Processor Trace in a Holistic World

DAC-2018 San Francisco

RISC-V Foundation Booth

Processor Trace in a Holistic ~~World~~ SoC

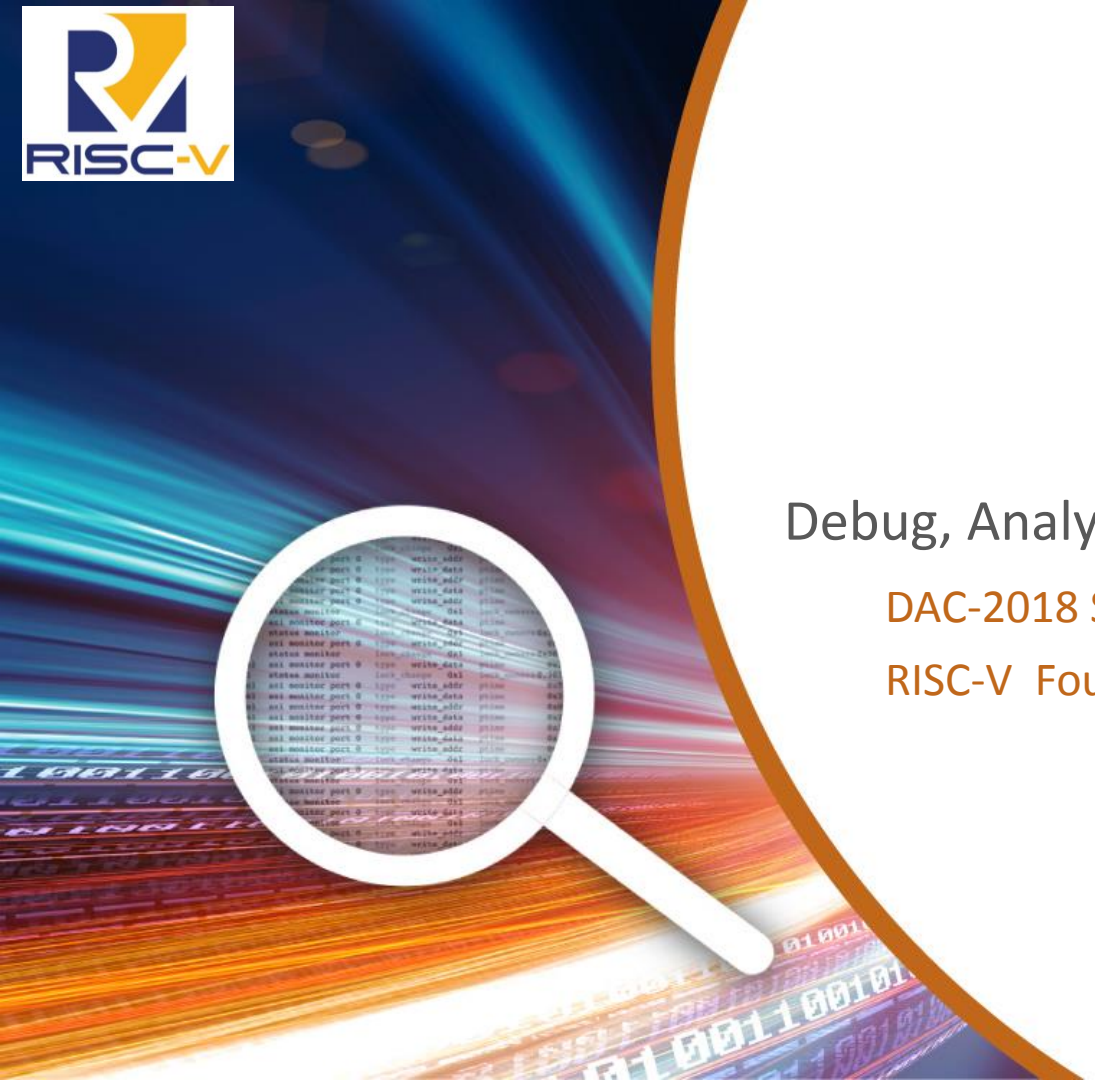
DAC-2018 San Francisco

RISC-V Foundation Booth

~~Processor~~ Debug, Analytics and Trace in a Holistic SoC

DAC-2018 San Francisco

RISC-V Foundation Booth



Debug, Analytics and Trace in a Holistic SoC

DAC-2018 San Francisco

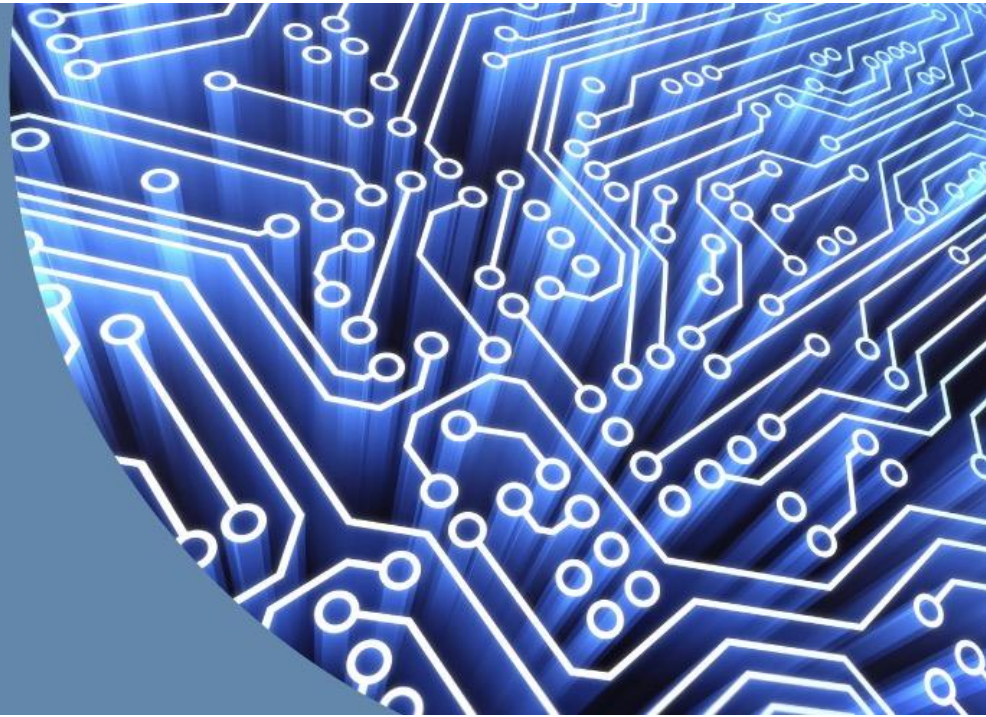
RISC-V Foundation Booth

Post-Silicon !
Debug, Analytics and Trace in a Holistic SoC

DAC-2018 San Francisco

RISC-V Foundation Booth

- Overview
- Holistic SoC
- Architectural Overview
- Example Scenarios
- RISC-V Processor Branch Trace
- Summary



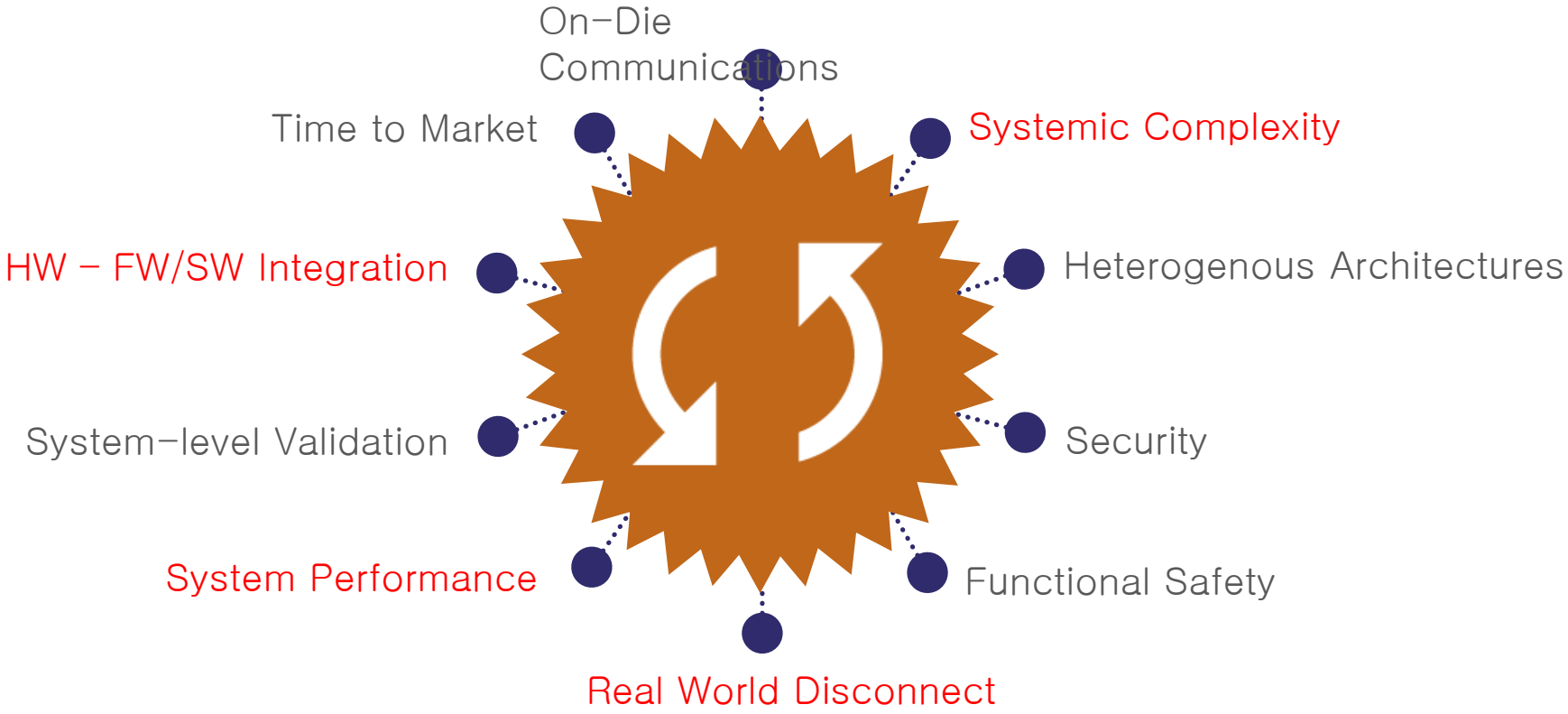
UltraSoC Overview

- Founded 2009
- VC-funded start-up
  - 2017 - Round of $7M
  - New Chairman October 2017
    Alberto Sangiovanni-Vincentelli
- Headquarters in Cambridge UK
- 44 patents
- ~35 employees
- Seasoned management team
- Key partners & ecosystem
- Proven technology and product-market fit

