International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 05, May-2015

Designing and Implementation of Low Power SPI-Switch Interface on FPGA

Sukhwinder Kaur
Student, Deptt. Of ECE
Amritsar College of Engg. & Technology
Amritsar, Punjab ,India

Narinder Sharma
Head of Department. Deptt. Of ECE and EE
Amritsar College of Engg. & Technology
Amritsar, Punjab ,India

***Abstract*— This paper describes an approach for reducing the dynamic power consumption of SPI-4-port switch interface by using RTL clock gating technique and analyzing the performance parameters of the interface on different Lattice FPGA families. The SPI is a full duplex serial data communication protocol (designed by Motorola) which uses four wires i.e. MOSI, MISO, SS and SCLK for establishing a communication between host processors and its peripherals. It operates either as a master or a slave. The master provides the clock signal and determines the status of the slave select signal to communicate with the other devices. A four port switch is used for programming the SPI control registers. The architecture consists of a 4-port switch having registers, single port RAM, SPI, SPI clock generator, Data selector and gated clock block. The RTL code and test-bench verification are carried out on Modelsim Altera v6.3 and performance analysis is carried out on Lattice Diamond XP2 FPGA families i.e. LFXP2-5E,8E,17E AND 30E.**

Keywords—SPI, I2C, ICG, FPGA

I. INTRODUCTION

The physical transfer of data over a point-to-point or point to point multipoint communication channel is called as data communication. For example computer buses, storage media etc. [1]. On the basis of signals data can be analog (continuous) or digital (discrete).

Data transmission can be carried out in two ways either parallel or serial. In case of serial data transmission data is transmitted one bit at a time, using a clock to maintain integrity between words. For eg SPI, I2C, UART, RS-232 etc. In case of parallel data transmission data is sent parallel. For eg. Printer communication, ISA, IDE, SCSI etc. Serial data transfer is quite advantageous as compared to parallel data transfer. In serial communication wiring is simpler and there is less interaction between the conductors of serial cables [2].

.

II. SERIAL PERIPHERAL INTERFACE

A. Operation

Serial peripheral interface (SPI) is a serial bus standard protocol designed by Motorola (also known as microwire). SPI is a synchronous serial data link protocol that operates in full duplex mode and is available on popular communication processors and microcontrollers. The SPI bus is a 3+N wire interface where N is the number of devices connected to a single master on the bus. Only one master can be on the bus. Fig 1 shows SPI master –slave communication [3][4].

SPI operates in two modes: master and slave. To begin a communication SPI-master first generates SCLK. according to the baud rate register. The master then pulls SS (Slave Select) low for the desired chip. SS is an active low signal.

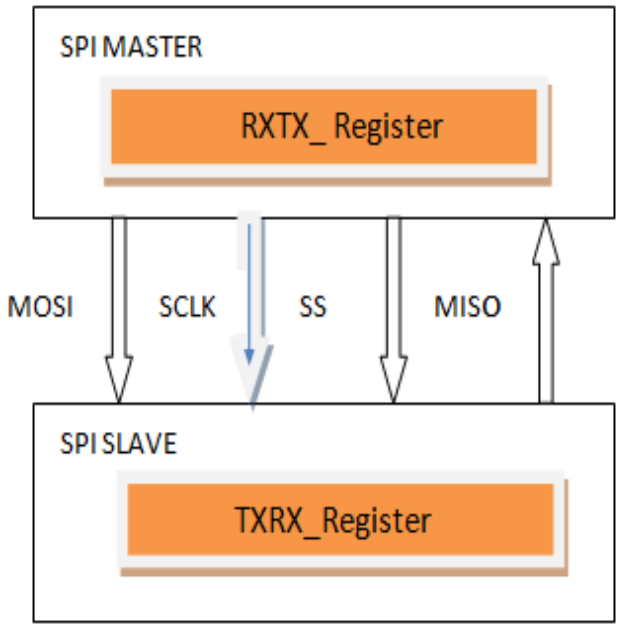

Figure 1 SPI master-slave communications

Transmission normally involve two shift registers of some given word size(eight bits) , one in the master and one in the slave i.e. TXRX_register and RXTX_register notations used in RTL code. [1] . Data is usually shifted out with the MSB first, while shifting a new LSB into the same register. After that register has been shifted out, the master and slave have exchanged register values. Then each device takes that value and write it to memory. Transmissions may involve any number of clock cycles. Transmissions often consist of 8-bit words, and a master can initiate multiple such transmissions if it is required. [5] [6].

