

# Proposed Low-Power FPGA Architecture Using an Autonomous Fine-Grain Power Gating

Kiruthiga M, Prakasam P and L.M.I Leo Joseph

**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 using the proposed fine-grain power gating method, the FPGA consumes only 34  $\mu$ W.

**Index Terms:** FPGA, Power Gating, Logic Block, Lookup Table

## I. INTRODUCTION

Due to the development of IC technology, FPGAs play the major role on an 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.

**Revised Manuscript Received on 30 June 2012.**

\* Correspondence Author

**Kiruthiga M\***, PG Scholar VLSI Design, Department of Electronics and Communication Engineering, MIET Engineering College, Trichy, Tamilnadu, India

**Dr. Prakasam P**, Professor, Department of Electronics and Communication Engineering, Tagore Institute of Engineering and Technology, Salem, Tamilnadu, India.

**Prof. L.M.I Leo Joseph**, Associate Professor, Department of Electronics and Communication Engineering, MIET Engineering College, Trichy, Tamilnadu, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

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.

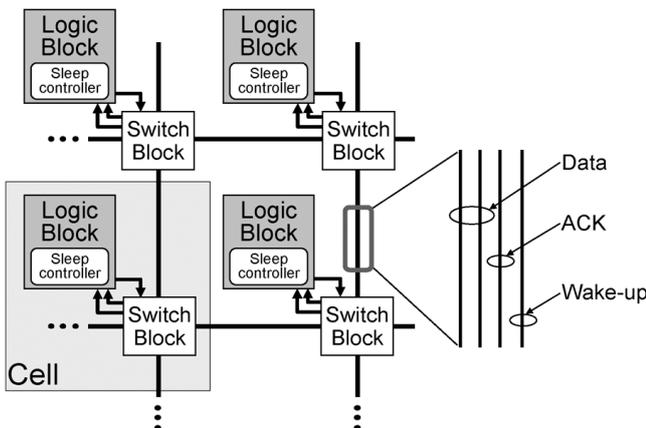
In fine-grain power gating, each LUT has its sleep controller, the number of the sleep controllers is much larger than that of coarse-grain power gating. In synchronous architectures, the sleep controller consists of some memory bits to store the sleep time. The sleep controller is always running in 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. Proposed 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. PROPOSED ARCHITECTURE**

Fig. 2.1 shows the overall architecture of the proposed 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.

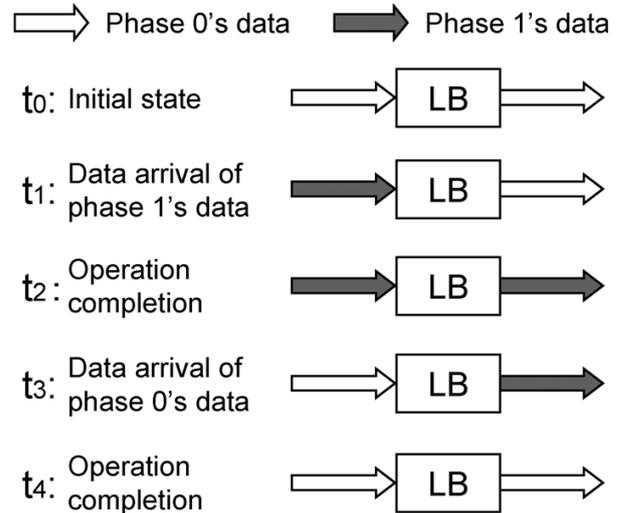


**Fig. 2.1 Overall architecture of the proposed FPGA**

The proposed 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.

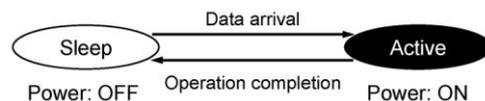
**A. Fundamental Principle of Autonomous Fine-Grain Power Gating**

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 with 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 Simple control strategy of autonomous power gating**

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.

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.

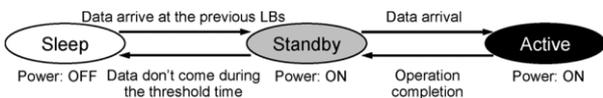


Fig 2.4 Control strategy of the proposed power gating method

As shown in Fig. 2.4, 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 arrives. Second, the dynamic power can be reduced since the number of the unnecessary switching of the sleep transistor is reduced.

