A Novel Low-Power FPGA Routing Switch

Jason H. Anderson and Farid N. Najm
Department of Electrical and Computer Engineering, University of Toronto
Toronto, Ontario, Canada

Abstract

We propose a new programmable FPGA routing switch that can operate in three different modes: high-speed, low-power or sleep. High-speed mode offers similar power and performance to a traditional routing switch. In low-power mode, power is reduced at the expense of speed. Leakage power is reduced 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. The applicability of the new switch 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.

I. Introduction

Managing power consumption, specifically leakage power, has become a major concern for semiconductor vendors and customers [1]. Field-programmable gate arrays (FPGAs) are a popular choice for digital circuit implementation; however, they are less power-efficient than custom ASICs as a result of the overhead required to provide programmability [2]. Low-power is therefore likely to be a key objective in the design of future FPGAs.

Several recent studies have considered FPGA power consumption and found that 60-70% of dynamic and static (leakage) power is dissipated in the interconnection fabric [3, 4, 5, 6]. 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 circuitry.

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 terminal 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 are proportional to total transistor width, and programmable interconnect accounts for the majority of transistor width in FPGAs [6].

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, an FPGA design uses only a portion of the underlying FPGA hardware and that leakage is dissipated in both the *used* and *unused* part of an 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.

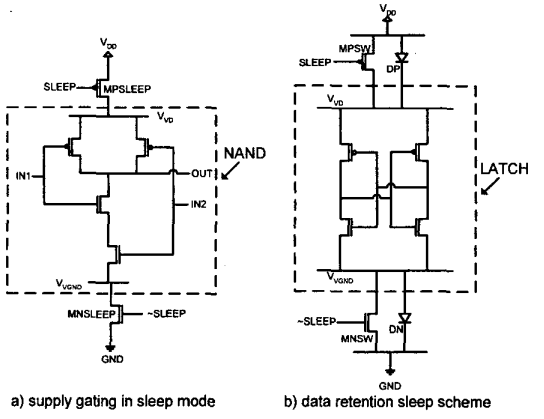
The chief role of interconnect in FPGA power consumption makes it a high-leverage target for power optimization. In this paper, we present a novel FPGA routing switch design that offers reduced leakage and dynamic power dissipation. It can be programmed to operate in one of three modes: high-speed, low-power 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.

II. Background and Related Work

A variety of techniques for leakage optimization in ASICs have been proposed in the literature; a detailed overview can be found in [7]. Our proposed switch design draws 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 [8], 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 thus, the technique cannot be directly applied to data storage elements.

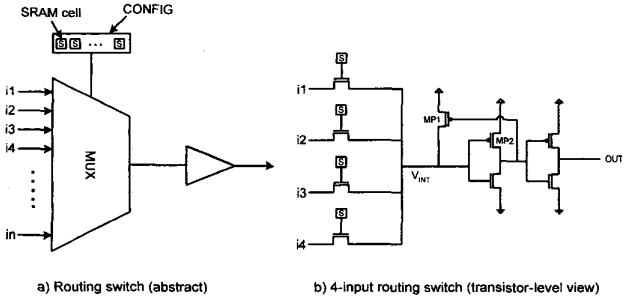
A way of dealing with the data retention issue was proposed in [9] and is shown in Fig. 1(b). Two diodes, DP and DN, are introduced in parallel with the sleep transistors. In active mode, the virtual V_{DD} voltage (V_{VD}) and the virtual ground voltage (V_{VGND}) are equal to rail V_{DD} and GND, respectively. 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) [7]. 2) Gate oxide leakage decreases superlinearly with reductions in gate/source potential difference (V_{GS}) [10].

FPGA interconnect is composed of variable length wire segments and programmable routing switches. Fig. 2(a) shows a typical buffered FPGA routing switch [6, 11]. It consists of a multiplexer, a buffer and SRAM configuration bits. 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. Programmabil-



a) supply gating in sleep mode b) data retention sleep scheme

Fig. 1. Sleep leakage reduction techniques [8, 9].



a) Routing switch (abstract) b) 4-input routing switch (transistor-level view)

Fig. 2. Traditional routing switch: abstract and transistor-level views.

ity 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. 2(b) [6]. NMOS transistor trees are used to implement multiplexers in FPGAs [11]. 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 [6]. 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.

