Specification of RISC-V Trace Connectors

Version 0.9.1, May 10, 2022: This document is in Stable state. Assume it may change.

Table of Contents

Preamble

History and status

STATUS: After approval be the group to be changed from Stable to Frozen

2022/5/10: v0.9.1: Created separated PDF (from original ADOC) and adjusted width of all table columns

2022/4/11: v0.9.0: Consolidated all notes discussed by emails and adjusted for MIPI Alliance White Paper

Chapter 1. Rationale

- Nexus standard does NOT define any small connectors with focus on trace as Nexus define message-based debug interface and it require more pins than JTAG. Namely:
  - S26x 1-104068-2, Low performance (1 MDO signal).
  - S40x 1-104549-6, Low performance (6 MDO signals - labelled as “not recommended”).
  - S50x 104549-7, Low performance (8 MDO signals).

As you can see smallest one with reasonable trace has 50 pins.

- There is a lot of hardware trace probes, which are being used for debugging and tracing of Arm cores. Arm defines two standard connectors for trace:
  - Based on MIPI 20-pin connector (MIPI does not provide that exact pinout) - this is for medium-performance trace (4-bit, 100 + MHz double edge captures, max trace bandwidth 800Mbps).
  - Based on Mictor 38-pin connector (also defined by MIPI) - this is for high-performance trace (16-bit, up to 400MHz double edge, max trace bandwidth 12.8Gbps).
- In july 2021 MIPI Alliance released White Paper which updates recommendations for debug and trace connectors.

This specification is slight extension of connectors described in MIPI Debug & Trace Connectors Recommendations. Extensions are as follows:

- Clarifying dual voltage debug and trace via Mictor 38 connector (re-defining obsolete pin #14).
- Defining MIPI20 pins #11 and #13 as optional TgtPwr pins (to supply 5V to evaluation target board).
- Allowing MIPI20 TRC_DATA[2] and TRC_DATA[3] to be optionally used as TRIGIN/TRIGOUT pins.
- Defining some signal as (optional) serial trace and application UART.

Chapter 2. MIPI20 Connector

This connector is an extension of MIPI10/MIPI20 connector as defined by RISC-V Debug Specification (version 0.13.2).

This specification adds 1/2/4-bit parallel trace, serial trace and some other alternative pin functions on same physical connector.

Table 1. MIPI20 Connector Layout

Signal	Odd Pin#	Even Pin#	Signal
VREF	1	2	TMS/TMSC
GND	3	4	TCK/TCKC
GND	5	6	TDO/SerialTrace
GND or KEY	7	8	TDI
GNDDetect	9	10	nRESET
GND/TgtPwr+Cap	11	12	TRC_CLK
GND/TgtPwr+Cap	13	14	TRC_DATA[0]/SerialTrace
GND	15	16	TRC_DATA[1]/nTRST
GND	17	18	TRC_DATA[2]/TRIGIN
GND	19	20	TRC_DATA[3]/TRIGOUT

Smaller MIPI10 version of this connector may still provide SerialTrace (assuming debug is usign cJTAG interface).

Table 2. Details of MIPI20 Signals

Pin#	Pin Name	Explanation
1	VREF	Reference voltage for all other pin and signals (same for debug and trace).
2	TMS/TMSC	JTAG TMS signal or cJTAG TMSC.
4	TCK/TCKC	JTAG TCK signal or cJTAG TMSC.
6	TDO/SerialTrace	Either TDO or serial trace (available in case cJTAG is used).
7	GND or KEY	May be removed pin (to prevent wrong insertion for non-shrouded connectors and cable with plug in pin#7). In case pin is not removed, it should be GND on target side.
8	TDI	JTAG TDI signal
9	GNDDetect	Must be GND on the probe. On-board debug circuitry can use this pin to disable itself when external debug probe is connected. If not used for that purpose should be GND on target side.

