The ARM Architecture

Ali Saidi

Agenda

- Introduction to ARM Ltd

ARM Architecture/Programmers Model

Data Path and Pipelines

System Design

Development Tools

ARM Ltd

- Founded in November 1990
  - Spun out of Acorn Computers
- Designs the ARM range of RISC processor cores
- Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers.
  - ARM does not fabricate silicon itself
- Also develop technologies to assist with the design-in of the ARM architecture
  - Software tools, boards, debug hardware, application software, graphics, bus architectures, peripherals, cell libraries

The Architecture for the Digital World

ARM designs technology that lies at the heart of advanced digital products

ARM's Activities

Connected Community

Development Tools

Software IP

Processors

System Level IP:

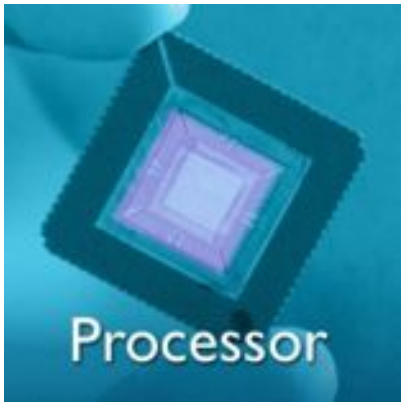
Data Engines

Fabric

3D Graphics

Physical IP

Product Areas



- Cortex-M0
  to
  Cortex-A9

- 180nm
  to
  28nm

- Compilers
  &
  OS
  profiling

- GPU
  to
  HD video

ARM Business Today

- Processor Shipped Last Year : ~4 Billion
- Processor Shipped In Total : >24 Billion
- Processor Licenses : 500+
- Semiconductor Partners : 200+
- Process Technology : 28 – 250 nm
- Connected Community Members : 700+

ARM Business Model Drivers

- Deliver more functionality to the end-user sooner and more cost-effectively
  - Integration
  - Economics
  - Focus
  - Ecosystem
  - Choice
  - Power efficiency

Global Company

700+


Silicon Partners
Design Support Partners
Software, Training and Consortia Partners

Nokia N95 Multimedia Computer

OMAP™ 2420 Applications Processor
ARM1136™ processor-based SoC, developed using Magma ® Blast® family and winner of 2005 INSIGHT Award for 'Most Innovative SoC'

Symbian OS™ v9.2
Operating System supporting ARM processor-based mobile devices, developed using ARM® RealView® Compilation Tools

S60™ 3rd Edition
S60 Platform supporting ARM processor-based mobile devices

Mobiclip™ Video Codec
Software video codec for ARM processor-based mobile devices

ST WLAN Solution
Ultra-low power 802.11b/g WLAN chip with ARM9™ processor-based MAC

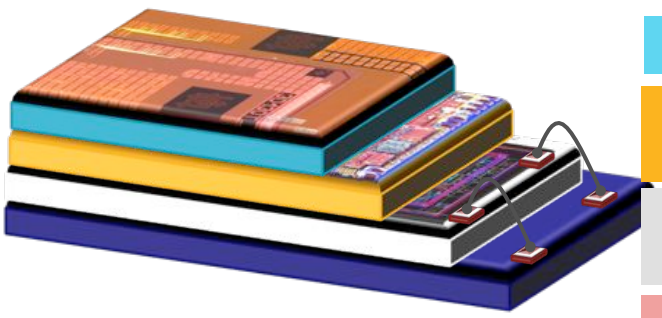
NOKIA
CONNECTING PEOPLE

Connect. Collaborate. Create.

ARM Processor Applications

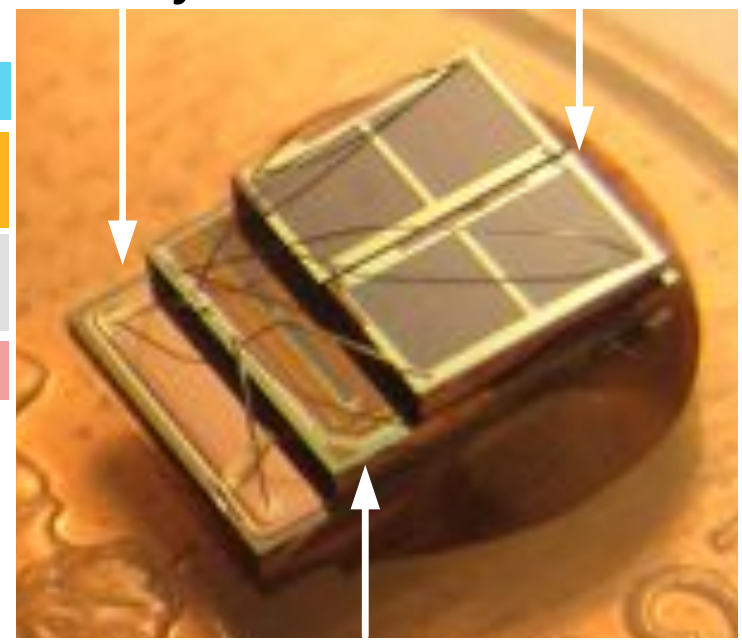
12

World's Smallest ARM Computer?



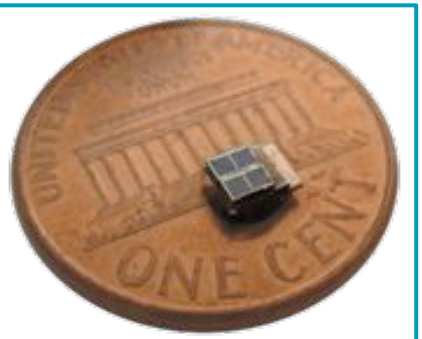
Wireless Sensor Network

Sensors, timers
Cortex-M0 +16KB RAM 65nm UWB Radio antenna
10 kB Storage memory ~3fW/ bit
12µAh Li-ion Battery



Processor, SRAM and PMU

Cortex-M0; 65¢



University of Michigan

Wirelessly networked into large scale sensor arrays

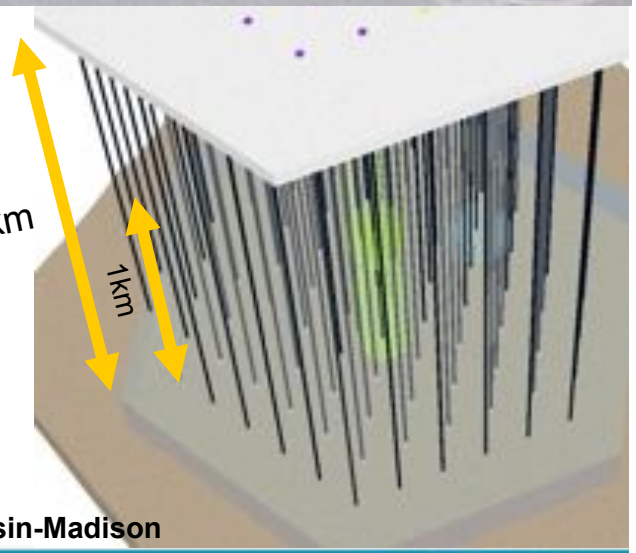
World's Largest ARM Computer?

