An Energy-Efficient Near/Sub-Threshold FPGA Interconnect Architecture Using Dynamic Voltage Scaling and Power-Gating

He Qi, Oluseyi Ayorinde, and Benton H. Calhoun
Charles L. Brown Department of Electrical and Computer Engineering
University of Virginia
Charlottesville, Virginia
⟨hq5tj,oaa4bj,bhc2b⟩@virginia.edu

***Abstract*—The rapid development of the Internet-of-Things requires hardware that is both low-energy and flexible, and a near/sub-threshold FPGA is a very promising solution. In the design of near/sub-threshold FPGAs, the biggest challenge is reducing global interconnect energy, which is the most energy-consuming part in the entire FPGA. Dynamic voltage scaling is an effective technique in reducing energy, but it is not widely used in FPGA interconnects because of the high area overhead of separately provisioning the buffers in the switch boxes to support different voltages on different paths. A low-swing interconnect, which removes buffers, allows this technique to be applied to the FPGA interconnects. In this paper, we propose a novel low-swing FPGA interconnect architecture that integrates dynamic voltage scaling and power-gating techniques with custom tool support. While the power-gating technique is widely used in existing designs for reducing leakage energy of idle drivers and buffers, we also apply power-gating to configuration bitcells in switch boxes, because it is a dominant energy consumer in near/sub-threshold. Including the energy overhead of voltage regulators, our work achieves a 10.1% energy saving in active circuits, 27.0% - 91.3% in idle circuits, and 19.0% - 53.1% in the entire FPGA on average, compared to an already optimized base case that only uses low-swing interconnect but no dynamic voltage scaling or power-gating. In addition, our dynamic voltage scaling allows us to adjust the delay of the low-swing FPGA interconnect from 0.14μs to 0.43μs or adjust its energy per operation from 5.5pJ to 35.7pJ when implementing the MCNC benchmarks at 0.6V.**

I. Introduction

The rapid development of wearable electronic devices and Internet-of-Things (IoT) such as wireless health monitors and pedometers requires future hardware to be low-energy, as well as maintaining adequate speed. Although near/sub-threshold (near/sub-V_T) ASICs can already meet these requirements [1], hardware flexibility is also desired for reducing design time and cost. In addition, flexible hardware plays an irreplaceable role in encryption on some of the low-power applications [2]. This new situation makes near/sub-V_T energy-efficient FPGAs an attractive solution. Since the interconnect fabric traditionally consumes the majority of the FPGA energy, an energy-efficient FPGA interconnect is highly desired.

Dynamic voltage scaling (DVS) with power-gating is a useful technique in reducing circuit energy in general. It reduces the leakage energy by disconnecting the idle circuits from their voltage supplies, and minimizes the active energy by adjusting the supply voltages of the circuits according to their dynamic loads. Applying this technique to FPGA interconnects can reduce the overall FPGA energy, so that the FPGAs can be used in more low-power applications. However, no existing works apply DVS to FPGA interconnects. In [5], a FPGA interconnect architecture that uses dual-V_{DD} along with power-gating was proposed. The authors reduced the active energy by applying the nominal supply voltage to the critical path, and applying a lower supply voltage to all the non-critical paths. [6] used the similar technique with some circuit level considerations. In [7], researchers applied DVS to a commercial FPGA, but it is not an on-chip solution. Although dual-V_{DD} is an effective method for reducing interconnect energy, the energy is not minimized without adjusting supply voltages dynamically. In addition, the existing dual-V_{DD} technique is not easy to be implemented on physical FPGA chips because of the high area overhead of adding headers to the buffers in switch boxes (SBs). Also, the designs in the existing works mainly focus on the traditional buffer-based interconnects at the nominal voltage, while wireless IoT applications require FPGAs optimized at near/sub-V_T. Recent works about a novel low-swing interconnect [3,4] provide a potential opportunity of applying DVS along with dual-V_{DD} on near/sub-V_T interconnects. The low-swing interconnect is proven to be an energy-efficient design in sub-V_T. Also, since there are no buffers inside the SBs in the low-swing interconnect, a big portion of the potential area overhead of implementing dual-V_{DD} no longer exists.

In this work, we propose a novel energy-efficient low-swing FPGA interconnect using dual-V_{DD} with headers to implement both per path DVS and power-gating techniques optimized at 0.6V. As the leakage energy in the bitcells becomes significant in near/sub-V_T, our approach reduces energy more effectively than existing works that are designed for high speed operations. To implement DVS, we also designed an on-chip voltage controller. Instead of applying

978-1-5090-5602-6/16/$31.00

DVS by changing the main supply voltage V_{DD}, we change an additional gate-boosted voltage V_{DDC} that is unique to the low-swing interconnect [3,4]. As a result, the FPGA interconnect with the proposed architecture consumes 14.0% - 36.0% lower energy than the best existing work when scaled to the same technology node. In addition, our dynamic voltage scaling allows us to adjust the delay of the low-swing FPGA interconnect from 0.14μs to 0.43μs or adjust its energy per operation from 5.5pJ to 35.7pJ when implementing the MCNC benchmarks at 0.6V.

The rest of this paper is arranged as follows: in Section II, we will introduce the proposed architecture in detail. Our design methodology and analysis, including the low-swing interconnect model, the custom tool flow, the DVS controlling algorithm, and the overheads will be discussed in Section III. In Section IV, we will show our results, observations, and comparisons with previous works. In Section V, the limitations and future work will be discussed. Finally, the conclusions will be addressed in Section VI.

