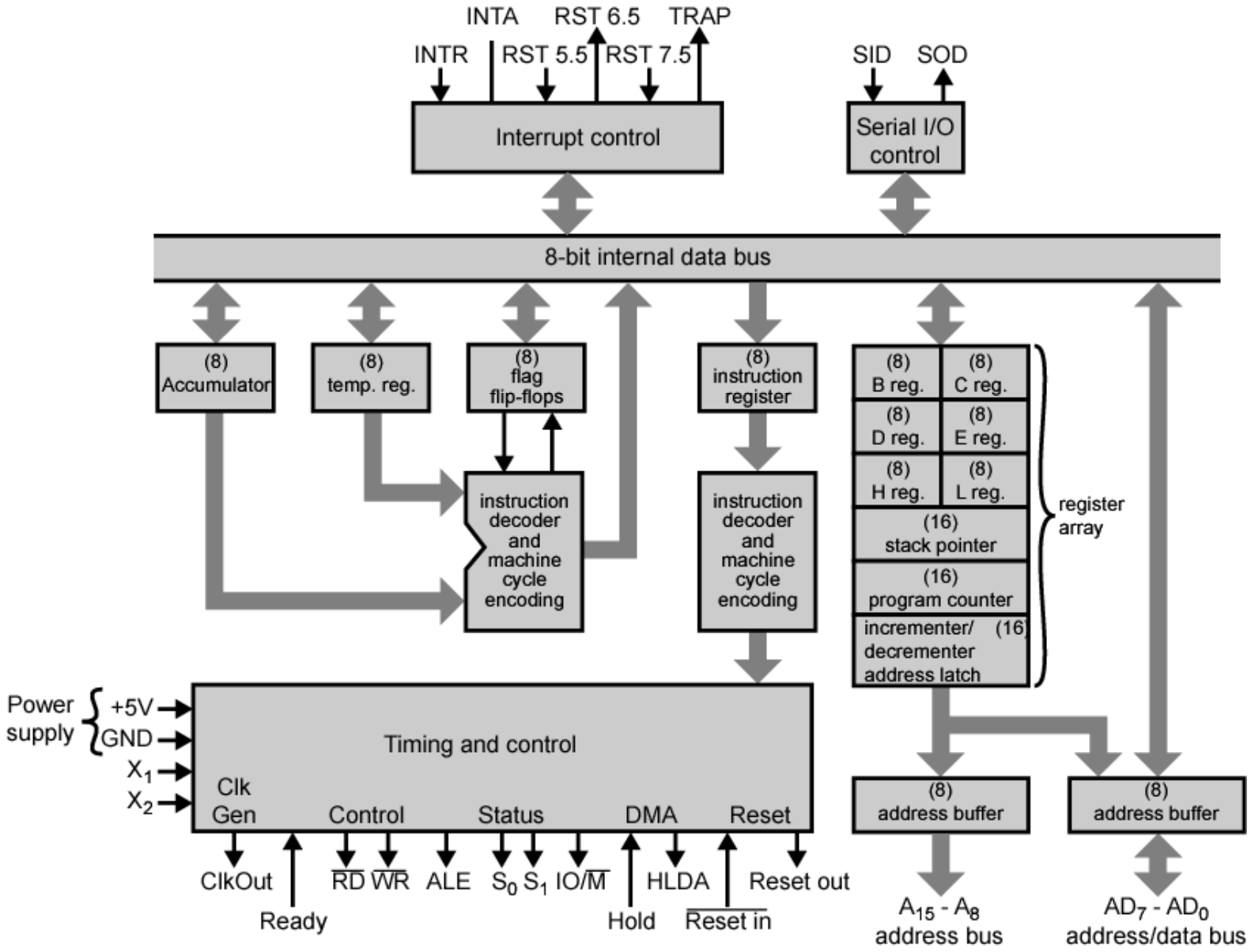
8085 and 8086 Introduction

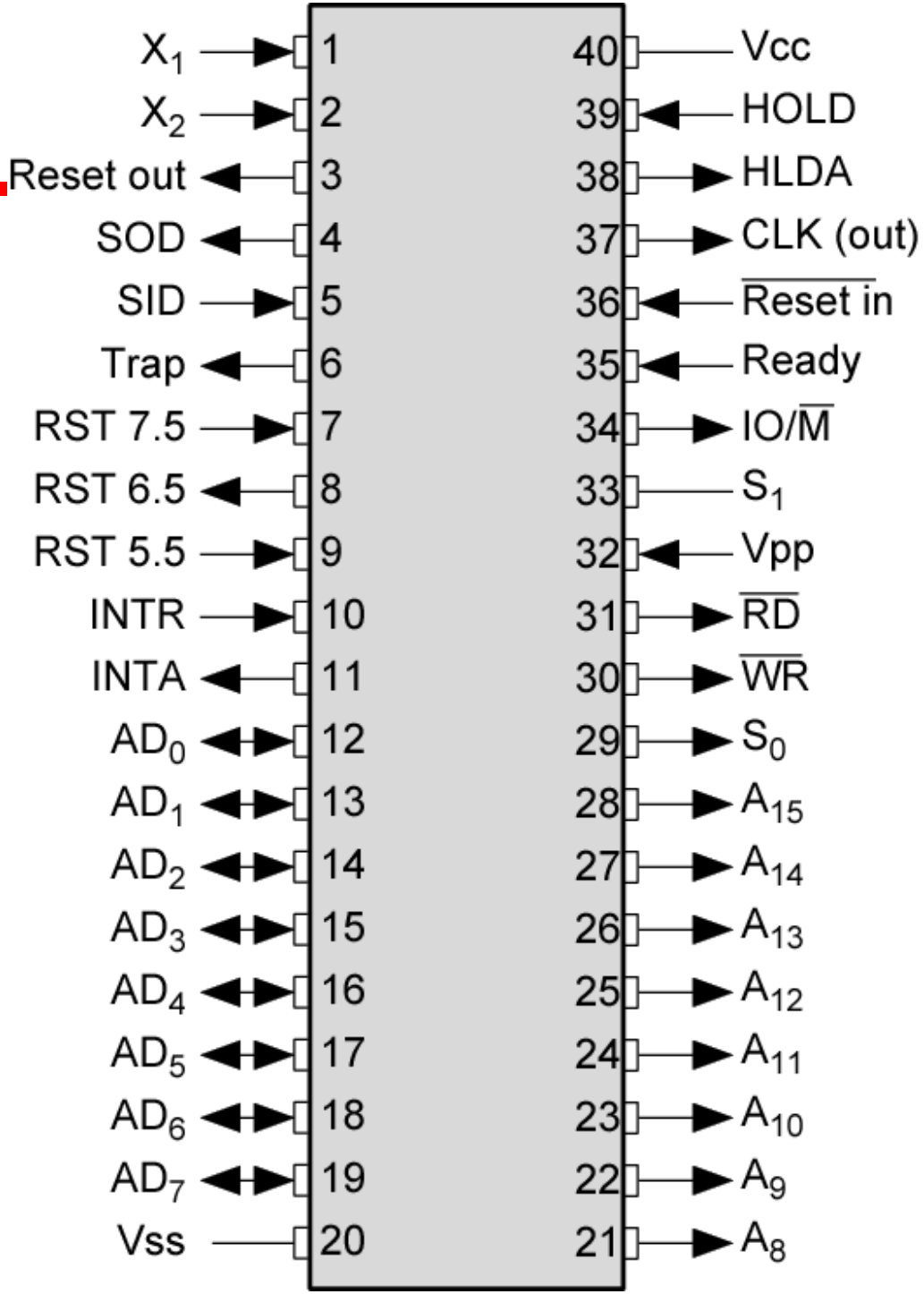
Intel 8085 CPU Block Diagram



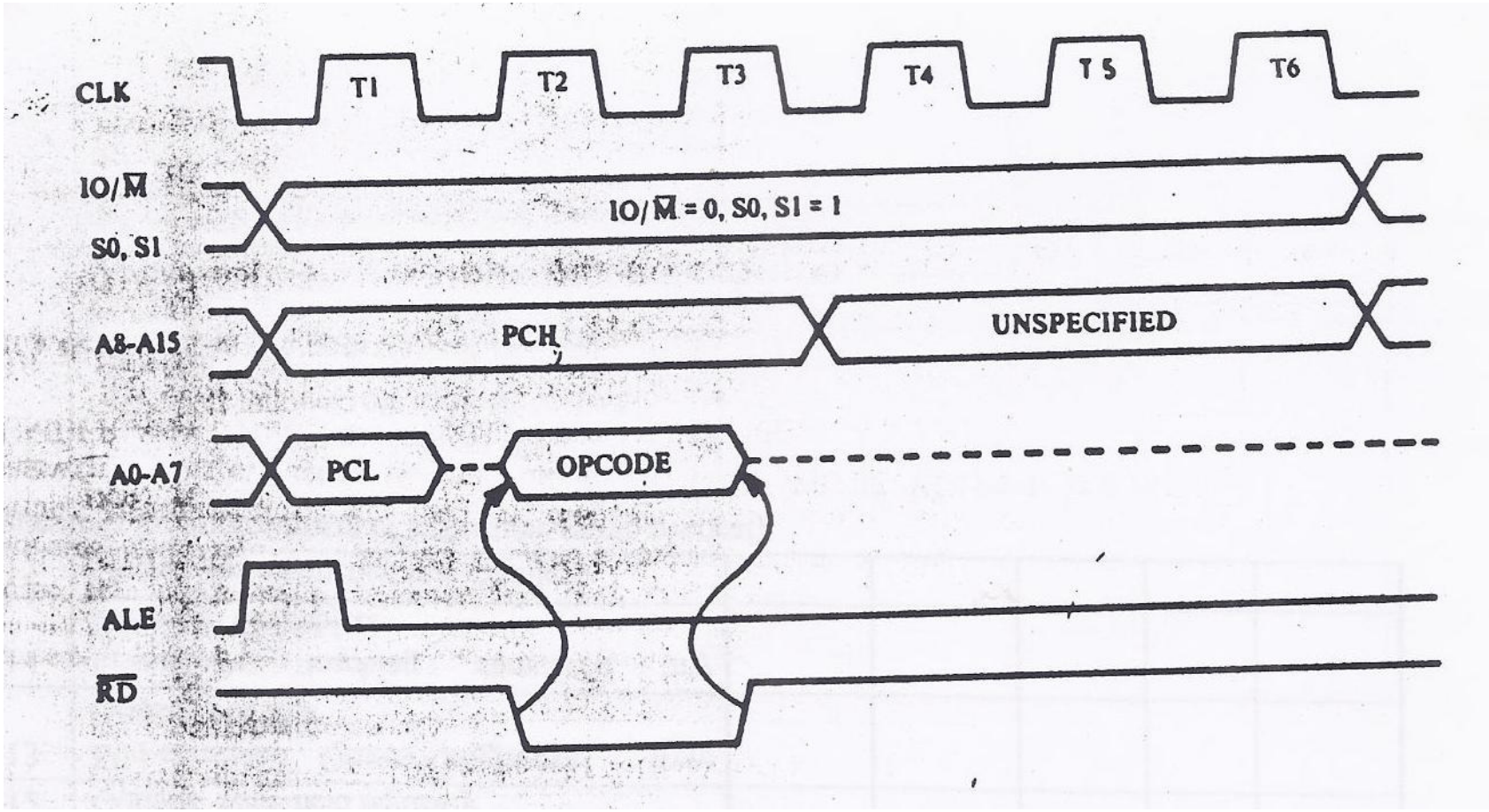
8085 Microprocessor

- Data Bus: 8 bits
  - Size of register: 8 bits
  - can handle 16 bits
- Address Bus: 16 bits
  - memory is byte organized
  - Memory size: 64 KB (2^{16})
- Data bus and address bus are multiplexed

