FPGA Power Reduction Using Configurable Dual-Vdd *

Fei Li, Yan Lin and Lei He

Electrical Engineering Department
University of California, Los Angeles, CA
{feil, ylin, lhe}@ee.ucla.edu

ABSTRACT

Power optimization is of growing importance for FPGAs in nanometer technologies. Considering dual-Vdd technique, we show that configurable power supply is required to obtain a satisfactory performance and power tradeoff. We design FPGA circuits and logic fabrics using configurable dual-Vdd and develop the corresponding CAD flow to leverage such circuits and logic fabrics. We then carry out a highly quantitative study using area, delay and power models obtained from detailed circuit design and SPICE simulation in 100nm technology. Compared to single-Vdd FPGAs with optimized Vdd level for the same target clock frequency, configurable dual-Vdd FPGAs with full and partial supply programmability for logic blocks reduce logic power by 35.46% and 28.62% respectively and reduce total FPGA power by 14.29% and 9.04% respectively. To the best of our knowledge, it is the first in-depth study on FPGAs with configurable dual-Vdd for power reduction.

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, configurable

1. INTRODUCTION

Due to the large number of transistors used to provide programmability for different applications, FPGAs have a much lower power efficiency compared to ASICs. Modeling and reduction of FPGA power has drawn increasing attention recently. [1] presented FPGA power analysis and evaluation. [2] introduced a hierarchical interconnect architecture with low-swing long wires. [3] investigated the possibility of power reduction via pre-defined dual-Vdd/dual-Vt fabrics. [4] studied power-aware CAD algorithms for conventional FPGA circuits and architectures. We believe that holistic research and development involving circuits, architectures and CAD algorithms is able to achieve the highest power efficiency for FPGAs, and in this paper present the first in-depth study of FPGA circuits, architectures and CAD algorithms considering configurable dual-Vdd for power reduction.

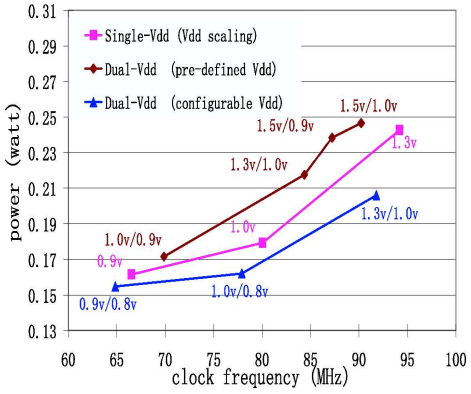

Figure 1: Comparison of three power reduction solutions for benchmark s38584.

A higher supply voltage leads to a higher performance but larger power. Leveraging this, Vdd scaling lowers the supply voltage of the entire design or a circuit module to reduce power. Alternatively, dual-Vdd applies high supply voltage (VddH) to logic on critical paths and low supply voltage (VddL) to logic not on critical paths. For given performance constraints, dual-Vdd is able to achieve more power reduction than Vdd scaling for ASICs [6]. The ASIC designer is able to customize Vdd layout for different applications, but such flexibility does not exist for the existing FPGA circuits and architectures.

Assuming a generic cluster-based FPGA architecture [7], we obtain the power and performance curves for MCNC benchmark circuit s38584 using Vdd scaling and dual-Vdd in Figure 1. We decide a uniform Vdd level for all clusters in Vdd scaling and use a pre-determined Vdd level in dual-Vdd as proposed by [3]. It turns out that this type of dual-Vdd

*This paper is partially supported by NSF CAREER award CCR-0093273 and NSF grant CCR-0306682. We used computers donated by Intel and SUN Microsystems. Address comments to lhe@ee.ucla.edu.

DAC 2004, June 7–11, 2004, San Diego, California, USA.

consumes more power than Vdd scaling with a same clock frequency for this circuit. Such power inefficiency is due to the fact that the pre-defined dual-Vdd fabric imposes extra placement constraints and increases interconnect delay (and power). In contrast, we also present the power and performance curve with a fully configurable dual-Vdd proposed in this paper. The configurable dual-Vdd reduces power significantly compared to Vdd scaling. The first primary contribution of this paper is to show that configurability of power supply is required to achieve FPGA power reduction by using dual-Vdd.

