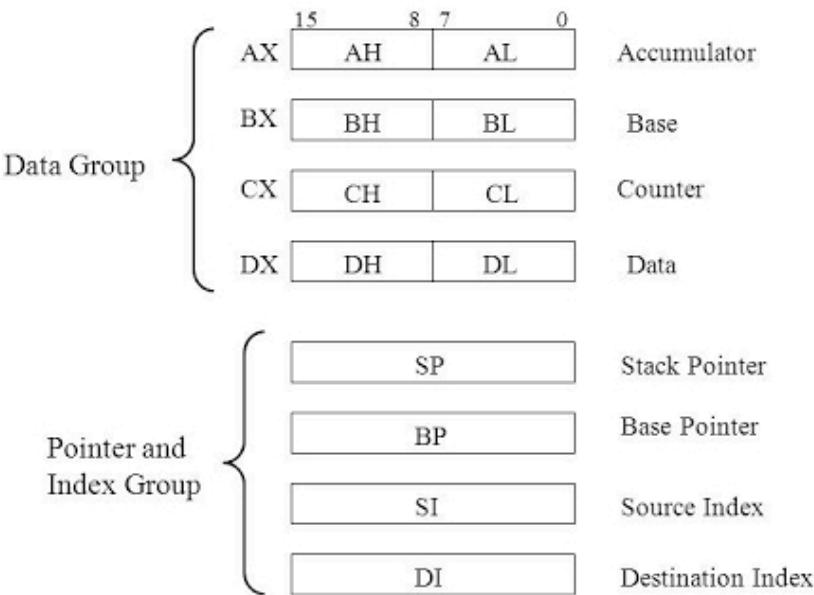


Binary/Decimal	Binary	Hexadecimal
10 Number System	Base 2 Number System	Base 16 Number System
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

1. General Purpose Registers



Segment		Offset Registers	Function
CS		IP	Address of the next instruction
DS		BX, DI, SI	Address of data
SS		SP, BP	Address in the stack
ES		BX, DI, SI	Address of destination data (for string operations)

OPCODE	OPERAND	DESTINATION	EXAMPLE
AND	D, S	D = D AND S	AND AX, 0010
OR	D, S	D = D OR S	OR AX, BX
NOT	D	D = NOT of D	NOT AL
XOR	D, S	D = D XOR S	XOR AL, BL
TEST	D, S	performs bit-wise AND operation and affects the flag register	TEST [0250], 06
ADD	D, S	D = D + S	ADD AX, [2050]
ADC	D, S	D = D + S + prev. carry	ADC AX, BX
SUB	D, S	D = D – S	SUB AX, [SI]
SBB	D, S	D = D – S – prev. carry	SBB [2050], 0050
MUL	8-bit register	AX = AL * 8-bit reg.	MUL BH
MUL	16-bit register	DX AX = AX * 16-bit reg.	MUL CX

XCHG	REG, memory memory, REG REG, REG	<p>İki operandın değerlerini yer değiştirir.</p> <p>Algoritma: operand1 < - > operand2</p> <p>Örnek: MOV AL, 5 MOV AH, 2 XCHG AL, AH ; AL = 2, AH = 5 XCHG AL, AH ; AL = 5, AH = 2 RET</p>
PUSH	REG SREG memory immediate	<p>16 bit değer stack'a saklanır. Store 16 bit value in the stack.</p> <p>Not: PUSH immediate sadece 80186 ve üstü işlemcilerde kullanılır.</p> <p>Algoritma: SP = SP - 2 SS:[SP] (top of the stack) = operand</p> <p>Örnek: MOV AX, 1234h PUSH AX POP DX ; DX = 1234h RET</p>
POP	REG SREG memory	<p>Stack'tan 16 bit değer alır ve operanda aktarır.</p> <p>Algoritma: operand = SS:[SP] (stack'ın en üstü) SP = SP + 2</p> <p>Örnek: MOV AX, 1234h PUSH AX POP DX ; DX = 1234h RET</p>
CMP	REG, memory memory, REG REG, REG memory, immediate REG, immediate	<p>Karşılaştırma yapar. Soldaki operand'ı sağdakinden çıkartır ancak operandlar değişmez sadece bayrak bitleri değişir.</p> <p>Algoritma: operand1 - operand2</p> <p>OF, SF, ZF, AF, PF, CF bayrakları sonuca göre yeniden düzenlenir.</p> <p>Örnek: MOV AL, 5 MOV BL, 5 CMP AL, BL ; AL=5, ZF=1 (eşit) RET</p>
HLT	-	<p>Program kapatılır.</p> <p>Örnek: MOV AX, 5 HLT</p>
INC	REG memory	<p>Artırma işlemi yapar.</p> <p>Algoritma: operand = operand + 1</p> <p>Örnek: MOV AL, 4 INC AL ; AL = 5 RET</p>
DEC	REG memory	<p>Azaltma işlemi yapar.</p> <p>Algoritma: operand = operand - 1</p> <p>Örnek: MOV AL, 255 ; AL = 0FFh (255 or -1) DEC AL ; AL = 0FEh (254 or -2) RET</p>

