An AHB on Chip Bus Tracer with Real Time Compression for SoC Support

B U V Prashanth
Asst.Professor, Dept of ECE
Anurag College of Engineering
Hyderabad, (A.P), India

ABSTRACT
This paper illustrates the system on chip (SoC) debugging and analyses its behavior at several test conditions by verifying the functional aspects of the on-chip bus. Here an Advanced High performance bus (AHB) is selected, since the AHB bus signals are difficult to observe as they are deeply embedded in the system on chip and these I/O pins to access these signals is not possible. Hence we embed a bus tracer in SoC to capture and compress the bus signals. The tracer is successfully verified in FPGA. In this manuscript the selected software is XILINX ISE software design environment for RTL synthesis and model-Sim simulation software to verify the timing diagrams of the AHB SoC modules designed. The FPGA used is SPARTAN 3E (XC3S500E).

Keywords
On-chip bus, AMBA, SoC debugging, COMPRESSION, FPGA

1. INTRODUCTION
The proposed multi resolution AHB on chip Bus tracer is named as SYS-HMRBT. The bus tracer adopts three trace compression mechanisms to attain the high trace compression ratio. Multi resolution tracing is supported by capturing traces at different timing and signal abstraction levels. The dynamic mode change is the add on feature to allow user to switch the resolution for different portions of the trace to match specific debugging/analysis need.

2. RELATED WORK
Since the traced data is very huge that limits the trace memory and there are some hardware methods to compress the traces they are lossy trace compression and lossless trace compression. Maximum compression ratio is achieved by lossy trace compression technique but the accuracy reduced. Anis and Nicolici [5] technique is suitable for repeatable and deterministic systems. But the complex System on chip with different IPs are neither repeatable nor deterministic therefore the appropriate method for on-chip bus tracers is lossless trace compression.

3. EXISTING SYSTEMS
The AMBA navigator[3] that is capable of tracing all the AHB bus signals but not providing any compression support and AMBA AHB trace macro cell(HTM) [2] developed by ARM provides the AHB bus trace with compression techniques and the Data value trace is not supported by HTM, AMBA Navigator, HTM both are having restricted abstraction ability in timing dimension only.

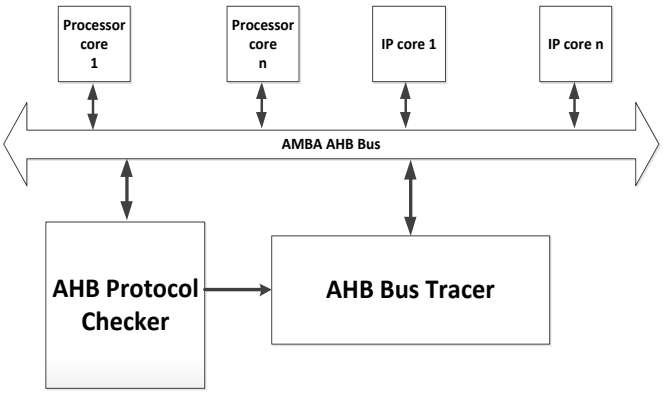

Fig 1: AMBA AHB Bus Tracer Block Diagram

4. PROPOSED SYSTEM
The SYS-HMRBT shown in figure 1 above mainly consists of four modules they are Event Generation, Abstraction, Compression and Packing as seen in figure 2. Firstly the Event generation module decides the starting &stopping of the trace and its trace mode it consists of configurable event registers which specify the triggering events on bus and it also have corresponding matching circuit to compare the bus activities with the other events in the register. Generally, this module can accepts events from external modules like AHB bus Protocol checker (HP Checker)[6].The format of event generation contains four parameters they are trigger conditions, trace mode, trace direction and trace depth among these, the trigger condition can be of any address value which can be the combination of address value, data value and control signal values. There is a mark field for each value to enable partial match. For every triggering condition, we can assign a desire trace mode, which allows to be dynamically switched between events. The trace captured before the target point is reached is referred as backward trace and similarly the trace captured after the target point is reached is referred to as forward trace. Since the initial state of the trace which is currently under compression the real time compression of the backward trace in a circular is difficult to achieve [7].

