

Vdd Programmability to Reduce FPGA Interconnect Power

Fei Li, Yan Lin and Lei He

Electrical Engineering Department
University of California, Los Angeles, CA 90095

ABSTRACT

Power is an increasingly important design constraint for FPGAs in nanometer technologies. Because interconnect power is dominant in FPGAs, we design Vdd-programmable interconnect fabric to reduce FPGA interconnect power. There are three Vdd states for interconnect switches: high Vdd, low Vdd and power-gating. We develop a simple design flow to apply high Vdd to critical paths and low Vdd to non-critical paths and to power gate unused interconnect switches. We carry out a highly quantitative study by placing and routing benchmark circuits in 100nm technology to illustrate the power saving. Compared to single-Vdd FPGAs with optimized but non-programmable Vdd level for the same target clock frequency, our new FPGA fabric on average reduces interconnect power by 56.51% and total FPGA power by 50.55%. Due to the highly low utilization rate of routing switches, majority of the power reduction is achieved by power gating unused routing buffers. In contrast, recent work that considers Vdd programmability only for logic fabric reduces total FPGA power merely by 14.29%. To the best of our knowledge, it is the first in-depth study on Vdd programmability for FPGA interconnect power reduction.

1. INTRODUCTION

FPGA is an attractive design platform due to its low NRE (non-recurring engineering) cost and the short time to market. However, the power efficiency of FPGA is known to be much lower than that of ASIC. FPGA power modeling and optimization have drawn increasing attention. [1, 2] present power evaluation frameworks for generic parameterized FPGA architectures and show that both interconnect and leakage power are significant power components for existing FPGAs. [3] analyzes the leakage power of a commercial FPGA architecture in 90nm technology and quantifies the leakage power challenge for nanometer FPGAs. Several FPGA power reduction techniques have also been proposed. [4] introduces an inversion method to reduce active leakage power of multiplexers in FPGA fabrics. [5] investigates power-gating and applies region-constrained placement to reduce leakage power of unused FPGA logic blocks. [6] proposes pre-defined dual-Vdd/dual-Vt FPGA fabrics to reduce both dynamic and leakage power. However, the pre-defined dual-Vdd fabric cannot always achieve effective power reduction due to the lack of flexibility to customize dual-Vdd layout pattern for different applications. Therefore, configurable dual-Vdd

*This paper is partially supported by NSF CAREER award CCR-0401682 and NSF grant CCR-0306682. We used computers donated by Intel. Address comments to lhe@ee.ucla.edu.

for FPGAs is proposed in [7] such that the Vdd level in any logic block can be programmed for different applications with negligible performance decay and a significant logic power reduction.

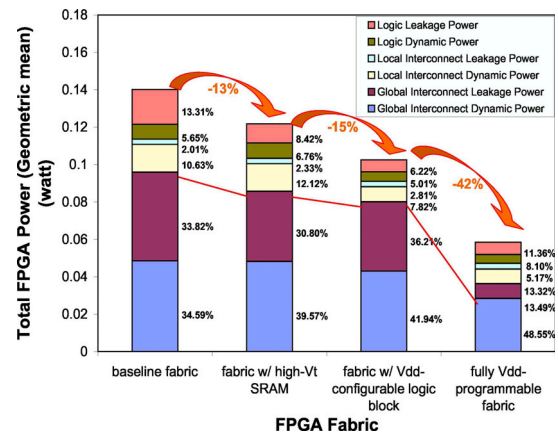


Figure 1: FPGA power breakdown after applying the existing power optimization techniques as well as the new technique in this paper. Power (and power reduction percentage) is reported in Geometric Mean over the MCNC benchmark suite.

