



# N-Trace for RISC-V PlanReview

(N-Trace means IEEE Nexus[based] Trace)

Chair: Robert Chyla, SiFive  
Vice-chair: Jay Gamoneda, NXP  
February 8-th, 2023, Tech Chairs Review

# What is handled by N-Trace TG

- Main page: <https://lists.riscv.org/g/tech-nexus>. Agenda:

The Nexus Trace Task Group is responsible for analysis of Nexus IEEE-ISTO 5001™ standard and its applicability for trace of RISC-V cores. The Nexus standard is well established, silicon proven and extensively documented. It's necessary to define parts of the standard that are applicable to RISC-V trace.

Github repo: <https://github.com/riscv/tg-nexus-trace> contains working documents and reference C code for encoder and decoder.

The following parts of Nexus specification will be addressed:

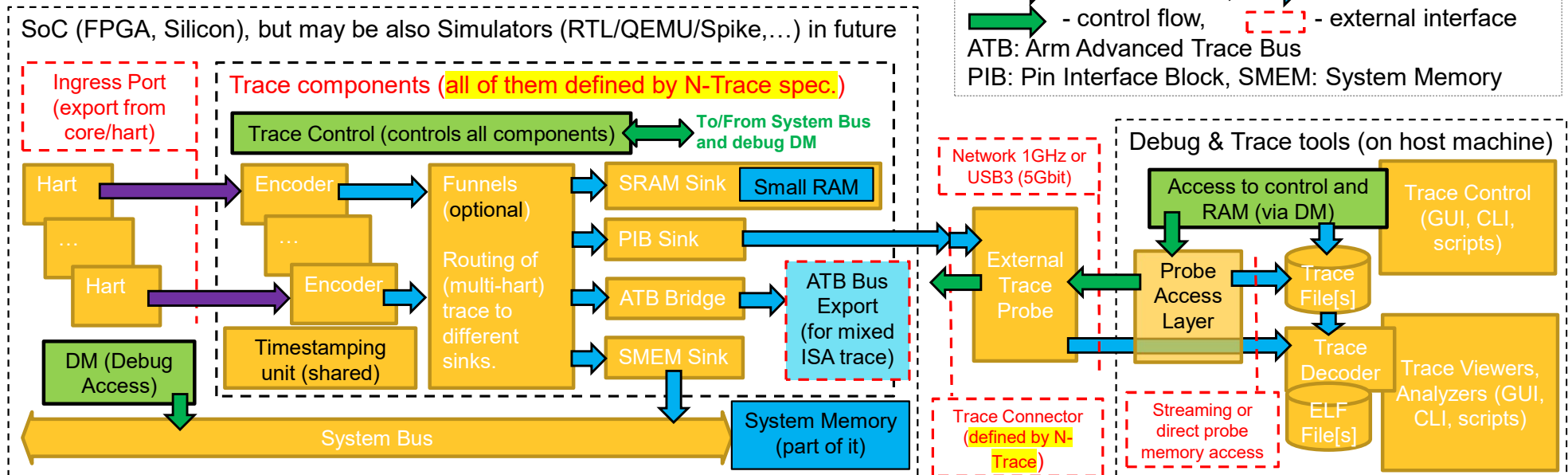
- Nexus compatible trace encoding
- Trace control
- Trace configuration
- On-chip and off-chip trace routing
- Physical trace connector options

This group will not address the debug part of the Nexus standard.

- Github repository: <https://github.com/riscv-non-isa/tg-nexus-trace>
  - Reference C encoder/decoder and tests are part of this github repository.
  - PDFs (almost internally frozen): <https://github.com/riscv-non-isa/tg-nexus-trace/tree/master/pdfs>
    - Finishing one of PDFs (some ADOC formatting issues recently – solved ...)