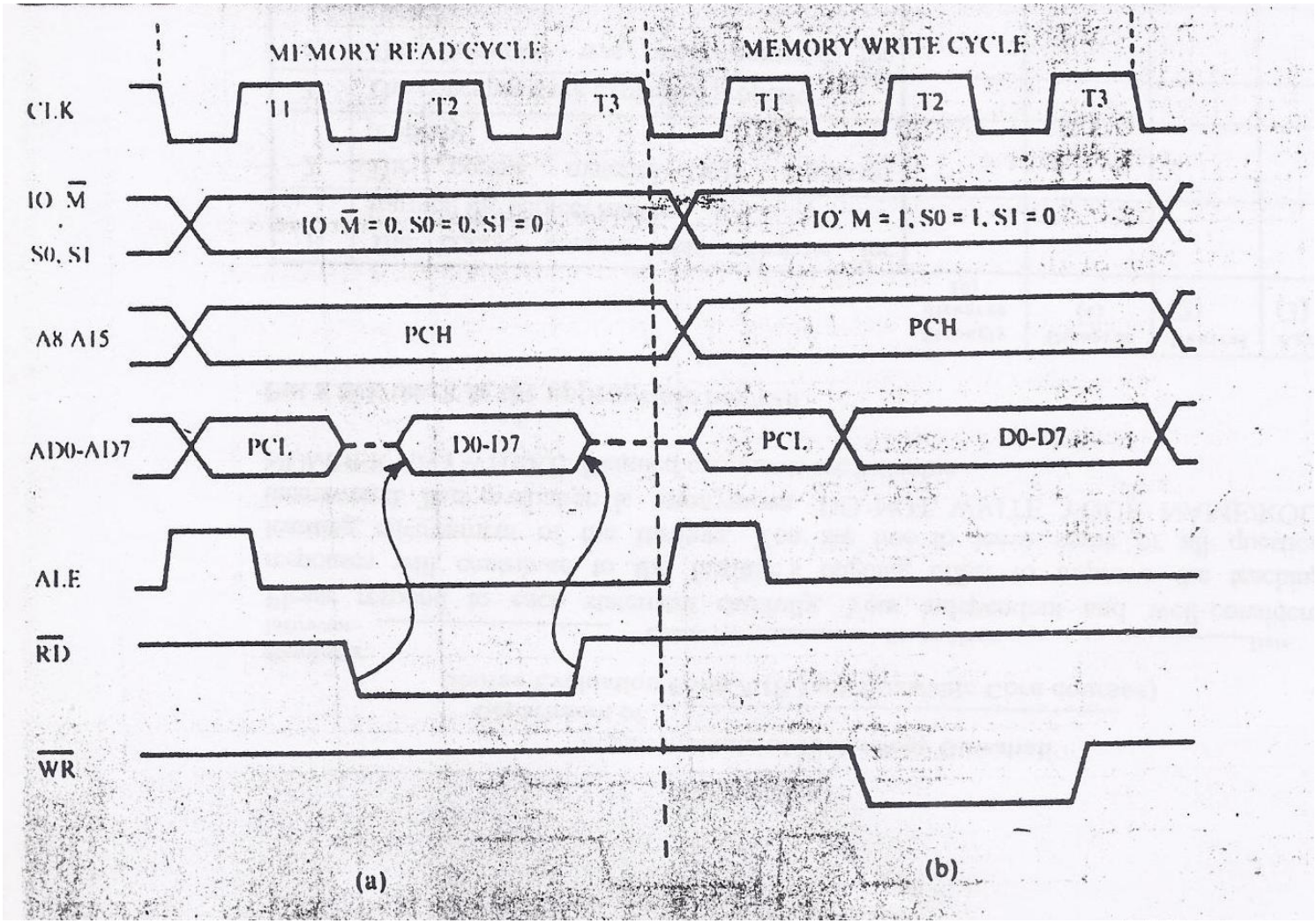
Intel 8085 Pin Configuration



Instruction Fetch Cycle

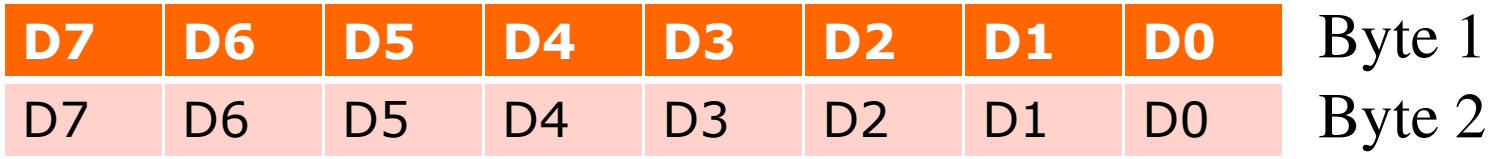


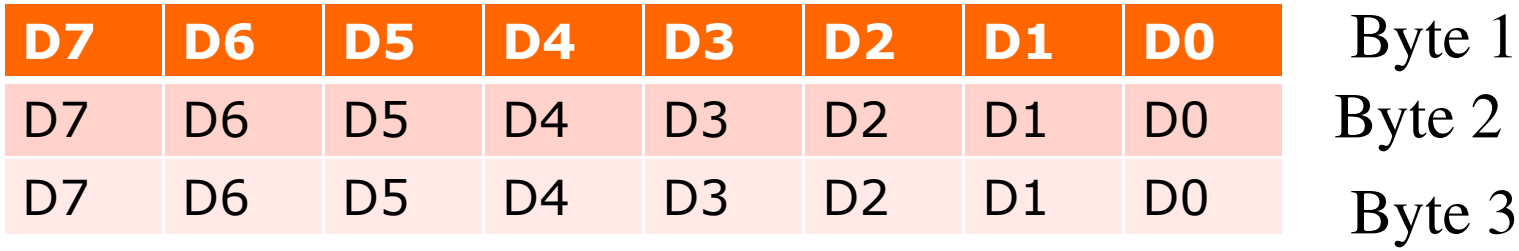
Memory Read and Write Cycle



8085 Instruction Format

- One Byte
  - 

- Two Bytes
  - 

- Three Bytes
  - 

8085 Instruction Set

SYMBOLS	MEANING
accumulator	Register A
addr	16-bit address quantity
data	8-bit quantity
data 16	16-bit data quantity
byte 2	The second byte of the instruction
byte 3	The third byte of the instruction
port	8-bit address of an I/O device
r,r1,r2	One of the registers A,B,C,D,E,H,L
DDD,SSS	The bit pattern designating one of the registers A,B,C,D,E,H,L (DDD = destination, SSS = source):

DDD or SSS	REGISTER NAME
111	A
000	B
001	C
010	D
011	E
100	H
101	L

8085 Instruction Set

rp	One of the register pairs: B represents the B,C pair with B as the high-order register and C as the low-order register; D represents the D,E pair with D as the high-order register and E as the low-order register; H represents the H,L pair with H as the high-order register and L as the low-order register; SP represents the 16-bit stack pointer register.
RP	The bit pattern designating one on the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

rh	The first (high-order) register of a designated register pair.
rl	The second (low-order) register of a designated register pair.
PC	16-bit program counter register (PCH and PCL are used to refer to the high-order and low-order 8 bits respectively).
SP	16-bit stack pointer register (SPH and SPL are used to refer to the high-order and low-order 8 bits respectively).
r_m	Bit m of the register r (bits are number 7 through 0 from left to right).

