

Low-Power Programmable FPGA Routing Circuitry

Jason H. Anderson, *Member, IEEE*, and Farid N. Najm, *Fellow, IEEE*

Abstract—We consider circuit techniques for reducing FPGA power consumption and propose a family of 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 traditional FPGA routing switches. In low-power mode, speed is curtailed in order to reduce power consumption. Leakage is reduced by 28-52% in low-power versus high-speed mode, depending on the particular switch design selected. Dynamic power is reduced by 28-31% in low-power mode. Leakage power in sleep mode, which is suitable for unused routing switches, is 61-79% lower than in high-speed mode. Each of the proposed switch designs has a different power/area/speed trade-off. All of the designs require only minor changes to a traditional routing switch and involve relatively small area overhead, making them easy to incorporate into current commercial FPGAs. The applicability of the new switches is motivated through an analysis of timing slack in industrial FPGA designs. It is observed that a considerable fraction of routing switches may be slowed down (operate in low-power mode), without impacting overall design performance.

Index Terms—Field-programmable gate arrays, FPGAs, power, leakage, interconnect, optimization.

I. 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 digital 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 [2]. A recent work by Kuon and Rose compared 90nm FPGAs to ASICs and found FPGAs consume 7-14 times more dynamic power than ASICs, and 5-87 times more leakage power [3]. 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 [4], [5], [6], [7], [8]. Interconnect dominates dynamic power in FPGAs due to the composition of the interconnect structures, which consist of pre-fabricated 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 [7].

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. Most of today’s FPGAs do not offer any sleep support and thus, it is valuable to consider FPGA circuit structures that can reduce *both* active and sleep leakage. An exception to this is the recently introduced Actel IGLOO FPGA, which offers an ultra low-power mode [9], similar to sleep mode.

The dominance of interconnect in total FPGA power consumption makes it a high-leverage target for power optimization. In this paper, we present a family of novel FPGA routing switch designs that offer reduced leakage and dynamic power dissipation. A property common to all of the proposed switch designs is the concept of “programmable mode”. Specifically, the switch 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 II presents related work and necessary background material. The proposed switch designs are described in Section III. Section IV analyzes the timing slack present in industrial FPGA designs implemented in the Xilinx Spartan-3 commercial FPGA [10] (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

The authors are with the Department of Electrical and Computer Engineering, University of Toronto, Toronto, Ontario, Canada. Jason H. Anderson is also with Xilinx, Inc., Toronto, Ontario, Canada.

given in Section V. Conclusions are offered in Section VI. A preliminary version of a portion of this work appeared in [11], [12]. Here, we provide more detailed experimental results at multiple temperatures, and we introduce additional circuit modifications, producing better leakage results, and we expand upon the prior work to address leakage in other parts of the FPGA interconnect.

II. BACKGROUND AND RELATED WORK

A. 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 [13]. 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 [14], 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 [15] 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) [13]. 2) Gate oxide leakage decreases superlinearly with reductions in gate/source potential difference (V_{GS}) [16].

B. FPGA Hardware Structures

FPGAs consists of an array of programmable logic blocks that are connected through a programmable interconnection network. Today's commercial FPGAs use look-up-tables with either 4 or 6 inputs (4-LUTs or 6-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 [10] 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

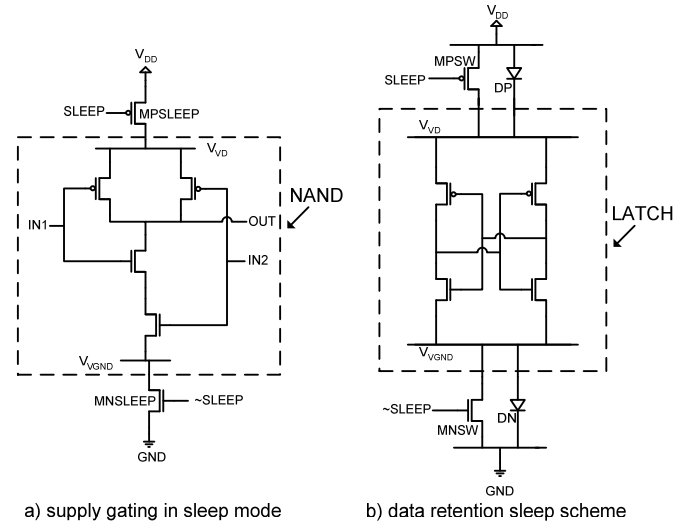


Fig. 1. Sleep leakage reduction techniques [14], [15].

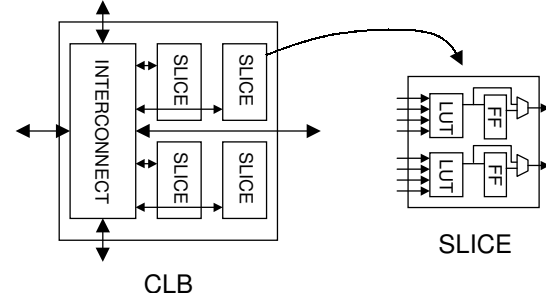


Fig. 2. Configurable logic block (CLB) tile in Xilinx Spartan-3 FPGA [10].

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.

Fig. 3(a) shows a typical buffered FPGA routing switch [17], [7], [18]. It consists of a multiplexer, a buffer and SRAM configuration cells. The multiplexer inputs (labeled $i1-i_n$) 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) [7]. NMOS transistor trees are used to implement multiplexers in FPGAs [18]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 [7]. Without *MP1*, if a logic-1 (V_{DD}) were passed through the

¹Note that full CMOS transmission gates are generally not used to implement multiplexers in FPGAs because of their larger area and capacitance [18].

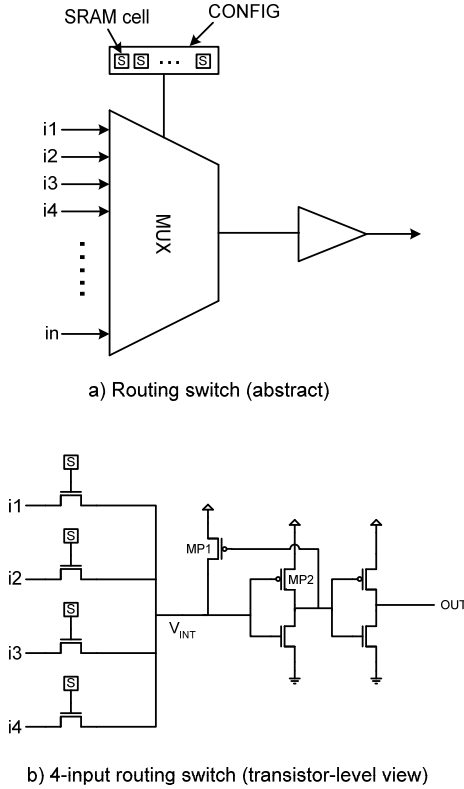


Fig. 3. Traditional routing switch: abstract and transistor-level views [7], [18].

