WK 3

Chapter 9

8086/8088 Hardware Specifications

Objectives

- Describe the functions of all 8086/8088 pins
- Understand DC characteristics:
  → Voltage levels and Noise margin
  → Current levels and Fan-out
- Use the clock generator 8284A chip
- Connect buffers and latches to the buses
- Interpret timing diagrams
- Describe wait states and design their circuits
- Explain the differences between minimum and maximum modes

The 8086/8088

- Fairly old microprocessors, but still considered a good way to introduce the Intel family
- Both microprocessors use 16-bit registers and data bus and 20-bit address bus (supporting

  1 MB memory), but:

  - The 8086 (1978): 16-bit external data bus:

    Memory required two “byte banks”

  - The 8088 (1979): 8-bit external data bus:

    Memory required One “byte bank”
- Still used in embedded systems (cost is less than $1)

Each processor
Can operate in two
modes

Maximum mode

Operation with a
Math Coprocessor

Minimum mode

Basic
Operation

I/P Selects
Min/Max Mode

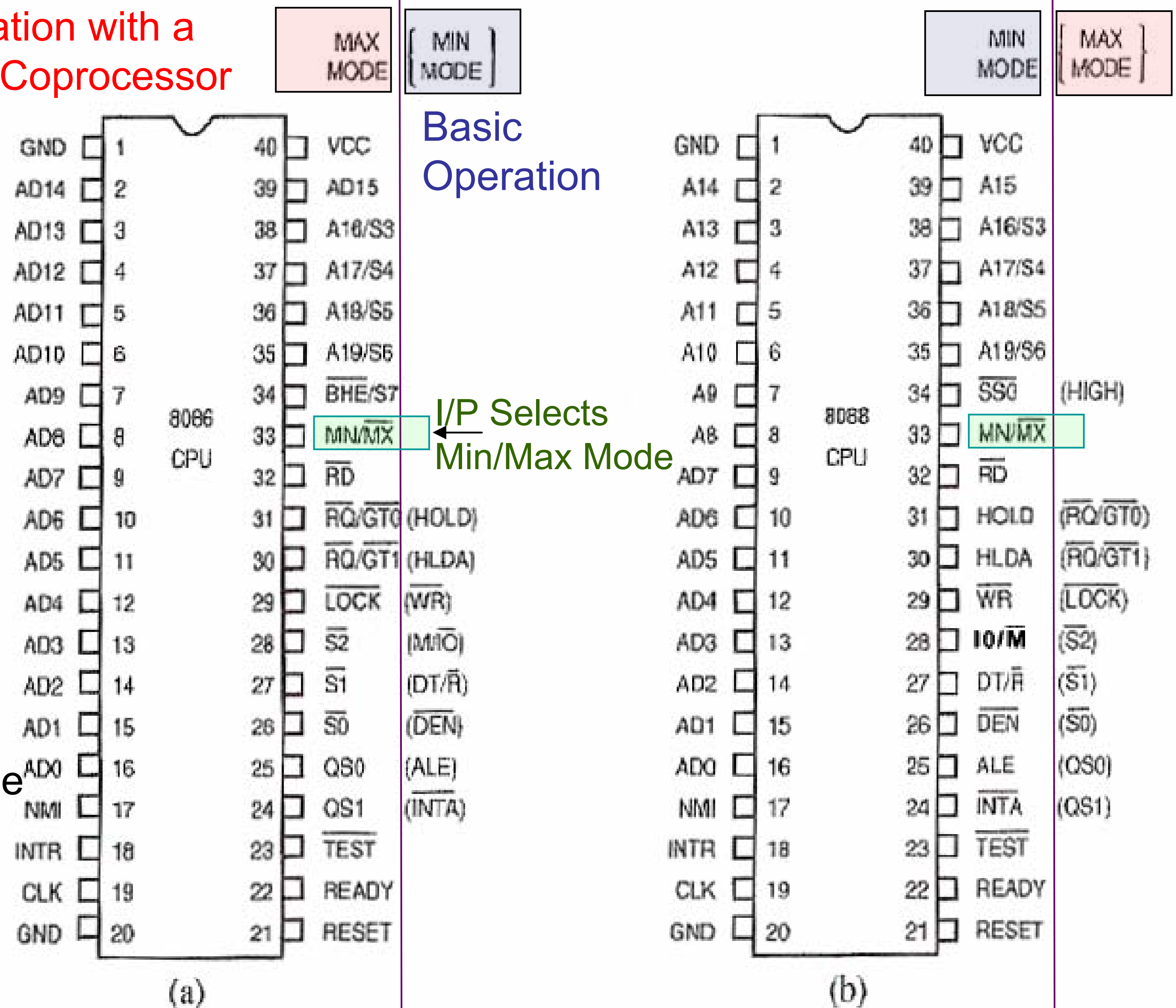
Pin budget:
8086, Min mode:

20 Address
16 Data
20 Control & Status
3 Power

---

59 Total
> The 40 pins available

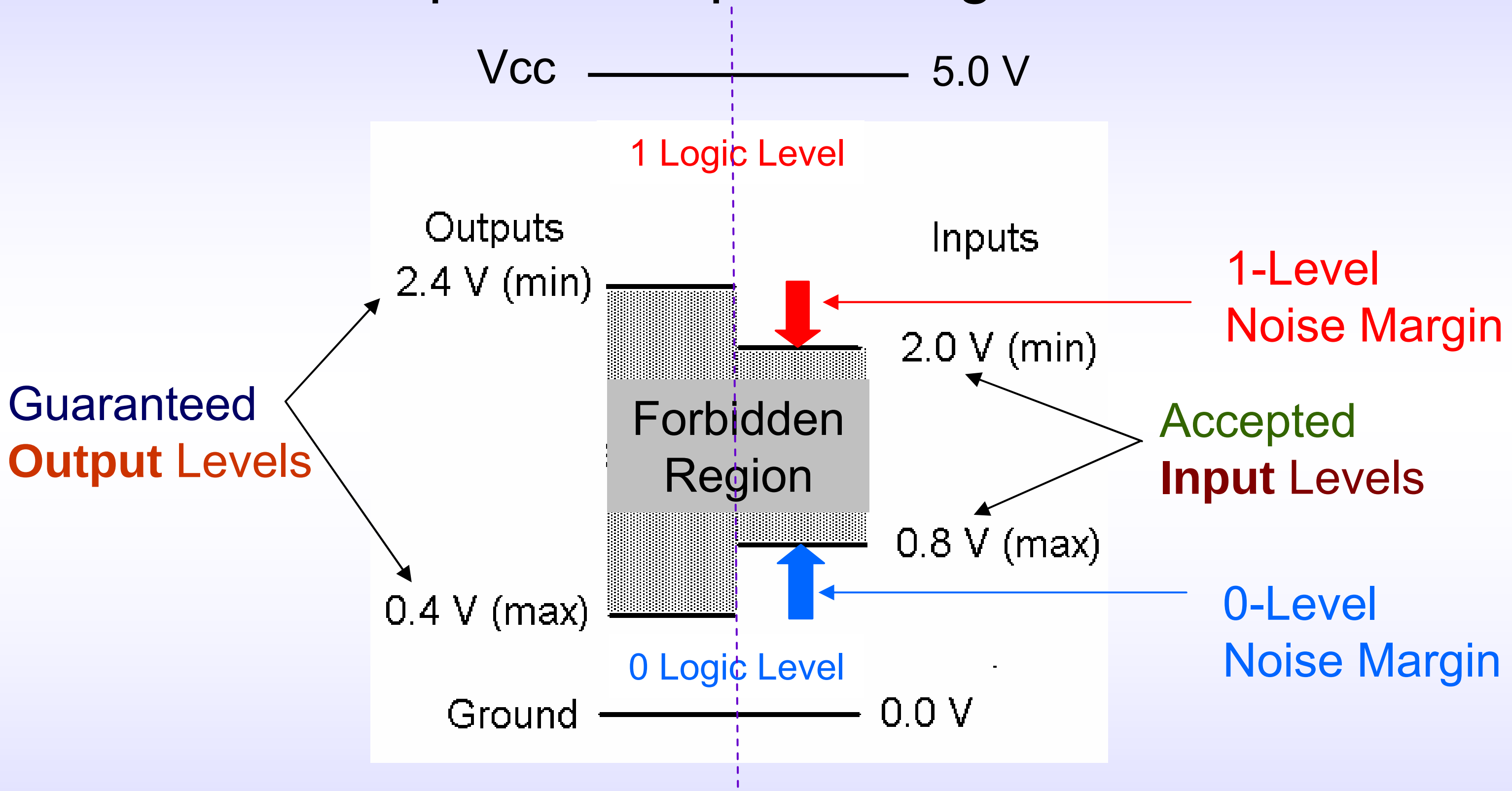
→ Use multiplexing,
e.g. ADi, A16/S3

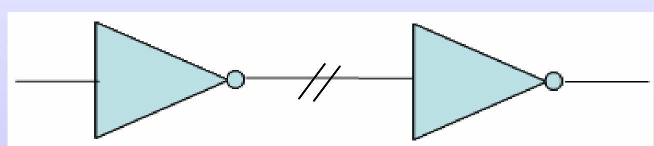


8086 8088

DC Pin Characteristics:
Voltage Levels & Noise Margins

Standard TTL Output and Input Voltage Levels





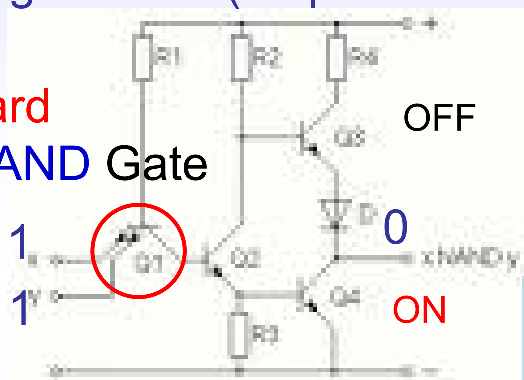
DC Pin Characteristics:
Current Levels & Fan-out

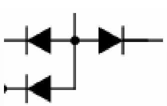
Fan out for a standard TTL output
How many inputs can an output support?

Source: Current out of pin
Sink: Current into pin

For the 0 logic Level: (output “sinks” current)

Standard
TTL NAND Gate

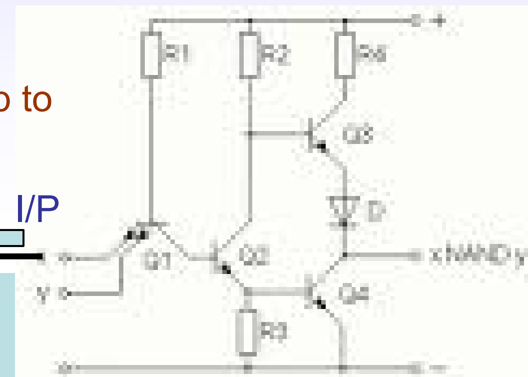


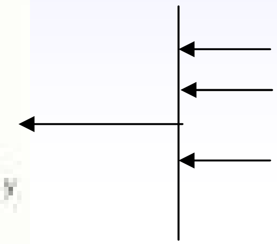


O/P can sink up to 16 mA max

I/P

An I/P sources up to 1.6 mA

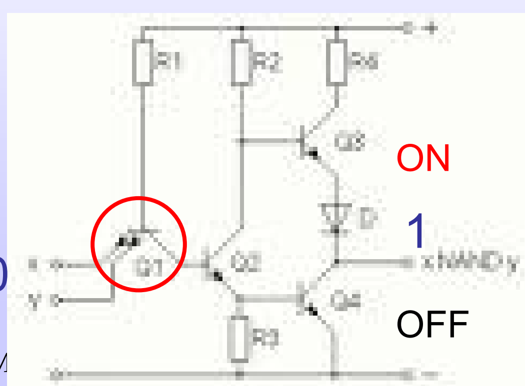




0-level Fanout = Maximum number of inputs that the output can support
= 16 mA/1.6 mA = 10

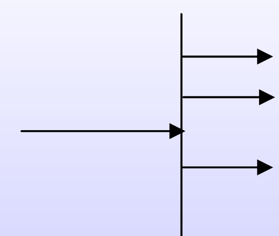
For the 1 logic Level: (output “sources” current)

Any 0



O/P can source up to 400 μA max

I/P sinks up to 40 μA



If different, take fan-out as the smallest of the two numbers

1-Level fan out = 10 also

8088/86 Pin Characteristics: DC

Output pins — Guaranteed Output levels

Logic Level	Voltage	Current
0	0.45 V maximum #	2.0 mA maximum *
1	2.4 V minimum	-400 uA maximum

* = 16 mA for standard 74 TTL

\# = 0.40 V for standard 74 TTL

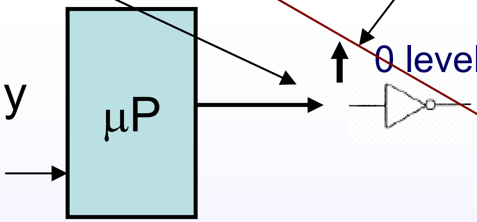
Input pins — Accepted Input levels

Logic Level	Voltage	Current
0	0.8 V maximum	-10 uA maximum *
1	2.0 V minimum	+10 uA maximum #

* = 1.6 mA for standard 74 TTL

\# = 40 µA for standard 74 TTL

8086/88 µp does not strictly comply with the DC characteristics of the TTL family



0 level noise margin = 0.8 – 0.45 = 0.35 V (µP)
= 0.8 – 0.40 = 0.40 V
(for standard 74 TTL O/P)

0 level fan-out to a TTL gate = 2 ÷ 1.6 ≈ 1 (8086/88 µP)
= 16 ÷ 1.6 = 10
(for standard 74 TTL O/P)

A processor output can drive only:

- **One 74XX input, or**
- **One 74SXX input, or**
- **Five 74LSXX inputs, or**
- **Ten 74ALSXX inputs, or**
- **Ten 74HCXX inputs**

Better or worse than standard TTL?

+: Current into pin (sink)

- : Current out of pin (source)

Two problems:

- Lower fan-out
- Lower noise margin

8088/86 Pin Characteristics: DC

- Input pins are TTL compatible and require only ±10μA of current (actually **better** than TTL inputs)
- Output pins are nearly TTL compatible, but have problems at logic 0:

  - A higher maximum logic 0 voltage of 0.45 V (instead of the TTL standard of 0.4 V)

  This **reduces** logic 0 noise margin from 400 mV to 350 mV…

  → So, be careful with long wiring from output pins

  - A lower logic 0 sinks a current of only 2.0 mA (instead 16 mA for the standard 74 TTL)

  This reduces fan out capability…Solutions:

  → Use 74LS, AL, or HC circuits for interfacing (they have a lower input current than standard 74 family circuits)

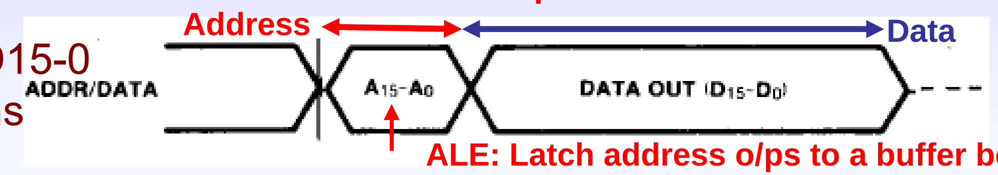
  → Or use **buffers**

The Buses: Address, Data, Status, Control

- Address, A (for memory & I/O)
- Data, D
- Status, S
- Control lines

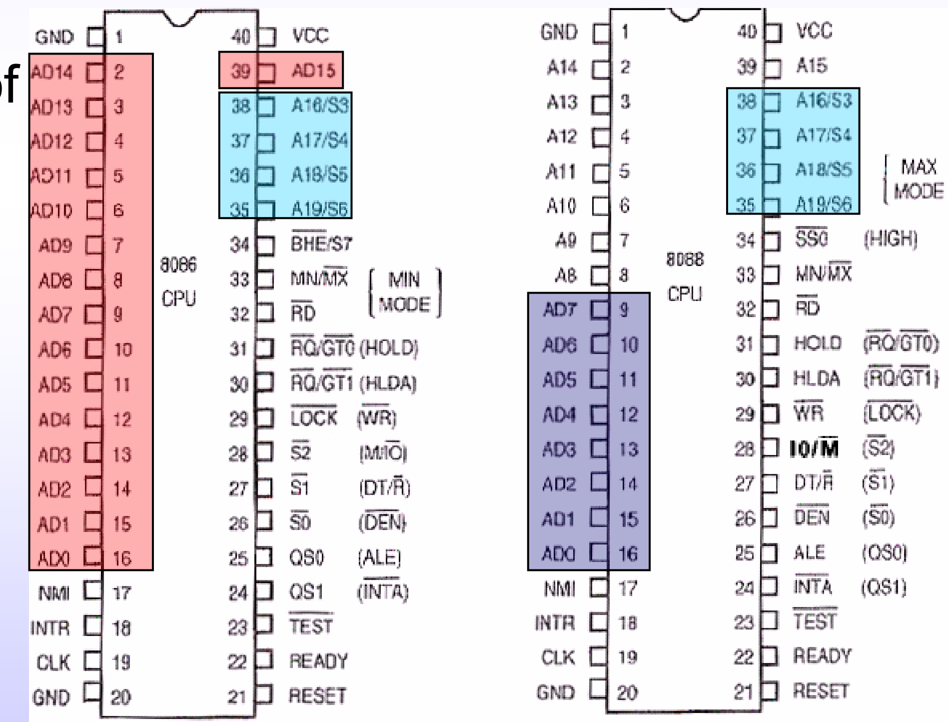
Some functions are multiplexed on the same pins to reduce chip pin count

AD15-0 pins



In a Write Cycle

- For both μPs: Address bus signals are A0-A19 (20 lines) for 1M byte of addressing space
- Data bus signals are
  - D0-D7 for the 8088 (8-bit)
  - D0-D15 for the 8086 (16-bit)
- The address & data pins are **multiplexed** as:
  - AD0-AD7 (8088)
  - or AD0-AD15 (8086)
- Address/Status pins are MUXed
  - A/S for A16-19 (both μPs)

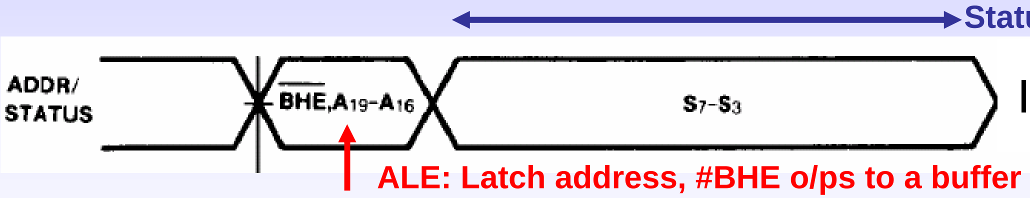


- The ALE O/P signal is used to **demultiplex** the address/data (AD) bus and also the address/status (A/S) bus.

