



EMBEDDED SYSTEM COURSE

LECTURE 4: ARM CORTEX-M ARCHITECTURE OVERVIEW

- Hỏi lại kiến thức bài cũ.
- Giới thiệu nội dung bài mới: Kiến trúc về Cortex-M, trong đó tập trung về Cortex M0/M0+.
- Mục đích của bài giảng, nhằm giúp học viên có khái niệm cơ bản về ARM CortexM, bao gồm: architecture, programer model, instruction set, core register (general/ status/ control registers).
- Trước khi vào bài mới, giới thiệu qua về ARM Ltd.

ARM (Advanced RISC Machines) founded in November 1990

Company headquarters: Cambridge, UK

- Processor design centers in Cambridge, Austin, and Sophia Antipolis
- Sales, support, and engineering offices all over the world

Best known for its range of RISC processor cores designs

Other products – fabric IP, software tools, models, cell libraries - to help partners develop and ship ARM-based SoCs

ARM does NOT manufacture silicon

More information about ARM: <http://www.arm.com/aboutarm/>

Learning Goals



- Introduce about the ARM Cortex M processor.
- Explain some core components in Cortex-M including NVIC, SysTick timer and Floating Point Unit.
- Explain about the basic concepts on Cortex-M instruction set.

2

- All the ARM Cortex M processors are 32-bit RISC (Reduced Instruction Set Computing) processors. They have:
 - 32-bit registers
 - 32-bit internal data path
 - 32-bit bus interface
- The Cortex-M processors contain the core of the processor, NVIC, the SysTick timer, and optionally the floating point unit (for Cortex-M4).
- All the ARM Cortex -M processors are based on Thumb technology, which allows a mixture of 16-bit and 32-bit instructions to be used within

Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- Programmer Model
- Instruction Set
- Summary

Giới thiệu về các mục của buổi học:

Phần 1: Giới thiệu thông tin tổng quan về ARM Cortex, các dòng chip của ARM, sự khác nhau giữa các dòng.

Phần 2: Giới thiệu về kiến trúc tổng quan của ARM Cortex

Phần 3: Programmer Model

Phần 4: Kiến trúc tập lệnh sử dụng trong Arm cortex

Phần 5: Tổng kết

Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- Programmer Model
- Instruction Set
- Summary

Giới thiệu về các mục của buổi học:

Phần 1: Giới thiệu thông tin tổng quan về ARM Cortex, các dòng chip của ARM, sự khác nhau giữa các dòng.

Phần 2: Giới thiệu về kiến trúc tổng quan của ARM Cortex

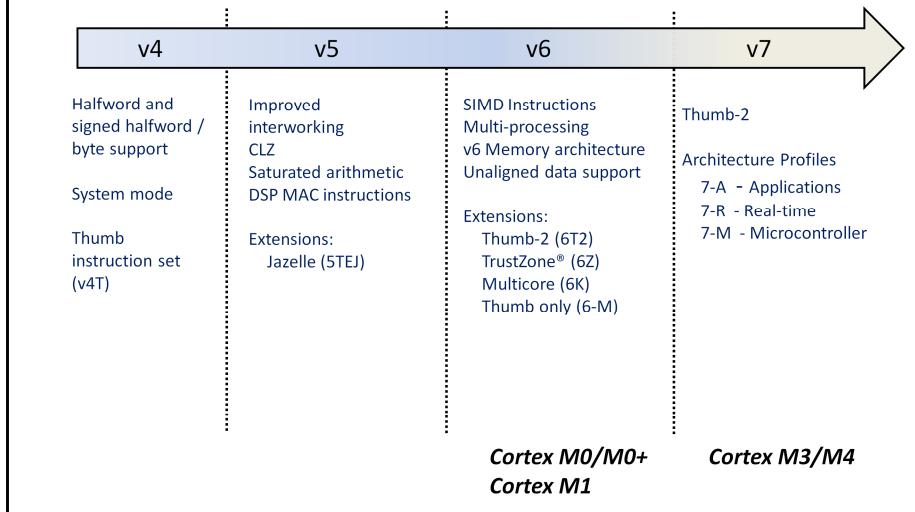
Phần 3: Programmer Model

Phần 4: Kiến trúc tập lệnh sử dụng trong Arm cortex

Phần 5: Tổng kết

General Information about the Cortex-M

Development of the ARM Architecture



Giới thiệu sự phát triển các kiến trúc trong ARM core, tập trung vào kiến trúc v6 và v7.

