Basic Computer Organization and Design

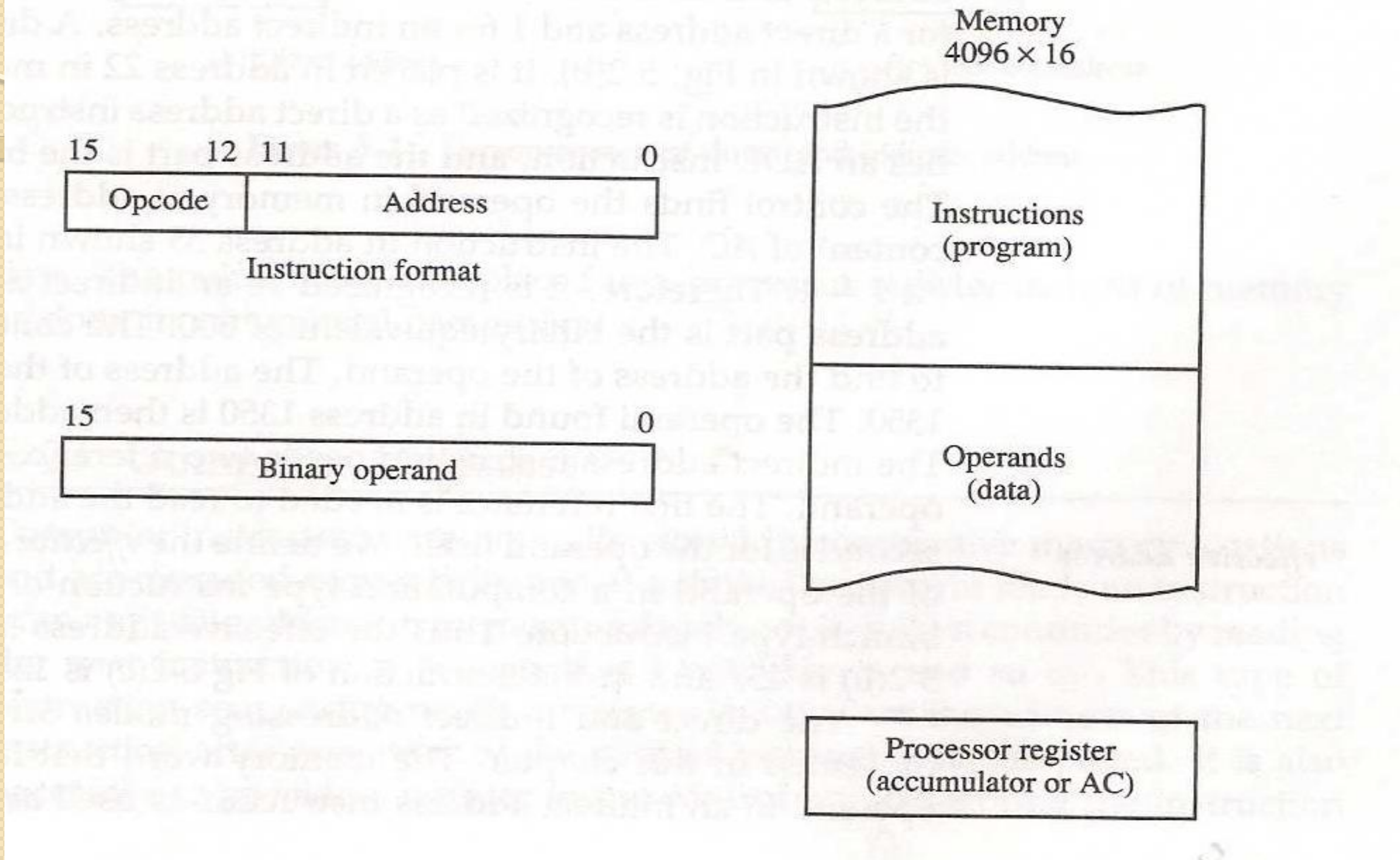
Computer Organization

- Organization of computer is defined by:
  - Internal Registers
  - Timing and Control Structure
  - Set of Instructions
- Internal Organization of digital system is defined by the sequence of microoperations it performs on data stored in registers
- General Purpose Computer is capable of
  - Executing various microoperations
  - It can be instructed as to what specific sequence of operations it perform

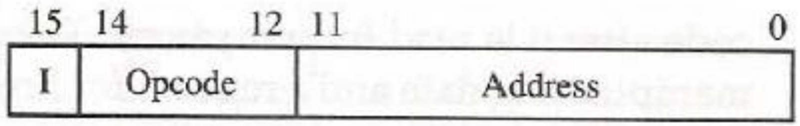
- User of a computer can control the process by means of a program
- Program is a set of instructions that specify the operations, operands and sequence by which processing has to occur
- A computer instruction is a binary code that specifies a sequence of microoperations for the computer
- Instruction code together with data are stored in memory
- Computer reads each instruction from memory and places it in control registers

