DATA MANIPULATION

COMS W1001
Introduction to Information Science

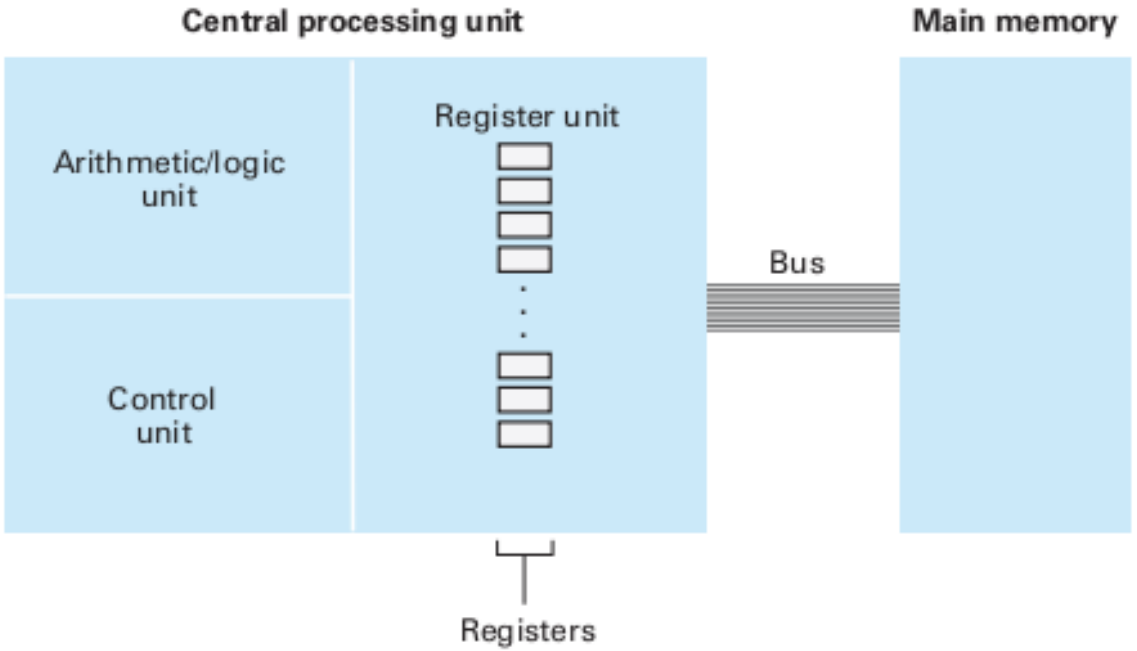
Boyi Xie

Today's Topics

- Computer Architecture
- Machine Language
- Program Execution
- Arithmetic/Logic Instructions
- Communication with Other Devices

CPU Basics

- Central Processing Unit (CPU)
  - The circuitry in a computer that controls the manipulation of data
  - Consists of
    - Arithmetic/logic unit – circuitry that performs operations on data
    - Control unit – circuitry for coordinating the machine's activities
    - Register unit – data storage cells, called registers (general-purpose registers & special-purpose registers)



Stored-Program Concept

- Early computers
  - Programs and data are different entities
  - Only data in memory
  - CPU could be conveniently rewired
- Stored-program concept
  - Programs can be encoded and stored in main memory
  - CPU to extract the program from memory, decode the instructions, and execute them
  - No CPU rewiring required

Machine Language

- CPU are designed to recognize the instructions encoded as bit patterns
- This collection of instructions along with the encoding system is called machine language
- An instruction expressed in machine language is called machine instruction

The Instruction Repertoire

- A typical CPU is able to decode only a limit number of machine instructions
- Once a machine can perform certain elementary but well-chosen tasks, adding more features does not increase the machine's theoretical capabilities
- Two philosophies of CPU architecture
  - RISC – reduced instruction set computer
    - To execute a minimal set of machine instructions
    - Machine is efficient and fast
  - CISC – complex instruction set computer
    - To execute a large number of complex instructions, even though many of them are technically redundant
    - Easier to program: a single instruction can be used to accomplish a task that would require a multi-instruction sequence in a RISC design

