Low-Power FPGA Using
Pre-defined Dual-Vdd/Dual-Vt Fabrics *

Fei Li[1], Yan Lin[1], Lei He[1] and Jason Cong[2]
[1]Electrical Engineering Department
[2]Computer Science Department
University of California, Los Angeles

ABSTRACT

Traditional FPGAs use uniform supply voltage Vdd and uniform threshold voltage Vt. We propose to use pre-defined dual-Vdd and dual-Vt fabrics to reduce FPGA power. We design FPGA circuits with dual-Vdd/dual-Vt to effectively reduce both dynamic power and leakage power, and define dual-Vdd/dual-Vt FPGA fabrics based on the profiling of benchmark circuits. We further develop CAD algorithms including power-sensitivity based voltage assignment and simulated-annealing based placement to leverage such fabrics. Compared to the conventional fabric using uniform Vdd/Vt at the same target clock frequency, our new fabric using dual Vt achieves 9% to 20% power reduction. However, the pre-defined FPGA fabric using both dual Vdd and dual Vt only achieves on average 2% extra power reduction. It is because that the pre-designed dual-Vdd layout pattern introduces non-negligible performance penalty. Therefore, programmability of supply voltage is needed to achieve significant power saving for dual-Vdd FPGAs. To our best knowledge, it is the first in-depth study on applying both dual-Vdd and dual-Vt to FPGA considering circuits, fabrics and CAD algorithms.

Categories and Subject Descriptors

B.7.1 [**Integrated Circuits**]: Types and Design Styles

General Terms

Design, Algorithms

Keywords

FPGA, low power, power efficient, dual-Vdd, dual-Vt

1. INTRODUCTION

Power has become an increasingly important design constraint. FPGAs are less power efficient than ASICs [1] due to the overhead for programmability. There have been several studies to reduce FPGA power. [1] introduced hierarchical interconnects to reduce interconnect power, but they do not consider deep sub-micron effects such as the increasingly large leakage power. [2] developed a flexible power evaluation framework fpgaEva-LP and performed dynamic and leakage power evaluation for FPGAs with cluster-based logic blocks and island style routing structure [14]. Uniform supply voltage Vdd and threshold voltage Vt are assumed in [2].

Supply voltage reduction is effective to limit the dynamic power consumption because dynamic power reduces quadratically as the supply voltage scales down. Dual-Vdd or multi-Vdd technique has been successfully employed in ASICs [3, 4, 5] and an optimized multi-Vdd system can achieve dynamic power reductions of roughly 40-45% [6, 7]. Dual-Vt or multi-Vt technique [8] has also been applied in ASICs to reduce leakage power.

Following the successful application of multi-Vdd/multi-Vt techniques in ASICs, FPGAs might also benefit from those techniques for power reduction. However, there are some unique challenges to apply multi-Vdd/multi-Vt to FPGAs. First, previous work [2] has shown that leakage power becomes a large portion of the total FPGA power in 100nm technology and below. It is mainly because LUT-based FPGAs use a large number of SRAMs to provide the programmability. Therefore, it is important to design FPGA circuits using dual-Vdd/dual-Vt without increasing the leakage power budget. Second, FPGAs do not have the freedom of using mask patterns to arrange different Vdd/Vt components in a flexible way as ASICs [9, 10]. Multi-Vdd/multi-Vt fabric and layout pattern must be pre-defined in FPGAs for a set of applications. This may limit the power reduction by multi-Vdd/multi-Vt techniques.

In this paper, we perform the first of its type of studies on the dual-Vdd and dual-Vt FPGA fabrics. We design FPGA circuits with dual-Vdd/dual-Vt to effectively reduce dynamic and leakage power. According to the profiling of benchmark circuits, we propose FPGA fabrics employing dual-Vdd/dual-Vt techniques. To leverage the new fabrics, we develop new CAD algorithms including power-sensitivity based voltage assignment and simulated-annealing based placement.

The paper is organized as follows. Section 2 addresses the dual-Vdd/dual-Vt FPGA circuit design as well as the new

FPGA fabrics. Section 3 presents our design flow for dual-Vdd/dual-Vt FPGAs. Section 4 presents the experimental results and Section 5 concludes the paper with discussions of future work.

2. FPGA CIRCUITS AND FABRICS USING DUAL-VDD/DUAL-VT

FPGAs consist of logic blocks and programmable routing channels. We first present detailed circuit design for dynamic and leakage power reduction. We then discuss our pre-defined dual-Vdd/dual-Vt FPGA fabrics. Lookup table (LUT) based FPGAs and the island style routing are assumed in this paper.
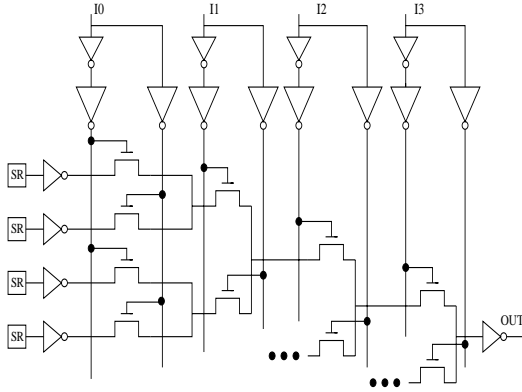
2.1 Lookup Table (LUT) Design

Figure 1: The schematic of a 4-LUT using single Vdd and single Vt (LUT-SVST). SR stands for SRAM cell.

2.1.1 Voltage Scaling for Single-Vdd/Vt LUTs

When dual Vdd is applied at LUT or logic block level, we use a single Vdd inside a LUT. Figure 1 shows part of a 4-LUT circuit using single Vdd and single Vt. It consists of SRAM cells and a MUX tree. The SRAM cells in the LUT can be programmed to implement any four-input logic function. We name LUTs using single Vdd and single Vt as *LUT-SVST*. Vdd scaling of LUT-SVST is effective to reduce dynamic power because dynamic power is quadratically proportional to the supply voltage. However, aggressive Vdd scaling can introduce large delay penalty. It is important to decide appropriate Vt corresponding to the Vdd level for best power-delay trade-off.

