

# Hard IP Group Design

Howard Sachs, Michael Barry, John Campbell  
Telairity Semiconductor, Inc.  
3375 Scott Blvd. Suite 300  
Santa Clara, CA 95054  
Email: hsachs@telairity.com

## Abstract

*Today there is significant evidence in the marketplace that there is a surprising 2x loss in performance for most new ASIC designs using normal static CMOS logic. Many have attributed this surprise to non-scaling of interconnects in the DSM range, causing signal integrity issues. But the technology has been scaling which indicates the problem lies somewhere else. The reason for this loss is that the average wire length is getting longer as the number of devices on the chip increases. A new approach will be described that uses generic hard IP blocks to design any type of digital system that overcomes these losses to achieve a 2x improvement in performance over the current methods.*

## 1. Introduction

Today most ASIC designs are very large, in the range of 500K to 20M gates. This level of complexity has caused individuals on the design teams to specialize in specific aspects of the design process. As time has progressed the notion of “The Tall Thin Engineer” has almost disappeared. Instead we now have specialists in Logic Design, Simulation/Verification, Pre Synthesis Floor Planning, and Physical Design. As a result the RTL Designer or Logic Designer really does “Logical Design” since he/she is not usually involved with the physical part of the design process. The expectation is that the RTL is independent of any physical parameters and that synthesis and backend tools will make the appropriate transformations to the logical design to create a good physical implementation. In fact there is a virtual wall between the logical design and physical design.

How important is the understanding of the technology to the actual logical design of a system? Looking at history may provide some insight.

From 1960 to 1980 the understanding of the physical implementation was considered very critical to the design process. Physical partitioning of the design was a key step in the design process. This was an exercise requiring an understanding of the critical paths and the physical distances between the partitions because the interconnect wiring was a significant contribution to the overall delay. This physical partitioning usually became the same as the logical partitioning and specifications were written that described the timing requirements for each partition. Today this is equivalent to what is now called the floor plan.

From 1980 to 1990 CMOS was introduced and became the choice for most designs. Density, adequate speed, high yields, and the resulting low cost made this the technology

of choice. ASSP designs were usually implemented with custom gates or standard cells. ASIC designs were implemented with Gate Arrays. Typically the Gate Array designs were much slower than the Cell Based designs because of the non-optimal nature of the building blocks, long wiring channels and the conservatism built into the models for yield purposes. The Standard Cell based designs were much smaller because of floor planning was used. And as a result the designs were significantly faster because the wiring was significantly shorter and the drivers were matched to the interconnect wiring and loads. During this same period Verilog became the de facto standard for net list design and was used for both Gate Arrays and Standard Cell based designs. Logic synthesis was introduced during this period with the ability to create up to 50K gates; later on this number was increased to 200K gates. This made a significant improvement in productivity for both types of designs, typically a factor of two or three, and with very little to no downside.

During the period from 1990 to 2000 CMOS speed continued to improve by 50% and density continued to improve according to Moore’s law at 2x every generation. By the end of the decade Gate Arrays were almost totally replaced with Standard Cell based designs in Europe and the United States because of the poor performance and high cost of the Gate Arrays. Wiring length was beginning to be more important but gate delays were still dominant because of the over designed Gate Arrays. The synthesis and backend tools produced adequate results for Standard Cells compared with Gate Arrays. Crosstalk was present in the standard cell designs but not noticed because the performance was so much better than Gate Arrays.

Starting in the year 2000 with 0.18um technologies new problems emerged. When we moved from 0.25um Standard Cells to 0.18um we expected a 50% increase in performance, which we did not get because of signal integrity issues. The actual wire lengths increased over the expected scaling because designers were synthesizing much larger blocks than before. When larger blocks are synthesized the amount of floorplaning done is reduced, causing average wire lengths to increase. These larger blocks also make the work of the placement tool more difficult, resulting in longer average interconnect wires as well. Longer wires also allow the crosstalk effect to negatively influence the speed of the circuit, causing surprises. Also, proper matching of transistor drive strengths to the loads (wire and gate loads) have not been very accurate. Both the longer wires, a result of less floorplaning, and inadequate transistor sizing have caused most designs to lose at least a factor of 2 in performance.

