PERA: Power Efficient Routing Architecture for SRAM-based FPGAs in Dark Silicon Era

Zeinab Seifoori, Behzad Omidi, Hossein Asadi *Senior Member, IEEE*

***Abstract*—The ever-increasing rate of static power consumption in nanoscale technologies, and consequently, the breakdown of Dennard scaling acts as a *power wall* for further device scaling. With intensified power density, designers are forced to selectively power off portions of chip area, known as *dark silicon*. With significant power consumption of routing resources in *Field-Programmable Gate Array* (FPGAs) and their low utilization rate, power gating of unused routing resources can be used to reduce the overall device power consumption. While power gating has taken great attention, previous studies neglect major factors that affect the effectiveness of power gating, e.g., *routing architecture*, *topology*, and *technology*. In this paper, we propose a *Power Efficient Routing Architecture* (PERA) for SRAM-based FPGAs, which is designed pursuant to the utilization pattern of routing resources with different topologies. PERA is applicable to different granularity from multiplexer to *Switch-Matrix* level. We examine the efficiency of the proposed architecture with different topologies, structures, and parameters of routing resources. We further propose a routing algorithm to reduce the scattered use of resources, and hence to take advantage of opportunities of power gating in routing resources. Our experiments using VPR toolset on FPGA architecture similar to commercial chips over an extensive set of circuits from MCNC, IWLS, VTR, and Titan benchmarks indicate that PERA reduces the static power consumption by 43.3%. This improvement is obtained at the expense of 7.4% area overhead. PERA along with the optimized routing algorithm offers a total routing leakage power reduction of up to 64.9% as compared to non-power gating architectures and 6.9% in comparison with the conventional routing algorithm across all benchmark circuits and architectures with various wire segment lengths. This is while the optimized routing algorithm degrades performance by *only* less than 3%.**

I. Introduction

The widespread usage of *Field-Programmable Gate Arrays* (FPGAs) in a diverse range of applications from embedded systems to parallel high-performance computing [1], [2] is due to their shorter time-to-market, reduced *Non-Recurring Engineering* (NRE) costs, and design flexibility as compared to *Application-Specific Integrated Circuits* (ASICs). The flexibility of FPGAs to meet different requirements of various applications comes at the cost of an abundance of logic and routing resources, which leads to about $7\times - 14\times$ higher power consumption as compared with ASICs [3]. Such power gap and absence of power management capabilities in FPGAs during standby, unlike in microprocessors and *Digital Signal Processors* (DSPs), has significantly limited the use of FPGAs in applications with limited power budget [4].

Static power consumption, which is dissipated independent of switching activity contributes to a significant portion of the ASIC-FPGA power gap. With a) transistor downscaling, b) the advent of nanoscale technologies, and c) the breakdown of Dennard scaling [5], the growth of static power consumption is much faster than dynamic power. It is predicted that static power consumption will increase by 5× in every generation, which leads to the formation of a *power wall* in upcoming technologies [6], [7]. Static power as a dominant contributor to the total power is dissipated in two major resources of FPGAs, *interconnect* and *logic blocks*. Utilization statistics of FPGA resources reveal that interconnect resources are the major part of unused resources [8]. Compared to logic blocks, interconnect resources consume 1.5-2× more static power [9], [10]. Due to the low utilization rate of interconnect resources and their high static power, efficient power management of interconnect is of paramount importance.

Different approaches in the device-, circuit-, and system level, including variable transistor gate length, use of triple-oxide transistors, and multiple Vth employed by Xilinx [11], reconfigurable hard logic [1], [12]–[20], circuit level amendment of resources [21]–[23], dual Vdd/Vth [24], [25], exploiting heterogeneous routing resources [26], using power-aware *Computer-Aided Design* (CAD) algorithms [27], [28], and power gating [29]–[36] techniques have been proposed to reduce the FPGAs power consumption. Previous studies carried out on optimizing FPGA power consumption mainly focus on reducing static power consumption through power gating. Alleviating static power consumption through coarse- or fine-grained power gating, which can be applied in logic and/or routing resources, establishes the skeleton of these studies [29], [30], [32]–[35], [37], [38]. Power gating domains can be controlled during configuration time (statically) or runtime (dynamically). Static power gating techniques reduce the static power consumption by cutting off unused resources, while dynamic power gating techniques achieve this by temporarily cutting off resources during their idle times. Since dynamic power gating techniques manage the power state of the regions through power controlling signals, if they are employed in a fine-grained manner, the huge number of power controlling signals and their routing do become major challenges. In addition, employing coarse-grained dynamic power gating is inefficient due to lower power gating opportunities, controlling *rush current*[1] challenge, modifying the CAD algorithm, and associated overheads including wake-up energy wasted in power state transitions and power controller energy.

In the face of the challenges of dynamic power gating, employing a static power gating technique with well-suited granularity can improve the power efficiency of FPGAs. A finer granularity increases power gating opportunities and provides more controllability, while at the same time incurring more delay and area overhead. Accordingly, due to the trade-

All authors are with the Department of Computer Engineering, Sharif University of Technology, Tehran, Iran. E-mails: zeinab.seifoori@sharif.edu, omidi@ce.sharif.edu, asadi@sharif.edu

[1] Current required to recharge floating nodes when a power-gated region is active.

off between the power-saving obtained by the power gating technique and its imposed overheads, and the effect of resource utilization rate on this trade-off, comprehensive profiling of resource utilization is crucial to select an appropriate granularity. Moreover, the investigation of different granularities in FPGAs with various routing architectures is essential because: a) Utilization rate and pattern of multiplexers are affected by *Switch Matrix* (SM) topology, hence an appropriate granularity should be selected accordingly. b) Since the number of multiplexers and their utilization pattern is affected by routing parameters such as wire length, the efficiency of different power gating granularities can be affected in the same way. As we show in this work, employing the same power gating granularity for different routing architectures reduces the efficiency of power gating. To our knowledge, *none* of the previous studies have examined the granularity of power gating for routing architectures.

In this paper, we present a *Power Efficient Routing Architecture* (PERA), which aims to mitigate the substantial static power consumption of unused resources in the routing fabric of SRAM- based FPGAs through an improved static power gating technique. In the proposed architecture, we first investigate the utilization rates and patterns of various routing architectures. Then we propose various power gating architectures with different granularities, which are called *SB,4*, *SB,2*, *SB,1*, *SB,4,1*, and *SB,4,2*. By investigating the efficiency of various power gating architectures in detail, an appropriate granularity is selected taking into account different SM topologies (i.e., *Wilton, Subset, Universal*) and different routing architectures (i.e., the length of wire segments). Our study reveals that a significant percentage (more than 50%) of multiplexers, which are the major power consumer of routing resources are unused; hence, we can take advantage of power gating to reduce the static power consumption of routing resources. Furthermore, to find the most efficient architecture for SMs, the power consumption of different granularities for various SM topologies and routing architectures is estimated and compared based on the experimental results. Our results indicate that an appropriate power gating granularity is completely different depending on SM topology and routing architecture, hence, the most power and area efficient architecture is selected considering the SM topology and routing architecture. In PERA, we use an SRAM cell, called PG-SRAM, to control the power gating transistor for each power gating region. We also modify the routing algorithm used in *Versatile Place and Route* (VPR) tool to increase the number of resources that can be power gated.

We evaluate PERA through HSPICE and VPR [39], [40] simulations by using a comprehensive set of different benchmark suites including MCNC, IWLS, VTR, and 19 benchmarks of the Titan suite [41], which cover a wide range of applications, in terms of power-saving and incurred overheads (e.g., area and delay). We use COFFE [42] to exploit the accurate circuit model and transistor sizing of the FPGA device. PERA is implemented on both minimum-size FPGAs (FPGA devices with the minimum array size and minimum channel width, which determines the number of routing tracks in each channel, reported by VPR) and commercial-scale FPGAs. The results show that PERA reduces the static power consumption of minimum-size FPGA (with minimum channel width) by 43.3%, while imposing 7.4% area overhead. In addition, the proposed power gating aware routing algorithm further increases the power gating opportunity by up to 33.9% and hence reduces the static power consumption by up to 16.9% as compared to the conventional routing algorithm. In summary, **our experimental results reveal the following observations**.

(a) Different SM topologies lead to significantly different power consumption and power gating efficiencies. *Wilton* has the minimum power consumption while the power consumption of *Subset* topology is the most. The most power efficient architecture reduces the static power consumption of FPGA architecture composed of routing network with *Wilton* topology by 1.12× as compared to the FPGA architecture with *Subset* topology.

(b) Evaluation of PERA on FPGA devices with various wire segment lengths reveals that the most power-efficient granularity for routing networks with wire segments of 1, 2, 3, 4, 6, 8, and 16 is *SB,4,1*, *SB,4,2*, *SB,4,2*, *SB,2*, *SB,2*, *SB,2* and *SB,4*, respectively.

(c) Power gating has a different impact on each routing architecture and SM topology. For example, the efficiency of the proposed power gating architecture in FPGAs with *Subset* topology is the least. This is due to their diverse utilization patterns and rate of resources.

This paper offers the following novel contributions:

(1) We present comprehensive profiling of routing resource utilization for various routing architectures with different parameters (e.g., wire segment length). We also investigate the impact of the routing architectures on FPGA performance and power consumption.

(2) Using the proposed profiling study, we present a novel routing architecture, called PERA, to reduce the static power consumption of routing resources through power gating, which can be applied with different granularities. We then investigate the imposed overheads and study the area, delay, and power tradeoff to achieve an effective comprise of area, delay, and power. We also offer a detailed analysis of the improvement or deterioration of PERA over FPGAs with different wire segment lengths compared with the baseline FPGA.

