Lecture-15

MACHINE CYCLES

OPCODE FETCH machine cycle:

Fig. shows the 8085 instruction fetch timing diagram. The instruction fetch cycle requires either four or six clock periods (T states). The other m/c cycles that follow OFMC will need three clock cycles.

The purpose of an OF is to read the contents of a memory location containing the opcode addressed by the program counter and to place it in the instruction register.

In the beginning of state T_1, the 8085A puts a low on the IO/$\overline{M}$ line of the system bus indicating a memory operation the 8085 sets $S_1 = 1$ and $S_0 = 1$ on the system bus, indicating the memory fetch operation this status information remains constant for the duration of the m/c cycle. During T_1 state, the 16 bit address $A_{15} - A_0$, of the memory location containing the op code is obtained from the program counter pc and placed in the address and address data latches the higher order 8 bits of the address appears on the address bus $A_8 - A_{15}$ remains constants until the end of the state T_3 during T_4 state the data on the address bus is unspecified. The low order 8 bits of the address is placed on the address/data bus, $AD_7 - AD_0$ at the beginning of T_1 this data however remains valid only until the beginning of state T_2 at which time the addr/data bus is floated (3 states) because this is time multiplexed bus and used on the data bus during $T_2 and$ T_3 state. Therefore addr latch enable

(ALE) signal issued by the μP during T_1 is used to latch this lower order addr in same external hardware 8212 on its falling edge the 16 bit addr select a particular memory location.

During state T_2, at the beginning. The $\overline{RD}$ single goes low indicating readf operation and the opcode to be fetched is placed on the data bus, $AD_7 - AD_0$, by the addressed memory location. The contents of (pc) is incremented be 1 during this state as during T_1 state the pc has sent the addr to addr bus . the access memory should be fast enough to output its data before $\overline{RD}$ goes high slower memories can gain more time by pulling the READY signal of 8085 LOW this will introduce an integral no. of T_{wait} sate between T_2 *and* T_1 as long as READY is low on the ring edge of the $\overline{RD}$ control signal in T_3 , the opcode obtaining from memory is transferred to the micro process instruction register.

During data T_4, the 8085 decodes the instruction and determines whether to enter state T_5 or to enter T_1 state of the next m/c cycle from the operation code the μP determines what other m/c cycles if any must be execute to complete the instruction cycle state T_5 & T_6, when entered, are used for internal μP operations necessitated by the instruction.

The micro RTL flow for 4 data OFMC is shown below.