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.

C. FPGA Power Optimization

A number of recent studies have considered optimizing FPGA power consumption at the architecture or circuit level. [2] 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. [19] considered dual- V_{DD} FPGAs in which some logic blocks are fixed to operate at high- V_{DD} (high speed) and some are fixed to operate at low- V_{DD} (low-power but slower). In [20], the same authors extended their dual- V_{DD} FPGA work to allow blocks to operate at *either* high or low- V_{DD} . [21] and [22] apply configurable dual- V_{DD} concepts to both logic blocks and interconnect. Power trade-offs at the architectural level were considered in [6], 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 [23], 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 [24], 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.

One of the few works to address leakage in FPGA interconnect is [7], which applies well-known leakage reduction techniques from the ASIC domain [7]. In particular, [7] 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. A more recent paper applies dual- V_{TH} techniques to the routing switch buffers in addition to the multiplexers [25]. Our proposed switch designs involve changes to both the switch buffer and the multiplexer, however impose none of the advanced process or biasing requirements of [25], [7]. The leakage improvements offered by our designs are, in large part, orthogonal to those offered by [25], [7].

III. LOW-POWER ROUTING SWITCH DESIGNS

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. This assertion will be demonstrated 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 commercial FPGAs, such as the Xilinx Spartan-3 FPGA [10]. Observation #1, above, permits such switches to produce “weak-1” signals. The main exceptions to this observation are switches that drive inputs on logic blocks.

Based on these three 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. This effect is illustrated in Fig. 5 which shows a switch’s output response and V_{VD} on a rising input.

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 versus high-speed mode. However, output swings are reduced by V_{TH} , reducing switching energy, and leakage is reduced

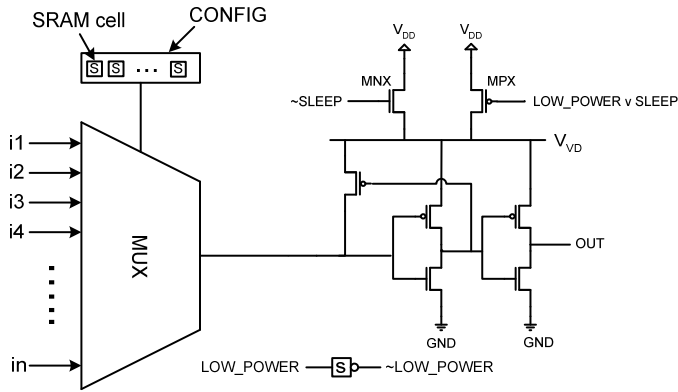


Fig. 4. Programmable low-power routing switch (basic design).

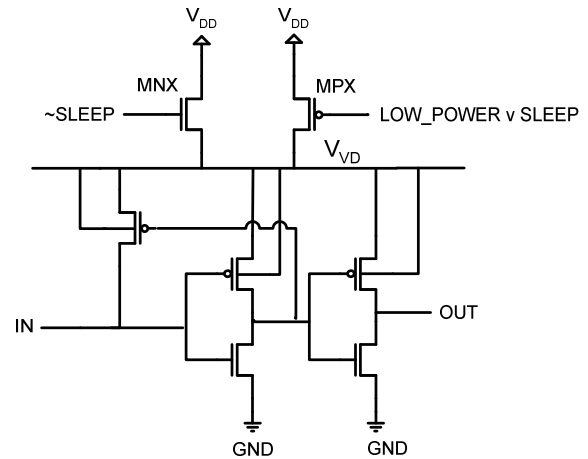


Fig. 6. Routing switch buffer alternate design.

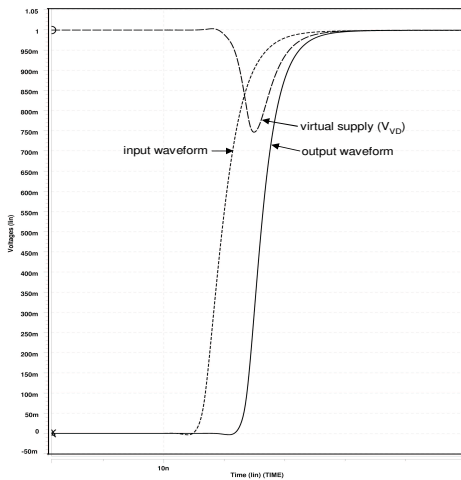


Fig. 5. HSPICE simulation results for switch: input, output and virtual V_{DD} waveforms.

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 in Fig. 4, a second buffer design is proposed in Fig. 6, and it 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 enhanced drive strength is that the sleep transistors can be made smaller, reducing the area overhead of the proposed switch versus 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 versus leakage in the initial design of Fig. 4. In this paper, the switch design in Fig. 4 is referred to as the *basic* design, and the one in Fig. 6 as the *alternate* design.

The *alternate* design offers a different *leakage/area* trade-off versus the *basic* design; that is, the *alternate* design requires less area, but is likely more “leaky”. For both designs, a straightforward extension can be made to realize a different *leakage/speed* trade-off. Specifically, one can apply the sleep structure discussed above to the multiplexer that precedes the buffer, as shown in Fig. 7. Two additional sleep transistors, MNX_M and MPX_M , are introduced into the pull-up network of the multiplexer and its configuration circuitry. The programmable multiplexer concept can be combined with both the *basic* buffer design, as well as the *alternate* design. These switch variants are referred to as *basic+MUX* and *alternate+MUX*, respectively. The aim of these switch variants is to reduce gate oxide leakage through the switch multiplexer inputs.

In high-speed mode, the multiplexer is powered by V_{DD} , similar to a standard routing switch. In low-power mode, the multiplexer is powered by $V_{DD} - V_{TH}$. Recall that the multiplexer select lines attach to the gate terminals of NMOS transistors (see Fig. 3). The reduced voltage on the select lines in low-power mode will lower gate oxide leakage in the multiplexer. Leakage in the SRAM configuration cells will also be reduced in low-power mode. Of course, signal propagation delay through the multiplexer will increase in low-power mode. Note that, because the contents of the SRAM configuration cells do not change during normal operation, the SRAM cell performance is not critical. Consequently, transistors MNX_M and MPX_M can be made very small. While the sizes of MNX and MPX strongly influence the FPGA’s performance, the sizes of MNX_M and MPX_M do not.

