

Microcontrollers & Embedded System Design
CSE 315

Md Awsaf Alam

August 9, 2018

Contents

I	8086/8088 Hardware Specifications	7
1	Lecture 1	9
1.1	Differences between 8086 & 8088	9
1.2	Pin Connections	11
2	Lecture 2	13
2.1	Pin Connections Continued	13
2.2	Minimum Mode Pins	13
2.3	Bus Control functions	15
3	Lecture 3	17
3.1	Maximum Mode Pins	17
3.2	Clock Generator (8284A)	18
3.2.1	Pin diagram	19
3.2.2	Pin Functions	19
3.3	Internal Block Diagram of 8284A Clock Generator	20
3.3.1	Operation of the RESET selection	21
4	Lecture 4	23
4.1	Bus Buffering and Latching	23
4.2	Basic of Demultiplexing	23
4.2.1	Demultiplexing 8088	24
4.2.2	Demultiplexing 8086	25
4.3	Buffered System	26
4.3.1	Fully Buffered 8088	26
4.3.2	Fully Buffered 8086	27
4.3.3	Number of chips required for fully buffered microprocessor	28
II	Memory Interface	29
5	Lecture 5	31
5.1	Memory Interface	31
5.1.1	ROM Memory (nonvolatile memory)	31

5.1.2	Delays in operation of an EPROM	32
5.1.3	Static memory or static RAM(SRAM) or Volatile memory	33
5.1.4	Dynamic Ram (DRAM)	33
6	Lecture 6	37
6.1	Address Decoding	37
6.1.1	Simple NAND Gate Decoder	37
6.1.2	3-to-8 Line Decoder(74LS138)	37
6.2	PROM Address Decoder	38
6.3	PLD programmable Decoders	38
7	Lecture 7	41
7.1	8088 and 80188 (8-bit memory interface)	41
7.1.1	Interfacing EPROM to the 8088	41
7.1.2	Interfacing RAM to the 8088	42
8	Lecture 8	45
8.1	Interfacing Flash Memory	45
8.2	Parity for memory error detection	46
8.3	Error correction	48
9	Lecture 9	51
9.1	8086, 80186, 80286, 8038 (16-bit) memory interface	51
9.2	8086 (16-bit) memory interface	54
10	Lecture 10	57
10.1	DRAM	57
10.2	EDO memory (Extended Data Output)	58
10.3	SDRAM (Synchronous Dynamic RAM)	58
10.4	DRAM controllers	58
III	Basic I/O Interface	61
11	Lecture 11	63
11.1	Isolated and memory-mapped I/O	63
11.2	Programmable Peripheral Interface	63
11.2.1	Programming 2C55: Command Registers	65
12	Lecture 12	67
12.1	Mode 0 Operation	67
12.1.1	Mode 0 Operation: Keyboard Matrix interface	68
12.2	Mode 1: Strobed Input	69
12.3	Mode 1: Strobed Output	71
12.3.1	Strobed Output: Printer	72
12.3.2	Comparison between Mode 1 Strobed Input and Output	74
12.4	Mode 2: Bidirectional Operation	74

IV	Interrupt	77
13	Lecture 13	79
13.1	Interrupt	79
13.1.1	Operation of a Protected Mode Interrupt	82
14	Lecture 14	83
14.1	Interrupt Flag Register	83
14.1.1	Enable Trap Flag	83
14.1.2	Disable Trap Flag	84
14.2	Hardware Interrupts	84
14.2.1	INTR and \overline{INTA}	85
14.2.2	Making INTR input edge triggered	86
14.2.3	Applying a fix interrupt vector number FFH	87
14.2.4	Using a 3-state buffer to place a vector type number 80H	88
14.2.5	Using 74ALS244 to expand to accomodate up to 7 interrupt inputs	89
14.2.6	Daisy-Chained Interrupts	90
14.3	8259A programmable interrupt controller (PIC)	90
15	Lecture 15	93
15.1	Direct Memory Access (DMA)	93
15.1.1	Control signal generation in 8086/8088	94
15.1.2	8237 DMA Controller	94
15.1.3	Arithmetic Coprocessor	95

Part I

**8086/8088 Hardware
Specifications**

Chapter 1

Lecture 1

1.1 Differences between 8086 & 8088

- Virtually no difference between these two μps . Both are packaged in 40-pin dual in-line packages (DIPs)

8086 16 bit μp with a 16-bit data bus ($AD_0 - AD_{15}$)

8088 16 bit μp with a 8-bit data bus ($AD_0 - AD_7$)

8086 : M/\overline{IO} ; 8088 : IO/\overline{M} ;

8086 (PIN 34): $\overline{BHE}/S7$; 8088 (PIN 34): SSO ;

Power Supply Requirements • Both : +5.0V with a supply voltage tolerance of $\pm 10\%$

- Both : $32^\circ F$ to $180^\circ F$
- 8086 : $360mA$; 8088 : $340mA$ (max supply current)
- CMOS version : 80C86 and 80C88 : $10mA$ and $-40^\circ F$ to $225^\circ F$

Pin diagram 8086

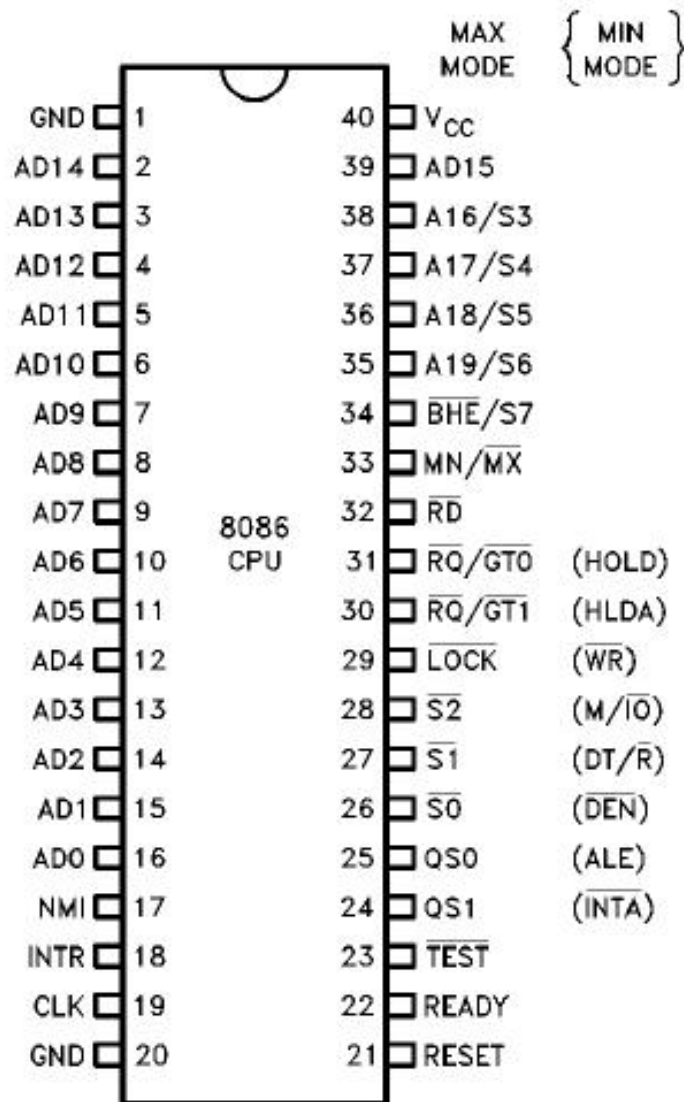
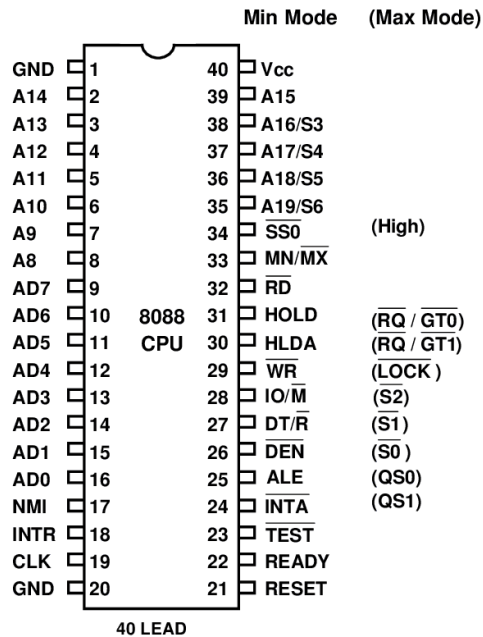


Figure 1.1: Pin Diagram for Intel 8086 Max mode Min Mode



- Nine pins have functions which depend on the state of MN/MX :

MN/MX = **high** -- 8088 operates in MINIMUM MODE

MN/MX = **low** -- 8088 operates in MAXIMUM MODE

- **Minimum mode:** - 8088 directly generates the control signals necessary for accessing memory and IO ports.
- **Maximum mode:-** external support chips are used to generate control signals; the processor can work in a system containing other processors

Figure 1.2: Pin Diagram for Intel 8088 Min mode Max Mode

Pin diagram 8088

1.2 Pin Connections

$AD_7 - AD_0$ • 8088 address / data bus lines

- Multiplexed address data bus
- Rightmost eight bits of the memory address or I/O port no. whenever ALE is active (Logic 1) or data whenever ALE is inactive (Logic 0)
- High impedance state during a hold acknowledge

$A_{15} - A_8$ • 8088 address bus (upper half memory address bits)

- High impedance state during a hold acknowledge.

$AD_{15} - AD_8$ • 8086 address/data bus lines

- Contains address bits $A_{15} - A_8$, when ALE is logic 1
- Enter in high-impedance state whenever a hold acknowledge occurs.

$A_{19}/S_6 - A_{16}/S_3$ • Multiplexed address/ status bus

- Enter in high-impedance during hold acknowledge.

S_6 Always 0

S_5 Indicates the condition of Interrupt flag

S_4, S_3 Indicate segment accessed during current bus cycle

S_4	S_3	Function
0	0	Extra Segment
0	1	Stack Segment
1	0	Code or no segment
1	1	Data Segment

Table 1.1: Segment accessed during current Bus cycle

- \overline{RD}
- Whenever this pin goes to logic 0, the data bus becomes receptive to data from the memory or I/O devices connected to the system.
 - Floats to high impedance state during a hold acknowledge
- READY**
- μp enters into **WAIT** state and remains idle if this pin is at logic 0
 - No effect on operations of μp , if this pin is at logic 1
- INTR**
- Used to request a h/w interrupt
 - If INTR is held high when $IF = 1$, the μp enters an interrupt acknowledge cycle (\overline{INTA} becomes active) after completion of the current instruction
- \overline{TEST}
- An input that is tested by the WAIT instruction
 - If TEST is logic 0, the WAIT instruction functions as NOP
 - If TEST is logic 1, the WAIT instruction waits for **TEST** to become 0
- NMI**
- Non maskable interrupt pin
 - Similar to the **INTR** except that NMI does not check IF (whether it is 1)

Chapter 2

Lecture 2

2.1 Pin Connections Continued

- RESET** • Causes the μp to reset itself if this pin remains high for a minimum of four clocking periods
- whenever the up gets reset , it begins executing instructions at memory location **FFFF0H** and disables future interrupts by clearing IF
- CLK** • Provides the base timing signal to the up
- Clock signal must have at least 33% duty cycle (high for the one-third of the clocking period and low for two-third of the period)
- VCC** • Power supply input
- Provides +5.0 volt with 10% tolerance to the up
- GND** • 2 pins, both must be connected to ground
- MN/ \overline{MX}** • Selects either minimum mode or maximum mode operation of the up
- $\overline{BHE}/S7$** • Bus high Enable
- Used in 8086 to enable the most significant data bus bits (D15 - D8) during a read or write operations
 - The state of S7 is always a logic 1

2.2 Minimum Mode Pins

- IO/ \overline{M} or M/ \overline{IO}** • Selects memory or I/O
- Indicates the μp 's address bus contains either a memory address or an I/O port address

- High impedance state during a hold acknowledge
- \overline{WR}
 - Indicates that the μp is outputting data to a mem or I/O device
 - Data bus contains valid data for memory or I/O during the time \overline{WR} remains 0
- \overline{INTA}
 - A response to the INTR input pin
 - Used to gate the interrupt vector number onto the databus in response to an interrupt request.
- \overline{ALE}
 - Address Latch Enable
 - Indicates that the μp 's address/ data bus contains address information
 - The address can be a mem address or I/O port number
 - [Does **NOT** float during a hold acknowledge]
- DT/\overline{R}
 - Data Transmit or Receive
 - Indicates that the μp 's data bus is transmitting ($DT/\overline{R} = 1$) or receiving ($DT/\overline{R} = 0$) data.
 - Used to enable external data bus buffers.
- DEN**
 - Data bus enable
 - Activates external data bus buffers.
- HOLD**
 - Requests a direct memory access (DMA)
 - If it is a logic 1, μp stops executing S/W and places its address, data and control bus at high impedance state
 - If it is a logic 0, the μp executes S/W normally
- HLDA**
 - Hold acknowledge
 - Indicates that the μp has entered the hold state
- \overline{SSO}
 - Equivalent to SO pin in maximum mode option of the μp
 - It is combined with IO/\overline{M} and DT/\overline{R} to decode function of the current bus cycle

2.3 Bus Control functions

IO/\overline{M}	DT/\overline{R}	\overline{SSO}	Function
0	0	0	Interrupt acknowl- edge
0	0	1	Memory read
0	1	0	Memory write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	I/O read
1	1	0	I/O write
1	1	1	Passive inactive

Table 2.1: Bus cycle status (8088) [Minimum mode]

$\overline{S_2}$	$\overline{S_1}$	$\overline{S_0}$	Function
0	0	0	Interrupt acknowl- edge
0	0	1	I/O read
0	1	0	I/O write
0	1	1	Halt
1	0	0	Opcode fetch
1	0	1	Memory read
1	1	0	Memory write
1	1	1	Passive inactive

Table 2.2: Bus control functions generated by the bus controller 8288 [Maximum mode]

Chapter 3

Lecture 3

3.1 Maximum Mode Pins

For using external coprocessors:

$\overline{S2}, \overline{S1}$ and $\overline{S0}$ • Indicate the function of current bus-cycle

- Normally decoded by 8288 bus controllers

$\overline{R1}/\overline{G1}$ and $\overline{R0}/\overline{G0}$ • Request/grant pins

- Request Direct Memory Access
- Bi-Directional lines
- used to both request and grant a DMA operations

\overline{LOCK} • Used to lock peripherals off the system

$\overline{QS_1}$ and $\overline{QS_0}$ • Queue status bits

- Show status of the internal instructions queue
- Accessed by numeric coprocessor (8087)

$\overline{QS_1}$	$\overline{QS_0}$	Function
0	0	Queue is idle
0	1	First byte of opcode
1	0	Queue is empty
1	1	Subsequent byte of opcode

Table 3.1:

3.2 Clock Generator (8284A)

Basic functions • Clock generation

- **RESET** synchronization
- READY synchronization
- TTL-level peripheral clock signal

3.2.1 Pin diagram

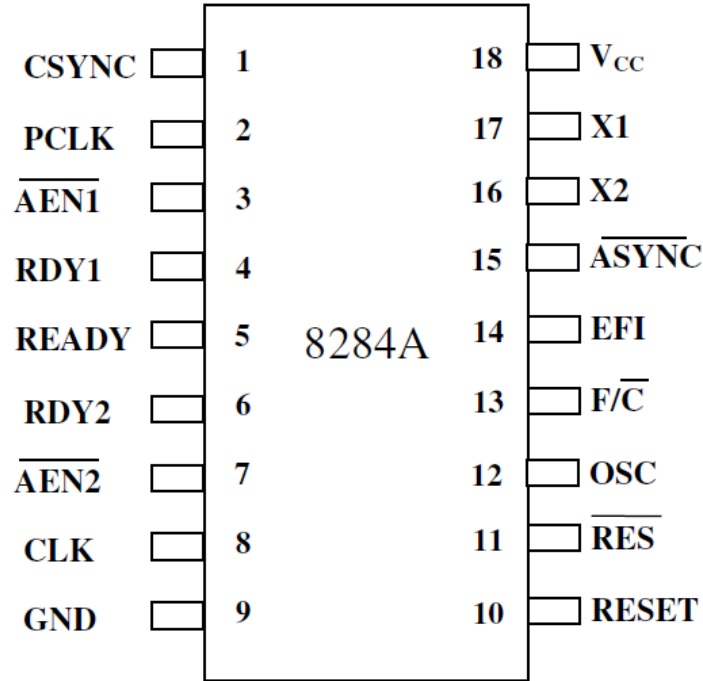


Figure 3.1: Pin Diagram for Intel 8284A

3.2.2 Pin Functions

AEN1 and AEN2 • Qualify the bus ready signals, RDY1 and RDY2 respectively

- wait states are generated by the **READY** pin of μP , which is controlled by $\overline{AEN1}$ and $\overline{AEN2}$ pins

RDY1 and RDY2 • Bus ready inputs

- Cause wait states in conjunction with $\overline{AEN1}$ and $\overline{AEN2}$ pins

\overline{ASync} • Ready synchronization

- Selects either one or two stages of synchronization for RDY1 and RDY2 inputs

READY • An output pin that connects to the μP 's READY input

- Synchronized with RDY1 and RDY2 inputs

X1 and X2 • Crystal oscillator Pins

- Connect to an external crystal, which is used as the timing source for the clock generator and all its functions

F/ \overline{C} • Frequency/ crystal select input

- Chooses the clocking source
- If it is held high, and external clock is provided to the EFI pin.
- If it is held low, the internal crystal oscillator provides the timing signal.

EFI • External frequency input

- Supplies timing whenever F/\overline{C} pin is held high

CLK • Clock output pin, which provides clock input to μP and other components

- Output signal and **one-third** of the crystal or EFI input frequency and has a duty cycle of 33% (as required by 8086/8088)

PCLK • Peripheral clock

- One sixth of the crystal or EFI input frequency, and has a 50% duty cycle.

OSC • Oscillator output

- At the same frequency as the crystal or EFI input
- Provides an EFI input to the other 8284A in a multiprocessor system.

 \overline{RES} • Reset input

- Often connected to an *RC network* that provides power on resetting

RESET • Reset output

- connected to the μP 's RESET input pin

CSYNC • Clock synchronization

- Used whenever the EFI input provides synchronization in a multiprocessor system
- If the *internal oscillator* is used, this pin must be grounded.

3.3 Internal Block Diagram of 8284A Clock Generator

- If a crystal is attached to X1 and X2, the oscillator generates a square wave signal at the same frequency as the crystal
- $CLK = \frac{\text{frequency}}{3}$; $PCLK = \frac{\text{frequency}}{6}$;

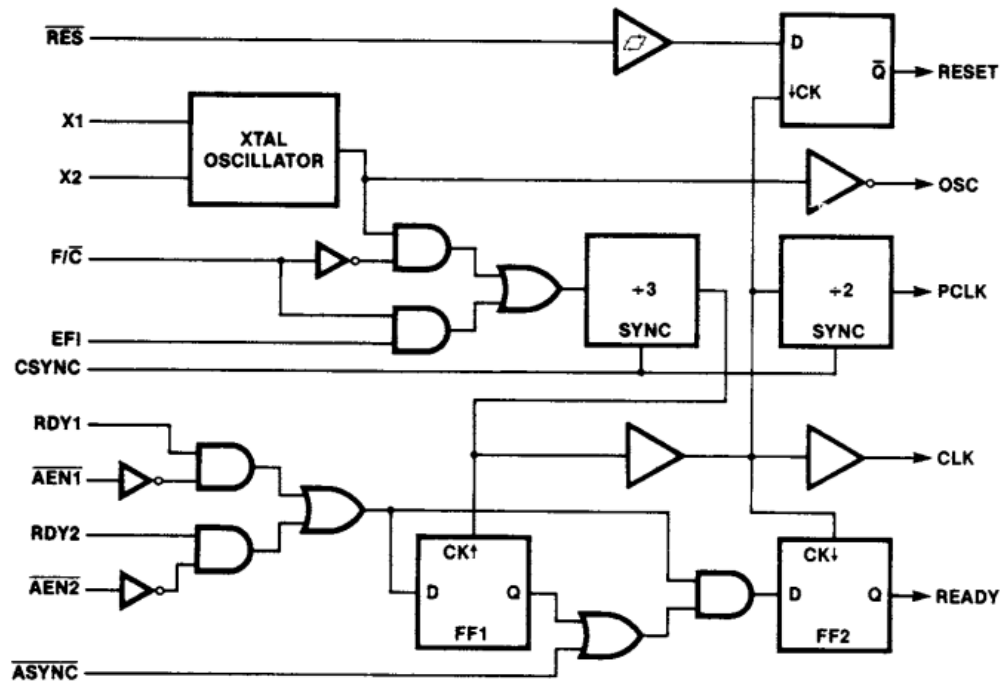


Figure 3.2: Internal Block Diagram of 8284A Clock Generator

3.3.1 Operation of the RESET selection

- \overline{RES} goes through a **Schmitt Trigger** and a D-type flip-flop, which ensures meeting timing requirements of the μP 's RESET input.
- The circuit applies RESET to the μP at the negative edge ($1 \rightarrow 0$), and the μP samples the RESET signal at the positive edge ($0 \rightarrow 1$)
- When power is first applied to the system, the RC circuit provides a **logic 0** to the \overline{RES}
- After a short time, \overline{RES} becomes **logic 1**, as the capacitor charges to the +5V through the resistor
- The push allows the μP to be reset by an operator
- Correct RESET timing requires the RESET input to become a **logic 1** no later than 4 clock cycles after the power is applied, and held high for at least $50\mu s$
- RESET goes high in 4 clock cycles: by FF

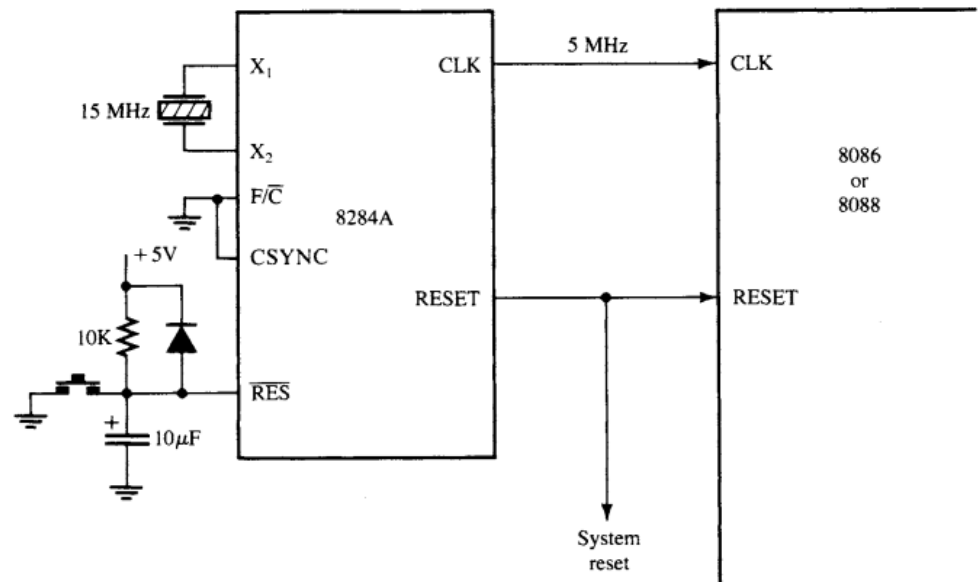


Figure 3.3: The clock generator (8284A) and the 8086 and 8088 microprocessors illustrating the connection for the clock and reset signals. A 15 MHz crystal provides the 5 MHz clock for the microprocessor

- RESET stays high for $50\mu s$: by RC time constant.

Chapter 4

Lecture 4

4.1 Bus Buffering and Latching

- The address / data bus on the μP is multiplexed (shared) to reduce the number of pins, which on the other hand, burdens with the task of extracting or *demultiplexing* info from these pins
- **Why not leave the buses multiplexed?**
 - Memory and I/O require that the address remains valid and stable throughout a read or write cycle. If the busses are multiplexed, the address can get changed causing read or write in wrong locations.
- All computer systems have three types of buses:
 1. **Address bus:** Provides memory address or I/O port numbers
 2. **Data bus:** Transfers data between μP and the memory I/O
 3. **Control bus:** Provides control signals to memory and I/O

4.2 Basic of Demultiplexing

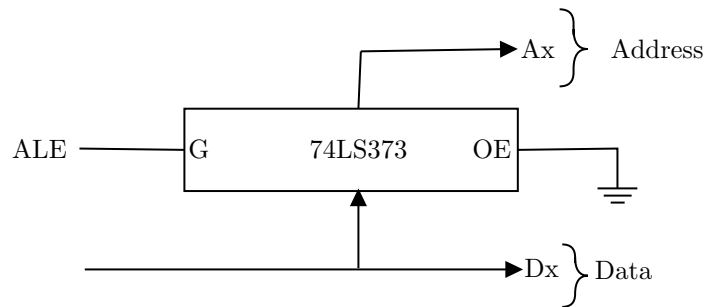


Figure 4.1: Using the 74LS373

4.2.1 Demultiplexing 8088

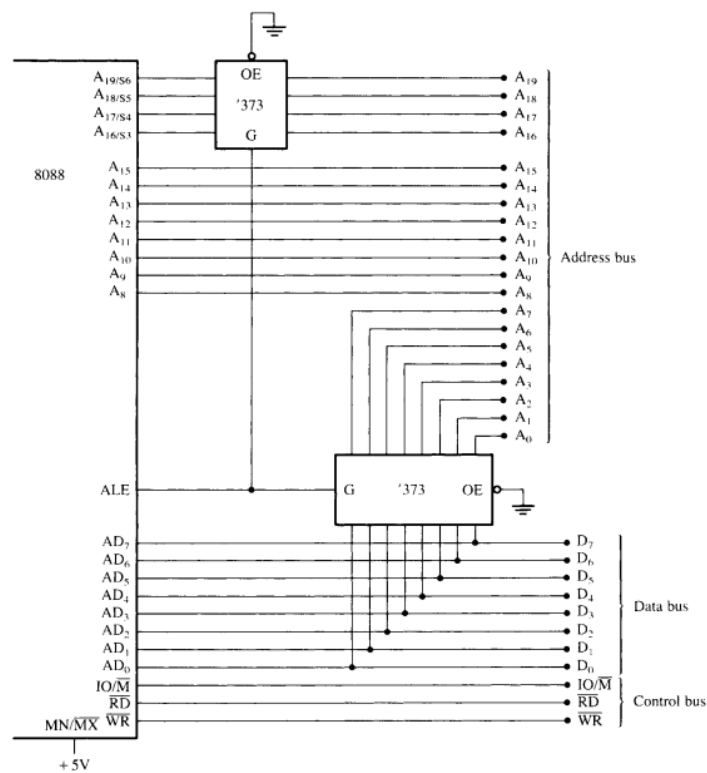


Figure 4.2: The 8088 microprocessor shown with a demultiplexed address bus. This is the model used to build many 8088-based systems.

- **74LS373** latches (OE for output enable and G or LE for latch enable)

inside) are used to demultiplex address/data bus connections and the address/status bus connections.

- **74LS373** passes inputs to outputs like wires when **ALE** is logic 1; when **ALE** is logic 0, the latches remember the inputs at the time of the change to logic 0.

4.2.2 Demultiplexing 8086

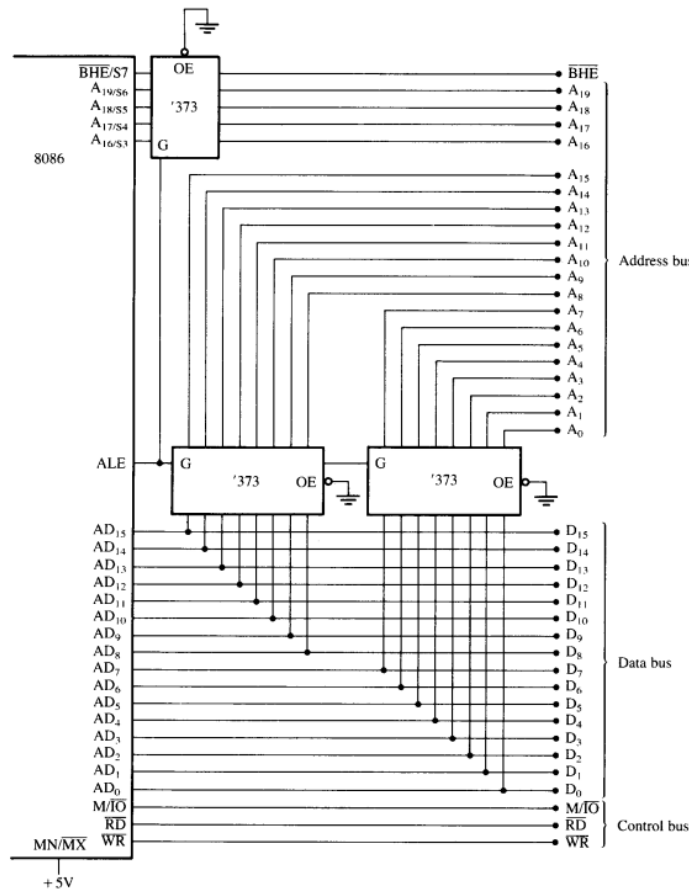


Figure 4.3: The 8086 microprocessor shown with a demultiplexed address bus. This is the model used to build many 8086-based systems.

- Difference from 8088 $AD_{15} - AD_8$ and $\overline{BHE}/S7$ is that \overline{BHE} selects a high-order memory bank in a 16-bit memory system in 8086.

4.3 Buffered System

- If more than 10 unit loads are attached to any bus pin, the entire μP system must be buffered (Buffer provides amplification in a digital circuit to drive output loads enabling more TTL unit loads to be driven)
- The *demultiplexed* pins are already buffered by the **74LS373** latches.
- A fully buffered signal will introduce a timing delay, which causes no difficulty unless memory or I/O devices are used that function at near the maximum speed of the bus.

4.3.1 Fully Buffered 8088

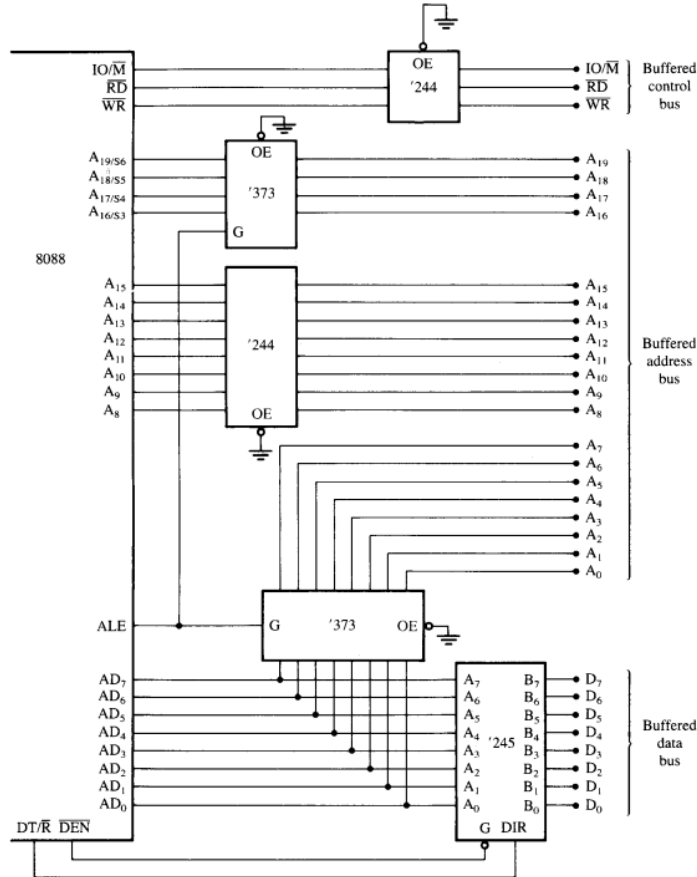


Figure 4.4: A fully buffered 8088 microprocessor.

- 74LS244: Octal buffer
- 74LS245: Octal bidirectional buffer

4.3.2 Fully Buffered 8086

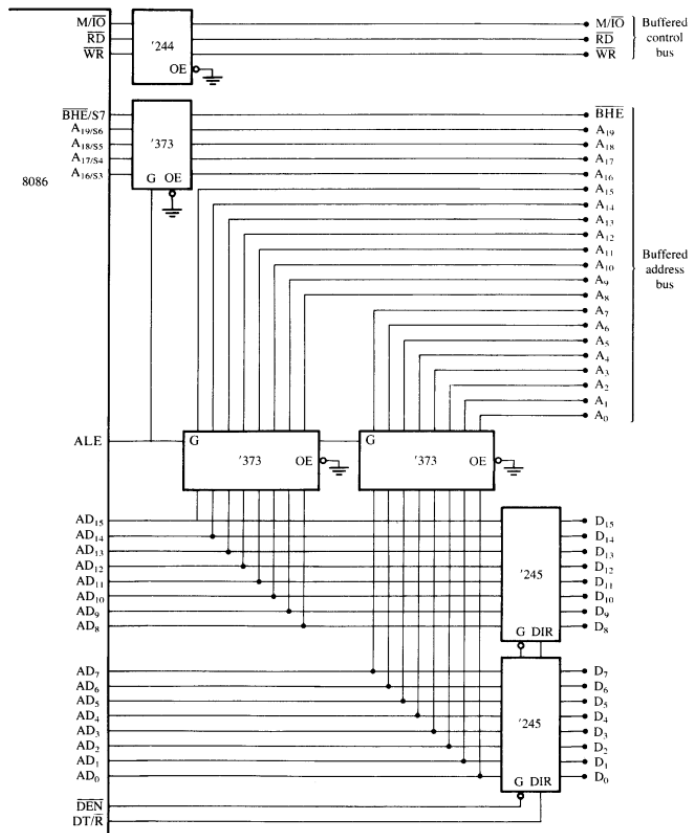


Figure 4.5: A fully buffered 8086 microprocessor.

