ABSTRACT

FPGAs are being increasingly used in a wide variety of applications. While power optimization has been only of secondary importance in many FPGA applications, growing importance of leakage in FPGAs designed in 90nm and below makes it imperative to treat power optimization as a first class citizen. In this paper, we propose a leakage-saving technique for FPGAs that involves dividing the FPGA fabric into small regions and switching on/off the power supply to each region using a sleep transistor in order to conserve leakage energy. Specifically, the regions not used by the placed design are supply gated. Next, we present a new placement strategy to increase the number of regions that can be supply gated. Finally, the supply gating technique is extended to exploit idleness in different parts of the same design during different time periods. Our experiments with different region sizes using various commercial and academic designs indicate that the proposed optimization outperforms conventional placement, and reduces leakage power consumption significantly.

Categories and Subject Descriptors

B.7.2 [Integrate Circuits]: Design Aids

General Terms

Design, Performance

Keywords

FPGA, leakage power, region-constrained placement.

1. INTRODUCTION

With the development of FPGAs in new technologies - 90nm and below, optimizing leakage power is becoming imperative. As the transistor feature sizes and threshold voltages reduce, and the number of transistors used in FPGAs increase, the overall leakage power is rapidly increasing. Consequently, the leakage problem is anticipated to be a major obstacle for FPGA applications in both high performance and low-power embedded designs. Due to this trend, there is a need to focus on leakage power optimizations going beyond prior power optimization techniques for FPGAs that focus primarily on reducing the dynamic energy [5, 7].

Reducing leakage power has already been the focus of optimization in various non-FPGA architectures. These optimizations have ranged from circuit to software approaches [2, 4, 15]. Among these techniques, a popular one to reduce both the subthreshold and gate leakage components is to switch off the power supply to the circuit by introducing a high-threshold voltage sleep transistor between the circuit and its supply rail. The sizing of this sleep transistor has an impact on both the performance and the area overheads imposed. Specifically, its sizing should be large for better performance. However, this increases both the area penalty and the ability to reduce leakage current (as wider transistors leak more). The optimal sizing of these sleep transistors has been the focus of prior efforts and the peak current required by the supply gated circuit serves as the reference for this sizing [6]. Since the peak current for different portions of a circuit do not normally occur simultaneously, prior work has used the approach of controlling a clustered group of circuits together with a single sleep transistor [6]. This optimization helps to reduce the area penalty as compared to using sleep transistors with each individual sub-circuit. It should be noted that sleep transistors can be used to control leakage in FPGAs as well. An obvious approach would be to place unused CLBs into low-power states using sleep transistors. However, such a fine-grain (at individual CLB level) power management of the FPGA fabric can introduce a significant area penalty, which may not be tolerable.

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

FPGA '04, February 22-24, 2004, Monterey, California, USA.

Copyright 2004 ACM 1-58113-829-6/04/0002 ...$5.00.
erated in many designs. Instead, in this paper, we propose a strategy, whereby the FPGA fabric is divided into regions, each of which can be independently controlled through a sleep transistor. A region is a rectangular array of CLBs, and is the minimum unit of power management. This approach is similar to the clustering technique mentioned in the paragraph above. Our experimental results indicate that area of the CLB arrays including the sleep transistor area overhead can be reduced by 5% when moving from unused regions with 4 logic slices to regions with 256 logic slices. By selecting a suitable region size, one can control the area overheads and at the same time achieve large leakage savings. Based on this region concept, we also propose a placement technique, referred to as the Region-Constrained Placement (RCP), that tries to use a minimum number of regions for a given application, thereby increasing the number of unused regions that can be switched off. A key observation from our results is that the leakage power savings obtained using RCP on an FPGA with coarse-grain regions is larger than that obtained using normal placement employed on an FPGA with fine-grain regions.