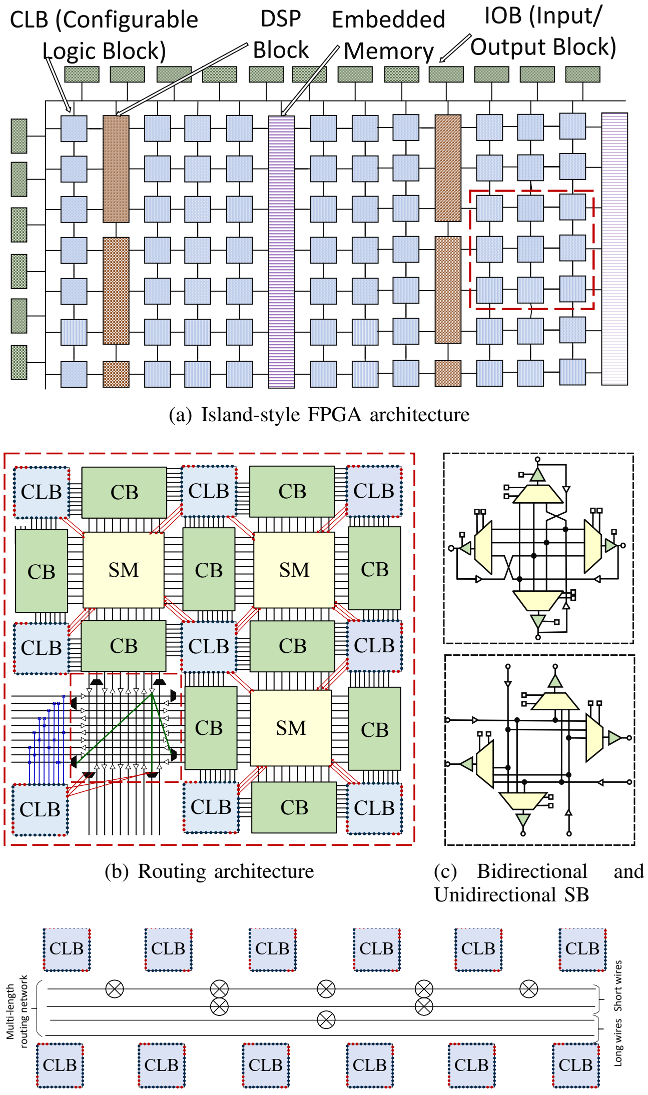
(3) We optimize the existing VPR routing algorithm to further increase the power gating opportunities and hereby the power saving of the proposed architecture. We decrease the performance degradation by leaving the delay-sensitive of cost function intact.

II. FPGA Architecture

Island-style and hierarchical are two different categorizations of FPGA architectures, each of which has its pros and cons. Hierarchical FPGA architecture offers higher routing speed and lower scalability as compared with island-style architecture. However, in this paper, our focus is on FPGAs with island-style architecture, which is preferred by most commercial vendors and academic research. A set of *Configurable Logic Blocks* (CLBs), RAM blocks, and embedded DSP slices, which are arranged in dedicated columns and surrounded by a pool of routing resources constitutes an island-style architecture (Fig. 1 (a)). CLBs, which are responsible for implementing the logic of mapped circuits, consist of several *Basic Logic Elements* (BLEs), each of which comprises a *Look-Up Table* (LUT), a *Flip-Flop* (FF), and a 2:1 multiplexer to optionally select the output of LUT or FF. The connections between CLBs are made through the routing network. The routing resources include *Connection Blocks* (CBs), *Switch Matrices* (SMs), routing tracks, and intra-cluster multiplexers. An output net of a CLB a) enters the routing network via SM, b) routes using routing SMs, and c) enters the destination CLB using CB (Fig. 1 (b)). CBs provide connectivity between routing tracks and CLBs inputs while the connectivity between routing tracks and CLBs outputs are provided through SMs.

Routing among CLBs is done using SM multiplexers that are placed at the intersection of horizontal and vertical channels. Each SM can be assumed as a set of *Switch Boxes* (SBs), each of which consists of a set of multiplexers, their associated buffers, and controlling SRAM configuration cells that select an appropriate multiplexer input [9]. The two most common SB structures are *unidirectional* and *bidirectional* as depicted in Fig. 1 (c). Unidirectional routing tracks afford better area-delay products than bidirectional tracks; hereby, our proposed architecture employs unidirectional routing tracks as in commercial devices [43]. Different arrangements of SBs within SMs lead to different routability and topologies. More routability in FPGA SMs means that FPGA with smaller channel width can be used to route the design nets. Topology determines which outgoing tracks can be connected to each incoming track of SM. The most widely-used SM topologies are *Subset*, *Wilton*, and *Universal* [44]–[46]. Table I provides different architectural parameters.

The routing tracks are divided into wire segments with a certain length. While in some FPGA architectures, all routing tracks are composed of wire segments with the same length, in most commercial FPGAs the routing channel is divided into sub-channels such that the tracks of each sub-channel consist of wire segments with a particular length. Multiple-length wire segments span some CLBs before connecting to the next SM multiplexer. By investigating the architecture of one of the most advanced devices (Stratix-IV device family), which is available as part of the Verilog-to-routing tool (VTR 8.0) [40], [47], it is revealed that its routing architecture is composed of horizontal and vertical routing channels of heterogeneous wire segments of various lengths (i.e., wire segments of 4 and 12 and wire segments of 4 and 20 logic blocks long in vertical and horizontal channel, respectively) [47], [48], as illustrated in Fig. 1 (d). Long wires (i.e., 12 and 20) can be only derived by short wires (i.e., 4), and short wires are accessible by logic blocks. Given that the routing channels are composed of wire segments with heterogeneous lengths, the SM multiplexers, which drive wire segments with various lengths, are in different sizes. Accordingly, the routing network of Stratix-IV devices contains multiplexers with two sizes small (12:1) and large (40:1), which derive short and long routing wires, respectively. In our study, similar to contemporary commercial devices, a two-level pass gate multiplexer structure is used due to its better area-delay efficiency as compared with conventional tree-based structures [49].


CLB (Configurable Logic Block)
DSP Block
Embedded Memory
IOB (Input/ Output Block)
CLB
CB
SM
Multi-length routing network
Short wires
Long wires

(a) Island-style FPGA architecture

(b) Routing architecture

(c) Bidirectional and Unidirectional SB

(d) Routing architecture composed of wire segments with different lengths and SBs that connect horizontal and vertical wire segments.

Fig. 1. FPGA architecture

TABLE I
ARCHITECTURAL PARAMETERS

Parameter	Definition
K	LUT size
N	Number of BLEs in each CLB
L	Track segment length
F_s	Number of branches of an input track in an SB
X_{loc}	Size of intra-cluster multiplexers
W	Number of tracks per channel (Channel Width)
I	Number of inputs of each CLB
F_{cin}	CB connectivity factor (specifies the number of tracks connected to each CB multiplexer)
F_{cout}	SB connectivity factor (specifies the number of SBs that each CLB output is connect to)

III. PREVIOUS WORK

Previous studies on reducing the routing static power in island-style FPGAs can be broadly classified into two main categories: a) using non-power gating techniques and b) using power gating techniques. The non-power gating techniques, which attempt to cope with significant static power consumption through device-level low-power techniques such as dual-vdd, dual-threshold, or ad-hoc techniques are *orthogonal* to our proposed architecture and can be used jointly to further reduce the static power consumption.

Since our focus in this paper is on employing power gating techniques to reduce static power consumption, we just review the most relevant proposed architectures to ours in this section. These studies aim to benefit from either fine- or coarse-grained power-gating techniques that can be implemented either statically (i.e., in the configuration time) or dynamically (i.e., during the runtime). In this regard, Bsoul et al. presented two techniques, a) one that aims to control the power state of individual logic blocks and their adjacent routing channels such as track isolation buffers and CBs by dynamically turning configuration bits "ON" and "OFF", and b) the other that controls the power of FPGA resources in a coarse-grained manner [37]. To route power control signals and preserve the flexibility of the routing network, SMs are "*always ON*" and are not designed to be power gated. Considering the dominant contribution of SM multiplexers to static power consumption (up to 50%) [9], [50], our focus in this paper is on reducing the static power consumption of SM multiplexers.

Various studies in the scope of reducing routing static power consumption attempt to reduce the static power consumption in SMs through power gating [34], [35], [51]. Some of these studies try to use power gating in a coarse-grained region, which may include one or more SMs. For instance, Bharadwaj et al. offered a *Power State Controller* (PSC) according to the data flow graph of an application [32], which extracts the idle periods of an application as an opportunity for power gating of FPGA resources. Due to the high rate of unused SM Multiplexers, there are tremendous opportunities to reduce the static power consumption in SMs. This is while clustering SMs along with other resources can waste these opportunities.

Few studies try to reduce the routing static power consumption through dynamic power gating. For example, Bsoul et al. proposed an architecture for reducing the static power consumption in SMs along with logic blocks [34]. This work divides the chip area into the same granularity of power-gating regions composed of logic blocks and their corresponding CBs. The power state of SMs is controlled by adding one multiplexer and SRAM configuration cells. Despite the abundance of power gating opportunities in partially used SMs and a large number of partially used SMs, partially used SMs have been neglected.

Furthermore, studies that employ dynamic power-gating techniques, try to modify CAD algorithms to provide more power-gating opportunities, which impairs the refinements done in the normal procedure of CAD tool development. In the technique proposed by Gayasen et al., a *Region-Constraint Placement* (RCP) algorithm is used to prevent scattering design placement [30]. The foundation of coarse-grained architecture, which has been suggested by Li et al. [33] is *Power Control Hard Macro* (PCHM). PHCM functionality includes clock-gating as well as power-gating of idle regions, which are composed of logic blocks and CBs. The cost function of the placement algorithm is modified in order to increase the probability of unused power gating regions. Another important challenge in employing dynamic power gating is to identify idle periods of a module, especially in an interactive and input-dependent application in which the behavior of the application/modules cannot be predetermined.

In the scope of static power-gating, Hoo et al. [35] proposed a coarse-grained power-gating method for unidirectional SMs. In this method, several buffers on each side of a SM are grouped as a power gating region and their supply voltage is controlled by a PMOS transistor. Besides, the VPR routing algorithm is modified to increase the power gating opportunity of low-utilized regions. The effectiveness of this method is tightly bounded to directional Wilton SMs in which the same used track number is rarely used in other directions. Coarse-grained power gating restricts the opportunity for power gating due to sporadically used resources within each coarse region, forcing it to remain powered on.

Yazdanshenas et al. [51] proposed a static fine- grained power-gating architecture to cope with the static power consumption of logic and routing resources in FPGAs. With the high percentage of unused or partially-used LUTs and their multiplexer-based structure, 6-LUTs are subdivided into a composition of two 5- or 4-LUTs, where unused sub-LUTs are power gated to save static power consumption. This architecture focuses on reducing the power consumption of SRAM cells. In addition, the target SM and CB architectures are based on pass transistors, which are no longer used in commercial FPGAs. We compare the power saving of our proposed power-gating architecture with the previously proposed architectures in [34], [35], [51] in Section V.

