

SoC Level Analytics, Trace & Debug for RISC-V Designs

Rupert Baines, CEO, UltraSoC

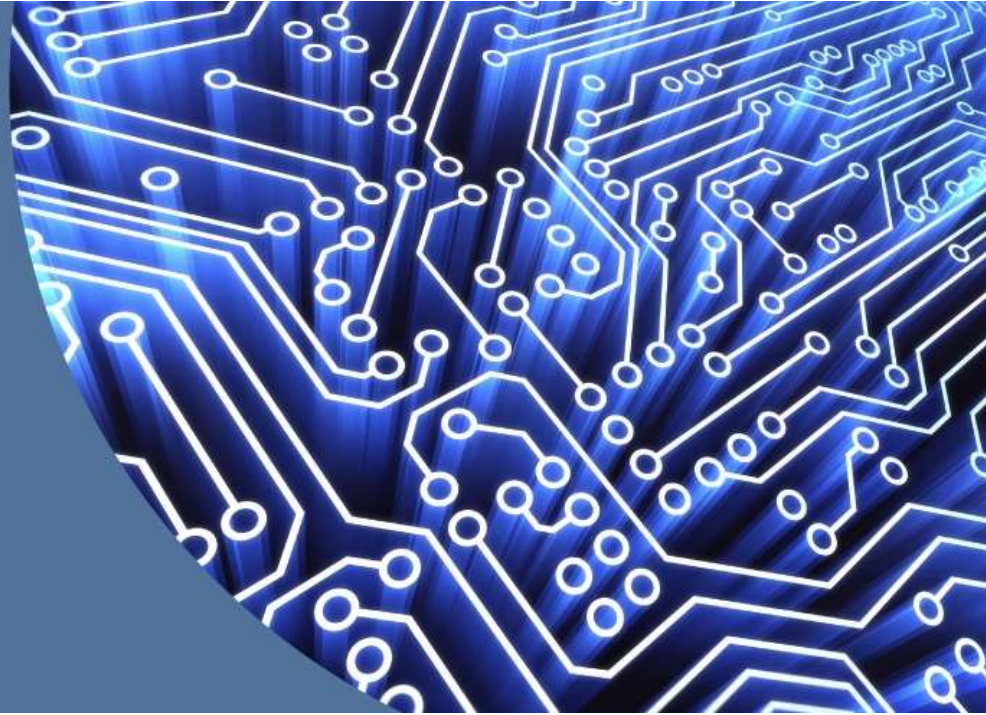
rupert.baines@ultrasoc.com

RISC-V Technical Symposium Moscow

20 May 2019



- Overview
- Processor Trace
- Algorithm
- Holistic System
- Demo System
- Summary





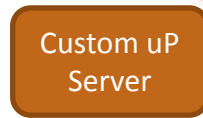
- Embedded analytics
 - On-chip hardware monitors delivered as silicon intellectual property (SIP)
 - Supporting debug in-lab, & safety and security in-life
- Silicon-proven with multiple customers
- Founded 2009: VC-funded
- 35 employees; 40+ patents; HQ Cambridge UK

