
Hello, and welcome to this presentation of the STM32 debug and trace interface. It covers the debug and trace capabilities offered by STM32WB devices.

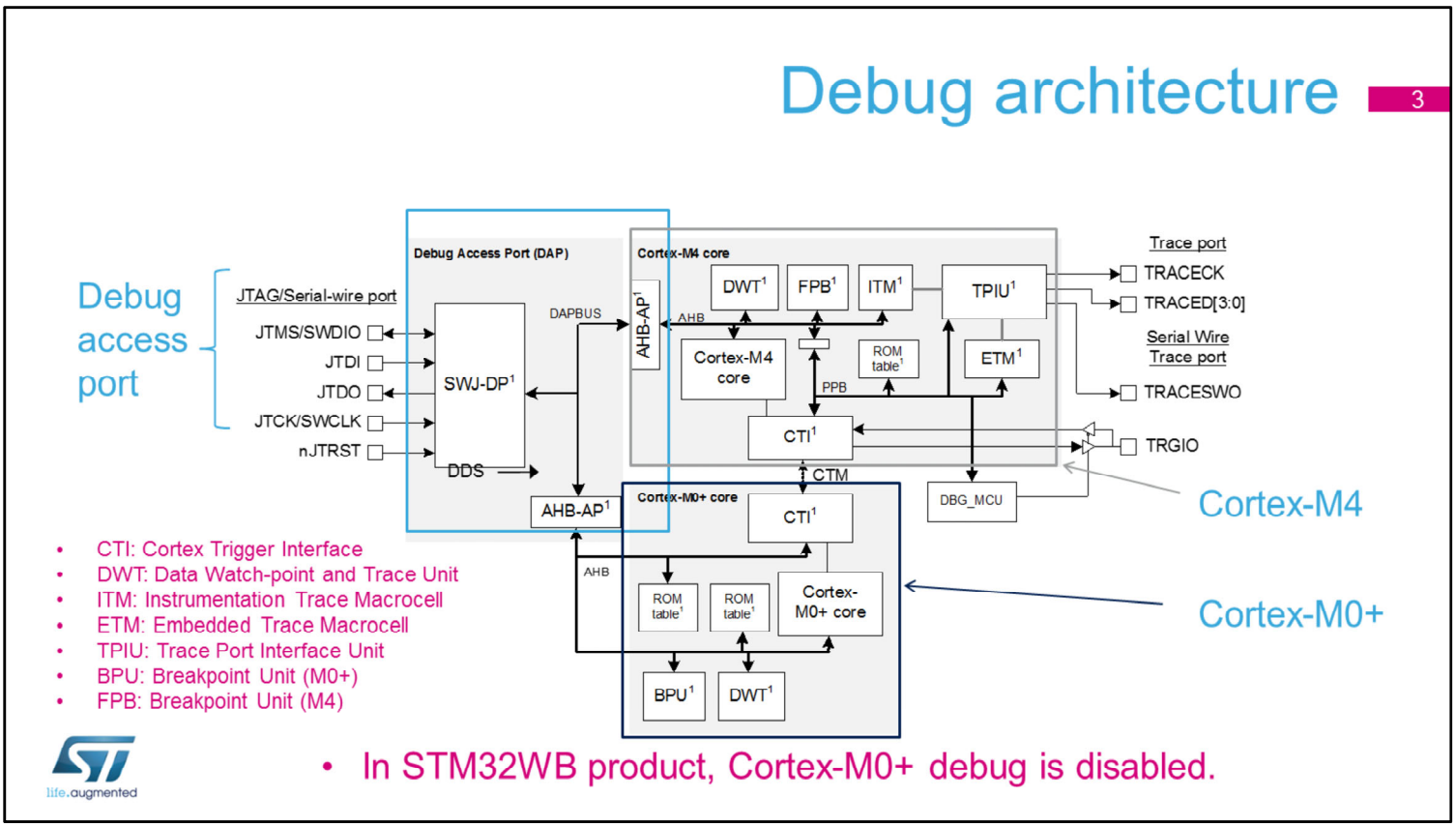
# Overview




- STM32WB 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 STM32WB 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 cross-triggering capabilities for co-debugging in multi-core versions. The debug and trace infrastructure uses the ARM CoreSight standard, well supported by most tools providers.




The debug and trace infrastructure is composed of three 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

• Cortex-M0+ core - includes the processor and associated trace and debug units (DWT, BPU, and CTI)

• Cortex-M4 core - includes the processor and associated trace and debug units (DWT, FPB, ITM, ETM, TPIU, and CTI)

In addition, there are system debug features including:

• Cross trigger matrix (CTM) – connects the CTIs to allow simultaneous halting of both cores, triggering of trace, etc.

