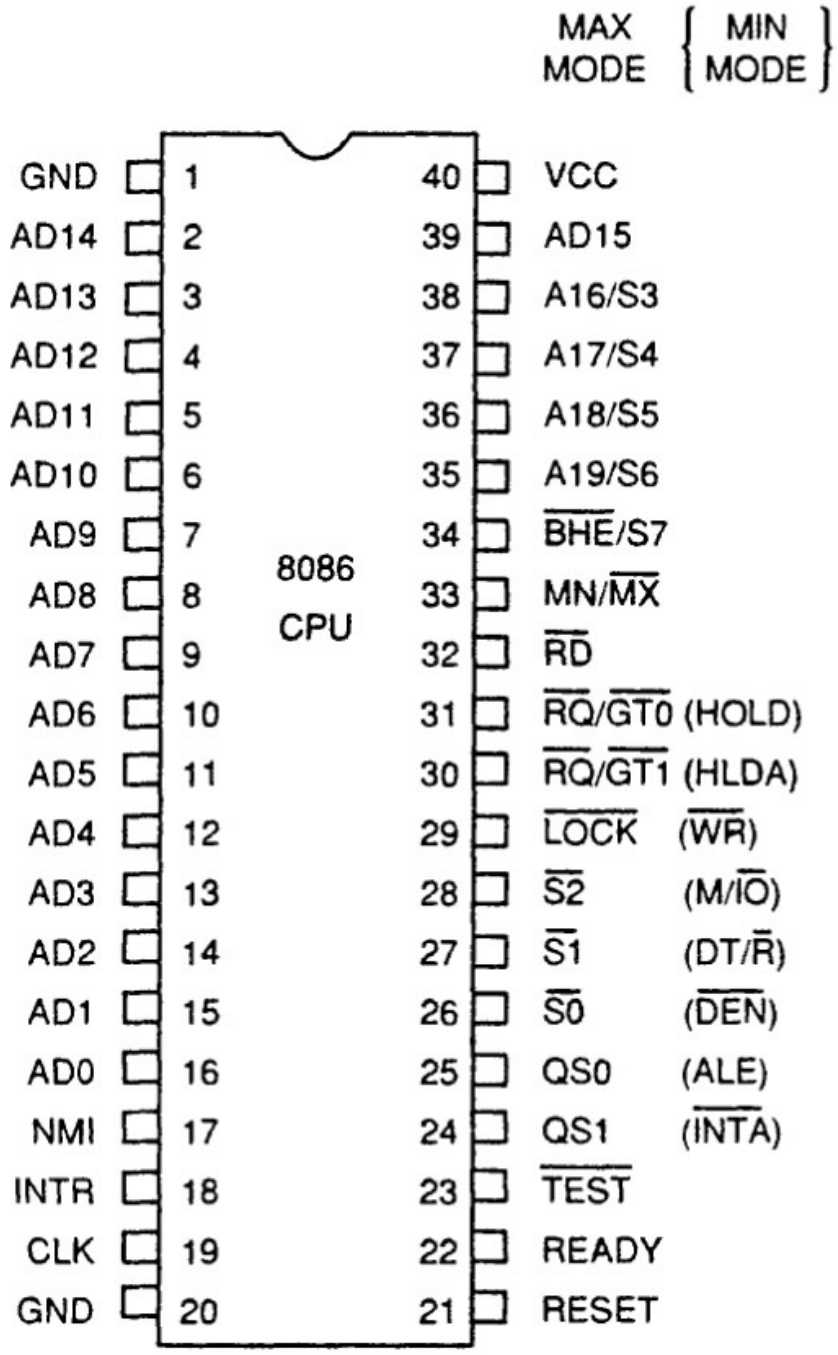
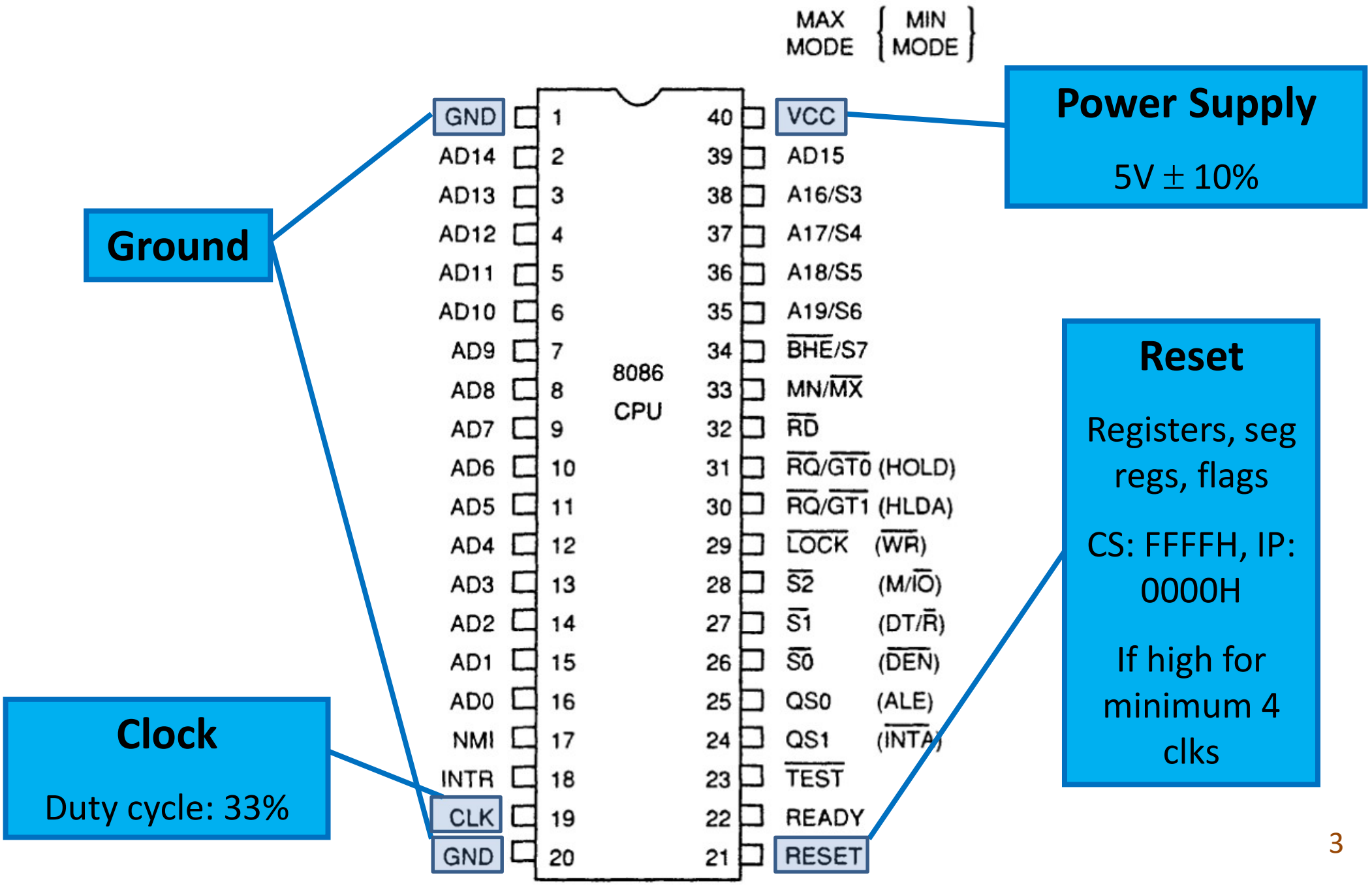
UEC1502 MICRPROCESSORS, MICROCONTROLLERS AND ITS INTERFACING