However, the existing work mainly reduces logic block power [5, 6, 7]. The total FPGA power reduction is either unreported or very limited because the dominant interconnect power has not been optimized. As an example, we use the first three bars in Figure 1 to show the power breakdown of a conventional (also baseline) FPGA architecture [8] and the component-wise power saving after applying techniques in [6] and [7] subsequently. The logic power is the power of LUTs, flip-flops and MUXes in logic blocks. The local interconnect power is the power of internal routing wires and buffers within logic blocks. Routing wires outside logic blocks, programmable interconnect switches in routing channels and their configuration SRAM cells contribute to global interconnect power. The figure shows that global interconnect power, 68% of total FPGA power, is dominant before applying any power reduction technique. By using high-Vt SRAMs for configuration bits in LUTs and interconnects [6] to reduce leakage, the total FPGA power is reduced by 13% compared to the baseline FPGA. When the configurable dual-Vdd for logic blocks [7] is further applied, the power of logic and local interconnect is reduced significantly and the total FPGA power is reduced by 15% compared to the FPGA with high-Vt SRAMs for configuration bits. As a result of the power reduction techniques in [6, 7], the portion of global interconnect power increases to 78% of the total power. Clearly, interconnect power minimization is the

key to effectively reduce total FPGA power and is the focus of this paper.

There is limited study on interconnect power reduction for FPGAs. [9] introduces a hierarchical interconnect structure and applies low-swing circuits to long interconnects. However, low-swing circuits are complicated to design and are less robust. They have not been widely used in either full custom designs or FPGA designs. Similar to low-swing circuit that applies reduced Vdd to FPGA interconnects to reduce power, in this paper we selectively apply low-Vdd to interconnect circuits such as routing and connection switches. Our interconnect circuits are still the same as conventional FPGA circuits but with a reduced Vdd level and the effort to design special low-swing circuits as in [9] is avoided. The Vdd selection for different applications is obtained by programmable dual-Vdd. In addition, we also show that the utilization rate of interconnect switches is extremely low due to the interconnection programmability ($\sim 12\%$ when using the smallest FPGA array that just fits for a given application). Therefore we develop our programmable dual-Vdd technique with capability of power gating for extra leakage power reduction. In contrast to [7] where programmable Vdd is used only for logic blocks, we apply programmable dual-Vdd to both logic blocks and interconnects, and name the resulting FPGA fabric as fully Vdd-programmable FPGA fabric. On average, the new fabric reduces total power by 50.55%.

The rest of the paper is organized as follows. Section 2 gives background knowledge. Section 3 presents the fully Vdd-programmable fabric and the underlying circuit design. Section 4 discusses the CAD algorithms and design flow. Section 5 presents the experimental results and Section 6 concludes this paper.

2. BACKGROUND

2.1 Island Style Interconnect Fabric

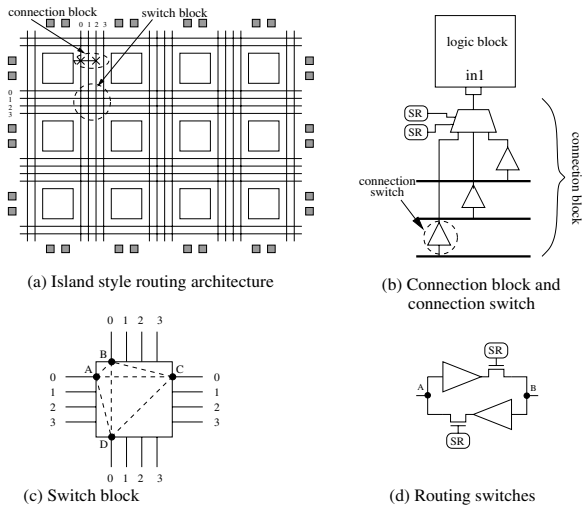


Figure 2: (a) Island style routing architecture; (b) Connection block; (c) Switch block; (d) Routing switches.

Interconnects consume most of the area and power of FPGAs. Figure 2 (a) shows the traditional island style routing architecture. The logic blocks are surrounded by routing channels consisting of wire segments. The input and output pins of a logic block can be connected to the wire segments in the surrounding channels via a *connection block* as shown in Figure 2 (b). There is a routing

switch block at each intersection of a horizontal channel and a vertical channel. Figure 2 (c) shows a subset switch block [10], where the incoming track can be connected to the outgoing tracks with the same track number¹. The connections in a switch block (represented by the dashed lines in Figure 2 (c)) are programmable routing switches. Routing switches can be implemented by tri-state buffers and each connection needs two tri-state buffers so that it can be programmed independently for either direction. FPGA interconnect power is the power consumed by the programmable switches in both switch blocks and connection blocks and the routing wires driven by the switches.

