Asic Design and Verification of Amba Apb Protocol using Uvm

K. Swetha Reddy, Punna Soujanya, D. Kanthi Sudha

***Abstract**: ASIC Implementation of AMBA APB convention with confirmation has been proposed right now. The structure presents Advanced Peripheral Bus Protocol (APB) in last part. To interface the peripherals, low data move capacity and low execution transport of APB is used. Henceforth, an altered ASIC plan with explicit less highlights, with better planning, low force necessity and less zone overhead, has been proposed. This plan is explicitly adept for advanced frameworks which have sequential transport interface necessity for on board correspondence. Additionally, the Firm IP centre of Master Controller has been intended for ASIC, which makes the structure exceptionally versatile on any ASIC chips or SOC plans. The whole custom ASIC execution of proposed configuration has been done in Synopsys Tool chain with 32nm standard cell library and this structure is verified utilizing Universal Verification Methodology (UVM).*

***Keywords: AMBA family, SoC, UVM(Verification),* ASIC Design.**

I. INTRODUCTION

Correspondence conventions for information exchange are normally fused on advanced frameworks utilizing an on-chip a master controller sub-framework. The conventions utilized for communication are commonly separated into two general classes: Parallel and Serial. Equal transports for all the interfaces are not a decent exchange off between cost, time, power and performance. Interchange preliminary transports are much efficient in ready information communication between various sub-frameworks on an Integrated Circuit (IC). The greater part of the peripherals on modern ASIC and SOC structures utilize sequential correspondence transports for information moves between processor or between processor a peripherals. Nowadays in this epoch of modern technology, multitudes of devices are integrated in SOC form. Numerous SoCs and ASICs that are provided by a few organizations. Today any reasonable person would agree that the ARM business standard for ASIC structure for versatile applications. Re-usable intellectual property(IP) which is fit for improving an ARM particularly imperative to any ASIC configuration focus. IP blocks and interconnection of many checked IP squares configuration rely upon Hardware Description Language (HDL) rather than schematic graphs. These RTL codes are robust tried for any utilization in the improvement of SOC. This area presents the protocol related work of the AMBA APB. Requirements and analysis of the protocol AMBA APB illustrated [1] . General definition for APB 3 Protocol adaptability and similarity appeared and the Verification of slave APB 3 Protocol.

They have given clear idea about Coverage analysis. It plays an important role in verification procedure; its significance contributes to what extent of source code has been tested for the DUT.

Verification efficiency can be increased by the Functional coverage analysis by allowing the verification engineer to separate the areas of un-tested function in the DUT [2].

Due to Rely, structure strategy is applied in the SOC (System-on-Chip) design.

An interface between the high-performance AXI bus and low-power APB domain can be provided by the bridge. Right now ACLK is independent.32-bit APB master to AXI slave information is moved [3]. The Memory Controller is a computerized circuit which controls the flow of information going to and from the primary memory.

It tends to be a different chip or can be incorporated into the framework chipset. For high performance microprocessor design, Power dissipation is turning into a restricting element because of consistently increasing device counts and clock rates.

Their suggested design of cutting edge AHB-MC, goal is to optimize power [4]. They explained the AMBA architecture in detail and UVM verification of the APB protocol. The UVM report outline additionally guarantees functional correctness of the structure [5].

The AMBA elite transport and extension among Master and slave with steady utilization of the memory controller is explained in detail. To keep away from the hand shaking process they utilized the FIFO. They primarily concentrated on area and speed [6][9].

Their paper gives a clear idea about AMBA bus design and clarifies the APB bus in detail. Using Verilog HDL the APB bus is designed according to the requirements and is verified utilizing Xilinx.

The AMBA (Advanced Microcontroller Bus Architecture) is an open source spec in controlling functional slices which includes System-on-chips SOCs (System-on-chips) [6][7]. Their paper portrays the structure age of Advanced Peripheral Bus (AMBA APB) protocol utilizing Perl scripting language. Here principle point is to decrease human interface in structure part such that we can reduce common sentence structure blunders.

Perl produces the Verilog configuration code of APB slave and its relating test bench, where every one of its determinations are there in XML content [8].

Revised Manuscript Received on July 30, 2020.
***** Correspondence Author

K. Swetha Reddy*, Assistant Professor, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology Hyderabad, India.

P. Soujanya, B.tech. ECE, Memorial College of Engineering and Technology, Devarakonda, Telangana, India.

D. Kanthi Sudha, M.Tech(Microwave Engineering)from Acharya Nagarjuna University, Andhra Pradesh, India.

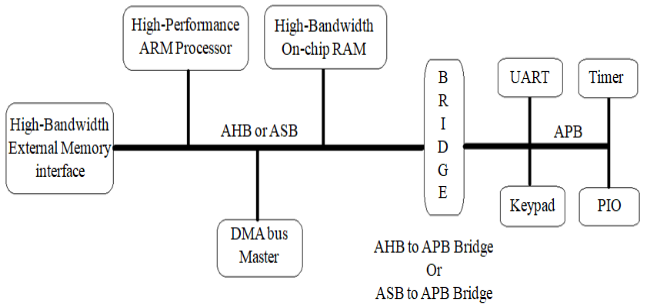

High-Performance ARM Processor
High-Bandwidth On-chip RAM
High-Bandwidth External Memory interface
AHB or ASB
DMA bus Master
BRIDGE
UART
Timer
APB
Keypad
PIO
AHB to APB Bridge Or ASB to APB Bridge

