

Exploring Regular Fabrics to Optimize the Performance-Cost Trade-Off[†]

L. Pileggi, H. Schmit, A.J. Strojwas, P. Gopalakrishnan, V. Kheterpal, A. Koorapaty, C. Patel, V. Rovner, K.Y. Tong

Carnegie Mellon University, Department of Electrical and Computer Engineering
5000 Forbes Avenue, Pittsburgh, Pennsylvania, 15213

ABSTRACT

While advances in semiconductor technologies have pushed achievable scale and performance to phenomenal limits for ICs, nanoscale physical realities dictate IC production based on what we can afford. We believe that IC design and manufacturing can be made more affordable, and reliable, by removing some design and implementation flexibility and enforcing new forms of design regularity. This paper discusses some of the trade-offs to consider for determination of how much regularity a particular IC or application can afford. A Via Patterned Gate Array is proposed as one such example that trades performance for cost by way of new forms of design regularity.

Categories and Subject Descriptors

B.7.1 [Hardware]: Integrated Circuits – *Gate arrays, Advanced technologies, VLSI (very large scale integration).*

General Terms

Performance, Design, Economics, Reliability.

Keywords

Integrated Circuits, Regularity, Cost, Performance.

I. INTRODUCTION

The phenomenal progress of IC manufacturing that has been evidenced by Moore's Law has created a pattern of pushing IC performance to technology limits to justify the cost of new emerging technologies. However, the history of this IC evolution has shown us that what we *can* build in a next generation technology is continually outpacing what we *can afford* to build in that technology. This situation is commonly referred to as the *design productivity gap*.

This gap, as depicted in Fig. 1, can be interpreted in a number of ways. One perspective is that design productivity is not improving at a fast enough rate. Which, given that innate human intellect cannot be expected to improve over time, suggests that EDA (electronic design automation) tools are not keeping pace with technologies and associated IC complexities. But is it reasonable to expect EDA technologies to keep pace with IC technologies that are pushed to the extremes of their performance and integration scale?

[†] This work was supported in part by the MARCO/DARPA Gigascale Silicon Research Center (GSRC) and the MARCO/DARPA Center for Circuits, Systems and Software, C2S2.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

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

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

As CMOS scales to finer feature sizes, and especially toward nanoscale, the complexity of what is technically feasible for integration grows exponentially, while physical details that must be managed and modeled grow increasingly complex as well. The number of ways in which a chip can, hence will fail, increases dramatically. Failures occur not only due to manufacturing defects and reliability faults, but as we push to higher performance, parametric (noise, delay, etc.) failures become increasingly problematic.

The increase in possibilities of what *can go wrong* not only increases the IC cost due to lower yield, but more importantly drives the *cost of design* to astronomical limits in an attempt to produce any yield at all. The corresponding design costs and expanded time-to-market schedules for application specific ICs (ASICs) have led to more products being designed using programmable and/or configurable standard products, such as FPGAs (field programmable gate arrays). These solutions offer extremely low NREs (non-recurring engineering costs), but at a high price in terms of performance, power and die area (Fig. 2).

Instead of avoiding application-specific IC customization completely, an alternative may be to exploit some of the untapped technology displayed in Fig. 1. For example, create ICs with some of the regularity and structure of standard ICs, but still offer some application-specific customization. Since cost is the limiting constraint for ASICs in general, our objective would be to trade off some performance and design flexibility for a simpler design flow and a shorter time-to-market for an ASIC. Moreover, given that microprocessors design teams are growing in size at a Moore's Law rate, some amount of regularity and structure to streamline the design flow will soon be warranted for these designs as well.

The cost of regularity and structure is in terms of IC area and performance primarily. So *how much regularity can we afford* for a particular application domain? It is important to note that a larger die may not necessarily be a more expensive die, since regularity can potentially improve the manufacturing and parametric yields[1]. Moreover, as we enter sub-100nm technologies, the need for fault tolerance and redundancy may become important, and die sizes will become larger, but potentially more affordable. It is conceivable that more regularity and structure could be used to improve the design robustness and fault tolerance.

