

ARM Cortex-M3 Introduction

ARM University Relations



Agenda

- Cortex-M3 Overview

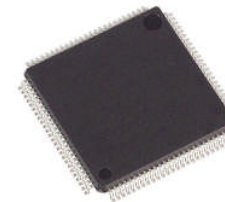
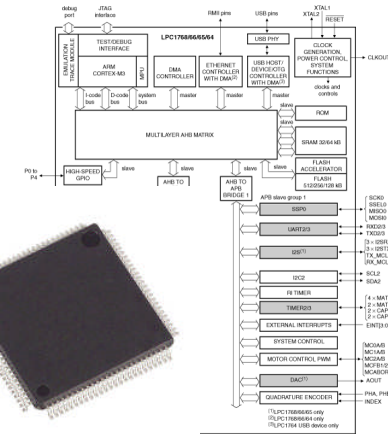
v7-M Architecture/Programmers Model

Data Path and Pipelines

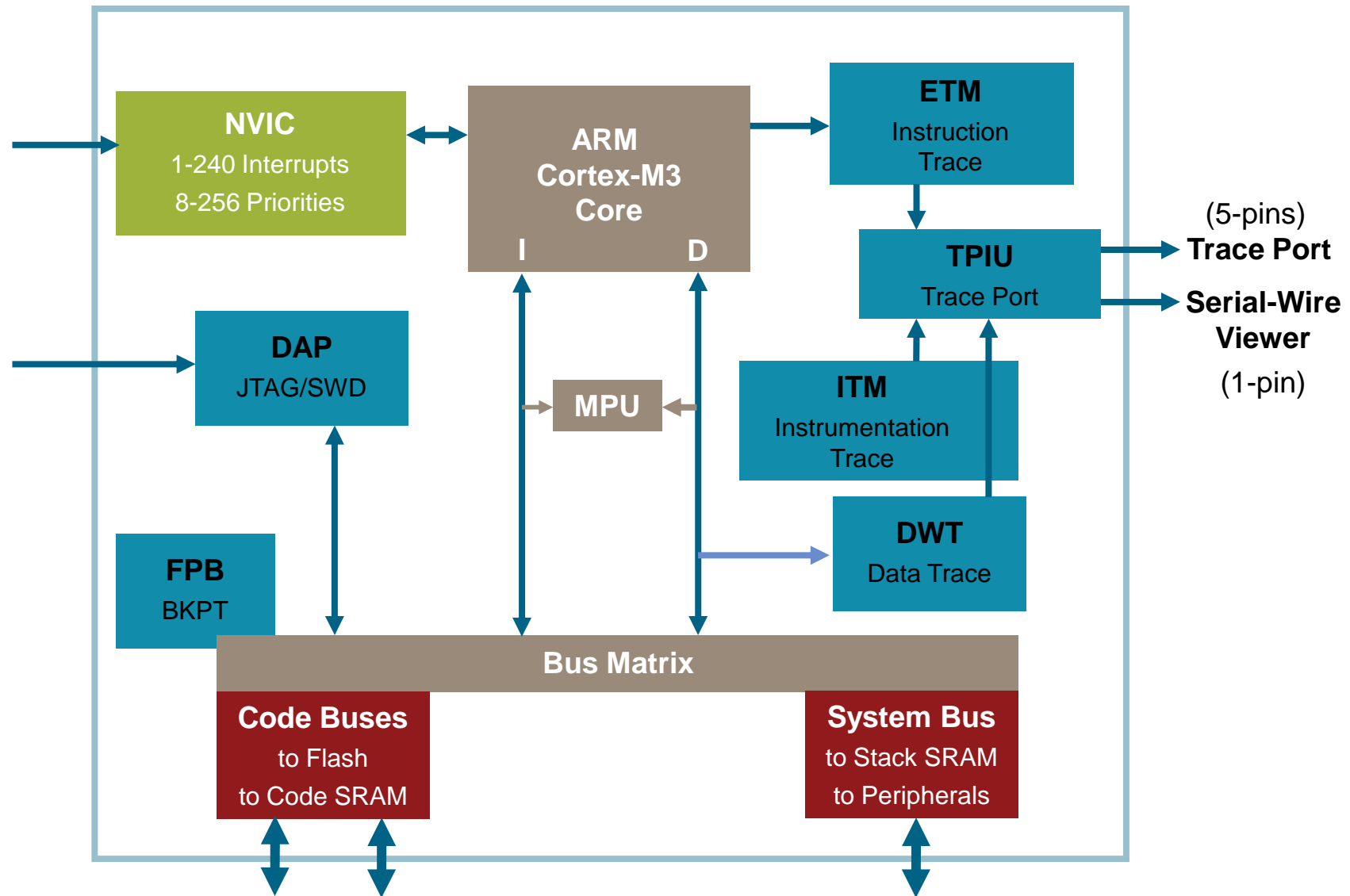
Tools and mbed Platform

What's Happening in Microcontrollers?

- Microcontrollers are getting **cheap**
 - 32-bit ARM Cortex-M3 Microcontrollers @ \$1
 - Some microcontrollers sell for as little as \$0.65
- Microcontrollers are getting **powerful**
 - Lots of processing, memory, I/O in one package
 - Floating-point is even available in some!
- Microcontrollers are getting **interactive**
 - Internet connectivity, new sensors and actuators
 - LCD and display controllers are common
- Creates new opportunities for microcontrollers



ARM Cortex-M3 Processor



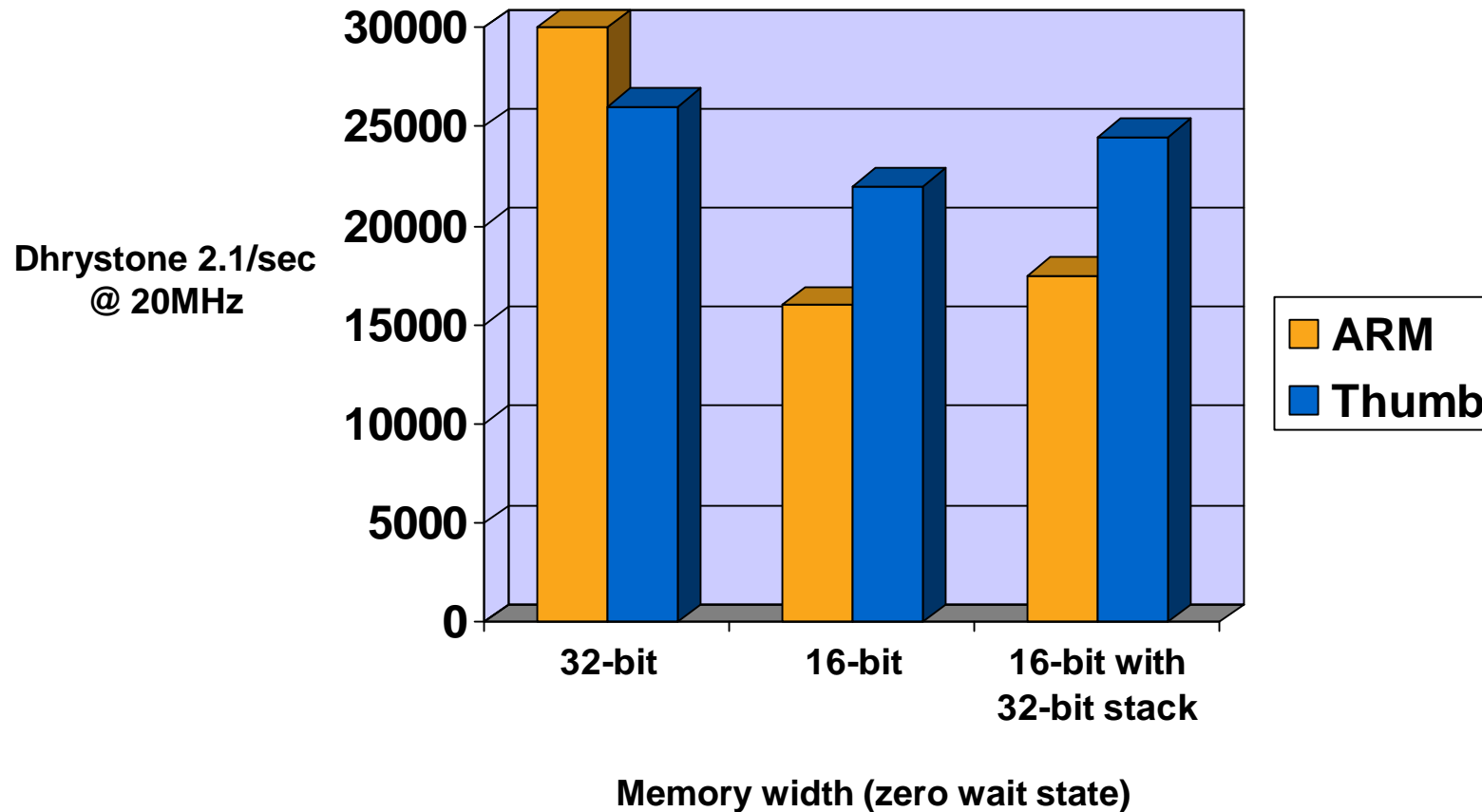
ARM Cortex-M3 Microcontroller

- 18 x 32-bit registers
- Excellent compiler target
- Reduced pin count requirements
- Efficient interrupt handling
- Power management
- Efficient debug and development support features
 - Breakpoints, Watchpoints,
 - Flash Patch support,
 - Instruction Trace
- Strong OS support
 - User/Supervisor model
 - OS support features
- Designed to be fully programmed in C (even reset, interrupts and exceptions)

