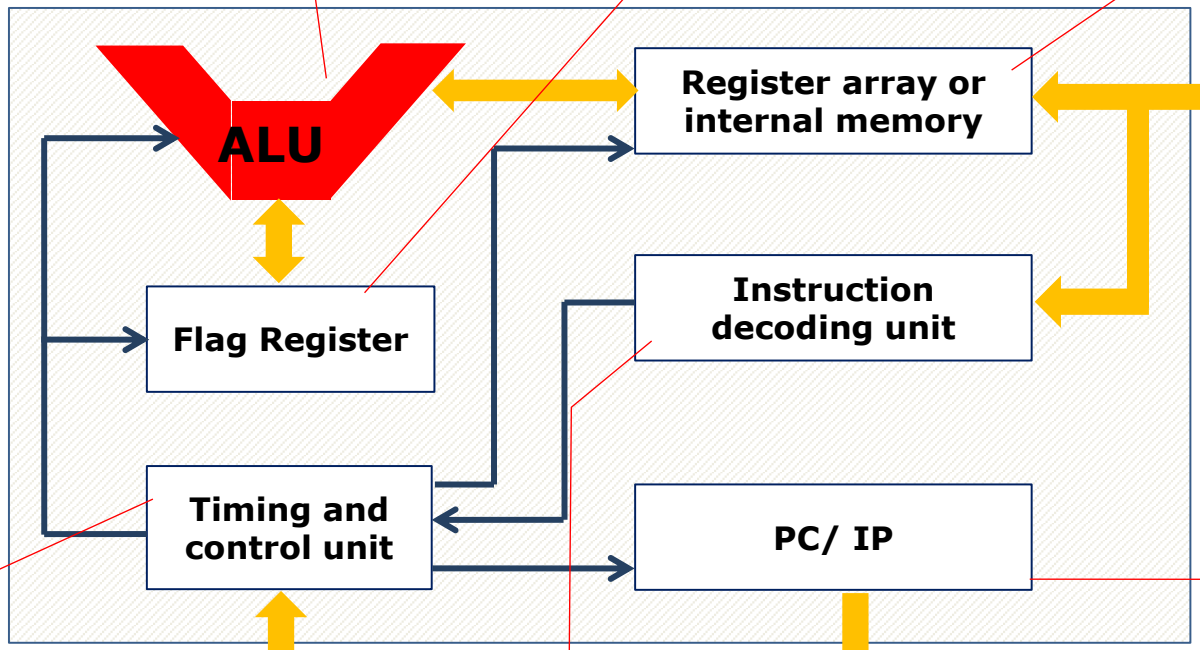
8086 Microprocessor

Functional blocks

Computational Unit; performs arithmetic and logic operations

Various conditions of the results are stored as status bits called flags in flag register

Internal storage of data



Generates the address of the instructions to be fetched from the memory and send through address bus to the memory

Generates control signals for internal and external operations of the microprocessor

Decodes instructions; sends information to the timing and control unit

Overview

First 16- bit processor released by INTEL in the year 1978

Originally HMOS, now manufactured using HMOS III technique

Approximately 29, 000 transistors, 40 pin DIP, 5V supply

Does not have internal clock; external asymmetric clock source with 33% duty cycle

20-bit address to access memory ⇒ can address up to 2^{20} = 1 megabytes of memory space.

Addressable memory space is organized in to two banks of 512 KB each; Even (or lower) bank and Odd (or higher) bank. Address line A_0 is used to select even bank and control signal is used to access odd bank

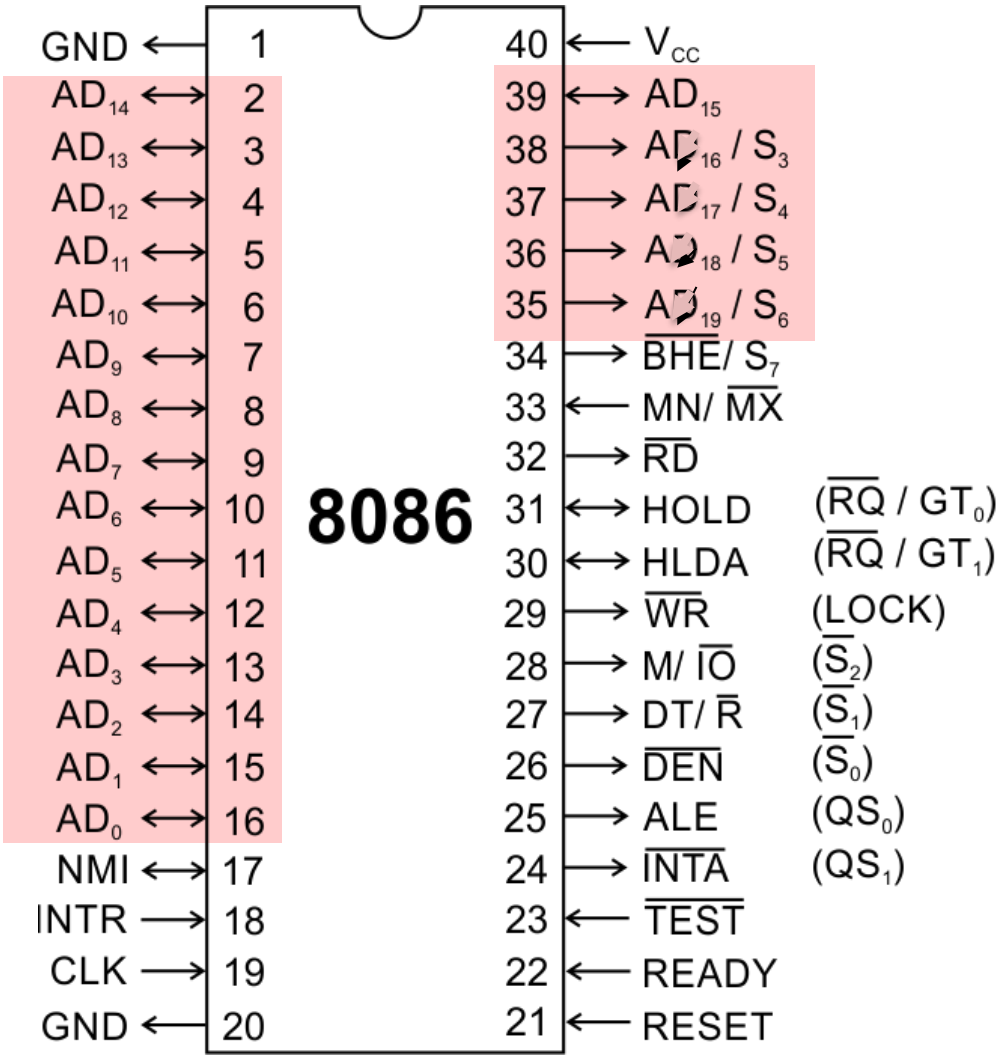
Uses a separate 16 bit address for I/O mapped devices ⇒ can generate 2^{16} = 64 k addresses.

