ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

An Efficient Multi Mode and Multi Resolution Based AHB Bus Tracer

Waheeda Begum
M.Tech, VLSI Design & Embedded System,
Department of E&CE,
Lingaraj Appa Engineering College, Bidar.

Mr. Basavalinga Swamy
Asst professor,
Department of E&CE,
Lingaraj Appa Engineering College, Bidar.

Abstract:

On-Chip program and data tracing is an essential part of any system level development platform for System-On-Chip (SOC). A multi-resolution AHB on-chip bus tracer named SYS-HMRBT (AHB multi-resolution bus tracer), which is vital for debugging & monitoring. The on-chip AHB-Lite bus tracer is with the provision of capturing the bus tracer with different resolutions i.e, with different signals & abstraction levels depending on the need to match with specific debug analysis needs. In addition it allows users to switch with the trace resolution dynamically so that appropriate resolution levels at different modes can be applied to different segments of the trace. An AHB tracer, traces address signals, data signals & control signals. On the other hand, SYS-HMRBT supports tracing after/ before an event triggering, named post-triggering trace/ pre-triggering trace, respectively. The SOC has been successfully verified in both field- programmable gate array and test chip using Xilinx ISE.

Key Words:

AHB Lite, multi resolution, multimode, periodical triggering, post-T trace, pre-T trace, system-on-chip (SOC) debugging, Xilinx.

1. INTRODUCTION:

A system on chip is an integrated circuit(IC) that integrates all components of a computer or other electronic system into a single chip. It may contain digital, analog, mixed-signal, and often radio-frequency functions-all on a single chip substrate.

A SOC consists of both the hardware and the software controlling the microcontroller, microprocessor or DSP cores, peripherals and interfaces. As more and more IP cores are integrated into a System On-Chip (SOC) design, the communication flow between IP cores has increased drastically and the efficiency of the on-chip bus has become a dominant factor for the performance of the system. In the present technology, high performance and speed are required which is convincingly met by AMBA-AHB Lite. AHB Lite is the subset of AMBA-AHB. The AMBA-AHB defines a point to point interface between two communicating entities such as IP cores and bus interfacemodules. One entity acts as a master of the AMBA-AHB instance, and the other as a slave. Only the master can present commands and is the controlling entity. The slave responds to commands presented to it, either by accepting data from the master, or presenting data to the master.

So monitoring the on-chip bus signals is very essential to make the SOC perform efficiently. But these signals are deeply embedded in the SOC which makes the user difficult to observe. Hence as a solution these signals are needed to be abstracted from the bus, which we call as tracing of signals, stored on some on-chip storage and will be off loaded to the analyzer for analysis. But unfortunately, the bus trace size grows at a rapid rate. This is because the rate of signals from different IP cores on a SOC is very high. The different abstraction levels are adopted depending on the designers needs some require all signals at cycle level, while others require only transactions. Thus, there must be a way for capturing traces at different abstraction levels based on the debugging and analysis needs. If the given Trace memory is fixed then the user can tradeoff between the trace granularities and trace length. This feature provides a more flexible tracing.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

1.1 RELATED WORK:

There are hardware approaches to compress the traces which are the lossy trace compression approach and lossless trace compression approach [1]. Debugging of software applications, distributed over several, possibly heterogeneous processor cores [2], [3].ARM provides the AMBA AHB trace macro-cell (HTM) [4] that is capable of tracing AHB bus signals, including the instruction address, data address, and control signals.

The bus signals are characterized into three categories: program address, data address/data and control signals. Dictionary based compression reduces the size of repeated data[5]. In the Embedded Trace Macrocell (ETM) [6] of ARM, slice compression takes advantage of the fact that the high-order bits of instruction addresses rarely change. The real-time compression and compression ratio achieved by using typical existing approaches [7].

Tabbara and Hashmi [8] propose the transaction-level SOC modeling and debugging method. The bus checker can check bus transaction if they obey bus protocol or not [10]. FS2 AMBA Navigator [13] supports bus clock mode and bus transfer mode to trace bus signals on every clock and bus transfer respectively. The transaction-level debugging provides software and hardware designers a common abstraction level to diagnose bugs. The abstraction level is in two dimensions: timing abstraction and signal abstraction. The timing dimension has two abstraction levels which are the cycle level and transaction level. The cycle level captures the signals at every cycle.

