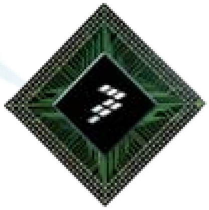
Nov 1, 2008

On-Chip Debugging of Multicore Systems



PN115

Jeffrey Ho

AP Technical Marketing, Networking Systems Division

freescale™
semiconductor
QorIQ

On-Chip Debugging of Multicore Systems: Objectives

►Objectives

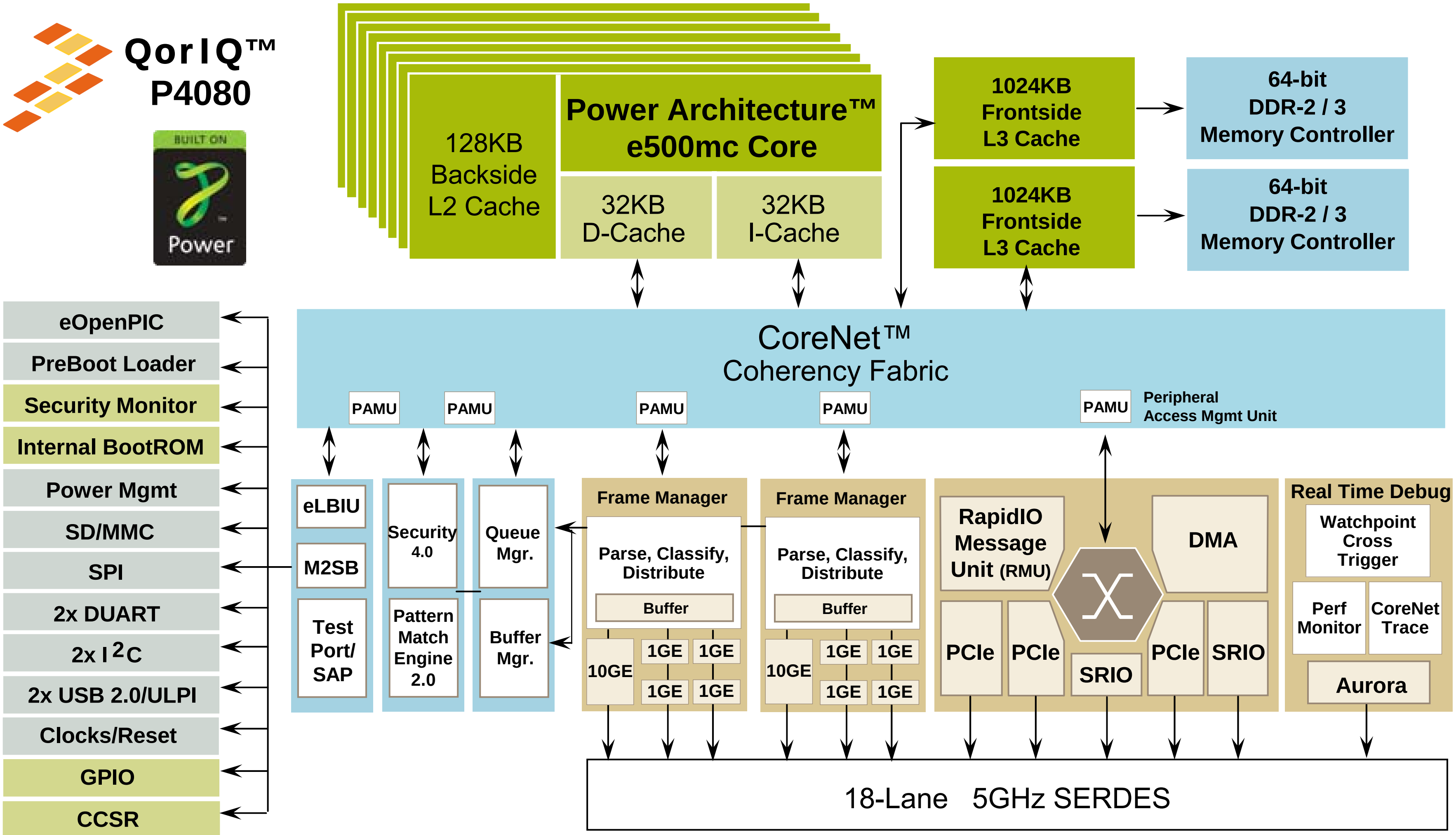
- Review impact of multicore on system debugging requirements
- Overview QorIQ™ P4080 on-chip debug support architecture
- Describe debug features of the e500mc core
- Describe debug features of the P4080 SoC platform
- Review run control, debug agent and trace debugging scenarios

