

Physical Synthesis Methodology for High Performance Microprocessors

Yiu-Hing Chan
IBM Server Group
Poughkeepsie, NY

Prabhakar Kudva Lisa Lacey Greg Northrop
IBM TJ Watson Research Center
Yorktown Heights, NY

Thomas Rosser
IBM Server Group
Austin, TX

ABSTRACT

Integrated logic synthesis and physical design (physical synthesis) continues to play a very important role in high performance microprocessor design methodologies. In this paper, we present the integrated physical synthesis timing closure methodology used in the current generation microprocessors. Physical synthesis techniques were aggressively used as part of logic and placement optimizations for performance, power and area. The design turn around times were significantly reduced and timing convergence was consistently achieved.

Categories and Subject Descriptors

B.7 [Integrated Circuits]: Design Styles—Design Aids

General Terms

Algorithms, Performance, Design

Keywords

Microprocessors, High-Performance, Synthesis

1. INTRODUCTION

With technology scaling as well as the increasing complexity of modern microprocessors, interconnect effects play an increasingly important role in determining design methodologies. The design of the POWER4 [15] and of newer generations of other IBM high performance microprocessors [1] required considerable integration between synthesis and physical design to manage interconnect effects. The first generation of chips that used the physical synthesis methodology described in this paper, were fabricated in a 0.18- μm CMOS 8S3 SOI (silicon-on-insulator) technology with seven levels of copper wiring [11]. This version of the chip [15] had a clock frequency of greater than 1.3 GHz with a transistor count of 174 million. Physical synthesis continues to be used in other IBM microprocessors as well [1]. Achieving timing closure on these chips continues to take significant innovations in physical synthesis techniques.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2003, June 2–6, Anaheim, California, USA.

Copyright 2003 ACM 1-58113-688-9/03/0006 ...\$5.00.

Microprocessor design entails very tight constraints on frequency, power, noise and packing density. Design closure and minimizing the effects of interconnect in the microprocessors while meeting tight performance constraints were accomplished by two main approaches: hierarchical design and physical synthesis with innovative techniques.

This paper describes a physical synthesis methodology used in the design of microprocessors in a hierarchical design environment. Section 2 discusses a hierarchical approach for managing physical design, noise and timing issues while achieving chip level design closure. The benefits of semi-custom library design and how physical synthesis benefits from such libraries is described in Section 3. In Section 4, the physical synthesis methodology to accomplish macro level timing closure is described. Some post physical synthesis optimizations performed are described in Section 5. Section 6 presents some of the representative results that were obtained as part of the timing closure of the microprocessors.

2. OVERVIEW OF METHODOLOGY

The physical synthesis tool [4] is capable of optimizing designs in both a hierarchical and flat chip design environments. In this paper, we will primarily focus on the hierarchical methodology as applied to microprocessor design.

The processor physical design makes extensive use of hierarchy, partitioning the chip into cores, cores into units and units into macros. For example, in one of the microprocessors, the 2 cores are identical. Each core is partitioned into 8 units (Instruction Unit, Floating Point Unit, etc). Each unit is partitioned into floorplannable macros which are either custom and semi-custom dataflow macros or synthesized control random logic macros (RLMs). There were 24 RLMs and 28 dataflow macros in the Instruction unit of the processor. RLMs in the rest of this paper will specifically mean control logic macros. A majority of the RLMs used physical synthesis. One microprocessor had 523 Random Logic macros (RLMs) which used physical synthesis.

2.1 Hierarchical Timing Optimization

A simplified timing and physical design flow is shown in Figure 1 and a floorplan is shown in Figure 2. Three levels of timing and physical resource optimizations are performed: the chip level, the unit level and macro level.

