Module No 3

Lecture No 16

Memory Interface of 8088 and 8086 processors

Objective:

- To introduce the read and write bus cycles of the 8088 and 8086 processors.

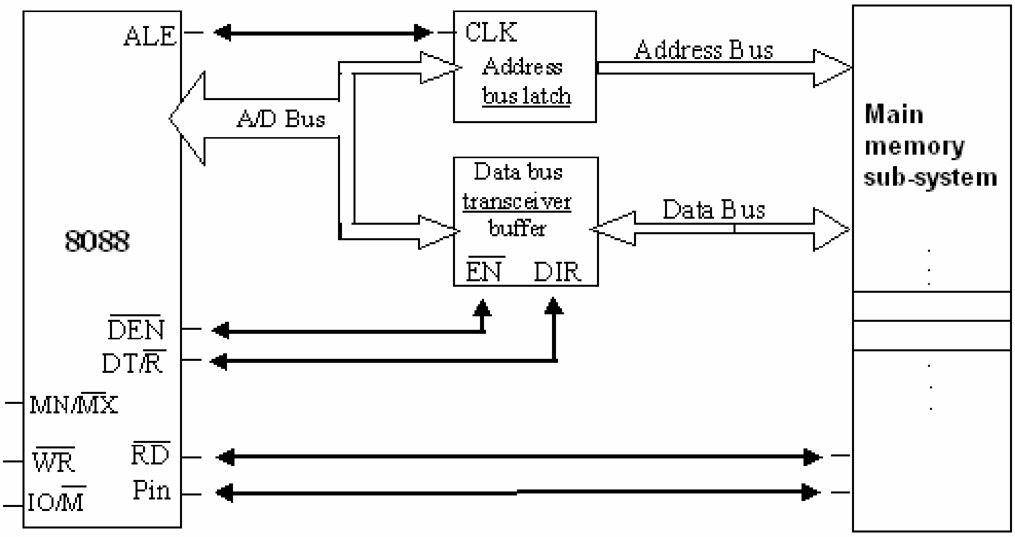
- To discuss the timing diagram associated with read/write bus cycles

- To investigate the digital circuits used in memory interface.

Slide 1: **Memory interface of a minimum mode 8088 system:**

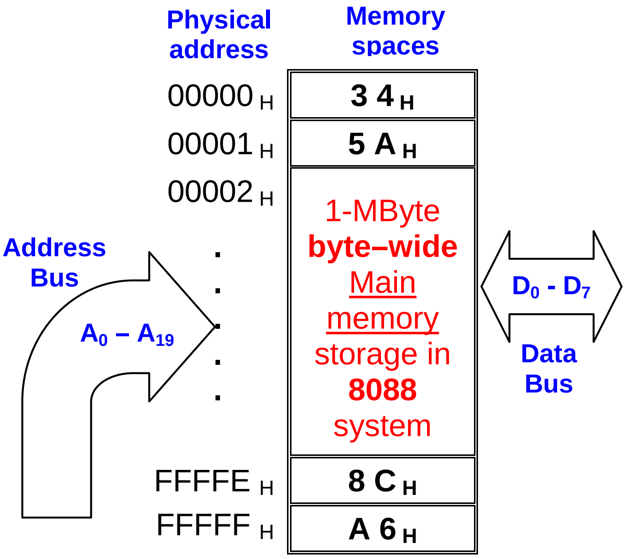
- The figure below shows the memory interface circuit of an 8088 based system operating in minimum mode.

-



- The 8088's memory system uses a single '1-Mega X 8 bit' memory bank with Physical Address from 00000_H to $FFFFF_H$.

- The 20-bit (≈ 5 Hex digit) physical address supplied by the address-bus (A_0-A_{19}) is used to select the memory location required to be accessed via data-bus (D_0-D_7)



Physical address	Memory spaces
00000_H	34_H
00001_H	$5A_H$
00002_H	1-MByte **byte–wide** Main memory storage in **8088** system
.	
.	
.	
.	
.	
$FFFFE_H$	$8C_H$
$FFFFF_H$	$A6_H$

Address Bus

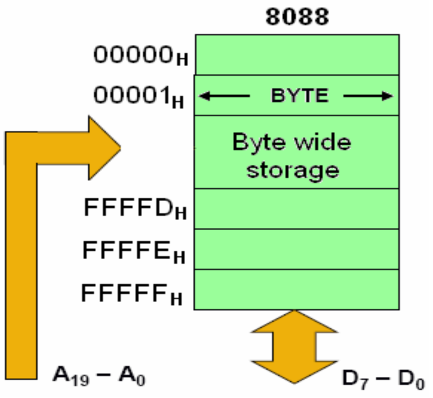
$A_0 – A_{19}$

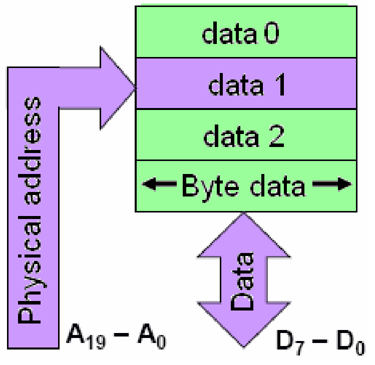
D_0 - D_7

Data Bus

Slide 2: **Accessing a Byte-data in a 8088-based Memory System:**

- In 8088 system, one bus cycle is required to complete the process of applying valid physical address and accessing (reading or writing) a byte data from that memory location.



