SPEED-POWER EXPLORATION OF 2-D INTELLIGENCE NETWORK-ON-CHIP FOR MULTI-CLOCK MULTI-MICROCONTROLLER ON 28nm FPGA (Zynq-7000) DESIGN

Anoop Kumar Vishwakarma[1], Uday Arun[2]

[1]*Student (M.Tech.), ECE, ABES Engineering College Ghaziabad, UP, India*
[2]*Membership, IEEE Circuit and System Society and ACM Embedded System Computing Society*

Abstract

Today's feature-rich multimedia products require embedded system solution with complex System-on-Chip (SoC) to meet market expectations of high performance at low cost and lower energy consumption. SoCs are complex designs with multiple embedded processors, memory subsystems, and application specific peripherals. The bus arbitration and synchronization of Multi-Microcontroller System-on-Chip (MMSoC) strongly influences the area, power and performance of the entire system as it uses switching. The synchronization is generally done by multi clocking which leads to power consumption. Future MMSoC designs will need novel on-chip communication architectures that can provide scalable and reliable data transport. On-chip network architectures are believed to be the ideal solution to many of today's SoC interconnection problems. Network-on-Chip (NoC) architectures may adopt design concepts and methodologies from Multi-Processor/Multi-Microcontroller architectures. Nevertheless, silicon implementation of networks requires a different perspective, because network architectures have to deal with the advantages and limitations of the silicon fabric. These characteristics will require new methodologies for both on-chip switch designs as well as routing algorithm designs. We envision that future on-chip systems will be communication-centric, in particular, energy and performance issues in designing the Multi-Processor/Multi-Microcontroller System-on-Chip will become challenging. In our work we have designed a switch-box for MMSoC for bus synchronization.

In this paper, we explore several critical aspects in the bus synchronization of MMSoC by using the generic Multi-Microcontroller System-on-Chip architecture as the experimental platform; this paper presents both quantitative and qualitative analysis on bus synchronization for Network-on-Chip (NoC). Commonly, we synchronize data bus only as it is more important over the other buses namely address bus, control bus and status bus. But by synchronizing other buses we can improve the overall performance of the entire Multi-Microcontroller System-on-Chip. New methodologies and solutions are also proposed to achieve better performance and power balance for MMSoCs.

Keywords: *SoC, MMSoC, NoC.*

----------***----------

1. INTRODUCTION

Current time embedded systems are increasingly based on Multi-Processor System-on-chip (MPSoC). These MPSoCs typically contain multiple storage elements (SEs), networks (NEs), I/O components, and a number of heterogeneous programmable processors for flexible application support as well as dedicated processing elements (PEs) for achieving high performance and power goals MPSoCs have been widely used in today's high performance embedded systems, such as network processors (NP), Mobile Phones (MP) and parallel media processors (PMP). They combine the advantages of data processing parallelism of multi-processors and the high level integration of system-on-chip (SoC). Driven by the advancement in semiconductor technology, future SoCs will continue to accelerate in system's complexity and capacity. SoCs in the next decade are expected to integrate hundreds, or even more of, processing elements (PEs) and/or storage elements (SEs) on a single chip.

But low cost embedded systems are limited to the microcontroller performance. To improve the performance of low cost embedded system we have to develop Multi-Microcontroller System-on-Chip. So we are migrating number of low cost Microcontrollers to a single chip. To achieve it we can use the generic architecture of MMSoC (Figure1). In generic MMSoC architecture microcontrollers are connected to the global system bus by using a switch box which is controlled by an arbitrator. To improve the performance the connected microntrollers operate parallely. As in most of cases only data bus is synchronized but by synchronizing all buses in time shared mode will improve performance and speed of MMSoC.

In order to cope with the design complexity of such systems in a time-efficient way, the abstraction level of the design process has in recent years been raised towards the system level. Design Space Exploration (DSE) is a key ingredient of such system-level design, during which a wide range of design choices are explored, especially during the early design stages. Therefore, such early design choices heavily

influence the success or failure of the final product, and we can avoid wasting time and effort in further design steps without the possibility of meeting design requirements because of an inappropriate system architecture design.

