EC 305 MICROPROCESSOR & MICROCONTROLLER MODULE 3

SOUMYA S
AP IN ECE
CEMP

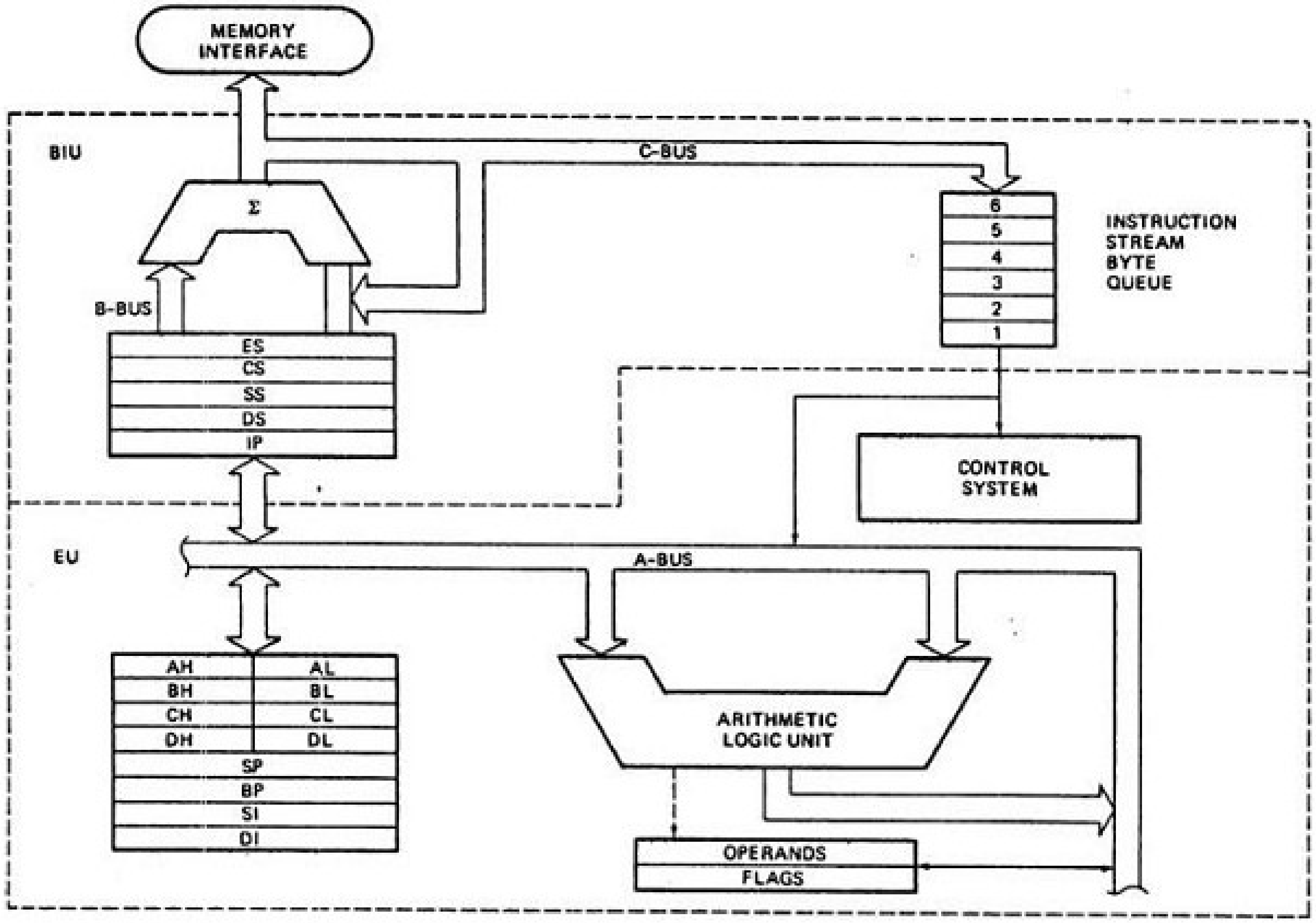
Microprocessor - 8086

- 8086 Microprocessor is an enhanced version of 8085 Microprocessor that was designed by Intel in 1976.
- It is a 16-bit Microprocessor having 20 address lines and16 data lines that provides up to 1MB storage.
- It consists of powerful instruction set, which provides operations like multiplication and division easily.
- It supports two modes of operation, i.e. Maximum mode and Minimum mode.
- Maximum mode is suitable for system having multiple processors .
- Minimum mode is suitable for system having a single processor.

Features of 8086

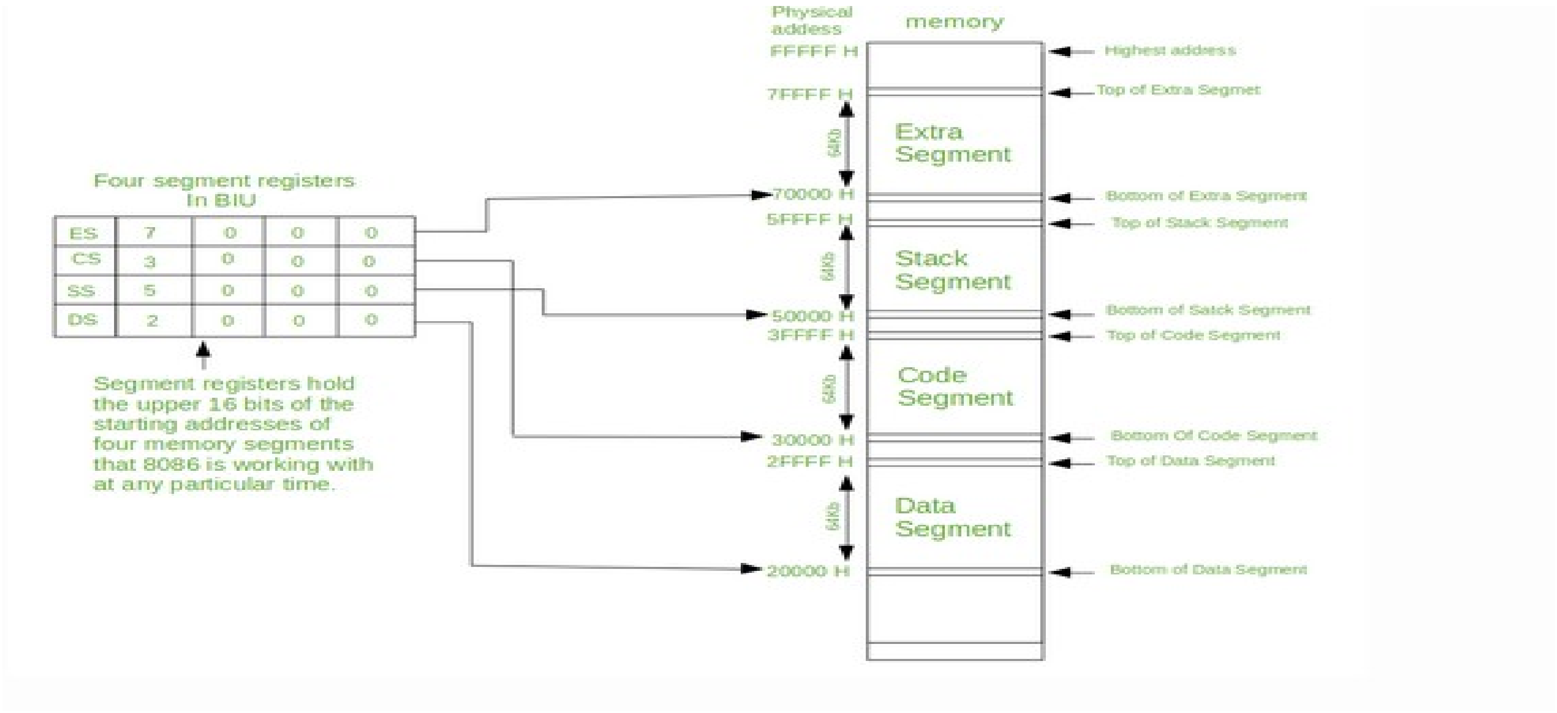
- It has an instruction queue, which is capable of storing six instruction bytes from the memory resulting in faster processing.
- It was the first 16-bit processor having 16-bit ALU, 16-bit registers, internal data bus, and 16-bit external data bus resulting in faster processing.
- It is available in 3 versions based on the frequency of operation −
  - 8086 → 5MHz
  - 8086-2 → 8MHz
  - (c)8086-1 → 10 MHz
- It uses two stages of pipelining, i.e. Fetch Stage and Execute Stage, which improves performance.
- Fetch stage can prefetch up to 6 bytes of instructions and stores them in the queue.
- Execute stage executes these instructions.

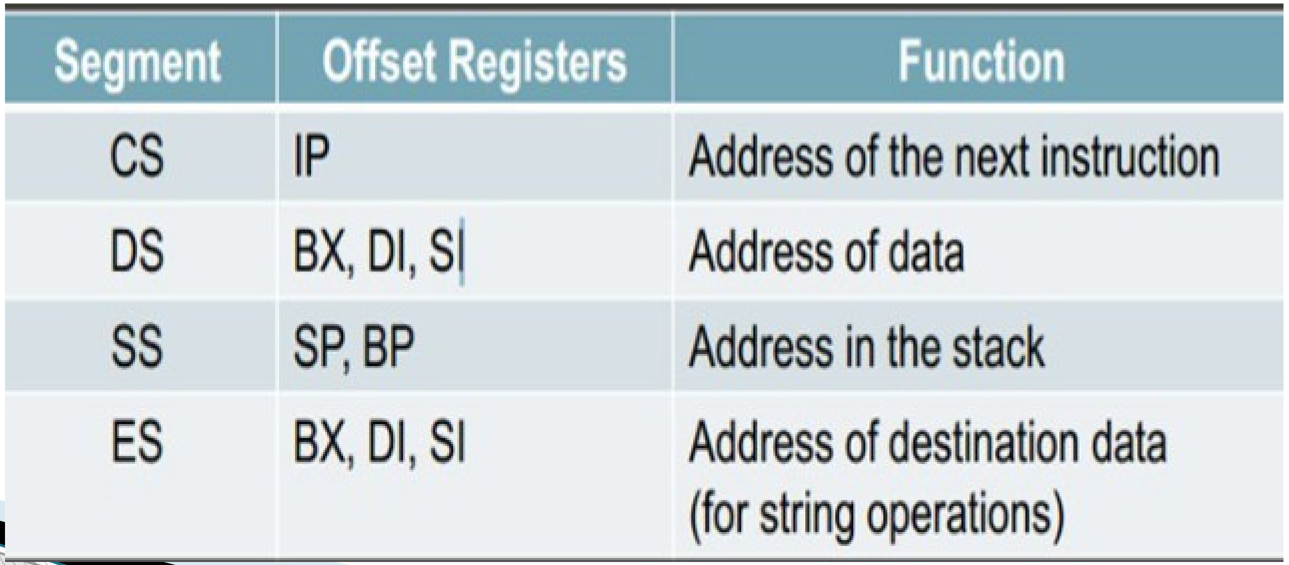
Architecture of 8086



- Divided into 2 independent units
  - Bus interface unit
  - Execution unit
- <u>Bus interface unit</u>
- <u>functions</u>
  - Fetch the instruction or data from memory.
  - Write the data to memory or I/O ports.
  - Read the data from memory or I/O ports.
- 3 functional parts
  - Instruction pointer(IP)
  - Segment registers
  - Instruction queue

- **Instruction pointer** − It is a 16-bit register used to hold the address of the next instruction to be executed.
- **Segment register** − The memory space 1 MB of 8086 is segmented into 4 blocks BIU has 4 segment buses,
- Code segment CS
- Data Segment DS
- Stack Segment SS
- Extra Segment ES.
- **Instruction queue -** BIU contains the instruction queue. BIU gets upto 6 bytes of next instructions and stores them in the instruction queue. When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.





Segment	Offset Registers	Function
CS	IP	Address of the next instruction
DS	BX, DI, SI	Address of data
SS	SP, BP	Address in the stack
ES	BX, DI, SI	Address of destination data (for string operations)

- each Segment has a corresponding 16-bit Segment Register which holds the Base Address (starting Address) of the Segment
- 8086 has 20bit address line. So the maximum value of address that can be addressed by 8086 is 2^20 = 1MB. So 8086 can address the locations ranging between 00000 H to FFFFF H. This 1MB memory is divided into 16 logical segments, each with a memory of 64KB.
- To locate any address in the memory bank, it needs the Physical address of that memory location. It cannot get the 20-bit Physical address using the 8086 Address Line or 16-bit Segment Registers alone.
- In order to access memory location, you cannot pass 20-bit address directly to the processor. You need to tell the 16-bit address with respect to the segment. This 16-bit address with respect to the part (segment of 64KB) of the memory bank is called the offset.
- So, Physical Address = Base Address + Offset.