As one example of regular logic that can be used to improve manufacturability, yield, and design robustness in comparison to

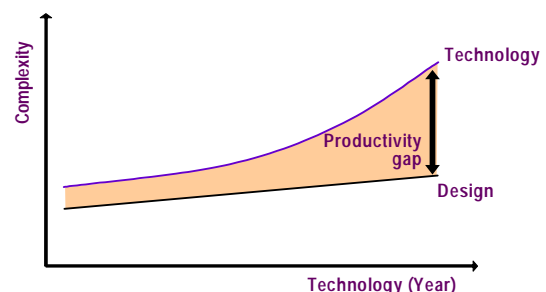


Fig. 1: Depiction of the design productivity gap.

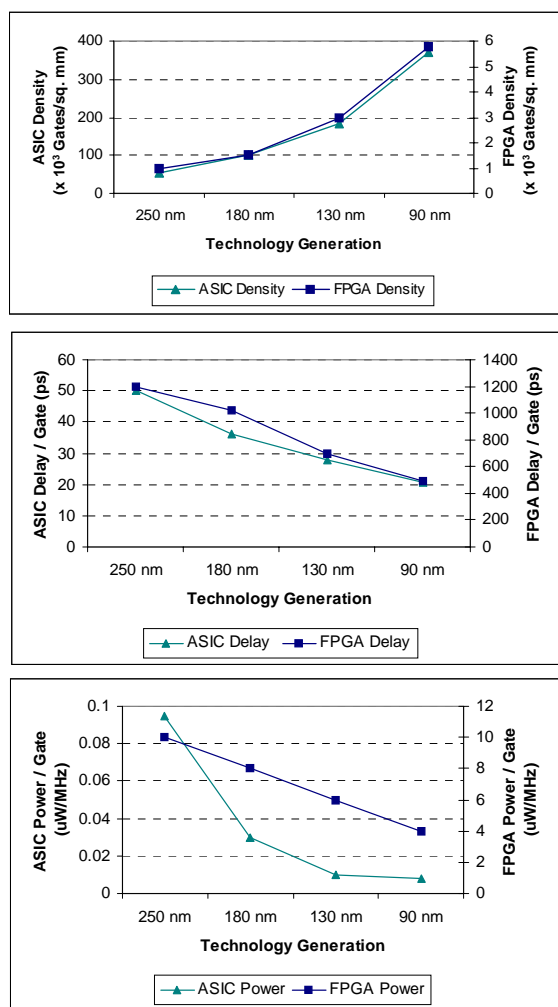


Fig. 2: Comparison of FPGA vs. ASIC from [2].

standard cells, we describe in Section IV a Via Patterned Gate Array (VPGA)[3]. A VPGA is formed by regular geometry, logic and routing layer structures that are customized for an application using via layers. In this paper we compare VPGA-like regularity with standard cells for several small logic block design implementations to consider the trade-offs of performance vs. cost. Most importantly, it is our conjecture that VPAs and other regular logic structures will ultimately provide better performance predictability such that some or all of the performance penalty due to regularity will be recovered by more accurately driven system level optimization.

VPAs, just like gate arrays in general, however, do not address the implementation of memory blocks, data paths, CPU cores, and analog components. There are two strategies to deal with this issue. First, it would be possible to approach the problem like an FPGA, and integrate an allocation of memory and other blocks such that the mixture of components matches the needs of the broadest possible market. The second strategy requires the construction of application-domain specific IC (ADSIC) implementation platforms, which contain those components most relevant to a particular family of applications. ADSICs would be partially optimized based on the particular choice of customized blocks that best fit a domain of applications, which would once again be an optimization of trade-offs between performance and cost.

The remainder of this paper begins with an outline of some of the manufacturability challenges in subwavelength lithography that can be alleviated by imposing more geometrical regularity. Section

III then describes other forms of design regularity that are equally important, followed by a description of our VPGA example in Section IV. We present some preliminary results in Section V regarding how well VPGA addresses the performance portion of the trade-off exploration, and follow with some conclusions and proposed future directions in Section VI.

