

# A Low-Power Field Programmable VLSI Based on Autonomous Fine-Grain Power Gating Technique

P. Durga Prasad, M. Tech Scholar, C. Ravi Shankar Reddy, Lecturer,  
V. Sumalatha, Associate Professor

Department of Electronics & Communication Engineering  
Jawaharlal Nehru Technological University  
Anantapur, India

**Abstract** - *FIELD-PROGRAMMABLE gate arrays (FPGAs) are widely used to implement special-purpose processors. FPGAs are economically cheaper for low quantity production because its function can be directly reprogrammed by end users. FPGAs consume high dynamic and standby power compared to custom silicon devices. This paper presents a low power field-programmable gate array (FPGA) based on lookup table (LUT) level fine-grain power gating with small overheads. The activity of each LUT can be easily detected using the proposed power gating technique by exploiting features of asynchronous architectures. In this paper, the novel Logic Block utilizing the LUT with autonomous power gating has been proposed and the developed model has been simulated and synthesized in a selected target device. Also the power analysis has been carried out and it has been found that power consumption is reduced by 48% compared to existing designs.*

**Index Terms** - *FPGA, Power Gating, Logic Block (LB), Lookup Table (LUT).*

## I. INTRODUCTION

Due to the development of IC technology, FPGAs play the major role on silicon industry. As compared with other silicon devices, FPGAs consume huge dynamic and standby power [1]. This increases the packaging costs and limit integrations of FPGAs into portable devices. These motivate the researchers to investigate new things in the area of power reduction using a suitable technique.

The clock network occupies a large proportion of the dynamic power in a FPGA because it has significantly more registers than custom VLSIs. The frequently used technique to reduce the clock network power is clock gating. In FPGAs, the customized clock network may be implemented using the programmable interconnects. Since FPGA vendors do not guarantee the worst case of the minimum delay of components, the worst case of clock skew cannot be estimated. As a result, it is impossible to guarantee that no hold-time violations occur.

The Xilinx ISE reference manual [3] recommends the use of clock gating without the customized global clock. In FPGAs, the circulation is employed to implement clock gating; The idea of circulation is to retain the contents of the flip-flop in the sleep state [4]. Circulation can reduce the dynamic power consumption of registers and the gates in the fan-out of the registers. However, the dynamic power consumption of the clock network cannot be reduced.

### A. Power Gating

The power consumption of power gating circuit is consumed by mainly the sleep controller, the sleep signal distribution network, and the sleep transistors. The fundamental challenge for any power gating technique is to ensure that the saved standby power outweighs the power overhead of the power gating. This technique uses high  $V_t$  sleep transistors which cut off  $V_{DD}$  from a circuit block when the block is not switching. The sleep transistor sizing is an important design parameter.

As compared with clock gating, power gating affects the design of architecture mainly. This increases time delays as power gated modes have to be safely entered and exited [9-11]. The possible amount of leakage power saving in such low power mode and the energy dissipation to enter and exit such mode introduces some architectural trade-offs. Shutting down the blocks can be accomplished either by software or hardware. Driver software can schedule the power down operations. Hardware timers can be utilized. A dedicated power management controller is the other option.

Switched power supply is the very basic form of power gating to achieve a very low leakage power in a long term manner. The internal power gating method is suitable to shut off the block for small intervals of time. The CMOS switches are controlled by power gating controllers then the output of the power gated block discharge slowly. Hence output voltage levels spend more time in threshold voltage level. This may lead to larger short circuit current. In standby or sleep mode, the power gating uses low-leakage PMOS transistors as switches to shut off power supplies. NMOS switches can also be used as sleep transistors. The cells can be turned off by inserting the sleep transistors into a permanent power network connected to the power supply and a virtual power network that drives the cells.

Power gating techniques can be classified into two types: coarse-grain power gating and fine-grain power gating. In coarse-grain power gating, a large number of LUTs share a single sleep controller so the area and power overheads of the sleep controller are relatively small. However, if any LUT within a coarse-grain power-gated domain is active, none of the LUTs which share the same sleep transistor can be set to the sleep mode. On the other hand, in fine-grain power gating, each LUT has its own sleep transistor and related sleep controller. Hence, when any LUTs are inactive, they can be set to the sleep mode immediately. This result in much lower standby power compared to coarse-grain power gating.

In fine-grain power gating, each LUT has its sleep controller, the number of the sleep controllers is much larger than that of coarse-grain power gating.

In synchronous architectures, the sleep controller consists of some memory bits to store the sleep time. The sleep controller is always running in a power gating circuit. This increase the area and dynamic power. Due to this, fine-grain power gating is commonly assumed to be less efficient than coarse-grain power gating, although it has the potential to cut most of the standby power [5], [9]. In spite of the importance of efficient sleep controllers, most studies on power gating focused on power-gated circuits or power-gated domain partitioning, but little work is carried out for sleep controllers.

### B. Implemented Method

In this paper, a low power FPGA employing a LUT-level power gating technique called autonomous fine-grain power gating has been proposed. Our asynchronous architecture detects the activity of a power-gated domain, and uses this activity to determine when to shut down and wake up the power-gated domain. By comparing the phases of the input data with that of the output data, the activity of a power-gated domain can be easily detected. In this technique, since the activity of each LUT can be detected easily, the area and the power overheads of the sleep controller are small. The power gating technique implemented in the proposed architecture can directly detect the activity of each look-up-table easily by exploiting features of asynchronous architectures. Moreover, detecting the data arrival in advance prevents the delay increase for waking-up and the power consumption of unnecessary power switching.

## II. EXISTING ARCHITECTURE

Fig. 2.1 shows the overall architecture of the existing FPGA which has a mesh-connected cellular array based on a bit-serial architecture. Each logic block (LB) has its sleep controller which controls the sleep transistor of the LUT.

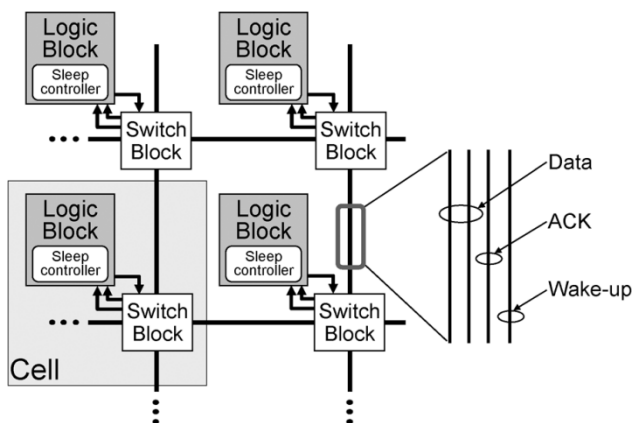


Fig. 2.1: Overall architecture of the proposed FPGA

The existing architecture requires four inputs, namely, two for a data, one for acknowledge (ACK), and one for wake-up. The wake-up signal is used to wake up the next LB in advance. Since the next LB has already been woken up before the data arrives, there is no penalty of the wake-

up time. Whether the LUT is used or not can be easily detected in an asynchronous architecture.

Fig. 2.2 demonstrates the principle of the activity detection using the asynchronous architecture. Each LB is assumed to operate a one-input and one-output function. As the initial state ( $t_0$ ), the phases of input and the output data of the LB are phase "0." If the new data arrives at the LB ( $t_1$ ), the phase of the input data changes from "0" to "1" and then the operation initiates.