8085 Instruction Set

1. *Data Transfer Group* — Moves data between registers or between memory locations and registers, includes moves, loads, stores, and exchanges. (See below).

2. *Arithmetic Group* — Adds, subtracts, increments, or decrements data in registers or memory.

3. *Logic Group* — ANDs, ORs, XORs, compares, rotates, or complements data in registers or between memory and a register.

4. *Branch Group* — Initiates conditional or unconditional jumps, calls, returns, and restarts.

5. *Stack, I/O, and Machine Control Group* — Includes instructions for maintaining the stack, reading from input ports, writing to output ports, setting and reading interrupt masks, and setting and clearing flags.

8085 Instruction Set

Data Transfer Group

This group of instructions transfers data to and from registers and memory. *Condition flags are not affected by any instruction in this group.*

MOV r1, r2 (Move Register)

(r1) ← (r2)

The content of register r2 is moved to register r1.

0	1	D	D	D	S	S	S

Cycles: 1
States: 4 (8085), 5 (8080)
Addressing: register
Flags: none

MOV r, M (Move from memory)

(r) ← ((H) (L))

The content of the memory location, whose address is in registers H and L, is moved to register r.

0	1	D	D	D	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

8085 Instruction Set

MOV M, r (Move to memory)
((H) (L)) ← (r)
The content of register r is moved to the memory location whose address is in registers H and L.

0	1	1	1	0	S	S	S

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: none

MVI r, data (Move immediate)
(r) ← (byte 2)
The content of byte 2 of the instruction is moved to register r.

0	0	D	D	D	1	1	0
data							

Cycles: 2
States: 7
Addressing: immediate
Flags: none

MVI M, data (Move to memory immediate)
((H) (L)) ←(byte 2)
The content of byte 2 of the instruction is moved to the memory location whose address is in registers H and L.

0	0	1	1	0	1	1	0
data							

Cycles: 3
States: 10
Addressing: immed./reg. indirect
Flags: none

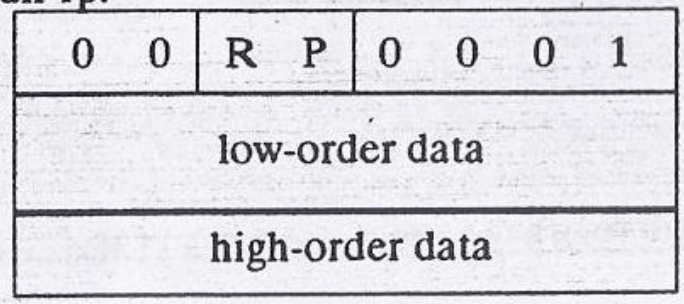
8085 Instruction Set

LXI rp, data 16 (Load register pair immediate)

(rh) ← (byte 3)

(rl) ← (byte 2)

Byte 3 of the instruction is moved into the high-order register (rh) of the register pair rp. Byte 2 of the instruction is moved into the low-order register (rl) of the register pair rp.



Cycles:	3
States:	10
Addressing:	immediate
Flags:	none

The bit pattern designating one on the register pairs B,D,H,SP:

RP	REGISTER PAIR
00	B-C
01	D-E
10	H-L
11	SP

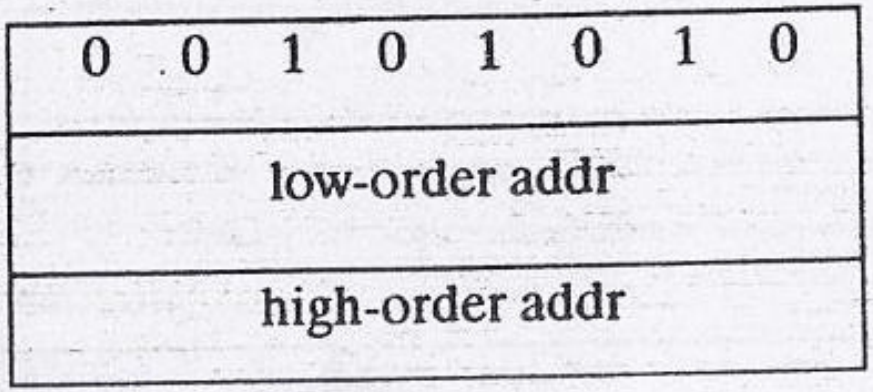
8085 Instruction Set

LHLD addr (Load H and L direct)

(L) ← ((byte 3) (byte 2))

(H) ← ((byte 3) (byte 2) + 1)

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register L. The content of the memory location at the succeeding address is moved to register H.



Cycles:	5
States:	16
Addressing:	direct
Flags:	none

8085 Instruction Set

LDA addr (Load Accumulator direct)

(A) ← ((byte 3) (byte 2))

The content of the memory location, whose address is specified in byte 2 and byte 3 of the instruction, is moved to register A.

0	0	1	1	1	0	1	0
Low-order addr							
high-order addr							

Cycles: 4
States: 13
Addressing: direct
Flags: none

STA addr (Store Accumulator direct)

((Byte 3) (byte 2)) ← (A)

The content of the accumulator is moved to the memory location whose address is specified in byte 2 and byte 3 of the instruction.

0	0	1	1	0	0	1	0
low-order addr							
high-order addr							

Cycles: 4
States: 13
Addressing: direct
Flags: none

8085 Instruction Set

ADD r (Add Register)

(A) ←(A) + (r)

The content of register r is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	S	S	S

Cycles: 1
States: 4
Addressing: register
Flags: Z,S,P,CY,AC

ADD M (Add memory)

(A) ←(A) + ((H)(L))

The content of the memory location whose address is contained in the H and L registers is added to the content of the accumulator. The result is placed in the accumulator.

