# CORE DEVELOPMENT OF AMBA AHB BUS TRACER USING FPGA

***SANJANA***

*[1]Research Scholar, ECE Dept, St.Mary's Group of Institutions, Hyderabad, AP-India*

*E-Mail id: sanjana1613@gmail.com*

***ABSTRACT--****This paper proposes an Ip core development of AMBA AHB on-chip bus tracer for versatile system-on-chip (SoC) debugging and monitoring. The bus trace with different AHB signals, with efficient built-in compression mechanisms for diverse range of needs. It allows users to switch the trace signals dynamically so that appropriate signal levels can be applied to different segments of the trace. It mainly contains four parts: Compression Modules, Event Generation Module, Packing Module and Abstraction Module. Event Generation Module controls the start/stop time, the trace mode, and the trace depth. This is sent to the following modules: based on the trace mode, Abstraction Module abstracts the signals in both timing and signal dimension.*

## 1. INTRODUCTION

Over the years, System-on-Chip (SoC) designs have evolved from fairly simple uni-processor, single-memory designs to massively complex multiprocessor systems with several on-chip memories, standard peripherals and ASIC blocks. As more components are integrated into these designs to share the ever increasing load, which increase the communication. Inter-component communication is often in the critical path of a SoC design and is a very common source of performance bottlenecks.

Silicon densities, for both ASICs and FPGAs, now support true systems-on-a-chip (SoCs). This level of design requires busing systems to connect various components, including one or more peripherals, microprocessors, special logic and memory. The Advanced Microprocessor Bus Architecture (AMBA) is ARM's on-chip busing solution.

ARM processors have several unique advantages over traditional microprocessor solutions in terms of performance, small die size, and extremely low power consumption. ARM provides developers with intellectual property (IP) in the form of SoC designs, application-specific standard products and processor core designs, cache, development tools and related software.

Bus protocol play an important role in the field of intellectual property reuse by standardizing the interface of hardware components, they simplify the task of transplanting a module from one system to another. The bus is the mechanism by which a processing element communicates with other processing elements, with memory and with devices. A bus is, at minimum a collection of wires, which also defines a rotocol by processing elements, memories and devices communicate.

THE ON-CHIP bus is an important system-on-chip (SoC) infrastructure that connects hardware components. Monitoring the on-chip bus signal is crucial to the SoC debugging and performance optimization/ analysis. Such signals are difficult to observe since they are deeply embedded in a SoC and there are often no sufficient I/O pins for accessing these signals. Hence, a straightforward approach is to embed a bus tracer in SoC to capture the bus signal trace and store the trace in on-chip storage such as the trace memory which could then be off loaded to outside world (the trace analyzer software) for analysis.

Unfortunately, the size of the bus trace increases rapidly. For example, to capture AMBA AHB 2.0 bus signals running at 200 MHz, the trace rises from 2 to 3 GB/s. Therefore, it is highly desirable to compress the trace on the fly in order to reduce the trace size. Even, simply capturing / compressing bus signal is not sufficient for SoC analysis and debugging since these are versatile: some designers need all signals at cycle-level; others care only about the transactions. Tracing all signals at cycle-level wastes a lot of memory. Hence, there should be a way to capture traces at different abstraction levels based on the specific debugging/analysis need.

This paper presents a real-time multi-Ip core development of AHB on-chip bus tracer, the bus tracer have three trace compression mechanisms to achieve high trace compression ratio. It support *multiresolution tracing* by capturing traces at different signal and timing abstraction levels. It provides the *dynamic mode change* feature to allow users to switch the resolution on-the-fly for different portions of the trace to match specific analysis/debugging needs. Given a trace memory of fixed size, the user can trade off between the granularities and trace length to make the most use of the trace memory. The bus tracer is capable of tracing signals before/after the event triggering. This feature provides a more flexible tracing to focus on the interesting points.

**AHB Master**

AHB (Advanced High-performance Bus) is the latest generation AMBA (Advanced Microcontroller Bus Architecture) bus. It is intended to address the requirements of high performance synthesizable designs. Many system-on-a-chip designs in the portable electronics, telecommunications, and embedded systems markets use AHB to interface application specific design blocks with standard microcontrollers. It is a high-performance system bus that supports multiple bus masters and provides high-bandwidth operation.

An AHB bus master has the most complex bus interface in AMBA system. Typically an AMBA system designer would use predesigned bus masters and therefore would not need to be concerned with the detail of the bus master interface.

