Cortex A8 Processor

Richard Grisenthwaite

ARM Ltd

Evolution of the ARM Architecture

- Original ARM architecture:
  - 32 bit RISC architecture
  - 16 Registers
    - 1 being the Program counter
  - Conditional execution on all instructions
  - Load/Store Multiple operations
    - Good for Code Density
  - Shifts available on data processing and address generation
  - Original architecture had 26 bit address space
    - Augmented by a 32 bit address space early in the evolution

- Thumb instruction set was the next big step
  - ARMv4T architecture (ARM7TDMI)
  - Introduced a 16 bit instruction set alongside the 32 bit instruction set
    - Switching ISA as part of branch or exception
    - Not a full instruction set – ARM still essential

Evolution of the Architecture (2)

- ARMv5TEJ (ARM926EJ-S) introduced:
  - Better interworking between ARM and Thumb
  - DSP focussed additional instructions
  - Jazelle-DBX for Java byte code interpretation in hardware
- ARMv6 (ARM1136JF-S) introduced :
  - Media processing – SIMD within the integer datapath
  - Enhanced exception handling
  - Overhaul of the memory system architecture
- ARMv7 rolls in a number of substantive changes:
  - Thumb-2*
  - TrustZone*
  - Jazelle-RCT
  - Neon
  - ARMv7 is split into 3 profiles

* - Introduced initially as extensions to ARMv6

Thumb-2

- Combined 32 and 16 bit instruction set
  - Instructions can be freely mixed
  - 16 bit instructions include the original Thumb instruction set
    - Complete compatibility with Thumb binaries
    - Some new 16 bit instructions for key code size wins
- Virtually all instructions available in ARM ISA available in Thumb-2
  - Some minor cleaning up of system management instructions
  - In principle can stand-alone as a complete ISA
- Unified assembly language for ARM and Thumb-2
  - Assembly can be targeted to either ISA
- Conditional execution made available via IT instruction:

ARM = 20 bytes

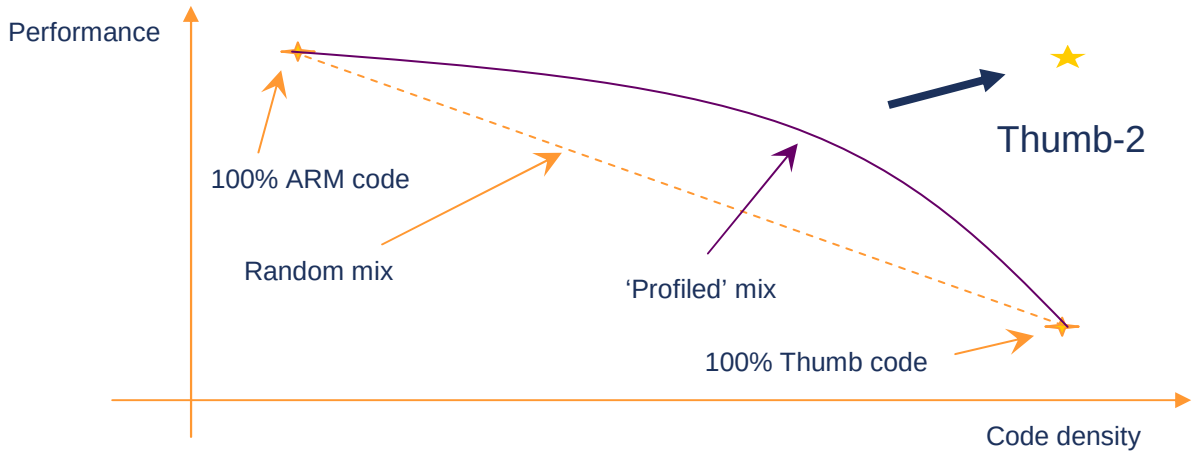
```
CMP     r3,#1
EOREQ   r1,r1,#0x4000
EOREQ   r1,r1,#2
MOVNE   r3,#0
MOVEQ   r3,#1
```

Thumb-2 = 14 bytes

```
CMP r3, #1              ;2 bytes
ITTET                   ;2 bytes
MOVWEQ r3, #0x4002      ; 4 bytes
EOREQ r1, r3            ; 2 bytes
MOVNE r3, #0            ; 2 bytes
MOVEQ r3, #1            ; 2 bytes
```

Thumb-2 (2)

- "Thumb Code density at ARM Performance"
  - In principle this could be achieved with ARM and Thumb previously
    - Much of the code running is not performance critical
    - With code knowledge, can compile non-critical code to Thumb
  - Much simpler with Thumb-2



Expect to see growing emphasis on Thumb-2 in the future

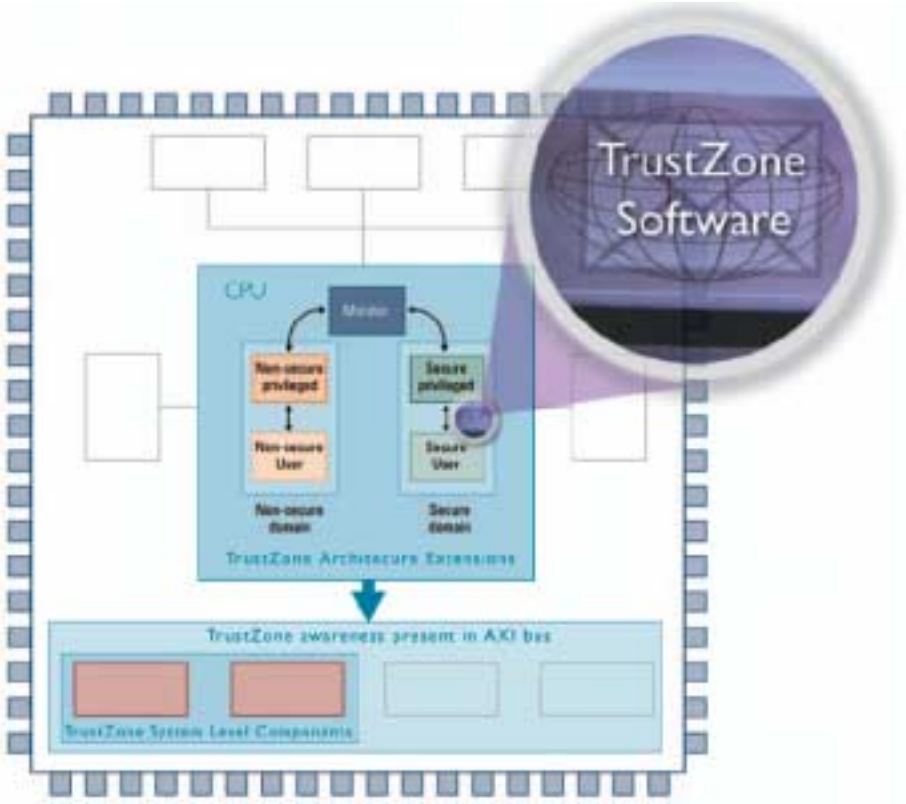
ARM still totally committed to ARM ISA compatibility

