© 2014 IJIRT | Volume 1 Issue 5 | ISSN : 2349-6002

Implementation of I2C-DMA & SPI-DMA Interface: A Comparative Study

Abhishek Singh, Anjali Kataria
Information Technology Department
Dronacharya College of Engineering, Gurgoan, Haryana

***Abstract-* This paper describes the designing and implementation of I2C (master)-DMA and SPI (master)-DMA interface. Direct Memory Access (DMA), being the feature of modern computers and microprocessors permits certain hardware subsystem in a computer machine to access system memory for reading/writing independently of CPU. Sometimes CPU is fully occupied for entire duration of read/write operation and unavailable to perform other work. With DMA, CPU would initiate data transfers and performs other operations while transfer is in progress. For intra-chip serial data transfers, SPI and I2C are used. Both serial protocols are widely used but both have some merits and demerits. In this research work both protocols are interfaced with DMA for SoC (System on Chip) applications and their performance analysis is carried out by implementing on FPGA. VERILOG-HDL is used for designing RTL Code and test-bench verification. Simulations are observed in Modelsim-Altera 6.3 (Quartus II 8.1) simulator.**

***Index Terms-* SPI, I2C, DMA, FPGA, CPU, SoC**

I. INTRODUCTION

DMA allows an I/O device to communicate directly to the main memory bypassing CPU (Central Processing Unit) in order to speed up memory operations. When the CPU is using programmed input/output (I/O) without DMA, it is fully occupied for the entire duration of the read/write operation, and is thus unavailable to perform other work. When used with DMA, the CPU initiates the transfer, does other operations while the transfer is in progress, and receives an interrupt from the DMA controller (a special purpose processor used to transfer data between memory and I/O devices) when the operation is done. This is especially useful in real-time computing applications where not stalling behind concurrent operations is critical. Another related application area is various forms of stream processing where it is essential to have data processing and transfer in parallel, in order to achieve sufficient throughput. Many hardware systems use DMA including disk drive controllers, graphics cards, network cards and sound cards.

DMA is also used for intra-chip data transfer in multi-core processors, especially in multiprocessor system-on-chips, where its processing element is equipped with a local memory (often called scratchpad RAM) and DMA is used for transferring data between the local memory and the main memory. Computers that have DMA channels can transfer data to and from devices with much less CPU overhead than computers without a DMA channel. Similarly a processing element inside a multi-core processor can transfer data to and from its local memory without occupying its processor time and allowing computation and data transfer concurrency.

II. SERIAL DATA COMMUNICATION: I2C & SPI

Serial data communication has become the standard for inter-computer communication. Inter-Integrated Circuit Bus (I2C) and Serial Peripheral Interface (SPI) are the two best suited protocols for serial communication with on board peripherals. Motorola designed SPI and Philips designed I2C.
The I2C bus was designed in 1982 to provide an easy way to connect a CPU to peripheral chips in a TV. Peripheral devices are often connected to microcontroller as memory mapped I/O devices. A simple way is to connect the peripherals to the microcontroller parallel address and data buses but this will result in lots of wiring on PCB and additional glue logic to decode the address bus. In order to overcome this situation I2C was invented which requires only two wires for connecting the peripherals to the microcontroller. I2C is a multi-master protocol that uses two signal lines; serial data (SDA) and serial clock (SCL). There is no need of chip select (slave select) or arbitration logic. Any number of slaves and any number of masters can be connected onto these two signal

lines and communicate between each other using a 7-bit slave addresses (each device connected to the bus has a unique address). As shown in Fig.1 the I2C bus consists of two wires SDA and SCL and pull up resistors. Both SDA and SCL are open drain drivers i.e. chip can drive its output low but cannot drive it high. For the line to go high pull up resistors are used with 5 volt supply.

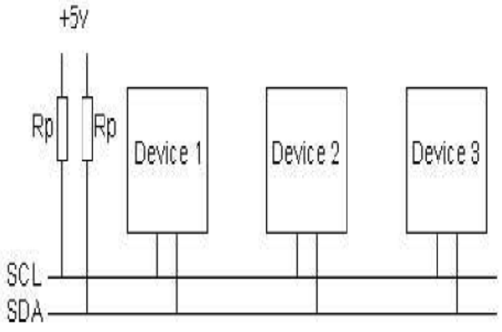


Figure1. I2C Communication

The device which initiates the data transfer is known as bus master and at that time, all other devices are treated as bus slaves. The bus master initiates the data transfer by issuing a START condition. After this master sends the address of the device which it wants to access along with read/write condition. After receiving the address, the devices connected to the bus will match with their own addresses. If the match doesn't occur they will wait till the STOP condition but if the address matches, an ACKNOWLEDGEMENT signal is produced by the chip. After receiving the ACKNOWLEDGMENT signal master initiates the data transfer. When data transfer is complete master issues a STOP condition. Fig.2 shows the START and STOP condition of I2C bus.

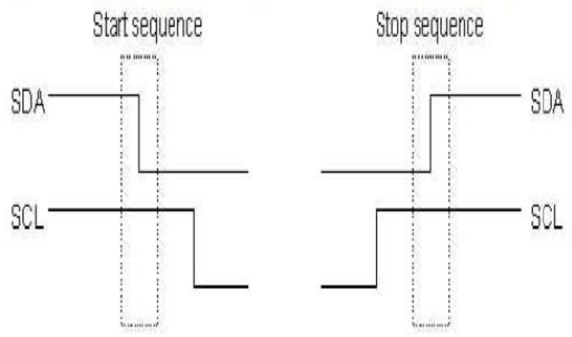


Figure2. START and STOP condition of I2C

Whereas in case of SPI, situation is little bit different. SPI was designed by Motorola (NXP semiconductors) in 1979. It uses 3+N wire where N in the number of slaves. Generally it is used for point to point communication i.e. single master–slave configuration.