# What is N-Trace?

- **Complete, end-to-end** program flow trace for RISC-V harts/cores/systems.
- Based on mature, well-established, silicon-proven **Nexus IEEE-ISTO 5001 Standard** and:
  - Utilizing existing trace probes and professional trace tools (**MIPI-based connectors, same as used in Arm-based SoCs**).
  - **Initial version implemented in many RISC-V designs** (SiFive for over 3 years), including mixed-architecture systems.
- Logical diagram of N-Trace based trace system:



# History of Nexus and N-Trace

- Nexus IEEE-ISTO 5001 is mature and well-established standard
  - Originally it was targetting embedded processors (these days everything is 'embedded' ...)
  - Version 1.0 released in 1999.
  - Version 2.0 released in 2003/12/15.
  - Version 3.0 released in 2012/06/01 (added high-speed Aurora trace option)
  - Implemented in (non-RISC-V ...) silicon many times.
- Over more than two decades Nexus trace was supported by trace and debug probe tool vendors
  - Different ISA: PowerPC, ARC, AVR32, even early Arm cores (MAC711 by Freescale/NXP).
  - However, Arm debug and trace (Coresight) did NOT embrace Nexus.
  - Nexus debug was not main-stream, but trace is simple and elegant.
- Dec 2019: RISC-V Nexus Trace TG proposed
  - Later renamed to N-Trace
- Mar 2020: SiFive Insight (Nexus-based trace) was released as part of standard offering.
- May 2020: Reference open-source encoder/decoder framework was published (as part of TG).
- June 2020: SiFive donated Trace Control Interface to Nexus TG
  - Improved Trace Control Interface will be ratified and shared with (ratified) E-Trace encoder protocol.
- Q1 2023: N-Trace (complete, end-to-end trace for RISC-V) is approaching ratification.

# N-Trace and E-Trace relations/compare

- E-Trace:
  - Defines Trace Ingress Port (data sent every cycle from inside of a hart – retired instruction blocks + minimal details).
  - Defines Trace Encoding protocol (packets of compressed trace).
  - Defines Data Trace and extends ingress port (added in version 2.0).
  - Provides reference encoder/decoder Python implementation and set of tests.
  - Based on UltraSoc/Siemens Trace IP.
- N-Trace (much wider scope ...)
  - Defines **all components** of end-to-end-trace program flow trace
    - Ingress port re-used from E-Trace
    - Messaging Protocol (different set of messages than E-Trace) with several timestamp options
    - Multi-core/hart transport and SoC routing
    - Storage (on chip in small buffer or system memory)
    - Export via pins (parallel or serial, with calibration)
      - Two MIPI compatible connectors (supported by many trace probe vendors)
  - Based on Nexus IEEE standard (industry standard used in the past and still use today)
    - Adopted by SiFive over 3 years ago and silicon proven since, Control Spec donated to Nexus TG
  - Trace Control Layer (MMIO register-based ...) to be shared with E-Trace
  - Reference C code (encoder/decoder provided 3 years ago as N-Trace messages are 100% Nexus compatible)
    - Recently E-Trace tests were run (to assess trace compression)