Figure 2 shows the delay increase of a 4-LUT in 100nm technology during Vdd scaling. Three different Vdd scaling schemes are presented in the figure. *constant-Vt* scheme scales down Vdd without changing the threshold voltage Vt. It is clear that the LUT delay at the lowest Vdd (0.8v) in our study is 3X larger than that at the highest Vdd (1.3v). To compensate for the large delay penalty at low Vdd, we can scale down the threshold voltage Vt accordingly as the Vdd scales down. The scaling scheme *fixed-Vdd/Vt-ratio* keeps a constant Vdd/Vt ratio. Figure 2 shows that it increases the LUT delay by only 40% at the lowest Vdd level. Although *fixed-Vdd/Vt-ratio* is promising to alleviate delay penalty compared to *constant-Vt*, leakage power increases greatly in this scaling scheme. This is because the leakage current
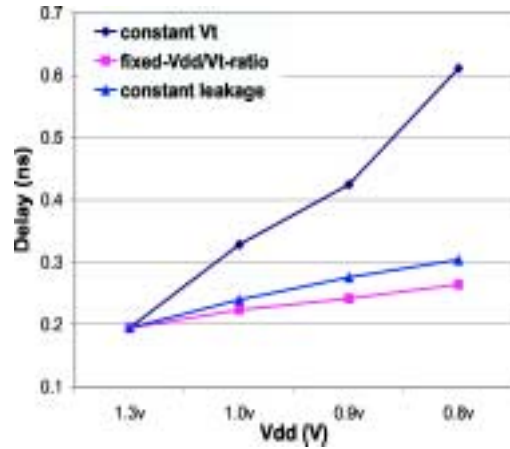

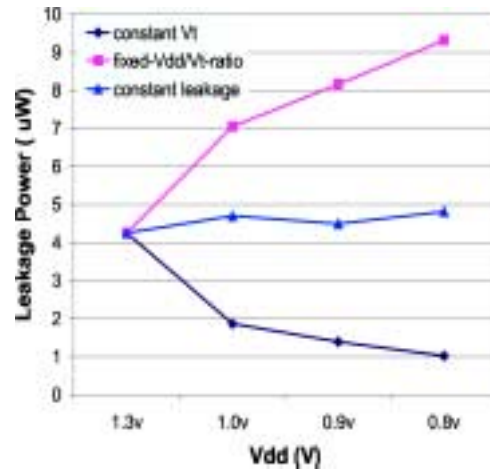**Figure 2: Delay versus different Vdd scaling schemes for a 4-LUT.**

Figure 3: Leakage power (at $100°C$) versus different Vdd scaling schemes for a 4-LUT.

increases exponentially when Vt reduces [12]. As shown in Figure 3, the leakage power for a 4-LUT almost doubles at the lowest Vdd (0.8v) in *fixed-Vdd/Vt-ratio* scaling even though the Vdd level is reduced. Since leakage power has already been a large portion of total FPGA power in nanometer technology [2], FPGA designs cannot afford the increasing leakage power by the *fixed-Vdd/Vt-ratio* scaling scheme.

Based on the above two Vdd scaling schemes, we propose the *constant-leakage* Vdd scaling scheme. For each Vdd level, we adjust the threshold voltage to maintain an almost constant leakage power across all the Vdd levels (see Figure 3). In Figure 2, we show that *constant-leakage* scaling scheme also limits the delay penalty almost as effectively as the *fixed-Vdd/Vt-ratio* scaling scheme. Therefore, *constant-leakage* scaling achieves much better power-delay trade-off at circuit level compared to the other two scaling schemes. Table 1 summarizes our Vdd/Vt combination used in the *constant-leakage* scaling scheme in the ITRS 100nm technology [11].

Vdd (V)	NMOS-Vt (V)	PMOS-Vt (V)
1.3	0.2607	-0.3030
1.0	0.2205	-0.2530
0.9	0.2105	-0.2389
0.8	0.1884	-0.2254

Table 1: Vdd/Vt combinations for *constant-leakage* Vdd scaling scheme in the ITRS 100nm technology.

2.1.2 Low-leakage SRAM and Dual-Vt LUTs

Although we only use a single Vdd inside a LUT, we still can apply dual Vt to a LUT. We design low-leakage LUTs with single Vdd and dual Vt (named as *LUT-SVDT*). Figure 4 shows the schematic of a LUT-SVDT. The entire LUT is partitioned into two different regions. All the SRAM cells belong to region I and provide the configuration signal for the LUT. The rest part, MUX-tree and input buffers, becomes region II. Note that the two regions are DC disconnected due to the inverters at the output of the SRAM cells. The content of the SRAM cells does not change after the LUT is configured and the SRAM cells always stay in the read status. Therefore, we can increase the threshold voltage of region I to reduce leakage power without introducing runtime delay penalty. We determine Vdd and Vt in a LUT-SVDT as follows. For region II, we decide the Vdd/Vt combination by *constant-leakage* Vdd scaling scheme. For region I, we use the same Vdd as region II but increase Vt. Figure 5 compares the delay and leakage power between LUT-SVST and LUT-SVDT. LUT-SVDT obtained an average 2.4X LUT leakage reduction compared to LUT-SVST at different Vdd levels. The delay of LUT-SVDT is almost same as LUT-SVST. Considering the increasingly large portion of leakage power in FPGAs under 100nm technology, LUT-SVDT is an effective design technique to reduce leakage for logic blocks.
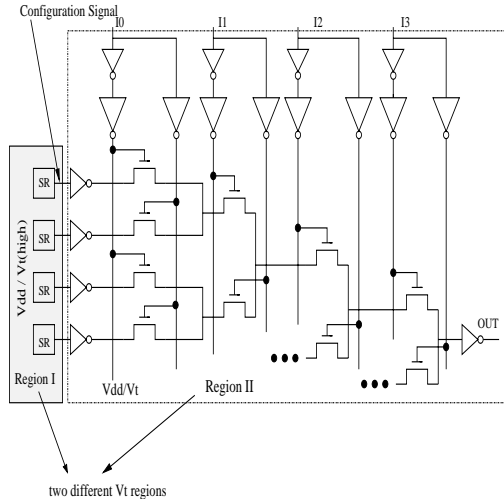

Figure 4: The schematic of a 4-LUT using single Vdd and dual Vt (LUT-SVDT). SR stands for SRAM cell.