Feeding Timing Assertions Down the Hierarchy

Timing assertions are fed forward to lower levels of the hierarchy after the higher level is partitioned. Timing assertions of macros and unit inputs consist of arrival times with phase tags and slew. Macro and unit outputs have required arrival times with phase tags and output capacitance loads. The

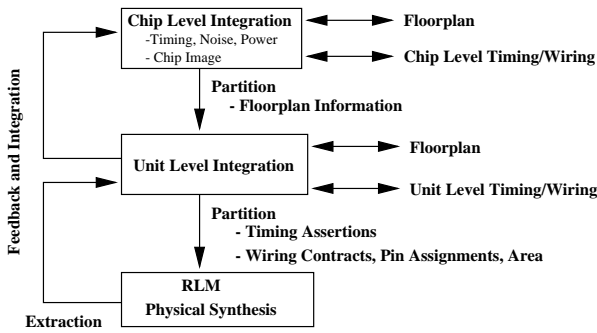


Figure 1: Hierarchical PD and Timing

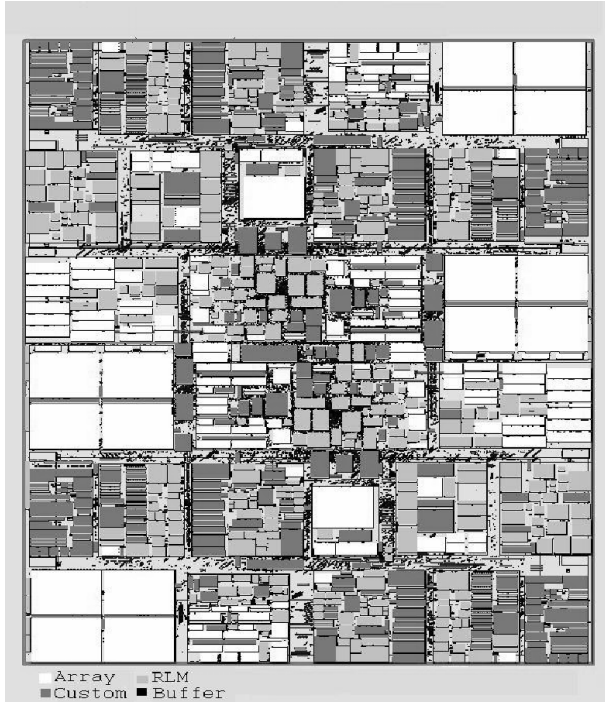


Figure 2: Floorplan of a Typical Processor

core/chip level timing run generates unit timing assertions. The unit timing run generates macro timing assertions.

For dataflow macros, assertions are used for timing analysis using a transistor level timing tool. Dataflow macros are implemented with hand designed schematic. The schematic design is followed by either manual transistor level and cell level physical design or a mix of manual and place and route physical designs to achieve high packing density and optimal implementation for high frequency.

For RLMs, macro assertions are used for timing optimization by physical synthesis. RLMs are optimized through synthesis [12] and physical synthesis [4] tools.

Feeding Timing and Noise Abstracts Up the Hierarchy

As timing closure on each design progresses, information of the design status are fed to upper levels of the hierarchy from the lower levels of the hierarchy. Macro timing abstracts are used for this purpose at the unit level. Macro timing abstracts contain timing equations for all input-to-latch, input-to-output and latch-to-output paths in the macro and

are generated through transistor level timing analysis.

At each macro level, noise analysis is also performed [7]. This provides noise characteristics for the internal circuits within the macro. It also provides a noise abstract with noise rules that is used at the unit level and then at the chip level.

At the early stage of the design cycle, macro timing and noise abstracts are generated from schematic netlists, which are generated by synthesis from VHDL. Early runs are based on estimated statistical wire loads between books based on fanout for very quick turn around time. As the design cycle progresses, the abstracts are derived from physically synthesized macros. Timing optimizations and noise avoidance transformations and assertions updates are made on a regular basis as the design cycle progresses.