• 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.

Note that in STM32WB devices containing the secure Root Security Services, the Cortex-M0+ debug unit is disabled.

# Debug port



- The debugger accesses the STM32WB via the JTAG/SWD 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

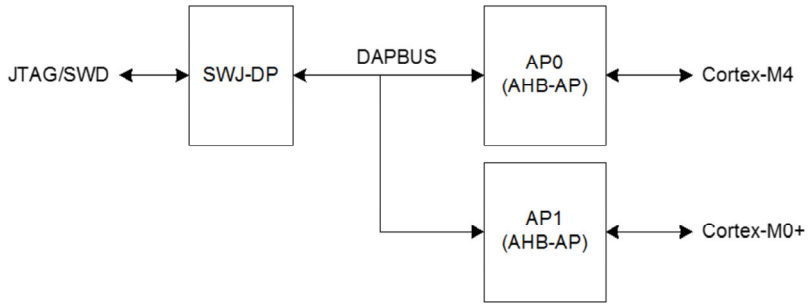
life.augmented

The minimum configuration for debug requires 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).
The ST-Link, and most 3rd party debug adaptors (eg. Ulink), support serial wire debug.

# Access ports



- Two access ports (AP) act as bus masters allowing the debugger to perform read/write transactions to memory and registers
- Enabled dual CPU debug



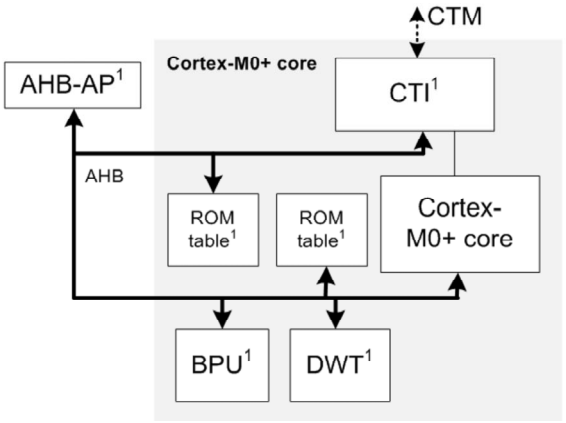

life.augmented

Access port AP0: Allows access to the debug and trace features integrated in the Cortex-M4 processor core via its internal AHB bus, as well as to the DBG_MCU.
Access port AP1: Allows access to the debug and trace features integrated in the Cortex-M0+ processor core via its internal AHB bus.
The single access ports enable dual CPU debugging.

# Cortex-M0+ T&D features



- The Cortex-M0+ core contains the following debug components:
  - System Control Space (SCS)
  - Data Watch-point and Trace Unit (DWT)
  - Breakpoint Unit (BPU)
  - Cross-Trigger Interface (CTI)
  - ROM tables

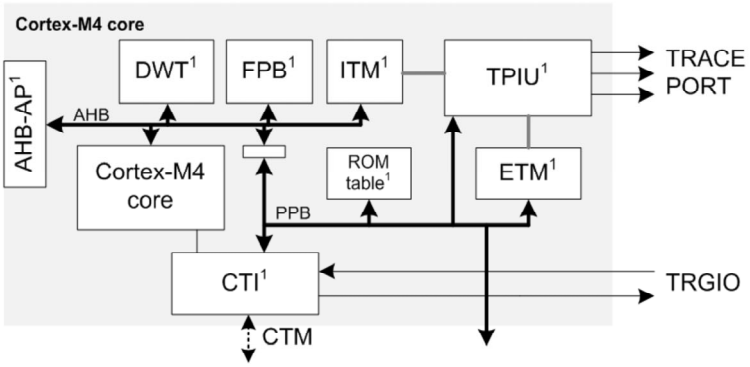



All debug related registers in the Cortex-M0+ core are accessed via the dedicated AHB access port AP1.
The ROM tables contain pointers to the base addresses of each debug component visible from the access port (AP). They are used by some debug tools to automatically detect the topology of the CoreSight 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.
Note that there is no support for trace on the Cortex-M0+ processor core.

