BUS CYCLE AND TIME STATE

A *bus cycle* defines the basic operation that a microprocessor performs to communicate with external devices.
Example of bus cycles are

- Memory read
- Memory write
- IO read
- IO write

a bus cycle corresponds to a sequence of events that start with an address being output on the system bus followed by a read or write data transfer. During these operations, the MPU produces a series of control signals to control the direction and timing of the bus.
The bus cycle of 8086 microprocessors consists of at least four clock periods (T_1, T_2, T_3, and T_4):

- the 8086 puts an address on the bus During T_1.
- *For write memory cycle*: During T_2 the 8086puts the data on the bus and maintained through T_3 and T_4.
- *For read cycle*: During T_2 the 8086puts the bus in high-Z state(high impedance) and then the data to read must be available on the bus during $T_3$and T_4.

Ex: if the frequency of the MPU system is 8MHz then the four clock states give a bus cycle duration of 125 ns × 4= 500 ns in an.

- **Idle States**

If no bus cycles are required, the microprocessor performs what are known as *idle state*. During these states, no bus activity takes place. Each idle state is one clock period long, and any number of them can be inserted between bus cycles. Idle states are performed if the instruction queue inside the microprocessor is full and it does not need to read or write operands form memory.

- **Wait States**

Wait states can be inserted into a bus cycle. This is done in response to request by an event in external hardware instead of an internal event such as a full queue. The READY input of the 8086is provided specifically for this purpose. As long as READY is held at the 0 level, wait states are inserted between states $T_3$and T_4 of the current bus cycle, and the data that were on the bus during T_3 are maintained. The bus cycle is not completed until the external hardware returns READY back to the 1 logic level.

Ex: What is the duration of the bus cycle in the 8086 - based microcomputer if the clock is 8 MHz and two wait states are inserted?

Solution:

The duration of the bus cycle in an 8-MHz system is given in general by

t_{cyc} = 500 ns + N * l25ns

In this expression N stands for the number of wait states. For a bus cycle with two wait states, we get

t_{cyc} = 500 ns + 2 * l25 ns = 500 ns + 250 ns = 750 ns

HARDWARE ORGANIZATION OF THE 8086 MEMORY ADDRESS SPACE

the 8086's 1Mbyte memory address space, as shown in Fig. -1-, is implemented as two independent 512Kbyte banks: the low (even) bank and the high (odd) bank. Data bytes associated with an even address (00000H, 00002H, etc.) reside in the low bank, and those with odd addresses (00001H, 00003H, etc.) reside in the high bank.

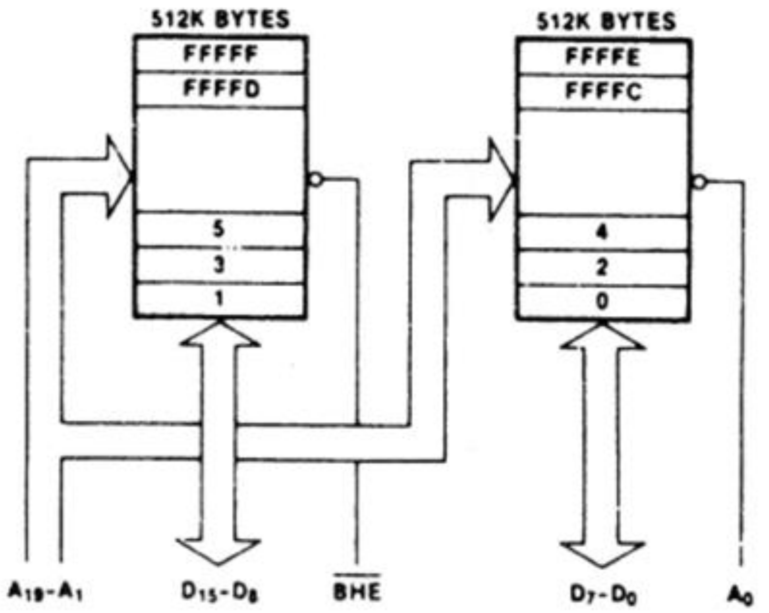

Fig-1-

The diagram in Fig. -1- shows that for the 8086 address bits, A_1 through A_{19}

select the storage location that is to be accessed. They are applied to both banks in parallel. A_0 and bank high enable $\overline{BHE}$ are used as bank-select signals. Logic 0 at A_0 identifies an even-addressed byte of data and causes the low bank of memory to be enabled. On the other hand, $\overline{BHE}$ equal to 0 enables the high bank to access an odd-addressed byte of data. Each of the memory banks provides half of the 8086's l6-bit data bus. Notice that the lower bank transfers bytes of data over data lines D0 through D7, while data transfers for a high bank use D8 through D15.