RML Timing Closure and Noise Avoidance

As control logic becomes more stable, RLMs are taken through physical synthesis [4, 6] for optimal placement and for timing, area, noise and power optimization. This is followed by routing with automatic router, verifying RLM wirability, pin placement optimization and post PD RC netlist extraction to generate a more accurate macro timing abstract. Noise avoidance is achieved through careful library cell selection, wire planning and decoupling capacitors.

Top Level Timing Closure

While physical synthesis is used at the RML level, timing optimization at higher levels of the hierarchy (unit and chip) can use a variety of optimizations such as: buffer insertion for nets with long wire length that exceeded slew target or for isolation of non-critical loads; pin optimization to minimize extra wire length due to pin misalignment between driving and receiving macros; usage of low resistance wires and or wider width wires to reduce wire RC delay; re-synthesizing macros and re-running unit timing in a particular order to force synthesis to work on current top paths based on updated assertions; logic re-structuring in macros for optimal synthesis results; logic re-partitioning and latch point optimization to minimize RLM crossing, wire RC delay and or removing unnecessary buffers; movement of macros to minimize wire lengths between macros; and usage of circuit tuner on RML level for slack reduction. Timing optimizations throughout the hierarchy are done concurrently to speed up the overall timing closure. Noise analysis and optimization at the top level is achieved through the use of macro level noise abstracts and extracted data for the global wiring.

2.2 Hierarchical Physical Resource Allocation

Concurrently with the timing data, physical information also has to be fed forward and backward through various levels of hierarchy. In the hierarchical flow, at each level, global wires (between macros and units) are optimized for timing with each iteration. Hierarchical design allows for concurrent design at all levels of the hierarchy. Since wiring resources are shared across the hierarchy, they are managed through top down wiring contracts. At the early stage of the design cycle, the wiring contract can be very simple. For example at the unit level, no M1 tracks, every 4th M2 tracks and every other M3 tracks are assigned. The remaining tracks belong to macro for macro level wiring. As the wiring on the unit level becomes stable, the wiring contract

can represent the completely/partially unused tracks on the unit level that are available to macro level wiring. Each level’s floorplan and wiring are done with the lower level’s floorplannable object’s image and top-down contracts from levels above. Pins and floorplannable objects placement in each level of the hierarchy (dataflow macros, unit and chip) are done with iterations to minimize wire lengths and improve wirability throughout the hierarchy.

The floorplans at each level are done carefully and with iterations to minimize wire lengths between timing critical units early on in the design cycle. Timing critical nets between units are analyzed with a circuit simulator to guide the usage of metal level with wider width and with lower wire resistance.

3. SEMI-CUSTOM PHYSICAL SYNTHESIS

The benefits of a semi-custom library can be best realized during physical synthesis. There are several ways in which physical synthesis helps to take full advantage of semi-custom cell libraries. Matching capacitive loads (including wire loads) with the driving cells selected from a large and varied semi-custom library provide a significant advantage in timing closure, since for a given gain value, wire loads can be much better matched from a wide selection gate sizes.

3.1 Library Design

The combinational logic portion of the standard cell library was predominantly simple (inverting) static CMOS gates. The complexity was limited to a maximum fan-in of 4, with series nfet stacks limited to 4 (nand4) and pfet to 3 (nor3). Though limited to a small number of types, a large number of sizes were produced for each type. The cells were generated by a tool that assumed a basic parameterization of each type, with the nfet tree scaled by a parameter NW and the pfet tree scaled by PW. The tool is capable of producing physical design for a broad and continuous range in this parameter space. Cells were generated for a matrix of points based on gain and beta ratio, where increasing the NW and PW together gives cells with fixed gain for various loads, while changing the PW changes the beta ratio of the gates. A set of PW/NW ratios (beta ratios) for each type is determined. The number of beta ratios varies from 1 to 4, depending on the utilization of the cell. Beta ratios were selected with both performance and noise margin considerations.

