

HDL IMPLEMENTATION OF AMBA AHB ON 28NM FPGA

Anshu Gaur¹, Piyush Sharma², Shiv Pratap Pandey³

¹M.Tech Scholar, Dept. of Electronics and Communication, SS College of Engineering Udaipur, Raj., India

²Asst. Professor, Dept. of Electrical, SS College of Engineering Udaipur, Raj., India

³Asst. Professor, Dept. of Electronics and Communication, SS College of Engineering Udaipur, Raj., India

Abstract

In this paper the proposed design of AHB (Advanced High Performance Bus) AMBA (Advanced Microcontroller Bus Architecture) bus has been implemented while considering the low power dissipation of On-chip communication. The FPGA device used for the implementation is 7k70tbg484-3 Kintex7 having speed grade-3 based on the 28nm technology. The design produces the total power dissipation of 0.08watt including the leakage power with the help of Xilinx Power Analyzer Tool.

Keywords: FPGA, AMBA, AHB.

1. INTRODUCTION

The AMBA AHB has been design working model and calculate the power consumption of this model, we can reduced the on- chip power of circuit using clock gating. One of the dominant power contributors in VLSI design is clock signal, which is about 70%-30% total of the dynamic power of the device [1]. To reduce the dynamic power many methods are used but clock gating is one of the efficient approach in reducing dynamic power. The clock gating technique used AND-OR gate to reduce the power of device. In the field of clock gating negative latch based method was proved best [6]. Disable the clock signal when it is not needed [2] [3].

The AMBA AHB is advanced microcontroller bus architecture is an on-chip communication standard which is used in high performance microcontroller. The AMBA AHB used for high performance high clock frequency system modules. The AHB is the high performance system back bone bus. There are three separate buses within the AMBA specification are Advanced High-performance Bus (AHB), Advanced System Bus (ASB), and Advanced Peripheral Bus (APB). The four main component is comprised by AMBA AHB are AHB master, AHB slave, AHB arbiter, AHB decoder. There are multiple masters and multiple slaves available. In AHB multiple masters and multiple slaves are available. Decoder decodes the address given by masters and interacts with respective slaves. But at a time only one slave can be used. Arbiter provides buses to master for a communication. If the number of arbiter is increased then the number of buses will be increased. All the operation has been done under the synchronized clock, i.e. some with positive edge of the clock and some with negative edge of the clock.

1.1 AHB Master

A master initiates the read/write operation by providing the control information and an address of respective slave to the arbiter. Firstly the master decides which slave it wants to communicate. The master requests for the bus HBUSREQ

signal to the arbiter. If the channel is free, the request is grant by arbiter then HGRANT signal set high which means bus has been provided to the master. Now the master can communicate with the slave by address of memory, location of slave. Read/Write operation can be performed. All the operations are done in the positive edge of the clock.

The synthesized AHB master block diagram is shown in figure1.

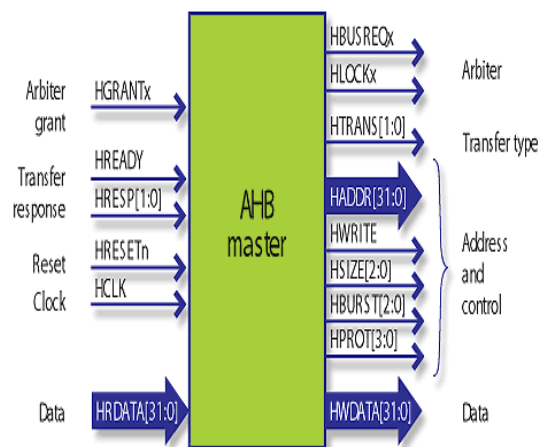


Fig-1: AMBA AHB MASTER

1.2 AHB Slave

The slave respond to the read write operation given by master. The address given by the master to which slave it want to communicate, then the adders of memory location in slave, it wants to read or write the content. Slave also mention the type of communicate take place. The four type of communication mention is these design. The acknowledge signal used to check the communication has been done successfully or fails, by slave signal back to master. If read operation requested then it will read the data from respective memory location and will give back to the master. If write operation is requested then it will take

memory location where the data has to be write in the memory. The slave always signals back to the master for its each request. Synthesized Block diagram is shown in figure2.

