Integrating MIPS Processors with CoreSight Debug

Application Note

Filename	:	Integrating MIPS Processors with CoreSight Debug
Version	:	1.0
Issue Date	:	05 Sep 2016
Author	:	Imagination Technologies Limited
Document No	:	MD01182

Table of Contents

1. Introduction

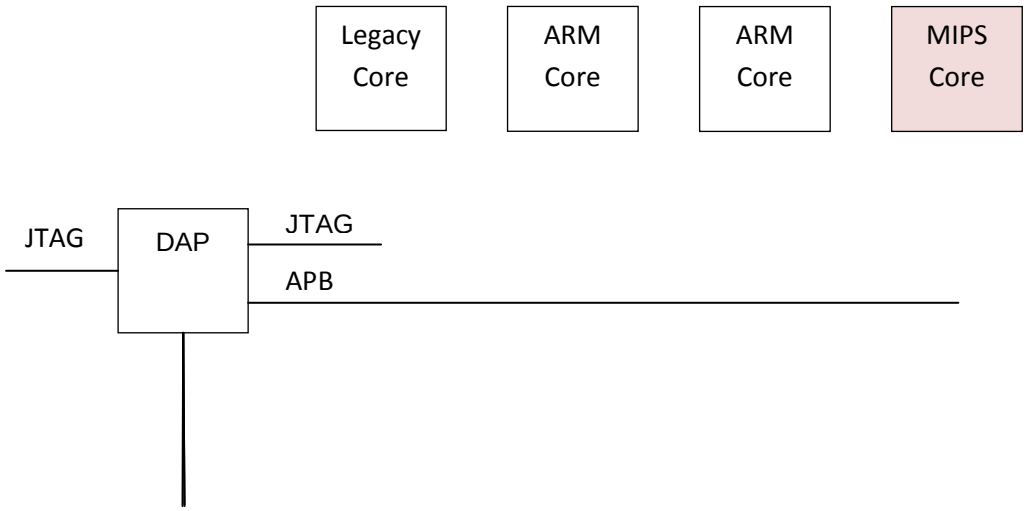
A system employing ARM processors and the associated CoreSight debug architecture can now include MIPS processors. This Application Note outlines the IP blocks and interconnections needed to integrate MIPS processors with a CoreSight system. There are three main areas to consider: debug control, trace collection, and cross-triggering.

2. Debug Control

CoreSight debug control is based around a Debug Access Port (DAP) consisting of one or more Debug Ports such as JTAG-DP and one or more Access Ports, including JTAG-AP and MEM-AP. A debug probe connects to the system via the JTAG pins and JTAG-DP. Using JTAG scans, the probe initiates actions on the JTAG-AP and MEM-AP interfaces that connect to the processors and other IP blocks within the system.

JTAG-AP is mainly intended for legacy IP cores that have only a JTAG debugging interface. Most recent processor cores hook up to the DAP using a parallel Memory Access Port (MEM-AP) interface. MEM-AP may implement AXI, AHB, or APB bus protocol. Bus bridge converters are available to adapt one bus protocol to another. See ARM document DDI0480G, the CoreSight Technical Reference Manual, for details.

Recent MIPS processors such as the M6200 series include an APB parallel bus debug interface. This interface should be connected to an APB-AP interface in the ARM Debug Access Port (DAP), or to another type of MEM-AP via a bus bridge.



Each core on the APB debug bus occupies 4K bytes of memory space. The upper few words (0xFCC – 0xFFF) are read-only identification registers that allow tools to identify the core type and revision. Lower words within the 4K space are used for controlling the core. The details of the register layout are found in the programmer's guide for the specific MIPS core, for example MD01093 for the M6200 Processor Core Family.

The APB debug port consists of the following signals:

APB Signal Name	Function
PCLKDBG	Rising edge of PCLKDBG times all transfers on the AMBA 3 APB interface
PRESETDBGn	Resets the APB interface and is active LOW.
PADDRDBG[11:2]	Bus indicates the address of the transfer.
PSELDBG	Output signal from DAP indicates that the slave device is selected and a data transfer is required. There is a separate PSELDBG signal for each slave.

