Indonesian Journal of Electrical Engineering and Informatics (IJEEI)
Vol. 11, No. 4, December 2023, pp. 1096~1109
ISSN: 2089-3272, DOI: 10.52549/ijeei.v11i4.5032

Efficient Dual Mode Arbitration Scheme for Multiprocessor Hardware Interface in System-on-Chip

S. M. A. Motakabber[1], M. I. R. Rokon[2], A. H. M. Zahirul Alam[3], M. A. Matin[4], Md Mahmud[5]
[1,2,3]Department of Electrical and Computer Engineering, International Islamic University Malaysia, Kuala Lumpur, Malaysia
[2,4]Department of Electrical and Computer Engineering, North South University, Dhaka, Bangladesh
[5]Phillip M. Drayer Department of Electrical Engineering, College of Engineering, Lamar University, Beaumont-77710, Texas, USA

Article Info

Article history:

Received Aug 18, 2023
Revised Dec 1, 2023
Accepted Dec 22, 2023

Keyword:

SoC
Multiprocessor
Arbitration
AMBA Bus Protocol
Processor Interface
Verilog
Simulation
Synthesis
FPGA

ABSTRACT

A processor transforms human needs into hardware operations in any SoC. The single processor was ubiquitous in previous systems. But as chip size, complexity, and speed increase, several processors are used nowadays to handle concurrent operations. To manage requests from several processors, a central hardware block will conduct the arbitration among the processors and allow a processor to access the bus. This paper addresses the multiprocessor arbitration in any System on Chip or ASIC. There are several arbitration algorithms available in the realm of technology, and any system can choose a specific arbitration to implement in hardware based on its own demands. Instead of using one type of arbitration in hardware, this research combined and used two schemes and implemented both possibilities in a hardware dual-mode arbiter system to be used in SoC. The proposed dual-mode arbiter was initially hardware modeled using Verilog HDL, then functionalities were verified using industry simulator Cadence and Modelsim, and finally synthesized and implemented using Xilinx XST EDA tool and FPGA device. The AMBA, the industry-standard bus protocol, is being considered for the master processors and the proposed dual-mode arbiter to ensure an efficient hardware interface and to use with any off-the-shelf macro available for the high-tech industry.

Corresponding Author:

Mohammed Iqbalur Rahman Rokon
Department of Electrical and Computer Engineering,
International Islamic University Malaysia, Gombak, 53100, Kuala Lumpur, Malaysia
Email: Iqbal.rokon@gmail.com

1. INTRODUCTION

Previously in industry single processor, be it internal and external, was used with ASIC and processor had full access to the decanted data, address, control and status signals. So anytime single processor needed access to any targets, it got the control of the bus right away with the grant signal [1]. As the size and complexity of various ASICs and SoCs system have grown to very high to coop with the increased number of hardware operations. So, industry felt the need to use several processors and thus multiprocessors system emerged [2]. In case of multiprocessor hardware system, a central hardware system is needed to allow each processor in the system to get control of bus and access the targets. When there are several bus masters trying to access to the bus by sending request to read or write data from targeted hardware addresses in ASIC, processor interface needs an arbitration scheme to decide which processor will be allowed and get grant signal to get hold of the bus to access the targeted area[3]. There are several bus arbitration schemes are used in industry to do the arbitration: Fixed Priority, Round Robin and Daisey Chain, etc. Each of this arbitration algorithm has strengths and weaknesses, and the system uses specific arbitration schemes based on its industry hardware needs. In

fixed priority arbitration is used when master processors have to be given level of priority to access the hardware. When higher priority processor issues any request, fixed priority arbitration allows any ongoing read or write operation by previously granted processor to finish its operation then issues grant signal to the higher priority [4]. The weakness of fixed priority is if higher priority processors keeps send requests, it'll always get priority to have grant signal, thus lower priority processor may not get the grant for long time. To address this situation, round robin arbitration scheme considered. In this scheme all requesting processors are given same priority level to ensure fairness in bus access. This algorithm is used in industry to ensure that no processor goes starving and all get fair access to the bus based on access request. In this research, a priority counter has been incorporated to keeps track of bus access by the processors and anytime any processors is allowed to access to the bus, its priority level goes to bottom and next requesting processors priority level goes up so that it gets its access when its turn arrives[5]. As per current ASIC in industry, once any arbitration is implemented in hardware, be it fixed priority or round robin, the system has to stick to that hardware even though system might need other priority algorithm later in hardware to keep up with the system's changing needs in the industry. This limits if hardware needs are changed: either from fixed priority to round robin or round robin to fixed priority. In that case whole ASIC hardware needs to be redesigned, verified and implemented. This research addresses this limitation of having certain arbitration scheme and gives a solution in hardware implementation that offers dual-mode arbitration with both fixed priority and round robin arbitration. This dual mode arbitration allows the processor interface to switch from one hardware arbitration scheme to another and it can be controlled by setting control register. This gives seamless transition to go for either arbitration scheme without going through whole hardware redesign and implementation.

2. MULTI PROCESSORS ARBITRATION SCHEME AND SIGNALS IN AMBA BUS PROTOCOL

Processors use bus interface protocol while they access any targets. There are standard protocols like Strong ARM and AMBA. The AMBA AHB is for high-performance, high-clock frequency system modules. The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories, and off-chip external memory interfaces with low-power peripheral microcell functions [1].

DMA controller, on-chip memory to access slave targets like register, RAM, FIFO, UART, etc. The previously single processor was typical to access the hardware. As technology advanced and so did the complexity and chip size, using several processors and concurrent access to targets are common nowadays to keep up with the operational needs. An interface for multiprocessor access requires an arbitration scheme to allow any selected bus master processor to receive the grant access and get control over the bus to perform its operation. The arbitration controller hardware resolves the bus master access based on the arbitration mode – fixed priority or round robin. The block diagram of the arbitration hardware is shown in Figure 1.

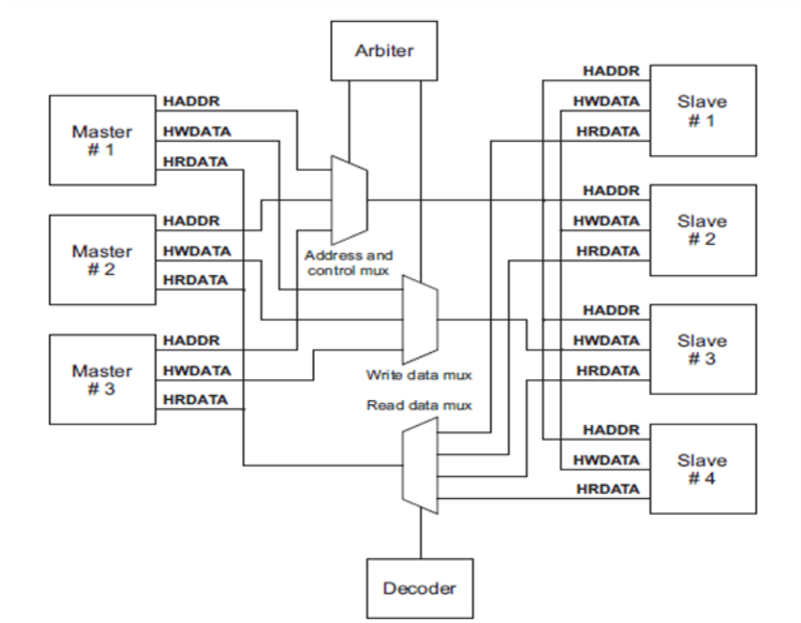


Figure 1. AMBA Multiprocessor Interconnection in Arbitration

In the dual mode arbitration controller, only one processor out of multiple bus master processors can get hold of the bus and perform access operation to the targets, be it data read or write, using all the control and status signals. When multiple processors assert a bus access request, the arbitration scheme first resolves which mode of arbitration the controller will go for – fixed priority or round robin and on the mode of arbitration, the state machine performs

2.1. HCLK (Host us clock)

AHB system clock HCLK is the high-speed system clock that controls the sequential nature of all the requesting bus processors' access. It goes to all the flip flops in the address path, data path, control signals path, and status signal path and provides sequential access and speed of the signal transfer. The external system clock generator generates the system clock to drive all the signals in the system. AMBA AHB bus protocol requires signal timings to be positive edge-triggered by HCLK.

