MAT JOURNALS
Journal of VLSI Design and Signal Processing
Volume 4 Issue 3

Design and Implementation of Shared Bus based Heterogeneous MPSoC

[1]V. Rukkumani, [2]V. Dharshini, [3]R. Suvetha, [4]N. Varsha
[1] Associate Professor, [2,3,4]UG Students
Department of Electronics and Instrumentation Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India
Email:[1]*rukkumani.v@srec.ac.in,*[2]*dharshinivasudevan99@gmail.com,*
[3]*karunya2557444@gmail.com,*[4]*suvetharaj99@gmail.com*
DOI: *http://doi.org/10.5281/zenodo.1465484*

Abstract

An MPSoC architecture is proposed with shared bus interconnect and its components mainly comprising of soft IPs. The proposed MPSoC architecture has four masters and four slaves communicated over a shared bus interconnect. Each master deals with two 16-bit inputs and process among an output of 32-bit. The slaves are four independent RAM soft IPs to be designed to handle 32-bit data. The main theme is to make the four masters and four slaves to get their tasks accessed through a 32-bit shared bus interconnect. Initially the soft IPs of processors and RAM memory elements are to be designed and to be verified using Modelsim simulation software. Before developing the proposed architecture, a prototype of one master to four slaves (1:4) with a simple address decoding scheme has to be developed and simulated in Modelsim simulation software. The prototype model architecture should be synthesized under target device Altera Cyclone II using Quartus synthesizing tool. The proposed architecture of 4 masters and 4 slaves with a common shared bus interconnect should be achieved and implement the entire architecture over Altera FPGA board and verify its functionality.

Keywords: *MPSoC, RAM,Shared bus, MIPS, Altera FPGA.*

INTRODUCTION

A Multi-Processor System-on-Chip (MPSoC) is a system-on-chip with multiple processing elements. When previously developed SoCs embedded a single processor, the MPSoCs with multiple masters share the overall control. The first Multiprocessors System-on-Chip (MPSoCs) emerged mix many embedded processors, reminiscences and specialised electronic equipment (accelerators, I/Os) interconnected through an avid infrastructure to produce a whole integrated system. Contrary to SoCs, MPSoCs embody 2 or additional master methodors managing the appliance process, achieving higher performances. MPSoCs contain giant amounts of scientific discipline parts. though all of the parts of the MPSoC are scientific discipline parts that were nonheritablefrom different sources, the configuration of the digital computer and therefore the programs that run on the digital computer are created for associate distinctive system.

MPSoC are of two types based on their architecture development. A heterogeneous MPSoC is a set of interconnected cores with different functionalities. Heterogeneous MPSoC, also referred to Chip Multi Processing or Multi (Many) Core Systems. A homogeneous MPSoC is a set of interconnected cores with identical functionalities and structure. The heterogeneous design ensures the next power potency and a smaller space occupation and is additional suited to low-power multimedia system process, like in mobile devices. The homogenized theme permits for the next flexibility and easier system measurability and is additional suited to all-purpose DSP tasks in power-supplied devices. The MPSoC projected during this paper could be a heterogeneous kind.

MPSOC ARCHITECTURE

AnMPSoC architecture is achieved by the proposed system which consists of four processing elements, which are designed as soft IPs and assigned as Masters, four RAM memory elements as soft IPs and assigning them as slaves, shared bus interconnect to complete the architecture and implementation of entire MPSoC architecture in Altera FPGA.

