



# An advanced tracing mechanism for optimum debugging support for soc's

S.Thripurna<sup>1</sup>, Ms.G.Renuka<sup>2</sup>, Dr.Syed mushtak Ahmed<sup>3</sup>

PG Student, Dept. of Electronics and Communication Engineering, SR Engineering College, India<sup>1</sup>

Assistant Professor, PG Student, Dept. of Electronics and Communication Engineering, SR Engineering College, India<sup>2</sup>

Professor & H.O.D, Dept. of Electronics and Communication Engineering, SR Engineering College, India<sup>3</sup>

**Abstract:** AMBA (Advanced Microcontroller based Bus Architecture) consists of AHB, APB, ASB and AXI. In this project we are Tracing AHB (Advanced High performance Bus) signals with Real time Compression and Multiresolution Techniques. A simple transaction on the AHB consists of an address phase and a subsequent data phase. Access to the target device is controlled through a MUX, thereby admitting bus-access to one bus-master at a time. In AHB Tracer we have to Trace Address signals, Data signals and Control signals the have to compress them depending on AHB protocols. A multiresolution AHB on-chip bus tracer is named as SYS\_HMRBT (AHB Multiresolution Bus Tracer) and is used monitoring. By using this SYS\_HMRBT, we can achieve 79%-96% of compression depending on selected resolution mode.

**Keywords:** AHB, AMBA, compression, multiresolution, post-T trace, real time trace.

## INTRODUCTION

In the present scenario With the deep submicron process technology matures, IC chip scale is more and more big. Digital IC based on the timing of the design method, driven to the design method based on IP reuse, and in the SOC design can be widely used. Based on IP reuse in the SoC design, chip bus design is the most critical problem. Therefore, the industries there are many chip bus standards. Among them, the company launched by ARM on-chip AMBA was broad IP bus aggregator of developers and the favor of SoC system, has become a popular industry standard chip structure. AMBA specification mainly includes AHB (on the High performance Bus) system Bus and APB (Peripheral Bus) on the periphery Bus

AHB mainly used for high-performance module (such as CPU, the DMA and DSP, etc.), as the connection between the SoC framework10 chip system bus, it includes the following some properties: a single clock edges operating; The three states realization ways; Support sudden transmission; Support subsection transmission; Support multiple main controller; Configurable 32-bit ~ 128-bit bus width; Support bytes, half bytes and word transmission. From the main module, AHB system from module and Infrastructure (Infrastructure) 3 parts, the

whole AHB bus by the main module transfers from module issued, responsible for the response.

An AMBA-based microcontroller typically consists of a high-performance system backbone bus, able to sustain the external memory bandwidth, on which the CPU and other Direct Memory Access (DMA) devices reside, plus a bridge to a narrower APB bus on which the lower bandwidth peripheral devices are located. Figure1 shows both AHB and APB in a typical AMBA system.

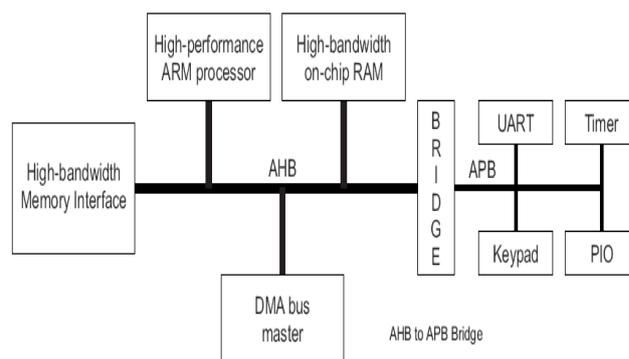


Figure1: A typical AMBA AHB-based system

Most hang on a bus module (including processors) just a single attribute function modules: main modules, or from module. The main module is to send the operation from module reading, such as CPU, DSP module; from module is to accept orders and respond module, such as chip RAM, AHB/APB Bridge, etc. In addition, some module and has two properties, such as direct memory access



(DMA) were from module, but when programming in system must read data transmission is main modules. If the bus exist multiple main modules, they need to decide how to control the arbitration device module of main kinds of bus access. Although arbitration standard is part of the bus, regulating AMBA but specific use a algorithm Engineers decided by RTL design, two of them the most commonly used algorithm is fixed priority algorithm and round robin algorithm. Basic structure is composed of arbitration device (arbiter), the main module to the many way from module, from the main module into how way, decoder (decoder), virtual from module (dummy Slave), virtual main module (dummy Master) together.

Maximum AHB bus has 16 main modules and multiple from module, if the main module number more than 16, it is required to add a layer of structure (specific refer to ARM the company launched the Multi – layer AHB standard). APB bus APB bridge is the only, but also the main module AHB system bus from module. Its main function is to latch AHB system bus from the address, data and control signals, and provide secondary decoding to produce APB peripherals selection signal, so as to realize the APB agreement AHB agreements to convert.

