

Microprocessors

CSE – 341

8086/8088 Hardware Specification

Chapter 9

Schmitt trigger

- The Schmitt trigger was invented by [US](#) scientist [Otto H. Schmitt](#) in 1934 while he was still a graduate student,^[1] later described in his doctoral dissertation (1937) as a "thermionic trigger."^[2] It was a direct result of Schmitt's study of the neural impulse propagation in [squid](#) nerves.^[2]

Squid

- <http://en.wikipedia.org/wiki/Squid>

Trigger Circuit

- A circuit or network in which the output changes abruptly with an infinitesimal change in input at a predetermined operating point. Also known as trigger.

Demonstration

- www.falstad.com/circuit/e-index.html
-



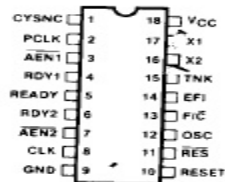
PRELIMINARY
 Notice: This is not a final specification. Some parametric limits are subject to change.

8284 CLOCK GENERATOR AND DRIVER FOR 8086, 8088, 8089 PROCESSORS

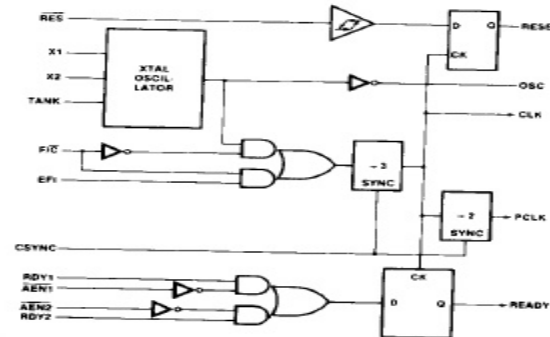
- Generates the System Clock for the 8086, 8088 and 8089
- Uses a Crystal or a TTL Signal for Frequency Source
- Single +5V Power Supply
- 18-Pin Package
- Generates System Reset Output from Schmitt Trigger Input
- Provides Local Ready and MULTIBUS™ Ready Synchronization
- Capable of Clock Synchronization with other 8284's

The 8284 is a bipolar clock generator/driver designed to provide clock signals for the 8086, 8088 & 8089 and peripherals. It also contains READY logic for operation with two MULTIBUS™ systems and provides the processors required READY synchronization and timing. Reset logic with hysteresis and synchronization is also provided.

8284 PIN CONFIGURATION



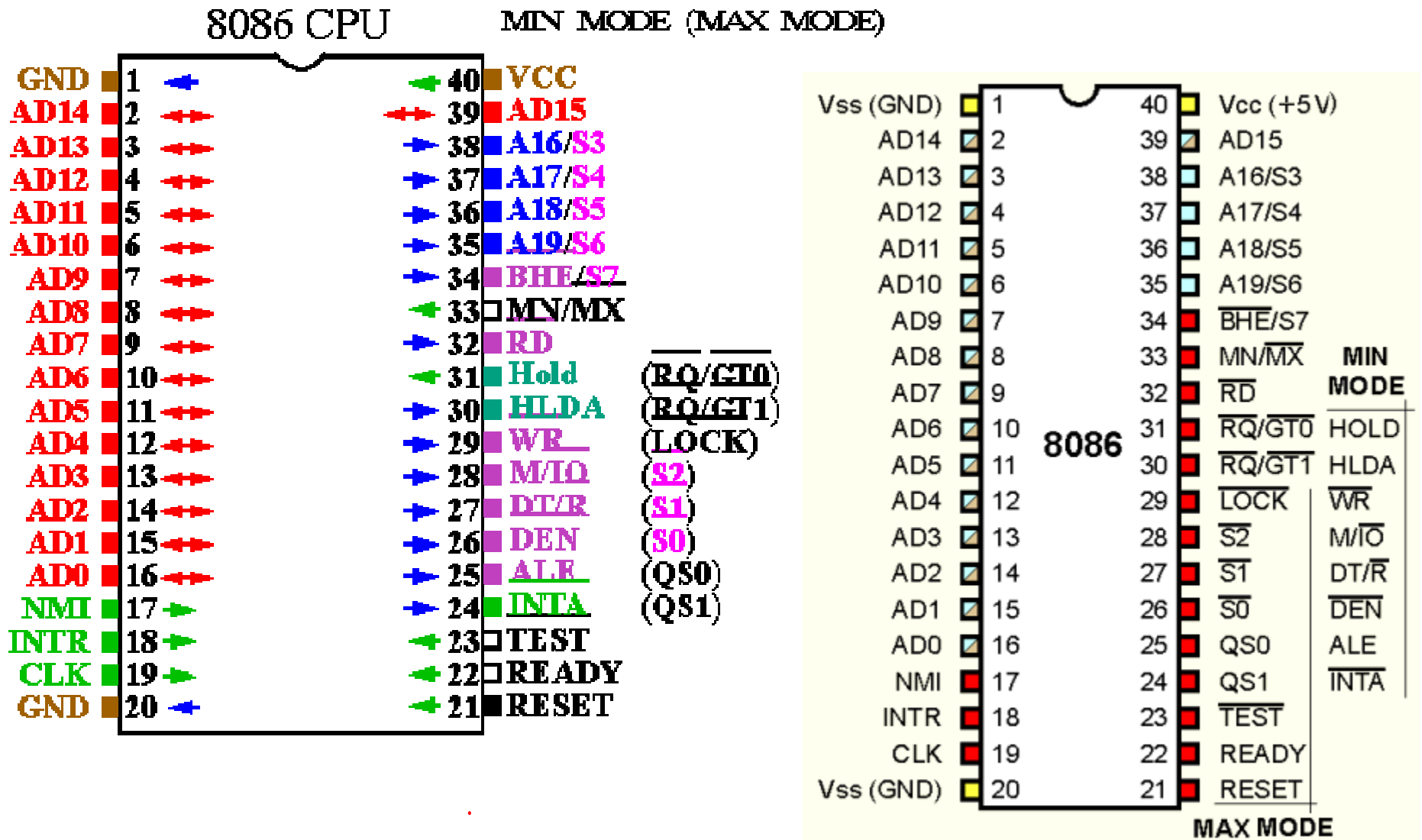
8284 BLOCK DIAGRAM



8284 PIN NAMES

X1	CONNECTIONS FOR CRYSTAL
X2	
TANK	USED WITH OVERTONE CRYSTAL
FIC	CLOCK SOURCE SELECT
EFI	EXTERNAL CLOCK INPUT
CYSNC	CLOCK SYNCHRONIZATION INPUT
RDY1	READY SIGNAL FROM TWO MULTIBUS™ SYSTEMS
RDY2	
AEN1	ADDRESS ENABLED QUALIFIERS FOR RDY1,2
AEN2	
RES	RESET INPUT
RESET	SYNCHRONIZED RESET OUTPUT
OSC	OSCILLATOR OUTPUT
CLK	MCS CLOCK FOR THE PROCESSOR
PCLK	TTL CLOCK FOR PERIPHERALS
READY	SYNCHRONIZED READY OUTPUT
VCC	+5 VOLTS
GND	0 VOLTS

8086 Pin Specification



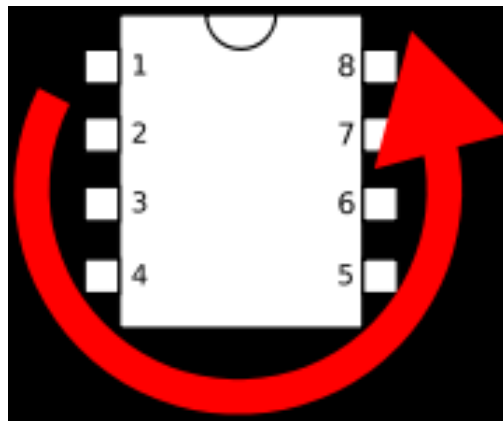
Both 8086 / 8088 are packaged as 40-pin DIPs.

In micro-electronics **DIP** stands for **Dual in-line package**.

DIP packaging refers to a rectangular housing with two parallel rows of electrical connection pins.

DIP chips have a notch one one end to show its correct orientation.

The pins are then numbered **ACW** as shown in the figure below.



8086 is a 16 bit microprocessor with a 16 bit data bus and the 8088 is a 16 bit microprocessor with an 8 bit data bus.

So why is 8088 a 16 bit microprocessor if its data bus is 8 bits ?

What factor decides how many bits a microprocessor is regarded to be?

Differences between 8086 and 8088:

- Data bus size is different.
- Control signal – $\overline{M}/\overline{IO}$ (8086)
 IO/\overline{M} (8088)
- Hardware Difference – Pin 34 – \overline{SSO} Pin (8088)
 $\overline{BHE}/S7$ pin (8086)

Power Requirements

Both Processors require +5.0V with a supply voltage tolerance of $\pm 10\%$.

8086 draws a maximum current supply of 360mA

8088 draws a maximum current supply of 340mA

Ambient temperature for use is 32°F – 180°F

Not suitable for outdoor use.

DC Characteristics

Remember we have been talking about 0 and 1s so far.

In every connection we said that we either send a 0 or a 1 through the pins.

So what is 0 and what is 1? How do we represent a 0 and 1 on the pin in real life ?

We use voltage.

But, then what voltage counts as 1 and what counts as 0?

It is also important to know the current of the pins so that we do not interface that may work incorrectly and may even damage the processor.

INPUT			OUTPUT		
Logic level	Voltage	Current	Logic level	Voltage	Current
0	0.8V max	+/- 10uA max	0	0.45V max	+2mA max
1	2.0V min	+/- 10uA max	1	2.4V min	- 400uA max

What do you notice here in regards to the gap between the voltage that represents 0 and 1.

I don't want you to memorize these numbers ? So why do you think I am showing you these ?

The gap in the voltage is known as **noise immunity**.

Noise refers to unwanted signal within some signal that you are trying to send.

The noise immunity is the amount of noise the can be tolerated before an error would occur.

The processor has a noise immunity of 350mV.

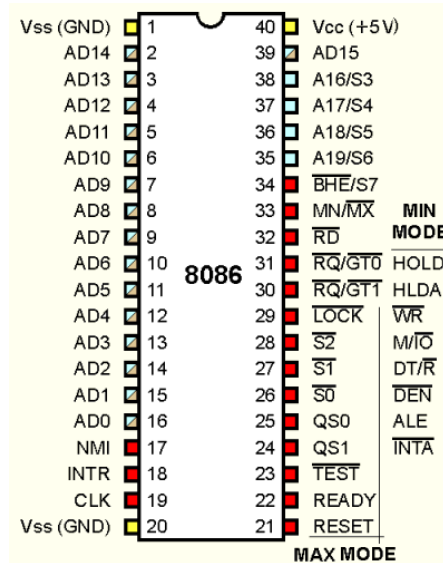
Noise can occur from long connection links or from connecting multiple devices.

Long links are hardly used during interfacing but multiple devices are connected to a microprocessor.

It is recommended that no more than 10 loads of any type or combination be connected to an output pin without **buffering**.

We will study this in a while when we speak about buffering in details.

First we need to discuss the pin connections that we have on the processor as shown in the diagram previously.



Pin Connections

AD15 – AD0:

Address/Data bus lines. These are multiplexed lines.

Line carries address when ALE = 1

Line carries data when ALE = 0

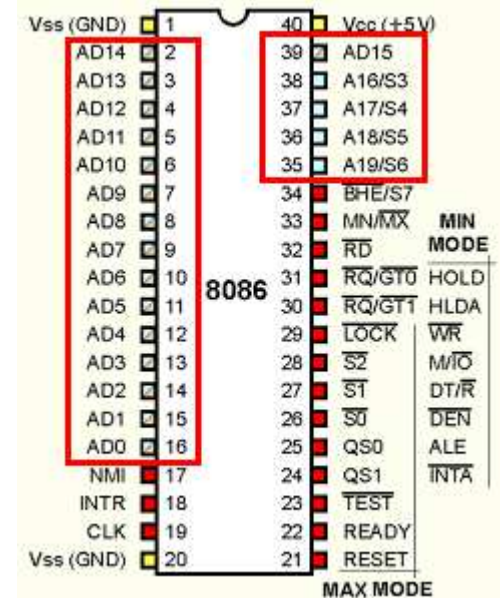
AD19/S6 – A13/S3:

Address/Status bus bits are multiplexed to provide address signals A19-A16 and status bits S6-S3.

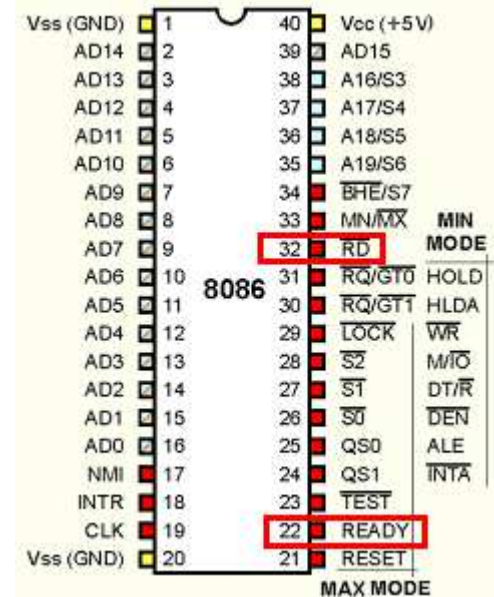
S6 – always remains 0

S5 – indicates the condition of the interrupt flag

S4 and S3 – Indicate the segment being accessed during current bus cycle.



S4	S3	Function
0	0	Extra segment
0	1	Stack segment
1	0	Code or no segment
1	1	Data segment



\overline{RD} :

When this **read signal** pin is at logic 0, the data bus is receptive to data from memory or I/O devices.

READY:

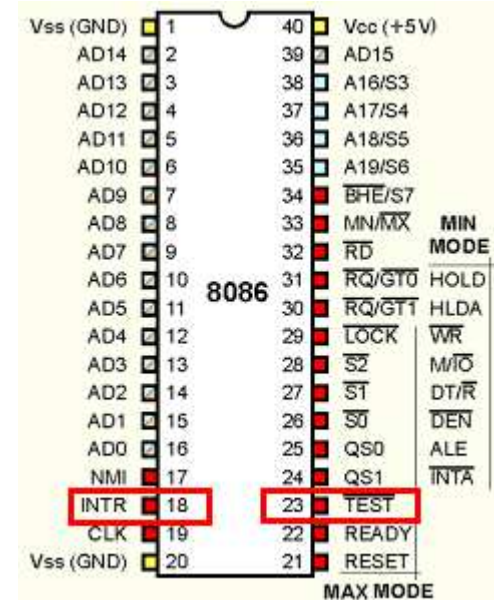
This pin is used to enforce a waiting state.

READY pin at 0 – the microprocessor goes into idle state.

READY pin at 1 – the microprocessor does normal operation.

INTR:

Interrupt request pin is used to request a hardware interrupt. If INTR is held at high when $IF = 1$, the processor goes into the interrupt acknowledgement cycle. INTA becomes active when interrupt is being serviced.

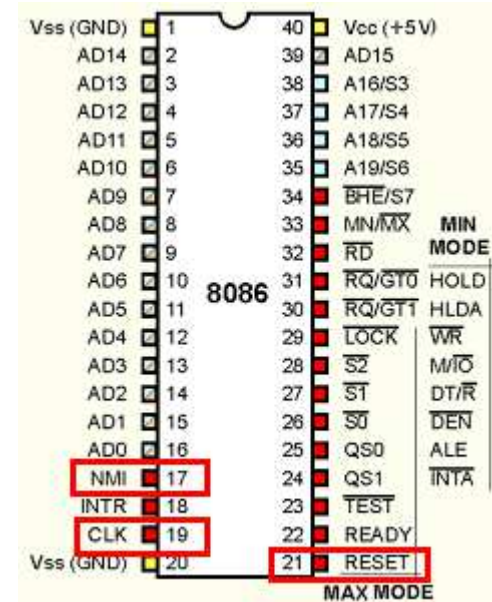


TEST:

Test pin is an input that is tested by the **WAIT** instruction. If the test pin is at logic 0 the WAIT instruction functions as NOP. If test is a logic 1, the WAIT instruction wait for TEST to become logic 0. Commonly used with 8087 numeric coprocessor connections.

