

FPGA Clock Network Architecture: Flexibility vs. Area and Power

Julien Lamoureux and Steven J.E. Wilton
Department of Electrical and Computer Engineering
University of British Columbia
Vancouver, B.C., Canada
{julienl | steview} @ece.ubc.ca

ABSTRACT

This paper examines the tradeoffs between flexibility, area, and power dissipation of programmable clock networks for Field-Programmable Gate Arrays (FPGA's). The paper begins by describing a parameterized clock network model that describes a broad range of programmable clock network architectures. Specifically, the model supports architectures with multiple local and global clock domains and varying amounts of flexibility at various levels of the clock network. Using the model, the architectural parameters that control the flexibility of the clock network are varied to determine the cost of this flexibility in terms of area and power dissipation. From these experiments, the study finds that area and power costs are highest for networks with flexibility close to the logic blocks. Furthermore, it found that clock networks with local clock domains have little overhead and are significantly more efficient than clock networks without local clock domains for applications with multiple clocks.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles – Gate Arrays

General Terms

Design

Keywords

FPGA, clock network, architecture, low-power

1. INTRODUCTION

Modern Field-Programmable Gate Arrays (FPGA's) are now being used to implement entire systems. These systems typically contain multiple clock domains. FPGA vendors have responded by incorporating high-speed, low-skew clock networks that supply multiple clock signals to logic, memory, and arithmetic elements in the FPGA. The design of these clock networks is of the utmost importance. No matter how well optimized the rest of the FPGA fabric is, if the clock network can not supply the required number of clock signals to the portion of the fabric that needs these signals, it may be impossible, or extremely inefficient, to implement many user circuits. In this paper, we focus on the architecture of these clock networks.

An FPGA clock network has the following requirements:

1. It must have low skew. That is, the differences in arrival times of a clock edge to different logic elements must be small. Not only can skew impact the achievable clock speed of a user circuit, but in the worst case, it could cause incorrect operation of the circuit. Compared to a fixed-function chip, such as an Application-Specific Integrated Circuit (ASIC), skew is not as critical for an FPGA, since FPGA cir-

cuits tend to run at lower clock frequencies, and thus, skew will be a smaller component of the clock period.

2. It should consume a reasonable amount of dynamic power. Since clock networks switch every cycle, they consume a significant amount of dynamic power. A well-designed clock network will allow parts of the network that are not being used to be "turned off", saving power.
3. It should require a reasonable amount of layout area, and should be easily to integrate into the tiled FPGA fabric. Although an H-tree topology has low skew [1], it is difficult to mesh such a topology onto a tiled FPGA. As a result, all FPGA vendors use a *spine-and-ribs* topology, in which the clock signal is distributed to each row using a spine network, and then within each row, the clock is connected to each logic element using a rib network.
4. Since the number of clock regions and the sizes of these regions vary from application to application, it is important that these networks are flexible. If the clock networks are not flexible enough, packing and placement constraint imposed by this inflexibility will reduce the quality of the implementation generated by the CAD tools. In the worst case, an inflexible clock network could limit the types of user circuits that can be mapped onto the device.

This balance between flexibility and speed/area/power makes the design of FPGA clock networks much more challenging than clock networks in ASIC's or other fixed-function chips. In this paper, we explore this balance. More specifically, this paper makes the following contributions:

1. We present a parameterized model that describes a family of FPGA clock networks. Such a model is important because it provides a framework within which we can reason about and compare alternative clock networks. The model encompasses the salient features of all commercial FPGA clock networks.
2. We determine the impact of several of these parameters on the area and power consumption of the clock network. In doing so, we identify certain key parameters in our model that have a significant impact on the efficiency of the clock network.

We purposely do not consider the impact of our architectural parameters on skew, since the design of a low-skew network is primarily a circuit-design issue. Once suitable architecture has been found, buffer sizing and placement, as well as phase-locked loop/delay-locked loop positioning can be used to minimize the

skew of the resulting network. In addition, CAD techniques that tolerate skew can be employed [2]. We also do not attempt to determine an optimum number of clock signals or clock regions. These decisions are typically made by FPGA vendors after talking to potential customers and are often based on the target market for a device. Instead, our approach in this paper is to ensure that our clock model is flexible enough to encompass any envisaged clock network, and to quantify how decisions regarding the clock model will impact the area and power dissipation of the network.

This paper is organized as follows. Section 2 provides some background on commercial FPGA clock networks. Section 3 describes our parameterized FPGA clock network model. Section 4 compares members of this parameterized family in terms of area and power, and determines which parameters have a significant impact on the efficiency of the network, and which parameters are unimportant. Finally, Section 5 presents our conclusions and future work.

2. BACKGROUND

This section describes clock networks on commercial FPGA's, as well as the way in which clock networks are modeled in previous academic studies.

2.1 Commercial Clock Networks

Current state-of-the-art FPGA's, such as the Altera Stratix II [3], the Actel ProASIC3 [4], and the Xilinx Virtex II Pro [5] devices, support multiple and local and global clock domains. In each of these devices, the FPGA is divided into four quadrants (see Figure 1). The Altera Stratix II provides 16 global clock signals, which can be connected to all the flip-flops on the FPGA, and 8 local clock networks in each of the four quadrants, which can be connected to all the flip-flops within the quadrant. Similarly, the Actel ProASIC3 devices provide 6 global clocks per quadrant and the Xilinx Virtex II Pro devices provide 16 global clock networks and 8 local clock networks. Unlike the Altera and Actel parts, however, the global clocks in the Virtex II Pro are not connected to flip-flops directly; instead, the global clocks drive local clocks within each quadrant.