Fig 1: AMBA Bus Architecture

The Memory Controller is a computerized circuit which controls the flow of information going to and from the primary memory It tends to be a different chip or can be incorporated into the framework chipset. Their paper rotates around building an AMBA (Advanced Microcontroller Bus Architecture) consistent Memory Controller as an AHB (Advanced High-execution Bus) slave. The structure has dealt with balance between area overhead and speed. The read \write activity is practiced with zero wait states from the outer ROM and the compose activity with zero states to the outside RAM [11]. The AMBA (Advanced Micro-controller Bus Architecture) bus protocols is a set of interlink requirements from ARM that bring up on chip exchanging information methods between different operational blocks or IP blocks for developing more efficient SOC structures. A High-Performance system backbone bus (AMBA Advanced High-Performance Bus (AHB) or AMBA Advanced System Bus (ASB)) existing in the AMBA–based microcontroller which can support the outer memory data transfer capacity, on which the CPU, on-chip memory and other DMA (Direct Memory Access) devices located on it. A High-Bandwidth interface between the components that is associated with the majority of data transfers. Likewise situated on the high-performance bus is a bridge to the lower data transmission APB, where the majority of the peripheral devices in the framework are found. UART, Keypad, Timer and PIO (Peripheral Input Output) devices are connected to the APB. The high performance AHB or ASB bus to the APB bus can be connected by bridge. Thus, for APB the master will be the bridge and the slave is all the devices connected on the APB bus. The transactions can be initiated by the component on the high performance bus and distributes it to the peripherals connected on the APB. Along these lines, one after another the bridge is utilized for transfer of information between the high performance bus and the peripheral devices.

II. BLOCK PROGRAM

The APB is the part of a group the AMBA 3 protocol family which executes an ease interface which limits the power utilization and decreases the interface unpredictability. Since APB has non-pipelined convention. Hence, it combines to have low transfer speed components that don't request the superior of the pipelined bus structure. With the rising edge of the clock, all the signal transitions are related which makes it easy to incorporate APB peripherals into any plan. AMBA AHB-Lite and AMBA Advanced Extensible Interface (AXI) can be utilized by APB. Likewise APB can be utilized to get to the programmable control registers of the peripheral gadgets.

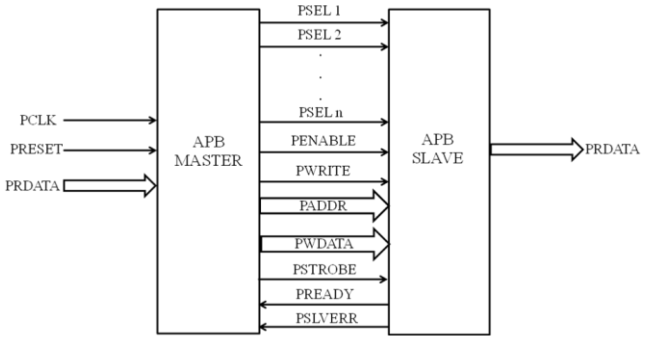


Fig 2: Communication between APB master and APB slave

In APB, for a communication protocol master is a model where one gadget or strategy hold back at least one different gadgets or procedures (known or slaves). The heading of control is consistently from the master to the slave, when the master/slave relationship is set up.

Table 1: APB Signal Description

Signal	Signal Description
PCLK	The bus clock source (Rising Edge- triggered)
PRESET	Reset signal
PADDR	The APB address bus (can be up to 32-bits wide)
PSEL	The selection line for each slave device
PENABLE	Indicates the second and subsequent cycle of an APB transfer
PWRITE	Indicates the transfer direction (Write=HIGH, Read=LOW)
PWDATA	The write data bus (can be up to 32-bits wide)
PREADY	Used to extend a transfer
PRDATA	The read data bus(can be up to 32-bits wide)
PSLVERR	Indicates a transfer Error (OKAY=LOW,ERROR=HIGH)
PSTROBE	Indicates which byte lanes to update during a write transfer. It shows that the bus contain valid data, when PSTRB[3:0]=1111

There is a solitary transport master on the APB, in this way there is no necessity for a mediator. The master tailgates the address and compose transports and furthermore plays out a combinatorial translate of the location to choose which PSEL x sign to initiate. It is likewise liable for driving the PENABLE sign to time the exchange. It also tailgates APB information onto the framework transport during a peruse move. APB slaves have a straightforward, yet adaptable, connection. The specific execution of the connection will be reliant on the planned approach utilized and various choices are conceivable.

Right now flags present, which principally ensure the misfortune information while move of information is occurring. Those signs are PSLVERR and PREADY.

III. APB CONTROLLER

Figure 3 depicts the key finite state machine that addresses action of the APB (Advanced Peripheral Bus). Typically it is classified into three states specifically i.e. IDLE, SETUP and ACCESS states. During the IDLE state, there is no change is being performed and it is default one. The declaration of the PSEL signal shows the start of the SETUP composes.When the information move is required, the transport goes into the SETUP arranges. During this stage the PWRITE, PADDR and PWDATA are also given.