# Cortex-M4 T&D features



- The Cortex-M4 core contains the following debug components:
  - System Control Space (SCS)
  - Data Watch-point and Trace Unit (DWT)
  - Breakpoint Unit (FPB)
  - Instrumentation Trace Macrocell (ITM)
  - Embedded Trace Macrocell (ETM)
  - Trace Port Interface Unit (TPIU)
  - Cross-Trigger Interface (CTI)
  - ROM table




All debug related registers in the Cortex-M4 core are accessed via the dedicated AHB access port AP0.
The ROM table contains pointers to the base addresses of each debug component visible from the access port (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 Watch point and Trace Unit

8

- The DWT provides four comparators that can be used as:
  - Watch point
  - ETM trigger (C-M4 only)
  - 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 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 matches 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 Units



- The FPB (Cortex-M4) and BPU (Cortex-M0+) allow hardware breakpoints to be set
  - They contain 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
- The FPB also supports flash memory patching.
  - This feature is intended for patching erroneous code by diverting execution to volatile memory.

life.augmented

The Cortex-M0+ processor core has 8 comparators allocated to instruction fetch address matching.
The Cortex-M4 processor core has 6 comparators for instruction fetch address matching, and 2 for literal load address matching (ie. data reads to code space). The latter can only be used for patching.

# Instrumentation Trace (C-M4 only)



- The ITM generates trace packets from three 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
- Trace packets are output via the TPIU or SWO trace port

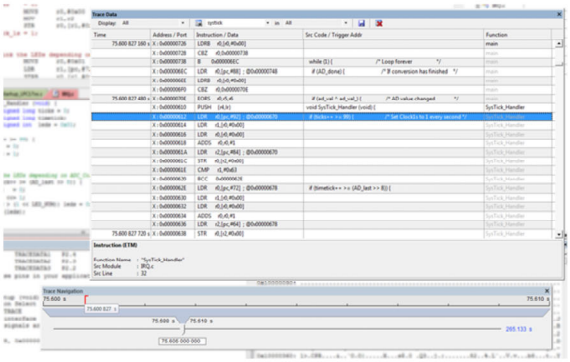
life.augmented

Software can write directly to any of 32 x 32-bit 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.

# Instruction trace (C-M4 only)



- 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 via the TPIU or SWO



life.augmented

In the STM32WB, the Embedded Trace Macrocell (ETM) is configured for instruction trace only; i.e. data accesses are not included in the trace information.

# Trace Sinks (C-M4 only)



- The trace packets are channeled to one of two destinations or “sinks”:
  - 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 analyser 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 trace port interface (TPIU) is only available from STM32WB packages supporting the parallel trace port. The single wire trace port (SWO) is available from all STM32WB packages.

# Trace Ports (C-M4 only)



- TPIU parallel port
  - 1-4 data pins and a clock can be assigned to trace (GPIOs by default)
  - TRACECLK can operate at up to 32MHz in DDR mode
  - ➢ 256Mbps maximum bandwidth
- SWO single wire serial port
  - 1 asynchronous data pin multiplexed with JTDO
  - 100Mbps bandwidth (Manchester encoded)

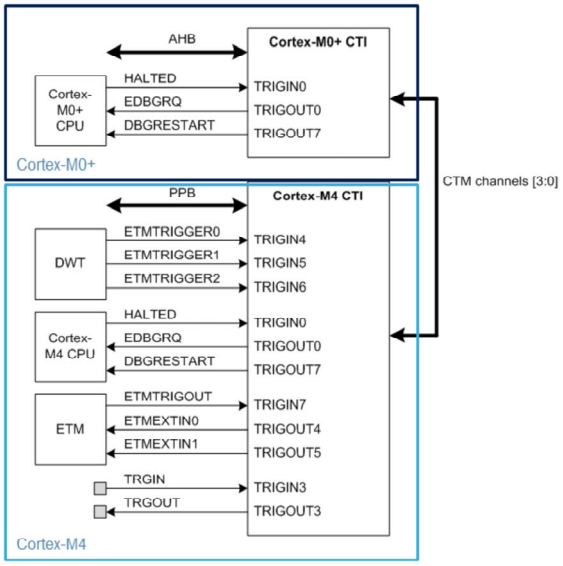
life.augmented

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).
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) in each of the Cortex-M processors.

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 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 timeout counter, while the processor is halted
  - External trigger direction
    - Controls the direction (input/output) of the bi-directional TRGIO external trigger pin

life.augmented

The DBGMCU_IDC regThe DBGMCU is located on the C-M4 PPB bus and can be accessed by the debugger via the AHB access port AP0. It is also accessible by the Cortex-M4 software.
ister 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).
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 (eg. 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 TRACECLKEN bit ensures that the trace port output is only clocked when needed. This avoids unnecessary power consumption.
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.