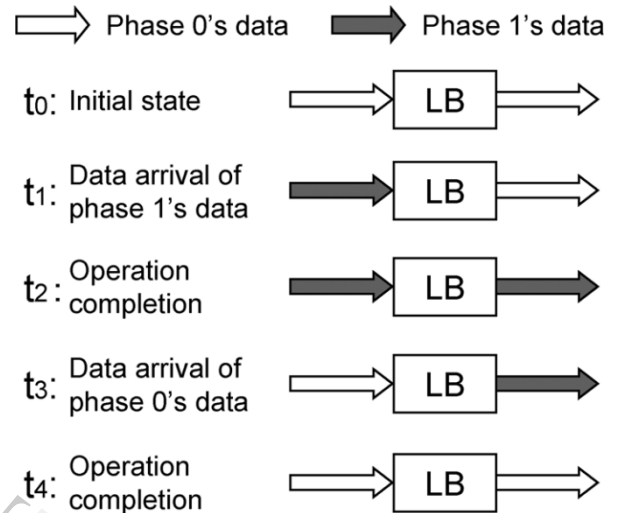


Fig. 2.2: Activity detection using the asynchronous architecture

When the operation is complete ( $t_2$ ), the phase of the output data changes from "0" to "1" as the same of the phase of the input data. Again, if the new data arrives at the LB ( $t_3$ ), the phase of the input data changes from "1" to "0" and then the operation initiates. When the operation is complete ( $t_4$ ), the phase of the output data changes again from "1" to "0" as the same of the phase of the input data. It is observed that when a new input data arrives at the LB, the phase of the input data is different from that of the output data. When the operation is complete, the phase of the input data is the same as that of the output data. Based on this operation, the activity of the LB can be detected just by comparing the phases of the input data and the output data. The activity information can be exploited to power OFF unused LBs and to wake them up. Therefore, the proposed sleep controller just extracts and compares the phases of input and output data. As a result, the area and power overheads of the proposed sleep controller are much smaller than that in synchronous architecture.

Fig. 2.3 shows the simple implementation of the autonomous fine-grain power gating. In this scheme, the sleep controller consists of XOR gate and a comparator. The XOR gate is used to extract the phases of the input and the output data. Then the comparator compares

whether the phases of the input and output are the same or not. The phases of the input and the output are different when LB is busy. Then the output of the comparator is “1.” If it is idle, the phases of the input and the output are the same. Then the output of the comparator is “0.” The output of the comparator is directly used as the control signal of the sleep transistor. In this method, the LB has two states: sleep and active. If the new input data arrives at the LB, the LB turns to the active state, and the sleep transistor turns ON to execute the operation. If the operation is complete, the LB turns to the sleep state, and the sleep transistor turns OFF to reduce the leakage current.

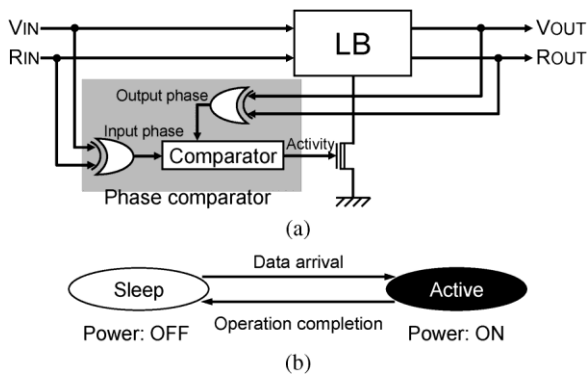


Fig. 2.3: Simple control strategy of autonomous power gating

At first, the wake-up time affects the delay time since the sleep transistor of the LB turns ON after the input data arrives. Then secondly the switching power may become larger than the saved power. This is due to the sleep transistor turns ON and OFF frequently when the input data comes frequently. To solve this problem, an efficient control strategy of the autonomous fine-grain power gating has been proposed.

### III. IMPLEMENTED ARCHITECTURE

#### A. Fundamental Principle of Autonomous Fine-Grain Power Gating

As shown in Fig. 3.1, the standby state is used to wake up the LB before the data arrives and power OFF the LB only when the data does not come for quite a while. The use of the standby state in between Sleep and active states has two major advantages. First, the wake-up time can be avoided since the LB has already been woken up when the data arrivals. Second, the dynamic power can be reduced since the number of the unnecessary switching of the sleep transistor is reduced.

Fig. 3.2 shows an illustration of the proposed power gating method using two LBs: LB1 and LB2 where LB1 is the “previous LB” of LB2. As shown in Fig. 3.2 (a), LB1 and LB2 are in the standby and sleep state as the

initial state respectively. To avoid the wastage of the wake-up time, LB1 is in either the standby state or the active state. As shown in Fig. 3.2 (b), when the new data arrives at “the previous LB” (LB1), a wake-up signal from LB1 is sent to LB2 to wake it up. Then, LB2 turns in to the standby state. As shown in Fig.3.2 (c), when the data arrives at LB2, LB2 turns in to the active state. In this state, the operation is executed immediately since the sleep transistor is woken up in the standby state. As shown in Fig. 3.2 (d), LB2 turns to the standby state because the operation of LB2 is complete. As shown in Fig. 3.2 (e), if no data arrives at LB2 during the threshold time, LB2 identifies that the data does not arrive for quite a while. Then, LB2 turns in to the sleep state and is powered OFF. The threshold time is determined such that the LB is not powered OFF in a busy condition where data arrives frequently.

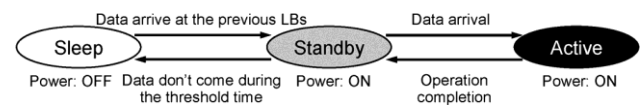


Fig 3.1: Control strategy of the implemented power gating technique

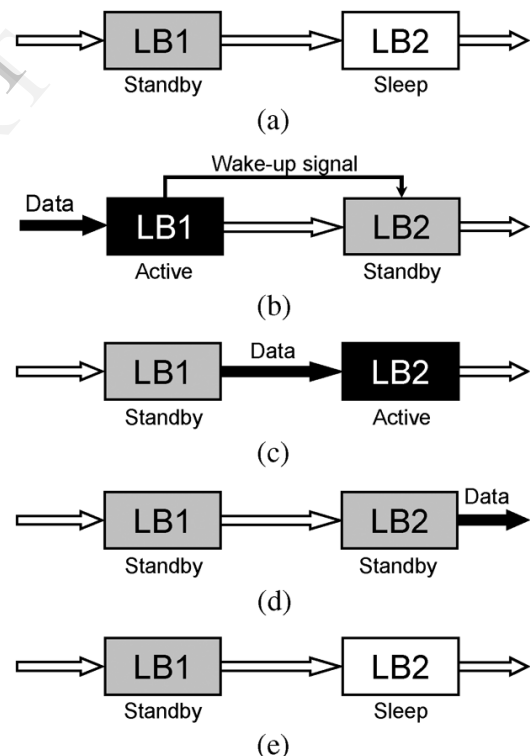


Fig. 3.2: Illustration of the implemented power gating technique

#### B. Circuit Implementation

Fig 3.3 shows the block diagram of a LB. Each LB mainly consists of a LUT, an output register, a sleep controller and a C-element. The LUT operates arbitrary with two-input and one output logic functions.

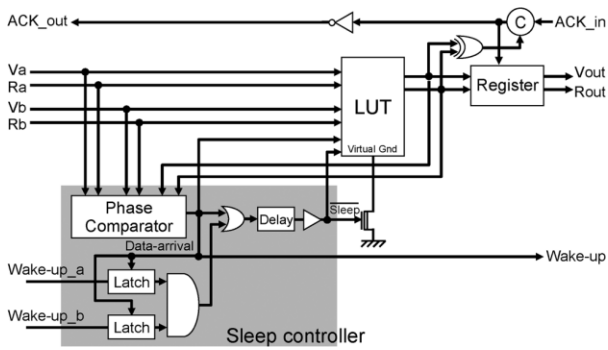


Fig. 3.3: Block Diagram of Logic Block

The gray region is the sleep controller. The Wake-up signals from previous LBs are used to wake up the LB before the new input data arrives. The Data-arrive signal is used to wake up the succeeded LB when the data arrives. The phase comparator detects the data arrivals. Two latches retain the Wake-up signals from previous LBs until all the input data arrive at the LB.

