

Computer Organization Laboratory

Class Notes

Instructor: Ken Q. Yang
Dept. of ECE, URI



1



Section 0

Course Objectives, Plans, and Lab Tools



2



Course Objectives: What to learn?

- Computer Architecture Concepts
 - Instruction Set Architecture
 - CPU, Memory, and I/O Organizations
- Interfacing and Communication
 - Serial and parallel ports
 - UART, DMA, PI/Timer
 - Wired and wireless networking
- Applying μ Processor to Design Systems
- Advanced Topics

All are based on one specific commercial embedded processor: ARM Processor



3



Course Plan: How to Learn?

- Regular lectures (twice a week, 1:15 each)
 - Covers basic concepts and knowledge
 - Explain tools and techniques necessary
- Weekly laboratory experiments (minimum 3 hours/week)
- Assessments:
 - 5 Laboratory experiments,
 - Each lab accounts for 5% of your grade
 - 1 Design project, 15% of your grade
 - Design and documentations, 5% of your grade
 - Project proposal Presentation (5 minutes), 2%
 - Final project presentation (10 minutes), 4%
 - demonstration (10 minutes), 4%
 - Exams
 - 3 midterm exams, each accounts for 15% of your grade
 - 1 final exam, 20% of your grade



4



Option 2

- Regular lectures (twice a week, 1:15 each)
 - Covers basic concepts and knowledge
 - Explain tools and techniques necessary
- One or two design teams will be selected to design an [autonomous vehicle](#) to participate in Freescale Cup in April.
 - Design and implement a autonomous car
 - Meet weekly for at least 3 hours to discuss progress and demonstrate its operations
 - No Lab for the team
 - Select anyone of the Exams



5



Pope's Inauguration

Then...

“When smartphones and tablets

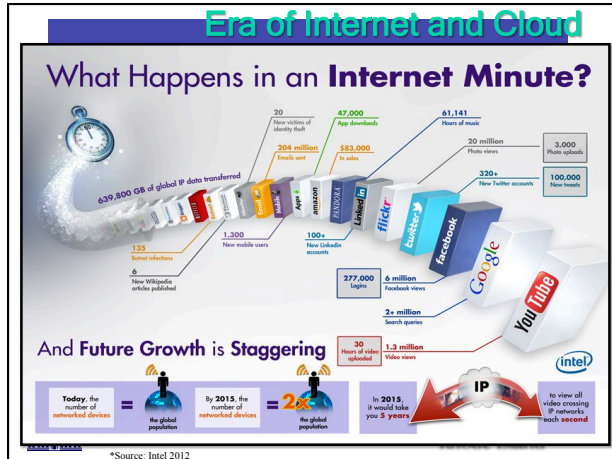
light up the sky, load up the clouds.”

Now...



Source: <http://www.aharvest.org/spokenword/2013-04-06-when-smartphones-and-tablets-load-up-the-sky>





All Boil Down to One Thing

Computer

A Very Large Fraction: Embedded Computers and Systems

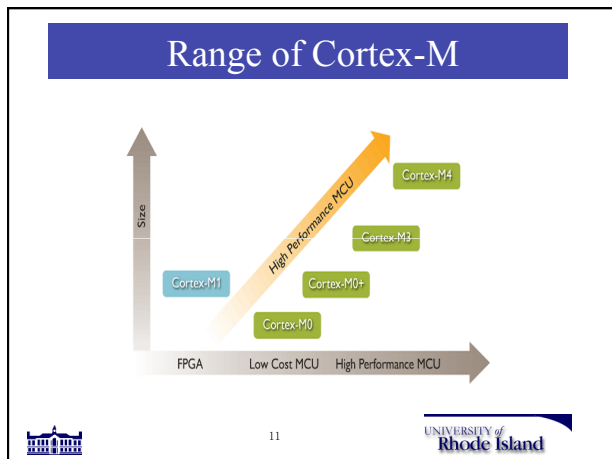
- End user devices
- Variety of appliances
- Network cores
- Consumer Electronics
- More and more

Sec. 1. ARM Family Processors

- ARM Cortex™-M Family
- Cortex™-M4 Features
- Kinetis Cortex™-M4 Enhancement

The Cortex™ Processor Family

Cortex™-A	servers	set top boxes	netbooks	mobile applications
Cortex™-R	disk drives	digital cameras	mobile baseband	
Cortex™-M	appliances	motors	audio	