In this paper, we decide the routing channel width W in the same way as the architecture study in [8], i.e., $W = 1.2W_{min}$, where W_{min} is the minimum channel width required to route the given circuit successfully. The channel width W represents a “low-stress” routing situation that usually occurs in commercial FPGAs for ‘average’ circuits. For each given circuit, we also use the smallest FPGA array that just fits the circuit for logic cell placement. We assume that the LUT size is 4 and cluster (i.e., logic block) size is 10, and use interconnect structure with 100% tri-state buffers (rather than a mix of buffers and pass transistors).

2.2 Configurable Dual-Vdd Technique

Dual-Vdd technique makes use of the timing slack in the circuit to minimize power. It applies high supply voltage (VddH) to devices on critical paths to maintain the performance, and applies low supply voltage (VddL) to devices on non-critical paths to reduce power. The configurable dual-Vdd technique has been introduced in [7] to provide Vdd programmability for logic blocks (see Figure 3). The Vdd-programmable logic block is obtained by inserting two extra PMOS transistors, called *power switches*, between the conventional logic block and the dual-Vdd power rails for Vdd selection and power gating. The same Vdd-programmable logic block in [7] is also assumed in this paper, but we further explore Vdd-programmable FPGA interconnects.

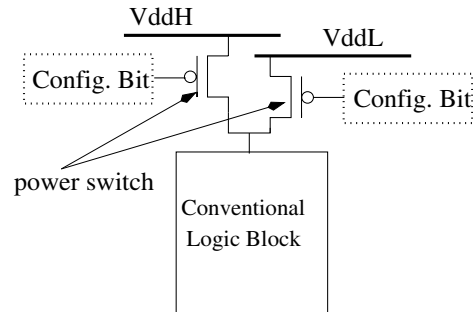


Figure 3: Vdd-programmable logic block.

3. VDD-PROGRAMMABILITY FOR INTERCONNECT FABRIC

We apply programmable dual-Vdd to each interconnect switch (either a routing switch or a connection switch). Our Vdd-programmable routing switch is shown in Figure 4 (a). The right part of the circuit is the Vdd-programmable routing switch. For the tri-state buffer in the routing switch, we insert two PMOS transistors M3 and M4 between the tri-state buffer and VddH, VddL power rails, respectively. Similar to [7], turning off one of the two power switches can

¹Without loss of generality, we assume subset switch block in our study.

select a Vdd level for the routing switch. In this case, we only need one configuration cell and name this scheme as *one-bit control* implementation of programmable-Vdd. Considering the extremely low interconnect utilization rate (an average of 11.90%² as shown in Table 1 for MCNC benchmark set), we can turn off both power switches and power gate an unused routing switch. In that case, we need two configuration cells to provide three Vdd states: high Vdd, low Vdd and power-gating. We name this scheme as the *two-bit control* implementation of programmable-Vdd.

The two-bit control implementation is very attractive because our SPICE simulation shows that power-gating of the routing switch can reduce its leakage power by a factor of over 300. We also consider the power and delay overhead associated with the power switch insertion. The dynamic power overhead is almost ignorable (See energy per switch in Table 2). This is because the power switches stay either ON or OFF and there is no charging and discharging at their source/drain capacitors. The main power overhead is the leakage power of the extra configuration cells for Vdd selection. We use the same high-Vt SRAM cells in [6] to reduce configuration cell leakage. Further, the Vdd-programmable routing switch has an increased delay compared to the conventional routing switch because the power switches are inserted between the buffer and power supply. We properly size the power switches for the tri-state buffer to achieve a bounded delay increase. For a routing architecture with all wire segments spanning four logic blocks, we assume 7X minimum width tri-state buffers and achieve 6% delay increase by inserting 25X minimum width power switches when compared to conventional Vdd non-programmable routing switches.