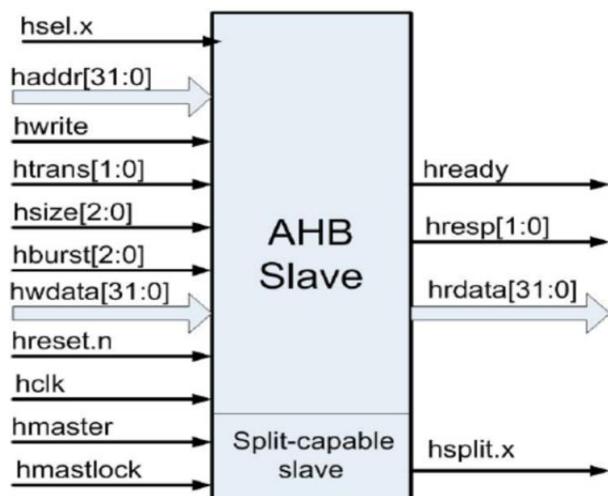


Fig -2: AMBA AHB SLAVE

1.3 AHB Arbiter

Arbiter provides bus for communication between master and slave. The master requests for bus to the arbiter. If the bus is free then arbiter grants the request.

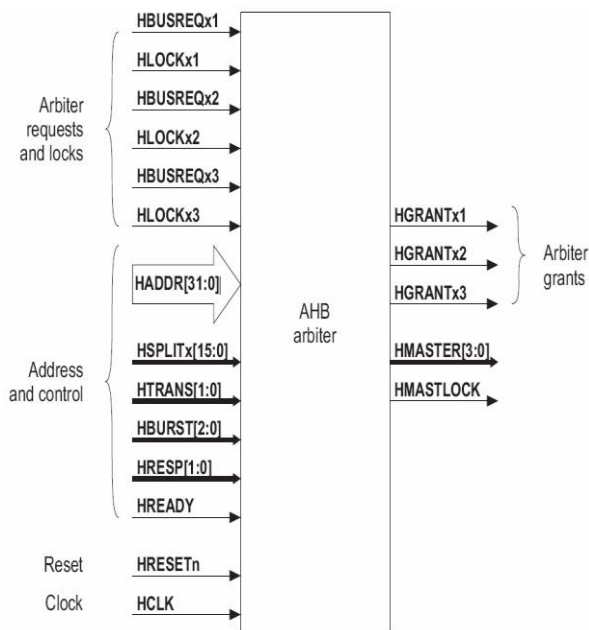


Fig -3: AMBA AHB ARBITER

If bus is not free then arbiter denied the request. The arbiter allowed one bus master to perform the data transfer. For a single bus there is need of only one arbiter. If two buses are in use, then master can transfer the data from both buses and there is need of two arbiters. The arbiter is needed because master can focus on data transfer only, not on the buses whether other master is using it or not. If arbiter is not in design then two master can request for same bus. The block diagram of AHB ARBITER is shown in figure3.

1.4 Decoder

The address of each transfer is decoded by decoder. The master provides the address of respective slave to it want Communicate. The address is in encoded form. The decoder decodes the address on slave side to complete the communication. This unit is responsible for decoding AHB as shown in figure 4.

