ARMv7 Processor Architecture
(Microprocessor Summit)

David Brash

MPS-900
New Product Announcements
San Francisco: 07-March-2005

The Architecture Challenge

ARM widely adopted:

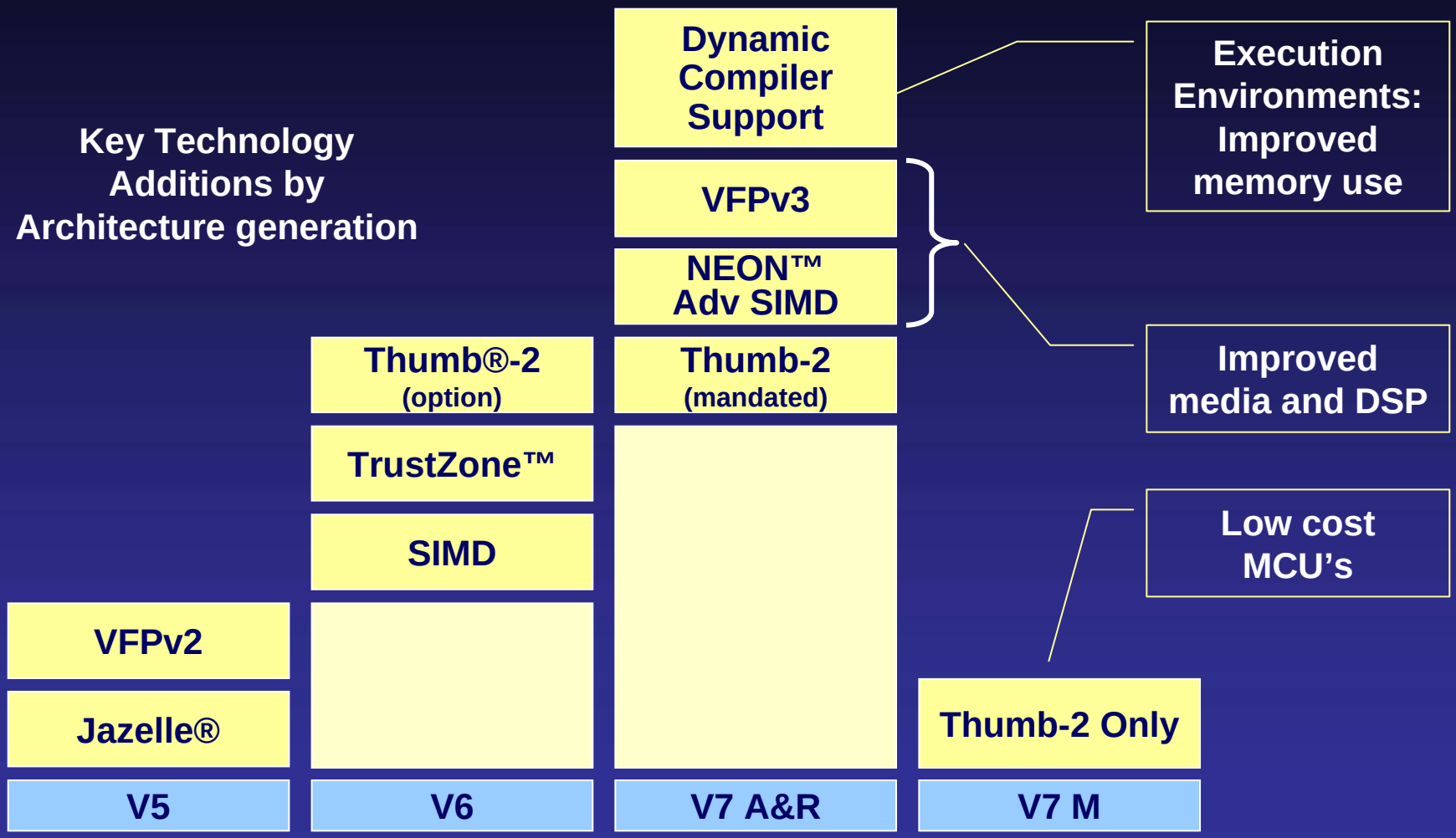
- **2004 shipments > 1.2B units (ARMv4T & ARMv5TE based)**
- **Rapid adoption of ARMv6 (ARM11™)**
- **New demands and usage diversity continue**