ARM Cortex-M3 Microcontroller

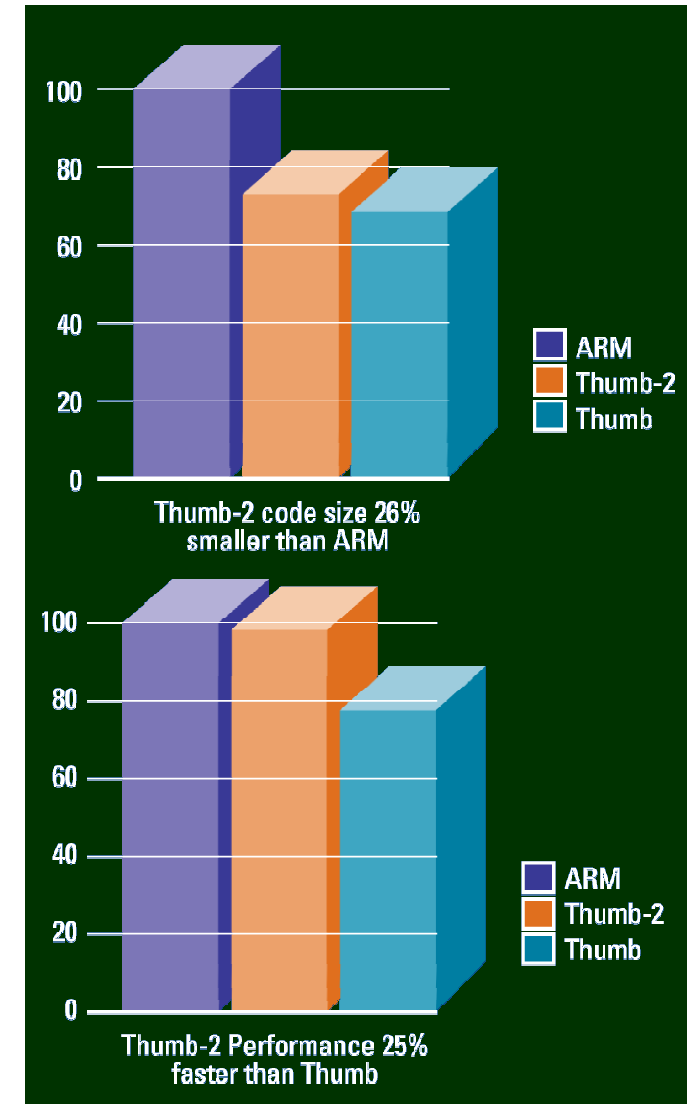
- ARMv7M Architecture
- No Cache - No MMU
- Debug is optimized for microcontroller applications
- Vector table contains addresses, not instructions
- DIV instruction
- Interrupts automatically save/restore state
- Exceptions programmed in C (No Coprocessor 15 - All registers are memory-mapped)
- Interrupt controller is part of Cortex-M3 macrocell
- Fixed memory map
- Bit-banding
- Non-Maskable Interrupt (NMI)
- Only one processor status reg
- Thumb-2 processing core
 - Mix of 16 and 32 bit instructions for very high code density
 - Gives complete Thumb compatibility

ARM and Thumb Performance



The Thumb-2 instruction set

- Variable-length instructions
 - ARM instructions are a fixed length of 32 bits
 - Thumb instructions are a fixed length of 16 bits
 - Thumb-2 instructions can be either 16-bit or 32-bit
- Thumb-2 gives approximately 26% improvement in code density over ARM
- Thumb-2 gives approximately 25% improvement in performance over Thumb



Agenda

Cortex-M3 Overview

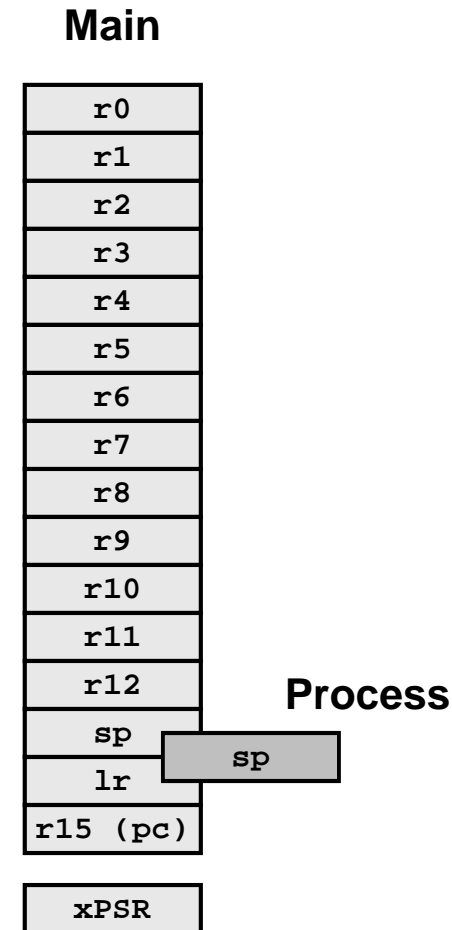
- v7-M Architecture/Programmers Model

Data Path and Pipelines

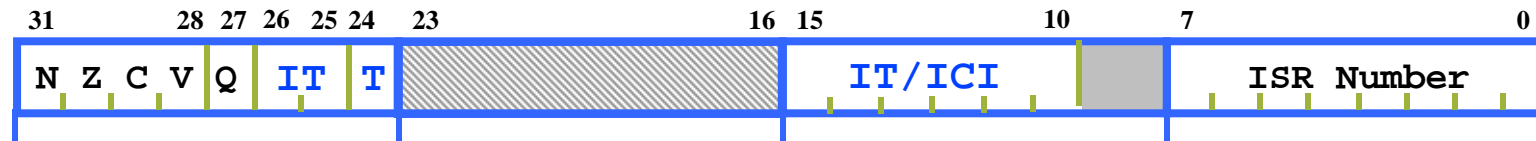
Tools and mbed Platform

Cortex-M3 Register Set

- Very compiler friendly
 - Load/Store Architecture
 - 32-bit registers
 - Flexible register scheme
 - Linear 32-bit address space

