

The ARM Architecture



Agenda

- Introduction to ARM Ltd

ARM Architecture/Programmers Model

Data Path and Pipelines

AMBA

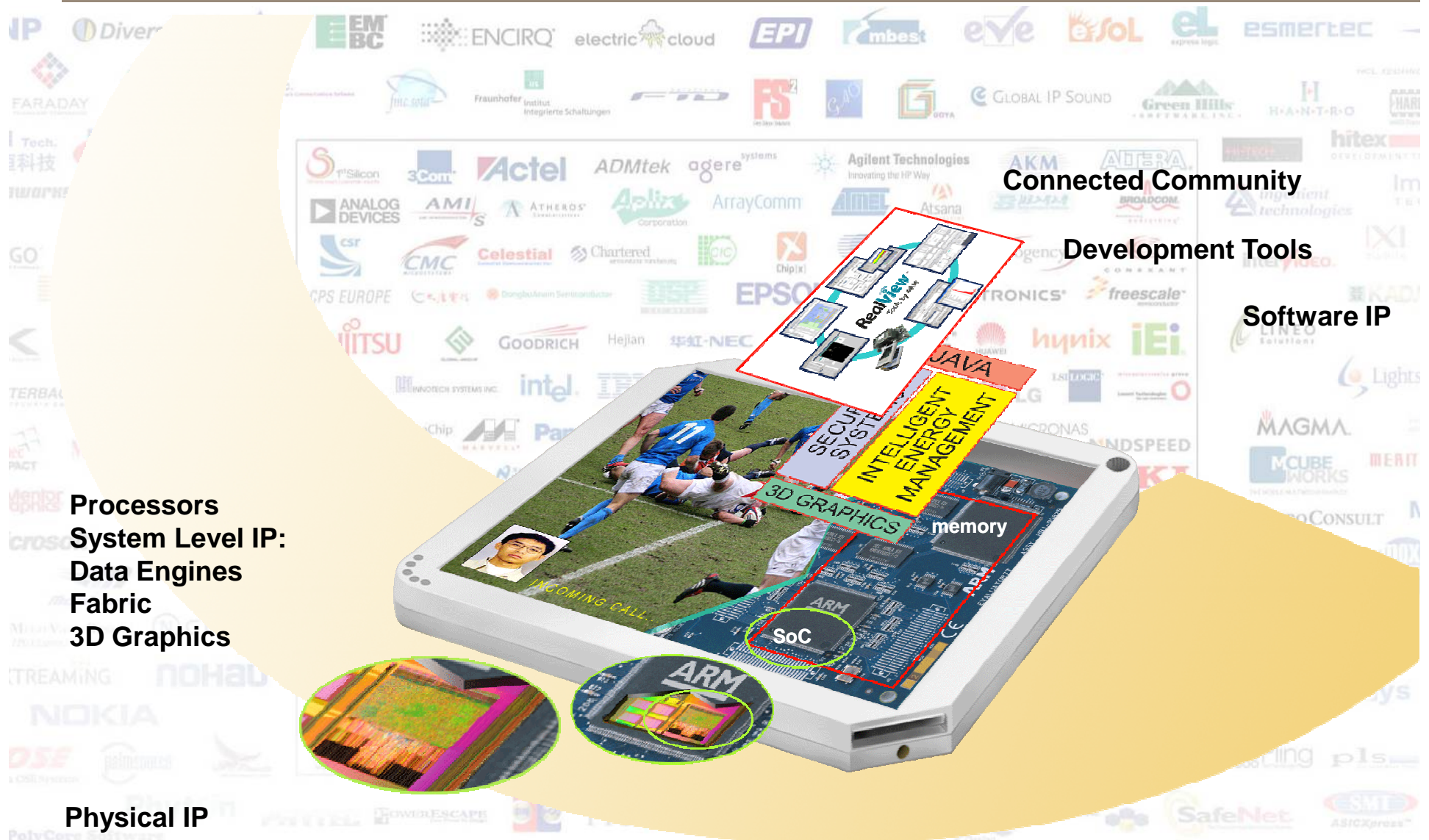
Development Tools

ARM Ltd

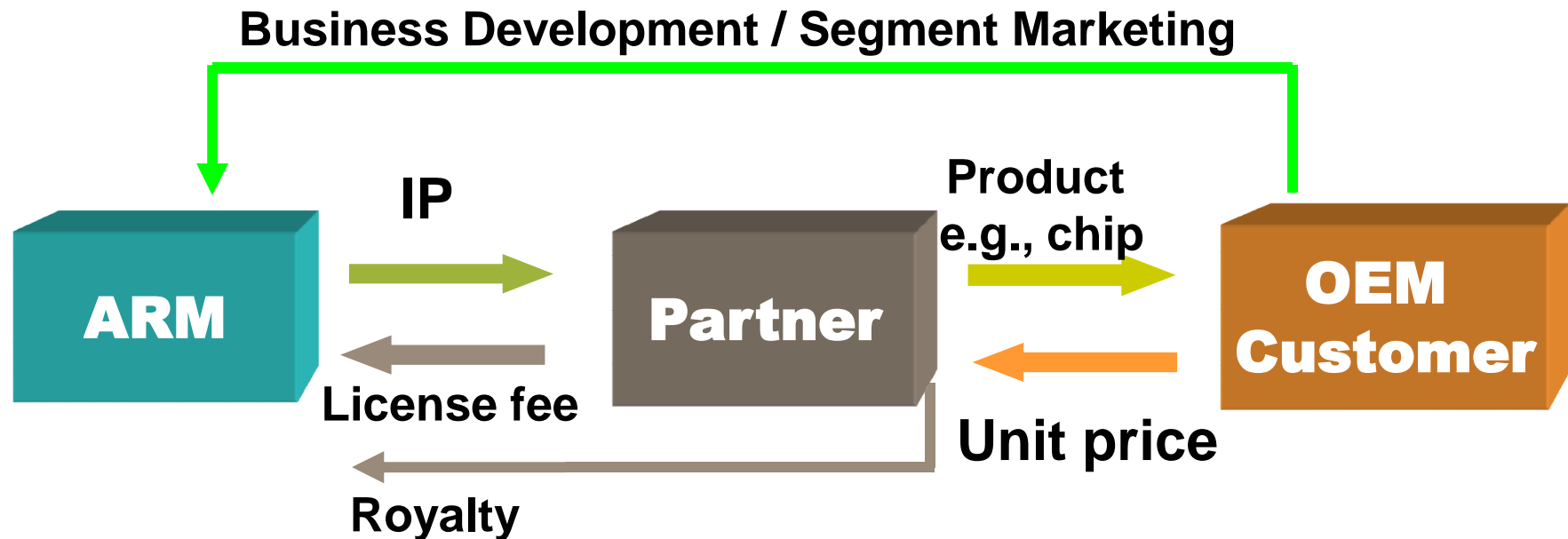
- Founded in November 1990
- Spun out of Acorn Computers
- Designs a range of RISC processor cores
- Licenses ARM core designs to semiconductor partners who fabricate and sell to their customers.
 - **ARM does not fabricate silicon itself**
 - **ARM is not the only company to design cores**
- Develops technologies to assist with the design-in of the ARM architecture
 - Software tools, application software
 - Development boards, debug hardware
 - Bus architectures, peripherals, etc.



ARM's Activities



The ARM Business Model



ARM creates IP through design activity

Partner combines ARM IP and Partner IP/technology into product

OEM builds ARM core-based product from Partner into end-system product

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

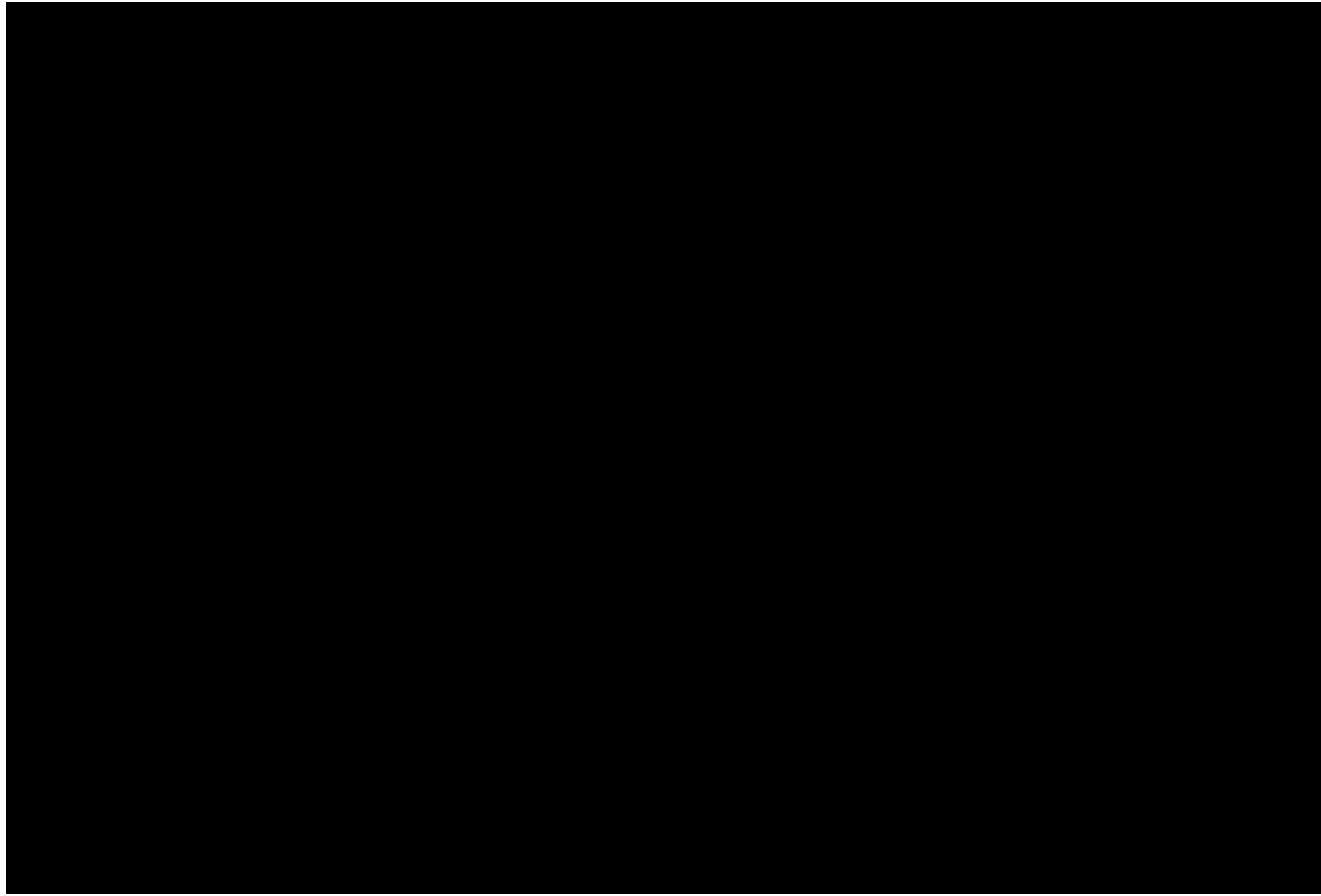
ST WLAN Solution

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



NOKIA
CONNECTING PEOPLE

Connect. Collaborate. Create.