2.2. HRESETn (Bus Reset Control Signal)

The system reset signal HRESETn is issued by the reset controller of AHB masters. Its active-low resets all flip-flops of the system when its value is 0 and puts all flips of hardware in a known value of 0 in this research. A low value of 0 at the HRESETn signal resets all sequential blocks in the system, thus putting all state machines in a known state too. Synchronous, i.e., dependent on the clock or asynchronous or independent from the clock reset, can be used based on system specification; in this research work, synchronous rest is used.

2.3. HBUSREQx (Bus Request Signal)

1-bit bus request signal. Bus masters use this signal to tell the processor it needs access to the targeted area. Usually, its value is 0. However, when any AMBA bus master processors need the bus to use and access any area of SoC, it asserts the request signal by setting the HBUSREQx value to 1. Central arbiter records that request signals understand that the processor needs access to the bus and issues a grant signal to the processor to access the bus.

2.4. HTRANSx[1:0] (Bus Trusfer Type)

A 2-bit transfer type signal; This signal indicates the transfer type by the master processor. A Each processor x issues a 2-bit signal HTRANSx to indicate its current transfer type. The 2 bits of this signal indicate 4 types of bus access: 00 IDLE, 01 for BUSY, 10 for NON-SEQUENTIAL. 11 for SEQUENTIAL.

2.5. HGRANTx (Bus Grant)

A 1-bit bus grant signal; This HGRANTx signal is specific for every master processor, where x indicates processor number. This signal indicates that a particular bus master is granted for its request and can now get on the AHB bus to the data transaction. An arbitration system accepts the requests from the processors and, based on the arbitration scheme, decides which processor is supposed to get permission to access and issues a grant signal to that processor.

It may take AMBA LOCK and SPLIT transfers into account too. For example, a locked transfer allows a master to complete a transfer without allowing other masters to access the bus. Similarly, a SPLIT response issued by a SPLIT-capable APB device prevents any master new bus access unless it signals to the arbiter that it is ready to complete the transfer [1].

2.6. HMASTERx[3:0] (Master Number)

A 4-bit master number signal. This HMASTERx signal, where x denotes the master number, is generated by the central arbitration hardware. Any particular value of this signal indicates which master has been granted access to the bus and is currently performing a data transfer.

3. AHB BUS PROTOCOL IN PROCESSOR ARBITRATION

When one of several processors issues any request signal, the address decoder uses the processor address to create a select target signal for a specific processor to carry out access operation [3][4].

3.1. Granting Bus Access

When they are several bus master processors, and all try to get a hold of the bus, the arbiter block resolves which processor would get bus access by using its arbitration scheme controller, generates a grant signal for that processor, and assigns the bus master number to HMASTER[3:0]. The timing diagram in Fig 2 shows the activity of the signals, and it explains how the arbiter issues GRANT signal to the processors and

allows the bus master to perform read or write operations based on the value of HTRANS as described in Figure 2.

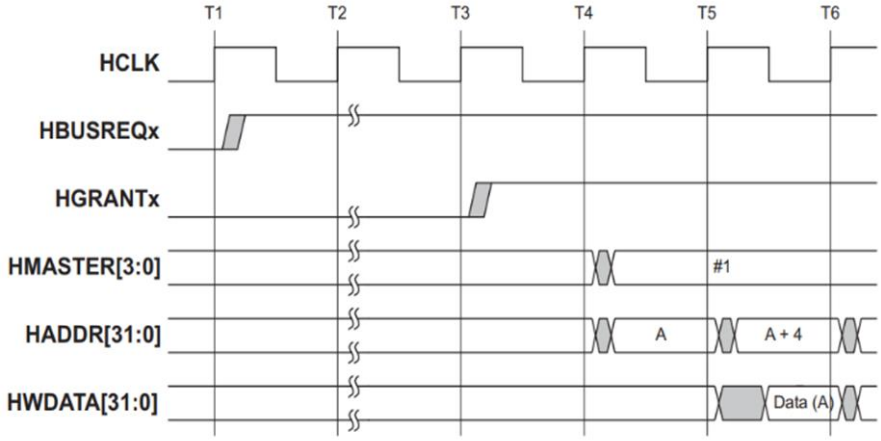


Figure 2. AMBA BUS Request and Grant Protocol [1][3]

3.2. Handover Bus Between Processors:

In a multiprocessor system, arbitration needs to be carried out by the arbiter to decide which processor is supposed to get access and issue an HGRANT signal for that processor before the last address is sampled and generates the GRANT signal for the next requesting processor according to priority as it sends a request signal, being served and designed arbitration scheme, which is shown in Figure 3.

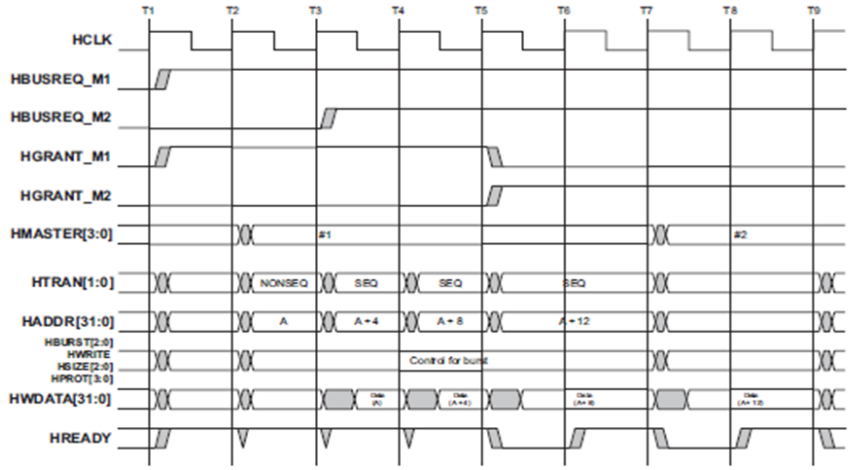


Figure 3. Bus arbitration and ownership for AMBA Multiprocessor access [5]

3.3. Bus Master Grant Signal

After getting bus access requests from several processors, the arbiter goes for arbitration scheme as per arbitration mode, decides which processor is supposed to get the access based on priority mode and level, and asserts HGRANTx to that processor, which allows the corresponding processor to gain control of the bus and initiate operation using its address, read/write data and other control signals.

