

BÖLÜM 5. KOMUT SETİ MİMARİSİ

❖ **Göreceli ve İndis Adresleme Modları için Adres Hesaplama Birimi**

❖ **Yığın (Stack) ve Yığın Göstercisi (Stack Pointer-SP)**

Yığınla İlgili Komutlar

Göreceli ve İndis Adresleme Modları için Adres Hesaplama Birimi

Göreceli ve indis adresleme modlarına sahip komutlardan sonra gelen bir byte'lık ofset değerinin, program sayıcısıyla (PC) veya indeks kaydedicisiyle (IX) toplanarak etkin adresin hesaplanıldığına değinilmişti.

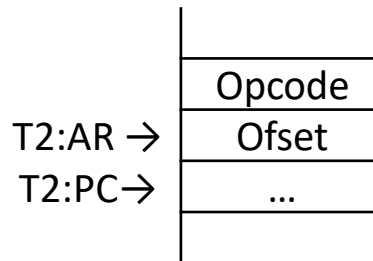
Bu toplama işlemini gerçekleştirmek için elimizde olan aritmetik mantık birimini (ALU) kullanabiliriz. Ancak, bu toplama işleminden önce, akümülatördeki ve data kaydedicisindeki verileri kaybetmememiz için yığına atmamız ve etkin adresi hesapladıktan sonra da tekrar yığından bu bilgileri geri yüklememiz gerekecektir.

Bu işlemler, bu adresleme modlarına sahip komutların icra edilme sürelerini uzatacaktır. Etkin adres hesabının, ALU yerine ayrı bir birimde yapılması, bu süre kaybını önleyecektir. Yol seçiminde kullanılan kod çözücülerin girişlerine 1010_2 verilerek, bu birimin hesapladığı etkin adres bilgisinin yola aktarılması sağlanır.

Göreceli ve İndis Adresleme Modları için Adres Hesaplama Birimi

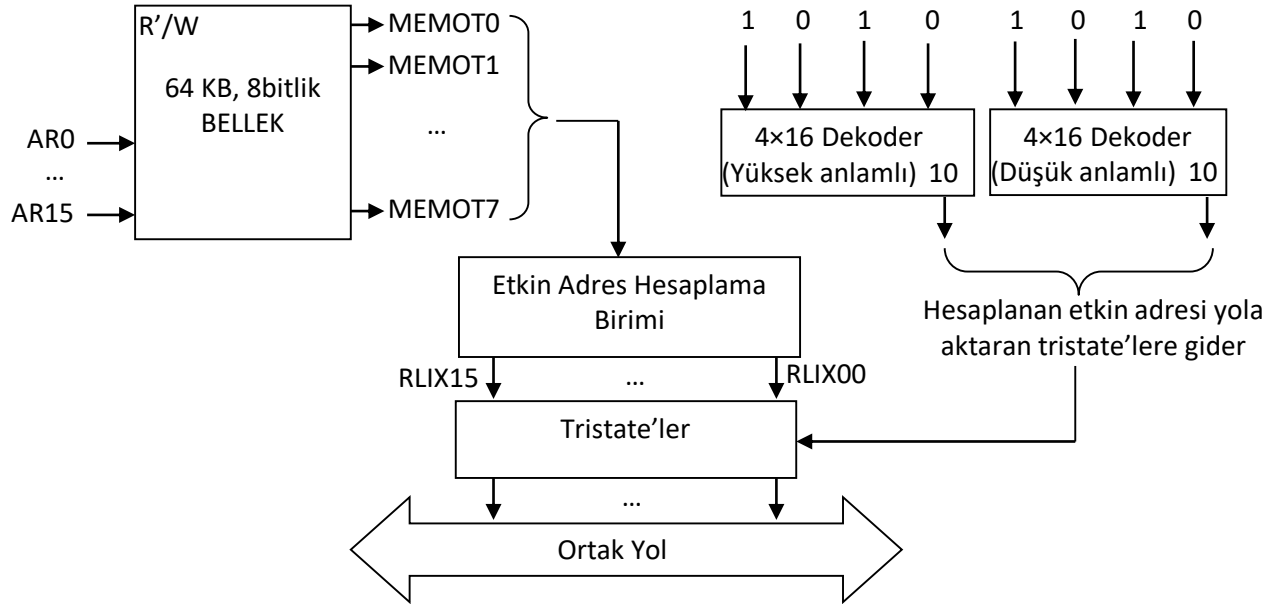
PC ve IX 16 bitlik kaydedicilerdir, bellek ise 8 bitlidir. Etkin adresin hesaplanabilmesi için bu kaydedicilerin komutun ofset kısmıyla toplanması gerekmektedir. İndis adresleme modunu kullanan komutların ofset kısmı işaretli sayı olarak, göreceli moda sahip komutların ofset kısmı ise işaretli sayı (2'ye tümleyen formunda) olarak düşünülmüştür.