In addition to the basic set of primitive cells, two additional degrees of freedom were added. The first was tapering, wherein the input pin delay is deliberately skewed, improving one pin at the cost of additional load and delay at other pins. This provides additional timing performance on paths with significantly different arrival times. The second set was a complete duplicate set with a low threshold device (lowVt) substituted for all fets. The lowVt gates offered about 10% improved performance at the cost of significantly higher static leakage. Appropriate use of tapered and lowVt cells has demonstrated up to 15-16% performance improvement.

The use of special optimizations such as lowvt cells, tapered cells and beta ratios are only meaningful during physical synthesis when timing, load and slew estimates are precise. For example, the limited use of lowvt cells to improve performance on critical paths while controlling leakage can only be determined after placement.

Table 1: Standard Cell Library

Gate	Num Sizes (input cap)	beta ratios	Vt	tap	total
inverter	25	4	2	0	200
nand2	16	4	2	64	192
nand3	13	3	2	52	130
nand4	10	2	2	0	40
nor2	11	3	2	44	110
nor3	9	2	2	18	54
aoi21	10	3	3	20	80
aoi22	10	3	2	0	60
oai21	10	3	2	20	80
oai22	10	3	2	0	60
xor2	5	5	2	0	50
xnor2	5	5	2	0	50

3.2 Gain Based Optimization

With the use of semi-custom libraries with large sets of power levels, beta ratios and tapering, gain based synthesis [14, 2, 10, 8, 13] is extensively used in the design process for a significant portion of the control logic of the processor. In gain based synthesis, library analysis is first performed on the libraries to create gain models for each cell type such as nand2s, nand3s etc. The libraries used in the design of these microprocessors were tailored for gain based modelling.

These models created by library analysis are consistently used in all steps of synthesis such as technology independent synthesis, technology mapping and timing correction. Each cell in the netlist has an associated gain based model, i.e, each cell is size independent. After synthesis within the gain based domain, the netlist is discretized [8] back to the standard cells in the library whose timing is based on load based polynomial timing models. Further timing optimizations to remove effects of model inaccuracies are performed after discretization.

4. PHYSICAL SYNTHESIS

Once the physical image and the timing assertions for the macros are obtained, placement and synthesis are performed concurrently on the RLMs using the physical synthesis tool.

4.1 Physical Synthesis Flow

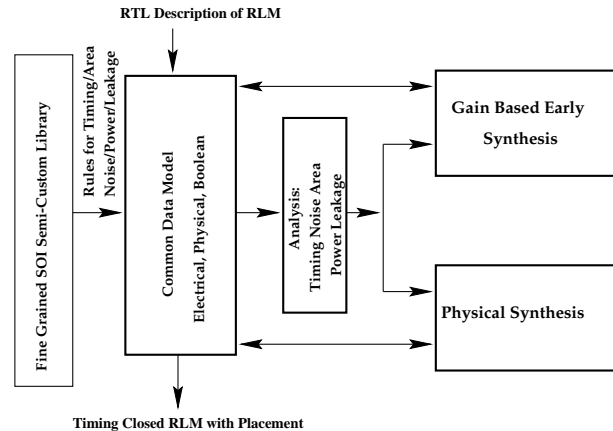


Figure 3: Physical Synthesis Flow

Physical synthesis can be performed on a technology mapped netlist where placement and synthesis proceed concurrently to perform timing, area and power optimizations. Additionally, a placed and optimized netlist may need further incremental optimization. This flow is supported as well. The physical synthesis flow for RLMs are shown in Figure 3.

All transformations use a common database where the boolean (functional representations), electrical (timing, noise) and physical (physical locations, congestion) characteristics can be evaluated and modified concurrently. Noise, timing, area and power analysis tools work on the same database and provide input to the physical synthesis algorithms which manipulate the netlist in an incremental fashion.

4.2 Wire Length Calculation

A key requirement for physical synthesis is the prediction of wire lengths based on physical locations. A Steiner tree model is used to measure wiring distances and capacitive loads are computed using these lengths. A Steiner tree model is reasonable for physical synthesis in a hierarchical design methodology where dimensions of each RLM are small.

