8086 Hardware Specifications

Dept. of Computer Science and Engineering
BRAC University
CSE 341 Team

Lecture References:

- **Book:**
  - *Microprocessors and Interfacing: Programming and Hardware,*

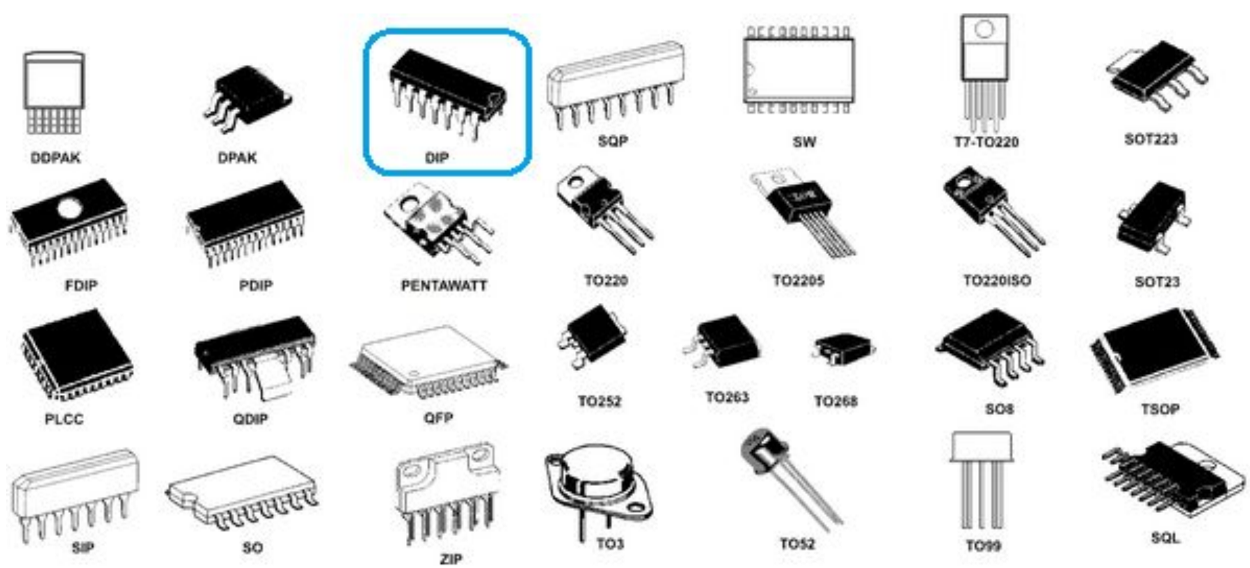
    Author: Douglas V. Hall

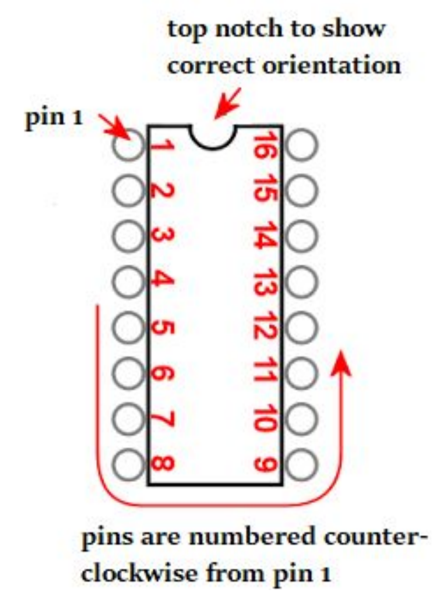
  - *The 8086/8088 Family: Design, Programming, And Interfacing,*

    Author: John Uffenbeck.

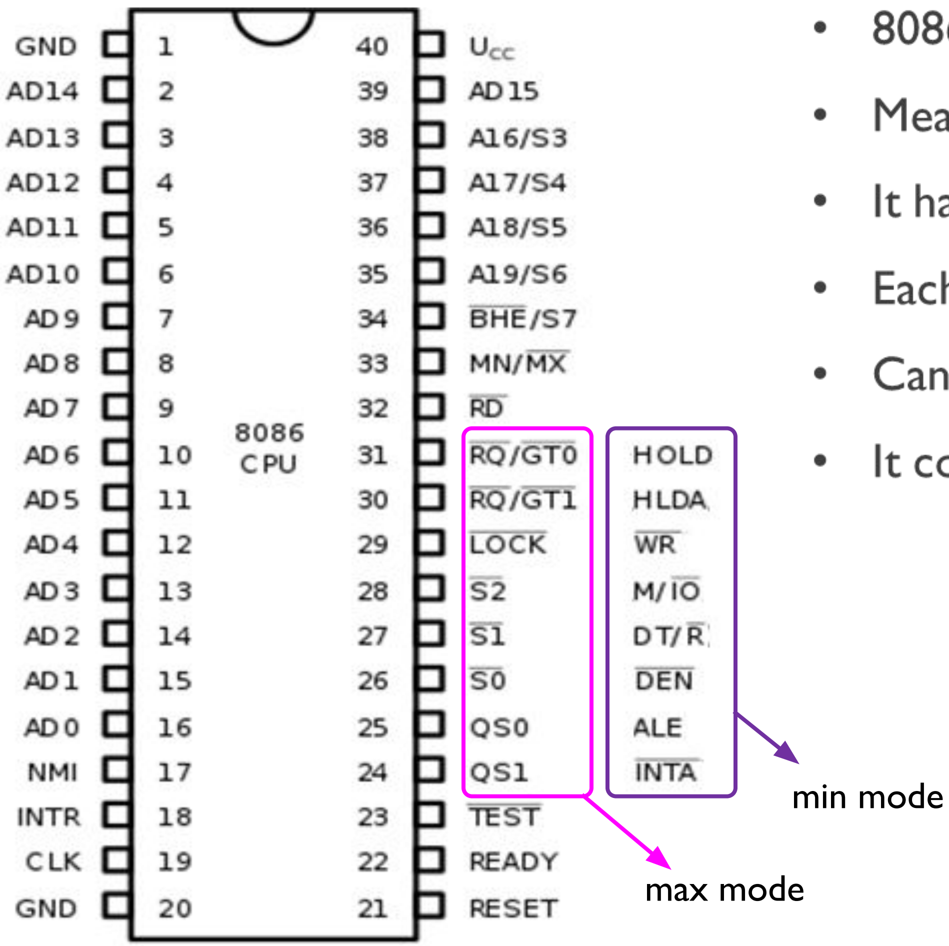
8086 Pin Specification

- is a 40-pin DIPs; **Dual in-line package**
- DIP refers to a rectangular housing with two parallel rows of electrical connection pins.
- DIPs have a notch on one end to show its correct orientation.
- The pins are then numbered as shown in the figure below.





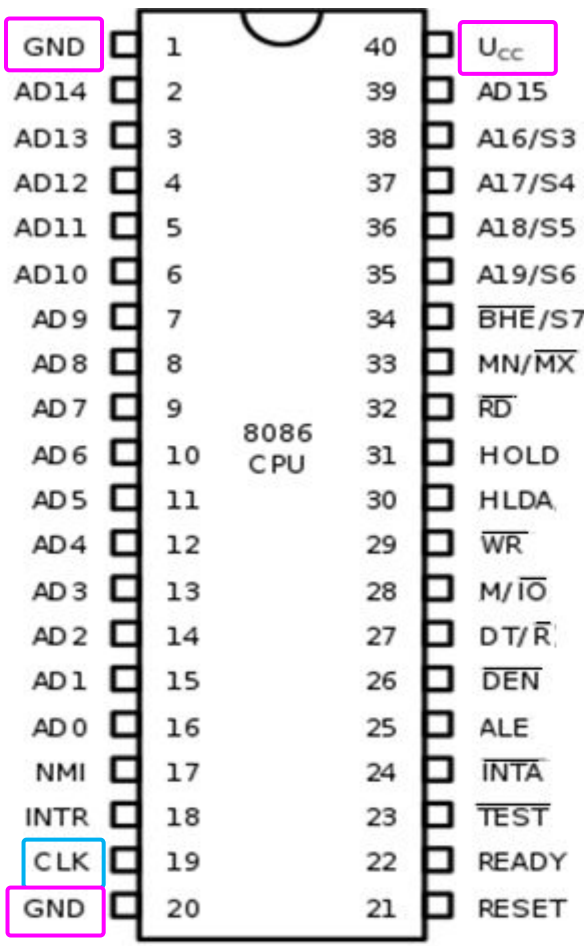
8086 Pin Diagram



Recap

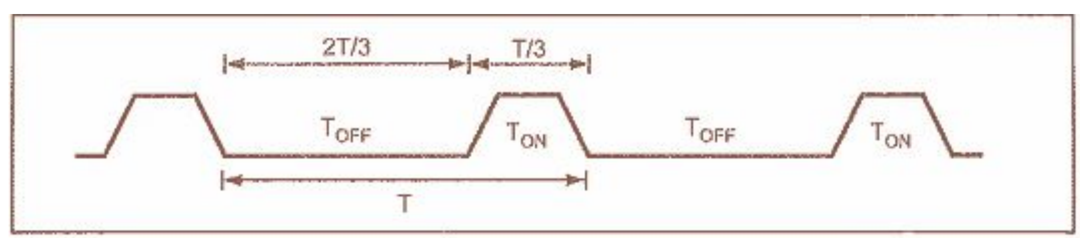
- 8086 is a 16-bit microprocessor.
- Meaning it has 16 data lines/bus.
- It has 20 address lines/bus.
- Each memory location holds only 1 byte
- Can address 2^{20} different memory locations
- It could address up to 1MB of memory.