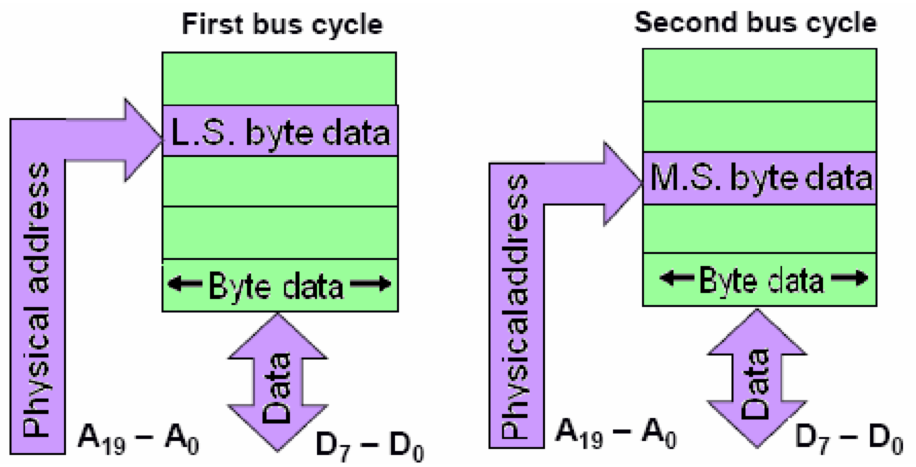


In this figure, appropriate physical address is applied to access the memory location storing 'data 1"

Slide 3: **<u>Accessing a Word-data in a 8088-based Memory System:</u>**

- Two bus cycles are required to access a word data in an 8088 memory system. The 1st bus cycle access the least significant byte (LSB) of the stored word, as show in figure below.

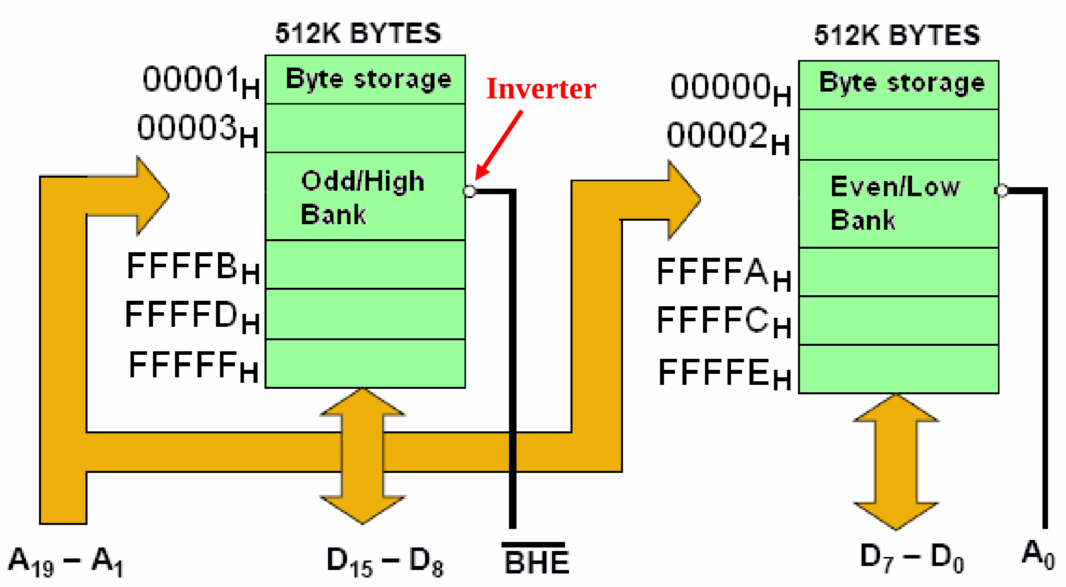
- During the 2nd bus cycle, the physical address in automatically incremented to access the most significant byte of the word.



Slide 4:

- The 8086 systems have two ‘0.5 Mega X 8 bit’ memory banks

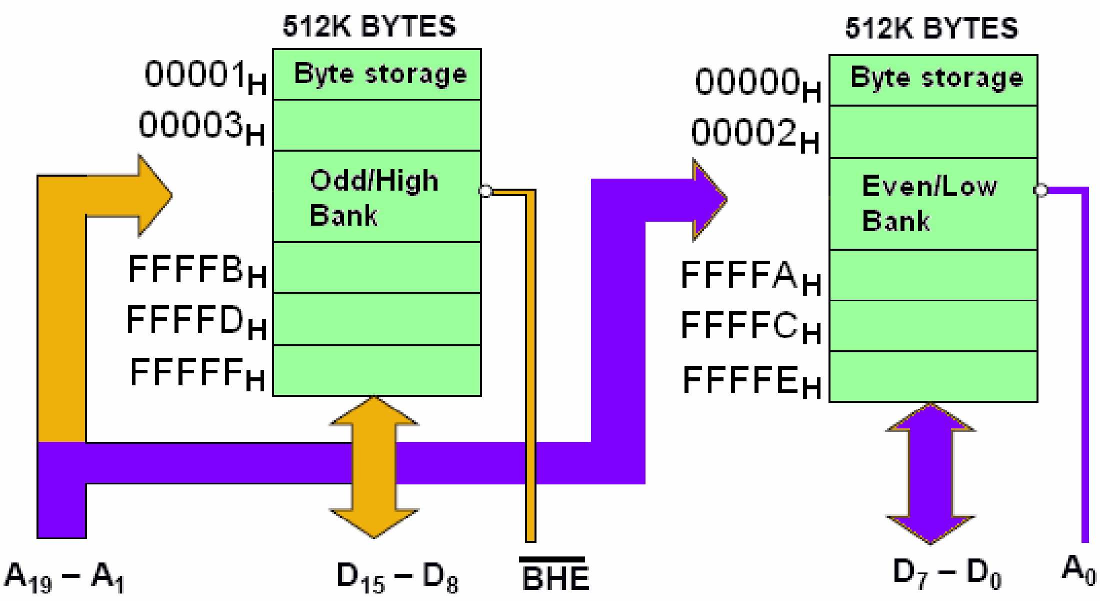
- Data-bytes associated with even-addresses reside in low bank and odd-address reside in high bank. Address pins A_1 to A_{19} selects the storage locations, whereas A_0 and $\overline{\text{BHE}}$ pins are used to enable high or low memory banks.