Agenda

Introduction to ARM Ltd

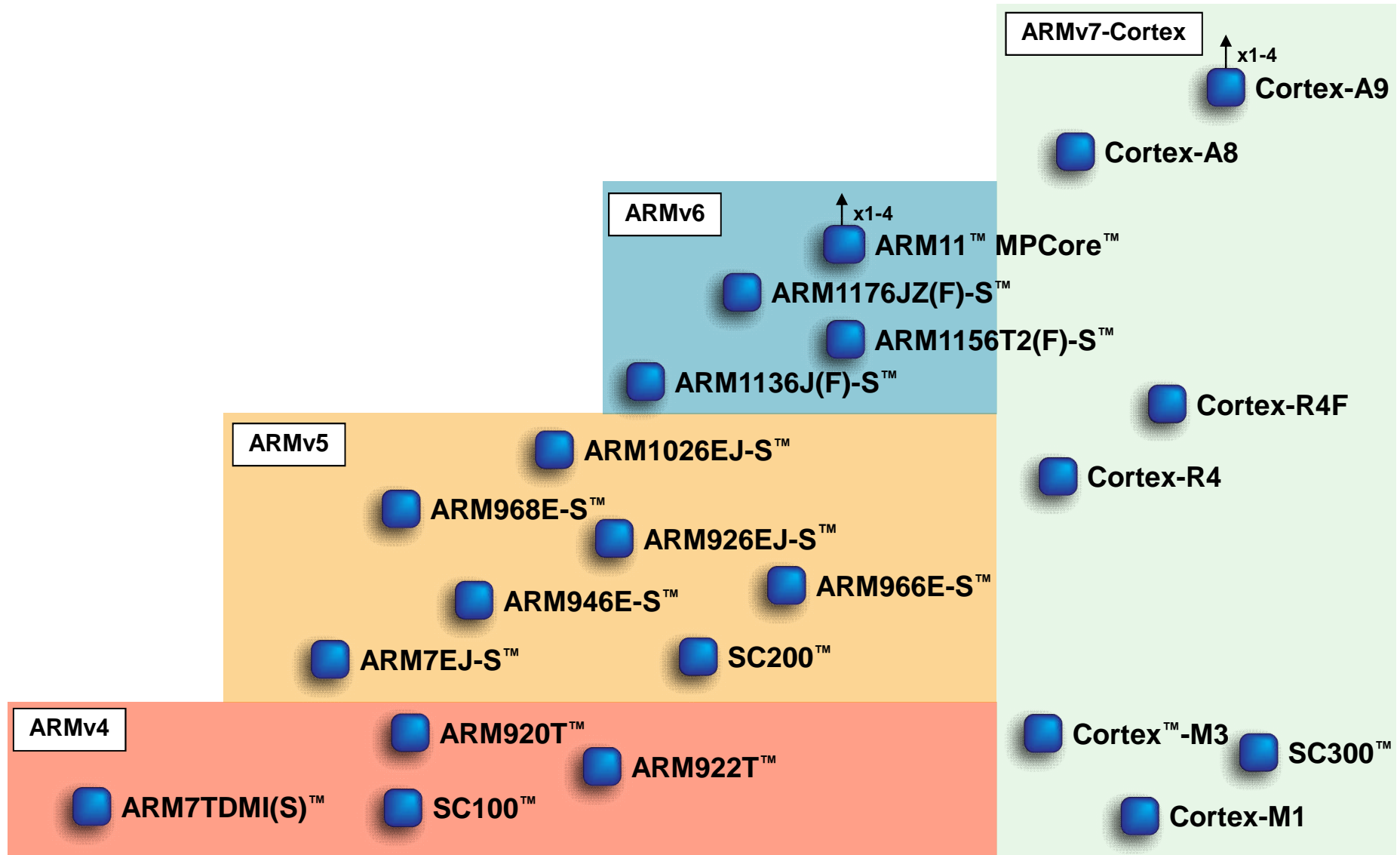
- **ARM Architecture/Programmers Model**