There are several reasons for introducing two additional sleep transistors (MNX_M and MPX_M) instead of simply using the existing sleep transistors (MNX and MPX) to control *both* the buffer and the multiplexer. First, as mentioned above, the V_{VD} signal powering the buffer may swing below $V_{DD} - V_{TH}$ during a 0-1 logic transition. If the same sleep transistors were shared between the multiplexer and buffer, such a voltage drop, depending on its magnitude, could destabilize the con-

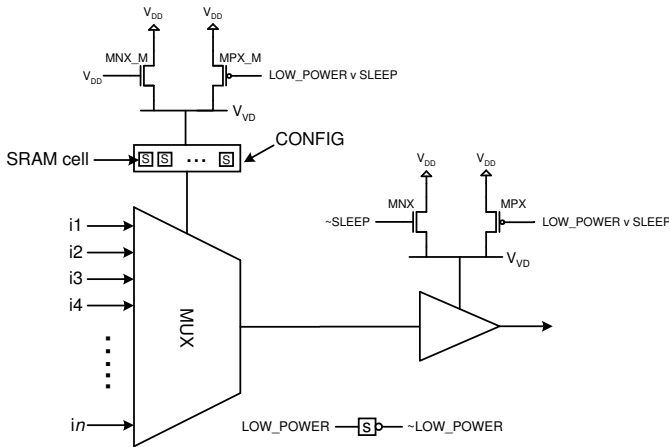


Fig. 7. Switch multiplexer with programmable mode.

tents of the SRAM configuration cells – a catastrophic device failure. Second, sleep mode works differently in the buffer versus the multiplexer. In the buffer, both MNX and MPX are turned OFF in sleep mode. In the multiplexer, sleep and low-power mode are identical. If MNX_M and MPX_M were turned OFF in sleep mode, the SRAM configuration cells would lose their state. Moreover, the voltages on the multiplexer select lines would not be well-defined, potentially turning ON one or more multiplexer paths, and introducing additional capacitive loading on upstream routing switches.

Finally, for all of the switch designs, we also consider a variant for sleep mode, shown in Fig. 8. Transistor $MSLEEP$ is added to pull node V_{INT} to ground in sleep mode. The intent of $MSLEEP$ is to set the buffer’s internal node voltages (V_{INT} , V_{INTB} , OUT) to a known state in sleep mode, thus improving buffer leakage. This differs from the switch designs described above, wherein the internal node voltages are allowed to “float” in sleep mode, possibly leading to a scenario in which both transistors in an inverter stage are (partially) ON. When $MSLEEP$ is ON, V_{INT} is pulled to logic-0, V_{INTB} is pulled to V_{DD} , and, provided V_{DD} is sufficiently high, OUT is pulled to logic-0. Note that since $MSLEEP$ is loading the multiplexer output, its size should be kept very small. Observe that V_{INT} cannot be pulled high (instead of low) in sleep mode. Doing so would cause $MP1$ to turn ON, pulling V_{DD} high to V_{DD} , thereby negating the benefit of MNX and MPX being OFF in sleep mode.

Fig. 9 summarizes all of the switch designs considered in this paper. As shown, the switch buffer can be of either the *basic* design (Fig. 4) or the *alternate* design (Fig. 6). Two different switch multiplexers are possible: one with the sleep structure in its pull-up network, and one without the sleep structure. This yields a total of four different switch designs. The NMOS pull-down transistor on the buffer input (for use in sleep mode) can be introduced into any of the four designs, and therefore, eight different sleep modes will be evaluated.

In essence, the new switch designs mimic the programmable dual- V_{DD} concepts proposed in [19], [20], [21], [22], while avoiding the costs associated with true dual- V_{DD} , such as

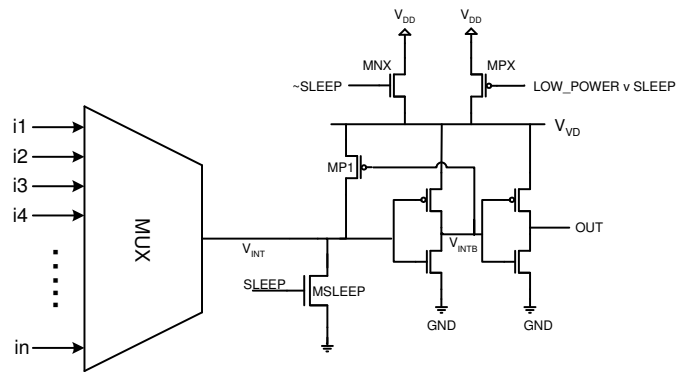


Fig. 8. Sleep mode variant.

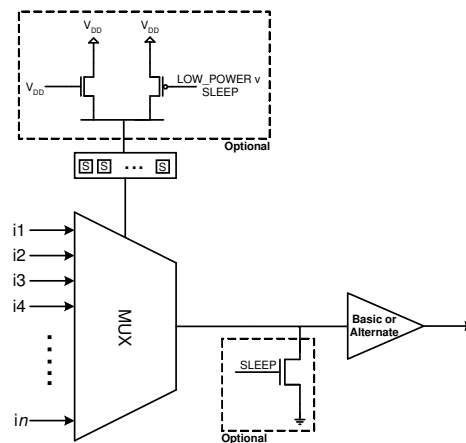


Fig. 9. Family of routing switch designs.

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: The fanout switch’s buffer is a level-restoring buffer, making the fanout switch’s inputs tolerant to lower input voltages. For high reliability, the trip point of proposed switch designs must be carefully tuned to maintain functionality under reduced input swing.

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, as we have included in the bottom of Fig. 4. 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. It is expected that they can be deployed in place of most existing routing switches in commercial FPGAs.

IV. 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 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 the proposed switch designs, timing slack was evaluated in 22 routed industrial designs implemented in the Xilinx Spartan-3 FPGA [10] (described in Section II-B). The designs range in size from 2,700 connections to route to over 150,000 connections to route. The Xilinx placement and routing tools were used 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. Timing slack was evaluated in the layout solution corresponding to the most aggressive, yet achievable, constraint observed throughout the entire iterative process. Evaluating slack with respect to such aggressive constraints ensures that the picture of available timing slack generated is not overly optimistic.

To gauge slack, the algorithm in [26] was implemented, which finds a maximal set of a design’s driver/load connections that may be slowed down by a *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 the aim here is to maximize the number of routing switches that operate in low-power mode, the algorithm was altered slightly to establish a preference for selecting connections (to be slowed down) that use larger numbers of routing switches in their routing solutions. In [26], each driver/load connection can be viewed as having “unit weight”. In our implementation, a simple heuristic is employed: 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 [26]), the same algorithm is applied to find a maximum weight set of connections that may be slowed down. The interested reader is referred to [26] for complete details.