Operates in two modes: minimum mode and maximum mode, decided by the signal at MN and pins.

Pins and signals

Pins and Signals

Common signals



AD_0-AD_{15} (Bidirectional)

Address/Data bus

Low order address bus; these are multiplexed with data.

When AD lines are used to transmit memory address the symbol A is used instead of AD, for example A_0-A_{15}.

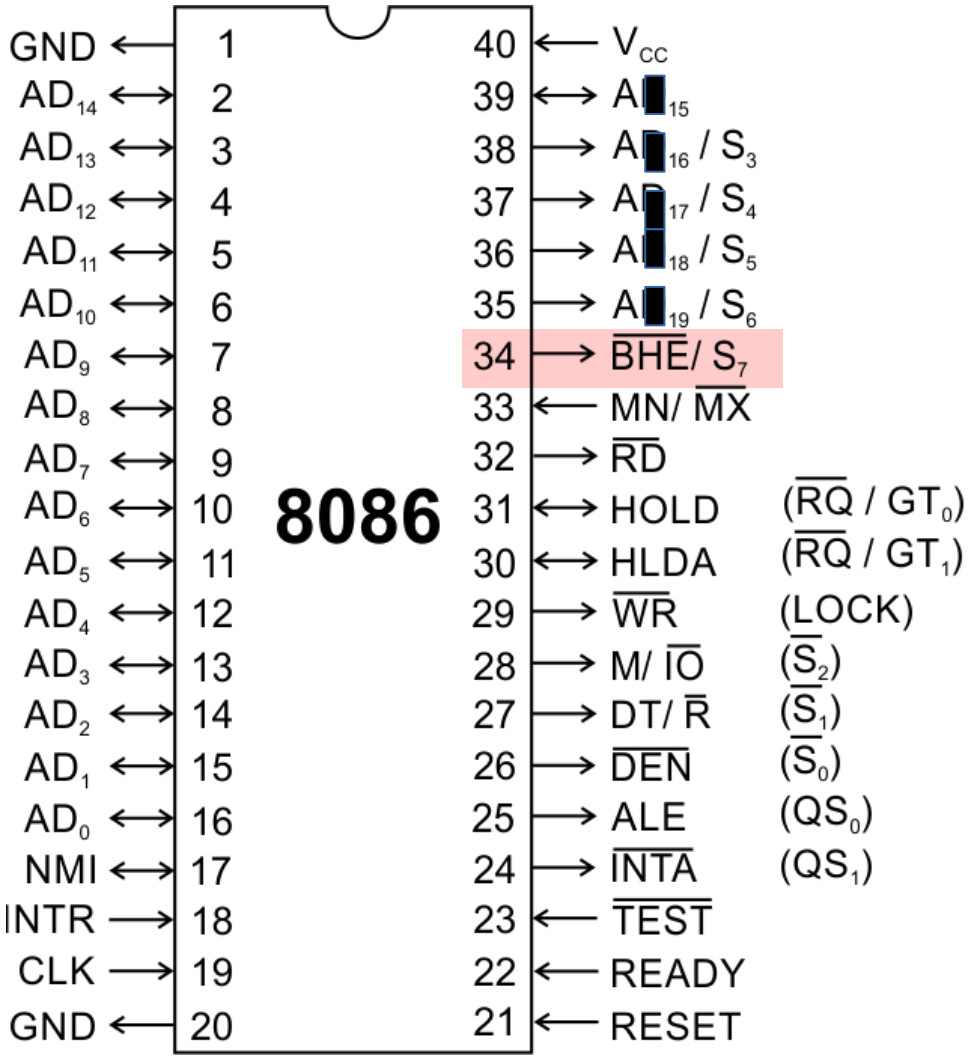
When data are transmitted over AD lines the symbol D is used in place of AD, for example D_0-D_7, D_8-D_{15} or D_0-D_{15}.

A_{16}/S_3, A_{17}/S_4, A_{18}/S_5, A_{19}/S_6

High order address bus. These are multiplexed with status signals

Pins and Signals

Common signals



BHE (Active Low)/S_7 (Output)

Bus High Enable/Status

It is used to enable data onto the most significant half of data bus, D_8-D_{15}. 8-bit device connected to upper half of the data bus use BHE (Active Low) signal. It is multiplexed with status signal S_7.