For one clock cycle and on the accompanying rising edge of the clock, the transport continues going into the SETUP arrange and it will move to the Entrance state. Declaration of the PENABLE sign shows the start of the ACCESS stage. All the address, control signs and the data signals remains stable during the advancement from the SETUP stage to the ACCESS stage. On the off chance that there ought to emerge an event of read movement the PRDATA is accessible on the transport during this stage. PENABLE sign in like manner remain high for 1 clock-cycle. If no additional data move is required, the bus will move to the IDLE state. However, in the occasion that additional data move is required, by then the transport will move to the SETUP stage.

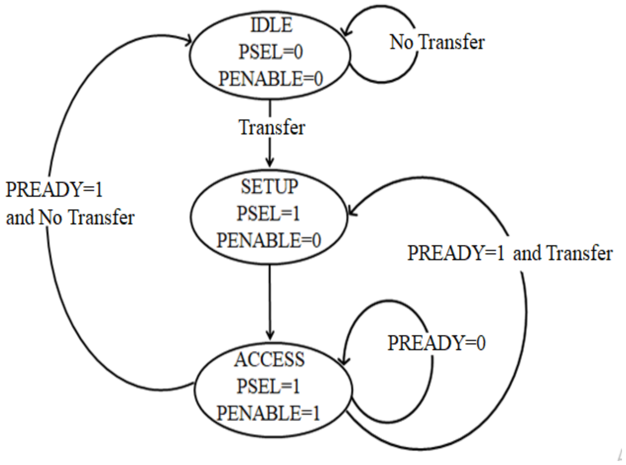


Fig 3: State diagram of APB

A.WRITE Cycle with no WAIT States:

After the rising edge of the clock, the compose activity begins with the location (PADDR), compose information (PWDATA), compose signal (PWRITE) and select sign (PSEL) all changing. The initial clock cycle for the start of process is known as the Setup stage. After going with clock edge the enable sign is expressed, PENABLE, and this shows the Access stage is happening.

The information, location and control flags all stay real all through the Access stage. The exchange completes close to the completion of this cycle. The enable signal, PENABLE, is de-asserted close to the completion of the exchange. The select sign, PSELx, likewise goes LOW except if the exchange is to be followed promptly by another exchange to a similar periphery.

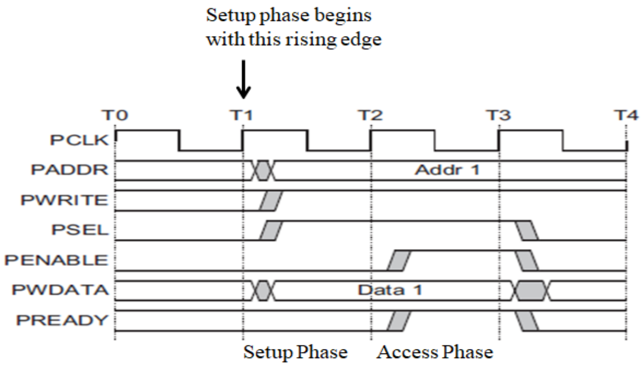


Fig 4: write cycle no wait states [1]

B.WRITE Cycle with WAIT States:

At the T1 clock-edge, the compose move activity, the PSEL, PWRITE, PADDR and PWDATA signals are attested which is known as the SETUP cycle. For the accompanying rising edge of the clock T2 and T3, the PENABLE sign is affirmed and PREADY is LOW for two clock cycles. This is known as the wait state. At the clock edge T4, PREADY signal is HIGH and PENABLE is HIGH is known as Access state. A high to low change happens on the PREADY signal at the clock edge PENABLE is de-asserted and if further information move is required.

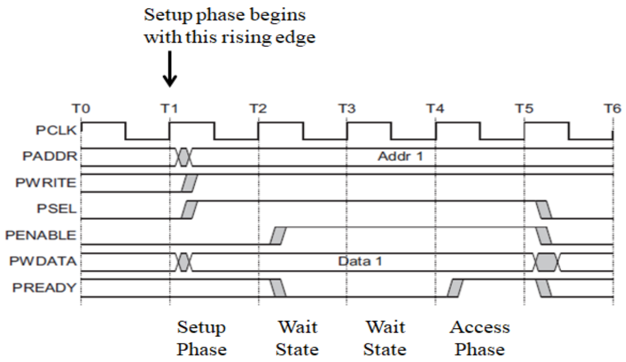


Fig 5: write cycle with wait states [1]

C.READ Cycle with no WAIT States:

During the read activity, the PADDR, PWRITE, PENABLE and PSEL signals are affirmed at the clock-edge T1 (SET-UP cycle). At the clock-edge T2, (ACCESS cycle), the PREADY, PENABLE are affirmed and PRDATA is likewise perused at this stage.

