Programmable Core Processor Using Data Driven Clock Gating

T.Subhashini[1] | M.Kamaraju[2] | K.Babulu[3]

[1,2]Department of ECE, GEC, Gudlavalleru, Andhra Pradesh, India.
[2]Department of ECE, JNTUK, Kakinada, Andhra Pradesh, India.

ABSTRACT

Today most of the embedded systems needs low power consumption because of all are working with the batteries. Low power consumption helps to reduce the heat, increases battery life and also increases the device reliability. Practical data driven clock gating is a most popular technique to reduce the power consumption by grouping FFs. In the proposed technique FFs merging (MBFF) in place of FFs grouping to reduce the power consumption and delay. A 16-Bit programmable core processor to be implemented using new technique (MBFF), consists of Arithmetic and Logical Unit (ALU) and other operations like Shifter and Shifter rotator. In the the VEDIC Multiplier proposed instead of conventional ALU for multiplication. This core expected to be consumes less power and whole design captured in Verilog and Implemented on FPGA and power is analyzed with X Power analyzer.

KEYWORDS: *Data driven clock gating, Power, FPGA, ALU*

I. Introduction

Nowadays, computers are indispensable implements for most of everyday activities. With the fast development of the Si technology and decrementing cost of the integrated circuit, processors is incrementing widely utilized in every field. The simple style provides exceptional performance and is right to be used in a very broad family of cost-efficacious, compatible systems. Applications include: business processing, computation -intensive scientific and engineering applications, and authentic -time management. The programmable core processor is a one type of processor architecture that utilizes a small, highly-optimized set of instructions. The architecture of advanced 16 bit low power programmable core consist of ALU ,control unit ,shifter & Barrel shifter rotator .The core is created with the Harvard architecture It may distinct program memory & data memory. In The advanced architecture 2 stage pipeline is using in the positive edge and the negative edge so speed is increased. A total 33 instruction/operation is designed first step in the processor development. The 2 stage pipelining have do the 4 operations Fetch, Decode, execute & write back. In the Fetch the 33-bit instruction and the data are drawn from the memory. Whereas in the case of decoder the data that are drawn from memory are separated by activating the components and the three operations arithmetic and logical unit (ALU), Barrel shifter and shifter as per the requirement it can activate through the practical data driven clock gating technique the low power is consumed. In the ALU block replace the VEDIC Multiplier (Urdhava Triyambica Sutra) at normal multiplication. And finally in the execution, the instruction is done, the data are manipulated and the result is stored is stored in the memory. In the 4 operations the write back operation are used. The practical data driven clock gating using MBFF is applied to the 16-bit low power programmable core to consume less delay, power and better device utilization. The control unit generates the signals from the given instructions. The architecture supports arithmetic, logical, shifting, and rotation operation.

II. LITERATURE SURVEY

The data-driven clock gating is developed based on the toggling activity of the constituent FFs. Data-driven clock gating requires, extensive simulations and statistical analysis of the FFs activity. Another grouping of FFs for clock switching power discount, known as Multi Bit FF (MBFF). MBFF tries to merge FFs right into a single cell such that the inverters driving the clock pulse into its master and slave latches are shared amongst all FFs. MBFFs the benefits are: 1) smaller layout location because of shared clock drivers and much less routing useful resource 2) much less postpone and much less strength of clock community because of fewer clock sinks 3) Controllable clock skew due to not unusual clock and permit alerts for the organization of Flip flops and reduced intensity of a clock tree. This MBFF technique is applied to the Processor lesser the power consumption.

Circuit diagram:

The figure 4 shows the circuit diagram for Multi Bit FF. Multi-Bit Flip-Flops are efficient for reducing the power consumption because they have shared inverter inside the flip flop. Clock skew is also minimized at the same at the same time. Flip flop grouping and flip flop merging uses same clock condition. Set and reset condition is also same.

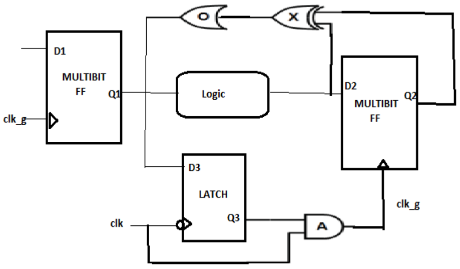


Figure 1: Practical Data Driven Clock Gating using MBFF

version, after your

III. PROGRAMMABLE CORE PROCESSOR

The programmable core is flexible processor architecture. The architecture of 16 bit core consist of ALU ,control unit ,shifter & Barrel shifter rotator .The Processor is designed with the Harvard architecture It may distinct program memory & data memory. In the suggested architecture 2 stage pipeline is using in the positive edge and the negative edge so the speed is increased. A total 33 instruction/operation is designed as a first step in the development of the processor. The 2 stage pipelining have do the 4 operations Fetch, Decode, execute &write back.