4200 ARM powered Neutrino Detectors

70 bore holes 2.5km deep

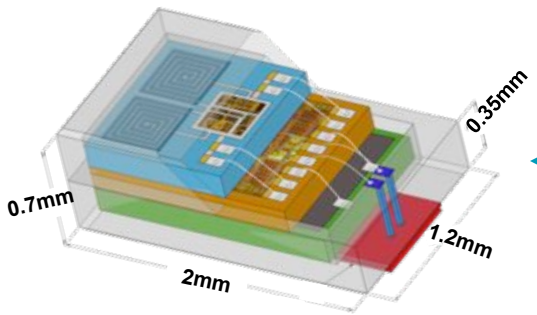
60 detectors per string starting 1.5km down

1km^3 of active telescope

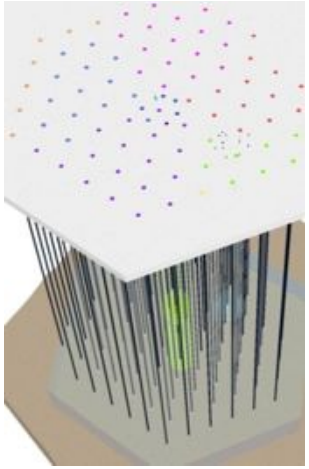


Work supported by the National Science Foundation and University of Wisconsin-Madison