Instruction Category

- Data transfer
  - Movement of data from one location to another
  - Transfer of data between CPU and main memory, e.g. LOAD, STORE
  - I/O instructions
- Arithmetic/Logic
  - Boolean operations, e.g. AND, OR, XOR
  - Shift or rotate the contents in registers, e.g. SHIFT, ROTATE
- Control
  - Direct the execution of the program, e.g. JUMP (unconditional jump and conditional jump)

Instruction Category

- An example – dividing values stored in memory

Step 1. LOAD a register with a value from memory.

Step 2. LOAD another register with another value from memory.

Step 3. If this second value is zero, JUMP to Step 6.

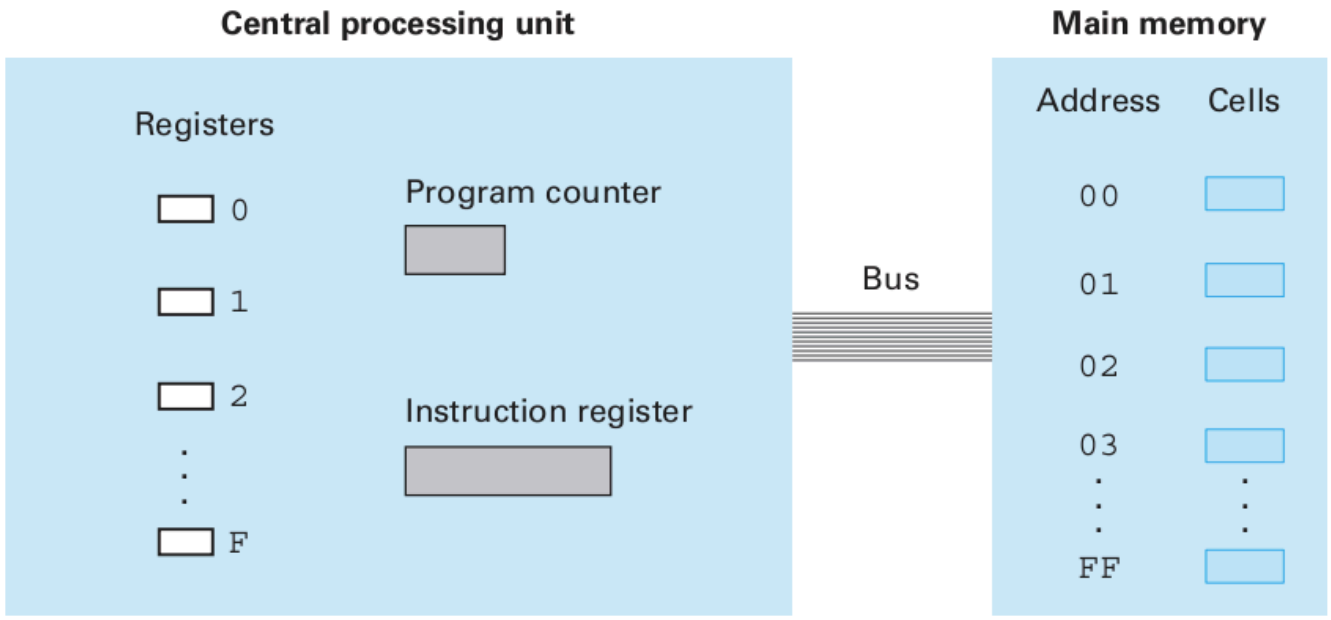
Step 4. Divide the contents of the first register by the second register and leave the result in a third register.

Step 5. STORE the contents of the third register in memory.

Step 6. STOP.

