Routing Track Duplication with Fine-Grained Power-Gating for FPGA Interconnect Power Reduction

Yan Lin, Fei Li and Lei He

Electrical Engineering Department
University of California, Los Angeles

ABSTRACT

Power has become an increasingly important design constraint for FPGAs in nanometer technologies, and global interconnects should be the focus of FPGA power reduction as they consume more power than logic cells. We design area-efficient circuits for programmable fine-grained power-gating of individual unused interconnect switches, and reduce interconnect leakage power dramatically because the interconnect switches have an intrinsically low utilization rate for the purpose of programmability. The low leakage interconnect via power-gating reduces total power by 38.18% for the FPGA in $100nm$ technology. Furthermore, it enables interconnect dynamic power reduction. We design a routing channel containing abundant or duplicated routing tracks with pre-determined high and low Vdd, and develop routing algorithm using low Vdd for non-critical routing to reduce dynamic power. The track-duplicated routing channel has small leakage power and increase the FPGA power reduction to 45.00%.

1. INTRODUCTION

The power efficiency of FPGA is much lower than that of ASIC due to its field programmability. FPGA power modeling and optimization have drawn increasing attention. [1, 2] present power evaluation frameworks, 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. Several FPGA power reduction techniques have been proposed. [4] introduces an inversion method to reduce active leakage power of multiplexers. [5] investigates power-gating and applies region-constrained placement to reduce leakage power of unused logic blocks. [6] proposes pre-defined dual-Vdd/dual-Vt fabrics to reduce both dynamic and leakage power, but the lack of flexibility to customize dual-Vdd layout pattern for different applications may diminish the power saving. Configurable dual-Vdd FPGA is further proposed in [7] such that the Vdd level in any logic block can be programmed for different applications with negligible performance decay and significantly larger logic power reduction. The aforementioned work mainly focuses on logic block power reduction, but the total power reduction is not significant. For example, when configurable dual-Vdd is applied to logic blocks [7] the logic and local interconnect power is reduced by 37.94% but the total FPGA power is reduced only by 14.29%. Therefore, global interconnect should be the focus of FPGA power reduction. [8] introduces a hierarchical interconnect structure and applies low-swing circuits to long interconnects. Low-swing circuits are complicated to design and are less robust, and they have not been widely used in either full custom designs or FPGA designs. Recently, [9] extends programmable Vdd to individual FPGA interconnect switches and achieves significant total power reduction. However, there is a large transistor area overhead associated with the fine-grained Vdd-programmable interconnect. The Vdd-programmable routing switch in [9] needs two power switches, one configurable level converter and extra configuration SRAM cells. Our first contribution of this work is to introduce programmable fine-grained power-gating for interconnects without using extra SRAM cells. We design a novel circuit using the existing SRAM cell in the conventional tri-state buffer to program power-gating. And the number of power switches is reduced almost by half for the same channel width because we only provide power-gating capability. This fine granularity obtains the largest leakage power reduction, equivalent to the total power reduction of 38.18% for the FPGA in $100nm$ technology, but without introducing any extra routing constraint.

Furthermore, the low leakage interconnect enables interconnect dynamic power reduction. Our second contribution is to design a routing channel containing abundant or duplicated routing tracks with 50% using high Vdd (VddH) and other 50% using low Vdd (VddL) repectively, and develop routing algorithm using VddL for non-critical routing to reduce dynamic power. The track-duplicated routing channel has small leakage power by using fine-grained power-gating and increases the FPGA power reduction to 45.00%. It is significantly higher than the total power reduction of 14.29% in [7] where only the power of logic blocks is reduced.

The rest of the paper is organized as follows. Section 2 describes the motivation for low leakage interconnect, the underlying circuit design and experimental results. Section 3 presents track-duplicated routing channel with dual-Vdd, the corresponding CAD algorithms and experimental results. We conclude this paper in Section 4.

2. LEAKAGE POWER REDUCTION

2.1 Background and Motivation

Interconnects are the largest area and power consumer in FPGAs. Figure 1 (a) shows the traditional island style routing architecture. The logic blocks are surrounded by routing channels consisting of wire segments. There is a routing *switch block* at each intersection of a horizontal channel and a vertical channel. Some of the wire segments incident to a switch block can be connected together to form longer connections by programming the routing switches in the switch block. The input and output pins of a logic block can

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

be connected to the wire segments in the surrounding channels via a *connection block*. The switch block flexibility F_s is defined

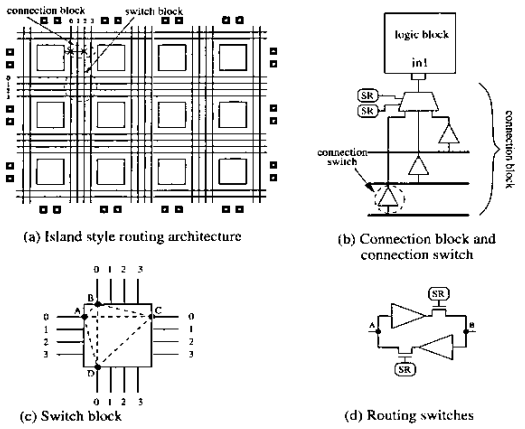

Figure 1: (a) Island style routing architecture; (b) Connection block and connection switch; (c) Switch block; (d) Routing switches.

as the number of outgoing tracks to which an incoming track can be connected. Figure 1 (c) presents a switch with $Fs = 3$. The connection pattern relating the incoming track and three outgoing tracks defines the switch block type. The switch block in Figure 1 (c) is the subset switch block [10]. The incoming track can be only connected to the outgoing tracks with the same track number [1]. The connections in a switch block (i.e., the dashed lines in Figure 1 (c)) are programmable routing switches. A routing switch in Figure 1 (d) can be implemented by a tri-state buffer and each connection needs two tri-state buffers to be programmed for either direction. Another type of routing resource is the connection block [11] (see Figure 1 (b)). The multiplexer-based implementation chooses only one track in the channel and connects it to the logic block input pin. The tri-state buffers between the routing track and the multiplexer are connection switches.

The utilization rate of FPGA interconnect switches is extremely low. As shown in Table 1, the average utilization rate for MCNC benchmark suite is only 11.90%. Note that for each given benchmark, we use the smallest FPGA array that just fits the circuit for placement. In reality, the chip size can be significantly larger than necessary for a given circuit, and the utilization rate can be even lower. Due to this extremely low utilization rate for interconnect switches, a large portion of the interconnect leakage power is consumed by those unused interconnect switches. We develop novel FPGA interconnects with fine-grained power-gating to reduce leakage power for unused interconnect switches.

2.2 FPGA Interconnects with Fine-grained Power-gating

We design FPGA interconnects with fine-grained programmable power-gating. The granularity to apply power-gating is individual programmable switch (routing or connection switch). Figure 2 (a) shows the circuit design for a power-gating capable switch. Based on a conventional tri-state buffer, we insert a PMOS transistor M2 between the power rail and the tri-state buffer for power-gating. When a switch is not used, transistor M1 is to be turned off by the SRAM cell SR. At the same time, we can turn off M2 to perform

[1]We use subset switch block in this paper to study low leakage FPGA interconnects.

circuit	total interconnect switches	unused interconnect switches	utilization rate
alu4	36478	31224	14.40%
apex4	43741	37703	13.80%
bigkey	43741	37703	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.

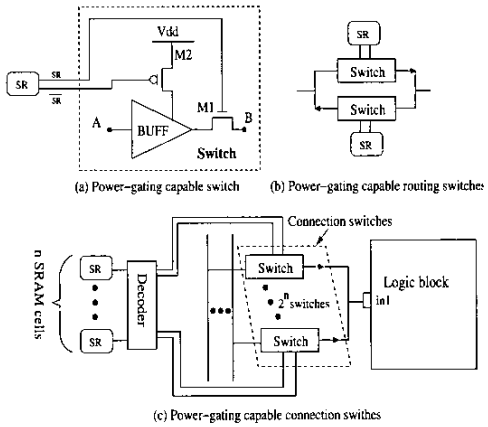

Figure 2: (a) Power-gating capable switch; (b) Power-gating capable routing switches; (c) Power-gating capable connection switches. (SR stands for SRAM cell)

power-gating for the unused switch. Similarly, both M1 and M2 are to be turned on by the SRAM cell SR when the switch is used. Thus, we do not need to introduce an extra SRAM cell for power-gating. Figure 2 (b) presents the power-gating capable routing switches. The SPICE simulation shows that power-gating the routing switch can achieve leakage power reduction by a factor of over 300. There is delay overhead associated with the M2 insertion in series. We size M2 for the tri-state buffer to achieve a bounded delay increase compared to a normal tri-state buffer. As shown in Table 2, delay increase bound of 6% is achieved by using 25X minimum width transistor M2. We also show the dynamic power in energy per signal switch for both conventional routing switch and our new routing switch. Because M2 is always ON when the routing switch is used and no charging or discharging occur at its source/drain capacitors, the power overhead due to transistor M2 is almost negligible.

Similarly, power-gating is also applied to connection switch to reduce leakage power. Figure 2 (c) shows the design of power-gating capable connection switches. We only need n SRAM cells to control 2^n connection switches in a connection block via a decoder and provide power-gating capability for connection switch. Note that a power gated switch has virtually no leakage in either routing

Vdd	routing switch delay (ns)		energy per switch (Joule)	
	without power-gating	with power-gating (increase %)	without power-gating	with power-gating
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: The delay and power of a power-gating capable routing switch. We use 7X minimum width tri-state buffer for routing switches and 25X minimum width PMOS transistor for M2.

or connection switches.

2.3 Experimental Results

We conduct experiments for MCNC benchmark suite and compare three architectures, arch-SV, arch-PV and arch-PV-PG. *arch-SV* represents the single-Vdd scaling [6] for the conventional FPGA. It scales down the supply voltage Vdd as well as the transistor threshold voltage Vt for the entire FPGA, and achieves power reduction at the cost of performance degradation. *arch-PV* is the FPGA architecture using programmable dual-Vdd for logic fabric [7]. *arch-PV-PG* is our new architecture that uses the same logic fabric as [7] and further uses power-gating capable interconnects.

Table 3 presents the experimental results. For each circuit, we choose the largest clock frequency achieved by arch-PV-PG 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 column 1, and the power saving by arch-PV in [7] is shown in column 2 for the purpose of comparison. As shown in column 3, our arch-PV-PG achieves 38.18% total power saving on average compared to arch-SV, In contrast, arch-PV in [7] reduces total FPGA power only by 14.29% because it only reduce FGPA logic power.

circuit	Column 1: arch-SV (baseline)		Column 2: arch-PV [7] w/o power-gating	Column 3: arch-PV-PG w/ power-gating
	interconnect power (W)	total power (W)	total power saving	total power saving
alu4	0.0657	0.0769	15.83%	29.09%
apex4	0.0437	0.0500	7.58%	30.70%
bigkey	0.1044	0.1375	24.89%	40.49%
clma	0.4918	0.5450	8.82%	45.69%
des	0.1688	0.2136	19.07%	31.79%
diffeq	0.0292	0.0360	11.01%	45.25%
dsip	0.1003	0.1280	24.17%	43.66%
elliptic	0.1060	0.1236	11.62%	47.04%
ex5p	0.0455	0.0534	8.49%	31.49%
frisc	0.1399	0.1603	9.57%	58.33%
misex3	0.0601	0.0682	8.12%	22.36%
pdc	0.2116	0.2317	8.32%	45.64%
s298	0.0600	0.0714	12.87%	28.64%
s38417	0.2484	0.2995	17.45%	36.60%
s38584	0.2131	0.2590	24.99%	43.28%
seq	0.0818	0.0924	8.54%	24.76%
spla	0.1519	0.1684	14.64%	39.88%
tseng	0.0262	0.0325	21.20%	42.55%
avg.	-	-	14.29%	38.18%

Table 3: Power saving by arch-PV-PG with power-gating compared to baseline arch-SV at the same maximum frequency.

3. DYNAMIC POWER REDUCTION

Dual-Vdd technique makes use of the circuit timing slack to minimize power. High Vdd (VddH) is applied to devices on the critical paths to maintain the performance while low Vdd (VddL) is applied to devices on non-critical paths to reduce power. FPGA applications usually have large amount of surplus timing slack. We may apply dual-Vdd to FPGA interconnect fabric and leverage the surplus timing slack to reduce interconnect dynamic power.

3.1 Pre-determined Dual-Vdd Routing Architecture

Figure 3 presents our dual-Vdd FPGA routing architecture. The routing channel is partitioned into VddH and VddL regions. We use VddH-track and VddL-track to denote the track in VddH and VddL regions respectively. The same circuit design for interconnect switches from Section 2.2 is used in dual-Vdd routing architecture. The only difference between a routing switch in VddH-track and that in VddL-track is the Vdd level. The ratio between number of tracks in VddH and VddL regions is an architectural parameter. To

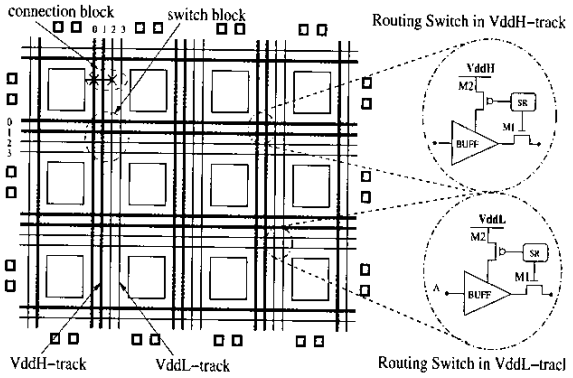

Figure 3: Dual-Vdd FPGA routing architecture.

apply dual-Vdd to FPGA interconnects, we need to perform a dual-Vdd assignment and decide the Vdd level of each circuit element. A routing tree includes driving logic block, driven logic blocks, wire segments and connection/routing switches which connect logic blocks. Subset switch block is used in this paper. As the incoming track can be only connected to the outgoing tracks with the same track number via a subset switch block, a VddH-track can never be connected to a VddL-track. No level converter is needed in switch blocks. Therefore a routing tree is a natural granularity in dual-Vdd assignment as we do not need to change Vdd level inside the routing tree. Table 4 shows that an average of 54.54% routing trees can be assigned with VddL without degrading system performance if there is no layout constraint, i.e., we have the freedom to decide Vdd level of each routing tree. The ratio between VddH and VddL routing trees without considering layout constraint is close to 1:1, which reflects the roughly equal demand for VddH and VddL routing resources. We decide the percentage of VddH and VddL tracks in FPGA routing channel to be 50% respectively.

3.2 Design Flow

We develop new design flow to leverage the fixed dual-Vdd interconnect FPGA fabric. A single-Vdd gate level netlist is given as the input. We apply technology mapping and timing driven packing [11] to obtain the single-Vdd cluster-level netlist. We then perform single-Vdd timing-driven placement and routing by VPR [11] and generate the basic circuit netlist (BC-netlist) [2]. A sensitivity-based Vdd assignment algorithm similar to that in [12] is further used to obtain a dual-Vdd assignment BC-netlist. VddH or VddL is assigned to each routing tree without consideration of layout constraint. Power sensitivity $\Delta P/\Delta V_{dd}$ with respect to supply voltage is calculated for routing tree. 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 device delay

circuit	# of nets	# of logic blocks	# of I/O blocks	VddL nets (%)	VddL logic blocks (%)
alu4	782	162	22	49.74	82.10
apex2	1246	213	41	30.02	69.95
apex4	849	134	28	35.45	78.36
bigkey	1542	294	426	67.77	85.03
clma	7995	1358	144	69.74	89.84
des	1325	218	501	52.83	80.73
diffeq	1291	195	103	63.36	85.64
dsip	1139	588	426	71.29	88.89
elliptic	2617	666	245	78.45	92.16
ex1010	3033	513	20	37.36	71.40
ex5p	834	194	71	20.38	53.66
frisc	3240	731	136	88.49	95.97
misex3	828	181	28	11.35	51.63
pdc	2933	624	56	58.95	88.20
s298	908	66	10	47.03	78.91
s38417	5426	982	135	64.17	80.05
s38584	4502	1046	342	90.94	95.88
seq	1138	274	76	20.74	61.62
spla	2091	461	122	54.52	88.47
tseng	918	305	174	78.21	87.02
Avg				54.54	80.28

Table 4: Percentage for VddL nets and VddL logic blocks given by dual-Vdd assignment under zero delay-increase (VddH = 1.5v and VddL = 1.0v).

at each Vdd level using SPICE simulation.

After the dual-Vdd assignment, we have two different design paths. If the channel width is set to $2.0W$, where W is the nominal channel width [2] to route the benchmark circuit successfully, then the dual-Vdd assignment for routing trees is always feasible given the enough VddH/VddL routing resources. The dual-Vdd BC-netlist with this feasible Vdd assignment is generated and simulated to obtain the power and delay. We enhance the FPGA evaluation package *fpgaEva-LP* [2] for our power and delay evaluation.

When the channel width in dual-Vdd architecture is less than $2.0W$, layout pattern constraint applies because we may not have sufficient VddH and VddL tracks. The corresponding design path goes through an additional step of dual-Vdd routing[3]. We develop our dual-Vdd routing based on the timing-driven routing algorithm in VPR. A directed routing resource graph is constructed to perform global and detailed routing. Logic block pins and wire segments are modeled as vertices and potential connections, i.e., connection/routing switches are modeled as edges in the routing resource graph. To route a k-terminal net, a wave expansion algorithm is invoked $k-1$ times to connect the net source to each of the net's $k-1$ sinks in the order of non-increasing criticality. The neighbors of wire vertex with minimum cost at the wavefront will be expanded first. The cost $TotalCost(n)$ of routing net T through wire segment n to the target sink j is

$$\begin{aligned} TotalCost(n) &= PathCostDv(n) \\ &\quad +\alpha \cdot ExpectedCostDv(n,j) \\ &\quad +\beta \cdot Matched(T,n) \end{aligned} \quad (1)$$

where $PathCostDv(n)$ is the total cost of the path from the current partial routing tree to wire segment n and $ExpectedCostDv(n,j)$ is the estimated total remaining cost from the current wire segment n to the target sink j. $PathCostDv(n)$ and $ExpectedCostDv(n,j)$ are modified from $PathCost(n)$ and $ExpectedCostDv(n,j)$ in VPR with consideration of dual-Vdd. $Matched(T,n)$ is a Boolean function describing the Vdd-matching state of a net T on the wire segment n and is defined as

$$Matched(T,n) = \begin{cases} 1 & \text{VddH net } T \text{ is routed on VddH wire } n \\ 1 & \text{VddL net } T \text{ is routed on VddL wire } n \\ 0 & \text{Otherwise} \end{cases}$$

[2]We define nominal routing channel width W as 1.2X minimum channel width which is defined in [11].

[3]We use configurable dual-Vdd for all logic blocks, and the placement algorithm does not need to consider layout constraint.

If the Vdd assigned to net T matches the Vdd at its physical wire segment n, $Matched(T,n)$ returns value '1'. Otherwise, it returns '0'. Weights α and β are determined experimentally for better power performance trade-off. The experimental result is discussed in the following.

3.3 Experimental Results

We conduct experiments on the MCNC benchmark set and compare four architectures, arch-SV [6], arch-PV [7], arch-PV-PG and arch-DV-PG, where *arch-DV-PG* is the architecture using configurable dual-Vdd for logic blocks and pre-determined dual-Vdd interconnect fabric with power-gating capability.

3.3.1 Architecture Comparison

We first present the power and delay evaluation result for benchmark *s38584* in Figure 4. The X-axis is the maximum clock frequency and the Y-axis is the total power. The four curves presented in the figure correspond to FPGA architectures arch-SV, arch-PV, arch-PV-PG and arch-DV-PG. We always use nominal channel width $1.0W$ for arch-SV, arch-PV and arch-PV-PG. It is because that these three architectures all use the conventional routing architecture and the nominal channel width gives us the best power performance tradeoff. For our pre-determined dual-Vdd routing architecture arch-DV-PG presented in Figure 4, we use channel width $1.5W$ to provide sufficient VddH and VddL routing resources. Comparing these two low leakage interconnect architectures arch-PV-PG and arch-DV-PG, it is clear that we can further reduce FPGA power consumption by using dual-Vdd interconnect with abundant routing resources. By increasing the channel width to $2.0W$ for arch-DV-PG, we can further reduce the total FPGA power due to the increased Vdd-matched rate for routing trees. This phenomenon is analyzed in Section *3.3.2*. The gap between power-performance curve arch-DV-PG and arch-PV-PG 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.

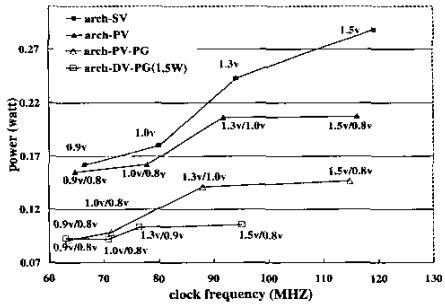

Figure 4: Power-performance for benchmark s38584.

For our dual-Vdd architecture arch-DV-PG, we present the complete evaluation results for MCNC benchmark set in Table 5. For each circuit, we choose the largest clock frequency achieved by arch-DV-PG under all VddH/VddL combinations and present the corresponding power saving at that clock frequency. The power consumption by the baseline arch-SV is presented in column 1, and the power saving by arch-PV from [7] is shown in column 2 for the purpose of comparison. The power saving by arch-DV-PG with channel width $2.0W$ is presented in column 5. We have pre-

circuit	Column 1		Column 2	Column 3			Column 4			Column 5		
	arch-SV (baseline)		arch-PV [7]'	arch-DV-PG (1.0W)			arch-DV-PG (1.5W)			arch-DV-PG (2.0W)		
	interconnect power (W)	total power (W)	total power saving	interconnect power saving		total power saving	interconnect power saving		total power saving	interconnect power saving		total power saving
				dynamic	leakage		dynamic	leakage		dynamic	leakage	
alu4	0.0657	0.0769	15.83%	-5.90%	82.01%	22.04%	6.86%	79.87%	29.15%	17.40%	77.98%	35.08%
apex4	0.0437	0.0500	7.58%	-77.27%	82.43%	18.48%	9.64%	80.33%	38.64%	8.88%	78.41%	36.69%
bigkey	0.1044	0.1375	24.89%	-10.93%	83.77%	29.98%	18.07%	82.17%	41.04%	33.04%	80.21%	48.65%
clma	0.4918	0.5450	8.82%	8.72%	86.41%	50.27%	9.63%	84.46%	48.89%	26.61%	82.49%	53.28%
des	0.1688	0.2136	19.07%	-13.34%	85.80%	24.10%	4.89%	84.11%	30.12%	22.13%	82.27%	40.66%
diffeq	0.0292	0.0360	11.01%	1.99%	79.97%	45.17%	34.71%	78.20%	51.97%	33.79%	76.51%	51.94%
dsip	0.1003	0.1280	24.17%	-10.31%	87.49%	37.97%	9.45%	85.70%	45.73%	33.81%	83.80%	54.28%
elliptic	0.1060	0.1236	11.62%	21.60%	84.10%	52.90%	31.90%	82.39%	53.73%	36.33%	80.34%	54.91%
ex5p	0.0455	0.0534	8.49%	-12.19%	83.25%	36.34%	-12.08%	81.07%	27.18%	10.58%	79.24%	36.19%
frisc	0.1399	0.1603	9.57%	5.46%	85.86%	56.61%	39.95%	83.92%	62.62%	47.93%	82.26%	63.91%
misex3	0.0601	0.0682	8.12%	-9.31%	81.18%	23.69%	-17.79%	79.62%	11.43%	5.76%	77.57%	25.86%
pdc	0.2116	0.2317	8.32%	-10.69%	85.92%	42.35%	-2.51%	83.70%	43.48%	15.60%	81.65%	48.41%
s298	0.0600	0.0714	12.87%	0.64%	80.91%	31.25%	-11.62%	79.07%	23.04%	7.00%	77.24%	32.49%
s38417	0.2484	0.2995	17.45%	-23.33%	83.15%	30.07%	0.68%	81.38%	23.10%	27.70%	79.47%	44.76%
s38584	0.2131	0.2590	24.99%	-3.21%	82.75%	35.34%	33.52%	80.96%	48.20%	49.94%	79.27%	56.52%
seq	0.0818	0.0924	8.54%	-47.99%	83.00%	13.95%	-33.90%	81.02%	11.16%	6.32%	79.18%	30.00%
spla	0.1519	0.1684	14.64%	-25.23%	84.69%	34.54%	8.86%	82.48%	42.77%	16.71%	80.71%	43.65%
tseng	0.0262	0.0325	21.20%	4.64%	78.91%	42.46%	45.86%	77.61%	54.05%	44.48%	75.75%	52.72%
avg.	-	-	14.29%	-11.48%	83.42%	34.86%	9.78%	81.56%	38.68%	24.78%	79.69%	45.00%

Table 5: Power saving by pre-determined dual-Vdd and power-gating capable interconnects architecture arch-DV-PG compared to baseline arch-SV at the same maximum frequency achieved by arch-DV-PG.

sented that arch-PV-PG can achieve average total power reduction of 38.18% in Section 2.3. By using routing track duplication with fine-grained power-gating, arch-DV-PG achieves total power reduction of 45.00%. In contrast, arch-PV in [7] reduces total FPGA power only by 14.29% because it only reduce FGPA logic power by Vdd-programmability for logic blocks.