From 1mm^3 to 1km^3



1mm^3 ⟷ 1km^3



10¢ ⟷ $1000

Mobile Home Mobile Computing Server

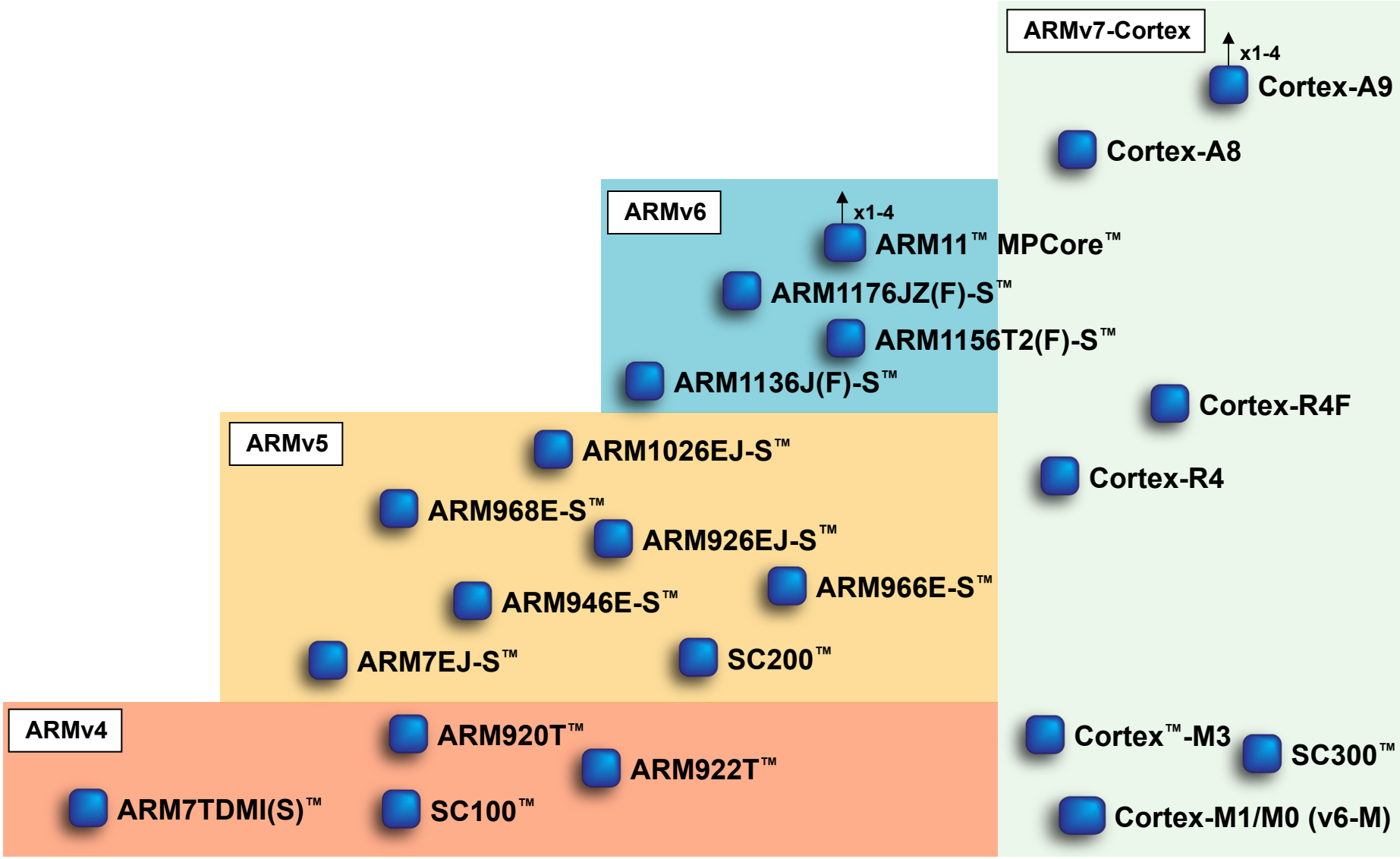
Embedded Consumer Enterprise PC HPC

The Architecture for the Digital World

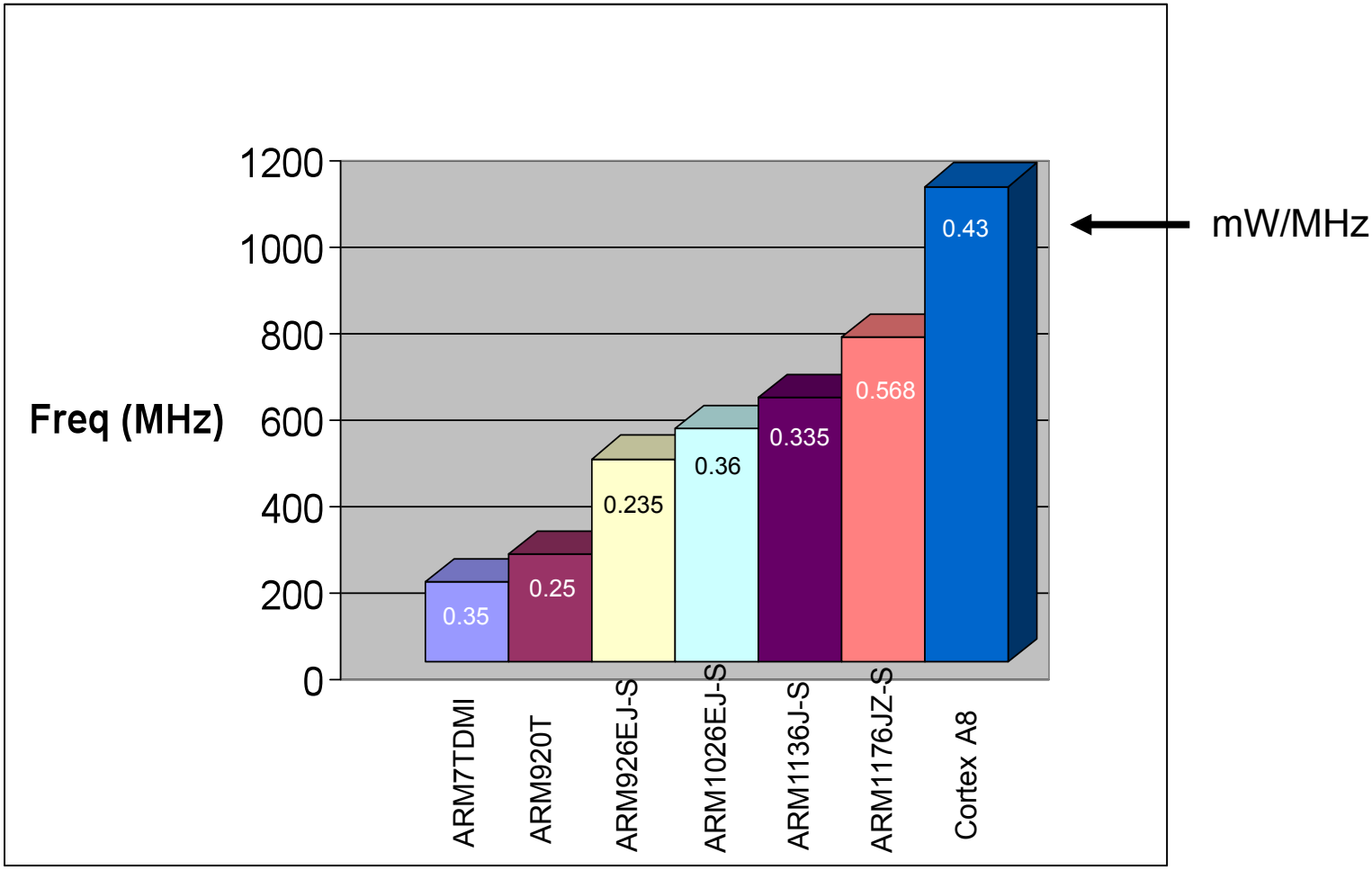
Agenda

- Introduction to ARM Ltd
- ARM Architecture/Programmers Model
- Data Path and Pipelines
- System Design
- Development Tools

Architecture Versions



Relative Performance*



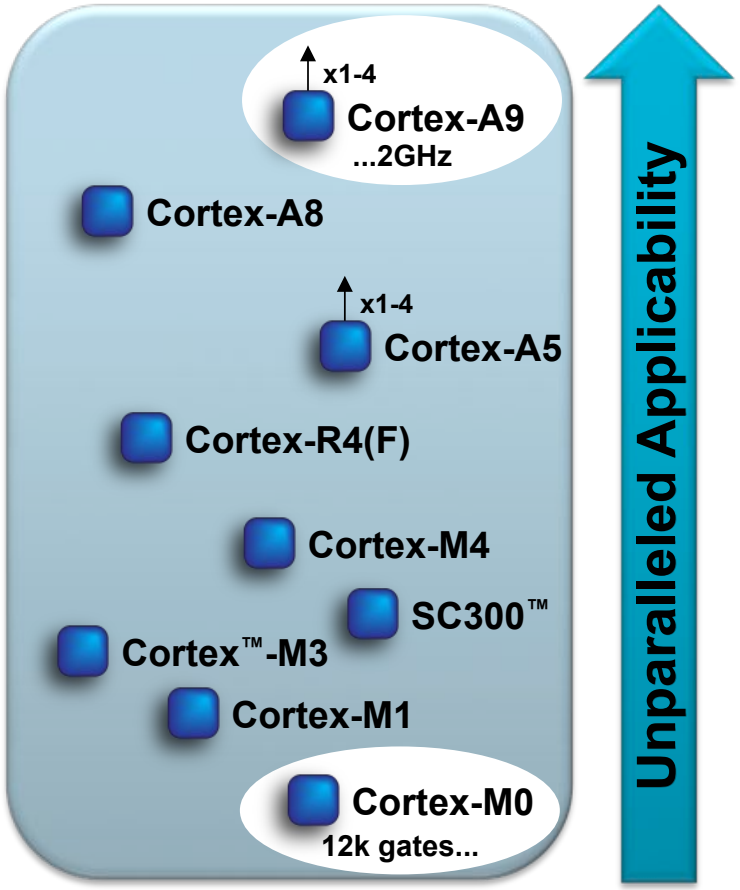
*Represents attainable speeds in 130, 90 or 65nm processes

ARM Cortex Advanced Processors

Architectural innovation, compatibility across diverse application spectrum

Cortex™
Low-Power Leadership from ARM®

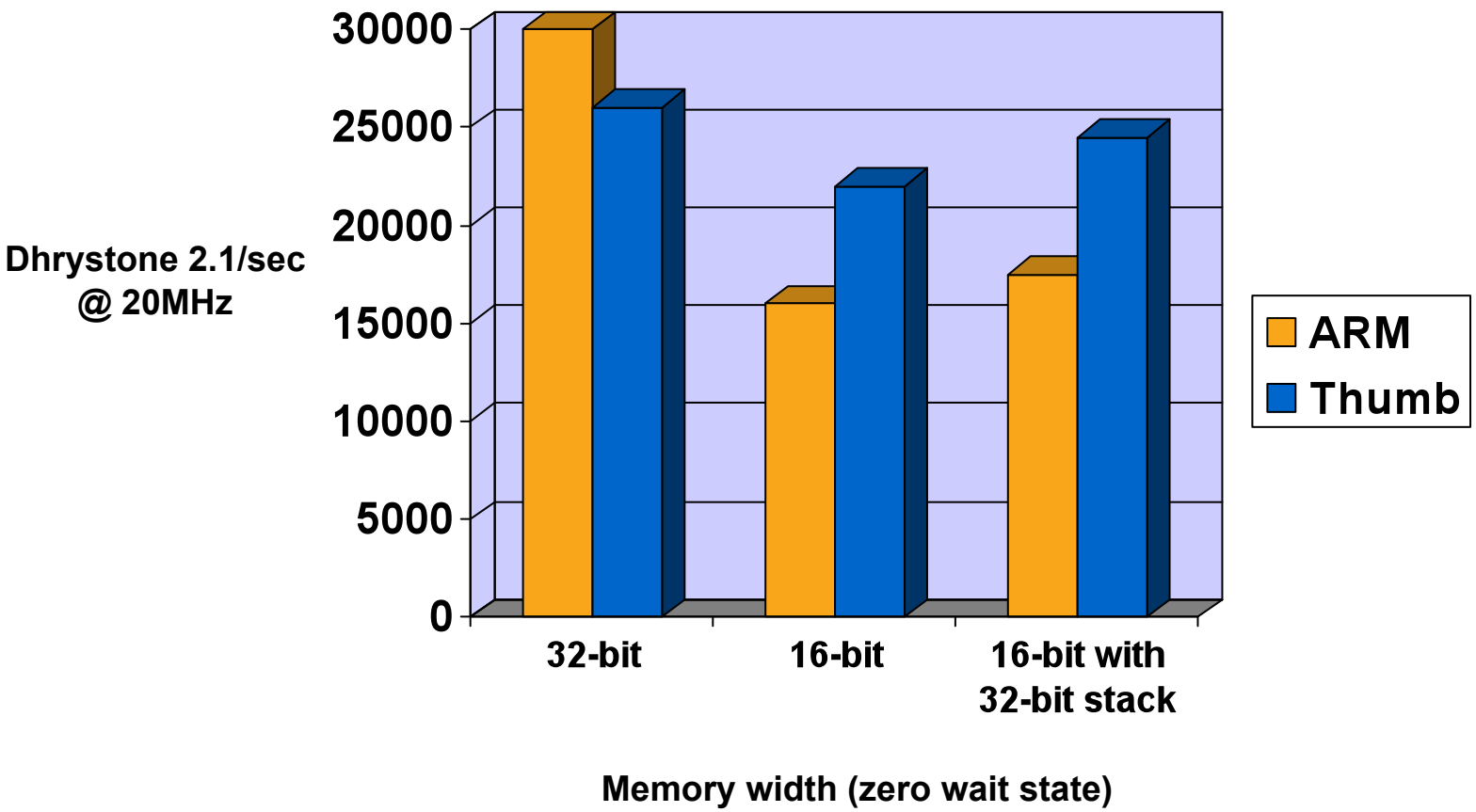
- ARM Cortex-**A** family:
  - Applications processors for feature-rich OS and 3rd party applications
- ARM Cortex-**R** family:
  - Embedded processors for real-time signal processing, control applications
- ARM Cortex-**M** family:
  - Microcontroller-oriented processors for MCU, ASSP, and SoC applications