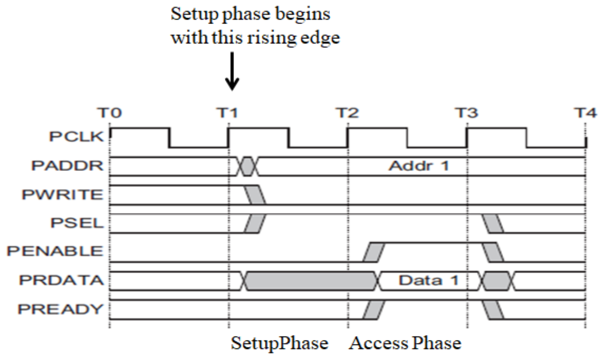


Fig 6: read cycle with no wait states [1]

D.READ cycle with WAIT states:

During the read activity, the PADDR, PWRITE, PENABLE and PSEL signals are stated at the clock-edge T1 (SET-UP Phase). For the next clock edges T2 and T3, (hold up express), the PENABLE is stated and PREADY is de-asserted. At the following edge T4,(ACCESS Phase),the PREADY and PENABLE is affirmed PRDATA is likewise perused during this stage.

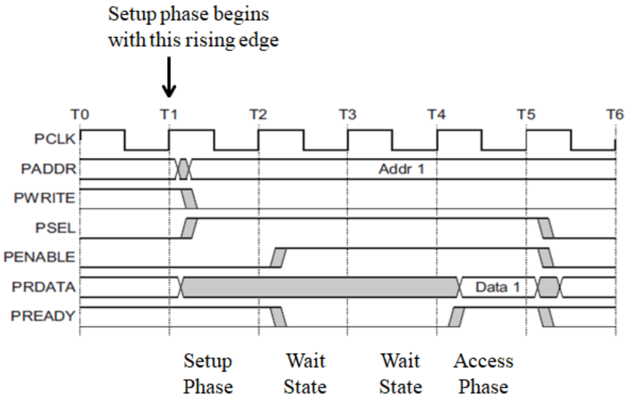


Fig 7: read cycle with wait states [1]

IV. VERIFICATION

Universal Verification Methodology (UVM) is a standard verification procedure used to confirm the RTL (Register-Transfer-Level) structure. It comprises of base class library coded in System Verilog. The verification specialist can enhance distinctive confirmation segments by expanding these classes. Also, UVM gives numerous other helpful verification highlights, for example, utilization of macros for executing complex capacity, manufacturing plant for object creation. Figure 8 shows the different UVM confirmation segments made to check APB structure.

In top clock is generated, DUT and interface are included and instantiated. The package is also included here. Mainly run test is done here. Here the environment is created in build phase. The actual interface is set for driver and monitor. The Different test cases can be created for the given verification-environment.

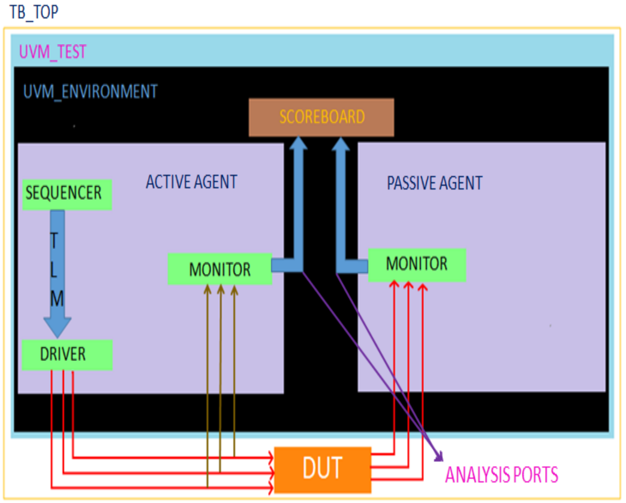


Fig 8: UVM Verification Environment

In environment the two agents namely, active agent passive agent and scoreboard are created .The agents and scoreboard connections using analysis ports. In active agent driver, sequencer and input monitor is instantiated and also connect these components using TLM port. Respective analysis port is also declared using uvm_analysis_port .In passive agent output monitor is instantiated. The analysis port is declared with only handle. Sequencer is in between driver and sequence. It sends the data packets to the driver. Driver sends the appeal for the following next data packet and drives the data to DUT. Here the virtual interface is handle gets the actual interface from the test. The input and output analysis port is created. For the comparison purpose uvm_analysis_fifo declared. The packets are received from the monitor to check the packets are matched or mismatched from the monitor.

V.ASIC DESIGN FLOW

In ASIC Design Style, Depending on the constraints given by user, Architecture for the design developed. RTL design for desired logic is obtained by using HDL, later simulation performed to check the desired functionality occurrence. Once functionality success hdl converted to netlist with synthesis process using netlist following physical design process, one can obtain layout for the logic. Physical comprises of steps like floorplanning, placement and routing. With clock tree synthesis, clock is distributed uniformly on the layout. Achieving optimal clock latency along with minimization of clock skew is the primary job for clocktree synthesis. In this paper, physical design of APB Protocol done with the use of Synopsys IC complier.