Version 4

- Add the signed, unsigned half-word and signed byte load and store instructions
- System mode is introduced

Version 4T

- 16-bit Thumb compressed form of the instruction set is introduced

Version 5T

- Introduced recently, a superset of version 4T adding the BLX, CLZ and BRK instructions

Version 5TE

Add the signal processing instruction set extension

Version 6/7

- Media processing extensions (SIMD)
 - 2x faster MPEG4 encode/decode
 - 2x faster audio DSP
- Improved cache architecture
 - Physically addressed caches
 - Reduction in cache flush/refill
 - Reduced overhead in context switches
- Improved exception and interrupt handling
 - Important for improving performance in real-time tasks
- Unaligned and mixed-endian data support
 - Simpler data sharing, application porting and saves memory

Note: V7: Support Thumb-2 Only

General Information about the Cortex-M ARM CORTEX



The ARM Cortex family includes processors based on the three distinct profiles of the ARMv7 architecture.

The A profile for sophisticated, high-end applications running open and complex operating systems

The R profile for real-time systems

The M profile optimized for cost-sensitive and microcontroller applications

Cortex-M : The ARM Cortex-M is a group of 32-bit RISC ARM processor cores licensed by ARM Holdings. **The cores are intended for microcontroller use,** and consist of the Cortex-M0, Cortex-M0+, Cortex-M1, Cortex-M3, and Cortex-M4

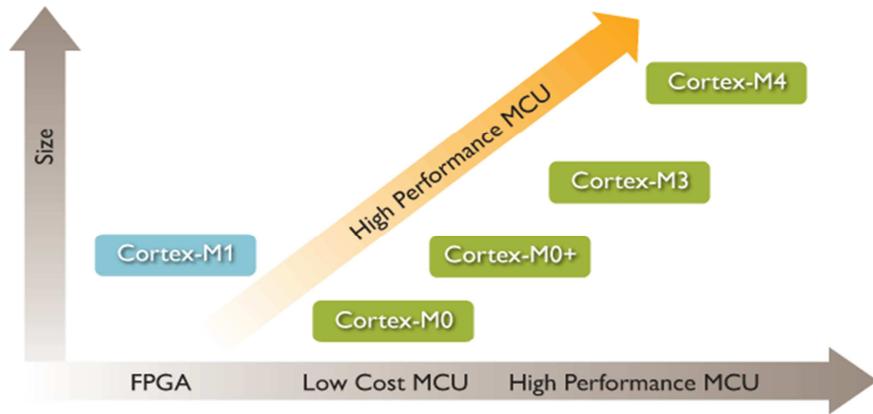
The Cortex-M family is optimized for cost and power sensitive MCU and mixed-signal devices for end applications such as smart metering, human interface devices, automotive and industrial control systems, white goods, consumer products and medical instrumentation.

- Cortex-R: The ARM Cortex-R is a group of 32-bit RISC ARM processor cores licensed by ARM Holdings. The cores are intended for robust real-time use, and consists of the Cortex-R4, Cortex-R5, Cortex-R7

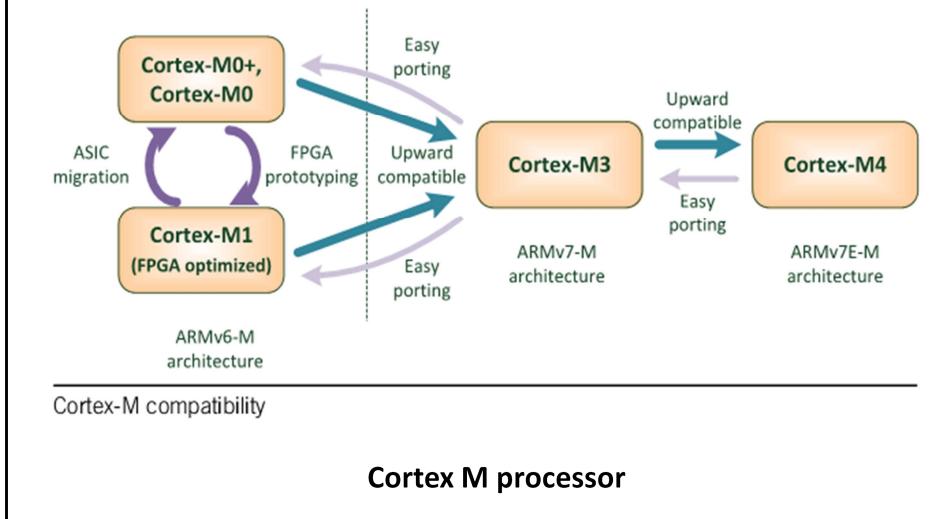
The ARM® Cortex®-R real-time processors offer high-performance computing solutions for embedded systems where reliability, high availability, fault tolerance, maintainability and real-time responses are required. Features:**High performance, real-time, Safe, Cost effective**