Fig.3 shows SPI. There are two wires for data transmission and reception, MOSI (Master Out Slave In) and MISO (Master In Slave Out). SS pin which is active low is used for slave select (in case when number of slave are more than one SS signal may be used depending upon number of slaves). SCLK signal is the SPI clock signal which is generated by Master and data communication will occur on the edges of SCLK.

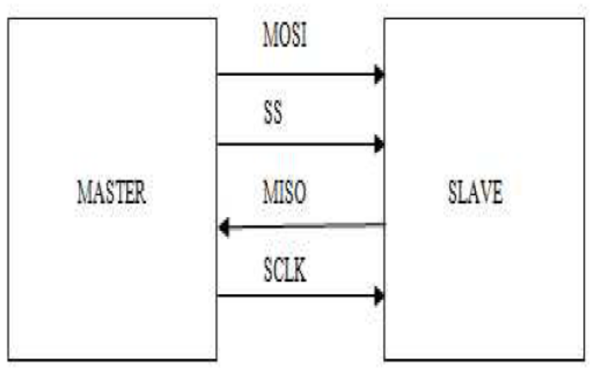


Figure3. SPI Configuration

Selection between I2C and SPI requires a good understanding of advantages and disadvantages of both the protocols. Following points shows the comparison of both the protocols:

- I2C requires only two wires, while SPI requires 3+N wires(N- Number of slaves).
- SPI supports higher speed full-duplex communication while I2C is half duplex and slower.
- I2C supports multiple devices on the same bus without additional select signal lines while SPI requires additional signal lines to manage multiple devices on the same bus (best suited for singles master single slave communication).
- I2C ensures that data sent is received by the slave device (Acknowledgement signal) while SPI does not verify that data is received correctly or not.
- I2C is cheaper to implement than the SPI communication protocol.
- SPI only supports one master device on the bus while I2C supports multiple master devices.
- I2C is less susceptible to noise than SPI.

- I2C can transmit data over much greater distances, although at low data rates as compared to SPI.

From the above discussion, both SPI and I2C are good communication options, but each has a few distinct advantage and preferred applications. Overall, SPI is better for high speed and low power applications while I2C is better for suited to communication with a large number of peripherals and dynamic changing of the master device role among the peripherals on the I2C bus.

Both SPI and I2C are robust, stable communication protocols for embedded applications that are well suited for the embedded world.

III. SYSTEM ARCHITECTURE

The Fig.4 shows the proposed system architecture comprising of I2C master/SPI master module, Transmit and receive interface and DMA block with transmit and receive FIFO. In this architecture both SPI master and I2C master are working in half duplex mode. Both IPs communicates with DMA's Transmit and Receive FIFO through transmit and receive IP interfaces. At one instance I2C master is interfaced with DMA and then SPI master is interfaced with DMA. Both IPs transmit data through TX register and receive data through RX register.

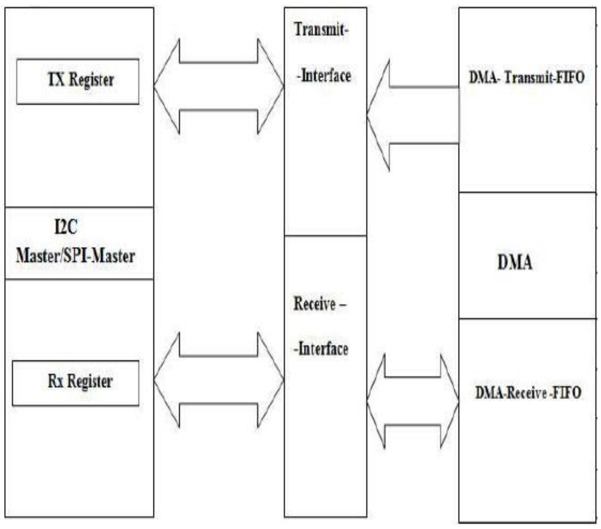


Figure4. DMA-I2C/SPI Interface

IV. RESULTS AND SIMULATIONS

The RTL view and Hierarchal view of I2C master is shown in Fig.5.

