Computer Architecture – An Introduction

CS2053 Computer Architecture

Computer Science & Engineering

University of Moratuwa

by

Dr. Sulochana Sooriyaarachchi &

Dr. Chathuranga Hettiarachchi

Acknowledgement: Dr. Dilum Bandara

Notice

- Lecture time: Friday 8.15 -10.15 am
- Lab time: Mon 1.15 – 3.15 pm
- Lab classes
  - Student grouping - 40 per group
  - Access
- Instructors:
  - Batch 19

From Outside

Output

Processor
Memory
Hard disk
DVD ROM
Graphics card
Ethernet

Input

Source: techwench.com

From Outside (Cont.)

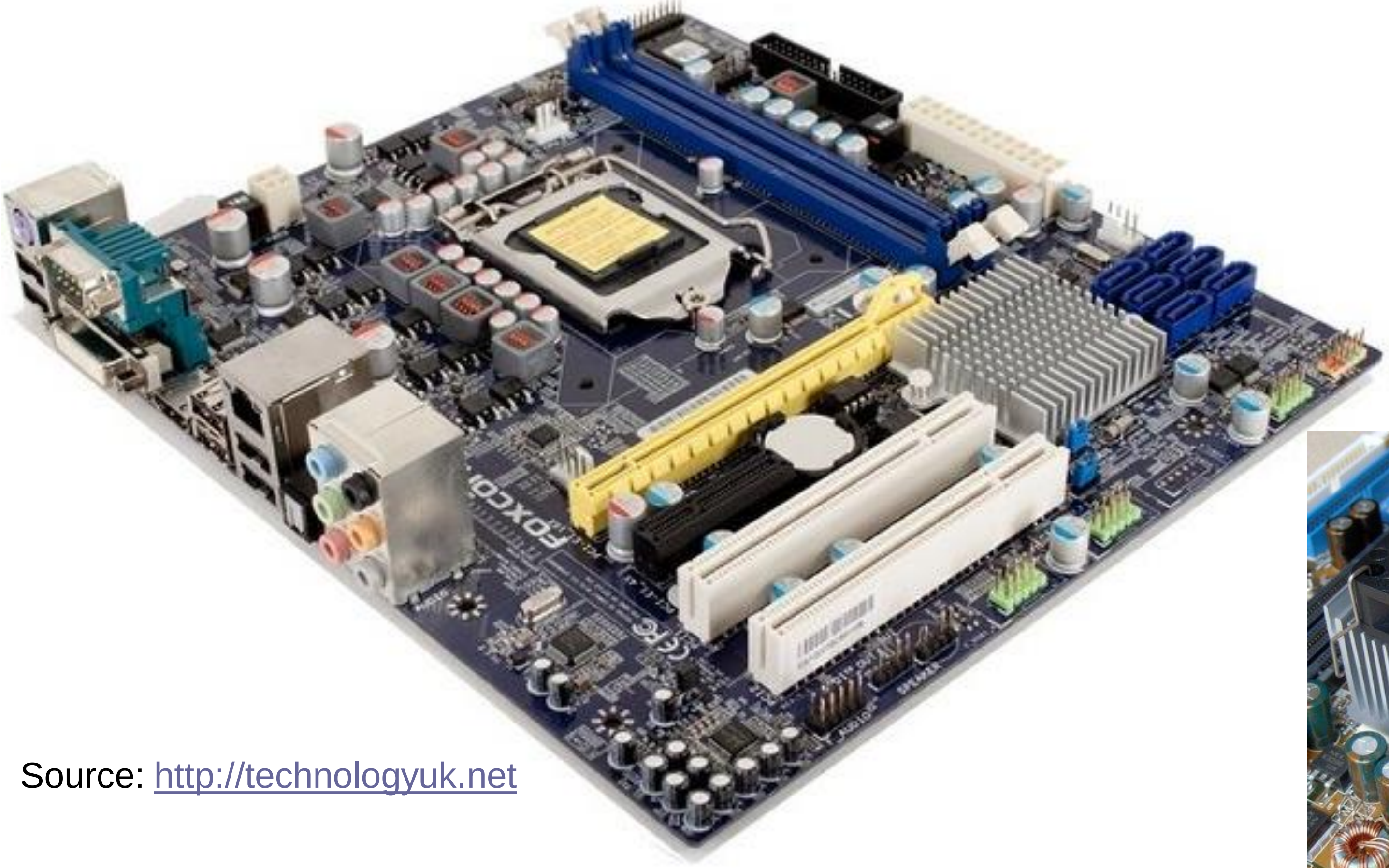
Source: Amazon.com

Touch pad
Touch screen
Wireless
Weight

Screen size
Battery capacity
Weight
Camera
Sensors