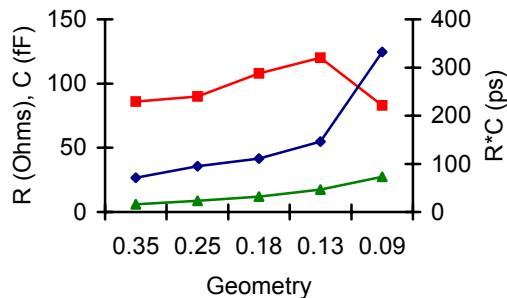
Floorplaning and synthesizing the design into smaller pieces, less than 10,000 gates in each partition, will keep the wires reasonably close to the optimal length, otherwise the wiring could increase by as much as a factor of four longer than optimal.

## 2. Wires in More Detail

### Scaling

Figure 1 shows the RC delay for a fixed 500um length wire in different geometries. This chart shows the familiar continual increase in RC delay shown in many presentations. Note how this chart shows a significant problem when going from 0.13um to 0.09um. Fortunately for ASIC designers, this chart does not tell the true story because the interconnect wires should be scaling at 70% along with the technology.

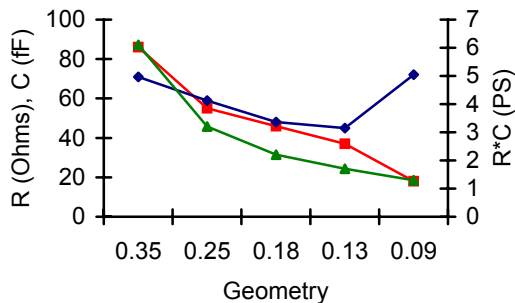
Figure 1 RC Delay with 500-um Fixed-Length Wires



—■— Total Capacitance —▲— M3 RC Delay  
—◆— M3 Resistance

Figure 2 shows the simple RC delay as a function of geometry for a wire that is 500um in 0.35um and scaled at each generation by the metal scaling rules. It is clear that the RC is not getting worse; in fact it has been getting better. All of the figures in this paper assume copper and low k dielectric for geometries at 0.13um and lower. The RC delay may go up slightly in the future but we do not have enough data at this time to accurately show the trend.

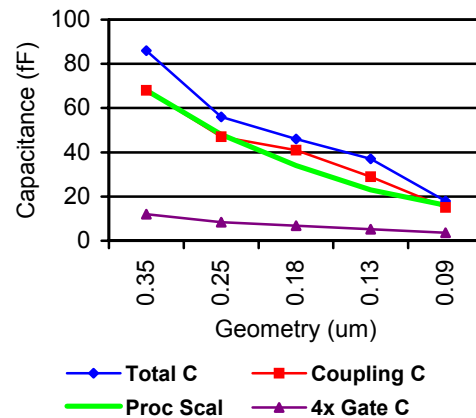
Figure 2 RC Delay with 500-um Wires Scaled



—■— Total Capacitance —◆— M3 Resistance  
—▲— M3 RC Delay

Figure 3 shows that the total wiring capacitance and crosstalk coupling capacitance is scaling as expected. The same design done in a 0.35um technology and a 0.09um technology will scale accordingly. If the wire lengths and capacitances are scaling, then the delays and any effects of crosstalk should be in the same ratio and therefore present no new surprises. Ho [2] also suggests that wire lengths, crosstalk, and wire delays are scaling with the transistors. It is obvious from Figure 3 that if the technology is scaling and yet there are excessive signal delays, then the wires must be longer.

Figure 3 Process Scaling



### Wire Lengths

Chang, Cong, & Xie [6] suggest that wire lengths are from 1.6 to 2.5 times longer than optimal because of non optimal placement algorithms. This data was taken from synthetic benchmarks that have known wire lengths. This data also indicates that there is a significant increase (up to 25%) in wire length when the number of modules or gates is increased by ten times.

Horowitz [3] argues that the major concern now, is that we have more long wires to deal with because the designs are larger. The following are summary quotes from his presentation.

“Communication on chip is no longer free”

“Back to the future – it looks like board/box design”

The back to the future comment is emphasizing the need for engineering the wires the way it was done years ago and not to let these wires get out of control. A typical wireload model also shows this expectation quite clearly. A set of typical standard cell wireload models for a standard 0.18u technology is shown in Table 1.