The maximum savings that can be obtained from the leakage management scheme discussed above is limited by the volume of the unused regions. Consequently, we also utilize a time-based control scheme that reduces leakage even in the utilized portions of the FPGA by switching off/on the power supply, exploiting the idleness in portions of the design. Specifically, the time-based scheme dynamically turns off power supply to all regions containing only idle modules. We investigate combinations of the time-based control scheme with two variants of RCP: (i) module-level RCP that places each module of the design that exhibits a distinct idleness profile using RCP individually and turns off power supply to all regions containing only idle modules, and (ii) design-level RCP that places the entire design using RCP and turns off power supply to all regions that contain only idle modules. Our experiments show that the time-based RCP scheme can provide additional energy savings as compared to statically switching off only unused portions.

The remainder of this paper is organized as follows. In Section 2, we revise the related work, focusing in particular on power optimizations for FPGAs. In Section 3, we discuss our region-based FPGA architecture. Sections 4 and 5 give details of the base RCP and the time-based control schemes with RCP. Section 6 describes the experimental methodology we used, and Section 7 presents experimental results. Section 8 concludes the paper.

2. RELATED WORK

Most of the previous works on power modeling, estimation and reduction in FPGA have focused primarily on dynamic power. In [7], the dynamic power of a Xilinx XC4003A FPGA was analyzed by taking measurements of test designs. [11] analyzed dynamic power consumption in Virtex-II FPGA family. [9] presented a flexible power model for FPGAs and evaluated different FPGA architectures for power efficiency. [12] presented a routability-driven bottom-up clustering technique for area and power reduction in clustered FPGAs.

Leakage in FPGAs has captured interest only very recently. [8] presents an FPGA architecture evaluation framework for power efficiency analysis of LUT-based FPGA architectures, and predicts leakage power to be dominant for future technologies. [14] makes a detailed analysis of leakage power in Xilinx CLBs. It concludes that significant reduction of FPGA leakage is needed to enable the use of FPGAs in mobile applications. [1] presents a fine-grained leakage control scheme using sleep transistors at gate level. Fine-grained control has the disadvantage of having to deal with sneak paths, which [1] has tried to minimize using some intelligent techniques. [3] presents a low-power high-level synthesis system for FPGA architectures.

3. ARCHITECTURE

We have chosen the island-style SRAM-based FPGA architecture for our study. Xilinx and Altera’s FPGAs fall into this category, and all our experiments were performed on Xilinx Virtex-II FPGAs. Note that this FPGA represents the current state-of-the-art, and therefore, is a suitable target for our optimization. The basic logic element in a Virtex-II is called a slice. A slice consists of 2 LUTs, 2 flip-flops, fast carry logic, and some wide MUXes [17]. A CLB in turn consists of 4 slices and an interconnect switch matrix. The FPGA contains an array of such CLBs along with block RAMs (BRAMs), multipliers and IO blocks as depicted in Figure 1.

The leakage distribution in a Xilinx FPGA in 90nm technology, with the exception of the BRAMs and multipliers, was shown to be 38% in the configuration SRAMs, 34% in the interconnect matrix, 16% in LUTs and 12% in other logic [14]. Since many of the techniques proposed for saving leakage energy in on-chip memory can be applied to BRAMs (and because they are not used by most of our designs), our leakage optimizations in this paper do not target them. In order to reduce the leakage energy in the configuration SRAM, we increased the threshold voltage of the configuration SRAM to obtain a 98% reduction in leakage energy while increasing configuration time by 20%. Since configuration time is not critical in most of our target designs, this tradeoff for power savings is reasonable. The resulting leakage breakdown in our system is shown in Figure 2. The focus of this work is on reducing the leakage energy in the LUTs, arithmetic logic and flip flops that account for 45% of the total leakage energy. While our work focuses on the slices,
the technique can be extended to switching off the routing resources as well. This is a part of our planned future work.