There are 2 types of SPI configurations i.e. Independent slave configuration and daisy chain configuration. In the independent slave configuration, an independent slave select line is available for each slave. This is the way in which the SPI is normally used. Since the MISO pins of the slaves are connected together, they are required to be tri-state pins. Whereas in daisy chain configuration the whole chain act as a SPI communication shift register. Such a feature only

requires a single SS line from the master rather than a separate SS line [12].

”.

III. RTL CLOCK GATING

The RTL clock gating is a technique which is used to control power dissipated by clock net. In digital synchronous circuits clock (net) is the main factor for dynamic power dissipation. RTL clock gating reduces the unwanted switching on the parts of clock net by disabling the clock [7].

Clock gating works by taking the enable conditions attached to the registers and uses them to gate the clocks. Hence it is required that a design must contain these enable conditions in order to use clock gating. This clock gating process can also save significant die area as well as dynamic power, since it removes large numbers of multiplexers and replaces them with clock gating logic. This clock gating logic is generally in the form of "Integrated clock gating" (ICG) cells. [8][9].

RTL Clock gating logic can be added into a design in a variety of ways:

1. RTL coding: Can be coded into the RTL code as enable conditions that can be automatically translated into clock gating logic by synthesis tools.
2. Manually inserting into the design by the designers (module level clock gating) by instantiating library specific ICG (Integrated Clock Gating) cells to gate the clocks of specific modules or registers.
3. Semi-automatically inserted into the RTL by automated clock gating tools. [10].

IV. SYSTEM INTERFACE

The Figure 2 shows the proposed system overview of SPI-Switch interface.

The system architecture consists of six blocks. The gated clock block is used for reducing the dynamic power consumption of the whole design. The data device selector is used to select either RAM or Switch at a time for communication with SPI. Switch is a 4-port device which is used to program the SPI control register. It consists of 3 registers. Destination address registers gives the information about the output ports of the switch.

There are 4 output ports of the switch. Last two bits of this registers decides the output port. Then comes control register which is used to control the 4-port switch operation and SPI control register having bits namely CPOL, CPHA, LSBFE and SPI-enable.

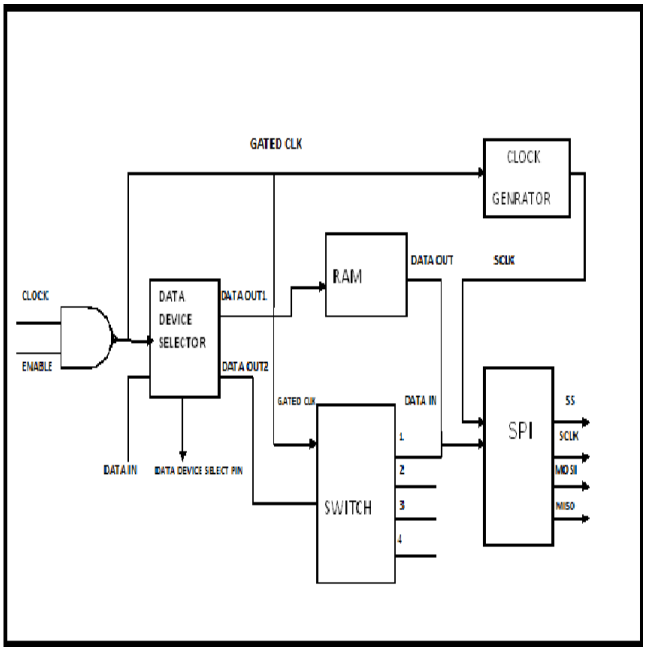

Figure 2 SPI-Switch block interface

RAM is used for storing the received and transmits data for SPI. Clock generator is used to generate the SPI clock according to the baud rate register and SPI is used for serial communication.

V. RESULTS AND SIMULATIONS

Figure 3 shows the hierarchal view of the SPI-Switch interface on Lattice Diamond FPGA.

