COMPUTER ORGANIZATION & ARCHITECTURE

ASSIGNMENT –I

1. Given R=10, PC=20, and index register X=30 show the value of the accumulator for the following instructions. All memory locations Q contain the value Q+1. Each instruction uses two memory locations.
   a) LDAC 10
   b) LDAC @10
   c) LDAC R
   d) LDAC (R)
   e) LDAC #10
   f) LDAC $10
   g) LDAC 10(X)

Ans: a) *AC* = 11 b) *AC* = 12 c) *AC* = 10 d) *AC* = 11 e) *AC* = 10 f) *AC* = 33 g) *AC* = 41

2. Show the code to perform the computation X=**A+ (B*C) +D** using microprocessors that uses the following instruction formats. Do not modify the values of A, B, C, D. If necessary use temporary location T to store intermediate results
   a) Three address instruction
   b) Two address instruction
   c) One address instruction
   d) Zero address instruction

Ans: a)

MUL R, B, C	R← M [B]*M[C]
ADD R, A, R	R← M [A] +R
ADD X, R, D	M[X]← R+M [D]

b)

MOV R, B	R←M [B]
MUL R, C	R←R*M[C]
ADD R, A	R←R+M [A]
ADD R, D	R←R+M [D]
MOV X, R	M[X]←R

c)

LOAD	B	AC←M [B]
MUL	C	AC←AC*M[C]
ADD	A	AC←AC+M [A]
ADD	D	AC←AC+M [D]
STORE	X	M[X]←AC

d) RPN: ABC*+D+

PUSH A	TOS←A
PUSH B	TOS←B
PUSH C	TOS←C
MUL	TOS ←(B*C)
ADD	TOS ←A+ (B*C)
PUSH D	TOS← D
ADD	TOS← A+ (B*C) +D
POP X	M[X] ← TOS

3. Show the code to perform the computation X=**A*B*C+D*(E+F)** using microprocessors that uses the following instruction formats. Do not modify the values of A, B, C, D, E, F. If necessary use temporary location T to store intermediate results
   a) Three address instruction
   b) Two address instruction
   c) One address instruction
   d) Zero address instruction

a)

MUL R1, A, B	R1←M[A]*M[B]
MUL R1, R1, C	R1←R1*M[C]
ADD R2, E, F	R2←M [E] +M [F]
MUL R2, D, R2	R2←M [D] *R2
ADD X, R1, R2	M[X]←R1+R2

b)

MOV R1, A	R1←M[A]
MUL R1, B	R1←R1*M [B]
MUL R1, C	R1←R1*M[C]
MOV R2, E	R2←M [E]
ADD R2, F	R2←R2+M [F]
MUL R2, D	R2←R2*M [D]
ADD R1, R2	R1←R1+R2
MOV X, R1	M[X]←R1

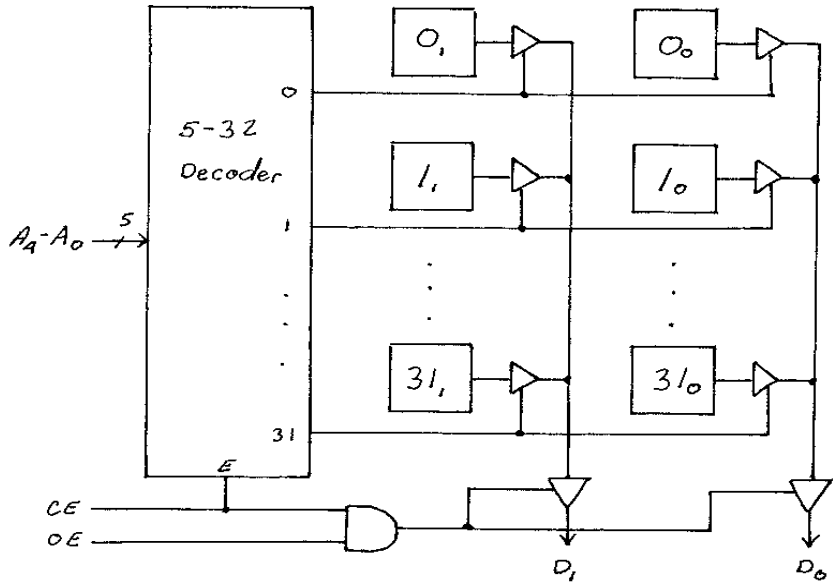
c)

LOAD A	AC←M[A]
MUL B	AC←AC*M [B]
MUL C	AC←AC*M[C]
STORE T	M [T]←AC
LOAD E	AC←M [E]
ADD F	AC←AC+M [F]
MULT D	AC←AC*M [D]
ADD T	AC←AC+M [T]
STORE X	M[X]←AC

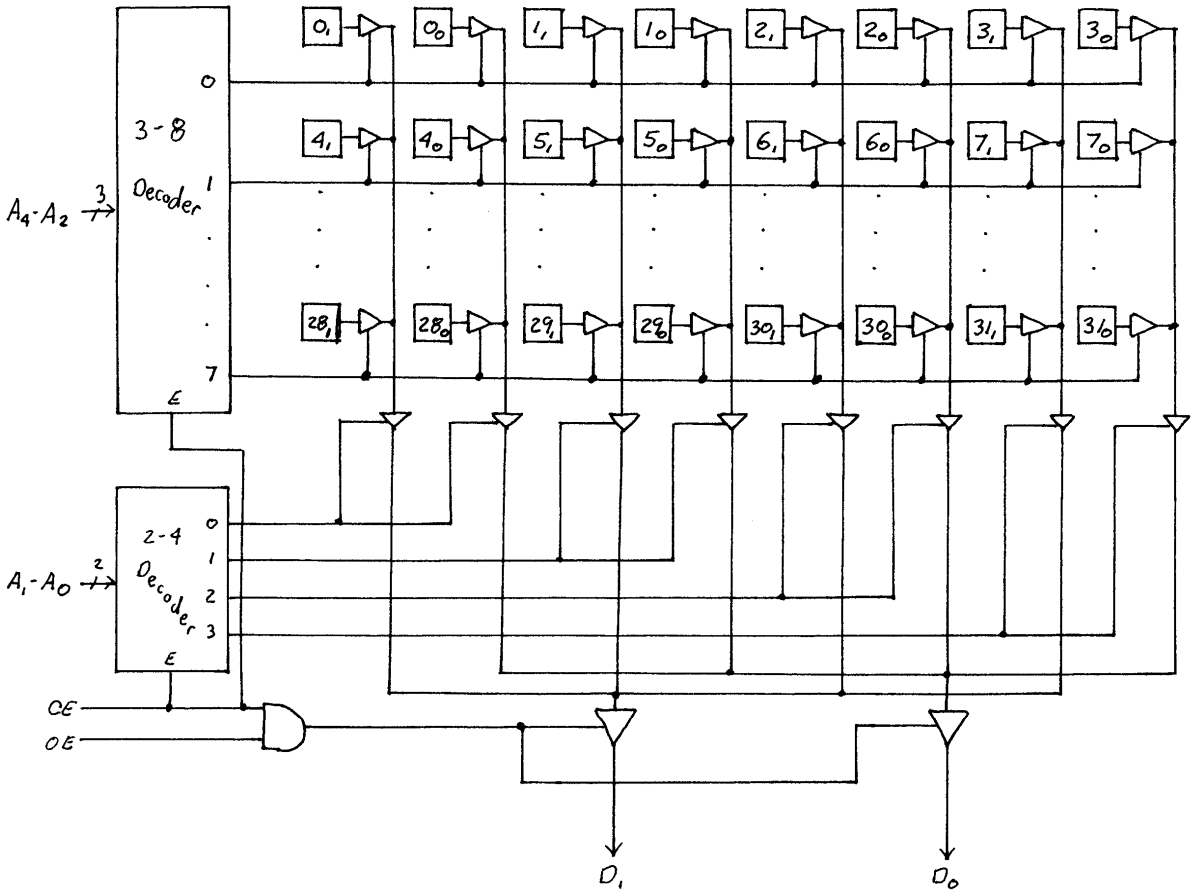
d) AB*C*DEF+*+

PUSH A	TOS←A
PUSH B	TOS←B
MUL	TOS ←(A*B)
PUSH C	TOS←C
MUL	TOS ←((A*B)*C)
PUSH D	TOS←D
PUSH E	TOS←E
PUSH F	TOS←F
ADD	TOS ←(E+F)
MUL	TOS ←(D*(E+F))
ADD	TOS ←((A*B)*C) + ((A*B)*C)
POP X	M[X]←TOS

4. Show the internal linear configuration of a 32 X 2 memory chip



5. Show the internal two-dimensional configuration of a 32 X 2 memory chip



6. Describe a simple computer architecture

⇒ A Relatively Simple Computer

- Consider a simple computer, that have 8K of ROM starting at address 0, followed by 8K of RAM. It will also have a memory mapped, bidirectional I/O port at address 8000H.

CPU

- for 16 bit addresses – 16 address pins – A_{15} to A_0
- data through D_7 to D_0, data pins.
- 2 control lines READ & WRITE.
- since it uses memory mapped I/O, it does not need a control signal such as IO/M.

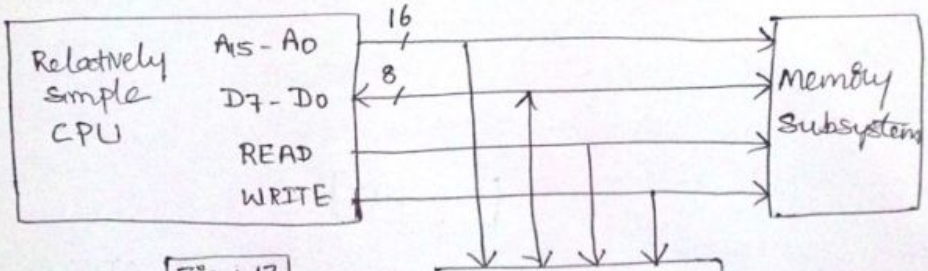


Fig 1.13

A Relatively Simple computer's CPU details only.

Memory Subsystem :-

- ROM contains address in the range 0 to 8K – 1
  0000 0000 0000 0000 to 0001 1111 1111 1111
- The 8K RAM immediately follows the ROM and has address 8K to 16K – 1
  0010 0000 0000 0000 to 0011 1111 1111 1111
- To access a memory chip, the processor must supply an address used by the chip, as well as the proper control signals.

- An 8K memory chip has 2^{13} internal memory locations.
- It has 13-bit address input to select one of these locations. (A_{12} to A_0 address bits).
- The remaining 3 bits (A_{15}, A_{14}, A_{13}) will be used to select one (or neither) of the memory.

