

# **The ARM9E Synthesizable Processor Family**

**Simon Segars, CPU Development Manager**

# Outline

- ARM9E Design Motivation & Goals
- Technical Review
  - ARM9E Core / ARM966E / ARM946E
- Cache Architecture & Write Buffer
- Ease of Synthesis & Integration
  - Improved AMBA Bus I/F
- Enhanced Development
  - Real Time Trace / I Trace/ D Trace / Non-Stop Debug
- Conclusion

# Design Motivation & Goals (1)

- Bridge DSP Chasm
  - Integrate DSP extensions into a single engine controller
- Develop Flexible Memory Systems
  - One memory size and system does not suit all applications
- Improve SoC Support Tools
  - Real Time Trace and Non-Stop Debug

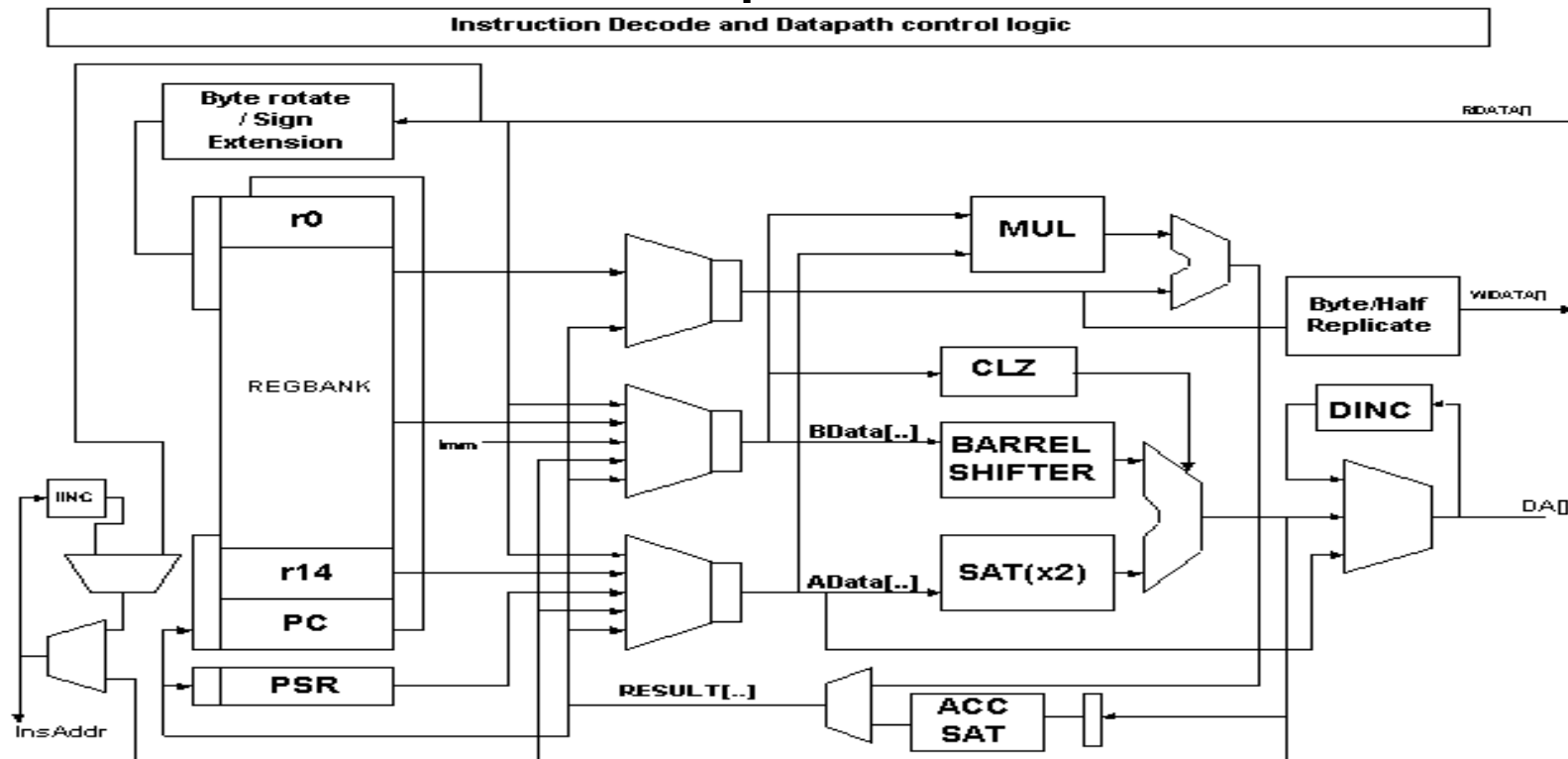
## Design Motivation & Goals (2)