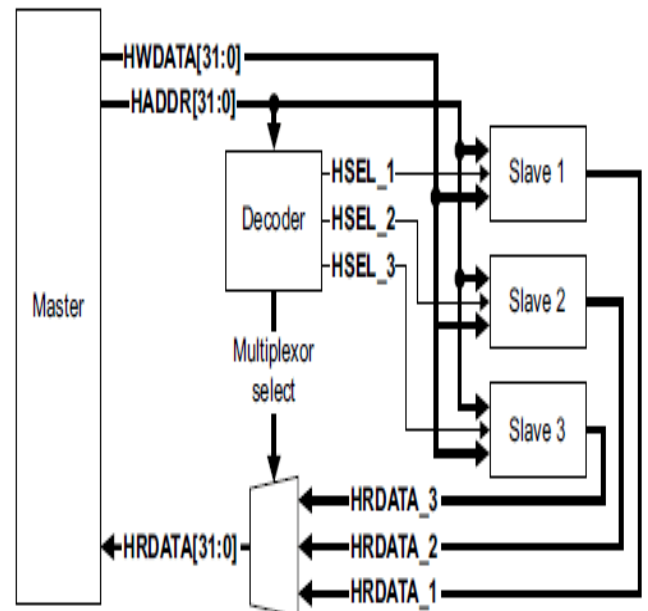


Fig -4: AMBA AHB Decoder

1.5 Clock Gating Technique

Using clock net in clock gating technique we can control power dissipated. In synchronized digital circuit clock net is responsible for significant power dissipation.

The clock gating is basically reducing unwanted switching of the clock in different part of the design by switching off the clock when it is not required. RTL level technique is common in clock gating to optimize and improve the efficiency of any synchronized digital design. The negative Latch Based clock gating technique is used to reduce the power of device. The negative latch based clock gating is best technique as stated in [11][12].

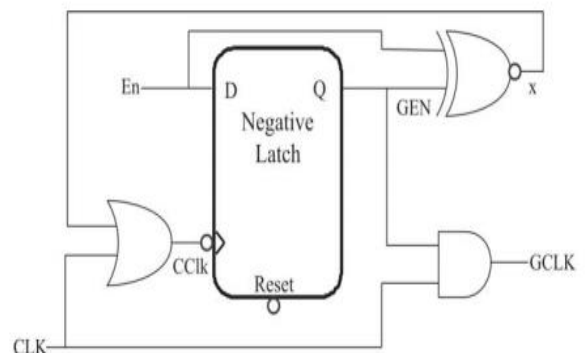


Fig -5: Negative Latch based Clock gating Technique

2. METHODOLOGY USED

In ASM of master there are three states. In the S0 state the HBUSREQ signal is requested for the bus to the arbiter. When HBUSREQ signal becomes high, then S0 moves to next state S1. The arbiter responded in S1 state. S1 remains its state till it gets response from arbiter. Arbiter give response in form of HGRANT signal is high. HREADY signal is high which shows transaction has been completed. When both HGRANT and HREADY signals are then it select the transfer type. HTRANS is the transfer type in which simple transfer take place. Then it move to next state S2. Again it looks for the HREADY signal. Then it check HWDATA if the signal is high then write operation will be performed otherwise read operation will be performed. The HTRANS [00] is idle transfer type, no data transfer is required. Then it move to in state S3 where new data is input of this state. If value of new data is high then again request for bus to arbiter. If new data value is 0 then it will remain in S3 state. These methods are explained with the help of ASM (Algorithmic state Machine) shown in fig 6.

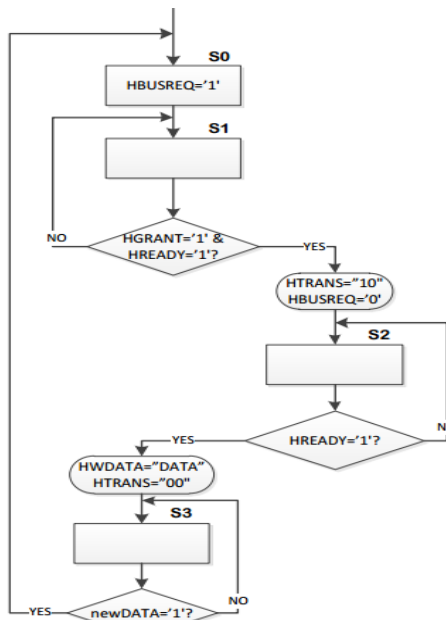


Fig -6: ASM blocks for AHB MASTER

ASM (Algorithmic State Machine) of SLAVE

In the ASM of master and slave shown in figure 6 and figure 7 respectively, block it consists of two state only. S0 and S1. In S0 state it waits for the HTRANS '10' which means for simple transfer type. The master wants a simple transfer, and slave acknowledge back with another signal. The HREADY signal is acknowledge by slave if it is not busy. HPOS signal store the ID of master which is communication by it. Now it moves in next state S1. The data from the master is stored in the vector. The slave give HRESP equal to "00" signal back to the master, which means data has been successfully received and HREADY=1 means communication is completed. Then it look for HTRANS=10 it means it want to start a new transfer. If HTRANS=10 then it will go back to state S0.