Data Path and Pipelines

AMBA

Development Tools

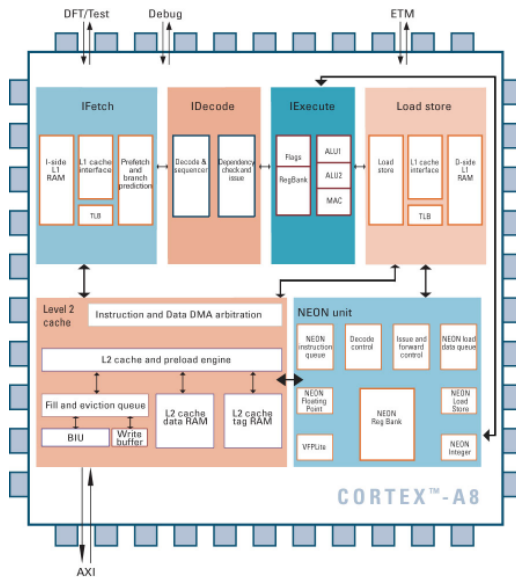
Architecture Versions



Cortex family

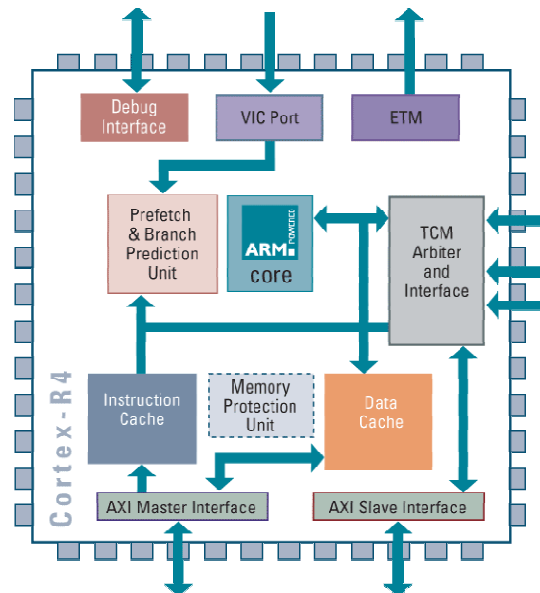
Cortex-A8

- Architecture v7A
- MMU
- AXI
- VFP & NEON support



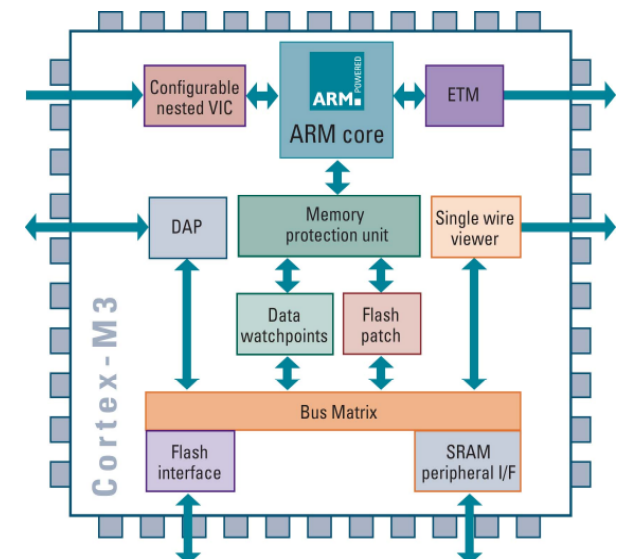
Cortex-R4

- Architecture v7R
- MPU (optional)
- AXI
- Dual Issue

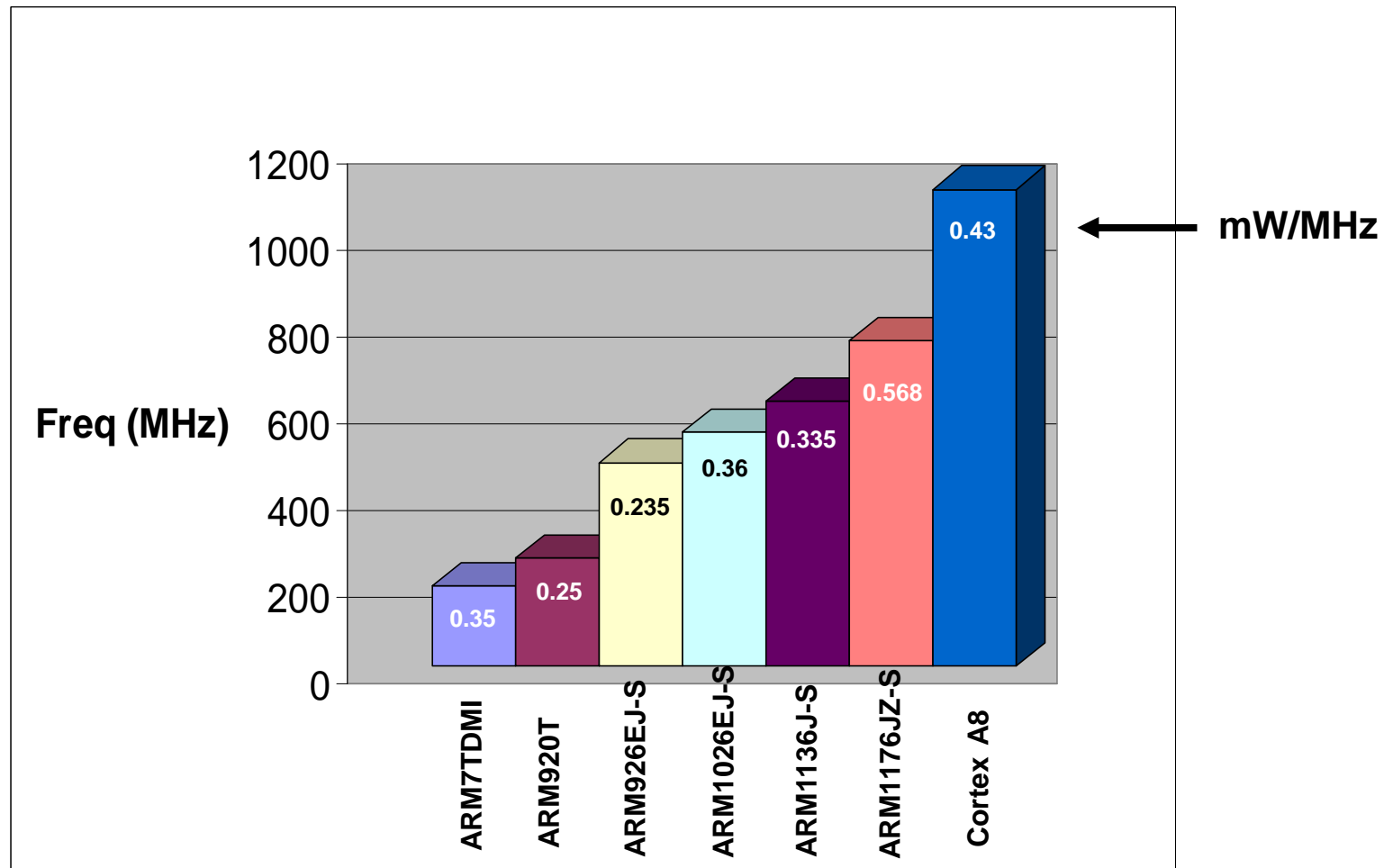


Cortex-M3

- Architecture v7M
- MPU (optional)
- AHB Lite & APB



Relative Performance*

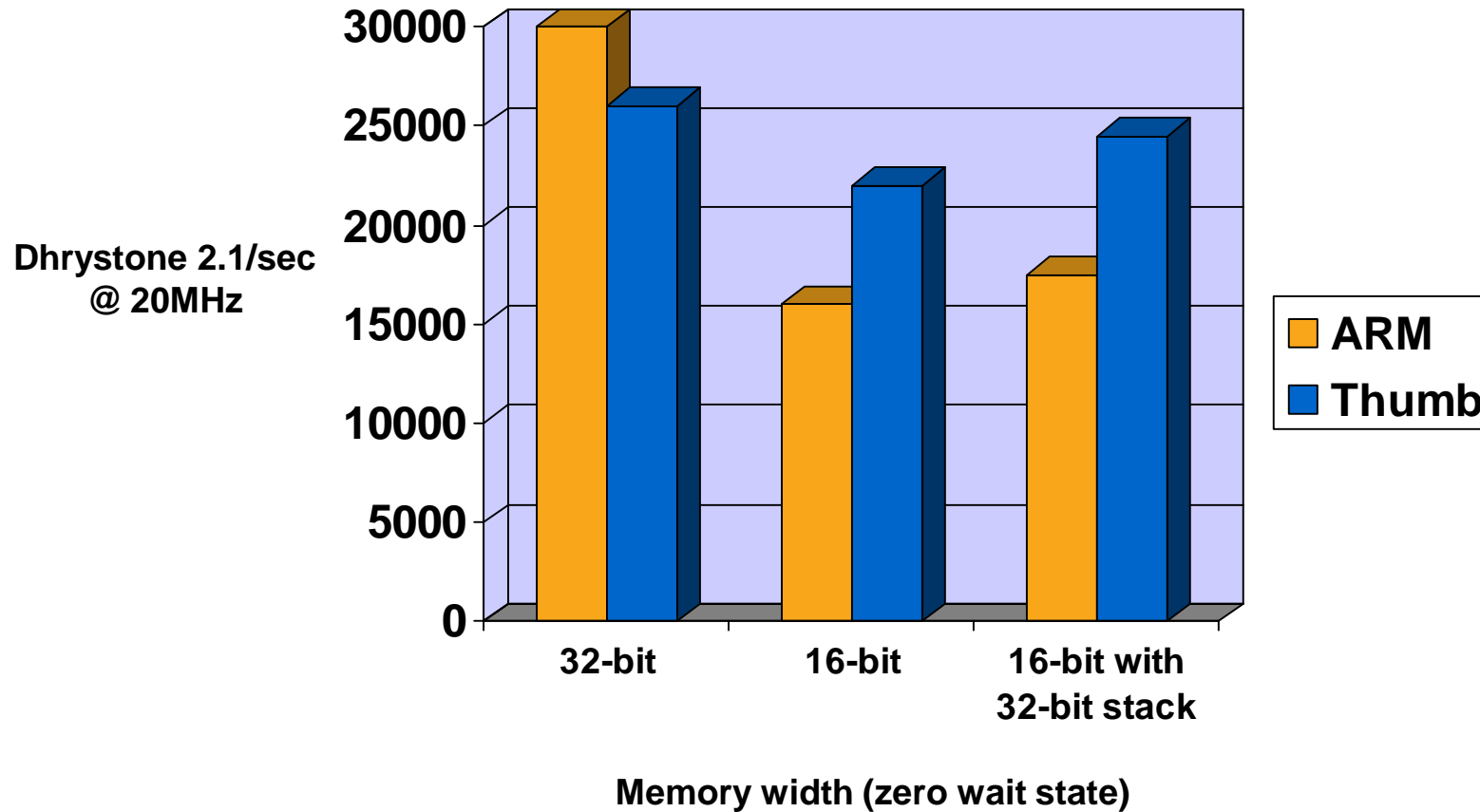


*Represents attainable speeds in 130, 90 or 65nm processes

Data Sizes and Instruction Sets

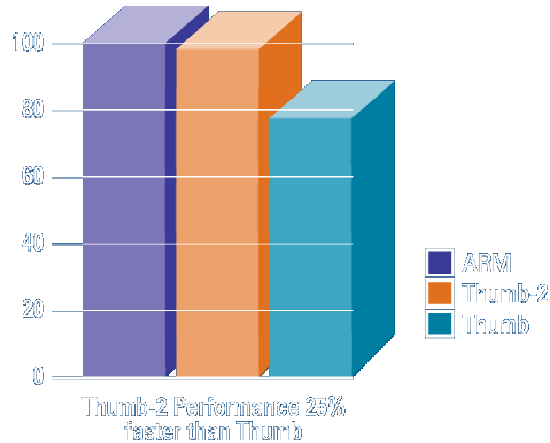
- The ARM is a 32-bit architecture.
- When used in relation to the ARM:
 - **Byte** means 8 bits
 - **Halfword** means 16 bits (two bytes)
 - **Word** means 32 bits (four bytes)
- Most ARM's implement two instruction sets
 - 32-bit ARM Instruction Set
 - 16-bit Thumb Instruction Set
- Jazelle cores can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

ARM and Thumb Performance



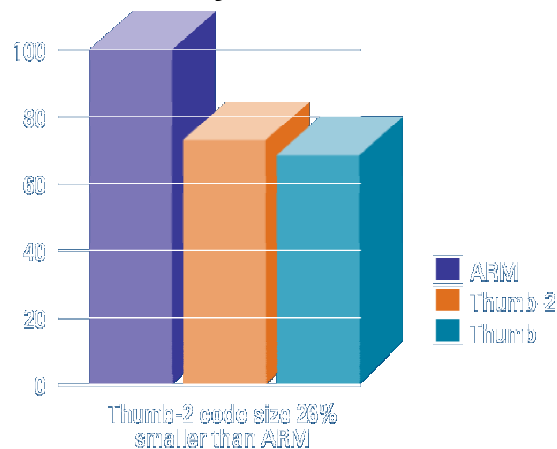
Thumb-2 Instruction Set

EEMBC Analysis - Performance



- 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

EEMBC Analysis – Code Size



Processor Modes

- 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

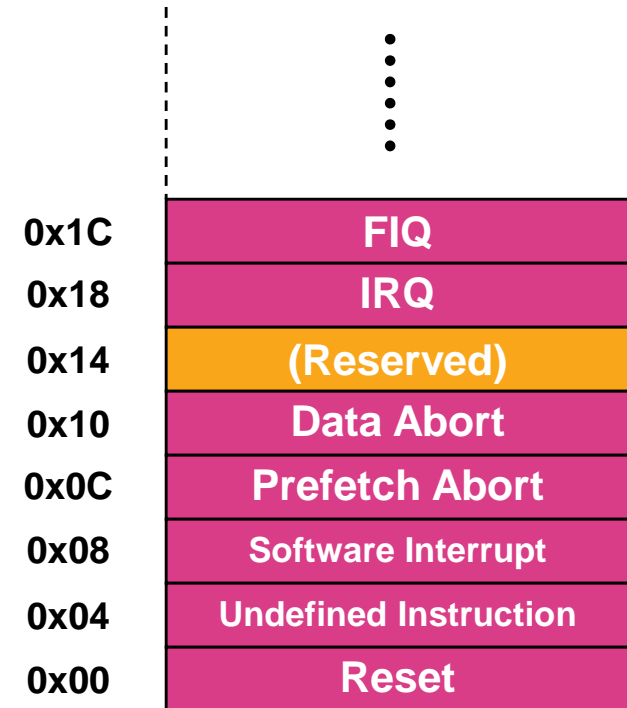
Banked out Registers

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 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>

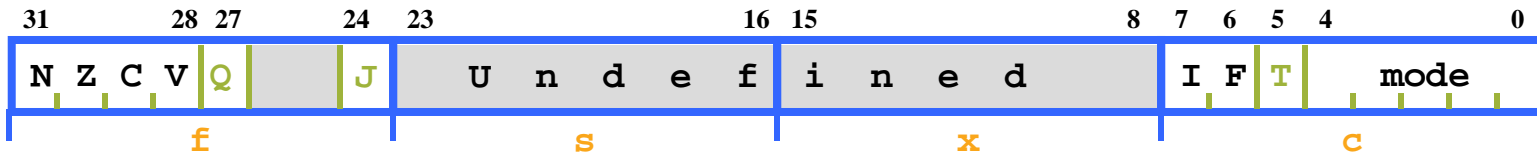
This can only be done in ARM state.



