Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Dynamic power-gating for leakage power reduction in FPGAs[#]

Hadi JAHANIRAD

Department of Electronics and Communication Engineering, University of Kurdistan, Sanandaj 66177-15175, Iran

E-mail: h.jahanirad@uok.ac.ir

Received Mar. 5, 2022; Revision accepted Oct. 10, 2022; Crosschecked Mar. 2, 2023

Abstract: Field programmable gate array (FPGA) devices have become widespread in electronic systems due to their low design costs and reconfigurability. In battery-restricted applications such as handheld electronics systems, low-power FPGAs are in great demand. Leakage power almost equals dynamic power in modern integrated circuit technologies, so the reduction of leakage power leads to significant energy savings. We propose a power-efficient architecture for static random access memory (SRAM) based FPGAs, in which two modes (active mode and sleep mode) are defined for each module. In sleep mode, ultra-low leakage power is consumed by the module. The module mode changes dynamically from sleep mode to active mode when module outputs evaluate for new input vectors. After producing the correct outputs, the module returns to sleep mode. The proposed circuit design reduces the leakage power consumption in both active and sleep modes. The proposed low-leakage FPGA architecture is compared with state-of-the-art architectures by implementing Microelectronics Center of North Carolina (MCNC) benchmark circuits on FPGA-SPICE software. Simulation results show an approximately 95% reduction in leakage power consumption in sleep mode. Moreover, the total power consumption (leakage+dynamic power consumption) is reduced by more than 15% compared with that of the best previous design. The average area overhead (4.26%) is less than those of other power-gating designs.

Key words: Field programmable gate array (FPGA); Leakage power; Power-gating; Transistor-level circuit design
https://doi.org/10.1631/FITEE.2200084 **CLC number:** TN79

1 Introduction

The in-field programmability of field programmable gate array (FPGA) has motivated system designers to generate several platforms based on such devices (Ma et al., 2018; Mitra and Nayak, 2018; Nguyen et al., 2019; Colleman and Verhelst, 2021; Kim et al., 2021; Li J et al., 2021). Generally, the main features of integrated circuits such as speed, area, and reliability are analyzed to boost the efficiency of digital systems (Jahanirad, 2019; Savari and Jahanirad, 2020; Rahimi and Jahanirad, 2021). In modern FPGAs, power consumption is a significant concern. Despite their advantages (reconfigurability, short time to market, and others), high power consumption makes FPGAs unattractive for applications where low power is a concern (Kuon and Rose, 2007). Standby or idle state is a common state in handheld systems, wherein the only source of power consumption is the static/leakage power (Amara et al., 2006). Researchers have proposed two approaches to address the leakage power problem in static random access memory (SRAM) based FPGAs. In the first approach, manufacturing-based low-power techniques are employed. The use of multiple threshold voltages (V_{th}), triple gate oxide, and variable gate length transistors has resulted in an approximately 40% power reduction in modern FPGAs (Singh et al., 2018; Tatsumura et al. 2018). Some recent studies have worked on developing different low-power architectures for memory cells (such as resistive random access memory (RRAM)) in

[#] Electronic supplementary materials: The online version of this article (https://doi.org/10.1631/FITEE.2200084) contains supplementary materials, which are available to authorized users

ORCID: Hadi JAHANIRAD, https://orcid.org/0000-0001-8586-6281

FPGAs (Kaur et al., 2018; Khaleghi and Asadi, 2018; Tang et al., 2018). These approaches have two main disadvantages—performance degradation (e.g., a greater gate delay due to high V_{th} transistors) and higher production cost due to manufacturing complexity.

The second category employs power-gating techniques in which the connection between the high/low supply rail and the main circuit is controlled using header/footer transistors (Bsoul and Wilton, 2010; Ishihara et al., 2011; Ahmed R et al., 2014, 2015; Bsoul et al., 2016). There are two main power-gating techniques: static and dynamic. In the static power-gating (Static-PG) approach, the unused resources of FPGA are disconnected from either supply rail and remain in sleep mode during circuit operation. On the other hand, in the dynamic power-gating (DPG) approach, the power modes of FPGA resources are determined online. The resource mode of operation (active/sleep mode) is determined dynamically during the circuit's runtime. In this paper, we are concerned with the second low-power design category. However, manufacturing-based techniques can be orthogonally applied to FPGA architecture design (Ebrahimi et al., 2017; Koppa and John, 2018).

In the power-gating approach, the power state of any FPGA module is defined as always-on for frequently used modules, always-off for unused resources, and dynamically controlled for modules with a long idle period (Bsoul et al., 2016). In the dynamically controlled module, the power-gating transistors are turned on and off to define the active and sleep states, respectively. Transition from sleep mode to active mode requires draining a relatively large inrush current (Ishihara et al., 2011). This current can cause a longer wake-up time and greater power overhead. Moreover, the required control voltages for header and footer transistors can be generated using a power management unit (PMU).

A modern FPGA architecture consists of an $M \times N$ array of configurable logic blocks (CLBs) that implement logic operations, surrounded by input–output (IO) pins that connect the internal blocks to the outside world. Some specific application modules (e.g., block random access memories (RAMs) and digital signal processors (DSPs)) are placed in the FPGA (by replacing some CLBs). Interconnections among the CLBs and other embedded blocks are realized using a flexible routing network. The routing network consists of variable types of wire segments, switch boxes (SBs) that connect the wires, and connection boxes (CBs) that connect the routing-wire segments to the input pins of the CLBs.

Two aspects are related to the power-gating-based design of FPGA architecture. The first is how to generate the power controller and deliver the power control signals (PCSs) to the FPGA modules (CLBs, SBs, CBs, and hardware-embedded modules). The power controller is implemented externally or internally. The external power controller is usually implemented using a processor (e.g., a microcontroller) that is programmed to appropriately generate the PCSs (Seomun and Shin, 2011). The internal power controller is implemented using the internal resources of the FPGA. Connecting the PCSs to the power-gating regions (PGRs) is carried out using two approaches: first, the required routes can be implemented using a flexible routing network (routing channels (RCs), SBs, and CBs); second, it can be implemented using pre-defined power-related routing tracks (Koppa and John, 2018). The first approach is necessarily included in the routing phase of the design flow.

The second aspect is how to switch the module's state between active and sleep modes. In the previous methods, this mode changing is applied by header or/and footer transistors. To change the module's state from active mode to sleep mode, the related PCS turns off the footer and/or header transistors, resulting in a significant leakage reduction. On the other hand, by turning on the footer and header transistors, the module returns to its normal (active) mode.

In summary, the problems of low-power FPGAs can be presented as follows:

1. Properly define PGRs and optimize routing of PCSs.

2. Minimize the leakage power consumption of internal FPGA modules by designing efficient architectures.

3. The overall design should impose minimum area and delay overheads along with a small complexity of the implementation process.

In this paper, we address the above three problems as follows: We propose a power-gating architecture, in which the entire FPGA area is partitioned into fixed PGRs. Pre-defined routing resources accomplish the routing of PCSs to the PGRs. Simulation results indicate that the proposed idea can efficiently handle the first problem with low complexity overhead. Further, efficient architectures are proposed for the main modules (CLBs, SBs, and CBs). The proposed design presents Static-PG for unused resources and DPG for the extra idle modules. As a novel element, in this paper, the proposed power-gating architecture handles leakage power consumption in active mode, which was ignored in previous studies.

The main contributions of this paper are outlined as follows: (1) the presentation of an efficient architecture for delivering PCSs to PGRs, (2) DPG of logic and routing resources with long idle periods, (3) the re-design of the main modules (CLBs, SBs, and CBs) so that the unused resources are statically power-gated, and (4) the handling of leakage power consumption in active mode of internal modules.

2 Related works

This section presents previous studies regarding circuit-level low-power FPGA designs. Tuan et al. (2007) employed various manufacturing (low-leakage SRAM) and power-gating techniques to reduce the power consumption of a 90-nm Spartan-3 FPGA.

Kumar and Anis (2007) developed a computer-aided design (CAD) tool, wherein the logic and routing resources were efficiently determined using high V_{th} and low V_{th}. Simulation results indicated a 50% reduction in leakage power.

Anderson and Najm (2006) focused on active leakage power reduction through two challenging techniques. The first is the interchanging of logic states of look-up tables (LUTs) so that the FPGA resources spend most of time in low-leakage states. In the second technique, a low-leakage-aware routing step is developed, wherein the low-leakage resources are used more for path construction. Simulation results showed an almost 30% reduction in active mode leakage.