Within a quadrant, the clocks are distributed to rows of logic blocks through a row multiplexer and rib routing channels. In the Stratix II devices, each row multiplexer chooses 6 clocks from the 24 local and global clocks, and provides them to all the flip-flops

in that row, as shown in Figure 2. In ProASIC3 devices, the row multiplexers choose from 6 global, 3 local, and several internal signals. In the Virtex II devices, the row multiplexers choose between 8 local clocks.

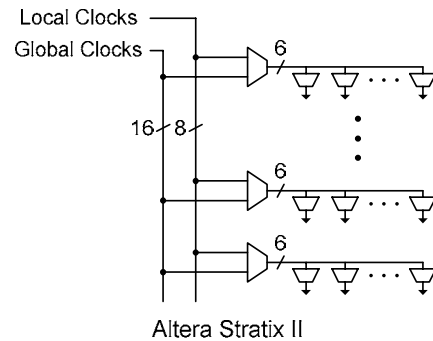


Figure 2: Multiplexer structure of Stratix II clock networks.

The circuitry that drives the clock networks is similar for each of the three devices. As shown in Figure 3, the local and global clock networks are driven by control blocks that select the clock signal and dynamically enables or disables the clock to reduce power consumption when the clock signal is not being used. The clock networks can be driven by an external source, an internal source, or by clock management circuitry which multiplies, divides and/or shifts an external source.

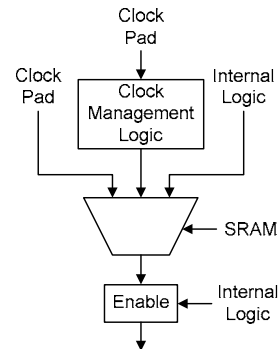


Figure 3: Control block for local and global clock signals.

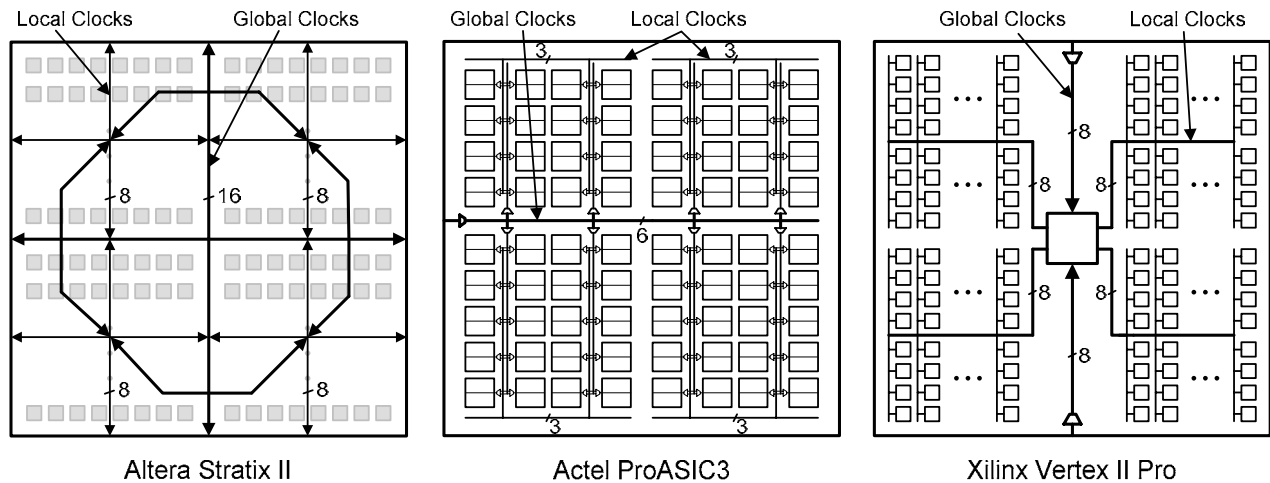


Figure 1: Commercial FPGA clock networks.

2.2 Previous FPGA Clock Network Research

The existing literature on FPGA's has mostly assumed simplified FPGA clock architectures. The study described in [6] examines the design of energy efficient FPGA's. The study focuses mostly on programmable logic and general routing resources and for clock networks the study proposes that edge triggered flip-flops are used within logic blocks to reduce the power dissipated by the clock network, since this reduces the switching rates by a factor of two. In [7,8], FPGA power models are described that assume simple H-tree clock networks with buffers inserted at the end of each branch. In both models, clock networks span the entire FPGA and that they are implemented using dedicated resources. In [8], additional buffers are inserted to separate long wire segments in order to optimize the clock delay. Neither model is parameterized since the intent was only to include the power dissipated by the clock networks to improve accuracy of their power estimates. In [2], a placement algorithm that minimizes clock skew is presented. Circuits with multiple clock domains are taken into account; however, the study focuses mainly on the algorithm rather than the design of the clock network.