II. PRINTABILITY CHALLENGES AND PROCESS VARIATIONS

Due to the overwhelming complexity of large nanoscale ICs and the corresponding fabrication process, to achieve acceptable levels of performance and yield requires both circuit and layout design that is carefully tailored to be robust with respect to the unavoidable process variations. Faithful reproduction of the IC layout shapes has been especially difficult to achieve in the sub-wavelength lithography for which minimum feature sizes are below one half of the illumination wavelength. Layout printability challenges have become extremely severe due to: 1) high NA (numerical aperture), off-axis illumination schemes (angular, quadrupole, dipole) and small depth of focus; and 2) large mask error enhancement factor (MEEF).

As a result, critical dimensions (CD), i.e. layer line widths, vary substantially as a function of layout density and neighborhood. This results in significant differences between dense and isolated lines. To minimize these differences, Optical Proximity Correction (OPC) techniques have been employed. But even with these techniques, it is impossible to optimize layout printability for all pitches, and some intermediate pitches result in large CD errors. These errors create what are referred to as the forbidden pitches, which are already a significant problem in the 130nm and 90 nm nodes, and may become show-stoppers at the 65nm node and below. Moreover, some patterns become increasingly difficult to print, such as isolated metal islands or line-ends in various configurations.

Layout printability is also strongly influenced by the etch and Chemical Mechanical Polishing (CMP) effects which depend on the intra-layer layout density variations as well as the underlying topography. The CMP challenges become very pronounced for Cu BEOL (back end of line) processing. To deal with these printability challenges, today's solutions are primarily based on the Resolution Enhancement Techniques (RET) such as: 1) Phase Shift Mask (PSM) lithography which helps with the printability of the densest features in the most critical layers and also reduces the mask error factor (MEEF); and 2) OPC techniques which reduce CD variations in various layout patterns. However, performing these corrections is computationally difficult since simple rule-based methods are no longer applicable, and simulation models of the lithography process are required. This approach is called model-based OPC. Of particular difficulty are corrections for line-ends as a function of layout neighborhood.

Therefore, to perform layout correction properly, larger neighborhoods must be considered and the model-based OPC becomes overwhelmingly complex for huge ICs with arbitrary layout patterns. Moreover, in Alternating Apertures (AA) PSM techniques, phase conflict resolution becomes prohibitive for huge chips with arbitrary layout patterns. Compounding this problem, other effects (such as etch and CMP) aggravate the situation.

The resulting printability variations are evidenced in terms of both functional yield loss as well as by parametric failures. For example, large width patterns at minimum spacing can result in intra-layer shorts, while the significant line end shortening may result in insufficient coverage of the via or contact hole by the metal and thus cause an open circuit. Examples of parametric failures include poly CD variations which will affect transistor performance, resulting in variations of gate strength, matching or clock skews. Incomplete coverage of the contact or via holes may also produce highly resistive contacts or vias, thereby producing soft failures.

With this expanding domain of manufacturing defects, the IC testing problem has become even more challenging since the traditional fault models used by the Automatic Test Pattern Generation (ATPG) are completely inadequate. Hence, it is extremely difficult to detect many of the failure modes which result in significant yield losses. Therefore, just like the design productivity gap, our testing capabilities are failing to keep pace with the advancing technologies.

More than just new optical correction and testing techniques are necessary to deal with these layout printability challenges. One class of solutions may be based on designing a significantly restricted set of layout design rules where all layout features are placed on grid, and forbidden pitches are eliminated and specific difficult-to-print layout patterns are disallowed. Such solutions are becoming necessary but extremely expensive to implement since all existing cell libraries and IP cores will have to be regenerated under more constraints to the designers. The problem becomes even more severe for the intermediate layer of interconnect levels (Metal 3 and above) where the design hierarchy is not valid.

To reduce the problem complexity and achieve the desired performance and yield objectives, new solutions must explore the concept of *regularity*[1]. However, this regularity must go beyond the design hierarchy based on small local layout patterns since the larger range interactions are critically important to guarantee the *predictable printability*. The regularity cannot be limited to the low level layers (such as poly and Metal 1) either, since the yield is significantly affected by the higher levels of interconnects (metal layers and vias/via stacks).