Tier-1 Automotive

ARMv8 Server

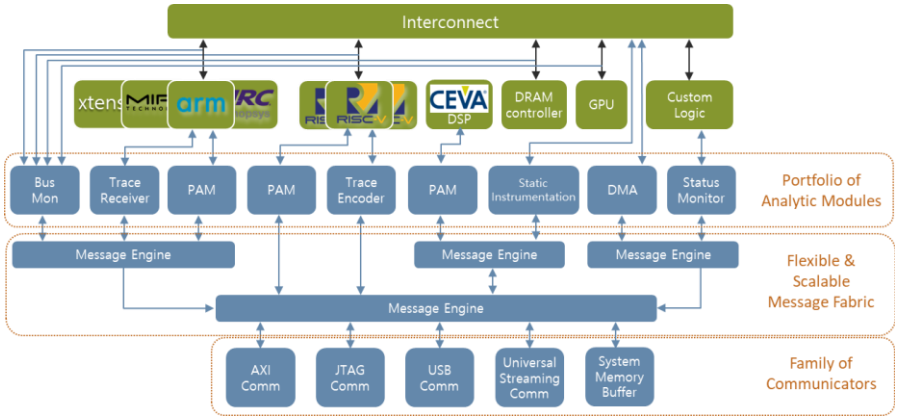
Challenges


On-Die Communications
Time to Market
Systemic Complexity
HW – FW/SW Integration
Heterogenous Architectures
System-level Validation
Security
System Performance
Functional Safety
Real World Disconnect

On-chip Analysis

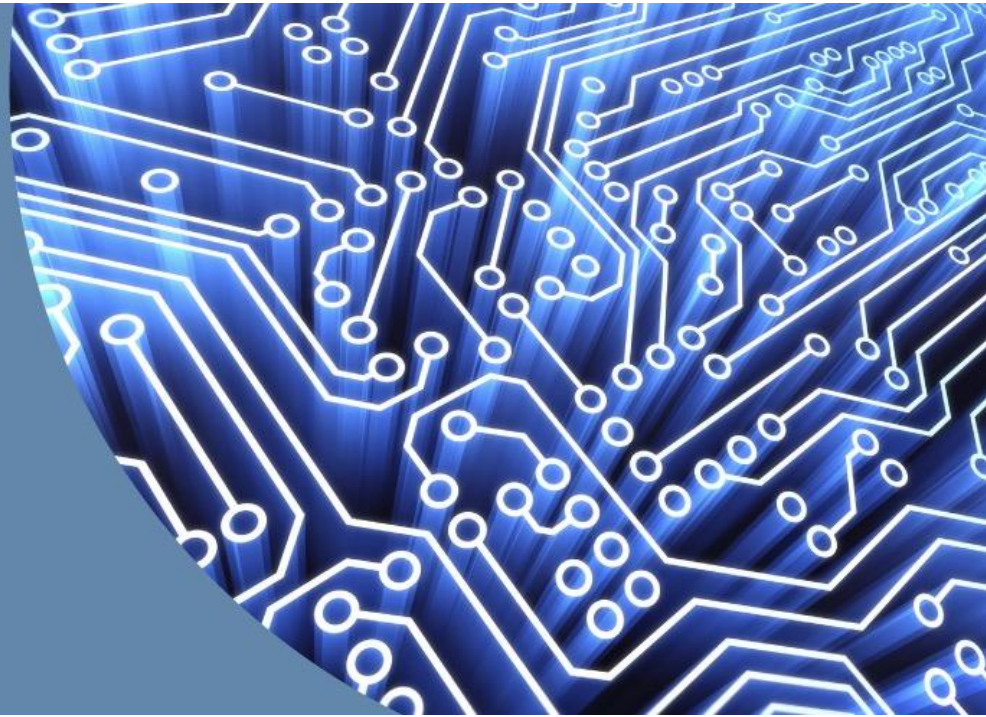
A coherent architecture to debug, develop, optimize & secure

- Full SoC visibility, HW & SW. System level
- Support “all” processor architectures: Freedom of IP selection
- Real-time & non-intrusive
- Advanced analytics & forensics
- Power/Performance optimization
- “in life” analytics & SLA compliance
- Supports Functional Safety
- Supports Bare Metal Security™
- High-speed debug over USB or SerDes



On-chip Analytics and Debug via RTL IP and SW

- Overview
- **Holistic SoC**
- Architectural Overview
- Example Scenarios
- RISC-V Processor Branch Trace
- Summary



Define Holistic

ho·lis·tic

/hōˈlistik/

adjective PHILOSOPHY

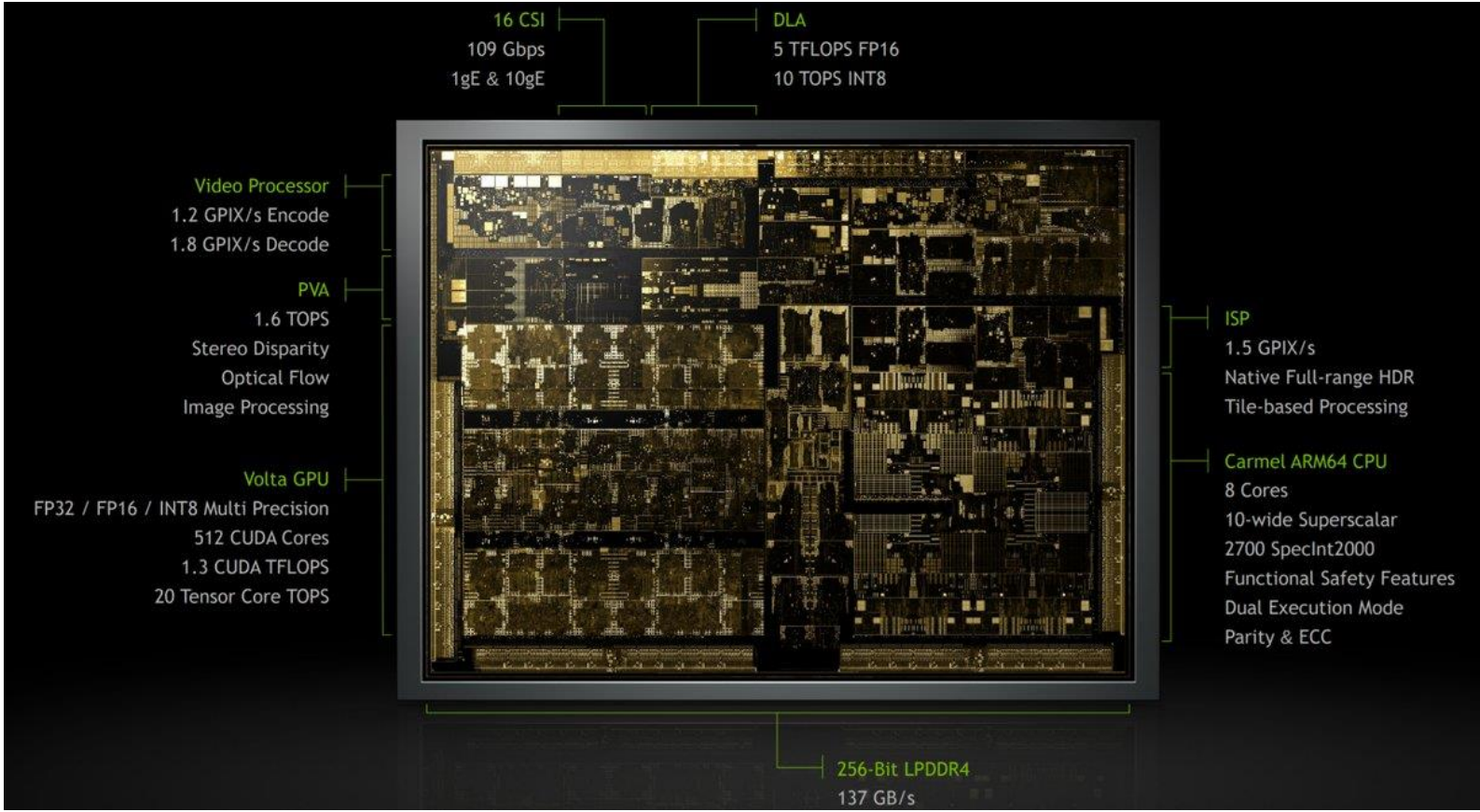
characterized by comprehension of the parts of something as intimately interconnected and explicable only by reference to the whole.

Define SoC?

"The first true SOC appeared in a Microma watch in 1974 when Peter Stoll integrated the LCD driver transistors as well as the timing functions onto a single Intel 5810 CMOS chip."