Impact of Multicore Processors on System Debug

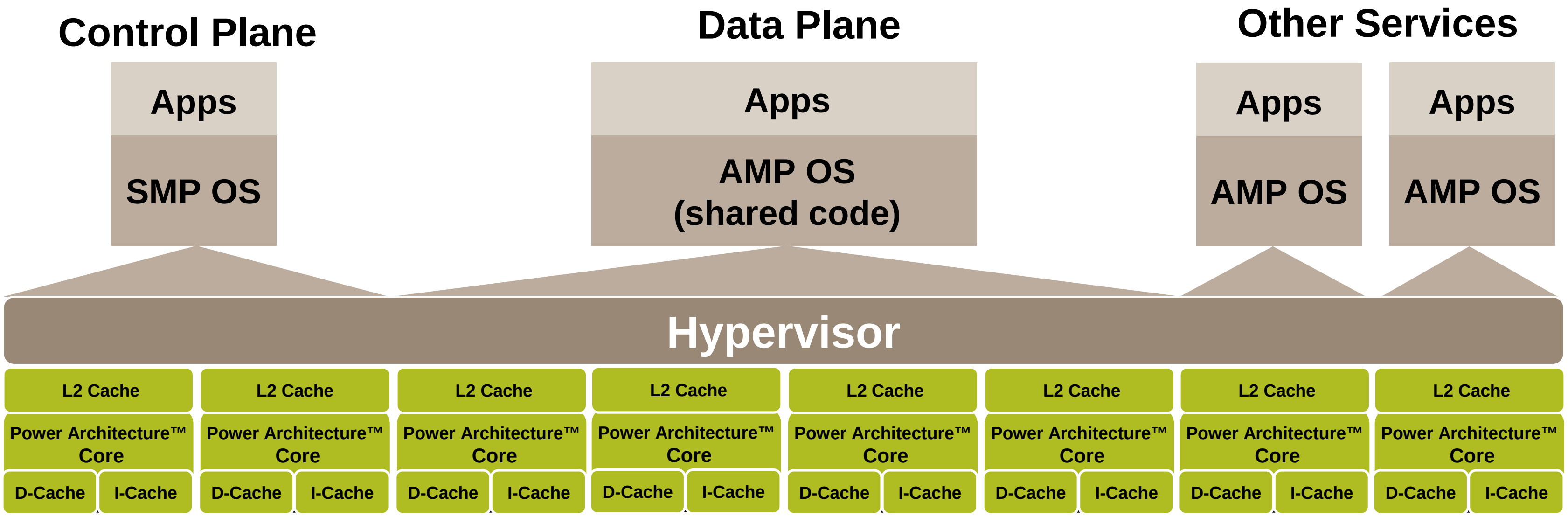
- Power Architecture® technology defines robust debug features
    - Implemented in MPC8548, MPC8572 and other PowerQUICC® III devices
- Growing requirement for higher levels of software instrumentation
    - Higher levels of visibility expected for software debug
    - Better integrated support desired for performance evaluation
- Single-chip multicore presents additional challenges
    - Higher integration of cores and peripherals
        - Reduces visibility for traditional tools such as logic analyzers
    - Applications typically more tightly coupled than before
        - Concurrency problems more likely to occur (or existing problems revealed)
        - Need to correlate code running on different cores with SoC interactions

