ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

A Low Power Asynchronous FPGA with Autonomous Fine Grain Power Gating and LEDR Encoding

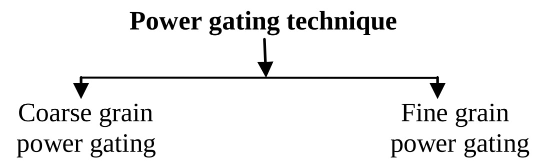
N.Rajagopala krishnan, k.Sivasuparamanyan, G.Ramadoss

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. In this project designing a reconfigurable low power Asynchronous FPGA cells are done. FPGAs consume high dynamic and standby power. In order to reduce the standby power the autonomous fine grain power gating method is used. The autonomous fine grain power gating method has lookup table which is controlled by sleep controller and sleep transistor. In the successive logic block if the data arrives to the first logic block, the next logic block goes to active state. Suppose if the data not arrives to the first logic block than the next logic block goes to standby state and remain in this state until it reaches threshold time. After reaching threshold time the logic block goes to sleep state from standby state. In this sleep state the logic block power goes OFF. Hence the power consumption of the FPGA becomes reduced. The proposed architecture used in both fine grain as well as coarse grain structure. The circuit is simulated using Xilinx tool. Power reduction is achieved by selectively setting the functional units into a low leakage mode when they are inactive.**

Index Terms— **Asynchronous Field Programmable Gate Array (FPGA); Power Gating; Level Encoded Dual Rail (LEDR) Encoding; Logic Block; Lookup Table,; sleep controller.**

I. Introduction

In FPGA design, the clock gating and power gating is important work. To implement clock gating, circulation is employed. The idea of circulation is to retain the contents of the flip-flop in the sleep state. Circulation can reduce the dynamic power consumption of registers and the gates in the fan-out of the registers. However, the standby power consumption of the clock network cannot be reduced. The standby power is a serious problem because it has an enormously large number of transistors to achieve its programmability. Low-cost FPGAs consume up to hundreds of milliwatts power. Power gating has emerged as the most effective design technique to achieve low standby power. Power gating techniques are based on selectively setting the functional units into a low leakage mode when they are inactive. Power gating technique designed by the following methods 1) Sleep controller 2) Sleep transistor 3) Sleep signal distribution network [1],[5]&[6].



A. Coarse grain power gating

Large number of lookup tables grouped and shared in single sleep controler. In FPGA all the lookup tables are not active state in same time. Depends on the progrose number of lookup tables are active state. If any of the lookup table active state in the group other lookup tabes does not goes to sleep state, so In this techniqe only consume little power consumetion[3].

B. Fine grain power gating

Over come the problems of coarse grain power gating we introduce the Fine grain power gating technique. In fine grain power gating technique each look up table having own sleep controller and related to sleep transistor, so any of the lookup table active states all other lookup table are goes to sleep state[1]. In this paper reduce the both standby power and dynamic power.

II. Related Work

A. Asynchronous architecture design

The asynchronous architecture it detects the activity of a power gated domain. The activities are 1) To determine when logic block is standby state, when sleep state & when active state. 2) It compares the phase of the input data and output data 3) It determine the function of lookup table. Dynamic power reducing purpose introduce dual rail encoding (existing)[2]& level encoding dual rail (proposed)architecture. Standby power reducing purpose introduced autonomous fine grain power gating technique.

Manuscript received Feb, 2013.

Mr ***N.Rajagoplakrishnan****, Electronics and Cionommunication Engineering, K.S.Rangasamy College of Technology, Tiruchengode, India, 9524470644.*

Mr **k.Sivasuparamanyan**, *Electronics and Cionommunication Engineering, K.S.Rangasamy College of Technology., Tiruchengode, India, 9443909123.*

Mr **G.Ramadoess,** *Electronics and Cionommunication Engineering, K.S.Rangasamy College of Technology, (Tiruchengode, India,9047238325.*

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

1) *Dual Rail Encoding*

They use four-phase dual-rail encoding because of relatively small hardware cost. in four-phase dual-rail encoding, a spacer must be inserted between two consecutive valid data values. This results in low throughput and high dynamic power consumption because of the large number of signal transitions.

