MICROPROCESSORS AND MICROCONTROLLERS

Dr. P. Lachi Reddy

Professor in ECE

LBRCE, Mylavaram

UNIT - I

8086 MICROPROCESSOR

TOPICS

- Architecture
- Pin Diagram
- Register Organization
- Minimum Mode and Maximum Mode
- Timing Diagrams
- Addressing Modes
- Instruction Set
- Interrupt Vector Table
- Assembly Language Programming
  - Data Transfer
  - Arithmetic
  - Logical and Decision Making Operations

8086 Microprocessor Features

- It operates 4.77-10 MHz frequency range
- It is a 16-bit Microprocessor
- 20-bit Address bus can access 1MB of memory
- It supports 64K I/O ports (Devices)
- It has 14 Registers of 16-bit each
- It has multiplexed address and data bus
- It has 6 byte queue
- It requires +5V power supply
- It is a 40 pin dual in line package IC
- It operates in two modes: Minimum and Maximum modes

8086 Block Diagram

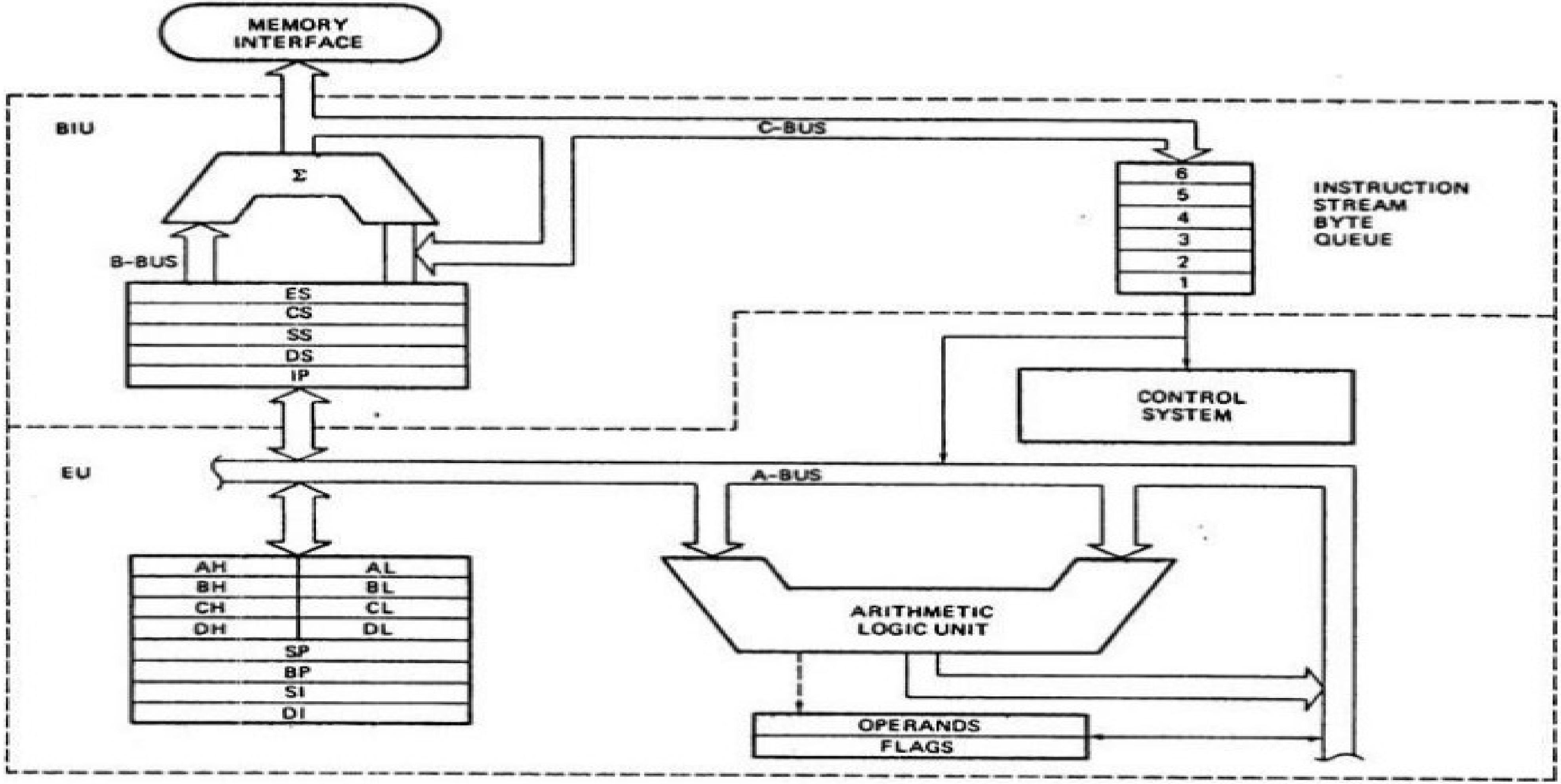
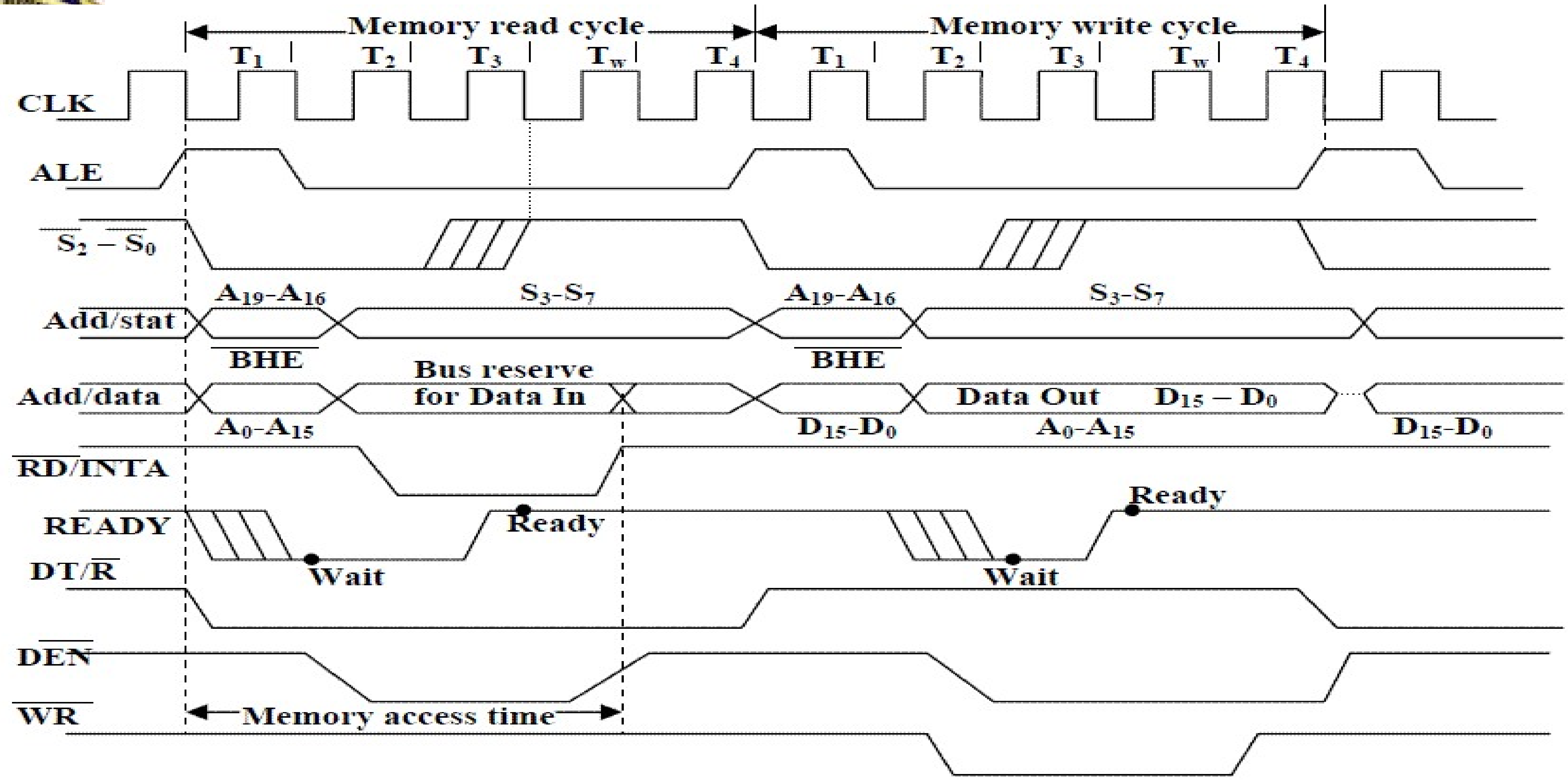
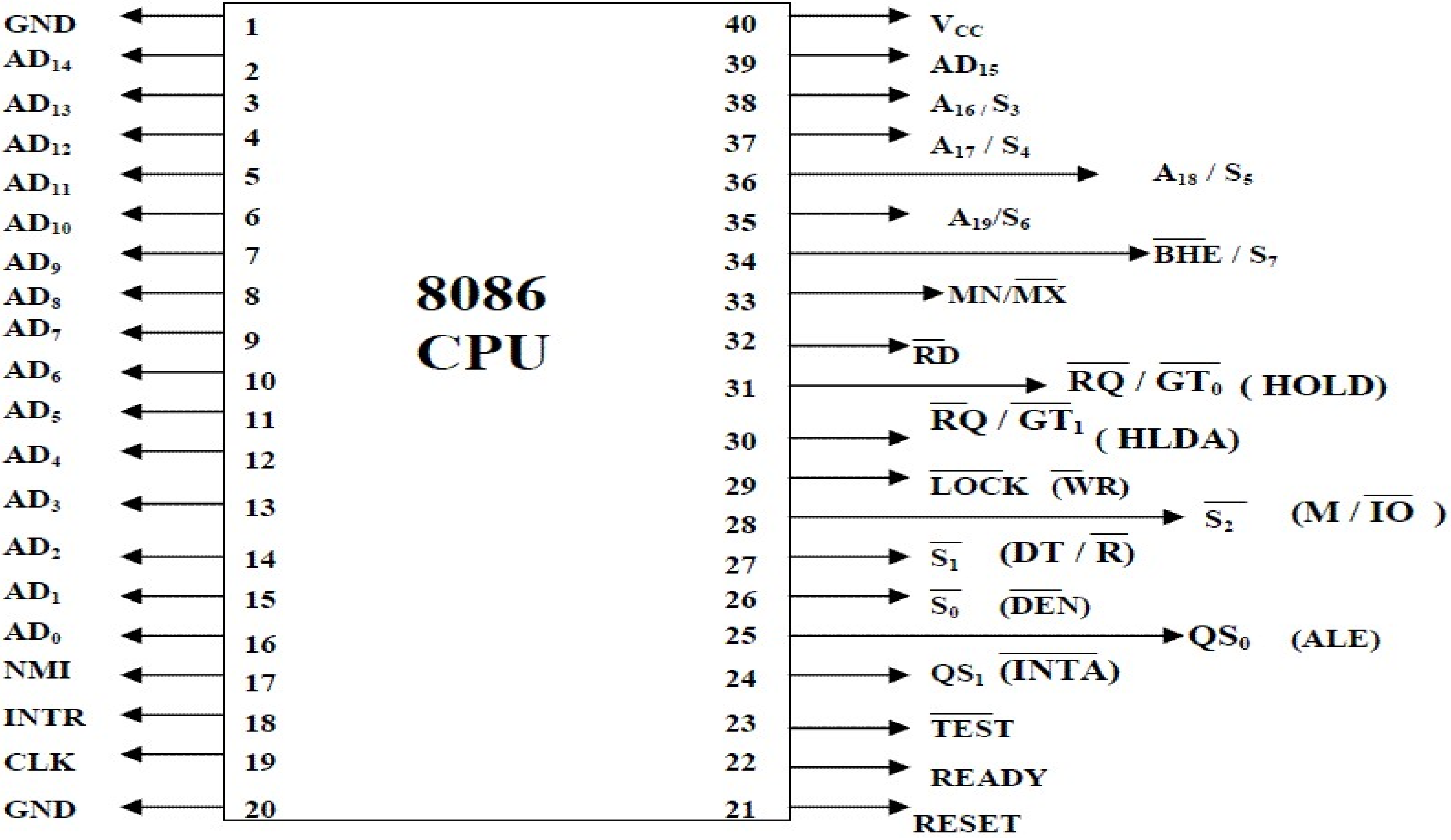