MN/ MX

MINIMUM / MAXIMUM

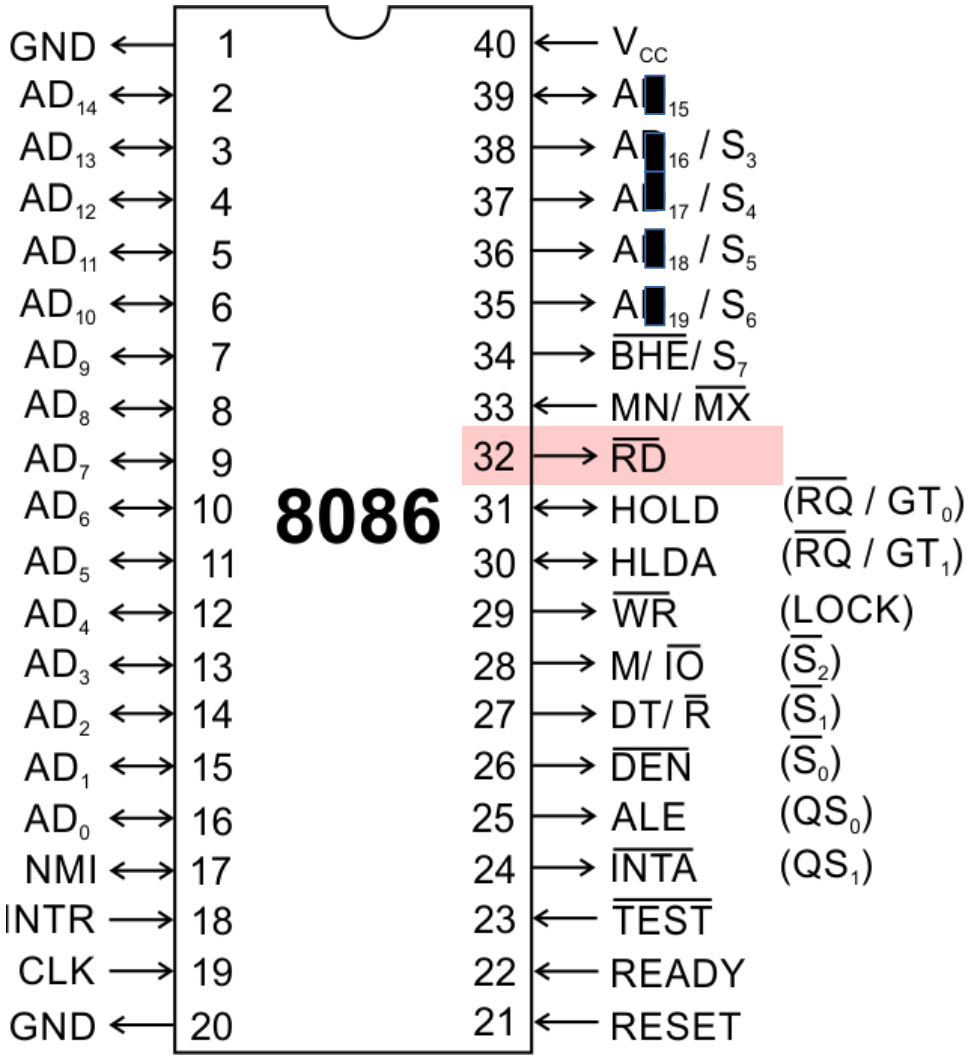
This pin signal indicates what mode the processor is to operate in.

RD (Read) (Active Low)

The signal is used for read operation.
It is an output signal.
It is active when low.

Pins and Signals

Common signals



TEST

input is tested by the 'WAIT' instruction.

8086 will enter a wait state after execution of the WAIT instruction and will resume execution only when the is made low by an active hardware.

This is used to synchronize an external activity to the processor internal operation.

READY

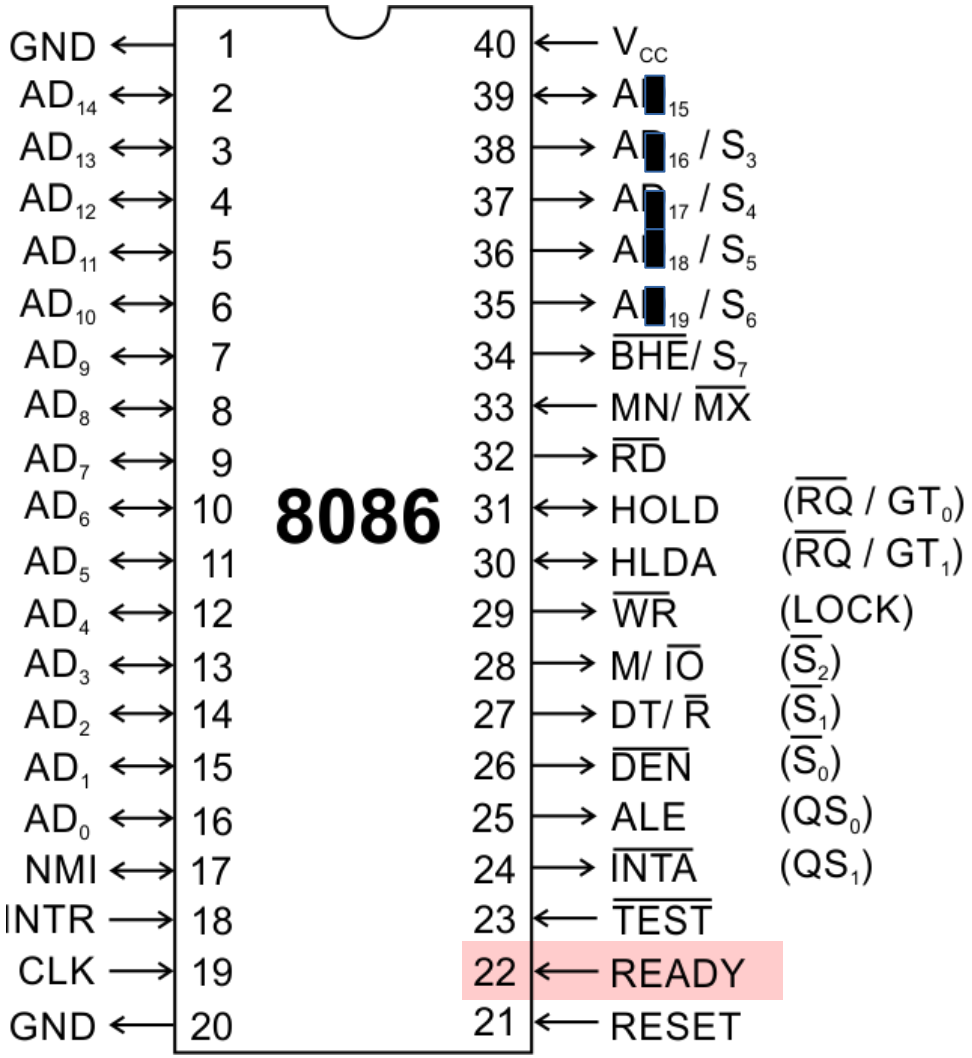
This is the acknowledgement from the slow device or memory that they have completed the data transfer.

