PROFILING FPGA FLOOR-PLANNING EFFECTS ON TIMING CLOSURE

Jaren Lamprecht, Brad Hutchings

NSF Center for High-Performance Reconfigurable Computing (CHREC)
Department of Electrical and Computer Engineering, Brigham Young University
Provo, UT, 84602, USA
email: jtlamprecht@byu.edu, brad_hutchings@byu.edu

ABSTRACT

The impact of shape, area allocation and timing constraints on partitions was determined by selecting a standard set of submodules and performing over 1,000,000 place/route experiments. Place/route experiments used different area and timing constraints and their resulting trace reports provided timing results. These results suggest that the best results are obtained when about 20% additional area (above synthesis estimates) is allocated for each submodule. The aspect ratio of submodules is largely a non-issue (there was one exception in the data). In some cases, carefully constraining area dramatically improves results.

1. INTRODUCTION

As FPGA vendors continue to fabricate ever larger devices, design complexity grows, place and route times increase, and timing constraints become more difficult to meet. To alleviate these issues FPGA vendors are now recommending that customers floor-plan their designs [1]. An FPGA floor-plan is essentially a map created by the designer that assigns circuit submodules to specific physical locations on the FPGA fabric. Effective FPGA floor-plans are often difficult to create; the process is manual, iterative, and there is not a lot of design data available to help guide designers as they create floor-plans for their designs.

Xilinx provides some limited guidance to help designers choose submodules and assign them to physical areas on the fabric [1] [2] [3]. FPGA floor-planning case-studies do exist [4] [5] and demonstrate that area constraints can aid in achieving timing closure but these success stories mostly suggest that floor-planning is beneficial and provide little direction in general. Rather than examine a single design with a few floor-plan modifications, this work takes a different tack and focuses on the submodules themselves and how area and timing constraints ultimately affect their performance. With this data, designers should be able to create more effective floor-plans.

This work was supported by the I/UCRC Program of the National Science Foundation under Grant No. 0801876

The primary contribution of this paper is to provide the data that helps to answer the following questions that are of keen interest to any designer creating FPGA floor-plans for Xilinx devices.

- What aspect ratios are best for submodules that comprise the floor-plan?
- How much area should be allocated for a submodule?
- What impact do area constraints have on the maximum clock rate for a submodule?
- What guidelines should be followed when assigning submodules to physical locations on the FPGA?

This paper arrived at the answers to these questions by performing over *one million* place and route runs across several carefully chosen submodules for Xilinx devices. Each submodule was placed and routed approximately 200,000 times with various placement and timing constraints and the resulting timing reports help answer the questions above. Note that all experiments were performed on Xilinx devices and the data presented here only applies to Xilinx devices.

2. BACKGROUND

Floor-planning Xilinx FPGAs is performed by adding physical area constraints to a design, often in the User Constraint File (UCF) [6]. Constraints are applied in a UCF as shown in Fig. 1. Hierarchical logical divisions of a design can be constrained to reside within a particular region of the device. This region is defined by an AREA_GROUP specifying physical ranges of SLICEs, DSPs, and BRAMs. Different grids exist for each of type of resource, so different ranges are specified for each.

Selecting the range for each resource type requires some knowledge of the layout of the device. Xilinx FPGA devices use a heterogeneous island-style architecture. The low level architecture details remain hidden from the designer, but tools such as PlanAhead, FPGA Editor, and Xilinx Description Language (XDL) [7] provide high level layout views.

978-1-4673-2256-0/12/$31.00

```
#Clock Period Constraint
NET “clk” TNM_NET = “clk1”;
TIMESPEC “TS_clk1” = PERIOD “clk1” 2.8 ns HIGH 50 %;

#Area Constraints
INST "fft1024_x0/*" AREA_GROUP = "fft_group";
AREA_GROUP "fft_group" RANGE=SLICE_X24Y81:SLICE_X53Y160;
AREA_GROUP "fft_group" RANGE=DSP48_X2Y34:DSP48_X3Y65;
AREA_GROUP "fft_group" RANGE=RAMB36_X2Y17:RAMB36_X3Y32;
```

Fig. 1. Sample constraints applied to a design. The clock period is constrained to 2.8 ns. The fft1024_x0 component is constrained to an AREA_GROUP comprised of separate ranges for SLICEs, DSPs, and BRAMs.