→ UltraSoC: partners and customers

ultraSOC



SEAGATE



SiFive



cadence





Actionable Insights across the whole SoC

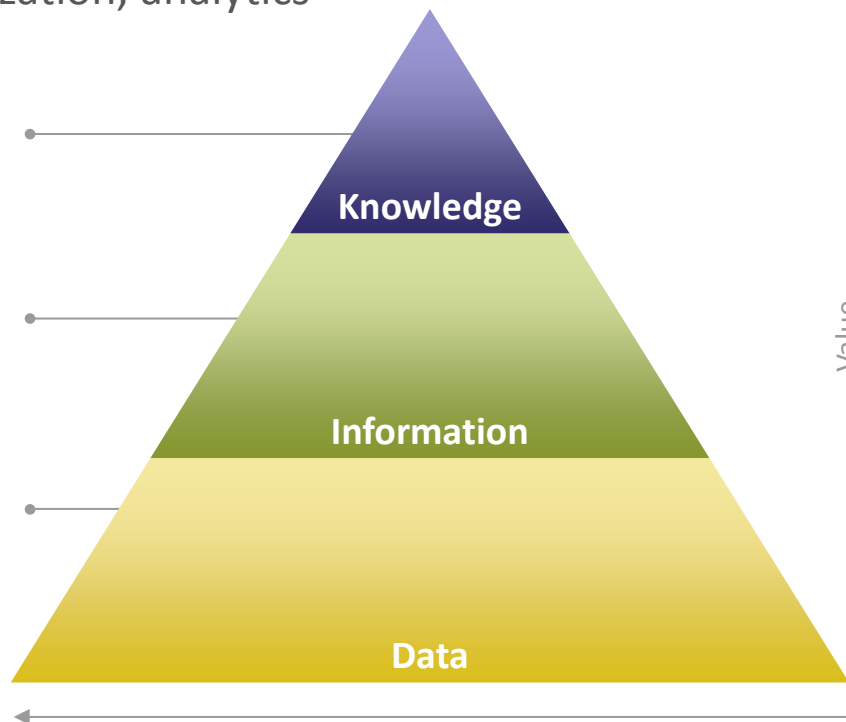


Debug, optimization, analytics

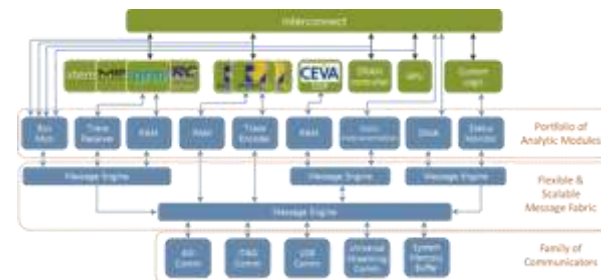
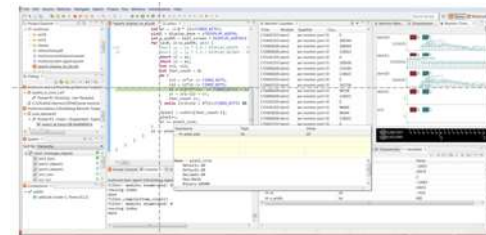
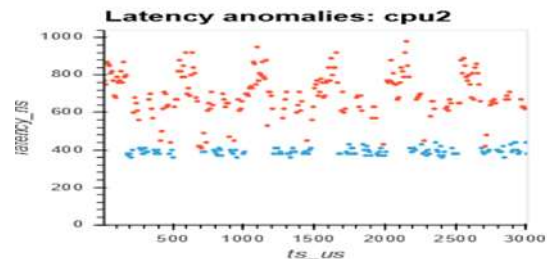
UltraSoC delivers actionable *insights*