Our other contributions include developing FPGA circuits, architectures and CAD algorithms considering configurable dual-Vdd for power reduction. The rest of the paper is organized as follows. Section 2 presents configurable dual-Vdd FPGA circuits and architectures. Section 3 discusses CAD algorithms and design flow including sensitivity-based dual-Vdd assignment and simulated annealing based dual-Vdd placement. We present the experimental results in Section 4 and conclude in Section 5.

2. DUAL-VDD FPGA CIRCUITS AND ARCHITECTURE

2.1 Logic Block Design

Figure 2(a) shows a generic logic cluster (also called logic block) [7]. A LUT and a flip-flop together form a *basic element* in a logic block. The output of the LUT can be programmed to be a registered output or a combinational output. A logic block consists of N basic elements fully connected by the local interconnects and multiplexers. For our dual-Vdd architecture, we design three types of logic blocks in Figure 2(c)-(d). The first two types, *H-block* and *L-block*, are connected to supply voltages VddH and VddL, respectively. A H-block has the highest performance, and a L-block reduces power consumption but increases delay. The third type is the Vdd-programmable logic block, named *P-block* in Figure 2(d). We implement a P-block by inserting two PMOS transistors between the VddH/VddL rails and a logic block. The PMOS transistors are called *power switches*, and configuration bits are used to control the power switches so that an appropriate supply voltage can be chosen for the P-block. Using one configuration bit to turn off only one of the power switches allows customizing the Vdd level for a P-block. In addition, using two configuration bits to turn off both power switches enables power gating for an unused P-block.

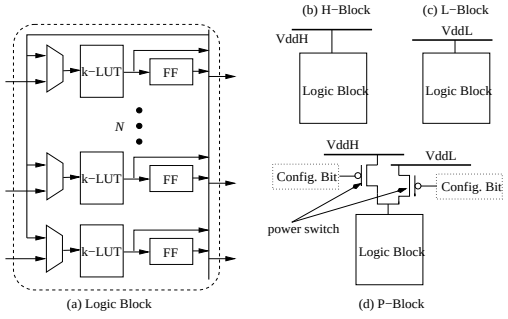

Figure 2: Logic blocks in a configurable dual-Vdd FPGA.

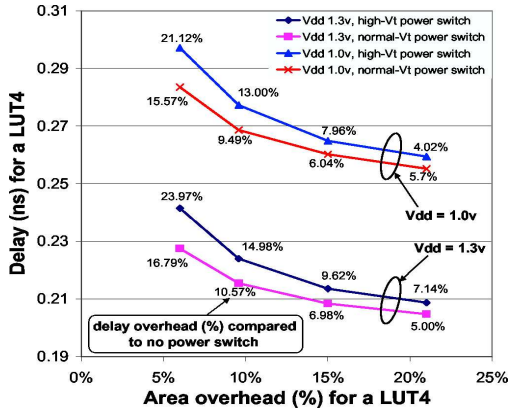

Figure 3: Area and delay overhead of the power switch for a 4-input LUT.

Vdd	P-block leakage power in power-off state (watt)	
	normal power switch	gate-boosted power switch
1.3v	3.46E-07	2.17E-09
1.0v	3.37E-07	9.28E-10

Table 1: Power-off state leakage for a P-block containing one 4-LUT.

The power switch is similar to the sleep transistor for power gating [5]. An important design aspect is how to determine the trade-off between sleep transistor size and circuit delay. In our design, we control the area overhead due to power switches in three ways. First, sleep transistors enable forced stacking to reduce leakage in standby mode and they are usually designed with high Vt to achieve even more leakage reduction. Transistors with high Vt have larger on-resistance and have to increase the size to achieve the specified performance. We design power switches with normal Vt so that the transistor area overhead can be reduced. Figure 3 presents the SPICE simulation results for a 4-input LUT with a power switch in 100nm technology. The X-axis is the power switch area in the percentage of original LUT4 area and Y-axis is the corresponding circuit delay. The area is calculated as the equivalent number of minimum width transistors. The delay overhead due to power switch insertion is labeled beside each data point. Clearly, for a same area overhead, a power switch with normal Vt has a smaller delay compared to a power switch with high Vt. Our simulation shows that, compared to 4-input LUT without any power switch at the same Vdd level, an optimized 4-input LUT with a power switch has 5% extra delay and 21% transistor area overhead. Compared to high Vt, normal Vt for the power switch leads to a relatively higher leakage. To compensate this, we introduce gate boosting for power switches. When a power switch is turned off, we apply a gate voltage one Vt higher than the Vdd at its source node. Table 1 shows that a gate-boosted power switch can reduce leakage by two orders of magnitude compared to normal power switches with the same transistor size. There is neither technology nor design barrier to gate-boosting as it has been used in commercial Xilinx FPGAs [7] to compensate the logic '1' degradation of NMOS pass transistors in routing switches.