Hence, the ultimate solution would be based on full chip layout being assembled out of a set of patterns that are guaranteed to print for a given lithography, etch and CMP process windows. Strict rules for assembling layout out of these patterns to satisfy printability constraints can be developed to control the layout neighborhood and make sure that all the interactions are within allowable limits.

This solution would also allow for very predictable performance estimation since these guaranteed-to-print patterns can be pre-characterized very accurately. Although we have focused so far on layout (or geometrical) regularity, it is absolutely necessary to develop a new synthesis approach in which the concept of regularity starts from architecture and logic levels and spans both functional blocks and global interconnect layers.

III. REGULAR IMPLEMENTATION FABRICS

Due to the manufacturability and printability challenges, regular logic fabrics are generally the first designs that are migrated to a new technology. In the past it was often memory designs that were used to tune a new fab line, but more recently FPGAs have been used for this purpose.

A. Geometrical Regularity

Due to their geometrical regularity, analysis and tuning of the masks for an FPGA can be performed over very small localized regions, such as the CLBs (configurable logic blocks), since this structure is repeated hundreds of thousands of times to construct the FPGA. This geometrical regularity can address several of the manufacturing challenges outlined in Section II. This is especially important for the silicon, polysilicon and lower-level-metal masks that are characterized by the finest pitches for the physical geometries. Even if the entire standard cell library is tuned for manufacturability, there is still the irregularity that occurs due to the abutment of all possible combinations of standard cells within a row or across rows for an adjacent column.

B. Logic and Routing Architecture Regularity

The advantages of FPGA regularity go beyond that obtained by the geometrical regularity. The regular logic cells and the regular

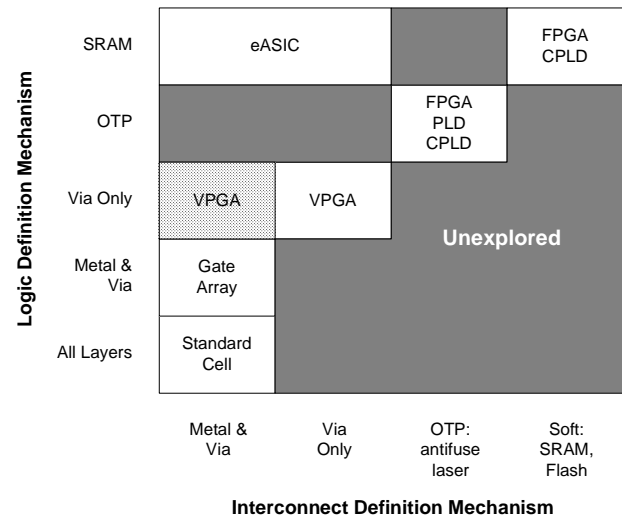


Fig. 3: Taxonomy of existing and yet-to-be-explored regular logic fabrics and arrays.

routing architecture which connects the cells greatly simplifies the performance predictability problem -- often referred to as the timing closure problem -- that plagues ASIC design. For standard cells in sub 130nm technologies, the gate delays are largely dominated by the load capacitance, which can be very dependent on the physical interconnect capacitance. For an FPGA, the logic cells, routing wires and buffering options are fixed, and therefore, much more predictable during the top-down design process. Of course it should be noted that this regularity “advantage” is obtained at a very high cost in terms of performance, area and power, as shown in Fig. 2. However, this predictability is not fully exploited with existing design flows to offset this penalty.

C. Exploring the Regularity Trade-offs

Given the alternatives of standard cells and FPGA for a particular product development or application area, the choice depends largely on expected volume, required time-to-market, and other cost and manpower constraints. While there has been a marked decrease in the number of new standard cell design starts, it is apparent from Fig. 2 that the price paid in terms of product performance, power and die size is clearly problematic for many applications. Therefore, we should consider: 1) adding more regularity to all ICs to make them more manufacturable and reliable; and 2) exploring the construction of more regular logic fabrics and architectures that lie in the trade-off space bounded by standard cell ASICs and FPGAs (and general purpose processors).