Bu adresleme modlarına sahip komutlar işletilirken, T2 zaman diliminde AR, bellekteki ofset değerine işaret ettiğinden ve bellek de AR ile direkt bağlantılı olduğundan dolayı belleğin çıkışındaki ofset değeri, etkin adres hesaplama birimine iletilecektir.



Göreceli ve İndis Adresleme Modları için Adres Hesaplama Birimi

Belleğin çıkışındaki 1 byte'lık ofset değeri MEMOT0 ... MEMOT7 ile isimlendirilmiştir.



Aslında göreceli ve indis adresleme modlarına sahip komutlar olmasa bile, her zaman etkin adres hesaplama birimine bir bilgi gitmektedir. Örneğin direkt adresleme moduna sahip bir komutun T2 zaman diliminde, operandın adres bilgisinin yüksek anlamlı kısmı bu birime iletilir. Her zaman bir etkin adres hesaplanılmasına rağmen, sadece indis ve göreceli adresleme modlarında, etkin adres bilgisi yola aktarılmaktadır.

Etkin Adresin Hesabı

Etkin adres hesaplama biriminde 16 bitlik toplama işlemleri yapılacağından dolayı 8 bitlik ofset değerinin 16 bit olarak ifade edilmesi gerekmektedir.

İndis adresleme modunda, ofset değeri işaretsiz sayı olduğundan dolayı, ofseti 16 bite genişletmek için 8 adet 0 ilave edilir.

Göreceli adresleme modunda ise ofset değeri işaretli sayı olduğundan, işaret bitine bağlı olarak, ofset değeri genişletilir. Yani bellekten okunan ofset değerinin en yüksek anlamlı kısmı (MEMOT7) ile genişletme yapılır. Bu işlemler neticesinde ofset değerinin asıl değerinde hiç bir değişme meydana gelmez.

Etkin Adresin Hesabı

Örnek: Ofset değeri 85h olsun. Etkin adresi her iki adresleme modu için hesaplayalım.

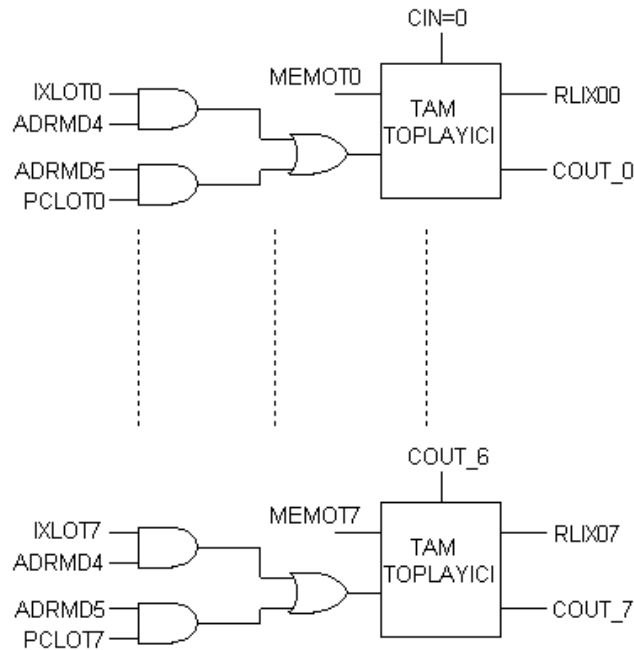
Göreceli adresleme moduna göre bu sayı negatif bir sayıdır. Değeri;
 $85h = 1000\ 0101_2 \rightarrow 2$ 'ye tümleyenini alıp önüne '-' işaretini koyarsak
 $-(0111\ 1011_2) = -123$ 'tür.