2.3 Full Crossbar and Concentrator Networks

In this paper, we will employ both full crossbar and concentrator crossbar networks as building blocks. An $n \times m$ single-stage crossbar is a single stage network that connects n inputs to m outputs, as shown in Figure 4. Figure 4(a) shows a full crossbar network in which each output can be driven by any input. Such a crossbar requires $n \cdot m$ switches. Figure 4(b) shows a concentrator network, in which any m -element set of the n input signals can be mapped to the m outputs (without regard for the order of the outputs). A concentrator built this way contains $m \cdot (n - m + 1)$ switches [9]. Sparse crossbar concentrators are most efficient when m is close to n (ie. the network is close to square) or when m is very small (close to 1). A perfectly square crossbar concentrator only has n switches. As the crossbar concentrator becomes more rectangular, the number of switches approaches that of a full crossbar. Full and concentrator crossbars can also be implemented using fanin-based switches (multiplexers followed by buffers); however, this implementation is not considered in this paper.

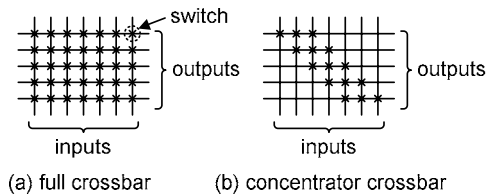


Figure 4: Two 7×5 crossbar networks.

3. PARAMETERIZED CLOCK NETWORK MODEL

In this section, we describe our parameterized clock network model. Such a model is an important precursor to any research on FPGA clock networks. The model describes a family of networks; in Section 4, we will compare the efficiency of members of this family. The model encompasses the salient features of all commercial FPGA clock networks.

Clock networks within our model use a three-stage clock distribution topology. The first stage programmably connects some number of clock sources (clock pads, PLL/DLL outputs, or internal

signals) to the center of some number of clock regions. The second stage programmably distributes the clocks to the logic blocks within each region. The third stage connects the clock inputs of each logic block to the flip-flops within that logic block. The following subsections discuss the parameters that describe the number and positioning of the regions, and then discusses the three network stages individually. A summary of all parameters appears in Table 1. Throughout, the parameters n_x and n_y will be used to denote the number of logic blocks in the X and Y directions, and N will be used to denote the number of logic elements per logic block (cluster). We assume there is one flip-flop per logic element; if this is not true, N can be redefined to be the number of flip-flops per logic block.

3.1 Clock Sources

The clock signals will typically be driven by selected outputs from phase-locked loops (PLL's) or delay-locked loops (DLL's), dedicated input pads, or signals from the FPGA core. In all cases, we assume the sources are distributed evenly around the periphery of the FPGA core. In our model, the number of potential clock sources is n_{source} , meaning there are $n_{source}/4$ potential clock sources on each side of the FPGA.

3.2 Global and Local Clock Regions

We assume that there are two types of clocks that the network must distribute: global and local clocks. Global clocks are distributed across the entire chip (but not necessarily to every logic block, as described in Section 3.3). Local clocks, on the other hand, are distributed to only a region of the FPGA. Although it is possible to implement a global clock by stitching together several local clocks, this would be an inefficient way to distribute a clock to the entire chip. By distributing the global clocks to each region using a dedicated network, lower skew for these clocks can be achieved.

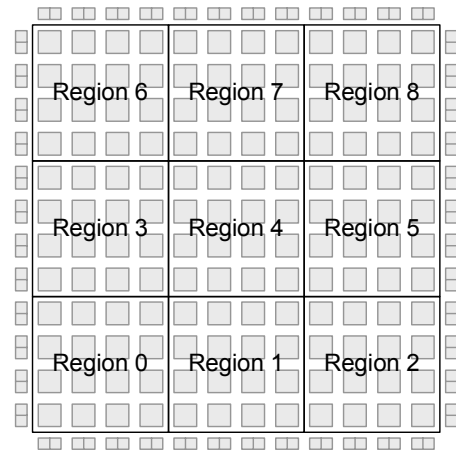


Figure 4: Example FPGA with 3×3 clock regions.

To support local clocks, the FPGA fabric is broken down into a number of regions, each of which can be driven by a different set of local clocks (the same local clock can also be connected to more than one region). The number of regions in the X dimension is denoted by n_{x_region} , and the number of regions in the Y dimension is denoted by n_{y_region} . The total number of regions is thus $n_{x_region} \cdot n_{y_region}$. In the commercial architectures in Figure 1, n_{x_region} and n_{y_region} are both 2. Figure 4 shows an example with both parameters equal to 3.

3.3 First Network Stage

The first stage of the clock network programmably connects the clock sources on the periphery of the chip to the center of each region.

The first stage consists of two parallel networks: one for the global clocks and one for the local clocks. We denote the total number of global clock signals as W_{global} . The global clock network selects $W_{global}/4$ signals from each of the $n_{source}/4$ potential clock sources on each side, as shown in Figure 5. This selection is done using a concentrator network on each side of the chip (see Section 2.3). The use of a concentrator network guarantees that any set of $W_{global}/4$ signals can be selected from the $n_{source}/4$ sources on each side. This architecture does not, however, guarantee that any set of W_{global} signals can be selected from the n_{source} potential sources; it would be impossible to select more than $W_{global}/4$ signals from any side. Relaxing this restriction would require an additional level of multiplexing (or larger concentrators and longer wires), which we do not include in our model.