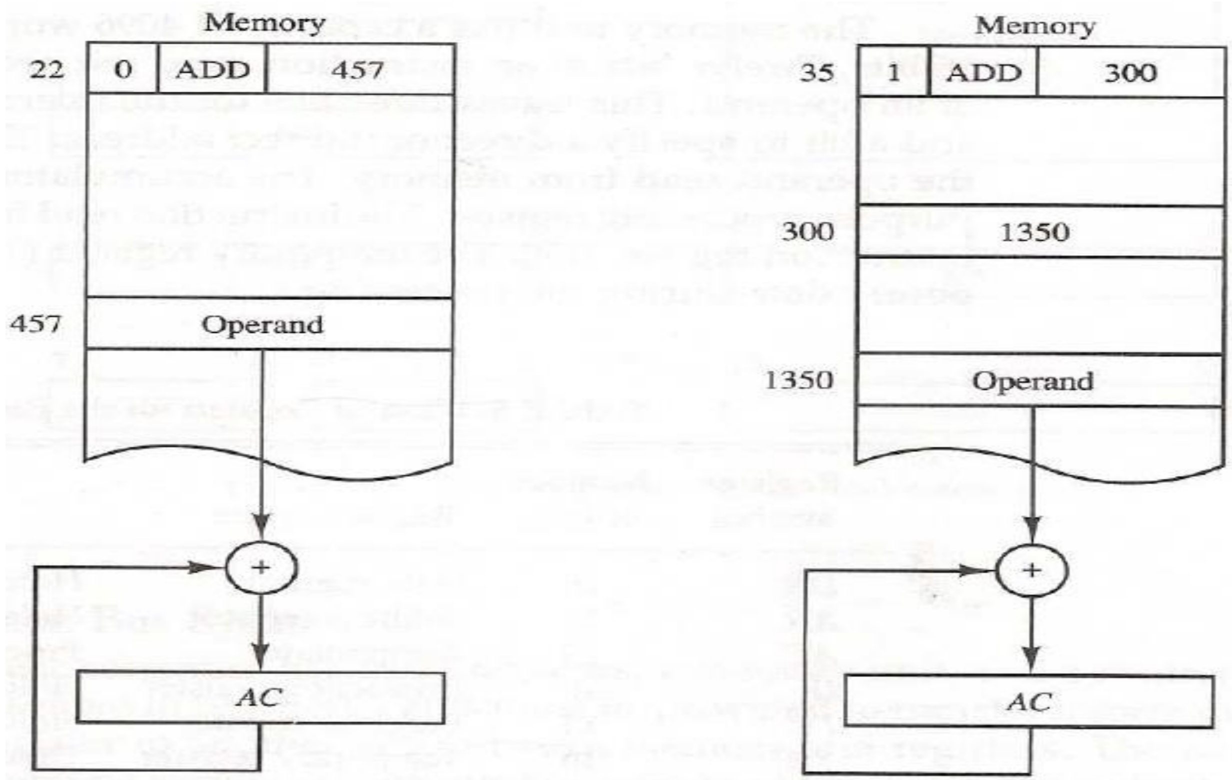
- Control then interprets the binary code of the instruction and proceeds to execute it by issuing a sequence of microoperations
- Every computer has its own unique instruction set
- An Instruction code is group of binary bits that instruct the computer to perform a specific operation
- It is usually divided into parts, the most basic is operation part
- The operation code of an instruction is a group of bits that define operations such as ADD,MUL,SUB

Stored Program Organization



Indirect Address





(b) Direct address

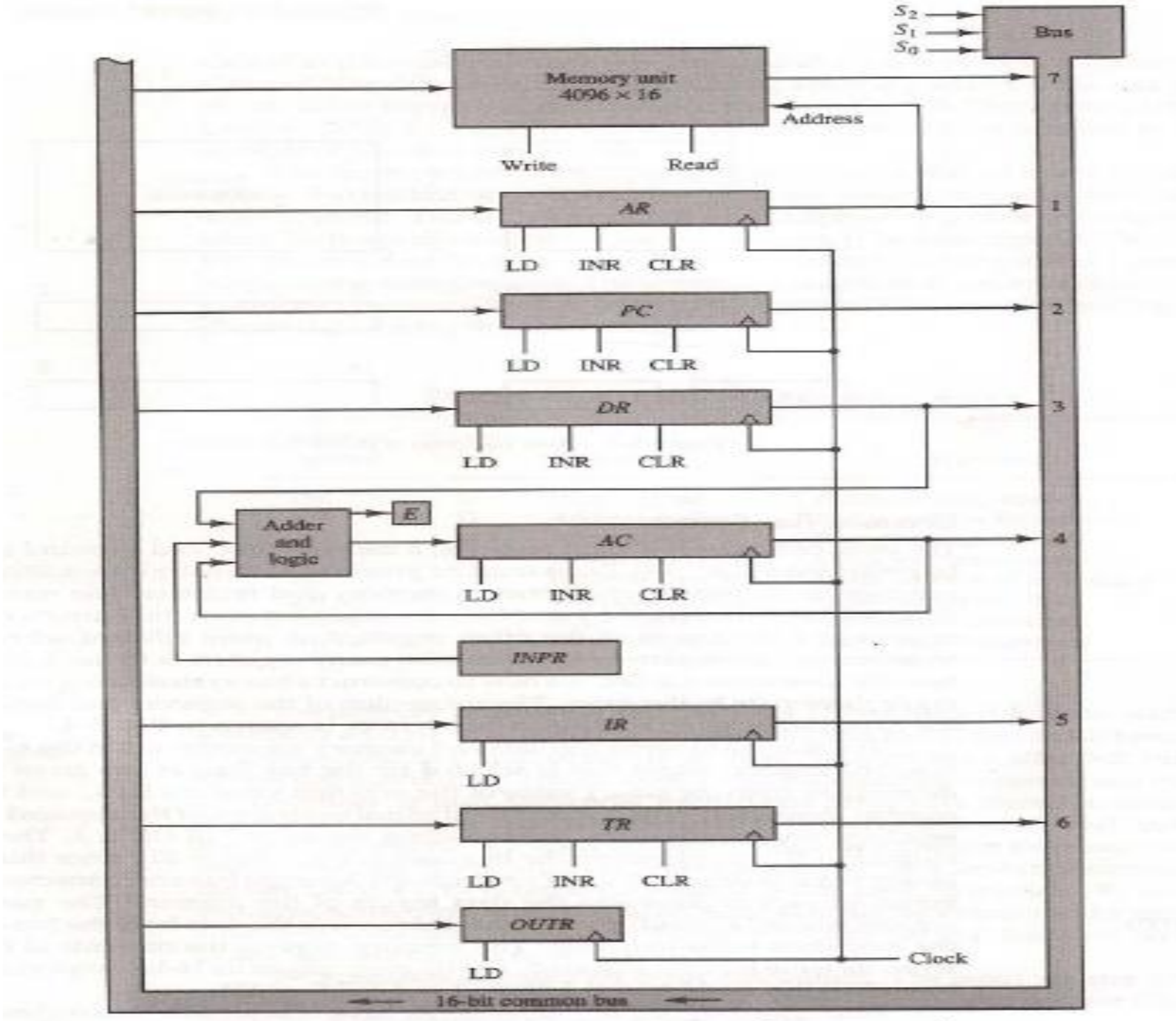
(c) Indirect address

List Of Registers in Basic Computer

Register symbol	Number of bits	Register name	Function
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