- Ease SoC Core Integration
  - Synthesis friendly to ease integration of cores into SoC designs flows
  - Enable use of standard ASIC library components
  - Improve time-to-market
- Continue Industry Leading Power Efficiency

# Technical Review

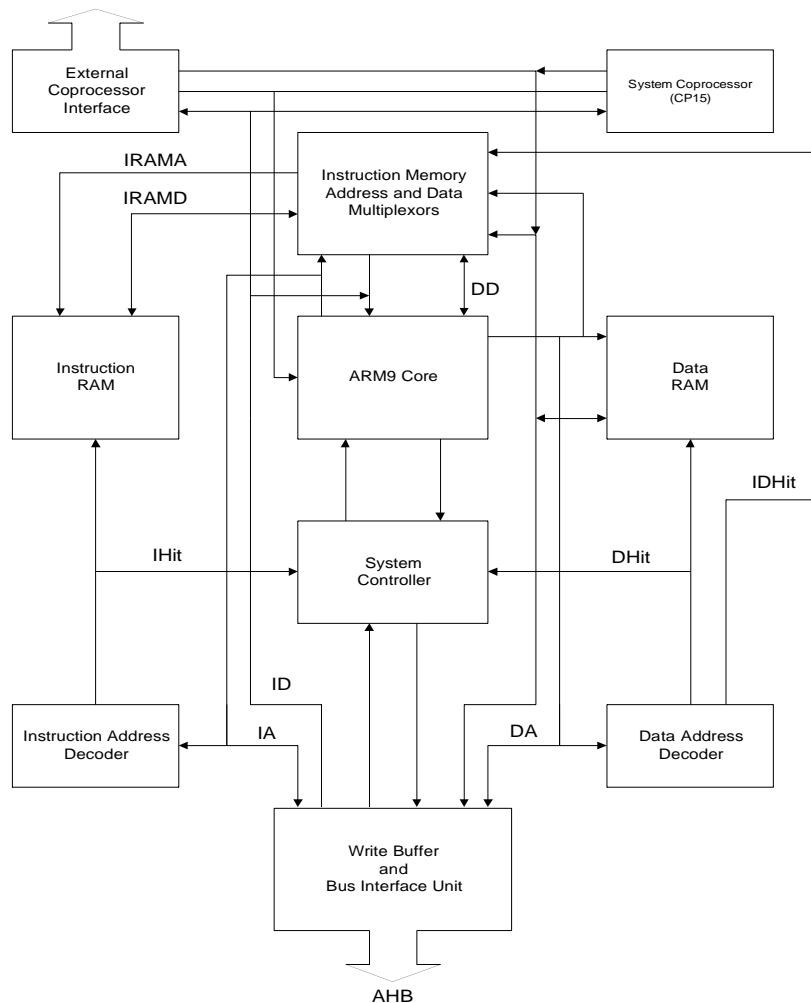
## ARM9E Processor Core

- An ARM9TDMI core with DSP Extensions
- ARM9E Core Datapath



# ARM9E DSP Extensions

- New 32x16 and 16x16 multiply instructions
  - SMLA<sub>xy</sub>, SMLAW<sub>y</sub>, SMLAL<sub>xy</sub>, SMUL<sub>xy</sub>, SMULW<sub>y</sub>
  - Allow independent access to 16-bit halves of registers
  - Give efficient use of 32-bit bandwidth for packed 16-bit operands
  - 32x32 multiply already in ARM ISA
- Zero overhead fractional saturating arithmetic
  - QADD, QSUB, QDADD, QDSUB
- Count leading zeros instruction
  - CLZ for faster normalization and division
- Single cycle 32x16 multiplier array
  - Speeds up all ARM9E multiply instructions