Li F et al. (2007) presented a programmable dual V_{dd} to assign proper supply voltage ($V_{\mathrm{dd,H}}$ and $V_{\mathrm{dd,L}}$) for various FPGA resources. They developed a CAD tool to select the proper V_{dd} for routing resources at configuration time. In addition, a voltage level converter circuit was applied to derive the input of a $V_{\mathrm{dd,H}}$ module with the output of the $V_{\mathrm{dd,L}}$ module. A total power reduction of 47% was achieved using this architecture. Lin Y and He (2006) proposed two techniques to eliminate the voltage level converter. The first technique constructs the routing path so that only a single V_{dd} level exists in the corresponding routing tree. The second technique implements the routing tree of each net using both $V_{\mathrm{dd,L}}$ and $V_{\mathrm{dd,H}}$. However, in each path, the $V_{\mathrm{dd,H}}$ switches are not derived from $V_{\mathrm{dd,L}}$ switches. These modifications resulted in a 53% total power reduction for Microelectronics Center of North Carolina (MCNC) benchmark circuits.

Determining the optimal $V_{\mathrm{dd,L}}$ value is the subject of Zhu et al. (2014). In this work, based on implemented circuit characteristics (delay distribution, path overlap, transition density, etc.), the best value for $V_{\mathrm{dd,L}}$ regarding performance and power reduction was calculated. The method achieved a 30% power reduction for MCNC benchmark circuits.

Ahmed I et al. (2018) adopted dynamic voltage scaling (DVS) as a common technique to reduce dynamic power consumption in application-specific integrated circuit (ASIC) for FPGAs. The developed CAD tool found the optimum scaled-down V_{dd} value for the implemented circuit, so the necessary functionality and performance were maintained. Ultimately, they could achieve a 30% total power reduction by DVS for various benchmark circuits.

Nunez-Yanez et al. (2016) presented a voltage and frequency scaling approach to determine the most efficient V_{dd} and f_{clk} for the internal system modules of modern FPGAs. Voltage and frequency scaling, implemented by in-situ detectors, allowed designers to calculate the valid supply voltage and working frequency at runtime. This study reported a 60% total power reduction for two motion-estimation systems.

Ahmed I et al. (2020) presented a new design for LUT, in which the delay was less sensitive to V_{dd} reduction. This approach achieved a 26% reduction in Energy× (Delay)^2. Ravishankar et al. (2012) used another methodology to reduce FPGA power consumption. They used the guarded evaluation technique, putting some portions of the implemented circuit (sub-circuit) in a hold state, and thus switching activity of the sub-circuit was reduced in the hold time. This technique resulted in a 24% dynamic power reduction. Hassan et al. (2008) developed a novel approach, wherein without adding any hardware resource, the pass transistors in LUT and routing resources were put in a low leakage state simply by manipulating the input's order of LUTs and routing multiplexers (MUXs). This approach reduced leakage-power consumption by 50%.

The modification of routing circuitry (especially switches), presented in Lin MJ and El Gamal (2009) and Ramesh et al. (2021), can significantly reduce routing power consumption. Chen DM et al. (2010) adopted technology mapping and circuit graph clustering for dual-V_{dd} FPGA architecture, wherein a 14% power reduction was achieved compared to the conventional dual-V_{dd} design.

Ishihara et al. (2011) designed an asynchronous FPGA architecture, wherein each CLB autonomously transits from sleep to active mode when its inputs take a level transition (falling/rising edge). Due to the high overhead, the proposed autonomous design is appropriate for small-input LUTs (two-input LUTs). The fabricated asynchronous FPGA could reduce the total power consumption by 38% compared with its synchronous competitor.

Bsoul et al. (2016) developed a CAD tool to dynamically control the power state of FPGA modules using PCSs. The additional MUXs and programming SRAM cells of CLB apply proper PCSs to power-gating switches at runtime. DPG reduced the leakage power by 80%.

Seifoori et al. (2019) developed a K-means-based approach for clustering MUXs to form PGRs. The reduction in static power consumption was 37.5%, while an extra 5.48% area overhead was imposed on the FPGA. More recently, they improved the performance of the approach by presenting the routing algorithm to be aware of power-gating opportunities (Seifoori et al. 2021). Consequently, the static power consumption in routing resources was reduced by 53%, and the area overhead was in the same range (5.61%).

A family of routing switch designs proposed by Anderson and Najm (2009) are programmable to operate in high-speed, low-power, and sleep modes. Experimental results showed that the dynamic and leakage power consumptions of interconnecting networks were reduced by 28% –52% and 28%–31% in low-power mode, respectively. Furthermore, the leakage power was reduced by 61%–79% in unused routing resources.

Qi et al. (2017) presented a near/sub-threshold FPGA architecture using power-gating, per-path voltage scaling, a folded SB, and a low-switched global interconnect. They achieved a 39.4% reduction in energy for a 130-nm CMOS-technology-fabricated 512-LUT FPGA.

Tan et al. (2018) applied a clock-gating technique to reduce the dynamic power consumption of the clock tree in Artix-7 from Xilinx with 28-nm technology and achieved a 24% power reduction.

Chen WT et al. (2016) proposed a high-speed, low-power programmable interconnect design. The design reduced the static power consumption by 37.4% and had 33.1% acceleration. The non-minimum channel-length technique reduced the leakage power consumption. Moreover, fast connection schemes between logic blocks and the optimization of wire segments were introduced to reduce path delay.

Wagle and Vrudhula (2022) attempted to apply the threshold logic gate (TLG) concept to improve the area, power, and performance of FPGAs. The proposed threshold logic FPGA (TLFPGA) reduced power consumption by 14%, and the area and delay overheads of the implemented circuits were reduced by 5% and 16%, respectively.

Herath et al. (2021) proposed a power-efficient mapping approach to implement large-scale applications on modern heterogeneous FPGAs. In the mapping approach, a communication-aware placement methodology found the optimal shape of the modules. Simulation results showed a 19% reduction in power consumption without any performance degradation.

Seifoori et al. (2017) proposed a power-gating SB architecture to reduce the power consumption of a routing network. In this architecture, a configurable controller turns off the unused switches in the routing network. The exploration of various patterns of MUXs used in SBs is the basis for controller configuration. Experimental results confirmed a 31% total power reduction at the cost of a 33% area overhead.

Ebrahimi et al. (2017) used a combination of reconfigurable hardware logic design and a small-input LUT to boost the effectiveness of power-gating in turning off the unused resources of FPGA. In this architecture, dynamic, leakage, and total power consumptions were reduced by 10%, 9%, and 9%, respectively. The acceleration of the architecture was 5% at the cost of an 18.9% area overhead.

Vo (2018) developed a merged clock-gating technique, wherein all clock-gating signals were grouped into a single signal. Then a demultiplexer (DEMUX) was employed to split the signal into many different clock signals. The method led to a 3.45%–26.53% reduction in dynamic power consumption for clock frequencies ranging from 100 MHz to 1 GHz.

Li F et al. (2004) proposed a $V_{\rm dd}$ programmable interconnect fabric that included high $V_{\rm dd}$, low $V_{\rm dd}$, and power-gating states. High $V_{\rm dd}$ and low $V_{\rm dd}$ states were assigned to the critical and non-critical paths, respectively. Moreover, unused switches were programmed using the power-gating mode. The proposed design achieved a 50.55% total power reduction at the cost of a 6.17% delay overhead.

Huda and Anderson (2017) leveraged unused conductors to reduce both dynamic and static power consumptions of routing resources. The unused conductors were placed adjacent to the used conductor to reduce the effective capacitance of the active nets. This technique reduced the dynamic power consumption by 25% of the interconnects. The static power was reduced by floating the conductors connected to the MUXs in the power-critical paths. The static power consumption of unused interconnects decreased by 81%, and the overall energy reduction ranged from 14.9% to 42.7%. Delay and area overheads were 1.8% and 2.6%–4.8%, respectively.

In this paper, we develop a new low-power design, wherein the PCSs are routed through dedicated wires. The structure of PGRs (PGR granularity) is defined based on the PCS architecture.

In the next step, we design a power-gating version of the basic logic element (PG-BLE). BLE consists of an LUT, a data flip flop (DFF), and MUXs. The primary novelty of the low-power BLE is sleep and active leakage-aware power-gating. Furthermore, a clock-gating version of DFF retains the last registered logic value. Finally, using the retention property of 2×1 MUX, the more recent output of BLE is held, and the greatest isolation in sleep mode is achieved.

In the last step of our design, a low-power architecture is used for routing resources (including CLB crossbar, SBs, and CBs). Reconfigurable MUX as the main part of these modules includes programmable pass transistors and a tapered buffer. After programming MUX, only a small number of pass transistors are connected. Thus, we efficiently turn off the unused pass transistors to significantly reduce the leakage power consumption in active mode. The tapered buffer is power-gated based on the PGR's PCS.

The proposed design reduces the leakage power consumption in all FPGA modules. A PGR's logic (BLE) and routing resources (SBs, CBs, and cross boxes) are power-gated simultaneously in sleep mode. Further, the active modules, which were not power-gated in previous studies, contain many unused resources. In our proposed design, such unused resources are power-gated in active mode. This technique leads to a significant leakage reduction of the active mode. On the other hand, the novel routing of PCSs, which define the PGRs, results in both area reduction and ease of PGR placement and routing. Experimental results indicated a more than 95% reduction in leakage power. Moreover, the total power consumption (dynamic power plus leakage power of sleep and active modes) decreased by 80%. This achieved a 15% improvement compared with the study most related to this one.

