Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Clock-Gating: A Novel Method for Reducing Dynamic Power Dissipation on FPGAS

Jisha Varghese[1]
Department of Electronics and Communication Engineering
Saintgits College of Enginnering,
Kottayam , Kerala.

Sreekala K S[2]
Department of Electronics and Communication Engineering
Saintgits College of Enginnering,
Kottayam, Kerala.

***Abstract*:- In all silicon devices, power dissipation is one of the major problem which has to be eliminated. Reducing power helps to minimize the precise needs for cooling, improved durability, longer autonomy in battery operated devices and lower costs. Besides, power also has significant role in choice of the computing platform right at the outset. In case of field-programmable gate arrays (FPGAs), power dissipation is more as compared to equivalent application-specific integrated circuit (ASIC), but often compare favorably to conventional processors used for same functional tasks. Previous Clock-Gating (CG) methods are not effective in implementation on FPGAs. The work includes a new Course Grained ON-OFF control method that reduces dynamic power by introducing a clock gating strategy. The work introduces a technique that aims to achieve power savings by selectively switching off the circuit part when they are not temporarily active by using a Clock enabling circuit. The proposed method has an advantage of reducing the dynamic power up to 45 % as compared with the existing methods and presents a methodology for energy efficient implementation of clock gating on FPGAs. This technique can be adopted for any application and then can finally be integrated into the synthesis stage of design flow.**

Keywords*:- *Clock-gating, dataflow, Clock enabling circuit

I. INTRODUCTION

Nowadays, power consumption is one of the major challenge in the VLSI design performance. Technological constraints and requirement constraints are the major factors causing the limiting technology improvements [1]. For a Silicon device, two components of power dissipations are: (1) Static component (2) dynamic component. Both these components has to be reduced so as to reduce overall power dissipation. In case of static power, it is the leakage current produced within the transistor. Whereas, dynamic power is the result of switching off the transistor from one logic to another. Since dynamic power is linearly dependant with frequency, clock gating strategy can be employed to reduce the power dissipation. It is a technique that is used to control power dissipation by using clock net. Up to 40% of power dissipation in synchronous circuit is caused by clock net [2]. Clock Gating reduces the unwanted switching on the clock net by disabling the clock. Clock frequency can be reduced by this clock gating method since it is directly related to power which in turn it reduces the power. Equation of dynamic power is:

$$P_{dynamic}=C_LFv_{DD}^2$$

The most optimized clock gating strategy is Register Transfer Level (RTL) clock gating. But the efficiency in the design based on this method is still a question [3]. To optimize power, Clock Gating is an accepted technique that can be applied at RTL, gate level and system level. If clock is cont inously provided, it consumes more power since registers and its associated logic toggles internally. So, Clock Gating plays a significant role in reducing the power consumption by shutting off the clock of a particular component which is temporarily inactive within the system.

In a chip that runs on battery or devices like mobile phones (with low power applications) instead of single clock gating, they would implement with different forms of gated clocks together. Manual clock gating at one end is given by using software and a driver is also provided to clock of each component which disables or enables clock used by idle components. On the rear end, an automatic clock gating is used where hardware detects whether the given clock has to be turn off if that particular clock is not currently needed. For instance, automatic gating can be used by internal bus so several peripherals which are unused on the board can be gated off and similarly bus can be gated off until CPU needs the bus.

Many authors initially stated AND gate clock gating [4] [5]. NOR based clock gating is another strategy similar to the former one [6]. Even though both the methods are simple in logic, problem of glitches is an existing issue in both cases. By using Latch based clock gating [7] problem of hazards can be reduced completely which makes this method to be dominant than previous methods. But in case of Latch based AND /OR clock gating, the component or block which is gated will consume one clock cycle extra for changing its state which produce time delay in the expected output.

MUX based is an effective strategy that eliminates the problems in the above methods .Here a multiplexer is used which controls the clocking activity of the gated circuit [6]. Negative side of this method is the expense of multiplexer and consumption of more power than rest of the methods. The Gated Clock Generation Circuit is the new approach in which more power is saved compared to previous works. This method comprises both negative and positive latch. Irrespective of clock of the target devices (whether it's clock is ON or OFF), clock of the controlling device is OFF all the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

time by which more power can be saved by avoiding unwanted switching of the clock [8].