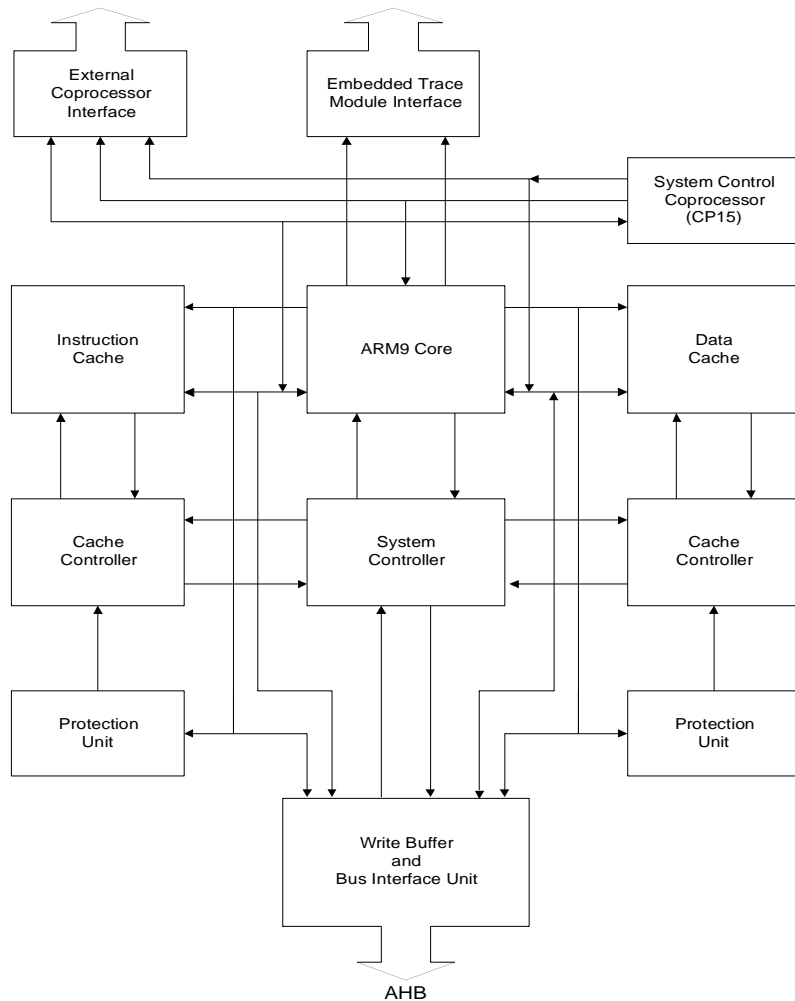


- Tightly coupled instruction and data RAM - variable size up to 64M. RAM fixed in memory map to ease implementation and reduce power.
- Data interface needs access to instruction RAM for constants embedded within code.
- Write buffer to minimize system loading. Buffer controlled by system coprocessor and address decoders.

# ARM966E

## Why no cache?

- Not all applications warrant the complexities of a cache
  - Still need the performance benefits of memory closely associated to processor core
- Processor core with local memory addresses
  - Solves complexities of feeding both interfaces of Harvard processor core.

