Multi Resolution AHB Tracer With Real Time Compression for SoC

[1]*Dr. Shaik Meeravali,* [2] *Praveena,*

[1]*. Professor, Department of Electronics and communication Engineering,RRS College of Engineering and Technology, Muthangi, Jawaharlal Technological University, Hyderabad, India.*

[2]*. .Master Of Scholar, Department of Electronics and communication Engineering, RRS College of Engineering and Technology, Muthangi,JNTUH,Hyderabad,India.*

Abstract— The bus tracing is used to catch related signals for further investigation and analysis. However, the trace size of cycle accurate tracing is large and the trace cycle is shallow unless using a proper compression mechanism. In this paper, we propose an embedded multi-resolution AMBA trace analyzerthat provides the trade-off between the trace granularityand the trace depth. It consists of two major trace approaches:(1) the signal monitor/tracing which provides thelevels of abstraction, and (2) the trace reduction. In the firstapproach, it allows designers to zoom-in/out to preferred level of abstraction to serve different debugging purposes.In the second approach, the trace analyzer compresses the traced data according to signal characteristics and the cost of on-chip storage is reduced. The trace data will be decompressed on the host for further observation and debugging.The experimental results show that the proposed approach can reach a good compression ratio of 96% and the trace depth is more than two thousand cycles at the higher abstraction level.

Keywords — AHB, compression, multi-resolution, tracing..

1. INTRODUCTION

In the System-on-a-Chip (SoC) era, it is a challenge to verify and debug system chip efficiently and rapidly. For design verification and debugging at system level and chip level, not only external I/O signals observation, but also internal signals tracing can help designer to efficiently analyze and verify the design such as the software program, hardware protocol, and system performance. SoC bus signal tracing can be accomplished with either software or hardware approaches. The software approach trace only program address, and the cost of hardware implementation of software approaches would be high.On the other hand, the hardware approach can trace signals at the target system directly. However, the main problem with hardware-based tracing techniques is that the cyclebased traces size is usually extremely large.

To address the bus tracing issue, we propose an embedded multi-resolution signal tracing for AMBA AHB. The bus tracer consists of two major tracing approaches: (1)signal monitor/tracing approach, and (2) trace reduction approach. In the first approach, it provides the trade-off between the trace accuracy and the trace depth. Designers can decide to trace AHB signals in detail or in rough depend on debugging objectives; moreover, the trace granularities can be changed while tracing is processed. In other words, different trace strategy results can be stored in a single trace file. In the second approach, the bus tracer compresses the trace data according to AHB signal characteristics such as address, data, and control signals. The trace data will be decompressed on the host, translated into VCD (Value Change Dump) [1] format, and displayed on a waveform viewer. The bus tracer is integrated into an ARM EASY (Example AMBA SYstem) [2] [3] environment with a 3D graphic hardware acceleration system to demonstrate

2. RELATEDWORK

The spirit of a hardware tracer is its data reduction or compression techniques. For program address tracing, an intuitive way is to discard the continuous instruction addresses and retain only the discontinuous ones, such as the addresses of branching and target instructions, with some

hardware filters. This approach has been used in some commercial processors, such as TriCore [4] [5], and ARM's Embedded Trace Macrocell [6]. The hardware overhead of these works is small since the filtering mechanism is simple to implement in hardware. However, the effectiveness of these techniques is mainly limited by the average basic block size, which is roughly around four or five instructions per basic block, as reported in [7] and [8]. For data address and value tracing, the most popular method is used the differential approach based on subtraction. Some researches have shown that using the differential method can reduce the data address and data values traces by about 40 percent and 14 percent respectively [9] [10]. Besides the address and data bus, there are several control signals on system bus that need to be traced. Some FPGA boards have built-in signal trace tools, such as the Altera SignalTap [11] and Xilinx Chip- Scope [12]. FS2 AMBA Navigator [13] supports bus clock mode and bus transfer mode to trace bus signals on every clock and bus transfer respectively. Trace buffer stores bus cycles or bus transfers based on local internal memory size. Although these approaches support multiple trace modes such as tracing at cycle-by-cycle or at signal transaction, only one mode can use during a tracing process. This paper presents the multi-resolution approach that can use different trace modes during a bus signal tracing process.

3. BUS TRACER ARCHITECTURE

This section presents the architecture of our bus tracer. Fig. is the bus tracer overview. It mainly contains four parts: Event Generation Module, Abstraction Module, Compression Modules, and Packing Module. The Event Generation Module controls the start/stop time, the trace mode, and thetrace 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.

