INSTRUCTION CYCLE

Instruction Cycle

- Computer Program consist of sequence of instruction residing in memory
- The program is executed by going through a cycle for each instruction.
- Each instruction cycle is subdivided into sequence of sub cycles as follows:
    1. Fetch an instruction from memory
    2. Decode the instruction
    3. Read the effective address from memory if the instruction has an indirect address
    4. Execute the instruction
- The above 4 step process continues until a HALT instruction is encountered

Fetch and Decode Phase

- Program counter(PC) is loaded with the address of first instruction initially.
- The sequence counter(SC) is cleared to zero, providing the timing signal T0.
- After each clock pulse, SC is incremented by 1, so timing goes through T1,T2,.....
- The Microoperation for fetch and decode phase are given below:

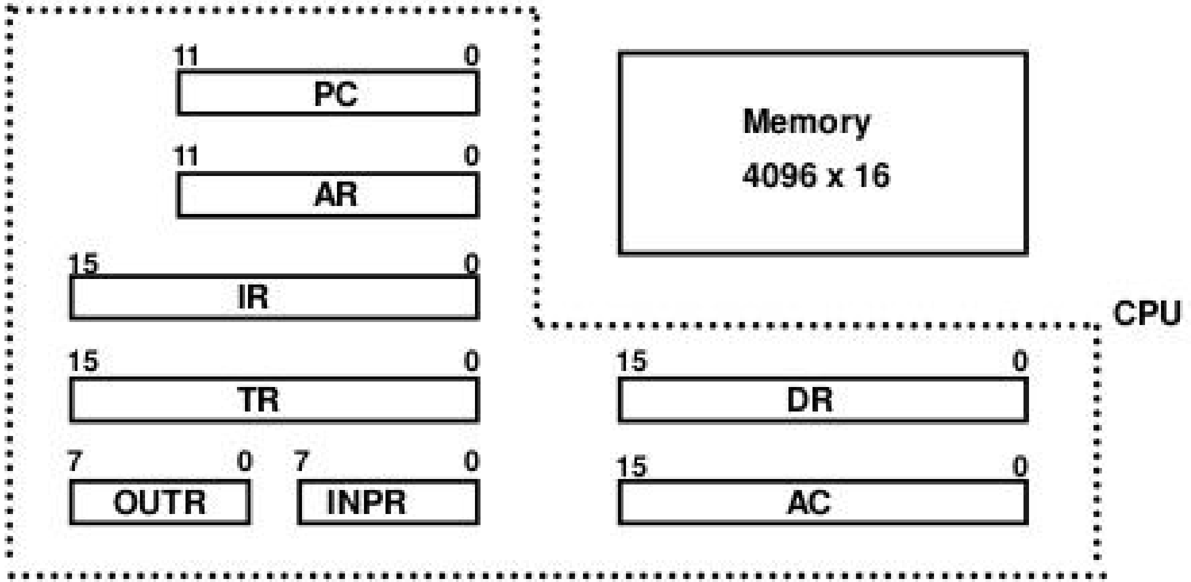
  T0: AR←PC

  T1: IR ←M[AR], PC ←PC + 1

  T2: D0, ..., D7 ←Decode IR(12-14), AR ←IR(0-11),
  I ←IR(15)