The high-Vt low-leakage SRAM cells can be used for programmability of both interconnects and logic blocks. Ideally, we can increase Vt as high as possible to achieve maximal leakage reduction without delay penalty. However, an ex-
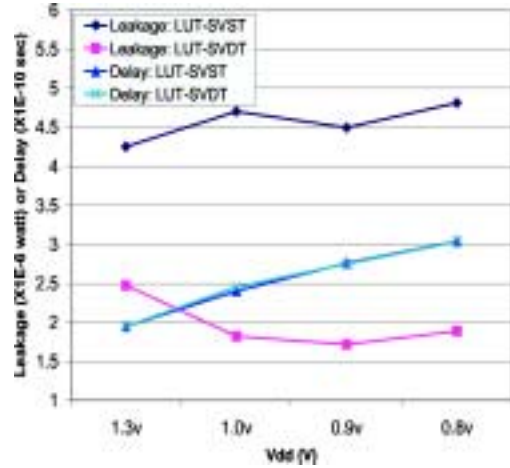

Figure 5: Delay and power comparison between LUT-SVST and LUT-SVDT in the ITRS 100nm technology. There is virtually no delay difference between LUT-SVST and LUT-SVDT.

tremely high Vt increases the SRAM write access time and slows down the FPGA configuration speed. We decide to increase the Vt of SRAM cells for 15X SRAM leakage reduction. It increases the configuration time only by 13%.

2.2 Level Converter Design

For a dual-Vdd FPGA fabric, the interface between a VddL device and a VddH device must be designed carefully to avoid the excessive leakage power. If a VddL device drives a VddH device and the VddL device output is logic '1', both PMOS and NMOS transistors in the VddH device will be at least partially "on", dissipating unacceptable amount of leakage power due to short circuit current. A level converter should be inserted to block the short circuit current. The level converter converts VddL signal swing to VddH signal swing [1]. Different level converter circuits have been used in dual-Vdd ASIC designs. DCVS level converters were proposed in [6, 3] and level converters with data latch function were proposed in [7, 13]. These level converters use both VddH and VddL as its supply voltages and create extra constraints for power/ground routing. Recently, new asynchronous level converters with single supply voltage have been proposed in [10] and is used in this paper. As shown in Figure 6, when the input signal is logic '1', the threshold voltage drop across NMOS transistor 'n1' can provide a virtual low supply voltage to the first-stage inverter (p2,n2) so that p2 and n2 will not be partially "on". When the input signal is logic '0', the feedback path from node 'OUT' to PMOS transistor 'p1' pulls up the virtual supply voltage to VddH and inverter (p2,n2) generates a VddH signal to the second inverter so that no DC short circuit current exists. For a particular VddH/VddL combination, we decide the transistor size in the level converter as follows. We start from a level converter with minimum transistor sizes. We size up the transistors to limit the level converter delay within 30% of a single LUT delay or 7% of a logic cluster delay. For transistor sizes that meet the delay bound, we

---

[1]Note that a VddH device can drive a VddL device without generating excessive leakage power. No level convert is needed in this case.
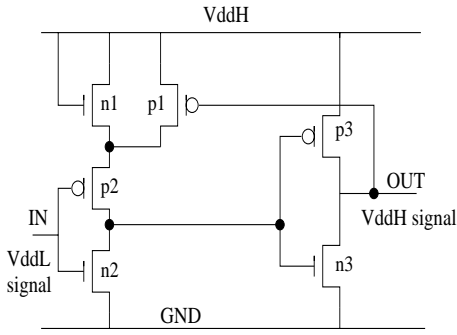
Figure 6: A level converter circuit with single supply voltage.

choose the sizing with the lowest power consumption. Table 2 shows the delay and leakage power of the sized level converters used in this paper. Note that the leakage power increases as the voltage difference between VddH and VddL increases. This is because the threshold voltage drop cannot provide a proper virtual low-supply as the gap between VddH and VddL is large. Therefore, the VddH/VddL ratio cannot be too large considering the leakage overhead of level converters.

| VddH/VddL	delay (ns)	energy per switch (fJ)	leakage power (uW)
1.3v/1.0v	0.0814	7.40	0.0104
1.3v/0.9v	0.0801	8.05	0.0139
1.3v/0.8v	0.0845	9.73	0.0240

Table 2: SPICE simulation results for single supply level converter in the ITRS 100nm technology.
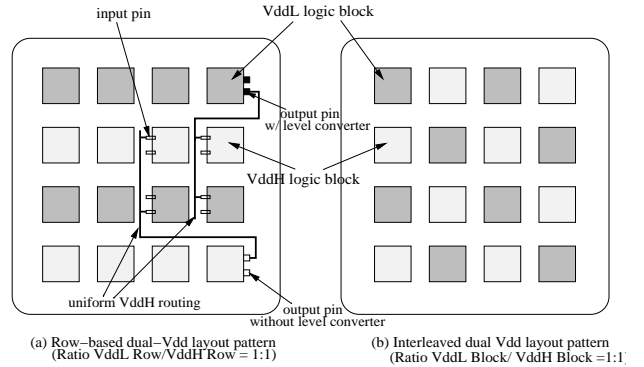
2.3 FPGA Fabrics

Traditional FPGA fabrics largely use uniform Vdd and Vt, but only provide limited power performance trade-off. To further achieve power efficiency, we design dual-Vdd/dual-Vt fabrics based on traditional uniform FPGA fabrics with cluster-based logic blocks and island-style routing structures [14]. Dual Vt is applied to configuration SRAM cells in both logic blocks and programmable interconnects. We limit ourself to only apply dual Vdd to logic blocks. The advantage of dual-Vdd routing fabric will be explored in the future.

