ISSN: 2277-9655
Impact Factor: 1.852

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Implementation of AHB Bus Tracer with Compression and Multi Resolution for SOC

D.Parvathi[*1], B. Chandrasingh[2]
[*1] Assistant Professor, Vardhaman college of Engineering, India
[2] Assistant Professor, J.b.i.e.t, India
d.parvathi@vardhaman.org

Abstract

In the system-on-chip (SOC) debugging and performance analysis/optimization, monitoring the on-chip bus signals are necessary. But, such signals are difficult to observe since they are deeply embedded in a SOC and no sufficient I/O pins to access those signals. Therefore, we embed a bus tracer in SOC to capture the bus signals and store them. The stored trace memory can be loaded to the trace analyzer software for analysis. In this paper, we propose an embedded multi-resolution AMBA trace analyzer that provides the trade-off between the trace granularity and the trace depth. It consists of two major trace approaches: (1) the signal monitor/tracing which provides the levels of. Abstraction, and (2) the trace reduction. In the first approach, 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. The SOC can be verified in field-programmable gate array. 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, post-T trace, pre-T trace, SOC debugging.

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 cycle based 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.

ISSN: 2277-9655
Impact Factor: 1.852

Bus Tracer Architecture

Fig. 1 is the bus tracer overview. It mainly contains four parts: Event Generation Module, Abstraction Module, Compression Modules, and Packing Module.

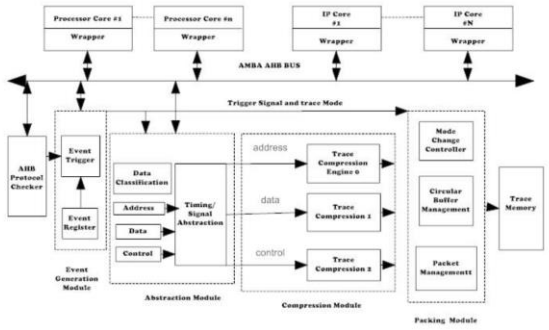


Fig 1.0 Multi Resolution Block diagram

A Event Generation Module

The trace and trace mode starting and stopping are decided by event generation module. The triggering events on the bus controlled by event registers. The matching circuit is used to compare bus activities with the events specified in the event registers. We can connect an AHB bus protocol checker (HPChecker) [10] to the Event Generation Module, as shown in Fig.3, to capture the bus protocol related trace.

B Abstraction Module

This Module monitors the AMBA bus and selects/filters signals based on the abstraction mode. Depending on the abstraction mode, some signals are ignored, and some signals are reduced.

C Compression Module

This Module monitors the AMBA bus and selects/filters signals based on the abstraction mode. Depending on the abstraction mode, some signals are ignored, and some signals are reduced.

D Packing Module

This Module receives the compressed data from the compression module. It processes them and writes them to the trace memory. Packet management, circular buffer management, and mode change control are managed by this module.

Compression Mechanism

To reduce the size, the 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. They are Program address compression, Branch/Target filtering, Dictionary based compression, Slicing, Data address/value compression, Control signal compression. Integrating the bus tracer into a SOC is done by simply tapping the bus tracer to the AHB bus. An on-chip processor or an external debugging host controls the bus tracer. Real-time tracing is achieved when the bus tracer is pipelined to meet the on-chip bus frequency. Since the trace data processing is stream-based, the bus tracer can be easily divided into more pipeline stages to meet aggressive performance requirements.

Trace Compression

A 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 sequentially. 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.

B 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%).

C 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

ISSN: 2277-9655
Impact Factor: 1.852

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.

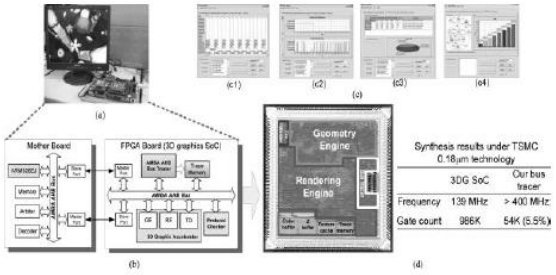
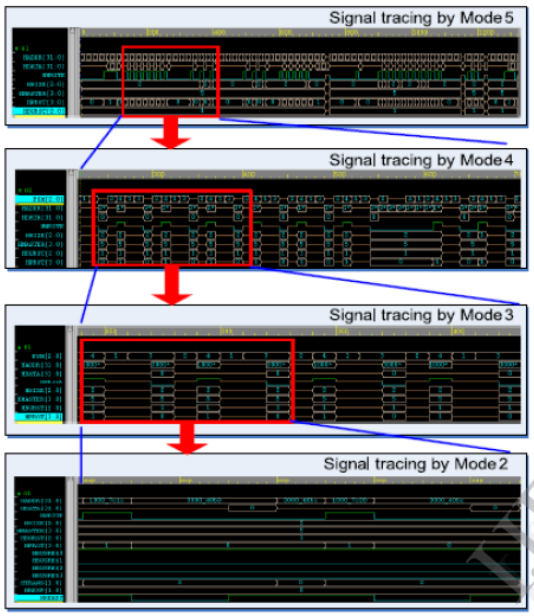


Fig 4.0 Chip and prototyping of the 3-D graphics SoC. (a) The ARM's Versatile integrated development board. (b) The SoC block diagram in the Versatile. (c)The graphic user interface of the trace analyzer. c1 shows the decompressed waveform. c2 shows the address/data distribution. c3 shows the protocol 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.

Simulation Results



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

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*

ISSN: 2277-9655
Impact Factor: 1.852

the Conference on Design, Automationand Test in Europe (DATE'05), vol. 3, Mar. 7–11,2005, pp. 148–152.

[5] T. A. Welch, ―A technique for high-performance data compression,‖*IEEE Trans. Comput.*, pp. 8–19, 1984.

[6] *Embedded Trace Macrocell Architecture Specification*, ARM.

[7] C. MacNamee and D. Heffernan, ―Emerging on-chip debugging techniques for real-time embedded systems,‖ *IEE Comput. Contr. Eng. J.*, pp. 295–303, Dec. 2000.

[8] B. Tabara and K. Hashmi, ―Transaction-level modeling and debug of SoCs,‖ presented at the IP SoC Conf., France, 2004.

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