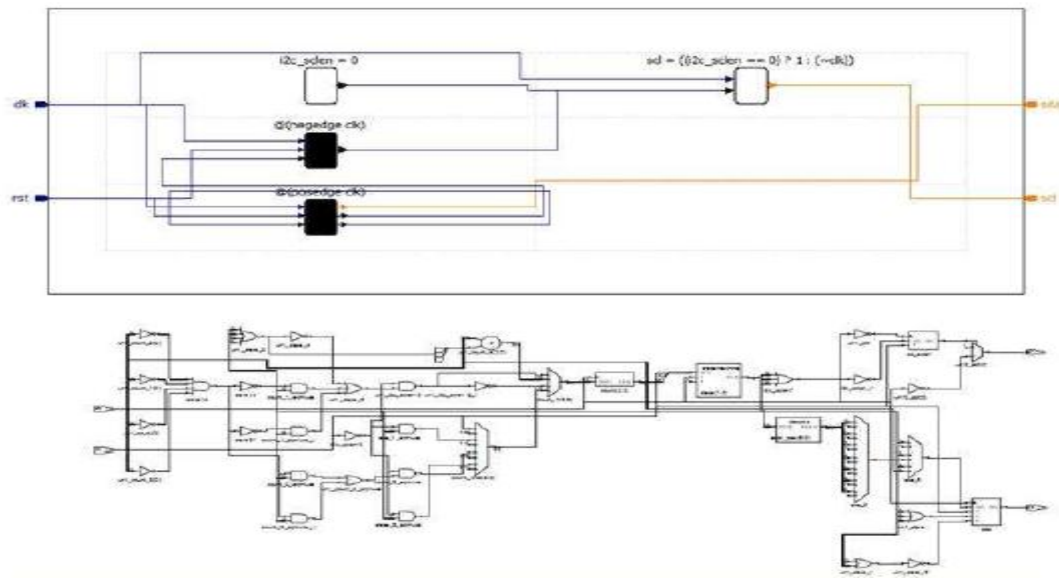


Figure5. Hierarchal and RTL view of I2C master

The hierarchal view is obtained on Lattice XP2 FPGA kit and RTL view is obtained on Synplify Pro tool. Fig.6 (a), 6(b) and 6 (c) shows the test bench simulation of I2C-master on Modelsim, calculated dynamic power, number of LUTs, Data path delay.

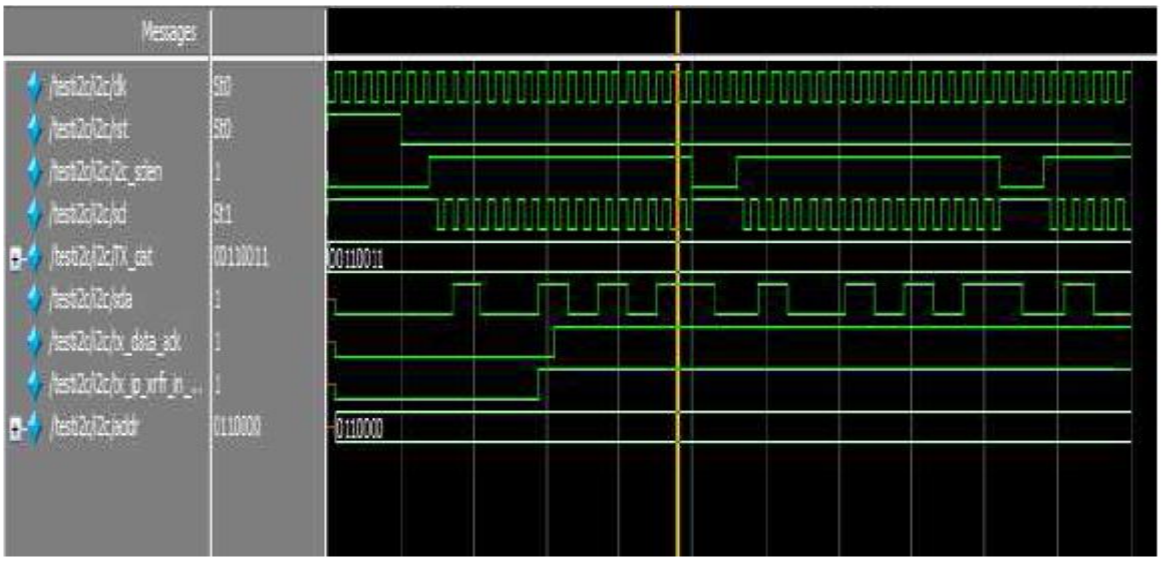
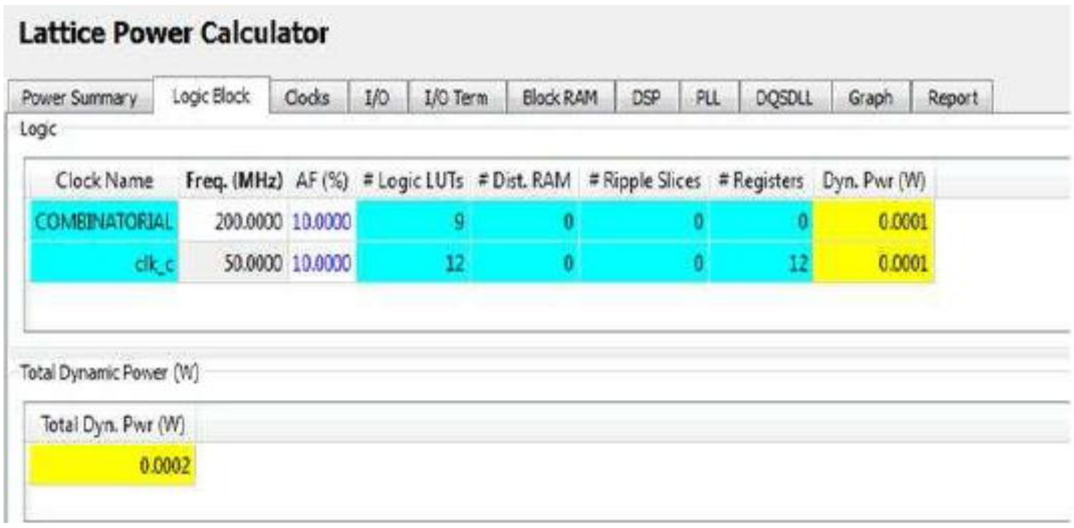


Figure6 (a). Test-bench waveform of I2C-Master

As shown in Fig. 6 (a), I2C- master consists of mainly 4 ports i.e. clock, Reset, SDA and SCL. TX_dat is the transmit data register. The data i.e. 00110011 is transmitted through SDA w.r.t. SCL.