To further reduce the power switch area, we leverage the

fact that peak current for different parts of a circuit normally do not occur at the same time [5]. We insert power switches for each logic block and then carry out SPICE simulation. For a logic block with cluster size of 10, only 12% area overhead is required to achieve the same 5% performance loss. Therefore, large granularity significantly reduces the power switch size and the area overhead. We decide to insert power switches at the logic block level. Two power switches per logic block are needed for dual Vdd selection. According to Figure 3, the area of a power switch for the same delay increase is not sensitive to the Vdd level. Therefore, a P-block logic cluster requires 24% area overhead for a bounded delay increase of 5%. Finally, we can reduce the percentage of P-blocks in our dual-Vdd architecture and hence reduce the number of power switches and area overhead in the entire FPGA. This will be discussed in detail in Section 4.

Power switches also introduce extra power. The power to charge and discharge source/drain capacitors of power switches is almost ignorable, as the power switch transistor stays either ON or OFF during normal operation and almost no charging or discharging happens. The primary extra power source is from the configuration SRAM cells to control the power switches. SRAM-based FPGAs use a large number of SRAM cells and our dual-Vdd FPGA further introduces more SRAM cells for supply configuration. Because configuration cells constantly stay in the read status after FPGAs are programmed (excluding dynamically reconfigurable designs), their read or write delays are irrelevant to the design performance. We can increase Vt as much as possible to achieve maximal leakage reduction without performance loss, but high Vt increases the configuration time and affects the signal integrity in SRAM cells. In our design, we increase Vt of SRAM cells for 15X leakage reduction while maintaining the signal integrity and increasing the configuration time by 13%. Because the configuration time is not critical in most FPGA applications, this trade-off is justified. Our experiments in Section 4 assume the same high-Vt SRAM cells for all FPGA architectures.

2.2 FPGA Architecture

Using the three types of logic blocks, we design our dual-Vdd FPGA architecture. Different supply voltages are applied to logic blocks while VddH is used uniformly for the routing resources. There is a Vdd level converter at each output of a L-block as an interface between VddL logic block and VddH routing channels to avoid excessive DC power. We use the asynchronous level converter [8] in Figure 4(a), and size the level converter as in [3] to achieve a bounded delay with minimum power consumption. Table 2 presents the delay and power of the sized level converter. There is also a level converter at each output of a P-block, and the output can be programmed to either go through the level converter or bypass it. No level converters are needed for a H-block. Because our smallest Vdd region is a logic block, we need to design the layout pattern using the three types of logic blocks. In this paper, we assume *interleaved* layout pattern as in Figure 4(b). H-blocks, L-blocks and P-blocks are interleaved in a fixed sequence within each row (H-block → L-block → P-block in the figure). The ratio between different types of blocks is an architectural parameter, determining the length of the repetitive sequence (in Figure 4(b), the ratio of H-block/L-block/P-block is 1/1/2 and sequence length is four logic blocks). For two adjacent

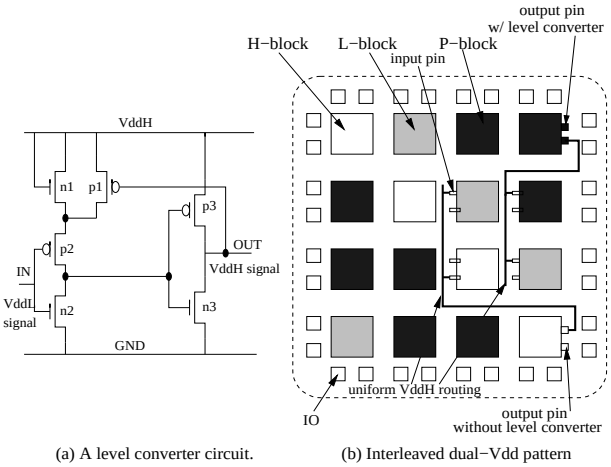

Figure 4: (a) A level converter; (b) Interleaved dual-Vdd layout pattern. (Ratio: H-block/L-block/P-block = 1/1/2; interleaved sequence is obtained by shifting one block between adjacent rows)