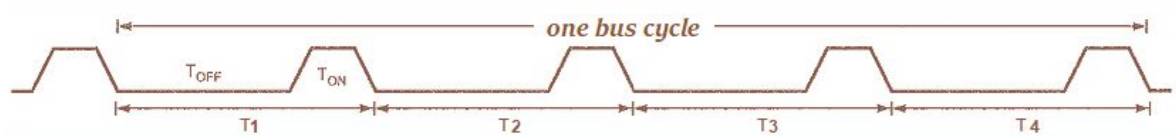
8086 Pin Specification



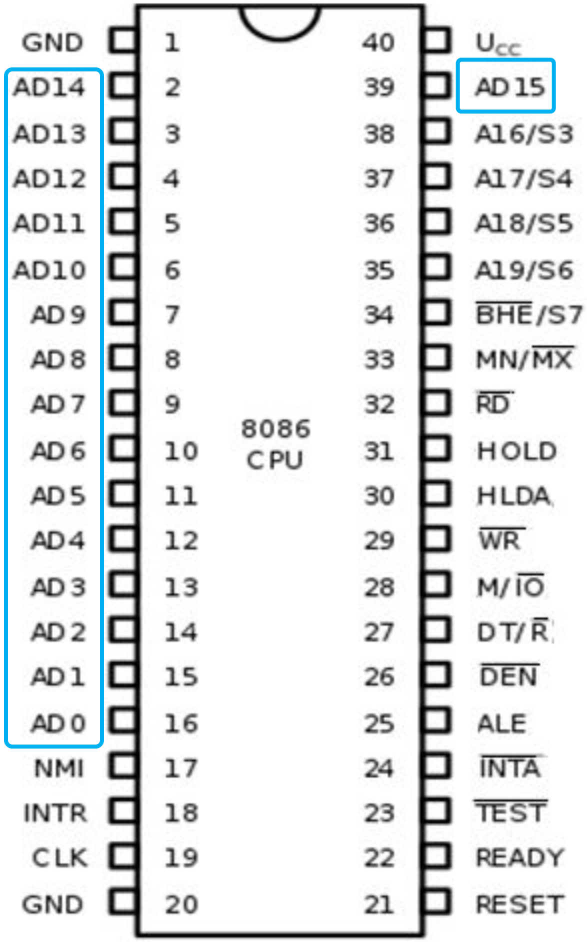
CLK, input

- provides basic timing to control processor operation
- frequencies of different versions are 5, 8 or 10 MHz
- asymmetric with a 33% duty cycle





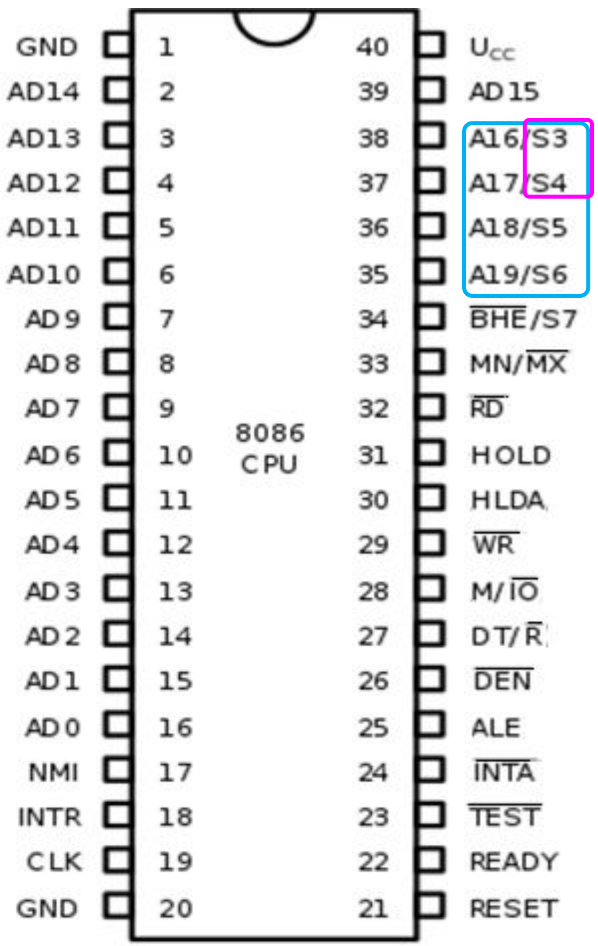
8086 Pin Specification



$$AD_0 - AD_{15}, \boldsymbol{bi-directional}$$

- lines are multiplexed bidirectional address/data bus.
- During T_1, they carry 16-bit address.
- In remaining clock cycles T_2, T_3 *and* T_4, 16-bit data.
- $AD_0 - AD_7$ carry lower order data byte
- $AD_8 - AD_{15}$ carry higher order data byte

8086 Pin Specification

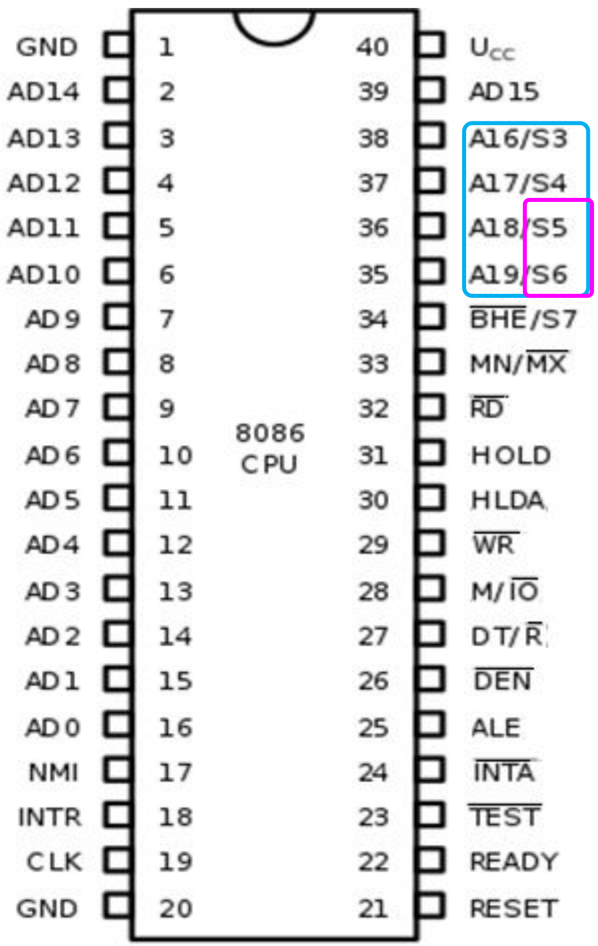


$$A_{19}/S_6, A_{18}/S_5, A_{17}/S_4, A_{16}/S_3, \textbf{\textit{output}}$$

- lines are multiplexed address and status bus.
- During T_1, they carry the highest order 4-bit address.
- During T_2, T_3 *and* T_4, status signals.
- S_3 *and* S_4, segment identifiers as in table below

S4	***S3***	***Function***
0	*0*	*Extra segment access*
0	*1*	*Stack segment access*
1	*0*	*Code segment access*
1	*1*	*Data segment access*

8086 Pin Specification



$$A_{19}/\boldsymbol{S_6}, A_{18}/\boldsymbol{S_5}, A_{17}/S_4, A_{16}/S_3, \boldsymbol{output}$$

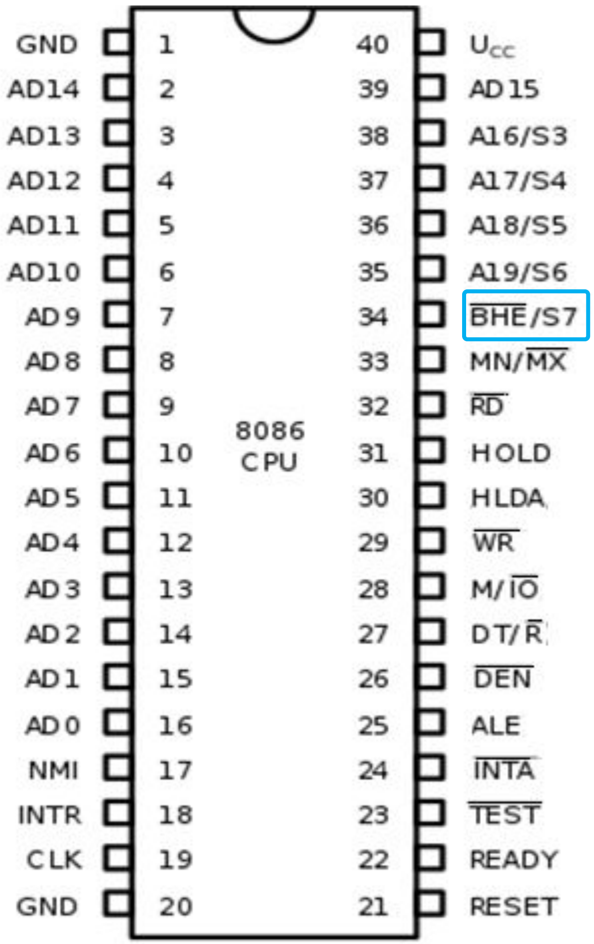
S_5 : Indicates if interrupt is enabled or disabled.

- If S_5 =1 , then the IF = 1, so the interrupt is enabled.
- If S_5 =0 , then the IF = 0, so the interrupt is disabled.

S_6 : Indicates if 8086 is the bus master or not

- If S_6 = 0 , 8086 is the bus master
- If S_6 = 1 , 8086 is not the bus master

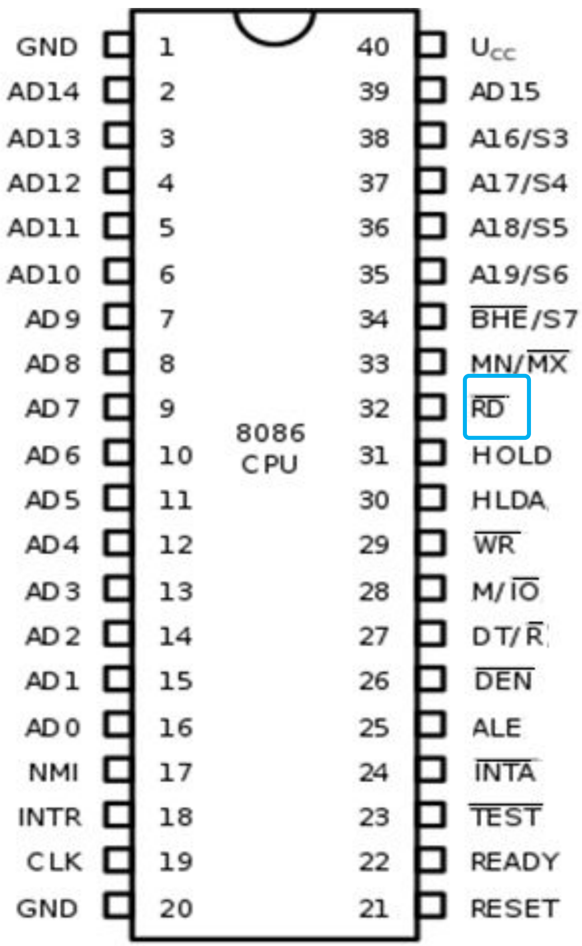
8086 Pin Specification



$\overline{BHE}/S_7$, ***output***

- Bus High Enable
- $\overline{BHE}$ is active low
- To indicate the transfer of data over $AD_8 - AD_{15}$
- Related to memory bank
- Selects odd/high memory bank when $\overline{BHE}$ is 0
- S_7 : Reserved for further development