The programmable delay is used to delay the sleep signal by the predetermined threshold time in powering OFF the LB. There is no penalty of the wake-up time despite that the whole sleep controller is composed of small-size and high-threshold voltage transistors. This is because the LB gets ready to wake up when the data arrives at previous LBs. As a result, the area and power overheads are very small.

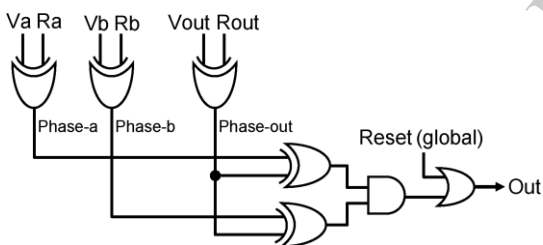


Fig. 3.4: Block diagram of a phase comparator

Fig. 3.4 shows the block diagram of a phase comparator for a two-input and one-output LB. The phase comparator is used to detect the data arrival. Phases of each data are extracted by XOR gates. If Phase-a and Phase-b are different from Phase-out, it means that all new data has arrived. In that case, the LB is active, and the output is “1.” Otherwise, it means that some data has not yet arrived and that the LB cannot start the operation. In that case, the LB is inactive, and the output is “0.”

Table I shows the truth table of the latch for the Wake-up signal. If the Wake-up once goes to “1,” the latch retains the signal until all data arrive at the LB. When all data arrive at the LB and no data arrives at the previous LBs, the output of the latch is reset to “0.”

Wake-up	Data-arrival	Out
0	0	No change
0	1	0
1	0	1
1	1	1

Table I: Truth Table of the Latch for the Wake-Up signal

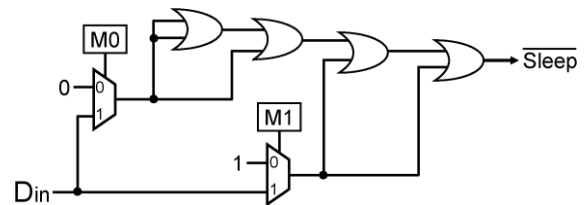


Fig. 3.5. Block diagram of the programmable delay.

Fig. 3.5 shows the block diagram of the programmable delay. As described in the last paragraph of Section III-B, if no data arrives at the LB during the predetermined threshold time, the LB predicts that the data does not arrive for quite a while. Then, the LB turns to the sleep state and is powered OFF. The programmable delay is used to power OFF an LB after it stays idle for the predetermined threshold time. Therefore, the function of the programmable delay is to delay the sleep signal by the predetermined threshold time in powering OFF. Note that the programmable delay does not delay the sleep signal in powering ON. The programmable delay consists of a series of OR gates and several memory bits. The memory bits are used to program the delay time.

M0	M1	Normalized threshold time
0	0	Not use power gating
0	1	0.5
1	0	Not defined
1	1	1

Table II: Relationship between the Memory Configuration and the Threshold Time of the Programmable Delay

Table II shows the relationship between the memory configuration and the threshold time. Let us consider a case when power gating is used. In powering ON the LB, turns from “0” to “1.” Since is used as an input of the last OR gate, of “1” makes the output of the last OR gate to “1” immediately. In powering OFF the LB, turns from “1” to “0.” The value “0” of propagates through the series of OR gates so that the sleep signal is delayed. The use of more OR gates and more memory bits make it possible to increase the number of choices of the delay time.



