Microprocessor

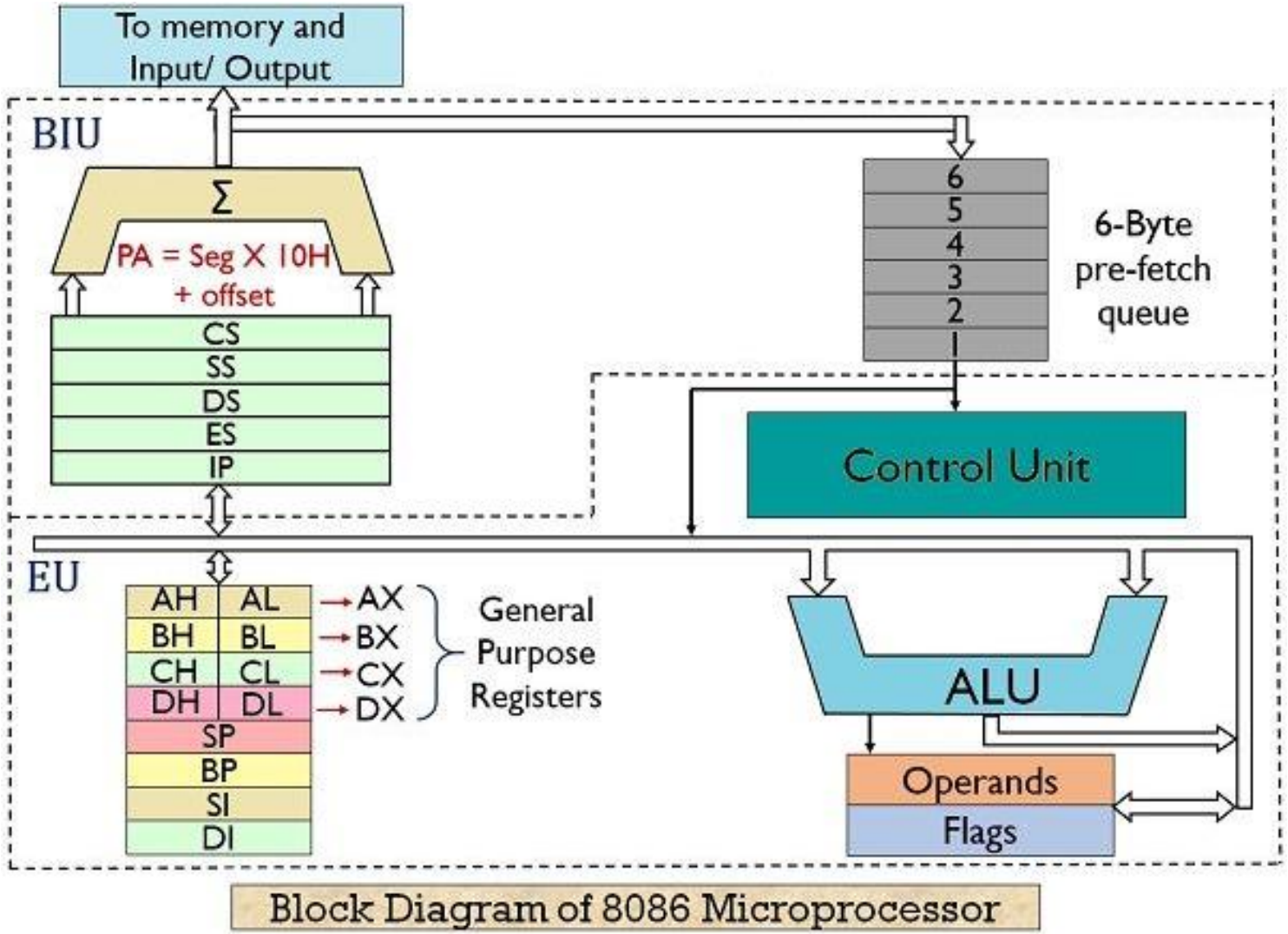
Unit 6:
8086, 80286, 80386

Course Outline

- 8086 Microprocessor
  - Logical Block Diagram
  - Segment Registers,
  - Memory Segmentation
  - Bus Interface Unit and Execution Unit
  - Pipelining
- 80286:
  - Architecture (Block Diagram) ,
  - Registers,
  - (Real/Protected mode),
  - Privilege Levels,
  - Descriptor Cache,
  - Memory Access in GDT and LDT,
  - Multitasking,
  - Addressing Modes,
  - Flag Register
- 80386:
  - Architecture (Block Diagram),
  - Register organization,
  - Memory Access in Protected Mode,
  - Paging (Up to LA to PA)

8086

8086



Block Diagram of 8086 Microprocessor

Bus Interface Unit (BIU)

Bus Interface Unit (BIU)

- BIU takes care of all data and addresses transfers on the buses for the EU
  - like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory.
- EU has no direction connection with System Buses so this is possible with the BIU.
- EU and BIU are connected with the Internal Bus.
- Function:
  - Sends out addresses
  - Fetches instructions from memory.
  - Read / write data from/to ports and memory i.e. handles all transfers of data and addresses on the busses

6 Byte Pre-fetch Queue

- BIU contains the instruction queue.
- BIU gets upto 6 bytes of next instructions and stores them in the instruction queue.
- When EU executes instructions and is ready for its next instruction, then it simply reads the instruction from this instruction queue resulting in increased execution speed.
- Fetching the next instruction while the current instruction executes is called **pipelining**.
- Gets flushed whenever a branch instruction occurs.

Segment Register

- BIU has 4 segment buses, i.e. CS, DS, SS& ES.
- It holds the addresses of instructions and data in memory, which are used by the processor to access memory locations.
- It also contains 1 pointer register IP, which holds the address of the next instruction to executed by the EU.
  - **CS** – It stands for Code Segment. It is used for addressing a memory location in the code segment of the memory, where the executable program is stored.
  - **DS** – It stands for Data Segment. It consists of data used by the program andis accessed in the data segment by an offset address or the content of other register that holds the offset address.
  - **SS** – It stands for Stack Segment. It handles memory to store data and addresses during execution.
  - **ES** – It stands for Extra Segment. ES is additional data segment, which is used by the string to hold the extra destination data.

Instruction Register (IR)

- It is a 16 bit register.
- It holds offset of the next instructions in the Code Segment.
- IP is incremented after every instruction byte is fetched.
- IP gets a new value whenever a branch instruction occurs.

Address Generation Circuit

- The BIU has a Physical Address Generation Circuit.
- It generates the 20 bit physical address using Segment and Offset addresses using the formula:

```
Physical Address  = Segment Address x 10H + Offset Address
```

Execution Unit (EU)

Execution Unit

- Execution unit gives instructions to BIU stating from where to fetch the data and then decode and execute those instructions.
- Its function is to control operations on data using the instruction decoder & ALU.
- EU has no direct connection with system buses, it performs operations over data through BIU.
  - Tells BIU where to fetch instructions or data from
  - Decodes instructions
  - Executes instructions

Control Unit

- Like the timing and control unit in 8085 microprocessor, the control unit in 8086 microprocessor produces control signal after decoding the opcode to inform the general purpose register to release the value stored in it.
- And it also signals the ALU to perform the desired operation.

Instruction Decoder & ALU

- Decoder in the EU translates instructions fetched from the memory into a series of actions which the EU carries out.
- 16-bit ALU in the EU performs actions such as AND, OR, XOR, increment, decrement etc.

General purpose Registers

- 8 GPRs AH, AL (Accumulator), BH, BL, CH, CL, DH, DL are used to store 8 bit data.
- Used individually for the temporary storage of data
- These can be used together (as register pair) to store 16-bit data words. Acceptable register pairs are:
  - **AH-AL** pair AX register (also known as accumulator register & used to store operands for arithmetic operations.)
  - **BH-BL** pair BX register (to store the 16-bit data as well as the base address of the memory location)
  - **CH-CL** pair CX register (to store 16-bit data and can be used as counter register for some instructions like loop)
  - **DH-DL** pair DX register (to store 16-bit data and also used to hold the result of 16-bit data multiplication and division operation)

Pointer registers: SP (Stack Pointer), BP (Base pointer)

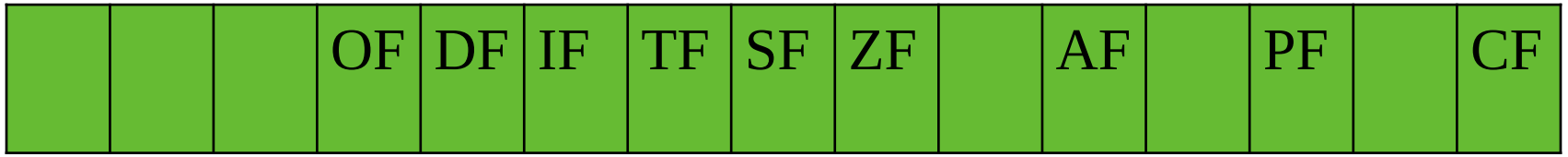
- The two pointer registers, SP and BP are used to access data in the stack segment.
- The **SP** is used as offset from current Stack Segment during execution of instruction that involve stack.
  - SP is automatically updated.
- **BP** contains offset address and is utilized in based addressing mode.
  - Overall, these are used to hold the offset address of the stack address.

Index registers: SI (Source Index), DI (Destination index)

- EU also contains a 16-bit source index (SI) register and 16-bit destination index (DI) register.
- These registers can be used for temporary storage of data similarly as the general purpose registers.
- However they are specially to hold the 16-bit offset of the data word.
- **SI** holds offset address in Data Segment during string operations
- **DI** holds offset address in Extra Segment during string operations.

FLAG Register

- It is a 16-bit register. 9-bit are used as different flags, remaining bits unused.



- Out of 9-flags, 6 are conditional (status) flags and three are control flags

Conditional flags

- These are set or reset by the EU on the basis of the results of some arithmetic or logic operation.
- 8086 instructions check these flags to determine which of two alternative actions should be done in executing the instructions.
  - **OF (Overflow flag)**: is set if there is an arithmetic overflow, i.e. the size of the result exceeds the capacity of the destination location.
  - **SF (Sign flag)**: is set if the MSB of the result is 1
  - **ZF (Zero flag)**: is set if the result is zero
  - **AF (Auxiliary carry flag)**: is set if there is carry from lower nibble to upper nibble or from lower byte to upper byte
  - **PF (Parity flag)**: is set if the result has even parity
  - **CF (Carry flag)**: is set if there is carry from addition or borrow from subtraction

Control flags

- They are set using certain instructions. They are used to control certain operations of the processor.
  - **TF (Trap flag):** for single stepping through the program
  - **IF (Interrupt flag):** to allow or prohibit the interruption of a program
  - **DF (Direction flag):** Used with string instructions

Operand

- It is a temporary register and is used by the processor to hold the temporary values at the time of operation.
- The reason behind two separate sections for BIU and EU in the architecture of 8086 is to perform fetching and decoding-executing simultaneously.