The Status (S) Bus: 8 bits

86: #S0-S2, S3-S7
88: #S0-S2, S3-S6, #SS0

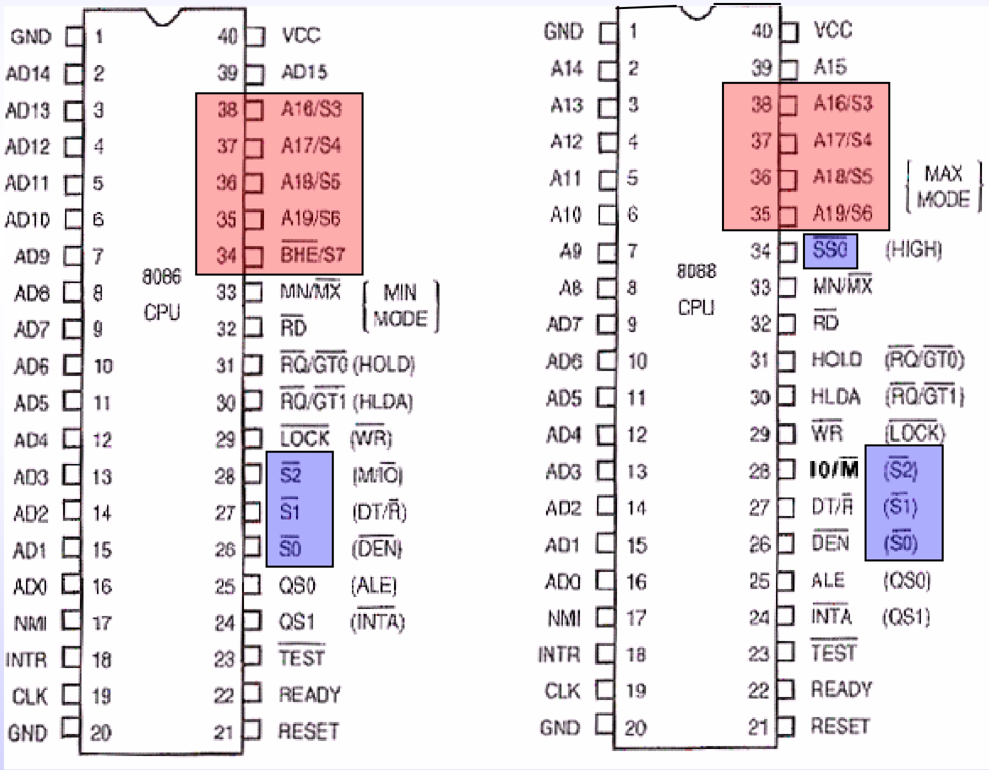
- 86: Address bits A16-A19 & #BHE: muxed with the status bits S3-S7.



Indicates status of processor and bus cycle

- S3 & S4 indicate which segment register is used with the current instruction:

S_4	S_3	Characteristics
0 (LOW)	0	Alternate Data (extra segment)
0	1	Stack
1 (HIGH)	0	Code or None
1	1	Data



- S5 = the IF (Interrupt flag) bit in FLAGS

- S6: 0
- S7: 1

Spare

#S0,1,2 are not MUXed. They encode bus status (current bus cycle)
Available only in the MAX mode for use by a bus controller chip

#SS0: Not Muxed, Min mode

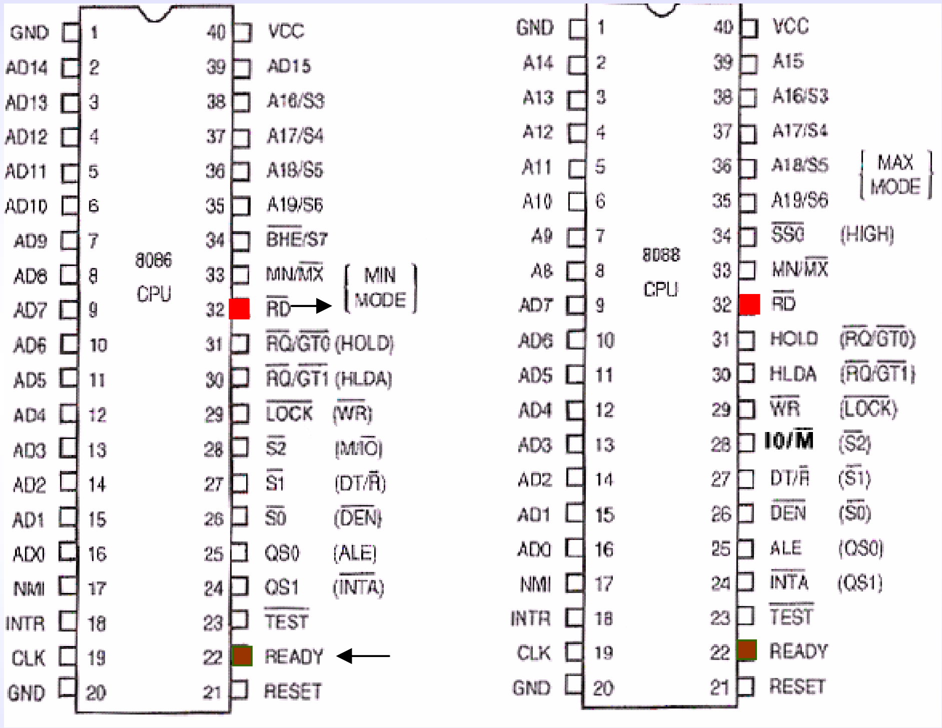
Control Bus: Main Control Signals

1. Signals that are common to both MIN and MAX modes:

- The read output (#RD) (i.e. RD): indicates a read operation

  Note: The write (#WR) output: indicates a write (a MIN mode output)

- The READY input: when low (= not ready), forces the processor to enter a wait state. Facilitates interfacing the processor to slow memory chips



or ___ = Active low signal

Main Control Signals, Contd.

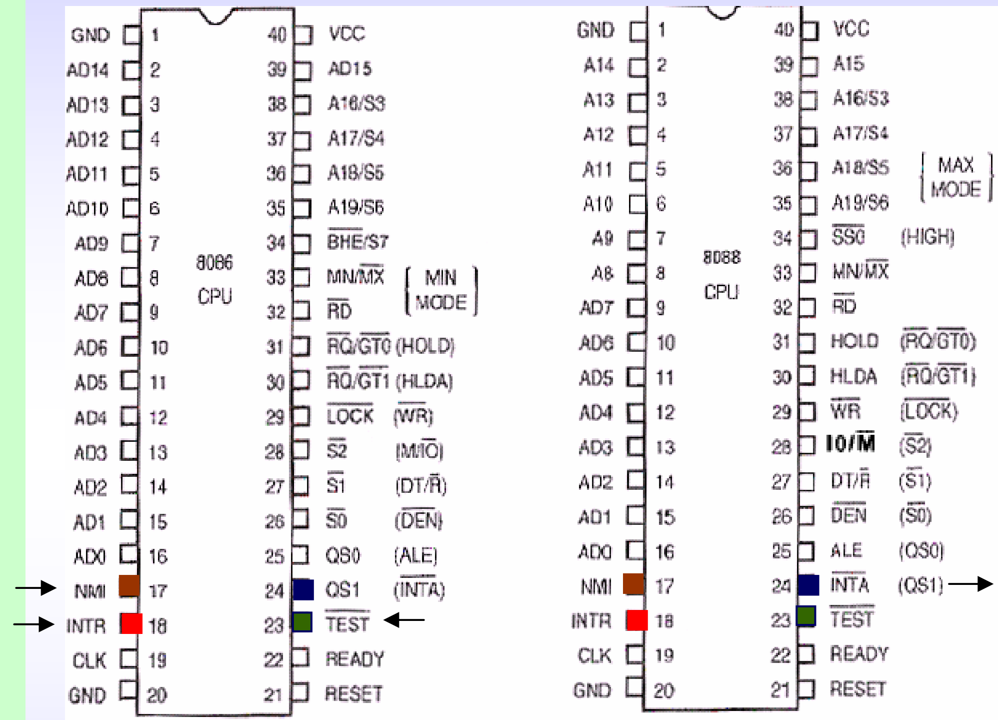
Two hardware interrupt inputs:

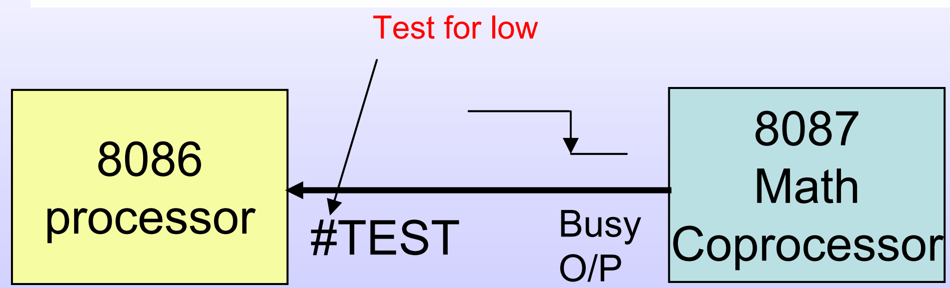
- INTR input: Hardware interrupt request. Entertained only if the IF flag is set. The µp enters an interrupt ACK cycle by lowering the #INTA output

  The IF flag bit is set (to enable interrupts) using the STI instruction, and cleared by CLI

- NMI input: Hardware non-maskable interrupt request. Entertained regardless of the status of the IF flag. Uses interrupt vector 2

- #TEST input: Example: interfacing the µP with the 8087 math coprocessor. Checked by the WAIT instruction that precedes each floating point instruction. If high, the instruction waits till the #TEST input signal goes low to determine that the FP math processor has finished





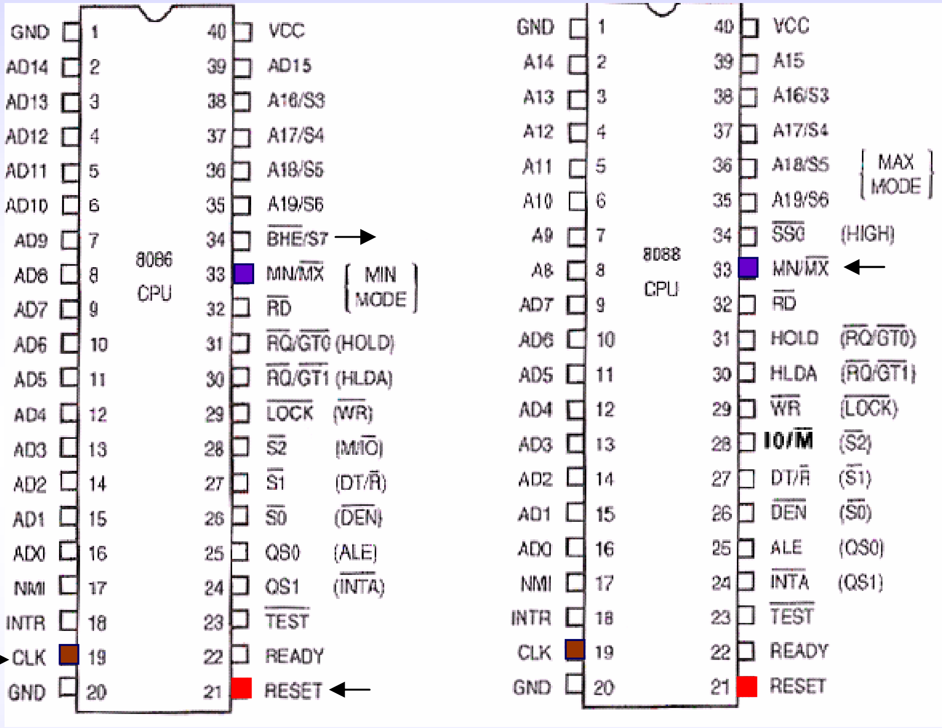
Synchronizes processor execution to external events

Main Control Signals, Contd.

- CLK input: Basic timing clock for the processor. 1:3 duty cycle
- MN/#MX input: Selects either Minimum (+ 5V directly) or Maximum mode (GND)
- #BHE/S7 output (MUXed):

  #BHE: (Bus High Enable) Enables writing to the high byte of the 16-bit data bus on the 8086

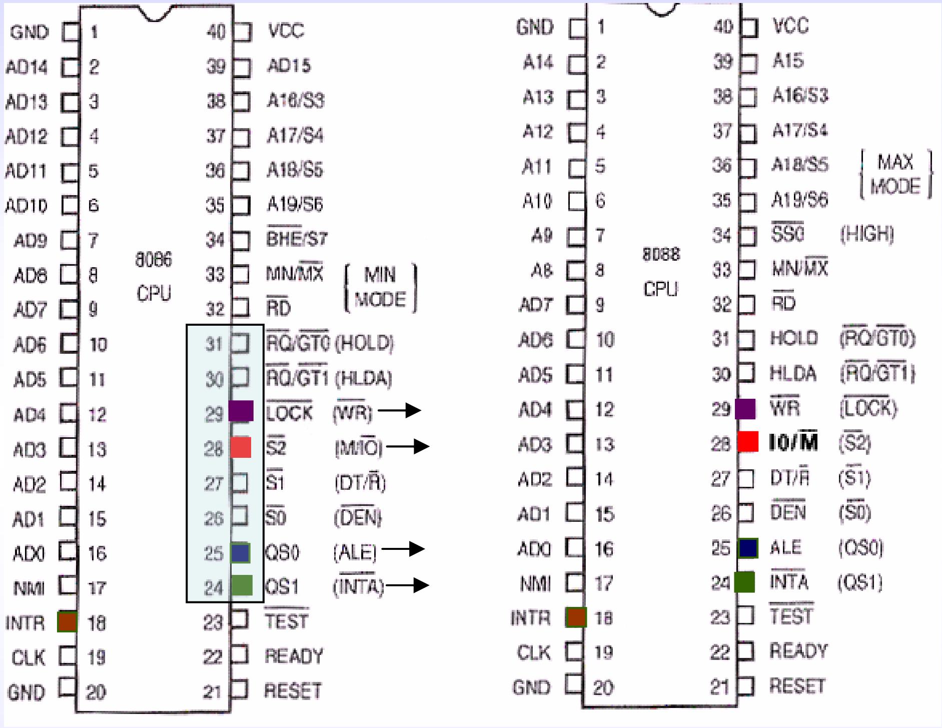
  Not on 8088 (it has only 8-bit data bus- no high byte!)
- RESET input: resets the microprocessor (reboots the computer). Causes the processor to start executing at address FFFF0H (Start of last 16 bytes of ROM at the top of the 1MB memory) after disabling the INTR input interrupts (CLR IF flag). Input must be kept high for at least 50 μs. Sampled by the processor at the + ive clock edge



2. Minimum Mode Signals

For the processor to operate in the minimum mode, connect MN/#MX input directly to +5V.

- M/#IO or IO/#M output: indicates whether the address on the address bus is a memory address (IO/#M = 0) or an I/O address (IO/#M = 1)
- #WR output: indicates a write operation.
- #INTA output: interrupt acknowledgement. Goes low in response to a hardware interrupt request applied to the INTR input. Interrupting device uses it to put the interrupt vector number on the data bus. The μp reads the number and identifies the ISR*
- ALE (address latch enable) output: Indicates that the muxed AD bus now carries address (memory or I/O). Use to latch that address to an external circuit before the processor removes it!.



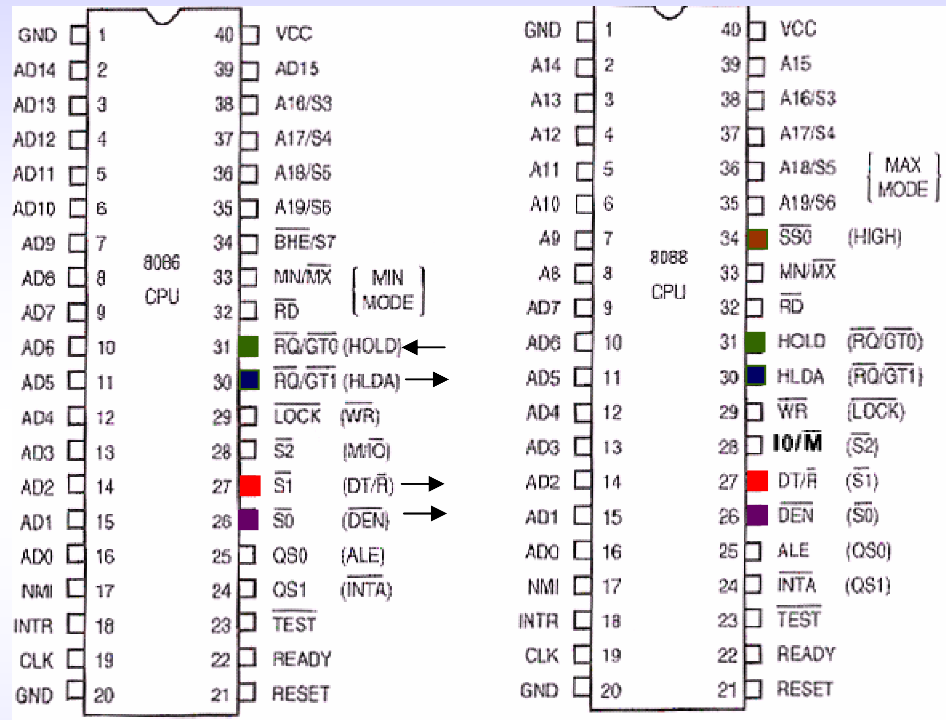
Note: Address on the bus can be either for memory or I/O devices. M/#IO signal indicates which

*ISR = Interrupt Service Routine

Minimum Mode Signals, Contd.

For the processor to operate in the minimum mode, connect MN/#MX input directly to +5V.

- DT/#R output: indicates if the data bus is transmitting (outputing) data (=1) or receiving (inputting) data (=0). Use to control external bidirectional buffers connected to the data bus.
- #DEN output: (data bus enable). Active when AD bus carries data not address. Use to activate external data buffers.
- HOLD input: Requests a direct memory access (DMA) from the μP. In response, the μP stops execution and places the data, address, and control buses at High Z state (floats them).
- HLDA output: Acknowledges that the processor has entered a hold state in response to HOLD.



- #SS0 output: Equivalent to the S0 status output of the maximum mode. Use with IO/#M and DT/#R to decode the current bus cycle (Table 9-5) (8088)

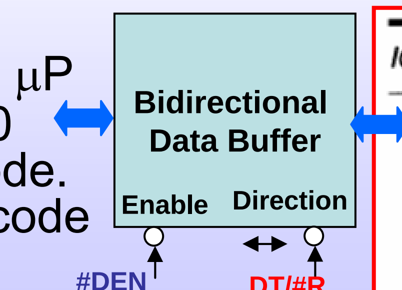


TABLE 9–5 Bus cycle status (8088) using $\overline{SS0}$.

$IO/\overline{M}$	$DT/\overline{R}$	$\overline{SS0}$	Function
0	0	0	Interrupt acknowledge
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

Multiprocessor System

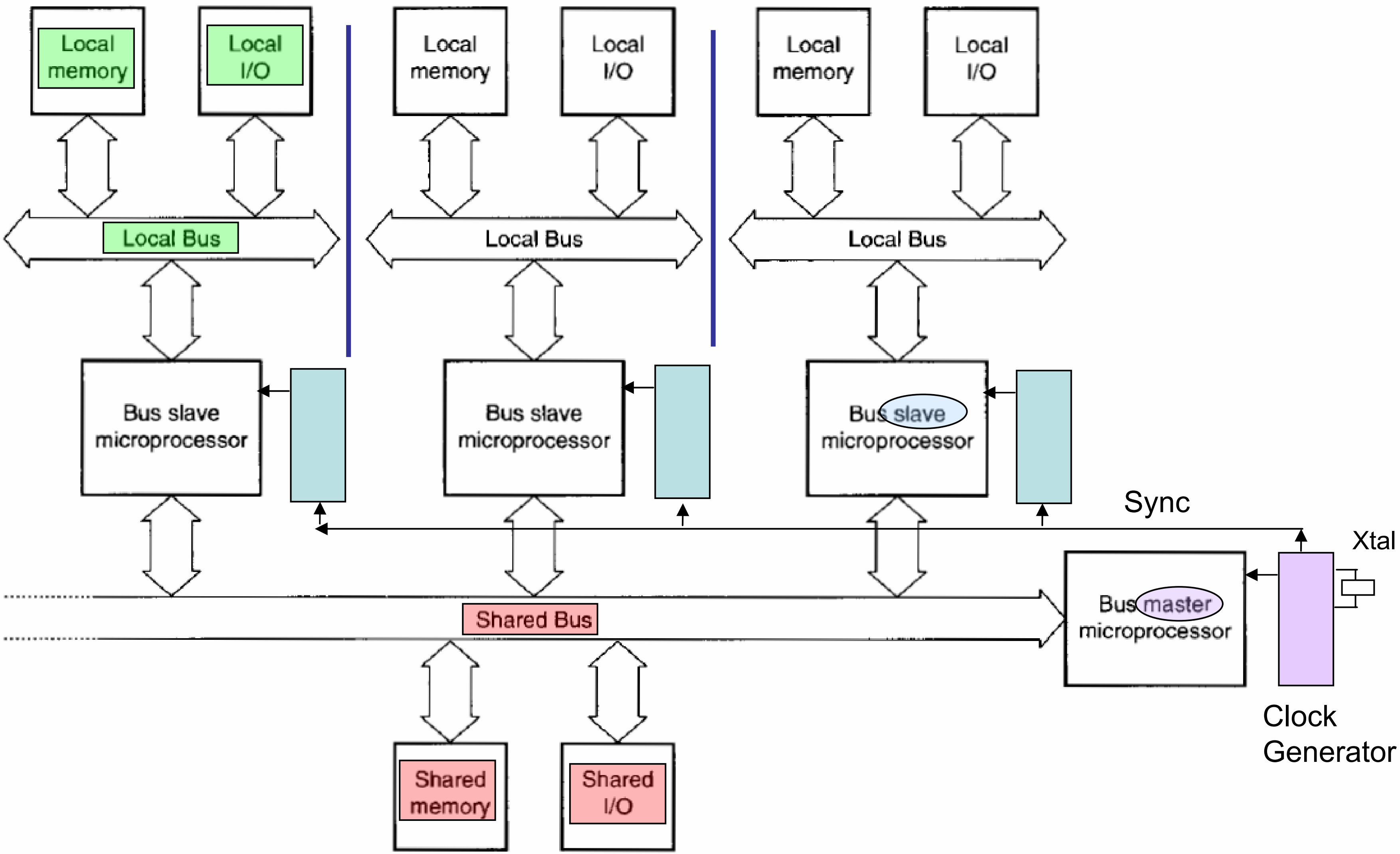


FIGURE 13–14 A block diagram illustrating the shared and local buses.