Table 1 Wireload Model for a Fanout of Four

Gates (K)	Capacitance (fF)	Wire Length (um)
10	5.2	26
20	13.1	66
40	17.0	85
80	19.0	95
160	20.0	100

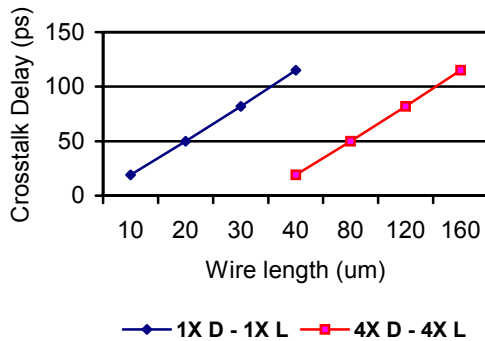
This model suggests that the average wire length increases from 26um to 100um when the number of gates in the same partition to be synthesized grows from 10K gates to 160K gates. Our experience shows that the wire lengths can vary from 5um to over 200um for small designs in the range of 3000 gates. This can result in significant RTL rework of the design after the physical placement and routing indicates this variation in the loading.

Dally & Chang [1] have shown that wire lengths for automated standard cells for a data path design can be much greater than the custom wire lengths. They show metal-2 wire lengths 34.9X longer and metal-3 longer by 7.92X than in a custom design. In this example much of the additional wiring is due to the replacement of the synthesized register files. The seven-port register file in the design was built using ordinary flip-flops and multiplexers, which is quite common, since custom register files are expensive to design or purchase. This additional wire length can cause additional rise time delays and crosstalk delays. Of course, if the cell library has sufficient control of the drive strengths, delays of the longer wires can be overcome by increasing the drive strengths of the gates.

#### Gate Sizing and Drive Strengths

One solution to longer wires is to match the drive strength of the driver to the wire load and gate load in the design. Figure 4 shows that a 4X gate will drive a 4X load with 160um of wire (equivalent to a 4X load) at the same speed as a 1X driver with an equivalent wire and gate load with the same total delay. (Note that in this paper a 1x gate is made with minimum size transistors, contrary to most cell libraries, where a 1x is really 4 times the minimum size allowed.) Therefore the delays for a net can be managed to the optimal delay with proper sizing of the driving gate to the loads.

Figure 4 Wire Crosstalk Delay



Many standard cell libraries have only a few different gate sizes and corresponding drive strengths, making the speed of the net (the rise and fall time) either too fast or too slow. In addition some standard cell libraries have multiple stages inside the cell to maintain a 1X fan in. This makes life very simple when wire loads change; all that is required is to replace the cell with one that has larger buffers inside. The problem is that the delay is no longer constant because of the intrinsic delay of the two additional inverters. Most designers use small complex gates and inverters with more

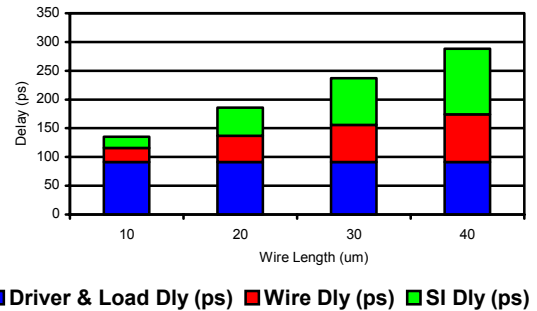
power to drive large loads. This can optimize the net for speed as well as area. The approach we recommend and have used in our designs have 16 different power levels for inverters to optimally drive the nets. In addition we have proprietary software for automatic sizing of the gates to the loads in our designs.

#### Crosstalk

Crosstalk has recently become an issue because the wires appear to no longer scale with the technology, but in fact this is a result of poor placement and/or no floorplaning and not the technology. As a result the amount of crosstalk and delay in the net can be increased significantly.

Figure 5 shows for a given driver and receiver the delay is constant. The wire delay, actually the rise time delay, increases, as the wire gets longer because of the added capacitance of the wire. This chart shows the added signal integrity delay due to aggressors switching in the opposite direction from the victim. For each case we simulated using the same 1X driver and receiver for the victim, with a 200-ps rise or fall time. The aggressors are driven with an ideal voltage source with a 200-ps rise and fall time. With longer wires the crosstalk coupling capacitance ratio to the total capacitance increases thereby causing the signal rise time to be further delayed. Increasing the size of the victim's driver can significantly reduce the signal integrity delay, since the equivalent driving resistance is reduced allowing the victim to recover faster. This is illustrated in Figure 4.