ARMv7: satisfying trends while maintaining consistency is key

Meeting the challenge with profiles

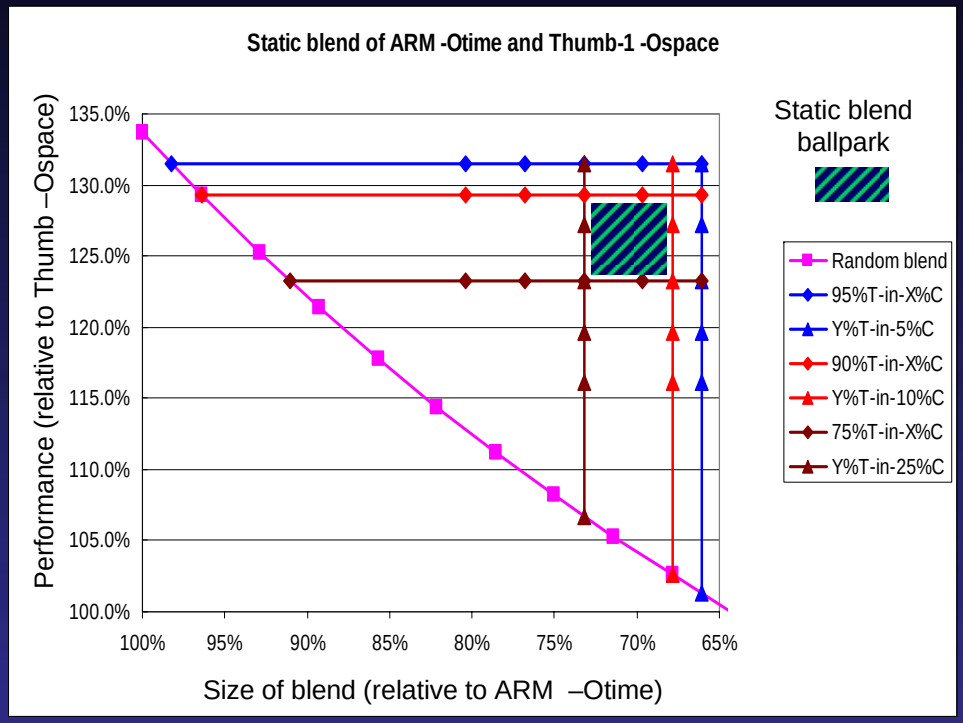
- The Application profile (ARMv7A)
  - Memory management support (MMU)
  - Highest performance at low power
    - Influenced by multi-tasking OS system requirements
- The Real-time profile (ARMv7R)
  - Protected memory (MPU)
  - Low latency and predictability ‘real-time’ needs
  - Evolutionary path for traditional embedded business
- The Microcontroller profile (ARMv7M)
  - Lowest gate count entry point
  - Deterministic behaviour a key priority
  - Deeply embedded – strong synergies with ‘ARMv7R’

Arch Evolution and Key Themes



Thumb-2

- Thumb required compiler tradeoffs
  - 'Osize' versus 'Otime'
  - Compile by function
- Thumb-2 offers a better compiler target
  - A single ISA
    - 16-/32-bit instructions
  - Optimise for a 'static blend'
- Delivering Performance and Code Density together