Slide 5: **Accessing a Byte-data in a 8086-based Memory System:**

- The two bank memory module of 8086 based storage system requires one bus-cycle to read/write a data-byte.

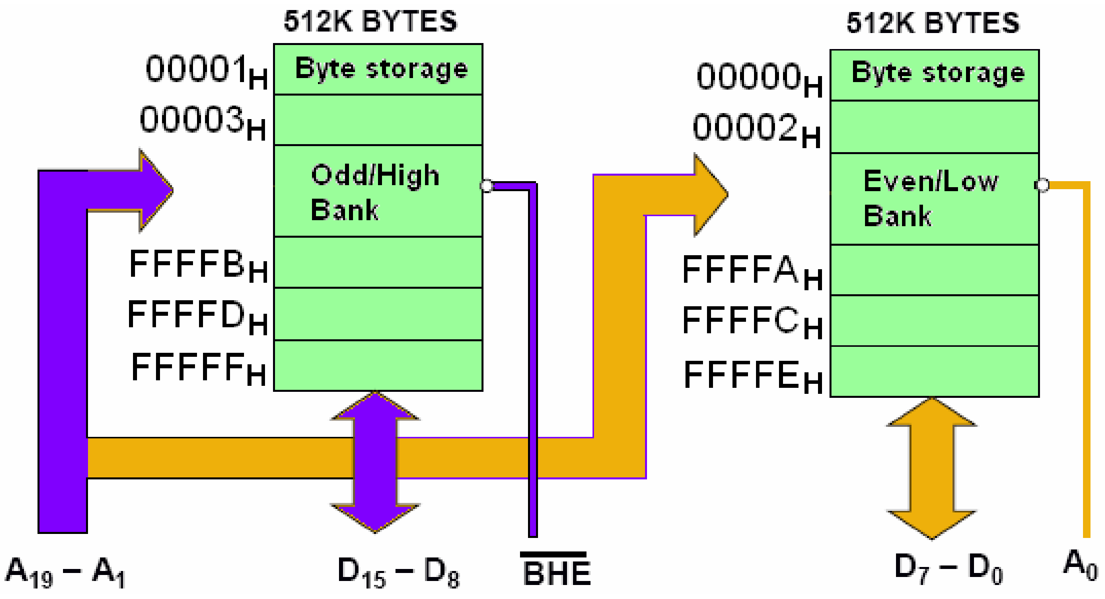
- To access a Byte of data in Low-bank, valid address is provided via address pins A_1 to A_{19} together with **A_0='0'** *and* **BHE='1**'.



Slide 6: **Accessing a Byte-data in a 8086 System (Cont'd):**

- Similarly to access a Byte of data in High-bank, valid address in pins A_1 to A_{19}, **A_0='1'** *and* **BHE='0'** are required to access the data through D_8 to D_{15} of the data-bus.

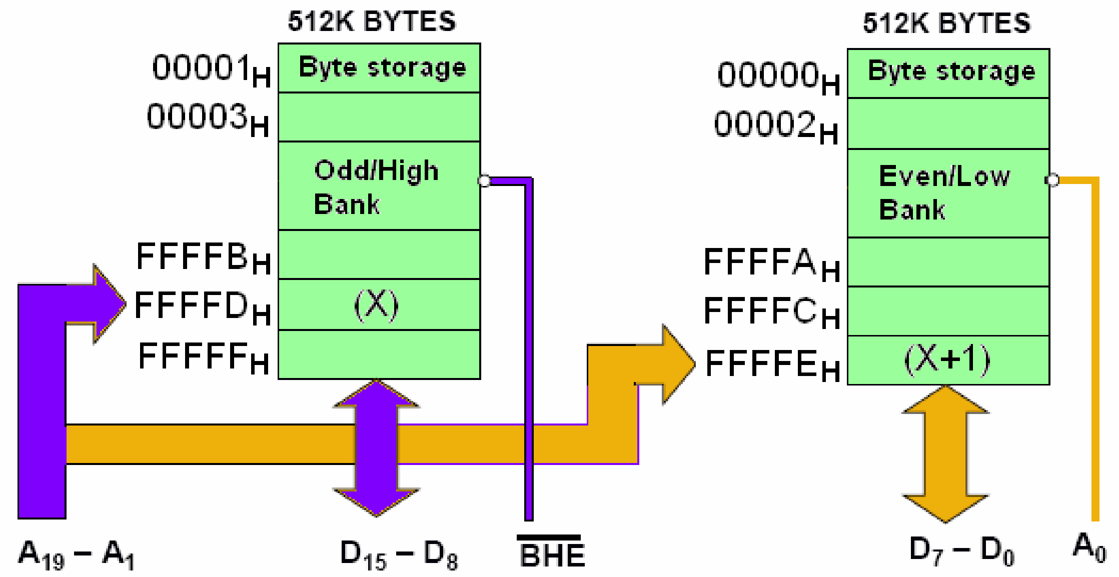
- These signals disable the Low bank and enable the High bank to transfer (in/out) data through D_8 to D_{15} of the data-bus.