# N-Trace compression results and analysis

- BTM and 3 HTM flavors
  - Gain is *(previous/this - 1)* (in %)
  - We look at embench-\* only as 'real-code'
  - CRC32 code is excluded (crazy looping?)
- Basic HTM** is on average 2.3 times better than BTM
- Repeated History** adds 7.9%
- Return Address Stack** adds 78.9%
- Total gain is  $1.289/0.200=6.595$  times
- Compression is code-dependent
  - Compiler options dependent as well
  - For some programs gains are HUGE (26x!)
- Average 0.2bits/instr is excellent
  - 1GHz core with 1 IPC will generate 200Mbps of trace
  - This needs to be scaled to core clock and IPC to estimate trace bandwidth
- External trace will be very good
  - PIB provides calibration generator option
  - Basic MIPI-20 connector can handle 1.2Gbps of 4-bit trace
  - Mictor-38 connector can do 12.6Gbps
  - USB 3 steaming is up to 5Gbps
- Use reference encode/decoder
  - Test your own code and implementation
  - This statistical data will be there (as CSV)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1			BTM, No Repeat				HTM, No Repeat			HTM		HTM, Repeat			Repeat		HTM, Repeat, RAS(8)			RAS	
2	Name	InstrCnt	Bytes	MsgCnt	Bits/instr	Bytes	MsgCnt	Bits/instr	gain (%)	Bytes	MsgCnt	Bits/instr	gain (%)	Bytes	MsgCnt	Bits/instr	gain (%)	Bytes	MsgCnt	Bits/instr	gain (%)
3	coremark	33399227	6308812	3098601	1.511	1573855	228644	0.377	300.9%	1497855	224644	0.359	5.1%	1284989	183733	0.308	16.6%				
4	dhystone	215010	71264	27228	2.652	44157	8041	1.643	61.4%	44157	8041	1.643	0.0%	7062	1023	0.263	525.3%				
5	embench-aha-mont64	4541661	714908	354473	1.259	108591	15698	0.191	558.3%	108591	15698	0.191	0.0%	103892	14842	0.183	4.5%				
6	embench-crc32 (not averaged!)	4028857	1051518	350388	2.088	876587	175285	1.741	20.0%	3290	694	0.007	26544.0%	2748	515	0.005	19.7%				
7	embench-cubic	7724337	1153325	493161	1.194	366621	51664	0.380	214.6%	366621	51664	0.380	0.0%	169218	24171	0.175	116.7%				
8	embench-edn	3493777	692741	345476	1.586	80683	11504	0.185	758.6%	80683	11504	0.185	0.0%	75591	11051	0.173	6.7%				
9	embench-huffbench	2461117	619722	307886	2.014	116641	16675	0.379	431.3%	116641	16675	0.379	0.0%	113009	16216	0.367	3.2%				
10	embench-matmult-int	2891806	778851	389413	2.155	92476	13214	0.256	742.2%	92476	13214	0.256	0.0%	91108	13084	0.252	1.5%				
11	embench-minver	2620515	547936	222193	1.673	260604	37878	0.796	110.3%	260604	37878	0.796	0.0%	85718	11690	0.262	204.0%				
12	embench-nbody	6394542	1128134	495050	1.411	354340	50579	0.443	218.4%	354340	50579	0.443	0.0%	183038	26149	0.229	93.6%				
13	embench-nettle-aes	4523969	139984	50429	0.248	17481	2466	0.031	700.8%	17481	2466	0.031	0.0%	16048	2293	0.028	8.9%				
14	embench-nettle-sha256	3874834	28604	8101	0.059	17661	2866	0.036	62.0%	17661	2866	0.036	0.0%	6202	954	0.013	184.8%				
15	embench-nsichneu	2241141	373340	186044	1.333	173916	24848	0.621	114.7%	173916	24848	0.621	0.0%	173886	24841	0.621	0.0%				
16	embench-picojpeg	4012848	686160	332728	1.368	110347	15432	0.220	521.8%	109997	15432	0.219	0.3%	74033	11365	0.148	48.6%				
17	embench-qrduno	3426824	610717	290071	1.426	126037	18037	0.294	384.6%	126037	18037	0.294	0.0%	115893	16557	0.271	8.8%				
18	embench-sglib-combined	2269619	553369	249202	1.951	221303	31056	0.780	150.1%	200093	27606	0.705	10.6%	116956	16818	0.412	71.1%				
19	embench-slre	2622477	514857	208689	1.571	276878	39748	0.845	86.0%	269441	39637	0.822	2.8%	118871	16982	0.363	126.7%				
20	embench-st	4412657	674721	253634	1.223	314834	44172	0.571	114.3%	302136	42576	0.548	4.2%	98065	14009	0.178	208.1%				
21	embench-statmate	1038135	171139	66888	1.319	82615	11805	0.637	107.2%	82615	11805	0.637	0.0%	22675	3239	0.175	264.3%				
22	embench-ud	1277146	114259	48981	0.716	28190	4453	0.177	305.3%	28190	4453	0.177	0.0%	22405	3201	0.140	25.8%				
23	embench-wikisort	2346529	518337	210279	1.767	244839	40063	0.835	111.7%	244839	40063	0.835	0.0%	65361	9784	0.223	274.6%				
24	median	15010	7675	3756	4.091	1565	247	0.834	390.4%	1565	247	0.834	0.0%	969	147	0.516	61.5%				
25	mm	297033	31019	14230	0.835	5359	809	0.144	478.8%	5359	809	0.144	0.0%	1739	272	0.047	208.2%				
26	mt-matmul	41449	14314	6593	2.763	2892	463	0.558	395.0%	2892	463	0.558	0.0%	587	96	0.113	392.7%				
27	mt-vvadd	61067	30235	14502	3.961	6295	1000	0.825	380.3%	6295	1000	0.825	0.0%	1220	201	0.160	416.0%				
28	multiply	55011	29981	14806	4.360	4601	768	0.669	551.6%	4601	768	0.669	0.0%	3133	456	0.456	46.9%				
29	qsort	235010	76348	38054	2.599	14494	2097	0.493	426.8%	14494	2097	0.493	0.0%	12572	1826	0.428	15.3%				
30	rsort	375011	36100	15906	0.770	4910	729	0.105	635.2%	4910	729	0.105	0.0%	2081	329	0.044	135.9%				
31	spmv	70010	7759	3427	0.887	1896	296	0.217	309.2%	1896	296	0.217	0.0%	1548	230	0.177	22.5%				
32	towers	15011	3386	1348	1.805	1811	354	0.965	87.0%	1811	354	0.965	0.0%	262	46	0.140	591.2%				
33	vvadd	10011	4411	2125	3.525	880	148	0.703	401.3%	880	148	0.703	0.0%	298	53	0.238	195.3%				
34	xrle	164959	12978	6233	0.629	3389	485	0.164	282.9%	3389	485	0.164	0.0%	2685	404	0.130	26.2%				
35	SUM of Embench	66202791	10021104	4512698	1.289	2994057	432158	0.385	234.7%	2955652	427695	0.357	7.9%	1651969	237246	0.200	78.9%				
36	SUM of all tests	101156610	16655386	7759507	1.372	4660161	676239	0.384	257.4%	4545756	667776	0.360	6.8%	2971114	426062	0.235	53.0%				
37			Average excludes -crc32				Average excludes -crc32			Average excludes -crc32			Average excludes -crc32			Average excludes -crc32			Average excludes -crc32		
38			Msg size: 2.22				Msg size: 6.93			Msg size: 6.91			Msg size: 6.96			Msg size: 6.96			Msg size: 6.96		
39			BTM has a lot of small messages!				HTM (branch history) is 234% better than BTM (no history)			For HTM History Repeat provides some moderate gain (7.9%)			Return Address Stack (RAS) is BIG gain (78.9%)			Return Address Stack (RAS) is BIG gain (78.9%)			Return Address Stack (RAS) is BIG gain (78.9%)		
40							For some cases it is BIG (over 7x)			For some cases it is BIG (over 26x)			For some cases it is BIG (over 26x)			For some cases it is BIG (over 5x)			For some cases it is BIG (over 5x)		
41																					