Fig 1.17 Architecture of 8086

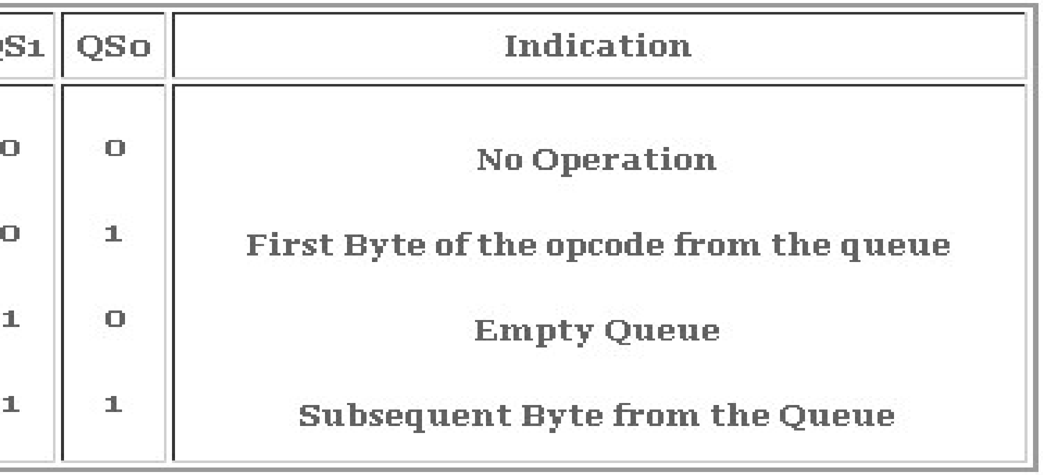
Bus Operation



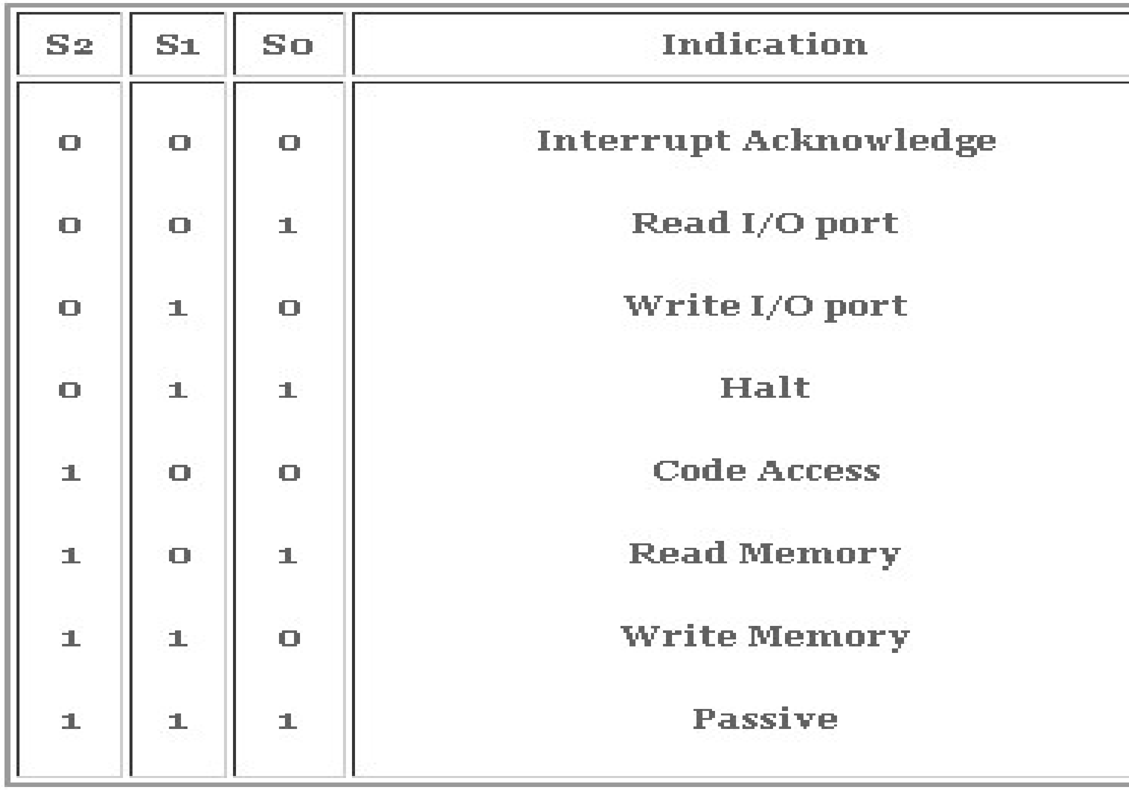
Pin Diagram of 8086



Status Signals

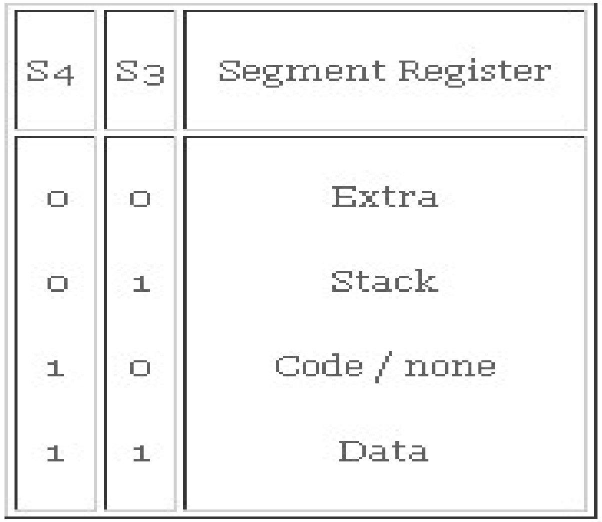


QS1	QS0	Indication
0	0	No Operation
0	1	First Byte of the opcode from the queue
1	0	Empty Queue
1	1	Subsequent Byte from the Queue



S2	S1	S0	Indication
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

Status Signals



S4	S3	Segment Register
0	0	Extra
0	1	Stack
1	0	Code / none
1	1	Data

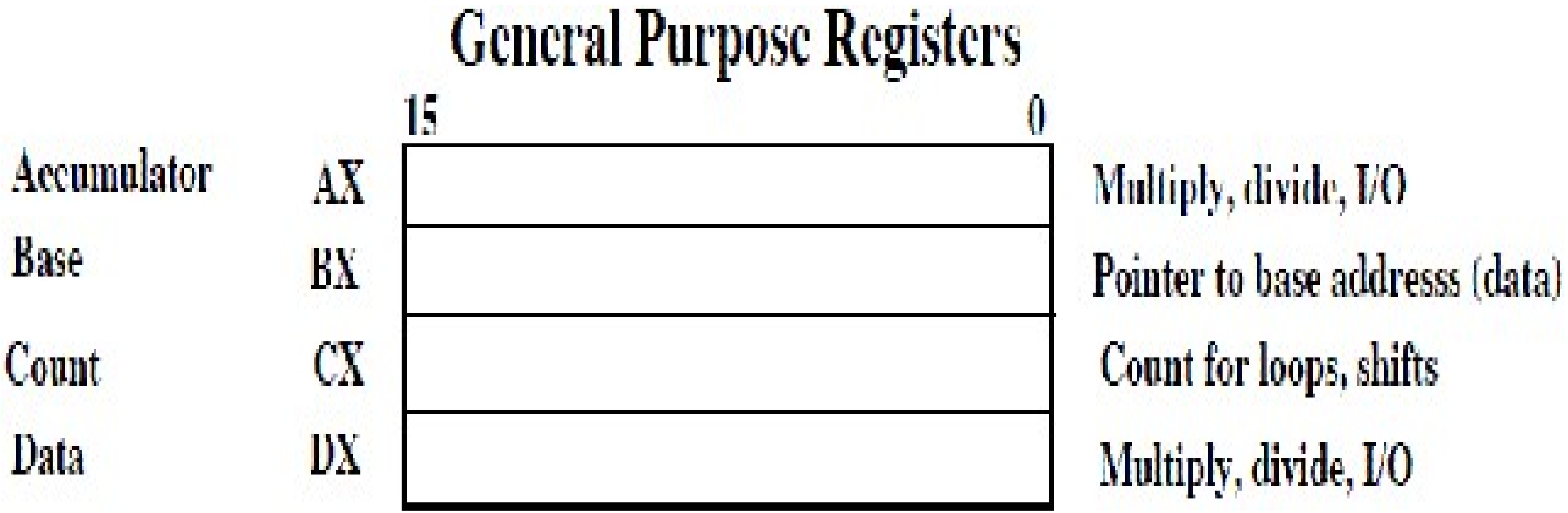
Memory segment status codes

BHE	A_0	Indication
0	0	Whole word
0	1	Upper byte to or from Odd address
1	0	Lower byte to or from Even address
1	1	None

Register Organization of 8086

- Four General purpose registers
- Four Index/Pointer registers
- Four Segment registers
- Two Other registers

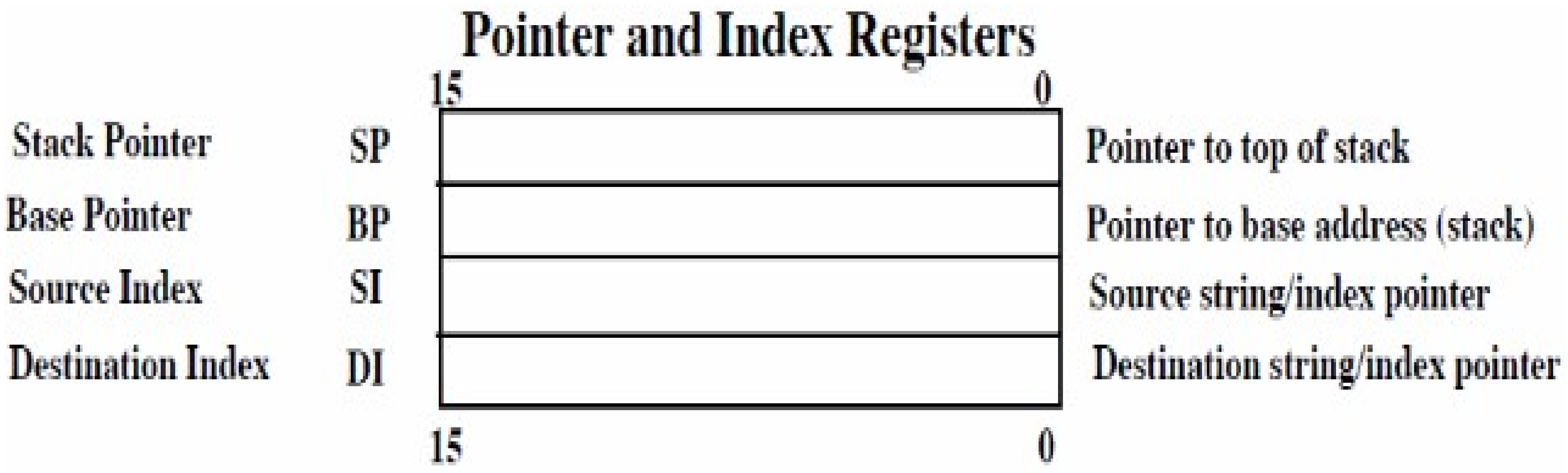
General purpose registers



General Purpose Registers

		15 0	
Accumulator	AX		Multiply, divide, I/O
Base	BX		Pointer to base addresss (data)
Count	CX		Count for loops, shifts
Data	DX		Multiply, divide, I/O