All of these Register Transfer Level clock gating strategies includes physical gates along the path of the clocks which leads to the problem of clock skew [9]. So these methods are not possible to implement in FPGAs [10]. Such limitation of previous methods can be solved by this work where a novel clock gating method is used which can be efficiently implemented on FPGAs. The remaining work in this paper is organized as follows : Section II includes the detailed methodology of Course Grained ON-OFFClock Gating Strategy, Section III presents the experimental results and dicussion and finally in the Section IV conclusions are finally analyzed.

II. METHODOLOGY

In this paper, the below methodology is based on Xilinx-FPGA which can later be used to support other architecture of FPGAs. Chain of action firings is included in a dataflow program execution [11] [12]. Execution Trace Graphing (ETG) is a graph related representation that can be used to relate firing actions with each other. ETG is a directed, acyclic graph in which each node is a firing action and the arc which is directed one is to represent a data between two firing actions. Paper [13] demonstrated the effectiveness of such ETG based dataflow program. In ETG, every firing action are represented along with its timing information, thus transforming it into weighted graph. By weighted ETG, more optimized buffer size can be obtained [14] [15] [16]. In the rest of this paper, it describes a methodology used for clock gating which can be implemented efficiently on FPGA architectures. It includes following sub sections:

A. Clock Gating (CG)Strategy

B. Clock Enabling Circuit

A. CG Strategy

Fig. 1 shows block diagram of the proposed Clock Gating Strategy. It mainly consists of 3 blocks: Queue, Actor and Clock Enabling Circuit. Actor can be chosen according to the application. In this work, an incrementer is taken as an actor. Actor is the block whose clock is controlled by using a Clock Enabling Circuit. When output of this block is full, then its clock should be switched off as this block is idle at that state. Switching off the clock of this block will not cause any impact on the design throughput.

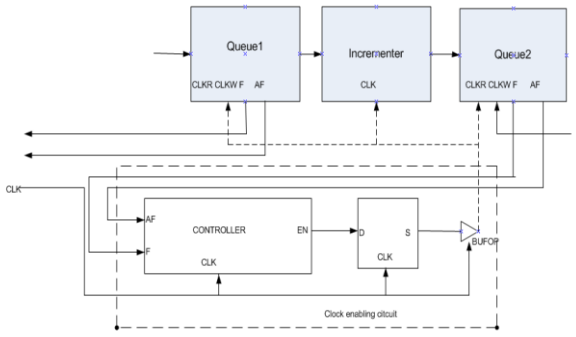

Fig. 1. Block diagram of Clock Gating Strategy.

This clock gating method is suitable to execute process which is communicated with asynchronous bufffers. To obtain loseless communication, the queue must be taken as asynchronous when the actor is provided with clock gating. This strategy includes a circuit for enabling and disabling the clock of the main block. This circuit includes: a controller, a flipflop and a buffer. Fig.1. represents an actor (incrementer) with one output port as gated one. Queues in the block diagram are asynchronous. Each queue have two clocks as inputs: one for taking the datas (CLKW)and another for producing the datas (CLKR). Besides, there are also two output ports for the queue: Almost Full (AF) and Full (F).

Through the first queue, data is given to the input of the incrementer and from the incrementer output data is collected by the second queue. Later from the second queue data has been readout. When the actor is full output from the first queue has to be stopped. For that an enabling circuit is provided to the clock of the actor which turns off the actor when it is full, thus saving more power. As seen in the figure, input actor is connected with the clock enabling circuit.

Controller is the main block within the enabling circuit. It decides and controls the clock of the actor by using a finite state machine. Output of the controller is connected with input of a flipflop. Flipflop is used to obtain output which is glitch-free. In a logic circuit which is clock gated, there is a chance of obtaining an unwanted and incorrect transition (glitch) on the output waveform. Connecting CG with flipflop ensures that the resultant waveform is glitch-free signals. This flipflop will produce clock delay when the clock is inactive, but this additional cycle will not have any impact on the incrementer block. So, this approach will have any impact on total performance of the system. The output of flipflop is then connected to a tri state buffer. This buffer is directly connected with actor's clock. Based on the decision from the controller, buffer either enables or disables the actor's clock. Since it is a tri state buffer when the enable signal from the controller is low, it will turn on the clock, else it will maintain actor's clock in high-impedance state.