Figure 4 shows that the tracking between the net-length of a Steiner tree and the final routing of the net. It shows the number of nets that have a certain percentage prediction error. The three data sets shown (left to right) were obtained by successively removing the shortest 10% and 20% of nets from the statistics. One can see that all larger error percentages disappear if the shortest nets are removed. The error due to short nets do not have a significant effect on delay. For the longer nets the precision of a Steiner length approximation is sufficient for the transformations that are done in the physical synthesis phase. This is especially true, if this Steiner tree is also being used to initialize the global router. In highly congested designs, the inaccuracies increase because of scenic routes. Such inaccuracies are first avoided by attempting to perform congestion control optimizations within physical synthesis. If congested areas and scenic routes persist, a variety of optimizations such as in-footprint post routing gate sizing, adding additional wiring contracts have proved helpful.

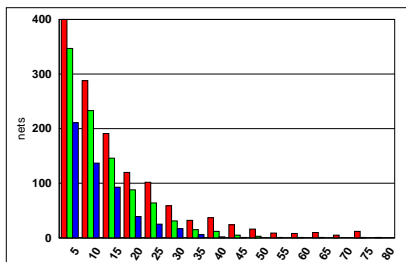


Figure 4: Wire load histogram

For short wires, an Elmore delay model is used. The wire load capacitances are estimated as lumped capacitances proportional to the Steiner estimates of the lengths of the wires. For longer wires where the RC component is significant, an appropriate delay model is chosen.

These models are registered as net-delay calculators in an incremental timing analysis engine [5]. Both changes to positions of cells and changes to the netlist may trigger incremental recalculations of the timing and Steiner trees.

4.3 Typical Physical Synthesis Optimizations

The physical synthesis approach opens up a wide range of optimizations where changing logic, electrical and physical characteristics of the design lead to significant benefits. A few examples of the kinds of transforms that were used are presented in this section.

Clock Optimization for Low Power

An algorithm that works with the last few stages of clock logic within macros has been developed. The logic is driven by a global mesh that distributes a synchronous clock to the processor in a tightly controlled manner. Within the macros, that mesh clock must be split into multiple clocks and delivered to the latches with very tight constraints. In addition to the mesh clock, there are a vast array of control signals to assist in power savings, set up LSSD scan mode, select between single and multiple cycle clocks, and to interface synchronous and asynchronous clocks.

Clock power can be as significant as 70% of the power in a processor. A large amount of this power is consumed in the last stages of clock driver (splitters) circuits which drive the latches. Typical placement algorithms tend to spread the latches far away from the splitters. Even taking locality into account, the introduction of the splitters results in large wire lengths, contributing to increased capacitance driven by the splitters and therefore power consumption. This is shown in Figure 5. Power optimization is possible by moving the latches closer to the splitters. Arbitrarily moving latches during placement can result in timing as well as wiring penalties. Efficient physical synthesis algorithms allow concurrent clock block optimization along with timing optimizations (to compensate for lengthening of the data wires) as well as congestion control.

When this clock optimization process is combined with physical synthesis, it is possible to balance the needs of the data flow into and out of latches with the tight constraints on the clocks themselves. The clocks are ignored during the early global placement until the locations of the latches starts to become obvious. At that point, an initial clock network is created and optimized based on the current latch placement. For the next placement step, the latches are intentionally drawn closer to the clock driver logic and then the optimization is performed again. For all remaining placement steps, the clock nets are constrained to not grow much. The fact that this all happens within the physical synthesis flow means that the logic can be optimized to reflect the significant placement changes that occur with the insertion of a lot of very large clock buffers or splitters.