Source: Daniel Zanetti, wikimedia.org

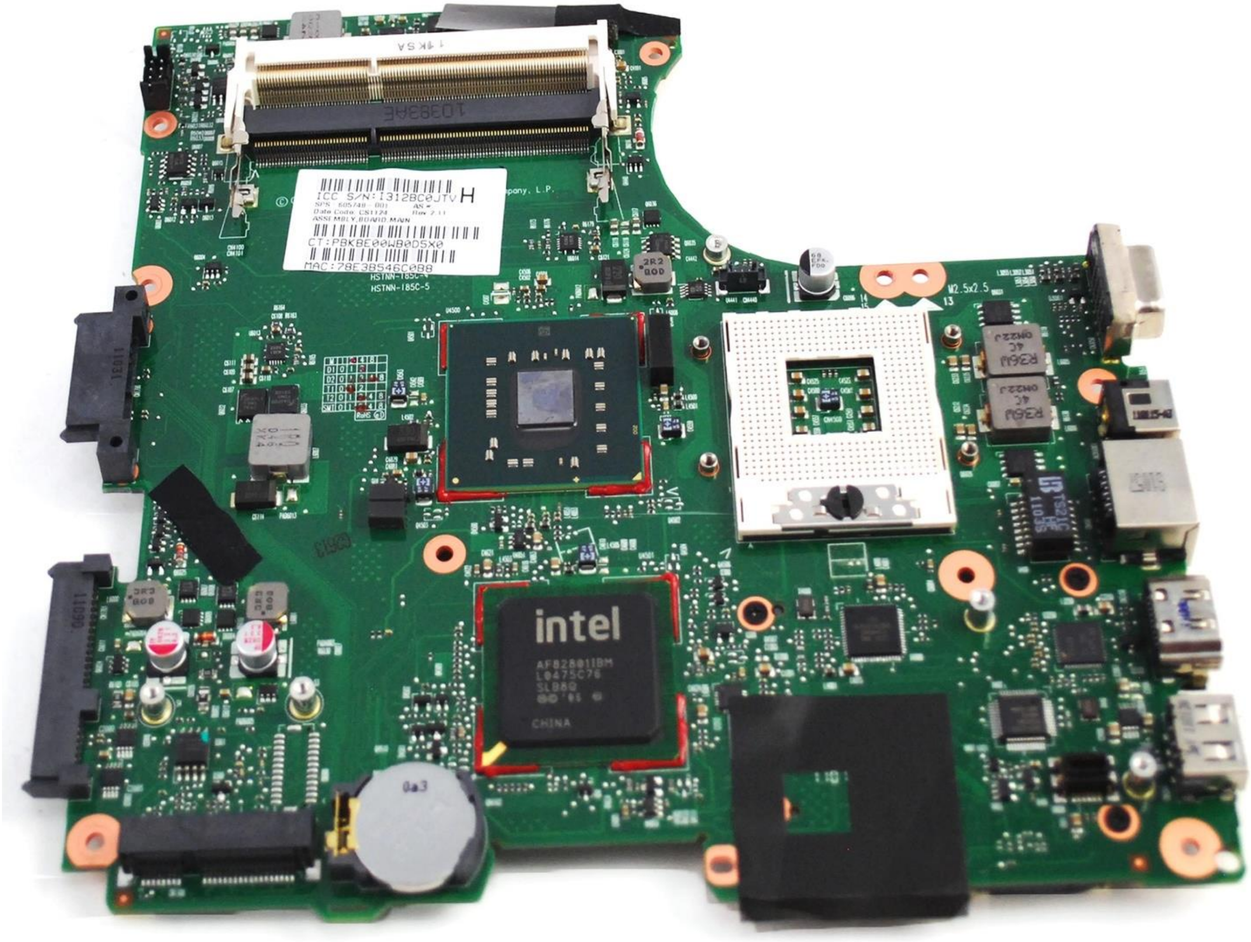
From Inside



Source: http://technologyuk.net

Source: http://rays-place.net

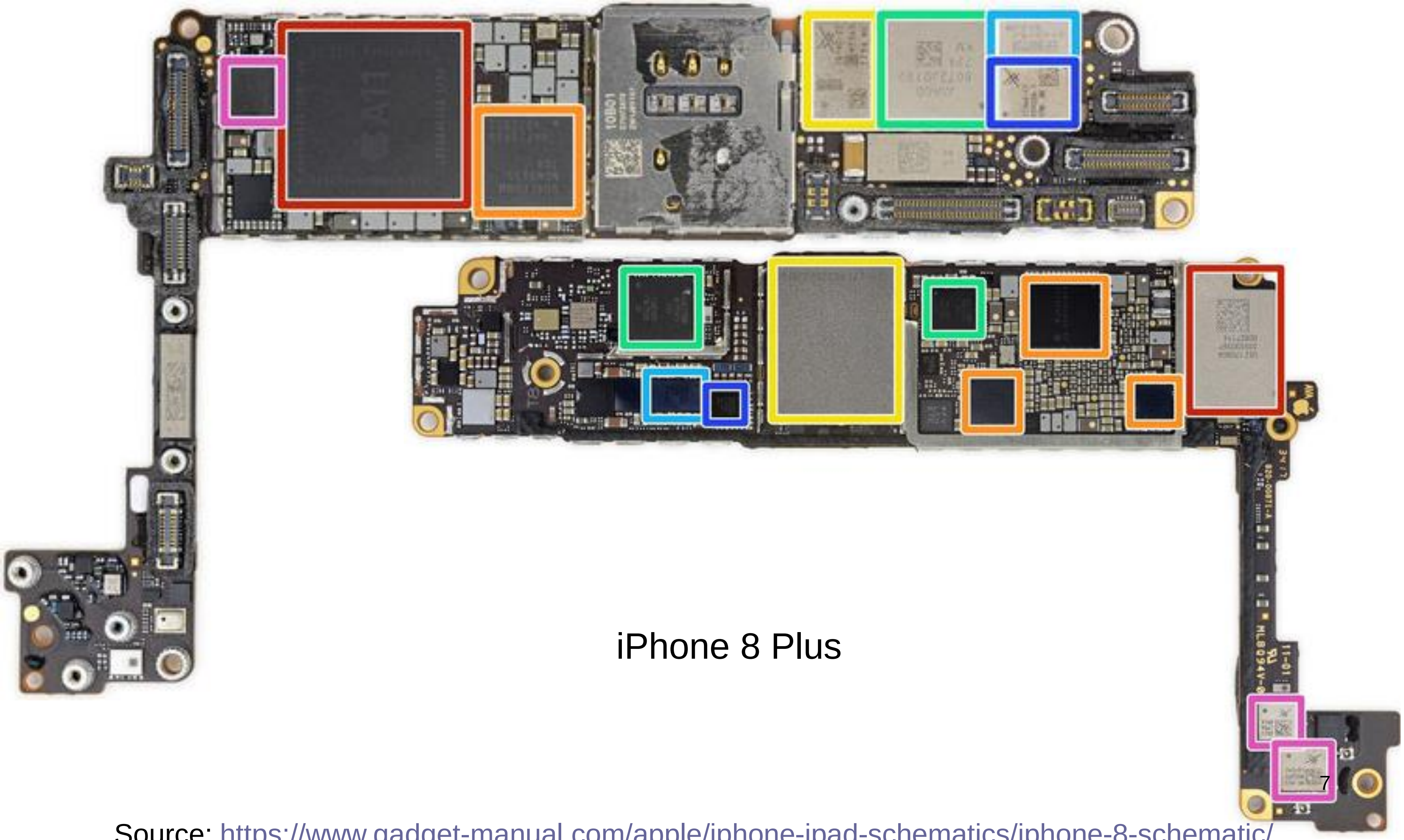
From Inside (Cont.)



Source: www.laptopaid.com

Source: http://techgoesboom.com

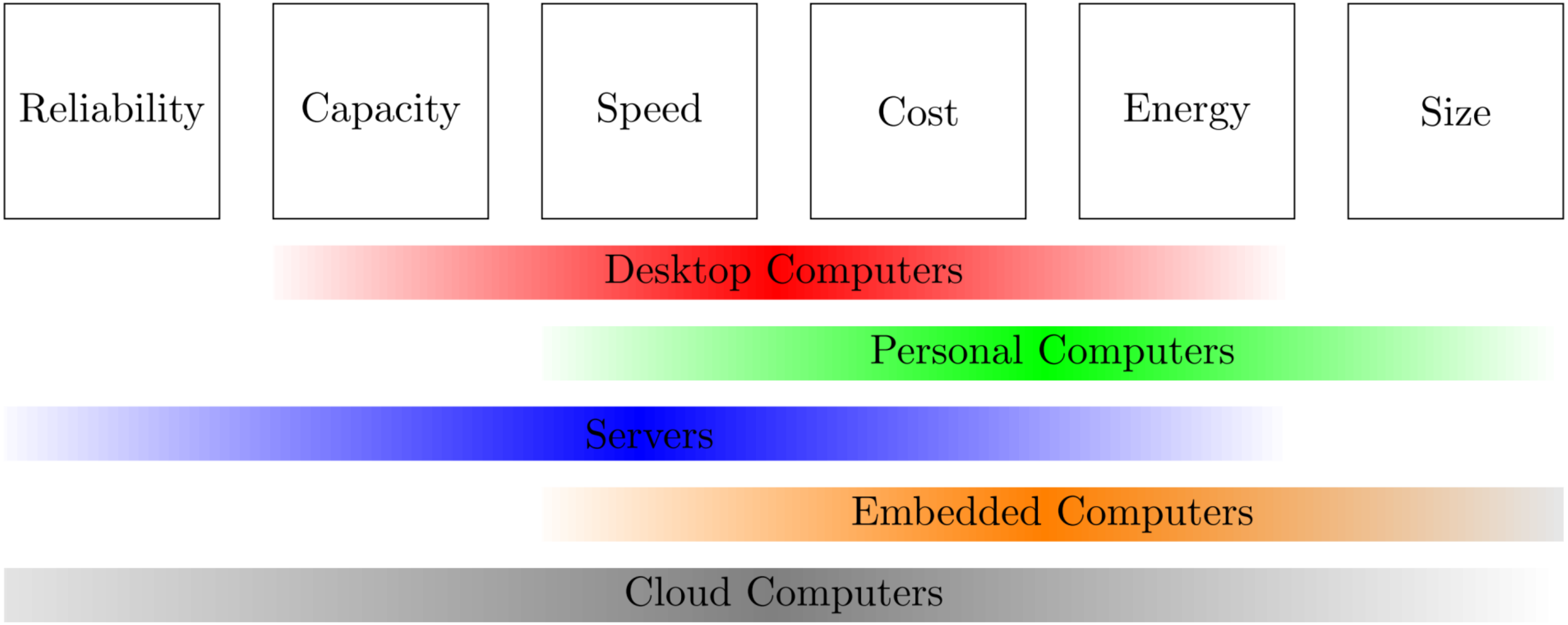
From Inside (Cont.)