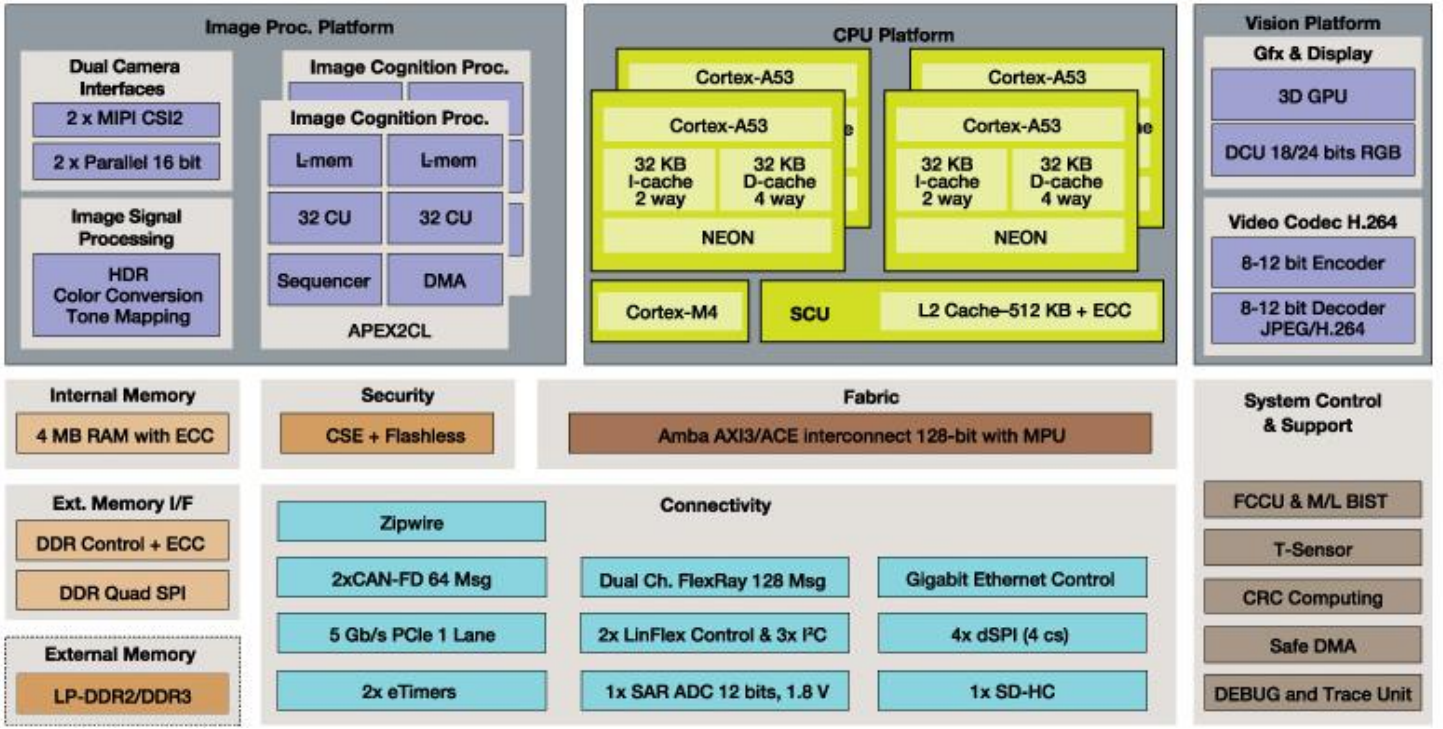
http://www.computerhistory.org

Modern SoC - Nvidia Carmel

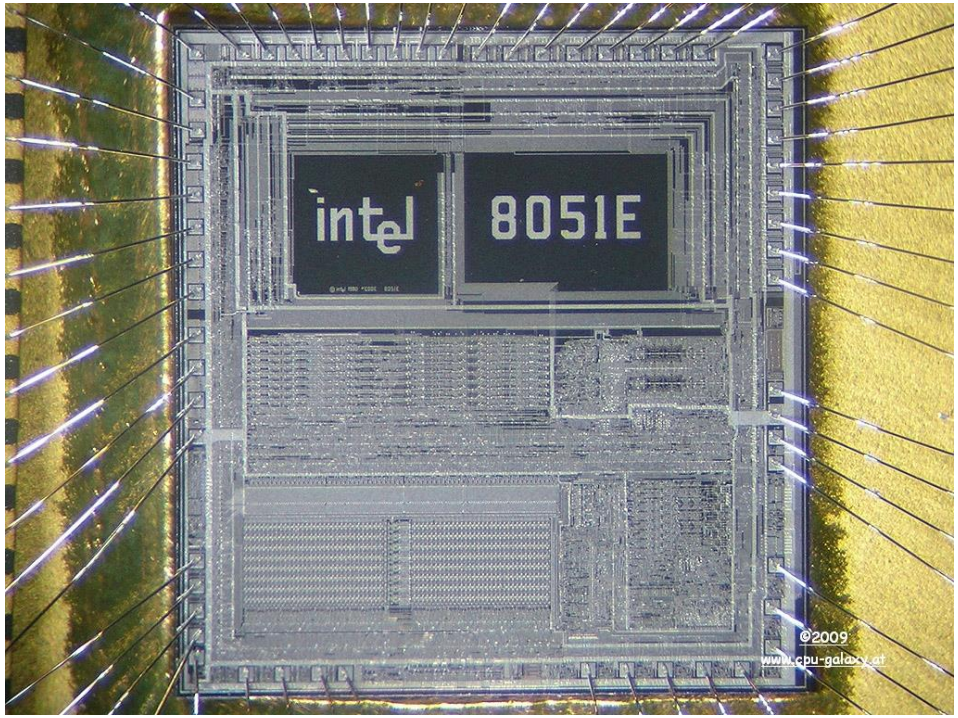


Modern SoC

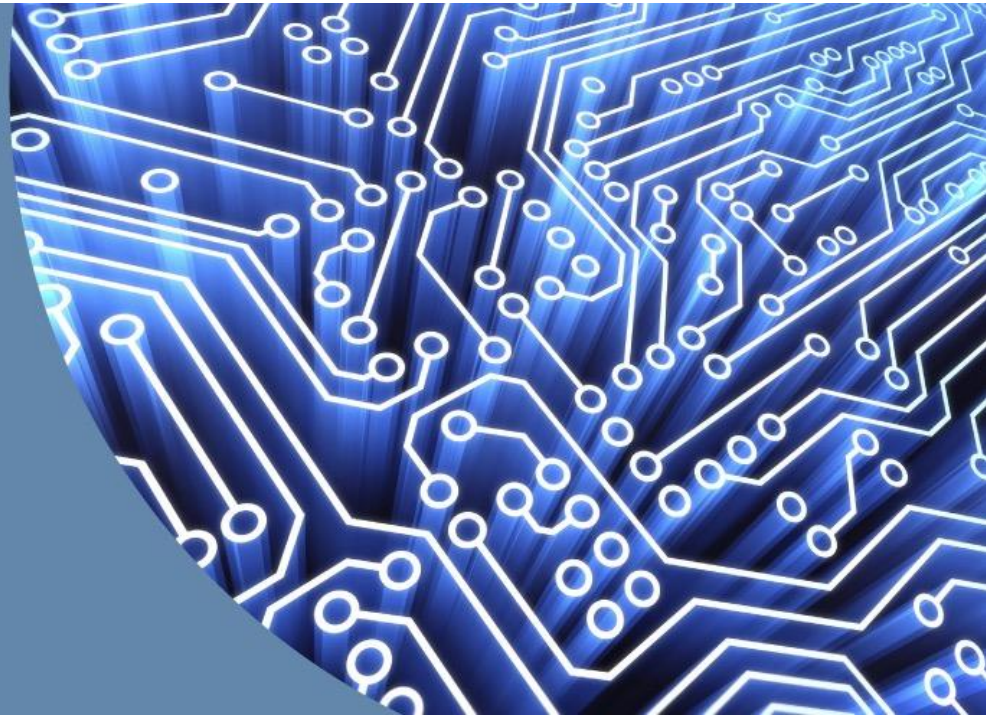
S32V234 Block Diagram



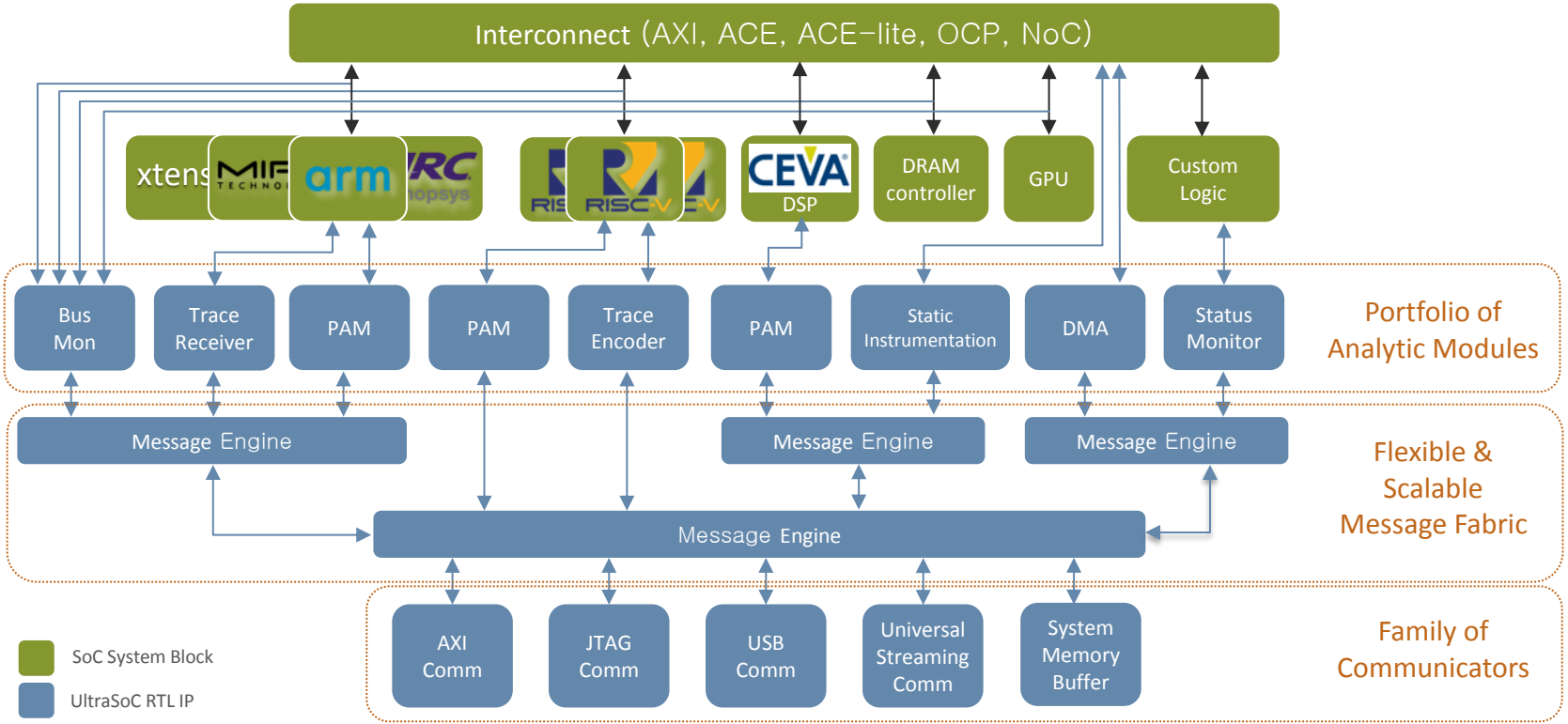
SoC Observability has Always Been an Issue