Logic

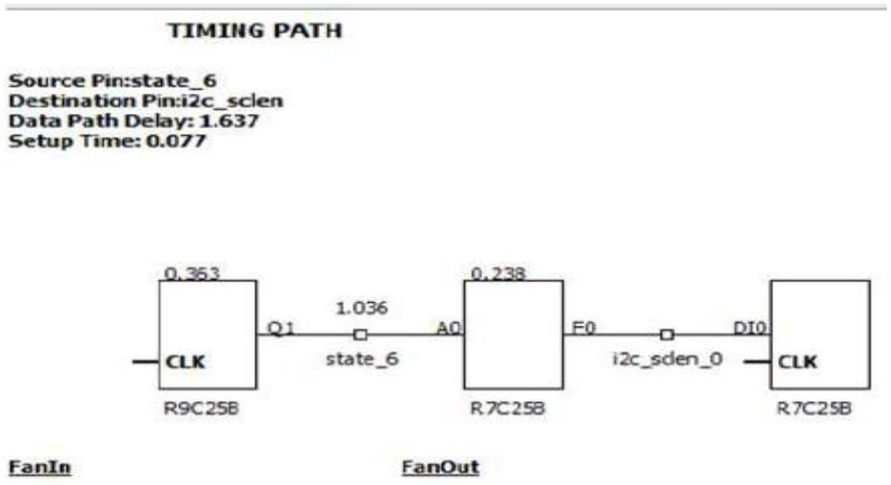
Clock Name	Freq. (MHz)	AF (%)	# Logic LUTs	# Dist. RAM	# Ripple Slices	# Registers	Dyn. Pwr (W)
COMBINATORIAL	200.0000	10.0000	9	0	0	0	0.0001
clk_c	50.0000	10.0000	12	0	0	12	0.0001

Total Dynamic Power (W)

Total Dyn. Pwr (W)
0.0002

Figure6 (b). Number of LUTs and Dynamic power calculations

The total number of LUTs consumed by the design is 21, dynamic power is 0.2mW and data path delay is 1.63ns.


TIMING PATH

Source Pin:state_6
Destination Pin:i2c_sclen
Data Path Delay: 1.637
Setup Time: 0.077

Figure6 (c). Timing Path considerations

Fig.7 shows the hierarchal and RTL view of SPI master on FPGA board. The hierarchal view is obtained on Lattice XP2 FPGA kit and RTL view is obtained on Synplify Pro tool.

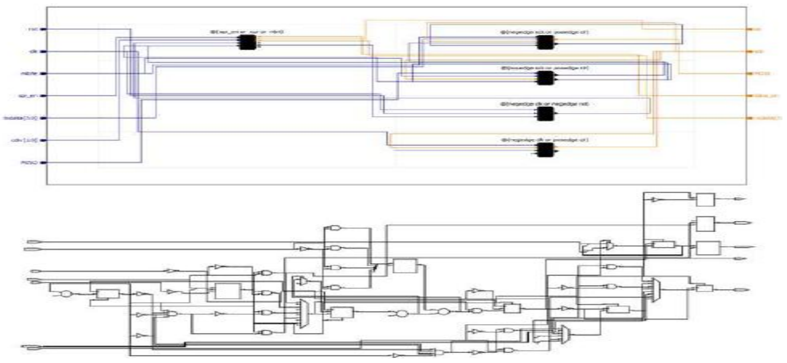


Figure7. Hierarchal and RTL view of SPI-Master

Fig. 8(a), 8(b) & 8(c) shows the test-bench simulation of SPI-master on Modelsim, number of LUTs, dynamic power consumed and timing considerations.

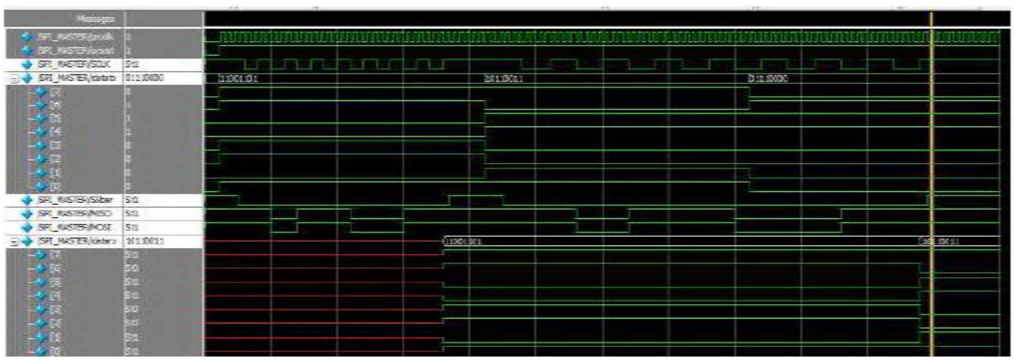
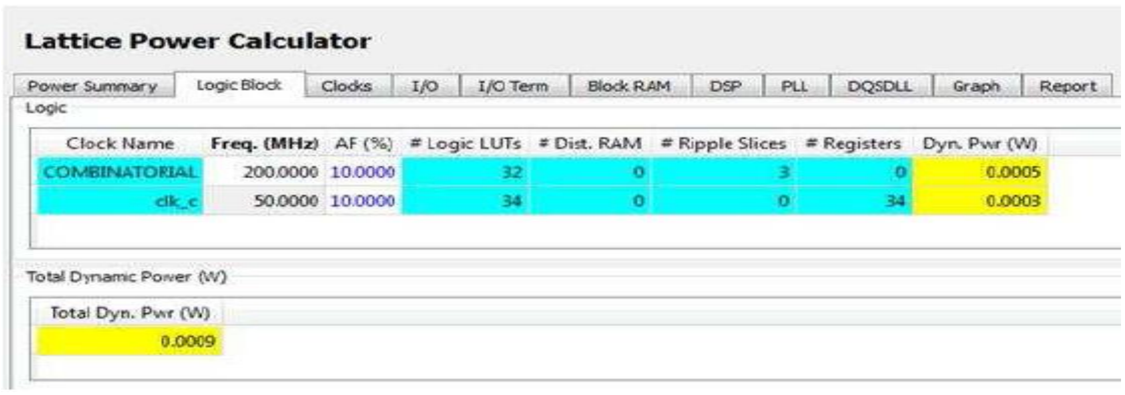