QorIQ™ P4080 Block Diagram

QorIQ™
P4080



QorIQ™ Operating Systems and Debug



- Multiple SMP, AMP operating systems, commonly above a hypervisor
- Debug scenarios vary with level of software of interest
- Different debug tool types most effective for different scenarios
  - One size does **not** fit all for debug tools
  - Layers of debug tools may be most effective

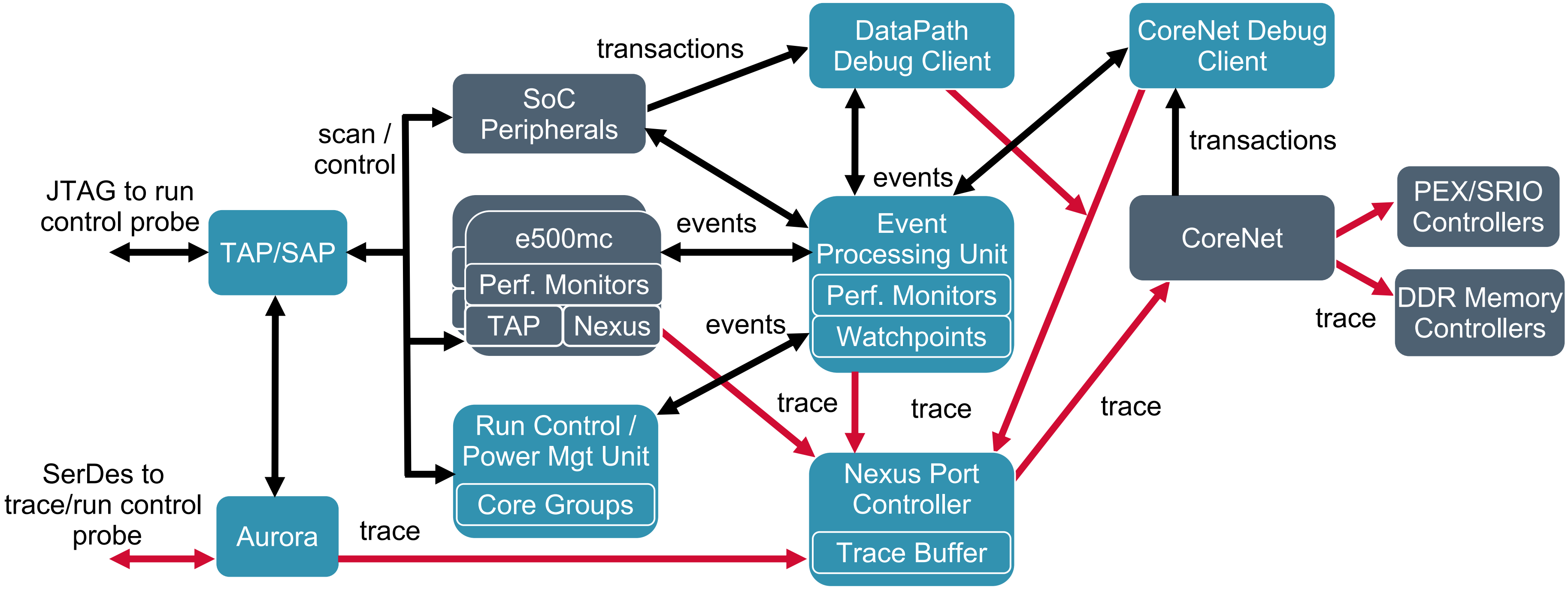
QorIQ™ P4080 Debug Features Overview

- P4080 provides more and better quality visibility for multicore debug
  - Programming model and debug registers visible to software agents
  - Run control enhanced over single core predecessors
    - Cross-core and SoC watchpoint triggering
  - High speed serial trace
    - e500mc and SoC trace clients
    - Nexus Instruction, Data, Data Acquisition, Watchpoint Trace
    - Utilizes industry-standard, high-bandwidth Aurora protocol

- Robust performance monitoring capabilities
  - Dedicated core and SoC platform counters