iPhone 8 Plus

Source: https://www.gadget-manual.com/apple/iphone-ipad-schematics/iphone-8-schematic/

Classes of Computers & Performance Metrics



Want to achieve these performance metrics?

Then you need to understand & design based on principles of computer architecture

What We Are Going To Study?

- How these internal components look like?
  - Top-down approach with schematics
- How do they fit together?
- How to program them?
- How to benefit from performance enhancement options?

- Focus on abstract views using schematic diagrams
- Not on how those are built using semiconductors

Terminology

Computer Architecture

Blueprint/plan that is visible to programmer

Key functional units, their interconnection, & instruction to program

Instruction Set Architecture (ISA)

e.g., x86 vs. ARM

Computer Organization

Internal details of operational units, their interconnection, & control

View of a computer designer

How to support multiplication – multiply circuit or repeated addition

e.g., Intel & AMD both support x86 with different organizations

Computer Design

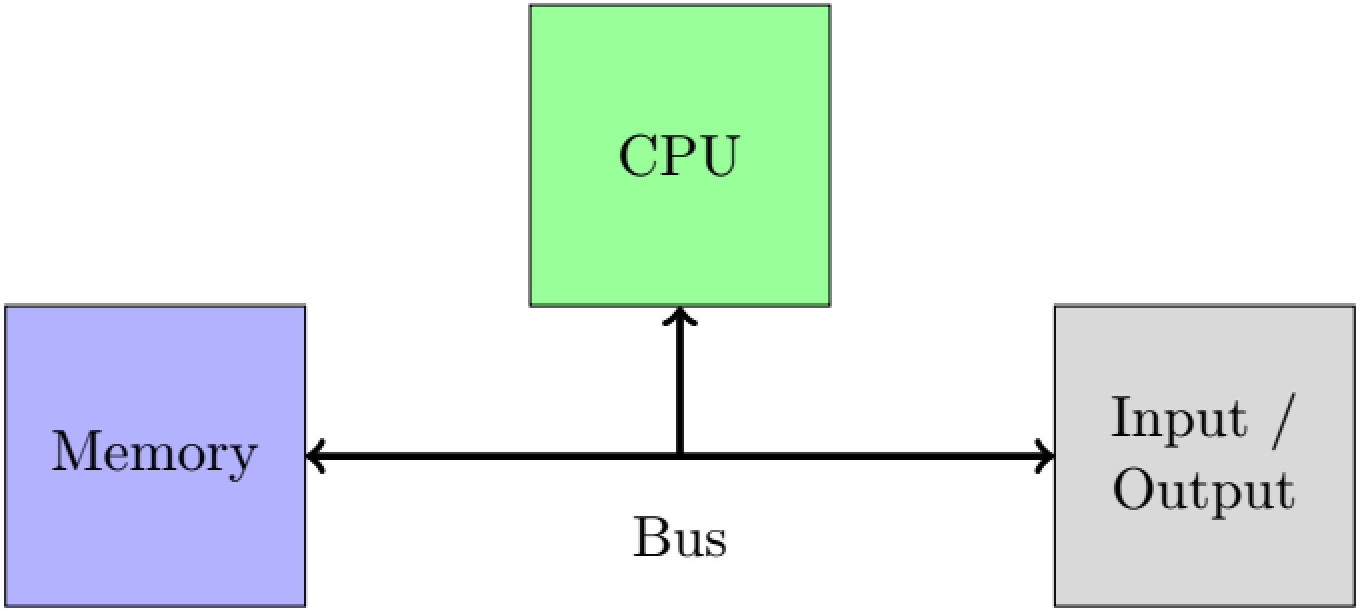
Maps a given organization to a logic design, logic design to a Silicon layout, & chip packaging

View of hardware designer

Design decisions based on constrains like circuit-level delays, Silicon real estate, heat generation, & cost