3. Maximum Mode Signals

For the processor to operate in the minimum mode, connect MN/#MX input to ground.

- #S0,#S1,#S2 outputs: Status bits that encode the type of the current bus cycle, Used by the 8288 bus controller and the 8087 coprocessor (Table 9-6) (3 Vs 8)
- #RQ/GT0, #RQ/GT1: Bidirectional lines for requesting and granting bus access (Request/Get). For use in multiprocessor systems. The RG/GT0 line has higher priority
- #LOCK output: Activated for the duration of μP instructions having the LOCK prefix. Can be used to prevent other microprocessors from using the system (shared) buses to access shared memory or I/O for the duration of such instructions, e.g.

  LOCK:MOV AL,[SI]

- QS0, QS1 (Queue Status) outputs: indicate the status of the internal instruction queue (Table 9-7). For use by the 8087 coprocessor to keep in step with the 8088/86

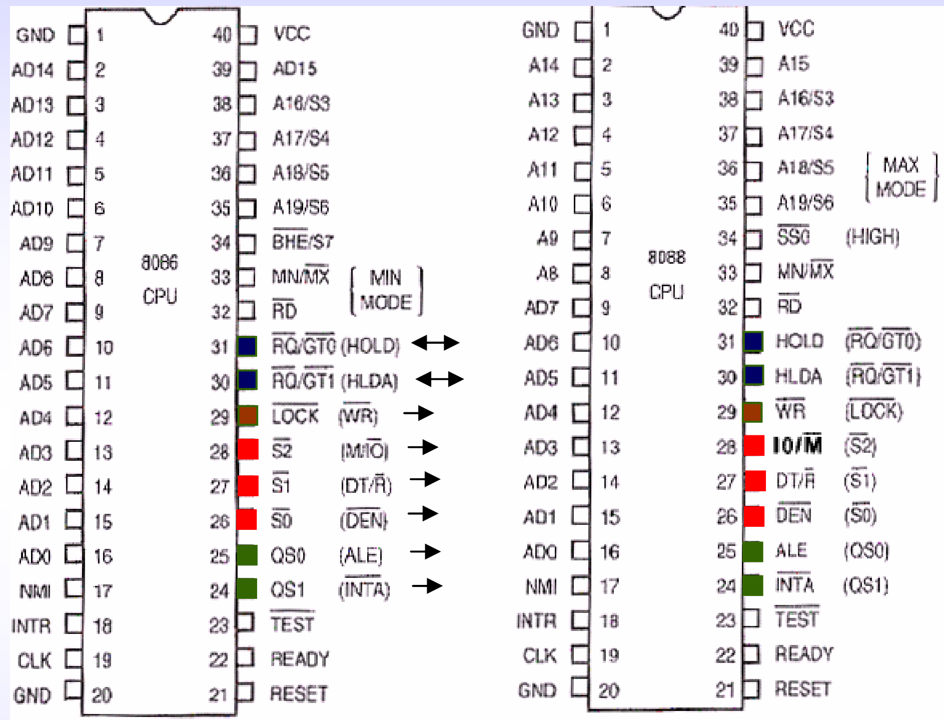


Table 9-6

S2	S1	S0	Function
0	0	0	Interrupt acknowledge
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

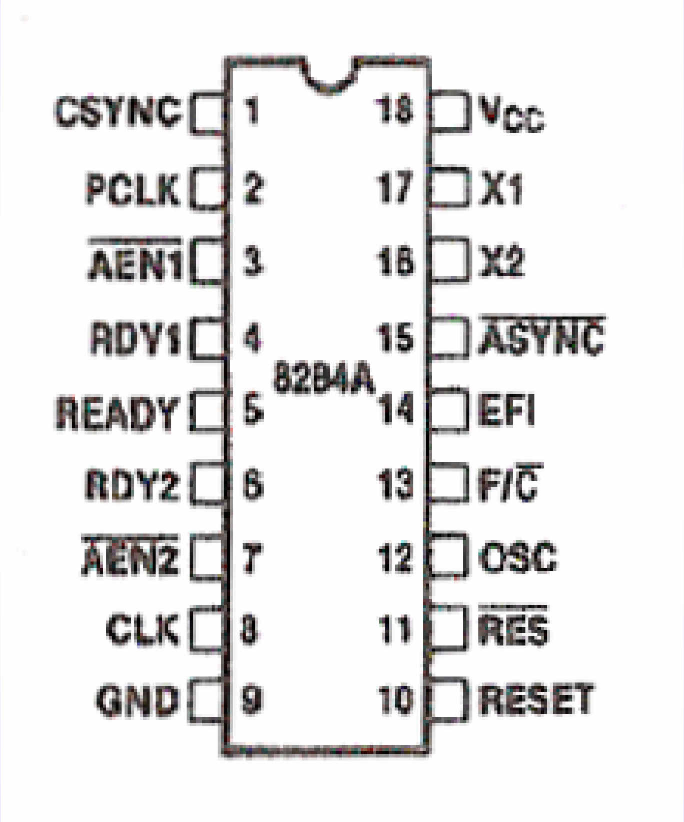
Table 9-7

QS1	QS0	Function
0	0	Queue is idle
0	1	First byte of opcode
1	0	Queue is empty
1	1	Subsequent byte of opcode

Clock Generator (8284A)

Provides the following functions:

- Clock and Sync:
  - Generates a CLK signal for the 8086/8088
  - Provides a CLK sync signal for use on multiprocessor 8086/8088 systems
  - Provides a TTL-level peripheral clock signal
- Provides RESET synchronization
- Provides READY synchronization for wait state generation

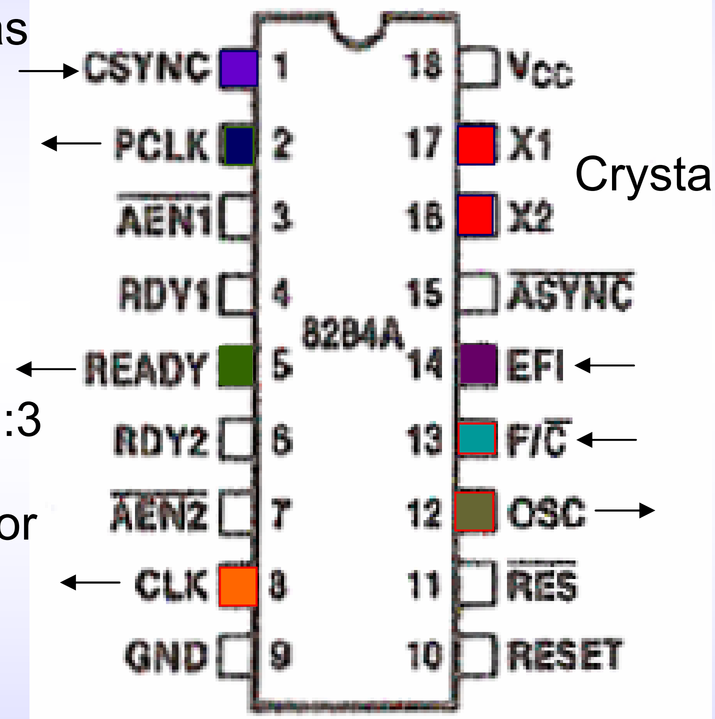


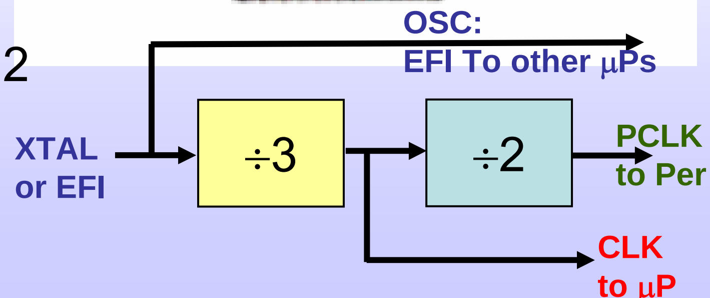
Clock Generator (8284A): Signals

Clocks & Clock Synchronization Signals

- X1 and X2: Crystal Oscillator pins. Connect a crystal of the correct frequency between these two terminals to generate the clock signal.
- EFI: External frequency input. Signal can be used as the clocking source to the 8284A instead of the crystal oscillator.
- F/#C input: Selects external EFI input (1) or the crystal oscillator (0) as the clocking source for the 8284A
- CLK output: The clock signal produced for connecting to the CLK input on the 8086/8088. At 1/3 rd of the crystal or EFI input frequency with 1:3 duty cycle: $f_{clock} = f_{xtal}/3 = f_{EFI}/3$
- OSC: Oscillator output. Same frequency as crystal or EFI. Connect to EFIs on 8284As of other μPs in multiprocessor systems (synchronized clocks)

  $f_{osc} = f_{xtal} = f_{EFI}$
- PCLK output: peripheral clock signal at 1/6 th of the crystal or EFI input frequency (1/2 clock freq) with 1:2 duty cycle. Use to drive peripheral equipment in the system $f_{pclk} = f_{xtal}/6 = f_{EFI}/6$
- CSYNC input: Clock synchronization input. Should be used if EFI is used, otherwise must be grounded.



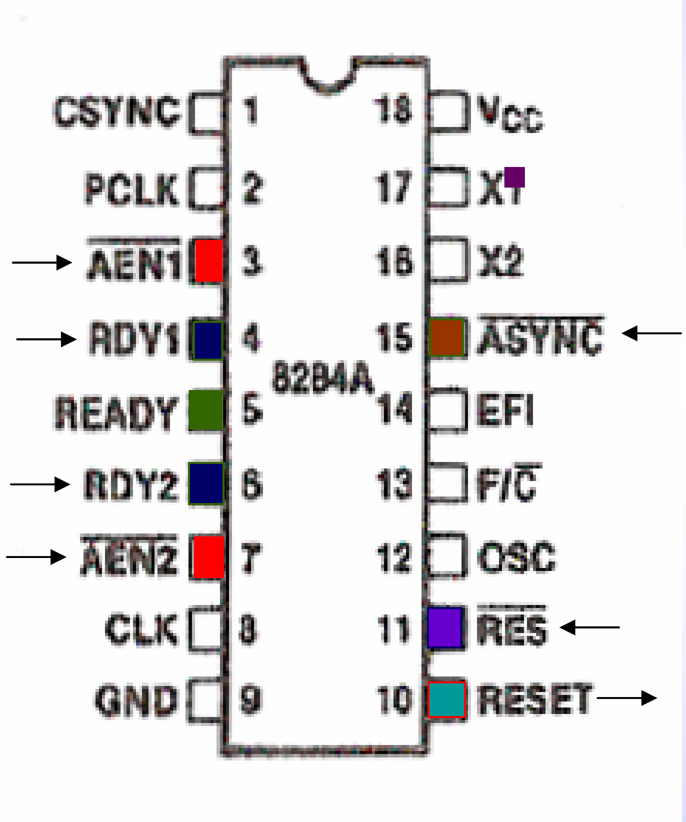


Clock Generator (8284A): Signals

RESET and ready Synchronization

RESET Signals

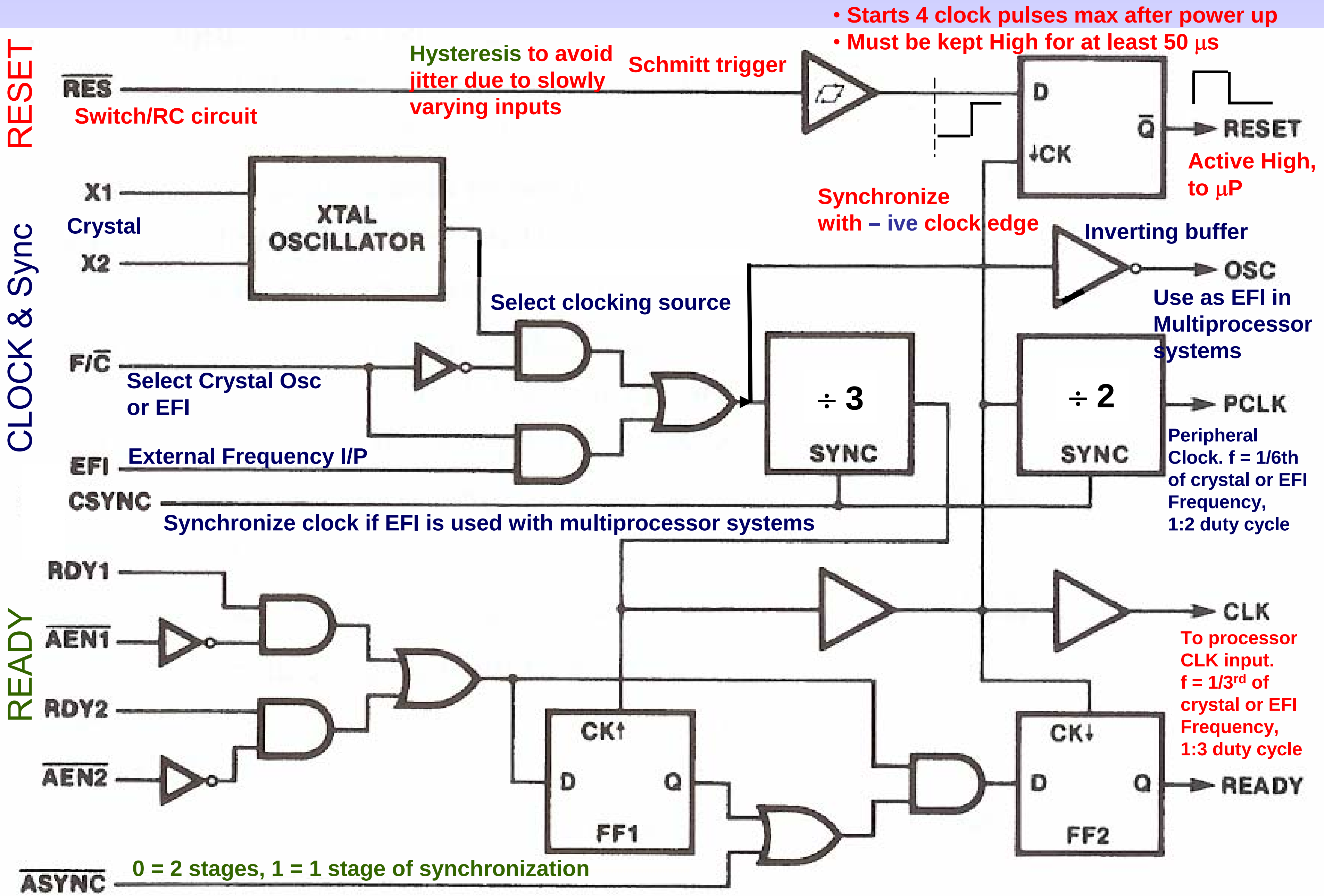
- #RES Reset input: Active low. Usually connected to an RC circuit to provide automatic reset at power on.
- RESET output: Synchronized to Clk. Connect to the 8086/8088 RESET input.



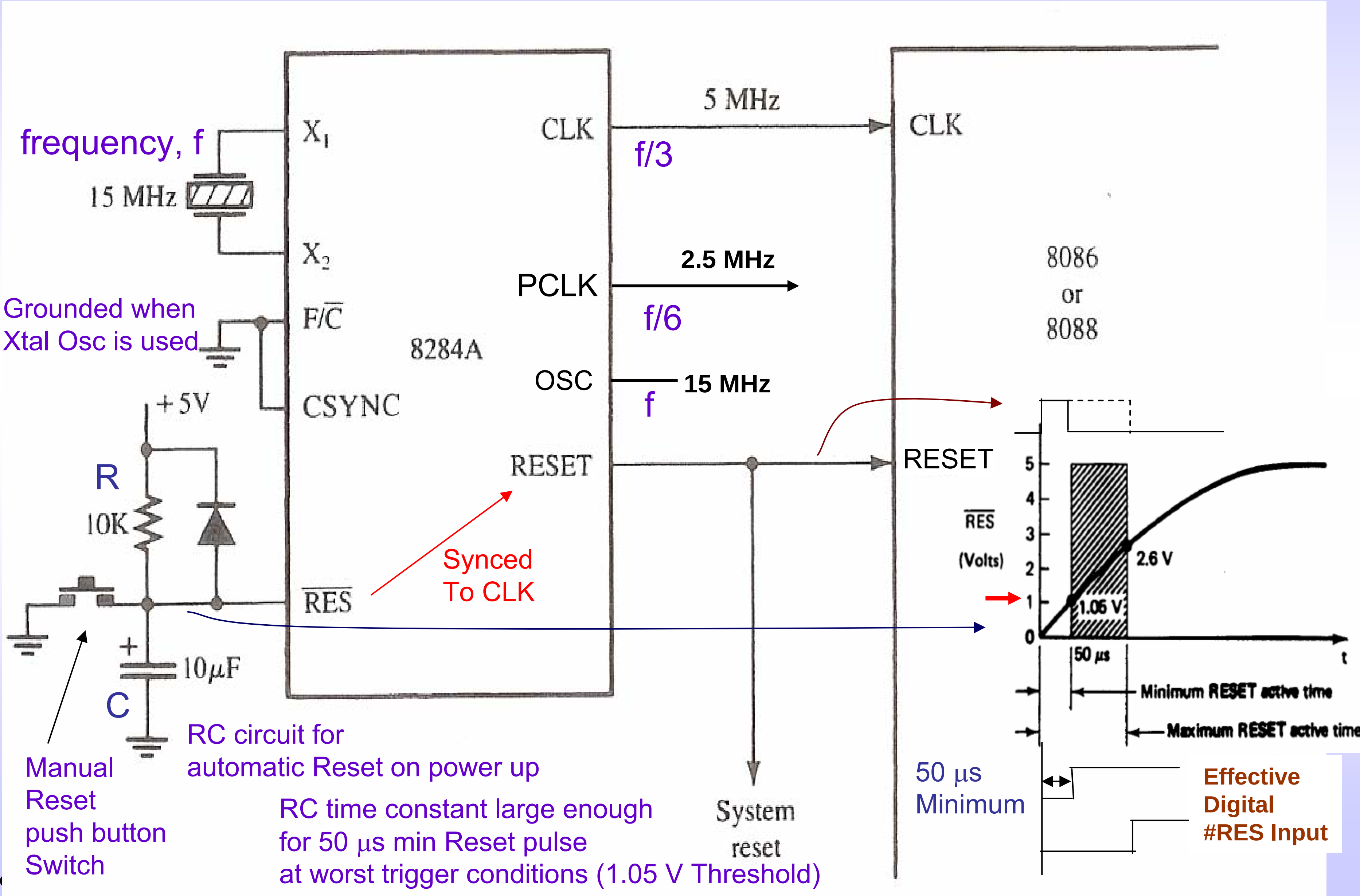
READY Signals

- #AEN1 and #AEN2 address enable inputs: Used with RDY1 and RDY2 inputs to generate the READY output. The READY output is connected to the READY input on the 8086/8088 µP to control memory wait states.
- #ASYNC input: for READY output synchronization. Selects 1 or 2 stages of synchronization for the RDY1 and RDY2 inputs.