- Supported by tools from Freescale and leading providers

QorIQ™ P4080 Debug Architecture Overview



- Run control is performed via Aurora or JTAG and other control signals
- Nexus Port Controller (NPC) manages trace buffer, Nexus trace port
  - Trace output over Aurora or to memory
- Event Processing Unit (EPU) monitors events, generates cross-triggers
- CoreNet™ Debug Client, DataPath Debug Client monitor peripherals, memory

QorIQ™ P4080 Software-Visible Debug Features

- ►Many P4080 debug features are provided for resident debug agents
  - User, Supervisor and Hypervisor programming model registers
  - Memory-mapped peripheral devices, on SoC platform or off-chip
  - e500mc debug interrupts and debug registers
  - Core and platform performance and watchpoint monitors, cross triggers
  - Ability to instrument code to generate trace messages

- ►Above features are also visible to non-resident debug tools

e500mc Debug Interrupts

- ►Resident debug software can field architected debug interrupts
  - Trap
  - Instruction Address Compare (IAC)
  - Data Address Compare (DAC)
  - Instruction complete
  - Branch taken
  - Return from interrupt
  - Interrupt taken
  - Unconditional Debug Event (UDE)

- ►These are carried over from earlier e500-based processors

e500mc IAC and DAC Registers

- Each e500mc core provides 2 Instruction Address Compare (IAC) and 2 Data Address Compare (DAC) registers

- Flexible control of debug events
  - Each IAC or DAC supports events on exact address match
  - Paired IAC or DAC registers support events over address ranges
  - Single DAC registers support events over mask-defined ranges ≤ 4KB

- IAC and DAC debug events
  - Can generate in-core debug interrupts
  - Can signal core events to Event Processing Unit (EPU)
  - DACs can generate data trace messages over mask-defined ranges

e500mc Performance Monitor Counters (PMC)

- Each e500mc core provides 4 performance monitor counters

- Count e500mc core events of interest
  - Chainable 32-bit counters with associated control registers
  - 128 wide-ranging event types reserved
    - Instructions, branches, pipeline stalls, load/store, MMU, cache, bus
  - Event occurrence or duration above threshold countable

- PMC overflows
  - Can signal interrupts
  - Can signal events to Event Processing Unit for cross-triggering
  - Can generate Nexus watchpoint trace messages

QorIQ™ P4080 Event Processing Unit (EPU)

- The Event Processing Unit generates event cross-triggers
  - EPU watchpoint monitors generate triggers by matching events
  - Event sources include e500mc cores, SoC blocks, performance counters, external triggers
  - Cross trigger targets include e500mc cores, Soc blocks, Nexus trace output, external triggers

- The EPU also contains the SoC platform PMCs

- e500mc cores can be assigned to “core groups”
  - Defined and managed by Run Control/Power Mgt. Unit (RCPMU)
  - Core groups can be targets of EPU-generated triggers
  - 4 core groups supported with 0-8 cores per group

QorIQ™ P4080 Performance Monitor Counters

- The EPU provides 32 SoC performance monitor counters (PMCs)

- Count SoC platform events of interest
  - 16-bit counters (with control registers) chainable to 32- and 64-bit
  - Large number of wide-ranging event types reserved
    - 64 “reference events” reserved to be countable by all PMCs
    - Other events typically countable only by specific individual PMCs
  - Counting can be triggered on/off by events

- PMC overflows or value matches
  - Can signal interrupts, routed by PIC
  - Can signal events for cross-triggering
  - Can generate Nexus watchpoint trace messages