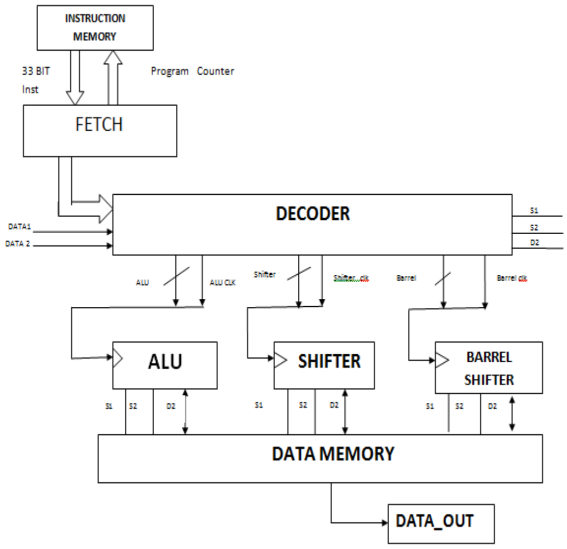


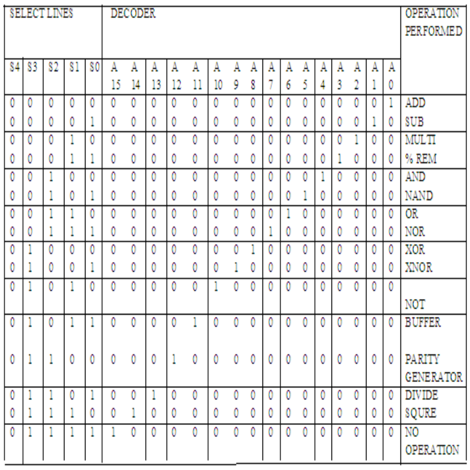
Figure 2: Block diagram of 16-Bit programmable core

Opcode (33 bits), Source1 (9 bits), Source2 (9 bits), Destination (9 bits)
Control unit- The control unit contains in 2 parts of operations. In our suggested architecture the control unit generates all the control signals needed to control the coordination between the whole components of the processor. It is important for generating the control signals that decide to use.

Fetch and Instruction memory- In the fetch33 bit of instructions are drawn from memory, in the instruction memory it comprises the instruction that are executed by the 16- bit core.

ALU: The ALU design has 16 operations are using ,containing two gadgets one is for logical operation inclusive of AND, NAND, OR, NOR, XOR, XNOR, BUFFER, Parity Generator are used and another one is arithmetic operation is used CSA Adder , subtract , multiplier and divider ,reminder, square features are operated. layout a 16 bit ALU include eight bit input bit, eight bit destination address and two addressing modes are used register addressing mode and immediate addressing mode, and the memory is also used for storing data through the given select lines particular operations are performed.

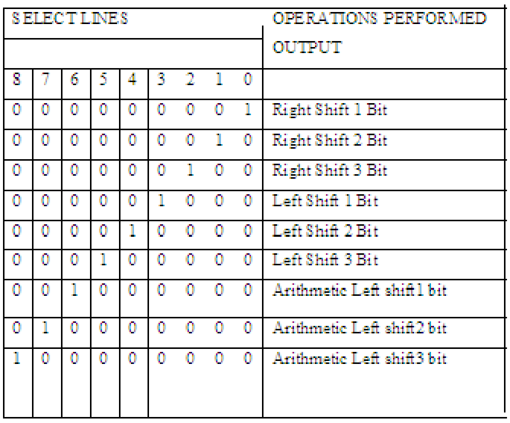
Table No: 1 ALU Operations



SELECT LINES					DECODER																OPERATION PERFORMED
S4	S3	S2	S1	S0	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8	A 7	A 6	A 5	A 4	A 3	A 2	A 1	A 0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	ADD
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	SUB
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	MULTI
0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	% REM
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	AND
0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	NAND
0	0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	OR
0	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	NOR
0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	XOR
0	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	XNOR
0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	NOT
0	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	BUFFER
0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	PARITY GENERATOR
0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	DIVIDE
0	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SQURE
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NO OPERATION