Pin#	Pin Name	Explanation
10	nRESET	Active-low, open-drain reset signal driven and monitored by the debug probe. Some debug probes may monitor this signal to handle resets from the target.
11	GND/TgtPwr+Cap	Should be GND for trace - see below for detailed explanation of GND/TgtPwr+Cap.
12	TRC_CLK	Parallel trace clock (from target to probe).
13	GND/TgtPwr+Cap	Should be shorted with pin#11 and share it's function
14	TRC_DATA[0]/SerialTrace	Either parallel trace or serial trace (from target to probe).
16	TRC_DATA[1]/nTRST	In case both nRESET and nTRST are needed, this pin can used as nTRST. NOTE: Still 1-bit parallel or serial trace is possible.
18	TRC_DATA[2]/TRIGIN	Either parallel trace signal (from target to probe) or input trigger (from probe to target) or application UART.
20	TRC_DATA[3]/TRIGOUT	Either parallel trace signal or output trigger (from target to probe) or application UART.

2.1. Explanation for GND/TrgPwr+Cap pins

Meaning and function of this pin is often misunderstood, so it deserves more elaborated explanation.

When target cannot be powered from MIPI20 both these pins should be GND (as most of pins on odd side of MIPI20 connector).

Another function of these pins (TgtPwr+Cap) is to provide target power supply voltage into evaluation target. This way to power-up evaluation target is equivalent to power from USB connector, so expected voltage is ~5V. Target should not assume this voltage is regulated - more or less same way as voltage provided by USB cable is.

Some debug probes may provide regulated voltage and dynamically measure total power consumption by the target.

Some targets provide jumpers to select power-source (either MIPI20 or USB), some provide diodes to prevent back-feeding voltage (in case it is provided by USB and MIPI20), but it is also OK to connect power from USB and MIPI20 together. Good debug probes sense voltage on these pins and not provide own voltage in case target is already powered.

Term '+Cap' means, that if this pins is used to provide power to the target, it should have capacitor (as close to the pin as possible) to improve quality of adjacent TRC_CLK and TRC_DATA pins. Another term for using a Cap on the supply pin is to make it an "AC ground" or "high frequency ground".

Leaving these pins not connected (NC) as can be seen on some schematics, is not very good option when trace is used. There is simply not enough groud around TRC_CLK and TRC_DATA[0] signals. Some leave it as NC is they perpahs worry that debug probes may provide voltage there and it will create problems - but debug probe should provide current protection and should disable TgtPwr function once it will detect, that target has this pin shorted to GND.

No matter what pins #11 and #13 should be **always** connected together - it is NOT possible that one of

them will function as GND and second as TgtPwr.

If you are in doubt, your board may have a jumper to either isolate these pins (NC) or connect then to GND or use them as target power. Jumper with 3 pins:

A-B-C

should work. Middle pin **B** should go to MIPI20, left pin **A** may be GND and right pin **C** may be 5V rail on the target. If there is no jumper MIPI20 pins are left NC, if there is a jumper **A-B**, MIPI20 pins are GND. If there is a jumper between **B-C**, then this pin will be able to supply power to the target.

2.2. Possible use of TDI/TDO and TRIGIN/TRIGOUT for application UART

Some debug probles may allow definition of pin functions and may serve as virtual UART terminal for the target. UART is often needed for testing and production and having both debug and UART on single connector is desired. Supporting UART over TDI/TDO will require 2-pin cJTAG to be used as debug interface. Supporting UART over TRIGIN/TRIGOUT pins will limit parallel trace to 1-bit or 2-bit options.

Chapter 3. Mictor 38 - bit Connector

Mictor-38 connector has all signals from MIPI20 connector and adds up to 16-bit trace and define more trigger pins. Mictor-38 connector is also designed for high-speed trace (it is rated for 400MHz double edge captures).

Mictor-38 connector provides also an option to have different reference voltages for debug and trace.