Figure8 (a). Test-bench waveform of SPI-Master

SPI-master transmits the data through MOSI and receives through MISO. The highlighted ports in test-bench waveform shows SCLK, data tx (transmit data), SSbar, MISO, MOSI and data rx (receive data). Here, SCLK is divide by 4 of proclk (system clock) depending upon the different baud rate divisor values.



Lattice Power Calculator

Power Summary | Logic Block | Clocks | I/O | I/O Term | Block RAM | DSP | PLL | DQSDLL | Graph | Report

Logic

Clock Name	Freq. (MHz)	AF (%)	# Logic LUTs	# Dist. RAM	# Ripple Slices	# Registers	Dyn. Pwr (W)
COMBINATORIAL	200.0000	10.0000	32	0	3	0	0.0005
clk_c	50.0000	10.0000	34	0	0	34	0.0003

Total Dynamic Power (W)

Total Dyn. Pwr (W)
0.0009

Figure8 (b). Number of LUTs and dynamic power calculation

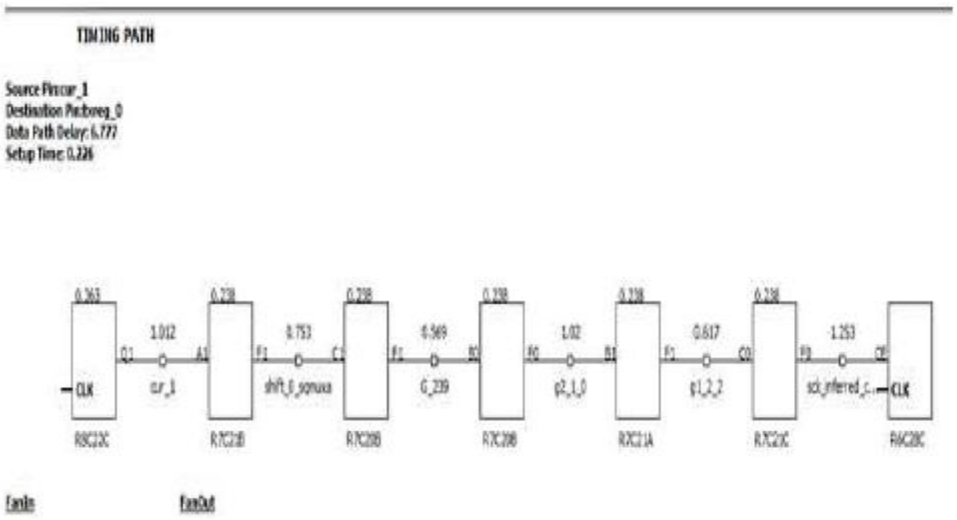


Figure8 (c). Timing Path considerations

The total dynamic power consumed is 0.9mW, number of LUTs are 66 and data path delay is 6.77 ns. Now, as compared to I2C master, all parameters are greater in number. So, from above discussion I2C consumes low power, less area (number of LUTs) and lesser delay.

Fig.9 shows the RTL view of I2C-DMA transmit interface. The synthesized RTL view is obtained in Synplify-Pro synthesis tool. As shown, it comprises of I2C master, DMA transmit FIFO (Tx FIFO) and Transmit IP interface (Tx_IP_Interface). I2C master communicates with DMA through transmit and receive interfaces.