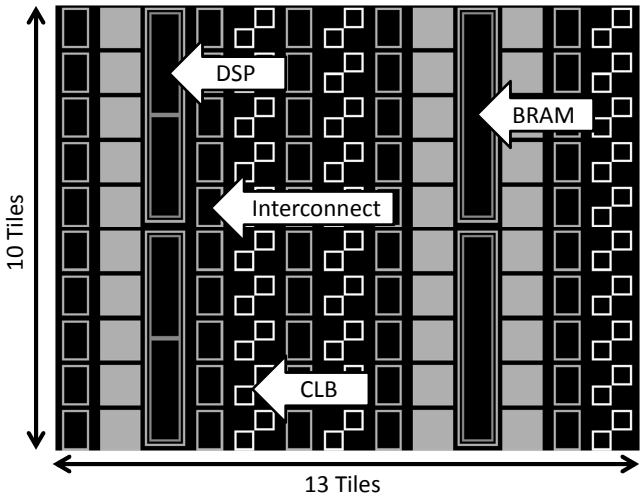

Fig. 2. A view from the RapidSmith device browser depicting a section of Xilinx tiles on the xc5vsx240t device. The different device resources all share the common tile grid.

Of the three, XDL is the lowest level view and provides the most information for choosing area constraints.

In the XDL device reports, all physical resources are assigned to tiles, and these tiles form a rectangular grid comprising the device. A section of these tiles is shown in Fig. 2. The universal tile grid resolves the different gridspaces which exist for SLICEs, DSPs, and BRAMs. For example, on the xc5vsx240t device, SLICE_X0Y0 and DSP48_X0Y0 reside at tiles (6,263) and (24,263), respectively. The universality of the tile gridspace allows it to function as a high level indicator of distance between resources, even resources of different types. This simplifies the selection of an area to a selecting a range of tiles. When a range of tiles is selected, the independent AREA_GROUP ranges for the different resource types can be easily extracted and applied to a UCF.

This effort used the RapidSmith XDL tool suite [8] extensively for the automated selection and visualization of tile ranges and their corresponding ranges for specific resource types. Both the universal nature of the tile grid and the automation available through RapidSmith motivate using tiles in the experiments that follow.

3. EXPERIMENT SETUP

The xc5vsx240t-2 device is selected as the largest device in the Virtex 5 SX family. The SX series has the greatest ratios of DSPs and BRAMs in relation to the number of CLBs. This gives the SX series architectures a more regular structure, having DSP and BRAM columns spread throughout the fabric. This feature allows for varied area constraints to be explored. The largest device is selected simply because it allows for greater area constraint variation and places a greater burden on the placer.

The submodules selected for testing are listed in Table 1. They represent a sampling of a few dataflow submodules as well as some control-heavy submodules. The Microblaze (MB) and Picoblaze (PB) microcontrollers are available from Xilinx, as well as the CoreGen LUT-based multiplier (Mult). The FIR filter and double-precision floating point quadratic equation solver (FP) are generated from C++ code by Xilinx AutoESL. The FFT originates from a design in SystemGenerator. The XST synthesis resource estimates are listed with each submodule, covering ranges that require between 1 and 10% of the device.

3.1. Implementing Submodules Independently

For these experiments, it is desirable to implement each submodule listed in Table 1 independent of the context of a full design. However, it is the nature of Xilinx XST and MAP to remove any unused outputs or logic. A design consisting of only a submodule without any connected outputs would therefore be optimized out of existence. This optimization can be avoided by surrounding the submodule with black box barriers. These barriers take the form of Xilinx hard macros and are connected to the inputs and outputs of the submodule. In this setup, instead of being surrounded by a full design, the submodule is now surrounded by hard macro barriers. However, the impact of the barriers on the submodule's implementation is negligible for two reasons:

- The barriers are initially completely unplaced and are not grouped into the submodule's area constraint.
- The nets to or from the barriers are not affected by any timing constraint.

Therefore, the placer is free to place these barrier components basically anywhere not occupied by the submodule components. With this freedom, the placer almost invariably places the barriers around the perimeter of the submodule. As far as routing is concerned, since there are no timing constraints on the nets to or from the barriers, any route will do. Essentially, the barrier presence can be ignored and the submodules can be treated as independent of a full design.

Table 1. Lists the sample of submodules and the Xilinx XST estimated resource utilizations when targeting the xc5vsx240t-2 device. Also listed are the clock period timing constraints and the percentage of implementations which met timing without area constraints. Implementations were over 100 Xilinx MAP (placer) cost table seeds.