The queue controller is shown in Fig. 2 and Fig. 3. The controller is a Moore finite state machine in which the output depends on present state and current input. The controller consists of: a reset, two inputs: Full (F) and Almost Full (AF), a clock and 'En' as the output which represents enable signal. Both of the inputs or at least 'F' input signal should be low to make the output 'En' signal high. Output 'En' signal becomes low when 'F' input or both inputs turn high. Input 'F' becomes high when the queue is completely occupied by data. Input 'AF' becomes high when only one space is there remaining on the queue.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

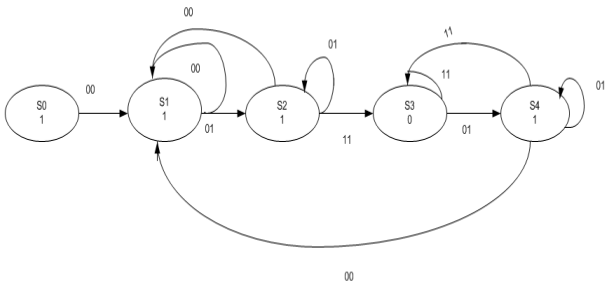

Fig. 2. Model 1: FSM with 5 states.

Fig. 2 shows a queue controller with 5 state FSM (Model1). Model 1 FSM has the following five states, S={S0, S1, S2, S3, S4}. It starts with state 'S0' which is the initial state where both inputs are low and maintains 'En' output at high state. The active high state at 'En' is maintained through the next 'S1'state. When the queue becomes almost full, the input 'AF' becomes high while 'F' input maintains in low state, the state changes to 'S2' state where still active high is maintained at 'En' output. When again both inputs becomes low, from 'S2'state it moves back to the 'S1' state. When both inputs become high i.e., when queue becomes full, then state changes to 'S3'state where output 'En' signal becomes low. That is, at this state, actor block is disabled, thus reducing the dynamic power dissipation. When the data is consumed from queue again, then the input 'F' becomes low, thus controller moves to the final 'S4' state and turns 'En' signal to active high. From the final state, depending on the status of the queue, it either moves to the state 'S3' or to the state 'S1'. That is, when the queue has some space then it moves to the previous 'S3' state from 'S4' and when queue is almost empty then it moved to the 'S1' state.

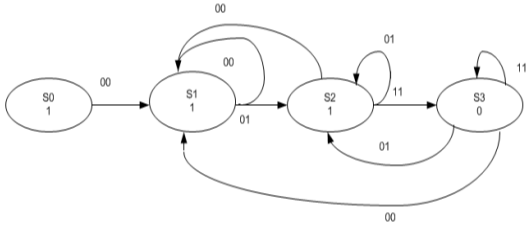

Fig. 3. Model 2: FSM with 4 states.

Fig. 3 shows a queue controller with 4 state FSM (Model 2). Model 2 FSM has the following four states, S={S0, S1, S2, S3}. It starts with state 'S0'which is the initial state where both inputs are low and maintains 'En' output at high state.

The active high state at 'En' is maintained through the next 'S1'state. When the queue becomes almost full, the input 'AF' becomes high while 'F' input maintains in low state, the state changes to 'S2' state where still active high is maintained at 'En' output. When again both inputs becomes low, from 'S2'state it moves back to the 'S1' state. When both inputs become high i.e., when queue becomes full, then state changes to 'S3'state where output 'En' signal becomes low. That is, at this state, actor block is disabled, thus reducing the dynamic power dissipation. From the final state, depending on the status of the queue, it either moves to the state 'S2' or to the state 'S1'. That is, when the queue has some space left then it moves to the previous 'S2' state from 'S3' and when queue is almost empty then it moved to the 'S1' state.

The CG block is generated at the synthesis stage together with the synthesis of computational kernels connected through queues (FIFO) constituting the dataflow network. Conceivably, these techniques can be extended to other dataflow methods.

III. RESULTS AND DISCUSSION

In this section, power reduction gain of the aforementioned methodology is evaluated. For the experimental evaluation, a Spartan 3a Starter board xc3s700a fg484 was used. The Verilog code of the CG methodology was generated and simulated with the Model Sim PE 5.5e to extract the switching activity information of the design. Following the simulation, methodology is synthesized with ISE Design Suite 14.7. The Xilinx XPower analyser was then used to determine power dissipation using the design netlist, the design constraints, and the simulation activity.