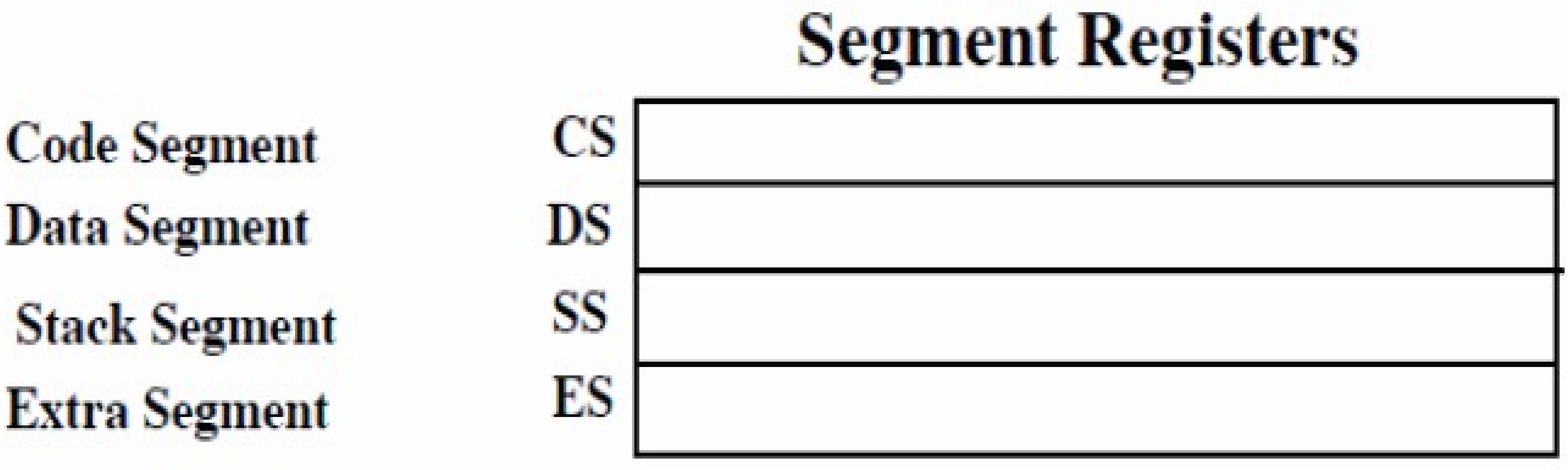
Index/Pointer registers



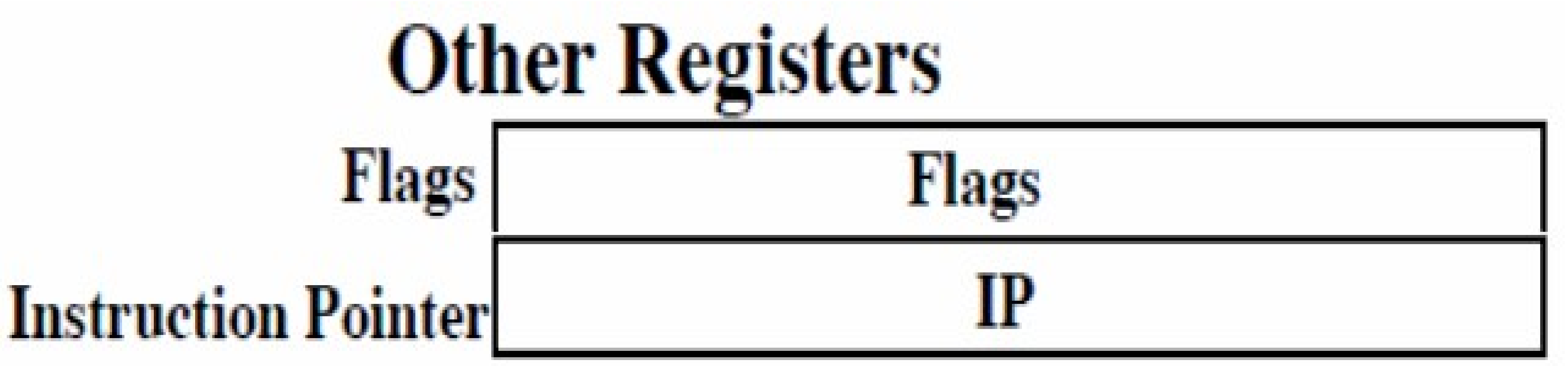
Pointer and Index Registers

		15 — 0	
Stack Pointer	SP		Pointer to top of stack
Base Pointer	BP		Pointer to base address (stack)
Source Index	SI		Source string/index pointer
Destination Index	DI		Destination string/index pointer
		15 — 0	