IV. Proposed Architecture: PERA

An efficient solution to cope with static power is power gating the inactive regions of the circuit. Power gating, if implemented efficiently, is a generally accepted method to alleviate the static power consumption in both ASICs and FPGAs [4], [52]. One of the major issues for efficient implementation of power gating is determining the suitable granularity. The low utilization rate of resources along with the centralized distribution of used resources is crucial to justify the use of power gating in a circuit.

Here, we first examine different baseline routing architectures (i.e., SRAM-based architecture with no power gating) from the power consumption, performance, and resource utilization rate perspectives in Sec. IV-A. Afterward, we represent various power gating schemes with different granularities and argue about their efficiency in bringing the opportunity to turn off resources in Sec IV-B. Finally, in Sec. IV-C, we elaborate the routing algorithm to increase the number of unused power gating regions and optimize the static power consumption.

A. Analysis of Different Routing Architectures

In this section, we examine how different routing architectures affect performance, power consumption, and the trade-off between performance and power consumption in the baseline FPGA architecture (i.e., SRAM-based architecture with no power gating facility). FPGA power consumption and performance vary with different routing architecture parameters such as wire segment length. That is, increasing the length

TABLE II
MINIMUM CHANNEL WIDTH AND NUMBER OF ROUTING MULTIPLEXERS (AVERAGED OVER ALL 25 SELECTED MAPPED ON FPGA WITH VARIOUS WIRE SEGMENT LENGTHS).

Wire segment length	L=1	L=2	L=3	L=4	L=6	L=8	L=16
Channel width	148.72	151.36	163.44	172.48	201.6	231.04	368.56
Number of multiplexers	1040057	521308	385768	313749	258642	237267	232132
Routing static power consumption (mW)	6.27	4.2	3.84	3.72	3.69	4.39	6.01
Routing delay (nS)	17.1	12.2	11.5	12.5	15.9	18.2	49.3

TABLE III
RATE OF UNUSED MUXES ON FPGA WITH MINIMUM SIZE AND MINIMUM CHANNEL WIDTH.

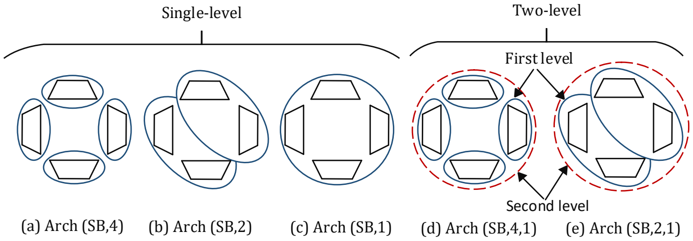
Benchmarks	L=1 (%)	L=2 (%)	L=3 (%)	L=4 (%)	L=6 (%)	L=8 (%)	L=16 (%)
mcml	64,2	57,4	57,2	52,3	51,5	49,5	48,5
LU32PEEng	59,6	54,8	54,7	50,9	48,7	47,0	43,5
stereovision2	60,1	56,0	61,0	68,9	74,8	80,9	87,5
vga_lcd	62,8	57,9	56,8	52,8	52,1	51,0	53,6
bgm	58,9	54,7	50,8	50,7	49,4	46,5	48,0
LU8PEEng	58,3	52,4	50,6	48,2	47,2	46,7	47,8
ethernet	57,6	52,6	50,5	49,4	47,7	48,4	46,8
mkDelayWorker32B	80,9	78,3	77,6	77,0	76,2	74,4	77,0
stereovision1	60,2	53,8	53,6	50,6	52,9	62,2	78,5
stereovision0	62,4	57,0	53,6	51,9	53,3	51,3	47,3
blob_merge	56,5	52,3	51,8	49,0	47,9	47,6	51,8
pci_bridge32	60,8	57,3	52,9	51,3	46,6	50,8	59,7
or1200	65,8	60,4	59,4	54,0	56,0	54,7	58,5
mem_ctrl	58,1	51,1	52,5	47,5	49,9	50,0	59,7
ex1010	47,9	44,0	45,0	42,9	44,4	50,8	54,4
usb_funct	57,7	52,6	49,1	46,7	48,3	51,7	61,7
clma	55,1	45,6	48,2	42,8	45,1	47,5	56,4
aes_core	54,1	56,8	53,1	53,5	56,3	57,0	59,8
pdc	58,7	47,5	45,8	45,5	48,8	48,7	58,8
boundtop	57,8	52,4	52,9	47,9	52,0	52,7	62,6
ac97_ctrl	60,3	57,6	54,5	50,3	50,8	52,2	60,1
mkSMAdapter4B	58,8	55,3	50,5	54,4	55,6	58,3	63,9
raygentop	57,5	50,1	52,8	45,9	55,8	65,7	79,9
systemcaes	50,9	47,2	49,5	49,0	49,3	53,4	62,2
s38417	55,7	46,4	50,0	49,7	51,8	53,5	58,5
Average	**59.2**	**54.1**	**53.4**	**51.3**	**52.5**	**54.1**	**59.5**

of the wire segment decreases the number of multiplexers that each track spans and increases the channel width (W) of FPGA. To investigate the effect of various architectural parameters on the performance and power consumption of FPGA, we carry out a set of experiments over a selection of the 25 largest MCNC, IWLS, and VTR benchmark suites. The architectural parameters used in these experiments are similar to the commercial devices [53], [54] and are summarized in Table V (i.e., the architectural parameters are described in Table I). To evaluate the efficiency of the proposed architecture in practical applications, we conduct the experiments under two different conditions: mapping circuits on FPGAs with a) the minimum size and minimum channel width specified by VPR, and b) the commercial size, which is set based on the smallest FPGA from the Xilinx Virtex-6 FPGA family and should be large enough to implement the benchmark. We also set the channel width to 320 in commercial-size FPGAs.

Table II lists the average results of minimum channel width, number of multiplexers, static power consumption, and routing delay average across all 25 selected benchmarks. The channel width is set to 1.2X of the minimum channel width to provide the required flexibility to route different circuits. By increasing the wire segment length, the minimum channel width typically increases in most circuits. The average number of multiplexers across all designs with different routing architectures is also shown in Table II. Unlike channel width, which grows as the length of wire segments increases, in the majority of circuits, the number of multiplexers in the routing network decreases as wire segment length increases. It is because that the number of multiplexers in each track with longer wire segment length decreases by $1/L$ (i.e., L denotes the length) while the channel width does not increase by L.

As the wire segment length (L) increases from 1 to 6 (Table II), the static power consumption of the majority of circuits decreases. Although elongating the length of wire segments increases the channel width and the size of multiplexers, the static power saving of the reduced number of multiplexers overcomes, and hence the total static power consumption of the routing network is reduced. In addition, although the number of multiplexers in routing networks containing longer wire segments is less, the static power consumption of SM multiplexers containing longer wire segments is higher.

Increasing the length of the wire segment up to 3 decreases the routing network delay (as presented in Table II). This is due to the fact that the number of SM multiplexers in the routing networks containing longer wires is reduced. On the other hand, longer wires cause higher switching delay, hence, if the delay reduction caused by decreasing the number of SM multiplexers overcomes the increase of switching delay, the delay of the routing network reduces. As the wire segment length increases from 4 to 8, though the trend of routing network delay is increasing, the delay is still less than the delay of routing networks comprising wire segments of length one. The increased delay caused by the higher switching delay of long wire segments overwhelms the delay reduction due to the fewer number of multiplexers in the routing network containing wire segments with the length of 16 and accordingly, the routing network delay increases significantly.



(a) Arch (SB,4) (b) Arch (SB,2) (c) Arch (SB,1) (d) Arch (SB,4,1) (e) Arch (SB,2,1)

Fig. 2. Different granularities for SBs

Previous studies reported that a significant fraction of static power is consumed by the interconnect multiplexers (about 60%) [9]. In this section, we examine the rate of unused multiplexers in FPGAs with diverse topologies of SMs and with various wire segment lengths. The percentage of unused multiplexers in FPGAs with different segment lengths is reported in Table III. According to this table, in the most of benchmarks, by increasing the wire segment length from $L = 1$ to $L = 4$, the rate of unused MUXes decreases, and then by increasing the wire segment length from $L = 6$ to $L = 16$, the rate of unused MUXes increases. As expected, the percentage of unused multiplexers on FPGAs with the commercial size and the minimum size is more than 80% and 50%, respectively.

B. Power Gating Architectures with Different Granularities

High power dissipation in routing resources besides their low utilization rate implies that power gating is an efficient method to reduce static power consumption. Fig. 2 shows various possible power gating schemes with different granularities for SBs including SB-level and intra-SB levels. The circles indicate the power gating regions wherein the power consumption is controlled by two cut-off transistors and one SRAM configuration bit (i.e., PG-SRAM). Each power gating

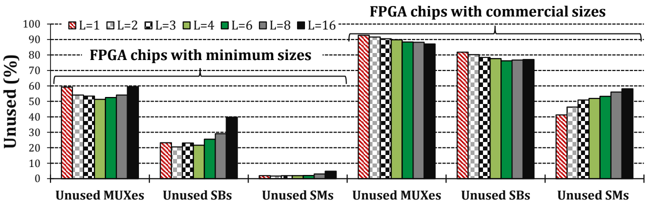


Fig. 3. Rate of unused resources on FPGA chips with minimum sizes and commercial sizes

region includes a few multiplexers inside SMs. Therefore, all routing SMs of FPGAs has the same power-gating regions and the same structure, and hence the proposed power-gating architectures do not conflict with the hardware regularity in FPGAs. The cut-off transistors connect the supply voltage of power-gated resources to the main supply voltage to support two operation modes of ON and OFF. Two cut-off PMOS and NMOS transistors are inserted between VDD and supply voltage and between GND and supply voltage, respectively. Therefore, the outputs of the unused power-gated multiplexers are switched to the high-impedance state, which does not cause any functional failure, since there is no net being routed through unused routing multiplexers. In addition, the connection between the outputs of routing multiplexers and LUT inputs is made by connection blocks, each of which includes one multiplexer with one latch at the output. So, if the routing signals are in high-impedance condition, the connection block latches keep the previous state.