It is possible to taxonomize the possible space of regular fabrics based on the mechanism used to customize a regular fabric for a specific application. Furthermore, it is possible to use different mechanisms for specializing both interconnect and logic. We graphically depict this space in terms of several existing products in Fig. 3. Most of the devices lie along the diagonal of this matrix. The eASIC product [7] is unique in selecting SRAM programmable logic cells and either via or metal & via specialization for interconnect. Gate arrays have standard transistor and poly masks, and custom metal and via masks. VPGAs have via patterned logic, and either fully regular metal with just via patterning, or ASIC-style routed interconnect.

The examples in Fig. 3 all trade-off performance and area for reduced costs by way of reduced design steps, and reduced manufacturing steps -- including the number of application specific masks. It is important to note that one substantial advantage of FPGAs is that there are no product specific masks. As the costs of mask sets grow out of control (Fig. 4), the NREs for multiple ASIC design

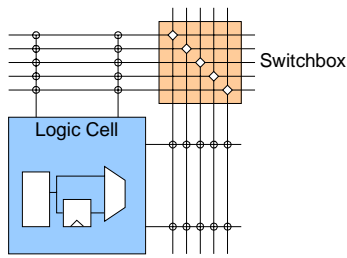


Fig. 5: A CLB and switchbox for an island style FPGA. Circles denote switch points that are created with CMOS switches and SRAM storage bits for an FPGA.

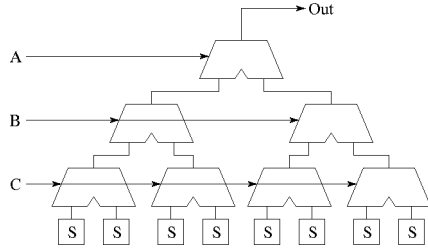


Fig. 6: FPGA 3-input LUT.

spins become unaffordable for all but the highest volume products. Therefore, part of the trade-off question must consider the total number of application-specific mask steps.

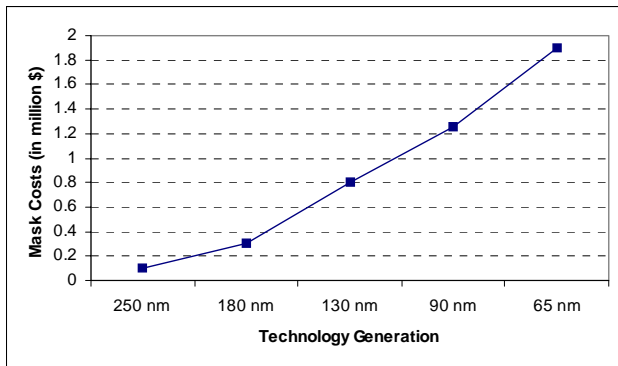


Fig. 4: Rising costs of a of CMOS standard cell mask set.

IV. VIA PATTERNED GATE-ARRAY (VPGA)

VPGA represents a compromise between FPGAs and standard cells by employing regular logic fabrics and interconnect structures, but customizing the routing and logic function using via mask patterns instead of field-programmable CMOS switches and SRAM storage bits. For example, consider the typical structure of the CLB and switchbox of an island-style FPGA (Fig. 5)[9]. Considering first the logic block, a simple replacement of the FPGA LUT (lookup table) in Fig. 6 would be the VPGA LUT (lookup table) in Fig. 7.

The VPGA regular logic cell should be significantly faster than that for the FPGA, since there is one less level in the LUT tree, and there is a via connection to one of the supplies, rather than a connection through a much more resistive SRAM cell. A transistor-level schematic of one version of a VPGA cell is shown in Fig. 8. Note that many other forms of regular logic LUTs are possible[4], but for this paper we have used a simple fully complementary structure.

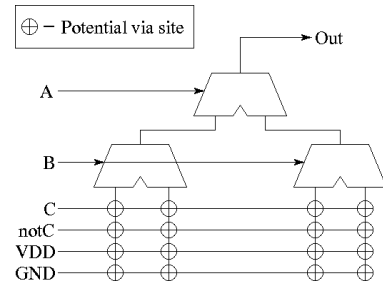


Fig. 7: VPGA 3-input LUT.