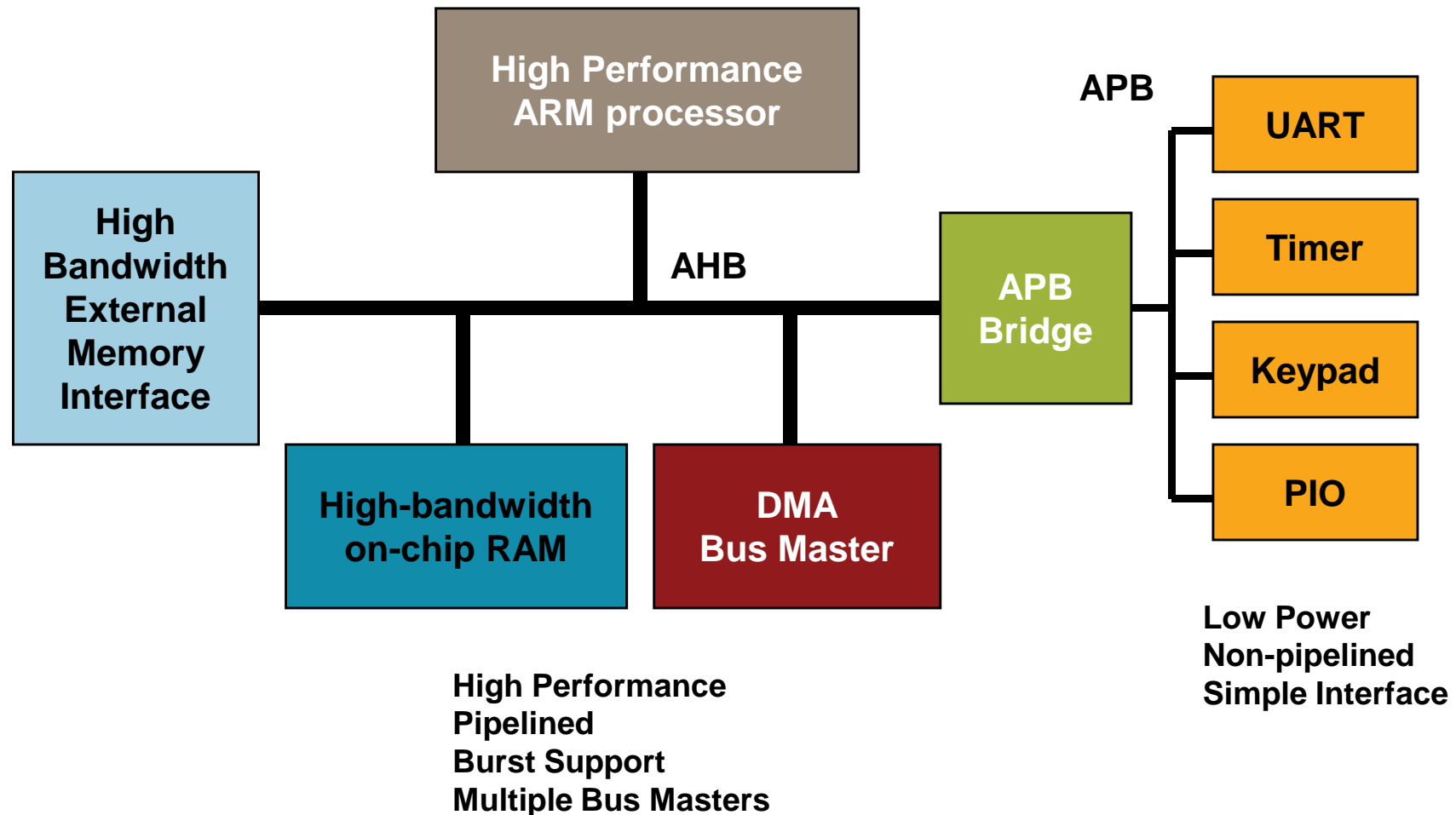


Program Status Register



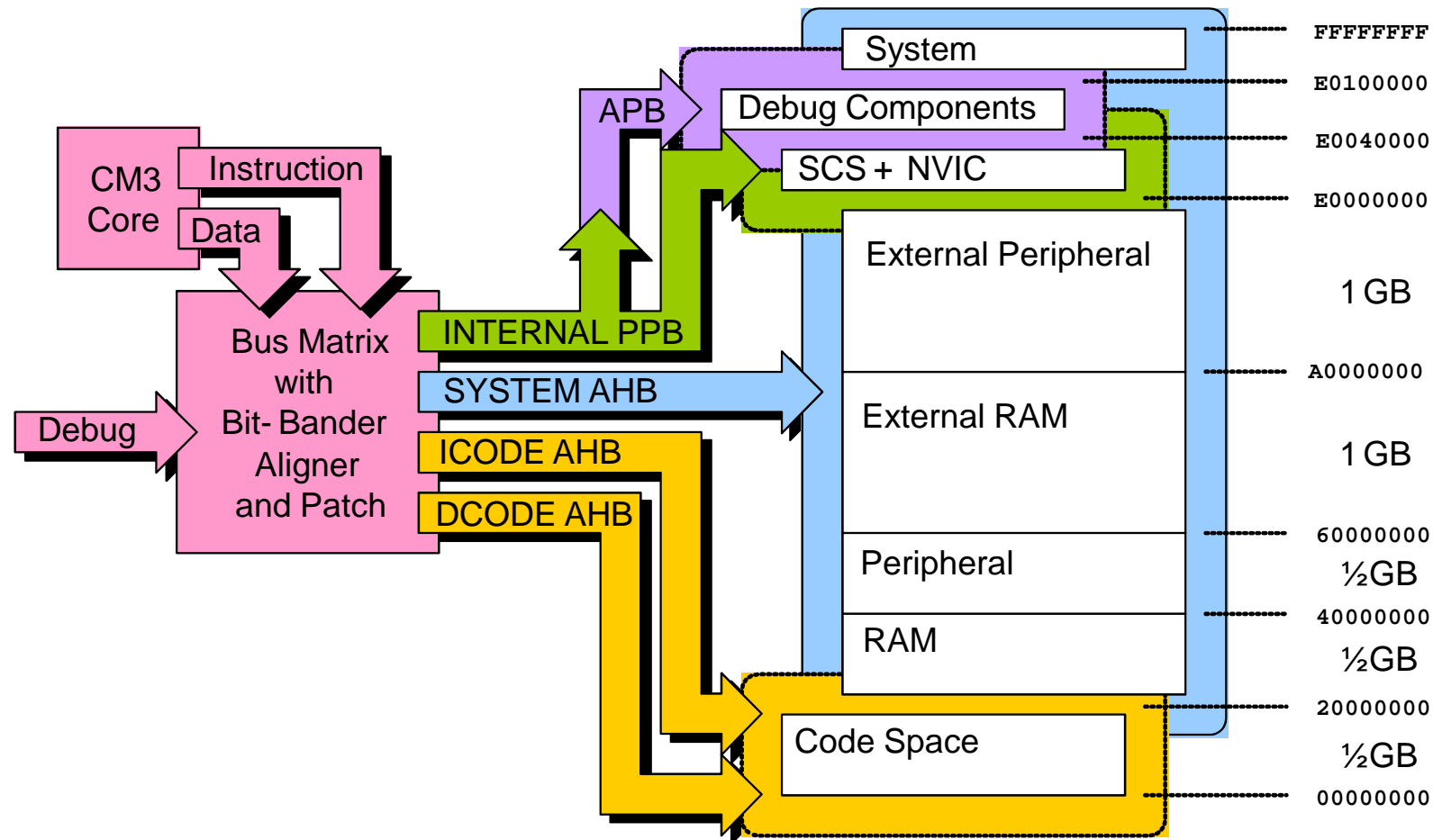
- One Status Register consisting of
 - APSR - Application Program Status Register – ALU flags
 - IPSR - Interrupt Program Status Register – Interrupt/Exception No.
 - EPSR - Execution Program Status Register
 - IT field – If/Then block information
 - ICI field – Interruptible-Continuable Instruction information
- xPSR
 - Composite of the 3 PSRs
 - Stored on the stack on exception entry