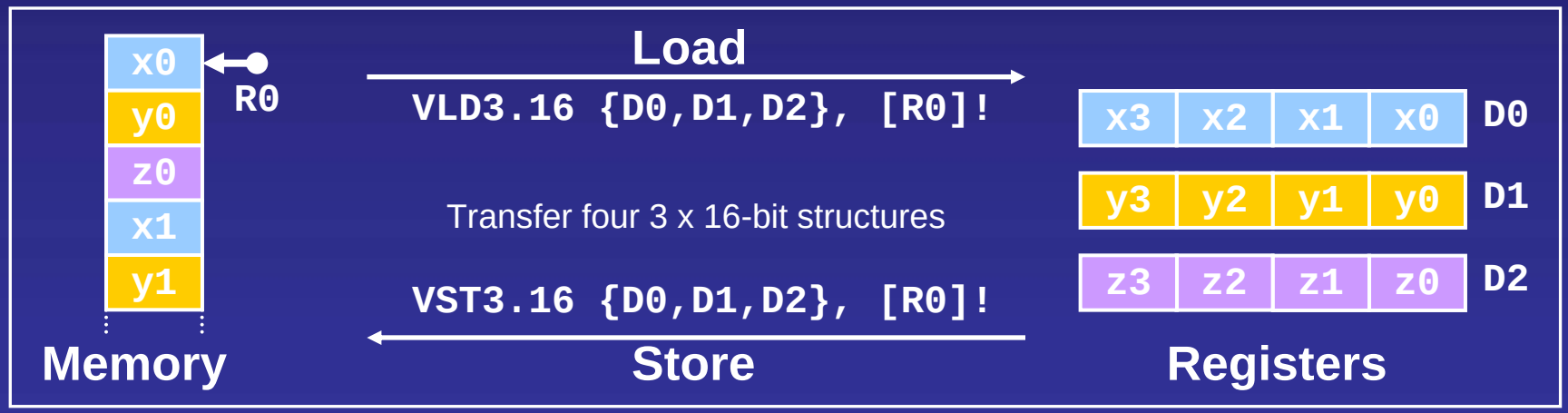


NEON - Advanced SIMD

- Packed SIMD processing
  - Integer arithmetic and single-precision FP option
    - 8-, 16-, 32- & 64-bit data types
    - (un)signed; float32; poly{8,16}
- Designed for vectorising compiler support
  - Register & Element/Structure (interleaved) load/stores
  - Promotion/demotion of data operation results
    - Removes need for packing/unpacking instructions
- Tightly integrated, separate execution hardware
  - Register file shared with VFP
    - Used as 32 x 64-bit or 16 x 128-bit registers
  - Provides a significant performance boost over ARMv6 SIMD

Adv SIMD – element LD/ST example

- In General:
  - LD/ST all elements to/from 1-4 registers
  - LD/ST a single element or replicate (LD) within a vector
  - For 2-4 element structures
    - Load/store an element to/from a set of structure vectors
    - Replicate a structure through a set of vectors
    - Load a set of structures across a set of vectors



VFPv3 – Floating Point support

- Builds on VFPv2
  - Double precision register count increased from 16 to 32
  - Fixed ⇔ Float conversion instructions
    - Signed and unsigned conversions
    - Integer value: 16- or 32-bit
    - FP value: single or double
  - Floating point constant loads
- User traps now an architecture option (VFPv3U)
  - A change of emphasis from VFPv2
  - Fits in with Advanced SIMD exception-free execution

  Trap free operation provides best performance
  & best suited to the majority of ARM target markets

Execution Environment Support

- Supports JIT/DAC and AOT techniques
  - JIT = ‘Just in Time’ compiler (real-time)
  - DAC = ‘Dynamic Adaptive Compilation’ (real-time)
  - AOT = ‘Ahead of Time’ compiler (install or on download)
- Improved memory efficiency
  - ROM, RAM and cache
  - Benefits apply to down-loadable applications in RAM as well as those pre-installed in ROM

Execution Environment - Features

- New Execution state
  - Derived from Thumb-2
  - Delivers excellent code density
- Key ISA changes from Thumb-2
  - New sub-routine (handler) mechanism
  - CHKA: Array checking instruction
  - Null pointer checks on LD/ST's
- Detailed announcement later this year

ARMv7M - Motivations

Improved support for deeply embedded and microcontroller markets

- Reduced hardware / core size
- Excellent interrupt latency
- Ease of high level language programming
- Strong upwards compatibility with the general embedded business

ARMv7M – a new programmer's model

- ARMv7A, ARMv7R
  - Banked registers in privileged modes
    - SP, LR and SPSR (plus r8-r12 in FIQ mode)
  - Software save/restore of registers ⇔ memory
  - CP15: System control coprocessor
  - ARM and Thumb-2 execution
  - CP14 or CoreSight™ debug
- ARMv7M
  - Only stack pointer (SP) banked
    - User/privilege support
  - Save/restore ⇔ stack in hardware (r0-r3, r12, LR, PC, xPSR)
    - Supports ARM's C/C++ procedure calling standard
  - Memory mapped system
    - Integral exception and interrupt control
    - Access to state, debug etc.
  - Thumb-2 only
  - Single-wire debug

Compatibility: ARMv7M <=> ARMv7R

OSs easily ported. Applications can be 100% compatible.

