MİKROİŞLEMCİLER (BLM202)

10. DERS

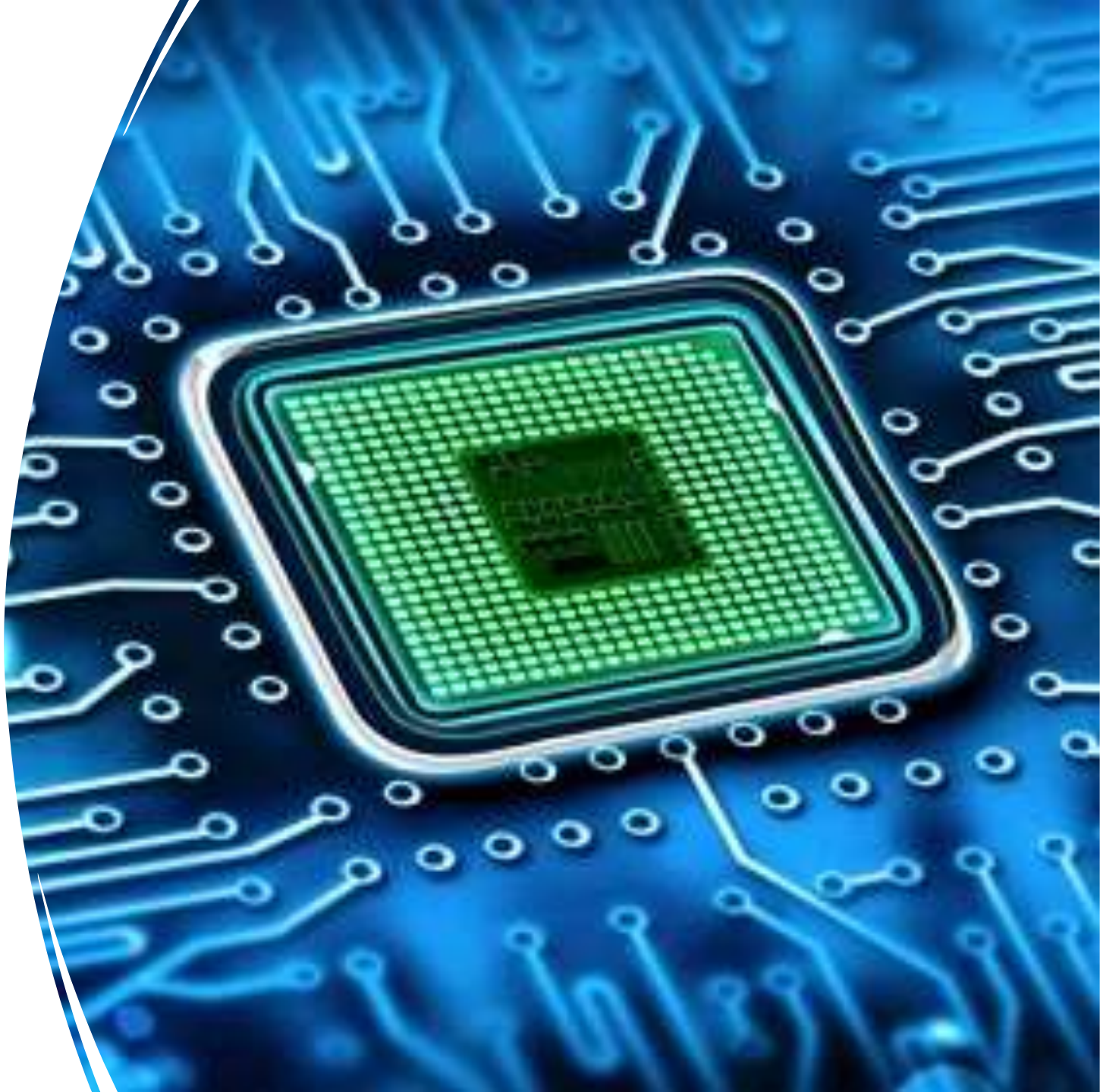
Dr. Bilgin YAZLIK, RTTP, PMP

NEVŞEHİR HACI BEKTAŞ VELİ ÜNİVERSİTESİ

BİLGİSAYAR MÜHENDİSLİĞİ

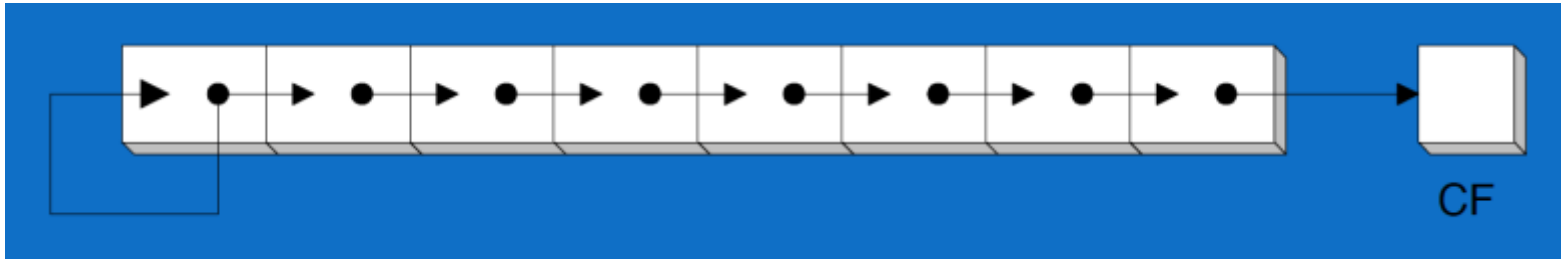
10. DERS

- SAR VE SAL Komutları
- Kontrol Komutları



SAR&SAL

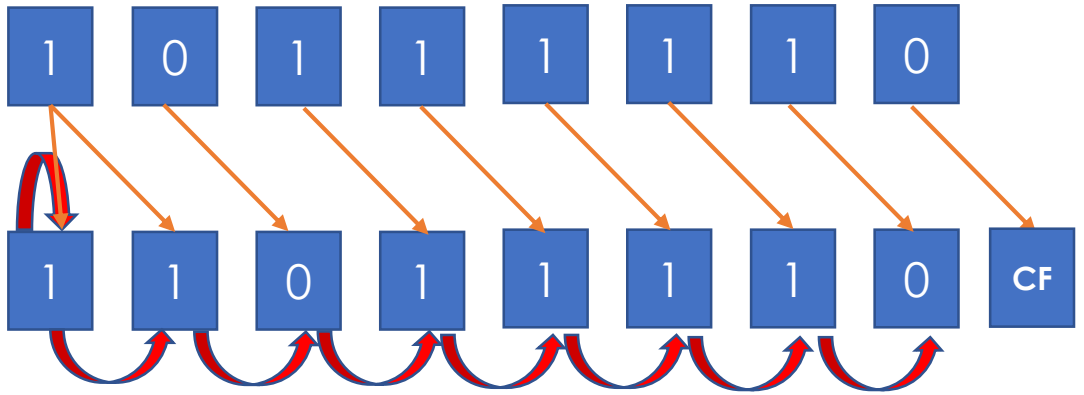
- **SAR (Shift Aritmetik Right):** Bitleri sağa doğru 1’er bit kaydırır. En soldaki bitin değeri de