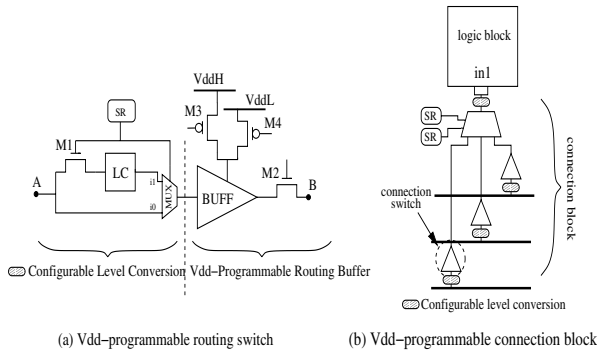


Figure 4: (a) Vdd-programmable routing switch; (b) Vdd-programmable connection block. (SR stands for SRAM cell and LC stands for level converter. The same configurable level conversion circuit is used in both (a) and (b))

The left part of the circuit in Figure 4 (a) is the level converter. When we apply dual-Vdd to the routing switches, we need a level converter whenever a VddL routing switch drives a VddH routing switch. Because each routing switch can be programmed to either VddH or VddL, a level converter must be pre-inserted in the fabric for any pair of wire segments that can be connected through a routing switch block. We insert the level converter right before the routing switch and use a multiplexer to either select this level converter or bypass it. The transistor M1 is used to prevent signal

²Note that we use the minimum FPGA array that just fits the application circuit. In reality, the chip size can be significantly larger than necessary and the interconnect switch utilization can be even lower.

circuit	total interconnect switches	unused interconnect switches	utilization rate
alu4	36478	31224	14.40%
apex4	43741	37703	13.80%
bigkey	63259	57017	9.87%
clma	653181	593343	9.16%
des	87877	79932	9.04%
diffeq	42746	36974	13.50%
dsip	75547	70138	7.16%
elliptic	140296	125800	10.33%
ex5p	45404	39288	13.47%
frisc	238853	216993	9.15%
misex3	39928	33819	15.30%
pdc	268167	238610	11.02%
s298	43725	37641	13.91%
s38417	243315	216577	10.99%
s38584	195363	174460	10.70%
seq	61344	53173	13.32%
spla	153235	134991	11.91%
tseng	29051	25026	13.85%
Avg.			11.90%

Table 1: Utilization rate of interconnect switches.

Vdd	routing switch delay (ns)		energy per switch (Joule)	
	without Vdd programmability	with Vdd programmability (increase %)	without Vdd programmability	with Vdd programmability
1.3v	5.90E-11	6.26E-11 (6.00%)	3.3049E-14	3.2501E-14
1.0v	6.99E-11	7.42E-11 (6.17%)	1.6320E-14	1.6589E-14

Table 2: Delay and power of a Vdd-programmable routing switch. We use 7X minimum width tri-state buffer for routing switches and 25X minimum width PMOS transistor for power switches.

transitions from propagating through the level converter when it is bypassed, and therefore eliminate the dynamic power of an unused level converter. Only one configuration bit is needed to realize the level converter selection and signal gating for unused level converters. We use the same asynchronous level converter circuit in [7, 11] and size the level converter as in [6] to achieve a bounded delay with minimum power consumption. The leakage of the sized level converter is around 6.4% of the leakage for a 7X min width tri-state routing buffer.

Another type of routing resources are the connection blocks [8] as shown in Figure 4 (b). The multiplexer-based implementation chooses only one track in the channel and connects it to the logic block input pin. The buffers between the routing track and the multiplexer are connection switches. Programmable-Vdd is also applied to the connection switch and, similarly, the configurable level conversion circuits are inserted before the connection switch. Because we apply programmable-Vdd to both logic blocks and programmable interconnect switches, it is possible that a VddL connection switch connects to a VddH logic block. To ensure that there is supply level conversion for this type of connection, we also insert the configurable level conversion circuit before each logic block input pin.

4. DESIGN FLOW

We develop a simple CAD flow to leverage the fully Vdd-programmable fabric (See Figure 5). Starting with a single-Vdd gate level netlist, we apply technology mapping and timing driven packing [8] to obtain the single-Vdd cluster-level netlist. We then perform single-Vdd timing-driven placement and routing by VPR [8] and generate the basic circuit netlist (BC-netlist). The BC-netlist is defined in [2] and it is annotated with capacitance, resistance and switching activity for each node. After that, dual-Vdd assignment is performed on the BC-netlist and Vdd configuration is decided

for each used logic block and programmable interconnect switches. Power gating is applied to all unused logic blocks and programmable switches. Finally, we perform the power and delay evaluation for the dual-Vdd design.