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

Table 2: Delay and power for a level converter.

rows, the starting point of the interleaved sequence is shifted by one logic block. Compared to the pre-defined dual-Vdd fabric [3], such interleaving makes it much easier to place a logic block close to the needed Vdd rail. The ratio between three types of logic blocks provides a trade off between power switch area and flexibility of the FPGA architecture.

3. DESIGN FLOW

We develop the design flow in Figure 5 to leverage the new dual-Vdd architecture. Given a single-Vdd gate-level netlist, we first apply single-Vdd technology mapping and timing driven packing [7] to obtain a cluster-level netlist. We then perform single-Vdd timing-driven placement and routing by VPR [7] and generate the basic circuit netlist (BC-netlist). The BC-netlist is defined in [1] and it is annotated with capacitance, resistance, and switching activity for each node.

As the first step to consider dual Vdd in our design flow, Vdd assignment is performed to obtain a dual-Vdd BC-netlist. A dual-Vdd BC-netlist is extended from the original BC-netlist by annotating supply voltage for each logic block. We do not consider layout pattern constraint for dual-Vdd assignment and apply a sensitivity-based algorithm similar to the Vdd assignment algorithm in [10]. Power sensitivity with respect to supply voltage, i.e., $\Delta P/\Delta V_{dd}$ is calculated for logic blocks. The total FPGA power P includes both switching power P_{sw} and leakage power P_{lkg}. For each node i, we have switching power $P_{sw}(i) = 0.5 f_{clk} \cdot E_i \cdot C_i \cdot V_{dd}^2$, where E_i and C_i are transition density and load capacitance. Leakage power $P_{lkg} = I_{lkg}(V_{dd}) \cdot V_{dd}$. We pre-characterize I_{lkg} and logic block delay at each Vdd level using SPICE simulation. Figure 6 presents the sensitivity-based assignment algorithm.

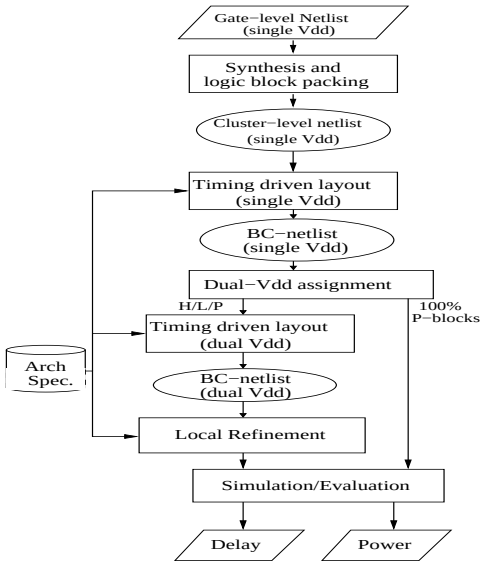

Figure 5: Design flow for dual-Vdd FPGAs.

Sensitivity-based dual-Vdd assignment algorithm:
Assign VddH to all logic blocks;
While (∃ logic blocks not tried)
{
  Obtain logic blocks on circuit path with largest timing slack;
  Calculate power-sensitivity S for these logic blocks;
  Assign VddL to the block with largest S if no critical delay increase;
  Update timing slack and mark the logic block as tried;
}

Figure 6: Sensitivity-based dual-Vdd assignment.

After the dual-Vdd assignment, we have two different design paths. If there are 100% P-blocks in the architecture, the dual-Vdd BC-netlist generated by the Vdd assignment is simulated to obtain the delay and power. We enhance the FPGA evaluation package *fpgaEva-LP* [1] for our power and delay evaluation.