Data Sizes and Instruction Sets

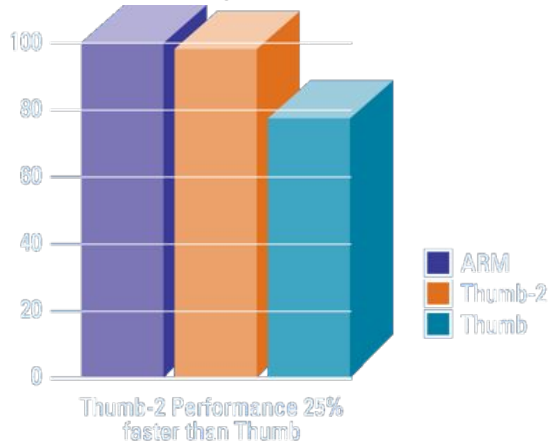
- The ARM is a 32-bit architecture.
- When used in relation to the ARM:
  - **Byte** means 8 bits
  - **Halfword** means 16 bits (two bytes)
  - **Word** means 32 bits (four bytes)
- Most ARM's implement two instruction sets
  - 32-bit ARM Instruction Set
  - 16-bit/32bit Thumb Instruction Set
- Jazelle cores can also execute Java bytecode

ARM and Thumb Performance

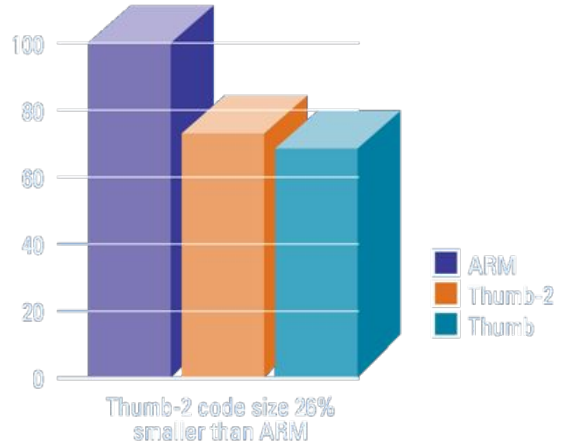


Thumb-2 Instruction Set

EEMBC Analysis - Performance



EEMBC Analysis – Code Size



- Second generation of the Thumb architecture
  - Blended 16-bit and 32-bit instruction set
  - 25% faster than Thumb
  - 30% smaller than ARM
- Increases performance but maintains code density
- Maximizes cache and tightly coupled memory usage

Processor Modes – A Class

- The ARM has seven basic operating modes:
  - **User** : unprivileged mode under which most tasks run
  - **FIQ** : entered when a high priority (fast) interrupt is raised
  - **IRQ** : entered when a low priority (normal) interrupt is raised
  - **Supervisor** : entered on reset and when a Software Interrupt instruction is executed
  - **Abort** : used to handle memory access violations
  - **Undef** : used to handle undefined instructions
  - **System** : privileged mode using the same registers as user mode

The ARM Register Set

Current Visible Registers

Abort Mode

r0
r1
r2
r3
r4
r5
r6
r7
r8
r9
r10
r11
r12
r13 (sp)
r14 (lr)
r15 (pc)

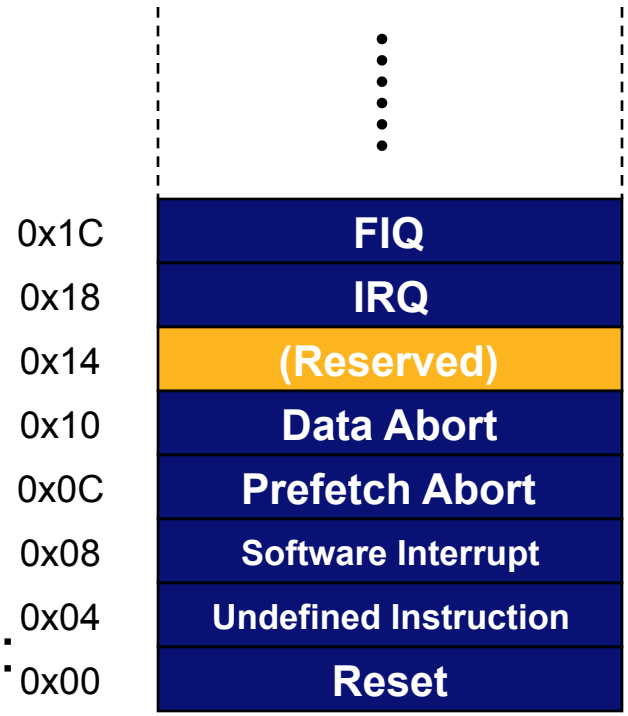
cpsr
spsr

Banked out Registers

User	FIQ	IRQ	SVC	Undef
	r8			
	r9			
	r10			
	r11			
	r12			
r13 (sp)	r13 (sp)	r13 (sp)	r13 (sp)	r13 (sp)
r14 (lr)	r14 (lr)	r14 (lr)	r14 (lr)	r14 (lr)
	spsr	spsr	spsr	spsr