The proposed scheme in Fig. 2(a) is the most fine-grained power gating architecture in which the power supply of each multiplexer is controlled by a unique controller. Due to the relatively identical *ON* and *OFF* states of adjacent multiplexers, the scheme illustrated in Fig. 2(b) with one controller for every two adjacent multiplexers is proposed. Among the proposed power gating schemes, Fig. 2(c) architecture is the most coarse-grained scheme where the power of each SB is controlled individually. Fig. 2(d) and Fig. 2(e) illustrate two-level power-gating architectures which comprise the advantages of both fine- and coarse-grained power-gating schemes. In these two schemes, when all SRAM configuration bits are zero and their corresponding multiplexers are unused, all multiplexers and their corresponding SRAM cells, as well as PG-SRAM cells, are also power gated.

Here, we investigate the effect of the power gating architectures with various granularities on the power gating opportunities. As it is clear, *SB,4,1* brings an opportunity of turning off resources as many as *SB,4* does. Analogously, the number of opportunities for turning off resources in both *SB,2* and *SB,2,1* is the same. Besides, the number of resources that can be turned off in *SB,1* is equal to the number of unused SBs, which is depicted in Fig. 3. With neglecting the imposed overheads, the finer the granularity is, the more unused resources can be turned off. Nonetheless, the decisive criterion of the effectiveness of one granularity is the magnitude of the difference between the obtained gains and the imposed overheads, provided that the former is greater than the latter.

As a general estimation, Fig. 4 illustrates the average rate of power-gated multiplexers through different levels of power gating in two-level power gating architectures averaged over all circuits (i.e., the experimental setup is the same as the configuration presented in Section IV-A). The number of power-gated multiplexers applying the second level of power gating circuit in both *SB,4,1* and *SB,2,1* is equal to the rate of unused SBs (illustrated in Fig. 3), which is 46.1% of unused multiplexers across all benchmark circuits as well as all architectures with various segment lengths. By employing the two-level power gating architecture such as *SB,4,1* (or *SB,2,1*), we can take advantage of two architectures of *SB,1* and *SB,4* (or *SB,2*). Since the architecture of *SB,4,1* (*SB,2,1*) can turn off the fully unused SB as well as partially used SBs, it has more power gating opportunities than *SB,1* and at the same time, it has less power overhead than *SB,4* (or *SB,2*) architecture. The higher the possibility of turning off resources through the second level of power gating is, the more efficient the two-level power gating architecture is.

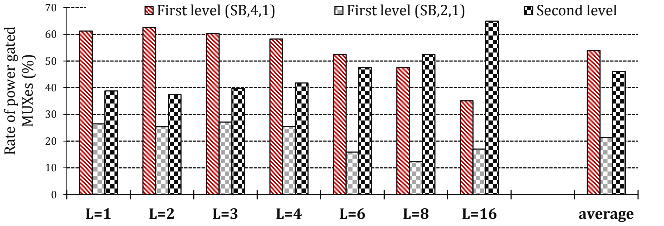


Fig. 4. Rate of power gated Muxes through different levels of power gating

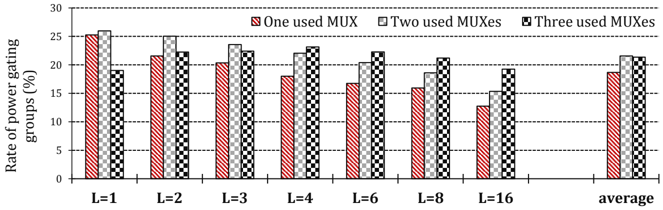


Fig. 5. Average rate of power gating regions with the different numbers of used multiplexers across all benchmark circuits as well as all routing networks with various segment lengths.

C. *Power Gating Aware Routing*

While the proposed power gating architecture with larger granularity than one multiplexer per power gating region can turn off a large number of unused multiplexers, it does not guarantee that it can turn off the maximum number of unused multiplexers. Fig. 5 illustrates the number of power gating regions with the different numbers of used multiplexers in the 25 largest benchmark circuits on FPGA with the minimum size of the chip across all routing networks with various segment lengths. Based on the results reported in this figure, there is a large number of power gating regions (18.7%, on average), which contain only *one* used multiplexer. Furthermore, the number of power gating regions with two and three used multiplexers is considerable (21.6% and 21.4%, respectively). If there is a possibility of moving the routing from the power gating regions with *one used* multiplexer to regions with *two* or *three used* multiplexers, the power gating regions with only one multiplexer can be turned off as well. An efficient solution to cope with this issue is modifying the routing policy to avoid routing through inactive power gating regions.

To this end, first, we briefly provide a concise survey of the routing algorithm of state-of-the-art FPGAs. *PathFinder*

[55] is a common routing algorithm used in conventional FPGAs. Subsequent to re-routing nets that are routed through uncongested areas, the available resources can serve to route the nets which are negotiating with other nets to get the shared resources. *PathFinder* algorithm consists of two main parts: a signal router, which routes one signal through breadth-first search, and a global router, which routes all the nets of the design by leveraging the signal router and managing the resource cost. The cost of sharing resources in the first round of the routing algorithm is free and gradually increases in subsequent rounds. The less critical nets with shared resources are forced to be routed through uncongested resources. As such, the probability of congestion decreases after each iteration.

The routing achieved at the end of each routing iteration can be somewhat illegal due to the overusing of resources. However, based on this iteration, we can do a full timing analysis to extract the net delay and slack time of each connection for improving the next routing iteration. The connections with large slack can be routed through slower resources without affecting the delay of circuits. Conversely, the connections with *zero* slack are on the critical path and any increase in their delays leads to an increase in the delay of the circuit. Equation 1 formulates the criticality of each connection from source i to destination j [56]. In this equation, D_{max} denotes the delay of the critical path of the circuit, and $slack(i, j)$ represents the slack of the source and sink j connection of net i.

$$Crit(i,j) = 1 - \frac{slack(i,j)}{D_{max}} \tag{1}$$

In each iteration, the cost of using each routing resource (represented by rr) for establishing the connection from source i to destination j is updated through Equation 2.

$$\begin{aligned} Cost(rr) &= Crit(i,j).delay(rr) \\ &+ [1 - Crit(i,j)] \, . \, [b(rr) + h(rr)] \, .p(rr) \end{aligned} \tag{2}$$

The first term of Equation 2 is based on the delay and the second term is based on the resource congestion. In this equation, $delay(rr)$ stands for the delay of each routing resource (e.g., SB or CB multiplexers). $h(rr)$ is the historical congestion of each routing resource which increases if it is overused in that routing iteration. $b(rr)$ stands for the base cost of using one routing resource which equals the delay of the routing resource (i.e., $delay(rr)$) [57]. $p(rr)$ represents the existing congestion cost, which is related to the number of signals overusing the routing resource currently as well as the number of routing iterations.

Our goal here is to modify the cost function of the routing algorithm such that it accounts for the utilization status of the power gating region where the multiplexer is located. To this end, one variable is needed for each multiplexer, which corresponds to the number of used multiplexers in the power gating region containing the aforementioned multiplexer. Due to the dependency of the critical path delay to the delay-sensitive term of the cost function, in our modified cost function, this term remains intact to prevent the modified routing policy from deteriorating the circuit performance. Hence, we augment the congestion-sensitive term of the cost function with a factor of $S \times b(rr) \times e^{-numofActiveMUXes}$, i.e., power gating cost (the second term in Equation 3) to encompass the utilization status of the power gating region.

$$\begin{aligned} Cost(rr) &= Crit(i,j).delay(rr) \\ &+ [1 - Crit(i,j)] \, . \, [[b(rr) + h(rr)] \, .p(rr) \\ &+ S \times b(rr) \times e^{-numofActiveMUXes}] \end{aligned} \tag{3}$$

In this equation, S in the updated term denotes the size of the power gating region (i.e., the number of multiplexers in the power gating region), which is considered to be *zero* for resources not located in a power gating region. This factor determines the weight of increasing the cost of utilizing resources in unused power gating regions, which increases more in utilizing larger unused power gating regions. Aimed at maximizing the utilization of each power gating region and discouraging the usage of unused power gating regions, we update the cost function with a factor of $e^{-numofActiveMUXes}$. Since the cost decreases exponentially with the number of used multiplexers in power gating groups, utilizing even one multiplexer in a power gating group decreases the cost exponentially. In addition, in our proposed cost function, the cost of using one multiplexer in a power gating group with fewer used multiplexers is more, which makes it less likely to be utilized in the subsequent routing iterations and hence, the utilization probability of the associated power gating group decreases.

V. Experimental Setup and Results

In this section, the experimental setup is detailed in Section V-A. Then, the effect of different power gating granularities, the impact of the cut-off transistor sizing, and transistor node technology are investigated in Section V-B. Next, the effect of the power gating aware routing algorithm on enhancing the power gating architecture efficiency is examined in Section V-C. Section V-D represents the effect of different SM topologies on the efficiency of different power gating granularities. Lastly, the comparison of PERA with the previous studies is reported in Section V-E.

*A. **General Setup***

The experimental results are obtained using a discriminative set of the 25 largest MCNC, IWLS, and VTR benchmarks and 19 benchmarks from the Titan suite [41]. The selected industrial-size benchmarks from the Titan suite, which cover a wide range of applications, can effectively reflect the modern designs due to their containing heterogeneous blocks (Table IV). We choose the architectural parameters and FPGA circuit topology similar to commercial devices following [53] and [54]. The architectural parameters are summarized in Table V (their definition are provided in Table I). To evaluate the proposed architectures and investigate their efficiency on commercial devices, we conduct our experiments with two different array sizes of FPGAs, which are VPR-defined minimum array size and predefined commercial sizes (i.e., which is chosen to be the size of the smallest device of Xilinx Virtex-6). In addition, we repeat our experiments for fixed-length FPGA architectures with different segment lengths of $L = 1, 2, 3, 4, 6, 8,$ and 16. To provide high performance along

TABLE IV
TITAN BENCHMARKS: NUMBER OF DSP BLOCKS, RECONFIGURABLE BLOCKS, AND RAM SLICES LISTED BY MURRAY ET AL. [41]. (SUITABLE FPGA SIZE FOR PLACE AND ROUTE EXTRACTED USING VTR 8.0.)