**II. DESIGN AND IMPLEMENTATION**

Figure3 is the bus tracer overview. It mainly contains four parts: Event Generation Module, Abstraction Module, Compression Modules, and Packing Module. Event Generation Module controls the start/stop time, the trace mode, and the trace depth of traces. This information is sent to the following modules. Based on the trace mode, the Abstraction Module abstracts the signals in both timing dimension and signal dimension. The abstracted data are further compressed by the Compression Module to reduce the data size. Finally, the compressed results are packed with proper headers and written to the trace memory by the Packing Module.

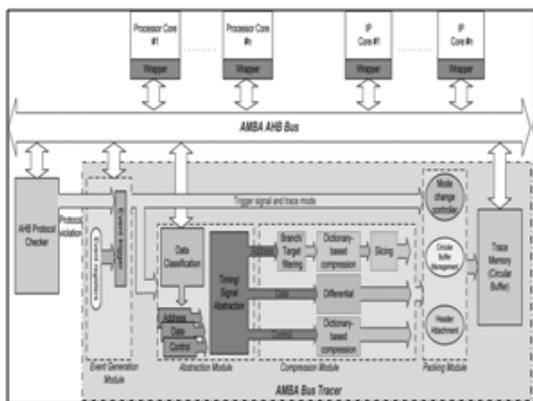


Figure2: Multiresolution Bus Tracer Block Diagram

**A. Event Generation Module:**

The Event Generation Module decides the starting and stopping of a trace and its trace 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 registers.

**B. Abstraction Module:**

The Abstraction Module monitors the AMBA bus and selects/filters signals based on the abstraction mode. The abstraction level is in two dimensions: timing abstraction and signal abstraction. At the timing dimension, it has two abstraction levels, which are the cycle level and transaction level.

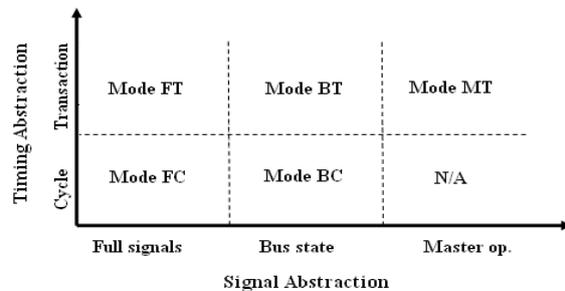


Figure3; Multiresolution trace modes

Combining the abstraction levels in the timing dimension and the signal dimension, we provide five modes in different granularities, as Figure4.4 shows. They are Mode FC (full signal, cycle level), Mode FT (full signal, transaction level), Mode BC (bus state, cycle level), Mode BT (bus state, transaction level), and Mode MT (master state, transaction level).

**C. Compression Module:**

The purpose of the Compression Module is to reduce the trace size. It accepts the signals from the abstraction module. To achieve real-time compression, the Compression Module is pipelined to increase the performance. Every signal type has an appropriate compression method. Although the Abstraction Module can reduce the trace size, the remaining trace volume is still very large. To reduce the size, the data compression approaches are necessary. Since the signal characteristics of the address value, the data value, and the control signals are quite different, we propose different compression approaches for them.



1. Program Address Compression: We divide the program address compression into three phases for the spatial locality and the temporal locality. Figure4 shows the compression flow. There are two approaches: branch/target filter, dictionary-based compression.

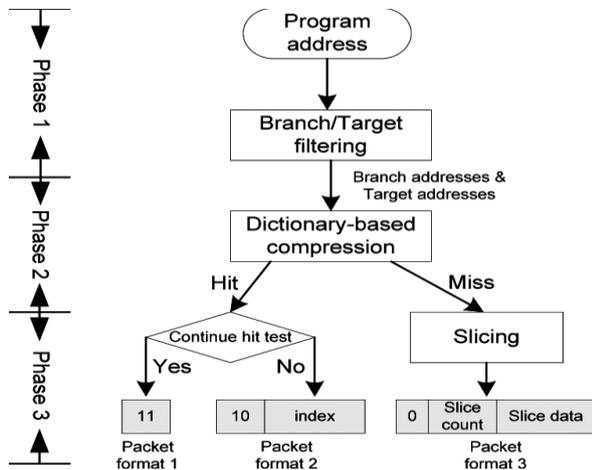


Figure4: Program address compression flow and trace format.