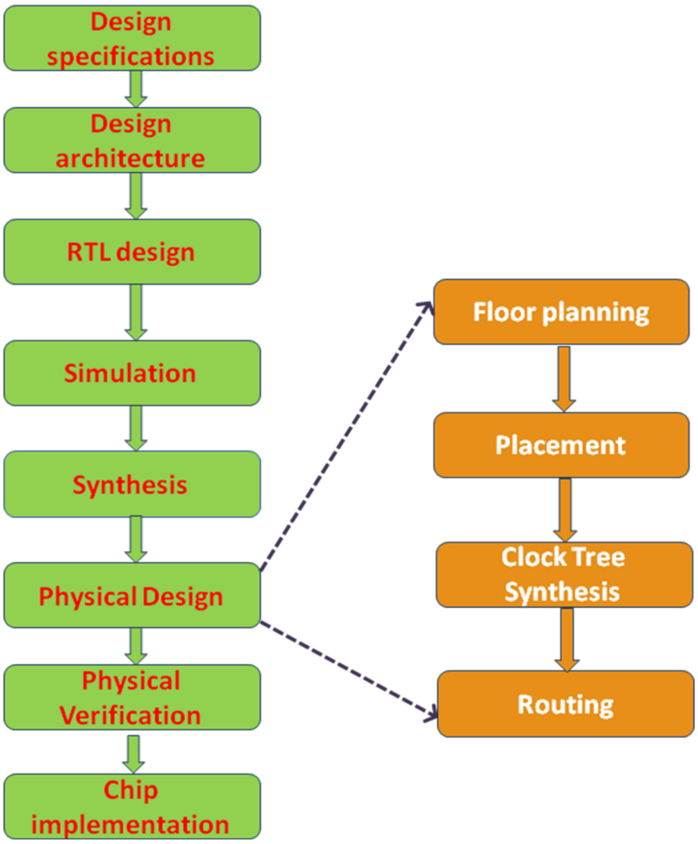


Fig 9: ASIC Design flow

VI. RESULTS

A.Simulation Results

Simulation results of APB protocol without wait states:
Figure 10 shows the recreation consequences of APB convention without hold up states got by Verilog Compiler Simulator (VCS) .Here the PREADY is high there is no sit tight for sending address and compose information .It is seen that in the reproduction results there are PADDR, PWDATA, PRDATA, PSLVERR, PSTRB.

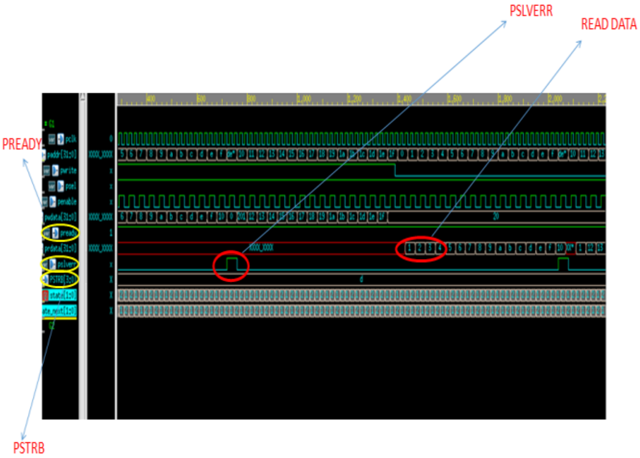
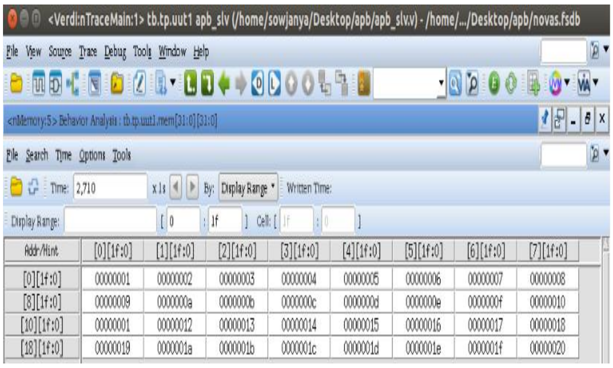


Fig 10 : Simulation results of APB protocol without wait states

These signs are of the memory which associated with the APB transport. In figure 10 the information 00000002 is kept in touch with the memory with the location 00000001. Similar information is perused from same memory area. The Unknown location is found PSLVERR sign will be high at the compose activity. The PSTRB signal is 1110; the inadequate information will be appeared in the memory. Figure 11 shows the memory is gotten by the Verdi environment. The 32-bit memory contains all 32 information.


<Verdi:nTraceMain:1> tb.tp.uut1 apb_slv (/home/sowjanya/Desktop/apb/apb_slv.v) - /home/.../Desktop/apb/novas.fsdb
File View Source Trace Debug Tools Window Help
<nMemory:5> Behavior Analysis : tb.tp.uut1.mem[31:0][31:0]
File Search Time Options Tools

Time: 2,710 x 1s By: Display Range Written Time:

Display Range: [0 : 1f] Cell: [1f : 0]

Addr/Hint	[0][1f:0]	[1][1f:0]	[2][1f:0]	[3][1f:0]	[4][1f:0]	[5][1f:0]	[6][1f:0]	[7][1f:0]
[0][1f:0]	0000001	0000002	0000003	0000004	0000005	0000006	0000007	0000008
[8][1f:0]	0000009	000000a	000000b	000000c	000000d	000000e	000000f	0000010
[10][1f:0]	0000001	0000012	0000013	0000014	0000015	0000016	0000017	0000018
[18][1f:0]	0000019	000001a	000001b	000001c	000001d	000001e	000001f	0000020

Fig 11: Memory behavior analysis

Simulation results of APB protocol with wait states:
Figure 12 shows the reproduction consequences of APB with hold up states got by the Verilog Compiler Simulator. The PREADY signal is low for the two clock cycles to get the location and information.

