Computer Organization Laboratory

Class Notes

Instructor: Ken Q. Yang

Dept. of ECE, URI

Section 0

Course Objectives, Plans, and Lab Tools

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 µProcessor to Design Systems
- Advanced Topics

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

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:
  - 6 Laboratory experiments,
    - Each lab accounts for 4% of your grade
  - 1 Design project, 20% of your grade
    - Design and documentations, 5% of your grade
    - Project proposal Presentation (5 minutes), 2%
    - Final project presentation (10 minutes), 5%
    - demonstration (10 minutes), 8%
  - Exams
    - 3 exams, each accounts for 20% of your grade
    - Two written exams and one lab exam

Pope’s Inauguration

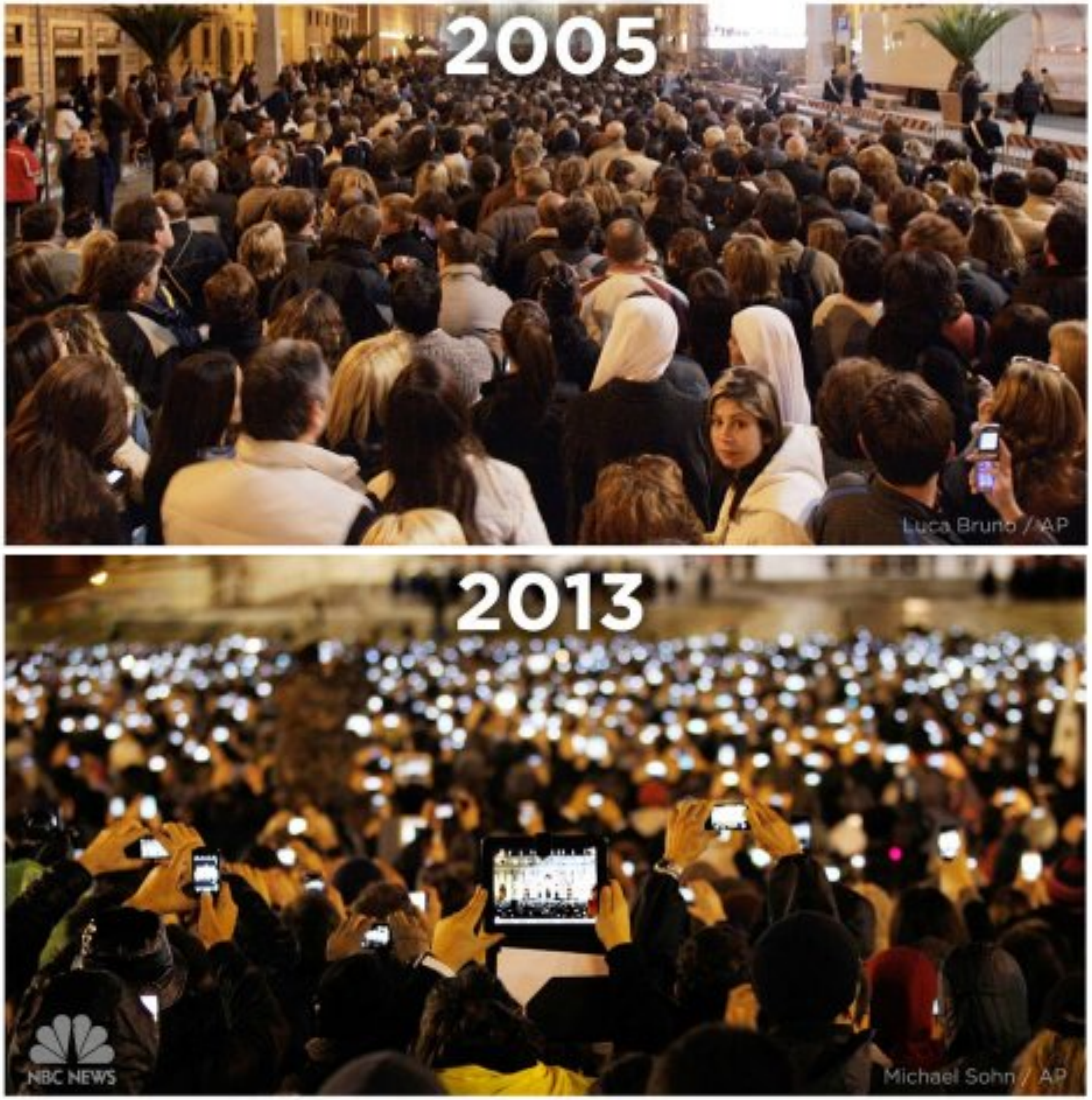
Then…

“When smartphones and tablets

light up the sky,

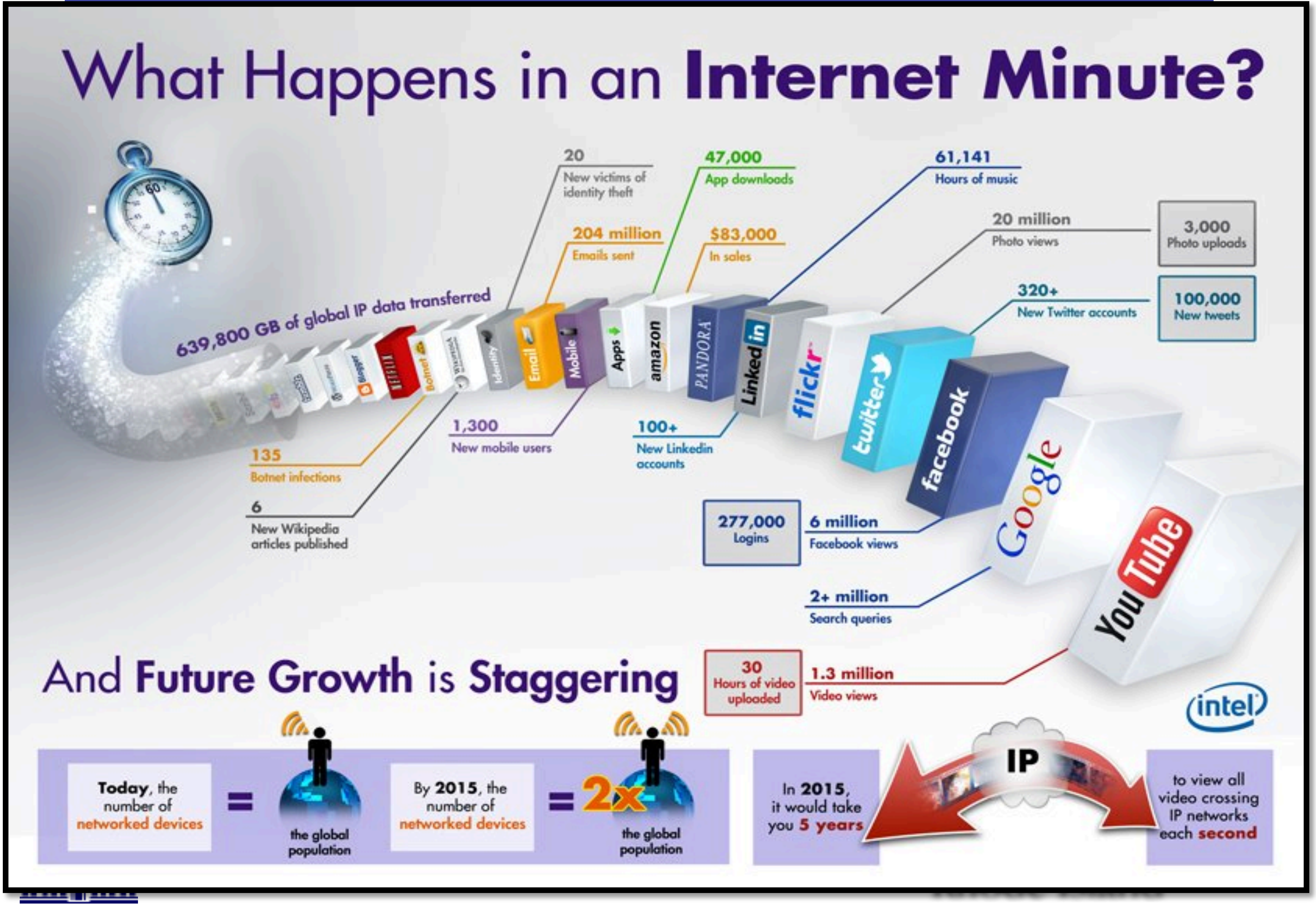
load up the clouds.”

Now…



Source : http://www.alternet.org/speakeasy/alyssa-figueroa/recording-memories-why-must-we-capture-our-every-moment

Era of Internet and Cloud



What Happens in an Internet Minute?