Exception Handling

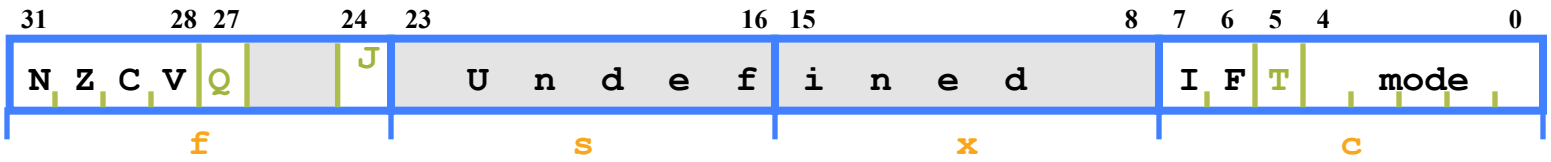
- When an exception occurs, the ARM:
  - Copies CPSR into SPSR_<mode>
  - Sets appropriate CPSR bits
    - Change to ARM or Thumb state
    - Change to exception mode
    - Disable interrupts (if appropriate)
  - Stores the return address in LR_<mode>
  - Sets PC to vector address
- To return, exception handler needs to:
  - Restore CPSR from SPSR_<mode>
  - Restore PC from LR_<mode>



Vector Table

Vector table can be at **`0xFFFF0000`** on ARM720T and on ARM9/10 family devices

Program Status Registers



- Condition code flags
  - N = **N**egative result from ALU
  - Z = **Z**ero result from ALU
  - C = ALU operation **C**arried out
  - V = ALU operation o**V**erflowed
- Sticky Overflow flag - Q flag
  - Architecture v5+ only
  - Indicates if saturation has occurred
- J bit
  - Architecture v5+ only
  - J = 1: Processor in Jazelle state
- Interrupt Disable bits.
  - I = 1: Disables the IRQ.
  - F = 1: Disables the FIQ.
- T Bit
  - Architecture v5+ only
  - T = 0: Processor in ARM state
  - T = 1: Processor in Thumb state
- Mode bits
  - Specify the processor mode

Conditional Execution and Flags

- ARM instructions can be made to execute conditionally by postfixing them with the appropriate condition code field.
  - This improves code density *and* performance by reducing the number of forward branch instructions.

```
  CMP   r3,#0
  BEQ   skip
  ADD   r0,r1,r2
skip
```

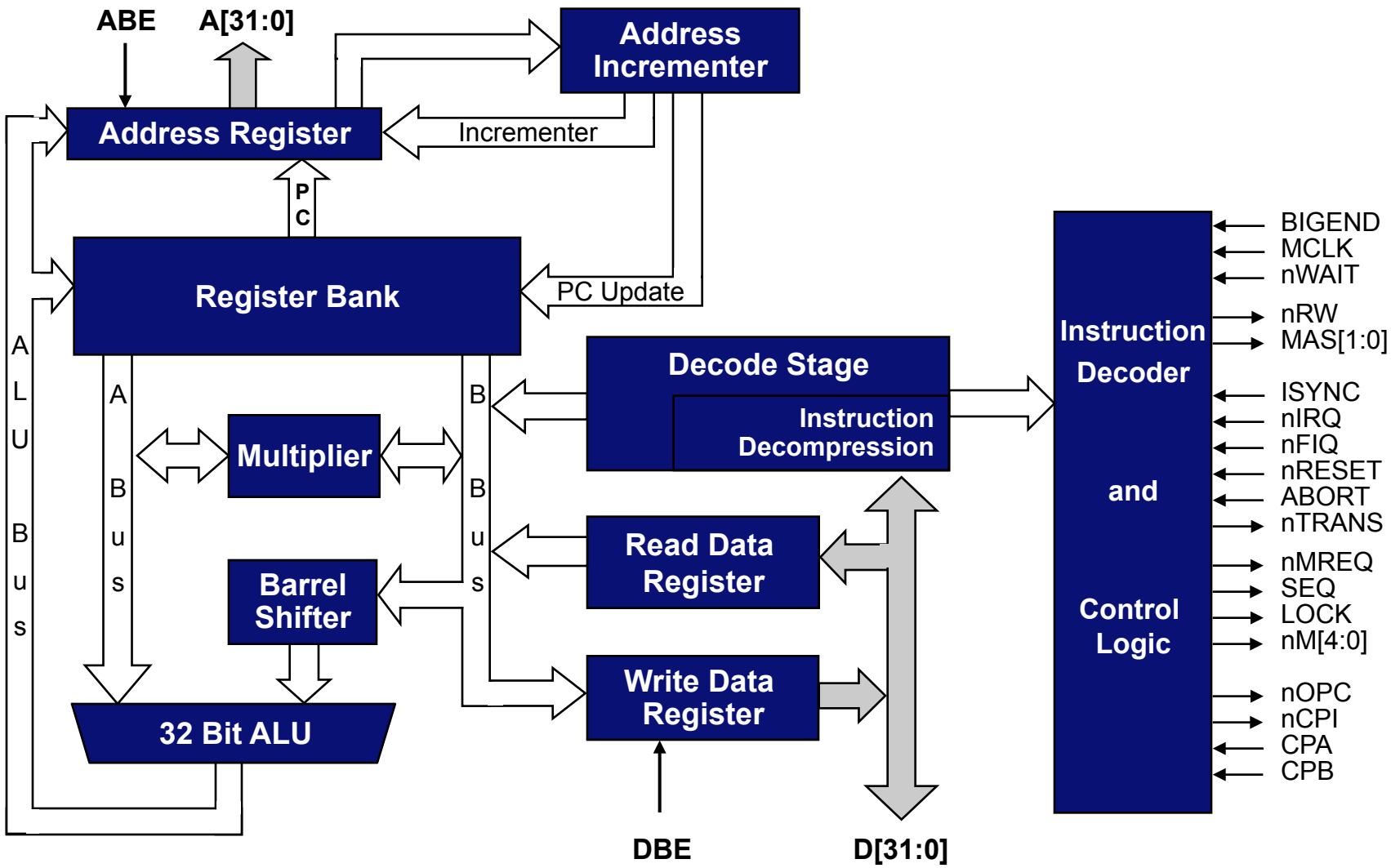
```
CMP    r3,#0
ADDNE  r0,r1,r2
```

- Why was this developed?
  - When would you want to use it? Always? Any downsides?