Figure 7 shows a generic FPGA fabric with cluster-based logic blocks and island style routing structures. The Basic Logic Element (BLE) consists of one Lookup Table (LUT) and one flip-flop. A set of fully connected BLEs become a logic block or a cluster. The number of BLEs is the logic block size. The logic blocks are embedded into the programmable routing resources. When all the logic blocks and routing resources use the same Vdd and Vt, it is the traditional FPGA fabric with uniform Vdd and Vt and is called Single-Vdd Single-Vt fabric (*arch-SVST*). The Vdd/Vt ratio is determined by the *constant leakage* Vdd scaling discussed in Section 2.1 and the lookup tables in the logic blocks are *LUT-SVST*. [14] defines a complete list of architectural parameters for this FPGA fabric. Based on arch-SVST, we design another FPGA fabric *arch-SVDT*, which uses LUT-SVDT and low-leakage SRAM cells presented in Section 2.1.
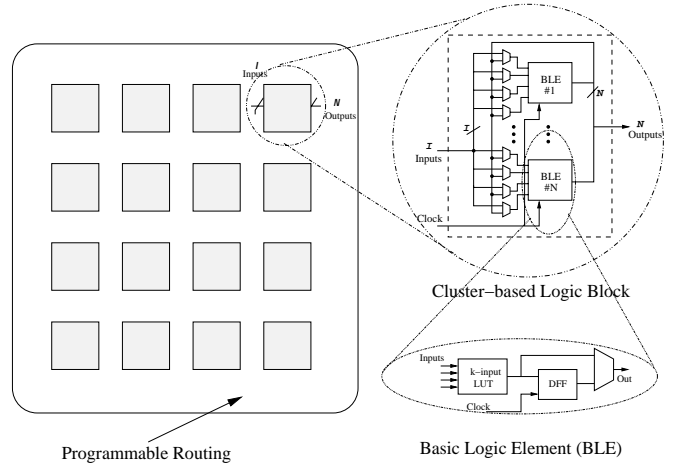
Figure 7: A FPGA with cluster-based logic blocks and island style routing structures.

(a) Row–based dual–Vdd layout pattern (Ratio VddL Row/VddH Row = 1:1)

(b) Interleaved dual Vdd layout pattern (Ratio VddL Block/ VddH Block =1:1)

Figure 8: Pre-designed dual-Vdd layout patterns for dual-Vdd logic block fabric.

I.e., the only difference between arch-SVST and arch-SVDT is that the latter one uses low-leakage SRAM cells for both LUTs and interconnects.

The above two fabrics both use a single Vdd in the entire FPGA chip. To design a dual-Vdd FPGA fabric, we need to determine: (i) What is the granularity to apply dual Vdd? (ii) What is the layout pattern formed by the physical locations of devices using dual-Vdd? We name the new fabric with dual Vdd and dual Vt *arch-DVDT*. It uses low-leakage SRAM cells for all LUTs and interconnects, and employs one single Vdd inside one logic block. But logic blocks across the FPGA chip can have different supply voltages. The physical locations of these logic blocks define a dual-Vdd layout pattern. Figure 8 shows two possible layout patterns. One is the row-based pattern with a ratio of VddL-row/VddH-row as 1:1. Another is the interleaved layout pattern with a ratio of VddL-block/VddH-block as 1:1. In general, the ratio of VddL-row/VddH-row or of VddL-block/VddH-block is an architectural parameter and is determined experimentally in this paper. Note that the routing resources in arch-DVDT use uniform VddH. Figure 8 also shows example routing paths connecting logic blocks with different supply voltages. The output signals from a VddL logic block must go through level converters before entering the routing channels. If the

Fabric name	Vdd level across chip	Vt level across chip	SRAM cell	LUT type	Layout Pattern (in terms of Vdd)
arch-SVST	single	single	normal-Vt	SVST	uniform
arch-SVDT	single	dual	high-Vt	SVDT	uniform
arch-DVDT	dual	dual	high-Vt	SVDT	row-based or interleaved

Table 3: Summary of the three FPGA fabrics studied in this paper.

VddL logic block size is N, i.e., it has N output pins, we need N level converters at output pins. On the other hand, VddH logic blocks do not need any level converters. The signal in the uniform VddH routing finally reaches another logic block, which can be either VddH or VddL. In either case, no level converters are needed at the input pins of a logic block. We summarize the three FPGA fabrics in Table 3.

3. DESIGN FLOW FOR DUAL-VDD/DUAL-VT FPGAS

CAD algorithms need to be developed to leverage the proposed FPGA fabrics with dual Vdd and dual Vt. Figure 9 presents our low-power design flow. The input data is a single-Vdd gate-level netlist and it is optimized by SIS [15] and mapped to LUTs by RASP [16]. We then start the physical design as shown in the large dotted box in Figure 9. After the physical design, we generate the basic circuit netlist (BC-netlist) extended from the one in [2]. The BC-netlist is annotated with capacitance, resistance as well as supply voltage level if dual Vdd is applied. We perform power estimation and timing analysis on the BC-netlists to obtain the power and performance. An enhanced version of *fpgaEva-LP* [2] is developed to handle dual-Vdd/dual-Vt FPGA power estimation.

Our physical design flow has two parallel design paths. One is the traditional FPGA physical design flow labeled as (a) in Figure 9. Timing-driven packing, placement and routing are carried out for a single-Vdd netlist by using VPR [14]. Note that this design path can also handle our new fabric arch-SVDT presented in Table 3, as only difference between arch-SVST and arch-SVDT is that arch-SVDT uses low-leakage SRAM cells with higher Vt. At the logic block level, it can still be viewed as a uniform fabric and traditional design flow can be readily applied. Another design path is proposed for dual-Vdd FPGA fabric such as *arch-DVDT*. After the generation of single-Vdd BC-netlist, two extra steps are needed: dual-Vdd assignment and timing driven layout for dual-Vdd fabric. Below, we discuss the two steps.

3.1 Dual-Vdd Assignment

The dual-Vdd assignment determines the Vdd level for each logic block in the mapped netlist. It makes use of the surplus timing slack in a circuit and perform power optimization using dual Vdd. Sensitivity-based optimization algorithms have been used in ASIC circuit tuning either for delay optimization [17] or for power-delay trade-off [18]. We use a similar sensitivity-based algorithm for dual-Vdd assignment. First, we define the *power sensitivity* as follows,

