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

An on Chip AHB Bus Tracer with Real Time Compression and Dynamic Multi Resolution Supports SOC

A Renuka
M.Tech,
Geetanjanli College of Engineering,
Hyderabad.

M Uma Rani, M.Tech
Assistant Professor,
Geetanjanli College of Engineering,
Hyderabad.

1.INTRODUCTION:

The On-Chip bus is an important system-on-chip (SoC) infrastructure that connects major hardware components. Monitoring the on-chip bus signals is crucial to the SoC debugging and performance analysis/optimization. Unfortunately, such signals are difficult to observe since they are deeply embeddedin a SoC and there are often no sufficient I/O pins to access these signals. Therefore, a straightforward approach is to embed a bus tracer in SoC to capture the bus signal trace and store the trace in an 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 grows rapidly. For example, to capture AMBA AHB 2.0 bus signals running at 200 MHz, the trace grows at 2 to 3 GB/s. Therefore, it is highly desirable to compress the trace on the fly in order to reduce the trace size. However, simply capturing / compressing bus signals is not sufficient for SoC debugging and analysis, since the debugging / analysis needs are versatile: some designers need all signals at cycle-level, while some others only care about the transactions. For the latter case, tracing all signals at cycle-level wastes a lot of trace memory. Thus, there must be a way to capture traces at different abstraction levels based on the specific debugging/analysis need.

1.1 Introduction to the AMBA Buses:
1.1.1 Overview of the AMBA specification:

The Advanced Microcontroller Bus Architecture (AMBA) specification defines an on-chipcommunications standard for designing high-performance embeddedmicrocontrollers.Three distinct buses are defined within the AMBA specification:

» the Advanced High-performance Bus (AHB)
» the Advanced System Bus (ASB)
» theAdvanced Peripheral Bus (APB).

A test methodology is included with the AMBA specification which provides an infrastructure for modular macrocell test and diagnostic access.

1.1.2 Advanced High-performance Bus (AHB):

The AMBA AHB is for high-performance, high clock frequency system modules.The AHB acts as the high-performance system backbone bus. AHB supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro-cell functions. AHB is also specified to ensure ease of use in an efficient design flow using synthesis and automated testtechniques.

1.1.3 Advanced System Bus (ASB):

The AMBA ASB is for high-performance system modules.AMBA ASB is an alternative system bus suitable for use where the high-performance features of AHB are not required. ASB also supports the efficient connection of processors, on-chip memories and off-chip external memory interfaces with low-power peripheral macro-cell functions.

1.1.4 Advanced Peripheral Bus (APB):

The AMBA APB is for low-power peripherals. AMBA APB is optimized for minimal power consumption and reduced interface complexity to support peripheral functions. APB can be used in conjunction with either version of the system bus.

1.1.5Objectives of the AMBA specification:

The AMBA specification has been derived to satisfy four key requirements:
• to facilitate the right-first-time development of embedded microcontroller products with one or more CPUs or signal processors

• to be technology-independent and ensure that highly re-usable peripheral andsystem macro-cells can be migrated across a diverse range of IC processes and beappropriate for full-custom, standard cell and gate array technologies

• to encourage modular system design to improve processor independence,providing a development road-map for advanced cached CPU cores and thedevelopment of peripheral libraries

• to minimize the silicon infrastructure required to support efficient on-chip andoff-chip communication for both operation and manufacturing test.

1.2 A Typical AMBA-Based Microcontroller:

An AMBA-based microcontroller typically consists of a high-performance system backbone bus (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth, on which the CPU, on-chip memory and other Direct Memory Access (DMA) devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high-performance bus is a bridge to the lower bandwidth APB, where most of the peripheraldevices in the system are located.