639,800 GB of global IP data transferred

20 New victims of identity theft

204 million Emails sent

47,000 App downloads

$83,000 In sales

61,141 Hours of music

20 million Photo views

3,000 Photo uploads

320+ New Twitter accounts

100,000 New tweets

135 Botnet infections

6 New Wikipedia articles published

1,300 New mobile users

100+ New LinkedIn accounts

277,000 Logins

6 million Facebook views

2+ million Search queries

30 Hours of video uploaded

1.3 million Video views

And Future Growth is Staggering

Today, the number of networked devices = the global population

By **2015**, the number of networked devices = 2x the global population

In **2015**, it would take you **5 years** to view all video crossing IP networks each **second**

*Source: Intel 2012

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 Enhencement

The Cortex™ Processor Family

Cortex™-A

servers

set top boxes

netbooks

mobile applications

Cortex™-R

disk drives

digital cameras

mobile baseband

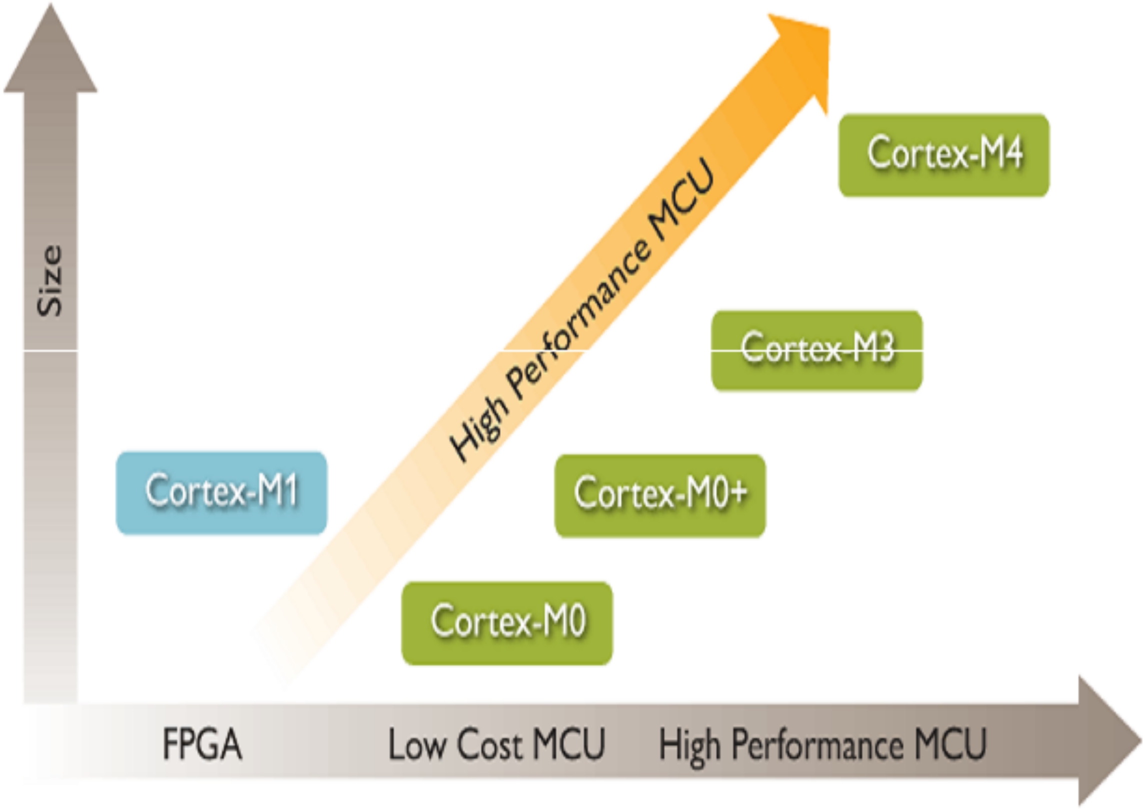
Cortex™-M

appliances

motors

audio

Range of Cortex-M

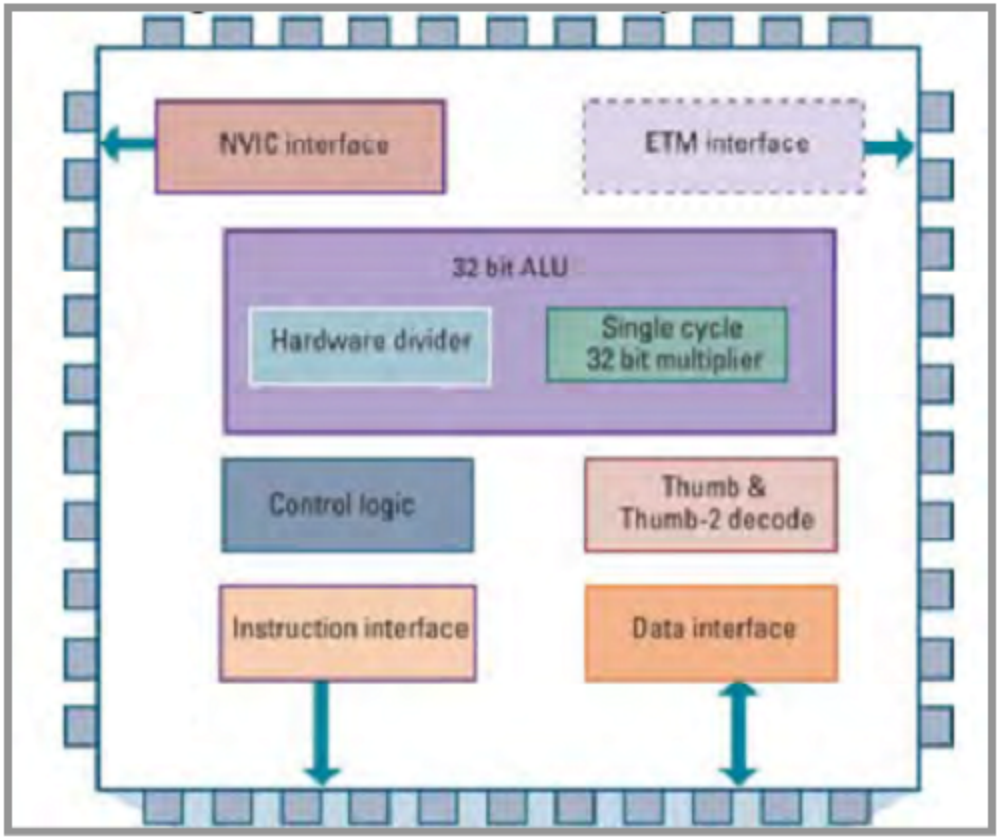


Outline

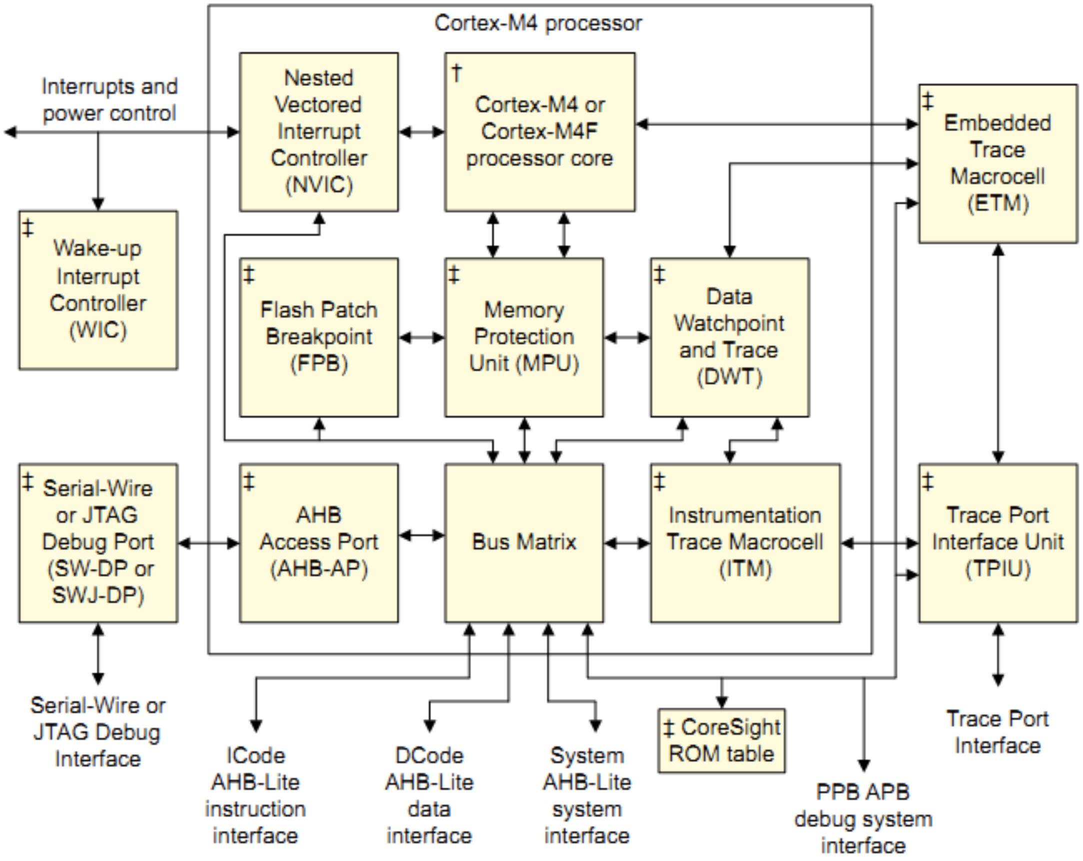
- ARM Cortex™-M Family
- Cortex™-M4 Features
- Kinetis Cortex™-M4 Enhencement