- Overview
- Holistic SoC
- **Architectural Overview**
- **Example Scenarios**
- RISC-V Processor Branch Trace
- Demo Summary

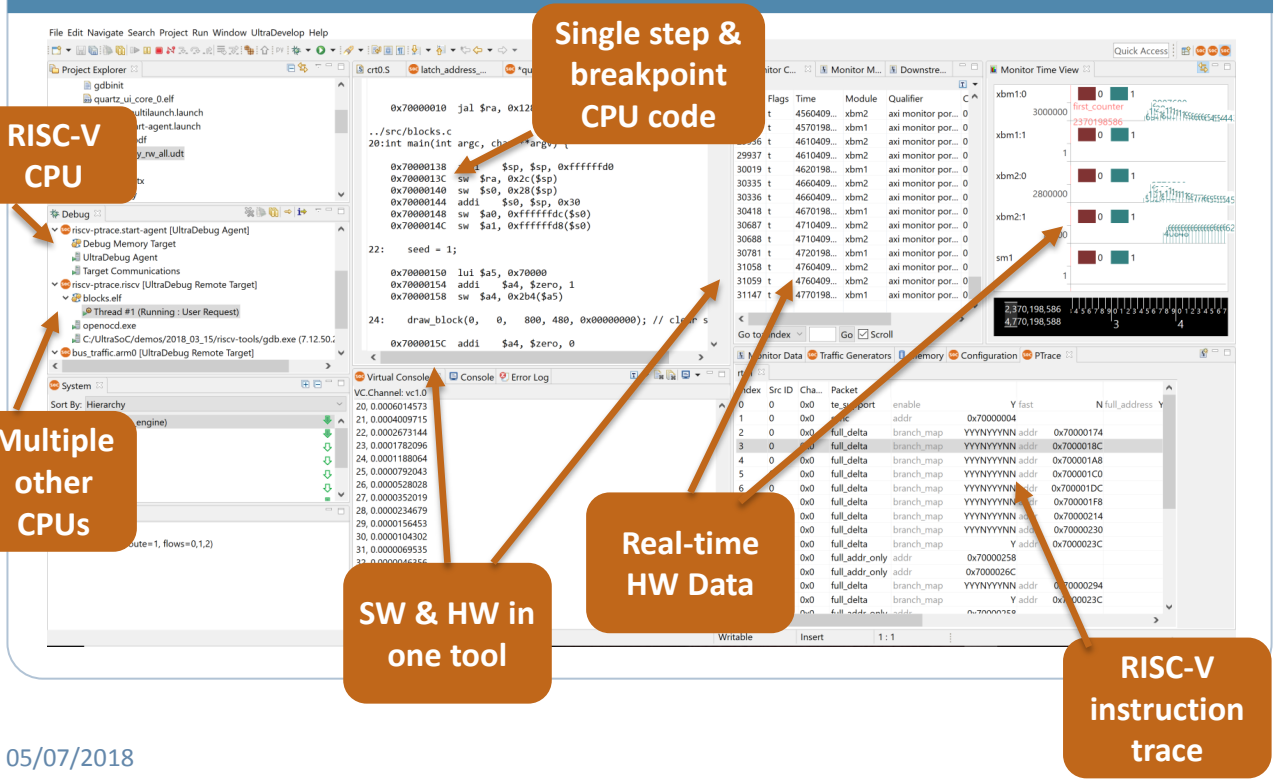


Advanced Analytics for the Holistic SoC



Software Tools for Data Driven Analysis

Eclipse based UltraDevelop IDE

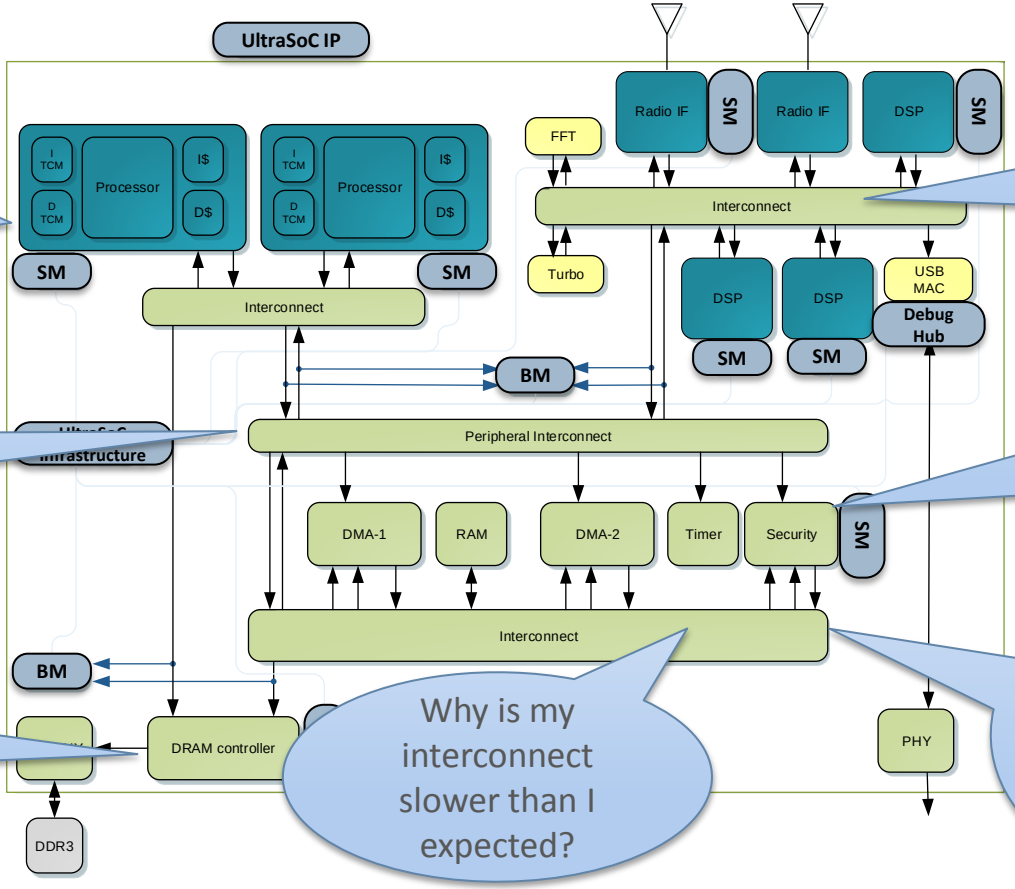


Third Party Tool Vendor Partnerships

UltraDevelop interfaces with almost all common validation and verification solutions:

ARM
cadence
CEVA
Green Hills SOFTWARE
eclipse
IMG Systems
LAUTERBACH DEVELOPMENT TOOLS
SYNOPSYS
TELEDYNE LECROY Everywhereyoulook
imperas

Technical Problems UltraSoC Solves



Why is the CPU not as fast as expected?

What is the mismatch between the host & the DSP?

Why do some DMA transfers take too long?

Can I trust system security?

What is going on with my memory controller?

Why is my interconnect slower than I expected?

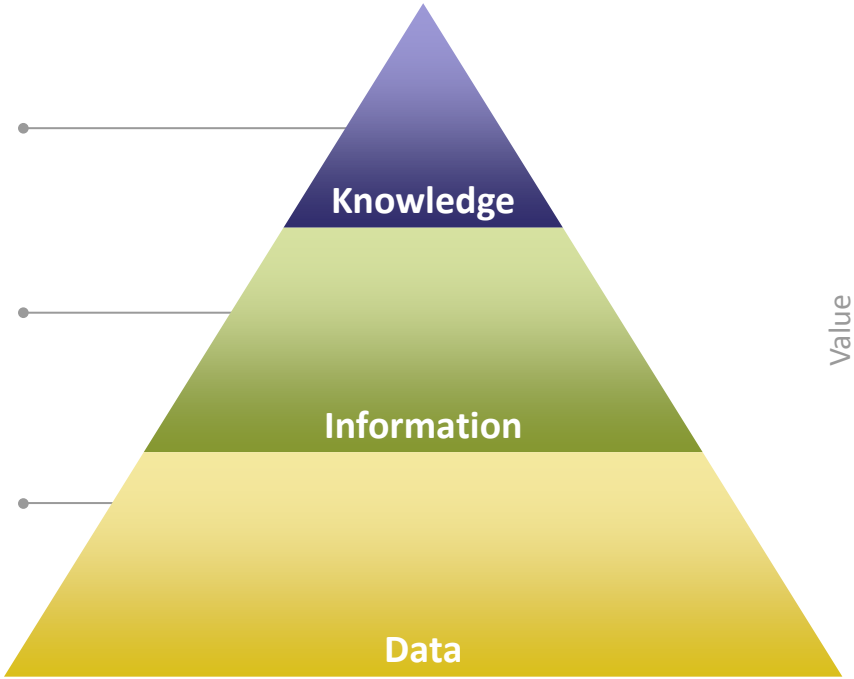
Why does the system occasionally hang or deadlock?

