Microprocessors and Microcontrollers (EE-231)

Lecture-11

Main Objectives

- Hardware Specs of 8086/8088
  - ➢Bus Timing
  - ➢Ready and Wait state Generation
  - ➢Maximum Mode vs Minimum Mode
  - ➢8288 Bus Controller

Bus Timing

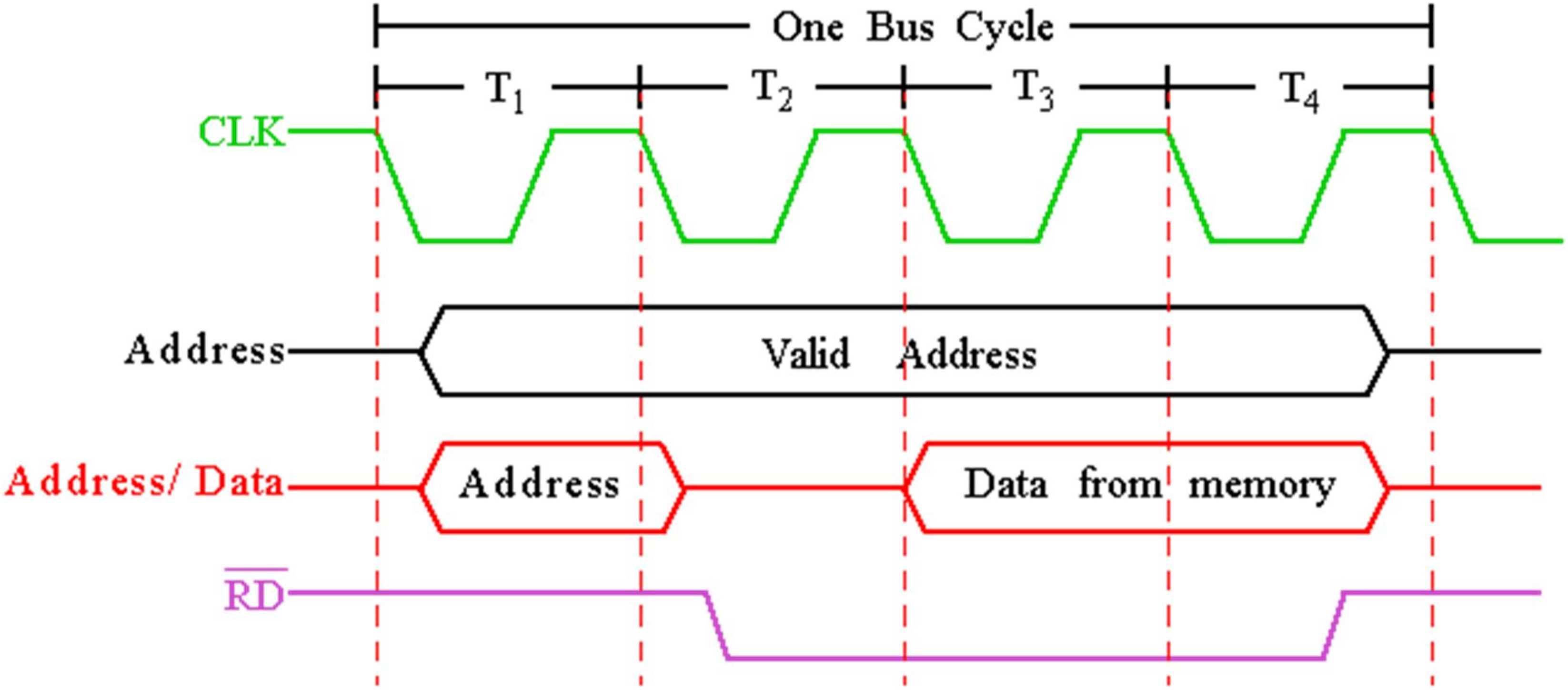
The 8086/8088 microprocessors use the memory and I/O in periods called **bus cycles.**

Each bus cycle consists of 4 clock cycles.

Thus for 8086 running at 5MHz it would take 800ns for a complete bus cycle.

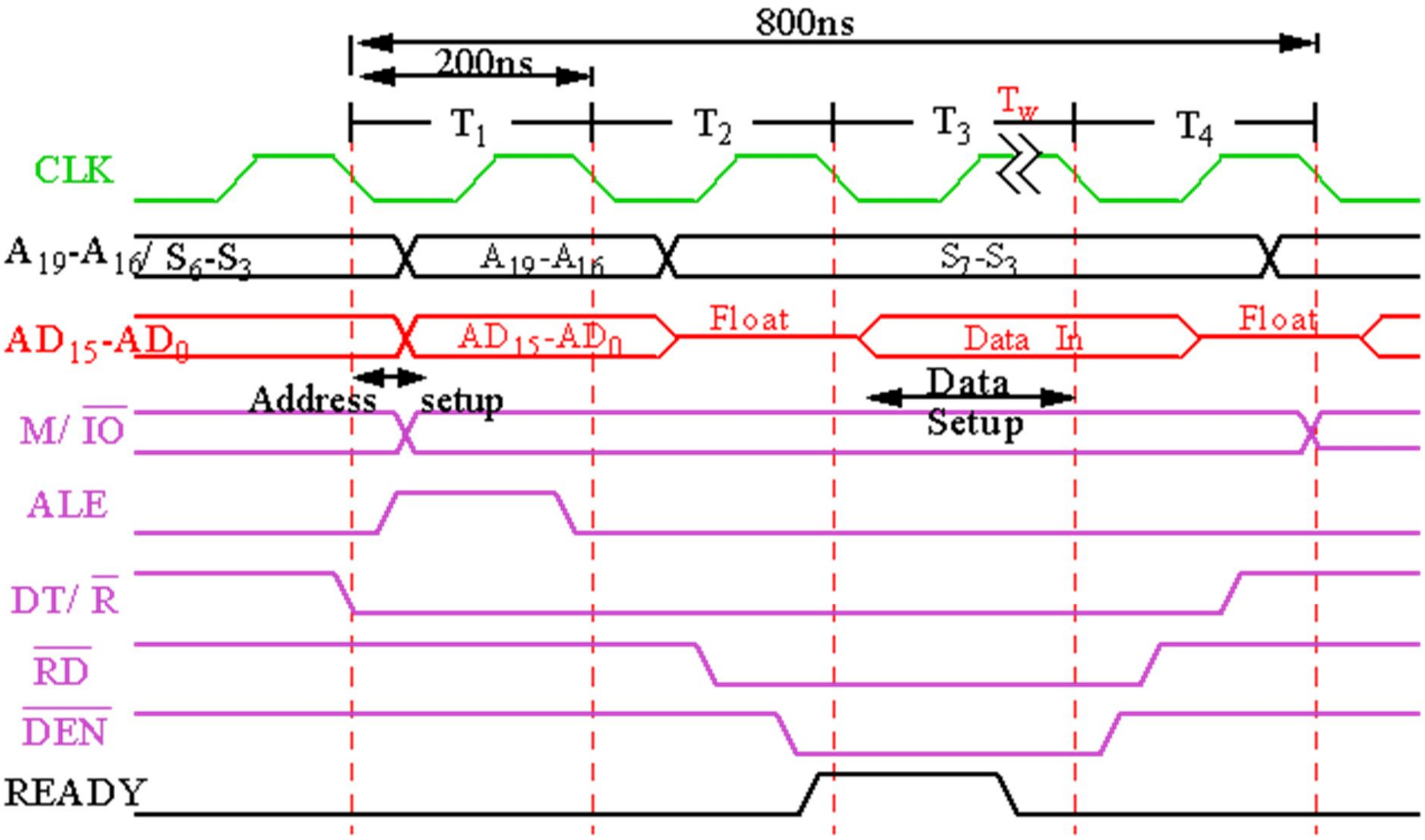
Each read or write operation take 1 bus cycles.

Read Timing (Simplified)



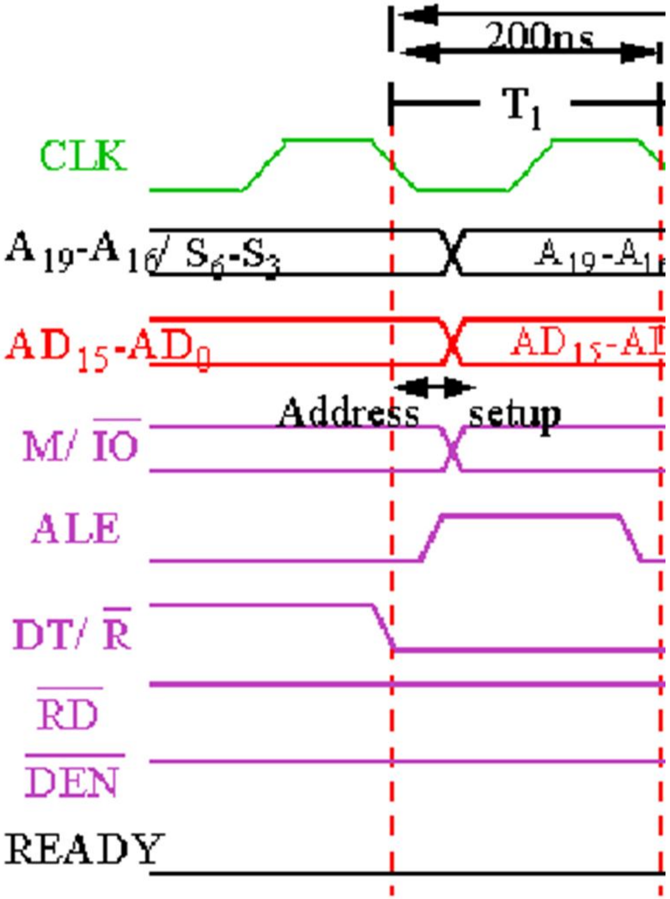
Simplified 8086 Read Bus Cycle

Read Timing



Bus Timing for a Read Operation

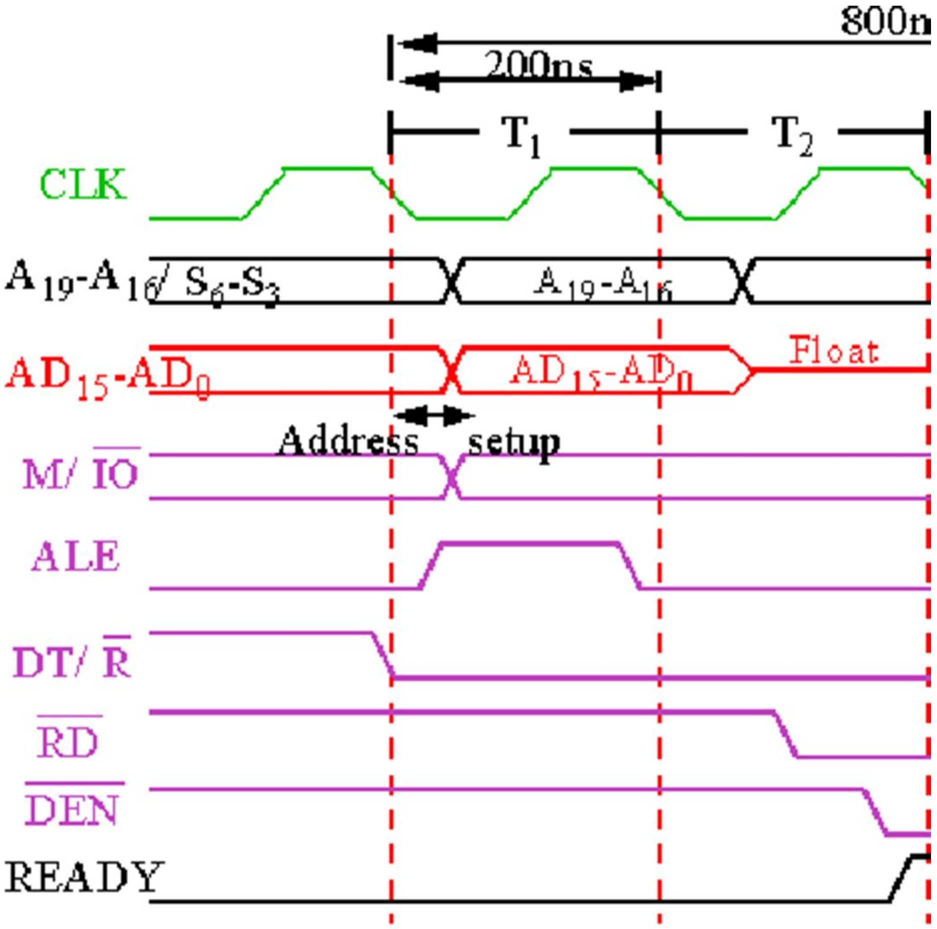
Read Timing



During T $_1$:

- *The address is placed on the Address/Data bus.*
- *Control signals M/* IO *, ALE and DT/* R *specify memory or I/O, latch the address onto the address bus and set the direction of data transfer on data bus.*

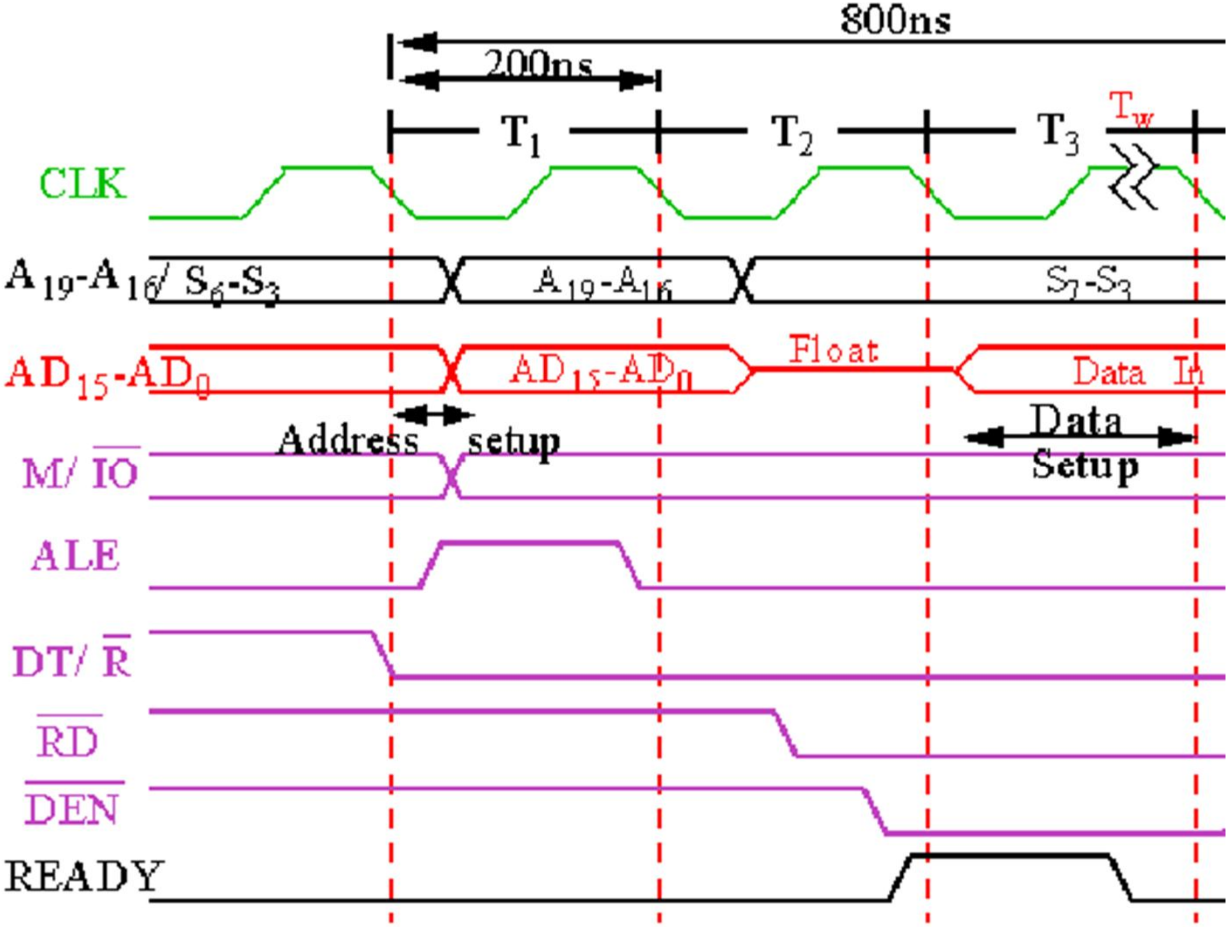
Read Timing



During T $_{\mathbf{2}}$ *:*

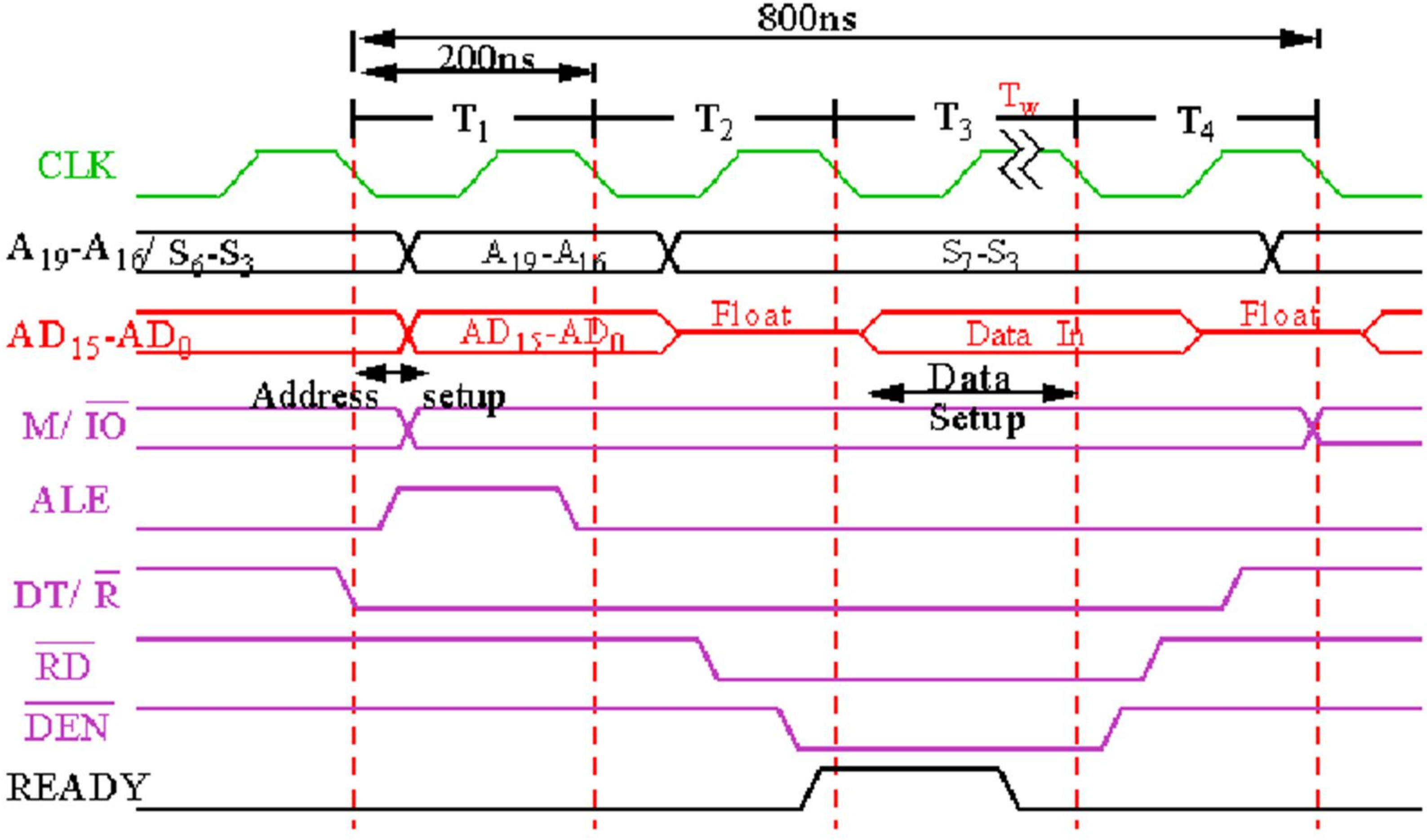
- *8086 issues the* RD *or* WR *signal,* DEN , *and, for a write, the data.*

- DEN *enables the memory or I/O device to receive the data for writes and the 8086 to receive the data for reads.*



During T_3 :

- *This cycle is provided to allow memory to access data.*
- *READY is sampled at the end of T_2 .*
    - *If low, T_3 becomes a wait state.*
    - *Otherwise, the data bus is sampled at the end of T_3 .*

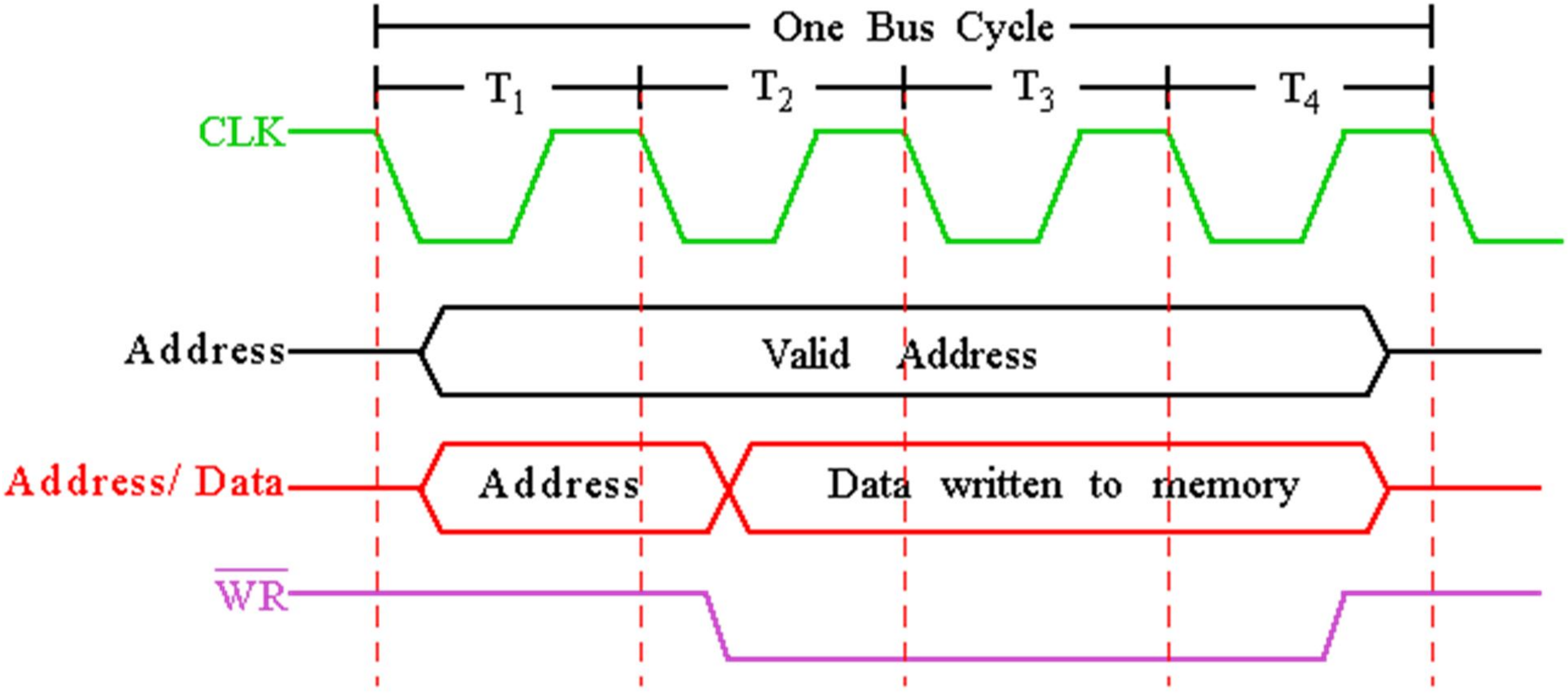


Bus Timing for a Read Operation

During T $_4$:

- •*All bus signals are deactivated, in preparation for next bus cycle.*
- •*Data is sampled for reads, writes occur for writes.*

Write Timing (simplified)



Simplified 8086 Write Bus Cycle

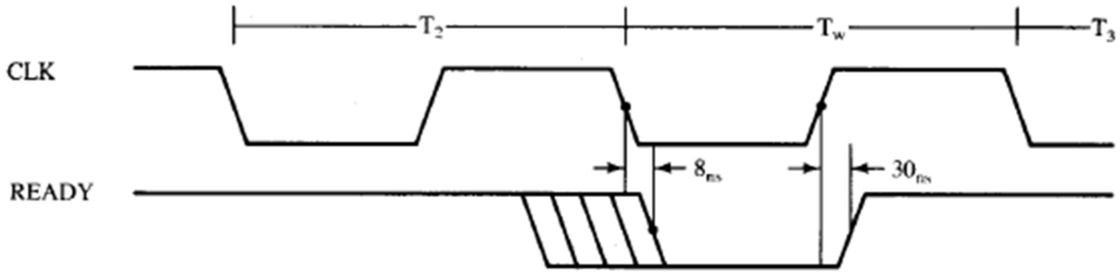
Convert this simple timeline to include all the necessary pins

Ready and the Wait State

- Wait state(T_W) :
  - It is an extra clocking period, inserted between T2 & T3 in order to give memory and other **slow** devices to complete data transfer.
- READY input :
  - This input causes wait states for slower memory & I/O components
  - It is sampled at the end of T2, and if applicable, in middle of T_W
  - if READY = 0 at end of T2 : T3 is delayed and T_W is inserted between T2 and T3
  - READY is next sampled at middle of T_W : to determine whether the next state is another T_W or T3

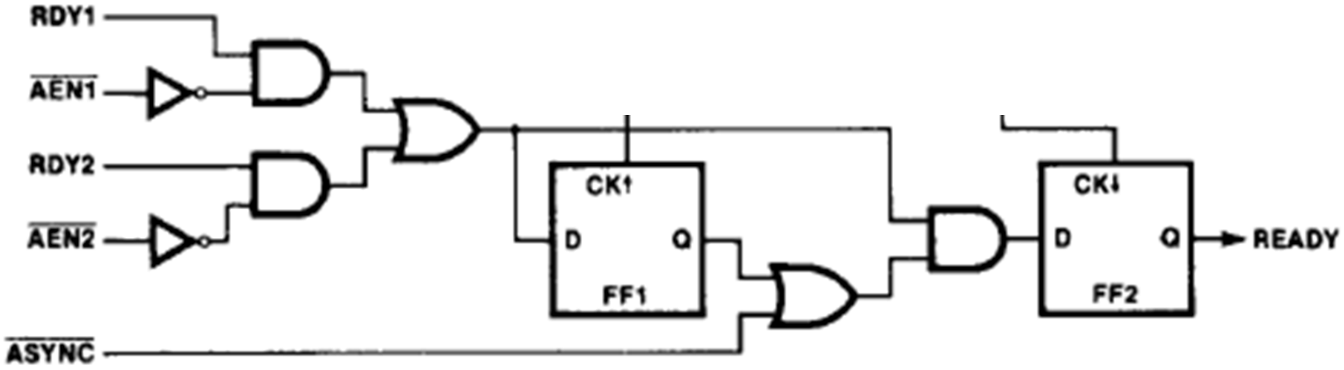
Ready and the Wait State

- The timing requirements of the Ready signal are given below.
- These requirements are guaranteed by the 8284A clock generator.
- 8284A receives wait state information via its RDY input and then through its internal synchronization circuitry, it generates Ready signal for micro Processor.
- If one wait state in inserted, then the memory access time, normally 460 ns with a 5 MHz clock, is lengthened by one clocking period (200ns) to 660 ns.



RDY and the 8284A

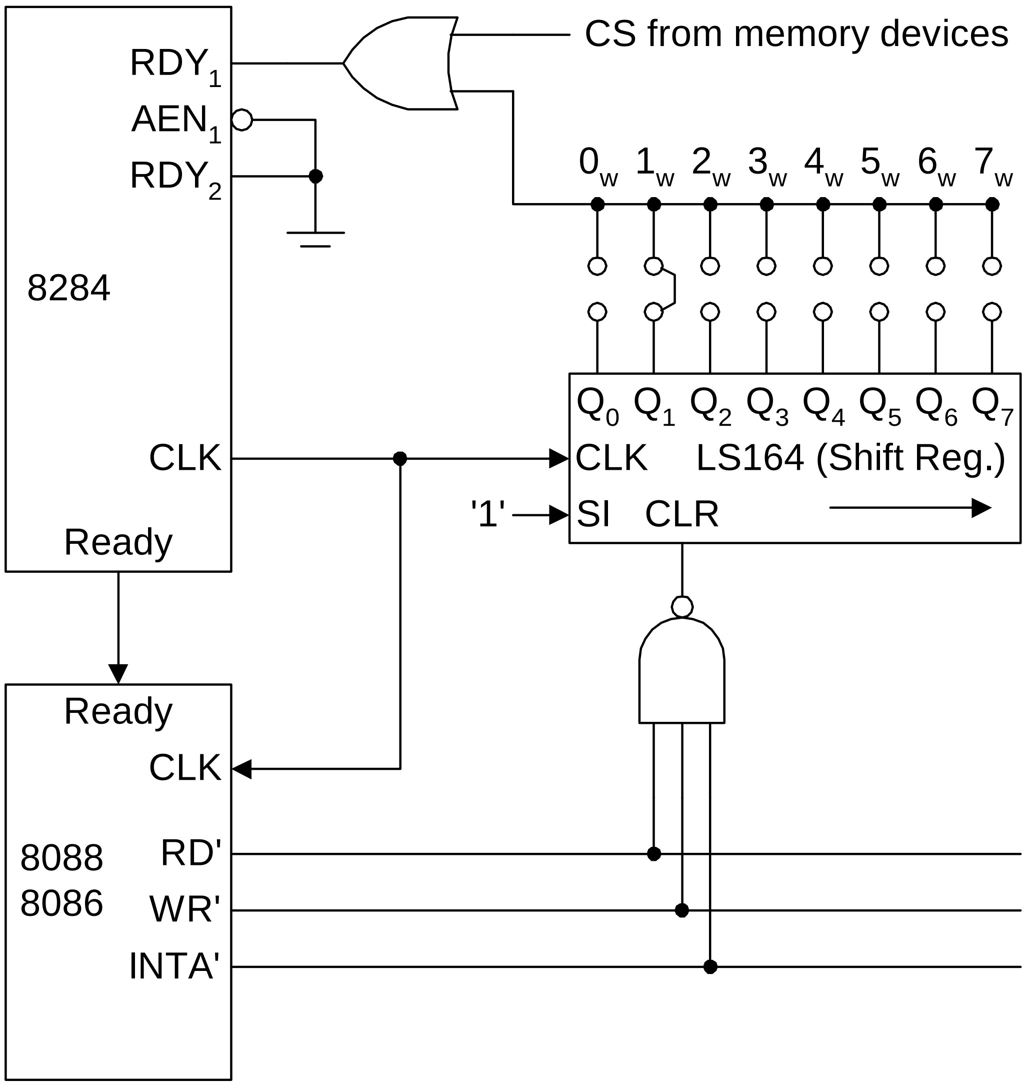
- Internal structure of 8284A
  - RDY1•AEN1’ + RDY2•AEN2’ : to generate input to one or two stage of synchronization
  - ASYNC’=1(high) : select one stage of synchronization
    - RDY : kept from reaching 8086/88 READY pin until the next negative edge of clock
  - ASYNC’=0(ground): select two stage of synchronization
    - on 1st positive edge of clock : 1st FF capture RDY
    - output of 1st FF : fed to 2nd FF
    - on next negative edge of clock : 2nd FF capture RDY



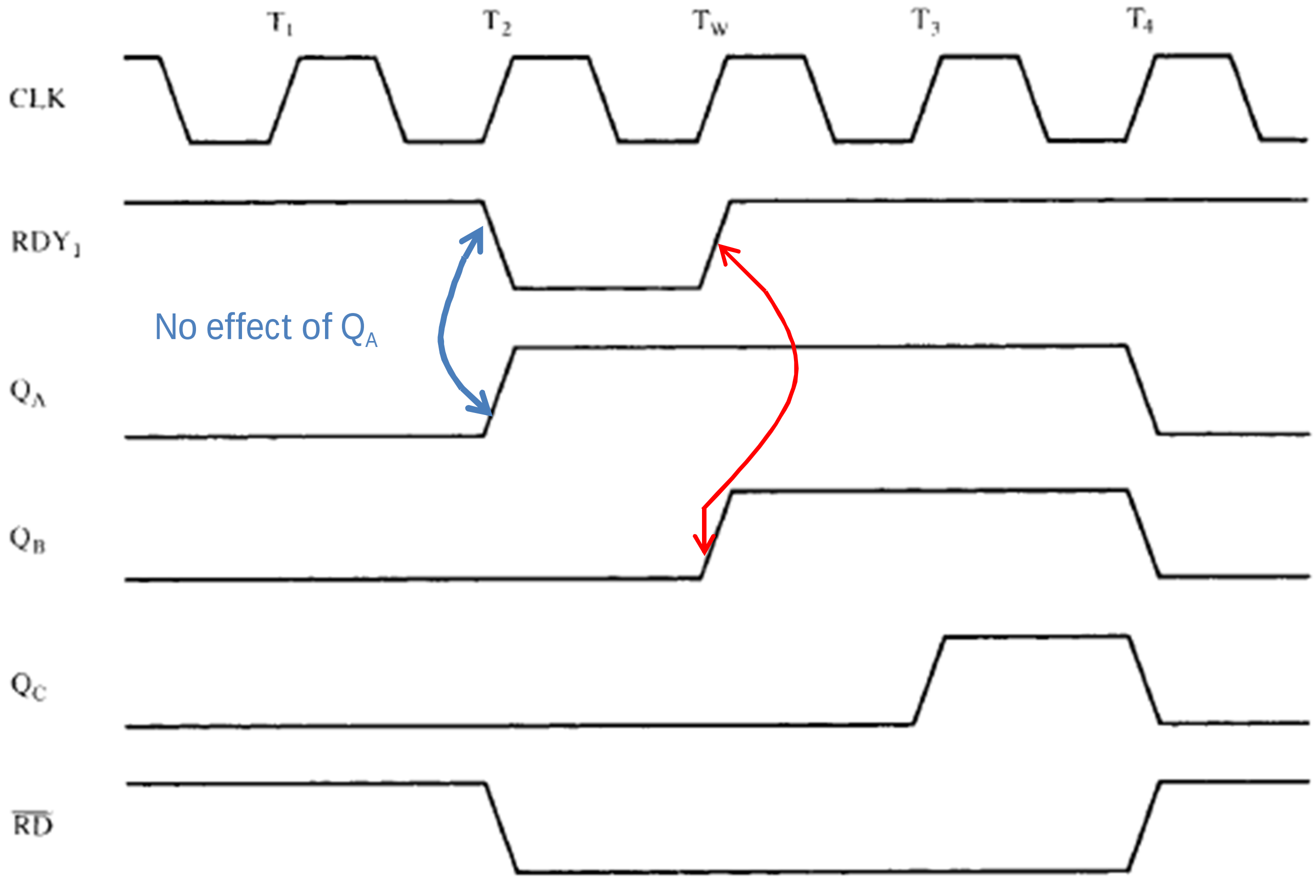
RDY and the 8284A

- A circuit that generates almost any no of wait states for µP can be implemented via 8-bit shift reg.(74LS164)
- Its Q outputs all have zero whenever cleared.
- It is cleared back when RD’=WR’=INTA’=1 ,this condition remains until state T2 and till then Q = 0
- At positive edge of T2 :
  - shift right happens, Q_A = 1(because Serial Input(SI=1).
  - RDY1 = 0 because CS’ = 0 || Q_B = 0
- Therefore a wait state is inserted
- At positive edge of Tw :
- Another shift right happens, Q_A = Q_B = 1
  - RDY1 = 1 because CS’ = 0 || Q_B = 1
- Therefore no more wait state is inserted

Wait state generation circuit



Wait state generation Timing



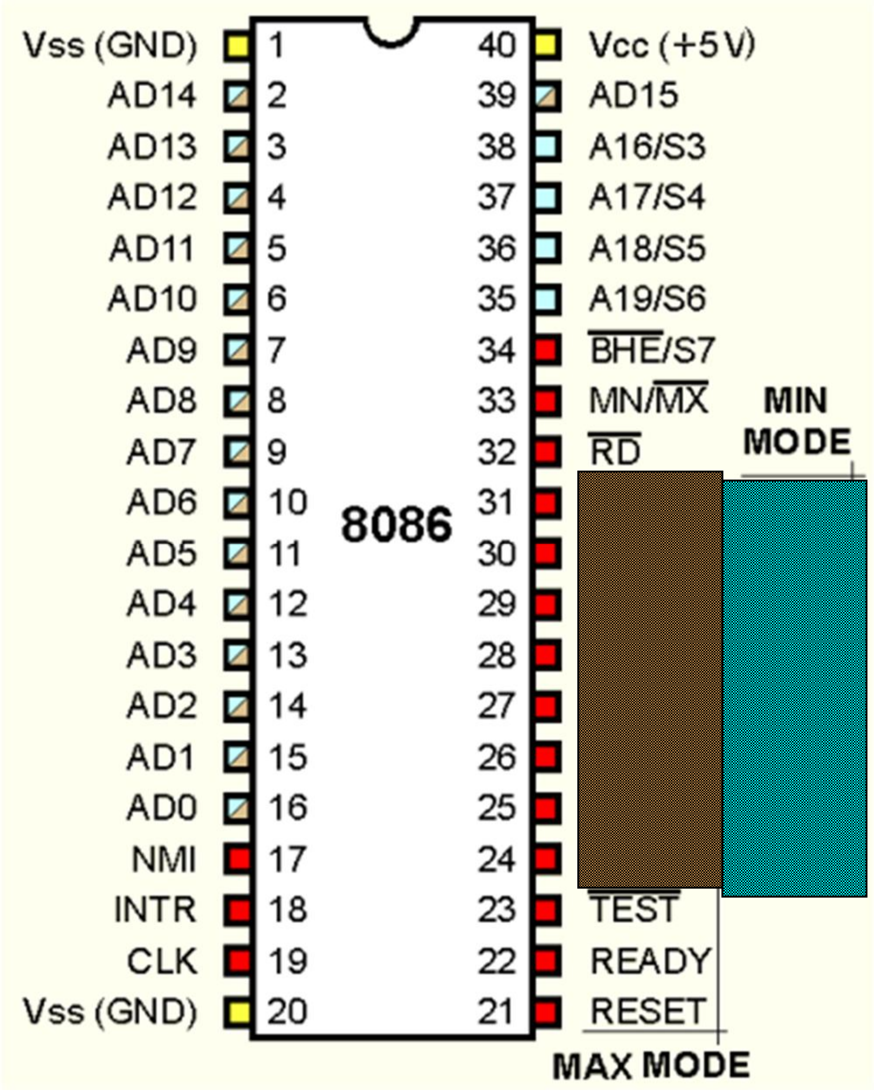
MIN and MAX Mode

- *Controlled through the MN/* MX *pin.*

- **Minimum mode** *is cheaper since all control signals for memory and I/O are generated by the microprocessor.*

- **Maximum mode** *is designed to be used when a coprocessor (8087) exists in the system.* Some of the signals must be externally generated. This requires the addition of an external bus controller—the 8288 bus controller
- This mode was dropped from Intel family starting from 80286

What is the apparent difference ?



The 8288 Bus Controller

- It provides signals eliminated from 8086/88 by maximum mode operation
- S2,S1,S0 (status) input :

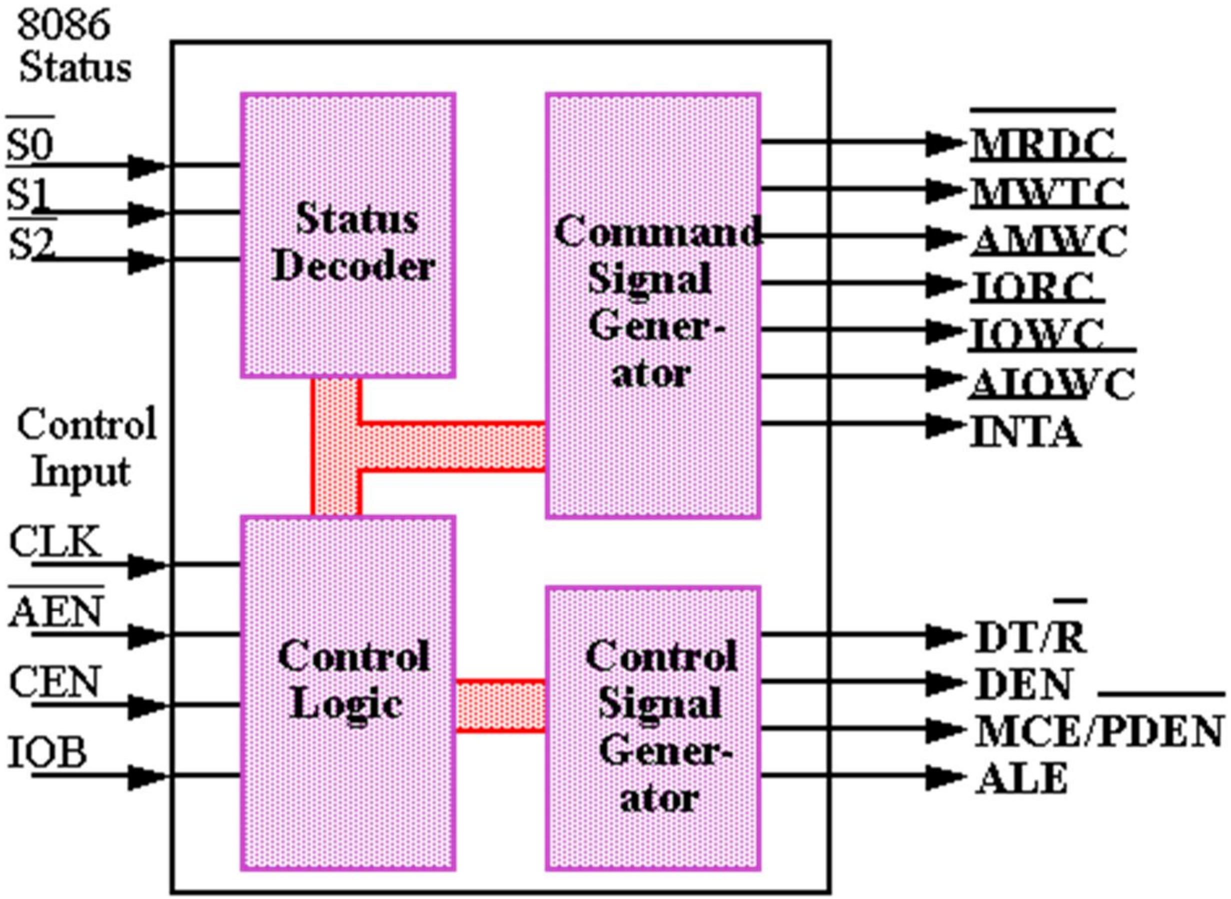
  It is connected to status output on µP. It is devoted to generate the timing signals for the system by monitoring the current status of the µP
- CLK input :

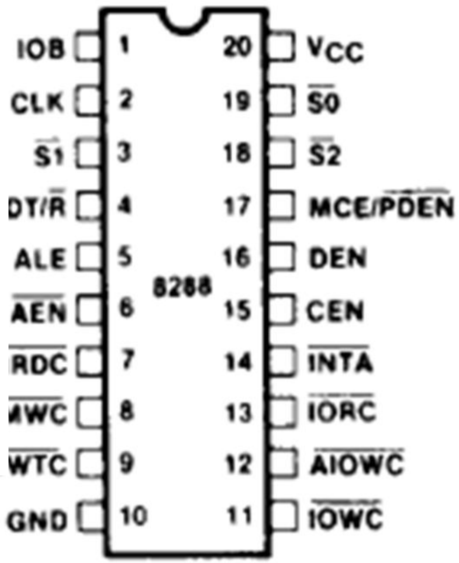
  It is connected CLK output of 8284A. It provides internal timing.
- ALE(address latch enable) output :

  It is used to demultiplex the address/data bus.
- DEN(data bus enable) output :

  It controls bi-directional data bus buffers.
- DT/R'(data transmit/receive) output :

  It tells the direction of data to the buffers.


8086 Status
S0
S1
S2
Status Decoder
Command Signal Gener-ator
MRDC
MWTC
AMWC
IORC
IOWC
AIOWC
INTA
Control Input
CLK
AEN
CEN
IOB
Control Logic
Control Signal Gener-ator
DT/R
DEN
MCE/PDEN
ALE


IOB 1
CLK 2
S1 3
DT/R 4
ALE 5
AEN 6
RDC 7
MWC 8
WTC 9
GND 10
8288
20 VCC
19 S0
18 S2
17 MCE/PDEN
16 DEN
15 CEN
14 INTA
13 IORC
12 AIOWC
11 IOWC

The 8288 Bus Controller

- AEN’(address enable) input :

  It cause to enable memory control signals
- CEN(control enable) input:

  It enables the command output
- IOB(I/O bus mode) input :

  It selects either system bus or I/O bus mode operation
- INTA’(interrupt acknowledge) output :

  It generates acknowledgement whenever interrupt is requested, to tell the device that the interrupt is being serviced.
- MCE/PDEN’(master cascade/peripheral data) output :

  It selects cascade operation for interrupt controller if IOB is 0, and enable the I/O bus transceivers if IOB is 1

The 8288 Bus Controller

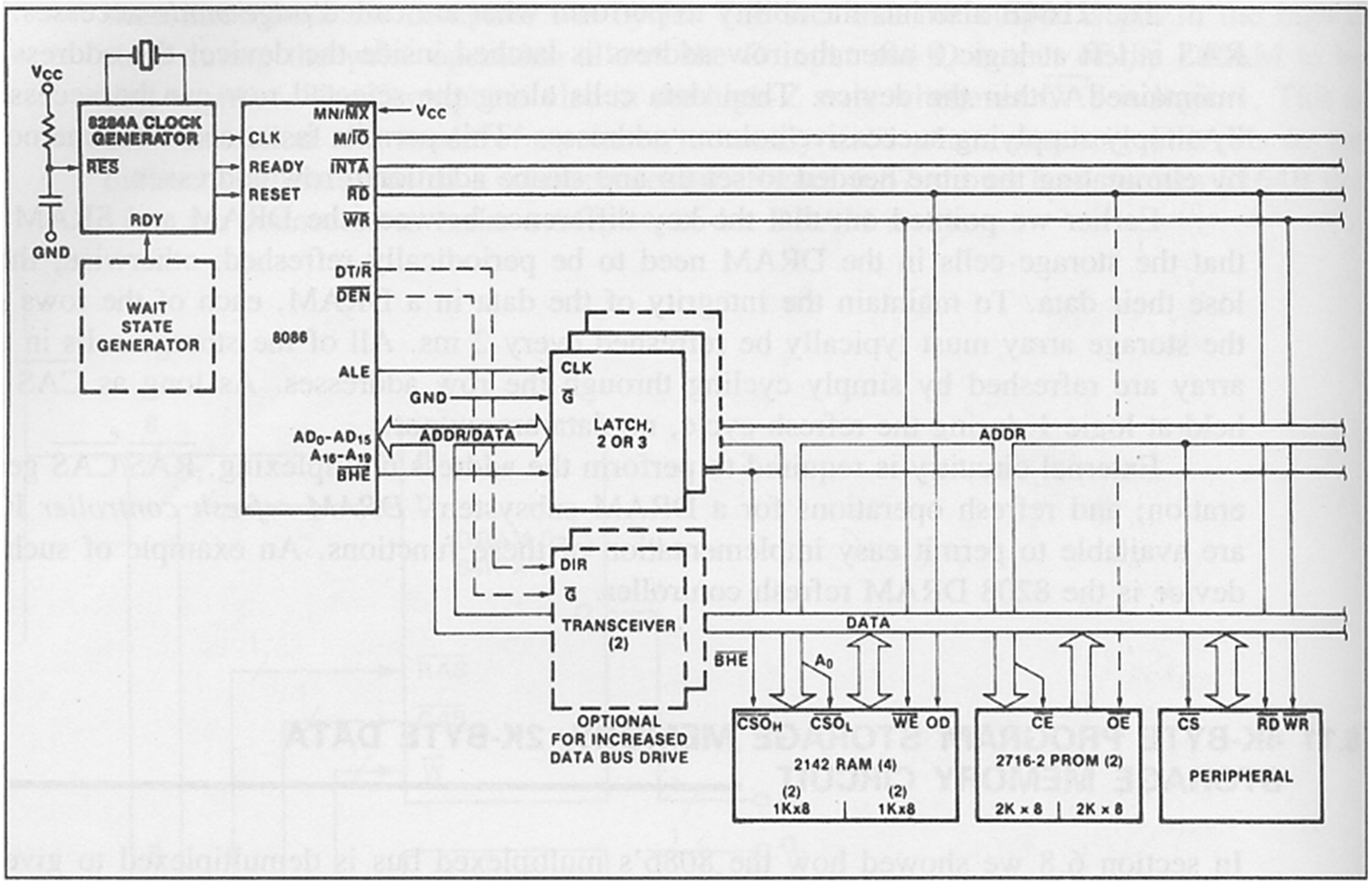
Note that In 8288 Bus controller there are separate Read and Write control signals for memory and I/O

- AIOWC'(advanced I/O write command) output :

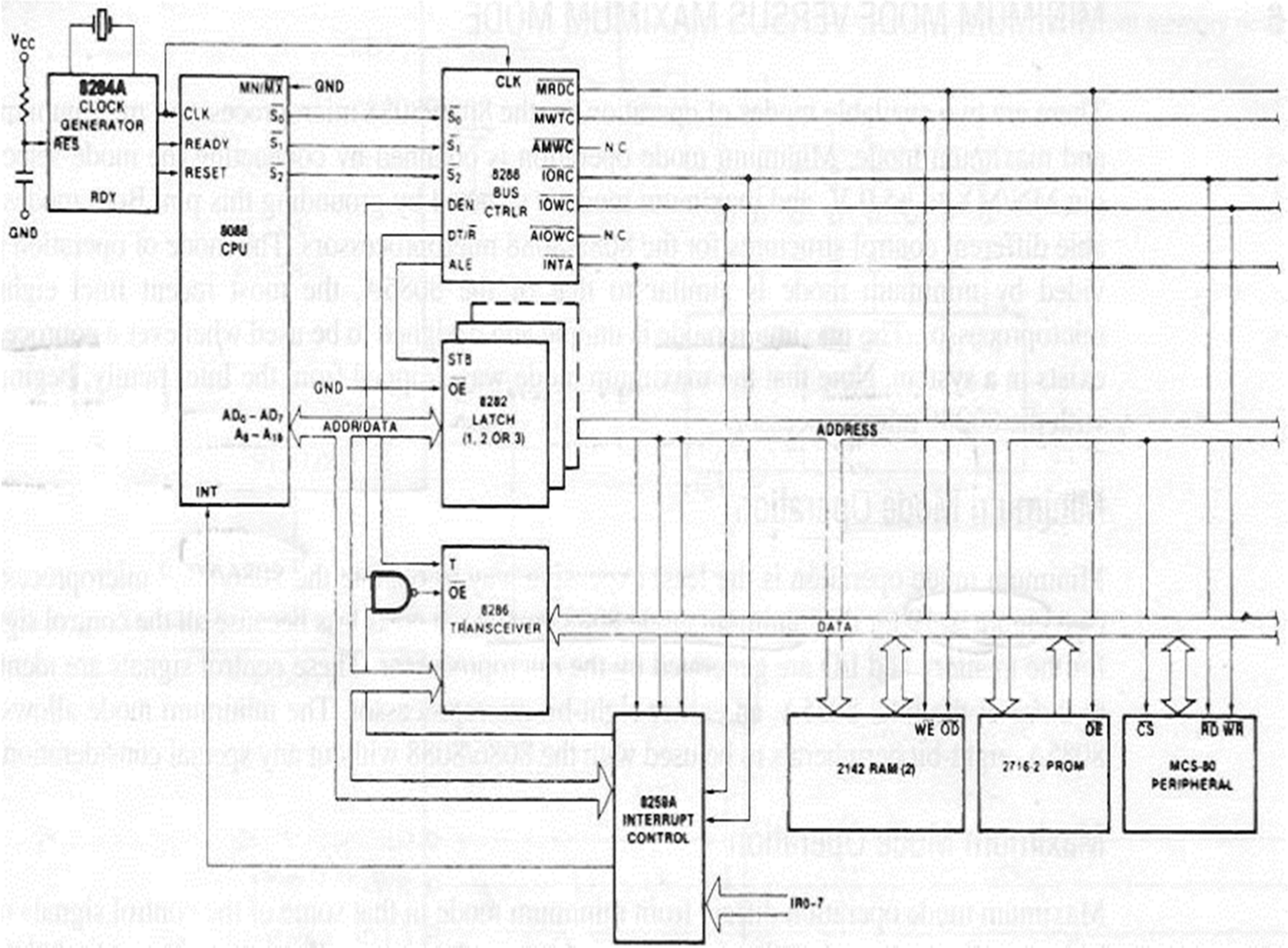
  It provide I/O with its advanced I/O write control signal
- AMWC'(advanced memory write command):
- It provides memory with an early or advanced write signal.
- IOWC'(I/O write) :

  It provide I/O with its main write signal
- IORC' (I/O Read command):
- It provides I/O with its read control signal.
- MWTC'(Memory write command):
- It provides memory with its normal write control signal.
- MRDC'(Memory Read command):
- It provides memory with a read control signal.

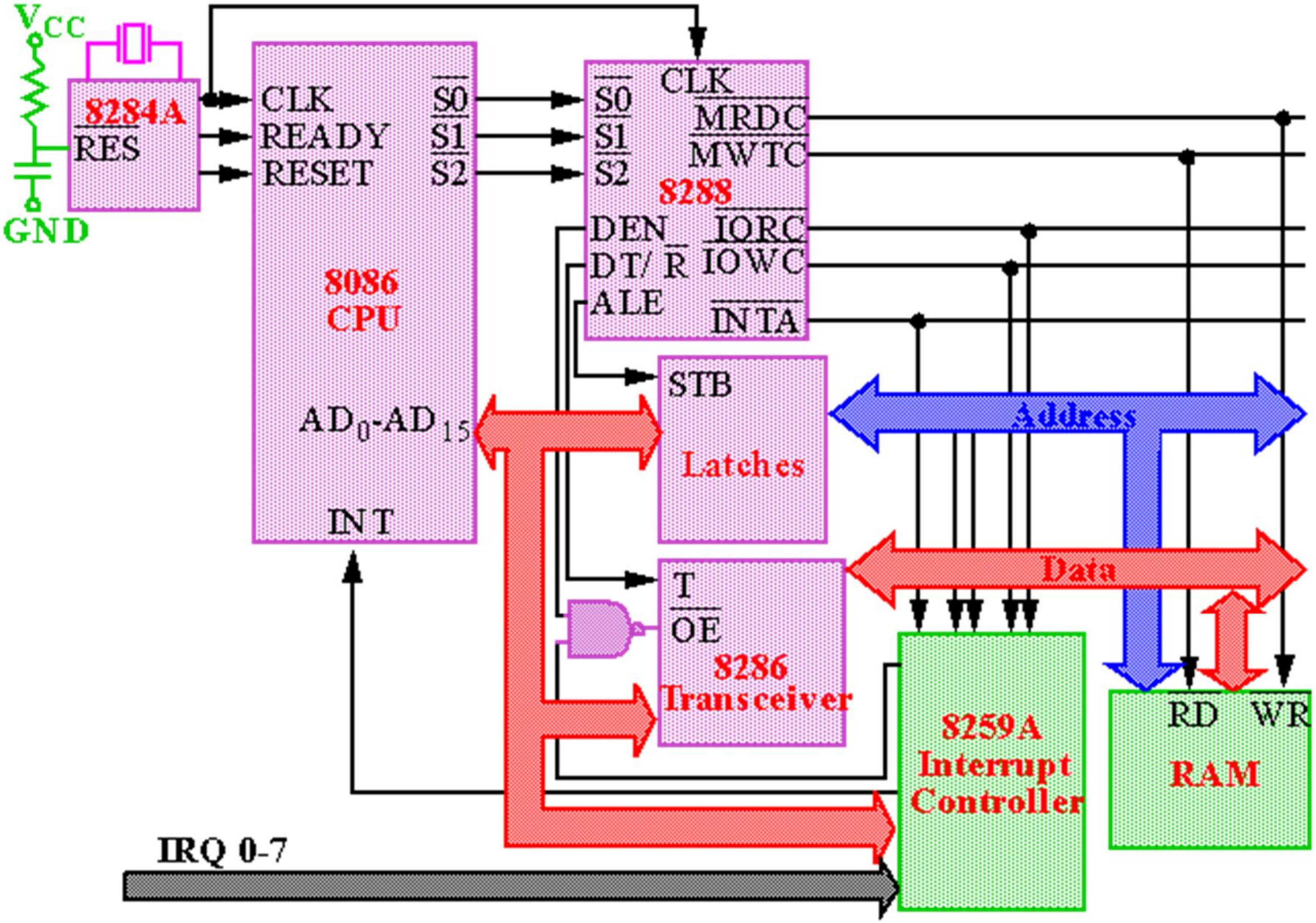
8088 Minimum Mode System Block Diagram



8088 Maximum Mode System Block Diagram



8086 Maximum Mode System simplified Block Diagram



Memory Types

- Two basic types:
  - ROM: Read-only memory
  - RAM: Read-Write memory
- Four commonly used memories:
  - ROM
  - Flash (EEPROM)
  - Difference between EEPROM and Flash
  - Static RAM (SRAM)
  - Dynamic RAM (DRAM)

Memory Chips

- The data pins are typically bi-directional in read-write memories.
  - The number of data pins is related to the size of the memory location. For example, an 8-bit wide (byte-wide) memory device has 8 data pins.
- Each memory device has at least one chip select (CS) or chip enable (CE) or select (S) pin that enables the memory device.
  - This enables read and/or write operations.
  - If more than one are present, then all must be 0 in order to perform a read or write.

SRAM vs. DRAM

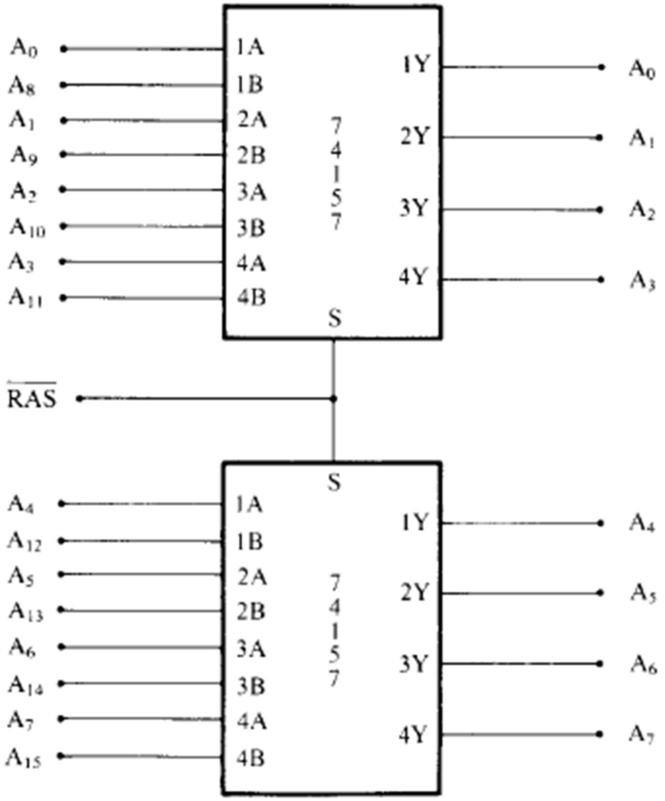
- SRAMs
  - SRAMs used for caches have access times as low as 10ns .
- DRAMs
  - SRAMs are limited in size (up to about 128Kb).
  - DRAMs are available in much larger sizes, e.g., 64M X 1.
  - DRAMs MUST be refreshed every 2 to 4 ms
  - Since they store their value on an integrated capacitor that loses charge over time.

  - What is SDRAM ?
  - What is EDO ?

DRAM Addressing

- Dram has quite large addresses. Therefore it is not possible to provide all the signals at a time.
- Therefore, one way to provide address is to divide it into portions.
- We use signals CAS and RAS to select column and a Row address.
- First RAS is activated and the first half of address is latched on the row address latch.
- Second CAS is activated and second half of address is latched to the column address latch.

DRAM Addressing

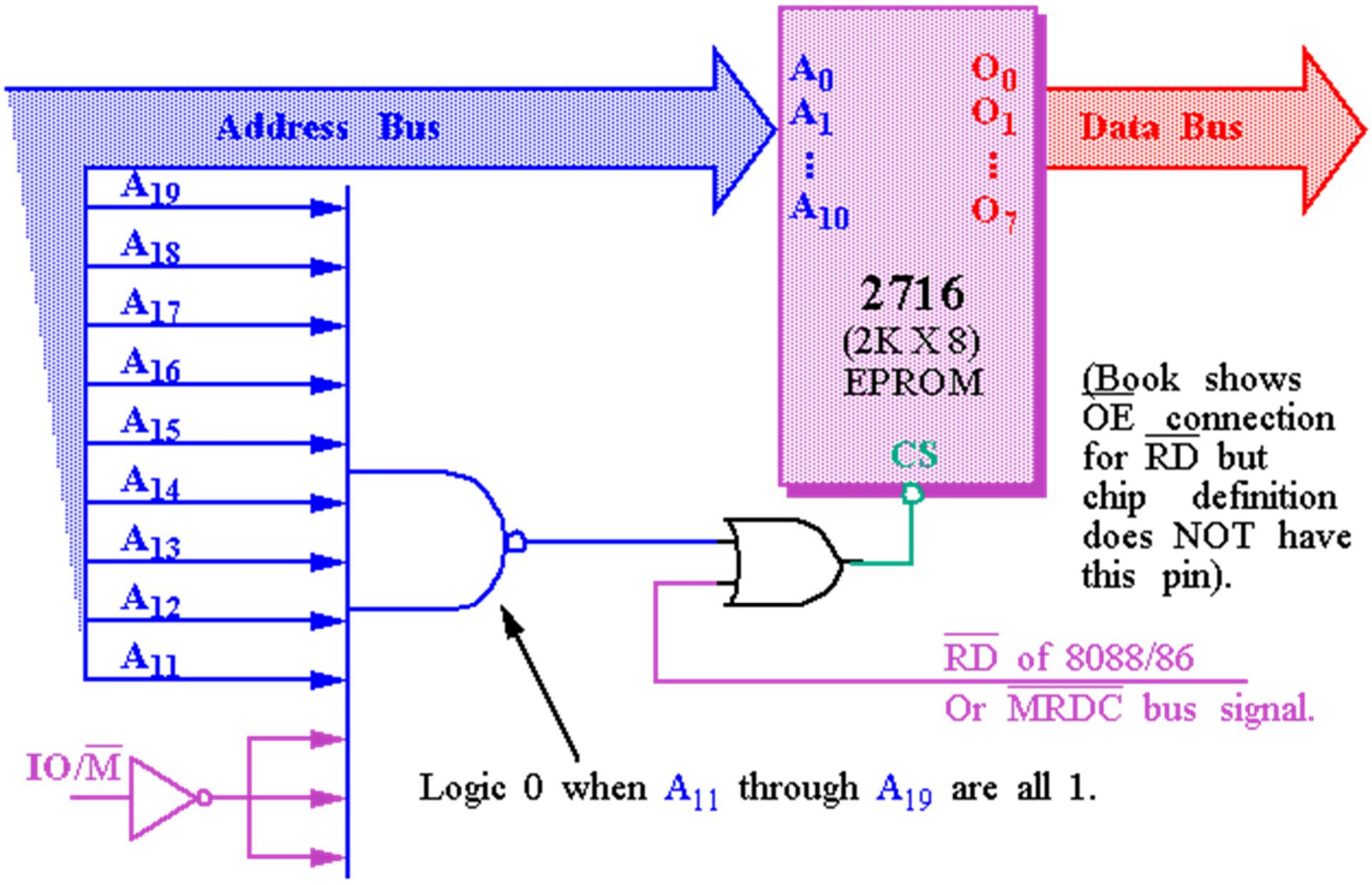


Memory Address Decoding

- 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 8088 issues 20-bit addresses for a total of 1MB of memory address space.

Ex. Memory Address Decoding

- 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.



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

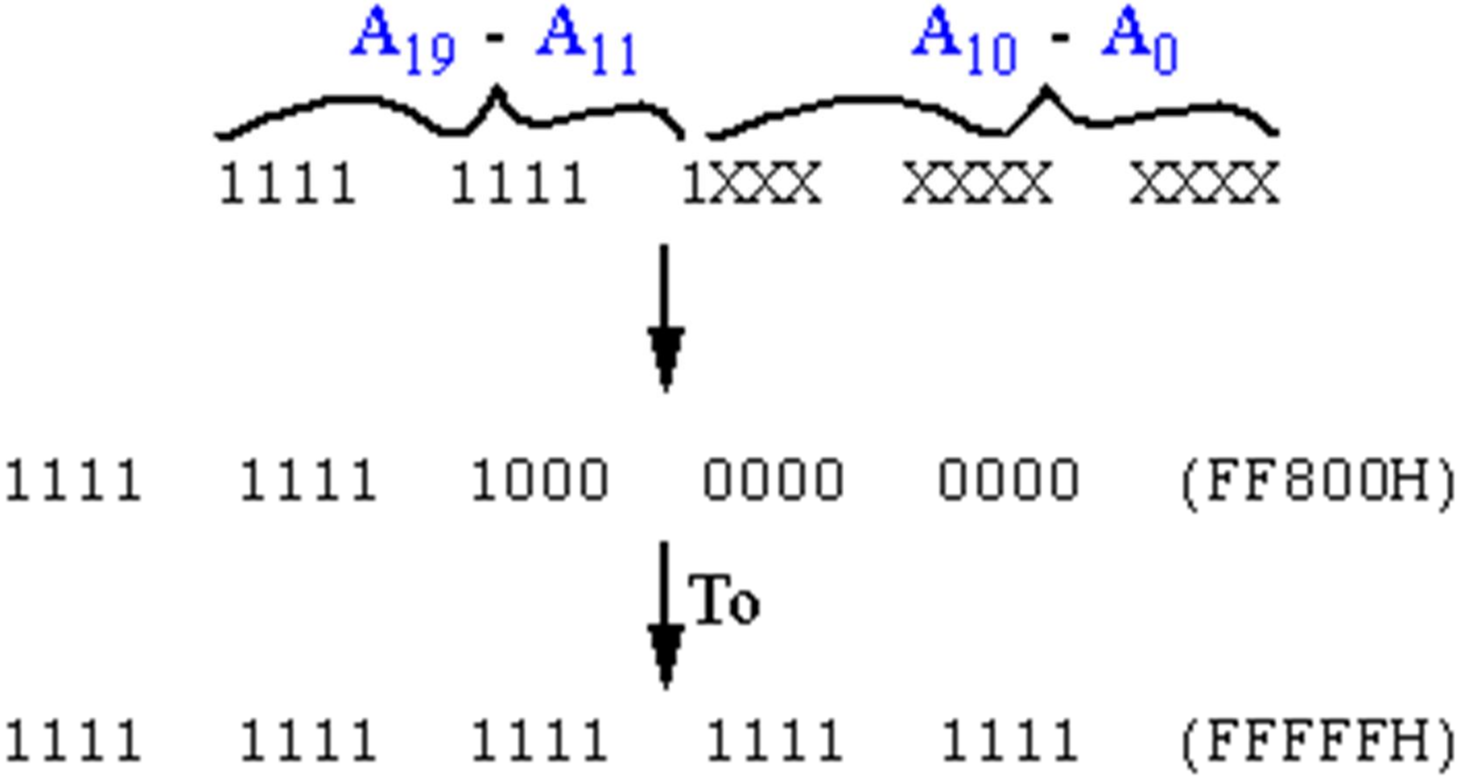
$\overline{RD}$ of 8088/86

Or $\overline{MRDC}$ bus signal.

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

Ex. Memory Address Decoding

- To determine the address range that a device is mapped into:

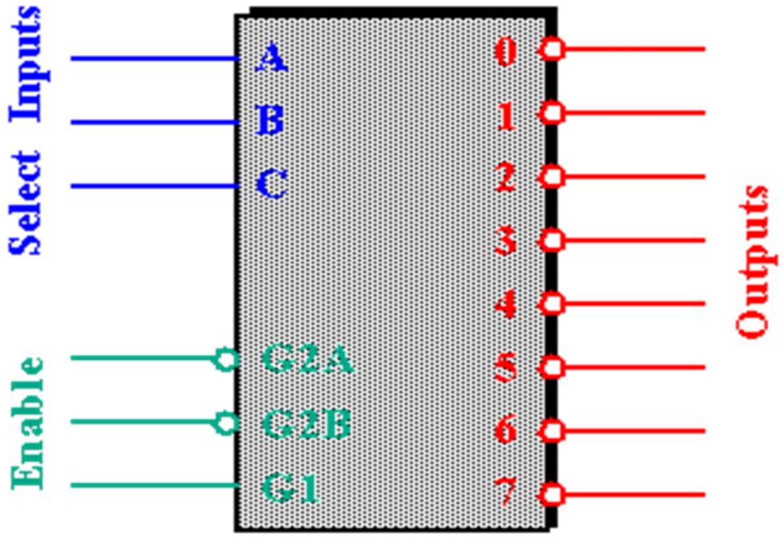

A_{19} - A_{11}
A_{10} - A_0
1111 1111 1XXX XXXX XXXX
1111 1111 1000 0000 0000 (FF800H)
To
1111 1111 1111 1111 1111 (FFFFFH)

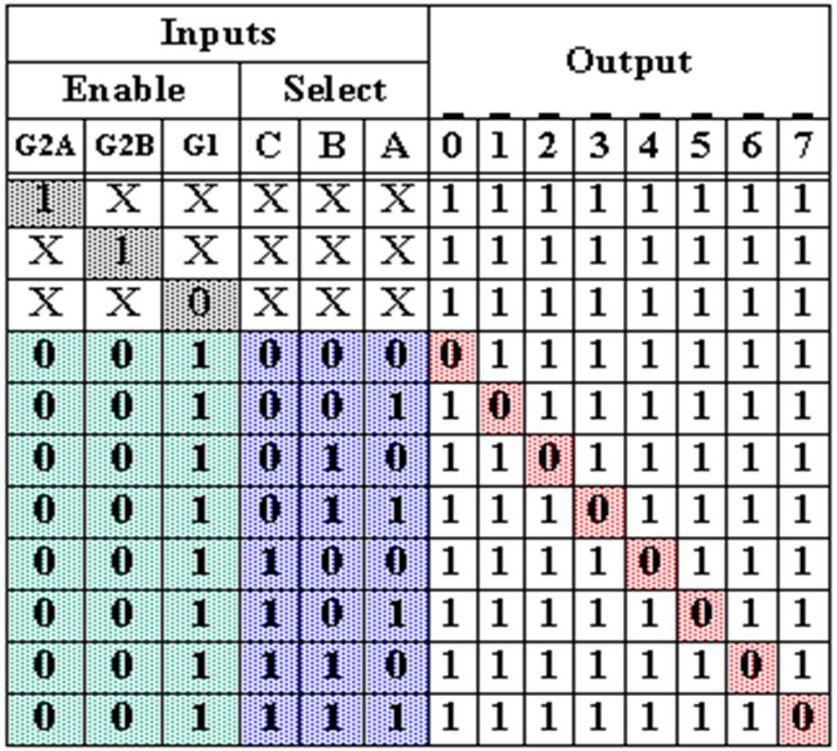
Ex. Memory Address Decoding

- This 2KB memory segment maps into the reset location of the 8086/8088 (FFFF0H).

- NAND gate decoders are not often used. Rather the 3-to-8 Line Decoder (74LS138) is more common.

3-to-8 Line Decoder





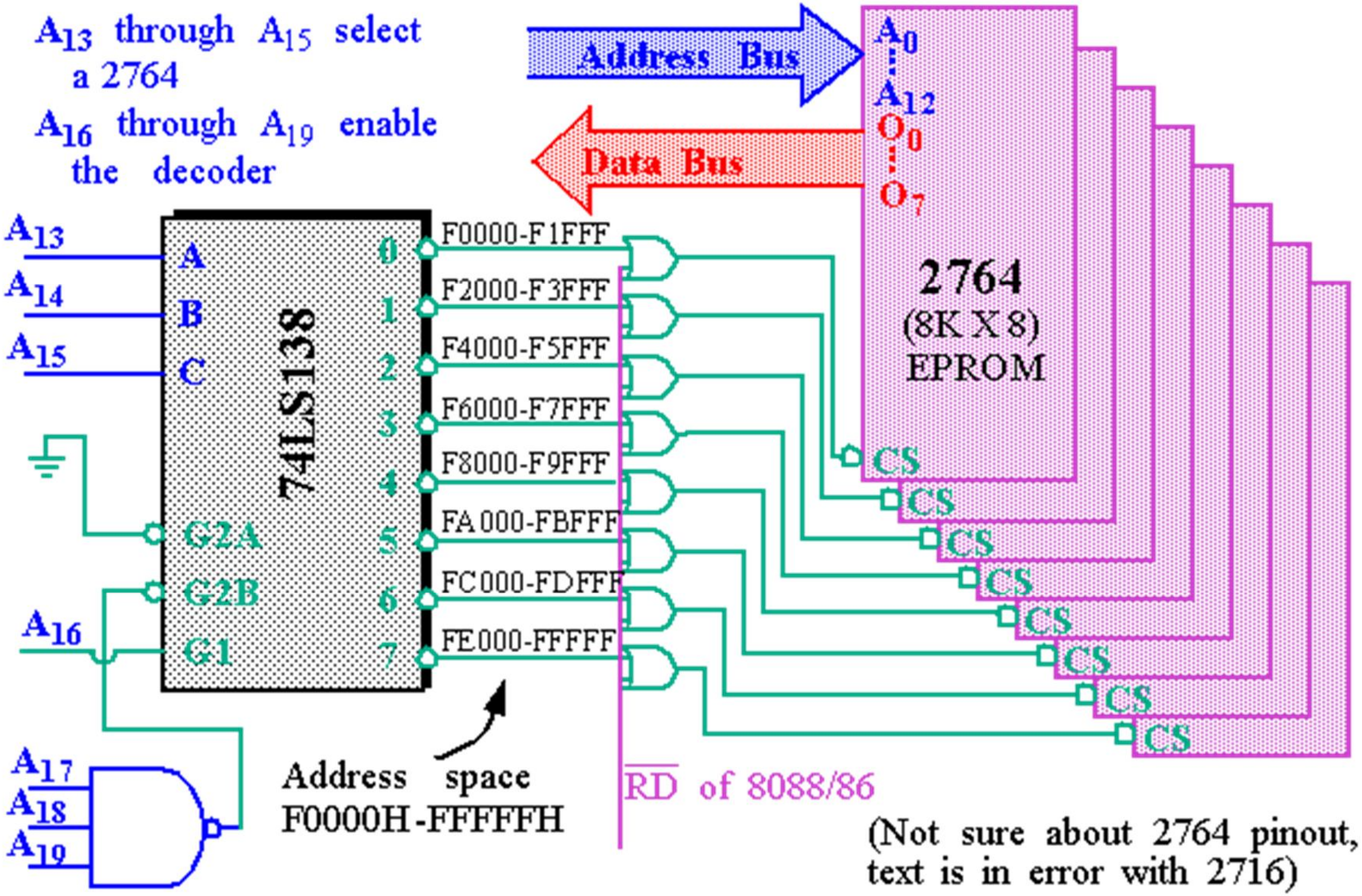
Inputs						Output							
Enable			Select										
G2A	G2B	G1	C	B	A	$\overline{0}$	$\overline{1}$	$\overline{2}$	$\overline{3}$	$\overline{4}$	$\overline{5}$	$\overline{6}$	$\overline{7}$
1	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	0	X	X	X	1	1	1	1	1	1	1	1
0	0	1	0	0	0	0	1	1	1	1	1	1	1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

- G2A, G2B, and G1 must be active.
- Each output of the decoder can be attached to an 2764 EPROM (8K X 8).

EPROM 2764 x 8

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

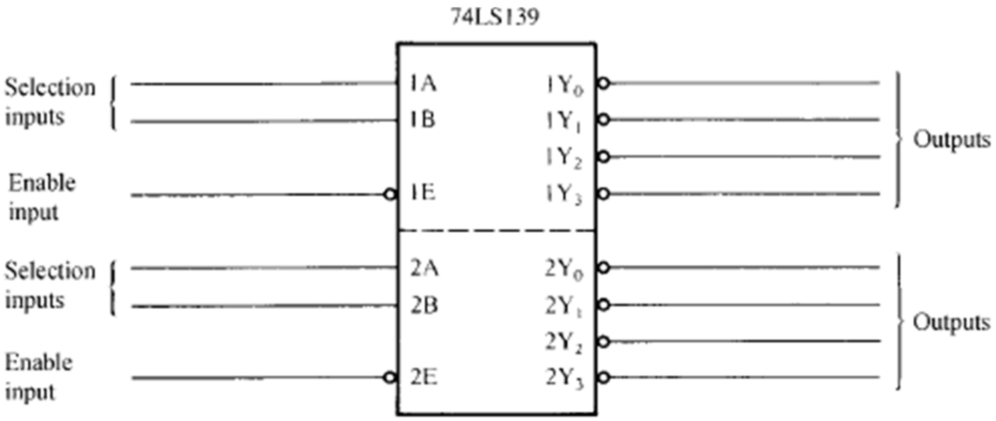
A_{16} through A_{19} enable the decoder

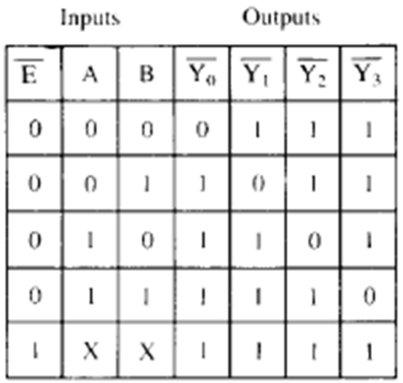


(Not sure about 2764 pinout, text is in error with 2716)

	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
f0000:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
fffff:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROMx:								0	0	0	0	0	0	0	0	0	0	0	0	0
								1	1	1	1	1	1	1	1	1	1	1	1	1
ROM1:	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM2:	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
					0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM3:	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
					0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
ROM8:	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

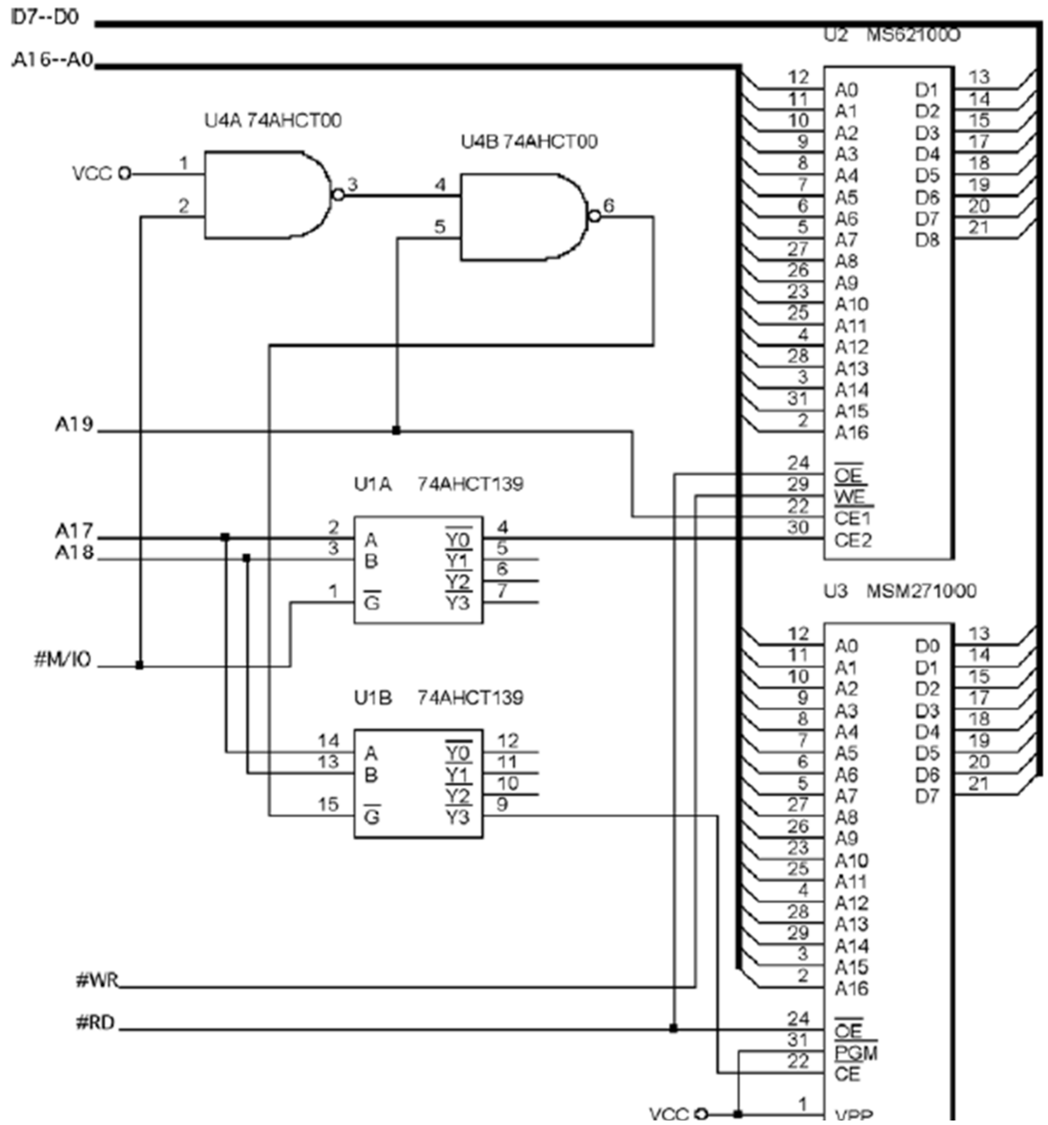
74LS139 Dual Decoder





Inputs			Outputs			
$\overline{E}$	A	B	$\overline{Y_0}$	$\overline{Y_1}$	$\overline{Y_2}$	$\overline{Y_3}$
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0
1	X	X	1	1	1	1

74LS139 Dual Decoder



D7--D0
A16--A0
U2 MS621000
U4A 74AHCT00
U4B 74AHCT00
VCC
1
2
3
4
5
6
A19
U1A 74AHCT139
A17
A18
A
B
G
Y0
Y1
Y2
Y3
#M/IO
U1B 74AHCT139
14
13
15
12
11
10
9
A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
A12
A13
A14
A15
A16
OE
WE
CE1
CE2
D1
D2
D3
D4
D5
D6
D7
D8
U3 MSM271000
D0
OE
PGM
CE
VPP
#WR
#RD
VCC