Low-Power Programmable Routing Circuitry for FPGAs

Jason H. Anderson
Department of ECE
University of Toronto
Toronto, Ontario, Canada
janders@eecg.utoronto.ca

Farid N. Najm
Department of ECE
University of Toronto
Toronto, Ontario, Canada
f.najm@utoronto.ca

ABSTRACT

We propose two new FPGA routing switch designs that are programmable to operate in three different modes: high-speed, low-power or sleep. High-speed mode provides similar power and performance to a traditional routing switch. In low-power mode, speed is curtailed in order to reduce power consumption. Our first switch design reduces leakage power consumption by 36-40% in low-power vs. high-speed mode (on average); dynamic power is reduced by up to 28%. Leakage power in sleep mode is 61% lower than in high-speed mode. A second switch design offers a 36% smaller area overhead and reduces leakage by 28-30% in low-power vs. high-speed mode. The proposed switch designs require only minor changes to a traditional routing switch, making them easy to incorporate into current FPGA interconnect. The applicability of the new switches is motivated through an analysis of timing slack in industrial FPGA designs. Specifically, we show that a considerable fraction of routing switches may be slowed down (operate in low-power mode), without impacting overall design performance.

1. INTRODUCTION

Technology scaling trends have made power consumption, specifically leakage power, a major concern of the semiconductor industry [1]. Continued improvements in the speed, density and cost of field-programmable gate arrays (FPGAs) make them a viable alternative to custom ASICs for digitial circuit implementation. However, the ability to program and re-program FPGAs involves considerable hardware overhead and consequently, an FPGA implementation of a given circuit design is less power-efficient than a custom ASIC implementation [8]. Traditionally, research on FPGA CAD and architecture has centered on area-efficiency and performance. Low-power is likely to be a key objective in the design of future FPGAs.

A number of recent studies have considered the breakdown of power consumption in FPGAs, and have shown that 60-70% of dynamic and static (leakage) power is dissipated in the interconnection fabric [20, 21, 14, 18, 17]. Interconnect dominates dynamic power in FPGAs due to the composition of the interconnect structures, which consist of prefabricated wire segments with used and unused switches attached to each segment. Wirelengths in FPGAs are generally longer than in ASICs due to the silicon area consumed by SRAM configuration cells and other configuration circuitry. FPGA interconnect thus presents a high capacitive load, representing a considerable source of dynamic power dissipation.

Subthreshold and *gate oxide* leakage are the dominant leakage mechanisms in modern ICs and both have increased significantly in recent technology generations. Subthreshold leakage current flows between the source and drain terminals of an OFF MOS transistor. It increases exponentially as transistor threshold voltage (V_{TH}) is reduced to mitigate performance loss at lower supply voltages. Gate oxide leakage is due to a tunneling current through the gate oxide of an MOS transistor. It increases exponentially as oxides are thinned, which is done to improve transistor drive strength in modern IC processes. Both forms of leakage generally increase in proportion to transistor width, and the programmable interconnection fabric accounts for the majority of transistor width in FPGAs [18].

Prior work on leakage optimization in ASICs differentiates between *active* and *sleep* (or *standby*) leakage. Sleep leakage is that dissipated in circuit blocks that are temporarily inactive and have been placed into a special "sleep state", in which leakage power is minimized. Active leakage is that dissipated in circuit blocks that are in use ("awake"). Note that unlike ASICs, a design implemented on an FPGA uses only a portion of the underlying FPGA hardware and that leakage is dissipated in both the *used* and the *unused* part of the FPGA. Today's FPGAs do not offer sleep support and thus, it is valuable to consider FPGA circuit structures that can reduce *both* active and sleep leakage.

The dominance of interconnect in total FPGA power consumption makes it a high-leverage target for power optimization. In this paper, we present two novel FPGA routing switch designs that offer reduced leakage and dynamic power dissipation. The designs can be programmed to operate in one of three modes: high-speed, low-power or sleep mode. In high-speed mode, power and performance characteristics are similar to those of current FPGA routing switches. Low-power mode offers reduced leakage and dynamic power, albeit at the expense of speed performance. Sleep mode, which is suitable for unused switches, offers leakage reductions significantly beyond those available in low-power mode. The remainder of the paper is organized as follows: Section 2 presents related work and necessary background material. The proposed switch designs are described in Section 3. Section 4 analyzes the timing slack present in industrial FPGA designs implemented in the Xilinx Spartan-3 commercial FPGA [23] (a 90nm FPGA), and demonstrates that a large fraction of routing switches may operate in low-power mode, without compromising overall circuit performance. Experimental results are given in Section 5. Conclusions are offered in Section 6.

0-7803-8702-3/04/$20.00 ©2004 IEEE.

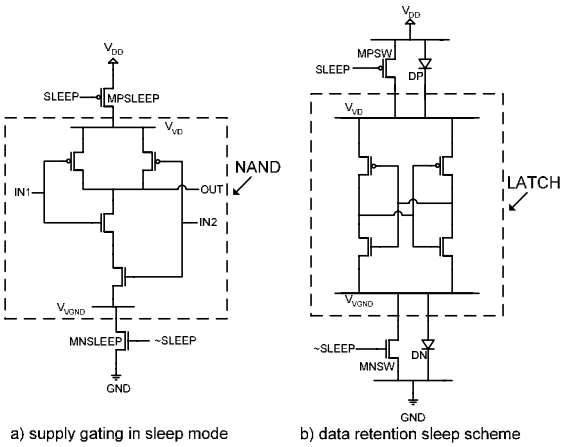

Figure 1: Sleep leakage reduction techniques [4, 11].

