

Physical address = Starting add  $\times$  10 + offset  
 Address bus : 20 bit  
 Data bus : 16 bit

## 8086

### CISC      RISC

- Complex      • Reduced
- Hardware      • Software based
- Variable size      • Fixed size inst.
- Small code      • Large code
- More than one clock cycle less      • Single clock cycle highly
- Pipelined      • Pipeline
- High power      • Low power
- Less Ram      • More Ram
- Slower      • Faster

### SRAM      DRAM

- Faster      • Slower
- Static      • Dynamic
- Expensive      • Cheap
- Store in transistors      • Store in capacitors
- Low-density device      • High density
- Less power      • More power
- High data transfer rate      • Low data transfer rate

### Von Neumann

- Ancient
- Same memory add. for instruction & data
- Two cycles for 1 instruction
- Cheaper
- Cannot access instruction & read/write at same time
- Common bus for data & instruction transfer

### Harvard

- Modern
- Separate memory address for inst. & data
- Single cycle
- Costly
- Can access at same time
- Separate bus for data & instruc. transfer

### GPR:

$16 = 8 + 8$	bit
AX	AH AL
BX	BH BL
CX	CH CL
DX	DH DL

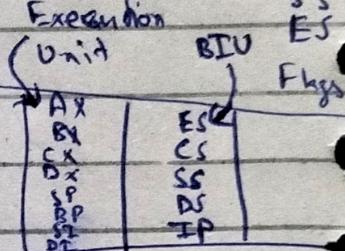
Accumulator  
Base  
Count  
Data

### SR:

CS : code  
DS : data  
ES : extra  
SS : stack

IP  $\rightarrow$  CS  
SP  $\rightarrow$  SS (+top)  
SI  $\rightarrow$  DS

SRR : SP & BP ] pointer  
base SI ] index  
source DI IP destination instruction pointer



### Addressing Mode

#### Register : AX, BX

#### Immediate : AX, 16H

#### Direct : [1234H], BL write real

#### Register Indirect : [BX], CL

#### Register Relative : BL, [BX+4] base

#### Base plus Index : BL, [BX+SI]

#### Base Relative plus Index : BL, [BX+SI+4]

#### Flags : CF, PF, AF, ZF, SF, OF

CMP:	ZF	SF
a <b>&lt;</b> b	1	0
a <b>&gt;</b> b	0	0
a <b>=</b> b	0	1

AX AH AL

remainde quotient

#### Logical: AND : clear

OR : set

NOT : ~

XOR : Invert

TEST : if same, ZF=0  
if diff, ZF=1

CX: counter / decrement logic

Leading 0's : MSB side

Trailing 0's : LSB side

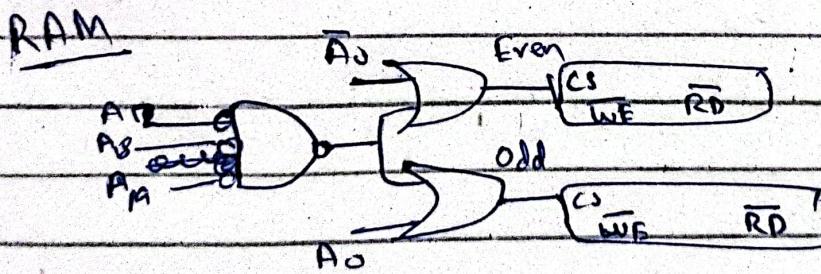
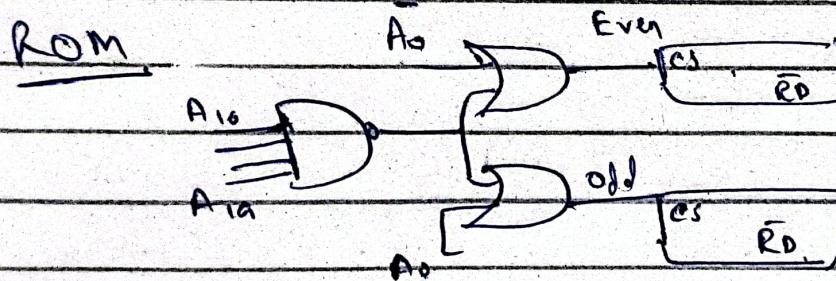
## Memory Interface