Slide 7: **Accessing a Unaligned Word-data in a 8086 System:**

- For odd-addressed (unaligned) words (with odd P.A of the LSB), two bus-cycles are required to access the Word-data.

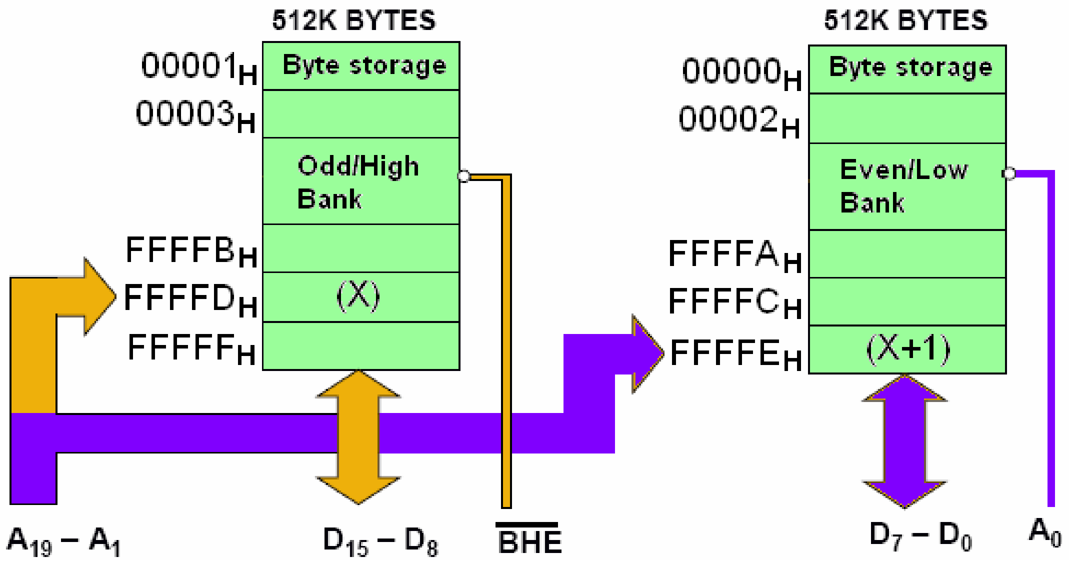
- During the 1st bus-cycle, odd addressed LSB of the word is accessed from the High-memory -bank via D_8 to D_{15} of data bus



Slide 8: **Accessing a Unaligned Word-data in a 8086 (Cont'd):**

- During 2nd bus-cycle, P.A. is auto-incremented to access the even address MSB of the word from the Low bank via D_0 to D_7.

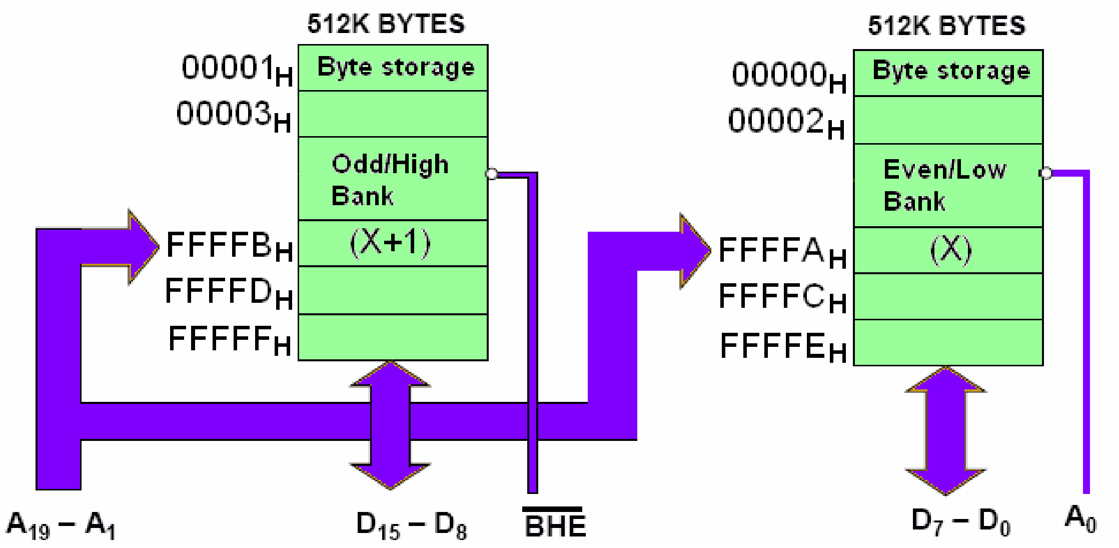
- Note that $\mathbf{A_0}$ *and* $\overline{\textbf{BHE}}$ signals are reset (violet) accordingly to enable the required memory bank.



Slide 9: **Accessing a Alighted Word-data in a 8086 System:**

- For even-addressed (aligned) words, only one bus-cycle is needed to access the word, as both low and high banks are activated at the same time using **A_0='0'** *and* **$\overline{\text{BHE}}$='0'**

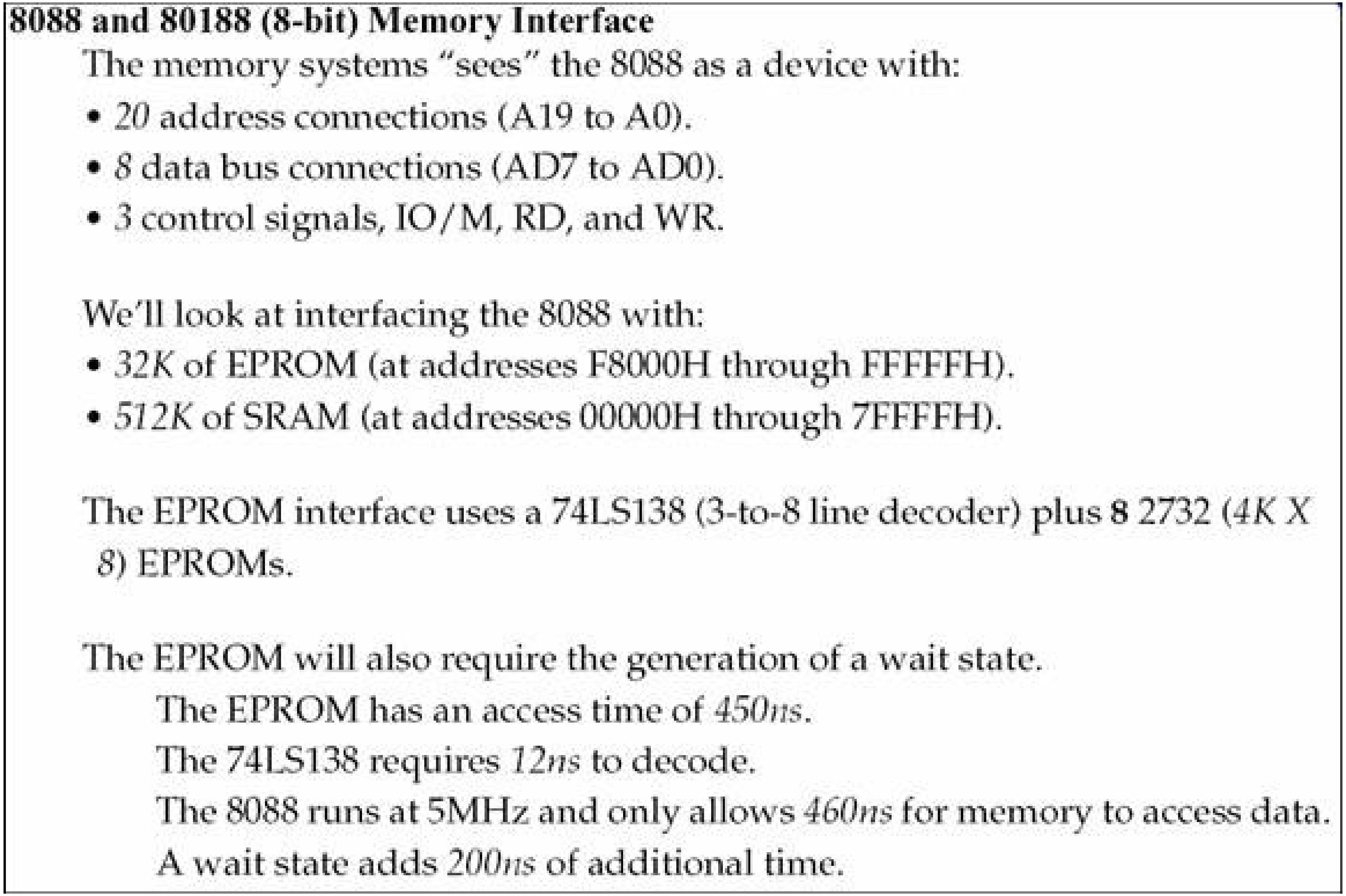
- Note that during this bus-cycle, all 16-bit data is transferred via D_0 to D_{15} of the data bus.



Slide 10: **Example on accessing 8088 and 8086 memory spaces:**

Q1: In an 8086's based memory system, how may bus cycles are needed to access a data-word stored in a physical address starting from of DS:$124E_H$.

Sol: One bus cycles are required, as the data stored is an even addressed or aligned word (as physical address of the LSB is even)



8088 and 80188 (8-bit) Memory Interface

The memory systems "sees" the 8088 as a device with:

- *20* address connections (A19 to A0).
- *8* data bus connections (AD7 to AD0).
- *3* control signals, IO/M, RD, and WR.

We'll look at interfacing the 8088 with:

- *32K* of EPROM (at addresses F8000H through FFFFFH).
- *512K* of SRAM (at addresses 00000H through 7FFFFH).

The EPROM interface uses a 74LS138 (3-to-8 line decoder) plus **8** 2732 (*4K X 8*) EPROMs.

The EPROM will also require the generation of a wait state.

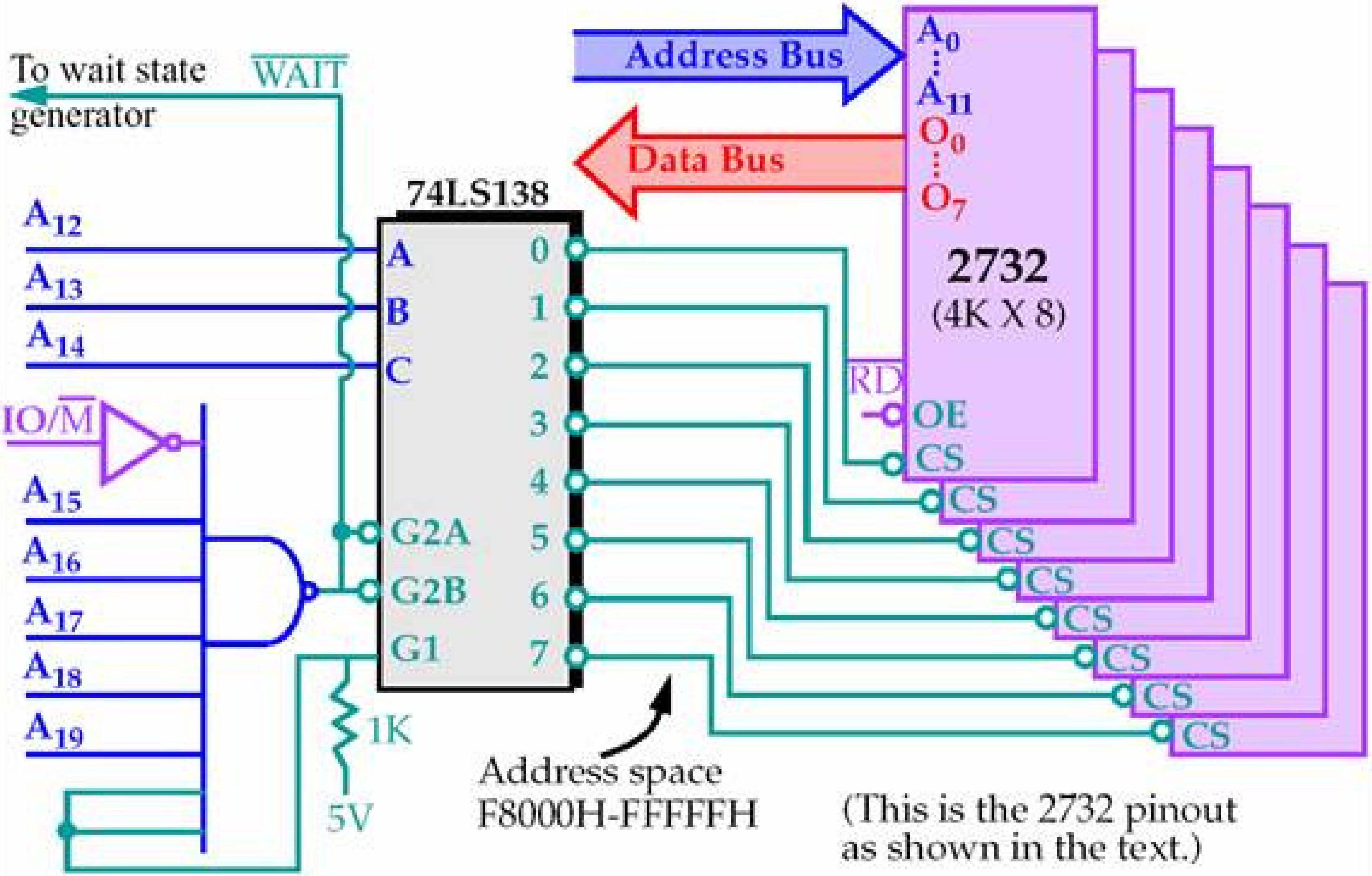
The EPROM has an access time of *450ns*.

The 74LS138 requires *12ns* to decode.

The 8088 runs at 5MHz and only allows *460ns* for memory to access data.

A wait state adds *200ns* of additional time.