Segment registers



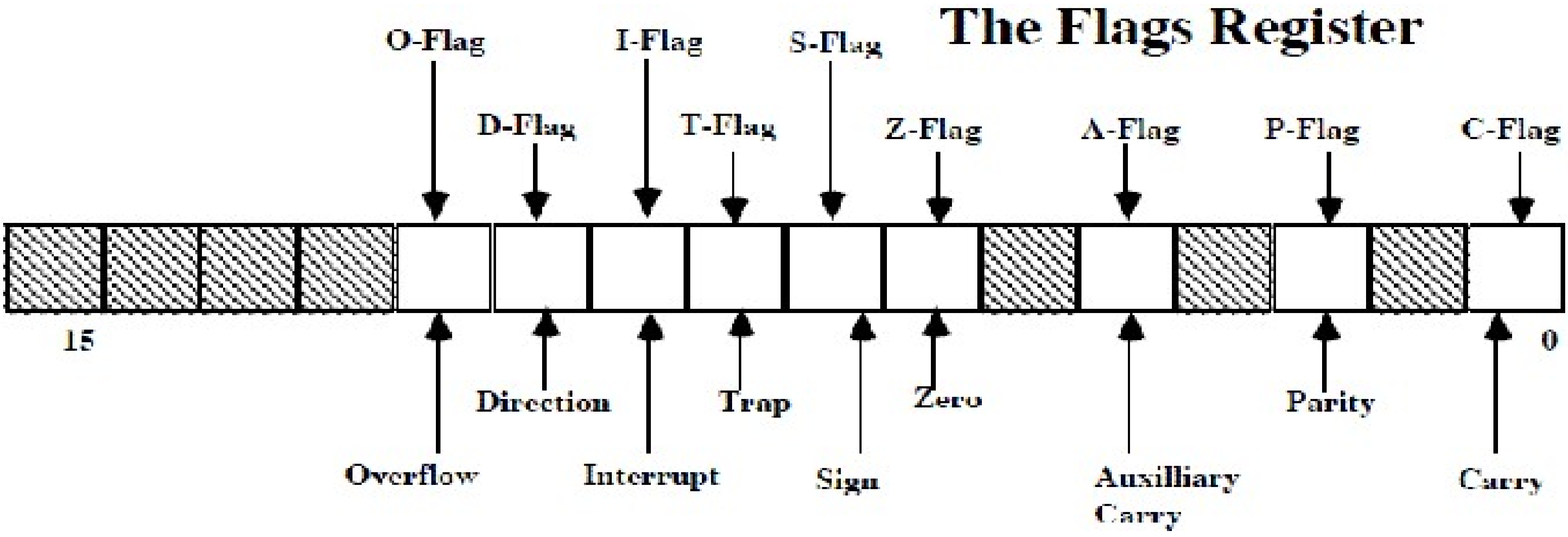
Other registers



Other Registers

Flags	Flags
Instruction Pointer	IP

Flags Register



Memory Segmentation