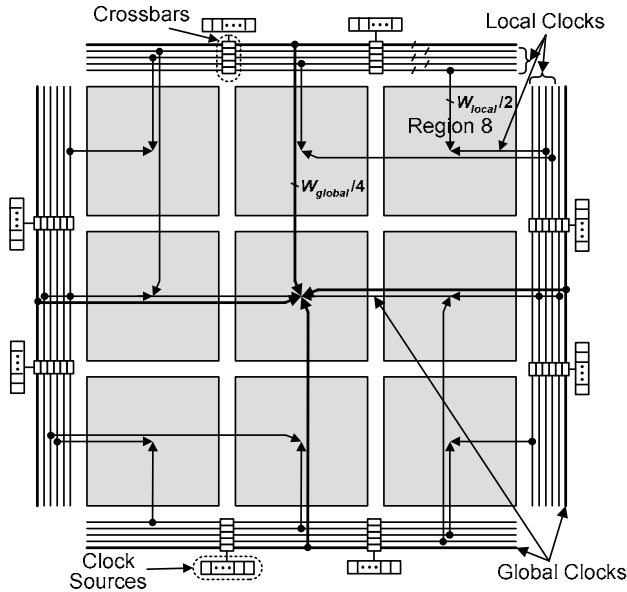


Figure 5: Connections from the periphery to the center of regions (local clocks) and of the FPGA (global clocks).

All W_{global} signals are routed on dedicated wires to the center of the chip. Since the sources all come from the periphery of the chip, the connection between the sources and the center of the chip will introduce little or no skew. In an architecture in which some sources come from inside the chip, the drivers can be sized such that skew is minimized. From the center of the chip, all W_{global} clocks are distributed to the center of all regions using an spine-and-ribs distribution network with n_{y_region} ribs, as shown in Figure 6. Although an H-tree topology would have a lower skew, it is difficult to mesh such a topology onto a tiled FPGA with an arbitrary number of rows and columns.

There is one local clock network per region. We denote the number of local clocks per region as W_{local} . The W_{local} signals are selected from the two sides of the FPGA that are closest to the re-

gion, as shown in Figure 5 (if the number of regions in either dimension is odd, the selection of the “closest” side is arbitrary for regions in the middle). Half of the W_{local} signals are selected from the sources on each of the two sides using a concentrator network. The use of a concentrator network guarantees that any set of $W_{local}/2$ signals can be selected from the $n_{source}/4$ potential sources on each side. Driver sizing can be used to minimize the skew among the clocks connected to each region. Skew between regions is not as important, since global clocks will likely be used if a clock is to drive multiple regions.

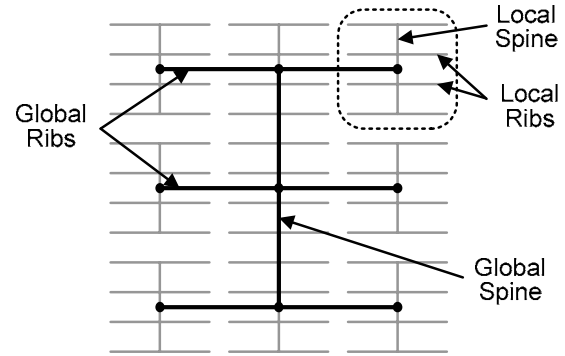


Figure 6: Global clock connections from the center of the FPGA to the center of regions.

3.4 Second Network Stage

The second network stage programmably connects clock signals from the center of each region to the logic blocks within that region. There is one of these networks for each region.

The input to each second stage network consists of W_{global} global clocks and W_{local} local clocks from the first stage described in Section 3.3. These clocks are distributed using a spine-and-ribs topology as shown in Figure 7. The spine contains $W_{global} + W_{local}$ wires. In each row, a concentrator network is used to select any set of W_{rib} clocks from the spine. These clocks are distributed to the logic blocks in that row through a local rib. Each logic block in the row connects to the rib through another concentrator network; the concentrator is used to select any set of W_{lb} clocks from the rib.

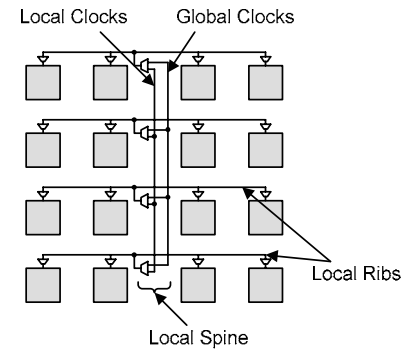


Figure 7: Second stage of the clock network.

3.5 Third Network Stage

The third network stage programmably connects the W_{lb} logic block clocks to the N logic elements within the logic block. This is illustrated in Figure 8. In order to provide flexibility to the

clustering tool, we assume that each the clock pin of each logic element can be driven by any of the W_{lb} logic block clocks.

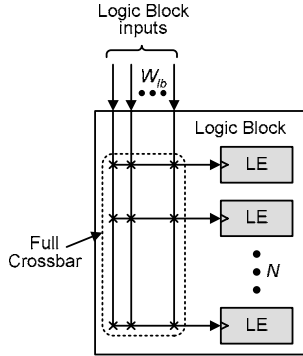


Figure 8: Third stage of the clock network.

Table 1 summarizes the parameters used to describe the FPGA fabric and the clock network.

Table 1: Clock network parameters.