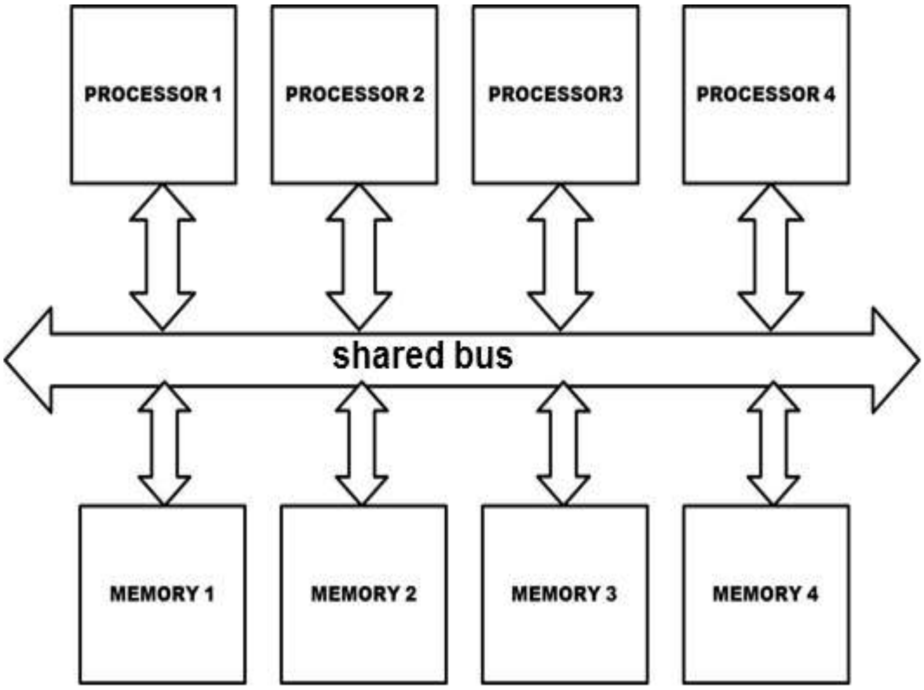

Fig: 1. *Block diagram of proposed MPSoC architecture*

The proposed scheme has a 4:4 based architecture, as shown in Fig. 1, which means four masters and four slaves to be controlled over a common shared bus interconnect. The single master processor holds two inputs of 16-bits wide each. The slaves are RAM memories of each 32-bit data and 10-bit address lines. The RAM memories are designed as soft IPs in register transfer level. Each of them holds capacity of nearly 3kilobytes of memory to get stored. There are totally four RAM soft IP memory modules assigned as slaves in the architecture. The architecture involved is now interconnected through a shared bus interface. The masters connect the slaves through the address involved to meet up each slave module. When each master involves taking control over the slave module, arbiter involves an arbitration scheme to grant the access to the master of slave. The arbitration scheme could be round robin based. The priorities of masters are statically fixed not dynamically. Hence the working mechanism over a proposed MPSoC architecture would be carried throughout the prototype and also same in the final implementations and verification environments developed.

Processor unit

The proposed processor unit consists of major modules such as input ports, ALU, control unit and a PIPO shift register. The input ports are a and b, each holding 16-bit wide binary digits. The ALU is a combination of an arithmetic and logic unit. The arithmetic and logic units are each 32-bit wide. ALU is controlled by a control unit and corresponding selection lines. The control unit is a 2:1 Multiplexer. It switches among the two 32-bit outputs, based on the selection line address given as the input to the control unit.

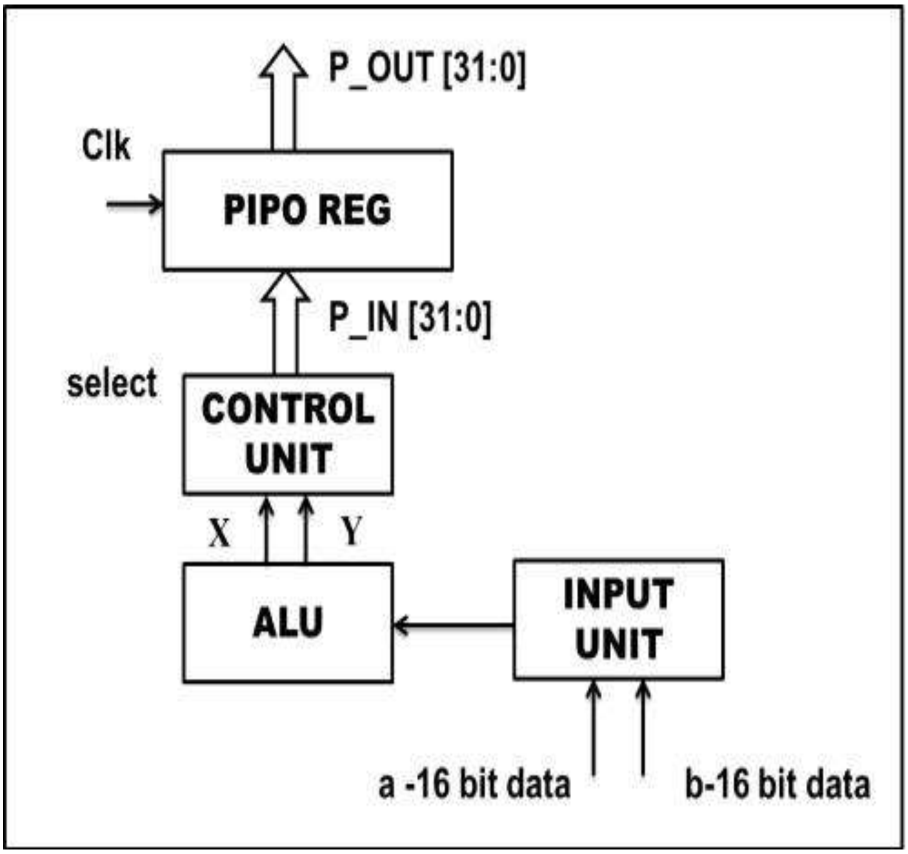

***Fig: 2.** Architectural diagram for processor unit*

The output is controlled by a PIPO Shift register of 32-bit wide, controlled by single synchronous clocking. The basic leaf cell in the PIPO will be a D-flipflop. So here, 32 D-flipflops are used to design the PIPO shift register. With a single synchronous clock, it shifts out the 32-bit data out from the PIPO register. The Fig. 2 illustrates the architectural structure of the proposed processor unit.

Memory Unit

RAM memory is used as slaves in the MPSoC architecture. The RAM Inputs are clock, enable, rd/wr, address, data-in and output is data-out. The data inputs are 32-bit wide with address of 10-bit wide and the data_out is referred to hold 32-bits. There are wide ranges of memory elements available as Intellectual Property (IP). Here we have used a program in Verilog HDL to model RAM. The Fig. 3 illustrates the structure of RAM soft IP module. For every positive clock edge the data on the input data bus has to be written to the memory, the data within the memory has to read out to the out bus depending on the control signal rd/wr. For rd/wr control signal value "0" data is written to the memory and rd/wr =1 data is read out. An enable signal is used to keep the module inactive when enable in low. This is a memory element whose width is 32-bit and whose depth is 3072 bits. So to address these memory locations you need a total of 10 address bits.

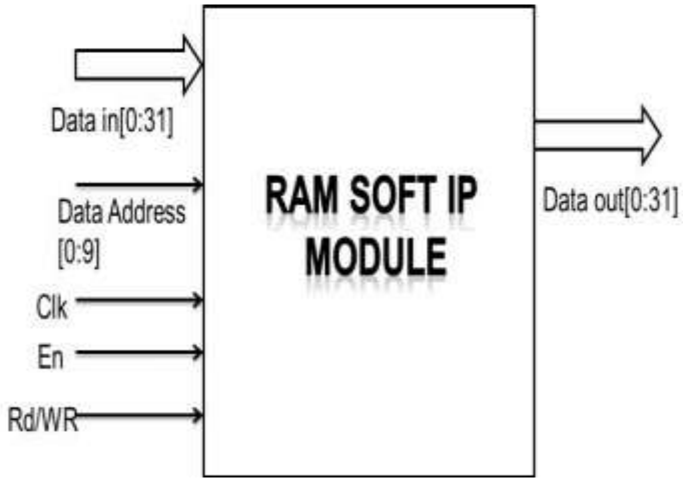

Fig: 3. *Block diagram for RAM soft IP module.*

Shared BUS

The Shared bus operation is becoming popular in many industrial applications due to it's advantages such as reduces the number of connection between the computers, cost reduction, reduced area requirements and improved reliability. The multilayer architecture acts as cross bar switch between masters and slaves. It is compatible with high speed integration microprocessors and also allows reuse of the peripherals between systems.