APB Signal Name	Function
PENABLEDBG	Appendix A. Signal indicates the second and subsequent cycles of an AMBA 3 APB interface transfer.
PWRITEDBG	Output signal from Master; when HIGH this signal indicates a write access, and when LOW a read access.
PWDATADBG[31:0]	Write data bus is driven by the master during write cycles, when PWRITEDBG is HIGH. 32-bits wide.
PREADYDBG	This read signal is an input to Master driven by the slave to extend an AMBA 3 APB interface transfer.
PRDATADBG[31:0]	Read data bus driven by selected slave during read cycles, when PWRITEDBG is LOW. 32bits per slave. The MDH includes a multiplexer to select one of n PRDATADBG input sets when there is more than one device in the debug system.
PSLVERRDBG	Input to the Master is a signal returned by a slave in the second cycle of the transfer, indicating an error response. CoreSight components use this signal to indicate that the component is unavailable, for example because of powerdown.
DAPABORT	When high, requests an abort of any APB transaction in progress.

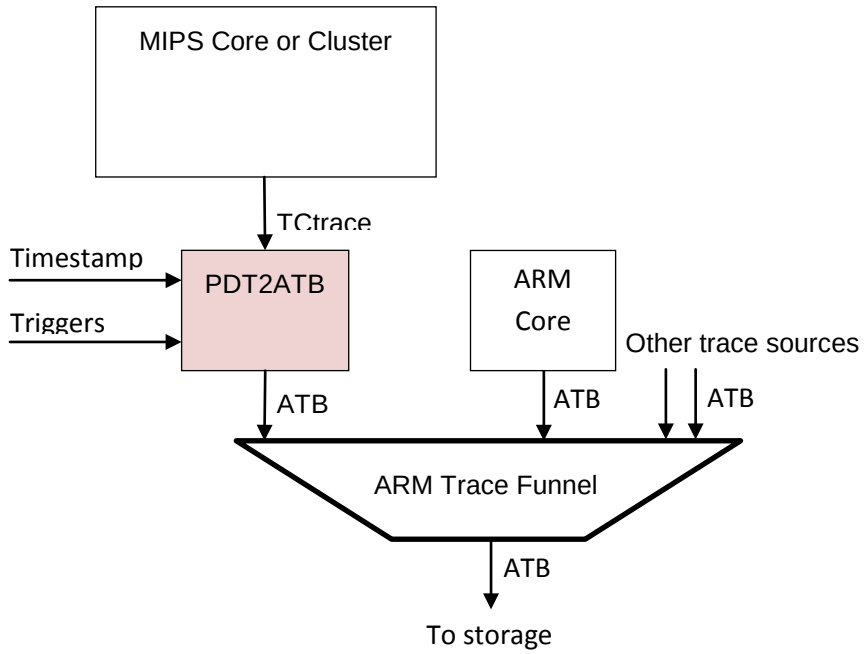
Additional MIPS core signals used in this App Note:

APB Signal Name	Function
SI_ClkIn	Clock input. All MIPS core inputs and outputs are sampled or asserted relative to the rising edge of this signal.
EJ_DINT	Debug exception is requested when a low-to-high transition is detected. Includes synchronization logic.
EJ_DebugM	Asserted when the core is in Debug Mode.
DBG_OUT	General-purpose debug output signals sourced by a register accessible from APB. If PDT2ATB is present, then DBG_OUT[6:0] should connect to tc_atid[6:0] to provide programmability on this input.
SI_Int[7:0]	Active High Interrupt pins. These signals may be connected to the cross-trigger interface to cause an exception to the core. Asynchronous to SI_ClkIn.
SI_IBS[7:0]	Hardware Instruction Breakpoint status. These outputs are asserted when the corresponding MIPS instruction breakpoint conditions are satisfied. The breakpoint may be programmed to stop execution or to just output this status.
SI_DBS[3:0]	Hardware Data Breakpoint status. These outputs are asserted when the corresponding MIPS Data breakpoint conditions are satisfied. The breakpoint may be programmed to stop execution or to just output this status.