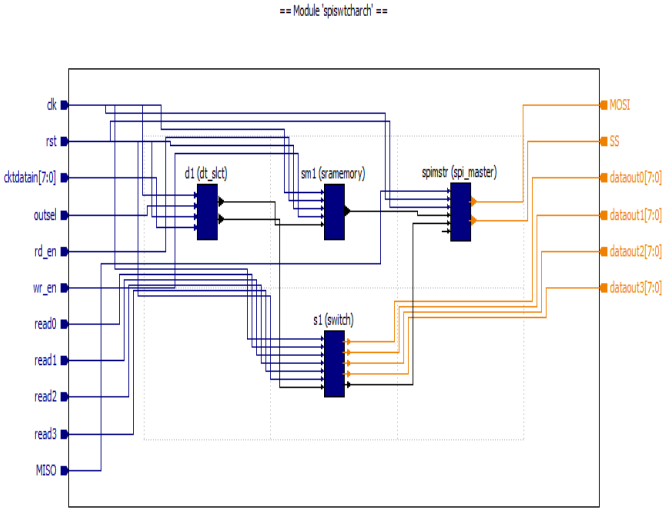

Figure 3 Hierarchal view of SPI-switch interface on XP2 FPGA

The whole design is synthesized on Lattice Diamond XP2 FPGA families i.e. LFXP2-5E, LFXP2-8E, LFXP2-17E and LFXP2-30E for the calculation of area, dynamic power, total dynamic power and power consumption by logic blocks.

The blue blocks are the main blocks i.e. data selector, SPI, Switch and RAM. The blue lines are the input ports and wires and orange lines are the output ports.

The Figures 4 shows the performance analysis of the non-gated architecture on LFXP2-5E.

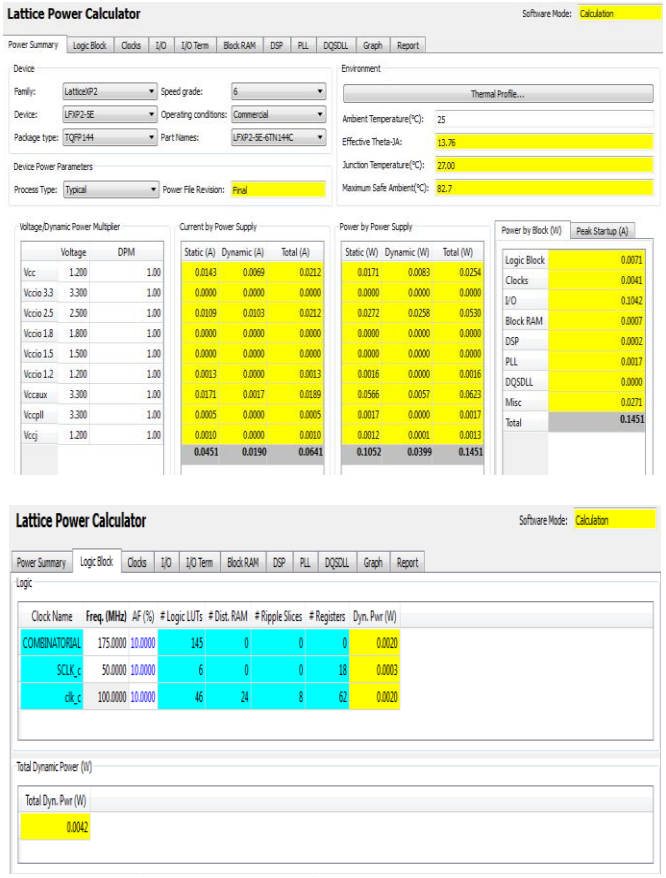

Figure 4 Performance analysis of non-gated architecture on LFXP2-5E device

Figure 5, 6 and 7 shows the performance analysis of non-gated architecture on other XP2 FPGA families.

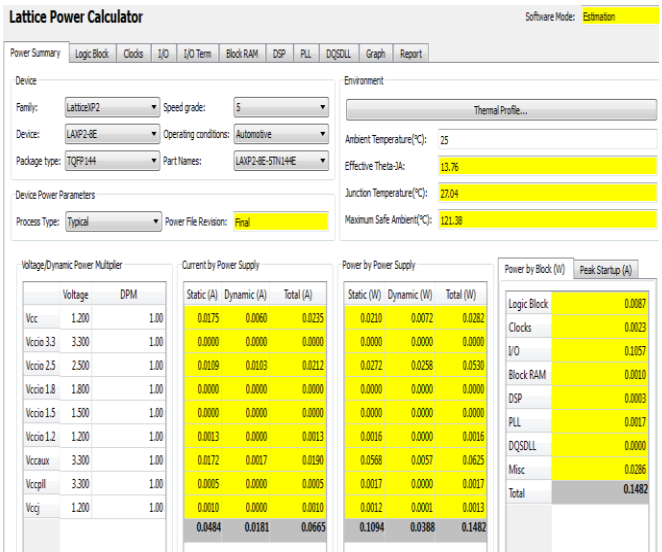

Figure 5 Performance analysis of non-gated architecture on LFXP2-8E device

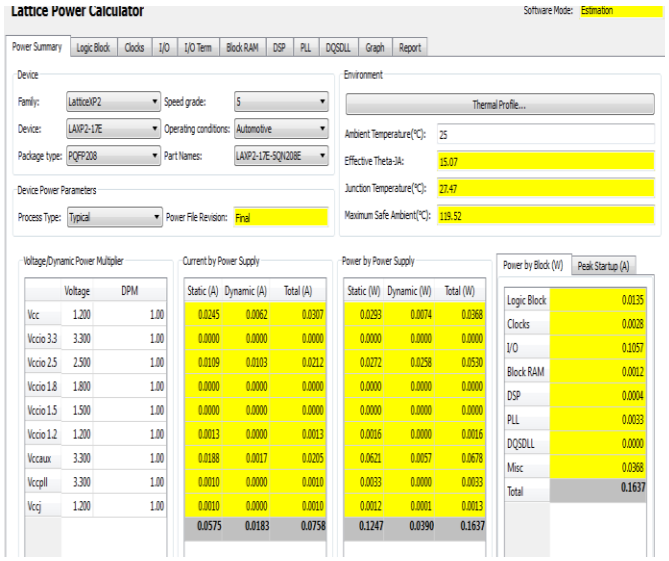

Figure 6 Performance analysis of non-gated architecture on LFXP2-17E device

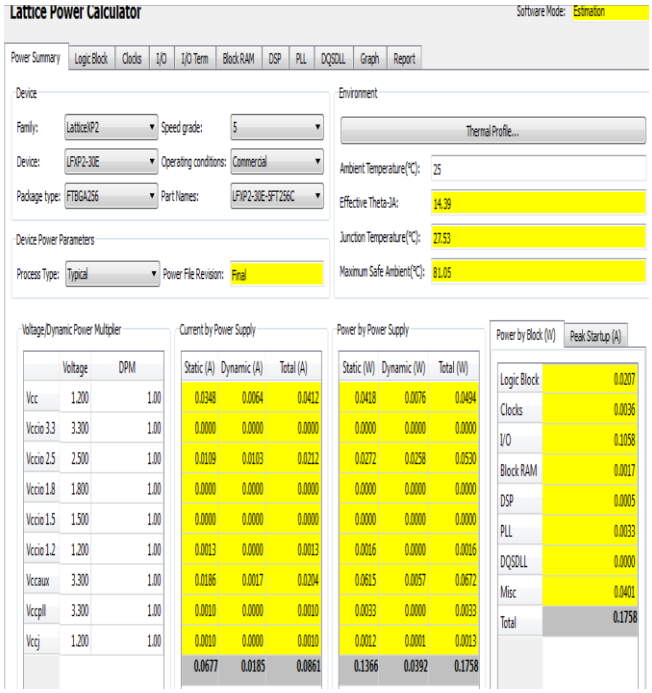

Figure 7 Performance analysis of non-gated architecture on LFXP2-30 E device

The performance parameters to be taken under observation are dynamic power (clock frequency, power supply and logic blocks) and area (number of LUTs). Similarly the performance analysis of the non-gated architecture is carried out on LFXP2-8E, 17E and 30 E.

Table 1 show the area and power calculation of the design on different logic families.