- 1) The value of Code Segment (CS) Register is 4042H and the value of different offsets is as follows: BX: 2025H , IP: 0580H , DI: 4247H Calculate the effective address of the memory location pointed by the CS register.
- **Ans:**
- The offset of the CS Register is the IP register.
- Therefore, the effective address of the memory location pointed by the CS register is calculated as follows:
- Effective address= Base address of CS register X 10_H + Address of IP
- = 4042_H X 10_H + 0580_H = $(40420 + 0580)_H$ = 41000_H

- **EU (Execution Unit)**
- Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions.
- Its function is to control operations on data using the instruction decoder & ALU.

Functional parts

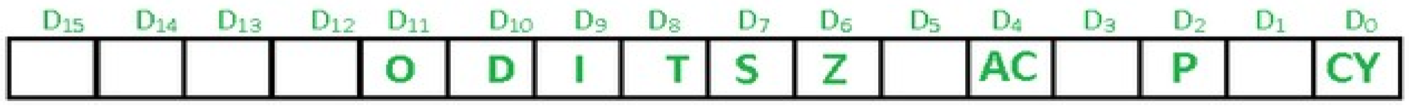
- General purpose reg
- Pointer and index reg
- ALU
- Flag reg
- Timing and Control Unit

- **General purpose register**
- Consist of 4 general purpose regs AX,BX,CX,DX
- There are 8 general purpose registers.
- i.e. AH, AL, BH, BL, CH, CL, DH, and DL.
- These registers can be used individually to store 8-bit data and can be used in pairs to store 16bit data.
- **AX register** − It is also known as accumulator register. It is used to store operands for arithmetic operations.
- **BX register** − It is used as a base register. It is used to store the offset address of the memory area within the data segment.
- **CX register** − It is referred to as counter. It is used in loop instruction to store the loop counter.
- **DX register** − This register is used to hold I/O port address for I/O instruction.

- **Pointer &Index regs –**
- **Contain offset within the segments**
- **Stack Pointer:**
  Points to Stack top. Stack is in Stack Segment, used during instructions like PUSH, POP, CALL, RET etc.
- **Base Pointer:**
  BP can hold offset address of any location in the stack segment. It is used to access random locations of the stack.
- **Source Index:**
  It holds offset address in Data Segment during string operations.
- **Destination Index:**
  It holds offset address in Extra Segment during string operations.

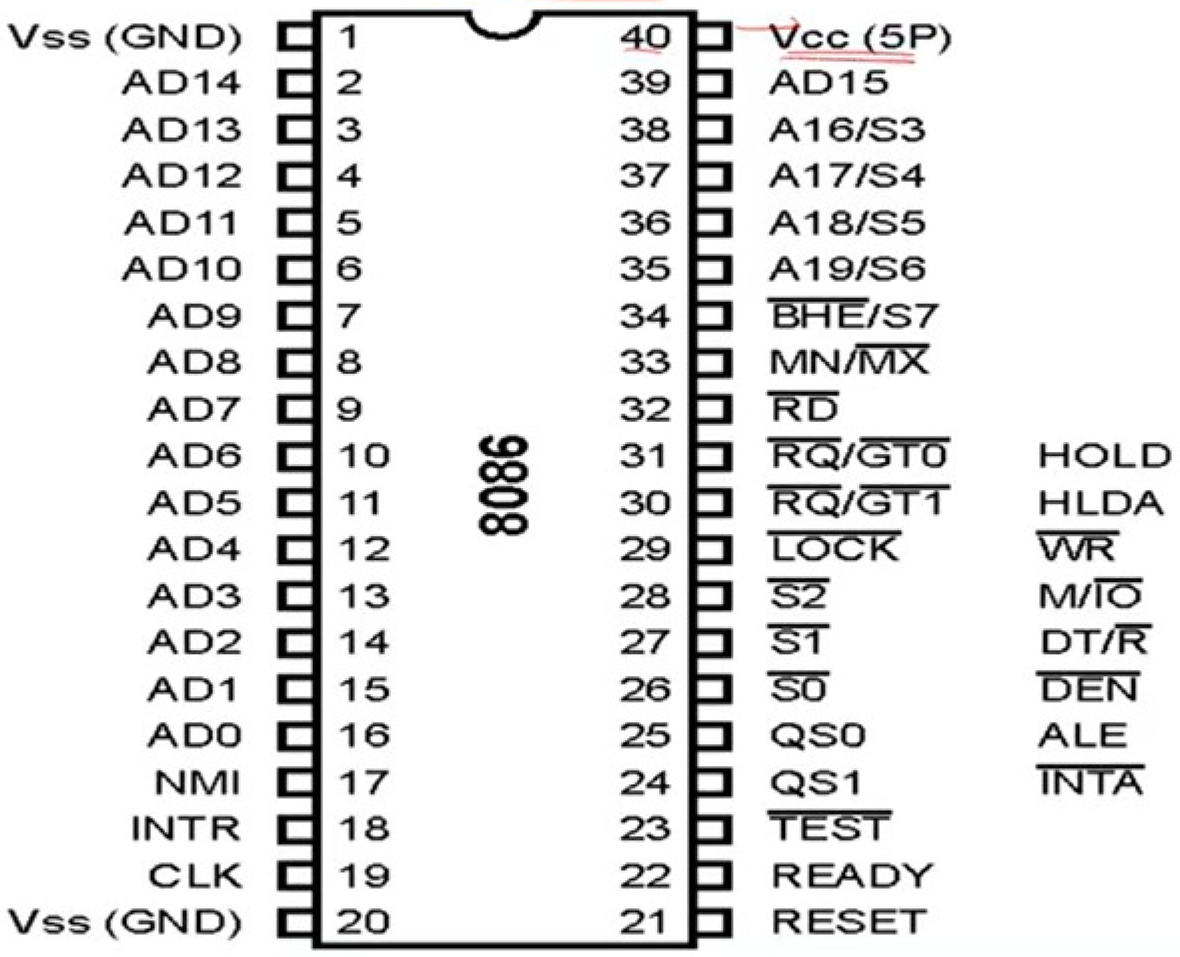
- **Arithmetic Logic Unit (16 bit):**
  Performs **8 and 16 bit** arithmetic and logic operations.
- **Flag Register**
- It is a 16-bit register that behaves like a flip-flop, i.e. it changes its status according to the result stored in the accumulator. It has 9 flags and they are divided into 2 groups − Status Flags and Control Flags.
- **6 Status flags:**
- carry flag(CF)
- parity flag(PF)
- auxiliary carry flag(AF)
- zero flag(Z)
- sign flag(S)
- overflow flag (O)
- Status flags are updated after every arithmetic and logic operation.

- **3 Control flags:**
- trap flag(TF)
- interrupt flag(IF)
- direction flag(DF)



- **Timing and Control Unit**
- Directs all internal operations and also responsible for generation of control signals.

8086 Pin description



Categorized in to three groups:

1. Signals for minimum mode only.
2. Signals for maximum mode only.
3. Signals common to both minimum and maximum mode.

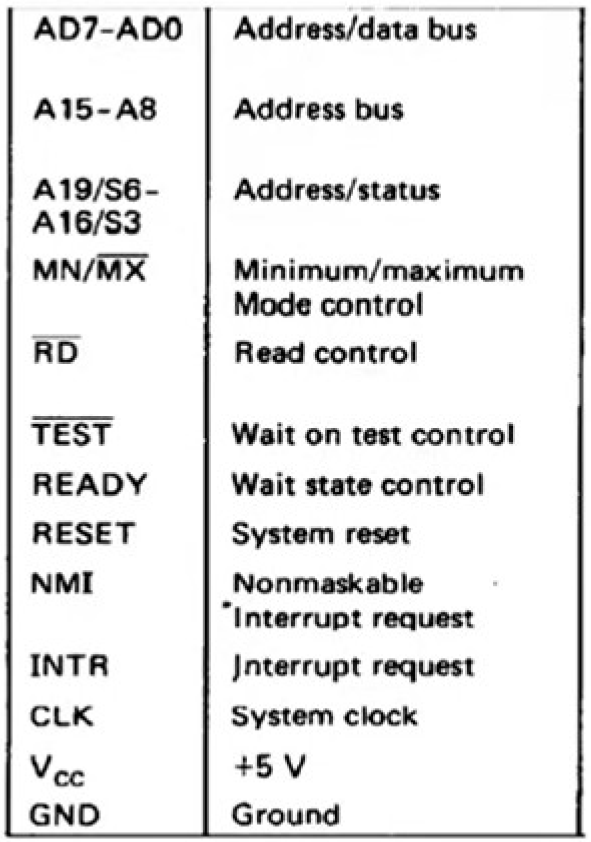
Minimum mode only

Name	Function
HOLD	Hold request
HLDA	Hold acknowledge
$\overline{WR}$	Write control
IO/$\overline{M}$	IO/memory control
DT/$\overline{R}$	Data transmit/receive
$\overline{DEN}$	Data enable
$\overline{SSO}$	Status line
ALE	Address latch enable
$\overline{INTA}$	Interrupt acknowledge

Maximum mode only

Name	Function
$\overline{RQ}/\overline{GT1, 0}$	Request/grant bus access control
$\overline{LOCK}$	Bus priority lock control
$\overline{S2}-\overline{S0}$	Bus cycle status
QS1, QS0	Instruction queue status

Common to both minimum & maximum mode:

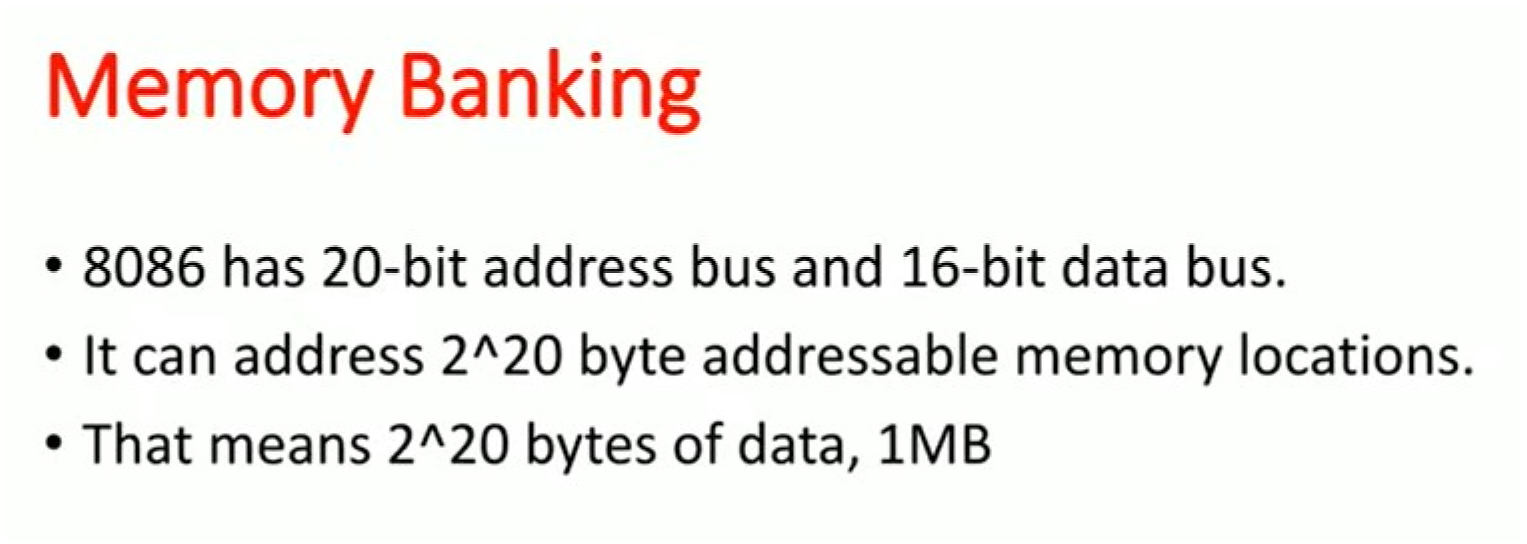


AD7–AD0	Address/data bus
A15–A8	Address bus
A19/S6–A16/S3	Address/status
MN/$\overline{\text{MX}}$	Minimum/maximum Mode control
$\overline{\text{RD}}$	Read control
$\overline{\text{TEST}}$	Wait on test control
READY	Wait state control
RESET	System reset
NMI	Nonmaskable Interrupt request
INTR	Jnterrupt request
CLK	System clock
V_{CC}	+5 V
GND	Ground

- The 8086 works in two modes, minimum and maximum. Accordingly pin configuration changes. They are 32 signals common to both modes. Remaining 8 pins are different for each mode.
- **Address/Data bus**: Carries 16-bit address/data. AD7-AD0 carries lower byte data/address. AD15-AD8 carries higher byte data/address.
- When **ALE (Address Latch Enable)** is high, address/data bus carries address. Otherwise it carries data.
- **A16/S3- A19/S6: Address/status bus**. When ALE is high, it carries address. Otherwise it carries status signal.