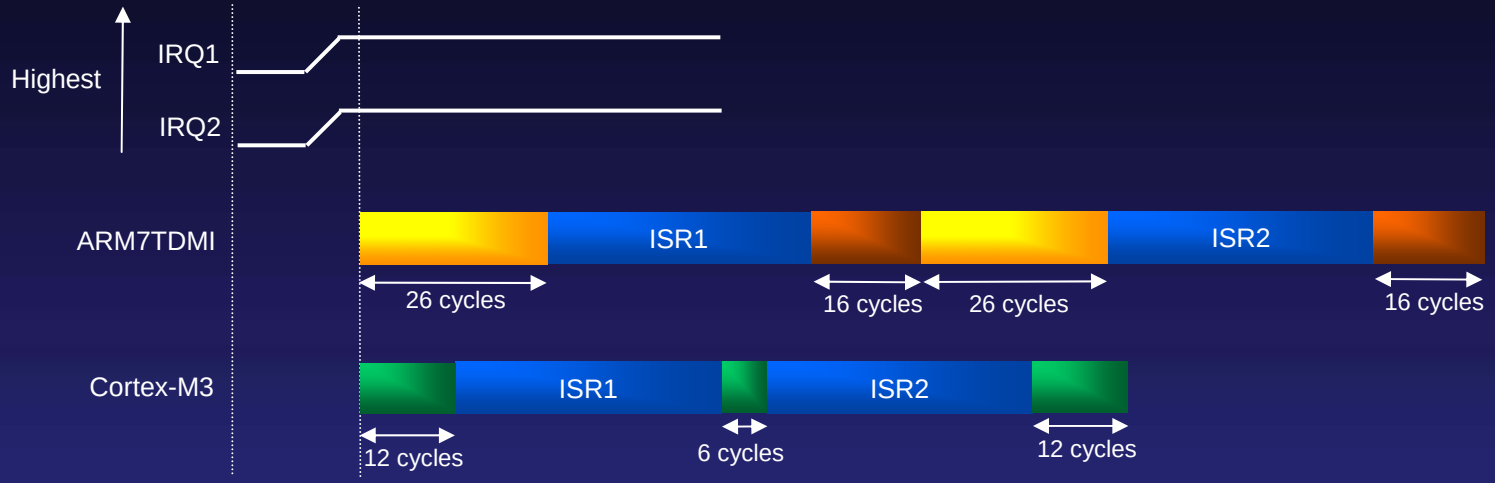
ARMv7M - features

- Integrated, vectored exception controller
  - Flexible priority scheme
  - Pre-emption and tail-chaining capabilities
- Multiple stacks
  - User/Privileged operation supported
    - Recognises growing need in the micro-controller space
  - Fits common OS and RTOS usage models
- Architected memory map includes:
  - System access: exception/interrupt control, debug
  - Bit-banding support for simple read-modify-write
    - Simplifies bit manipulation in memory
  - Optional MPU (PMSA model aligns with ARMv7R)

ARMv7M – Exceptions and Interrupts

- Features contributing to minimum latency and improved performance
  - Vectored entry
  - Push/pop to the stack in hardware
  - Flexible prioritisation scheme
- Two key interrupt latency optimisations supported
  - Late-arrival Pre-emption
    - Switches to higher priority interrupt during exception entry
  - Tail-chaining – exit/entry optimisation
    - Avoids unnecessary pop + push of registers