- 1) Address lines
- 2) Data lines
- 3) Control lines
- 4) Chip Select

64KB ROM  
Add.: 64K  
 $2^6 \times 2^{10}$   
→ 16 lines  
Data: 8 lines (constant)  
ROM only read  
Control: 1 line  $\bar{RD}$

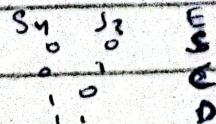
128KB RAM  
Add: 128K  
 $2^7 \times 2^{10}$   
→ 17 lines  
Data: 8 lines  
Control: 2 lines  $\bar{RD}$  &  $\bar{WR}$

	A <sub>16</sub>	A <sub>15</sub>	A <sub>14</sub>	A <sub>13</sub>	A <sub>12</sub>	A <sub>11</sub>	A <sub>10</sub>	A <sub>9</sub> -4	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Address
Even RAM	0	0	0	0	0	0	0	0	0	0	0	0	00000H
Odd RAM	0	0	0	0	1	1	0	0	1	1	1	0	
Odd RAM	0	0	0	0	0	0	0	0	0	0	0	1	
Even RAM	0	0	0	0	1	1	1	1	1	1	1	1	
Even RAM	1	1	1	1	1	0	0	0	0	0	0	0	
Odd RAM	1	1	1	1	1	1	1	1	1	1	1	0	
Odd RAM	1	1	1	1	1	1	0	0	0	0	0	1	
Odd RAM	1	1	1	1	1	1	1	1	1	1	1	1	FFFFFH



A<sub>0</sub> - A<sub>15</sub> : Address/Data lines  
when ALE=1, Address  
ALE=0, Data

A<sub>16</sub>/S<sub>3</sub> - A<sub>19</sub>/S<sub>6</sub> : Address / Switch lines



BHE : Bus high enable

Enable D<sub>8</sub>-D<sub>15</sub>

MN/MX : When connected, minimum  
When GND '1', maximum

RD : read

READY : when 0, wait  
when 1, continue

INTR : when IF=1, INTR=1, interrupt

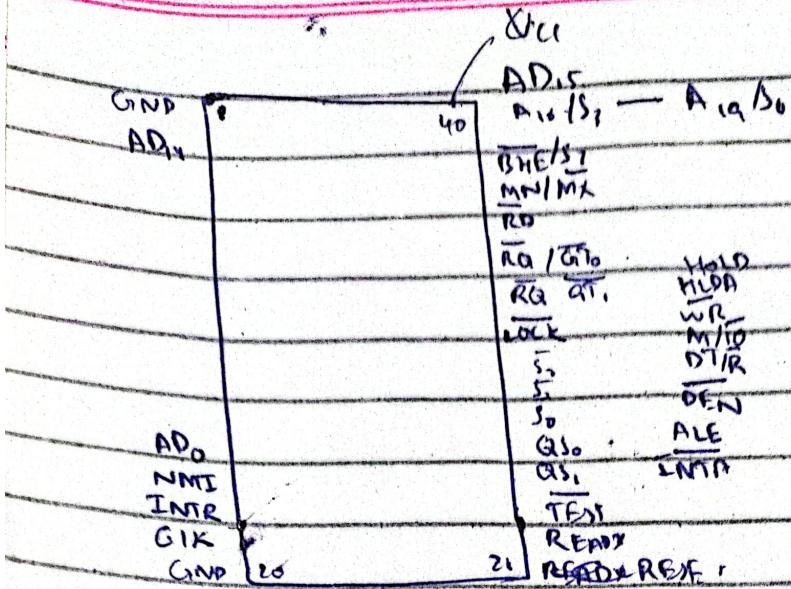
IF=0, INTR=1, no interrupt

NMI : interrupt

TEST : when 0, wait

RESET : When 1 for 4 cycles, reset

CLK : clock



V<sub>CC</sub> 5V

GND

AD<sub>0</sub>-AD<sub>15</sub> (ALE=1  $\Rightarrow$  Address, ALE=0  $\Rightarrow$  Data)