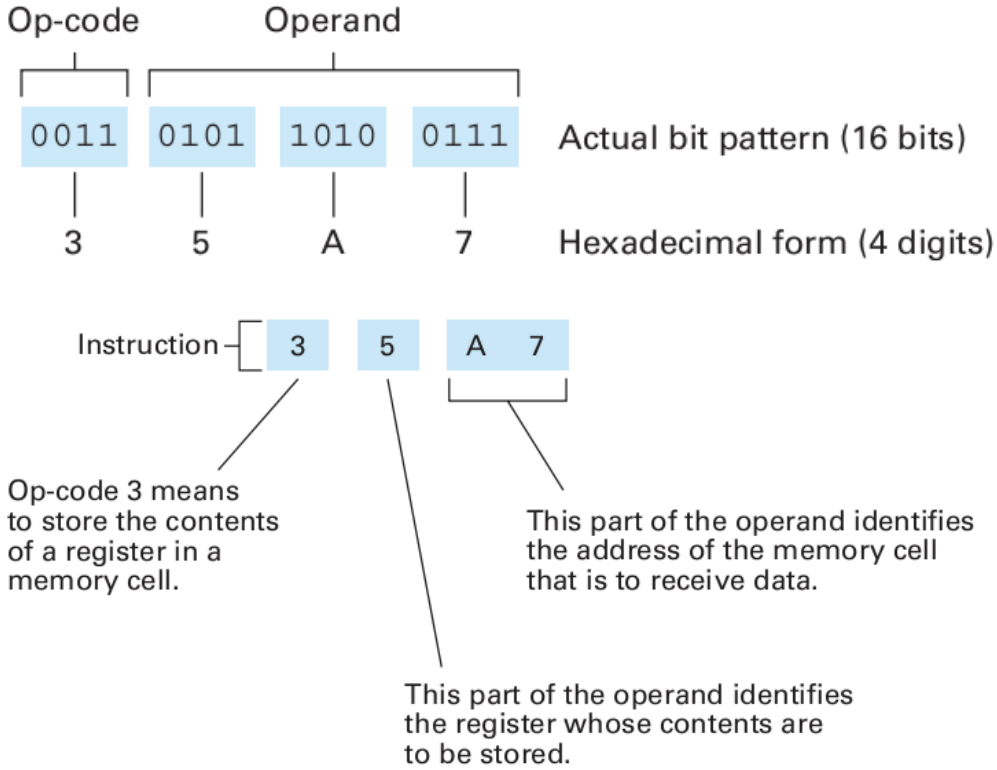
An Illustrative Machine Language

- Assume a computer with
  - 16 general-purpose registers
  - 256 main memory cells, each with a capacity of 8 bits
  - Machine instructions of 16 bits