M. Anbuselvi
ASP/ECE

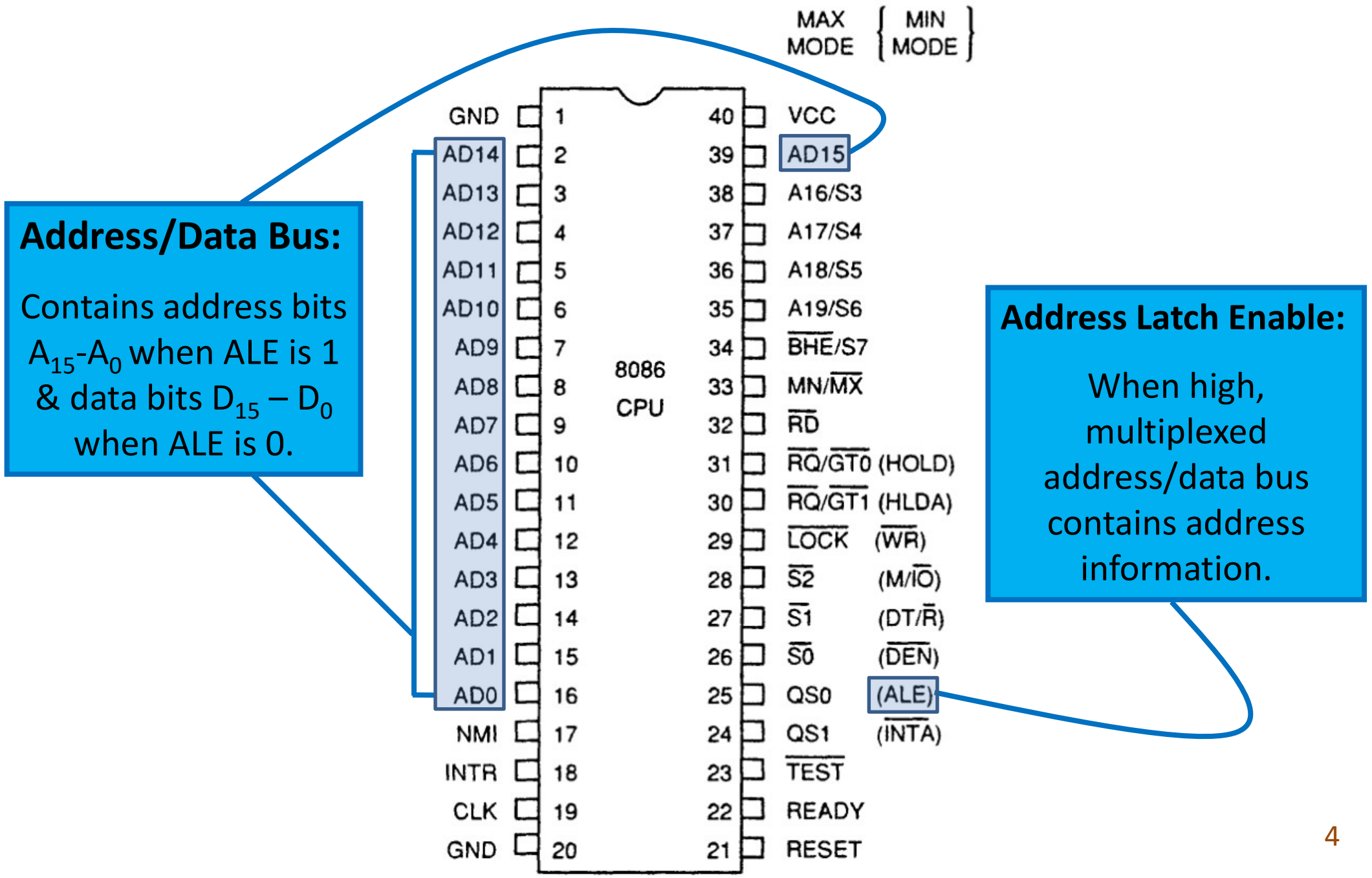
INTEL 8086 - Pin Diagram



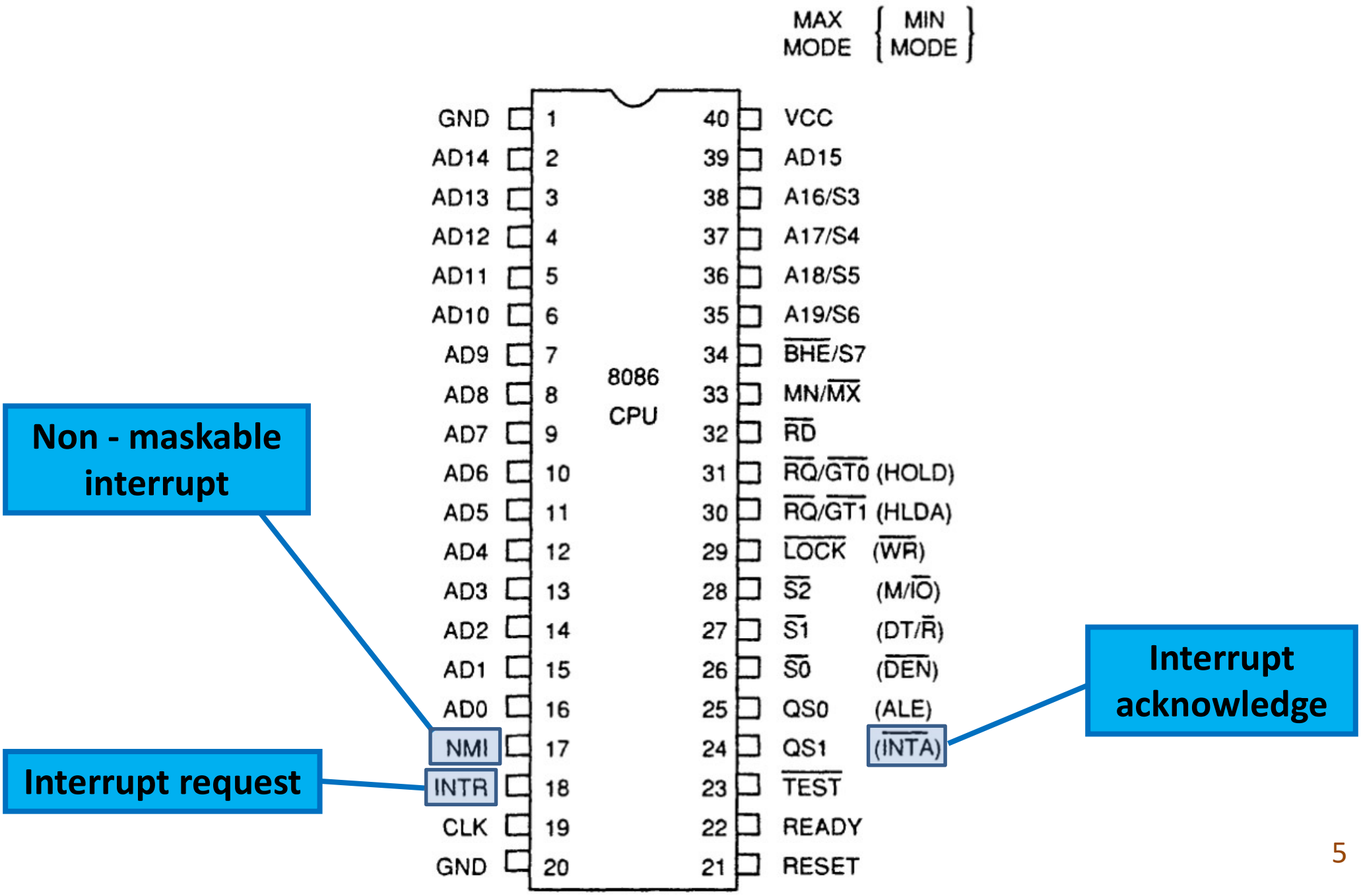
INTEL 8086 - Pin Details



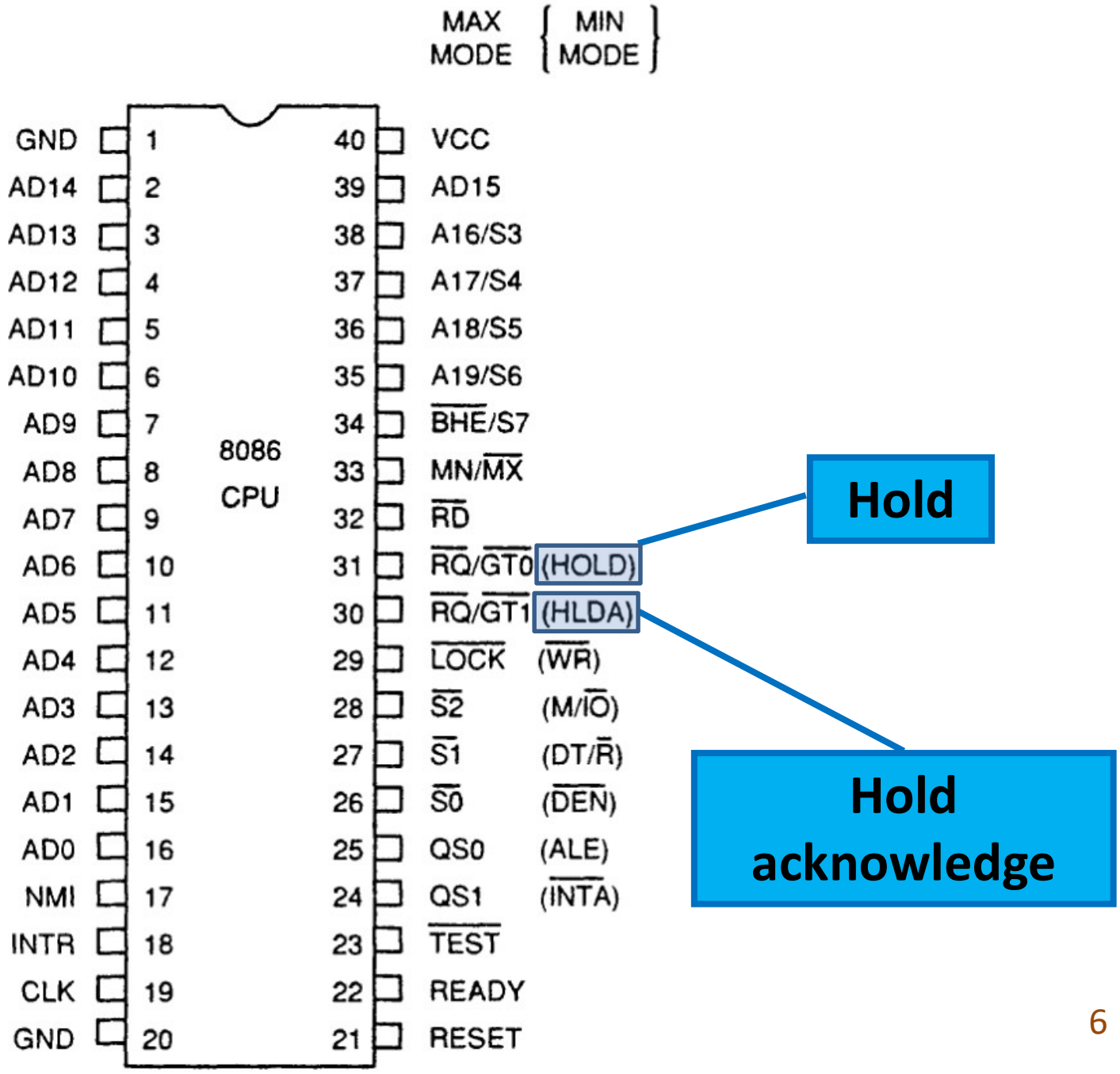
INTEL 8086 - Pin Details



INTEL 8086 - Pin Details



INTEL 8086 - Pin Details



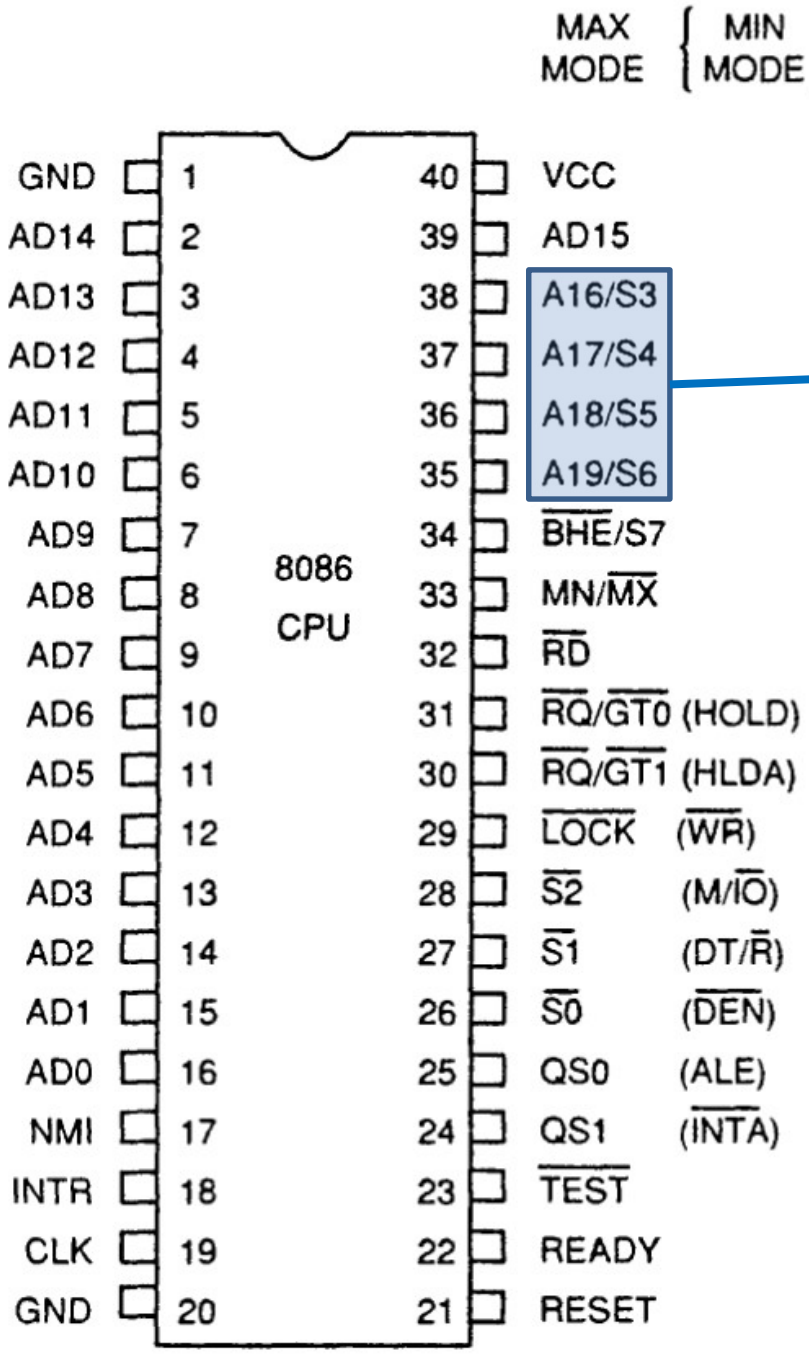
INTEL 8086 - Pin Details

S6: Logic 0.

S5: Indicates condition of IF flag bits.

