. MORPH: A System Architecture for Robust High Performance Using Customization. In *Proceedings of the The Sixth Symposium on The Frontiers of Massively Parallel Computation (Frontiers'96)* (Oct. 1996), pp. 336–345.
4. CHOU, P., ORTEGA, R., AND BORRIELLO, G. Synthesis of the Hardware/Software Interface in Microcontroller-Based Systems. In *Proceedings of the IEEE International Conference on Computer-Aided Design* (Santa Clara, Nov. 1992), pp. 488–495.
5. CHUNG, K.-S., GUPTA, R. K., AND LIU, C. L. An algorithm for synthesis of system-level interface circuits. In *Proceedings of the IEEE International Conference on Computer-Aided Design* (Nov. 1996).
6. COELHO, C., AND MICHELI, G. D. Dynamic scheduling and synchronization synthesis of concurrent digital systems under system-level constraints. In *Proceedings of the IEEE International Conference on Computer-Aided Design* (Nov. 1994), pp. 175–181.
7. CONG, J., PAN, Z., HE, L., KOH, C.-K., AND KHOO, K.-Y. Interconnect Design for Deep Submicron ICs. In *Proceedings of the IEEE International Conference on Computer-Aided Design* (Nov. 1997), pp. 478–485.
8. GUPTA, R. K., AND LIAO, S. Y. Using a Programming Language for Digital System Design. *IEEE Design and Test of Computers* (Apr. 1997).
9. GUPTA, R. K., AND ZORIAN, Y. Introducing Core-Based System Design. *IEEE Design and Test of Computers* (Oct. 1997).
10. HENNESSY, J. L., AND PATTERSON, D. A. *Computer Architecture: A Quantitative Approach*. Morgan-Kaufman, 1990.
11. HU, C. Low-voltage cmos device scaling. In *Proceedings of the IEEE International Solid State Circuits Conference* (1994).
12. KANG, S. Metal-metal matrix for high-speed mos vlsi layout. *IEEE Transactions on CAD* (Sept. 1987), 886–891.
13. RABAHEY, J. M. *Digital Integrated Circuits*. Prentice-Hall, 1996, ch. 8.
14. SATAPATHY, R., AND GUPTA, R. Analysis of Semiconductor Technology Scaling and Its Impact on On-Chip System Designs. Technical memo, University of California, Irvine, 1998.
15. SORKIN, G. Asymptotically perfect trivial global routing: A stochastic analysis. *IEEE Transactions on CAD* (1987).
16. ZHANG, X., DASDAN, A., SCHULZ, M., GUPTA, R., AND CHIEN, A. Architectural Adaptation for Application-Specific Locality Optimization. In *Proceedings of the International Conference on Computer Design* (Oct. 1997).