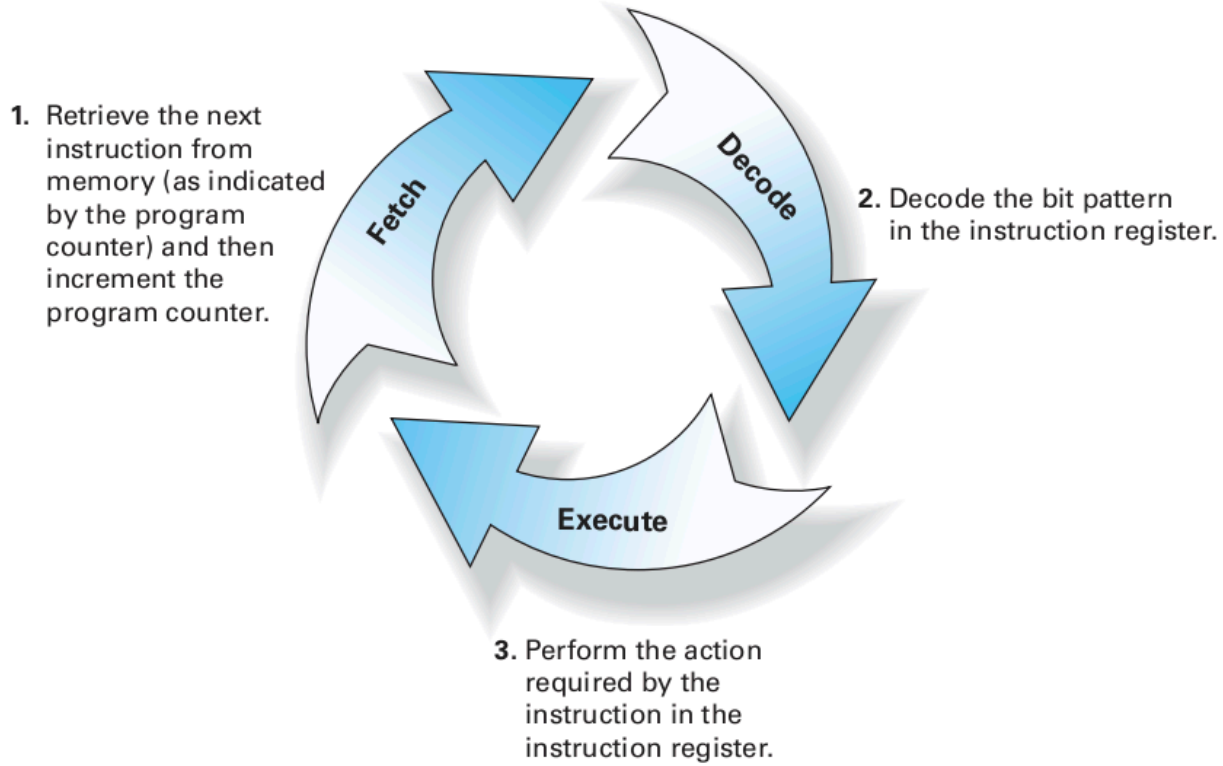
An Illustrative Machine Language

- Assume a computer with
  - 16 general-purpose registers
  - 256 main memory cells, each with a capacity of 8 bits
  - Machine instructions of 16 bits