The ARM instruction set is still completely supported

No plans to "downgrade" the ARM ISA in the applications space

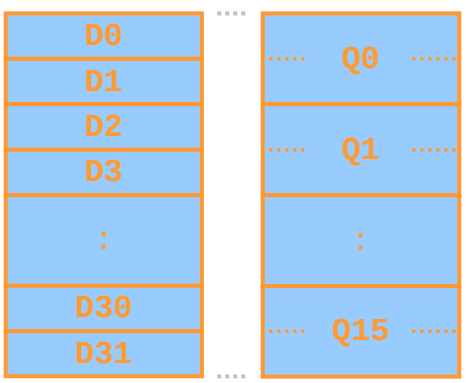
TrustZone

- Architectural extensions to introduce a “Security” state
  - Orthogonal to User/Privileged split
- Effectively two virtual CPUs separated by a new mode
  - Monitor mode the gatekeeper for switching CPUs
  - Some hardware registers duplicated to aid switching
- Memory tagged as secure and non-secure by the system
  - Only the secure CPU can access the secure memory & peripherals
  - System can include secure and non-secure peripherals

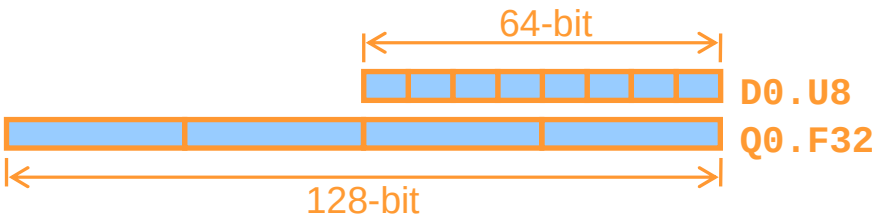


ARM NEON™ Technology Overview

- 64/128-bit Hybrid SIMD architecture
- Independent Register file with 2 aliased views:
  - 32 x 64-bit registers (D0-D31)
  - 16 x 128-bit registers (Q0-Q15)



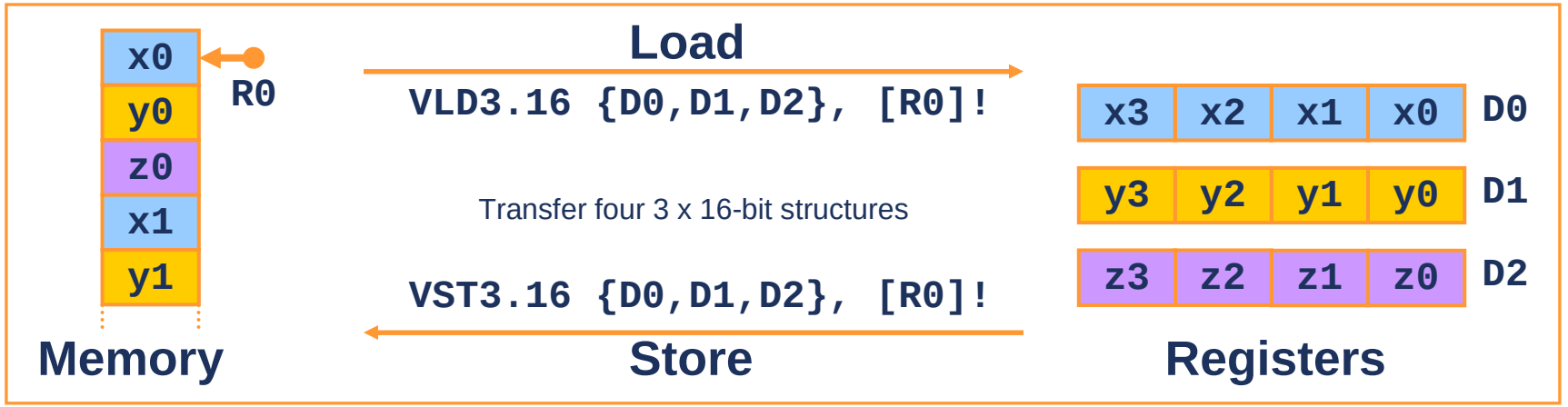
- Integer and SP Floating-point processing
  - 8, 16, 32, 64-bit Integers
  - Single-precision Floating-point



- Encoded in ARM and Thumb-2
- 2 to 4x performance improvement over ARMv6 SIMD
- Accelerates audio, video, and 3D-graphics