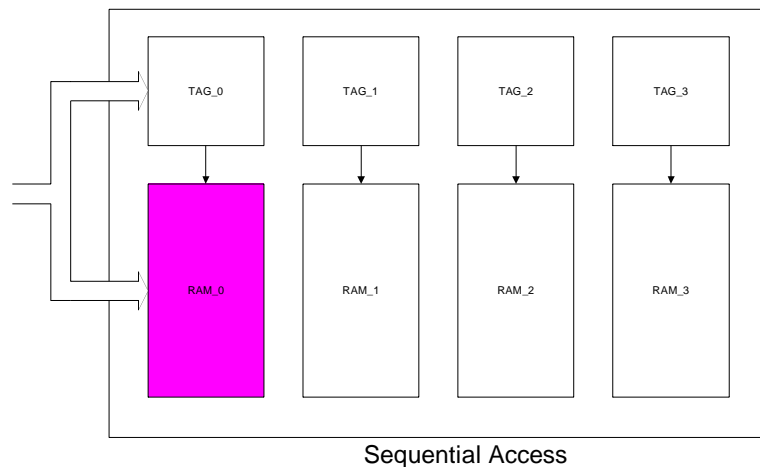
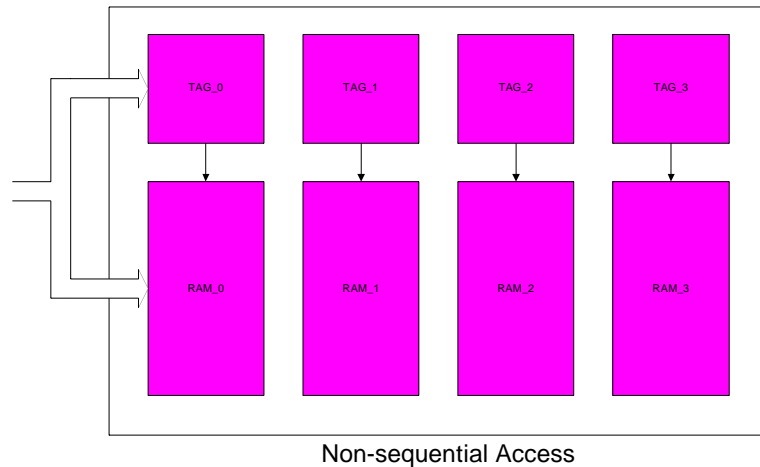


- 4 Way set associative cache - size is variable
- Protection units allow memory partitioning and attribute controls (cacheable, access permissions) for each region.
- Instruction and data address space can have 8 regions of variable size.
- Coprocessor interface for additional functionality closely coupled with processor core.
- Write buffer to minimize system loading.

# Cache Architectures

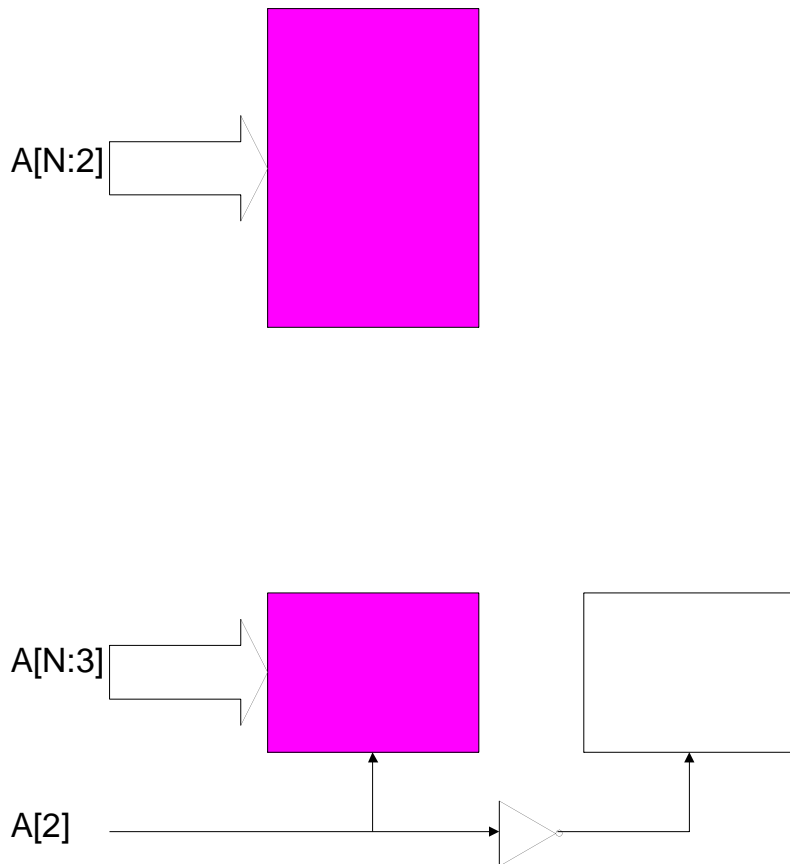
- Previous cache architecture
  - Used 64 way set associative cache
  - Relied on full custom design techniques
- Synthesizable cache architecture
  - 4 way set associative cache (good compromise between performance and complexity)
  - Makes use of ASIC library components
- Cache treated as Synchronous RAM
  - Simple memory interface allows connection to ASIC library RAM cells
  - Minimizes rework as cache size changes