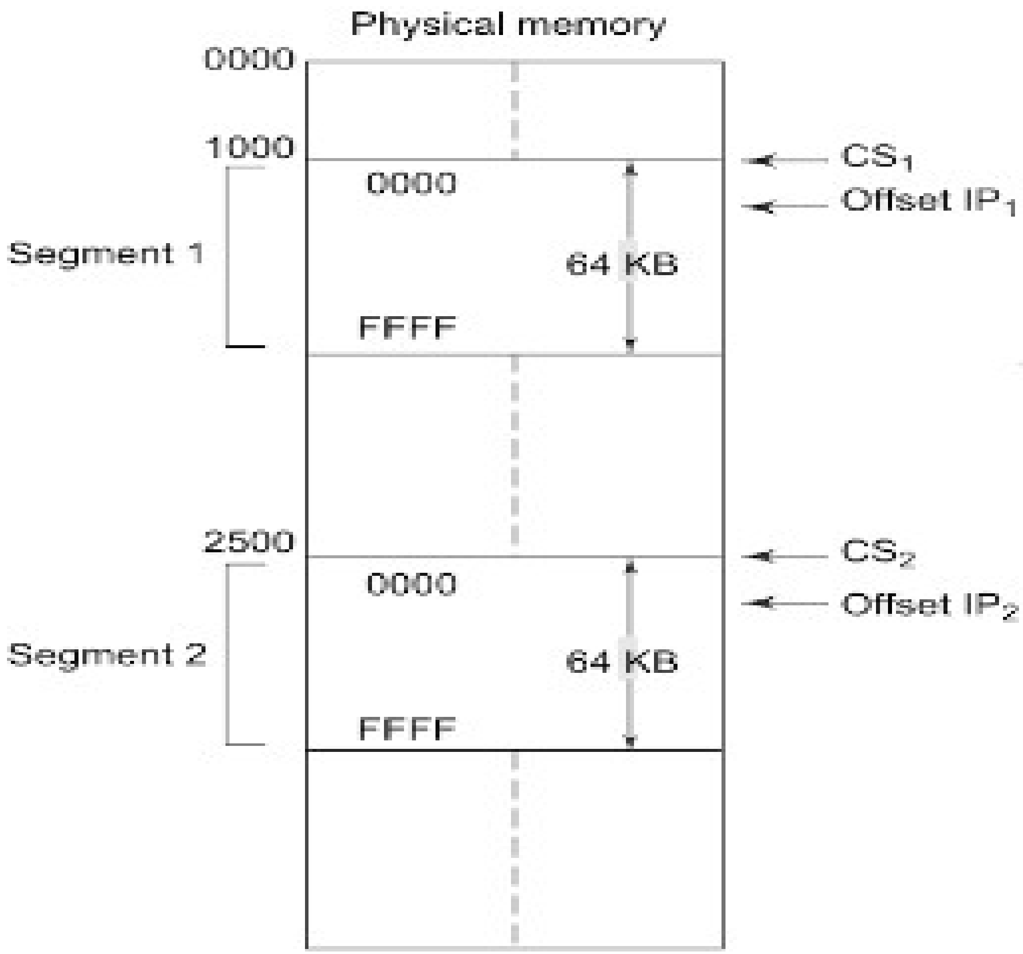


Fig. 1.3(a) *Non-overlapping Segments*

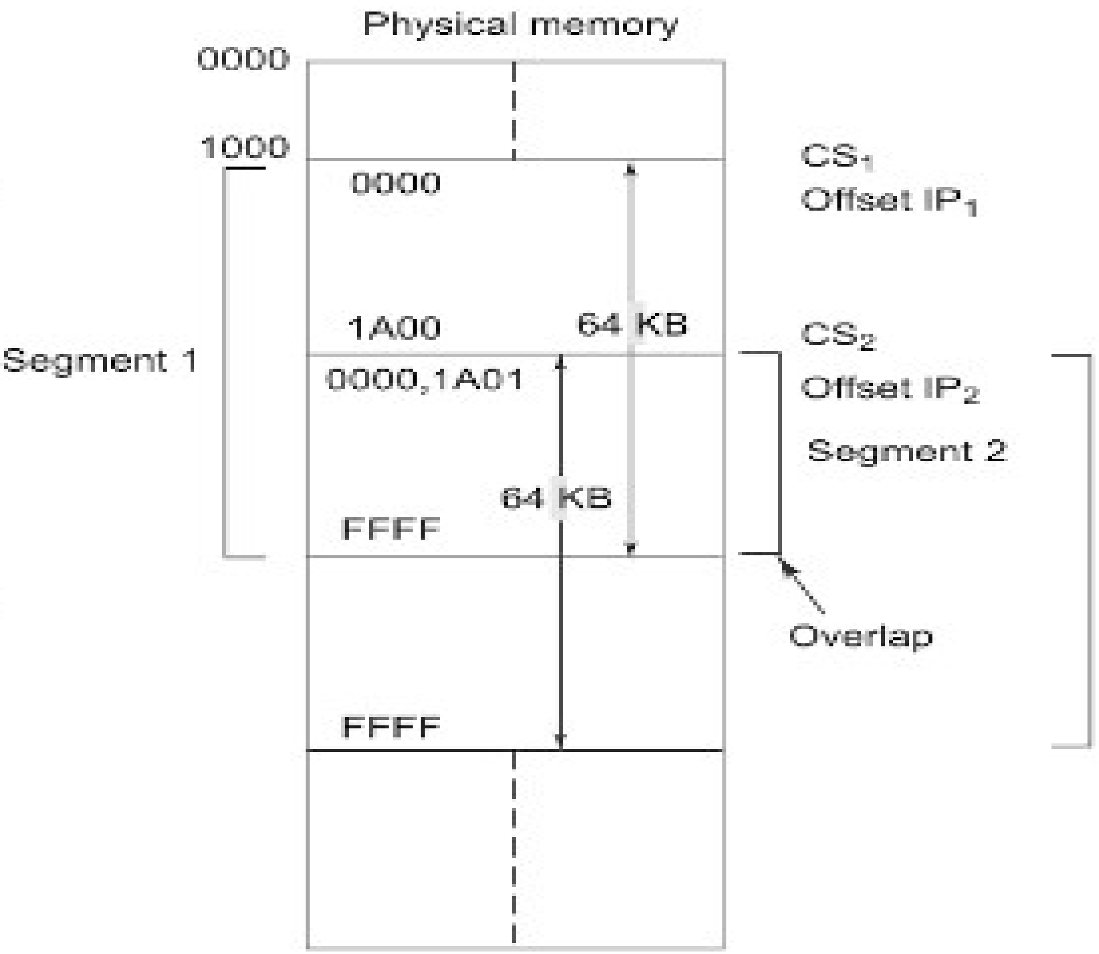
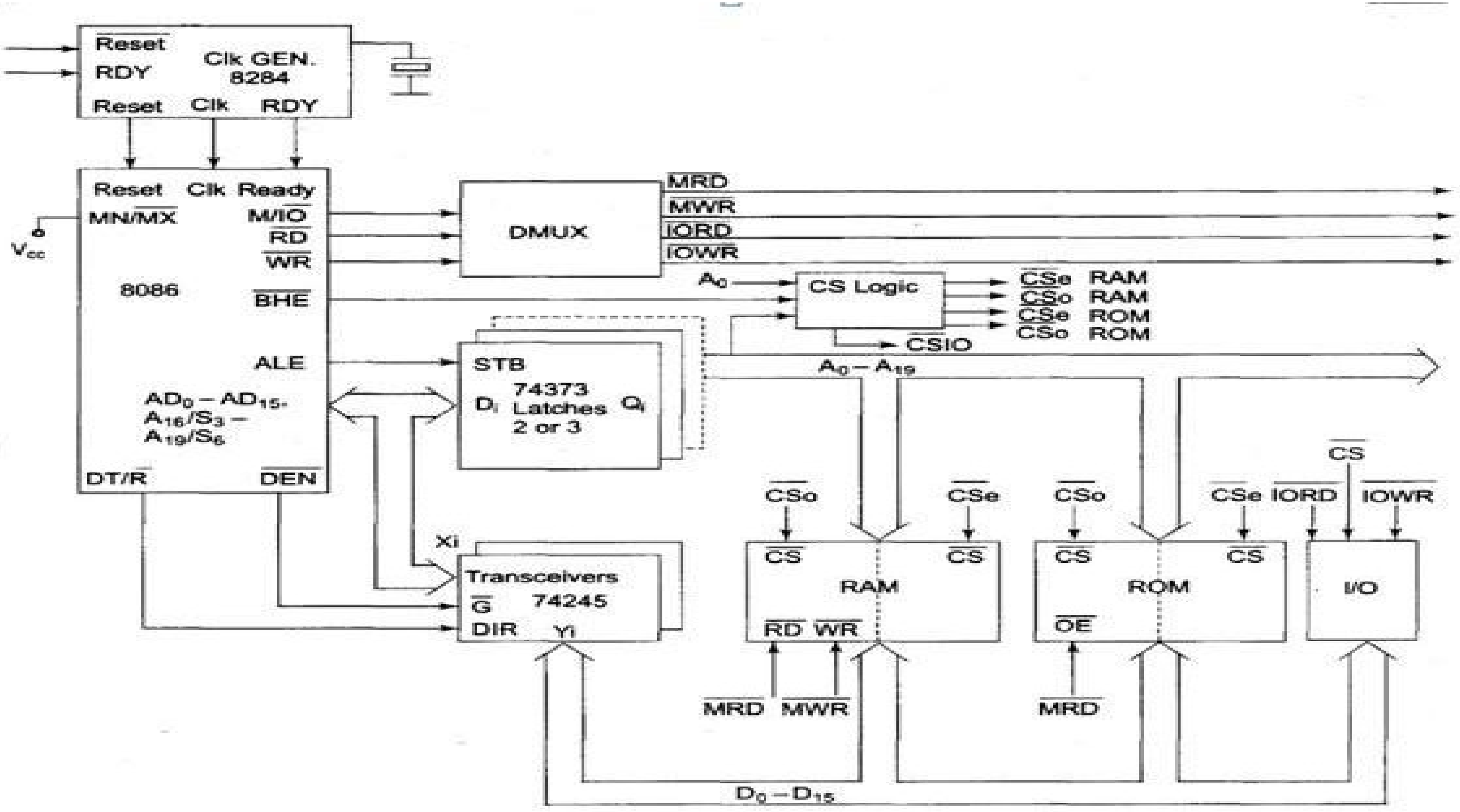
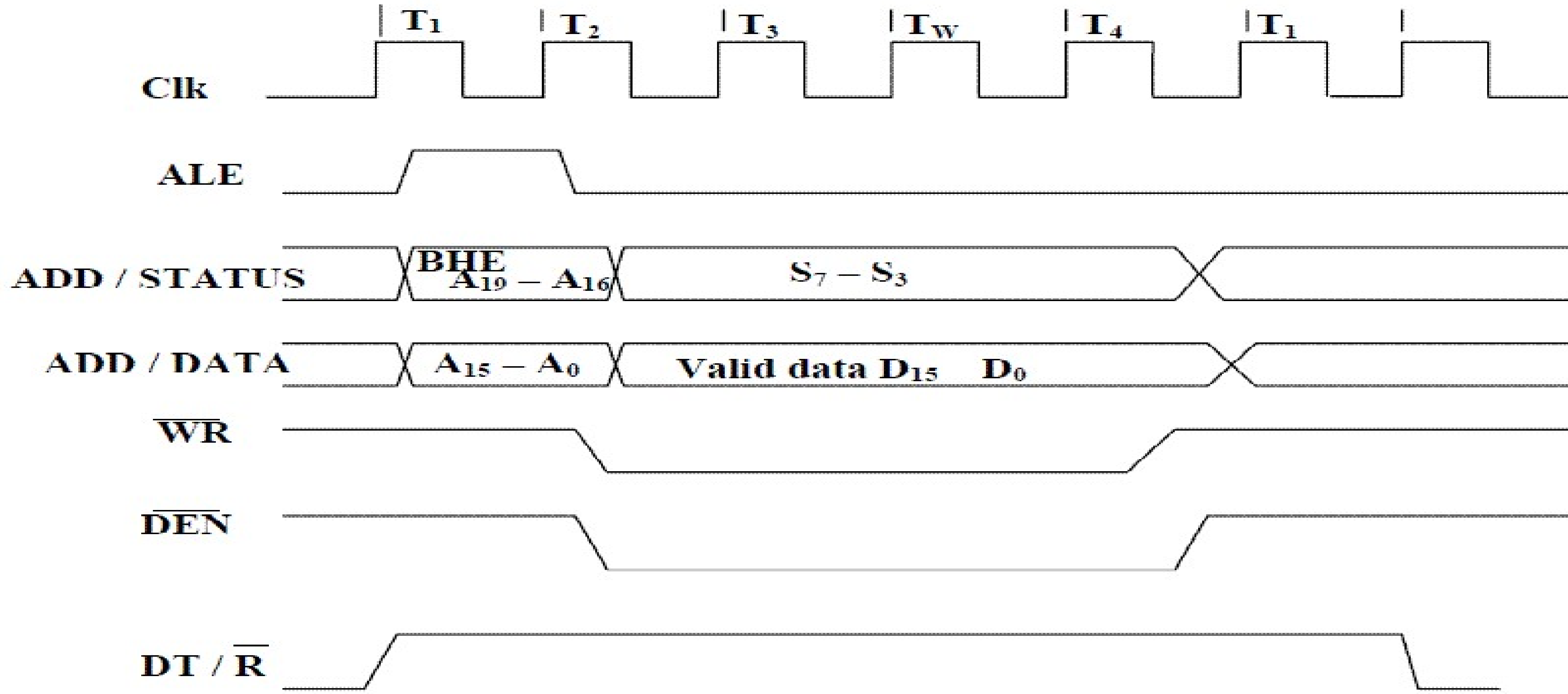