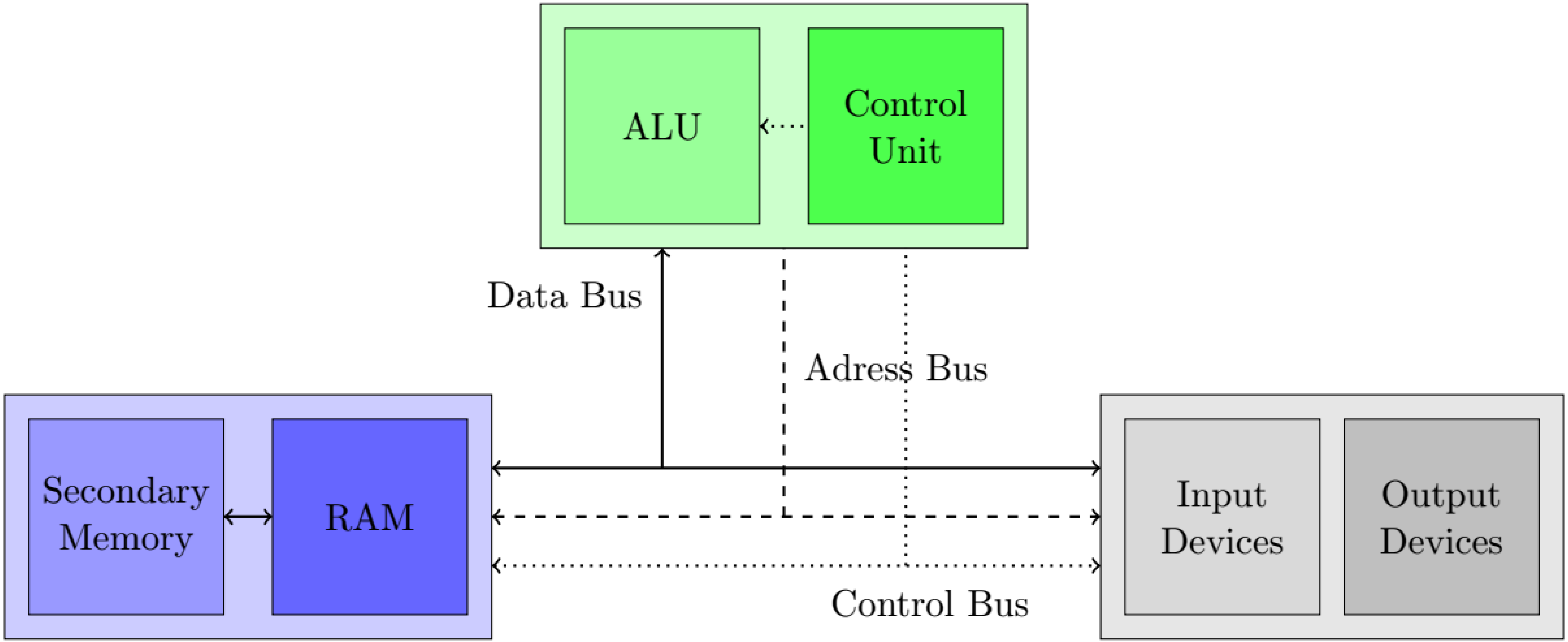
e.g., Intel Core i7-6800K vs. Xeon E5-2643 v4

High-Level View of a Computer

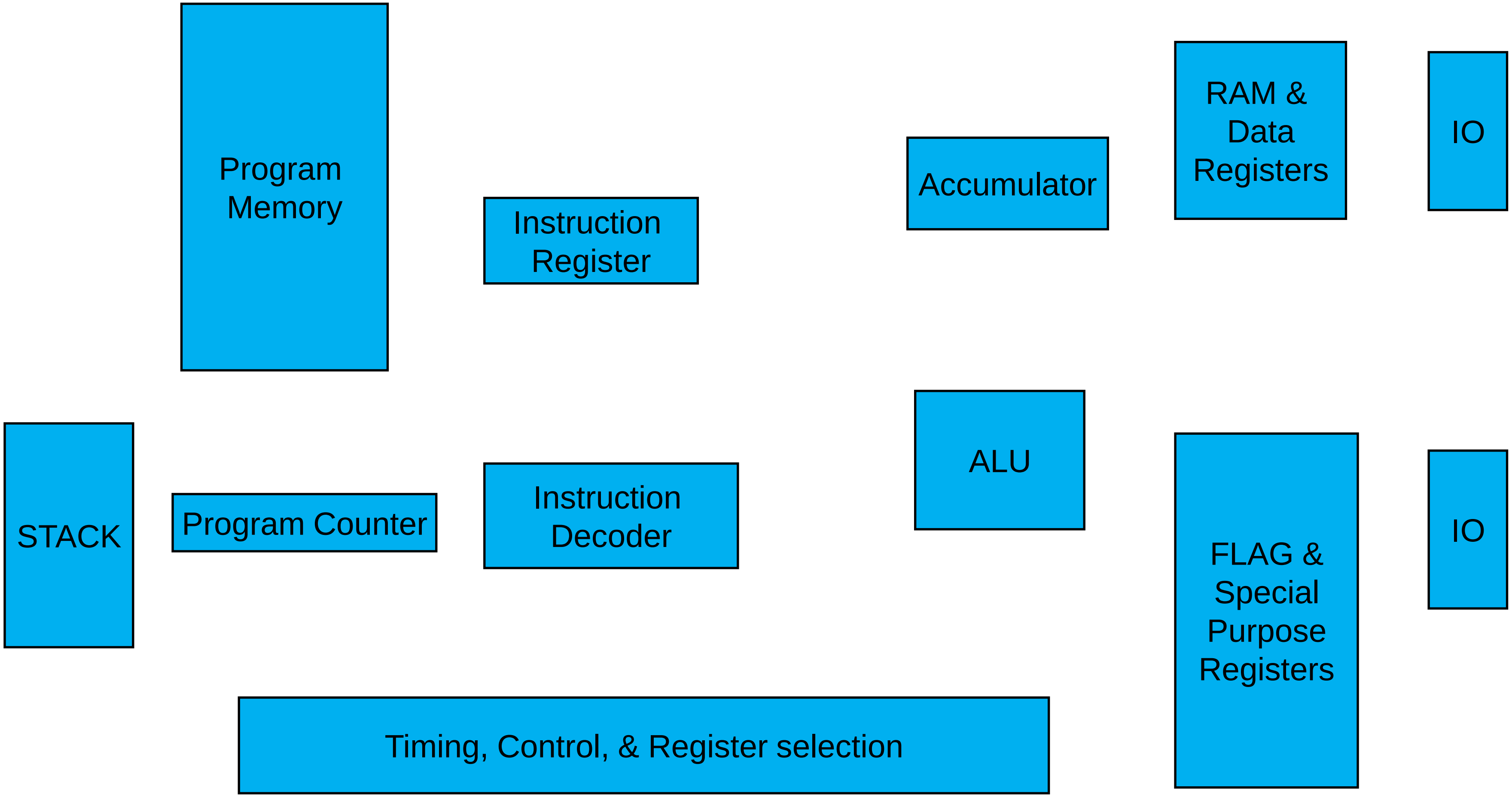


- CPU – execute instructions
- Memory – store program & data
- IO devices – receive inputs & produce outputs
- Bus – interconnects everything by transferring data

