

# INSTRUCTION CODES

# INSTRUCTION CODES

- Every different processor type has its own design (different registers, buses, microoperations, machine instructions, etc)
- It contains
  - Many registers
  - Multiple arithmetic units, for both integer and floating point calculations
  - The ability to pipeline several consecutive instructions to speed execution

## INSTRUCTION CODES

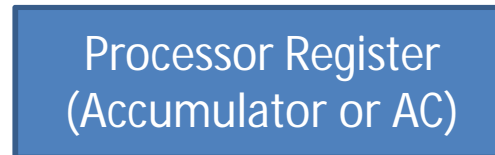
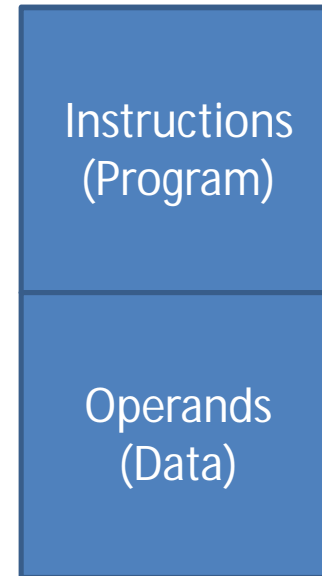
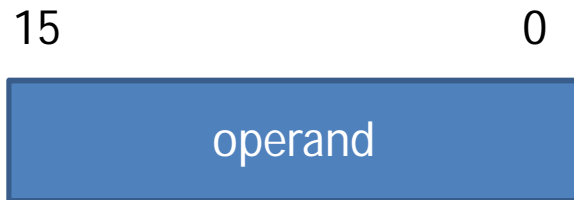
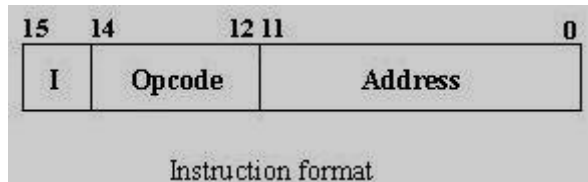
- Program
  - A sequence of (machine) instructions
- (Machine) Instruction
  - A group of bits that tell the computer to *perform a specific operation* (a sequence of micro-operation)
- The instructions of a program, along with any needed data are stored in memory
- The CPU reads the next instruction from memory
- It is placed in an *Instruction Register (IR)*

## INSTRUCTION CODES

- Control circuitry in control unit then translates the instruction into the sequence of microoperations necessary to implement it
- A computer instruction is often divided into two parts
  - An *opcode* (Operation Code) that specifies the operation for that instruction
  - An *address* that specifies the registers and/or locations in memory to use for that operation

# Stored Program Organization

Memory 4096 x 16



# Stored Program Organization

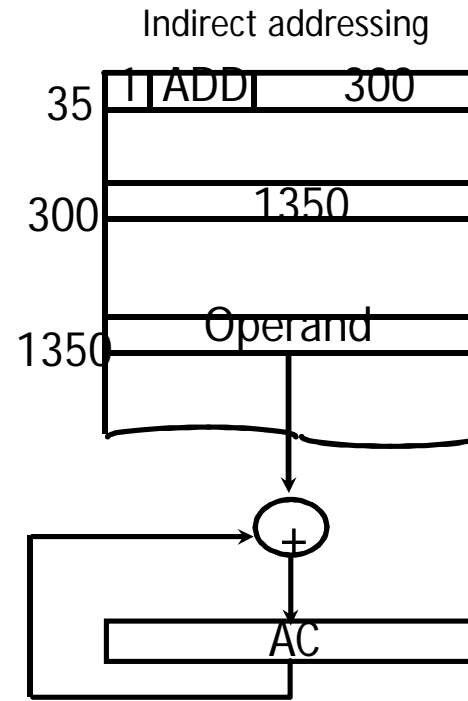
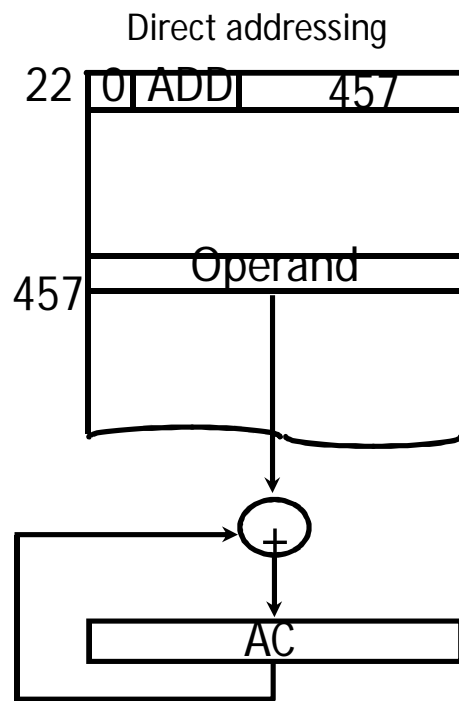
- The Basic Computer has two components, a processor and memory
- The memory has 4096 words in it
  - $4096 = 2^{12}$ , so it takes 12 bits to select a word in memory
- Each word is 16 bits long
- In the Basic Computer, since the memory contains 4096 ( $= 2^{12}$ ) words, we need 12 bits to specify which memory address.

# Stored Program Organization

- In the Basic Computer, bit 15 of the instruction specifies the *addressing mode* (0: direct addressing, 1: indirect addressing)
- If the operation in an instruction does not need an operand from memory, the rest of the bits in the instruction can be used for other purposes
- Ex: Clear AC, Complement AC, Increment AC.

# Addressing Modes

- The address field of an instruction can represent either
  - Direct address
  - Indirect address





# Direct Addressing

- The instruction is placed at the location 22 in memory.
- The opcode specifies ADD instruction
- Address part specifies the address of the operand that is 457.
- The operand at the location 457 is added with AC content and the result is stored in AC by overwriting its previous content.

# Indirect Addressing

- The instruction is placed at the location 35 in memory.
- The opcode specifies ADD instruction.
- Address part specifies the address 300 and mode bit 1 specifies indirect address.
- The control goes to the address 300 to find the address of the operand.
- In this case the address of the operand is 1350
- The operand at the location 1350 is added with AC content and the result is stored in AC by overwriting its previous content.

# Indirect Addressing

- The indirect address instruction needs two references to memory to fetch an operand.
- First reference to fetch address of operand
- Second reference to fetch the operand itself.