NEON SIMD Structure Load/Store

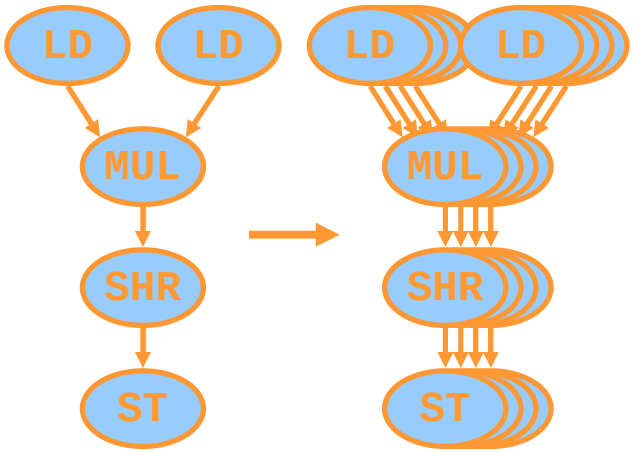
- Native support for structures
  - e.g. Complex Numbers, Pixels, Co-ordinates
  - Memory treated as an Array of Structures (AoS)



- Eliminates ‘shuffling’ overhead
  - Optimised memory access as single transfer
  - Data arranged for efficient SIMD processing

NEON Vectorising Compiler Target

- **NEON** provides a consistent algorithm mapping
  - Apply narrowing analysis
  - Vectorize over loop iterations



- Enabled by architectural model
  - Orthogonal instruction framework
  - Few inter-lane operations
  - Fused Data Type conversion

- **NEON** designed in conjunction with compiler technology
  - Ensure architecture optimised for this compiled mode
  - Benefits of CSE, unrolling, scheduling, register allocation
  - Portable solutions by avoiding hand coding or intrinsics

jazelle®-RCT: Runtime Compilation Target

- Beneficial to Java and a wide range of emerging languages
  - Microsoft .NET MSIL, Perl, Python etc
- Enables high performance in smallest memory footprint
  - Optimal balance between speed and code density with run-time compilers
- Low cost and low power
  - Less than 8K gates and small memory footprint result in lower power
- Complementary to jazelle® DBX on mid-tier devices
  - for optimum Java performance and efficiency
- Broad industry adoption
  - Sun Microsystems, Aplix and Esmertec are early adopters
- Builds on success of jazelle® DBX technology

Cortex-A8 Processor Highlights

- First implementation of the ARMv7 instruction-set architecture, including the Advanced SIMD media instructions (NEON™)

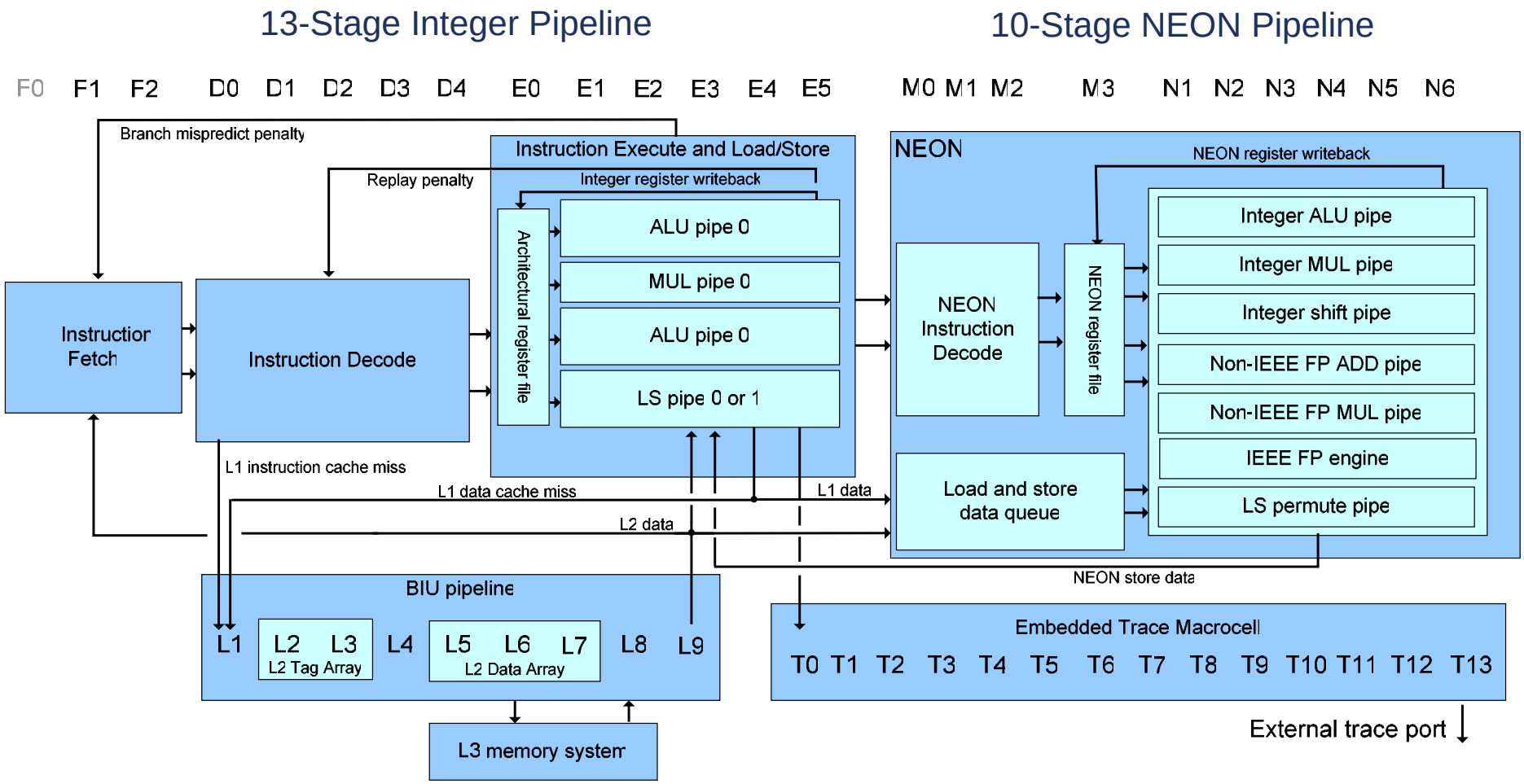
- In-order, dual-issue, superscalar microprocessor core
  - 13-stage main integer pipeline
  - 10-stage NEON media pipeline
  - dedicated L2 cache with 9-cycle latency
  - branch prediction based on global history