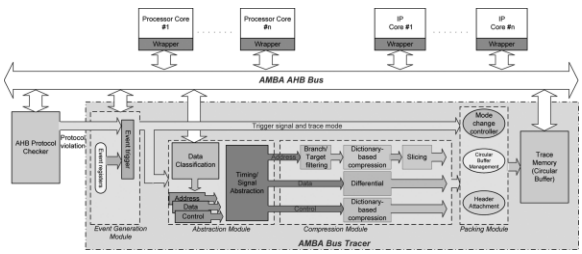

Fig.1Multiresolution bus tracer block diagram.

Figure 2 shows that the AMBA AHB signal tracing can be divided into two abstract dimensions: signal abstraction and timing abstraction.The abstraction here means the granularity of signal and timing observation. At signal dimension, it has three abstraction levels which are all signals, bus state, and master operation. The all signals level means all bus signals according to the bus transfer operation will be traced; bus statelevel means that using master transition states to represent the transfer status and does not record related control signals. Master operation level means that bus tracer traces signals only related to the current bus master transfer operation such as address/data bus signals, and omits the bus transition states. At timing dimension, it has two abstraction levels which are cycle level and transaction level. The bus tracer records bus signals cycle-by-cycle in cycle level; in transaction level, bus tracer records the trace only when signals transactions occurred. In other words, if a signal remains the same value during a transfer, the bus tracer does not record the signals value besides at the first time. For example, the transfer direction such as READ or WRITE would not be changed during a bus transfer, and the bus tracer records the transfer direction only at the first cycle of current bus transfer in transaction level. According to different levels of abstraction, we define five trace modes of combinations of these levels, as shown in Figure 2.

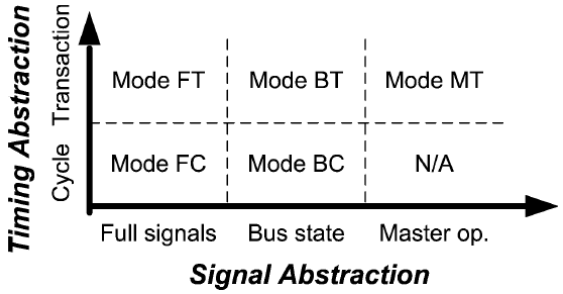

Fig.2 Multiresolution trace modes.

For the bus tracing, lower abstraction levels provide more detail trace information, and higher abstraction levels provide smaller trace size. Designers can select and change the trace mode during a bus tracing process depend on the debugging purposes. The advantage of signal and timing abstraction is that the user can choose lower abstraction level,for example, mode 1 or mode 2, to observe detail signals transitions. On the other hand, designers choose higher abstraction level to observe the bus signals

variation behavior.Because the lower abstraction level provides more accuracy of bus tracing, the trace size would also be large. Fortunately,it is reasonable that designers use lower abstraction level to do the detail observation only during several critical cycles, and the trace size is acceptable.Since higher abstraction levels reduce more unnecessary signals to be traced, the trace size would be small and the trace depth could be extended. Also, the multi-resolution approach aids designers in signal observation and debugging more easily. By using the multi-resolution trace approach, the hardware debugging process can be treated as the software program debugging methodology. Using higher abstraction level to trace bus signal just as debug the program in step over or step out mode; using lower abstraction level, such as mode 1, to trace bus signal is similar to debug the program in single stepping mode. Trigger points for the trace mode change can be treated as breakpoints/ watchpoints. Figure 3 illustrates the trace size variations for different trace modes. For example, in trace mode 5, only the address and data bus signals would be traced at transaction level and resulted in smallest trace size. On the other hand, in trace mode 1, not only address and data bus signals, but transfer control signals would be traced cycleby-cycle and led to largest trace size. It is observably that the signal/timing abstraction approach can reduce the trace size efficiently.

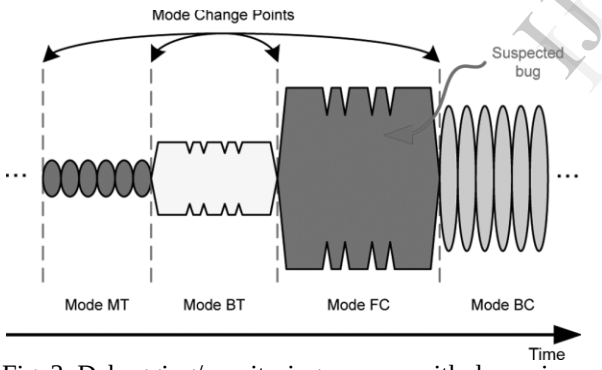

Fig. 3. Debugging/monitoring process with dynamic mode change. The trace size varies with the trace modes. Mode FC consumes the largest space. Mode MT consumes the smallest space.