8088 and 80188 (8-bit) EPROM Memory Interface

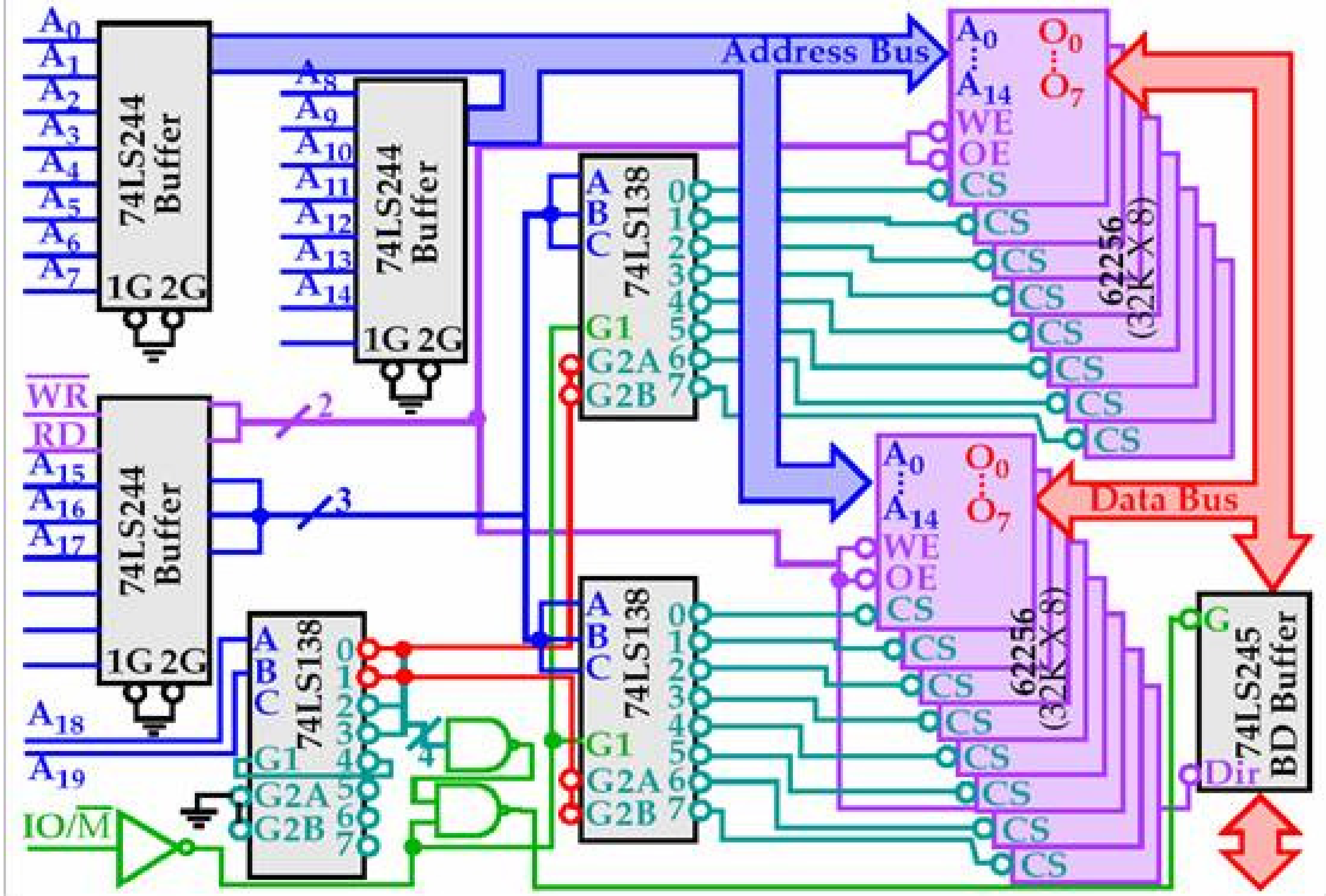


Address space F8000H-FFFFFH

(This is the 2732 pinout as shown in the text.)

The 8088 cold starts execution at **FFFF0H**. JMP to F8000H occurs here.

8088 and 80188 (8-bit) RAM Memory Interface



8086 - 80386SX 16-bit Memory Interface

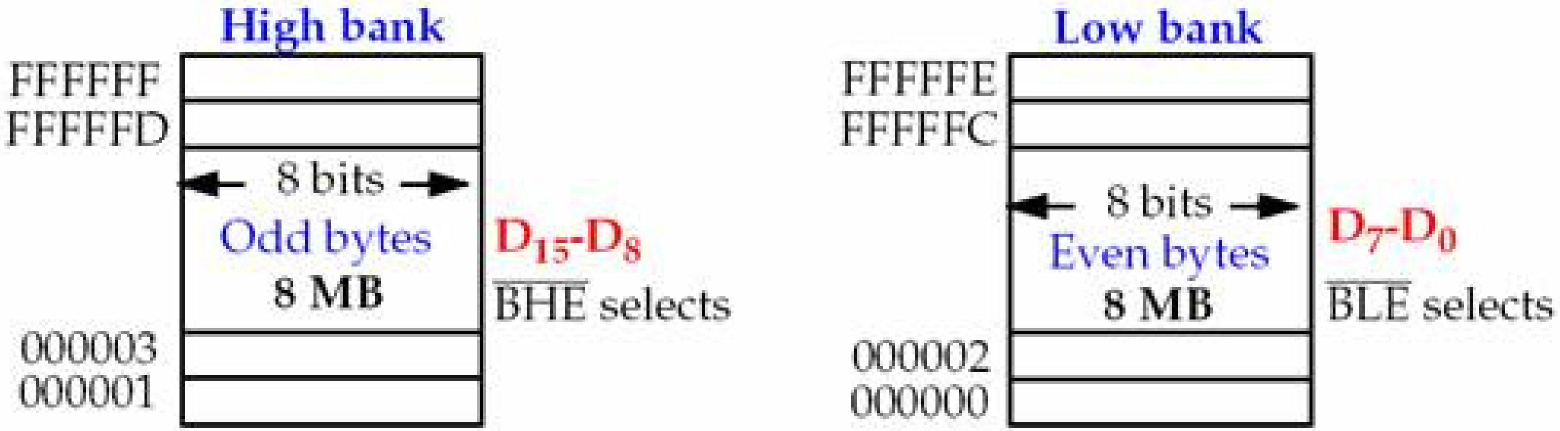
These machines differ from the 8088/80188 in several ways:

- The data bus is *16-bits* wide.
- The IO/$\overline{\text{M}}$ pin is replaced with M/$\overline{\text{IO}}$ (8086/80186) and $\overline{\text{MRDC}}$ and $\overline{\text{MWTC}}$ for 80286 and 80386SX.
- $\overline{\text{BHE}}$, **Bus High Enable,** control signal is added.
- Address pin A_0 (or $\overline{\text{BLE}}$, **Bus Low Enable**) is used differently.

The 16-bit data bus presents a new problem:

The microprocessor must be able to read and write data to any 16-bit location in addition to any 8-bit location.

