

Architecting Hard Crossbars on FPGAs and Increasing their Area Efficiency with Shadow Clusters

Peter Jamieson and Jonathan Rose

Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto
Toronto, Ontario, Canada M5S 3G4

jamieson@eecg.toronto.edu, jayar@eecg.toronto.edu

Abstract—We explore the architecture of on-chip hard crossbars in FPGAs and show that the area efficiency of such FPGAs can be improved when combined with shadow clusters (which are soft-logic LUT-based clusters that are architected to sit “behind” the multiplier), as an exemplar of an application circuit that appears less commonly in the designs targeting FPGAs.

The metric that we seek to improve is the “frequency” that the need for hard crossbars must appear in the FPGA’s target application suite for the inclusion of the hard crossbar to appear to be area-neutral. For example, we show that this break-even point for a hard 32 full-way crossbar changes from 32% of benchmarks needing to require crossbars to 9% for FPGAs with shadow clusters.

I. INTRODUCTION

Modern heterogeneous Field-Programmable Gate Arrays (FPGAs) employ hard specific purpose circuits to narrow the area, speed and power gap between FPGAs and Application-Specific Integrated Circuits (ASICs). As discussed in [10], these structures have the potential to dramatically narrow the area gap, if: first, the hard circuit provides a significant area, speed, or power benefit when implemented on the FPGA compared to implementation using just the regular soft fabric, and second, that the target application market has sufficient demand for the hard circuit.

FPGA novice architects, who exist inside FPGA companies (but typically outside the main architecture team) and outside of FPGA companies (and are typically ASIC designers or IP developers) will often propose the exact specific hard circuit that they are working on to be incorporated in the next-generation FPGA, regardless of the two factors above. These arguments can rage at length, driven by insufficient scientific and economic justification. In this paper, we use a careful and scientific way to measure the efficacy of a hard circuit, and employ a recent architecture concept, called shadow clusters, to enhance the impact of a hard circuit that has less demand in its target market.

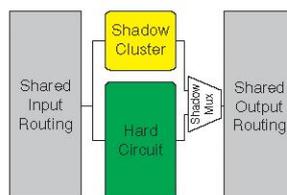


Fig. 1. Illustration of Shadow Cluster Concept

The new architectural concept is called a *shadow cluster*, in which a hard circuit can be programmably replaced with a soft logic cluster at configuration time, as shown in Figure 1. It has been shown that a soft logic cluster combined with a hard circuit, can significantly improve the area efficiency of FPGAs for those hard circuits that are in relatively high demand, such as multipliers. The key to their success is that the hard circuit and shadow cluster logic is architected to have similar routing demand, and therefore, can use the same, very expensive (in area), programmable routing.

In this work, we show that shadow clusters make it more practical to include hard circuits with lower demand in the target market by reducing the penalty incurred in those application circuits that cannot make use of the hard circuit. This concept is illustrated by focusing on crossbar hard circuits, which up to now, have been considered to have a demand too low for inclusion on FPGAs. Both a single bit crossbar and bus-based crossbar, which has a greater area benefit when fully utilized than a single bit crossbar, are included on FPGAs in this study.

The metric that we seek to improve (by decreasing it) is the “frequency” that the need for hard crossbars must appear in the FPGA’s target application suite for the inclusion of the hard crossbar to appear to be area-neutral. For example, we will show that the break-even point for a hard 32 full-way crossbar changes from 32% of benchmarks that demand 16-16 crossbars, for FPGAs without shadow clusters, to 9% for FPGAs with shadow clusters. This kind of change could have significant impact on the architecture “argument” that goes on inside FPGA companies on which hard circuits to include on the device.

To appropriately prepare for this discussion we will also architect the features and parameters of a hard crossbar, including varying the bit-width of the hard crossbar and using either a bus-based crossbar or a single bit crossbar.

The remainder of this paper is organized as follows: in Section II, we describe relevant basic terminology of FPGA architecture and introduce terminology that allows us to speak about heterogeneous architectures including shadow clusters. In Section III, we describe the architecture of the hard crossbar that we explore including in an FPGAs. In Section IV, we describe the experimental methodology that is used to measure the area-efficiency of crossbars combined with shadow clusters, and Section VI presents the results of these experiments and analysis.

TABLE I
FPGA ARCHITECTURAL PARAMETERS USED IN PAPER

Parameter	W	N	K	F_{cin}	F_{cout}	F_s
Architecture	180	10	4	0.18	0.1	3

II. HETEROGENEOUS FPGAS

The basic soft logic cluster *tile* of an FPGA consists of a logic block surrounded by programmable routing. This includes all multiplexers for input and output to the logic block as well as transistors used for switching between global routing tracks. An array of tiles is connected together to form the soft fabric of an FPGA. The soft logic alone is capable of implementing all logic functions. A typical soft logic block consists of a cluster of several Basic Logic Elements (BLEs) which in turn are often some form of Lookup Table (LUT) together with a flip-flop [14].

The architecture of the soft logic fabric has many parameters, including the number of BLEs (N) per cluster, the input size of the LUT (K), the number of routing tracks per channel (W), the input connectivity to the soft logic cluster (F_{cin}), the output connectivity (F_{cout}), the switch block flexibility (F_s) [4] among several other parameters.

