Computer Fundamentals

Dr. Safdar Nawaz Khan Marwat
DCSE, UET Peshawar

Lecture 8

safdar@uetpeshawar.edu.pk

Microprocessors

- Microprocessor acts as "brain"
- Key element of all computers
- Provides mathematical and decision making ability
- Current microprocessors contain complex circuits
  - Pentium
  - Athlon
  - SPARC
  - PowerPC
- Consist of tens of millions of transistors
- Operate at ultra-fast speeds
  - Over a billion operations per second

Source: *https://www.elprocus.com/microprocessor-history-and-brief-information-about-its-generations*

Microprocessor (cont.)

- Microprocessors are powerful but require other components
- Human brain needs other body parts to be useful
  - Hands
  - Feet
  - Eyes
  - Ears
  - Mouth
- Microprocessor system requires
  - …

Integrated Circuits

- Commonly known as IC or chip
- Tiny piece of silicon with several electronic parts
- Silicon occupies very little IC volume
  - Most of the size due to pins and packaging
- Components of IC are very small
  - Smaller than thickness of a human hair

Source: *http://simple.wikipedia.org/wiki/Integrated_circuit*

Integrated Circuits (cont.)

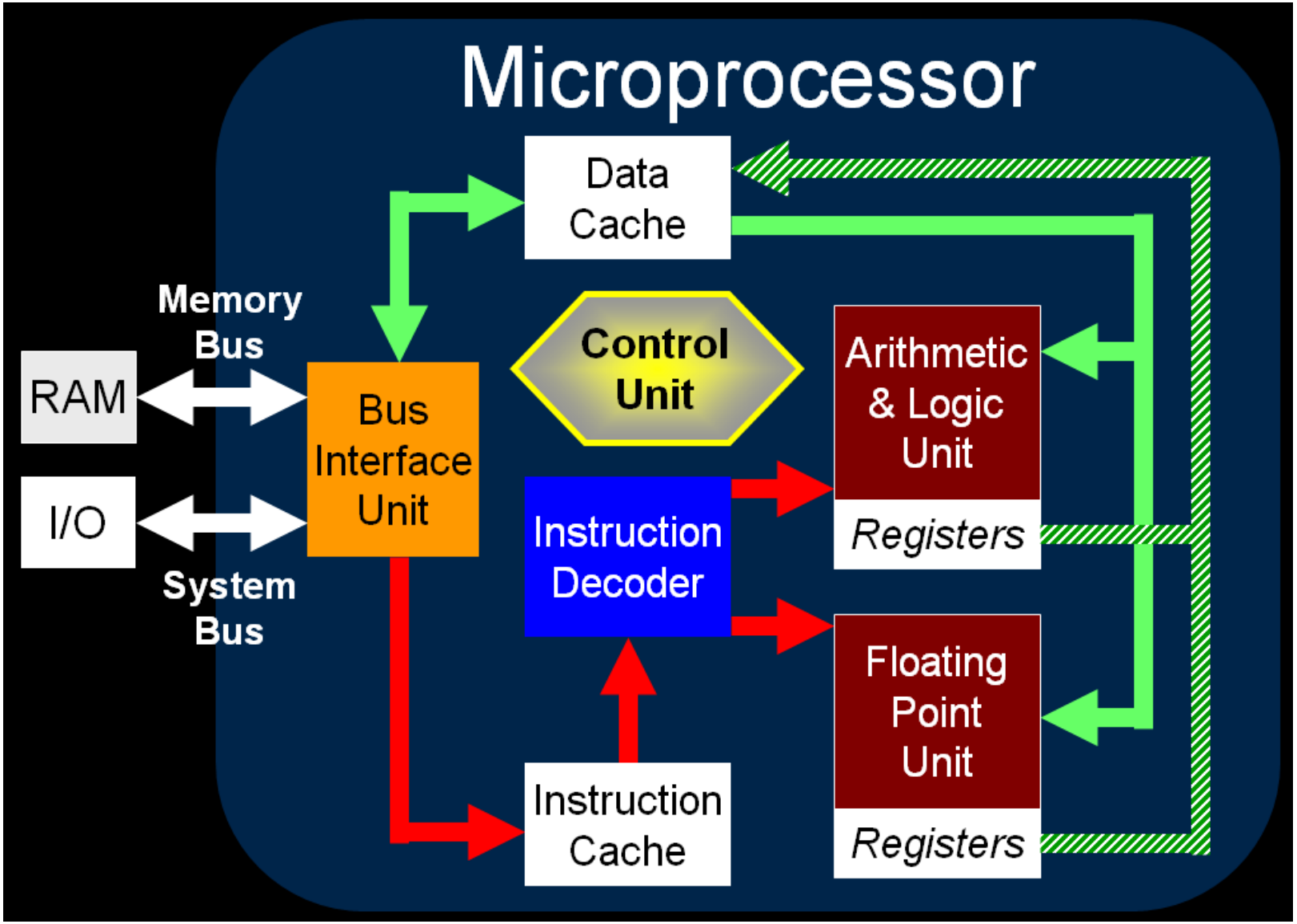
- Devices in ICs
  - Transistors
  - Diodes
  - Resistors
  - Capacitors
  - Wires