Table -1: Area and power calculation of non-gated design

Lattice FPGA XP2 Family	Area(LUTs)	Dynamic Power (clock frequency) mW	Total Dynamic power (Power supply) mW	Power consumption (Logic blocks) W
LFXP2-5E	197	4.2	39.9	0.1451
LAXP2-8E	197	3.7	38.8	0.1482
LAXP2-17E	197	3.7	39.0	0.1637
LFXP2-30E	197	3.7	39.2	0.1758

Fig.8 shows the hierarchal view of gated SPI-Switch interface**.** The whole design is synthesized on Lattice Diamond XP2 FPGA families i.e. LFXP2-5E, LFXP2-8E, LFXP2-17E and LFXP2-30E for the calculation of area, dynamic power, total dynamic power and power consumption by logic block.

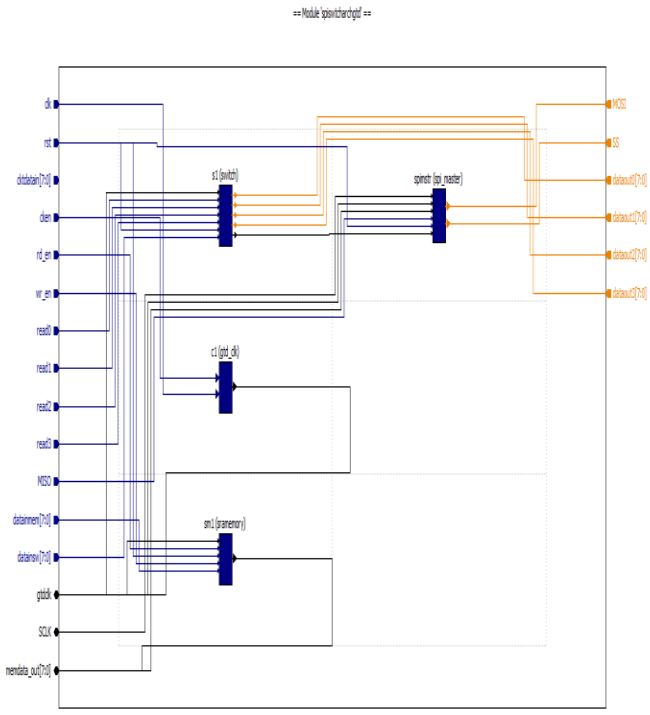

Figure 8 Hierarchal view of gated architetcure

In fig.8, blue blocks are the main blocks i.e. Gated clock, SPI, Switch and RAM. The blue lines are the input ports and wires and orange lines are the output ports.
The Figures 9 shows the performance analysis of the gated architecture on LFXP2-5E.