The bus implementation involves modelling the arbiter, control multiplexers and an address decoders. The arbiter used in this project based on the round robin algorithm. The priorities fixed type and statically allocated not dynamically. The arbiter has four inputs request signals as REQ0,REQ1,REQ2 and REQ3 and four grant signals as GNT0,GNT1,GNT2and GNT3. It also has individual clock and reset.

The control multiplexer is the second components over the system bus structure to be designed. The control multiplexer actually has three multiplexers,two 4:1 typed and one 2:1 typed component. The POUT [1:4] from the master processors are processed through the two 4:1 multiplexers and the arbiter's grant output signals are supplied as the selection line of the control multiplexers

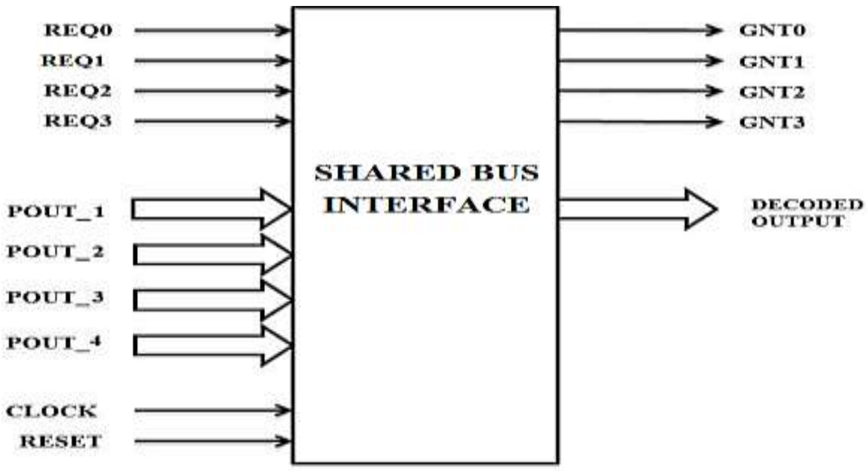

Fig: 4. *Shared Bus interface diagram*

The final component in the shared bus structure is an address decoder. A 2:4 address decoder is used further to decode the address and put the data on respective slave RAM memory. Based on the address decoding scheme the slaves are selected. The 2:4 decoding scheme comprises of the four selection addresses to select the RAM memories.

PROTOTYPE DEVELOPED

The MPSoC prototype developed is illustrated in the Fig. 5, with its internal configurations. Each processing element has its own request signal to access the bus resource. The requests are driven through the arbiter of shared bus.