(Note: Cortex A: series of applications processors provide an entire range of solutions for devices hosting a rich OS platform and user applications ranging from ultra-low-cost handset through smartphones, mobile computing platforms, digital TV and set-top boxes to enterprise networking, printers and server solutions.

General Information about the Cortex-M



General Information about the Cortex-M



Cortex-M family processors are all **binary upwards compatible**, enabling software reuse and a seamless progression from one Cortex-M processor to another.

Note: Cortex M0/M0+/M1: **Von Neumann architecture** / Thumb-2 base on arm v6-m

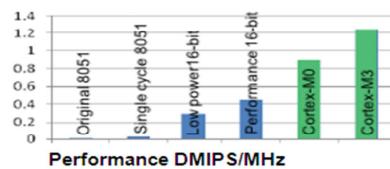
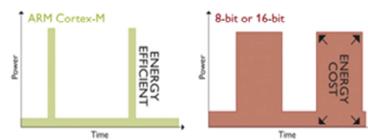
Cortex M3/4 : **Harvard** / Thumb-2 base on ARMv7M

Topic thảo luận: Chương trình viết cho Cortex M0, có build đc trên Cortex M3 ko ? Và ngược lại ? Tại sao?

General Information about the Cortex-M

Cortex-M Advantages:

1. Energy efficiency
2. Smaller code
3. Ease of use
4. High performance



Energy efficiency

Lower energy costs, longer battery life

Run at lower MHz or with shorter activity periods

Architected support for sleep modes

Work smarter, sleep longer than 8/16-bit

Smaller code

Lower silicon costs

High density instruction set

Achieve more per byte than 8/16-bit devices

Smaller RAM, ROM or Flash requirement

Ease of use

Faster software development and reuse

Global standard across multiple vendors

Code compatibility

Unified tools and OS support

High performance

More competitive products

Powerful Cortex-M processor

Delivers more performance per MHz

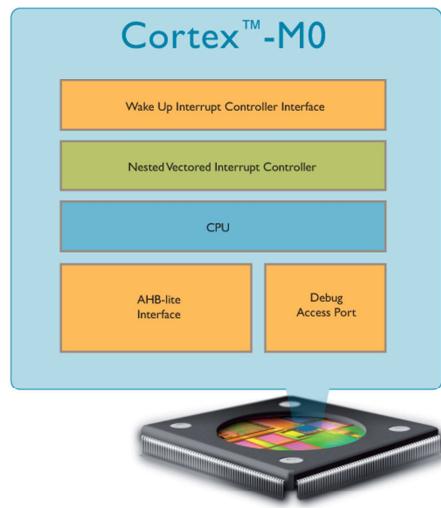
Enables richer features at lower power

General Information about the Cortex-M



Cortex M0

- 32-bit RISC processor
- 3-stage pipeline von Neumann architecture
- ARMv6-M architecture
- 16-bit Thumb instruction set with Thumb-2 technology.
- Load-Store Architecture
- 56 Instructions
- Low power support



Note: Cortex M0+: 2 stage pipeline

Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- Programmer Model
- Instruction Set
- Summary

Giới thiệu về các mục của buổi học:

Phần 1: Giới thiệu thông tin tổng quan về ARM Cortex, các dòng chip của ARM, sự khác nhau giữa các dòng.