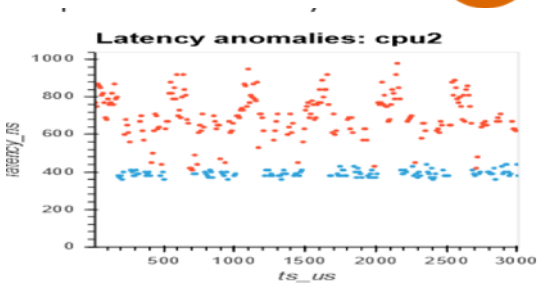
Actionable Insights Across the Whole SoC

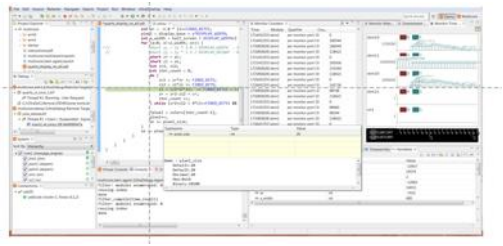
UltraSoC delivers actionable *insights*

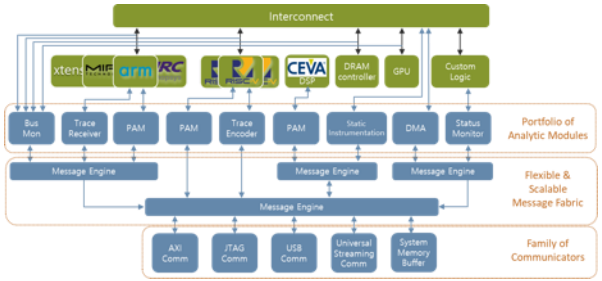
With system-wide *understanding*

From rich *data* across the whole SoC

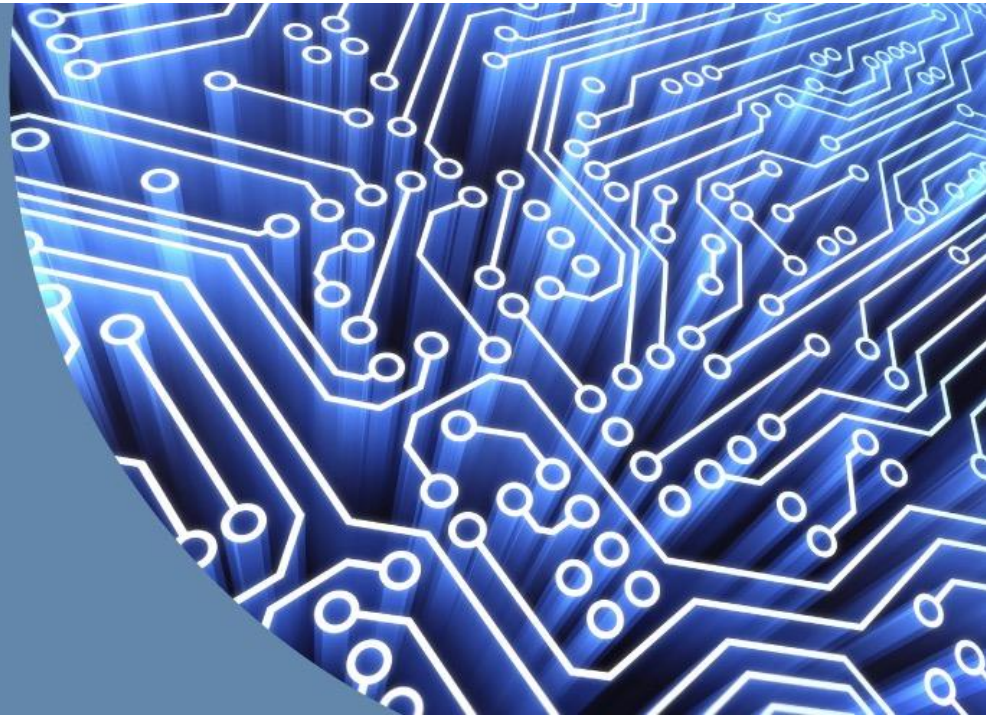








- Overview
- Holistic SoC
- Architectural Overview
- Example Scenarios
- **RISC-V Processor Branch Trace**
- Demo Summary



Need for Processor Branch Trace in RISC-V

- Detailed presentation at: https://riscv.org/2018/05/risc-v-workshop-in-barcelona-proceedings/
- In complex systems understanding program behaviour is not easy under real-time conditions
- Software/Firmware rarely behave as planned interactions with other cores' software, peripherals, real-time events, poor implementation, ...
- Using a debugger is not always possible since realtime behaviour can be affected
- Providing visibility of program execution real-time is important
- One method of achieving this is via Processor Branch Trace
  - Encoder
  - Filtering and triggering schemes

Processor Branch Trace

- Track execution from a known address and send messages about deltas taken by program
- Deltas result from *jump, call, return* and *branch* type instructions, interrupts and exceptions
- RISC-V instructions are executed unconditionally or their execution can be determined based on the program. Instructions between the deltas are assumed to be executed sequentially
- No need to report them via the trace, only whether the branch was taken or not and the address of indirect branches or jumps taken

Interrupts and Exceptions

- Interrupts can occur asynchronously to a program's execution
- Exceptions can be thought of in the same way
- The decoder generally does not know where an interrupt occurs in the instruction sequence. The trace encoder reports the address and an indication of the asynchronous destination
- When an interrupt or exception occurs, or the processor is halted, the final instruction executed must be traced

Trace Control

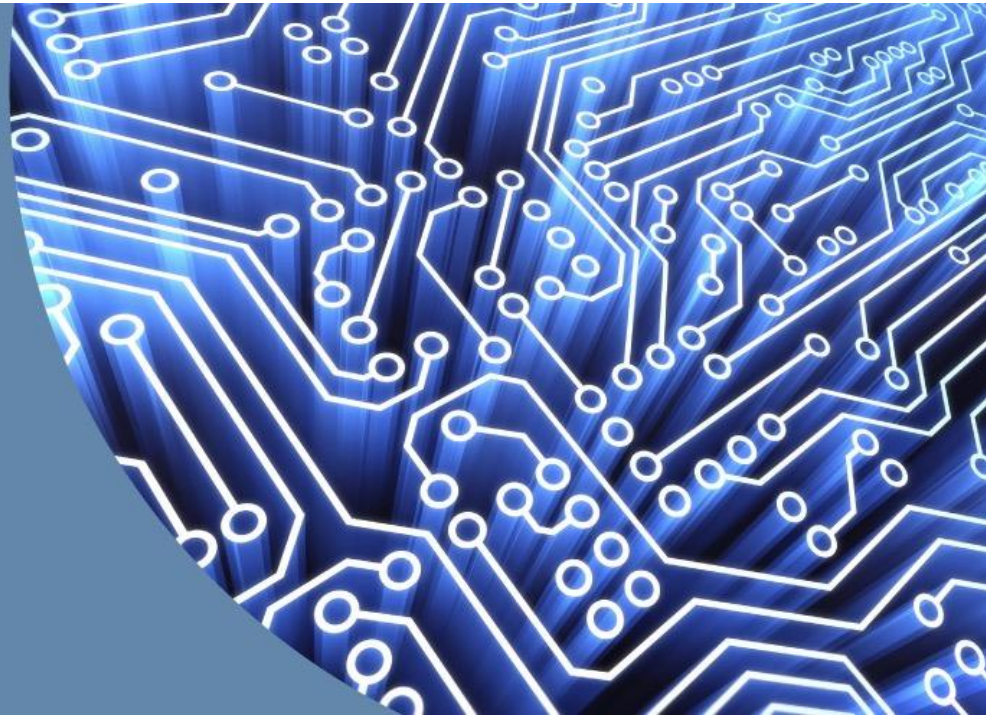
- Controlling when trace is generated is critical for reducing the volume of trace data
- Filters are required
- Filters enable trace based on: address range, privilege level, and interrupt service routines
- Other examples
  - Trace for fixed period of time
  - Start trace when external (to the encoder) event detected
  - Stop trace when an external (to the encoder) event detected

Encoding Efficiency

Benchmark	Instructions	Packets	Payload Bytes	Bits per instruction
dhrystone	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**

- Validated in FPGA, soon in silicon
- 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

- Overview
- Holistic SoC
- Architectural Overview
- Example Scenarios
- Processor Branch Trace
- Trace Encoder Interface
- **Summary**

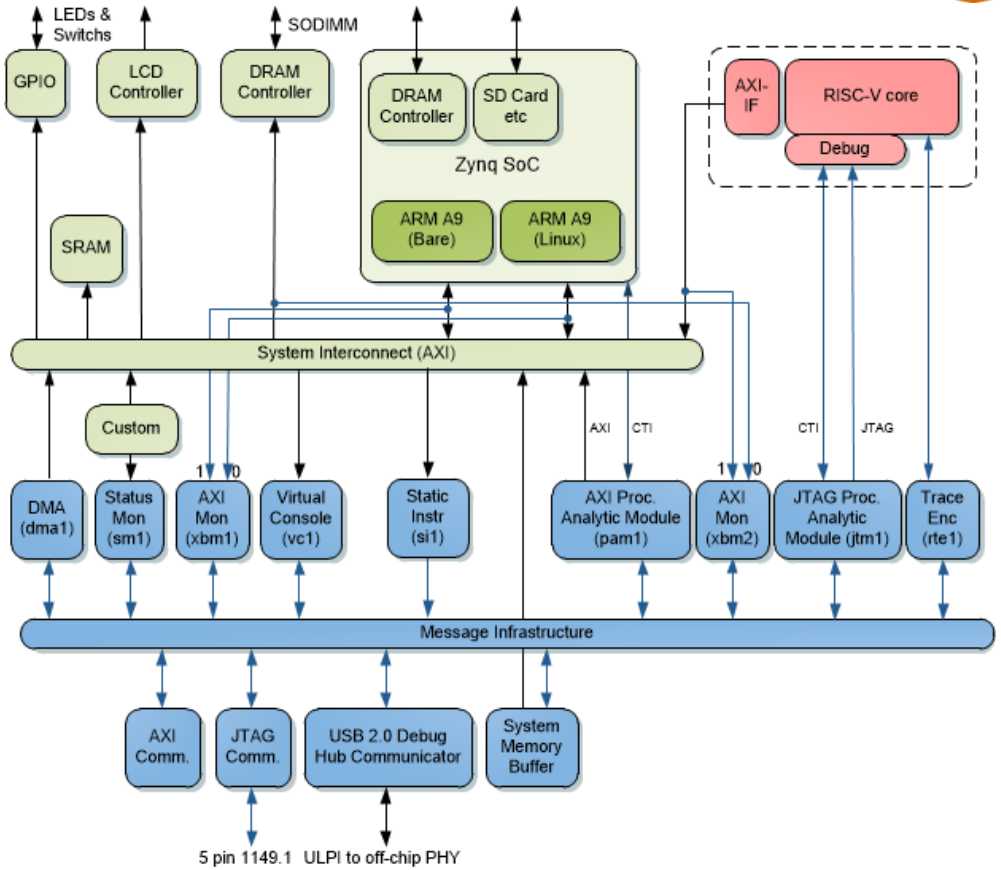


UltraSoC and RISC-V

- UltraSoC has been a member of the RISC-V Foundation since 2016
  - Leadership role in Debug Working Group
  - Trace specification, announced @ DAC 54, offered as open standard spec
- Industry's first and only commercial RISC-V debug & monitoring products
  - Including trace & run control
  - Scalable from lightweight IoT to heterogeneous multicore designs
- Publicly endorsed by major processor and tools vendors
  - Andes, Baysand, Codasip, Esperanto, Imperas, Lauterbach, Microsemi, Roa Logic, SiFive, Syntacore
- FPGA demonstrator available, IDE integrated
- Silicon proven

RISC-V Demo System Architecture

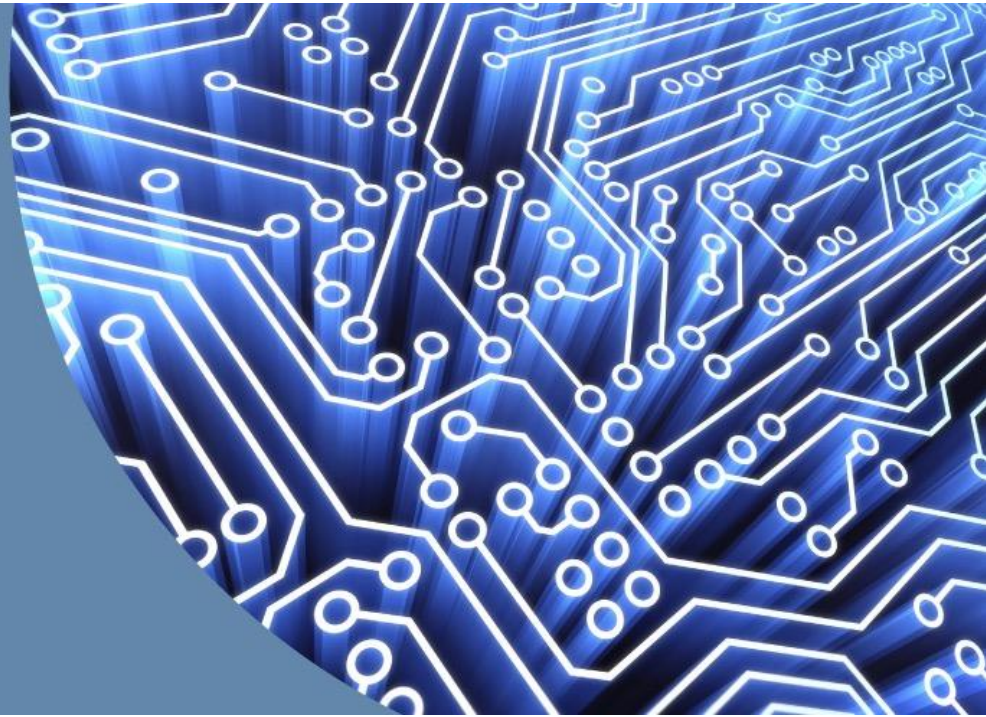
- Zynq FPGA platform
  - RV32 RISC-V
  - Custom logic
  - 2x ARM A9
  - UltraSoC architecture
- Demo shows:
  - Trace Encoder configuration
  - RISC-V run control
  - Capture of instruction trace
  - Disassembly of instruction trace
  - Bus state
  - Performance histogram



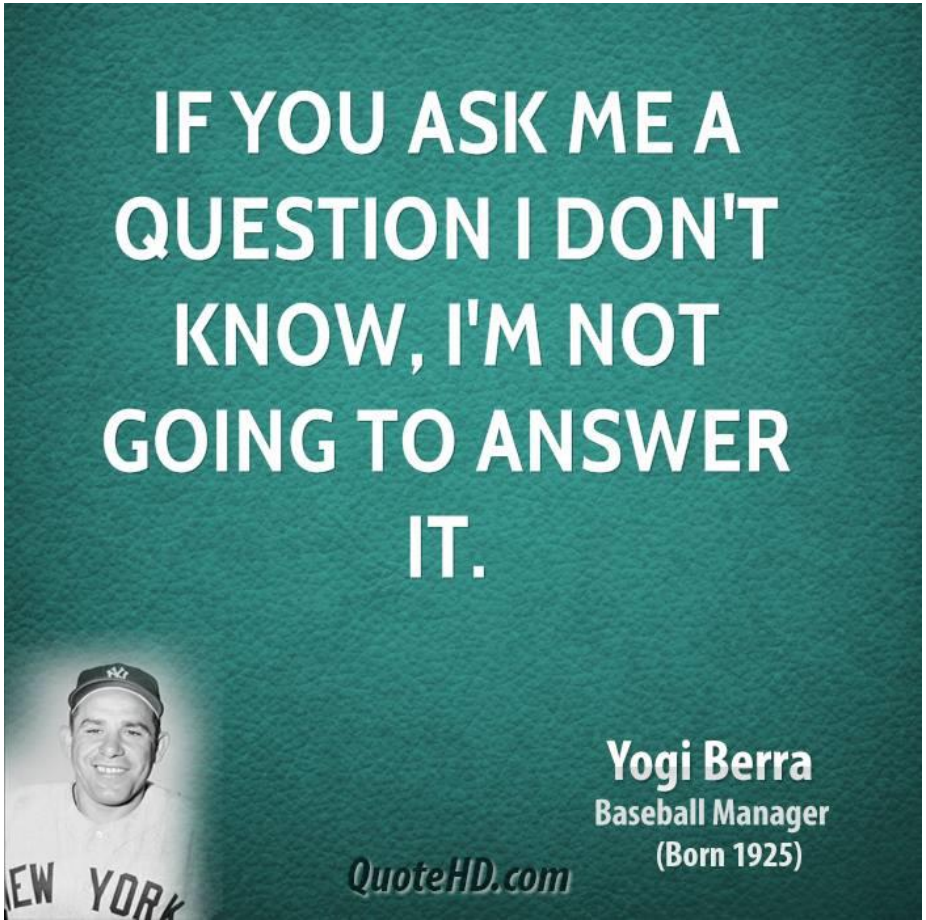
Contact details:

Randy Fish

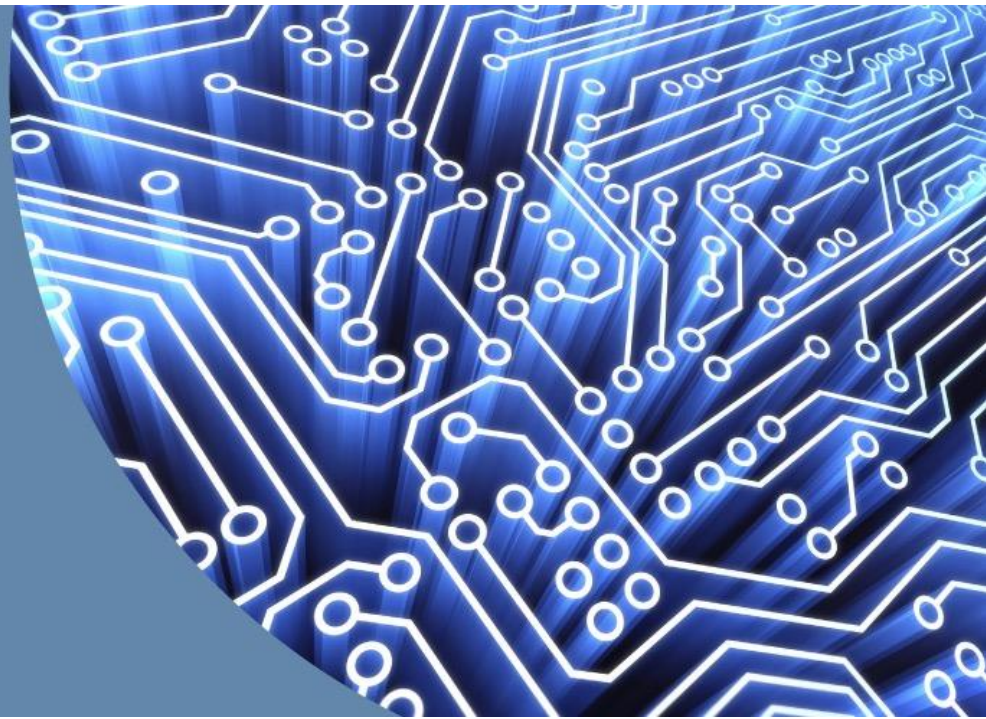
randy.fish@ultrasoc.com
www.ultrasoc.com
@UltraSoC