kaydırılır fakat eski değeri korunur. Bu bit işaretli sayılarda sign biti olarak kullanılır.



- **SAL (Shift Aritmetik Left):** Bitleri sola doğru 1’er bit kaydırır.

SAR Örnek

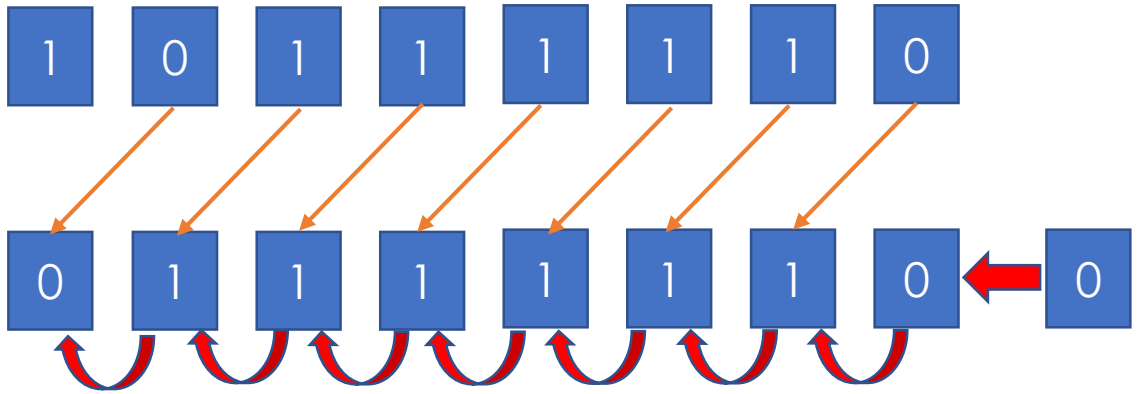
- MOV AL,1011 1110B;BE
- SAR AL,1;1101 1111B;DF
- SAR AL,1;1110 1111B;EF
- SAR AL,1;1111 0111B;F7
- SAR AL,1;1111 1011B;FB
- SAR AL,1;1111 1101B;FD
- SAR AL,1;1111 1110B;FE
- SAR AL,1;1111 1111B;FF



