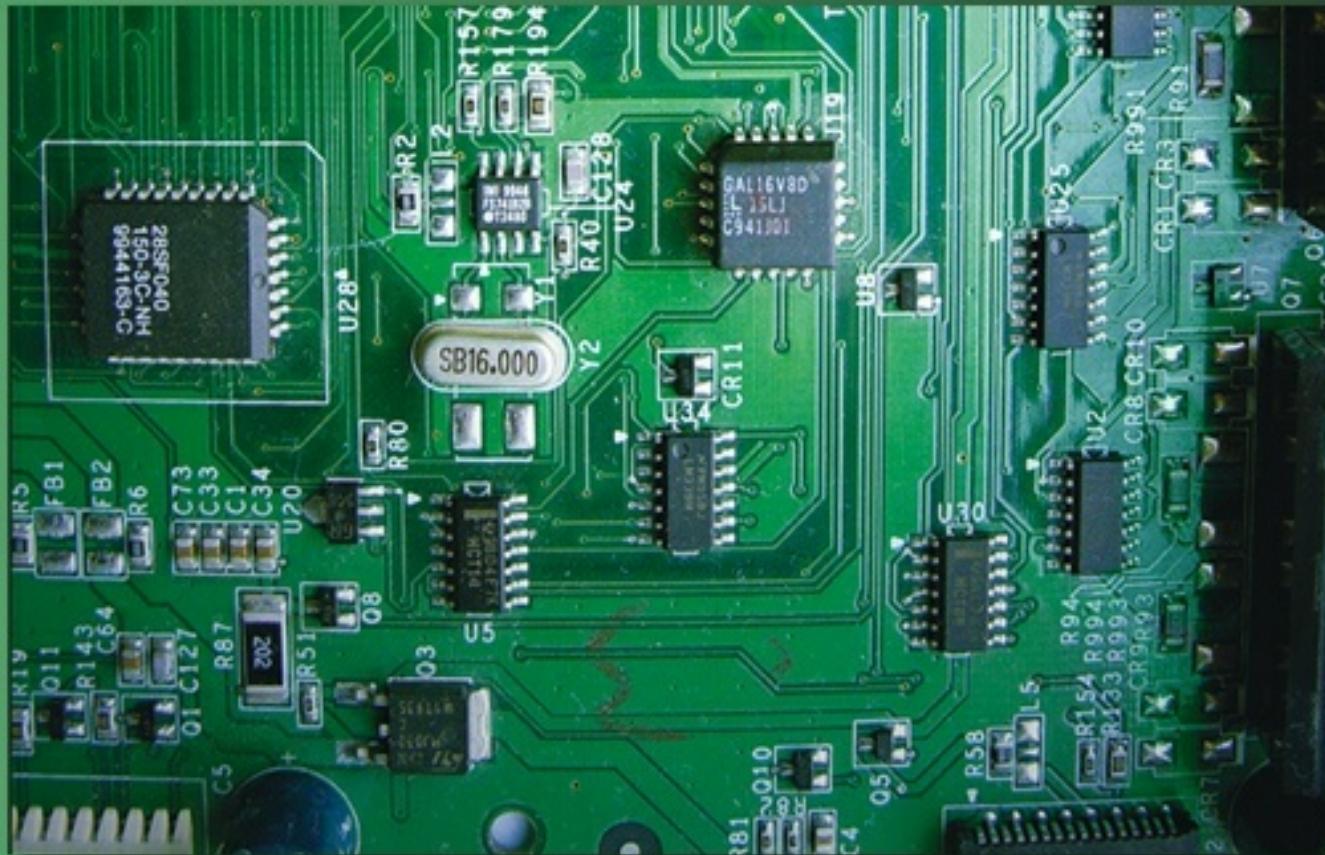


# The Intel Microprocessors

8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions

# Architecture, Programming, and Interfacing



EIGHTH EDITION

Barry B. Brey



# Chapter 9: 8086/8088 Hardware Specifications

# Introduction

- In this chapter, the **pin functions** of both the 8086 and 8088 microprocessors are detailed and information is provided on the following hardware topics: **clock generation, bus buffering, bus latching, timing, wait states, and minimum mode operation versus maximum mode operation.**





*The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium,  
Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions  
Architecture, Programming, and Interfacing, Eighth Edition*  
Barry B. Brey

---

Copyright ©2009 by Pearson Education, Inc.  
Upper Saddle River, New Jersey 07458 • All rights reserved.

(cont.)



*The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium,  
Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions  
Architecture, Programming, and Interfacing, Eighth Edition*  
Barry B. Brey

---

Copyright ©2009 by Pearson Education, Inc.  
Upper Saddle River, New Jersey 07458 • All rights reserved.

# PIN-OUTS AND THE PIN FUNCTIONS

- In this section, we explain the function and the **multiple functions** of each of the microprocessor's pins.
- In addition, we discuss the **DC characteristics** to provide a basis for understanding the later sections on **buffering** and **latching**.

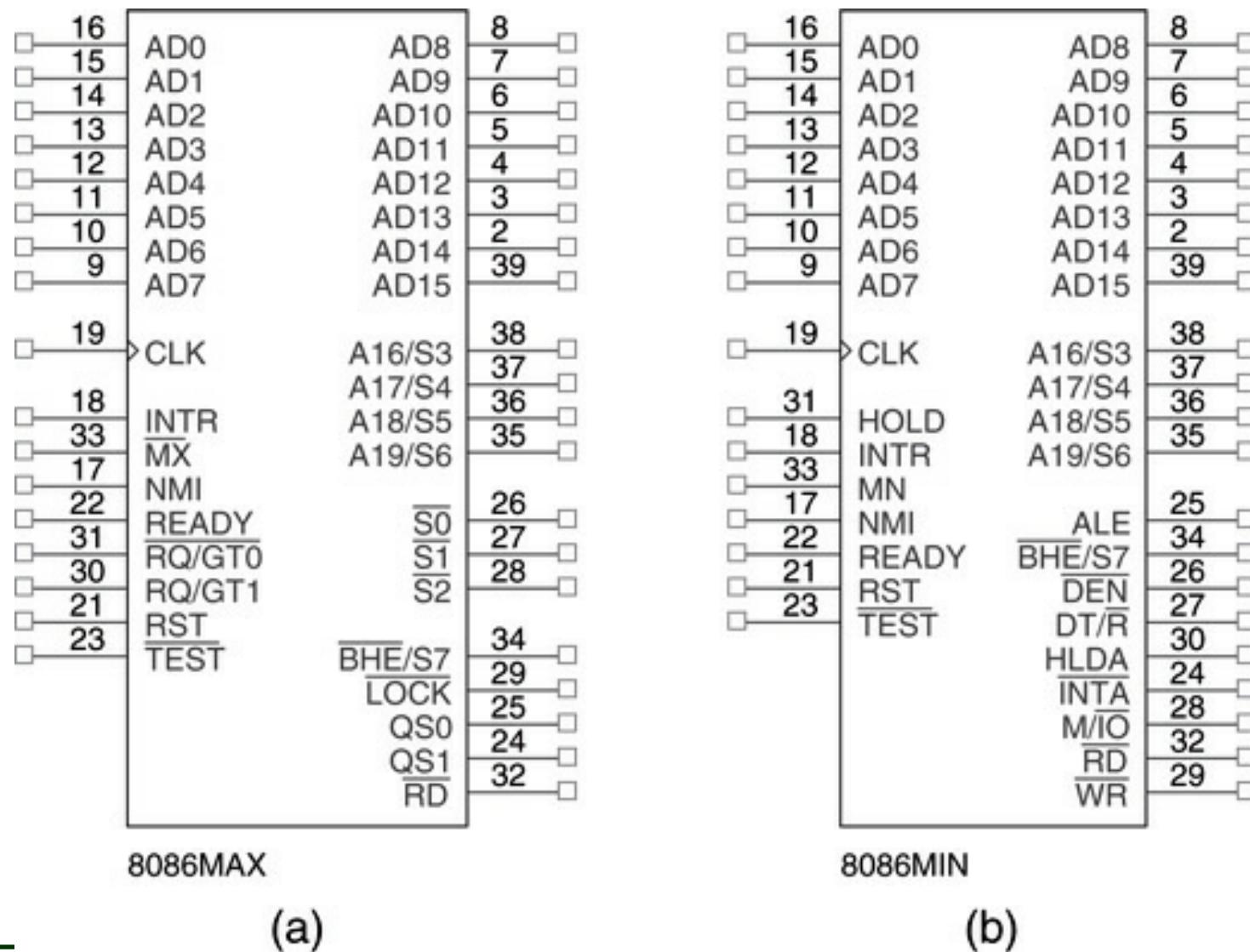


# PIN-OUTS AND THE PIN FUNCTIONS

- Figure 9–1 illustrates pin-outs of 8086 & 8088.
- 8086 is a **16-bit** microprocessor with a 16-bit data bus; 8088 has an **8-bit** data bus.
  - 8086 has pin connections  $AD_0-AD_{15}$
  - 8088 has pin connections  $AD_0-AD_7$
- Data bus width is the only major difference.
  - thus 8086 transfers 16-bit data more efficiently



**Figure 9–1** (a) The pin-out of the 8086 in **maximum mode**; (b) the pin-out of the 8086 in **minimum mode**.



# Power Supply Requirements

- Both microprocessors require +5.0 V with a supply voltage tolerance of **±10** percent.
  - 8086 uses a maximum supply current of 360 mA
  - 8088 draws a maximum of 340 mA
- Both microprocessors operate in ambient temperatures of between **32° F** and **180° F**.
- 80C88 and 80C86 are **CMOS versions** that require only **10 mA** of power supply current.
  - and function in temperature extremes of **-40° F** through **+225° F**



# DC Characteristics

- It is impossible to connect anything to a microprocessor without knowing **input current** requirement for an input pin.
  - and the **output current drive capability** for an output pin
- This knowledge allows hardware designers to select **proper interface** components for use with the microprocessor
  - without the fear of damaging anything



# *Input Characteristics*

- Input characteristics of these microprocessors are compatible with all the standard logic components available today.
- The input current levels are very small because the inputs are the gate connections of MOSFETs and represent only **leakage** currents.



# *Output Characteristics*

- The logic 1 voltage level of the 8086/8088 is compatible with most standard logic families.
  - logic 0 level is not
- Standard logic circuits have a **maximum** logic 0 voltage of 0.4 V; 8086/8088 has a maximum of 0.45 V.
  - a difference of 0.05 V



- This difference reduces the **noise immunity** from 400 mV (0.8 V – 0.45 V) to 350 mV.
  - noise immunity is the difference between logic 0 output voltage and logic 0 input voltage levels
- Reduction in noise immunity may result in problems with **long wire connections or too many loads.**
- No more than **10 loads** of any type should be connected to an output pin without buffering
  - if this factor is exceeded, **noise** will begin to take its toll in timing problems



# Pin Connections AD<sub>7</sub> - AD<sub>0</sub>

- 8088 **address/data bus** lines are multiplexed
  - and contain the rightmost 8 bits of the memory **address** or I/O port number whenever **ALE** is active (logic 1)
  - or data whenever **ALE** is inactive (logic 0)



# Pin Connections A<sub>15</sub> - A<sub>8</sub>

- 8088 **address bus** provides the upper-half memory address bits that are present throughout a bus cycle.



# Pin Connections AD<sub>15</sub> - AD<sub>8</sub>

- 8086 **address/data bus** lines compose upper multiplexed address/data bus on the 8086.
- These lines contain address bits A<sub>15</sub>–A<sub>8</sub> whenever **ALE** is a logic 1, and data bus connections D<sub>15</sub>–D<sub>8</sub> when **ALE** is a logic 0.
- These pins enter a **high-impedance state** when a hold acknowledge occurs.



# Pin Connections $A_{19}/S_6 - A_{16}/S_3$

- **Address/status bus** bits are multiplexed to provide address signals  $A_{19}-A_{16}$  and **status bits**  $S_6-S_3$ .
  - status bit  $S_6$  is always logic 0,
  - bit  $S_5$  indicates the condition of the IF flag bit
- $S_4$  and  $S_3$  show which segment is accessed during the current bus cycle.
  - these **status bits** can address four separate 1M byte memory banks by decoding as  $A_{21}$  and  $A_{20}$



# Pin Connections RD<sup>—</sup>

- When **read signal** is logic 0, the data bus is receptive to data from memory or I/O devices
  - pin floats high-impedance state during a hold acknowledge

## Ready

- **Inserts wait states into the timing.**
  - if placed at a logic 0, the microprocessor enters into wait states and remains idle
  - if logic 1, no effect on the operation



# Pin Connections INTR

- **Interrupt request** is used to request a hardware interrupt.
  - If INTR is held high when IF = 1, 8086/8088 enters an interrupt acknowledge cycle after the current instruction has completed execution

## NMI

- The **non-maskable interrupt** input is similar to INTR.
  - does not check IF flag bit for logic 1
  - if activated, uses interrupt vector 2



# Pin Connections $\overline{\text{TEST}}$

- The  $\overline{\text{Test}}$  pin is an input that is tested by the WAIT instruction.
- If  $\overline{\text{TEST}}$  is a logic 0, the WAIT instruction functions as an NOP.
- If  $\overline{\text{TEST}}$  is a logic 1, the WAIT instruction waits for  $\overline{\text{TEST}}$  to become a logic 0.
- The  $\overline{\text{TEST}}$  pin is most often connected to the 8087 numeric coprocessor.



# Pin Connections RESET

- Causes the microprocessor to reset itself if held high a minimum of four clocking periods.
  - when 8086/8088 is reset, it executes instructions at memory location FFFF0H
  - also disables future interrupts by clearing IF flag

## CLK

- The **clock** pin provides the **basic timing signal**.
  - must have a duty cycle of 33 % (high for one third of clocking period, low for two thirds) to provide proper internal timing



# Pin Connections VCC

- This **power supply** input provides a **+5.0 V, ±10 %** signal to the microprocessor.

## GND

- The **ground** connection is the return for the power supply.
  - 8086/8088 microprocessors have two pins labeled GND—both must be connected to ground for proper operation



# Pin Connections MN/ $\overline{MX}$

- **Minimum/maximum** mode pin selects either minimum or maximum mode operation.
  - if minimum mode selected, the MN/ $\overline{MX}$  pin must be connected directly to +5.0 V

## BHE S<sub>7</sub>

- The **bus high enable** pin is used in 8086 to enable the most-significant data bus bits (D<sub>15</sub>–D<sub>8</sub>) during a read or a write operation.
- The state of S<sub>7</sub> is always a logic 1.