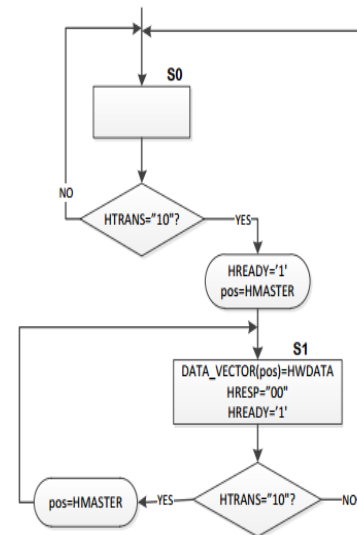


Fig -7: ASM blocks for AHB SLAVE

ASM (Algorithmic State Machine) of ARBITER

The figure 8 shows the ASM of ARBITER. In S0 state waits for the assertion of any bit HBUSREQ bus. It means at least one master want to access the bus. The for loop is there for master that requires the bus. When it found the bus, the HGRANT signal is asserted. In next state S1 wait for deassertion of HBUSREQ and HGRANT. Then process will start again when more master are requested for bus. Else go back to state S0.

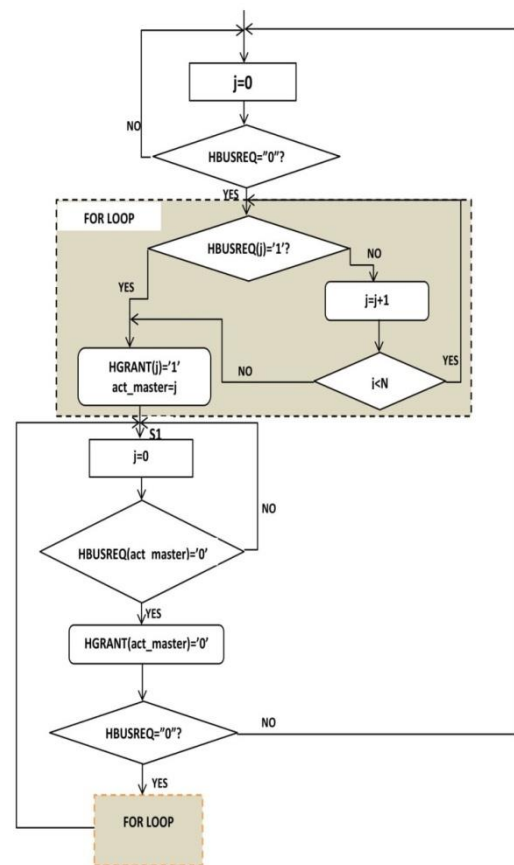


Fig -8: ASM blocks for AHB ARBITER

3. IMPLEMENTATION and RESULTS

The circuit of AMBA AHB is implemented on the kintex 7 FPGA device with the help of Xilinx 14.7 version simulation. The design produces the total power dissipation of 0.08watt which is low as compared [8].The synthesized and simulated results are shown below in figure 9.