Figure-2- shows four different cases that happen during accessing data:

1. When a **byte** of data at an even address (such as X) is to be accessed:
   - A_0 is set to logic 0 to enable the low bank of memory.
   - $\overline{\text{BHE}}$is set to logic 1 to disable the high bank. (Figure-2-(a))

2. When a **byte** of data at an odd address (such as X+1) is to be accessed:
   - A_0 is set to logic 1 to disable the low bank of memory.
   - $\overline{\text{BHE}}$is set to logic 0 to enable the high bank. (Figure -2-(b)).

3. When a **word** of data at an even address (**aligned word**) is to be accessed:
   - A_0 is set to logic 0 to enable the low bank of memory.
   - $\overline{\text{BHE}}$ is set to logic 0 to enable the high bank. (Figure -2-(c)).

   Note:

   - Whenever an even-addressed word of data is accessed, both the high and low banks

   are accessed at the same time.

   - The bytes of an even-addressed word are said to be aligned and can be transferred with a memory operation that takes just one bus cycle.

4. When a **word** of data at an odd address (**misaligned word**) is to be accessed the 8086 need two bus cycles to access it (Figure -2-(d)):

   a. During the first bus cycle, the odd byte of the word (in the high bank) is addressed:
      - A_0 is set to logic 1 to disable the low bank of memory.
      - $\overline{\text{BHE}}$ is set to logic 0 to enable the high bank.

   b. During the second bus cycle, the odd byte of the word (in the low bank) is addressed:

- A_0 is set to logic 0 to enable the low bank of memory.
- $\overline{\text{BHE}}$ is set to logic 1 to disable the high bank.

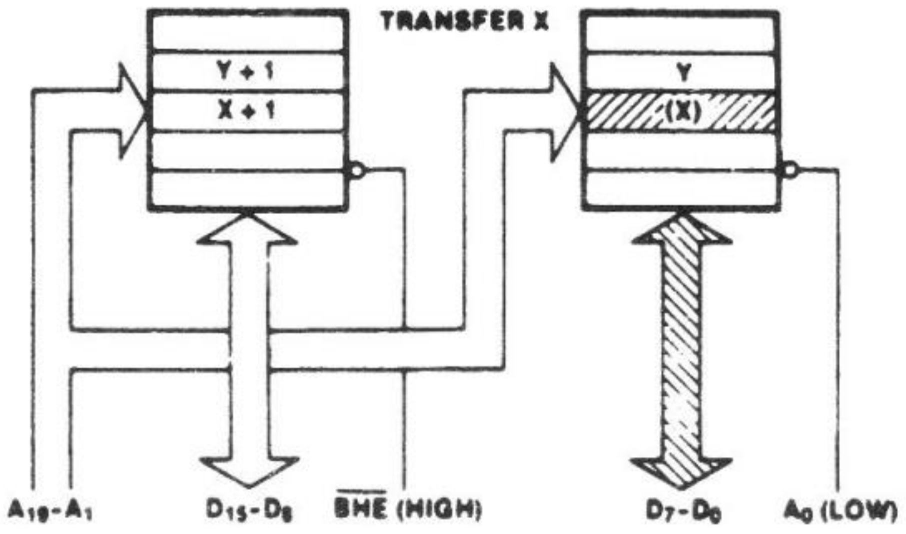

Fig-2- (a)

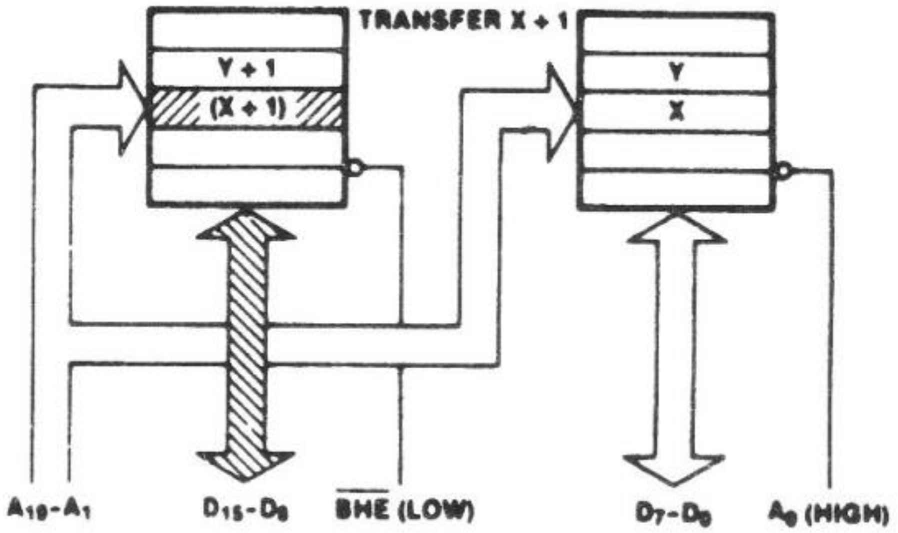

Fig-2- (b)