No.	Name	# Blocks	DSPs	RAM Slices	FPGA Size	Application
B1	cholesky_mc	108,239	452	5,123	125 × 93	Matrix Decomposition
B2	bitcoin_miner	1,061,829	0	59,968	225 × 167	SHA Hashing
B3	bitonic_mesh	192.648	676	61,616	242 × 179	Sorting
B4	cholesky_bdti	257,750	1,027	4,920	169 × 125	Matrix Decomposition
B5	denoise	343,263	192	11,827	150 × 111	Image Processing
B6	des90	109,962	352	16,256	171 × 127	Multi μP system
B7	mes_noc	548,047	0	25,728	192 × 142	On Chip Network
B8	gsm_switch	487,454	0	35,776	255 × 189	Communication Switch
B9	LU_Network	630,212	896	41,647	221 × 164	Matrix Decomposition
B10	minres	252,600	614	17,608	224 × 166	Control Systems
B11	neuron	90,779	565	3,799	129 × 96	Neural Network
B12	openCV	212,616	740	16,993	242 × 179	Computer Vision
B13	segmentation	174,072	107	5,658	136 × 101	Computer Vision
B14	SLAM_spheric	124,648	296	16,256	124 × 92	Control Systems
B15	sparcT1_chip2	814,799	24	14,355	279 × 207	Multi-core μP
B16	sparcT1_core	91,235	8	4,277	82 × 61	μP Core
B17	sparcT2_core	287,839	0	8,883	152 × 113	μP Core
B18	stap_qrd	237,193	579	9,747	158 × 117	Radar Processing
B19	stereo_vision	92,662	152	4,287	129 × 96	Image Processing

TABLE V
ARCHITECTURAL PARAMETERS USED IN COFFE

Parameter	Value	Parameter	Value
K	6	F_s	3
N	10	F_{cin}	0.2
W	320	F_{cout}	0.1
L	1, 2, 3, 4, 6, 8, 16	X_{local}	$\frac{N+I}{2} = 25$
I	40	Or	2

with high logic density in FPGA, we choose $K = 6$ (same as commercial devices [58], [59]).

We first generate the accurate Hspice netlist utilizing COFFE [42], which is an automated transistor sizing tool for FPGAs. The purpose of COFFE is to size the transistors of FPGA to meet user constraints (by precisely taking into account the parasitic effects of both transistors and wire loads). To obtain the realistic measurement, COFFE models all transistors and wires load, even the short metal, which connects two transistors in a multiplexer. COFFE is fed with a) architectural parameters, which are provided in Table V, b) circuit topology, c) *Predictive Technology Model* (PTM) models [60]. The parasitic capacitance and resistance in each technology node should be provided for COFFE simulation. To extract the parasitic capacitance and resistance of metal layers, we employ the scaling presented in [61], which reports the capacitance and resistance of metal layers with technology scaling.

Next, we run VPR from VTR 7.0 [39] and VTR 8.0 open-source toolset [40] to extract the delay, area, and resource utilization of routing resources and thereby the static power consumption of FPGA architectures with various configurations. The experimental results of the baseline architecture and the proposed architectures for the 25 largest MCNC, IWLS, and VTR benchmarks are generated using VTR 7.0. Due to the limitation of VPR 7.0 in supporting the custom switch blocks and routing networks with a mixture of various wire segment lengths, which is implemented in modern FPGAs such as the Stratix-IV device family, we leverage VPR 8.0 to examine the effect of the proposed FPGA architectures in order to implement the industrial-sized Titan benchmarks on Stratix-IV device family, which is the most advanced FPGA that is included in VTR 8.0 tool [47].

To investigate the power, area, and delay characteristics of the proposed power gating architecture in comparison with the baseline architecture (i.e., SRAM-based architecture with no power gating facility), we leverage the HSPICE simulation fed with COFFE transistor sizing. In this way, we estimate the power, area, and delay of each SM in the baseline architecture and the proposed power gating architecture in various utilization patterns. Then using the aforementioned measurements provided by HSPICE and the utilization pattern of different resources extracted from placing and routing benchmarks by VPR, the power, area, and delay characteristics of the baseline and the proposed power gating architectures are extracted. At the same time, we feed COFFE with different technology models including 16nm, 22nm, 32nm, and 45nm PTM LP [61] to investigate the effect of the proposed architectures on enhancing the power consumption and the delay of FPGA devices in various node technologies. These technology models are used for low-power applications.

B. Effect of Power Gating Granularity

Routing structures, parameters, and topologies affect the efficiency of power gating architectures, and the most efficient granularity for one routing configuration is not necessarily the most efficient one in another routing configuration. Accordingly, we should examine the efficiency of the proposed architectures in FPGAs with different routing configurations to extract the most efficient architecture for each routing configuration.

Fig. 6 illustrates the average normalized power consumption of different power gating granularities with respect to the baseline architecture (i.e., SRAM-based architecture with no power gating facility) across all benchmarks mapped on FPGAs with various wire segment lengths. As Fig. 4 depicts, the rate of power-gated multiplexers by employing the two-level power-gating architecture is increased by increasing the wire segment length. The rate of power saving achieved through two-level power-gating architectures, however, is not enhanced by increasing the wire segment length (Fig. 6). This is due to the fact that by increasing the wire segment length, the size of the cut-off transistor in the first level is increased to prevent the performance degradation. Hence, the power consumption of power gating circuit (i.e., power overhead), which includes the configuration SRAM cell and cut-off transistors, grows and overwhelms the power saved through employing the power gating architecture. For the wire segment lengths of $1 \preceq L \prec 4$, the two-level power-gating architectures (i.e., *SB,4,1* and *SB,2,1*) achieve the most power saving. Furthermore, the power gating architecture of *SB,4* is the most efficient power-gating architecture among the proposed ones for the wire segment lengths of $4 \preceq L \preceq 16$. The power consumption of the most fine-grained granularity is the sum of the power consumption of all the powered-off and all the powered-on power gating regions, each of which includes the routing resources and power gating circuitry. Furthermore, as it is clear in this figure, *SB,4* achieves about 49% power saving and provides the best power efficiency among the proposed ones. *SB,2*, *SB,1*, *SB,4,1*, and *SB,2,1* power-gating architectures improve the power efficiency of the baseline architecture by 45.7%, 31.2%, 45.3%, and 45.6%, respectively. Therefore, the average power efficiency improvement across all proposed power-gating architectures, as well as all benchmarks, is about

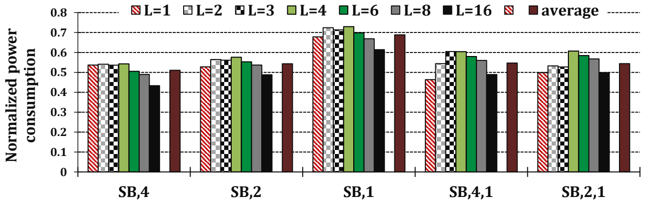


Fig. 6. Normalized power consumption of different power gating granularities with respect to the baseline across all benchmarks mapped on FPGAs with

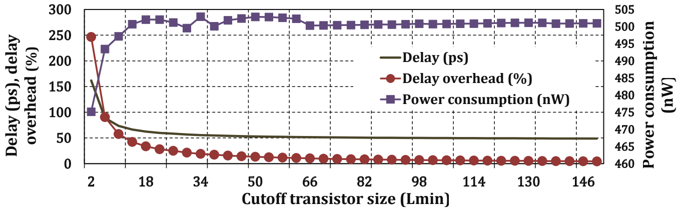


Fig. 7. Dependency between the size of cutoff transistor and delay overhead as well as power consumption.

43.3%. Additionally, the power efficiency of all proposed power gating granularities is improved by increasing the length of the routing network wire segments (for wire lengths of $4 \prec L \preceq 16$). We achieve the best power efficiency in routing networks composed of wire segments with the length of 16. This is because that the rate of unused resources in FPGA chips with minimum size increases for wire segment lengths greater than four (see Table III). Unexpectedly, the two-level power gating granularity does not achieve the best power efficiency across all segment lengths. This is due to the fact that the size of the cut-off transistors should be proportional to the size of multiplexers to avoid imposing significant delay overhead on the routing network. Accordingly, in routing networks consisting of long wire segments, the size of power gating transistors is large, which leads to more area and power overheads that overwhelm the power saving achieved using two-level granularities.

To prevent the cell supply node (e.g., routing multiplexers) from being floated during the cutoff time, a pair of nMOS and pMOS cutoff transistors are added between the cell supply node and the power rail. Since driving current flows through the aforementioned cutoff transistors when the output of multiplexers switches, the size of cutoff transistors affects the delay of multiplexers. This is while there is not any dependency between the size of cutoff SRAM cell transistors and multiplexer delay. Fig. 7 depicts the dependency between the size of pMOS cutoff transistor and delay overhead, as well as the power consumption of the used routing multiplexer equipped with power gating circuitry in routing networks, consisting of wire segments with the length of 4 in 22nm technology. As shown in this figure, the delay overhead of power gating circuitry is greatly reduced as the size of the cutoff transistor increases. Unlike the delay overhead, the size of the cutoff transistor has a negligible impact on power consumption. To compute the increased area footprint of the proposed architecture, we employ the minimum-width transistor model [53], which estimates the layout area of an NMOS pass-transistor based on relative strength (x) through Equation (4) and the area of a CMOS transistor through equation (5) [42], [53].

The area overhead of the proposed power gating architectures for FPGAs with routing networks composed of different wire segment lengths in 22nm technology is reported in Table VII. To calculate the area overhead of the proposed architecture with various granularities, we perform circuit-level Hspice simulations fed with accurate transistor sizing generated by COFFE using the 22nm PTM LP model [60]. We measure the area overhead by extracting the area of normal SMs (i.e., without power gating circuitry) and the area of SMs augmented with power gating circuitry. The area overhead is the difference between the two aforementioned areas, which is in fact the area of the cut-off circuit including cut-off sized transistors and SRAM configuration cells to control power gating regions.

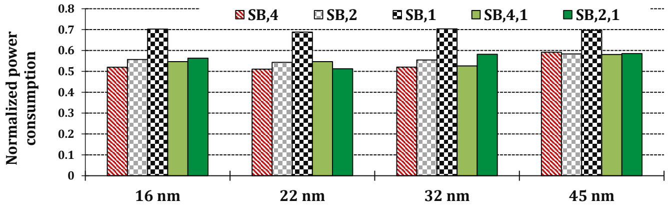


Fig. 8. Normalized power consumption of different power gating granularities with respect to the baseline across all benchmarks mapped on FPGAs and various wire segment lengths for 16nm, 22nm, 32nm, and 45nm.