1	0	0	0	0	1	1	0

Cycles: 2
States: 7
Addressing: reg. indirect
Flags: Z,S,P,CY, AC

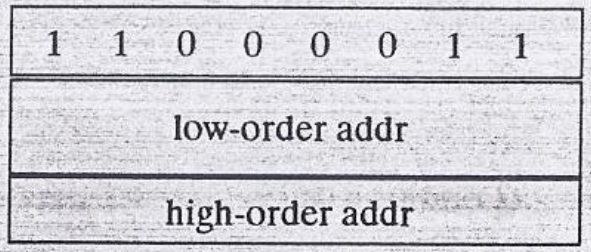
8085 Instruction Set

CONDITION	CCC
NZ - not zero (Z = 0)	000
Z - zero (Z=1)	001
NC - no carry (CY=0)	010
C - carry (CY=1)	011
PO - parity odd (P=0)	100
PE - parity even (P=1)	101
P - plus (S=0)	110
M - minus (S=1)	111

JMP addr (Jump)

(PC) ← (byte 3) (byte 2)

Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.



Cycles: 3
States: 10
Addressing: immediate
Flags: none

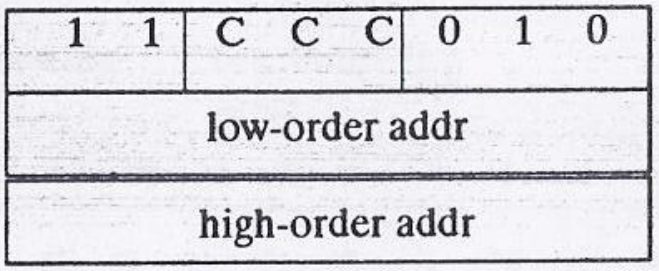
8085 Instruction Set

J condition addr (Conditional jump)

If (CCC),

(PC) ←(byte 3)(byte 2)

If the specified condition is true, control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction; otherwise, control continues sequentially.

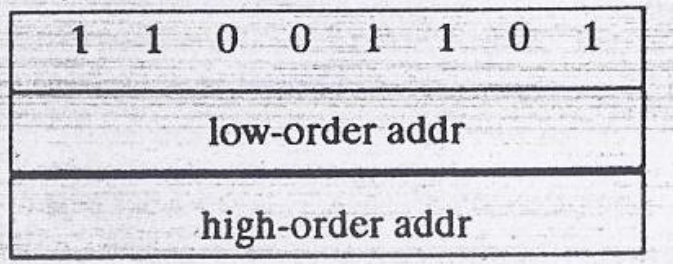


Cycles:	2/3 (8085), 3 (8080)
States:	7/10 (8085), 10 (8080)
Addressing:	immediate
Flags:	none

8085 Instruction Set

CALL addr (Call)
((SP) - 1) ←(PCH)
((SP) - 2) ←(PCL)
((SP) - (SP) - 2
(PC) ←(byte 3) (byte 2)

The high-order eight bits of the next instruction address are moved to the memory location whose address is one less than the content of register SP. The low-order eight bits of the next instruction address are moved to the memory location whose address is two less than the cotent of register SP. the content of register SP is decremented by 2. Control is transferred to the instruction whose address is specified in byte 3 and byte 2 of the current instruction.



Cycles:	5
States:	18 (8085), 17 (8080) immediate/
Addressing:	reg. indirect
Flags:	none

Computer Program

- Write a program segment to add 100 integers.

```
sum = 0;
for (i=100; i>0; i--)
        sum = sum + a[i];
```

Programming 8085

- Write a program to add 100 (64H) integers that are stored in consecutive memory locations stating from address 3000H and store the result in memory location 4000H

Assembly Program: 8085

Label	Assembly Code		
	MVI B 64H		
	LXI H 3000H		
	MVI A 00H		
Loop	ADD M		
	INX H		
	DCR B		
	JNZ Loop		
	STA 4000H		
	HLT		

Assembly Program with M/C Code: 8085

Label	Assembly	M/C Code	M/C Code in Hex
	MVI B 64H	00 000 110	06
	LXI H 3000H	00 10 0001	21
	MVI A 00H	00 111 110	3E
Loop	ADD M	10000 110	86
	INX H	00 10 0011	23
	DCR B	00 000 101	05
	JNZ Loop	11 000 010	C2
	STA 4000H	00 110 010	32
	HLT	0111 0110	76

Memory Location of the program: 8085

Memory	Op-Code	M/C Code	M/C in Hex, Data
1000	MVI B 64H	00 000 110	06 64
1002	LXI H 3000H	00 10 0001	21 00 30
1005	MVI A 00H	00 111 110	3E 00
1007	ADD M	10000 110	86
1008	INX H	00 10 0011	23
1009	DCR B	00 000 101	05
100A	JNZ Loop	11 000 010	C2 07 10
100D	STA 4000H	00 110 010	32 00 40
1010	HLT	0111 0110	76

Computer Program

- Write a program segment to add 100 integers.

```
sum = 0; i=100
while (i>0){
        sum = sum + a[i];
        i--;
}
```

Computer Program

- Write a program segment to add 100 integers.

```
sum = 0; i=100
while (i>0){
        b = a[i];
        sum = sum + b;
        i--;
}
```

Computer Program

- Write a program segment to add 100 integers.

```
sum = 0; i=100
while (i>0){
        b = a[i];
        sum = sum + b;
        i--;
}
```

```
sum = 0; i=100
while (i>0){
        b = a[i];
        i--;
        sum = sum + b;
}
```

Memory Location of the program: 8085

Memory	Op-Code	M/C Code	M/C in Hex, Data
1000	MVI B 64H	00 000 110	06 64
1002	LXI H 3000H	00 10 0001	21 00 30
1005	MVI A 00H	00 111 110	3E 00
1007	ADD M	10000 110	86
1008	INX H	00 10 0011	23
1009	DCR B	00 000 101	05
100A	JNZ Loop	11 000 010	C2 07 10
100D	STA 4000H	00 110 010	32 00 40
1010	HLT	0111 0110	76

If you interchange INX H and DCR B
Any effect in the result???

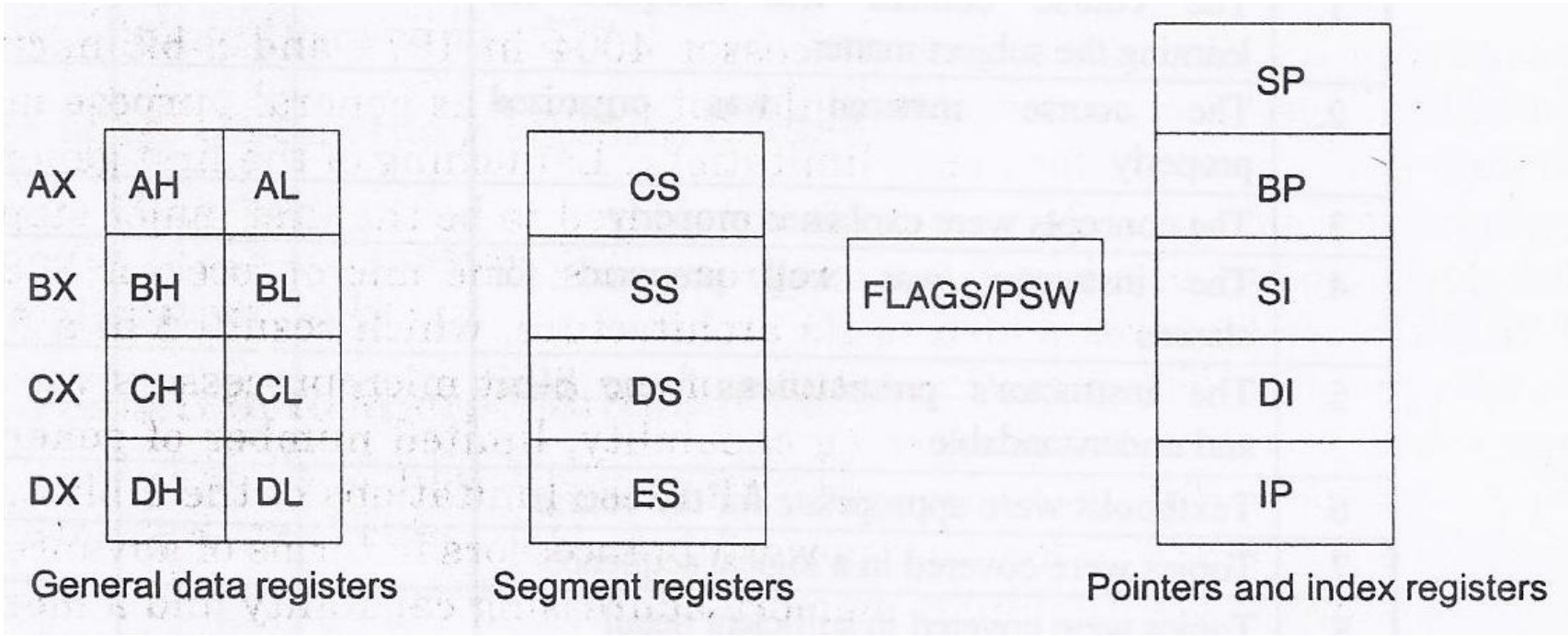
Input/Output

- How to bring the data into memory
  - Through Input instruction
  - Need to specify the address of the device
  - Addressing Scheme

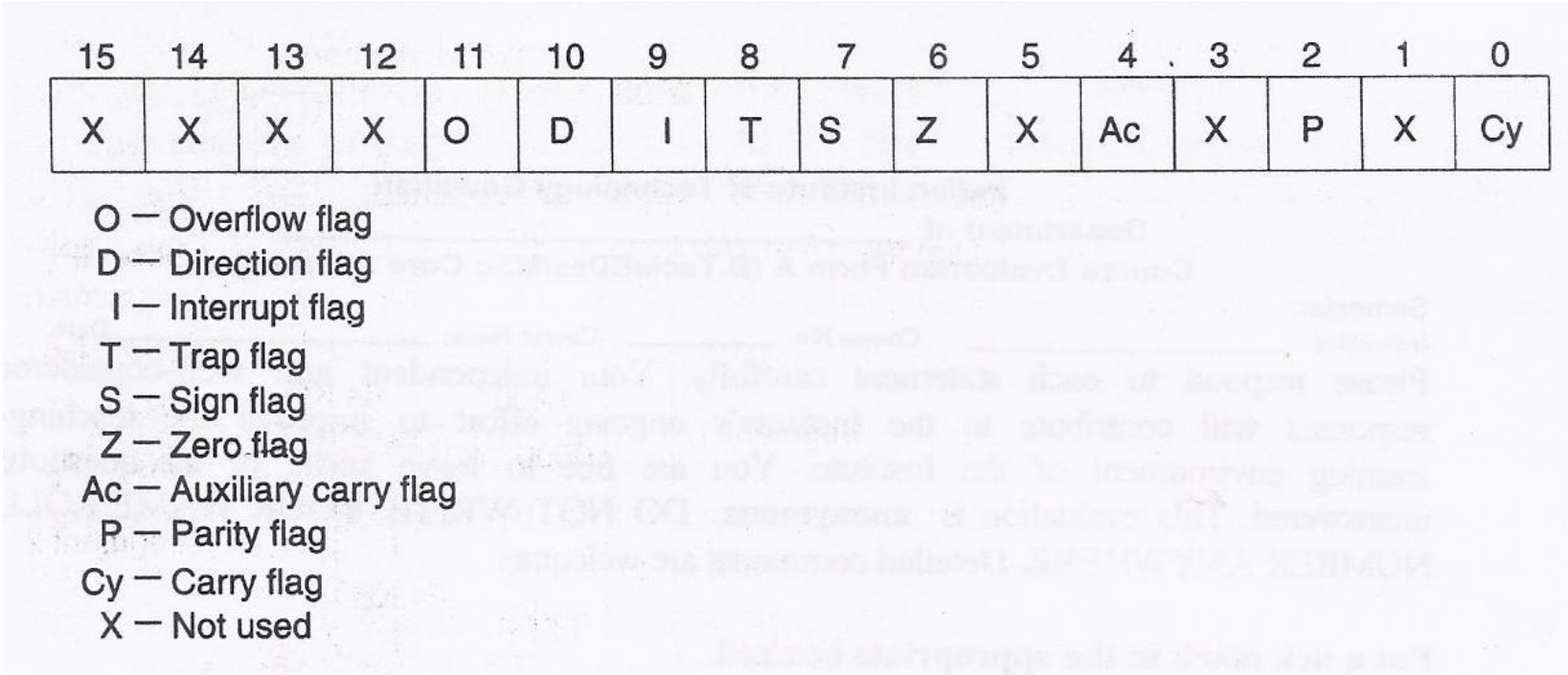
Assembly Program

Label	Assembly Code		
	MVI B 64H		
	LXI H 3000H		
Loop	IN 10H	Address of the device is 10H	
	MOV M A		
	INX H		
	DCR B		
	JNZ Loop		
	HLT		