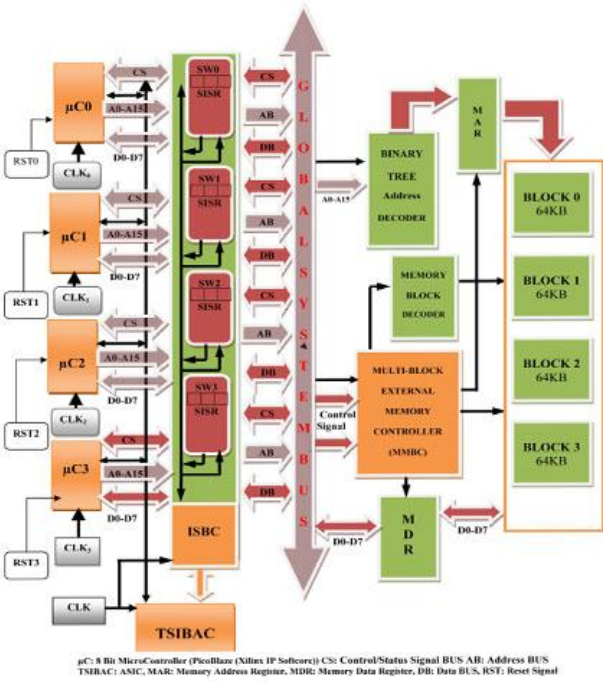


Fig -1: Generic Architecture of multi-Micro Controller System-On-Chip

2. ANATOMY OF BUS SYSTEM AT BOARD, BACKPLANE AND I/O LEVEL

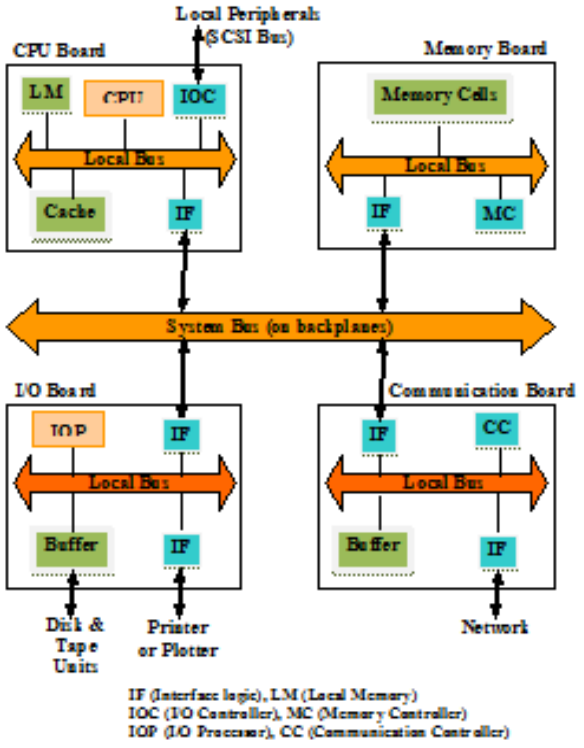


Fig -2: Bus system at board backplane and I/O level

3. ANATOMY OF BUS-CONNECTED MULTI-PROCESSOR SYSTEM

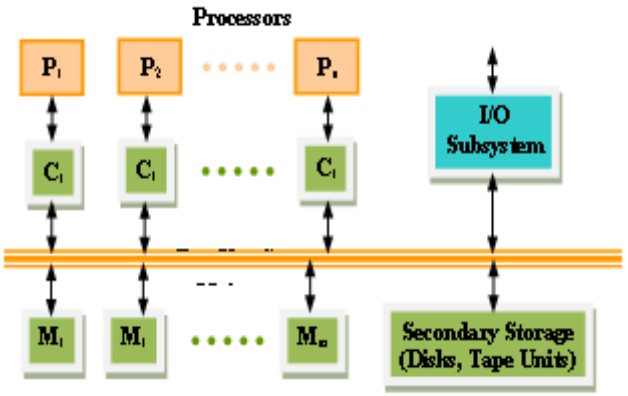


Fig -3: A Bus-connected multiprocessor system, such as the Sequent Symmetry

4. BUS ARBITRATION MODEL

Bus arbitration models are used in MPSoC/MMSoC systems. These models have only data bus is synchronized.