What is Cortex™-M

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



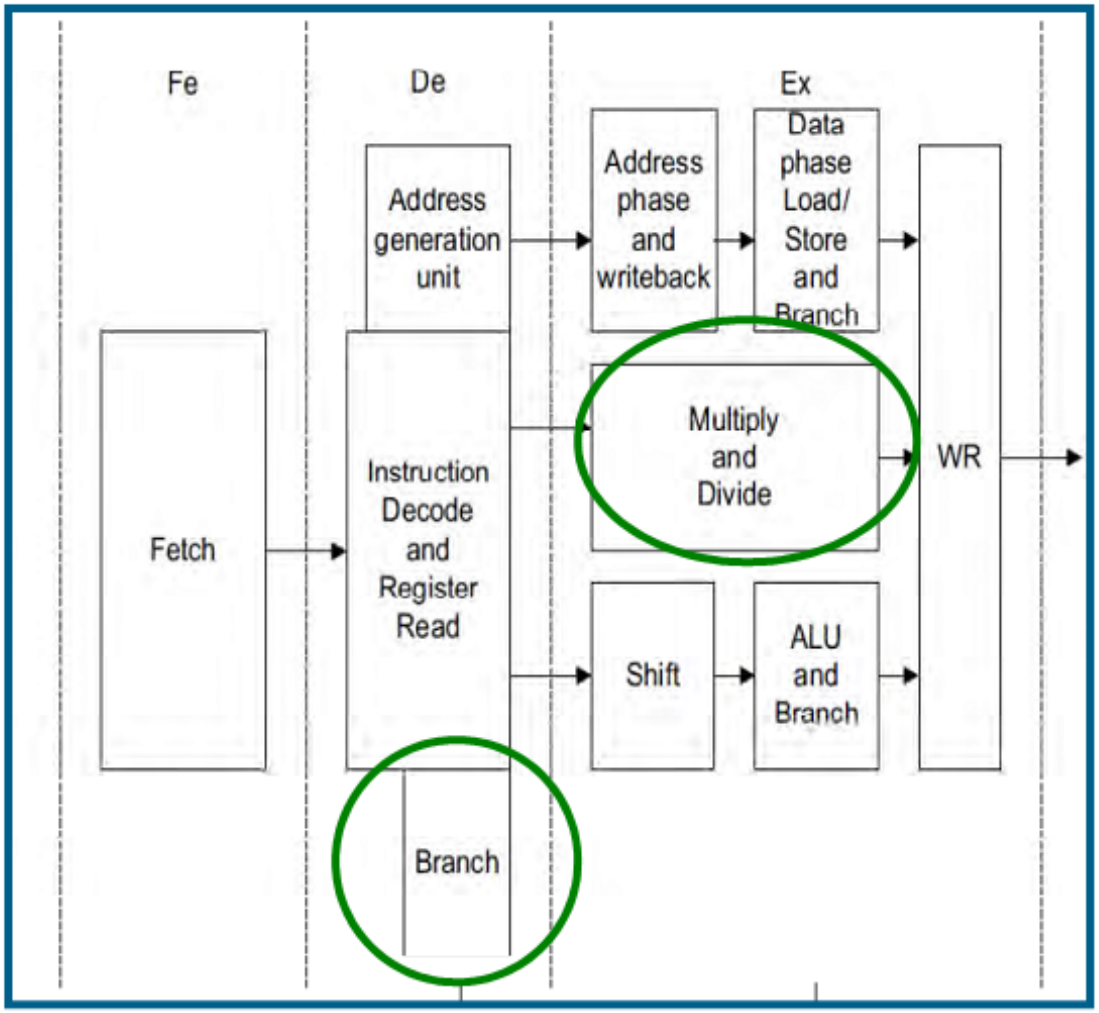
CORTEX M-4


Cortex-M4 processor
Interrupts and power control
Nested Vectored Interrupt Controller (NVIC)
† Cortex-M4 or Cortex-M4F processor core
‡ Embedded Trace Macrocell (ETM)
‡ Wake-up Interrupt Controller (WIC)
‡ Flash Patch Breakpoint (FPB)
‡ Memory Protection Unit (MPU)
‡ Data Watchpoint and Trace (DWT)
‡ Serial-Wire or JTAG Debug Port (SW-DP or SWJ-DP)
‡ AHB Access Port (AHB-AP)
Bus Matrix
‡ Instrumentation Trace Macrocell (ITM)
‡ Trace Port Interface Unit (TPIU)
Serial-Wire or JTAG Debug Interface
ICode AHB-Lite instruction interface
DCode AHB-Lite data interface
System AHB-Lite system interface
‡ CoreSight ROM table
PPB APB debug system interface
Trace Port Interface