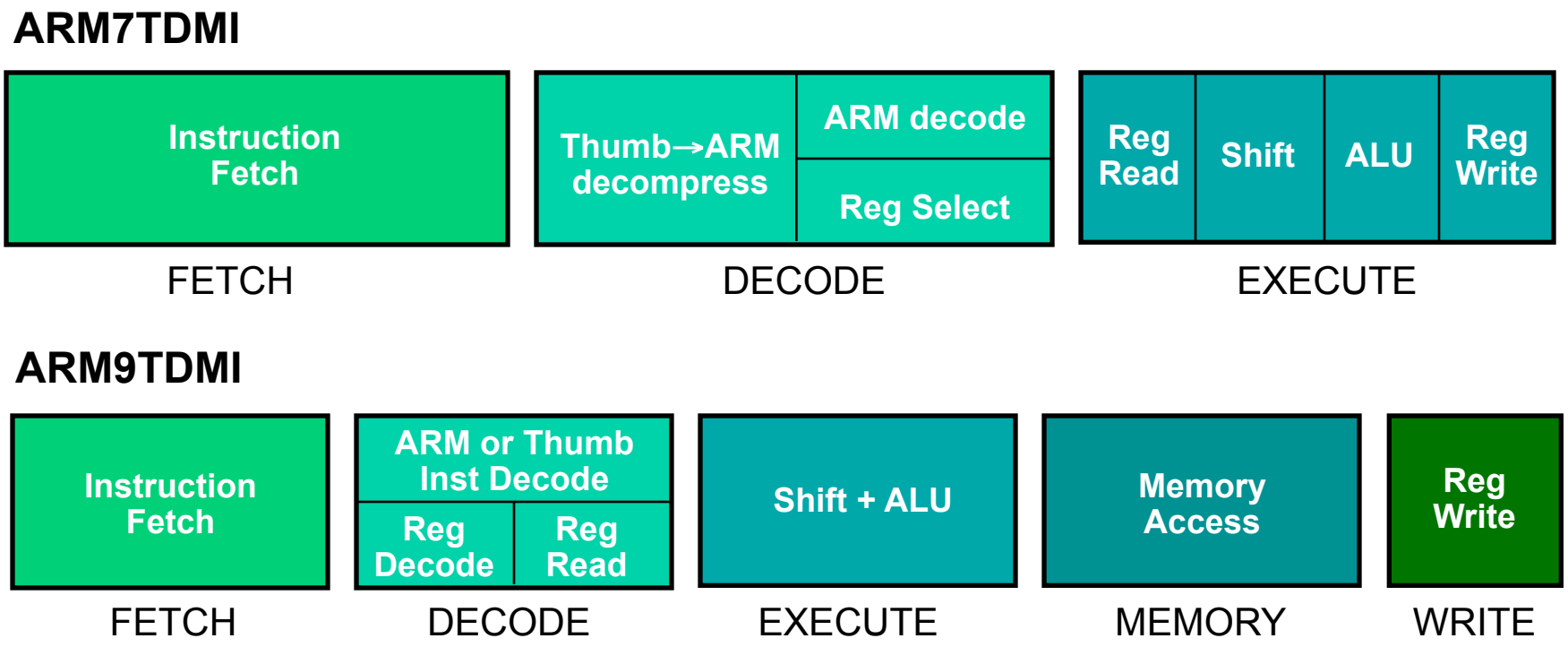
Agenda

- Introduction to ARM Ltd
- ARM Architecture/Programmers Model
- Data Path and Pipelines
- System Design
- Development Tools