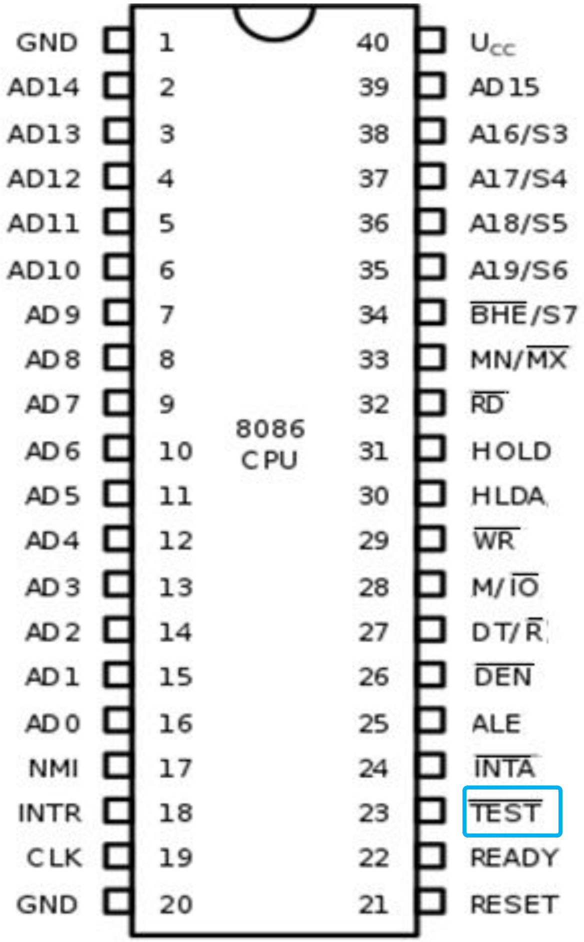
8086 Pin Specification



$\overline{RD}$, ***output***

- is active low
- Indicates read operation when low
- Processor reading from memory or I/O device
- Is low during T_2, T_3 *and* T_w states of the read cycle

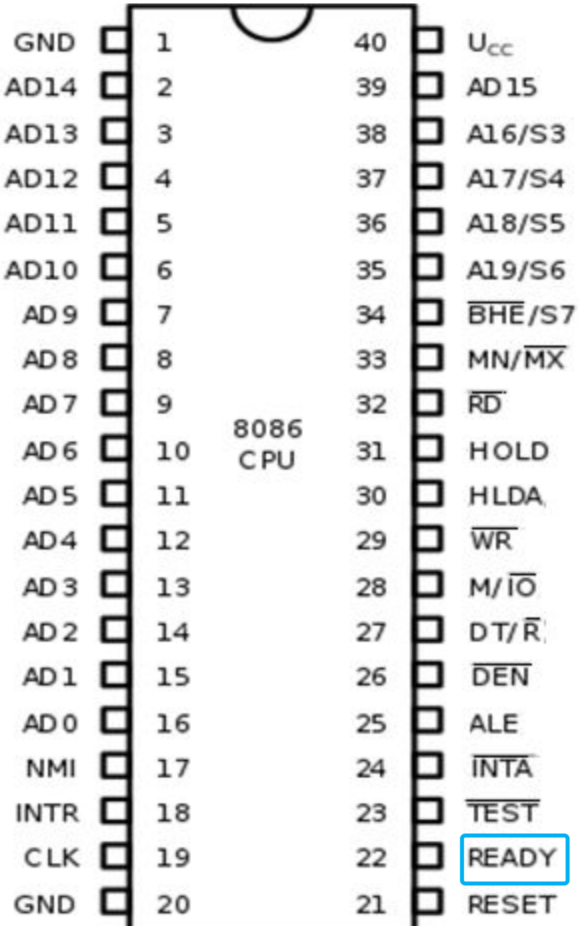
8086 Pin Specification



$\overline{TEST}$, ***input***

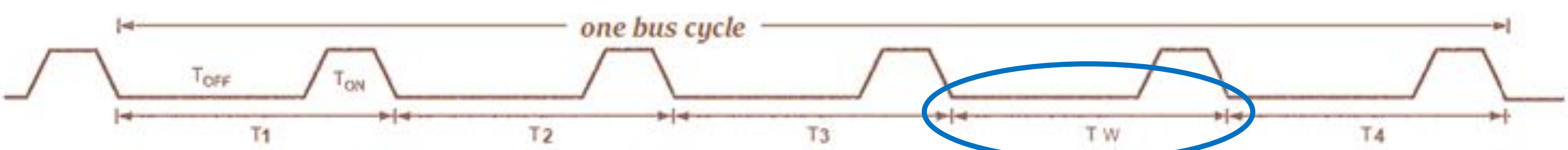
- Is examined by the WAIT instruction.
- If this pin is Low, execution continues.
- Else the processor waits in an idle state.

8086 Pin Specification

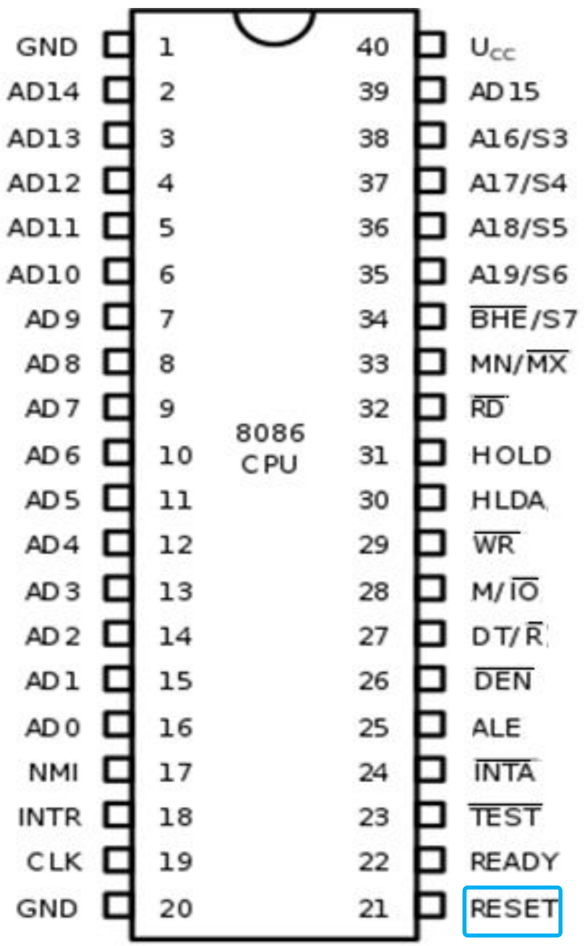


READY, ***input***

- acknowledgement from a slow I/O device or memory
- To indicate ready/completion of data transfer
- When low, microprocessor enters wait state, T_W.



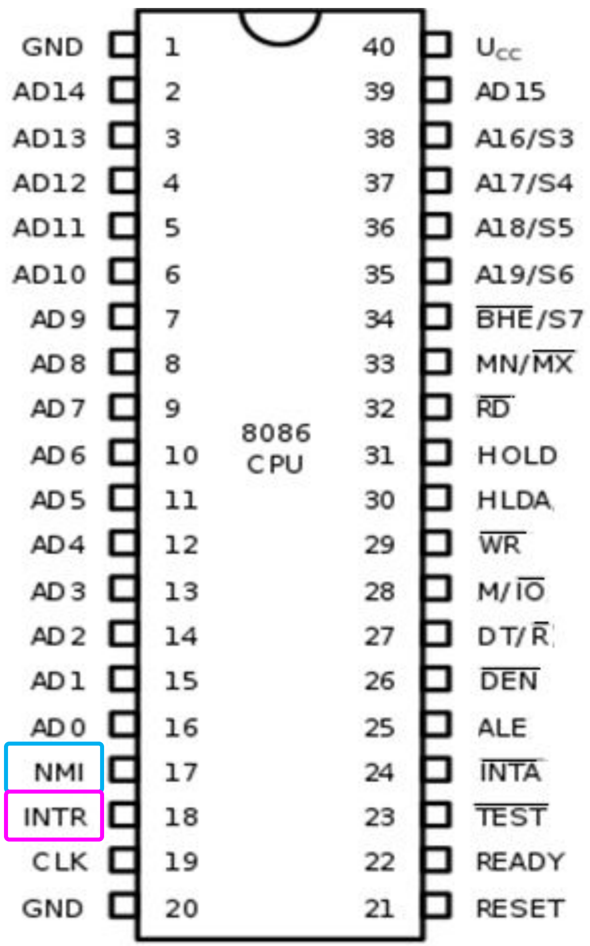
8086 Pin Specification



RESET, ***input***

- To reset the system reset.
- And terminates the current activity.
- Must be active for at least four clock cycles

8086 Pin Specification



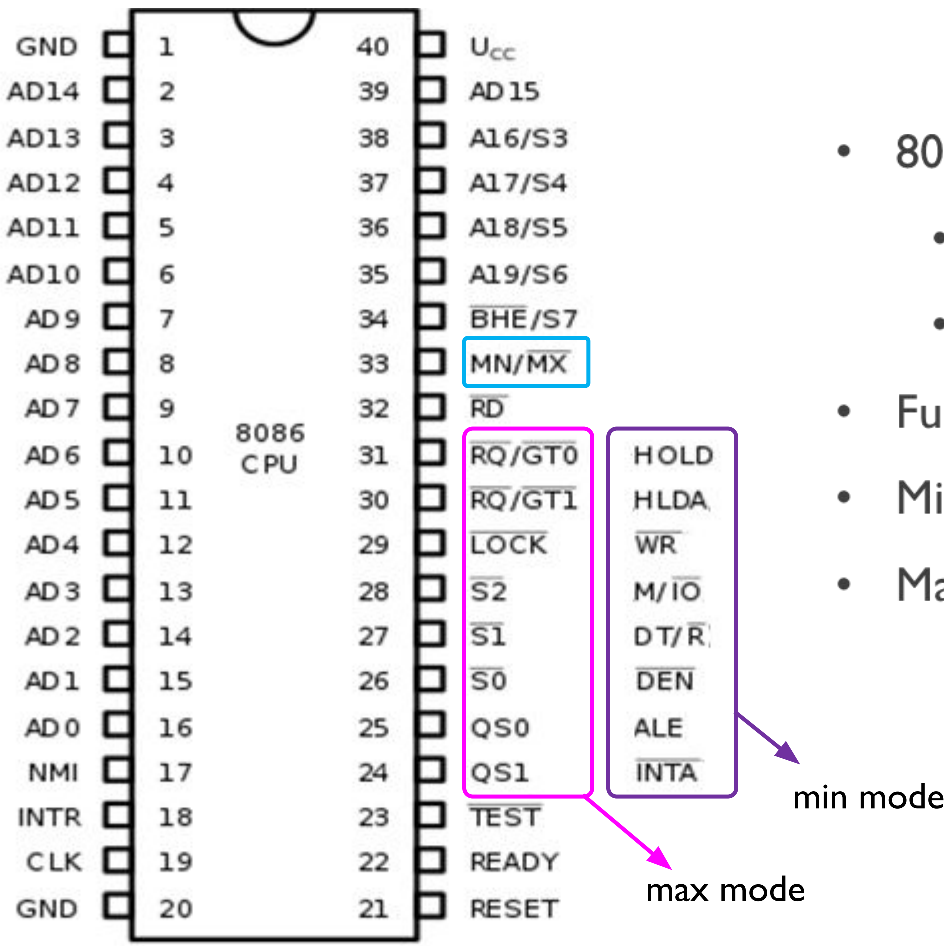
*INTR, **input***

- Interrupt request
- Used to request a hardware interrupt.
- Can be masked.

*NMI, **input***

- Non-maskable interrupt signal.
- Causes a type-2 interrupt.
- Initiates the interrupt at the end of the current instruction.

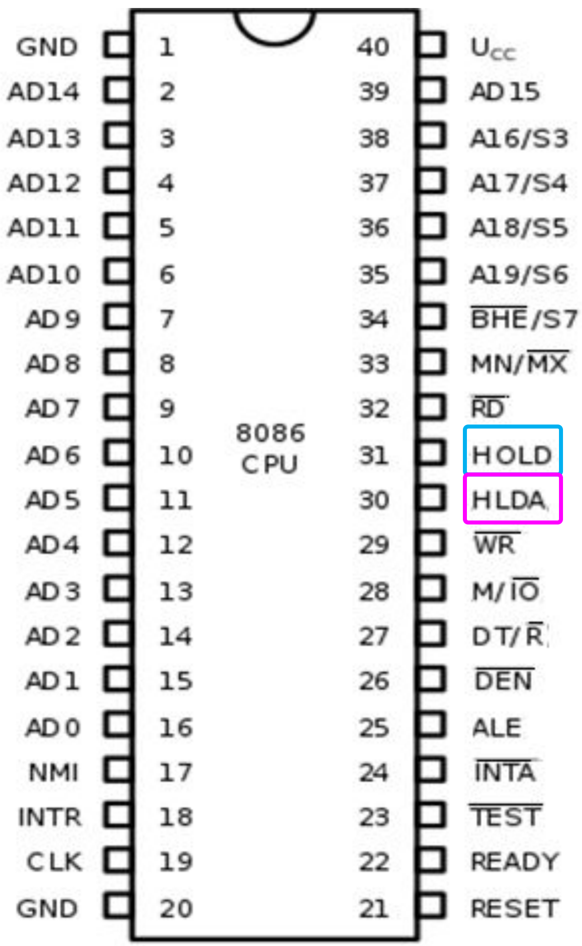
8086 Pin Specification



$MN/\overline{MX}$, ***input***

- 8086 works in two modes:
  - Minimum Mode if high
  - Maximum Mode if low
- Functions of pins 24-31 depend on the mode
- Minimum Mode - single processor
- Maximum Mode - multi processor

Minimum Mode Pin Specification



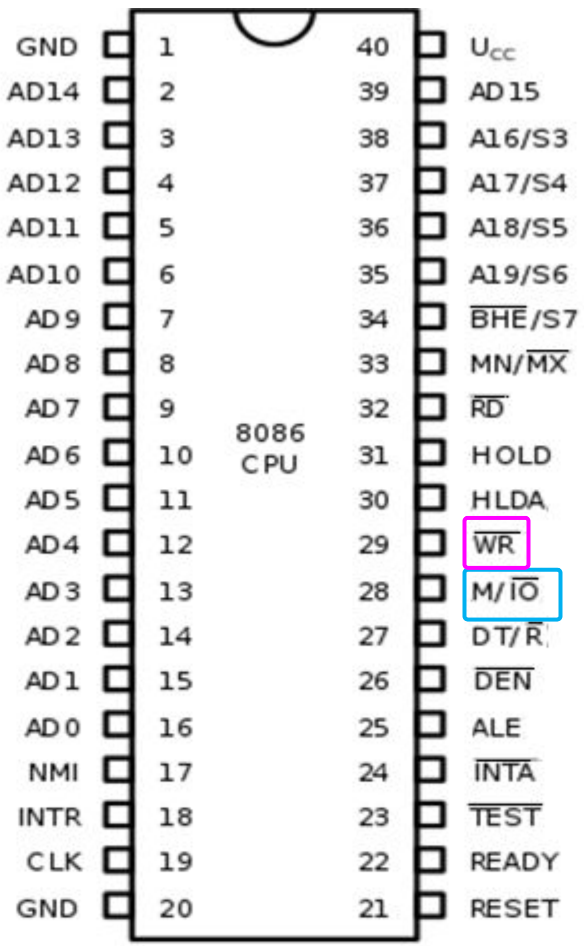
HOLD, ***input***

- To request for bus by another device.
- It is an active HIGH signal.

HLDA, ***output***

- Hold Acknowledgment.
- When acknowledged, it relinquish the bus to the requesting device

Minimum Mode Pin Specification



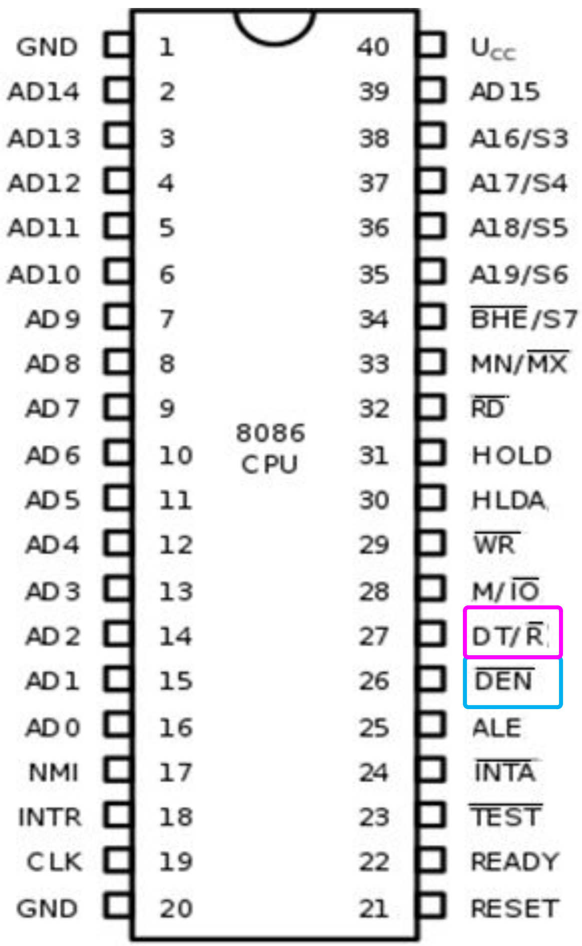
$\overline{WR}$, ***output***

- Active low write signal.
- Writes data to memory or output device depending on $M/\overline{IO}$ signal.

$M/\overline{IO}$, ***output***

- Differentiates memory access from I/O access.
- When high, memory is accessed.
- When low, I/O devices are accessed.

Minimum Mode Pin Specification



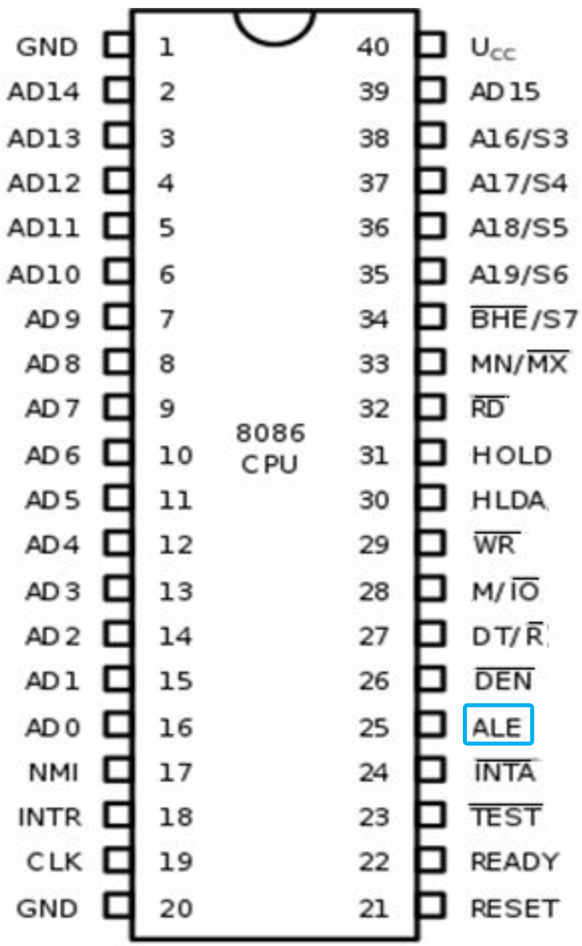
$$DT/\bar{R}, \boldsymbol{output}$$

- Data Transmit/Receive signal.
- indicates the direction of flow *through the transceiver*.
- When high, data is transmitted out i.e. written to.
- When low, data is received in i.e. read in.