Universal Shift Register: The **Shift Register** is a type of sequential circuit is used for the storage or transfering of data in the form of binary numbers. The Shift Register is used for data storage that are used in calculators or computers to store data such as 2 binary numbers before they are added together, or to convert the data from either a serial to parallel or parallel to serial format. The individual data latches that make up a single shift register are all driven by a common clock (clk) signal making them synchronous devices. The nine operations are performed in the Shifter according to the select line it will work by right shift ,left shift ,and arithmetic left shift operations are

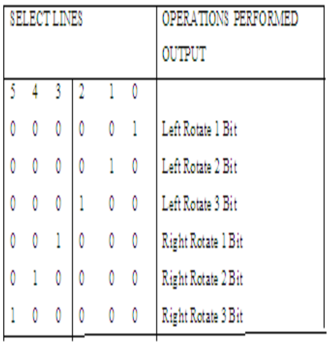
Table No: 2 Universal Shifter Operations



SELECT LINES									OPERATIONS PERFORMED
									OUTPUT
8	7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	1	Right Shift 1 Bit
0	0	0	0	0	0	0	1	0	Right Shift 2 Bit
0	0	0	0	0	0	1	0	0	Right Shift 3 Bit
0	0	0	0	0	1	0	0	0	Left Shift 1 Bit
0	0	0	0	1	0	0	0	0	Left Shift 2 Bit
0	0	0	1	0	0	0	0	0	Left Shift 3 Bit
0	0	1	0	0	0	0	0	0	Arithmetic Left shift1 bit
0	1	0	0	0	0	0	0	0	Arithmetic Left shift2 bit
1	0	0	0	0	0	0	0	0	Arithmetic Left shift3 bit

Barrel Shifter: Basically, a barrel shifter works to shift data by incremental stages which avoids extra clocks to the register and reduces the time spent shifting or rotating data (the specified number of bits are moved/shifted/rotated the desired number of bit positions in a single clock cycle). A barrel shifter is commonly used in Digital Signal Processing (DSP), and is useful for most applications that shift data left or right A barrel shifter can be used to rotate data. The clock is applied to the barrel shifter to shift the number of operations. In the barrel shifter performs 6 operations.

Table No: 3 Barrel Shifter Operations



SELECT LINES						OPERATIONS PERFORMED
						OUTPUT
5	4	3	2	1	0	
0	0	0	0	0	1	Left Rotate 1 Bit
0	0	0	0	1	0	Left Rotate 2 Bit
0	0	0	1	0	0	Left Rotate 3 Bit
0	0	1	0	0	0	Right Rotate 1 Bit
0	1	0	0	0	0	Right Rotate 2 Bit
1	0	0	0	0	0	Right Rotate 3 Bit

Programmable Core Processor Using Practical Data Driven Clock Gating (MBFF):

Figure .6 shows an example of 16-Bit low power programmable core using Practical Data Driven Clock Gating and we can replace the VEDIC Multiplier in place of normal multiplication .By applying practical data driven clock gating to the 16-Bit low power programmable core to reduce the Power dissipation, gate count and delay.

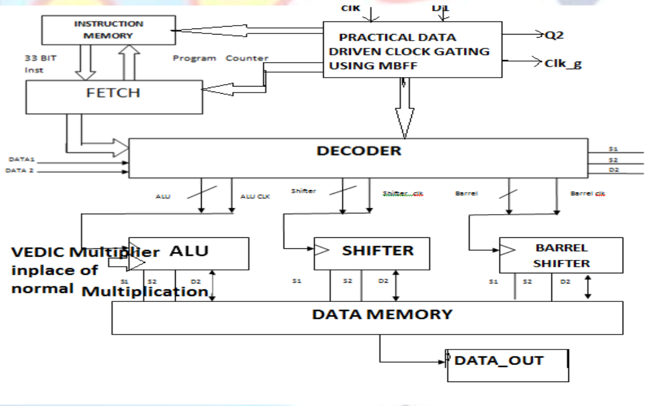


Figure 3: Block Diagram of 16 Bit

Programmable Core Using Practical Data

Driven Clock Gating (MBFF)

IV. IMPLEMENTATION

Figure 4 shows the flow chart for 16-bit low power programmable core. First step is set the program counter (PC) and fetch the instruction from the instruction memory and the next step is to increment program counter and decode from the instruction memory, based on the opcode to perform ALU, Shifter, Shifter rotator and this result is stored in the data memory.