Table 3. Mictor-38 Connector Layout

Signal	Ref Voltage	Odd Pin#	Even Pin#	Ref Voltage	Signal
NC		1	2		NC
NC		3	4		NC
GND		5	6	Trace	TRC_CLK
TRIGIN	Debug	7	8	Debug	TRIGOUT
nRESET	Debug	9	10	Trace	EXTTRIG
TDO	Debug	11	12	Trace	VREF_TRACE
RTCK	Debug	13	14	Debug	VREF_DEBUG
TCK/TCKC	Debug	15	16	Trace	TRC_DATA[7]
TMS/TMSC	Debug	17	18	Trace	TRC_DATA[6]
TDI	Debug	19	20	Trace	TRC_DATA[5]
nTRST	Debug	21	22	Trace	TRC_DATA[4]
TRC_DATA[15]	Trace	23	24	Trace	TRC_DATA[3]
TRC_DATA[14]	Trace	25	26	Trace	TRC_DATA[2]
TRC_DATA[13]	Trace	27	28	Trace	TRC_DATA[1]
TRC_DATA[12]	Trace	29	30	Trace	Logic'0'
TRC_DATA[11]	Trace	31	32	Trace	Logic'0'
TRC_DATA[10]	Trace	33	34	Trace	Logic'1'
TRC_DATA[9]	Trace	35	36	Trace	EXT/TRC_CTL
TRC_DATA[8]	Trace	37	38	Trace	TRC_DATA[0]

Above table is using names compatible with MIPI specifications (however MIPI specifications is showing rows of pins starting from 38 down to 1).

3.1. Explanation for additional pins (comparing to MIPI20)

All debug signals share alternate functions as defined for Mictor connector (see above).

Table 4. Micror-38 additional pins (comparing to MIPI20 defined above)

Pin#	Pin Name	Explanation (comparing to MIPI20)
7	TRIGIN	Same as MIPI20 #18 alternative function but not shared with trace.
8	TRIGOUT	Same as MIPI20 #20 alternative function but not shared with trace.
10	EXTTRIG	External trace trigger from target (some trace probes may use it).
13	RTCK	Return trace clock (not applicable to RISC-V, may not be supported by all probes).
21	nTRST	Same as MIPI20 #16 alternative function but not shared with trace.
36	EXT/TRC_CTL	Not applicable (should be 0). May be also used to denote valid/idle state, but it may not be supported by all trace probes.

3.2. Dual voltage (different for debug and different for trace) configurations

Sometimes (due to speed reasons) it may be benefitial to drive SoC trace pins with different (usually lower) voltage then the debug signals. Such a configuration may be supported using sigle Mictor connector or two connectors (Mictor for trace only and MIPI for debug only). Be aware, that two different voltages may not be supported by simpler trace probes.

Single voltage - single Mictor (Recommended)

- Mictor #12: VREF_TRACE=VREF_DEBUG (Required)
- Mictor #14: VREF_DEBUG (Recommended, see NOTE *1 below) or NC

Single voltage - trace via Mictor, debug via extra JTAG connector (NOT Recommended)

- Mictor #12: VREF_TRACE=VREF_DEBUG (Required)
- Mictor #14: NC (Recommended, see NOTE #1 below) or VREF_DEBUG
- Mictor JTAG pins: Connected or NC (Recommended, see NOTE #2 below)
- JTAG connector VTREF (#1): VREF_DEBUG (Required)
- JTAG connector JTAG pins: Connected (Required)

Dual voltage - single Mictor (NOT Recommended)

- Mictor #12: VREF_TRACE (Required)
- Mictor #14: VREF_DEBUG via jumper on PCB (Required, see NOTE #3 below)

Dual voltage - trace via Mictor, debug via extra connector (Recommended)

- Mictor #12: VREF_TRACE (Required)
- Mictor #14: NC (Required, see NOTE #3 below)
- Mictor JTAG pins: NC (Required, see NOTE #4 below)
- JTAG connector VTREF (#1): VREF_DEBUG (Required)
- JTAG connector JTAG pins: Connected (Required)

#1 *Jumper (on PCB) between Mictor pin#14 and VREF_DEBUG rail on PCB can be used to select NC or VREF_DEBUG. Some trace probes (such as TRACE32 from Lauterbach) require VTREF_DEBUG to be present on pin #14.*

#2 *If JTAG pins are NC, JTAG quality/speed may be better as there will be no stubs introduced by extra routing on PCB.*