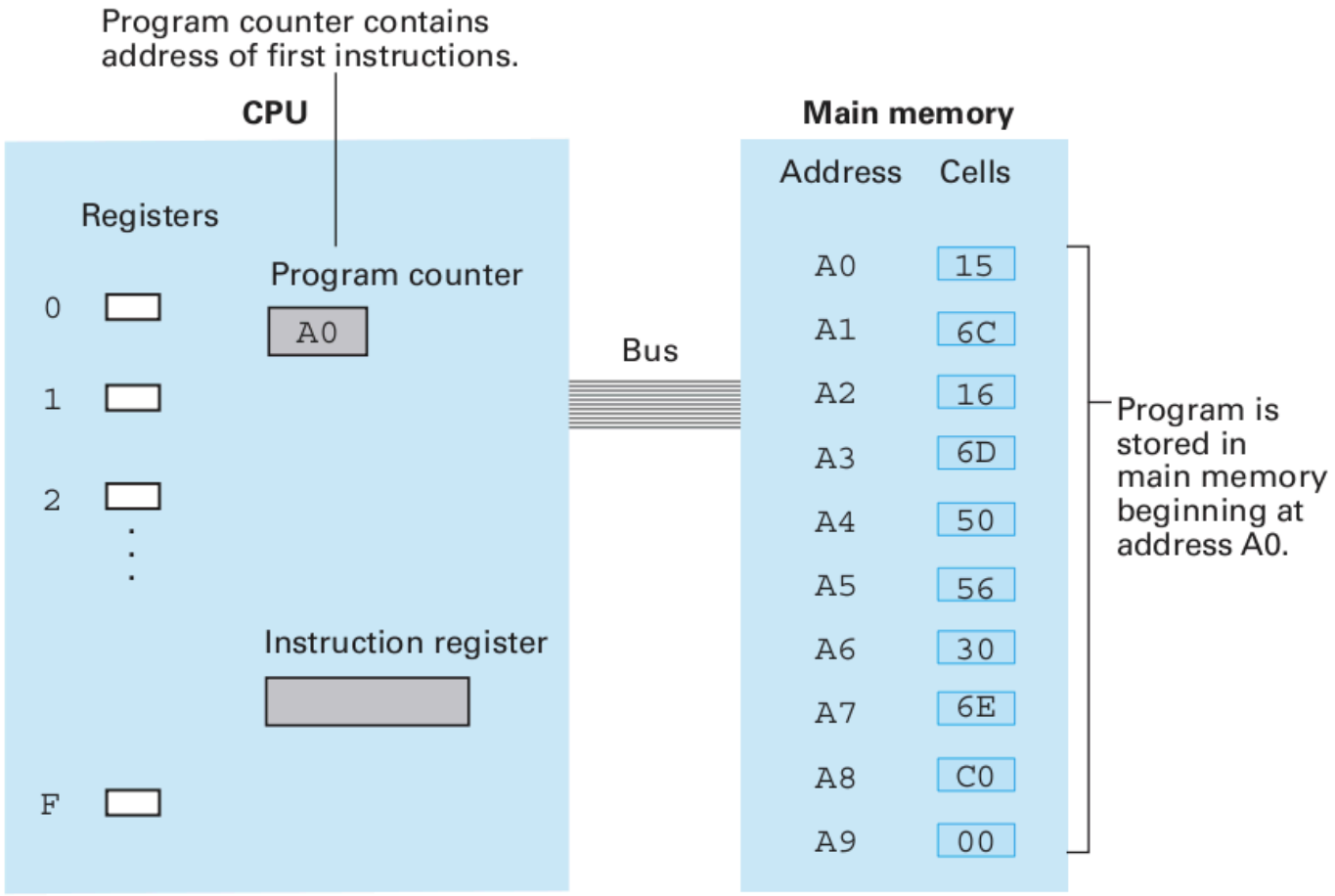
Program Execution

- Two special purpose registers
  - Instruction register – hold the instructions being executed
  - Program counter – contains the address of the next instruction
- The machine cycle – a three-step process