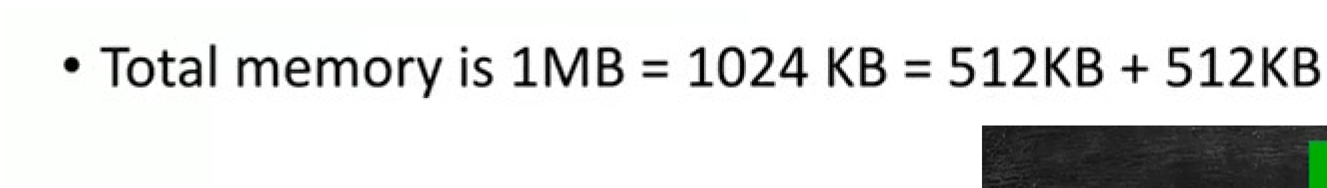
S4	S3	Function
0	0	Extra Segment
0	1	Stack Segment
1	0	Code Segment
1	1	Data Segment

- S5: indicated the presence of interrupt.
- S6:shows the status of bus master for current operation. When S6 is zero, indicates currently 8086 is holding the bus.
- **INTR: Interrupt Request:** used to request a hardware interrupt of INTR is held high when interrupt enable flag is set, the 8086 enters an interrupt acknowledgement cycle after the current instruction has completed its execution
- **NMI: Non-maskable interrupt.** Similar to INTR except that the NMI interrupt does not check for interrupt enable flag is at logic 1, i.e, NMI is not maskable internally by software.
- **Clock :** Clock signal is provided through this pin.It provides timing to the processor for operations. Its frequency is different for different versions, i.e. 5MHz, 8MHz and 10MHz.


Memory Banking

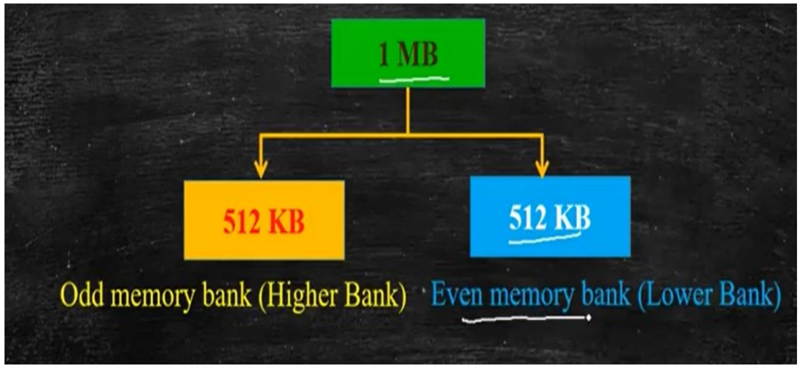
- 8086 has 20-bit address bus and 16-bit data bus.
- It can address 2^20 byte addressable memory locations.
- That means 2^20 bytes of data, 1MB

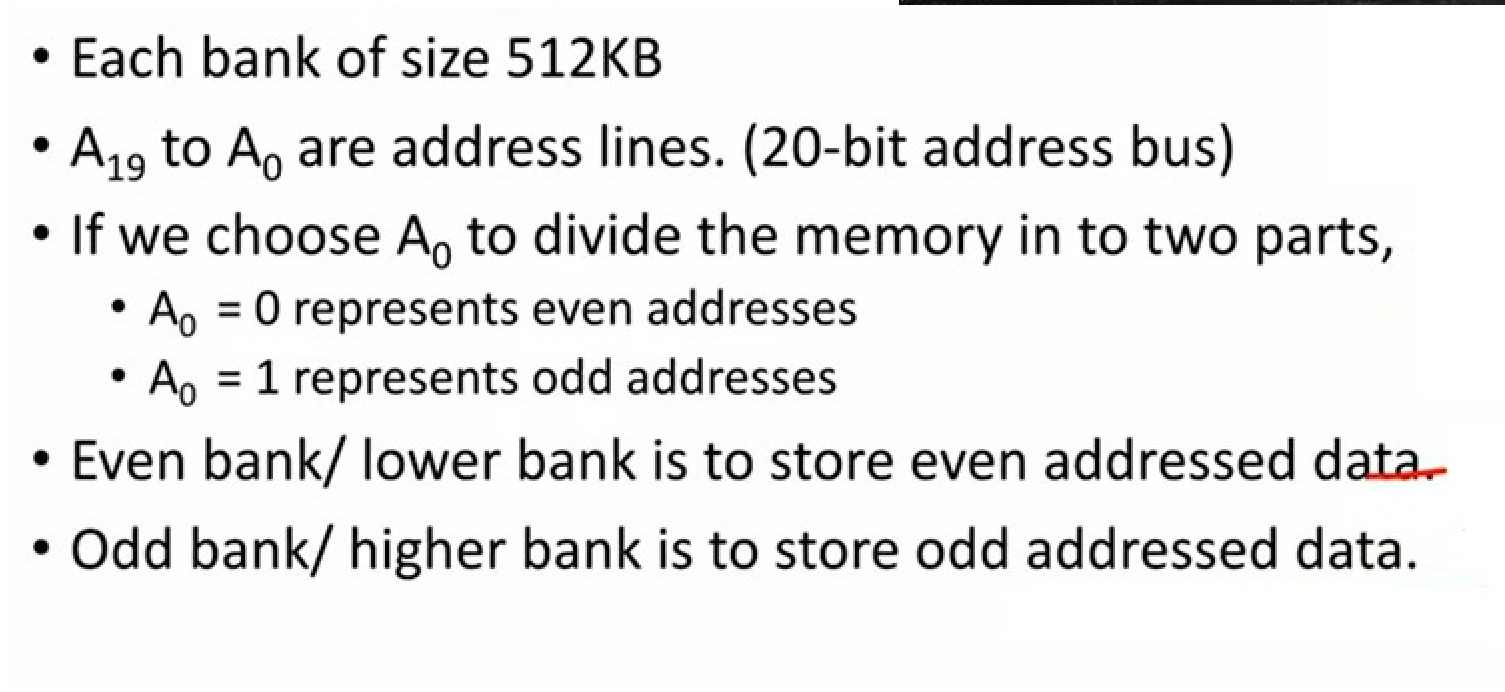
- 2^20 =M
- 2^10=K
- B=BYTES
- b=bits



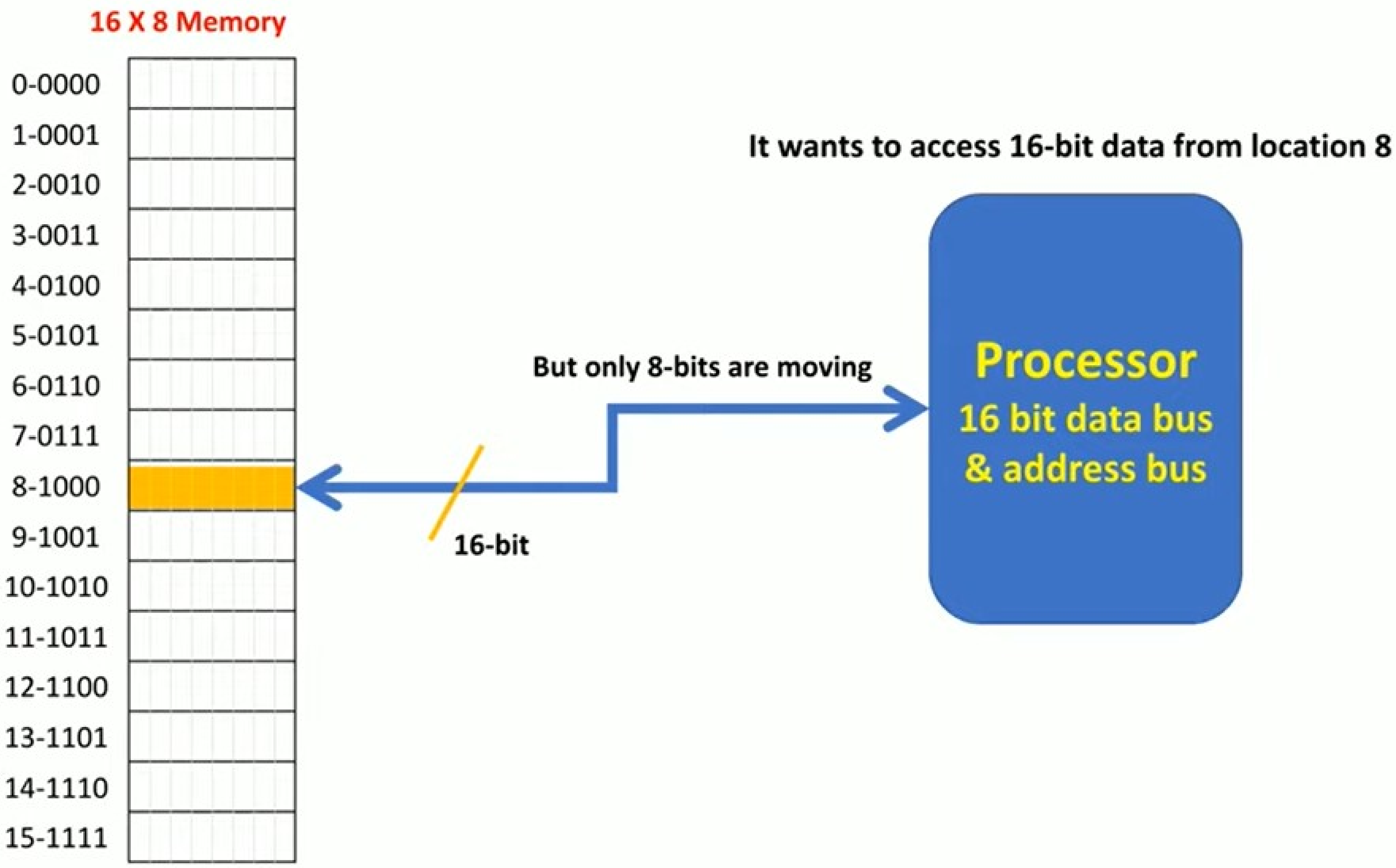
- Total memory is 1MB = 1024 KB = 512KB + 512KB

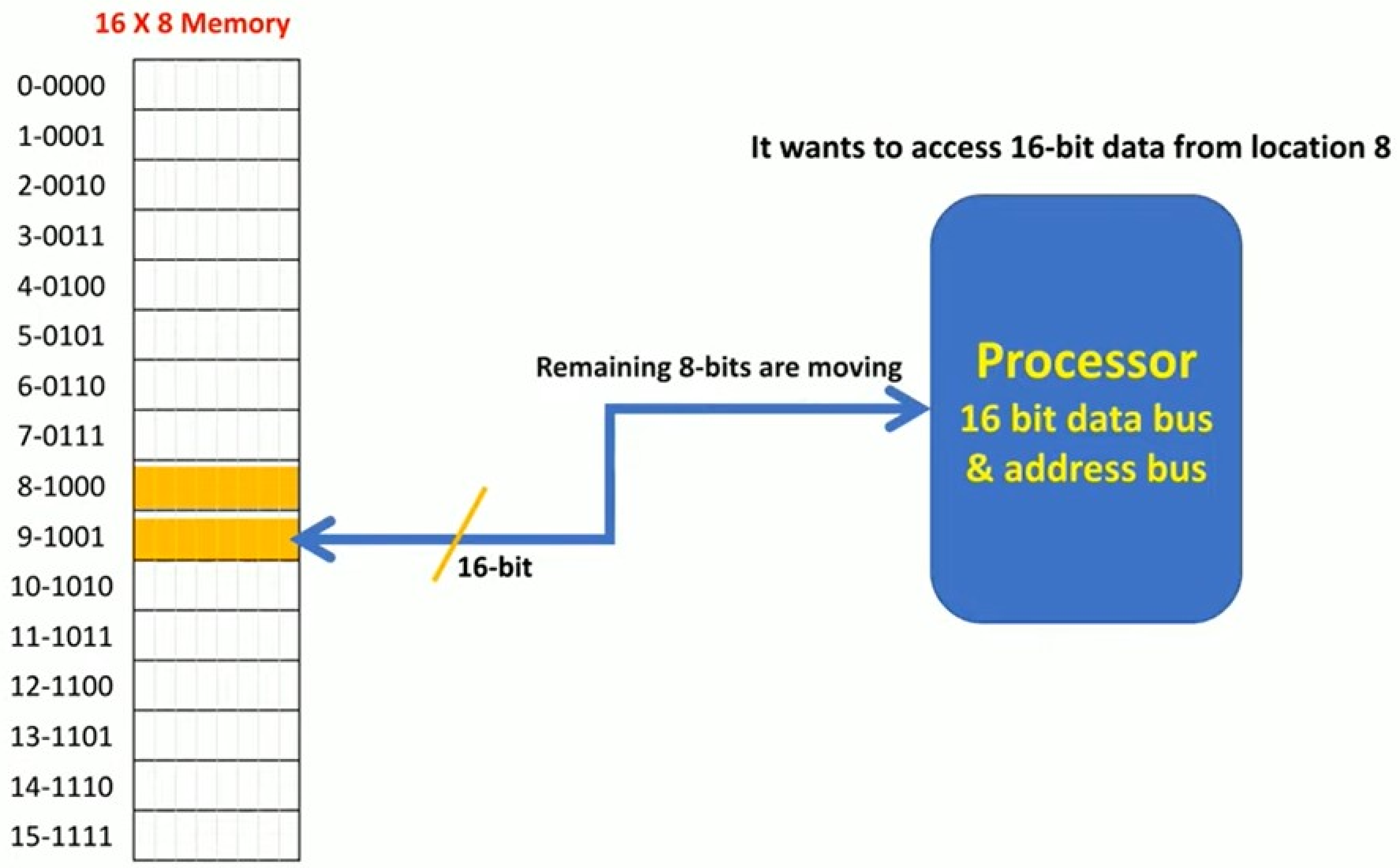
- 1 MB = $2^{10} \times 2^{10}$ Byte
  - =1024XK Bytes