OFMC: status IO/$\overline{M}$=0, $S_1 = 1$

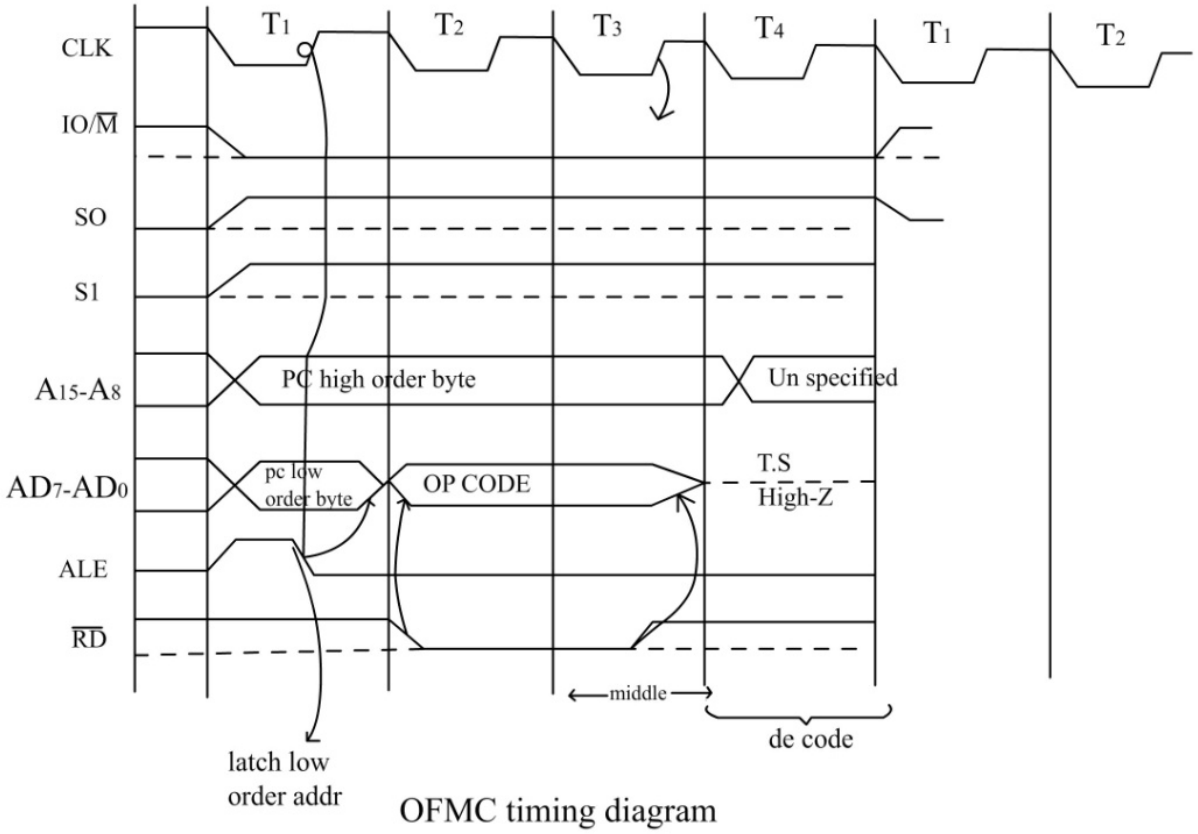
$S_0 = 1$

T_1: $AD_7 - AD_0$ ← (PCL), $A_{15} - A_8$ ← (PCH), ALE=

T_2 : $\overline{RD} = 0, (PC)$ ← (PC)+1, $AD_7 - AD_0$ ← M(AB)

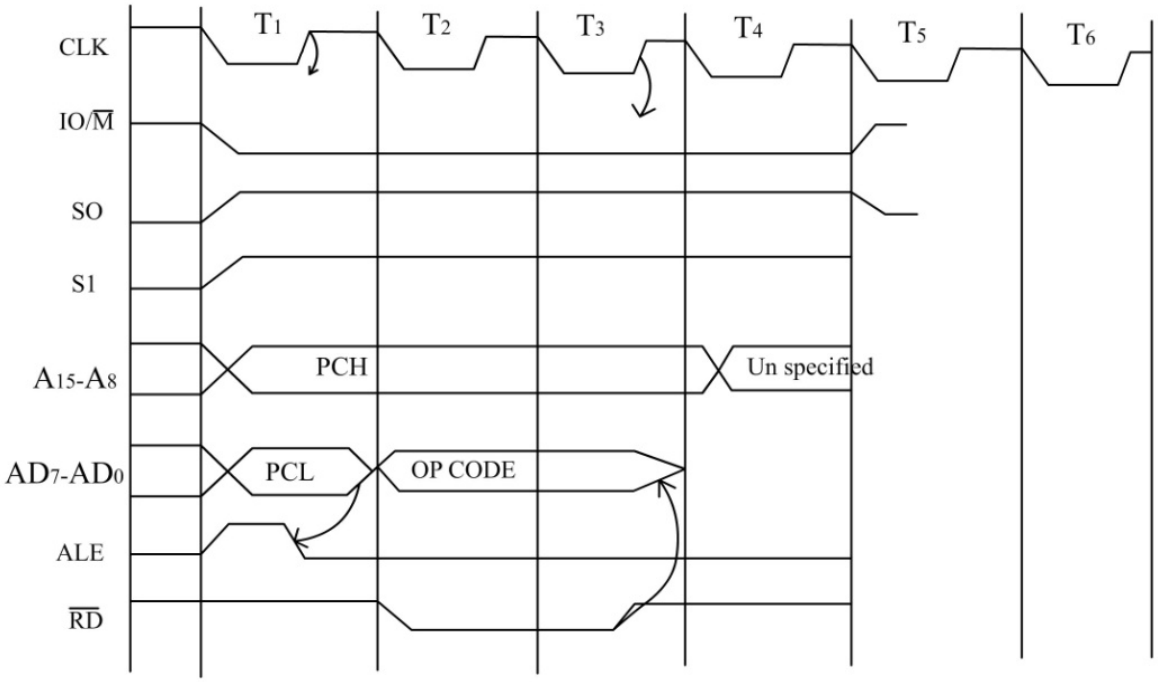
T_3 : $\overline{RD} = 1$, ↑ (IR) ← $(AD_7 - AD_0)$

T_4 : μP decodes the opcode and decides whether T_5 and T_6 states are required or next m/c cycle is executed. During T_4 –T_6 states, T_1 $AD_7 - AD_0$ = Tri-stated $A_{15} - A_0$ = unspecified



OFMC timing diagram

Fig below shows the timing diagram for a 6-state OFMC



Note: whenever the addr information is sent from the program counter to the external world during T_1 state, then the pc is incremented by 1 during the subsequent T_2 state so that pc points to the next subsequent byte. However if the addr information from pc has not been sent out during the T_1 state to the external until then pc will not be incremented during T_2 state.

Memory READ m/c cycle:

It requires 3 states T_1 to T_3 the purpose of the memory READ operation is to read the contents of a memory location addressed by a and place the data in a μP register by a register pair, the source of address issued during T_1 is not always the program

counter but may be any one of the several other register pairs in the μP depending on the particular instruction of which the m/c cycle is a part.

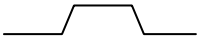
The 8085 uses m/c cycle MC1 to fetch and decode the instruction. It then performs the memory read operation in MC2. E.g. in LXIH, Addr.

The IO/$\overline{M}$ signal made low to indicate the external world that a memory reference is required. Then μP made $S_0 = o, \&\ S_1 = 1$ indicating that memory READ operation is to be performed. During the 8085 places the contents of high byte of the memory address register, such as that contents of the (PCH) or (H) register on lines $A_{15} - A_8$ and the contents of the low byte of the memory address register such as contents of the (PCL) or (L) reg. On lines $AD_7 - AD_0$. The 8085 sets ALE to HIGH, indicating the beginning of MC-2 AS soon as ALE goes to low, the 8085 latches the low byte of the address lines, since the same lines as going to be used as data lines.

During T_2 state, the $\overline{RD}$ signal goes LOW indicating a READ operation. If the addr sent and during T_1 state is from pc, then pc is incremented by 1 otherwise not the external logic gets the data from the memory location addressed by the memory address register such as (H,L) pair and places the data on to data bus $AD_7 - AD_0$.

During T_3 state, $\overline{RD}$ signal goes high, this low to high signal transfer the data from the data bus to internal register such as the accumulator.

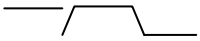
MRMC: status signals IO/$\overline{M}$=0, $S_0 = 0$ $S_1 = 1$

T_1: $AD_7 - AD_0$ ← (PCL) , $A_{15} - A_8$ ← (PCH), ALE= 

T_2 : $\overline{RD} = 0, (PC)$ ← (PC)+1, $AD_7 - AD_0$ ← M(AB)

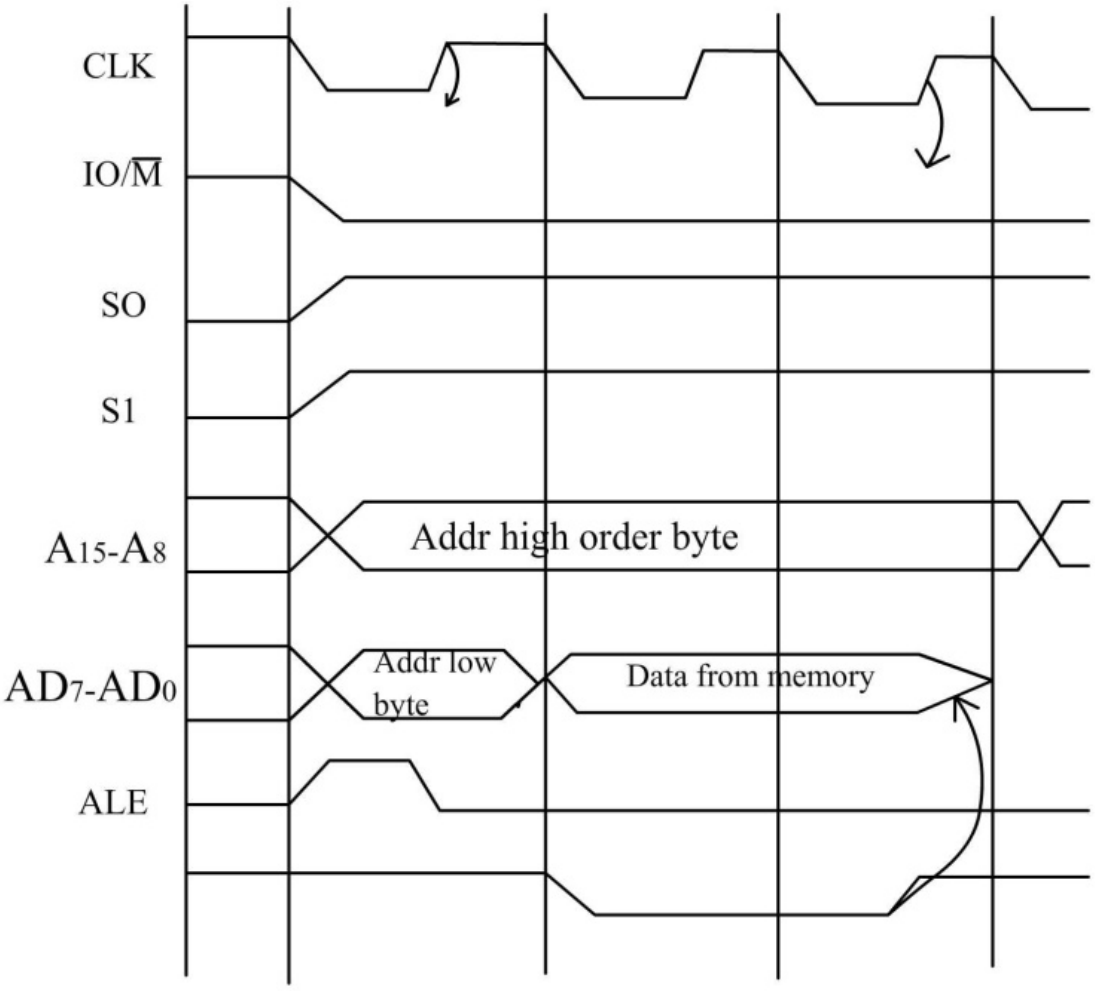
T_3 : $\overline{RD} = 1$, ↑, (internal reg.) ← $(AD_7 - AD_0)$

Or

T_1: $AD_7 - AD_0$ ← (PCL) , $A_{15} - A_8$ ← (PCH), ALE= 

T_2 : $\overline{RD} = 0$, $AD_7 - AD_0$ ← M(AB)

T_3 : $\overline{RD} = 1$, ↑ , (internal reg.) ← $(AD_7 - AD_0)$



MEMORY WRITE MC:

It also requires only T_1 to T_3 states the purpose of memory write is to store the contents of any of the 8085 reg. such as the accumulator into a memory location addressed by a register pair such as HL.

The 8085 μP made IO/$\overline{M}$=0 in the beginning of T_1 state to indicate memory reference operation then it puts $S_0 = 1$, $S_1 = 0$ indicates a memory write operation.

During T_1 state 8085 places the memory address register high byte such as the contents of the H register on lines $A_{15} - A_8$ and also places the MAR low byte such as the contents of the L-register on lines $AD_7 - AD_0$. the 8085sets ALE to HIGH, indicating the beginning of MWRMC. As soon as ALE goes to low, the 8085 latches the low byte of the address lines since the same lines are going to be used as data lines. During T_2 state, $\overline{WR}$ goes low indicating memory write operation It also places the contents of the internal register say accumulator on data lines $AD_7 - AD_0$.

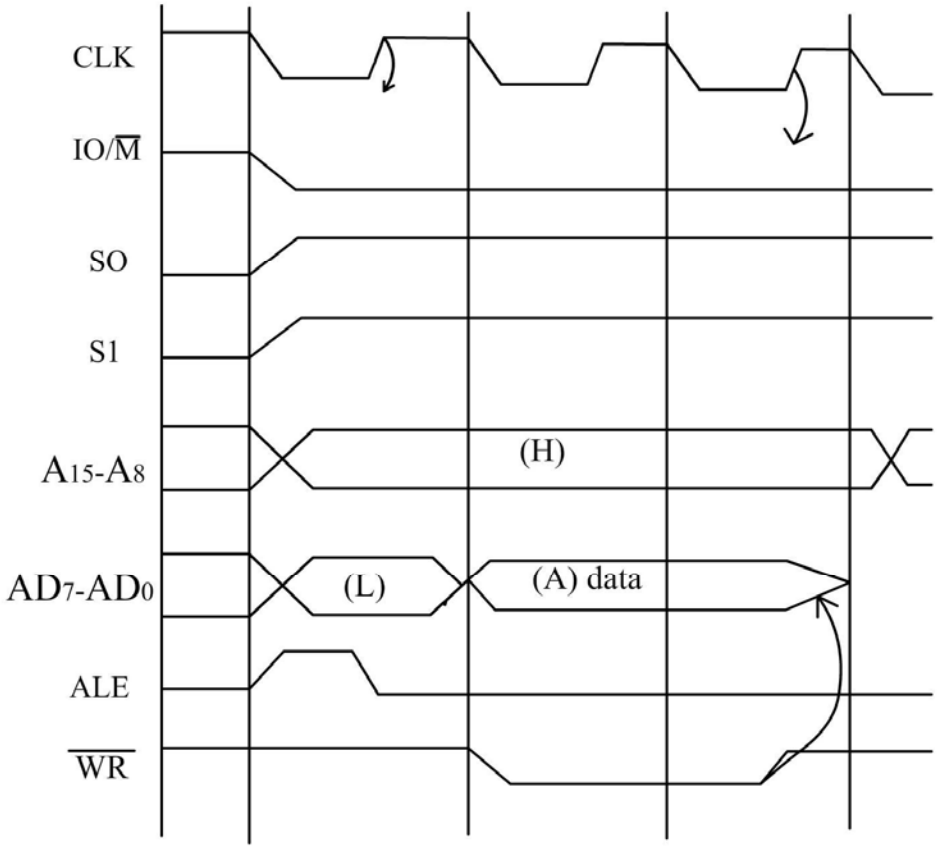
During T_3 state, $\overline{WR}$ goes high this low to high transition is used to transfer the data from the data lines to the memory location address by MAR such as HL reg. pair.

MWRMC: IO/$\overline{M}$=0, $S_1 = 0$ $S_0 = 1$

T_1: $AD_7 - AD_0$ ⟵ (L), $A_{15} - A_8$ ⟵ (H), ALE= ——/‾‾\——

T_2 : $\overline{MR} = 0$ $AD_7 - AD_0$ ⟵ (μP internal register)

T_3 : $\overline{WR} = 1$, ↑ M(AB) ⟵ $(AD_7 - AD_0)$



I/O READ and I/O WRITE M/C cycle:

These are identical to MRMC & MWRMC respectively except that appropriate status signals are issued at the beginning of T_1 state. IO/$\overline{M}$ signal goes high at the beginning to indicate I/O device reference is needed in case of I/O mapped input/output device in these m/c cycles higher & lower address bytes are identical and equal to the 8 bit address of the I/O port while in case of MRMC or MWRMC, the address bus output is the true 16bits address.