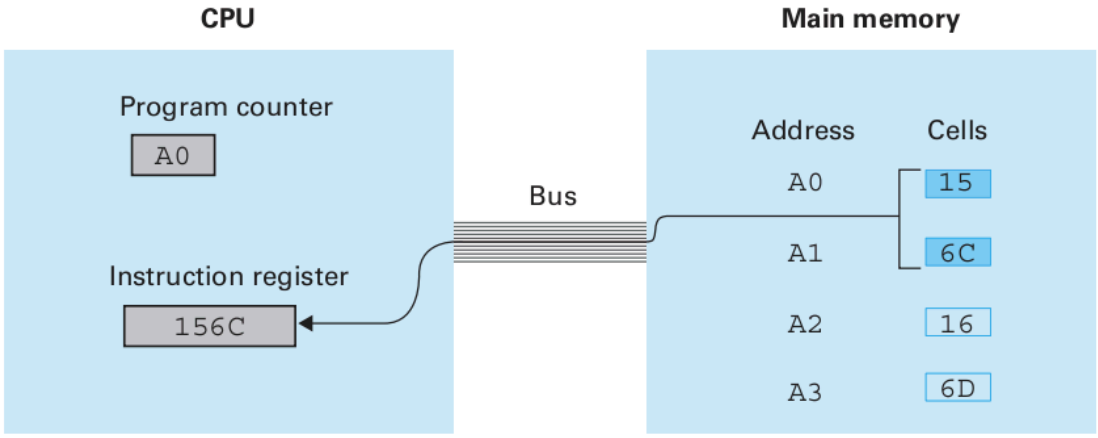


Program Execution

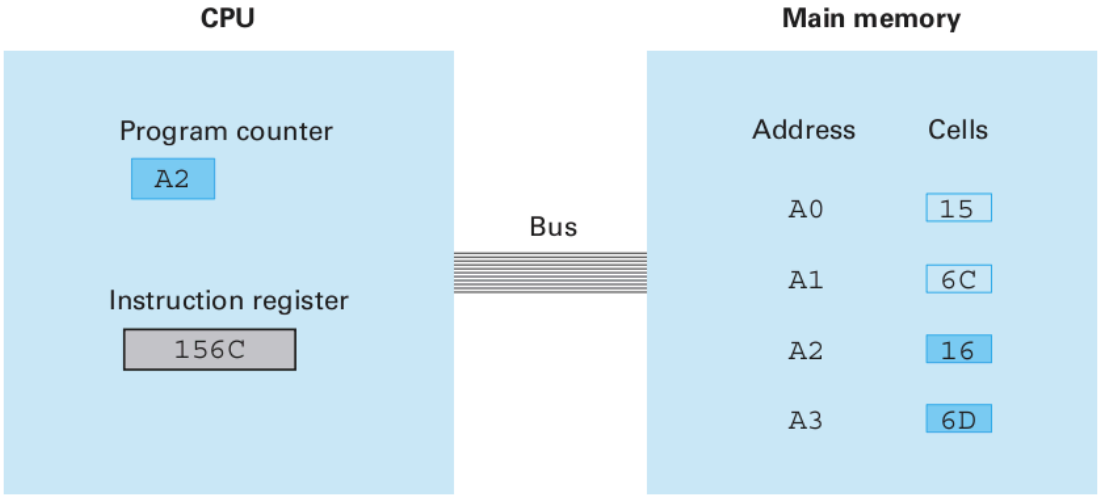
- An example of program execution



Program Execution



a. At the beginning of the fetch step the instruction starting at address A0 is retrieved from memory and placed in the instruction register.



b. Then the program counter is incremented so that it points to the next instruction.

Arithmetic/Logic Instructions

- Logic operations

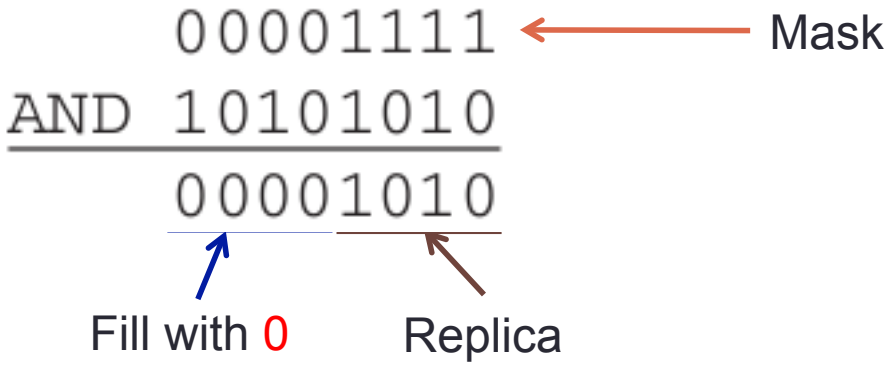
```
    10011010          10011010          10011010
AND 11001001      OR  11001001      XOR 11001001
    --------          --------          --------
    10001000          11011011          01010011
```

- Rotation and shift operations
  - Circular shift, or rotation – place the bit that fell off in the hole on the other side
  - Logical shift – discard the bit that falls off and always fill with 0
  - Arithmetic shift – shifts that leave the sign bit unchanged

- Arithmetic Operations
  - Add, subtract, multiply, and divide

Logic Instructions

- Use of AND
  - Masking – produce a result that is partial replica of one of the operands