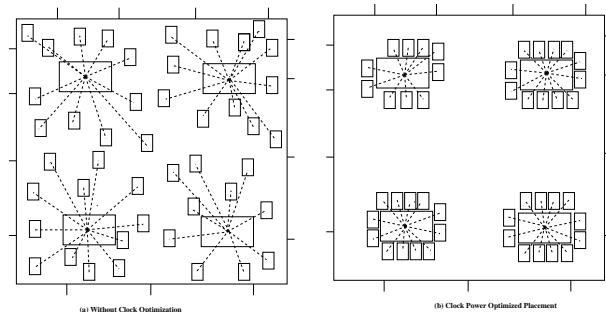


Figure 5: Before (a) and after (b) Clock Optimization

The algorithm is based on an annealing process that allows moves that are not legal in the early stages. In the later stages, only legal moves are accepted. Annealing was chosen as the base technique since the algorithm has various parameters that can be set for its use by very different processor teams for very different clocking structures. The rules that need to be honored are: maximum distance between clock logic and latches, maximum fanout of any particular form of clock logic, maximum capacitance for each clock logic pin, average distance between clock logic and latches, maximum slews for clock nets and control nets, maximum RC seen by a sink pin of a clock net, support of multiple power levels and net termination to reach capacitance target. In order to satisfy some or all of the constraints as required, an annealing approach was the best fit. Clock logic with a high ratio of latches to clock blocks are produced.

Circuit Relocation

Placement algorithms optimize the physical location of circuits to optimize total wire lengths. The use of netweights improves the timing characteristics of the placed design but does not always produce placements that are optimal for timing. During physical synthesis, locations of select timing critical circuits should be changed for timing purposes. Such relocations need to be performed carefully since moving a particular circuit in one direction may increase the wirelengths of its fanins, thus creating other critical paths. Examples of circuits locations for timing are shown in Figures 6 and 7. In the first case gates on a meandering path are moved to reduce the net capacitances driven by the gates and therefore improving timing. In the second case, buffer insertion on the gate driving the long wire is avoided by redistributing the distances between the logic gates.

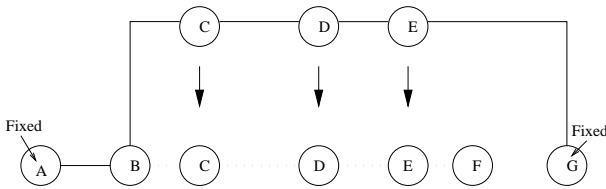


Figure 6: Relocation for Timing 1

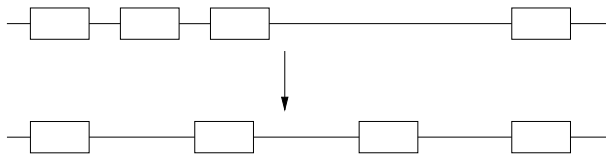


Figure 7: Relocation for Timing 2

Circuit Remapping

Technology mapping is performed early in synthesis when actual wire loads are not available. In the design there are often many instances of suboptimal configurations of technology mapped gates. For example in Figure 8 an XNOR followed by an inverter is used to drive a long wire. The inverter is placed close to the XNOR output so as to drive most of the wire load. Alternately a NAND decomposition can also be considered. Depending on the locations of the fanin gates, the length of the wire, the choice of the configuration may vary.

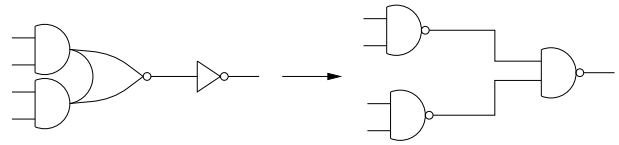


Figure 8: Remapping of Circuits

Buffer insertion

Buffer insertion prior to placement can result in poor topological trees causing both wiring congestion as well as degradation in performance. Due to the total wire length objective of placement algorithms, buffer tree placement configurations such as those shown in Figure 9(a) occur frequently. Rebuilding buffer trees during physical synthesis can significantly improve the overall quality of the results. These techniques are based on the method presented in [10].

