

As we have learnt earlier that address and data bus are multiplexed, we make use of control and status signals for data transfer. In this section we learn this concept in detail.

Control and Status Signals

- ALE – It is an Address Latch Enable signal. It goes high during first T state of a machine cycle and enables the lower 8-bits of the address, if its value is 1 otherwise data bus is activated.
- $\text{IO}/\overline{\text{M}}$ – It is a status signal which determines whether the address is for input-output or memory. When it is high (1) the address on the address bus is for input-output devices. When it is low (0) the address on the address bus is for the memory.
- S0, S1 – These are status signals. They distinguish the various types of operations such as halt, reading and instruction fetching or writing.
- $\overline{\text{RD}}$ – It is a signal to control READ operation. When it is low the selected memory or input-output device is read.
- $\overline{\text{WR}}$ – It is a signal to control WRITE operation. When it goes low the data on the data bus is written into the selected memory or I/O location.
- READY – It senses whether a peripheral is ready to transfer data or not. If READY is high (1) the peripheral is ready. If it is low (0) the microprocessor waits till it goes high. It is useful for interfacing low speed devices.

| IO/M' | S1 | S0 | Data Bus Status |
|-------|----|----|-----------------------|
| 0 | 1 | 1 | Opcode fetch |
| 0 | 1 | 0 | Memory read |
| 0 | 0 | 1 | Memory write |
| 1 | 1 | 0 | I/O read |
| 1 | 0 | 1 | I/O write |
| 1 | 1 | 1 | Interrupt acknowledge |
| 0 | 0 | 0 | Halt |

Important Note 8085 works at half the frequency of crystal frequency. Therefore, to operate a system at 3MHz the crystal should have frequency of 6MHz.

INPUT/OUTPUT

Input/output devices are the means through which the MPU communicates with “the out-side world.” The MPU accepts binary data as input from devices such as keyboards and A/D converters and sends data to output devices such as LEDs or printers. There are two different methods by which I/O devices can be identified: one uses an 8-bit address and the other uses a 16-bit address. These methods are described briefly in the following sections.

I/Os with 8-Bit Addresses (Peripheral-Mapped I/O)

In this type of I/O the MPU uses eight address lines to identify an input or an output device; this is known as peripheral-mapped I/O (also known as I/O-mapped I/O). This is an 8-bit numbering system for I/Os used in conjunction with Input and Output instructions. This is also known as I/O space, separate from memory space, which is a 16-bit numbering system. The eight address lines can have 256 (2^8 combinations) addresses; thus, the MPU can identify 256 input devices and 256 output devices with addresses ranging from 00H to FFH. The input and output devices are differentiated

by the control signals; the MPU uses the I/O Read control signal for input devices and the I/O Write control signal for output devices. The entire range of I/O addresses from 00 to FF is known as an I/O map, and individual addresses are referred to as I/O device addresses or I/O port numbers.

I/Os with 16-Bit Addresses (Memory-Mapped I/O)

In this type of I/O the MPU uses 16 address lines to identify an I/O device; an I/O is connected as if it is a memory register. This is known as memory-mapped I/O. The MPU uses the same control signal (Memory Read or Memory Write) and instructions as those of memory. In memory-mapped I/O, the MPU follows the same steps as if it is accessing a memory register.

Instruction cycle is defined as the time required to complete the execution of an instruction. The 8085 instruction cycle consists of one to six machine cycles or one to six operations.

Machine cycle is defined as the time required to complete one operation of accessing memory, I/O, or acknowledging an external request. This cycle may consist of three to six T states.

T-state is defined as one subdivision of the operation performed in one clock period. These subdivisions are internal states synchronized with the system clock, and each T-state is precisely equal to one clock period. The terms T-state and clock period are often used synonymously.

Timing Diagram

MOV

Problem

Draw the timing diagram of the given instruction in 8085, MOV B, C

Given instruction copies the contents of the source register into the destination register and the contents of the source register are not altered.

Example

MOV B, C

Opcode: MOV

Operand: B and C

B is the destination register and C is the source register whose contents need to be transferred to the destination register.

Algorithm

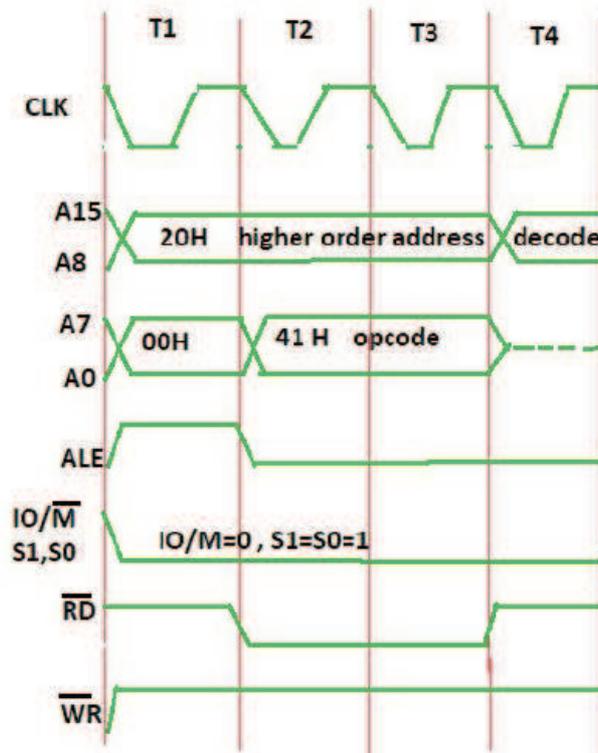
The instruction MOV B, C is of 1 byte; therefore the complete instruction will be stored in a single memory address.