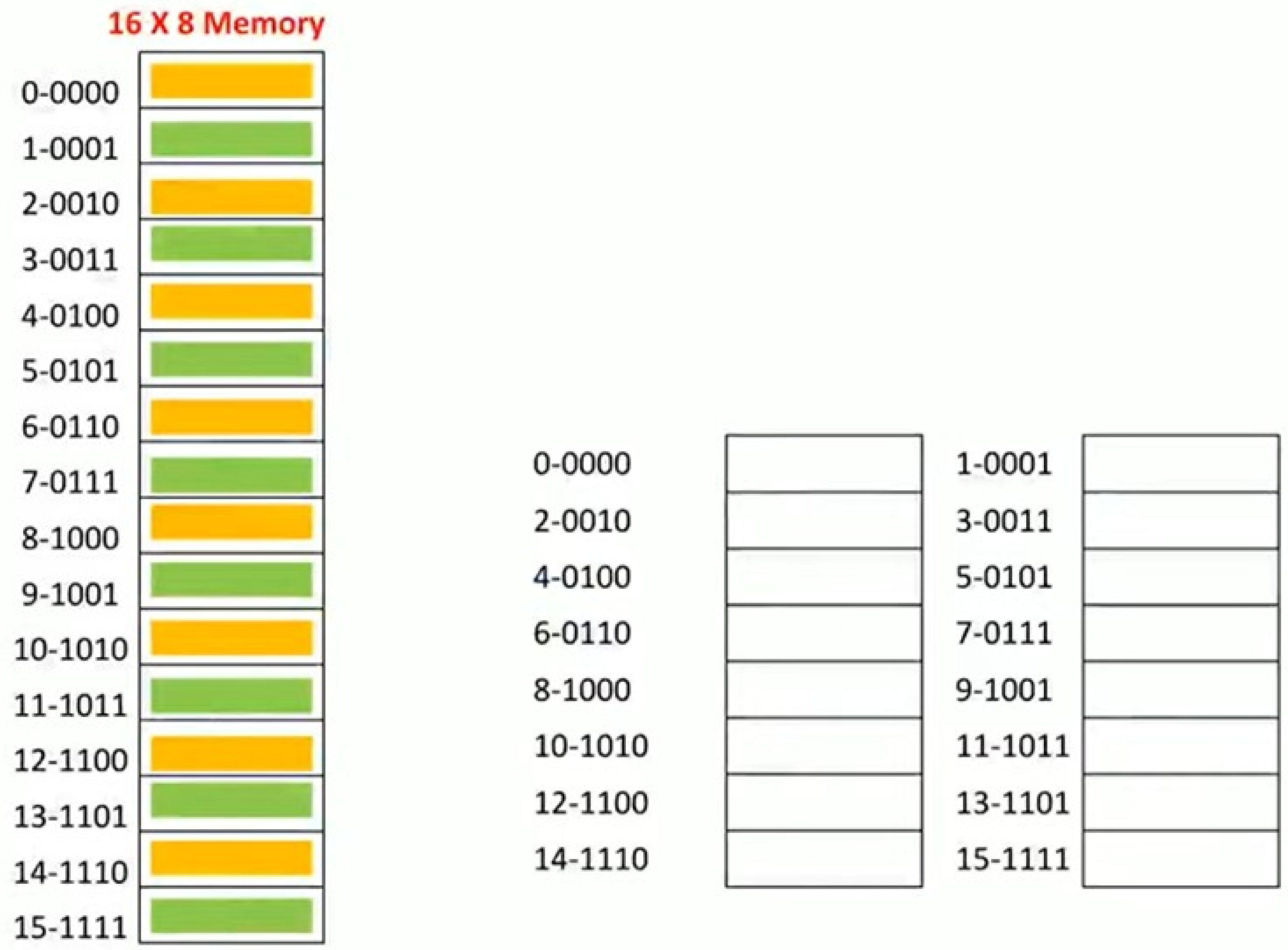


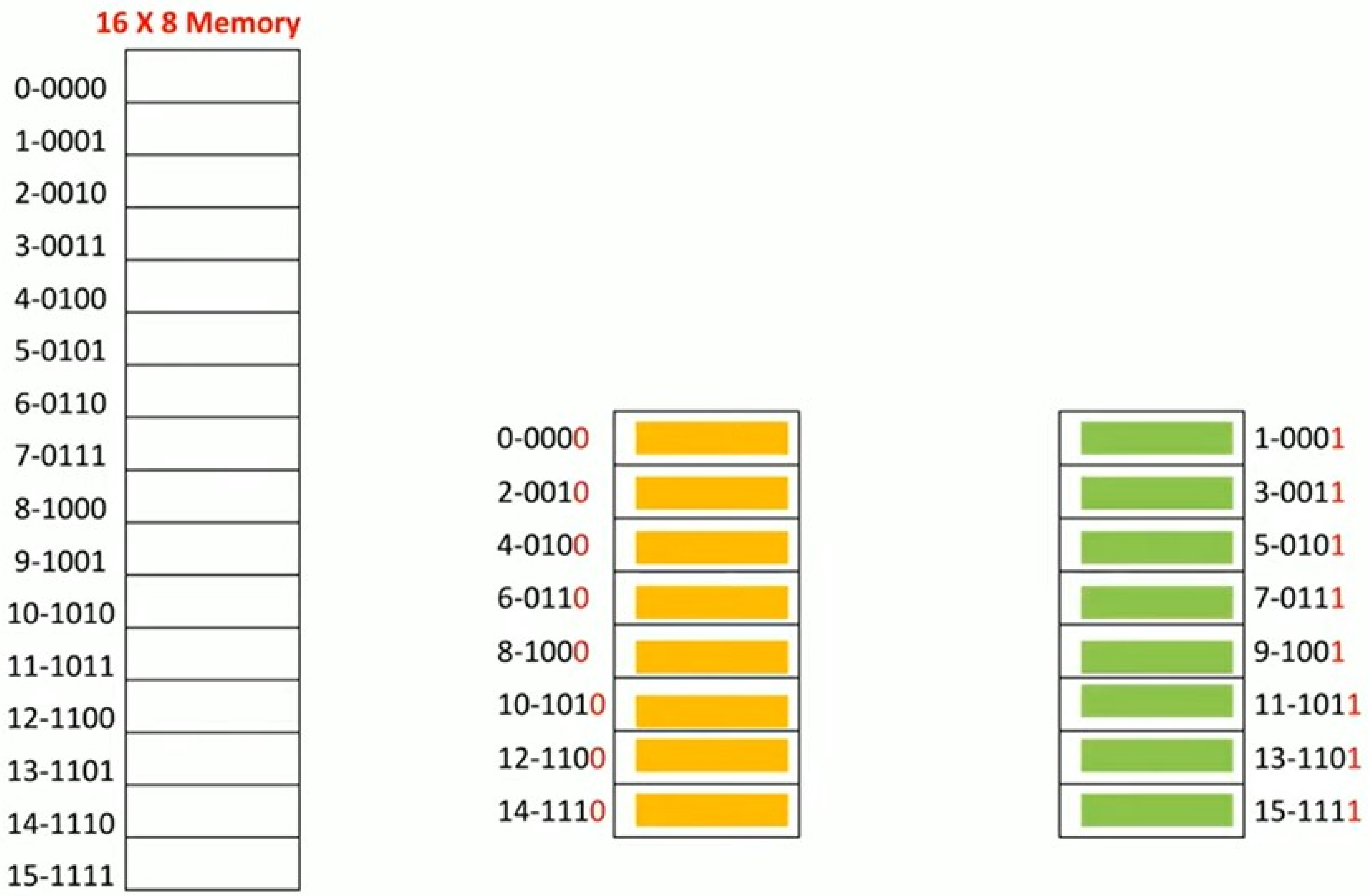


- Each bank of size 512KB
- A_{19} to A_0 are address lines. (20-bit address bus)
- If we choose A_0 to divide the memory in to two parts,
  - A_0 = 0 represents even addresses
  - A_0 = 1 represents odd addresses
- Even bank/ lower bank is to store even addressed data.
- Odd bank/ higher bank is to store odd addressed data.


16 X 8 Memory
0-0000
1-0001
2-0010
3-0011
4-0100
5-0101
6-0110
7-0111
8-1000
9-1001
10-1010
11-1011
12-1100
13-1101
14-1110
15-1111
It wants to access 16-bit data from location 8
But only 8-bits are moving
16-bit
Processor
16 bit data bus
& address bus

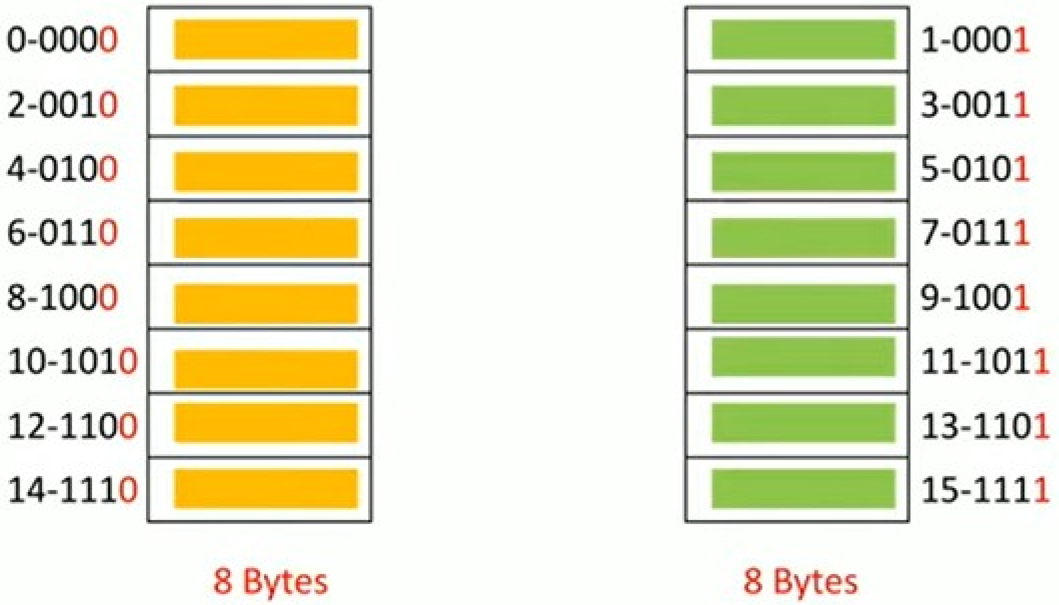

16 X 8 Memory
0-0000
1-0001
2-0010
3-0011
4-0100
5-0101
6-0110
7-0111
8-1000
9-1001
10-1010
11-1011
12-1100
13-1101
14-1110
15-1111
It wants to access 16-bit data from location 8
Remaining 8-bits are moving
16-bit
Processor
16 bit data bus
& address bus