Compared to the traditional FPGA design flow, we only introduce one extra step (highlighted in Figure 5) and none of the original steps is changed. This is because the full Vdd-programmability in our new FPGA fabric eliminates any extra layout constraint in placement and routing. In contrast, both pre-defined dual-Vdd fabric [6] and partially Vdd-programmable fabric [7] significantly complicate the CAD flow by either adding extra design steps or changing the original placement/routing.

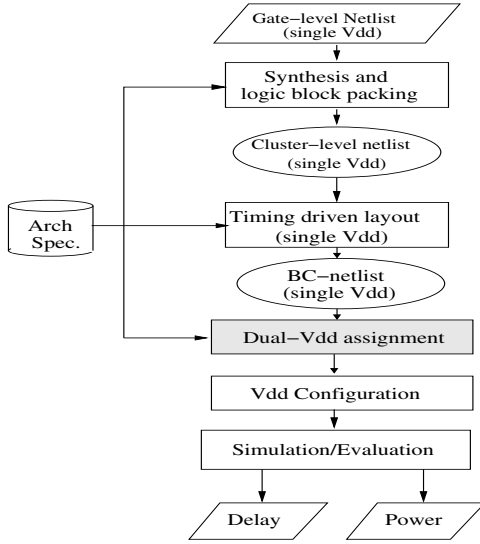


Figure 5: Design flow for the fully Vdd-programmable FPGA fabric.

We apply sensitivity-based algorithm similar to that in [12] to perform the dual-Vdd assignment. A circuit element is defined as either a logic block or a routing/connection switch. Power sensitivity of a circuit element is the FPGA power consumption change due to the Vdd level change for that circuit element.

Assuming that all the circuit elements are initially assigned to VddH, we iteratively carry out the following steps. Timing analysis is performed to obtain the circuit elements on the path with the largest timing slack. We then calculate the power sensitivity of those circuit elements and assign VddL to the element with the largest power sensitivity. The configurable level converter can be enabled as needed. After updating the circuit timing, we accept the assignment if the critical path delay does not increase. Otherwise, we reject the assignment and restore the the circuit element supply voltage to VddH. In either case, the circuit element will be marked as ‘tried’ and will not be re-visited in subsequent iterations. After the dual-Vdd assignment, we obtain a dual-Vdd BC-netlist without degrading the system performance.

5. EXPERIMENTAL RESULTS

In this section, we conduct experiments on the MCNC benchmark set and compare three architectures, arch-SV, arch-PV-logic and arch-PV-fpga. *arch-SV* [6] scales down the supply voltage Vdd as well as the transistor threshold voltage Vt for the entire FPGA. It achieves power reduction at the cost of performance loss. *arch-PV-logic* is the FPGA architecture using programmable dual-Vdd

for logic blocks [7]. *arch-PV-fpga* is our fully Vdd-programmable fabric applying programmable dual-Vdd to both logic blocks and interconnects.

5.1 Ideal Low Vdd Utilization Rate

In Table 3, we present the ideal low Vdd utilization rate obtained by the dual-Vdd assignment algorithm for MCNC benchmarks. 85% of used interconnect switches and 75% of used logic blocks are assigned to VddL without increasing the circuit delay. This shows that circuits implemented in a single-Vdd FPGA can have large amount of surplus timing slack, and intuitively justifies that the application of programmable dual-Vdd to both interconnects and logic blocks can greatly reduce FPGA power. Certainly, power evaluation needs to be performed to obtain the power reduction considering the power overhead of level converters. The results presented in the rest of this paper all consider power and delay impact of programmable dual-Vdd and level converters.

circuit	# of logic blocks	# of I/O blocks	VddL interconnect switches (%)	VddL logic blocks (%)
alu4	162	22	73.64	74.69
apex2	213	41	75.57	47.42
apex4	134	28	72.42	62.69
bigkey	294	426	85.63	89.12
clma	1358	144	86.68	80.78
des	218	501	86.92	75.69
diffeq	195	103	94.72	83.59
dsip	588	426	89.12	53.70
elliptic	666	245	95.84	90.74
ex1010	513	20	75.74	74.24
ex5p	194	71	81.03	62.60
frisc	731	136	99.48	95.13
misex3	181	28	75.07	58.17
pdcc	624	56	81.15	70.07
s298	266	10	90.64	82.81
s38417	982	135	93.32	88.55
s38584	1046	342	98.09	96.88
seq	274	76	72.80	53.03
spla	461	122	82.67	81.20
tseng	305	174	98.13	86.26
Avg			85.43	75.37

Table 3: Percentage for VddL interconnect switches and VddL logic blocks given by dual-Vdd assignment under zero delay-increase (VddH = 1.3v and VddL = 0.8v).