With system-wide *understanding*

From rich *data* across the whole SoC

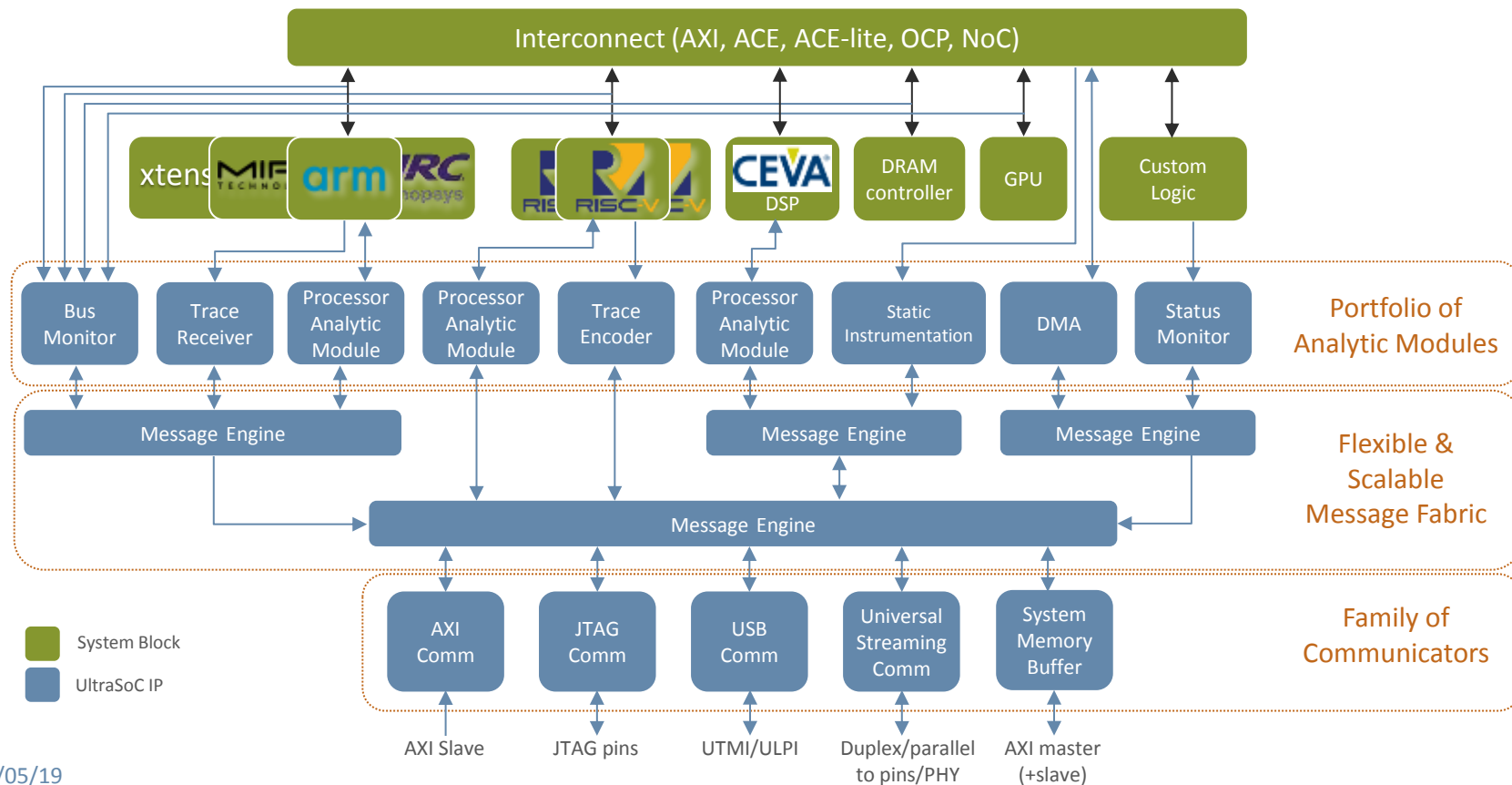


UltraSoC enables *full* visibility of SoC





Advanced debug/monitoring for the whole SoC





Software tools for data-driven insights



Eclipse based UltraDevelop 2 IDE

Control

Multiple CPUs

Configuration

Single step & breakpoint CPU code

SW & HW in one tool

Real-time HW Data

Instruction trace

Third Party Tool Vendor Partnerships





UltraSoC creates value both **in-lab** and **in-life**



Lab test

Field trial

In Life

Capital and Labor Expenditures



Comparison of Market Metrics and Sales Results

	Using UltraSoC	Without UltraSoC	Total Market
M Units	41.3	25.6	83.0
Market Share Units	50.0%	30.7%	
Market Revenue M\$	\$341.9	\$181.6	\$652.2
Market Share	52.4%	27.9%	
Profits M \$	\$131.0	\$57.3	
Average Gross Margin	38.3%	31.5%	
Break-even M Units	7.6	14.3	
Aggregate ASP	\$8.28	\$7.17	Over entire product life

Source: Semico Research Corp.