In order to provide leakage control, the FPGA is divided into regions. A region consists of one or more neighboring slices (potentially across different CLBs), and is the minimum power management unit (granularity). Sleep transistors are embedded into the FPGA fabric controlling the power supply to the individual regions. In this architecture, the control bit for the power switch (See Figure 1) of the region determines whether the region is supply gated or not. The control bits of the different regions are set during the configuration of the FPGA. The area overhead associated with the control bits (and the associated wiring) is proportional to the number of regions, while their impact on leakage energy is relatively small due to the use of high threshold transistors for the configuration bits. Thus, the area overhead favors a smaller number of large regions.

An important issue in the design of this architecture is the sizing of the power switches. The power switches should be large enough to support the peak current requirements of the logic slices that it controls to have negligible impact on performance. Since the peak current for a larger region is less than the sum of the peak currents of smaller regions constituting the larger region, it is possible to have a smaller area overhead when moving to larger regions with similar performance. In order to show this impact, we experimented with two different region sizes of 256 slices and 4 slices using XPower [17]. A single region of 256 slices had a peak current that was 68% of the sum of peak currents of 64 regions each of 4 slices constituting the same area. Next, we performed SPICE simulations to estimate the sleep transistor size for various region sizes. Assuming a slice area of 5000 sq. micron (from custom layout), it was estimated that the area penalty for a region size of 4 slices was around 15%, while that for 256 was 10%. This motivates the need for using large region sizes.

The amount of leakage reduction due to the introduction of the power switch is also influenced by the sizing and threshold voltage of the sleep transistor and whether a PMOS or NMOS transistor is used to gate the $V_{DD}$ or ground power supply rail. The leakage reduction varies from 85-98% based on these factors, incurring performance degradation varying from 0-30% [10]. In our experiments, we use a PMOS gate switch providing 90% leakage reduction.

4. RCP: REGION-CONSTRAINED PLACEMENT

The placement of the design has a significant impact on the ability to supply gate the logic slices in our region-based architecture. Employing the PAR tool in the normal design flow due to lack of region concept tends to scatter the utilized slices across different regions (See Figure 4(a)). Since the regions with partially used slices cannot be supply gated, the potential for leakage energy savings reduces. Thus, we propose a new region constrained placement strategy, RCP, that takes into account the region concept explicitly.

The basic principle of RCP is to constrain the placement of the design to specific regions of the FPGA (See Figure 4(b)) and leave some regions of the FPGA completely unused, so that they can be supply gated. This in turn helps to maximize the potential leakage savings. In our implementation of RCP, we place the design into contiguous regions to the extent possible and utilize two different styles: horizontal and vertical placements as shown in Figure 3. While the horizontal and vertical placements utilize the same number of logic slices, they do not provide similar performance results due to asymmetry in the target Virtex-II architecture. For example, there are fast carry chains running vertically in the FPGA, but not horizontally. Furthermore, there are more slices in a column than in a row in all Virtex-II parts (except XC2V40, which has 16 slices in both directions). While we confine the utilization of logic slices to specified regions, in order to circumvent issues with routing congestion, routing of IO signals and unroutability; the constraints on routing resources are kept as “soft”. This permits the use of routing resources outside the regions that have logic placed in them. As part of our future work, we plan to investigate a supply gating mechanism that also switches off interconnect muxes.

5. COMBINING RCP AND TIME-BASED CONTROL

It should be observed that RCP is essentially a static technique where the unutilized FPGA space (regions) can be shut off at configuration time (before the execution). While it is easy to implement, it may not be as effective in designs that occupy large portion of the FPGA space (which in turn limits the potential leakage savings). However, for the designs with modules that remain inactive over signifi-
significant durations of time, we can employ a time-based control scheme that reduces leakage even in the utilized portions of the FPGA by switching off/on the power supply, exploiting the idleness in portions of the design. Specifically, the time-based scheme turns off power supply to all regions containing only idle modules. We investigate combinations of the time-based control scheme with two variants of RCP: (i) module-level RCP that places each module of the design that exhibits a distinct idleness profile using RCP individually, and turns off power supply to all regions containing only idle modules, and (ii) design-level RCP that places the entire design using RCP and turns off power supply to all regions that contain only idle modules.