	Parameter	Meaning	Range in paper
Parameters that describe the FPGA fabric	nx	Number of logic blocks in the X dimension	100
	ny	Number of logic blocks in the Y dimension	100
	N	Number of logic elements per logic block	10
Parameters that describe the FPGA clock network	nx_region	Number of clock regions in the X dimension	2 - 4
	ny_region	Number of clock regions in the Y dimension	2 - 4
	$nsource$	Number of potential clock sources (one quarter of these are on each side)	128
	W_{global}	Number of global clocks that can be routed to the fabric at one time	0-24
	W_{local}	Number of local clocks that can be routed to each region at one time	0-12
	W_{rib}	Number of clocks that can be routed to all logic blocks in a single row of a region at one time.	1-12
	W_{lb}	Number of clocks that can be connected to each logic block.	1-6

4 EXPERIMENTAL RESULTS

In this section, we measure the cost of flexibility of members of the clock network family described in Section 3. The cost is measured in terms of area and power, which are measured using the area and power models described in Section 4.1. For area and leakage power, the entire clock network is considered, regardless of how the network is used. For dynamic power; however, only

the parts of the clock network being used are considered since the unused parts are disabled.

In this section, we consider the impact of each parameter separately. In all cases, we use the baseline architecture described in Table 2 (which is similar to the commercial architectures described in Section 2) and vary one parameter at a time. Our goal is not to exhaustively measure the cost of every possible combination of parameters, but instead to determine the impact that each parameter has on the area and power of the clock network.

Table 2: Baseline architecture parameters

Parameter Name	Parameter Value
nx	100
ny	100
N	10
nx_region	2
ny_region	2
$nsources$	64
W_{global}	16
W_{local}	8
W_{rib}	6
W_{lb}	2

4.1 Area and Power Models

Detailed area and power models were constructed based on transistor-level layouts of the clock circuitry. The area model is a count of the Minimum Transistor Equivalents (MTE's) as described in [10]. The power model consists of terms for static power, dynamic switching power, and short-circuit power, and is similar to that described in [7]. Wiring capacitance, buffer input and output capacitances, and switch input and output capacitances were considered in the power model. Each wire in the clock network (spine, rib, and wire between the periphery and the center of the chip/region) was assumed to contain a repeater (20X minimum size) at each tile. Each switch in the clock network is assumed to consist of three cascaded buffers (1X, 5X, and 20X) followed by a transmission gate switch (consisting of a 20X NMOS transistor and a 1X PMOS transistor). The PMOS transistor acts as a level restorer and is sized to minimize area and power. Finally, switches between the intra-logic block clock lines and the logic elements were assumed to consist of a 5X buffer followed by a transmission gate switch. Experiments have shown that these layout assumptions work well over a range of architectural parameter assumptions. A 180nm TSMC process and a clock frequency of 200 MHz was assumed throughout.

4.2 Impact of W_{lb}

We first consider the flexibility within the logic blocks by varying W_{lb} , the number of clock inputs to each logic block. Intuitively, the larger this number, the simpler the task for the clustering algorithm (since the constraint on the number of clocks per logic block is not as severe), however, the larger W_{lb} the more power-hungry the clock network will be. In this section, we measure the cost of increasing W_{lb} .

We varied W_{lb} from 1 to 6 and set the remaining parameters as in Table 2. Figure 9 shows the impact on the area of each component of the clock network. As shown, the area due to clock buffers (repeaters) and the spine-to-rib connections is not affected. The area of the switches between the logic block clock wires and the logic elements increases significantly; recall that a full inter-

connect pattern was assumed within the logic block. The area due to the switches between the ribs and the logic blocks, increases as W_{lb} increases from 1 to 3, and then decreases as W_{lb} increases from 4 to 6. This is because the number of switches in a concentrator network is smallest when the number of outputs is either very small or close to the number of inputs. Because of this, the incremental area cost of increasing W_{lb} when W_{lb} is more than half of W_{rib} (the number of clocks in each rib) is small.

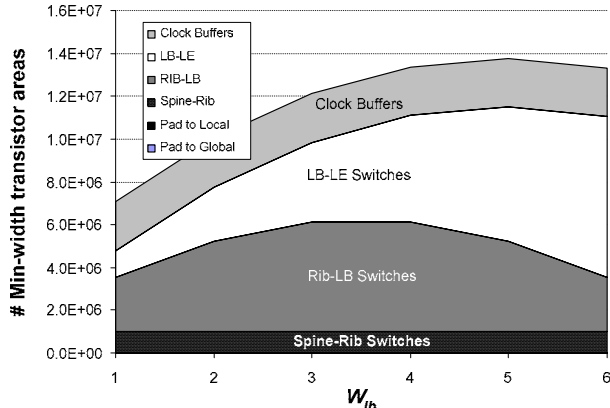


Figure 9: Area vs. W_{lb} .

Figure 10 shows the impact of W_{lb} on power. The power depends not only on the number of clocks available to each logic block, but also the number of these clocks that are actually used (unused clocks do not switch and do not consume dynamic power). In the graph, each solid line corresponds to a number of clocks that are used within each logic block and the dotted line on the graph connects the points in which all available clocks are being used; this line illustrates the worst-case cost of adding more clocks within each logic block. Clearly, the number of used clocks in a logic block can not be more than the number of clocks available in each logic block (W_{lb}).

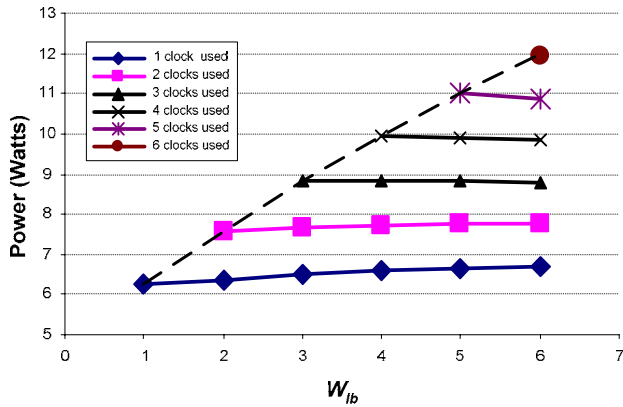


Figure 10: Power vs. W_{lb} .