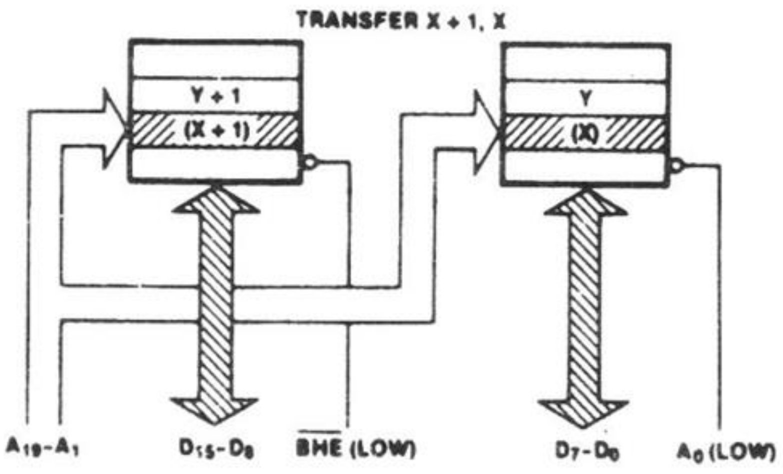
TRANSFER X + 1, X
Y + 1
(X + 1)
Y
(X)
A19-A1
D15-D8
BHE (LOW)
D7-D0
A0 (LOW)

Fig-2-(c)

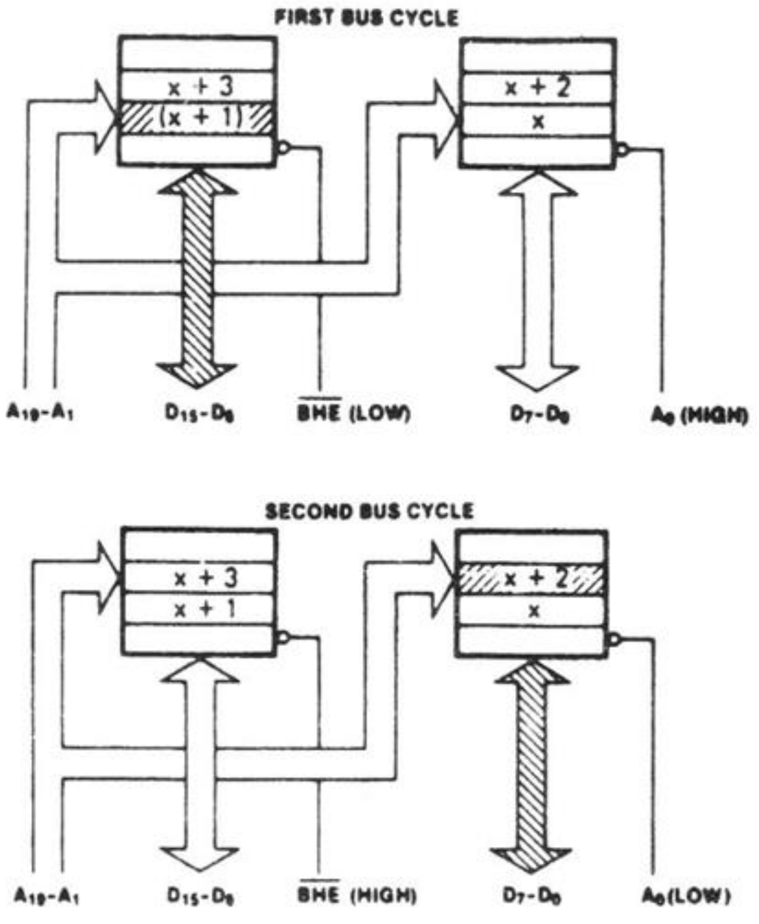
FIRST BUS CYCLE
x + 3
(x + 1)
x + 2
x
A19-A1
D15-D8
BHE (LOW)
D7-D0
A0 (HIGH)
SECOND BUS CYCLE
x + 3
x + 1
x + 2
x
A19-A1
D15-D8
BHE (HIGH)
D7-D0
A0 (LOW)

Fig-2-(d)