CALL	procedure name etiket 4-byte adres	<p>Bir prosedürü çağırır ve dönüş adresini (IP) stack'a push eder. 4 byte adres kullanılarak çağırma yapılabilir. 1234h:5678h adresinde ilk değer segment ikinci değer offset adresini belirtir. Bu far çağırmadır ve IP ile birlikte CS stack'a push edilir.</p> <p>Örnek:</p> <pre>#make_COM# ORG 100h ; for COM file. CALL p1 ADD AX, 1 RET ; return to DOS. p1 PROC ; procedure declaration. MOV AX, 1234h RET ; return to caller. p1 ENDP</pre>
IN	AL, im.byte AL, DX AX, im.byte AX, DX	<p>AL veya AX'e porttan giriş alır. İkinci operand port numarasıdır.</p> <p>Second operand is a port number. Eğer 255 üstündeki portlara erişim gerekiyorsa DX register'ı kullanılır.</p> <p>Örnek:</p> <pre>IN AX, 4 ; get status of traffic lights. IN AL, 7 ; get status of stepper-motor.</pre>
OUT	im.byte, AL im.byte, AX DX, AL DX, AX	<p>AL veya AX register'ı verilen porta gönderilir. İlk operand port numarasıdır. Eğer 255'in üstündeki portlara erişim gerekirse DX register'ı kullanılır.</p> <p>Örnek:</p> <pre>MOV AX, 0FFFh ; Turn on all OUT 4, AX ; traffic lights. MOV AL, 100b ; Turn on the third OUT 7, AL ; magnet of the stepper-motor.</pre>

JNC	Elde yok ise dallan
JC	Elde var ise dallan
JNZ	Sıfır değil ise dallan
JZ	Sıfır ise dallan
JG	Büyük ise dallan
JNG	Büyük değil ise dallan
JGE	Büyük ve eşit ise dallan
JS	Küçük ise dallan
JNS	Küçük değil ise dallan
JA	Üzeri ise dallan (ABOVE)
JNA	Üzeri değil ise dallan
JE	Eşit ise dallan
JNE	Eşit değil ise dallan

1) SHR : Sağa Kaydırma

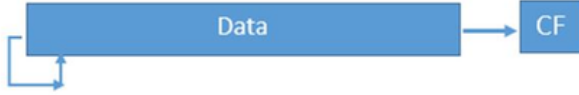
SHR talimatı 'Sağa Kaydırma'nın kısaltmasıdır. Bu komut, sol uçtan aynı sayıda (kaydırılan bitler) sıfırları ekleyerek, kayıttaki bahsedilen bitleri birer birer sağ tarafa kaydırır. Kaydırılan en sağdaki bit, Taşıma Bayrağında (CF) saklanır.



SHR Register, Bits to be shifted
SHR AX, 2

2) SAR : Aritmetik Sağa Kaydırma

SAR talimatı 'Aritmetik Sağa Kaydır' anlamına gelir. Bu komut, yazmaçtaki bahsedilen bitleri birer birer sağ tarafa kaydırır, ancak sol uçtan sıfırları eklemek yerine MSB'yi geri yükler. Kaydırılan en sağdaki bit, Taşıma Bayrağında (CF) saklanır.



SAR Register, Bits to be shifted
SAR BX, 5

3) SHL : Sola Kaydır

SHL talimatı 'Sola Kaydır'ın kısaltmasıdır. Bu komut, sağ uçtan aynı sayıda (kaydırılan bitler) sıfırları ekleyerek, kayıttaki bahsedilen bitleri birer birer sol tarafa kaydırır. Kaydırılan en soldaki bit, Taşıma Bayrağında (CF) saklanır.



SHL Register, Bits to be shifted
SHL AX, 2

4) SAL : Aritmetiği Sola Kaydır

SAL talimatı 'Aritmetik Sola Kaydır' ifadesinin kısaltmasıdır. Bu talimat SHL ile aynıdır.



SAL Register, Bits to be shifted
SAL CL, 2

5) ROL : Sola Döndür

ROL talimatı 'Sola Döndür'ün kısaltmasıdır. Bu komut, yazmaçtaki bahsedilen bitleri birer birer sola döndürür, öyle ki döndürülen en soldaki bit, yine yazmaçta en sağdaki bit olarak depolanır ve aynı zamanda Taşıma Bayrağında (CF) saklanır.

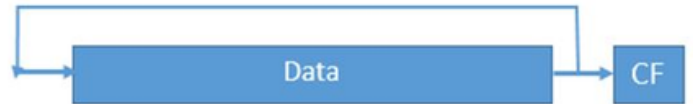
ROL Register, Bits to be shifted
ROL AH, 4



6) ROR : Sağa Döndür

ROR talimatı 'Sağa Döndür' anlamına gelir. Bu komut, yazmaçtaki bahsedilen bitleri birer birer sağ tarafa döndürür, öyle ki döndürülen en sağdaki bit yine yazmaçta MSB olarak depolanır ve aynı zamanda Taşıma Bayrağında (CF) saklanır.

ROR Register, Bits to be shifted
ROR AH, 4



7) RCL : Taşımayı Sola Döndür

Bu komut, yazmaçtaki bahsedilen bitleri birer birer sola döndürür, böylece döndürülen en soldaki bit Taşıma Bayrağında (CF) depolanır ve CF'deki bit, yazmaçta LSB olarak hareket eder.

RCL Register, Bits to be shifted
RCL CH, 1



8) RCR : Sağa Döndür Taşıma

Bu komut, kayıttaki bahsedilen bitleri sağ tarafa döndürür, böylece döndürülen en sağdaki bit, Taşıma Bayrağında (CF) depolanır ve CF'deki bit, kayıttaki MSB olarak hareket eder.

RCR Register, Bits to be shifted
RCR BH, 6