4. MULTIPROCESSORS ARBITRATION SCHEME

Several types of arbitrations schemes can be used to implement an arbitration system: fixed priority, daisy chain, or round-robin. All have their strengths and weakness. All system targets one type of arbitration, and hardware implements that in a chip. But this research considers two types of arbitration - fixed priority and round-robin to hardware implementation in a chip. It gives the system a better and wider option to address the arbitration for its bus master processors. It is implanted using an efficient algorithm for efficient arbitration and ensures optimized grant signal generation logic to allow faster hardware access to the processors [5][6].

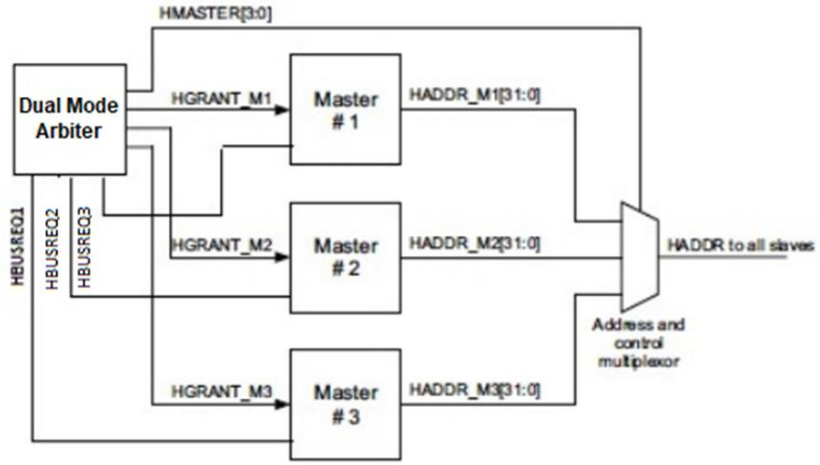


Figure 4. AMBA bus arbitration scheme

In AMBA bus master operation, processors use HBUSREQ1, HBUSREQ2, and HBUSREQ3. Then the system goes for either fixed priority or round robin arbitration based on hardware needs, synchronize request signal = s with system clock HCK Arbitration controller receives the bus requests after those are sampled then goes for arbitration scheme to resolve the request and issue grant signal. After that, grants signals from the state machines are sampled by the system clock again to avoid setup and hold time violations before sending it to the allowed bus master processor to get control of the bus.
For using a grant signal, the arbiter allows the permitted processor to access the bus as long as the processor issues one NONSEQ cycle and several SEQ cycles. Arbiter also sets the value of 4 bits HMASTER to indicate which master has been granted access to the bus and is currently performing a data transfer so that its address, data, and control signals can be forwarded to the targeted access address area.

5. HARDWARE of DUAL MODE ARBITER

As mentioned, the proposed Dual Mode Arbiter in this research work addresses two types of arbitration schemes and proposed hardware implementation of a special arbiter which, after getting a request from any processor, can resolve the arbitration based on fixed priority arbitration and round robin arbitration. The top block of this Dual Mode Arbiter is shown in Figure 5. This arbiter receives bus request signals from all the bus master processors and, as per the value or arbiter mode signal, decides the priority of the arbitration scheme and uses one of its two significant blocks inside it. It then issues the grant signal accordingly and keeps asserting the grant as per transfer types to allow specifically granted processors so it can get control of the AMBA bus to send or receive data by that processor [10][11]

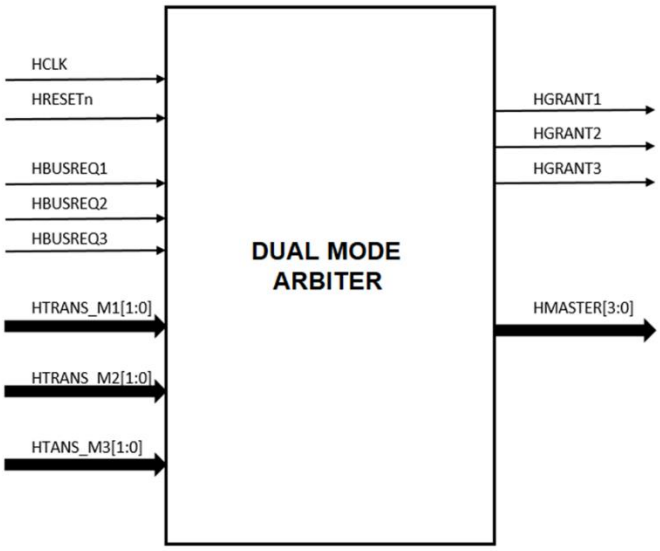


Figure 5. Top Level Block of AMBA DUAL MODE BUS Arbiter

The three bus master processors, M1, M2, and M3, are used in the arbitration scheme. The two-state machines control the access to those processors as per a dual mode scheme to either select fixed priority or round robin to avoid bus starvation. Two of the most significant signals associated with the state machine are the arbitration mode select, bus request, and the type of transfer signals generated by the processors. These signals cause all the state machine transitions, be it fixed priority or round robin, allowing three processors to access the target areas to perform read or write using an address, data path, and all the control and status signals according to the designed AMBA bus protocols [10][11].

Whatever the arbiter is – fixed or round robin, it considers the currents state, mode of arbitration, processors' request signals, and AHB transfer type if it's IDLE, BUSY, NONSEQ, and SEQ types transfer, and decides which processor is to be served under a particular mode of arbitration and issues grant signal to one of requesting processors to perform its operation, be it hardware read or write data from desired targets[12]. The hierarchical block diagram of the dual-mode arbiter is shown below. The top block of the dual mode arbiter consists of three major sub-blocks: fixed mode arbiter, Round Robin arbiter, and grant controller, as shown in Figure 6.

The firmware sets the value of a bit in a register that determines the value of the hardware signal named arbiter mode, where its value 0 is for fixed priority. The state machine under 0 value of arbiter mode signal determines the operations of dual-mode arbitration hardware under fixed priority hardware as per the state diagram shown in Figure 7. The controllers have three states, M1_FP, M2_FP, and M3_FP, to cover operations of three processors M1, M2, and M3 under fixed priority mode. Simultaneously the state machine under value 1 of the arbiter mode signal determines the operations of Dual Mode Arbitration Hardware under round robin arbitration hardware as per the state diagram shown in Figure 8. The controller has three states M1_RR, M2_RR, and M3_RR to cover operations of three processors M1, M2 and M3 under round robin mode of the entire arbitration of the hardware.

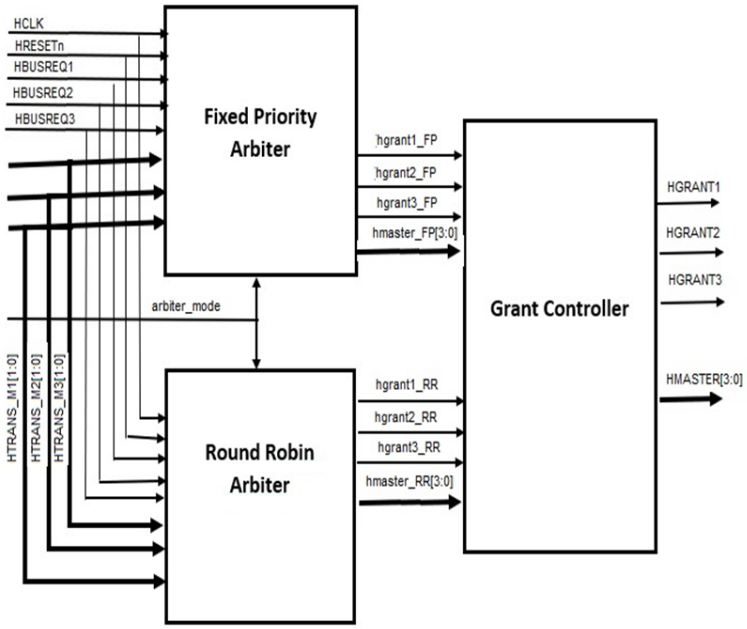


Figure 6. Block Diagram of Dual Arbitration Scheme

6. DUAL MODE ARBITER BLOCK DESCRIPTION

Description of three major blocks in Dual Mode Arbiter with input-output signals and controlling arbitration schemes are as follows:

6.1. Fixed Mode Arbiter [7][8]

Hardware module Fixed Mode Arbiter implements the fixed priority arbitration scheme in dual mode arbiter. The state machine that controllers this scheme invokes when any of three bus master processors sends a bus request by asserting BUSRER1, BUSREQ2 or BUSREQ3 and the arbiter mode signal are low or 0. The state machine that controllers this arbitration go thru following four states as per the priority level given to the processors:

State Machine

Fixed Mode Arbiter uses a state machine to control processors' accesses as per fixed priority given to each processor in a design specification. First, it receives the HBUSREQx issued by the processors. Then, based on the priority and transfer types of a particular processor, M1, M2, or M3, it keeps granting the HGRANTx signals to the allowed processor for its read-write access to the targeted device. Then, it sets the value of HMSTER, which is used to figure out which address, HREAD, HWRITE, and control signal will get control of the bus [7].

Under research, the state machine in fixed mode arbiter is the main controller for fixed mode arbitration in dual mode arbiter. Part of the dual mode arbitration scheme is designed for the multiprocessor hardware interface. For example, suppose there is a request from any processor, and the value of the arbitration mode selection signal is zero. In that case, hardware enters to fixed mode arbitration scheme, performs the arbitration accordingly, and issues a grant signal to allow processor hardware access and length of access. States, encoded value, and descriptions are as follows:

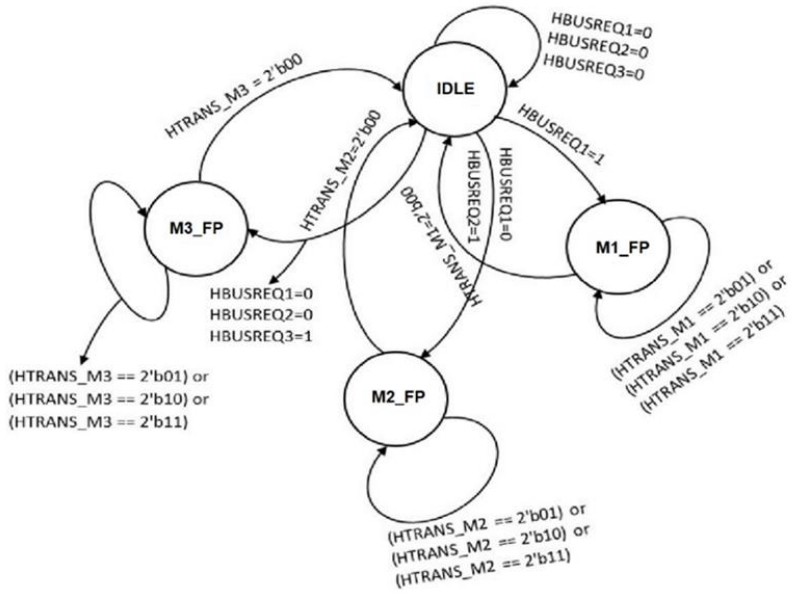


Figure 7. State transitions of Dual Mode Arbitration under Fixed Priority [7][8]

IDLE (2'b00)

If there is no request from any processor M1, M2, or M3, the state machine stays in an IDLE state without going thru any arbitration and issuing a grant signal to any processor.

M1_FP (2'b01)

Processor M1 is given the highest priority in fixed priority mode, and as soon as it issues a request by asserting the BUSREQ1 signal, it gets the bus control right away by receiving the HGRANT1 signal as high or 1. The state machine operates in this state as long as the master exchanges data with targets by issuing first NONSEQ and then several SEQ transfers by controlling the value of the HTRANS_M1 signal from processor M1.

M2_FP (2'b10)

Processor M2 is given second priority level after M1 and M2. So as soon as processor M2 issues a bus access request by asserting BUSREQ2 and for sure there is no bus request from processor M1, the state indicates to this state and asserts HGRANT2. It keeps the grant signal high as long as processor M2 issues one NONSEQ and several SEQ transfers using the HTRANS_M2 signal.

M3_FP (2'b11)

Processor M3 is given the third priority level after M1 and M2. So as soon as processor M3 issues a bus access request by asserting BUSREQ3 and there is no request from either processor M1 or M2, the state machine goes to M3_FP state and asserts grant signal HGRANT2. It keeps the grant high as long as processor M3 issues one NONSEQ and then several SEQ transfer cycles using the HTRANS_M3 signal.

6.2. Round Robin Arbiter [9]

Hardware module Round Robin Arbiter implements round robin arbitration scheme in dual mode arbiter. The state machine that controls this scheme invokes when any of three bus master processors send a

bus request by asserting BUSRER1, BUSREQ2, or BUSREQ3 and the arbiter mode signal is high or 1. The state machine that controls Round-Robin arbitration goes through the following four states to ensure fairness in bus access among all three processors, M1, M2, and M2, and ensure no processor stops from accessing the bus. State names, encoding and

State Machine

Round Robin Arbiter uses a state machine to control processors' accesses as per the round-robin arbitration scheme. First, it receives the HBUSREQx issued by the processors. Then, based on the priority and transfer types of a particular processor, M1, M2, or M3, it keeps granting the HGRANTx signals to the allowed processor for its read-write access to the targeted device. Finally, it sets the value of HMSTER, which is used to figure out which address, HREAD, HWRITE, and control signal will get control of the bus [7][8].

Under research, the state machine in fixed mode arbiter is the main controller for fixed mode arbitration in dual mode arbiter. The part of the dual mode arbitration scheme is designed for the multiprocessor hardware interface. For example, suppose there is a request from any processor, and the value of the arbitration mode selection signal is zero. In that case, hardware enters to fixed mode arbitration scheme, performs the arbitration accordingly, and issues a grant signal to allow processor hardware access and length of access. States, encoded value and descriptions are as follows [9].

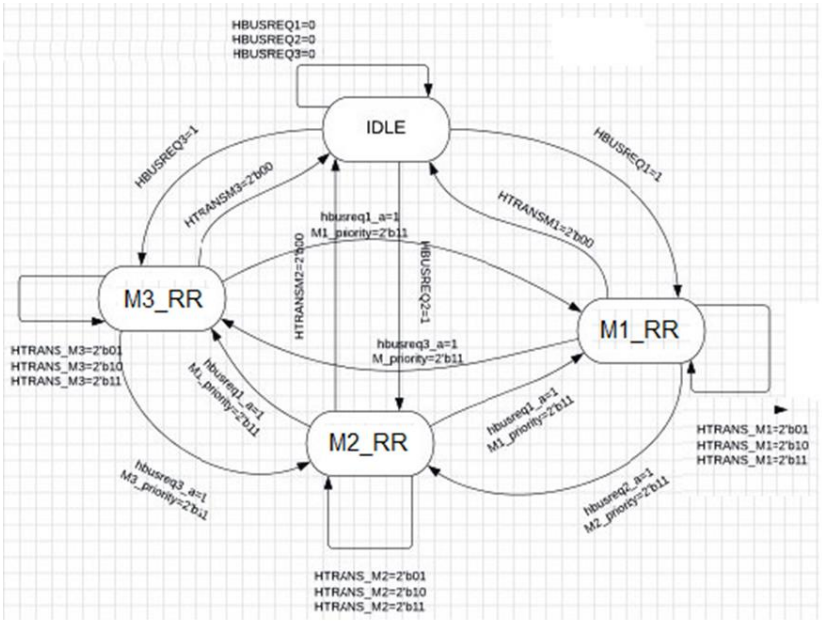


Figure 8. State transitions of Dual Mode Arbitration under Round Robin [9]

This state machine has three 2-bit priority counters for each processor. It keeps track of when each processor is served its request and adjusts the priority level based on the counter instead of fixed priority arbitration, the value not fixed.

IDLE (2'b00)

If there is no request from any bus master processor M1, M2 or M3, the state machine stays in the IDLE state without going thru any arbitration and issuing a grant signal to any processor.

M1_RR (2'b01)

Processor M1 issues the BUSREQ1 signal, and the state machine goes to this state. If the value of its priority counter is 11, it gets grants grant signal HGRANT1 high to access the bus. State machines stay in this state as long as the master processor transfers data by issuing one NONSEQ and several SEQ transfers using the HTRANS_M1 signal. Otherwise, it checks the processor request and its priority counter value and goes to the states to serve the next processor with the request and priority counter value of 11.

M2_RR (2'b10)

When processor M2 issues the BUSREQ1 signal, the state machine goes to this state. If the value of requesting processor's priority counter is 11, it sets grant signal HGRANT2 high to allow processor access to the bus. State machines stay in this state as long as processor 2 transfers data by issuing one NONSEQ and several SEQ transfers using the HTRANS_M2 signal. Otherwise, it checks other processors' requests and priority counter values and goes to the states to serve the next processor with a request and priority counter value of 11.

M3_RR (2'b11)

Processor M3 issues the BUSREQ3 signal, and the state machine goes to this state. If the value of its priority counter is 11, it sets the grant signal HGRANT3 high to allow processor 3 access to the bus. State machines stay in this state as long as processor 3 transfers data by issuing one NONSEQ and several SEQ transfers using the HTRANS_M3 signal. Otherwise, it checks other processors' requests and priority counter value to go to the required states to serve the next processor with a request, and the priority counter value is 11.

7. SIMULATION AND DISCUSSION

7.1. Simulation Algorithm

The simulation algorithm of this research work is to determine appropriate master processor gets the grant signal according to arbitration scheme in dual mode arbiter and is allowed to access to the target device area, be it register, FIFO, UART, PCI bus, or Fiber Channel Network link port. The AMBA Bus Functional Model (BFM) M1, M2, and M3 are used to carry out the simulation of this research work's proposed Dual Mode Arbiter. All three processors send bus access request both simultaneously or one in all possible sequences and orders. After receiving one or multiple requests from one processor or multiple processors, hardware modeled in this research work hardware will trigger its arbitration scheme controller and based on the arbitration mode set in the control register, fixed or round robin, Dual Mode Arbiter is to decide which processor is to get the grant signal to get on the AMBA BUS and access target device. Hardware must also assign the vale value of the bus master signal so that address and control signals of the granted signal are used to access the targets. All participation hardware signals from processors and the proposed dual-mode arbiter would be displayed in text form generated by the simulator or as a timing diagram in waveform viewer. The complete simulation will be carried out for several cycles of processor requests from all three processors concurrently in the research work's possible sequences, orders, and arbitration schemes.

The simulation algorithm planned to simulate the hardware under research has been implemented in Test Bench and shown in the block diagram as follows:

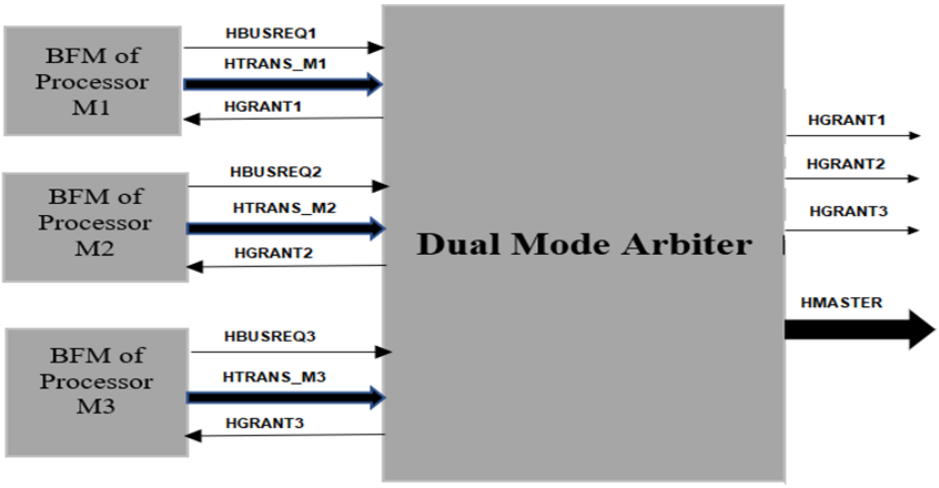


Figure 9. Simulation Test Bench Environment

7.2. Functional Simulation Result

With all this research's ideas, findings, and features, the proposed dual-mode arbiter has been hardware-modeled using Verilog HDL. Then testbench was created to verify the HDL model of the dual-mode arbiter. Then simulation is carried out using both Modelsim and Cadence simulators to ensure the arbiter's correctness and justify the hardware functions of the research work. Figure 8 and Figure 9 show the simulation result of dual-mode arbiter under both arbitration schemes: fixed mode with arbiter mode signal value 0 and round robin with arbiter mode signal value 1. Furthermore, the simulation shows all the bus masters tried to access the bus by the corresponding bus request signal created in the test bench, and grants were provided by the dual mode arbiter, considering both arbitration schemes in Figure 10-11a.