ARMv7M – Tail-chaining example



ARM7TDMI®

Software pre/post amble

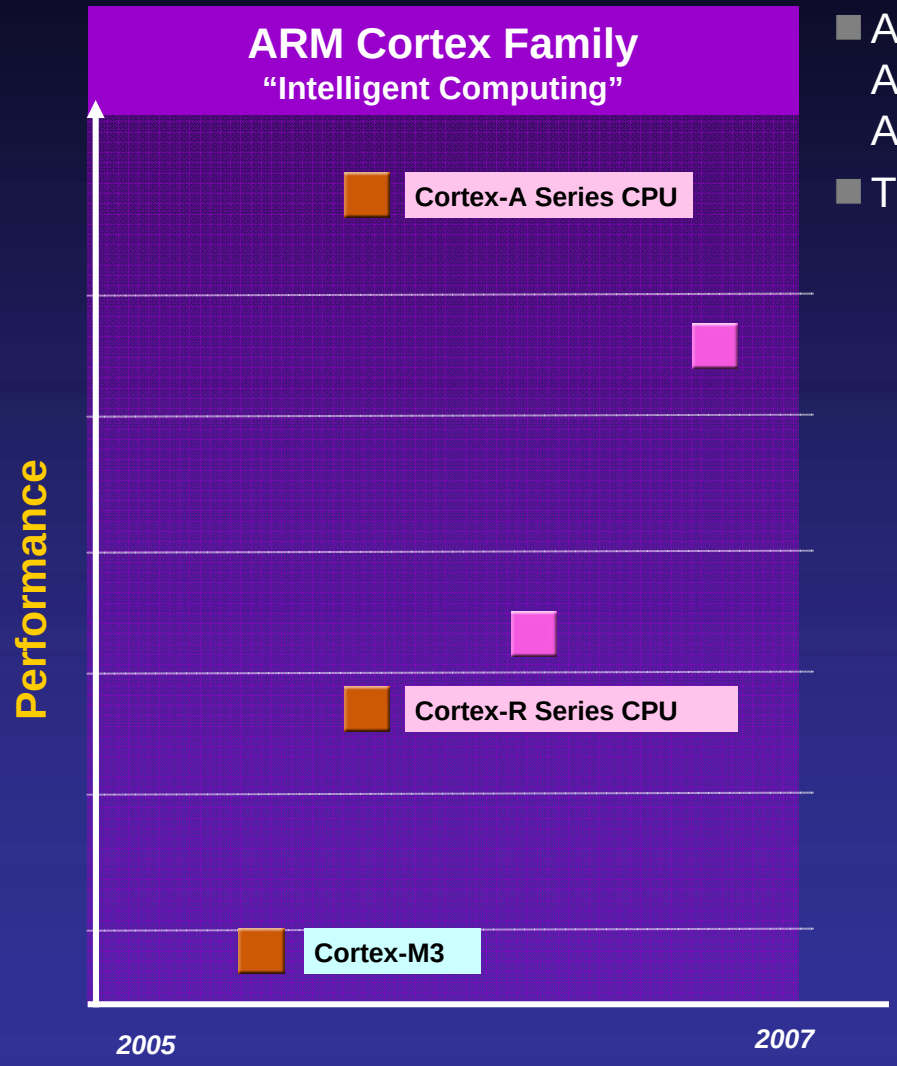
- 26 cycles from IRQ1 -> ISR1 entered
- 42 cycles from ISR1 exit -> ISR2 entry
- 16 cycles to return from ISR2 -> interrupted code

Cortex™-M3

Hardware ISR entry/exit

- 12 cycles from IRQ1 -> ISR1 entered
- 6 cycles from ISR1 exit -> ISR2 entry
- 12 cycles to return from ISR2 -> interrupted code.

The ARM Cortex™ Family



- A New family of products from ARM based on the ARMv7 Architecture, Thumb®-2 Instruction Set and AMBA™ AXI interface specification
- Three main series
  - ***ARM Cortex A Series -*** ***Applications*** CPUs focused on the execution of complex OS and user applications
    - First Product: To be announced
    - Executes ARM, Thumb-2 instructions
  - ***ARM Cortex R Series*** **-** Deeply embedded processors focused on ***real-time*** environments
    - First Product: To be announced
    - Executes ARM, Thumb-2 instructions
  - ***ARM Cortex M Series*** **-** ***Microcontroller*** *cores* focused on very cost sensitive, deterministic, interrupt driven environments
    - First Product: ARM Cortex-M3
    - Executes Thumb-2 instructions