4.3.3 Number of chips required for fully buffered microprocessor

μP	74LS244 (Octal buffer)	74LS245 (Octal bidirectional buffer)	74LS373
8088	2	1	2
8086	1	2	3

Table 4.1: Number of chips required for fully buffered microprocessor

Part II

Memory Interface

Chapter 5

Lecture 5

5.1 Memory Interface

- Four common types of memory:
 1. Read Only Memory (ROM)
 2. Flash memory (EEPROM)
 3. Static Random Access Memory (SRAM)
 4. Dynamic Random Access Memory (DRAM)
- Pin connections common to all memory devices :
 1. address inputs
 2. data outputs (inputs/outputs)
 3. some type of selection input
 4. Atleast one control input used to select a read or write operations
- Control connections:
 1. **ROM** \rightarrow Only one control input (\overline{OE} [Output Enable] or \overline{G} [Gate])
 2. **RAM** \rightarrow Only one(R/\overline{W}) or two ($\overline{WE}/\overline{W}$ and $\overline{OE}/\overline{G}$) control input. [$\overline{WE}/\overline{W}$ and $\overline{OE}/\overline{G}$ do not get actualued at the same time]

5.1.1 ROM Memory (nonvolatile memory)

- Permanently stores programs that are resident to the system and must **not** change when power supply is disconnected (permanently programmed) for example BIOS.
- **EPROM** (Erasable Programmable ROM): Programmed using a device called EPROM programmer; erasable if exposed to high-intensity UV light for about 20 minutes or less.

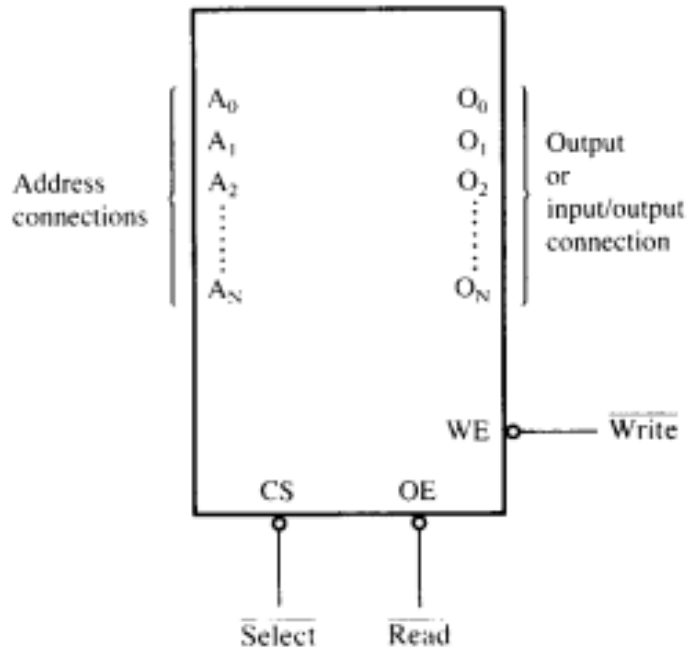


Figure 5.1: A pseudo-memory component illustrating the address, data, and control connections.

- **PROM** (Programmable ROM): Programmed by burning open tiny NI-Chrome or Silicon Oxide fuses; Once programmed, it cannot be erased
- **Read Mostly Memory**(RMM) or **Flash memory** or **EEPROM**(Electrically Erasable Programmable ROM) or **EAROM**(Electrically Alterable ROM) or **NOVRAM**(Non volatile RAM) : Electrically erasable however, needs more time to erase than a normal RAM.

5.1.2 Delays in operation of an EPROM

t_{acc1} : Address to output delay
 t_{OH} : Address to output hold
 t_{CO} : Chip Select to Output delay
 t_{DF} : Chip Deselect to Output float

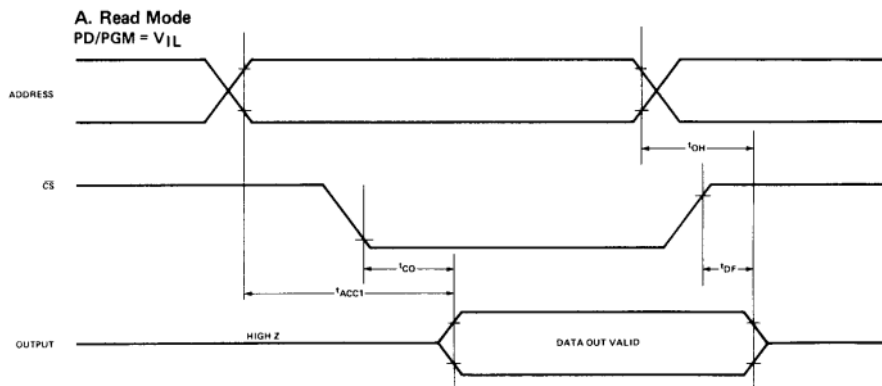


Figure 5.2: The timing diagram of AC characteristics of the 2716 EPROM.

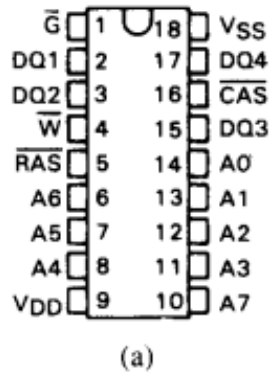
5.1.3 Static memory or static RAM(SRAM) or Volatile memory

- Retain data as long as DC power is applied (no data without power)
- Difference between ROM and RAM :
 - **RAM** \rightarrow written under normal operation
 - **ROM** \rightarrow programmed outside the computer and is normally read.

5.1.4 Dynamic Ram (DRAM)

- DRAM is essentially same as SRAM, except that it retains data for only 2 or 4ms on an integrated capacitor
- After 2 or 4 ms, the contents of DRAM must be completely rewritten (refreshed) because the capacitors (which store Logic 1/0) lose their charges.
- Refreshing also occurs during a write, a read, or during a special refresh cycle.
- Have much larger sizes compared to SRAM. Its obvious disadvantage is requirement of many address pins. To reduce the number of address pins, the address inputs are multiplexed.

Example of multiplexed address pins : 64K x 4 DRAM Here, for 64K, 16



PIN NOMENCLATURE	
A0-A7	Address Inputs
\overline{CAS}	Column Address Strobe
DQ1-DQ4	Data-In/Data-Out
\overline{G}	Output Enable
\overline{RAS}	Row Address Strobe
V_{DD}	+ 5-V Supply
V_{SS}	Ground
\overline{W}	Write Enable

(b)

Figure 5.3: The pin-out of the TMS4464, 64K x 4 dynamic RAM (DRAM).

address pins are required. However, 8 address pins are used through multiplexing.

- At first, A0 - A7 are stored in internal row latch as row address through enabling \overline{RAS}
- Next, A8-A15 are stored in internal column latch as column address through enabling \overline{CAS}

Address multiplexer of 64K x 4

- If \overline{RAS} is 0, then A pins get connected and A0-A7 are stored in the internal Row Address latch
- *Internal Row Address Latch* is edge triggered, and therefore the row address gets captured into the latch before the address changes to column address.

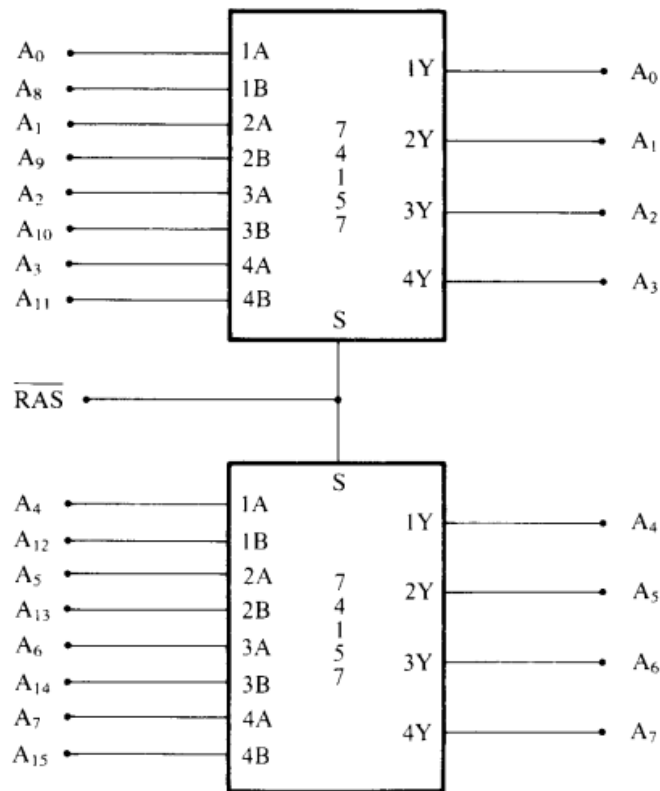


Figure 5.4: Address multiplexer for the TMS4464 DRAM.

- If \overline{RAS} is 1, the the 8 pins get connected and A8-A15 are stored in *Internal Column Address Latch*

Chapter 6

Lecture 6

6.1 Address Decoding

- Decoding address sent from μP is necessary, as without it, only one memory device can be connected to a μP
- Another reason of decoding is possible mismatch (or difference) between the number of connections in μP and memory.

Example: a 2K x 8 EPROM (2716) has 11 address pins, whereas the μP 8088 has 20 address pins.

6.1.1 Simple NAND Gate Decoder

- Here, the 2K EPROM is decoded at memory address locations **FF899H** - **FFFFFFH**. (The FF8 corresponds to 1's in left nine positions)
- In such a decoding, one NAND gate decoder selects one 2K EPROM out of many 2K EPROMs that appear in 1M address space.
- The obvious disadvantage is that each EPROM needs one NAND gate decoder

6.1.2 3-to-8 Line Decoder(74LS138)

- **Input:** Enable = 001 and Selection(ABC) = n
- **Input:** $n^{th} output = 0$; all reset are 1
- **Input:** Enable \neq 001
- **Input:** All outputs are 1

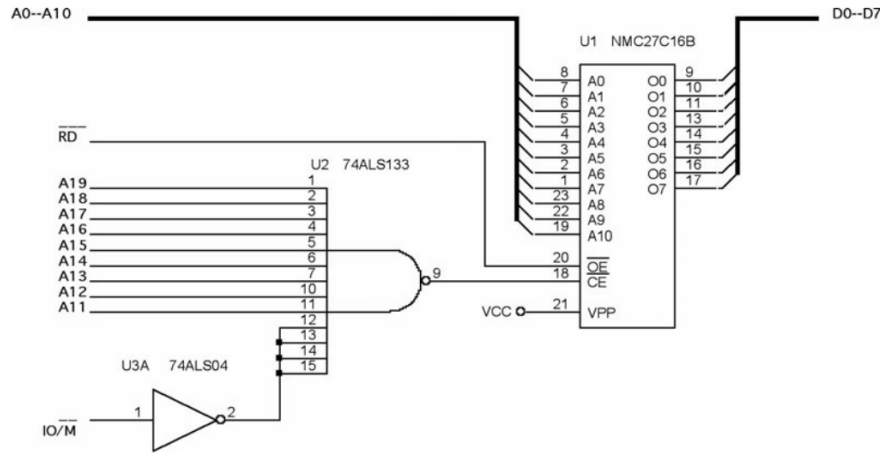


Figure 6.1: A simple NAND gate decoder that selects a 2716 EPROM for memory location FF800H–FFFFFFH

- A16-A19 must have 1 for activating 74LS138. Therefore, all addresses begin with "1111" at the left.
- Eight 2764 EPROMs (each 8K x 8, having 13 pins) are decoded over address space **F0000H–FFFFFFH**(64K x 8)

6.2 PROM Address Decoder

Bipolar PROM is used because of its larger number of input connections, which reduces the number of other circuits required in the decoding system.

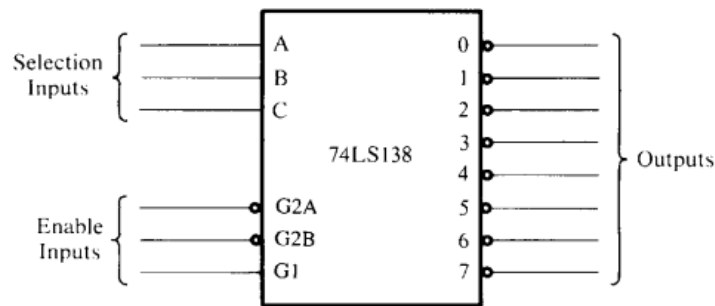
Example: 82S147 (512 x 8) PROM has 10 input connections and 8 output connections. Among the input connections, 9 are used for memory addressing (of PROM) with 1 control (G) input.

Here, the control input (\overline{G} or \overline{CE}) must be grounded, as if this PROM's outputs float to their high-impedance state, one or more of the PROMs might be selected by noise impulse of the system.

- Main advantage of using PROM is that address map can be easily changed.
- As the PROM comes with all the locations programmed as logic 1, only eight of the 512 locations must be programmed.

6.3 PLD programmable Decoders

There are three **Programmable Logic Devices**(PLDs) that function in the same manner:



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

Figure 6.2: The 74LS138 3-to-8 line decoder and function table.

1. PLA (Programmable Logic Array)
 2. PAL (Programmable Array Logic)
 3. GAL (Gated Array Logic)
- These devices can be programmed by blowing fumes to establish connections.
 - The decoding circuit using a PLD is similar as that with a PROM, where the PROM gets replaced by the PLD (programmed with the logic representing the input-output pattern realized by PROM, as shown in a table in the last decoding)
 - No control or selection logic is required hence.

- The address space (F8000H - FFFFFH) includes the upper 32K bytes of memory containing FFFF0H, where 8088 starts executing instructions after a hardware reset. FFFF0H location is often called the "cold start" location. S/W stored at the FFFF0H location would contain a JMP instruction at FFFF0H that jumps to F8000H so the remainder of the program can execute.

7.1.2 Interfacing RAM to the 8088

- Most RAMs **do not** require wait states
- Interrupt vectors (often modified by S/W packages) reside in RAMs.

In the Figure 7.1, 16 *62256*(32K x 8) static RAMs are interfaced to 8088 beginning at memory location 00000H to 7FFFFH(512K x 8). Here, two decoders select from 1 different RAM components and a third to select the other decodes for appropriate memory selection.

- U4 selects the other two decoders. U3 selects addresses beginning with 00 and U9 selects addresses beginning with 01. Extra pins of U4 remain for future extension.
- All address, data and control (\overline{RD} and \overline{WR}) are buffered. Buffering is important when many devices appear in a single system. Excessive load without buffering can cause logic 0 output to rise above 0.V, maximum allowed in a system.

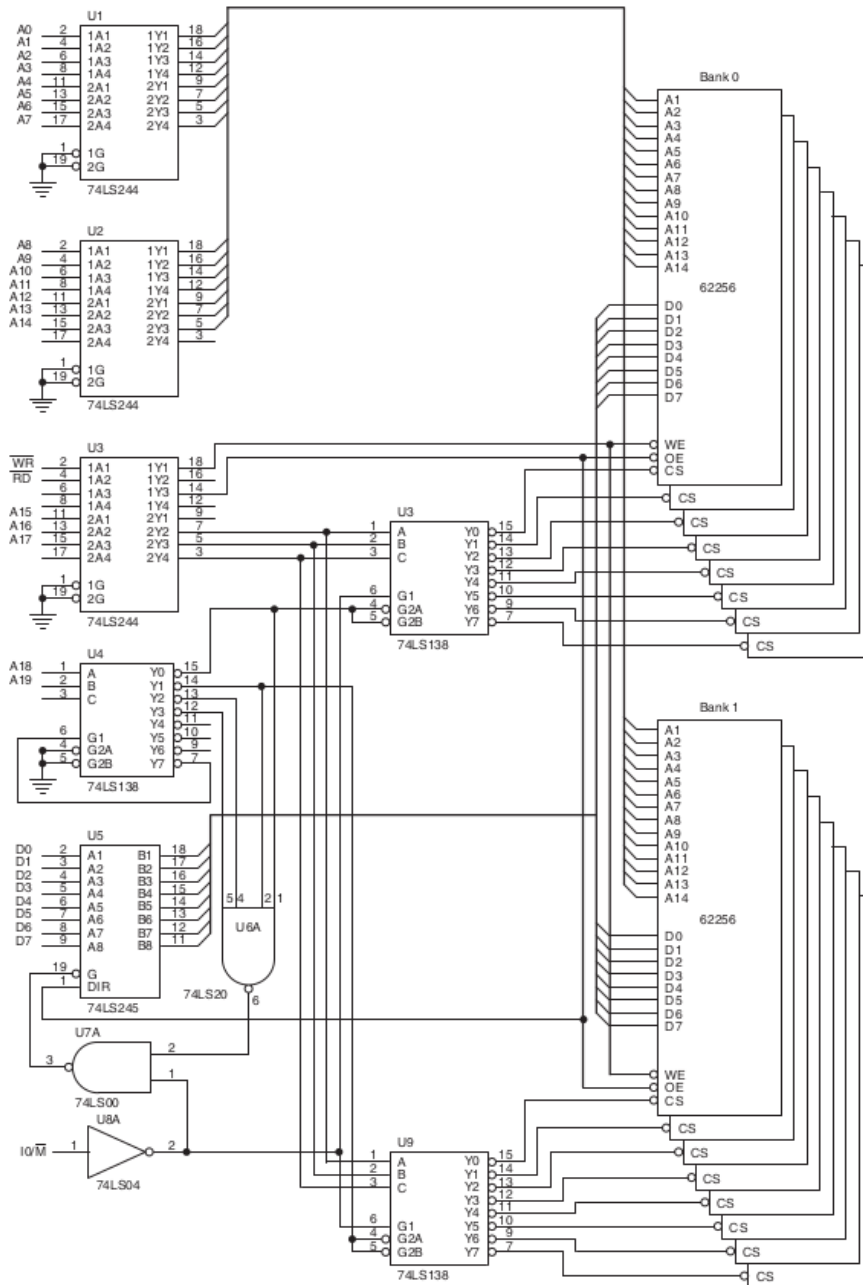


Figure 7.1: A 512K-byte static memory system using 16 6225 SRAMs.

Chapter 8

Lecture 8

8.1 Interfacing Flash Memory

Flash Memory requires a 12V programming voltage to erase or write new data. The 12V can be available either at the power supply or through 5V-to-12V converter.

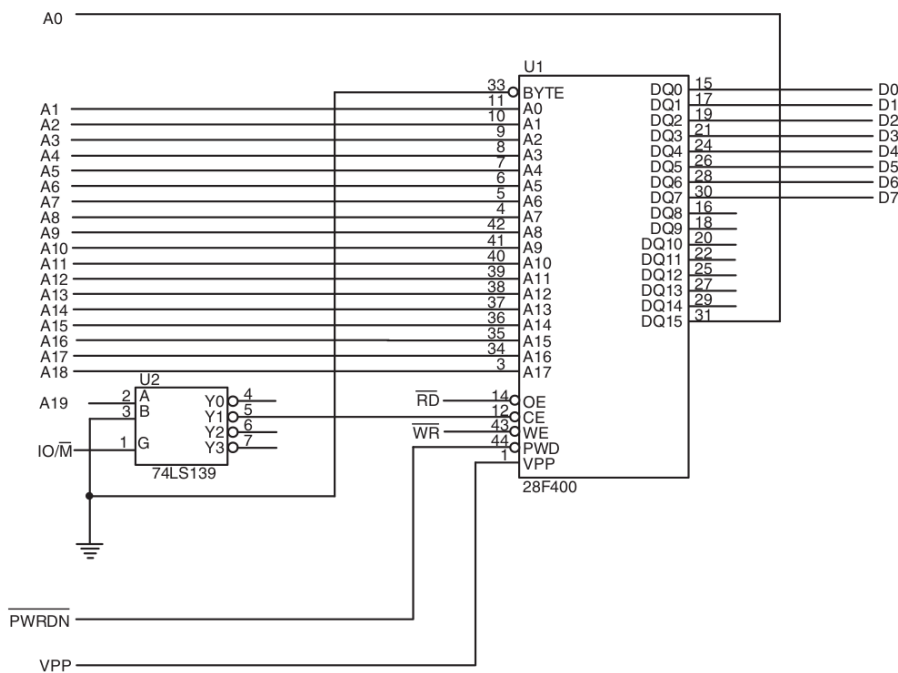


Figure 8.1: The 28F400 flash memory device interfaced to the 8088 microprocessor.

- **28F400** can be used either as a 512K x 8 or as a 512K x 16 memory device. For interfacing with 8088, it is used as a 512K x 8
- New pins in flash memory compared to SRAM :
 1. VPP, which is connected to 12V for erasing and programming
 2. \overline{PWD} , which selects power down mode when a logic 0 is applied and also used for programming.
 3. BYTE, which selects byte(0) or word(1) operation.
- Pin **DQ15** functions as the least-significant address input when operated in byte mode
- Flash memory is much slower than SRAM (can need around 10^7 times more time).
- The single decoder (**74LS139**) uses address connection A19 and IO/\overline{M} as inputs (location in the example : **80000H-FFFFFFH**)

8.2 Parity for memory error detection

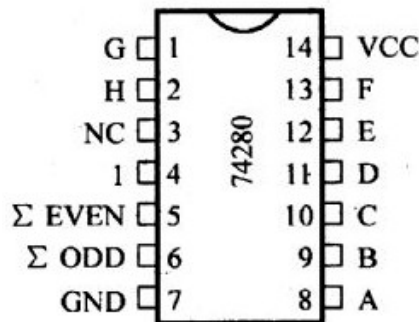
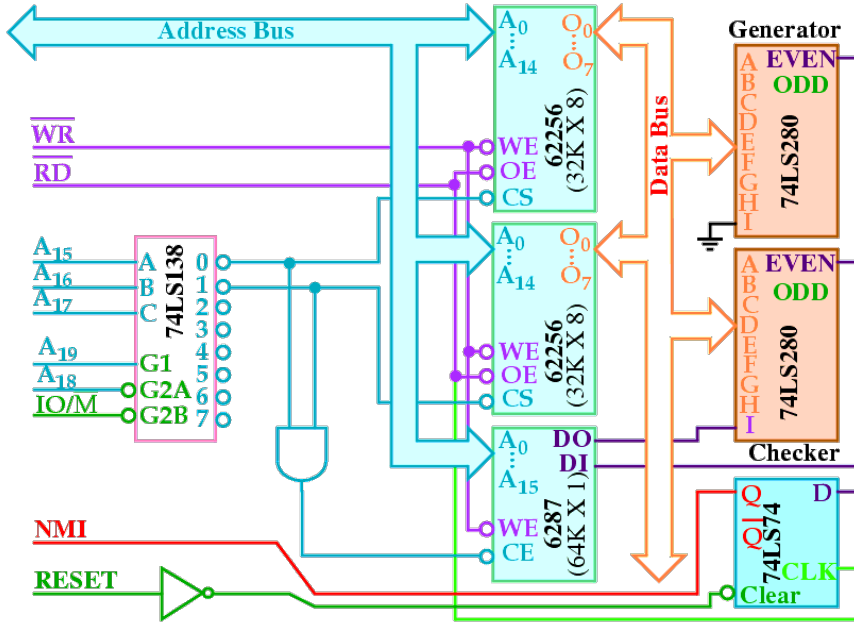


Figure 8.2: 9-bit parity generator/decoder

Input: Number of Logic 1's in A-I (9 pins)	Outputs	
	$\sum Even$	$\sum Odd$
0,2,4,6,8	H	L
1,3,5,7,9	L	H

Table 8.1: Number of chips required for fully buffered microprocessor

Example below shows 64K x 8 static RAM using two 62556 (32K x 8) SRAMs having a parity bit stored in 6287 (64k x 1) SRAM generated by 74AS280. Decoded memory space is at **80000H-8FFFFH**. In Figure 8.2 as I of first



74LS280 is grounded, a 1 is stored in the parity RAM **6287**. If an even number of 1's appear in the data bus (connected to A-H). Thus, including the stored parity bit, an odd parity is stored for each byte.

- **Parity SRAM:** No \overline{OE} connection for \overline{RD} . It reads data from output pin when selected (\overline{CE} is enabled) and writes data when selected with $\overline{WE} = 0$.
- If overall parity is odd (everything is okay), the even parity output of

second **74LS280** will become logic 1. The parity output is connected to 8088's NMI.

- Data read from the memory are settled to the final state before an NMI input (error detection) occurs through being timed by a D flip-flop
- The DFF latches output of the parity checker **74LS280** at the end of an \overline{RD} cycle on the memory.

8.3 Error correction

- **74LS636** : An -bit error correction and detection circuit that corrects any single bit memory read error and flags any 2-bit error.
- Corrects error by storing five parity bits with each byte of memory data.
- If more than two bits are in error(rare), the circuit ,may **not** detect it.
- Whenever a memory component fails completely, its bits are all high or all low. In this case, the circuit flags the processor with a multi-bit error detection.
- 8 data pins, 5 check bit pins, 2 control pins(S_0 and S_1), and 2 error outputs (**Single Error Flag**(SEF) and **Double Error Flag**(DEF))

S_0	S_1	Function	SEF	DEF
0	0	Write check word	0	0
0	1	Correct data word	*	*
1	0	Read data	0	0
1	1	Latch data	*	*

Table 8.2: Control inputs for 74LS636

*** : These levels are determined by the type of error**

- When a single error is detected, the **74LS636** goes through an error correction cycle:
It places 01 on S_0 and S_1 by causing a wait and then read following error correction.
- Difference in connection in memory components :
 \overline{S} is grounded, which enable data to be accessed from memory before \overline{RD} goes low.

- On the next negative edge of the clock after an \overline{RD} , the 74LS636 checks **SEF**. If a single bit error is detected, a correction cycle causes the single-bit error to be corrected. If a double-bit error is detected, **DEF** generates an NMI.

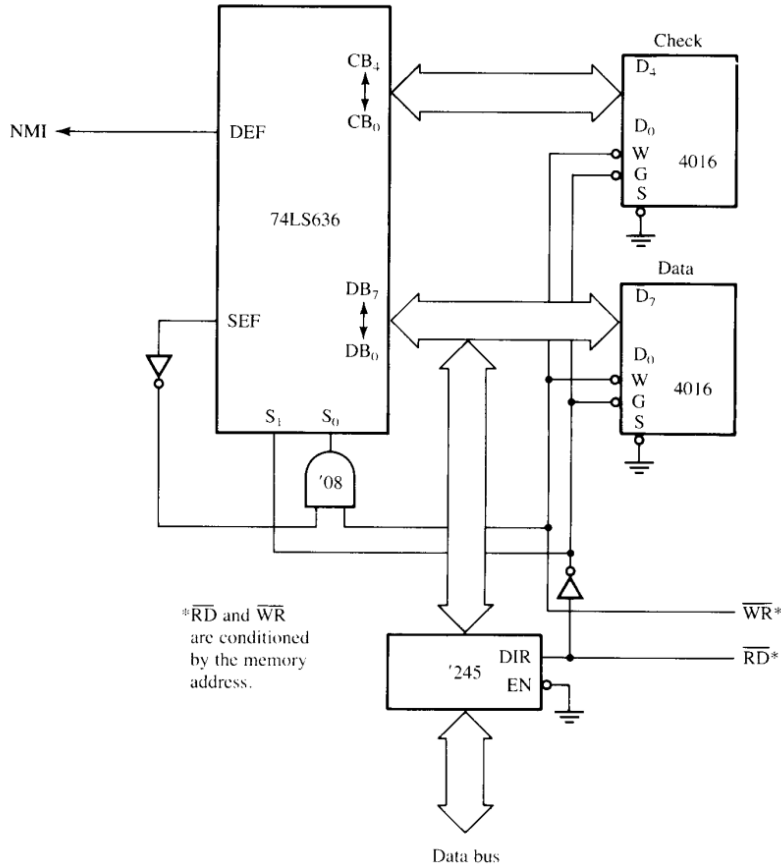


Figure 8.3: An error detection and correction circuit using the 74LS636.

Chapter 9

Lecture 9

9.1 8086, 80186, 80286, 80386 (16-bit) memory interface

- **80286/80386:** 24-bit address bus. \overline{MRDC} and \overline{MWTC} instead of \overline{RD} and \overline{WR} .
- These μP s must be able to write data to any 16-bit or 8-bit location. Therefore, the 16-bit data bus must be divided into two separate sections (banks), which are 8-bit wide.

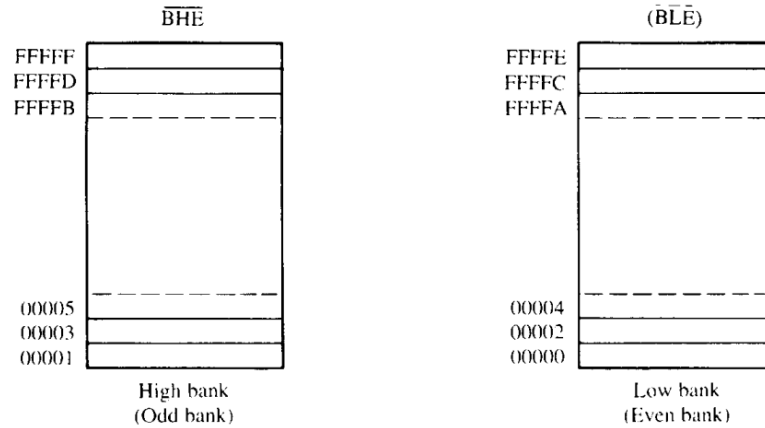


Figure 9.1: The high (odd) and low (even) 8-bit memory banks of the 8086/80286/80386SX microprocessors.

\overline{BHE}	\overline{BLE}	Function
0	0	Both banks enabled for a 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 bank enabled

Figure 9.2: Memory bank selection using \overline{BHE} and \overline{BLE} (A_0).

Bank selection is accomplished in two ways:

1. A separate write signal is developed to select a write to each bank (least costly)
2. Separate decoders are used for each bank
 - Two 74LS138 decoders are used to select 64K RAM for 80386 μP (24-bit address)
 - An enable pin (G2A) of U3[First 74LS138] is enabled by \overline{BHE} and of U2[Second 74LS138] is enabled by \overline{BLE} (A_0).

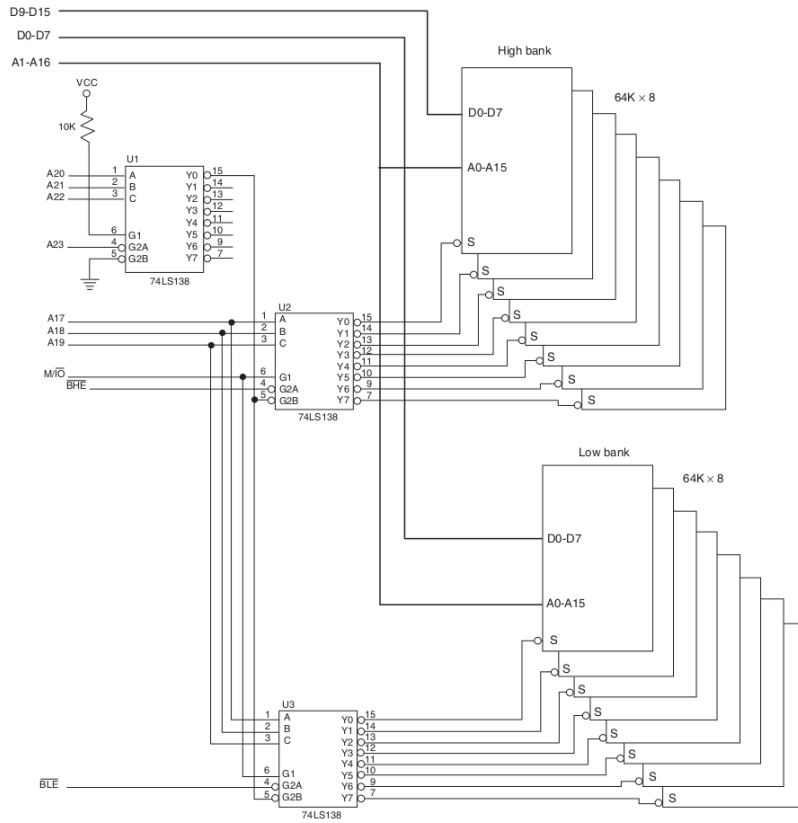


Figure 9.3: Separate bank decoders.

- U3 controls enabling the high bank, and U2 controls enabling the low bank.
- Decoded memory location range is **000000H-0FFFFFFH**(1M).
- A_0 (or \overline{BLE}) is **not** connected to the memory to ensure having separate memory address (odd or even) in separate banks. If it would be connected to the memory address pin, the half of the memory will be wasted. [A_0 is **NOT** even a pin in 80386 μP]

9.2 8086 (16-bit) memory interface

Separate bank write strobes for memory: \overline{WR} combines with A_0 for low bank selection (\overline{LWR}), and \overline{BHE} for high bank selection (\overline{HWR})
 For 80286 and 80386: \overline{MWTC} is used instead of \overline{WR}

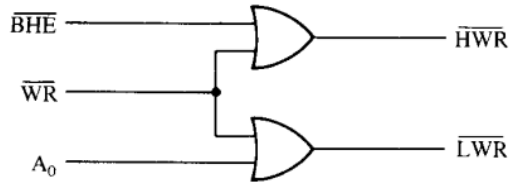
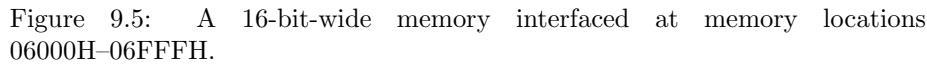


Figure 9.4: The memory bank write selection input signals: \overline{HWR} (high bank write) and \overline{LWR} (low bank write).

Why not also generate read strobe for each memory bank ?

- It is usually unnecessary as 086, 80286, 80386 μP s read only the byte of data they need at any given time ignoring the other parts without causing any problem.
- **Example:** 16-bit memory stored at locations 060000H-06FFFFH for 80286 or 80386 μP using PAL
- PAL logics:
 1. $\overline{CS} = \overline{A23} \cdot \overline{A22} \cdot \overline{A21} \cdot \overline{A20} \cdot \overline{A19} \cdot \overline{A18} \cdot \overline{A17} \cdot \overline{A16}$
 2. $\overline{LWR} = \overline{MWTC} \cdot \overline{A_0}$
 3. $\overline{HWR} = \overline{MWTC} \cdot \overline{BHE}$

Extend this concept of separate banking for all subsequent design (32, 64, ... bits) * SELF STUDY *****



- \overline{RD} is connected to all \overline{OE} inputs (enabling all 16-bits while reading).
- \overline{LWR} and \overline{HWR} are connected to different banks of RAM. Here, both \overline{LWR} and \overline{HWR} go low. For 8-bit eriting, either of them goes low. Such writing is **not** needed for EPROM
- A 74LS139 (dual 2-to-4 decoder) is used to select EPROM with one half and RAM with the other half. It decodes memory as 1-bit wide (not 8-bit).
- EPROM's decoder signal is sent to 8086 wait state generator.

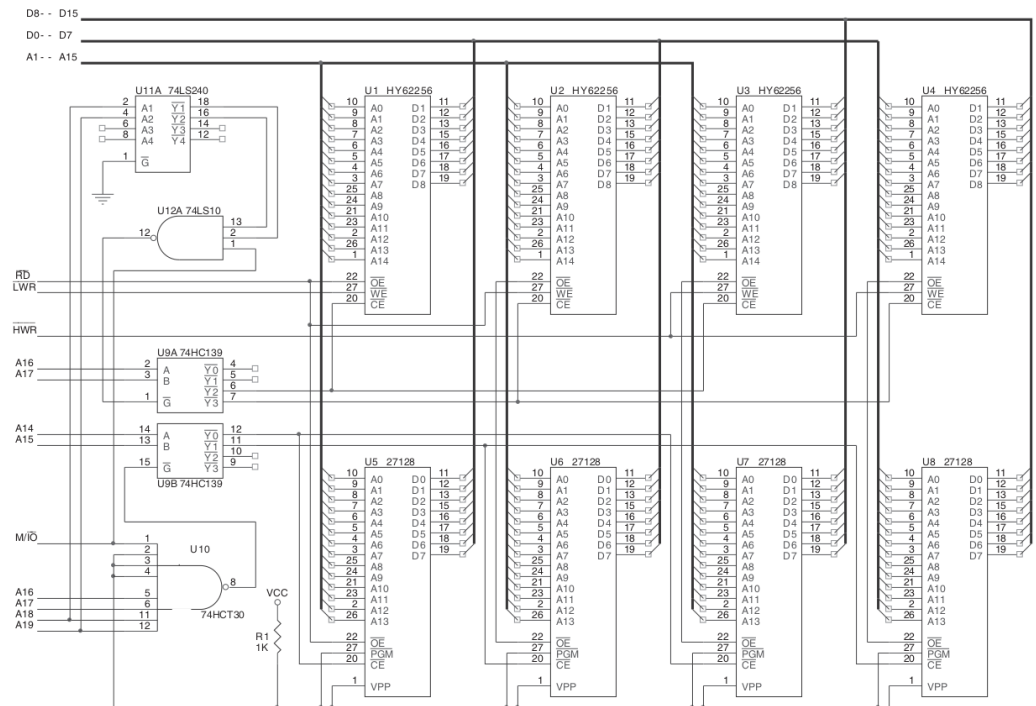


Figure 9.6: A memory system for the 8086 that contains a 64K-byte EPROM and a 128K-byte SRAM.

Chapter 10

Lecture 10

10.1 DRAM

- Needs refreshing periodically (every 2-4ms) as capacitor stores data and they lose their charge. Refreshing \rightarrow read/write/Internal (hidden refresh, transparent refresh/ cycle stealing).
- DRAM's internal organization contains a series of rows and columns. A 256K x 1 DRAM has 25 columns, each containing 256 bits, or eow organized into four section of 64 bits each.

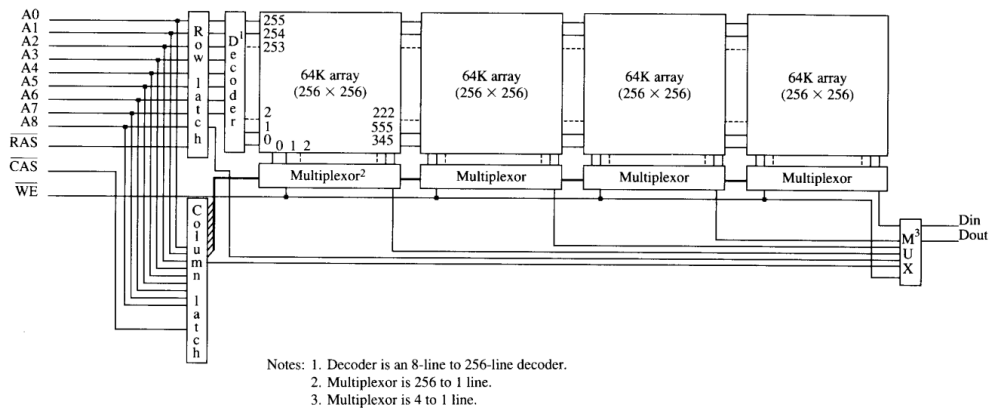


Figure 10.1: The internal structure of a 256K x 1 DRAM. Note that each of the internal 256 words are 1024 bits wide.

- For refreshing 256 rows in 4ms, refresh cycle is $15.6 \mu s$
- For 8086/8088(for example), clock rate is 5 MHz with 00ns for rd/wr.

So, $1 \text{ rd/wr} \approx 800 \text{ ns}$ and $1 \text{ refresh cycle} \approx 1.5 \mu\text{s}$
 So, $1 \text{ refresh cycle} \approx \frac{15.6 \mu\text{s}}{800 \text{ ns}} = 19 \text{ rd/wr}$
 So, \log of 5% ($\approx \frac{1}{19}$) compared to computer time, this is a small price.

10.2 EDO memory (Extended Data Output)

- A slight modification of DRAM, where for any memory access, (including a refresh), stores the 25 bits selected by \overline{RAS} into latched.
- The latches hold the next 256 bits of information, so that data becomes available without wait in case of sequential execution
- Increases system performance by 15-25%

10.3 SDRAM (Synchronous Dynamic RAM)

- At the beginning of rd/wr, it behaves like a standard RAM with the same number of wait states.
- Subsequent (second, third, and fourth) accesses need 1c.c. each
- Therefore to read 4 64-bit numbers, a total of 3(for 1st access) + 1 + 1 + 1 (for next 3 accesses) = 6 cc. are required. For DRAM, it is 3+3+3+3 = 12cc.
- 10% performance increase over EDO.

10.4 DRAM controllers

- Performs address multiplexing and generation of DRAM control signals
- **Example:** 82C08 - Contains an address multiplexer to multiplex a 1-bit address onto 9-bit address connections for 256K memory. It generates \overline{CAS} and \overline{RAS} for DRAM. These signals are developed internally by $CLK, \overline{S_1}$ and $\overline{S_2}$. It takes $\overline{S_1}$ and $\overline{S_0}$ as its \overline{RD} or \overline{WR} inputs. the $\overline{AACK}/\overline{XACK}$ provides an acknowledgement output that is used to indicate ready condition of the μP (normally connected to READY of μP)

Example: 82C08 connected to a series of four 256K x 8 bit memory composing 1M

- 82C08 is connected to a series of four 256K x 8 bit memory
- U3 and U5 form the high bank, and U4 and U form the low bank.
- PAL 16L8 combines \overline{WE} and A_0 to generate write signals for U4 and U6 and combines \overline{WE} and \overline{BHE} to generate write signals for U3 and U5

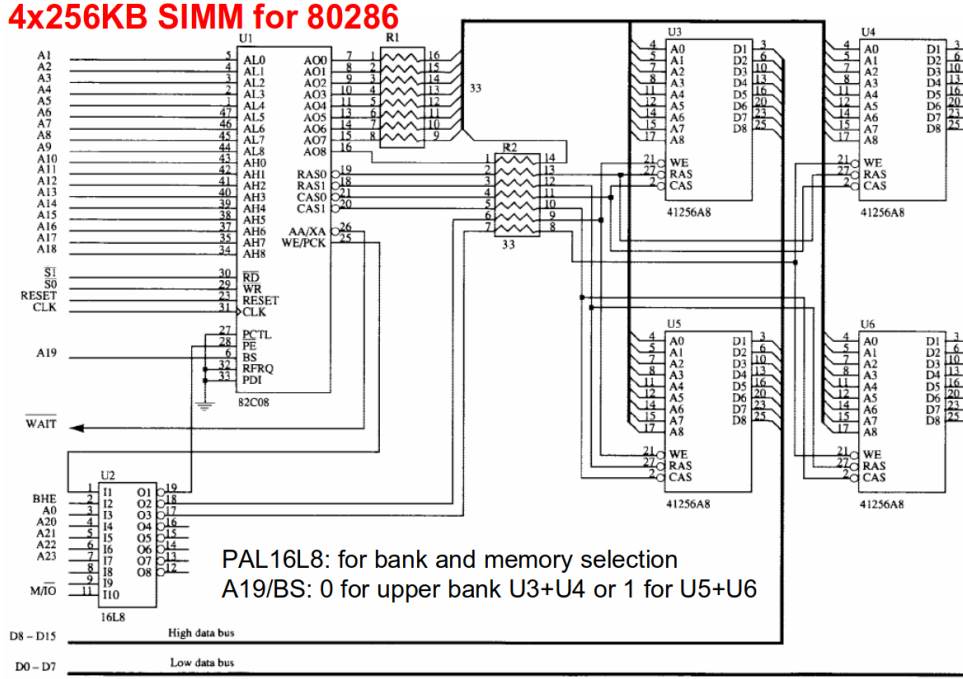


Figure 10.2: A 1M-byte memory system using four 256K SIMM memory devices and the 82C08 DRAM controller. This section of memory is decoded at locations 000000H-0FFFFFFH by the PAL 1L8

- PAL also develops the controller selection signal(\overline{PE}) by combining M/\overline{IO} and other address lines A20-A23
- Memory locations 000000H-0FFFFFFH
- A19 selects the upper bank (U3 and U4) or the lower bank through the BS (Bank Select)input to 82C08.
- PAL logic:

$$1. \overline{HWR} = \overline{BHE} \cdot \overline{WE}$$

$$2. \overline{LWR} = \overline{A_0} \cdot \overline{WE}$$

$$3. \overline{PE} = \overline{A_{20}} \cdot \overline{A_{21}} \cdot \overline{A_{23}} \cdot \overline{MIO}$$

Part III

Basic I/O Interface

Chapter 11

Lecture 11

11.1 Isolated and memory-mapped I/O

Isolated I/O Separate I/O addresses called ports

- **Advantage:** User can expand the memory to its full size.
- **Disadvantage:** New instructions (IN,OUT etc) are required for data transfer.

Memory-mapped I/O Uses any instructions that transfer data between microprocessor and memory (*Advantage*)

- \overline{IORC} and \overline{IOWC} have no functions reducing circuit for decoding (*Advantage*)
- **Disadvantage:** Reduces the amount of memory available to an application.

SELF-STUDY:

1. Basic Input Output Interface
2. I/O port address decoding (Similar as memory)

11.2 Programmable Peripheral Interface

- 82C55**
- low-cost interfacing component having 24 pins for I/O that are programmable in *groups* of 12 pins
 - It operates in three modes of operations
 - it is compatible with any TTL-compatible I/O device
 - Requires insertion of WAIT states in case of operating with a μP having more than 8 MHz clock

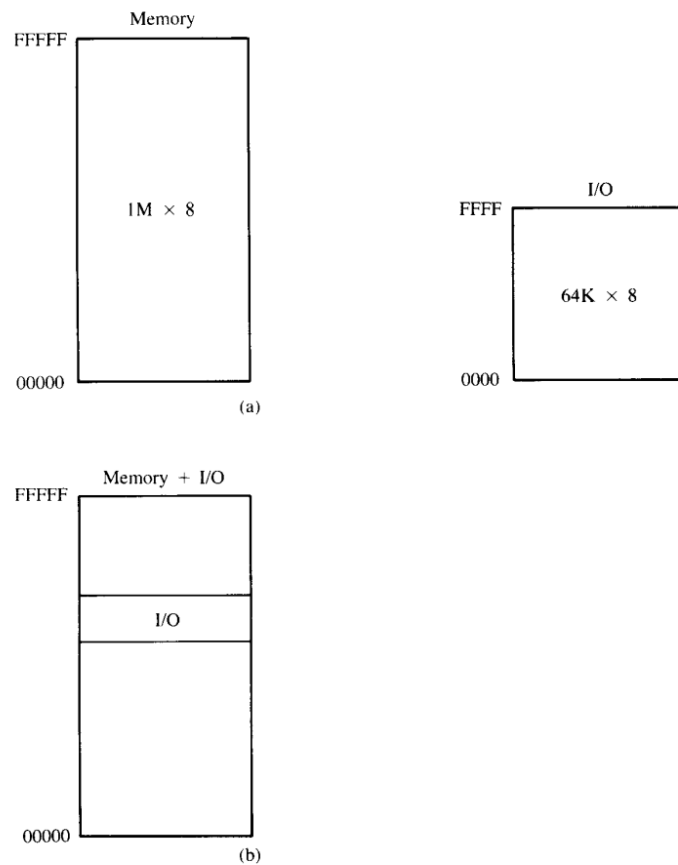


Figure 11.1: The memory and I/O maps for the 8086/8088 microprocessors. (a) Isolated I/O. (b) Memory-mapped I/O.