Phần 2: Giới thiệu về kiến trúc tổng quan của ARM Cortex

Phần 3: Programmer Model

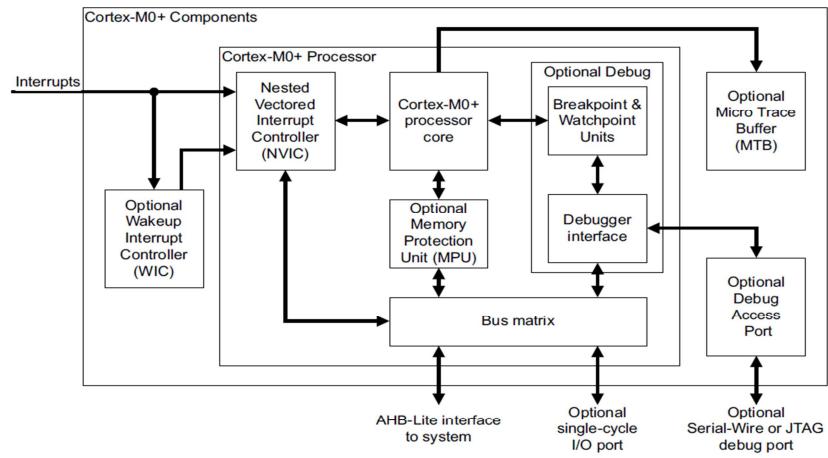
Phần 4: Kiến trúc tập lệnh sử dụng trong Arm cortex

Phần 5: Tổng kết

Introduction to the architecture



Simplified Block Diagram



Introduction to the architecture



Cortex-M0 Functional Blocks

- ARMv6-M Thumb instruction set
- NVIC: 32 external interrupt inputs
- Debug: 4 HD breakpoints, 2 watchpoints.
- Bus interfaces: 32-bit AMBA-3 AHB-Lite system interface

Introduction to the architecture



Memory model

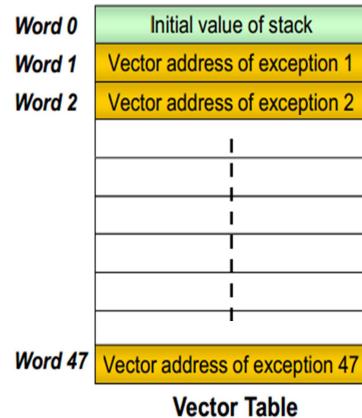
- 32-bit address space
- Virtual memory is not supported in ARMv6-M.
- Instruction fetches are always half-word-aligned
- Data accesses are always naturally aligned

Introduction to the architecture



Exception model

- An exception may be an interrupt or a hardware error.
- Each exception has exception number, priority number and vector address
- Vector table base address is fixed at 0x00.

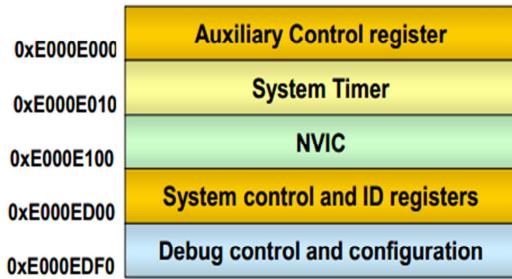


Introduction to the architecture

System control space

Consists of the following groups:

- CPUID space.
- System control, configuration and status.
- SysTick system timer
- Nested Vectored Interrupt Controller (NVIC)



System Control Space:

System control register

Cortex-M0 status and operating mode

- Including CPUID, CortexM0 interrupt priority and Cortex M0 management

SysTick: 24-bit clear-on-write, decrementing, wrap-on-zero counter.

- be used as a Real Time Operating System (RTOS) tick timer or as a simple counter.

- The reference clock can be the core clock or an external clock source.

NVIC (Nested Vectored Interrupt Controller)

Upon stack based exception model

Restore registers to resume to normal execution automatically

Remove redundant PUSH/POP operations needed by traditional C-based ISRs.

Supports up to 32 (IRQ[31:0]) discrete interrupts which can be either level-sensitive or pulse-sensitive.

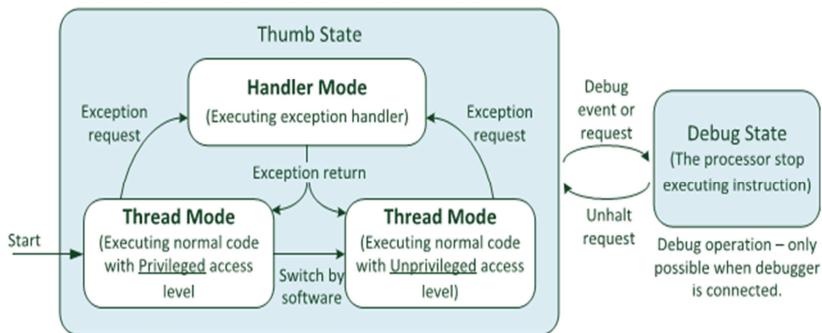
Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- **Programmer Model**
- Instruction Set
- Summary

Programmer's Model

Operation Modes & States



ARM Cortex M support two operation mode:

Thread mode

Used to execute application software. The processor enters Thread mode when it comes out of reset.