Three slack analyses were performed for each design and sets of connections that may be slowed down by 25%, 50%, and 75% were computed. Then, the fraction of routing re-

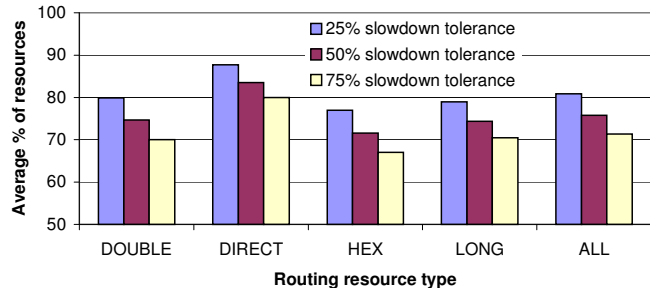


Fig. 10. Timing slack in industrial FPGA designs.

sources that were used in the routing of the selected connections was determined; that is, the fraction of *used* routing resources that may be slowed down. The results are shown in Fig. 10. 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. For example, 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. 10 provides average results across all resource types. Observe, for example, that $\sim 75\%$ of all routing resources can be slowed down by 50%, on average. A design-by-design breakdown for this particular data point is shown in Fig. 11, which gives, for each benchmark circuit, the fraction of its routing resources that may be slowed by 50%. Observe that the results are fairly circuit dependent, however, the fraction lies above 60% for all but two circuits.

Interestingly, though these slack analysis results are for commercial FPGA circuits collected from Xilinx customers, the results here agree closely with prior work by both Betz and Hutton et. al., which showed that only 20% of an FPGA’s routing resources need to be high-speed [27], [28]. The results above, and those in [27], [28] confirm that there is considerable slack in typical FPGA designs, which bodes well for the proposed routing switch designs.

V. EXPERIMENTAL STUDY

A. Methodology

Unless noted otherwise, all HSPICE simulation results reported in this paper were produced at 85°C using the Berkeley Predictive Technology Models (BPTM) for a 1 Volt 70nm technology [29]. The technology models were enhanced to account for gate oxide (tunneling) leakage using four voltage controlled current sources, as shown in Fig. 12, and described in [30]. 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, the first step was to develop a 16-input traditional routing switch [see Fig. 3(b)],

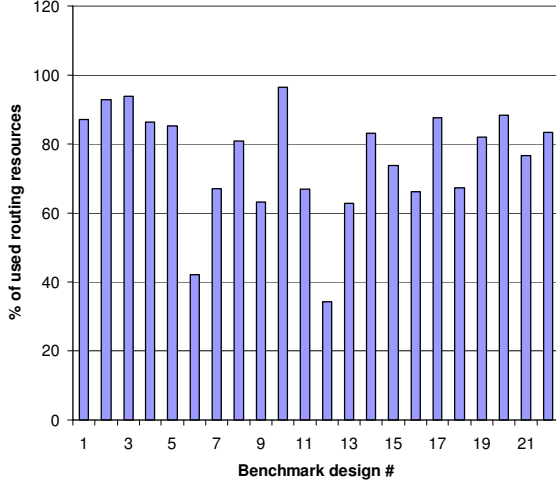


Fig. 11. Fraction of routing resources that may be slowed by 50% for each benchmark circuit.

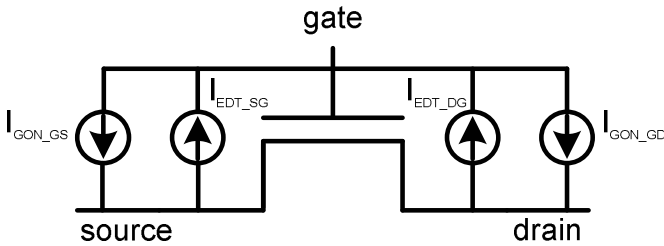


Fig. 12. Model for transistor gate oxide leakage [30].

representative of those in current commercial FPGAs [10]². 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. 13, and it is believed to reflect a reasonable trade-off between speed and area. Two stages of 4-to-1 multiplexers are used to form the 16-input multiplexer. Input-to-output paths through the multiplexer consist of three NMOS transistors. As in [7], 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.

The 16-input traditional switch was then used as a basis for developing the proposed switch designs. Specifically, in the *basic* design, transistor *MPX* (see Fig. 4) 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. Transistor *MNX* was sized to achieve 50% slower speed performance in low-power versus high-speed mode. From Fig. 10, one can expect that $\sim 75\%$ of routing switches designed as such could operate in low-power

²A 16-input switch was selected as it is similar to the switches driving DOUBLE-length segments in Xilinx Spartan-3 [10].

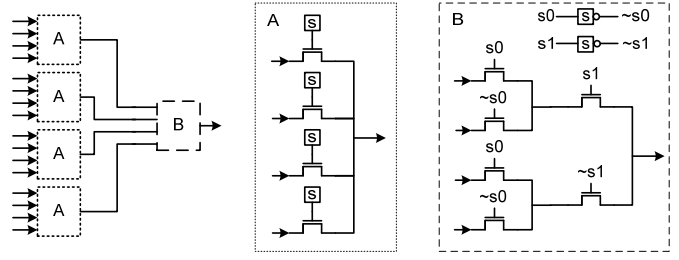


Fig. 13. 16-to-1 multiplexer implementation.

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.

Both a *basic* version of the proposed switch (Fig. 4), as well as an *alternate* version (Fig. 6) were developed. Both versions have the same performance characteristics; however, in the *alternate* version, it was possible to reduce the total width of the sleep transistors by 36% compared to the *basic* version, while maintaining speed. The *basic* and *alternate* switches were then extended to create two additional switch types: *basic+MUX* and *alternate+MUX* (see Fig. 7). In these designs, where the programmable mode concept is applied to the multiplexer, the low-power mode is 80% slower than high-speed mode. Therefore, if these designs are used, slightly fewer routing switches would be permitted to operate in low-power mode.

To study the power characteristics of the proposed switch designs, the conditions of a used switch in an actual FPGA were simulated using the test platform shown in Fig. 14. 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. 14. The 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., [31]). Although not shown in Fig. 14, the presence of the metal routing conductor driven by each switch was modeled by adding a 100 femtofarad capacitor to each switch output, loading representative of routing conductors in the Xilinx Virtex-5 FPGA – a 65nm FPGA [32].

B. Leakage Power Results

In this section, we investigate the leakage power of the different switch designs and their associated modes of operation. We use the high-speed mode of the switches as the baseline

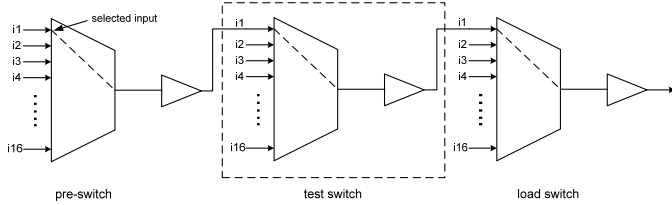


Fig. 14. Baseline test platform.

leakage power to which we compare the leakage power of the low-power and sleep modes.

