Overview

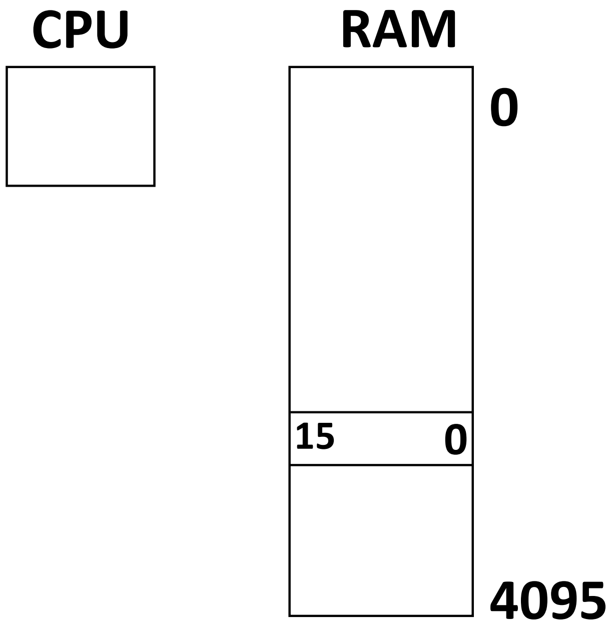
- **Instruction Codes**
- **Computer Registers**
- Computer Instructions
- Timing and Control
- Instruction Cycle
- Memory Reference Instructions
- Input-Output and Interrupt
- Complete Computer Description

Introduction

- **Every different processor type has its own design (different registers, buses, microoperations, machine instructions, etc)**
- **Modern processor is a very complex device**
- **It contains**
  - **Many registers**
  - **Multiple arithmetic units, for both integer and floating point calculations**
  - **The ability to pipeline several consecutive instructions to speed execution**
  - **Etc.**
- **However, to understand how processors work, we will start with a simplified processor model**

Basic Computer

- 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



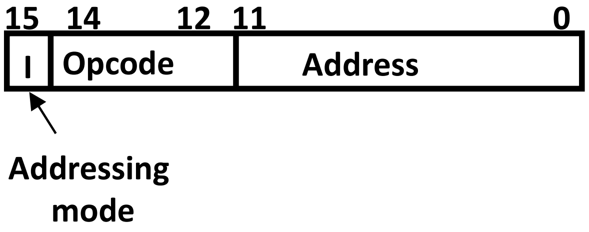
Instruction

- **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)**
- **Control circuitry in control unit then translates the instruction into the sequence of microoperations necessary to implement it**

Instruction Format

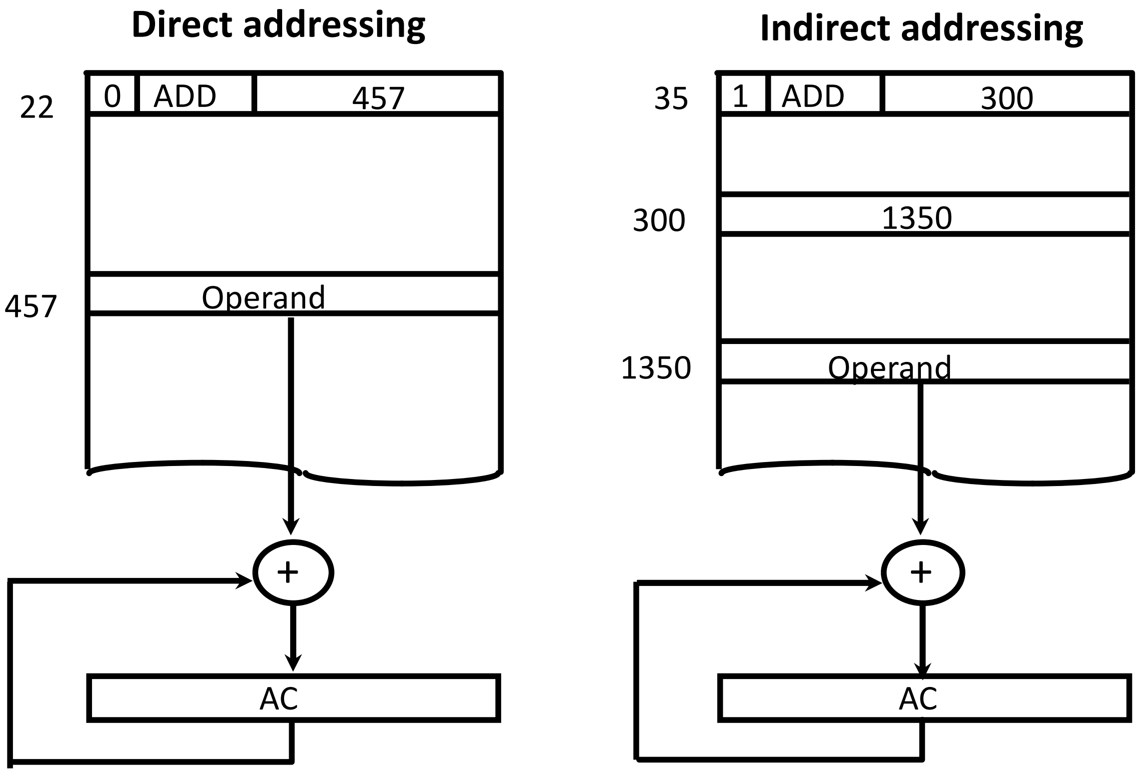
- **A computer instruction is often divided into two parts**
  - **An opcode (Operation Code) that specifies the operation for that instruction**
  - **An address that specifies the registers and/or locations in memory to use for that operation**
- **In the Basic Computer, since the memory contains 4096 (= 2^{12}) words, we needs 12 bit to specify which memory address this instruction will use**
- **In the Basic Computer, bit 15 of the instruction specifies the addressing mode (0: direct addressing, 1: indirect addressing)**
- **Since the memory words, and hence the instructions, are 16 bits long, that leaves 3 bits for the instruction's opcode**

Instruction Format



Addressing Mode

- **The address field of an instruction can represent either**
  - **Direct address: the address in memory of the data to use (the address of the operand), or**
  - **Indirect address: the address in memory of the address in memory of the data to use**



- **Effective Address (EA)**
  - **The address, that can be directly used without modification to access an operand for a computation-type instruction, or as the target address for a branch-type instruction**

Processor Register

- A processor has many registers to hold instructions, addresses, data, etc
- The processor has a register, the Program Counter (PC) that holds the memory address of the next instruction to get

  Since the memory in the Basic Computer only has 4096 locations, the PC only needs 12 bits
- In a direct or indirect addressing, the processor needs to keep track of what locations in memory it is addressing: The Address Register (AR) is used for this

  The AR is a 12 bit register in the Basic Computer
- When an operand is found, using either direct or indirect addressing, it is placed in the Data Register (DR). The processor then uses this value as data for its operation
- The Basic Computer has a single general purpose register – the Accumulator (AC)

Processor Register

- **The significance of a general purpose register is that it can be referred to in instructions**

  e.g. load AC with the contents of a specific memory location; store the contents of AC into a specified memory location

- **Often a processor will need a scratch register to store intermediate results or other temporary data; in the Basic Computer this is the Temporary Register (TR)**

- **The Basic Computer uses a very simple model of input/output (I/O) operations**

  Input devices are considered to send 8 bits of character data to the processor

  The processor can send 8 bits of character data to output devices

- **The Input Register (INPR) holds an 8 bit character gotten from an input device**

- **The Output Register (OUTR) holds an 8 bit character to be send to an output device**