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

‡ Optional component

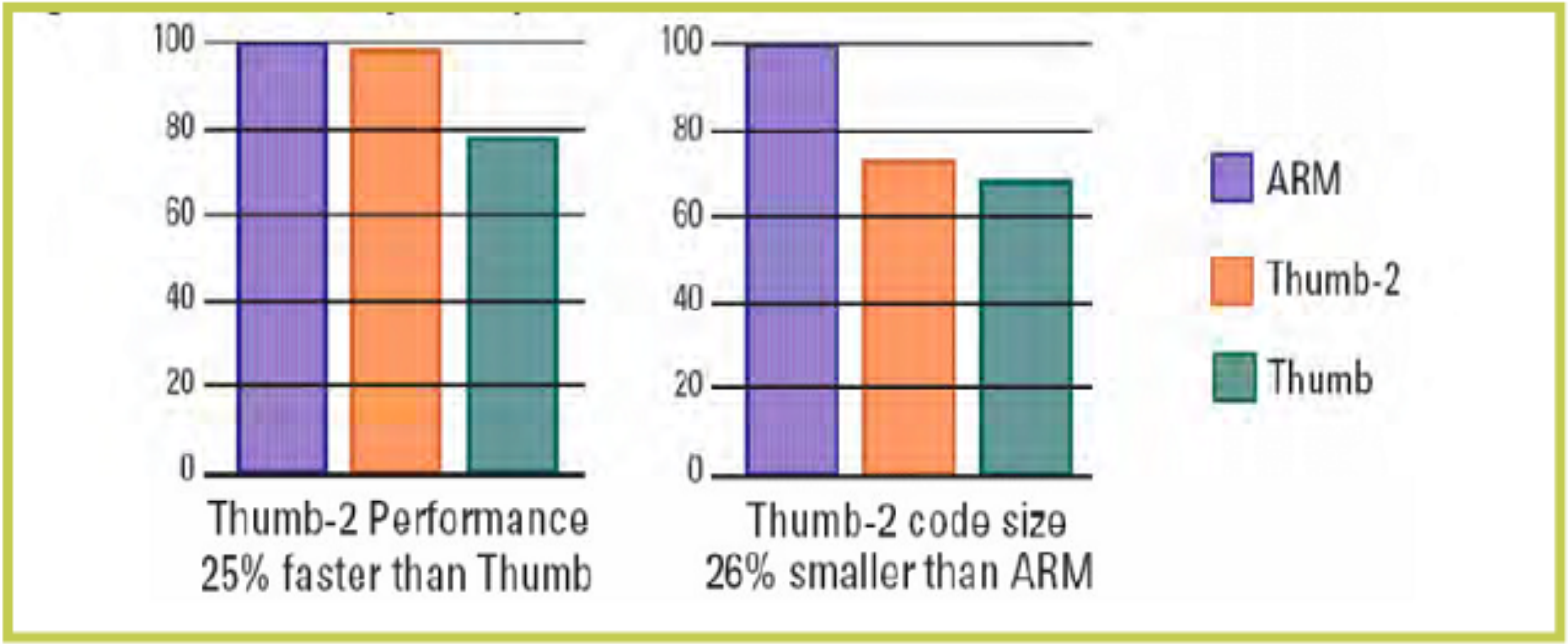
Cortex™-M Pipeline


Fe
De
Ex
Address generation unit
Address phase and writeback
Data phase Load/ Store and Branch
Fetch
Instruction Decode and Register Read
Multiply and Divide
WR
Shift
ALU and Branch
Branch