QorIQ™ P4080 Run Control Features

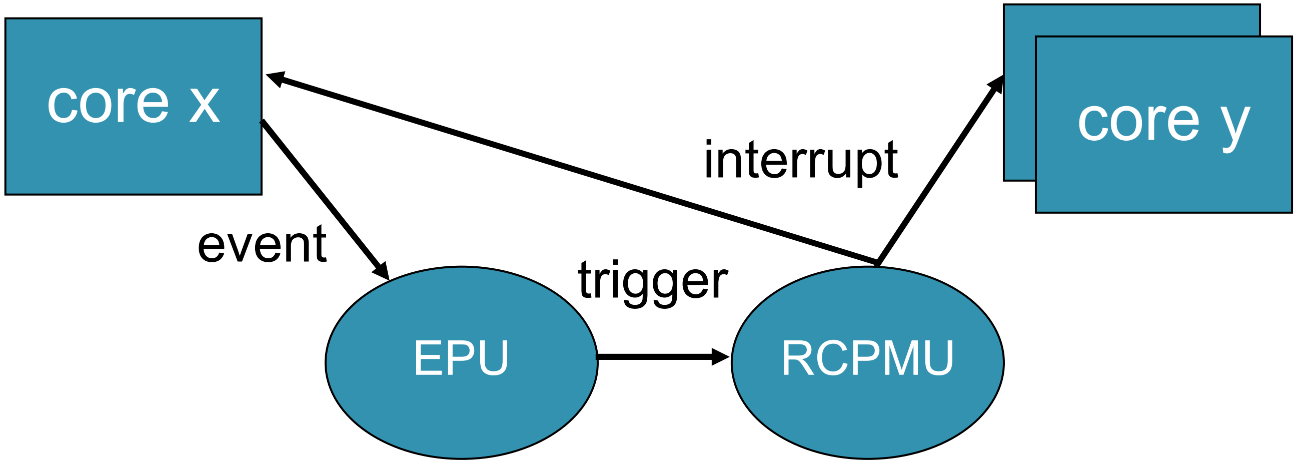
- The P4080 run control port supports a wide range of capabilities
  - JTAG or Aurora-based
  - Supports run control including start/stop of single or multiple cores
  - Read/write of all programming model features in e500mc cores, platform
    - System Access Port (SAP) for read/write of memory mapped I/O devices
  - Read/write of non-programming model features of cores, platform
    - Cache arrays, MMU arrays
- e500mc instruction jamming feature
  - Run control probes can execute “useful” instructions while core is halted
  - Simplifies many operations, including memory read/write

e500mc Debug Notify Halt (dnh)

- ►Debug Notify Halt (dnh) is a new opcode implemented by e500mc
  - Used by run control probe as replacement opcode for breakpoints
  - Places executing core into halted state
    - Allows probe to perform related debug activities
    - Run control probe must resume core execution after dnh executes

- ► dnh execution can signal core event to EPU
  - With platform watchpoints, generates cross-core correlated breakpoints
  - Breakpoint on one core halts execution on any desired core group

e500mc DEVENT



- ►e500mc software can use the DEVENT SPR to signal events
  - Signaled core event selectable
  - Can generate cross-triggers to core groups and/or SoC blocks
- ►Supports software debug agent-based cross-core breakpoints
  - mtspr to DEVENT generates event when executed
  - EPU programmed to cross-trigger debug interrupt to target core group
- ►Flexible capability supports many possible uses
  - Dynamically enable/disable trace, performance counting
  - Software-controlled core resets

QorIQ™ P4080 Nexus Trace Overview

- ►Nexus is an industry standard realtime interface for embedded debug
  - Developed in 1998 from a joint Motorola (Freescale)/HP (Agilent) paper
  - Nexus specification (IEEE-ISTO 5001) available at http://nexus5001.org
    - Simple packet-based messaging protocol and debug application registers
    - Four classes of debug implementations
    - New standard version scheduled for release later in 2008
  - Nexus is driving next-generation debug interfaces
    - High-speed serial debug interfaces through Xilinx's Aurora protocol
    - Cooperation with Power.org™, MIPI, IEEE® 1149.7, Multicore Assoc. standards
- ►The P4080 includes Nexus instruction and data trace capability
  - Supports class 2 trace including optional timestamps
    - Instruction trace, process ID trace, watchpoint trace
  - Supports class 3 data trace including optional timestamps
  - Enabled within cores or by event cross triggers

Power.ORG™ MEMBER

QorIQ™ P4080 Nexus Port Controller (NPC)