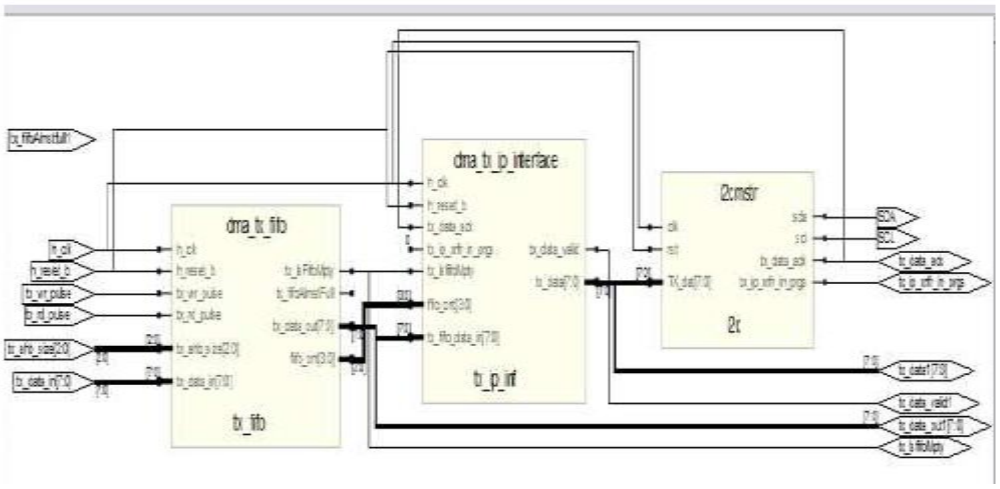
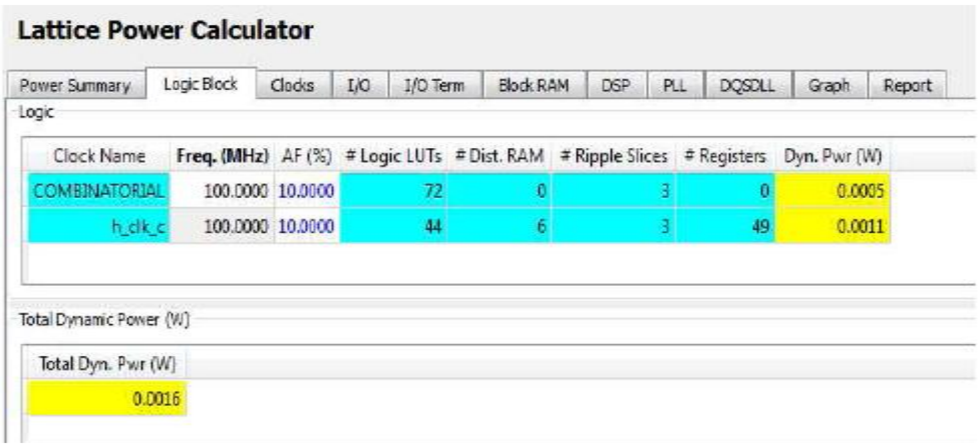


Figure9. I2C-DMA TX-interface

Fig.10 shows the power and area calculation of the design on Lattice diamond XP2 FPGA.


Lattice Power Calculator

Power Summary | Logic Block | Clocks | I/O | I/O Term | Block RAM | DSP | PLL | DQSDLL | Graph | Report

Logic

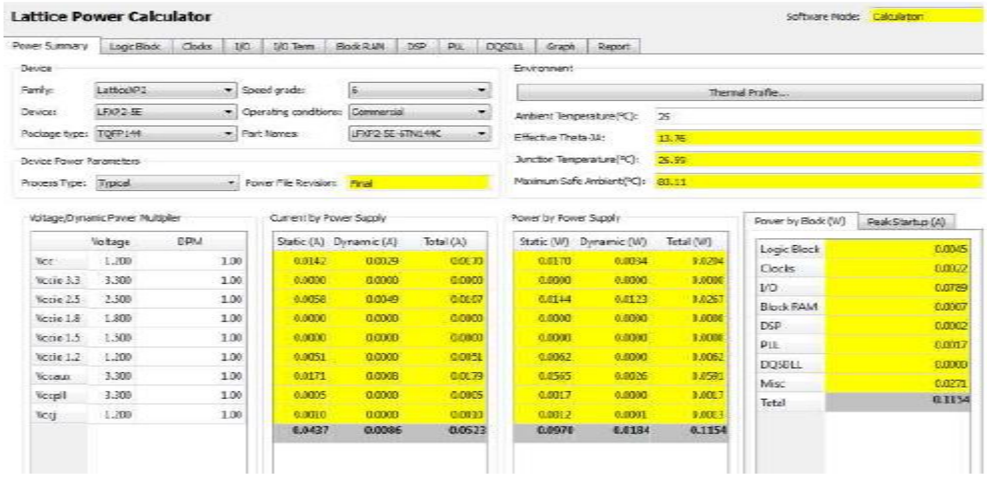
Clock Name	Freq. (MHz)	AF (%)	# Logic LUTs	# Dist. RAM	# Ripple Slices	# Registers	Dyn. Pwr (W)
COMBINATORIAL	100.0000	10.0000	72	0	3	0	0.0005
h_clk_c	100.0000	10.0000	44	6	3	49	0.0011

Total Dynamic Power (W)

Total Dyn. Pwr (W)
0.0016

Figure10. Power and area calculation of I2CDMA interface

At frequency of 100 MHz (h_clk_c) the area (LUTs) consumed by the design is 116 LUTs and dynamic power by both the clocks is 1.6mW. The total power consumed by the design is shown in Fig.11.


Lattice Power Calculator

Software Mode: Calculation

Power Summary | Logic Block | Clocks | I/O | I/O Term | Block RAM | DSP | PLL | DQSDLL | Graph | Report

Device

Family: LatticeXP2
Speed grade: 6
Device: LFXP2-5E
Operating conditions: Commercial
Package type: TQFP144
Part Names: LFXP2-5E-6TN144C

Device Power Parameters

Process Type: Typical
Power File Revision: Final

Environment

Thermal Profile...

