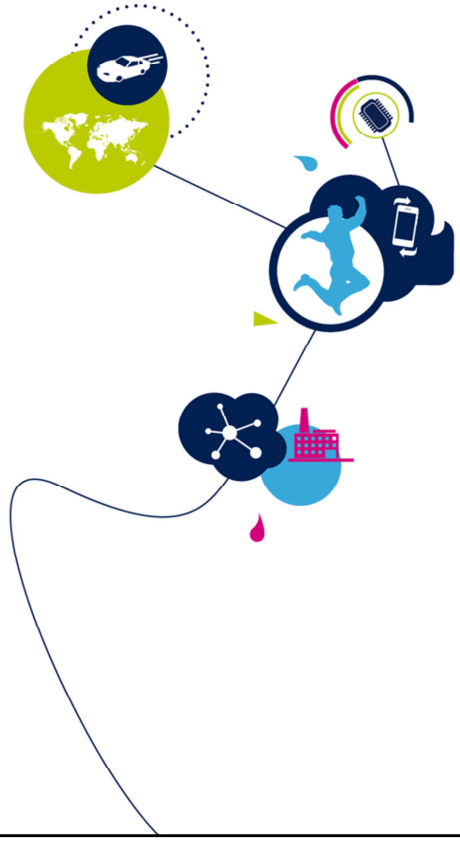
STM32H7 - DBG

Debug and trace

Revision 1.0



life.augmented

Hello, and welcome to this presentation of the STM32 debug and trace interface. It covers the debug and trace capabilities offered by STM32H7 devices.