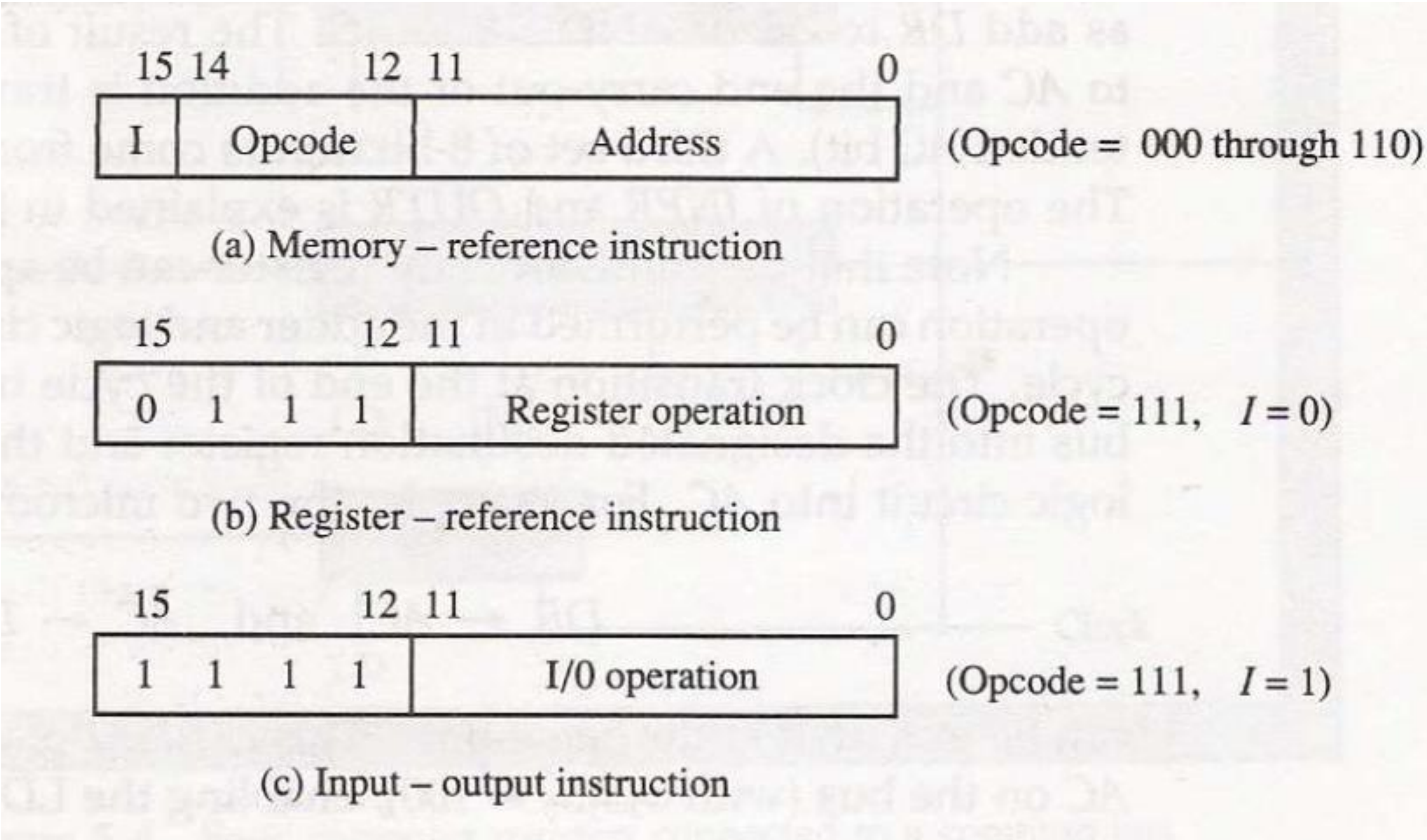
Common Bus System

- Basic computer has 8 registers, control unit and a memory unit
- Paths must be provided to transfer information from one register to another and between memory and registers
- Lots of wires will be required to connect output of each registers to the input of each register
- An efficient scheme is needed to transfer of information



S_2
S_1
S_0
Bus
Memory unit
4096 × 16
Address
Write
Read
7
AR
LD INR CLR
1
PC
LD INR CLR
2
DR
LD INR CLR
3
Adder and logic
E
AC
LD INR CLR
4
INPR
IR
LD
5
TR
LD INR CLR
6
OUTR
LD
Clock
16-bit common bus

Computer Instructions



Basic Computer Instructions

Symbol	Hexadecimal code I = 0	Hexadecimal code I = 1	Description
AND	0xxx	8xxx	AND memory word to AC
ADD	1xxx	9xxx	Add memory word to AC
LDA	2xxx	Axxx	Load memory word to AC
STA	3xxx	Bxxx	Store content of AC in memory
BUN	4xxx	Cxxx	Branch unconditionally
BSA	5xxx	Dxxx	Branch and save return address
ISZ	6xxx	Exxx	Increment and skip if zero
CLA	7800		Clear AC
CLE	7400		Clear E
CMA	7200		Complement AC
CME	7100		Complement E
CIR	7080		Circulate right AC and E
CIL	7040		Circulate left AC and E
INC	7020		Increment AC
SPA	7010		Skip next instruction if AC positive
SNA	7008		Skip next instruction if AC negative
SZA	7004		Skip next instruction if AC zero
SZE	7002		Skip next instruction if E is 0
HLT	7001		Halt computer
INP	F800		Input character to AC
OUT	F400		Output character from AC
SKI	F200		Skip on input flag
SKO	F100		Skip on output flag
ION	F080		Interrupt on
IOF	F040		Interrupt off