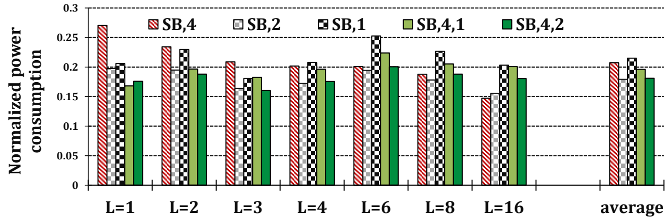


Fig. 9. Normalized power consumption of different power gating granularities with respect to the baseline across all benchmarks mapped on FPGAs with commercial size and various wire segment lengths for 22nm.

Furthermore, unlike the previous studies [62], which consider zero delays overhead through sizing the cut-off transistors, our HSPICE simulations demonstrate that sizing cut-off transistors cannot remove the delay overhead completely. By sizing the cut-off transistors based on Table VII, the imposed delay overheads for the proposed architectures are less than 8%. To alleviate the delay overhead, the size of cut-off transistors can be tuned to be very large, which may overwhelm the achieved gains.

$$Area(x) = 0.447 + 0.128x + 0.391\sqrt{x} \tag{4}$$

$$Area(x) = 0.518 + 0.127x + 0.428\sqrt{x} \tag{5}$$

Fig. 8 demonstrates the average effect of different power gating granularities on the power consumption of the routing network of FPGAs with 16nm, 22nm, 32nm, and 45nm transistor technologies across all benchmarks and routing networks with different wire segment lengths. As shown in this figure, FPGAs with different node technologies are affected differently by various power gating architectures. Similar to FPGAs with 22nm node technology, on average, *SB,4* has the best power efficiency among the proposed power gating architectures in FPGAs with 16nm and 32nm node technologies. *SB,4,1* with a negligible difference with *SB,2* and *SB,4* achieves the best power efficiency in FPGAs with 45nm node technology. Since in larger technology nodes,

TABLE VI

RATE OF POWER GATING GROUPS WITH DIFFERENT NUMBER OF USED MULTIPLEXERS EMPLOYING CONVENTIONAL AND PROPOSED ROUTING ALGORITHMS. HERE, THE RATE OF UNUSED POWER GATING GROUPS AND POWER GATING GROUPS CONTAINING ONE, TWO, AND THREE MULTIPLEXERS EMPLOYING THE CONVENTIONAL ROUTING ALGORITHM ARE COMPARED WITH THE PROPOSED ROUTING ALGORITHM.

Benchmarks	**# Unused Conventional**	**# Unused Proposed**	**# One used Conventional**	**# One used Proposed**	**# Two used Conventional**	**# Two used Proposed**	**# Three used Conventional**	**# Three used Proposed**	**# Fully used Conventional**	**# Fully used Proposed**
mcml	15.6	18.7	21.8	12.9	28.7	20.2	24.3	35.2	9.7	13.1
LU32PEEng	14.6	19.6	21.1	18.1	28.6	11.1	25.1	35.4	10.7	15.9
stereovision2	43.3	51.1	17.9	6.8	17.1	15.5	14.7	17.6	6.9	8.9
vga_lcd	18.4	22.2	21.6	19.9	25.4	11.7	22.6	31.5	11.9	14.7
bgm	15.6	18.1	20.2	18.3	27.7	15.6	25.5	33.2	10.9	14.7
LU8PEEng	14.8	20.2	17.7	8.9	26.9	15.9	27.6	38.4	12.8	16.4
ethernet	16.2	18.9	19.2	15.5	25.8	14.3	25.1	30.1	13.7	21.1
mkDelayWorker32B	56.1	62.7	17.5	1.3	11.1	15.5	10.1	12.3	5.2	8.3
stereovision1	16.3	22.6	20.2	9.3	26.8	14.1	25.2	36.4	11.6	17.6
stereovision0	18.1	22.1	20.1	16.0	26.5	13.1	24.2	34.2	11.1	14.7
blob_merge	19.1	22.7	16.8	8.4	22.8	14.9	25.2	34.6	16.0	19.4
pci_bridge32	21.9	29.6	18.5	9.4	21.6	11.9	22.4	28.4	15.6	20.6
or1200	30.1	41.6	17.0	5.7	15.4	3.2	19.4	27.2	17.9	22.3
mem_ctrl	22.8	27.1	15.8	8.8	18.1	9.9	22.9	29.1	20.3	24.9
ex1010	18.4	22.4	13.9	4.3	18.4	4.9	25.5	31.9	23.8	36.4
usb_funct	18.9	24.9	16.2	7.9	21.7	5.7	25.4	33.7	17.7	27.7
clma	17.8	24.0	14.9	6.6	18.4	4.4	24.7	34.8	24.2	30.2
aes_core	22.5	30.8	20.5	8.3	22.7	14.8	21.2	26.5	13.1	19.6
pdc	20.9	25.1	15.3	5.8	18.3	8.4	24.3	33.9	21.2	26.7
boundtop	18.4	23.2	18.8	14.8	22.9	8.0	24.4	30.7	15.4	23.2
ac97_ctrl	24.9	32.1	15.9	3.3	18.0	3.8	22.6	31.0	18.4	29.6
mkSMAdapter4B	29.1	41.3	16.4	4.9	18.9	5.2	21.8	27.2	13.7	21.3
raygentop	15.9	19.1	19.1	13.3	24.5	11.4	25.8	34.9	14.6	21.3
systemcaes	20.9	27.2	17.3	10.7	22.8	8.1	24.0	29.8	14.9	24.2
s38417	21.7	24.3	16.5	13.8	21.8	8.8	24.3	33.2	15.6	19.7
Average	**22.1**	**27.7**	**18.0**	**10.1**	**22.1**	**10.8**	**23.1**	**30.9**	**14.7**	**20.5**

TABLE VII

AREA OVERHEAD OF PROPOSED ARCHITECTURES IN 22NM TECHNOLOGY.

Arch. / Seg. length	*SB,4*	*SB,2*	*SB,1*	*SB,4,1*	SB,2,1
$L = 1$	22.4	13.5	6.7	33.0	26.9
$L = 2$	5.6	4.8	2.4	9.6	6.3
$L = 3$	6.2	5.3	2.7	10.4	8.1
$L = 4$	5.4	4.6	2.3	10.1	7.7
$L = 6$	5.1	3.9	1.9	9.2	6.9
$L = 8$	4.1	3.1	1.6	8.1	6.4
$L = 16$	3.1	2.0	1.1	5.3	4.2

TABLE VIII

EXECUTION TIME (SECOND) OF THE PROPOSED POWER-GATING AWARE ROUTING ALGORITHM VS CONVENTIONAL ROUTING ALGORITHM

Benchmarks	**Conventional routing algorithm**	**Proposed routing algorithm**	**Execution time change rate**
mcml	94710.7	97988,2	3.46
LU32PEEng	77985.2	81971.2	5.1
stereovision2	19902.5	25765.6	29.5
vga_lcd	3265.9	3678.2	12.6
bgm	2927.5	5367.2	83.3
LU8PEEng	6056.8	6956.3	14.8
ethernet	5229.7	5607.9	7.2
mkDelayWorker32B	11158.7	13078.5	17.2
stereovision1	2616.5	3377.7	29.1
stereovision0	347.7	733.6	110.9
blob_merge	600.1	2566.8	327.8
pci_bridge32	652.9	988.6	51.4
or1200	1585.1	857.5	-45.9
mem_ctrl	232.5	305.7	31.5
ex1010	1557.5	590.1	6.1
usb_funct	158.8	125.1	-21.2
clma	162.3	180.7	11.3
aes_core	97.9	102.9	5.1
pdc	133.5	154.1	15.4
boundtop	148.2	163.5	10.3
ac97_ctrl	203.7	245.6	20.6
mkSMAdapter4B	345.8	66.8	-80.7
raygentop	108.7	103.2	-5.1
systemcaes	102.8	120.45	17.2
s38417	94.4	101.9	8.1
Average	**9175.4**	**10047.9**	**26.6**

the power consumption of SRAM cells is dominant in the routing network [63], the *SB,4,1* architecture, which turns off more SRAM cells, saves more power consumption as compared with *SB,4* architecture. However, due to the large area overhead of *SB,4,1*, the architecture of *SB,2* is more affordable. *SB,1* has the least power efficiency among all power gating architectures.

To demonstrate the applicability and extensibility of the proposed architectures in commercial devices, we modify the VPR to map the circuits on FPGAs with Xilinx Virtex-6 device sizes. Fig. 9 illustrates the average normalized power consumption of the proposed architectures with respect to the baseline across all benchmarks mapped on commercial FPGAs in 22nm technology. Since the size of existing commercial devices is much larger than the required minimum size of the circuits, there is a high percentage of unused resources on the device (Fig. 3), which brings more opportunities for power saving. As depicted in this figure, employing PERA on commercial devices can decrease power consumption by 80.4% (up to 85.3%).

C. Effect of Power Gating Aware Routing Algorithm

The proposed power gating aware routing algorithm (discussed in Section IV-C) attempts to route nets through multiplexers that reside in used power gating groups to increase the number of deactivated power gating groups. Table VI shows the impact of routing algorithms on the rate of power gating groups with different numbers of used multiplexers. For the sake of brevity, only the results of the proposed routing algorithm on the routing network composed of wire segment of length 4 in 22nm technology (using *SB,1* power gating architecture) are listed in this table. As presented, the proposed power gating aware routing algorithm increases the power gating opportunity by 26.1% (up to 41.9%), as compared to the conventional routing algorithm while imposing negligible performance overhead (less than 3%). Table VIII reports the elapsed time for routing the circuits on FPGA with routing network composed of wire segment length of 4 in 22nm technology using the proposed power-gating aware routing algorithm and conventional routing algorithm. As reported in this table, the proposed power-gating aware routing algorithm and the conventional routing algorithm, on average, require 10047 and 9175 seconds to route the circuits, respectively.

Fig. 10 shows the impact of the proposed routing architec-

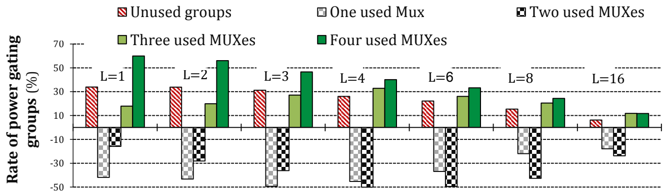


Fig. 10. Power gating groups rate using the proposed routing algorithm vs. the conventional routing algorithm.