$$\overline{DEN}, \boldsymbol{output}$$

- Data Enable signal.
- Used to enable a transceiver connected to the µP

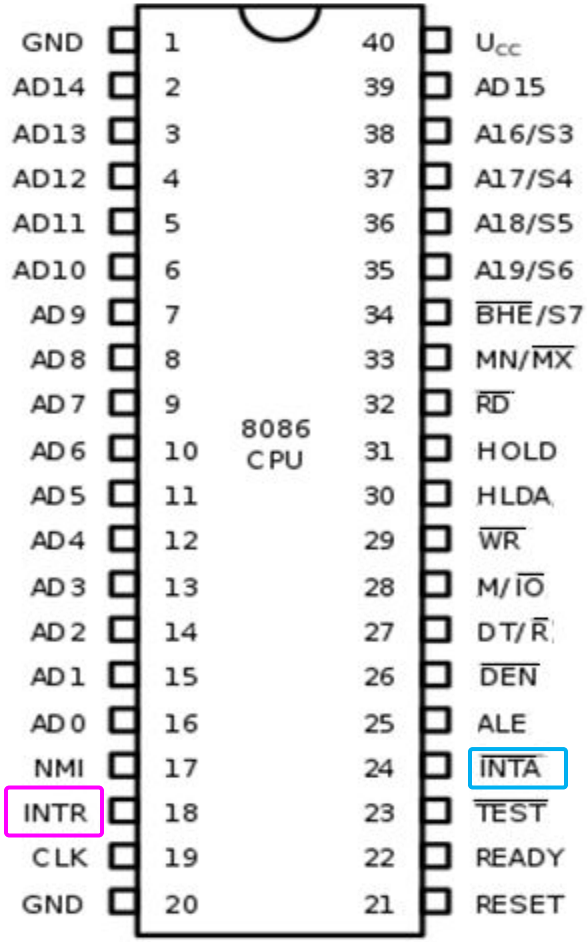
Minimum Mode Pin Specification



ALE, $\boldsymbol{output}$

- Address Latch Enable
- indicates an address is available on bus $AD_0 - AD_{15}$.
- active high during T_1 state

Minimum Mode Pin Specification



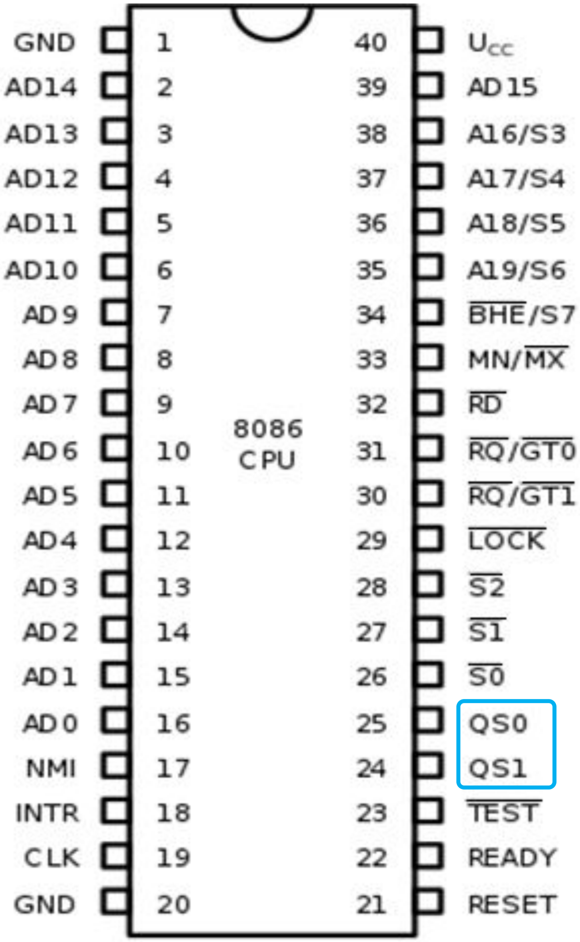
$\overline{INTA}$**, *output***

- An active low signal.
- An interrupt acknowledge signal.
- When microprocessor receives an INTR signal, it acknowledges the interrupt by generating this signal
- When low it indicates an interrupt is being serviced.

8086 Maximum Mode Pins

Dept. of Computer Science and Engineering
BRAC University
CSE 341 Team

Maximum Mode Pin Specification

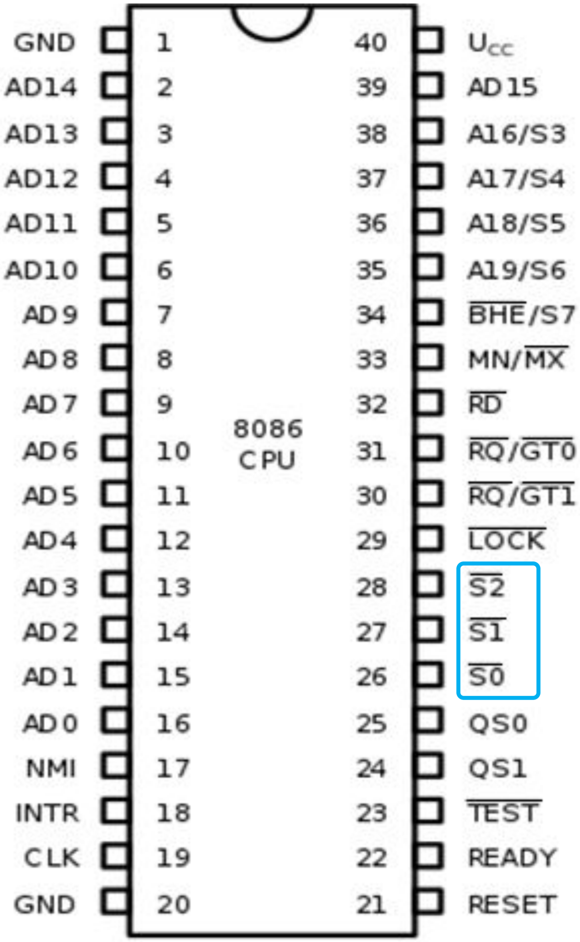


QS_0 *and* QS_1, ***output***

- Instruction queue status.
- Instruction queue is 6 bytes long

		Function
0	*0*	*No Operation. During the last clock cycle, nothing was taken from the queue.*
0	*1*	*First Byte. The byte taken from the queue was the first byte of the instruction.*
1	*0*	*Queue Empty. The queue has been reinitialized as a result of the execution of a transfer instruction.*
1	*1*	*Fetch subsequent byteSubsequent Byte. The byte taken from the queue was a subsequent byte of the instruction.*

Maximum Mode Pin Specification

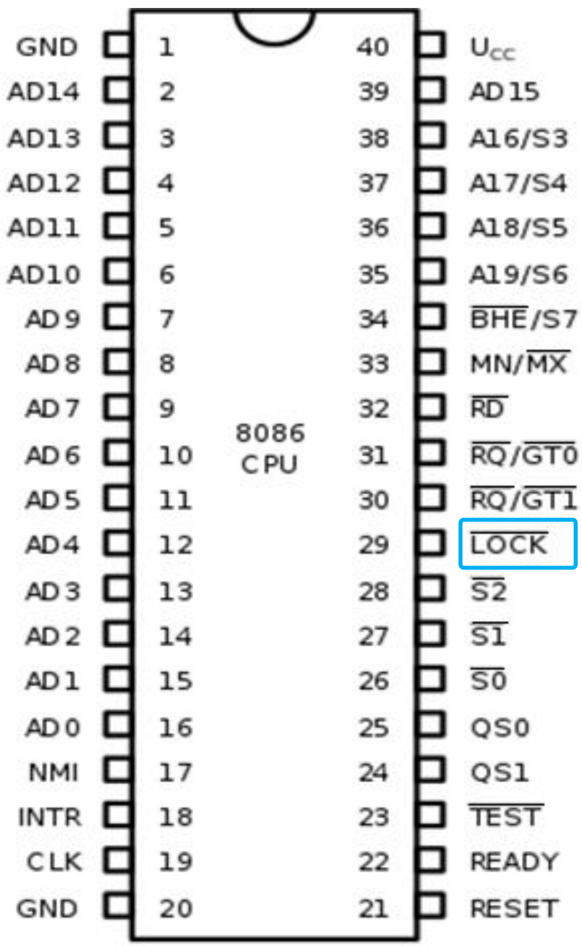


$\overline{S_0}, \overline{S_1}, \overline{S_2}$, ***output***

- Status Signals.
- indicate operation done by the microprocessor
- Related to memory and I/O access control signals.

			Function
0	0	*0*	*Interrupt acknowledgement*
0	0	*1*	*Read data from I/O port*
0	*1*	*0*	*Write data from I/O port*
0	*1*	*1*	*Halt*
1	0	*0*	*Opcode fetch*
1	0	*1*	*Memory read*
1	*1*	*0*	*Memory write*
1	*1*	*1*	*Passive state*

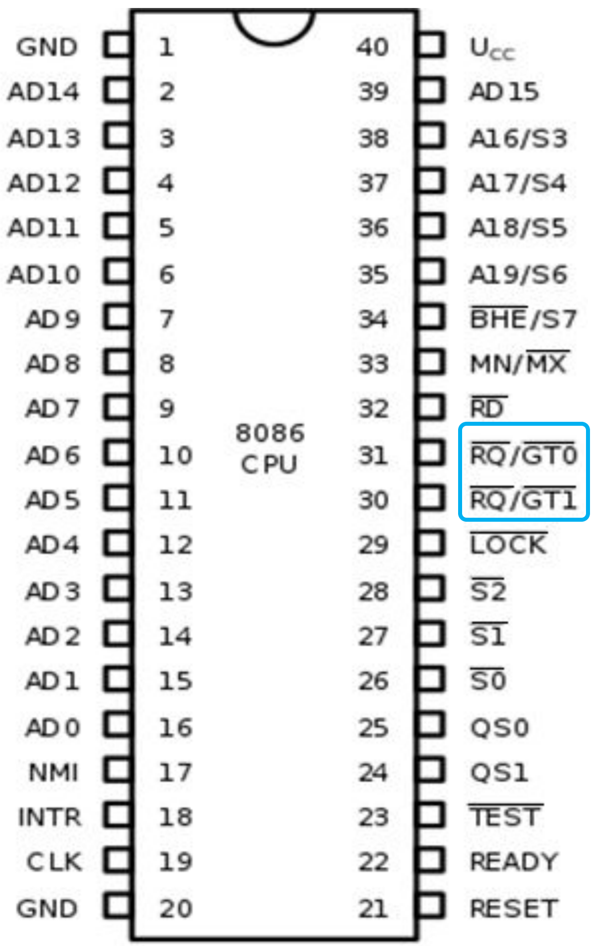
Maximum Mode Pin Specification



$\overline{LOCK}$, ***output***

- When low, all interrupts are masked
- Indicates to other processors to not request for system bus.
- No HOLD request is granted.
- No bus is relinquished to the other processors

Maximum Mode Pin Specification



$\overline{RQ/GT_0}$ **and** $\overline{RQ/GT_1}$, ***bi – directional***

- Request/Grant pins.
- Other processors request the CPU through these for system bus.
- CPU sends acknowledge signal on the same lines.
- $\overline{RQ/GT_0}$ has higher priority than $\overline{RQ/GT_1}$.

QUIZ

1) Assuming you want to type a secret message using a keypad connected to an 8086 microprocessor, deduce the values of the following pins during that time. Justify your anwers too.

   a) $\overline{RD}$ e.g. mention if low (0) / high (1) and why.

   b) $\overline{WR}$

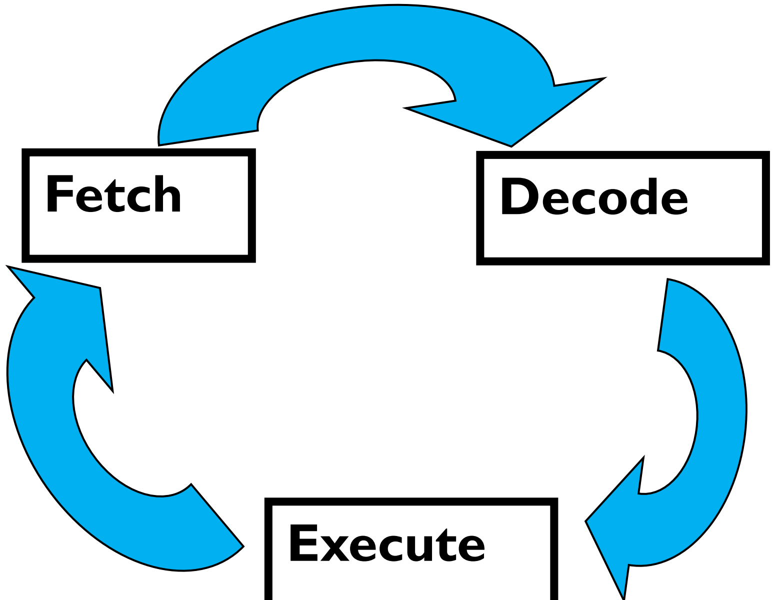
   c) $M/\overline{IO}$

2) Do you think there may be other pins involved? If so, justify your answer.

8086 Clocks & Timing Diagrams

Dept. of Computer Science and Engineering
BRAC University
CSE 341 Team

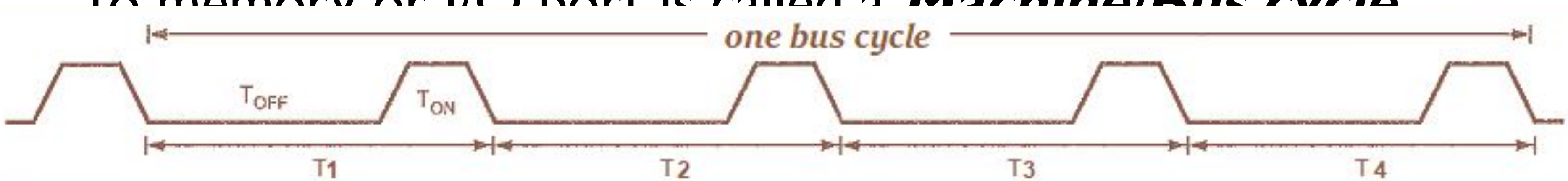
Microprocessor Operation

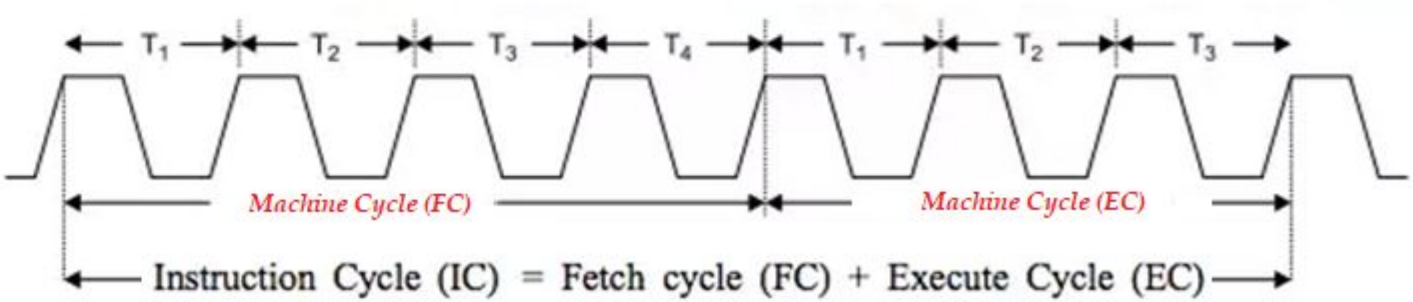


- An instruction e.g. MOV [7531h], AX ; SUB CH, [0ABCh] etc
- The time a μP requires to complete **fetch-(decode)-execute** operation of a single instruction is known as ***Instruction Cycle***

Microprocessor Operation

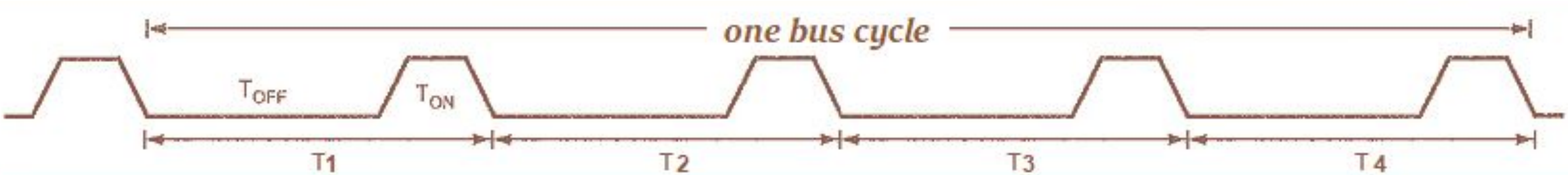
- ***Instruction Cycle*** consists of one or more ***Machine Cycles***
- A basic µP operation such as reading/writing a byte from or to memory or I/O port is called a ***Machine/Bus cycle***





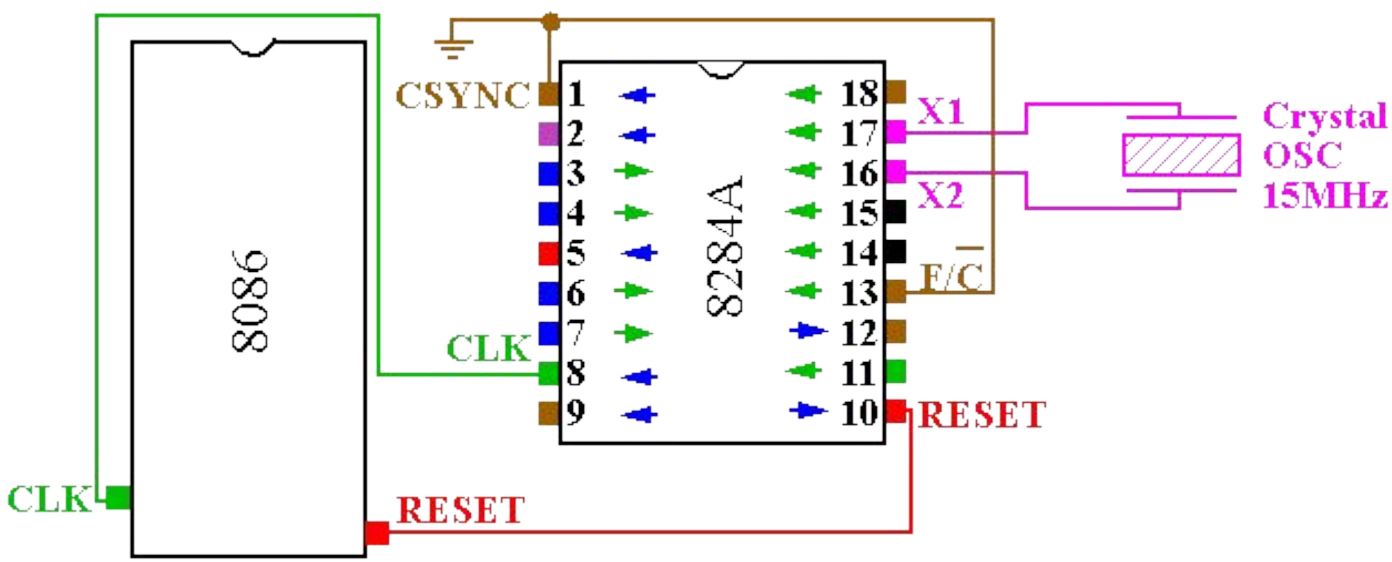
Microprocessor Operation

- A ***Machine (bus) cycle*** consists of at least **four** clock cycles, called **T** states.
- One cycle of a clock is called a **State**
- Each read or write operation takes 1 bus cycle.



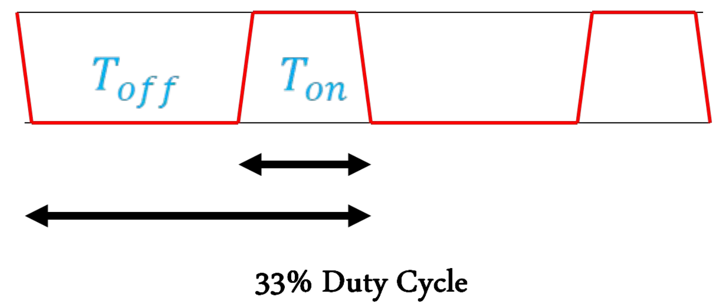
Clock Generation

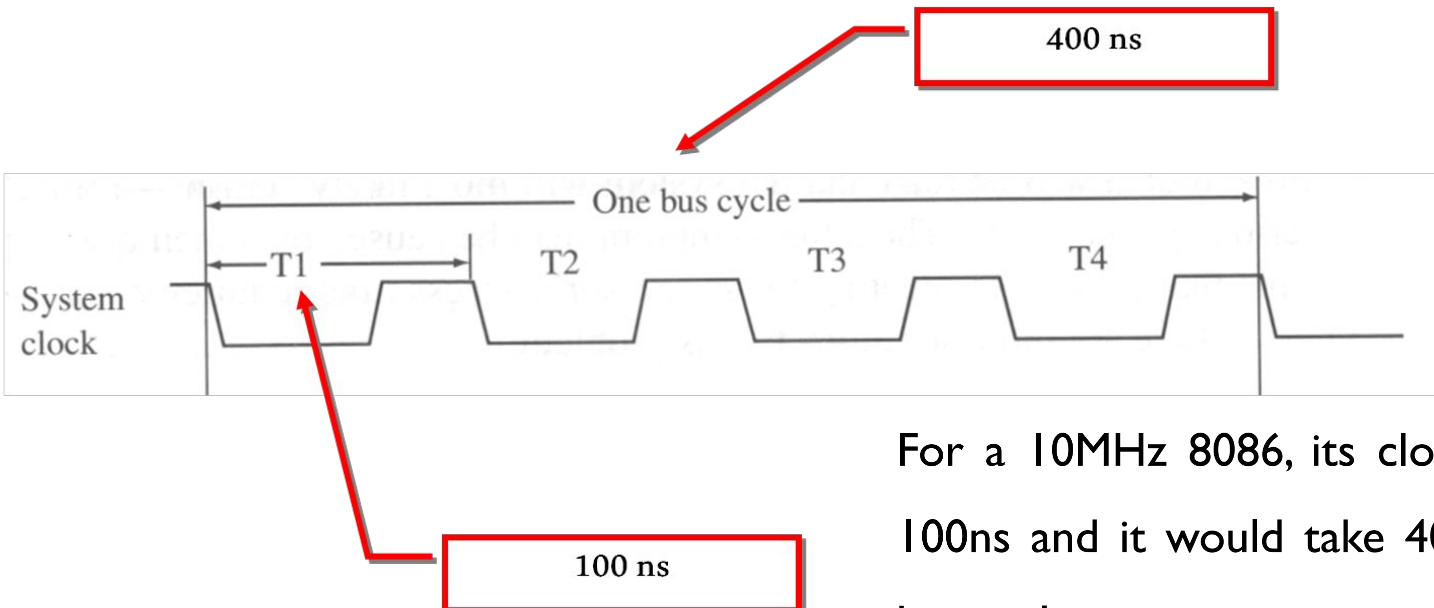
- Clock generator circuit is 8284A and connected to **pin 19** (**CLK**) of 8086.



System Clock Concept

- 8086 is found to operate in between 5 to 10 Mhz.
- An 8086 running at 5MHz, its clock pulses will be of 200ns and it would take 800ns for a complete bus cycle.





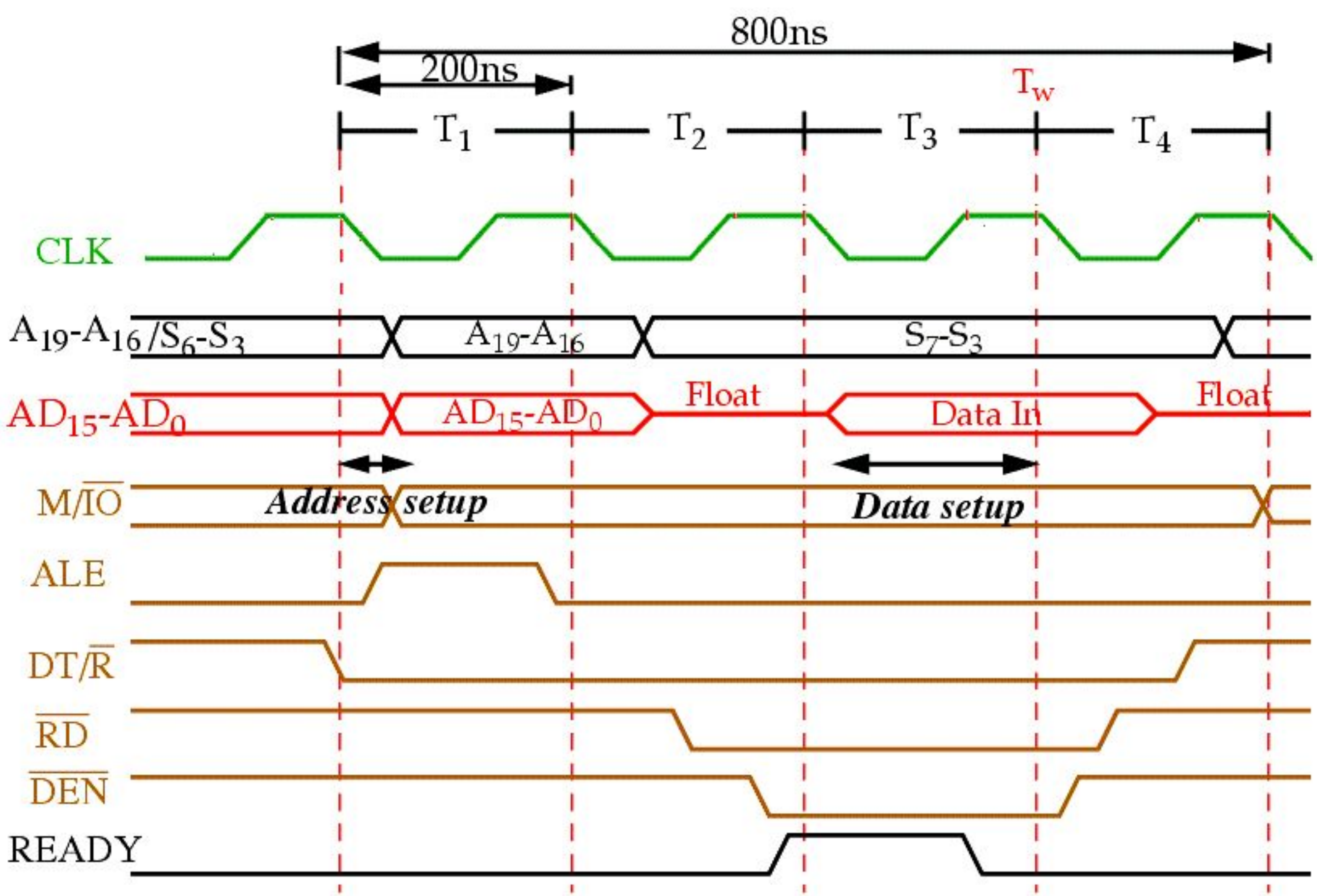
For a 10MHz 8086, its clock pulses will be of 100ns and it would take 400ns for a complete bus cycle.

Clock States - Why are there T states?

- In 8086, address and data lines are multiplexed to reduce number of pins e.g. AD_{0-15} else 32 pins would have been needed instead of 16
- The µp needs time to change the signals during each bus cycle.
- Memory devices need time to interpret the address value and then **read/write** the data (*access time*)
- A specific defined action occurs during each T state (***T_1* - *T_4***)
  - ***T_1*** : Address is output
  - ***T_2*** : Bus cycle type (Mem/IO, read/write)
  - ***T_3***: Data is supplied / Data is received
  - ***T_4*** : Data latched by CPU, control signals removed

READ BUS Timing (Complete BUS Cycle)

T_1: Address is output

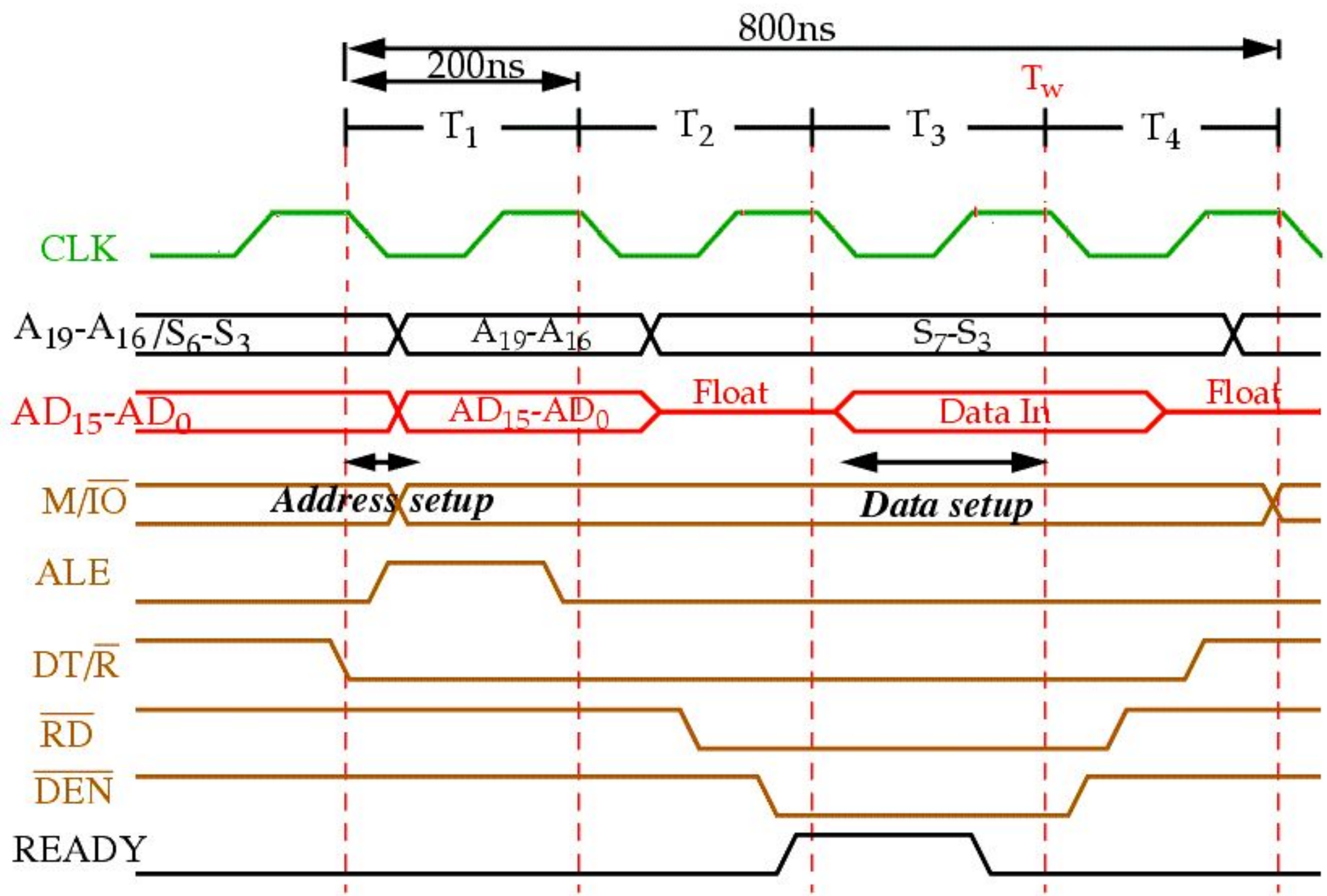


Address of memory is sent out by 8086 via address bus

Used Control signals: ALE, DT/R’, M/IO’ shows some output

READ BUS Timing (Complete BUS Cycle)

T_2*: *Bus cycle type (MEMORY/IO, READ/WRITE)

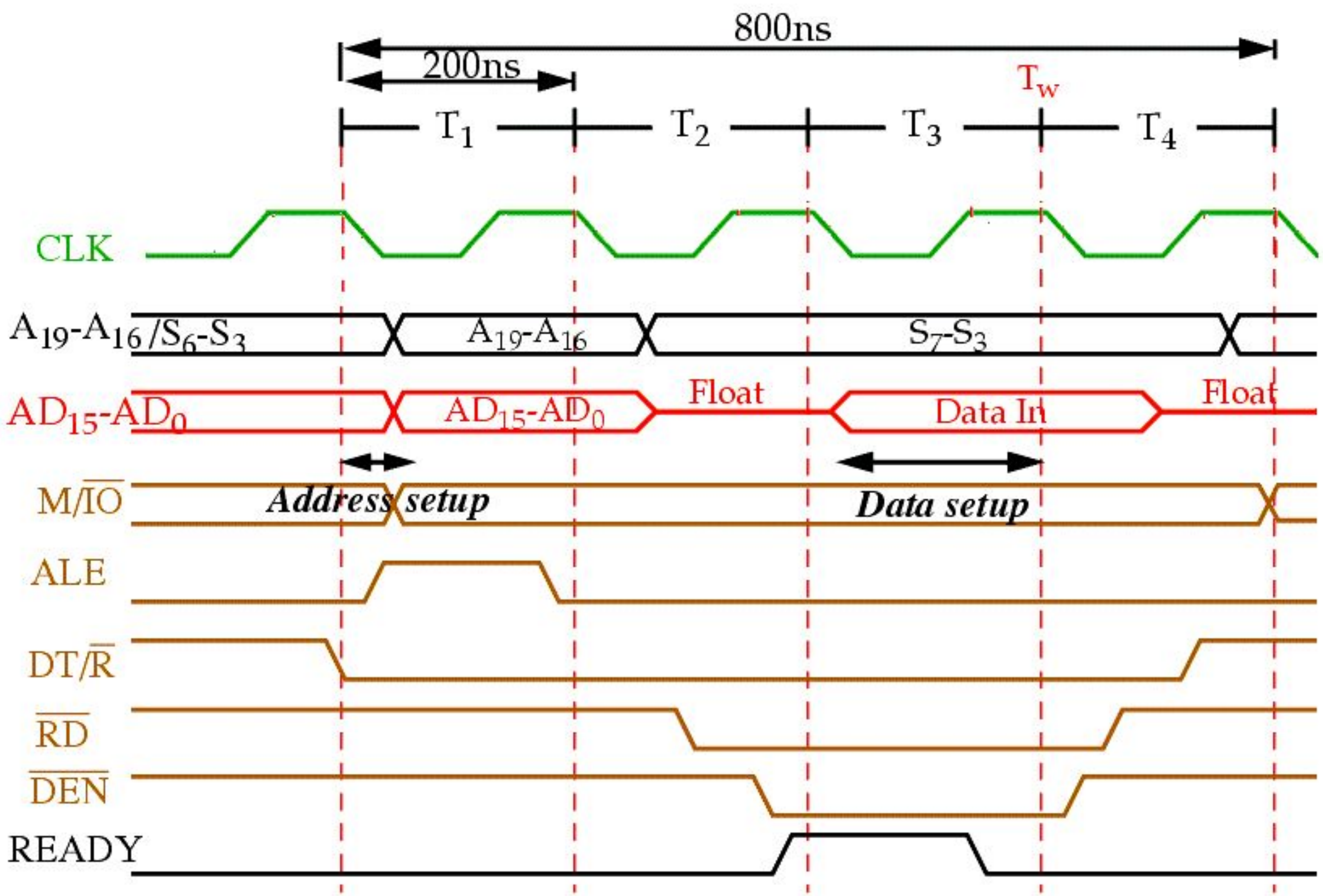


8086 issues either RD' or WR' and DEN'

In case of WRITE (WR) operation, data to be written appear on data bus

READ BUS Timing (Complete BUS Cycle)

T_3: Data is supplied



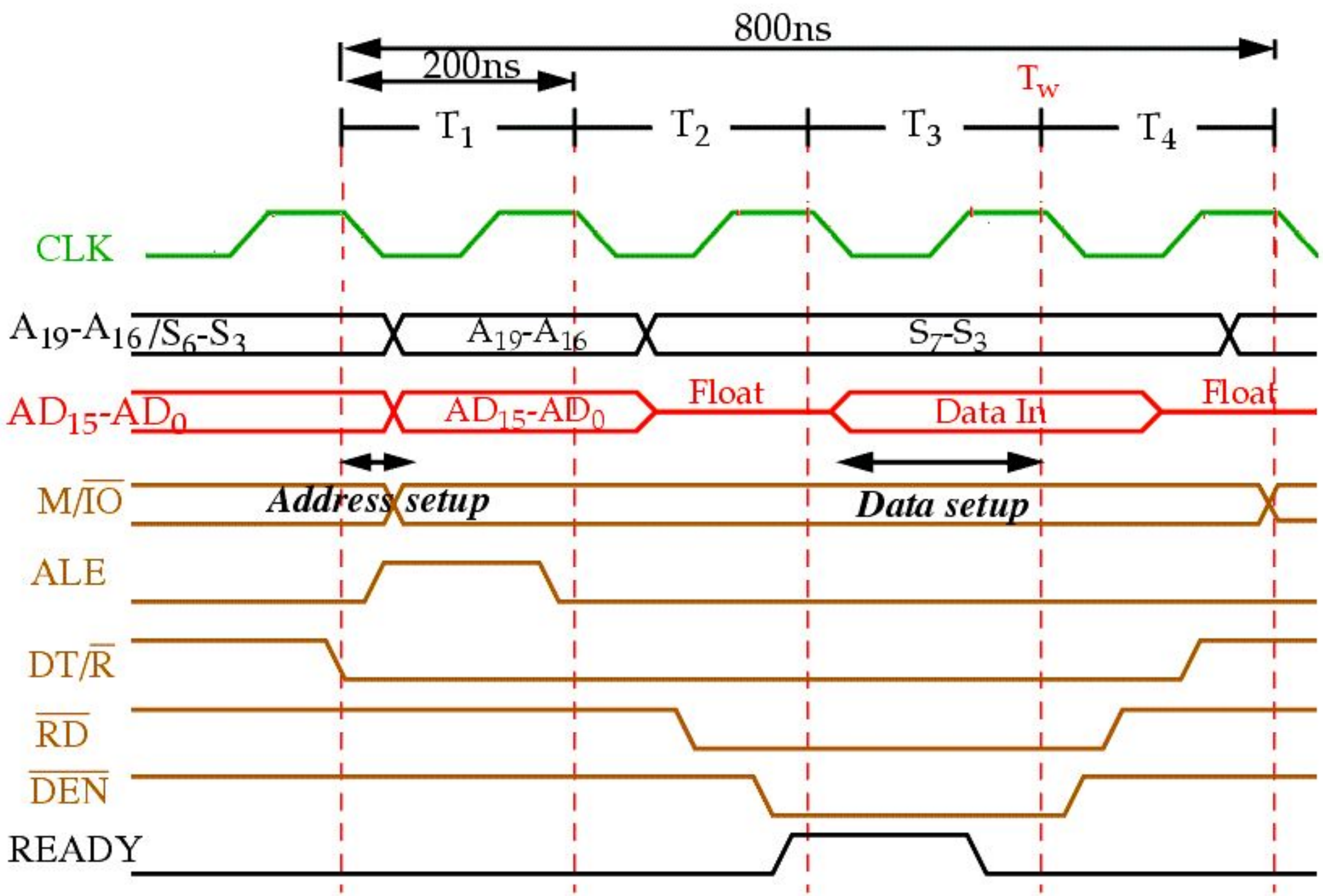
READY is sampled at the end of T2

If READY is low, T3 becomes a wait state (Tw), means no operation (NOP).

In READ bus cycle data bus is sampled at end of T3

READ BUS Timing (Complete BUS Cycle)

T_4: Data latched by μP, control signals removed



All bus signals deactivated in preparation for next bus cycle

μP sampled data bus for data that read from M or I/O

Clock States

- A specific, defined action occurs during each T states (T_1 – T_4)
- **T_1**: Address is output
  - Address of memory is sent out by 8086 via address bus
  - Used Control signals: ALE, DT/R', M/IO' shows some output
- **T_2**: Bus cycle type (MEMORY/IO, READ/WRITE)
  - 8086 issues either RD' or WR' and DEN'
  - In case of **WRITE (WR)** operation, data to be written appear on data bus

Clock States

- **T_3**: Data is supplied
  - READY is sampled at the end of T_2
    - If READY is low, T_3 becomes a wait state (T_w), means no operation (NOP).
    - In **READ** bus cycle data bus is sampled at end of T_3
- **T_4**: Data latched by μP, control signals removed
    - All bus signals deactivated in preparation for next bus cycle
    - μP sampled data bus for data that read from M or I/O
    - At trailing edge of WR', transfer data to M or I/O

8086 Ready pin

- The READY input is controlled to insert "Wait states" into the timing of the microprocessor for slower memory and I/O components..
- If the READY pin is at a logic 0 level, the micro-processor enters into wait states and remains idle.
- When it is high (logic 1), it indicates that the device is ready to transfer data.
- A wait state is a situation in which a computer processor is waiting for the completion of some event before resuming activity.
- A program or process in a wait state is inactive for the duration of the wait state.

Ready pin and Wait state

- When a computer processor works at a faster clock speed than the random access memory (RAM) that sends it instructions, it is set to go into a wait state for one or more clock cycles so that it is synchronized with RAM speed. In general, the more time a processor spends in wait states, the slower the performance of that processor.
- Wait states are a pure waste for a processor's performance. Modern designs try to eliminate or hide them using a variety of techniques: CPU caches, instruction pipelines, instruction prefetch, simultaneous multithreading and others.

Thank You

Questions are welcome in the discussion class