The signal made available by the devices is synchronized by the 8284A clock generator to provide ready input to the 8086.

The signal is active high.

Pins and Signals

Common signals



RESET (Input)

Causes the processor to immediately terminate its present activity.

The signal must be active HIGH for at least four clock cycles.

CLK

The clock input provides the basic timing for processor operation and bus control activity. Its an asymmetric square wave with 33% duty cycle.

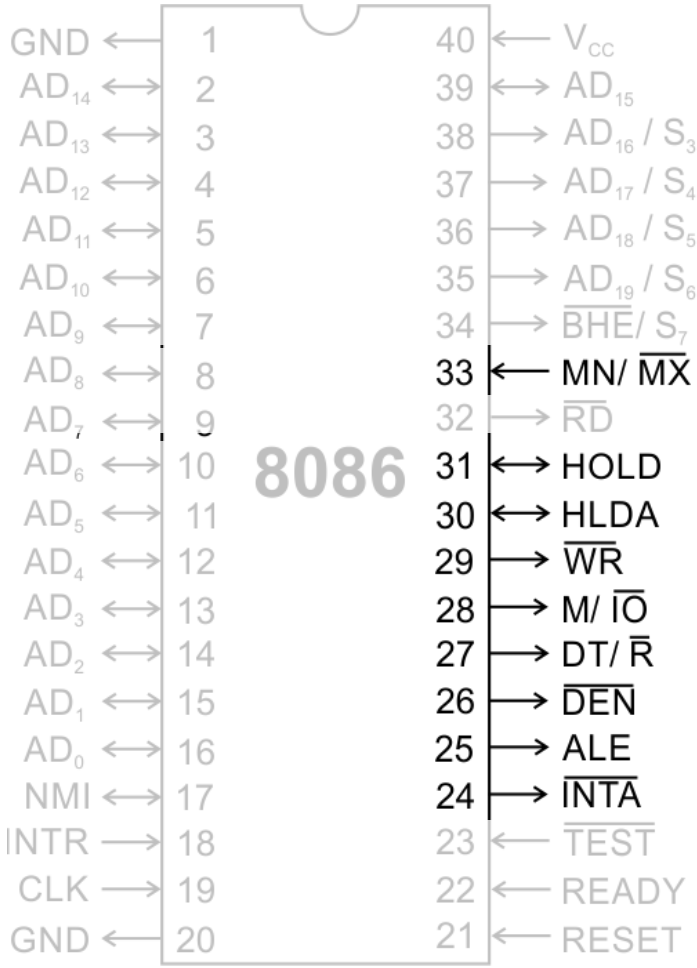
INTR Interrupt Request

This is a triggered input. This is sampled during the last clock cycles of each instruction to determine the availability of the request. If any interrupt request is pending, the processor enters the interrupt acknowledge cycle.

This signal is active high and internally synchronized.

Pins and Signals

Min/ Max Pins



The 8086 microprocessor can work in two modes of operations : Minimum mode and Maximum mode.

In the minimum mode of operation the microprocessor do not associate with any co-processors and can not be used for multiprocessor systems.

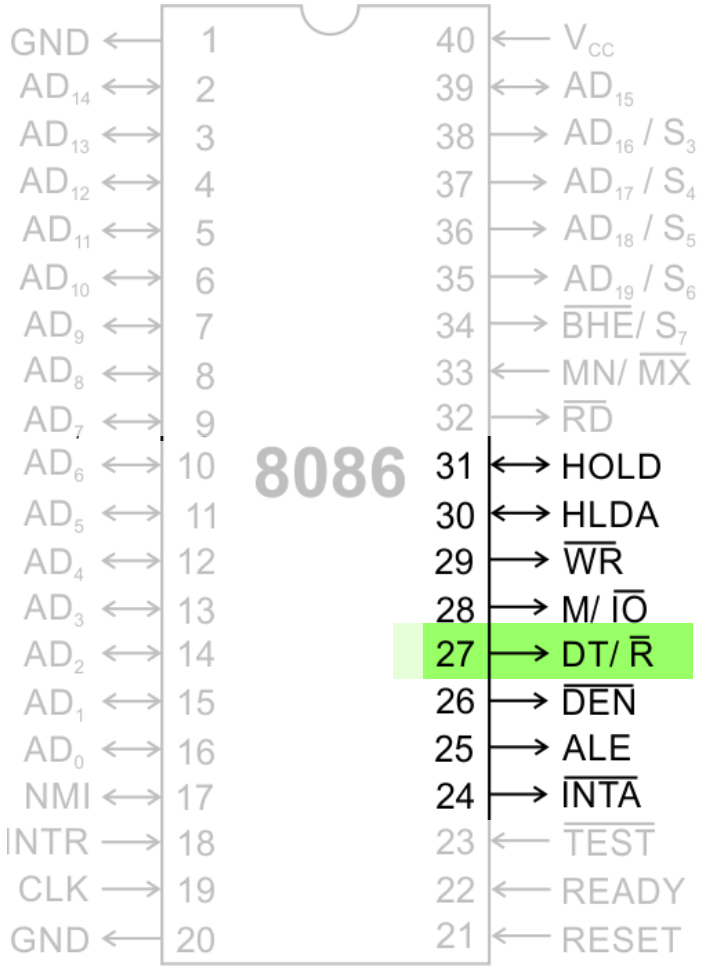
In the maximum mode the 8086 can work in multi-processor or co-processor configuration.

Minimum or maximum mode operations are decided by the pin MN/ MX(Active low).

When this pin is high 8086 operates in minimum mode otherwise it operates in Maximum mode.

Pins and Signals

Minimum mode signals



Pins 24 -31

For minimum mode operation, the MN/ $\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals

DT/ R — (**Data Transmit/ Receive**) Output signal from the processor to control the direction of data flow through the data transceivers

DEN — (**Data Enable**) Output signal from the processor used as out put enable for the transceivers

ALE — (**Address Latch Enable**) Used to demultiplex the address and data lines using external latches

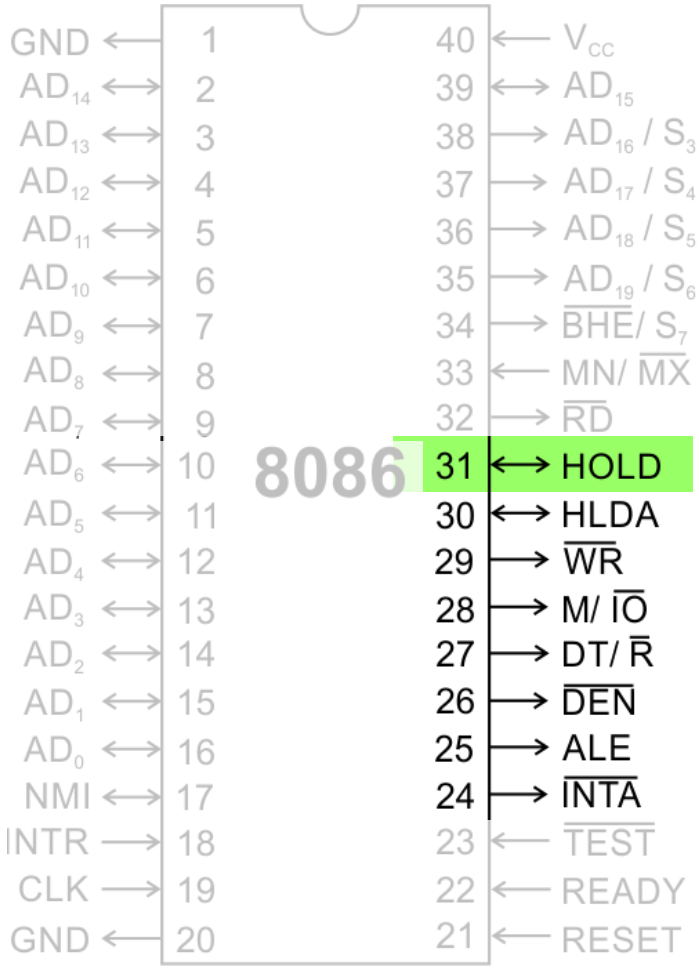
M/IO — Used to differentiate memory access and I/O access. For memory reference instructions, it is **high**. For IN and OUT instructions, it is **low**.

WR — Write control signal; asserted **low** Whenever processor writes data to memory or I/O port

INTA — (**Interrupt Acknowledge**) When the interrupt request is accepted by the processor, the output is **low** on this line.

Pins and Signals

Minimum mode signals



Pins 24 -31

For minimum mode operation, the MN/ $\overline{MX}$ is tied to VCC (logic high)

8086 itself generates all the bus control signals

HOLD — **Input signal to the processor form the bus masters as a request to grant the control of the bus.**

Usually used by the DMA controller to get the control of the bus.

HLDA — **(Hold Acknowledge) Acknowledge signal by the processor to the bus master requesting the control of the bus through HOLD.**

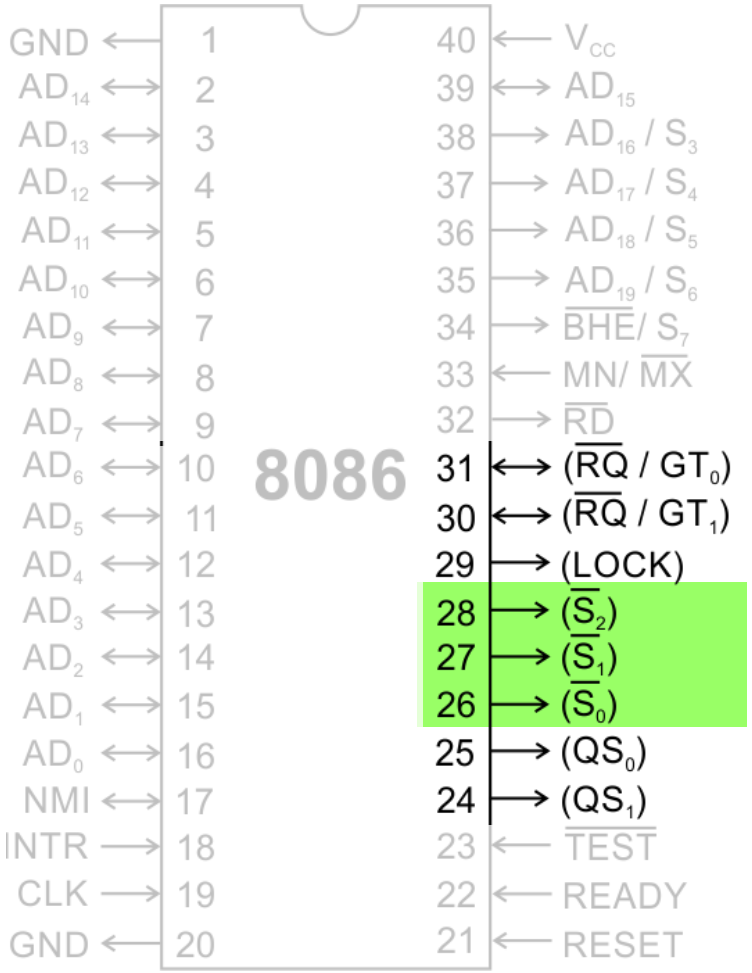
The acknowledge is asserted high, when the processor accepts HOLD.

Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



S_0, S_1, S_2 — **Status signals; used by the 8086 bus controller to generate bus timing and control signals. These are decoded as shown.**

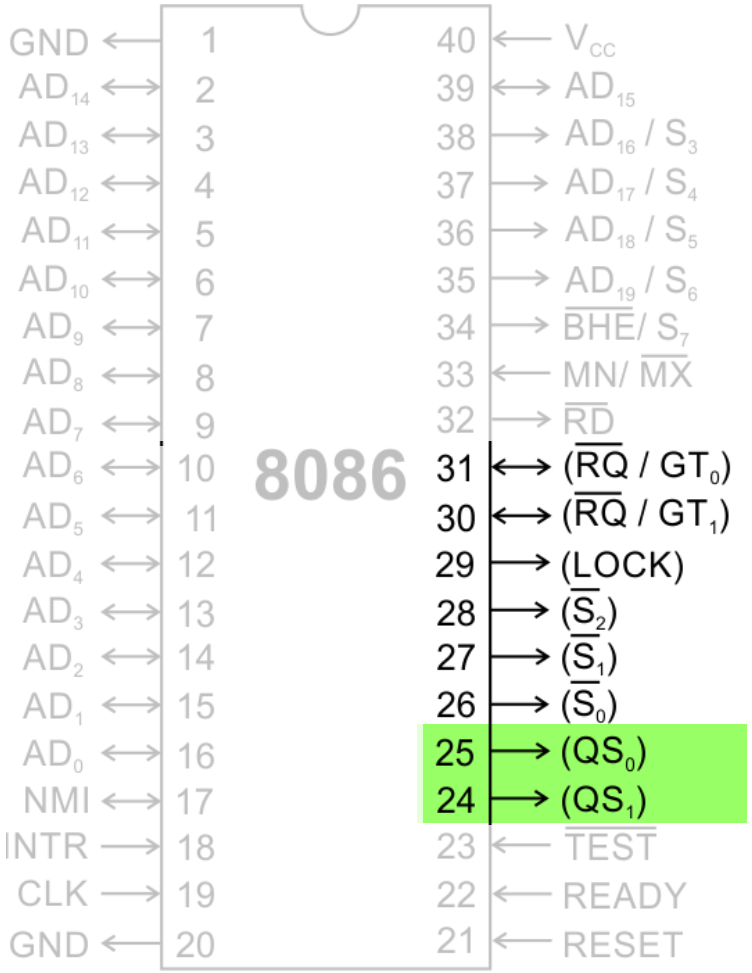
Status Signal			Machine Cycle
$\overline{S}_2$	$\overline{S}_1$	$\overline{S}_0$	
0	0	0	Interrupt acknowledge
0	0	1	Read I/O port
0	1	0	Write I/O port
0	1	1	Halt
1	0	0	Code access
1	0	1	Read memory
1	1	0	Write memory
1	1	1	Passive/Inactive

Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



QS_0, QS_1 **(Queue Status) The processor provides the status of queue in these lines.**

The queue status can be used by external device to track the internal status of the queue in 8086.

The output on QS_0 and QS_1 can be interpreted as shown in the table.