II. The Proposed Architecture

A. The concepts of DVS, dual-V_{DD}, and power-gating

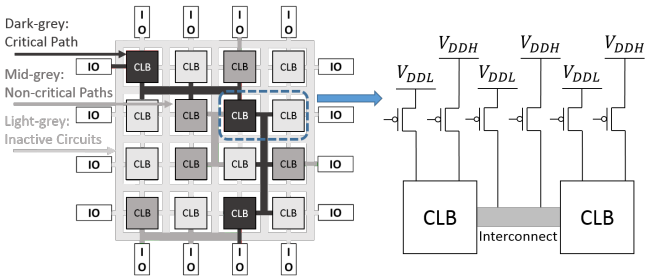

Fig. 1. The concept diagram of per path DVS and power-gating techniques applied to FPGAs

Ultra-Low-Energy FPGAs should be as energy efficient as possible. Reducing the supply voltage on the non-critical paths can effectively minimize energy while maintaining the overall FPGA speed [12]. The dual-V_{DD} scheme is a low-power technique based on this concept. As shown in Fig. 1, the circuits in dark-gray represent the critical path that are attached to V_{DDH} through headers. The circuits in mid-gray represent non-critical paths that are attached to V_{DDL}. The circuits in light-gray are in idle mode and are power-gated. The energy reduction is completed by turning on/off a pair of header transistors. By using two voltage rails, circuits on the critical path are attached to a higher supply voltage V_{DDH}, while the rest of the circuits are attached to a lower voltage V_{DDL}. When both transistors are turned off, the circuit component is power-gated in order to reduce leakage [5,6].

B. Dual-V_{DD} on the low-swing interconnect

The per path DVS is already proven to be capable of minimizing energy in ASICs [13]. However, this technique was not easy to be applied to FPGA interconnects due to the large area overhead. Fig. 2 shows an interconnect model that starts from the driver at the output of a CLB to the sense amplifier (SA) at the input of the destination CLB. In each SB, a configuration bitcell is used to control the switch, and the leakage paths to the ground are included. Because the traditional interconnect fabric has buffers in each SB, implementing fine-grained DVS with dual-V_{DD} on the interconnect requires adding headers and configuration bitcells to each SB. This will substantially bloat the already large area of the SBs. However, as the energy budget of today's IoT applications becomes lower and lower, the traditional interconnect that suffers from high energy in the buffers is no longer the optimal design for ultra-low-energy FPGAs. In [3] and [4], researches developed a low-swing interconnect to achieve extremely low energy and adequate speed by removing buffers. As shown in Fig. 2, the buffers no longer exist in the low-swing interconnect, and the headers used for the voltage assignment are not needed anymore. As a result, applying DVS with dual-V_{DD} to the low-swing interconnect introduces much smaller area overhead than applying the same technique to the traditional FPGA interconnects.

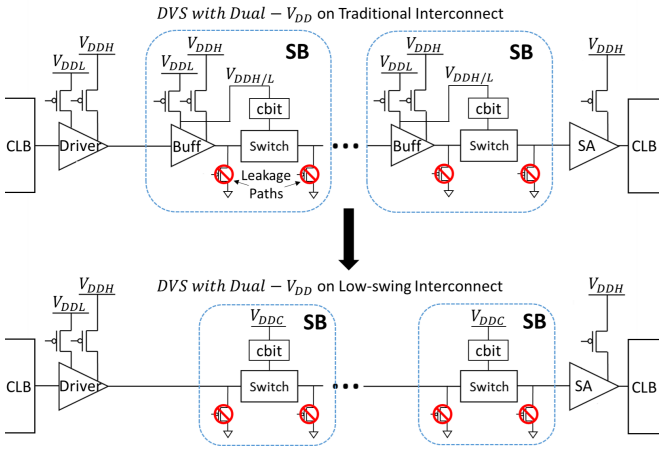

Fig. 2. The comparison of applying dual-V_{DD} technique on the traditional interconnect and the low-swing interconnect

C. The proposed per path DVS architecture

In [3] and [4], the bitcells in SBs are attached to an additional boosted voltage rail V_{DDC}. In this work, we adjust the circuit delay and energy dynamically by sweeping V_{DDC}. Although changing the main supply voltage V_{DD} can adjust delay and energy of circuits, our method is much more energy-efficient for implementing DVS on FPGA interconnects. Since the energy of V_{DDC} is purely leakage in the bitcells, which is much smaller than the energy of the main V_{DD} in the drivers when power-gating is applied to the unused circuits, the energy overhead of the voltage regulator for adjusting V_{DDC} is much smaller than that of V_{DD}. After applying power-gating to the idle bitcells, the energy of the drivers still dominates the total energy of the active circuits. Based on this observation, we propose an architecture that uses dual-V_{DD} to implement per path DVS in this work. Fig. 3 shows the proposed architecture. The upper half is our power management unit (PMU), while

the bottom half is the model of a net in the interconnect. In the PMU, a low-drop voltage regulator (LDO) is used to generate V_{DDL}. A boost converter is used to generate multiple V_{DDC} values. A delay chain based logic block is used to adjust the value of V_{DDC} as needed. As the figure shows, we applied dual-V_{DD} scheme by attaching V_{DDH} or V_{DDL} to different nets during the configuration phase, while adjusting V_{DDC} based on the dynamic speed requirement.