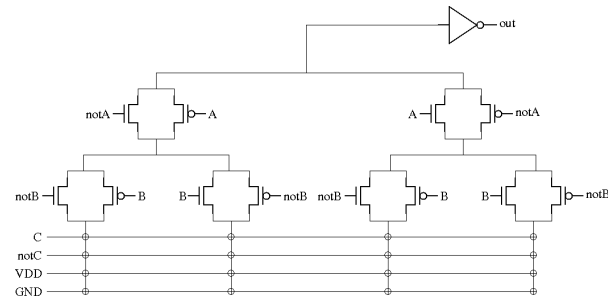


Fig. 8: A fully complementary 3-input VPGA LUT.

When optimized, we have found that this via-patternable 3-input LUT has excellent power-performance characteristics when compared to complex standard cell functions, such as XOR. For example, in the 0.13 μ m CMOS technology for which we base all of our experiments in this paper, the VPGA LUT in Fig. 8 had a 7% better delay and a 6% better energy consumption than a highly optimized XOR standard cell with a sizing selected to optimize the loading capacitance[4]. There was an area penalty of about 50% when we used potential via sites between metal 2 and metal 3. Higher layout density would be possible using potential vias on lower layers, but this would defeat some of the intent of regular lower level layers, since these are the most challenging to manufacture.

Compared to simple logic functions, however, such as a 3-input NAND, the VPGA LUT was substantially inferior in terms of delay, power and area. For example, using an experiment similar to the one used for XOR, the LUT implementation of NAND3 was 67% slower and consumed 25% more energy. Therefore, while most FPGAs are based on homogeneous CLBs comprised of identical LUTs, our VPGA CLB should be heterogeneous and include a combination of LUTs and simple logic functions. Using a fabric-specific technology mapping engine, we explored the possible combinations of CLB logic functions for a set of benchmark combinational circuit netlists[5]. The results suggested a CLB comprised of a via patternable LUT and two 3-input NANDs with via-patternable input signal inversion[5].

The layout for our heterogeneous CLB is shown in Fig. 9. To improve printability and manufacturability of this CLB, we have optimized its layout as follows[6]. Poly lines are placed at regular spacings and generous extensions beyond active layer are implemented to avoid line-end shortening. Also, poly linewidths on field oxide (actually Shallow Trench Isolation regions) are widened around n/p transitions. Poly patterns are designed to avoid phase conflicts if the AA PSM is to be employed. To reduce yield loss due to failing contacts or vias in CLB, contact via redundancy is imple-

mented and the metal island sizes are expanded beyond minimum allowable dimensions.

With a regular CLB, there are various options for overlaying a regular routing fabric, analogous to that for an FPGA, but with via patterning instead of pairs of tri-state buffers and SRAM storage bits. We explored some of the routing possibilities using a VPGA architecture and the VPR FPGA tool. Preliminary results were reported in [4]. One example of a VPGA switch-box fabric is shown in Fig. 10. The via-patterned switches allow routing in any direction without wasting a routing track. This figure shows a 4 by 4 array of routing switches. It is important to note, that since there are no active elements in these switches, the switchboxes can be constructed on top of the CLBs, in contrast to adjacent to the CLBs as required in an FPGA. Printability and hence manufacturability are improved by implementing regular metal linewidth/spacing patterns and providing for via borders at the bottom metal layer.

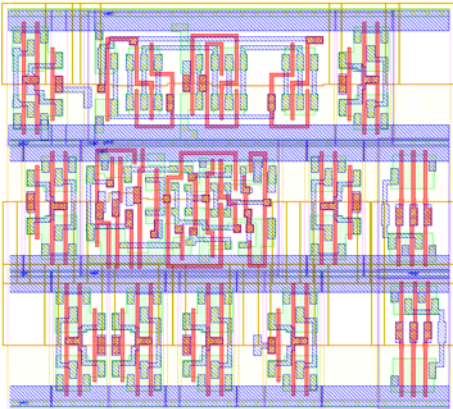


Fig. 9: Heterogeneous VPGA CLB layout comprised of a LUT, 2 input invertable NAND3's, 7 inverters, and 1 full-scan flip-flop.