2^{nd}-Level View of a Computer

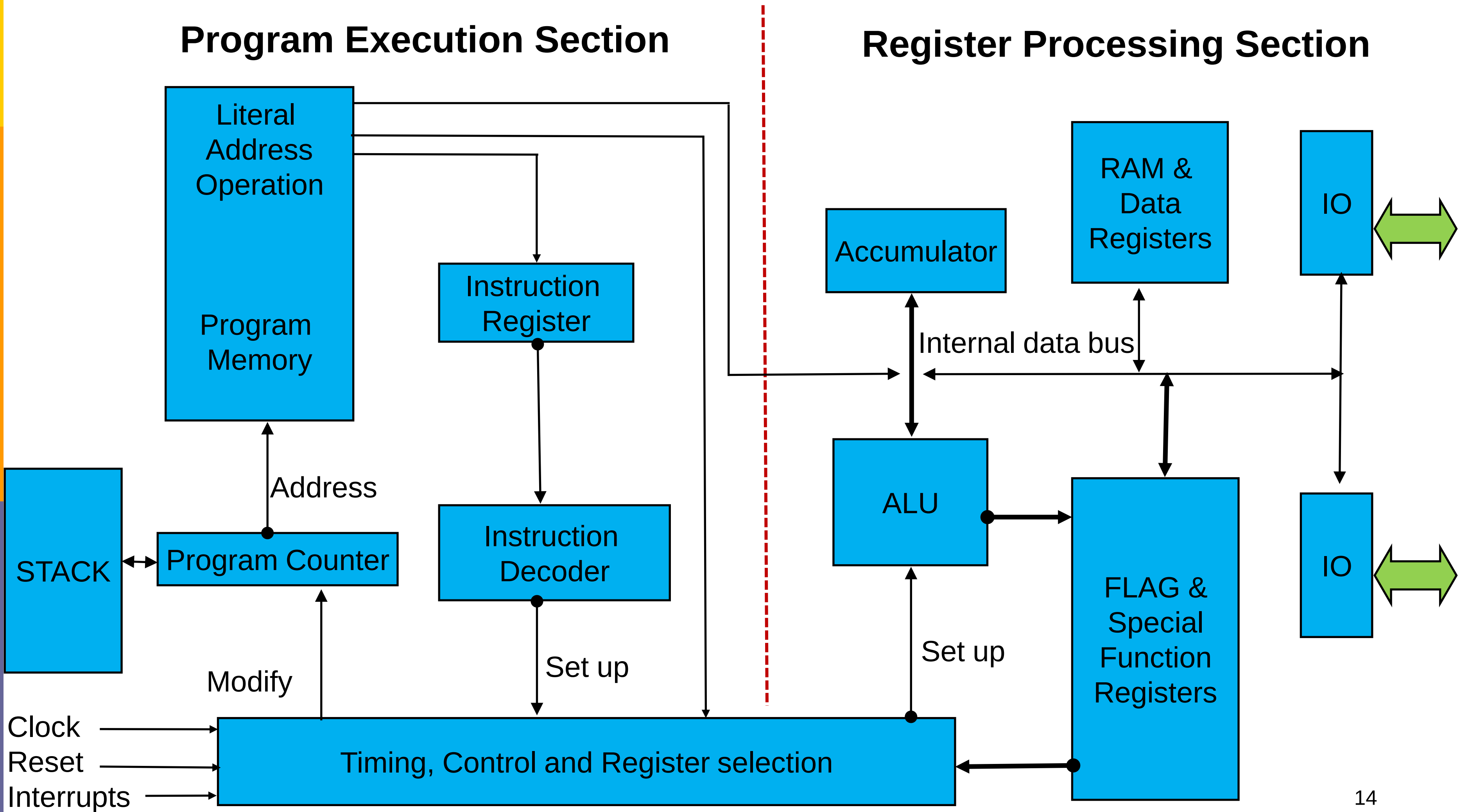


3rd-Level View of a Computer



Source: Makis Malliris & Sabir Ghauri, UWE

Blocks of a Microprocessor (Cont.)



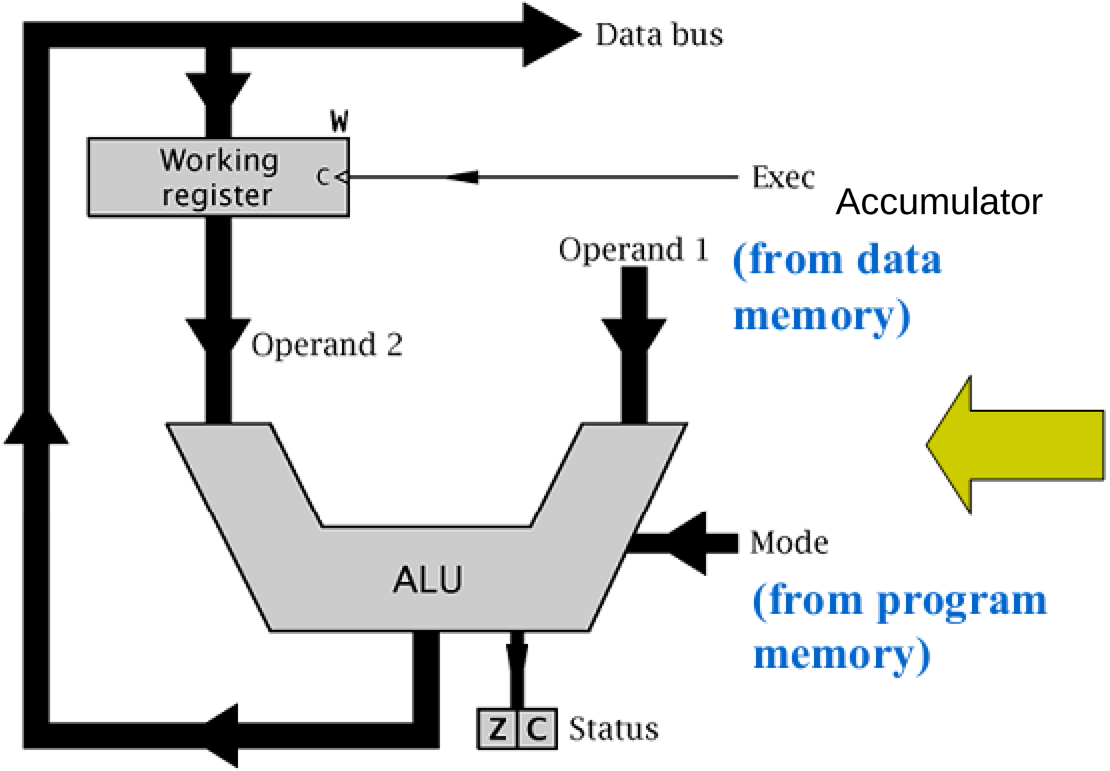
Source: Makis Malliris & Sabir Ghauri, UWE

Question(s)

- Which of the following points to the memory address of the next instruction to be executed?
  a) Program Counter (PC)
  b) Instruction Register (IR)
  c) STATUS register
  d) Accumulator (A)

- ________________ interprets an instruction.