Ambient Temperature(°C): 25
Effective Theta-JA: 13.76
Junction Temperature(°C): 26.99
Maximum Safe Ambient(°C): 83.11

Voltage/Dynamic Power Multiplier

	Voltage	DPM
Vcc	1.200	1.00
Vccio 3.3	3.300	1.00
Vccio 2.5	2.500	1.00
Vccio 1.8	1.800	1.00
Vccio 1.5	1.500	1.00
Vccio 1.2	1.200	1.00
Vccaux	3.300	1.00
Vccpll	3.300	1.00
Vccj	1.200	1.00

Current by Power Supply

Static (A)	Dynamic (A)	Total (A)
0.0142	0.0029	0.0170
0.0000	0.0000	0.0000
0.0058	0.0049	0.0107
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000
0.0051	0.0000	0.0051
0.0171	0.0008	0.0179
0.0005	0.0000	0.0005
0.0010	0.0000	0.0010
0.0437	0.0086	0.0523

Power by Power Supply

Static (W)	Dynamic (W)	Total (W)
0.0170	0.0034	0.0204
0.0000	0.0000	0.0000
0.0144	0.0123	0.0267
0.0000	0.0000	0.0000
0.0000	0.0000	0.0000
0.0062	0.0000	0.0062
0.0565	0.0026	0.0591
0.0017	0.0000	0.0017
0.0012	0.0001	0.0013
0.0970	0.0184	0.1154

Power by Block (W) | Peak Startup (A)

Logic Block	0.0045
Clocks	0.0022
I/O	0.0789
Block RAM	0.0007
DSP	0.0002
PLL	0.0017
DQSDLL	0.0000
Misc	0.0271
Total	0.1154

Figure11. Total power consumption

The total dynamic power consumption of I2C-DMA interface is 18.4mW. Also, power consumed by different blocks (logic blocks, PLL, IO and RAM) is 0.1154 W. Maximum frequency for clock (h_clk) is 150.580 MHz.

Fig. 12 shows the SPI-DMA transmit interface. Similar to I2C-DMA interface, SPI also communicates with DMA through transmit and receive interfaces.

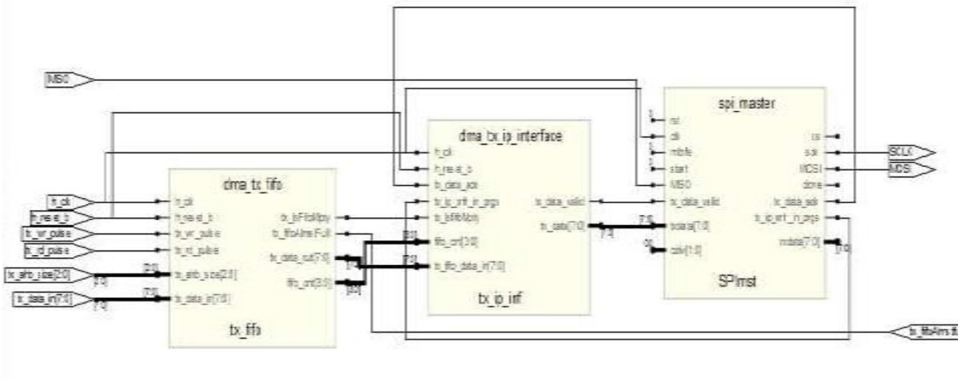
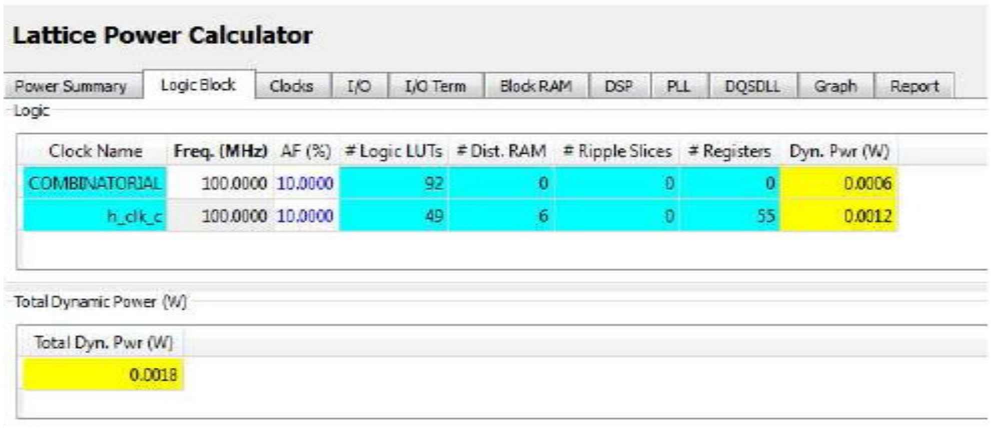


Figure12. SPI-DMA TX-interface

Figure 13 shows the power and area calculation of the design on Lattice diamond XP2 FPGA.


Lattice Power Calculator

Power Summary | Logic Block | Clocks | I/O | I/O Term | Block RAM | DSP | PLL | DQSDLL | Graph | Report

Logic

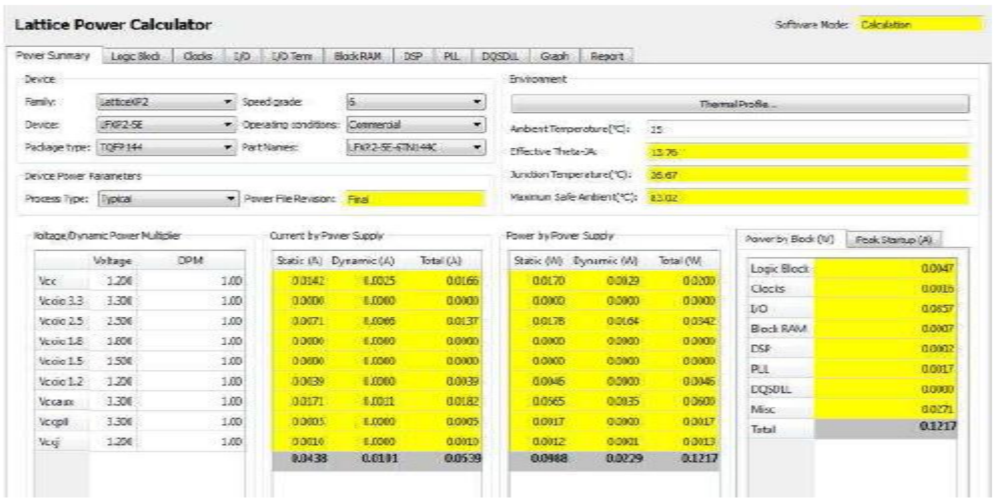
Clock Name	Freq. (MHz)	AF (%)	# Logic LUTs	# Dist. RAM	# Ripple Slices	# Registers	Dyn. Pwr (W)
COMBINATORIAL	100.0000	10.0000	92	0	0	0	0.0006
h_clk_c	100.0000	10.0000	49	6	0	55	0.0012