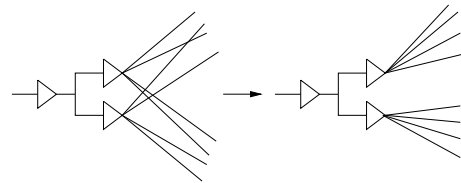


Figure 9: Buffer Reinsertion

Placement driven buffer insertion was extensively used to improve performance as well as reduce wiring congestion.

Early mode padding

Performing early mode padding prior to physical synthesis can result in excessive padding. Often the wire loads add to the delays along certain paths, making them less critical in the early mode. Therefore early mode padding is performed only during physical synthesis in our methodology. A timer which performs both early mode and late mode timing analysis concurrently is used to prevent some paths from becoming critical due to the early mode padding.

5. POST PHYSICAL SYNTHESIS

After the macros have gone through physical synthesis for optimized books placement with late and early mode timing optimization, the macros go through routing. For very large macros with over 60K transistors, routing is done with a net priority list that prioritizes the order of the route according to the slack of the nets in the RLM. By doing that, the topology of timing critical nets and capacitance loads are guaranteed to be as close as possible to Steiner estimates that have been used during placement/timing optimization phase within physical synthesis.

A few of the macros are not wirable with the initial wiring contracts. These contracts have to be updated with more wiring resources from unit level to make the RLM wirable. In some cases that the RLM pin placement affected the wirability. The RLM pins had to be re-arranged and spread farther apart to enable the RLM to be wired.

After routing is completed, RC netlist extraction is done for post PD timing analysis. There are cases because of multiple constraints used during placement/timing optimization phases within physical synthesis that some of the stages driving primary outputs (POs) with heavy external loads are

placed farther away from the PO pins than expected. This resulted in high wire RC delay and longer path delay on these PO paths. These output stages are manually moved and placed closer to the PO pins to reduce wire RC delay due to heavy external capacitance load.

There are cases where the slew of some of the nets exceed the target. Either manual repowering or circuit tuner [3], is used to tune the design to meet the slew target and improve the slack. For macros with network changes due to late logic updates or timing optimization after routing is completed, ECO (engineering change option) that preserves the initial placement and retains most of the routes is used to do a delta place and route on any of the updated and affected instances and nets. This way the original RLM timing and slack on other paths are preserved.

6. PHYSICAL SYNTHESIS EVALUATION

The methods described earlier have been used extensively in high end microprocessor designs. Figure 10 shows the slack and area improvements of some of the processor macros run using the physical synthesis method compared to the previous methodology which alternated between placement and synthesis. All positive numbers indicate an improvement in our methodology. Points in upper half with positive slack delta show slack improvements. Points with negative area delta show area improvements. A majority of the points fall in the region that shows both area and slack improvements. The methodology emphasized slack improvement, although in most cases both slack and area improvements were obtained. In a few cases very small area penalties were incurred for significant slack improvements.

In some cases there was a degradation in wirability and therefore slack due to a variety of factors such as inherent complexity of the RTL specification and early synthesis optimizations. The relationship between graph structure and wirability [9] needs further investigation. It is important to have techniques where wiring effects are analyzed early in the design flow.

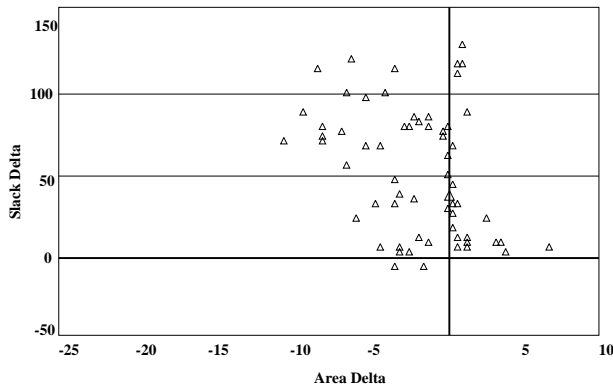


Figure 10: Slack and Area Results

7. SUMMARY

The paper presents a comprehensive physical synthesis methodology and techniques that have been used in the design of high speed microprocessors. The extensive use of physical synthesis has demonstrated the importance of integrating placement and synthesis tightly. Our experience has shown the need for further research in predicting and

optimizing for interconnect and physical design effects much earlier in the design flow.