The ARM7TDM Core

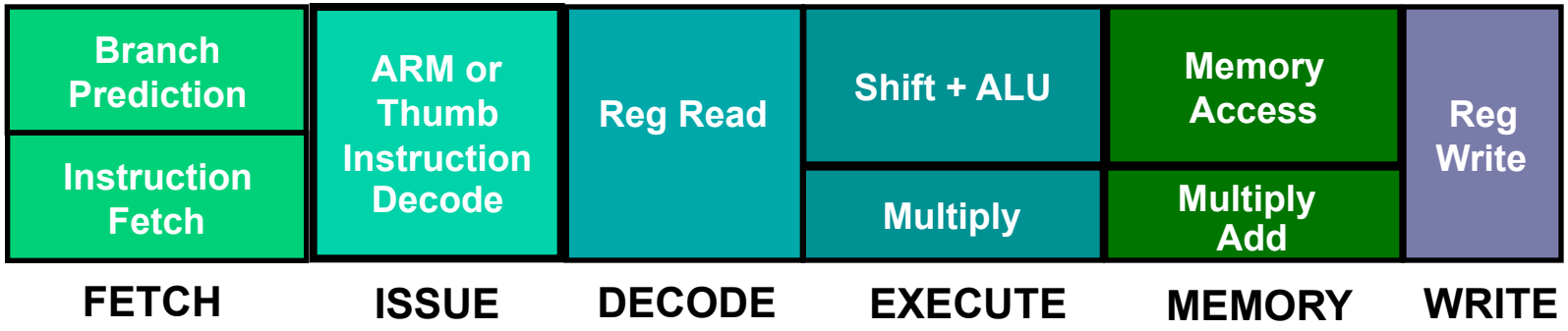


Pipeline changes for ARM9TDMI

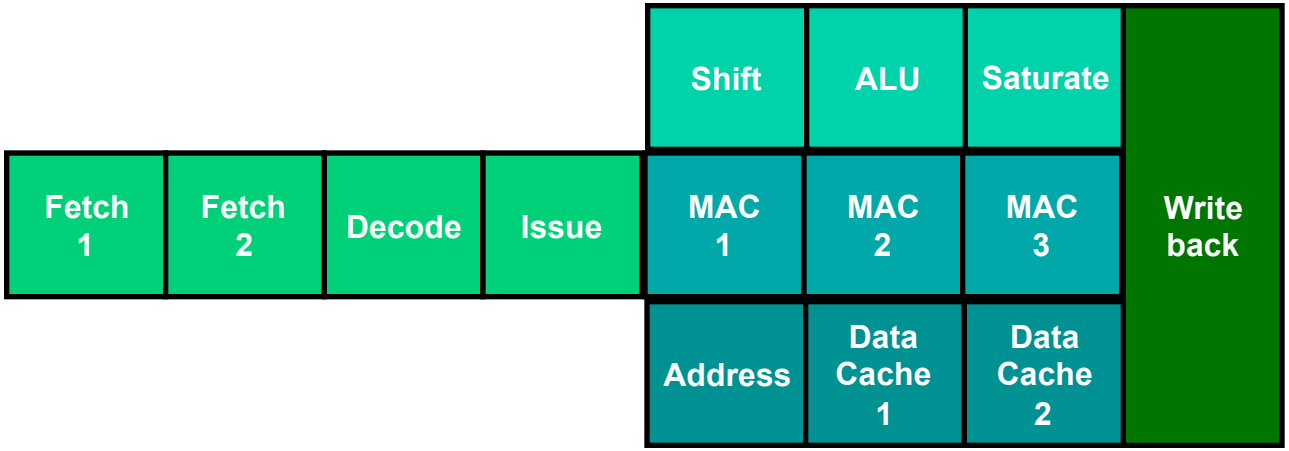


ARM10 vs. ARM11 Pipelines

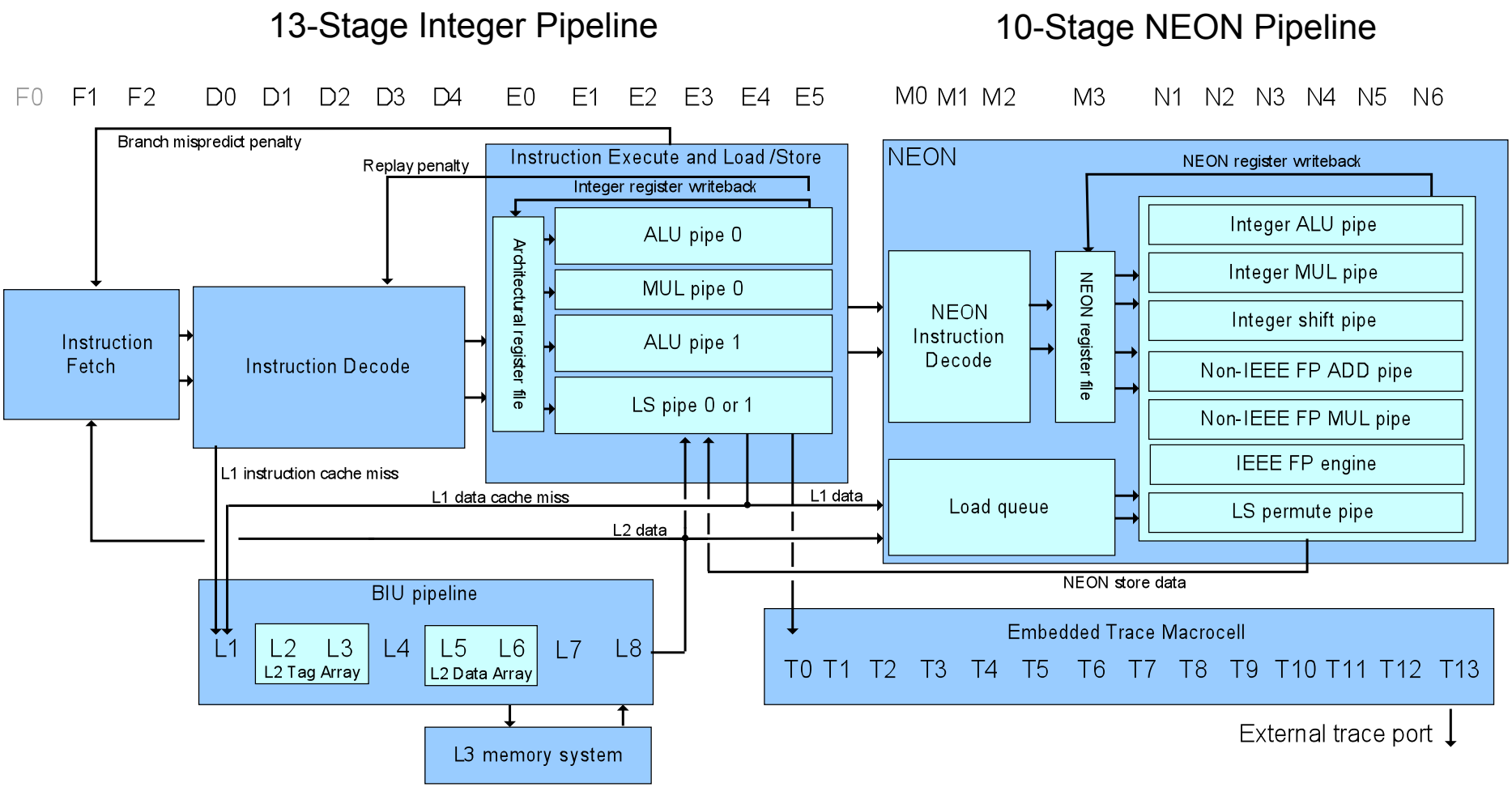
ARM10



ARM11



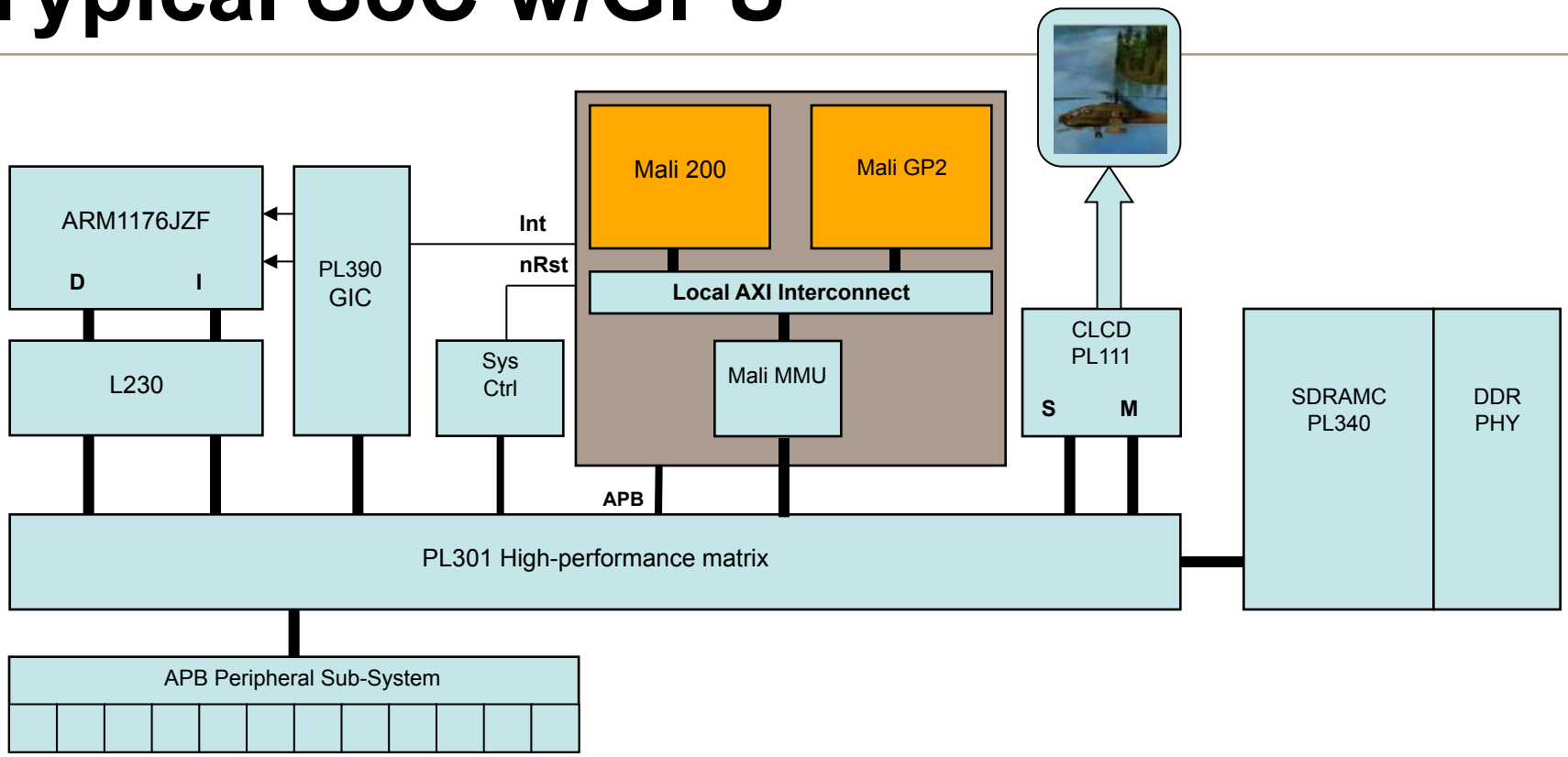
Full Cortex-A8 Pipeline Diagram



Agenda

- Introduction to ARM Ltd
- ARM Architecture/Programmers Model
- Data Path and Pipelines
- System Design
- Development Tools