When a parameterized dual-Vdd architecture contains H-blocks or L-blocks, layout pattern constraint applies because the supply voltages for H-blocks and L-blocks are not configurable. The corresponding design path goes through additional steps of dual-Vdd placement and local refinement[1]. We develop our dual-Vdd placement based on the simulated annealing algorithm in VPR. It models an FPGA as a set of legal slots at which logic blocks or I/O pads can be placed via a simulated annealing procedure. *Moves* are defined as either swapping two logic blocks or moving a logic block to an empty slot. We change the placement cost function to consider dual-Vdd architecture. The cost of moving a logic block j to an empty slot is

$$\begin{aligned} Cost(move) &= term1 + \alpha \cdot \Delta matched(j) \\ &\quad + \gamma \cdot (1 - matched(j)) \\ &\quad + \beta \cdot \Delta prog(j) + \theta \cdot prog(j) \end{aligned} \quad (1)$$

where $term1$ is the original cost function in VPR considering both congestion and wire length. $matched(j)$ is a Boolean function describing the Vdd-matching state of a logic block in the new slot and is defined as

$$matched(j) = \begin{cases} 1 & \text{VddL block } j \text{ in slot of L-block or P-block} \\ 1 & \text{VddH block } j \text{ in slot of H-block or P-block} \\ 0 & \text{Otherwise} \end{cases}$$

If the Vdd assigned to block j matches the Vdd at its physical location, $matched(j)$ returns value '1'. Otherwise, it returns '0'. Because the power supply of a P-block is configurable, any logic block placed in a P-block slot returns a matched value. $\Delta matched(j)$ is the difference of $match(j)$ to penalize moving block j from a Vdd-matched location to an unmatched location. The term $1 - matched(j)$ penalizes moving block j from a Vdd-unmatched location to another unmatched location. Considering the power and delay overhead of a P-block, we further penalize the Vdd-matched location at a P-block slot other than a H-block or L-block slot. Similar to $matched(j)$, $prog(j)$ is the Boolean function that designates whether the current location of block j is a P-block slot or not. The term $\Delta prog(j)$ penalizes moving block j from a Vdd non-programmable slot to a Vdd programmable slot, and the term $prog(j)$ penalizes moving block j from a Vdd programmable slot to another Vdd programmable slot. Weights α, β, γ and θ are determined experimentally for better power performance trade-off. The cost of swapping two logic blocks is the sum of the costs given by (1) for the two blocks.

After dual-Vdd placement, we perform local refinement to further reduce power or enhance performance. We go through all P-blocks in the architecture and perturb their power supply configurations. Two types of perturbation are performed. First, if a P-block programmed to VddH can use VddL without increasing critical path delay, we re-program it to VddL. This perturbation reduces power without delay penalty and is always beneficial. Second, we re-program all VddL P-blocks on critical paths to VddH. This perturbation increases circuit performance at the cost of larger power consumption. In our experiments, the critical path consists of only a small number of logic blocks due to large interconnect delay in FPGAs. Therefore, our second perturbation can improve the circuit performance with only little power increase. The effectiveness of local refinement is shown by the results in Section 4.

4. EXPERIMENTAL RESULTS

We carry out evaluation experiments for three architectures: arch-SV, arch-DV and ideal-DV. We use low-leakage SRAM cells for configuration bits in all three architectures and therefore our comparison does not include SRAM leakage power savings. *arch-SV* is the conventional single-Vdd architecture considering uniform-Vdd scaling and correspondent Vt scaling. *arch-DV* is our dual-Vdd architecture and two types of arch-DV are studied in this paper. One is the arch-DV with 100% P-blocks. The other is the arch-DV with a fixed mixture of L-blocks, H-blocks and P-blocks, and we name it as ratio-ed arch-DV. *ideal-DV* does not have any P-blocks but assumes that the mixture and placement of H-blocks and L-blocks can be *perfectly* customized for each individual application. Compared to arch-DV with 100% P-blocks for customizing power supply for every logic block, ideal-DV has neither power and delay overhead associated with P-blocks nor the capability to turn off the unused logic blocks by power-gating.