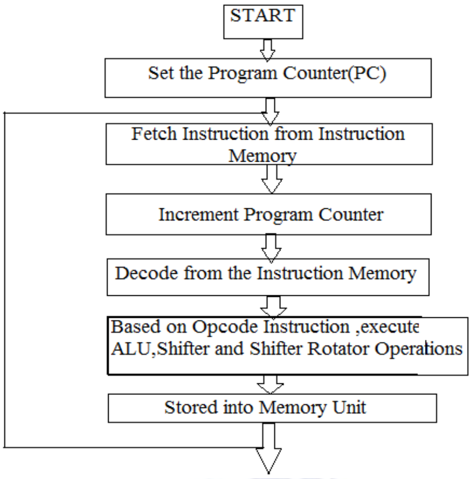

START
Set the Program Counter(PC)
Fetch Instruction from Instruction Memory
Increment Program Counter
Decode from the Instruction Memory
Based on Opcode Instruction ,execute ALU,Shifter and Shifter Rotator Operations
Stored into Memory Unit

Figure 4: Flow Chart for the 16-Bit low power programmable core

The programmable core has flexible architecture. All the implementation done on XILINX VIRTEX-5 XUPV5-LX110T.

V. RESULTS

Simulation results for FFs grouping using practical data driven clock gating:

The simulation results for FFs grouping using practical data driven clock gating is shown below figure. First the rst is not applied the output is undefined condition. If rst is applied clk is high and D input is high, the clk_g is high, the output Q2 is same as that of D1 input. And the next time interval clk is low and D1 input is high ,theclk_g is low, the output Q2 is shows the previous input until the next clock edge is high.

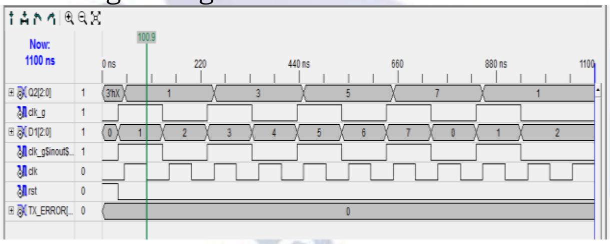


Figure 8 : FFs Grouping using practical data driven clock gating

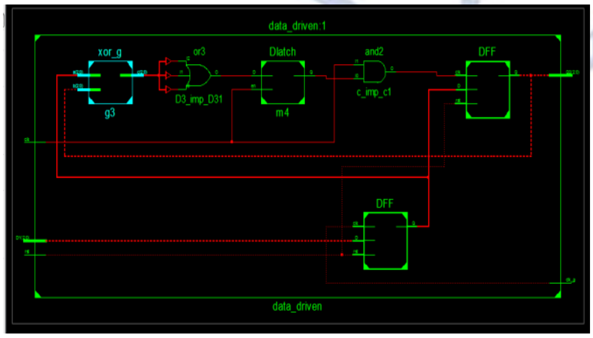


Figure9 :RTL Internal diagram for FFs Grouping using practical data driven clock gating

The figure shows the RTL representation internal diagram for FFs grouping using practical data driven clock gating.RTL opens an NGR file that can be viewed as a gate-level schematic. This RTL schematic is generated after the HDL synthesis phase of the synthesis process. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND, OR gates, that are independent of the targeted Xilinx device.

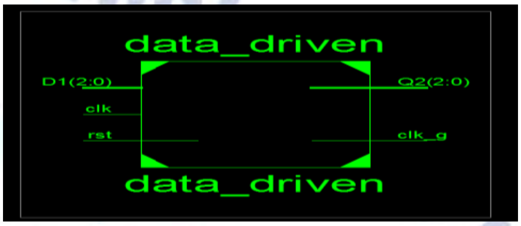


Figure 10 : RTL representation for FFs Grouping using practical data driven clock gating

The figure 10 shows the block box representation of FFs Grouping using practical data driven clock gating. In this representation how many number of inputs and outputs are used.

Simulation results for FFs Merging (MBFF) using practical data driven clock gating:

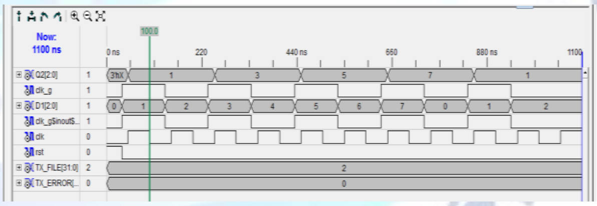


Figure 11 : FFs Merging using practical data driven clock gating

The simulation results for FFs grouping using practical data driven clock gating is shown below figure. First the rst is not applied the output is undefined condition. If rst is applied clk is high and D input is high, the clk_g is high, the output Q2 is same as that of D1 input. And the next time interval clk is low and D1 input is high ,theclk_g is low, the output Q2 is shows the previous input until the next clock edge is high.