3. Trace Collection

When implementing a system consisting of mixed ARM and MIPS processors, each with trace output capability, it is often desirable to combine the trace output from multiple cores into a single trace buffer or trace port. The ARM Advanced Trace Bus (ATB) (ARM doc IHI0032B) is a published interface specification for the ARM Trace Funnel. The PDtrace™ to ATB converter (PDT2ATB) receives a

MIPS PDtrace or iFlowtrace™ data stream and outputs an ATB data stream, allowing MIPS trace to be recorded in ARM trace buffers.



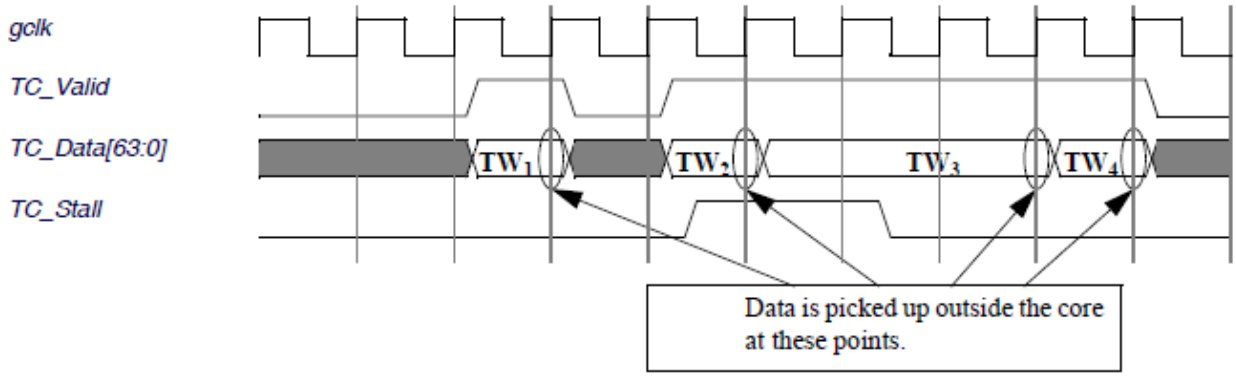
3.1. MIPS TCtrace Interface

PDT2ATB connects to the PDtrace or iFlowtrace system at the TCtrace interface point. This is a published intermediate interface used to send trace data from the MIPS trace funnel to a Probe Interface Block (PIB) or custom trace recording logic.

TCtrace Signal	I/O	Function
tc_clk	I	TCtrace interface clock. All signals in this interface are synchronous to this clock.
tc_trenable	I	Trace enable. When low, PDT2ATB is put in a low-power state.
tc_valid	I	Transfer is valid in this cycle
tc_stall	O	Request MIPS to stall sending
tc_data[63:0]	I	PDtrace data word
tc_pibpresent	O	Static output, set to 1'b1
tc_crmax[2:0]	O	Static output, set to 3'b100
tc_crmin[2:0]	O	Static output, set to 3'b100
tc_probewidth[1:0]	O	Static output, set to 2'b11 (64-bit transfer width)
tc_databits[1:0]	O	Static output, set to 3'b100 (64-bit transfer width)
tc_chiptrigin	O	Optionally requests MIPS to flush trace (PDtrace only)