We first examine the difference in leakage power in low-power versus high-speed mode. That is, we ask: how much leakage power is saved in low-power mode? For this task, two instances of the test platform were 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 configuration produced the most pessimistic power results for low-power mode; that is, this configuration shows the proposed switch in the worst-case conditions from the leakage viewpoint. Both the high-speed and low-power platforms were simulated with identical vector sets, consisting of 2,000 random input vectors³. In roughly half of the vectors, the switch is passing a logic-0 and in the remainder, it is passing a logic-1. The leakage power consumed in the test switch was captured for each vector in both platforms. The results for the *basic* switch design are shown in Fig. 15(a). The horizontal axis shows the percentage reduction in leakage power in the low-power switch versus 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 versus logic-1, due primarily to the different leakage characteristics of NMOS versus PMOS devices. On average, in the *basic* design, low-power mode offers a 36% reduction in leakage power compared with high-speed mode.

Fig. 15(b) gives results for the *alternate* switch design. Observe that, as expected, leakage reductions in the logic-0 state are not as substantial as compared to the *basic* design [Fig. 15(a)], due to the lower threshold voltage, and increased subthreshold leakage of the PMOS transistors when the *alternate* switch operates in low-power mode. As is apparent in the figure, it is no longer possible to differentiate between the case of the output being logic-1 or logic-0. On average, the low-power mode of the *alternate* switch design offers a 28% reduction in leakage versus high-speed mode.

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

³Random signals were presented to all 46 inputs in each test platform.

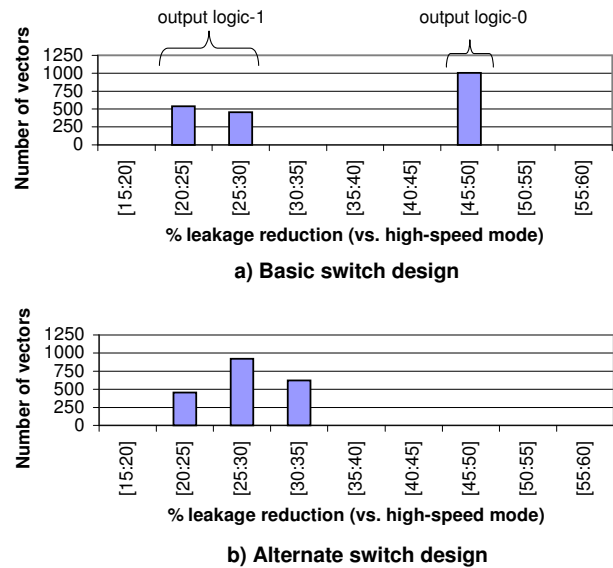


Fig. 15. Leakage reduction results (low-power mode versus high-speed mode).

Routing conductors in FPGAs have multiple used and unused switches attached to them. Consequently, the sensitivity of the low-power mode results to multi-fanout conditions was studied. In one scenario, the test platform was augmented 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 results for all scenarios considered are summarized in Table I, which gives the average percentage reduction in leakage power for each scenario versus the proposed switch designs 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.

The last row of Table I gives data comparing the average leakage power of the proposed switch designs in high-speed mode with that of the *traditional* routing switch used as the development basis. Observe that the leakage of the proposed switch designs in high-speed mode is roughly 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 [5]. It is therefore possible that large regions of an FPGA may be lightly utilized, and that the die temperature in lightly utilized regions is somewhat lower than in heavily utilized regions. To gain insight into how the leakage results presented above scale with temperature, the leakage characteristics of the proposed designs were evaluated at low temperature (25°C). The results are summarized in Table II. The interpretation of the rows and columns in Table II is the same as that of Table I.

TABLE I

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 as in Fig. 14)	36.0%	27.6%
sleep mode	60.8%	61.3%
low-power mode (+ 5 unused fanouts)	39.7%	28.7%
low-power mode (+ 5 used fanouts)	38.7%	29.5%
traditional switch (single fanout)	0.3%	0.25%

Looking first at the 25°C simulation results for the single fanout scenario (row 2 of Table II), one can see 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 III). Subthreshold leakage increases exponentially with temperature, whereas gate oxide leakage is almost insensitive to temperature [33]. At low temperature, gate oxide leakage comprises a larger fraction of total leakage. Gate oxide leakage is smaller in the *alternate* versus 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 multi-fanout scenarios.

To gain more insight into the temperature-dependent leakage results, Fig. 16 shows the DC characteristics of the NMOS transistor model used in this work. Three curves are shown: the top two curves show subthreshold leakage at 85°C and 25°C, respectively, the third curve shows gate oxide leakage. The subthreshold leakage curves show drain current as a function of drain voltage, while gate and source voltages are ground. The gate oxide leakage curve shows gate current versus gate voltage, while the drain and source voltages are ground. Observe that, at high temperature, subthreshold leakage is nearly an order of magnitude higher than gate leakage, whereas, at low temperature, gate and subthreshold leakage fall in the same range. Gate oxide leakage reduction techniques will show considerably more benefit at low temperatures.

In sleep mode (row 3 of Table II), 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 I). 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.

Tables III and IV present the leakage power results for the *basic+MUX* and *alternate+MUX* designs at 85°C and 25°C,

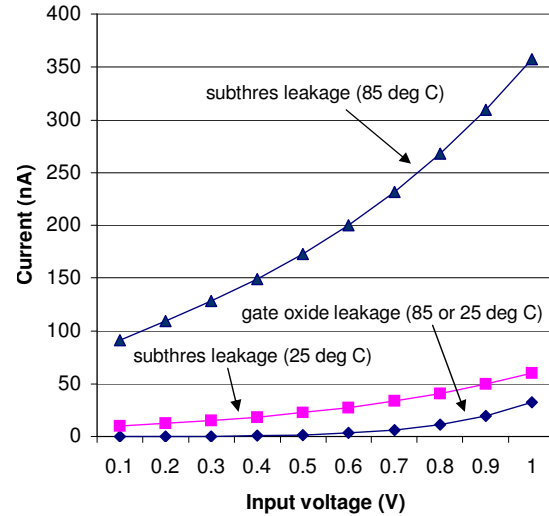


Fig. 16. DC leakage characteristics of NMOS transistor.

TABLE II

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 as in Fig. 14)	33.3%	30.0%
sleep mode	64.7%	72.3%
low-power mode (+ 5 unused fanouts)	34.8%	31.4%
low-power mode (+ 5 used fanouts)	33.3%	31.4%
traditional switch (single fanout)	1.6%	1.3%

respectively. Recall that in these designs, the programmable low-power/high-speed concept is applied to the switch MUX, in *addition* to the switch buffer. At 85°C, the leakage improvements over the original *basic* and *alternate* designs are modest. For example, in the single fanout case, the low-power mode of the *basic+MUX* (*alternate+MUX*) design offers a 42% (33%) leakage reduction versus high-speed mode. This is a moderate improvement over the *basic* (*alternate*) design, which yields a 36% (28%) leakage reduction in low-power mode.