3 Power controller and power control signals

When a digital system (consisting of M different modules) is transformed into a power-gating version, the modules are partitioned into clusters. Each cluster is called a PGR, wherein the sleep and active modes of the modules are similar. PGRs can be determined based on various approaches. A typical methodology is based on the task of the modules. For example, modules belonging to the 32-bit full adder would be considered a single PGR. The analysis of the PGR determination is beyond the scope of this paper.

The power controller, which can be internal or external, generates the necessary PCS to determine the PGR power mode (sleep or active). In Fig. 1, we conceptually show a digital system containing M=6 modules and two PGRs. Therefore, the power controller should generate two PCSs, one for PGR1 and the other for PGR2. The PCS wires for PGR1 and

PGR2 are indicated in green and red, respectively. The power-controller design is application-dependent. Thus, in the ASIC design, the power controller is implemented similarly to the other modules' designs. In the FPGA, the internal power controller is implemented using the available resources (CLBs, SBs, etc.). As noted in Section 1, if the PGRs experience a long idle time, the power-gating approach is cost-efficient. For instance, supposing that the PGR1 in Fig. 1 is in the idle mode for 200 ms, this accounting for about 4×10^{10} cycles of a 200-MHz FPGA's clock. The internal power controller uses an embedded hard core (a counter) to produce this significant delay (200 ms). On the other hand, the external power controller (a microcontroller) uses embedded timer/counter modules to achieve this goal.

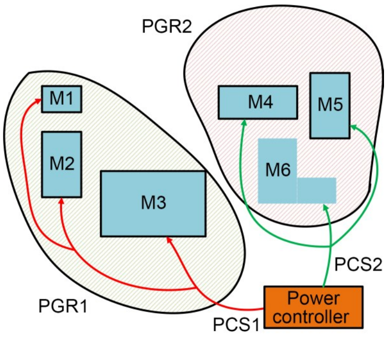

Fig. 1 A digital system consisting six modules and two PGRs (PCS: power control signal; PGR: power-gating region)
References to color refer to the online version of this figure

PCSs are routed using wire segments, SBs, and CBs in the routing step of the implementation flow. Bsoul et al. (2016) developed a low-overhead routing architecture for PCSs that operates based on the locality of the modules in a single PGR. The modules in a PGR correspond to a single task (e.g., an arithmetic and logic unit (ALU)) and consequently are highly interconnected. The current FPGA placement algorithms consider the compaction of the net's (a source module and its related sink) bounding box as their fundamental objective. Consequently, the modules related to a single task are placed in a compact portion of the FPGA's area (neighbor CLBs).

A PCS connects the output of the power controller to each CLB (and to the other blocks) of the related PGR. In our proposed method, the FPGA area is divided into *P* power-gating domains. The granularity of power-gating domains is defined from a CLB to the total area of the FPGA. The granularity level of the power domains is determined based on the power saving achieved by the power-gating technique. In Section 1 of the supplementary materials, we discuss how the granularity of PGR affects the energy saving of the power-gating approach.

The number of outputs of the power controller equals the number of the defined FPGA's power domains. Some dedicated routing tracks connect a power-controller output to the related power-domain CLBs. For a PGR placed on a portion of the FPGA, the power domains which cover all the PGR's blocks are determined, and similar PCSs are generated by the power controller for the covering power domains. Note that it is possible for several PGRs to be placed in a power domain. For this situation, the power controller generates the AND of all the PCSs for the power-gating of such a power domain. Each FPGA block supports the choice of Static-PG (for the blocks in a power domain that are not used in the implementation) or DPG (through the PCS).

Fig. 2 shows a typical PCS routing network. Our proposed method consumes only two PCSs for the power-gating of eight tiles in a 4-tile granularity (green lines), while the 1-tile granularity requires eight PCSs (red lines).

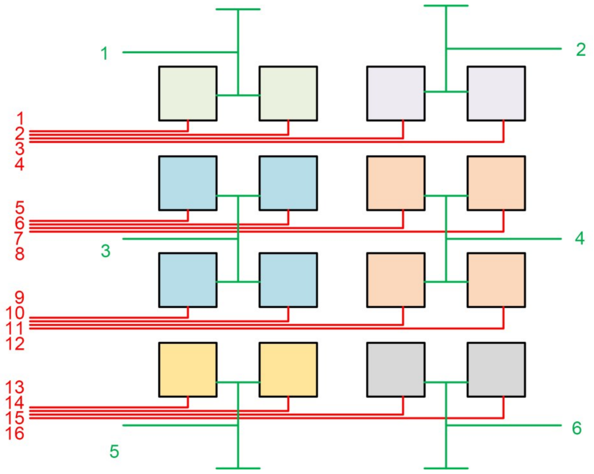

Fig. 2 Power control signal routing network
References to color refer to the online version of this figure

The dedicated routing tracks for PCSs impose area overhead to RCs. This overhead is unified for all the RCs in our proposed architecture. On the other hand, for the approaches wherein the PCSs are routed similarly to the other signals (using the FPGA's routing resources), this overhead is smaller due to on-demand routing. Despite this, our proposed approach has two advantages. First, the power overhead due to the buffers, NOTs, and MUXs in PCS routing is eliminated. Second, there is no need to involve PCS routing in the routing phase of the implementation process.

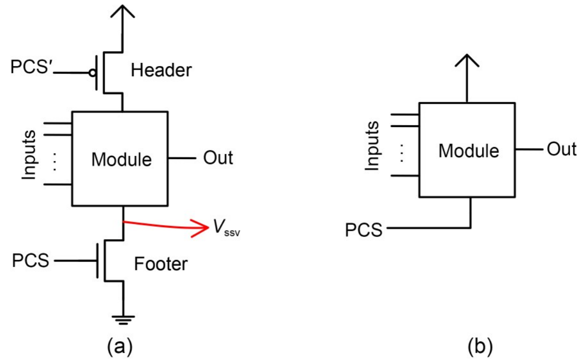

Fig. 3 Power-gating of a module: (a) conventional method; (b) proposed method (PCS: power control signal)

4 Power-gating strategy

Two main leakage mechanisms, sub-threshold leakage and gate leakage currents, exist in a CMOS transistor. A brief description of these mechanisms can be found in Section 2 of the supplementary materials.

In the Static-PG approach for a module, the PCS turns off the header and footer switches (Fig. 3a). Due to the smaller size of NMOS transistors (the carrier mobility of electrons in the NMOS channel is greater than the holes in the PMOS channel), the footer is the preferred approach for power-gating. The leakage current (due to various mechanisms such as sub-threshold conduction, gate leakage current, and others) encounters a significant barrier when the footer is turned off. In this case, power-gating leads to a substantial reduction in the leakage power of the related module. When the footer is turned off, the voltage value of the virtual ground node in Fig. 3a rises to a positive value (V_{ssv}). The gradual increase in voltage is due to the effects of the leakage currents. The footer imposes two major drawbacks in a power-gating module: the first is the area overhead by adding the power switch; the second is the delay overhead due to the increase of the body effect.

In this paper, we use PCS as V_{ssv} to define the lower supply voltage of the module (Fig. 3b). The proposed architecture eliminates the footer transistor, which removes the extra delay due to the body effect. As explained later, the various modules in the FPGA are constructed by inverters (NOT gates) and transmission gates (TGs) (or equivalent pass transistors). When the lower supply voltage of a NOT gate rises from GND to V_{dd}, two situations may occur (Fig. 4). In the first situation (Fig. 4a), the input of the NOT gate is V_{dd}, the NMOS transistor turns on, and the PMOS transistor turns off. When the NOT gate enters sleep mode (by connecting the lower supply rail to V_{dd}), the NMOS transistor charges the load capacitance to $V_{dd}-V_{th,n}$. In this case, the voltage differences between the terminals of both NMOS and PMOS transistors are minimal. This situation leads to very low leakage currents (sub-threshold and gate leakage currents would be minimal). In the second situation (Fig. 4b), the input of the NOT gate connects to the GND rail, and the NMOS and PMOS transistors turn off and on, respectively. When the NOT gate enters sleep mode, the drain and source terminals of NMOS and PMOS transistors connect to V_{dd}, but their gate terminals connect to GND. In this case, due to equal voltages in the drain and source terminals, the sub-threshold leakage current is negligible. However, the gate leakage current is much more than that in the first situation, due to the maximum voltage difference between the gate and drain/source terminals. Consequently, our proposed approach achieves a low leakage configuration in all sleep modes.