4.1 Central Arbitration

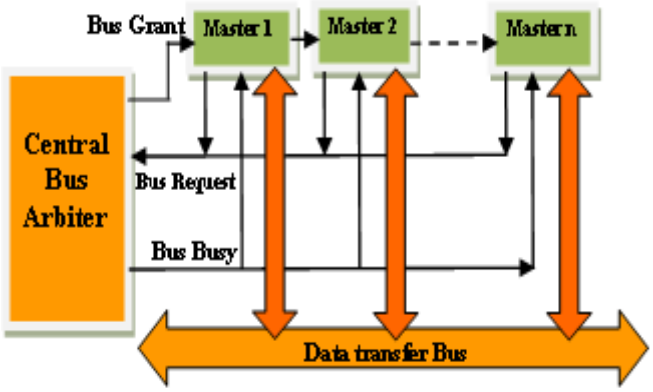


Fig -4: Central Arbitration

4.2 Independent Request with Central Arbiter

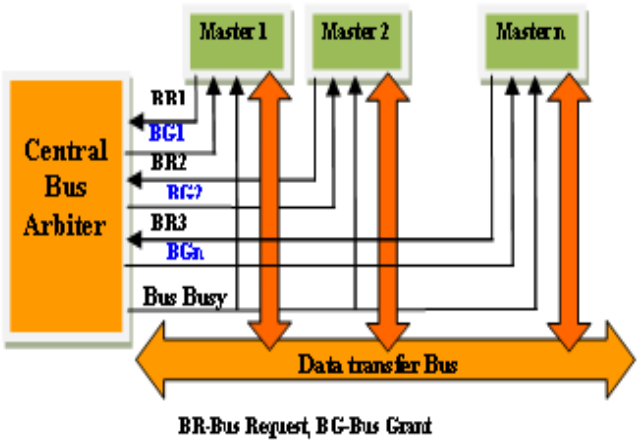


Fig -5: Independent Request with Central Arbiter

4.3 Independent Request with Distributed Arbiter

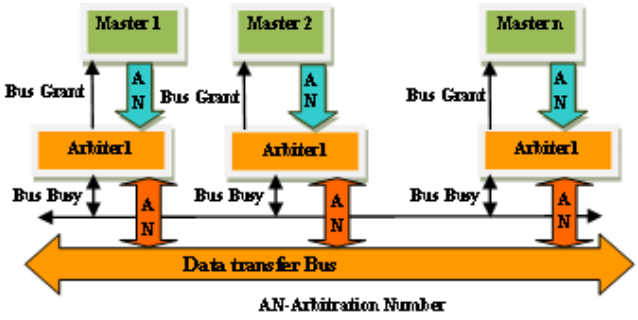


Fig -6: Independent Request with Central Arbiter

5. ANATOMY OF SWITCH BOX

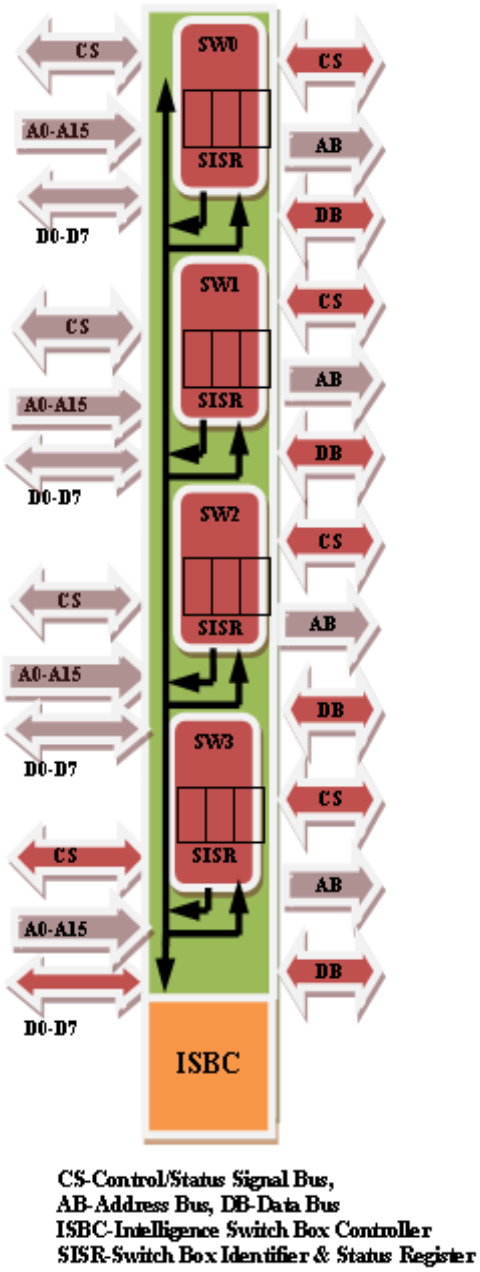


Fig -7: Anatomy of Switch Box

6. RELATED WORK AND DESIGN ANALYSIS

The modeling and designing of intelligence switch box is done. The intelligence switch box shown in fig.7 has been successfully simulated and synthesized for Zynq-7000 by Xilinx Vivado 2013.4.

6.1 RTL Modeling of Intelligence Switch Box

SW00:process(ISB_Clk,Load_Addr,ISBC_ISB_DB_DIR,ISBC_ISB_CB_DIR,ISBC_ISB_SB_DIR,LSB_AddrBus_IN00,LSB_DataBus00,LSB_CntlBus00,LSB_StatusBus00,ISBC_TO_ISB_SBS,ISBC_TO_ISB_MCID,
ISBC_TO_ISB_BUSID)
variable SISR00 : std_logic_vector (2 downto 0):= "00Z";
variable ISBC_TO_ISB00_MCID : std_logic_vector (1 downto 0):= "ZZ";
Begin

End Process

SW01:process(ISB_Clk,Load_Addr,ISBC_ISB_DB_DIR,ISBC_ISB_CB_DIR,ISBC_ISB_SB_DIR,LSB_AddrBus_IN01,LSB_DataBus01,LSB_CntlBus01,LSB_STATUSBUS01,ISBC_TO_ISB_SBS,ISBC_TO_ISB_MCID,ISBC_TO_ISB_BUSID)
variable SISR01 : std_logic_vector (2 downto 0):= "00Z";
variable ISBC_TO_ISB01_MCID : std_logic_vector (1 downto 0):= "ZZ";

begin

End Process

SW10:process(ISB_Clk,Load_Addr,ISBC_ISB_DB_DIR,ISBC_ISB_CB_DIR,ISBC_ISB_SB_DIR,LSB_AddrBus_IN02,LSB_DataBus02,LSB_CntlBus02,LSB_STATUSBUS02,ISBC_TO_ISB_SBS,ISBC_TO_ISB_MCID,ISBC_TO_ISB_BUSID)
variable SISR10 : std_logic_vector (2 downto 0):= "00Z";
variable ISBC_TO_ISB10_MCID : std_logic_vector (1 downto 0):= "ZZ";

begin

End Process

SW11:process(ISB_Clk,Load_Addr,ISBC_ISB_DB_DIR,ISBC_ISB_CB_DIR,ISBC_ISB_SB_DIR,LSB_AddrBus_IN03,LSB_DataBus03,LSB_CntlBus03,LSB_STATUSBUS03,ISBC_TO_ISB_SBS,ISBC_TO_ISB_MCID,ISBC_TO_ISB_BUSID)
variable SISR11 : std_logic_vector (2 downto 0):= "00Z";
variable ISBC_TO_ISB11_MCID : std_logic_vector (1 downto 0):= "ZZ";

```
begin
----
----
----
End Processs
```

6.2 RTL Modeling of Latch Box

```
Port (
LBx_Clk_in : in STD_LOGIC;
LBx_TriStateBuffer_En : in STD_LOGIC;
LBx_LAdd_in : in STD_LOGIC_VECTOR (15 downto 0);
LBx_LAdd_Out : out STD_LOGIC_VECTOR (15 downto
0);

LBx_LData_In_LGSB : in STD_LOGIC_VECTOR (7
downto 0);
LBx_LData_Out_LGSB : out STD_LOGIC_VECTOR (7
downto 0);
LBx_LData_In_GSBL : in STD_LOGIC_VECTOR (7
downto 0);
LBx_LData_Out_GSBL : out STD_LOGIC_VECTOR (7
downto 0);
LBx_Latch_DB_DIR : in STD_LOGIC;

LBx_LStatus_In_LGSB : in STD_LOGIC_VECTOR (1
downto 0);
LBx_LStatus_Out_LGSB : out STD_LOGIC_VECTOR (1
downto 0);
LBx_LStatus_In_GSBL : in STD_LOGIC_VECTOR (1
downto 0);
LBx_LStatus_Out_GSBL : out STD_LOGIC_VECTOR (1
downto 0);
LBx_Latch_SB_DIR : in STD_LOGIC;

LBx_LCtrl_In_LGSB : in STD_LOGIC_VECTOR (5
downto 0);
LBx_LCtrl_Out_LGSB : out STD_LOGIC_VECTOR (5
downto 0);
LBx_LCtrl_In_GSBL : in STD_LOGIC_VECTOR (5
downto 0);
LBx_LCtrl_Out_GSBL : out STD_LOGIC_VECTOR (5
downto 0);
LBx_Latch_CB_DIR : in STD_LOGIC
);
```

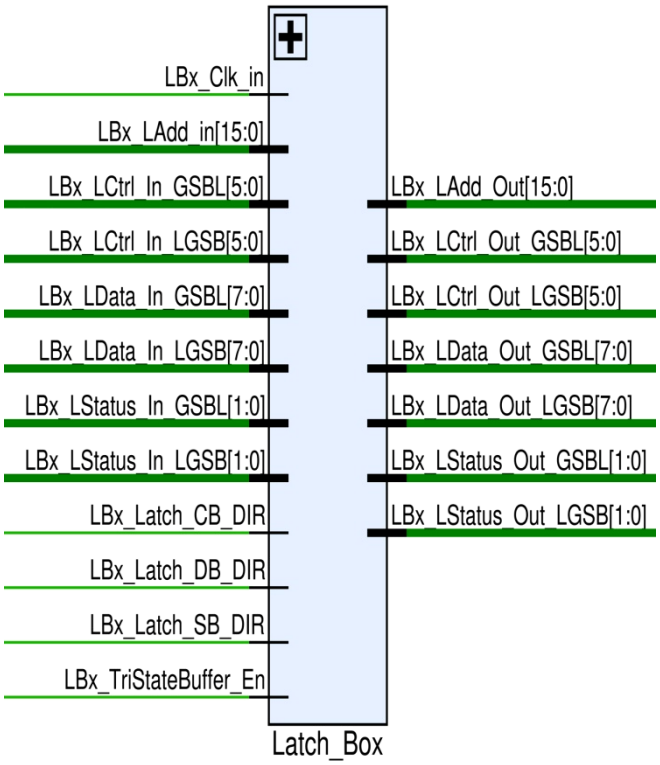


Fig -8: RTL Schematic of latch Box

7. RESULT

7.1 Utilization Graph

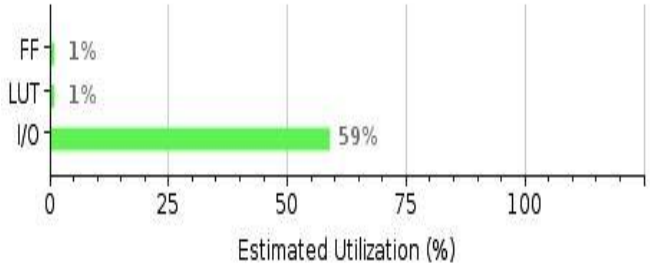


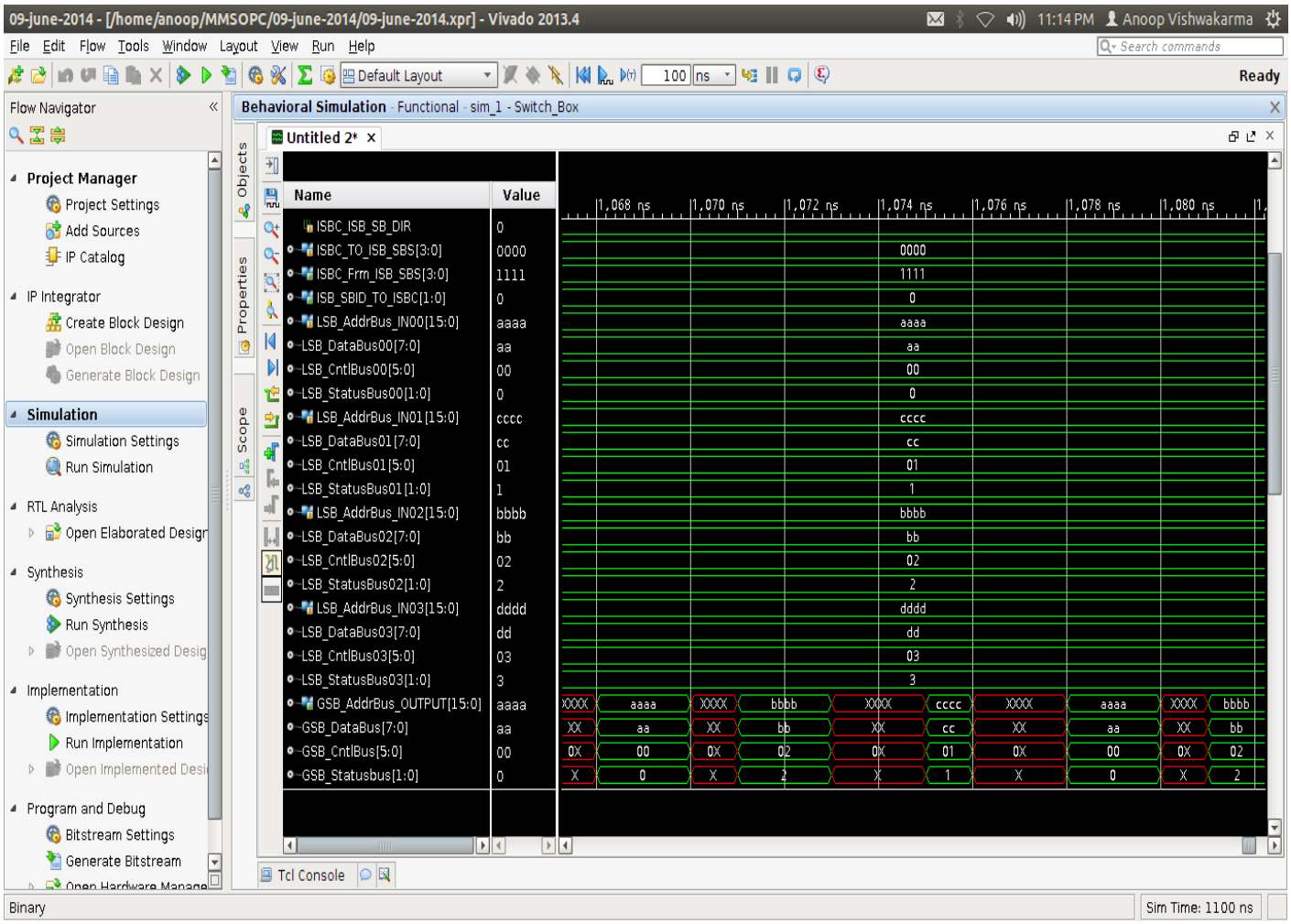
Fig -9: Utilization Graph

7.2 Utilization Table

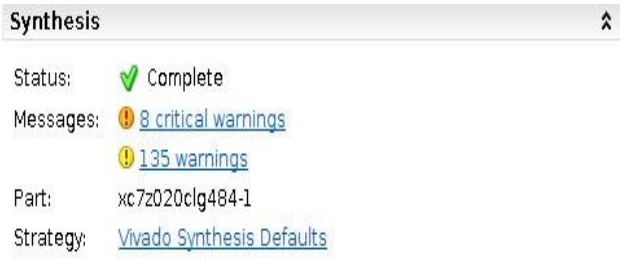
Table -1: Utilization Table

Resource	Estimation	Available	Utilization %
FF	16	106400	1
LUT	278	53200	1
I/O	118	200	59

7.3 Behavioral Simulation of Switch Box

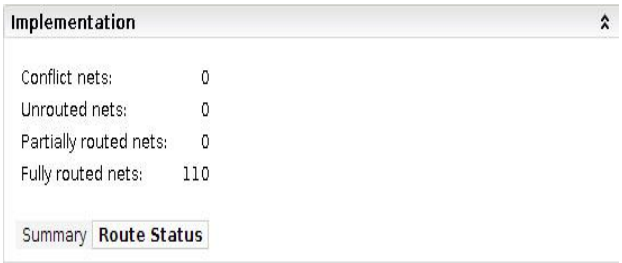


7.4 Synthesis Report

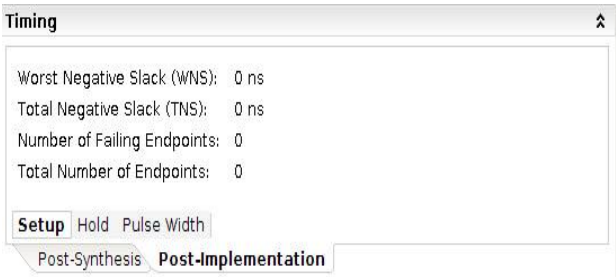


7.5 DRC Report

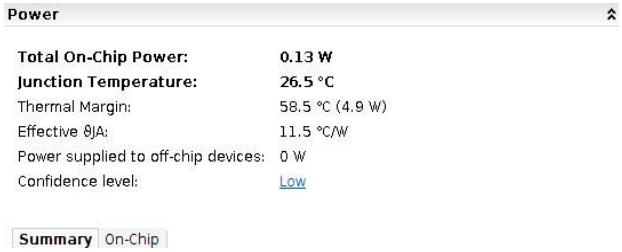
7.6 Implementation Report



7.7 Timing Report



7.8 Power Report



7.9 Power Summary (Graph)

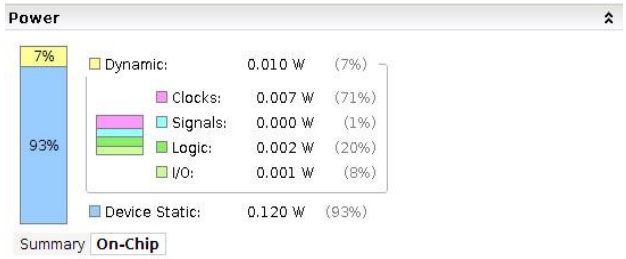


Fig -9: Power Summary

7.10 Implemented Design

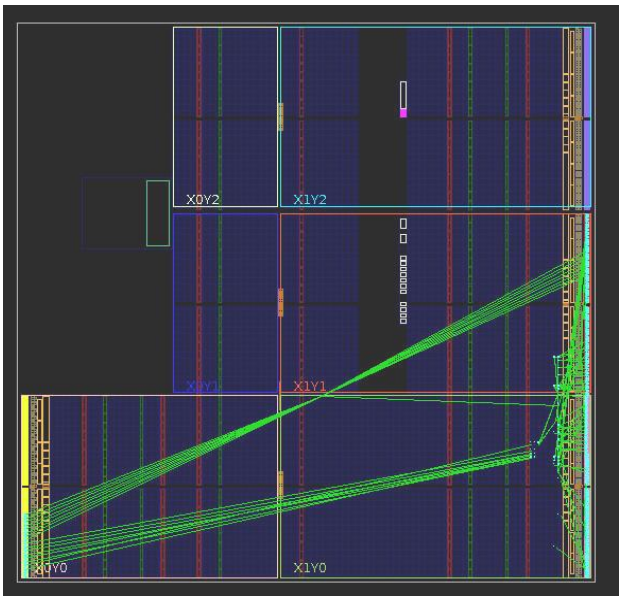


Fig -10: Implemented Design

8. CONCLUSIONS

In the above result, architecture of intelligence switch box for bus arbitration and its synchronization for multi microcontroller embedded system on programmable chip for address bus, data bus, control bus and status bus is successfully done.

Also the algorithm for synchronization of address bus, data bus, control bus and status bus for multi-microcontroller embedded system on programmable chip at a master clock frequency of 10-12 MHz for all microcontrollers and slave clock frequency of 100 MHz for tri-state buffers is proposed.

The proposed algorithm and techniques are as:

Algorithm:

Switch 00: Propagation delay(P_d)+Constant timing delay(C_{td})
Switch 01: Propagation delay(P_d)+Constant timing delay(C_{td})
Switch 10: Propagation delay(P_d)+Constant timing delay(C_{td})
Switch 11: Propagation delay(P_d)+Constant timing delay(C_{td})

Techniques:

Switch (0-3): Number of NOT Gate (P_d)+Constant Timing Delay (C_{td})
Switch 00: 0 (P_d) + 1ns (C_{td})
Switch 01: 3 (P_d) + 3ns (C_{td})
Switch 10: 8 (P_d) + 5ns (C_{td})
Switch 11: 13 (P_d) + 7ns (C_{td})

Hence, the proposed algorithm and techniques getting verified and tested for global bus synchronization using multi-clock and multi phases DLL (Delay Lock Loop).
$\Delta P_{dsw0\text{-}1}=3\Delta C_{tdsw0\text{-}1}=2ns$
$\Delta P_{dsw1\text{-}2}=5\Delta C_{tdsw1\text{-}2}=2ns$
$\Delta P_{dsw2\text{-}3=}\ 5\Delta C_{tdsw2\text{-}3}=2ns$

The prototyping & verification has been done using following tools & technology by considering various design constraints:

Software operating environment:

- Operating System: UBUNTU 12.04 LTS
- Hardware Design Software: Vivado 2013.4
- Modeling and Simulation Software: Vivado 2013.4, ISim
- Synthesis Software: Compiler-II
- Static Timing Analyzer: Prime-Time
- Power Analyzer: XPA
- Power Estimator: XPE

Hardware Prototyping Environment:

- FPGA Device: Zynq-7000 (Ultra-Fast and Ultra-Low-Power)
- Zynq-7000 FPGA Device Series: xc7z020clg484-1
- Device Technology: 28nm
- Device Category: General Purpose
- Clock Frequency: 50 Hz to 1GHz
- Speed Grade: -1
- No. of Pins: 484
- Operating Voltage: 1.16 V
- Operating Temperature: -20^0 C to 80^0 C
- Operating Clock Frequency: 10MHz and 15-20 MHz

REFERENCES

[1] Benini, L. and De Micheli, G.,2002. Networks on Chips: A new SoC paradigm. IEEE Computer, 35(1), 2002; 70-78.