We determine the ratio of three types of logic blocks in a ratio-ed arch-DV by profiling benchmarks. We perform

[1]we use uniform VddH for routing resources, and the routing algorithm does not need to consider dual-Vdd.

dual-Vdd assignment to MCNC benchmark circuits and obtain the percentage of logic blocks assigned with VddL. The percentage varies from 53% to 96% and the average is around 75%. To cover this wide range of variation while limit the overhead due to P-blocks, we set the ratio H-block/L-block/P-block (or H/L/P)[2] to 1/1/3. Table 3 presents the transistor area overhead due to the power switches for P-blocks. arch-DV with 100% P-blocks has power switch area around 24% of total logic block transistor area. The overhead is reduced to 14% by using the ratio H/L/P = 1/1/3. Because FPGA area is mainly dominated by routing area in the channels [7], such area overhead not necessarily leads to the same amount of chip area increase.

	arch-SV	arch-DV	
		H/L/P = 1/1/3	100% P-blocks
area overhead	0%	*14%*	24%

Table 3: Power switch transistor area (% of total logic block transistor area).

4.1 Local Refinement

Before we present the complete evaluation results, we show the effectiveness of the local refinement by comparing the results before and after refinement. Columns 2 - 3 in Table 4 show the number of P-blocks configured as VddL. After the local refinement, we increase the number of VddL P-blocks by 6.52% on average and intuitively reduce the logic power. Local refinement also reduces critical path delay by 4.23% (see columns 5 - 7 of Table 4). Therefore, our local refinement is effective to improve circuit performance and reduce power consumption in logic blocks.

circuit	# of VddL P-blocks			critical path delay (ns)		
	before refine.	after refine.	(% diff)	before refine.	after refine.	(% diff)
alu4	83	70	-15.66%	11.3	10.2	-9.07%
apex4	54	59	9.26%	11.2	10.8	-3.76%
bigkey	164	170	3.66%	6.84	6.53	-4.45%
clma	774	793	2.45%	25.3	24.3	-3.75%
des	105	110	4.76%	11.9	11.0	-6.98%
diffeq	100	110	10.00%	14.0	13.7	-2.43%
dsip	58	78	34.48%	5.94	5.61	-5.60%
elliptic	229	233	1.75%	17.4	16.6	-4.45%
ex5p	49	47	-4.08%	11.8	11.2	-4.81%
frisc	344	342	-0.58%	24.6	24.2	-1.55%
misex3	59	83	40.68%	10.7	10.7	0%
pdc	268	294	9.70%	16.0	15.6	-2.36%
s298	119	120	0.84%	22.9	21.8	-5.07%
s38417	459	467	1.74%	15.9	15.0	-5.90%
s38584.1	406	409	0.74%	11.5	11.4	-0.95%
seq	72	97	34.72%	10.7	10.3	-3.48%
spla	226	187	-17.72%	15.2	13.7	-10.1%
tseng	70	70	0%	14.2	14.1	-1.38%
avg.			6.52%			-4.23%

Table 4: Local refinement results in the design flow.

4.2 Architecture Comparison

We carry out evaluation experiments on MCNC benchmark circuits. arch-DV with 100% P-blocks and ratio-ed arch-DV with H/L/P = 1/1/3 are compared to arch-SV.

[2]Our experiments show that most P-blocks are configured to VddL for many of our benchmarks. We tried higher percentage of L-blocks to further reduce P-block overhead, but the reduced layout flexibility incurs larger delay penalty.

Figure 7 presents the evaluation result for benchmark circuit *s38584*. The X-axis is the clock frequency calculated as the reciprocal of critical path delay. The Y-axis is the total power consumption. The four power-performance trade-off curves in each figure correspond to arch-SV, the two types of arch-DV and ideal-DV. For arch-SV, we show the power and performance trend as we perform the Vdd scaling. arch-SV also serves as our baseline for the power savings of arch-DV. To obtain the power performance curves for arch-DV, we try different dual-Vdd combinations. After we get all the data points, we prune the inferior data points (i.e., those with larger power consumption and lower clock frequency). These remaining solutions form the spectrum of power-performance trade-off using different VddH/VddL combinations. We label the VddH/VddL combination for each data point. For the two types of arch-DV, it is clear that arch-DV with 100% P-blocks reduces more power compared to arch-DV with ratio H/L/P as 1/1/3, but use more area for power switches as shown in Table 3. In the clock frequency range of our experiments, the power saving by arch-DV is larger at the higher frequency end. This is because high clock frequency usually requires high supply voltage and more surplus timing slack can be utilized at the logic block level. Arch-DV with full supply programmability is almost as good as ideal-DV. For benchmark s38584, our arch-DV with 100% P-blocks even beats ideal-DV at high frequency because the benefit of power-gating capability in our architecture exceeds the power overhead of dual-Vdd.