BASIC COMPUTER REGISTERS

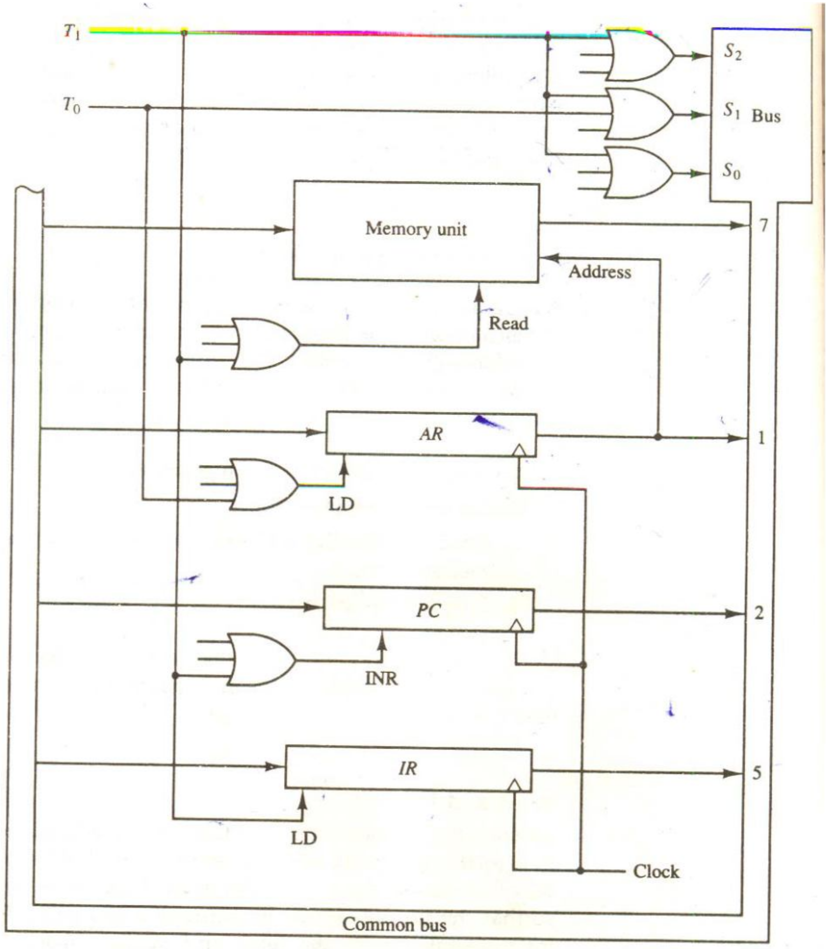
Registers in the Basic Computer



List of BC Registers

DR	16	Data Register	Holds memory operand
AR	12	Address Register	Holds address for memory
AC	16	Accumulator	Processor register
IR	16	Instruction Register	Holds instruction code
PC	12	Program Counter	Holds address of instruction
TR	16	Temporary Register	Holds temporary data
INPR	8	Input Register	Holds input character
OUTR	8	Output Register	Holds output character

Register transfers for the fetch phase



Fetch and Decode Phase

T0: AR←PC

- Since only AR is connected to the address inputs of memory, the address of instruction is transferred from PC to AR.
- Place the content of PC onto the bus by making the bus selection inputs S2S1S0 = 010.
- Transfer the content of the bus to AR by enabling the LD input