The data bus and memory are divided into banks:



8086 - 80386SX 16-bit Memory Interface

BHE and BLE are used to select one or both:

$\overline{\text{BHE}}$	$\overline{\text{BLE}}$	Function
0	0	Both banks enabled for 16-bit transfer
0	1	High bank enabled for an 8-bit transfer
1	0	Low bank enabled for an 8-bit transfer
1	1	No banks selected

Bank selection can be accomplished in two ways:

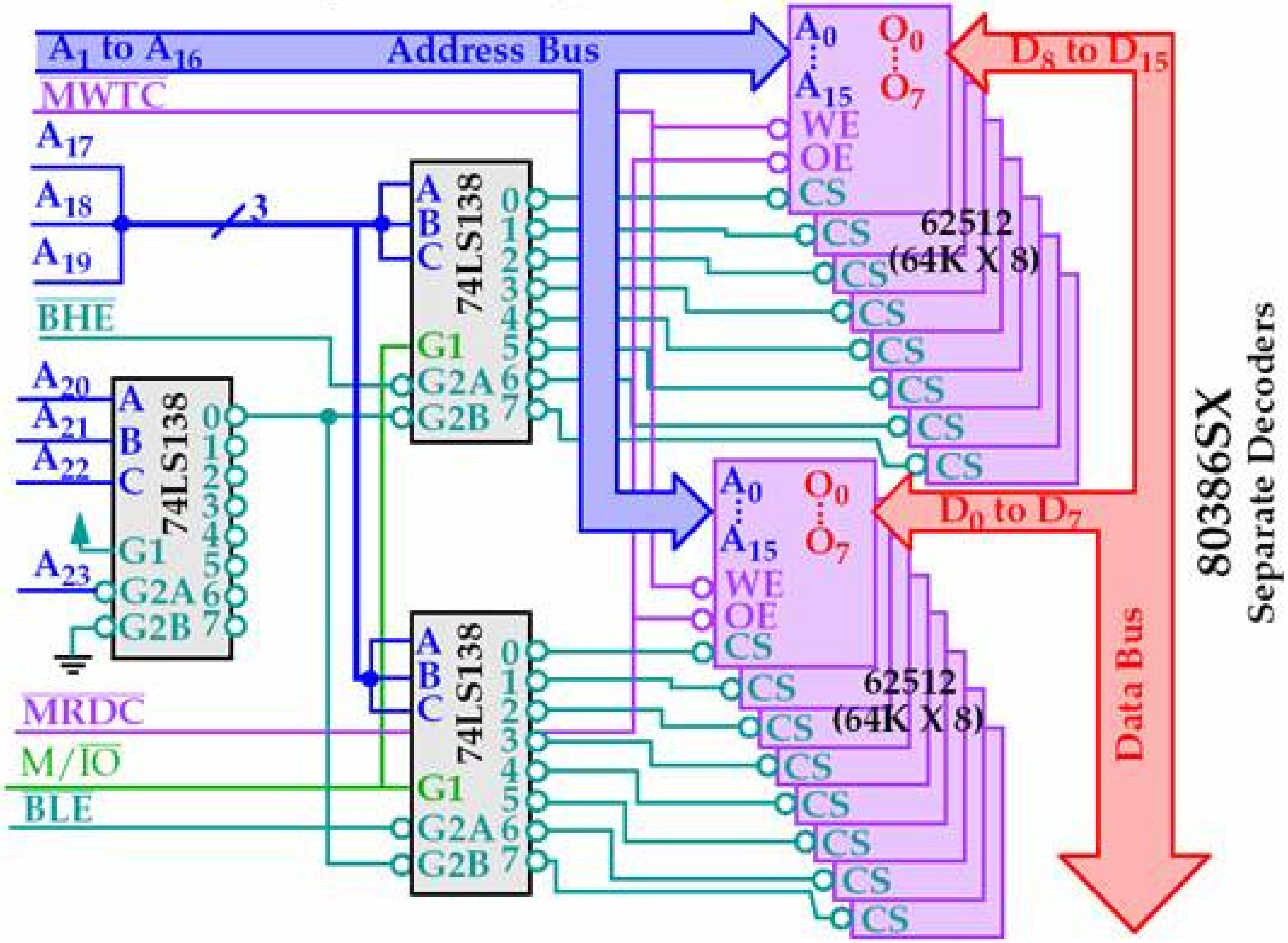
- Separate write decoders for each bank (which drive $\overline{\text{CS}}$).
- A separate write signal (strobe) to each bank (which drive $\overline{\text{WE}}$).

  Note that 8-bit read requests in this scheme are handled by the microprocessor (it selects the bits it wants to read from the 16-bits on the bus).

There does not seem to be a big difference between these methods although the book claims that there is.

Note in either method that A_0 does not connect to memory and bus wire A_1 connects to memory pin A_0, A_2 to A_1, etc.

80386SX 16-bit Memory Interface (Separate Decoders)


A_1 to A_{16}
Address Bus
$\overline{MWTC}$
A_{17}
A_{18}
A_{19}
3
$\overline{BHE}$
A_{20}
A_{21}
A_{22}
A_{23}
$\overline{MRDC}$
$M/\overline{IO}$
$\overline{BLE}$
74LS138
A B C G1 G2A G2B
0 1 2 3 4 5 6 7
A_0 ⋮ A_{15}
O_0 ⋮ O_7
WE
OE
CS
62512 (64K X 8)
D_8 to D_{15}
D_0 to D_7
Data Bus
80386SX
Separate Decoders