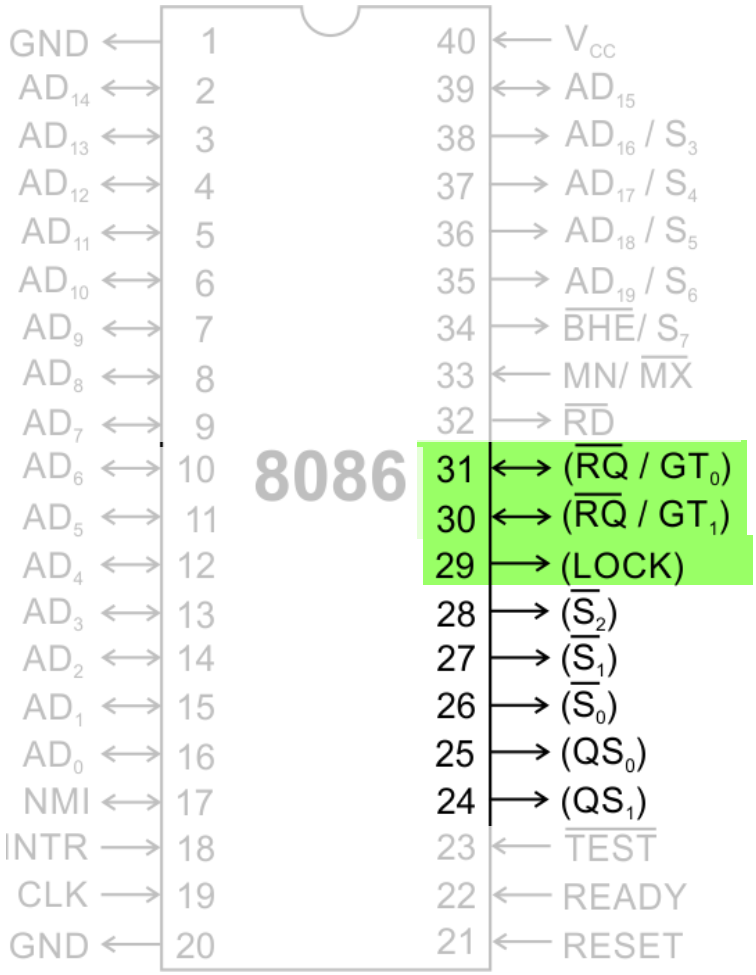
Queue status		Queue operation
QS_1	QS_0	
0	0	No operation
0	1	First byte of an opcode from queue
1	0	Empty the queue
1	1	Subsequent byte from queue

Pins and Signals

Maximum mode signals

During maximum mode operation, the MN/ $\overline{MX}$ is grounded (logic low)

Pins 24 -31 are reassigned



$\overline{RQ}/\overline{GT_0}$, $\overline{RQ}/\overline{GT_1}$ — (Bus Request/ Bus Grant) These requests are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle.

These pins are bidirectional.

The request on $\overline{GT_0}$ will have higher priority than $\overline{GT_1}$

(Bus Request/ Bus Grant) These requests are used by other local bus masters to force the processor to release the local bus at the end of the processor's current bus cycle.

These pins are bidirectional.

The request on $\overline{GT_0}$ will have higher priority than $\overline{GT_1}$

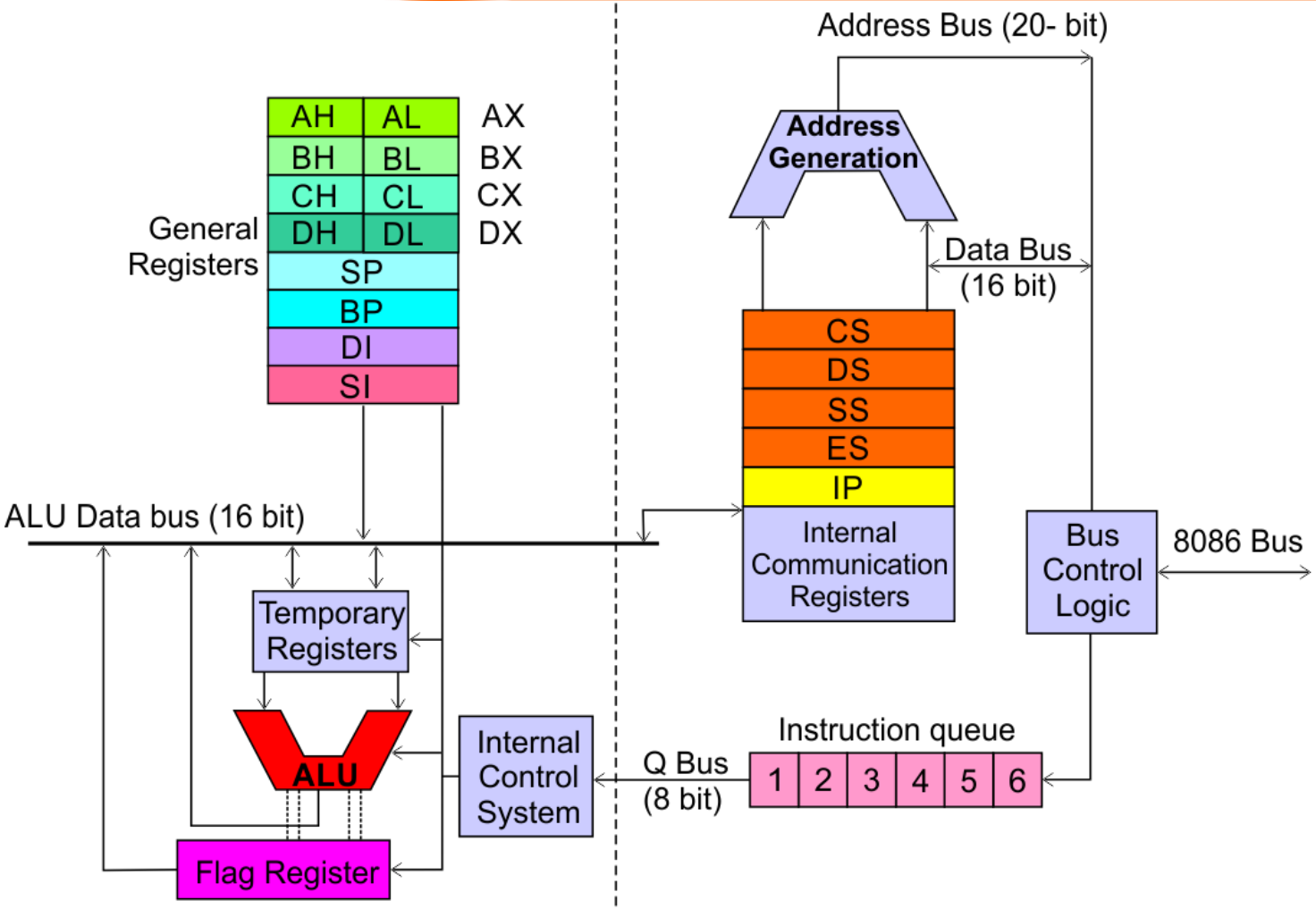
LOCK — **An output signal activated by the LOCK prefix instruction.**

Remains active until the completion of the instruction prefixed by LOCK.