# Minimizing Power Within the Cache (1)



- Non-sequential accesses require all TAG and RAM blocks to be accessed. This avoids a stall cycle while the TAG is accessed followed by the RAM on the next clock cycle.
- Sequential accesses do not need to access the TAG arrays. Only one RAM block is active.
  - This has greater affect for instruction accesses than data accesses

# Minimizing Power within the Cache (2)

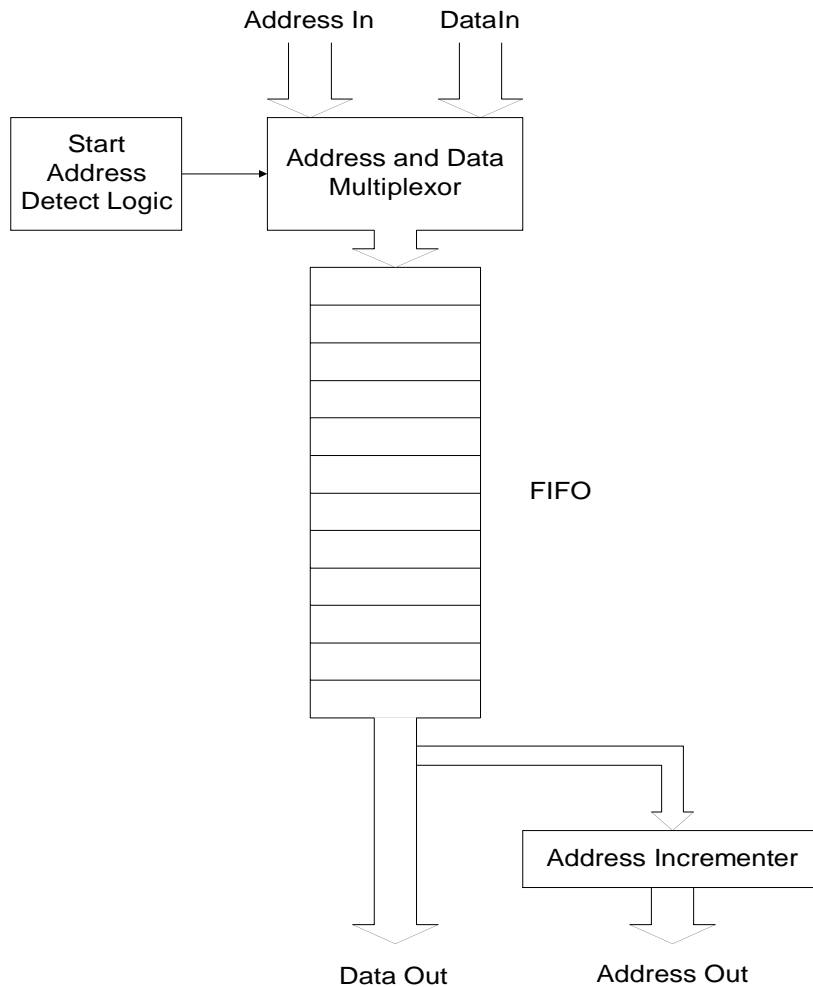


- Large memory arrays burn large amounts of power.
- Splitting memory and using simple address decodes reduces power.
  - A memory half the size uses more than half the power of a full size memory, but is accessed only half as often.
- Splitting memory allows more efficient cache evictions.
- Helps with Data cache power efficiency.

# Write Buffer

- De-couples processor core from system memory bus
  - Improves processor performance
- Previous designs used separate full custom address and data FIFOs
- Synthesizable processors use an adaptive buffer
  - Entries can be either address or data to maximize use of available storage

# Adaptive Write Buffer (1)



- Each entry can be address or data
- Only start address is stored
  - Address incrementer generates sequential addresses.

# Adaptive Write Buffer (2)

A0
A1
A2
A3

D0
D1
D2
D3

A0

D0
D1
D2
D3
D4
D5
D6
D7

A0
D0
A1
D1
A2
D2
A3
D3

A0
D0
D1
D2
D3
D4
D5
D6
D7

- Adaptive buffer makes better utilization of available storage
  - Separate address FIFO quickly fills with single writes.
  - Separate data FIFO fills with long sequential writes.
  - Adaptive buffer still has space in each case.

# Ease of Synthesis

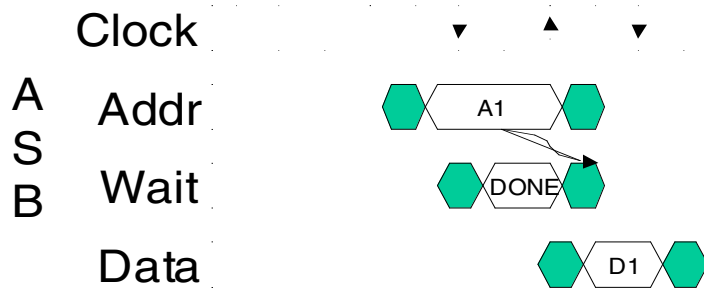
- Single rising edge clock design
- Prime deliverable is RTL code
  - Allows silicon vendors to exploit individual strengths
  - Ensures ARM compliance
    - Formal verification
    - Compliance test suite

# Ease of Integration

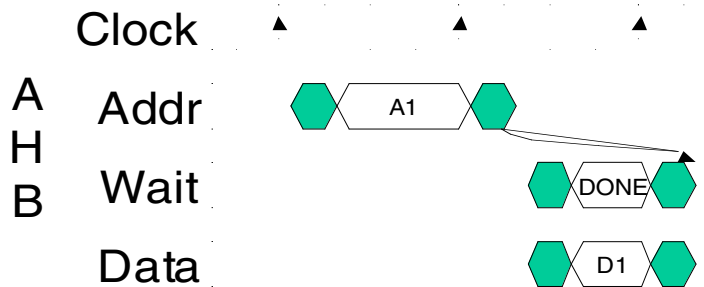
- AMBA High-Speed Bus (AHB) Interface
  - Fully pipelined bus allowing higher operating frequencies.
  - Can operate at integer multiples of processor core clock period.
  - Supports multi-master operation.

# AHB versus ASB

## AHB Transfer Compared to ASB



- ¥ Clock inverted
- ¥ Full cycle pipeline
- ¥ Slaves have twice as long to generate wait