Pipelining

Pipelining

- The process of fetching the next instruction when the present instruction is being executed is called as pipelining.
- Pipelining has become possible due to the use of queue.
- BIU (Bus Interfacing Unit) fills in the queue until the entire queue is full.
- BIU restarts filling in the queue when at least two locations of queue are vacant.

Advantages of pipelining

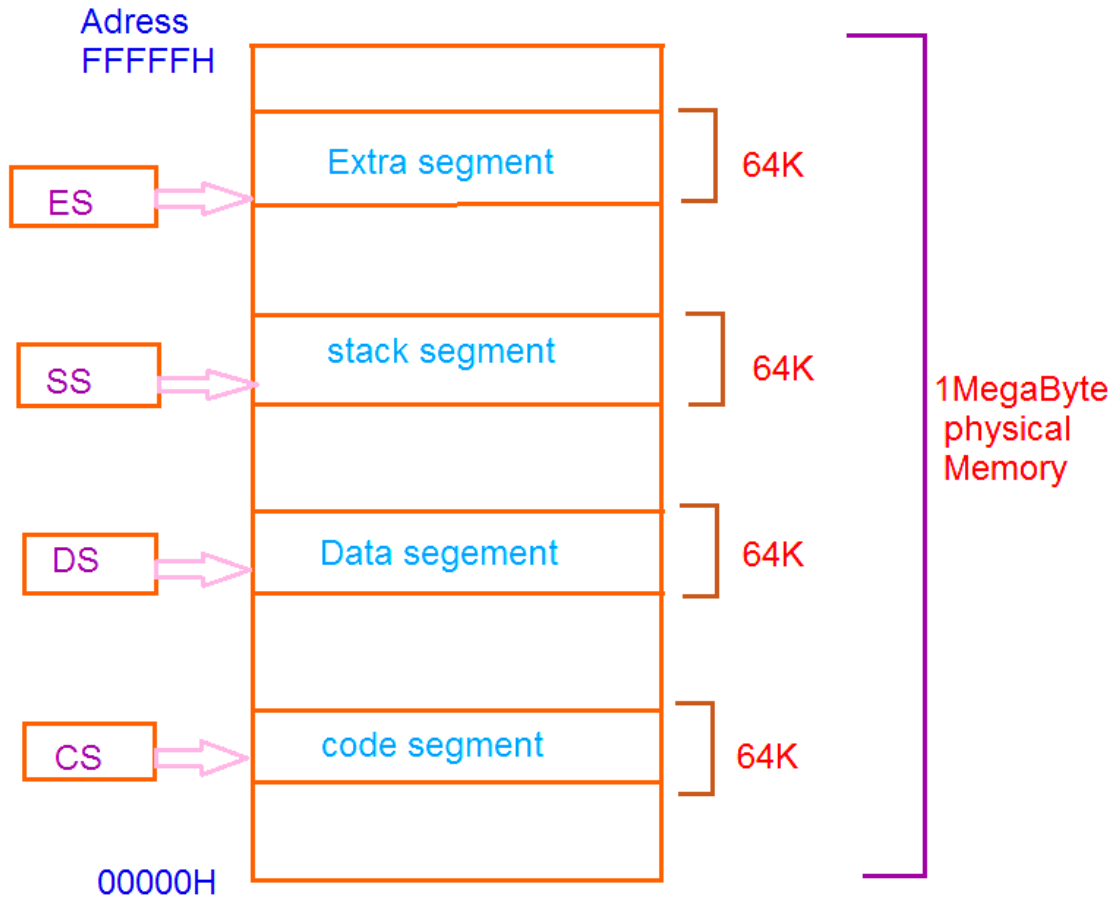
- The execution unit always reads the next instruction byte from the queue in BIU.
- This is faster than sending out an address to the memory and waiting for the next instruction byte to come.
- In short pipelining eliminates the waiting time of EU and speeds up the processing.
  - The 8086 BIU will not initiate a fetch unless and until there are two empty bytes in its queue.
  - 8086 BIU normally obtains two instruction bytes per fetch.
- *Final catch:* ***Pipelining increases the overall performance of the processor****.*

Memory Segmentation

Memory segmentation

- To increase execution speed and fetching speed, 8086 segments the memory.
- It’s 20 bit address bus can address 1MB of memory, it segments it into 16 64kB segments.
- 8086 works only with four 64KB segments within the whole 1MB memory.
- So the four segments will store 256 KB of memory locations.
- The remaining memory locations are free and in these locations, user can perform any other processes.

Memory segmentation



Memory segmentation: Types

- **Overlapping Segment**
  - A segment starts at a particular address and its maximum size can go up to 64kilobytes.
  - But if another segment starts along with this 64kilobytes location of the first segment, then the two are said to be *Overlapping Segment*.
- **Non-Overlapped Segment**
  - A segment starts at a particular address and its maximum size can go up to 64kilobytes.
  - But if another segment starts before this 64kilobytes location of the first segment, then the two segments are said to be *Non-Overlapped Segment*.

Memory segmentation: Rules

1. The starting address of a segment should be such that it can be evenly divided by 16.
2. Minimum size of a segment can be 16 bytes and the maximum can be 64 kB.

Advantages of Memory segmentation

- It allows the memory addressing capacity to be 1 Mbyte even though the address associated with individual instruction is only 16-bit.
- It allows instruction code, data, stack, and portion of program to be more than 64 KB long by using more than one code, data, stack segment, and extra segment.
- It facilitates use of separate memory areas for program, data and stack.
- It permits a program or its data to be put in different areas of memory, each time the program is executed i.e. program can be relocated which is very useful in multiprogramming.

Addressing Modes

Addressing Modes in 8086

- The way of specifying data to be operated by an instruction is known as **addressing modes**.
- This specifies that the given data is an immediate data or an address.
- It also specifies whether the given operand is register or register pair.
- 8 Different Types:
  1. Immediate Addressing Mode
  2. Register Addressing Mode
  3. Direct Addressing Mode
  4. Register Indirect Addressing Mode
  5. Based Addressing Mode
  6. Indexed Addressing Mode
  7. Based-index Addressing Mode
  8. Based-Indexed with Displacement Addressing Mode

Addressing Modes in 8086

1. **Immediate Addressing Mode**
   - The addressing mode in which the data operand is a part of the instruction itself is known as immediate addressing mode.
   - `MOV AX, 4231H,`
   - `ADD CX, 4567H,`
   - `MOV AL, FFH`
2. **Register Addressing Mode**
   - In this type of addressing mode both the operands are registers.
   - `MOV AX, BX,`
   - `ADD AX, DX`

Addressing Modes in 8086

3. Direct Addressing Mode

- In this type of addressing mode the effective address is directly given in the instruction as displacement.
- `MOV AX, [1234H],`
- `MOV AL, [4567H]`

4. Register Indirect Addressing Mode

- This addressing mode allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI & SI.
- `MOV AX, [DI]`
- `ADD AL, [BX]`
- `MOV AX, [SI]`

Addressing Modes in 8086

5. Based Addressing Mode

- In this addressing mode, the offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.
- `MOV DX, [BX+04],`
- `ADD CL, [BX+08]`

6. Indexed Addressing Mode

- In this addressing mode, the operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.
- `MOV BX, [SI+16],`
- `ADD AL, [DI+16]`

Addressing Modes in 8086

7. Based-Indexed Addressing Mode

- In this addressing mode, the offset address of the operand is computed by summing the base register to the contents of an Index register.
- `ADD CX, [AX+SI],`
- `MOV AX, [AX+DI]`

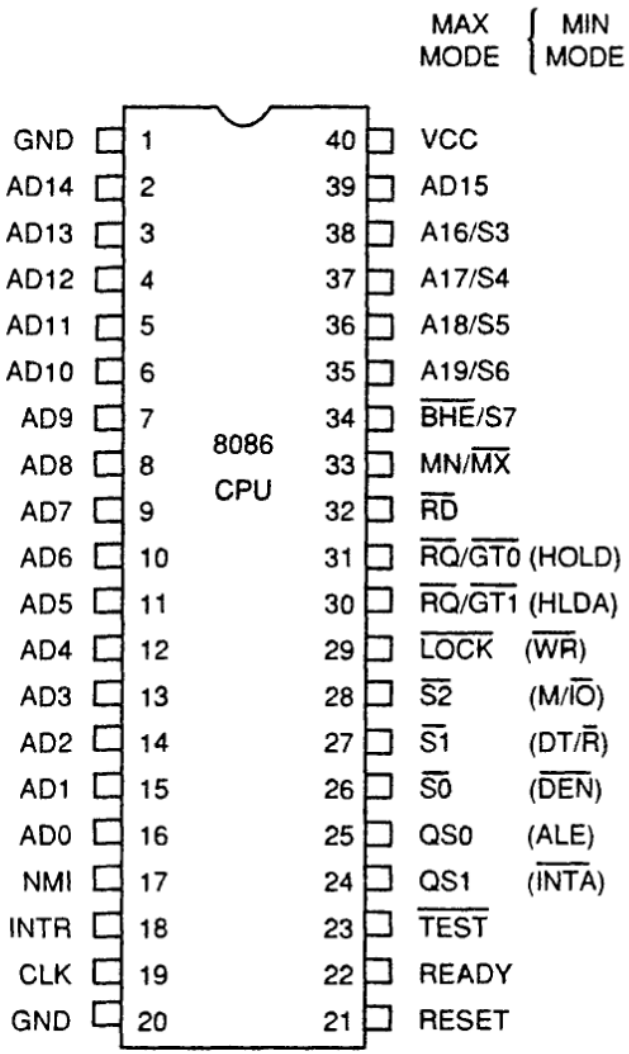
8. Based-Indexed with Displacement Addressing Mode

- In this addressing mode, the operands offset is computed by adding the base register contents. An Index registers contents and 8 or 16-bit displacement.
- `MOV AX, [BX+DI+08],`
- `ADD CX, [BX+SI+16]`

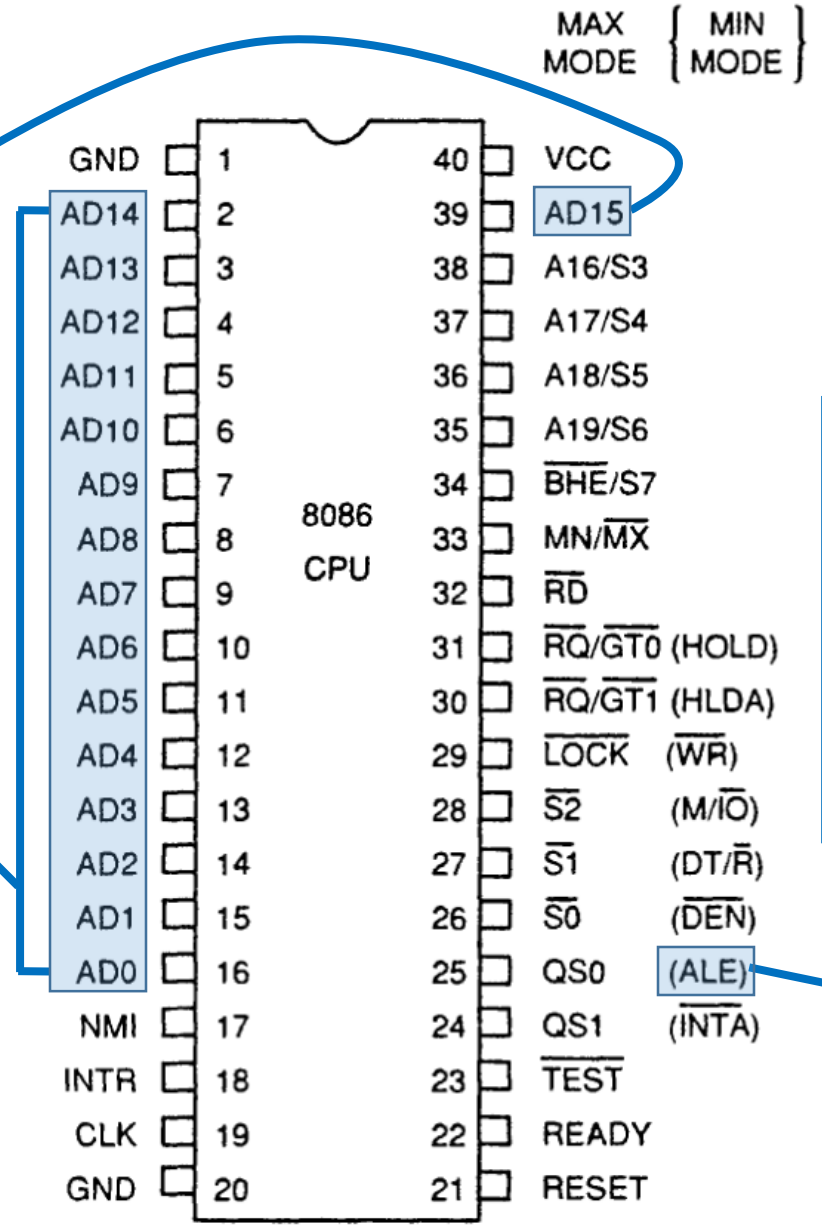
8086

8085 Microprocessor	8086 Microprocessor
It is an 8-bit microprocessor.	It is a 16-bit microprocessor.
It has a 16-bit address line.	It has a 20-bit address line.
It has a 8-bit data bus.	It has a 16-bit data bus.
The memory capacity is 64 KB.	The memory capacity is 1 MB.
The Clock speed of this microprocessor is 3 MHz.	The Clock speed of this microprocessor varies between 5, 8 and 10 MHz for different versions.
It has five flags.	It has nine flags.
8085 microprocessor does not support memory segmentation.	8086 microprocessor supports memory segmentation.
It does not support pipelining.	It supports pipelining.
It is accumulator based processor.	It is general purpose register based processor.
It has no minimum or maximum mode.	It has minimum and maximum modes.
In 8085, only one processor is used.	In 8086, more than one processor is used. An additional external processor can also be employed.
It contains less number of transistors compare to 8086 microprocessor. It contains about 6500 transistor.	It contains more number of transistors compare to 8085 microprocessor. It contains about 29000 in size.
The cost of 8085 is low.	The cost of 8086 is high.

8086: Pin Diagram



8086: Pin Details



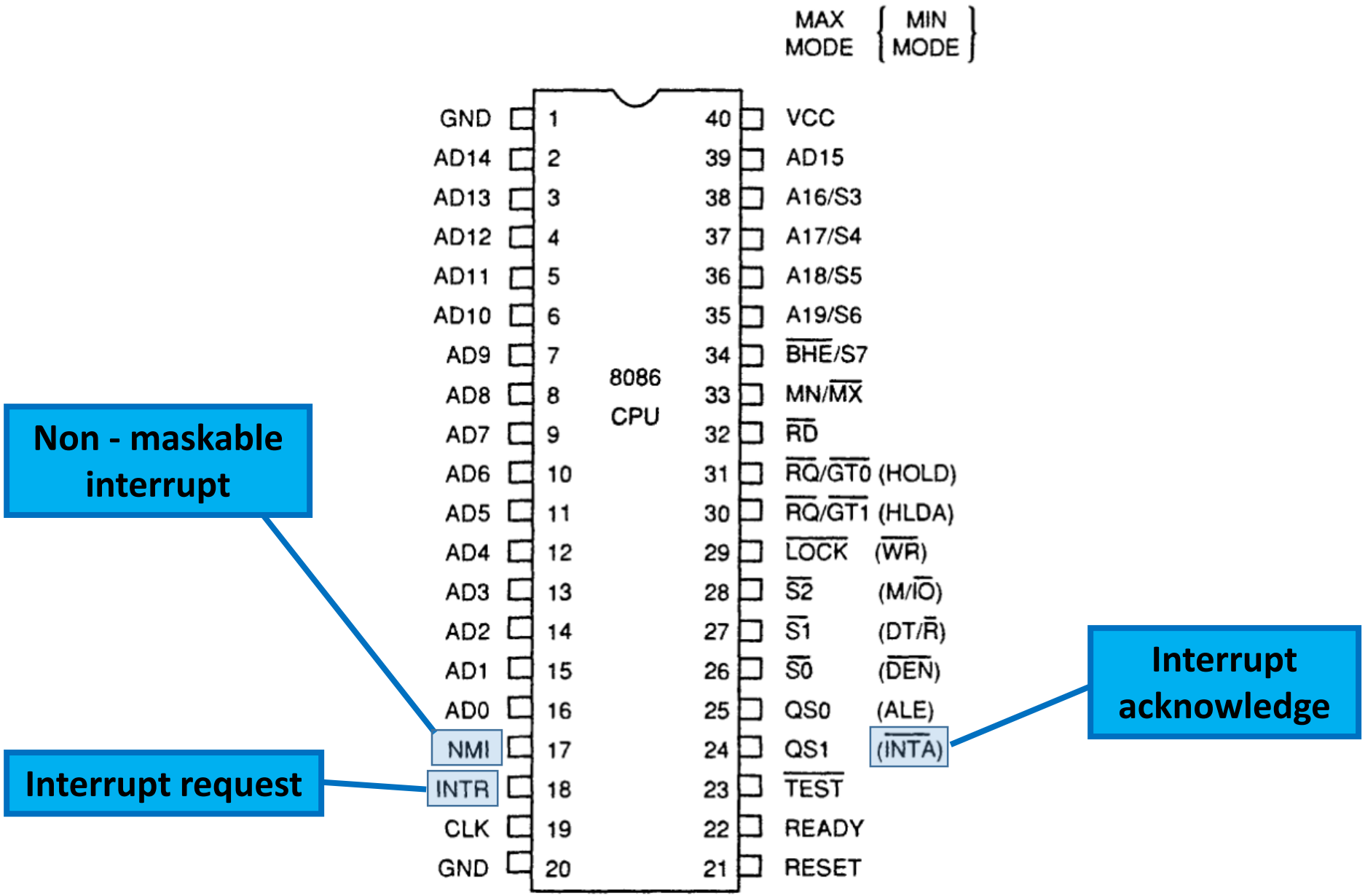
Address/Data Bus:

Contains address bits A_{15}-A_0 when ALE is 1 & data bits D_{15} – D_0 when ALE is 0.

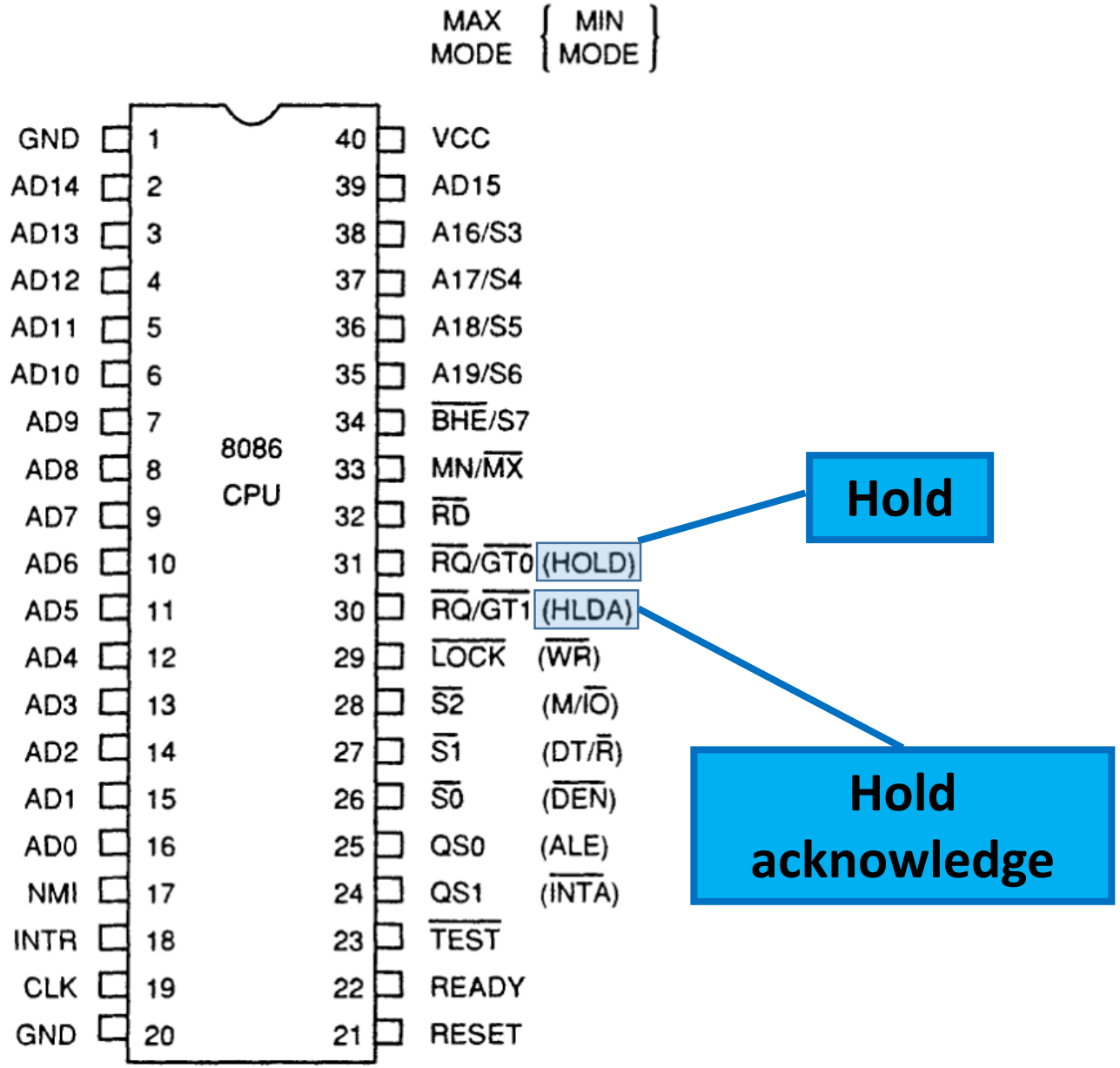
Address Latch Enable:

When high, multiplexed address/data bus contains address information.

8086: Pin Details



8086: Pin Details



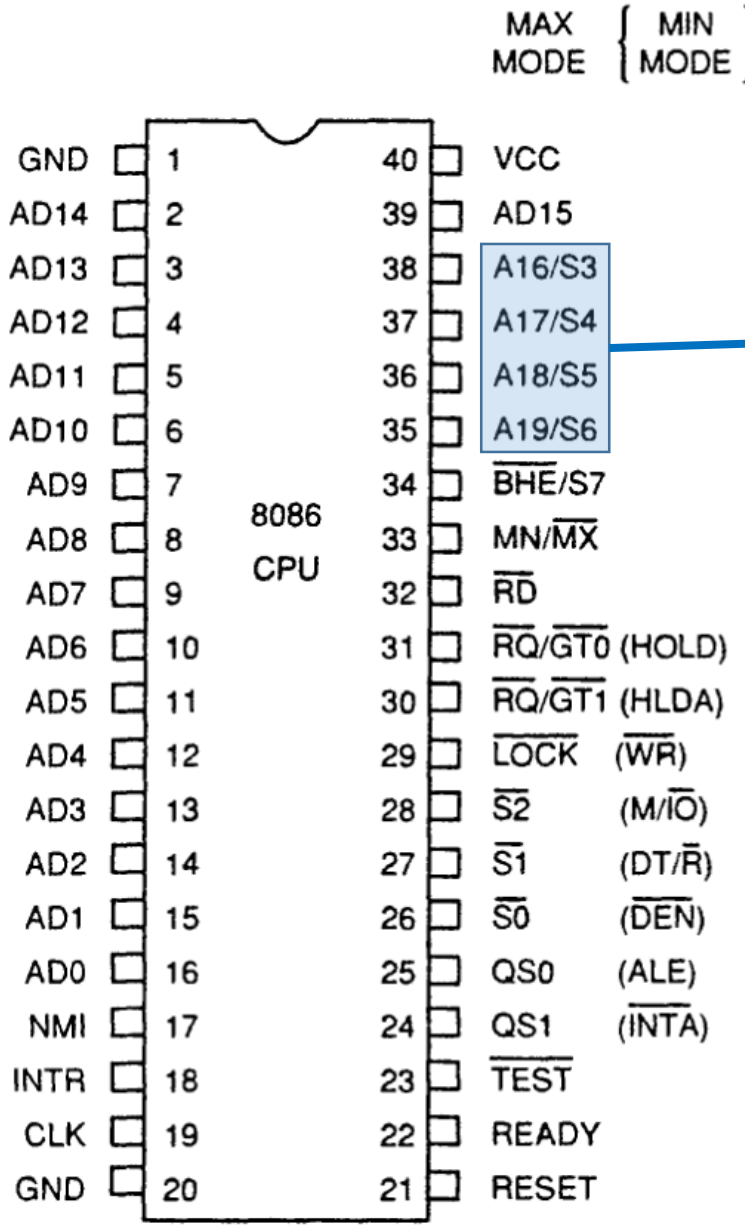
8086: Pin Details

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

8086: Pin Details

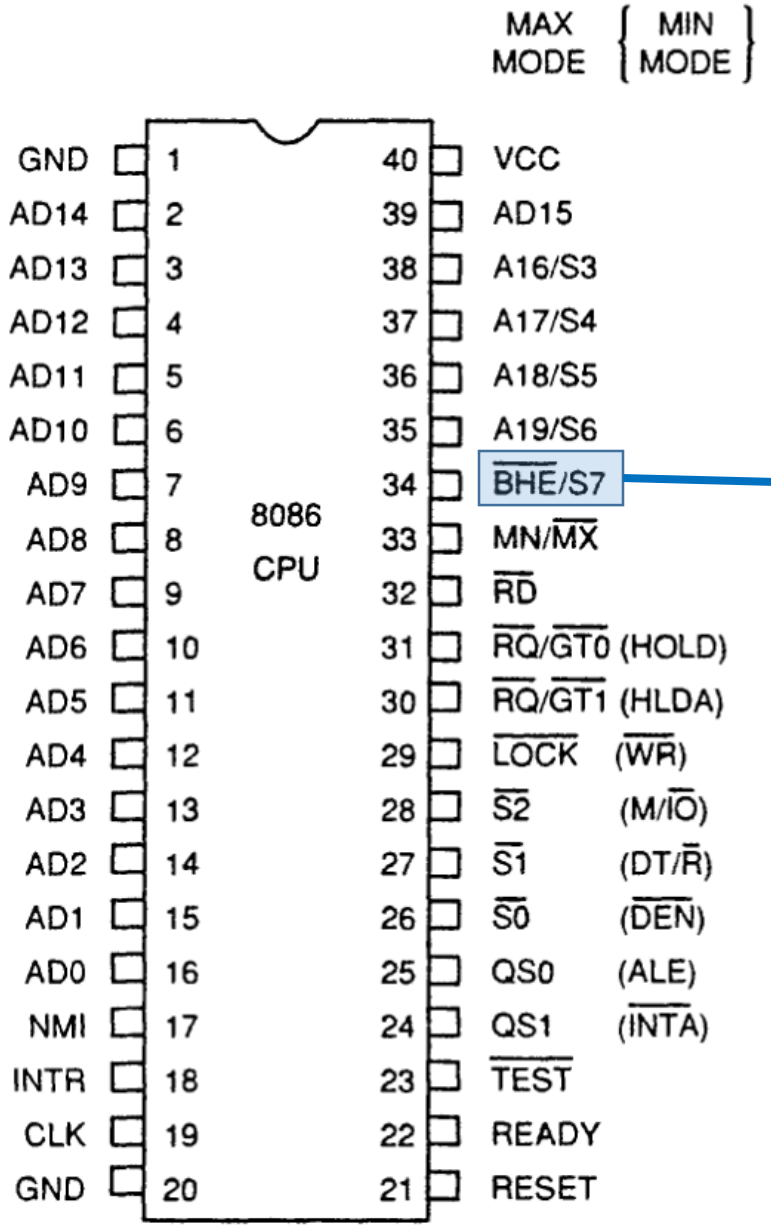
BHE#, A_0:

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

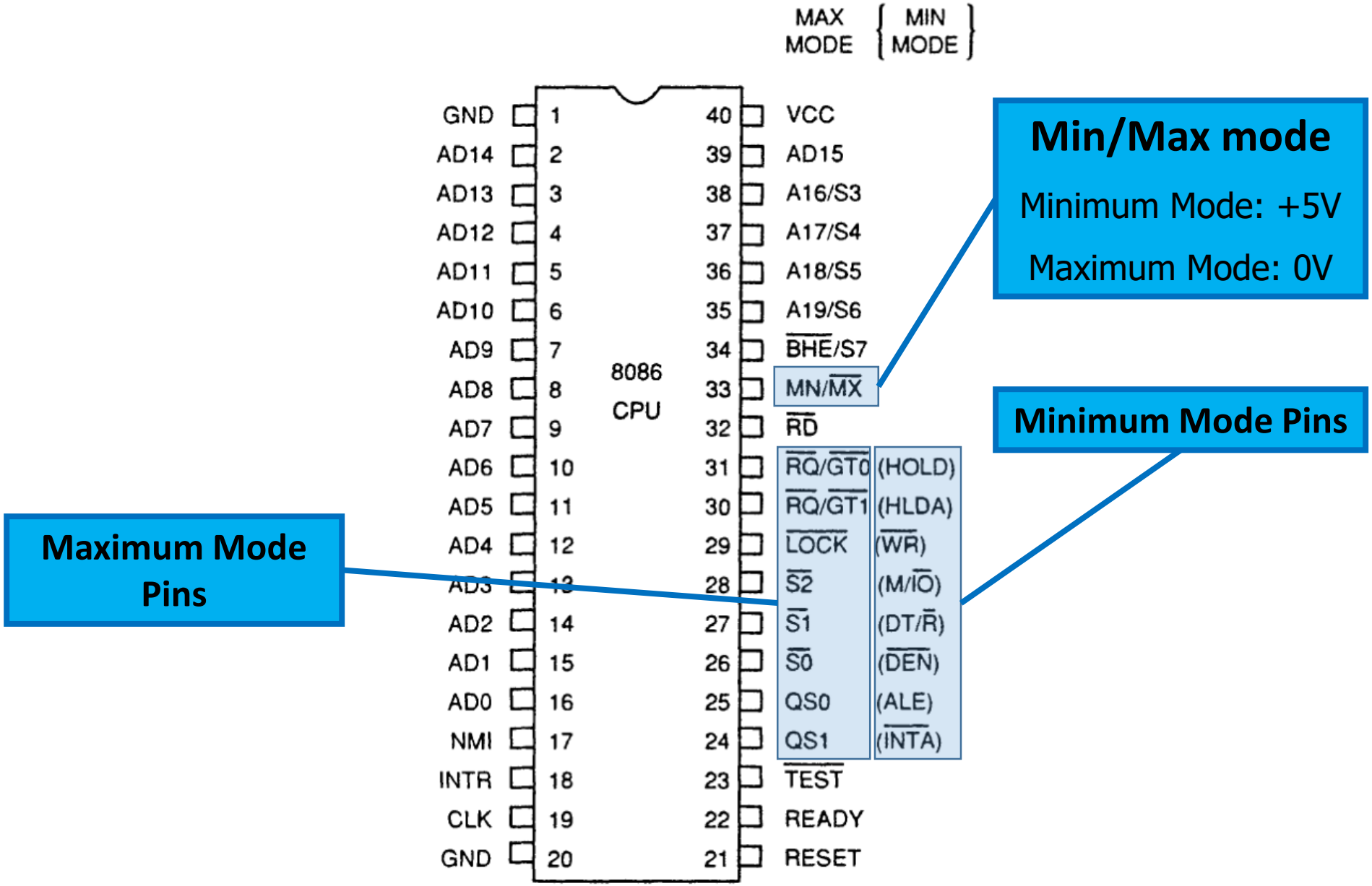


Bus High Enable/S7

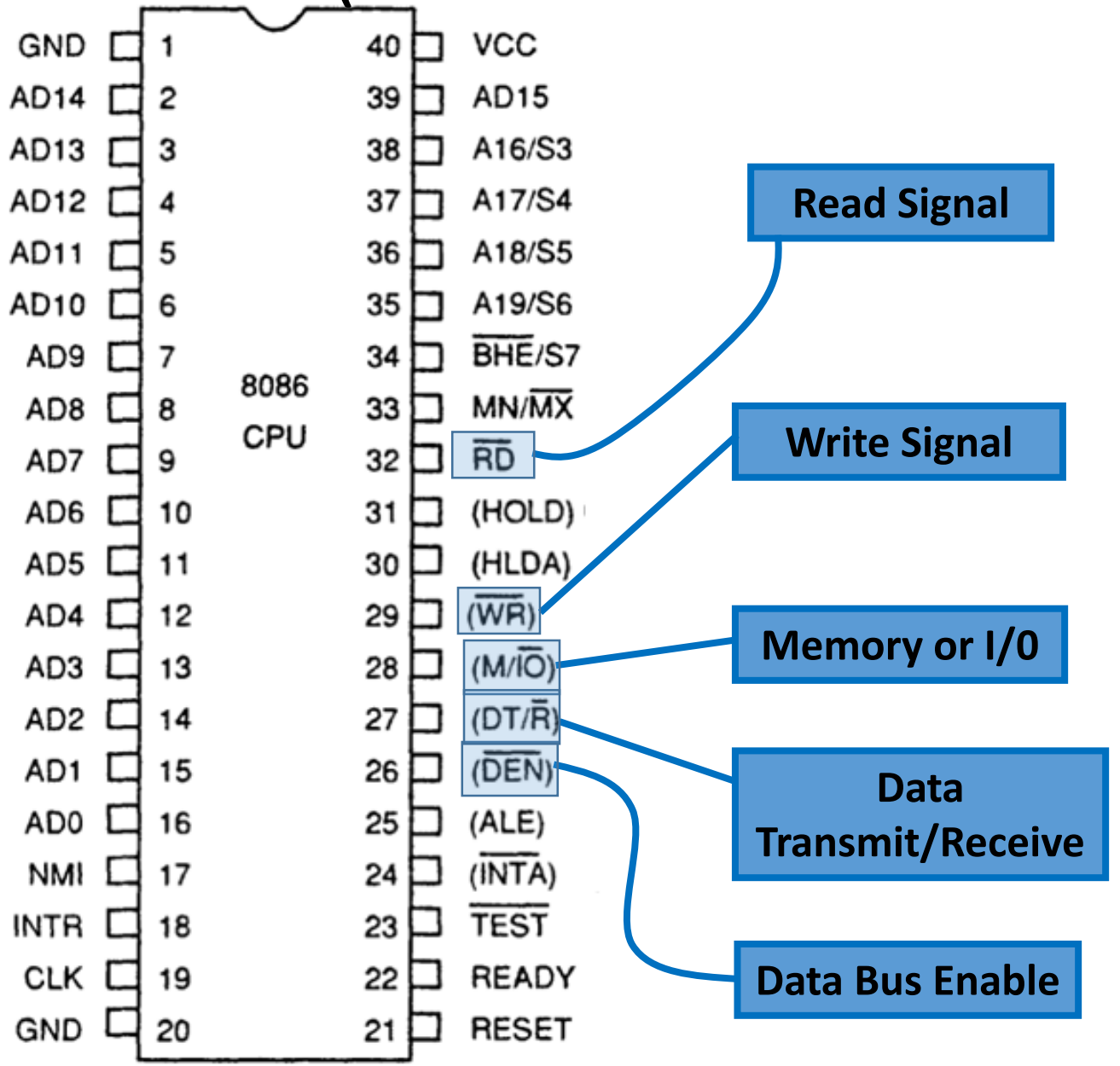
Enables most significant data bits $D_{15} – D_8$ during read or write operation.

S_7: Always 1.

8086: Pin Details



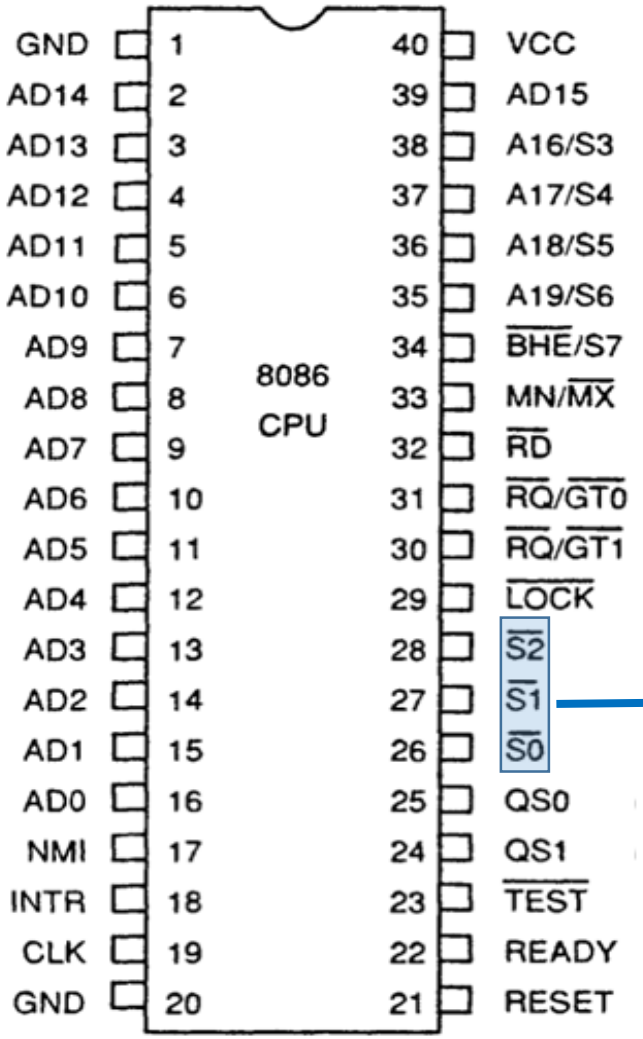
8086: Pin Details (minimum mode)



8086: Pin Details (maximum mode)

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

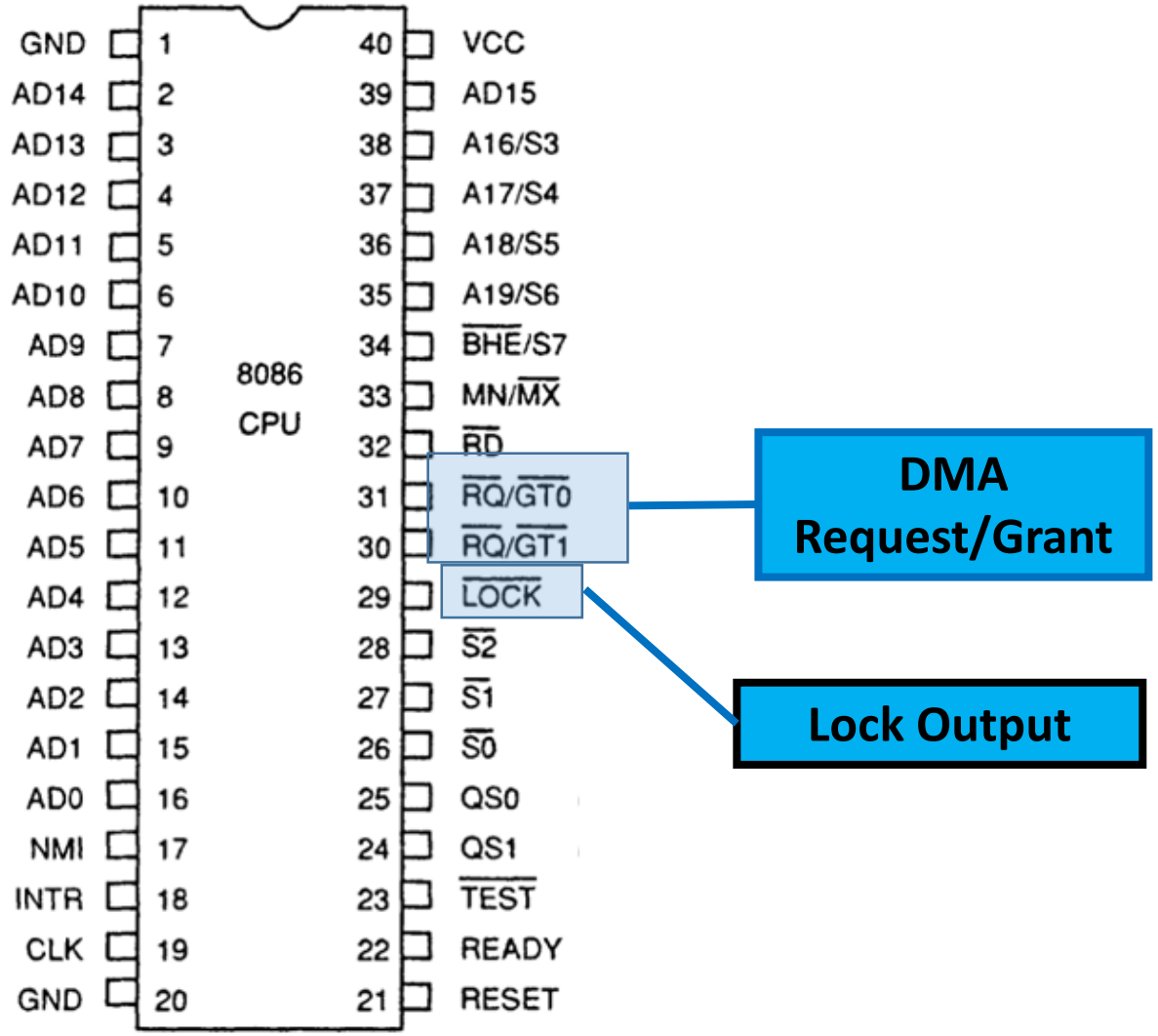
Inputs to 8288 to generate eliminated signals due to max mode.

8086: Pin Details (maximum mode)

Lock Output

Used to lock peripherals off the system

Activated by using the LOCK: prefix on any instruction



8086: Pin Details (maximum mode)

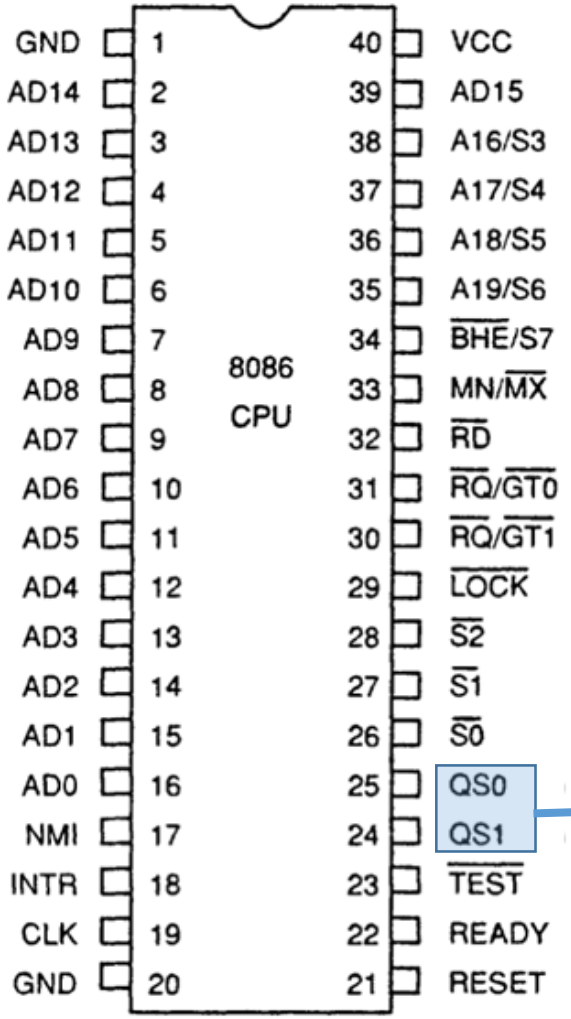
QS1 QS0

00: Queue is idle

01: First byte of opcode

10: Queue is empty

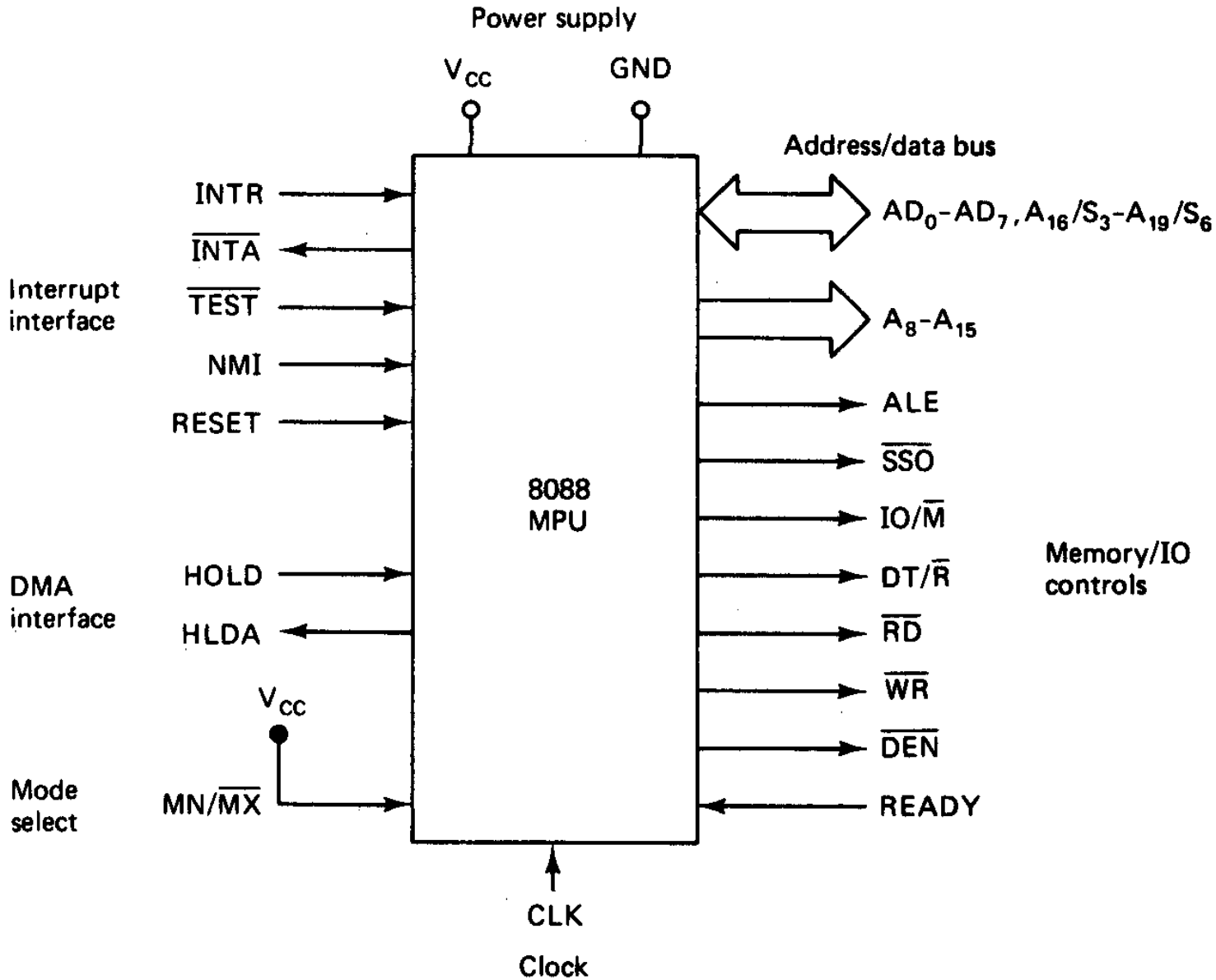
11: Subsequent byte of opcode



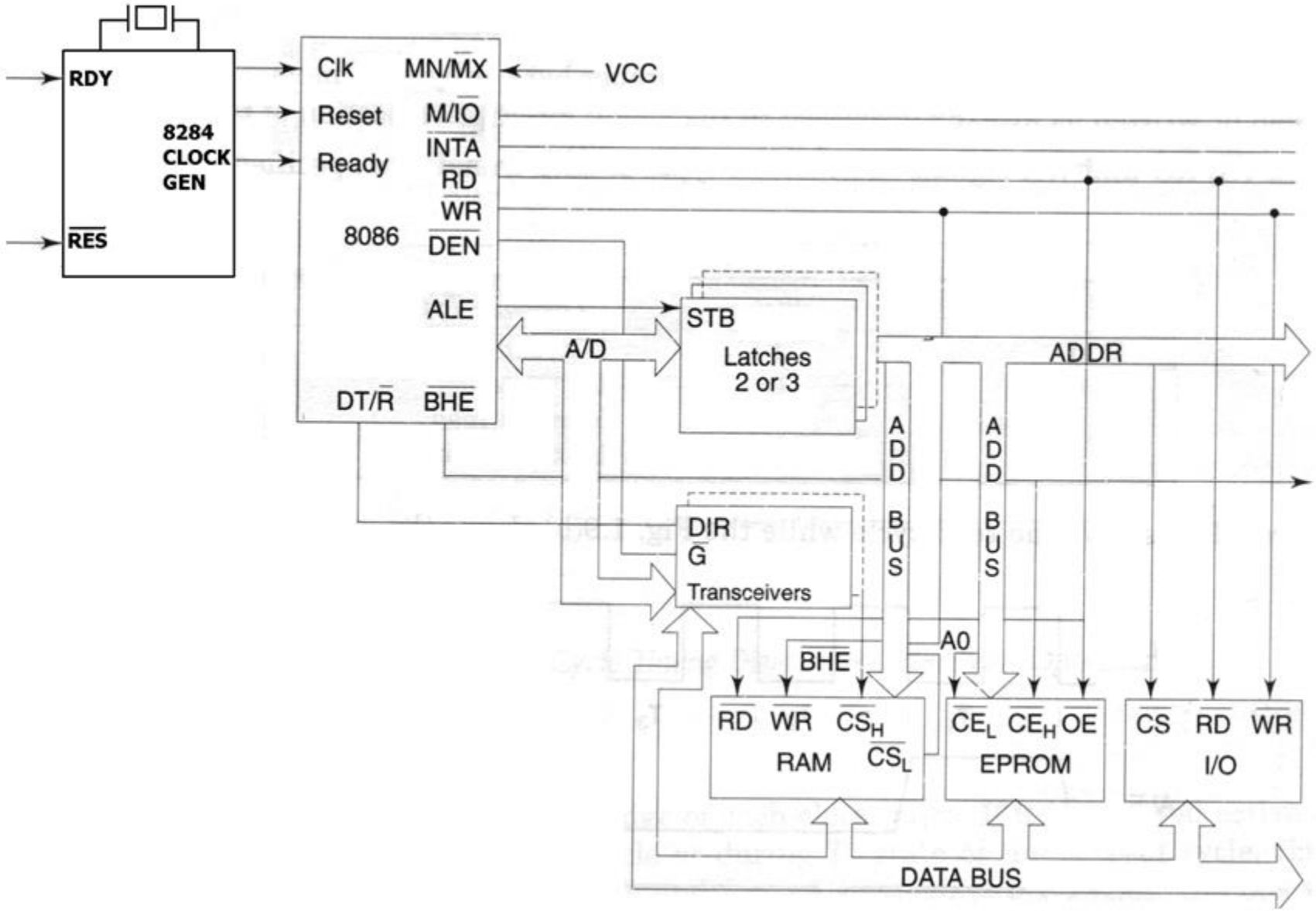
Queue Status

Used by numeric coprocessor (8087)

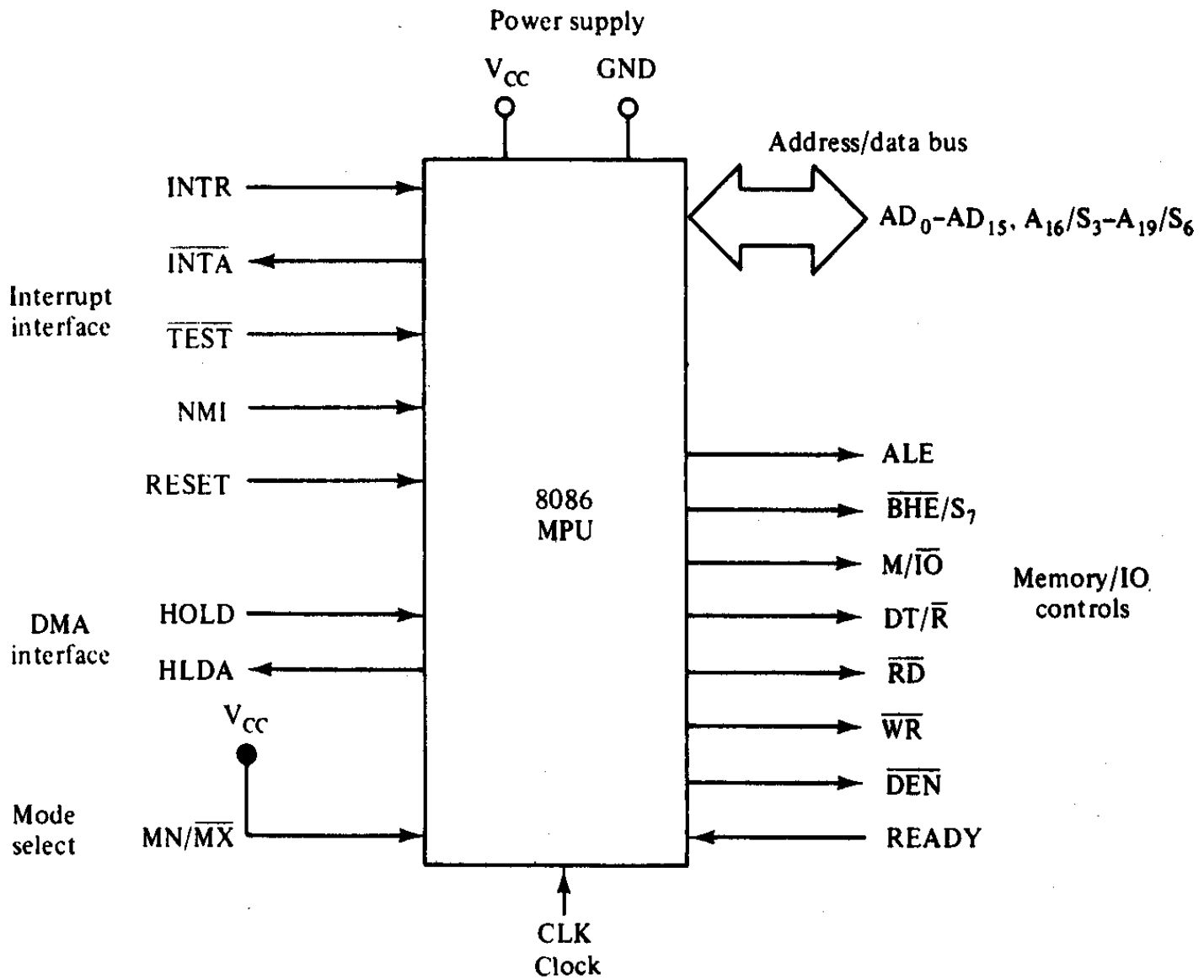
Minimum Mode 8086 System



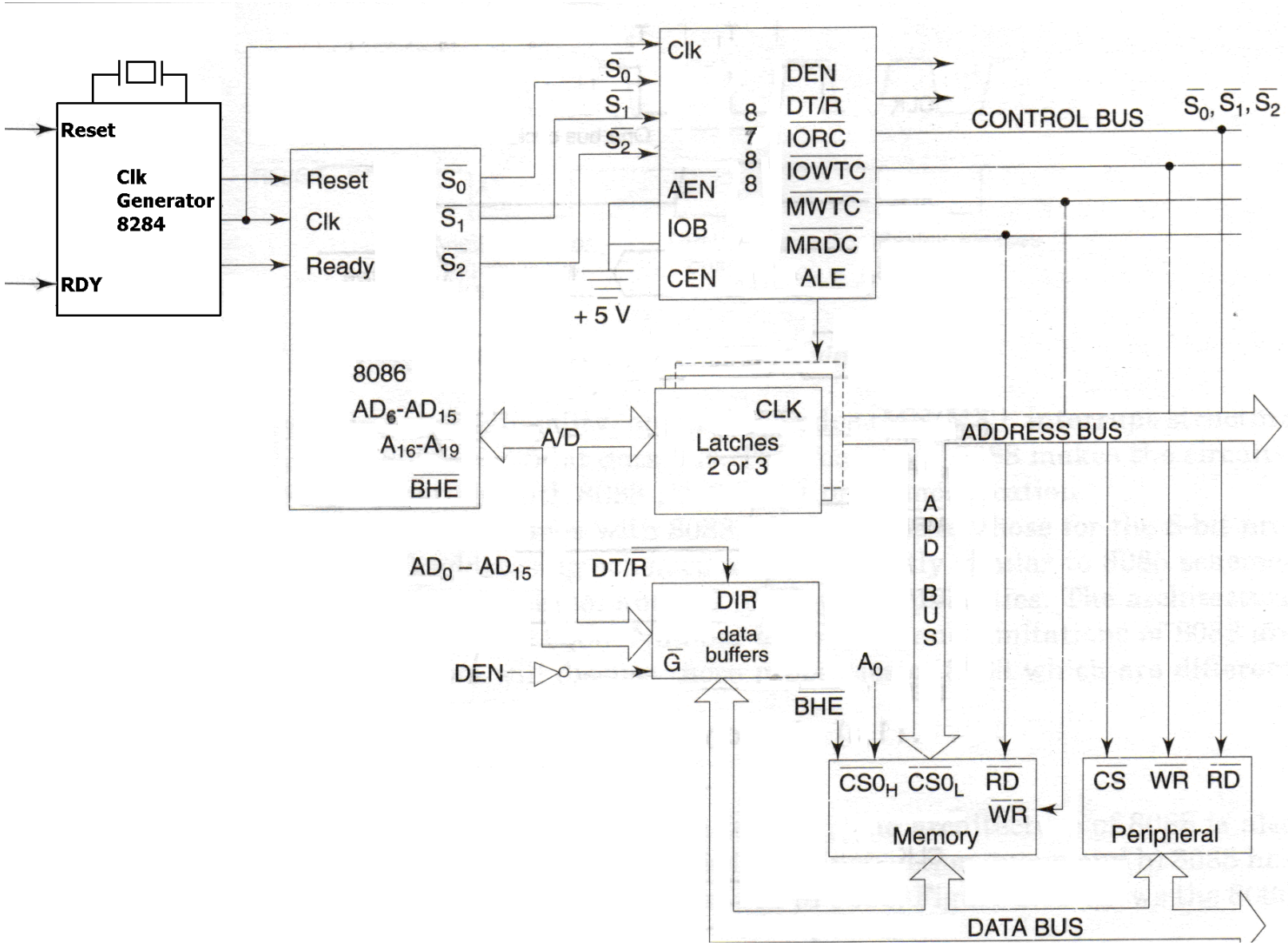
Minimum Mode 8086 System



Maximum Mode 8086 System



Maximum Mode 8086 System



80286

80286

- 80286 Microprocessor is a 16-bit microprocessor
- 80286 is just a modified version of 8086.
- It was invented in February 1982 by Intel.
- High performance microprocessor with **memory management and protection**
  - 80286 is the first member of the family of advanced microprocessors
  - with built-in/on-chip **memory management** and **protection abilities** primarily designed for multi-user/multitasking systems
- It has non-multiplexed data and address bus.
- There is 16-bit data bus and 24-bit address bus -> 16MB memory locations.
- housed in 68-pin package

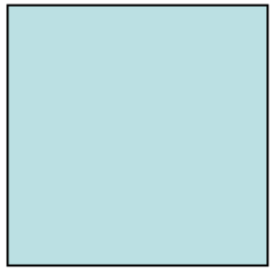
Improvement of 80286

- It has non-multiplexed address and data bus that reduces operational speed.
- The addressable memory in case of 80286 is 16 MB.
- It offers an additional adder for address calculation.
- 80286 has faster multipliers that lead to quick operation.
- The performance per clock cycle of 80286 is almost twice when compared with 8086 or 8088.

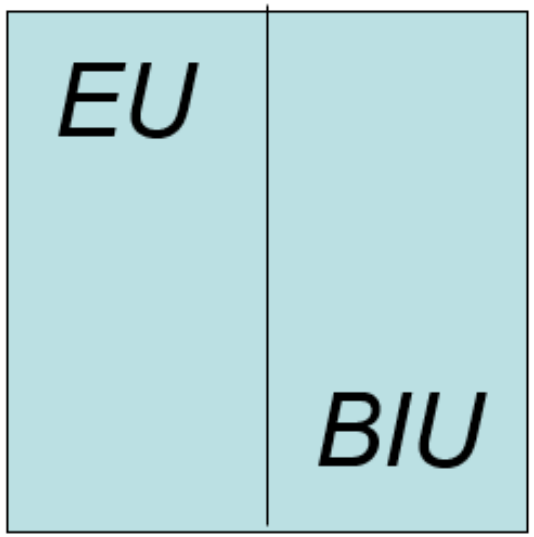
Salient Features of 80286

- Two mode of operation.
- More addressable memory.
- Virtual Memory in Protected Mode.

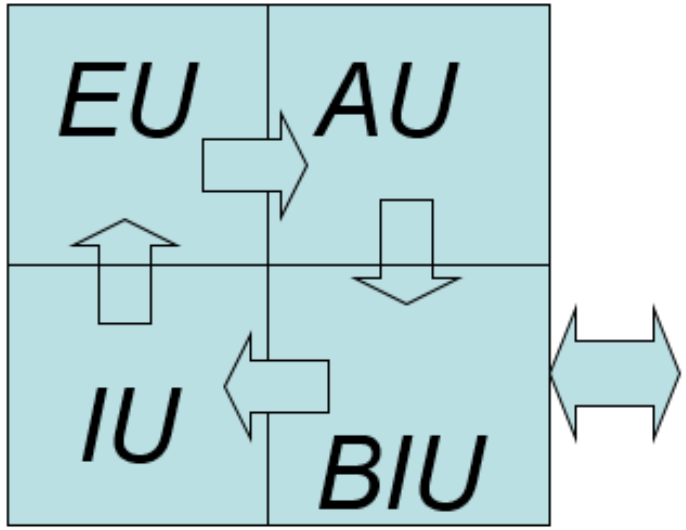
Architectural Sum up



8085

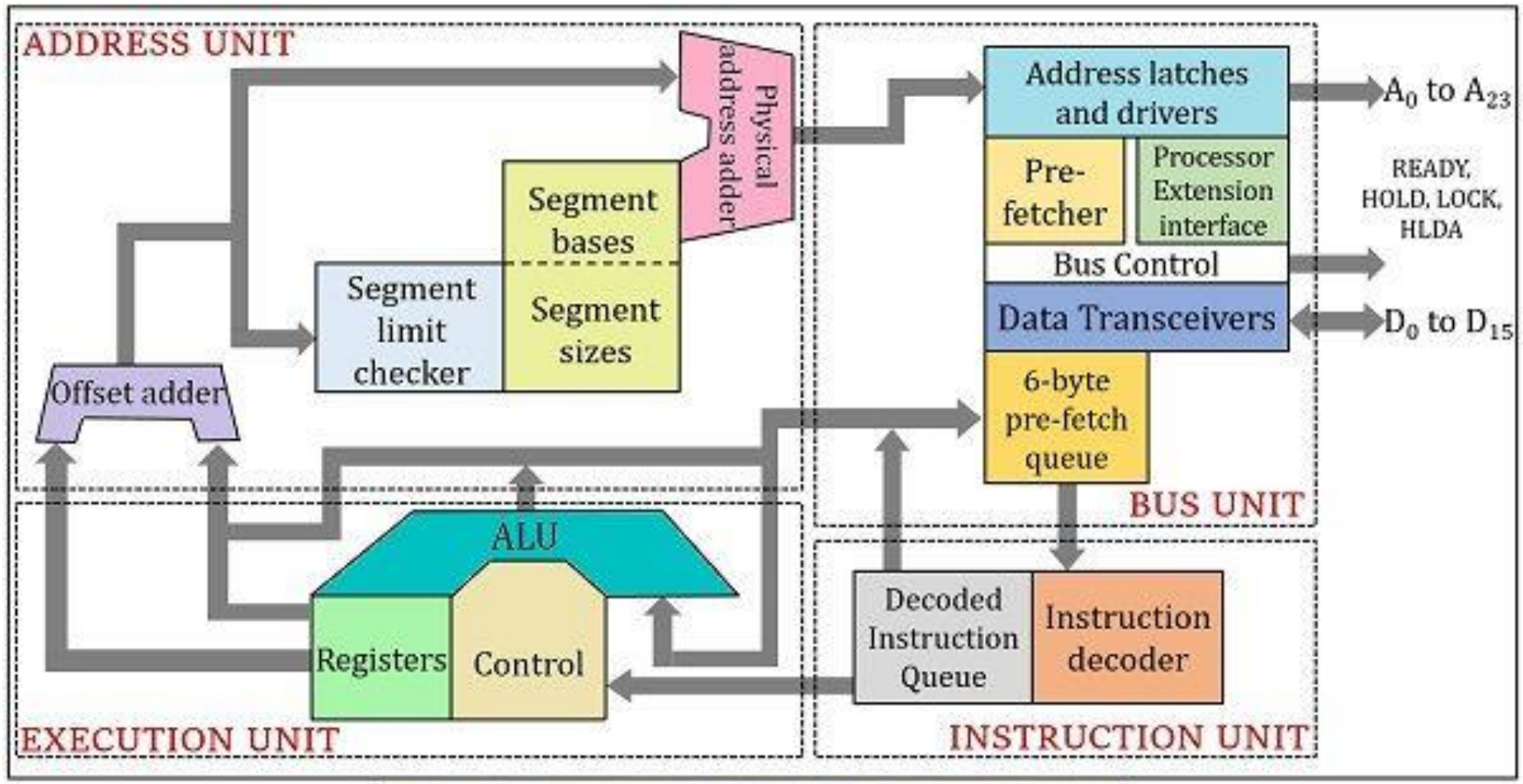


8086



80286

Architecture of 80286



Block Diagram of 80286 Microprocessor

Internal Block Diagram of 80286

- The CPU may be viewed to contain four functional parts:
  1. Address unit (AU)
  2. Bus Unit (CU)
  3. Instruction Unit (IU)
  4. Execution Unit (EU)
- The **Address Unit (AU)** is responsible for calculating the physical address of instructions and data that CPU wants to access.
  - This physical address is computed by the address unit is handed over to the Bus Unit (BU) of the processor.
- The major function of the **Bus Unit (BU)** is to fetch instruction bytes from the memory.
- The **Instruction Unit(IU)** accepts instructions from the prefetch queue and an instruction decoder decodes them one by one.
- The output from the decoding circuit drives a control circuit in the **Execution Unit (EU)** is responsible for instructions received from the decoded instruction queue, which sends the data part of the instruction over the data bus.

Address Unit

- Calculate the physical addresses of the instruction and data that the processor wants to access.
- Address lines derived by this unit may be used to address different peripherals.
- Physical address computed by the address unit is handed over to the Bus Unit (BU).
- It consists of segment registers, offset address and a physical address adder

Bus Unit

- Transmit the physical address over address bus A0 – A23.
- Instruction pipelining.
  - **Prefetcher module** in the bus unit performs this task of prefetching.
- **Bus controller** controls the prefetcher module.
- Fetched instructions are arranged in a **6-byte prefetch queue**.
- **Processor Extension Interface Module** take care of the communication between CPU and a coprocessor.

Instruction Unit

- Receive arranged instructions from 6 byte prefetch queue.
- **Instruction decoder** decodes the instruction one by one and are latched onto a **decoded instruction queue**.
- Output of the decoding circuit drives a control circuit in the Execution Unit.

Execution Unit

- Control unit is responsible for executing the instructions received from the decoded instruction queue.
- Contains Register Bank.
- ALU is the heart of the execution unit.
- After execution ALU sends the result either over data bus or back to the register bank.

Register Organization of 80286

- The 80286 contains almost the same set of registers, as in 8086.
  1. Eight 16-bit general purpose registers (AX, BX, CX, DX, SP, BP, SI, DI).
  2. Four 16-bit segment registers (CS, SS, DS, ES).
  3. Status and control register (Flag Register) .
  4. Instruction pointer (IP)

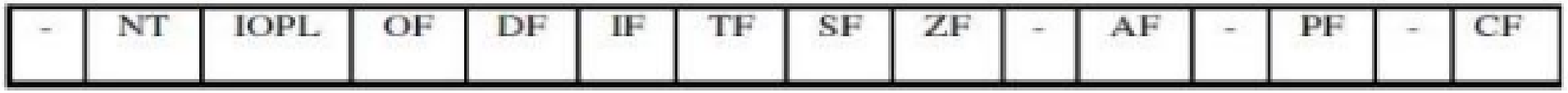
Operating Modes of 80286

1. **Real Address Mode :**
   - 80286 is just a fast 8086 --- up to 6 times faster
   - All memory management and protection mechanisms are disabled
   - And program for 8086 can be executed without modification in 80286.
   - It offers memory addressability of 1 MB of physical memory.
   - Lines A20-A23 are not used
2. **Protected Virtual Address Mode**
   - 80286 works with all of its memory management and protection capabilities with the advanced instruction set.
   - 80286 supports multitasking.
   - Able to protect memory space for another program using virtual memory.
   - This mode of 80286 offers memory addressability of 16 MB of physical memory along with 1 GB of virtual memory.
   - Swapping and unswapping

Operating Modes of 80286

- 80286 can treat external storage as it were physical memory.
- Execute programs that are too large to be contained in physical memory.
- Program can be upto 2^30 bytes.

Flag Register



- D2, D4, D6, D7 and D11 are called as status flag bits.
- The bits D8 (TF) and D9 (IF) are used for controlling machine operation and thus they are called control flags.
- The additional fields available in 80286 flag registers are:
  1. IOPL - I/O Privilege Field (bits D12 and D13)
  2. NT - Nested Task flag (bit D14)
  3. PE - Protection Enable (bit D16)
  4. MP - Monitor Processor Extension (bit D17)
  5. EM - Processor Extension Emulator (bit D18)
  6. TS – Task Switch (bit D19)

Descriptor

- Large programs are divided into smaller segments which are arranged in appropriate sequence and are swapped in or out of primary memory as per the requirements, for the complete execution of program.
- A data structure called **descriptor** is associated with this segment, which contains the information regarding the segment.
- A set of such descriptors arranged in a proper sequence describes the complete program.

Descriptor

- It carry all relevant information regarding a segment and its access rights.
- The descriptor contains information of a segment, like
  - Segment base address
  - Segment limit
  - Segment type
  - Privilege level
  - Segment availability in physical memory
  - Descriptor type
  - Segment use by another task
- The set of descriptors is called as **descriptor table**.

Segment Descriptor Cache Register

- The concept of caching was introduced in 80286.
- Caching is a method to minimize the time required for fetching the frequently required descriptor information from the memory.
- Caching is the process of maintaining the most frequently required data for execution in a high speed memory called cache memory.
- 6-byte segment descriptor cache register is assigned to each of the four segments.
- A segment descriptor is automatically loaded in a segment descriptor cache register, whenever the associated segment register is loaded.
- Once a cache register is loaded, all the information regarding the segmet is obtained from the cache register instead of referring to the main memory fro the descriptor again and again,