Master Transition States

We abstract the AMBA AHB bus master transition states from its bus transfer behavior as shown in Figure 4. Each state represents the combination of corresponding control signals under current bus transfer. For example, the state number 2 (Normal) means that the bus master is performing a transfer and the control signals of HREADY must be 'OK'and HTRANS must be 'Non-SEQ' or 'SEQ'.

When the user chooses the trace mode 3 or mode 4 (bus state abstraction level), the bus tracer records the bus states instead of control signals, and can save the trace size.

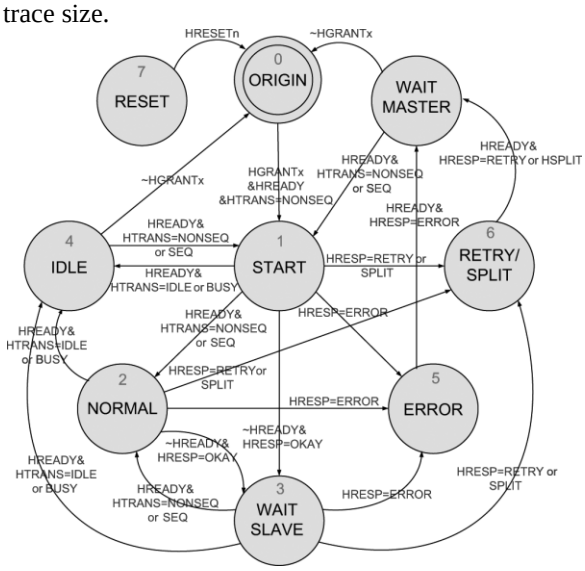

Fig.4 BSM for encoding bus master behaviors

4. Trace Compression

Address Bus Trace Compression

We use two phases approach to compress address data. In the first phase, we omit the sequential addresses and only record the non-sequential addresses. In the second phase, we use a dictionary table to store the *recently used of nonsequentialaddresses*, and record the index value instead of original address value. *Phase 1: Branch/Target Filtering Approach*. A software program, when compiled to the assembly or binary code, consists of a number of basic blocks. A basic block consists of a sequence of linearly executed instructions. The first and last instruction in a basic block is called a target and branch instruction respectively. Since the instructions within a basic block are executed as a group, it suffices to record only the addresses of the target and branch instructions when tracing the (program) address bus signals. *Phase 2: Dictionary Approach*. In this phase, branch and target addresses are stored in a CAM-based dictionary table equentially. If the current address can be found in the table (*dictionary hit*), the corresponding index value would be recorded. On the other hand, if the current address cannot be found in the table (*dictionary miss*), the full address value would be recorded and this address would be stored in the table. When the table is full, the next 'miss address' would be stored in the first entry of the table and replace the original address value.

Data Bus Trace Compression

Since the signal variations on the data bus are not regular that compared with program address bus. Using the differential approach based on subtraction is the convenience way to reduce the data bus trace size and the hardware cost of subtraction is small but the compression ratio is low (about 20%-30%).

Control Signals Trace Compression

When a bus master is performing a bus transfer, the control signals, such as read/write, width of the transfer, transfer size, etc. don't change their value during a complete bus transfer. Therefore, we can use few bits to encode the combinations of these control signals, and record the encoded value instead of record all control signals value For example, in an AMBA platform, the control signals, e.g. HWRITE, HBURST[2:0], HSIZE[2:0], HPROT[3:0] , and HMASTER[3:0] don't change their value during a bus transfer. Therefore the original trace size of these control signals is 15bits. If we use 3bits to encode the combination of these control signals, we can reduce trace size by about (1 - 3/15) x 100% = 80%. In an AMBA system, the combinations of control signals are more than 8 (23), the control signals trace compression module provides a CAMbased dictionary table. The concept is similar to compress the address bus (phase 2). If the current combination of control signals is appeared in the table, the index value (3- bit) would be recorded. On the other hand, we will record the 15-bits control signals when the table miss occurred.