2. BACKGROUND AND RELATED WORK

2.1 Leakage Power Optimization

A variety of techniques for leakage optimization in ASICs have been proposed in the literature; a detailed overview can be found in [19]. Our proposed switch designs draw upon ideas from two previously published techniques for sleep leakage reduction, briefly reviewed here. The first is to introduce sleep transistors into the N-network (and/or P-network) of CMOS gates [4], as shown in Fig. 1(a). Sleep transistors ($MPSLEEP$ and $MNSLEEP$) are ON when the circuit is active and are turned OFF when the circuit is in sleep mode, effectively limiting the leakage current from supply to ground. A limitation of this approach is that in sleep mode, internal voltages in sleeping gates are not well-defined and therefore, the technique cannot be directly applied to data storage elements.

A way of dealing with the data retention issue was proposed in [11] and is shown in Fig. 1(b). Two diodes, DP and DN, are introduced in parallel with the sleep transistors. In active mode, the virtual V_{DD} voltage (V_{VD}) and the virtual ground voltage (V_{VGND}) are equal to rail V_{DD} and GND, respectively. In sleep mode, the sleep transistors are turned OFF and $V_{VD} \approx V_{DD} - V_{DP}$, where V_{DP} is the built-in potential of diode DP. Likewise, $V_{VGND} \approx GND + V_{DN}$ in sleep mode. The potential difference across the latch in sleep mode is well-defined and equal to $V_{DD} - V_{DP} - V_{DN}$, making data retention possible. In sleep mode, both subthreshold and gate oxide leakage are reduced as follows: 1) The reduced potential difference across the drain/source (V_{DS}) of an OFF transistor results in an exponential decrease in subthreshold leakage. This effect is referred to as drain-induced barrier lowering (DIBL) [19]. 2) Gate oxide leakage decreases superlinearly with reductions in gate/source potential difference (V_{GS}) [10].

2.2 FPGA Hardware Structures

FPGAs consists of an array of programmable logic blocks that are connected through a programmable interconnection network. Most commercial FPGAs use 4-input look-up-tables (4-LUTs) as the combinational logic element in their logic blocks. 4-LUTs are small memories that can implement any logic function having no more than 4 inputs. Each 4-LUT is generally coupled with a flip-flop for implementing sequential logic. Logic blocks in modern FPGAs contain clusters of 4-LUTs and flip-flops. For example, the primary tile in the Xilinx Spartan-3 FPGA [23] is called a configurable logic block (CLB) tile. A CLB contains 4 SLICEs, each comprised of 2 4-LUTs and 2 flip-flops, as shown in Fig. 2. FPGA interconnect is composed of variable length wire segments and programmable routing switches. A switch and the wire segment it drives are often referred to as a *routing resource*. In Spartan-3, LOCAL, DIRECT, DOUBLE, HEX and LONG routing resources are available. LOCAL resources are for connections internal to a CLB. DIRECT resources allow a CLB to connect to one of its immediate neighbors. DOUBLE and HEX resources span 2 and 6 CLB tiles, respectively. LONG resources span the entire width or height of the FPGA.

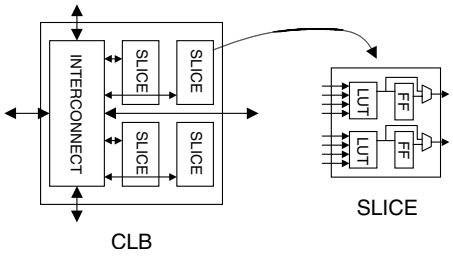

Figure 2: Configurable logic block (CLB) tile in the Xilinx Spartan-3 FPGA [23].

Fig. 3(a) shows a typical buffered FPGA routing switch [13, 18, 12]. It consists of a multiplexer, a buffer and SRAM configuration cells. The multiplexer inputs (labeled i1-in) connect to other routing conductors or to logic block outputs. The buffer's output connects to a routing conductor or to a logic block input. Programmability is realized through the SRAM configuration cells, which select an input signal to be passed through the switch.

A transistor-level view of a switch with 4 inputs is shown in Fig. 3(b) [18]. NMOS transistor trees are used to implement multiplexers in FPGAs [12][1]. Observe that the buffer is "level-restoring" – transistor $MP1$ serves to pull the buffer's input to rail V_{DD} when logic-1 is passed through the switch [18]. Without $MP1$, if a logic-1 (V_{DD}) were passed through the multiplexer, a "weak-1" would appear on the multiplexer's output ($V_{INT} \approx V_{DD} - V_{TH}$), causing $MP2$ to turn partially ON, leading to excessive buffer leakage.

2.3 FPGA Power Optimization

A number of recent studies have considered optimizing FPGA power consumption at the architecture or circuit level. [8] focussed on reducing dynamic power dissipation and proposed a low-energy FPGA fabric with logic and interconnect considerably different than that of today's commercial FPGAs. [16] considered dual-V_{DD} FPGAs in which some logic blocks are fixed to operate at high-V_{DD} (high speed) and some are

[1]Note that full CMOS transmission gates are generally not used to implement multiplexers in FPGAs because of their larger area and capacitance [12].