For the soft logic fabric we use to illustrate the concepts in this paper, we will select a set of parameters chosen to be close to the typical parameters of modern FPGAs, including the use of direct-drive (also known as uni-directional) routing [11]. The parameters we use in this paper are given in Table I.

A. Supply and Demand Ratio

A key architectural parameter of a heterogeneous FPGA is the ratio of the number of hard circuit tiles to soft logic cluster tiles, which is called the supply ratio - R_s . For example, an FPGA architecture with a supply ratio equal to 1:10 will have one hard circuit tile for every ten soft logic cluster tiles.

We can also describe a given digital design in terms of its demand for hard circuits. Demand ratio, R_d , is the number of hard circuit tiles to the number of soft logic cluster tiles that a digital design requires when implemented on an FPGA, if all circuits capable of being implemented in the hard tile actually are.

A benchmark suite can be described in terms of its average demand ratio. The average demand ratio is calculated by arithmetically averaging the demand ratios of each benchmark. Besides the average demand ratio, it is useful to know the variance of demand ratios among the benchmarks. This statistical variance around the average has important architectural impact as shown in our previous work [6].

B. Hard Circuit Routing Architecture

We now briefly describe how the programmable routing in an FPGA must be architected to accommodate hard circuits based on first principles: a homogeneous soft-logic fabric FPGA with a given soft logic cluster size (N) and LUT size (K) in each tile will require a specific number of tracks per channel to successfully route most benchmark circuits. The parameters N, K, and the routing architecture parameters, are used to calculate the number of pins to be connected to the programmable routing, which is called the *pin demand*. Pin demand includes both the input pins entering the tile and

output pins emanating from the tile. The number of tracks needed in an architecture is a function of pin demand and the other routing architecture parameters given in Table I, as well as the impact on layout. This number is usually determined experimentally by the FPGA architecture team.

If the FPGA includes a hard circuit tile that has a higher pin demand per logical tile than the soft logic tile, it could require more tracks per channel than with the soft logic tile alone. An alternative to increasing the expensive tracks, which we adopt, is to “stretch” the hard circuit over multiple tiles so that the pin demand is roughly matched to that of the soft-logic structure. For example, the pin demand of the soft logic cluster tile for our architecture (described in Table I) is equal to 32 (22 input and 10 output pins). To include a hard circuit that had a pin demand of 62 then we would stretch the hard circuit over two tiles with each tile having a pin demand of approximately 32.

C. Shadow Clusters

Figure 1 illustrates the shadow cluster concept with a tile containing a hard circuit and a shadow cluster. In this Figure, the programmable input routing drives both the BLEs of the shadow cluster and the hard circuit, and a multiplexer selects which output to employ, under the usual programmable control. Only one will be active at a time.

The benefit of a shadow cluster is that adding this structure to a hard circuit allows the tile to be used even when the hard circuit cannot. This means that 50% to 90% of the area used for programmable routing to connect the hard circuit is not wasted. In previous work [6], it is shown that adding shadow clusters to multiplier tiles often results in an improvement in area-efficiency for realistic benchmark suites. In the best results, the shadow cluster concept reduced implementation area by 12.5% compared to an FPGA without shadow clusters.

The size of a shadow cluster to be combined with a hard circuit is related to the number of input and output pins connecting to the hard circuit. Previously, we discussed that hard circuit tiles have a pin demand matching that of the soft logic cluster tile. Since the pin demand describes the input and output pins, and the pin demand of a hard circuit matches that of the soft logic cluster tile, the shadow cluster size is the same size as the soft logic cluster tile (N is the same for both shadow and soft logic cluster tile).

III. ARCHITECTING HARD CROSSBARS FOR FPGAS

We are using hard crossbars as an exemplar of a circuit currently not included on FPGAs likely due to the fact that there are not a sufficient number of designs in the FPGAs target market that would benefit from its inclusion. Crossbars are circuits commonly used in communication applications as well as other digital circuits such as the interconnect for a multiprocessor system [18]. FPGAs are used to implement crossbars in the soft logic with published works in both academia [5], [9] and industry [1], [16], [17]. Jones and Wilton have two patents [8], [7] on adding a bus-based crossbar to a programmable device. The structure of their bus-based crossbar is very similar to the one we will present in this work.

In this section, we will describe the design of the hard crossbars we include on FPGAs. This description includes

the range of crossbar sizes that can be implemented on each crossbar, the number of logical tiles the crossbar is stretched across, and the crossbar architecture including a single bit crossbar and a bus-based crossbar.

A. Definition of Crossbar Terms included on a FPGA

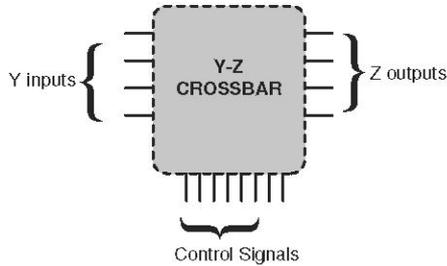


Fig. 2. A full-way crossbar

Figure 2 shows a crossbar in which Y inputs pins can be routed to Z output pins. We will consider a full-way crossbar that can be dynamically controlled to broadcast one input to all outputs, unicast each input to a unique output, or a mixture of both. This type of crossbar can be implemented as Z multiplexers where each multiplexer is a Y to 1 multiplexer.