3.3.2 *Impact of Routing Channel Width*

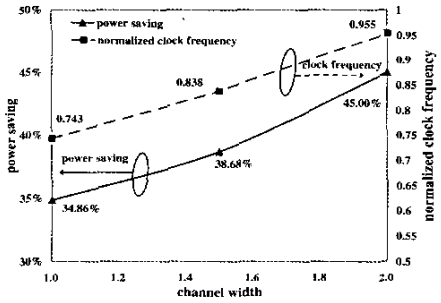

Figure 5: Power reduction and normalized clock frequency versus channel width for arch-DV-PG (VddH = 1.5V).

The routing channel width has an impact on the power saving and the system maximum clock frequency of our new architecture arch-DV-PG. We present the results in Figure 5. The X-axis is the channel width. The right Y-axis is the clock frequency achieved by arch-DV-PG using the best VddH/VddL combination normalized to the maximum clock frequency at single Vdd. The left Y-axis is the total power reduction at that clock frequency compared to baseline arch-SV. There are two curves in the figure. The upper curve presents the normalized clock frequency with arch-DV-PG. The normalized clock frequency is around 0.743 with 1.0X channel width, increases to 0.838 with 1.5X channel width and further increases to more than 0.955 with 2.0X channel width. We can see that the system maximum clock frequency can be improved by providing more routing tracks for Vdd-matched routing trees. It is because that the number of Vdd-matched routing trees increases and routing detours to match the Vdd level reduces as channel width increases. We cannot achieve normalized clock frequency '1' due to the overhead of configurable dual-Vdd logic blocks and switches with power-gating capability. The lower curve presents the power saving with arch-DV-PG. The average total power saving ratio is 34.86% with 1.0X channel width, and increases to 38.68% with 1.5X channel width and 45.00% with 2.0X channel width. We can see that the total power saving also increases with increased channel width. The increased total power reduction is mainly due to the increase of interconnect dynamic power reduction. Table 5 presents the total power saving as well as power saving breakdown with channel width $1.0W$, $1.5W$ and $2.0W$ for arch-DV-PG in column 3-5. The interconnect dynamic power saving is -11.48% with channel width $1.0W$, i.e., we cannot achieve interconnect dynamic power saving at the same clock frequency without increasing channel width. It is because that there is not sufficient tracks for Vdd-matched routing trees and the system performance degrades 25.7% with channel width $1.0W$. The interconnect dynamic power saving increases to 9.78% with channel width $1.5W$ and 24.78% with $2.0W$. By providing more tracks for Vdd-matched routing trees, we can achieve more interconnect dynamic power saving due to the increase of Vdd-matched rate for routing trees and the clock frequency. In general, the increase of channel width will introduce more unused interconnect switches and result in more leakage consumption. As shown in Table 5, the interconnect leakage power reduction is 83.42% with channel width $1.0W$ and slightly decreases to 81.56% with $1.5W$ and 79.69% with $2.0W$. We can see that the interconnect leakage power reduction only slightly decreases when channel width increases. By applying power-gating technique to our dual-Vdd FPGA interconnect fabric, we can achieve interconnect dynamic power saving and control leakage at the same time. The track-duplicated dual-Vdd channel with fine-grained power-gating achieves the best total FPGA power reduction.

3.3.3 *Area Overhead*

Routing track duplication increases the routing area and further

	arch-SV (baseline)	arch-DV-PG (1.0W)	arch-DV-PG (1.5W)	arch-DV-PGA (2.0W)	[9]
total FPGA area (geo. mean over 20 circuits)	7077044	11092744	15420197	20249865	22678225
area overhead (%)	-	57%	118%	186%	220%

Table 6: Total device area for different FPGAs. We use the VPR device area model and the area is in the number of minimum width transistor areas.