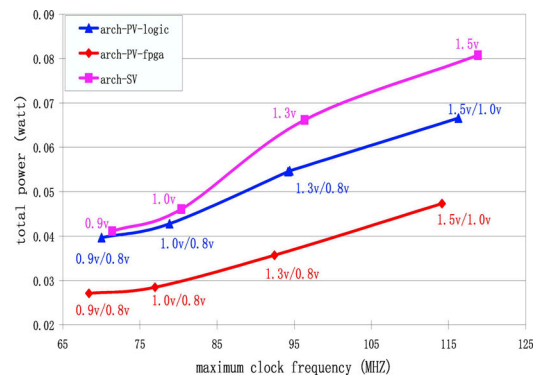


Figure 6: Power-performance curve for the combinational benchmark alu4.

5.2 Power and Performance Comparison

We then present the power and delay evaluation results for a combinational benchmark *alu4* and a sequential benchmark *s38584* in Figure 6 and Figure 7, respectively. The X-axis is the maximum clock frequency decided by the critical path delay. The Y-axis is the total FPGA power consumption. There are three curves in each figure, corresponding to FPGA architectures arch-SV, arch-PV-logic and arch-PV-fpga. For arch-SV, we present the power and

1	2	3	4	5	6
circuit	arch-SV (baseline)		arch-PV-logic [7]	arch-PV-fpga	
	interconnect power (W)	total power (W)	total power saving	total power saving	
				One-bit control	Two-bit control
alu4	0.0657	0.0769	15.83%	23.93%	39.12%
apex4	0.0437	0.0500	7.58%	13.75%	41.31%
bigkey	0.1044	0.1375	24.89%	28.88%	49.12%
clma	0.4918	0.5450	8.82%	20.13%	60.57%
des	0.1688	0.2136	19.07%	31.22%	49.60%
diffeq	0.0292	0.0360	11.01%	14.47%	52.10%
dsip	0.1003	0.1280	24.17%	28.23%	57.03%
elliptic	0.1060	0.1236	11.62%	22.92%	60.98%
ex5p	0.0455	0.0534	8.49%	11.00%	38.45%
frisc	0.1399	0.1603	9.57%	13.98%	64.42%
misex3	0.0601	0.0682	8.12%	15.65%	33.43%
pdc	0.2116	0.2317	8.32%	16.76%	56.37%
s298	0.0600	0.0714	12.87%	25.06%	44.43%
s38417	0.2484	0.2995	17.45%	23.88%	48.84%
s38584	0.2131	0.2590	24.99%	39.52%	62.97%
seq	0.0818	0.0924	8.54%	17.93%	38.76%
spla	0.1519	0.1684	14.64%	21.82%	53.50%
tseng	0.0262	0.0325	21.20%	30.65%	58.91%
avg.	-	-	14.29%	22.21%	50.55%

Table 4: Power savings by fully Vdd-programmable fabric compared to baseline arch-SV at the same maximum frequency in our experiments.

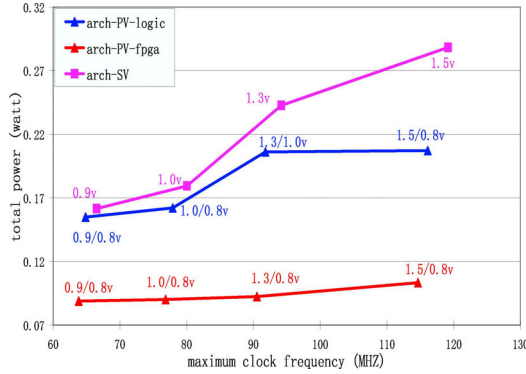


Figure 7: Power-performance for the sequential benchmark s38584.

performance trend as we scale down Vdd for the entire FPGA. The Vdd level is labeled beside each data point. When comparing two adjacent data points of arch-SV, e.g. Vdd=1.5v and Vdd=1.3v, one can see that the maximum clock frequency is degraded significantly when the Vdd is reduced.