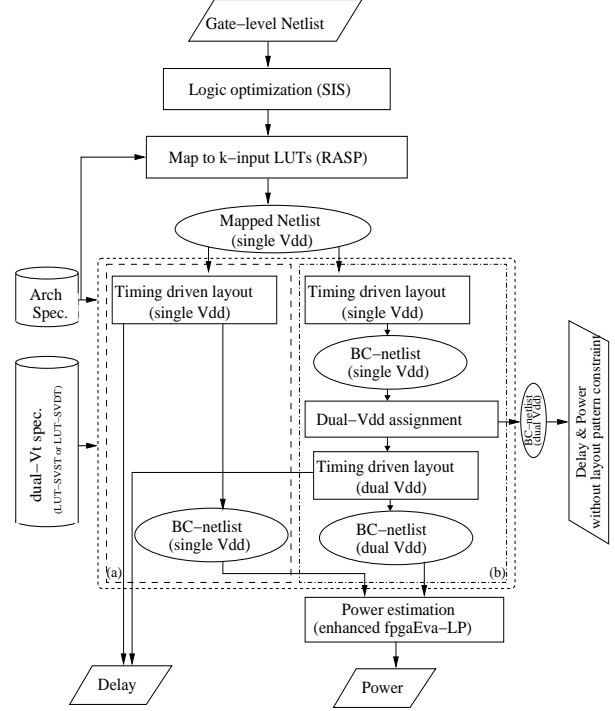DEFINITION 1 (POWER SENSITIVITY S_x). *For a given de-*

Figure 9: Design flow for dual-Vdd/dual-Vt FPGAs.

sign variable x, the power sensitivity is calculated as

$$
\begin{aligned}
S_x &= \frac{\Delta P}{\Delta x} \\
&= \frac{\Delta P_{sw}}{\Delta x} + \frac{\Delta P_{lkg}}{\Delta x}
\end{aligned}
$$

where P_{sw} is the switching power and P_{lkg} is the leakage power.

In our dual-Vdd assignment problem, the design variable x becomes supply voltage Vdd. To calculate the power sensitivity, we need the relationship between power and supply voltage. We use the FPGA power model proposed in [2]. The switching power P_{sw} of a primitive node i in the BC-netlist is calculated as follows,

$$P_{sw}(i) = 0.5f \cdot \hat{E}_i \cdot C_i \cdot V_{dd}^2 \tag{1}$$

where f is the clock frequency, $\hat{E}_i$ is the effective transition density considering glitches and C_i is the load capacitance. The leakage power P_{lkg} of node i is calculated as follows,

$$P_{lkg}(i) = I_{lkg}(V_{dd}) \cdot V_{dd} \tag{2}$$

where I_{lkg} is the leakage current at supply voltage V_{dd}. Because we use the constant-leakage Vdd scaling, $\Delta P_{lkg}/\Delta x$ is

zero. The power sensitivity of a logic block B can be calculated as the sum of sensitivities for all the nodes inside this logic block, i.e.

$$S_x(B) = \sum_{node\ i \in B} S_x(i) \qquad (3)$$

We present our dual-Vdd assignment algorithm in Figure 10. It is a greedy algorithm with an iteration loop. Given the single-Vdd BC-netlist, we analyze the timing and obtain the circuit path with the largest timing slack. Power sensitivity is calculated for logic blocks on this path but not on the critical path. We select the logic block with the largest power sensitivity and assign low Vdd to it, and update the timing information. If the new critical path delay exceeds the user-specified delay increase bound, we reverse the low-Vdd assignment. Otherwise, we keep this assignment and go to next iteration. In either case, the logic block selected in this iteration will not be re-visited in other iterations. Right after the dual-Vdd assignment, we can estimate the power and delay for the dual-Vdd BC-netlist as shown in Figure 9. However, this dual-Vdd BC-netlist does not consider the layout constraint imposed by the pre-designed dual-Vdd pattern. It assumes the flexibility to assign low-Vdd to a logic block at arbitrary physical location. We call it *ideal case* for fabric arch-DVDT. To obtain *real case* power and delay considering the layout pattern constraint, we use this dual-Vdd netlist as an input and perform dual-Vdd placement and routing.

Sensitivity-based dual-Vdd assignment algorithm:
input: single-Vdd BC-netlist N
output: dual-Vdd BC-netlist N'
      (with original Vdd and another low Vdd)
constraint:
$\frac{crit\_path\_delay(N') - crit\_path\_delay(N)}{crit\_path\_delay(N)} < delay\_increase\_bound$

Let partially assigned BC-netlist N_p be input netlist N;
While(N_p has logic blocks not tried)
begin
      Find path p with largest timing slack in N_p;
      Get logic blocks on path p but not on critical path;
      Calculate power-sensitivity for those logic blocks;
      Select logic block B with largest sensitivity;
      Assign low Vdd to B and update timing information;
      If(delay constraint not met)
      begin
            Reverse the low-Vdd assignment;
      end
      mark logic block B as 'tried';
end
Let the output netlist N' be N_p

Figure 10: Sensitivity-based dual-Vdd assignment algorithm.

3.2 Placement and Routing for Dual-Vdd FPGA Fabric

We present our placement and routing flow in Figure 11. The input data is the dual-Vdd BC-netlist generated by dual-Vdd assignment. Our dual-Vdd placement considers the layout constraint in arch-DVDT. The traditional global and detail routing algorithms [14] are applied because the

routing fabric in arch-DVDT uses uniform Vdd. After placement and routing, we obtain a new dual-Vdd BC-netlist that satisfies the layout pattern constraint.
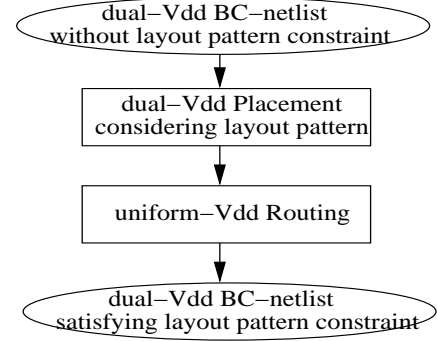
Figure 11: Placement and routing for dual-Vdd fabric arch-DVDT.