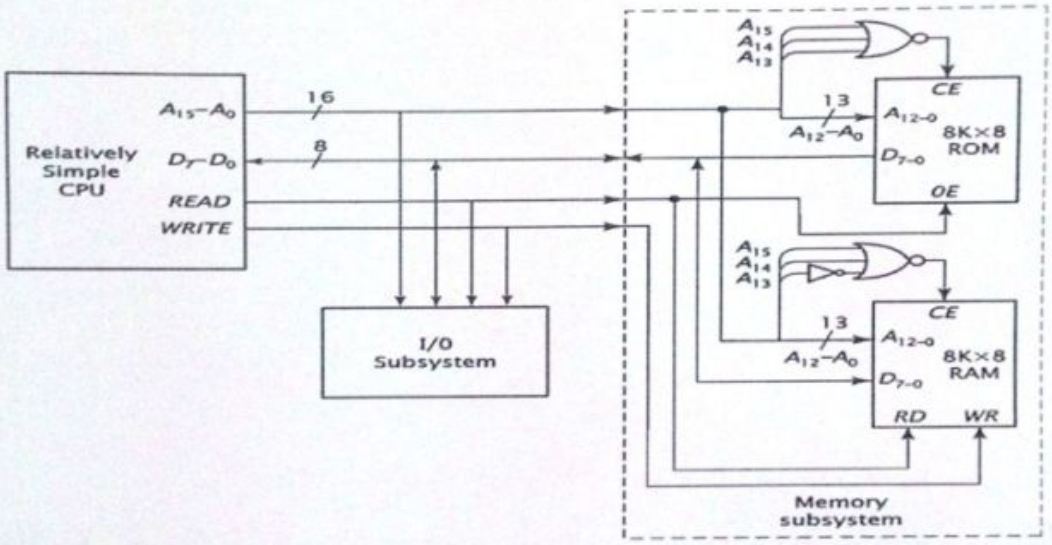


Fig 1.14 A RELATIVELY SIMPLE COMPUTER: MEMORY SUBSYSTEM DETAILS

- The top three address bits of ROM chip are 000 for all addresses.
- Similarly the RAM chip is enabled when $A_{15}A_{14}A_{13} = 001$.
- Read signal is connected to ROM's OE.
- ROM chip outputs data when CE & OE are enabled.
- The RAM has two control inputs, RD and WR. The READ and WRITE signals from the control bus can drive these two signals

→ I/O subsystem :-

- The system specifies one bidirectional I/O port at address 8000H or 1000 0000 0000 0000 in binary.
- We use the address and the control signals to enable the i/p & o/p interface circuitry for this port.

- We use a register to store data output to the device and tri-state buffers to pass through data to be input from the device.

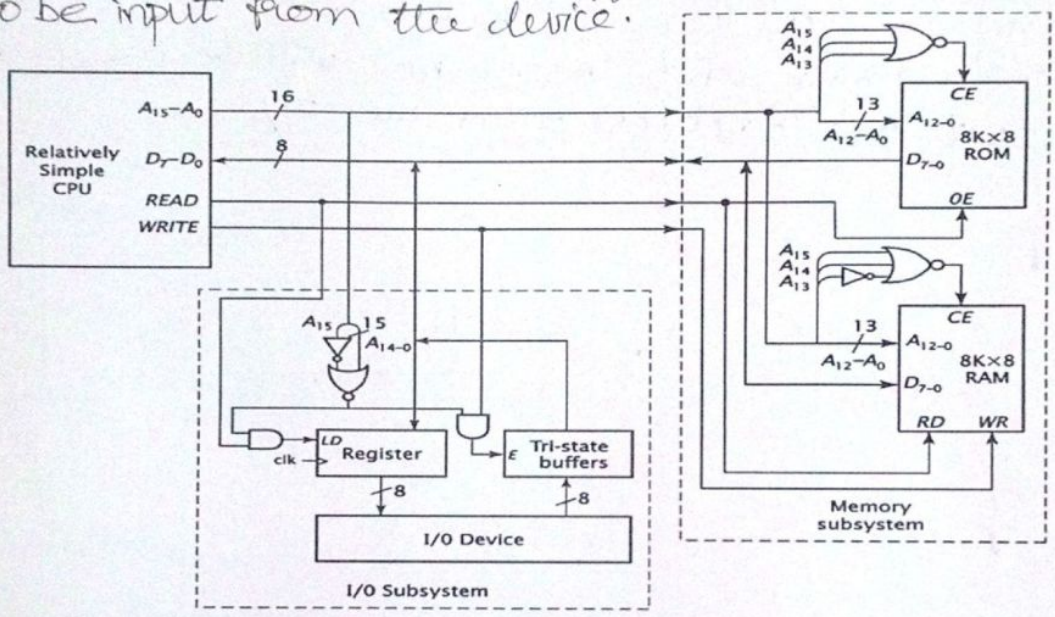


Fig 1.15 A RELATIVELY SIMPLE COMPUTER: FINAL DESIGN

- We need to design the address logic so that it recognizes 8000H.