CS-235: Computer Organization & Assembly Language

Intel Memory Architecture

Topic # 8

Fall 2020

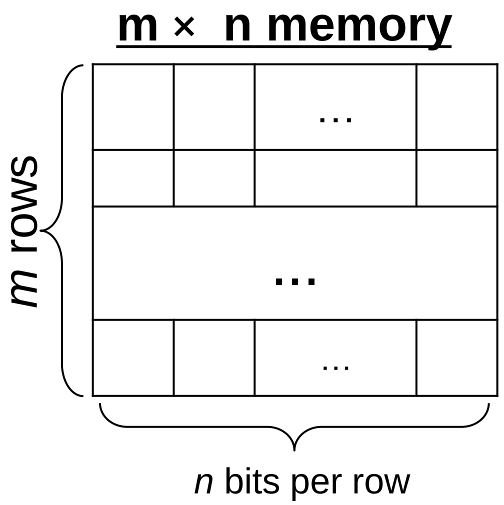
Muhammad Imran Abeel <imran.abeel@seecs.edu.pk>

Book Reference

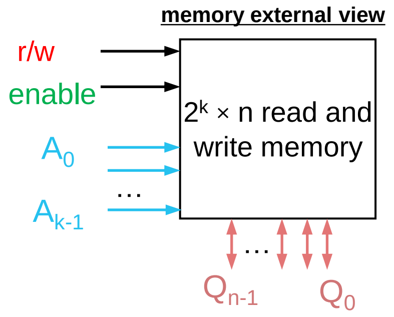
- Intel Microprocessor by Barry B. Brey (8th Edition)
- Chapter 10

Basics of Memory Architecture

- **Stores large number of bits**
  - *m* x *n*: *m* rows of *n* bits each
  - k = Log$_2$(*m*) address input signals
  - or $m = 2^k$ rows
  - e.g., 4k x 8 memory:
    12 address inputs
    8 data lines



- Memory access
  - r/w: selects read or write
  - enable: read or write only when asserted



[3]

Memory Types

- **Traditional ROM/RAM distinctions**
  - ROM
    read only, bits stored without power
  - RAM
    read and write, lose stored bits without power
- **Traditional distinctions blurred**
  - Advanced ROMs can be written to
    e.g., EEPROM
  - Advanced RAMs can hold bits without power
    e.g., NVRAM, DDRAM
- **Write Ability**
  - Speed, a memory can be written
- **Storage Permanence (Reliability)**
  - ability of memory to hold stored bits after they are written

Write Ability

- **Ranges of Write Ability**

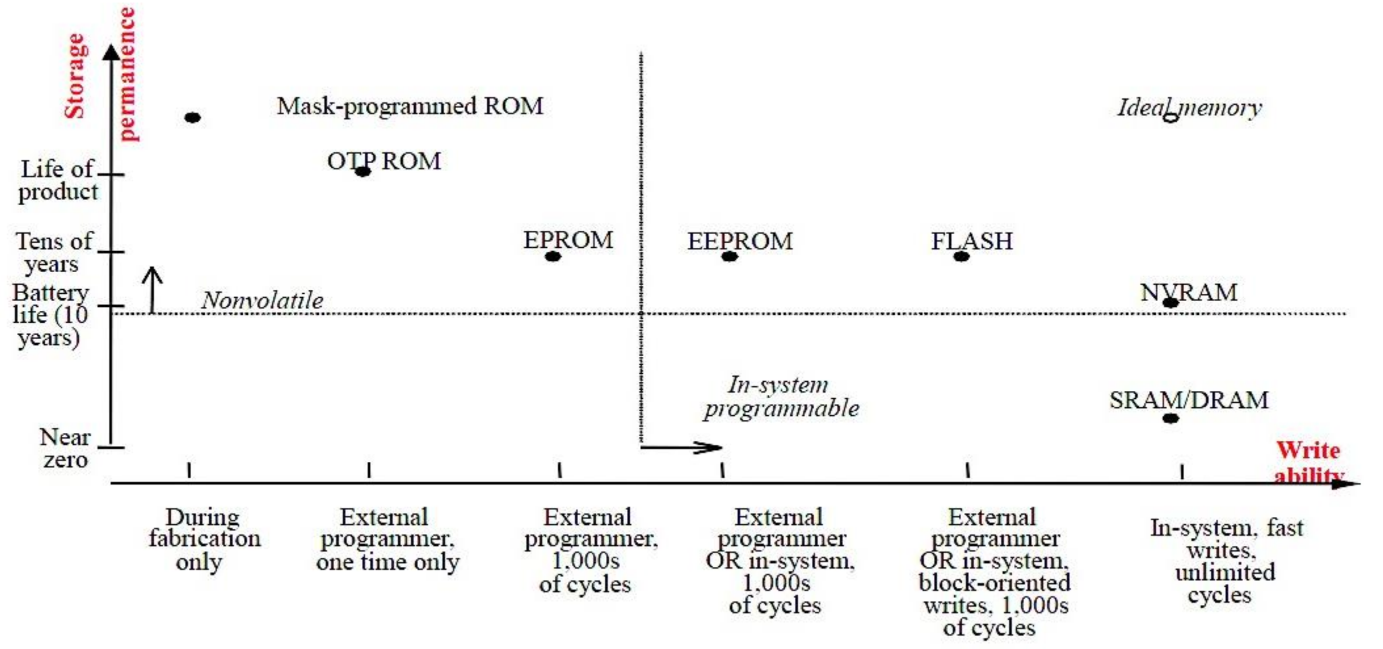
  - High End
    processor writes to memory simply and quickly
    e.g., RAM
  - Middle Range
    processor writes to memory, but slower
    e.g., FLASH, EEPROM
  - Lower Range
    special equipment, “programmer”, must be used to write to memory
    e.g., EPROM, OTP ROM
  - Low End
    bits stored only during fabrication
    e.g., Mask-programmed ROM

Performance (Reliability)

- **Range of Storage Permanence**
  - High End
    essentially never loses bits
    e.g., mask-programmed ROM
  - Middle Range
    holds bits days, months, or years after memory's power source turned off
    e.g., NVRAM
  - Lower Range
    holds bits as long as power supplied to memory
    e.g., SRAM
  - Low End
    begins to lose bits almost immediately after written
    e.g., DRAM

[6]

Performance


Storage permanence
Mask-programmed ROM
Ideal memory
Life of product
OTP ROM
Tens of years
EPROM
EEPROM
FLASH
Battery life (10 years)
Nonvolatile
NVRAM
In-system programmable
SRAM/DRAM
Near zero
Write ability
During fabrication only
External programmer, one time only
External programmer, 1,000s of cycles
External programmer OR in-system, 1,000s of cycles
External programmer OR in-system, block-oriented writes, 1,000s of cycles
In-system, fast writes, unlimited cycles

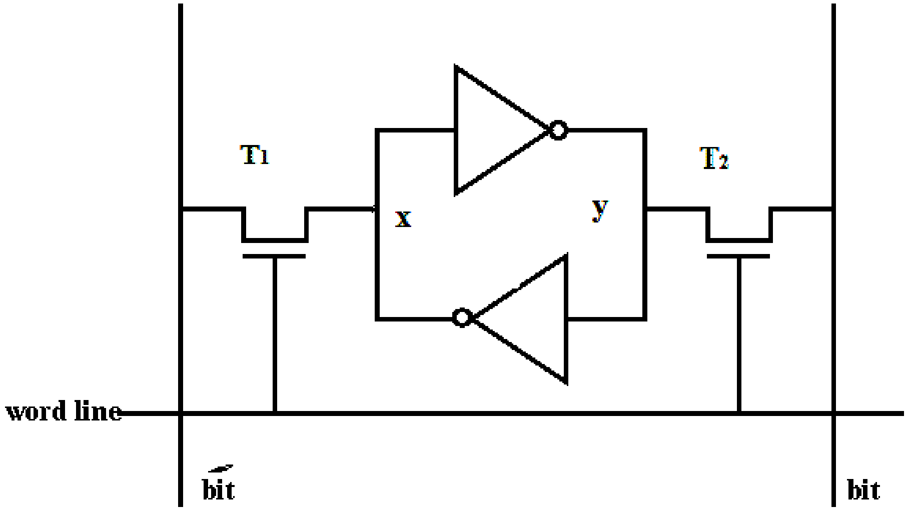
Write ability and storage permanence of memories, showing relative degrees along each axis (not to scale).

[7]

Basic Types of RAM

- SRAM: Static RAM
  - Memory cell uses Flip-Flop to store a Bit
  - Requires 6 Transistors
  - Holds Data as long as power supplied

<u>Memory Cell Internals</u>



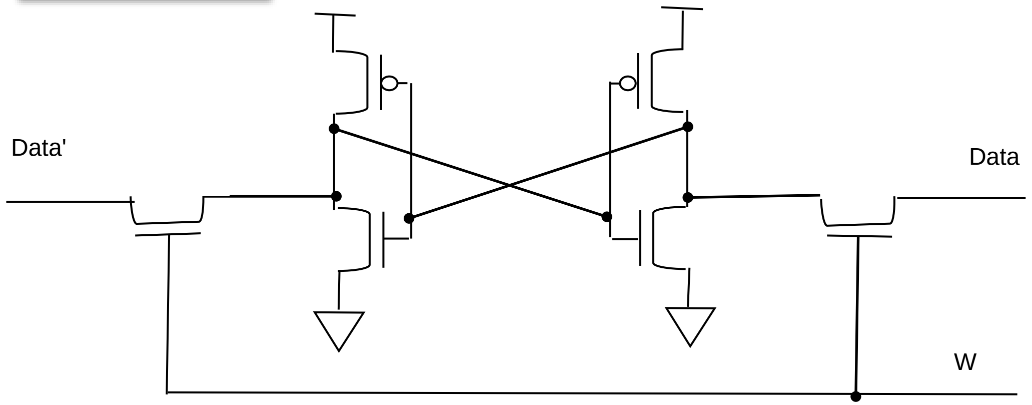
[8]

Basic Types of RAM

- SRAM: Static RAM
  - Memory cell uses Flip-Flop to store a Bit
  - Requires 6 Transistors
  - Holds Data as long as power supplied

SRAM

Memory Cell Internals



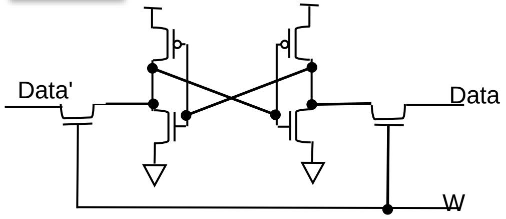
[9]

Basic Types of RAM

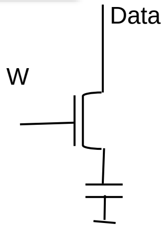
- SRAM: Static RAM
  - Memory cell uses Flip-Flop to store a Bit
  - Requires 6 Transistors
  - Holds Data as long as power supplied
- DRAM: Dynamic RAM
  - Memory cell uses MOS transistor and capacitor to store a bit
  - More compact than SRAM
  - “Refresh” required due to capacitor leakage
  - word’s cells refreshed when read
  - Typical refresh rate 15.625 micro sec
  - **Slower to access than SRAM**

Memory Cell Internals

SRAM

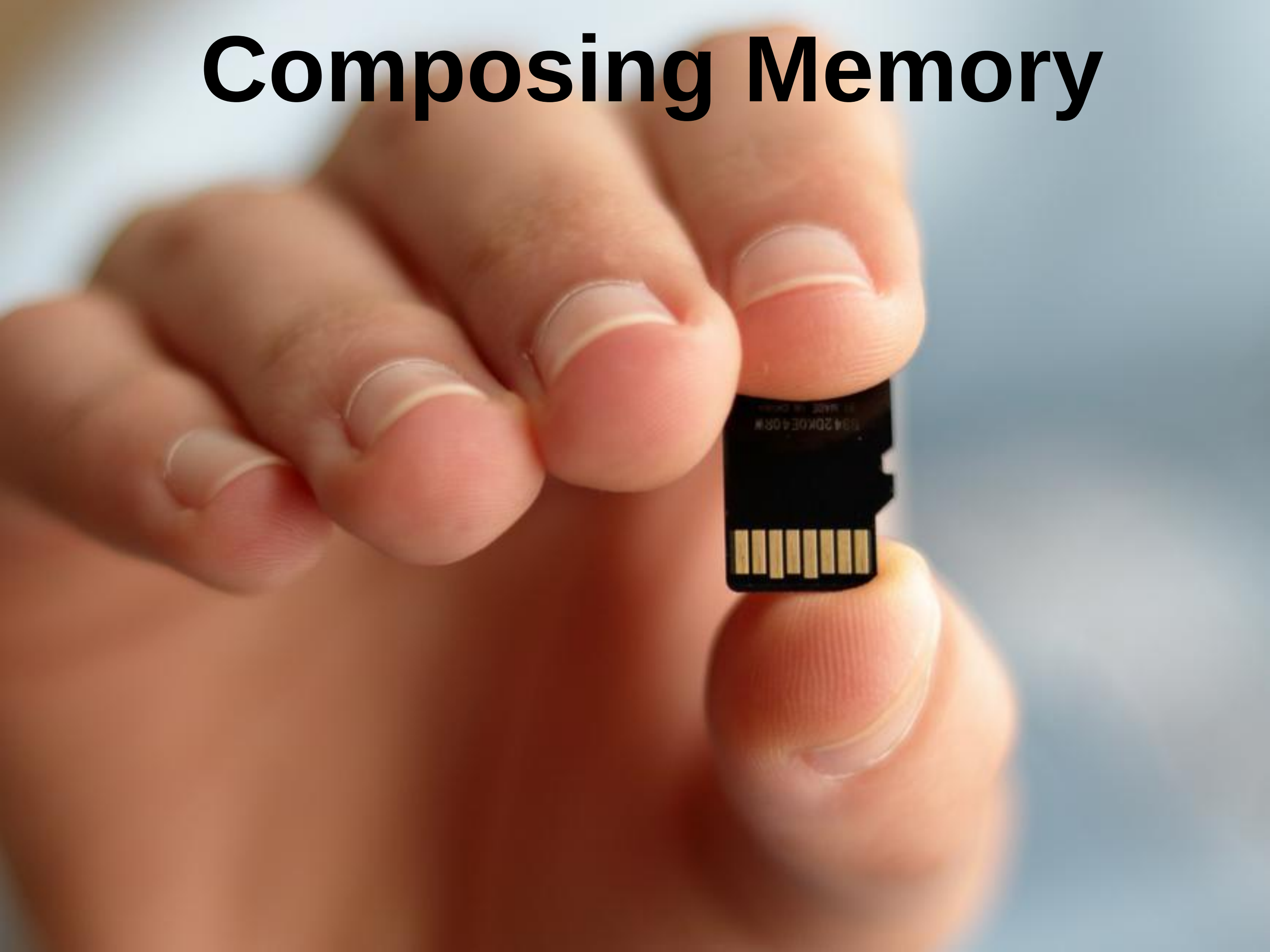


DRAM



[10]

Composing Memory



Composing Memory

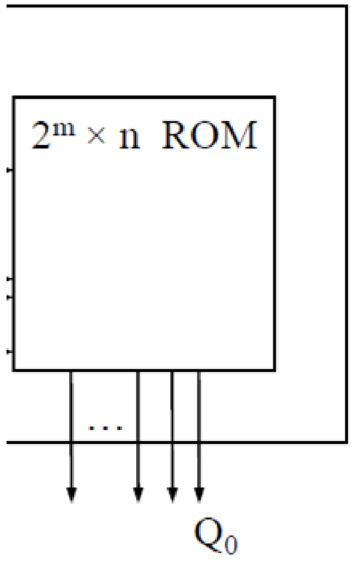
- **Memory Size needed often differs from size of readily available Memories**
  - **Required size 4K x 16 but available size is 2K x 8**

- When available Memory is larger
  - simply ignore unneeded high-order address bits and higher data lines

- When available Memory is smaller
  - compose several smaller memories into one larger memory

Composing Memory

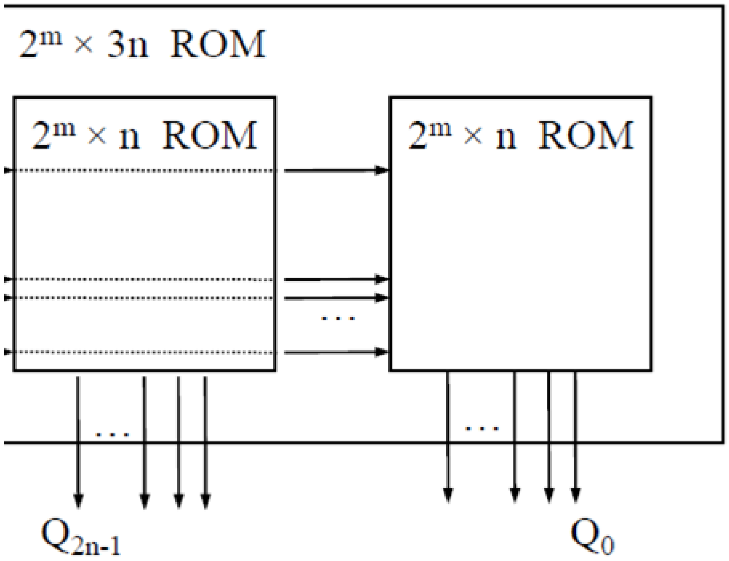
- Connect side-by-side to increase width of words



[13]

Composing Memory

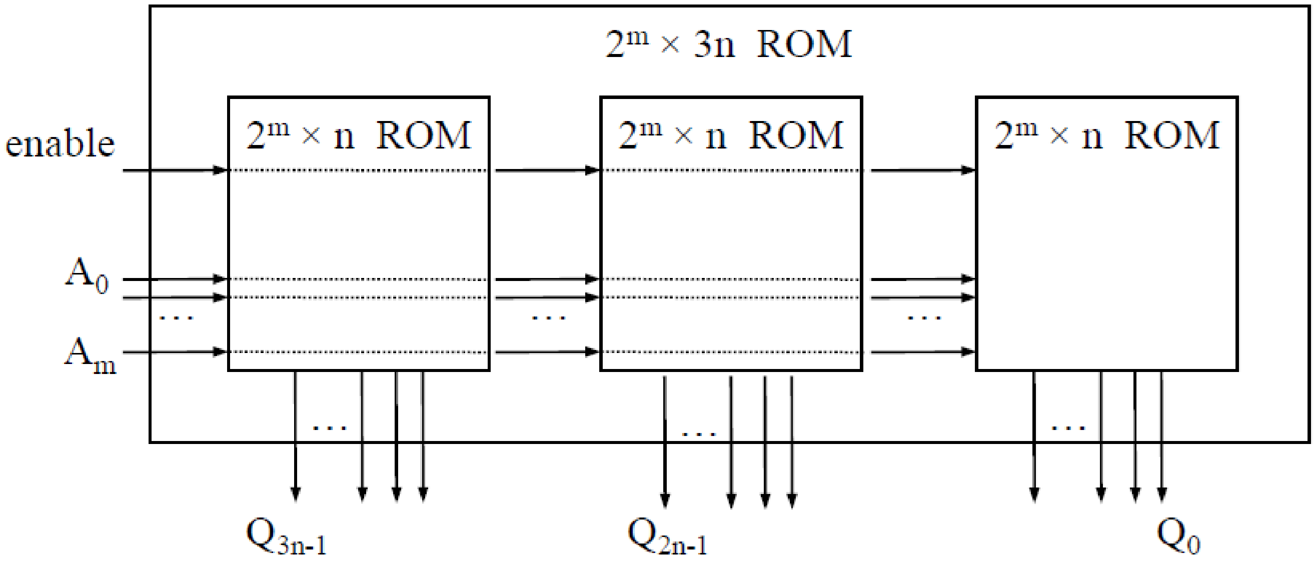
- Connect side-by-side to increase width of words



[14]

Composing Memory

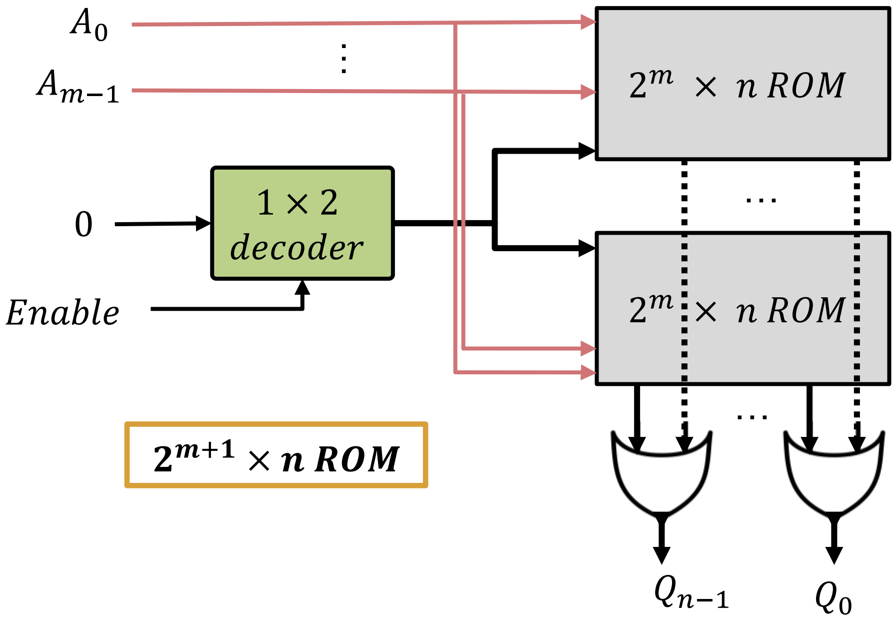
- Connect side-by-side to increase width of words



[15]

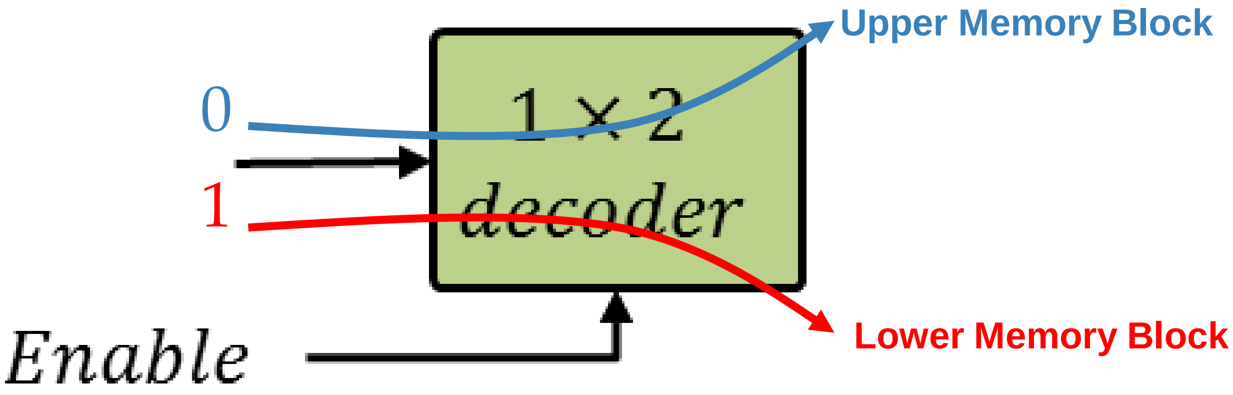
Composing Memory

- Connect top to bottom to increase number of words
  - Added high-order address line selects smaller memory containing desired word using a decoder



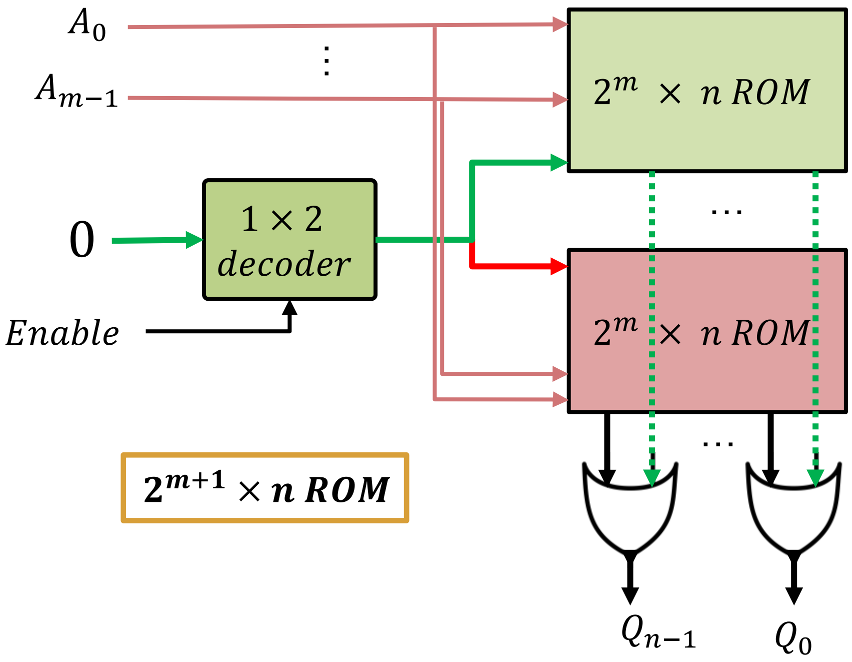
Composing Memory: Decoder

- Connect top to bottom to increase number of words
  - Added high-order address line selects smaller memory containing desired word using a decoder



Composing Memory

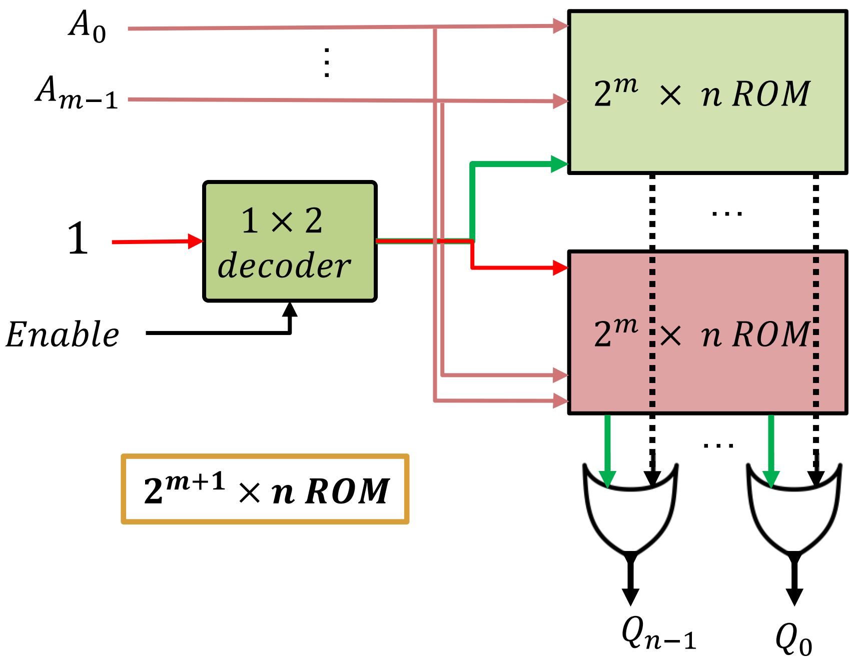
- Connect top to bottom to increase number of words
  - Added high-order address line selects smaller memory containing desired word using a decoder



[18]

Composing Memory

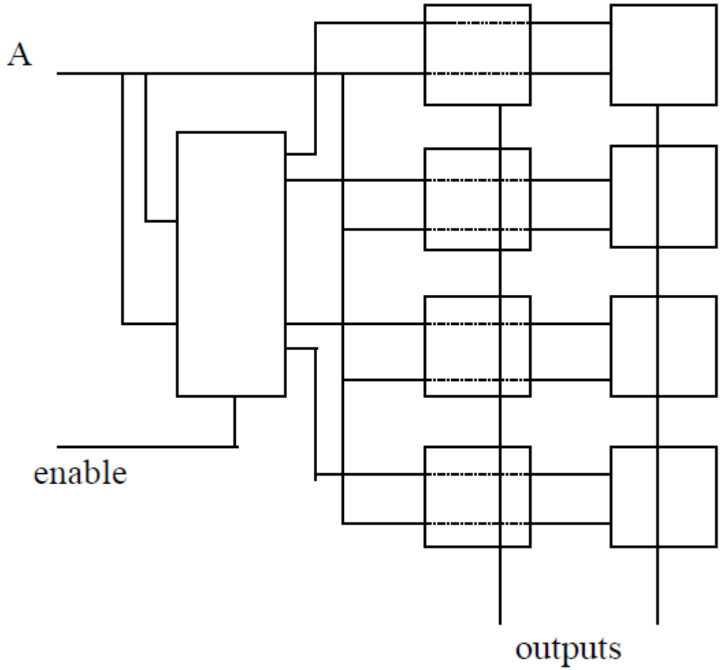
- Connect top to bottom to increase number of words
  - Added high-order address line selects smaller memory containing desired word using a decoder



[19]

Composing Memory

- Combine techniques to increase number and width of words

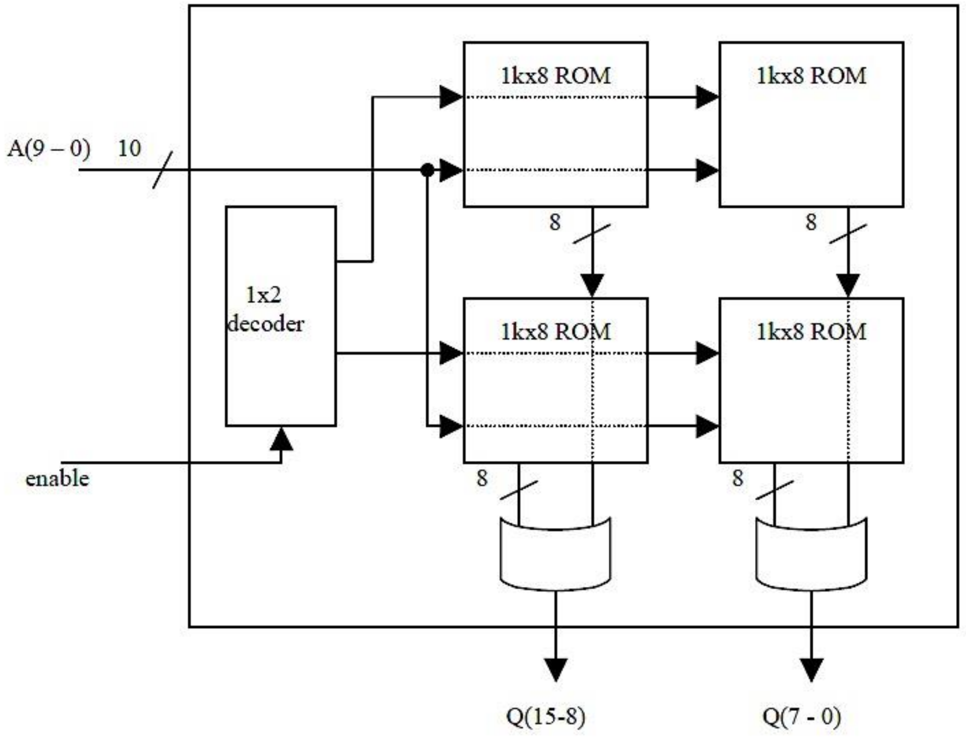


Activity

- Compose 1K x 8 ROM into a 2K x 16 ROM

Activity

- Compose 1K x 8 ROM into a 2K x 16 ROM

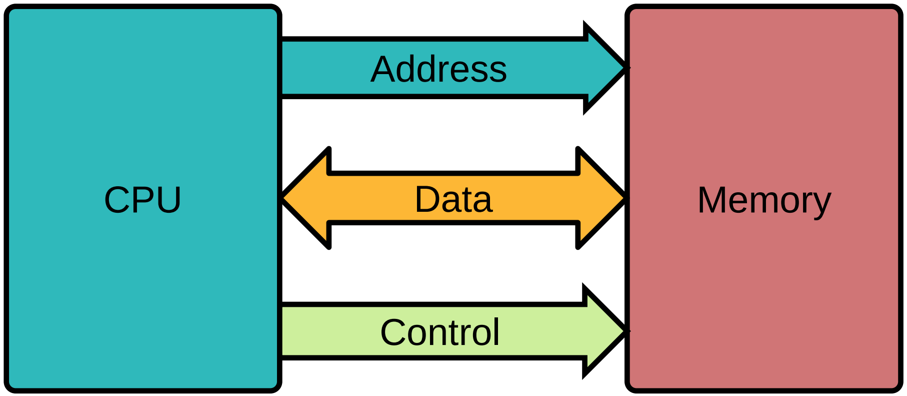


[22]

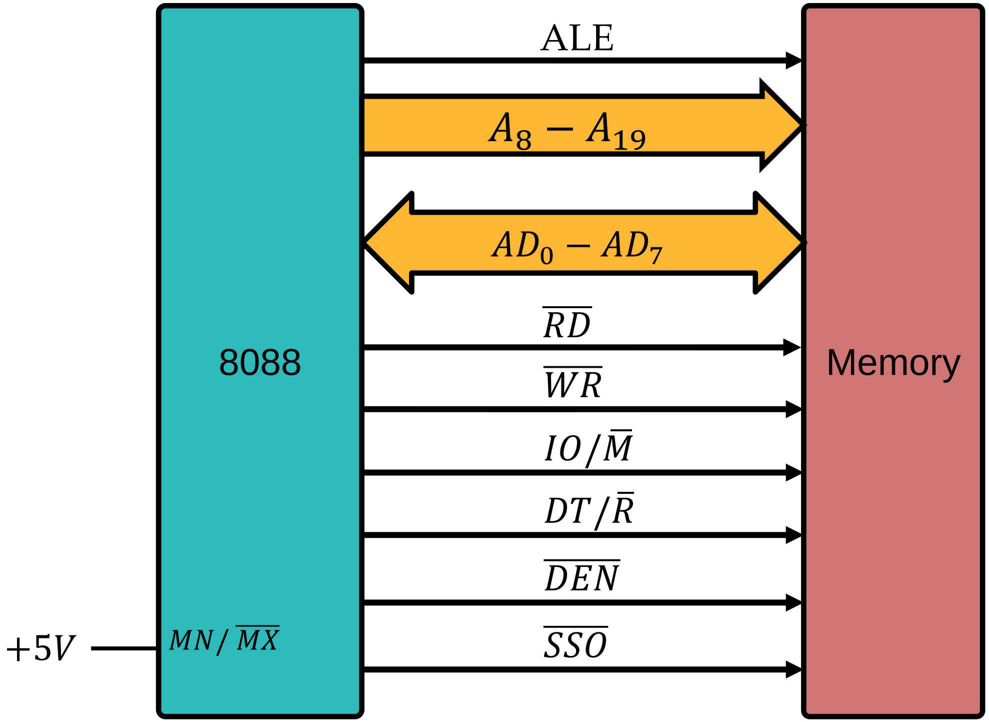
Interfacing Signals

- Interfacing memory with CPU
  - Address Lines
  - Data Lines
  - Control Lines

Enable, Read, Write, Ready, Size etc.



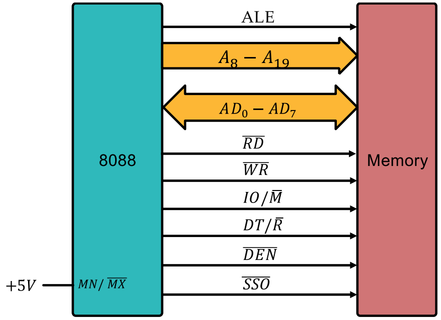
8088 Minimum Memory Interface



Minimum-mode 8088 memory interface

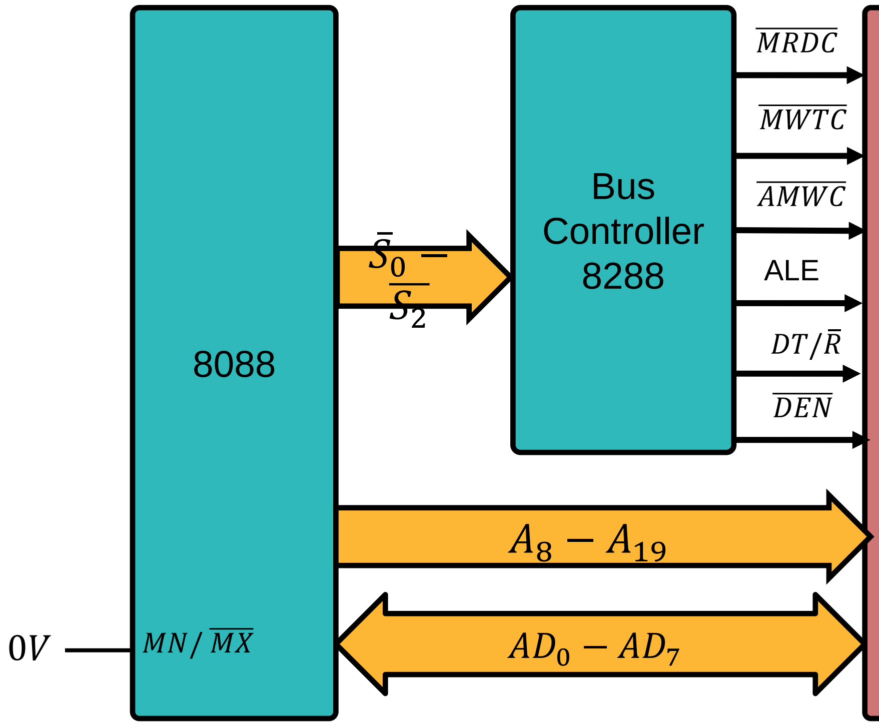
Memory Control Signals

ALE	**Address Latch Enable** : used to latch the address in external memory
$IO/\overline{M}$	**Input-output/Memory** : signal external circuity whether memory or I/O bus cycle in progress
$DT/\overline{R}$	**Data Transmit/Receive** : signal external circuity whether 8088 is transmitting or receiving data over the bus.
$\overline{WR}$	**Write :** identifies a write cycle in progress
$\overline{RD}$	**Read :** identifies a read cycle in progress
$\overline{DEN}$	**Data Enable :** used to enable data bus.
$\overline{SSO}$	**Status Line :** identifies whether a code or data access is in progress



Minimum-mode 8088 memory interface

8088 : Maximum Memory Interface



Maximum-mode 8088 memory interface

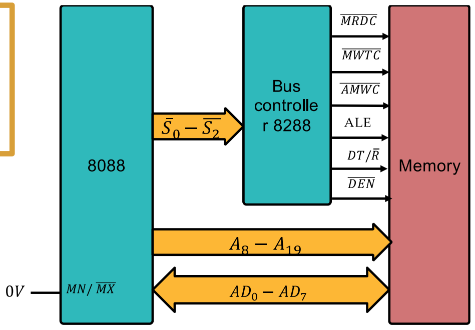
Memory Control Signals

- Maximum Mode Memory Control Signals

$\overline{\text{MRDC}}$ – Memory Read Command

$\overline{\text{MWTC}}$ – Memory Write Command

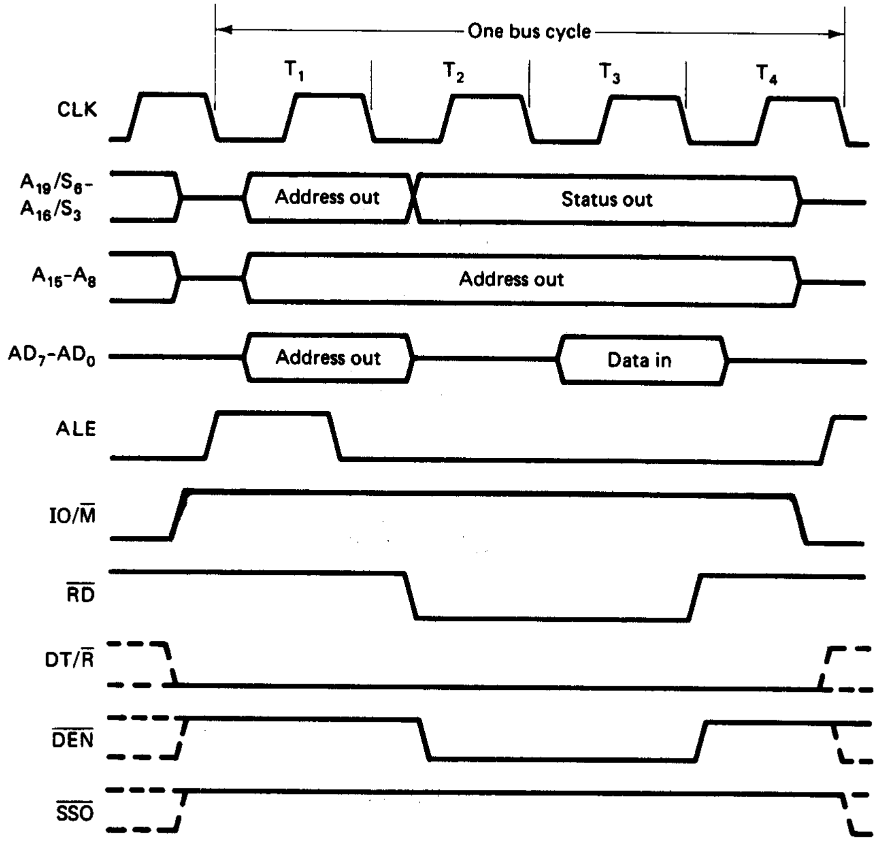
$\overline{\text{AMWC}}$ – Advanced Memory Write Command



Maximum-mode 8088 memory interface

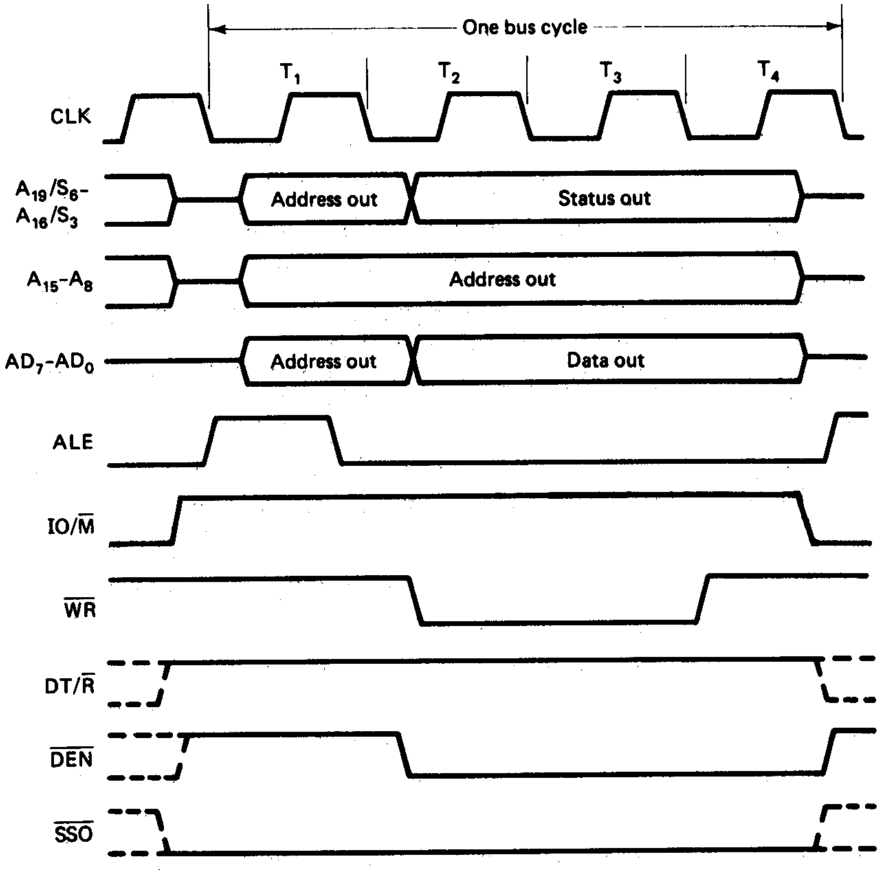
Status Inputs			CPU Cycle	8288 Command
$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$		
0	0	0	Interrupt Acknowledge	$\overline{\text{INTA}}$
0	0	1	Read I/O Port	$\overline{\text{IORC}}$
0	1	0	Write I/O Port	$\overline{\text{IOWC}}$, $\overline{\text{AIOWC}}$
0	1	1	Halt	None
1	0	0	Instruction Fetch	$\overline{\text{MRDC}}$
1	0	1	Read Memory	$\overline{\text{MRDC}}$
1	1	0	Write Memory	$\overline{\text{MWTC}}$, $\overline{\text{AMWC}}$
1	1	1	Passive	None

8088 Memory Read Cycle (min. mode)



[28]

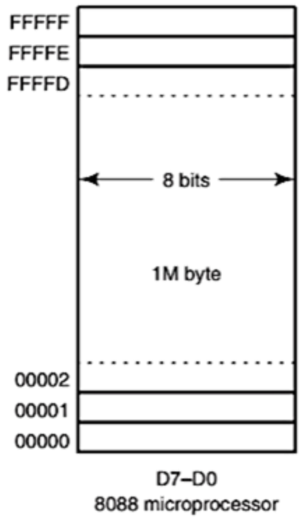
8088 Memory Write Cycle (min. mode)

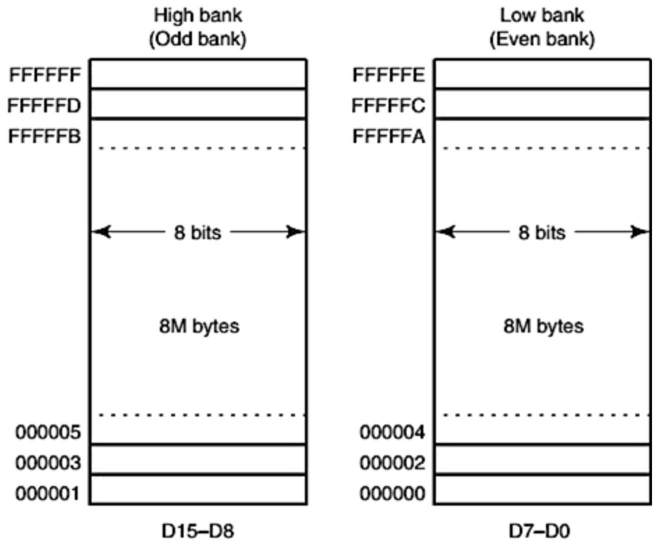


Reading Assignment

- Maximum Mode Read and Write Memory Bus cycles

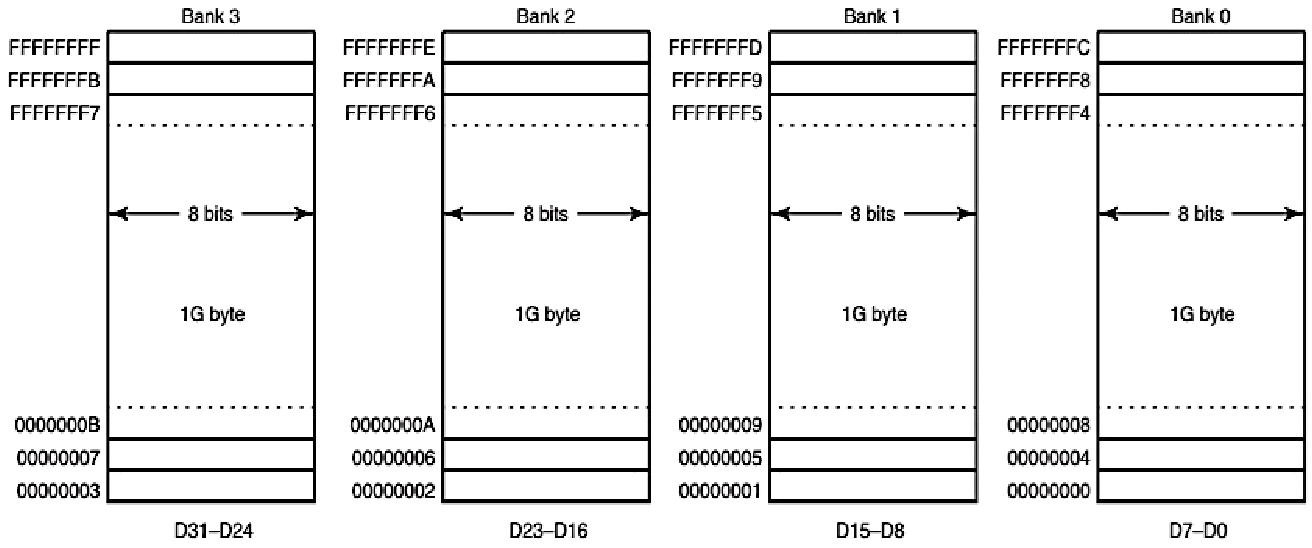
Hardware Organization





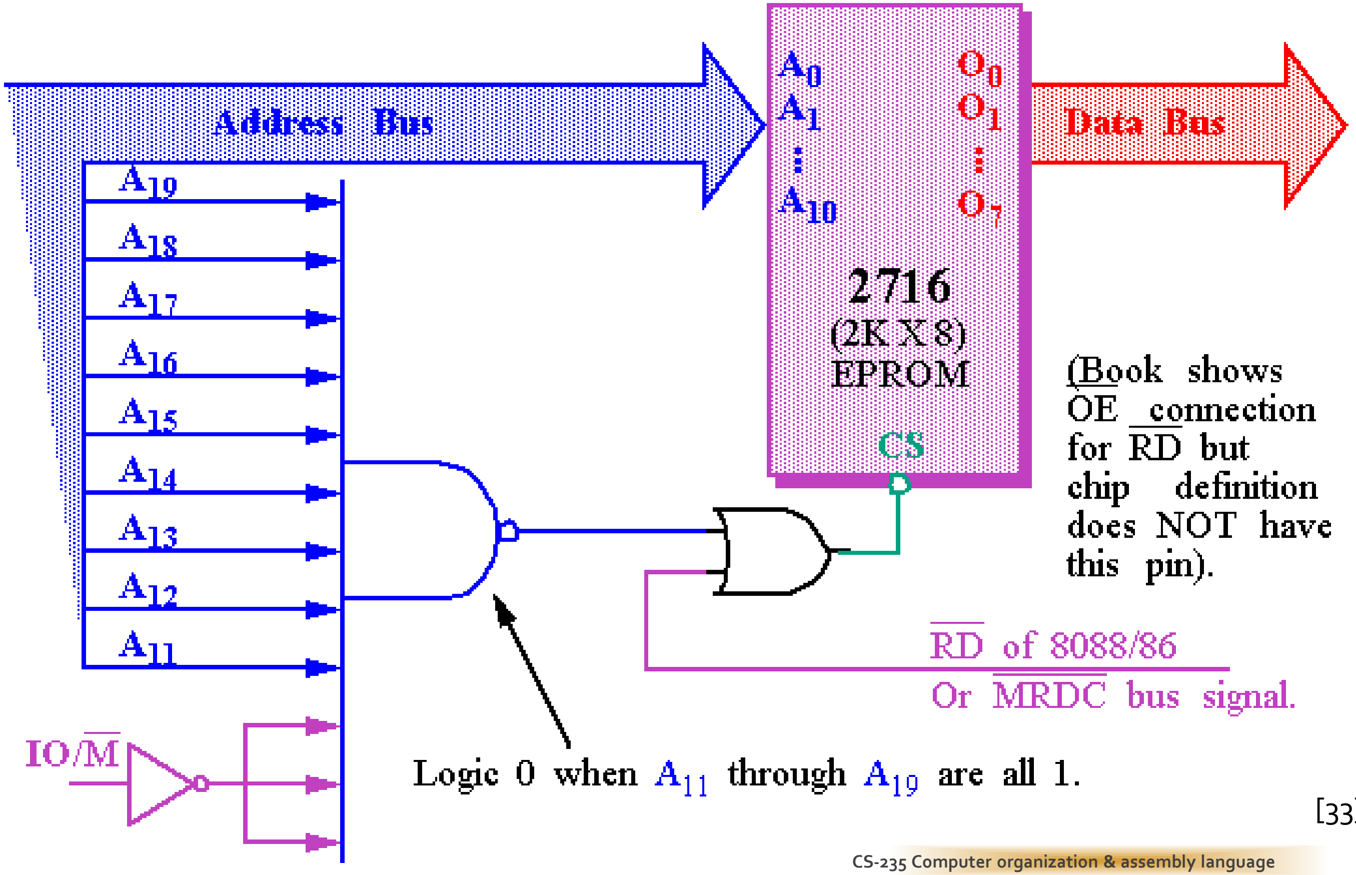
8086 microprocessor (memory is only 1M bytes)
80286 microprocessor
80386SX microprocessor
80386SL microprocessor (memory is 32M bytes)
80386SLC microprocessor (memory is 32M bytes)

Hardware Organization



80386DX microprocessor
80486SX microprocessor
80486DX microprocessor

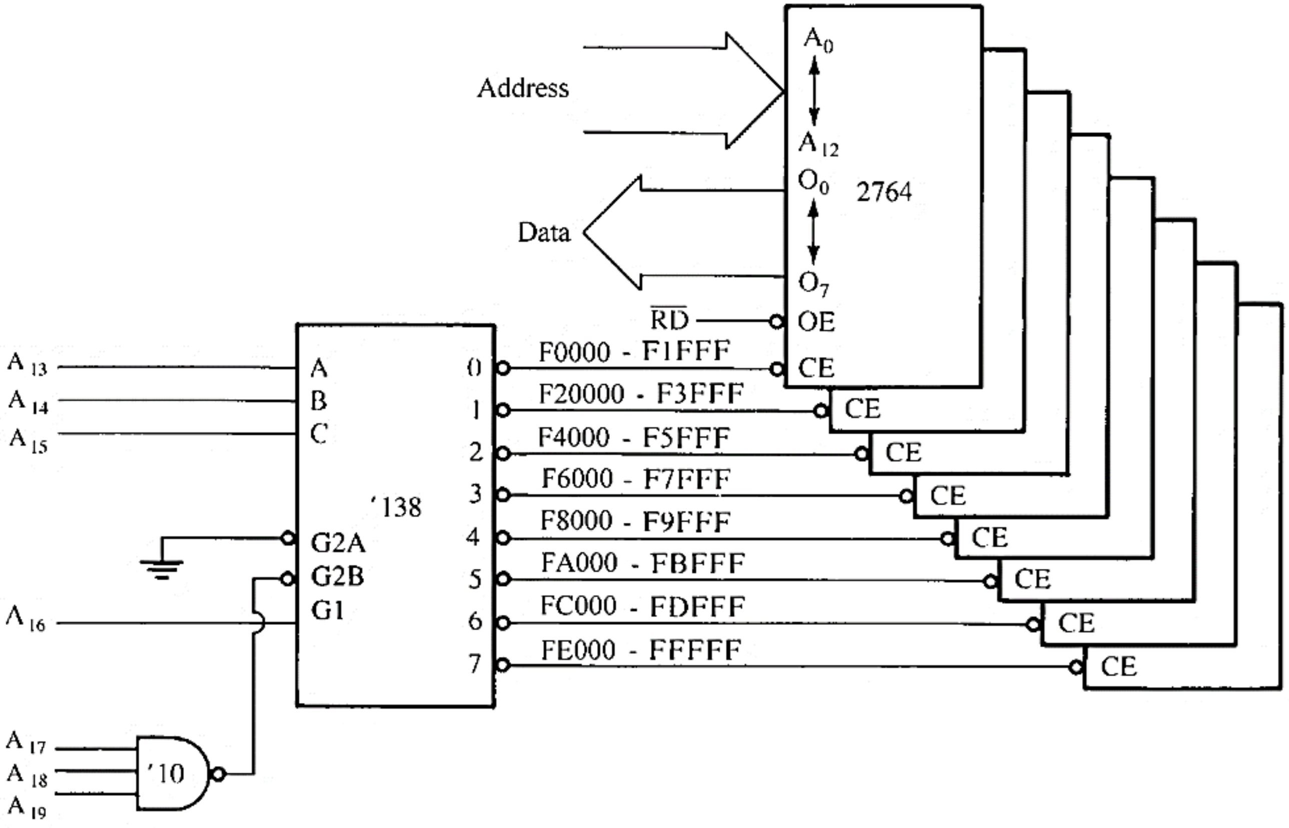
Memory Address Decoding : NAND



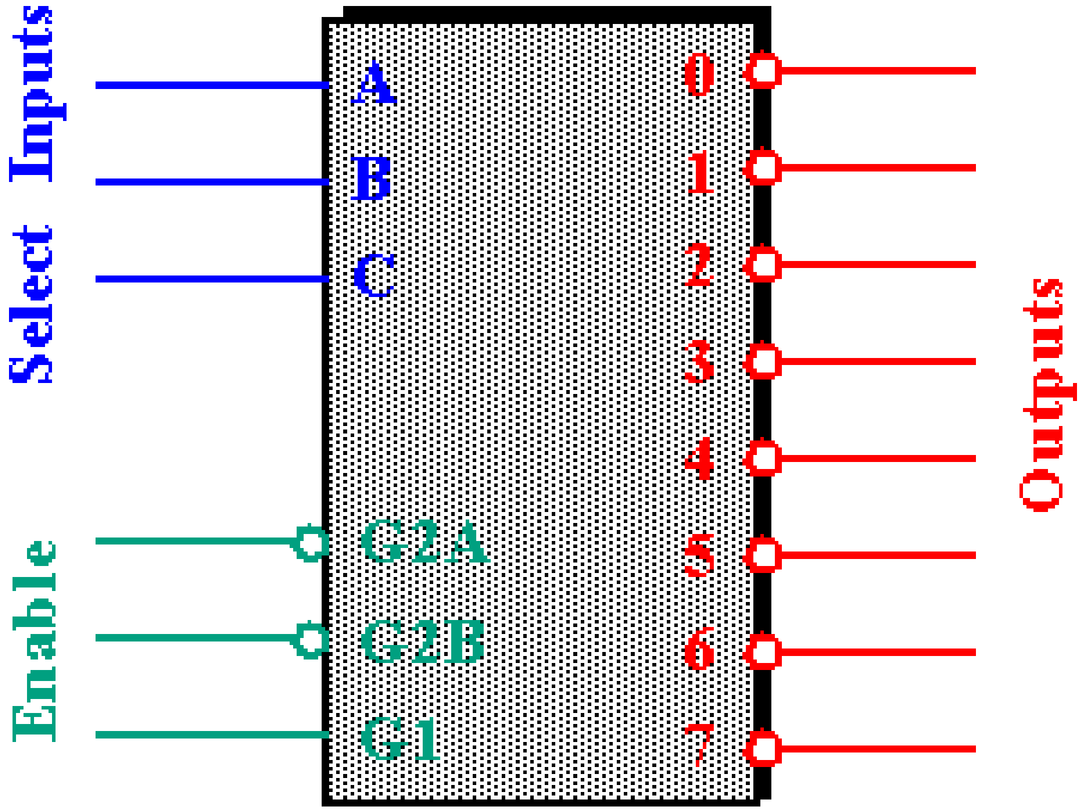
(Book shows $\overline{OE}$ connection for $\overline{RD}$ but chip definition does NOT have this pin).

Logic 0 when A_{11} through A_{19} are all 1.

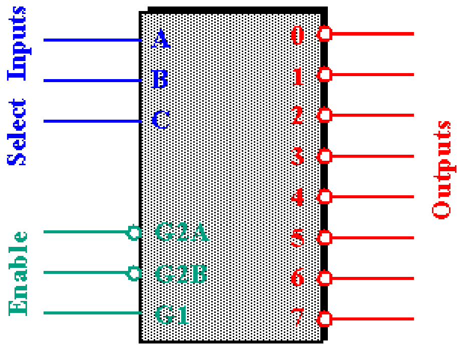
Memory Address Decoding : 3 to 8



Memory Address Decoding : 3 to 8



Memory Address Decoding : 3 to 8

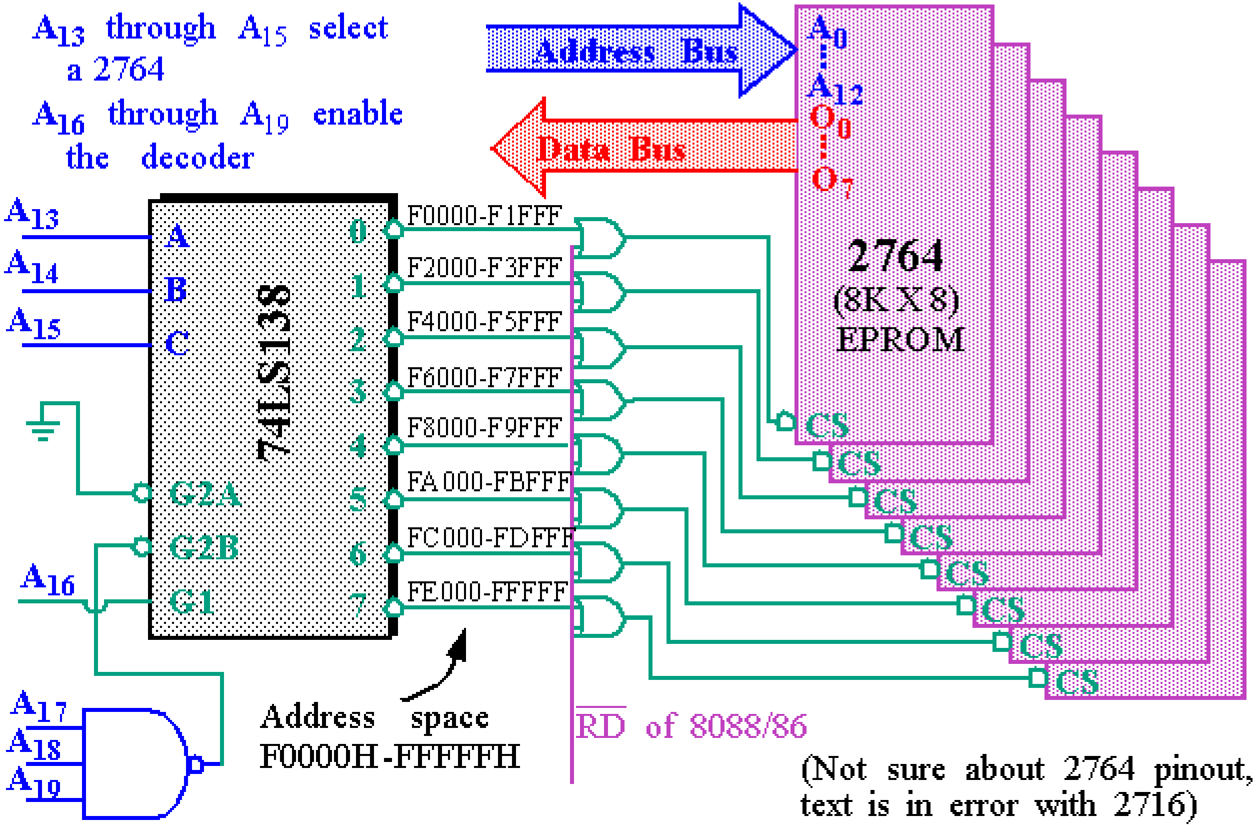


Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1
0	0	1	0	0	1	1	0	1	1	1	1	1	1
0	0	1	0	1	0	1	1	0	1	1	1	1	1
0	0	1	0	1	1	1	1	1	0	1	1	1	1
0	0	1	1	0	0	1	1	1	1	0	1	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	1	0

Memory Address Decoding : 3 to 8

A_{13} through A_{15} select a 2764

A_{16} through A_{19} enable the decoder



(Not sure about 2764 pinout, text is in error with 2716)

Memory Interfacing

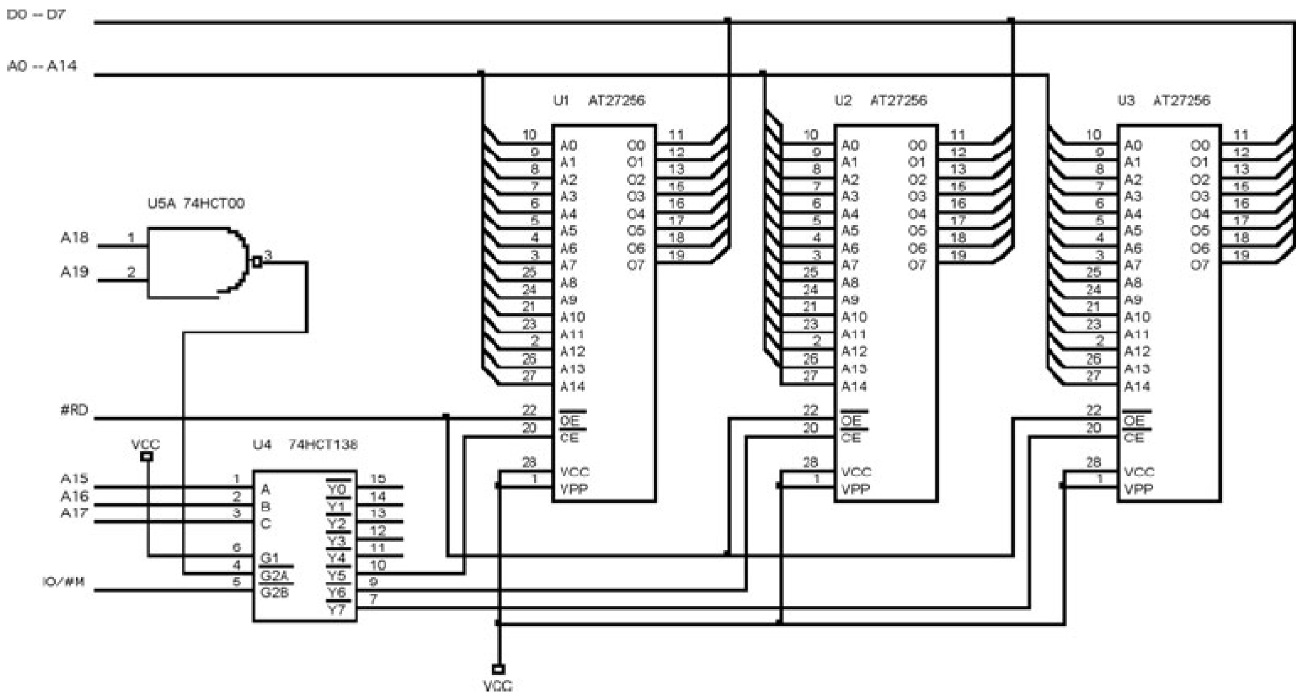


FIGURE 10–20 Three 27256 EPROMs interfaced to the 8088 microprocessor.

Memory Address

Method 3

Programmable Logic Devices (PLDs)

These devices implement a Boolean function against each memory chip connection

- E-g. Programmable Logic Array (PLA)

HM6264 & 27C256 RAM/ROM devices

- Low-cost low-capacity memory devices

- First two numeric digits indicate device type
  - RAM: 62
  - ROM: 27
- Subsequent digits
  - HM6264 → 8KB (13 address lines, 8 data lines)
  - 27C256 → 32KB (15 address lines, 8 data lines)

Questions?

THANK YOU!