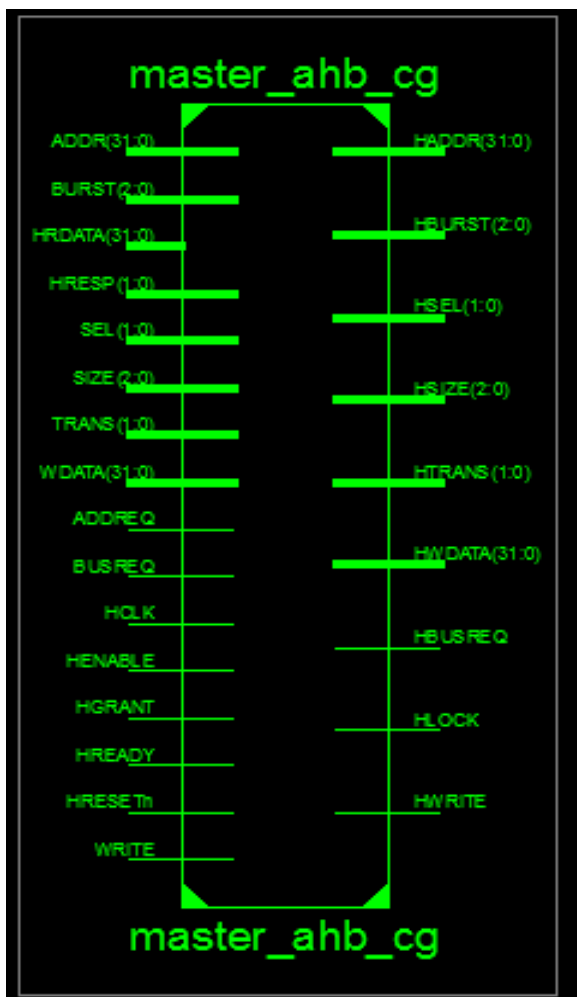


Fig -9: Synthesized module of AHBMaster



Fig -10: Technology Schematic of AHB module

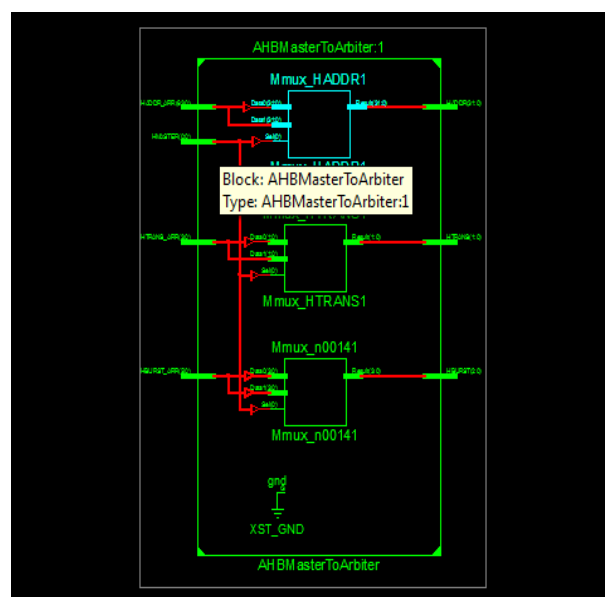


Fig -11: Technology Schematic of AHB Master to Arbitrer

The Technology schematic figure is shown in figure 10 and figure 11.

Simulated Waveform for the AHB communication protocol for the transmission of data from master to slave is shown in figure 12.

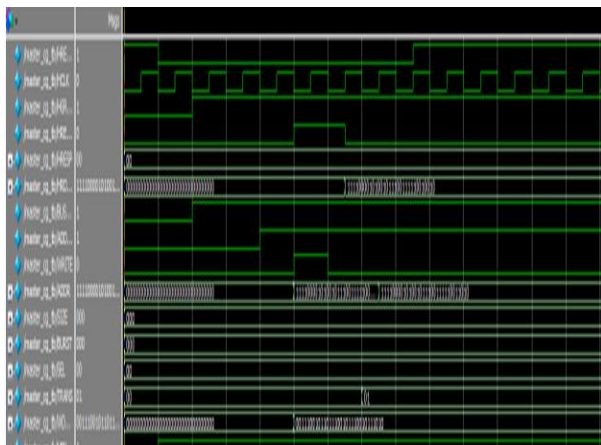


Fig 12: Simulated waveform of data from Master to Slave

4. POWER DISSIPATION

The Power dissipation is calculated using the Xpowerflow tool using Xilinx Power Flow analyser tool shown in figure 13 and table of Device utilization summary shown I table 1.

A	B	C	D	E
On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	46	41000	0.11
Signals	0.000	160	---	---
I/Os	0.000	150	285	53
Leakage	0.080			
Total	0.080			

Fig -13: On Chip Power dissipation of AMBA AHB.