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

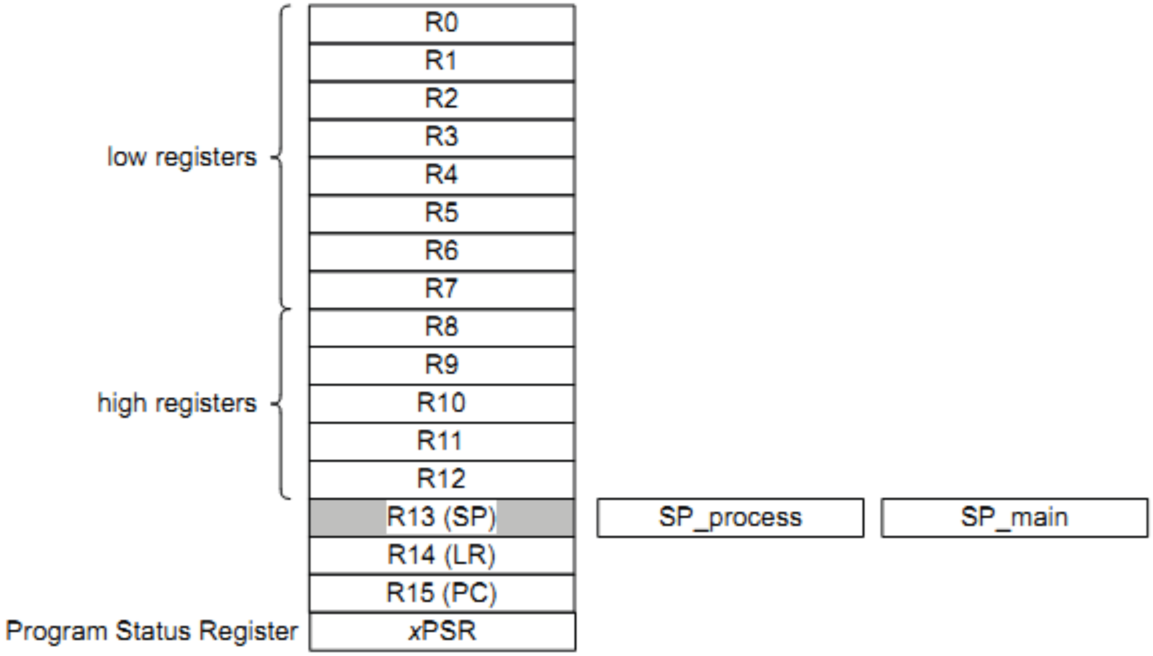
Effects of the Thumb-2 ISA



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).



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.

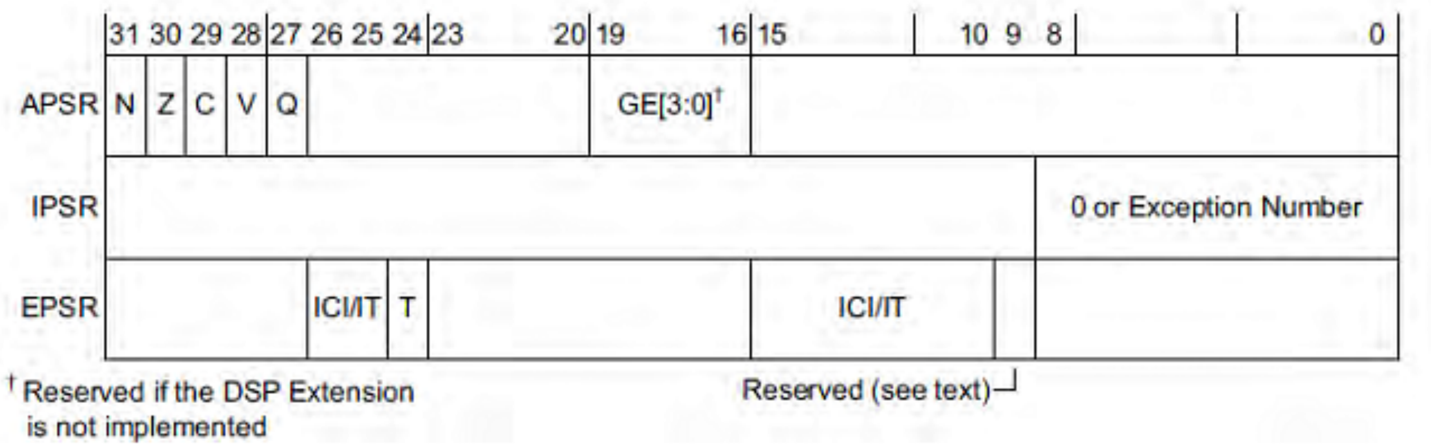
At all other times, you can treat R14 as a general-purpose register.

Program counter

Register R15 is the Program Counter (PC).