- Key metrics
  - delivers 2000 DMIPS for next-generation consumer applications
  - average IPC of 0.9 across multiple benchmark suites
    - includes EEMBC, SpecInt95, Mediabench, and partner-provided applications
  - achieves 1GHz when fabricated in high-performance technologies
  - consumes less than 300mW in low-power devices
  - less than $4mm^2$ at 65nm, excluding NEON, L2 cache, and Embedded Trace

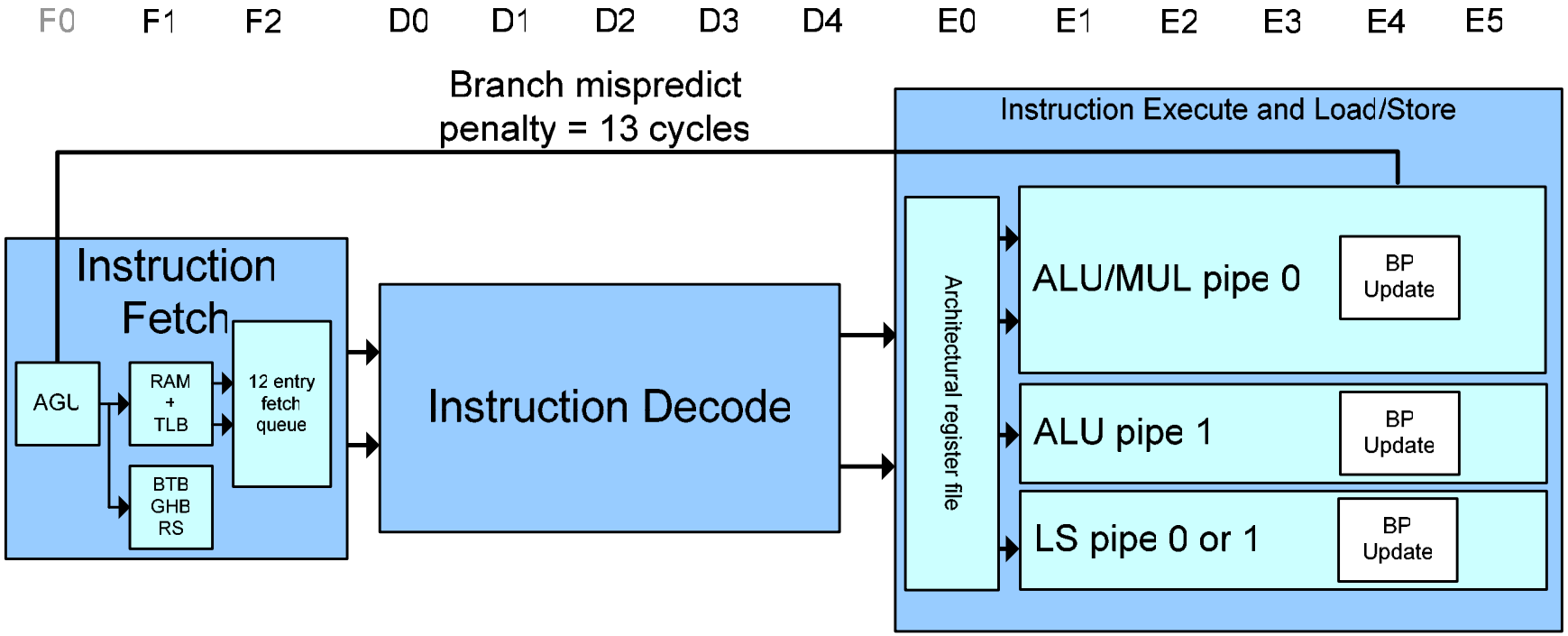
ARM Cortex-A8: why Superscalar?

- In-order instruction issue
  - less complex than out-of-order
    - fewer structures means lower power
    - less need for custom design
  - can maintain high IPC with
    - fully symmetric ALU pipelines
    - all critical forwarding paths supported
    - dual-issue of dependent instruction pairs
- Static scheduling with instruction replay on memory stall
  - low-power consumption due to early availability of gate enables
  - fire-and-forget instruction issue removes critical paths from the design
- Net result
  - *high-frequency design with out-of-order performance, but in-order clock frequency and power consumption*
  - *Average **IPC** of **0.9** across 150+ ARM and industry benchmarks*