An Example AMBA System

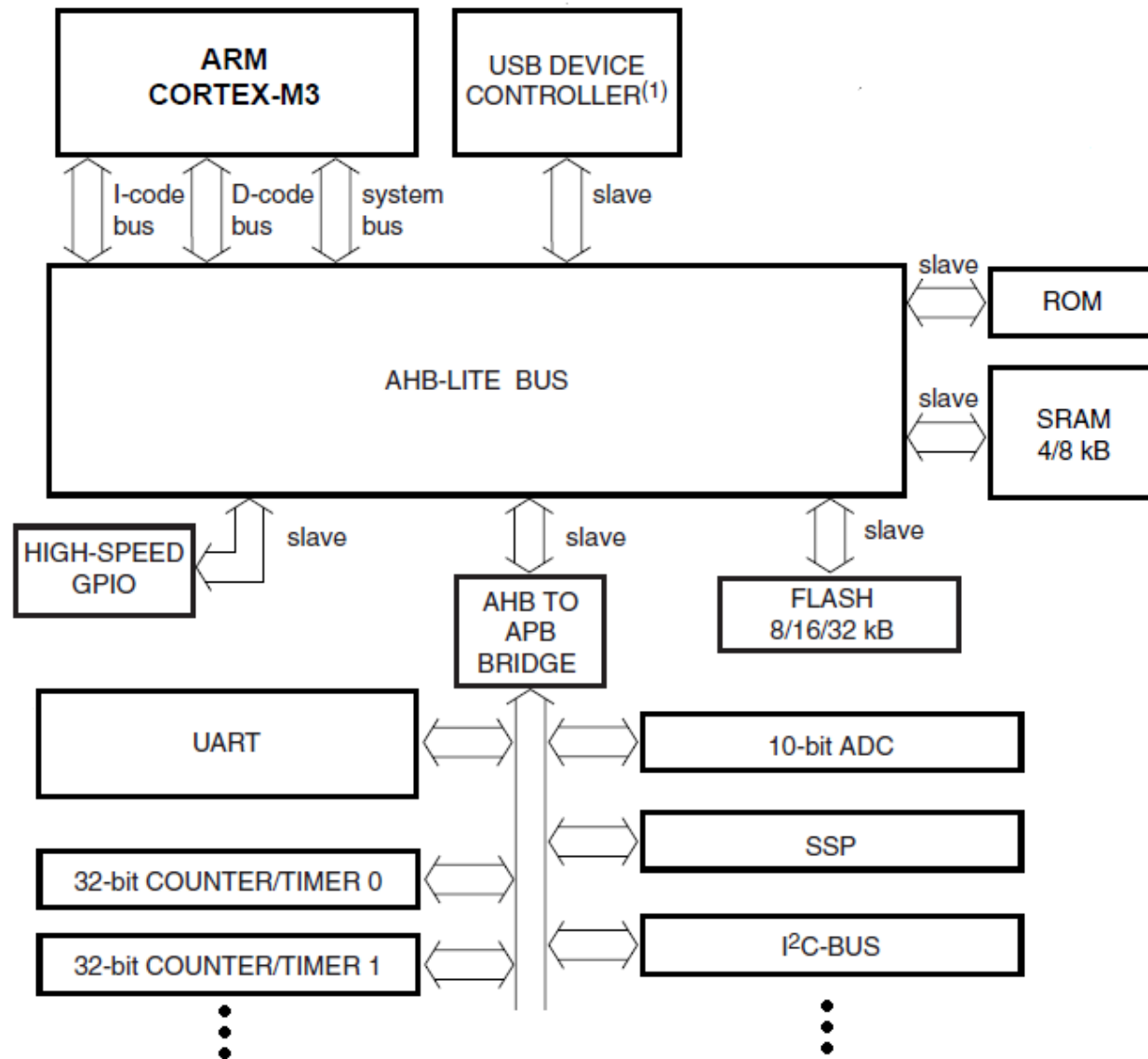


Memory Map

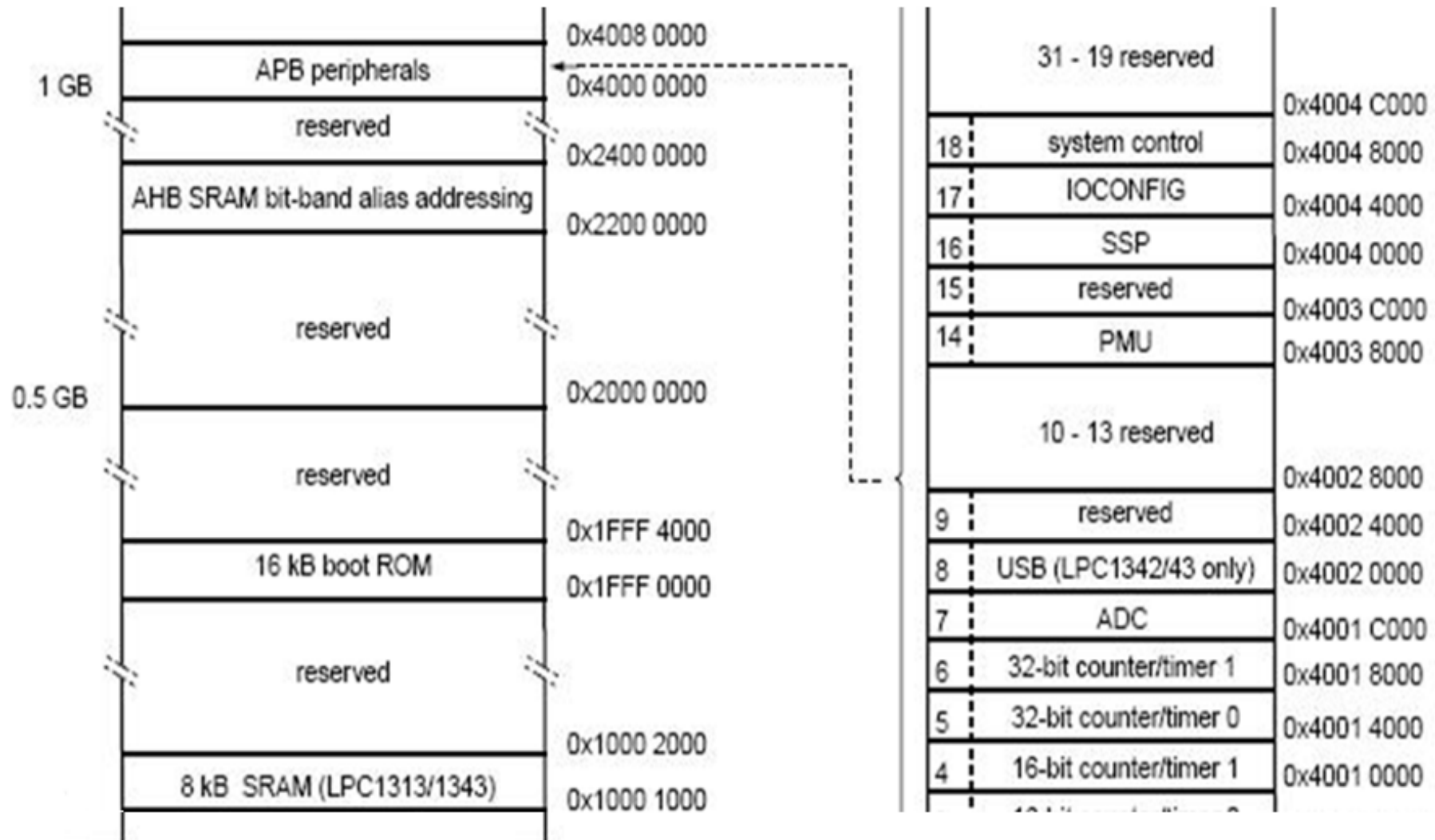
- Very simple linear 4GB memory map
- The Bus Matrix partitions memory access via the AHB and PPB buses