UltraSoC **accelerates innovation** and maximizes profitability
Faster TTM, higher quality, lower cost & higher margin

UltraSoC **detects threats** and hazards
an order of magnitude faster than
any other solution – radically
increasing security and safety



UltraSoC allows rapid **optimization of application SW**: improving performance, reducing TCO



RISC-V Ecosystem





RISC-V Ecosystem





- UltraSoC has the only commercial development environment for RISC-V
 - Includes run control and trace
 - Heterogeneous, massively multicore
 - FPGA demonstrator, Eclipse IDE (gdb, gcc, openOCD, Imperas MPD)
- Silicon proven solution
- Partnerships with leading core vendors
- RISC-V Foundation member since 2016
 - Chair of trace group, member/contributor debug group





- In complex systems understanding program behavior is not easy
- Software often does not behave as expected
 - Interactions with other cores' software, peripherals, realtime events, poor implementation or some combination of all of the above
 - Hiring better software engineers is not always an option
 - But usually because engineers write code with bugs in
- Using a debugger is not always possible
 - Realtime behavior is affected
- Providing visibility of program execution is important
 - This needs to be done without swamping the system with vast amounts of data
- One method of achieving this is via Processor Trace



Standardization



- Debug
 - Run-control, halt, single step etc
 - Ratified by Foundation
 - Supported by all core vendors
 - Support from standard tools (GDB etc)
- Trace
 - Working Group has "working consensus" for first release (instruction trace)
 - Supported by most core vendors (SweRV, SiFive, Andes etc)
 - Supported by open source (Boom and –soon – Pulp)
 - Commercial encoder IP (UltraSoC)
 - Open source encoder soon (ETH)
 - Support from tools (Lauterbach, IAR etc)



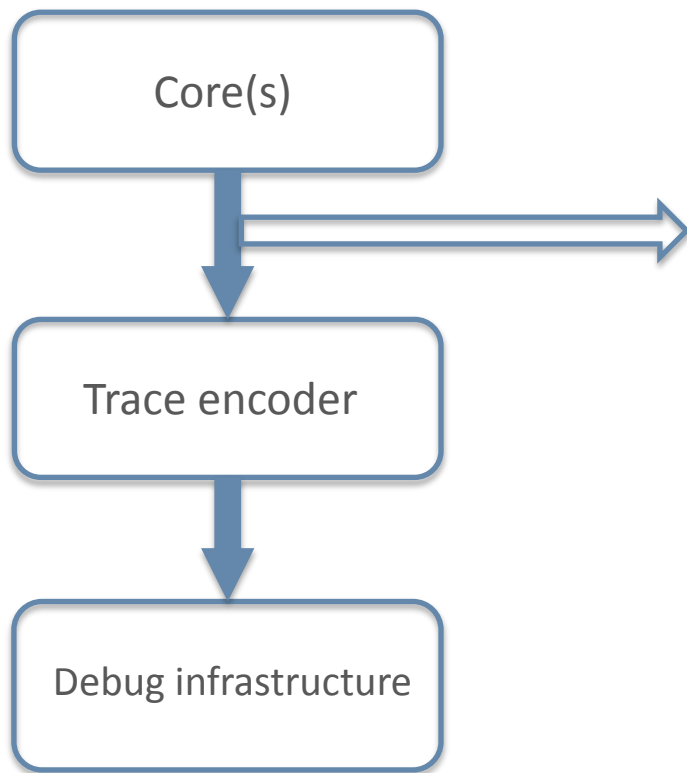
Branch vs Cycle-Accurate Trace



- Branch trace tracks execution from a known start address and sends messages about the deltas taken by the program
 - Jump, call, return and branch type instructions; interrupts and exceptions
 - Instructions between the deltas can be assumed to be executed sequentially
- Cycle-accurate trace tracks execution per-cycle
 - Required for real-time code optimization



Trace Encoder Ingress Port



Signal	Function
invalid	Instruction has retired or trapped (exception)
iexception	Exception
interrupt	0 if the exception was synchronous; 1 if interrupt
cause [CAUSELEN-1:0]	Exception cause
tval[XLEN-1]	Exception data
priv[PRIVLEN-1:0]	Privilege mode during execution
iaddr [XLEN-1:0]	The address of the instruction
instr[ILEN-1:0]	The instruction

- For cores retiring N instructions per clock cycle the interface is replicated N times



Trace Encoder Output

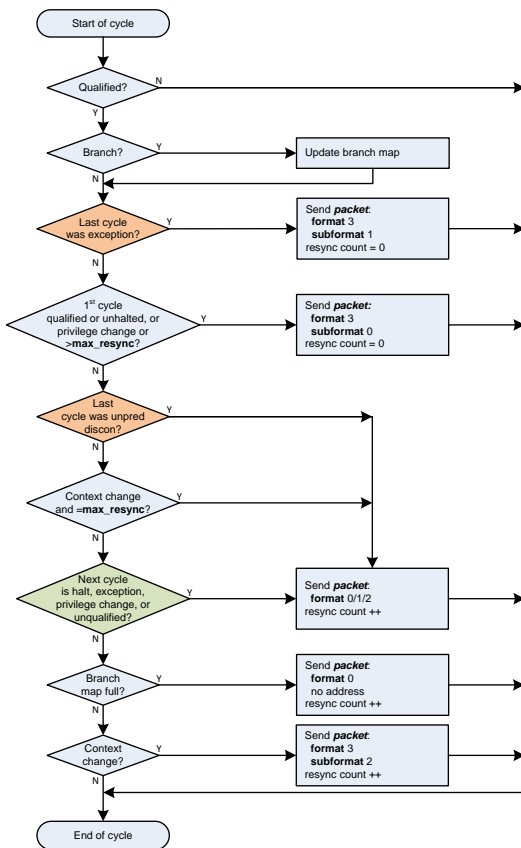


- The Encoder sends a packet containing one of the following:
 1. Update – a branch map with or without a differential destination address/next address
 2. Update – a full destination/next address and branch map
 3. Update – a differential destination/next address with no branch or instruction related fields.
 4. Synchronise - a context with or without a full current address

The above ensures an efficient packing to reduce data being routed on and subsequently transported off-chip



Instruction Trace Algorithm



- Formats 0 and 1 send branch map and address
- Format 2 is address only
- Format 3 is a sync packet
 - Subformat 0 for when starting or resume from halt. No **ecause**, **interrupt** and **tval**
 - Subformat 1 for exception. All fields present
 - Subformat 2 for context change. No **address**, **ecause**, **interrupt** and **tval**.



Trace Control



- Controlling when trace is generated is important
 - Helps reduce volume of trace data
- Filters are required.
- Using filters the following trace examples are available:
 - Trace in an address range
 - Start trace at an address end trace at an address
 - Trace particular privilege level
 - Trace interrupt service routines
- Other examples
 - Trace for fixed period of time
 - Start trace when external (to the encoder) event detected
 - Stop trace when an external (to the encoder) event detected



Encoding Efficiency



Benchmark	Instructions	Packets	Payload Bytes	Bits per instruction
dhystone	215015	1308	5628	0.209
hello_world	325246	2789	10642	0.262
median	15015	207	810	0.432
mm	297038	644	2011	0.054
mt-matmul	41454	344	953	0.184
mt-vvadd	61072	759	2049	0.268
multiply	55016	546	1837	0.267
pmp	425	7	39	0.734
qsort	235015	2052	8951	0.305
rsort	375016	683	2077	0.044
spmv	70015	254	1154	0.132
towers	15016	72	237	0.126
vvadd	10016	111	316	0.252
Mean				0.252