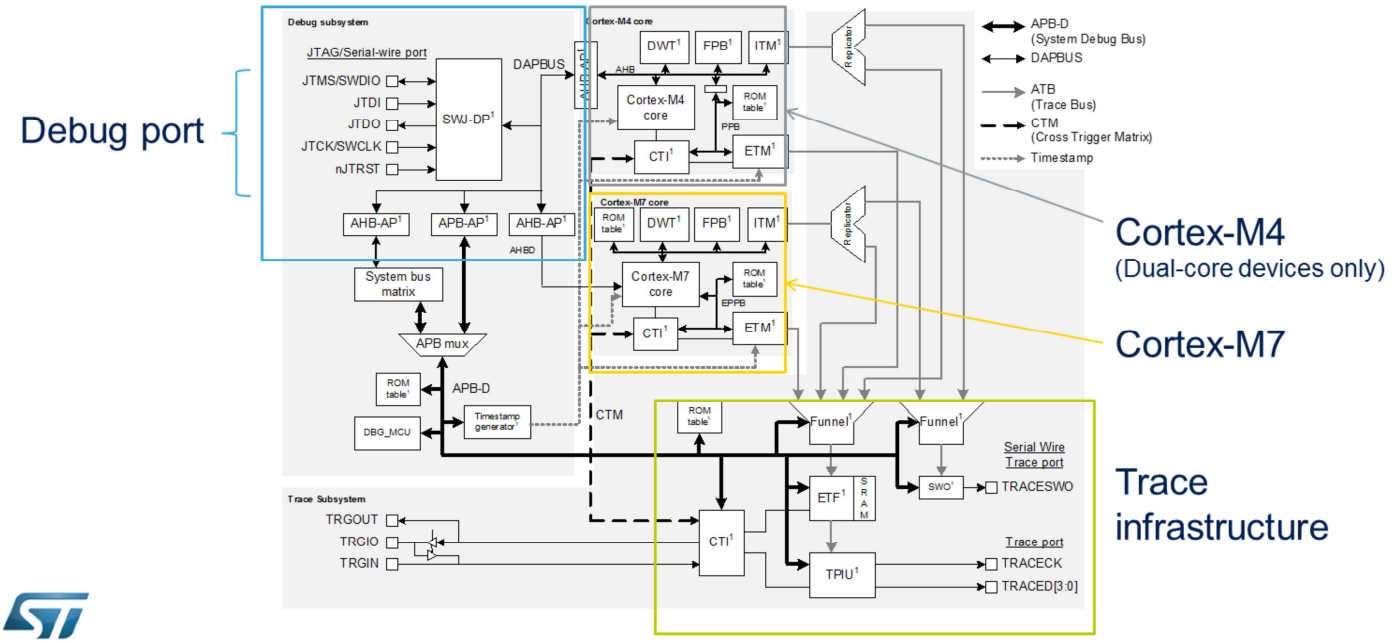
Overview

- STM32H7 provides rich support for debug and trace
  - Download programs into RAM or Flash memory
  - Examine memory and register contents
  - Insert breakpoints and halt the processor
  - Run or Single-step through programs
  - Trace program execution
- Based on ARM® CoreSight™ architecture
  - Wide range of compatible tools
  - Standard interface (JTAG/serial wire)

life.augmented

The STM32H7 incorporates all the familiar debug capabilities provided by the STM32 family of MCUs – flash download, breakpoint debugging, register and memory view, serial wire trace – and adds high bandwidth instruction trace as well as cross-triggering capabilities in multi-core versions of the STM32H7 family. The debug and trace infrastructure uses the ARM® CoreSight™ standard, well supported by most tool providers.

Debug architecture



The debug and trace infrastructure is composed of four distinct functional domains:

- Debug access infrastructure – includes the debug port (SWJ-DP) and access ports (AP) which allow access by an external debugger to the target's trace and debug features.
- Trace infrastructure – includes the serial (SWO) and parallel (TPIU) trace ports, the trace FIFO (ETF) used for smoothing the trace flow, and the trace funnels which combine the trace from each source into a single flow.
- Cortex-M7 core – includes the processor and associated trace and debug units (DWT, FPB, ITM, and ETM)
- Cortex-M4 core (dual-core devices only)

In addition, there are system debug features including:

- Cross trigger interfaces and matrix (CTI, CTM) – these allow simultaneous halting of both cores, triggering of trace, etc.
- Global timestamp generator – provides a common time reference for the different trace sources
- DBG_MCU – provides proprietary features such as freezing of

timers during debug
• External trigger input/output – allows an external signal to trigger debug or trace, or generates a trigger pulse for synchronizing external equipment or components

Debug port

- The debugger accesses the STM32H7 via the JTAG/SWD debug port
  - Standard 5-pin JTAG port also used for boundary scan and DFT
  - Serial Wire Debug (SWD) port uses only 2 of the JTAG port pins
  - When debug is not required, all debug pins can be reallocated for functional use

Available debug ports	PA13	PA14	PA15	PB3	PB4
Full JTAG/SWD*	JTMS	JTCK	JTDI	JTDO	NJTRST
Full JTAG/SWD without nJTRST	JTMS	JTCK	JTDI	JTDO	
JTAG-DP disabled, SW-DP enabled	SWDIO	SWCLK			
Both JTAG-DP and SW-DP disabled					