Our dual-Vdd placement is based on the simulated annealing algorithm implemented in VPR [14]. VPR placement tool models an FPGA as a set of legal slots or discrete locations, at which logic blocks or I/O pads can be placed. A linear congestion cost function is used in VPR placement, which is shown as follows,

$$Cost_{lin-cgst} = \sum_{i=1}^{N_{nets}} q(i) \left[\frac{bb_x(i)}{C_{av,x}(i)^\beta} + \frac{bb_y(i)}{C_{av,y}(i)^\beta} \right] \qquad (4)$$

The summation is done over the number of nets N_{nets} in the circuit. For each net i, $bb_x(i)$ and $bb_y(i)$ represents the horizontal and vertical spans of its bounding box, respectively. The $q(i)$ compensating factor is due to the fact that the bounding box wire length model underestimate the wiring required to connect nets with more than three terminal. Its value depends on the number of terminals in net i. $C_{av,x(i)}$ and $C_{av,y(i)}$ are the average channel capacities in x and y directions, respectively, over the bounding box of net i. When the channel capacities are different across the FPGA chip, the cost function penalizes placements which require more routing in the narrower channels and hence reduce the routing congestion.

We develop new cost function for the dual-Vdd placement. The placement cost difference caused by moving logic block j to a new location is defined as follows,

$$
\begin{aligned}
\Delta Cost \ =\ & \Delta Cost_{lin-cgst} + \alpha \cdot \Delta matched(j) \\
& + \gamma \cdot (1 - matched(j)) \qquad (5)
\end{aligned}
$$

$matched(j)$ is a boolean function related to the assigned Vdd level for block j and its the physical location. If the Vdd level assigned to block j does not match the Vdd level at its physical location determined by the dual-Vdd layout pattern, $matched(j)$ returns value '0'. Otherwise, it returns '1'. $\Delta matched(j)$ is the difference between $matched(j)$ in the previous placement and $matched(j)$ in the current placement. It penalizes a 'move' that brings a logic block from a Vdd-matched location to a Vdd-unmatched location. Term $1 - matched(j)$ penalizes a 'move' that brings a logic block from a Vdd-unmatched location to another Vdd-unmatched location. α and γ are weights set to the similar order of magnitude as typical linear congestion cost value. They are tuned to have a better trade-off between power and delay.

The new cost function is integrated into VPR placement tool and the adaptive annealing schedule same as that in [14] is used to perform the placement.

4. EXPERIMENTAL RESULTS

We have carried out power evaluation for the following three FPGA fabrics. The first is the single-Vdd single-Vt fabric *arch-SVST*. This is the traditional uniform FPGA fabric. We study its low-power application in the context of supply voltage scaling, i.e., the system-level trend of power and performance as we scale down the supply voltage. The circuit design for arch-SVST uses the constant-leakage Vdd scaling scheme proposed in Section 2.1. The second fabric we studied is *arch-SVDT*. It uses a single supply voltage for the entire FPGA, but explores dual-Vt technique for lookup tables (LUT) and programmable FPGA interconnects. The SRAM cells in a LUT are designed with higher threshold voltage than the rest logic circuits in the LUT. Although the LUT design involves dual threshold voltage, arch-SVDT can still be viewed as a uniform fabric at logic block level. Further, for all the configuration SRAM cells for interconnects in arch-SVDT, we use our low-leakage SRAM with high Vt. All the interconnect buffers and routing switches use normal Vt to maintain the performance. The last fabric is *arch-DVDT*. It uses the same dual-Vt technique as arch-SVDT, but further applies dual Vdd at the logic block level. There are two types of logic blocks in arch-DVDT: VddH logic block and VddL logic block (VddH and VddL must be different in arch-DVDT). The physical locations of these logic blocks define the dual-Vdd layout pattern. Row-based and interleaved layout patterns are studied in this paper. The ratio between VddL row (cell) number and VddH row (cell) number is an architectural parameter. For all the three fabrics, we use LUT size 4 and logic block size 10 in our experiments.

Before we present the complete experimental results, we need to determine the ratio between VddL row (cell) number and VddH row (cell) number for arch-DVDT. We decide the ratio according to the dual-Vdd assignment. Table 4 shows the percentage of logic blocks assigned with VddL for 20 benchmark circuits. The assignment constraint is set to zero delay-increase compared to corresponding *arch-SVDT* with a uniform supply voltage VddH. VddH and VddL are set to 1.3v and 0.8v, respectively. On average, we can assign 75% of logic blocks with VddL. It clearly shows that circuits implemented on uniform FPGA fabric have a large amount of surplus timing slack, which can be utilized for power reduction. According to the dual-Vdd assignment results, the ratio between VddL row (cell) number and VddH row (cell) number should be set to 3:1. However, the layout pattern constraint for placement increases critical path delay and we usually cannot achieve the ideal ratio. We set the ratio to 2:1 in our experiments for both row-based and interleaved layout patterns.

We carry out experiments on 20 MCNC benchmarks for the three FPGA fabrics. Both row-based and interleaved layout patterns in Figure 8 have been tried for arch-DVDT. However, our experimental results show no significant power and performance difference between these two layout patterns. Considering that row-based layout pattern is easier to route the power/ground network, we only present the experimental results of row-based layout pattern for arch-DVDT. Figure 12 and Figure 13 show the experimental results for

circuit	# of logic blocks	# of I/O blocks	% of VddL logic blocks
alu4	162	22	74.07
apex2	213	41	46.01
apex4	134	28	60.45
bigkey	294	426	89.12
clma	1358	144	80.93
des	218	501	74.31
diffeq	195	103	83.59
dsip	588	426	54.32
elliptic	666	245	90.74
ex1010	513	20	75.66
ex5p	194	71	60.98
frisc	731	136	95.13
misex3	181	28	57.52
pdc	624	56	69.54
s298	266	10	82.81
s38417	982	135	88.67
s38584	1046	342	96.73
seq	274	76	53.03
spla	461	122	79.70
tseng	305	174	86.26
Avg			74.98

Table 4: Percentage of VddL logic blocks given by dual-Vdd with zero delay-increase and no layout restrictions. (VddH = 1.3v and VddL = 0.8v)

a combinational circuit *alu4* and a sequential circuit *bigkey*, respectively. The X-axis is the clock frequency calculated as the reciprocal of critical path delay. The Y-axis is the total power consumption. There are four curves in each figure. The first two curves represent the two single-Vdd FPGA fabrics. We show the power and performance trend as we scale down the supply voltage. The supply voltages are marked in both figures. Because we use constant-leakage scaling, the leakage power is kept almost the same during voltage scaling for *arch-SVST*. At lower supply voltage, the proportion of leakage power for arch-SVST increases because the dynamic power is reduced quadratically as voltage scales down. Because *arch-SVDT* mainly reduces leakage power via dual Vt, the power saving obtained by arch-SVDT increases as the leakage portion increases. For the Vdd range in our experiments, arch-SVDT achieves power saving from 9% to 18% for alu4 and from 12% to 26% for bigkey. Note that our arch-SVDT has virtually no performance loss compared to uniform fabric arch-SVST.