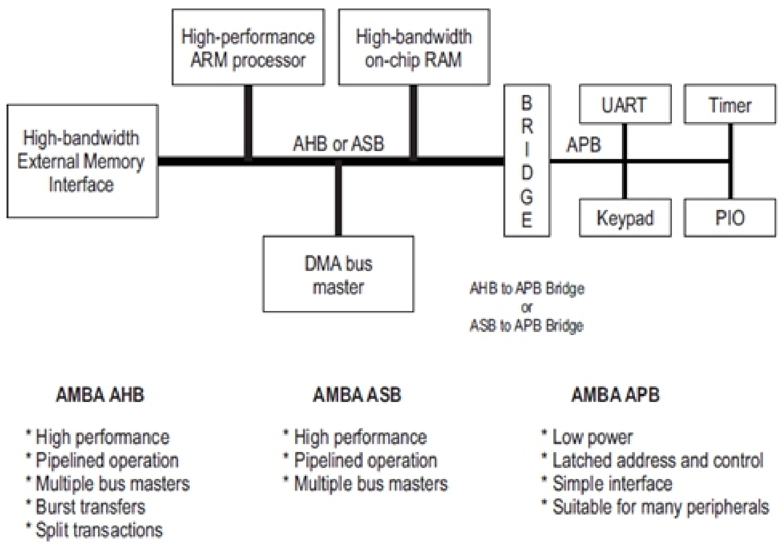


Figure 1

AMBA APB provides the basic peripheral macro-cell communications infrastructure asa secondary bus from the higher bandwidth pipelined main system bus. Such peripherals typically:

• have interfaces which are memory-mapped registers
• have no high-bandwidth interfaces
• are accessed under programmed control.

The external memory interface is application-specific and may only have a narrow data path, but may also support a test access mode which allows the internal AMBA AHB, ASB and APB modules to be tested in isolation with system-independent test sets.

1.3 Terminology:

The following terms are used throughout this specification.

1.3.1 Bus cycle :

A bus cycle is a basic unit of one bus clock period and for the purpose of AMBA AHB or APB protocol descriptions is defined from rising-edge to rising-edge transitions. An ASB bus cycle is defined from falling-edge to falling-edge transitions. Bus signal timing is referenced to the bus cycle clock.

1.3.2 Bus transfer:

An AMBA ASB or AHB bus transfer is a read or write operation of a data object, which may take one or more bus cycles. The bus transfer is terminated by a completion response from theaddressed slave.

1.3.3 Burst operation:

A burst operation is defined as one or more data transactions, initiated by a bus master, which have a consistent width of transaction to an incremental region of address space. The increment step per transaction is determined by the width of transfer (byte, half-word, word). No burst operation is supportedon the APB.A typical AMBA AHB system design contains the following components:

1.3.4 AHB master :

A bus master is able to initiate read and write operations by providing an address and control information. Only one bus master is allowed to actively use the bus at any one time.

1.3.5 AHB slave :

A bus slave responds to a read or write operation within a given address-space range. The bus slave signals back to the active master the success, failure or waiting of the data transfer.

2. LITERATURE SURVEY:

Since the huge trace size limits the trace depth in a trace memory, there are hardware approaches to compress the traces. The approaches can be divided into lossy and lossless trace compression.The lossy trace compression approach achieves high compression ratio by sacrificing the accuracy; the original signals cannot be reconstructed from the trace. The purpose of this approach is to identify if a problem occurs. Anis and Nicolici [2] use the multiple input signature register (MISR) to perform lossy compression. The results are stored in a trace memory and compared with the golden patterns to locate the range of the erroneous signals.

The locating needs rerunning the system several times with finer and finer resolution until the size of the search range can fit in the trace memory. Such approach is suitable for deterministic and repeatable system behaviors. However, for a complex SoC with multiple independent IPs, the on-chip bus activities are usually not deterministic and repeatable. Therefore, lossless compression approaches are more appropriate for real-time on-chip bus tracing.

2.1 AHB Trace Macro-cell (HTM):

The HTM provides address and data trace information about AHB buses. The information from an HTM can be used with the debugger to enable easy, accurate debugging on AHB-based embedded systems. The HTM provides extensive resources for event recognition to generate trigger events. The HTM generates trace data for output through the AMBA Trace Bus (ATB). The trace debug function is non-intrusive and HTM can be controlled using an APB (AMBA v3) interface.The trace operation indicates the data transfers that have taken place to defined memory locations or regions. Other AMBA control information can also be included. However, other operations like IDLE cycles and BUSY cycles in AHB are not traced.The HTM is an Advanced Microcontroller Bus Architecture (AMBA) compliant System-on-Chip (SoC) debug component.

3.ON CHIP AHB BUS TRACER WITH REAL TIME COMPRESSION AND DYNAMIC MULTIRESOLUTION:

As mentioned, compressing all signals at the cycle-accurate-level does not always meet the debugging needs. As SoCs become more complex, the transaction-level debugging becomes increasingly important, since it helps designers focus on the functional behaviors, instead of interpreting complex signals. Tabbara and Hashmi [10] propose the transaction-level SoC modeling and debugging method. The proposed transactors, attaching to the on-chip bus, recognize/monitor signals and abstract the signals into transactions. The transactions, bridging the gap between algorithm-level and the signal-level, enable easy design exploration/debugging/monitoring. Vermeulen propose a communication-centric intrusive debuggingmethod based on the transaction level.

They point out that thetraditional hardware and software debugging cannot work collaboratively, since the software debugging is at the functionallevel and the hardware debugging is at the signal level. As asolution, the transaction-level debugging can provide softwareand hardware designers a common abstraction level to diagnosebugs collaboratively, and thus, help focus problems quickly. Both works indicate that the transaction-level debugging is amust in SoC development.

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

Motivated by the related works, our bus tracer combines abstraction and compression techniques in a more aggressive way.The goal is to provide better compression quality and multiple resolution traces to meet the complex SoC debugging needs.For example, our bus tracer can provides traces at cycle-level and transaction-level to support versatile debugging needs. Besides, features such as the dynamic mode change and bidirectional traces are also introduced to enhance the debugging flexibility.

3.1. Trace Granularity—Multi-resolution Trace:

This section first introduces the definitions of the abstraction level. Then, it discusses the application for each abstraction mode.The abstraction level is in two dimensions: timing abstraction and signal abstraction.

3.1.1 Timing and Signal Abstraction Definition:

At the timing dimension, it has two abstraction levels, which are the cycle level and transaction level, as shown in Table I. The cycle level captures the signals at every cycle.

The transaction level records the signals only when their value 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-by-cycle, the transaction-level trace is similar to the cycle-level trace.

4.ARCHITECTURE OF ON CHIP BUS TRACER:

This section presents the architecture of our bus tracer. We first provide an overview of the architecture for the post-T trace. We then discuss the three major compression methods in this architecture. Finally, we show the extension of the post-T architecture to support the pre-T trace.

4.1 Post-T Tracer Architecture Overview:

It mainly contains four parts:
1. Event Generation Module,
2. Abstraction Module,
3. Compression Modules and
4. Packing Module.

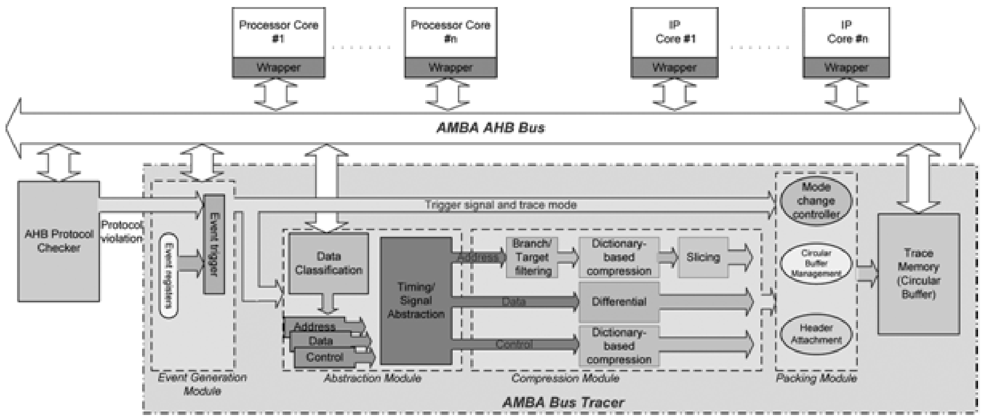


Figure 4.1 Multi-resolution bus tracer block diagram.

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 compressedresults are packed with proper headers and written to the trace memory by the Packing Module.

5. RESULTS AND DISCUSSION

5.1 INTRODUCTION

The On-Chip AHB Bus Tracer with Real-Time Compression and Multi-resolution architecture and the implementation were discussed in the previous chapters. Now this chapter deals with the simulation and synthesis results of the implemented On-Chip AHB Bus Tracer with Real-Time Compression and Multi-resolution. Here Modelsim tool is used in order to simulate the design and checks the functionality of the design. Once the functional verification is done, the design will be taken to the Xilinx tool for Synthesis process and the netlist generation.

5.2 SIMULATION RESULTS

5.2.1 MODE FC

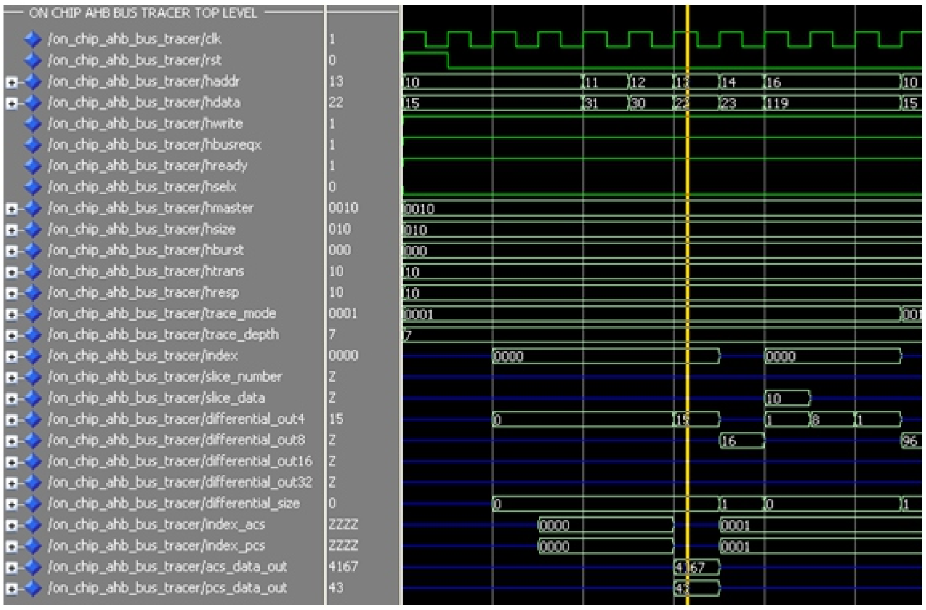

ON CHIP AHB BUS TRACER TOP LEVEL
/on_chip_ahb_bus_tracer/clk 1
/on_chip_ahb_bus_tracer/rst 0
/on_chip_ahb_bus_tracer/haddr 13
/on_chip_ahb_bus_tracer/hdata 22
/on_chip_ahb_bus_tracer/hwrite 1
/on_chip_ahb_bus_tracer/hbusreqx 1
/on_chip_ahb_bus_tracer/hready 1
/on_chip_ahb_bus_tracer/hselx 0
/on_chip_ahb_bus_tracer/hmaster 0010
/on_chip_ahb_bus_tracer/hsize 010
/on_chip_ahb_bus_tracer/hburst 000
/on_chip_ahb_bus_tracer/htrans 10
/on_chip_ahb_bus_tracer/hresp 10
/on_chip_ahb_bus_tracer/trace_mode 0001
/on_chip_ahb_bus_tracer/trace_depth 7
/on_chip_ahb_bus_tracer/index 0000
/on_chip_ahb_bus_tracer/slice_number Z
/on_chip_ahb_bus_tracer/slice_data Z
/on_chip_ahb_bus_tracer/differential_out4 15
/on_chip_ahb_bus_tracer/differential_out8 Z
/on_chip_ahb_bus_tracer/differential_out16 Z
/on_chip_ahb_bus_tracer/differential_out32 Z
/on_chip_ahb_bus_tracer/differential_size 0
/on_chip_ahb_bus_tracer/index_acs ZZZZ
/on_chip_ahb_bus_tracer/index_pcs ZZZZ
/on_chip_ahb_bus_tracer/acs_data_out 4167
/on_chip_ahb_bus_tracer/pcs_data_out 43

Figure 5.1 Simulation results of Mode FC

Simulation results of On-Chip AHB Bus Tracer with Mode FC (Mode Full Signal, Cycle by cycle) as shown in Figure 5.1.Input signal for On-Chip AHB Bus Tracer are AMBA-AHB Bus signals which includes program address, Address /Data value and Control signals(ACS,PCS).

5.2.2 MODE FT

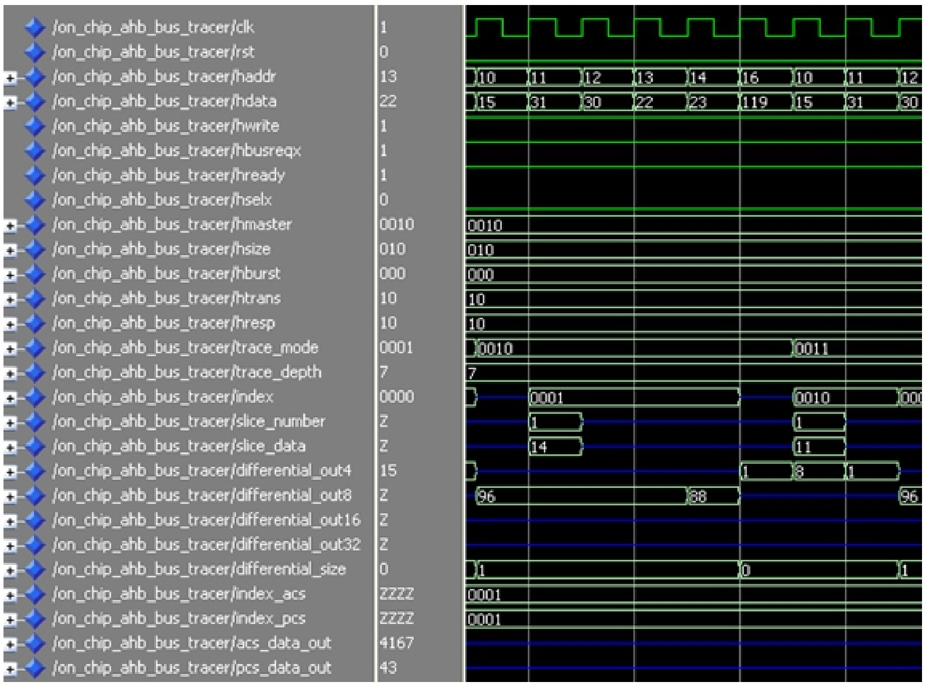

/on_chip_ahb_bus_tracer/clk 1
/on_chip_ahb_bus_tracer/rst 0
/on_chip_ahb_bus_tracer/haddr 13
/on_chip_ahb_bus_tracer/hdata 22
/on_chip_ahb_bus_tracer/hwrite 1
/on_chip_ahb_bus_tracer/hbusreqx 1
/on_chip_ahb_bus_tracer/hready 1
/on_chip_ahb_bus_tracer/hselx 0
/on_chip_ahb_bus_tracer/hmaster 0010
/on_chip_ahb_bus_tracer/hsize 010
/on_chip_ahb_bus_tracer/hburst 000
/on_chip_ahb_bus_tracer/htrans 10
/on_chip_ahb_bus_tracer/hresp 10
/on_chip_ahb_bus_tracer/trace_mode 0001
/on_chip_ahb_bus_tracer/trace_depth 7
/on_chip_ahb_bus_tracer/index 0000
/on_chip_ahb_bus_tracer/slice_number Z
/on_chip_ahb_bus_tracer/slice_data Z
/on_chip_ahb_bus_tracer/differential_out4 15
/on_chip_ahb_bus_tracer/differential_out8 Z
/on_chip_ahb_bus_tracer/differential_out16 Z
/on_chip_ahb_bus_tracer/differential_out32 Z
/on_chip_ahb_bus_tracer/differential_size 0
/on_chip_ahb_bus_tracer/index_acs ZZZZ
/on_chip_ahb_bus_tracer/index_pcs ZZZZ
/on_chip_ahb_bus_tracer/acs_data_out 4167
/on_chip_ahb_bus_tracer/pcs_data_out 43

Figure 5.2 Simulation results of Mode FT

Simulation results of On-Chip AHB Bus Tracer with Mode FT (Mode Full Signal, Transaction level) as shown in Figure 5.2.Input signal for On-Chip AHB Bus Tracer are AMBA-AHB Bus signals which includes program address, Address / Data value and Control signals. Control signals includes Access Control Signals-ACS, Protocol Control Signals –PCS.

5.2.3 MODE BC

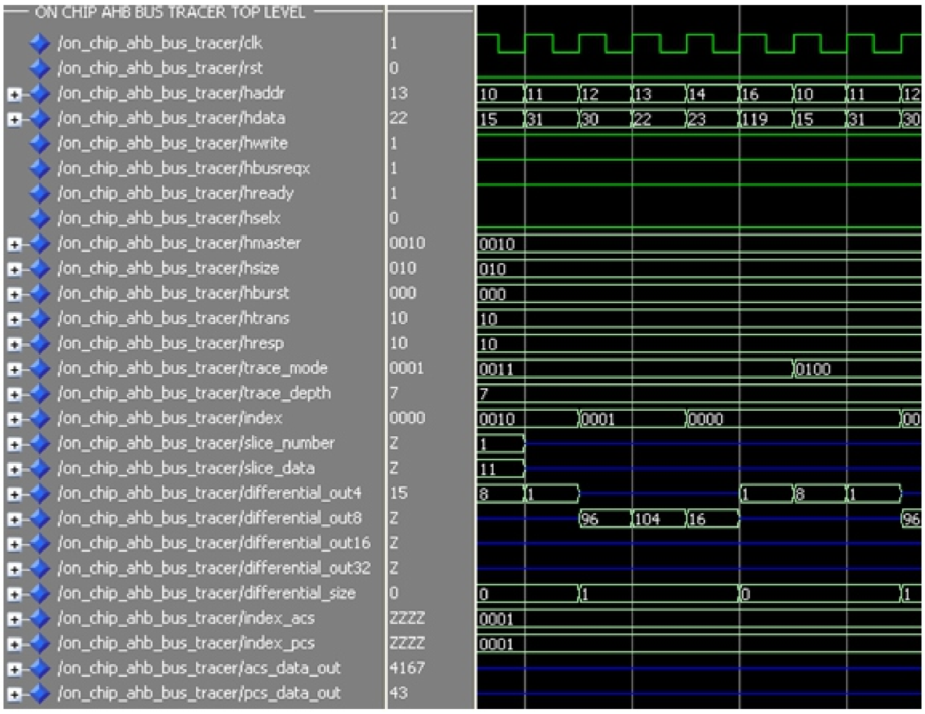

ON CHIP AHB BUS TRACER TOP LEVEL
/on_chip_ahb_bus_tracer/clk 1
/on_chip_ahb_bus_tracer/rst 0
/on_chip_ahb_bus_tracer/haddr 13
/on_chip_ahb_bus_tracer/hdata 22
/on_chip_ahb_bus_tracer/hwrite 1
/on_chip_ahb_bus_tracer/hbusreqx 1
/on_chip_ahb_bus_tracer/hready 1
/on_chip_ahb_bus_tracer/hselx 0
/on_chip_ahb_bus_tracer/hmaster 0010
/on_chip_ahb_bus_tracer/hsize 010
/on_chip_ahb_bus_tracer/hburst 000
/on_chip_ahb_bus_tracer/htrans 10
/on_chip_ahb_bus_tracer/hresp 10
/on_chip_ahb_bus_tracer/trace_mode 0001
/on_chip_ahb_bus_tracer/trace_depth 7
/on_chip_ahb_bus_tracer/index 0000
/on_chip_ahb_bus_tracer/slice_number Z
/on_chip_ahb_bus_tracer/slice_data Z
/on_chip_ahb_bus_tracer/differential_out4 15
/on_chip_ahb_bus_tracer/differential_out8 Z
/on_chip_ahb_bus_tracer/differential_out16 Z
/on_chip_ahb_bus_tracer/differential_out32 Z
/on_chip_ahb_bus_tracer/differential_size 0
/on_chip_ahb_bus_tracer/index_acs ZZZZ
/on_chip_ahb_bus_tracer/index_pcs ZZZZ
/on_chip_ahb_bus_tracer/acs_data_out 4167
/on_chip_ahb_bus_tracer/pcs_data_out 43
10 11 12 13 14 16 10 11 12
15 31 30 22 23 119 15 31 30
0010
010
000
10
10
0011 0100
7
0010 0001 0000 00
1
11
8 1 1 8 1
96 104 16 96
0 1 0 1
0001
0001

Figure 5.3 Simulation results of Mode BC

Simulation results of On-Chip AHB Bus Tracer with Mode BC (Mode Bus State, Cycle level) as shown in Figure 5.3.Input signal for On-Chip AHB Bus Tracer are AMBA-AHB Bus signals which includes program address, Address / Data value and Control signals. Control signals includes Access Control Signals-ACS, Protocol Control Signals –PCS.

5.4.3Technology Schematics View

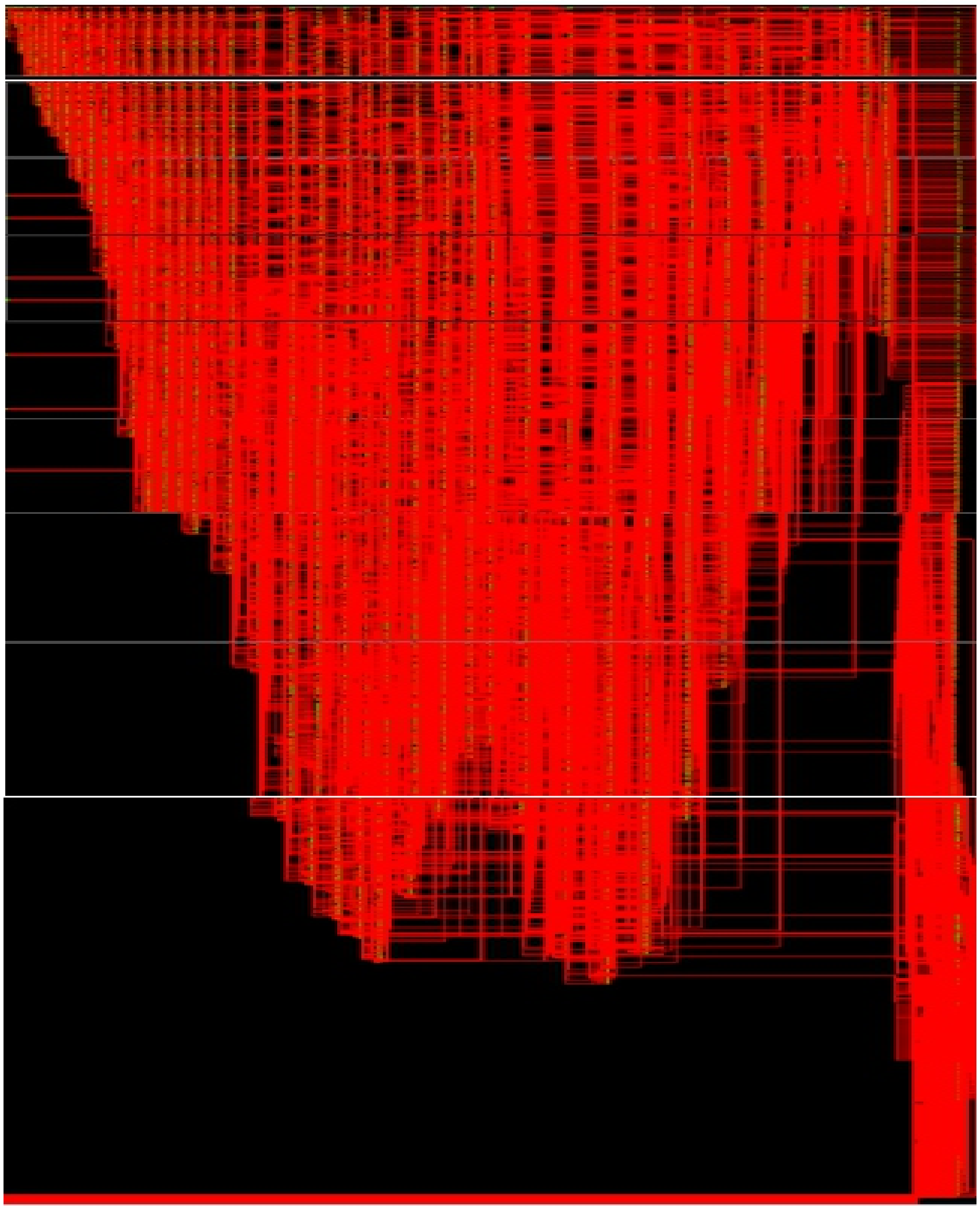
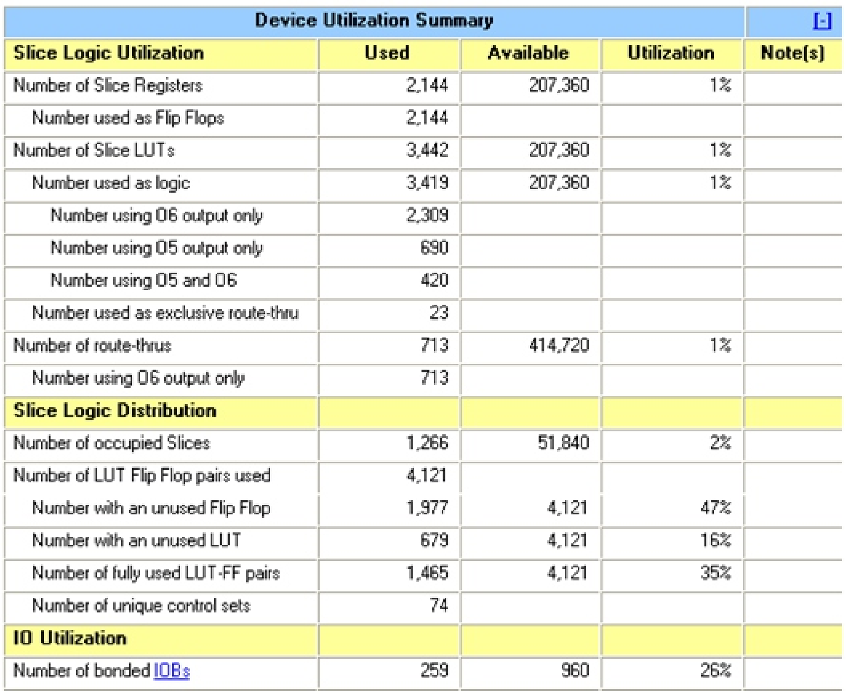


Figure 5.9 TechnologySchematic View of On-Chip AHB Bus Tracer

5.4.4 Device utilization summary:

This device utilization includes the following.
» Logic Utilization
» Logic Distribution
» Total Gate count for the Design



Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	2,144	207,360	1%	
Number used as Flip Flops	2,144			
Number of Slice LUTs	3,442	207,360	1%	
Number used as logic	3,419	207,360	1%	
Number using O6 output only	2,309			
Number using O5 output only	690			
Number using O5 and O6	420			
Number used as exclusive route-thru	23			
Number of route-thrus	713	414,720	1%	
Number using O6 output only	713			
Slice Logic Distribution				
Number of occupied Slices	1,266	51,840	2%	
Number of LUT Flip Flop pairs used	4,121			
Number with an unused Flip Flop	1,977	4,121	47%	
Number with an unused LUT	679	4,121	16%	
Number of fully used LUT-FF pairs	1,465	4,121	35%	
Number of unique control sets	74			
IO Utilization				
Number of bonded IOBs	259	960	26%	

Figure 5.7 Design summary of On-Chip AHB Bus Tracer

The device utilization summery is shown above in which its gives the details of number of devices used from the available devices and also represented in %. Hence as the result of the synthesis process, the device utilization in the used device and package is shown above.

6.CONCLUSION:

The On-chip AHB bus tracer with Real-time Compression and Dynamic Multi-Resolution (SYS-HMRBT) was designed successfully and the coding was done in VHDL. The RTL simulations were performed using Modelsim from Mentor Graphics. The synthesis was done using Xilinx ISE. The On-chip AHB bus tracer with Real-time Compression and Dynamic Multi-Resolution works at a frequency of 198.515MHz. The SYS-HMRBTworks properly for all the Modes such as Mode FC, Mode FT, Mode BC, Mode BT, Mode MT .SYS-HMRBTTracer design verified for all test cases.The specification of the implemented SYS-HMRBT bus tracer has been implemented, RTL, FPGA,. The synthesis result with Xilinx Synthesis technology-XST is shown in Figure 5.7. The bus tracer costs only about 2144 slice registers which uses 2144 flip-flops , 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 one event register. More registers can be added if necessary. Compared with our previous work , the gate count is reduced. The reason is that this paper optimizes the ping-pong architecture by sharing most of the data path instead of duplicating all the hardware components.

As for the circuit speed, the bus tracer is capable of running at 198.515 MHz, which is sufficient for most SoC's with a synthesis approach under Xilinx Synthesis technology. If a faster clock speed is necessary, our bus tracer could be easily partitioned into more pipeline stages due to its streamlined compression/packing processing flow.

7.FUTURE SCOPE:

As future work:

» This work can be improved by implementing it with Advanced RiscMachines(ARM) Processors.
» This design can also be used in all System-On-A-Chip SoC applications where debugging and performance analysis is difficult

References:

[1] ARM Ltd., San Jose, CA, “AMBA Specification (REV 2.0) ARMIHI0011A,” 1999.

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

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

[4] Infineon Technologies, Milipitas, CA, “TC1775 Tri-Coreusers manual system units,” 2001.

[5] ARM Ltd., San Jose, CA, “Embedded trace macrocel-larchitecture specification,” 2006.

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