A number of recent studies have considered optimizing FPGA power consumption at the architecture or circuit level [2, 12, 5, 13]. To our knowledge, the only work to specifically address leakage in FPGA interconnect is [6], which applies well-known leakage reduction techniques to interconnect multiplexers. In particular, [6] 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 design involves changes to the switch buffer (not the multiplexer) and is therefore compatible with the ideas proposed in [6].

III. Low-Power Routing Switch Design

The proposed switch design is 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. 2(b)).
2. There exists sufficient timing slack in typical FPGA designs to allow a considerable 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 commercial FPGA [14]. 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. 3. The switch includes NMOS and PMOS sleep transistors in parallel (MNX and MPX). It 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 essence, the new switch design mimics the programmable dual-V_{DD} concept proposed in [12], 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 design is that it has 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 design quite practical. We anticipate it can be deployed in place of most existing routing switches in commercial FPGAs.

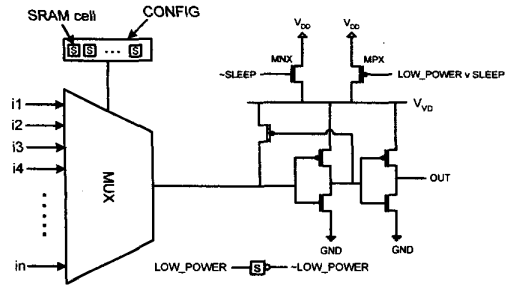


Fig. 3. New programmable low-power FPGA routing switch.

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 (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 considerable.

To motivate our switch design, we evaluated the timing slack in 22 routed industrial designs implemented in the Xilinx Spartan-3 FPGA [14]. We used the Xilinx place and route tools to generate a performance-optimized layout for each design. Slack was evaluated relative to an aggressive but achievable (clock period) timing constraint, determined independently for each design to be within 3% of the design's maximum performance. By using such 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 [15], 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.

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. 4. 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. The right-most set of bars in Fig. 4 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.

V. Results

All HSPICE simulation results in this paper were produced at 85°C using the Berkeley Predictive Technology Models (BPTM) for a 70*n*m technology [16]. The models were enhanced to account for gate oxide leakage as decribed in [17], and correspond to an oxide thickness of 1.2*n*m [1].

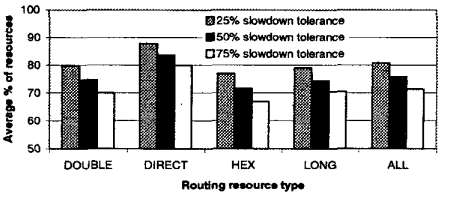


Fig. 4. Timing slack in industrial FPGA designs.

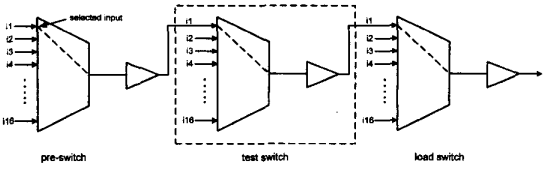


Fig. 5. Baseline test platform.

To study the proposed switch, we first developed a 16-input traditional routing switch (see Fig. 2(b)), representative of those in current commercial FPGAs[1]. The buffer was sized for equal rise and fall times, with the second inverter stage being 3 times larger than the first stage. For the 16-to-1 multiplexer, we selected a design that reflects a reasonable speed/area trade-off. The design is symmetric, requires 6 SRAM cells, and has a depth of three NMOS transistors from any input to its output. We then transformed the traditional switch into the proposed switch. Transistor MPX was sized to provide (high-speed mode) performance within 5% of the traditional switch. We sized transistor MNX to achieve 50% slower speed performance in low-power vs. high-speed mode.

To study the power characteristics of the proposed switch, we simulate the conditions of a used switch in an actual FPGA using the test platform shown in Fig. 5. Power and performance measurements are made for the second switch, labeled "test switch" in Fig. 5. Our power measurements include current drawn from all sources, including gate oxide leakage in the multiplexer and sleep transistors. However, 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., [18]).