At 25°C, the benefits of reduced gate oxide leakage in the multiplexer (in *basic+MUX* and *alternate+MUX*) are more apparent. Consider row 2 of Table IV, which gives the low-power mode leakage results for the single fanout scenario. Leakage is reduced by 52% in *basic+MUX* and 48% in *alternate+MUX* versus high-speed mode. This can be compared with the *basic* and *alternate* designs which offer 33% and 30% leakage reductions, respectively (row 2 of Table II). Subthreshold and gate oxide leakage exhibit different technology scaling trends. Should gate oxide leakage come to dominate total leakage in deep sub-100nm technologies, the benefits of the *basic+MUX*

TABLE III

85°C LEAKAGE POWER REDUCTION RESULTS FOR *basic+MUX* DESIGN (UNSHADED) AND *alternate+MUX* DESIGN (SHADED).

Test scenario	Avg. leakage pwr reduction (%) vs. high-speed mode (<i>basic+MUX</i>)	Avg. leakage pwr reduction (%) vs. high-speed mode (<i>alternate+MUX</i>)
low-power mode (single fanout as in Fig. 14)	42.2%	33.0%
sleep mode	67.4%	64.0%
low-power mode (+ 5 unused fanouts)	42.0%	32.9%
low-power mode (+ 5 used fanouts)	41.3%	32.6%

TABLE IV

25°C LEAKAGE POWER REDUCTION RESULTS FOR *basic+MUX* DESIGN (UNSHADED) AND *alternate+MUX* DESIGN (SHADED).

Test scenario	Avg. leakage pwr reduction (%) vs. high-speed mode (<i>basic+MUX</i>)	Avg. leakage pwr reduction (%) vs. high-speed mode (<i>alternate+MUX</i>)
low-power mode (single fanout as in Fig. 14)	52.1%	47.8%
sleep mode	68.7%	75.8%
low-power mode (+ 5 unused fanouts)	51.4%	47.6%
low-power mode (+ 5 used fanouts)	50.2%	46.5%

and *alternate+MUX* designs will be amplified.

Finally, we consider the benefits of the variant sleep mode, depicted in Fig. 8, where the internal buffer nodes are pulled to a known state, instead of being allowed to “float”. Leakage reduction results for the variant sleep mode relative to high-speed mode are shown in column 3 of Table V. For comparison, column 2 of the table summarizes the sleep results already presented above for the original sleep mode. Observe that, for all but one of the switch types at both temperatures, the variant sleep mode offers better leakage results. The only exception is the *alternate* design at low temperature, for which similar leakage results are observed for both sleep modes (72% leakage reduction). Pulling internal buffer nodes to a known voltage state in sleep mode ensures that there are at least two OFF transistors on each path from supply to ground in the buffer. This significantly reduces subthreshold leakage due to the stack effect [34]. At high temperature, the variant sleep mode offers a 73-79% leakage reduction versus high-speed mode, whereas the original sleep mode offers a 61-67% leakage reduction.

Most of the leakage benefits of the proposed switches are due to changes to the switch buffer. The results above are for 16-input switches, corresponding those that drive the DOUBLE-length segments in Xilinx Spartan-3 [10]. Different wire segment types will typically have different switch sizes. For example, the HEX-length segments in Spartan-3 are driven by 8-input switches; the DIRECT-length segments in Spartan-

TABLE V

SLEEP MODE LEAKAGE RESULTS 85°C (UNSHADED) AND 25°C (SHADED).

Switch type	Orig. sleep mode % leak. reduction	Sleep mode variant % leak. reduction
basic	60.8%	77.1%
<i>basic+MUX</i>	67.4%	79.0%
<i>alternate</i>	61.3%	73.1%
<i>alternate+MUX</i>	64.0%	75.0%
basic	64.7%	66.9%
<i>basic+MUX</i>	68.7%	73.0%
<i>alternate</i>	72.3%	71.8%
<i>alternate+MUX</i>	75.8%	77.8%

TABLE VI

DYNAMIC POWER RESULTS FOR ALL SWITCH DESIGNS.

Switch type	Switching energy reduction in low-power vs. high-speed mode
basic	28.2%
<i>basic+MUX</i>	28.8%
<i>alternate</i>	30.9%
<i>alternate+MUX</i>	31.2%

3 are driven by 24-input switches, and therefore, have larger multiplexers. Applying the proposed techniques to switches with fewer than 16 inputs will likely produce even greater leakage reductions; whereas, applying them to switches with more than 16 inputs may yield smaller leakage reductions.

C. Dynamic Power Results

The dynamic power characteristics of the switch designs in low-power mode were evaluated; the results are given in Table VI. The dynamic energy benefits of all of the switch designs are similar, ranging from 28-31%, and 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 designs versus 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, a rough estimate of the area overhead is attempted below.

As mentioned previously, the 16-input traditional switch used as the development basis requires 6 SRAM configuration cells. An additional cell to control the switch mode increases the SRAM cell count by ~17%. Based on transistor width, the area overhead for the remainder of the *basic* switch design, versus the traditional switch, is estimated as ~31%, mainly due to the need for large sleep transistors to achieve high performance. 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 switches. Pessimistically, we can assume that deploying the *basic* switch design increases an FPGA’s interconnect area by 30%, and that interconnect accounts for

$\sim 2/3$ (66%) of an FPGA’s base tile area [7]. Given this, the overall tile area increase to include the proposed *basic* switch amounts to $\sim 20\%$. Assuming a square tile layout, the tile length in each dimension would increase by $\sim 9.5\%$. 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. This means that, at most, dynamic power benefits may be reduced from the 28-31% mentioned above, to the range of 19-22%. The capacitance increase is surpassed considerably by the dynamic power reductions offered by the *basic* switch. The projected tile area breakdowns for the traditional and *basic* switch types are summarized graphically in Figs 17(a) and (b), respectively.

As mentioned previously, 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 $\sim 6.5\%$ [see Fig. 17(c)]. The area overheads of the *basic+MUX* and *alternate+MUX* designs are similar to those of the *basic* and *alternate* designs, since sleep transistors *MNX_M* and *MPX_M* (see Fig. 7) can be made small for the reasons mentioned in Section III, and the above area estimates are quite pessimistic to begin with.

D. Summary of Results

In summary, the results show that all of the proposed switch designs have attractive qualities: the *basic* design offers large leakage reductions at high speed; the *alternate* design requires less silicon area; the *basic+MUX* and *alternate+MUX* designs offer the largest reduction in gate oxide leakage. The leakage/area/speed trade-offs between the switch designs are illustrated in Fig. 18; the data values in the figure are normalized to those of the traditional switch design.

E. Projection of Overall Leakage Reduction in an FPGA Tile

The leakage results presented above were for a single routing switch. A “back of the envelope” analysis can be used to project the overall leakage reductions offered by the proposed switch designs in an FPGA tile, which contains *both* logic and interconnect. Based on prior work, one can assume that $\sim 40\%$ of leakage in a tile is in unused circuitry and $\sim 60\%$ in used circuitry [5]. Furthermore, as above, one can assume that about $\sim 66\%$ of leakage in the used and unused circuitry is due to interconnect, with the remainder due to logic.