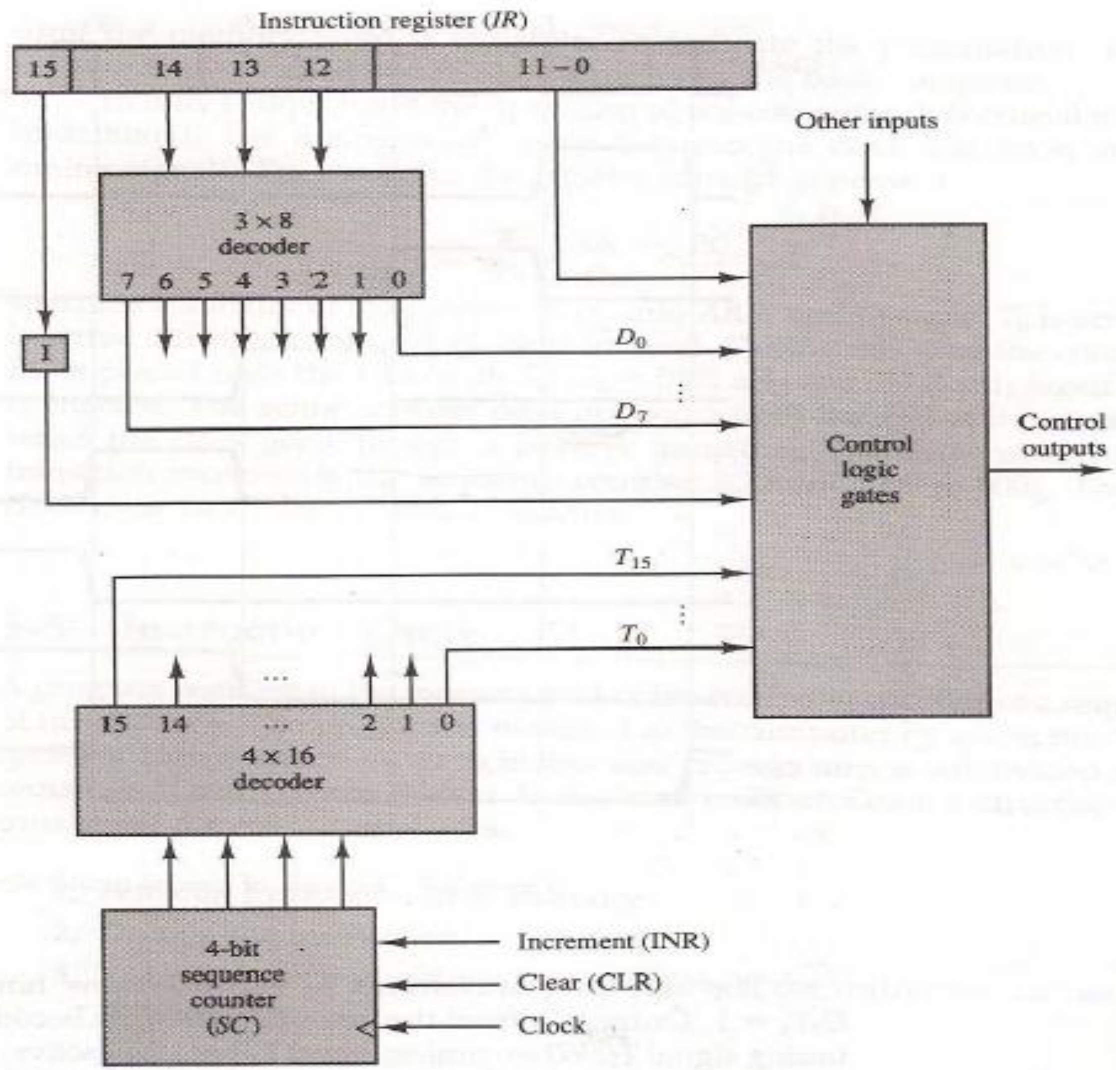
What type of instructions be included?

- A computer should have a set of instructions so that the user can construct machine language programs to evaluate any function that is computable
- Set of instructions are said to be complete if computer have sufficient number of instructions in the following categories:
  - Arithmetic, Logical and Shift instructions
  - Instructions for moving information to and from memory and processor registers
  - Program control instructions to check status conditions
  - Input and Output instructions

Timing and Control

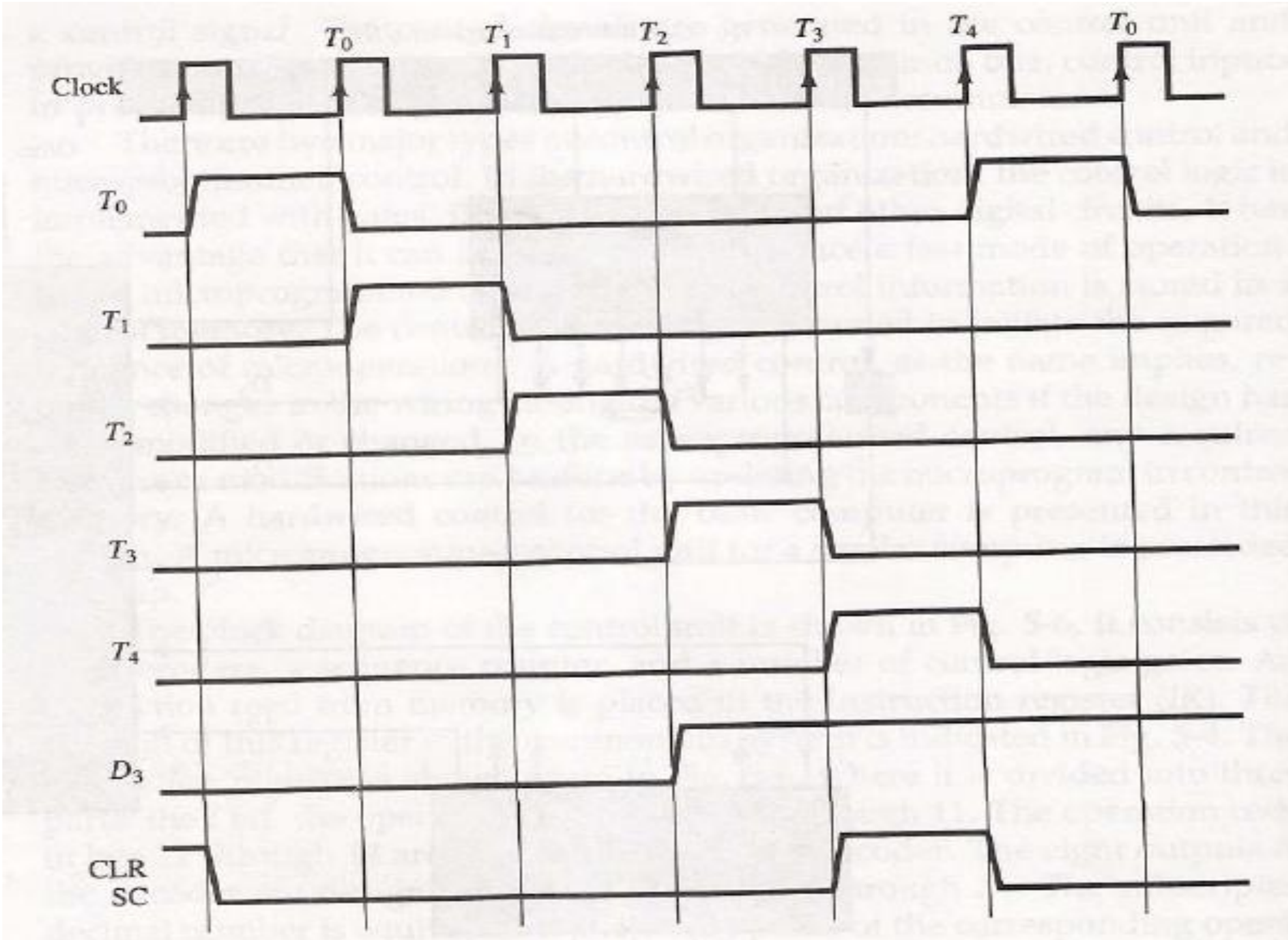
- Timings in basic computer is generated by master clock generator
- Clock Pulses are applied to all flip flops and registers in the system including those in the control unit
- Clock Pulses do not change the state of register unless the register is enabled by the control signal
- Control signals are generated in the control unit and provide control inputs for the multiplexers in the common bus, control inputs in processor registers and microoperations for the control input