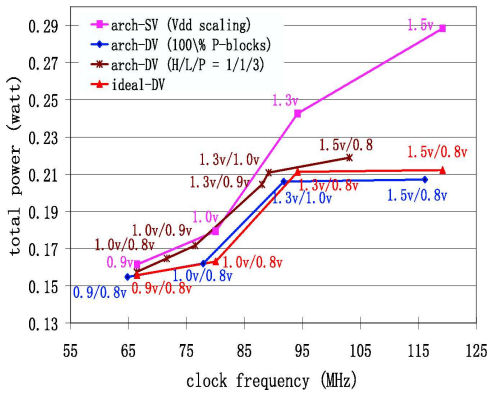

Figure 7: Evaluation results for benchmark s38584.

Table 5 summarizes the evaluation for all the benchmarks. For each benchmark circuit, we choose the maximum clock frequency achieved by arch-DV (H/L/P ratio of 1/1/3) among all Vdd combinations and present the corresponding power saving at that clock frequency. On average, arch-DV with 100% P-blocks achieves 35.46% logic power saving and arch-DV with ratio-ed H/L/P achieves 28.62% logic power saving. We further break down the logic power saving into two parts: power reduction due to power-gating of unused logic blocks and power reduction due to dual-Vdd for utilized logic blocks. Table 5 shows that power-gating of unused blocks has very limited contribution in our overall logic power saving. Most of our logic power saving comes from the dual-Vdd for utilized logic blocks. The average logic power saving due to power-gating for the two types of arch-DV is 4.20% and 7.00%, respectively. Because VPR uses the smallest chip size that just fits a benchmark for placement

circuit	arch-SV (baseline)		arch-DV (H/L/P = 1/1/3)			arch-DV (100% P-blocks)		
	logic P (W)	total P (W)	logic P saving	(by power-gating)	total P saving	logic P saving	(by power-gating)	total P saving
alu4	0.0112	0.0769	27.06%	(0.94%)	12.66%	34.20%	(1.57%)	15.83%
apex4	0.0063	0.0500	24.33%	(2.84%)	4.18%	22.18%	(4.74%)	7.58%
bigkey	0.0331	0.1375	39.79%	(12.42%)	19.16%	53.39%	(20.71%)	24.89%
clma	0.0532	0.5450	24.78%	(0.34%)	3.07%	30.07%	(0.57%)	8.82%
des	0.0448	0.2136	46.12%	(18.13%)	10.36%	56.26%	(30.22%)	19.07%
diffeq	0.0068	0.0360	20.47%	(0.49%)	7.02%	25.39%	(0.81%)	11.01%
dsip	0.0277	0.1280	49.20%	(20.78%)	22.27%	66.46%	(34.64%)	24.17%
elliptic	0.0176	0.1236	26.49%	(1.54%)	7.89%	35.10%	(2.56%)	11.62%
ex5p	0.0079	0.0534	27.51%	(3.90%)	10.51%	22.94%	(6.51%)	8.50%
frisc	0.0204	0.1603	23.55%	(1.88%)	4.51%	33.36%	(3.14%)	9.57%
misex3	0.0081	0.0682	21.67%	(2.75%)	2.17%	22.06%	(4.59%)	8.12%
pdc	0.0201	0.2317	20.26%	(0.89%)	4.41%	28.56%	(1.48%)	8.32%
s298	0.0114	0.0714	23.36%	(0.13%)	6.21%	26.32%	(0.22%)	12.87%
s38417	0.0511	0.2995	23.01%	(1.38%)	4.45%	31.27%	(2.30%)	17.45%
s38584	0.0459	0.2590	36.34%	(0.66%)	15.47%	49.88%	(1.09%)	24.99%
seq	0.0106	0.0924	25.35%	(3.40%)	3.38%	27.11%	(5.65%)	8.54%
spla	0.0165	0.1684	28.46%	(0.48%)	15.25%	32.32%	(0.79%)	14.64%
tseng	0.0063	0.0325	27.39%	(2.67%)	9.81%	41.47%	(4.44%)	21.20%
avg.	-	-	28.62%	(4.20%)	9.04%	35.46%	(7.00%)	14.29%

Table 5: Power savings by arch-DV compared to baseline arch-SV at the same max frequency in experiments.

and routing, we have a high average logic utilization of 84% in our experiments. But in reality, the saving by power-gating can be much larger due to the much lower utilization in FPGAs [11]. Our arch-DV only tackles power reduction for logic blocks because it uses uniform VddH for routing resources. As shown in Table 5, the average total FPGA power savings for arch-DV with 100% P-blocks and arch-DV with ratio-ed H/L/P are 14.29% and 9.04%, respectively.

5. CONCLUSIONS AND DISCUSSIONS

We have shown that configurable Vdd is required to obtain a satisfactory performance and power tradeoff in FPGAs. We have designed circuits and logic fabrics using programmable dual Vdd, and have developed a CAD flow to leverage such circuits and logic fabrics. We have carried out a highly quantitative study using area, delay and power models obtained from detailed circuit design and SPICE simulation in 100nm technology. Compared to single-Vdd FPGAs with Vdd level optimized for the same target clock frequency, dual-Vdd FPGAs with full supply programmability for logic blocks reduce logic power by 35.50% and increase logic block area by 24%. Dual-Vdd FPGAs with partial supply programmability reduce logic power by 28.62% and increase logic block area by 14%. Because FPGA chip area is mainly determined by routing area, such logic area increase is not significant. To the best of our knowledge, it is the first in-depth study on FPGAs with configurable dual-Vdd.

Power supply network to support configurable Vdd or dual-Vdd may introduce extra routing congestion. Leveraging our recent research on optimal synthesis of sleep transistors and power supply network [12, 13], we will study power delivery design and optimization for configurable dual-Vdd FPGAs. Currently, we only apply configurable Vdd to logic blocks. The total power reduction percentage for dual-Vdd FPGAs is significantly lower than the logic power reduction percentage. We will study how to reduce interconnect power by dual-Vdd in the future.

6. REFERENCES

[1] F. Li, D. Chen, L. He, and J. Cong, "Architecture evaluation for power-efficient FPGAs," in *FPGA Symp.*, pp. 42–50, Feb 2003.

[2] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *ISLPED*, pp. 155–160, August 1998.

[3] F. Li, Y. Lin, L. He and J. Cong, "Low-Power FPGA Using Pre-defined Dual-Vdd/Dual-Vt Fabrics," in *FPGA Symp.*, pp. 42–50, Feb 2004.

[4] J. Lamoureux and S. J. Wilton, "On the interaction between power-aware FPGA CAD algorithms," in *ICCAD*, pp. 701–708, Nov 2003.

[5] J. Kao, S. Narendra and A. Chandrakasan, "MTCMOS hierarchical sizing based on mutual exclusive discharge patterns," in *DAC*, pp. 495–500, Jun 1998.

[6] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D. W. Stout, S. W. Gould and J. M. Cohn, "Managing power and performance for SoC designs using voltage islands," in *ICCAD*, pp. 195–202, Nov 2002.

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

[8] 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 *DAC*, pp. 788–793, 2003.

[9] J. Cong and Y. Ding, "FlowMap: An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA Designs," in *Trans. on CAD*, pp. 1–12, Jan 1994.

[10] R. W. Brodersen, M. A. Horowitz, D. Markovic, B. Nikolic and V. Stojanovic, "Methods for true power minimization," in *ICCAD*, pp. 35–42, 2002.

[11] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. J. Irwin and T. Tuan, "Reducing Leakage Energy in FPGAs Using Region-Constrained Placement," in *FPGA Symp.*, pp. 51–58, 2004.

[12] C. Long and L. He, "Distributed Sleep Transistor Network for Leakage Power Reduction," in *DAC*, pp. 181–186, Jun 2003.

[13] C. Long, J. Xiong and L. He, "On Optimal Physical Synthesis of Sleep Transistors," in *ISPD*, Apr 2004.