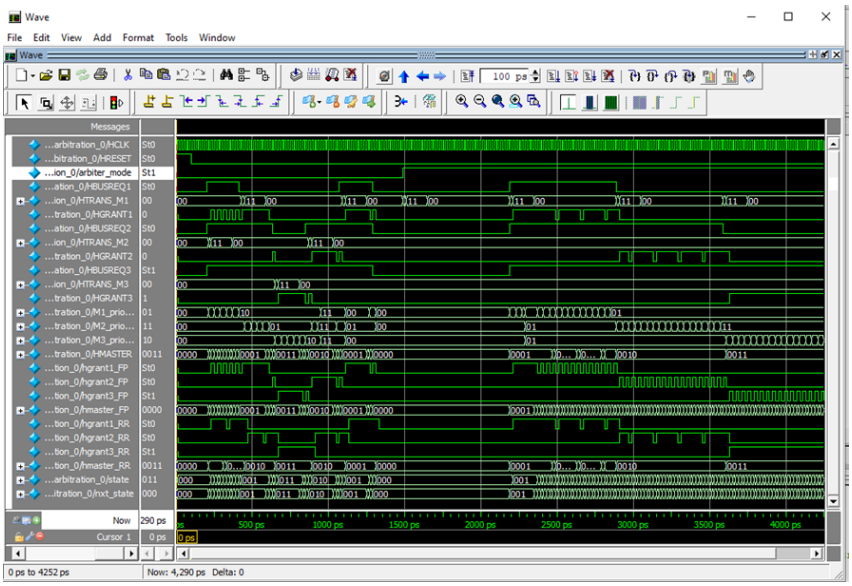


Figure 10. Full Simulation of Dual Mode Arbitration participating 3 processors

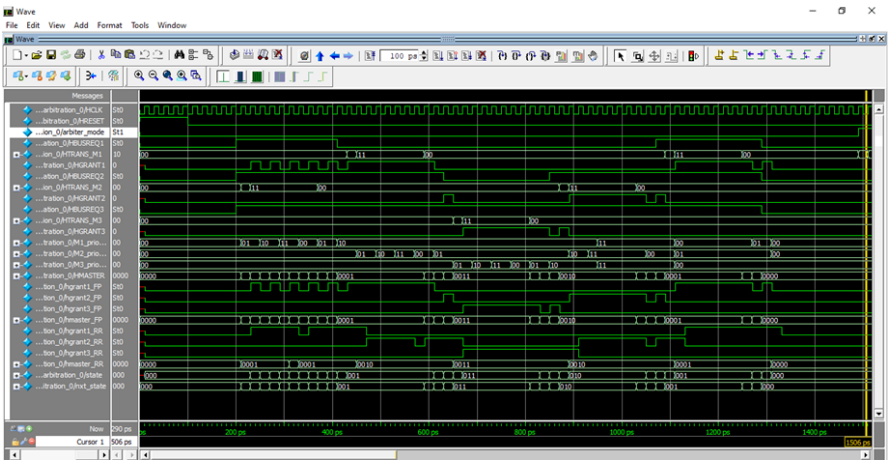


Figure 11. Arbitration of 3 processors under Fixed Mode at mode signal 0

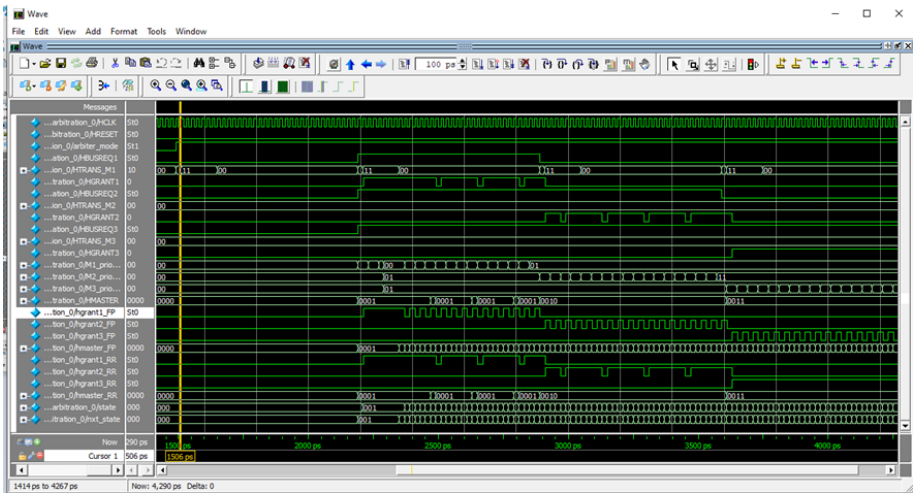
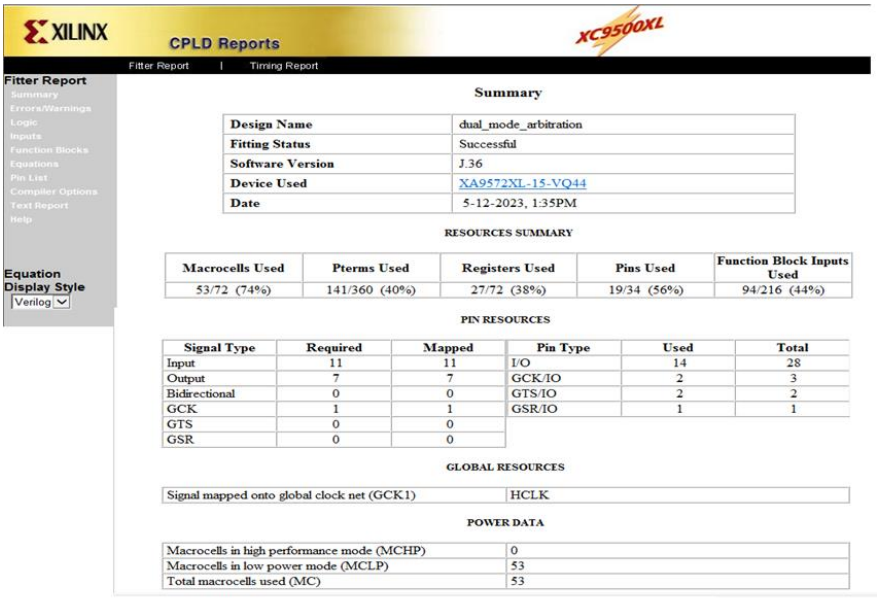


Figure 11. Arbitration of 3 processors under Round Robin Mode at mode signal

8. HARDWARE SYNTHESIS AND IMPLEMENTATION

The Verilog hardware model of this research work has been synthesized by Xilinx's XST, targeting its FPGA device XA9572XL-15-VQ44. Figure 12 shows the synthesis report summary and the hardware usage in the targeted FPGA device.



XILINX CPLD Reports XC9500XL

Fitter Report | Timing Report

Fitter Report
Summary
Errors/Warnings
Logic
Inputs
Function Blocks
Equations
Pin List
Compiler Options
Text Report
Help

Equation Display Style
Verilog

Summary

Design Name	dual_mode_arbitration
Fitting Status	Successful
Software Version	J.36
Device Used	XA9572XL-15-VQ44
Date	5-12-2023, 1:35PM

RESOURCES SUMMARY

Macrocells Used	Pterms Used	Registers Used	Pins Used	Function Block Inputs Used
53/72 (74%)	141/360 (40%)	27/72 (38%)	19/34 (56%)	94/216 (44%)

PIN RESOURCES

Signal Type	Required	Mapped	Pin Type	Used	Total
Input	11	11	I/O	14	28
Output	7	7	GCK/IO	2	3
Bidirectional	0	0	GTS/IO	2	2
GCK	1	1	GSR/IO	1	1
GTS	0	0			
GSR	0	0			

GLOBAL RESOURCES

Signal mapped onto global clock net (GCK1)	HCLK

POWER DATA

Macrocells in high performance mode (MCHP)	0
Macrocells in low power mode (MCLP)	53
Total macrocells used (MC)	53

Figure 12. Synthesis Summary Report

Figure 13 shows the top-level block with all the input and output signals after the synthesis.