Figure 5 Components of Delay



### 3. A Solution to Improve Performance

The solution is a large number of small reusable IP blocks, called groups that can be used over and over and are kept in hardened form to guarantee the performance, area and power of the IP. The groups are designed with custom slices or standard cells to achieve optimal performance, area and power. In order for this solution to be economically viable these groups must be reusable to a very high degree and port between processes automatically; otherwise the added engineering costs outweigh the advantages. The focus for this solution is as follows:

- Custom-like performance for groups
- Wiring is engineered and minimized early in the design
- Transistor drive strength is optimally engineered
- Global wires are isolated from local group wires

- Automated group portability from Fab-to-Fab and process-to-process

Design re-use is a very alluring methodology because of its inherent simplicity, high productivity, and repeatability. If one looks at cell libraries, the re-use factor is always 100% and is the standard method for assembling logic. Each cell typically has between 1 to 4 gates. The other end of the spectrum is block level re-use with 50,000 to 200,000 gates, but this has a very poor re-use factor, less than 33%. Choosing the proper building block size is critical to this approach.

We have examined many designs and determined there is a flat 95% re-use curve from 4 gates to about 3,000 gates. After that the re-use factor starts to drop off rapidly. This reinforced our notion of a group with an optimal group size from 300 gates up to a maximum of 3,000 gates. The average “Group” will have ~1,000 gates.

We have identified a natural partition of any design into four fundamental types of logic: Data path, Control, I/O, and Memory and have demonstrated that any digital circuit can be built using these building block groups. Characteristics of the groups are described below;

- Control

Control structures are typically PLA-like structures and state machines that contain RAM’s or ROM’s.

- Data Path

The data path is the implementation of algorithms and is only concerned with how data flows. Typical groups are adders, shifters, multiplier parts, etc.

- I/O

General I/O interfaces such as UARTS, USB, SDRAM interface typically have low implementation or licensing cost and hence are usually soft cores that would not be built out of groups.

- Memory

Memories such as register files, FIFO’s, & CAM’s are key components of this technology. Moderate size SRAM’s for caches are used. DRAM’s are not addressed since they are readily available from a number of sources.

Table 2 shows a few examples of different types of groups. It is interesting to note that the range in average wire length for groups vary in length by over a factor of four. The group with the smallest wire lengths, 25um, is a 2 input 32-bit adder (Ad2\_32). The group with the longest wires, 121um, is the Perm\_80, which is an eighty-bit permuter that is primarily composed of multiplexers. The average wire length does not include the wiring within slices or standard cells. Therefore the average wire length would be somewhat shorter if we included all of the very small interconnect wires. The Ad32\_32 agrees quite closely with most small size wire load model tables with a fan out of 4 at 25um. The wide variation in actual average wire length, as shown in table 2, assuming a fan out of 4 demonstrates why assuming an average wire length that is dependent only on the number of gates being driven can lead to significantly over loaded or under loaded nets.

Table 2

Group Name	Ave Length of Wire (um)	Total number of Equivalent gates
Buf_32x20_r2w1	111	2265
Iseq_8	42	2473
Ad2_32	25	873
Adsb3_80	29	3029
Perm_80	121	654
Bth_mux_32	39	2719

We have done an analysis of the critical path on a hand crafted 32-bit adder group with 15 stage delays (flip flop not included). The average wire length for the complete adder is 25um. The average wire length in the critical path however is 43um with a range from 5um to 227um. This variation is so great that a normal wireload model would be dramatically in error on most of the stages causing most of the nets to be either over driven or under driven. The driving stage can vary from a 1x in the 5um case all the way to an 8x in the case of 227um of wire length. Of course, if the gates were sized up or down in each case to optimally drive the wires the delays would be optimal. As described earlier some cell libraries use multiple stages to power up the load driving capability. This makes for easy insertion, no downstream effect, but usually has three stages and the resulting delay is not the minimum. Sutherland [4] describes this problem in detail on page 20 of the reference.