Table -1: Table of Device utilization summary

AHB	Frequency	I/O (%)	Number of fully used LUT Flop pair	Signal	Logic used (out of 41000)	Total Power (W)
ARBITER	501.832 MHz	7	41 out of 98	131	96	0.080
DECODER	N.A	2	0 out of 3	7	2	0.080
MASTER TO ARBITER	N.A	39	0 out of 37	112	19	0.080
MASTER TO SLAVE	1759.015 MHz	84	1 out of 81	242	42	0.080
SLAVE DUMMY	N.A	14	0 out of 2	18	11	0.080
SLAVE TO MASTER	718.752 MHz	53	2 out of 46	160	46	0.080

5. CONCLUSION

The proposed AMBA (advanced microcontroller bus architecture) AHB (advanced high performance bus) design has been successively implemented on 28 nm technology device of FPGA 7k70tfg484-3 Kintex-7. The clock gating technique is the major factor in terms of power reduction. Negative latch based clock technique is used in this design and proved best.

REFERENCES

- [1] Benini, L.; Siegel, P.; De Micheli, G., "Saving power by synthesizing gated clocks for sequential circuits," Design & Test of Computers, IEEE ,vol.11, no.4, pp.32,41, Winter 1994
- [2] A. Farrahi, C. Chen, A. Srivastava, G. Tellez, and M. Sarrafzadeh, "Activity-driven clock design," IEEE Trans
- [3] Alhalabi, B.; Al-Sheraidah, A., "A novel low power multiplexer-based full adder cell," Electronics, Circuits and Systems, 2001. ICECS 2001. The 8th IEEE International Conference on , vol.3, no., pp.1433,1436 vol.3, 2001.
- [4] C. Chunhong, K. Changjun, and S. Majid, "Activity-sensitive clock tree construction for low power," in Proc.Int. Symp. Low Power Electron. Design, 2002, pp. 279–282.
- [5] Ashutosh Gupta and Kota Solomon Raju, "Design and Implementation of 32-bit Controller for Interactive Interfacing with Reconfigurable Computing Systems" International Journal of Computer Science and Information Technology (IJCSIT), Vol.1, No.2, pp 80-87, Nov 2009. ISSN: 0975-3826(online);0975-4660
- [6] Jagrit Kathuria, M. Ayoubkhan, Arti Noor, MIT International Journal of Electronics and Communication Engineering," A Review Of clock Gating Technique", MIT Publications, ISSN 2230-7672, Vol1, No.2, Aug 2011.
- [7] V. G. Oklobdzija, Digital System Clocking—High-Performance and Low-Power Aspects. New York, NY, USA: Wiley, 2003.
- [8] A. Gupta, K. Rawat, S. Pandey, P. Kumar, S. Kumar and H. P. Singh, "Physical design implementation of 32-bit AMBA ASB APB module with improved performance," 2016 International Conference on Electrical, Electronics, and Optimization Techniques

- (ICEEOT), Chennai, 2016, pp. 3121-3124.doi: 10.1109/ICEEOT.2016.7755276
- [9] Murgai, S.; Gupta, A.; Muthukrishnan, G., "Energy efficient and high performance 64-bit Arithmetic Logic Unit using 28nm technology," in Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on , vol., no., pp.453-456, 10-13 Aug. 2015doi: 10.1109/ICACCI.2015.7275650
- [10] Gaur, N.; Gupta, A.; Sharma, A.K.; Malviya, R., "HDL implementation of prepaid electricity billing machine on FPGA," Confluence The Next Generation
- [11] Information Technology Summit (Confluence), 2014 5th International Conference - , vol., no., pp.972,975, 25-26 Sept.2014. doi: 10.1109/CONFLUENCE.2014.6949328.
- [12] Ashutosh Gupta, ShrutiMurgai, AnmolGulati, and Pradeep Kumar, "Design and implement of low power clock gated 64 bit ALU on ultra scale FPGA", AIP Conference Proceedings 1715, 020001 (2016); doi: 10.1063/1.4942683
- [13] AnmolGulati, Ashutosh Gupta, ShrutiMurgai,LalaBhaskar, "Design and implement of power efficient 10-bit dual port SRAM on 28nm technology", AIP Conference Proceedings1715, 020002 (2016); doi: 10.1063/1.4942684.