A<sub>16</sub>/S<sub>7</sub>-A<sub>19</sub>/S<sub>6</sub>  $\Rightarrow$   $\begin{matrix} S_4 \\ 0 \end{matrix}$   $\begin{matrix} S_3 \\ 0 \end{matrix}$   $\begin{matrix} E \\ 0 \end{matrix}$   
 $\overline{\text{BHE}}/\text{S}_7 \Rightarrow \text{P}_8-\text{D}_0$  enable

CLK

INTR ( $\text{IF} \neq 1, \text{INTR}=1 \Rightarrow \text{Interrupt}$ )

NMI ( $\text{INTR}=1 \Rightarrow \text{Interrupt}$ )

RESET (When 1 for 4 cycles, reset\*)

READY (When 0, wait)

TEST (When 0, wait)

Q<sub>50</sub> (ALE)

Q<sub>51</sub> (INTA) (interrupt acknowledged)

S<sub>2</sub> WR M/I/O When 0, address bus = IO address  
When 1, address bus = Memory address

S<sub>1</sub> DT/R Data transmit/receive enable (external bus)

S<sub>0</sub> DEN Data bus enable (external)

LOCK WR Write

RG/GT<sub>0</sub> HOLD Hold

RQ/GT<sub>1</sub> HLDA Hold acknowledge  
read

RD MN/MX Minimum when 1, maximum when 0

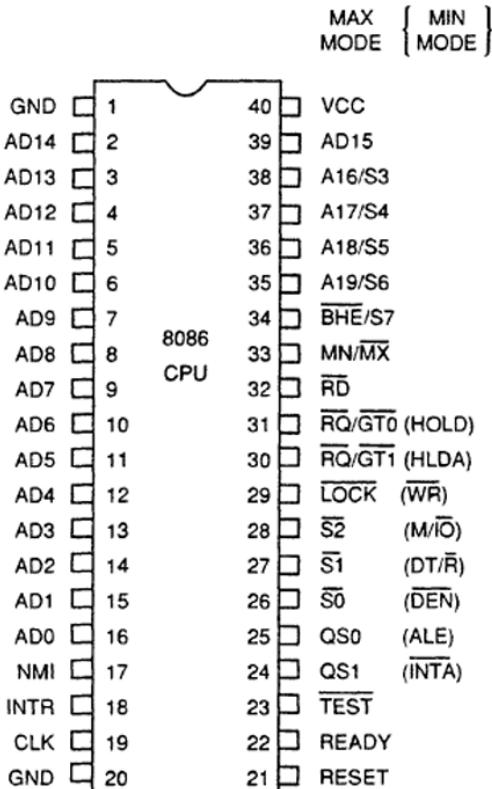


Figure (1-1): The pin layout of 8086 microprocessor.

TABLE 6-1 Conditional jump instructions.

Assembly Language	Tested Condition	Operation
JA	Z = 0 and C = 0	Jump if above
JAE	C = 0	Jump if above or equal
JB	C = 1	Jump if below
JBE	Z = 1 or C = 1	Jump if below or equal
JC	C = 1	Jump if carry
JE or JZ	Z = 1	Jump if equal or jump if zero
JG	Z = 0 and S = 0	Jump if greater than
JGE	S = 0	Jump if greater than or equal
JL	S != O	Jump if less than
JLE	Z = 1 or S != O	Jump if less than or equal
JNC	C = 0	Jump if no carry
JNE or JNZ	Z = 0	Jump if not equal or jump if not zero
JNO	O = 0	Jump if no overflow
JNS	S = 0	Jump if no sign (positive)
JNP or JPO	P = 0	Jump if no parity or jump if parity odd
JO	O = 1	Jump if overflow
JP or JPE	P = 1	Jump if parity or jump if parity even
JS	S = 1	Jump if sign (negative)
JCXZ	CX = 0	Jump if CX is zero
JECDZ	ECX = 0	Jump if ECX equals zero
JRCXZ	RCX = 0	Jump if RCX equals zero (64-bit mode)