S4-S3: Indicate which segment is 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



Address/Status Bus

Address bits A_{19} – A_{16} & Status bits S_6 – S_3

INTEL 8086 - Pin Details

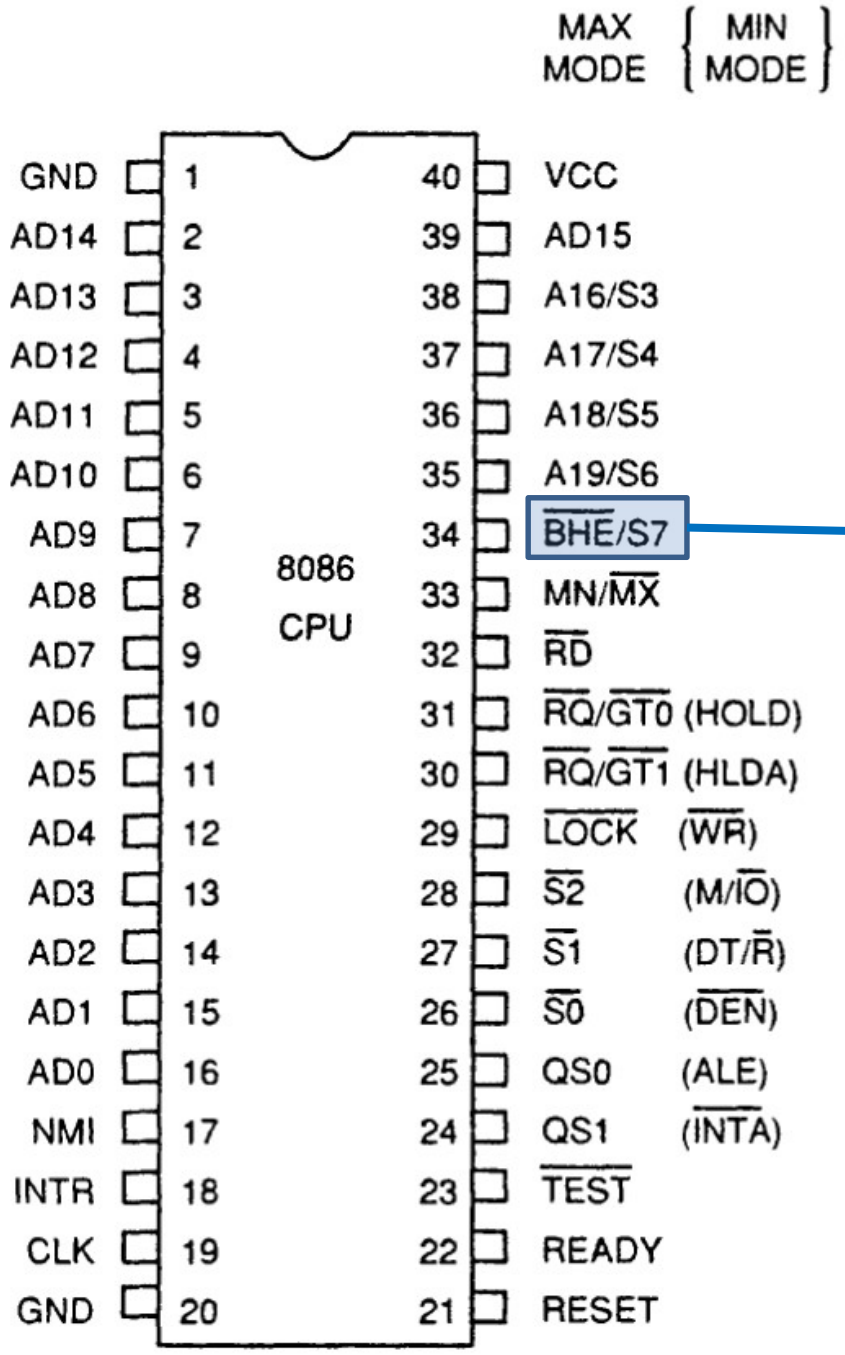
BHE#, A_0:

0,0: Whole word (16-bits)

0,1: High byte to/from odd address

1,0: Low byte to/from even address

1,1: No selection

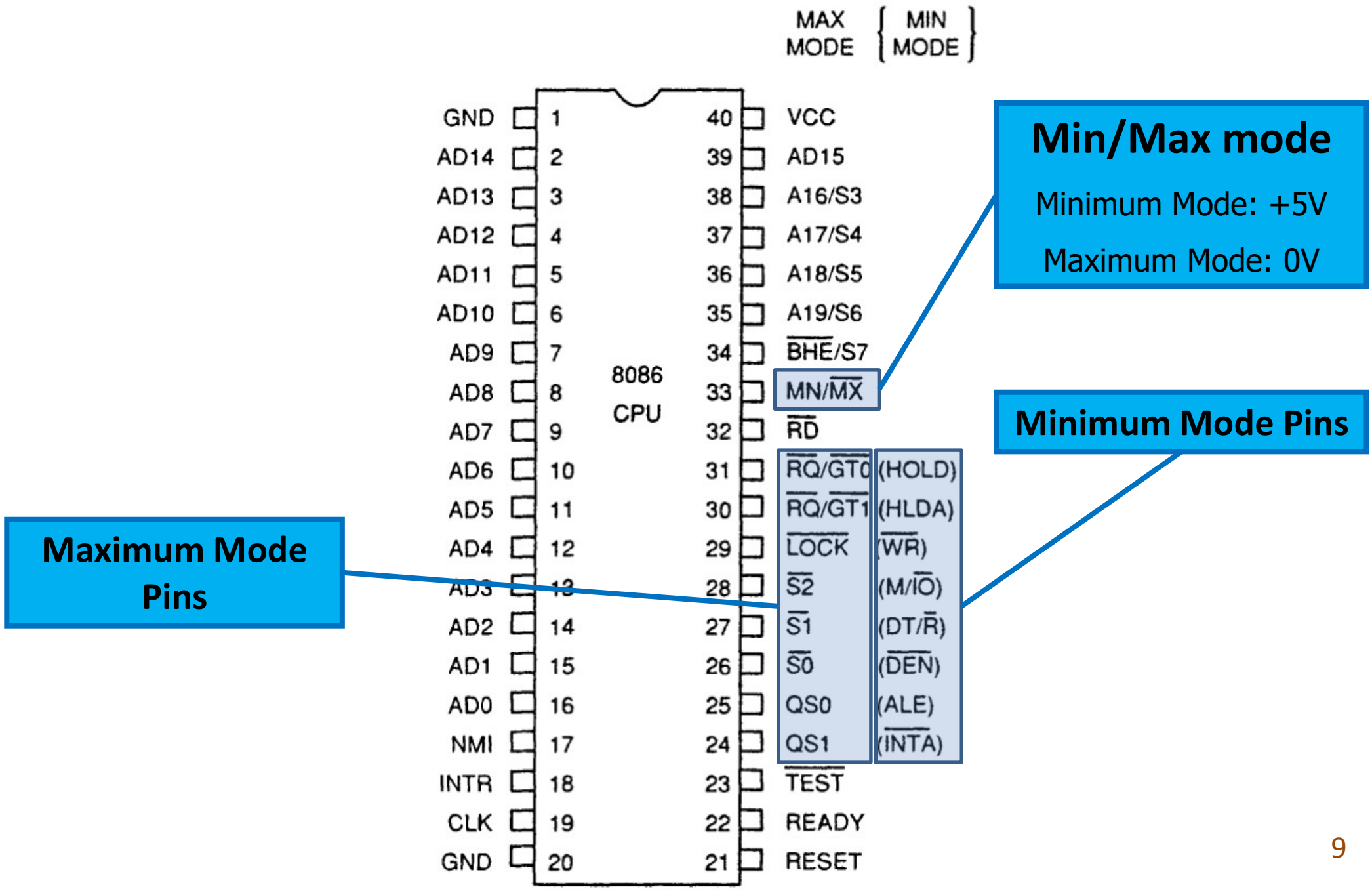


Bus High Enable/S7

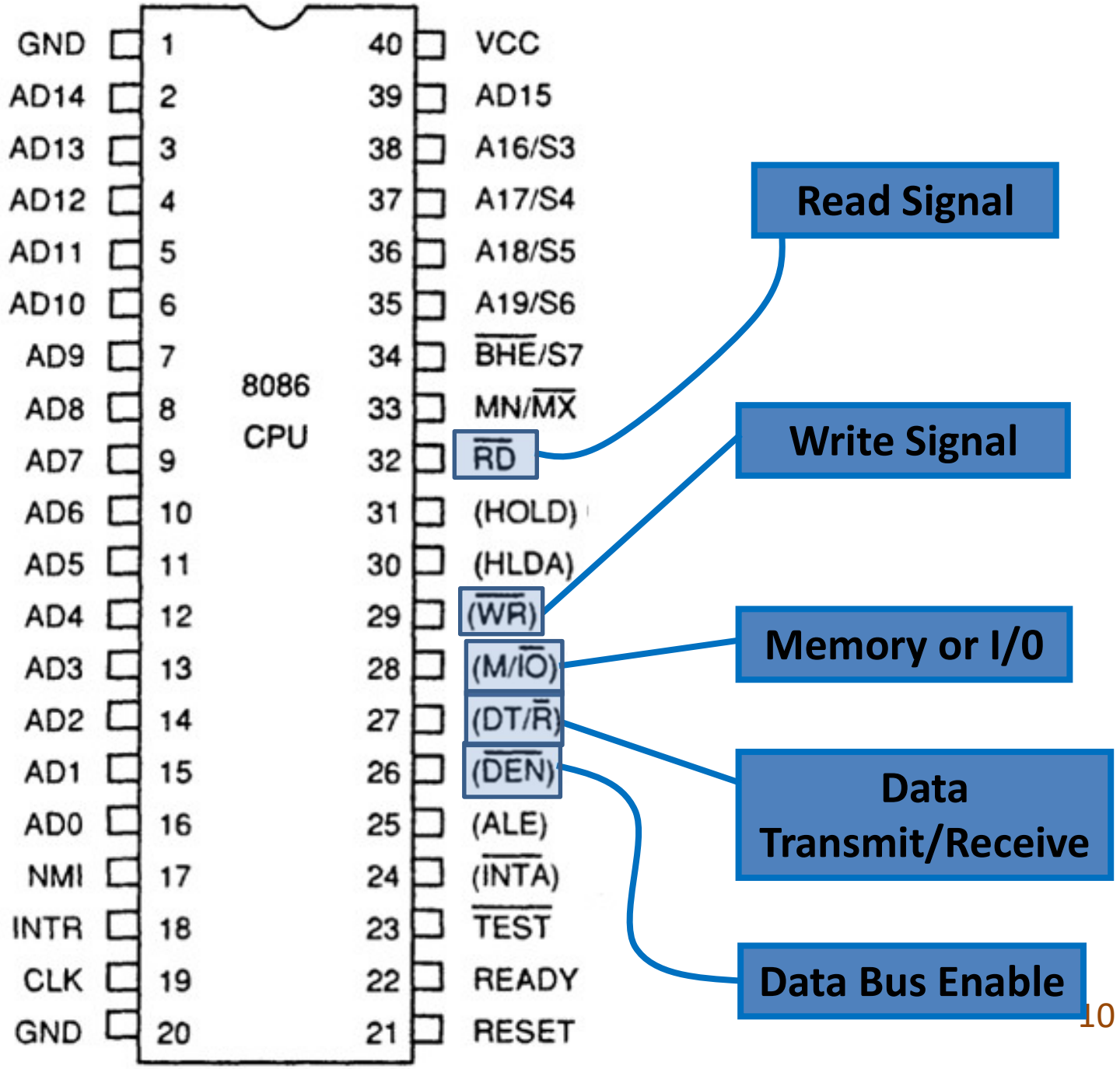
Enables most significant data bits $D_{15} – D_8$ during read or write operation.

S_7: Always 1.