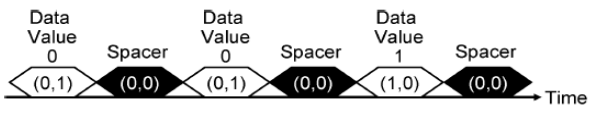


Figure 2.1 Dual Rail Encoding Data Transmission

2) *Level Encoding Dual Rail*

Proposed asynchronous FPGAs based on LEDR encoding. LEDR is one of several two-phase dual-rail encodings. In LEDR encoding, no spacer is required. This results in high throughput and low dynamic power consumption because of the number of signal transitions reduced by half.[3],[4].

3) *Autonomous Fine Grain Power Gating*

The proposed autonomous fine grain power gating shown Figure 2.2

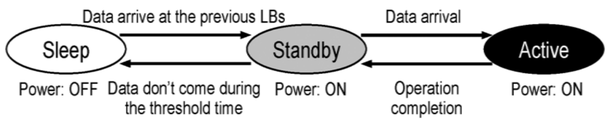


Figure 2.2 Control strategy of the proposed power gating method

To solve this problem, we propose an efficient control strategy of the autonomous fine-grain power gating. The standby state is used to do the following:
a) wake up the LB before the data arrives
b) power OFF the LB only when the data does not come for quite a while.

The use of the standby state has two major advantages. First, the wake-up time can be hidden since the LB has already been woken up when the data arrivals. Second, the dynamic power can be saved since the number of the unnecessary switching of the sleep transistor is reduced

4) *Wave-Pipelining For Bit-Serial FPGAs*

In FPGAs, area for routing is dominant. To reduce the routing area without performance degradation, wave-pipelining is combined with bit-serial architecture. To reduce the area in FPGAs, reducing the complexity of interconnect using bit-serial architecture is efficient. However, bit-serial data transmission decreases the throughput. To achieve a both a simple interconnect and high throughput, bit-serial wave-pipelining FPGA architectures have been proposed.

III. ARCHITECTURE DESIGN

The overall architecture proposed in reference [3]. The block diagram of the proposed FPGA shown below Figure 3.1 Mesh controlled cellular array based on bit serial architecture used. Each logic block mainly consists of a look up table, an output register and a sleep controllers. The look up table operates arbitrary of two input and one output logic function. The registers store the data value and produce the output to switch block. Sleep controller monitor wake up the successive block when it gets data. The switch block consists of pass transistor switches. In a switch block, a wire-set consists of four wires: two for data lines (V_{out} and R_{out}), one for the acknowledge signal and one for the wake-up signal. A pass-switch block consists of four pass switches and a single memory bit. This FPGA architecture logic block can be connected to the switch block. In the switch block there are four signals 1) data signal (first bit) 2) data signal (second bit) 3) acknowledgement signal 4) data arrival signal. The above four signal acknowledgement signal and data arrival signal connected to pervious logic block.

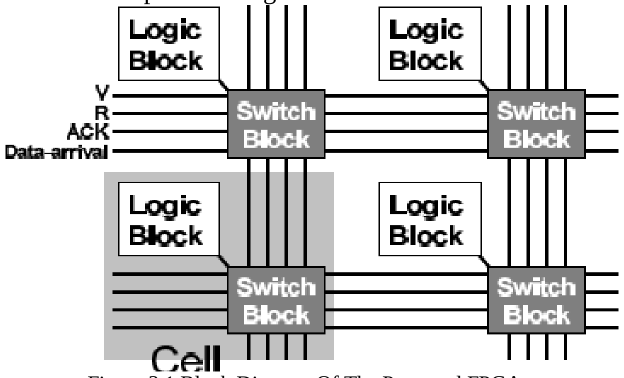


Figure 3.1 Block Diagram Of The Proposed FPGA

The two pass switches are used for the four wires of the wire-set, one V_a, R_a, ack and wakeup signal wires respectively. The pass switches are controlled by the same memory bits.

A. *LEDR Encoding design*

The design of LEDR encoding proposed in reference [3],[4].