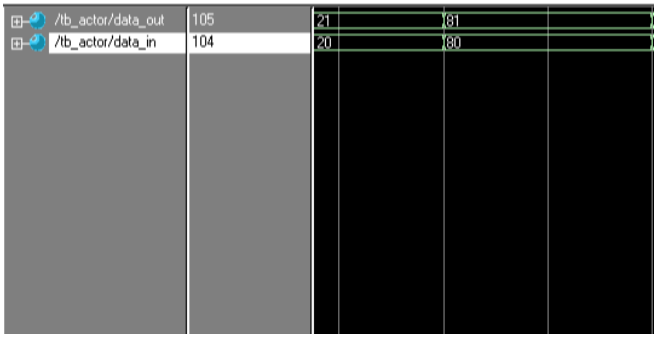

Fig.4. Simulation result of Actor (incrementer).

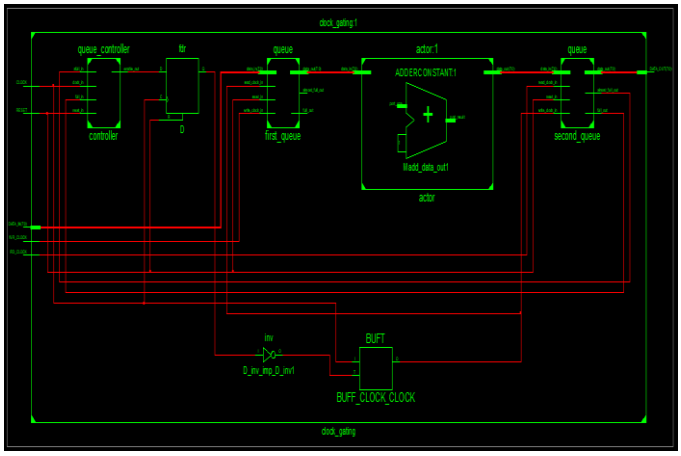

Fig. 5. RTL view of CG block diagram.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

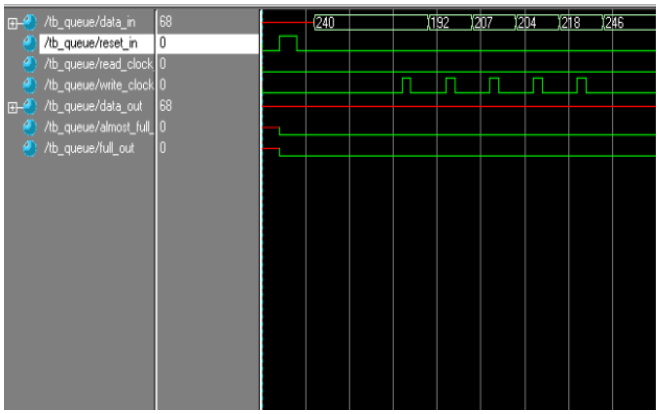

Fig.6. Simulation result of queue.

The Verilog code of the CG methodology was generated and simulated with the Model Sim PE 5.5e to extract the switching activity information of the design. FIG. 4 shows the result of an incrementer which is used as an actor. In the logic design phase of the IC Register Transfer Level is used. Fig. 6 shows the result of the queue. When the write signal is given, data is written in to data input and when read signal is given, data is read out from data output.

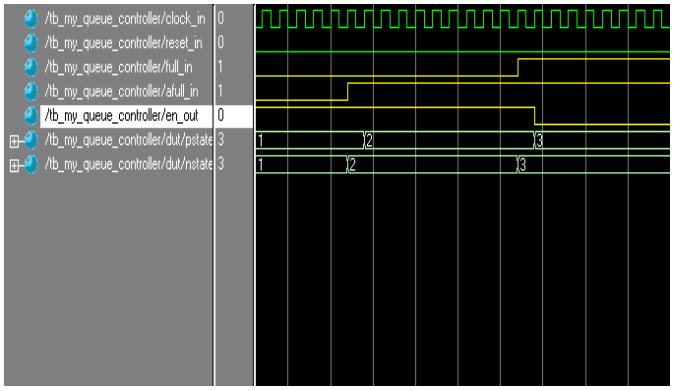

Fig.7. Result of Queue Controller when incrementer is full (Model 1)