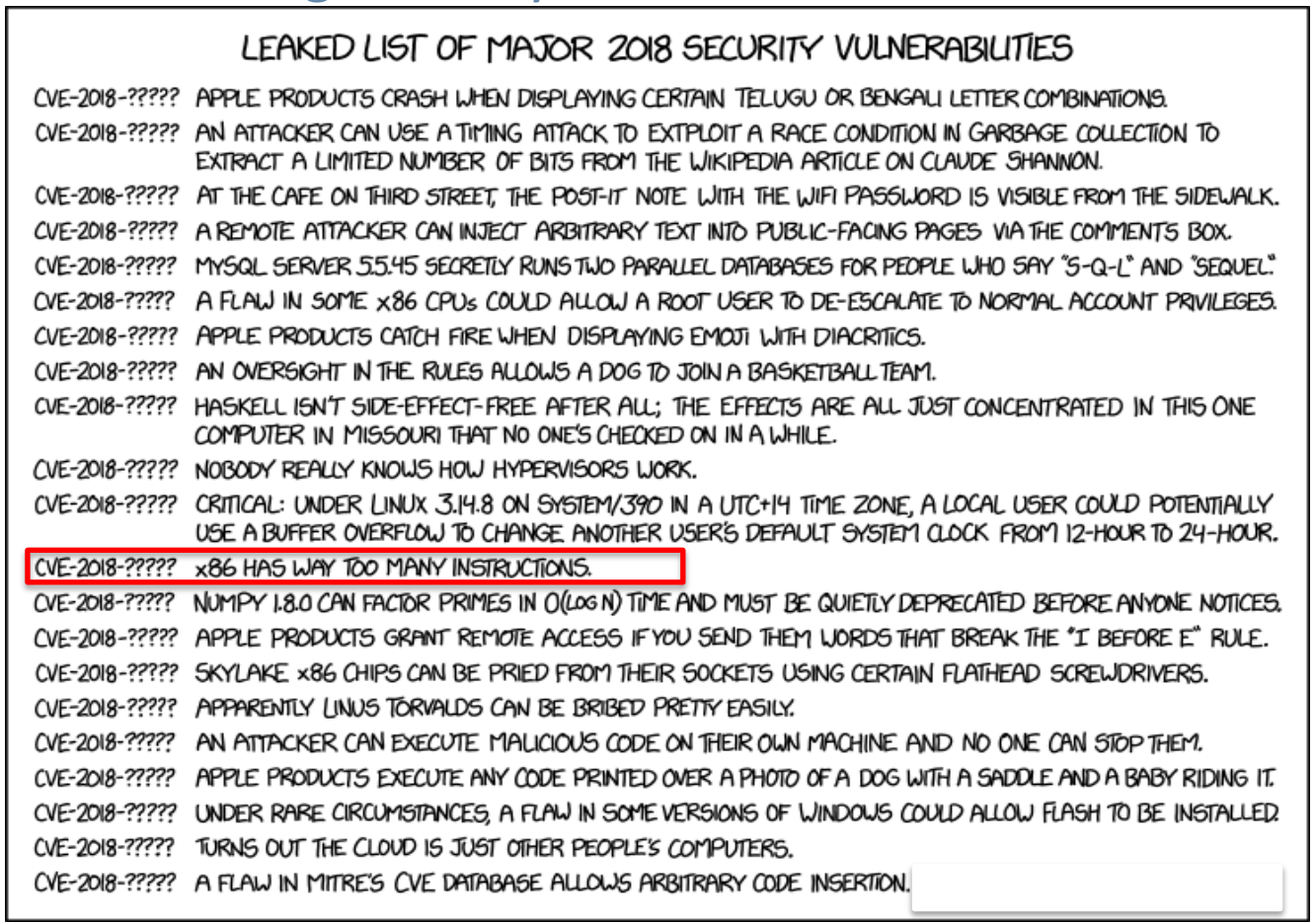
Q&A


IF YOU ASK ME A QUESTION I DON'T KNOW, I'M NOT GOING TO ANSWER IT.
Yogi Berra
Baseball Manager
(Born 1925)
QuoteHD.com

Backup



Concerning Security



LEAKED LIST OF MAJOR 2018 SECURITY VULNERABILITIES

CVE-2018-????? APPLE PRODUCTS CRASH WHEN DISPLAYING CERTAIN TELUGU OR BENGALI LETTER COMBINATIONS.

CVE-2018-????? AN ATTACKER CAN USE A TIMING ATTACK TO EXTPLOIT A RACE CONDITION IN GARBAGE COLLECTION TO EXTRACT A LIMITED NUMBER OF BITS FROM THE WIKIPEDIA ARTICLE ON CLAUDE SHANNON.

CVE-2018-????? AT THE CAFE ON THIRD STREET, THE POST-IT NOTE WITH THE WIFI PASSWORD IS VISIBLE FROM THE SIDEWALK.

CVE-2018-????? A REMOTE ATTACKER CAN INJECT ARBITRARY TEXT INTO PUBLIC-FACING PAGES VIA THE COMMENTS BOX.

CVE-2018-????? MYSQL SERVER 5.5.45 SECRETLY RUNS TWO PARALLEL DATABASES FOR PEOPLE WHO SAY "S-Q-L" AND "SEQUEL."

CVE-2018-????? A FLAW IN SOME x86 CPUs COULD ALLOW A ROOT USER TO DE-ESCALATE TO NORMAL ACCOUNT PRIVILEGES.

CVE-2018-????? APPLE PRODUCTS CATCH FIRE WHEN DISPLAYING EMOJI WITH DIACRITICS.

CVE-2018-????? AN OVERSIGHT IN THE RULES ALLOWS A DOG TO JOIN A BASKETBALL TEAM.

CVE-2018-????? HASKELL ISN'T SIDE-EFFECT-FREE AFTER ALL; THE EFFECTS ARE ALL JUST CONCENTRATED IN THIS ONE COMPUTER IN MISSOURI THAT NO ONE'S CHECKED ON IN A WHILE.

CVE-2018-????? NOBODY REALLY KNOWS HOW HYPERVISORS WORK.

CVE-2018-????? CRITICAL: UNDER LINUX 3.14.8 ON SYSTEM/390 IN A UTC+14 TIME ZONE, A LOCAL USER COULD POTENTIALLY USE A BUFFER OVERFLOW TO CHANGE ANOTHER USER'S DEFAULT SYSTEM CLOCK FROM 12-HOUR TO 24-HOUR.

CVE-2018-????? x86 HAS WAY TOO MANY INSTRUCTIONS.

CVE-2018-????? NUMPY 1.8.0 CAN FACTOR PRIMES IN O(LOG N) TIME AND MUST BE QUIETLY DEPRECATED BEFORE ANYONE NOTICES.

CVE-2018-????? APPLE PRODUCTS GRANT REMOTE ACCESS IF YOU SEND THEM WORDS THAT BREAK THE "I BEFORE E" RULE.

CVE-2018-????? SKYLAKE x86 CHIPS CAN BE PRIED FROM THEIR SOCKETS USING CERTAIN FLATHEAD SCREWDRIVERS.

CVE-2018-????? APPARENTLY LINUS TORVALDS CAN BE BRIBED PRETTY EASILY.

CVE-2018-????? AN ATTACKER CAN EXECUTE MALICIOUS CODE ON THEIR OWN MACHINE AND NO ONE CAN STOP THEM.

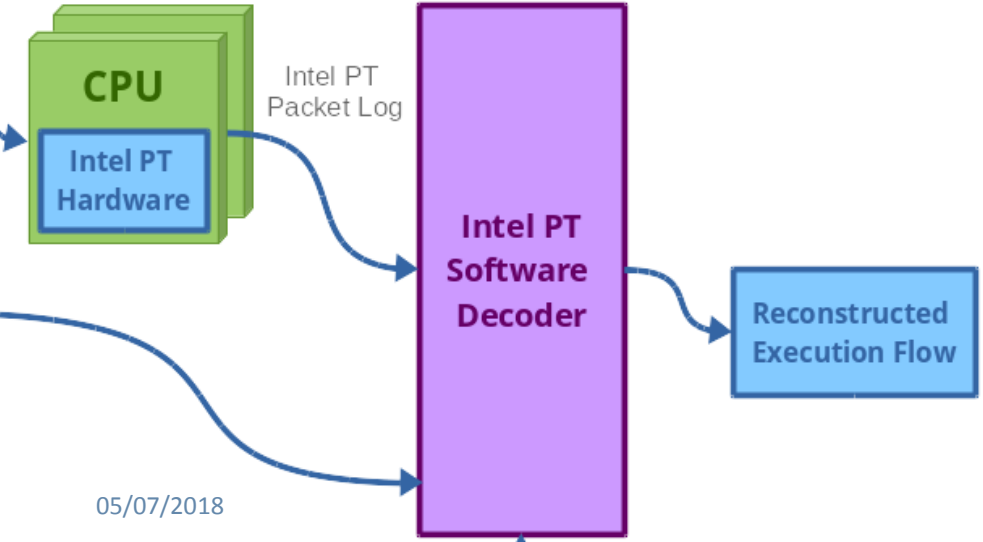
CVE-2018-????? APPLE PRODUCTS EXECUTE ANY CODE PRINTED OVER A PHOTO OF A DOG WITH A SADDLE AND A BABY RIDING IT.

CVE-2018-????? UNDER RARE CIRCUMSTANCES, A FLAW IN SOME VERSIONS OF WINDOWS COULD ALLOW FLASH TO BE INSTALLED.

CVE-2018-????? TURNS OUT THE CLOUD IS JUST OTHER PEOPLE'S COMPUTERS.

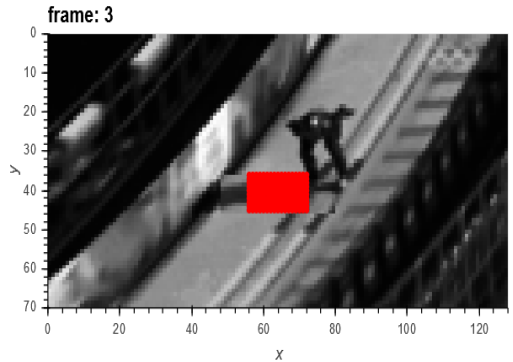
CVE-2018-????? A FLAW IN MITRE'S CVE DATABASE ALLOWS ARBITRARY CODE INSERTION.

Processor Trace

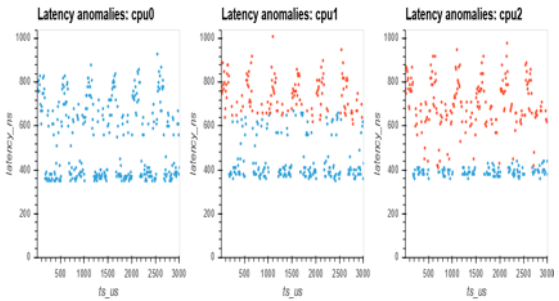


Embedded Analytics

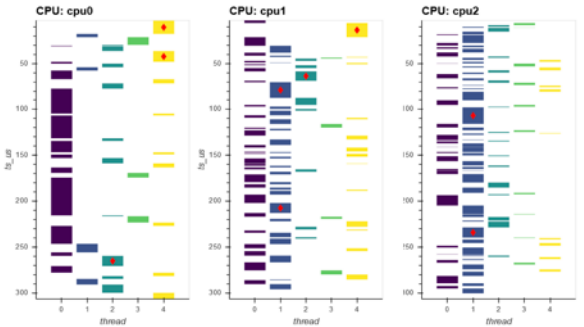
Safety
HW “stuck pixel detect”