İŞARET BİTİ KORUNUR

SAL Örnek

- MOV AL,1011 1110B;BE
- SAL AL,1;0111 1100B;7C
- SAL AL,1;1111 1000B;F8
- SAL AL,1;1111 0000B;F0
- SAL AL,1;1110 0000B;E0
- SAL AL,1;1100 0000B;C0
- SAL AL,1;1000 0000B;80
- SAL AL,1;0000 0000B;0

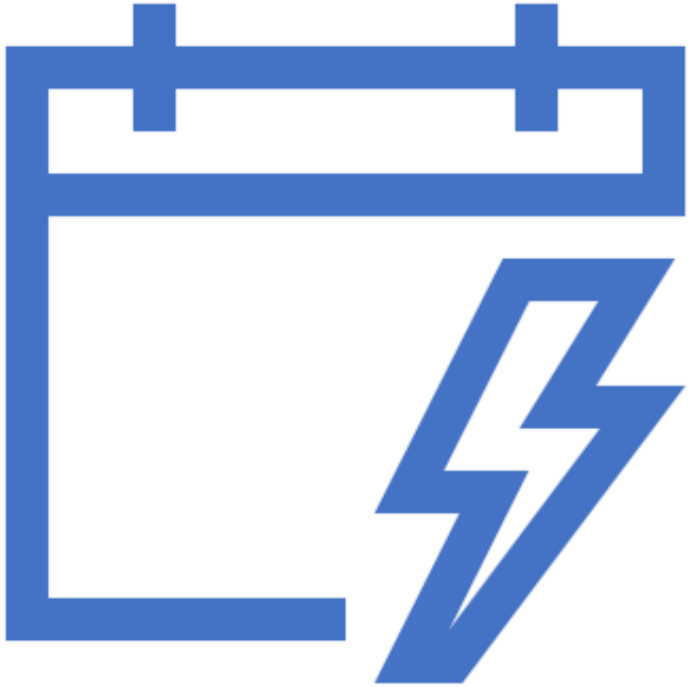


SAL İLE SHL KOMUTU AYNI ŞEKİLDE ÇALIŞIR

KONTROL KOMUTLARI

PROGRAM KONTROL KOMUTLARI

- Program akışını bir noktadan bir başka noktaya yönlendirmek amacı ile kullanılan komutlardır. Bu komutlar aşağıda listelenmiştir.
- Şartsız Dallanma Komutu: JMP
- Döngü Komutları: LOOP
- Karşılaştırma Komutu: CMP
- Şartlı Dallanma: JE,JZ,JNZ….
- Alt Program Çağrısı: CALL
- Bayraklar İle İlgili Komutlar: CLC,STC...