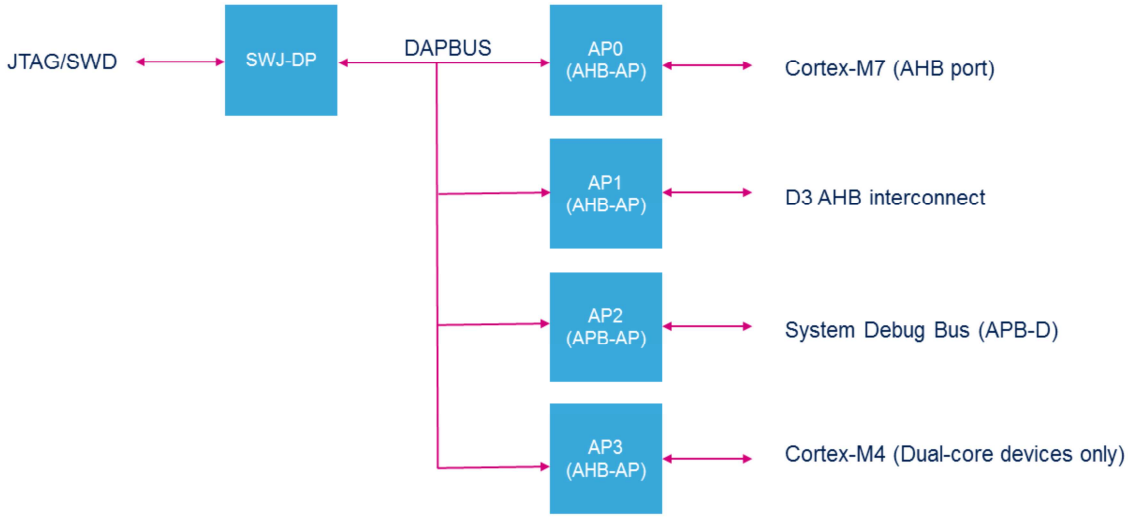
* Reset state

life.augmented

The minimum configuration for debug requires pins PA13 and PA14 to be allocated to serial wire debug (SWDIO and SWCLK respectively).
Serial wire debug uses a special serial code driven by the debugger on the SWDIO (JTMS) input. This is recognized by the SWJ-DP which switches to SWD mode (after reset JTAG mode is configured by default).
ST-Link, and most 3rd party debug adaptors (for example, Ulink), support serial wire debug.

Access ports

- Four access ports (AP) act as bus masters allowing the debugger to perform read/write transactions to memory and registers

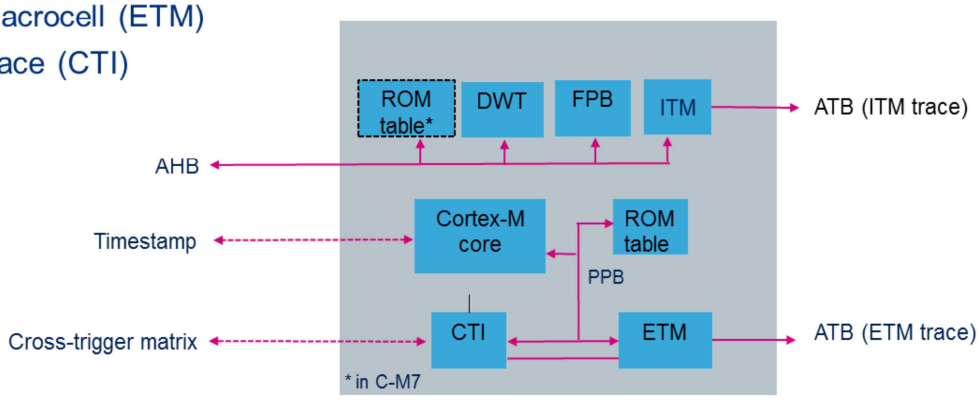


life.augmented

AP0: Allows access to the debug and trace features integrated in the Cortex-M7 processor core via an AHB-Lite bus connected to the AHBD port of the processor.
AP1: Allows access to the AHB bus matrix in the D3 domain. This gives visibility of the D3 domain memory and peripherals when the D1 and D2 domains are switched off.
AP2: Allows access to the debug and trace features on the system APB debug bus, that is, all components not included in one of the processor cores.
AP3: (Dual-core devices only) Allows access to the debug and trace features integrated in the Cortex-M4 processor core via its internal AHB bus.

Cortex-M T&D features

- The Cortex-M cores contain the following debug components:
  - System Control Space (SCS)
  - Data Watchpoint and Trace Unit (DWT)
  - Breakpoint Unit (FPB)
  - Instrumentation Trace Macrocell (ITM)
  - Embedded Trace Macrocell (ETM)
  - Cross-Trigger Interface (CTI)
  - ROM tables



life.augmented

ETM

All debug related registers in the Cortex-M7 core are accessed via the dedicated AHB access port AP0.
The ROM tables contain pointers to the base addresses of each debug component visible from the AP. They are used by some debug tools to automatically detect the topology of the CoreSight™ infrastructure in the target.
The SCS (system control space) contains the registers for controlling the processor core while in debug mode.
The other units are described in the following slides.