Additional delays can be attributed to crosstalk in the wiring. This added delay due to crosstalk could be as large as 0.7 times the normal delay, using equal rise and fall times of 200ps. The total delay for the 32-bit adder without crosstalk is 1635ps. So the impact of crosstalk could be zero or as small as 10 to 20% to as much as 70% of the stage delay. Figure 5 shows that the heavily loaded gate could be delayed as much as 114 ps because of crosstalk. In the 32-bit adder case if this was the only stage affected there would be a 6.9% overall delay in the path. Note that most clock cycles have 15 stages of logic, and the crosstalk typically only affects one or two stages out of the fifteen. For example, the 32-bit adder referenced above averaged 81ps delay per gate and 28ps average wire length delay. Even doubling the wire delay by crosstalk for four stages of the fifteen would result in a delay, which would be quite small:

$$(4 \times 28\text{ps}) / ((81\text{ps} + 28\text{ps}) \times 15) \text{ stages} = 6.8\%$$

So, typically we would expect that large delays due to crosstalk would be primarily concentrated in long global wires where the global wire is a large percentage of the cycle time.

#### 4. Demonstration chip

We have designed a demonstration chip, which includes a SIMD FFT [5] to illustrate the use of our group methodology in a typical design. This circuit was designed to run at 400MHz in a UMC 0.18um standard logic process with worst-case temperature and voltage and nominal

process. The chip is expected to be back from the factory in early March with actual speed results available in mid-March. HSIM [7] was used to determine the speed of the design dynamically using functional vectors. Using the adder reference above with an average delay of 109ps per stage and a total of 15 stages between flip-flops as a typical group, the total path delay would be 1635ps. Adding a flip-flop stage at the output adds another 400 ps for a total of 2035 ps, which is 491 MHz.

Table 3 shows the mean and average wire length for the entire FFT design. The mean length of the M2/M3 wires in the adder group was 15um. The mean global wire length was 125um. The mean wire length is more meaningful since there are always long wires in a design but many of the long wires are not in the critical path. The limiting path in the FFT design was the booth encoder and carry save adder and that path was 2439ps limiting the design to 410MHz – which is 2 to 3x faster than a typical synthesized design.

Table 3 Wire Lengths for the FFT

	Ave wire length (um)	Mean wire length (um)
<b>M2/M3</b>	<b>51</b>	<b>15</b>
<b>M4/M5</b>	<b>284</b>	<b>125</b>

Table 4 shows the metal utilization for each metal layer in the design. Metal layers 1, 3, and 5 are the most utilized since these wires run in the direction of the data path. The cell library was designed using 14 tracks, which is much wider than most generic standard cell libraries. Most generic libraries are in the 8 to 10 track range to increase the reported cell density. The lack of porosity however causes more pressure on the routing resources. Our intention is to make sure that all of our groups can each be exclusively routed on the first three metal layers, insuring that no global wires would ever be routed through the groups. Metal-4 and metal-5 are global wires and are very under utilized. This means that as the designs get larger there will be adequate routing resources without increasing the wire lengths unnecessarily.

Table 4 Metal utilization for the FFT

	Percent Routing Utilization
<b>Metal 1</b>	<b>55.0</b>
<b>Metal 2</b>	<b>23.6</b>
<b>Metal 3</b>	<b>43.4</b>
<b>Metal 4</b>	<b>6.4</b>
<b>Metal 5</b>	<b>20.1</b>

## 5. Conclusion

We have shown that the technology has been scaling quite nicely, contrary to popular belief. Recent ASIC designs have not run at the expected frequencies because the wiring on the chip has not scaled accordingly. These longer-than expected wires are the result of two factors. The first is that designs are getting more complex at each generation, which causes placement tool problems and results in longer

wires, and the second is that the methodologies used do not focus on engineering the wires. We have shown a new approach based on pre-designed hard IP building blocks that are optimized for speed that can improve the performance of a typical ASIC chip by 2 to 3 times.

## 6. Acknowledgements

The authors would like to thank the following engineers for their efforts in providing data for this paper. Richard Dickson, Luigi Di Gregorio, Joe Varghese, Sarita Thakar.

## References

- [1] W. Dally and A. Chang, “The Role of Custom Design in ASIC Chips”, *DAC* 2000
- [2] R. Ho, K. Mai, M. Horowitz, “Wires: A users Guide” *SRC/Marco Workshop at CIS, Stanford, May 1999*
- [3] M. Horowitz, R. Ho, K. Mai “Wires: A users guide” *PDF Internet*, no date
- [4] I. Sutherland, B. Sproul, D. Harris “Logical Effort: Designing fast CMOS Circuits”
- [5] S. Arya, “Designing High Speed DSPs” *ISPC* 2003
- [6] C. Chang, J. Cong, M. Xie, “Optimality and Scalability Study of Existing Placement Algorithms”, *ASP-DAC Conference Japan*
- [7] HSIM is a product of Nassda Corporation.