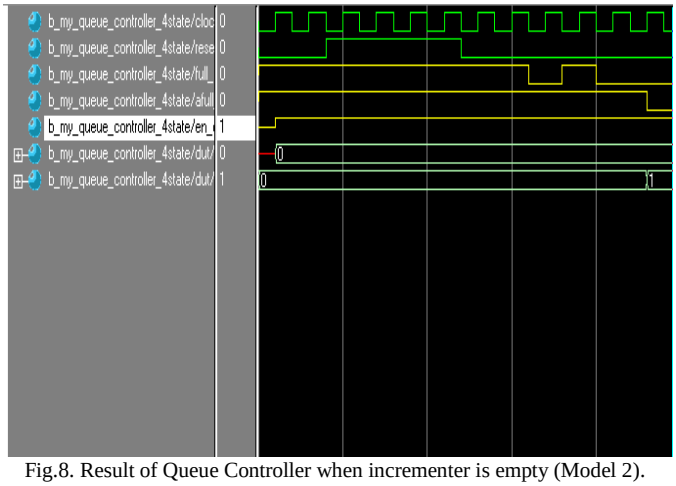

Fig.8. Result of Queue Controller when incrementer is empty (Model 2).

When the clock of the incrementer is high, this data is been incremented by 1 and is given to the input of the queue at the output port. Fig. 7 shows result of queue controller of Model 1 with 5 state FSM which controls the clock of the incrementer. When input is 11, ie., when the main incrementer block is full then the controller changes from S0 state to S1 and output signal becomes active high. Fig. 8 shows simulation of queue controller of Model 2 with 4 state FSM when incrementer is empty. Fig. 9 shows the RTL view of queue. Fig. 10 shows the RTL view of queue controller. Moore finite state machine is used as the queue controller. Table 1 depicts the power consumption of the incrementer including the circuit of the CG methodology. Two test cases were considered:

- FSM with 5 states (Model 1)
- FSM with 4 states (Model 2)

Static power of both the FSM Models remains the same but the dynamic power is reduced by more than 45 %. Since there is reduction in dynamic power total power is reduced by 7.7 %. The actors clocks label only the power consumption of the clock nets of the actor. The clocks cell contains the actors clock nets and the enabling of clock buffer nets.

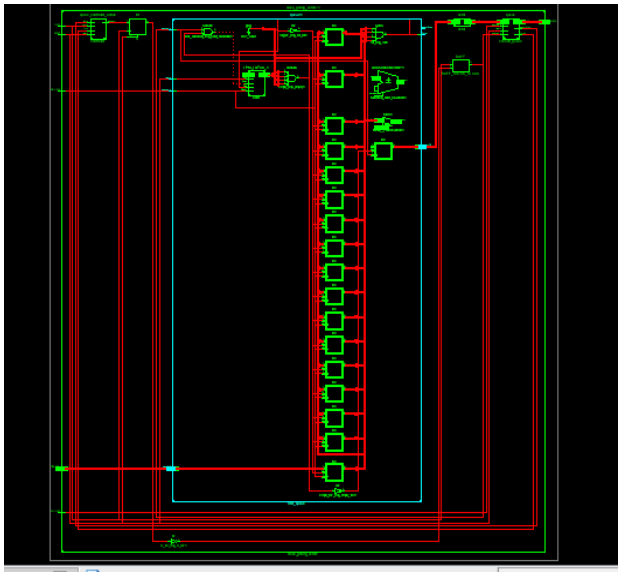

Fig.9. RTL view of queue.

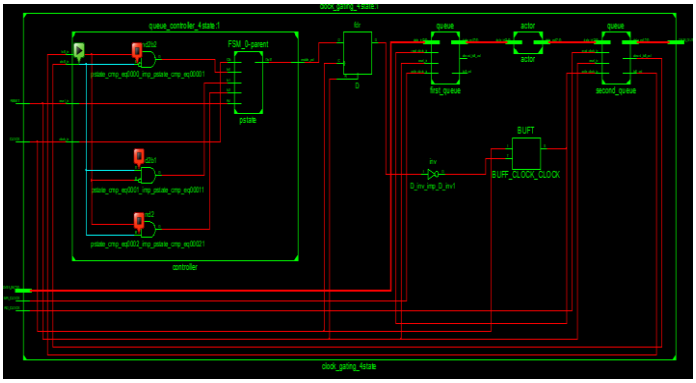

Fig.10. RTL view of queue controller.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Table 1: Power Dissipation

Model	Dynamic Power (mW)	Static Power (mW)	Total Power (mW)
Model 1	31.55	6.38	37.94
Model 2	31.55	3.46	35.01

Table 2: Clock to Setup on destination clock: CLOCK (Model 1/Model 2)

The clocks cell contains the actors clock nets and the enabling of clock buffer nets. From this Table, it can be concluded that dynamic power of the Clock Gating module has a less contribution in power as compared to the static power.

