

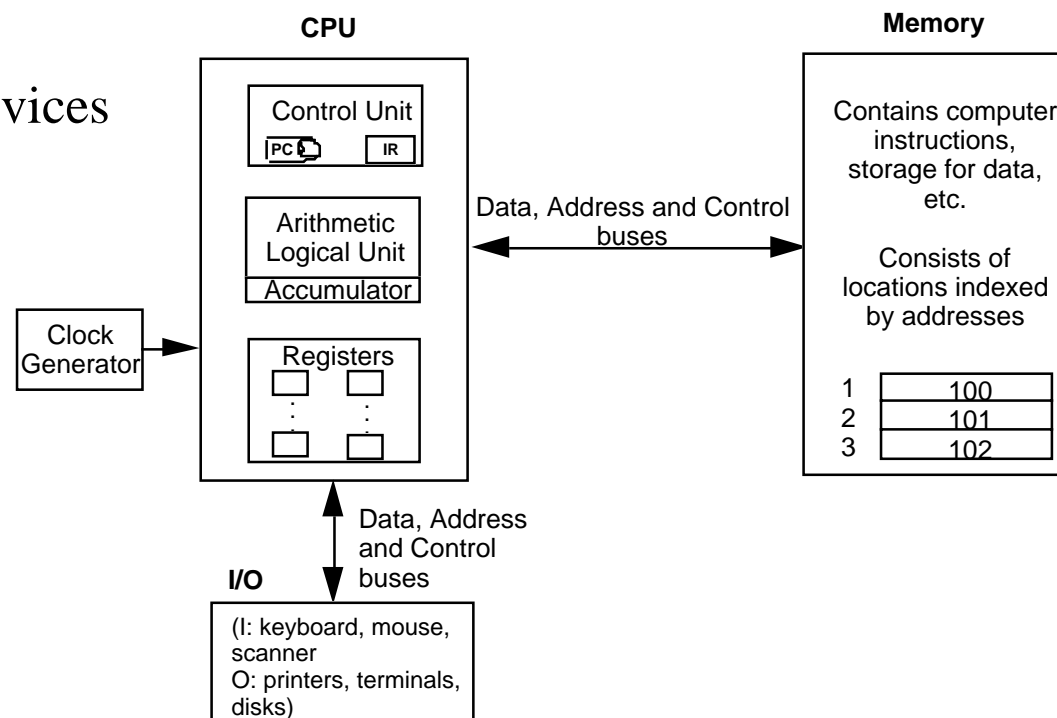
Computer Architecture

2/26/01 Lecture #9 16.070

- On a previous lecture, we discussed the software development process and in particular, the development of a software architecture
- Recall the output of the Conceive phase: **functions**, **concept** and high level **partition** which collectively are known as the architecture
- Today's lecture discusses the architecture of the hardware underlying the software

Computer Architecture

- An overview of how computers are organized
- A digital computer consists of an interconnected system of processors, memories and input/output devices
- A digital computer consists of three basic components
 - Central Processing Unit - "brain" to coordinate activities
 - Memory Unit
 - Input/Output Devices



- **Central Processing Unit**

- The Central Processing Unit (CPU) is the "brain" of the computer
- CPU consists of
 - Control unit which fetches instructions from main memory and determines type
 - Arithmetic and logical unit (ALU) which executes instructions
 - Registers which are high-speed memory to store temporary results
- Function: Execute programs stored in main memory
 - Fetch instructions
 - Examine instructions -- decode
 - Execute instructions

Registers

- Program Counter
 - Holds the address of the memory cell to be fetched
 - "Points" to memory location of next instruction to be executed
 - After Fetch operation, program counter is incremented to point to the next instruction
- Instruction Register
 - Holds the instruction currently being executed
- Accumulator Register
 - Stores result of functions performed by arithmetic logic unit

Memory

- Memory: part of the computer where programs and data are stored
- Memory consists of a number of sequential storage locations
- Each memory location is capable of storing one data element or one computer instruction

Computer memory

Memory location 0
Memory location 1
Memory location 2
Memory location 3
...
...

Memory Structure

- The basic unit of memory is the binary digit, called a bit 0 1
- Number of bits per memory location is computer-dependant
 - IBM 370: 8 bits per cell
 - Dec PDP-8: 12 bits per cell
 - Honeywell 6180: 36 bits per cell
 - CDC Cyber: 60 bits per cell
- A memory location that is 8 bits wide is referred to as a byte

00000000

11111111

00101011

- A memory location that is 16 bits wide is referred to as a word

0000000000000000

1111010100001111

Memory Addresses

- The memory location is the smallest addressable unit
- Each memory location has a unique address, consecutive locations have addresses differing by 1
- For byte-sized memory locations

0 00001111

1 10000001

2 00000000

3 00000000

← 8 bits →

Computer Instructions Store in Memory

- Computer instructions are made up of two parts
 - Operation Code, or op-code
 - Specifies action to be performed by the CPU
 - Operand
 - Specifies source and/or destination of the data acted on by the op-code

E.g., ADD 2

op-code = ADD

operand = 2

function = add 2 to the contents of the accumulator, and store result back into accumulator

- A computer instruction set is list of valid op-codes that a CPU can execute

Memory Types

- Most computer systems have two types of memory
 - Main memory, which contains two kinds of memory
 - RAM - Random Access Memory
 - Used for reading/writing
 - Contents are lost when power is removed
 - ROM - Read Only Memory
 - Contents of ROM not lost when computer is turned off
 - Useful for storing "boot" program
 - Can contain firmware - software permanently stored in hardware
 - Secondary storage
 - Used to save programs and data normally stored in RAM
 - Contents not lost when unit is turned off
 - E.g., Floppy disk drive

Input/Output Devices

- Computer I/O devices are often referred to as Peripherals
- Examples: disk drives, video monitors, printers, keyboards
- Two general classes of I/O interface
 - Serial interface port
 - Transmits/Receives data one bit at a time
 - Can transmit over long distances
 - Parallel interface port
 - High speed data transfer
 - Limited by cable length

Timing

- Basic timing is controlled by a clock generator circuit
- Clock signal is used to synchronize all activities within the computer
- Clock determines how fast instructions can be fetched from memory and executed
 - 16 MHz
 - 64 MHz
 - 800 MHz - Pentium 3
 - ...

Bus Architecture

- A bus is a collection of electronic signal lines all dedicated to a particular task
- The CPU, memory, and I/O are separate electronic modules that are interconnected by buses
- In a computer, there typically are three buses
 - Address Bus
 - Data Bus
 - Control Bus

Data Bus

- The width of the data bus in bits is used to classify the processor
 - 8-bit processor has an 8-bit data bus
 - 16-bit processor has a 16-bit data bus (Intel 8086 processor)
- The width of the data bus determines how much data the processor can read or write in one memory or I/O cycle
- If width of data exceeds capacity of data bus, reading/writing requires more than one read/write cycle -- less efficient
- The data bus is a bi-directional line

Address Bus

- The address bus is used to identify the location that the CPU will communicate with
- The address bus can identify a memory location or an I/O device, also called an I/O port
- For the 8086, the address bus is 20 bits wide, which allows for output to 2^{20} unique addresses
- The address bus is an output line

Control Bus

- The control bus identifies if the address on the address bus is for a memory location or for an I/O port
- The control bus identifies the direction of data flow on the data bus
- The CPU activates a control bus signal
 - MEMORY READ - read data from memory into the CPU
 - MEMORY WRITE - write data from the CPU to memory
 - I/O READ - read data from an input device into the CPU
 - I/O WRITE - write data from the CPU to an output device
- The control bus is an output line

Instruction Execution

- The CPU executes instructions in the following series of steps
 1. Fetch next instruction from memory (PC points to memory location)
 2. Move instruction into instruction register
 3. Change program counter to point to next instruction
 4. Determine type of instruction just fetched
 5. If instruction uses data in memory, determine where they are
 6. Fetch data, if any, into internal CPU registers
 7. Execute instruction
 8. Store result in proper place
 9. Go to step 1 and begin executing the following instruction

Fetch and Execute Cycle Example

- To monitor fetch/execution cycle, compile C code to Assembly code

Fragment of C code:

```
x = 0;
while (x <= 2)
    x = x + 1;
/* end while */
```

Compiles into Assembly Code:

Addr	instruction	comments
10	LOAD R1,0	Load 0 into register 1
11	STORE R1,X	Store contents of register 1 into mem location for x
12	LOAD R2,2	Load 2 into register 2
13	TEST R1,R2	Compare contents of reg 1 with contents of reg 2
14	JGT 17	If contents of reg 1 greater than reg 2, jump to line 17
15	ADD X,1,R1	Add 1 to contents of x and store in register 1
16	BRANCH 11	Unconditionally branch to step 11
17		continuation of program

Fetch and Execute Cycle Example

Trace execution:

	step 1	step 2	step 3	step 4	step 5	step 6	step 7	step 8	step 9	step 10
PC										
IR										
R1										
R2										
CCR										
X										

Review

- Computer architecture consists of CPU, Memory, and I/O, interconnected via busses
- Computers operate on a fetch, examine, execute cycle
- For Wed. and Fri, read Sections C5.9-5.13
- Lab sessions today and tomorrow
- In-class exam next Monday, 3/5/01
 - Covers material up to and including lecture #8 (2/23)
 - Covers material up to and including Problem Set #3
 - Closed book