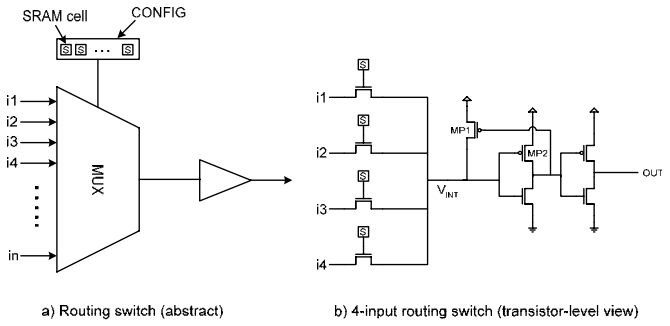

Figure 3: Traditional routing switch: abstract and transistor-level views [18, 12].

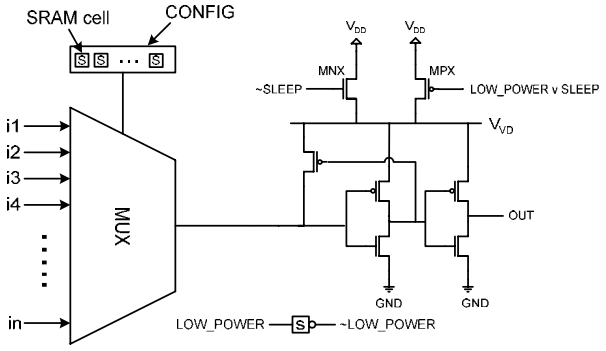

Figure 4: Programmable low-power routing switch.

fixed to operate at low-V_{DD} (low-power but slower). In [15], the same authors extended their dual-V_{DD} FPGA work to allow blocks to operate at *either* high or low-V_{DD}. Power trade-offs at the architectural level were considered in [14], which studied the effect of wire segmentation, LUT size and cluster size on FPGA power efficiency. Optimization of sleep mode leakage in FPGA logic blocks was addressed in [6], which proposed the creation of fine-grained "sleep regions", making it possible for a logic block's LUTs and flip-flops to be put to sleep independently. In [7], the authors propose a more coarse-grained sleep strategy in which entire regions of unused logic blocks may be placed into a low-leakage sleep state.

To our knowledge, the only work to specifically address leakage in FPGA interconnect is [18], which applies well-known leakage reduction techniques to interconnect multiplexers. In particular, [18] proposes: 1) using a mix of low-V_{TH} and high-V_{TH} transistors in the multiplexers, 2) using body-bias techniques to raise the V_{TH} of multiplexer transistors that are OFF, 3) negatively biasing the gate terminals of OFF multiplexer transistors, and 4) introducing extra SRAM cells to allow for multiple OFF transistors on "unselected" multiplexer paths. Our proposed switch designs involve changes to the switch buffer (not the multiplexer), and impose none of the advanced process or biasing requirements of [18]. The leakage improvements offered by our designs are orthogonal to those offered by [18].

3. LOW-POWER ROUTING SWITCH DESIGN

The proposed switch designs are based on three key observations that are specific to FPGA interconnect:

1. Routing switch inputs are tolerant to "weak-1" signals. That is, logic-1 input signals need not be rail V_{DD} – it is acceptable if they are lower than this. This is due to the level-restoring buffers that are *already* deployed in FPGA routing switches (see Fig. 3(b)).
2. There exists sufficient timing slack in typical FPGA designs to allow a sizable fraction of routing switches to be slowed down, without impacting overall design performance. We address this in the next section.
3. Most routing switches simply feed other routing switches (via metal wire segments). This observation holds for the majority of switches in the Xilinx Spartan-3 FPGA [23]. Observation #1 (above) permits such switches to produce "weak-1" signals. The main exceptions to this are switches that drive inputs on logic blocks.

Based on these observations, we propose the new switch design shown in Fig. 4. The switch includes NMOS and PMOS sleep transistors in parallel (MNX and MPX). The sleep structure is similar to that in Fig. 1(b), with diode DP being replaced by an NMOS transistor, MNX. The new switch can operate in three different modes as follows: In high-speed mode, MPX is turned ON and therefore, the virtual V_{DD} (V_{VD}) is equal to V_{DD} and output swings are full rail-to-rail. The gate terminal of MNX is left at V_{DD} in high-speed mode, though this transistor generally operates in the cut-off region, with its $V_{GS} < V_{TH}$. During a 0-1 logic transition however, V_{VD} may temporarily drop below $V_{DD} - V_{TH}$, causing MNX to leave cut-off and assist with charging the switch's output load.

In low-power mode, MPX is turned OFF and MNX is turned ON. The buffer is powered by the reduced voltage, $V_{VD} \approx V_{DD} - V_{TH}$. Since $V_{VD} < V_{DD}$, speed is reduced vs. high-speed mode. However, output swings are reduced by V_{TH}, reducing switching energy, and leakage is reduced for the same reasons mentioned above in conjunction with Fig. 1(b). Lastly, in sleep mode, both MPX and MNX are turned OFF, similar to the supply gating notion in Fig. 1(a).

In addition to the switch buffer in Fig. 4, we also propose the alternate design shown Fig. 5, which offers a different power/area trade-off. In the alternate design, the bodies of the PMOS transistors are tied to V_{VD}, rather than the typical V_{DD}. This lowers the threshold voltage of the PMOS transistors in low-power mode (via the "body effect"), thus increasing their drive strength. In high-speed mode, as mentioned above, V_{VD} drops temporarily below V_{DD} during a 0-1 logic transition, and therefore, improved PMOS drive capability may also be exhibited in this mode. The benefit of stronger drive strength is that the sleep transistors can be made smaller, reducing the area overhead of the proposed switch vs. a traditional switch. The downside is that the reduced threshold voltage of the PMOS transistors will likely lead to greater subthreshold leakage in these transistors vs. leakage in the initial design of Fig. 4. We refer to the

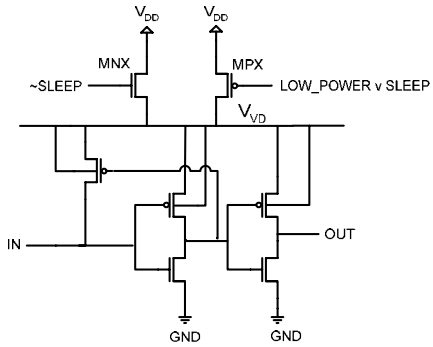

Figure 5: Routing switch buffer alternate design.

switch design in Fig. 4 as the *basic* design and that in Fig. 5 as the *alternate* design. We investigate the characteristics of both designs in our experimental study.

In essence, the new switch designs mimic the programmable dual-V_{DD} concepts proposed in [16, 15], while avoiding the costs associated with true dual-V_{DD}, such as distributing multiple power grids and providing multiple supply voltages at the chip level. In traditional dual-V_{DD} design, level converters are required to avoid excessive leakage when circuitry operating at low supply drives circuitry operating at high supply. However, in this case, because of observation #1, no level converters are required when a switch in low-power mode drives a switch in high-speed mode.

We envision that the selection between low-power and high-speed modes can be realized through an extra configuration SRAM cell in each routing switch. Alternately, to save area, the extra SRAM cell could be shared by a number of switches, all of which must operate in the same mode. We expect that today's commercial FPGA routing switches already contain configuration circuitry to place them into a known state when they are unused. This circuitry can be used to select sleep mode, as appropriate. A key advantage of the proposed designs is that they have no impact on FPGA router complexity – the mode selection can be made at the post-routing stage, when timing slacks are accurately known.

The relatively low hardware cost and negligible software impact make the proposed switch designs quite practical. We anticipate they can be deployed in place of most existing routing switches in commercial FPGAs.

4. SLACK ANALYSIS

The benefits of a routing switch that offers a low-power (slow) mode depend on there being a sufficient fraction of routing resources that may actually operate in this mode, without violating design performance (timing) constraints. This depends directly on the amount of "timing slack" present in typical FPGA designs. In custom ASICs, any available slack is generally eliminated by sizing down transistors, saving silicon area and cost. In the FPGA domain however, the device fabric is fixed, and therefore, it is conceivable that for many designs the available timing slack is substantial.

To motivate our switch designs, we evaluated the timing slack in 22 routed industrial designs implemented in the Xilinx Spartan-3 FPGA [23] (described in Section 2.2). We used the Xilinx placement and routing tools to generate a performance-optimized layout for each design as follows: First, each design was placed and routed with an easy-to-meet (clock period) timing constraint. Then, based on the performance achieved, a more aggressive constraint was generated and the place and route tools were re-executed using the new constraint. The entire process was repeated until a constraint that could not be met by the layout tools was encountered. We evaluate timing slack in the layout solution corresponding to the most aggressive (but achievable) constraint observed throughout the entire iterative process. By evaluating slack with respect to such aggressive constraints, we ensure that the picture of available timing slack we generate is not overly optimistic.

To gauge slack, we implemented and applied the algorithm in [22], which finds a maximal set of a design's driver/load connections that may be slowed down by *pre-specified* percentage (without violating timing constraints). The algorithm was originally used to select sets of transistors to have high-V_{TH} in a dual-V_{TH} ASIC design framework. Since our aim is to maximize the number of routing switches that operate in low-power mode, we altered the algorithm slightly to establish a preference for selecting connections (to be slowed down) that use larger numbers of routing switches in their routing solutions. In [22], each driver/load connection can be viewed as having "unit weight". In our implementation, we employ a simple heuristic: each connection is assigned a weight corresponding to the number of routing switches in its routing solution. Instead of finding a maximum size set of connections that may be slowed down (as in [22]), we apply the same algorithm to find a maximum weight set of connections that may be slowed down. The interested reader is referred to [22] for complete details.