- The NPC is the focal point for Nexus messaging in the P4080
  - Selected cores, SoC blocks trace messages collected into 16 KB buffer
    - Trace messages of varying lengths depending on type
  - Selectable ports to deliver Nexus trace messages off-chip
    - 2 dedicated Aurora SerDes lanes (full duplex egress/ingress)
    - Unused SerDes lanes can expand total to 8 Aurora egress (2/4/8)
    - Up to 5 Gbps signaling bandwidth per Aurora lane
    - Supports alternative egress through DDR, PCI Express®

- If enabled, NPC stalls core execution to avoid possible trace loss
  - Stalls if NPC buffer would overrun due to egress bandwidth limits
  - Trace messages already captured within a core will not be lost
  - Nexus always provides sync messages after losses to assist tools

e500mc Data Trace and Data Acquisition Messages

►Data Trace (DTM)

- DTM managed by e500mc Data Address Compare (DAC) registers
- Supports mask-defined address ranges up to 4 KB
- Data writes supported, data reads unsupported

►Data Acquisition Trace (DQM)

- Allows software to generate data trace messages for values of interest
- Performed by storing values to e500mc DDAM SPRs
  - Nexus-related SPRs are privileged, except for data acquisition trace/events

►Data Acquisition Events (DQE)

- Uses DQM infrastructure (same mtspr instructions)
- With EPU watchpoint monitors, can generate event cross-triggers

QorIQ™ P4080 In-Circuit Trace (ICT)

- ►CoreNet™ Debug Client (CNDC)
  - Supports trace filtering of CoreNet-connected interface accesses
    - Data Address/Attribute Compare (4, with each interface visible to 2 max)
    - Data Value Compare (1)
  - CoreNet is software-transparent, but offers a central observation point
- ►Data Path Debug Client (DPDC)
  - Provides trace visibility to Data Path packet flow
    - Trace architected queuing operations for trace points in the QM
    - Trace debug context collected at each processing stage of the FM
- ►Additional ICT visibility supported for DDR, PCI Express® interfaces
- ►All ICT sources generate trace messages to Nexus Port Controller

Wide Range of QorIQ™ P4080 Debug Scenarios

►Debug via run control

- Useful when target system H/W and S/W not stable (board bring-up)
- Useful when non-instrumented but non-real-time debug required
  - No debug software required to run on P4080
  - Cores halt during debug; does affect real-time characteristics of execution
- Provides most complete access to e500mc, SoC resources

►Debug via Nexus trace

- Useful when low-intrusive, real-time debug required
  - No debug software required to run on P4080, cores do not halt
- Host debugger reconstructs, correlates instruction, data path history
- Captured trace history can be "played backward" by host debugger
  - Enables identification of upstream cause of downstream errors

►Debug via software agent

- Useful for OS-aware debug, and S/W debug of one core by another
  - Host-based debugger can provide "system viewer" functionality
- Useful for debug of software running on a simulator
  - Agent should execute identically as on real silicon
- Debug software required to run on P4080, potentially intrusive

On-Chip Debugging of Multicore Systems: Summary

► Software complexity of multicore systems stress system debugging

► QorIQ™ P4080 implements a wide range of on-chip debug features

- Implementation in e500mc processor cores and datapath SoC
- Powerful run control features
- Advanced high-speed Nexus trace capabilities
- Flexible event processing and control with cross triggers

► Tools from Freescale’s world-class ecosystem will give you the visibility and control to debug tough multicore problems

Related Session Resources

Session Location – Online Literature Library

http://www.freescale.com/webapp/sps/site/homepage.jsp?nodeId=052577903644CB

Sessions

Session ID	*Title*
PE102	Debugging Multicore Power Architecture® Platforms with CodeWarrior™ Tools
PN103	Secure Multicore Solutions (Crypto Acceleration, Deep Packet Inspection, Platform Trust)
PN107	Multicore Solutions and Applications
PN110	Multicore Boot Process
PN111	Performance Analysis with Hybrid Simulation

Demos

Pedestal ID	*Demo Title*

freescale™
semiconductor