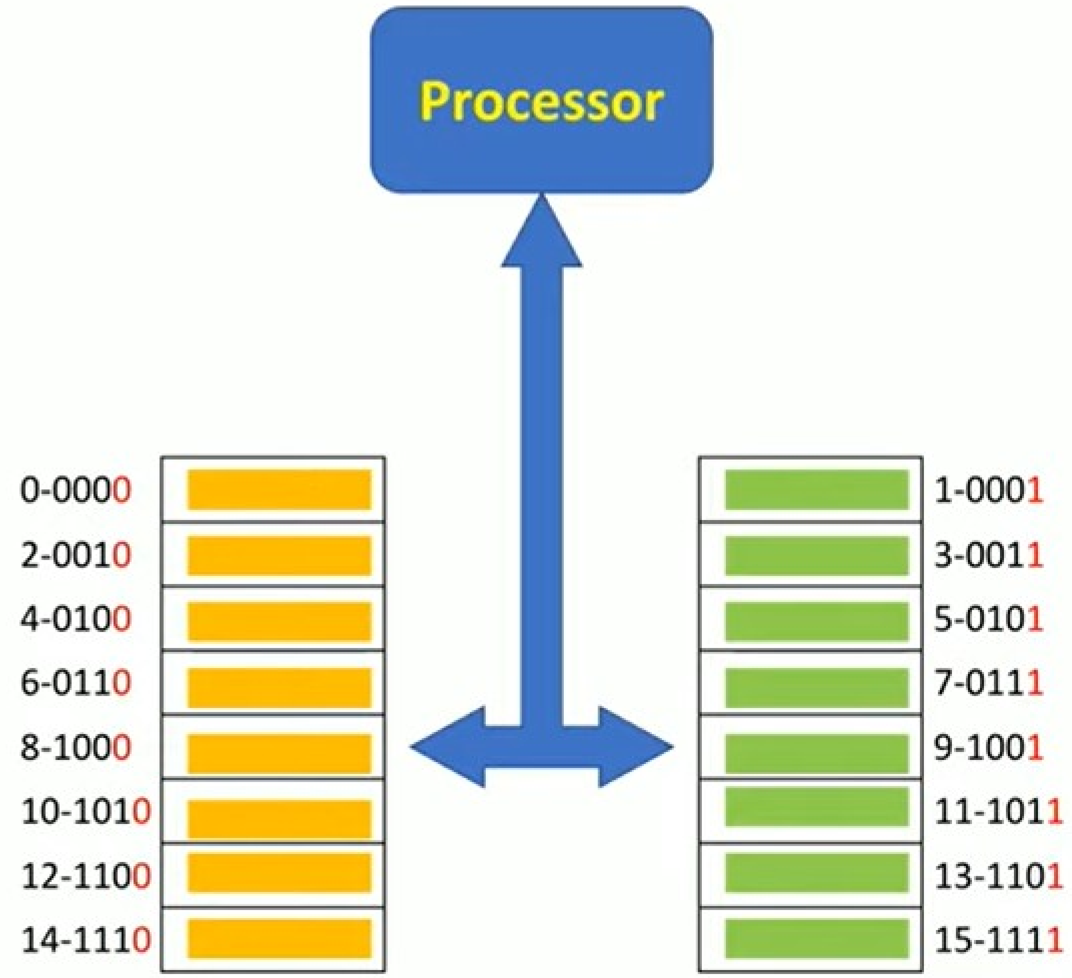

16 X 8 Memory
0-0000
1-0001
2-0010
3-0011
4-0100
5-0101
6-0110
7-0111
8-1000
9-1001
10-1010
11-1011
12-1100
13-1101
14-1110
15-1111
0-0000
2-0010
4-0100
6-0110
8-1000
10-1010
12-1100
14-1110
1-0001
3-0011
5-0101
7-0111
9-1001
11-1011
13-1101
15-1111


16 X 8 Memory
0-0000
1-0001
2-0010
3-0011
4-0100
5-0101
6-0110
7-0111
8-1000
9-1001
10-1010
11-1011
12-1100
13-1101
14-1110
15-1111
0-0000
2-0010
4-0100
6-0110
8-1000
10-1010
12-1100
14-1110
1-0001
3-0011
5-0101
7-0111
9-1001
11-1011
13-1101
15-1111

A3A2A1 A0

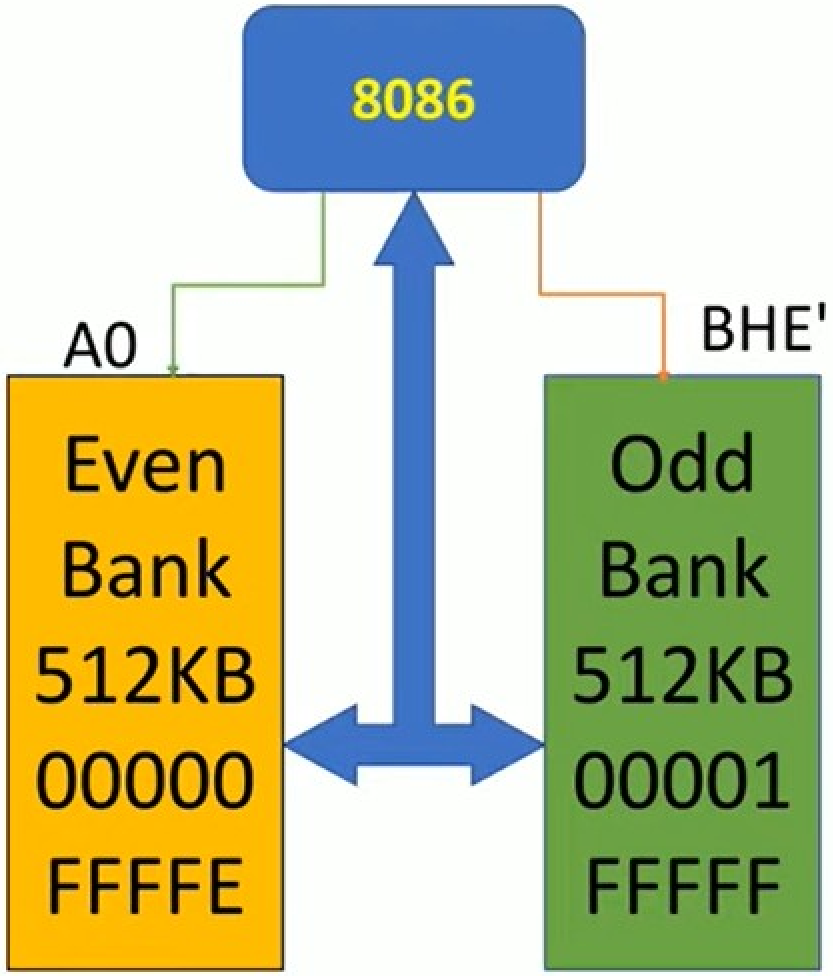
A3A2A1 are used to address memory banks

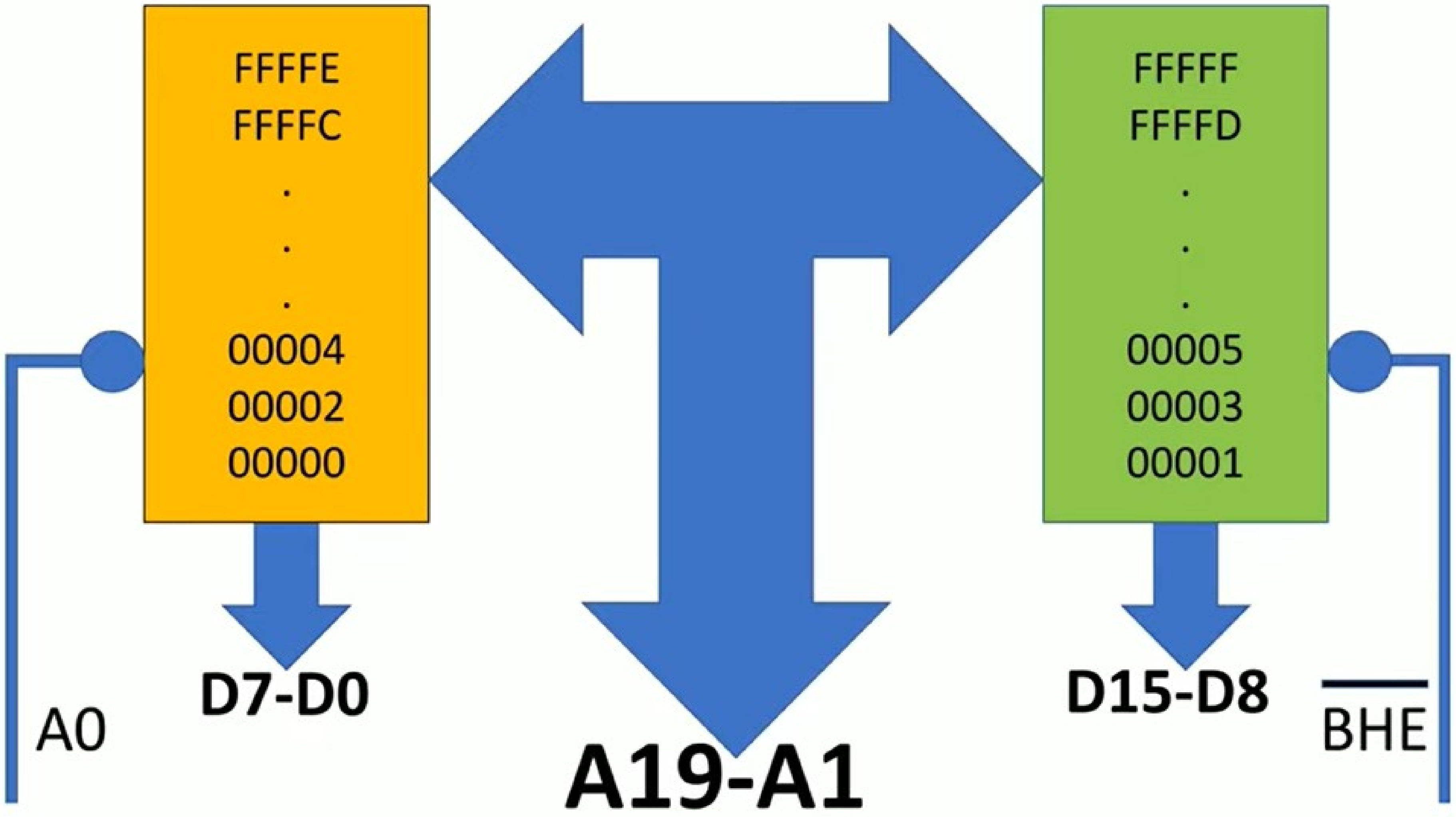


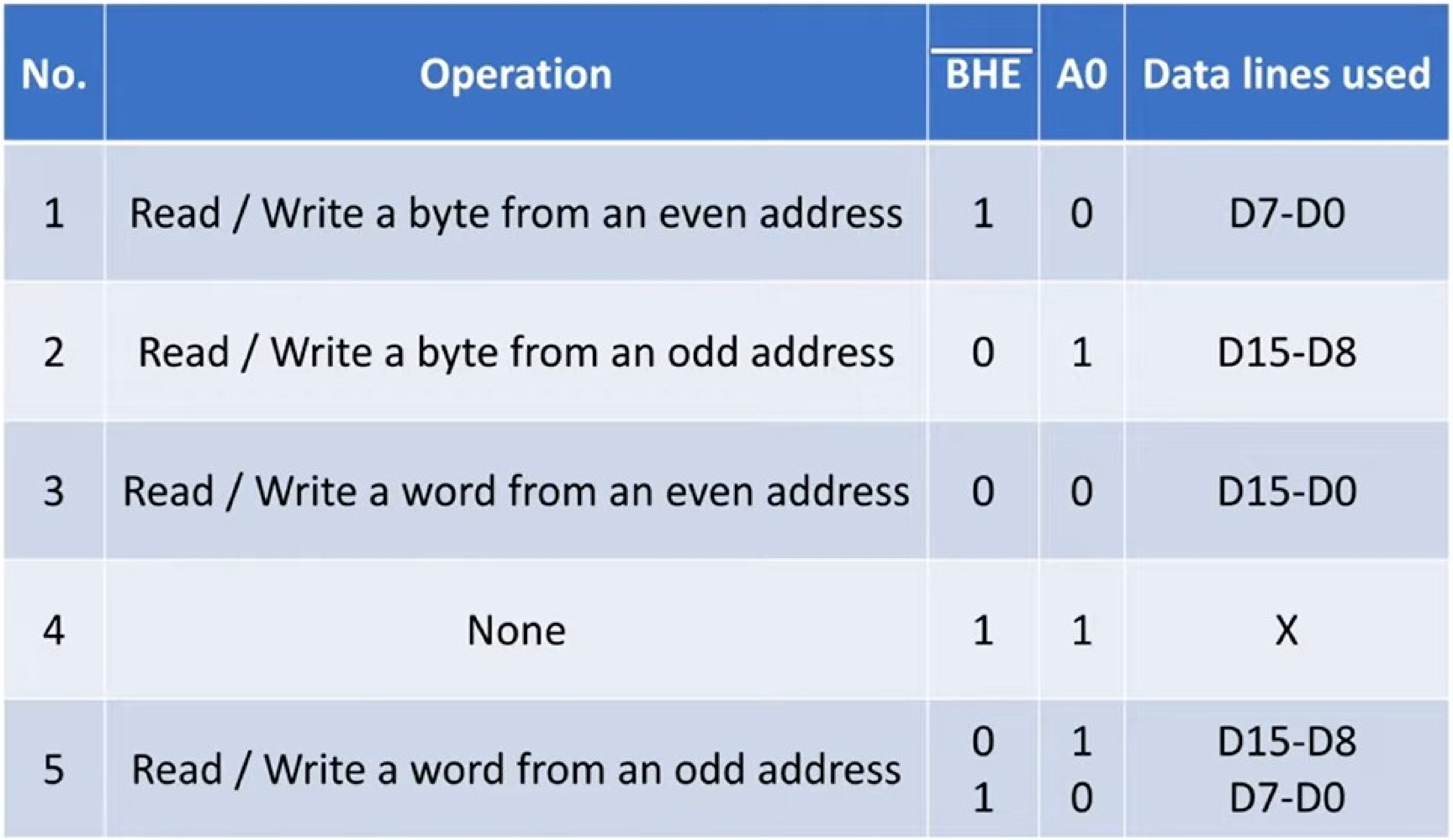


Processor
0-0000
2-0010
4-0100
6-0110
8-1000
10-1010
12-1100
14-1110
1-0001
3-0011
5-0101
7-0111
9-1001
11-1011
13-1101
15-1111