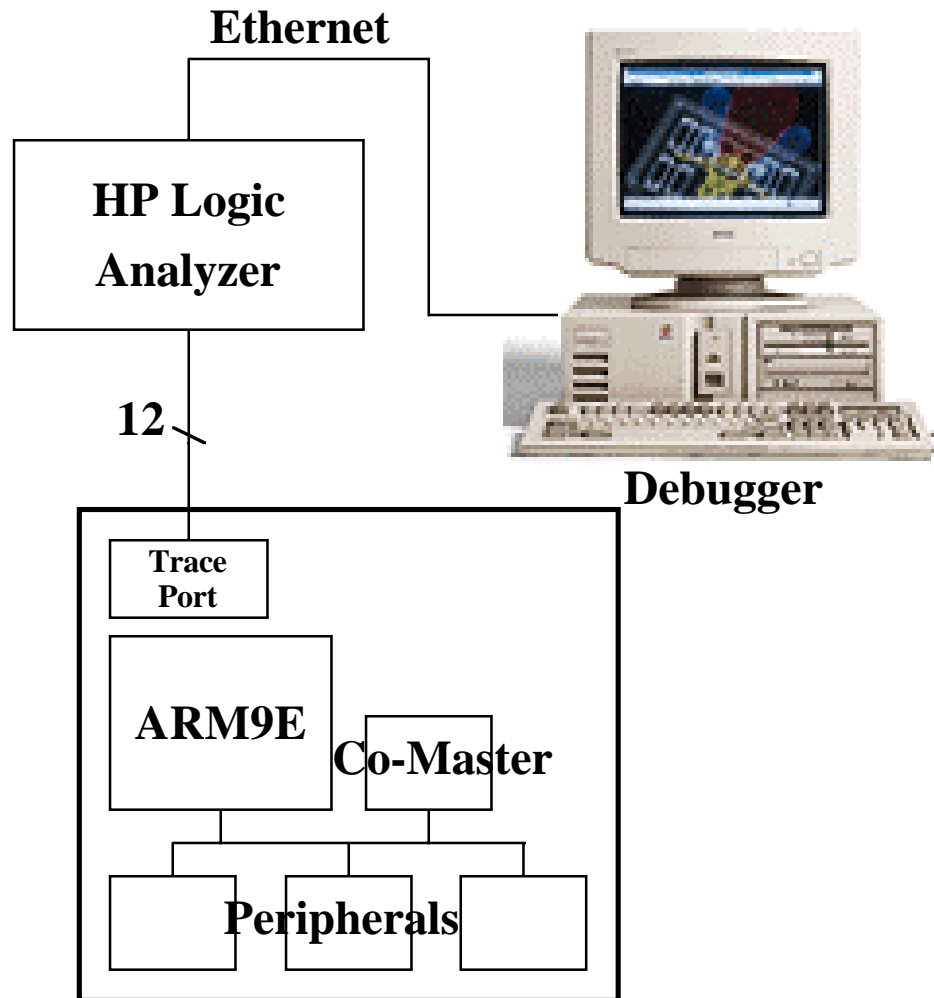


# Enhanced Development

## Real Time Trace & Non Stop Debug

- Enhancements for SoC Debug
  - EmbeddedICE module allows intrusive debug
  - Additions for Real Time Debug allows system operation to continue while interrogating system state.
  - Embedded Trace Module allows real time monitoring of processor execution

# Trace Components



- **On-chip trace port module**
  - Compresses real-time trace information for instructions and data.
- **Logic Analyzer**
  - Collects trace information in deep trace memory.
- **Debugger**
  - Extracts and decompresses trace information.
  - Displays trace information linked back to source code.

# Instruction (PC) Trace

- Only instruction address is required.
- To reduce bandwidth only branch address with pipeline status is required.
  - This provides the entry and exit point for every sequence of code
- To reduce bandwidth further only indirect branches need to be broadcast.
  - Destination of direct branches can be inferred from the code, e.g

<u>Address</u>	<u>Code</u>	<u>Branch Type</u>
0x1C	ADD R1, R2, R3	None
0x20	MOV R3, R4	None
0x24	BL 0x120	Direct

*Destination of branch can be calculated from code*

0x144	ADD R1, R2, R3	None
0x148	MOV R3, R4	None
0x14C	MOV PC, R14	Indirect to 0x28

*Destination of branch not known until execution*

0x028	MOV R3, R5	None
-------	------------	------

# Data Trace

- Data accesses (loads/stores) can also be recorded in the trace stream.
- An encoding in the execution status indicates a data access has been sent to the trace port.
- Not all data accesses are required, trace is limited to certain address ranges.

# Non-stop debug

- Core Logic
  - Allows debugging of a system without completely stopping the processor core.
  - Enables a debugger to stop and debug one task while background interrupt routines continue to run.
    - EmbeddedICE hardware generates an exception which allows a monitor program to execute while allowing higher priority exceptions to be serviced.
- Debug Monitor Program
  - Communicates with the debug host via the debug communications channel

# Conclusion (1)

## Performance Review

- Performance (limited by integer core and synthesis library)
  - > 200 MHz (0.18 micron process)
  - 160 MHz (0.25 micron process)
- ARM966ES Gate Count
  - 90K - 100K gates excluding RAM
- ARM946ES Gate Count
  - Approx. 150K gates excluding RAM

# Conclusion (2)

## Power Control Review

- Power management with sleep feature
- Management of memory to minimize power
  - Minimize RAM on time
  - Splitting RAM into banks minimizes size of array activated at any time
- Write buffer allows system bus to run at lower frequency without penalizing core performance

# Conclusion (3)

## Improving Test Coverage

- Scan insertion for processor
- Built In Self Test (BIST) for memory test
  - Flexible test architecture can be tailored to match memory architecture e.g. programmable seed values, choice of algorithm.
  - Simple programmers interface
  - Can also be activated using scan chains if desired
- ARM946E / 966E Availability: 1Q 2000