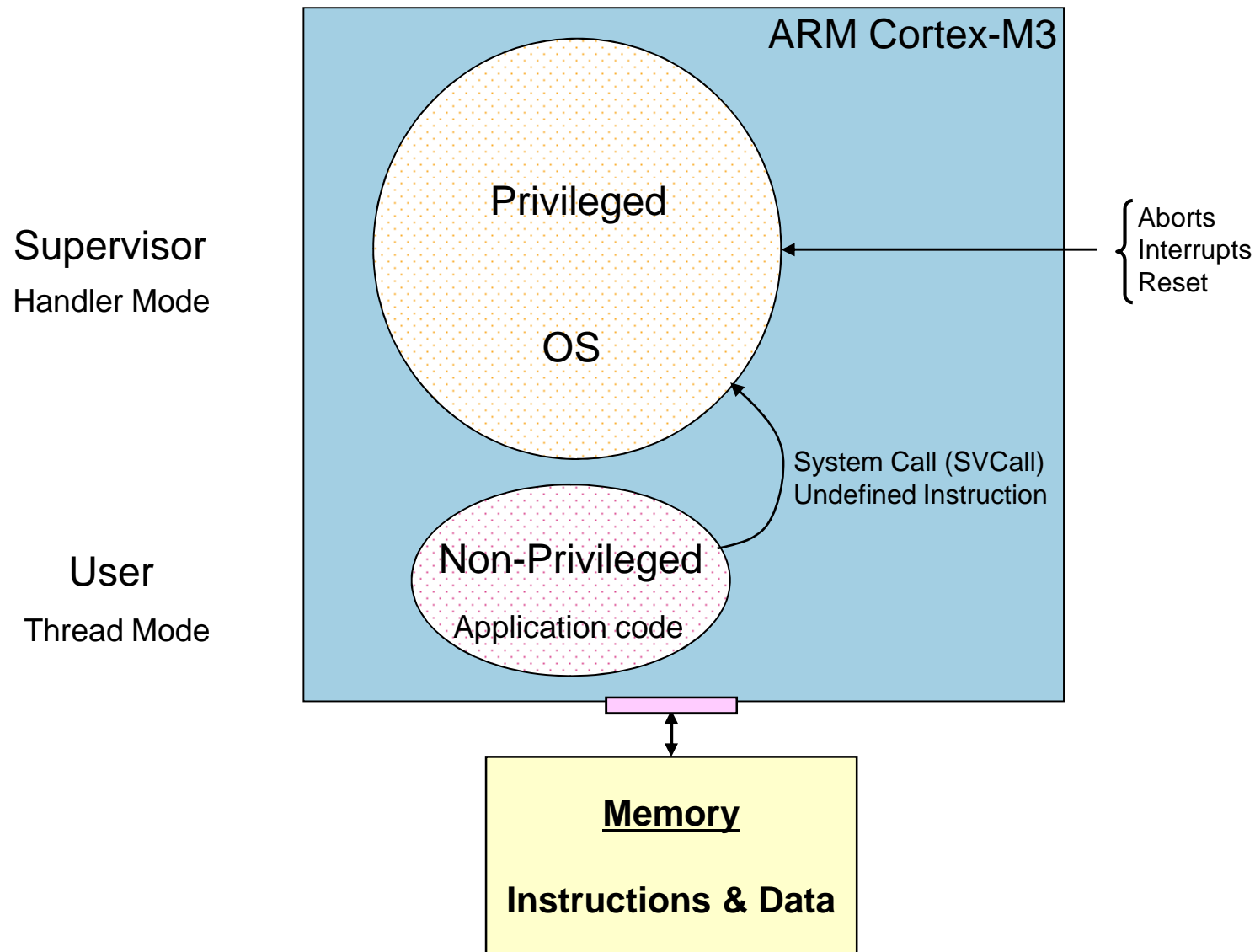
NXP LPC1311/13/42/43 Block Diagram



NXP LPC1311/13/42/43 Memory Map



Processor Privilege

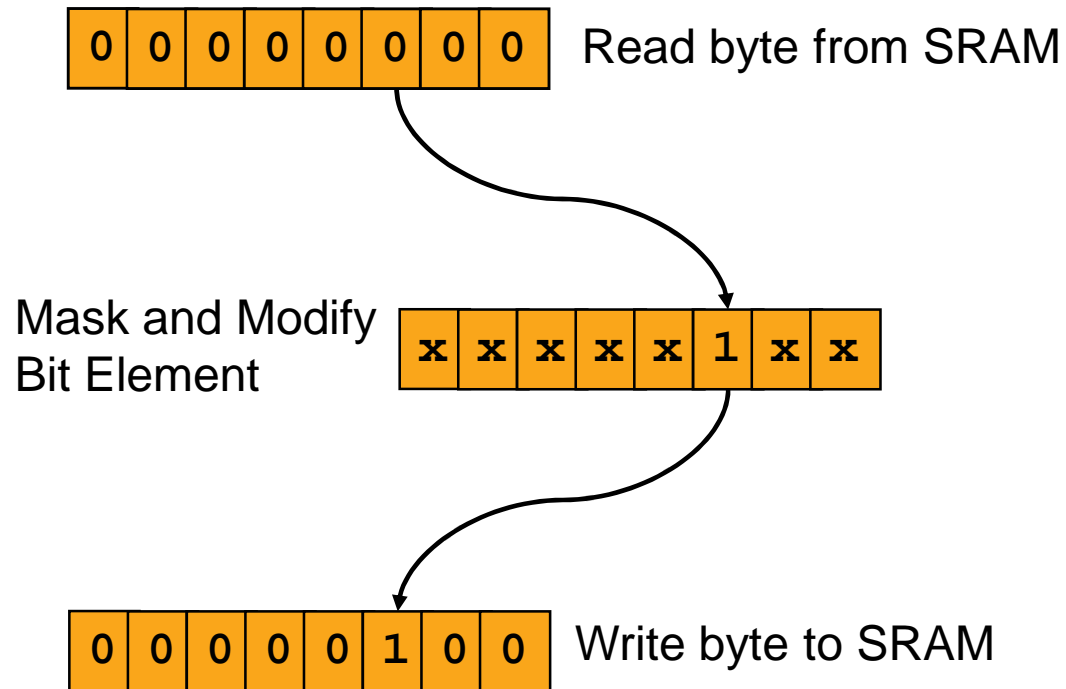


Memory Protection Unit (MPU)

- MPU provides access control for various memory regions
- Zero Latency Memory Protection
 - 8 register-stored regions
 - Same regions used for instructions and data
 - Minimum region size 32 Bytes (max 4GB)
 - No address translation or page tables
- Configured via memory-mapped control registers

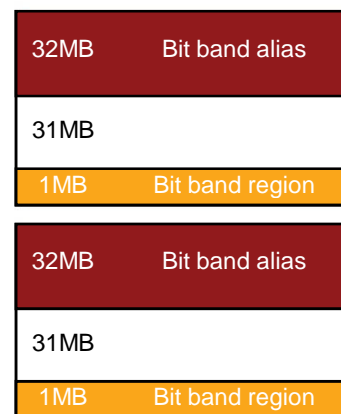
Cortex-M3 Bit Banding

Traditional Method of Atomic Manipulation

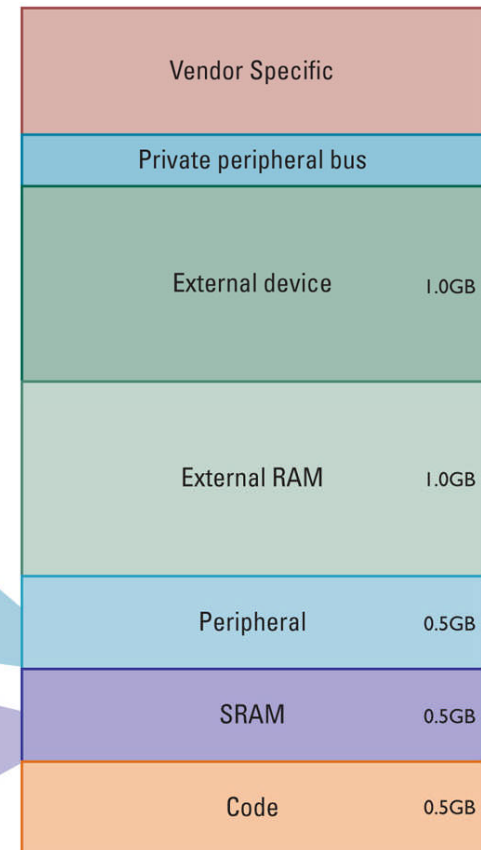


Cortex-M3 Bit Banding

- Writes to a word address in the bit band alias affect a single bit in the bit band region
- The write is translated to an atomic read-modify-write by the Cortex-M3 bus matrix
- Bit 0 of the stored register is written to the appropriate bit

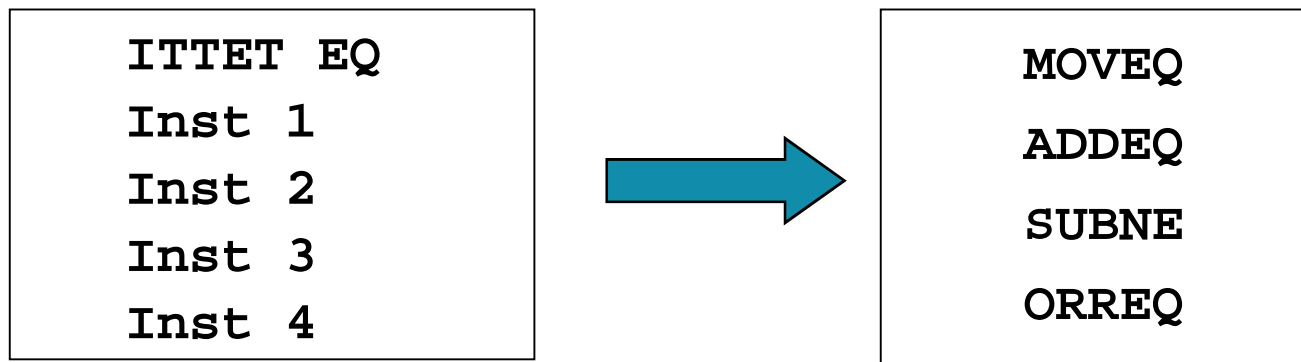


Memory map



Conditional Execution

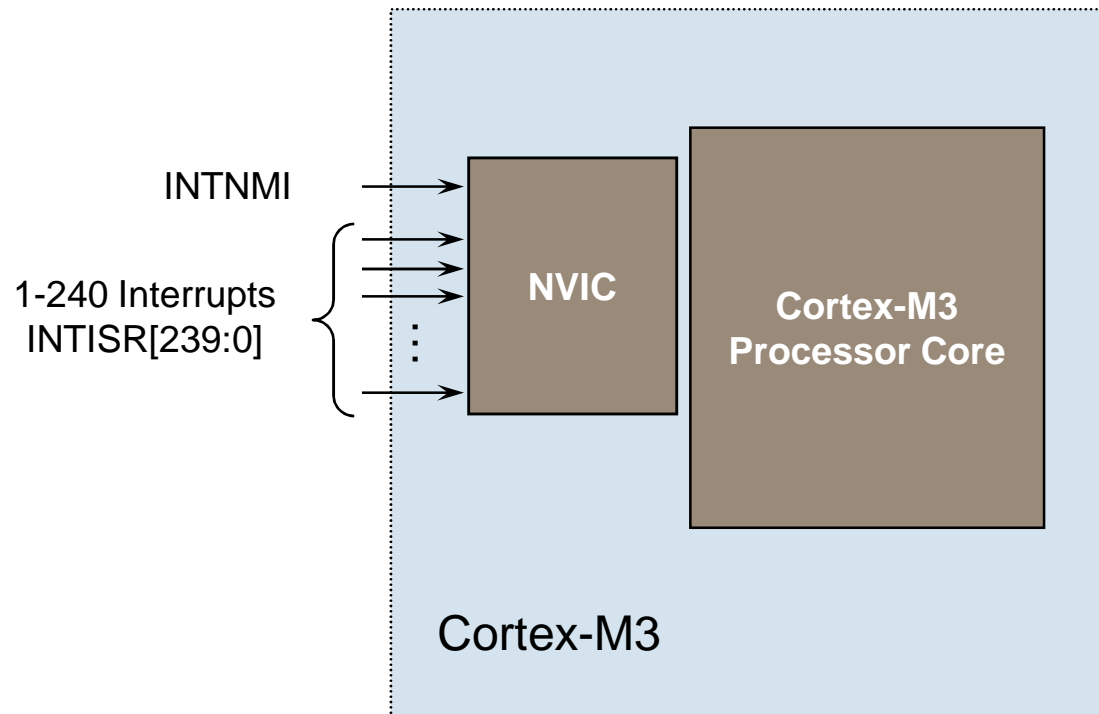
- If – Then (IT) instruction added (16 bit)
 - Up to 3 additional “then” or “else” conditions maybe specified (T or E)
 - Makes up to 4 following instructions conditional



- Any normal ARM condition code can be used
- 16-bit instructions in block do not affect condition code flags
 - Apart from comparison instruction
 - 32 bit instructions may affect flags (normal rules apply)
- Current “if-then status” stored in CPSR
 - Conditional block maybe safely interrupted and returned to
 - Must NOT branch into or out of ‘if-then’ block

Interrupt Handling

- One Non-Maskable Interrupt (INTNMI) supported
- 1-240 prioritizable interrupts supported
 - Interrupts can be masked
 - Implementation option selects number of interrupts supported
- Nested Vectored Interrupt Controller (NVIC) is tightly coupled with processor core
- Interrupt inputs are active HIGH



Exception Handling

- **Reset**
- **NMI**
- **Faults**
 - Hard Fault
 - Memory Manage
 - Bus Fault
 - Usage Fault
- **SVCall**
- **Debug Monitor**
- **PendSV**
- **SysTick Interrupt**
- **External Interrupt**

Power Management

- Multiple sleep modes supported
 - **Controlled by NVIC**
 - **Sleep Now – Wait for Interrupt/Event instructions**
 - **Sleep On Exit – Sleep immediately on return from last ISR**
 - **Deep Sleep**
 - Long duration sleep, so PLL can be stopped
 - Exports additional output signal SLEEPDEEP
- Cortex-M3 system is clock gated in all sleep modes
 - Sleep signal is exported allowing external system to be clock gated also
 - NVIC interrupt Interface stays awake
- Wake-Up Interrupt Controller (WIC)
 - External wake-up detector allows Cortex-M3 to be fully powered down
 - Effective with State-Retention / Power Gating (SRPG) methodology

Cortex-M3 Debug

- Single stepping
- ITM (Instrumentation Trace Module)
 - Support for instrumented code
 - Like “printf” debugging, but single cycle writes to ITM module can be exported via serial interface
- Optional ETM (Embedded Trace Module) which provides instruction trace
- Access to all memory and registers via Debug Access Port (DAP)
- Profiling Support
 - Helps you optimize your code
 - A hardware triggered PC-sampler is provided by the Cortex-M3 core
- Flash patch / Breakpoints
 - 6 instruction comparators, ARMv5T BKPT instruction is supported
 - 2 literal comparators
 - Allows flash code to be remapped into SRAM or system address space
- Data Watchpoint and Trace (DWT) that implements 4 h/w watchpoints

Agenda

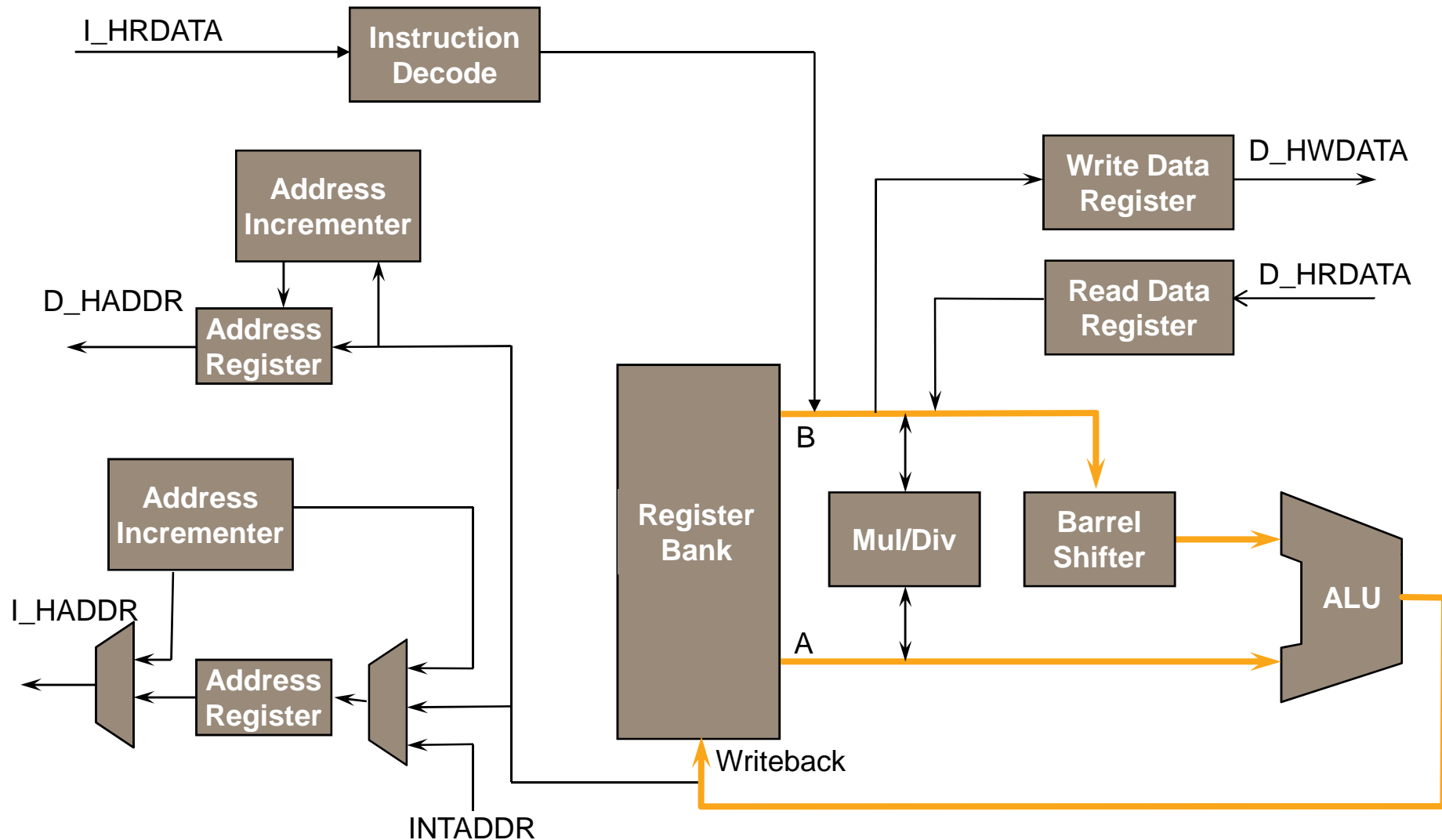
Cortex-M3 Overview

v7-M Architecture/Programmers Model

■ **Data Path and Pipelines**

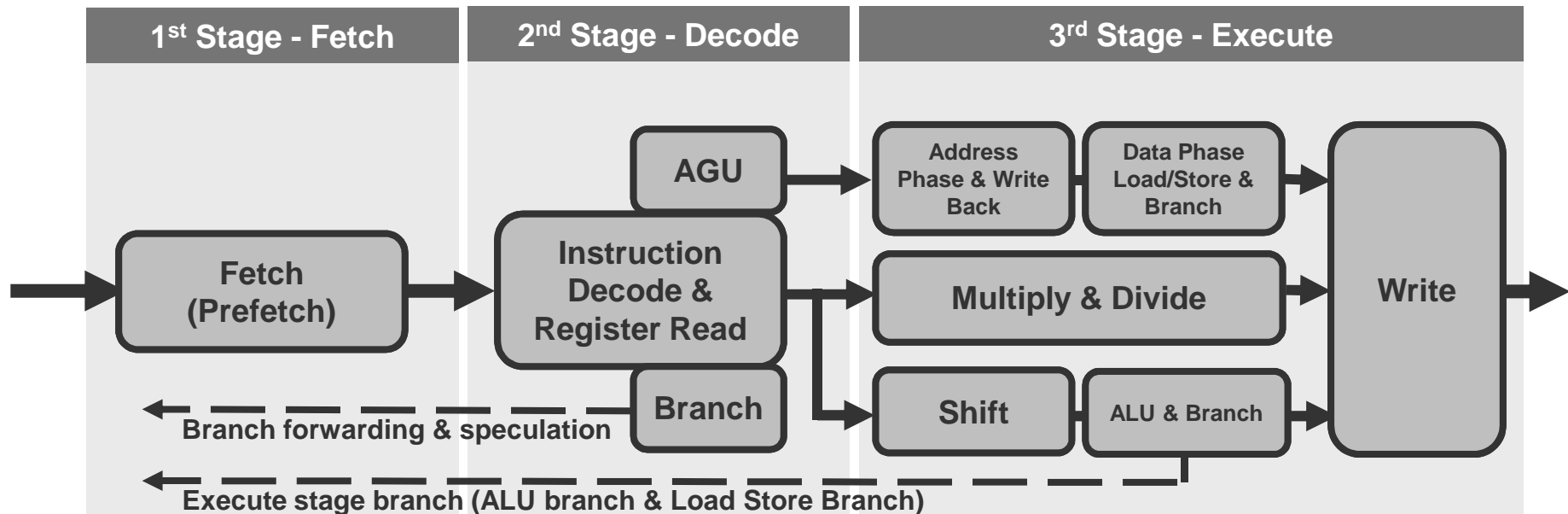
Tools and mbed Platform

Cortex-M3 Datapath



Cortex-M3 Pipeline

- Cortex-M3 has 3-stage fetch-decode-execute pipeline
 - Similar to ARM7
 - Cortex-M3 does more in each stage to increase overall performance



Optimal Pipelining

Cycle		1	2	3	4	5	6	7	8	9
Operation										
ADD	F	D	E							
SUB		F	D	E						
ORR			F	D	E					
AND				F	D	E				
ORR					F	D	E			
EOR						F	D	E		

F - Fetch D - Decode E - Execute

- All operations here are on registers (single cycle execution)
- In this example it takes 6 clock cycles to execute 6 instructions
- Clock cycles per Instruction (CPI) = 1

Branch Pipeline Example

Cycle		1	2	3	4	5	6	7	8	9
Address	Operation									
0x8000	BX r5	F	D	E						
0x8002	SUB		F	D						
0x8004	ORR			F						
0x8FEC	AND				F	D	E			
0x8FEE	ORR					F	D	E		
0x8FF0	EOR						F	D	E	
F - Fetch D - Decode E - Execute										

- It takes 3 cycles to complete the branch
- Worst case scenario – indirect branch taken

LDR Pipeline Example

Cycle		1	2	3	4	5	6	7	8	9
Operation										
ADD	F	D	E							
SUB		F	D	E						
LDR			F	D	Ea	Ed				
AND				F	D	S	E			
ORR					F	S	D	E		
EOR						F	D	E		

F - Fetch **D** - Decode **E** - Execute **S** - Stall
Ea - LDR address phase **Ed** - LDR data phase

- In this example it takes 7 clock cycles to execute 6 instructions
- Clock cycles per Instruction (CPI) = 1.2
- The read cycle must complete on the bus before the LDR instruction can complete since there is only one writeback port in the register file

STR-STR Pipeline Example

Cycle		1	2	3	4	5	6	7	8	9
Operation										
ADD	F	D	E							
SUB		F	D	E						
STR			F	D	Ea	Ed				
STR				F	D	Ea	Ed			
ORR					F	D	E			
EOR						F	D	E		

F - Fetch **D** - Decode **E** - Execute **S** - Stall
Ea - Execute / STR address phase **Ed** - STR data phase

- Store buffer allows STR instruction to finish before store cycle completes on the bus
- Back-to-back STR instructions pipeline on the AHB-Lite bus
- Also works for LDR-STR-LDR-STR