Consider first the *basic* design with the variant sleep mode, operating at 85°C . In this design, leakage is reduced by 77.1% in sleep versus high-speed mode, and by $\sim 40\%$ in low-power versus high-speed mode (see results tables above). Assuming that all of the unused interconnect can be put into sleep mode, leakage in the unused part of a tile is reduced by: $0.66 \cdot 77.1\% = 50.9\%$. Leakage in the used part of a tile is reduced by: $0.66 \cdot 0.75 \cdot 40\% = 19.8\%$, where the “0.75” represents the average fraction of used interconnect that may

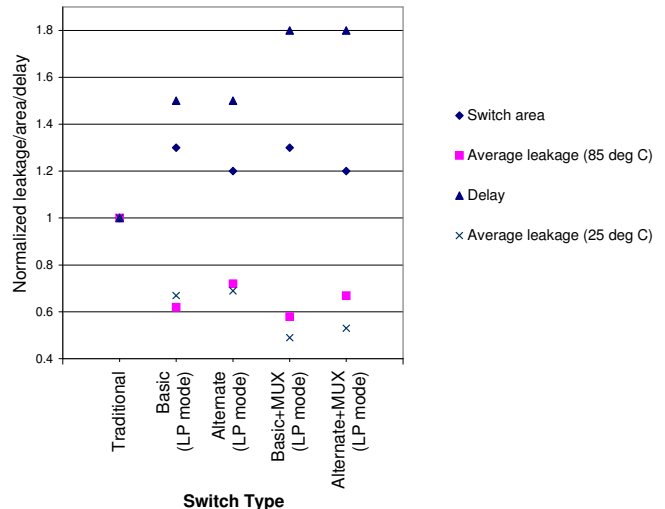


Fig. 18. Leakage, area, and speed of switch designs.

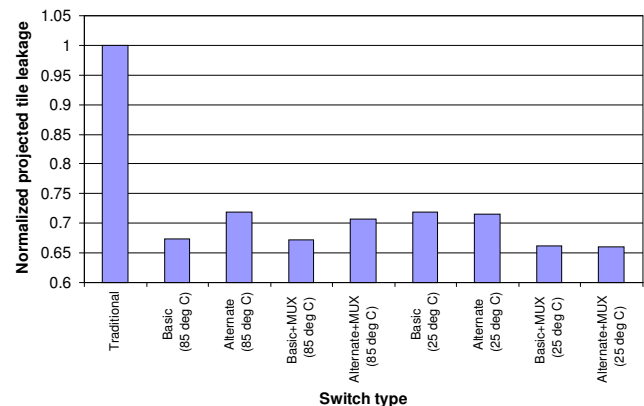


Fig. 19. Overall leakage reduction in an FPGA tile incorporating the proposed switch design versus a tile incorporating a traditional switch design.

be slowed down and operate in low-power mode (from the slack analysis in Section IV). Given these partial results, the projected reduction in overall tile leakage for deploying the *basic* switch design is: $0.4 \cdot 50.9\% + 0.6 \cdot 19.8\% = 32\%$. Applying the same analysis to all the switch types produces the data in Figure 19. The overall benefit ranges from 28% to nearly 35% leakage reduction, depending on the switch design. Note that the data in the figure corresponds to use of the variant sleep mode, which consistently offers better leakage.

VI. CONCLUSIONS

Static and dynamic power dissipation in FPGAs is dominated by consumption in the interconnection fabric, making low-power interconnect a mandatory feature of future low-power FPGAs. In this paper, we have proposed a number of new FPGA routing switch designs that can be programmed to operate in high-speed, low-power, or sleep mode. Each of the proposed designs offers a different power/area/speed trade-off. At high temperature, leakage reductions in low-power versus high-speed mode range from 28-42%. Depending on

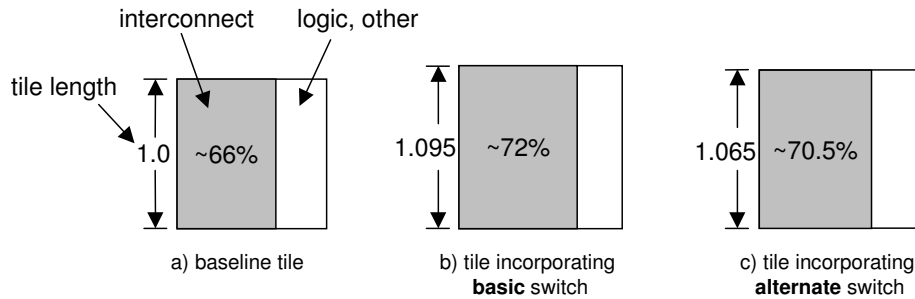


Fig. 17. Projected tile area breakdown for traditional and proposed switch types.

the design selected, such leakage reductions come with varying levels of performance and/or area overhead. At low temperature, leakage reductions range from 30-52% in low-power versus high-speed mode. Sleep mode leakage reductions range from 61-79% relative to high-speed mode. All of the proposed designs reduce dynamic power by up to 28-31%. An analysis of the timing slack in commercial FPGA benchmark circuits showed that the proposed switch designs are well-motivated. A majority of routing switches can be slowed down, and operated 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.

ACKNOWLEDGMENT

The authors thank Navid Azizi for his assistance with the 70nm technology models. We also thank the anonymous reviewers for their comments, which have considerably strengthened the manuscript.

