Memory Banks

Lecture 9

Decoding Memory Address

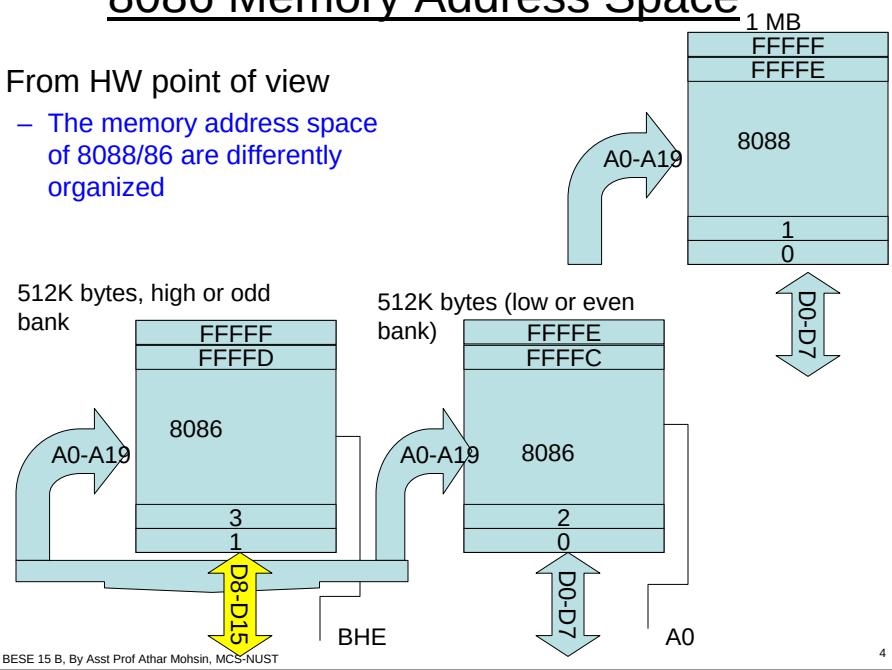
- The processor can usually address a memory space that is much larger than the memory space covered by an individual memory chip.
  - In order to splice a memory device into the address space of the processor, decoding is necessary.
    - For example, the 8086 issues **20-bit**addresses for a total of **1MB** of memory address space.
    - However, the BIOS on a 2716 EPROM has only 2KB of memory and **11**address pins.
  - A decoder can be used to decode the additional 9 address pins and allow the EPROM to be placed in **any** 2KB section of the 1MB address space

Data Bus

- The width of the data bus dictates how much information can flow in each clock cycle
  - In order to take advantage of the full width of the processor's data bus, it is necessary to arrange the system memory so that each clock cycle, the full data bus width can be transferred at once.
  - In fact, most systems require the system memory to be arranged so that this is the case.

8086 Memory Address Space

- From HW point of view
  - The memory address space of 8088/86 are differently organized



Low and high banks

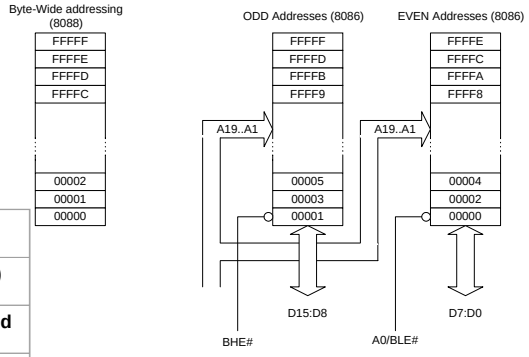
- For 8086 address bits A1 to A19 selects the storage location that is to be accessed
  - These are applied to both the banks in parallel
- A0 (BLE) and BHE are used as bank select signals
  - A0 = 0
    - identifies an even addressed byte of data and low bank of memory will be enabled
  - BHE' = 0
    - enables the high bank and an odd addressed byte of data
  - Both the banks transfer byte size data

Memory Banks

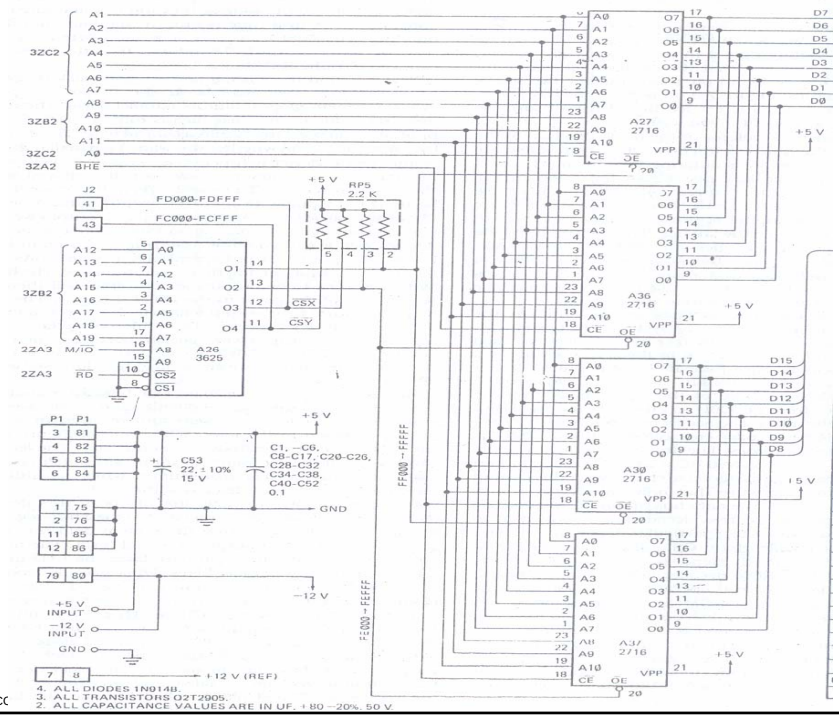
- 8086 has a 20 bit address bus so a 1MB
  - Each address represents a stored byte
  - The word is actually written at two consecutive memory addresses
    - To make it possible to read and write a word in one machine cycle the memory is setup as two separate banks of 512 bytes each
  - One memory bank contains all the bytes which have an even address
    - 00000, 00002 and 00004, the data line of this bank is connected to the lower eight data line D0-D7
  - Other memory bank contains all the bytes which have an odd addresses like 00001, 00003 and 00005
    - The data line of this bank is connected to upper eight lines of data bus
- Address line A0 is used as part of enabling for the lower memory bank → if A0 will be low, lower bank will be enabled
  - A1-A19 will be used to select desired memory device and location in the bank
- BHE' will be used as enabling the upper memory bank → BHE' low

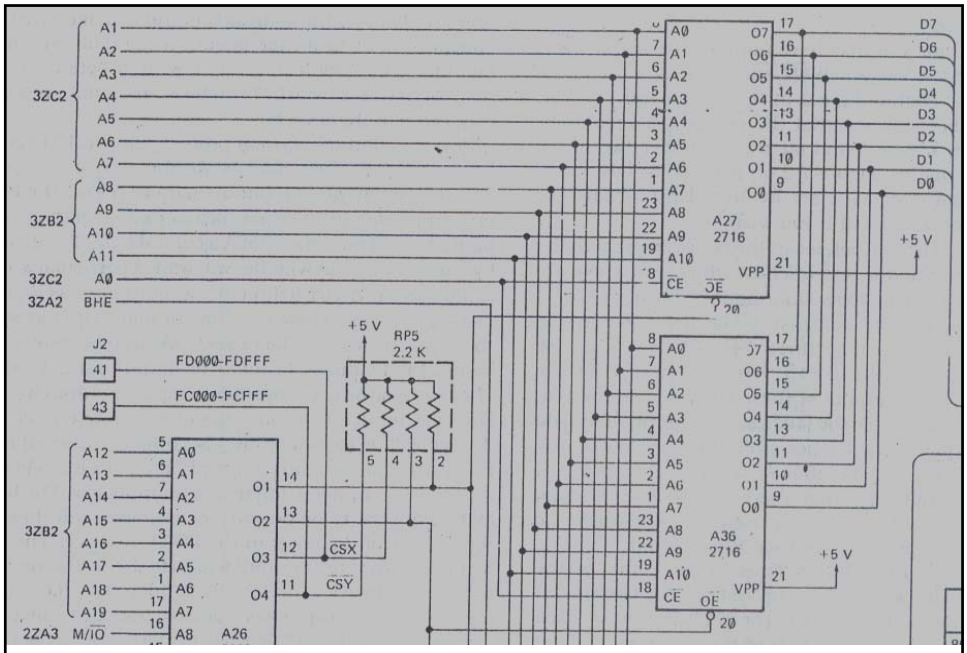
BHE Signal (8086 Only)

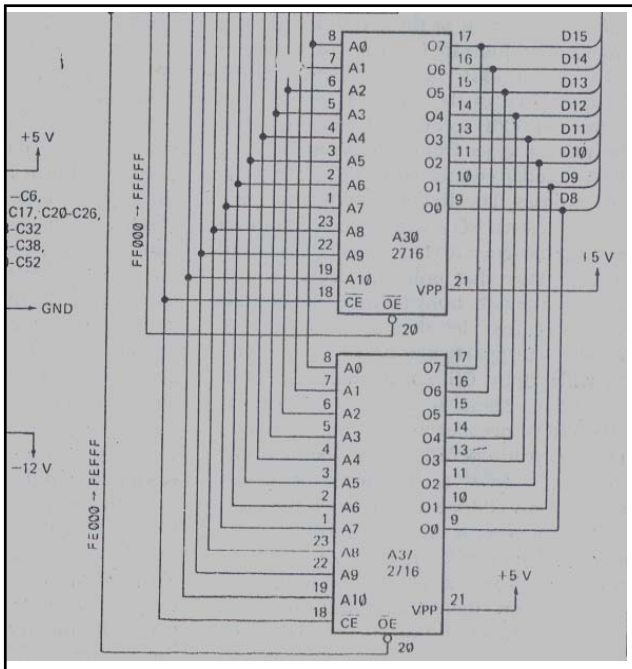
- The 8086 processor can address memory a byte at a time
  - Its data bus is 16b wide
  - It uses the BHE# signal and A0 (sometimes called BLE#) to address bytes using its 16b bus
- Use of BHE#/A0

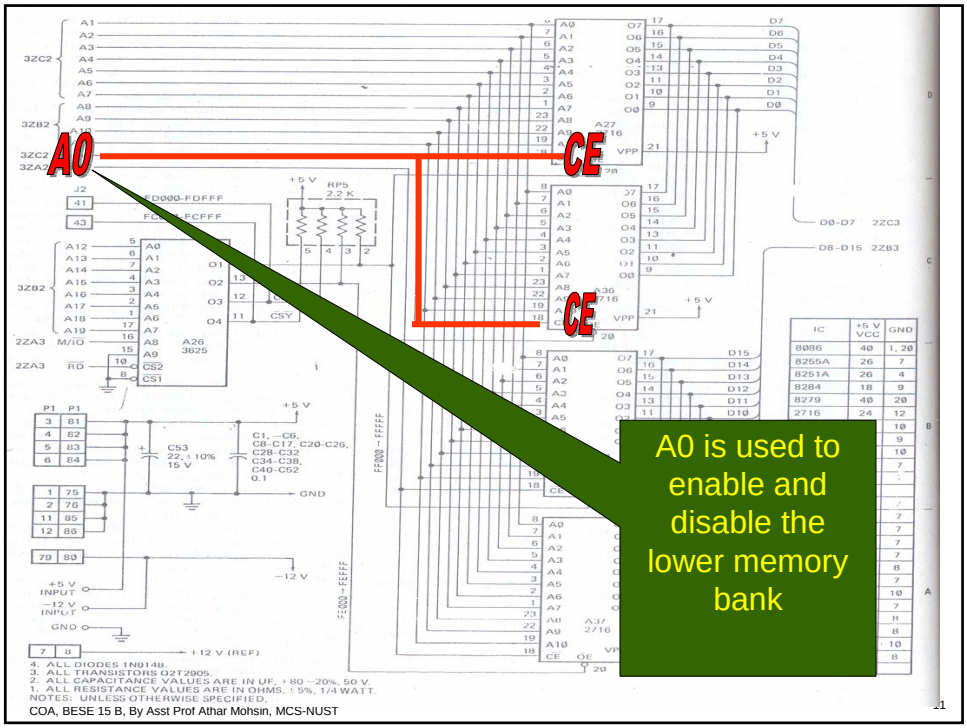


BHE	A0	Selection
0	0	Whole word (16-bits)
0	1	High byte to/from odd address
1	0	Low byte to/from even address
1	1	No selection




3ZC2
A1
A2
A3
A4
A5
A6
A7
3ZB2
A8
A9
A10
A11
3ZC2
A0
3ZA2
BHE
J2
41
FD000-FDFFF
43
FC000-FCFFF
+5 V
RP5
2.2 K
3ZB2
A12
A13
A14
A15
A16
A17
A18
A19
2ZA3
M/IO
A26
O1
O2
O3
O4
CSX
CSY
A27
2716
A36
2716
CE
OE
VPP
20
21
D7
D6
D5
D4
D3
D2
D1
D0


+5 V
–C6,
C17, C20-C26,
-C32
-C38,
-C52
GND
–12 V
FF000 → FFFFF
FE000 → FEFFF
A30
2716
A37
2716
CE
OE
VPP
20
21
D15
D14
D13
D12
D11
D10
D9
D8

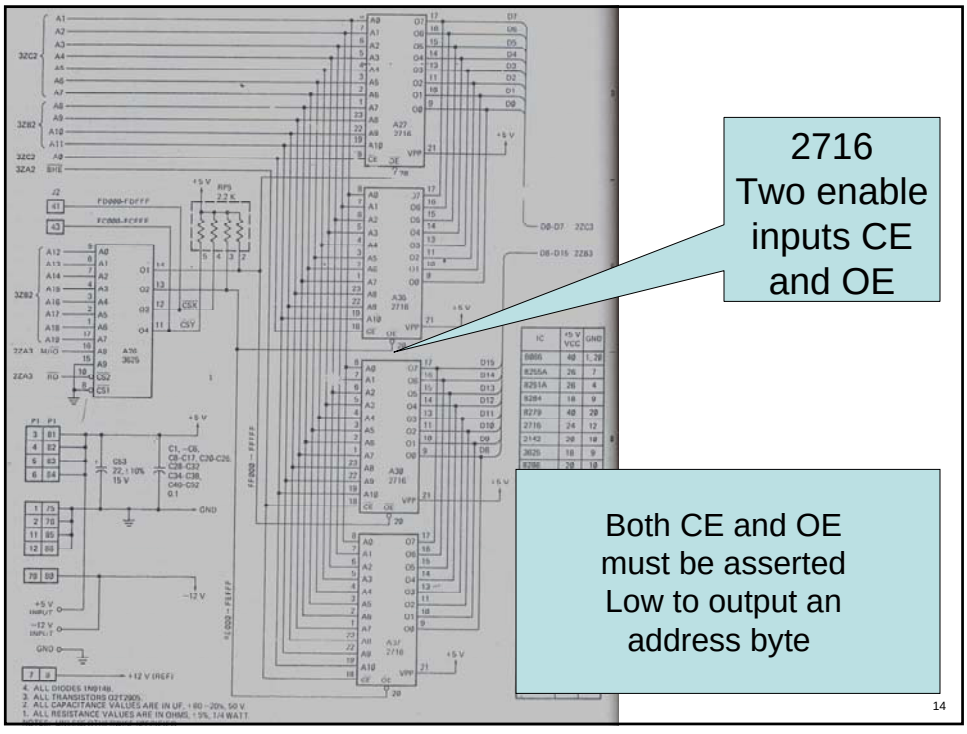


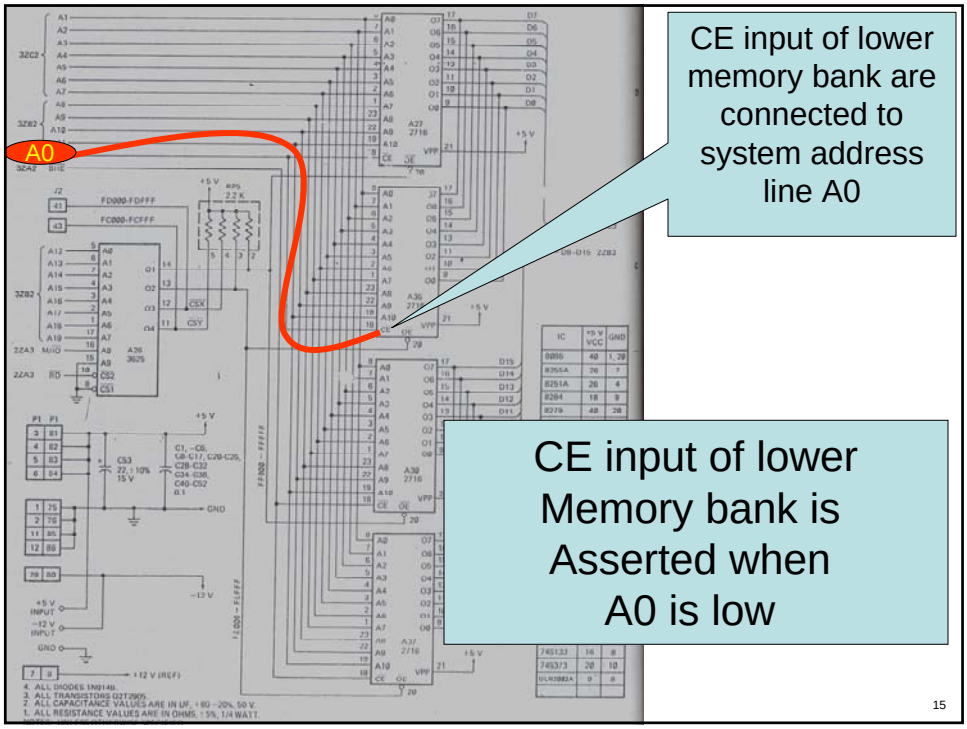
Memory bank

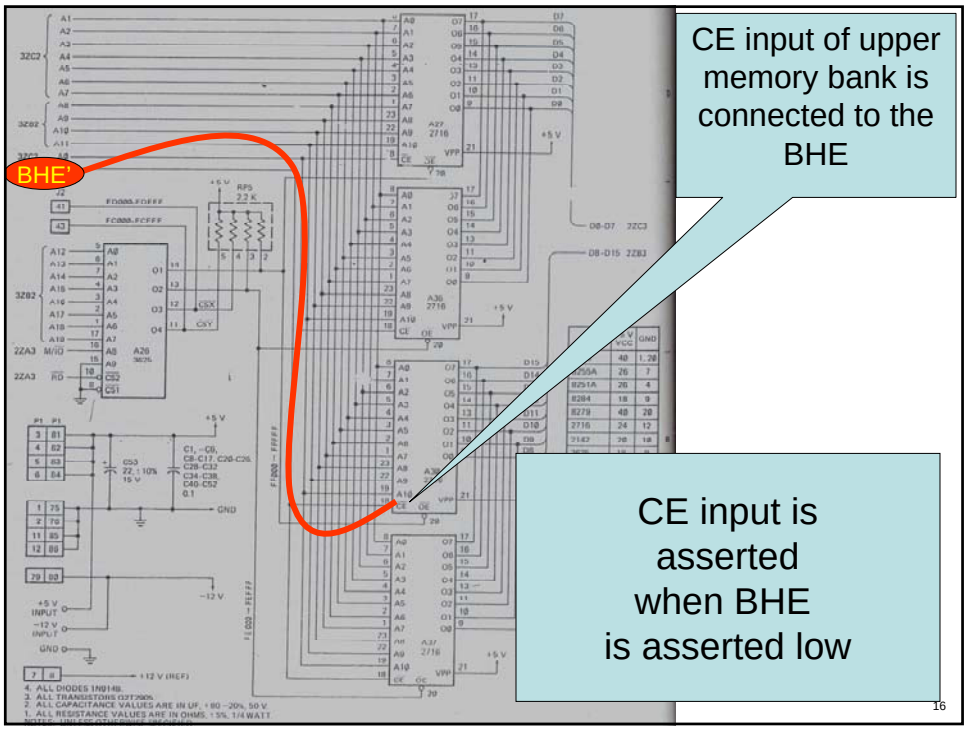
- Address line A0 is used as part of the enabling bit for the memory device in the lower bank
  - The addressed memory device of this bank will be enabled when A0 line is low
- A1 – A11 will be used to select the desired byte from memory device in the bank
  - The upper memory bank also uses A1 –A11 lines for the selection of desired memory device and the desired byte in that bank
- BHE' signal multiplexed with ALE signal from 8086 is used to enable the upper memory bank
  - ALE strobe the latch and latch grab the address and hold it stable for rest of the machine cycle