Basic description of 82C55

- Has 3 I/O ports (labeled as A, B and C)
- Programmed as groups
 1. Group A: Port A[PA7 - PA0] and upper half of Port C[PC7 - PC4]
 2. Group B: Port B[PB7 - PB0] and lower half of Port C[PC3 - PC0]
- RESET input causes all ports to setup as simple input ports using Mode 0 operation (as setup as input, damage is prevented when power is first applied)

A_1	A_0	Function
0	0	Port A
0	1	Port B
1	0	Port C
1	1	Command Register

Table 11.1: Register selection by A_1 and A_0

11.2.1 Programming 2C55: Command Registers

- **Group A** \rightarrow 0/1/2 mode
Group B \rightarrow 0/1 mode
- **Mode 0** \rightarrow SimpleI/O
Mode 1 \rightarrow Strobed I/O [Double handshaking I/O]
Mode 2 \rightarrow Bidirectional Handshaking I/O

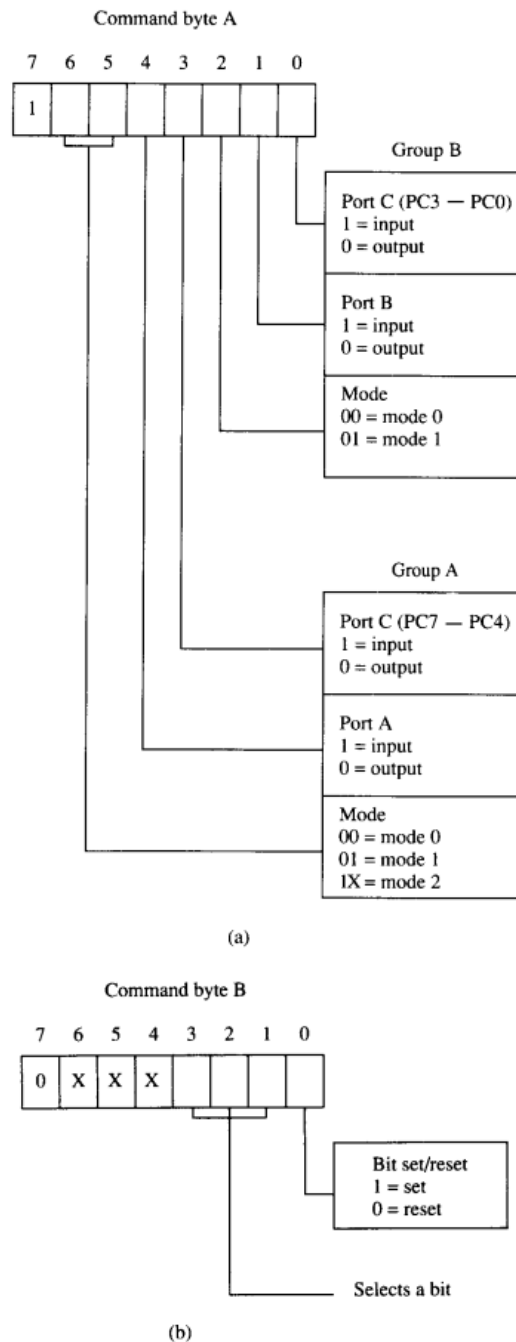


Figure 11.2: The command byte of the command register in the 82C55. (a) Programs ports A, B, and C. (b) Sets or resets the bit indicated in the select a bit field.

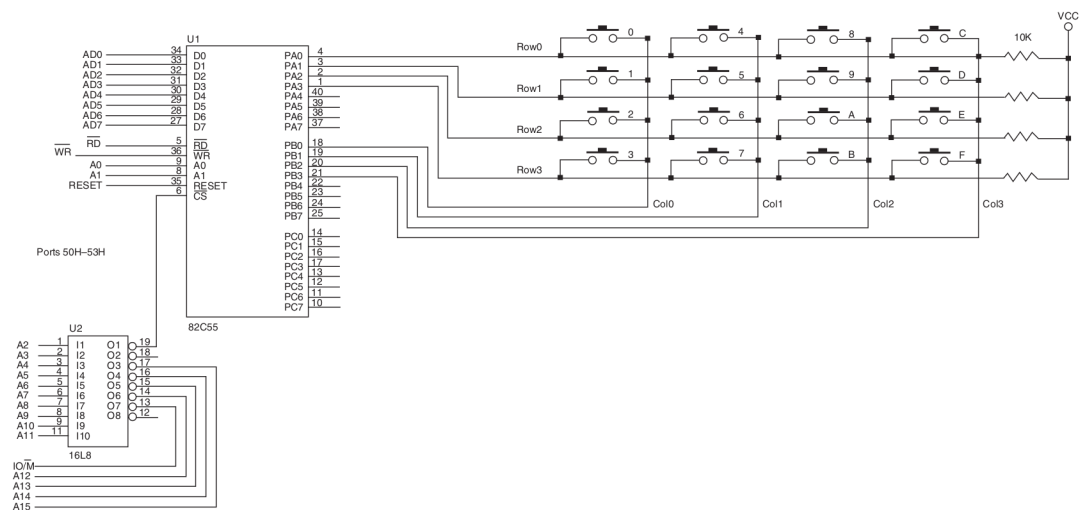
4. Different assembly code (**SELF-STUDY**)**12.1.1 Mode 0 Operation: Keyboard Matrix interface**

Figure 12.2: A 4 x 4 keyboard matrix connected to an 8088 microprocessor through the 82C55 PIA.

- **Port A:** Input
- **Port B:** Output
- One 0 is put in Port B (Ex:0111); Column with 0 gets selected. Any key pressing in that column gives 0 to corresponding row
- Have logic for debouncing (Flow Chart)

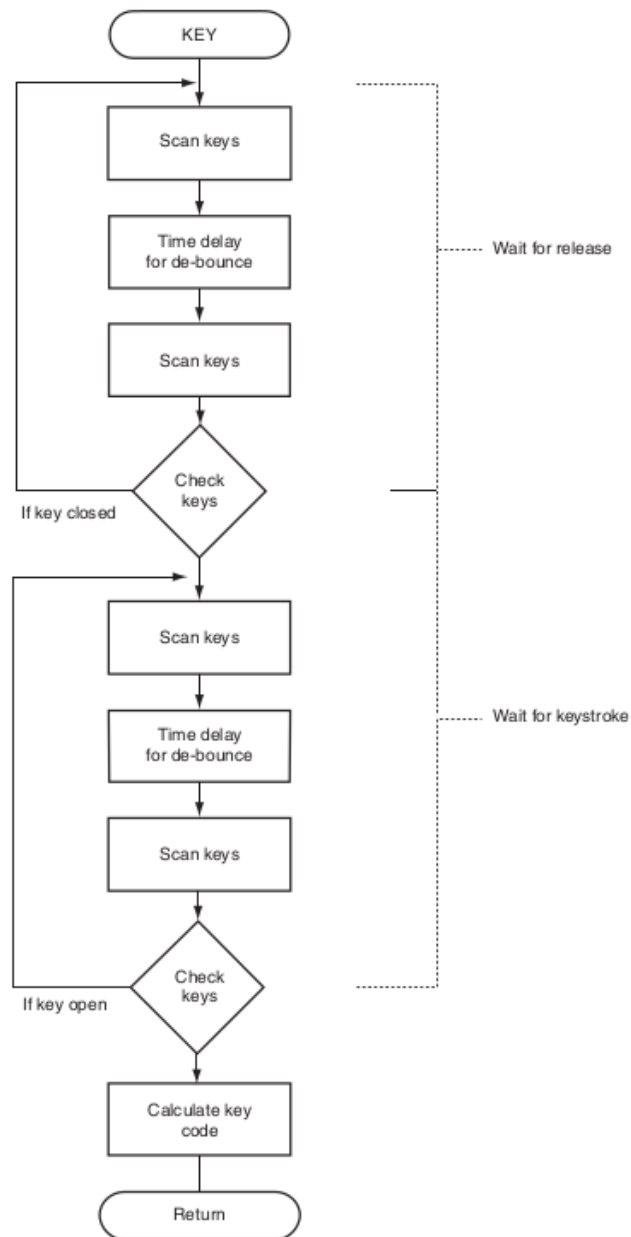


Figure 12.3: The flowchart of a keyboard-scanning procedure.

12.2 Mode 1: Strobed Input

Allows external data to be stored into the port until the microprocessor is ready to receive it.

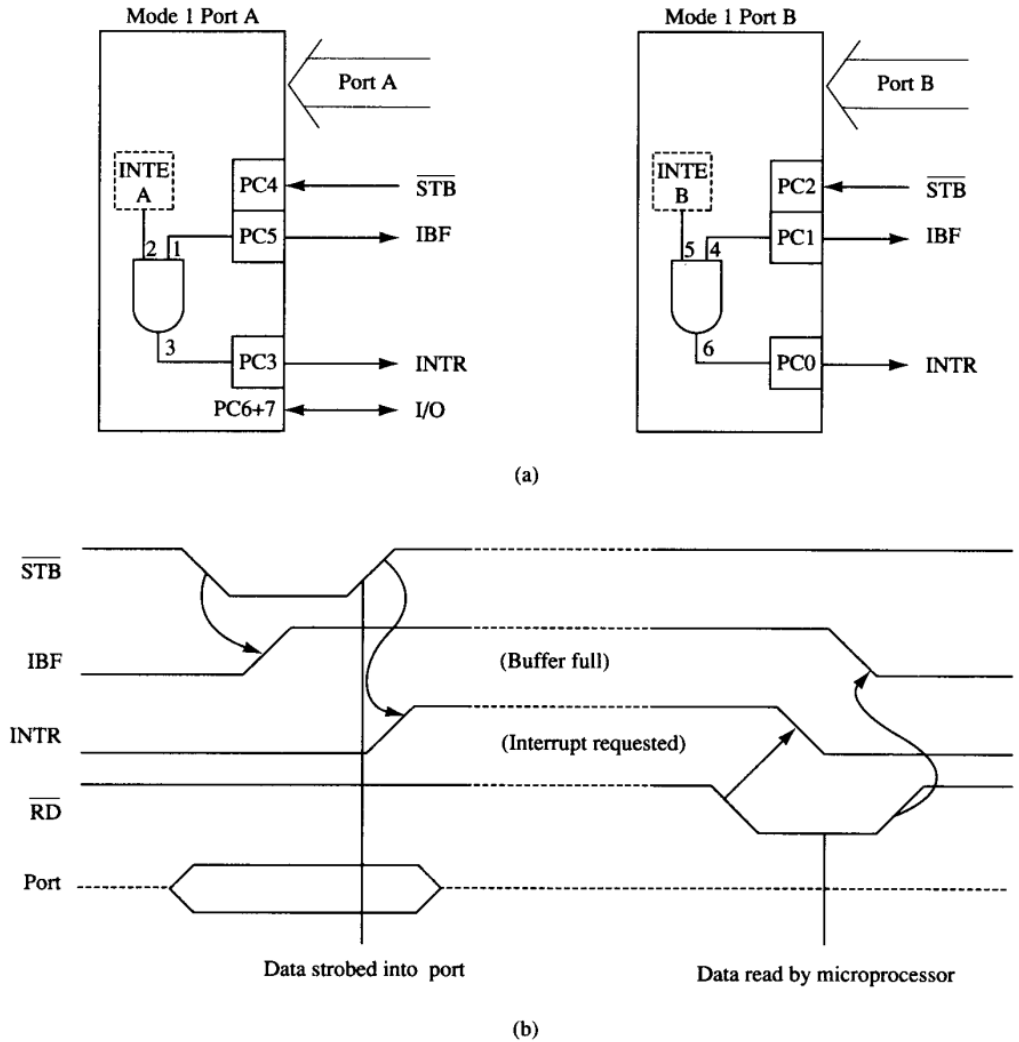


Figure 12.4: Strobed input operation (mode 1) of the 82C55. (a) Internal structure and (b) timing diagram.

\overline{STB} The **strobe** input loads data into the port latch, which holds the information until it is input to the microprocessor via the IN instruction.

IBF **Input buffer full** is an output indicating that the input latch contains information.

INTR **Interrupt request** is an output that requests an interrupt. The INTR pin becomes a logic 1 when the \overline{STB} input returns to a logic 1, and is

cleared when the data are input from the port by the microprocessor.

INTE The **Interrupt enable** signal is neither an input nor an output; it is an internal bit programmed via the port PC 4 (port A) or PC 2 (port B) bit position.

PC 6,7 The port C pins 7 and 6 are general-purpose I/O pins that are available for any purpose.

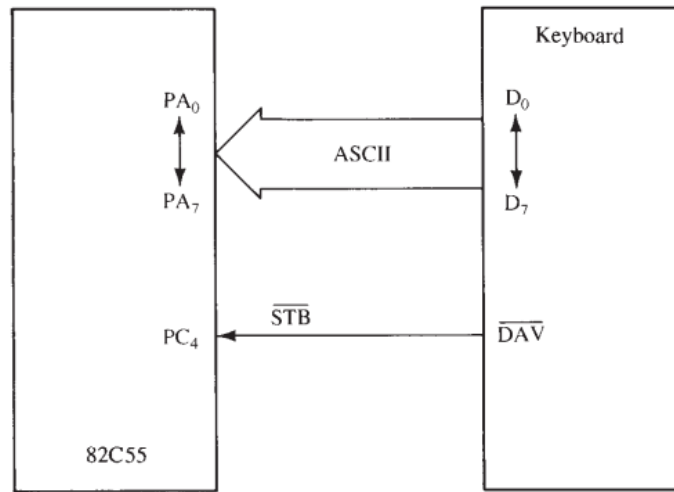


Figure 12.5: Using the 82C55 for strobed input operation of a keyboard.

- The keyboard encoder debounces keyswitches, and gives \overline{STB} if a key is pressed.
- \overline{DAV} (Data available) is activated for $1.0\mu s$ each time a key is typed (Gives \overline{STB})

12.3 Mode 1: Strobed Output

\overline{OBF} Output buffer full is an output that goes low whenever data are output (OUT) to the port A or port B latch. This signal is set to a logic 1 whenever the ACK pulse returns from the external device.

\overline{ACK} The acknowledge signal causes the OBF pin to return to a logic 1 level. The ACK signal is a response from an external device, indicating that it has received the data from the 82C55 port.

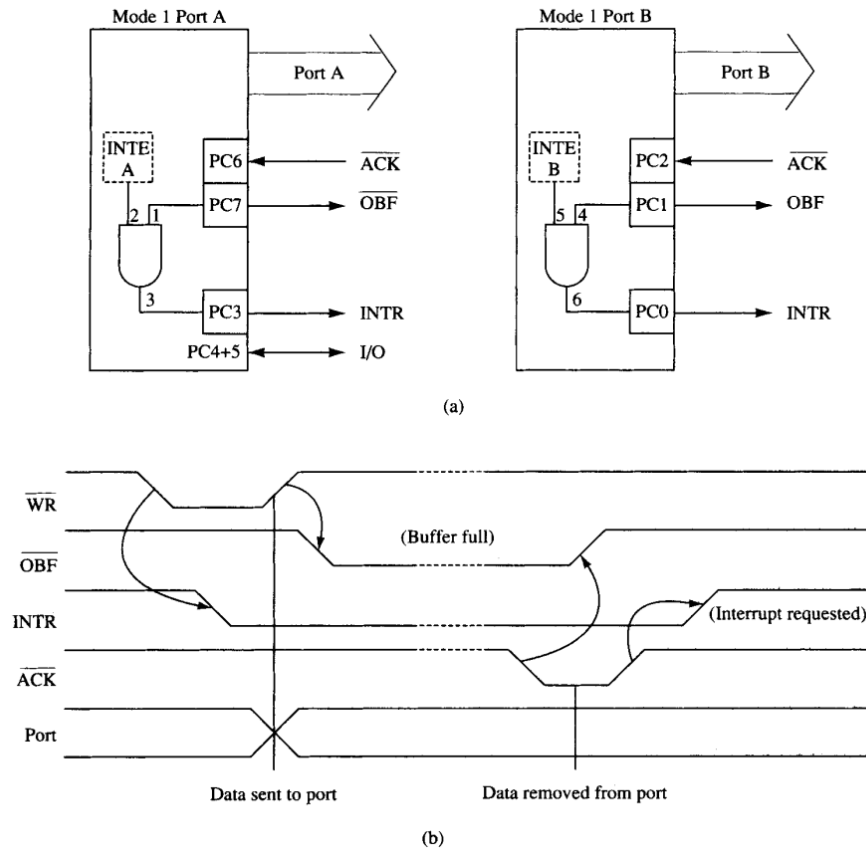


Figure 12.6: Strobed output operation (mode 1) of the 82C55. (a) Internal structure and (b) timing diagram.

INTR Interrupt request is a signal that often interrupts the microprocessor when the external device receives the data via the ACK signal. This pin is qualified by the internal INTE (interrupt enable) bit.

INTE Interrupt enable is neither an input nor an output; it is an internal bit programmed to enable or disable the INTR pin. The INTE A bit is programmed using the PC 6 bit and INTE B is programmed using the PC 2 bit.

PC_4, PC_5 Port C pins PC 4 and PC 5 are general-purpose I/O pins. The bit set and reset command is used to set or reset these two pins.

12.3.1 Strobed Output: Printer

- Port B is connected to a parallel Printer

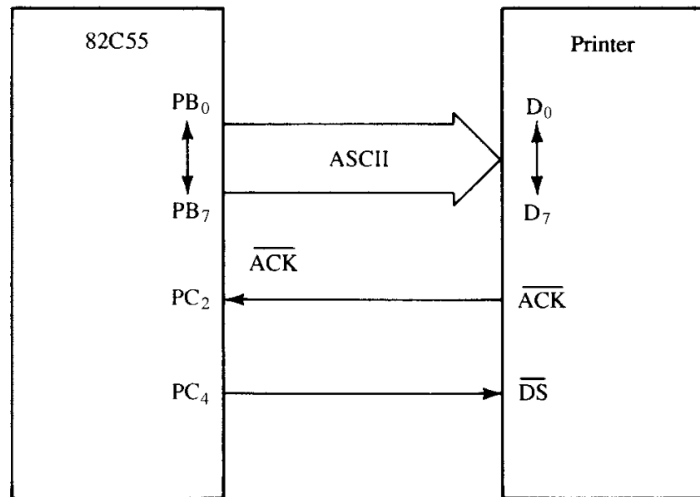
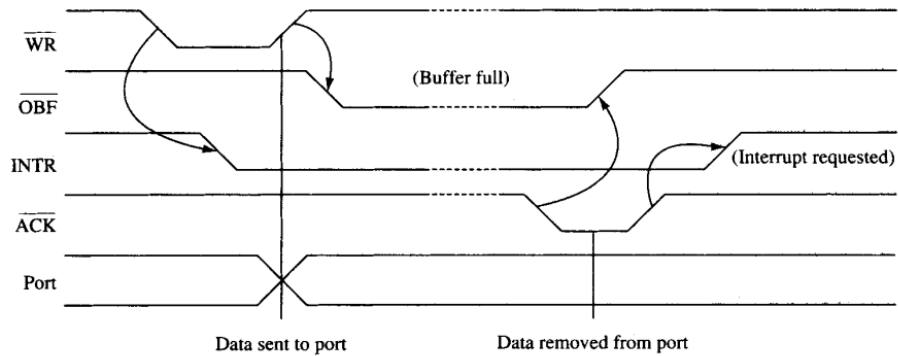
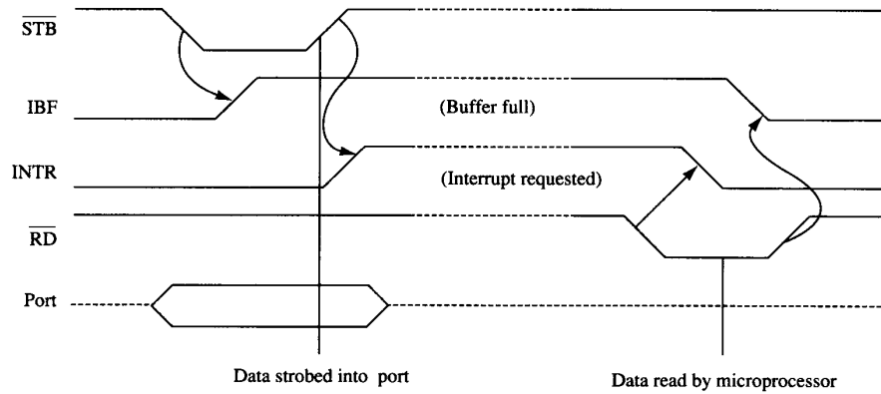


Figure 12.7: Strobed output operation (mode 1) of the 82C55. (a) Internal structure and (b) timing diagram.

- No signal to generate the \overline{DS} signal to the printer. So, PC_4 is used with S/W generated \overline{DS} .

12.3.2 Comparison between Mode 1 Strobed Input and Output



12.4 Mode 2: Bidirectional Operation

Group A only; Allows both transmit and receive over the same wires. Useful when interfacing two computers.

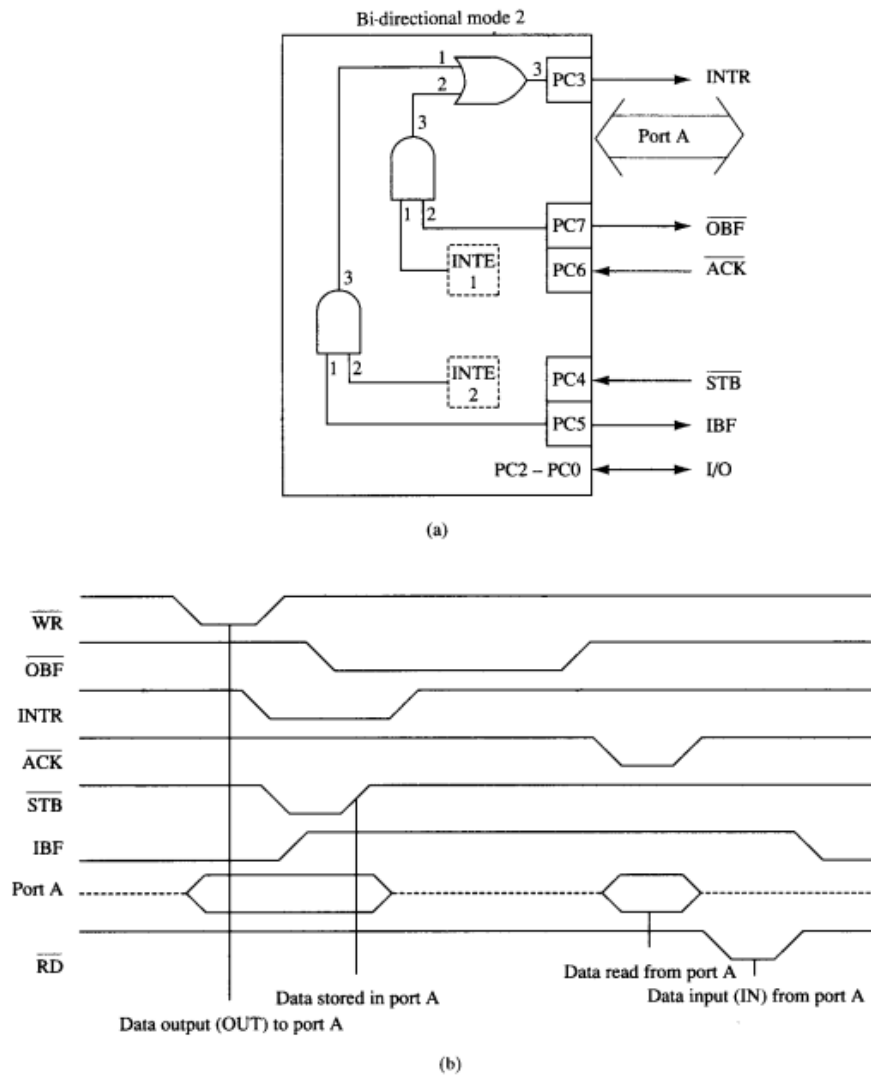


Figure 12.8: Mode 2 operation of the 82C55. (a) Internal structure and (b) timing diagram.

Part IV

Interrupt

Chapter 13

Lecture 13

13.1 Interrupt

- An interrupt is a Hardware initiated procedure that interrupts whatever program is currently executing.
- Useful when interfacing with I/O devices that provide or require data at relatively low data transfer rate.
- Two H/W pins for requesting interrupts: INTR and NMI.
One H/W pin for acknowledging interrupt requested by INTR : INTA
S/W interrupts: INT, INT0, INT3 and BOUND
Two flag bits: IF (Interrupt flag) and TF (Trap flag)
A special return instruction: IRET (or IRETD in 80386, 486,..)
- **Interrupt vector:** Contains address (Segment, offset) of the interrupt service procedure
- **Interrupt vector Table:** Located in the first 1024 bytes of memory at address 000000H - 0003FFH. It contains 256 different 4-byte interrupt vectors.
- First 32 interrupt vectors : Reserved
- Other 224 interrupt vectors : User defined interrupt vectors

Examples of dedicated Interrupts:

Type 0 Divide Error → Division overflow or divide by zero

Type 1 Single step or Trap → Occurs after execution of each instruction.
If **IF** flag is set. Upon accepting this interrupt, IF is cleared to execute interrupt service procedure executes at full speed.

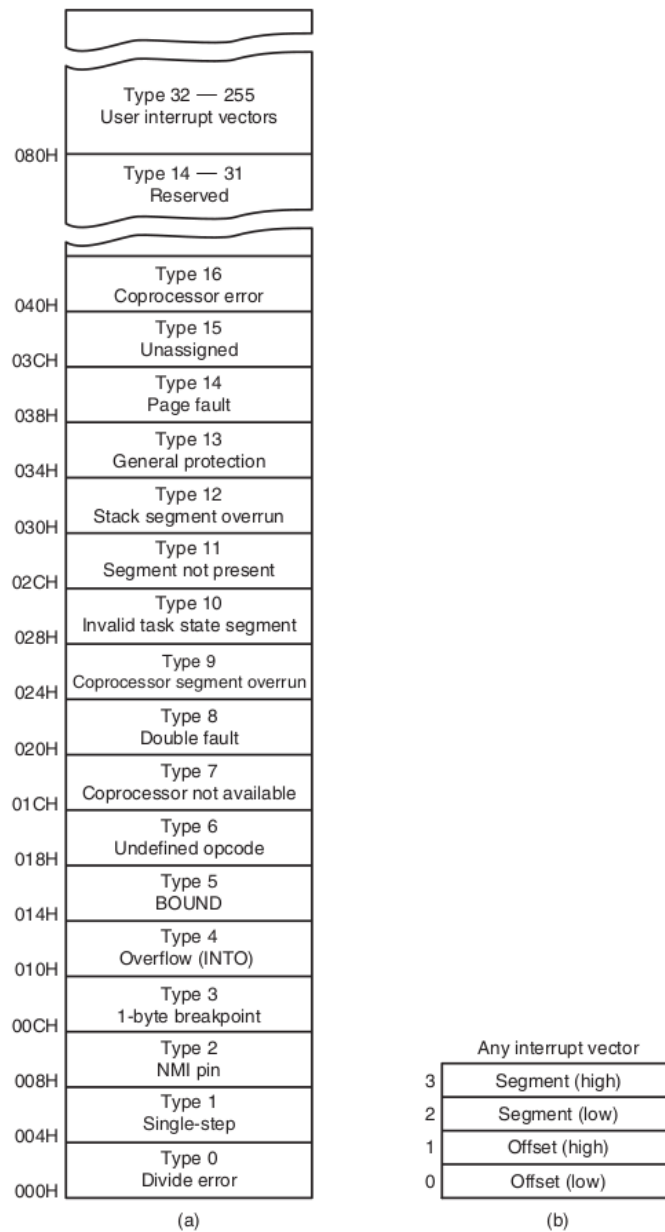


Figure 13.1: (a) The interrupt vector table for the microprocessor and (b) the contents of an interrupt vector.

Type 2 NMI \rightarrow If 1 is placed in the NMI pin; it cannot be disabled

Type 3 One-Byte Interrupt \rightarrow INT3; Often used to store a breakpoint in a program for debugging.

Type 4 Overflow \rightarrow INT0 ; Interrupts if an overflow condition exists as reflected by Overflow Flag(OF)

Type 5 Bound \rightarrow Compares a register with boundary stored in memory. If $1^{st}wordinmemory \leq reg \leq 2^{nd}wordinmemory$, no interrupt occurs. Otherwise, an interrupt occurs.

INT instruction : INTn instruction call the interrupt service procedure that begins at the address represented in the vector number n

- **Example:** INT 80H or INT 128 call interrupt service procedure whose address is stored in vector type number 80H(000200H - 000203H)
- Each INT instruction is stored in 2 bytes of memory \rightarrow first byte for opcode, and second byte for interrupt type number.
- **Only exception:** INT3 \rightarrow 1-byte instruction; It is 1 byte as it is easy to insert 1-byte instruction in a program for breakpoint.
- **IRET:** Removes six bytes from the stack \rightarrow 2 for IP , 2 for CS, and 2 for flags
- **Return address of an interrupt:** Next instruction in the program.
- **Return address of BOUND interrupt:** BOUND instruction itself; *NOT* the next instruction.
- **Similar exception applies for INT type number:** 0,5,6,7,,10,11,12 and 13.

Operation of a Real Mode Interrupt :

1. Push flag registers in stack
2. Clear both IF and TF
3. Push CS (Code Segment Register) in stack
4. Push IP (Instruction Pointer) in stack
5. Fetch the interrupt vector contents, and place into both IP and CS so that the next instruction executes at the service procedure addressed by the vector.

13.1.1 Operation of a Protected Mode Interrupt

- Uses a set of 25 interrupt descriptors that are stored in any interrupt descriptor table (IDT). The IDT is 256 x bytes (2KB) long with each descriptor containing bytes.
- The IDT can be located at any place of memory by IDT address register (IDTR)

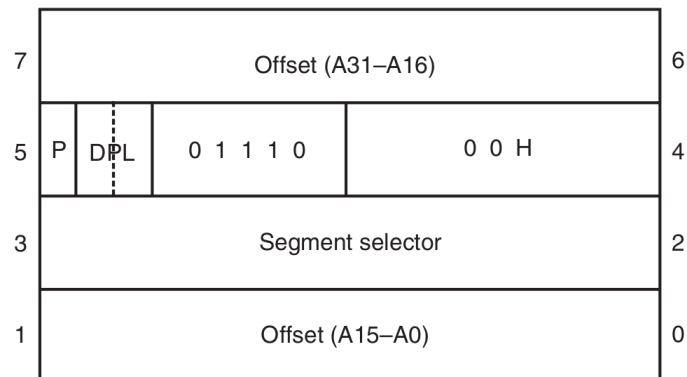


Figure 13.2: The protected mode interrupt descriptor.

- **P bit:** Present
- **DPL bits:** Priviledge level of the interrupt

Chapter 14

Lecture 14

14.1 Interrupt Flag Register



Figure 14.1: The flag register. (Courtesy of Intel Corporation.)

- Clearing TF and IF (2nd step in operation in Real mode)
- Set or clear IF: STI or CLI instructions
- Set or clear TF: No special instruction(Interrupt Service procedure)

14.1.1 Enable Trap Flag

;A procedure that sets the TRAP flag bit to enable trapping

```
TRON PROC NEAR
    PUSH AX;
    PUSH BP

    MOV BP,SP ;Get SP
    MOV AX,[BP+8] ;Get flags
    OR AH,1 ; Set TF
    MOV [BP+8],AX ;Save flags

    POP BP ;Restore Reg
```

```

    POP AX
    TRON ENDP

```

14.1.2 Disable Trap Flag

;A procedure that sets the TRAP flag bit to enable trapping

```

    TRON PROC NEAR
    PUSH AX;
    PUSH BP

    MOV BP,SP ;Get SP
    MOV AX,[BP+8] ;Get flags
    AND AH,0FEH ;Clear TF
    MOV [BP+8],AX ;Save flags

    POP BP ;Restore Reg
    POP AX
    TRON ENDP

```

14.2 Hardware Interrupts

NMI : (+)ve edge triggered; After a positive edge , NMI must retain 1 until it is recognized by the microprocessor. NMI is used for parity errors and other major system faults such as power failure.

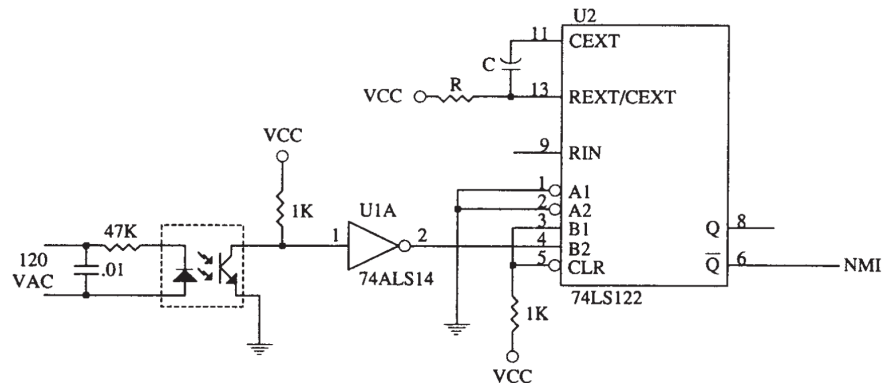


Figure 14.2: A power failure detection circuit.

- An optical isolator provides isolation from the AC power lines
- Output of the isolator is shapped by a **Schmitt-trigger** inverter to feed to 74LS122 monostable multivibrator. As long as AC power is applied, Q remains at logic 1 having \overline{Q} at 0.

- If power fails, 4LS122 no longer receives trigger pulse from 74ALS14. It makes \overline{Q} to go 1 interrupting the μP by NMI pin.
- It is assumed that the system power supply has a large enough filter capacitor to provide energy for at least 75 ms after the AC power ceases.

14.2.1 INTR and \overline{INTA}

- INTR is level sensitive (must be held 1 until recognized)
- INTR is set by external event and cleared inside the interrupt service procedure

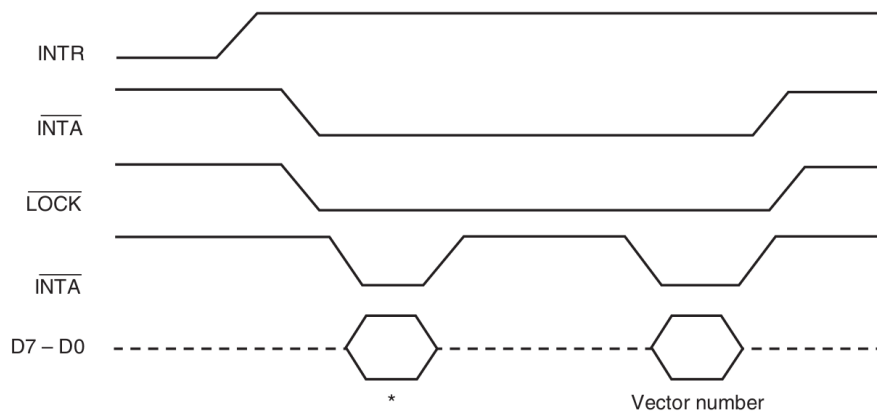


Figure 14.3: The timing of the INTR input and INTA output. *This portion of the data bus is ignored and usually contains the vector number.

14.2.2 Making INTR input edge triggered

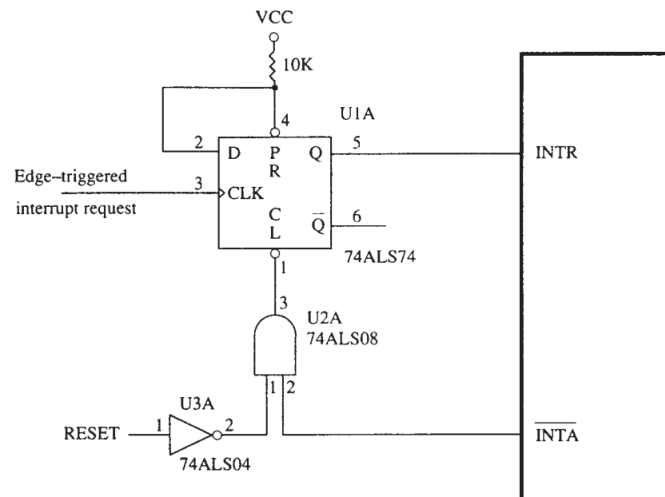


Figure 14.4: Converting INTR into an edge-triggered interrupt request input.

14.2.3 Applying a flex interrupt vector number FFH

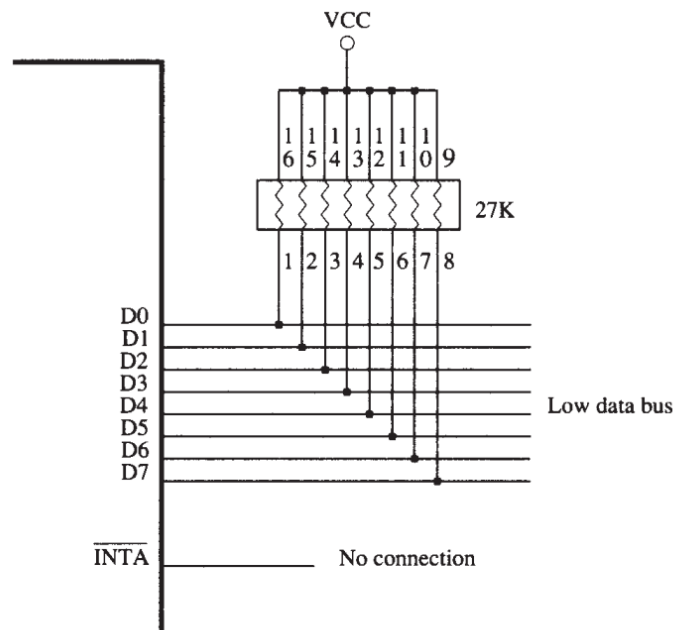


Figure 14.5: A simple method for generating interrupt vector type number FFH in response to INTR.

14.2.4 Using a 3-state buffer to place a vector type number 80H

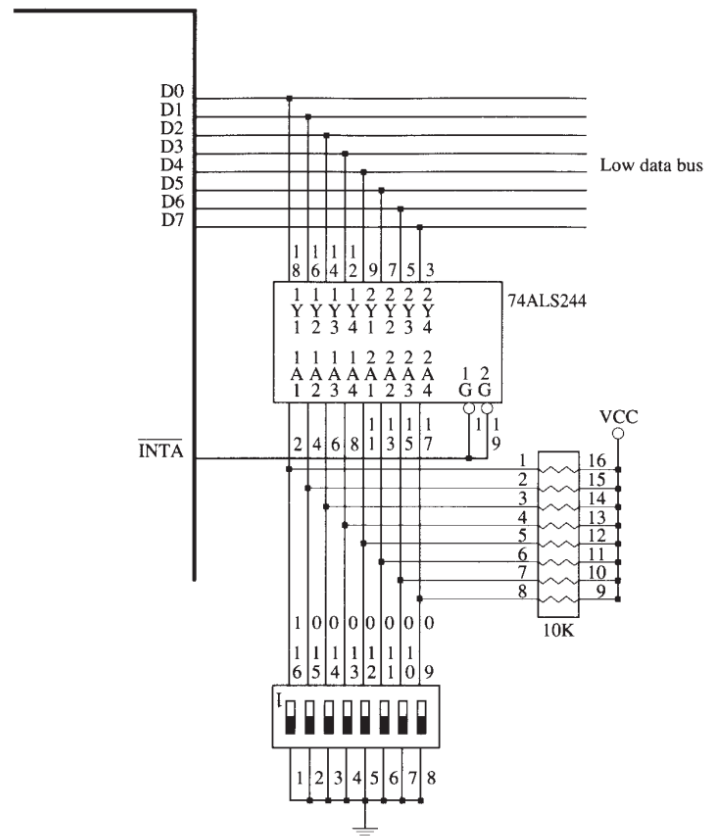


Figure 14.6: A circuit that applies any interrupt vector type number in response to INTA . Here the circuit is applying type number 80H.

14.2.5 Using 74ALS244 to expand to accomodate up to 7 interrupt inputs

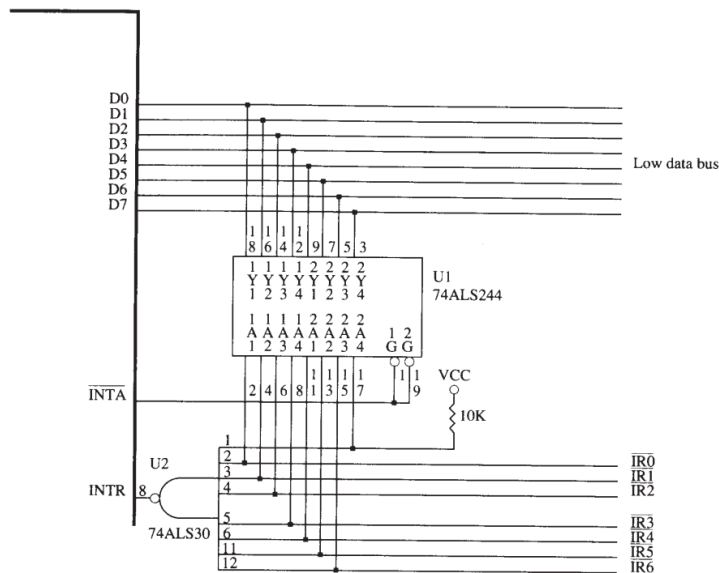


Figure 14.7: Expanding the INTR input from one to seven interrupt request lines.

$\overline{IR6}$	$\overline{IR5}$	$\overline{IR4}$	$\overline{IR3}$	$\overline{IR2}$	$\overline{IR1}$	$\overline{IR0}$	Vector
1	1	1	1	1	1	0	FEH
1	1	1	1	1	0	1	FDH
1	1	1	1	0	1	1	FBH
1	1	1	0	1	1	1	F7H
1	1	0	1	1	1	1	EFH
1	0	1	1	1	1	1	DFH
0	1	1	1	1	1	1	BFH

Figure 14.8: Single interrupt requests for Figure 14.7.

2^7 locations must be used which is wasteful. the solution is **Daisy-Chained Interrupts**

14.2.6 Daisy-Chained Interrupts

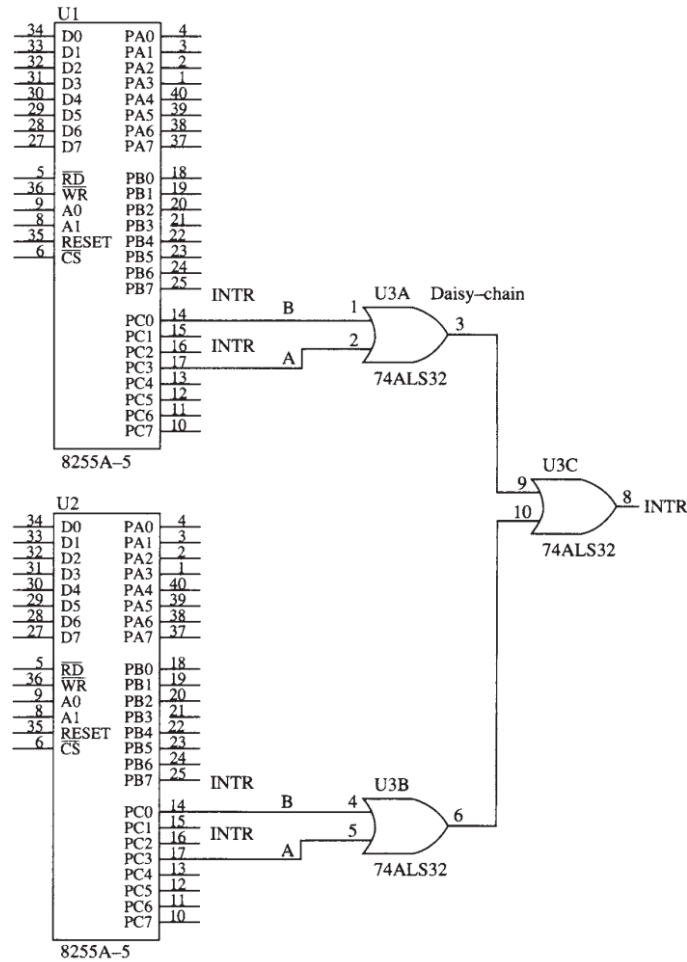


Figure 14.9: Two 82C55 PIAs connected to the INTR outputs are daisy-chained to produce an INTR signal.

14.3 8259A programmable interrupt controller (PIC)

- Adds eight vectored priority encoder interrupts to the μP
- Can be expanded to accept 64 interrupt requests without additional hardware (One Master 259A , 8 slave 8259A)

- Pins of 259A
 1. **IR7-IRO**: Interrupt requests
 2. **CAS2-CAS0**: Cascade lines
 3. SP/\overline{EN} : Slave program/ Enable buffer
 4. **A0**: Selects different command words.
- Command Words:
 1. Initialization command words (ICW)
 - **ICW1,ICW2**: Always
 - **ICW3**: for cascading
 - **ICW4**: for 8086 onward
 2. Operation COmmand Words (OCW) :3

Details of Command Word : *****SELF-STUDY*****

Chapter 15

Lecture 15

15.1 Direct Memory Access (DMA)

There are 3 forms of I/O :

1. Basic
2. Interrupt-processed
3. DMA

Direct Memory Access (DMA): Provides direct access to memory while the microprocessor is temporarily disabled.

Applications: DRAM refresh, video displays for refreshing the screen, disk memory system read/write etc.

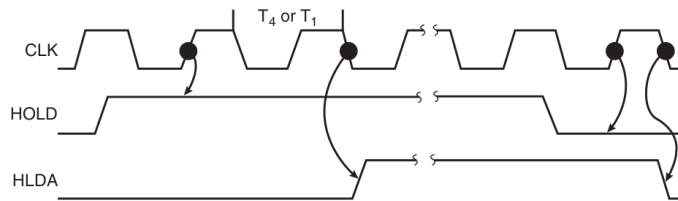


Figure 15.1: HOLD and HLDA timing for the microprocessor.

- HOLD is sampled in the middle of any clock cycle. As soon as the microprocessor recognized the hold, it stops executing software and enters hold cycle.
- HOLD is a higher priority than INTR and NMI. Interrupts take effect at the end of an instruction, while a HOLD takes effect in the middle of an instruction.

- The only microprocessor pin having a higher priority than HOLD is RESET
- The HLDA becomes active to indicate that the microprocessor has places its buses at high-impedence state.
- **HOLD**: DMA requested
- **HLDA**: DMA grant output.
- **DMA read**: Transfers data from memory to I/O (\overline{MRDC} , \overline{IOWC})
- **DMA write**: Transfers data from memory to I/O (\overline{MWTC} , \overline{IORC})

15.1.1 Control signal generation in 8086/8088

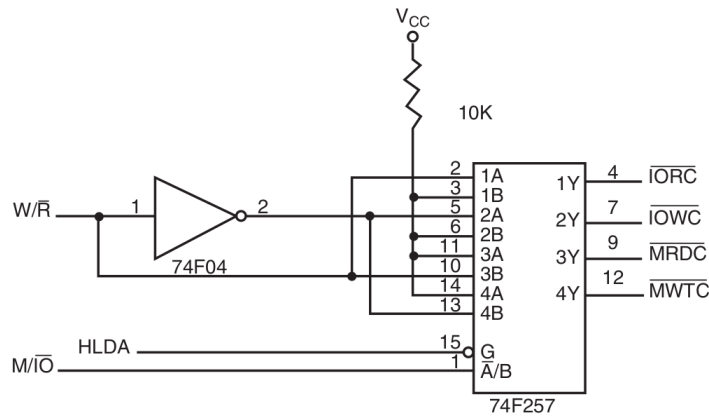


Figure 15.2: A circuit that generates system control signals in a DMA environment.

15.1.2 8237 DMA Controller

A special purpose microprocessor whose job is high-speed data transfer between memory and I/O.

Pins of 8237 :

DREQ3 - DREQ0 DMA request input (four-channel)

DACK3 - DACK0 DMA channel acknowledge.

HRQ HOLD request

HLDA HOLD acknowledge

M/ \overline{IO}	W/ \overline{R}	Output
M	W \rightarrow NOT	$\overline{MWT\overline{C}}$
M	\overline{R}	\overline{MRDC}
\overline{IO}	W \rightarrow NOT	$\overline{IOW\overline{C}}$
\overline{IO}	\overline{R}	$\overline{IOR\overline{C}}$

Table 15.1: Control signal generation in 8086/8088

\overline{EOP} End of Process; Used to terminate a DMA process or to signal the end of a DMA transfer.

\overline{IOR} I/O read is a bidirectional pin used during programming and during a DMA write cycle.

\overline{IOW} I/O write is a bidirectional pin used during programming and during a DMA read cycle.

$A_0 - A_3$ These address pins select an internal register during programming and also provide part of the DMA transfer address during a DMA action. The address pins are outputs that provide part of the DMA transfer address during a DMA action.

$DB_0 - DB_7$ The data bus pins are connected to the microprocessor data bus connections and are used during the programming of the DMA controller.

CLK The clock input is connected to the system clock signal as long as that signal is 5 MHz or less. In the 8086/8088 system, the clock must be inverted for the proper operation of the 8237.

\overline{CS} Chip select enables the 8237 for programming. The CS pin is normally connected to the output of a decoder. The decoder does not use the 8086/8088 control signal IO/\overline{M} (M/\overline{IO}) because it contains the new memory and I/O control signals (MEMR , MEMW , IOR , and IOW).

Internal Register ***SELF-STUDY*******

15.1.3 Arithmetic Coprocessor

8087, 80287, 80387X,

80x87

Ability to multiply, divide, add, Subtract, Square Root, Partial tangent, Partial arctangent, tangent **Data types:** 16-, 32-, 64-bit signed integers; 18-digit BCD data, and 32-, 64-, and 80-bit floating point.

- Operations performed by the 80x87 generally executes many times faster than equivalent operations written with the most efficient program that use the microprocessor's normal instruction set.
- Can operate concurrently with microprocessor.
- **Coprocessor instructions:** escape(ESC) instructions used by the microprocessor to generate memory address for the Coprocessor so that the Coprocessor can execute a Coprocessor instruction.

Divided into two major sections: Control unit and Numeric execution unit.

Control Unit Interfaces the Coprocessor to the microprocessor. Both monitor the instruction stream. Coprocessor executes if it is an ESC instruction. Microprocessor executes if it is not.

Numeric execution unit Eight-register stack for holding operands and results of an arithmetic instruction. Each register is 80-bit. It has a status register that reflects overall operation of the Coprocessor. "FSTSW AX" instruction copies the status register to AX, which gives the only way to communicate with the microprocessor (for 80287 and above).

80386 and 80486 basics ***SELF-STUDY*******