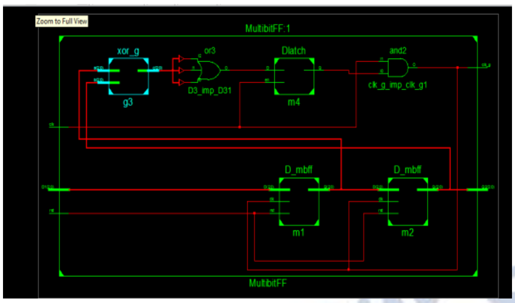


Figure12 :RTL Schematic Internal diagram for FFs Merging using practical data driven clock gating

The figure shows the RTL representation internal diagram for FFs merging using practical data driven clock gating.RTL representation opens an NGR file that can be viewed as a gate-level schematic. This RTL schematic is generated after the HDL synthesis phase of the synthesis process. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND, OR gates, that are independent of the targeted Xilinx device.

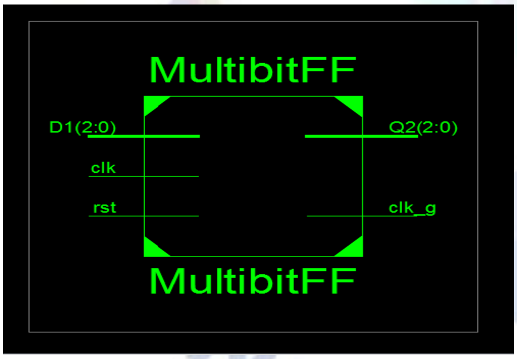


Figure 13 : RTL representation for FFs Merging using practical data driven clock gating

The figure 13 shows the block box representation of FFs Merging using practical data driven clock gating. In this representation how many number of inputs and outputs are used.

Simulation Results for 16-Bit programmable core without clock gating:

The low power programmable core using practical data driven clock gating technique (MBFF) and is evaluated using XILINX VIRTEX-5 XUPV5-LX110T, 28 nm technology processor is properly validated and the simulation result show that the processor is capable of implementing on the two clock cycles.The 16 bit low power programmable core using two stages pipeline that reduces the latency and increases the speed. A low power design technique called clock gating technique was employed to reduce the power consumption .This clock gating method reduce the power consumption upto 0.177W.

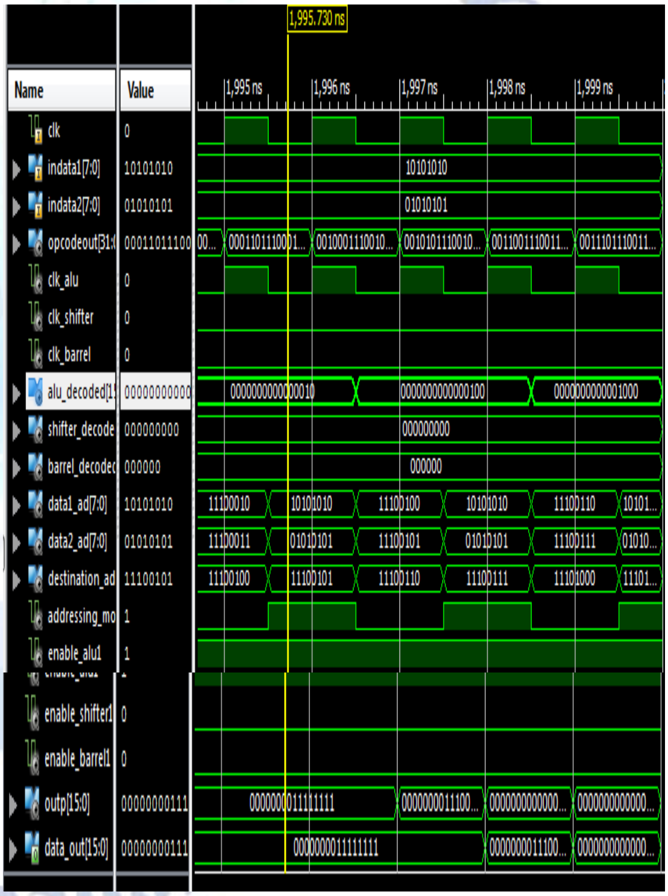


Figure 14: 16-Bit programmable core without Clock gating

The figure 14 shows the simulation results for 16-bit core without clock gating. First two data inputs indata1,indata2 and clock is applied to the processor .the instruction set is generated, based on the opcode which operation is performed either ALU, shifter, shifter rotator operations. For suppose ALU operation is performed clk_alu is activated, alu_decoder is enabled to fetch the inputs, the ALU operation is performed and finally result is stored on the data memory.