Clock Generator (8284A): Block Diagram



Typical Application of the 8284A for clock and Reset signal generation



Br

Bus Demultiplexing and Buffering

- Demultiplexing:

  The address/data and address/status buses have been multiplexed to reduce the device pin count. These buses must be demultiplexed (separated) to obtain the signals required for interfacing other circuits to the μP

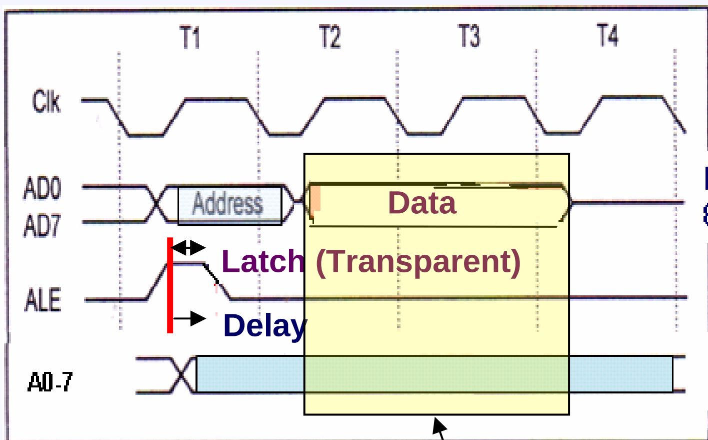
  - Use the ALE output from the microprocessor to latch the address/status information that appear briefly on the multiplexed bus
  - This makes the latched address information available for long enough time for correct interfacing, e.g. to memory

- Buffering:

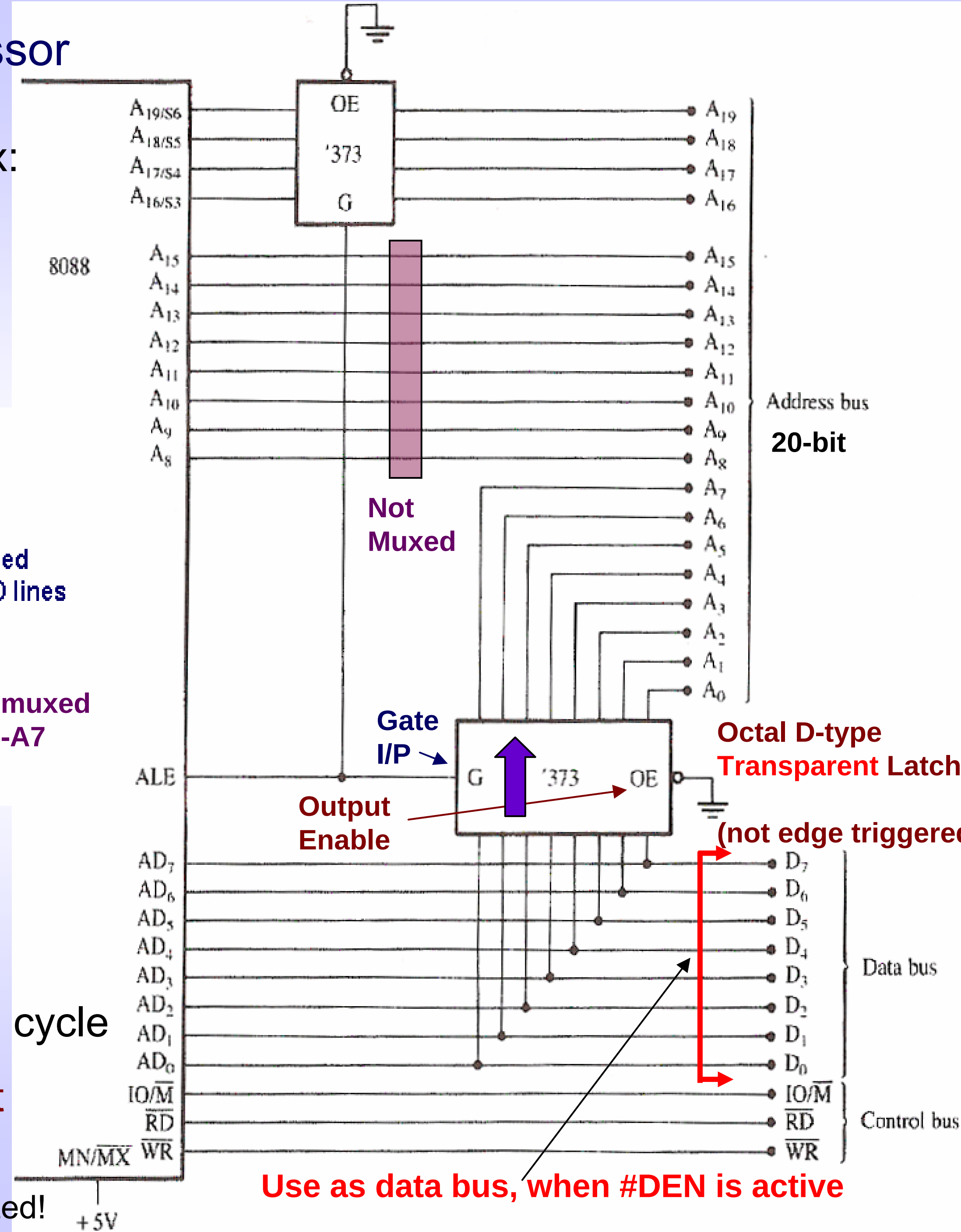
  Fan out of output pins is limited, so output signals should be buffered in large systems

Demultiplexing the 8088 Processor

Using the ALE signal to Demultiplex:
-The Address lines A0-7
from the AD0-7 muxed bus
-The A16-19 lines from the
A16/S3-A19/S6 muxed bus



Memory write cycle for the 8088
(non-muxed lines are not shown)



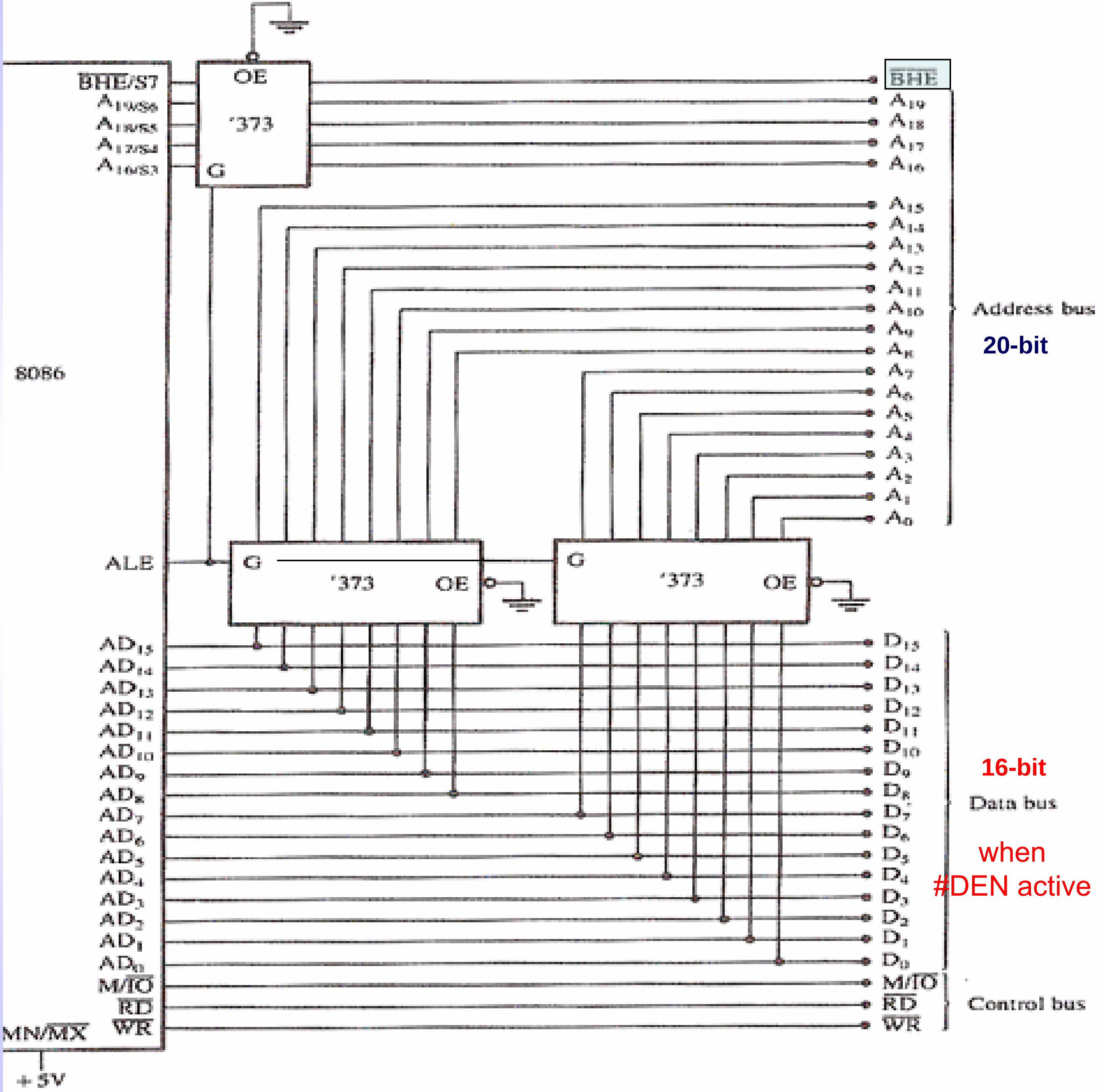
Data and address lines must remain
valid and stable for the duration of the cycle

74373 is an Octal D-type transparent
Latch with 3-state outputs

Feature **Not** utilized!

Brey: The Intel Microprocessors, 7e

Demultiplexing the 8086 Processor Address/Data bus



Brey: The Intel Microprocessors, 7e

DM74LS373 /LS374

3-STATE Octal D-Type Transparent /Edge-triggered Latches

General Description

These 8-bit registers feature totem-pole 3-STATE outputs designed specifically for driving highly-capacitive or relatively low-impedance loads. The high-impedance state and increased high-logic level drive provide these registers with the capability of being connected directly to and driving the bus lines in a bus-organized system without need for interface or pull-up components. They are particularly attractive for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

The eight latches of the DM74LS373 are transparent D-type latches meaning that while the enable (G) is HIGH the Q outputs will follow the data (D) inputs. When the enable is taken LOW the output will be latched at the level of the data that was set up.

The eight flip-flops of the DM74LS374 are edge-triggered D-type flip flops. On the positive transition of the clock, the Q outputs will be set to the logic states that were set up at the D inputs.

A buffered output control input can be used to place the eight outputs in either a normal logic state (HIGH or LOW logic levels) or a high-impedance state. In the high-impedance state the outputs neither load nor drive the bus lines significantly.

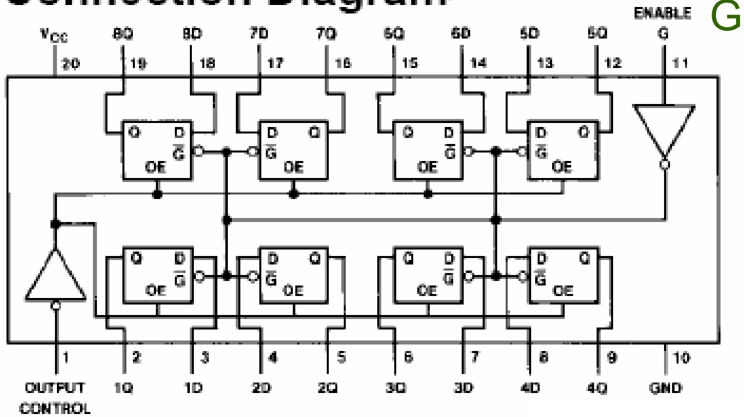
The output control does not affect the internal operation of the latches or flip-flops. That is, the old data can be retained or new data can be entered even while the outputs are OFF.

Features

Transparent (373) Edge-triggered (374)

- Choice of 8 latches or 8 D-type flip-flops in a single package
- 3-STATE bus-driving outputs
- Full parallel-access for loading
- Buffered control inputs
- P-N-P inputs reduce D-C loading on data lines

Connection Diagram



Function Table

LS373

Output Control	Enable G	D	Output
L	H	H	H
L	H	L	L
L	L	X	Q_0
H	X	X	Z

D → O/P Transparency

Last O/P Maintained (latched)

H = HIGH Level (Steady State) L = LOW Level (Steady State)

X = Don't Care Z = High Impedance State

Q_0 = The level of the output before steady-state input conditions were established.

Bre

Buffering

Since the microprocessor output pins provide minimum drive current at the 0 logic level, buffering is often needed if more TTL loads are connected to any bus signal: Consider 3 types of signals

- For demuxed signals: Latches used for demuxing, e.g. ‘373, can also provide the buffering for the demuxed lines:
  - 0-level output can sink up to 32 mA (1 load 1.6 mA loads)
  - 1-Level output can source up to 5.2 mA (1 load = 40 μA)

So, Fan out = ? Which case sets the limit?

- For non-demuxed unidirectional (always output) address and control signals (e.g. A8-15 on the 8088), buffering is required- often using the 74ALS244 (unidirectional) buffer.
- For non-demuxed bidirectional data signals (pin used for both in and out), buffering is often accomplished with the 74ALS245 bidirectional bus buffer

Caution: Buffering introduces a small delay in the buffered signals. This is acceptable unless memory or I/O devices operate close to the maximum bus speed

Fully DeMuxed and buffered 8088

Non-DeMuxed Address Lines (unidirectional- Always O/Ps)
→ 244 Buffer

74244 is an Octal Buffer with 3-state outputs

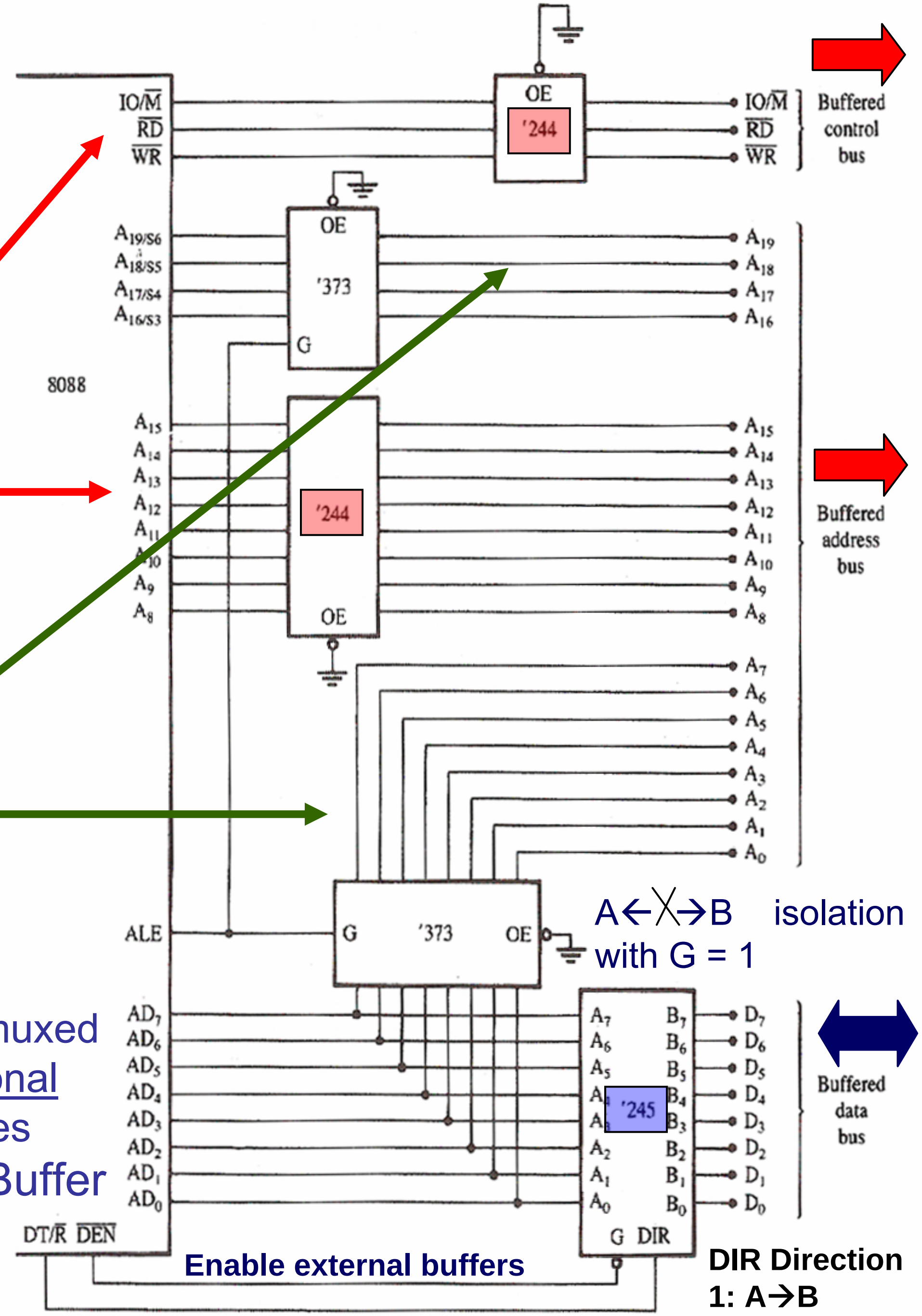
Feature Not utilized!

DeMuxed Address Lines (unidirectional- Always O/Ps)
→Latch provides the buffering

Non-Demuxed Bidirectional Data Lines
→ 245 Buffer

74245 is an Octal Bus Transceiver with 3-state outputs

Feature utilized!