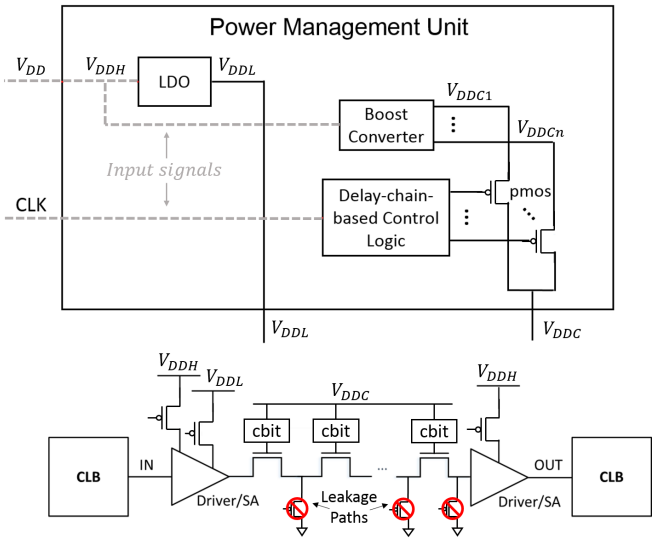

Fig. 3. The concept diagram of the proposed architecture and the power management unit

D. Voltage controller

Due to the varying speed requirement of applications, we sometimes need to dynamically adjust the supply voltages of FPGAs. To achieve this, a proper voltage controller is needed. In the existing works, people use switch-capacitor-based voltage regulators to control voltages according to the detected chip delay found by the commercial IBM Critical Path Monitor [8] and the Intel Droop Detector [9]. However, those designs are large, complicated, and time-consuming in the design process. In this work, we propose a simpler voltage controller based on the idea of the Logic Delay Measurement Circuit (LDMC) discussed in [7]. As shown in Fig. 3, the proposed controller takes a clock signal as the only input. The frequency of this clock dynamically determines the value of V_{DDC} voltage applied to the FPGA interconnect. In other word, by changing the frequency of this clock signal, the interconnect delay can be adjusted during runtime. The higher the frequency, the higher V_{DDC} value will be used. In this work, we assume this signal comes from the outside of the FPGA through a pad. Thus, the delay of the interconnect cannot be adjusted automatically, but can be controlled from the outside of the FPGA.

The detailed circuits of the proposed controller is shown in Fig. 4. The key functionality of this circuit includes understanding the desired interconnect delay indicated by the frequency of the input clock signal and assigning the corresponding V_{DDC} value for the interconnect. The proposed design uses a group of delay chains connected in serial to achieve this. Each delay chain is made by a chain of buffers with a flip-flop (FF) at the output of each buffer. The clock signal of the controller is connected to both the input of the left most delay chain and the clock ports of the FFs. When the falling edge of the clock comes, the clock signal has half cycle of time to propagate though the delay chain before the rising edge comes and triggers the FFs. Thus, only a portion of the FFs close to the input clock have outputs of 0, while the rest stay at 1. Based on the pattern made of 0s and 1s of the FF outputs, an OR-gate-based logic tree determines the V_{DDC} value applied to the FPGA interconnect by turning on the corresponding power-switch. Each power-switch will be turned on only when the output pattern of the delay chain directly controls this power-switch is "00...00" (all 0s) and the output pattern of the delay chain next to it on the right is not all 0s. This guarantees selecting the right V_{DDC} value while avoiding turning on multiple power-switches at the same time. In addition, since the value of V_{DDC} is typically higher than the main voltage V_{DD}, we use level converters at the input of the headers to avoid the short circuit current.

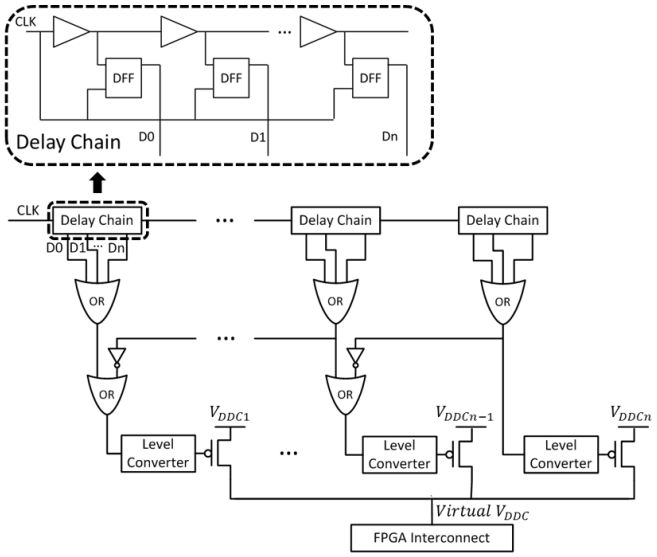

Fig. 4. The concept diagram of the delay chain circuit and the proposed delay detector and voltage controller architecture

E. Level conversion

Applying dual-V_{DD} to the traditional buffer based interconnects requires level converters at the input of the CLBs to minimize the short circuit current of converting V_{DDL} back to V_{DDH}. In the low-swing interconnect, however, the SAs naturally perform as the level converters. As discussed in [3], a SA is a modified Schmitt trigger that has a low 0-to-1 transition threshold. So no additional level converters are needed. However, the dual-V_{DD} scheme still introduce additional energy in the SAs. According to our simulation results, this part of energy overhead is about 5% of the voltage regulator overhead. Since we consider the voltage regulator

energy overhead when estimating the total energy reduction when using dual-V_{DD}, the overhead from SAs is relatively small and can be ignored.