We first examine the difference in leakage power in high-speed vs. low-power 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 in both platforms. The results are shown in Fig. 6. The horizontal axis shows the percentage reduction in leakage power in the low-power switch vs. the high-speed switch. The vertical axis shows

[1] A 16-input switch was selected as it is similar to the switches driving double-length segments in Xilinx Spartan-3 [14].

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

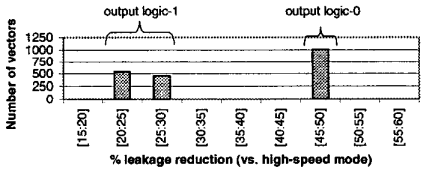


Fig. 6. Leakage reduction results (high-speed vs. low-power).

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, low-power mode offers a 36% reduction in leakage power compared with 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%.

Routing conductors in FPGAs have multiple used and unused switches attached to them. 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 results for all scenarios considered are summarized in Table I. 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.

Lastly, we evaluated the dynamic power benefits of low-power mode. We found that 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 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 switch (vs. a 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. 2(b), which will reduce the area overhead of the proposed switch. Pessimistically, we can assume the new switch increases an FPGA's interconnect area by 30% and that interconnect accounts for ~2/3 (66%) of an FPGA's base tile area [6]. Given this, the overall tile area increase to include the new 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.

TABLE I
LEAKAGE POWER REDUCTION RESULTS SUMMARY.

Test scenario	Avg. % power reduction versus high-speed mode
low-power mode (single fanout)	36.0%
sleep mode	60.8%
low-power mode (additional unused fanout)	39.7%
low-power mode (additional used fanout)	38.7%

VI. 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 a new FPGA routing switch 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%. We showed that timing slack in typical FPGA designs permits the majority of switches to operate in low-power mode. The switch requires only minor changes to a traditional FPGA routing switch and has no impact on router complexity, making it easy to deploy in current commercial FPGAs.

References

[1] *2002 International Technology Roadmap for Semiconductors (ITRS).*
[2] V. George and J. Rabaey. *Low-Energy FPGAs: Architecture and Design.* Kluwer Academic Publishers, Boston, MA, 2001.
[3] L. Shang, A. Kaviani, and K. Bathala. Dynamic power consumption the Virtex-II FPGA family. In *ACM FPGA*, 2002.
[4] T. Tuan and B. Lai. Leakage power analysis of a 90nm FPGA. In *IEEE CICC*, 2003.
[5] F. Li, D. Chen, L. He, and J. Cong. Architecture evaluation for power-efficient FPGAs. In *ACM FPGA*, 2003.
[6] A. Rahman and V. Polavarapuv. Evaluation of low-leakage design techniques for field-programmable gate arrays. In *ACM FPGA*, 2004.
[7] 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.
[8] 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 DAC*, 2002.
[9] K. Kumagai et. al. A novel powering-down scheme for low Vt CMOS circuits. In *IEEE Symp. on VLSI Circuits*, 1998.
[10] R.K. Krishnamurthy, A. Alvandpour, V. De, and S. Borkar. High-performance and low-power challenges for sub-70nm microprocessor circuits. In *IEEE CICC*, 2002.
[11] G. Lemieux. Design of interconnection networks for programmable logic devices. In *Ph.D. Thesis*. ECE Department, University of Toronto, 2003.
[12] F. Li, Y. Lin, L. He, and J. Cong. Low-power FPGA using pre-defined dual-Vdd/dual-Vt fabrics. In *ACM FPGA*, 2004.
[13] B.H. Calhoun, F.A. Honore, and A. Chandrakasan. Design methodology for fine-grained leakage control in MTCMOS. In *ACM/IEEE ISLPED*, 2003.
[14] Xilinx, Inc., San Jose, CA. *Spartan-3 FPGA Data Sheet*, 2003.
[15] Q. Wang and S. B. K. Vrudhula. Algorithms for minimizing standby power in deep submicrometer, dual-Vt CMOS circuits. *IEEE Transactions on CAD*, 21(3):306–318, March 2002.
[16] *http://www.device.eecs.berkeley.edu/~ptm/.*
[17] N. Azizi and F.N. Najm. An asymmetric SRAM cell to lower gate leakage. In *IEEE ISQED*, 2004.
[18] 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 ISLPED*, 2003.