The last two curves in both figures are for dual-Vdd fabric *arch-DVDT*. One is the *ideal case* result which is the power and performance for dual-Vdd BC-netlist without considering pre-defined layout pattern. We obtain the ideal case curve as follows. A target clock frequency is set as the delay constraint in dual-Vdd assignment. After voltage assignment considering several different VddH/VddL combinations, we prune the *inferior* data points (i.e., those with larger power consumption and smaller clock frequency). The other curve is the *real case* result obtained after our entire design flow, which considers the dual-Vdd layout pattern. Similarly, we try several different VddH/VddL combinations and prune inferior solutions. The VddH/VddL combinations are also labeled in both figures. Figure 12 shows that arch-DVDT can further obtain more power reduction at the higher clock frequency by applying dual-Vdd to arch-SVDT. It illustrates the benefit of employing dual-Vdd techniques in FPGAs. However, the dual-Vdd technique does
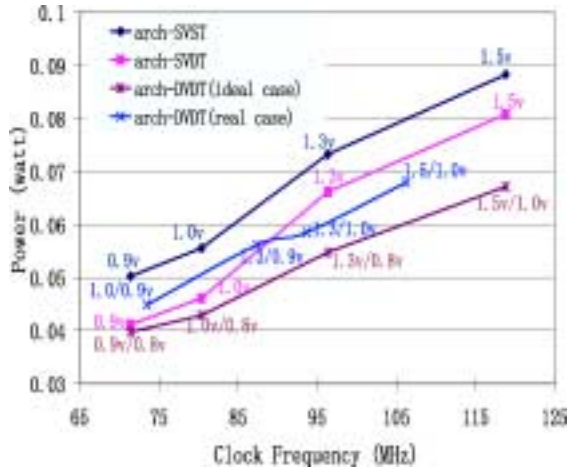
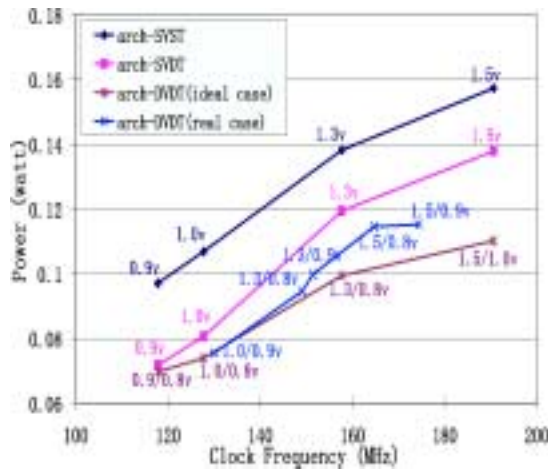**Figure 12: Power versus delay for *alu4*.**

Figure 13: Power versus delay for *bigkey*.

have some extra overhead. Level converters inserted between VddL block and VddH block consume extra power. The pre-determined layout pattern of dual-Vdd fabric imposes placement constraint, which can increase the delay and further reduce the power saving calculated at a target frequency. As shown in the lower frequency region of Figure 12, the overhead of dual-Vdd fabric exceeds the benefit it can bring and arch-DVDT achieves less power savings compared to arch-SVDT. Figure 13 presents the similar comparison for benchmark *bigkey* and arch-DVDT obtains consistently better power savings than arch-SVDT. For both benchmarks, we do see a gap between real case curve and ideal case curve for arch-DVDT. It implies that not all the potential power reduction via introducing dual Vdd is achieved by our current fabric and CAD algorithms.

We present the experimental results for 10 combinational benchmarks and 10 sequential benchmarks in Table 5. For the simplicity of presentation, we choose the maximum clock frequency achieved by arch-DVDT for each individual benchmark, and present the corresponding power savings at the maximum clock frequency. Compared to fabric arch-SVST using uniform Vdd and Vt, the dual Vt fabric arch-SVDT obtains 11.6% and 14.6% power saving on average for com-

binational and sequential circuits, respectively. The dual-Vdd fabric arch-DVDT achieves 13.6% and 14.1% power saving on average for combinational and sequential circuits, respectively. For individual circuits, the dual-Vdd fabric arch-DVDT can achieve up to 10% more power savings compared with arch-SVDT (see circuits *spla* and *bigkey*). However, the dual-Vdd fabric is not always effective to achieve power reduction as the pre-defined layout pattern introduces no-negligible delay penalty (detail discussion in Section 5). For several benchmarks, the overhead of applying dual-Vdd and the associated layout constraint in arch-DVDT offset its benefit, and power savings are smaller compared to arch-SVDT which uses dual Vt but single Vdd. In order to fully explore the potential power reduction offered by introducing dual Vdd, we discuss the possible ways to alleviate the layout constraint in Section 5.

Results for Combinational Circuits		
	arch-SVST (baseline)	arch-SVDT	arch-DVDT
circuit	Power (watt)	power saving	power saving
alu4	0.0798	8.5%	14.9%
apex2	0.108	9.3%	7.7%
apex4	0.0536	12.3%	16.8%
des	0.234	10.7%	13.6%
ex1010	0.179	17.3%	12.3%
ex5p	0.059	11.6%	16.1%
misex3	0.0753	9.4%	13.1%
pdc	0.256	14.7%	15.0%
seq	0.0927	9.4%	4.3%
spla	0.180	12.4%	22.2%
avg.		11.6%	13.6%
Results for Sequential Circuits		
	arch-SVST (baseline)	arch-SVDT	arch-DVDT
circuit	Power (watt)	power saving	power saving
bigkey	0.148	12.3%	22.1%
clma	0.632	14.8%	18.7%
diffeq	0.0391	19.7%	13.8%
dsip	0.134	14.5%	22.2%
elliptic	0.140	16.3%	12.0%
frisc	0.190	19.2%	18.0%
s298	0.0736	13.4%	9.3%
s38417	0.307	11.7%	6.9%
s38584	0.261	10.2%	5.6%
tseng	0.0351	14.0%	11.8%
avg.		14.6%	14.1%

Table 5: Power saving obtained by pre-defined dual-Vdd/dual-Vt fabrics at the same target clock frequency. The clock frequency is chosen as the maximum clock frequency achieved by arch-DVDT.