- And are made of the following material
  - Silicon - semiconductor
  - Copper - conductor
  - Silicon Dioxide - insulator

Microprocessors (cont.)

- Also known as CPU
  - Central Processing Unit
  - Control Unit
    - Controls resources in computer
    - Instruction set
  - Arithmetic and Logic Unit
    - Simple math operations
    - Integer unit
    - Have own registers
  - Floating-Point Unit
    - Also simple math operations
    - Numeric unit
    - Also have own registers

Building Blocks of Microprocessor



Bus Interface Unit

- Receives instructions and data from main memory
- Instructions sent to instruction cache
- Data sent to data cache
- Also receives processed data and sends to main memory

Instruction Decoder

- Receives programming instructions
  - Reads the next instruction in from memory
- Decodes instructions for understanding at processing units
  - Sends component pieces of instruction to destinations
- Passes on decoded instruction to ALU or FPU

Arithmetic and Logic Unit (ALU)

- Also known as "Integer Unit"
- Performs whole-number math
  - Calculations (subtract, multiply, divide etc.)
  - Comparisons (is greater than, is smaller than, etc.)
  - Logical operations (NOT, OR, AND, etc.)
- Latest microprocessors have two identical ALU's
  - Perform calculations simultaneously
  - Double the capability

Arithmetic & Logic Unit (ALU) (cont.)

Operations Performed by the Arithmetic Logic Unit

Arithmetic Operations		Logical Operations	
+	add	=, ≠	equal to, not equal to
−	subtract	>, ≯	greater than, not greater than
×	multiply	<, ≮	less than, not less than
÷	divide	≥, ≱	greater than or equal to, not greater than or equal to
^	raise by a power	≤, ≰	less than or equal to, not less than or equal to

Floating-Point Unit (FPU)

- Also known as the "Numeric Unit"
- Performs calculations involving numbers with scientific notation
  - Floating-point numbers
- Represents extremely small and extremely large numbers
  - In compact form
- Floating-point calculations are required for working in
  - Graphics
  - Engineering
  - Science and research
- ALU can do these calculations as well
  - But very slowly

Registers

- ALU and FPU have private memory called registers
- Register also known as accumulator
  - Small amount but **super-fast**
  - Placed right next to ALU/FPU
    - For exclusive use
- Storage of intermediate and final results from calculations in registers
- Processed data goes back to data cache from registers
- Data cache sends data to main memory

Control Unit

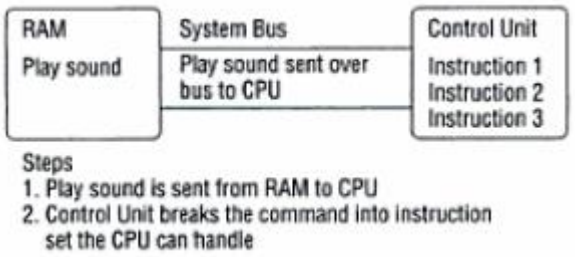
- Manages the whole microprocessor
- Directs operation of processor
- Tells the memory, ALU and I/O devices how to respond to program's instructions
  - Hold instruction set
- Tasks include
  - Fetching instructions and data
  - Storing data
  - Managing input/output devices