# *Minimum Mode Pins*

- Minimum mode operation is obtained by connecting the MN/ $\overline{MX}$  pin directly to +5.0 V.
  - do not connect to +5.0 V through a pull-up register; it will not function correctly

## **IO/ $\overline{M}$ or M/IO**

- The IO/ $\overline{M}$  (8088) or M/IO (8086) pin **selects** memory or I/O.
  - indicates the address bus contains either a memory address or an I/O port address.
  - **high-impedance state** during **hold acknowledge**



# *Minimum Mode Pins* $\overline{\text{WR}}$

- **Write line** indicates 8086/8088 is outputting data to a memory or I/O device.
  - during the time  $\overline{\text{WR}}$  is a logic 0, the data bus contains valid data for memory or I/O
  - high-impedance during a hold acknowledge

# $\overline{\text{INTA}}$

- The **interrupt acknowledge** signal is a response to the **INTR** input pin.
  - normally used to gate the interrupt vector number onto the data bus in response to an interrupt



# *Minimum Mode Pins ALE*

- **Address latch enable** shows the 8086/8088 address/data bus contains an address.
  - can be a memory address or an I/O port number
  - ALE signal doesn't float during hold acknowledge

## DT/R

- The **data transmit/receive** signal shows that the microprocessor data bus is transmitting ( $DT/\bar{R} = 1$ ) or receiving ( $DT/\bar{R} = 0$ ) data.
  - used to enable external data bus buffers



# *Minimum Mode Pins DEN*

- **Data bus enable** activates external data bus buffers.

## HOLD

- **Hold input** requests a direct memory access (DMA).
  - if **HOLD** signal is a logic 1, the microprocessor stops executing software and places address, data, and control bus at **high-impedance**
  - if a logic 0, software executes normally



# *Minimum Mode Pins HLDA*

- **Hold acknowledge** indicates the 8086/8088 has entered the **hold state**.

## SS0

- The  $\overline{SS0}$  status line is **equivalent** to the  $S_0$  pin in maximum mode operation.
- **Signal** is combined with  $IO/M$  and  $DT/R$  to decode the function of the current bus cycle.



# *Maximum Mode Pins*

- In order to achieve maximum mode for use with external coprocessors, connect the MN/  
**MX** **pin to ground.**

**$\overline{S2}$ ,  $\overline{S1}$ , and  $\overline{S0}$**

- **Status bits** indicate function of the current bus cycle.
  - normally decoded by the 8288 bus controller



# *Maximum Mode Pins* $\overline{RQ/GT1}$

- The **request/grant** pins request direct memory accesses (DMA) during maximum mode operation.
  - bidirectional; used to request and grant a DMA operation

## LOCK

- The **lock** output is used to lock peripherals off the system. This pin is activated by using the **LOCK:** prefix on any instruction.



# *Maximum Mode Pins QS<sub>1</sub> and QS<sub>0</sub>*

- The **queue status bits** show the status of the internal instruction queue.
  - provided for access by the 8087 coprocessor

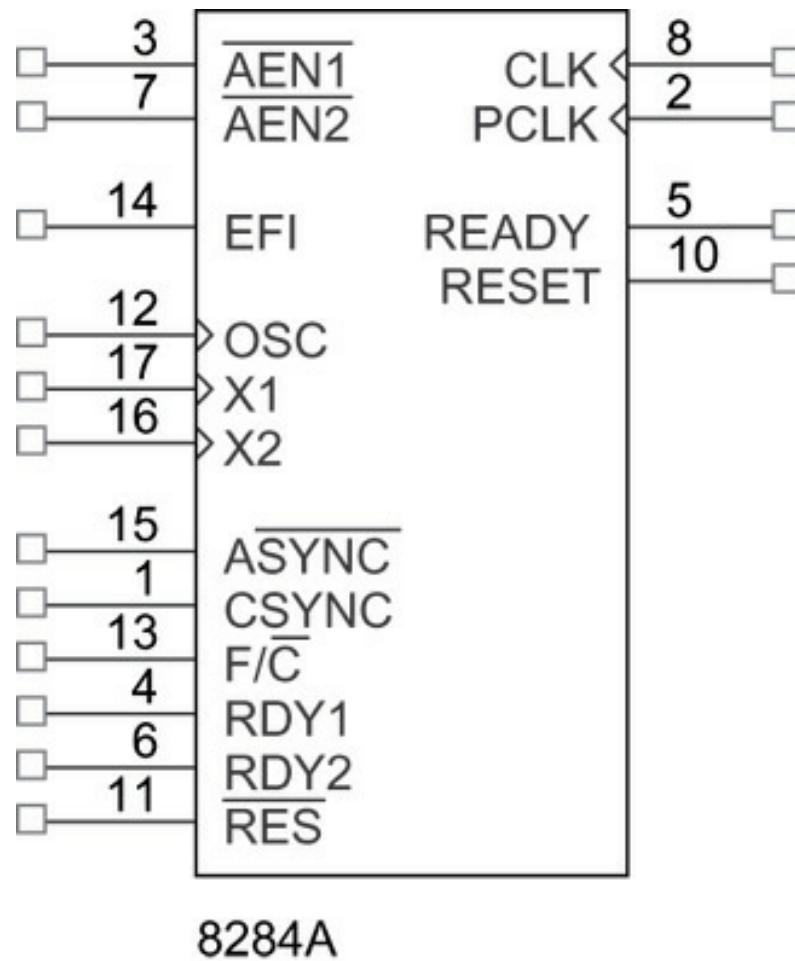


## 9–2 CLOCK GENERATOR (8284A)

- This section describes the 8284A clock generator and the **RESET** signal.
  - also introduces the **READY** signal for 8086/8088
- With no clock generator, many circuits would be required to generate the clock (CLK).
- 8284A provides the following basic functions:
  - clock generation; RESET & READY synch;
  - TTL-level peripheral clock signal
- Figure 9–2 shows **pin-outs** of the 8284A



# Figure 9–2 The pin-out of the 8284A clock generator.



# 8284A *Pin Functions*

- 8284A is an 18-pin integrated circuit designed specifically for use 8086/8086.

## AEN1 and AEN2

- The **address enable** pins are provided to qualify bus ready signals, RDY1 and RDY2.
  - used to cause wait states
- **Wait states** are generated by the **READY** pin of 8086/8088 controlled by these two inputs.



# *Pin Functions RDY<sub>1</sub> and RDY<sub>2</sub>*

- The **bus ready** inputs are provided, in conjunction with the  $\overline{\text{AEN}1}$  &  $\overline{\text{AEN}2}$  pins, to cause wait states in 8086/8088.

## ASYNC

- The **ready synchronization** selection input selects either one or two stages of synchronization for the RDY<sub>1</sub> and RDY<sub>2</sub> inputs.



# *Pin Functions* READY

- 8284 **Ready** is an output pin that connects to the 8086/8088 READY input.
  - synchronized with the RDY<sub>1</sub> and RDY<sub>2</sub> inputs

## X<sub>1</sub> and X<sub>2</sub>

- The **crystal oscillator** pins connect to an external crystal used as the timing source for the clock generator and all its functions



# *Pin Functions F/C<sup>-</sup>*

- The **frequency/crystal** select input chooses the clocking source for the 8284A.
  - if held high, an **external clock** is provided to the EFI input pin
  - if held low, the **internal crystal oscillator** provides the timing signal
- The external frequency input is used when the **F/C** pin is pulled high.
- EFI supplies timing when the **F/C** pin is high.



# *Pin Functions CLK*

- The **clock output** pin provides the CLK input signal to 8086/8088 and other components.
  - output signal is one third of the crystal or EFI input frequency
  - 33% duty cycle required by the 8086/8088

# PCLK

- The **peripheral clock** signal is one sixth the crystal or EFI input frequency.
  - PCLK output provides a clock signal to the peripheral equipment in the system



# *Pin Functions* OSC

- **Oscillator output** is a TTL-level signal at the same frequency as crystal or EFI input.
  - OSC output provides EFI input to other 8284A clock generators in multiple-processor systems

## RES

- **Reset input** is an active-low input to 8284A.
  - often connected to an RC network that provides power-on resetting



# *Pin Functions* RESET

- **Reset output** is connected to the 8086/8088 RESET input pin.

# CSYNCH

- The **clock synchronization** pin is used when the EFI input provides synchronization in systems with multiple processors.
  - if internal crystal oscillator is used, this pin must be grounded



# *Pin Functions* GND

- The **ground pin** connects to ground.

## VCC

- This **power supply pin** connects to +5.0 V with a tolerance of  $\pm 10\%$ .

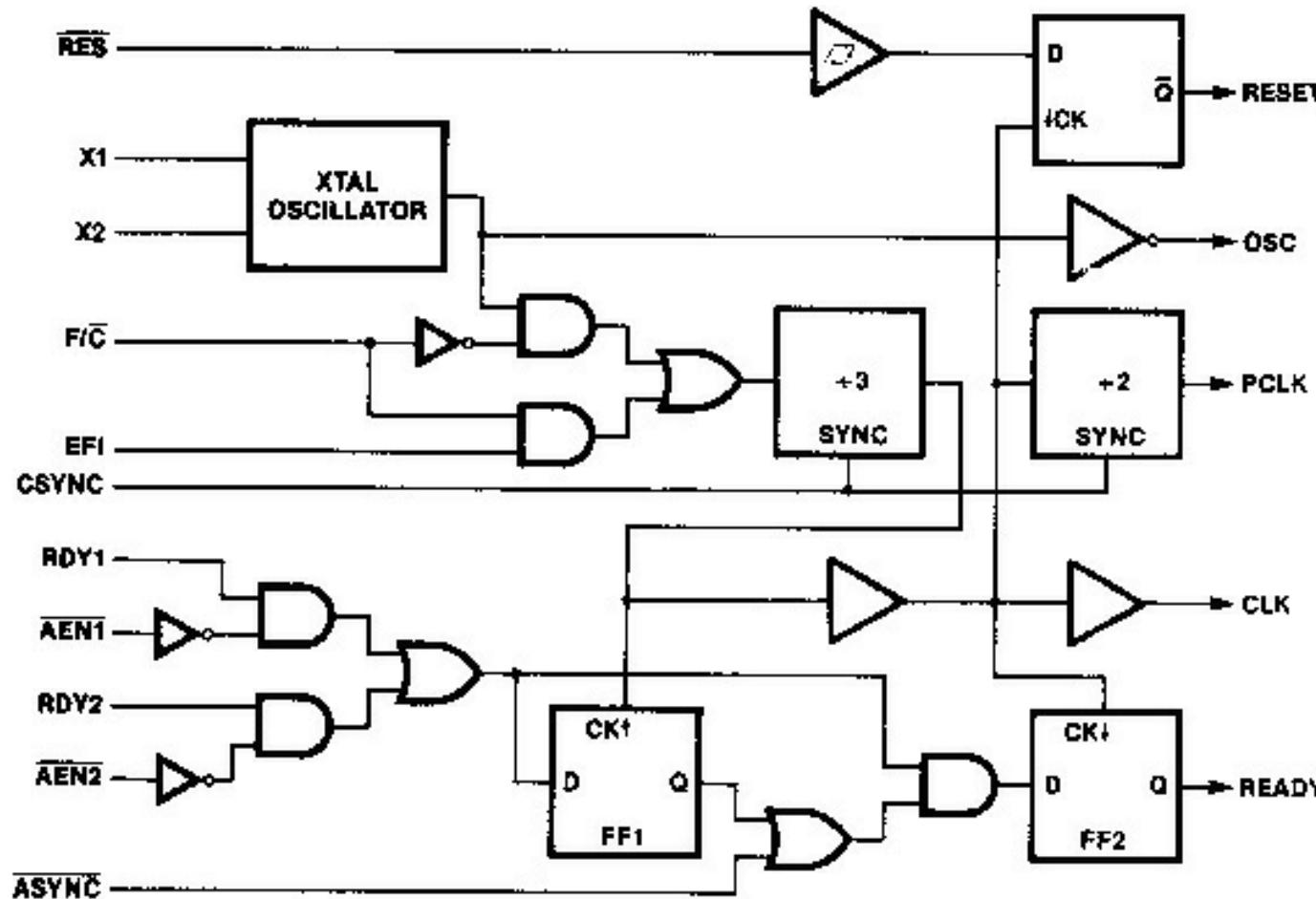


# Operation of the 8284A

- The 8284A is a relatively easy component to understand.
- Figure 9–3 illustrates the internal timing diagram of the 8284A clock generator.
- The top half of the logic diagram represents the clock and synchronization section of the 8284A clock generator.



**Figure 9–3** The internal block diagram of the 8284A clock generator.

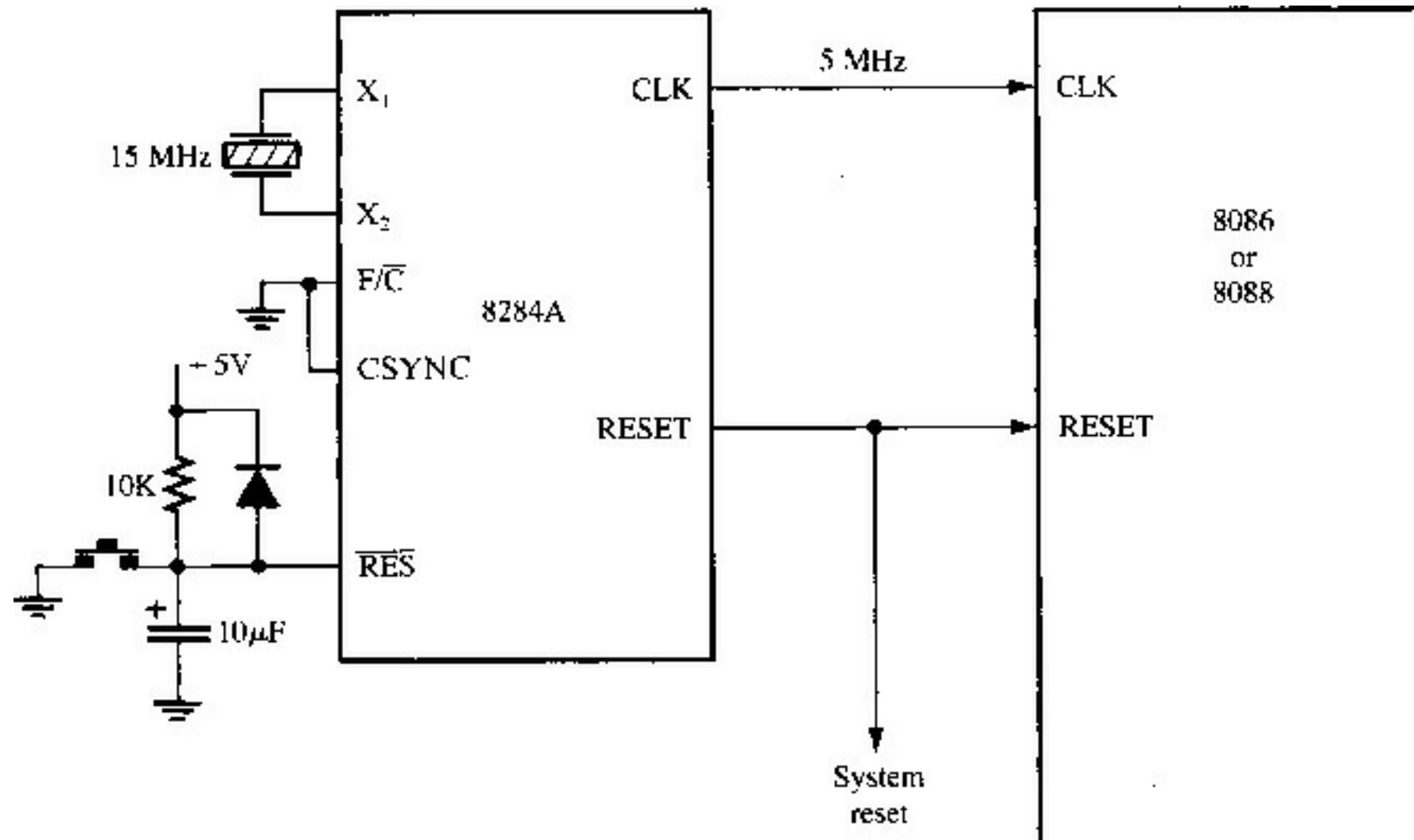


# *Operation of the Clock Section*

- Crystal oscillator has two inputs:  $X_1$  and  $X_2$ .
  - if a crystal is attached to  $X_1$  and  $X_2$ , the oscillator generates a square-wave signal at the same frequency as the crystal
- The square-wave is fed to an AND gate & an inverting buffer to provide an OSC output.
- The OSC signal is sometimes used as an E/FI input to other 8284A circuits in a system.
- Figure 9–4 shows how an 8284A is connected to the 8086/8088.



**Figure 9–4** 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.



# *Operation of the Reset Section*

- The reset section of 8284A consists of a **Schmitt trigger buffer** and a **D-type flip-flop**.
  - the D-type flip-flop ensures timing requirements of 8086/8088 RESET input are met
- This circuit applies the RESET signal on the **negative edge (1-to-0 transition)** of each clock.
- 8086/8088 microprocessors sample RESET at the **positive edge (0-to-1 transition)** clocks.
  - thus, this circuit meets 8086/8088 timing requirements



# 9–3 BUS BUFFERING AND LATCHING

- Before 8086/8088 can be used with memory or I/O interfaces, their **multiplexed buses** must be **demultiplexed**.
- This section provides detail required to demultiplex the buses and illustrates how the **buses** are buffered for very large systems.
  - because the maximum fan-out is 10, the system must be buffered if it contains more than 10 other components



# Demultiplexing the Buses

- The **address/data** bus of the 8086/8088 is **multiplexed** (shared) to reduce the number of pins required for the integrated circuit
  - the hardware designer must **extract** or **demultiplex** information from these pins
- Memory & I/O require the address remain valid and stable throughout a read/write cycle.
- If buses are **multiplexed**, the address changes at the memory and I/O, causing them to read or write data in the wrong locations



- All computer systems have **three buses**:
  - an **address bus** that provides memory and I/O with the memory address or the I/O port number
  - a **data bus** that transfers data between the microprocessor and the memory and I/O
  - a **control bus** that provides control signals to the memory and I/O
- These buses must be present in order to interface to memory and I/O.



# *Demultiplexing the 8088*

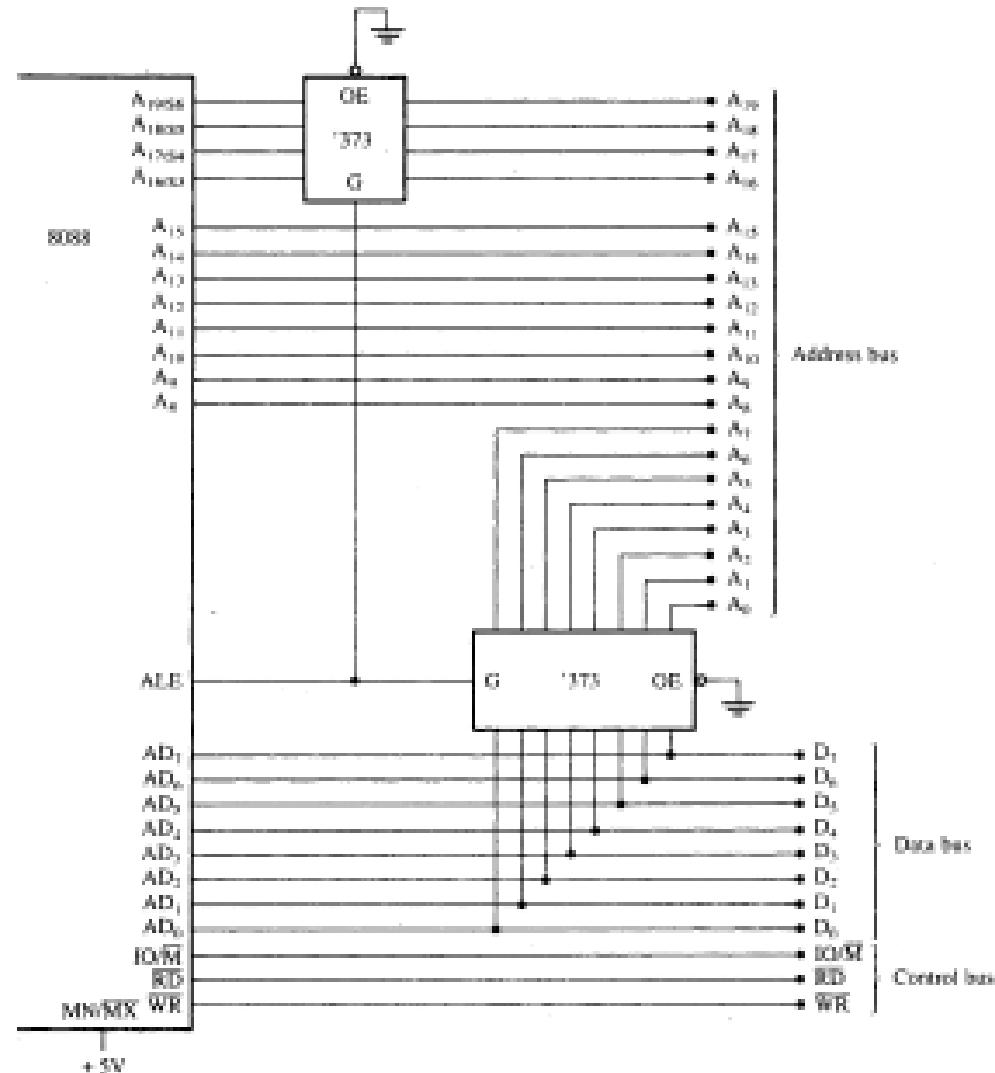
- Figure 9–5 illustrates components required to demultiplex 8088 buses.
  - two 74LS373 or 74LS573 **transparent latches** are used to demultiplex the address/data bus connections  $AD_7 - AD_0$
  - and address/status connections  $A_{19}/S_6 - A_{16}/S_3$
- The latches, which are like **wires** whenever the **address latch enable** pin (**ALE**) becomes a logic 1, pass the inputs to the outputs.



**Figure 9–5** The 8088 microprocessor shown with a demultiplexed address bus. This is the model used to build many 8088-based systems.

8086/8088 HARDWARE SPECIFICATIONS

311



- After a short time, ALE returns to logic 0 causing the latches to remember inputs at the time of the change to a logic 0.
- This yields a **separate address bus** with connections  $A_{19}-A_0$ .
  - these allow 8088 to address 1Mb of memory
- The **separate data bus** allows it to be connected to any 8-bit peripheral device or memory component.

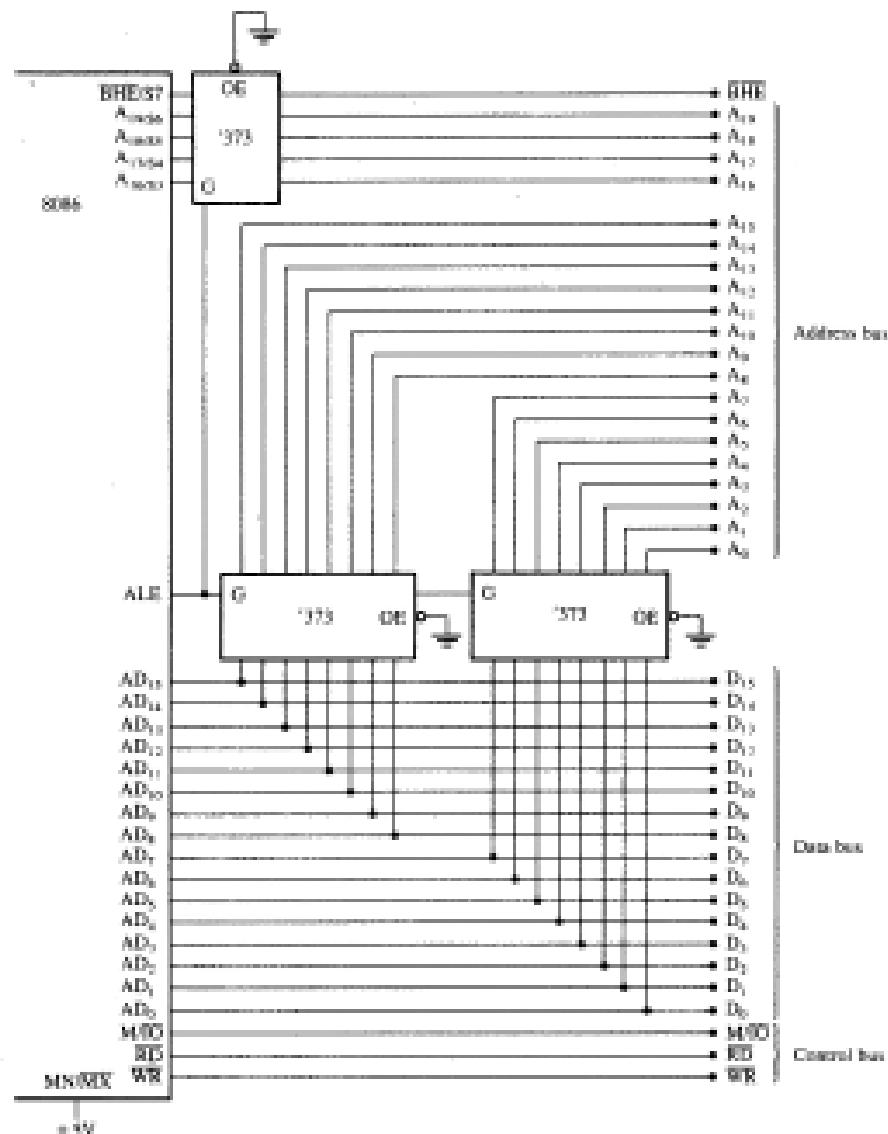


# *Demultiplexing the 8086*

- Fig 9–6 illustrates a demultiplexed 8086 with all three buses:
- **address** ( $A_{19}-A_0$  and  $\overline{BHE}$ )
- **data** ( $D_{15}-D_0$ ), \_\_\_\_\_
- **control** (M/IO,RD, and WR )
- Here, the memory and I/O system see the 8086 as a device with:
  - a **20-bit address bus;16-bit data bus**
  - and a **three-line control bus**



**Figure 9–6** The 8086 microprocessor shown with a demultiplexed address bus. This is the model used to build many 8086-based systems.



# The Buffered System

- If more than 10 unit loads are attached to any bus pin, the entire system must be **buffered**.
- Buffer output currents have been increased so that more TTL unit loads may be driven.
- A fully buffered signal will introduce a **timing delay** to the system.
- No difficulty unless memory or I/O devices are used which function at near **maximum bus speed**.