INTEL 8086 - Pin Details



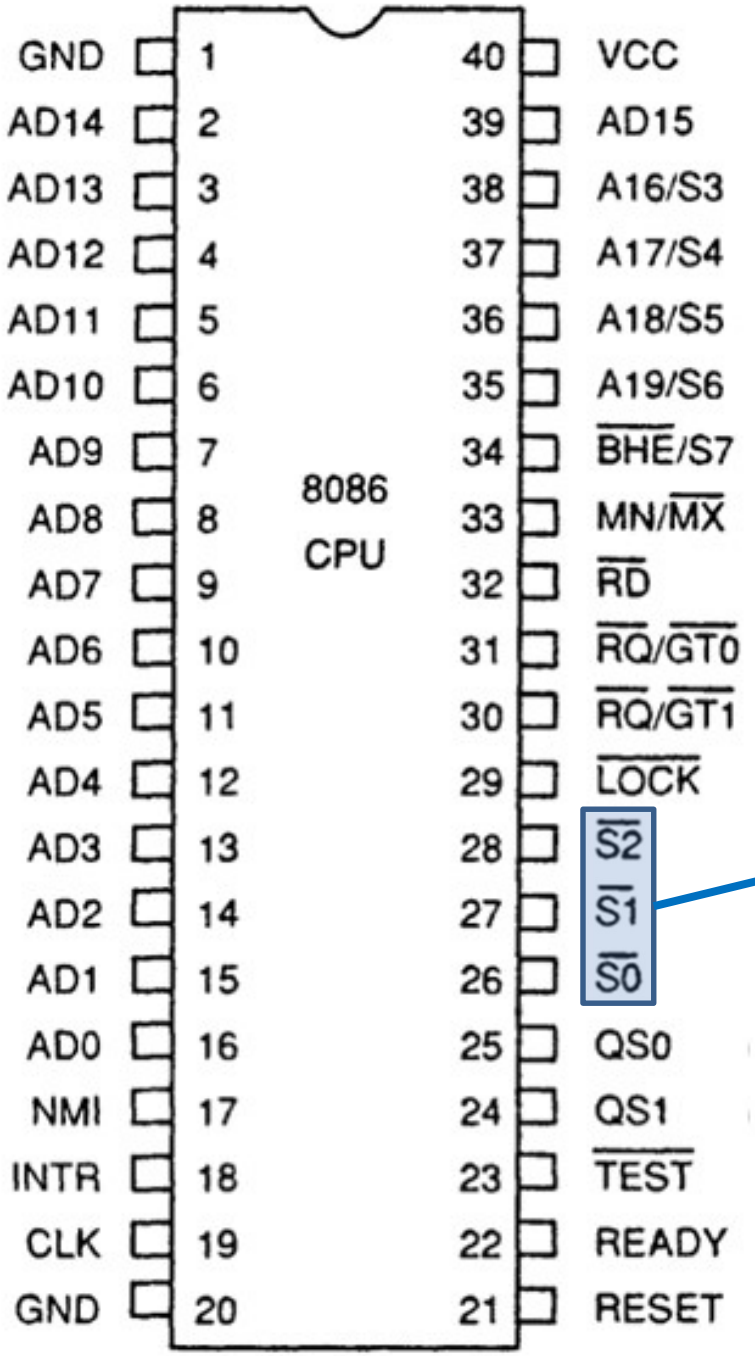
Minimum Mode- Pin Details



Maximum Mode - Pin Details

S2 S1 S0

000: INTA
001: read I/O port
010: write I/O port
011: halt
100: code access
101: read memory
110: write memory
111: none -passive



Status Signal

Inputs to 8288 to generate eliminated signals due to max mode.

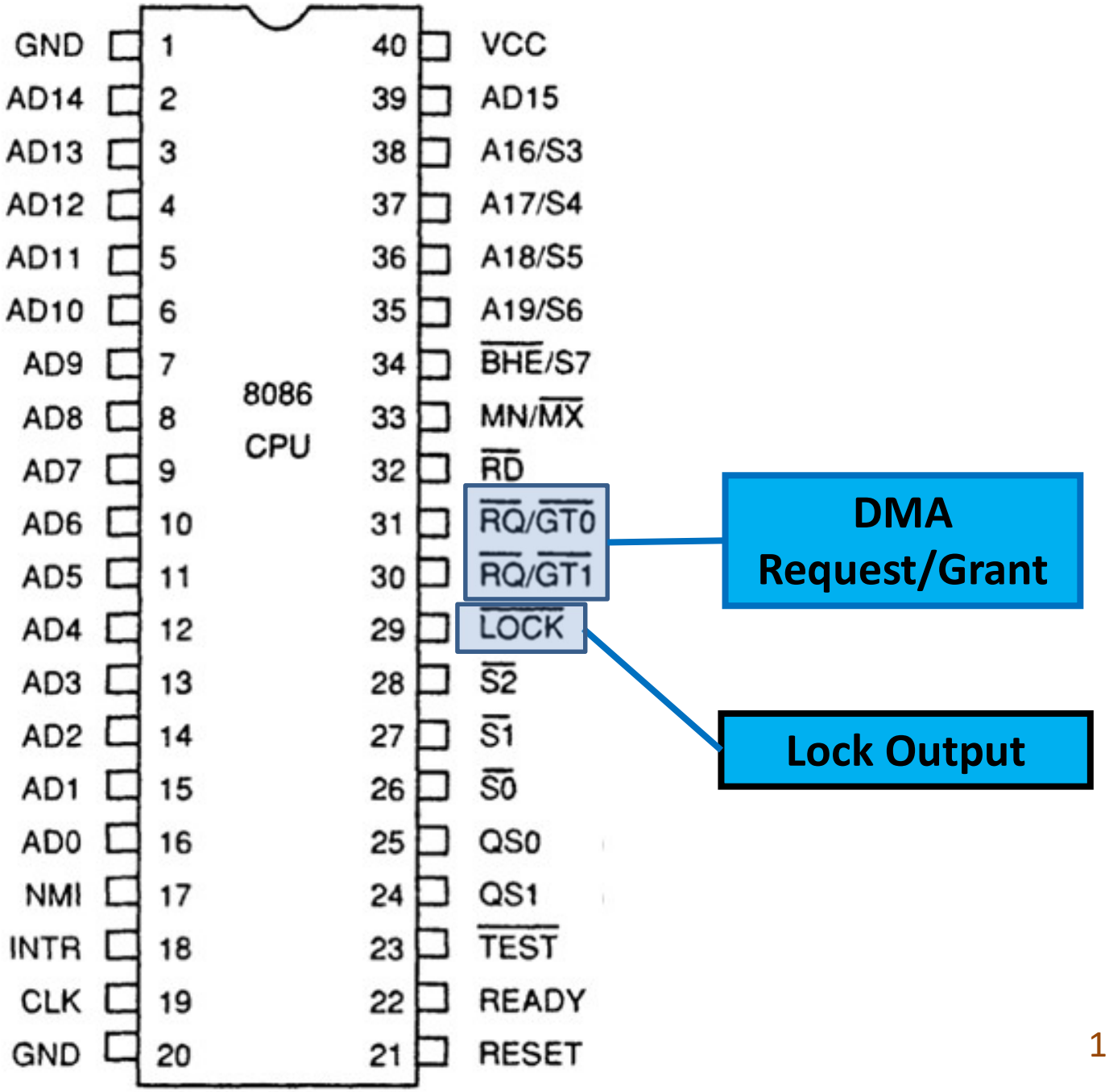
Maximum Mode - Pin Details

Lock Output

Used to lock peripherals off the system

Activated by using the LOCK:

prefix on any instruction



Maximum Mode - Pin Details

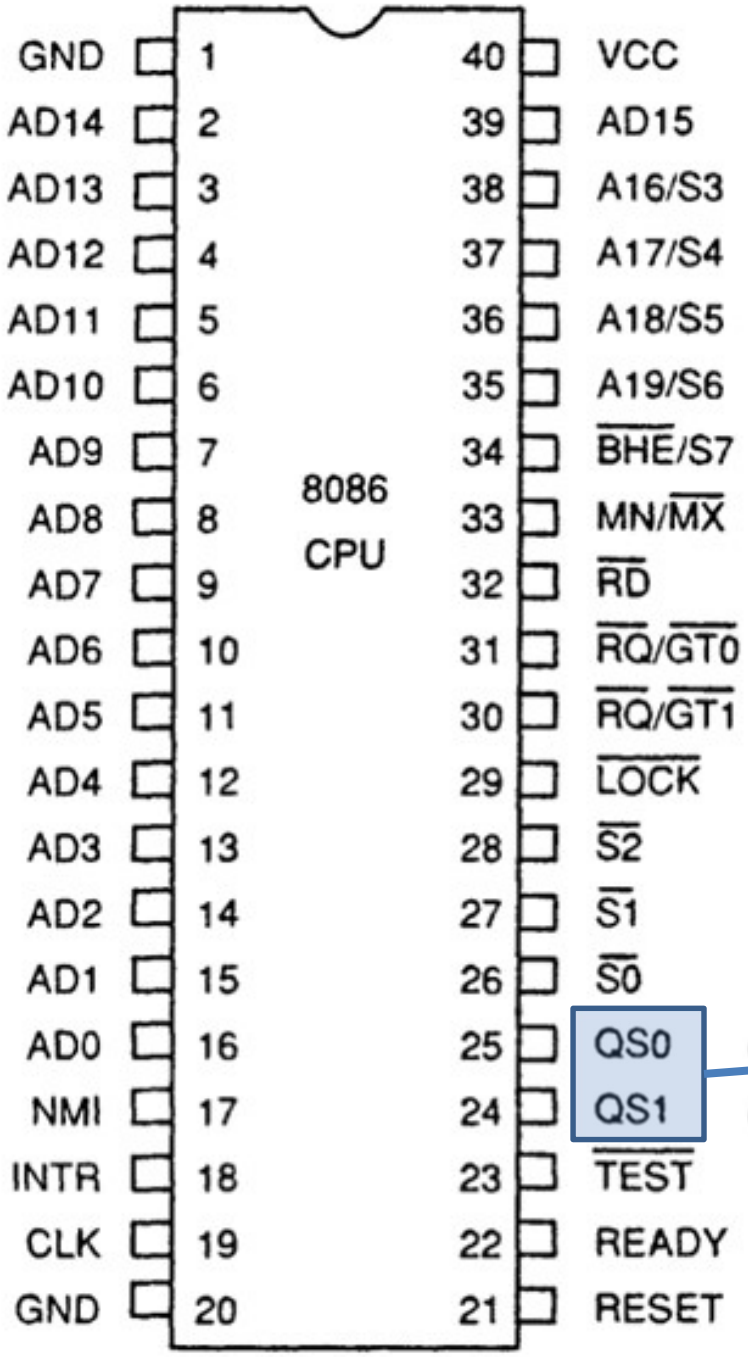
QS1 QS0

00: Queue is idle

01: First byte of opcode

10: Queue is empty

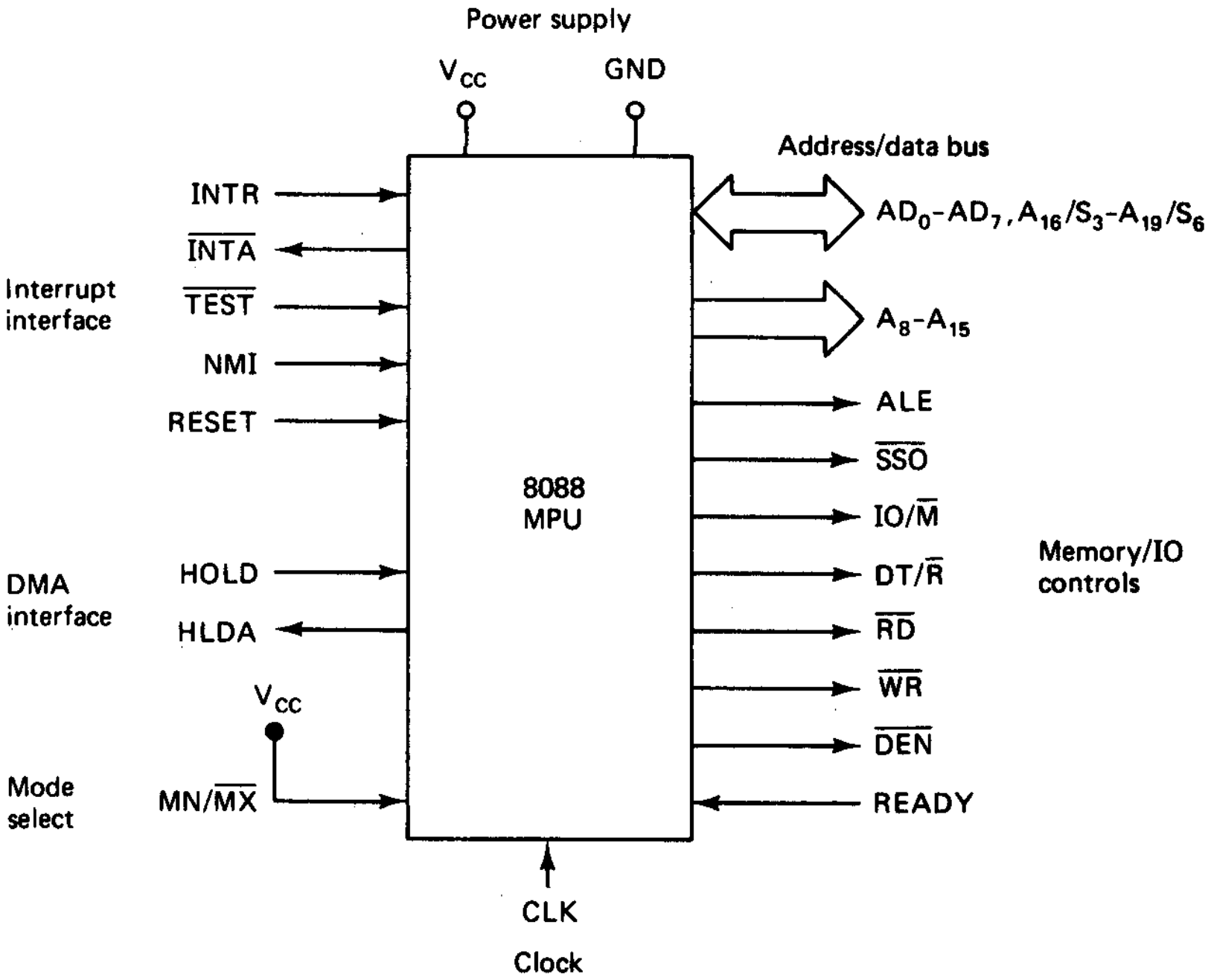
11: Subsequent byte of opcode



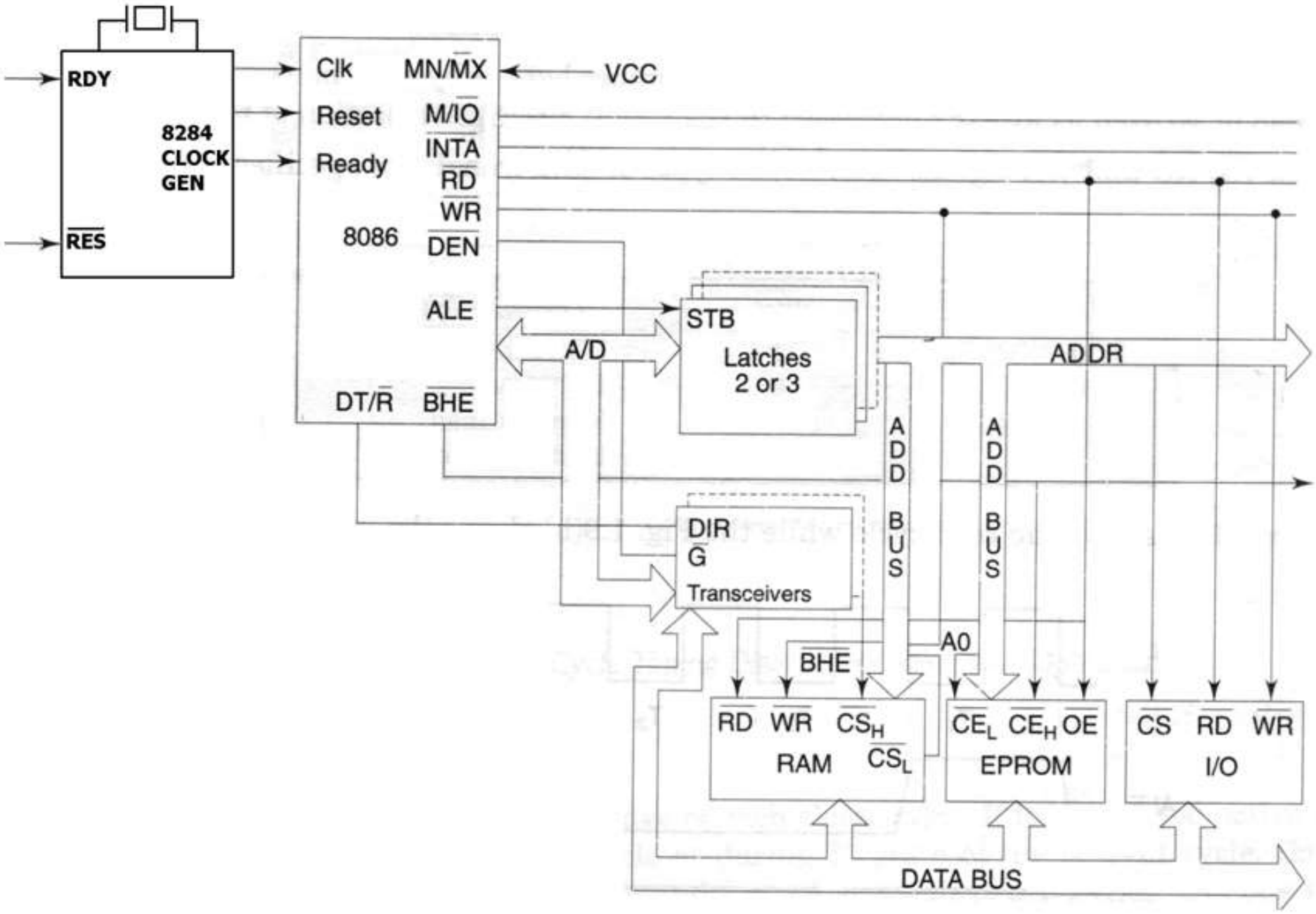
Queue Status

Used by numeric coprocessor (8087)