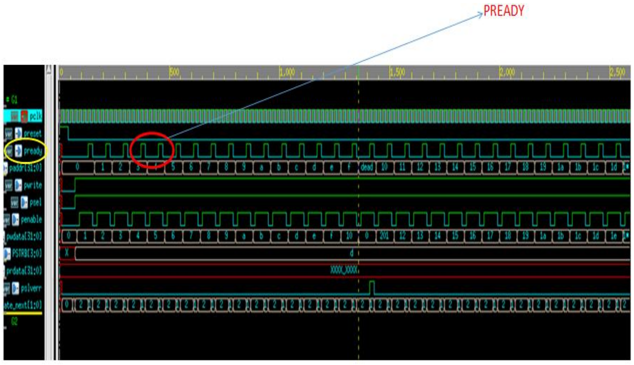
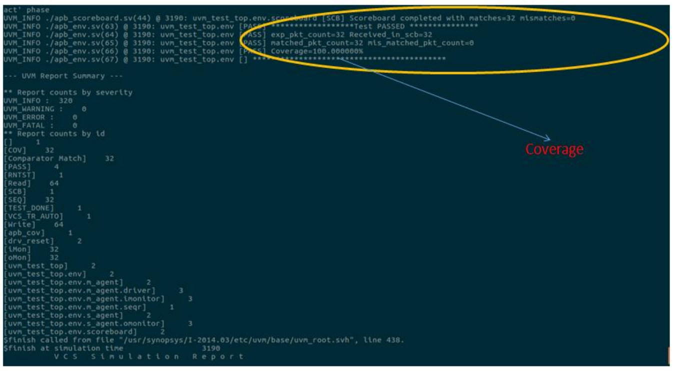


Fig 12: Simulation results of APB with wait states

B. Verification of APB Using UVM:

Figure 13 shows the UVM report rundown produced subsequent to running all the UVM stages. The UVM_INFO in the UVM report outline in figure 13 shows that there are three twenty data messages. The information gave by the UVM report rundown guarantees that structure is without blunder and doesn't create any admonitions or lethal mistake since the UVM_ERROR, UVM_WARNING and UVM_FATAL is equivalent to zero and the parcel tally is coordinated is 32, jumbles 0.The inclusion is acquired 100%.i.e all conceivable experiments are passed



```
act' phase
UVM_INFO ./apb_scoreboard.sv(44) @ 3190: uvm_test_top.env.scoreboard [SCB] Scoreboard completed with matches=32 mismatches=0
UVM_INFO ./apb_env.sv(63) @ 3190: uvm_test_top.env [PASS] ******************Test PASSED ***************
UVM_INFO ./apb_env.sv(64) @ 3190: uvm_test_top.env [PASS] exp_pkt_count=32 Received_in_scb=32
UVM_INFO ./apb_env.sv(65) @ 3190: uvm_test_top.env [PASS] matched_pkt_count=32 mis_matched_pkt_count=0
UVM_INFO ./apb_env.sv(66) @ 3190: uvm_test_top.env [PASS] Coverage=100.000000%
UVM_INFO ./apb_env.sv(67) @ 3190: uvm_test_top.env [] ****************************************************

--- UVM Report Summary ---

** Report counts by severity
UVM_INFO :  320
UVM_WARNING :    0
UVM_ERROR :    0
UVM_FATAL :    0
** Report counts by id
[]     1
[COV]    32
[Comparator Match]    32
[PASS]     4
[RNTST]     1
[Read]    64
[SCB]     1
[SEQ]    32
[TEST_DONE]     1
[VCS_TR_AUTO]     1
[Write]    64
[apb_cov]     1
[drv_reset]     2
[iMon]    32
[oMon]    32
[uvm_test_top]     2
[uvm_test_top.env]     2
[uvm_test_top.env.m_agent]     2
[uvm_test_top.env.m_agent.driver]     3
[uvm_test_top.env.m_agent.imonitor]     3
[uvm_test_top.env.m_agent.seqr]     1
[uvm_test_top.env.s_agent]     2
[uvm_test_top.env.s_agent.omonitor]     3
[uvm_test_top.env.scoreboard]     2
$finish called from file "/usr/synopsys/I-2014.03/etc/uvm/base/uvm_root.svh", line 438.
$finish at simulation time                 3190
           V C S   S i m u l a t i o n   R e p o r t
```

Fig 13: UVM report summary

C.Synthesis Result:

Fig13 shows amalgamation process. For the blend procedure we utilized Design Compiler (Dc) Tool. Here we are utilizing the standard cell library 32_nm. In the amalgamation procedure is first set the libraries (Target, Symbol and Link Library) and fundamental directions are (Read, Analyze and Elaborate) the structure. It peruses the (Register Transfer Logic) RTL Design and creates the planning, region, power reports and netlist.