Source Clock	Rise:Rise	Fall:Rise	Rise :Fall	Fall: Fall
CLOCK	4.53 nS/ 4.89 nS	9.73 nS/ 11.41 nS	0.82 nS/ 1.62 nS	4.57 nS/ 4.71 nS

Table 2 represents the timing constraint of source clock. Minimum timing constraint is 0.82 nS for Model 1 FSM and 1.62 nS for Model 2 FSM which is obtained at the rising edge of source and falling edge of designation signal and maximum timing constraint delay is 9.73 nS for Model 1 FSM and 11.41 nS for Model 2 FSM at the falling edge of source and rising edge of designation signal.

Table 3: Clock to Setup on destination clock: READ CLOCK (Model 1/Model 2)

Source Clock	Rise:Rise	Fall:Rise	Rise :Fall	Fall: Fall
CLOCK	10.88 nS/ 11.31 nS	10.11 nS/ 10.73 nS	1.40 nS/ 1.03 nS	4.57 nS/4.73 nS

Table 4: Clock to Setup on destination clock: WRITE CLOCK (Model 1/Model 2)

Source	Rise:Rise	Fall:Rise	Rise :Fall	Fall: Fall
CLOCK	4.37 nS/ 4.04 nS	3.73 nS/ 5.04 nS	5.70 nS/ 5.71 nS	4.63 nS/ 4.50 nS

From this analysis, operating frequency of main clock of Model 1 FSM is calculated as 190.24MHz and operating frequency of main clock of Model 2 FSM as 115.34MHz ie., clock frequency of main clock is reduced. Since the clock frequency is directly related to dynamic power, it can be concluded that dynamic power also get reduced in Model 2 FSM. Table 3 represents timing constraint of READ CLOCK. Minimum timing constraint is 1.40 nS for Model 1 FSM and 1.03 nS for Model 2 FSM which is obtained at the rising edge of source and falling edge of designation signal and maximum timing constraint delay is 10.88 nS for Model 1 FSM and 11.31 nS for Model 2 FSM at the rising edge of source and rising edge of designation signal. Table 4 represents the timing constraint of WRITE CLOCK. Minimum timing constraint is 3.73 nS for Model 1 FSM and 5.04 nS for Model 2 FSM which is obtained at the falling edge of source and rising edge of designation signal and maximum timing constraint delay is 5.70 nS for Model 1 FSM and 5.611 nS for Model 2 FSM at the rising edge of source and falling edge of designation signal.

Table 5 shows the summary of delay of FSM with 5 states and FSM with 4 states. It is concluded that in both the cases delay remains the same ie., power can be reduced in Model 2 FSM without sacrificing the delay constraint. Table 6 shows synthesis results of the incrementer block synthesized for Spartan3a Starter board xc3s700a fg484 with 2 queue controllers, one with 5state finite state machine and another with 4state finite state machine.

From this table it can be concluded that savings in the power dissipation in the proposed methodology is achieved with a slight increase in control logic (when it is compared with circuit without Clock Gating) without any reduction in throughput have been achieved. But when 2 queue controllers (Model 1 and Model 2) are compared with respect to their device utilization, it can be analyzed that there is a slight reduction in utilization of devices in Model 2 FSM. This is because of the fact that in Model 2 FSM the states are reduced to 4 which results in the reduction in flipflops used by the controller. Power consumption includes power of overall clocks, the signals, logic and the total dynamic power consumption of the incrementer block.