Handler mode

Used to handle exceptions. The processor returns to Thread mode when it has finished all exception processing.

Note:

Other ARM architectures support the concept of privileged or unprivileged software execution. This processor does not support different privilege levels. Software execution is always privileged, meaning software can access all the features of the processor.

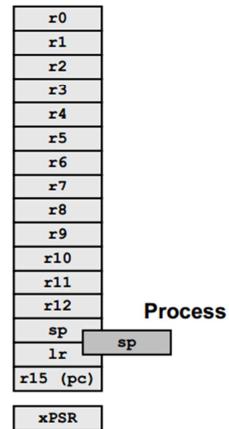
The debug state is used for debugging operations only. This state is entered by a halt request from the debugger, or by debug events generated from debug components in the processor.

Programmer's Model



Core Registers

- All registers are 32 bits wide
- 13 general purpose registers
- Registers r0 – r7 (Low registers)
- Registers r8 – r12 (High registers)
- 3 registers with special meaning/usage
 - Stack Pointer (SP) – r13
 - Link Register (LR) – r14
 - Program Counter (PC) – r15
- Special-purpose registers
 - xPSR shows a composite of the content of
 - APSR, IPSR, EPSR



Register Usage

R0 – R3: Arguments into function Result(s) from function otherwise corruptible
(Additional parameters passed on stack)

R4-R11: Register variables. Must be preserved

R12: Scratch register (corruptible)

R13 Stack Pointer (SP)

R14 Link Register (LR)

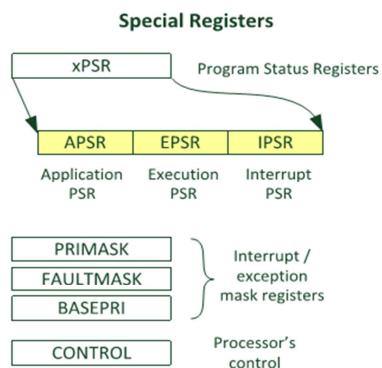
R15 Program Counter (PC)

Note: The compiler has a set of rules known as a Procedure Call Standard that determine how to pass parameters to a function (see AAPCS) CPSR flags may be corrupted by function call. Assembler code which links with compiled code must follow the AAPCS at external interfaces. The AAPCS is part of the ABI for the ARM Architecture

Programmer's Model



Special Registers



APSR, IPSR, and EPSR

| | | | | | | | | | | | | | | | | |
|------|----|----|----|----|----|---------|----|-------|-------|---------|---|---|---|---|---|------------------|
| | 31 | 30 | 29 | 28 | 27 | 26:25 | 24 | 23:20 | 19:16 | 15:10 | 9 | 8 | 7 | 6 | 5 | 4:0 |
| xPSR | N | Z | C | V | Q | I/O/I/T | T | | GE* | I/O/I/T | | | | | | Exception Number |

Register Usage

R0 – R3: Arguments into function Result(s) from function otherwise corruptible
(Additional parameters passed on stack)

R4-R11: Register variables. Must be preserved

R12: Scratch register (corruptible)

R13 Stack Pointer (SP)

R14 Link Register (LR)

R15 Program Counter (PC)

Note: The compiler has a set of rules known as a Procedure Call Standard that determine how to pass parameters to a function (see AAPCS) CPSR flags may be corrupted by function call. Assembler code which links with compiled code must follow the AAPCS at external interfaces. The AAPCS is part of the ABI for the ARM Architecture

Programmer's Model



Stack

- Full descending: stack pointer indicates the last stacked item on the stack memory.
- Two stacks, two independent stack pointers.
- Handler mode always uses the MSP (Main Stack Pointer)
- Thread mode can use MSP (Main Stack Pointer) by default, or PSP (Process Stack Pointer).
- In an OS environment, ARM recommends that threads running in Thread mode use the process stack and the kernel and exception handlers use the main stack

Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- Programmer Model
- **Instruction Set**
- Summary

Instruction Set Architecture

- The ARM Architecture is a Load/Store architecture
 - No direct manipulation of memory contents
 - Memory must be loaded into the CPU to be modified, then written back out

Instruction Set Architecture

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

Introduction của từng mục lớn Trong Topic
- Goal → Bắt buộc

Instruction Set Architecture

Cortex M0 ISA Overview:

- ARMv6-M supports Thumb-2 technology
(The ARM instruction set is not supported)
- Thumb-2 technology supports mixed 16-bit/32-bit instructions
- Small number of additional 32-bit instructions supported
- Conditional execution is supported
- Optimized for compilation from C
 - Thumb-2 instructions are not designed to be written by hand
 - Easy to learn due to small number of mnemonics

Introduction của từng mục lớn Trong Topic
- Goal → Bắt buộc

Instruction Set Architecture

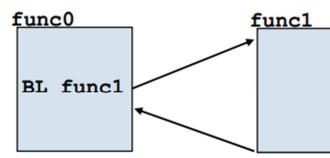
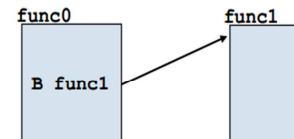
Instruction Classes

- Branch instructions
- Data-processing instructions
- Load and store instructions
- Status register access instructions
- Miscellaneous instructions

Instruction Set Architecture

Branch instructions

- B – Branch
 - Absolute branch to a target address, relative to Program Counter (PC)
 - +/- 256 bytes range, conditional execution supported
 - +/- 1MB range, no conditional execution supported
- BL – Branch with Link
 - Branch to a subroutine – Link register is updated
 - +/- 16MB range, relative to Program Counter (PC)



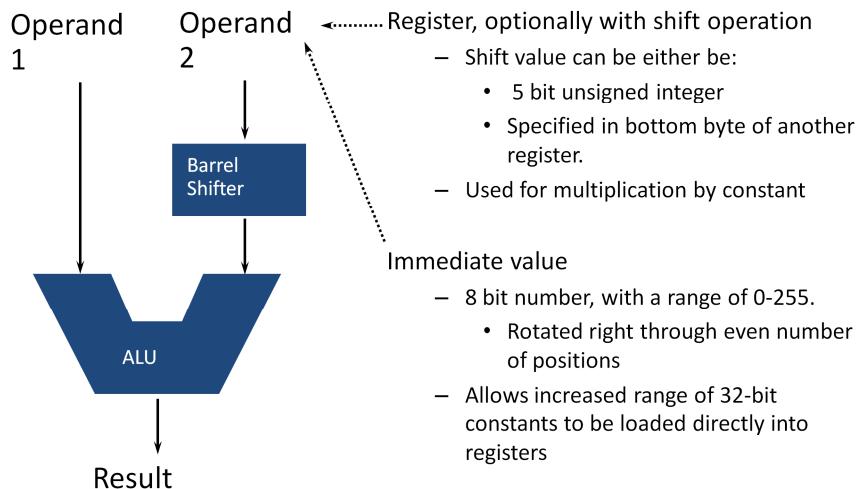
Instruction Set Architecture

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.

Instruction Set Architecture

Using Barrel Shifter:



Instruction Set Architecture

Load and store instructions:

| | | |
|--------------|-------------|----------------------|
| 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**

Instruction Set Architecture

Status Register Access Instructions

MRS/MSR - Move data between a general purpose register and status register

- MRS (Register \leftarrow Status Register)
- MSR (Status Register \leftarrow Register)

- Syntax:

- **LDR**{<cond>}{{<size>}} Rd, <address>
 - **STR**{<cond>}{{<size>}} Rd, <address>

Table of contents



- General Information about the Cortex-M
- Introduction to the architecture
- Programmer Model
- Instruction Set
- Summary

Giới thiệu về các mục của buổi học:

Phần 1: Giới thiệu thông tin tổng quan về ARM Cortex, các dòng chip của ARM, sự khác nhau giữa các dòng.

Phần 2: Giới thiệu về kiến trúc tổng quan của ARM Cortex

Phần 3: Programmer Model

Phần 4: Kiến trúc tập lệnh sử dụng trong Arm cortex

Phần 5: Tổng kết

Summary



- All the ARM Cortex M processors are 32-bit RISC (Reduced Instruction Set Computing) processors. They have:
 - 32-bit registers
 - 32-bit internal data path
 - 32-bit bus interface
- The Cortex-M processors contain the core of the processor, NVIC, the SysTick timer, and optionally the floating point unit (for Cortex-M4).
- All the ARM Cortex -M processors are based on Thumb technology, which allows a mixture of 16-bit and 32-bit instructions to be used within

Question & Answer



Thanks for your attention !

Copyright



- This course including **Lecture Presentations, Quiz, Mock Project, Syllabus, Assignments, Answers** are copyright by FPT Software Corporation.
- This course also uses some information from external sources and non-confidential training document from Freescale, those materials comply with the original source licenses.

35