# N-Trace ratification plan

- This is non-ISA extension (one of 'grandfathered' TGs ...)
- Most parts of N-Trace specification are on 'internal freeze'
  - Several components as defined in charter (described in own documents) are unchanged for some time.
  - Main work is now to document messaging protocol correctly and without ambiguities.
    - Nexus specification itself is very flexible (too flexible?) format.
    - We have an ambition for N-Trace specification to be standalone.
  - Reference C encoder and decoder should be useful to assure conformance and help with development.
- We are not seeking any waivers
  - Reference C code and available for over 2 years (it was used for some implementations ...)
  - Nexus itself is silicon proven.
  - Implementation by SiFive is used in production for about 3 years.
  - There are several other implementations in different stages.
- Ratification plan: [https://docs.google.com/document/d/1Elt9-ECIAyzVe7snwCMq\\_K7RHPlrTe3PaDX6JaKAXJA/edit#heading=h.hym1mtuc89he](https://docs.google.com/document/d/1Elt9-ECIAyzVe7snwCMq_K7RHPlrTe3PaDX6JaKAXJA/edit#heading=h.hym1mtuc89he)
- Checklist: <https://docs.google.com/spreadsheets/d/13vKaon759gWw5JTd7kplhMaymnhTsn-egnScl6pnEzE/edit#gid=257164574>

# N-Trace Ratification (proof of concept, compatibility, testing)

- Proof of concept
  - Nexus standard was used in silicon for 20+ years (and it is still used)
  - SiFive Insight Debug and Trace is N-Trace/Nexus compatible
    - SiFive donated silicon proven spec to Nexus TG (explained on History ... page)
- Compatibility
  - N-Trace based implementation (by SiFive) is available for about 3 years and present in many SoC parts (including mixed-architecture SoC).
  - Two standard connectors (low/high-bandwidth) are fully MIPI and Arm compliant.
    - Most (all?) trace probe vendors support it already. Namely: Lauterbach (since 2018?), IAR(since 2019), Segger(since 2022/11), Ashling(since 2022/12), GHS(since?).
  - Nexus trace (for RISC-V) was demonstrated on many RISC-V Summit sessions over the years. One is here: <https://github.com/IARSystems/trace-riscv/blob/main/docs/riscv-v-summit-2020/risc-v-summit-2020-trace-overview-tutorial.pdf>
- Implementation may be cross-checked with reference C encoder/decoder
  - <https://github.com/riscv-non-isa/tg-nexus-trace/blob/master/refcode/c/NexusTraceTG-RefCode.pdf>
  - It was used to aid several implementations (besides SiFive)
  - It was run on same set of tests which were used by E-Trace