Instruction Set

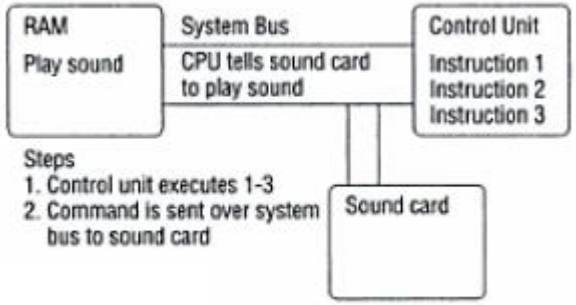
- Set of machine instructions
  - Recognizable at microprocessor
  - Executable at microprocessor
  - The only language microprocessor knows
- An instruction set includes low-level, a single step-at-a-time instructions
  - E.g. add, subtract, multiply, divide
- Each microprocessor family has its unique instruction set
- Bigger instruction-sets mean more complex chips
  - Higher costs
  - Reduced efficiency
  - Shorter programs

Machine Cycles

- Steps by CPU to process data
- Instruction cycle
  - CPU gets the instruction
    - Fetch
    - Decode

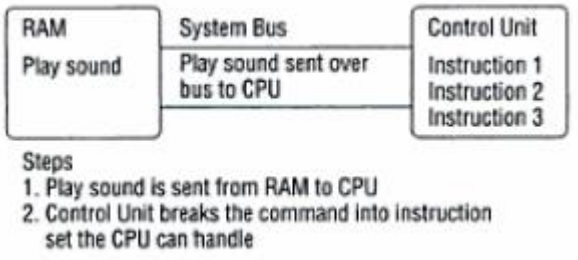


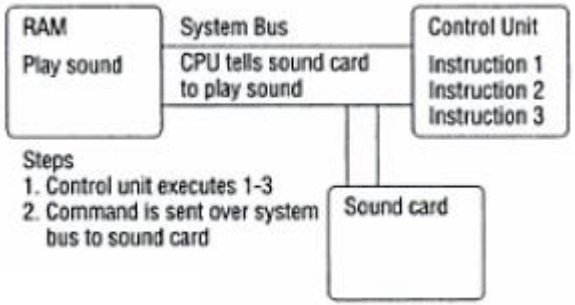
- Execution cycle
  - CPU performs the instruction
    - Execute
    - Store



Machine Cycles (cont.)

- Fetching
  - CU must fetch a command or data from computer's memory.
- Decoding
  - CU break down the commands into instructions
- Executing
  - CPU carries out the instructions in order
- Storing
  - Store results in memory (not always required)





Machine Cycles (cont.)

- Billions of cycles per second
- Pipelining processes more data
  - Begins fetching new instruction before current cycle completed
  - 20 instructions simultaneously
  - Introduced with Intel Pentium IV

Instr. No.	Pipeline Stage						
1	Fetch	Decode	Execute	Store			
2		Fetch	Decode	Execute	Store		
3			Fetch	Decode	Execute	Store	
4				Fetch	Decode	Execute	Store
5					Fetch	Decode	Execute
Clock Cycle	**1**	**2**	**3**	**4**	**5**	**6**	**7**

Source: *https://en.wikipedia.org/wiki/Instruction_pipelining*

Machine Cycles (cont.)

- Multitasking allows multiple instructions
  - CPU makes threads of programs (thread = instruction)
  - Execute one thread from a program at a time
  - User thinks as if each program is being run simultaneously
- Hyperthreading allows execution of several threads at once
  - Multiple threads executed at one time
  - Processor can execute two (rather than one) concurrent streams (or threads) of instructions
    - CPU and simulated CPU
  - Processor uses stall time (waiting for instruction or data fetch from slow memory) and execute another thread

Hyperthreading vs. Standard Modes