[2] Terry Tao Ye 2003. On-Chip Multiprocessor Communication Network Design And Analysis PhD thesis, Stanford University.

[3] Beltrame G., Sciuto, D., Silvano. C., Paulin, P.and Bensoudane,E.,2006. An application mapping Methodology and case study for multi-processor on –chip architectures.Procceding of VLSI SoC 2006. 16-18 October 2006, Nice.France 146-151.

[4] T. S. Rajesh Kumar 2008. On-Chip Memory Architecture Exploration of Embedded System on Chip PhD thesis Supercomputer Education and Research Centre Indian Institute of Science, Bangalore, India.

[5] Xue, L., Ozturk, O., Li.F. Kandemir, M. and Kolcu. I.,2006.Dynamic partitioning of processing and memory resources in embedded MPSoC architectures .Proceeding of DATE 2006 ,6-10 March 2006, Munich,Germany, 690-695.

[6] Benini L. and De Micheli, G., 2006. Networks on Chips: Technology and tools. Morgan Kaufmann, San Francisco, 2006, ISBN-10:0-12-370521-5.

[7] L. Benini, D. Bertozzi, D. Bruni, N. Drago, F. Fummi, M. Poncino, "SystemC Cosimulation and Emulation of Multiprocessor SoC Designs", IEEE Computer, Volume: 36 Issue: 4, April 2003.

[8] F. Poletti, D. Bertozzi, A. Bogliolo, L. Benini, "Performance Analysis of Arbitration Policies for SoC Communication Architectures", Journal of Design Automation for Embedded Systems, pp.189-210, Vol. 8, June/Sep 2003.

[9] A novel on-chip communication network .Proceeding of International Symposiumon System-on-Chip, 16-18 November 2004.

[10] Advance Computer Architecture, Parallelism, Scalability, programmability by Kai Hwang.

[11] Ha, S., 2007. Model-based programming environment of embedded software for MPSoC. Proceeding of ASP-DAC'07, 23-26 January 2007, Yokohamma, Japan, 330-335.

[12] Jerraya, A. and Wolf W., 2005. Hardware-software interface codesign for embedded systems. Computer, 38(2), February 2005; 63-69.

[13] Jerraya, A., Bouchhima, A and Petrot, F., 2006. Programming models and HW/SW interfaces abstraction for Multi- Processor SoC. Proceeding of DAC 2006 .San Francisco.USA, 280-285.

[14] Angiolini, F., Ceng, J., Leupers, R., Ferrari, F., Ferri, C., and Benini, L. 2006. An integrated open framework for heterogeneous MPSoC design space exploration. In Proceedings of the Design, Automation and Test in Europe (DATE'06), 1145-1150.

[15] Erbas, C. 2007. System-level modeling and design space exploration for multiprocessor system-on-chip architectures. PhD thesis, Amsterdam University Press, Amsterdam, the Netherlands.

[16] Jia, Z. J., Pimentel, A. D., Thompson, M., Bautista, T., and Nunez, A. 2010. NASA: A generic infrastructure for system-level MPSoC design space exploration. In Proceedings of the IEEE 8th Workshop on Embedded systems for Real time Multimedia, 41-50.

[17] Jia, Z. J., Bautista, T., Nunez, A., Guerra, C., and Hernandez, M. 2008. Design space exploration and performance analysis or the modular design of CVS in a heterogeneous MPSoC. In Proceedings of the Conference on Reconfigurable Computing and FPGA (ReConFig'08). 193-198.

[18] Martin, G. 2006, Overview of the MPSoC design challenge. In Proceedings of Design Automation Conference (DAC'06).

BIOGRAPHIES

He did his B. Tech. (Electronics and Telecommunication Engineering) from UPTU Lucknow, M. Tech. (pursuing) in Electronics and Communication Engineering from UPTU Lucknow. His research interests include Embedded System Design, System-On-Chip Design.

He did his B. Tech. (Electronics and Communication Engineering), M. Tech. (Computer Science & Engineering), PGD(VLSI Design), Ph.D doing in Computer Science and Engineering. He is IEEE affiliate of Circuit and System Society and membership of IEEE organization. He is also ACM affiliate of Embedded Computing System of Association of Computing Machine. His research interests include Embedded System Design, System-On-Chip Design, VLSI Design, Advance Computer Architecture Design, RTOS Design, MPSOPC Design, FPGA Design based Embedded System.