#3 *Jumper provides extra safety in case trace probe/adapter which does not support dual-voltage is used. Before fitting this jumper, make sure probe/adapter you are using is NOT shorting Mictor pin#12/#14 internally. If this is the case, two voltage rails may be shorted and target may be permanently damaged. Some trace probes (such as TRACE32 from Lauterbach) require VTREF_DEBUG to be present on pin #14.*

#4 *All JTAG pins should be NC from a reason mentioned in NOTE 2. But mainly to make sure, that there will be only single voltage present on this connector.*

EXTRA NOTES (related to debug and trace voltages)

1. Lower voltage allows faster trace, but it is then more critical to have correct PCB design.
2. Allowed reference voltage ranges (for JTAG and trace) are different for different probes.
3. Lower voltage for trace may be good choice with FPGA-based development boards.
    - Trace pins may be available on FPGA bank, which is setup for lower IO voltage.
4. When high-speed trace is important Mictor-38 should be the only debug and trace connector on particular PCB.
    - In case two connectors are used trace signals should have routing priority.
    - Many probe vendors provide adapters from Mictor to standard JTAG-only connectors, so non-trace probes can be used with target/PCB with Mictor-only connector.
5. Not all trace probes which support Mictor-38 connector are capable of handling dual voltage trace.
    - In the moment of this writing at least I-jet-Trace-A/R/M (by IAR Systems) and Trace32 (by Lauterbach) probes support such a mode (in both single Mictor and two Mictor + JTAG connectors).
6. It is not recommended to add buffers on PCB to adjust JTAG (usually higher) voltage to trace voltage.
    - It is not only affects signal quality but also introduces extra delays, what may create problems for simple probes.
    - It is very hard to properly handle fast switching bidirectional signal, so cJTAG and SWD debug protocols may never reliably work.
    - It makes PCB more complicated without really good reason.

3.3. Explanation for Mictor-38 pins #30/32/34/36

It may be hard to understand why TRC_DATA[0] is not together with other TRC_DATA[?] signals and why pins #30/32/34 have specific fixed values.

This is caused by desire to provide compatibility with initial versions of Arm trace. These older version used these 4 pins to denote idle state. Modern trace probes ignore these signals, but just in case they do not, it better to provide logic level as above. As TRC_CTL is not used, it should be tied to 0, but may be optionally used as extra external trigger (from target to probe).

Chapter 4. Adapters, multiple connectors and on-board debug considerations

It is often seen that some evaluation boards provide more than one standard connector. This is not only costly, but also not necessary as most trace and debug probe vendors provide passive adapters or cables to adapt different pinouts as part of standard offering.

In case several connectors must be used, highest performance connector should be placed as closest one to trace MCU pins. For example if you want to have Mictor for high-speed trace and MIPI10 for casual-debug (and/or slow serial trace), Mictor should have all JTAG and trace signals connected. All JTAG signals should go 'through' that Mictor connector and go to MIPI10 connector. All high-speed trace signals should not go any further than to Mictor connector pins.

In rare case more than one trace connector is desired, it is suggested to place 0R/DNP resistors to reduce fanout on trace lines. Be aware, that every PCB 'disruption' (via, test-point, resistor) will cause reflections and signal degradation.

It is also very important to provide good GND on all GND pins for high quality high-quality trace. Assure all trace lines on PCB are of similar length and have identical impedance. In case trace pins are shared as functional IO, make sure that it is possible to cut-out devices connected to trace data lines (via 0R resistors or solder bridges - jumper are not recommended at these provide additional signal degradation).

In case scoping of trace signals is necessary, it is suggested to have good GND test point (where wire can be soldered) close to where scope can be connected.

MIPI Aliance White Paper (referenced at the beginning) provides extra details as far as routing signal trace on target PCB.

In case when on-board ciruitry is used for debug, that circuitry should monitor GNDDetect pin (MIPI20/MIPI10 #9). In case GND is detected there, it means that external debug probe is connected to that connector and in such a case on-board debug chip should tri-state all it’s outputs and disable all pull-up/pull-down on all pins, so external debug probe operation will not be disturbed by on-board debug circuitry.