# N-Trace ratification plan

- From 'Ratification Plan'

The following steps will be taken as the timeline and milestones:

1. **Plan milestone: Review the [status checklist](#) requirements with [RISC-V Staff](#) and present the completed plan to the Chairs meeting**

Reviewed status checklist link:	<a href="#">Nexus Trace Status Checklist</a>
Planned Plan Approval Date:	N/A (grandfathered, started 12/17/2019)
Actual Plan Approval Date:	N/A (grandfathered)
Status:	N/A (grandfathered)
(Planned, Needs sign-off, Approved)	

2. **Freeze: Pass non-ISA DoD requirements, including DoD sign-offs**

Planned Freeze Approval Date:	2/28/2023
Actual Freeze Approval Date:	[MM/DD/YYYY]
Status:	Planned
(Planned, Needs sign-off, Approved)	

3. **Specification review**

Reviewers:	Public review on isa-dev
(Public review on isa-dev, Public review other, Technical Chairs, or ...)	
Review rationale if not Public review on isa-dev:	Not applicable
Duration (Days)	45
Rational if duration is not 45 days:	Not applicable

Planned Review Start Date:	3/14/2023
Actual Review Start Date:	[MM/DD/YYYY]
Status:	Planned
(Planned, In progress, Complete)	

4. **Vote-ready: Pass non-ISA DoD requirements, including DoD signoffs**

Planned DoD Sign-off Date:	4/21/2023
Actual DoD Sign-off Date:	[MM/DD/YYYY]
Status:	Planned
(Planned, Sign-off requested, Approved)	

5. **TSC Approval**

Planned TSC Approval Date:	5/5/2023
Actual TSC Approval Date:	[MM/DD/YYYY]
Status:	Planned
(Planned, Approval requested, Approved)	

6. **Board ratification**

Planned BOD Ratification Date:	5/19/2023
Actual BOD Ratification Date:	[MM/DD/YYYY]
Status:	Planned
(N/A, Planned, Ratification requested, Ratified)	
Justification if no Board ratification: <a href="#">[EXPLANATION IF REQUIRED]</a>	

# N-Trace ratification plan (Checklist)

•

Freeze Checklist			
Name	Task Description	Status	Notes
Document Complete	<a href="https://github.com/riscv-non-isa/tg-nexus-trace/blob/master/pdfs/RISC-V-N-Trace-Specification.pdf">https://github.com/riscv-non-isa/tg-nexus-trace/blob/master/pdfs/RISC-V-N-Trace-Specification.pdf</a>	In process	
Code	<a href="https://github.com/riscv-non-isa/tg-nexus-trace/tree/master/refcode/c">https://github.com/riscv-non-isa/tg-nexus-trace/tree/master/refcode/c</a>	Done	
Tests	<a href="https://github.com/riscv-non-isa/tg-nexus-trace/tree/master/refcode/c/examples/all">https://github.com/riscv-non-isa/tg-nexus-trace/tree/master/refcode/c/examples/all</a>	Done	
Arch Review	Email <a href="mailto:tech-arch-review@lists.riscv.org">tech-arch-review@lists.riscv.org</a> for review. <a href="#">Policy</a> in development.		
Proof of Concept	This is based on Nexus 5001 IEEE Standard, which was implemented many times over last 20+ years. SiFive Insignth (available since 2019) implements version 0 of this specification. There are some other preliminary implementations as well. Also many trace tool vendors supports this already.	Done	
RISC-V Spec Policies	Abide by policies: encumbered information, friendly terminology, anonymous contributor (completed by RISC-V staff)		
Committee Chair Signoffs	OpaVote for all Committee Chairs (completed by RISC-V staff)		
CTO Signoff	Final check by CTO		
		In process	
		Waiver requested	
		Waiver granted	
		Done	