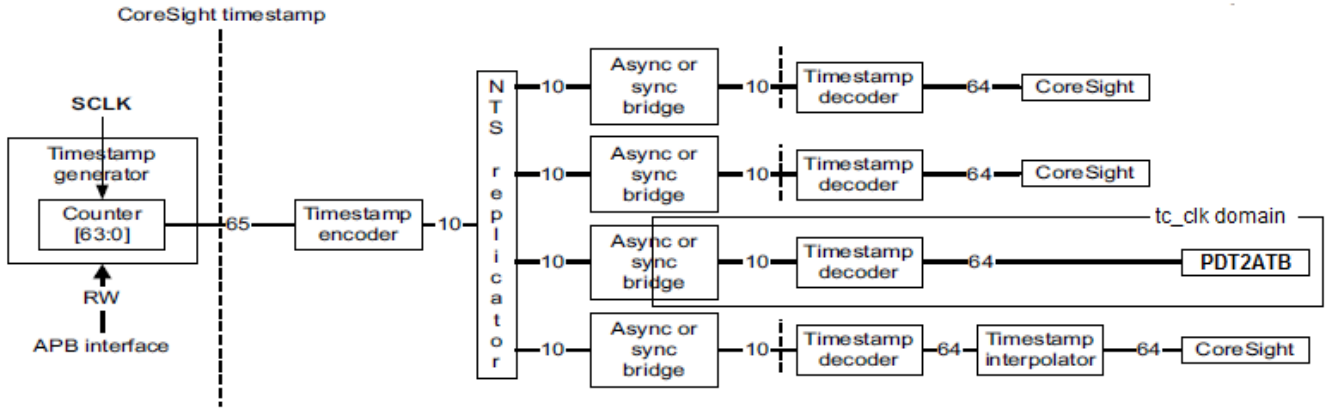
TCtrace Signal	I/O	Function
tc_probetrigin	O	Optionally requests MIPS to record a sync marker (PDtrace only)
tc_chiptrigout	I	Trigger signal from MIPS to trace system (PDtrace only). Tie low if not used.
tc_atid[6:0]	I	ID of this trace capture unit on ATB. This is normally connected to DBG_OUT[6:0] of the M6200 core so is programmable by debug software.
tc_timestamp[55:0]	I	Timestamp counter. Tie low if not used.
tc_timestamp_req	I	Low-to-high transition requests a timestamp trace record. Tie low if not used.

PDT2ATB accepts 64-bit trace data from TCtrace at up to one word per tc_clk cycle. Ordinarily, data is transferred on every cycle with tc_valid=1, but clock rate and handshake on the ATB may require stalling the TCtrace interface via the tc_stall signal. When tc_stall is asserted, transfer is stalled on the following clock edge. The timing of tc_valid and tc_stall is shown below. Sufficient buffering is included in PDT2ATB so that no data is lost.



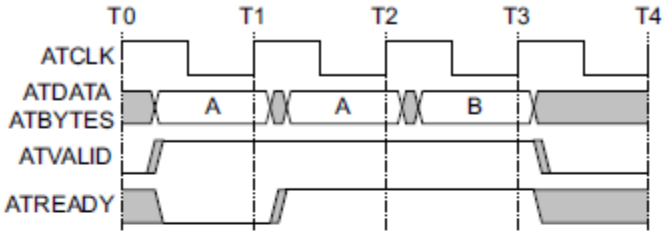
Several static signals configure the MIPS TCB in a compatible mode, namely 64-bit transfers controlled by tc_stall, and no options for trace clock ratio.

The timestamp in an ARM CoreSight system is generated from a single source and distributed as shown below. ARM provides a block called Narrow timestamp asynchronous bridge for crossing clock domain boundaries. One of those should be used to generate the tc_timestamp input, which must be synchronized to tc_clk. It is also the responsibility of the SoC designer to generate tc_timestamp_req to request a timestamp record to be inserted into the ATB trace stream when desired. This signal is in the tc_clk domain. If the timestamp function is not used, then the corresponding inputs should be tied to 0.



3.2. Advanced Trace Bus Interface

PDT2ATB connects to the ARM Trace Funnel via the ATB standard bus (technically, AMBA 4 ATB Protocol v1.1). ATB consists of 64 trace data bits and 7 additional bits for Source ID. There is a handshake protocol that operates on a cycle-by-cycle basis. Data transfer occurs if atvalid=1 indicating there is data waiting to transmit and atready=1 indicating the downstream slave can accept that data.



Clock cycle	State
T1	Stalled, **ATREADY** not asserted
T2	A accepted
T3	B accepted
T4	Ignored, not valid

ATB Signal	I/O	Function
atclk	I	ATB interface clock. All signals in this interface are synchronous to this clock.
atclken	I	ATB enable. When low, all other inputs in this interface are ignored.
atreset_n	I	ATB reset. This signal is asserted asynchronously, but negated synchronously to atclk. Resets entire block.

ATB Signal	I/O	Function
atbytes[2:0]	O	Number of bytes of valid trace data present on atdata, minus 1. This field can take on two different values in this design. 0 = trace trigger present on atdata[7:0] 7 = 64-bit trace word is present on atdata[63:0]
atdata[63:0]	O	ATB trace data word.
atid[6:0]	O	The Source ID of the current trace data word. This field can take on one of two different values in this design. Value driven into this block on tc_atid. Values of 00 and 70-7F are reserved and may not be used. 7D, indicating a trace trigger.
atready	I	Downstream slave is ready to accept data.
atvalid	O	Data valid for transfer. Transfer occurs if atvalid=1 and atready=1.
afvalid	I	Downstream slave is requesting a buffer flush.
afready	O	Informs slave that buffer flush is complete.
syncreq	I	Downstream slave is requesting a synchronization record to be written into the trace stream.

3.3. PDtrace ⇔ ATB Triggering Functions

The ATB spec includes three triggering functions that MIPS systems with PDtrace can accommodate to some extent. In each case, the TCB Triggering system is used to implement the MIPS behavior, and it should be noted that the behavior can therefore be changed and/or triggering functions disabled by programming the TCB Triggers differently. Note that none of these three triggering functions is available with MIPS iFlowtrace.

1. **AFVALID (Trace Flush).** This signal from ATB requests that the trace generator flush any partially-filled trace words. It is generally used when trace is being turned off to ensure all collected data has been written to memory. This function is handled on the MIPS side as follows:

- PDT2ATB asserts tc_chiptrigin.
- A MIPS TCB Trigger with Type=01 (Trace End) may be programmed to fire on CHTri.
- MIPS clears TCBCONTROLB.EN and flushes out remaining trace words.
- In a multi-core system, tc_chiptrigin feeds into all cores and TCB triggers can be set up in each core to flush that core's last partial trace word.
- PDT2ATB waits a fixed delay to allow for propagation of tc_chiptrigin, then waits for tc_valid=0, then asserts AFREADY to the ATB to indicate that all trace has been flushed.

2. **SYNCREQ (Sync Message Insertion).** This signal from ATB is asserted for one cycle to request that the trace generator insert an implementation-dependent synchronization marker. This function is handled on the MIPS side as follows:

- PDT2ATB asserts tc_probetrigin.
- A MIPS TCB Trigger with Type=11 (Trigger Info) and Trace=1 may be programmed to fire on PDTri.
- In a multi-core system, tc_probetrigin feeds into all the cores and the TCB trigger may be programmed in one or more than one core.
- The MIPS TCB responds by inserting a TF6 message into the trace stream with the TCBinfo field set to the value in the TCB trigger.

3. **Trace Trigger**. This function is defined in ATB as a way for the trace generator (MIPS) to signal downstream trace logic that a significant, system-dependent event has occurred. This function is handled on the MIPS side as follows:

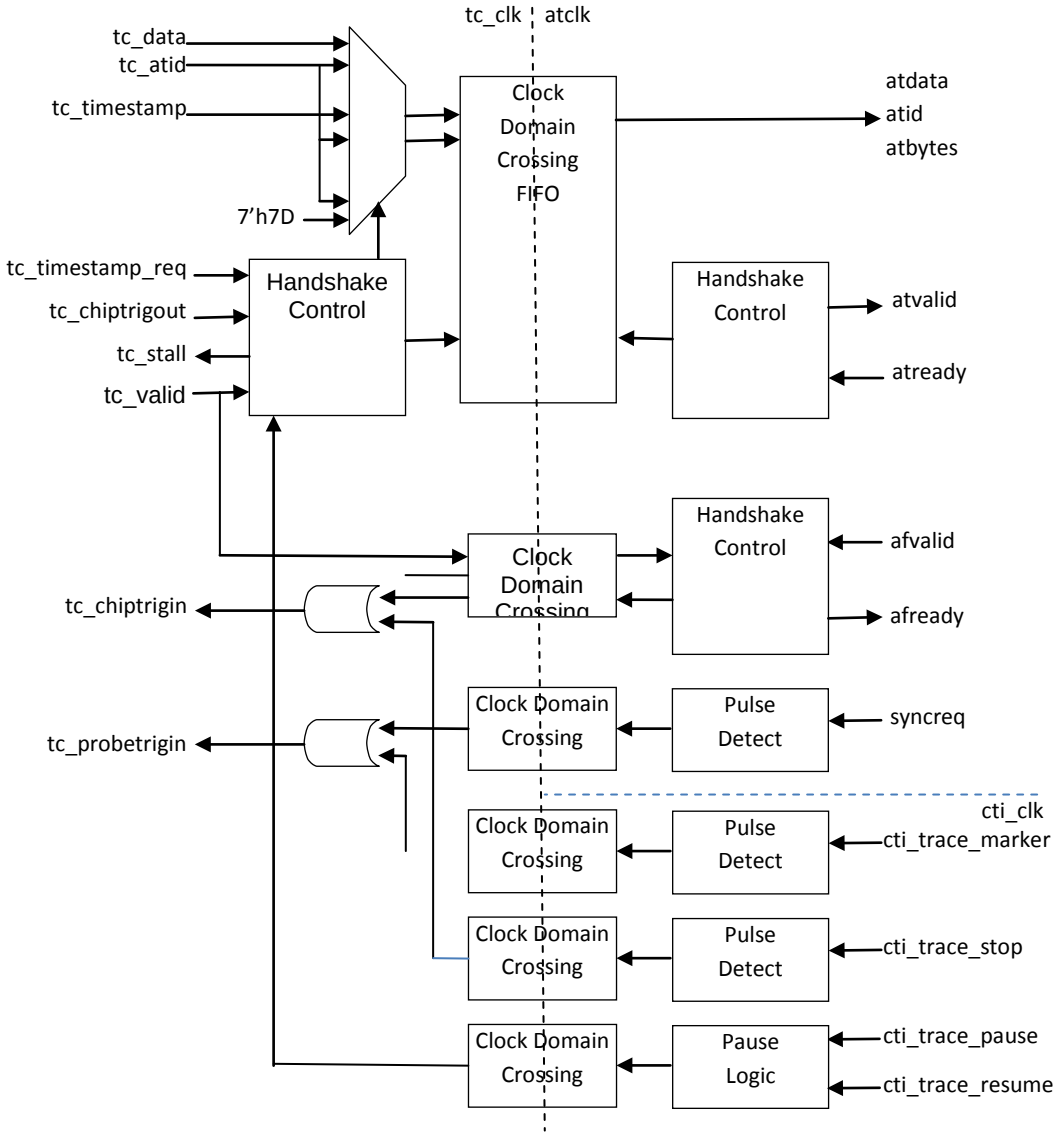
- A MIPS TCB Trigger with Type=11, Trace=0, and CHTro may be programmed to fire on DM (entry to debug mode).
- In a multi-core system, tc_chiptrigout is OR'ed from all the cores. It may be desirable to program the TCB trigger in just one core or in all the cores.
- PDT2ATB monitors tc_chiptrigout and when a pulse occurs, generates a trace message on the ATB with ATBYTES=0, ATDATA=source ID, and ATID=0x7D indicating a trace trigger.

3.4. Cross-Trigger Interface Triggering Functions

The ARM CoreSight Cross-Trigger Interface (CTI) includes a number of programmable inputs that can be programmed to cause events on a selectable subset of outputs. PDT2ATB includes event inputs for up to 4 different actions generated by the CTI. If not all are used, then the unused inputs should be tied low.

CTI Signal	I/O	Function
cti_clk	I	CTI interface clock. All signals in this interface are synchronous to cti_clk.
cti_trace_stop	I	A low-to-high transition on this signal causes PDT2ATB to perform the same action as ATFLUSH, above. Once trace is stopped, it cannot be restarted except by debug software. Available with PDtrace only.
cti_trace_pause	I	A low-to-high transition on this signal causes PDT2ATB to stop accepting trace words from TCtrace, empty any internal FIFOs, then insert a "gap" trace marker into the ATB trace stream. If trace is already paused or stopped, then this signal has no effect.
cti_trace_resume	I	A low-to-high transition on this signal causes PDT2ATB to resume accepting words from TCtrace. If trace is already running, then this signal has no effect.
cti_trace_marker	I	A low-to-high transition on this signal causes PDT2ATB to perform the same action as SYNCREQ, above. Available with PDtrace only.

3.5. PDT2ATB Implementation



Incoming data and trace triggers are monitored by the Handshake Control block in the tc_clk domain. If a trace trigger or timestamp request occurs, it has priority. The following formats of trace words are generated:

Word type	atbytes	atid	atdata	
Normal trace	7	tc_atid	tc_data[63:0]	
Trace trigger	0	7'h7D	57'b0	tc_atid[6:0]
Timestamp	7	tc_atid	tc_timestamp[55:0]	8'h40
Gap	7	tc_atid	56'hFF_FFFF_FFFF_FFFF	8'h80

The handshake module manages tc_stall if the FIFO fills up or if a slot is needed for a trace trigger or timestamp. Note that all ATB trace records generated by PDT2ATB use the same Source ID (tc_atid) even if the MIPS stream itself is made up of aggregated streams from more than one MIPS source. The particular MIPS core that generated the trace word is included in the data portion of the trace record. The entire aggregated MIPS trace (all records with ID=tc_atid) is passed to the MIPS dequeuer, which then separates MIPS trace into individual threads. When parsing a PDtrace or

iFlowtrace capture, no normal trace words end with 4'b0000, so a captured word with that encoding is known to software to be a special trace word inserted by PDT2ATB.

Outgoing data is popped off the FIFO when the downstream slave is ready to receive it. As discussed above, atdata, atid, and atbytes carry the data record and it is assumed that the slave accepts the record if atvalid=1 and atready=1.

The trace flush trigger is requested by afvalid. Through a clock domain crossing block, tc_chiptrigin is asserted and logic in the flush handshake control then determines when all trace has been flushed using a combination of a timer and tc_valid. Once flushed, afready is asserted.

The sync request input is monitored on every atclk cycle and when detected, causes a pulse on tc_probetrigin. Similarly, cti_trace_marker is monitored on every cti_clk cycle. Programming within the MIPS TCB causes a specific trace record to be inserted into the MIPS trace stream and this will eventually pass through the FIFO and be written to trace memory.

4. Cross-Triggering

Some of the cross-triggering features are detailed above in the discussion of Trace Collection. There are additional optional cross-trigger inputs and outputs available on the M6200 core that can be connected to the ARM CoreSight Embedded Cross Trigger Cross-Trigger Interface (CTI) if desired. The full list of potential CTI trigger inputs and outputs along with the clock domain in which they are defined may be found in the table below.

CTI Trigger Output	Clock	Function
cti_trace_stop	cti_clk	A low-to-high transition on this signal causes PDT2ATB to stop trace collection. Once trace is stopped, it cannot be restarted except by debug software. Available with PDtrace only.
cti_trace_pause	cti_clk	A low-to-high transition on this signal causes PDT2ATB to stop accepting trace words from TCtrace, empty any internal FIFOs, then insert a "gap" trace marker into the ATB trace stream. If trace is already paused or stopped, then this signal has no effect.
cti_trace_resume	cti_clk	A low-to-high transition on this signal causes PDT2ATB to resume accepting words from TCtrace. If trace is already running, then this signal has no effect.
cti_trace_marker	cti_clk	A low-to-high transition on this signal causes PDT2ATB to insert a trace marker into the compressed trace stream. Available with PDtrace only.
SI_Int[7:0]	async	These are interrupt inputs of the MIPS processor and may be programmed to cause a MIPS interrupt when a CTI event occurs.
EJ_DINT	async	Request MIPS processor to enter debug mode (stop execution).

CTI Trigger Input	Clock	Function
SI_IBS[7:0]	SI_ClkIn	Hardware Instruction Breakpoint status. These outputs are asserted when the corresponding MIPS instruction breakpoint conditions are satisfied. The breakpoint may be programmed to stop execution or to just output this status.
SI_DBS[3:0]	SI_ClkIn	Hardware Data Breakpoint status. These outputs are asserted when the corresponding MIPS Data breakpoint conditions are satisfied. The breakpoint may be programmed to stop execution or to just output this status.

CTI Trigger Input	Clock	Function
EJ_DebugM	SI_ClkIn	This signal indicates that the MIPS processor has entered debug mode (due to breakpoint or asynchronous halt command). Feeding this into the CTI allows this to cause a CTI system-wide event.