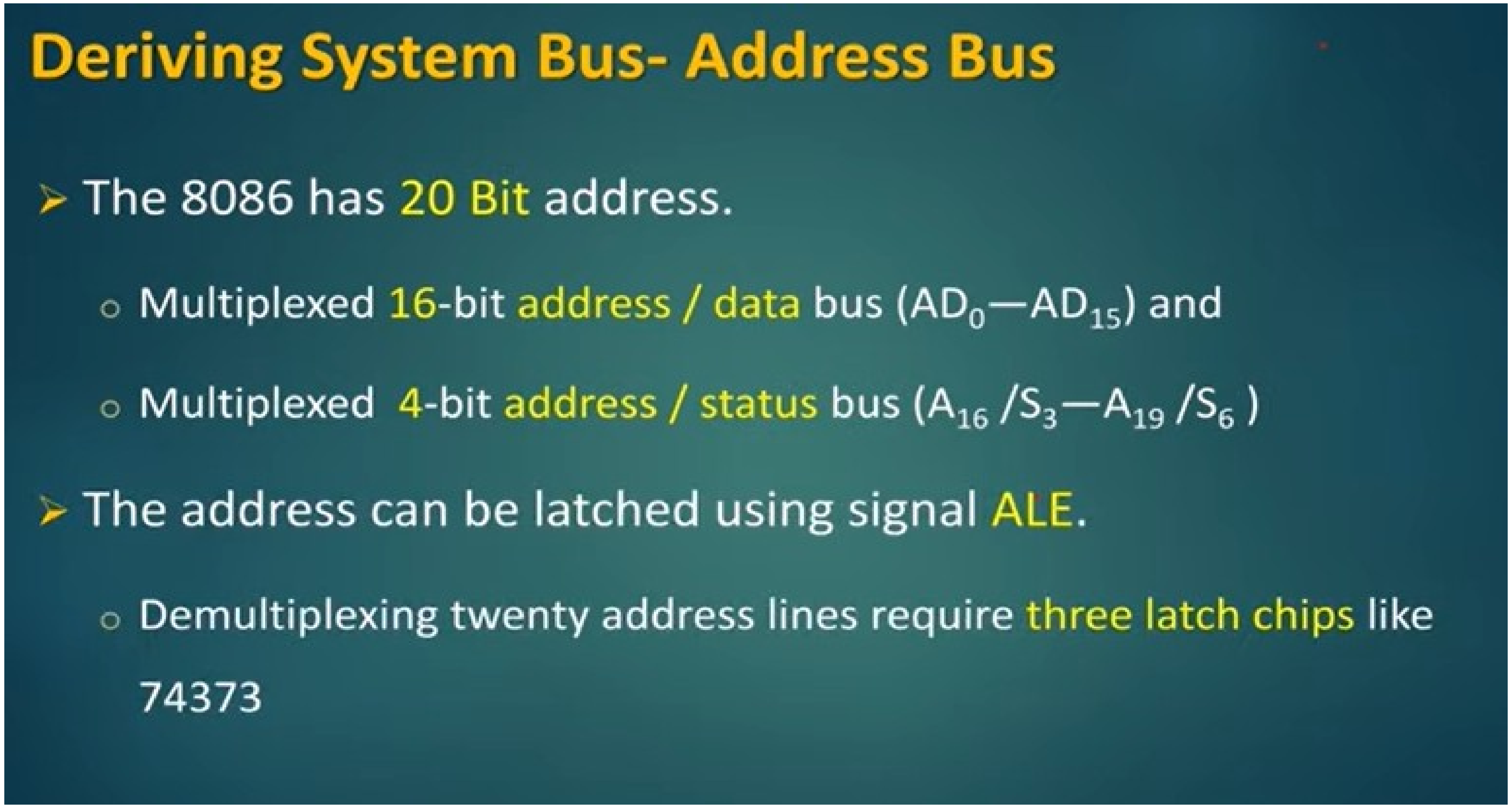
- A19-A1 are used to address memory banks.
- A0 is used to select Even / Lower Bank.
- BHE is used to select Odd / Higher Bank.
- Common chip select signal is not used.
- Because the processor always doesn't wants 16-bit data.




FFFFE
FFFFC
.
.
.
00004
00002
00000
FFFFF
FFFFD
.
.
.
00005
00003
00001
A0
D7-D0
A19-A1
D15-D8
$\overline{BHE}$



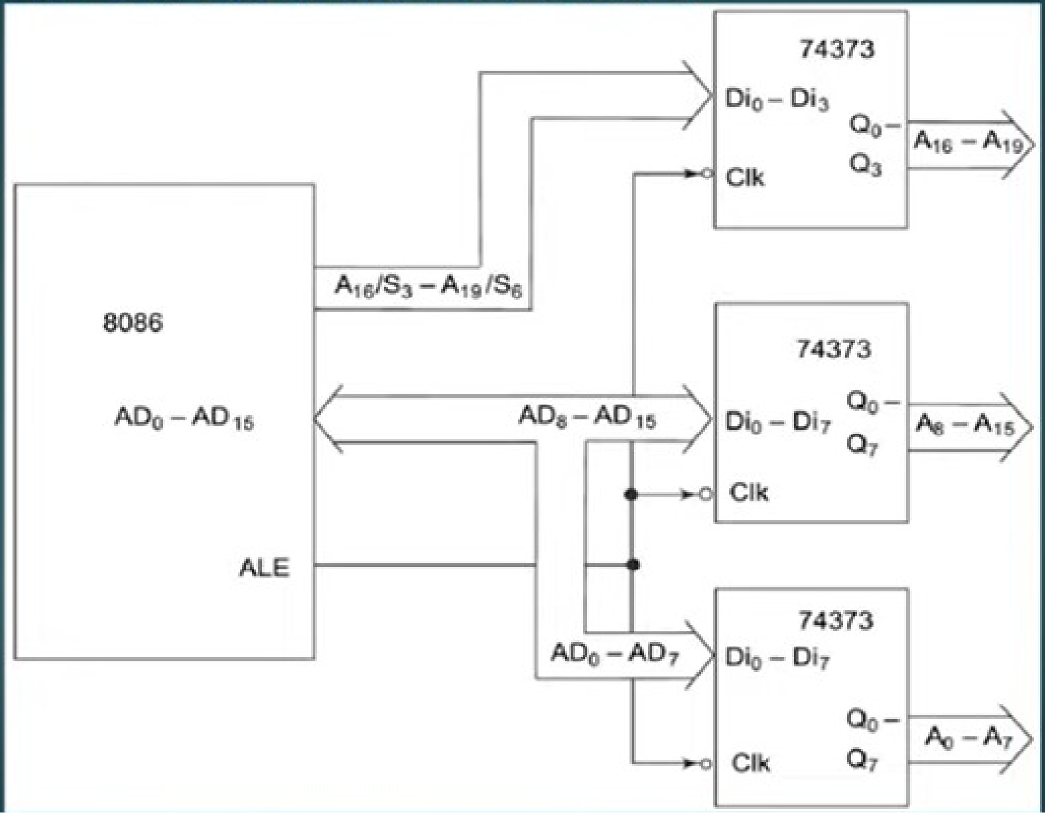
No.	Operation	$\overline{BHE}$	A0	Data lines used
1	Read / Write a byte from an even address	1	0	D7-D0
2	Read / Write a byte from an odd address	0	1	D15-D8
3	Read / Write a word from an even address	0	0	D15-D0
4	None	1	1	X
5	Read / Write a word from an odd address	0 1	1 0	D15-D8 D7-D0