Submodule	Logic LUTs	Memory LUTs	Registers	BRAMs	DSPs	Clock	Met Timing
(FFT) 1024-point 18-bit FFT	2574	571	4001	5	12	2.8 ns	59 %
(FIR) 128-tap 18-bit FIR filter	3106	36	7376	0	100	3.4 ns	35 %
(FP) Double-precision quadratic	13270	450	21584	12	40	5.0 ns	23 %
(MB) Microblaze	1395	84	1443	0	3	5.0 ns	27 %
(Mult) 18-bit LUT multiplier	362	23	466	0	0	2.7 ns	41 %
(PB) Picoblaze	113	34	135	0	0	2.8 ns	27 %

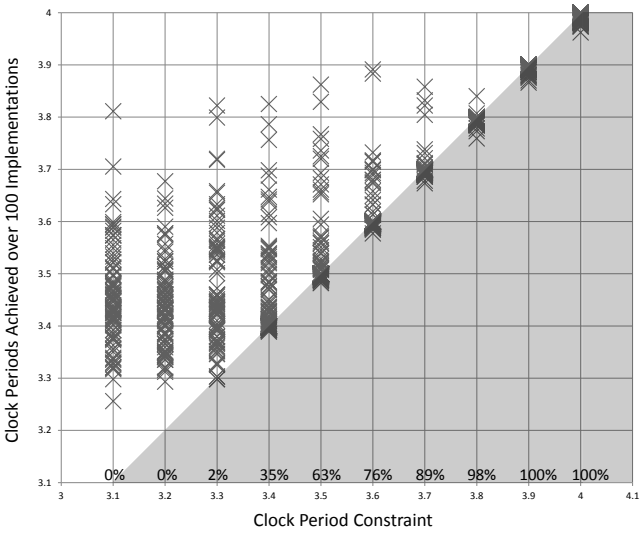

Fig. 3. FIR submodule implemented without area constraints, but varying clock period constraints. At each constraint, the clock period achieved is plotted for 100 placement seeds. The percentage of implementations that met the constraint are also indicated. The shaded region overlaps all implementations that met their constraints.

3.2. Establishing Baseline Clock Constraints

Before applying any area constraints, a baseline clock constraint is selected for each submodule in Table 1. A baseline clock period constraint is selected for each submodule that is not simple to achieve, yet not overly difficult. Each submodule is placed and routed over a range of 100 ps steps of clock constraint without any area constraints. An example of this is shown in Fig. 3 for the FIR submodule. Each clock constraint is implemented for each of 100 placer seeds. At the high end of the clock constraint range all implementations meet the constraint. At the low end none of the implementations meet their constraint. The first constraint that is realizable for the FIR submodule is 3.3 ns, but only 2% of implementations achieve that. This constraint may be unrealizable as soon as this submodule is placed in a full design. The next constraint, 3.4 ns, is realized by 35% of the implementations, and is selected as the clock constraint which is applied in conjunction with area constraints. The other submodules follow a similar baseline clock constraint selection process. The baseline clock constraint and percentage of implementations that meet that clock constraint without an area constraint are listed in Table 1 for each submodule.

3.3. Variation of Area Constraints

Only rectangular area constraints will be considered in these experiments. Obscure area constraints are avoided not only to simplify the exploration but also because area constraints do not constrain the routing, so the routing will naturally overlay a rectangular area even if the area of the placement is an "L" or "T" shape. The rectangular area constraints are varied in two dimensions. The first dimension is the aspect ratio of the rectangle defined by (1).

$$Aspect\ Ratio = \frac{Area\ Constraint\ Tile\ Width}{Area\ Constraint\ Tile\ Height} \quad (1)$$

For the implementations of the submodules, the aspect ratio is allowed to vary over all unique ratios from combinations of integers between 1 and 5 (1:1, 1:2, 1:3, 1:4, 1:5, 2:1,...,5:4). The second dimension is the area overhead percentage or the rectangle defined by (2).

$$Overhead\ \% = \left(\frac{Resources\ Within\ Area}{Est.\ Resources\ Required} - 1 \right) \times 100 \quad (2)$$

By this definition, an area overhead of 0% implies that the area constraint encloses just the resource requirement estimated by synthesis, while an area overhead of 50% implies that the area constraint encloses 50% more resources than the required number of resources. For the implementations of the submodules, the overhead is allowed to vary between 0 and 150% at 10% increments.