Branch/target filter technique aims at the spatial locality of the program address. Spatial locality exists since the program addresses are sequential mostly. Software programs (in assembly level) are composed by a number of basic blocks and the instructions in each basic block are sequential. Because of these characteristics, Branch/target filtering can records only the first instruction's address (Target) and the last instruction's address (Branch) of a basic block. The rest of the instructions are filtered since they are sequential and predictable. The state diagram for the Branch-Target Filtering is given in Figure5

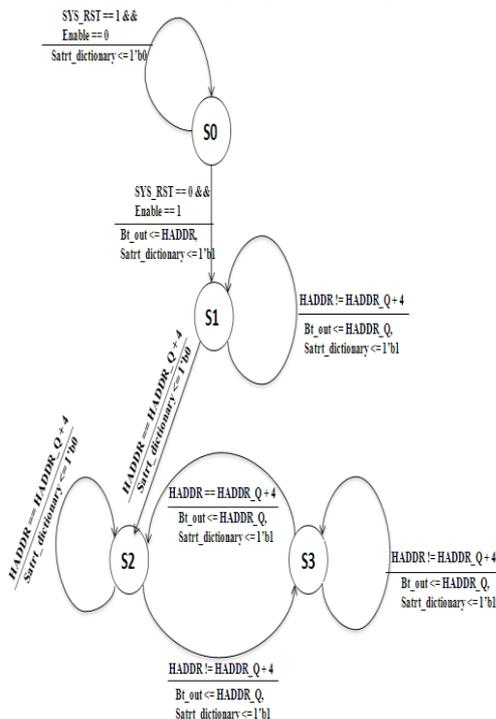


Figure5: State Diagram for Branch/Target Filtering

Dictionary based compression circuit further reduce the size, we take the advantage of the temporal locality. Temporal locality exists since the basic blocks repeat frequently (loop structure), which implies the branch and target addresses after Phase 1 repeat frequently. Therefore, we can use the dictionary-based compression. The idea is to map the data to a table keeping frequently appeared data, and record the table index instead of the data to reduce size. Figure6 shows the hardware architecture. The dictionary keeps the frequently appeared branch/target addresses. To keep the hardware cost reasonable, the proposed dictionary is implemented with a CAM-based FIFO. When it is full, the new address will replace the address at the first entry of FIFO. For each input datum ( $din_i$ ), the comparator compares the datum with the data in the dictionary (Table []). If the datum is not in the table (match = Miss), the datum (uncompressed data) is written into the table and also recorded in a trace. Otherwise (match = Hit), the index (match index) of the hit table entry is recorded instead of the datum.

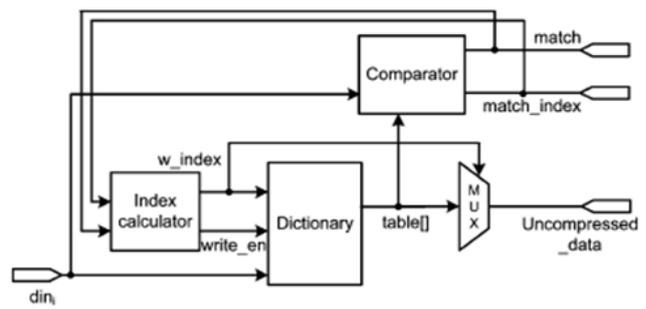


Figure6: Block diagram of the dictionary-based compression circuit.

2. Data Address/Value Compression:

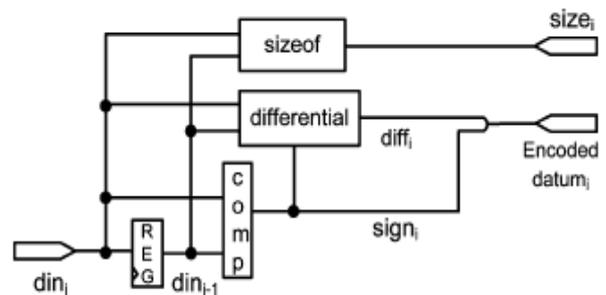


Figure7: Block diagram of differential compression circuit

Data address and data value tends to be irregular and random. Therefore, there is no effective compression approach for data address/value. Considering using minimal hardware resources to achieve a good compression ratio, we use a differential approach based on





data width. If the tracing stops and the data size in the FIFO buffer is smaller than the data width, one additional cycle is required to output the remaining data to the trace memory.