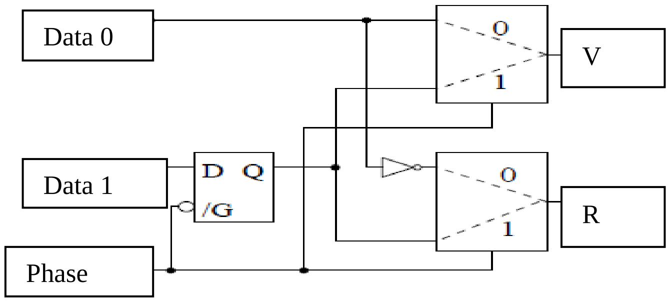


Figure 3.2 Level Encoding Dual Rail Architecture

This architecture operation based on data's and phase signal. The D latch signal given to selectors and selectors produce the output of level encoding signal.

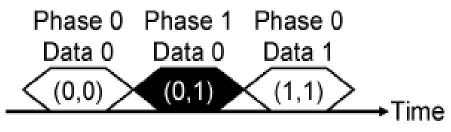


Figure 3.3 LEDR Encoding Data Transmission

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

In LEDR encoding, no spacer is required. Table 3.1 shows the code table of LEDR encoding. In LEDR encoding, each data value has two types of code words with different phases. Fig. 3.3 shows the example where data values “0,” “0,” and “1” are transferred. The main feature is that the sender sends data values alternately in phase 0 and phase 1. Because no spacer is required, the number of signal transitions is half of four-phase dual-rail encoding. As a result, the throughput is high and the power consumption is small. Based on this observation, in the proposed FPGA, LEDR encoding is employed for implementing the asynchronous architecture to reduce the dynamic power.

TABLE 3.1 Code Table of LEDR Encoding

		Code word (V, R)
Phase 0	Data 0	(0,0)
	Data 1	(1,1)
Phase 1	Data 0	(0,1)
	Data 1	(1,0)

B. Circuit Implementation

The circuit can be implemented by using below the architectures. The logic block can be connected to the switch block. Each switch block connected to the other switch block. This logic block and switch block described below the architecture diagrams.

1) Logic Block Design

The overall structure of logic block shown Figure 3.2

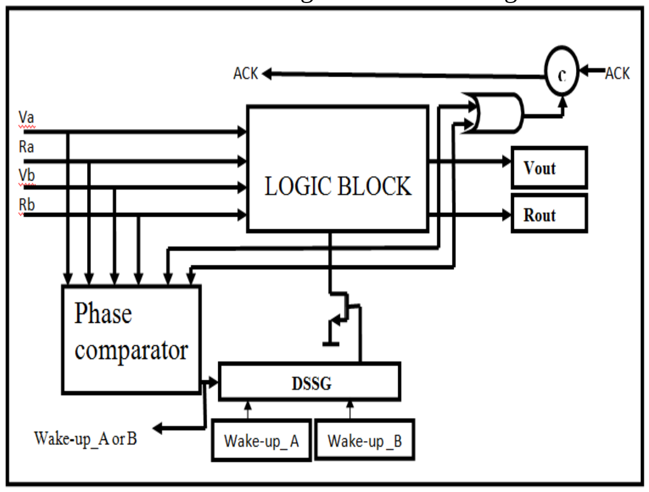


Figure 3.4 Block Diagram of Logic Blok

In this logic block contains lookup table, sleep controller, registers and programmable delay elements presented. The description of the logic block described below.

a) Lookup Table Design

This architecture contains four sub modules. Each sub modules consist of a decoder, a multiplexer and memory bits. The decoder designed by two four input AND gates. The output of the decoder is given to the multiplexer. The multiplexer is designed by four pass transistor logic and one inverter logic. The decoders exclude invalid input patterns with different phases. The valid data are fed to the multiplexer. As a result, the numbers of multiplexers are reduced and the transistor count is reduced compared to the multiplexer type LOOKUP TABLE. If the combination of inputs are invalid (i.e., if the two inputs have the different phases) all pass-transistors turn OFF according to the output of the decoder. The decoder and multiplexer based lookup table shown Figure 3.3. The reference [4] .

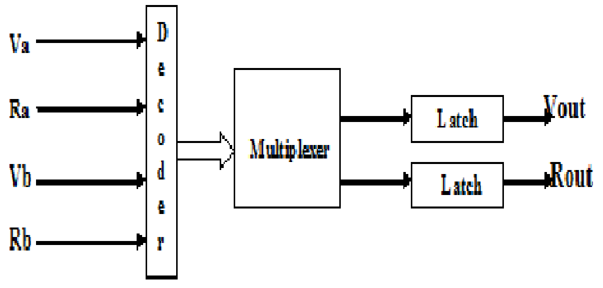


Figure 3.5 Block Diagram of The Proposed Lookup Table

The detailed structures of the decoder and multiplexer based lookup table showed below Figure 3.4. In that diagram a AND gates are used for decoder and pass transistor used for multiplexer logic.

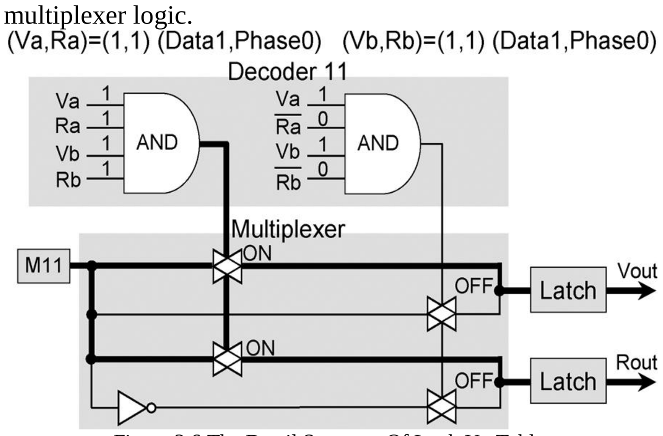
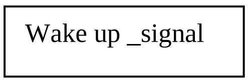


Figure 3.6 The Detail Structure Of Look Up Table

The previous outputs stored in latch, if input patterns are valid (i.e., if the two inputs have the same phase), according to the corresponding pass-transistors turn ON. The value of the memory bit is selected as outputs; the outputs are stored in the latches.

b) Sleep Signal Generation Design

It contains phase comparator, latch and programmable delay design. It is discussed below Figure 3.5. The sleep controller design designed reference [2].



ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 2, Issue 4, April 2013

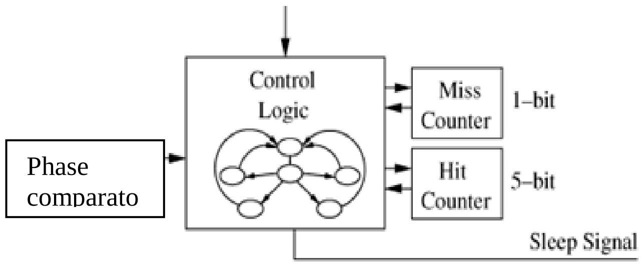


Figure 3.7 Block Diagram of The Sleep Controller Design

The each block in the sleep controller described in below designs.

c) Sleep Signal Control Logic Design

1) Idle

In this state, the Dynamic Sleep Signal Generation is idle while the functional unit is being used.

2) Threshold Wait

This state is initiated when the functional unit idles. In this state, the Dynamic Sleep Signal Generation counts the number of clock cycles since the functional unit idled and compares them to the threshold. As soon as the threshold is reached, the Dynamic Sleep Signal Generation raises the sleep signal to high. The first two states of the Dynamic Sleep Signal Generation resemble the expected operation of the Static Sleep Signal Generation

3) BEP Hit

This state is reached when the functional unit stays idle longer than the breakeven point. Whenever the Dynamic Sleep Signal Generation reaches this state, the total leakage savings exceed the overheads. These overheads include the dynamic power consumed by both the Dynamic Sleep Signal Generation and sleep transistor network.

4) BEP Hit Limit

A small counter is attached to the Dynamic Sleep Signal Generation to count the number of times the Break Even Point Hit state is reached. When the Break Even Point Hit is reached a predefined number of times denoted by hit-limit the Dynamic Sleep Signal Generation reduces the threshold by a step equal to step-.

5) BEP Miss

This state is reached if the idle period is longer than the threshold but shorter than the break-even point.

6) BEP Miss Limit

Similar to the Break Even Point Hit Limit, the Dynamic Sleep Signal Generation keeps track of the number of Break Even Point misses. When they reach a predefined value denoted by miss-limit, the Dynamic Sleep Signal Generation increases the threshold by a step that is equal to step+

d) Sleep controller design

1) Phase Commparatear Design

The block diagram of a phase comparator for a two-input and one-output LOGIC BLOCK shown Figure 3.6. 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 LOGIC BLOCK is active, and the output is '1'. Otherwise, it means that some data has not yet arrived and that the LOGIC BLOCK cannot start the operation. In that case, the LOGIC BLOCK is inactive, and the output is '0'.

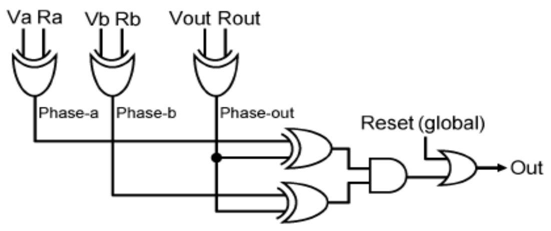


Figure 3.8 Block Diagram Of The Phase Commparatear Design

The phase comparator output and pervious logic block wakeup signal given to the programmable delay.

2) Latch Design

The latch design truth table showed table 3.1. If the Wake-up signal once goes to '1', the latch retains the signal until all data arrive at the LOGIC BLOCK. When all data arrive at the LOGIC BLOCK and no data arrives at the previous Logic Blocks, the output of the latch is reset to '0'.

TABLE 3.1 Truth Table of the Latch For The Wake-Up Signal

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

3) Architecture Design by Using Fine Grain as well as Coarse Grain Structure

The power gating methods fine grain unit and coarse grain unit used in same architecture. So, proposed FPGA reduced area and standby power in same architecture. It employs an island-style fine-grained FPGA structure with dedicated columns for coarse-grained units. Both fine grained and coarse-grained units are reconfigurable. The coarse grained part contains embedded fixed-function operation. The connection between coarse-grained units and fine-grained units is similar to the connection between embedded blocks (embedded multiplier, DSP blocks or blocks RAM).

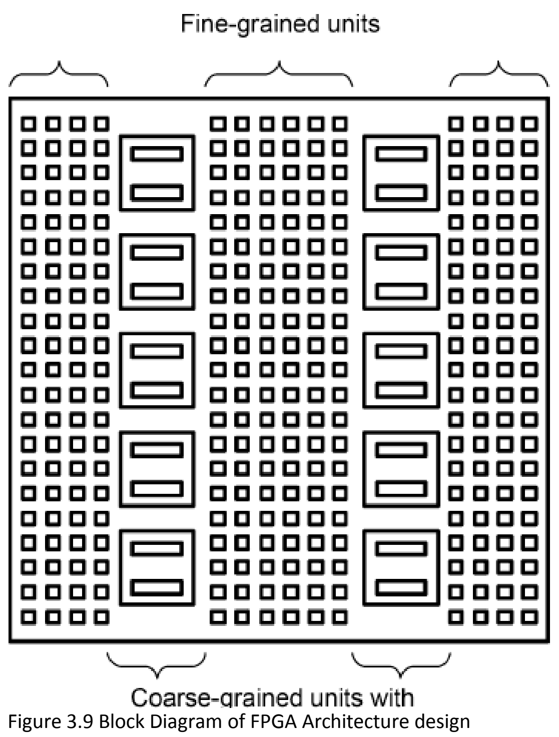


Figure 3.9 Block Diagram of FPGA Architecture design

4) FPGA with Full Adder *operation*

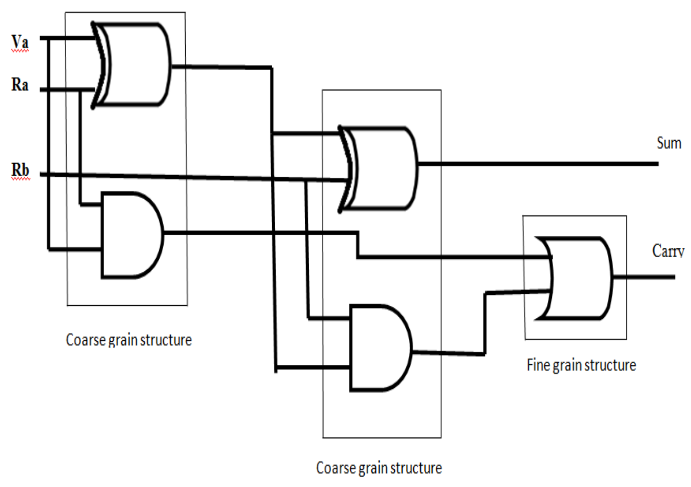


Figure 3.10 Block Diagram of the Logic Block with Full Adder operation

The full adder circuit having there are two Xor gate and two, gate and one or gate. The inputs are a, b & c, the outputs are sum and carry. The above FPGA IC two perform the full adder operation means to take the five logic gates, on the time the current five logic blocks are active state all other logic blocks are sleep state. The sleep signals connected to the succeed logic block of sleep controller. the each logic block having own sleep controller, so the logic block arrive at the input the sleep controller also start work and output is given to the next logic block. The next logic blocks also active state. This method power easily minimized. The sum output of first logic block and carry output of first logic block are designed by using coarse grain structure and sum of output second logic block and carry output of second logic block are designed by using another coarse grain structure and carry output of last logic block is designed by using fine grain structure.

IV. 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 net list. 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. 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

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 Logic block is shown in Figure 4.1.

a) *Logic Block with Sleep Controller*

The simulation result shows that the Logic block with sleep controller 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. The sleep signal when it's one the next logic block goes to active state. The sleep signal when it's zero the next logic block goes to standby state up to delay time after goes to sleep state.

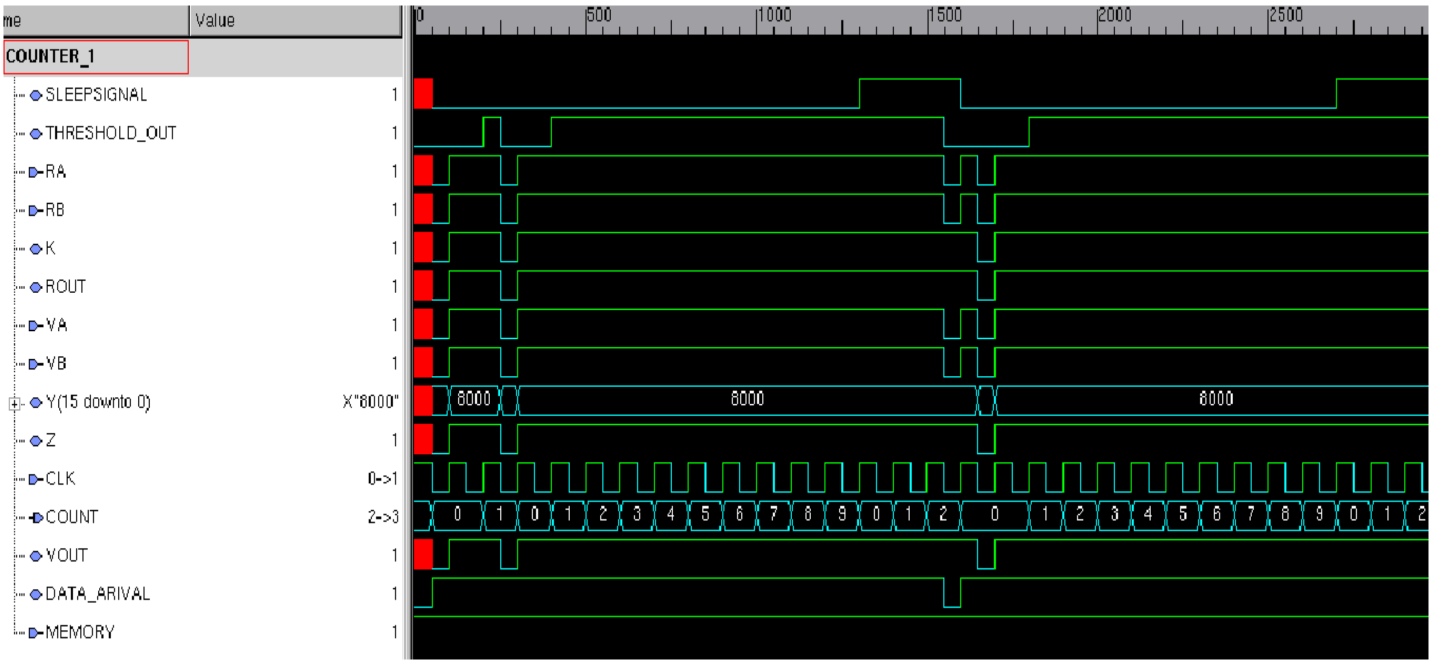


Figure 4.1 The Simulation Result For Logic Block Design With sleep controller

b) Full Adder Simulation Result

The full adder logic block showed Figure 3.8. in the full adder circuit there are two Xor gate and two AND gate and one OR gate are used to perform the full adder. The corresponding logic blocks are active state all other logic blocks are sleep state. The first logic block worked on Xor gate that time the same logic block generate the sleep signal is zero. The output of the first logic block and third input given to the next logic block that time the second logic block worked on Xor operation and sleep signal is one it's given to the third logic block, so the third logic block goes to sleep state.

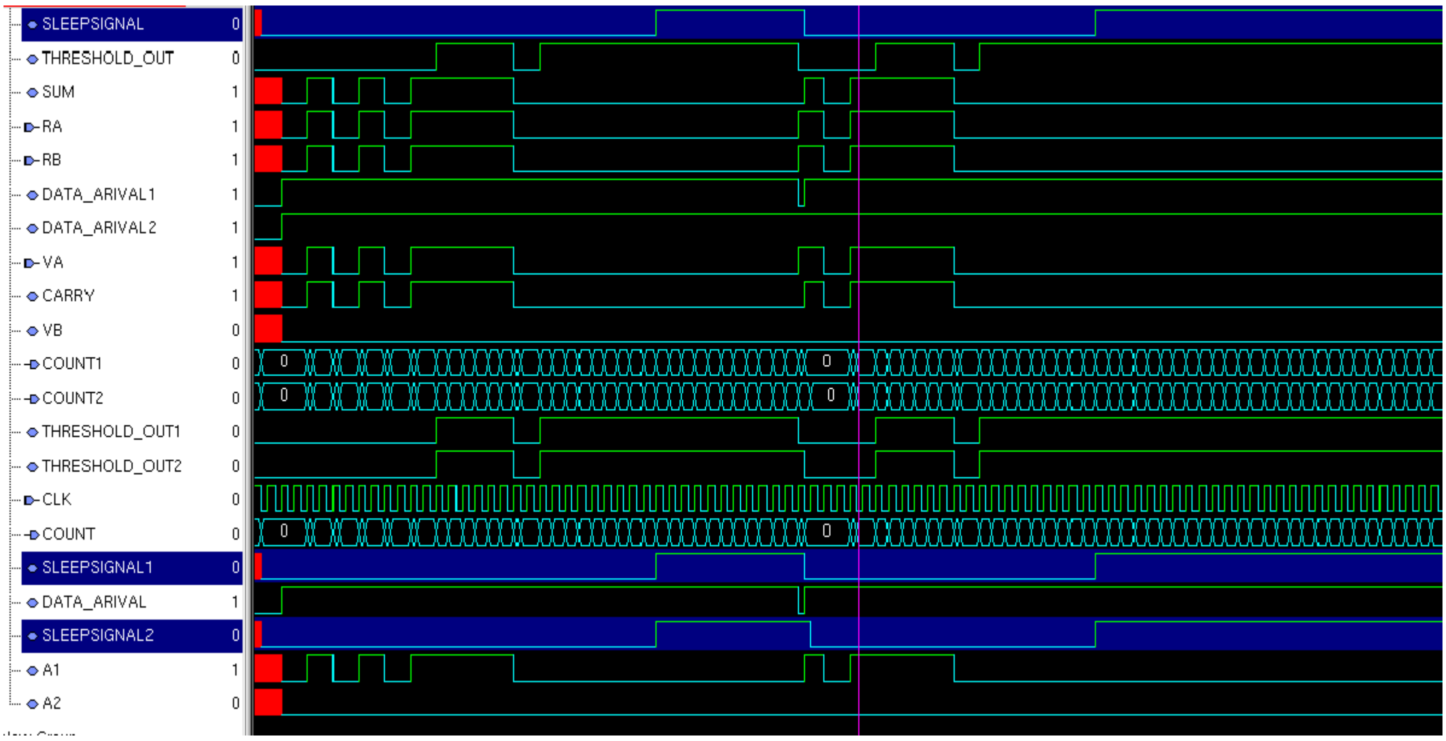


Figure 4.2 The Simulation Result For Full Adder With sleep controller

B) POWER ANALYSIS

TABLE 4.1 power and Area reult

	EXISTING SYSTEM	PROPOSED SYSTEM
POWER(mw)	**15(single cell)**	**11(single cell)**
AREA(um^2)	**249(single cell)**	**240(single cell)**

The power required to implement the Logic block after incorporating the developed fine-grain power gating LUT using the target device has been computed and listed in Table 4.1 for individual components.

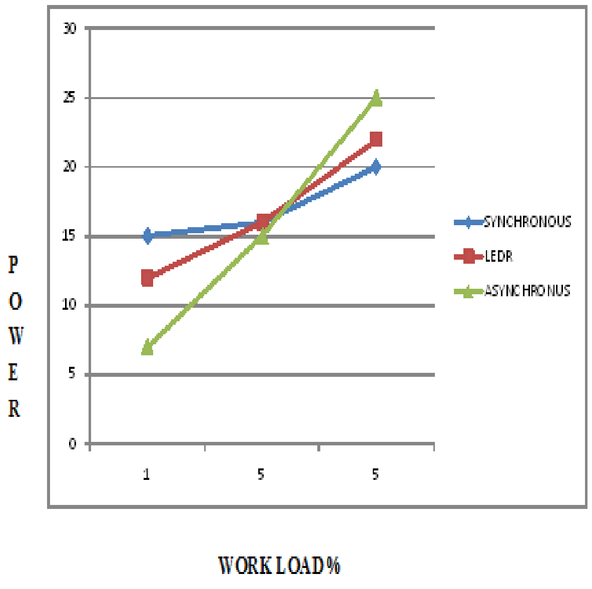


Figure 4.3. Power Report for single logic block and full adder logics

The power analysis chart compare the three types power reduction techniques for synchronous FPGA based techniques, simple LEDR encoding FPGA based techniques and asynchronous FPGA based techniques. Compare to existing techniques the proposed techniques is efficient techniques for power gating.

V. *FUTURE WORK*

In this paper only used for LEDR encoding but both LEDR and dual rail encoding used in same FPGA architecture means again reduce the power and increase the through put since 4-phase dual rail encoding achieves small area and low power for function unit, while LEDR encoding achieves high throughput and low power for data transfer.

VI. *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 Logic block before the data arrives and power OFF the Logic block only when the data does not come for quite a while. As a result, the wake-up time has been reduced. Since their data paths change dynamically and frequently, it is more difficult than FPGAs to determine the control parameters for each Logic block 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 7 mW powers.

VII. *REFERENCES*

[1] Masanori Hariyama, "A Low Power FPGA Based on Fine- Grain Power Gating Shota Ishihara, Student Member," IEEE Trans. (VLSI) SYSTEMS, VOL. 19, NO. 8, AUGUST 2011.

[2] J. Teifel and R. Manohar, "An asynchronous dataflow FPGA architecture," IEEE Trans. Computers, vol. 53, no. 11, pp. 1376–1392, Nov. 2004.

[3] M. Hariyama, S. Ishihara, C. C. Wei, and M. Kameyama, "A fieldprogrammable 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.

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

[5] 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.

[6] 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.

[7] A. Youssef, M. Anis, and M. Elmasry, "Dynamic standby prediction for leakage tolerant microprocessor functional units," in Proc. Ann. IEEE/AC, M Int. Symp. Microarch., Washington, DC, 2006, pp. 371

N.Rajagopalakrishnan ME. Applied Electronics, presented in three IEEE international conference, published in one paper on IEEE Explore .

k.SivasuparamanyanME., Ph.D, no. of IEEE international conference , no. of journals and no. of conference publications published.