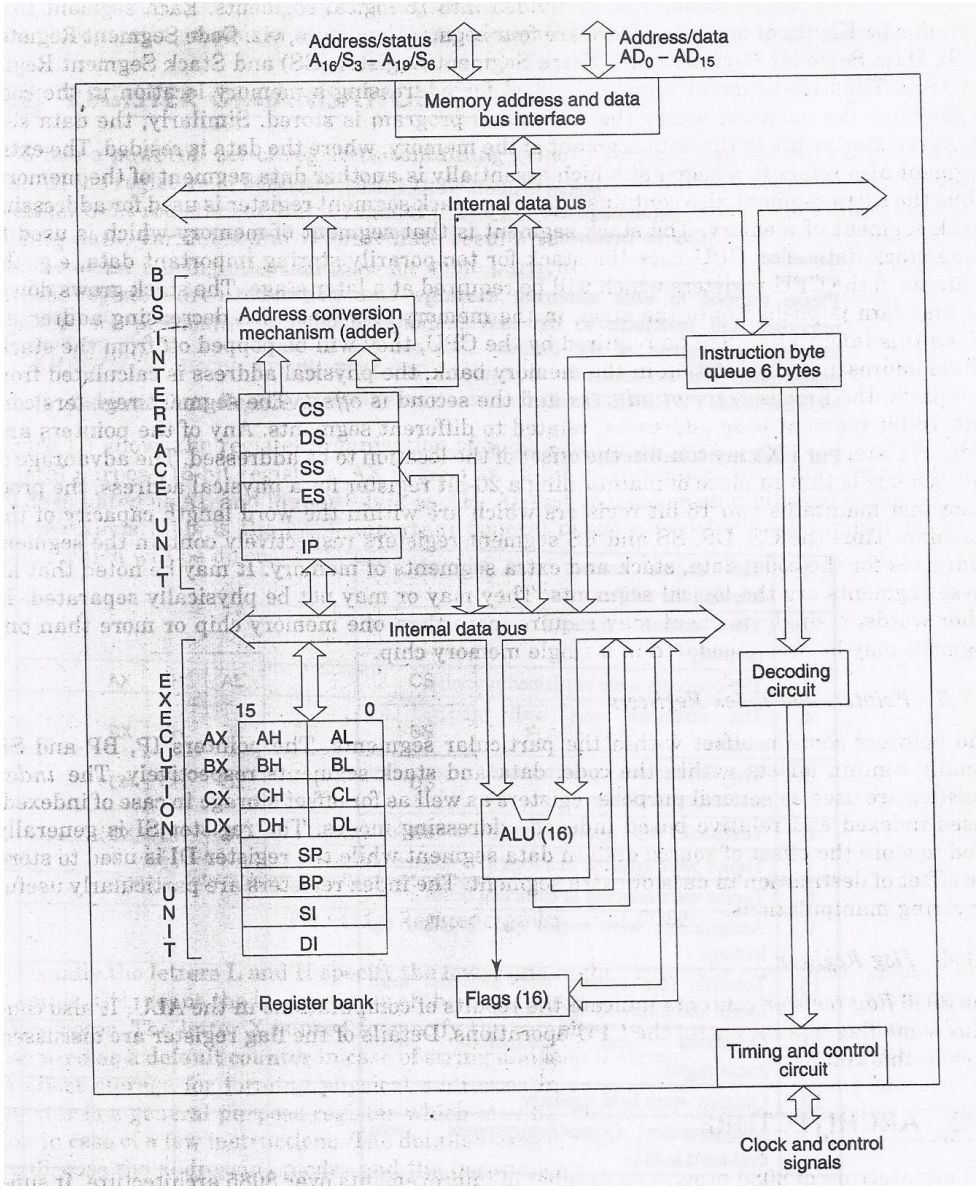
8086: Registers



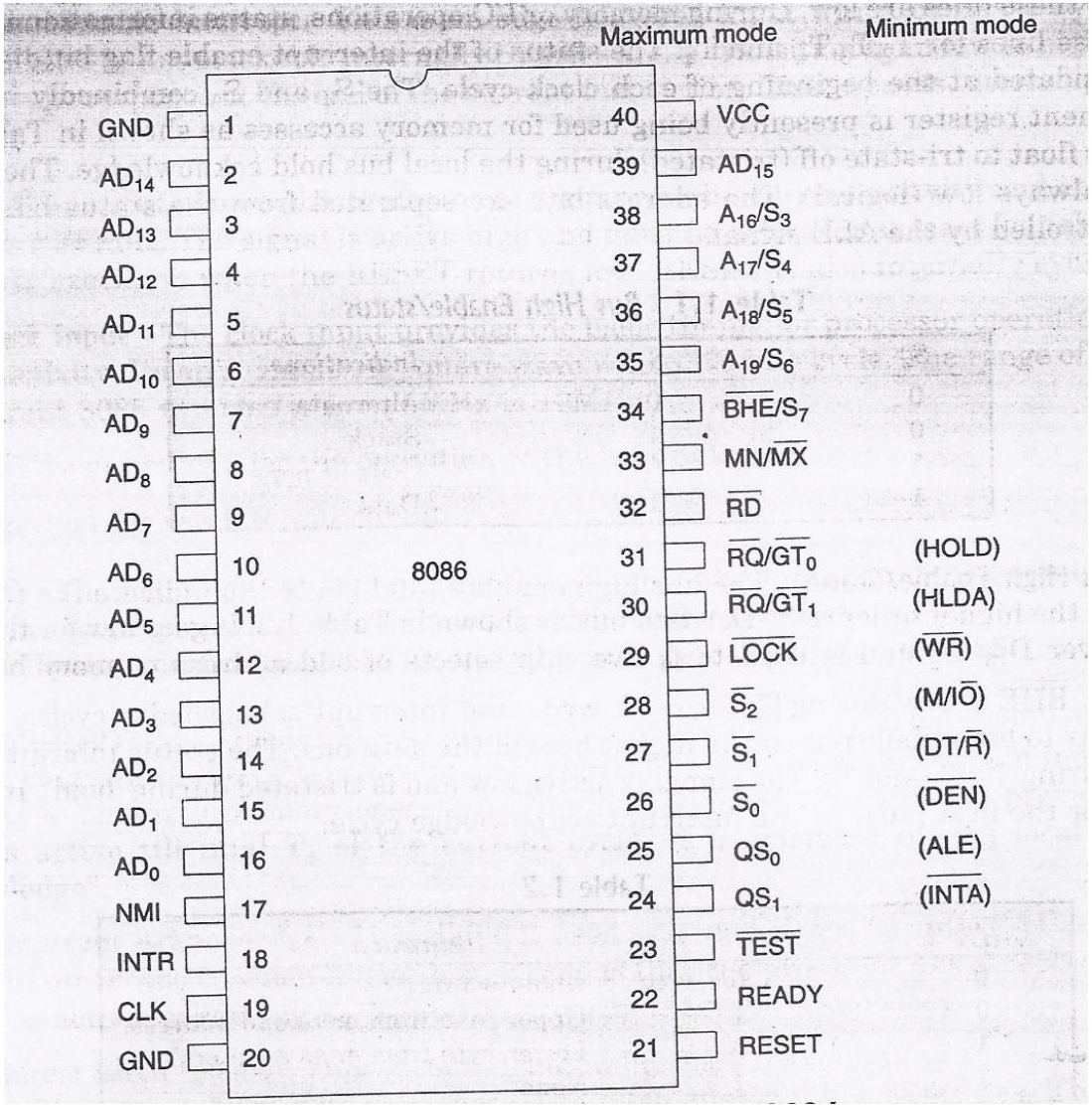
8086: Flags


15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
X X X X O D I T S Z X Ac X P X Cy
O — Overflow flag
D — Direction flag
I — Interrupt flag
T — Trap flag
S — Sign flag
Z — Zero flag
Ac — Auxiliary carry flag
P — Parity flag
Cy — Carry flag
X — Not used

8086: Organization



8086 Pin layout



8086: Memory Address

BHE	A0	Indication
0	0	Whole word
0	1	Upper Byte
1	0	Lower Byte
1	1	None