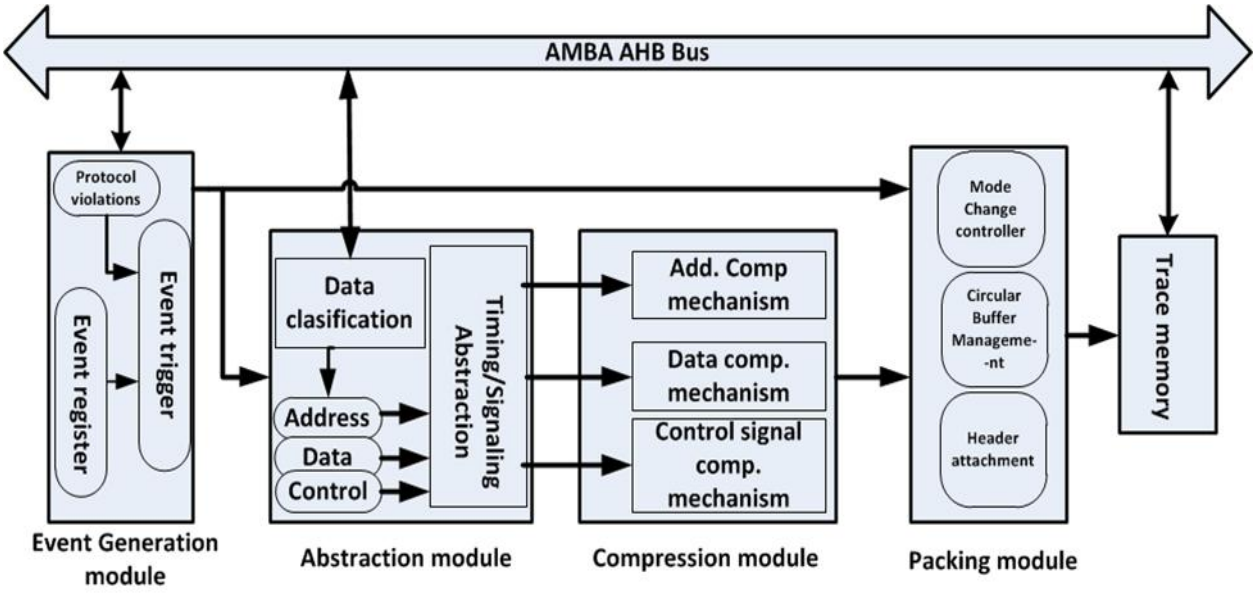

Fig 2: AHB Bus Tracer Block Diagram

The next module after the Event generation module is Abstraction module and the main function of abstraction module is, it monitors the AMBA AHB bus, filters the signals. The abstraction is in two dimensions timing, signal. At the timing dimension cycle level, transaction level two levels of abstractions are there, we can define three levels of abstraction for signal dimension they are bus state, full signal and master operation. The AHB bus signals classified into four categories they are program address, protocol control signals, access control signals and data address/value. All the signals are captured by the full signal level. Bus state level will captures all the signals and encodes the protocol control signals. The bus master transfer activity captured at the master state level. Integrating the abstraction levels in both dimensions we can get five modes they are mode full signal cycle level (FC), mode full signal transaction level (FT), mode bus state cycle level (BC), mode bus state transaction level (BT) and mode master state transaction level(MT). We can dynamically change the trace mode in real-time to analyze the bus trace and also achieves the dynamic mode change feature.

The compression module is to compress the signals to reduce trace size. Here we achieves the compression by using three effective compression mechanisms as shown in fig.2, they are address compression mechanism, data compression mechanism and control signal compression mechanism.

4.1 Address Compression Mechanism

The program addresses can be compressed in three phases by using three compression techniques they are branch/target filtering technique, dictionary based compression technique [4] and slicing technique [3].