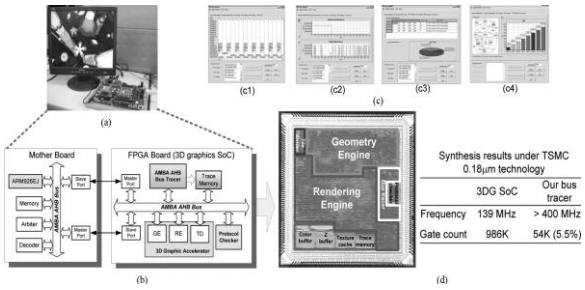

Fig. 18. Chip and prototyping of the 3-D graphics SoC. (a) The ARM's Versatile integrated evelopment board. (b) The SoC block diagram in the Versatile. (c)The graphic user interface of the trace analyzer. c1 shows the decompressed waveform. c2 shows the address/data distribution. c3 shows the protocol analysis in pie chart. c4 shows the master behavior analysis. (d) Chip photo of the 3-D graphics SoC with our bus tracer and the synthesis results.

5. EXPERIMENTAL RESULTS

Tracing Results at Different Trace modes

The specification of the implemented SYS-HMRBT bus tracer is shown in Table V. It has been implemented at C, RTL, FPGA, and chip levels. The synthesis result with TSMC 0.13- technology is shown in Table VI. The bus tracer costs only about 41 K gates, which is relatively small in a typical SoC. The largest component is the FIFO buffer in the packing module. The second one is the compression module. The cost to support both the pre-T and post-T capabilities (periodical triggering module) is only 1032 gates. The major component of the event generation module is the event register, which is roughly 1500 gates per register. The implementation in this paper has two event registers. More registers can be added if necessary. Compared with our previous work [13], the gate count is reduced by 31%. The reason is that this paper optimizes the ping-pong architecture by sharing most of the data path

TABLE V
SPECIFICATION OF THE IMPLEMENTED SYS-HMRBT BUS TRACE

Feature	Configuration
Trace Mode	FC, FT, BC, BT, MT
Trace Direction	Pre-T, Post-T
Input AHB signals	91 bits (HADDR, HRDATA, HWDATA, ACS's, PCS's)
Output trace word	32 bits
Pipeline stage	5
Max. # of masters	16
FIFO buffer	512 bits

TABLE
TRACE COMPRESSION RATIO AT DIFFERENT TRACE MODES

Program	Uncompressed trace size	Compression Ratio = 1 − (Trace size after compression / Original trace size): Mode FC	Mode FT	Mode BC	Mode BT	Mode NT
Perpetual Calendar	910,000	77.58%	83.00%	90.05%	94.67%	96.99%
Fibonacci Sequence	910,000	78.32%	80.05%	87.91%	90.45%	94.73%
G.C.D.	910,000	83.22%	83.45%	89.14%	92.76%	95.68%
Towers of Hanoi	910,000	78.64%	80.61%	85.99%	88.58%	92.56%
Knight Problem	910,000	78.85%	80.51%	96.39%	97.16%	98.32%
Quick Sort	910,000	78.27%	79.99%	99.64%	99.91%	99.80%
Geometric mean		79.00%	81.26%	91.39%	93.81%	96.32%

TABLE
TRACE DEPTH ANALYSIS IN DIFFERENT MEMORY SIZE

Trace memory size	Full signals, no compression (AHBTRACE [5])	Mode FC	Mode FT	Mode BC	Mode BT
2 KB	180	821 (4.7x)	961 (5.3x)	1,303 (7.2x)	1,591 (8.8x)
4 KB	360	1,641 (4.6x)	1,925 (5.3x)	2,641 (7.3x)	3,243 (9.0x)
8 KB	720	3,303 (4.6x)	3,885 (5.4x)	5,333 (7.4x)	6,559 (9.1x)
16KB	1440	6,619 (4.6x)	7,761 (5.4x)	10,719 (7.4x)	13,001 (9.0x)

The number in parentheses indicates the trace depth improvement over the noncompression method for the corresponding trace memo

investigate that retarget the bus tracer into various types of bus system such as multi-layer AMBA and AXI.

Simulation Results

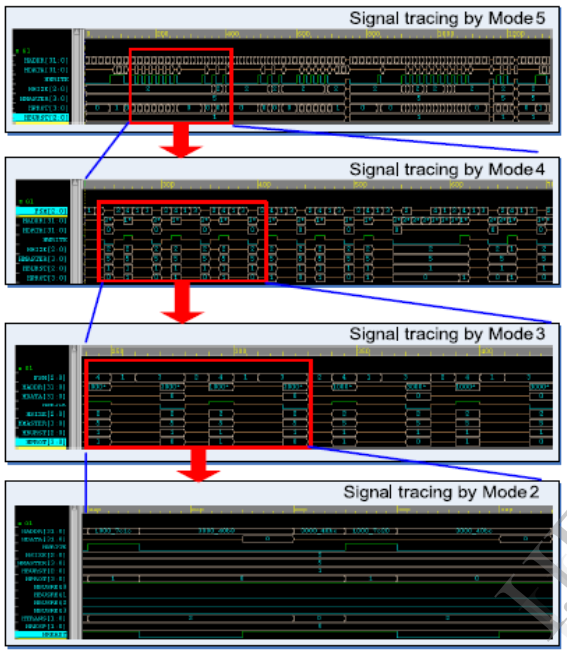

Figure : AMBA signal tracing using different trace modes (MP3 decoder application).