Fig. 1.3(b) *Overlapping Segments*

Minimum Mode 8086 System

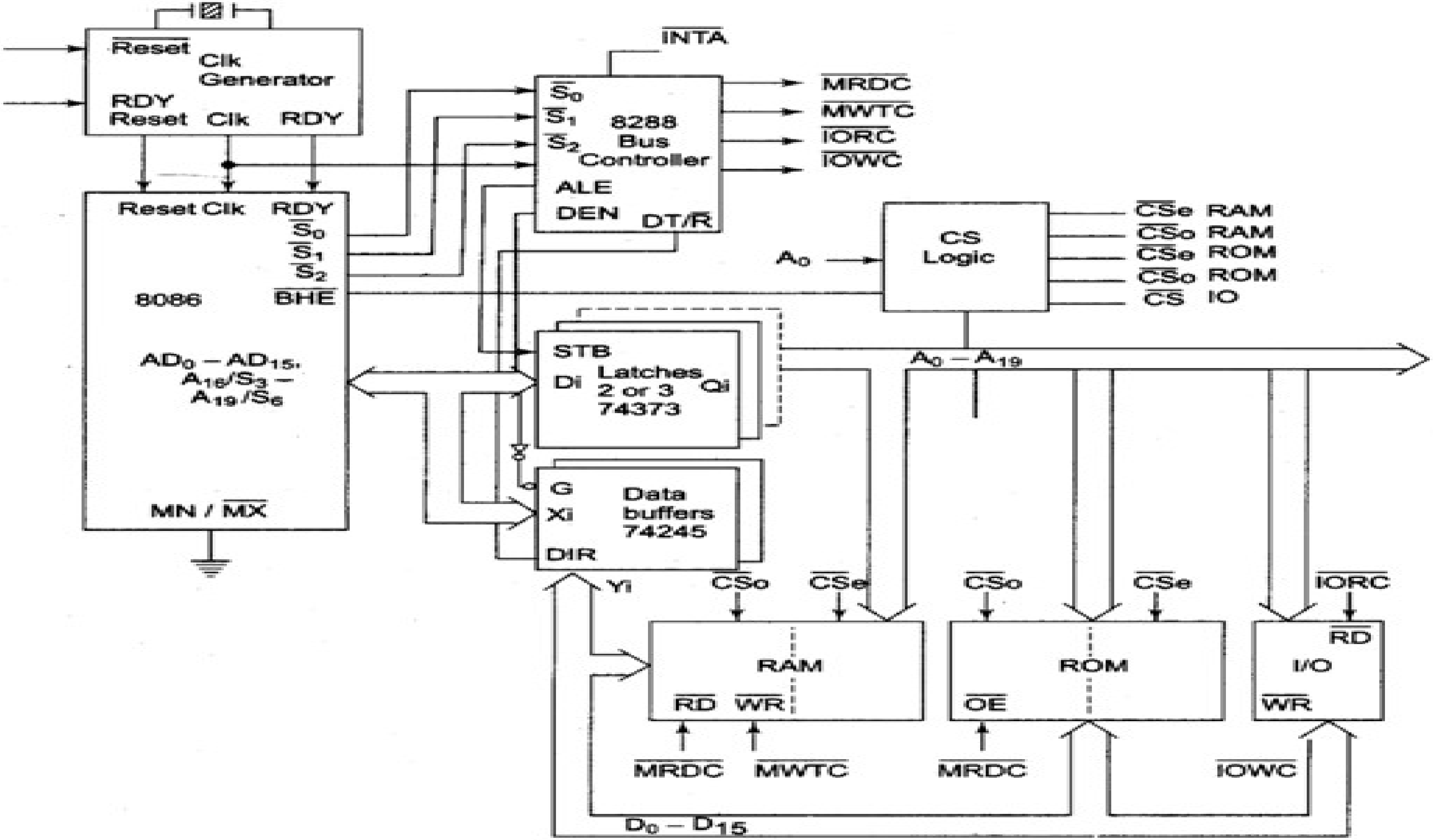


Minimum Mode Timing Diagram

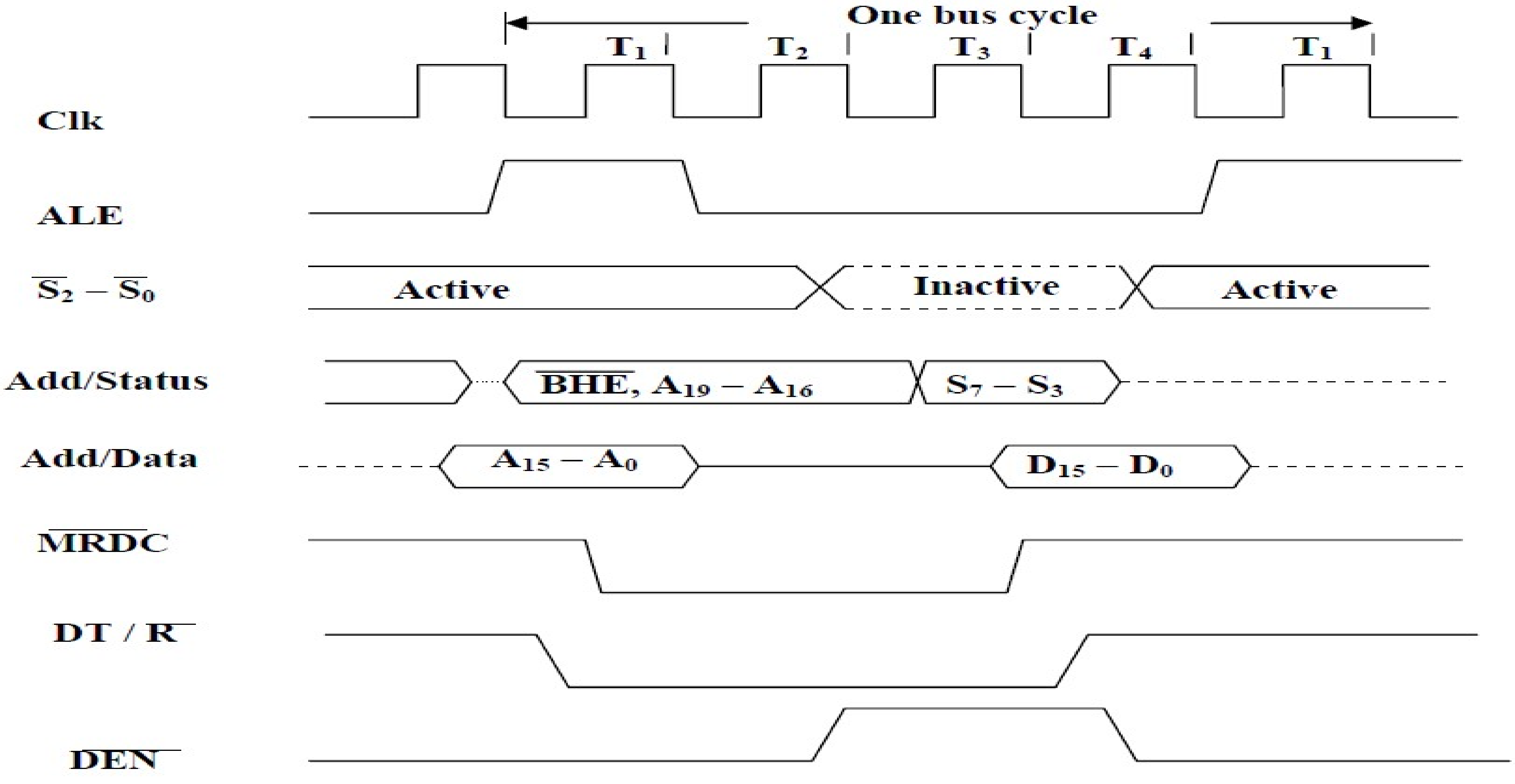


Write Cycle Timing Diagram for Minimum Mode

Maximum Mode 8086 System



Maximum Mode Timing Diagram



Memory Read Timing in Maximum Mode

Addressing Modes of 8086

- Immediate Addressing Mode
- **Example:** MOV AX, 0005H
- Direct Addressing Mode
- **Example:** MOV AX, [5000H]
- Register Addressing Mode
- **Example:** MOV BX, AX
- Register Indirect Addressing Mode
- **Example:** MOV AX, [BX]

Addressing Modes of 8086

- Indexed Addressing Mode
- **Example:** MOV AX, [SI]
- Register Relative Addressing Mode
- **Example:** MOV AX, 50H [BX]
- Base Indexed Addressing Mode
- **Example:** MOV AX, [BX][SI]
- Relative Base Indexed Addressing Mode
- **Example:** MOV AX, 50H [BX] [SI]

Addressing Modes for control transfer instructio

- Intersegment direct
- **Example:** JMP 5000H:2000H
- Intersegment indirect
- **Example:** JMP [2000H]
- Intrasegment direct
- **Example:** JMP SHORT LABEL
- Intrasegment indirect
- **Example:** JMP [BX]; Jump to effective address stored in BX

Instruction Set of 8086

- Data transfer Instructions
- Arithmetic & Logical Instructions
- Program control transfer Instructions
- Machine Control Instructions
- Shift / Rotate Instructions
- Flag Manipulation Instructions
- String Instructions

Data transfer Instructions

- MOV BX, 00F2H
- PUSH CX
- POP CX
- XCHG BX, CX
- IN AL, 0F8H
- OUT DX, AL
- XLAT
- LEA
- LDS/LES
- LAHF
- SAHF
- PUSHF
- POPF

Arithmetic & Logical Instructions

- ADD AX, BX
- ADC BX, CX
- INC DX
- SUB BX, AX
- SBB AX, CX
- DEC CX
- CMP BX, CX

Arithmetic& Logical Instructions

- NEG AX
- MUL BX
- IMUL BL -Signed Multiplication
- CBW
- CWD
- DIV BX
- IDIV CX
- AND AX, BX
- OR BX, DX
- XOR BX, AX
- NOT AX
- TEST AX, BX

Program control transfer Instructions

- JZ / JE LABEL/ADDRESS
- JNZ / JNE
- JS
- JNS
- JO
- JNO
- JP / JPE
- JNP
- JB / JNAE / JC
- JNB / JAE / JNC

Program control transfer Instructions

- JBE / JNA
- JNBE / JA
- JL / JNGE
- JNL / JGE
- JLE
- JNLE / JG
- CALL ADDRESS
- RET
- IRET
- INTO
- INT N - N can be between 0 and 255

Machine Control Instructions

- WAIT
- HLT
- NOP
- ESC
- LOCK

Shift / Rotate Instructions

- SHL/SAL AX
- SHR BX, CL
- SAR AX, 1
- ROR AX
- ROL AX
- RCR AX
- RCL AX

Flag Manipulation Instructions

- CLC
- CMC
- STC
- CLD
- STD
- CLI
- STI

String Instructions

- REP EX: REP ADD AX, BX
- REPE / REPZ EX: REPE CMPSW
- REPNE / REPNZ
- MOVS / MOVSB / MOVSW
- CMPS / CMPSB / CMPSW
- SCAS / SCASB / SCASW EX: REPNZ SCASW
- LODS / LODSB / LODSW
- STOS / STOSB / STOSW
- LOOP
- LOOPE / LOOPZ
- LOOPNE / LOOPNZ