Full Cortex-A8 Pipeline Diagram



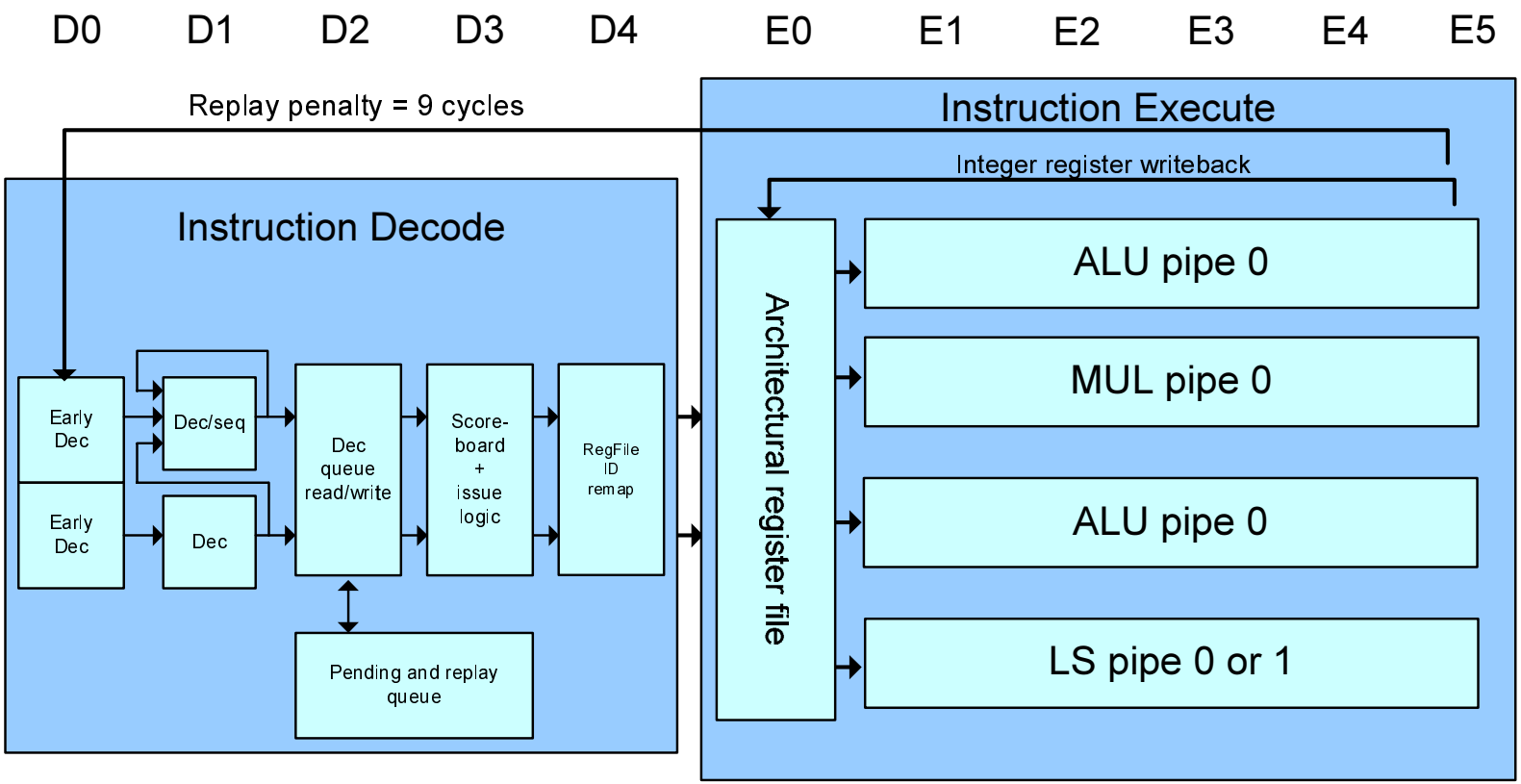
Control Flow



- Dynamic branch predictor components
  - 512-entry 2-way BTB
  - 4K-entry GHB indexed by branch history and PC
  - 8-entry return stack
- Branch resolution
  - all branches are resolved in single stage
  - Maintains speculative and non-speculative versions of branch history and return stack

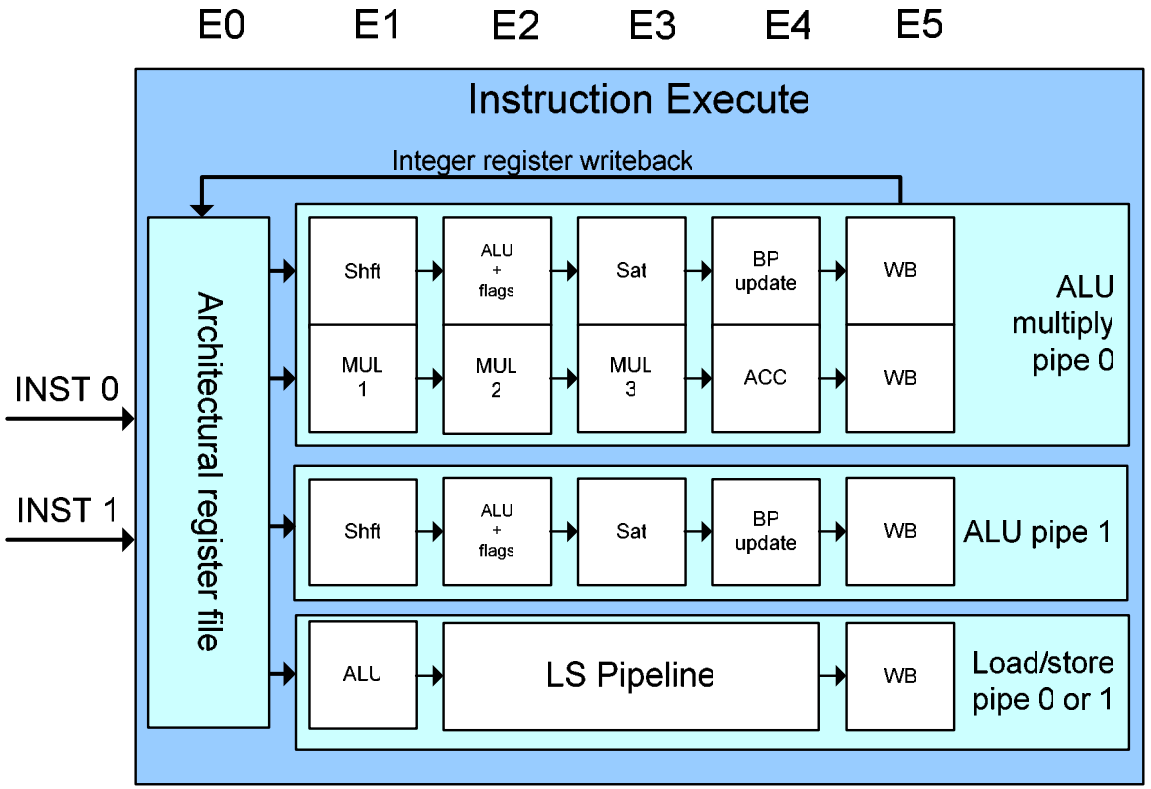
Branch prediction maintains 95% accuracy over a wide codebase

Instruction Decode



- Instruction decode highlights
  - pending queue reduces Fetch stalls and increases pairing opportunities
  - replay queue keeps instructions for reissue on memory system stall
  - scoreboard predicts register availability using static scheduling techniques
  - cross-checks in D3 allow issue of dependent instruction pairs