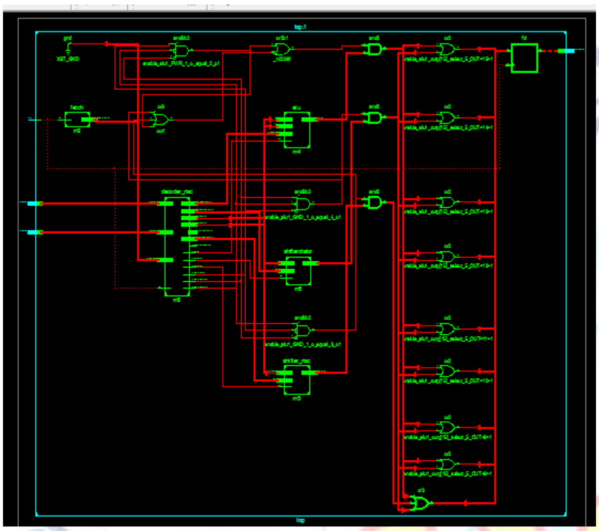


Figure 15 : RTL Schematic Internal Diagram for 16-Bit programmable core without Clock Gating

The figure 15 shows the RTL Schematic internal diagram for 16-bit low power programmable core without clock gating technique. RTL representation opens an NGR file that can be viewed as a gate-level schematic.This schematic is generated after the HDL synthesis phase of the synthesis process. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND, OR gates, that are independent of the targeted Xilinx device.

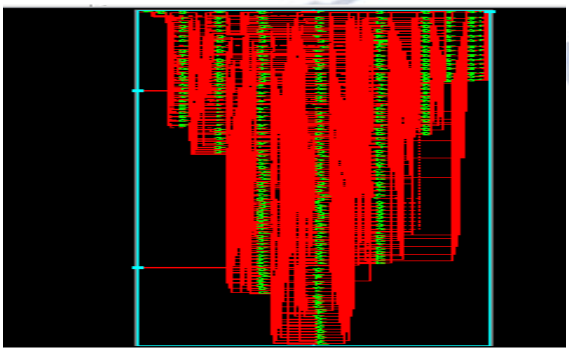


Figure 16 : Technology Schematic Internal Diagram for 16-Bit programmable core without Clock Gating

The figure 16 shows the technology schematic for 16-Bit low power programmable core without clock gating. Technology schematic opens an NGC file that can be viewed as an architecture-specific schematic. It shows a representation of the design in terms of logic elements optimized to the target Xilinx device, for example, in terms of LUTs, carry logic, I/O buffers, and other technology-specific components.

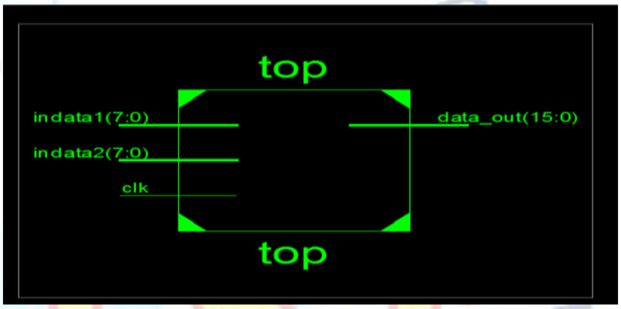
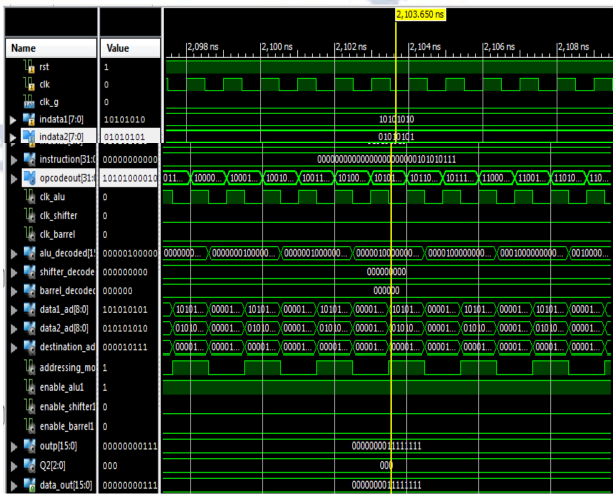


Figure 17 :RTL Representation of 16-Bit programmable core without Clock Gating

The figure 17 shows the block box representation of 16-bit low power programmable core without clock gating technique. In this representation how many inputs and outputs are used.