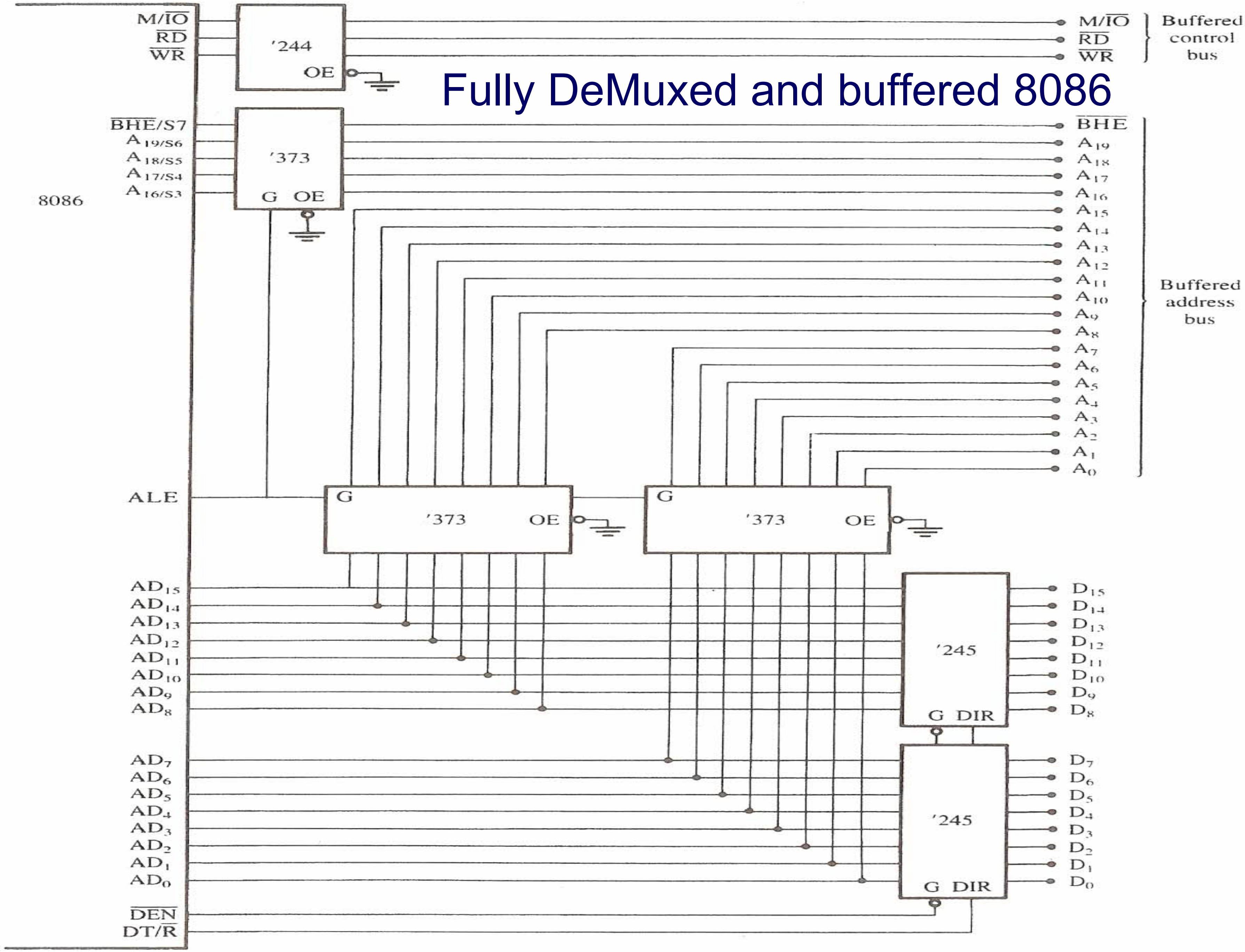
A←╳→B isolation with G = 1

Enable external buffers

DIR Direction
1: A→B
0: A←B

Brey: The Intel Microprocessors, 7e

Fully DeMuxed and buffered 8086



DM74LS244

Octal 3-STATE Buffer/Line Driver/Line Receiver

(Not Transceivers)

General Description

These buffers/line drivers are designed to improve both the performance and PC board density of 3-STATE buffers/drivers employed as memory-address drivers, clock drivers, and bus-oriented transmitters/receivers. Featuring 400 mV of hysteresis at each low current PNP data line input, they provide improved noise rejection and high fanout outputs and can be used to drive terminated lines down to 133Ω.

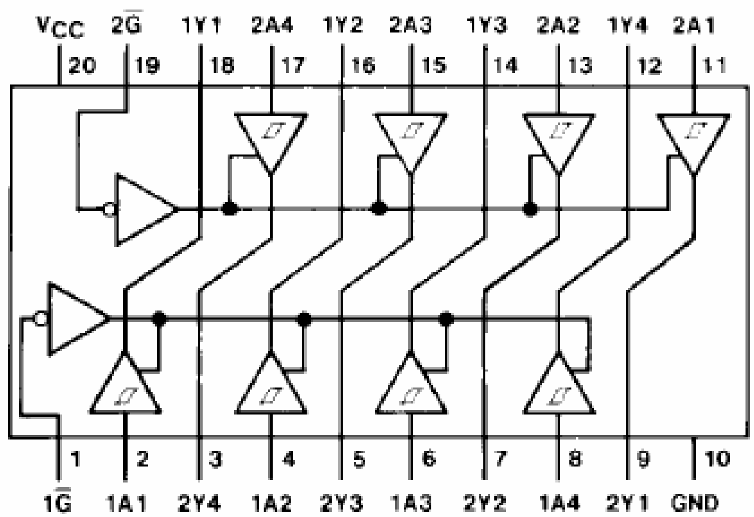
- **Unidirectional**
- **Just a Buffer- No latching**
- **Non-inverting**

Features

- 3-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at data inputs improves noise margins
- Typical I_{OL} (sink current) 24 mA
- Typical I_{OH} (source current) –15 mA
- Typical propagation delay times
  - Inverting 10.5 ns
  - Noninverting 12 ns
- Typical enable/disable time 18 ns
- Typical power dissipation (enabled)
  - Inverting 130 mW
  - Noninverting 135 mW

Use to determine Fan-out

Connection Diagram



Function Table

Inputs		Output
$\overline{G}$	A	Y
L	L	L
L	H	H
H	X	Z

L = LOW Logic Level
H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
Z = High Impedance

Enabled- Normal Operation

Disabled- HiZ O/P

54LS245/DM54LS245/DM74LS245
TRI-STATE® Octal Bus Transceiver

General Description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control function implementation minimizes external timing requirements.

The device allows data transmission from the A bus to the B bus or from the B bus to the A bus depending upon the logic level at the direction control (DIR) input. The enable input ($\overline{G}$) can be used to disable the device so that the buses are effectively isolated.

Features

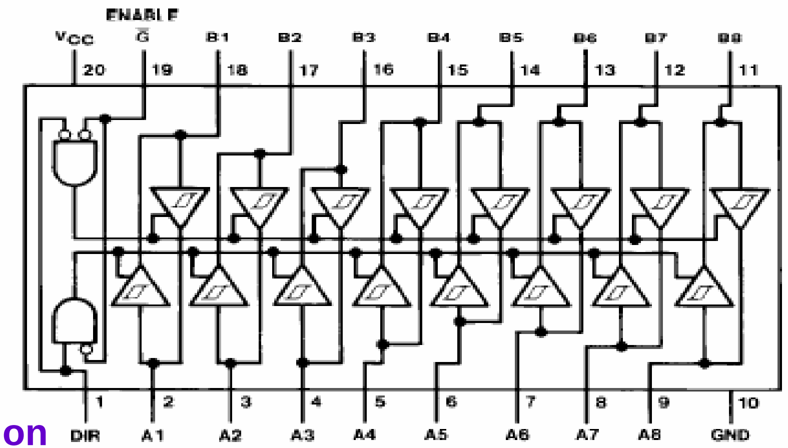
- Bi-Directional bus transceiver in a high-density 20-pin package
- TRI-STATE outputs drive bus lines directly
- PNP inputs reduce DC loading on bus lines
- Hysteresis at bus inputs improve noise margins
- Typical propagation delay times, port-to-port 8 ns
- Typical enable/disable times 17 ns
- I_{OL} (sink current)
  - 54LS 12 mA
  - 74LS 24 mA
- I_{OH} (source current)
  - 54LS −12 mA
  - 74LS −15 mA
- Alternate Military/Aerospace device (54LS245) is available. Contact a National Semiconductor Sales Office/Distributor for specifications.

Connection Diagram

- **Bidirectional**
- **Just a Buffer- No latching**
- **Non-inverting**

#Enable

Dual-In-Line Package



Direction

TL/F/6413–1

Order Number 54LS245DMQB, 54LS245FMQB, 54LS245LMQB, DM54LS245J, DM54LS245W, DM74LS245WM or DM74LS245N
See NS Package Number E20A, J20A, M20B, N20A or W20A

Function Table

Enable $\overline{G}$	Direction Control DIR	Operation
L	L	B data to A bus
L	H	A data to B bus
H	X	Isolation

H = High Level, L = Low Level, X = Irrelevant

Enabled- Normal Operation

Disabled- HiZ O/P

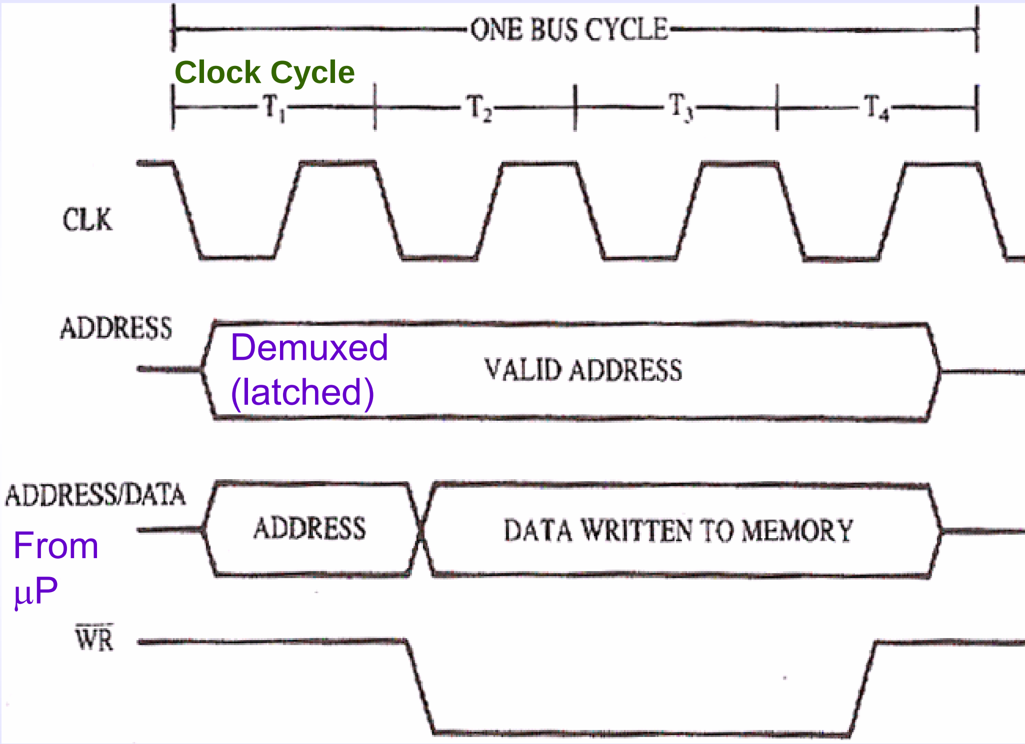
A-B: Open circuit, No connection

Bus Timing

Timing in General

- A data transfer operation to/from the μP requires at least one bus cycle
- Each bus cycle consists of 4 clock cycles, T1, T2, T3, T4, each of period T
- With a 5 MHz processor clock:

  - T = 1/5 MHz = 0.2 μs
  - Bus cycle = 4 T = 0.8 μs
  - Max rate for memory and I/O transfers = 1/0.8 = 1.25 M fetches per sec (Fetch speed).
  - Processor executes 2.5 Million Instructions per sec (MIPS) (Execute speed)



→ Fetch is slower than execute. Effect on pipelining?

Bus Timing in General, Contd.

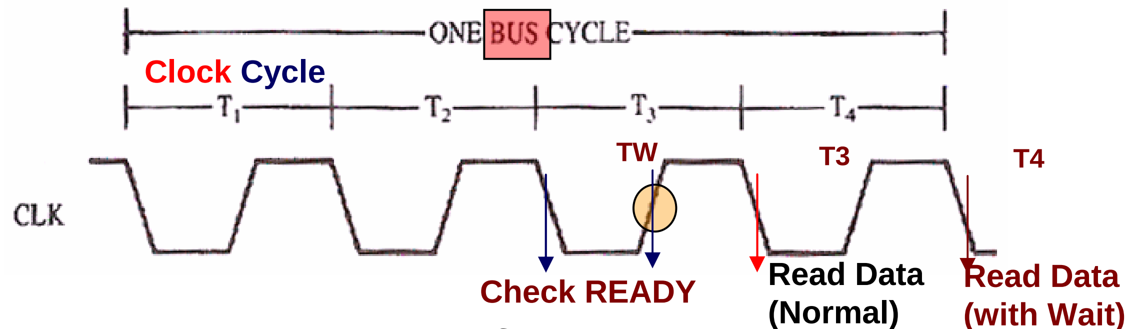
• T1:

- Address is emitted from the Processor
- Control signals such as ALE, DT/#R, IO/#M, etc. are also initiated

Note: #RD,#WR,#INTA are all inactive high during T1

• T2

- Used primarily for changing the direction of the AD bus during read operations (→ then ←)
- Read or write controls are setup, e.g. #DEN, #RD (#INTA) or #WR



* If a Write operation, Data to be written is put on the bus for the external device to take
* If a Read operation, AD bus is floated, so external device can put the read data on it

• T3 & T4:

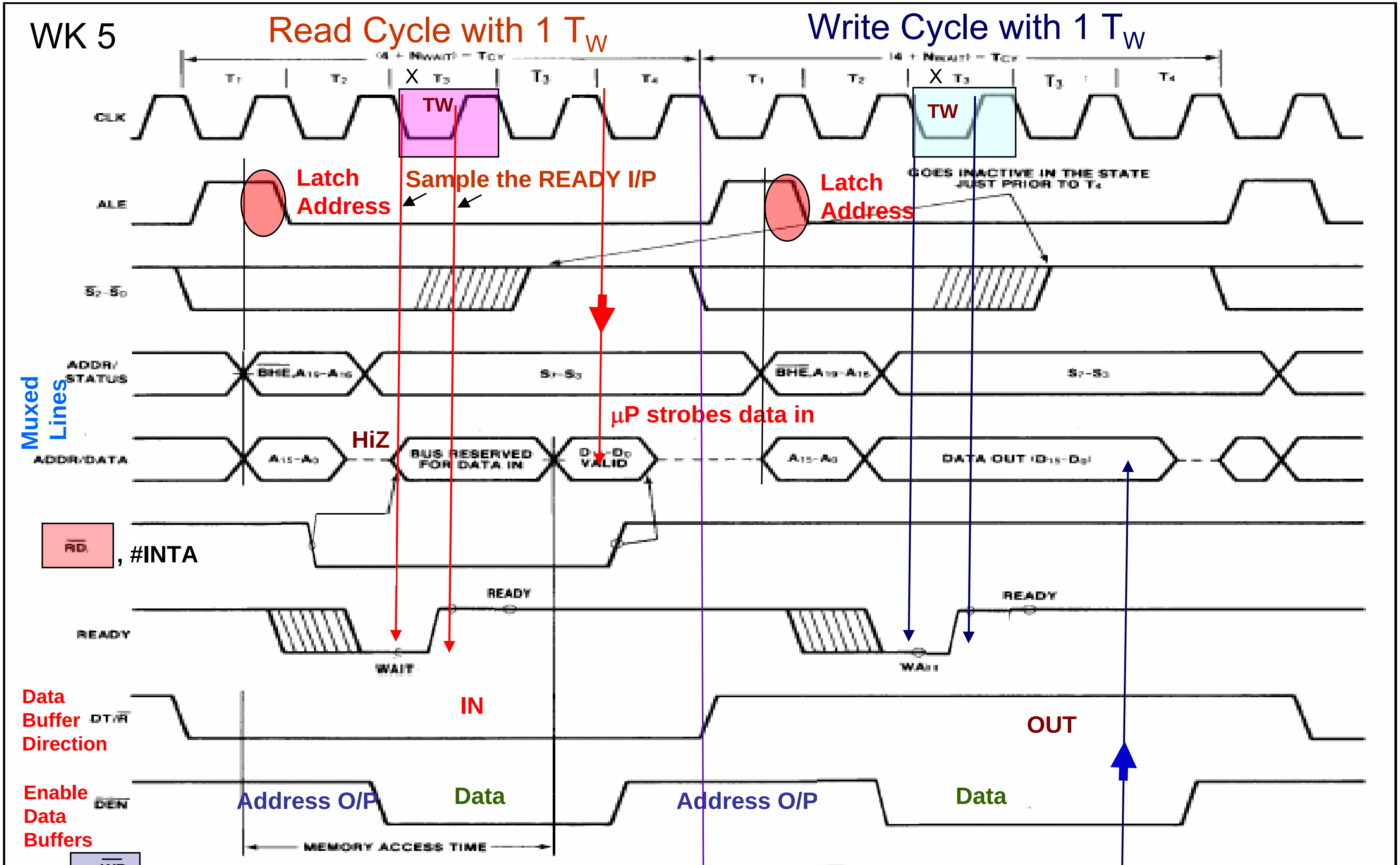
Actual Data transfer occurs during T3 & T4.

- In Read operations, “Data In” on the data bus is normally strobed into the processor at the start of T4
- In Write operations, “Data out” on the bus is strobed into the external device at the trailing edge of the #WR signal

• Wait States: If addressed device is too slow to allow normal data transfer scheme, it sets the READY input low (i.e. indicates NOT READY)

* The processor samples the READY input at the end of T2. If found low, T3 is considered a “Wait" state (TW). Ready is checked again at the middle of that wait state. If high, it is followed by proper T3 and T4. If low (not ready), the next clock cycle is considered an additional wait state, and so on

Timing in General: Read & Write with waits



Note: #RD,#WR,#INTA are all inactive high during T1

Bus Timing, Contd.

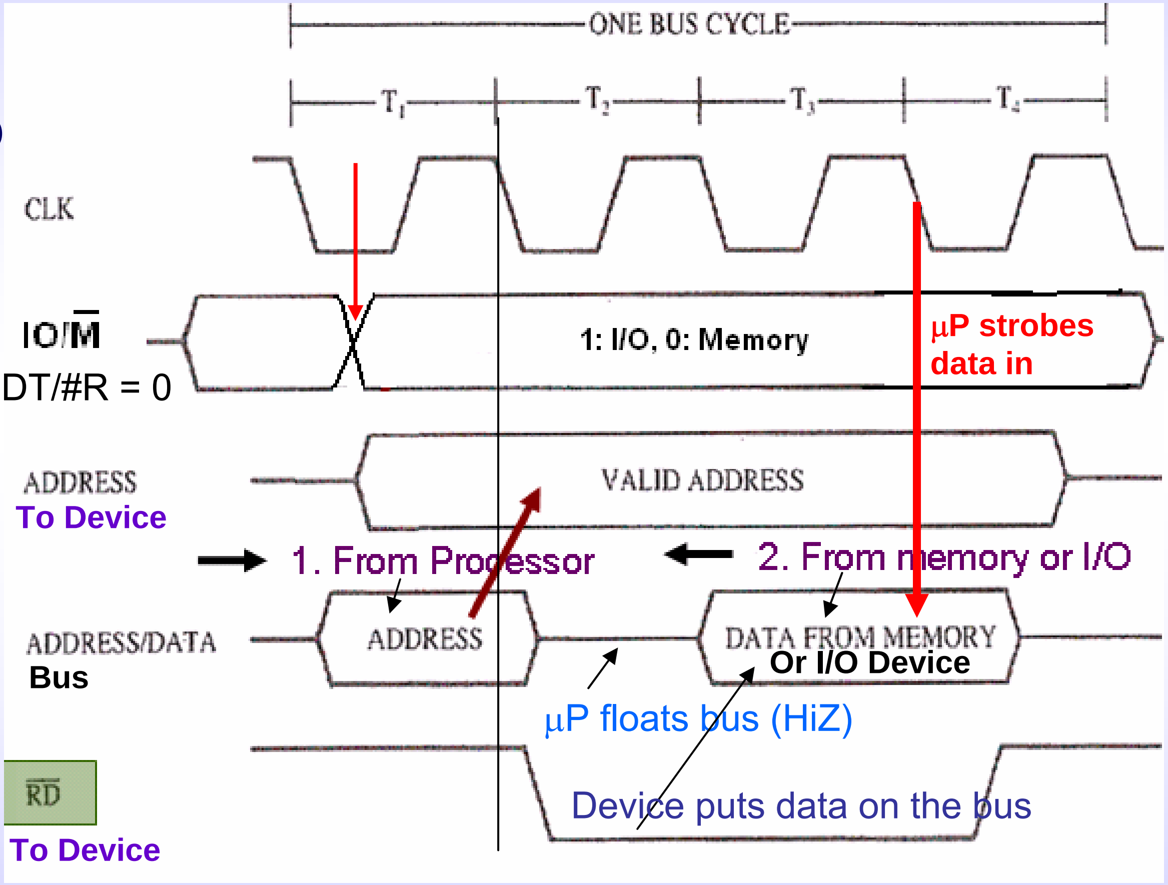
Basic Data Read Timing

Timing for a basic Read cycle

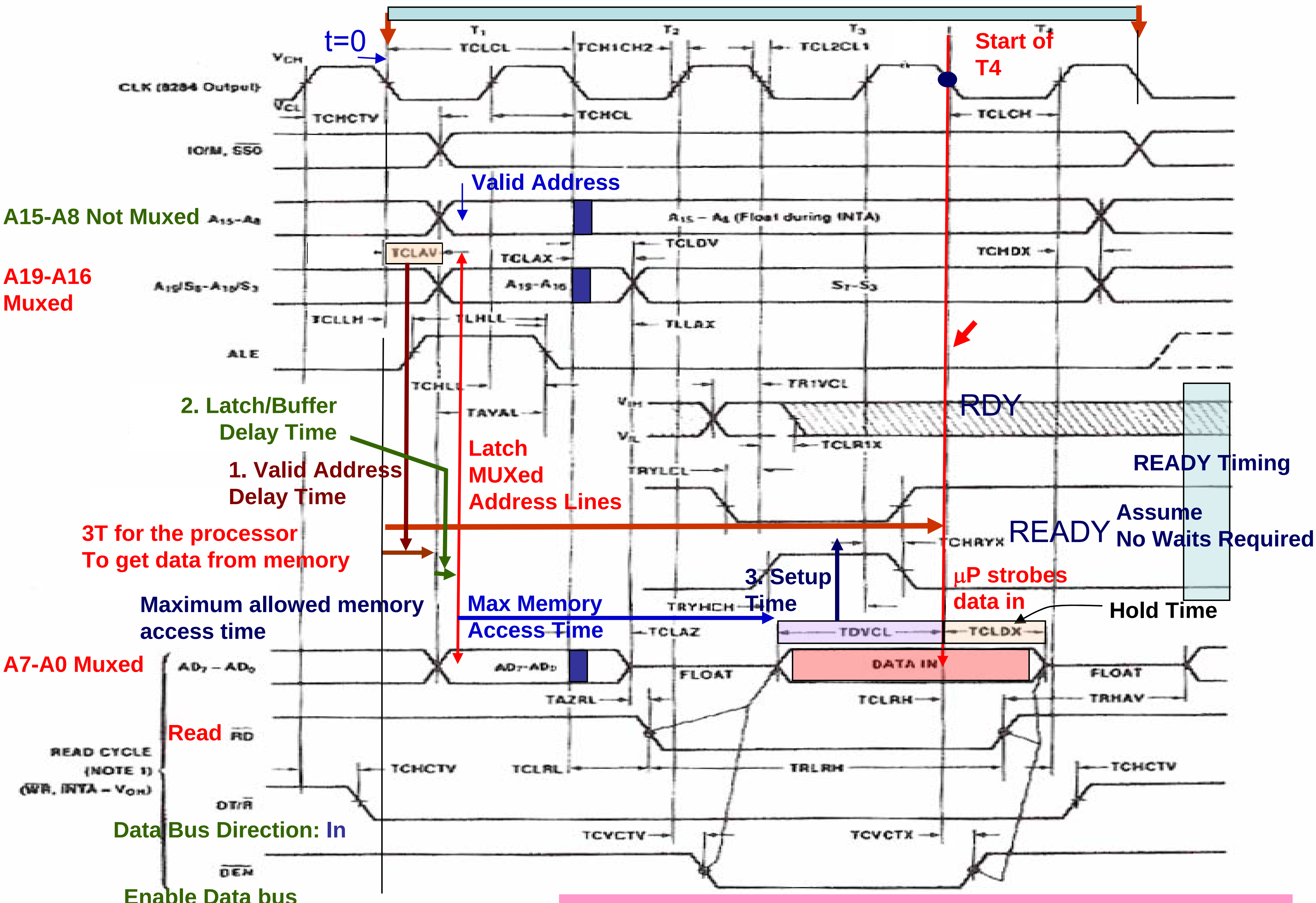
- Can be for either memory or I/O

On the 8088:
- For memory, IO/#M = 0
- For I/O: IO/#M = 1

Also DT/#R = 0



Detailed Memory READ Timing for the 8088



See Fig. 9-12 for Detailed Timing Specifications

Standard READ Timing Budget: 8088 @ 5MHz

- It takes the processor 3 clock cycles (3T = 3 x 200 = 600 ns with a 5 MHz clock) to take-in the memory data
- Not all this time is available for the memory device to retrieve the data and put it on the bus, there is: (See Fig. 9-12)

Incurred Delays

1. Address Valid Delay, TCLAV = 110 ns max
2. Delay in address latch/buffer and decoders ≈ 40 ns

Required

3. Data-in Setup time (required min), TDVCL = 30 ns

- Maximum allowed memory access time to operate without waits

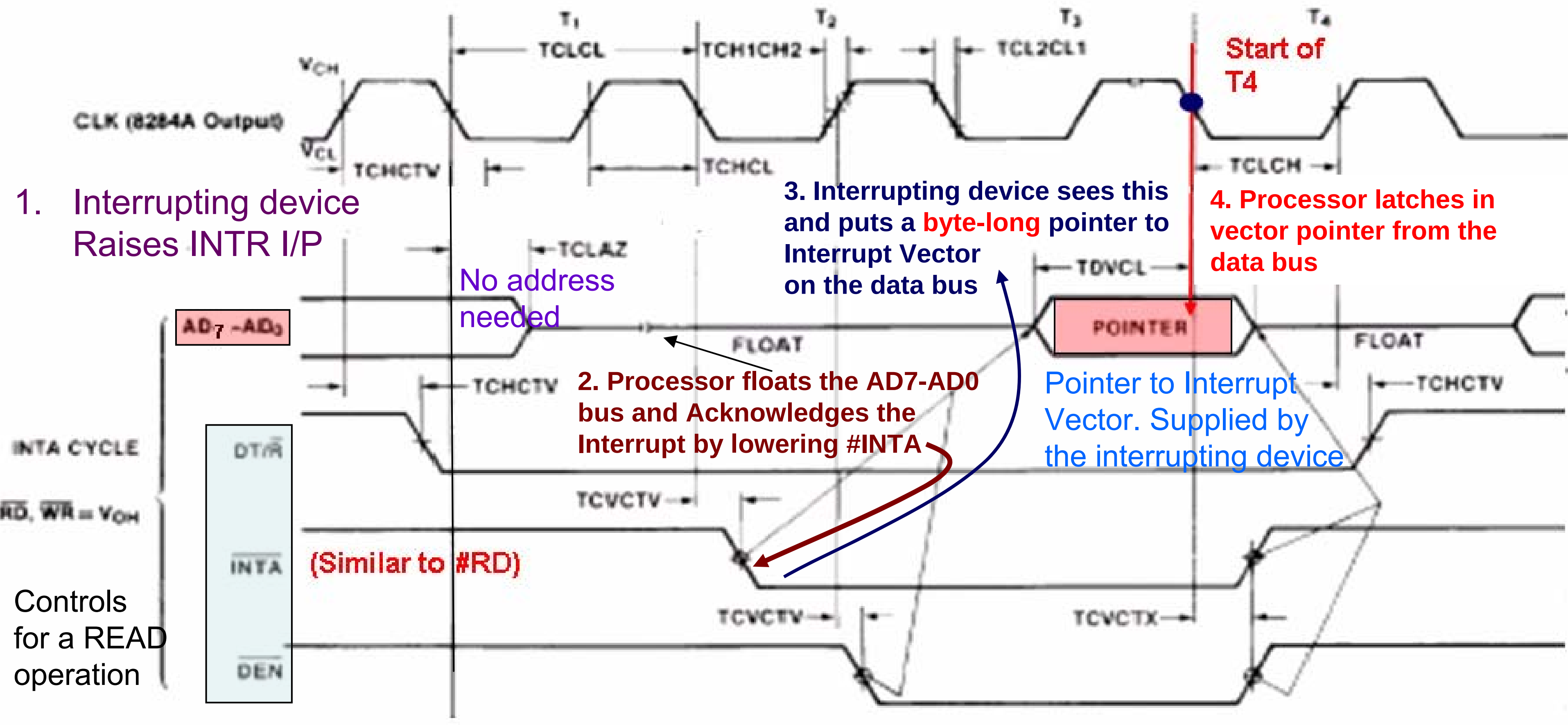
  = 3 x 200 – (110+40+30) = 420 ns
- If memory has a longer access time, it needs to request wait states from the processor using the READY input
- #RD signal should be wide enough, TRLRH = 325 ns,

  as there may be hold time requirements for the Data-In.

  The #RD signal is extended with the insertion of wait states

INTA Read Timing, 8088

Similar to a memory or I/O read cycle, with #INTA replacing #RD



Upon accepting a hardware interrupt request from a device (on INTR I/P), the processor acknowledges this to the device and initiates an #INTA read cycle for the 1-byte interrupt number which the processor reads and uses as a pointer to the interrupt service routine to be executed

See Fig. 9-12 for Detailed Timing Specifications

Brey: The Intel Microprocessors, 7e

Bus Timing

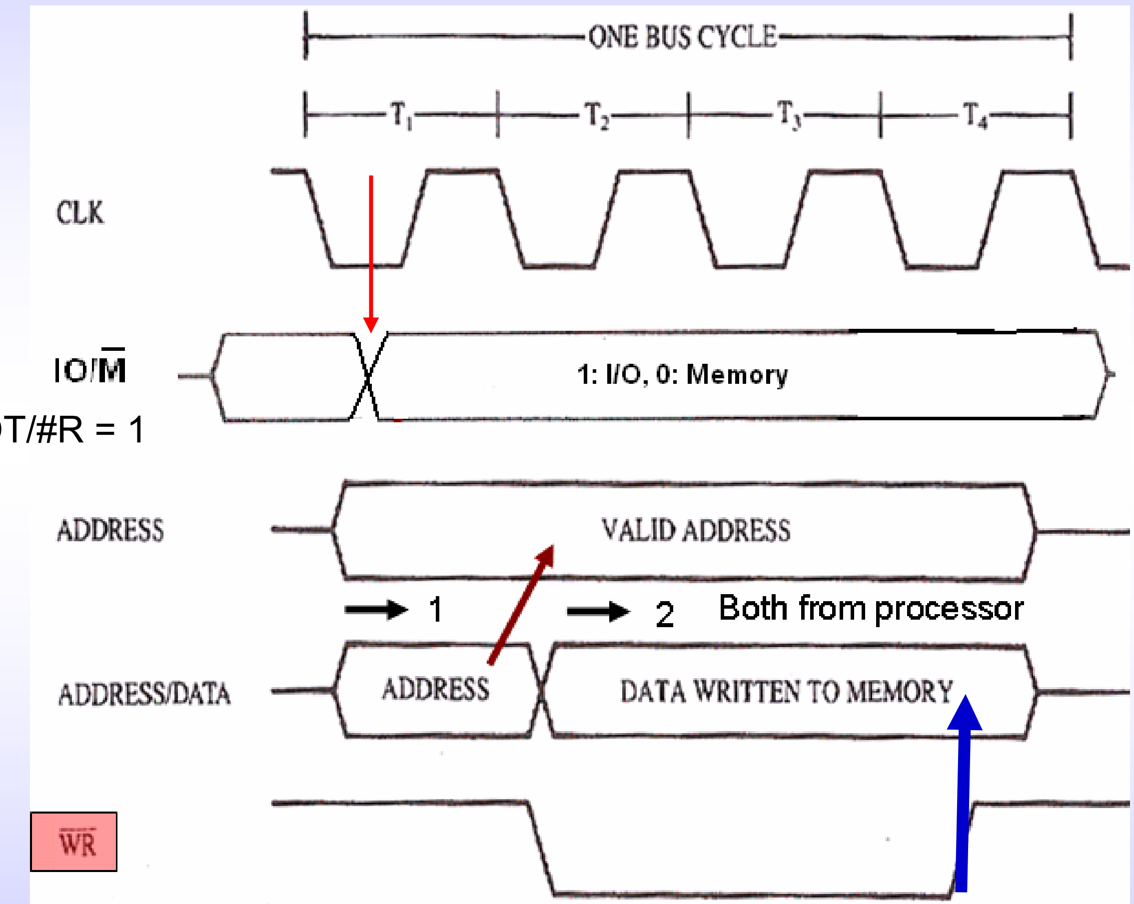
Basic Data Write Timing

Timing for a basic write cycle

- Can be for either memory or I/O

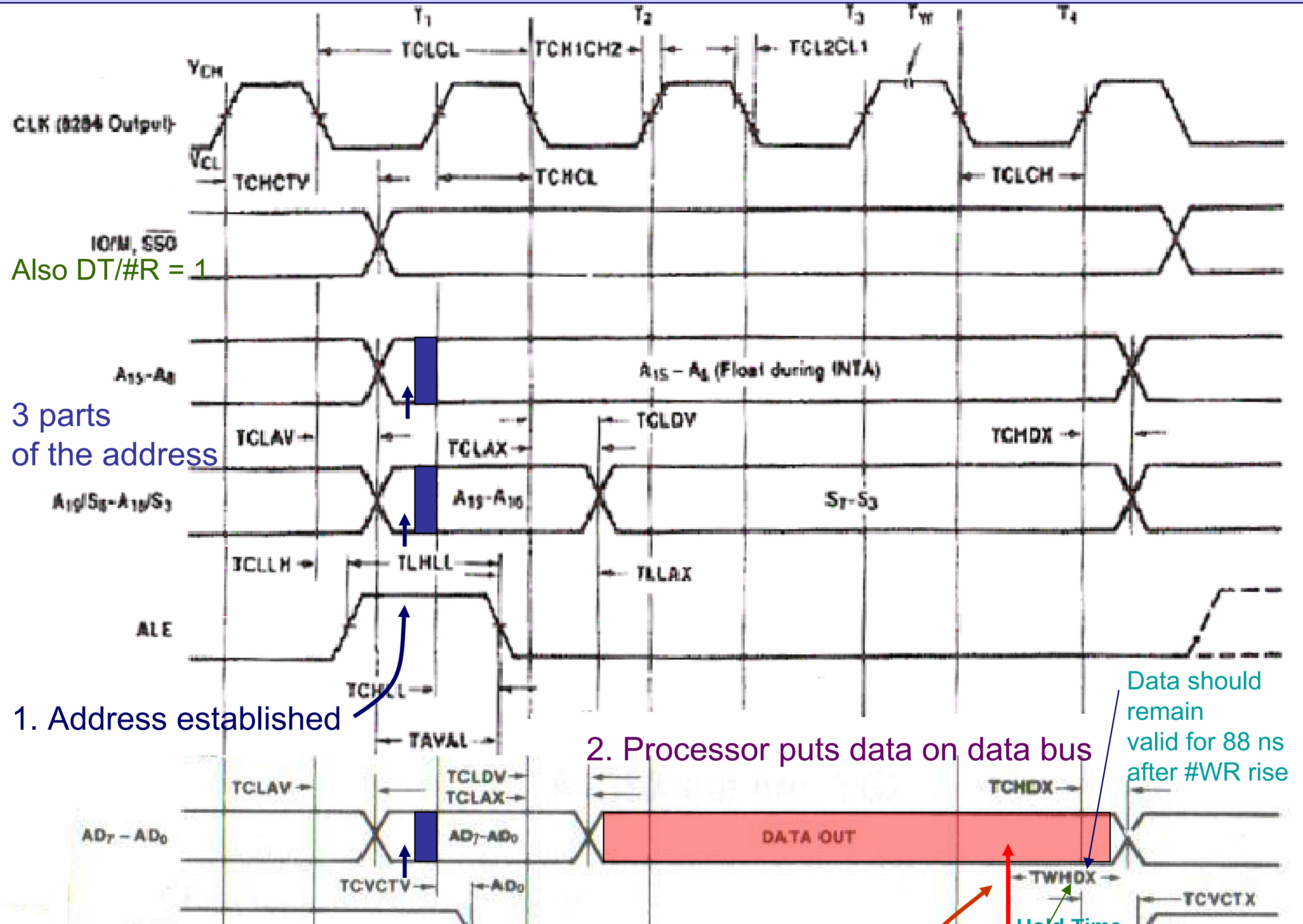
On the 8088:
- For memory, IO/#M = 0
- For I/O: IO/#M = 1

Also DT/#R = 1



Device strobes data in

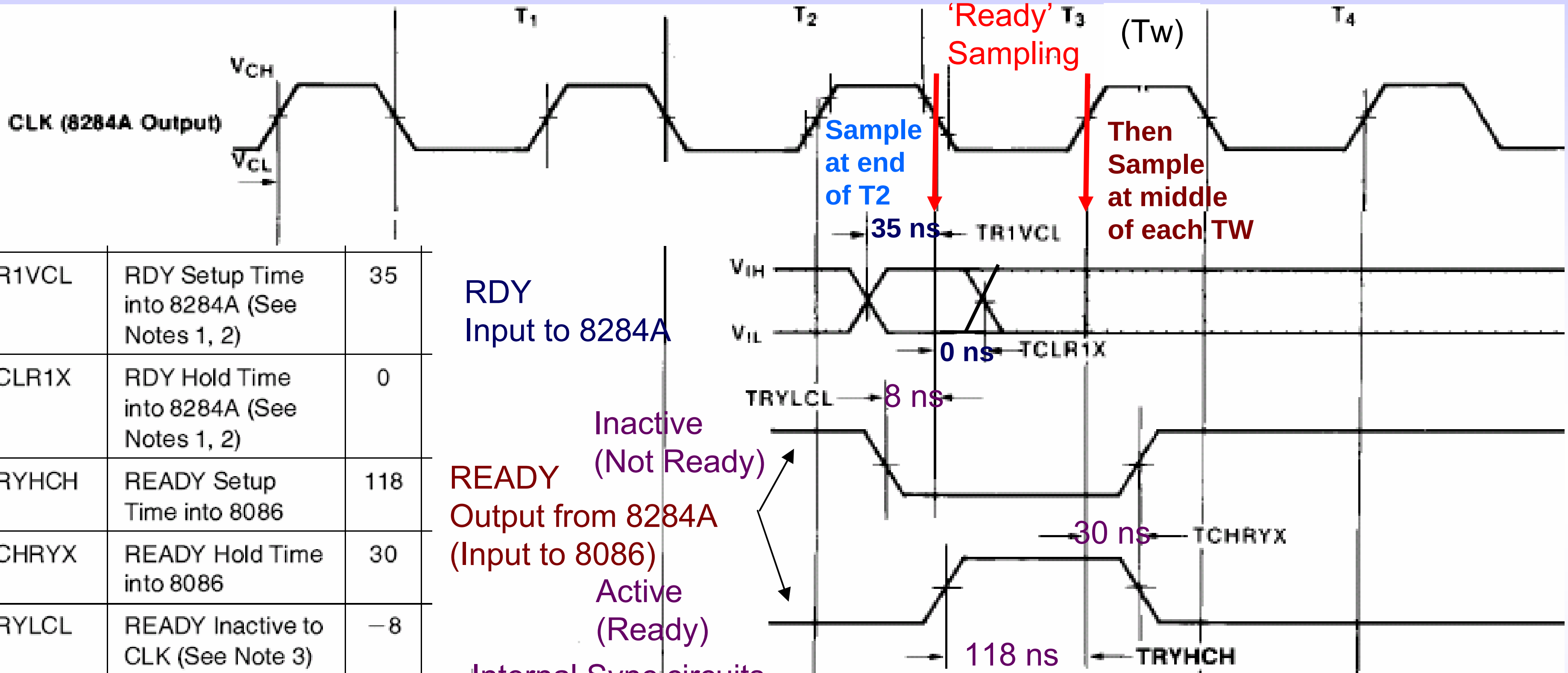
Detailed Write Timing for the 8088



READY and Wait States: 5 MHz Clock

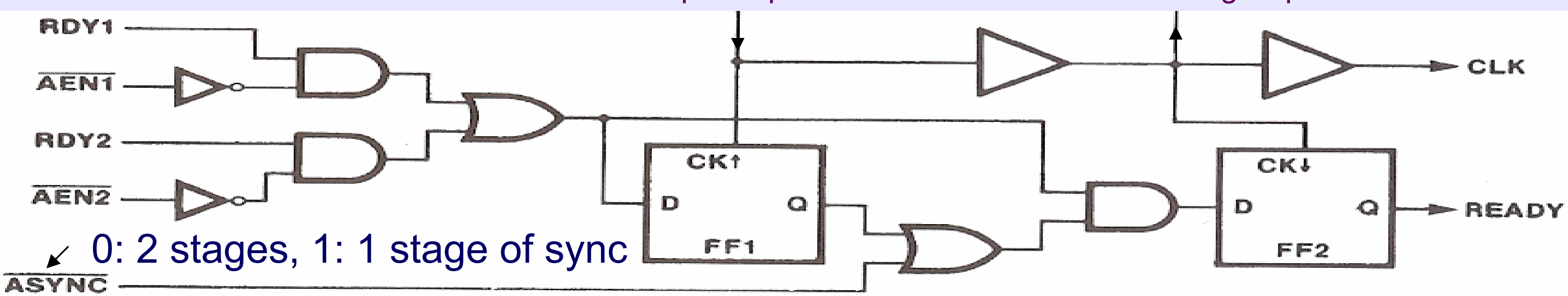
- If the memory or I/O device is too slow to use the standard 4T read cycle (i.e. if access time > 420 ns) , wait states must be inserted into the cycle by using the READY input to the processor
- Wait states are additional clock cycles that increase the length of access time allowed for memory or I/O devices
- Wait states are inserted as multiple clock cycles (1, 2, 3, etc.) between the standard T2 and T3 cycles
- Wait time is inserted as whole clock cycles: i.e. if access time = 430 ns → insert 1 complete wait state of 200 ns!
- Inserting n wait states increase the maximum allowed access time from the normal typical value of 420 ns (with no waits) to 420 + n 200 ns ,
- i.e. Memory $t_{access} \leq 420 + n\ 200$

Detailed Ready Timing

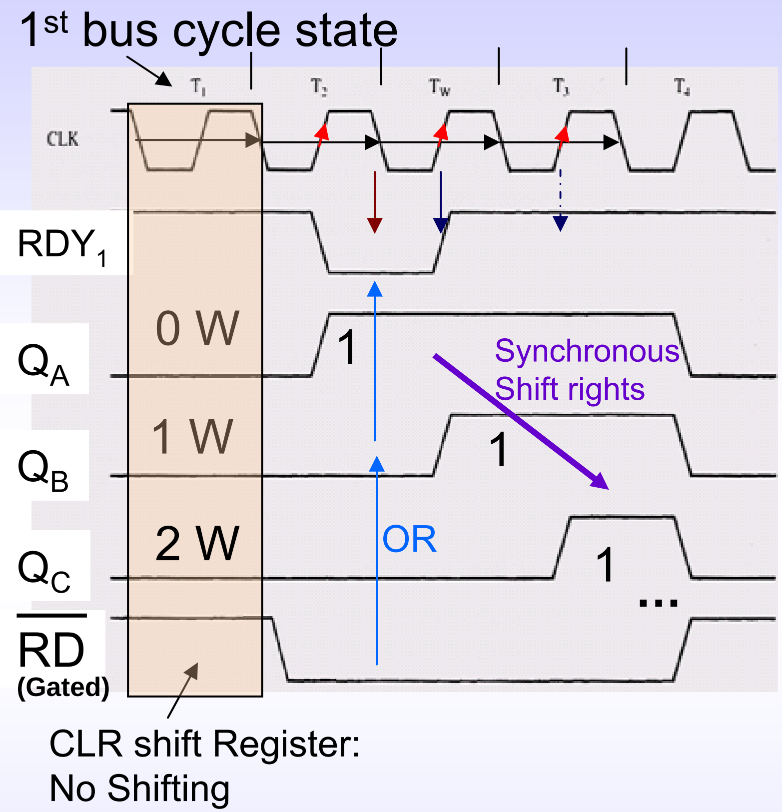


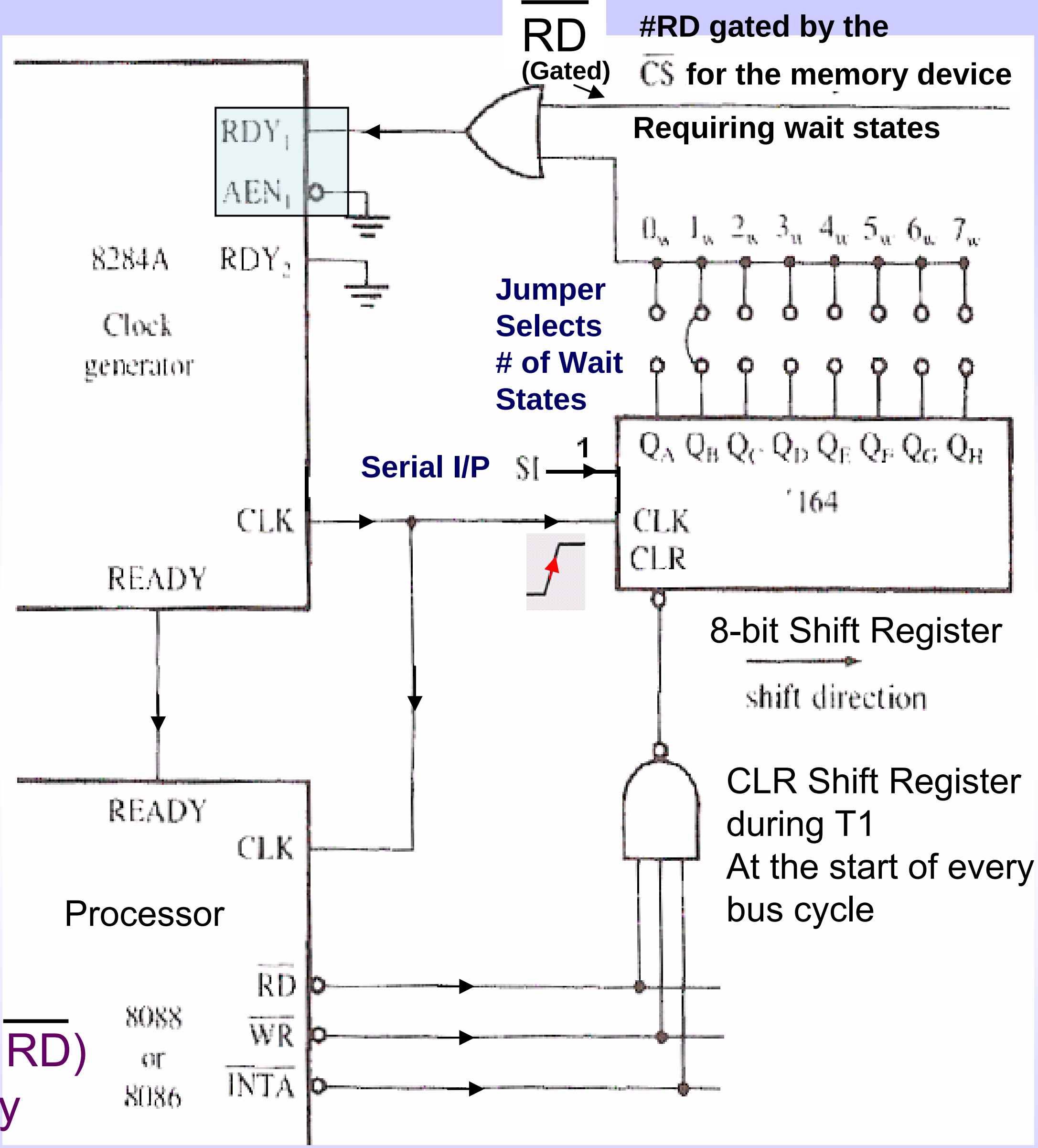
TR1VCL	RDY Setup Time into 8284A (See Notes 1, 2)	35
TCLR1X	RDY Hold Time into 8284A (See Notes 1, 2)	0
TRYHCH	READY Setup Time into 8086	118
TCHRYX	READY Hold Time into 8086	30
TRYLCL	READY Inactive to CLK (See Note 3)	−8

Internal Sync circuits
In the 8284A ensure that
READY output to processor meets the above timing requirements



Generation of 0-7 wait states Using one of the two RDY inputs to the 8284A





Effectively,
RDY1 = (Selected Q + $\overline{\text{RD}}$)
(when the slow memory device is accessed)

Note: #RD is extended with the addition of wait states

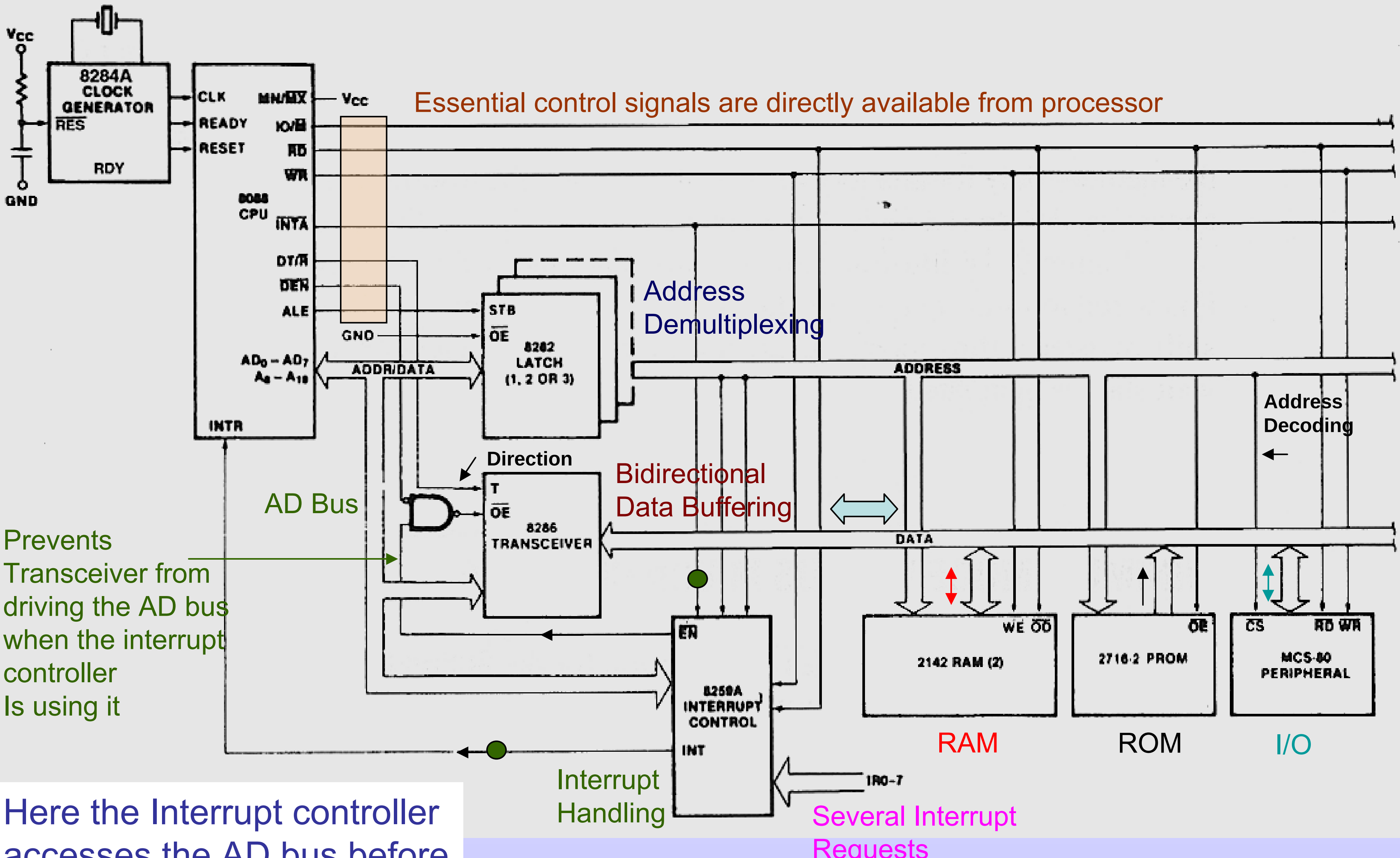
Note: #RD,#WR,#INTA are all inactive high during T1

Minimum and Maximum Modes

- MN/#MX input on 8088/8086 selects min (+5V) or max (0V) mode
- Minimum mode is the least expensive way to configure an 8086/8088 system:
  - Bus control signals are generated **directly** by processor
  - Good backward compatibility with earlier 8085A 8 bit processor
    - Same control signals
    - Support same peripherals
- Maximum mode provides greater versatility at higher cost.
  - New control signals introduced to support 8087 coprocessor (e.g. QS0 & QS1) and multiprocessor operation (e.g. #RQ/GT0 & RQ/GT1)
  - But important control signals omitted must be externally generated using an external bus controller, e.g. 8288. The controller decodes those control signals from the now compressed form of 3 control bits (#S0,#S1,#S2)
  - Can be used with the 8087 math coprocessor
  - Can be used with multiprocessor systems
- Maximum mode no longer supported since 80286

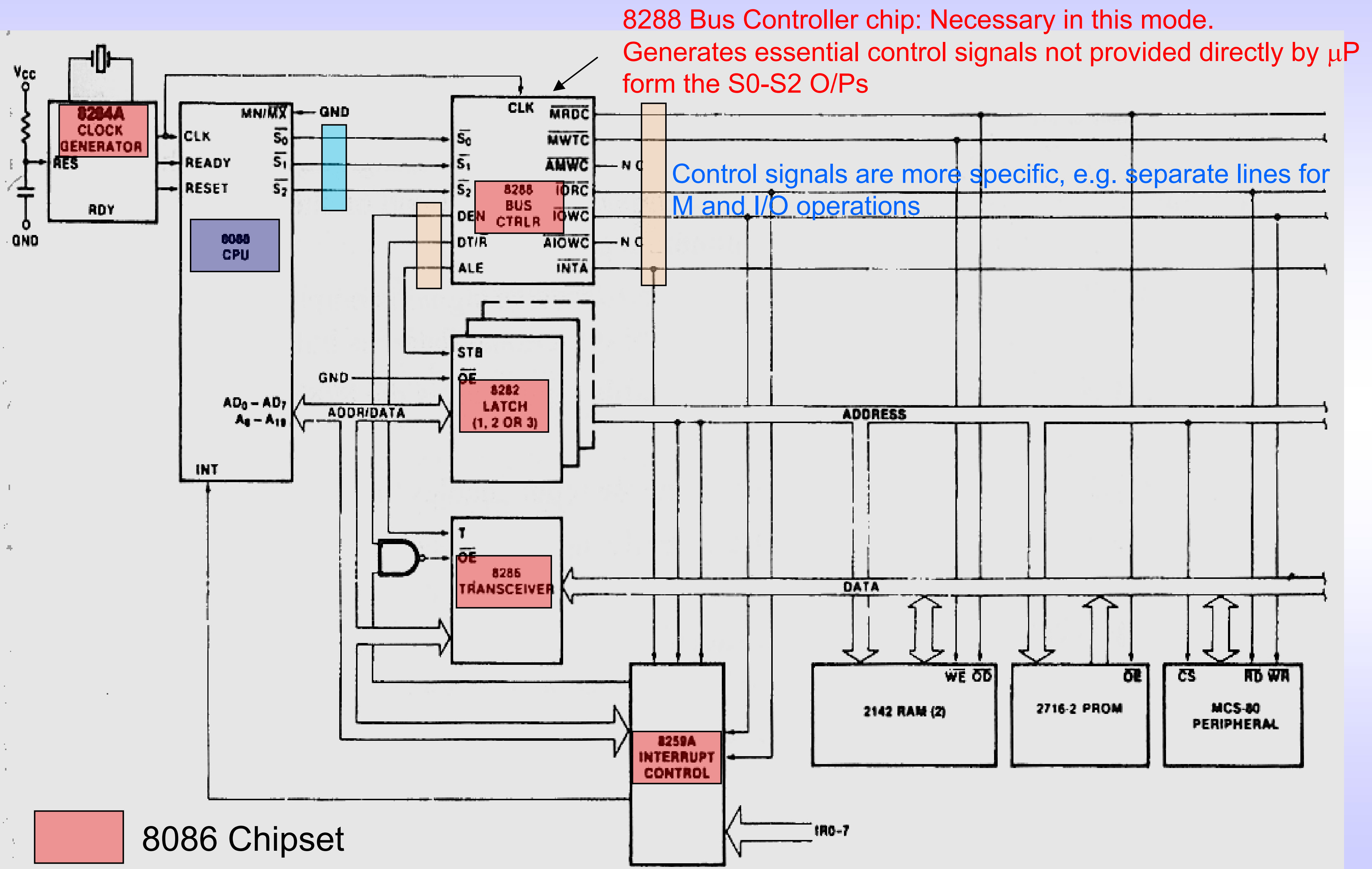
Use of 8086 in the Minimum Mode

Microprocessor-based System

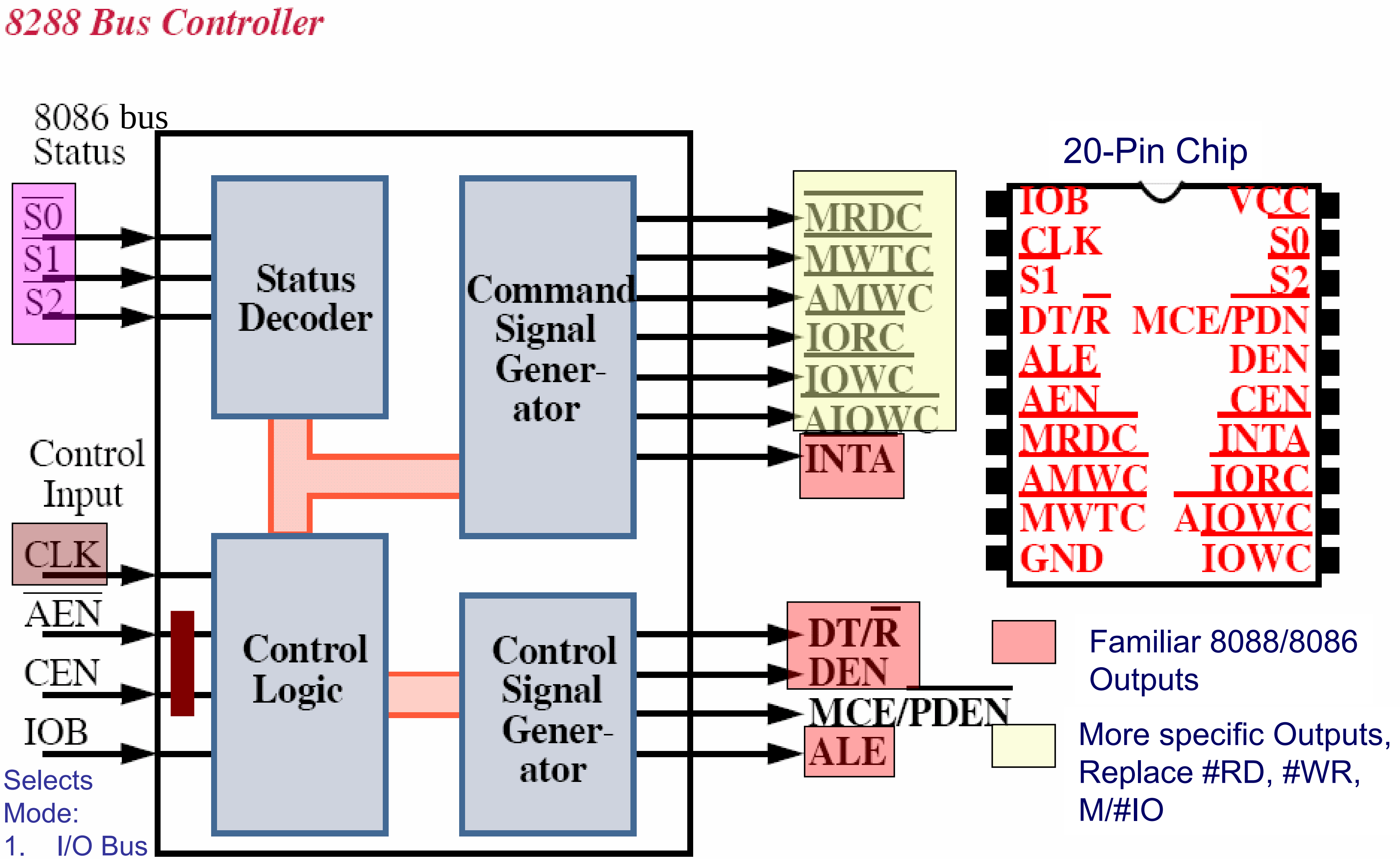


Here the Interrupt controller accesses the AD bus before demultiplexing - careful!

8086 Maximum Mode



8288 Bus Controller



Selects Mode:

1. I/O Bus
2. System Bus

8288 Bus Controller: Pin Functions

- S0, S1, S2 inputs: Status bus bits from processor. Decoded by the 8288 to produce the normal control signals
- CLK input: From the 8284A clock generator
- ALE output: Address latch enable output for demuxing address/data
- DEN output: Data bus enable output to enable data bus buffer. Note opposite polarity to #DEN output in minimum mode.
- DT/#R output: Data transmit/Receive output to control direction of the bi directional data bus.
- #INTA output: Acknowledge a hardware interrupt applied to the INTR input of the processor.
- IOB input: I/O bus mode input. Selects operation in either I/O bus mode or system bus mode.
- #AEN input: Address Enable input. Used by the 8288 to enable memory control signals. Supplied by a bus arbiter in a multiprocessor system
- CEN input: Control Enable input. Enables the generation of command outputs from the 8288.

Effective only in the system bus mode

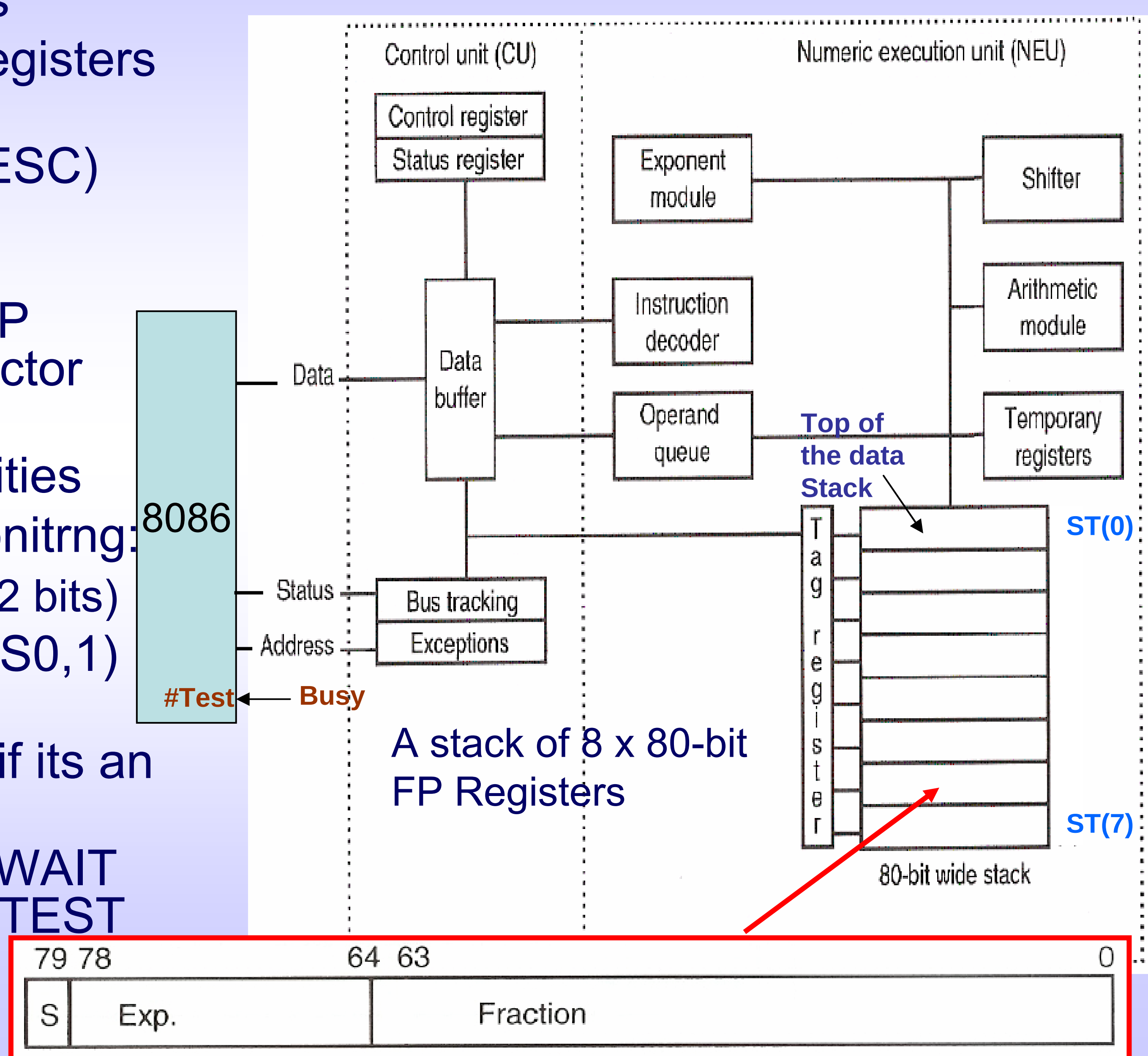
- #IORC output: Input/Output read control signal.
- #IOWC output: Input/Output write control signal.
- #AIOWC output: Advanced Input/Output control signal.
- #MRDC output: Memory read control signal.
- #MWTC output: Memory write control signal.
- #AMWT output: Advanced Memory write control signal.
- MCE/#PDEN output: Master cascade/Peripheral data output. Selects cascade operation if IOB=0 or enables I/O bus transceivers if IOB=5V

The Math Coprocessor: Chapter 14 (Numeric Data Processor (NDP))

- The 8086 performs integer math operations
- Floating point operations are needed, e.g. for Sqrt (X), sin (x), etc.
- These are complex math operations that require large registers, complex circuits, and large areas on the chip
- A general data processor avoids this much burden and delegates such operations to a processor designed specifically for this purpose - e.g. math coprocessor (8087) for the 8086
- The 8086 and the 8087 coprocessors operate in parallel and share the busses and memory resources
- The 8086 marks floating point operations as ESC instructions, will ignore them and 8087 will pick them up and execute them

The 8087 Coprocessor: Organization

- CU and NEU units
- Eight 80-bit FP Registers (data stack)
- Supports 68 FP (ESC) instructions
- Speeds up 8086 performance on FP operations by a factor of 50-100 time
- 8087 Tracks activities of the 8086 by monitrng:
  - Bus status (S0-S2 bits)
  - Queue status (QS0,1)
  - Instruction being fetched (to check if its an ESC instruction)
- Synchronize with WAIT using the BUSY-#TEST signals



Brey: The Intel Microprocessors, 7e

8086 Maximum mode outputs for NDP Connection

- **Bus Status Outputs S0-S2:** Status bits that encode the type of the current bus cycle

Table 9-7

$\overline{S2}$	$\overline{S1}$	$\overline{S0}$	Function
0	0	0	Interrupt acknowledge
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

- **Bus Request/Grant Outputs RQ0/GT0:** Allow 8087 to request use of the bus, e.g. for DMA memory access

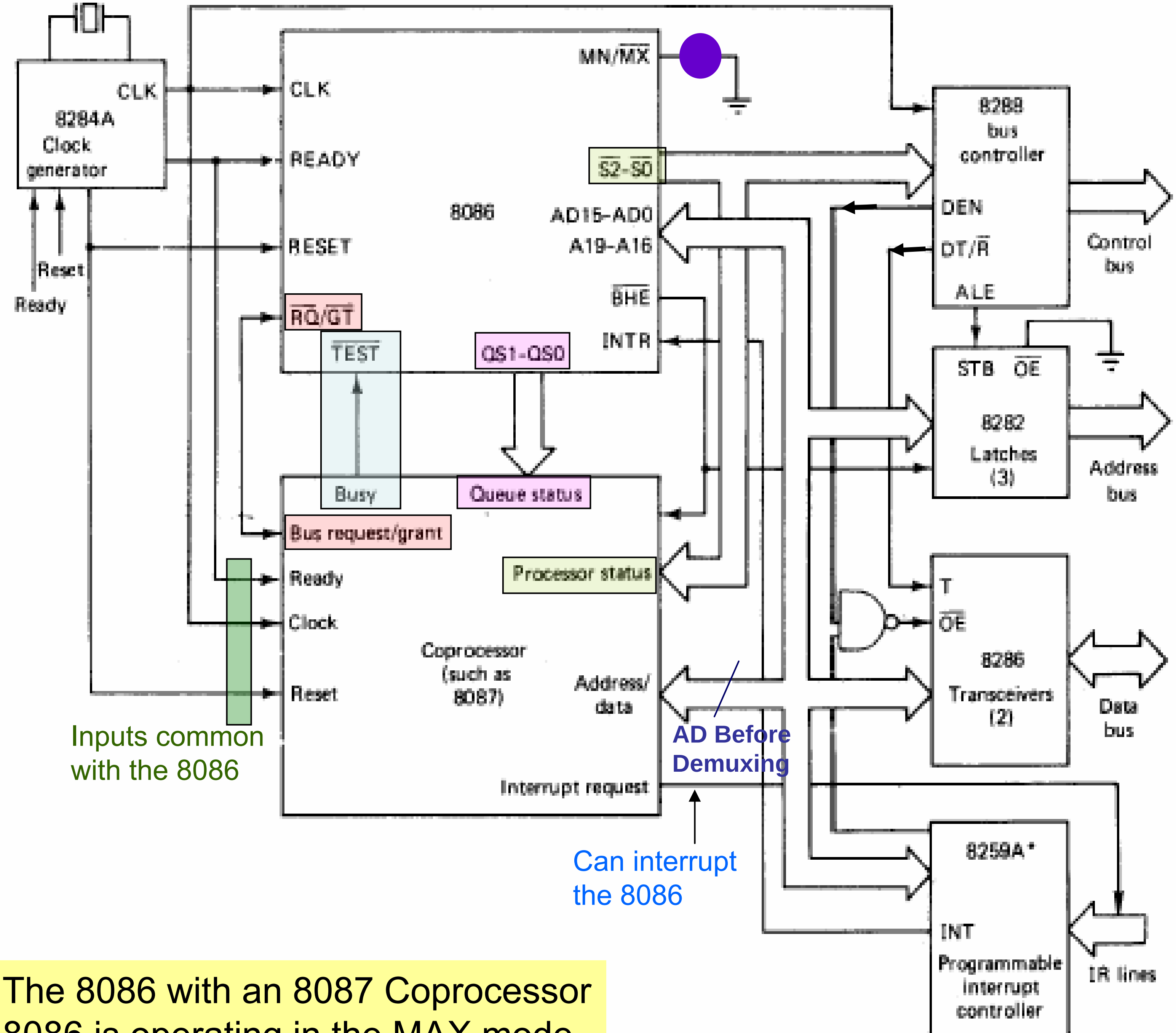
- **Queue Status Outputs QS1,QS0:**

  - For use by coprocessors that receive their instructions via ESC prefix.

  - Allow the coprocessor to track the progress of an instruction through the 8086 queue and help it determine when to access the bus for the escape op-code and operand.

  - Indicate the status of the internal instruction queue as given in the table:

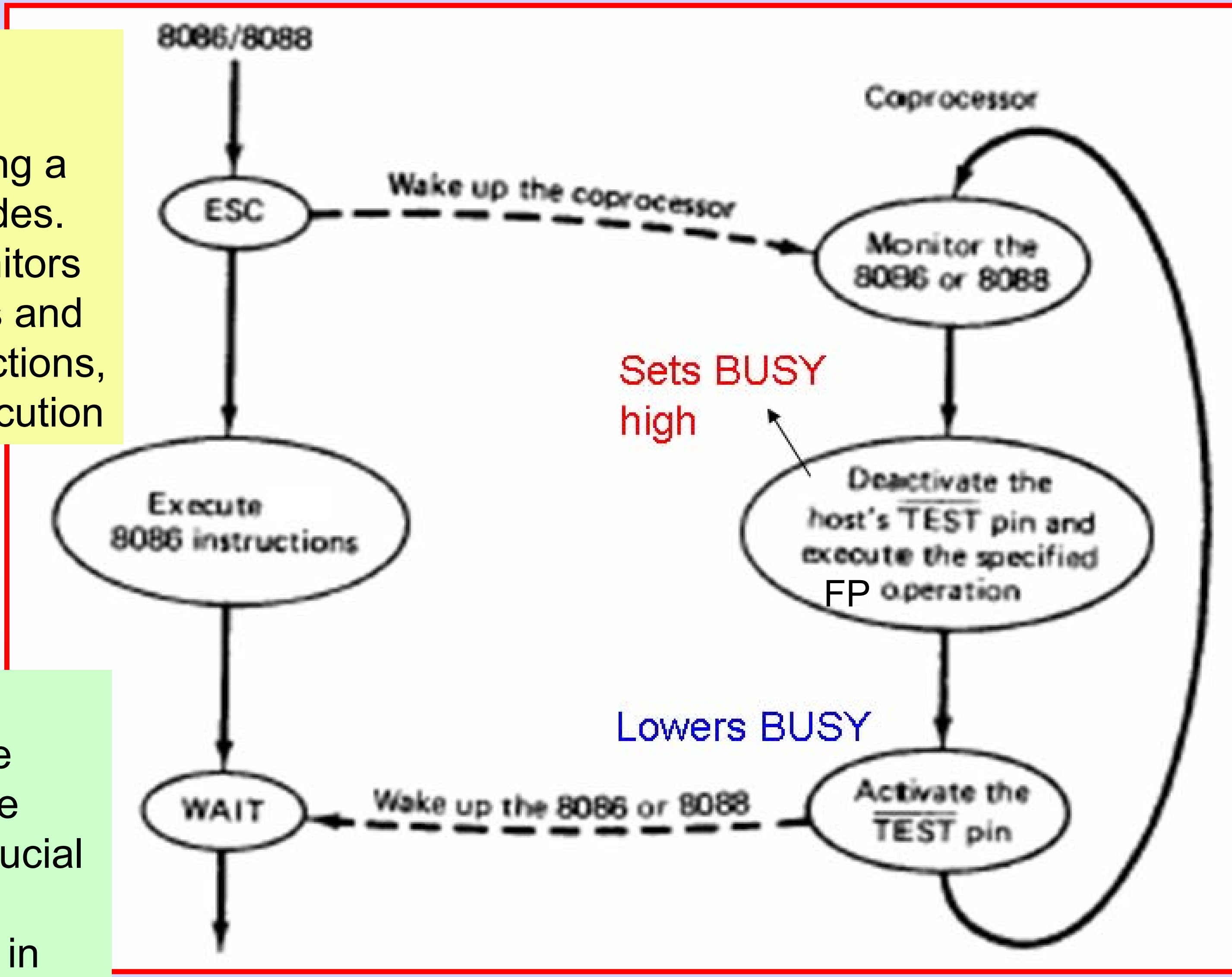
QS1	QS0	
0	0	Queue is idle
0	1	First byte of opcode from queue
1	0	Queue is empty
1	1	Subsequent byte of opcode from queue



The 8086 with an 8087 Coprocessor
8086 is operating in the MAX mode

Synchronization between 8086 & the 8087 Coprocessor

The assembler marks all FP instructions as ESC instructions having a special range of opcodes. The Coprocessor monitors the 8086 bus activities and Intercepts such instructions, captures them for execution



WAIT instructions can be used to halt the 8086 to ensure that the 8087 has finished a crucial step, e.g. storing FP results in memory.

Programming the 8087

Sequence of FP operations:

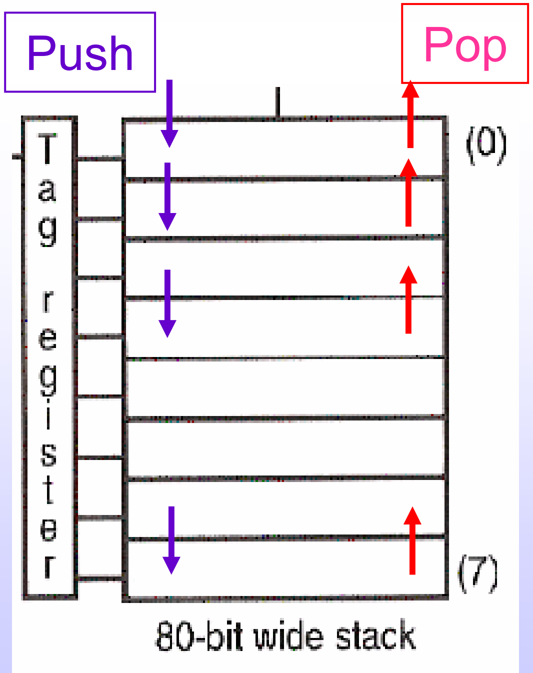
1. Operand data is loaded from memory into 8087 registers
2. Do the FP operation in the 8087
3. Store FP results from the 8087 to memory

• FP Instructions use the top of the 80-bit register data stack (ST (0)) as the default operand (needs not be mentioned), e.g.

FLDPI ; loads PI (= π) into the top of the stack. i.e. into register ST(0)

• When something is put on top of the stack, a stack PUSH occurs automatically

• When something is removed from the top of the stack, an automatic stack POP occurs



Programming the 80387: Example

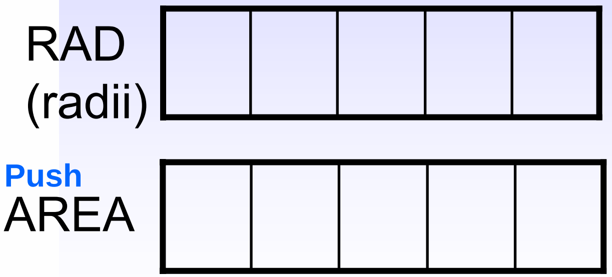
```
;A short procedure that finds the areas of 5 circles whose radii are stored
;in array RAD.
RAD     DD      2.34                ;array of radii
        DD      5.66
        DD      9.33
        DD      234.5
        DD      23.4
AREA    DD      5 DUP(?)            ;array for areas
FINDA   PROC    NEAR
        FLDPI                       ;load pi into Stack top ST(0), ST Push
        MOV  ECX,0                  ;initialize pointer ECX
                                    ; also used as loop counter
        .REPEAT
                FLD   RAD[ECX*4]    ;get radius  Into ST, Stack push
                FMUL  ST,ST(0)      ;square radius . ST = Stack top which is ST(0), result in ST
                FMUL  ST,ST(1)      ;multiply radius squared times pi
                FSTP  AREA[ECX*4]   ;store area into AREA array. Stack pop
                INC   ECX           ;index next radius
        .UNTIL ECX = 5              ;repeat 5 times
        FCOMP                       ;clear pi from coprocessor stack
        RET
FINDA   ENDP
```

Define Double Word (DD)

Instructions Starting with F are FP Instructions for the 80387

80386 Program

Two 5 x 4-byte array



Results in stack after Instruction Execution

Instruction	ST(0)	ST(1)
FLDPI	π	
FLD RAD[ECX*4]	2.34	π
FMUL ST,ST(0)	5.4756	π
FMUL ST,ST(1)	17.202	π
FSTP AREA[ECX*4]	π	

Push (FLDPI), Push (FLD), Pop (FSTP); Loop back

Note: 8087 automatically converts Data from integer to FP when moving it from memory to its data stack