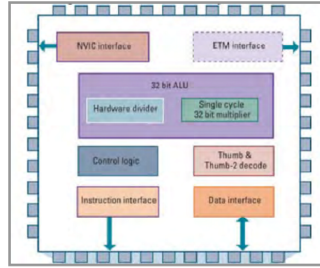


Outline

- ARM Cortex™-M Family
- Cortex™-M4 Features
- Kinetis Cortex™-M4 Enhancement

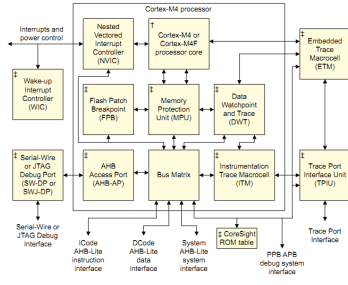
What is Cortex™-M

- Harvard Architecture
- 3 stage pipeline
- Single cycle multiply
- Hardware Divide
- Thumb-2 Instruction Set
- Vectored Interrupt Controller



13

CORTEX M-4

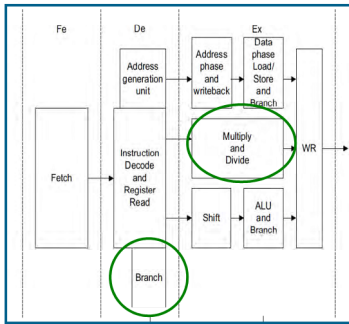


† For the Cortex-M4F processor, the core includes a Floating Point Unit (FPU)
‡ Optional component



14

Cortex™-M Pipeline



15

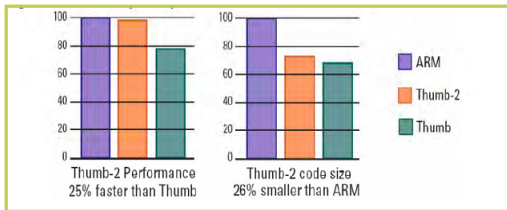
ARM® Instruction Set – Some History

- ARM Instruction Set – Original 32-bit Instruction Set
- Thumb Instruction Set – 16-bit Instruction Set
- Thumb-2 Instruction Set – Mixed 16/32 bit Instruction Set



16

Effects of the Thumb-2 ISA

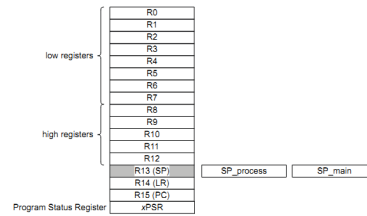


17

Register Sets 1

The processor has the following 32-bit registers:

- 13 general-purpose registers, R0-R12
- Stack Pointer (SP), R13 alias of banked registers, P_process and SP_main
- Link Register (LR), R14
- Program Counter (PC), R15
- Special-purpose Program Status Registers, (xPSR).



18

Register Sets 2

Low registers

Registers R0-R7 are accessible by all instructions that specify a general-purpose register.

High registers

Registers R8-R12 are accessible by all 32-bit instructions that specify a general-purpose register. Registers R8-R12 are not accessible by any 16-bit instructions.

Registers R13, R14, and R15 have the following special functions:

Stack pointer

Register R13 is used as the Stack Pointer (SP). Because the SP ignores writes to bits [1:0], it is auto aligned to a word, four-byte boundary. Handler mode always uses SP_main, but you can configure Thread mode to use either SP_main or SP_process.

Link register

Register R14 is the subroutine Link Register (LR). The LR receives the return address from PC when a Branch and Link (BL) or Branch and Link with Exchange (BLX) instruction is executed. The LR is also used for exception return. At all other times, you can treat R14 as a general-purpose register.

Program counter

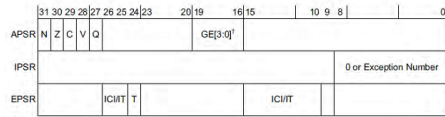
Register R15 is the Program Counter (PC). Bit [0] is always 0, so instructions are always aligned to word or half word boundaries.



19

UNIVERSITY of
Rhode Island

Programmers Model 1



[†] Reserved if the DSP Extension is not implemented

Reserved (see text)

- **APSR** – Application process status register
- **IPSR** – Interrupt process status register
- **ECSR** – Execution process status register



20

UNIVERSITY of
Rhode Island

Programmers Model 2



PRIMASK – Disable all interrupt except NMI and hard fault

FAULTMASK – Disable all interrupt except NMI

BASEPRI – Disable all interrupt of specific priority and lower



21

UNIVERSITY of
Rhode Island

Programmers Model 3

Control Register

nPRIV, bit[0] Defines the execution privilege in Thread mode:

- 0 Thread mode has privileged access
- 1 Thread mode has unprivileged access.

Note
In Handler mode, execution is always privileged.

SPSEL, bit[1] Defines the stack to be used:

- 0 Use SP_main as the current stack
- 1 In Thread mode, use SP_process as the current stack. In Handler mode, this value is reserved.

FPCA, bit[2], if the processor includes the FP extension
Defines whether the FP extension is active in the current context:

- 0 FP extension not active.
- 1 FP extension is active.



22

UNIVERSITY of
Rhode Island