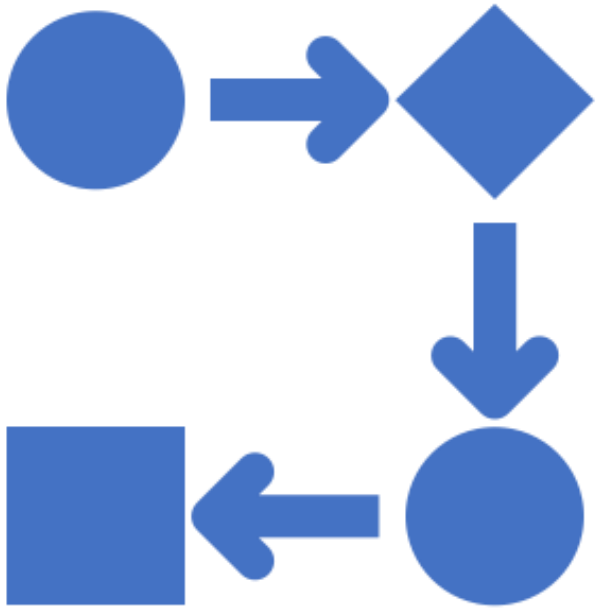


ŞARTSIZ DALLANMA KOMUTU: JMP

- JMP

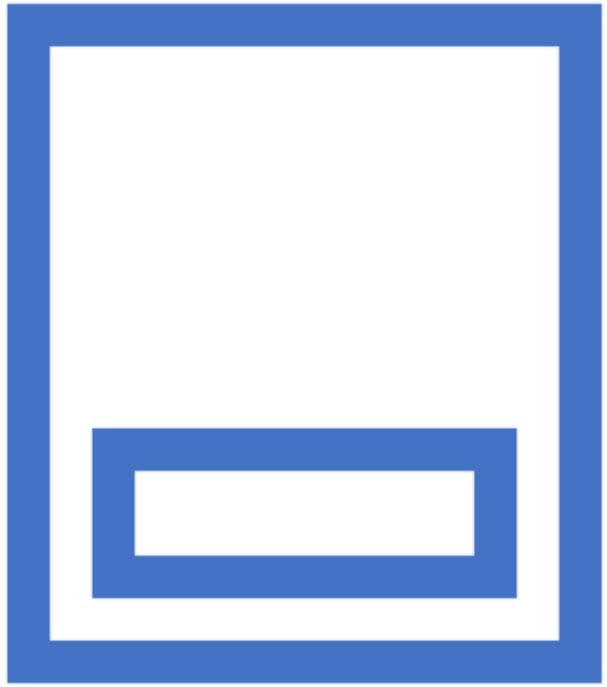
Programı belirtilen etiketin olduğu yere dallandırmakta ve program buradan çalışmaya devam etmektedir.

- JMP Hedef



DÖNGÜ KOMUTU: LOOP

- LOOP komutu, genellikle bir iş birden fazla yapılacağı zaman CX registeri ile birlikte döngü kurmayı sağlamaktadır.
- CX registerı içinde döngünün adedi tutulur. Her loop komutu çalıştığında CX'in değeri 1 azalır ve CX sıfırlandığında döngü biter.
- MOV CX,5
- Topla:.....
- .....
- LOOP Topla



KARŞILAŞTIRMA KOMUTU: CMP

- CMP komutunun kullanımı
- CMP *deger1*, *deger2*

CMP register, register;
CMP AX, BX
CMP register, memory;
CMP AX, SONUC
CMP register, immediate;
CMP AX,5

CMP memory, register;
CMP SONUC, AX
CMP memory, immediate;
CMP SONUC,5

- CMP komutu kullanıldığı zaman aşağıdaki etkilenen bayrakların durumu verilmiştir.
- **Not**: CMP 50 (veri), 50 (veri) şeklinde **kullanılamaz** aşağıda küçük ve büyük olma durumlarını açıklamak için o şekilde örnek verilmiştir.
- CMP AH, 50; kullanılabilir

	C	Z	S
CMP 50,60	1	0	1
CMP 50,50	0	1	0
CMP 50,40	0	0	0

ŞARTLI DALLANMA KOMUTLARI

- Şartlı Dallanma Komutları genellikle bir CMP komutunu takiben program akışını başka bir noktaya kaydırmak amacıyla kullanılır. Şartlı dallanma komutları bayrakların durumuna bakarak hangi noktaya (etiket-label) gidileceğini belirlemektedir.
- Koşullu dallanmada dallanma aralığı 8 bit ile sınırlıdır (8 bit ile maksimum 255 sayısı gösterilebilir)
- -128 veya +127 değerinden daha uzak noktalara dallanma söz konusuysa koşulsuz dallanma komutları kullanılmalıdır.

Bayrakların Durumunu Test Etmek İçin Kullanılan Koşullu Dallanma Komutları

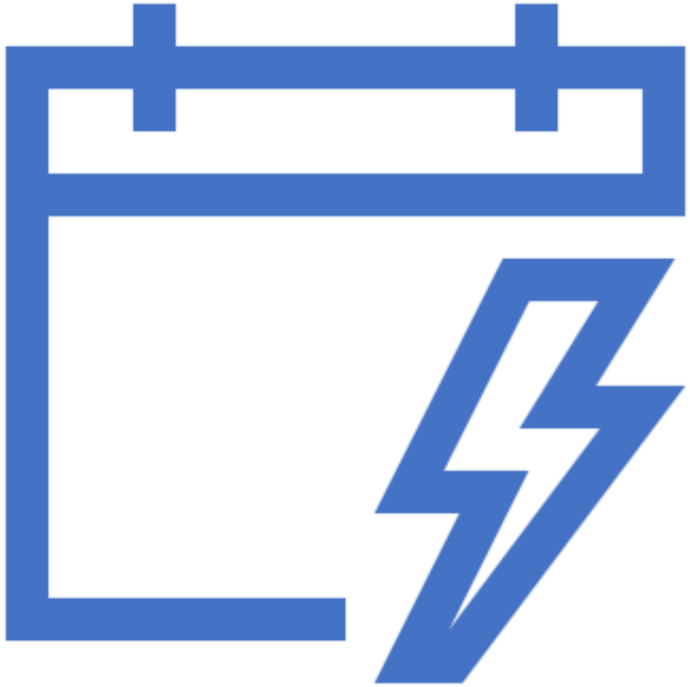
Komut	Tanımlama	Durum	Karşıtı
JZ, JE	Sıfır (eşit) - Jump if Zero (Equal)	Z = 1	JNZ, JNE
JC, JB, JNAE	Carry (küçük; eşitten büyük değil) - Jump if Carry (Below, Not Above Equal)	C = 1	JNC, JNB, JAE
JS	Yönlü - Jump if Sign	S = 1	JNS
JO	Overflow - Jump if Overflow	O = 1	JNO
JPE, JP	Çift parity - Jump if Parity Even	P = 1	JPO, JNP
JNZ, JNE	Jump if Not Zero (Not Equal) (sıfır değil ise)	Z = 0	JZ, JE
JNC, JNB, JAE	Jump if Not Carry (küçük değil; eşitten büyük - Not Below, Above Equal)	C = 0	JC, JB, JNAE
JNS	Yönlü değil - Jump if Not Sign	S = 0	JS
JNO	Overflow değil - Jump if Not Overflow	O = 0	JO
JPO, JNP	Tek parity - Jump if Parity Odd (No Parity)	P = 0	JPE, JP