Each submodule is assigned a starting tile around which its area constraints should be centered. This tile is chosen to be near DSPs or BRAMs if they are required, or to be in a CLB rich region if they are not. It is chosen to be nearly centered on one side of the device's central clock column. This location is chosen to maximize the allowable horizontal and vertical growth of the area constraints considered while avoiding, as much as possible, horizontal growth that crosses the central clock column. It has been noticed that wire segments that cross the central clock column can have greater delays than corresponding wire segments elsewhere on the device (more on this in the results). For each submodule, the UCF area constraints that correspond to each combination of aspect ratio and overhead are generated with a custom Java program based on the RapidSmith framework.

Each submodule is implemented for each of 100 MAP (placer) cost table seeds for each combination of aspect ratio and overhead, requiring 182,400 iterations of Xilinx v13.1 MAP and PAR. PAR options are left defaulted while MAP options include "-t *" to vary the cost table seed and "-u" to prevent the entire submodule from being optimized away. The experiments were scripted across a variety of machine clusters available at the Brigham Young University supercomputing facility.

4. RESULTS

The results of the experiments described in the previous section are shown in Fig. 4. Each graph displays the percentage of implementations of a specific submodule which met its clock period constraint at each combination of aspect ratio and area overhead.

A somewhat unexpected generality found in any submodule other than the FIR filter is that any combination of aspect ratio and area overhead can meet a reasonably difficult clock period. This is noticed by the existence of nonzero percentages at any point in the graphs of Fig. 4 for any submodule except the FIR submodule. This is not necessarily intuitive because it is generally understood that deviating from a square aspect ratio can increase total wire length and that restricting area can adversely affect the quality of the placement or subsequent routability. This may be partially explained by the fact that the submodules all have at least about a 1:1 LUT to register ratio, so there should be a small percentage of paths which could become the critical path.

4.1. Area Overhead Effects

Although generally any area overhead can meet the target clock constraint, it is noticed that very low area overheads of 0-20% tend to have the fewest implementations that meet the clock constraints, often fewer than when the submodule is implemented without an area constraint. An exception to this occurs when a near ideal aspect ratio is used in conjunction with low overhead such as seen with the Mult submodule at an aspect ratio of 0.2-.33 and overhead of 0%. The Mult submodule has a large number of vertical shift and carry chains, and a small overall area, so it prefers area constraints with lower (vertically oriented) aspect ratios. As the area overhead increases, there is more room for the placer to skew a submodule away from the area constraint's aspect ratio and toward an aspect ratio more preferable for the submodule, whatever that may be. Therefore, higher area overhead can lead to a greater percentage of the implementations meeting the timing constraints.

Generally, above 20% overhead, the percentages of implementations meeting the timing constraints are similar to percentages when the submodule is implemented without area constraints. The Microblaze and Picoblaze, however, display positive exceptions to the trend with dramatic improvements when implemented with area constraints. The Microblaze in particular only met its timing constraint in 27% of implementations without area constraints, yet most of the results with area constraints meet the timing constraint in over 80% of implementations.

4.2. Aspect Ratio Effects

Aspect ratio generally appears to have little effect on any of the results once area overhead exceeds about 20%. At lower area overheads, the effects of aspect ratio are most pronounced at the extremes. In particular, the Microblaze and Picoblaze implementations suffer at low (vertically oriented) aspect ratios and low overhead. This is in contrast to the Mult submodule which showed a strong preference for low aspect ratios. Meanwhile, the FP submodule has greater difficulty at both the high and low aspect ratio extremes with low overhead. Breaking the generality, the FIR submodule implementations simply will not meet its target clock constraint except at moderate aspect ratios, regardless of the area overhead.

The FIR filter demonstrates an exception to the generalities noticed among the other submodules. This may be due to its composition. The FIR submodule differs from the other submodules by utilizing a disproportionately large number of DSPs in relation to the number of other resources. The overall tile area selected is dominated by the high DSP requirement. This condition makes it difficult for the relatively few registers to adequately fill in the total area, requiring longer path delays between registers. Path delays are found to be minimized when the tile area is minimized and the tile aspect ratio is near 2.0. The actual underlying physical geometries of the tiles are unknown, but it is noticed that wire delays are about equal for an equivalent number of interconnect tile hops in either the horizontal or vertical direction. Consider a wire beginning at the lower left interconnect tile in Fig. 2. The nearest interconnect tile to the north is one tile away, while the nearest interconnect