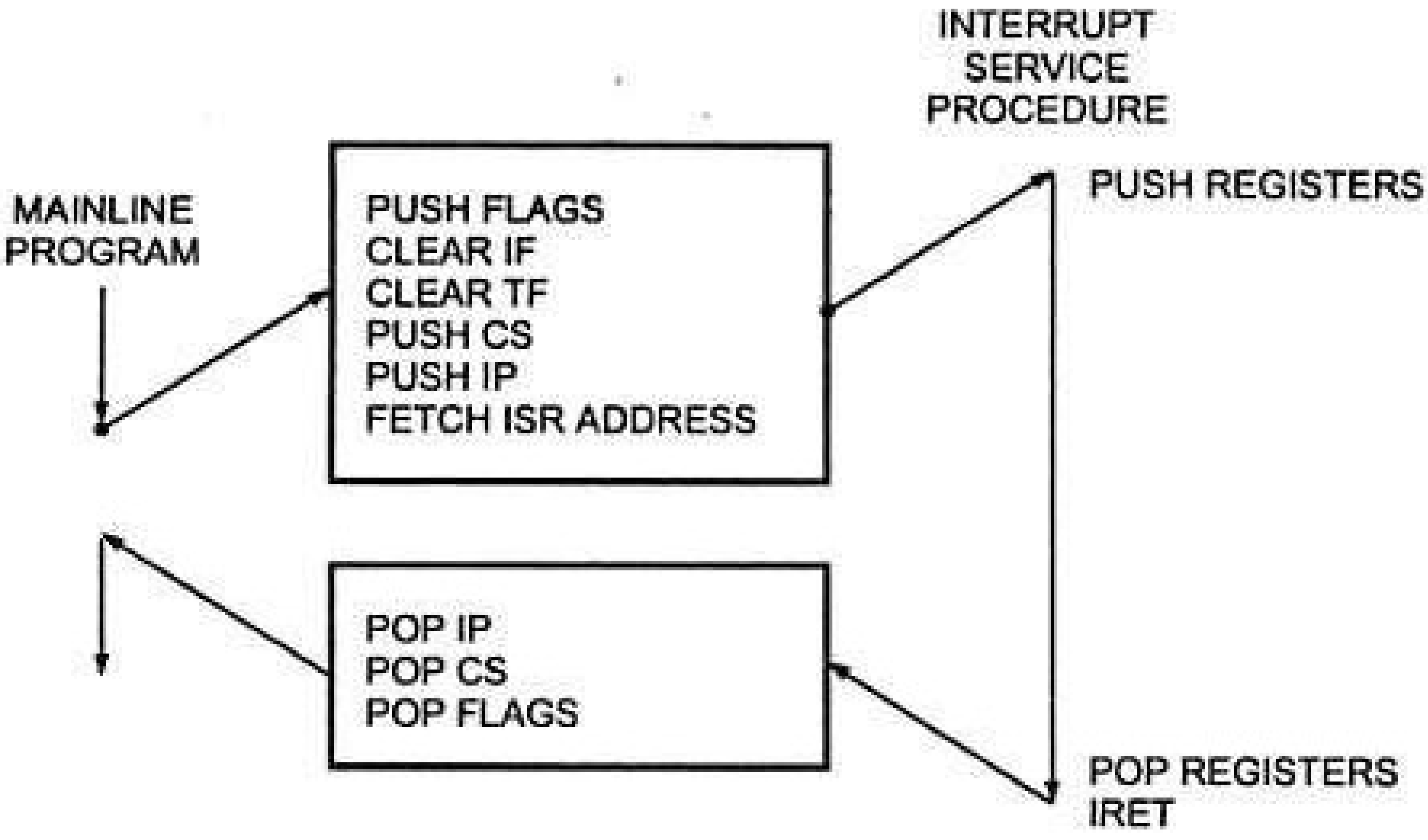
Interrupts

- An interrupt is an event which informs the CPU that its service (action) is needed.
- Sources of interrupts:
- internal fault (e.g.. divide by zero, overflow)
- software
- external hardware :
  - maskable
  - nonmaskable
- reset

Basic Procedure for Processing Interrupts

- When an interrupt is executed, the μp:
  - finishes executing its current instruction (if any).
  - saves (PUSH) the flag register, IP and CS register in the stack.
  - goes to a fixed memory location.
  - reads the address of the associated ISR.
  - Jumps to that address and executes the ISR.
  - gets (PULL) the flag register, CS:IP register from the stack.
  - continues executing the previous job (if any).

Steps involved to service the Interrupts



8086 interrupt response

8088/86 Hardware Interrupts pins

INTR: Interrupt Request.

- Input signal into the CPU
- If it is activated, the CPU will finish the current instruction and respond with the interrupt acknowledge operation
- Can be masked (ignored) thru instructions CLI and STI

- NMI: Non Maskable interrupt.
- Input signal
- Cannot be masked or unmasked thru CLI and STI
- Examples of use: power frailer. Memory error

- INTA: Interrupt Acknowledge.
- Output signal

The Interrupt flag

- IF (Interrupt Enable Flag): used to mask any hardware interrupt that may come in from the INTR pin.
- When IF=0, all hardware interrupt requests through INTR are masked.
- This has no effect on interrupts coming from the NMI pin or “INT nn” instructions.
- CLI sets IF to 0, STI sets IF to 1.

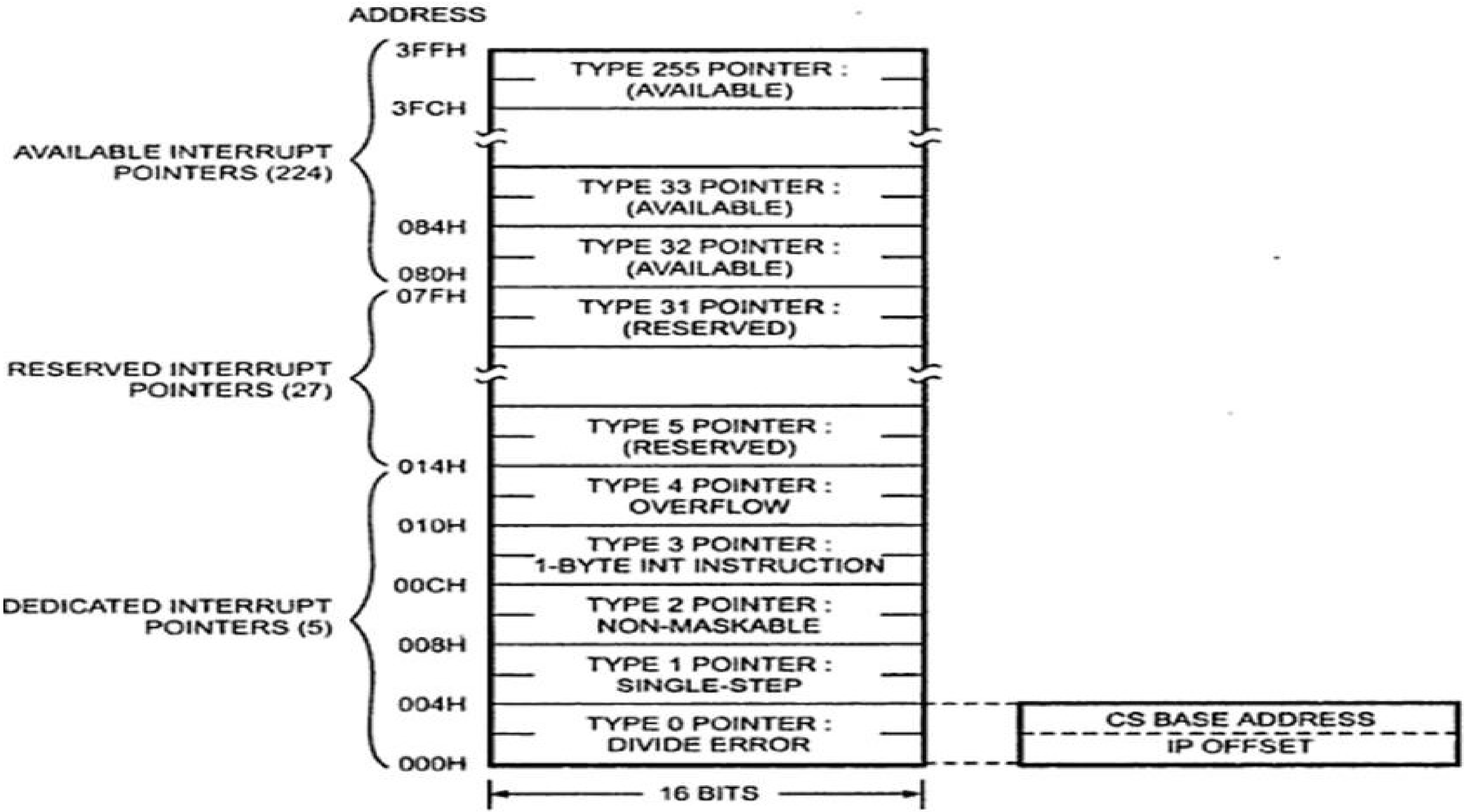
8086 supports 256 Interrupts

- The lowest five types are dedicated to specific interrupts
- The next 27 interrupt types, from 5 to 31 are reserved by Intel for use in future microprocessors.
- The upper 224 interrupt types, from32 to 255, are available to users.

INT *n* and ISR

- *n* is multiplied by 4
- In the address “*4n”* the offset address the ISR is found.
- Example: Intel has set aside INT 2 for the NMI interrupt.
- Whenever the NMI pin is activated, the CPU jumps to physical memory location 00008 to fetch the CS:IP of the interrupt service routine associated with the NMI.

Interrupt Vector Table


ADDRESS
3FFH
3FCH
084H
080H
07FH
014H
010H
00CH
008H
004H
000H
AVAILABLE INTERRUPT POINTERS (224)
RESERVED INTERRUPT POINTERS (27)
DEDICATED INTERRUPT POINTERS (5)
TYPE 255 POINTER : (AVAILABLE)
TYPE 33 POINTER : (AVAILABLE)
TYPE 32 POINTER : (AVAILABLE)
TYPE 31 POINTER : (RESERVED)
TYPE 5 POINTER : (RESERVED)
TYPE 4 POINTER : OVERFLOW
TYPE 3 POINTER : 1-BYTE INT INSTRUCTION
TYPE 2 POINTER : NON-MASKABLE
TYPE 1 POINTER : SINGLE-STEP
TYPE 0 POINTER : DIVIDE ERROR
CS BASE ADDRESS
IP OFFSET
16 BITS

TUTORIAL

- Develop an ALP to find the largest number.
- Develop an ALP to find the factorial of 6.
- Write an ALP to calculate the sum of squares of 5 natural numbers.
- Develop an ALP to find the number of Even and Odd numbers from a given series.
- Develop an ALP to find the number of Positive and Negative numbers from a given series.

End