- Two major types of control organization
  - Hardwired Control
  - Micro programmed Control
- Basic control unit for Hardwired Control organization consists of
  - Two decoders
  - Sequence Counter
  - No. of control logic gates



Instruction register (IR)
15
14
13
12
11 – 0
Other inputs
3 × 8
decoder
7 6 5 4 3 2 1 0
I
D0
D7
Control
logic
gates
Control
outputs
T15
T0
15 14 ... 2 1 0
4 × 16
decoder
4-bit
sequence
counter
(SC)
Increment (INR)
Clear (CLR)
Clock

D_3T_4: SC ← 0



Instruction Cycle

- Program residing in the memory unit of computers contains sequence of instructions
- Program is executed in computer by going through a cycle for each instruction
- Each instruction cycle is sub divided into different sub cycles or phases
- In basic computer each instruction cycle consists of :
  - Fetch an instruction from memory
  - Decode the instruction
  - Read the effective address from the memory if instruction has indirect address
  - Execute the instruction

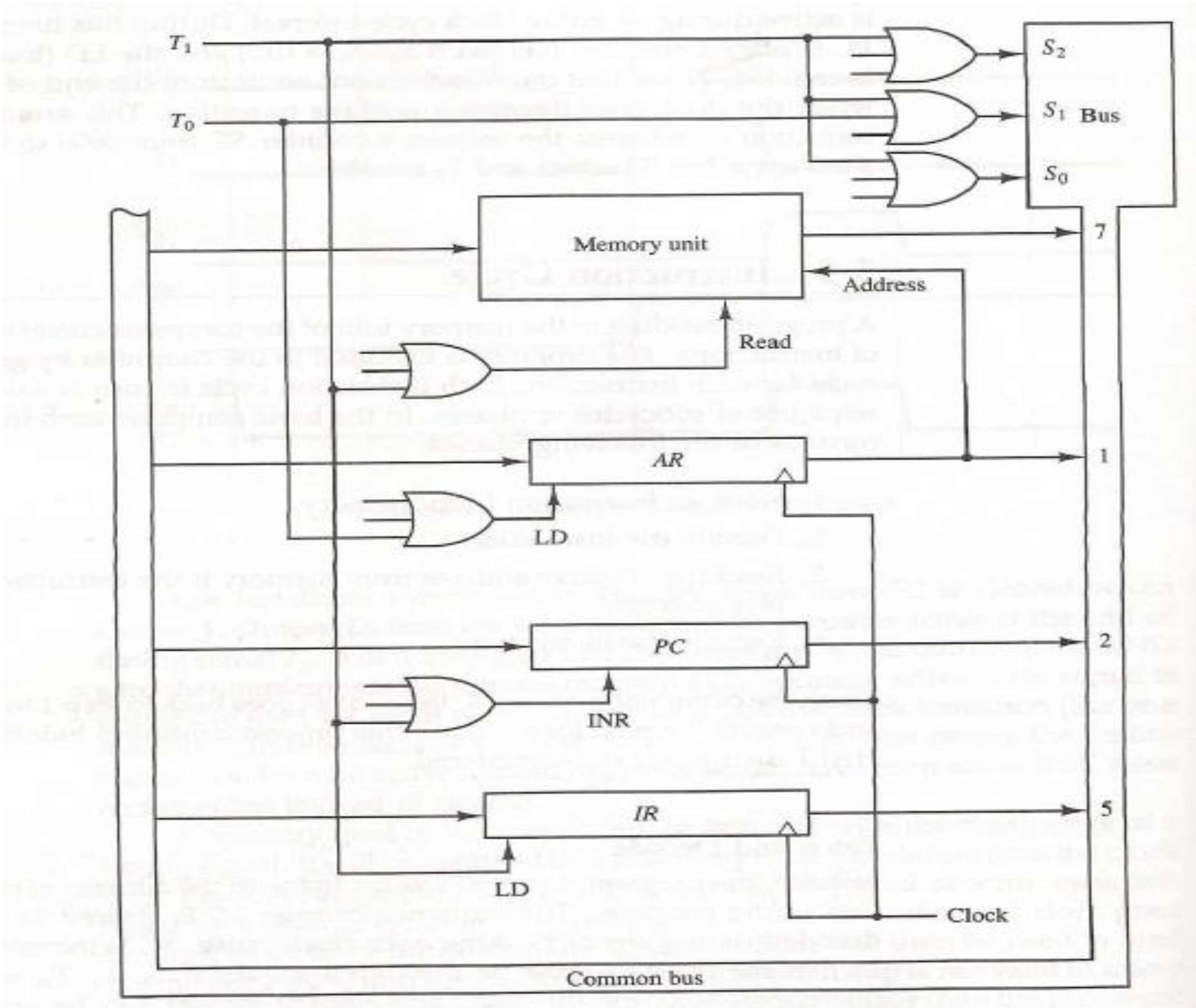
Fetch & Decode

- Initially, the program counter PC is loaded with address of the first instruction
- SC is cleared to 0 to decode T_0
- Microoperations for fetch and decode phase can be specified by following register statements:

T_0: $AR \leftarrow PC$

T_1: $IR \leftarrow M[AR], \quad PC \leftarrow PC + 1$

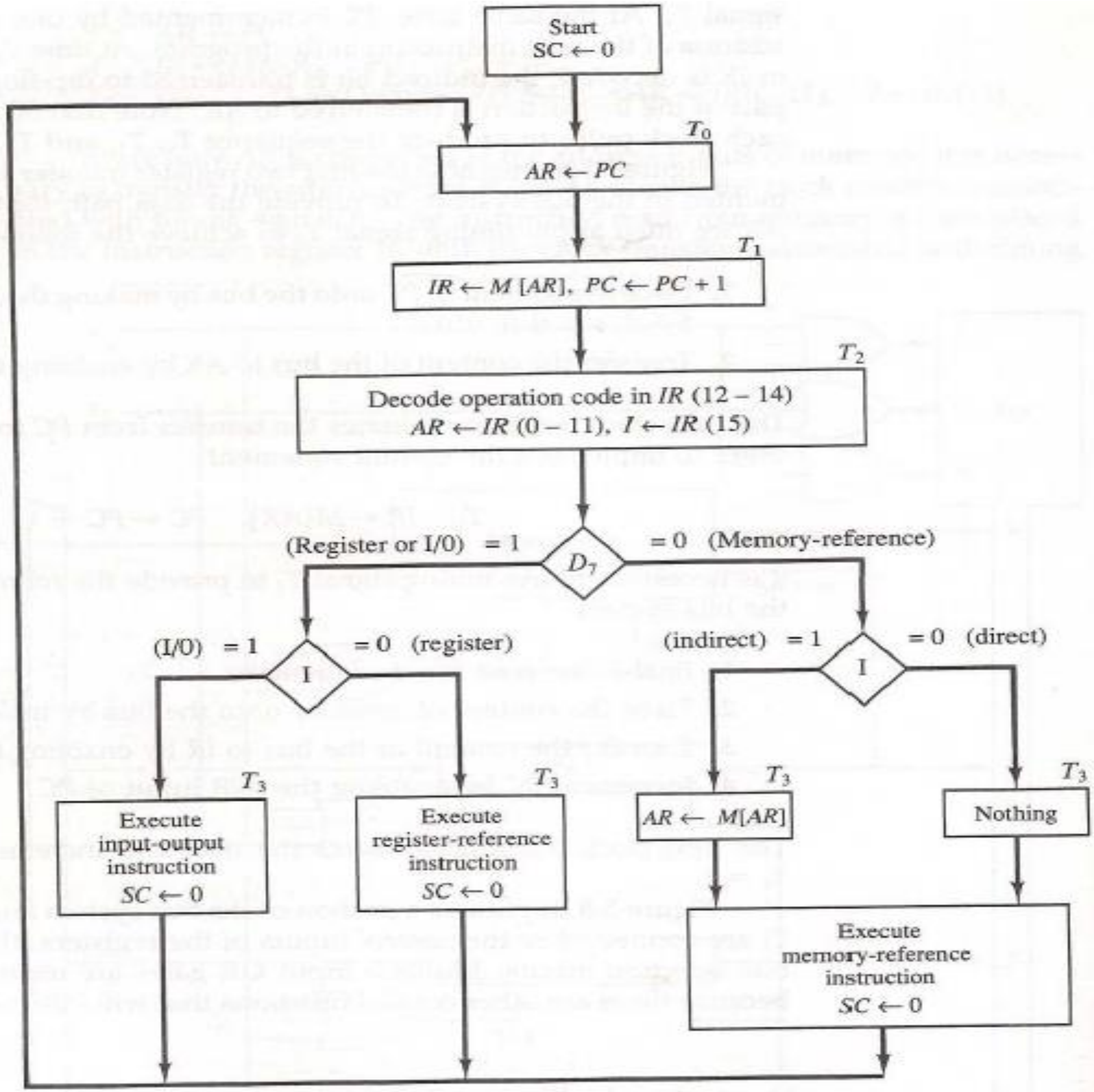
T_2: $D_0, \ldots, D_7 \leftarrow \text{Decode } IR(12\text{–}14), \quad AR \leftarrow IR(0\text{–}11), \quad I \leftarrow IR(15)$



T_1
T_0
S_2
S_1 Bus
S_0
Memory unit
7
Address
Read
AR
1
LD
PC
2
INR
IR
5
LD
Clock
Common bus