Simulation Results for 16-Bit low power programmable core with practical data driven clock gating:



Name	Value
rst	1
clk	0
clk_g	0
indata1[7:0]	10101010
indata2[7:0]	01010101
instruction[31:(	00000000000
opcodeout[31:(	10101000010
clk_alu	0
clk_shifter	0
clk_barrel	0
alu_decoded[1!	00001000000
shifter_decode	000000000
barrel_decodec	000000
data1_ad[8:0]	101010101
data2_ad[8:0]	010101010
destination_ad	000010111
addressing_mo	1
enable_alu1	1
enable_shifter1	0
enable_barrel1	0
outp[15:0]	00000000111
Q2[2:0]	000
data_out[15:0]	00000000111

Figure 19: 16-Bit low power programmable core with Practical Data Driven Clock gating

The figure 19 shows the simulation results for 16-bit low power programmable core without clock gating. First two data inputs indata1, indata2 and clock and clk_g is applied to the processor. The practical data driven clock gating helps to reduce the power consumption. The instruction set is generated, based on the opcode which operation is performed either ALU, shifter, shifter rotator operations. For suppose ALU operation is performed clk_alu is activated, alu_decoder is enabled to fetch the inputs,the ALU operation is performed and finally result is stored on the data memory.

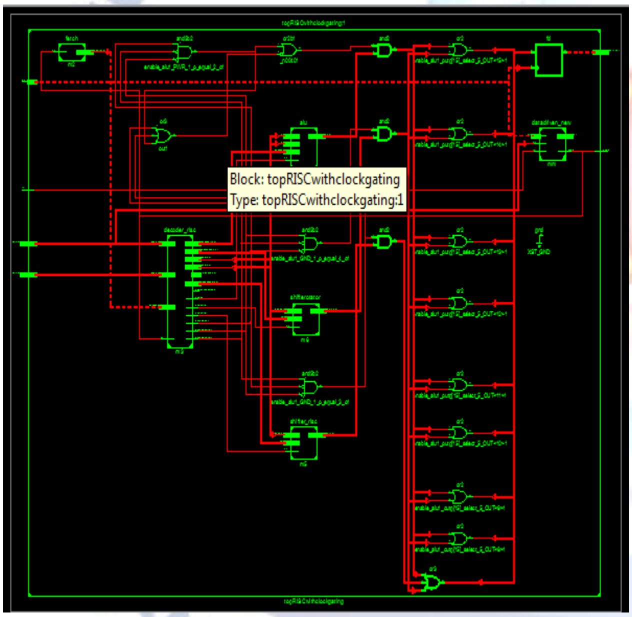


Figure 20 : RTL Schematic Internal Diagram for 16-Bit low power programmable core with Practical Data Driven Clock Gating

The figure 20 shows the RTL Schematic internal diagram for 16-bit low power programmable core with practical data driven clock gating technique. RTL representation opens an NGR file that can be viewed as a gate-level schematic. This RTL schematic is generated after the HDL synthesis phase of the synthesis process. It shows a representation of the pre-optimized design in terms of generic symbols, such as adders, multipliers, counters, AND, OR gates, that are independent of the targeted Xilinx device.

Table 5: Programmable core processor with VEDIC multiplier Clock gating

With clock gating and conventional multiplier	With clock gating and VEDIC multiplier
1. Device Utilization Summary: 16 out of 285 5%	1.Device Utilization Summary: 17 out of 400 4%
2.Delay: 5.845ns	2.Delay: 4.521ns
3.Frequency: 171.086MHz	3.Frequency: 221.190MHz
4.Power: 0.237W	4.Power: 0.177W

VI. CONCLUSION

The 16-bit programmable core processor architecture implemented on FPGA. By applying practical data driven clock gating, observed that reduces the power dissipation, number of IOBs and Delay.Comparing the programmable core processor with and without clock gating, the power is reduced upto 20%.