In each module (including MUXs), the signal usually travels along a path (Fig. 5). The path originates from the output of a NOT gate and terminates

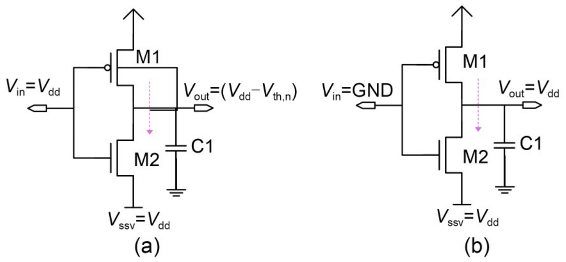

Fig. 4 NOT gate: (a) $V_{in}=V_{dd}$; (b) V_{in}=GND

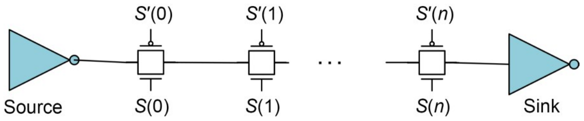

Fig. 5 A typical path in FPGA

at the input of another NOT gate. The middle stages of the path consist of TGs. If the path turns on, then all middle TGs turn on. If such a path enters sleep mode, then all nodes take a voltage equaling $V_{dd}-V_{th}$. In this case, the sub-threshold and gate leakage currents of NMOS transistors are minimal. Furthermore, the voltage values of the drain and source terminals of PMOS transistors effectively reduce the sub-threshold leakage current. However, the gate leakage current is not minimal.

On the other hand, if the path turns off, then some or all of the TGs of the path turn off. The drain and source terminals of turned-off TG transistors become floating. In the path illustrated in Fig. 5, the first TG experiences a different condition in which the left terminals of the transistors do not float. We now compare the state of leakage currents in our approach with its conventional counterpart. For the transistors with floating drain and source terminals, the leakage current of our proposed approach is similar to that of the conventional design. For the first turned-on TG of the path, in sleep mode, the voltage of the left terminal is V_{dd} or $V_{dd}-V_{th,n}$ for our proposed design. However, in conventional design, the first TG experiences V_{dd} or V_{ssv} in this terminal. In this case, the difference between V_{ssv} and $V_{dd}-V_{th,n}$ may cause a tiny change in the leakage current of sleep mode (especially for gate leakage current). This difference is negligible when the level of V_{ssv} reaches its steady-state value (almost V_{dd}).

In summary, connecting the PCS to the lower supply rail of the modules results in unifying the voltage level in all internal nodes, and effectively reduces sub-threshold and gate leakage currents. The proposed architectures for power-gating blocks (CLB and routing resources) are presented in the following subsections.

4.1 Power-gating of configurable logic block

The overall architecture of a CLB is illustrated in Fig. S1 of the supplementary materials. A cluster of N_{BLE} BLEs comprises the core of the logical part of CLB. Generally, a BLE consists of an LUT with k_{LUT} inputs, a DFF, and a 2×1 MUX (Fig. S3 of the supplementary materials). In each CLB, a full crossbar (containing $N_{BLE}\times k_{LUT}$ MUXs) facilitates the connection of any CLB input and output (using feedback paths) to each N_{BLE} LUT input.

In the proposed power-gating CLB (PG-CLB), first, a PCS is connected to all the CLB modules through a dedicated pin. Second, when a BLE enters sleep mode, its output data must be retained by a special latch (R-latch). The R-latch's power mode is always-on to capture the last generated data at the BLE output. When a BLE starts to enter sleep mode, the input TG of R-latch (TG-R) disconnects the R-latch from the previous stage so that the transition of the 2×1 MUX output does not corrupt the valid stored data in the R-latch.

The internal circuitry of a k_{LUT}-input LUT is given in Fig. 6. A 2^k×1 MUX connects one of the 2^k SRAM cell outputs to the LUT output. Several low-leakage architectures have been proposed for SRAM cells that could be employed in the proposed PG-LUT; therefore, we do not focus on the SRAM cell design. Fig. 6 shows the overall architecture of the 2^k×1 MUX. A k-level binary tree of TGs makes it possible to connect each output of the 2^k SRAM cells to the LUT output. The inverted and buffered versions of each LUT input are generated using minimum-size NOT gates and delay-optimized buffer gates, respectively. The SRAM

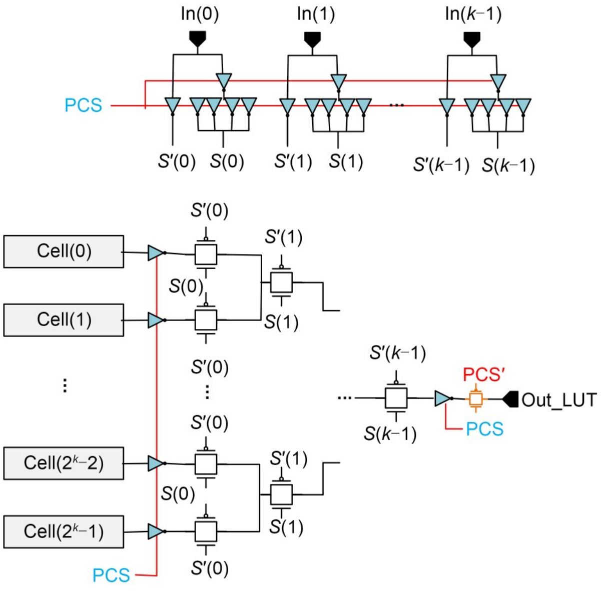

Fig. 6 Power-gating look-up table (LUT)

cell output is connected to the TG of the first level in 2^k×1 MUX by a minimum-size NOT gate. On the other side, a NOT gate is inserted into the 2^k×1 MUX's output. In the proposed PG-LUT, the related PCS connects to the lower supply voltage of all the LUT's inverters and buffers.

Fig. 7 shows the internal circuitry of a power-gating DFF (PG-DFF). The clock signal is disconnected using TG in sleep mode, while the slave latch of PG-DFF retains the last captured input data. The internal circuitry of the 2×1 PG-MUX is shown in Fig. 8. The main contribution is inserting the R-latch before the output tapered buffer. The output tapered buffer is a three-stage buffer, wherein stages 1, 2, and 3 comprise 1, 4, and 16 minimum-size inverters that are connected in parallel, respectively. The tapered buffer can efficiently drive the following routing resources (SBs and wires). All the inverters of the 2×1 MUX are power-gated using PCS, and the R-latch retains the last produced data at the BLE output.

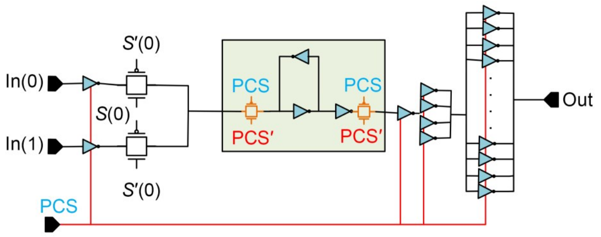

Fig. 7 Power-gating data flip flop (DFF)

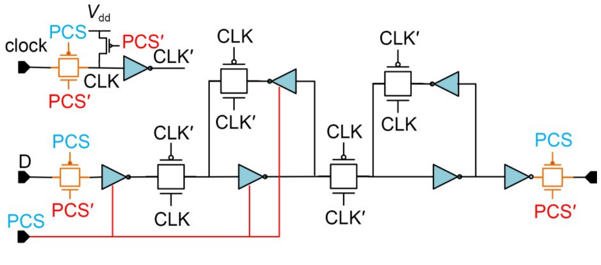

Fig. 8 Power-gating 2×1 MUX

The fully connected crossbar architecture is shown in Fig. 9. $N_{\text{BLE}}\times k$ MUXs feed the proper signal to each BLE input. If $N_{\text{CLB-inp}}$ input lines come from CLB's adjacent RCs, each MUX takes $N_{\text{CLB-inp}}$ input tracks plus N_{BLE} output wires and connects them to an input of BLEs selectively. The selection is performed using programming SRAM cells at the FPGA configuration phase. After configuration, only one of the $N_{\text{CLB-inp}}+N_{\text{BLE}}$ possible paths is on, and the others are useless. The main modifications we have applied to these MUXs are as follows: the input and output NOT gates are power-gated using PCS, and a TG is inserted into the MUX's output that turns off in sleep mode.