We can implement the idea of time-based control as follows. The gate voltage of a sleep transistor is still controlled by a configuration bit. However, instead of configuring this bit statically when the design is loaded on the FPGA, dynamic reconfiguration [16] of these control bits is used to switch a sleep transistor on or off. In order to limit the overhead of reconfiguring these control bits, the sleep transistor should not change state very frequently. Furthermore, support for just reconfiguring these control bits may be useful as opposed to the minimum reconfigurable block in current Virtex-II technology, which is a frame [16]. Reconfiguration time for one frame varies from 2\(\mu\) seconds for smallest to 23\(\mu\) seconds for the largest FPGA. However, support for reconfiguring only the sleep transistor configuration bits can reduce this time, but may increase area overheads due to the configuration circuit.

With increasing FPGA sizes, it’s possible to envision an entire system on FPGA. In such designs, many parts of the design may remain inactive for long durations. Time-based control seems to be a very promising approach for such designs. Figure 4(c) shows an example design placement using module-level RCP for time-based leakage control. We see from this figure that modules of the design get placed on non-overlapping regions, thus maximizing the number of regions that can be dynamically switched-off. Note that this slightly decreases the statically unused portion on the FPGA (because in order to ensure the inter-module region exclusivity needed for module-level RCP, some regions can only be partially filled). Still, our experiments show a significant increase in leakage savings due to module-level RCP.

6. EXPERIMENTATION

In order to investigate the energy savings due to the proposed approach, we selected a set of applications and used the Xilinx Virtex-II FPGA as our target hardware. The selected applications include 14 publicly available reference designs provided by Xilinx, 4 designs from ITC’99 benchmark suite, 3 academic designs and 14 commercial designs available internally at Xilinx. Table 1 provides the important characteristics of each application and lists the number of slices, IO blocks (IOBs), block RAMs (BRAMs) and multipliers (MULTs) used in the designs along with the target FPGA device used for the mapping. Note that on an average only 62% of the slices were used. Industry6 is an extreme case, where although only 4% of the slices are used; but due to the I/O requirements of the design, it cannot be mapped to a smaller FPGA.

These designs were then implemented using the experimental flow illustrated in Figure 5 to evaluate the energy savings possible due to the proposed optimizations. The specific steps in this design flow are elaborated below:

All the designs were synthesized for area-optimization from their HDL representation using the Xilinx Synthesis Technology(XST). This synthesis step produced a gate-level netlist. Next, the designs were mapped on to the smallest possible Virtex-II FPGA device, setting the place and route effort level high (level 5). After the mapping and completion of place and route (PAR), an NCD file that contains the placed and routed design is generated. The map process also generates a MAP report which is used to implement RCP. The maximum clock frequency for the design was estimated by using the post-PAR static timing analysis tool, TRACE on the mapped design. The NCD file was translated to an ASCII file in XDL format using the xd1 tool. This ASCII file was processed using a customized tool developed for this project to determine the unused regions of the FPGA given the region sizes. Using this information, the leakage savings possible in the standard placement process was obtained (assuming that the regions that are completely unused are switched off).

In order to determine the leakage savings using RCP, the synthesized gate-level (NGC file) was re-used. The MAP report from the normal mapping was used to determine the number of logic slices used in the design. Based on the number of slices obtained and the size of the regions, a User Con-
Table 1: Characteristics of benchmark designs