REFERENCES

[1] ShamuelWimmer,IsraelKoren "Design Flow for Flip-Flop Grouping in Data Driven Clock Gating" *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 22, No. 4, April 2014

[2] S. Wimer and I. Koren, "The Optimal fan-out of clock network for power minimization by adaptive gating," *IEEE Trans. Very Large ScaleIntegr. (VLSI) Syst.*, vol. 20, no. 10, pp. 1772–1780, Oct. 2012.

[3] PriyankaTrivedi, Rajan Prasad Tripati "Design and Analysis of 16-Bit RISC Processor sing Low Power Pipelining" International Conference on Computing, Communication and Automation (ICCCA2015)@2015*IEEE*.

[4] Samiappa Sakthikumaran1, S. Salivahanan, V. S. Kanchana Bhaaskaran2 "16-Bit RISC Processor Design for Convolution Application" *IEEE*-International Conference on Recent Trends in Information Technology, ICRTIT 2011.

[5] B. Ramkumar and Harish M Kittur ,feburaury 2012 “Low-Power and Area-Efficient Carry Select Adder” , *IEEE Transactions On Very Large Scale Integration (VLSI) Systems*, Vol. 20, no. 2,

[6] R.uma , apr2012 “ Design and Performance analysis of 8 bit RISC Processor Using Xilinx Tool”, International Journal of Engineering Research and Application.

[7] Indu., Arun Kumar “Design of Low Power Pipelined RISC processor” , International Journal of Advanced Research in Electrical & electronics & instrumentation Engineer.

[8] Li Li, Wei Wang and Ken Choi, Seongmo Park and Moo-Kyoung Chung “SeSCG: Selective Sequential Clock Gating for Ultra-Low-Power Multimedia Mobile Processor Design” 2010 *IEEE.*

[9] Xiaotao Chang, Mingming Zhang, Ge Zhang, Zhimin Zhang, Jun Wang, “Adaptive Clock Gating Technique for Low Power IP Core in SoC Design” , © 2007 *IEEE.*

[10] Hai Li, SwarupBhunia, Yiran Chen, T. N. Vijaykumar, and Kaushik Roy “Deterministic Clock Gating for Microprocessor Power Reduction” The Ninth International Symposium on High-Performance Computer Architecture © 2002 *IEEE.*

[11] W. Aloisi and R. Mita, “Gated-clock design of linear-feedback shift registers,” *IEEE Trans. Circuits Syst., II, Brief Papers*, vol. 55, no. 5, pp. 546–550, Jun. 2008.

[12] K. Roy, S. Mukhopadkyay, and H. Mahmoodi-meimand, “Leakage current mechanisms and leakage reduction techniques in deep sub micrometer CMOS circuits,” *Proc. IEEE*, vol. 91, no. 2, pp. 305–327, Feb. 2003.

[13] Y. Tsai, D. Duarte, N. Vijaykrishnan, and M. Irwin, “Characterization and modeling of run-time techniques for leakage power reduction,” *IEEETrans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 11, pp. 1221– 1233, Nov. 2004.

[14] M. Muller, S. Simon, H. Gryska, A. Wortmann, and S. Buch, “Low power synthesizable register Files for processor and IP cores,” *IEEETrans. Very Large Scale Integr. (VLSI) Syst., Low-Power Design Tech.*, vol. 39, no. 2, pp. 131–155, Mar. 2006.

[15] W. Shen, Y. Cai, X. Hong, and J. Hu, “Activity-aware registers placement for low power gated clock tree construction,” in *Proc. IEEE Comput.Soc. Ann. Symp. VLSI*, Mar. 2007, pp. 383–388.

[16] K. Usami and N. Ohkubo, “A design approach for fine-grained run-time power gating using locally extracted sleep signals,” in *Proc. Int. Conf.Comput. Design*, 2006, pp. 151–161.

[17] S. Jairam, M. Rao, J. Srinivas, P. Vishwanath, H. Udayakumar, andJ. Rao, “Clock gating for power optimization in ASIC design cycle theory & practice,” in *Proc. Int. Symp.Low Power Electron. Design*, Aug. 2008, pp. 307–308.

[18] Thom J. Eguia, Sheldon X.-D. Tan, RuijingShen,Duo Li, Eduardo H. Pacheco, MurliTirumala, and Lingli Wang, “General Parameterized Thermal Modeling for High-Performance Microprocessor Design” *IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 20, No. 2, February 2012*