The 8086 output low on the pin while executing an instruction prefixed by LOCK to prevent other bus masters from gaining control of the system bus.

Architecture

Architecture



Execution Unit (EU)

EU executes instructions that have already been fetched by the BIU.

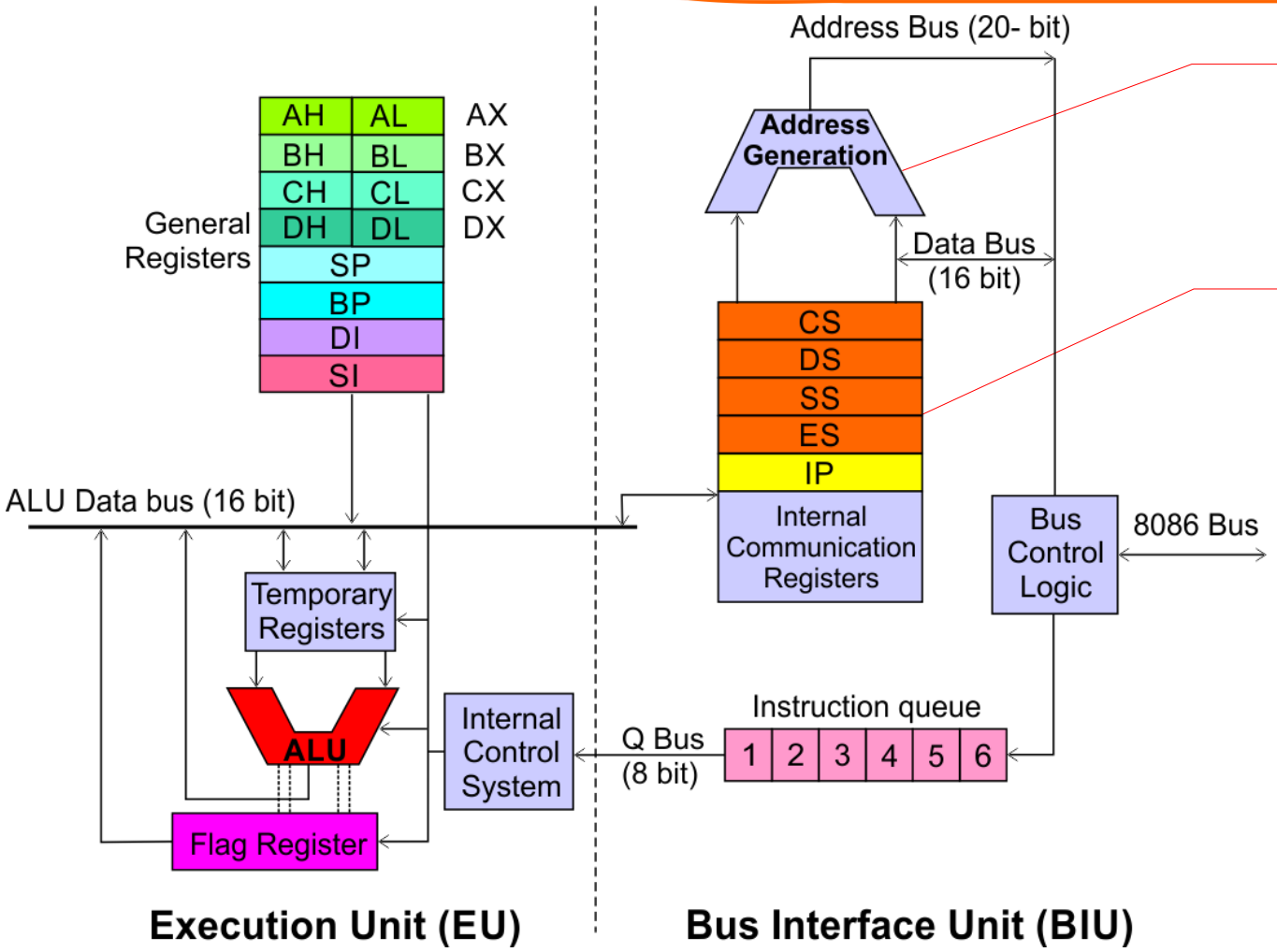
BIU and EU functions separately.

Bus Interface Unit (BIU)

BIU fetches instructions, reads data from memory and I/O ports, writes data to memory and I/ O ports.

Architecture

Bus Interface Unit (BIU)



Dedicated Adder to generate 20 bit address

Four 16-bit segment registers

Code Segment (CS)
Data Segment (DS)
Stack Segment (SS)
Extra Segment (ES)

Addition using Assembly Languages

8 bit addition

```
data segment
a db 09h
b db 02h
c dw ?
data ends

code segment
assume cs:code,ds:data
start:
mov ax,data
mov ds,ax
mov al,a
mov bl,b
add al,bl
mov c,ax
int 3
code ends
end start
```

Addition using Assembly Languages

16 bit addition

```
MOV AX, 4343
MOV BX, 1111
ADD AX, BX
```