Agenda

Cortex-M3 Overview

v7-M Architecture/Programmers Model

Data Path and Pipelines

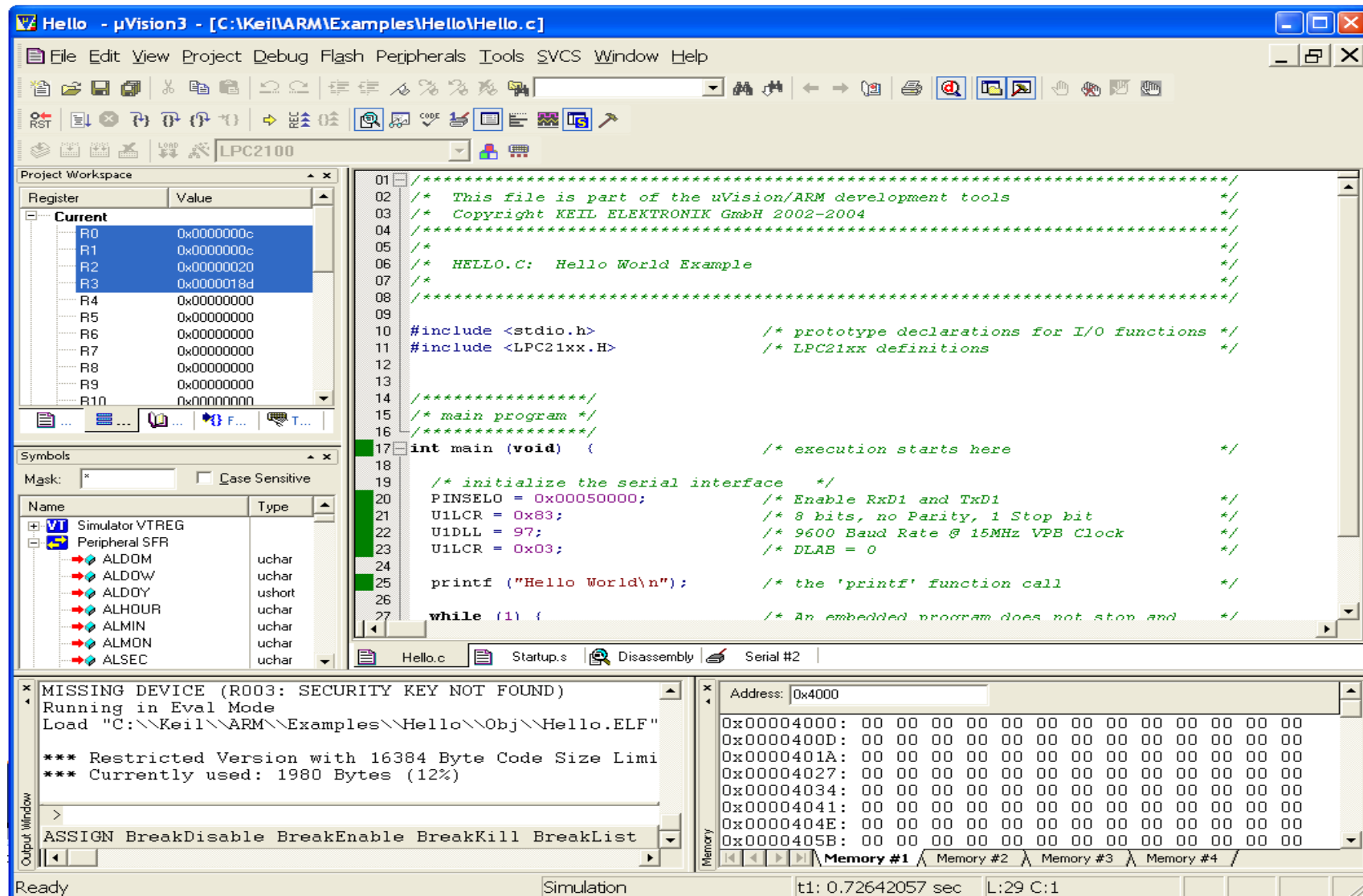
- **Tools and mbed Platform**

RVMDK Software Development Tools

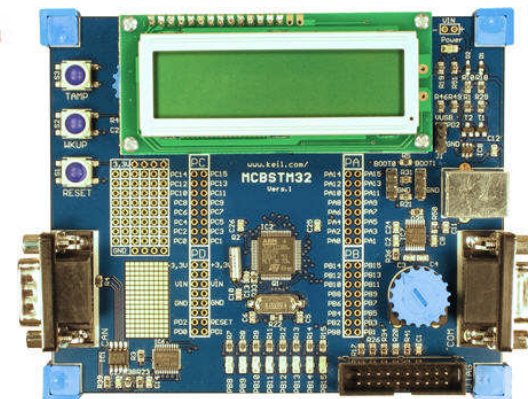
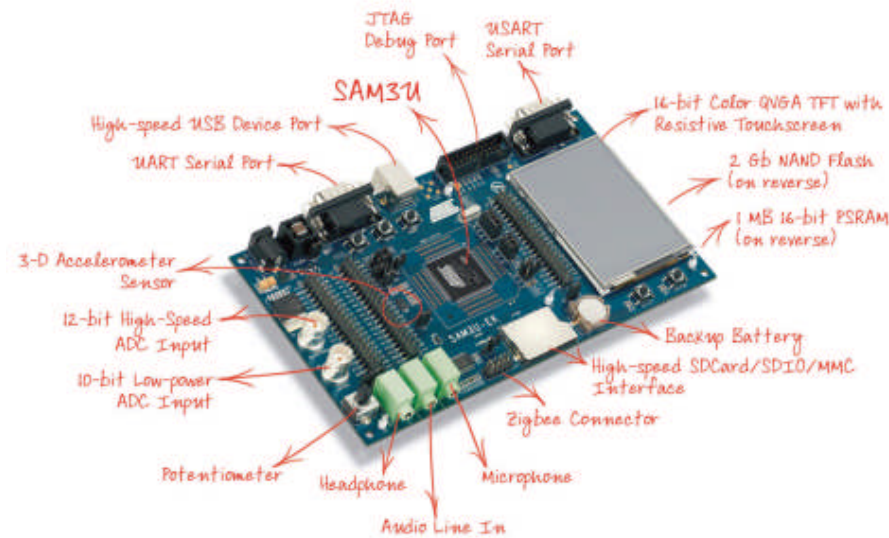
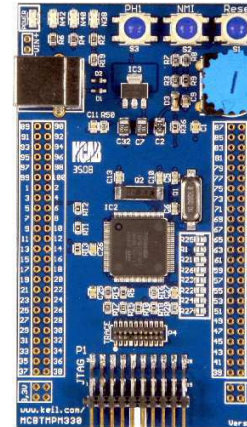
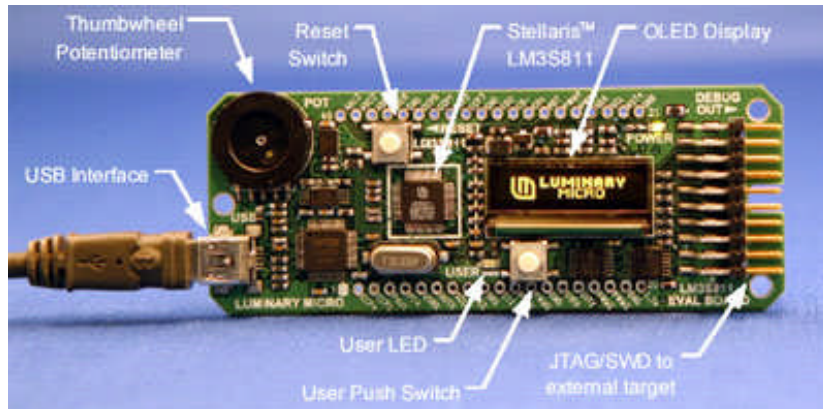


- Includes ARM macro assembler, compilers (ARM RealView C/C++ Compiler, Keil CARM Compiler, or GNU compiler), ARM linker, Keil uVision Debugger and Keil uVision IDE
- Keil uVision Debugger accurately simulates on-chip peripherals (I²C, CAN, UART, SPI, Interrupts, I/O Ports, A/D and D/A converters, PWM, etc.)
- Evaluation Version
 - 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/>

RVMDK Software Development Tools



Cortex-M3 Development Platforms



Rapid Prototyping

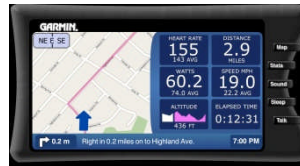
- **Rapid Prototyping helps industries create new products**
 - Control, communication and interaction increasingly define products
 - Development cycles for microelectronics have not kept pace



3D Moulding



3D Printing



2D/3D Design

django

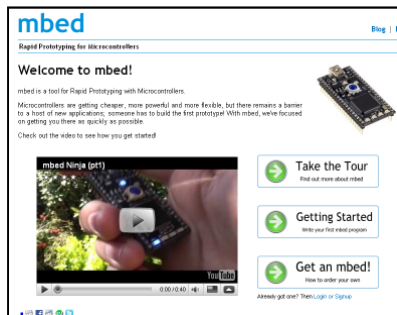


Web Frameworks

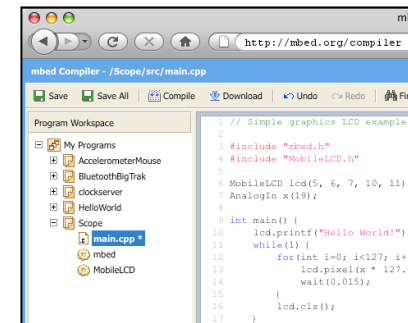
mbed

Getting Started and Rapid Prototyping with ARM MCUs

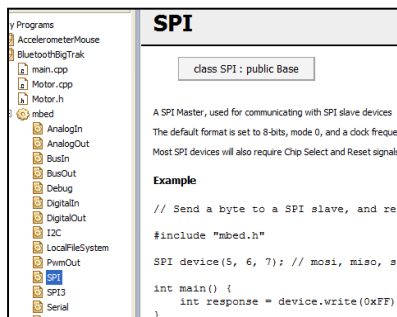
Complete Targeted Hardware, Software and Web 2.0 Platform



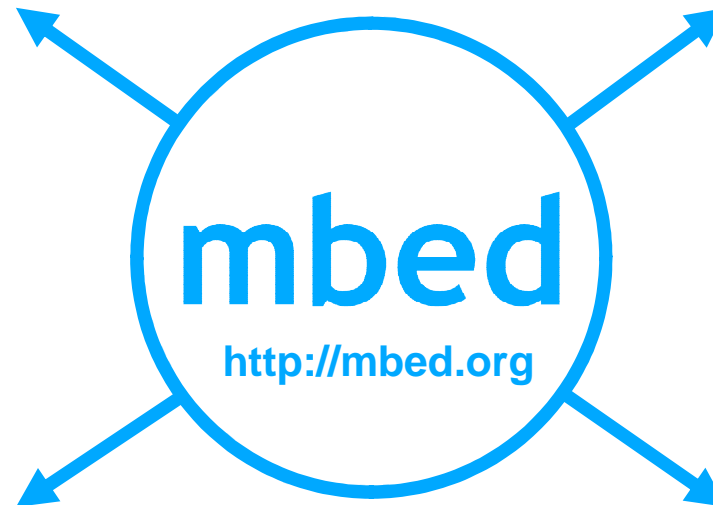
Dedicated Developer Web Platform



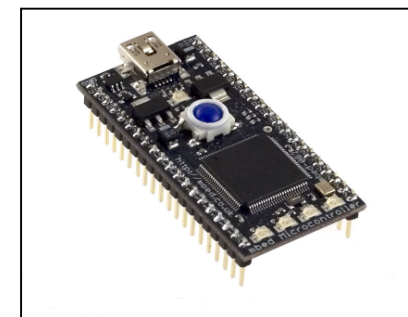
Lightweight Online Compiler



High-level Peripheral APIs



**Rapid Prototyping
for Microcontrollers**



**Cortex-M3 MCU in a
Prototyping Form-Factor**

mbed Motor Controller Example

```
#include "mbed.h"
DigitalOut leftfwd(p16);
DigitalOut leftrev(p17);
PwmOut pwm(p21);
AnalogIn pot(p20);
int main(){
    pwm.period_ms(1);
    while(1){
        float potmod = pot - 0.5;
        if (potmod >= 0){
            leftfwd = 1;
            leftrev = 0;
            pwm=((1.8 * potmod) + 0.1);
        } else {
            leftfwd = 0;
            leftrev = 1;
            pwm = fabs((1.8 * potmod) - 0.1);
        }
    }
}
```

Documentation

- ARM v7-M Architecture Reference Manual (ARM v7-M ARM)
- Cortex-M3 Technical Reference Manual (TRM)
- ARM Debug Interface V5 Architecture Spec (ADIv5)
- “The Definitive Guide To The ARM Cortex-M3” by Joseph Yiu (Elsevier)

University Resources

- <http://www.arm.com/support/university/>
- university@arm.com
- <http://www.mbed.org/>
- <http://www.keil.com/>

Fin



The Architecture for the Digital World®

ARM®