4.1.1 Branch/target filtering

Mostly the program address is sequential so the address of the first instruction (target) and last instruction (branch) are recorded and the size is further reduced by dictionary based compression.

4.1.2 Dictionary Based Compression:

In this approach a dictionary stores the frequently appeared target-branch address; we use a comparator to compare data with the previous data in the dictionary if it found that existed then simply stores the index value otherwise it stores data in the dictionary.

4.1.3 Slicing:

This is the last phase in the address compression if there is any missed address in the previous phase can be compressed by slicing approach. In the slicing approach the address are divided in the form of slices with equal size, in this approach we use a register and a slice comparator the register is used to store the previous address and the slice comparator compares the present data with the previous data. If the data differs with the previous data in one slice then only that slice is recorded remaining are ignored for example x = 0000 8066 is the previous data and y = 0000 8020 is the present, after slicing only 20 is recorded for y

4.2 Data Compression Mechanism

The data address/value mostly random and irregular. We propose difference method based on subtraction, in this method the present data subtracted from the previous one and also removes the leading zeros and then stored, if the difference is greater than 65535 then it records present 32-bit data value.

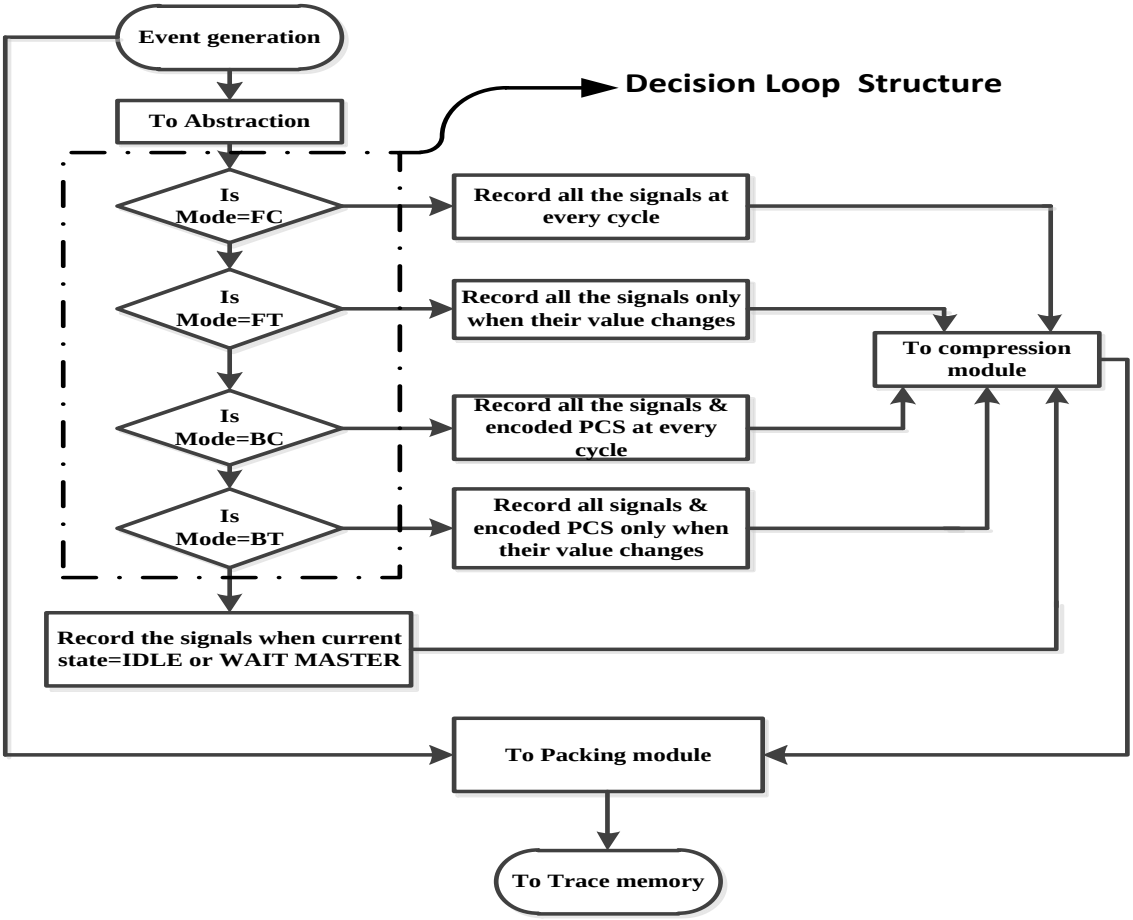

Fig 3: Tracer Implementation Flow

4.3 Control Signal Compression Mechanism

The AMBA AHB Control signals are classified in to two groups one is access control signals and another is protocol control signals. In these some of the signals are frequently repeat and some signals does not occur or occur rarely so that we choose the Dictionary Based Compression technique.

5. FLOW CHART