F. Power-Gating idle switch boxes

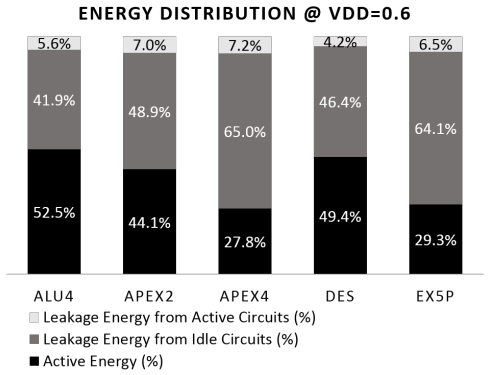

Fig. 5. Energy breakdown of the low-swing FPGA interconnect implementing MCNC benchmarks at 0.6V

Power-Gating is another widely used technique to save the energy of FPGAs, especially the leakage energy in idle circuits. Researchers traditionally use this technique to reduce the leakage energy of the buffers in SBs [5,6]. Because the low-swing interconnect has no buffers, this part of energy is naturally zero [3,4]. However, as leakage energy becomes the dominant part when the supply voltage is scaled down to near/sub-V_T, the leakage in the configuration bitcells becomes significant [12]. The bitcells are 5T SRAM cells used in FPGAs to turn on or off the switches in the interconnect or to store the look-up-table values in the CLBs. The interconnect energy breakdown of using FPGAs with the proposed interconnect architecture to implement the MCNC benchmarks is shown in Fig. 5. About 50% of the total FPGA energy at 0.6V is contributed by the leakage energy in idle circuits. In the idle circuit leakage energy, the major portion is contributed by the configuration bitcells, although the bitcell leakage is not directly shown in the figure for making the figure more clear and readable. Unfortunately, no existing works have looked into leakage reduction of the bitcells in near/sub-V_T. In this work, we explored both the coarse-grained and the fine-grained power-gating for the bitcells in the SBs. For the coarse-grained power-gating, we assign one high-V_T header for each SB. All bitcells in a single SB can only be power-gated or not at the same time. As a result, the leakage energy of the SBs is reduced by up to 100X after using power-gating according to the circuit level simulation results using SPICE. We estimated the area of the SBs in a custom layout of a low-swing SB with 84 tracks. The area overhead of the header is only less than 5% of the total area of the SB. Compared to the coarse-grained power-gating, the fine-grained scheme allows us to power-gate each bitcell separately. However, it introduces about 14.0% area overhead to the SBs.

III. Methodology

A. Low-swing interconnect modeling and simulation

In this paper, we give the definition of "net" to any signal paths start from an output of a CLB to an input of another CLB. Each net includes one or more SBs. Besides, we give the definition of "path" to any signal paths from a FPGA input pad to an output pad. Each path involves multiple SBs and CLBs. All the delay and energy numbers of the benchmarks are estimated using a self-developed tool with the simulation results of nets using SPICE. Therefore, the first step is to build a circuit model for the nets of the interconnect, and simulate the delay and the energy of each net in SPICE. Our low-swing interconnect model is shown in Fig. 3. Each interconnect segment between two CLBs is modeled as a pass-gate chain with a SA used as both drivers and receivers at the two ends. Each pass-gate indicates one SB. V_{DDH} & V_{DDL} are used to implement the dual-V_{DD} scheme, while V_{DDC} is used to perform dynamic delay and energy adjustment. The length of the pass-gate chain, the optimal size of the transistors, and the path distribution are already studied in [4]. In this research, we borrowed these optimal parameters.

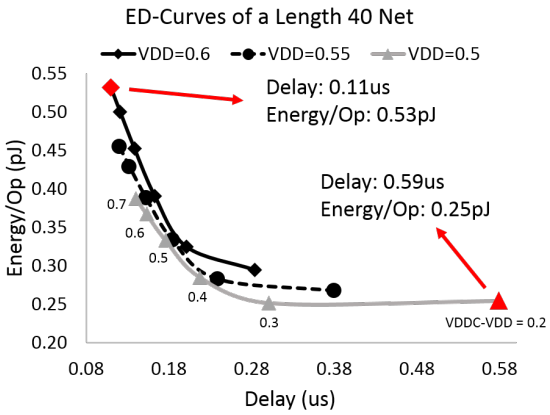

Fig. 6. The ED-Curves of a length 40 low-swing interconnect net at different supply voltages

In this work, we did all the simulations in 130nm CMOS technology. In Fig. 6, we show the delay-energy curve of a length 40 net from running SPICE. By sweeping V_{DD} and V_{DDC}, we can adjust the delay of this net from 0.11μs to 0.59μs or adjust its energy per operation from 0.25pJ to 0.53pJ. On each curve, the V_{DDC} value is swept from 0.1V higher than V_{DD} to 0.6V higher than V_{DD} from right to left. We also did the same simulation for nets with different lengths. These results indicate potential large room for adjusting the delay and energy of the FPGA interconnect dynamically when implementing benchmarks, which will be discussed soon.

B. The custom dual-V_{DD} assignment tool

By assigning the nets on the non-critical paths to V_{DDL}, the overall energy of the FPGA can be reduced. There are two important knobs for dual-V_{DD} assignment: the portions of the

nets assigned to and the values of V_{DDH} and V_{DDL}. We need to find the best combination of the V_{DDH} and V_{DDL} values that can minimize the interconnect energy without increase the critical path delay. In this work, we created a custom timing analysis tool to automatically do this optimization.