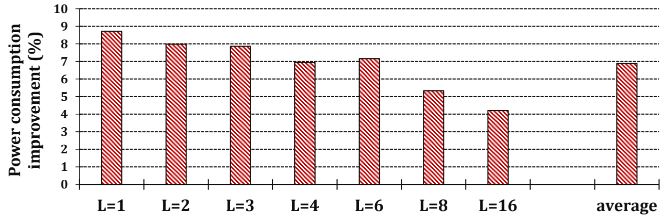


Fig. 11. Power consumption improvement in FPGA equipped with power gating architecture (*SB,1*) using proposed routing algorithm vs. FPGA equipped with power gating architecture (*SB,1*) using conventional routing algorithm.

ture on the rate of unused power gating groups as compared to the rate of power gating groups when employing the conventional routing algorithm in routing networks with different wire segment lengths across all benchmarks. By leveraging the proposed routing algorithm, the rate of power gating groups with two and three used multiplexers decreases in all routing architectures (i.e., by 36.6% and 35.1%, respectively), which is consistent with the goal of our proposed routing algorithm. In other words, we discourage routing through power-gating groups with fewer used multiplexers to increase the number of completely unused power gating groups. However, as the wire segment length converges to 16, the impact of the proposed routing algorithm decreases because the number of routing multiplexers in routing networks composed of longer wire segments is more limited and hence there is less opportunity to change routing. Nevertheless, the proposed power-aware routing algorithm increases the power gating opportunity by 24.1%. The power consumption improvement of *SB,1* in PERA in comparison with the conventional routing algorithm is depicted in Fig. 11. Analogous to Fig. 10, Fig. 11 implies that the effect of the proposed routing algorithm in routing networks composed of longer wire segments decreases. The proposed routing algorithm decreases the static power consumption of FPGAs equipped with power gating architecture of *SB,1* by 8.7% in a routing network composed of wire segment of one. Furthermore, the power consumption of *SB,1* is reduced by 6.9%, on average, using the proposed routing algorithm.

D. *Effect of SM Topology*

Due to the different utilization patterns and various utilization rates of FPGAs with different SM topologies, while implementing the same circuit, their power consumption is dissimilar and hence the power gating architectures affect them differently. Fig. 12 demonstrates the average static power consumption of FPGAs with various SM topologies of *Wilton*, *Subset*, and *Universal* and different wire segment lengths of routing network across all benchmarks in 22nm technology. The most power efficiency is achieved through using *Wilton* SM topology in routing networks of FPGAs. Furthermore, as depicted in this figure, exploiting *Universal* SM topology in the routing network of FPGA leads to more power efficiency as compared to *Subset* topology.

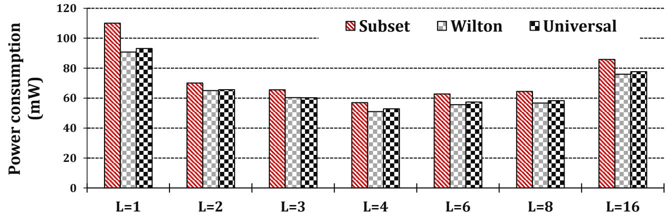


Fig. 12. Average static power consumption of FPGAs with various SM topologies across all benchmarks in 22nm technology.

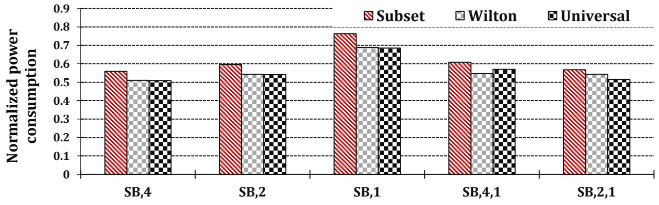


Fig. 13. Normalized power consumption of different power gating granularities in FPGAs with various SM topologies.

The efficiency of the proposed power gating architectures in FPGAs with different SM topologies across all benchmarks and routing networks with various wire segment lengths are depicted in Fig. 13. As shown in this figure, FPGA architectures with *Subset* SM topology in routing networks are less affected by the proposed power gating architectures. This could also be inferred from the feature of *Subset* SM topology, which is less routable than *Wilton* and *Universal* topologies, and hence, the utilization rate of resources in FPGAs with *Subset* SM topology is higher, which leads to fewer power gating opportunities. The efficiency of the proposed power gating architectures in FPGAs with *Wilton* and *Universal* is almost similar. *SB,4* is the most power-efficient power gating architecture among the proposed ones across all SM topologies and all wire segment lengths. The aforementioned architecture decreases the static power consumption of the routing network by up to 59.1% (53.2%) and by 49.2% (44.1%), on average, in FPGAs with *Universal* (*Subset*) SM topology.

E. *Comparison of PERA with the Previous Studies*

Here we compare PERA with previous power-gating studies including Bsoul et al. [34] and Hoo et al. [35] over Titan benchmarks. As previously described in Section III, Hoo et al. divide the routing resources into power gating regions, each of which contains an SM. This is while Bsoul et al. cluster the multiplexers and buffers on each side of SM as a power gating region. Given that the routing architecture of modern FPGAs (e.g., Stratix-IV) consists of two kinds of small and large multiplexers, we investigate two versions of the power gating architecture proposed by Hoo et al., Hoo-1, and Hoo-2. The former clusters the small and large multiplexers on each side of SM as different power gating groups and the latter clusters all multiplexers on each side of SM as a power gating group.

Fig. 14 depicts the rate of power gated multiplexers by employing PERA as compared to the previous studies [34],

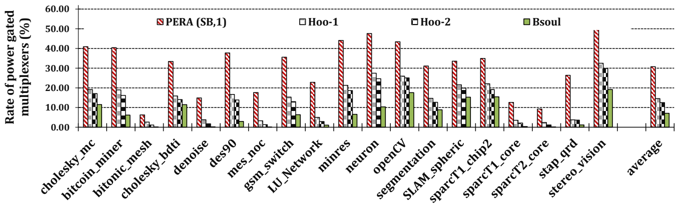


Fig. 14. Rate of power gated multiplexers by employing PERA (*SB,1*) compared with the architectures in related studies.

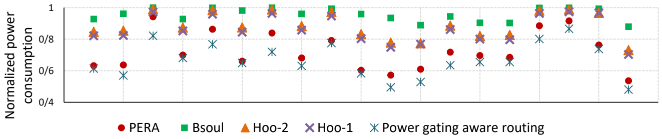


Fig. 15. Normalized power consumption of PERA (*SB,1*) as compared with the architectures presented in [34], [35]. Each marker corresponds to the static power consumption of one benchmark on an FPGA augmented with an architecture.

[35]. PERA (*SB,1*) outperforms the architectures proposed in the previous studies by 182%. PERA can switch off about 30.8% of all multiplexers. This is while the Hoo-1, Hoo-2, and Bsoul power gating architectures can switch off about 14.5%, 12.5%, and 7.1% of all multiplexers, on average, respectively. Fig. 15 illustrates the normalized power consumption of Titan benchmarks on an FPGA augmented with PERA (i.e., *SB,1*) in comparison with FPGA augmented with the power gating architectures proposed in related studies [34], [35]. As illustrated, employing the power gating aware routing algorithm increases the power gating opportunities and hence decreases the static power consumption. Each marker corresponds to the static power consumption of one benchmark employing one power gating architecture.

VI. Conclusion And Future Work

In this paper, we analyzed different routing architectures in commonly used island-style SRAM-based FPGA architectures in terms of power consumption, performance as well as resource utilization rate. By demonstrating the low utilization rate of routing resources as well as their high power consumption, we aimed to reduce the static power consumption by employing the power gating architectures. We proposed a novel power gating architecture (called PERA) with different granularities and evaluated the power-saving efficiency in various routing architectures and different node technologies. Since a large number of routing resources in used power gating groups remain unused, we modified the routing algorithm to use routing resources residing in used power gating groups as far as possible to increase the rate of power gating opportunities. Experimental results show up to 56.6% and on average, by 43.3% reduction in static power through employing PERA in minimum size FPGA with minimum channel width. Our analysis also shows that the efficiency of the proposed power gating architecture is highly dependent on the baseline FPGA architecture as well as node technology. In addition, the proposed power gating aware routing algorithm increases the power gating opportunities by up to 33.9% (24.1%, on average), which leads to up to 16.9% (6.9%, on average) reduction in static power consumption.

Since our proposed power-efficient architectures turn off unused resources in the routing network, it prevents a large amount of unwanted signal switching, and hence reduces dynamic power consumption, which will be further investigated in our future work. Furthermore, due to the high rate of unused CBs, decreasing the static power consumption of CBs is a promising future direction for this research. Lastly, considering the high power consumption of clock networks in FPGAs, investigation and mitigation of their power consumption is an interesting direction for future work.

References

[1] S. Tamimi, Z. Ebrahimi, B. Khaleghi, and H. Asadi, "An efficient sram-based reconfigurable architecture for embedded processors," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 3, 2019.

[2] S. S. Roy, K. Järvinen, J. Vliegen, F. Vercauteren, and I. Verbauwhede, "Hepcloud: An fpga-based multicore processor for fv somewhat homomorphic function evaluation," *Transactions on Computers*, vol. 67, no. 11, 2018.

[3] I. Kuon and J. Rose, "Measuring the gap between fpgas and asics," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 2, 2007.

[4] T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90nm low-power fpga for battery-powered applications," in *International symposium on Field programmable gate arrays*, 2006.

[5] R. H. Dennard, F. H. Gaensslen, L. Kuhn, and H. Yu, "Design of micron mos switching devices," *IEEE Solid-State Circuits Society Newsletter*, vol. 12, no. 1, 2007.

[6] M. B. Taylor, "A landscape of the new dark silicon design regime," *IEEE Micro*, vol. 33, no. 5, 2013.

[7] L. Wang and K. Skadron, "Implications of the power wall: Dim cores and reconfigurable logic," *IEEE Micro*, vol. 33, no. 5, 2013.

[8] A. Rahman, S. Das, T. Tuan, and S. Trimberger, "Determination of power gating granularity for fpga fabric," in *Custom Integrated Circuits Conference*, 2006.