5. CONCLUSIONS AND DISCUSSIONS

We have developed FPGA circuits, fabrics and CAD algorithms for employing dual Vdd and dual Vt to reduce dynamic and leakage power in FPGAs. We proposed constant-leakage Vdd scaling to effectively reduce dynamic power consumption without increasing FPGA leakage power. We have also designed low-leakage SRAM cells and dual-Vt LUTs without runtime delay penalty. We have then developed a leakage-efficient dual-Vt fabric, using low-leakage SRAM cells for programmable interconnects and dual-Vt LUTs for logic blocks. Furthermore, we have designed a dual-Vdd FPGA fabric containing logic clusters of different Vdd levels. Finally, to leverage the new fabrics, we have developed CAD algorithms including sensitivity based Vdd assignment and simulated annealing based placement considering pre-defined dual-Vdd layout pattern.

Compared to the conventional FPGA fabric using uniform

Vdd and Vt, our new fabric using dual-Vt obtains 11.6% and 14.6% total power reduction on average for combinational and sequential circuits, respectively. Our dual-Vdd and dual-Vt fabric obtains 13.6% and 14.1% total power reduction on average for combinational and sequential circuits, respectively. For individual benchmark, our new dual-Vdd dual-Vt fabric can achieve up to 22% power reduction compared to the conventional FPGA fabric. Note that all the power reductions are obtained by comparing the power consumption of different fabrics at a same clock frequency.

Our experiments have shown that there is a significant power gap between the pre-defined dual-Vdd layout pattern and the ideal dual-Vdd case without considering layout constraint. Such gap is due to the fact that the pre-defined dual-Vdd pattern leads to an extra constraint of matching Vdd level during placement of clusters, which introduces non-negligible delay penalty, in turn, non-negligible power penalty to achieve the target clock frequency. One possible solution to reduce the delay/power penalty is the programmability of supply voltage. Figure 14 presents a schematic for a Vdd-programmable logic cluster. Two transistor switches can be configured to obtain the desired Vdd level for any logic cluster. This removes the contraint to match Vdd level during cluster placement. Certainly, the extra programmability added to the existing FPGA fabric is associated with additional circuit-level power and delay overhead. For example, a level converter is required for every Vdd-programmable logic block. A quantitative study needs to be carried out to justify the programmable Vdd. The preliminary study in [19] has shown that Vdd-programmability can achieve power saving very close to the ideal case power saving.
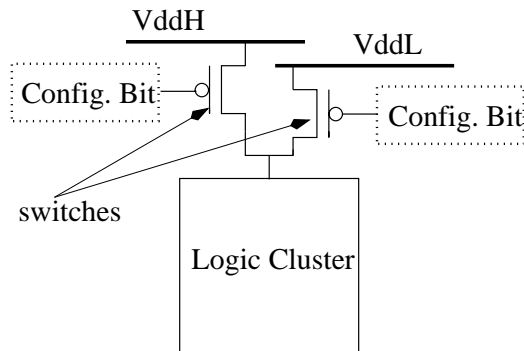
Figure 14: A schematic for a Vdd-programmable logic cluster.

Interconnect power is a large portion of total power in FPGAs. We have applied dual-Vt to FPGA interconnects to reduce leakage power without introducing runtime delay penalty. In the future, we will study how to apply dual-Vdd to FPGA interconnects for dynamic power reduction and how to leverage the dual-Vdd routing fabrics.

6. REFERENCES

[1] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *ISLPED*, 1998.

[2] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient fpgas," in *ISFPGA*, 2003.

[3] K. Usami and M. Horowitz, "Clustered voltage scaling techniques for low-power design," in *ISLPED*, 1995.

[4] C. Chen, A. Srivastava, and M. Sarrafzadeh, "On gate level power optimization using dual-supply voltages," *IEEE Trans. on VLSI Systems*, 2001.

[5] M. Hamada, Y. Ootaguro, and T. Kuroda, "Utilizing surplus timing for power reduction," in *Proc. CICC*, 2001.

[6] K. Usami and et al, "Automated low-power technique exploiting multiple supply voltages applied to a media processor," *IEEE Journal of Solid-State Circuits*, 1998.

[7] M. Hamada and et al, "A top-down low power design technique using clustered voltage scaling with variable supply-voltage scheme," in *CICC*, 1998.

[8] J. T. Kao and A. P. Chandrakasan, "Dual-Threshold Voltage Techniques for Low-Power Digital Circuits," in *IEEE Journal of Solid-state circuits*, 2000.

[9] D. E. Lackey and et al., "Managing power and performance for system-on-chip designs using voltage islands," in *ICCAD*, 2002.

[10] R. Puri and et al, "Pushing ASIC performance in a power envelope," in *DAC*, 2003.

[11] International Technology Roadmap for Semiconductors, 2003 Edition, http://public.itrs.net/Files/2003ITRS/Home2003.htm

[12] S.Sze., *Physics of Semiconductor Devices.* John Wiley and Sons, 1981.

[13] F. Ishihara, F. Sheikh, and B. Nikolic, "Level conversion for dual-supply systems," in *ISLPED*, 2003.

[14] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for Deep-Submicron FPGAs.* Kluwer Academic Publishers, Feb 1999.

[15] E. M. Sentovich et. al., "SIS: A system for sequential circuit synthesis," in *Department of Electrical Engineering and Computer Science, Berkeley, CA 94720*, 1992.

[16] J. Cong, J. Peck, and Y. Ding, "RASP: A general logic synthesis system for SRAM-based FPGAs," in *ISFPGA*, 1996.

[17] J. P. Fishburn and A. E. Dunlop, "TILOS: A posynomial programming approach to transistor sizing," in *ICCAD*, 1985.

[18] R. W. Brodersen and et al., "Methods for true power minimization," in *ICCAD*, 2002.

[19] F. Li, Y. Lin, L. He, and J. Cong, "FPGA power reduction using configurable dual-Vdd," Tech. Rep. UCLA Eng. 03-224, Electrical Engineering Department, UCLA, 2003.