Total Dynamic Power (W)

Total Dyn. Pwr (W)
0.0018

Figure 13 Power and area calculation of SPI-DMA interface

At the frequency of 100 MHz (h_clk_c) the area (LUTs) consumed by the design is 141 LUTs and dynamic power by both the clocks is 1.8mW. The total power consumed by the design is shown in Fig.14


Lattice Power Calculator

Software Mode: Calculation

Power Summary | Logic Block | Clocks | I/O | I/O Term | Block RAM | DSP | PLL | DQSDLL | Graph | Report

Device

Family: LatticeXP2
Speed grade: 5
Device: LFXP2-5E
Operating conditions: Commercial
Package type: TQFP144
Part Names: LFXP2-5E-6TN144C

Device Power Parameters

Process Type: Typical
Power File Revision: Final

Environment

Thermal Profile...

Ambient Temperature(°C): 25
Effective Theta-JA: 13.76
Junction Temperature(°C): 26.67
Maximum Safe Ambient(°C): 83.02

Voltage/Dynamic Power Multiplier — Current by Power Supply — Power by Power Supply

	Voltage	DPM	Static (A)	Dynamic (A)	Total (A)	Static (W)	Dynamic (W)	Total (W)
Vcc	1.20V	1.00	0.0142	0.0025	0.0166	0.0170	0.0029	0.0200
Vccio 3.3	3.30V	1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Vccio 2.5	2.50V	1.00	0.0071	0.0066	0.0137	0.0178	0.0164	0.0342
Vccio 1.8	1.80V	1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Vccio 1.5	1.50V	1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
Vccio 1.2	1.20V	1.00	0.0039	0.0000	0.0039	0.0046	0.0000	0.0046
Vccaux	3.30V	1.00	0.0171	0.0011	0.0182	0.0565	0.0035	0.0600
Vccpll	3.30V	1.00	0.0005	0.0000	0.0005	0.0017	0.0000	0.0017
Vccj	1.20V	1.00	0.0010	0.0000	0.0010	0.0012	0.0001	0.0013
			0.0438	0.0101	0.0539	0.0988	0.0229	0.1217

Power by Block (W) | Peak Startup (A)

Logic Block	0.0047
Clocks	0.0016
I/O	0.0857
Block RAM	0.0007
DSP	0.0002
PLL	0.0017
DQSDLL	0.0000
Misc	0.0271
Total	0.1217

Figure14. Total power consumption

The total dynamic power consumption of I2C-DMA interface is 22.9mW. Also, power consumed by different blocks (logic blocks, PLL, IO,RAM) is 0.1217 W. Maximum frequency for clock(h_clk) is 162.048 MHz

The performance comparison between both the protocols and I2C-DMA and SPI-DMA interface is shown in table1.

Table 1 Performance comparison

Lattice XP2 (Performance parameters)	**I2CMaster (50 MHz)**	**SPIMaster (50 MHz)**	**I2CDMA (100 MHz)**	**SPIDMA (100 MHz)**
AREA (LUTs)	21	66	116	141
POWER (mW)	0.2	0.9	1.6	1.8
Delay(ns)	1.637	6.77	4.898	4.617

V. CONCLUSION

A practical comparative study has been discussed. This comparison is limited to the master side of I2C and SPI. Both protocols are interfaced with DMA in transmit mode. The paper has shown up the results of FPGA implementation of I2C/SPI –DMA interface (transmit mode). Area, power and delay of I2C master is better than SPI-master. I2C master-DMA interface consumes less area (LUTs) and low power as compared to SPI master-DMA interface. A difference of 45 LUTs has been observed between both the designs. But, in case of delays SPI master-DMA interface is better. So, both the designs have their own importance in SoC communication.

REFERENCES

[1] www.wikipedia.com/DMA.

[2] Module3 Embedded systems I/O /DMA, Version 2 EE, IIT, Kharagpur.

[3] Design of Digital and Microcomputer systems, Winter session term 2,” Direct Memory Access”.

[4] A.K Oudjida, M.L. Berandjia, R. Tiar, A.Liacha, K. Tahraoui,” FPGA implementation of I2C and SPI Protocols: A comparative study”, IEEE, 2009.

[5] “I2C Tutorial, ROBOT Electronics.

[6] Tomas Matousek,” Inter Integrated Circuit Bus by Philips Semiconductors”.

[7] Frederic Leens,” An Introduction to I2C and SPI Protocols”, IEEE, 2009.

[8] Paul Myers,” Interfacing using Serial Protocols: SPI and I2C”, EMRT Consultants, 2005.