İşaretsiz Sayılarda Şartlı Dallanma Komutları

Instruction	Description	Condition	Opposite
JE, JZ	Eşit ise - Jump if Equal (=) Sıfır ise - Jump if Zero	Z = 1	JNE, JNZ
JNE, JNZ	Eşit değil ise - Jump if Not Equal (≠) Sıfır değil ise - Jump if Not Zero	Z = 0	JE, JZ
JA, JNBE	Büyük ise - Jump if Above (>) Küçük veya eşit değil ise - Jump if Not Below or Equal	C = 0 and Z = O	JNA, JBE
JBE, JNA	Küçük veya eşit ise - Jump if Below or Equal (<=) Büyük değil ise - Jump if Not Above	C = 1 or Z = 1	JNBE, JA
JB, JNAE, JC	Küçük ise - Jump if Below (<) Büyük veya eşit değil ise - Jump if Not Above or Equal Carry ise - Jump if Carry	C = 1	JNB, JAE, JNC
JAE, JNB, JNC	Büyük veya eşit ise - Jump if Above or Equal (>=) Küçük değil ise - Jump if Not Below Carry değil ise - Jump if Not Carry	C = 0	JB, JNAE, JC

İşaretli Sayılarda Şartlı Dallanma Komutları

Instruction	Description	Condition	Opposite
JE, JZ	Eşit is e - Jump if Equal (=) Sıfır ise - Jump if Zero	Z = 1	JNE, JNZ
JNE, JNZ	Eşit değil ise - Jump if Not Equal (≠) Sıfır değil ise - Jump if Not Zero	Z = 0	JE, JZ
JG, JNLE	Büyük ise - Jump if Greater (>) Küçük veya eşit değil ise - Jump if Not Less or Equal (not <=)	Z = 0 and S = O	JNG, JLE
JL, JNGE	Küçük ise - Jump if Less (<) Büyük veya eşit değil ise - Jump if Not Greater or Equal	S ≠ O	JNL, JGE
JGE, JNL	Büyük veya eşit ise - Jump if Greater or Equal (>=) Küçük değil ise - Jump if Not Less	S = O	JNGE, JL
JLE, JNG	Küçük veya eşit ise - Jump if Less or Equal (<=) Büyük değil ise - Jump if Not Greater	Z = 1 or S ≠ O	JNLE, JG



Alt Program Çağrısı: CALL & RET

- Adına prosedür denilen program parçaları ana program içerisinde her çağrılmak istendiğinde şartsız dallanma komutu CALL kullanılır.
- Prosedürün sonunda bulunan RET komutuyla program kaldığı yere geri döner.

Komut	Etki
CLC	C=0
CMC	C=C’
STC	C=1
CLD	D=0
STD	D=1
STI	I=1
CLI	I=0
LAHF	AH=bayrak
SAHF	bayrak=AH

Bayraklar ile İlgili Komutlar

- Bazı komutların istenen şekilde çalışabilmesi için ön koşul olarak bayrakların değerlerinin ayarlanması gerekir.
- Sık ihtiyaç duyulan bayrakların değerini bağımsız olarak değiştiren komutlara karşılık, seyrek ihtiyaç duyulan bayrak değerlerinde değişim için AH registeri kullanılır.

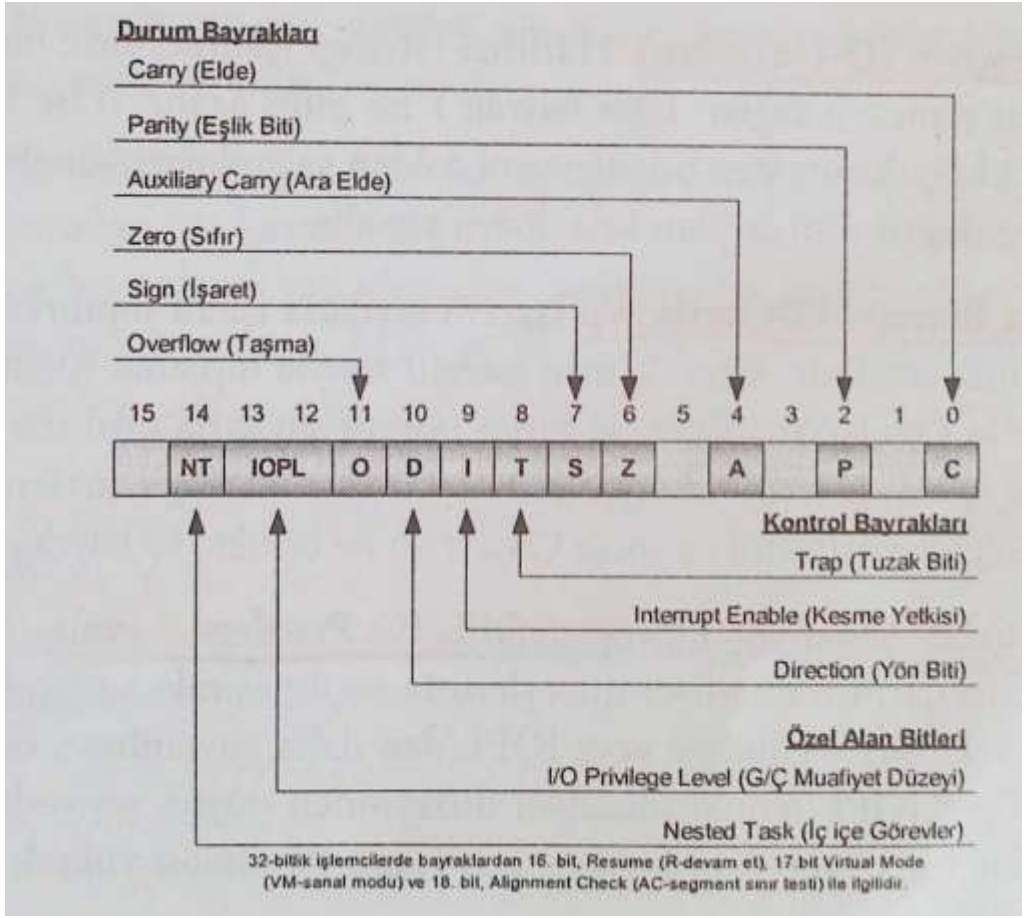
BAYRAKLAR (FLAGS)

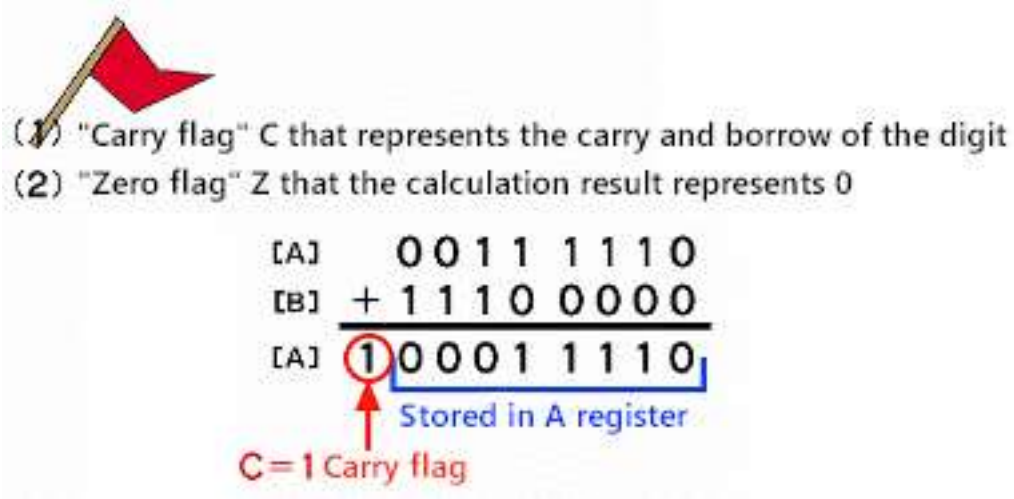
- Elde Bayrağı (C-Carry): Eğer toplama sonucunda elde, çıkarma sonucunda borç ortaya çıkıyorsa C=1 aksi taktirde 0 olur. Aynı zamanda C bayrağı kaydırma ve yönlendirme işlemleri sonucunda kaydedicinin MSB bitinden veya LSB bitinden düşen verileri üzerinde tutar ve karşılaştırma işlemlerinin sonucunu yansıtır. Ayrıca C bayrağı çarpma işlemi için sonuç göstericisi gibi hareket eder.