As we illustrated in the flow diagram in figure 3 the functionalities of the four modules in the proposed system now the implementation flow of the tracer, in this the AHB protocol checker will gives protocol which describes the errors caused by protocol violation to the event trigger and also it has event registers with these the Event trigger triggers the starting &stopping of the trace and its trace mode and the abstraction module has five modes in which if the designer selects Mode FC then the tracer records all the signals at every cycle, if it is Mode FT then the tracer records all the signals only when their value changes, if the selected mode is BC then it records all the signals and encodes the protocol control signals at every cycle and the abstraction module sends this information to the compression module to reduce the traced data size. The compressed data finally packed with proper headers then it is written to trace memory by packing module.

6. ALGORITHM DESIGN

The algorithm according to the flow chart as illustrated above is as follows

a. Start
b. Initialize the event generation module
c. Scale down to abstraction
d. Enter into the decision loop structure
e. Record the signals when the current state is IDLE or WAIT MASTER
f. If the condition in decision loop structure is true then
    i. Record all signals at every cycle
    ii. Record all signals only when their value changes
    iii. Record all the signals and encode Protocol Control Signal at every cycle
    iv. Record all the signals and encode Protocol Control Signal only when their value changes.
g. Send all the above true values into the compression modules.
h. Perform the compression
i. Obtain traced and compressed data
j. Send the data to the packing module

7. RESULTS OBTAINED

After performing the compression the traced and compressed data is sent to the Packing module here the header attachment is done by providing the mode change information and circular buffer management and writes it to the trace memory. The output results obtained are as elucidated in figures 4 to 7.