REFERENCES

- [1] *International Technology Roadmap for Semiconductors (ITRS)*, <http://www.itrs.org>, 2002.
- [2] V. George and J. Rabaey, *Low-Energy FPGAs: Architecture and Design*. Boston, MA: Kluwer Academic Publishers, 2001.
- [3] I. Kuon and J. Rose, "Measuring the gap between FPGAs and ASICs," *IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, pp. 203–215, Feb. 2007.
- [4] L. Shang, A. Kaviani, and K. Bathala, "Dynamic power consumption in the Virtex-II FPGA family," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2002, pp. 157–164.
- [5] T. Tuan and B. Lai, "Leakage power analysis of a 90nm FPGA," in *IEEE Custom Integrated Circuits Conference*, San Jose, CA, 2003, pp. 57–60.
- [6] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2003, pp. 175–184.
- [7] 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*, Monterey, CA, 2004, pp. 23–30.
- [8] K. Poon, A. Yan, and S. J. E. Wilton, "A flexible power model for FPGAs," in *International Conference on Field-Programmable Logic and Applications*, Montpellier, France, 2002, pp. 312–321.
- [9] *IGLOO Low-Power Flash FPGAs General Description*, Actel, Corp., Mountain View, CA, 2007.
- [10] *Spartan-3 FPGA Data Sheet*, Xilinx, Inc., San Jose, CA, 2004.
- [11] J. Anderson and F. Najm, "A novel low-power FPGA routing switch," in *IEEE Custom Integrated Circuits Conference*, Orlando, FL, 2004, pp. 719–722.
- [12] —, "Low-power programmable FPGA routing circuitry," in *IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2004, pp. 602–609.
- [13] 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*, Feb. 2003, pp. 305–327.
- [14] 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*, New Orleans, LA, 2002, pp. 480–485.
- [15] 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*, Honolulu, HI, 1998, pp. 44–45.
- [16] R. 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*, San Jose, CA, 2002, pp. 125–128.
- [17] D. Lewis, V. Betz, D. Jefferson, A. Lee, C. Lane, P. Leventis, S. Marquardt, C. McClintock, B. Pedersen, G. Powell, S. Reddy, C. Wysocki, R. Cliff, and J. Rose, "The Stratix routing and logic architecture," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2003, pp. 12–20.
- [18] G. Lemieux, "Routing architecture for field-programmable gate arrays," in *Ph.D. Thesis*. Toronto, Ontario, Canada: Department of Electrical and Computer Engineering, University of Toronto, 2003.
- [19] F. Li, Y. Lin, L. He, and J. Cong, "Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2004, pp. 42–50.
- [20] F. Li, Y. Lin, and L. He, "FPGA power reduction using configurable dual-Vdd," in *ACM/IEEE Design Automation Conference*, San Diego, CA, 2004, pp. 735–740.
- [21] A. Gayasen, K. Lee, N. Vijaykrishnan, M. Kandemir, M. Irwin, and T. Tuan, "A dual-Vdd low power FPGA architecture," in *International Conference on Field-Programmable Logic and Applications*, Antwerp, Belgium, 2004, pp. 145–157.
- [22] F. Li and L. He, "Vdd programmability to reduce FPGA interconnect power," in *IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, 2004, pp. 760–765.
- [23] B. Calhoun, F. Honore, and A. Chandrakasan, "Design methodology for fine-grained leakage control in MTCMOS," in *ACM/IEEE International Symposium on Low-Power Electronics and Design*, Seoul, Korea, 2003, pp. 104–109.
- [24] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. Irwin, and T. Tuan, "Reducing leakage energy in FPGAs using region-constrained placement," in *ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, Monterey, CA, 2004, pp. 51–58.
- [25] L. Ciccarelli, A. Lodi, and R. Canegallo, "Low leakage circuit design for FPGAs," in *IEEE Custom Integrated Circuits Conference*, Orlando, FL, 2004, pp. 715–718.
- [26] 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*, vol. 21, no. 3, pp. 306–318, Mar. 2002.
- [27] V. Betz, "Architecture and CAD for the speed and area optimization of FPGAs," in *Ph.D. Thesis*. Toronto, Ontario, Canada: Department of Electrical and Computer Engineering, University of Toronto, 1998.

- [28] M. Hutton, V. Chan, P. Kazarian, V. Maruri, T. Ngai, J. Park, R. Patel, B. Pedersen, J. Schleicher, and S. Shumarayev, "Interconnect enhancements for a high-speed PLD architecture," in *ACM/SIGDA International Symposium on FPGAs*, Monterey, CA, 2002, pp. 3–10.
- [29] *Berkeley Predictive Technology Model* (<http://www.device.eecs.berkeley.edu/~ptm/>), University of California, Berkeley, 2004.
- [30] N. Azizi and F. Najm, "An asymmetric SRAM cell to lower gate leakage," in *IEEE International Symposium on Quality Electronic Design*, San Jose, CA, 2004, pp. 534–539.
- [31] C. 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*, Seoul, Korea, 2003, pp. 6–9.
- [32] *Virtex-5 FPGA Data Sheet*, Xilinx, Inc., San Jose, CA, 2007.
- [33] A. Agarwal, C. Kim, S. Mukhopadhyay, and K. Roy, "Leakage in nanoscale technologies: Mechanisms, impact and design considerations," in *ACM/IEEE Design Automation Conference*, San Diego, CA, 2004, pp. 6–11.
- [34] S. Narendra, S. Borkar, V. De, D. Antoniadis, and A. Chandrakasan, "Scaling of stack effect and its application for leakage reduction," in *ACM/IEEE International Symposium on Low Power Electronics and Design*, Huntington Beach, CA, 2001, pp. 195–200.

PLACE
PHOTO
HERE

Farid N. Najm (S'85-M'89-SM'96-F'03) is a Professor in the department of electrical and computer engineering (ECE) at the University of Toronto, Ontario, Canada. He received the B.E. degree in electrical engineering from the American University of Beirut in 1983 and the Ph.D. degree in ECE from the University of Illinois at Urbana-Champaign (UIUC) in 1989. From 1989 to 1992, he worked with Texas Instruments, Dallas, Texas. He then joined the ECE department at UIUC as an Assistant Professor and became Associate Professor in 1997. In 1999, he joined the ECE department at the University of Toronto, where he is now Professor, and where he has served as department Vice-Chair from 2004 to 2007.

Dr. Najm is a Fellow of the IEEE, and is Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He has received an IEEE Transactions on CAD Best Paper Award, an NSF Research Initiation Award, an NSF CAREER Award, and was an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 1997 to 2002. He serves on the executive committee of the International Symposium on Low-Power Electronics and Design (ISLPED), and has served on the technical committees of various conferences, including ICCAD, DAC, CICC, ISQED, and ISLPED. His research is on CAD for VLSI, with an emphasis on circuit level issues related to power, timing, variability, and reliability.

PLACE
PHOTO
HERE

Jason H. Anderson (S'96-M'05) is a Principal Engineer and Manager at Xilinx, Inc., and an Adjunct Professor of electrical and computer engineering (ECE) at the University of Toronto (U of T), Ontario, Canada. He received the Ph.D. and M.A.Sc. degrees in ECE from the U of T in 2005 and 1997, respectively. He received the B.Sc. degree in computer engineering from the University of Manitoba, Winnipeg, Canada, in 1995. In 1997, he joined the field-programmable gate array (FPGA) implementation tools group at Xilinx in San Jose,

California, where he developed placement and routing tools for Xilinx Virtex and Spartan series FPGAs. He currently manages a group at Xilinx focused on strategic research and development projects. His research interests include all aspects of computer-aided design (CAD) and architecture for FPGAs.

Dr. Anderson received the Ross Freeman Award for Technical Innovation, the highest innovation award given by Xilinx, for his contributions to the Xilinx placer technology in 2000. He was also awarded the Natural Sciences and Engineering Research Council (NSERC) of Canada Postgraduate Scholarship in 2001, and the Ontario Graduate Scholarship in 2003 and 2004. He has authored numerous papers in refereed conferences and journals, and holds over a dozen issued U.S. patents. He serves on the technical program committees of the IEEE International Conference on Computer-Aided Design (ICCAD) and the IEEE International Conference on Field Programmable Logic and Applications (FPL).