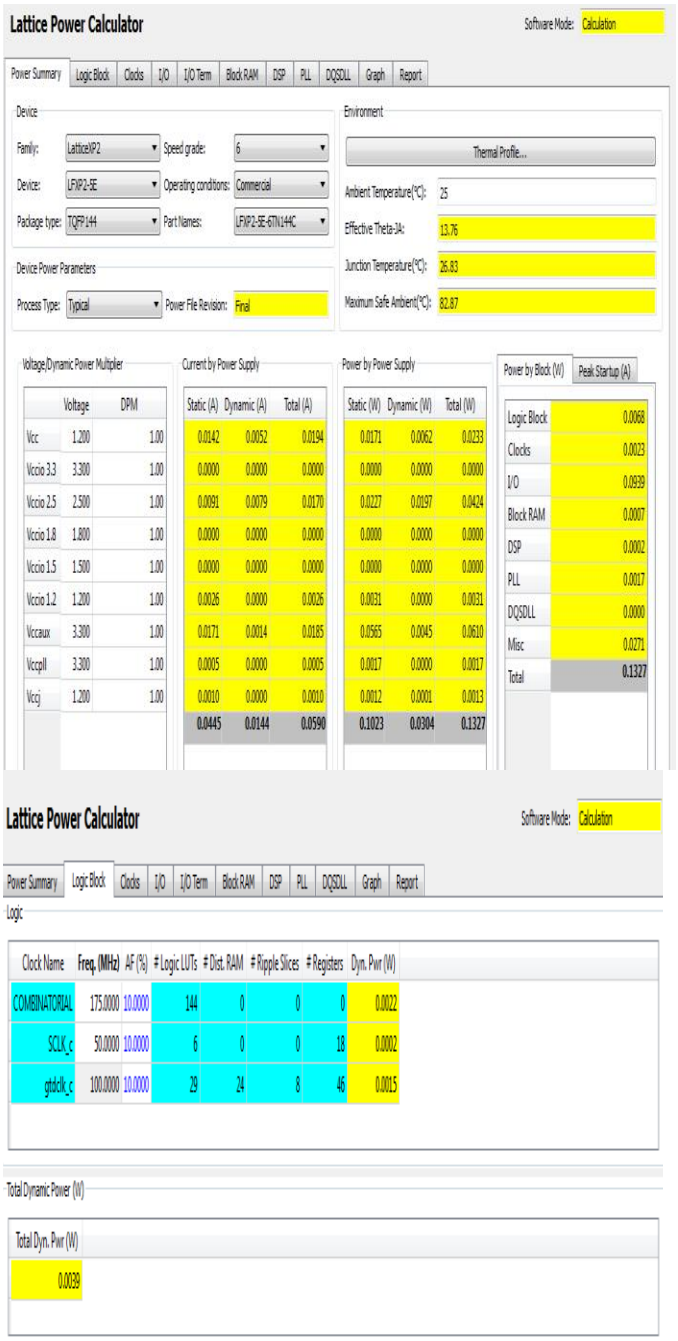

Figure 9 Performance analysis of gated architecture on LFXP2-5E

Figure 10, 11 and 12 shows the performance analysis of gated architecture on other XP2 FPGA families.

The performance parameters to be taken under observation are dynamic power (clock frequency, power supply and logic blocks) and area (number of LUTs). Similarly the performance analysis of the gated architecture is carried out on LFXP2-8E, 17E and 30 E.

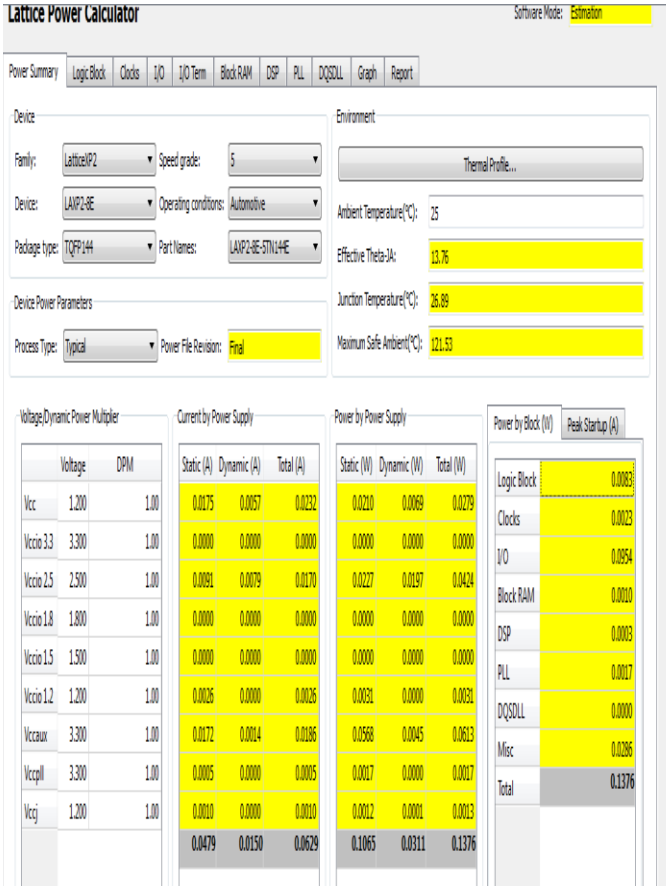

Figure 10 Performance analysis of gated architecture on LFXP2-8E

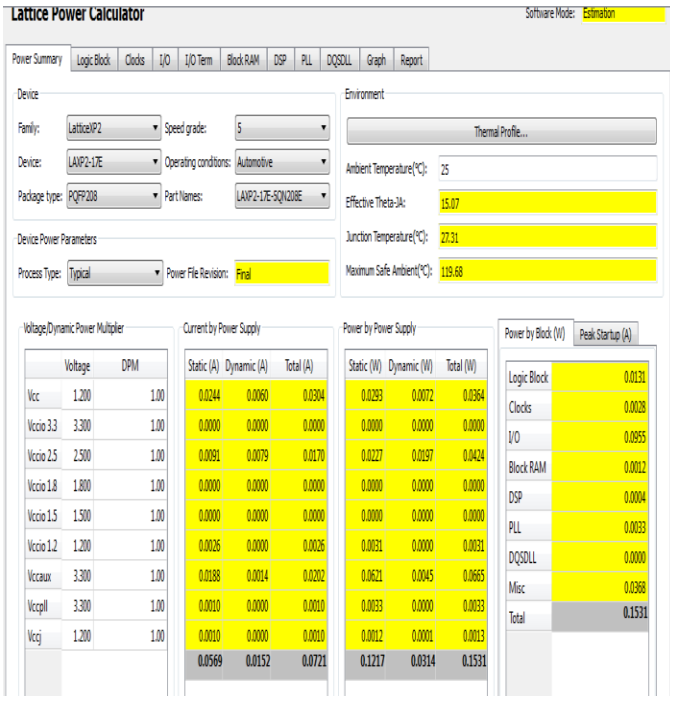

Figure 11 Performance analysis of gated architecture on LFXP2-17E

Figure 12 Performance analysis of gated architecture on LFXP2-30E

Table 2 shows the area and power calculation of the design on different logic families.

Table 2 Area and power calculation of gated design

Lattice FPGA XP2 Family	Area(LUTs)	Dynamic Power (clock frequency) mW	Total Dynamic power (Power supply) mW	Power consumption (Logic blocks) W
LFXP2-5E	179	3.9	30.4	0.1327
LAXP2-8E	197	3.4	31.1	0.1376
LAXP2-17E	197	3.4	31.4	0.1531
LFXP2-30E	179	3.4	31.6	0.1651

VI. CONCLUSION

A practical comparative study of SPI-SWITCH interface has been discussed in this paper. This comparison is limited to the SPI-master-SWITCH interface with gated and non-gated architectures. The research work has shown up the results of FPGA implementation of gated and non-gated SPI- SWITCH interface. Total area and total dynamic power consumption (due to clock frequency, power supply and logic blocks) of gated SPI-Switch interface is less as compared to non-gated architecture. A difference of 9.9 mW in case of total dynamic power and 18 LUTs in case of area has been observed. Gated SPI-switch interface has shown up improvement in overall area and dynamic power consumption. So, overall the designing and implementation of low power SPI-SWITCH interface on FPGA XP2 is carried out successfully.

REFERENCES

(1) Dr. Winncy Du, E310F, MAE Dept., SJSU, " Communication protocols and Interface".
(2) Ashwin Bhandekar, S.S Thakare, D. S. Chaudhar (2013) "A Review on Effectuation of Serial Communication Inter-IC Protocol" , IJARCSSE , ISSN: 2277 128X.
(3) Sukhwinder kaur, Abhishek, Amita," Performance Analysis of SPI on FPGA", IJERT,ISSN2278-081, April 2014.
(4) Frederic Leens," An Introduction to I2C and SPI Protocols", IEEE, 2009.
(5) Paul Myers," Interfacing using Serial Protocols: SPI and I2C", EMRT Consulatants, 2005.
(6) Gurkesar ,Abhishek, Anju," Implementation of I2c-DMA and SPI-DMA interface: A comparative study", IJERA ,June 2014.
(7) Jagrit Kathuria et.al,"A review of Clock Gating Techniques",MITIJEC, Vol2,August 2011. .
(8) Xiaotao Chang, Mingming Zhang, Ge Zhang, Zhimin Zhang, Jun Wang," Adaptive Clock Gating Technique For Low Power IP Core in SOC Design", IEEE, 2007.
(9) SOC Central," Design and Verification Techniques for Clock Gating", 2009.
(10) Frederic Rivoallon," Reducing Switching Power with Intelligent Clock Gating", Xilinx Paper, 2011.
(11) A.K. Oudjida, M.L. Berrandjia, R. Tiar, A. Liacha, K. Tahraoui (2009) "FPGA Implementation of I2C & SPI Protocols: a Comparative Study", CDTA Algiers, Algeria 978-1-4244-5091-6/09 IEEE.
(12) http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus