Lecture 9

8086 Memory Organization

Topics

- 8086 Memory Organization
- 8086 Address Decoding Methods

Memory Types

- ROM (Read Only Memory) : Contents are not erased when power is off.
  Used to store programs such as BIOS permanently that does not change.
- Non-volatile memory types (ROM):
  - Read only memory (ROM)
  - Programmable read-only memory (PROM)
  - Erasable programmable ROM (EPROM)
  - Electrically erasable programmable ROM (EEPROM)
  - Flash memory

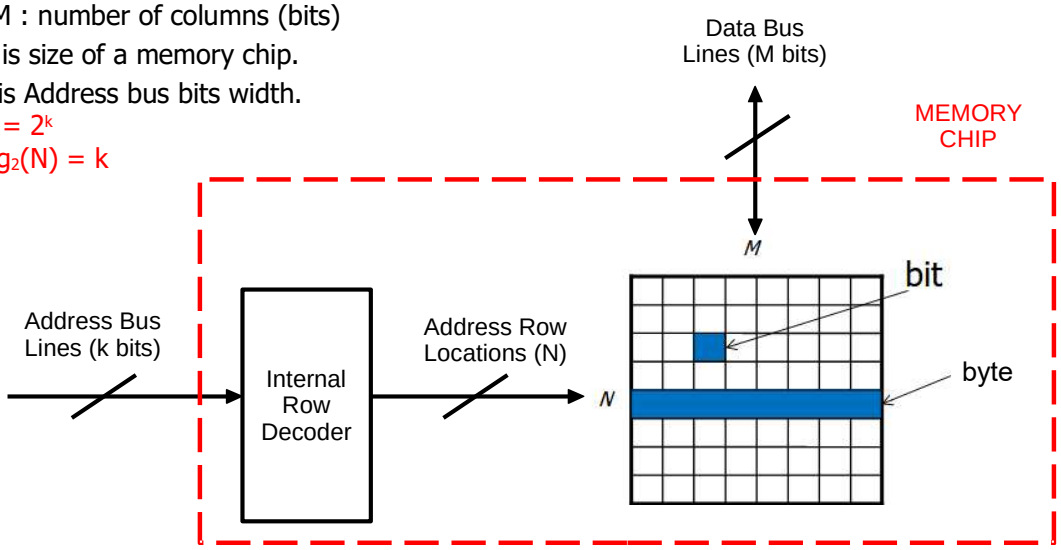
- RAM (Random Access Memory) : Contents are erased when power is off.
  Used to store programs and data temporarily.
- Volatile memory types (RAM):
  - Dynamic random-access memory (DRAM)
    Refreshed periodically.
    Used in Main memory.
  - Static random-access memory (SRAM)
    Read/write is faster.
    Used in Cache memory for high-speed.

Comparison of Memory Units

Memory Unit	Example Capacity	Example Speed
Registers	8, 16, 32, 64 Bits	1 nano seconds
Cache Memory	4, 8 Mega Bytes	10 - 100 nano seconds
Main Memory	2, 4, 8, 16, 32 Giga Bytes	100 - 150 nano seconds

Memory Chips

- Each row location in a memory chip is called a **word**.
- Each word is composed of M bits (Data bus bits width).
- CPU can access memory at the word level, not at the bit level.
- Number of bits in a word can be different for different microprocessors.
- Memory is a matrix of NxM bits
  N : number of row locations (words)
  M : number of columns (bits)
- N is size of a memory chip.
- k is Address bus bits width.
  $N = 2^k$
  $\log_2(N) = k$



Address lines calculations

Example1: Suppose a memory chip has 10 address lines. Calculate the number of memory row locations.

Answer: 2^{10} = 1024 locations (1K).

Example2: Suppose a memory chip is 4 K Byte capacity. Calculate the number of address lines required.

Answer: 4 KB = 4*1024 = 4096 row locations.
$\log_2(4096) = \log_2(2^{12}) = 12$ address lines required.

8086 Memory Organization

- Every microprocessor based system has a memory subsystem.
- All systems contain two basic types of memory,
  - Read only memory (ROM, PROM, EPROM, EEPROM, etc.)
  - Random access memory (RAM) (read/write memory).
- ROM contains system software and permanent system data such as lookup tables, Interrupt Vector Table, etc.
- RAM contains temporary data and application software. Content of RAM are lost when system power is turned off.
- ROMs/PROMs/EPROMs should be mapped to cover the CPU's reset address, since these are non-volatile.
- When 8086 is turned on, the first instruction is fetched from the memory location FFFF0H (20-bit absolute address).
- In a 8086-based computer, the address location FFFF0H must be always in ROM.

8086 Memory Organization

- In 8086-based systems, mostly 8-bit memory chips are used.
- The memory chips have 8 bits (1 byte) in each memory location.
- 8086 CPU is a 16-bit microprocessor, so it can transfer 16-bit data.
- In addition to byte (8-bit), word (16-bit) has to be stored in the memory.
- To allow both byte-level and word-level transfer, the entire memory is divided into two memory banks: High bank and Low bank.
- High bank is selected only when A0 line in Address Bus is zero.
- Low bank is selected only when $\overline{\text{BHE}}$ signal in Control Bus is zero.

Logical and Physical Memory Organizations

- The memory address space of the 8086-based microcomputers has logical and physical organizations.

- Logically, memory is organized as a single 1M × 8 bit memory chunk.
- The 8-bit data storage locations are assigned consecutive addresses (20 bit) over the range from 00000H through FFFFFH.

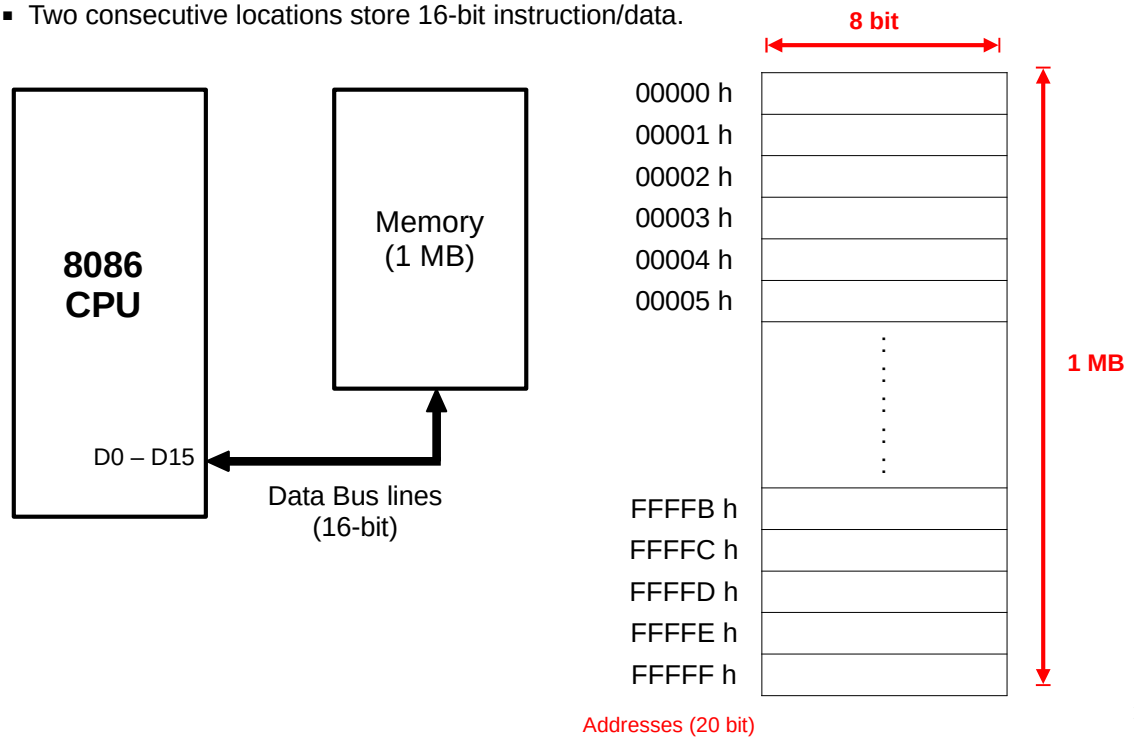
- Physically, memory is organized as two independent 512 Kbyte banks. The low (even) bank and the high (odd) bank.
- Data bytes associated with an even address (00000H, 00002H, etc.) reside in the low bank.
- Those with odd addresses (00001H, 00003H, etc.) reside in the high bank.
- Address bits A1 through A19 select the storage location that is to be accessed.
- They are applied to both banks in parallel.

- The A0 address line and the Bus High Enable ($\overline{BHE}$) signal are used as bank-select signals.
- The memory locations 00000-FFFFF are designed as odd and even bytes.

- A bank can be implemented by using only one memory chip (512 KB), or by using multiple memory chips with smaller KB capacity.

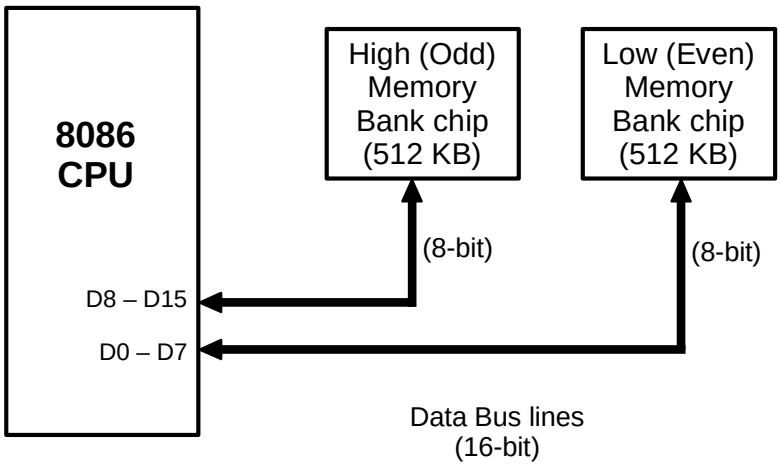
8086 Logical Memory Organization

- For the programmer, 8086 memory address space is a sequence of 1 M bytes.
- One location stores an 8-bit instruction/data.
- Two consecutive locations store 16-bit instruction/data.

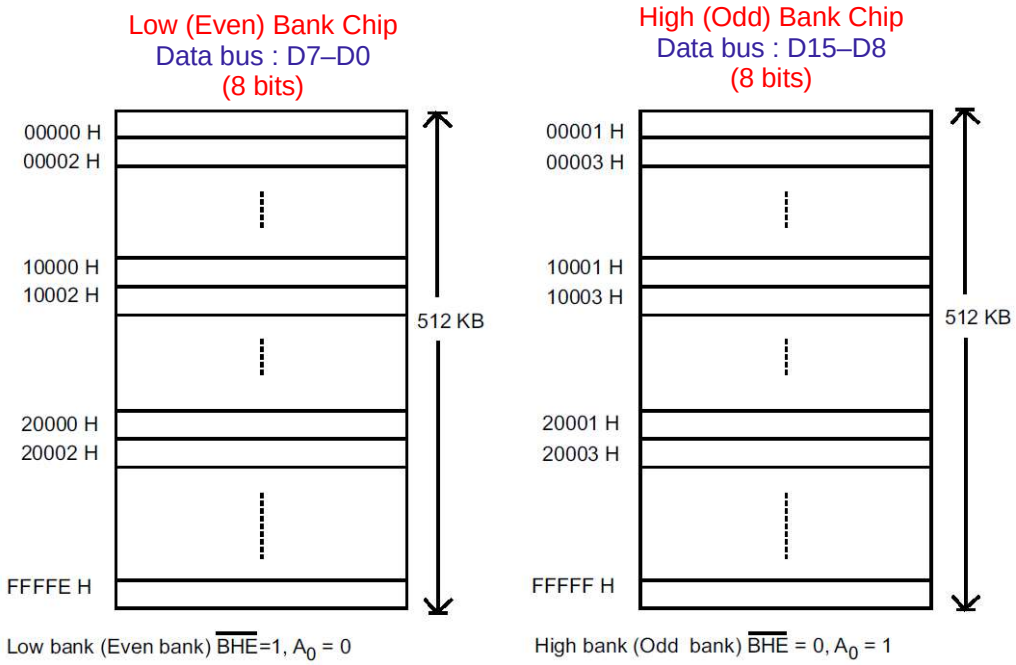


8086 Physical Memory Chip Organization (High and Low Banks)

- Physically (in hardware), the 1M byte memory space is organized as two memory banks of 512K byte. (512 KB + 512 KB = 1 MB).
- The two memory banks are called Low (Even) bank and High (Odd) bank.
- Each memory bank can be implemented with one or more memory chips.



8086 Physical Memory Chip Organization (Address Map of High and Low Banks)



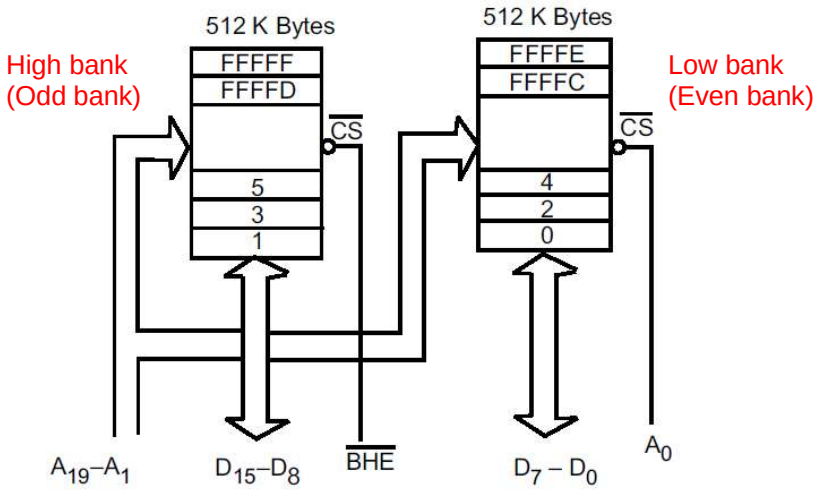
CPU Signals for Bank Selection

- The memory is divided into odd and even banks.
- CPU can access both banks concurrently to read 16 bits of data in one clock cycle.
- CPU can also access the banks separately to read 8 bits of data.
- To distinguish between Odd and Even bytes, the CPU provides a signal called BHE (bus high enable).
- BHE signal and A0 line (lowest bit in address bus) are used to select the odd and even byte.

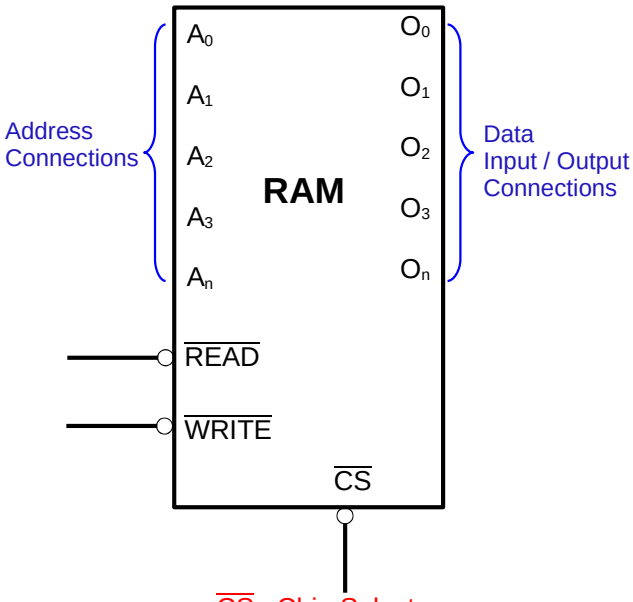
$\overline{\text{BHE}}$ (Bus High Enable)	A0 (First line of address bus)	Selected Memory Bank
0	0	Both odd and even banks. (16-bit word transfer on AD15 - AD0)
0	1	Only odd (high) bank. (1 byte transfer on AD15 - AD8)
1	0	Only even (low) bank. (1 byte transfer on AD7 - AD0)
1	1	None of the banks is selected.

CPU Signals for Bank Selection

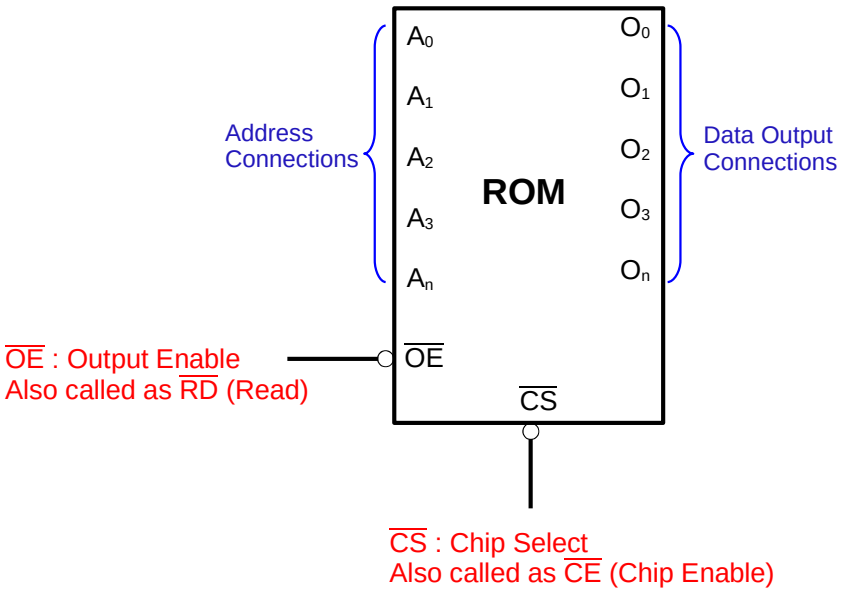
- In High bank, the high data bits D15-D8 are selected. (Also called as Odd bank, which means addresses are odd : 1, 3, 5, 7, etc.)
- In Low bank, the low data bits D7-D0 are selected. (Also called as Even bank, which means addresses are even : 0, 2, 4, 6, etc.)
- Each bank is = 512 KB, Total memory is = 512*2 = 1 MB
- Selection of a memory bank is determined by its CS (Chip Select) input.



General Pin Diagram of RAM chips



General Pin Diagram of ROM chips



Connections of ROM and RAM chips

Address connections:

- All memory devices have address inputs that select a memory location within the memory device.
- Address inputs are labeled from A0 to An.

Data connections:

- All memory devices have a set of data outputs or input/outputs.
- ROM data connections are one-directional.
- RAM data connections are two-directional.

Selection connections:

- Each memory device has an input that selects or enables the memory device.
- The input is also called as Chip Select ($\overline{CS}$), or Chip Enable ($\overline{CE}$).
- RAM memory generally has at least one $\overline{CS}$ input.
- ROM memory has at least one $\overline{CE}$ input.
- If $\overline{CS}$ or $\overline{CE}$ input is active, the memory device perform the read or write operation.
- If it is inactive, the memory device cannot perform read or write operation.

Connections of ROM and RAM chips

Control connections:

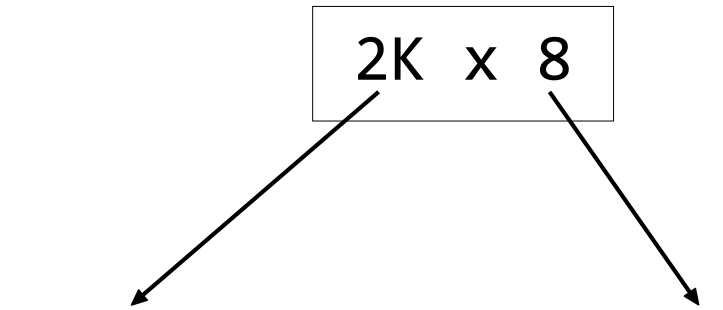
- ROM usually has only one control input.
- The control signal most often found on the ROM is the Output Enable (OE), or the Gate ($\overline{G}$) .
- It allows data to flow out of the output data pins of the ROM.
- RAM memory device has either one or two control inputs.
- If there is one control input it is called as R / $\overline{W}$.
- It selects a Read operation or a Write operation, only if the device is selected by the selection input ($\overline{CS}$).

EPROM Chip Examples (8-bit data bus)

EPROM Chip Name	Number of Address Bus Lines	Total Bits Capacity	Number of Row Locations and Data Bus Bits	Maximum address (Hexadecimal)
2716	11	16k bits	2k x 8	7FF
2732	12	32k bits	4k x 8	FFF
2764	13	64k bits	8k x 8	1FFF
27128	14	128k bits	16k x 8	3FFF
27256	15	256k bits	32k x 8	7FFF
27512	16	512k bits	64k x 8	FFFF
27010	17	1024k bits	128k x 8	1FFFF
27020	18	2048k bits	256k x 8	3FFFF
27040	19	4096k bits	512k x 8	7FFFF
27080	20	8192k bits	1M x 8	FFFFF

In chip names, 27 is a standard prefix.
Other numbers indicate the number of kilo bits.

Example : Chip capacity notation
(For a 2 Kilo Bytes memory chip)

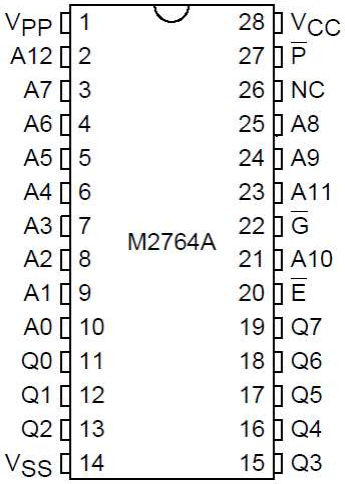


Number of row locations
2K rows
= 2 * 1024 rows
= 2048 rows

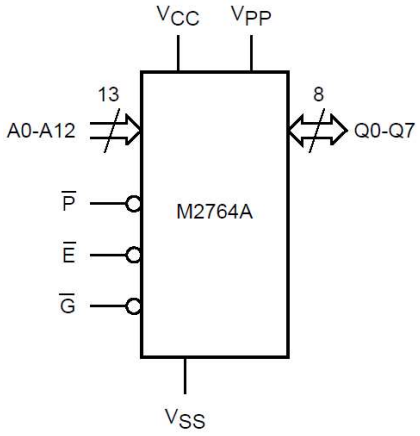
Number of Data Bus bits
(Number of bits at each row location)
= 8 bits

Example1 : Memory Chip 2764 (8K x 8 bit EPROM)

Pin Connections



Logic Diagram



Operating Modes

Mode	$\overline{E}$	$\overline{G}$	$\overline{P}$	A9	V_{PP}	Q0 - Q7
Read	V_{IL}	V_{IL}	V_{IH}	X	V_{CC}	Data Out
Output Disable	V_{IL}	V_{IH}	V_{IH}	X	V_{CC}	Hi-Z
Program	V_{IL}	V_{IH}	V_{IL} Pulse	X	V_{PP}	Data In

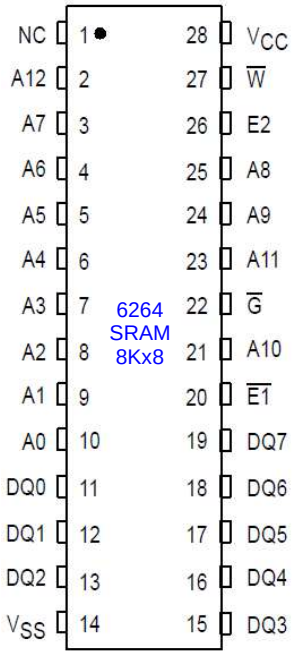
RAM Chip Examples (8-bit data bus)

RAM Chip Name	Number of Address Lines	Total Bits Capacity	Number of Locations and Data Bus Bits	Maximum address (Hexadecimal)
6116	11	16k bits	2k x 8	7FF
6264	13	64k bits	8k x 8	1FFF
62256	15	256k bits	32k x 8	7FFF
628128	17	1M bits	128k x 8	1FFFF
628512	19	4M bits	512k x 8	7FFFF

In chip names, 61 and 62 are standard prefixes.
Other numbers indicate the number of kilo bits.

Example2 : Memory Chip 6264
(8K x 8 Bit Static RAM)

Pin Connections



PIN NAMES	
A0 – A12	Address Input
DQ0 – DQ7	Data Input/Data Output
$\overline{W}$	Write Enable
$\overline{G}$	Output Enable
$\overline{E1}$, E2	Chip Enable
V_{CC}	Power Supply (+ 5 V)
V_{SS}	Ground

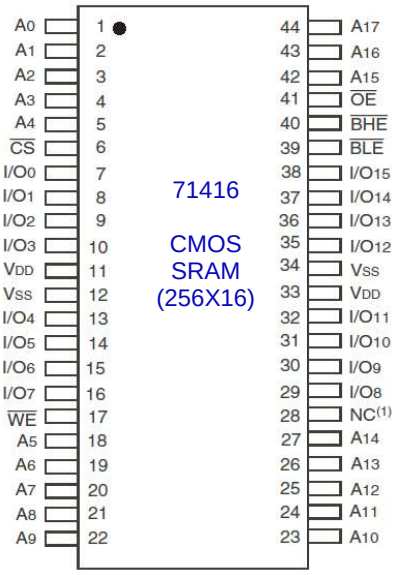
G: Gate data to output pin (Output Enable)
VCC: Voltage Common Collector
VSS: Voltage Source Supply

TRUTH TABLE (X = Don't Care)

$\overline{E1}$	E2	$\overline{G}$	$\overline{W}$	Mode
H	X	X	X	Not Selected
X	L	X	X	Not Selected
L	H	H	H	Output Disabled
L	H	L	H	Read
L	H	X	L	Write

Example3 : Memory Chip 71416
(256K x 16 Bit Static RAM)

Pin Connections



Pin Descriptions

A0 - A17	Address Inputs	Input
$\overline{CS}$	Chip Select	Input
$\overline{WE}$	Write Enable	Input
$\overline{OE}$	Output Enable	Input
$\overline{BHE}$	High Byte Enable	Input
$\overline{BLE}$	Low Byte Enable	Input
I/O0 - I/O15	Data Input/Output	I/O
VDD	3.3V Power	Pwr
VSS	Ground	Gnd

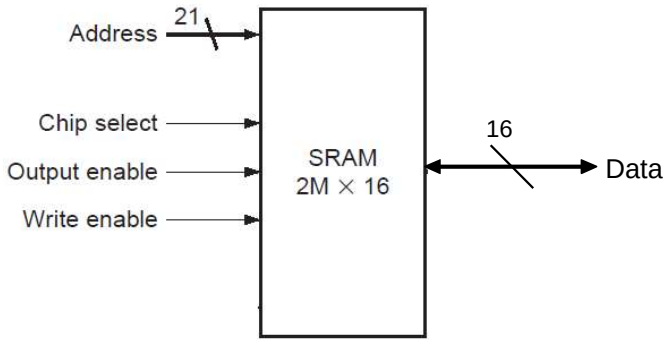
Truth Table(1)

$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	$\overline{BLE}$	$\overline{BHE}$	I/O0-I/O7	I/O8-I/O15	Function
H	X	X	X	X	High-Z	High-Z	Deselected - Standby
L	L	H	L	H	DATAOUT	High-Z	Low Byte Read
L	L	H	H	L	High-Z	DATAOUT	High Byte Read
L	L	H	L	L	DATAOUT	DATAOUT	Word Read
L	X	L	L	L	DATAIN	DATAIN	Word Write
L	X	L	L	H	DATAIN	High-Z	Low Byte Write
L	X	L	H	L	High-Z	DATAIN	High Byte Write
L	H	H	X	X	High-Z	High-Z	Outputs Disabled
L	X	X	H	H	High-Z	High-Z	Outputs Disabled

NOTE:
1. H = VIH, L = VIL, X = Don't care.

Example4 : Generic Memory Chip (2M x 16 Bit Static RAM)

- To initiate a read or write access, the Chip select signal must be made active.
- For read operation, the Output enable signal must also be activated, that controls whether or not the data selected by the address is actually driven on the pins.
- The Output enable is useful for connecting multiple memories to a single-output bus and using Output enable to determine which memory drives the bus.
- For write operation, the followings must be supplied:
  Data to be written, Address, Control signals to cause the write to occur.
- When both Write enable and Chip select are true, the data on data input lines is written into the location cell specified by the address.

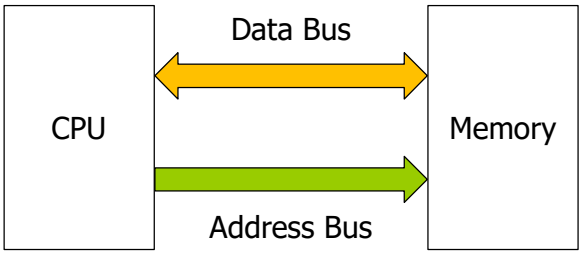


Topics

- 8086 Memory Organization
- 8086 Address Decoding Methods

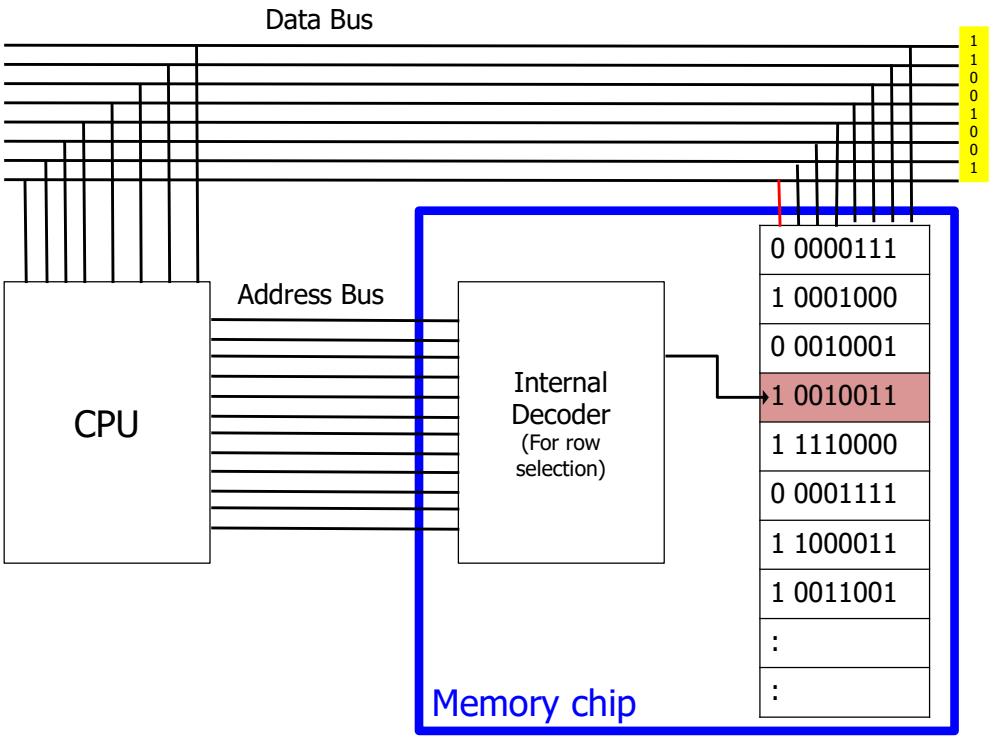
Memory Addressing

- Each location in memory has an address.
- CPU first identifies the location's address and then passes the address on address bus.
- Data are transferred between memory and CPU along data bus.



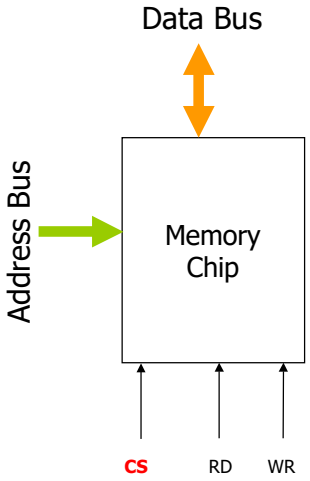
- Number of bits that can be transferred on data bus at once is called the **data bus width** of processor.

Accessing a memory location



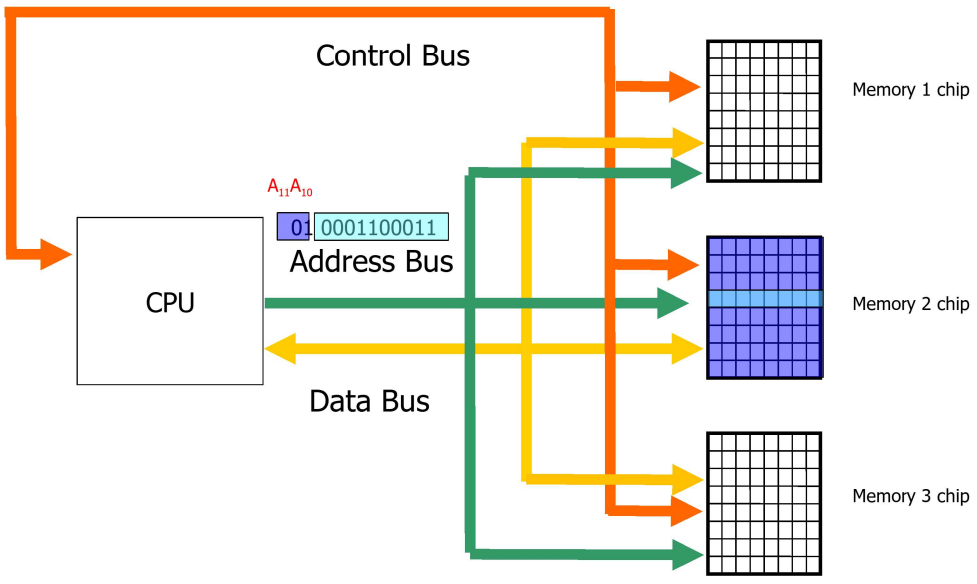
Memory Chip Selection

- Each memory chip has a **Chip Select (CS) input** (EN=Enable).
- The chip will only work if an active signal (high or low) is applied on CS.
- To allow use of **multiple memory chips**, some of address lines are used for the purpose of **chip selection**.



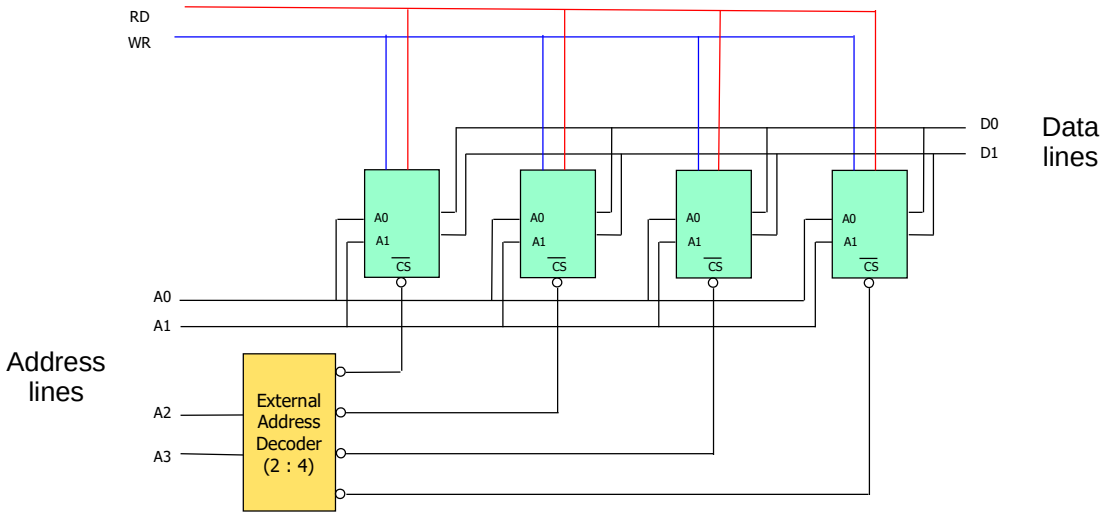
Memory Access thru Chip Selection

- Example: Three memory chips have consecutive memory addresses.
- **A11** and **A10** address lines are used for **chip selection** (Using 2-to-4 address decoder).
- **A9-A0** lines are used for **location selection** within a chip.



Using Address Decoder for Memory Chip Selection

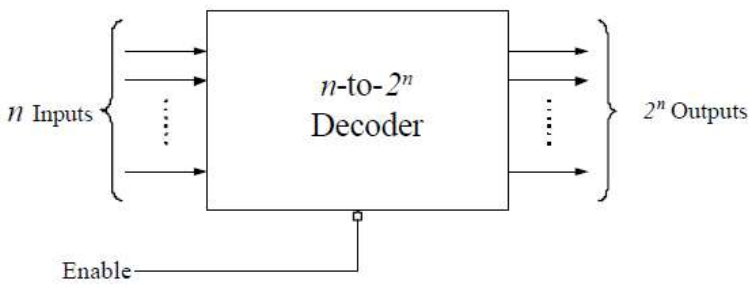
- The following is a simple memory system made up of 4 memory chips.
- Each memory chip has 4x2 bits.
- An external **Address Decoder** (2 to 4) is used for chip selections.
- The Decoder outputs are connected to chip selects (CS) of memory chips.



Address Decoder

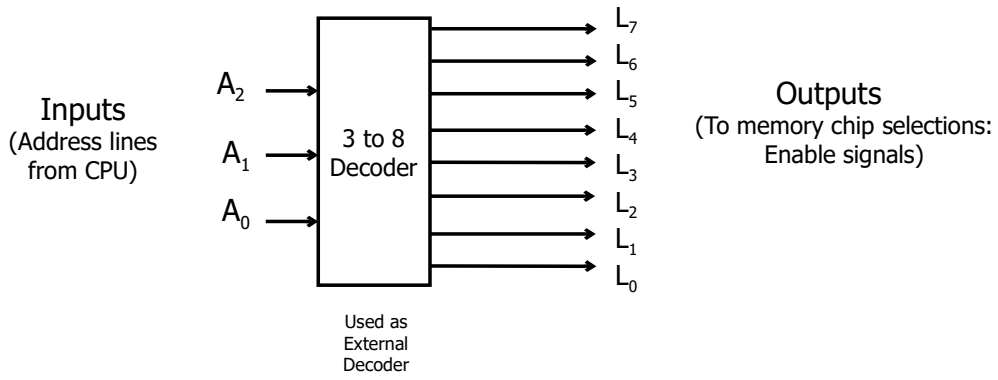
- Decoder is a circuit (usually as a chip) that converts binary information from n-coded inputs to maximum of 2^n outputs.
- It is also called as Demultiplexer.
- Only one output is 1, others are 0.

- **Address Decoders** are used for **location (row) selection** within a memory chip. (Internal Decoder)
- They are also used to for **memory chip selection** among multiple memory chips. (External Decoder)



Example : 3-to-8 Decoder

- There are 3 input lines.
- There are 8 output lines. (2^3)
- Depending on inputs, only one output line will be 1, other output lines will be 0.



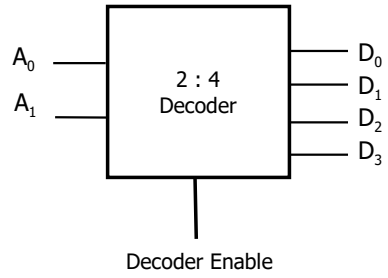
Truth Table for 3-to-8 Decoder

Only one output line is selected (as 1) at a time.

INPUTS			OUTPUTS							
A_2	**A_1**	**A_0**	**L_7**	**L_6**	**L_5**	**L_4**	**L_3**	**L_2**	**L_1**	**L_0**
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Example: 2-to-4 Decoder

Block Diagram

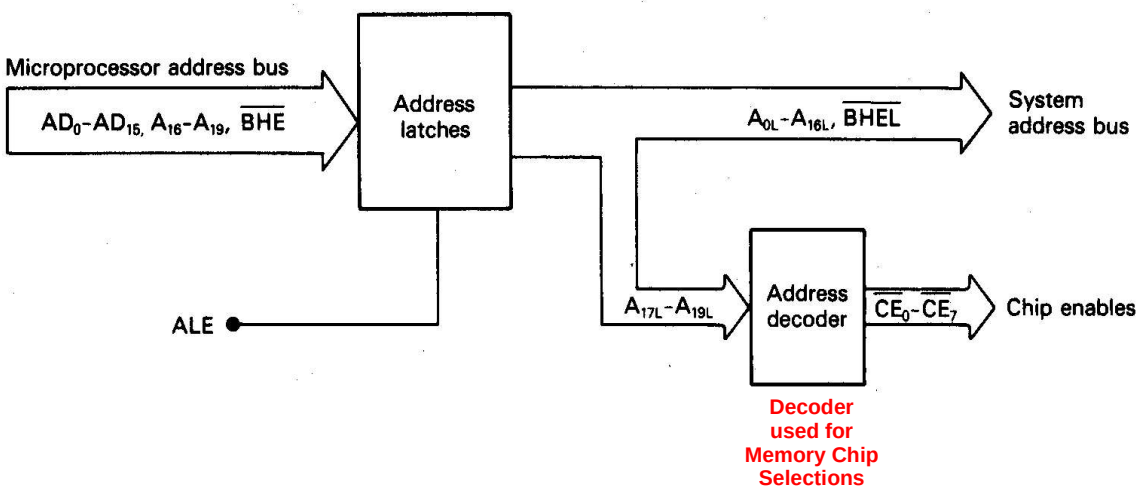


Truth Table

Inputs		Outputs			
A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

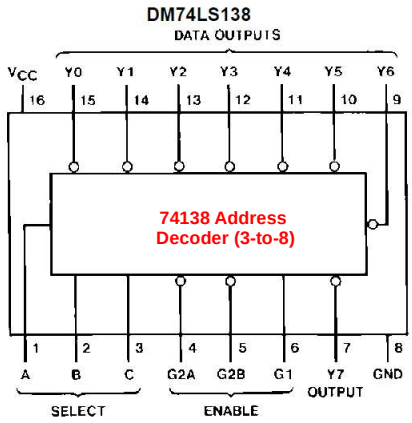
Example : Using a Decoder for Memory Chip Selections in 8086 CPU

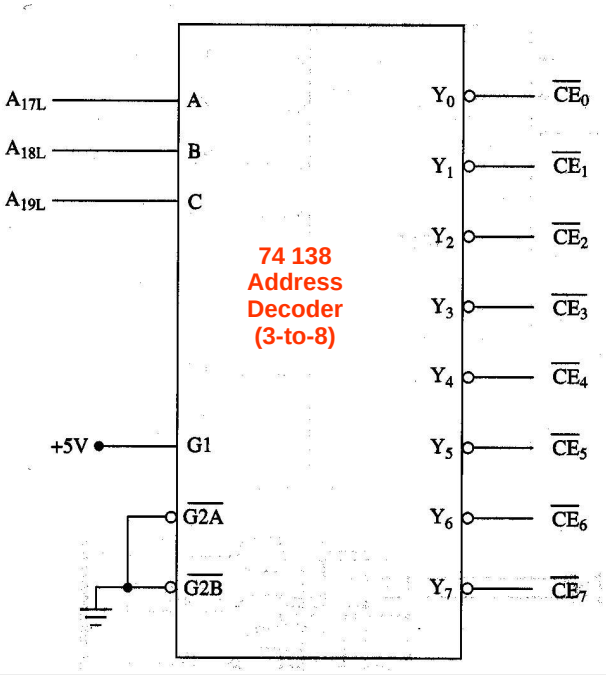
- If more than one memory chip and I/O interface chip is used, then using an Address Decoder is necessary.
- It uses some of the address lines to generate memory chip selection (enable) signals.
- Address decoder is used as part of the buffered/latched address design.



Address Decoder 74138 chip

- Suppose 74138 Address Decoder chip (3-to-8) is used for memory chip selections.
- Address inputs A19L , A18L , A17L (L suffix means Latched) of the Decoder are decoded to produce 8 independent Chip Enable outputs CE0 through CE7.





Address Decoding Methods for Memory Chip Selections

- In a 8086 based system, usually at least two memory chips are used (8-bit High bank chip, and 8-bit Low bank chip).
- High bank and Low bank chips should be identical.

- Alternative method: Instead of using two identical 8-bit memory chips, it is also possible to use one 16-bit memory chip.
- The 16-bit memory chips have two internal banks : High Bank and Low Bank. They support $\overline{BHE}$ (Bus High Enable) , and $\overline{BLE}$ (Bus Low Enable) selection inputs.

- If multiple memory chips (at least two chips) are used, then Address Decoding is necessary in order to distinguish and select a memory chip from others, as part of the addressing process.

- Before Address Decoding, we assume that the Latched address bus, $\overline{BHE}$ signal, and Demultiplexed Data bus are readily available for decoder interfacing.

- The followings are the basic decoding methods for memory chip, and I/O chip selection.
  1. Decoding with Logic Gates
  2. Decoding with an Address Decoder chip

1. Decoding with Logic Gates

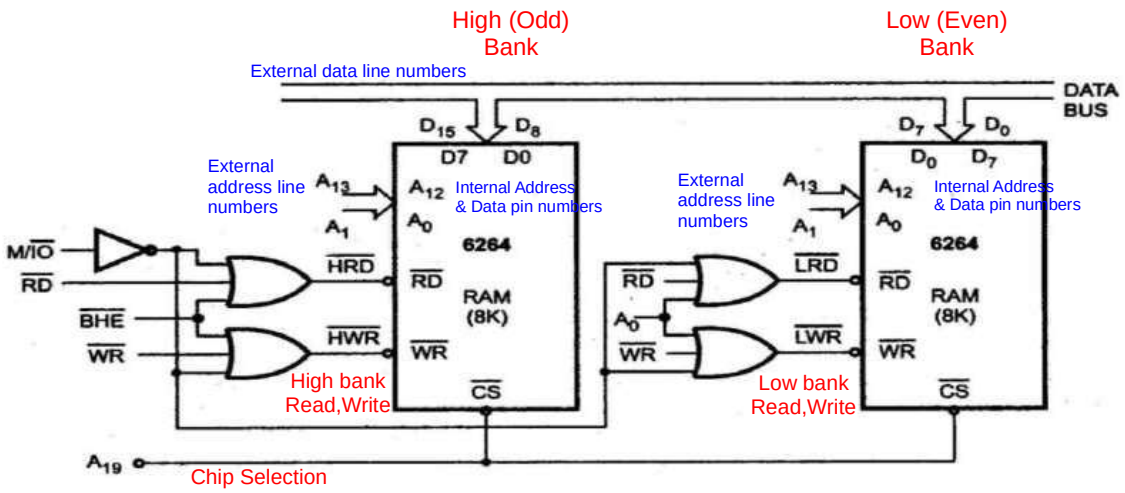
- Decoding logic can be implemented by using only the required number of addressing lines (not all), other address lines are not used.
- Control signal BHE and address line A0 are used to enable the High (Odd) bank and Low (Even) bank, respectively.
- In the following example, two 8Kx8 RAM chips are used (6264 chips).
- Total memory is 16KB.
- Calculation of number of address lines needed for location selection, in each memory chip:

  $\log_2(8K) = \log2\ (2^{13}) = 13$ lines.

- Logic gates (such as AND, OR, NAND, NOT, etc.) are used to provide the Decoding Logic for memory chip selections.

Example: Decoding with Logic Gates

- A1-A13 address lines are used for memory locations within each bank chip.
- A0 address line is used for Even Bank selection.
- $\overline{BHE}$ signal is used for Odd Bank selection.
- A19 address line is used for CS (Chip Select) inputs of both memory chips.
- When A19 is low, both memory chips are selected, otherwise both of them are disabled.
- The status of A14 thru A18 does not affect the chip selection logic. (But they cause shadow addresses.)
- Advantage : It reduces cost of decoding circuit.
- Disadvantage : It causes multiple addresses (shadow addresses).



Memory Map Table

- The memory map table shows the address range (smallest and biggest addresses) for each memory chip.
- Total 16 KB RAM memory is implemented by using two RAM chips.
- They are installed at sequential addresses without gap.

Memory Chip	Addresses (Hexadecimal)		A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAM1 8Kx8 (Even bank)	Smallest	**00000**	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Biggest	**03FFE**	0	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	0
RAM2 8Kx8 (Odd bank)	Smallest	**00001**	0	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Biggest	**03FFF**	0	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A19 is used for memory chip selection

A14 - A18 are not used (Can be assumed as zero)

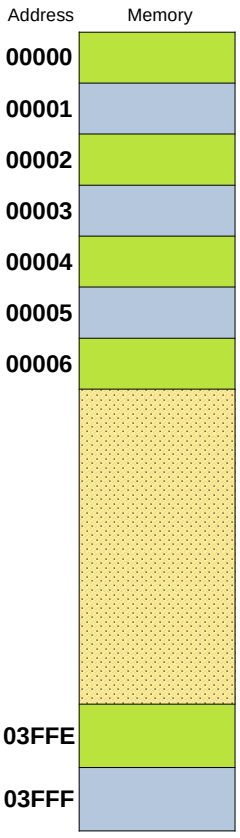
A1 - A13 are used for 8K location selection within memory chips

A0 is used for low bank read/write

($\overline{BHE}$ signal is used for high bank read/write)

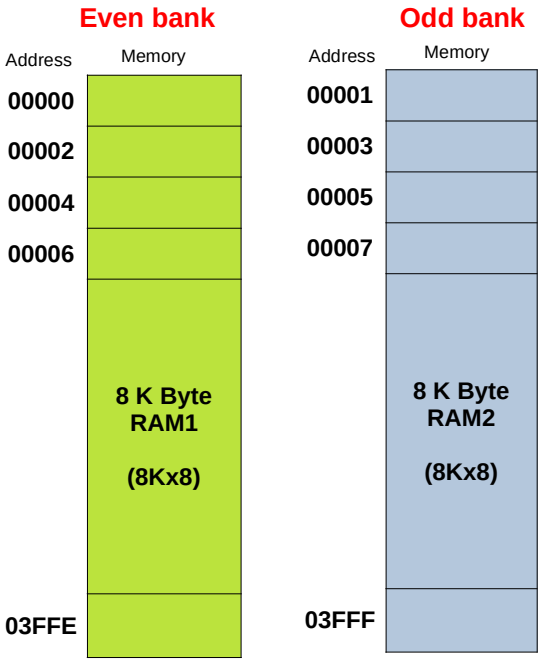
Logical Memory Map diagram

- Total = 16 K Byte RAM
- Two 8Kx8 RAM chips used.
- Even and Odd addresses are drawn as mixed.
- Each row is 1 byte.
- Memory is considered as flat array.



Physical Memory Map diagram

- Total = 16 K Byte RAM (Two 8Kx8 RAM chips used.)
- Even and Odd addressed banks are drawn as separate.



2. Decoding with an Address Decoder chip

- In a microcomputer system, the memory array is often consists of several blocks of memory chips.
- Each block of memory requires Decoding circuit.
- To avoid separate decoding logic circuits for each memory block, a special Address Decoder chip can be used to generate chip select signal for each memory block.

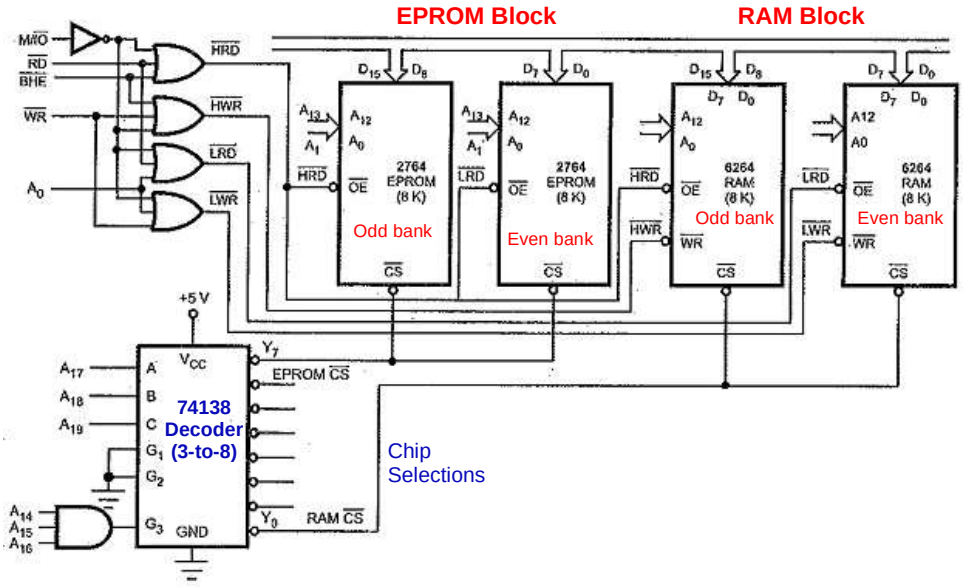
- In the following example, total of 32 KB memory system is designed.
- Two EEPROM chips are used (2764 chips).
- Each EPROM is 8Kx8 bit.
- Total of EPROM block = 16 KB

- Also two RAM chips are used (6264 chips).
- Each RAM is 8Kx8 bit.
- Total of RAM block = 16 KB

- Calculation of number of address lines needed for location selection:
  log2(8K) = log2 (2^{13}) = 13 lines.

Example: Decoding with Address Decoder chip

- Address line A0 is used for Low(Even) bank selection.
- $\overline{BHE}$ signal is used for High(Odd) bank selection logic.
- Address lines A1 thru A13 are used for location selection within a memory chip.
- Address lines A14-A15-A16 are used with AND gate, then as G3 enable input of the Decoder chip.
- Address lines A17-A18-A19 are used for A-B-C inputs of the Decoder chip.
- Y0 output Decoder is used for memory chip selection of EPROM Block.
- Y7 output Decoder is used for memory chip selection of RAM Block.



Memory Interfacing with 8086 CPU

1) Connect the Address latches and Data Buffers to CPU Address/Data bus and obtain the demultiplexed (separated) Address bus lines and buffered Data bus lines.

2) Arrange the available memory chips so as to obtain 16-bit data bus width.

3) The upper 8-bit bank is called “Odd address memory bank”.
4) The lower 8-bit bank is called “Even address memory bank”.

5) Connect available memory address lines (up to 20 bit) of memory chips with the demultiplexed address lines (20 bit).

6) Connect the RD and WR inputs of memory chips to the corresponding processor control signals.

7) Connect the 16-bit data bus of memory bank with the buffered Data bus lines (16 bit).

8) The remaining address lines of the CPU, $\overline{BHE}$ and A0 are used for Decoding the required chip select signals for the odd and even memory banks.

9) The CS (Chip Select) input of memory chips is derived from the output of the Decoding circuit (either Logic gates, or a Decoder chip).

Example1 : Memory Design

QUESTION:

- Design and draw the block diagram of a memory system for 8086 CPU based computer, with the following memory chips.
  - Use two 4Kx8 EPROM chips (Total 8 KB).
  - Use two 4Kx8 RAM chips (Total 8 KB).
  - The overall total memory will be 16 KB.

- Also write the suitable memory maps, showing the smallest addresses and biggets addresses for each memory chip (high and low banks).

SOLUTION:

- In a 8086 CPU based computer, the address **FFFF0** must lie in the EPROM always. 8086 initializes its Instruction Pointer (IP) register with the value from the reset address FFFF0. (If the FFFF0 address does not exist, the IP register is initialized to 0.)

- The address of RAM may be selected anywhere in the 1MB address space of 8086 CPU.
- In general, the RAM address map should be organized as continuous without any gaps.

Solution

- Each memory chip (EPROMs and RAMs) is 4Kx8.
- Calculation of number of address lines needed for location selection within a memory chip:
  $\log_2(4K) = \log_2(2^{12}) = 12$ address lines.

- $\overline{BHE}$ signal is connected to the A0 pin of Decoder.
- A0 address line is connected to the A1 pin of Decoder.
- A13 address line is connected to the A2 pin of Decoder.

- Address lines A1 - A12 are used for location selection within memory chips.

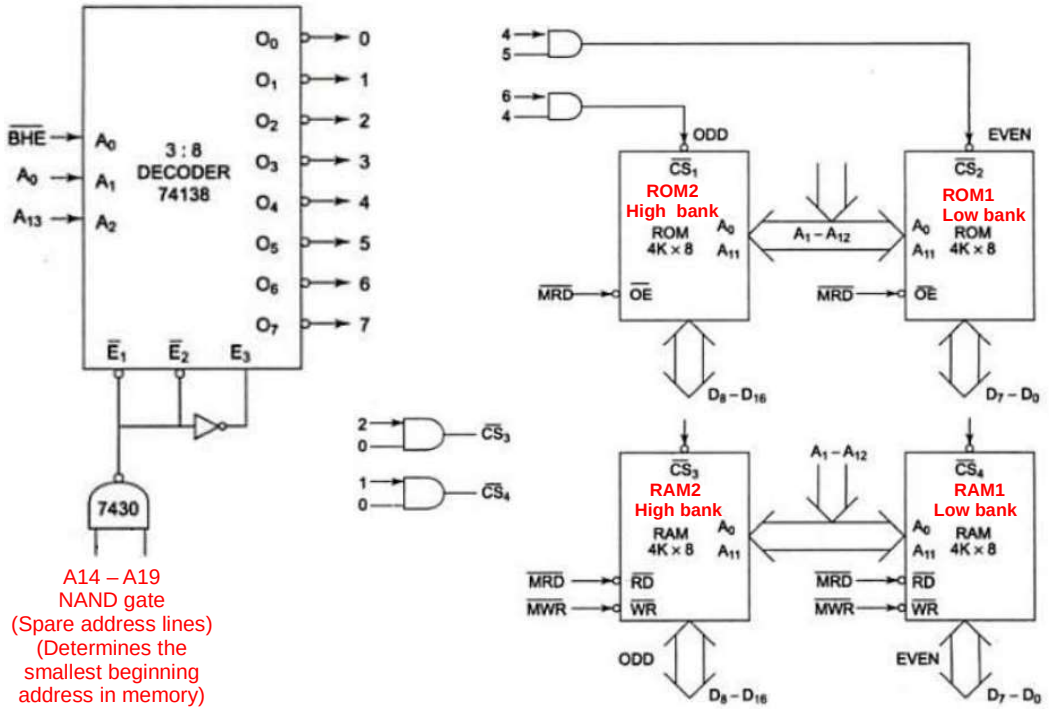
- Address lines A14 - A19 (Spare lines) are used as inputs of AND gate (7430 logic AND chip).

- The output of the AND gate is used as inputs of $\overline{E1}$, $\overline{E2}$, E3 pins (Decoder enable pins) of Decoder.

- Outputs of the Decoder are used with AND gates, so that the memory chip selection signals are obtained.
  $\overline{CS1}$: Chip select of ROM2 (Odd bank)
  $\overline{CS2}$: Chip select of ROM1 (Even bank)
  $\overline{CS3}$: Chip select of RAM2 (Odd bank)
  $\overline{CS4}$: Chip select of RAM1 (Even bank)

- Total of four 4K x 8 memory chips are used.
- Two 4Kx8 RAM chips, and two 4Kx8 ROM chips are arranged in parallel to obtain 16-bit data bus width.

Solution



Solution

Memory Chip Selections Table

Decoder Input Pins --->	A2	A1	A0	Selected Memory Chip and High/Odd Bank Address
CPU Output Pins --->	A13	A0	$\overline{BHE}$	
Word transfer (D0 - D15)	0	0	0	Even and Odd address in RAM
Byte transfer (D0 - D7)	0	0	1	Only Even address in RAM
Byte transfer (D8 - D15)	0	1	0	Only Odd address in RAM
Word transfer (D0 - D15)	1	0	0	Even and Odd address in ROM
Byte transfer (D0 - D7)	1	0	1	Only Even address in ROM
Byte transfer (D8 - D15)	1	1	0	Only Odd address in ROM

Solution

Memory Map Table

Memory Block	Address (Hexadecimal)		A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
RAM1 4Kx8 (Even Bank)	Smallest	**FC000**	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Biggest	**FDFFE**	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0
RAM2 4Kx8 (Odd Bank)	Smallest	**FC001**	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	Biggest	**FDFFF**	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
EEPROM1 4Kx8 (Even Bank)	Smallest	**FE000**	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Biggest	**FFFFE**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
EEPROM2 4Kx8 (Odd Bank)	Smallest	**FE001**	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
	Biggest	**FFFFF**	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

A14 - A19 are used as NAND gate inputs (Then for Decoder enabling)

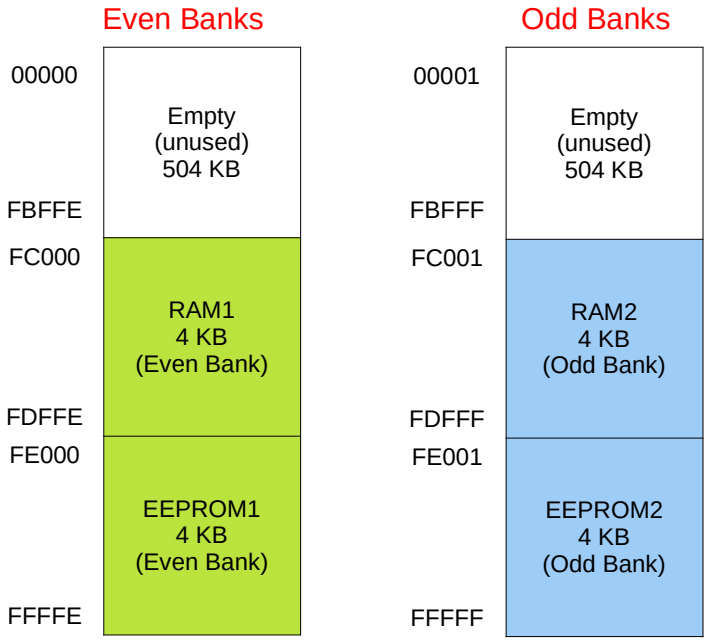
A13 is used as Decoder input

A1 - A12 are used for location selection within memory chips

A0 is used as Decoder input

Physical Memory Map Diagram

Overall total used memory = 16 KB
Addressable memory capacity = 1 MB
Empty (unused) space = 1 MB – 16 KB
= 1024 KB – 16 KB
= 1008 KB



Example2 : Memory Design

QUESTION: Design of a memory system for 8086 CPU with the following requirements.
Total 128 KB EPROM memory : By using 32Kx8 chips
Total 64 KB SRAM memory : By using 16Kx8 chips
Overall total memory = 128 KB + 64 KB = 192 KB

- Calculate how many chips are required for EPROM and for SRAM memory.
- Calculate minimum number of address lines required to support the memory chips.

- Draw the following block diagrams, showing all necessary connection details.

  Part1: Draw a block diagram with 8086 CPU, Address Latches, Data Transceiver Buffers, Control Signals Decoder, Address Bus, Data Bus, and Control signal connections.

  Part2: Draw a separate block diagram with Address Decoder, Memory chips, and their connections.

- Two separate Decoders will be used for different purposes.
  - In Part1, a Decoder will be used as Control Signals Decoder.
  - In Part2, another Decoder will be used as Address Decoder.

- Also write the Memory Map table, showing the smallest addresses and biggest addresses for each memory chip (high and low banks).

Solution

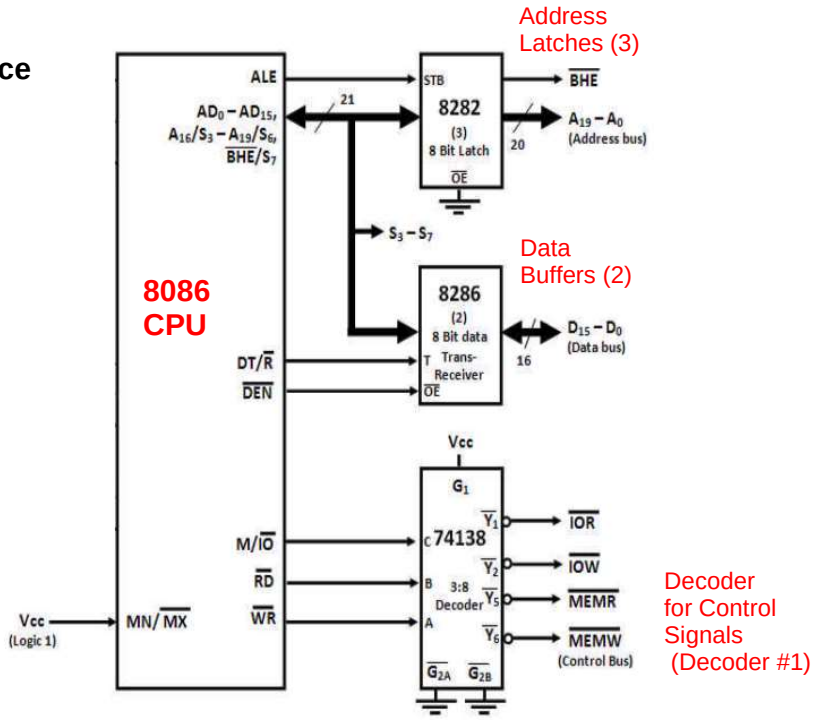
Calculation of number of address lines required for location selection:

EPROMs:
Total Required Memory = 128 KB
Available Memory chips = 32Kx8
Number of chips required = 128 / 32 = 4 chips
Number of Address Lines required = 15 lines (A15 – A1)
$\log_2 (32K) = \log_2 (2^{15}) = 15$ lines

RAMs:
Total Required Memory = 64 KB
Available Memory chips = 16Kx8
Number of chips required = 64 / 16 = 4 chips
Number of Address Lines required = 14 lines (A14 – A1)
$\log_2 (16K) = \log_2 (2^{14}) = 14$ lines

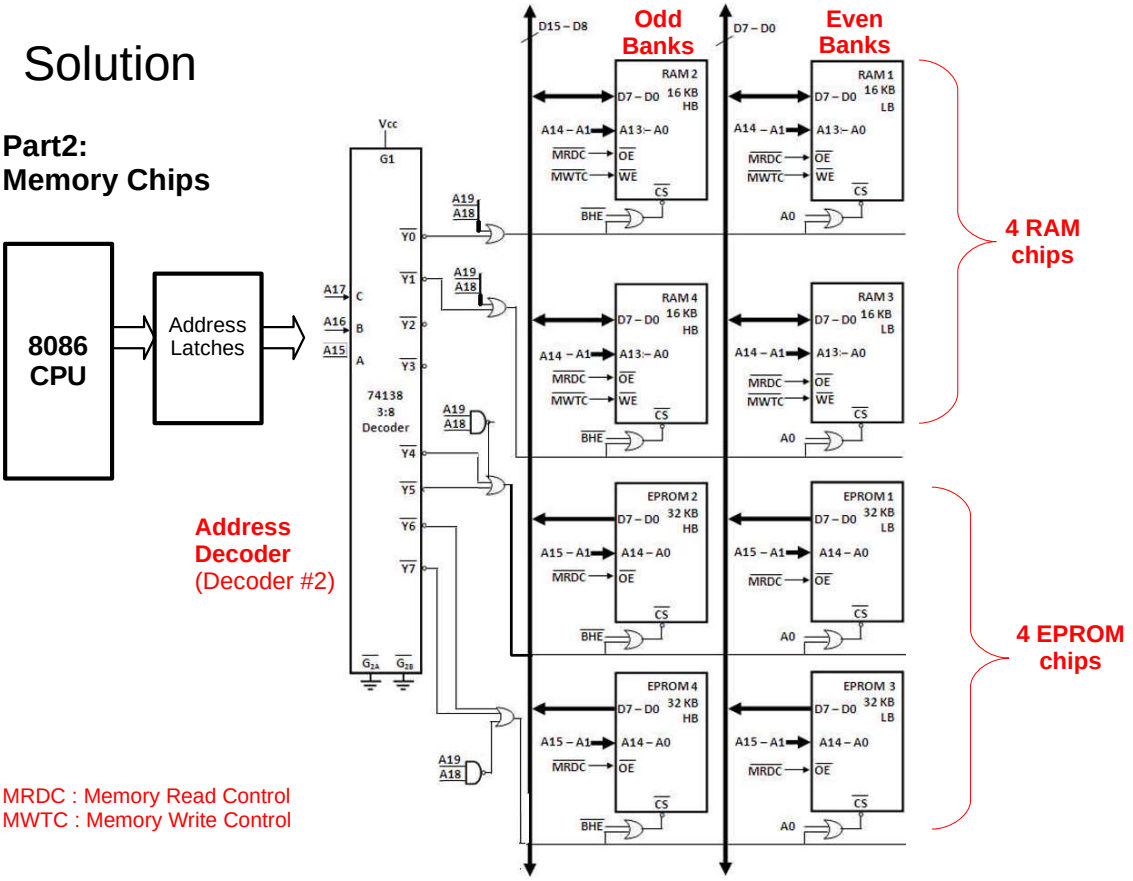
Solution

Part1:
Bus Interface



Solution

Part2:
Memory Chips



MRDC : Memory Read Control
MWTC : Memory Write Control

Solution

RAM Map Table

LB: Low Bank, HB: High Bank

Memory Chip	Chip Select					Address Lines														Memory Address
	A19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
RAM 1 (LB)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	00000H
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	07FFEH
RAM 2 (HB)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	00001H
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	07FFFH
RAM 3 (LB)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	08000H
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0FFFEH
RAM 4 (HB)	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	08001H
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0FFFFH

ROM Map Table

Memory Chip	Chip Select					Address Lines														Memory Address
	A19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	A0	
ROM 1 (LB)	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	E0000H
	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	EFFFEH
ROM 2 (HB)	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	E0001H
	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	EFFFFH
ROM 3 (LB)	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	F0000H
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	FFFFEH
ROM 4 (HB)	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	F0001H
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH