CoreSight Trace Memory Controller

The new CoreSight Trace Memory Controller provides SoC designers with more design options for the trace infrastructure. TRACE32 already has support for the first designs which use the TMC.

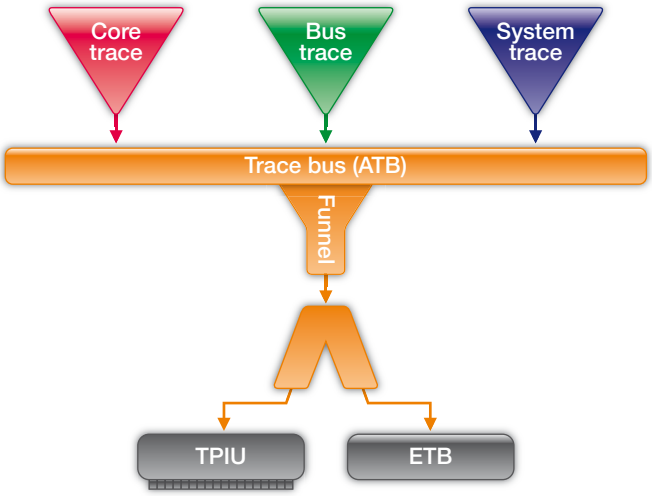

Fig. 1: CoreSight Funnel combines all trace data produced by trace macrocells into a single data stream.

Through CoreSight, the diagnosis data needed for the analysis of SoC-internal processes is produced by 'trace macrocells'. There are three types of trace macrocells:

- **Core trace macrocells** are assigned to a core and generate trace information about the instructions processed by that core. Information about process switches and load/store operations is generated depending on the design of the trace cell.
- **Bus trace macrocells** are firmly assigned to a bus and generate trace information on data transfers that occur on the bus.
- **System trace macrocells** generate trace information for hardware trigger (system event tracing) or provide diagnostic information produced by code instrumentation of the application software.

The CoreSight Funnel combines all of the trace data into a single data stream (see figure 1). This trace data stream is then either stored in an on-chip memory buffer (ETB) or exported to an external tool using a trace port (TPIU). The IP for CoreSight trace being implemented today is sometimes pushed to the limit when dealing with complex multicore SoCs that contain many trace macrocells.

ARM CoreSight

With CoreSight, ARM makes available an extensive set of IP blocks, which enables SoC designers to build a custom debug and trace infrastructure.

A single debug interface is enough to control and coordinate all cores of the SoC, as well as access all memory.

One trace interface is sufficient for providing diagnostic data about the processes occurring within the SoCs without any impact on real-time performance.

- **ETB:** The on-chip trace memory is often too small to record enough trace data for any meaningful future analysis. The typical size for the ETB is still between 4 and 16 KByte.
- **TPIU:** System states may occur where more trace data is being generated than the trace port can output. The CoreSight design is such that trace data from the trace macrocells is only taken over if the trace data can be exported by the TPIU. If the trace data generated remains in the trace macrocells for too long, the FIFOs there can overflow and important data may be lost.

The new CoreSight Trace Memory Controller should provide a solution for both of the above scenarios.

TMC as Embedded Trace Buffer

To be able to store more trace data on-chip for later analysis, the chip manufacturer can theoretically connect up to 4 GByte of SRAM to the Trace Memory Controller (see figure 2).

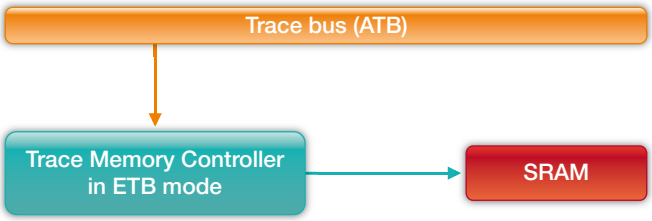

Fig. 2: In ETB mode, the Trace Memory Controller can make up to 4 GByte of on-chip trace memory available.

TMC as Embedded Trace FIFO

Inspections of the trace data streams being exported by the TPIU have shown that the bandwidth of most trace ports is large enough for normal operation. Overload, and therefore loss of trace data, only happens when peaks occur.

The Trace Memory Controller can be integrated into the trace infrastructure of the SoCs, so that the Trace Memory Controller acts as an Embedded Trace FIFO and

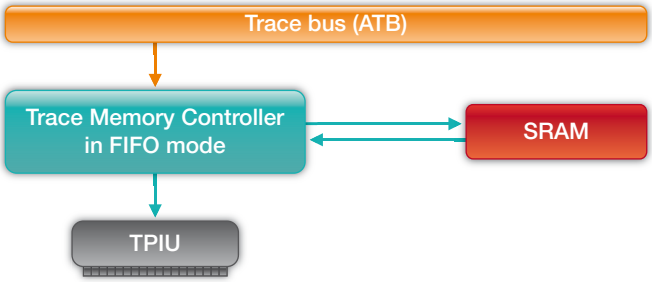

Fig. 3: In FIFO mode, the Trace Memory Controller can cushion load peaks on the TPIU. By doing this, trace data loss can be avoided.

cushions peaks in the load on the TPIU (see figure 3). This ETF is designed so that no trace data loss can occur. The size of the ETF can be freely defined from 512 Bytes to 4 GBytes.
Both integrations of the Trace Memory Controller in the trace infrastructure depicted are simple examples. Of course, you can build the TMC IP block into the CoreSight system in much more complex and flexible ways.

Modifications in TRACE32

As you would expect, Lauterbach has to modify the TRACE32 software for the configuration and handling of the Trace Memory Controller. This applies especially when the Trace Memory Controller is integrated in the SoC using new, previously unsupported ways. The TRACE32 user only needs to configure the basic address for the TMC. Then all the proven trace display and analysis features can be used as usual.

TMC as Router to High-Speed Link

The idea of moving away from dedicated trace ports has long been discussed within the embedded community. There are certainly several good arguments for this move. For the first time CoreSight traces can now connect to a high-speed standard interface by using the Trace Memory

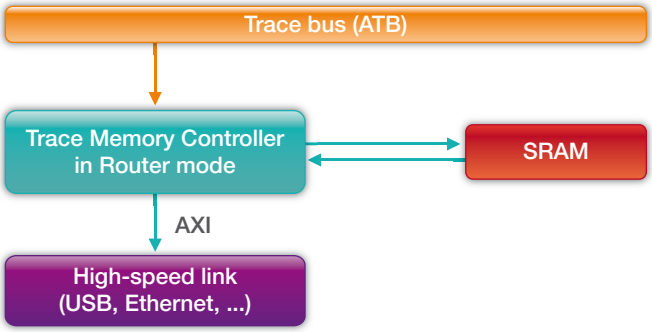

Fig. 4: In Router mode, the Trace Memory Controller forwards the trace data for the export to a high-speed standard interface.

Controller. USB or Ethernet interfaces are common favorites, especially as they are available in many end products. Ideally, the external trace tool will share the interface with the other connected devices.

Within the SoC, the TMC operates as Embedded Trace Router and has the task of passing on the trace data through the AXI bus for the export to the IP of the high-speed interface (see figure 4).

This new method of trace export will need completely new trace tools. Lauterbach is currently in close contact with leading semiconductor manufacturers to develop the appropriate tools for this switch in technology.

TRACE32 CoreSight Features

- Open for use with all cores which can be integrated into CoreSight; Lauterbach offers debug solutions for all ARM/Cortex cores and for numerous DSPs, as well as for configurable cores.
- Support for asymmetric multiprocessing (AMP) and symmetric multiprocessing (SMP)
- Debugging via JTAG interface and 2-pin Serial Wire Debug
- Synchronized debugging of all cores
- Support for the CoreSight Cross Trigger Matrix
- Support for all types of trace macrocells (ETM, PTM, HTM, ITM, STM, and more)
- Tools for parallel and serial trace ports
- Multicore tracing