TABLE II
CROSSBARS INCLUDED ON AN FPGA

Crossbar Name	Max Size	Num. 16-16 Crossbars	Num. 32-32 Crossbars	Num. 64-64 Crossbars
16-16	16	1	NA	NA
32-32	32	2	1	NA
64-64	64	4	2	1

We will consider three different crossbars on an FPGA; Table II summarizes these three crossbars. Column 1 shows the type and name we will use to identify the crossbar; column 2 shows the maximum size crossbar that the hard circuit can implement using just the hard circuit. Columns 3, 4, and 5 show how each hard circuit can potentially be configured to implement different sized crossbars. For example, the 64-64 crossbar can be configured to implement either four 16-16 crossbars, two 32-32 crossbars, or one 64-64 crossbar.

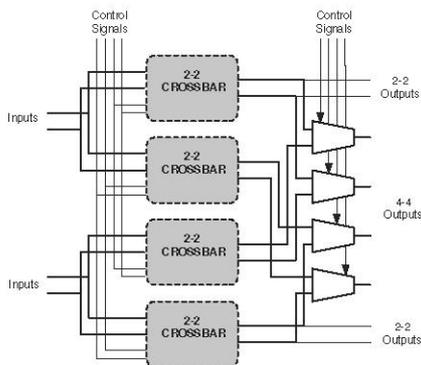


Fig. 3. 4-4 crossbar implemented with 2-2 crossbars

To build a flexible hard crossbar like the 32-32 and 64-64 in Table II, the design includes some shared inputs and additional

multiplexers. Figure 3 shows an example of a 4-4 crossbar that can also be used to implement two 2-2 crossbars. In this figure, it takes four 2-2 crossbars to implement one 4-4 crossbar, and if this crossbar is in 4-4 mode then the first four control signals control selection in each of the 2-2 crossbars, and the last 4 control signals control the multiplexers at the output of the 2-2 crossbars. If this structure is used to implement two 2-2 crossbars then the top and bottom 2-2 crossbars implement these operations, and this mode only uses the first four control signals. This sharing construction principle is used for both the 32-32 and 64-64 crossbar noting that some active area (making up the multiplexers) will be wasted depending on the mode of the crossbar.

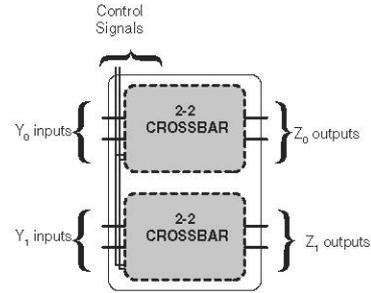


Fig. 4. A bus-based crossbar consisting of two 2-2 crossbars

The most straightforward crossbar is one that allows individual control of each data bit. We call this a *single bit* crossbar. As an alternative, more than one data bit can be controlled by a single set of control signals, which we call a *bus-based* crossbar. Figure 4 shows the structure of bus-based crossbar with a bus size of 2. In this figure, Y_0 can be routed to Z_0 outputs sharing the control signals with the other 2-2 crossbars. Since the control signals are shared in the bus-based crossbar it reduces its pin demand and increases the area-benefit of these crossbars if the pins of the hard circuit are highly utilized.

B. Hard Crossbar Pin Demand and Number of Tiles

Given these hard crossbars to include on an FPGA and using the architecture parameters for Architecture 1 described in Table I for our experimental FPGAs, we now determine how many logical tiles the hard crossbars will be stretched over by dividing the total number of pins used by the crossbar divided by the pin demand of the soft logic cluster tile. The total number of pins needed to implement a Y - Z crossbar equals:

$$totalpins = Y + Z + (Z \lceil \log_2 Y \rceil) \quad (1)$$

In this equation, Y and Z represent the input and output pins. The final term represents the number of pins needed for the control signals to select paths through the crossbar.

The total number of pins needed to implement X Y - Z crossbars where X represents the bit-width of a bus-based crossbar equals:

$$totalpins = X * (Y + Z) + (Z \lceil \log_2 Y \rceil) \quad (2)$$

The major pin cost in the single bit crossbar is the number of control pins, and this is due to the choice that crossbars are full-way needing many control pins to make each possible switch pattern. It is possible to implement a crossbar with

less flexibility and consequently less control pins, but without detailed knowledge on how targeting designs use crossbars, we take a worst case approach and use fully flexible crossbars. Some of the pin cost is reduced in the bus-based crossbar since the control signals are shared between all the crossbars in the hard circuit.

TABLE III
TILES PER HARD CROSSBAR

Crossbar Name (Y-Z)	Bus Size (x)	Pins in Crossbar	Pin Demand	Number of Tiles
16-16	1	96	32	3
16-16	4	192	32	6
16-16	16	576	32	18
64-64	1	512	32	16
64-64	4	896	32	28
64-64	16	2432	32	76

Table III shows the number of tiles hard crossbars will be stretched over for our architecture as described in Table I. Column 1 shows the name of the crossbar, column 2 shows the number of bits in the bus, and column 3 shows the total number of pins in each crossbar. Columns 4 shows the soft logic cluster tile pin demand, and columns 5 shows how many logical tiles are needed to implement one hard crossbar. We can see that a bus-based crossbar will implement more crossbars in less tiles.