### III. EXPERIMENTAL RESULTS

#### A. Address Compression Result

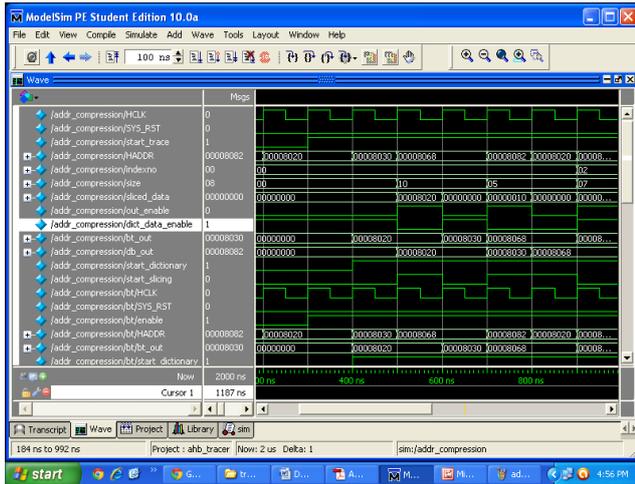


Figure 9: Address Compression Simulation Result

#### B. Data Compression Result

Figure10: Data Compression Simulation Result

#### C. Control Compression Result

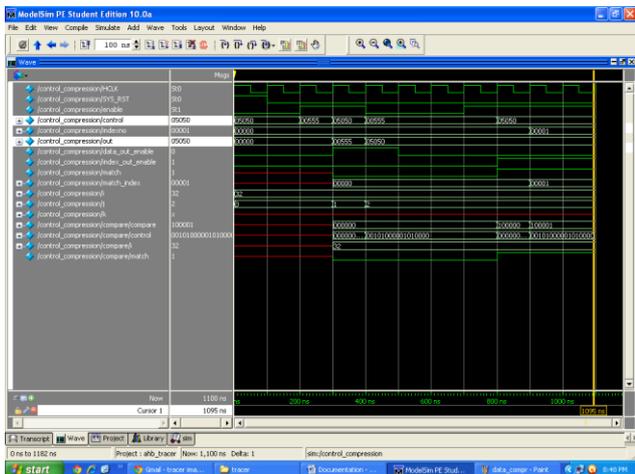


Figure11: Control Compression Simulation Result

#### D. Trace Memory Report

```
# 2500 trace_memory[ 1] 0000a000
# 2600 trace_memory[ 2] 0a0c0100
# 2700 trace_memory[ 3] 00060100
# 2900 trace_memory[ 4] 28300402
# 3000 trace_memory[ 5] 81c02000
# 3100 trace_memory[ 6] 00038080
# 3200 trace_memory[ 7] 141c0201
# 3300 trace_memory[ 8] 42e0100a
# 3400 trace_memory[ 9] 17008050
# 3500 trace_memory[ 9] 17008050
# 4400 trace_memory[ 11] 00200028
# 4500 trace_memory[ 12] 5c014000
# 4600 trace_memory[ 13] 2011ad15
# 4700 trace_memory[ 14] 78080288
# 4800 trace_memory[ 15] c0401440
# 4900 trace_memory[ 16] 0200a200
# 5000 trace_memory[ 17] f000a20
# 5100 trace_memory[ 18] 01005100
# 5200 trace_memory[ 19] 08028800
# 5400 trace_memory[ 19] 08028800
# 5600 trace_memory[ 21] 7008000a
```

Trace Memory Result

### IV. CONCLUSION

We have presented an on-chip bus tracer SYS-HMRBT for the development, integration, debugging, monitoring, and tuning of AHB-based SoC's. It is attached to the on-chip AHB bus and is capable of capturing and compressing in real time the bus traces with five modes of resolution. These modes could be dynamically switched while tracing. The bus tracer also supports both directions of traces: pre-T trace (trace before the triggering event)



and post-T trace (trace after the triggering event). In addition, a graphical user interface, running on a host PC, has been developed to configure the bus tracer and analyze the captured traces. With the aforementioned features, SYS-HMRBT supports a diverse range of design/debugging/ monitoring activities, including module development, chip integration, hardware/software integration and debugging, system behavior monitoring, system performance/power analysis and optimization, etc. The users are allowed to tradeoff between trace granularity and trace depth in order to make the most use of the on-chip trace memory or I/O pins.

In the future, we would extend this work to more advanced buses/connects such as AXI or OCP. In addition, with its real time abstraction capability, we would like to explore the possibility of bridging our bus tracer with ESL design methodology for advanced hardware/software co development/debugging/ monitoring/analysis, etc.

## REFERENCES

- [1]Infineon Technologies, Milipitas, CA, "TC1775 TriCore users manual system units"
- [2]ARM Ltd., San Jose, CA, "Embedded trace macrocell architecture specification"
- [3]ARM Ltd., San Jose, CA, "AMBA Specification (REV 2.0) ARM IHI0011A"
- [4]First Silicon Solutions (FS2) Inc., Sunnyvale, CA, "AMBA navigator spec sheet"
- [5] B. Tabara and K. Hashmi, "Transaction-level modeling and debug of SoCs," presented at the IP SoC Conf., France,Naveen Verma, A.P. Chandrakasan.
- [6]YANG et al.: On-Chip AHB Bus Tracer with Real-Time Compression.