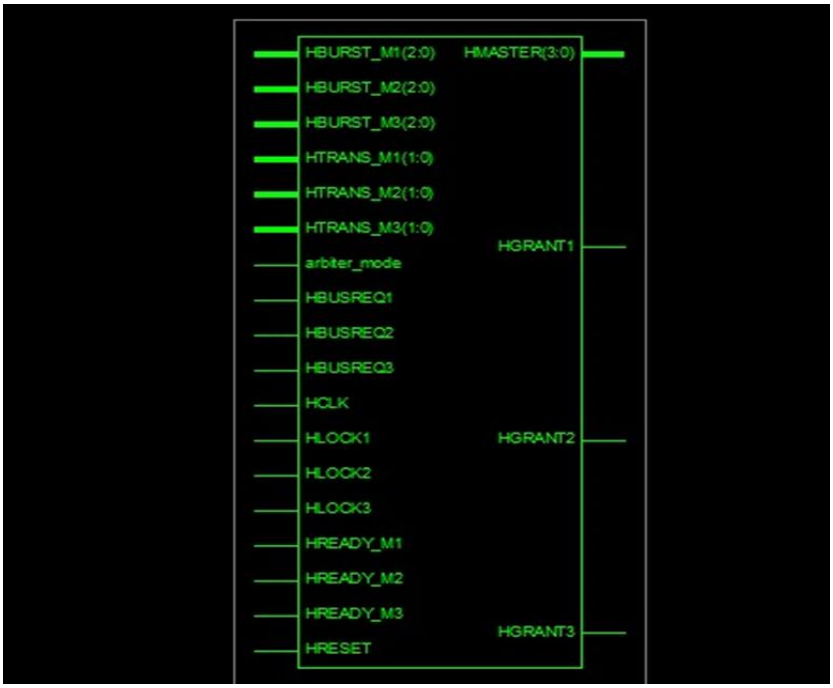


Figure 13. Top Block of Dual Mode Arbiter after synthesis

Figure 14 shows the hierarchical detail schematic of the Dual Mode Arbiter after synthesis. The detailed schematic shows all the major blocks inside the top block, which includes the Fixed Mode arbiter, Round Robin Arbiter, Grant Controller, and the rest of the glue logic.

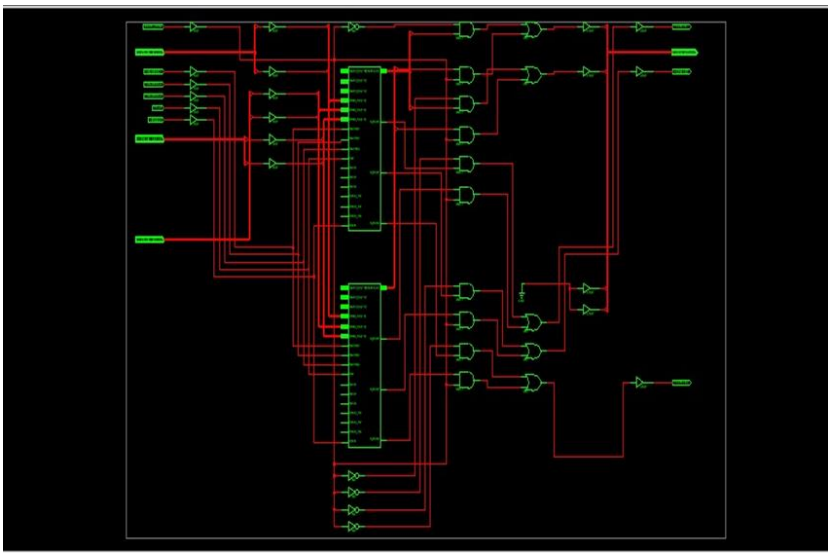


Figure 14. Hierarchical Schematic Diagram of Dual Mode Arbiter.

Post synthesis detailed internal schematic of sub-blocks Fixed Mode arbiter, Round Robin Arbiter have been shown in Figure 15 and Figure 16.

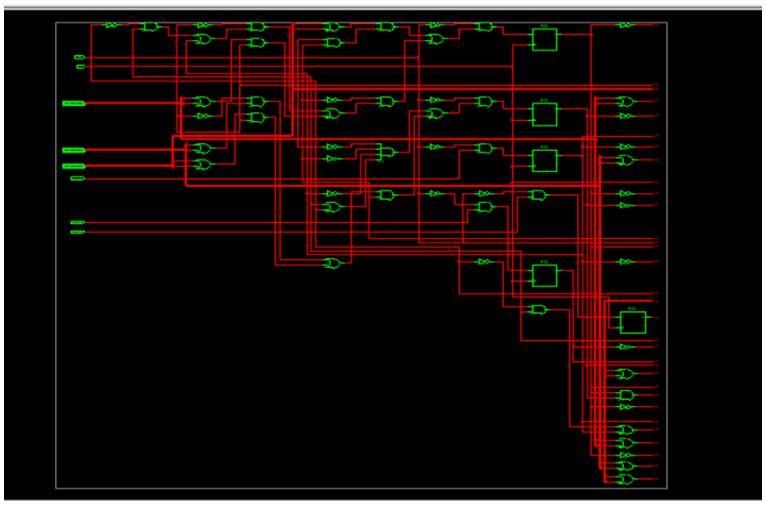


Figure 15. Schematic of Sub Block Fixed Mode Arbiter

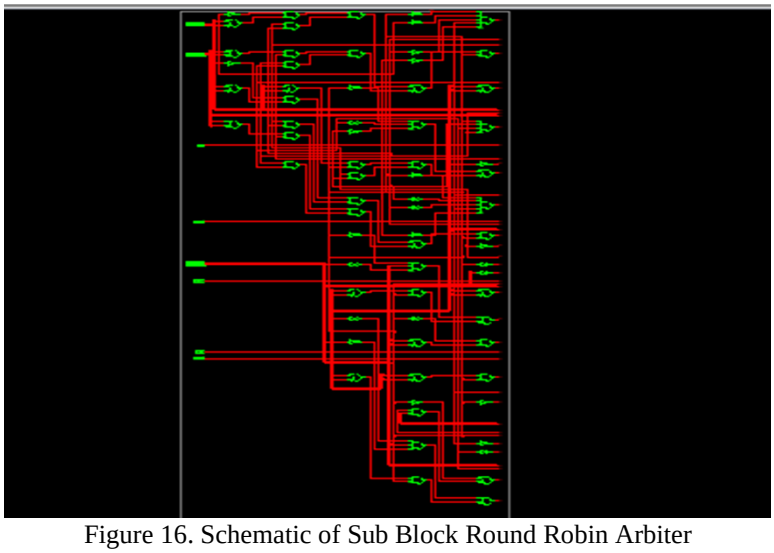


Figure 16. Schematic of Sub Block Round Robin Arbiter

9. CONCLUSION

As opposed to a single processor interface in SoC, a multiprocessor hardware interface in SoC is a critical need in current technology to keep up with the vast size and complexity of high-speed chips nowadays. AMBA protocol is an industry-standard bus protocol popularly used in modern chips. Also, for efficient access to targets and based on the system's need to manage several processors' access to target devices, an efficient arbitration scheme is a critically necessary hardware to implement system hardware. Single-mode arbitration implements one arbitration scheme in hardware, and the chip has to stick to it still. Even modern system requires flexibility to use different arbitration scheme too. The dual mode arbiter of this research work and hardware implementation gives the system the hardware option to go for two types of arbitration in the chip as per system need. This research is designed and modeled using Verilog HDL hardware modeling methodology, simulated using both Modelsim and Cadence EDA simulator, and finally synthesized and implemented in Xilinx FPGA device. For better speed, power, and area, this hardware can be fabricated in ASIC or SoC in the future at a foundry.