The transaction level records the signals only when their value changes. The signal dimension involves grouping of AHB bus signals into four categories: program address, data address/value, access control signals (ACS), and protocol control signals (PCS). Then, we define three abstraction levels for those signals. They are full signal level, bus state level, and master operation level. The full signal level captures all bus signals.

The bus state level further abstracts the PCS by encoding them as states according to the bus-state-machine (BSM). The master state level further abstracts the bus state level by only recording the transfer activities of bus masters and ignoring the handshaking activities within transactions.

This level also ignores the signals when the bus state is IDLE, WAIT, and BUSY. The BSM is designed based on the AMBA AHB 2.0 protocol to represent the key bus handshaking activities within a transaction. The transitions between BSM states follow the AMBA protocol control signals. Combining the abstraction levels in the timing dimension and the signal dimension, we provide five modes in different granularities. The post-T trace captures signals after a triggering event, while the pre-T trace captures signals before the triggering event.

The post-T trace is usually used to observe signals after a known event. The pre-T trace is used to diagnose the cause for unexpected errors by capturing the signals before the errors. In order to overcome the problem, we adopt a periodical triggering technique. We divide the entire trace into several independent small traces. Destroying the initial state of one trace does not affect other traces since every trace has its own initial state.

This technique can be accomplished by periodically triggering a new trace. With minor modification to their control circuitry it can be easily accomplished by the existing trace compression engines. This paper presents the multi-resolution approach that can use different trace modes during a bus signal tracing process.

1.2 AMBA PROTOCOLS:

Figure 1. Shows the different protocols performances from the time of initialization[9].

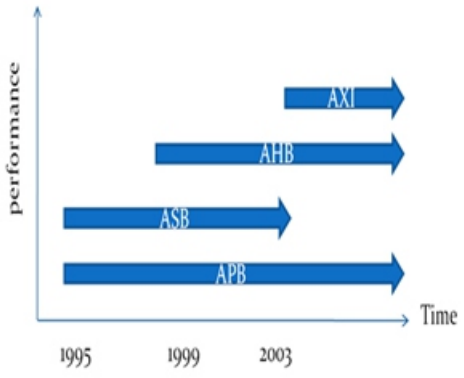

Fig -1: Protocols of AMBA[9]

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

• APB (Advanced Peripheral Bus) mainly used as an ancillary or general purpose register based peripherals such as timers, interrupt controllers, UARTs, I/O ports, etc. It is connected to the system bus via a bridge, helps reduce system power consumption. It is also easy to interface to, with little logic involved and few corner-cases to validate.

• AHB (Advanced High Performance Bus) is for high performance, high clock frequency system modules with suitable for medium complexity and performance connectivity solutions. It supports multiple masters.

• AHB-Lite is the subset of the full AHB specification which intended for use where only a single bus master is used and provides high-bandwidth operation.

2. Top Level Block Diagram:

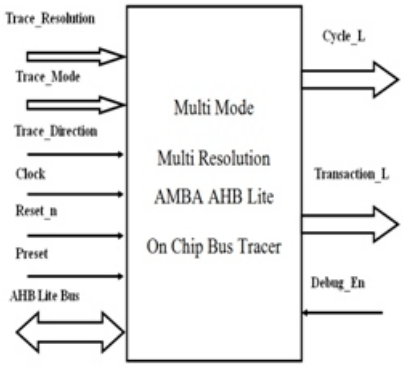

Fig-2: Multi mode multiresolution AHB Lite Tracer

AHB-Lite is a subset of AHB which is the part of AMBA. It includes the features which make it very suitable for high speed sub-micrometer interconnect.The transaction-level debugging provides software andHardware designers a common abstraction level to diagnose bugs. The abstraction level is in two dimensions timing abstraction and signal abstraction. The timing dimension has two abstraction levels which are the cycle level and transaction level. The cycle level captures the signals at every cycle. The transaction level records the signals only when their value changes. The signal dimension involves grouping of AHB bus signals into four categories: program address, data address/value, access control signals (ACS), and protocol control signals (PCS). Then, we define three abstraction levels for those signals. The master state level further abstracts the bus state level by only recording the transfer activities of bus masters and ignoring the handshaking activities within transactions.