Data Watchpoint and Trace Unit

7

- The DWT provides four comparators that can be used as:
  - Watch point
  - ETM trigger
  - PC sampling trigger
  - Data address sampling trigger
  - Data comparator
  - Clock cycle counter comparator
- It also contains counters for software profiling:
  - Clock cycles
  - Folded instructions
  - Load Store Unit (LSU) operations
  - Sleep cycles
  - Number of cycles per instruction
  - Interrupt overhead

life.augmented

A Data Watchpoint (DWT) comparator compares one of the following with the value held in its DWT_COMP register:

- A data address
- An instruction address
- A data value
- The cycle count value, for comparator 0 only.

For address matching, the comparator can use a mask, so it can match a range of addresses.

On a successful match, the comparator generates one of the following:

- One or more DWT Data trace packets, containing one or more of:
  - The address of the instruction that caused a data access
  - An address offset, bits[15:0] of the data access address
  - The matched data value.

- A watchpoint debug event, on either the PC value or the accessed data address.
- A CMPMATCH[N] event that signals the match outside the DWT unit.

Breakpoint Unit

8

- The FPB allows hardware breakpoints to be set.
  - It contains eight comparators which monitor the instruction fetch address and return a breakpoint instruction when a match is detected.
  - When the breakpoint instruction is executed, the processor halts in debug mode

life.augmented

In dual-core devices, the Cortex-M4 Breakpoint Unit (FPB) also supports Flash memory patching. This feature is intended for patching erroneous code by diverting execution to volatile memory at a given address.
In the Cortex-M7 Flash memory patching is not supported by the FPB.

Instrumentation Trace

- The ITM generates trace packets from four sources:
  - Software trace:
    - Software write to any of the 32 stimulus registers
  - Hardware trace packet from DWT
    - This may be a data trace event, a PC sample or a counter wrap-around
  - Local timestamp
    - A 21-bit counter in the ITM provides a timestamp for each trace packet, relative to the previous packet
  - Global timestamp
    - Timestamps can also be generated using the system-wide 64-bit timestamp from the TSGEN
- Trace packets are output on the ATB trace bus

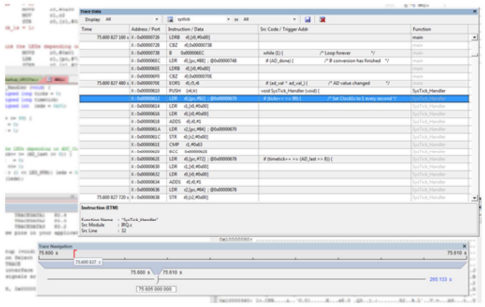
life.augmented

Software can write directly to any of 32 x 32-bit Instrumentation Trace Macrocell (ITM) stimulus registers to generate packets. The permission level for each port can be programmed. When software writes to an enabled stimulus port, the ITM combines the identity of the port, the size of the write access, and the data written, into a packet that it writes to a FIFO. The ITM outputs packets from the FIFO onto the trace bus. Reading a stimulus port register returns the status of the stimulus register (empty or pending) in bit 0.
If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. The sources are listed here in descending order of priority.
The timestamp generator (TSGEN) provides a 64-bit common time base for all trace packet timestamps. This allows the trace analyzer to align traces coming from different sources according to the time at which the trace was generated. The local timestamps are not synchronized, and can run at

different frequencies, making it impossible to know the precise timing at which a trace was generated.
Note: The Cortex-M4 only uses the 48 LSBs of the global timestamp. The Cortex-M7 uses all 64 bits.

Instruction Trace

- The ETM generates trace packets which allow the execution of the software to be observed. The trace information includes:
  - The number of instructions executed in the same cycle
  - Changes in program flow
  - The current processor instruction state
  - The addresses of memory locations accessed by load and store instructions
  - The type, direction and size of a transfer
  - Condition code information
  - Exception information
  - Wait for interrupt state information
- Trace packets are output on the ATB trace bus