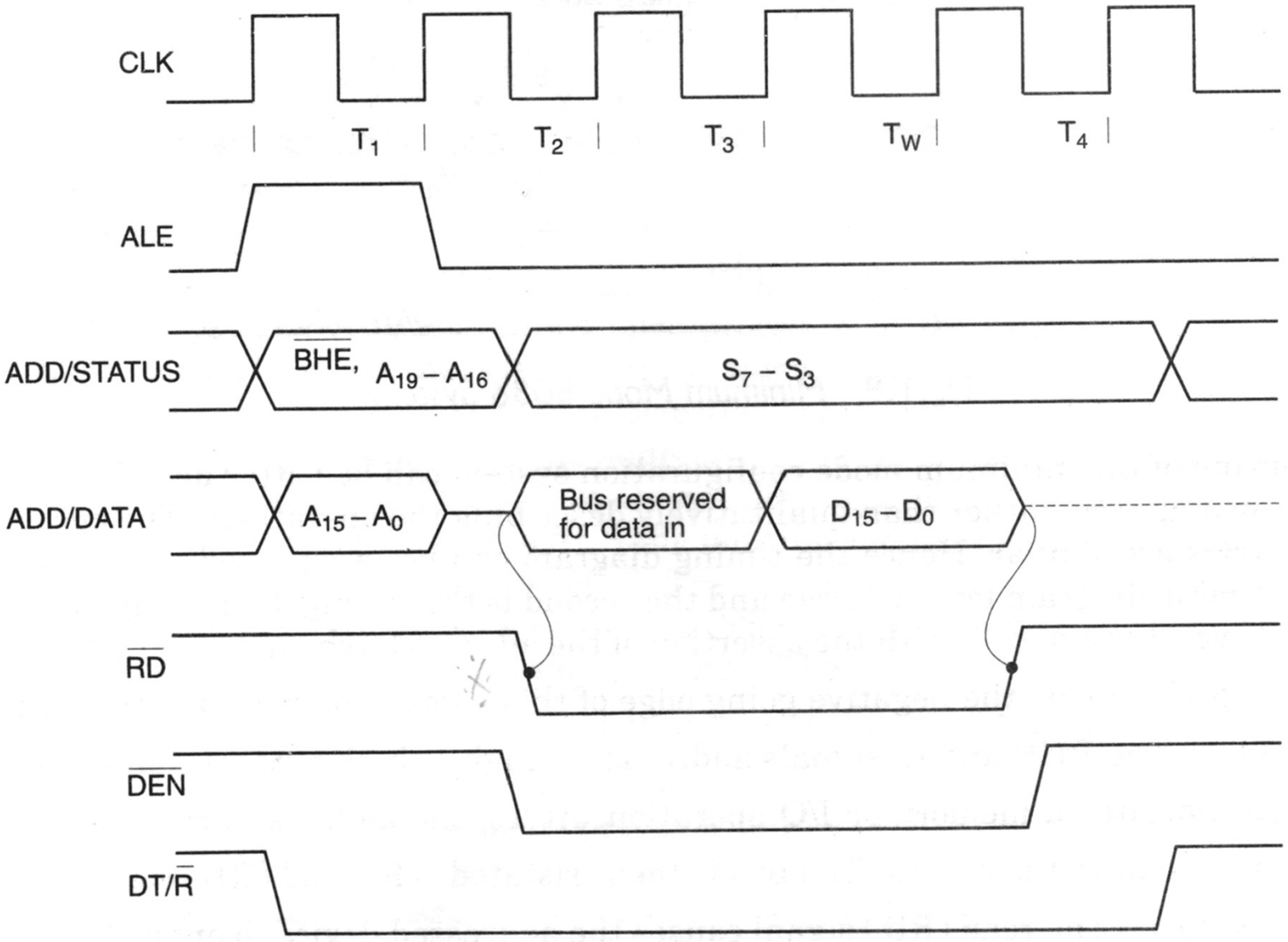
Minimum Mode 8086 System



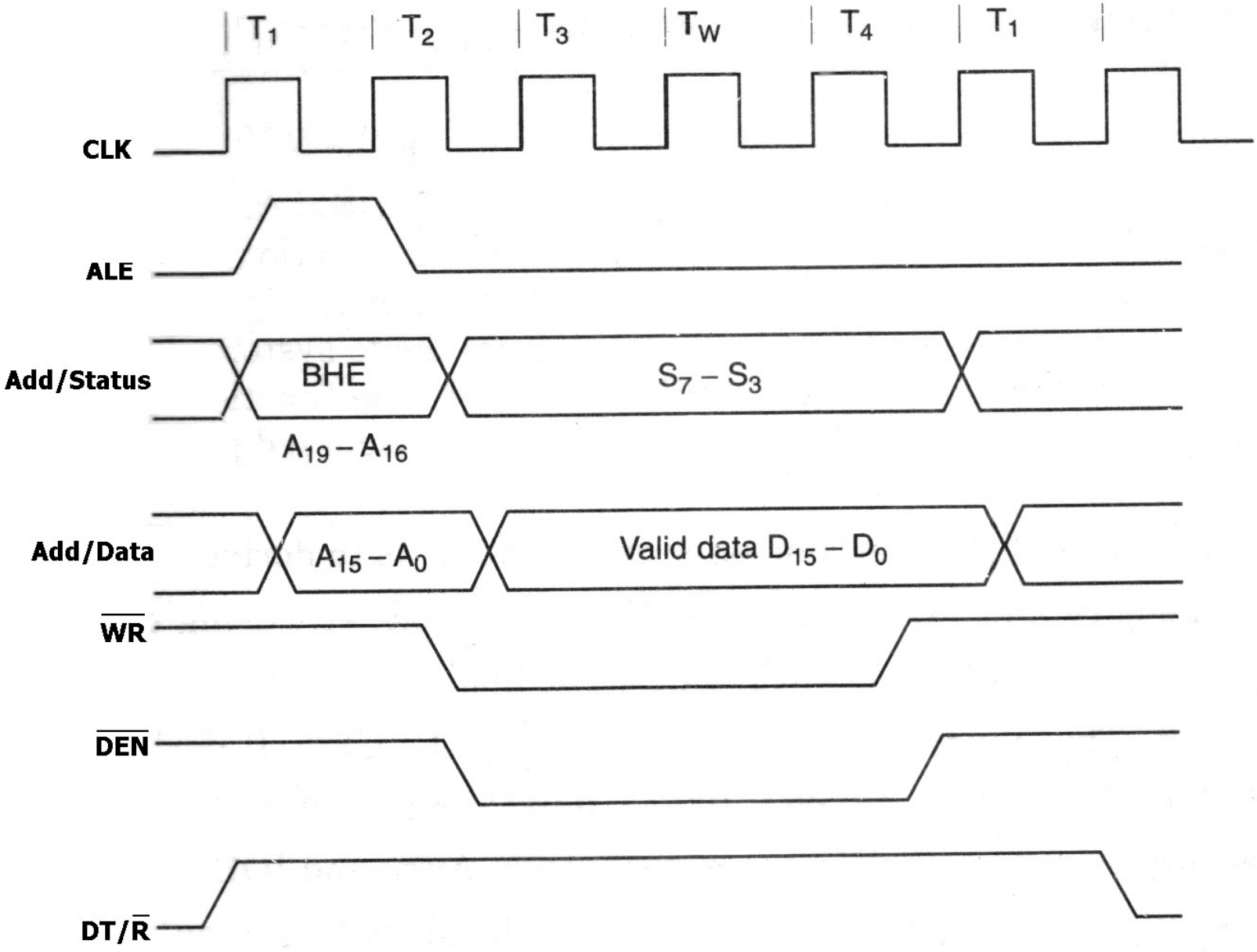
Minimum Mode 8086 System



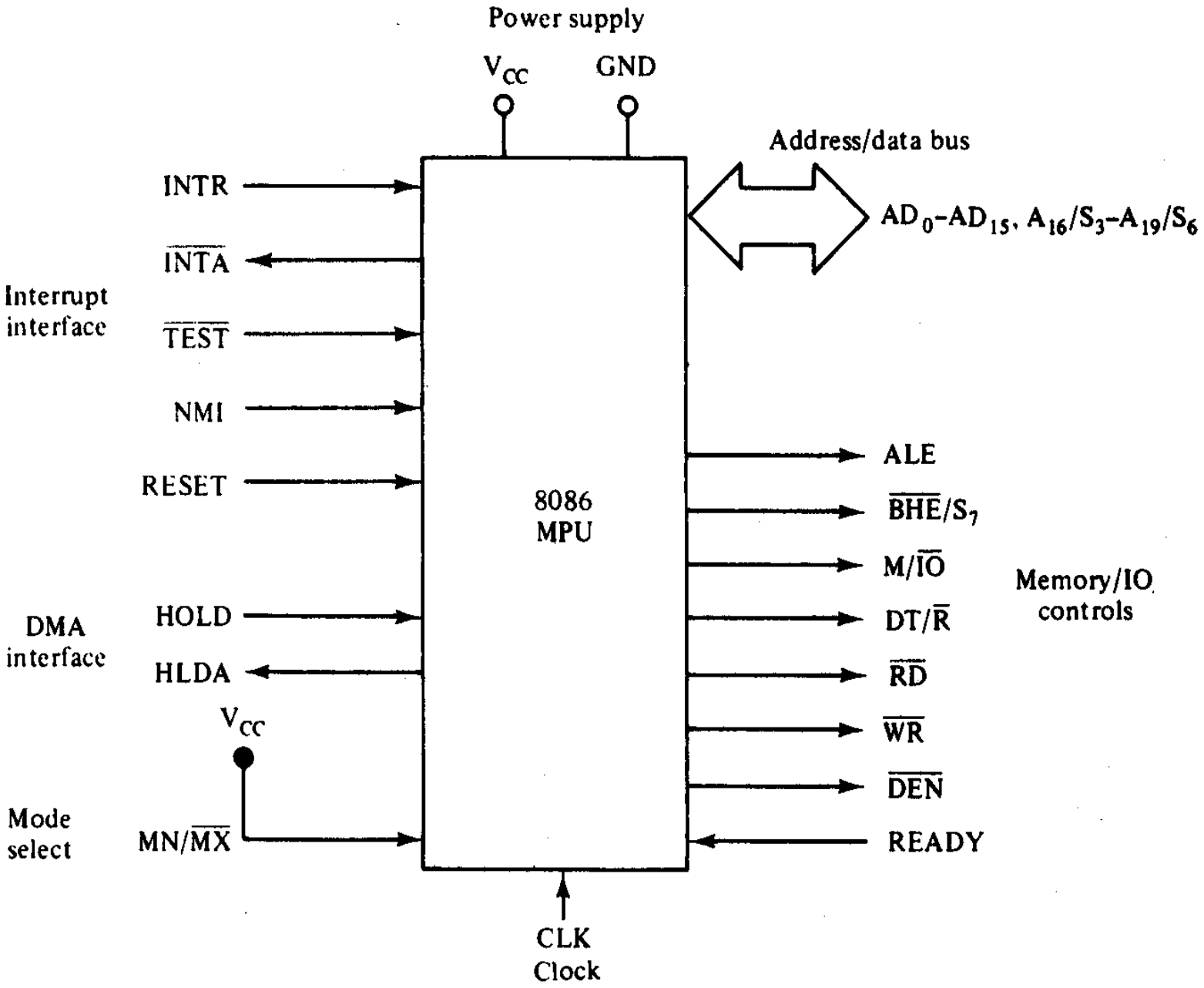
‘Read’ Cycle timing Diagram for Minimum Mode



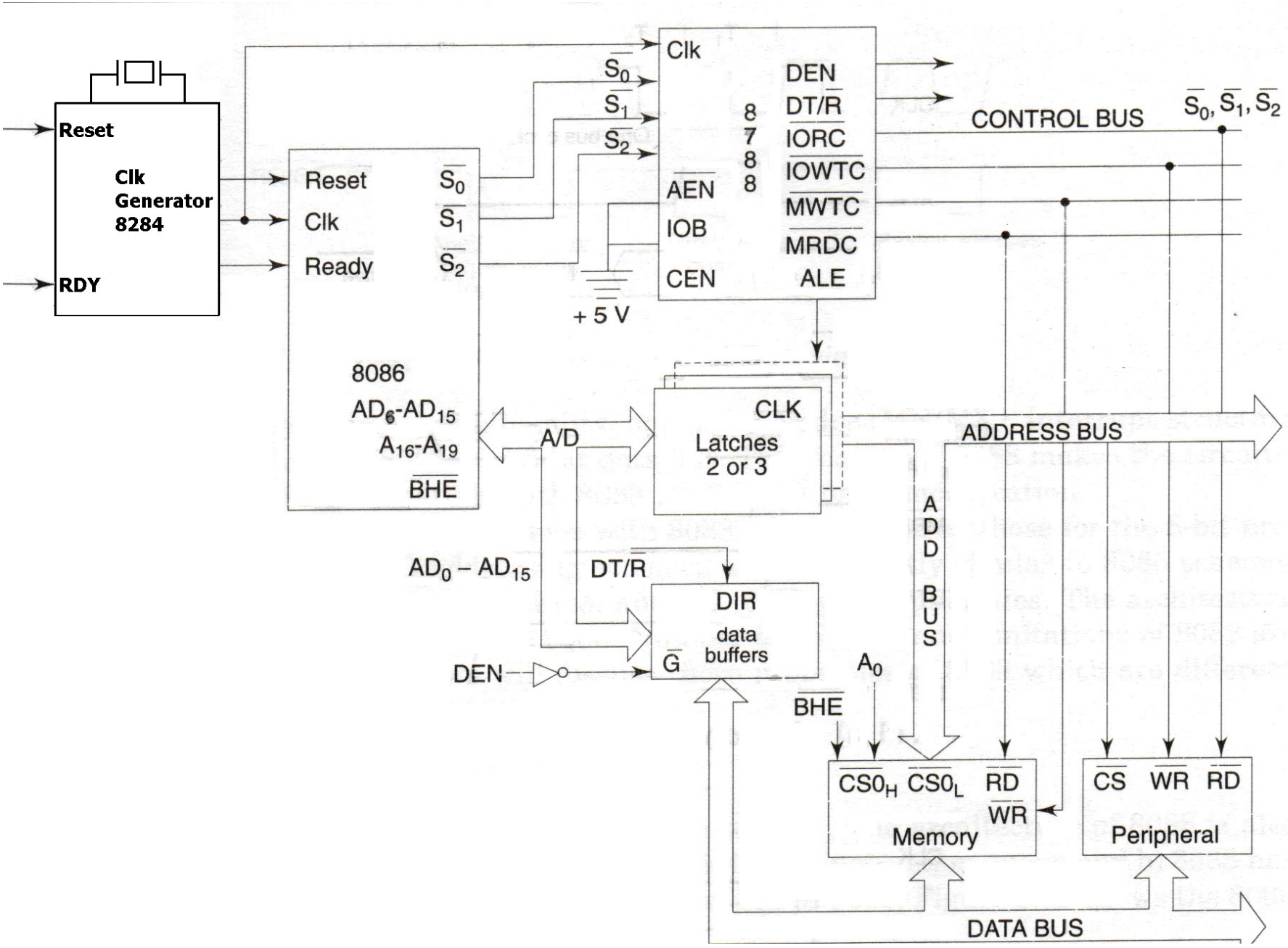
'Write' Cycle timing Diagram for Minimum Mode



Maximum Mode 8086 System



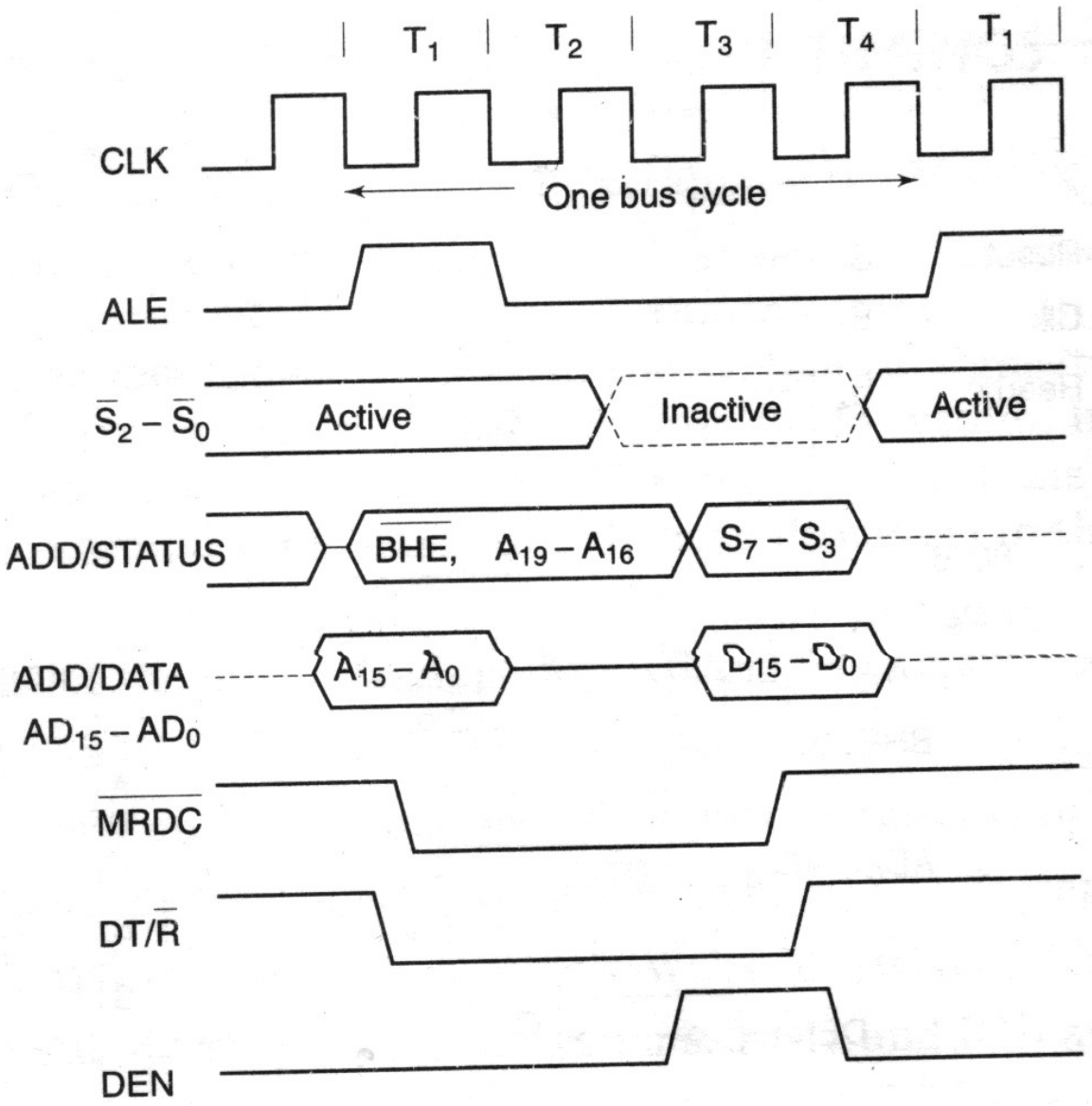
Maximum Mode 8086 System



Maximum Mode 8086 System

- Here, either a numeric coprocessor of the type 8087 or another processor is interfaced with 8086.
- The Memory, Address Bus, Data Buses are shared resources between the two processors.
- The control signals for Maximum mode of operation are generated by the Bus Controller chip 8788.
- The three status outputs S0*, S1*, S2* from the processor are input to 8788.
- The outputs of the bus controller are the Control Signals, namely DEN, DT/R*, IORC*, IOWTC*, MWTC*, MRDC*, ALE etc.

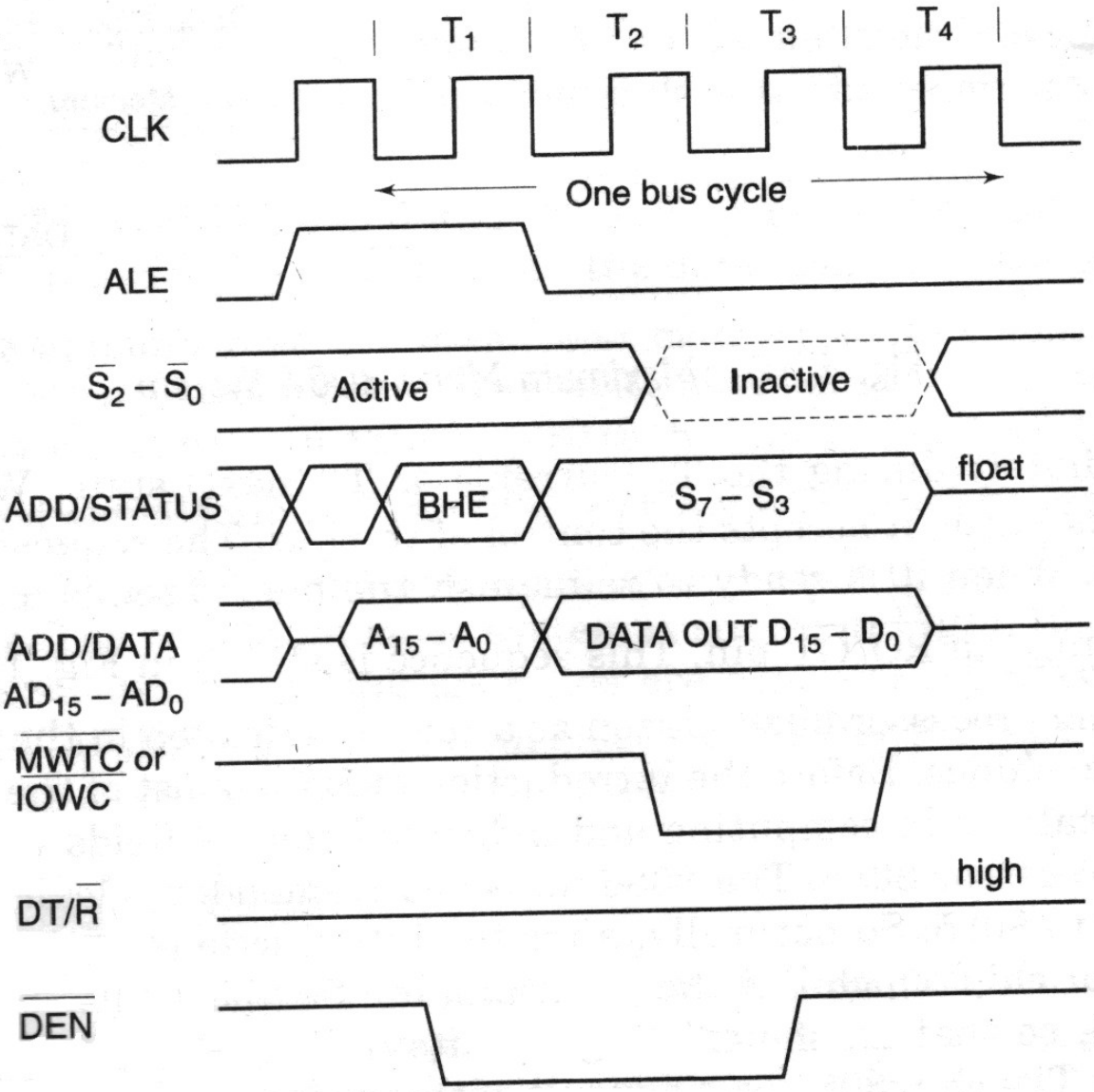
Memory Read timing in Maximum Mode



S2	S1	SO	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 8–6 Bus control functions generated by the bus controller (8288) using $\overline{S2}$, $\overline{S1}$, and $\overline{SO}$

Memory Write timing in Maximum Mode



$\overline{S2}$	$\overline{S1}$	$\overline{SO}$	*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 8–6 Bus control functions generated by the bus controller (8288) using $\overline{S2}$, $\overline{S1}$, and $\overline{SO}$

8086 Control Signals

1. **ALE**
2. **BHE**
3. **M/IO**
4. **DT/R**
5. **RD**
6. **WR**
7. **DEN**

Coprocessor and Multiprocessor configuration

- Multiprocessor Systems refer to the use of multiple processors that **executes instructions simultaneously** and **communicate with each other** using mail boxes and Semaphores.

- Maximum mode of 8086 is designed to implement 3 basic multiprocessor configurations:

  1. Coprocessor (8087)

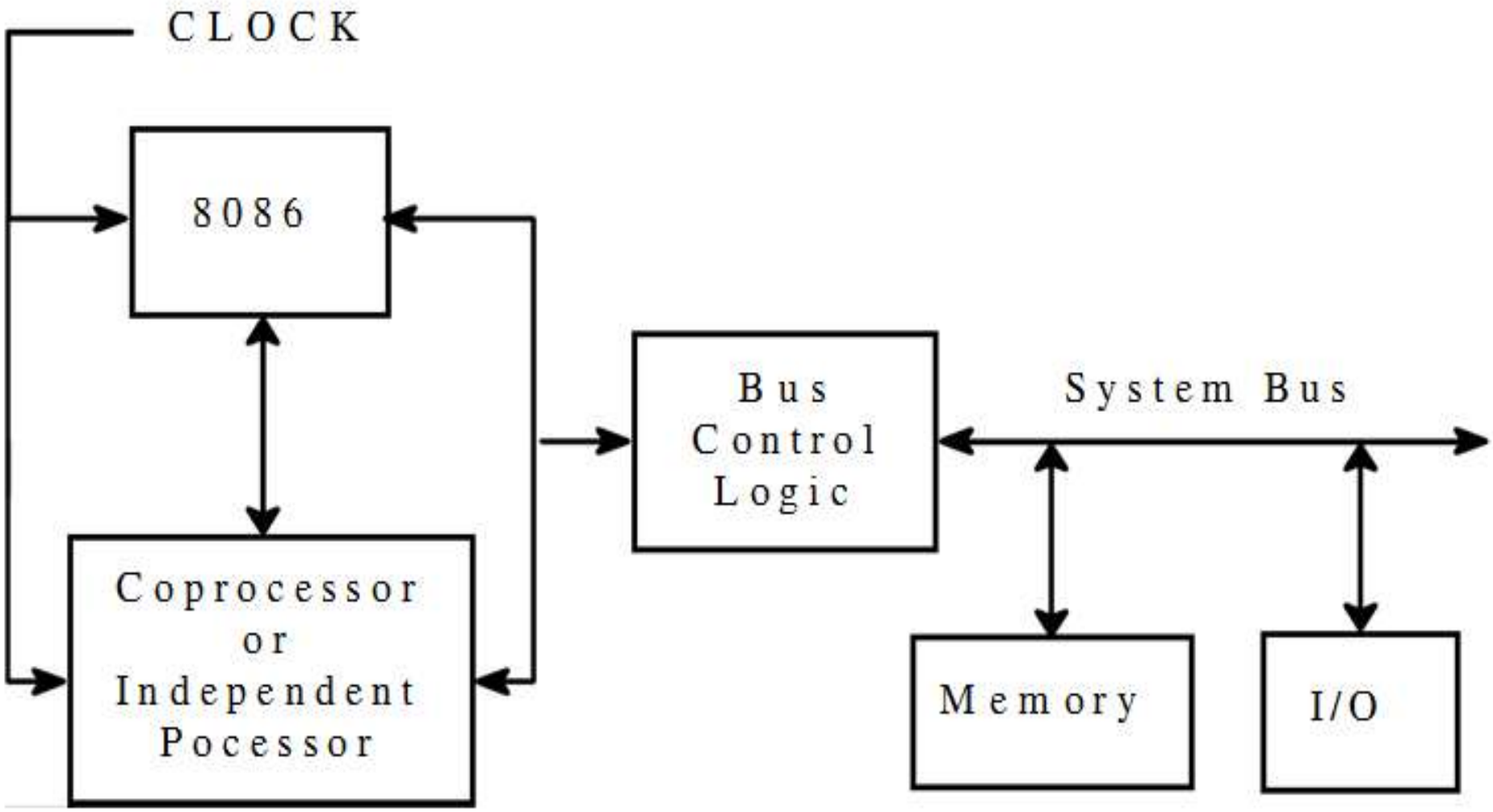
  2. Closely coupled (8089)

  3. Loosely coupled (Multibus)

Coprocessor and Multiprocessor configuration

- **Coprocessors** and **Closely coupled** configurations are similar in that both the 8086 and the external processor shares the:
  - **Memory**
  - **I/O system**
  - **Bus & bus control logic**
  - **Clock generator**

Coprocessor / Closely Coupled Configuration

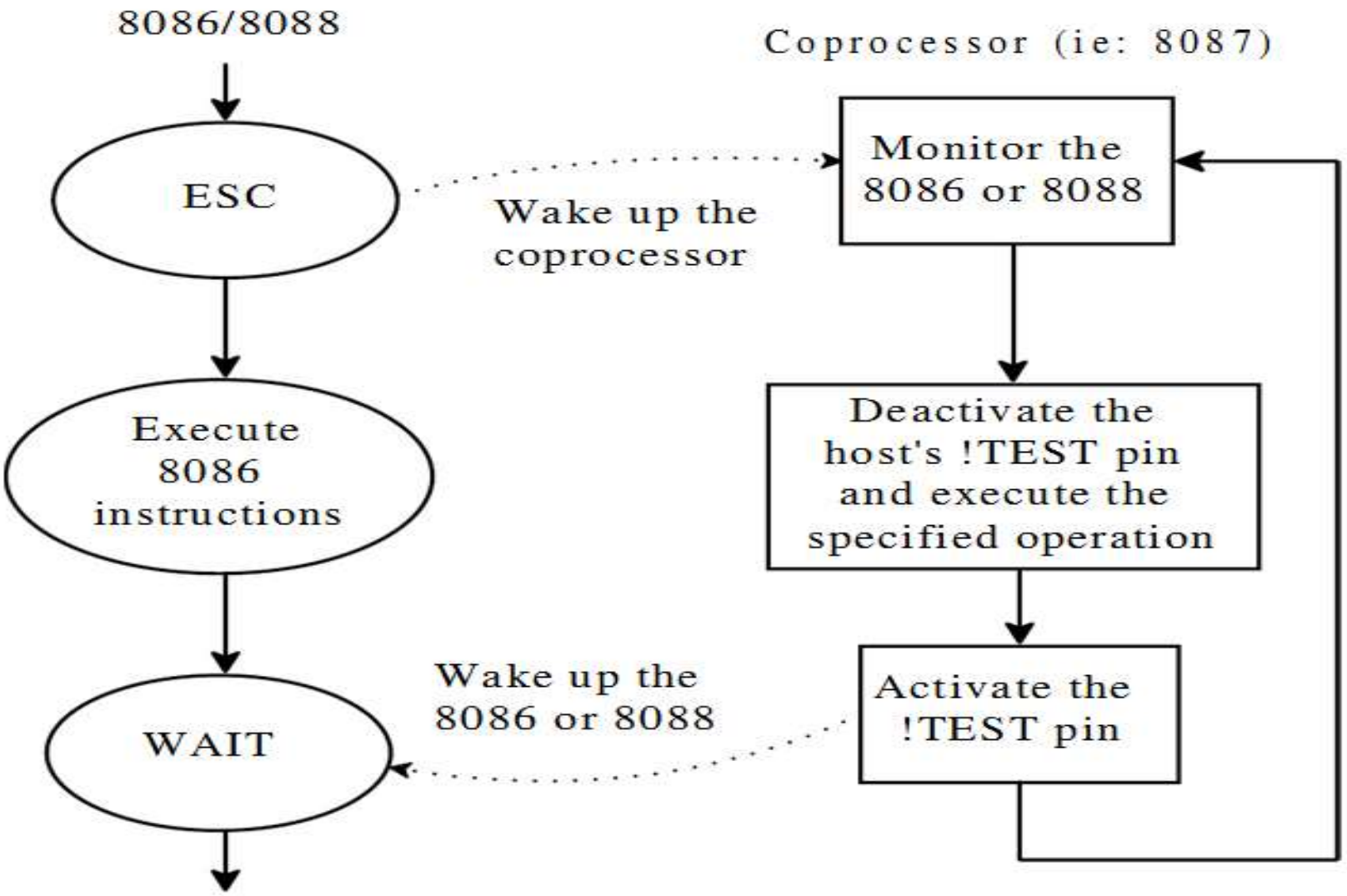


TEST pin of 8086

- Used in conjunction with the WAIT instruction in multiprocessing environments.
- This is input from the 8087 coprocessor.
- During execution of a wait instruction, the CPU checks this signal.
- If it is low, execution of the signal will continue; if not, it will stop executing.

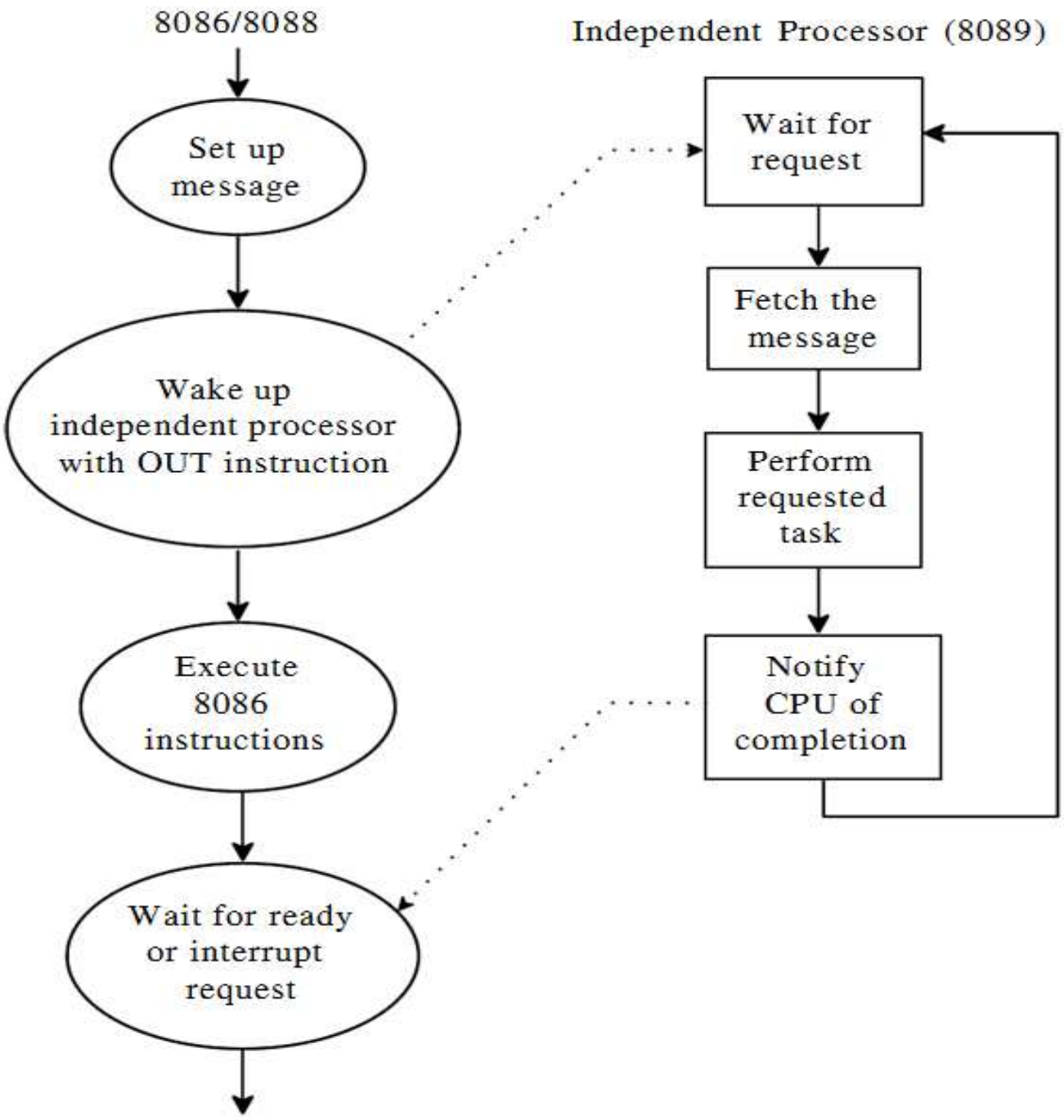
Coprocessor Execution Example

Coprocessor cannot take control of the bus, it does everything through the CPU



Closely Coupled Execution Example

- **Closely Coupled processor may take control of the bus independently.**
- **Two 8086's cannot be closely coupled.**



Loosely Coupled Configuration

- has **shared system bus, system memory, and system I/O**.
- each processor has **its own clock** as well as **its own memory** (in addition to access to the system resources).
- Used for medium to **large multiprocessor systems**.
- Each module is capable of being the **bus master**.
- Any module could be a processor capable of being a bus master, a coprocessor configuration or a closely coupled configuration.

Loosely Coupled Configuration

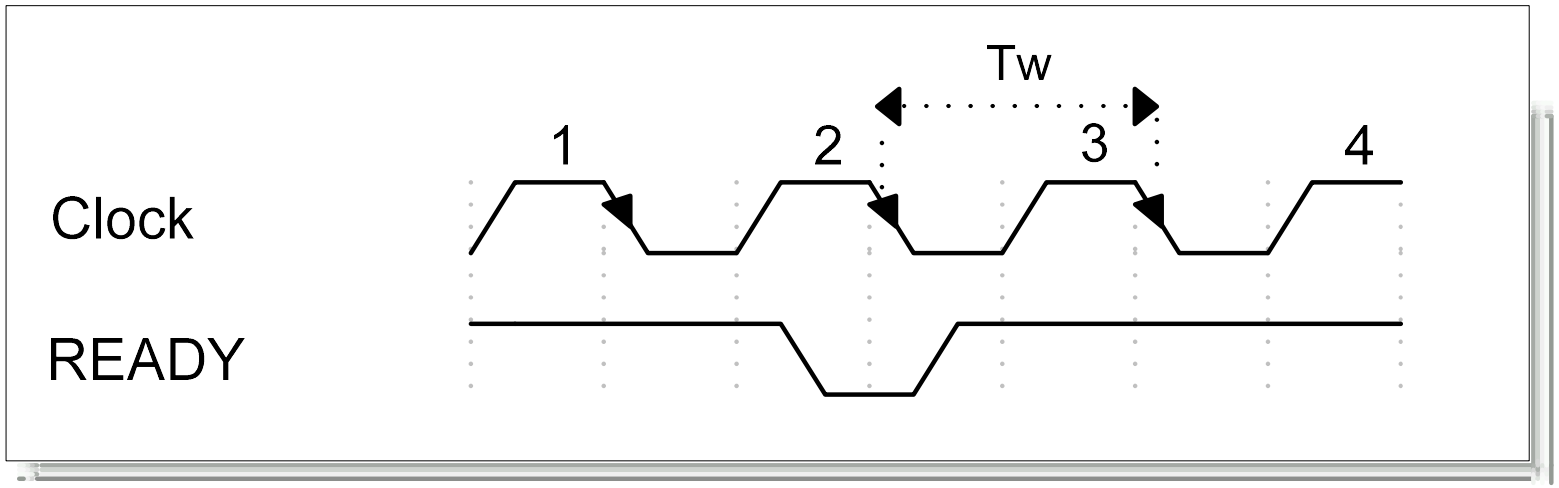
- **No direct connections** between the modules.
- Each share the system bus and **communicate through shared resources.**
- Processor in their separate modules can simultaneously access their private subsystems through their local busses, and perform their local data references and instruction fetches independently. This results in **improved degree of concurrent processing.**
- **Excellent** for real time applications, as separate modules can be assigned specialized tasks

Advantages of Multiprocessor Configuration

1. High **system throughput** can be achieved by having more than one CPU.
2. The system can be **expanded** in modular form.

   Each bus master module is an independent unit and normally resides on a separate PC board. One can be added or removed without affecting the others in the system.
3. A **failure** in one module normally does not affect the breakdown of the entire system and the faulty module can be easily detected and replaced
4. Each bus master has its own local bus to access dedicated memory or IO devices. So a greater **degree of parallel processing** can be achieved.

WAIT State



- A **wait state (T_w)** is an extra clocking period, inserted between **T2** and **T3**, to lengthen the bus cycle, allowing slower memory and I/O components to respond.

- The **READY** input is sampled at the end of **T2**, and again, if necessary in the middle of Tw. **If READY is '0' then a Tw is inserted.**

Summary

- Learnt the pin diagram of 8086 microprocessor
- Understood the minimum/maximum mode of 8086

Books

1. Ian R Sinclair, Sensors and Transducers, Newnes publishers, Third Edition, 2001.
2. Ramon Pallás Areny, John G. Webster, Sensors and Signal conditioning, John Wiley and Sons, Second Edition, 2000.