Instruction Execution

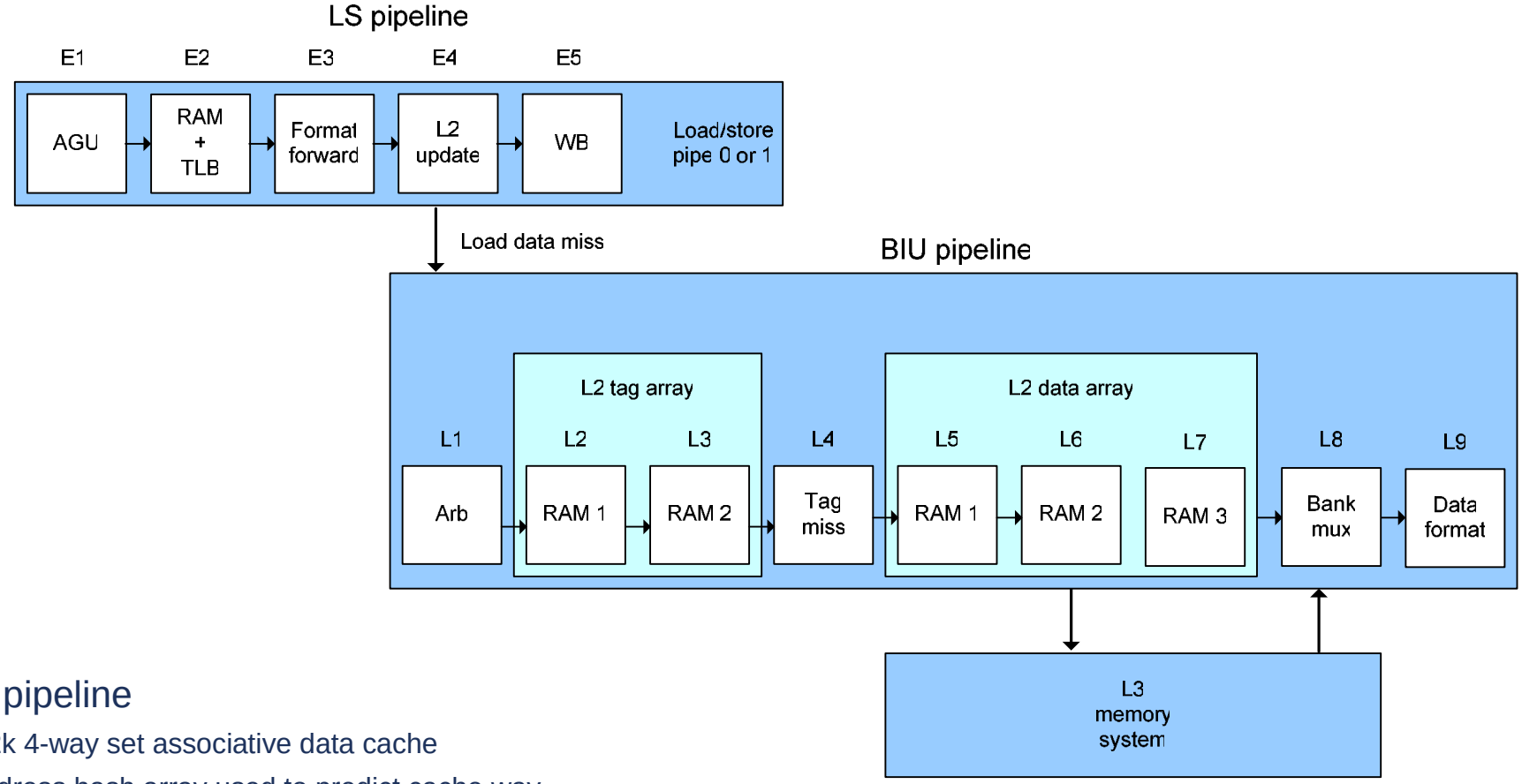


- Execution pipeline highlights
  - 2 symmetric ALU pipelines: Shift/ALU/SAT
  - Load/store pipe used by instructions in either pipeline
  - Multiply instructions are tied to pipe 0
  - All key forwarding paths supported
  - Static scheduling allows for extensive clock gating

Memory System on Cortex-A8

- Harvard Level 1 Caches – both 32KByte 4 way set associative
  - VIPT Instruction cache; VIPT Data cache with alias detection
  - Level 1 Data cache is blocking
    - Non-Neon read misses cache cause replay of subsequent instructions
    - Reduces complexity in later pipeline stages
      - Good for power and clock frequency
    - Neon data not allocated to L1 (but will read/update in L1 if necessary)
- Unified Level 2 Cache
  - PIPT, 8 way set associative
  - Fully pipelined and non-blocking
    - Up to 9 memory transactions in flight
  - Streams to the Neon processing unit; up to 16GByte/s bandwidth
- 64 or 128 bit AMBA AXI interconnect to memory
  - Split transaction burst based protocol
  - Supports multiple outstanding memory transactions to minimise memory latencies