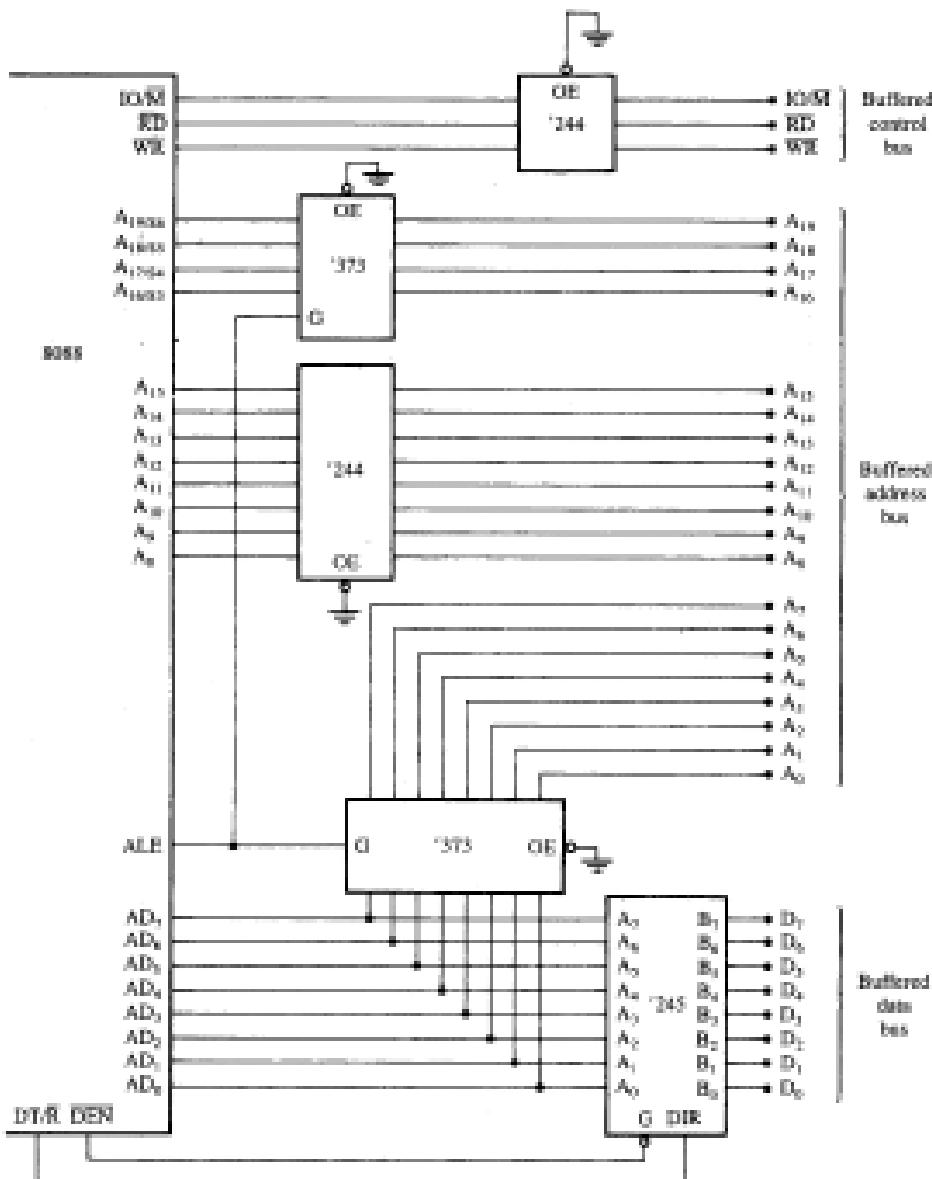


# *The Fully Buffered 8088*

- Figure 9–7 depicts a fully buffered 8088 microprocessor.
  - a fully buffered 8088 system requires two 74LS244s, one 74LS245, and two 74LS373s
- **Direction** of the 74LS245 is controlled by the **DT/R** signal.
  - **enabled** and disabled by the **DEN** signal



# Figure 9–7 A fully buffered 8088 microprocessor.



The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions Architecture, Programming, and Interfacing, Eighth Edition  
Barry B. Brey

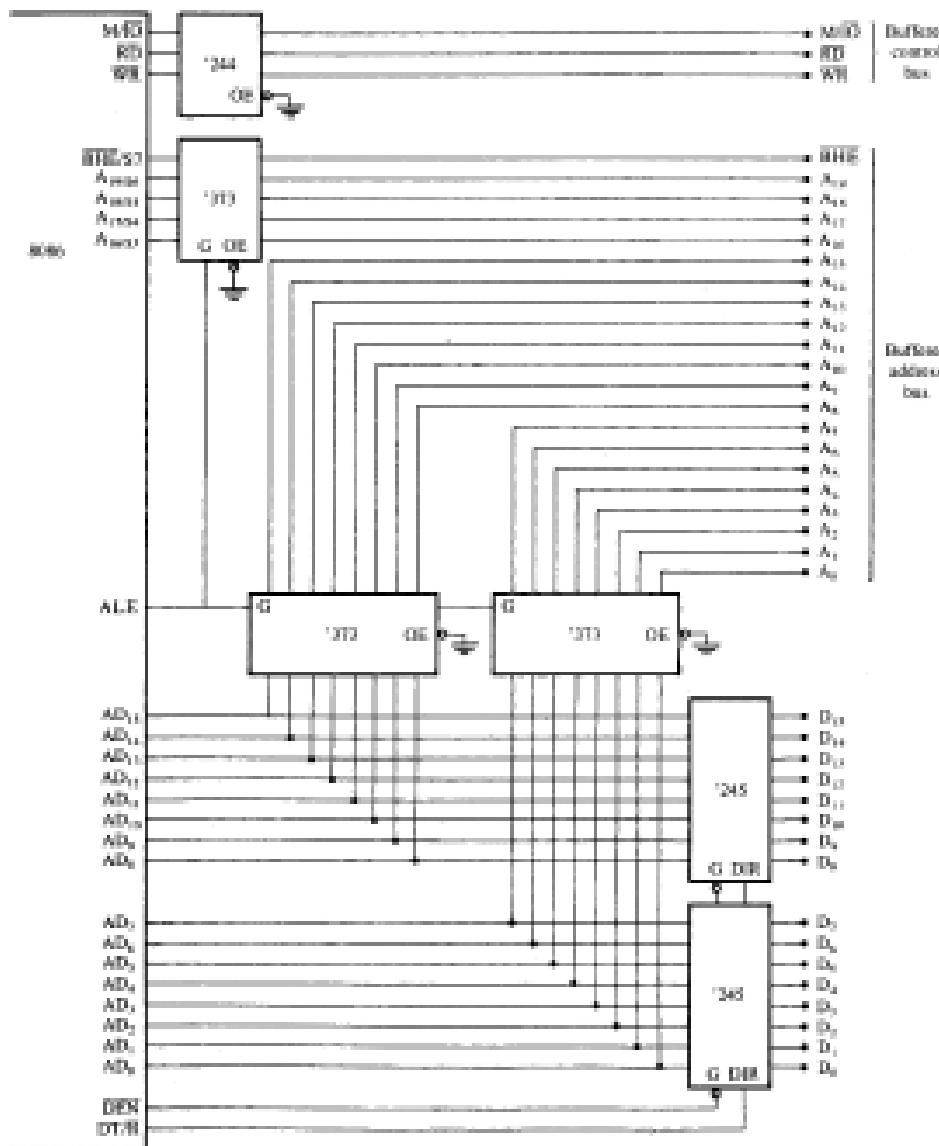
Copyright ©2009 by Pearson Education, Inc.  
Upper Saddle River, New Jersey 07458 • All rights reserved.

# *The Fully Buffered 8086*

- Figure 9–8 illustrates a **fully buffered** 8086.
  - a fully buffered 8086 system requires **one** 74LS244, **two** 74LS245s, and **three** 74LS373s
- 8086 requires one more buffer than 8088 because of the extra eight data bus connections,  $\overline{D_{15}} - D_8$ .
- It also has a **BHE** signal that is buffered for **memory-bank selection**.



**Figure 9–8** A fully buffered 8086 microprocessor.



## 9–4 BUS TIMING

- It is essential to understand system **bus timing** before choosing memory or I/O devices for interfacing to 8086 or 8088 microprocessors.
- This section provides insight into operation of the bus signals and the **basic read/write timing** of the 8086/8088.

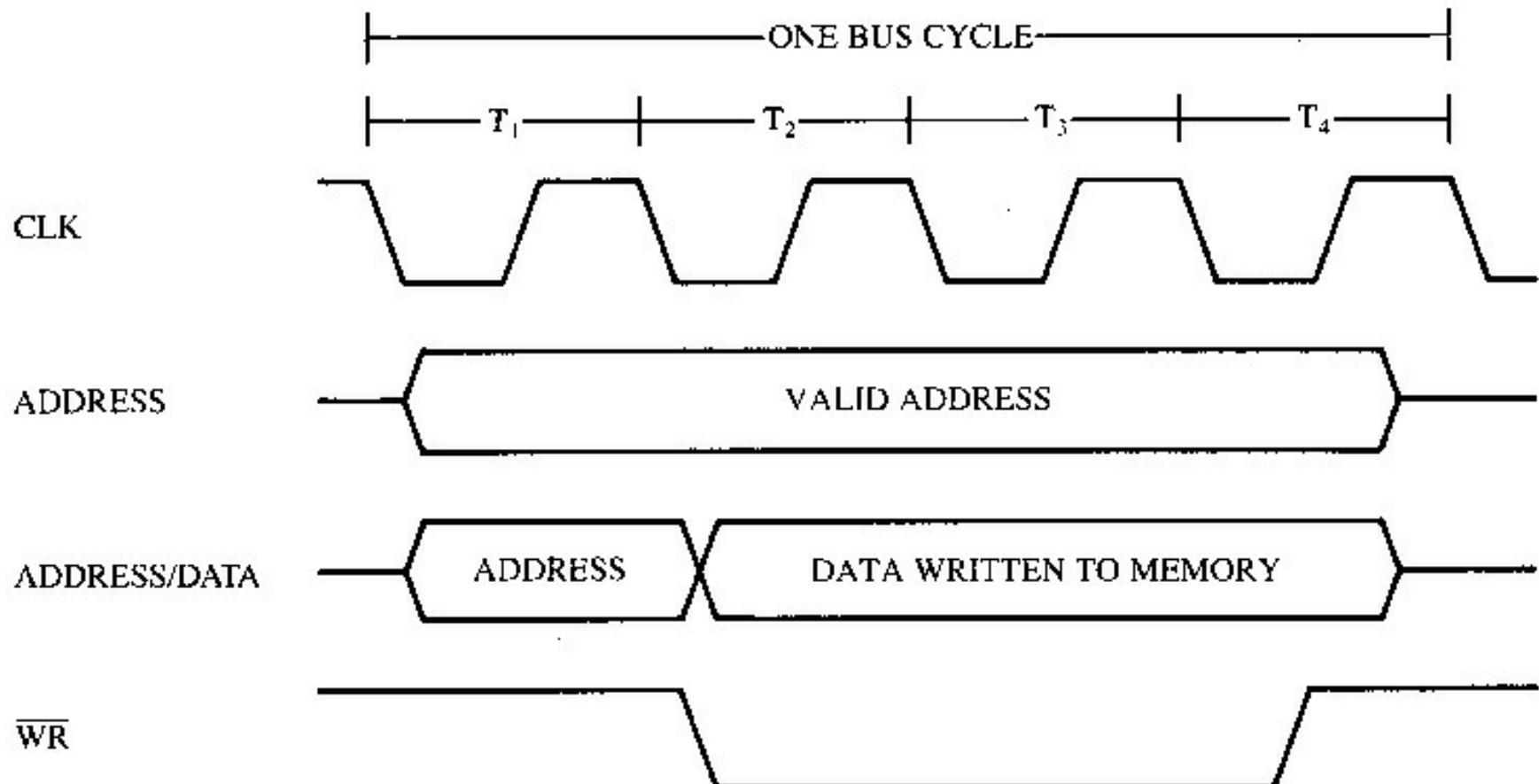


# Basic Bus Operation

- The three buses of 8086/8088 function the same way as any other microprocessor.
- If data are **written to memory** the processor:
  - **outputs** the memory address on the address bus
  - **outputs** the data to be written on the data bus
  - **issues** a write ( $\overline{WR}$ ) to memory
  - and  $IO/\overline{M} = 0$  for 8088 and  $IO/\overline{M} = 1$  for 8086
- See simplified timing for write in Fig 9–9.



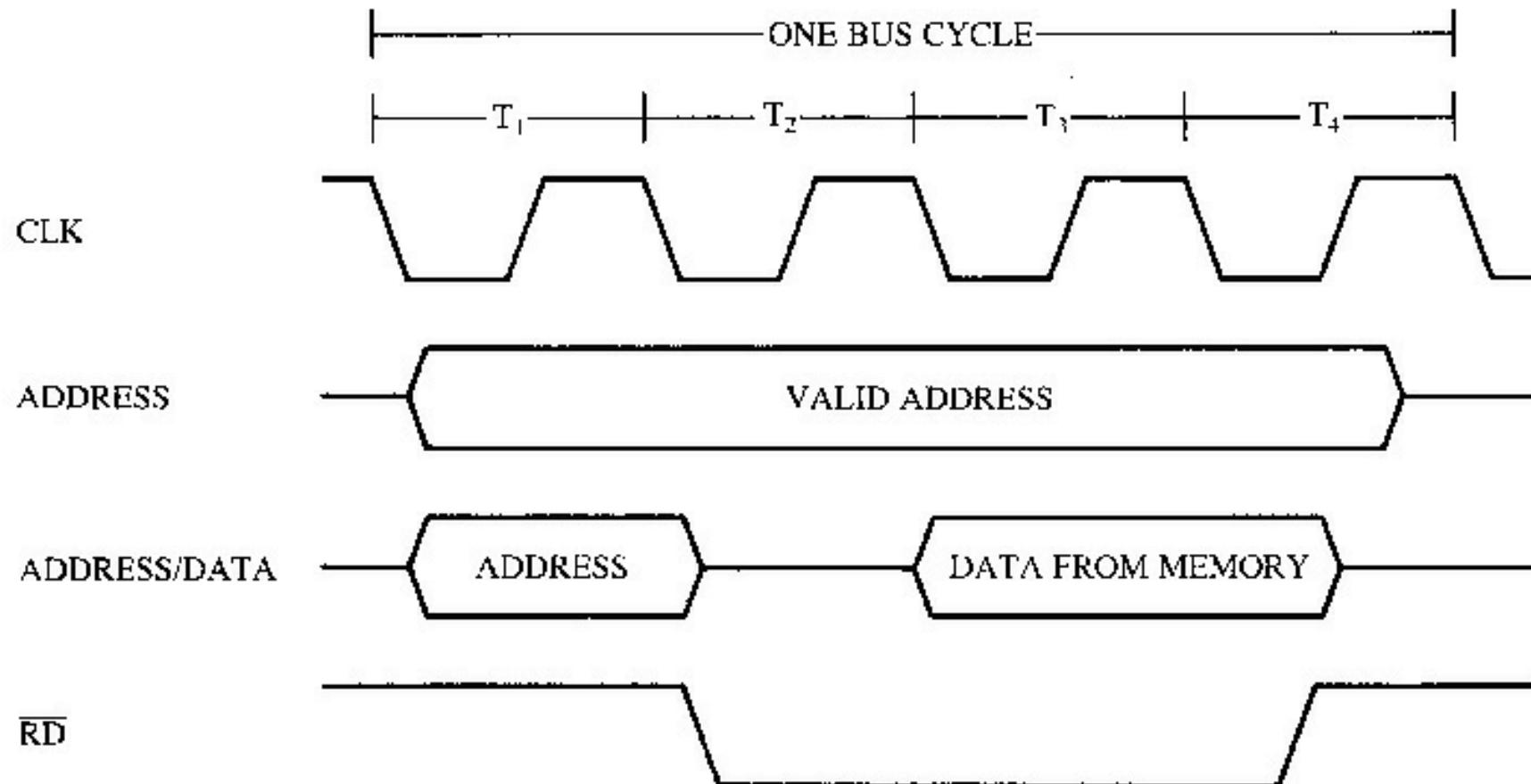
# Figure 9–9 Simplified 8086/8088 write bus cycle.



- If data are read from the memory the microprocessor:
  - outputs the memory address on the address bus
  - issues a read memory signal ( $\overline{RD}$ )
  - and accepts the data via the data bus
- See simplified timing for read in Fig 9–10.