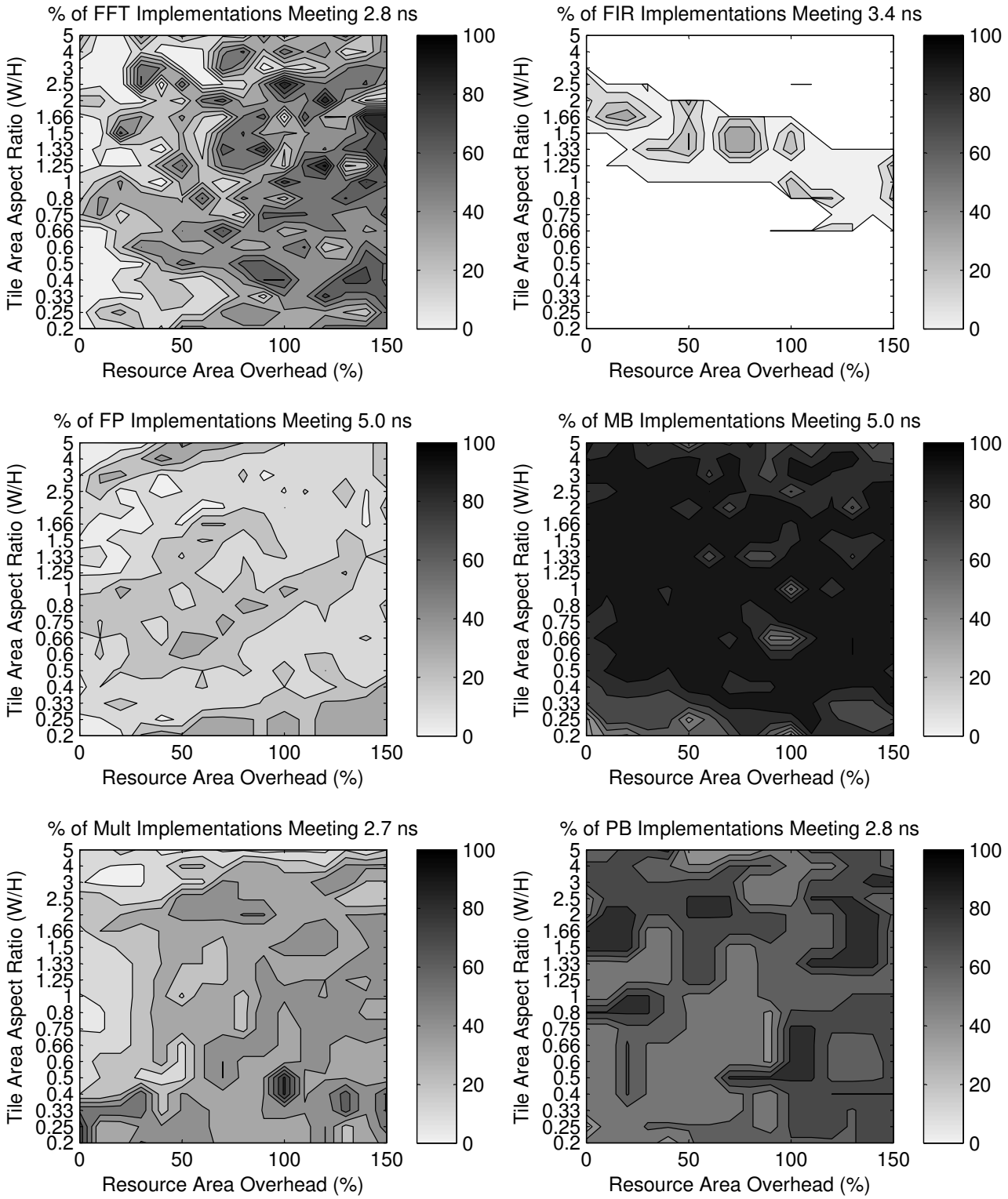

Fig. 4. Each graph shows the percentage of implementations of a specific submodule meeting its specified clock constraint when rectangular area constraints are applied. The rectangle aspect ratio (width/height) varies along the vertical axes, while the area overhead varies along the horizontal axes. The percentage of implementations meeting timing is mapped to the gradient bar at the right of each graph.

tile to the east is three tiles away. The wire going to either has essentially the same delay. Over a large range of tiles on any Xilinx device, the average horizontal spacing of interconnect tiles is about double that of the average vertical spacing. This leads to a preferred tile aspect ratio of 2.0 to minimize path delays. Therefore, at low area overhead, the FIR implementations which meet the clock constraint form a band centered on an aspect ratio of 2.0.