amplifies the area overhead due to sleep transistor insertion for fine-grained power-gating. This section studies the total area overhead of Vdd-gatable interconnect fabric with routing track duplication. Assuming that two metal layers are available and the wires can be routed over the devices, the larger one of routing wire area and routing device area determines the final routing area. Channel width, wire width and spacing are used to decide the routing wire area. Table 7 shows the wire width and spacing that we use for 100nm technology. To obtain the device area in square microns, we use the same area model in VPR to get the number of minimum width transistor areas and multiply it by the square microns per minimum width transistor in 100nm technology. For all benchmark circuits, we have found that the routing device area is larger than the routing wire area. Previous work [11] also shows that the area of typical commercial FPGAs is dominated by device area. Therefore, we calculate the total FPGA area as the sum of routing device area and logic device area. We compare the total FPGA area for different FPGA architectures in Table 6. As the routing channel width increases, the area overhead for our arch-DV-PG increases from 57% to 186%. Recently, fine-grained Vdd-programmable FPGA interconnects are proposed in [9], which provides the capability of Vdd selection and power gating to reduce both interconnect dynamic and leakage power. Compared to [9], our Vdd-gatable interconnect with routing track duplication (i.e., channel width 2.0W) also reduces both leakage and dynamic power, but it has less area overhead (see columns 5 and 6 in Table 6). This is because that fine-grained Vdd programmability requires two sleep transistors for each routing switch, one configurable Vdd level converter before each routing switch and associated SRAM cell for configuration, but our arch-DV-PG needs only one sleep transistor for each routing switch and no Vdd level converters in routing channels since only power gating capability is provided. Because our Vdd-gatable interconnect architecture does not have the leakage overhead of level converters, it achieves larger total power saving compared to Vdd-programmable interconnect architecture (45% by our Vdd-gatable interconnect architecture vs. 26% by Vdd-prommable interconnect architecture as verified by the power model in this paper).

wire width	wire spacing	wire thickness
0.56um	0.52um	1.08um

Table 7: Interconnect geometries at 100nm technology.

4. CONCLUSIONS AND DISCUSSIONS

We show that interconnect power is the bottleneck of reducing total FPGA power. Because the FPGA interconnect circuit has an extremely low utilization rate ($\sim$ 12%) for purpose of programmability, we design area-efficient circuits for fine-grained power-gating of interconnects without introducing extra configuration cells to program power-gating. We power-gate each unused interconnect switch and reduce total FPGA power by 38.18% compared to the case without power-gating. Furthermore, the low leakage interconnects enable interconnect dynamic power reduction. We design a routing channel containing abundant or duplicated routing tracks with pre-determined high and low Vdd, and develop corresponding dual-Vdd routing algorithm using low Vdd for non-critical routing to reduce interconnect dynamic power. The track-duplicated channel with fine-grained power-gating reduces interconnect dynamic power by 24.78% and total FPGA power by 45.00%, with a area overhead of 186%. In contrast, as verified by the power model in this paper, the previous Vdd-programmable FPGA interconnect [9] only reduces total FPGA power by 26% due to the large leakage overhead of Vdd level converters in routing channels and has a area overhead of 220%. To further bring down the area overhead for our Vdd-gatable FPGA interconnect, we may reduce the size of the sleep transistors and increase Vdd level to avoid performance loss. We will study the energy saving and area tradeoff in the future. and also plan to extend our fine-grained and SRAM-efficient power-gating techniques to logic blocks.

5. 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 CICC*, 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*, Februray 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*, Februray 2004.

[7] F. Li, Y. Lin, and L. He, "FPGA power reduction using configurable dual-vdd," in *Proc. Design Automation Conf.*, June 2004.

[8] E. Kusse and J. Rabaey, "Low-energy embedded FPGA structures," in *Proc. Intl. Symp. Low Power Electronics and Design*, pp. 155–160, August 1998.

[9] Fei Li, Yan Lin and Lei He, "Vdd programmability to reduce fpga interconnect power," in *Proc. Intl. Conf. Computer-Aided Design*, November 2004.

[10] G. G. Lemieux et. al., "A detailed router for allocating wire segments in field-programmable gate arrays," in *Proceedings of the ACM Physical Design Workshop*, April 1993.

[11] V. Betz et. al., *Architecture and CAD for Deep-Submicron FPGAs*. Kluwer Academic Publishers, Feb 1999.

[12] R. W. Brodersen et. al., "Methods for ture power minimization," in *Proc. Intl. Conf. Computer-Aided Design*, pp. 35–42, 2002.