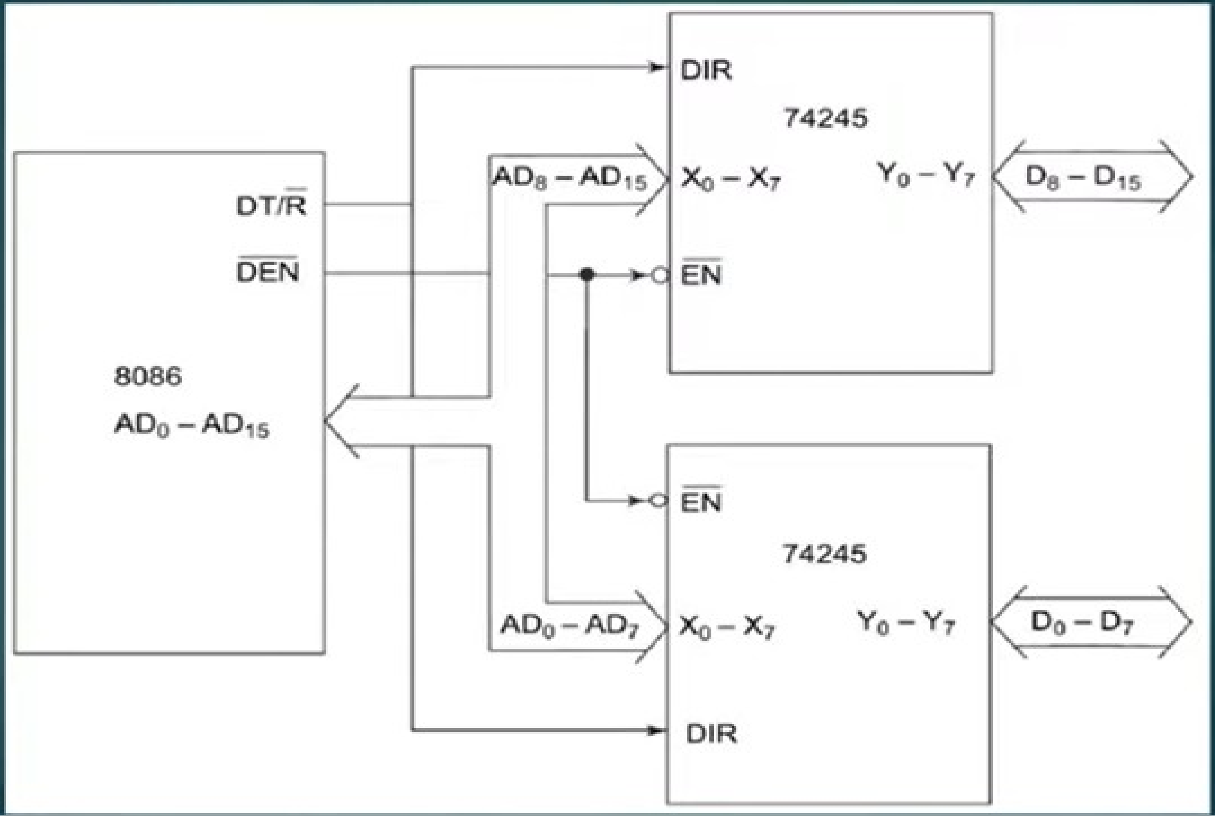
Deriving System Bus- Address Bus

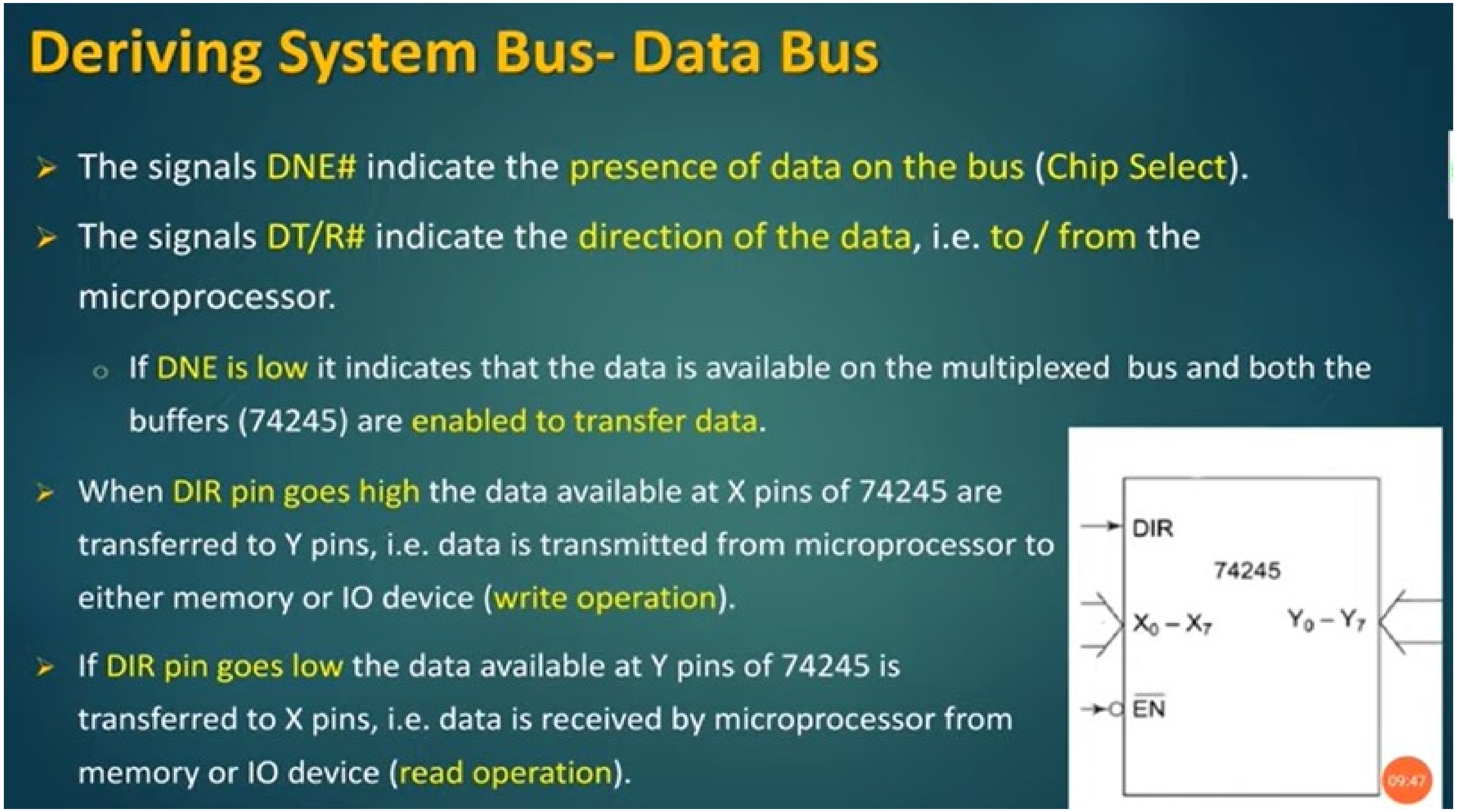
- The 8086 has 20 Bit address.
  - Multiplexed 16-bit address / data bus (AD_0—AD_{15}) and
  - Multiplexed 4-bit address / status bus (A_{16} /S_3—A_{19} /S_6)
- The address can be latched using signal ALE.
  - Demultiplexing twenty address lines require three latch chips like 74373

Latching 20 Bit address of 8086



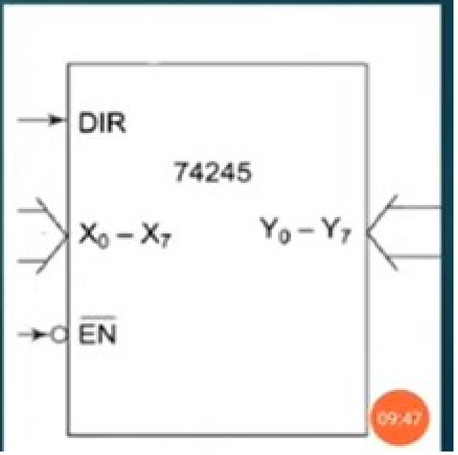
Buffering Data Bus of 8086

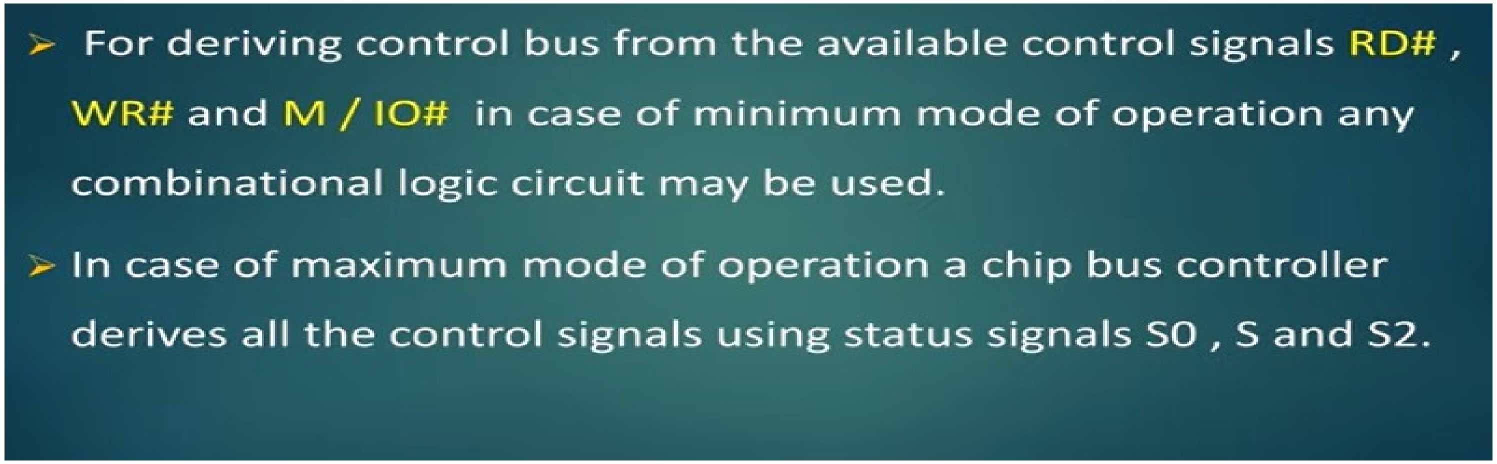




Deriving System Bus- Data Bus

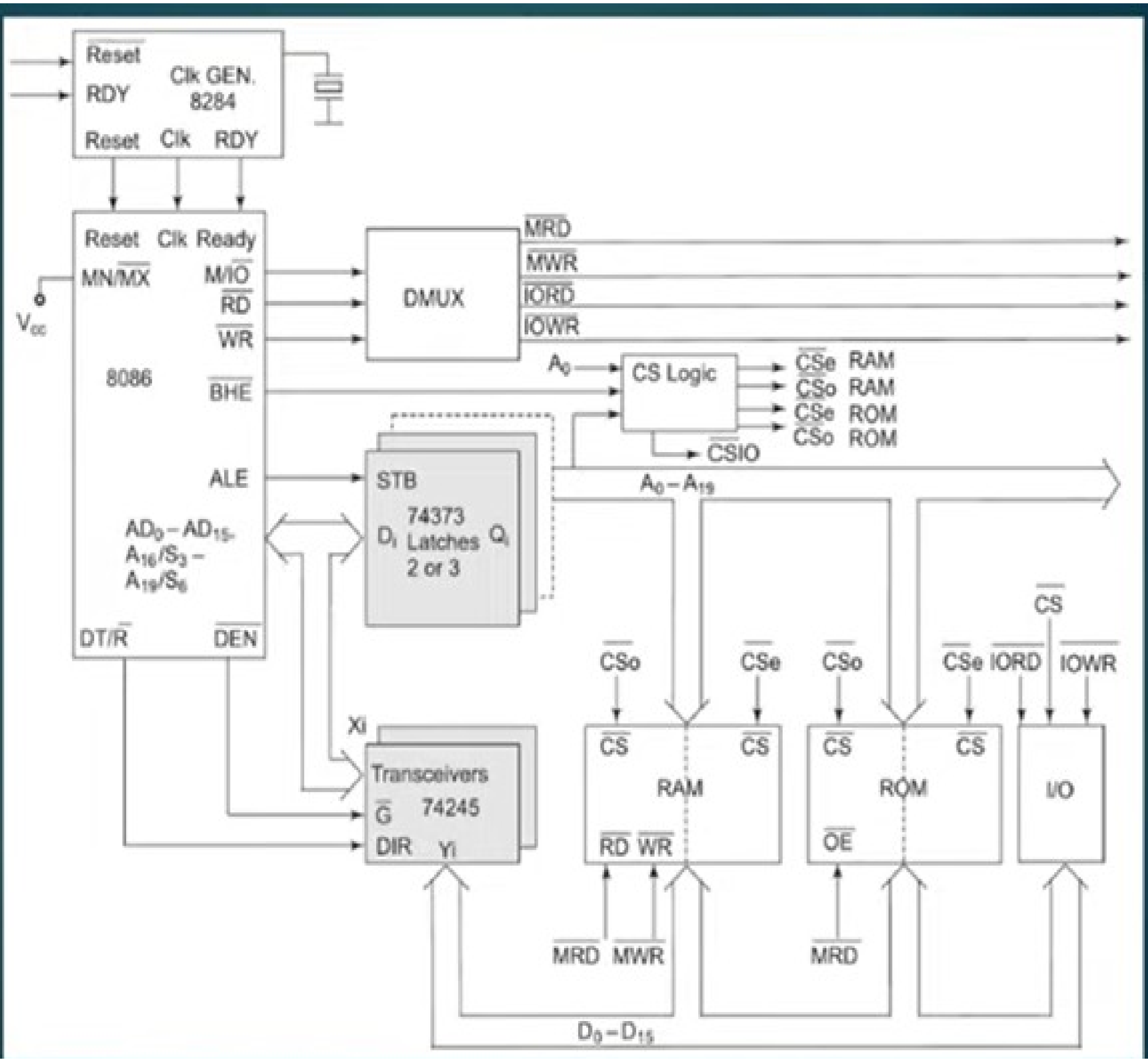
- The signals DNE# indicate the presence of data on the bus (Chip Select).
- The signals DT/R# indicate the direction of the data, i.e. to / from the microprocessor.
  - If DNE is low it indicates that the data is available on the multiplexed bus and both the buffers (74245) are enabled to transfer data.
- When DIR pin goes high the data available at X pins of 74245 are transferred to Y pins, i.e. data is transmitted from microprocessor to either memory or IO device (write operation).
- If DIR pin goes low the data available at Y pins of 74245 is transferred to X pins, i.e. data is received by microprocessor from memory or IO device (read operation).