Table 2: Synthesis Report

Total Cell Area	Dynamic Power	Leakage Power	Total Power	Clock Period	Slack	Total Cells	Compile time
13136.449469	1.4870 mW	757.2704 uW	2.544mw	4.0 ns	1.72	3426	5.52

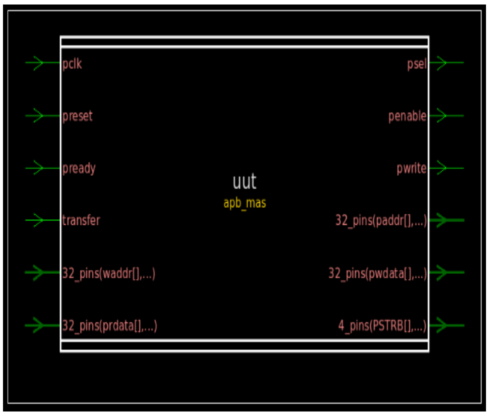


Fig 15: Schematic diagram of APB master

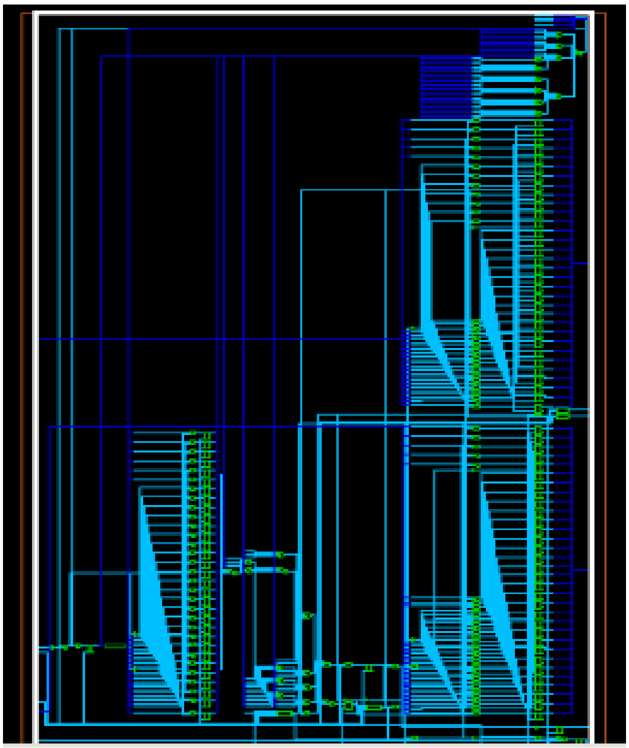


Fig 16: Internal Schematic view of APB master

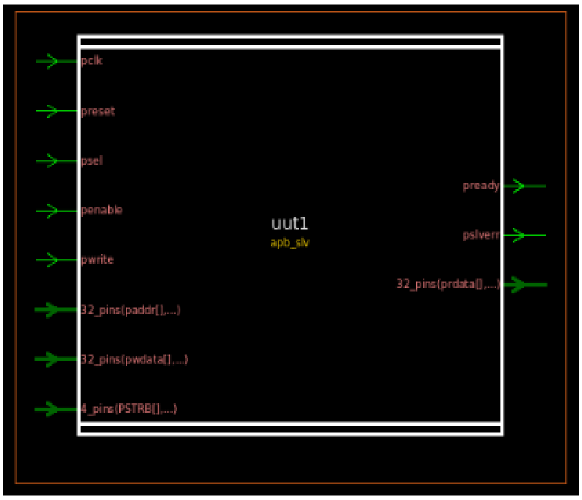


Fig 17: Schematic Diagram of APB Slave

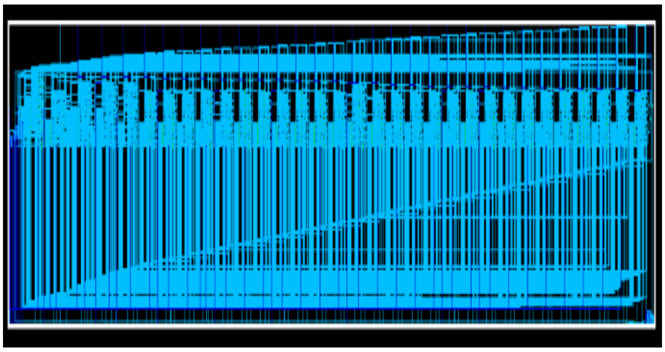


Fig 18: Internal Schematic view of APB Slave

D. Physical design:

For the Physical plan stream we utilized the integrated chip(IC) compiler. Here the standard library32_nm is utilized. The created netlist(which depicts the structure) is made from the union stride. As indicated by the netlist, the cell see produced. Fig19 shows course the APB cells. Steering is only interfacing the different squares in the chip with one another. The squares were just barely set on the chip. The order utilized for the directing is rout_opt. Here we will see the total format of the structure.