Memory Bank

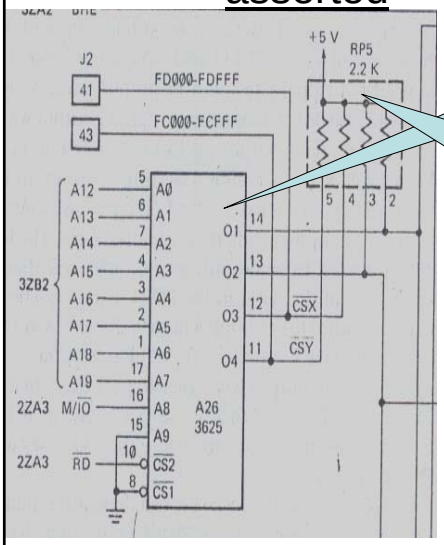
- To read a byte from or writ a byte to an even address A0 will be low and BHE' will be high
  - The lower bank will be enabled and the upper bank will be disabled
  - A byte will be transferred to or from addressed location in the lower bank (D0- D7)
- For odd address A0 will be high and BHE will be low and lower bank will be disabled and higher bank will be enabled
  - A byte will be transferred to or from addressed location in the upper bank (D8- D15)
- Odd bank Selection
    - Eleven address lines (A1-A11) are required to address the 2 Kbytes in each device
    - So A1- A11 are connected to the ROM in parallel
  - *A0 is not used to select a byte in PROM so it is used to enable or disable the lower memory bank*




A0
CE input of lower memory bank are connected to system address line A0
CE input of lower Memory bank is Asserted when A0 is low
15


BHE'
CE input of upper memory bank is connected to the BHE
CE input is asserted when BHE is asserted low
16

How OE enable input get asserted



3625 1k x4 bipolar PROM functions as address decoder

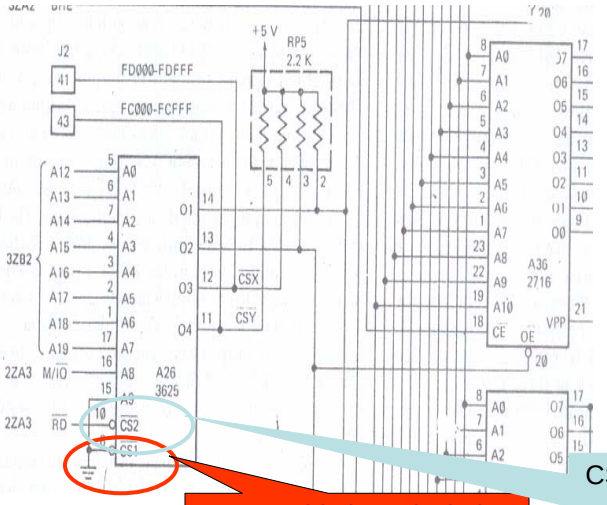
3625 is an open collector outputs
A pull up resister pack to +5v is connected to each output

Address decoder translates an address to a signal which is used to enable the desired device

Why 3625?

- It has an advantage over other Decoders as:-
  - It can be programmed
  - Can move the memory device to new addresses in memory by simple programming
  - Large number of inputs allows the selection of a specific area of memory
- Decoder's truth table
  - To analyze the decoder circuit the truth table (vendor's worksheet) is used because:
    - The input output can not be determined simply by schematic layout
  - To assert the O1 output low
    - M/Io has to be high (to enable memory not to port)
    - Address lines A12 – A19 high

What signals are required to enable the decoder?



To avoid accidental write operation

CS1 enable input is tied to ground to enable it permanently

CS2 enable input is tied to RD signal from 8086 enable it during read operation only

PROM Decoder truth table

PROM INPUT				PROM OUTPUT				PROM Address Block Selected
M/IO	A14-A19	A13	A12	04	03	02	01	
1	1	1	1	1	1	1	0	FF000H-FFFFFH
1	1	1	0	1	1	0	1	FE000H-FEFFFH
1	1	0	1	1	0	1	1	FD000H-FDFFFH
1	1	0	0	0	1	1	1	FC000H-FCFFFH
All Other States				1	1	1	1	NONE