Vector Table

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

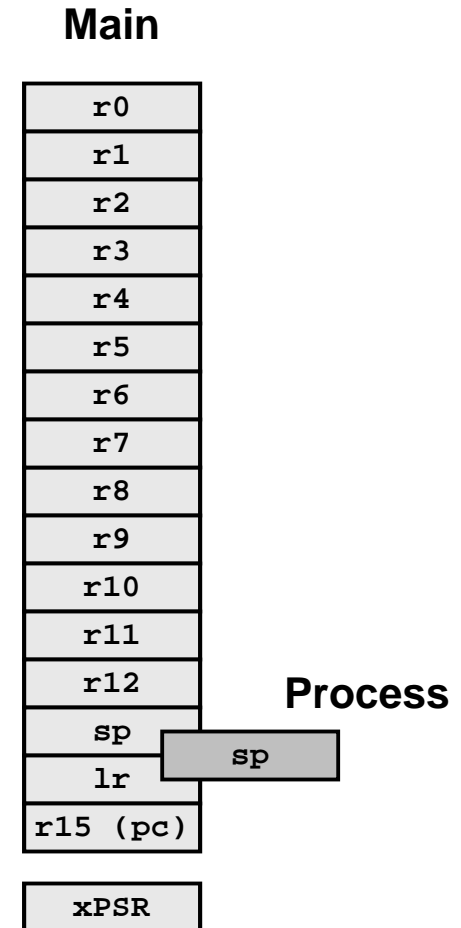
Program Status Registers



- 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**Verflowed
- **Sticky Overflow flag - Q flag**
 - Architecture 5TE/J only
 - Indicates if saturation has occurred
- **J bit**
 - Architecture 5TEJ only
 - J = 1: Processor in Jazelle state
- Interrupt Disable bits.
 - I = 1: Disables the IRQ.
 - F = 1: Disables the FIQ.
- **T Bit**
 - Architecture xT only
 - T = 0: Processor in ARM state
 - T = 1: Processor in Thumb state
- Mode bits
 - Specify the processor mode

Cortex-M3 Programmer's Model


- Fully programmable in C
- Stack-based exception model
- Only two processor modes
 - Thread Mode for User tasks
 - Handler Mode for OS tasks and exceptions
- Vector table contains addresses