Table 5: Delay

Model	Delay(nS)
Model 1	5.531
Model 2	5.531

Table 6: Device Utilization

Logic Utilization	Model 1	Model 2
Number of Slices	36	34
Number of Slice Flip Flops	29	26
Number of 4 input LUTs	70	67
Number used as logic	54	51
Number used as Shift registers	16	16
Number of IOs	20	20
Number of bonded IOBs	20	19
Number of GCLKs	4	2

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

IV. CONCLUSION

A Clock Gating methodology is presented in this paper. This power saving method can be applied to any application and reduces the need for extra effort during the design process of the required application at the dataflow level. During the synthesis level, this CG logic is produced. .The main part of Clock Gating is Controller which is designed as: Model 1 with 5 state FSM and Model 2 with 4 state FSM and the power dissipation of the total system is synthesized on Spartan 3A starter board. Experimental results shows new clock which is disabled when the main block is inactive temporarily, thus switching power can be reduced. Dynamic power in case of Model 2 FSM is reduced up to 45 % as compared with that of Model 1 FSM. Here, methodology is effective, simple method to recover power that is lost in cycles that are idle in state. This method is very effective when power dissipation is given more priority during design phase. Further, to implement more effectively complex applications in to limited clock domains it is necessary to develop new tools.

REFERENCES

[1] S.C Brunet *et al.,* "Partitioning and optimization of high level stream applications for multi clock domain architectures," *in.Proc.IEEE Workshop Signal Process Syst.,* Taipei, Taiwan, pp.1777-182, Oct.2013.

[2] D. W. Dobberpuhl *et al.,* "A 200-MHz 64-b dual-issue CMOS microprocessor," in *IEEE Journal of Solid-State Circuits,* vol. 27, no. 11, pp. 1555-1567, Nov. 1992.

[3] Mitch Dale, "Utilizing Clock-Gating Efficiency To Reduce Power," *EE Times India*, January 2008.

[4] Frank Emnett, Mark Biegel, "Power Reduction Through RTL Clock Gating," *SNUG* San Jose, 2000.

[5] John F. Wakerly, "Digital Design Principles and Practices", *Prentice Hall*, 2005.

[6] Hubert Kaeslin, ETH Zurich, "Digital Intergrated Circuit Design from VLSI Architectures to CMOS Fabrication,"*Cambridge University Press*, 2008.

[7] Vishwanadh Tirumalashetty, Hamid Mahmoodi, "Clock Gating and Negative Edge Triggering for Energy Recovery Clock," *ISCAS 2007*, New Orleans, LA, pp. 1141-1144, 2007.

[8] Jagrit Kathuria, M. Ayoubkhan, Arti Noor, "A Review of Clock Gating Tehniques," *MIT Publications*, vol.1, pp. 106-114, Aug 2011.

[9] E. Bezati, S. Casale-Brunet, M. Mattavelli and J. W. Janneck, "Clock-Gating of Streaming Applications for Energy Efficient Implementations on FPGAs," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 36, no. 4, pp. 699-703, April 2017.

[10] Yongwang Zhou, Xiaohong Peng, Ligang Hou, Peiyuan, Wan, Pingfen Lin, "Clock gating: A power optimization technique for smart card," *IEEE International Conference in Solid-State and Integrated Circuit Technology*, pp.1-3, 2014.

[11] Qing Wu, M Pedram and Xunwei Wu, "Clock-gating and its application to low power design of sequential circuits," *in IEEE Transactions on Circuits and Systems: Fundamental Theory and Applications*, vol. 47, no. 3, pp. 415-420, March 2000.

[12] H Mahmoodi, V Tirumala shetty, M Cooke and K Roy," Ultra Low-Power Clocking Scheme Using Energy Recovery and Clock Gating," *in IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 33-44, Jan. 2009.

[13] S. Casale -Brunet, "Analysis and optimization of dynamic dataflow programs," *Ph.D. dissertation, STI Elect. Eng., STI,* Lausanne, Switzerland, 2015.

[14] S. C. Brunet, M. Mattavelli and J. W. Janneck, "Buffer optimization based on critical path analysis of a dataflow program design," *2013 IEEE International Symposium on Circuits and Systems (ISCAS2013),* Beijing, pp. 1384-1387,2013.

[15] B. Ghavami and H. Pedram, "High performance asynchronous design flow using a novel state performance analysis method," *Computational Electrical Engineering*, vol.35, no.6, pp. 929-941, November 2009.

[16] S Suhaib, D Mathaikutty, and S Shukla, "Dataow architectures for GALS," *Electron. Notes Theor. Comput. Sci.*, vol. 200, no. 1, pp. 33–50, 2008.