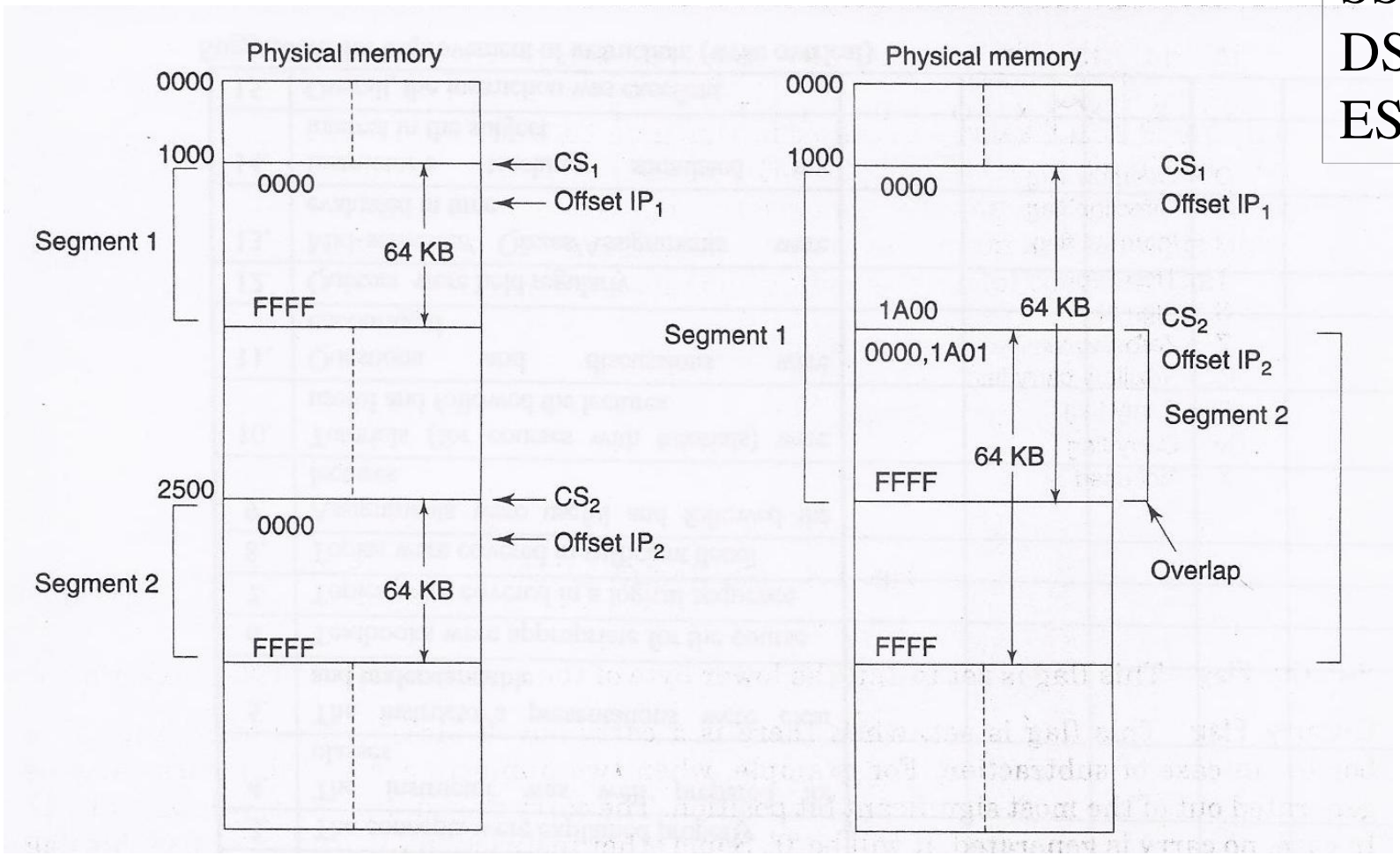
Address	A0
00005	1
00006	0
FFFFE	0
FFFFF	1
96787	1
96788	0

Odd bank (8)	Even bank(8)	Address
		00000
		00002
		00004
		00006
00009	00008	
		FFFFE

Address Bus: 20 bits
Data Bus: 16 bits

8086: Memory Segment

CS
SS
DS
ES



8086: Physical Memory Address

Segment address ⟶ 1005H
Offset address ⟶ 5555H
Segment address ⟶ 1005H ⟶ 0001 0000 0000 0101
Shifted by 4 bit positions ⟶ 0001 0000 0000 0101 0000
+
Offset address ⟶ 0101 0101 0101 0101
Physical address ⟶ 0001 0101 0101 1010 0101
1 5 5 A 5