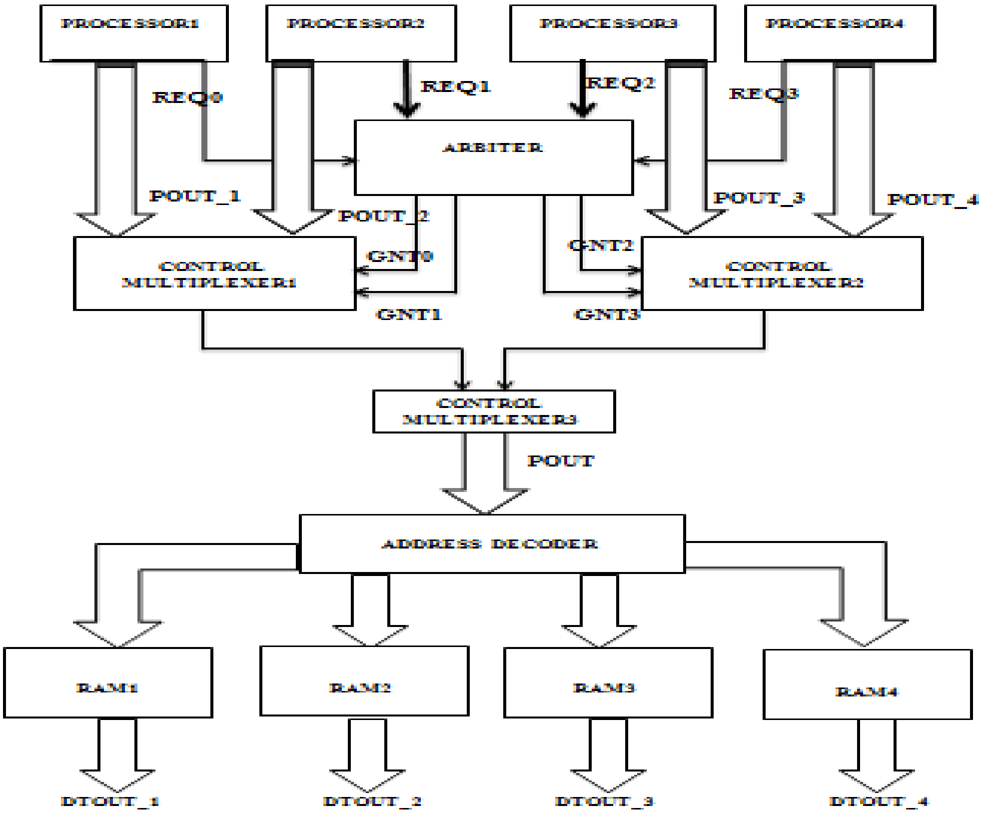

Fig: 5. *Prototype for MPSoC architecture*

The REQ0, REQ1, REQ2 and REQ3 are the request signals, the priority is statically fixed. The initial processor1 has the highest priority considered to the following processors; hence processor4 has its least priority. Three control multiplexers and an address decoder is used to complete the bus structure.

The grant signals are correspondingly connected across two control multiplexers. From two control multiplexers the switching of data takes place. A flag is set, when GNT0==1 or GNT1==1, and the flag is cleared when GNT2==1 or GNT3==1, so as to access the final control multiplexer. The address decoder drives the data_out and according to the corresponding address selection the processor output is saved in the RAM memory slave unit.

The Table.1 shows the GNT signals and corresponding processor getting access over the slaves memory elements.

***Table: 1.** Grant access to the slave*

GNT0	GNT1	ACCESS
1	0	PROCESSOR0
0	1	PROCESSOR1

When GNT0 is high, processor1 gets control to access any one of the slave memory unit, when GNT1 goes high processor2 gets control to access the memory unit.

SIMULATION RESULTS

The MPSoC prototype is designed using Verilog HDL and it is a RTL based design. The entire architecture is called by its instances to make use of reuse of IP techniques. The design is verified by creating suitable test benches and simulated using Modelsim EDA tool. The input for each processor is forced simultaneously and slaves are activated so as to get accessed by master elements as per arbiter granting service.