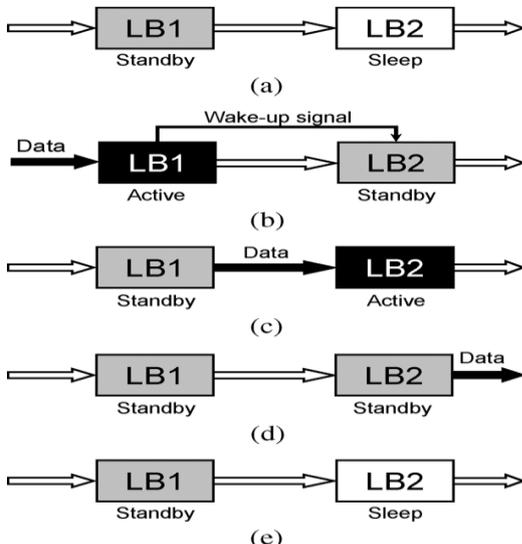


Fig. 2.5. Illustration of the proposed power gating method

Fig. 2.5 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. 2.5(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. 2.5(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. 2.5(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. 2.5(d), LB2 turns to the standby state because the operation of LB2 is complete. As shown in Fig. 2.5(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.

**B. Logic Block**

Fig 2.6 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 arbitrarily with two-input and one output logic functions.

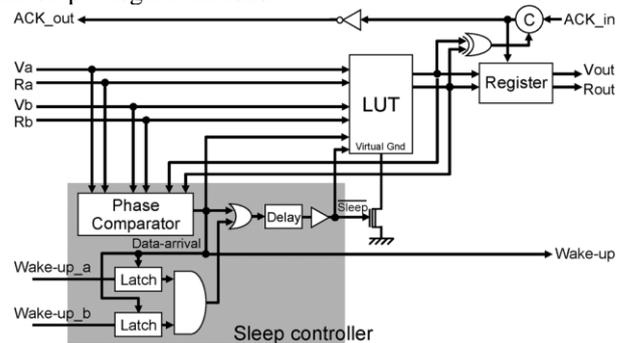


Fig. 2.6 Block Diagram of Logic Block

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.

**C. Design of Proposed Compact LUT**

In the proposed FPGA, the major consideration is designing efficient LB utilizing a compact LUT. Fig. 2.5 shows the block diagram of the proposed LUT, which consists of four sub-modules. Each sub-module is the combination of a decoder, a multiplexer and a memory bit. The decoders exclude invalid input patterns with different phases and only valid data are fed to the multiplexer.

As a result, the number of multiplexer is reduced, and the transistor count is also reduced by 36% compared to the conventional multiplexer type LUT.

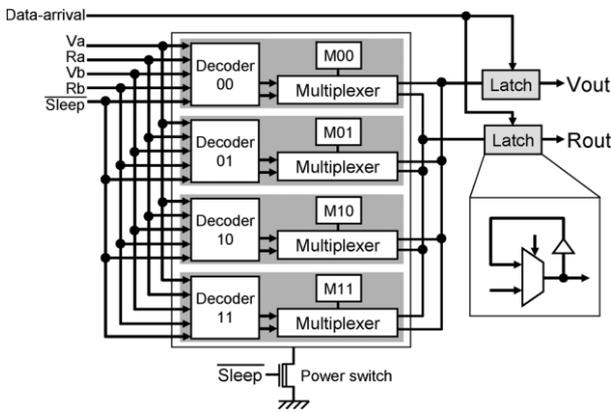


Fig. 2.5 Block diagram of the proposed LUT

III. RESULT AND DISCUSSION

The developed models can be analyzed for functional correctness using a top-down design methodology and starting from a high level description at the system/algorithm level. The detailed models can be generated by increasing the description details considering the hardware implementation aspects. The RTL (Register Transfer Level) code is expected to provide better model for synthesis. The functionally correct code, describing the Entities and Architectures, may then be simulated for verification and synthesized into actual hardware. There are various software tools that support design of individual components and then integration into the system to verify the design using simulation. The synthesis involves analyzing the VHDL code, synthesizing for the target architecture, optimizing subject to design constraints such as placement directives or delay specifications, and

generating an optimized FPGA netlist. Placement and routing tools generate an optimal placement subject to delay constraints and then interconnect the logic using the available routing resources on the particular FPGA. A bit file containing FPGA configuration data that can be downloaded onto the chip is finally generated.