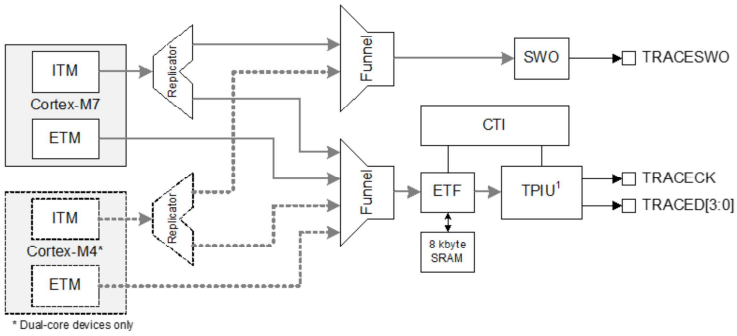


life.augmented

In the STM32H7, the Embedded Trace Macrocell (ETM) is configured for instruction trace only; i.e. data accesses are not included in the trace information.
Note that the ETM in the Cortex-M7 is quite different to the one in the Cortex-M3/M4.

Trace Infrastructure

- Trace information from the ITM and ETM is routed via the Amba Trace Bus (ATB) to one or more of the trace sinks
- The routing is performed by two ATB infrastructure components
  - Replicator – duplicates the trace packets on two ATB branches
  - Funnel – multiplexes several ATB branches onto one



life.augmented

The ITM and ETM both generate trace streams which are combined using the trace funnel. Some funnel parameters can be modified; for example, the number of bytes received on one input before switching to another – the less switching that occurs, the lower the overhead, but at the cost of increased latency. It is also possible to filter trace, for example the ITM trace can be removed from the TPIU (it can be output on the SWO instead).

Trace from the ITM (not the ETM) can be directed to the single wire trace port. In dual-core devices, the ITM trace from both cores can be directed to the SWO and combined in the SWO trace funnel. However since there is no formatting in the SWO, it is not possible for a trace port analyzer to separate the trace streams. Therefore it is recommended that the funnel be used to manually select one ITM at a time for output on the SWO. If both are needed the TPIU should be used.

Trace Sinks

- The trace packets are channeled to one of three destinations or “sinks”:
  - Embedded Trace FIFO (ETF)
    - This is an 8-Kbyte memory that can store trace packets in a circular buffer. The trace can be read out by software or by the debugger
  - Trace Port Interface (TPIU)
    - Trace packets are streamed out of the device via a 4-pin parallel port, accompanied by a synchronous clock signal. This requires connection of a trace port analyzer probe such as ULINKpro or DStream.
  - Single Wire Trace port (SWO)
    - ITM trace can be directed to the SWO and output using an asynchronous protocol (NRZ or Manchester). This can be read using the ST-Link or other adaptor and most commercial debugger tools.

life.augmented

The ETF can be used as a trace buffer for storing traces on-chip. The trace can be read by software, or by the debugger, or flushed via the trace port. If configured as a circular buffer, the trace will be stored continuously, so the most recent trace will overwrite the oldest. Alternatively, the FIFO full flag can be used to stop a trace when the buffer is full, and hence capture a trace at a particular point in time.
The ETF also acts (in hardware mode) to smooth the flow of trace to the TPIU. Since the trace stream tends to be bursty in nature, and the instantaneous bandwidth is much higher than that of the trace port, the buffer absorbs the peaks and regulates the flow to the trace port’s maximum continuous bandwidth.

Trace Ports

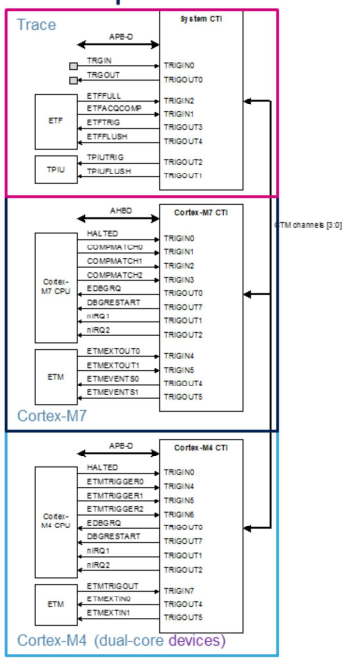
- TPIU parallel port
  - 1 to 4 data pins and a clock can be assigned to trace (GPIOs by default)
  - TRACECLK can operate at up to 133 MHz in DDR mode
  - ➢ 1Gbps maximum bandwidth (800 Mbps with ULINKpro)
- SWO single wire serial port
  - 1 asynchronous data pin multiplexed with JTDO
  - 100 Mbps bandwidth (Manchester encoded)