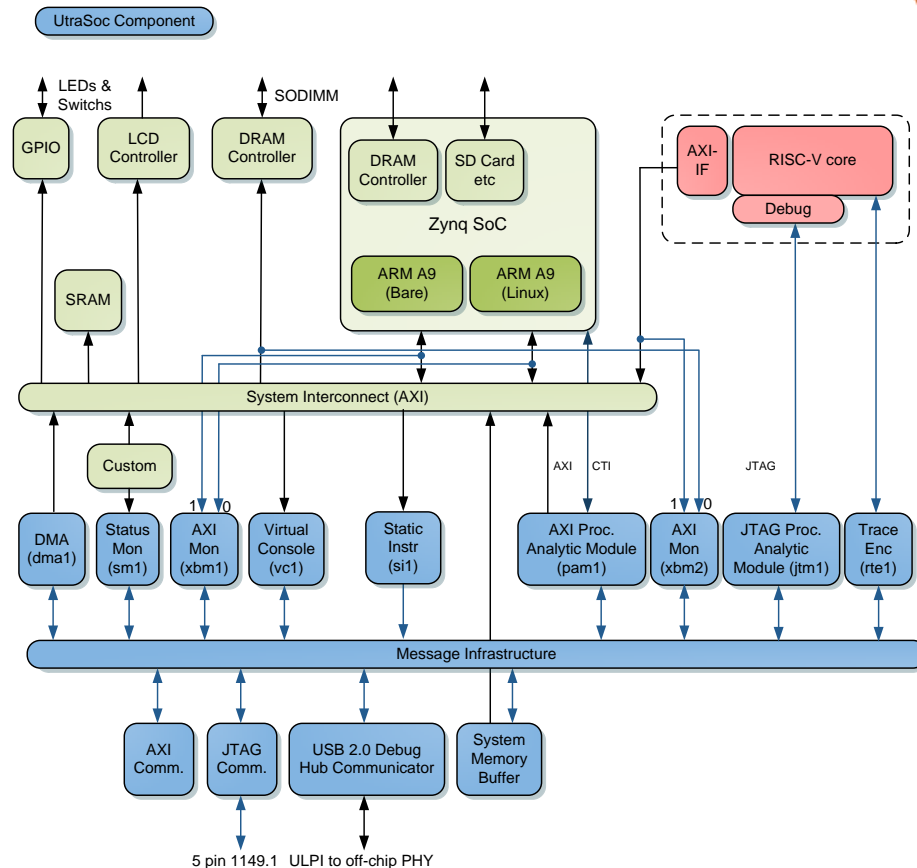
- Table shows encoding efficiency of the algorithm
- Does not include any overhead for encapsulating into messages or routing
- Different program types will have different overheads



Demo System Architecture



- Zynq ZC706 FPGA platform
 - Arm
 - Plus RV32 RISC-V
 - Plus custom logic
- Demo shows:
 - Bus state
 - Traffic
 - Performance histogram
 - Memory
 - Processor control
 - Bus deadlock detection
 - RISC-V Processor trace



5 pin 1149.1 ULPI to off-chip PHY



Software tools for data-driven insights



Eclipse based UltraDevelop 2 IDE

Control

Single step & breakpoint CPU code

Multiple CPUs

Configuration

SW & HW in one tool

Real-time HW Data

Instruction trace

Third Party Tool Vendor Partnerships





Feature	Standard RISC-V	UltraSoC Trace Encoder
Trace	✓	✓
Filters	✓	✓
Counters		✓
Timestamps	✓	✓
Comparators	✓	✓
GPIO		✓
Security		✓
Data trace		✓
Interval timer		✓
		✓
Multiple retirement	✓	✓
Implicit return mode		✓
Whole system solution		✓
Branch prediction*		✓
Cycle-accurate tracing*		✓



Summary



- RISC-V eco-system maturing
 - Development tools & infrastructure available
 - Standardization moving fast
 - Both commercial and open-source
- Determining program behavior is not always possible using source level debugging
- Understanding program behavior in-field and realtime is needed
- An efficient trace scheme provides this
- Couple this with a holistic non-intrusive monitoring infrastructure provides the means of understanding complete SoC behavior