The simulation and synthesis of the model has been carryout using and Xilinx ISE foundation series 10.1 as per the design environment discussed. The designed model has been simulated and synthesized using Xilinx Virtex-5vfx200tff1738 FPGA processor. The speed grade for the above mentioned target device is -2. The results obtained during the each process are explained in this section.

A. Outcome of Simulation

The developed model has been simulated using Xilinx ISE Simulator for various combinations of input data. The simulation result of the proposed LB is shown in Fig 2.6. The simulation result shows that the LB is in active state when the data arrives at the wakeup signal. If the data arrives also the comparator output state is changing from “0” to ‘1”. This means it detects the data arrival. This has been repeated for various combinations of data and the simulated result coincides with the predicted result.

B. Outcome of Synthesis

The first stage of the synthesis is to analyze the generated VHDL code to check the compatibility for synthesizing. After analyzing the source code, the target device has been synthesized and the netlist has been created. The device utilization summary for the target architecture of 5vfx200tff1738 has been obtained and it has been illustrated in Table I.

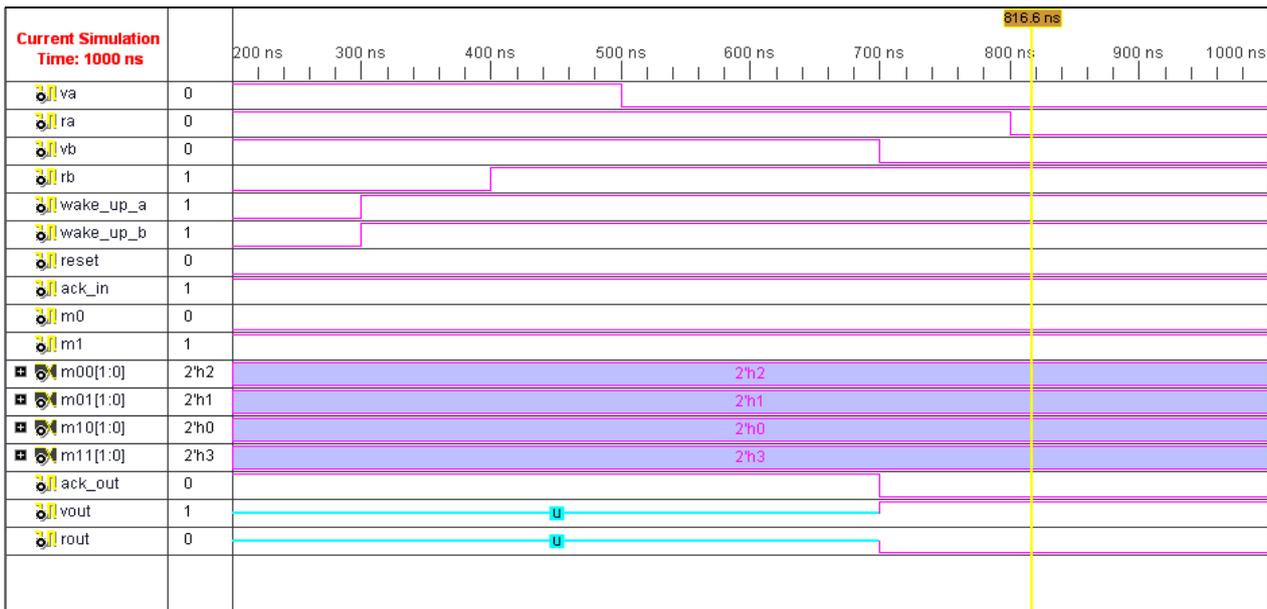


Fig 2.6 Simulation result of the LB

Table I Device utilization summary after synthesis

Name of blocks	Available	Utilized	% of Utilization
Number of Slice	7	122880	0%
Number of Slice LUTs	16	122880	0%
Number of bonded IOBs	20	960	2%
Number with an unused LUT	1	17	5%
Number of fully used LUT-FF pairs	6	17	35
Number of GCLKs	4	1	25%

It is seen from the Table I that the Logic Block utilizing the developed LUT model 2% of bonded IOBs. The device utilization clearly indicates that the selected target device 5vfx200tff1738 is well suited to implement the developed model. The summary of the timing report during the synthesis process is illustrated in Table II.

Table II. Timing report for 5vfx200tff1738 device

Parameter	Duration (ns)
Minimum period	1.112
Minimum input arrival time before clock	3.070
Maximum output required time after clock	5.602
Maximum combinational path delay	6.781

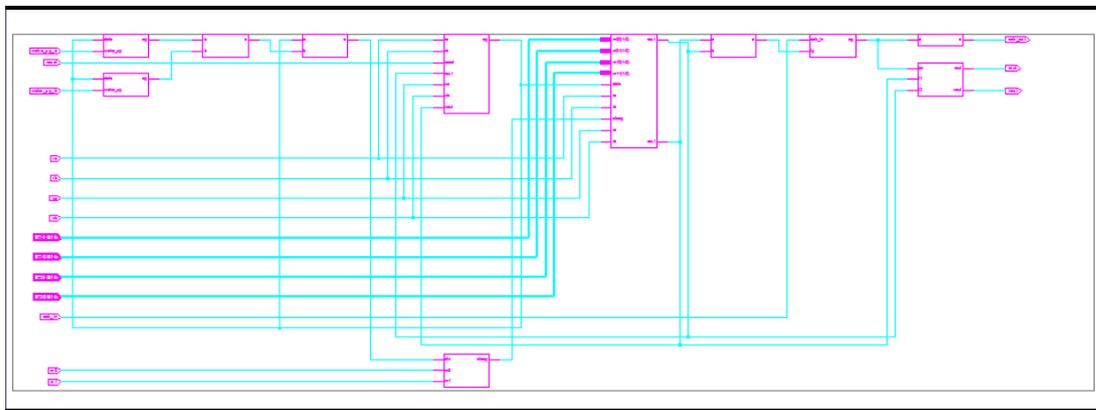
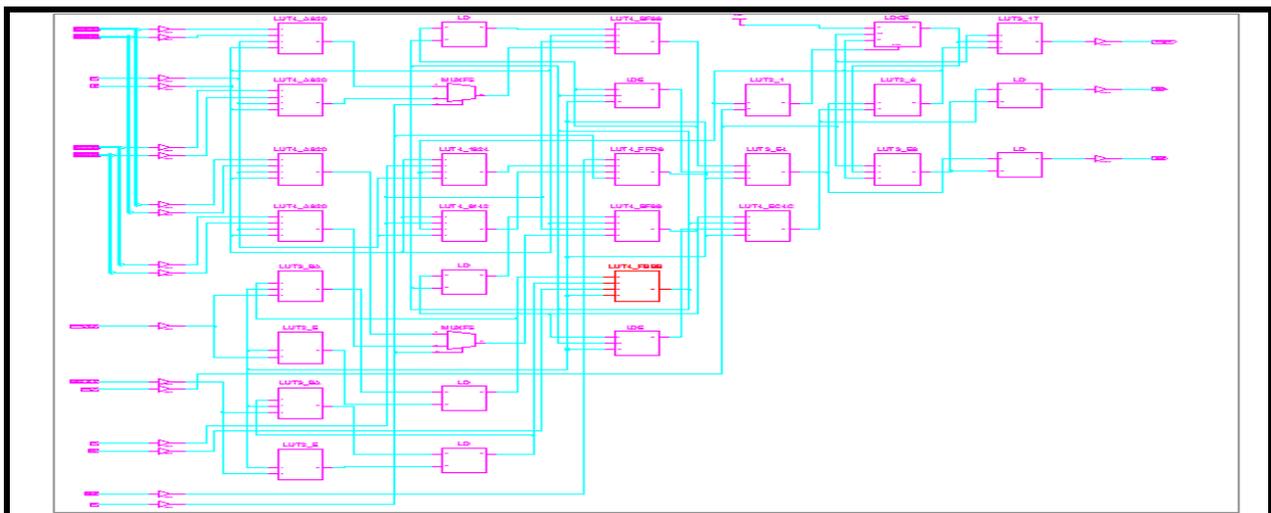


Fig 2.7 Technology level schematic for the developed model



The technology level schematic for the generated model and the gate level schematic for a single block are shown in Fig 2.7 and 2.8 respectively.

The technology level schematic is found to have higher complexity and can be reduced using the gate level schematic for the individual block. Hence the gate level schematic has been generated and analyzed for the all block. The performance of the synthesis has been analyzed after the gate level schematic analysis is carried out for entire block.

C. Power utilization

The power required to implement the LB after incorporating the developed fine-grain power gating LUT using the target device 5vfx200tff1738 has been computed and listed in Table III for individual components. It is found from the Table 3.3 that the total power required to implement the proposed module in Virtex-5vfx200tff1738 is 34 uW.

Table III Power required for the target device

Name	Power (uW)
Total $V_{ccint}$	10
Total $V_{ccaux}$	20
Total $V_{cco}$	4
<b>Total Power</b>	<b>34</b>

IV. CONCLUSION

In this paper, an asynchronous FPGA architecture based on autonomous fine-grain power gating with small overheads has been developed and analyzed. The implementation of the autonomous fine-grain power gating has been done efficiently using the standby state 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 has been reduced to 1.112 ns. Since their data paths change dynamically and frequently, it is more difficult than FPGAs to determine the control parameters for each LB using offline analysis. The power analysis also has been carried out and it has been found that using the proposed fine-grain power gating method, the FPGA consumes 34 uW.

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-3 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 Integr. Circuits Conf. (CICC)*, 2006, pp. 9–12.
10. M. Hariyama, S. Ishihara, and M. Kameyama, "A low-power field-programmable VLSI based on a fine-grained power-gating scheme," in *Proc. IEEE Int. Midw. Symp. Circuits Syst. (MWSCAS)*, Knoxville, Aug. 2008, pp. 430–433.
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.

AUTHOR PROFILE



**Kiruthiga. M** has obtained her BE degree in Electronics and Communication Engineering from Anna University in 2008. Currently he is doing his ME in VLSI Design at MIET Engineering College, Trichy. Presently she is involving in developing a automated fine-grain power gating model for effective FPGA. She has published more than three research papers in national and international conferences/journals. Her special areas of interest are Low power VLSI Design, Application of Low power VLSI in Mobile Communication systems.



**Prakasam. P** has obtained his BE degree in Electronics and Communication Engineering from Madras University in 1994. He received his M.Tech degree in Advanced Communication Systems from Sastra University, Tanjore, India in 2002. He obtained his Ph.D from Anna University Chennai, India in 2010 in the field of Signal Processing in Communication Systems.

At present he is a Principal of Tagore Institute of Engineering and Technology, Attur, India. He has authored over twenty research publications in international and national journals and conferences. His special areas of interest are Signal Processing, Communication Systems and Applications of signal processing in Mobile Communication Systems.

Mr.P. Prakasam is a member of IEEE (USA), life member of ISTE, IACSIT, IAENG and VSI (India). He is named in Marquis Who's Who in the world in the year 2009. He has also coordinated Conferences, AICTE and ISTE sponsored SDP and STTP programme organized by the Department.



**L.MILLEO JOSEPH** has obtained his BE Degree in Electronics and Communication Engineering from Bharathidasan University, Trichy. He received his ME Degree from Satyabama University, Chennai. Currently he is doing his Ph.D at Satyabama University, Chennai. He is having more than 12 years of teaching experience in various Engineering colleges. He published more than eight research papers in various national and international conferences/journals.

At present he is working as Associate Professor in the Department of Electronics and Communication Engineering in M.I.E.T. Engineering College. He has participated in various national level workshops and seminars at various colleges.