C. Hard Crossbar Benefit over Soft Logic Implementation

With a description of the architectures and sizes of the hard crossbars that we will include on an FPGA, we can make preliminary calculations as to what area benefit using a hard crossbar will have over implementing a crossbar in soft logic cluster tiles. This benefit is calculated by taking the number of soft logic cluster tiles needed to implement the crossbars in a design multiplied by the size of the soft logic cluster tile divided by the number of tiles a hard crossbar is stretched over multiplied by the size of the hard crossbar tile.

Hard crossbar tiles and soft logic cluster tiles use approximately the same area since the dominating area component in both tiles is the programmable routing (both a LUT and crossbar are essentially a few pass transistors that implement multiplexers). In this case, we can simplify the calculations and simply divide the number of soft logic cluster tiles needed to implement a crossbar by the number of tiles in a hard crossbar.

To calculate the number of soft logic cluster tiles needed to implement a crossbar in a design, we use Altera's Quartus CAD tool [3] to map crossbars to the soft logic of a Stratix I FPGA [2] (similar to the architecture parameters we use in this work).

Table IV shows different sized crossbars in a design and what benefit these crossbars will have when implemented on a 16-16, 32-32, and 64-64 single bit hard crossbar. For example, Table IV shows that a 32-32 design crossbar uses 7 hard logical tiles versus 77 soft logic cluster tiles.

Table V shows the gains of a 4-bit bus-based, 32-32 hard crossbar implementing 32-32 crossbars in a design with different bus width utilization compared to implementing those same crossbars in soft logic. Column 1 and column 2 show the size of the crossbars in the design and how many bits of

TABLE V
RELATIVE BENEFIT OF 4-BIT BUS-BASED 32-32 HARD CROSSBAR

Design Crossbar Size (Y-Z)	Bus Utilization	Soft Cost in clusters (N=10)	Hard Crossbar	
			16-16	Gain Factor
32-32	1	77	17	4.53
32-32	2	154	17	9.06
32-32	3	231	17	13.59
32-32	4	308	17	18.11

the bus these circuits use. Column 3 and 4 shows how many soft logic cluster tiles and how many bus-based hard crossbar tiles it takes to implement a crossbar. Column 5 shows the gain factor of the hard implementation over a soft.

Comparing a single bit hard 32-32 crossbar's gain factor (which is equal to 11) to the results in Table V. We can see that the bus-based crossbar provides an area benefit greater than that of the single bit crossbar when more than 25% of the bits in the bus-based crossbar are used. When the bus-based crossbar is fully utilized there is a significant area benefit, but these gains will only be seen for an FPGA architecture when the all of the hard bus-based crossbars are fully utilized.

IV. MEASUREMENT METHODOLOGY

Our goal is to measure the area effectiveness of hard crossbars combined with shadow clusters to determine if it is beneficial to add this type of tile to FPGAs. We use an empirical approach that measures the area consumed by a suite of benchmarks mapped to different architectures. This section describes how benchmarks are mapped to different architectures, how the area of a soft logic tile and the hard crossbar tiles is calculated, and how synthetic benchmarks are created to represent possible target markets.

We use a measurement methodology that first maps a benchmark to the different tiles available on the FPGA, and then calculates the area of the FPGA based on the tiles used.

A. Mapping Benchmarks to Architectures

To measure the area consumed by a design we map a benchmark to tiles available on the FPGA. We map benchmarks to three types of FPGA architectures: without hard crossbars, with hard crossbars, and with hard crossbars including shadow clusters.

The benchmarks to be mapped to these architectures are modelled as requiring a number of soft logic cluster tiles and crossbars. The mapping step assigns crossbars to either hard crossbar tiles, soft logic cluster tiles, or a mixture of both. This is necessary since an FPGA may either not have enough or any hard crossbars, or the design crossbars may be of a size larger than the hard crossbars can implement. In the case that the design crossbar is larger than the hard crossbar, a combination of hard crossbar tiles and soft logic cluster tiles can be used to implement the design crossbar in less area than soft logic cluster tiles alone.

We will follow the usual practice in FPGA architecture research [13] and allow the size of the FPGA to be matched to the size of each benchmark, while maintaining the key FPGA architectural parameters.

The number and type of tiles required for a benchmark on a particular FPGA architecture is determined by increasing

TABLE IV
RELATIVE BENEFIT OF HARD CROSSBARS OVER SOFT CROSSBARS

Design Crossbar Size (Y-Z)	Soft Cost in clusters (N=10)	16-16 tiles	Gain Factor	Hard Crossbar		64-64 tiles	Gain Factor
				32-32 tiles	Gain Factor		
8-8	4	3	1.33	7	0.57	16	0.24
16-16	18	3	6.00	7	2.57	16	1.13
32-32	77	-	-	7	11	16	4.81
64-64	308	-	-	-	-	16	19.3

the number of soft logic cluster tiles and hard crossbar tiles until the benchmark design fits the FPGA. This is done by incrementally increasing the number of hard crossbar tiles, mapping the crossbars in the design to either available hard crossbars or soft logic cluster tiles, and determining if there is enough soft logic cluster tiles (calculated with the supply ratio) for the design.