8. ACKNOWLEDGEMENTS

The authors would like thank all members of the PDS team as well as the the Microprocessor PDS team including: Bob Hatch, David Kung, Andrew Sullivan, Lakshmi Reddy, Michael Kazda, Michael Bowen, Rainer Clemen, Marianne Knirsch, Allan Dansky.

9. REFERENCES

- [1] AVERILL, R. M. ET AL.. Chip integration methodology for the ibm S/390 G5 and G6 custom microprocessors. *IBM Journal of Research and Development* 43, 5 (1999), 681–707.
- [2] BEEFTINK, F. ET AL. Combinational cell design for CMOS libraries. *Integration* 29, 1 (2000), 67–93.
- [3] CONN, A. R. ET AL.. Gradient-based optimization of custom circuits using a static-timing formulation. In *Proc. ACM/IEEE Design Automation Conference* (June 1999), IEEE Computer Society Press.
- [4] DONATH, W., KUDVA, P., STOK, L., VILLARRUBIA, P., REDDY, L., SULLIVAN, A., AND CHAKRABORTY, K. Transformational placement and synthesis. In *Design Automation and Test in Europe (DATE)* (Mar. 2000).
- [5] HATHAWAY, D., ABATO, R., DRUMM, A., AND VAN GINNEKEN, L. Incremental timing analysis. Tech. rep., 1996. IBM, U.S. patent 5,508,937.
- [6] HOJAT, S., AND VILLARUBIA, P. An integrated placement and synthesis approach for timing closure of PowerPC microprocessors. *Proc. International Conf. Computer Design (ICCD)* (1997), 206–210.
- [7] K.L.SHEPARD, V.NARAYANAN, AND R.ROSE. Harmony: Static noise analysis of deep submicron digital integrated circuits. *IEEE Transactions on Computer-Aided Design* 18, 8 (1999), 1132–1150.
- [8] KUDVA, P. Continuous optimizations in synthesis: The discretization problem. In *International Workshop in Logic Synthesis* (1998), pp. 188–191.
- [9] KUDVA, P., DOUGHERTY, W., AND SULLIVAN, A. Metrics for structural logic synthesis. In *International Conference on Computer Aided Design (ICCAD)* (Nov. 2002), IEEE Computer Society Press.
- [10] KUNG, D. A fast fanout optimization algorithm. In *Proc. ACM/IEEE Design Automation Conference* (June 1998), pp. 352–355.
- [11] LEOBANDUNG, E. ET AL.. High performance 0.18 mm SOI CMOS technology. In *IEEE IEDM Technical Digest* (1999), IEEE Computer Society Press, pp. 679–682.
- [12] STOK, L. ET AL. Booleadozer: Logic synthesis for ASICs. *IBM Journal of Research and Development* 40, 4 (2001), 407–430.
- [13] STOK, L., SULLIVAN, A., AND IYER, M. Wavefront technology mapping. In *Design Automation and Test in Europe (DATE)* (Mar. 1999).
- [14] SUTHERLAND, I. E., AND R.F.SPROULL. Theory of logical effort: Designing for speed on the back of an envelope. In *Advanced Research in VLSI: Proceedings of the 1991 University of California Santa Cruz Conference, C. Sequim Ed. The MIT Press* (1991).
- [15] WARNOCK, J. D. ET AL. The circuit and physical design for the power4 microprocessor. *IBM Journal of Research and Development* 46, 1 (2002), 27–53.