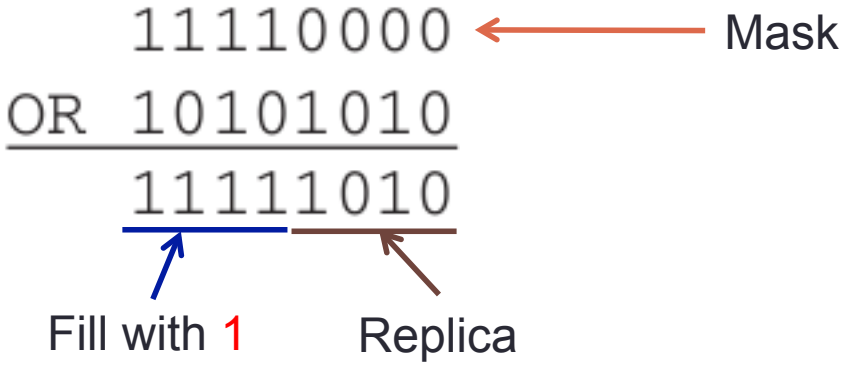


  - Force 0 in a position

```
    11011111
AND 10101010
------------
    10001010
```

Logic Instructions

- Use of OR
  - Masking – produce a result that is partial replica of one of the operands

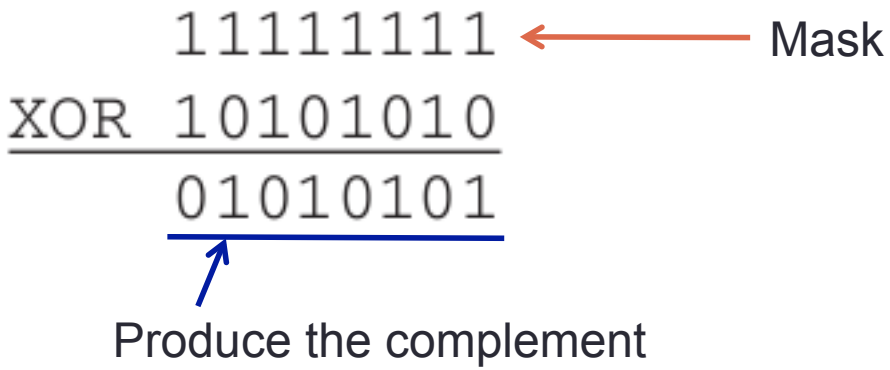


  - Force 1 in a position

```
    00100000
AND 10001110
------------
    10101110
```