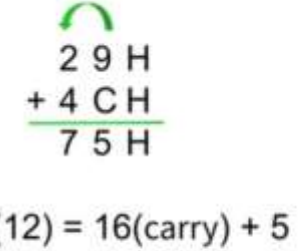
- Eşlik biti (P-Parity): İşlemin sonucunda kaydedicideki mantıksal birlerin sayısı çift ise P=1 aksi halde P=0 olur. Eşlik biti genelde veri iletişiminde karşılıklı verilerin güvenli iletilip iletilmediğinin kontrolünde kullanılır.

- Yardımcı Elde Bayrağı: (AC-Auxilary Carry): Elde bayrağı ile aynı işlemi görür fakat sadece 3. bitten bir fazlalık ortaya çıkarsa bu bayrak 1 aksi durumda 0 olur. AC bayrağı paketlenmiş ondalık verilerin işlenmesinde çok kullanışlıdır.

BAYRAKLAR







Auxiliary Carry Flag in Hexadecimal Representation

Let us consider the same example in binary representation.

29H = 0010 1001

+4CH = 0100 1100

75H = 0111 0101

^ here there is carry generated and forwarded to next nibble, so the auxiliary carry flag is set to one.

BAYRAKLAR

•Sıfır Bayrağı (Z-Zero): İşlem sonunda sonuç 0 ise Z=1 aksi halde Z=0 olur. Mesela bu işlem sonunda AX kaydedicisindeki değer 0000 ise sıfır bayrağı 1 olur diğer durumlarda bayrak 0 kalır.

•İşaret bayrağı (S-Sign): İşaretli sayılarla yapılan işlemlerde bu bayrak anlam ifade etmektedir. Eğer aritmetik mantık, kaydırma ve yönlendirme işlemleri negatif sonuç üretiyorsa S=1 aksi halde S=0 olur. Diğer bir deyimle S bayrağı sonucun 8 bit veya 16 bit olmasına bakılmaksızın MSB bitini yansıtır.

•Tuzak Bayrağı (T-Trap): Hata ayıklama işlemlerinde komutların adım adım işlenmesi maksadıyla kullanılır. Bayrak 1 yapıldığında Debug işlemi yapmak için komutler tek tek çalıştırılır.

•Kesme Yetkilendirme bayrağı (I-Interrupt Enable): Sisteme bağlı harici cihazlardan gelen kesme taleplerine izin verir. I bayrağının 0 olması kesme isteklerine cevap verilmemesini sağlar. Ancak 1=1 olduğunda tekrar istekler göz önüne alınır.

BAYRAKLAR

•Yön Bayrağı (D-Direction): String işlemlerinde indis kaydedicisinin ileri yada geri hareket etmesini sağlar. Eğer bayrak 1 ise indis azalır, 0 ise indis değeri artar. Eğer D=0 ise, işlemci küçük adresten büyüğe yani soldan sağa doğru yönelir. Eğer D=1 ise, büyük adresten küçüğe doğru yani sağdan sola doğru yönelir. Eğer D=1 ise büyük adresten küçük adrese yani sağdan sola doğru yönelir.

•Taşma bayrağı(O-Owerflow): İşaretli sayılarla işlem yapılırken bir hatanın ortaya çıkması durumunda gözükür. Eğer iki aynı sayıyla toplama işlemi yapılıyor ve sonuç farklı işaretli çıkıyorsa 0=1 olur. Eğer matematik bir işlem sonucunda sonuç kaydedici kapasitesini aşıyorsa C bayrağı ile birlikte O bayrağı da 1 olur.

•Giriş/Çıkış Muafiyet düzeyi (IOPL-IO Privilege Level): Korumalı mod operasyonlarında G/Ç cihazlarının muafiyet düzeylerinin seçilmesinde kullanılır. Eğer o andaki muafiyet düzeyi yüksek seçilmişse veya IOPL'den daha güvenilirse, G/Ç herhangi bir engellemesiz çalışır.

BAYRAKLAR

- İçiçe Geçmiş Görevler (NT-Nested Task): Korumalı mod operasyonlarında o andaki görevin başka bir görevle iç içe girmesi işlemidir. Görev başka bir görevle yazılım tarafından iç içe girdirildiğinde bu bayrak 1 olur.

- İşleme devam (R-Resume): Hata ayıklama işlemlerinde (Debug), bir sonraki işlenecek komuta devam edilmesinin kontrolünde kullanılır.

- Sanal Mod(VM-Virtual Mode): Korumalı mod sisteminde sanal mod işleminin seçilmesinde kullanılır. Sanal mod, DOS sisteminde belleğin birkaç parçaya bölümlenmesini sağlar.
- Segment Sınır Tespiti(AC-Alignment Check): Eğer word veya doubleword tanımlamaları kendilerine uygun adres sınırlarında değilse bu bayrak 1'e kurulur. Bu bayrak sadece 486SX işlemcide kullanılmaktadır.

Örnek: 1'den 100 'e kadar olan sayıların toplamını bulup sonucu SONUC değişkenine atan programı yazınız.

- ŞARTLI DALLANMA İLE ÇÖZÜM

So, the directive .model small tells the assembler that you intend to use the small memory model - one code segment, one data segment and one stack segment - and the values of the segment registers are never changed.

Assembler Directives
.model [tiny | small | compact | medium | large | huge]
.data (defines the start of data segment)
.code (defines the start of code sector)
.stack n (defines the size of stack segment)
@data (data segment allocated by OS.)

```
.MODEL SMALL
.STACK 64
.DATA
SONUC DW ?
.CODE
ANA PROC FAR
MOV AX,@DATA
MOV DS, AX
MOV AX,00
MOV CX,100
BAS:
ADD AX, CX
DEC CX
JNE BAS; Sonuc sifir degilse BASa git
MOV SONUC, AX
MOV AH, 4CH
INT 21H
ANA ENDP
END ANA
```

Örnek: 1’den 100 ‘e kadar olan sayıların toplamını bulup sonucu SONUC değişkenine atan programı yazınız.

- LOOP KOMUTU İLE ÇÖZÜM

```
.MODEL SMALL
.STACK 64
.DATA
SONUC DW ?
.CODE
ANA PROC FAR
MOV AX,@DATA
MOV DS, AX
MOV AX,00
MOV CX,100
BAS:
ADD AX, CX
LOOP BAS; CXi 1 azalt sifira esit degilse BASa git
MOV SONUC, AX
MOV AH,4CH
INT 21H
ANA ENDP
END ANA
```

Örnek 2: 5 ile 100 arasındaki sayıların toplamını bulup, sonucu SONUC değişkenine atan program kodunu yazınız.

```
.MODEL SMALL
.STACK 64
.DATA
SONUC DW ?
.CODE
ANA PROC FAR
MOV AX,@DATA
MOV DS, AX
MOV AX,5
BAS:
ADD BX, AX
INC AX
CMP AX, 100
JBE BAS; AX 100den kucuk ve esitken BASa git
MOV SONUC, BX
MOV AH,4CH
INT 21H
ANA ENDP
END ANA
```

Örnek 3

- ORG 100h
- MOV AX, 5
- MOV BX, 2
- JMP hesapla
- geri: JMP dur ;dur etiketine git
- Hesapla:
- ADD AX, BX ; AX’e BX’i ekle
- JMP geri ; geri etiketine git
- dur:
- RET ; İşletim sistemine dön
- END ; derleyiciyi sonlandır

Örnek 4

- ORG 100h
- MOV AL, 25 ; AL=25
- MOV BL, 10 ; BL=10
- CMP AL, BL ; AL ile BLyi karşılaştır
- JE esit ; eğer AL = BL (ZF = 1) ise esite git
- MOV CL,’H’ ; Buraya gelirse AL <> BL demektir
- JMP dur ; Bu yüzden CLye ‘H’ yükle ve dura git
- esit: ; buraya gelirse
- MOV CL,’E’ ; AL = BL demektir bu yuzden CL’ye ‘E’ yaz
- dur:
- RET
- END

Örnek 5

- ORG 100h
- MOV AL, 25 ; AL=25
- MOV BL, 10 ; BL=10
- CMP AL, BL ; AL ile BL’yi karşılaştır
- JNE esitdegil; AL <> BL (ZF = 0)ise dallan
- JMP esit
- esitdegil:
- MOV CL,‘H' ; buraya geldiyse AL <> BL demektir.
- JMP dur; bu yüzden CL=‘H’ ve dur’a git
- esit: ; buraya geldiyse
- MOV CL,‘E' ; AL = BL demektir bu yüzden CL=‘E’
- dur:
- RET
- END

Kaynaklar

- Feza Buzluca, İTÜ Ders Notları, Bilgisayar Mimarisi
- Wikipedia
- Emel Soylu, Kadriye Öz, Karabük Üniversitesi, Mikroişlemciler Ders Notları
- 1) Bilgisayar Mimarisi – Doç. Dr. Şirzat KAHRAMANLI
- 2) Ders Notları – Yrd. Doç. Dr. Rıfat KURBAN
- Wikipedia
- https://edukedar.com/difference-between-cisc-and-risc/
- Dr. B. B. Hegde First Grade College, Kundapura