We performed three slack analyses for each design and computed sets of connections that may be slowed down by 25%, 50%, and 75%. We then determined the fraction of routing resources that were used in the routing of the selected connections – i.e. the fraction of *used* routing resources that may be slowed down. The results are shown in Fig. 6. The vertical axis shows the fraction of routing resources that may be slowed down by a specific percentage, averaged across all 22 designs. The horizontal axis shows the main routing resource types in Spartan-3. For each resource type, three bars represent the fraction of used routing resources of that type that may be slowed down. That is, the left-most set of bars indicate that roughly 80%, 75% and 70% of used DOUBLE resources may be slowed down by 25%, 50% and 75%, respectively. The right-most set of bars in Fig. 6 provides average results across all resource types. Observe, for example, that ∼75% (on average) of all routing resources can be slowed down by 50%. The considerable slack in typical FPGA designs bodes well for the proposed routing switch designs.

5. EXPERIMENTAL STUDY

5.1 Methodology

Unless noted otherwise, all HSPICE simulation results reported in this paper were produced at 85°C using the Berke-

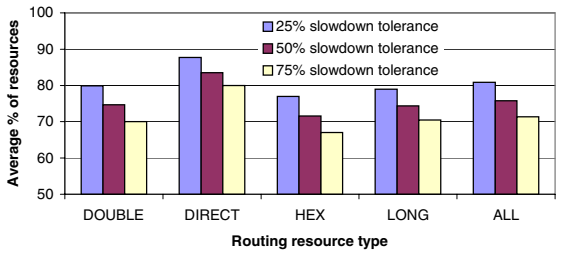

Figure 6: Timing slack in industrial FPGA designs.

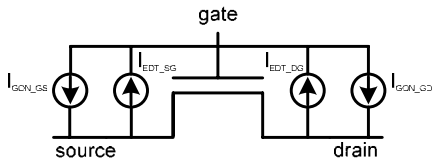

Figure 7: Model for transistor gate oxide leakage [5].

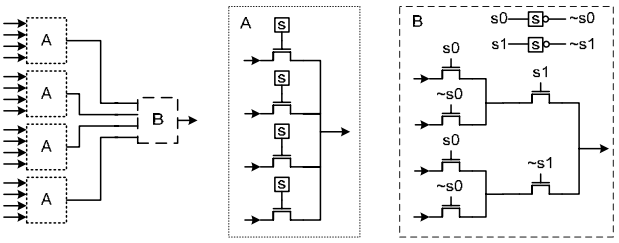

Figure 8: 16-to-1 multiplexer implementation.

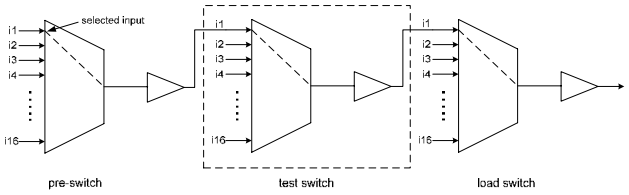

Figure 9: Baseline test platform.

ley Predictive Technology Models (BPTM) for a 70nm technology [2]. The technology models were enhanced to account for gate oxide (tunneling) leakage using four voltage controlled current sources, as shown in Fig. 7, and described in [5]. Both *direct tunneling* current (in an ON transistor) as well as *edge-directed tunneling* (in an OFF transistor) are modeled through current sources I_{GON_GS}, I_{GON_GD} and I_{EDT_SG}, I_{EDT_DG}, respectively. The results presented correspond to an oxide thickness of 1.2nm [1].

To study the proposed switch designs, we first developed a 16-input traditional routing switch (see Fig. 3(b)), representative of those in current commercial FPGAs [23]. The buffer was sized for equal rise and fall times, with the second inverter stage being 3 times larger than the first stage. The 16-to-1 input multiplexer was constructed as shown in Fig. 8, and we believe it reflects a reasonable trade-off between speed and area. Two "stages" of 4-to-1 multiplexer are used. Input-to-output paths through the multiplexer consist of three NMOS transistors. As in [18], SRAM configuration cells are assumed to be shared amongst the four 4-to-1 multiplexers in the first stage; thus, the entire 16-to-1 multiplexer requires 6 SRAM cells to select a path from one of its inputs to its output.

Having developed a traditional switch, we used it as a basis for developing the proposed switch designs. Specifically, in the proposed designs, transistor MPX was sized to provide (high-speed mode) performance within 5% of the traditional switch. Interconnect delay typically comprises about half of total path delay in FPGAs and therefore, a 5% increase in interconnect delay would produce a 2.5% performance degradation overall. We sized transistor MNX to achieve 50% slower speed performance in low-power vs. high-speed mode. From Fig. 6, we expect that ∼75% of routing switches designed as such could operate in low-power mode in a typical design. Certainly, the sizes of sleep transistors MNX and MPX can be adjusted to realize different area/power/performance trade-offs, as desired. We developed both a basic version of the proposed switch (Fig. 4) as well as an alternate version (Fig. 5). Both versions have the same performance characteristics; however, in the alternate version, we were able to reduce the total width of the sleep transistors by 36% vs. the basic version.

To study the power characteristics of the proposed switch designs, we simulate the conditions of a used switch in an actual FPGA using the test platform shown in Fig. 9. The test platform corresponds to a contiguous path of three switches through an FPGA routing fabric; the multiplexers in all three switches are configured to pass input $i1$ to their outputs. Power and performance measurements are made for the second switch, labeled "test switch" in Fig. 9. Our power measurements include current drawn from all sources, including gate oxide leakage in the multiplexer and sleep transistors. Subthreshold leakage current *through* the *inputs* of the test switch is not included, as this is attributable to the buffer(s) in the preceding switch stage(s). Note also that we ignore the power dissipated in the SRAM configuration cells. Since the contents of such cells changes only during the initial FPGA configuration phase, their speed performance is not critical. We envision that in a future leakage-optimized FPGA, the SRAM configuration cells can be slowed down and their leakage reduced or eliminated using previously published low-leakage memory techniques (e.g., [9]).