Memory System



LS pipeline

32k 4-way set associative data cache

Address hash array used to predict cache way

Saves power and improves timing

load data forwarding in E3 to all critical sources

one-cycle load-use penalty for ALU

store data not required until E3

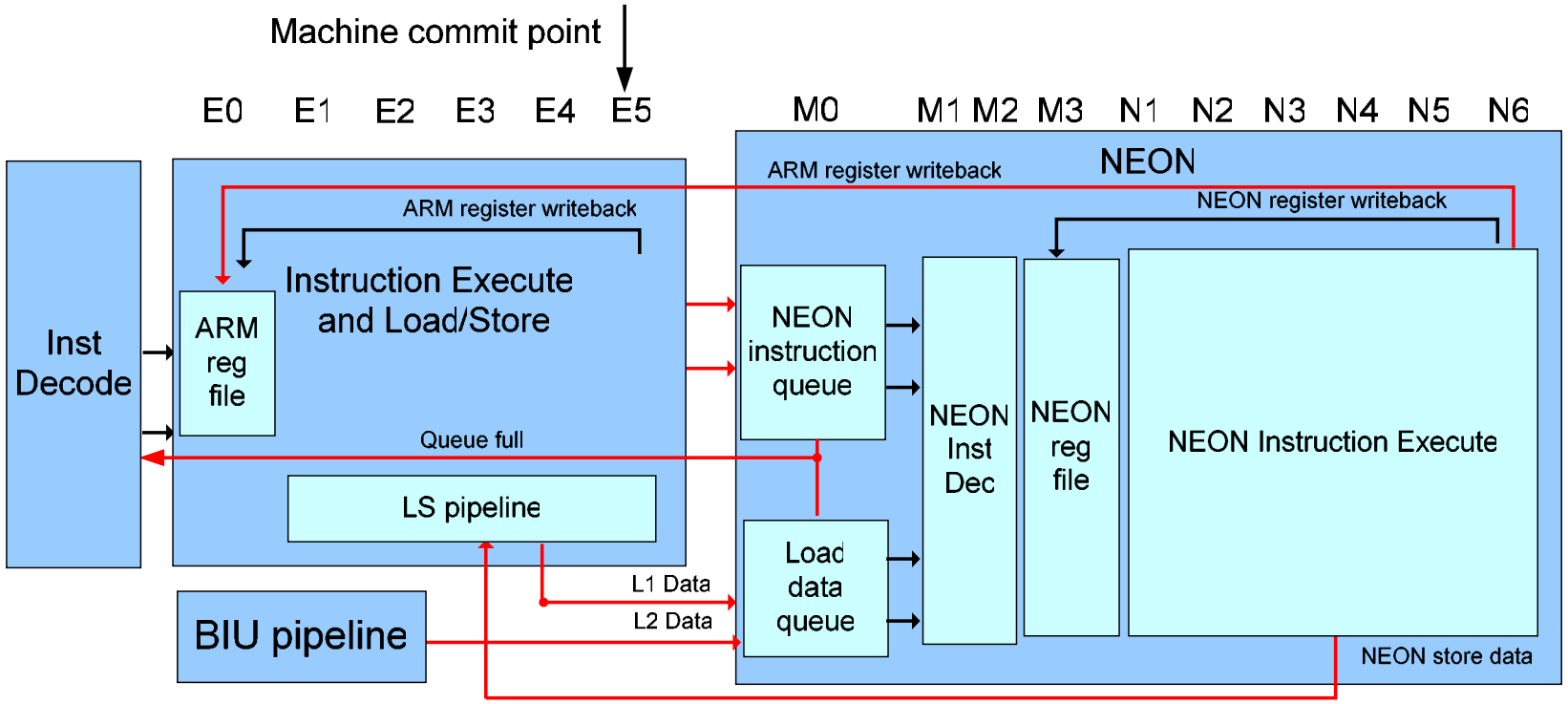
BIU pipeline

9-cycle minimum access latency to L2 cache

L2 built using standard compiled RAMS (64k-2MB configurable size)

64/128bit AXI L3 bus interface supports up to 9 outstanding transactions

NEON Interfaces



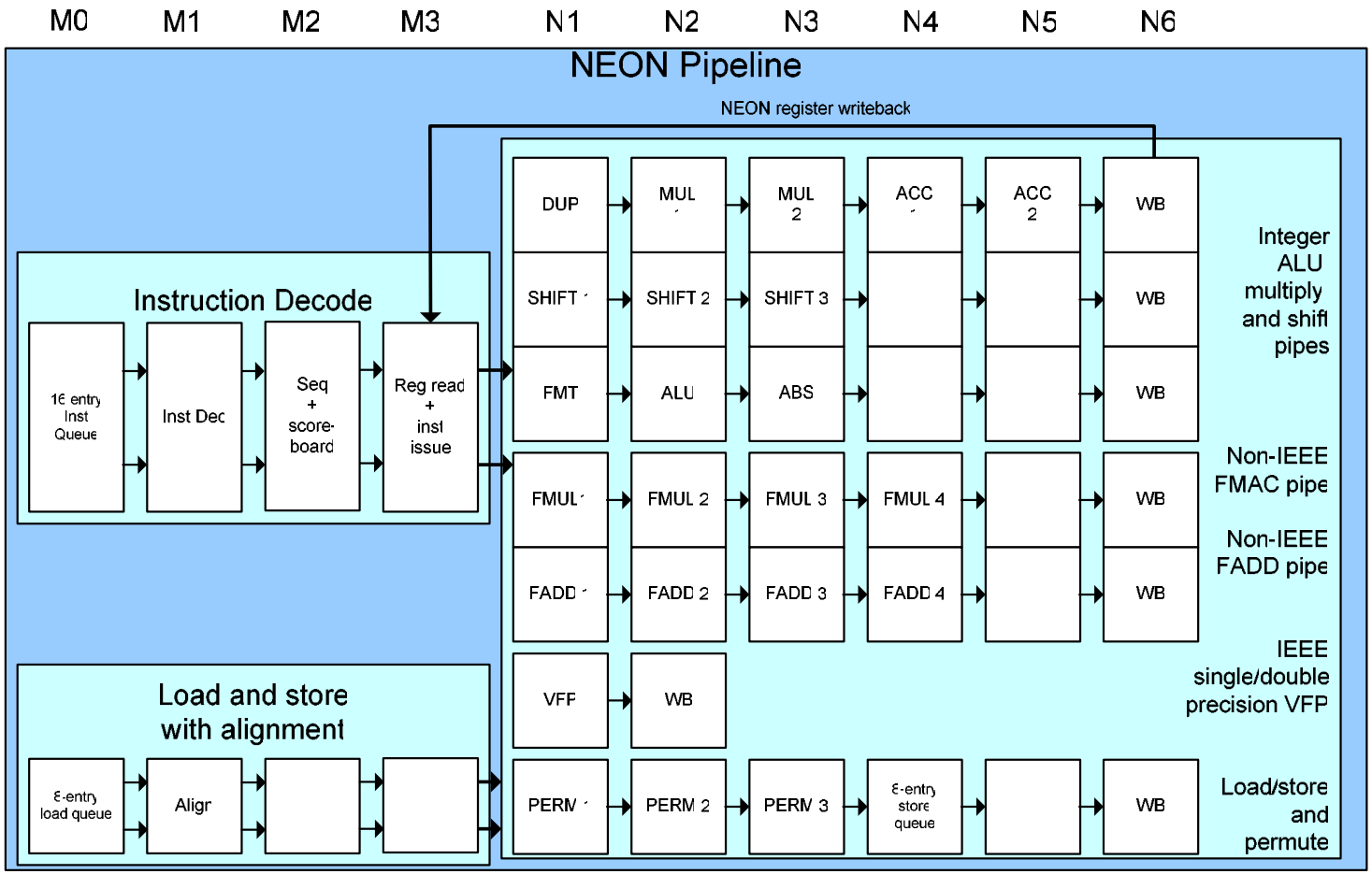
Skewed late in pipeline, past the retire point