The graph shows that the power of the clock network is far more affected by the number of clocks in each logic block that are used, than the number of clocks available in each logic block. This is important; it shows that FPGA vendors can choose a large value of W_{lb} (likely close to W_{rib} from the area results) to support those few applications that require this many clocks per logic block. For other applications which do not need this many clocks per logic block, the power impact of choosing such a large W_{lb} is minimal. We can also conclude that it would be beneficial to

expend extra effort in the cluster tool to reduce the number of used clocks within each logic block.

The graph also shows that increasing W_{lb} increases power when one or two clocks are used but decreases power otherwise. This effect is caused by a tradeoff between the power dissipated by LE clock wires and the power dissipated by LB clock wires. When one or two LB wires are used, increasing W_{lb} increases power since the increase in LB-LE switch power is greater than the decrease in Rib-LB switch power. When 3 or more LB wires are used, however, the reverse is true.

4.3 Impact of W_{rib}

We next consider the impact of varying the number of wires in each rib within a region. Intuitively, the placement tool has to ensure that the total number of clocks used by all logic blocks lying on a single rib is no larger than W_{rib} . The higher this value, the easier the placement task becomes, however, a larger value of W_{rib} means the clock network will be larger and consume more power. In this section, we measure the cost of increasing W_{rib} .

We used the parameter values from Table 2, and varied W_{rib} from 2 to 12. Although Section 4.2 shows that W_{lb} should be close to W_{rib} , here we keep W_{lb} constant at 2 to isolate the impact of changing W_{rib} . As shown in Figure 11, the area of the rib-to-logic block switches dominates the total clock area for large values of W_{rib} . This is primarily because the mismatch between W_{lb} and W_{rib} leads to larger concentrator networks, as described in Section 4.2. The number of spine-to-rib programmable connections also rise, however, this is a smaller contribution to the total area. Finally, the number of clock buffers increase as W_{rib} increases, since there are more rib wires that need repeaters.

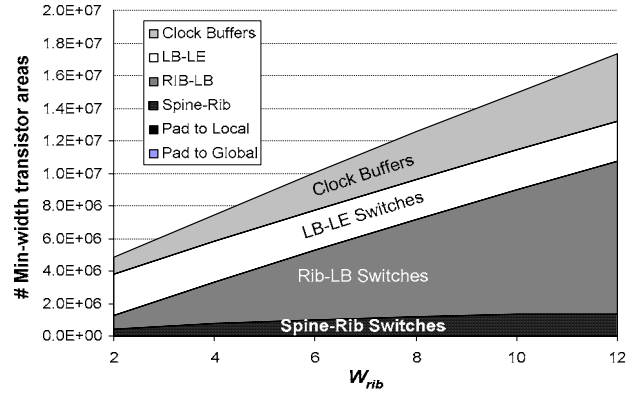


Figure 11: Area vs. W_{rib} .

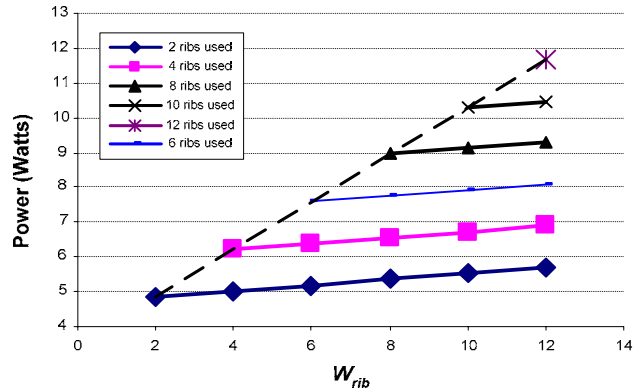


Figure 12: Power vs. W_{rib} .

The power results are shown in Figure 12. Again, the power depends not only on the available number of ribs, but the number of these ribs that are actually used. As was the case for the logic block clocks, the number of clocks in each rib has less impact on the total power than the number of these clocks that are actually used. Again, this implies that FPGA vendors should include as many rib wires as possible, since the extra rib wires have little impact on power for applications that do not use them. Unlike in Section 4.2, however, this must be balanced with the area cost of choosing a large W_{rib} .

4.4 Impact of the Spine Width

In this section, we consider the impact of changing the spine width ($W_{local} + W_{global}$) on area and power. Intuitively, a wider spine means more clocks can be distributed within a given region.

We used the parameter values from Table 2, and varied ($W_{local} + W_{global}$) from 6 to 36, keeping a 1:2 ratio between W_{local} and W_{global} (matching the baseline architecture in Table 2). Figure 13 shows the area results. The number of switches between the spine and the rib increases, however, this is a small component of the total area. The number of switches between the clock sources and the spine also increases, but this area is negligible, and not visible in Figure 13. Overall, the increase in area is not significant. The power results are shown in Figure 14. Again, the number of used wires in each spine has significantly more impact than the number of available wires.