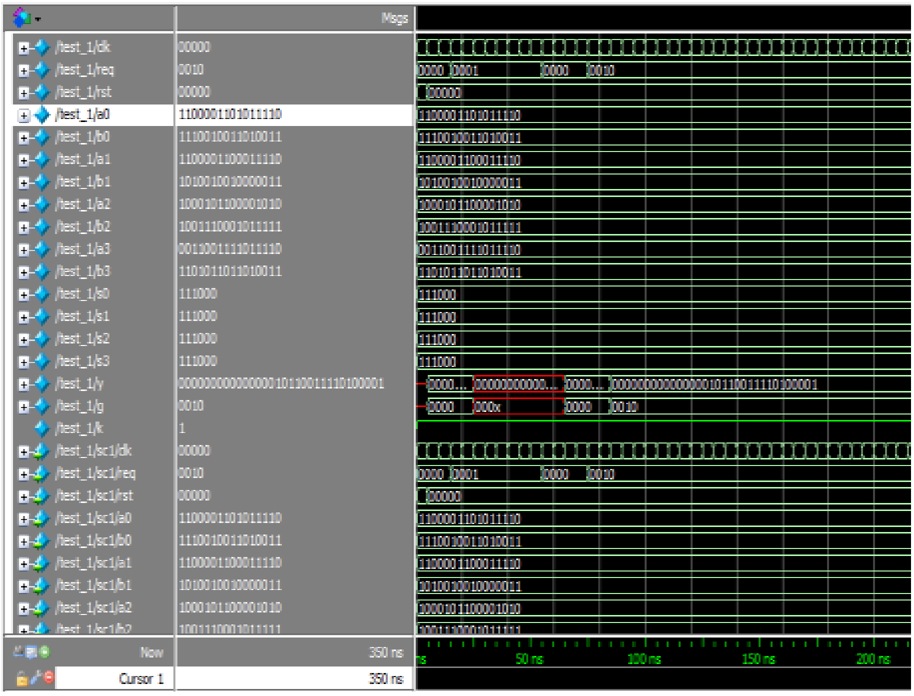

***Fig: 6.** Simulation Result from developed Test bench for processor*

The individual clocks are given to the slave modules, so as to keep them activated. The required request signals through the processors are fed as inputs and the corresponding outputs are carried out. The address decoder switches the input to RAM memory and gives out the output with required address lines given to the decoder. The processor's output stays over the requires RAM slave memory module. The Fig. 6 shows the simulation results obtained from Modelsim software tool by developing corresponding test bench to verify the RTL based design of MPSoC prototype.

CONCLUSION

A heterogeneousMPSoC prototype design was presented with its elements of processors, shared bus and RAM memory modules. The entire design was done in Verilog HDL and simulated using Modelsim EDA tool. The FPGA synthesis results are obtained from Quartus tool. The test bench developed here are Verilog HDL based which obtains the simulation results. The future enhancements includes designing prescribed verification environments by using some Hardware Verification Language (HVLs) such as System Verilog to make the verification a more understandable task and also reduction in time required for verification of the design.

REFERENCES

1. Chin-Yao Chang and Kuen-Jong Lee, “On Deadlock Problem of On-chip buses Supporting Out-of-Order Transactions” in IEEE transactions on Very Large Scale Integration (VLSI) systems, vol. 22, no. 3, March 2014.
2. K. Lahiri, A. Raghunathan, and G. Lakshminarayana, “The LOTTERYBUS on-chip communication architecture,” in IEEE Transaction on Very Large Scale Integration (VLSI) Syst., vol. 14, no. 6, pp. 596–608, Jun. 2006.
3. W. Wolf, A. Jerraya, and G. Martin, “Multiprocessor System-on-Chip (MPSoC) Technology,” in IEEE Transactions on Computer Aided Design Integrated Circuits Syst., vol. 27, no. 10, Oct. 2008.
4. W. Zhang et al., “Design of a hierarchy-bus based MPSoC on FPGA,” in International conference on Solid-State and Integrated Circuit Technology, pp. 1966-1968, March 2006.
5. H.W. Wang, C.S. Lai, C.F. Wu, S.A. Hwang, and Y.H. Lin, “On-Chip Interconnection Design and SOC Integration with OCP,” in IEEE International Symposium VLSI Design, Automation Test, April 2008, pp. 25–28.
6. O. Ogawa, S. Bayon de Noyer, P. Chauvet, K. Shinohara, Y. Watanabe, H. Niizuma, T. Sasaki, and Y. Takai, “A practical approach for bus architecture optimization at transaction level,” in Design, Automation Test, Europe Conference. Exhibit., 2003, pp. 176–181.
7. T. S. Cummins, “Method and apparatus for detecting a bus deadlock in an electronic system,” U.S. Patent 6 292 910, Sep. 18, 2001.
8. H. Park, “Easily Adaptable On-Chip Debug Architecture for Multi-core Processor,” in IEEE Transaction on Very Large Scale Integration, vol. 34, no. 1, Feb. 2013, pp. 44-54.
9. Wayne Wolf “Modern VLSI Design: IP-Based Design”, Fourth Edition
10. Samir Palnitkar “Verilog HDL: A guide to digital design and synthesis”, second edition.

Cite this article as: V. Rukkumani, V. Dharshini, R. Suvetha, & N. Varsha. (2018). Design and Implementation of Shared Bus based Heterogeneous MPSoC. Journal of VLSI Design and Signal Processing, 4(3), 37–43. http://doi.org/10.5281/zenodo.1465484