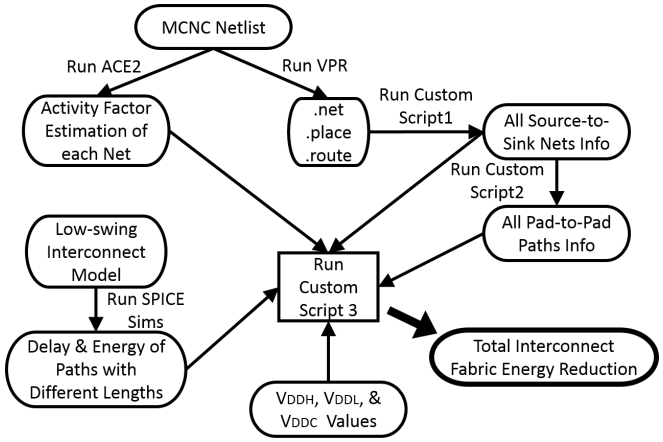

Fig. 7. The flow chart of the custom dual-V_{DD} assignment tool flow

Our tool is based on VPR [10], which can do complete timing analysis for many FPGA architectures. VPR estimates interconnect timing based on constant delay values of circuit components (switches, buffers, and wires) described in its architecture files. However, it assumes the circuit components of the same type always have same delay. This is not true for an architecture using DVS. For example, VPR assumes a switch in net #1 and another switch in net #2 have same delay. However, when applying V_{DDH} to net #1 and V_{DDL} to net #2, the delay of the two switches aren't the same. In this case, VPR timing analysis no longer accurate. To solve this problem, we can keep using VPR by creating a model for each net. This makes the architecture files extremely complicated. Also, since the nets assigned to V_{DDH} and V_{DDL} vary among applications, this method requires us to make a specific architecture file for each benchmark. For this reason, instead of using the entire VPR flow, we extracted the routing info (the length of each net, the start point and end point of each net, and how the nets build up paths) from VPR output files, then calculated the delay of each path by adding up the delay of each net on the path. Since our interconnect circuit and operating voltage are very different from the assumptions of VPR, we obtained the delay of nets by running SPICE simulation. Comparing to the timing analysis results of VPR, our tool found the same critical paths. The only difference is the absolute delay values. Since our tool allows us to recalculate the delay of every path in the dual-V_{DD} assignment process, we can keep trying to assign different voltages to each net until the FPGA archives the lowest energy point without changing the critical path. This is what VPR cannot do.

The details of our custom tool flow is shown in Fig. 7. In order to do dual-V_{DD} assignment as while as timing analysis for a benchmark, we need the routing info from running VPR, the simulated delay and power of each net from running SPICE, the activity factor of each net by running ACE 2.0 [11], and a script (the custom script 3 in Fig. 7) to do dual-V_{DD} assignment, critical path delay calculation, and energy saving calculation. In this script, we use a brute-force algorithm to initially assign all the nets to V_{DDH} and try to reassign V_{DDL} to each net. If the critical path does not change after the assignment, we keep that net on V_{DDL}, otherwise assign it back to V_{DDH}. The script exports the info of the portions of the nets assigned to V_{DDH} and V_{DDL}, the energy before and after using the dual-V_{DD} scheme, and the energy distribution of the FPGA interconnect at every V_{DDH} and V_{DDL} value combinations. We then estimated the energy overhead of using the voltage regulator based on the exported info. To use the routing info from VPR, we also created two additional scripts to parse the text-based ".net", ".place", and ".route" files generated by VPR. The custom script 1 in Fig. 7 creates a dictionary to store the detailed info of each net, while the script 2 creates that of each path. For the activity factors, we use 0.2 for all FPGA inputs. The activity factors of internal nets are then automatically generated by running ACE 2.0.

C. Delay detector and controller algorithm

Desired Critical Path Delay (us)	Equivalent Frequency of the Input Clock (MHz)	Selected V_{DDC} (V)
0.44 ~	~ 4.5	0.8
0.33 ~ 0.44	4.5 ~ 6.0	0.9
0.28 ~ 0.33	6.0 ~ 7.1	1.0
0.25 ~ 0.28	7.1 ~ 8.0	1.1
0.23 ~ 0.25	8.0 ~ 8.7	1.2
0.21 ~ 0.23	8.7 ~ 9.2	1.3

Fig. 8. Mapping of the desired critical path delay (of MCNC benchmarks) and the corresponding V_{DDC} values required at 0.6V

The architecture and functionality of the voltage controller have already been described in Section II. The frequency of the input clock signal determines which V_{DDC} to be selected. However, we haven't discussed what exact value of the clock frequency is required to turn on each power-switch. In Fig. 8, we provide these details. The leftmost column represents the expected critical path delay of the FPGA interconnect we want to adjust to. The column in the middle suggests the input clock frequency required to achieve the expected critical path delay. The column on the right indicates the corresponding V_{DDC} values applied to the interconnect. We selected the range of the desired critical path delays in the figure based on the simulated max and min interconnect delay when the FPGA implements MCNC benchmarks. The area overhead of the voltage controller is about 55 gates, 20 flip-flops, and 20 SAs, which is less than 1% of the area of the interconnect. On the other hand, the energy overhead of the controller is 0.15pJ at 0.6V. This energy overhead is less than 1% of the total FPGA interconnect energy. In addition, the time needed to switch between different V_{DDC} is less than 1.5 cycles.

IV. Results & Analysis

A. Energy savings from using dual-V_{DD}

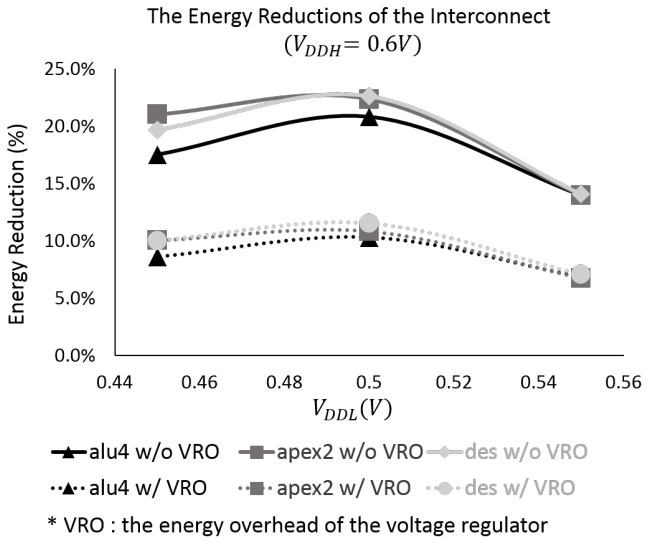

Fig. 9. The energy reductions of the low-swing interconnect implementing the MCNC benchmarks at 0.6V after using the dual-V_{DD} technique alone