5.2 Results

We first examine the difference in leakage power in low-power vs. high-speed mode. Two instances of the test platform are used: one in which all three switches are in high-speed mode, and one in which all three switches are in low-power mode (this produced the most pessimistic power results for low-power mode). We simulated both the high-speed and low-power platforms with identical vector sets, consisting of 2000 random input vectors[2]. We captured the leakage power consumed in the test switch for each vector

[2]Random signals were presented to all 46 inputs in each test platform.

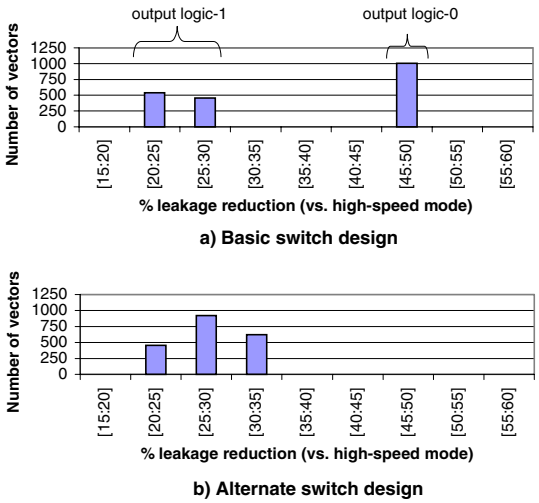

Figure 10: Leakage reduction results (low-power mode vs. high-speed mode).

in both platforms. The results for the *basic* switch design are shown in Fig. 10(a). The horizontal axis shows the percentage reduction in leakage power in the low-power switch vs. the high-speed switch. The vertical axis shows the number of vectors that produced a leakage reduction in a specific range. Observe that larger leakage reductions are realized when the switch output signal is logic-0 vs. logic-1, due primarily to the different leakage characteristics of NMOS vs. PMOS devices. On average, in the basic design, low-power mode offers a 36% reduction in leakage power compared with high-speed mode.

Fig. 10(b) gives results for the *alternate* switch design. Observe that, as expected, leakage reductions in the logic-0 state are smaller than in the basic design (Fig. 10(a)), due to the lower threshold voltage (and increased subthreshold leakage) of the PMOS transistors when the switch operates in low-power mode. On average, the low-power mode of the alternate switch design offers a 28% reduction in leakage vs. high-speed mode.

To evaluate sleep mode leakage, we altered the test platform by attaching the output of the test switch to a different (non-selected) input of the load switch. We also configured the multiplexer in the test switch to disable all paths to the multiplexer output (SRAM cell contents are all 0s). As above, we simulated the modified platform with random vectors and found the average reduction in leakage power for sleep mode vs. high-speed mode to be 61%. Similar results were observed for both the basic and alternate switch designs.

Routing conductors in FPGAs have multiple used and unused switches attached to them. Consequently, we investigated the sensitivity of the low-power mode results to alternate fanout conditions. In one scenario, we augmented the test platform to include 5 unused switches (in sleep mode) on the test switch output. In a second scenario, the test platform was augmented to include 5 used switches on the test switch output. Average leakage power reduction

Table 1: Leakage power reduction results for basic design (unshaded) and alternate design (shaded).

Test scenario	Avg. leakage pwr reduction (%) vs. high-speed mode (basic)	Avg. leakage pwr reduction (%) vs. high-speed mode (alternate)
low-power mode (single fanout)	36.0%	27.6%
sleep mode	60.8%	61.3%
low-power mode (+ unused fanout)	39.7%	28.7%
low-power mode (+ used fanout)	38.7%	29.5%
traditional switch (single fanout)	0.3%	0.25%

results for all scenarios considered are summarized in Table 1, which gives the average percentage reduction in leakage power for each scenario versus the proposed switch in high-speed mode. The unshaded portion of the table gives results for the basic switch design; the shaded portion of the table gives results for the alternate design. Observe that the dependence of the low-power mode results on fanout is relatively weak – the results are slightly better in the more realistic multi-fanout scenarios.

Table 1 also gives data comparing the average leakage power of the proposed switch designs with that of the *traditional* routing switch (used as the development basis). The leakage of the proposed switch designs in high-speed mode is approximately equivalent to that of the traditional switch. Thus, there is no significant "penalty" for deploying the proposed switch designs from the leakage viewpoint, even if they are operated in high-speed mode.

An FPGA implementation of a circuit uses only a fraction of the FPGA's available hardware resources [21]. It is therefore possible that large regions of an FPGA may be "lightly utilized", and that the die temperature in lightly utilized regions is considerably lower than in heavily utilized regions. To gain insight into how the leakage results presented above scale with temperature, we evaluated the leakage characteristics of the proposed designs at low temperature (25°C). The results are summarized in Table 2. The interpretation of the rows and columns in Table 2 is the same as that of Table 1.