The actual mapping of crossbars to hard crossbars on the FPGA is done in the following way. Given the set of crossbars in the design and a table similar to Table IV, which is extended for all crossbar sizes found in the designs, we rank each design crossbar in order of the largest gain factor to least. Once the crossbars in the designs are ranked in order of benefit we start mapping the highest ordered crossbars to available hard crossbars.

After all the hard crossbars have been mapped to on the FPGA, the remaining unmapped crossbars in the design are mapped to soft logic cluster tiles using a simple lookup from a table generated by mapping crossbars of all sizes to soft logic implementations on a Stratix I FPGA [2]. In the case where the architecture has shadow clusters, then the mapping algorithm uses the shadow clusters when a hard crossbar is not being used.

After this mapping, the number of each type of tile is known, and the area of the FPGA is calculated by multiplying the tile requirements by each tiles area.

B. Transistor Area Estimation of Tiles

The relative area of the soft logic cluster tile and the crossbar tile are determined in a 90nm CMOS process [15], [12] on a transistor-level design of the tiles, and our own automated transistor sizing method. Space limitations prevent the description of the details of this design process.

For the hard crossbar, multiplexers are implemented using pass transistors, and any wires that connect across one logical tile or the connections between the hard crossbars used to build larger crossbars (such as the crossbar in Figure 3) use a level restoring buffer. These circuits are included in our automatic sizing method.

Table VI shows the area profile of some of the different tiles (including the soft logic tile, the hard crossbar tile, and the shadowed hard crossbar tile) on a percentage basis for our experimental architecture. The final columns show the size of each tile relative to the soft logic cluster size N=10 for Table VI. For each crossbar the values represent only one of the logical tiles used to make the entire crossbar.

Table VII shows the size of each tile relative to the soft logic cluster size N=10 for Table VI for a subset of the hard bus-based crossbars we study. The hard bus-based crossbars tile's area slightly increases in size as bit-width increases. This per tile area increase is due to the buffers that drive the shared

TABLE VI
PERCENTAGE AREA WITHIN A TILE AND RELATIVE AREA

Tile Type	BLEs	Crossbar	Routing	Relative Size
Cluster (N=10)	18%	-	82%	1.00
Single bit Crossbar 16-16 (1 of 3 tiles)	-	2%	98%	0.97
Single bit Crossbar 16-16 shadow cluster (1 of 3 tiles)	15 %	1%	84%	1.15
Single bit Crossbar 64-64 (1 of 16 tiles)	-	6%	94%	1.10
Single bit Crossbar 64-64 shadow cluster (1 of 16 tiles)	15%	4%	81%	1.27

TABLE VII
RELATIVE TILE AREA FOR HARD CROSSBARS COMPARED TO A SOFT LOGIC CLUSTER TILE

Tile Type	Bus bit-width	Relative Size per N=10	Relative Size per N=10 with Shadow Cluster
Cluster (N=10)	1	1.0	-
64-64 (1 of 16)	1	1.10	1.27
64-64 (1 of 28)	4	1.21	1.35
64-64 (1 of 76)	16	1.27	1.39

control signals and more transistors being packed into each logical tile.

V. BENCHMARKS

We now discuss the model used to describe benchmark applications that include crossbars. Our measurements only require the number of soft logic tiles required in a circuit, and the number and type of crossbars in a design including how many bus bits they use (if they are bus-based crossbars). This allows us to model the benchmarks with just these numbers, but leaves us with the problem of validating whether any of these numbers realistically represent actual FPGA markets. Part of this problem is solved by the way we posed the question in the introduction - we wanted to show the effect on the demand of crossbars in the target market required for the area-efficiency measurement to break even. To answer this, we vary the benchmark crossbar demand, and so our results give that demand as an output rather than an input. Within each benchmark, however, there are different possibilities for how the circuit can demand the crossbar - they could be small or large, and therefore have a specific internal distribution of demand that needs to be realistic as well.

Figure 5 shows the general form of benchmark distributions

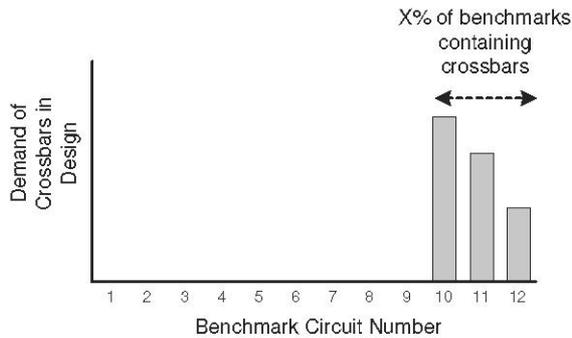


Fig. 5. Example Distribution Crossbar Demand in Synthetic Benchmarks

we generate to represent the benchmarks targeting FPGAs. It is based on our observations of real benchmarks that suggest that only subset of circuits will have non-zero demand for crossbars. There is support for this observation in the fact that no widely used commercial FPGA yet contains crossbars of the nature we have described. The two key parameters of the distribution are the percentage of benchmarks containing crossbars and the average demand ratio for crossbars within those benchmarks. Creating our benchmarks in this fashion allows us to change the percentage of benchmarks containing crossbars so that we can model a range of benchmark distributions.