Conditional Execution and Flags

- ARM instructions can be made to execute conditionally by postfixing them with the appropriate condition code field.
 - Increases code density
 - Improves 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
```

- By default, data processing instructions do not affect the condition code flags but the flags can be optionally set by using “S”. CMP does not need “S”.

loop

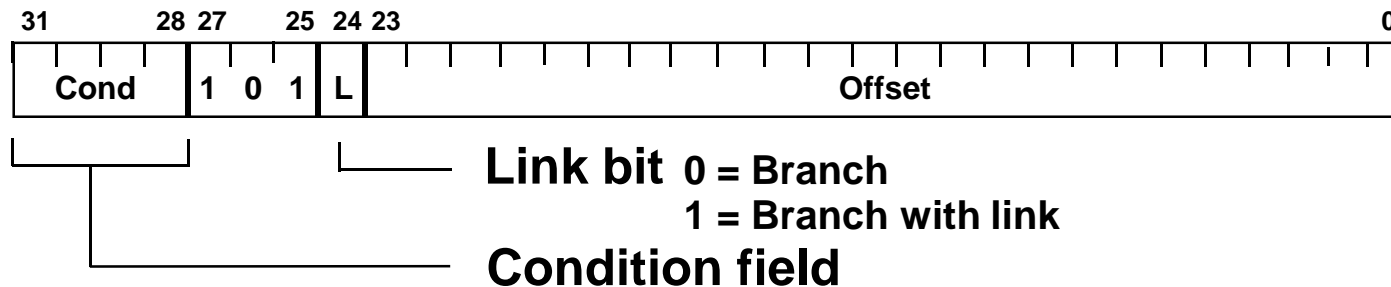
```
...
SUBS  r1,r1,#1
BNE  loop
```

decrement r1 and set flags

if Z flag clear then branch

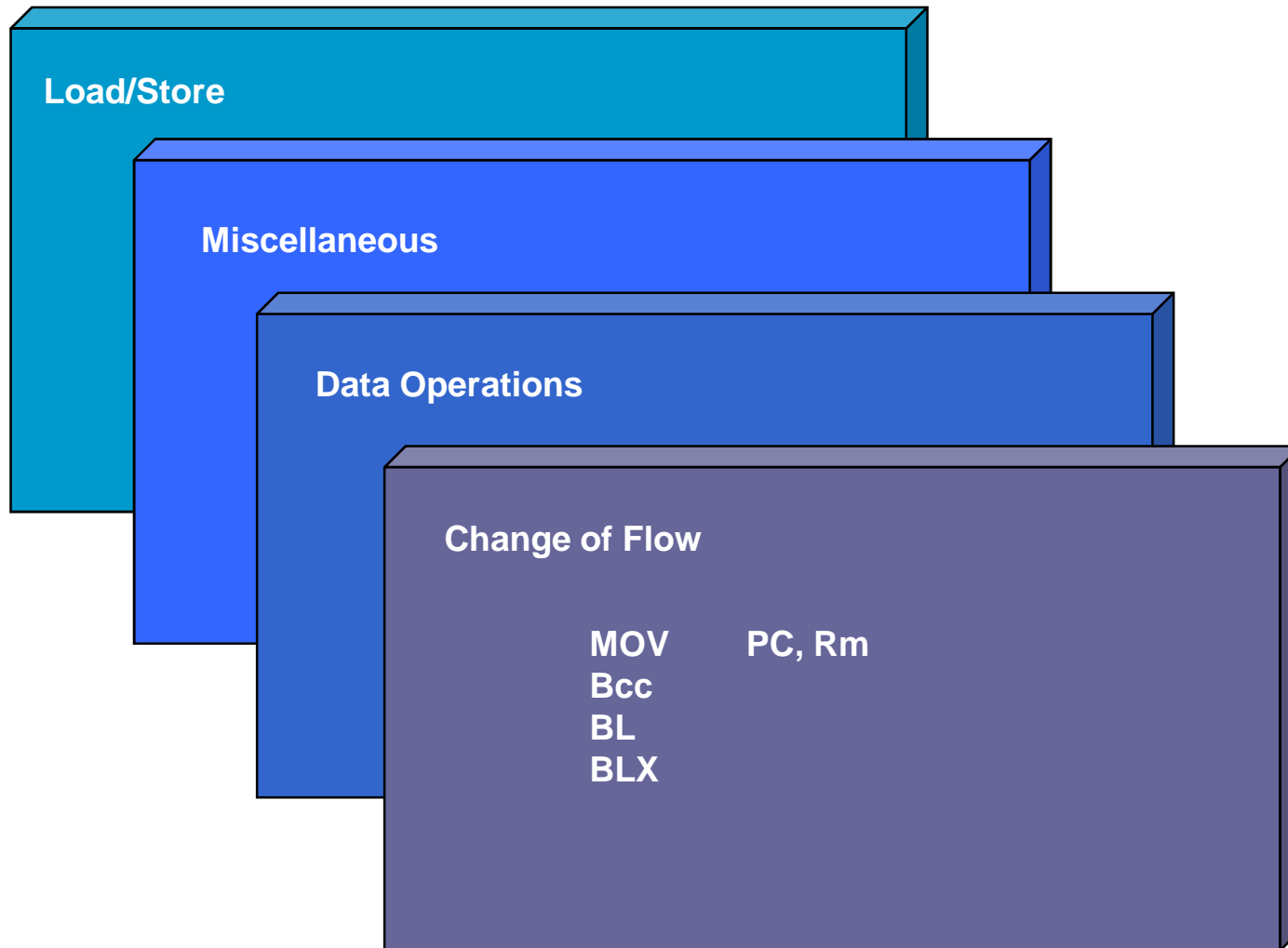
Branch instructions

- Branch : `B{<cond>} label`
- Branch with Link : `BL{<cond>} subroutine_label`



- The processor core shifts the offset field left by 2 positions, sign-extends it and adds it to the PC
 - ± 32 Mbyte range
 - How to perform longer branches?

Classes of Instructions (v4T)



Data processing Instructions

- Consist of :

- Arithmetic: **ADD** **ADC** **SUB** **SBC** **RSB** **RSC**
- Logical: **AND** **ORR** **EOR** **BIC**
- Comparisons: **CMP** **CMN** **TST** **TEQ**
- Data movement: **MOV** **MVN**

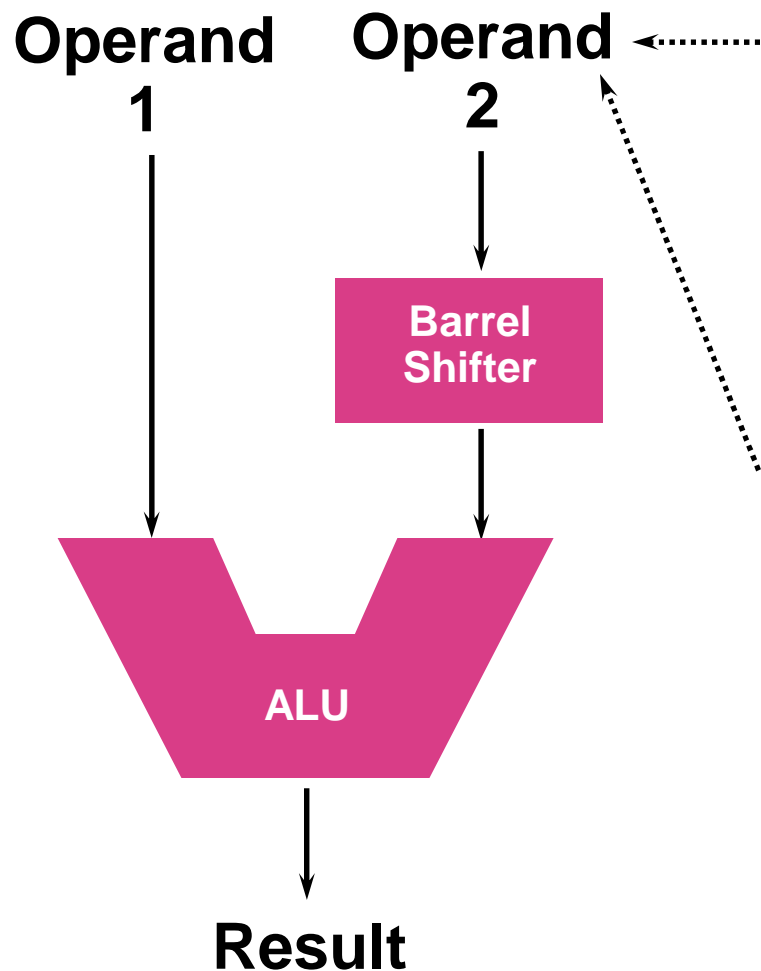
- These instructions only work on registers, NOT memory.

- Syntax:

<Operation>{<cond>}{S} Rd, Rn, Operand2

- Comparisons set flags only - they do not specify Rd
- Data movement does not specify Rn
- Second operand is sent to the ALU via barrel shifter.

Using a Barrel Shifter: The 2nd Operand



Register, optionally with shift operation

- Shift value can be either be:
 - 5 bit unsigned integer
 - Specified in bottom byte of another register.
- Used for multiplication by constant

Immediate value

- 8 bit number, with a range of 0-255. Rotated right through even number of positions
- Allows increased range of 32-bit constants to be loaded directly into registers

Single register data transfer

LDR	STR	Word
LDRB	STRB	Byte
LDRH	STRH	Halfword
LDRSB		Signed byte load
LDRSH		Signed halfword load

- Memory system must support all access sizes
- Syntax:
 - **LDR**{<cond>}{<size>} Rd, <address>
 - **STR**{<cond>}{<size>} Rd, <address>

e.g. **LDREQB**

Agenda

Introduction to ARM Ltd

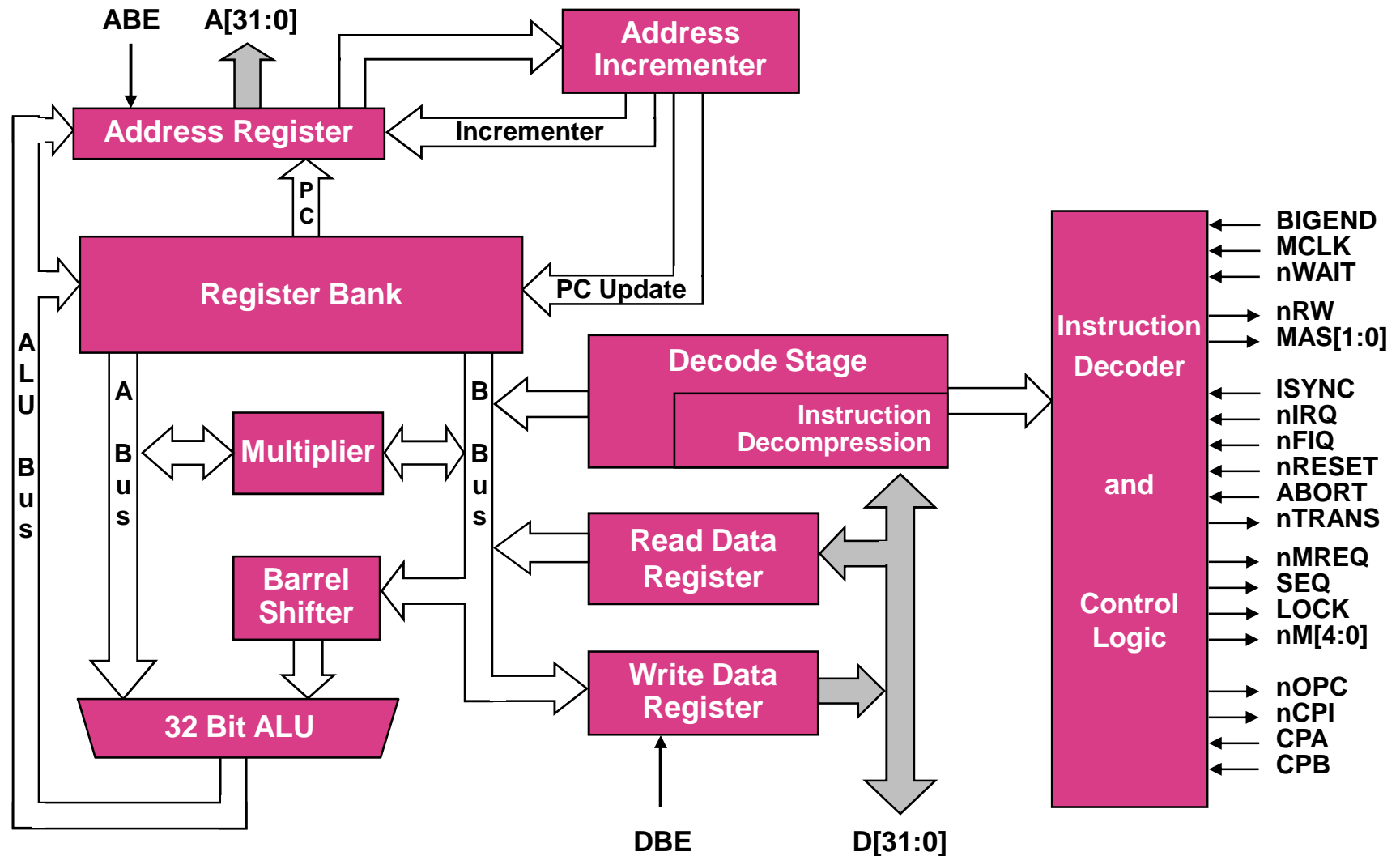
ARM Architecture/Programmers Model

- **Data Path and Pipelines**

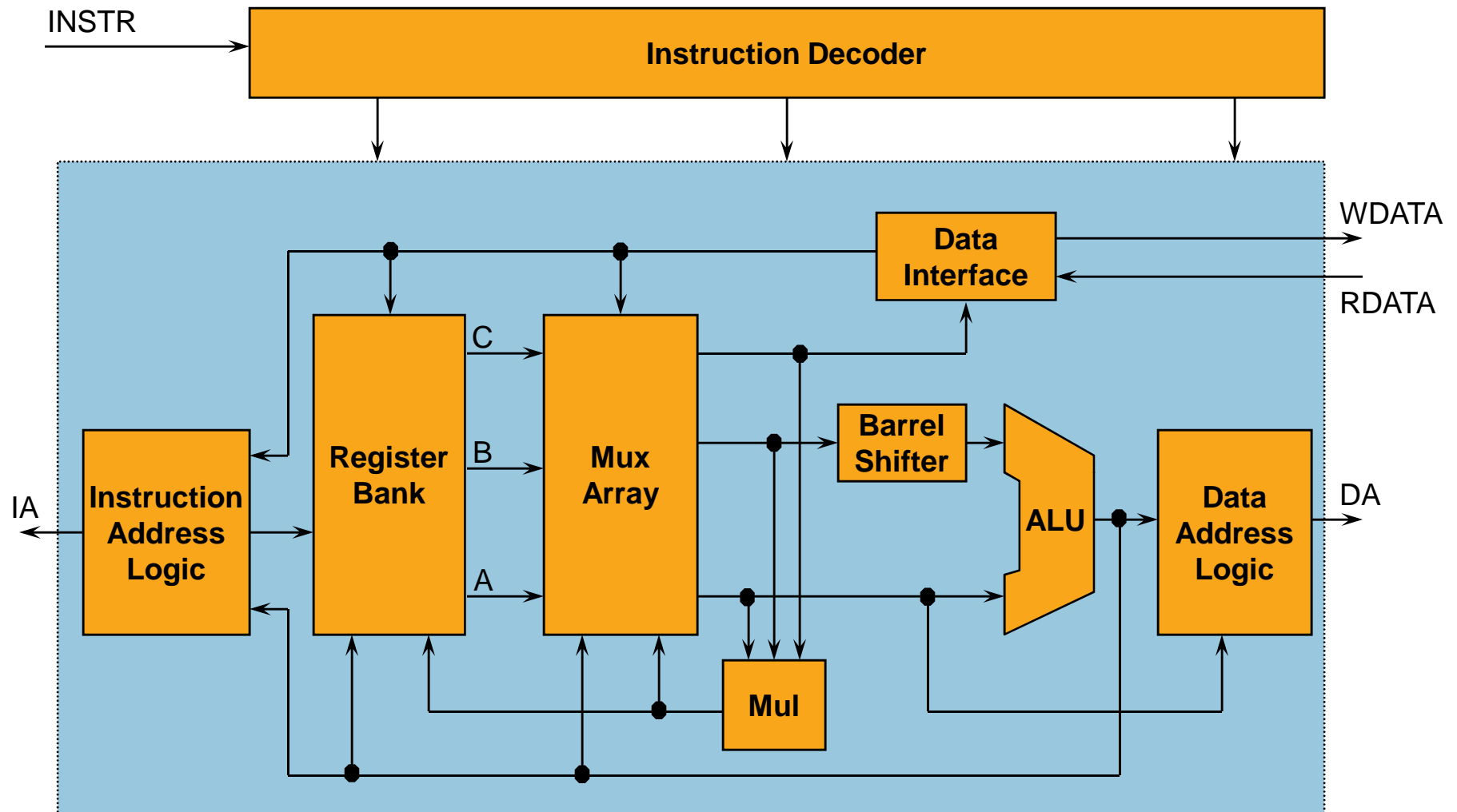
AMBA

Development Tools

The ARM7TDM Core

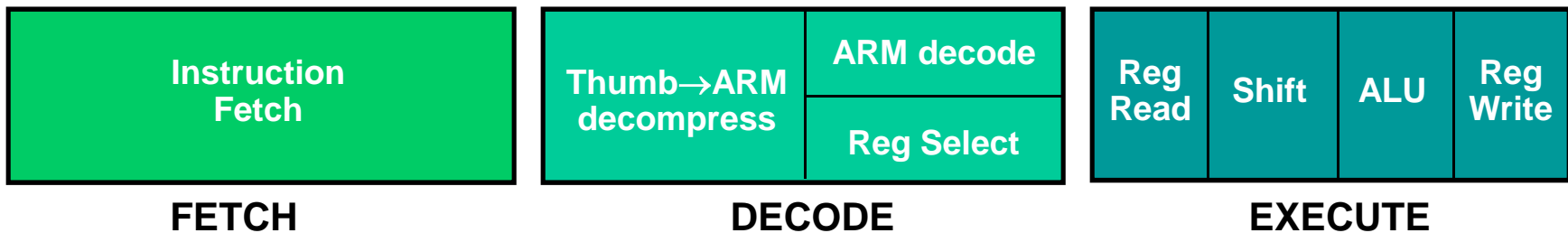


ARM9E-S Datapath

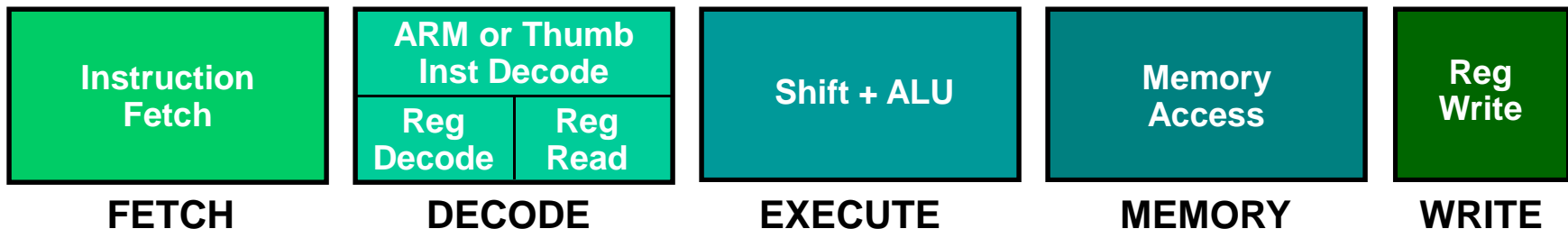


Pipeline changes for ARM9TDMI

ARM7TDMI

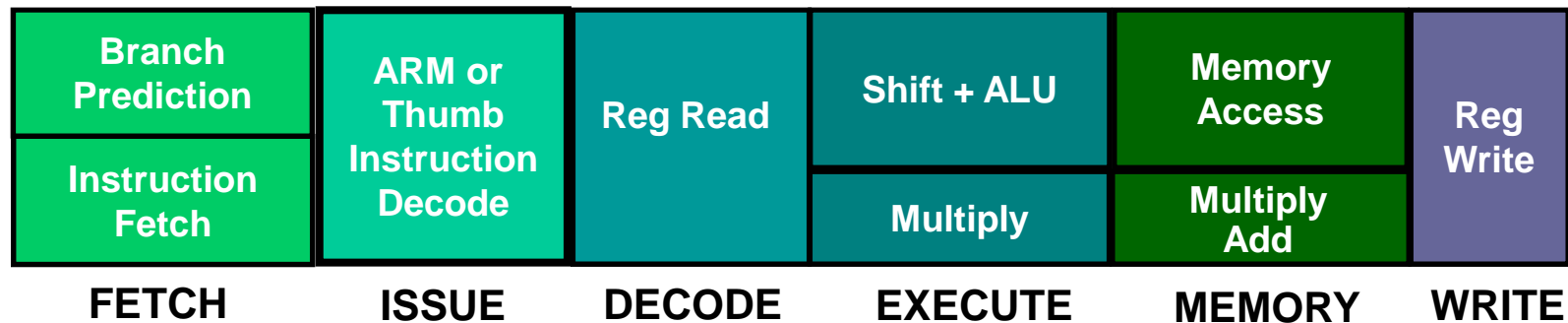


ARM9TDMI

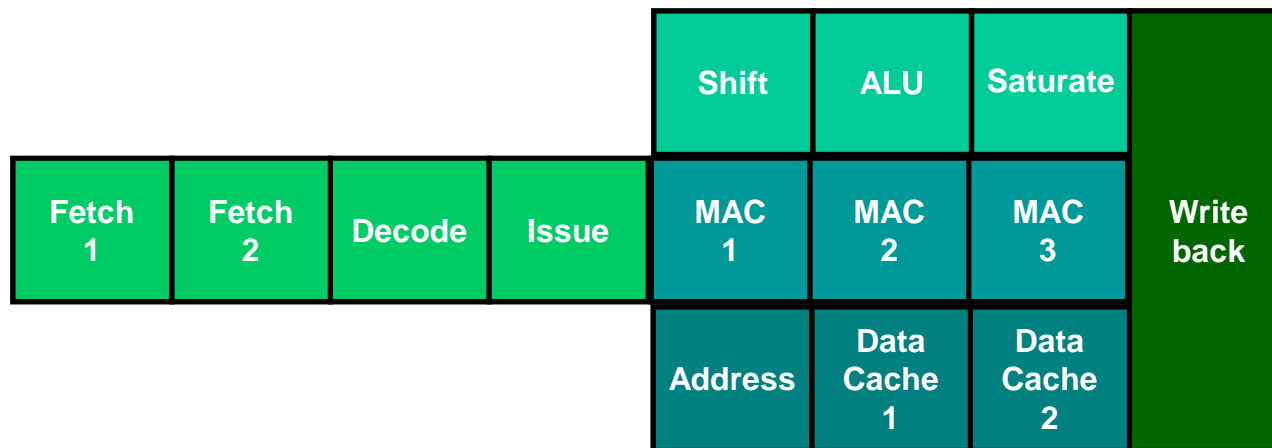


ARM10 vs. ARM11 Pipelines

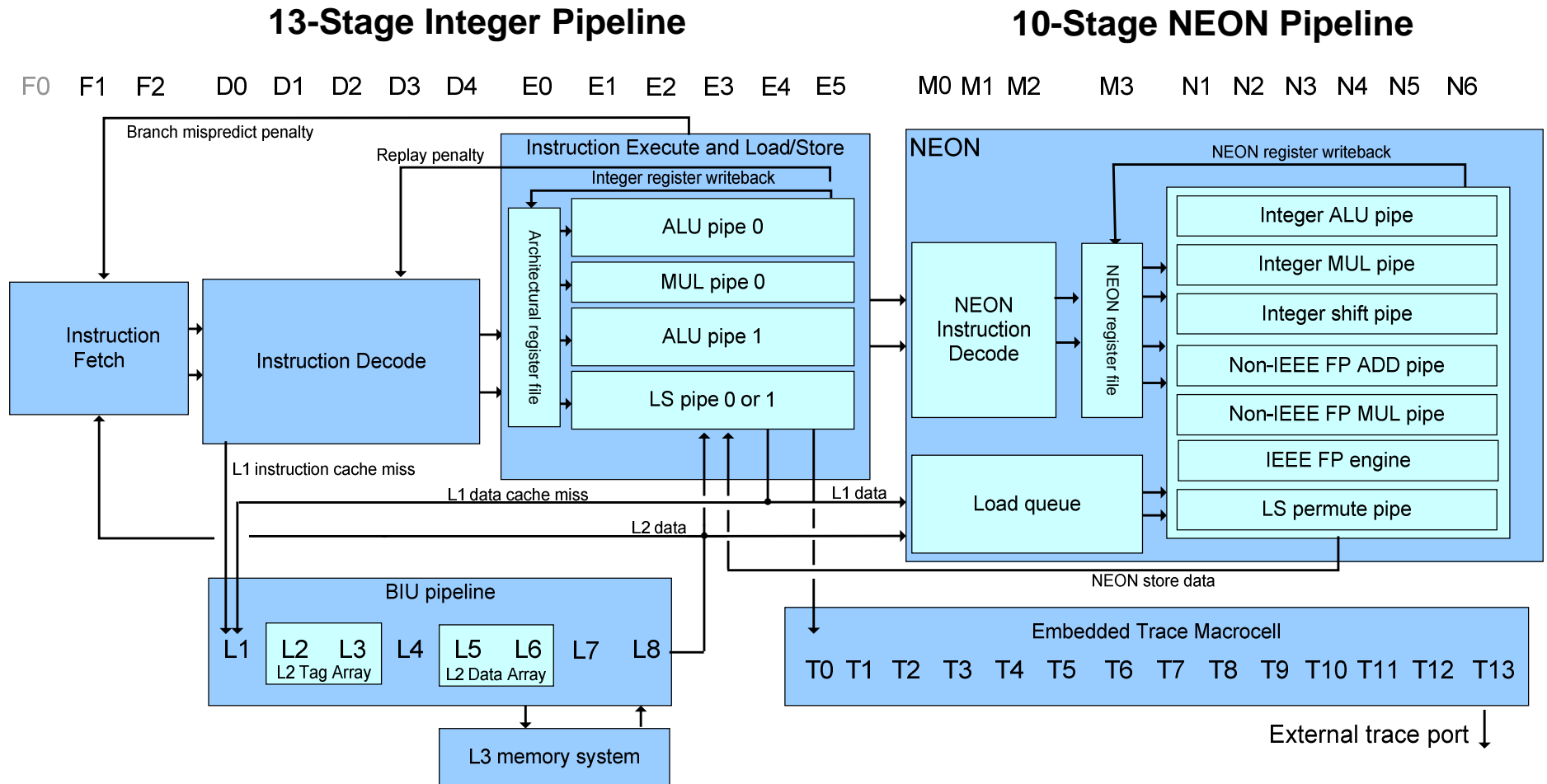
ARM10



ARM11



Full Cortex-A8 Pipeline Diagram



Agenda

Introduction to ARM Ltd

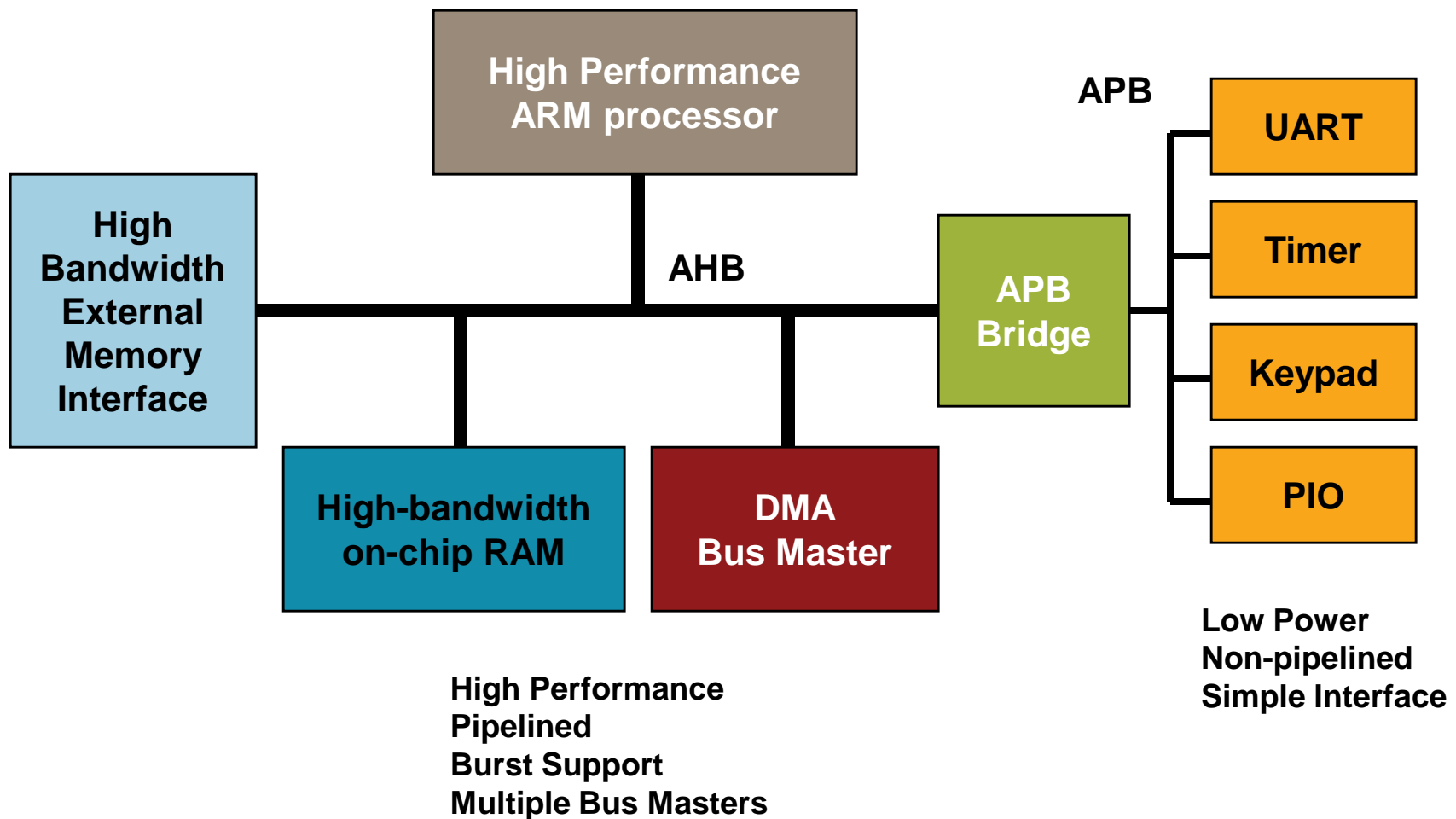
ARM Architecture/Programmers Model

Data Path and Pipelines

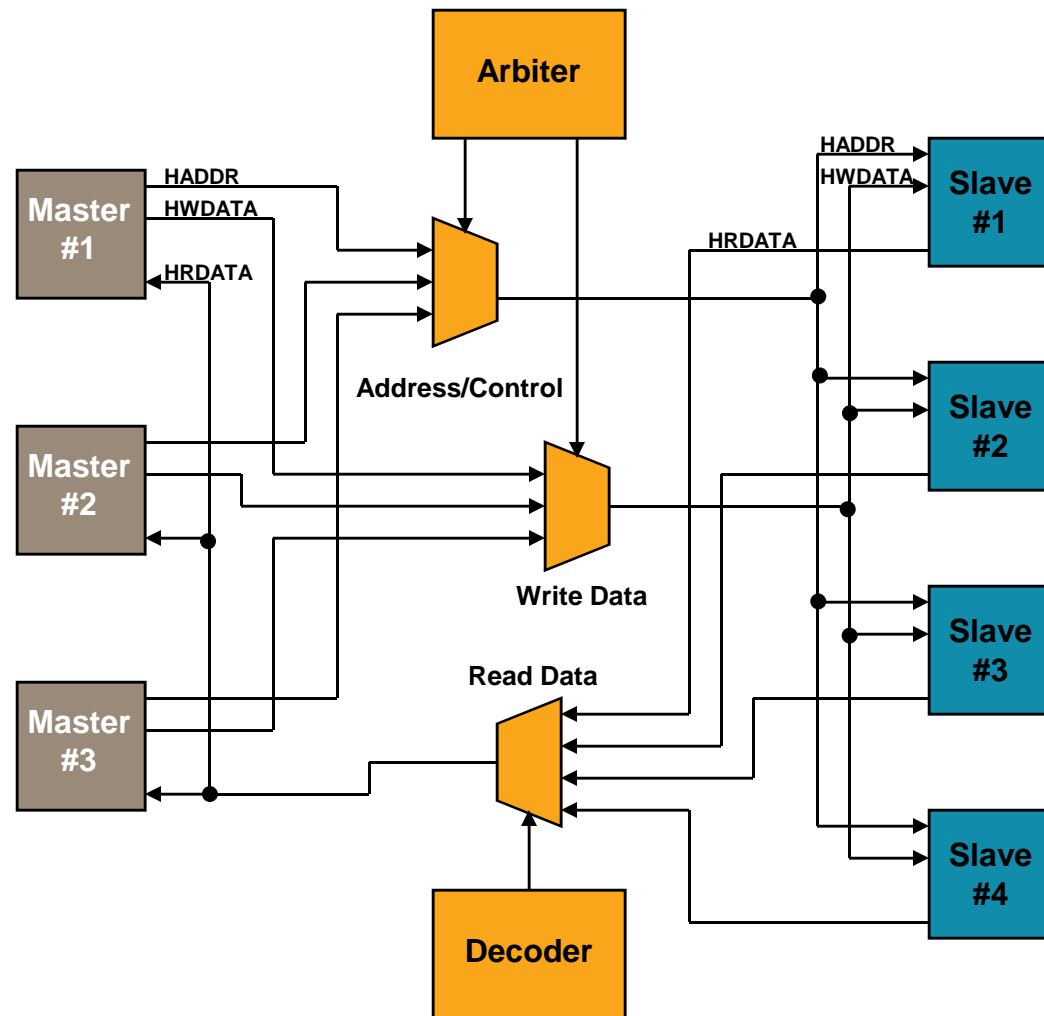
■ **AMBA**

Development Tools

An Example AMBA System



AHB Structure



Agenda

Introduction to ARM Ltd

ARM Architecture/Programmers Model

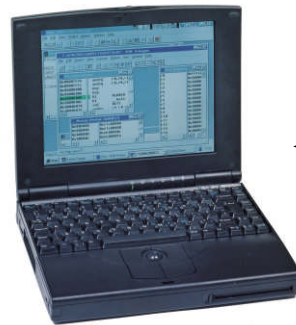
Data Path and Pipelines

AMBA

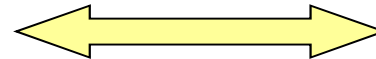
- **Development Tools**

ARM Debug Architecture

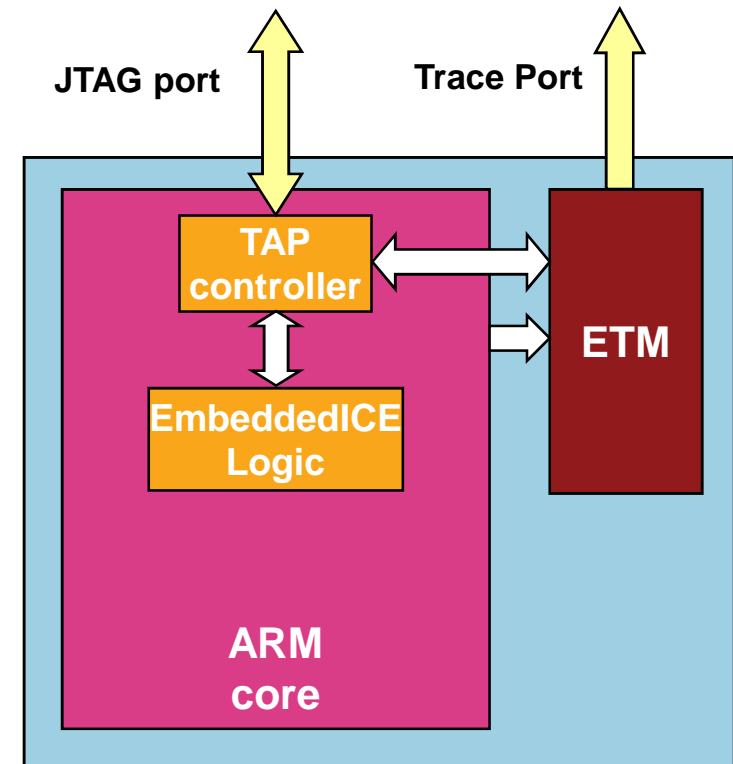
Debugger (+ optional trace tools)



Ethernet



- *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 (I²C, CAN, UART, SPI, Interrupts, I/O Ports, A/D and D/A converters, PWM, etc.)
- Evaluation Limitations
 - 32K byte object code + 32K 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

The screenshot displays the Keil uVision3 IDE interface for a project named 'Hello' targeting an LPC2100 microcontroller. The main window shows a C source file 'Hello.c' with the following code:

```
01 //*****  
02 /* This file is part of the uVision/ARM development tools */  
03 /* Copyright KEIL ELEKTRONIK GmbH 2002-2004 */  
04 //*****  
05 /*  
06 /* HELLO.C: Hello World Example  
07 /*  
08 //*****  
09  
10 #include <stdio.h> /* prototype declarations for I/O functions */  
11 #include <LPC21xx.H> /* LPC21xx definitions */  
12  
13  
14 //*****  
15 /* main program */  
16 //*****  
17 int main (void) ( /* execution starts here */  
18  
19 /* initialize the serial interface */  
20 PINSELO = 0x00050000; /* Enable RxD1 and TxD1 */  
21 U1LCR = 0x83; /* 8 bits, no Parity, 1 Stop bit */  
22 U1DLL = 97; /* 9600 Baud Rate @ 15MHz VPB Clock */  
23 U1LCR = 0x03; /* DLAB = 0 */  
24  
25 printf ("Hello World\n"); /* the 'printf' function call */  
26  
27 while (1) { /* An embedded program does not stop and */
```

The Project Workspace shows the Register table with the following values:

Register	Value
R0	0x0000000c
R1	0x0000000c
R2	0x00000020
R3	0x0000018d
R4	0x00000000
R5	0x00000000
R6	0x00000000
R7	0x00000000
R8	0x00000000
R9	0x00000000
R10	0x00000000

The Symbols table shows the following entries:

Name	Type
Simulator VTREG	
Peripheral SFR	
ALDOM	uchar
ALDOW	uchar
ALDOY	ushort
ALHOUR	uchar
ALMIN	uchar
ALMON	uchar
ALSEC	uchar

The Output window displays the following error message:

```
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
```

The Memory window shows the following data:

Address	0x4000
0x00004000:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004004:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004008:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000400C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004010:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004014:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004018:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000401C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004020:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004024:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004028:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000402C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004030:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004034:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004038:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000403C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004040:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004044:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004048:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000404C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004050:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004054:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004058:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000405C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004060:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004064:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004068:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000406C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004070:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004074:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004078:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000407C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004080:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004084:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004088:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000408C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004090:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004094:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004098:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000409C:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040A0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040A4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040A8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040AC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040B0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040B4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040B8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040BC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040C0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040C4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040C8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040CC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040D0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040D4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040D8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040DC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040E0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040E4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040E8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040EC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040F0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040F4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040F8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000040FC:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004100:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004104:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004108:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000410C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004110:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004114:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004118:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000411C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004120:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004124:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004128:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000412C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004130:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004134:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004138:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000413C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004140:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004144:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004148:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000414C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004150:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004154:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004158:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000415C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004160:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004164:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004168:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000416C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004170:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004174:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004178:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000417C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004180:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004184:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004188:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000418C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004190:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004194:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004198:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000419C:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041A0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041A4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041A8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041AC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041B0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041B4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041B8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041BC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041C0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041C4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041C8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041CC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041D0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041D4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041D8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041DC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041E0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041E4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041E8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041EC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041F0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041F4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041F8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000041FC:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004200:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004204:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004208:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000420C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004210:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004214:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004218:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000421C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004220:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004224:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004228:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000422C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004230:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004234:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004238:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000423C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004240:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004244:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004248:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000424C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004250:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004254:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004258:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000425C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004260:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004264:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004268:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000426C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004270:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004274:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004278:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000427C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004280:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004284:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004288:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000428C:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004290:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004294:	00 00 00 00 00 00 00 00 00 00 00 00
0x00004298:	00 00 00 00 00 00 00 00 00 00 00 00
0x0000429C:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042A0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042A4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042A8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042AC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042B0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042B4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042B8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042BC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042C0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042C4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042C8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042CC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042D0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042D4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042D8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042DC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042E0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042E4:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042E8:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042EC:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042F0:	00 00 00 00 00 00 00 00 00 00 00 00
0x000042F4:	00 00 00 00 0

TI's Beagle Board

> 1000 participants
and growing

Active &
technical
community

Open access to
hardware
documentation

Opportunity
to tinker and
learn

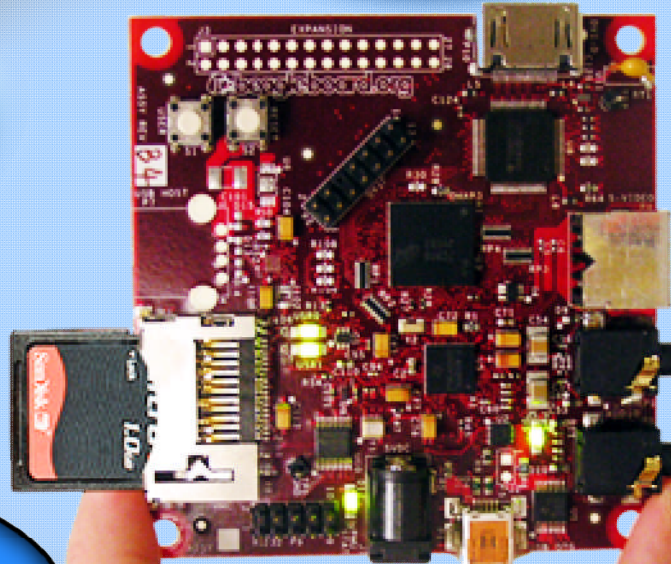
Personally
affordable

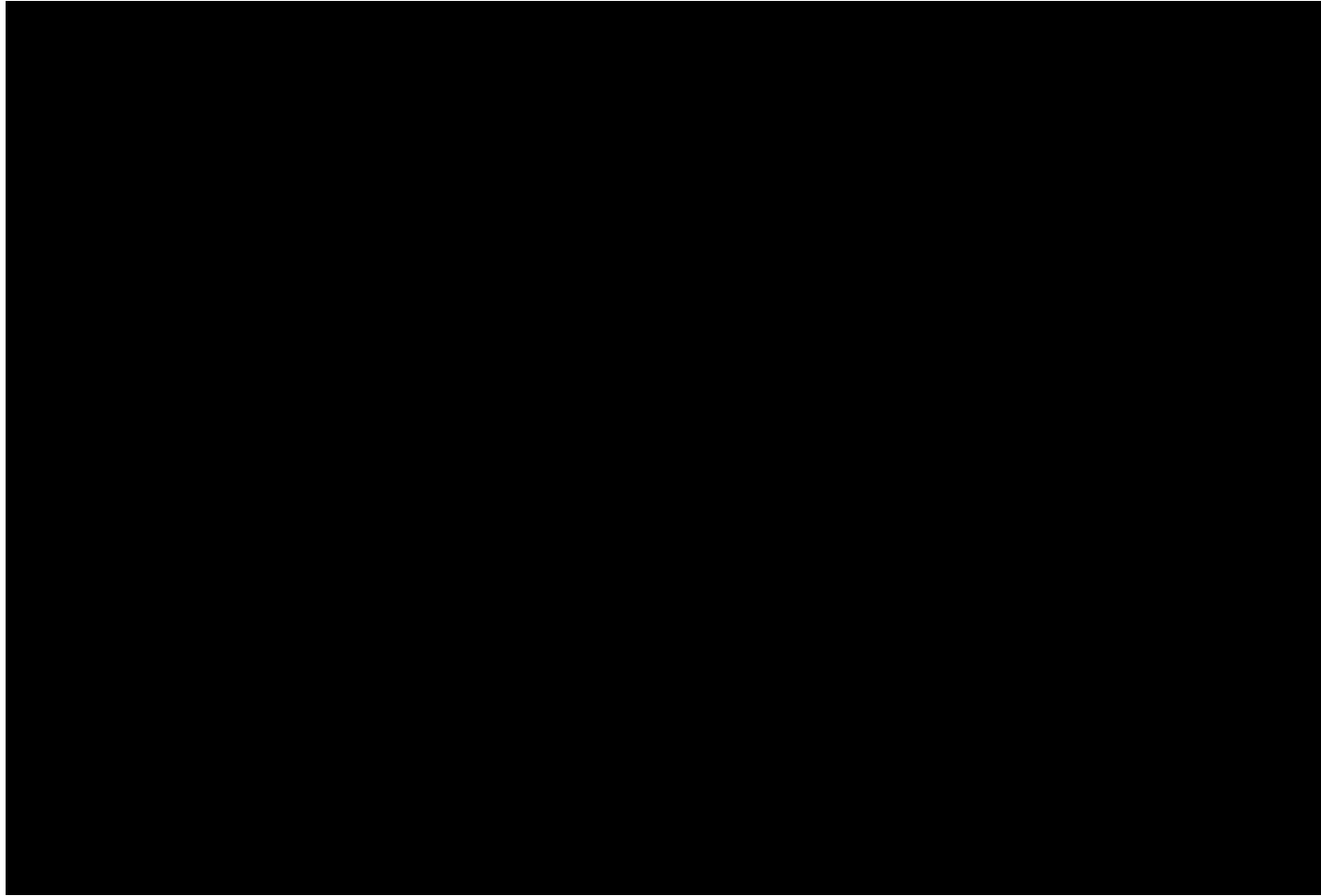
Wikis, blogs,
promotion of
community
activity

Freedom to
innovate

Instant access to
>10 million lines
of code

Free
software





Questions:
university@arm.com

More information:
www.arm.com/community/university



Fin



The Architecture for the Digital World®

ARM®