life.augmented

The trace port width can be programmed from 1 to 4 pins. The bandwidth scales proportionally to the number of pins and the TRACECLK frequency (selectable via a divider in the RCC). Full dual core instruction trace at maximum clock frequency is likely to require the maximum bandwidth. By applying filters and triggers to the trace sources (ETM notably), the average amount of trace data can be reduced, allowing a lower clock rate or reduced number of pins.
The TRACESWO pin is multiplexed with the JTDO signal, which is part of the JTAG interface. Hence single wire trace is only available when the serial wire debug (SWD) interface is enabled.

Cross Trigger Interface

- The CTI propagates trigger events to other debug and trace components.
  - Trigger event sources can include:
    - Data watchpoints
    - Hardware breakpoints
    - Profiling counter events
    - Trace buffer full/empty
    - External trigger signal
    - Processor halt/restart
  - Trigger events at the destination can cause:
    - Trace start/stop
    - Trace buffer flushing
    - Processor halt/restart
    - External trigger signal output
    - Processor interrupt



Cross triggering can be used in dual-core devices to halt both cores simultaneously. When one core hits a breakpoint, its “halted” output (indicating it has entered debug mode) propagates to the other core and causes it to enter debug mode as well. Similarly, both cores can restart simultaneously.
The cross-trigger feature can also be used to halt the processor with an external trigger signal (this might be an edge on one of the IO pins).
There is a Cross-Trigger Interface (CTI) dedicate to each of the Cortex-M processors, as well as a system CTI connected to the trace components (ETF, TPIU) and external trigger signals.
To use any of the cross-trigger features, the CTIs must be programmed accordingly by the debugger. The required trigger input signals (TRIGINn) and trigger output signals (TRIGOUTn) need to be connected to the cross-trigger matrix (CTM). The CTM comprises up to four channels allowing four different events to be propagated in parallel. Trigger inputs can be combined in the CTI so that any one of the combined inputs will cause an event on the

connected channel. Similarly, a channel can be connected to several trigger outputs, so that one event can trigger multiple actions.

DBGMCU

- The “MCU debug” block enables device-specific debug features
  - Device identity
    - Standard location for the reading the device identity code register
  - Emulation of low power modes
    - Maintains the power and clock when the device enters a low power mode (SLEEP, STOP, STANDBY) so that debug access is still possible
  - Peripheral clock “freeze” in debug mode
    - Freezes the RTC, TIM, LPTIM and watchdog (IWDG,WWDG) timer counters, as well as the SMBUS and FDCAN timeout counters, while the processor is halted
  - Domain debug clock enable
    - Disables the clocks to debug devices in each power domain when not required, to save power
  - External trigger direction
    - Controls the direction (input/output) of the bi-directional TRGIO external trigger pin

life.augmented

The DBGMCU is located on the debug APB bus and can be accessed by the debugger via the APB access port AP2. It is also accessible by the processor(s) in the debug APB address space.
The DBGMCU_IDC register provides the device ID and revision codes in STM32 standard format. The information is also available in the debug port (DP_TARGETID register – accessible only to an external debugger) and in the system debug ROM table registers (SYSROM_PIDR[2:0] – accessible also by software).
Low power mode emulation means that the debugger connection is not lost when entering low power mode. It eliminates the need to replace the low power entry command (for example, WFI/WFE) by a while() loop. On exit, the device is in the same state as if the emulation was not active (apart from any changes made by the debugger during the low power mode emulation).
Peripheral clock freeze is particularly useful to prevent a watchdog timeout from resetting the device while debugging, without having to re-arm the watchdog with the debugger. It also allows

timer values to be inspected and corresponding interrupts to be suspended until “normal” operation is resumed.
The debug clock enable bits ensure that the debug blocks are only clocked when needed. This avoids unnecessary power consumption, since apart from the DAP, all blocks are clocked with the ungated domain clock.
On certain packages, the TRGIN and TRGOUT pins are not available, only the bi-directional pin is used, and the direction must be chosen using the TRGOEN bit.