<u>AAA</u>	<u>CMPSB</u>	<u>JAE</u>	<u>JNBE</u>	<u>JPO</u>	<u>MOV</u>	<u>RCR</u>	<u>SCASB</u>
<u>AAD</u>	<u>CWD</u>	<u>JB</u>	<u>JNC</u>	<u>JS</u>	<u>MOVSB</u>	<u>REP</u>	<u>SCASW</u>
<u>AAM</u>	<u>DAA</u>	<u>JBE</u>	<u>JNE</u>	<u>JZ</u>	<u>MOVSW</u>	<u>REPE</u>	<u>SHL</u>
<u>AAS</u>	<u>DAS</u>	<u>JC</u>	<u>JNG</u>	<u>LAHF</u>	<u>MUL</u>	<u>REPNE</u>	<u>SHR</u>
<u>ADC</u>	<u>DEC</u>	<u>JCXZ</u>	<u>JNGE</u>	<u>LDS</u>	<u>NEG</u>	<u>REPNZ</u>	<u>STC</u>
<u>ADD</u>	<u>DIV</u>	<u>JE</u>	<u>JNL</u>	<u>LEA</u>	<u>NOP</u>	<u>REPZ</u>	<u>STD</u>
<u>AND</u>	<u>HLT</u>	<u>JG</u>	<u>JNLE</u>	<u>LES</u>	<u>NOT</u>	<u>RET</u>	<u>STI</u>
<u>CALL</u>	<u>IDIV</u>	<u>JGE</u>	<u>JNO</u>	<u>LODSB</u>	<u>OR</u>	<u>RETF</u>	<u>STOSB</u>
<u>CBW</u>	<u>IMUL</u>	<u>JL</u>	<u>JNP</u>	<u>LODSW</u>	<u>OUT</u>	<u>ROL</u>	<u>STOSW</u>
<u>CLC</u>	<u>IN</u>	<u>JLE</u>	<u>JNS</u>	<u>LOOP</u>	<u>POP</u>	<u>ROR</u>	<u>SUB</u>
<u>CLD</u>	<u>INC</u>	<u>JMP</u>	<u>JNZ</u>	<u>LOOPE</u>	<u>POPA</u>	<u>SAHF</u>	<u>TEST</u>
<u>CLI</u>	<u>INT</u>	<u>JNA</u>	<u>JO</u>	<u>LOOPNE</u>	<u>PUSH</u>	<u>SAL</u>	<u>XCHG</u>
<u>CMC</u>	<u>INTO</u>	<u>JNAE</u>	<u>JP</u>	<u>LOOPNZ</u>	<u>PUSHA</u>	<u>SAR</u>	<u>XLATB</u>
<u>CMP</u>	<u>IRET</u>	<u>JNB</u>	<u>JPE</u>	<u>LOOPZ</u>	<u>RCL</u>	<u>SBB</u>	<u>XOR</u>
<u>IA</u>							