Benchmark	LUT Count	FF Count	I/O Count
alu4	1522	Combinational	22
apex2	1878	Combinational	41
apex4	1262	Combinational	28
des	1591	Combinational	501
ex5p	1064	Combinational	71

Fig. 10. Benchmark Characteristics

We swept the V_{DDH} value from 0.45V to 0.6V, and V_{DDL} from 0.15V lower than V_{DDH} to V_{DDH} to find the optimal values of the supply voltages that allows the proposed interconnect architecture to reach the minimum energy point. We found that the higher V_{DDH}, the more energy savings we can get from using dual-V_{DD}. Fig. 9 shows the overall energy reduction of using the proposed FPGA interconnect implementing three of the largest MCNC benchmarks at different values of the supply voltages. In the figure, "VRO" represents the energy overhead of the voltage regulators. The curves in solid lines do not include the voltage regulator overhead, while the rest of the curves do. In this research, we use LDO to estimate the voltage regulator overhead. The LDO energy overhead approximately equals to the difference between V_{DDH} value and V_{DDL} value divided by the V_{DDH} value. As a result, when V_{DDH} equals to 0.6V, the maximum energy saving is obtained when V_{DDL} equals to 0.5V, which is 0.1V lower than V_{DDH}. We draw the similar conclusion at the other V_{DDH} and V_{DDC} values. When not considering the LDO, we archived an average energy saving of 20.1% for the MCNC benchmarks. When considering the LDO overhead, the average energy saving drops to 10.1%. In this work, we used an FPGA architecture with 4-input LUTs, 8-LUT CLBs, and SBs at each intersection of horizontal and vertical channels. The benchmark characteristics are shown in Fig. 10.

B. Energy savings from using power-gating

In Fig. 11, we show the overall energy reduction of the interconnect implementing the five of the largest MCNC benchmarks after using dual-V_{DD} and power-gating together. For each benchmark, the left bar shows the energy distribution of the low-swing interconnect without using dual-V_{DD} and power-gating (marked as "1" in the figure). The middle bar shows the energy distribution with dual-V_{DD} and coarse-grained power-gating (marked as "2"), while the right bar shows the energy distribution with dual-V_{DD} and fine-grained power-gating (marked as "3"). In the last sub-section, we discussed the energy reductions of using dual-V_{DD} only. However, only the active circuits are considered in the last sub-section. If we consider the entire FPGA including all of the idle circuits, the overall energy reduction of using dual-V_{DD} drops to about 5%. On the other hand, using the coarse-grained power-gating saves the energy of the idle circuits and the full FPGA by 27.0% and 19.0% on average, respectively. If using fine-grained power-gating, these energy savings can be further increased to 91.3% and 53.1%. Since the low-swing interconnect naturally has no buffers in the SBs, this part of the energy reduction is an improvement to the existing power-gating works. We fabricated an 8x8 FPGA in 130nm CMOS with the low-swing interconnect and coarse-grained power-gating in 130nm technology. The initial measurement results show that the leakage energy in the SBs can be reduced by 91.1% at 0.6V by applying power-gating.

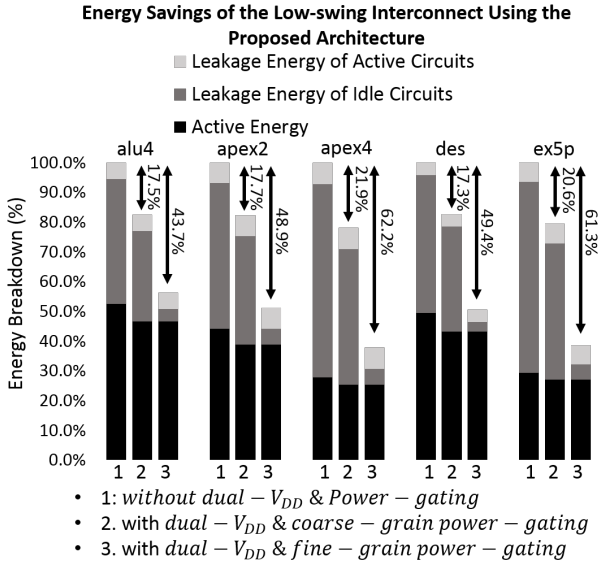

Fig. 11. The energy distribution and reduction of the low-swing interconnect implementing the MCNC benchmarks at 0.6V after using both dual-V_{DD} and power-gating techniques

C. Speed and energy adjustment by using DVS

Finally, we simulated the maximum and minimum delay and energy of the interconnect we can achieve by using DVS. To do so, we swept V_{DDC} from 0.2V higher than V_{DDH} to 0.7V higher than V_{DDH} when the FPGA interconnect is implementing the five of the largest MCNC benchmarks. In Fig. 12, we show an example of the apex2 benchmark. When implementing this benchmark, our DVS architecture allows us to adjust the delay of the low-swing FPGA interconnect from 0.22μs to 0.43μs or adjust its energy per operation from 21.9pJ to 35.7pJ at 0.6V. When implementing the five of the largest MCNC benchmarks, we can adjust the delay from 0.14μs to 0.43μs or adjust the energy per operation from 5.5pJ to 35.7pJ. When V_{DDC} is less than 0.2V higher than V_{DDH}, the swing of the signals in the interconnect will reduce to a level that cannot be detected by the SAs. This leads to potential functionality failures of the FPGAs. This situation can be avoided by inserting repeaters that are abbreviated as "R" in the Fig. 12. The repeaters are level converters in the interconnect for regenerating full swing signals. However, when reducing V_{DDC} from 0.2V higher than V_{DDH} to V_{DDH}, both the delay and energy of the interconnect increase. The long tail shown in the figure where V_{DDC} = V_{DD} or V_{DD} + 0.1V indicates that. Thus, although the repeaters guarantee functionality in some cases [4], inserting repeaters for increasing the adjustable ranges of the delay and energy is unnecessary. This conclusion also valid when the interconnect is implementing the other MCNC benchmarks.