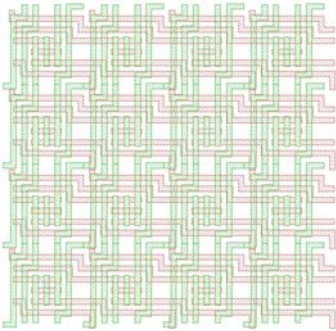


Fig. 10: One example of a VPGA switchbox routing fabric.

V. VPGA REGULARITY TRADE-OFF RESULTS

The efficacy and viability of a VPGA IC relies on the availability of enabling CAD tools and methodologies. For example, there are no ASIC or FPGA routers that can accommodate the routing fabric shown in Fig. 10. While we build some of the required tools, flows and methodologies, we can partially assess the regularity trade-offs for the proposed VPGA.

For example, we have already considered the area, power and performance trade-offs of VPGA cells versus ASIC standard cells, but how will they compare in terms of path delays? To assess two

of the regularity trade-offs, we compared a commercial 0.13 μ m CMOS standard cell methodology with two partial VPGA methodologies: the first of which treated each component of our CLB in Fig. 9 as a triple-height standard cell with the proper area, and applied ASIC-style routing. The second packed the VPGA standard cell type placement into physical regions to mimic a VPGA gate array. The gate array was routed with an ASIC-style router.

The VPGA cell library was characterized for timing and power using Silicon Metric's CellRater tool. For compatibility with existing commercial synthesis tools we created a library which contained all possible 3-input functions to represent all possible configurations and delays of the VPGA LUTs. Areas were represented by the proper corresponding function of the total CLB area.

Starting with several small RTL netlists, we applied the following three flows:

- Perform synthesis with Synopsys Design Compiler (DC) using the commercial standard cell library, and complete physical synthesis and design using Monterey Design's Dolphin tool to produce gds2.
- Perform synthesis with DC using a VPGA standard cell library comprised of all three input functions and fully invertable three input nands. Exploit the FPGA-like structure of the cells to perform compaction on the logic. Complete the physical synthesis and design using Dolphin to produce gds2.
- Perform synthesis with DC using the VPGA standard cell library comprised of all three input functions and fully invertable three input nands. Exploit the FPGA-like structure of the cells to perform compaction on the logic. Complete the physical synthesis placement using Dolphin to produce a coarse placement. Apply a simple packing algorithm to greedily pack cells based on slack and physical proximity. Complete the detailed routing using Dolphin to produce gds2.

We compared these three flows for four small logic block designs. All four designs were synthesized with a 0.5ns cycle to push the performance as aggressively as possible. The comparison is summarized in Table 1. Since worst-case path slack data is somewhat noisy, we show the average slack for the 10 worst-case paths. All designs used 0.13 μ m CMOS worst case libraries at 105 $^{\circ}$ C and 1.08V supplies.

	No. of	Av. Slack paths 1-10 (ns)			Area (sq. microns)		
	gates	flow a	flow b	flow c	flow a	flow b	flow c
ALU	651	-0.45	-0.30	-0.31	5600	7800	18225
DLX Controller	552	-1.05	-0.942	-0.961	5476	9216	16875
Firewire	4247	-1.31	-1.45	-1.77	27027	40944	56250
FPU	24640	-7.68	-7.81	-9.35	409600	562500	1103625

Table 1: Preliminary results for assessment of the VPGA regularity trade-offs.

It is apparent from these results that the performance of the regular logic is quite competitive with that based on standard cells. As expected, however, there is a substantial area penalty for the regular logic, especially for the designs with little or no flipflops (e.g. ALU) since there is an unused flipflop in every CLB with flow c. Even for the designs which contain flipflops, such as FPU, the ratio of LUTs to flipflops can greatly impact the area. For example, we designed a slightly larger CLB by adding one LUT to the configuration shown in Fig. 9, and implemented the FPU using flow c. The total area was reduced to 765000 sq. microns, and the average slack of the ten most critical paths was improved by 15% to -8.90ns.