It is noticed that the lower bound on the aspect ratio of the FIR submodule decreases as the area overhead increases. This is easily explained because as the area overhead increases, progressively lower (vertically oriented) aspect ratio constraints can still fit a higher (horizontally oriented) aspect ratio placement within the area boundaries. The decrease in the aspect ratio upper bound as the overhead increases is not as intuitive. The reason for this trend has to do with crossing the central clock column of the device.

4.3. Central Clock Column Effects

The central column of the device contains specialized tiles for clocks, I/O, etc. Wire segments which cross the central column can have greater delay than corresponding wire segments elsewhere on the device. The exact differences in delay depend on segment length and load, but can be on the order of hundreds of picoseconds. Slightly increased delays can also be found on wires crossing horizontal clock region boundaries, but these delays are only on the order of tens of picoseconds. The upper bound on the aspect ratio for the FIR submodule corresponds precisely with the area constraint crossing over the central column, causing routes that cross the central column to have increased delay. However, it should be noted that the central column effect may be overly pronounced in the FIR submodule because of the impact of the disproportionate number of resources. The FP submodule spans the central column at higher (horizontally oriented) aspect ratios as well and has many implementations meeting the clock constraint in those cases. So, while it does not disrupt all implementations, an area constraint which includes the central column may have a negative effect on timing.

5. CONCLUSION

The results profile how area constraints affect timing in submodules independent of a full design. They help answer the following questions:

What aspect ratios are best for submodules that comprise the floor-plan? The general results suggest that most submodules are aspect-ratio agnostic. The exceptional case, however, indicates that the tile aspect ratio should be selected to be near 2.0 if possible.

How much area should be allocated for a submodule? The minimum area overhead of 0% can provide implementations that meet the clock constraint, however, to meet timing most often, an area overhead of at least 20% is preferred.

What impact do area constraints have on the maximum clock rate for a submodule? In general, area constraints do not prohibit meeting maximum clock rate. Variations of aspect ratio and area overhead can impact how often the maximum clock rate is achieved, but it is achievable with an area constraint. For some submodules, area constraints can substantially improve how often the maximum clock rate is met when compared with their unconstrained counterparts.

What guidelines should be followed when assigning submodules to physical locations on the FPGA? If possible, submodules should not be assigned to areas that cross the central column of the device.

This work does not consider effects introduced by the context of a full design, but this is a natural direction for future work. Future work could also explore less regular architectures, such as those with only one DSP column or large areas occupied by fixed functionality silicon. Variation of timing results could also be explored when the area constraint is assigned to different locations on the FPGA fabric. In addition to area constraint effects on timing, there are effects on routes which spill outside the constrained area. This spillover can impact routed submodule reuse when wire use conflicts with that of another submodule. Understanding the routing spillover could improve the effectiveness of incremental or partial-reconfiguration design flows.

6. REFERENCES

[1] Xilinx, "Floorplanning methodology guide," 2011, v13.1.

[2] ——, "Hierarchical design methodology guide," 2011, v13.1.

[3] ——, "Planahead user guide," 2011, v13.1.

[4] A. Kaviani, "Using design hierarchy to improve quality of results in fpgas," in *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*, ser. Lecture Notes in Computer Science, M. Glesner, P. Zipf, and M. Renovell, Eds. Springer Berlin / Heidelberg, 2002, vol. 2438, pp. 791–819.

[5] K. Li, L. Lei, Q. Guang, J.-Y. Shi, and Y. Hao, "Improving the performance of an soc design for network processing based on fpga with planahead," in *Electronics, Communications and Control (ICECC), 2011 International Conference on*, sept. 2011, pp. 297 –300.

[6] Xilinx, "Constraints guide," 2011, v13.1.

[7] C. Beckhoff, D. Koch, and J. Torresen, "The xilinx design language (xdl): Tutorial and use cases," in *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2011 6th International Workshop on*, June 2011, pp. 1–8.

[8] C. Lavin, M. Padilla, J. Lamprecht, P. Lundrigan, B. Nelson, and B. Hutchings, "Rapidsmith: Do-it-yourself cad tools for xilinx fpgas," in *Proceedings of the 21th International Workshop on Field-Programmable Logic and Applications (FPL'11)*, September 2011.