Arithmetic & Logic Unit (ALU)



Accumulator

(from data memory)

(from program memory)

Source: Introduction to PIC Microcontroller – Part 1 by Khan Wahid

- Data processing unit
- Arithmetic unit
  - Performs arithmetic operations
- Logic unit
  - Performs logic operations

Registers

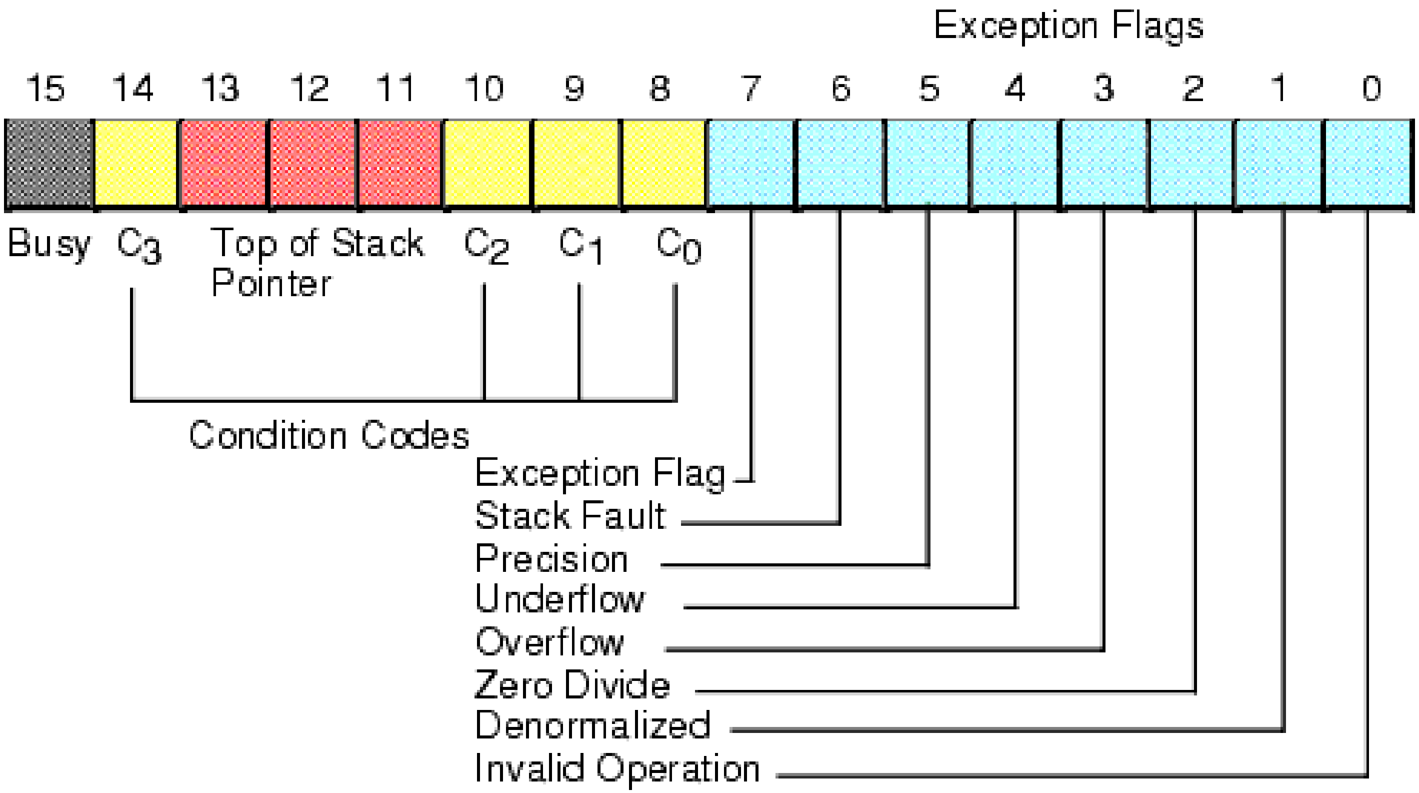
- Type of memory located inside CPU
- Can hold a single piece of data
  - Useful in both data processing & control functionalities
- Special purpose registers
  - Program Counter (PC)
  - Instruction Register (IR)
  - Accumulator or working register
  - Flag/Status register
- General purpose registers
  - Used to store data

Special Purpose Registers

Register	Function
Accumulator (A) / Working Register (W)	Results of arithmetic & logic operations always go to accumulator Connected directly to output of ALU
Program Counter (PC)	Used to keep track of memory address of next instruction to be executed When instructions are *fetched*, instruction pointed by PC is fetched into CPU Once the instruction is fetched, PC is updated to point to next instruction, i.e., $\text{PC} = \text{PC} + d$
Instruction Register (IR)	Once fetched, instructions are stored in IR for execution Located closely to control unit, which decodes the instruction

FLAG/STATUS Register

- Individual bits indicate status of ALU operations



Source: www.plantation-productions.com/Webster/www.artofasm.com/Linux/HTML/RealArithmetic.html

THANK YOU