Looking first at the 25°C simulation results for the single fanout scenario (row 2 of Table 2), we observe that the difference between the two designs is less pronounced than at high temperature. This is explained by recalling that the superior leakage characteristics of the basic switch design are primarily due to its smaller subthreshold leakage current (see Section 3). Subthreshold leakage increases exponentially with temperature, whereas gate oxide leakage is almost insensitive to temperature [3]. At low temperature, gate oxide leakage therefore comprises a significantly larger fraction of total leakage. Gate oxide leakage is smaller in the alternate vs. the basic design due to its smaller sleep transistors. This leads to a narrower "gap" between the two designs from the leakage perspective at low temperature. Similar results are evident for the alternate fanout scenarios (extra used and unused fanout).

Table 2: 25°C leakage power reduction results for basic design (unshaded) and alternate design (shaded).

Test scenario	Avg. leakage pwr reduction (%) vs. high-speed mode (basic)	Avg. leakage pwr reduction (%) vs. high-speed mode (alternate)
low-power mode (single fanout)	33.3%	30.0%
sleep mode	64.7%	72.3%
low-power mode (+ unused fanout)	34.8%	31.4%
low-power mode (+ used fanout)	33.3%	31.4%
traditional switch (single fanout)	1.6%	1.3%

In sleep mode (row 3 of Table 2), the alternate design actually offers lower leakage than the basic design at low temperature, again due to the increased significance of gate oxide leakage. At high temperature, where subthreshold leakage dominates, the two designs exhibit roughly equivalent leakage (see row 3 of Table 1). Thus, although the smaller sleep transistors in the alternate design result in lower gate oxide leakage, they do not appear to yield a significant reduction in subthreshold leakage (in sleep mode).

Finally, we evaluated the dynamic power benefits of low-power mode. We found that for the basic switch design, the switching energy consumed in low-power mode was 28% lower than in high-speed mode, due chiefly to the reduced output swing and smaller short-circuit current in the buffer. Note however, that this may represent an optimistic estimate of the dynamic power reduction. The area overhead of the new switch vs. a traditional switch will lead to a larger base FPGA tile, resulting in longer wire segment lengths and increased metal capacitance (higher dynamic power). A precise measurement of the area overhead for incorporating the new switch designs into a commercial FPGA is difficult, as it depends on available layout space and existing transistor sizings, both of which are proprietary. Nevertheless, we attempt a rough estimate of the area overhead below.

As mentioned previously, the 16-input traditional switch we began with requires 6 SRAM configuration cells. An additional cell to control the switch mode increases the SRAM cell count by ∼17%. Based on transistor width, we estimate the area overhead for the remainder of the *basic* switch design (vs. the traditional switch) as ∼31%, mainly due to the need for relatively large sleep transistors. Certainly, routing switches in commercial FPGAs have additional configuration and test circuitry beyond that shown in Fig. 3(b), which will reduce the area overhead of the proposed switch. Pessimistically, we can assume that deploying the basic switch design increases an FPGA's interconnect area by 30% and that interconnect accounts for ∼2/3 (66%) of an FPGA's base tile area [18]. Given this, the overall tile area increase to include the proposed switch amounts to ∼20%. Assuming a square tile layout, the tile length in each dimension would increase by ∼9.5%. However, the metal wire segment represents only a fraction of the capacitance seen by a switch output – significant capacitance is due to fanout switches that *attach* to the metal segment. This "attached switch capacitance" is unaffected by a larger tile length. Thus, 9.5% is a loose upper bound on the potential increase in capacitance seen by a switch output. The capacitance increase is surpassed considerably by the dynamic power reductions offered by the proposed switch. The projected tile area breakdowns for the traditional and basic switch types are summarized graphically in Figs. 11(a) and (b), respectively.

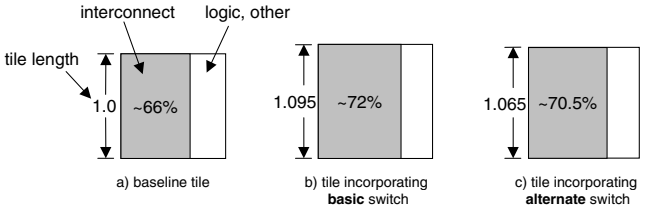

Figure 11: Projected tile area breakdown for traditional and proposed switch types.

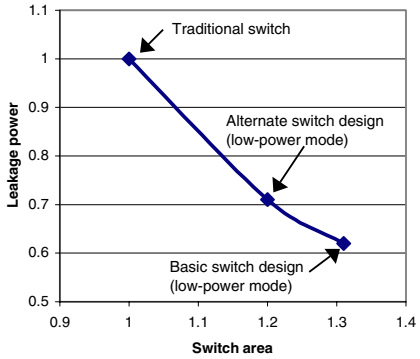

Figure 12: Leakage and area of switch designs.

We also examined the dynamic power characteristics of the *alternate* switch design and observed results similar to those of the basic design – switching energy was reduced by 29% in low-power vs. high-speed mode. As mentioned previously however, the alternate switch design has a considerably lower area overhead compared to the basic design. Applying the same "rough" analysis used above, we expect that incorporating the alternate switch design into an FPGA would increase the base tile length in each dimension by only ∼6.5% (see Fig. 11(c)).