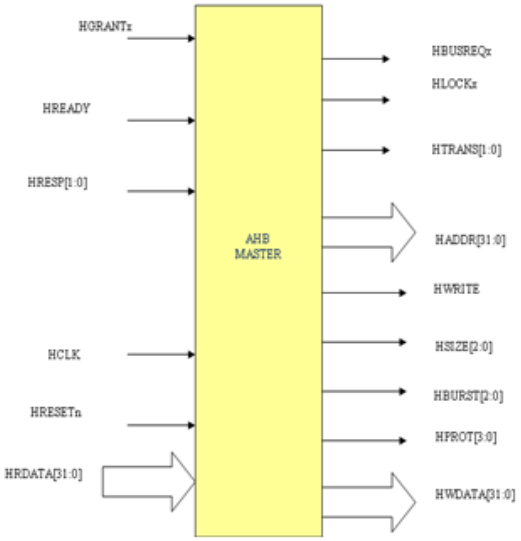


**Fig 1: AMBA AHB Master**

## 2. AMBA AHB TRACER

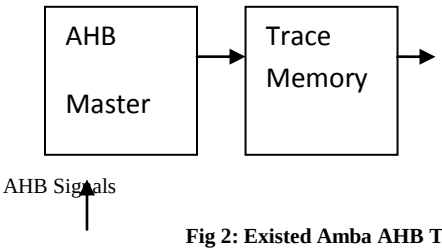


**Fig 2: Existed Amba AHB Tracer**

It mainly contains four parts: Event Generation Module, Packing Module, Compression Modules, and Abstraction Module. Event Generation Module controls the trace mode, start/stop time, and the trace depth. This information is sent to the following modules.

Based on the trace mode, Abstraction Module abstracts the signals in both timing and signal dimension. This abstracted data is further compressed by the Compression Module to reduce the data size. These compressed results are packed with proper headers and written to the trace memory by the Packing Module.

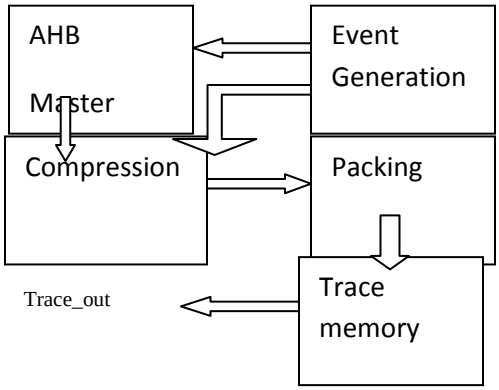


**Fig 3: proposed System Amba AHB Tracer**

***Event Generation Module:***

*The Event Generation Module* decides the starting and stopping of a trace and its mode. The module has configurable event registers which specify the triggering events on the bus and a corresponding matching circuit to compare the bus activity with the events specified in the event register

***Compression Module:***

The purpose of the Compression Module is to reduce the trace size. This accepts the signals from abstraction module. To achieve real time compression, the Compression Module is pipelined to increase the performance. *Dictionary-Based to reduce the* size, we use temporal locality. Temporal locality exists since the basic blocks repeat frequently (loop structure), which implies the addresses of the Location. We can use the dictionary- based compression, where the idea is to map the data to a table, by placing frequently appeared data, and record into the table index instead of the data to reduce size of tracing memory. The dictionary keeps the frequently appeared target/branch addresses. For keeping the hardware cost reasonable, proposed dictionary is implemented using a ROM based Memory.

***Packing***

The compressed results are packed with proper headers and written to the trace memory by the Packing Module. It is the last phase. It receives compressed data from the compression module, processes, and writes to the trace memory. It is having three jobs: circular buffer management, mode change control and packet management. For packet management, the compressed data length and type are variable; every compressed data require a header for interpretation. This generates a proper header and is attached to each compressed datum, which is called as a packet in this paper. As the header generation takes long cycle time, to avoid this header generation is implemented in one pipeline stage.

***Trace Memory***

The tracing packet comes out from the packet module stored into the Trace memory. This trace memory we can design using FIFOs. The circular buffer management manages the accesses to the trace memory. The size of a packet is variable but the data width of the trace memory is fixed, it collects the trace data in a first-input, first-output (FIFO) buffer and outputs them to the trace memory until the data size in the FIFO buffer is equal/larger than the data width as the tracing stops. The data size in the FIFO buffer is smaller than data width; one more cycle is required to output the remaining data to the trace memory.

## 3. SIMULATION RESULTS

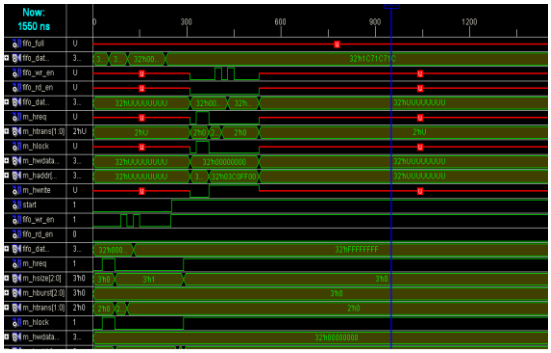

**Fig 4: Simulation Results of AHB Tracer without compression**.

## 4. CONCLUSION

Tracing of Amba AHB Bus successfully developed using VHDL. Compression Technique is successfully implemented to reduce the storage of Trace Memory size. And Total design functionally verified using ISE simulator and Synthesized by Xilinx 9.1i.