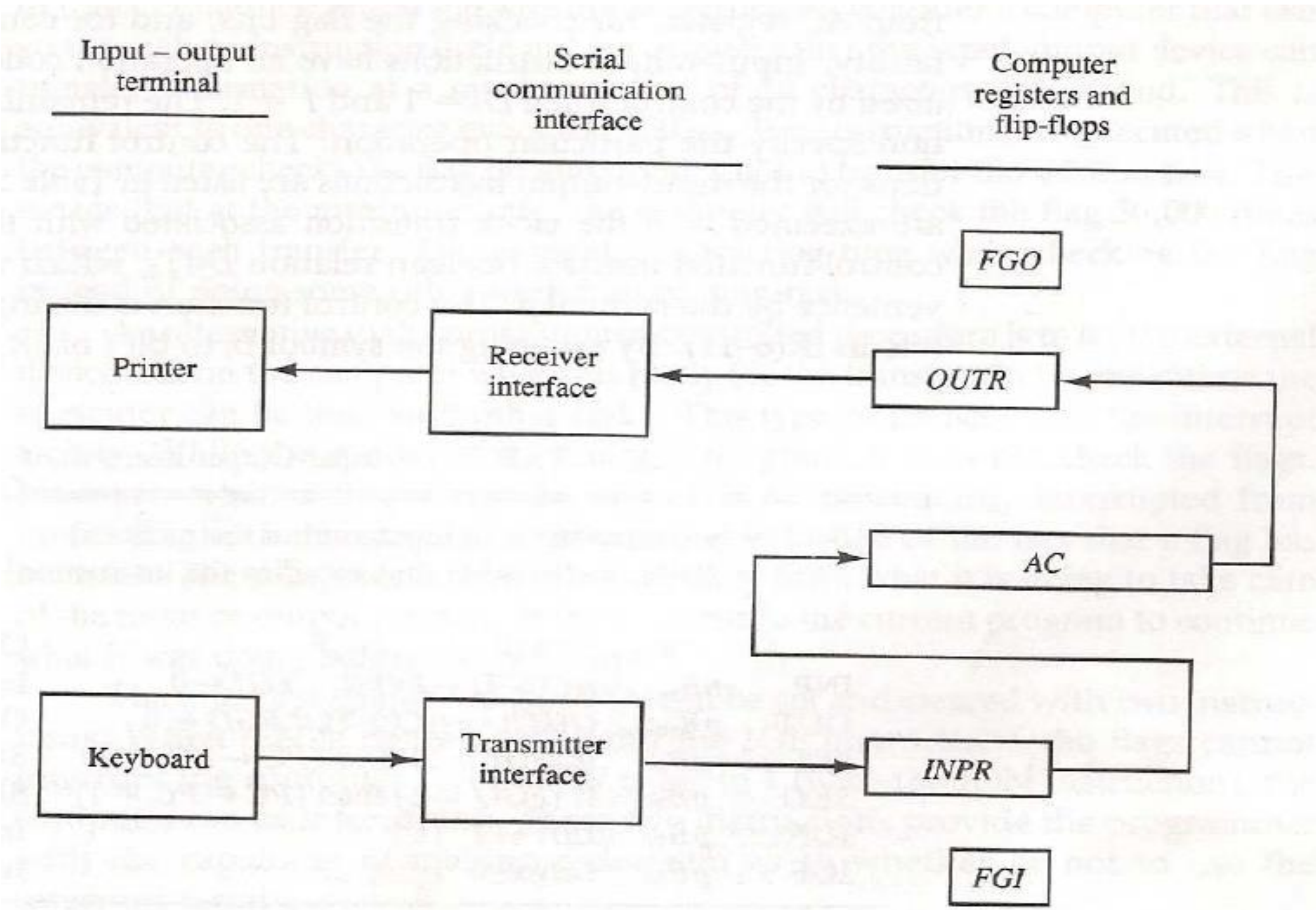
- To provide the data path for the transfer of PC to AR we must apply timing signal T_0 achieve the following:
  - Place the content of PC onto the bus by making the bus selection inputs $S_2S_1S_0$ equal to 010
  - Transfer the content of bus to AR by enabling the LD input of AR
- Then, next clock transition initiates the transfer of PC to AR since $T_0 = 1$

- To implement the next statement it is necessary to use timing signal T_1 to provide the following connections in the bus system
  - Enable the read input of memory
  - Place the content of memory onto bus by making $S_2S_1S_0$ equal to 111
  - Transfer the content of the bus to IR by enabling the LD input of IR
  - Increment the PC by enabling the INR input of PC



Start
SC ← 0
T_0
AR ← PC
T_1
IR ← M [AR], PC ← PC + 1
T_2
Decode operation code in IR (12 – 14)
AR ← IR (0 – 11), I ← IR (15)
(Register or I/0) = 1
D_7
= 0 (Memory-reference)
(I/0) = 1
I
= 0 (register)
(indirect) = 1
I
= 0 (direct)
T_3
Execute
input-output
instruction
SC ← 0
T_3
Execute
register-reference
instruction
SC ← 0
T_3
AR ← M[AR]
T_3
Nothing
Execute
memory-reference
instruction
SC ← 0

I/O Configuration



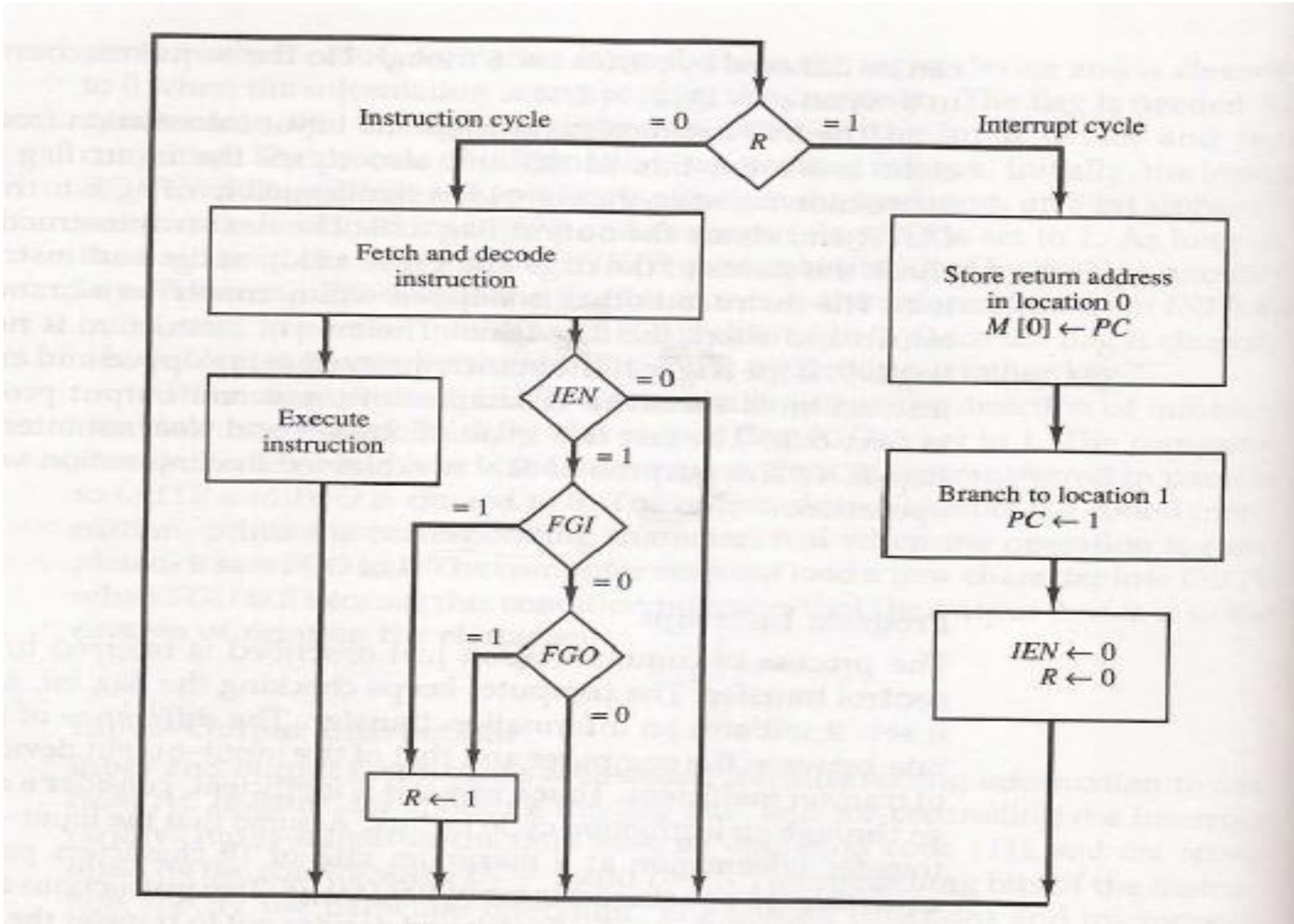
Program Interrupt

- The process of communication just discussed is referred to as programmed control transfer
- Computer keeps checking the flag bits, and if set, it initiates information transfer
- Difference of information flow rate between computer and that of I/O device makes this type of transfer inefficient
- Why???

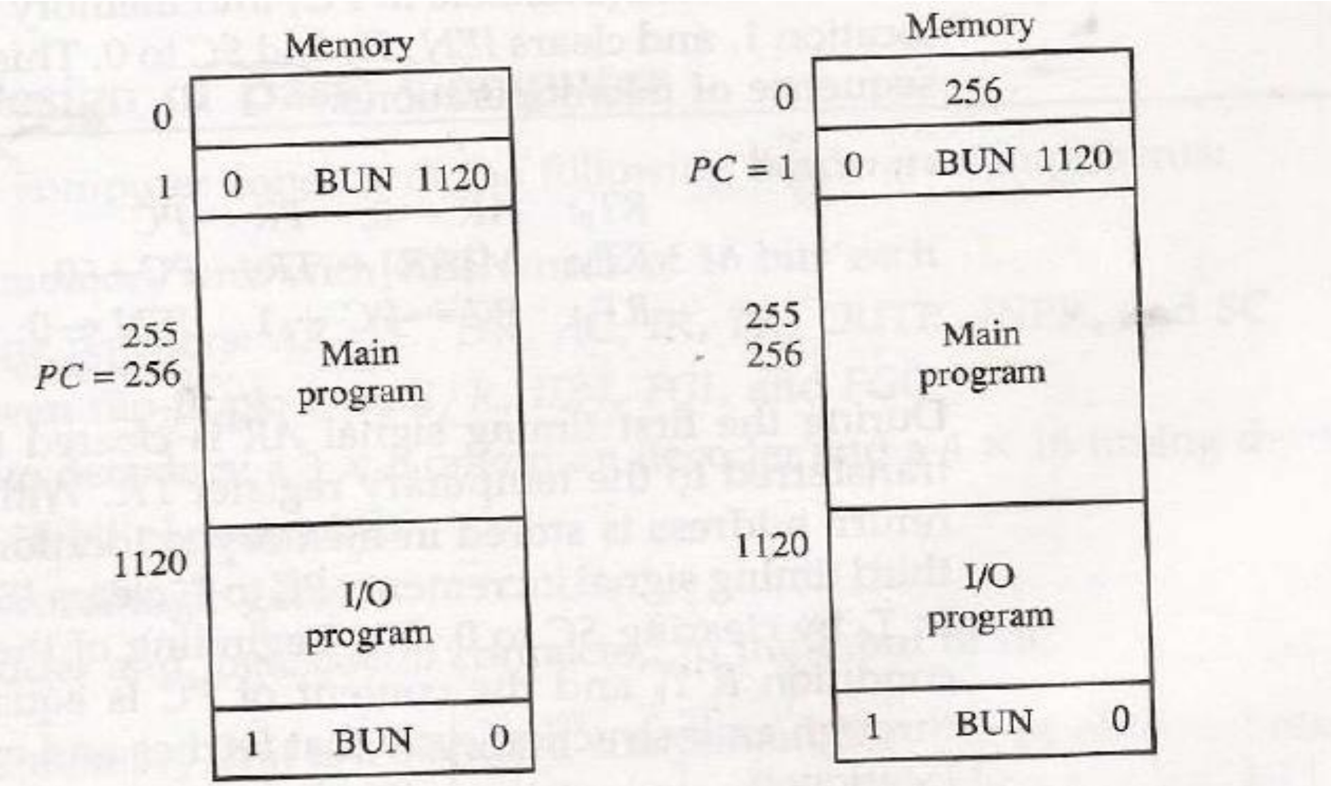
- Consider a computer with instruction cycle time 1 μs and I/O device can transfer information at a rate of 10 characters/sec
  - i.e 1 character every 100,000 μs
- Two instructions are executed when computer checks the flag bit and decides not to transfer information
  - i.e At max rate the computer will check the flag 50,000 times between each transfer
- Computer is wasting time while checking the flag instead of doing some other useful processing task

- Alternative to the programmed controlled procedure is to let the external device inform the computer when it is ready for transfer
- This type of transfer uses the interrupt facility
- IEN flip-flop helps to interrupt the processing of a computer program
- IEN=0, the interrupt cannot be made
- IEN=1, the interrupt can be made

Flowchart of Interrupt Cycle



- Interrupt cycle is h/w implementation of branch and save return address



(a) Before interrupt

(b) After interrupt cycle