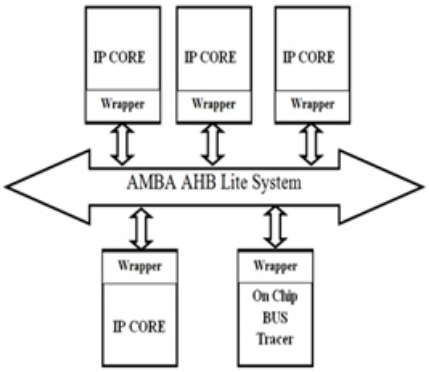

Fig -3: Multiresolution Bus Tracer Block Diagram

This level also ignores the signals when the bus state is IDLE, WAIT, and BUSY. The BSM is designed based on the AMBA AHB 2.0 protocol to represent the key bus handshaking activities within a transaction.

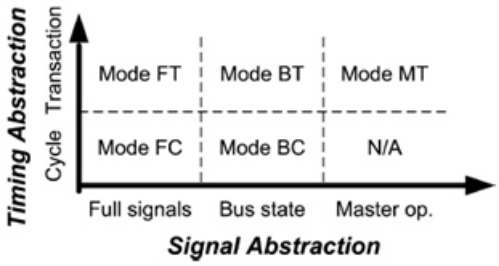

Fig -4: Multiresolution Trace Modes

At the timing dimension, it has two abstraction levels, which are the cycle level and transaction level. The cycle level captures the signals at every cycle. The transaction level records the signals only when their values change (event triggering). For example, since the bus read/write control signals do not change during a successful transfer, the tracer only records this signal at the first and last cycles of that transfer. However, if the signal changes its value cycle-level trace.

A) Mode FC:

The tracer traces all bus signals cycle-by-cycle so that designers can observe the most detailed bus activities. This mode is very useful to diagnose the cause of error by looking at the detail signals. However, since the traced data size of this mode is huge, the trace depth is the shortest among the five modes. Fortunately, it is acceptable since designers using the cycle-level mode trace only focus on a short critical period.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

B) Mode FT:

The tracer traces all signals only when their values are changed. In other words, this mode traces the untimed data transaction on the bus. Comparing to Mode FC, the timing granularity is abstracted. It is useful when designers want to skim the behaviour of all signals instead of looking at them cycle-by-cycle. Another benefit of this mode is that the space can be saved without losing meaningful information. Thus, the trace depth increases.

C) Mode BC:

The tracer uses the BSM, such as NORMAL, IDLE, ERROR, and so on, to represent bus transfer activities in cycle accurate level. Comparing to Mode FC, although this mode still captures the signals cycle-by-cycle, the signal granularity is abstracted. Thus, designers can observe the bus hand-shaking states without analyzing the detail signals. The benefit is that designers can still observe bus states cycle-by-cycle to analyze the system performance.

D) Mode BT:

The tracer uses bus state to represent bus transfer activities in transaction level. The traced data is abstracted in both timing level and signal level; it is a combination of Mode BC and Mode BT. In this mode, designers can easily understand the bus transactions without analyzing the signals at cycle level.

E) Mode MT:

The tracer only records the masters behavior, such as read, write, or burst transfer. It is the highest abstraction level. This feature is very suitable for analyzing the master transactions. The major difference compared with Mode BT is that this mode does not record the transfer handshaking activities and does not capture signals when the bus state is IDLE, WAIT, and BUSY. Thus, designers can focus on only the master transactions

2.1 Dynamic Mode Change:

Our bus tracer also supports dynamic mode change (DMC) feature. This feature allows designers to change the trace mode dynamically in real-time. As Fig. 5 shows, the trace mode changes seamlessly during execution.

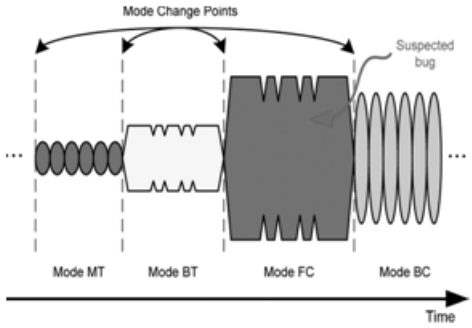

Fig -5: debugging process with dynamic mode change

Dynamic mode change has two benefits. One is that it provides customized traces according to the debugging purpose. The other is that designers can make tradeoffs between the trace granularity and the trace depth. Thus, the trace memory utilization is more efficient. Fig. 5 shows an example using dynamic mode change to diagnose a suspected bug. At first, designers can use Mode MT to have the top view so that they can skim the master behaviors very quickly. Then, when the time is closed to the suspected bug, they can switch to Mode BT.

This provides more information about all operations on the bus and thus, designers can check the detail operations. It also helps establishing the time line of system behaviors. After, designers switch to the Mode FC to focus on every signal at cycle level for error diagnosis. Finally, after Mode FC, designers can switch to Mode BC to see what operations are affected by this bug. Since the behavior at every cycle is worth noticing, this mode preserves the cycle-level trace. However, since designers only care about the behaviors instead of all signals, this mode abstracts the signal level and speeds up the debugging process.

The dynamic mode change is achieved by setting up the event registers. The event registers define the start/stop time of a trace and the trace mode. Thus, when the trigger condition meets and a new trace begins, the new trace starts in the trace mode specified in the event registers.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

To provide better debugging flexibility, the captured tracescan be abstracted into higher abstraction level traces by our trace analyzer software. For example, the traces of mode FC can be abstracted into traces of mode FT, mode BC, mode BT, and mode MT. The traces of mode BC can be abstracted into traces of mode BT and mode MT. This feature increases the debugging flexibility.

2.2 Trace Direction: pre-T/Post-T Trace :

Supporting both trace directions provides the flexible debugging strategies. The post-T trace captures signals after a triggering event, while the pre-T trace captures signals before the triggering event. The post-T trace is usually used to observe signals after a known event. The pre-T trace is useful for diagnosing the causes of unexpected errors by capturing the signals before the errors.

The mechanisms of the pre-T trace and the post-T trace are different. The Post-T trace is simpler since the start time and the stop time are known. It is activated when the target event is matched and is turned off when the trace buffer is full. On the other hand, the stop time of the pre-T trace is unpredictable. The solution is to start tracing as soon as system reset (or some other turning-on event). When the trace buffer is full, the new trace data wrap around the trace buffer, which means the oldest data are sacrificed for the newest ones.

Wrapping around the trace buffer causes a problem when the trace needs to be compressed. Typical loss-less compression algorithms work by storing some initial (previous) states of the trace first and then calculate the relationship between the current data and the previous states. Since the size of the relationship is smaller than the data size, e.g., the difference, it saves spaces. The initial state of the trace, which is stored at the head of the buffer, will be destroyed if the trace is wrapped around, and thus making the rest of the trace not de compressable. The tracing packet come out from the packet module stored into the Trace memory. This trace memory we can design using FIFOs. The trace for circular buffer management, it manages the accesses to the trace memory. Since the size of a packet is variable but the data width of the trace memory is fixed, this module 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. 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.

3 EXPERIMENTAL RESULTS:

By simulation and synthesis the following results are obtained for each cycle at different abstraction levels at different modes. Here Xilinx tool is used in order to synthesize and simulate the design process and also the netlist generation.

A.Simulation Result:

Figure 6 shows the input output declaration for five different modes.

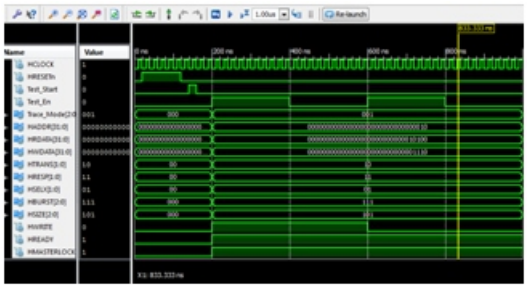

Fig-6: AHB Lite tracer input outputs with different modes

Mode FC- write:

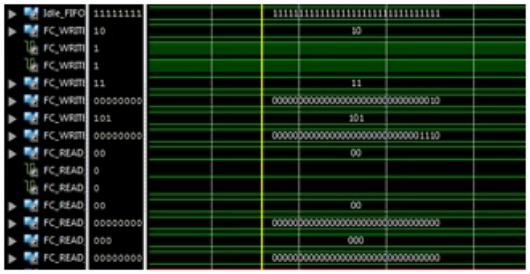

Fig-7: Mode FC-write

Mode FC-Read:

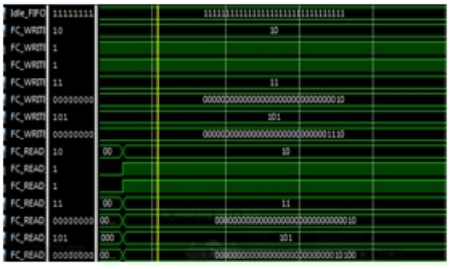

Fig-8: Mode FC-Read

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Figure 7 shows the mode FC write operation, whenever test-en is active high the write data is active high and the read data is active low. When test-en becomes active low the write data remains the same and the read data becomes active high as shown in figure 8. Hence the process is repeated for all the multi-resolution trace modes.

B.Synthesis Result:

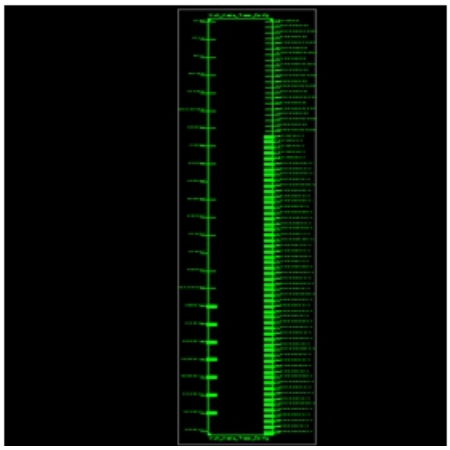

Fig-9: RTL schematic view of multimode bus tracer

Area Report:
Multi Resolution Area Report

```
Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:             358  out of  18224     1%
 Number of Slice LUTs:                  260  out of   9112     2%
    Number used as Logic:               260  out of   9112     2%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    360
   Number with an unused Flip Flop:       2  out of    360     0%
   Number with an unused LUT:           100  out of    360    27%
   Number of fully used LUT-FF pairs:   258  out of    360    71%
   Number of unique control sets:         7

IO Utilization:
 Number of IOs:                         366
 Number of bonded IOBs:                 364  out of    232   156% (*)
    IOB Flip Flops/Latches:               1

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:              1  out of     16     6%
```

Fig-10: Area Report for Multi resolution bus tracer

Multi Mode Area Report

```
Device utilization summary:
---------------------------

Selected Device : 6slx16csg324-3


Slice Logic Utilization:
 Number of Slice Registers:             943  out of  18224     5%
 Number of Slice LUTs:                  884  out of   9112     9%
    Number used as Logic:               884  out of   9112     9%

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    981
   Number with an unused Flip Flop:      38  out of    981     3%
   Number with an unused LUT:            97  out of    981     9%
   Number of fully used LUT-FF pairs:   846  out of    981    86%
   Number of unique control sets:        15

IO Utilization:
 Number of IOs:                         953
 Number of bonded IOBs:                 951  out of    232   409% (*)
    IOB Flip Flops/Latches:               1

Specific Feature Utilization:
 Number of BUFG/BUFGCTRLs:              1  out of     16     6%
```

Fig-11: Area Report for Multi Mode Multi resolution bus tracer

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 3.016ns (Maximum Frequency: 331.574MHz)
   Minimum input arrival time before clock: 5.911ns
   Maximum output required time after clock: 3.634ns
   Maximum combinational path delay: No path found
```

Timing Report:
Multi Resolution Timing Report:

Fig-12: Timing Report for Multi Resolution Bus tracerMulti mode

Multi Resolution Timing Report:

```
Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 3.248ns (Maximum Frequency: 307.885MHz)
   Minimum input arrival time before clock: 5.913ns
   Maximum output required time after clock: 3.634ns
   Maximum combinational path delay: No path found
```

Fig-13: Timing Report for Multi mode Multi Resolution Bus tracer

4. CONCLUSION:

AHB-Lite is capable of achieving high performance with a maximum frequency of 159.894MHz.The bus traces with 5 modes of resolution and the design is verified for all cases of 5 modes.

The modes can be switched dynamically while tracing. 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. The reason is that this paper optimizes the pingpong architecture by sharing most of the of the data path.

REFERENCES:

[1]. E. E. Johnson, J. Ha, and M. B. Zaidi, —Lossless trace compression, IEEE Trans. Comput., vol. 50, pp. 158–173, Feb. 2001.[2]. CoreSight: V1.0 Architecture Specification, ARM.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[3]. R. Leatherman and N. Stollon, —An embedded debugging architecture for SoCs, IEEE Potentials, vol. 24, no. 1, pp. 12–16, Feb-Mar 2005.

[4]. ARM Ltd., San Jose, CA, —ARM. AMBA AHB Trace Macrocell (HTM) technical reference manual ARM DDI 0328D, 2007.

[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]. J. Gaisler, E. Catovic, M. Isomaki, K. Glembo, and S. Habinc, "GRLIB IP core user's manual, gaisler research," 2009.

[8] AMBA Navigator Spec Sheet, First Silicon Solutions (FS2) In.