[9] T. Tuan and B. Lai, "Leakage power analysis of a 90nm fpga," in *Custom Integrated Circuits Conference*, 2003.

[10] V. Degalahal and T. Tuan, "Methodology for high level estimation of fpga power consumption," in *Asia and South Pacific Design Automation Conference*, 2005.

[11] M. Klein, "Power consumption at 40 and 45 nm," *White Paper*, vol. 298, 2009.

[12] I. Ahmadpour, B. Khaleghi, and H. Asadi, "An efficient reconfigurable architecture by characterizing most frequent logic functions," in *Proceedings of 25th International Conference on Field Programmable Logic and Applications (FPL)*, 2015.

[13] J. H. Anderson and Q. Wang, "Area-efficient fpga logic elements: Architecture and synthesis," in *Asia and South Pacific Design Automation Conference*, 2011.

[14] P. A. Jamieson and J. Rose, "Enhancing the area efficiency of fpgas with hard circuits using shadow clusters," *Transactions on very large scale integration (VLSI) systems*, vol. 18, no. 12, 2010.

[15] H. Parandeh-Afshar, H. Benbihi, D. Novo, and P. Ienne, "Rethinking fpgas: elude the flexibility excess of luts with and-inverter cones," in *International symposium on Field Programmable Gate Arrays*, 2012.

[16] Y. Okamoto, Y. Ichinomiya, M. Amagasaki, M. Iida, and T. Sueyoshi, "Cogre: A configuration memory reduced reconfigurable logic cell architecture for area minimization," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2010.

[17] Y. Hu, S. Das, S. Trimberger, and L. He, "Design and synthesis of programmable logic block with mixed lut and macrogate," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 4, 2009.

[18] Z. Ebrahimi, B. Khaleghi, and H. Asadi, "Peaf: A power-efficient architecture for sram-based fpgas using reconfigurable hard logic design in dark silicon era," *IEEE Transactions on Computers*, vol. 66, no. 6, 2017.

[19] A. Ahari, B. Khaleghi, Z. Ebrahimi, H. Asadi, and M. B. Tahoori, "Towards dark silicon era in fpgas using complementary hard logic design," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2014.

[20] A. Wagle and S. Vrudhula, "Heterogeneous fpga architecture using threshold logic gates for improved area, power, and performance," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2021.

[21] J. H. Anderson and F. N. Najm, "Active leakage power optimization for fpgas," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 3, 2006.

[22] S. Srinivasan, A. Gayasen, N. Vijaykrishnan, and T. Tuan, "Leakage control in fpga routing fabric," in *Asia and South Pacific Design Automation Conference*, 2005.

[23] M. Hasan, A. K. Kureshi, and T. Arslan, "Leakage reduction in fpga routing multiplexers," in *International Symposium on Circuits and Systems (ISCAS)*, 2009.

[24] J. H. Anderson and F. N. Najm, "Low-power programmable fpga routing circuitry," *Transactions on very large scale integration (VLSI) systems*, vol. 17, no. 8, 2009.

[25] B.-L. Tan, K.-M. Mok, J.-J. Chang, W.-K. Lee, and S. O. Hwang, "Risc32-lp: Low-power fpga-based iot sensor nodes with energy reduction program analyzer," *Internet of Things Journal*, 2021.

[26] A. Rahman, S. Das, T. Tuan, and A. Rahut, "Heterogeneous routing architecture for low-power fpga fabric," in *Proceedings of the Custom Integrated Circuits Conference*, 2005.

[27] J. Lamoureux and S. J. Wilton, "On the interaction between power-aware fpga cad algorithms," in *International conference on Computer-aided design*. IEEE Computer Society, 2003.

[28] K. Herath, A. Prakash, S. A. Fahmy, and T. Srikanthan, "Power-efficient mapping of large applications on modern heterogeneous fpgas," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2020.

[29] Y. Lin, F. Li, and L. He, "Routing track duplication with fine-grained power-gating for fpga interconnect power reduction," in *Asia and South Pacific design automation conference*, 2005.

[30] A. Gayasen, Y. Tsai, N. Vijaykrishnan, M. Kandemir, M. J. Irwin, and T. Tuan, "Reducing leakage energy in fpgas using region-constrained placement," in *International Symposium on Field Programmable Gate Array*, 2004.

[31] Z. Seifoori, H. Asadi, and M. Stojilović, "A machine learning approach for power gating the fpga routing network," in *2019 IEEE International Conference on Field-Programmable Technology (ICFPT)*, 2019, pp. 10–18.

[32] R. P. Bharadwaj, R. Konar, P. T. Balsara, and D. Bhatia, "Exploiting temporal idleness to reduce leakage power in programmable architectures," in *Asia and South Pacific Design Automation Conference*, 2005.

[33] C. Li, Y. Dong, and T. Watanabe, "New power-aware placement for region-based fpga architecture combined with dynamic power gating by pchm," in *International Symposium on Low-power Electronics and Design*, 2011.

[34] A. A. Bsoul and S. J. Wilton, "An fpga with power-gated switch blocks," in *International Conference on Field-Programmable Technology (FPT)*, 2012.

[35] C. H. Hoo, Y. Ha, and A. Kumar, "A directional coarse-grained power gated fpga switch box and power gating aware routing algorithm," in *International Conference on Field Programmable Logic and Applications (FPL)*, 2013.

[36] Z. Seifoori, H. Asadi, and M. Stojilović, "Shrinking fpga static power via machine learning-based power gating and enhanced routing," *IEEE Access*, vol. 9, pp. 115 599–115 619, 2021.

[37] A. A. Bsoul and S. J. Wilton, "An fpga architecture supporting dynamically controlled power gating," in *International Conference on Field-Programmable Technology (FPT)*, 2010.

[38] Z. Seifoori, Z. Ebrahimi, B. Khaleghi, and H. Asadi, "Introduction to emerging sram-based fpga architectures in dark silicon era," in *Advances in Computers*, 2018, vol. 110.

[39] J. Luu, J. Goeders, M. Wainberg, A. Somerville, T. Yu, K. Nasartschuk, M. Nasr, S. Wang, T. Liu, N. Ahmed *et al.*, "Vtr 7.0: Next generation architecture and cad system for fpgas," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 7, no. 2, 2014.

[40] K. E. Murray, O. Petelin, S. Zhong, J. M. Wang, M. Eldafrawy, J.-P. Legault, E. Sha, A. G. Graham, J. Wu, M. J. Walker *et al.*, "VTR 8: High-performance cad and customizable FPGA architecture modelling," *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, vol. 13, no. 2, pp. 1–55, May 2020.

[41] K. E. Murray, S. Whitty, S. Liu, J. Luu, and V. Betz, "Titan: Enabling large and complex benchmarks in academic CAD," in *International Conference on Field programmable Logic and Applications*, Porto, Portugal, Oct. 2013.

[42] C. Chiasson and V. Betz, "Coffe: Fully-automated transistor sizing for fpgas," in *International Conference on Field-Programmable Technology (FPT)*, 2013.

[43] G. Lemieux, E. Lee, M. Tom, and A. Yu, "Directional and single-driver wires in fpga interconnect," in *International Conference on Field-Programmable Technology*, 2004.

[44] G. G. Lemieux and S. D. Brown, "A detailed routing algorithm for allocating wire segments in field-programmable gate arrays," in *ACM/SIGDA Physical Design Workshop, Lake Arrowhead, CA*, 1993.

[45] S. J. E. Wilton, J. Rose, and Z. Vranesic, "Architectures and algorithms for field-programmable gate arrays with embedded memory," *University of Toronto, Toronto, Ont., Canada*, 1997.

[46] Y.-W. Chang, D. Wong, and C.-K. Wong, "Universal switch modules for fpga design," *Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 1, no. 1, 1996.

[47] "StratixIV device handbook." Handbook, Altera, January, 2016.

[48] (2009 (accessed November 1, 2020)) Quartus ii university interface program. [Online]. Available: http://www.altera.com

[49] B. Khaleghi, B. Omidi, H. Amrouch, J. Henkel, and H. Asadi, "Estimating and mitigating aging effects in routing network of fpgas," *Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 3, 2019.

[50] F. Li, Y. Lin, L. He, D. Chen, and J. Cong, "Power modeling and characteristics of field programmable gate arrays," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 11, 2005.

[51] S. Yazdanshenas and H. Asadi, "Fine-grained architecture in dark silicon era for sram-based reconfigurable devices," *Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 10, 2014.

[52] F. Li, Y. Lin, L. He, and J. Cong, "Low-power fpga using pre-defined dual-vdd/dual-vt fabrics," in *International symposium on Field programmable gate arrays*. ACM, 2004.

[53] C. Chiasson and V. Betz, "Should fpgas abandon the pass-gate?" in *FPL*, vol. 13, 2013.

[54] S. Yazdanshenas, K. Tatsumura, and V. Betz, "Don't forget the memory: Automatic block ram modelling, optimization, and architecture exploration," in *International Symposium on Field-Programmable Gate Arrays*, 2017.

[55] L. McMurchie and C. Ebeling, "Pathfinder: a negotiation-based performance-driven router for fpgas," in *Reconfigurable Computing*, 2008.

[56] V. Betz, J. Rose, and A. Marquardt, *Architecture and CAD for deep-submicron FPGAs*. Springer Science & Business Media, 2012, vol. 497.

[57] C. Ebeling, L. McMurchie, S. A. Hauck, and S. Burns, "Placement and routing tools for the triptych fpga," *Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 3, no. 4, 1995.

[58] "Virtex-6 fpga configurable logic block." User Guide, Xilinx, Februray, 2012.

[59] "7 series fpgas data sheet: Overview." Data Sheet, Xilinx, March, 2017.

[60] (2013 (accessed April 15, 2019)) Predictive technology model (ptm). [Online]. Available: http://ptm.asu.edu/

[61] R. Ho, *On-chip wires: scaling and efficiency*. Stanford University, 2003.

[62] Z. Seifoori, B. Khaleghi, and H. Asadi, "A power gating switch box architecture in routing network of sram-based fpgas in dark silicon era," in *Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2017.

[63] S. Yazdanshenas, B. Khaleghi, P. Ienne, and H. Asadi, "Designing low power and durable digital blocks using shadow nanoelectromechanical relays," *Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 12, 2016.