For arch-PV-logic and arch-PV-fpga, we try different VddH/VddL combinations and obtain all the data points. After the inferior data points (i.e., those with larger power consumption and lower maximum clock frequency) are pruned, the remaining solutions give the spectrum of power-performance trade-off using different VddH/VddL combinations. We also label the VddH/VddL combination for each data point in both figures. Compared to arch-SV with VddH as the uniform supply voltage, the performance degradation in dual-Vdd architectures (arch-PV-logic or arch-PV-fpga) is negligible. It is because we select VddH for devices on critical paths to maintain performance but reduce power significantly by selecting VddL for devices on non-critical paths.

Compared to arch-PV-logic with Vdd-programmability only for logic blocks, our fully Vdd-programmable architecture arch-PV-fpga reduces total FPGA power by a larger margin (see Figure 6 and Figure 7). This significant improvement is due to the much reduced interconnect power in our new fabric. The power saving by both arch-PV-logic and arch-PV-fpga decreases at lower clock frequency.

It is because that lower clock frequency generally implies lower supply voltage and therefore less timing slack can be utilized for power optimization. In addition, the power-performance curve is relatively flat for s38584 because it is a much larger circuit than alu4 and therefore has more timing slack to achieve a larger power reduction in dual-Vdd optimization.

1	2	3	4	5	6
circuit	arch-SV (baseline)		arch-PV-fpga (two-bit control)		
	interconnect power (W)	interconnect lkg. power (%)	interconnect power saving		
			overall	leakage	dynamic
alu4	0.0657	27.42%	41.52%	78.35%	27.51%
apex4	0.0437	44.02%	47.43%	79.04%	22.63%
bigkey	0.1044	32.73%	52.51%	80.61%	38.85%
clma	0.4918	56.81%	66.97%	82.83%	46.07%
des	0.1688	22.52%	51.54%	82.54%	42.50%
diffeq	0.0292	66.35%	64.46%	77.09%	39.69%
dsip	0.1003	37.57%	57.12%	84.07%	40.89%
elliptic	0.1060	58.09%	69.73%	80.71%	54.46%
ex5p	0.0455	44.26%	47.15%	79.87%	21.24%
frisc	0.1399	72.28%	73.06%	82.65%	48.12%
misex3	0.0601	28.83%	38.07%	78.40%	21.83%
pdc	0.2116	55.76%	61.78%	82.02%	36.15%
s298	0.0600	36.53%	51.92%	77.80%	37.03%
s38417	0.2484	43.06%	56.30%	79.89%	38.46%
s38584	0.2131	40.87%	68.02%	79.61%	59.94%
seq	0.0818	32.63%	43.77%	79.73%	26.36%
spla	0.1519	46.56%	58.65%	81.11%	39.0%
steng	0.0262	50.63%	67.21%	76.26%	57.89%
Avg.		44.27%	56.51%	80.14%	38.81%

Table 5: Interconnect power saving breakdown for arch-PV-fpga with two-bit control.

5.3 Power at Highest Clock Frequency

We present the complete evaluation results for MCNC benchmark set in Table 4. For each circuit, we choose the highest clock frequency achieved by arch-PV-fpga under all VddH/VddL combinations and present the corresponding power saving at that clock frequency. The power consumption for the baseline arch-SV is presented in columns 2-3, and the power saving by arch-PV-logic from [7] is shown in column 4 for the purpose of comparison. For our fully Vdd-programmable architecture arch-PV-fpga, we present the results of both one-bit control and two-bit control implementations. The one-bit control only provides Vdd selection but the

two-bit control has the additional power-gating capability. Compared to arch-SV, our fully Vdd-programmable fabric with one-bit control achieves an average of 22.21% total power reduction. The power reduction ratio³ increases to 50.55% when two-bit control is used. The extra power saving (on average 32% of total interconnect power saving) is obtained via power gating of unused logic blocks and interconnect switches. In contrast, arch-PV-logic in [7] reduces total FPGA power merely by 14.29% because it provides Vdd-programmability only for logic blocks.