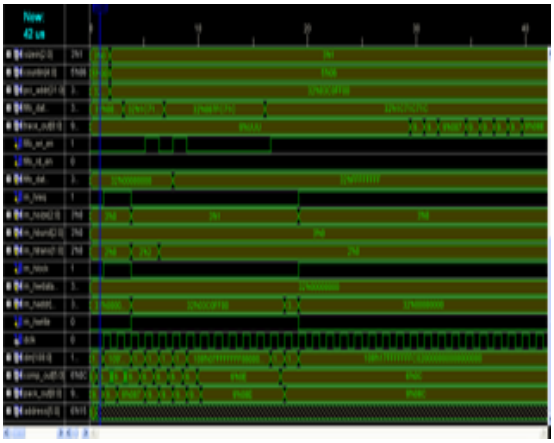

**Fig.5: simulation results of AMBA AHB Tracer with compression**

The Fig 4. Shows the AMBA AHB Tracer without compression. Fig 5. Shows the AMBA AHB Tracer with compression Technique. The Design report of Tracer of Spartan 3e FPGA showed in Fig.6

| | | | |
|---|---|---|---|
| Project File: | pt11vl076.ise | Current State: | Synthesized |
| Module Name: | top_tracer | • Errors: | No Errors |
| Target Device: | xc3s250e-4tq144 | • Warnings: | 13 Warnings |
| Product Version: | ISE 9.1i | • Updated: | Tue Sep 4 02:34:52 2012 |

| PT11VL076 Partition Summary |
|---|
| No partition information was found. |

| Device Utilization Summary (estimated values) | | | |
|---|---|---|---|
| Logic Utilization | Used | Available | Utilization |
| Number of Slices | 283 | 2448 | 11% |
| Number of Slice Flip Flops | 230 | 4896 | 4% |
| Number of 4 input LUTs | 502 | 4896 | 10% |
| Number of bonded IOBs | 122 | 108 | 112% |
| Number of BRAMs | 1 | 12 | 8% |
| Number of GCLKs | 2 | 24 | 8% |

**Fig 6: Design summary report for AHB Tracer**

## 5. REFERENCE

*[1]A Multi-resolution AHB Bus Tracer for Real-time Compression of Forward/Backward Traces in a Circular Buffer Yi-Ting Lin, Wen-Chi Shiue, and Ing-Jer Huang Department of Computer Science and Engineering National Sun Yat-Sen University Kaohsiung Taiwan 804.*

*[2]An On-Chip AHB Bus Tracer With Real-Time Compression and Dynamic Multiresolution Supports for SoC Fu-Ching Yang, Member, Yi-Ting Lin, Chung-Fu Kao, and Ing-Jer Huang, Member,IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, APRIL 2011*

*[3] ARM. Example AMBA SYstem User Guide ARM DUI0092C, Aug. 1999.*

*[4] ARM. Embedded Trace Macrocell Architecture Specification, Feb. 2006.*

*[5] ARM. AMBA AHB Trace Macrocell (HTM) Technical Reference Manual, 2007.*

*[6] First Silicon Solutions. Preliminary Technical Data for AMBA Navigator AMBA On-Chip Bus Analyzer for AHB Bus Systems.*

*[7] J. Gaisler, M. Isomaki, E. Catovic, K. Glembo, and S. Habinc. GRLIB IP Core User's Manual Research.*

*[8] R.-T. Gu, T.-C. Yeh, T.-Y. Huang, W.-S. Hunag, C.-H. Tsai, C.-N. Lee, Y.-N. Chang, M.-C. Chiang, S.-F. Hsiao, and I.-J. Huang. A low cost tile-based 3D graphics pipeline with real-time performance monitor support for OpenGL ES in consumer electronics 2007.*

*[9] W.-J. Huang, N. Saxena, and E. J. McCluskey. A reliable LZ data compressor on reconfigurable coprocessors. In IEEE Symposium on Field-Programmable Custom Computing Machines, 2000.*

*[10] Infineon Technologies. TC1775 TriCore User's Manual System Units section 20, on-chip debug support, Feb. 2001.*

*[11] E. E. Johnson, J. Ha, and M. B. Zaidi. Lossless trace compression. IEEE Trans. Comput., 50:158–173, Feb. 2001*