Table VIII shows some examples of synthetic benchmark suites used in our experiments. Within this table we report the benchmark name, the number of benchmarks, the percentage of benchmarks containing crossbars, the average demand ratio of benchmarks containing crossbars, the benchmark suite’s average demand ratio, the range of BLEs per benchmark, and the range of the number of crossbars per benchmark.

In all benchmarks, the size of the design crossbars are either all 16-16, 32-32, or 64-64. Regardless of the size of the crossbars, the demand ratio remains the same per benchmark since demand ratio is normalized to 64-64 hard crossbars and a soft logic cluster size of 10 LUTs per cluster. For example, a design with 2400 soft logic cluster tiles and a demand ratio equal to 1:15 will include either 160 16-16 hard crossbars, 40 32-32 hard crossbars, or 10 64-64 hard crossbars.

Each individual benchmark that contains crossbars has a demand ratio between 1:1 (representing a design similar to a digital router with a primary function to route packets to destinations) to 1:227 (representing a design that needs very few crossbars such as a multi processor system that needs a network to communicate with each processor in the system). The average demand ratio for each benchmark suite depends on the percentage of benchmarks containing crossbars, and for the benchmarks that do contain hard crossbars the average demand ratio for these benchmarks is set to 1:15.

The last parameter that we vary within our synthetic benchmarks is the bus utilization of the benchmarks that use crossbars. We do not create a benchmark suite in which each design has different bus bit-width demands, and instead, for all benchmarks in the suite and for all crossbars used each bus-based crossbar has the same bus bit-width. For example, one of the benchmark suites will consist of 12% of the benchmarks that use hard bus-based crossbars of bit-width 5.

Our approach is to create a range of benchmark suites

based on demand for crossbars, bus utilization for bus-based crossbars, and crossbar size to represent a range of possible markets that would target FPGAs. In this way, we can make observations of the area efficiency of different FPGA architectures serving markets with different demand characteristics. Given the characteristics of a target market, we can then state what are the FPGA architectures that result in the most area-efficient implementation of the designs within each market. This approach is not the most desirable approach (which would use real benchmarks), but generating a wide range of target markets based on synthetic benchmarks is a reasonable method to at least observe architectural trends based on possible market characteristics.

VI. RESULTS

We measure the relative area efficiency of FPGAs with hard crossbars and with hard crossbars and shadow clusters compared to FPGAs without hard crossbars to determine the area effectiveness of hard crossbars. As described above, this is done by mapping suites of benchmark circuits into each type of FPGA and then measuring the area of the resulting FPGAs.

The benchmark suites will be mapped to the soft logic architectures described by the parameters in Table I. We map the benchmarks to FPGAs that contain 16-16, 32-32, or 64-64 hard crossbars (as shown in Table II) that in some cases will be bus-based and include shadow clusters.

We use the area of the pure soft logic FPGA as the basis for comparison for the crossbar-based architectures - normalizing by dividing the soft logic-only area by the area for the other architectures. Thus, if this area ratio is greater than one it means that the experimental architecture is using less area than the pure soft logic FPGA. Finally, we will geometrically average all these area ratios for a set of benchmarks implemented on a particular architecture, and the average represents how well the experimental architecture compares to a soft logic FPGA when implementing that particular benchmark suite.

One of the metrics that we seek for each experimental FPGA architecture is the “frequency” that the need for hard crossbars must appear in the benchmark suite for the inclusion of the hard crossbar to appear to be area neutral. This “frequency” is determined by the area break-even point where this break-even point is determined as follows. We will map the benchmark SB_1 (which has 1% of the benchmarks that use crossbars) to the architecture under study, where the mapping includes varying the supply ratio for hard crossbars and finding the best supply ratio (as described in our previous work [6]). If the geometrically averaged area ratio is greater than one then the break-even point is 1% for this particular architecture. Otherwise, we now map SB_2 to the architecture and repeat the process. This is continued until we find the percentage of benchmarks containing hard crossbars at which the experimental and soft logic FPGAs are area neutral.

A. Effectiveness of Hard Crossbars with and without Shadow Clusters

In our first experiment, we will look at how a shadow cluster changes the area efficiency of an FPGA that includes hard crossbars to determine how this changes the argument for including hard crossbars on FPGAs. We will measure the

TABLE VIII
EXAMPLES OF SYNTHETIC BENCHMARK SUITES WITH CROSSBARS

Name	Num. Bmarks	Percent with Crossbars	Avg. Demand of Bmarks with Crossbars	Avg. Demand	BLE Range	Crossbar Range
SB_5	100	5%	1:15	1:300	10000 to 25000	0 to 350
SB_10	100	10%	1:15	1:150	10000 to 25000	0 to 350
SB_15	100	15%	1:15	1:100	10000 to 25000	0 to 350

area effectiveness of an FPGA with hard single bit crossbars and an FPGA with hard single bit crossbars combined with shadow clusters. The required demand is determined by the “frequency” of the use of crossbars in the designs which results in an area-neutral architecture compared to a purely soft FPGA as described above.