For example:

2000: MOV B, C

Only opcode fetching is required for this instruction and thus we need 4T states for the timing diagram. For the opcode fetch the IO/M (low active) = 0, S1 = 1 and S0 = 1.

The timing diagram of MOV instruction is shown below:



In Opcode fetch (T1-T4 T states)

00 – Lower bit of address where opcode is stored, i.e., 00

20 – Higher bit of address where opcode is stored, i.e., 20.

ALE – provides signal for multiplexed address and data bus. Only in T1 it is used as address bus to fetch lower byte of address otherwise it will be used as data bus.

RD (low active) – signal is 1 in T1 & T4 as no data is read by microprocessor.

Signal is 0 in T2 & T3 because here the data is read by microprocessor.

WR (low active) – signal is 1 throughout, no data is written by microprocessor.

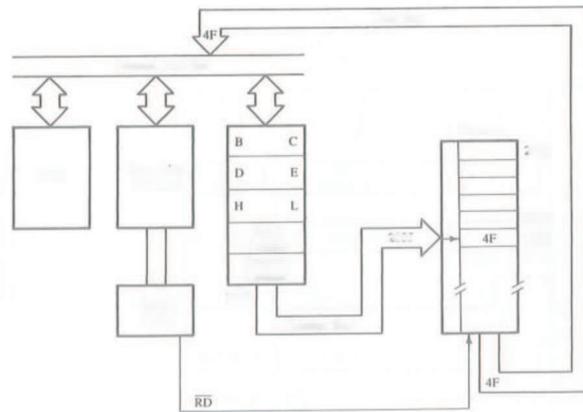
IO/M (low active) – signal is 1 in throughout because the operation is performed on memory.

S0 and S1 – both are 1 in case of opcode fetching.

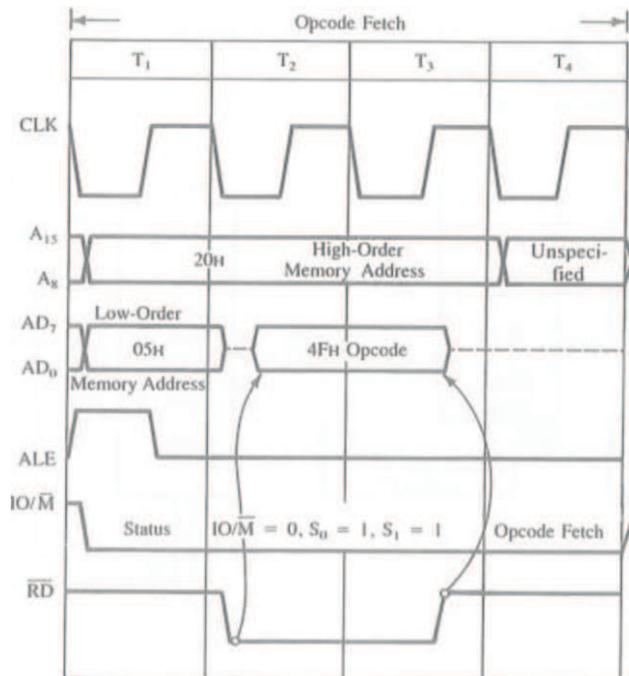
Example

Illustrate the steps and the timing of data flow when the instruction code 0100 1111 (4FH—MOV C,A), stored in location 2005H, is being fetched.

To fetch the byte (4FH), the MPU needs to identify the memory location 2005H and enable the data flow from memory. This is called the Fetch cycle. The data flow is shown in Figure below:



Now look at the following figure:



The above figure shows the timing of how a data byte is transferred from memory to the MPU; it shows five different groups of signals in relation to the system clock.

The address bus and data bus are shown as two parallel lines. This is a commonly used practice to represent logic levels of groups of lines; some lines are high and others are low. The crossover of the lines indicates that a new byte (information) is placed on the bus, and a dashed straight line indicates the high impedance state.

To fetch the byte, the MPU performs the following steps:

Step 1: The microprocessor places the 16-bit memory address from the program counter (PC) on the address bus.

The figure shows that at T1, the high-order memory address 20H is placed on the address lines A15-A8, the low-order memory address 05H is placed on the bus AD7-AD0, and the ALE signal goes high. Similarly, the status signal IO/M goes low, indicating that this is a memory-related operation.

Step 2: The control unit sends the control signal RD to enable the memory chip. The control signal

RD is sent out during the clock period T2, thus enabling the memory chip. The RD signal is active during two clock periods.

Step 3: The byte from the memory location is placed on the data bus. When the memory is enabled, the instruction byte (4FH) is placed on the bus AD7-AD0 and transferred to the microprocessor. The RD signal causes 4FH to be placed on bus AD7-AD0 (shown by the arrow), and when RD goes high, it causes the bus to go into high impedance.

Step 4: The byte is placed in the instruction decoder of the microprocessor, and the task is carried out according to the instruction. The machine code or the byte (4FH) is decoded by the instruction decoder, and the contents of the accumulator are copied into register C. This task is performed during the period T4.