Typical SoC w/GPU

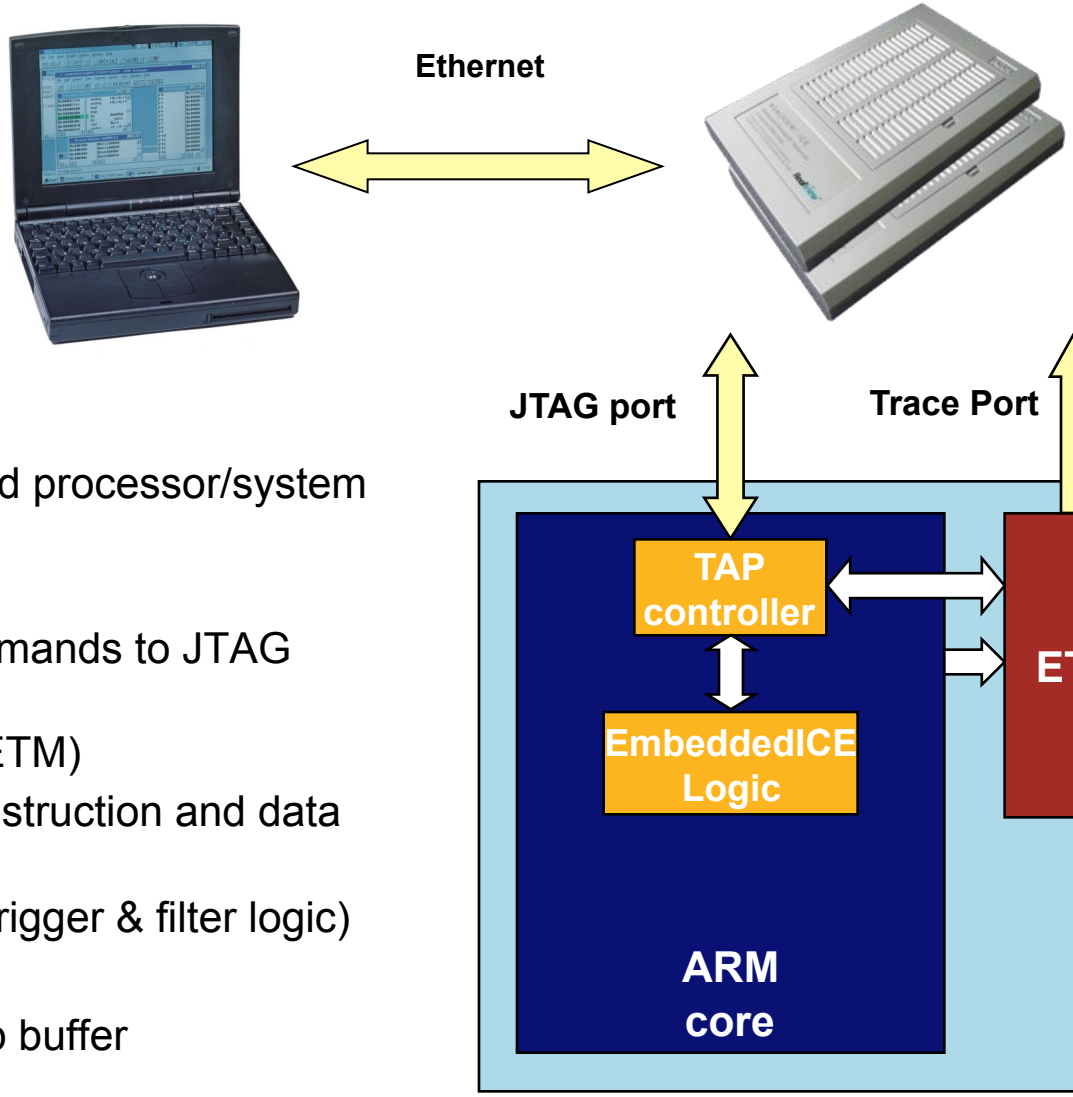


- Designed and optimised for AMBA: provides easier integration with ARM cores and fabric IP
- Unified Memory Architecture

Agenda

- Introduction to ARM Ltd
- ARM Architecture/Programmers Model
- Data Path and Pipelines
- System Design
- Development Tools

ARM Debug Architecture



- EmbeddedICE Logic
  - Provides breakpoints and processor/system access
- JTAG interface (ICE)
  - Converts debugger commands to JTAG signals
- Embedded trace Macrocell (ETM)
  - Compresses real-time instruction and data access trace
  - Contains ICE features (trigger & filter logic)
- Trace port analyzer (TPA)
  - Captures trace in a deep buffer

Keil Development Tools for ARM

- Includes ARM macro assembler, compilers (ARM RealView C/C++ Compiler, Keil CARM Compiler, or GNU compiler), ARM linker, Keil uVision Debugger and Keil uVision IDE
- Keil uVision Debugger accurately simulates on-chip peripherals (I^2C, CAN, UART, SPI, Interrupts, I/O Ports, A/D and D/A converters, PWM, etc.)
- Evaluation Limitations
  - 16K byte object code + 16K data limitation
  - Some linker restrictions such as base addresses for code/constants
  - GNU tools provided are not restricted in any way
- http://www.keil.com/demo/

Keil Development Tools for ARM

Hello - µVision3 - [C:\Keil\ARM\Examples\Hello\Hello.c]

File Edit View Project Debug Flash Peripherals Tools SVCS Window Help

```
/*  This file is part of the uVision/ARM development tools
/*  Copyright KEIL ELEKTRONIK GmbH 2002-2004
/*
/*  HELLO.C:  Hello World Example

#include <stdio.h>                  /* prototype declarations for I/O functions */
#include <LPC21xx.H>                /* LPC21xx definitions

/* main program */
int main (void)  {                  /* execution starts here

  /* initialize the serial interface  */
  PINSEL0 = 0x00050000;             /* Enable RxD1 and TxD1
  U1LCR = 0x83;                     /* 8 bits, no Parity, 1 Stop bit
  U1DLL = 97;                       /* 9600 Baud Rate @ 15MHz VPB Clock
  U1LCR = 0x03;                     /* DLAB = 0

  printf ("Hello World\n");         /* the "printf" function call

  while (1) {                       /* An embedded program does not stop and
```

```
MISSING DEVICE (R003: SECURITY KEY NOT FOUND)
Running in Eval Mode
Load "C:\\Keil\\ARM\\Examples\\Hello\\Obj\\Hello.ELF"

*** Restricted Version with 16384 Byte Code Size Limi
*** Currently used: 1980 Bytes (12%)

ASSIGN BreakDisable BreakEnable BreakKill BreakList
```

Ready | Simulation | t1: 0.72642057 sec | L:29 C:1

University Resources

- http://www.arm.com/support/university/
- University@arm.com

Fin