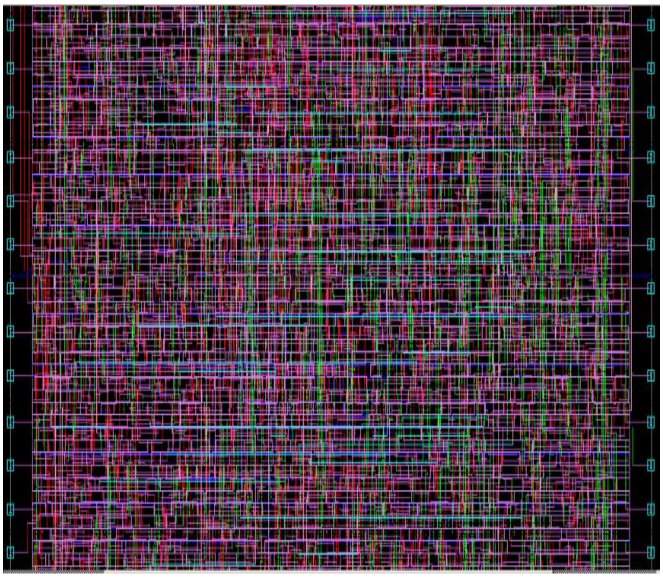


Fig 19: Placement and Routing of APB Master-Slave Interface

VII. CONCLUSION

This paper gives an ASIC design approach for APB. It is developed utilizing the Verilog HDL as indicated by the constraints. The design has dealt with balance between region overhead and speed. The read-write activity is practiced without hold up states and with two clock cycles hold up from the outer ROM and the write activity without zero states and with two clock cycles to the outside RAM and is validated utilizing UVM. The outcomes reveals that the information read from a specific data storage area is same as the information kept in touch with the given data storage area. The UVM report outline additionally guarantees the accuracy of the design and furthermore how the error has been decreased without loss of information while transferring.

REFERENCES

1. ARM, “AMBA Specification (Rev 2.0)” @ http://www.arm.com.
2. Akhilesh Kumar, Richa Sinha, “Design and Verification analysis of APB Protocol with Coverage”, IJAET, Nov 2011.
3. Santhi Priya Sarekokku1, K.Rajasekhar, “Design And Implementation of APB Bridge Based On AMBA AXI 4.0” International Journal of Engineering Research & Technology (IJERT), 2012
4. Shilpa Rao and Arati S. Phadke, “Implementation of AMBA compliant Memory Controller on a FPGA”, IJETEE, 2013.
5. Shankar, Dipti Girdhar, Neeraj Kr. Shukla “Design and Verification of AMBA APB Protocol”
6. ShashisekharRamagundam & others “Design and Implementation of High-Performance Master/Slave Memory Controller with Microcontroller Bus Architecture”, IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings,2014
7. Heli Shah P, Chinmay modi P, Bhargav Tarpara P "Design & Implementation of Advance Peripheral Bus Protocol", International journal of scientific engineering and applied science (IJSEAS) vol.1, no.3, June 2015
8. Kiran J P ,Dr. R Jayagowri, “Abstract, Correct By Construction and Faster Register Modeling of AMBA APB Bus” IARJSET, 2016.
9. Kommirisetti Bheema Raju,Kommirisetti Bheema Raj "Design And Verification of AMBA APB Protocol" International Journal of Engineering Research and Applications, 2017
10. M. Kiran Kumar, Amrita Sajja & Dr. Fazal Noorbasha “Design and FPGA Implementation of AMBA APB Bridge with Clock Skew Minimization Technique” IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), Volume 7, Issue 3, Ver.1 PP 42-45,(May. - June. 2017).
11. Venkat Rao, Pasam Abhigna, J.Naveen, S.Vineeth Reddy "Design of AMBA APB Protocol".IJRASET, 2018.
12. Bevoor, S. and Gowda “Design of high performance master/slave memory controller for grey scale image transfer”. In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) (pp. 611-614). IEEE.
13. KiranRawat, KanikaSahni, SujataPandey, “RTL Implementation for AMBA ASB APB Protocol at System on Chip Level” in 2nd International Conference on Signal Processing & Integrated Networks, SPIN 2015, Amity University, Noida, 19-20 Feb 2015.
15. URL:http://www.testbench.com.
16. https://www.verificationguide.com/p/uvm-tutorial.html
17. https://www.chipverify.com/uvm/uvm-tutorial
18. https://nptel.ac.in/content/storage2/courses/106106089/magma_tutorial/magma_tutorial.html

AUTHORS PROFILE

K. Swetha Reddy is working as an Assistant Professor in VNRVJIET, Hyderabad since 2016 and worked as an intern in seerakademi during 2014-2016. She is currently pursuing Ph.D in VLSI Design from JNTU, Hyderabad. She carried out Project titled “Mil-STD 1553 Bus Protocol” funded by Ananth technologies Pvt. ltd., Hyderabad. Her area of interests include are in Design Verification and Low Power VLSI.

P. Soujanya received B.tech in ECE from Memorial College of Engineering and Technology, Devarakonda, Telangana, India, in 2014 and she is pursuing M.Tech in VLSI system Design at VNR VJIET, Hyderabad, India. Her research interest areas are verification methodologies, low Power VLSI.

D. Kanthi Sudha has completed her B.Tech (ECE) and M.Tech(Microwave Engineering)from Acharya Nagarjuna University, Andhra Pradesh. She has worked as Project Engineer in Microwave Systems Group, RCI, DRDO from 2004-2007. Her work involved Design and Simulation of Radio Proximity fuse used in Astra Missile. As an Assistant Project Engineer in Astra Microwave Products Limited from 2007-2008, she was responsible for building the MMIC library of components in ADS for the company. She has been working as an Assistant Professor in VNRVJIET since 2008 and is currently pursuing her Ph.D. in RF MEMS from JNTU Hyderabad. Her Research areas of interest include RF MEMS, Smart Antennas, and RF Transmit/Receive Modules. She is a Senior Member, IEEE, Life Member ISTE and Life Member IETE. She has mentored a team of students who have emerged as winners in Smart India Hackathon – Hardware Edition organized by MHRD & AICTE, for the problem statement given by Oil India Limited.