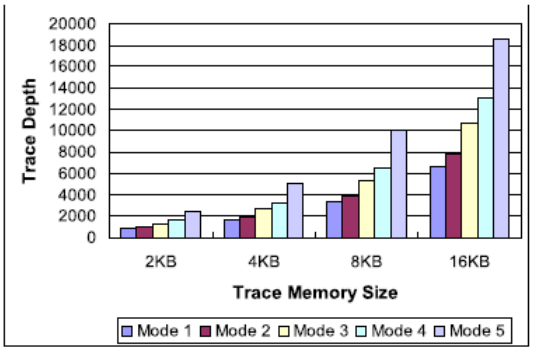

Figure : Trace depth vs. different trace modes (Benchmark:Tower of Hanoi).

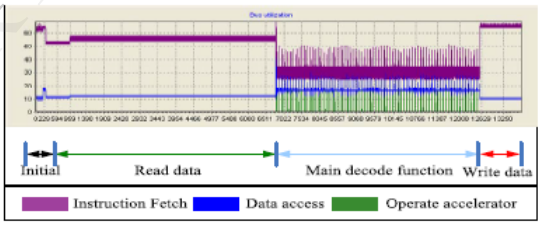

Figure : AMBA bus utilization in MP3 decoder application.

6. CONCLUSION

Bus signal tracing can help designer to debug and analysis system design during implementation stage and final chip testing. Unlike the traditional cycle-based trace, the multiresolution trace approach is proposed which supports both cycle and transaction-based trace mechanisms, and provides the levels of abstraction for easy debugging. Furthermore, the different types of trace result can be stored in a single trace file. For trace reduction issue, AMBA address, data , and control signals trace can be compressed according to their own characteristics. The experimental results show that the proposed approach can reach a good compression ratio of 96% and the trace depth is more than two thousand cycles at the higher abstraction level. In our future work, we will

REFERENCES

[1] Verilog Hardware Description Language, Section 18:Value change dump (VCD) files, IEEE Std. 1364-2005,2006.

[2] Example AMBA SYstem User Guide ARM DUI 0092C, ARM, Aug. 1999.

[3] AMBA Specification (Rev 2.0) ARM IHI0011A, ARM, May 1999.

[4] A. Mayer, H. Siebert, and K. D. McDonald-Maier,"Debug support, calibration and emulation for multiple processor and powertrain control socs," in Proceedings of the Conference on Design, Automationand Test in Europe (DATE'05), vol. 3, Mar. 7–11,2005, pp. 148–152.

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

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

[7] E. Rotenberg, S. Bennett, and J. E. Smith, "A trace cache microarchitecture and evaluation," IEEE Trans. Comput., vol. 48, pp. 111–120, Feb. 1999.

[8] J. Huang and D. J. Lilja, "Extending value reuse to basic blocks with compiler support," IEEE Trans. Comput., vol. 49, pp. 331–347, Apr. 2000.

[9] A. B. T. Hopkins, R. G. S. Scottow, and K. D. McDonald-Maier, "Generic data trace unit and trace compression for system-on-chip," in IEESoC Design,Test and Technology Seminar, Loughborough, UK, Sep. 15, 2004.

[10] A. B. T. Hopkins and K. D. McDonald-Maier, "Debug support strategy for systems-on-chips with multiple processor cores," IEEE Trans. Comput., vol. 55, pp. 174–184, Feb. 2006.

[11] Design Debugging Using the SignalTap II Embedded Logic Analyzer, Altera Inc., Dec. 2004.

[12] ChipScope Pro Software and Cores User Guide, Xilinx Inc., Oct. 2004.

[13] AMBA Navigator Spec Sheet, First Silicon Solutions (FS2) Inc

Author's Profile

FIRST AUTHOUR:

Dr. SHAIK MEERAVALI,

Professor and Head, Department of electronics and communication Engg, RRS college of engineering and Technology, Muthangi, Andhra Pradesh, India.

Second author

.Master Of Scholar, RRS College of Engineering and Technology, Muthangi, , Andhra Pradesh, India.