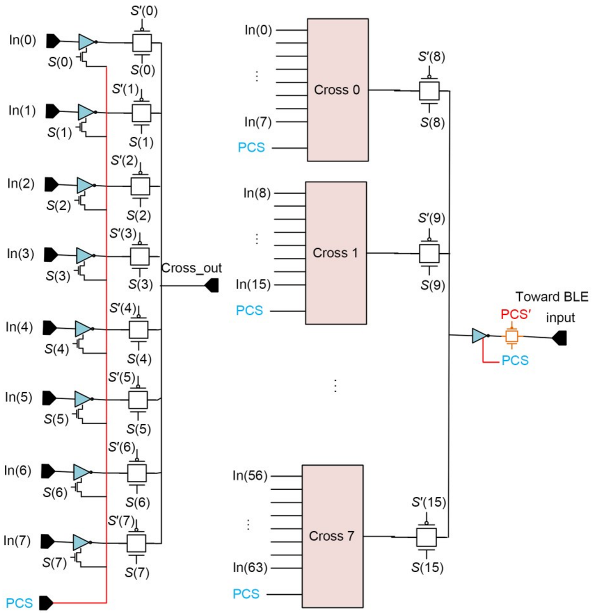

Fig. 9 Power-gating crossbar

Further, due to very low resource utilization in the crossbar MUX, another modification is applied to reduce the MUX's leakage power in active mode (P_{leakage}(on)). The NOT gates on the input side of the MUX are implemented using Static-PG structure (employing a footer switch). The input of the footer transistor connects to the corresponding programming SRAM-cell outputs. If the output of this SRAM cell is 1, then the footer transistor turns on, and the NOT gate enters active mode (if the CLB is in active mode). Thus, only one NOT gate is functional, and the other seven NOT gates are in the low-leakage regime.

4.2 Power-gating of SBs and CBs

The main resources of the routing network are wire segments (or RCs), SBs, and CBs. A sample path from the output of a source CLB to the input of a destination CLB is shown in Fig. 10. The source CLB output is connected to a wire segment in the horizontal RC through its adjacent SB. Then the wire segment is connected to the vertical wire segment using another SB. Finally, the last wire segment is connected to the input of the destination CLB through the proper CB.

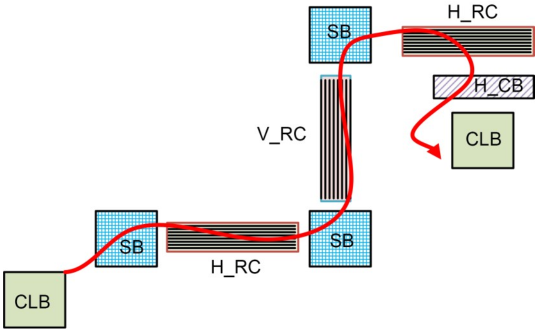

Fig. 10 A sample path

Each CLB is accompanied by two CBs, one for horizontal RC (H_CB) and the other for vertical RC (V_CB). H_CB (V_CB) supports the facility for connecting the wire segments of the horizontal (vertical) channel to the upper (lower) half of the external crossbar inputs of the CLB. The connections from wire segments of an RC to a specific input of the CLB crossbar are realized using $\mathrm{MUX}_{\mathrm{CB}}$ (Fig. 11). The number of RC tracks that can connect to a CLB input is an architectural parameter. In our proposed architecture, PG-$\mathrm{MUX}_{\mathrm{CB}}$ is power-gated using the related PCS. PCS connects to the lower supply rail of all NOT gates and the tapered buffer at the output of PG-$\mathrm{MUX}_{\mathrm{CB}}$.

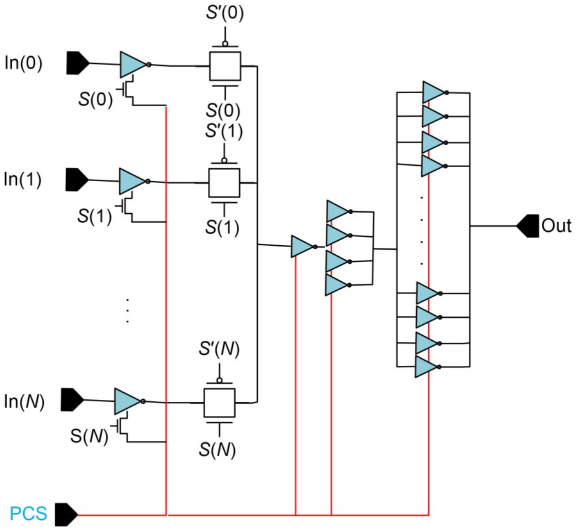

Fig. 11 Internal circuitry of the switch box (SB)/connection box (CB) multiplexer

Suppose that we decide to connect 10 wire segments to an input pin of a CLB. If the CLB has 16 inputs, then 160 NOT gates and 16 tapered buffers exist in H_CB and V_CB. This means that at most 16 NOT gates are functional in active mode. To reduce the leakage in active mode, we apply a similar approach to $\mathrm{MUX}_{\mathrm{cross}}$. The programming SRAM cell, which configures the TGs in $\mathrm{MUX}_{\mathrm{CB}}$, controls the on-off state of the footer switch.

SB is responsible for connecting two wire segments belonging to two different RCs, or the output of the adjacent CLB to the RC. $\mathrm{MUX}_{\mathrm{SB}}$'s realize such connections (Fig. 11). The inputs of $\mathrm{MUX}_{\mathrm{SB}}$ are connected to the related TGs using NOT gates, and the output of $\mathrm{MUX}_{\mathrm{SB}}$ is connected to the wire segment through tapered buffers. A similar power-gating strategy is applied to $\mathrm{MUX}_{\mathrm{SB}}$. The footer switches are controlled using corresponding programming SRAM cells. An important point relates to the type of signals routed using an SB. If the signal belongs to the power domain of the SB, then the local PCS is used in sleep mode. On the other hand, if the signal does not belong to the SB's power domain but uses only the SB's neighbor RC to pass through this domain, then the related path's $\mathrm{MUX}_{\mathrm{SB}}$ in the SB should not be power-gated. To address this issue, we use a 2×1 PG-$\mathrm{MUX}_{\mathrm{SB}}$ that can be programmed to connect local PCS or GND to the low-voltage supply rail for the first two cases.

5 Simulation results

In this section, we report the results of our simulations that prove our proposed design efficiency in the suppression of leakage power of SRAM-based FPGAs. First, the amount of single-tile leakage reduction in both sleep and active modes in our proposed design is compared with that of the Static-PG approach. Then, for the MCNC benchmark circuits, the leakage power reduction and area overhead of non-gated (NG), Static-PG, and DPG designs are compared with those of our proposed design.

To evaluate the electrical characteristics of a very large scale integration (VLSI) circuit, HSPICE is a powerful tool. Tang et al. (2019) developed an efficient tool called FPGA-SPICE that generates the transistor-level SPICE netlist of the FPGA fabric. The FPGA-SPICE flow starts with the establishment of the traditional place and path of a benchmark circuit on a user-defined FPGA architecture. This step is accomplished using the verilog to routing (VTR) tool (Luu et al., 2014). Next, based on the signal activity of the circuit nodes (which are derived using the ACE2 tool) and the SPICE netlist of the basic modules (TG, NOT, wire, DFF, and others), the necessary simulation decks are generated. Three different simulation levels are available for generated simulation decks (full-chip, grid, and component levels). The outputs of the simulations are the power consumption (both active and leakage) and delay reports for the implemented benchmark circuit. In our simulations, we use HSPICE-FPGA to derive the power, delay, and necessary signal waveforms. We use 65-nm technology in all simulations (http://ptm.asu.edu).

5.1 Simulation results for a tile

In this subsection, we compare the performance of a tile in our proposed design with those of the NG and Static-PG designs. In the Static-PG approach, a single footer transistor with proper sizing defines the module sleep/active mode. Various performance measurements were recorded for these three architectures: active mode leakage power (Fig. 12), sleep mode leakage power (Fig. 13), and tile area by transistor count (Fig. 14). Different cluster sizes (*N*=1, 4, 6, 10) were considered in this experiment.

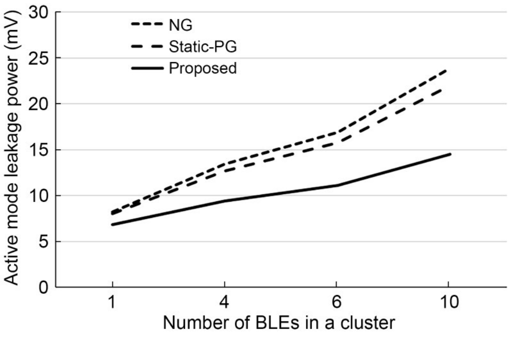

Fig. 12 Tile leakage reduction in active mode

Our proposed design add the following components to the NG design:

In PG-BLE, a retention latch is inserted in 2×1 MUX, along with DFF being replaced by PG-DFF. Assuming that the PMOS transistor size is two times larger than the NMOS size, then the transistor counts of NG-BLE and the proposed PG-BLE are 1557 and 1584, respectively. This indicates a 1.7% area overhead in PG-BLE. When we follow the path through which the signal travels from the BLE's SRAM cell to the output of 2×1 MUX, two NOT gates and three TGs are added to our design. According to these extra elements, the propagation delay of BLE increases by 18.3 ps (delays of the path in NG-BLE and PG-BLE are 167.62 ps and 185.75 ps, respectively) which achieves a 10.81% delay overhead.

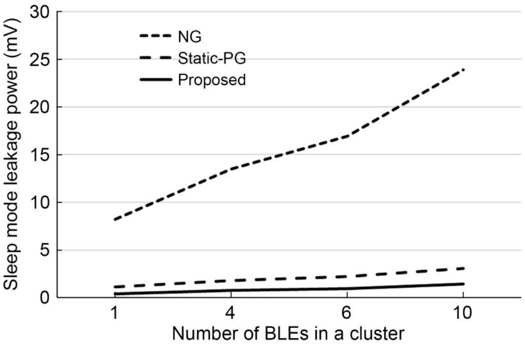

Fig. 13 Tile leakage reduction in sleep mode

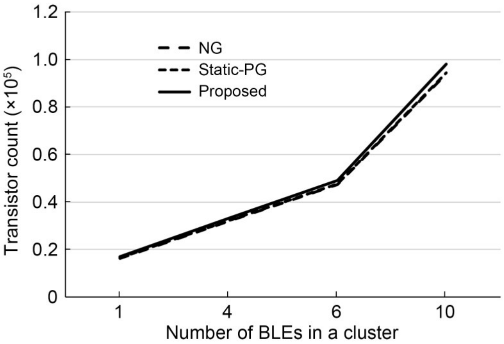

Fig. 14 The tile transistor count

The other major component in a tile is the crossbar, in which we add an NMOS transistor to each input-connected NOT gate. The transistor counts of the NG- and PG-crossbars are 579 and 643, respectively, which indicates an 11% area overhead in our design. Due to the extra NMOS transistors, the delay of the crossbars increases from 75.91 ps in the NG design to 87.57 ps in the PG-crossbar (a 15.36% delay overhead).

CBs contain MUXs for connecting the wire segments in adjacent channels to the input pins of the BLE. The extra NMOS transistors, which are controlled by the local SRAM cells, cause delay and area overheads. If the MUX_{CB}'s are of type 4×1, and if there are 40 such MUXs in H-CB and V-CB, then the transistor counts of NG-CB and PG-CB are 1320 and 1480, respectively (a 12.1% area overhead). Furthermore, the delay of each MUX increases from 60.78 ps in NG-CB to 66.25 ps in PG-CB, indicating a 9% delay overhead.

Another point of note is the efficiency of the proposed design regarding active leakage power reduction. The policy of inserting extra transistors to turn off the unused resources in the crossbar, CB, and SB modules results in 40% and 35% active leakage reduction compared to Static-PG and NG approaches, respectively.

The main sources of the delay overhead in our design are the retention latch in PG-LUT and the extra NMOS transistors, which are used to power-gate the unused resources in the crossbar, SBs, and CBs.

The leakage power is reduced by about 95% in sleep mode in our proposed design.

5.2 Comparisons

In this subsection, we compare our design with three other designs including NG, Static-PG, and DPG. The last design was introduced by Bsoul et al. (2016), in which the following major contributions were employed.

The power-gating of BLEs, RCs, and SBs is done by separated 3×1 MUXs wherein the first, second, and third inputs are related to GND, V_{dd}, and PCS, respectively. The granularity of each PGR determines how many BLEs are power-gated by a 3×1 MUX. However, PGRs, RCs, and SBs could be power-gated independently. Moreover, the internal switches in each SB can be power-gated at different levels of granularity (from one switch to all switches in the SB). This property could allow us to route PCS using only the necessary switches in the SB.

Based on the placement of the circuit modules, the required PCSs are routed using a conventional routing network. In the first step, the routing of PCSs is completed, and the related resources are configured in the always-on mode. This affects the SB and RC power states. For example, if the SB's granularity is the coarsest (to be controlled by only one 3×1 MUX), then passing a PCS through this SB would force all switches to be in the always-on mode.

To compare the proposed method with three other designs, we generate synthetic circuits using MCNC benchmark circuits. Each synthetic circuit contains several MCNC benchmark circuits, wherein each benchmark circuit acts as a PGR. Table S1 of the supplementary materials shows the properties of the synthetic circuits. In the experiments, we assume that the power controller is implemented externally. As noted earlier, the internal controller consumes more power and hardware resources. Thus, the external power controller is suitable for applications in which the imposed overhead is rational compared to the total power consumption.

Each synthetic circuit is implemented on a minimum-size FPGA. We run FPGA-SPICE for various PGR sizes and architecture parameters. Fig. 15 shows the total leakage power reduction for Static-PG, DPG, and our proposed architectures. The proposed method outperforms both Static-PG and DPG by 55% and 15%, respectively.

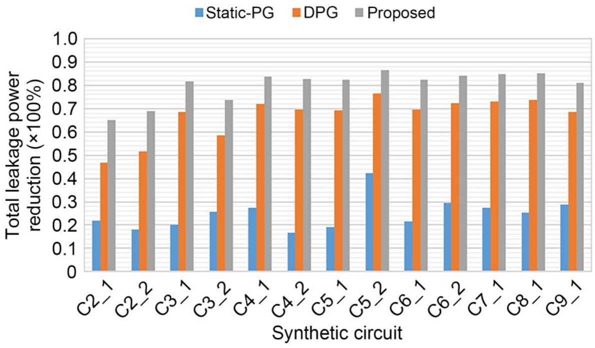

Fig. 15 Total leakage power reduction

Fig. 16 shows the impact of the average active time of the circuit on leakage power reduction. Because it spends more time in sleep mode, the power reduction increases with the decrease in average active time. Furthermore, our proposed design demonstrates better performance for circuits with highly active modules. This is due to the strategies employed to reduce the active-mode leakage.

Further, we examined various PGR sizes as shown in Fig. 17. Increasing the PGR size reduces total leakage reduction. When the PGR size increases, the probability of multiple modules being placed in a single PGR increases, and the average active time of the circuit increases.

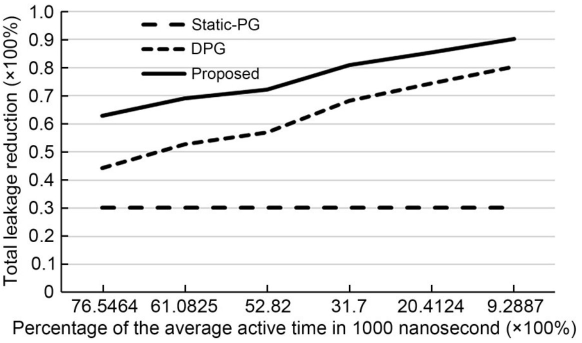

Fig. 16 Impact of modules' active time on leakage reduction

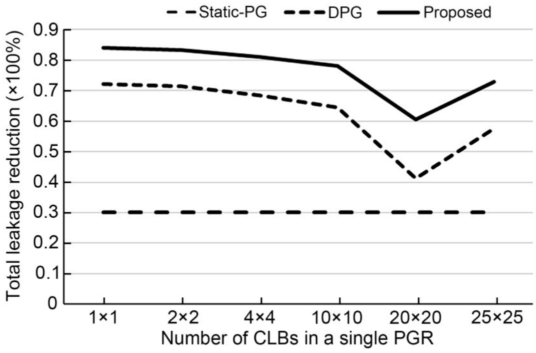

Fig. 17 Impact of power-gating region (PGR) size on leakage reduction

For a specific PGR size, the area overhead of various designs is fixed. In the DPG design, the following modules are added to a cluster: a 3×1 MUX to derive the gate terminal of the power switch (a PMOS transistor), and an $N_{\text{neighb}}\times1$ MUX (N_{neighb} is the number of neighborhood PGRs). The output of each BLE in a logic cluster is accompanied by a 2×1 MUX and an NMOS transistor to isolate it from surrounding RCs. A power-gating switch is dedicated to each RC, wherein a 3×1 MUX drives its gate terminal. The third input of this MUX is related to the dynamic mode that is generated by a two-input AND gate. If there are N_{SB} power regions in an SB, then N_{SB} power-gating setups including a power switch, a 3×1 MUX, and an isolation transistor are added to the SB architecture.

Table 1 shows the area overhead of various designs. The Static-PG architecture has the lowest area overhead, and the area overhead of our proposed method does not exceed 4.3% for different BLE cluster sizes. As mentioned earlier, the DPG strategy depends highly on the granularity of the SBs. Thus, in Table 1, we report the area overhead for three SB granularity cases. The area overhead for all cluster sizes is much larger than the area overhead of our proposed method (5.4%–33.1%).

Table 1 Area overhead comparison

Number of BLEs in a cluster	Satic-PG (%)	Proposed (%)	DPG (%)		
			1	3	Fine grained
1	1.42	4.13	6.5	9.8	35.0
4	0.58	3.51	5.7	9.5	33.1
6	0.35	3.66	5.6	9.3	32.0
10	0.48	4.26	5.4	9.1	31.5

Three columns of DPG are for various SB granularity

Table 2 shows a comparison with state-of-the-art, low-power FPGA designs. The second, third, and fourth columns report the reductions of dynamic, leakage, and total power consumption, respectively. The delay and area overheads are given in the last two columns. There are empty cells in the table due to the lack of data in related research papers. Moreover, the negative numbers of power consumptions indicate increase in power values, and the negative numbers of delay overhead indicate reduction in critical path delay.

The power-gating method is not concerned with dynamic power consumption. Thus, the pure power-gating methods (including our proposed method) slightly increase dynamic power consumption. Our proposed method degrades the dynamic power consumption (3%) less than those in Bsoul et al. (2016) and Tuan et al. (2007). The designs of Lin MJ and El Gamal (2009) and Chen DM et al. (2010) ignore leakage power consumption and focus only on the reduction of dynamic power consumption. The ignorance of leakage power consumption analysis in Chen DM et al. (2010) is not acceptable in modern technologies, where the dynamic and leakage power consumptions are approximately equal. Due to this, despite a 52.7% reduction in dynamic power consumption, the total power consumption decreases only by 14.3% (Chen DM et al., 2010).

The leakage (static) power consumption reduction of our proposed design outperforms that of all the other designs due to the power-gating of all resources in the FPGA (logic and routing parts). The leakage reduction described in Tuan et al. (2007) is the closest to that of the proposed design. However, considering the leakage power consumption in active

Table 2 Comparison of state-of-the-art designs

Low-power design	Dynamic power reduction (%)	Leakage power reduction (%)	Total power reduction (%)	Delay (%)	Area (%)
Tuan et al. (2007)	−5	92	51	27	40
Kumar and Anis (2007)		43		5.8	
Anderson and Najm (2006)		30		0.30	
Li F et al. (2007)	22.4–27.3	38.9	47.6	17.65	
Lin Y and He (2006)			52.5	20	66
Zhu et al. (2014)			29–33		
Ravishankar et al. (2012)	15.4–16.8			18–43	
Hassan et al. (2008)		50.3			
Lin MJ and El Gamal (2009)	57			−35	
Ramesh et al. (2021)		28–42		3.6	
Chen DM et al. (2010)	52.7		14.3		1.2
Seifoori et al. (2019)			26.25		5.48
Anderson and Najm (2009)	19.6–21.7	42.7–55.3			
Qi et al. (2017)			39.4		
Tan et al. (2018)	24				
Chen WT et al. (2016)		26.2		−33.1	
Wagle and Vrudhula (2022)			14	−5	−16
Herath et al. (2021)			13.3		
Seifoori et al. (2017)			31		30
Ebrahimi et al. (2017)	10	9	9	−5	18.9
Vo (2018)			3.45@100 MHz, 26.53@1 GHz		
Li F et al. (2004)			50.5	6.17	
Huda and Anderson (2017)	17.5	56.7	10.43–29.9	1.8	2.6–4.8
Seifoori et al. (2021)			37.5		5.61
Bsoul et al. (2016)	−7	55	66.7	10	5.4–35
Proposed design	−3	95	80.1	10	4.26

mode, the total power reduction of our design is 30% more than that of Tuan et al. (2007). Furthermore, our method outperforms Tuan et al. (2007)'s method in dynamic power consumption, delay overhead, and area overhead.

The total power consumption of our method is reduced by more than 14% compared to those of the best previous design (Bsoul et al., 2016). The main reasons are active leakage reduction, PCS definition, and power-gating of all resources of FPGA in our proposed method. Among the low-power strategies, V_{dd} programmability techniques such as those in Lin Y and He (2006), Li F et al. (2007), and Ebrahimi et al. (2017) have a significant impact on the total power consumption. As noted in Section 1, these techniques could be applied orthogonally in our design, which would improve the efficiency of the proposed method.

The delay overhead of the proposed method is moderate in comparison with those of other designs. The methods using new routing fabric or a low-power paradigm (Lin MJ and El Gamal, 2009; Wagle and Vrudhula, 2022) reduce critical path delay significantly. These methods can be included in our proposed design to reduce delay overhead. Other designs (Chen WT et al., 2016; Ebrahimi et al., 2017) modify the structure of LUT and SB circuits to achieve a small reduction in critical path delay.

The area overhead of the proposed method is lower than those of power-gating counterparts. The main rival in leakage power consumption (Tuan et al., 2007) consumes almost 10 times the chip area.

6 Conclusions

In this paper, a low-power FPGA architecture is developed in which the dynamic power-gating technique is employed to reduce the leakage power consumption of internal modules. Two main features distinguish the proposed design from previous ones.

First, a low-cost routing network is proposed to implement the power control signals. Second, each module is re-designed to reduce the leakage power in both active and sleep modes. A comparison between our proposed architecture and previous designs is carried out for synthetic benchmark circuits using the FPGA-SPICE framework. Simulation results demonstrate a 95% reduction of leakage power consumption in sleep mode. The total power consumption is reduced by 80% in the proposed design, which is 15% better than that of the best design in the literature. Meanwhile, the area overhead of the proposed design (less than 4.3%) outperforms those of the previous architectures.

Compliance with ethics guidelines

Hadi JAHANIRAD declares that he has no conflict of interest.

Data availability

The data that support the findings of this study are available from the author upon reasonable request.

References

Ahmed I, Zhao SZ, Trescases O, et al., 2018. Automatic application-specific calibration to enable dynamic voltage scaling in FPGAs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 37(12):3095-3108. https://doi.org/10.1109/TCAD.2018.2801222

Ahmed I, Shen JL, Betz V, 2020. Optimizing FPGA logic circuitry for variable voltage supplies. *IEEE Trans Very Large Scale Integr Syst*, 28(4):890-903. https://doi.org/10.1109/TVLSI.2019.2962501

Ahmed R, Bsoul AAM, Wilton SJE, et al., 2014. High-level synthesis-based design methodology for dynamic power-gated FPGA. Proc 24th Int Conf on Field Programmable Logic and Applications, p.1-4. https://doi.org/10.1109/FPL.2014.6927433

Ahmed R, Wilton SJE, Hallschmid P, et al., 2015. Hierarchical dynamic power-gating in FPGAs. Proc 11th Int Symp on Applied Reconfigurable Computing, p.27-38. https://doi.org/10.1007/978-3-319-16214-0_3

Amara A, Amiel F, Ea T, 2006. FPGA vs. ASIC for low power applications. *Microelectron J*, 37(8):669-677. https://doi.org/10.1016/j.mejo.2005.11.003

Anderson JH, Najm FN, 2006. Active leakage power optimization for FPGAs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 25(3):423-437. https://doi.org/10.1109/TCAD.2005.853692

Anderson JH, Najm FN, 2009. Low-power programmable FPGA routing circuitry. *IEEE Trans Very Large Scale Integr Syst*, 17(8):1048-1060. https://doi.org/10.1109/TVLSI.2009.2017443

Bsoul AAM, Wilton SJE, 2010. An FPGA architecture supporting dynamically controlled power gating. Int Conf on Field-Programmable Technology, p.1-8. https://doi.org/10.1109/FPT.2010.5681533

Bsoul AAM, Wilton SJE, Tsoi KH, et al., 2016. An FPGA architecture and CAD flow supporting dynamically controlled power gating. *IEEE Trans Very Large Scale Integr Syst*, 24(1):178-191. https://doi.org/10.1109/TVLSI.2015.2393914

Chen DM, Cong J, Dong C, et al., 2010. Technology mapping and clustering for FPGA architectures with dual supply voltages. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 29(11):1709-1722. https://doi.org/10.1109/TCAD.2010.2061770

Chen WT, Li L, Lu P, et al., 2016. Design of FPGA's high-speed and low-power programmable interconnect. Proc 13th IEEE Int Conf on Solid-State and Integrated Circuit Technology, p.707-709. https://doi.org/10.1109/ICSICT.2016.7999018

Colleman S, Verhelst M, 2021. High-utilization, high-flexibility depth-first CNN coprocessor for image pixel processing on FPGA. *IEEE Trans Very Large Scale Integr Syst*, 29(3): 461-471. https://doi.org/10.1109/TVLSI.2020.3046125

Ebrahimi Z, Khaleghi B, Asadi H, 2017. PEAF: a power-efficient architecture for SRAM-based FPGAs using reconfigurable hard logic design in dark silicon era. *IEEE Trans Comput*, 66(6):982-995. https://doi.org/10.1109/TC.2016.2636141

Hassan H, Anis M, Elmasry M, 2008. Input vector reordering for leakage power reduction in FPGAs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 27(9):1555-1564. https://doi.org/10.1109/TCAD.2008.927673

Herath K, Prakash A, Fahmy SA, et al., 2021. Power-efficient mapping of large applications on modern heterogeneous FPGAs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 40(12):2508-2521. https://doi.org/10.1109/TCAD.2020.3047722

Huda S, Anderson JH, 2017. Leveraging unused resources for energy optimization of FPGA interconnect. *IEEE Trans Very Large Scale Integr Syst*, 25(8):2307-2320. https://doi.org/10.1109/TVLSI.2017.2691409

Ishihara S, Hariyama M, Kameyama M, 2011. A low-power FPGA based on autonomous fine-grain power gating. *IEEE Trans Very Large Scale Integr Syst*, 19(8):1394-1406. https://doi.org/10.1109/TVLSI.2010.2050500

Jahanirad H, 2019. CC-SPRA: correlation coefficients approach for signal probability-based reliability analysis. *IEEE Trans Very Large Scale Integr Syst*, 27(4):927-939. https://doi.org/10.1109/TVLSI.2018.2886027

Kaur I, Rohilla L, Nagpal A, et al., 2018. Different configuration of low-power memory design using capacitance scaling on 28-nm field-programmable gate array. In: Muttoo SK (Ed.), System and Architecture. Springer, Singapore, p.151-161. https://doi.org/10.1007/978-981-10-8533-8_15

Khaleghi B, Asadi H, 2018. A resistive RAM-based FPGA architecture equipped with efficient programming circuitry. *IEEE Trans Circ Syst I Regul Pap*, 65(7):2196-2209. https://doi.org/10.1109/TCSI.2017.2778113

Kim S, Na S, Kong BY, et al., 2021. Real-time SSDLite object detection on FPGA. *IEEE Trans Very Large Scale Integr*

Syst, 29(6):1192-1205. https://doi.org/10.1109/TVLSI.2021.3064639

Koppa S, John E, 2018. Performance tradeoffs in the design of low-power SRAM arrays for implantable devices. *J Low Power Electron*, 14(1):18-27. https://doi.org/10.1166/jolpe.2018.1528

Kumar A, Anis M, 2007. Dual-threshold CAD framework for subthreshold leakage power aware FPGAs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 26(1):53-66. https://doi.org/10.1109/TCAD.2006.882595

Kuon I, Rose J, 2007. Measuring the gap between FPGAs and ASICs. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 26(2):203-215. https://doi.org/10.1109/TCAD.2006.884574

Li F, Lin Y, He L, 2004. Vdd programmability to reduce FPGA interconnect power. IEEE/ACM Int Conf on Computer Aided Design, p.760-765. https://doi.org/10.1109/ICCAD.2004.1382678

Li F, Lin Y, He L, 2007. Field programmability of supply voltages for FPGA power reduction. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 26(4):752-764. https://doi.org/10.1109/TCAD.2006.884848

Li J, Chow P, Peng YX, et al., 2021. FPGA implementation of an improved OMP for compressive sensing reconstruction. *IEEE Trans Very Large Scale Integr Syst*, 29(2):259-272. https://doi.org/10.1109/TVLSI.2020.3030906

Lin MJ, El Gamal A, 2009. A low-power field-programmable gate array routing fabric. *IEEE Trans Very Large Scale Integr Syst*, 17(10):1481-1494. https://doi.org/10.1109/TVLSI.2008.2005098

Lin Y, He L, 2006. Dual-Vdd interconnect with chip-level time slack allocation for FPGA power reduction. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 25(10):2023-2034. https://doi.org/10.1109/TCAD.2006.870858

Luu J, Goeders J, Wainberg M, et al., 2014. VTR 7.0: next generation architecture and CAD system for FPGAs. *ACM Trans Reconfig Technol Syst*, 7(2):6. https://doi.org/10.1145/2617593

Ma YF, Cao Y, Vrudhula S, et al., 2018. Optimizing the convolution operation to accelerate deep neural networks on FPGA. *IEEE Trans Very Large Scale Integr Syst*, 26(7):1354-1367. https://doi.org/10.1109/TVLSI.2018.2815603

Mitra J, Nayak TK, 2018. An FPGA-based phase measurement system. *IEEE Trans Very Large Scale Integr Syst*, 26(1):133-142. https://doi.org/10.1109/TVLSI.2017.2758807

Nguyen DT, Nguyen TN, Kim H, et al., 2019. A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection. *IEEE Trans Very Large Scale Integr Syst*, 27(8):1861-1873. https://doi.org/10.1109/TVLSI.2019.2905242

Nunez-Yanez JL, Hosseinabady M, Beldachi A, 2016. Energy optimization in commercial FPGAs with voltage, frequency and logic scaling. *IEEE Trans Comput*, 65(5):1484-1493. https://doi.org/10.1109/TC.2015.2435771

Qi H, Ayorinde O, Calhoun BH, 2017. An ultra-low-power FPGA for IoT applications. IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conf, p.1-3. https://doi.org/10.1109/S3S.2017.8308753

Rahimi H, Jahanirad H, 2021. An evolutionary approach to implement logic circuits on three dimensional FPGAs. *Expert Syst Appl*, 174:114780. https://doi.org/10.1016/j.eswa.2021.114780

Ramesh NVK, Uday Kiran K, Reshma NKSVS, et al., 2021. An efficient way to optimize the FPGA routing architecture. Proc 6th Int Conf on Communication and Electronics Systems, p.209-211. https://doi.org/10.1109/ICCES51350.2021.9489142

Ravishankar C, Anderson JH, Kennings A, 2012. FPGA power reduction by guarded evaluation considering logic architecture. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 31(9):1305-1318. https://doi.org/10.1109/TCAD.2012.2192478

Savari MA, Jahanirad H, 2020. NN-SSTA: a deep neural network approach for statistical static timing analysis. *Expert Syst Appl*, 149:113309. https://doi.org/10.1016/j.eswa.2020.113309

Seifoori Z, Khaleghi B, Asadi H, 2017. A power gating switch box architecture in routing network of SRAM-based FPGAs in dark silicon era. Design, Automation & Test in Europe Conf & Exhibition, p.1342-1347. https://doi.org/10.23919/DATE.2017.7927201

Seifoori Z, Asadi H, Stojilović M, 2019. A machine learning approach for power gating the FPGA routing network. Int Conf on Field-Programmable Technology, p.10-18. https://doi.org/10.1109/ICFPT47387.2019.00010

Seifoori Z, Asadi H, Stojilović M, 2021. Shrinking FPGA static power via machine learning-based power gating and enhanced routing. *IEEE Access*, 9:115599-115619. https://doi.org/10.1109/ACCESS.2021.3085005

Seomun J, Shin Y, 2011. Design and optimization of power-gated circuits with autonomous data retention. *IEEE Trans Very Large Scale Integr Syst*, 19(2):227-236. https://doi.org/10.1109/TVLSI.2009.2033356

Singh P, Reniwal BS, Vijayvargiya V, et al., 2018. Ultra low power-high stability, positive feedback controlled (PFC) 10T SRAM cell for look up table (LUT) design. *Integration*, 62:1-13. https://doi.org/10.1016/j.vlsi.2018.03.006

Tan BL, Lee WK, Mok KM, et al., 2018. Clock gating implementation on commercial field programmable gate array (FPGA). Proc 4th Int Conf on Electrical, Electronics and System Engineering, p.102-106. https://doi.org/10.1109/ICEESE.2018.8703530

Tang XF, Giacomin E, De Micheli G, et al., 2018. Post-P&R performance and power analysis for RRAM-based FPGAs. *IEEE J Emerg Sel Top Circ Syst*, 8(3):639-650. https://doi.org/10.1109/JETCAS.2018.2847600

Tang XF, Giacomin E, De Micheli G, et al., 2019. FPGA-SPICE: a simulation-based architecture evaluation framework for FPGAs. *IEEE Trans Very Large Scale Integr Syst*, 27(3):637-650. https://doi.org/10.1109/TVLSI.2018.2883923

Tatsumura K, Yazdanshenas S, Betz V, 2018. Enhancing FPGAs with magnetic tunnel junction-based block RAMs. *ACM Trans Reconfig Technol Syst*, 11(1):6. https://doi.org/10.1145/3154425

Tuan T, Rahman A, Das S, et al., 2007. A 90-nm low-power FPGA for battery-powered applications. *IEEE Trans*

Comput-Aided Des Integr Circ Syst, 26(2):296-300. https://doi.org/10.1109/TCAD.2006.885731

Vo MH, 2018. The merged clock gating architecture for low power digital clock application on FPGA. Int Conf on Advanced Technologies for Communications, p.282-286. https://doi.org/10.1109/ATC.2018.8587596

Wagle A, Vrudhula S, 2022. Heterogeneous FPGA architecture using threshold logic gates for improved area, power, and performance. *IEEE Trans Comput-Aided Des Integr Circ Syst*, 41(6):1855-1867. https://doi.org/10.1109/TCAD.2021.3099780

Zhu JF, Pan LY, Yan YR, et al., 2014. A fast application-based supply voltage optimization method for dual voltage FPGA. *IEEE Trans Very Large Scale Integr Syst*, 22(12):2629-2634. https://doi.org/10.1109/TVLSI.2013.2296791

List of supplementary materials