Fetch and Decode Phase

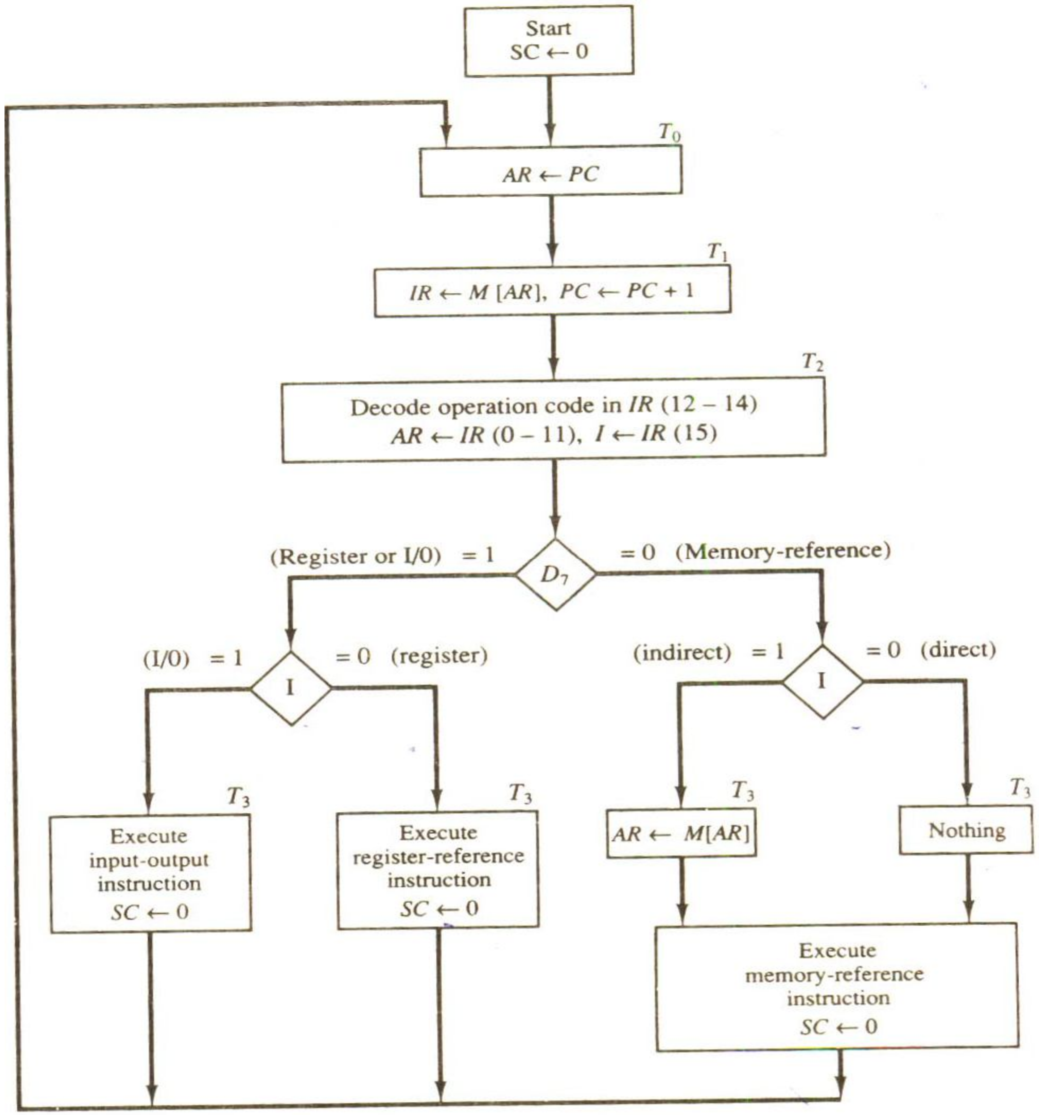
T1: IR ←M[AR], PC ←PC + 1

1. Enable the read input of memory.
2. Place the content of memory onto the bus by making S2S1S0 = 111.
3. Transfer the content of the bus to IR by enabling the LD input of IR.
4. Increment PC by enabling the INR input of PC.

DETERMINE THE TYPE OF INSTRUCTION

- The timing signal that is active after the decoding is T3.
- During time T3, the control unit determines the type of instruction that was just read from memory.
- The following flowchart presents an initial configuration for the instruction cycle and shows how the control determines the instruction type after the decoding.

DETERMINE THE TYPE OF INSTRUCTION



DETERMINE THE TYPE OF INSTRUCTION

- Decoder output D7 is equal to 1 if the operation code is equal to binary 111.
- We determine that if D7 = 1, the instruction must be a register-reference or input-output type.
- If D7 = 0, the operation code must be one of the other seven values 000 through 110, specifying memory reference instruction.
- Control then inspects the value of the first bit of the instruction, which is now available in flip-flop I.
- If D7 = 0 and I = 1, we have a memory-reference instruction with an indirect address.
- It is then necessary to read the effective address from memory. The micro operation for the indirect address condition can be symbolized by the register transfer statement

$$AR \leftarrow M\,[AR]$$

DETERMINE THE TYPE OF INSTRUCTION

- Initially, AR holds the address part of the instruction. This address is used during the memory read operation.
- The word at the address given by AR is read from memory and placed on the common bus.
- The LD input of AR is then enabled to receive the effective address of the operand

DETERMINE THE TYPE OF INSTRUCTION

- The three instruction types are subdivided into four separate paths.
- The selected operation is activated with the clock transition associated with timing signal T3.
- This can be symbolized as follows:

D7 = 0, I = 1 --- D7’ IT3: AR← M [AR]

D7 = 0, I = 0 --- D7’ I’T3: Nothing

D7 = 1, I = 0 --- D7 I’ T3: Execute a register-reference instruction

D7 = 1, I = 1 --- D7 IT3: Execute an input-output instruction