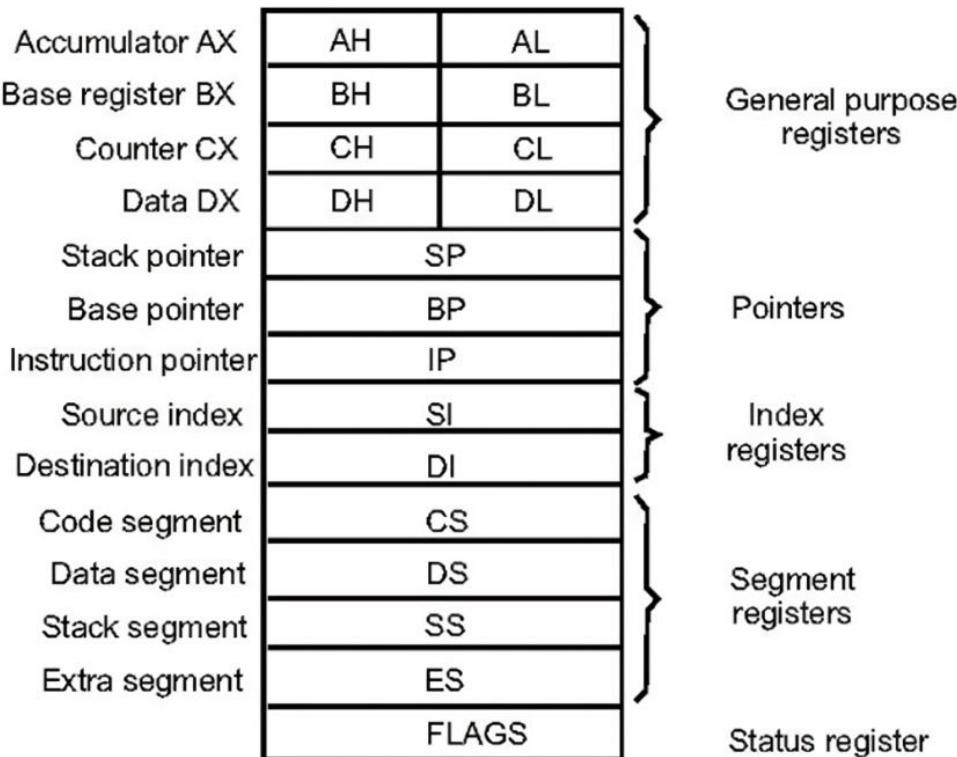
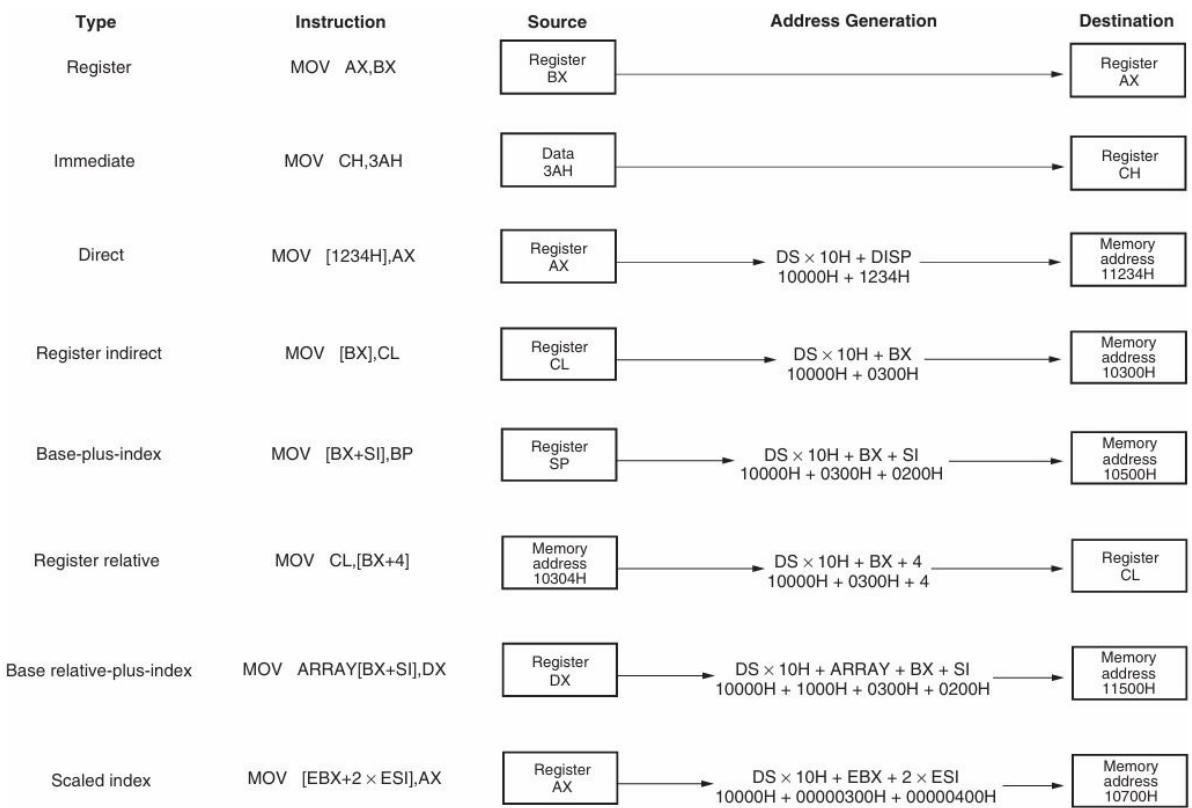


Fig.11.6: Schematic diagram of intel 8086 registers



Notes: EBX = 00000300H, ESI = 00000200H, ARRAY = 1000H, and DS = 1000H

**FIGURE 3–2** 8086–Core2 data-addressing modes.