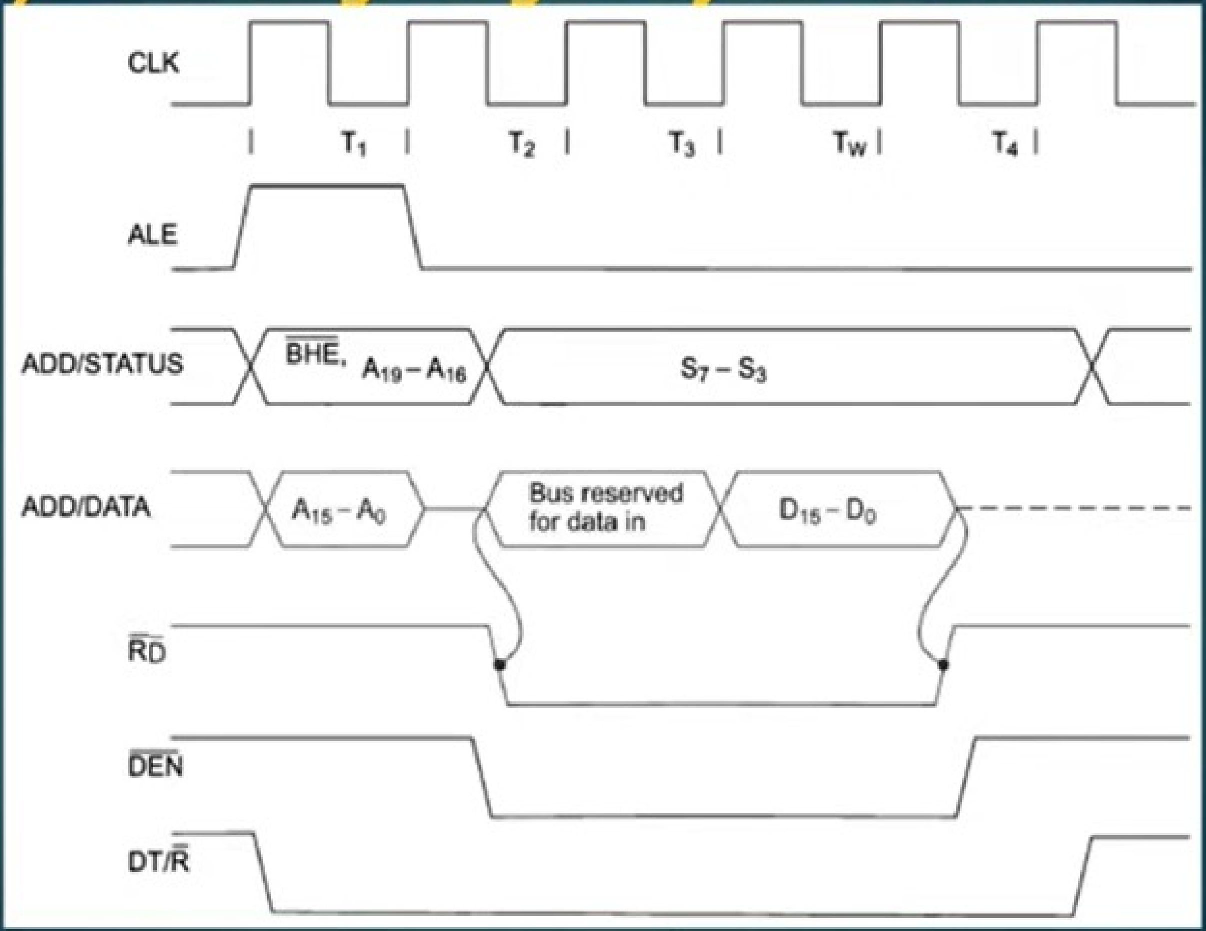


- For deriving control bus from the available control signals RD# , WR# and M / IO# in case of minimum mode of operation any combinational logic circuit may be used.
- In case of maximum mode of operation a chip bus controller derives all the control signals using status signals S0 , S and S2.

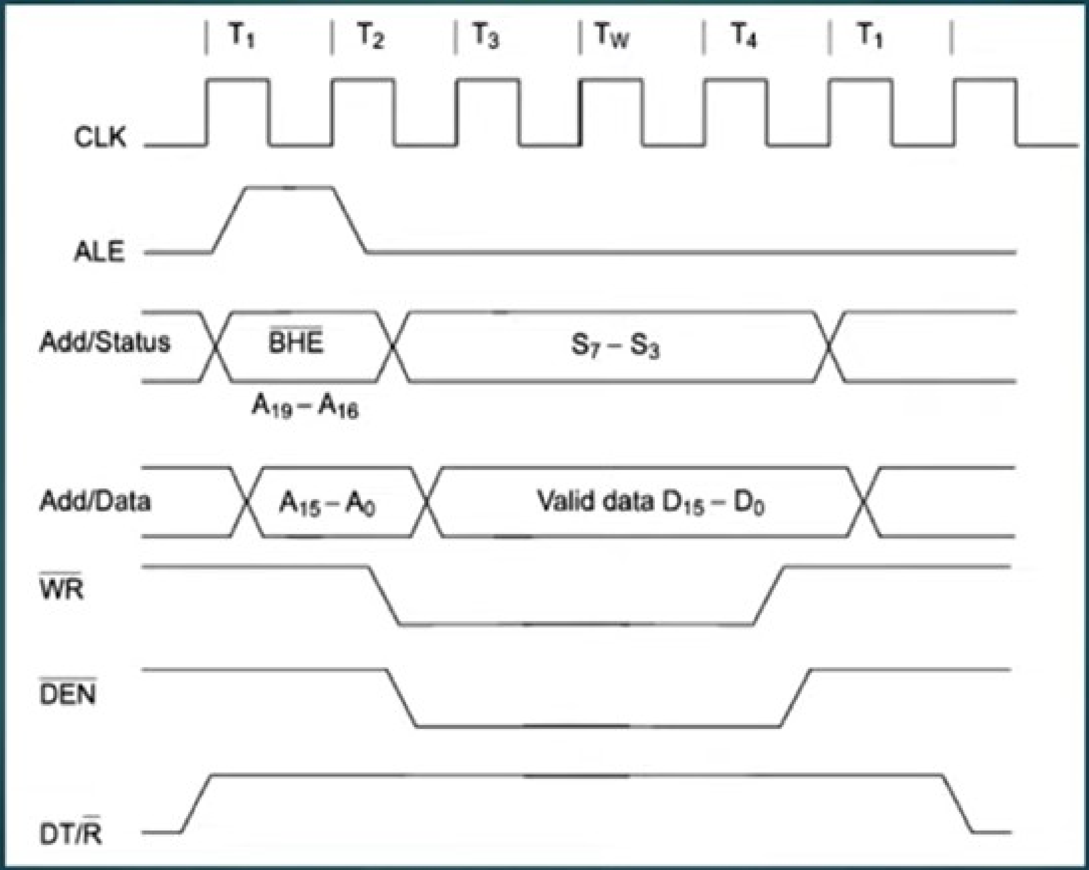
Minimum Mode of 8086



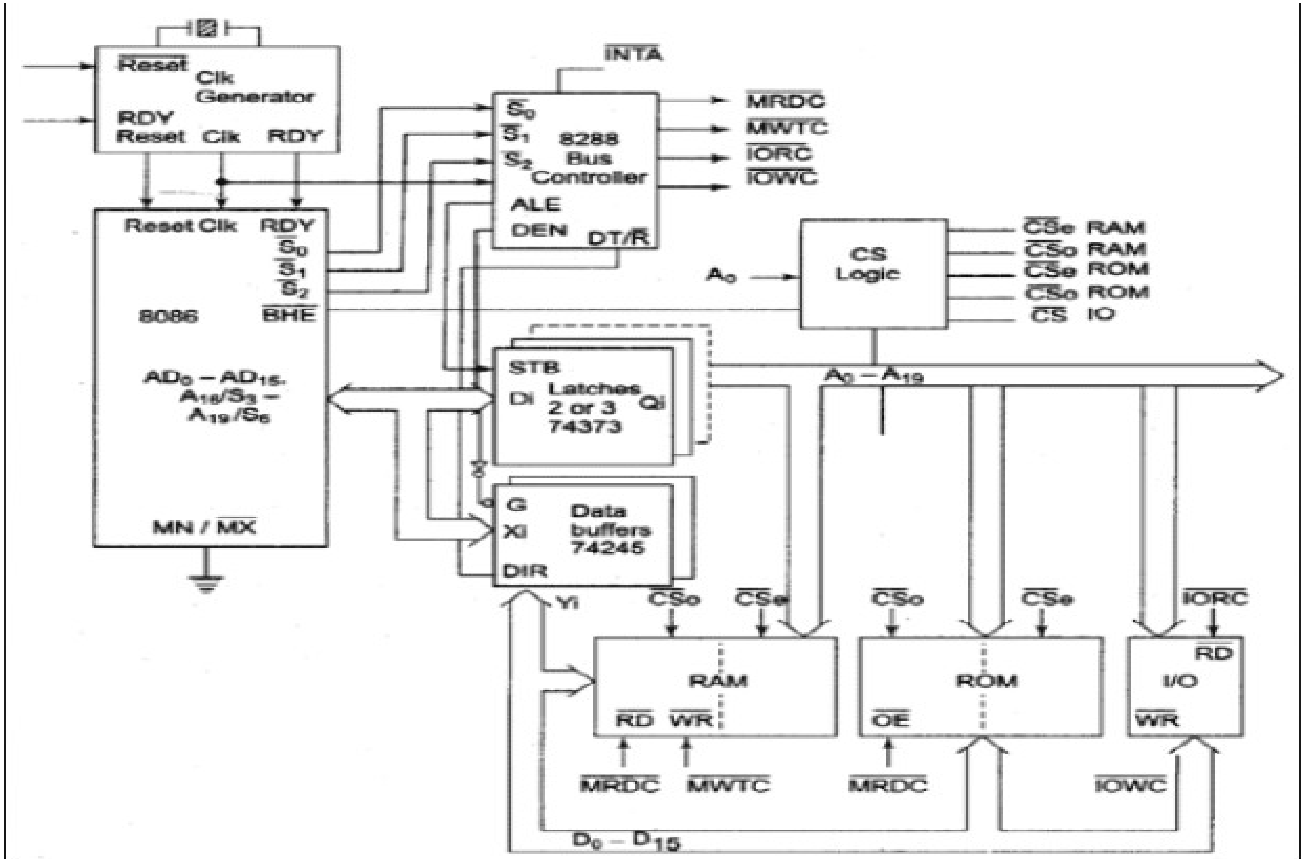
Read Cycle Timing Diagram for Minimum Mode



Write Cycle Timing Diagram for Minimum Mode



MAXIMUM MODE OF 8086



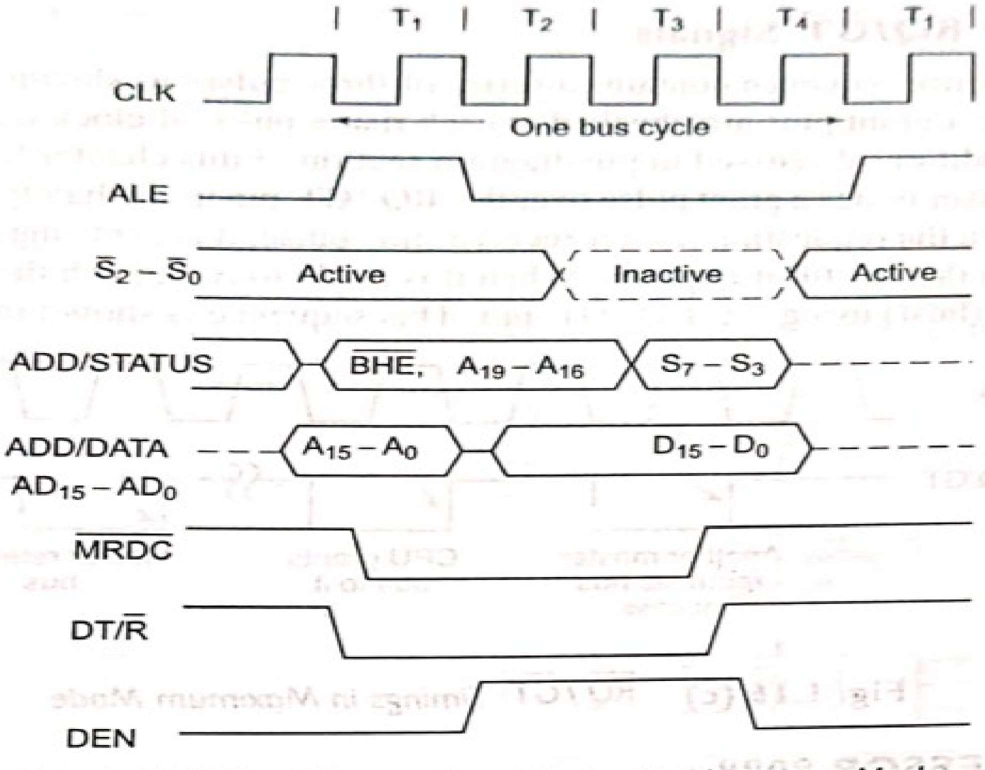


Fig. 1.16 (a) *Memory Read Timing in Maximum Mode*

TIMINGS:

. Timing and Timing diagram plays a vital role in microprocessors.

The timing diagram is the diagram which provides information about the various conditions of signals such as high/low, when a machine cycle is being executed.

Without the knowledge of timing diagram it is not possible to match the peripheral devices to the microprocessors. These peripheral devices includes memories, ports etc. Such devices can only be matched with microprocessors with the help of timing diagram.

Before dealing with timing diagram, we have to make ourselves familiar with certain terms.

VivTronics

Machine cycle: A basic microprocessor operation such as reading a byte from memory or writing a byte to a port is called a machine cycle (Bus cycle A machine cycle consists of at least 4 clock cycles/ clock states (T-states) for accessing the data.

Instruction cycle: The time a microprocessor requires to fetch and execute an entire instruction is referred to as an instruction cycle. An instruction cycle consists of one or more machine cycles.

T-state:T-state is nothing but one subdivision of the operation performed in one clock period. These subdivisions are internal state of the microprocessor synchronized with system clock.

VivTronics