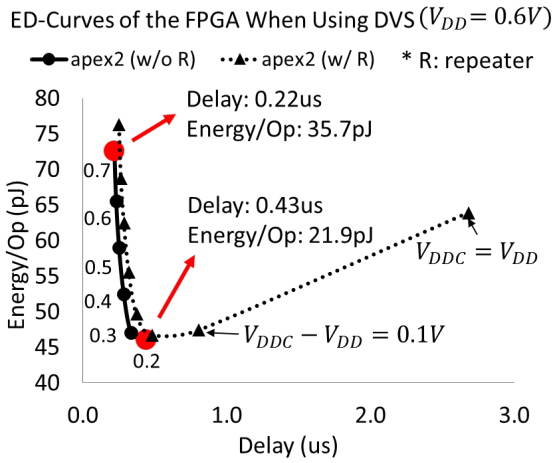

Fig. 12. The ED-Curves of the low-swing interconnect implementing the apex2 benchmark at 0.6V when using per path DVS

D. The impacts of noise and cross-talk

Since we minimized the switch box area in physical layout, the space between two interconnect wires is close to the minimum space defined in the design rule. The large parasitic caps (about 21fF between wires and 5fF between wires and the substrate) then make our circuit suffering from cross-talk problem. In the interconnect model we used to do all the simulations in this work, we included the parasitic caps between the adjacent wires to consider worst case cross-talk. In Fig. 13, we show the comparisons of critical path delay and energy reduction of our design (implementing alu4) with considering cross-talk and without. Besides, the impacts of V_{DD} and V_{DDC} noise are also shown as percentage of change in delay and energy per 10mV of voltage noise. The positive signs in the figure are used when the delay decreases and the energy increases as the supply voltages increase.

	Worst Case Crosstalk	No Crosstalk
Critical Path Delay (us)	0.23	0.14
Energy Reduction of the Full FPGA when Using Dual-VDD (%)	9.8	11.0
Sensitivity of Critical Path Delay to VDDH & VDDL Noise (%/10mV)	+ 2.1	+ 3.1
Sensitivity of Full FPGA Energy to VDDH & VDDL Noise (%/10mV)	- 3.2	+ 0.4
Sensitivity of Critical Path Delay to VDDC Noise (%/10mV)	+ 1.3	+ 0.9
Sensitivity of Full FPGA Energy to VDDC Noise (%/10mV)	+ 0.9	+ 0.7

Fig. 13. The impacts of cross-talk at V_{DDH} = 0.6V, V_{DDL} = 0.5V, V_{DDC} = 0.9V (alu4)

E. Overall energy saving comparing to previous work

Specs	[6]	[7]	[5]	This work
VDDH/VDDL (V)	1.1/0.9	1.8/1.26 ~ 1.57	1.3/0.8 ~ 1.0	0.6/0.45 ~ 0.6
Interconnect type	Uni-directional	Uni-directional	Bi-directional	Unidirectional Low-swing
Relative interconnect energy at the same VDD and technology node (x)	1	1.47	1.39	0.64 ~ 0.86
The adjustable speed range by using DVS (MHz)	Not support DVS	Not provided	Not support DVS	2.3 ~ 7.1
The adjustable energy range by using DVS (pJ/Op)	Not support DVS	Not provided	Not support DVS	5.5 ~ 35.7

Fig. 14. The comparisons of the key specifications of this work and the existing works

Compared to the existing works on the energy reduction of the FPGA interconnects, there are three main unique contributions of this work. Firstly, this is the first work of applying on-chip per path DVS to the FPGA interconnects in near/sub-V_T. We take advantage of a low-swing design that has no the area overhead of the headers in the SBs of the traditional interconnects, and use a novel method to adjust the delay and energy of the interconnect dynamically. Secondly, besides the power-gating technique is widely used in the existing designs for reducing the leakage energy of the idle drivers and buffers, we also apply power-gating to the configuration bitcells in the SBs, because it is a dominant energy consumer at low voltages. Finally, this work is the first one to combine dual-V_{DD}, DVS, and power-gating techniques all together with

considering voltage regulator overhead in depth on circuit level.

The detailed comparisons of this work and the existing works are shown in Fig. 14. In the row of "relative interconnect energy at the same V_{DD} and technology node", we set the energy of the interconnect using the design in [6] to 1 as a base line, and normalized the energy of the interconnects using the designs in [7], [5], and this work. For a fair comparison, we also scaled the existing works to the same supply voltage and technology node used in this work. The results show that the energy of the FPGA interconnect using the proposed architecture consumes 14.0% lower energy per operation than the best design in the existing works. In this comparison, we assume that the existing works considered the voltage regulator overhead when estimating the overall energy reduction. Otherwise, our work saves 36.0% more energy per operation than the existing works. Furthermore, we can also adjust the delay and energy of the interconnects dynamically, something that existing works are unable to do.