<table>
<thead>
<tr>
<th>Design</th>
<th>#Slices</th>
<th>#IOBs</th>
<th>#BRAMs</th>
<th>#MULTs</th>
<th>FPGA device</th>
</tr>
</thead>
<tbody>
<tr>
<td>xapp248</td>
<td>96 (37%)</td>
<td>17 (19%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>xapp270\des</td>
<td>4,723 (92%)</td>
<td>189 (58%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg456</td>
</tr>
<tr>
<td>xapp270\triple-des</td>
<td>14,273 (99%)</td>
<td>301 (62%)</td>
<td>0</td>
<td>0</td>
<td>XC2V3000fg676</td>
</tr>
<tr>
<td>xapp288\seg\encoder</td>
<td>50 (19%)</td>
<td>20 (22%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>xapp288\ps\encoder</td>
<td>107 (41%)</td>
<td>28 (31%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>xapp289</td>
<td>614 (39%)</td>
<td>166 (83%)</td>
<td>0</td>
<td>0</td>
<td>XC2V250-fg456</td>
</tr>
<tr>
<td>xapp298</td>
<td>70 (27%)</td>
<td>16 (18%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>xapp299</td>
<td>973 (31%)</td>
<td>326 (99%)</td>
<td>1 (2%)</td>
<td>0</td>
<td>XC2V500-fg456</td>
</tr>
<tr>
<td>xapp610</td>
<td>1,369 (89%)</td>
<td>20 (21%)</td>
<td>8 (33%)</td>
<td>0</td>
<td>XC2V250-cs144</td>
</tr>
<tr>
<td>xapp611</td>
<td>1,534 (99%)</td>
<td>20 (21%)</td>
<td>16 (66%)</td>
<td>0</td>
<td>XC2V250-cs144</td>
</tr>
<tr>
<td>xapp615</td>
<td>1,155 (75%)</td>
<td>45 (48%)</td>
<td>0</td>
<td>2 (8%)</td>
<td>XC2V250-cs144</td>
</tr>
<tr>
<td>xapp621</td>
<td>1,305 (84%)</td>
<td>29 (31%)</td>
<td>0</td>
<td>0</td>
<td>XC2V250-cs144</td>
</tr>
<tr>
<td>xapp625\video</td>
<td>254 (99%)</td>
<td>63 (71%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>xapp645</td>
<td>550 (10%)</td>
<td>278 (85%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg456</td>
</tr>
<tr>
<td>itc99\b04</td>
<td>110 (42%)</td>
<td>21 (23%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>itc99\b05</td>
<td>290 (50%)</td>
<td>39 (42%)</td>
<td>0</td>
<td>0</td>
<td>XC2V80-cs144</td>
</tr>
<tr>
<td>itc99\b12</td>
<td>214 (83%)</td>
<td>13 (13%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-cs144</td>
</tr>
<tr>
<td>itc99\b14</td>
<td>2,432 (79%)</td>
<td>88 (51%)</td>
<td>0</td>
<td>2 (6%)</td>
<td>XC2V500-fg256</td>
</tr>
<tr>
<td>ava\k4</td>
<td>724 (47%)</td>
<td>85 (92%)</td>
<td>0</td>
<td>0</td>
<td>XC2V250-cs144</td>
</tr>
<tr>
<td>ava\k5</td>
<td>2,034 (66%)</td>
<td>85 (49%)</td>
<td>0</td>
<td>0</td>
<td>XC2V500-fg256</td>
</tr>
<tr>
<td>ava\k9</td>
<td>2,895 (94%)</td>
<td>84 (49%)</td>
<td>0</td>
<td>0</td>
<td>XC2V500-fg256</td>
</tr>
<tr>
<td>industry1</td>
<td>1,954 (38%)</td>
<td>279 (64%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry2</td>
<td>2,488 (80%)</td>
<td>185 (70%)</td>
<td>0</td>
<td>0</td>
<td>XC2V500-fg456</td>
</tr>
<tr>
<td>industry3</td>
<td>2,513 (81%)</td>
<td>132 (50%)</td>
<td>0</td>
<td>0</td>
<td>XC2V500-fg456</td>
</tr>
<tr>
<td>industry4</td>
<td>5,777 (75%)</td>
<td>182 (34%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1500-fg896</td>
</tr>
<tr>
<td>industry5</td>
<td>5,153 (67%)</td>
<td>65 (12%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1500-fg896</td>
</tr>
<tr>
<td>industry6</td>
<td>206 (44%)</td>
<td>287 (66%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry7</td>
<td>3,505 (68%)</td>
<td>251 (58%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry8</td>
<td>1,602 (52%)</td>
<td>60 (22%)</td>
<td>0</td>
<td>0</td>
<td>XC2V500-fg456</td>
</tr>
<tr>
<td>industry9</td>
<td>2,280 (44%)</td>
<td>293 (67%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry10</td>
<td>3,663 (71%)</td>
<td>224 (51%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry11</td>
<td>4,364 (85%)</td>
<td>172 (40%)</td>
<td>0</td>
<td>0</td>
<td>XC2V1000-fg896</td>
</tr>
<tr>
<td>industry12</td>
<td>97 (31%)</td>
<td>80 (90%)</td>
<td>0</td>
<td>0</td>
<td>XC2V40-fg256</td>
</tr>
<tr>
<td>industry13</td>
<td>411 (80%)</td>
<td>84 (70%)</td>
<td>0</td>
<td>0</td>
<td>XC2V80-fg256</td>
</tr>
<tr>
<td>industry14</td>
<td>1,288 (83%)</td>
<td>186 (93%)</td>
<td>2 (8%)</td>
<td>0</td>
<td>XC2V250-fg456</td>
</tr>
<tr>
<td>Average</td>
<td>61.89%</td>
<td>50.82%</td>
<td>0.29%</td>
<td>3.23%</td>
<td>-</td>
</tr>
</tbody>
</table>