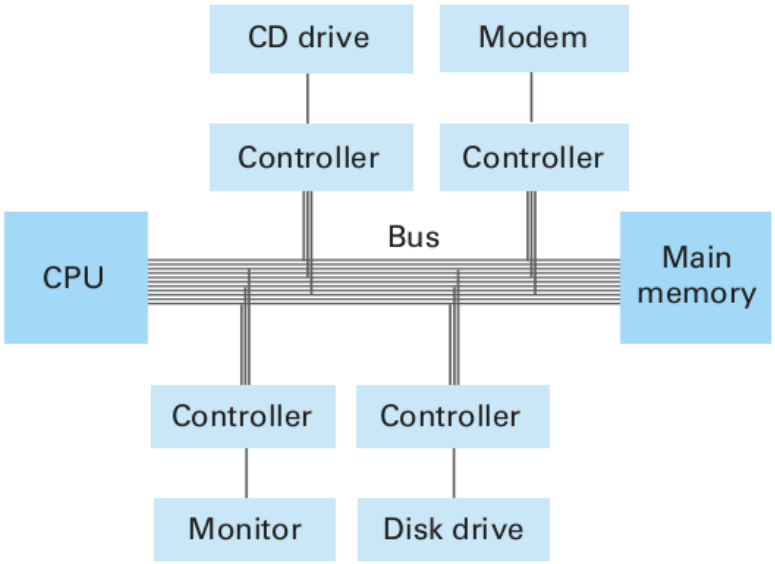
Logic Instructions

- Use of XOR
  - Form the complement of a bit string



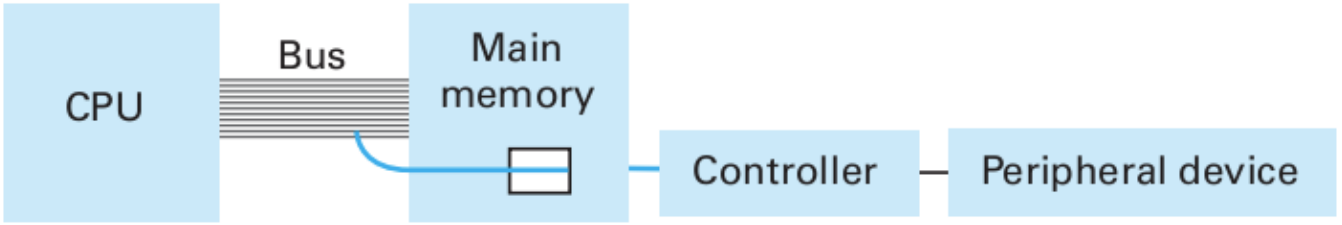
Communicating with Other Devices

- Controller – an intermediary apparatus that handles the communication between a computer and other devices
  - Originally, each controller was designed for a particular type of device
  - Gradually, a single controller is able to handle a variety of devices, e.g. universal serial bus (USB) and Thunderbolt
- Each controller communicates with the computer itself by means of connections to the same bus that connects the computer's CPU and main memory



Communicating with Other Devices

- Memory-mapped I/O
  - Computer's input/output devices appear to be in various memory locations
  - The transfer of data to and from controllers is directed by the same LOAD and STORE op-codes that are already provided for communication with main memory
  - Each controller is designed to respond to references to a unique set of addresses while main memory is designed to ignore references to these locations



Communicating with Other Devices

- Direct Memory Access (DMA)
  - A controller carry on its own communication with main memory
  - Enhance the computer's performance, e.g. the computing resources of the CPU are not wasted during the relatively slow data transfer from disk to memory
  - Complicate the communication taking place over a computer's bus
- Von Neumann bottleneck
  - Von Neumann architecture in which a CPU fetches its instructions from memory over a central bus
  - Coordination of all the activities on the bus is a major design issue
  - The central bus can become an impediment as the CPU and the controllers compete for bus access

Communicating with Other Devices

- Handshake
  - A two-way dialogue between the computer and the peripheral device to exchange information about the device's status and coordinate their activities
  - Status word is often involved
    - A bit pattern generated by the peripheral device and sent to the controller
    - Reflect the conditions of the device, e.g. printer out of paper, ready for additional data, paper jam, etc.

Communicating with Other Devices

- Popular Communication Media
  - Parallel communication – several signals transferred at the same time, each on a separate line, e.g. a computer's internal bus
  - Serial communication – signals transferred one after the other over a single line, e.g. USB, Thunderbolt
  - Long distance communication
    - Modem (modulator-demodulator) – convert bit patterns into audible tones
    - DSL (Digital Subscriber Line) – uses frequencies above the audible range to transfer digital data while leaving the lower frequency spectrum for voice
- Communication rates
  - Measured in bits per second (bps), Kbps, Mbps, Gbps, etc.
  - Bandwidth – the maximum rate available on the communication path

Other Architectures

- Pipelining
  - Increasing execution speed is not the only way to improve a computer's performance
  - Real goal is to improve throughput – the total amount of work the machine can accomplish in a given amount of time
  - Use pipelining – the technique of allowing the steps in the machine cycle to overlap
- Multiprocessor machines
  - Parallel processing – process several activities at the same time
  - MIMD (multiple-instruction stream, multiple-data stream) architecture
  - SISD (single-instruction stream, single-data stream) architecture
  - SIMD (single-instruction stream, multiple-data stream) architecture

References & Photo Credits

- Brookshear, J. Glenn (2011-04-13). Computer Science: An Overview (11th Edition). Prentice Hall. Kindle Edition.