A key question that we posed in this paper was to determine the demand for crossbars at which the area-efficiency of an FPGA with hard crossbars (with and without shadow clusters) would be area neutral compared to a pure soft logic FPGA.

Table IX answers this question - it shows the percentage of benchmarks containing crossbars at which the average implementation area is the same (the area ratio is 1.0) for both a soft logic FPGA and an FPGA with hard crossbars (with or without shadow clusters). Column 1 shows the type of hard crossbar included on the FPGA and column 2 shows the size of the crossbar used by the benchmarks within each benchmark suite. Columns 3 and 4 show the area break-even point and average demand ratio of the benchmark suite that is area-neutral for an FPGA that includes hard crossbars. Similarly, Columns 5 and 6 show the same data for an FPGA that includes hard crossbars combined with shadow clusters.

For example, for an FPGA with 16-16 hard crossbars and no shadow clusters implementing benchmarks with 16-16 crossbars, 18% of the benchmarks must use crossbars (with an average demand ratio of 1:83) for that FPGA to have the same area-efficiency as the pure soft logic FPGA.

The shadow cluster architectures always “break-even” with significantly less demand for crossbars than those without shadow clusters. For example, the 16-16 shadowed architecture implementing benchmarks with 16-16 crossbars requires only 3% of the benchmarks to demand crossbars and an average demand ratio of 1:215 compared to 18% and 1:83 average demand ratio for the same architecture without shadow clusters.

These results show that shadow clusters make it far more practical to include lower-demand circuits on FPGAs, and have potential to alter the architecture argument in FPGA companies in a substantial way. These results also apply to an architecture that includes hard bus-based crossbars.

B. Effectiveness of Bus-based Hard Crossbars

In this experiment, we will compare the hard crossbar architectures by measuring the area efficiency of hard bus-based crossbars and hard single bit crossbars. As discussed in section III hard bus-based crossbars share the control pins between each crossbar in the bus thus reducing the pin demand of the hard circuit and increasing the area benefit if more than one of the bits in the bus are used.

In this experiment, we will fix the benchmark suite to 20% of the benchmarks containing crossbars, which is equivalent

to an average demand ratio of 1:75 for the benchmark suite. For each FPGA that includes hard bus-based crossbars with a specified bus size, we will map each our benchmark suite, SB_20, with a specified crossbar bus utilization ranging from 1 bit to 16 bits. When we map each benchmark to the architecture, we pick the supply ratio that results in the most area-efficient architecture.

We use the area-efficiency ratio to compare each of the architectures implementing the benchmark suite. In each case, an area-efficiency ratio is calculated as the area used to implement the benchmarks on a purely soft FPGA divided by the area to implement the same benchmarks on an experimental architecture, and an area-efficiency ratio greater than one means that the experimental FPGA is smaller than the purely soft FPGA. These area-efficiency ratio are geometrically averaged for each benchmark in the benchmark suite.

Table X shows the area-efficiency ratios for FPGAs with 64-64 hard bus-based crossbars. Column 1 and column 2 shows the size of the hard bus-based crossbars and the bus bit-width. Column 3 and column 4 show the size and bus utilization of the crossbars in the benchmark. Column 5 shows the supply ratio that results in the best area-efficiency ratio for the given benchmark suite mapped to this architecture without shadow clusters, and column 6 shows the area-efficiency ratio. Columns 7 and 8 show the same data except for an architecture with shadow clusters.

These results show that hard bus-based crossbars on an architecture without shadow clusters provides an area-efficiency benefit over a hard single bit crossbar depending on the how much of the bus is utilized. The 4-bit hard bus-based crossbar needs to have a bus utilization of 2 or more to be more area-efficient compared to the hard single bit crossbar. Similarly, the 8-bit hard based crossbar with a bus utilization of 3 and the 16-bit hard based crossbar with a bus utilization of 5 are more area-efficient than the hard single bit crossbar. The same is true if the architecture includes shadow clusters noting that the best supply ratio decreases due to the area-efficiency improvement due to the inclusion of the architectural concept.

We can conclude that the hard bus-based crossbar is a more area-efficient architecture for a hard crossbar included on an FPGA if the target market has sufficient bus utilization of at least a quarter of the bus-based crossbar bits.

VII. CONCLUSIONS

In this paper, we introduced a hard crossbar as a hard circuit to include on an FPGA. Hard crossbars have not been included in FPGAs since there aren’t sufficient number of designs that use hard crossbars. We measured how effective hard crossbars combined with shadow clusters are at improving these FPGAs area efficiency such that the frequency of hard crossbars

TABLE IX
AREA BREAK-EVEN DEMAND POINTS

Hard Crossbar Type	Design Crossbar Size	Crossbar Architecture		Crossbar+Shadow Architecture	
		Break-Even Percent of Benchmarks with Crossbars	Average Demand Ratio	Break-Even Percent of Benchmarks with Crossbars	Average Demand Ratio
16-16	16-16	18%	1:83	3%	1:500
16-16	32-32	10%	1:150	2%	1:750
16-16	64-64	18%	1:83	3%	1:500
32-32	16-16	32%	1:47	9%	1:167
32-32	32-32	12%	1:125	3%	1:500
32-32	64-64	5%	1:300	2%	1:750
64-64	16-16	49%	1:30	12%	1:125
64-64	32-32	15%	1:100	5%	1:300
64-64	64-64	8%	1:188	2%	1:750