Clearly, however, the ultimate assessment of regularity trade-offs can only be made once the regular logic fabrics have CAD tools and design flows as mature as those for standard cells. In particular, we know that our simple packing step for flow c substantially degrades the timing, and that there is much room for improvement. In

addition, there are opportunities for exploiting the regularity for even further improvement. For example, the simple logic compaction performed for flows b and c improves the area by approximately 20% by exploiting the FPGA-like structure of the logic fabric. While we expect further performance and area penalty using regular routing fabrics (e.g. Fig. 10), we believe that the routing regularity may bring additional predictability that can be further exploited as part of the synthesis and performance optimization process[10].

VI. FUTURE CONSIDERATIONS AND CONCLUSIONS

There are a lot of directions for this regular logic fabric work to go from here, in terms of circuits, architectures, CAD tools, and implementation methodologies. For example, one of our most important objectives is to exploit the regularity in order to achieve better top-down predictability for system level optimization. We believe that this predictability could ultimately compensate for some of the performance lost on the back-end of the flow due to the regularity and increased area.

There is also substantial work to be investigated to answer: how much regularity can we afford for an application? In addition, how regularity can help with testability, fault tolerance, and design robustness are also areas under investigation as well.

As we consider regular fabrics such as VPGA, however, we must concurrently consider how integrated systems would utilize such fabrics. If one objective is to derive an implementation based on a subset of manufacturing masks to control cost, then how would one configure an implementation platform that includes the VPGA fabric, memory, analog, clocking, etc., while providing sufficient flexibility for the application-domain customization. As part of our research we are exploring the development of affordable application-domain specific ICs (ADSICs) that are analogous to the FPGA and core platforms that are available today, but without attempting to accommodate all applications. Instead they are focused on implementation platforms for specific application domains, thereby providing additional performance versus cost trade-off scenarios.

VII. ACKNOWLEDGEMENTS

The authors would like to thank ST Microelectronics, and particularly, Davide Pandini, Michele Borgatti, and Pier Luigi Rolandi, for the useful discussions and the access to their 0.13 μ m CMOS technology. The authors would also like to thank Ruchir Puri and

John Cohn of IBM for their technical contributions to various aspects of this project.

REFERENCES

- [1] M. Palusinski, A. J. Strojwas and W. Maly, "Regularity in Physical Design", GSRC Workshop, Las Vegas, NV, June 17-18, 2001.
- [2] P.S. Zuchowski, C.B. Reynolds, R.J. Grupp, S.G. Davis, B. Cremen, B. Troxel, "A hybrid ASIC and FPGA architecture," Proc. of International Conference on Computer Aided Design, 2002, pp. 187-194.
- [3] L. Pileggi, H. Schmit, J. Shah, Y. Tong, C. Patel, V. Chandra, "A Via Patterned Gate Array (VPGA)," Technical Reports Series of the CMU Center for Silicon System Implementation, No. CSSI 02-15, Mar 2002.
- [4] K.Y. Tong, V. Kheterpal, V. Rovner, L. Pileggi, H. Schmit, R. Puri, "Regular Logic Fabrics for a Via Patterned Gate Array (VPGA)," submitted to CICC 2003.
- [5] A. Koorapaty, L. Pileggi, and H. Schmit, Heterogeneous Logic Block Architectures for Via-Patterned Programmable Fabrics, Submitted to Int'l Conf. on Field Programmable Logic and Applications, Sept. 2003.
- [6] S. Rovner, "Design for Manufacturability of Via Programmable Gate Array Fabrics," MS Thesis Report, Carnegie Mellon University, May 2003.
- [7] Z. Or-Bach, Z. Wurman, R. Zeman, L. Cooke, "Customizable and programmable cell array," US Patent 6,331,790, 18 Dec 2001.
- [8] C. Patel, A. Cozzie, H. Schmit, L. Pileggi, "An Architectural Exploration of Via Patterned Gate Arrays," Proc. of Int'l Symposium on Physical Design, 2003.
- [9] P. Chow et al, "The design of a SRAM-based field-programmable gate array – part II: circuit design and layout," IEEE Trans. on VLSI Systems, Vol. 7, No. 3, Sept 1999, pp. 321-330.
- [10] R.H.J.M. Otten and R.K. Brayton, "Planning for Performance", Proc. of Design Automation Conference, 1998, pp. 122-127.