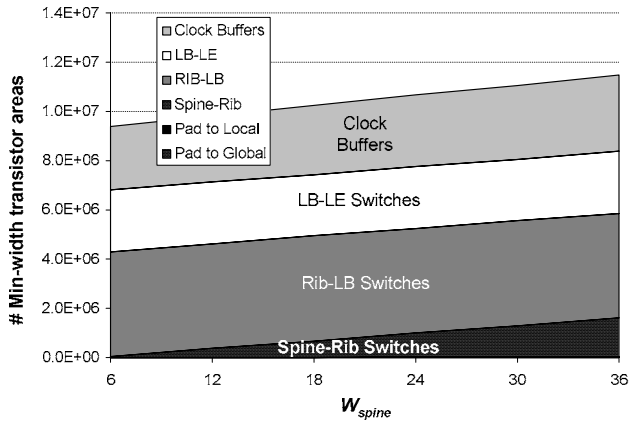


Figure 13: Area vs. W_{spine} .

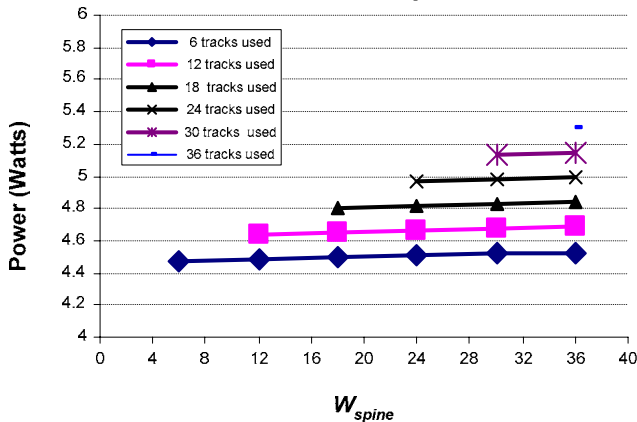


Figure 14: Power vs. W_{spine} .

4.5 Impact of the Number of Regions

As described in Section 3, we assume the FPGA has been broken down into $n_x_region * n_y_region$ regions, each of which contains its own set of local clocks. An architecture with many clock regions is suitable to implement applications with many clock domains. In the best case, we can map each domain of a user circuit to a single region of the FPGA. Intuitively, this would save power because each clock would only be routed to logic blocks within the region that uses it.

To investigate the area cost of increasing the number of regions, we focus on the design of an FPGA which is to support 64 clock domains. We consider three alternative architectures: one with one region, one with four regions (similar to the commercial architectures described in Section 2), and one with sixteen regions. Clearly, as we increase the number of regions and reduce their size, the number of clock domains of the user circuit that is mapped to each region decreases. In the ideal case, the number of clock domains mapped to each region, and hence the value of W_{local} , is inversely proportional to the number of regions. On the other hand, we would expect that increasing the number of regions has a smaller impact on the number of wires that should be included in each rib. Even though fewer regions leads to longer ribs, it would be unlikely that more than a few domains would be mapped to each row of a region. To reflect these observations, in this section, we assume that W_{local} is inversely proportional to the number of regions in the FPGA, and W_{rib} is two for all architectures. To isolate the impact of the local clocks within the regions, we further assume that W_{global} is 0. Finally, we assume that W_{lb} is one, although this choice has little impact on the conclusions drawn in this section.

Figure 15 shows the area of each of our three architectures. As the figure shows, the area of the FPGA drops as the number of regions increases. An FPGA with more regions requires fewer local clocks within each region, leading to a smaller spine within each region. This is countered by two effects. First, as the number of regions increases, the total length of the spine networks increases (in an architecture with 4x4 regions, there are 16 spines, each of which spans one quarter of the height of an FPGA). Second, as the number of regions increases, the size of the first level network (the network that connects the clock sources to the center of each region) increases. Neither of these effects is as significant as the reduction in the width of each spine, however.

The power consumed by each of our three architectures depends on the number of clocks that are actually being used in each domain. To estimate this, we assume a hypothetical user circuit with 100 000 logic elements. The number of clock domains in this circuit is varied from 1 to 64. In each case, we assumed all domains are of the same size. As we increase the number of domains, the number of logic elements per domain drops, as shown in Table 3. When the hypothetical circuit is implemented on each of the three architectures, the user domains are assumed to be mapped equally to the available regions. The number of local clocks (and hence W_{local}) used therefore depends on the number of domains in the circuit, as shown in Table 3. Within each region, the clock domains of the user circuit are evenly mapped to a subset of the rows within the region. In most cases, this means only one rib wire per row is used. For very large regions, or very small user domains, it may be necessary to share a row between two clock domains; in this case, two rib wires are used in that row. The numbers in Table 3 reflect the average number of rib wires that are used per row.

The power results are shown in Figure 16. When all 64 clocks are used, the architecture with 16 regions consumes far less power than the architecture with 1 or 4 regions. There are several reasons for this. The primary reason is that in the architecture with 16 regions, the total wire length of the spine is shorter, leading to a reduction in both metal capacitance and the number of repeaters on the spine wires. A secondary reason is that smaller regions have smaller ribs -- this reduces the chance that two different logic blocks within the same row will require different clocks. Although it is not evident from the graph, if the user circuit has only one clock domain, the FPGA with only one region performs slightly better; this is because the first-level network that connects the clock sources to the center of each region is smaller.