6.1 Time-based leakage control

The experiments for time-based leakage control were performed using an academic design implementing an Adaptive Viterbi Algorithm (AVA) decoder [13]. The design consists of 3 AVA decoders of varying constraint lengths (4, 6, and 9). Different decoders are selected depending on the noise levels in the transmission channel. If the noise level is high, then the decoder with a larger constraint length is selected. In [13], the authors utilize reconfiguration to switch between decoders of different constraint lengths. We modified the design by statically mapping 3 different sizes of decoders on the FPGA, and selecting the right decoder depending on noise in the channel. For this work, we assumed that an input coming into the FPGA decides which decoder to choose. The design was mapped onto an XC2V1500-hg575. The resource usage was 5469 slices (71%), 90 IOBs (22%), 0 BRAM choices were explored. Further, as explained earlier, two styles of RCP: horizontal and vertical were explored. Thus, a total of 32 different varieties of RCP were explored.
and 0 multiplier. The three different decoders occupied 718, 1846, and 2854 slices respectively. Another module, which remained active all the time (branch metric generator) occupied 51 slices. The advantage of this design is that the decoding can be done much more rapidly if the channel is not noisy. The drawback is that at any given time, two decoders are sitting idle. This gives a scope for switching-off the unused decoders. We estimated and compared the leakage savings for this design for design-level RCP, module-level RCP and normal placement, assuming run-time leakage control. We also compared savings obtained from run-time leakage control with static control. In order to estimate the leakage savings from run-time control, we assumed that each of the 3 decoders is active for equal durations. Thus, any of the three decoders can be switched-off for two-thirds of the total time.

7. RESULTS AND ANALYSIS

Figure 6 plots the average estimated leakage power savings by switching off the unused regions in FPGA. The savings are represented as percent of total leakage (that occurs without any switching off). A region represented as 2_4 means that the region is 2 slices wide and 4 slices high. Plots for RCP as well as without RCP have been shown. For both, RCP and normal placement, leakage savings decrease with increase in region size. But, the decrease for RCP is very small compared to normal placement. As is evident from the plots, RCP clearly outperforms normal placement. Especially for large region sizes, RCP provides more than 6 times the savings of normal placement. This happens because, although the number of slices used is the same in both cases; in case of normal placement, they are scattered across regions. Larger regions can accentuate this problem.