- reduces interface complexity
  - exception handling not required
  - decoupling queues from integer machine
- removes load-use penalty
- negative impact on NEON -> ARM transfers
  - nonblocking ARM register file helps hide latency

Streaming to and from L2 memory system

- up to 8 outstanding transactions
- can receive 128 bits/cycle
- can receive data from L1 or L2 memory system
- independent NEON store buffer

NEON Media Engine Unit



Instruction issue

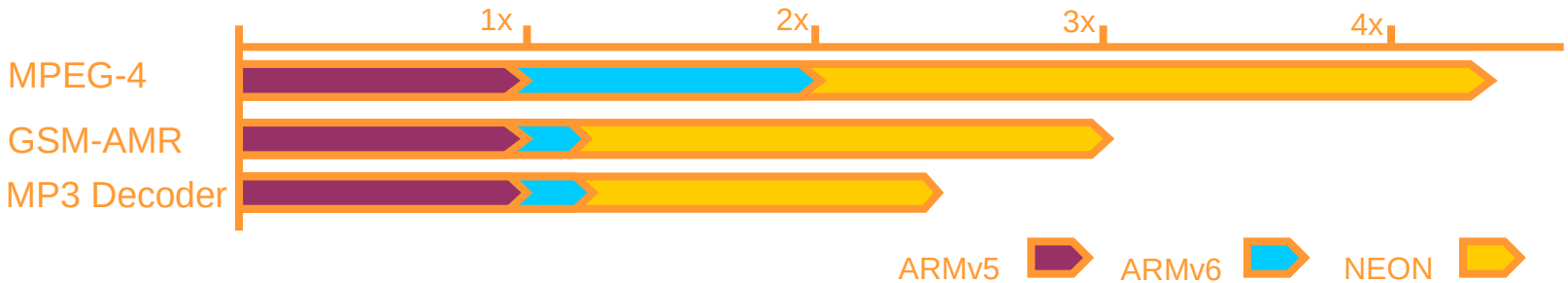
- static scheduling with fire-and-forget issue
- 1 LS + 1 NINT/NFP can issue each cycle

Execution pipelines

- all pipelines are 64-bit SIMD
- floating-point MAC executed using both FADD and FMUL pipelines

Cortex-A8 NEON Technology

- Accelerating standardization of media processing for next generation mobile and consumer products
- The ideal software target to run rapidly evolving downloadable media players such as Windows Media Player 10 and Real Player



Video, 30fps VGA decode	
MPEG-4 including de-ring and de-block filters, yuv2rgb[1]	275MHz
H.264 (estimated)[4]	350MHz
GSM-AMR, worst case[2]	13MHz
MP3 decode, 320kbps 48kHz, worst case[3]	9.4MHz

1) MPEG-4 Simple Profile @ 30fps 512kbps , 133MHz SDRAM 10-1-1-1-1-1-1-1 memory, includes deblocking and deringing filters

2) MP3 Decoder @ 320kbps 48kHz (worst case), 133MHz SDRAM 10-1-1-1-1-1-1-1 memory

3) GSM-AMR (worst case), 3 cycle per word memory

4) H.264 Decoder Baseline profile

Coresight Debug and Trace

- Hardware Debug and Trace are key components
  - Valued by the people who use the systems!
- ARMs Coresight moves to a system-centric debug philosophy
  - SoC are not just the core any more
  - Multiple sources of trace data – cores, buses, software instrumentation
  - Multiple debug components – cores, buses watchers etc
  - Cross-triggering of debug events to multiple cores
  - System identification of components in the SoC essential to debug
    - Topology identification methodology as well
- Coresight is a debug and trace focussed system architecture
  - Debug components part of a debug memory space
    - Standardised interface to JTAG or Serial-Wire Debug
  - Open standards to encourage 3rd party adoption
- Cortex-A8 incorporates Coresight compliant interfaces

Implementation Strategy: Motivation

- Why use a semicustom design flow?
  - required to achieve project frequency, area, and power targets
- Why not deliver a hard macrocell?
  - too many restrictions on circuit and layout optimizations possible
  - design porting does not scale well with increases in design size and complexity
- The goal:
  - provide our partners with an alternative method of IP delivery that
    - achieves Cortex-A8 power, area, and frequency targets
    - minimizes the additional effort required from the silicon partner

ARM Cortex-A8 Processor Summary

- Industry-leading performance and power efficiency
  - Greater than 2000 DMIPS for demanding tethered applications
  - Less than 300mW for low power mobile applications
- More than 7 major new technology innovations:
  - NEON, Jazelle-RCT, Thumb-2, TrustZone, AMBA AXI, CoreSight, IEM
- Supported end-to-end by ARM Technology
  - RealView ARCHITECT ESL Models – Artisan AdvantageCE Libraries
- Industry momentum fueling wide adoption
  - 5 licensees, 1/3 of the Top 15 WW Semiconductor Vendors *

Cortex
Intelligent Processors by ARM

Artisan®
ARM® Physical IP

RealView
ARM® Tools by ARM

* Source: Gartner Dataquest (March 2005)

Questions?