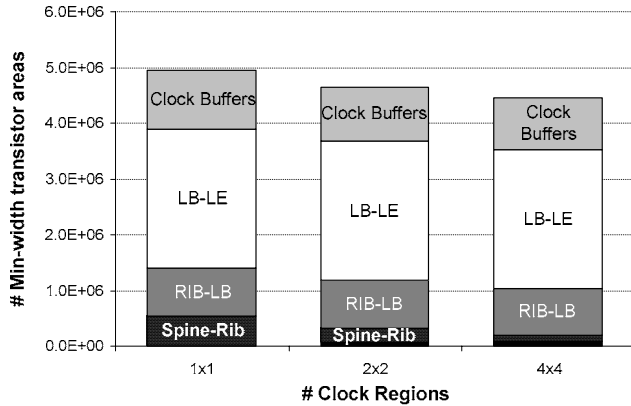


Figure 15: Area vs. number of clock regions.

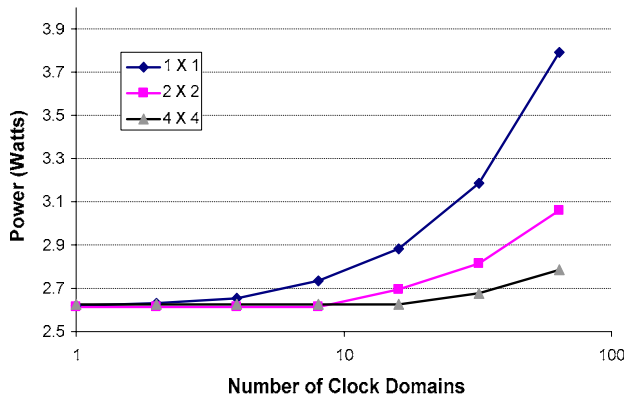


Figure 16: Power vs. number of clock regions.

Table 3: Lower bound clock resources usage.

# Clocks per Circuit	# LEs per Regions	1 Region (1x1)		4 Regions (2x2)		16 Regions (4x4)	
		W_{spine}	W_{rib}	W_{spine}	W_{rib}	W_{spine}	W_{rib}
1	100000	1	1	1	1	1	1
2	50000	2	1	1	1	1	1
4	25000	4	1	1	1	1	1
8	12500	8	1.04	2	1	1	1
16	6250	16	1.12	4	1.04	1	1
32	3125	32	1.28	8	1.12	2	1.04
64	1562 or 1563	64	1.60	16	1.28	4	1.12

5. CONCLUSIONS AND FUTURE WORK

In this paper we examined the tradeoffs between FPGA clock network flexibility, area, power dissipation. We described a new parameterized clock network model that describes a broad range of programmable clock network architectures with multiple local and global clock domains. Using the model, we then examined the cost of flexibility at various levels of the clock network.

At the logic blocks, we showed that the incremental area cost of increasing the number of logic block clock inputs (W_{lb}) is large when $W_{lb} < W_{rib}/2$ but is small when $W_{lb} > W_{rib}/2$. For power, we showed that the number of used clocks in each logic block is far more important than the number of clocks that are available. At the ribs, we showed that the area cost of increasing the number of clocks in each rib (W_{rib}) is large and dominated by the area of the rib-to-logic block switches. For power, the number of clocks that are used in each rib to be far more important than the number of clocks that are available. At the spines, we found that the area and power costs of increasing the number of clocks in each spine (W_{spine}) is not significant compared to that of increasing W_{lb} and W_{rib} . Finally, we showed that increasing the number of the clock regions generally decreases area and power by reducing the number of clocks required in each region.

As future work, an important part of this research will involve examining the impact that clustering and placement constraints imposed by clock network have on the implementation of applications. Another important issue to be considered is how our conclusions hold for smaller process technologies which tend to dissipate more leakage power.

6. REFERENCES

1. D.A. Hodges, H.G. Jackson, and R.A. Saleh, Analysis and design of digital integrated circuits: In deep submicron technology, 3rd edition, McGraw-Hill, New York, NY, 2004.
2. K. Zhu, D.F. Wong, Clock skew minimization during FPGA placement, ACM/IEEE Design Automation Conference (DAC), 1994, pp. 232-237.
3. Altera, Stratix II Device Handbook, Vol. 1, Chapter 2: Stratix II Architecture, March 2005.
4. Actel, ProASIC3 Flash Family FPGAs Device Architecture, July 2005.
5. Xilinx, Vertex-II Pro and Vertex-II Pro X Platform FPGAs: Complete Data Sheet, March 2005.
6. V. George, H. Zang, and J. Rabaey, The design of a low-energy FPGA, Intl. Symp. on Low-Power Electronic Design (ISLPED), 1999, pp. 188-193.
7. K.K.W Poon, S.J.E Wilton, A detailed power model for field-programmable gate arrays, ACM Transactions on Design Automation of Electronic Systems (TODAES), April 2005, Vol. 10, Issue 2, 2005, pp. 279-302.
8. F. Li, D. Chen, L. He, and J. Cong, Architecture evaluation for power-efficient FPGAs, Intl. Symp. on Field-Programmable Gate Arrays (FPGA), 2003, pp. 175-184.
9. S. Nakamura, G. M. Masson, Lower bounds on crosspoints in concentrators, IEEE Trans. on Computers, vol. 31, no. 12, pp. 1173-1178, 1982.
10. V. Betz, J. Rose, and A. Marquardt, Architecture and CAD for Deep-Submicron FPGAs, Kluwer Academic Publishers, 1999.