We observed that the leakage power savings are strongly dependent on the resource usage of a design. Figure 7 plots...
the variation, across all designs, of leakage power savings for a single region choice. It shows that the leakage power savings vary significantly depending on the design. For some designs, there is no leakage saving because those designs occupy all the regions of the FPGA. Leakage power is reduced by more than 20% for 40% of the designs.

However, the constraint on the placement due to RCP can influence the timing of the signals. Figure 8 plots the average estimated clock frequencies achieved using RCP, expressed as a percentage of frequency estimated for normal placement. A region represented as $2_4^h$ refers to horizontal style of RCP with region of height 4 slices and width 2 slices. Similarly, a region represented as $2_4^v$ refers to vertical style with the same region size and shape. The plot shows that for all regions, the average clock frequency is within 8% of original clock frequency.

The performance penalty can result in longer execution time and consequently increase the duration of leakage. To capture this impact, Figure 9 plots average estimated leakage energy savings for RCP as well as for normal placement. We note that except for very fine-grain regions, RCP always results in higher leakage energy savings. The difference between the two increases for large region sizes. Again note that small region size incurs larger area overhead due to larger effective sleep transistor size, more routing and control signals, and more configuration bits (which increases configuration time too).

### 7.1 Time-based Leakage Control

Figure 10 plots leakage power savings for dynamic and static leakage controls for the AVA decoder design. The savings from dynamic control are shown for a module level RCP (modules get placed in non-overlapping regions), design level RCP, and for normal placement. The savings from static leakage control are shown for design-level RCP, and for normal placement.

It is observed that time-based leakage control results in very large savings compared to static control. Further, among the different placement strategies for time-based control, module-level RCP outperforms the others. Design-level RCP performs better than normal placement in most cases, but in some cases normal placement results in larger savings. This happens because in case of normal placement, the 3 different modules are placed slightly separated (because the placer has a larger area available to place the modules). Therefore, only a few regions are common among the different modules. In case of design-level RCP, the placer has a smaller area in which to fit the entire design. This increases the overlap among the 3 modules, thus disabling the dynamic switching-off of those regions.

Figure 11 plots leakage energy savings for time-based control (module-level RCP, design-level RCP, and normal placement with no RCP) and static control (RCP, normal placement with no RCP) for the AVA decoder design. It is observed that time-based control results in very large savings compared to static control. Also, in all but two cases, module-level RCP results in the largest energy savings.

It must be observed that the plots shown above do not account for additional overhead for dynamic reconfiguration of the control bits. However, even assuming that recon-
configuration incurs a 10% increase in overall execution time and consequent leakage energy penalty; we find that module-level RCP with time-based leakage control provides 19% (is 27% without reconfiguration overhead) more leakage savings than a normal placement with static leakage control.

8. CONCLUSION AND FUTURE WORK

Our work demonstrates that switching off parts of FPGA can result in significant leakage savings in most designs. The savings can be further increased by using Region Constrained Placement (RCP). Further, if RCP is used then the switch-off granularity need not be very fine, since leakage savings decrease very gradually with increasing region size. Thus, considering the area overhead of having very small regions, a large region size coupled with RCP looks to be a practical choice. Module-level RCP is a promising enhancement for designs in which some modules stay inactive for significant durations of time.

The present work calls for some interesting future work. First of all, we plan to extend our work to switch-off interconnect multiplexers in addition to logic slices in the CLB array. Further, in this paper, we have discussed an architecture, in which all regions are of the same size. A hybrid architecture, in which regions are of different sizes can be studied. This will increase the complexity of the placer, but it may increase the leakage savings. We also want to consider the use of multi-$V_{DD}$ and multi-$V_T$ regions to reduce energy consumption further while impacting performance minimally.

9. ACKNOWLEDGEMENTS

We would like to to thank Russell Tessier for providing us with the VHDL code of the AVA decoder design.

This work was supported in part by NSF CAREER Awards 0093082 and 0093085, and a grant from MARCO/GSRC-PAS.

10. REFERENCES