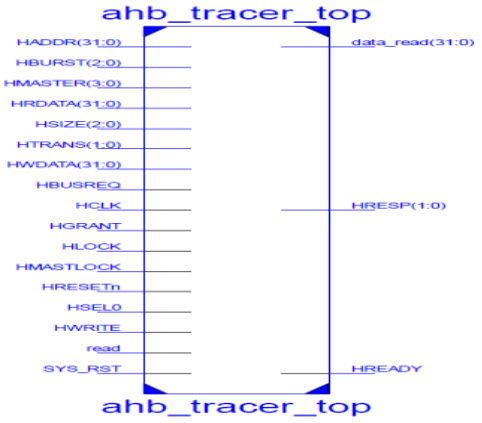

Fig 4.AHB Tracer top module

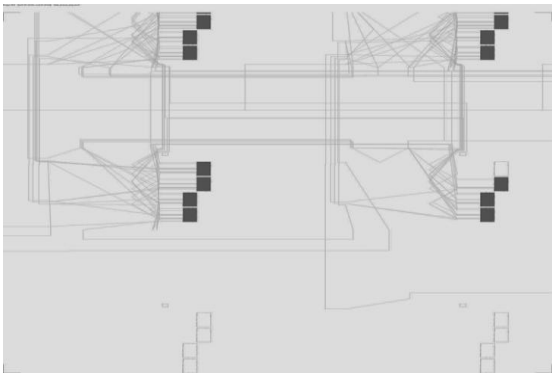

Fig 5. Placement of AHB Tracer

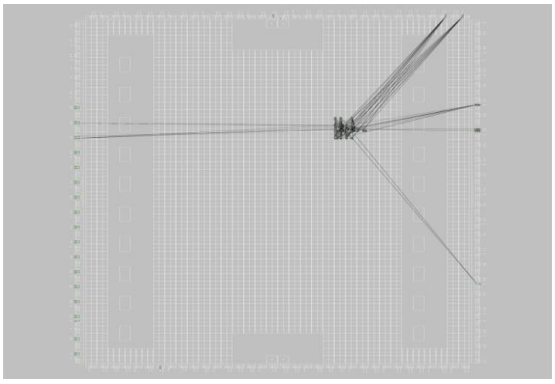

Fig 6. Routing diagram of AHB Tracer

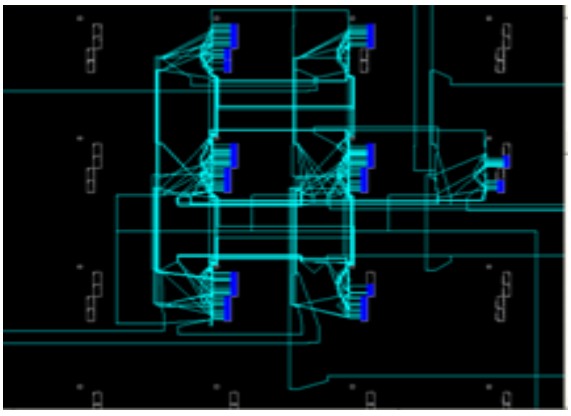

Fig 7. Complete Placement and Routing of AHB Tracer

8. CONCLUSION

We have presented the system on chip (SoC) debugging and analyze its behavior at several test conditions by verifying the functional aspects of the on-chip bus. To capture and compress the bus tracer we embed a bus tracer in System on Chip (SoC). The tracer is successfully verified on FPGA, The FPGA here we selected is XC3S500E SPARTAN 3E and also we obtained the complete placement and routing of our tracer and also the simulation results are obtained that are matching with theoretical calculations. This will shows our tracer achieves a good compression ratio ranging. In addition, the proposed arbiter selects the best possible arbitration schemes based on the priority-level notifications and the desired transfer length from the masters to allow the arbitration to lead to the maximum performance. Experimental results show that, although the area of the proposed arbitration scheme is larger than those of other arbitration schemes, our arbiter improves the throughput compared with other schemes. We therefore expect that it would be better to apply our arbitration scheme to an application specific system because it is easy to tune the arbitration scheme according to the features of the target system. For future scope of work, it is felt that the configuration of the arbitration scheme with the maximum throughput needs to be found automatically during run time.

9. REFERENCES

[1] ARM Ltd., San Jose, CA, "AMBA Specification (REV 2.0) ARM IHI0011A"

[2] ARM Ltd., San Jose, CA, "AMBA AHB trace macrocell (HTM) technical reference manual"

[3] First Silicon Solutions (FS2) Inc., Sunnyvale, CA, "AMBA navigator spec sheet"

[4] Heikkinen J and Takala J, "Programmability in Dictionary based compression" *IEEE symposium on* System on chip 2006.

[5] E. Anis and N. Nicolici, "Low cost debug architecture using lossy compression for silicon debug," in *Proc. IEEE Des., Autom. Test Eur.Conf.*,Apr. 16–20, 2007

[6] Y.-T. Lin, C.C.Wang and I.-J. Huang, "AMBA AHB bus protocol checker with efficient debugging mechanism," in *Proc. IEEE Int. Symp.Circuits Syst.*, Seattle, WA, May 18–21, 2008, pp. 928–931.

[7] Chung-Fu Kao , Ing-Jer Huang , Chi-Hung Lin, An embedded multi-resolution AMBA trace analyzer for microprocessor-based SoC integration, Proceedings of the 44th annual Design Automation Conference, June 04-08, 2007, San Diego, California [doi>10.1145/1278480.1278604]