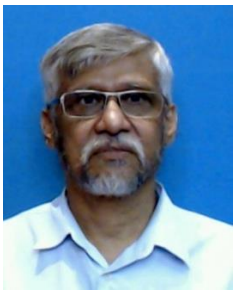
REFERENCES

[1] P. Giridhar and. P. Choudhury, "Design and Verification of AMBA AHB," 1st International Conference on Advanced Technologies in Intelligent Control, Environment, Computing & Communication Engineering (ICATIECE), 19-20th Mar 2019, India.
[2] H. Saluja and N. Grover, "Multiple Master Communication in AHB IP using Arbiter," International Journal of Engineering and Manufacturing (IJEM), vol. 10(1), pp. 29-40, 2020.
[3] L. Deekhsa and B. R. Shivakumar, "Effective Design and Implementation of AMBA AHB Bus Protocol using Verilog," 2019 International Conference on Intelligent Sustainable Systems (ICISS), 21-22th Feb 2019, Palladam, India.
[4] J. Tu and C. Ou, "A practice of 2-to-2 fixed priority arbiter used to multi-core processors," International Conference on Applied System Innovation (ICASI 2017), 13-17th May 2017, Sapporo, Japan.
[5] A. Mishra, "Design of a Round Robin Bus Arbiter using System Verilog," International Research Journal of Engineering and Technology (IRJET), vol. 7(1), pp. 29-40, 2020.
[6] A. K. Qahtan and A. M. A. El-Kustaban, "A Bus Arbitration Scheme with an Efficient Utilization and Distribution," International Journal of Advanced Computer Science and Applications ((IJACSA 2017), vol. 8(3), pp. 1-6, No. 2017.
[7] M. I. R. Rokon, S. M. A. Motakabber, M. Hadi Habaebi, A. H. M. Zahirul Alam, M. A. Matin, "Smart Arbitration System for Multiprocessor AMBA Interface in System on Chip," 2021 8th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, Malaysia. 22-23 June
[8] M. I. R. Rokon, S. M. A. Motakabber, A. H. M. Zahirul Alam, M. Hadi Habaebi, M. A. Matin, "Multiprocessor Arbitration for AMBA Interface in ASIC," Asian Journal of Electrical And Electronic Engineering, vol. 1, no. 2, pp. 20-27, 2021.
[9] M. I. R. Rokon, S. M. A. Motakabber, A.H.M. Zahirul Alam, M. Hadi Habaebi, M. A. Matin, "Fairness in Multiprocessor Hardware Access to System-on-Chip," IEEE 5th International Symposium on Robotics and Manufacturing Automation (ROMA 2022), 6-8 August 2022, Malakka, Malaysia.
[10] Lei Zhu; Lixin Yu, "A design of decentralized dual mode redundant hot standby arbitration switch-over logic and architecture," IEEE 2018 International Conference on Electronics Technology (ICET), DOI: 10.1109/ICET44212.2018, 23-27 May 2018.
[11] Reza Mirosanlou, "Adaptive Dual-Mode Arbitration for High-Performance Real-Time Embedded Systems," University of Waterloo thesis. January 25, 2022.
[12] Bekim Cilku; Alfons Crespo; Peter Puschner; Javier Coronel; Salvador Peiro, "A TDMA-Based arbitration scheme for mixed-criticality multi-core platforms," "IEEE 2015 International Conference on Event-based Control, Communication, and Signal Processing (EBCCSP). 17-19 June 2015.

BIOGRAPHIES OF AUTHORS

Mohammed Iqbalur Rahman Rokon received his B.Sc. in Electrical and Electronic Engineering from Bangladesh Institute of Technonology, Rajshahi, (currently RUET) in 1991 and MS in Electrical Engineering from Califirnia State University, Northridge, USA, in 1997. Then he started his industry career in US high tech company Emulex Corporation, Costamesa, Cailoforn, USA, as ASIC Development Engineer. There he worked until 2003. Then he returned to Bangladesh and started his teaching career as a faculty member of Electrical and Computer Engineering Department at North South University (NSU), Dhaka, Bangladesh and still he is a faculty member of NSU. In 2019, he started his PhD program at International Islamic University Malaysia and he is expecting to complete his PhD by October, 2023.

S. M. A. Motakabber who was born in Naogaon, Bangladesh received the BSc (Honours) and Master's degrees from the University of Rajshahi in 1986 and 1987, and Ph.D from University Kebangsaan Malaysia in 2011. He served as a scientific officer at the Bangladesh Atomic Energy Commission, and Bangladesh Scientific and Industrial Research, Dhaka from 1991-1992 and 1992-1993, respectively. He started his teaching career as a lecturer in the Department of Applied Physics and Electronics, University of Rajshahi in 1993. The following year he was appointed an Assistant Professor in the same department. He joined as an Associate Professor in the Department of Computer and Communication Engineering at International Islamic University Chittagong in 2003; also served as Head of the Department and Dean of the Engineering Faculty.
He served as an Associate Professor in the Department of Applied Physics and Electronic Engineering from 2008 to 2012 at the University of Rajshahi. Dr. Motakabber joined as an Assistant Professor in the Department of Electrical and Computer Engineering at the International Islamic University Malaysia (IIUM) in 2012.



AHM Zahirul Alam received the B.Sc. and M.Sc. degrees in Electrical and Electronic Engineering from Bangladesh University of Engineering and Technology (BUET) in 1984 and 1987, respectively. He obtained his Doctor of Engineering degree from Kanazawa University, Japan in 1996. He was working as a faculty member in BUET from 1985 to 1991 and from 1996 to March 2002. He became a professor in BUET in 1999. He worked in the MIRAI project in Low-k group in Advanced Semiconductor Research Center, Tsukuba, Japan through Japan Science and Technology fellow from April 2002 to October 2003. He is currently serving as a Professor of Electrical and Computer Engineering Department, Faculty of Engineering, International Islamic University Malaysia (IIUM). Detailed activities of his profile can be found in his website at https://www.iium.edu.my/zahirulalam.

Dr. Mohammad A Matin is a Professor of the Department of Electrical and Computer Engineering at North South University (NSU), where he has been since 2008. He was first appointed as Assistant Professor and then promoted to Associate Professor at North South University in 2011 and later on Professor. While in that post he was also the coordinator of EEE program. During 2012-2017, he was an Associate Professor at Universiti Teknologi Brunei (UTB), Brunei Darussalam (QS World University ranking 379). He received his B.Sc. degree in Electrical and Electronic Engineering from BUET (Bangladesh), his M.Sc. degree in Digital Communication from Loughborough University, UK and PhD in Wireless Communication from Newcastle University, UK. He has also taught several courses in communications, electronics and signal processing at KUET, Khulna University, BRAC University, and UKM (top ranked uni. In Malaysia), UM (top ranked Univ. in Malaysia) at Masters and undergraduate level as well engaged himself with Maters theses and projects during his career. He has published over 150 peer-reviewed journals and conference papers as a sole author and with the students of NSU and other universities.

Md Mahmud is doing MES in Electrical Engineering at Lamar University, Texas, USA and he completed his Masters of Science in Electronics Engineering (2021) from International Islamic University Malaysia. He obtained his Bachelor of Science in Electrical and Electronic Engineering (2017) from Daffodil International University (DIU), Bangladesh. Md Mahmud has a very strong research interest in the fields of science and technology. He has published over 12 scientific journal papers and conference proceedings. Md Mahmud is currently a member of IEEE, IEEE Consultants Network, and Past President, Rotaract Advisor, Rotaract Club of Daffodil International University (DIU), Bangladesh. I am very self-motivated, energetic, driven and people person. I was born in Noakhali, Bangladesh.