NMI:

Non-maskable interrupt input is similar to INTR expect that the NMI interrupt does not check IF or priority. Use Interrupt Vector 2.



RESET:

If this **reset** pin is held high for 4 clock cycles the microprocessor resets. When 8086 or 8088 is reset it begins execution at memory location FFFF0H and clears the IF.

CLK:

The **clock pin** is used to connect a clock generator

Vcc:

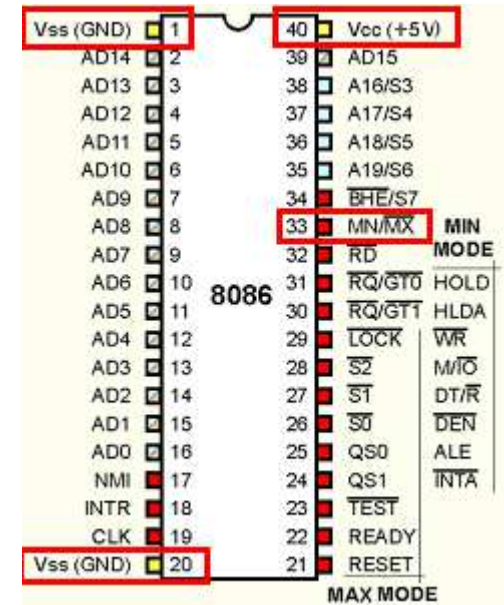
The power supply. +5V should be connected to this pin.

GND:

The **ground** connection for the microprocessor.

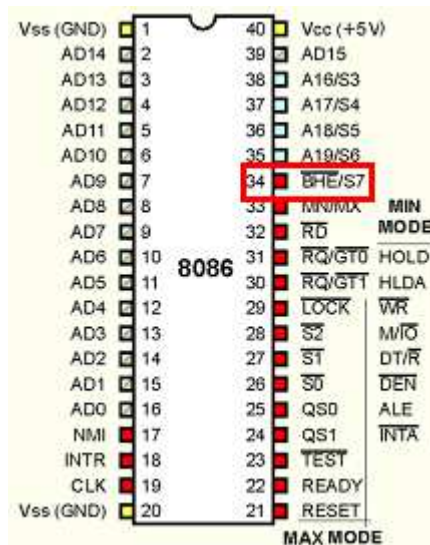
MN/ $\overline{\text{MX}}$:

The **minimum/maximum mode** pin selects the mode for the processor. To select minimum mode processor should be connected directly to +5.0V and to select maximum mode processor should be connected directly to GND.

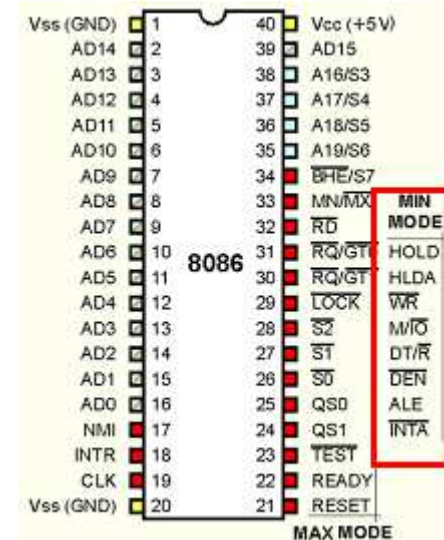


BHE/S7:

The **bus high enable** pin is used in the 8086 to enable the Most significant data bus bits during a read or write operation.



Pin Connections (Minimum)



$\overline{\text{IO/M}}$

This pin indicates whether the address bus contains a memory address or an I/O port address.

$\overline{\text{WR}}$:

The **write line** is used when the microprocessor is writing data to memory and the memory bus contains a valid address.

INTA:

Interrupt acknowledgement signals is a response to INTR input pin. This is used when the interrupt vector is placed on the address bus by the microprocessor.

ALE:

Address Latch enable shows whether the multiplexed AD lines carry address or data.

DT/ \bar{R} :

Data transmit/receive shows that the microprocessor data bus is transmitting(1) or receiving(0) data. This is used to control buffers.

DEN:

Data Enable bus activates external data bus buffers.

HOLD:

HOLD pin is used to input request DMA. Hold set to 1 microprocessor gives up control of buses to DMA controller.

$\overline{SS0}$:

This is equivalent **S0** in the maximum mode pins.

Maximum Mode Pins

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

These signal bits indicate the function of the current bus cycle. These pins are used for special purpose which we will discuss in a bit.

$\overline{R0/GT1}$ and $\overline{R0/GT0}$:

Requests/grants pins request direct memory access during maximum mode operation.

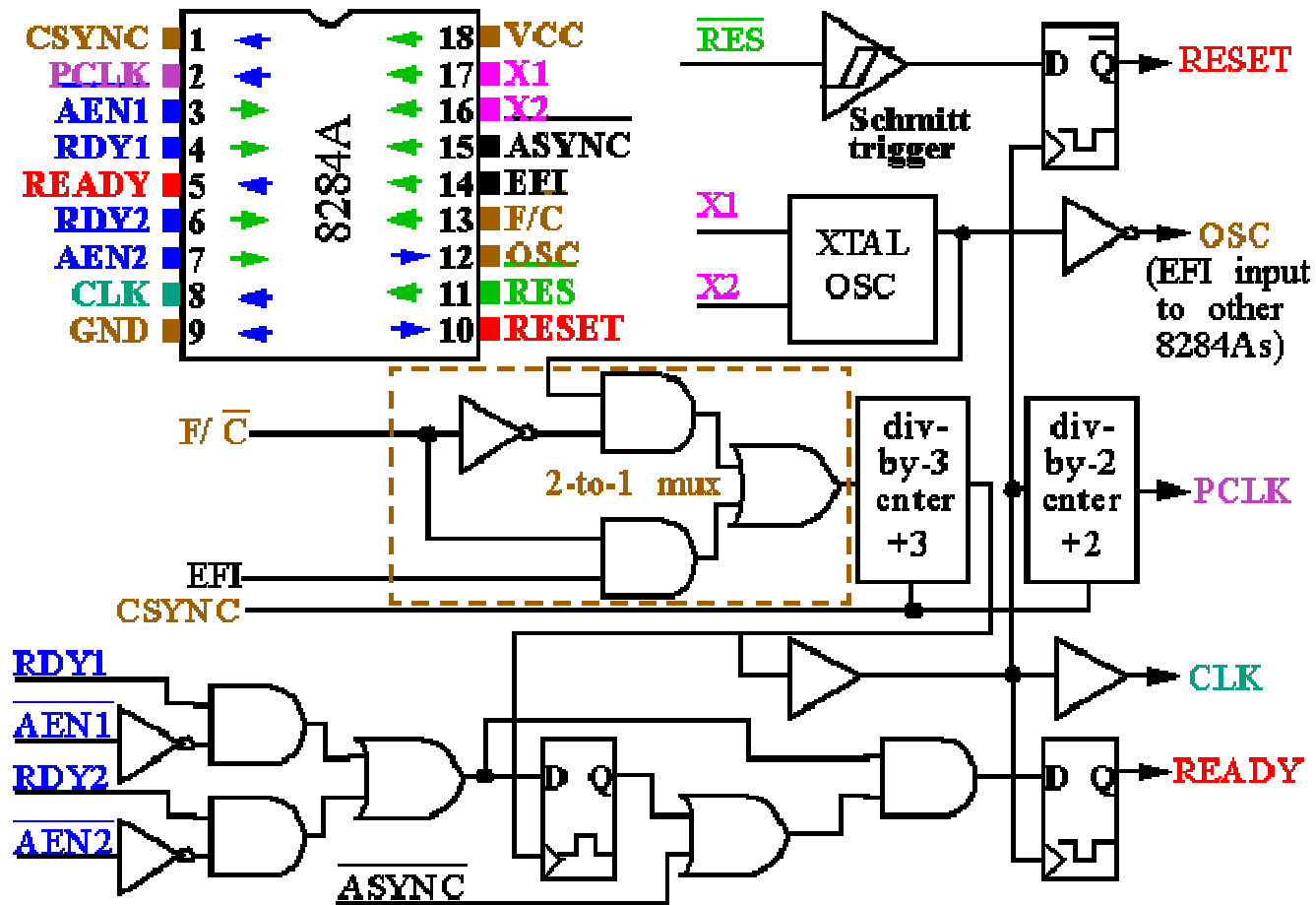
\overline{LOCK} :

Lock output is used to lock peripherals off the system.

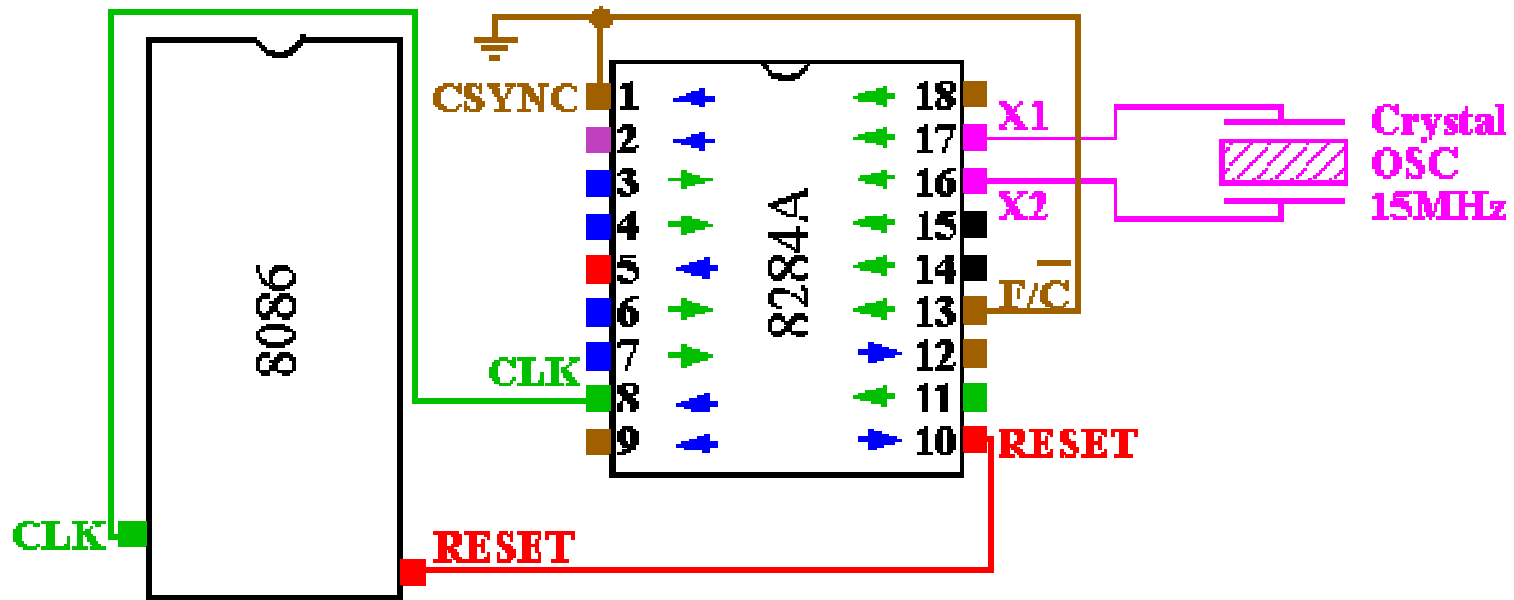
QS0 and QS1:

Queue status bits show the status of the internal instruction queue.

Clock Generator 8284A



Clock Generator



Functions of Interest to us now:

- Clock generation – The clock generation uses a crystal. Crystal generates square waves at the frequency of the crystal. There is a divide-by-3 counter causing the frequency of the crystal to be divided by 3 at the 8086. So a 15 MHz XTAL causes the 8086 to run at 5MHz.
- RESET Synchronization – The reset are synchronized and to reset the processor pin must be held high for 4 cycles.
- Ready Synchronization – DMA sync may require clock to be stopped during waiting state.
- Peripheral clock signal. – The peripheral frequency has a further divide-by-2 counter. So 15 MHz XTAL gives clock frequency of 2.5MHz at the peripherals.

Bus Buffering and Latching

The microprocessor has 3 buses:

- Control
- Address
- Data

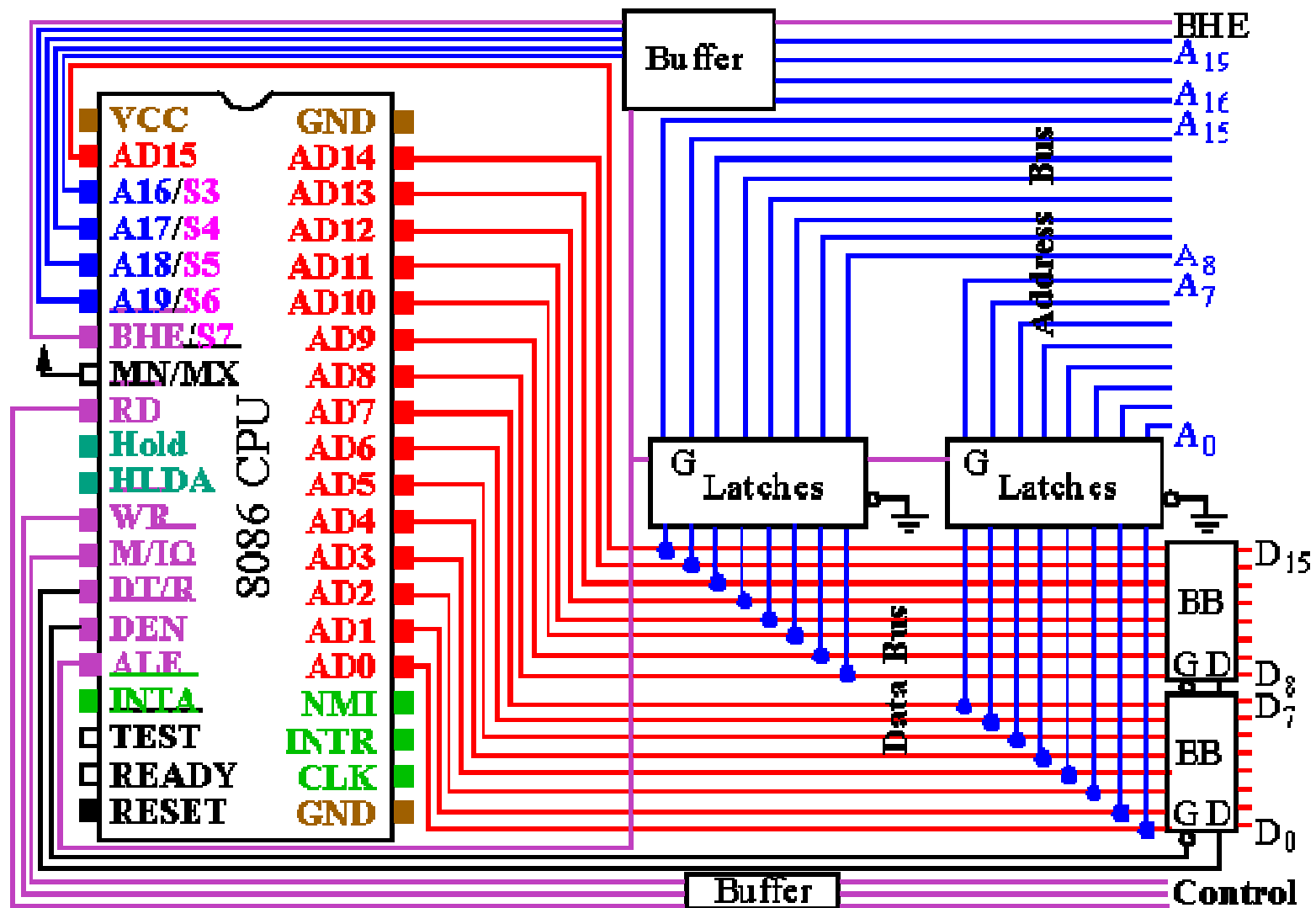
The address and the data bus are multiplexed due to pin limitations. The ALE pin is used to control the set of latches.

Why latch?

So that we can control devices that use the same common buss which may be carry either data or address information.

Why Buffer?

For read and write operations to be correctly implemented the data on the lines need to be stable throughout the instructions. Due to multiplexing and switching between instructions that make different use of buses we need to buffer data in order for us to have stable data through out an instruction.



Bus Timing

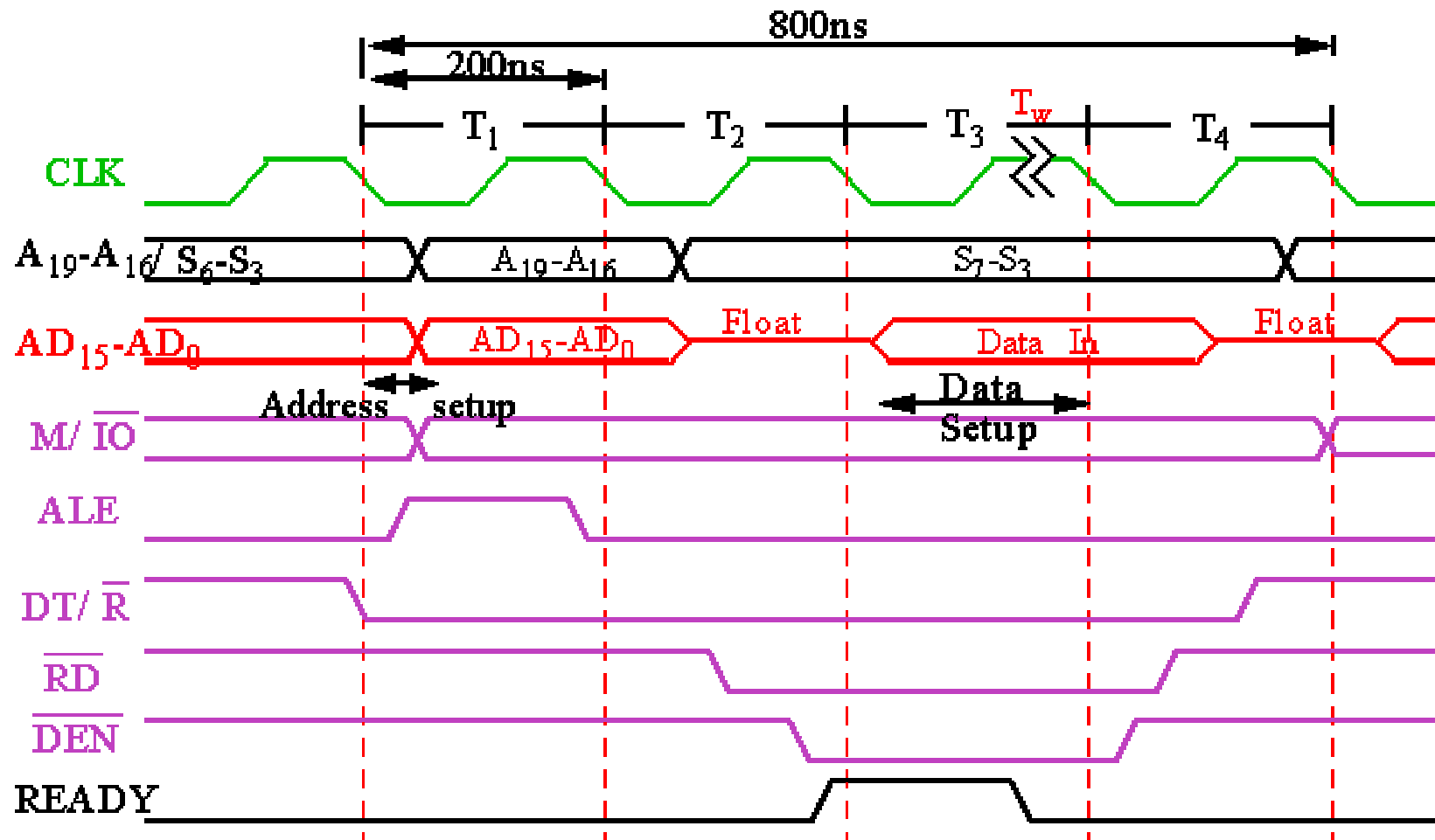
The 8086/8088 microprocessors use the memory and I/O in periods called **bus cycles**.

Each bus cycle consists of 4 clock cycles.

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

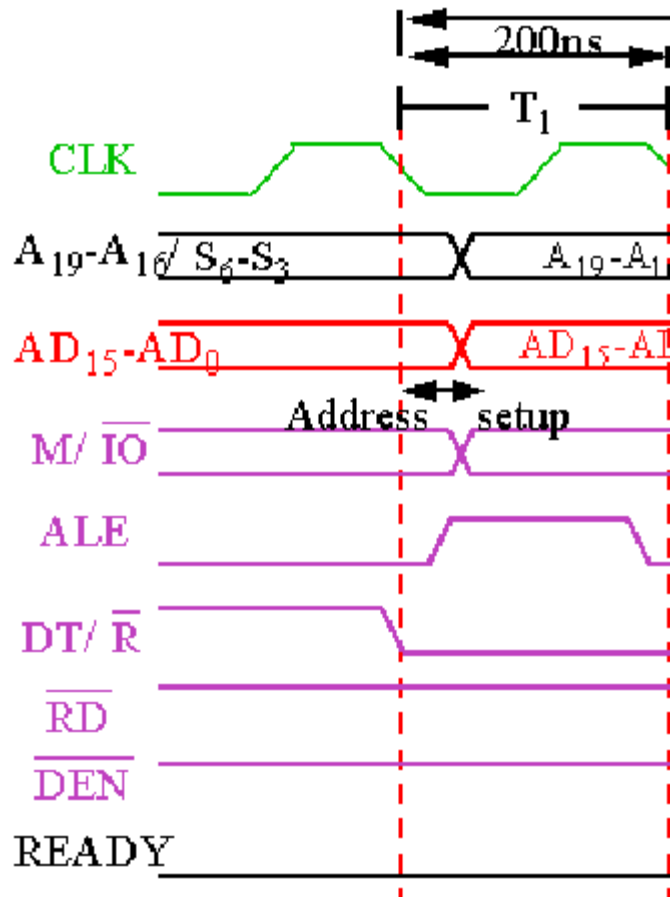
Each read or write operation take 1 bus cycles.

Read Timing



Bus Timing for a Read Operation

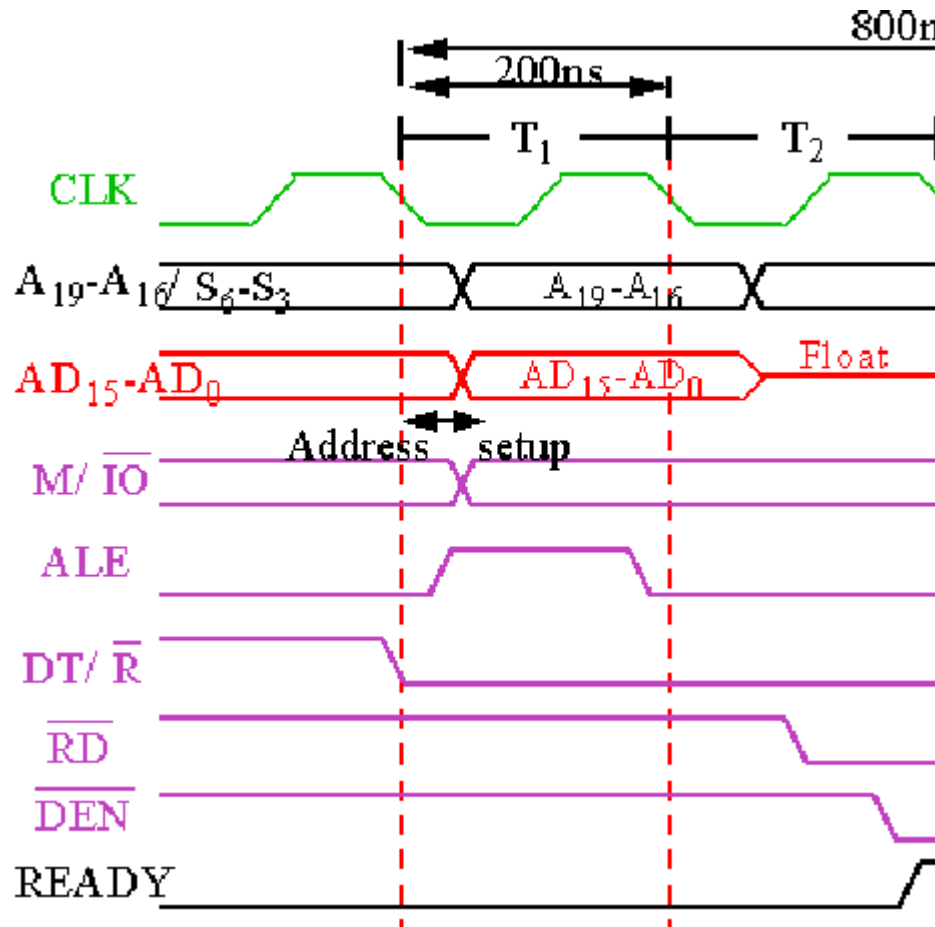
Read Timing



During T_1 :

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

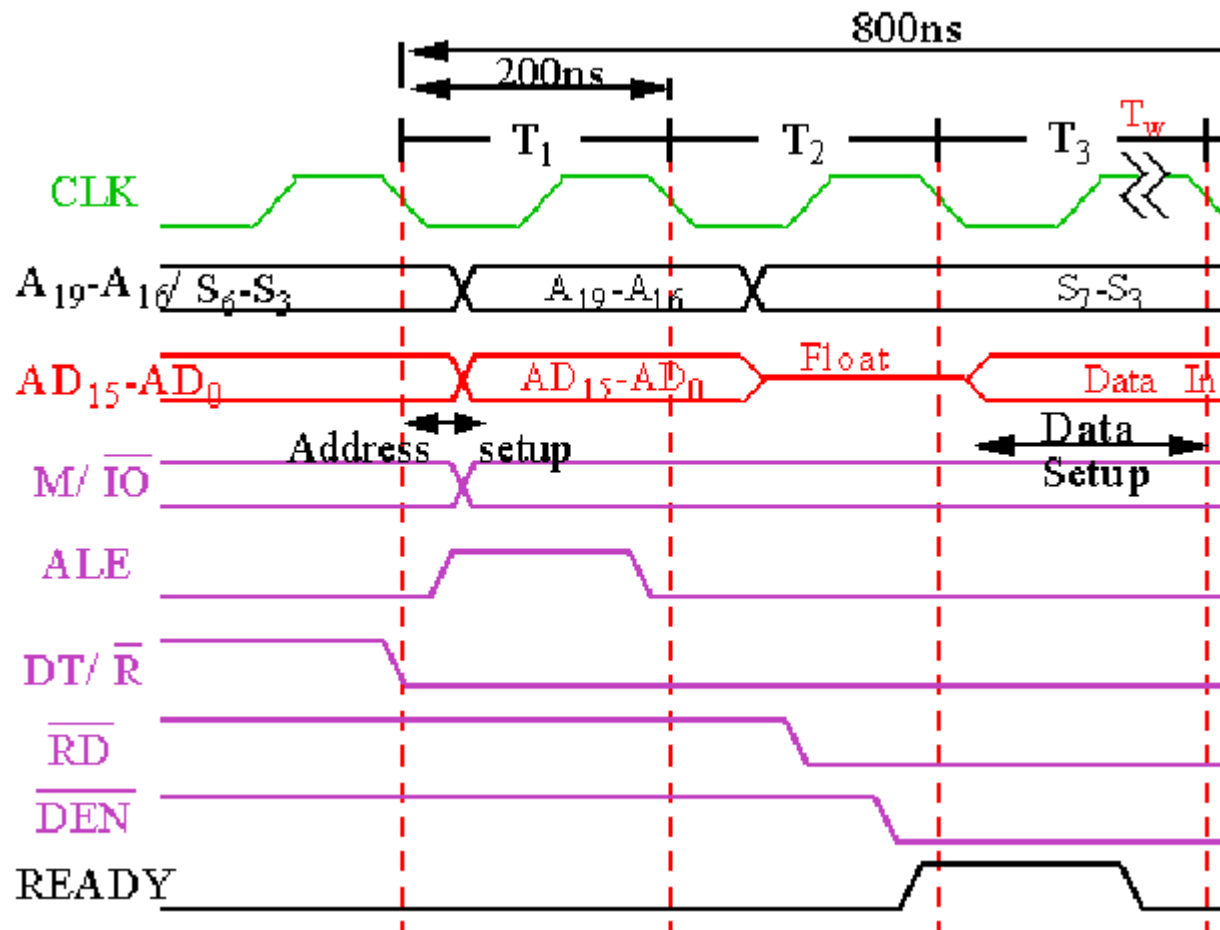
Read Timing



During T₂:

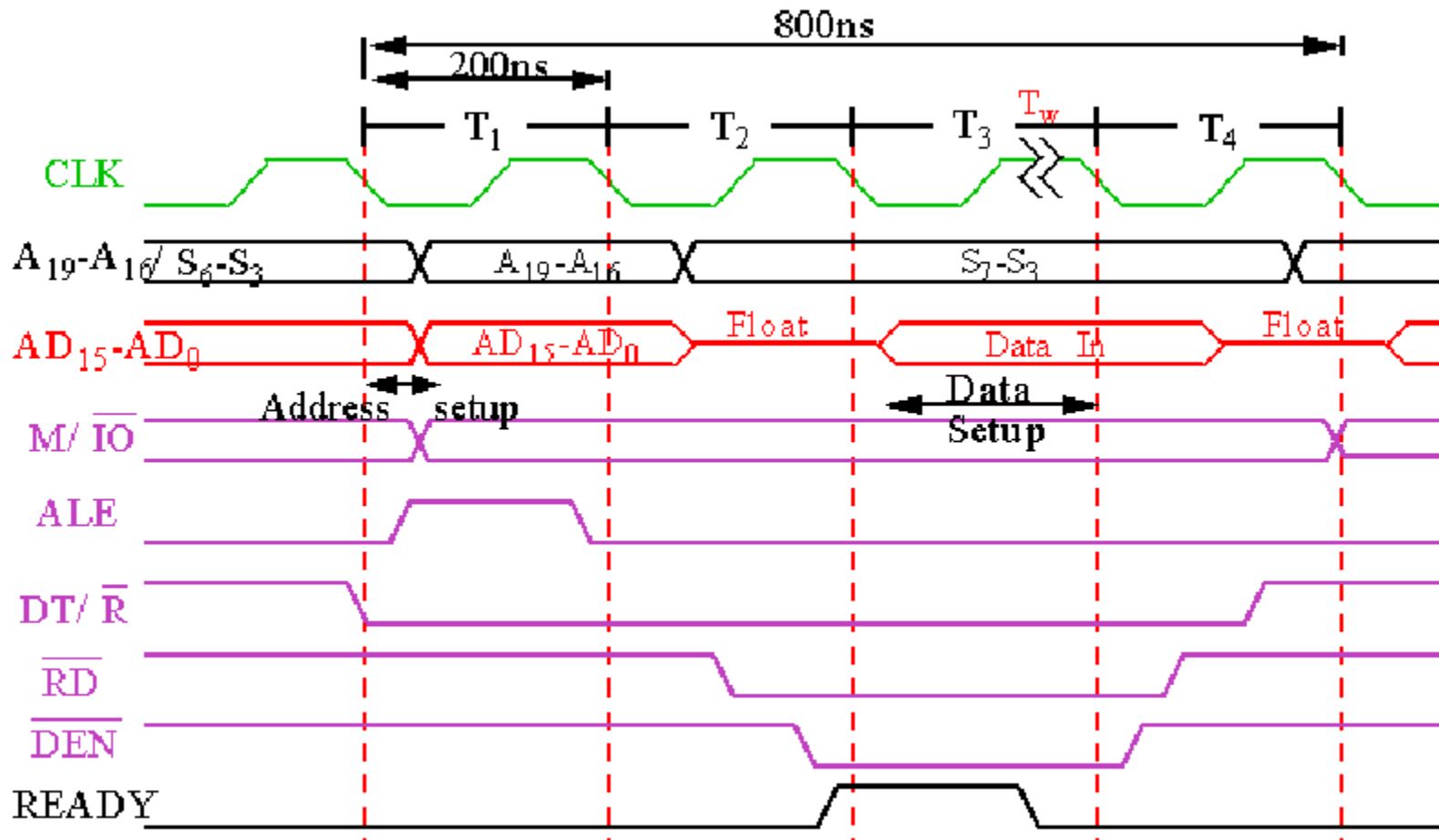
- 8086 issues the RD or WR signal, DEN , and, for a write, the data.

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



During T_3 :

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

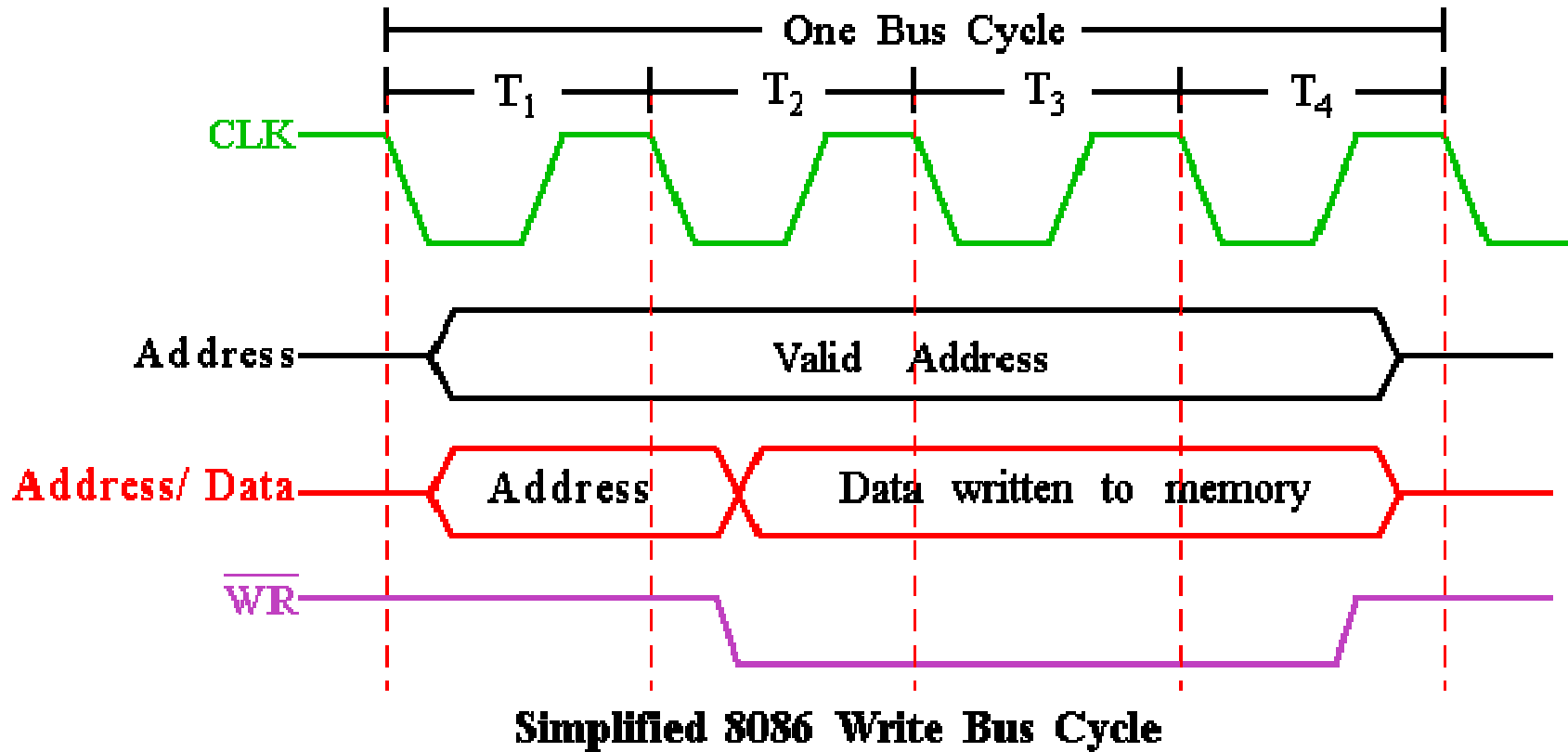


Bus Timing for a Read Operation

During T₄:

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

Write Timing

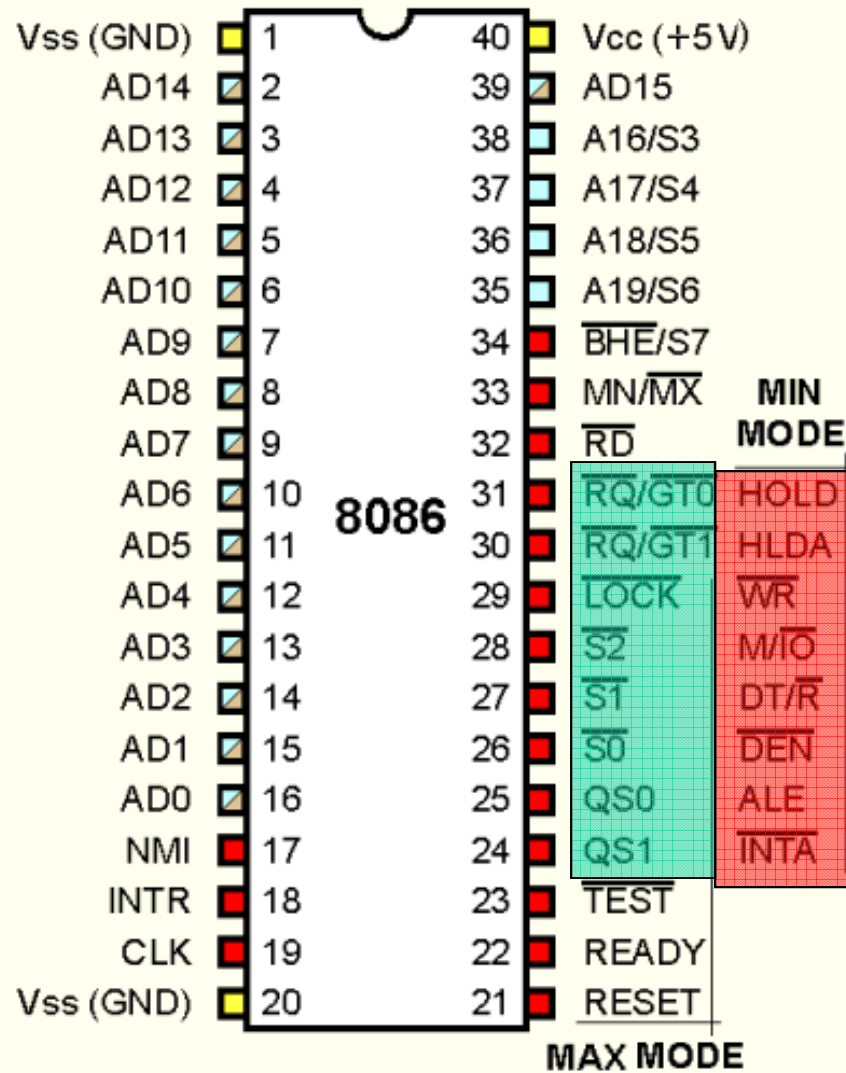


**Convert this simple timeline to include all the necessary pins
Do as an EXERCISE !**

MIN and MAX Mode

- *Controlled through the MN/MX pin.*
- **Minimum mode** *is cheaper since all control signals for memory and I/O are generated by the microprocessor.*
- **Maximum mode** *is designed to be used when a coprocessor (8087) exists in the system.*

What is the apparent problem ?



- Some of the control signals must be generated externally, due to redefinition of certain control pins on the 8086.

- The following pins are lost when the 8086 operates in **Maximum mode** .

- ALE

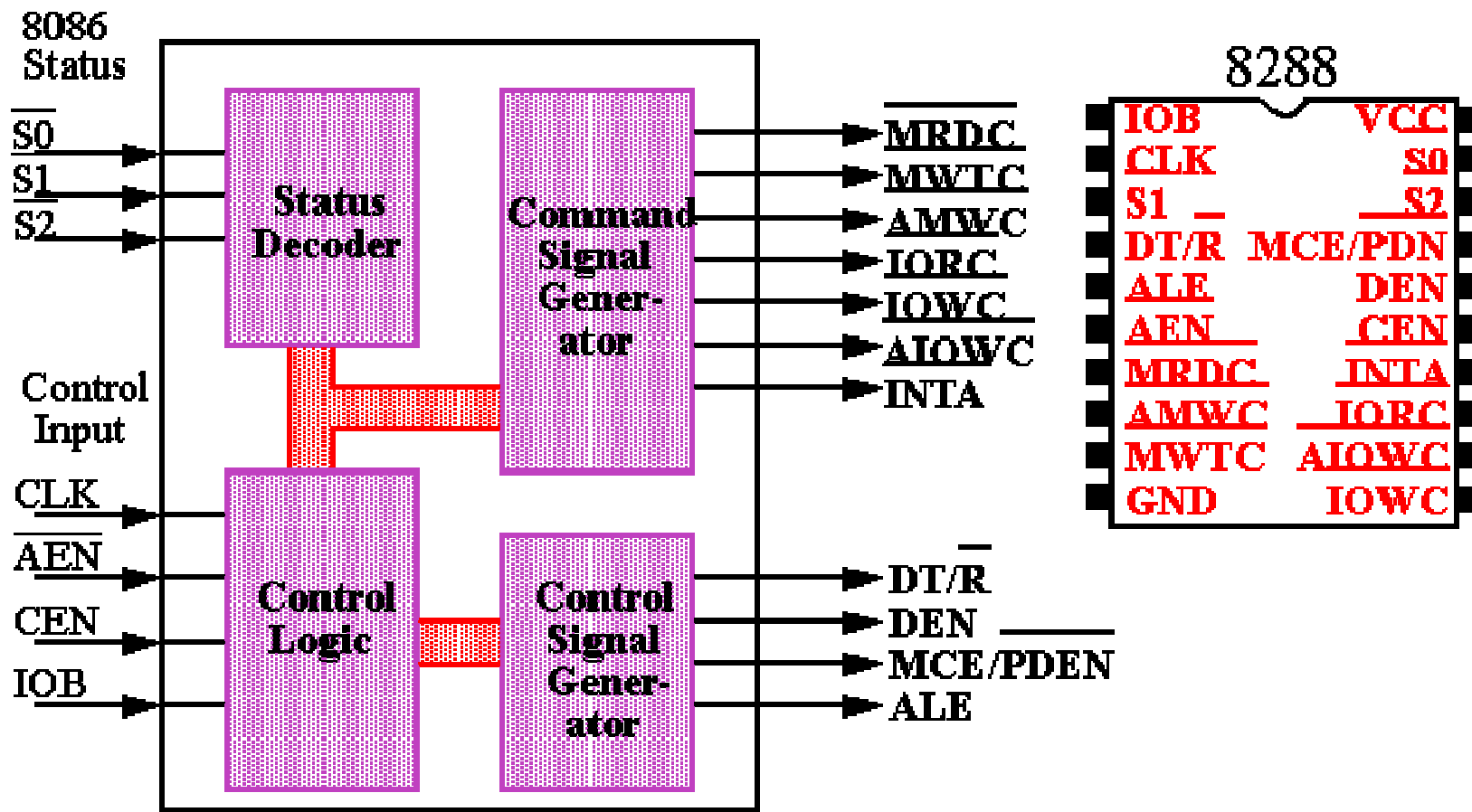
- WR

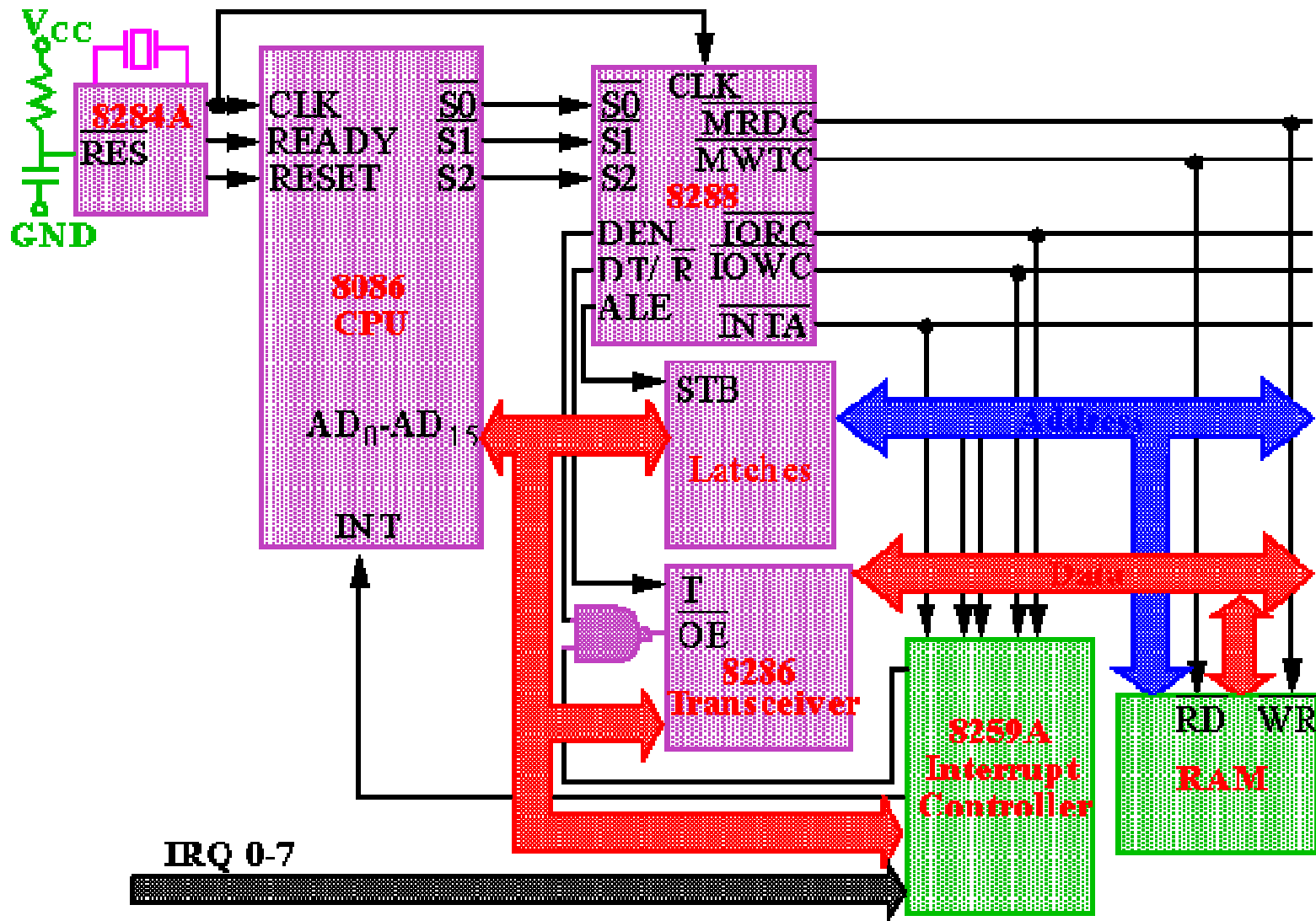
- IO/ M

- DT/ R

- DEN

- INTA





No need to learn 8288 architecture. Its just for reference!

Questions ?