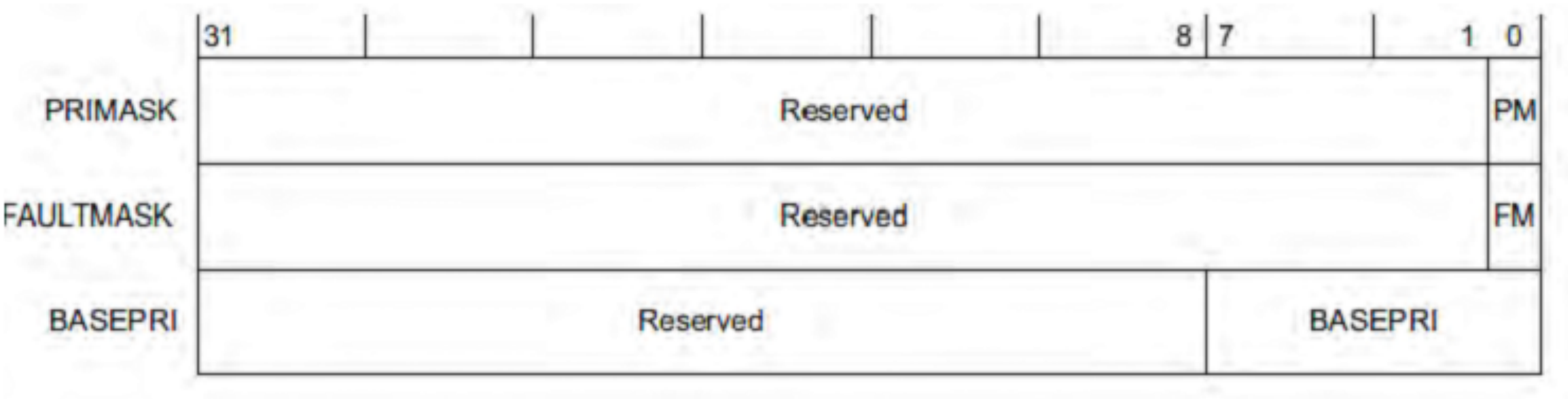
Bit [0] is always 0, so instructions are always aligned to word or half word boundaries.

Programmers Model 1



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

Programmers Model 2



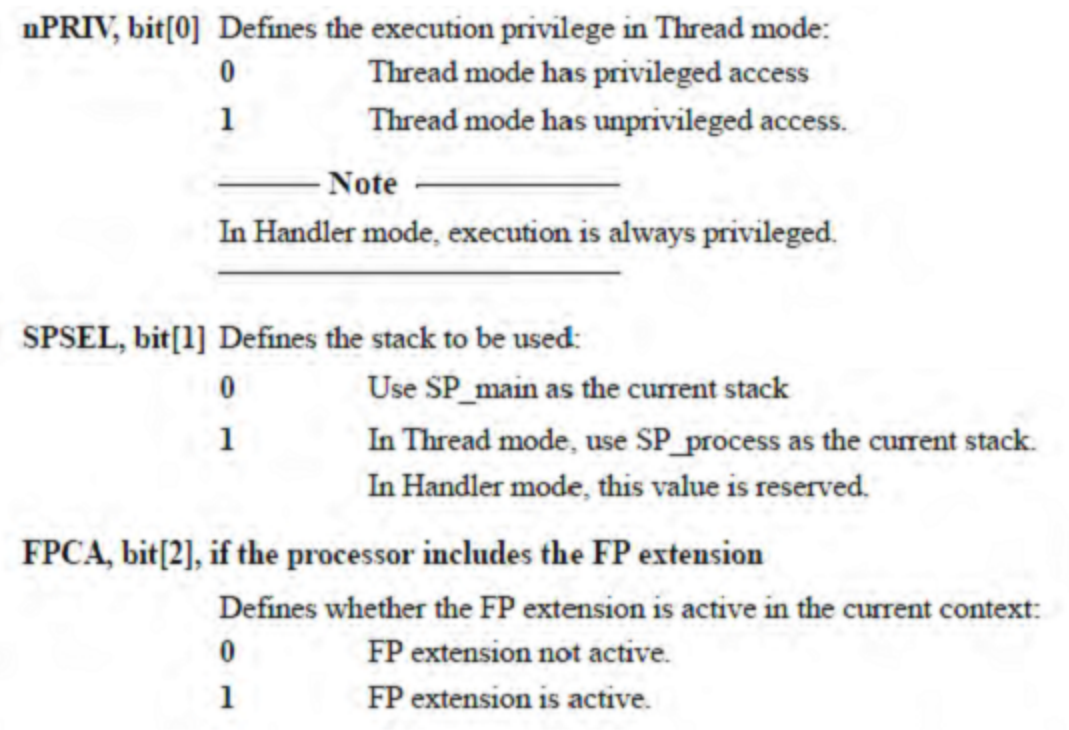
PRIMASK – Disable all interrupt except NMI and hard fault

FAULTMASK – Disable all interrupt except NMI

BASEPRI – Disable all interrupt of specific priority and lower

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.