We further present the interconnect power saving by our arch-PV-fpga with two-bit control in Table 5. Column 2 is the interconnect power and column 3 is the percentage of interconnect leakage power among total interconnect power. We reduce interconnect leakage power by 80.14% (see column 5) and reduce interconnect dynamic power by 38.81% (see column 6). The interconnect leakage power reduction is mainly obtained by power gating a large number of unused interconnect switches.

6. CONCLUSIONS

We have shown that interconnect power reduction is the key to reduce FPGA total power. We have designed Vdd-programmable interconnect circuits and fabrics to reduce interconnect power. There are three states for our interconnect switches: high Vdd, low Vdd and power-gating. We developed a simple design flow to apply high Vdd to critical paths and low Vdd to non-critical paths and to power gate unused interconnect switches. We performed a highly quantitative study by placing and routing benchmark circuits in 100nm technology. Compared to single-Vdd FPGAs with Vdd level optimized for the same target clock frequency, our new fabric with power-gating capability reduces FPGA interconnect power by 56.51% and total FPGA power by 50.55%. In contrast, the previous configurable dual-Vdd techniques used only for logic blocks [7] reduces total FPGA power merely by 14.29%. Because of the extremely low utilization rate of interconnect switches ($\sim 12\%$ in our profiling using MCNC benchmarks), power gating reduces total FPGA interconnect power by 32%. We use the smallest FPGA array that just fits a given circuit for placement in our experiments. In reality, the FPGA chip size can be significantly larger than that needed by the application circuit. Therefore, power gating unused interconnect switches can reduce even more power in general.

Power supply network to support configurable Vdd or dual-Vdd can introduce extra routing congestion. Our future work includes power delivery design and optimization for our fully Vdd-programmable fabric. We will also study how to reduce the number of SRAM cells used for Vdd programmability.

7. REFERENCES

- [1] K. Poon, A. Yan, and S. Wilton, "A flexible power model for FPGAs," in *Proc. of 12th International conference on Field-Programmable Logic and Applications*, Sep 2002.
- [2] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, Feb 2003.
- [3] T. Tuan and B. Lai, "Leakage power analysis of a 90nm FPGA," in *Proc. IEEE Custom Integrated Circuits Conf.*, 2003.
- [4] J. H. Anderson, F. N. Najm, and T. Tuan, "Active leakage power optimization for FPGAs," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, February 2004.
- [5] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and T. Tuan, "Reducing leakage energy in FPGAs using region-constrained placement," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, February 2004.
- [6] F. Li, Y. Lin, L. He, and J. Cong, "Low-power FPGA using pre-defined dual-vdd/dual-vt fabrics," in *Proc. ACM Intl. Symp. Field-Programmable Gate Arrays*, February 2004.
- [7] F. Li, Y. Lin, and L. He, "FPGA power reduction using configurable dual-vdd," in *Proc. Design Automation Conf.*, pp. 735–740, June 2004.
- [8] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, Feb 1999.
- [9] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *Proc. Intl. Symp. Low Power Electronics and Design*, pp. 155–160, August 1998.
- [10] G. G. Lemieux and S. D. Brown, "A detailed router for allocating wire segments in field-programmable gate arrays," in *Proceedings of the ACM Physical Design Workshop*, April 1993.
- [11] R. Puri, L. Stok, J. Cohn, D. Kung, D. Pan, D. Sylvester, A. Srivastava, and S. Kulkarni, "Pushing ASIC performance in a power envelope," in *Proc. Design Automation Conf.*, pp. 788 – 793, 2003.
- [12] R. W. Brodersen, M. A. Horowitz, D. Markovic, B. Nikolic, and V. Stojanovic, "Methods for ture power minimization," in *Proc. Intl. Conf. Computer-Aided Design*, pp. 35–42, 2002.

³Note that the power reduction ratio in Table 4 is calculated as arithmetic mean for 20 benchmark circuits while the power and power reduction ratio in Figure 1 is calculated as geometric mean.