TABLE X
AREA-EFFICIENCY RESULTS FOR HARD 64-64 BUS-BASED CROSSBARS

Crossbar Type	Bus Size on FPGA			Crossbar Architecture		Crossbar+Shadow Architecture	
		Design crossbar size	Bus utilization by design	Best Supply Ratio	Area-Efficiency Metric	Best Supply Ratio	Area-Efficiency Metric
64-64	1	64-64	1	1:14	1.075	1:7	1.149
64-64	4	64-64	1	1:14	1.030	1:5	1.111
64-64	4	64-64	2	1:14	1.088	1:7	1.161
64-64	4	64-64	3	1:18	1.119	1:9	1.185
64-64	4	64-64	4	1:18	1.139	1:12	1.197
64-64	16	64-64	2	1:20	1.002	1:5	1.078
64-64	16	64-64	4	1:15	1.056	1:6	1.134
64-64	16	64-64	6	1:15	1.091	1:8	1.160
64-64	16	64-64	8	1:18	1.113	1:9	1.177
64-64	16	64-64	10	1:18	1.128	1:11	1.188
64-64	16	64-64	12	1:20	1.141	1:13	1.195
64-64	16	64-64	14	1:20	1.149	1:14	1.200
64-64	16	64-64	16	1:20	1.157	1:17	1.204

appearing in designs is reduced making FPGAs area-neutral with a purely soft programmable logic one.

Our measurements show that in all cases, the combination of a shadow cluster and a hard crossbar results in an architecture that needs much less demand for hard crossbars. Our results also show that a bus-based hard crossbar will provide a benefit over single bit hard crossbars when approximately 25% of the bus is utilized regardless of if the architecture includes shadow clusters.

REFERENCES

- [1] Altera. *Using Stratix GX in Switch Fabric Systems*, 2002. Altera White Paper.
- [2] Altera. *Stratix Device Handbook*, Jul 2003.
- [3] Altera. *Quartus II Handbook, Volumes 1, 2, and 3*, 2004.
- [4] V. Betz, J. Rose, and A. Marquardt. *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, 1999.
- [5] G. Brebner and D. Levi. Networking on Chip with Platform FPGAs. In *IEEE International Conference on Field-Programmable Technology*, pages 13–20, Dec 2003.
- [6] P. Jamieson and J. Rose. Enhancing the area-efficiency of FPGAs with hard circuits using shadow clusters. In *IEEE International Conference on Field-Programmable Technology*, pages 1–8, 2006.
- [7] C. Jones and S. Wilton. Cascadable bus based crossbar switch in a Programmable Logic Device. *U.S. Patent 6,590,417. Issued July 8th, 2003.*
- [8] C. Jones and S. Wilton. Cascadable bus based crossbar switching in a Programmable Logic Device. *U.S. Patent 6,710,623. Issued March 23rd, 2004.*
- [9] H. Kariniemi and J. Numi. A Crossbar-Based ATM Switch on FPGA for 2.488 Gbits/s CATV Network with Scaleable Header Remapping Function. In *Communication Systems, Networks, and Digital Signal Processing Symposium*, pages 82–85, July 2002.
- [10] I. Kuon and J. Rose. Measuring the Gap Between FPGAs and ASICs. In *ACM/SIGDA International Symposium on FPGAs*, pages 21–30, Feb 2006.
- [11] G. Lemieux and D. Lewis. Directional and Single-Driver Wires in FPGA Interconnect. In *IEEE International Conference on Field-Programmable Technology*, pages 41–48, Dec 2004.
- [12] C. Microsystems, 2007. <http://www.cmc.ca>.
- [13] J. S. Rose, R. J. Francis, P. Chow, and D. Lewis. The Effect of Logic Block Complexity on Area of Programmable Gate Arrays. In *IEEE Custom Integrated Conference*, pages 5.3.1 – 5.3.5, May, 1989.
- [14] A. Singh and M. Marek-Sadowska. Efficient Circuit Clustering for Area and Power Reduction in FPGAs. In *ACM/SIGDA International Symposium on FPGAs*, pages 59–66, 2002.
- [15] STMicroelectronics. 90nm CMOS90 Design Platform, 2005. <http://www.st.com/stonline/prodpress/dedicate/soc/asic/90plat.htm>.
- [16] Xilinx. *High-Speed Buffered Crossbar Switch Design Using Virtex-EM Devices*, 2000. Xilinx Application Note 240.
- [17] S. Young, P. Alfke, C. Fewer, S. McMillan, B. Blodget, and D. Levi. A high I/O reconfigurable crossbar switch. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines*, pages 3–10, April 2003.
- [18] Y. Zhang, T. Jeong, F. Chen, H. Wu, R. Nitzsche, and G. Gao. A study of the on-chip interconnection network for the IBM Cyclops64 multi-core architecture. In *IEEE International Parallel & Distributed Processing Symposium*, pages 1–10, April 2006.