Security
HW-based attack detect

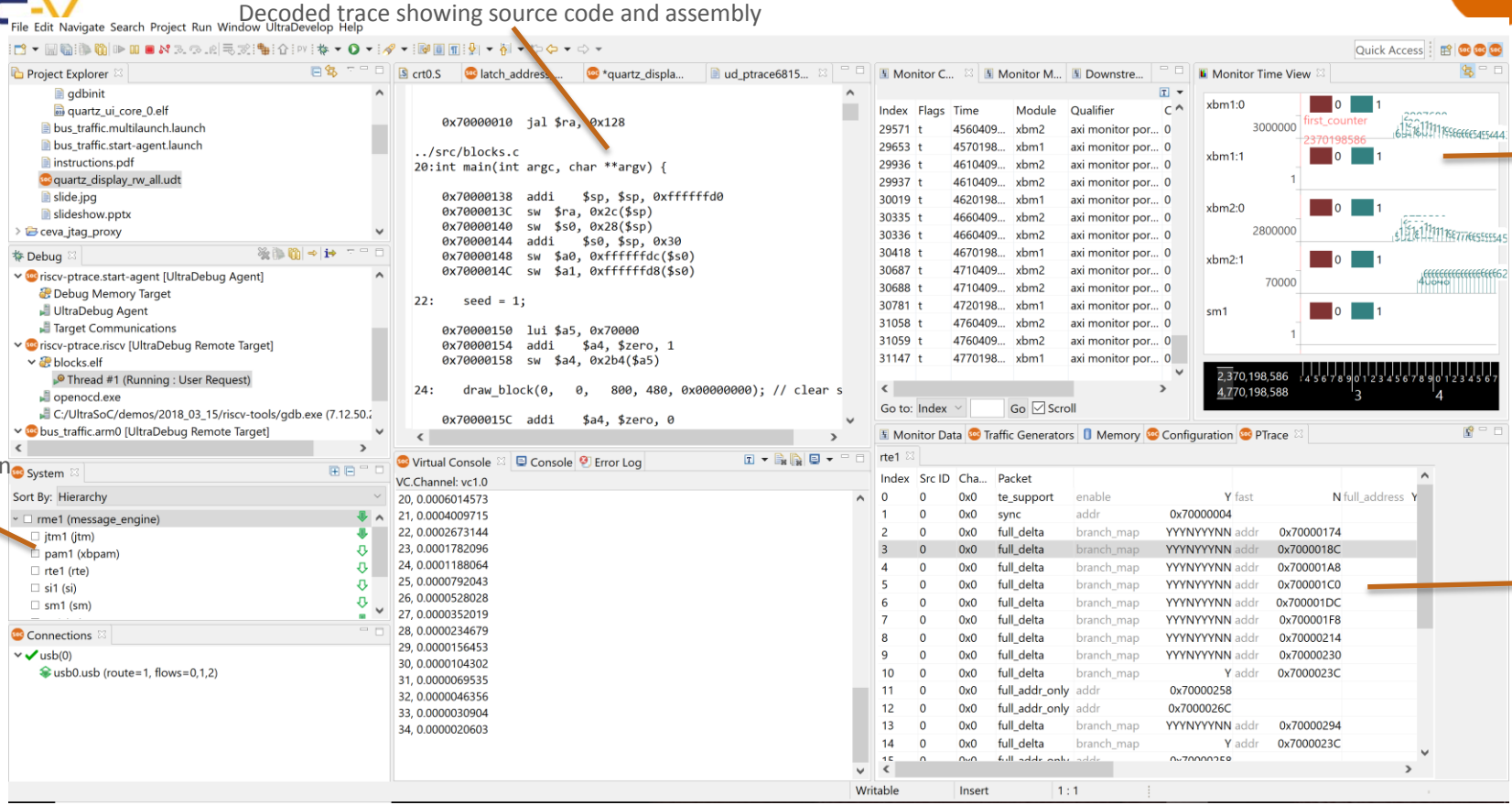


Performance
Run-time server optimization



- **Non-intrusive:** No performance impact or “warning”
- **Hardware:** Fast, react at HW timescale; invisible to software
- **Visibility:** Analyze software and system everywhere in SoC, see any problem

UltraSoC IDE



Bus activity

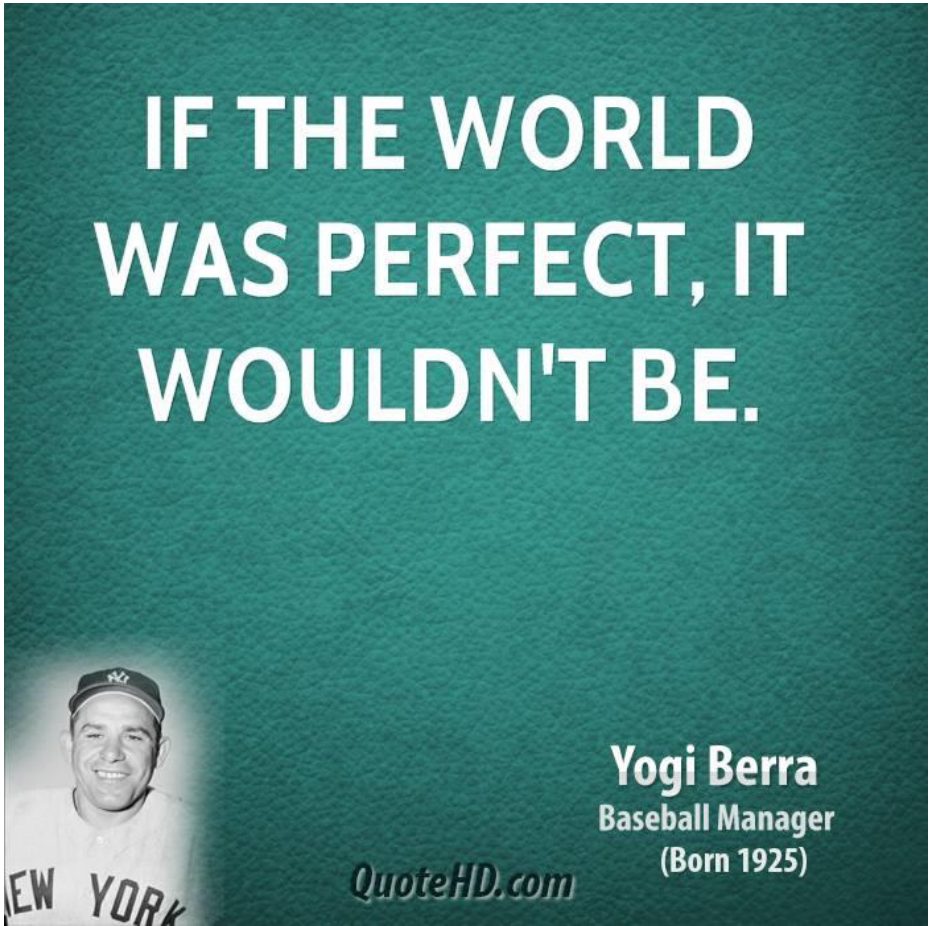
Control configuration

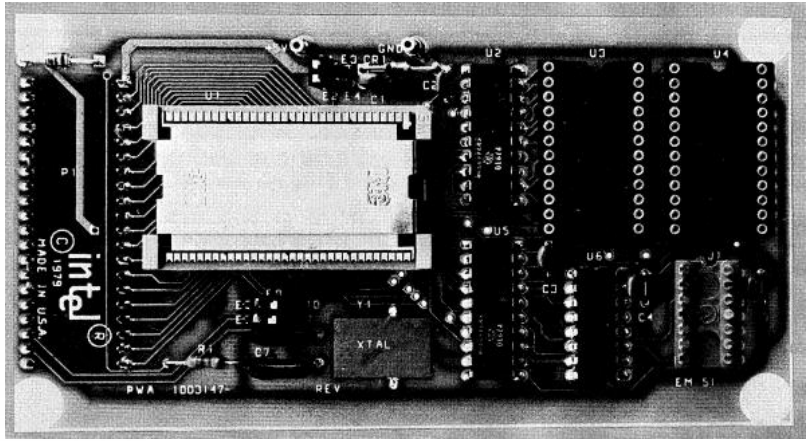
Trace Packets

- Clearly stated messages on each slides
- Netspeed integration: Show block diagram before/after
- Long Tail , which tail?
- The Challenge
  - Heterogenous, multicore, DSP, … DSA, busses, data rates and width
  - Debug of bits → transactions → ???
  - Show a simple slides of history and future of analytics
  - Maybe a hierarchy where analytics consists of debug?
  - The challenge is debugging FW and SW in a complex SoC or maybe even a multi-die, multi chip solution
  - OoO,
  - Workloads of tomorrow are unknown. You cannot predict and systems need to be optimized with real data. Thus, tuning with in-silicon analytics
- What we offer
  - Show blow-up of a few critical IP blocks and highlight configurability at compile time and run time
  - Cross triggering is reconfigurable at runtime (need slides for cross triggering
- Highlight key words graphically – Nonintrusive, heterogeneous, …

- Challenges change over time
  - Single core
  - Multi core
  - Hetero generous
  - Custom accelerators
  - Memory throughput
  - You cannot do all the analysis off-chip
    - Trace got there a long time ago
- SoCs and FPGAs
- Sim, Em, Pro, Post-Si, In-Field

- Visibility of whole SoC is critical. Maybe multiple SOCs
- Build time vs runtime: what can be done at each
- Analytics – debug ++, system level, not just a processor
- Analytics are critical for post-silicon optimization
- Workloads are unknow and systems are tuned to a workload during design
- Demo slide


IF THE WORLD WAS PERFECT, IT WOULDN'T BE.
Yogi Berra
Baseball Manager
(Born 1925)
QuoteHD.com



EM-51
8051 EMULATION BOARD

- **Emulates 8051/8751/8031 functions on a 2.75″ × 5.25″ board assembly**
- **Replaces the 8051/8751/8031 in prototype systems**
- **Plugs directly into 8051/8751/8031 sockets**
- **Includes a 2732A EPROM device for program memory**

The EM-51 emulation board is a small, ready-to-use microcomputer assembly that replaces an Intel 8051 family single-chip microcomputer in a prototype system. EM-51 includes sockets for 2716 or 2732 EPROMs, which substitute for the 8051/8751 on-chip program memory during prototype development. An Intel 2732A 4K x 8 EPROM is included with the board. With the memory in place, an EM-51 board becomes a full functional and electrical equivalent of the 8051 or 8751 microcomputer.

The 8051 is an SOC c1979