# Figure 9–10 Simplified 8086/8088 read bus cycle.



# Timing in General

- 8086/8088 use memory and I/O in periods called bus cycles.
- Each cycle equals **four** system-clocking periods (**T states**).
  - **newer** microprocessors divide the bus cycle into as few as **two** clocking periods
- If the clock is operated at 5 MHz, one 8086/8088 bus cycle is complete in 800 ns.
  - basic operating frequency for these processors



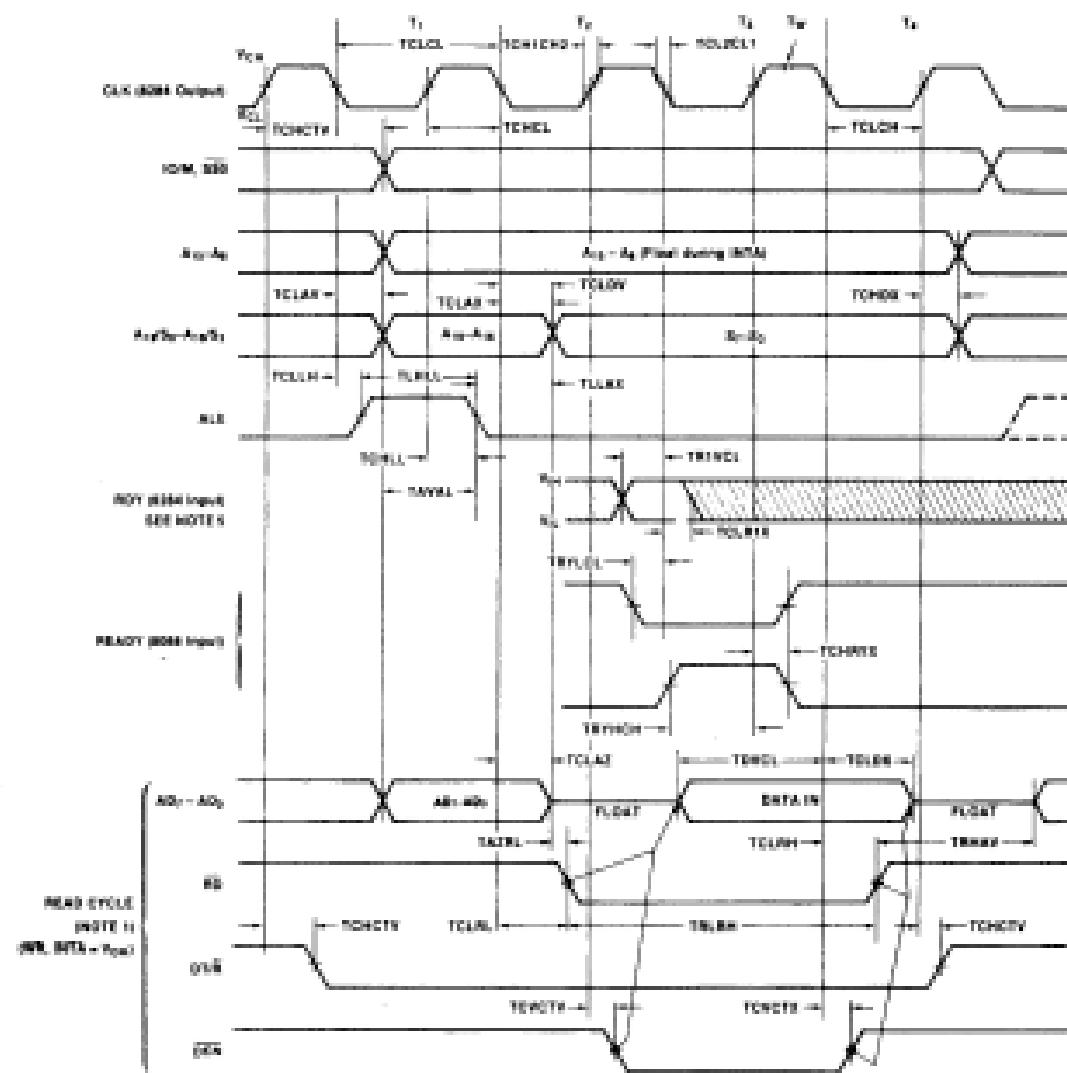
- During the **first** clocking period in a bus cycle, called **T1**, many things happen:
  - the **address** of the memory or I/O location is sent out via the address bus and the address/data bus connections.
- During **T1**, **control signals** are also output.
  - indicating whether the address bus contains a memory address or an I/O device (port) number
- During **T2**, the processors issue the **RD** or **WR** signal, **DEN**, and in the case of a write, the data to be written appear on the data bus.



- These events cause the memory or I/O device to begin to perform a read or a write.
- **READY** is sampled at the end of  $T_2$ .
  - if low at this time,  $T_3$  becomes a wait state ( $T_w$ )
  - this clocking period is provided to allow the memory time to access data
- If a read bus cycle, the data bus is sampled at the end of  $T_3$ .
- Illustrated in Figure 9–11.



**Figure 9–11** Minimum mode 8088 bus timing for a read operation.



In  $T_4$ , all bus signals are deactivated in preparation for the next bus cycle  
data bus connections are sampled for data read from memory or I/O

# Read Timing

- Figure 9–11 also depicts 8088 **read timing**.
  - 8086 has 16 rather than eight data bus bits
- Important item in the read timing diagram is time allowed for memory & I/O to read data.
- Memory is chosen by its access time.
  - the fixed amount of time the microprocessor allows it to access data for the read operation
- It is extremely important that memory chosen complies with the limitations of the system.



- The microprocessor timing diagram does not provide a listing for memory access time.
  - necessary to combine several times to arrive at the access time
- Memory **access time** starts when the address appears on the memory address bus and continues until the microprocessor **samples** the memory **data** at  $T_3$ .
  - about three T states elapse between these times
- The address does not appear until  $T_{CLAV}$  time (110 ns if a 5 MHz clock) after the start of  $T_1$ .



- $T_{CLAV}$  time must be subtracted from the three clocking states (600 ns) separating the appearance of the address ( $T_1$ ) and the sampling of the data ( $T_3$ ).
- The **data setup time** ( $T_{DVCL}$ ), which occurs before  $T_3$  must also be subtracted.
- Memory access time is thus three clocking states minus the sum of  $T_{CLAV}$  and  $T_{DVCL}$ .
- Because  $T_{DVCL}$  is 30 ns with a 5 MHz clock, the allowed memory access time is only 460 ns (access time =  $600 \text{ ns} - 110 \text{ ns} - 30 \text{ ns}$ )



Figure 9–12 8088 AC characteristics.

A.C. CHARACTERISTICS:  $t_{CK-1} = t_{IOH} \text{ to } t_{IOL}$ ,  $t_{CK-2} = t_{VDD} + t_{DQH}$   
 $t_{CK-3} = t_{IOH} \text{ to } t_{IOL}$ ,  $t_{CK-4} = t_{VDD} + t_{DQH}$

MINIMUM COMPLEXITY SYSTEM TIMING REQUIREMENTS

Symbol	Parameter	8086		8088-2		Test Conditions
		Min.	Max.	Min.	Max.	
T <sub>CKL</sub>	CLK Cycle Period	200	300	125	300	ns
T <sub>CLOCK</sub>	CLK Low Time	118		88		ns
T <sub>CHOL</sub>	CLK High Time	88		88		ns
T <sub>CHIGH2</sub>	CLK Rise Time		10		10	From 1.0V to 3.0V
T <sub>CLHOL1</sub>	CLK Fall Time		10		10	From 3.0V to 1.0V
T <sub>DVOL</sub>	Data in Setup Time	20		20		ns
T <sub>DUSH</sub>	Data in Hold Time	10		10		ns
T <sub>RDY<sub>0</sub></sub>	READY Setup Time into 8086 (See Notes 1, 2)	20		20		ns
T <sub>RDY<sub>1X</sub></sub>	READY Hold Time into 8086 (See Notes 1, 2)	0		0		ns
T <sub>RWCH</sub>	READY Setup Time into 8088	118		88		ns
T <sub>RWHY</sub>	READY Hold Time into 8088	20		20		ns
T <sub>RWHL</sub>	READY Inactive (in CLK) (See Note 3)	-8		-8		ns
T <sub>RWH</sub>	HOLD Setup Time	20		20		ns
T <sub>RTPI, RAS, TEST</sub>	RTPI, RAS, TEST Setup Time (See Note 2)	20		20		ns
T <sub>RIH</sub>	Input Rise Time (Except CLK)		80		20	From 0.0V to 3.0V
T <sub>RFH</sub>	Input Fall Time (Except CLK)		12		10	From 3.0V to 0.0V

- To find memory access time in this diagram:
  - locate the point in  $T_3$  when data are sampled
  - you will notice a line that extends from the end of  $T_3$  down to the data bus
  - at the end of  $T_3$ , the microprocessor samples the data bus.

- Memory devices chosen for connection to the 8086/8088 operating at 5 MHz must be able to access data in **less than 460 ns**.
  - because of the **time delay** introduced by the address decoders and buffers in the system
  - a 30- or 40-ns **margin** should exist for the operation of these circuits
- The memory speed should be **no slower than** about 420 ns to operate correctly with the 8086/8088 microprocessors.



# Strobe Width

- The other timing factor to affect memory operation is the width of the  $\overline{RD}$  strobe.
- On the timing diagram, the read strobe is given as  $T_{RLRH}$ .
- The time for this **strobe** at a **5 MHz** clock rate is **325 ns**.
- This is wide enough for almost all memory devices manufactured with an **access time** of **400 ns** or less.

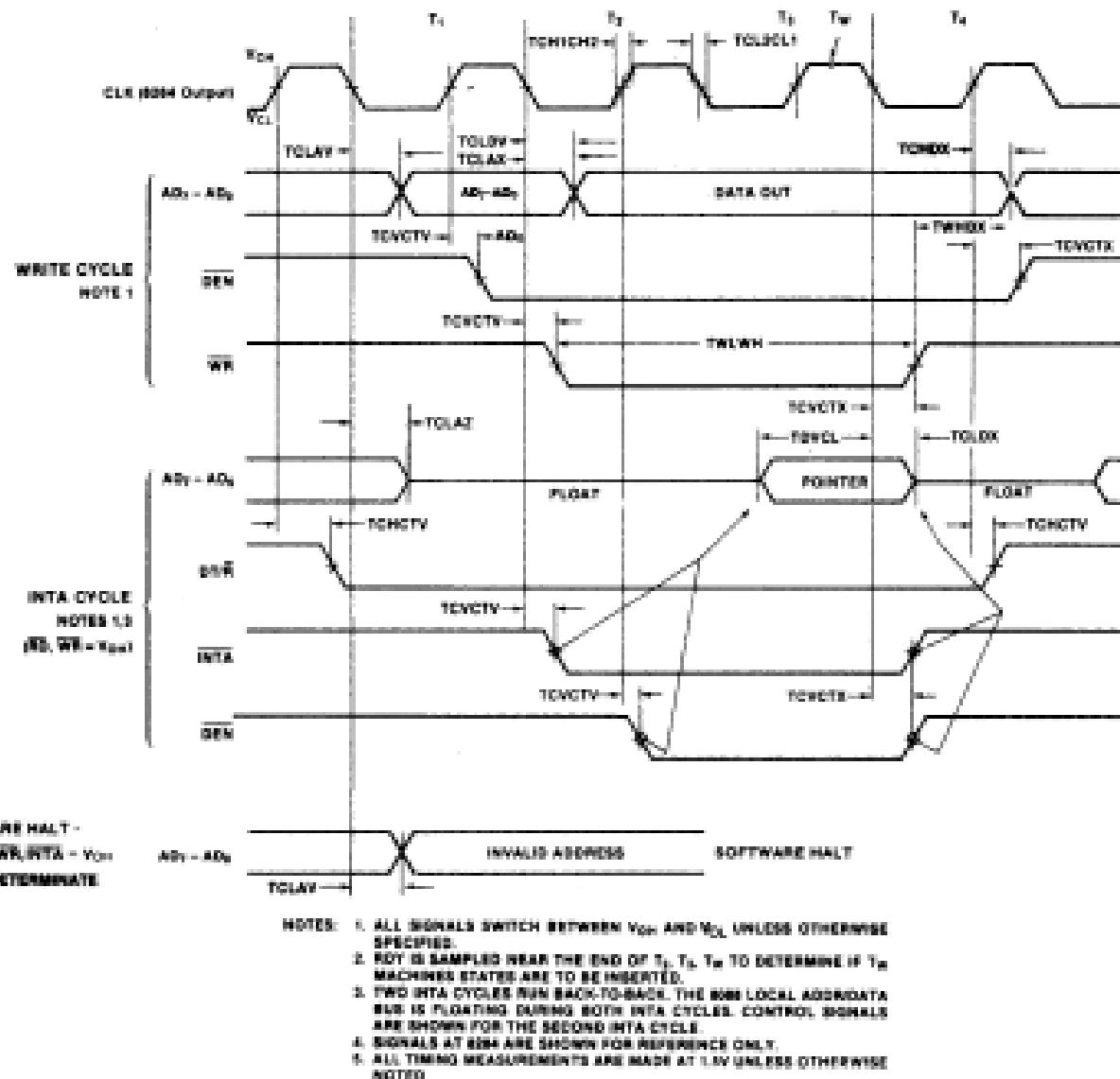


# Write Timing

- Figure 9–13 illustrates 8088 write-timing.
  - 8086 is nearly identical
- The  $\overline{RD}$  strobe is replaced by the  $\overline{WR}$  strobe,
  - the data bus contains information *for* the memory rather than information *from* the memory,
  - $DT/R$  remains a logic 1 instead of a logic 0 throughout the bus cycle
- When interfacing some devices, timing may be **critical** between when  $\overline{WR}$  becomes logic 1 and the data are removed from the data bus.



**Figure 9–13** Minimum mode 8088 write bus timing.



*The Intel Microprocessors: 8086/8088, 80186/80188, 80286, 80386, 80486 Pentium, Pentium Pro Processor, Pentium II, Pentium 4, and Core2 with 64-bit Extensions Architecture, Programming, and Interfacing*, Eighth Edition  
Barry B. Brey

Copyright ©2009 by Pearson Education, Inc.  
Upper Saddle River, New Jersey 07458 • All rights reserved.

- Memory data are written at the **trailing** edge of the **WR** strobe.
- On the diagram, this critical period is  $T_{WHDX}$  or 88 ns when 8088 on a 5 MHz clock.
- Hold time is often less than this.
  - in fact often 0 ns for memory devices
- The **width** of the **WR strobe** is  $T_{WLWH}$  or 340 ns with a 5 MHz clock.
- This rate is **compatible with** most memory devices with access time of 400 ns or less.



# 9–5 READY AND THE WAIT STATE

- The READY input causes wait states for slower memory and I/O components.
  - a **wait state** ( $T_w$ ) is an extra clocking period between  $T_2$  and  $T_3$  to lengthen bus cycle
  - on **one wait state**, memory access time of **460 ns**, is lengthened by one clocking period (**200 ns**) to **660 ns**, based on a 5 MHz clock
- This section covers **READY** synchronization circuitry inside the 8284A clock generator.

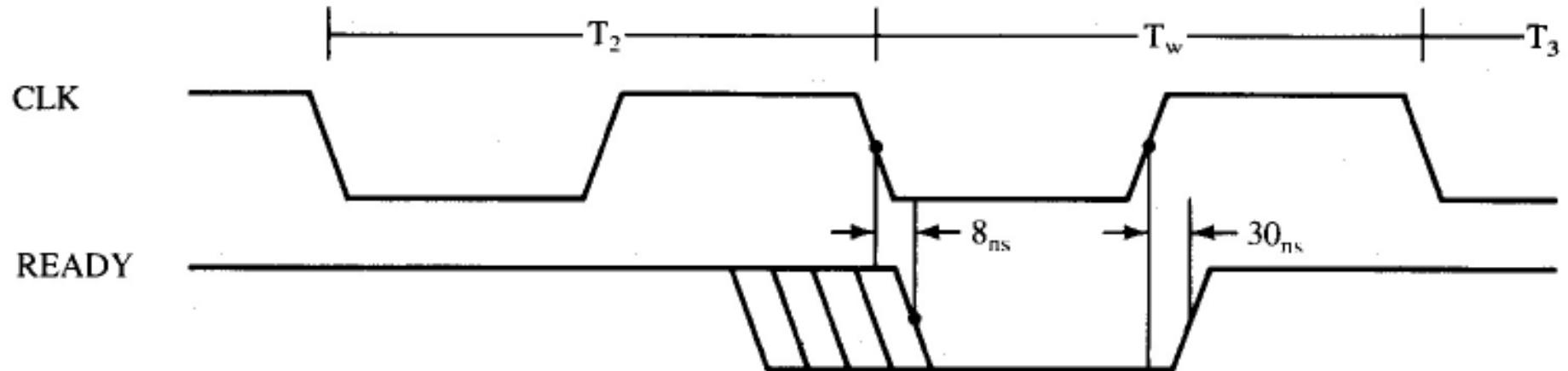


# The READY Input

- The READY input is sampled at the end of  $T_2$  and again, if applicable, in the middle of  $T_w$ .
- The READY input to 8086/8088 has stringent timing requirements.
- Fig 9–14 shows READY causing one wait state ( $T_w$ ), with the required setup and hold times from the system clock.
- When the 8284A is used for READY, the RDY (ready input to 8284A) input occurs at the end of each T state.



**Figure 9–14** 8086/8088 READY input timing.



- If READY is logic 0 at the end of  $T_2$ ,  $T_3$  is delayed and  $T_w$  inserted between  $T_2$  and  $T_3$ .
- READY is next sampled at the middle of  $T_w$  to determine if the next state is  $T_w$  or  $T_3$ .

# RDY and the 8284A

- RDY is the synchronized ready input to the 8284A clock generator.
- Internal 8284A circuitry guarantees the **accuracy** of the READY synchronization.

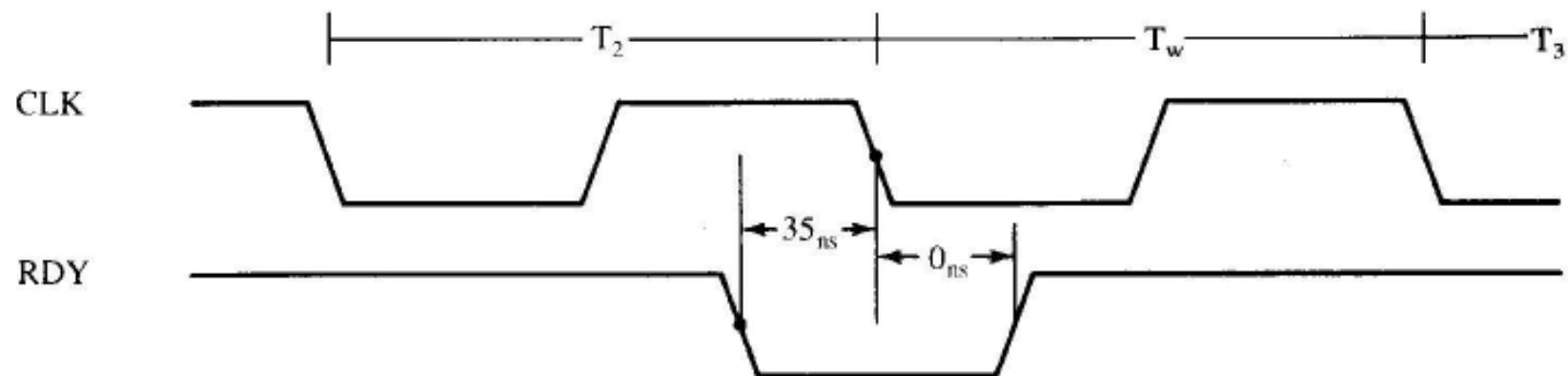


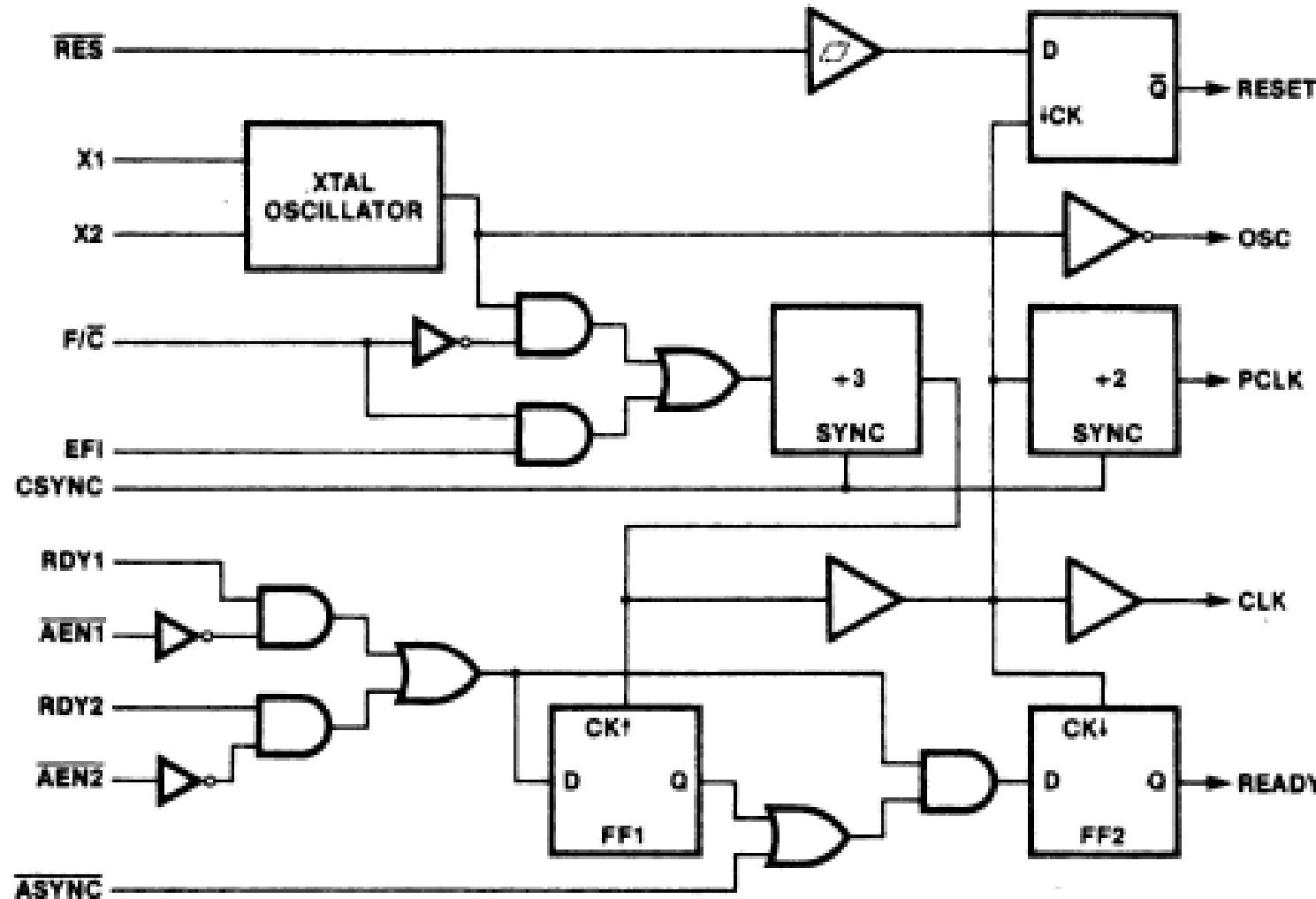
Figure 9–15 8284A RDY input timing.



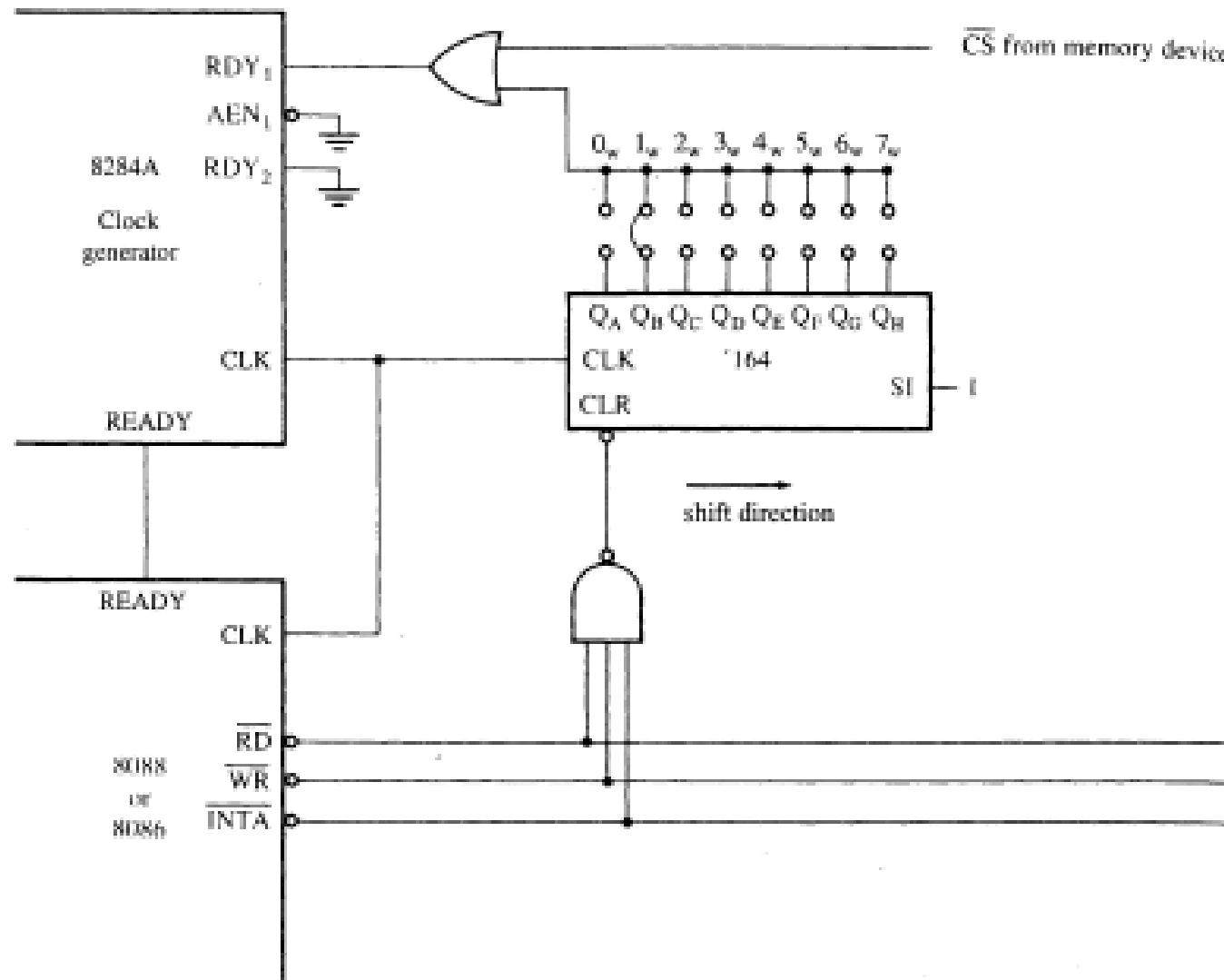
- Fig 9–16 depicts internal structure of 8284A.
  - the bottom half is the READY synch circuitry
- Fig 9–17 shows a circuit to introduce almost any number of wait states to 8086/8088.
- An 8-bit serial **shift** register (74LS164) shifts a logic 0 for one or more clock periods from one of its Q outputs through to the RDY1 input of the 8284A.
- With appropriate strapping, this circuit can provide **various numbers** of wait states.



**Figure 9–16** The internal block diagram of the 8284A clock generator. (Courtesy of Intel Corporation.)



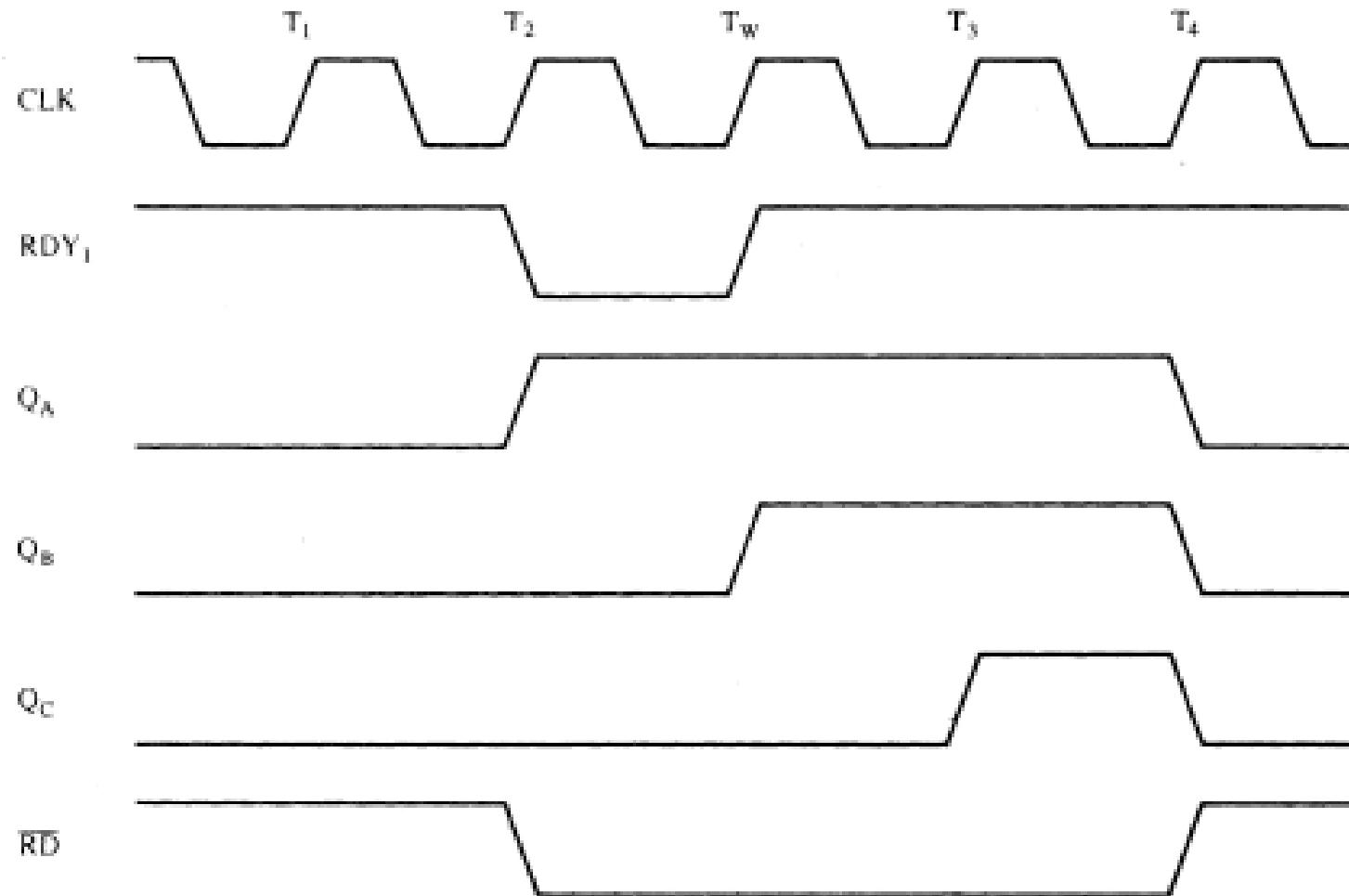
**Figure 9–17** A circuit that will cause between 0 and 7 wait states.



- Note in Fig 9–17 that this circuit is **enabled** only for devices that need **insertion of waits**.
  - if the selection signal is a logic 0, the device is selected and this circuit generates a wait state
- Figure 9–18 shows timing of this **shift register** wait state generator when wired to insert one wait state.
- The timing diagram also illustrates the internal contents of the shift register's flip-flops
  - to present a more detailed view of its operation
- **In this example, one wait state is generated.**



**Figure 9–18** Wait state generation timing of the circuit of Figure 9–17.



# 9–6 MINIMUM VS MAXIMUM MODE

- Minimum mode is obtained by connecting the mode selection **MN/MX** pin to +5.0 V,
  - maximum mode selected by grounding the pin
- The mode of operation provided by minimum mode is similar to that of the 8085A
  - the most recent Intel 8-bit microprocessor
- Maximum mode is designed to be used whenever a **coprocessor** exists in a system.
  - maximum mode was dropped with 80286

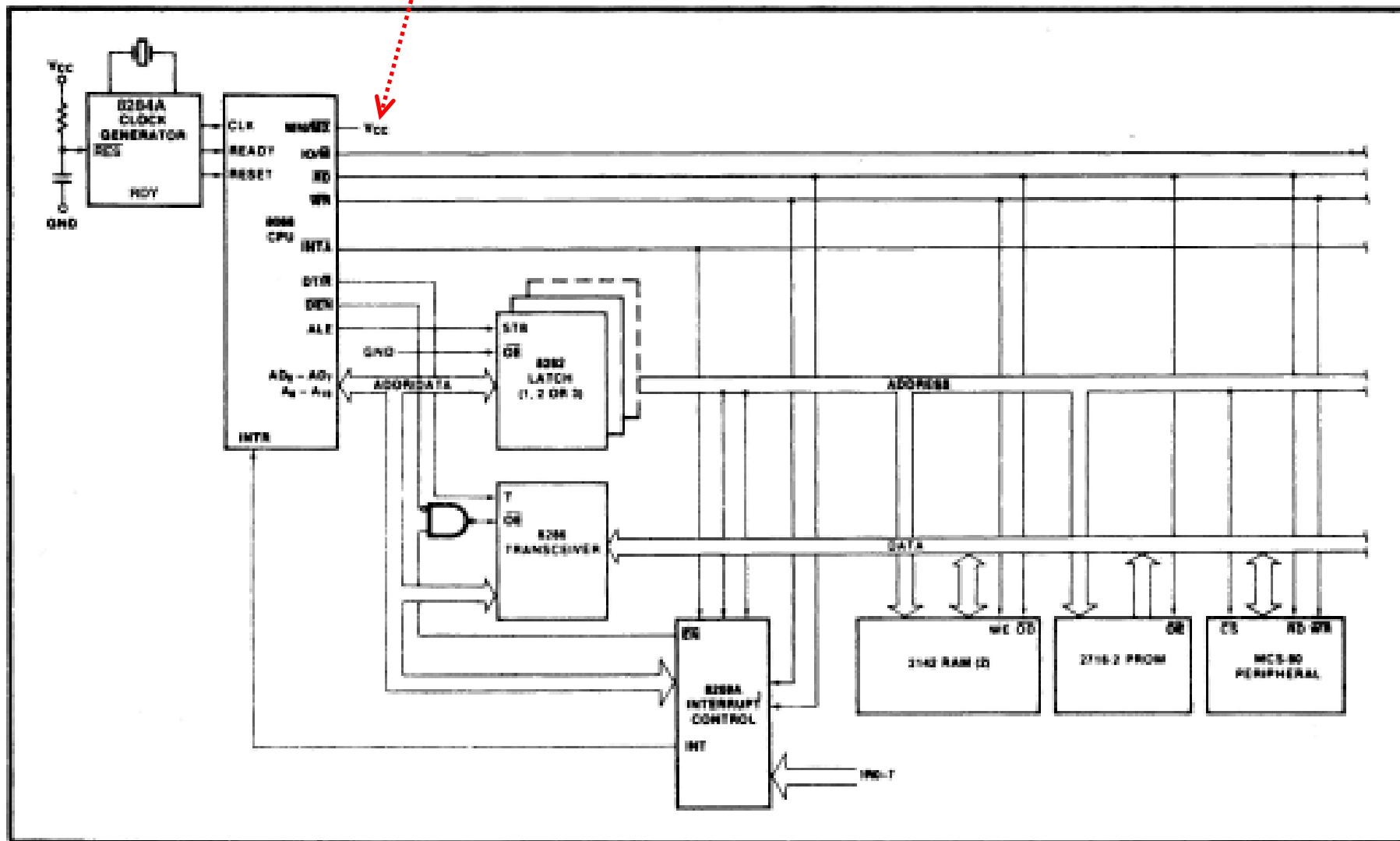


# Minimum Mode Operation

- Least expensive way to operate 8086/8088.
  - because all control signals for the memory & I/O are generated by the microprocessor
- Control signals are identical to Intel 8085A.
- The **minimum mode** allows 8085A **8-bit peripherals** to be used with the 8086/8088 without any special considerations.



**Figure 9–19 Minimum mode 8088 system.**

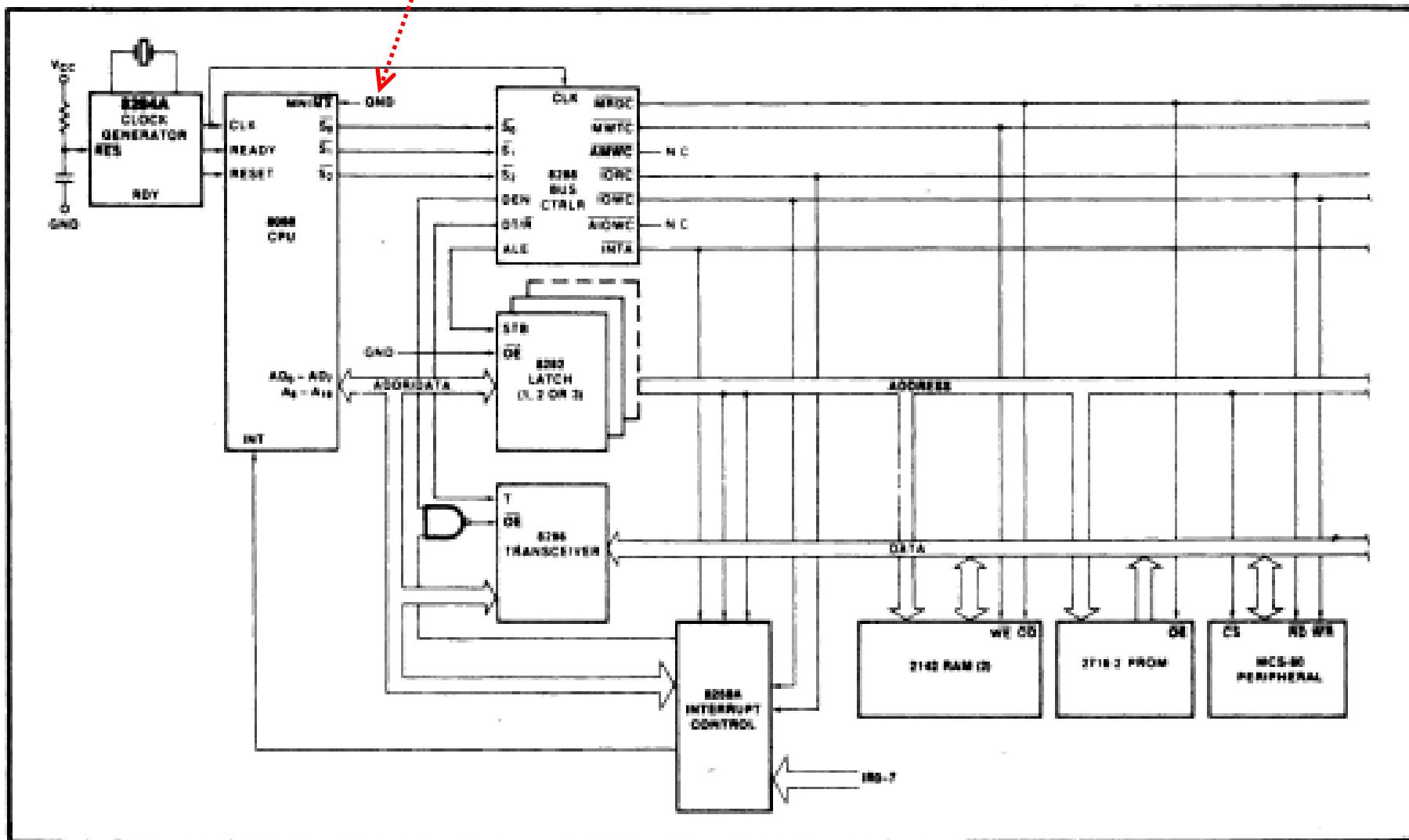


# Maximum Mode Operation

- Differs from **minimum mode** in that some control signals must be externally generated.
  - requires addition of the 8288 bus controller
- There are not enough pins on the 8086/8088 for bus control during maximum mode
  - new pins and features replaced some of them
- **Maximum mode** used only when the system contains **external coprocessors** such as 8087.



**Figure 9–20 Maximum mode 8088 system.**



# The 8288 Bus Controller

- Provides the signals eliminated from the 8086/8088 by the maximum mode operation.

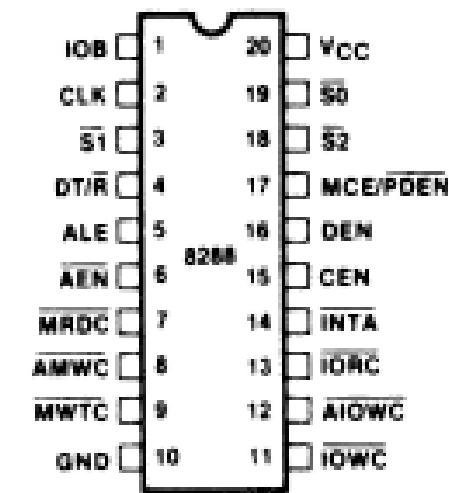
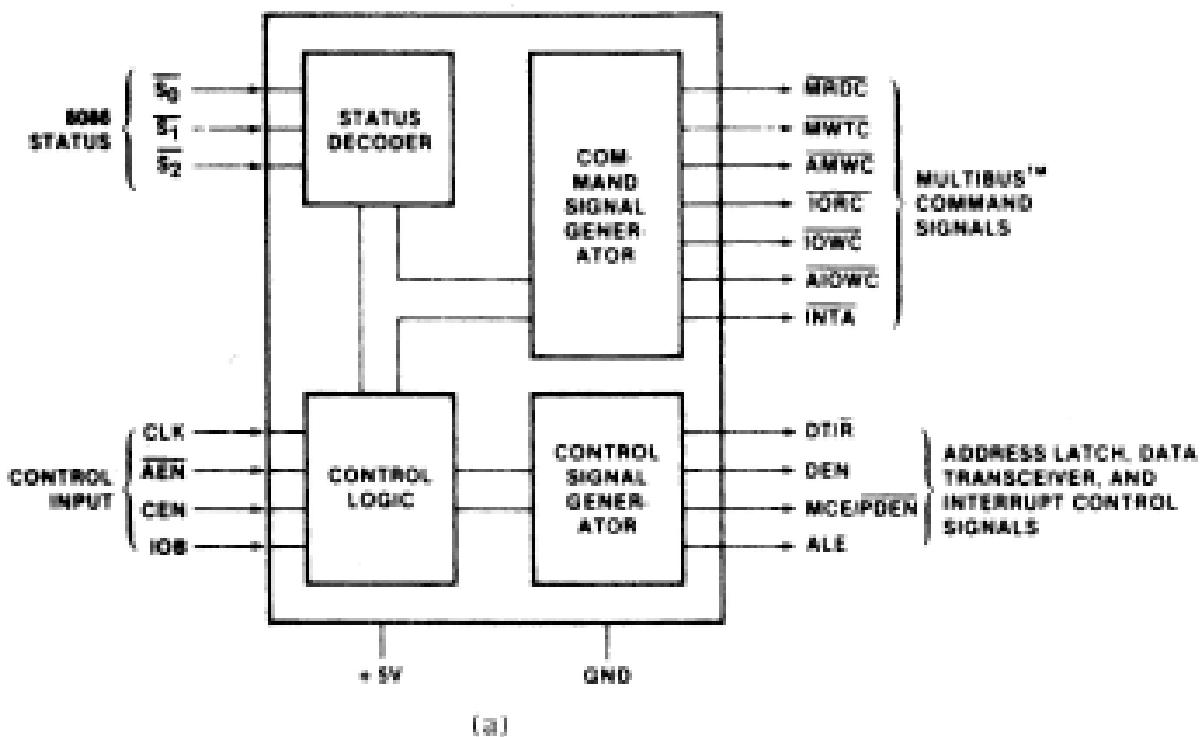


Figure 9–21 The 8288 bus controller; (a) block diagram and (b) pin-out.

# 8288 Bus Controller *Pin Functions*

## $S_2$ , $S_1$ , and $S_0$

- Status inputs are connected to the status output pins on 8086/8088.
  - three signals decoded to generate timing signals

## CLK

- The **clock** input provides internal timing.
  - must be connected to the CLK output pin of the 8284A clock generator



# 8288 Bus Controller *Pin Functions*

## ALE

- The **address latch enable** output is used to demultiplex the address/data bus.

## DEN

- The **data bus enable** pin controls the bidirectional data bus buffers in the system.

## DT/R

- **Data transmit/receive** signal output to control direction of the bidirectional data bus buffers.



# 8288 Bus Controller *Pin Functions*

## AEN

- The **address enable** input causes the 8288 to enable the memory control signals.

## CEN

- The **control enable** input enables the command output pins on the 8288.

## IOB

- The **I/O bus mode** input selects either I/O bus mode or system bus mode operation.



# 8288 Bus Controller *Pin Functions*

## AIOWC

- **Advanced I/O write** is a **command output** to an advanced I/O write control signal.

## TORC

- The **I/O read command** output provides I/O with its read control signal.

## IOWC

- The **I/O write command** output provides I/O with its main write signal.



# *8288 Pin Functions*

## AMWT

- **Advanced memory write** control pin provides memory with an early/advanced write signal.

## MWTC

- The **memory write** control pin provides memory with its normal write control signal.

## MRDC

- The **memory read** control pin provides memory with a read control signal.



# 8288 Bus Controller *Pin Functions*

## INTA

- The **interrupt acknowledge** output acknowledges an interrupt request input applied to the INTR pin.

## MCE/PDEN

- The **master cascade/peripheral data** output selects cascade operation for an interrupt controller if IOB is grounded, and enables the I/O bus transceivers if IOB is tied high.



# SUMMARY

- Both 8086 and 8088 require a single +5.0 V power supply with a **tolerance** of  $\pm 10\%$ .
- The 8086/8088 microprocessors are TTL-compatible if the **noise immunity** is derated to 350 mV from the customary 400 mV.
- The 8086/8088 microprocessors can drive **one** 74XX, **five** 74LSXX, **one** 74SXX, **ten** 74ALSXX, and **ten** 74HCXX unit loads.



# SUMMARY

(cont.)

- The 8284A clock generator provides the system clock (CLK), **READY** and **RESET** synchronization.
- The standard 5 MHz 8086/8088 operating frequency is obtained by attaching a 15 MHz crystal to the 8284A clock generator.
- The PCLK output contains a **TTL-compatible** signal at one half the CLK frequency.



# SUMMARY

(cont.)

- Whenever the 8086/8088 microprocessors are reset, they begin executing software at memory location FFFF0H (FFFF:0000) with the interrupt request pin disabled.
- Because the 8086/8088 buses are multiplexed and most memory and I/O devices aren't, the system must be demultiplexed before interfacing with memory or I/O.



# SUMMARY

(cont.)

- Demultiplexing is accomplished by an 8-bit **latch** whose clock **pulse** is obtained from the **ALE** signal.
- In a large system, the buses must be buffered because the 8086/8088 microprocessors are capable of driving only **10 unit loads**, and large systems often have many more.



# SUMMARY

(cont.)

- Bus timing is very important to the remaining chapters in the text. A bus cycle that consists of four clocking periods acts as the basic system timing.
- Each bus cycle is able to read or write data between the microprocessor and the memory or I/O system.



# SUMMARY

(cont.)

- The 8086/8088 microprocessors allow the memory and I/O 460 ns to access data when they are operated with a 5 MHz clock.
- **Wait states** ( $T_w$ ) stretch the bus cycle by one or more clocking periods to allow the memory and I/O additional access time.
- Wait states are **inserted** by controlling the READY input to the 8086/8088. READY is sampled at the end of T2 and during  $T_w$ .



# SUMMARY

(cont.)

- Minimum mode operation is similar to that of the Intel 8085A microprocessor, whereas maximum mode operation is new and specifically designed for the operation of the 8087 arithmetic coprocessor.
- The 8288 bus controller must be used in the maximum mode to provide the control bus signals to the memory and I/O.



# SUMMARY

- This is because the maximum mode operation of the 8086/8088 removes some of the system's control signal lines in favor of control signals for the coprocessors.
- The 8288 **bus controller** reconstructs these removed control signals.



## Pin Functions

The following list provides a description of each pin of the 8288 bus controller.

<b>S<sub>2</sub>, S<sub>1</sub>, and S<sub>0</sub></b>	Status inputs are connected to the status output pins on the 8086/8088 microprocessor. These three signals are decoded to generate the timing signals for the system.
<b>CLK</b>	The <b>clock</b> input provides internal timing and must be connected to the CLK output pin of the 8284A clock generator.
<b>ALE</b>	The <b>address latch enable</b> output is used to demultiplex the address/data bus.
<b>DEN</b>	The <b>data bus enable</b> pin controls the bidirectional data bus buffers in the system. Note that this is an active high output pin that is the opposite polarity from the DEN signal found on the microprocessor when operated in the minimum mode.
<b>DT/R</b>	The <b>data transmit/receive</b> signal is output by the 8288 to control the direction of the bidirectional data bus buffers.
<b>AEN</b>	The <b>address enable</b> input causes the 8288 to enable the memory control signals.
<b>CEN</b>	The <b>control enable</b> input enables the command output pins on the 8288.
<b>IOB</b>	The <b>I/O bus mode</b> input selects either the I/O bus mode or system bus mode operation.
<b>AIOWC</b>	The <b>advanced I/O write</b> is a command output used to provide I/O with an advanced I/O write control signal.
<b>IORC</b>	The <b>I/O read command</b> output provides I/O with its read control signal.
<b>IOWC</b>	The <b>I/O write command</b> output provides I/O with its main write signal.
<b>AMWT</b>	The <b>advanced memory write control</b> pin provides memory with an early or advanced write signal.
<b>MWTC</b>	The <b>memory write control</b> pin provides memory with its normal write control signal.
<b>MRDC</b>	The <b>memory read control</b> pin provides memory with a read control signal.
<b>INTA</b>	The <b>interrupt acknowledge</b> output acknowledges an interrupt request input applied to the INTR pin.
<b>MCE/PDEN</b>	The <b>master cascade/peripheral data output</b> selects cascade operation for an interrupt controller if IOB is grounded, and enables the I/O bus transceivers if IOB is tied high.

DATE:

## Multiple - Register Transfer

LDMB+ - Load multiple stack Registers

STM - Store multiple Register

LD MFD - Load multiple full descending  
r13 / r0-r5 , pop from a full descending wrh

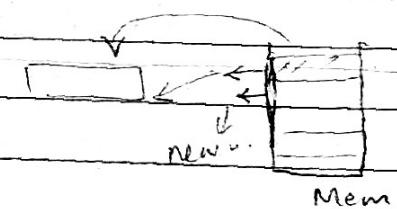
STMFD - Store multiple full descending . stam.  
STMFD r13! , {r0-r5} ; push into full

LD MFDA - Load multiple full Ascending . stam.  
descendin order

STMFA - Store multiple full ascending.

Syntax : <LDM | STM> {word} [Addressing mode] Rn{  
<Registers>

LDM - Load multiple register ;



{Rd}\*N ← mem32 [Start-address + 4\*N]

STM - Save multiple register ;

{Rd}\*N → mem32

[Start-Address + 4\*N]

BA = 0x14 to R0.

BA = 0x18

Rn (?) → Write back  
operation  
Base address

$$16 + 8 = \underline{\underline{24}}$$

$$4 \times 2 = 8 \\ 8 \\ \underline{\underline{16}}$$

DATE: 28 04 2020

Block Data Transfer (d<sup>m</sup>) ① 20

List of registers in {}.

- \* Load - store multiple instructions, can transfer multiple registers between memory and the processor in a single instruction.
- \* Multiple-register transfer instructions are more efficient from single-register transfers for moving Blocks of Data. (more than 4-bytes of memory)
- \* ARM implementations do not usually interrupt instructions while they are executing.

Read or

Text Area

Data area.

$$5 \times 4 = 20 \text{ bytes}$$

Labels ref to memory.

R0 = Init A address.

Example for LOMIA

Here      LMIA r0!, {r1-r3}

Load Multiple Instructions from memory to  
the register

- ① Here data will be initially available in memory location.
- ② This data will be copied from the memory location to the registers specified in the instruction.
- ③ Since this is 32 bit operation, each data size would be of 4 bytes.

(Defines words each with 4 bytes alignment)

memory loca $\rightarrow$  x      DCD    0x 11 11 11 11

$\downarrow$        $\downarrow$   
2 byte - 1 byte

y      DCD    0x 22 22 22 22

z      DCD    0x 33 33 33 33

Example for LD MIA

Here LD MIA r0!, {r1-r3}

Load Multiple Instructions from memory to  
the register

- ① Here data will be initially available in memory location.
- ② This data will be copied from the memory location to the registers specified in the instruction.
- ③ Since this is 32 bit operation, each data size would be of 4 bytes.

(Defines words each with 4 bytes alignment)

memory loca $\rightarrow$  X DCD 0x 1111 1111  
 $\downarrow$   $\downarrow$   
2bytes 1byte

y DCD 0x 22 22 22 22

z DGD 0x 33 33 33 33

(3)

DATE:    

Here  $r_0 \rightarrow$  base address ( $r_0$  will be pointing to the base address)

$\rightarrow$  write back operation

After execution of the instruction

Say if  $r_0$  is pointing to a memory location  $x$

i.e  $r_0 = x$

~~$\rightarrow$~~   $\rightarrow$  is at mem32 (0x80018)

After the transfer of each 32 bit-size data  
the memory location of  $r_0$  will be incremented by  
4 bytes.

In this example

After transferring  $3 \times 4$  bytes.

$= 12$  bytes

The memory location  $r_0$  will be pointing to:

$= 0x80018 + 12$

and the register contents will be

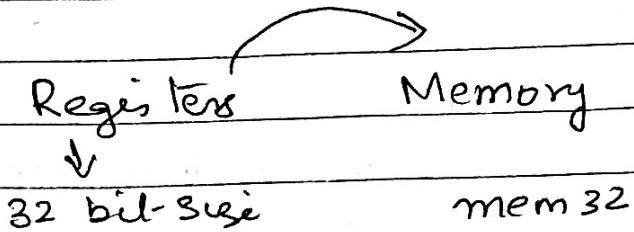
$r_3 = 0x1111\ 1111$

$r_2 = 0x2222\ 2222$

$r_3 = 0x3333\ 3333$

Example 2: STMIB and LDMA

STMIB → Store multiple registers to memory location



Before each transfer the memory location will be incremented by 4 bytes  
After each transfer the memory location will be incremented by 4 bytes

Syntax: ~~STMIB~~

STMIB r0! {r1 - r3}

(5)

DATE:

## Stack operation

- Pop operation uses a multiple load instruction
- PUSH operation uses multiple store instructions
  - SP points to empty location
- E → Empty stack (to check if stack is empty)
- F → Full stack → SP points to last-used or full

Stack would be in Ascending 'A' → Stack grows towards higher order memory

Descending 'D' → Stack decrements and moves towards lower order memory

to check stack is full STMFD → Store multiple registers full Descending  
STMFA → Store multiple registers full Ascending

to check empty STMEA → Store multiple registers empty Ascending  
STMED → Store multiple registers empty Descending

Example:

Before

STMFD SP! {r1, r4}

r1 = 0x 00000002

r4 = 0x 00000003

SP =

After execution?

Similarly

STMFD r13!, {r2-r9} → saves registers  
onto stack

LDMF D r13! {r2-r9} → restore from  
stack

Example:

STMFD SP! {r1, r4}

r1 = 0x 00000002

r4 = 0x 00000003

SP =

in execution?

Similarly

STMFD r13!, {r2-r9} → saves registers  
onto stack

LDMFD r13! {r2-r9} → restores from  
stack

DATE:

SWAP Instruction : Swap a byte between memory and a register.

- ① It is a special load store instruction
- ② It swaps the contents of memory with contents of registers.
- ③ It reads and writes in the same bus operation
- ④ It prevents other instructions from reading or writing to that location till until it completes (atomic operation)

Syntax:

SWP {B} {<cond>} Rd, Rm, [Rn];

[Rn]  $\rightarrow$  Rm,

Rm  $\rightarrow$  Rn

\*  $\text{temp} = \text{mem8[Rn]}$

$\text{mem8[Rn]} = \text{Rm}$

$\text{Rd} = \text{temp}$

DATE:

Write an ARM program to execute  
the following instruction

$$r0 = 0x00000000$$

$$\begin{array}{l} \text{R0: } \\ \text{R1: } \end{array} r1 = 0x11112222$$

$$\begin{array}{l} \text{R2: } \\ \text{R3: } \end{array} r2 = 0x0000\ 9000$$

SWP r0, r1, [r2]

After execution?