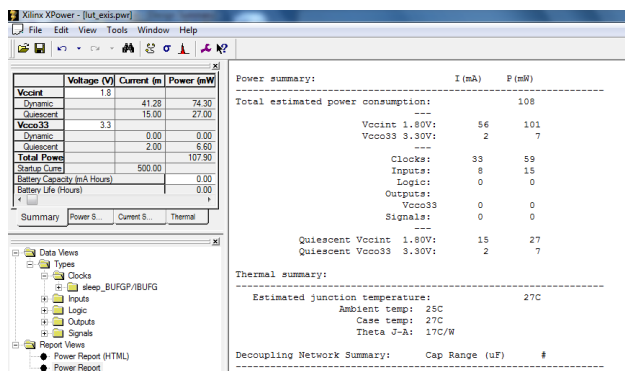


Fig.4.3: Power Report for the Existing Design

Comparison between the implemented design and previously existing design show that the implemented design can achieve better power results. The power consumption is reduced by 48% compared to existing designs.

## V.CONCLUSION

This paper proposed an asynchronous FPGA based on autonomous fine-grain power gating with small overheads. In asynchronous architecture, the activity of an LB is easily detected only by comparing the phases of the input and the output data. To implement the autonomous fine-grain power gating efficiently, the standby state is used to wake up the LB before the data arrives and power OFF the LB only when the data does not come for quite a while. As a result, the wake-up time can be hidden and the dynamic power of unnecessary switching of the sleep transistor can be saved.

## ACKNOWLEDGMENT

Durga Prasad would like to thank Mr. C. Ravi Shankar Reddy, Lecturer, who had been guiding through out to complete the work successfully, and would also like to thank the HOD, ECE Department and other Professors for extending their help & support in giving technical ideas about the paper and motivating to complete the work effectively & successfully.

## REFERENCES

- [1]. H. Z. V. George and J. Rabaey, "The design of a low energy FPGA," in *Proc. Int. Symp. Low Power Electron. Des.*, CA, Aug. 1999, pp.188–193.
- [2]. Synplicity Inc., Sunnyvale, CA, "Gated clock conversion with Synplicity's synthesis products," Jul. 2003.
- [3]. Xilinx Inc., San Jose, CA, "Synthesis and simulation design guide,"2008.[Online].Available: <http://www.xilinx.com/itp/xilinx10/books/docs/sim/sim.pdf>

[4]. Y. Zhang, J. Roivainen, and A. Mammela, "Clock-gating in FPGAs: A novel and comparative evaluation," in *Proc. EUROMICRO Conf. Digit. Syst. Des.*, 2006, pp. 584–590.

[5]. T. Tuan, S. Kao, A. Rahman, S. Das, and S. Trimberger, "A 90 nm low-power FPGA for battery-powered applications," in *Proc. FPGA*, Feb. 2006, pp. 22–24.

[6]. Xilinx Inc., San Jose, CA, "Spartan-2 FPGA family datasheet,"2009.

[Online]. Available: <http://www.xilinx.com>

[7]. Xilinx Inc., San Jose, CA, "Virtex-4 FPGA family datasheet," 2009.

[Online]. Available: <http://www.xilinx.com>

[8]. M. Keating, D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual: For System-on-Chip Design*. New York: Springer, 2007.

[9]. A. Rahman, S. Das, T. Tuan, and S. Trimberger, "Determination of power gating granularity for FPGA fabric," in *Proc. IEEE Custom Intergr. Circuits Conf. (CICC)*, 2006, pp. 9–12.

[10]. M. Hariyama, S. Ishihara, and M. Kameyama, "Evaluation of a field-programmable VLSI based on an asynchronous bit-serial architecture," *IEICE Trans. Electron.*, vol. E91-C, no. 9, pp. 1419–1426, 2008.

[11]. S. Ishihara, M. Hariyama, and M. Kameyama, "A low-power FPGA based on autonomous fine-grain power-gating," in *Proc. Asia South Pacific Des. Autom. Conf. (ASP-DAC)*, Yokohama, Japan, Jan. 2009, pp. 119–120.

[12]. M. Hariyama, S. Ishihara, C. C. Wei, and M. Kameyama, "A field-programmable VLSI based on an asynchronous bit-serial architecture," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Jeju, Korea, Nov. 2007, pp. 380–383.