V. Limitations & Future Work

In this work, we developed a custom tool flow to assign dual-V_{DD} to every nets. However, the algorithm we used is brute-force, which is slow when the benchmark is complicated. We will explore existing mapping algorithms we could adopt to improve the speed of our tool. Furthermore, we haven't developed a tool flow for generating the configuration bitstream for the fine-grained power gates. In addition, although we already optimized the layout of a low-swing SB with coarse-grained power-gating, we haven't optimized the more complicated layout of SBs with fine-grained power-gating and find the accurate area overhead of implementing fine-grained power-gating. Moreover, the voltage controller is designed based MCNC benchmarks. If using other benchmarks, the length of the delay chain in Fig. 4 need to be adjusted. Finally, we plan to complete the measurement of our 8x8 FPGA in 130nm CMOS implementing benchmarks when our tool support is ready.

VI. Conclusion

In this paper, we proposed a novel near/sub-V_T low-swing FPGA interconnect architecture that uses dual-V_{DD} to implement per path DVS. While DVS with dual-V_{DD} is not widely applied to the traditional FPGA interconnects because of the potential high area overhead of adding headers to the buffers in the SBs, we applied this technique to a low-swing interconnect that naturally removes all buffers. Besides the power-gating technique is widely used in the existing designs for reducing the leakage energy of the idle drivers and buffers, we also apply power-gating to the configuration bitcells in the switch boxes, because it is a dominant energy consumer in near/sub-V_T. This effort leads to more leakage energy reduction in the SBs. Including the energy overhead of voltage regulators, our work archives an 10.1% energy saving in the active circuits, 27.0% - 91.3% in the idle circuits, and 19.0% - 53.1% in the entire FPGA on average. Benefits from using DVS, we can adjust the delay of the low-swing FPGA interconnect from 0.14μs to 0.43μs or adjust its energy per operation from 5.5pJ to 35.7pJ when implementing the MCNC benchmarks at 0.6V. Compared to the existing works scaled to the same supply voltage and technology node, our design saves 14.0% - 36.0% more energy. Our measurement results of an 8x8 FPGA in 130nm CMOS indicate a 91.1% leakage energy reduction at 0.6V by applying coarse-grained power-gating on the SBs.

VII. Acknowledgment

This work was sponsored in part by Boeing and by DARPA ISI program under agreement [HR0011-13-2-0004]. The content, views and conclusions presented in this document do not necessarily reflect the position or the policy of Boeing, DARPA, or the U.S. Government, no official endorsement should be inferred.

References

[1] Klinefelter, A., N. Roberts, Y. Shakhsheer, P. Gonzalez, A. Shrivastava, A. Roy, K. Craig et al. "21.3 A 6.45W self-powered IoT SoC with integrated energy-harvesting power management and ULP asymmetric radios." In Solid-State Circuits Conference-(ISSCC), 2015 IEEE International, pp. 1-3. IEEE, 2015.

[2] Elbirt, A. J., W. Yip, B. Chetwynd, and C. Paar. "An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 9, no. 4 (2001): 545-557.

[3] Ryan, J. F., and B. Calhoun. "A sub-threshold FPGA with low-swing dual-VDD interconnect in 90nm CMOS." In CICC, pp. 1-4. 2010.

[4] Qi, H., O. Ayorinde, Y. Huang, and B. Calhoun. "Optimizing energy efficient low-swing interconnect for sub-threshold FPGAs." In Field Programmable Logic and Applications (FPL), 2015 25th International Conference on, pp. 1-4. IEEE, 2015.

[5] Li, F., Y. Lin, and L. He. "Vdd programmability to reduce FPGA interconnect power." In Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design, pp. 760-765. IEEE Computer Society, 2004.

[6] Gayasen, A., K. Lee, N. Vijaykrishnan, M. Kandemir, M. Irwin, and T. Tuan. "A dual-vdd low power fpga architecture." In Field Programmable Logic and Application, pp. 145-157. Springer Berlin Heidelberg, 2004.

[7] Chow, C. T., L. Tsui, P. Leong, W. Luk, and S. Wilton. "Dynamic voltage scaling for commercial FPGAs." In Field-Programmable Technology, 2005. Proceedings. 2005 IEEE International Conference on, pp. 173-180. IEEE, 2005.

[8] Drake, A., R. Senger, H. Deogun, G. Carpenter, S. Ghiasi, T. Nguyen, N. James, M. Floyd, and V. Pokala. "A distributed critical-path timing monitor for a 65nm high-performance microprocessor." In 2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers, pp. 398-399. IEEE, 2007.

[9] Muhtaroglu, A., G. Taylor, and T. Rahal-Arabi. "On-die droop detector for analog sensing of power supply noise." IEEE Journal of solid-state circuits 39, no. 4 (2004): 651-660.

[10] Betz, V., and J. Rose. "VPR: A new packing, placement and routing tool for FPGA research." In Field-Programmable Logic and Applications, pp. 213-222. Springer Berlin Heidelberg, 1997.

[11] Lamoureux, J., and S. Wilton. "Activity estimation for field-programmable gate arrays." In Field Programmable Logic and Applications, 2006. FPL'06. International Conference on, pp. 1-8. IEEE, 2006.

[12] Rabaey, J. M., A. Chandrakasan, and B. Nikolic. Digital integrated circuits. Vol. 2. Englewood Cliffs: Prentice hall, 2002.

[13] Putic, M., L. Di, B. Calhoun, and J. Lach. "Panoptic DVS: A fine-grained dynamic voltage scaling framework for energy scalable CMOS design." In Computer Design, 2009. ICCD 2009. IEEE International Conference on, pp. 491-497. IEEE, 2009.