In summary, the results show that both of the proposed switch designs have attractive qualities: the basic design offers greater leakage reductions; the alternate design requires less silicon area. The leakage/area trade-offs between the switch designs are illustrated in Fig. 12; area and power values in the figure are normalized to those of the traditional switch design.

6. CONCLUSIONS

Static and dynamic power dissipation in FPGAs is dominated by that consumed in the interconnection fabric, making low-power interconnect a mandatory feature of future low-power FPGAs. In this paper, we proposed two new FPGA routing switch designs that can be programmed to operate in high-speed, low-power or sleep mode. Leakage in

low-power mode is reduced by 36-40% vs. high-speed mode; dynamic power is reduced by up to 28%. Sleep mode offers leakage reductions of 61%. An alternate version of the switch design offers leakage reductions of 28-30% in low-power vs. high-speed mode, and has a 36% smaller area overhead. We showed that the timing slack in typical FPGA designs permits the majority of switches to operate in low-power mode. The switch designs require only minor changes to a traditional FPGA routing switch and have no impact on router complexity, making them easy to deploy in current commercial FPGAs.

Acknowledgements

The authors thank Navid Azizi for developing the enhanced 70*n*m technology models and for his assistance with using them.

7. REFERENCES

[1] *2002 International Technology Roadmap for Semiconductors (ITRS).*

[2] *http://www.device.eecs.berkeley.edu/∼ptm/.*

[3] A. Agarwal, C.H. Kim, S. Mukhopadhyay, and K. Roy. Leakage in nano-scale technologies: Mechanisms, impact and design considerations. In *ACM/IEEE Design Automation Conference*, pages 6–11, 2004.

[4] M. Anis, S. Areibi, M. Mahmoud, and M. Elmasry. Dynamic and leakage power reduction in MTCMOS circuits using an automated efficient gate clustering technique. In *ACM/IEEE Design Automation Conference*, pages 480–485, 2002.

[5] N. Azizi and F.N. Najm. An asymmetric SRAM cell to lower gate leakage. In *IEEE International Symposium on Quality Electronic Design*, pages 534–539, 2004.

[6] B.H. Calhoun, F.A. Honore, and A. Chandrakasan. Design methodology for fine-grained leakage control in MTCMOS. In *ACM/IEEE International Symposium on Low-Power Electronics and Design*, pages 104–109, 2003.

[7] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M.J. Irwin, and T. Tuan. Reducing leakage energy in FPGAs using region-constrained placement. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 51–58, 2004.

[8] V. George and J. Rabaey. *Low-Energy FPGAs: Architecture and Design*. Kluwer Academic Publishers, Boston, MA, 2001.

[9] C.H. Kim, J.-J. Kim, S. Mukhopadhyay, and K. Roy. A forward body-biased low-leakage SRAM cache: Device and architecture considerations. In *ACM/IEEE International Symposium on Low-Power Electronics and Design*, pages 6–9, 2003.

[10] R.K. Krishnamurthy, A. Alvandpour, V. De, and S. Borkar. High-performance and low-power challenges for sub-70nm microprocessor circuits. In *IEEE Custom Integrated Circuits Conference*, pages 125–128, 2002.

[11] K. Kumagai, H. Iwaki, H. Yoshida, H. Suzuki, T. Yamada, and S. Kurosawa. A novel powering-down scheme for low Vt CMOS circuits. In *IEEE Symposium on VLSI Circuits*, pages 44–45, 1998.

[12] G. Lemieux. Design of interconnection networks for programmable logic devices. In *Ph.D. Thesis*. ECE Department, University of Toronto, 2003.

[13] D. Lewis, V. Betz, D. Jefferson, A. Lee, C. Lane, P. Leventis, et. al. The Stratix routing and logic architecture. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 12–20, 2003.

[14] F. Li, D. Chen, L. He, and J. Cong. Architecture evaluation for power-efficient FPGAs. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 175–184, 2003.

[15] F. Li, Y. Lin, and L. He. FPGA power reduction using configurable dual-Vdd. In *ACM/IEEE Design Automation Conference*, pages 735–740, 2004.

[16] F. Li, Y. Lin, L. He, and J. Cong. Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 42–50, 2004.

[17] K.W. Poon, A. Yan, and S. J. E. Wilton. A flexible power model for FPGAs. In *International Conference on Field-Programmable Logic and Applications*, pages 312–321, 2002.

[18] A. Rahman and V. Polavarapuv. Evaluation of low-leakage design techniques for field-programmable gate arrays. In *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, pages 23–30, 2004.

[19] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand. Leakage current mechanisms and leakage reduction techniques in deep-submicrometer CMOS circuits. In *Proceedings of the IEEE*, pages 305–327, February 2003.

[20] L. Shang, A. Kaviani, and K. Bathala. Dynamic power consumption of the Virtex-II FPGA family. In *ACM International Symposium on Field Programmable Gate Arrays*, pages 157–164, Monterey, CA, 2002.

[21] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *IEEE Custom Integrated Circuits Conference*, pages 57–60, 2003.

[22] Q. Wang and S. B. K. Vrudhula. Algorithms for minimizing standby power in deep submicrometer, dual-Vt CMOS circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(3):306–318, March 2002.

[23] Xilinx, Inc., San Jose, CA. *Spartan-3 FPGA Data Sheet*, 2004.