PC'nin 1234h olduğunu düşünelim. Bu durumda etkin adres şu şekilde hesaplanır; ofset değerinin işaret biti 1 olduğundan, ofset değerinin soluna 8 adet 1 konulur. Bu durumda ofset değerinin 16 bitlik hali, FF85h olur. Bunun da decimal karşılığı -123'e tekabül etmektedir.

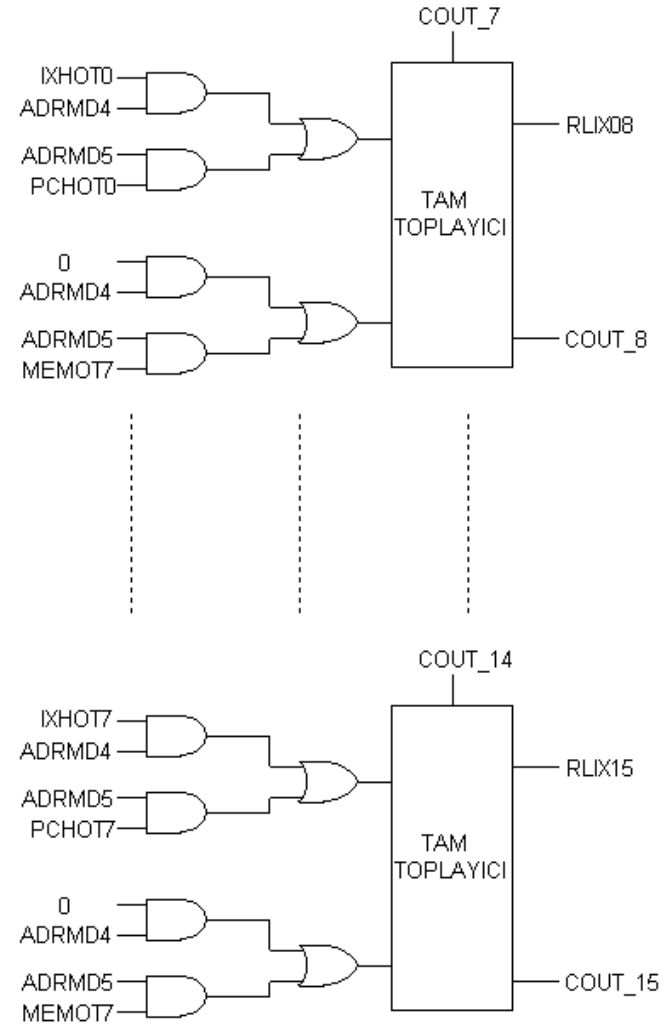
Etkin adres = $1234h + FF85h = 11B9h$ olarak bulunur.

İndis adresleme moduna göre bu sayı işaretsiz düşünülmüş olup decimal 133 değerine karşılık gelmektedir. Etkin adres hesaplanmadan önce ofset değerinin soluna 8 adet 0 konulur. Bu durumda ofset değerinin 16 bitlik hali 0085h olur. IX'in değerini 1234h alırsak, Etkin adres = $1234h + 0085h = 12B9h$ olur.

Etkin Adres Hesaplama Biriminin Donanımsal Yapısı



Düşük anlamlı kısmı



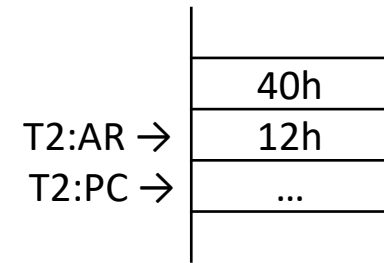
Yüksek anlamlı kısmı

Örnek: İndis moda sahip ADD komutunun işlem adımları

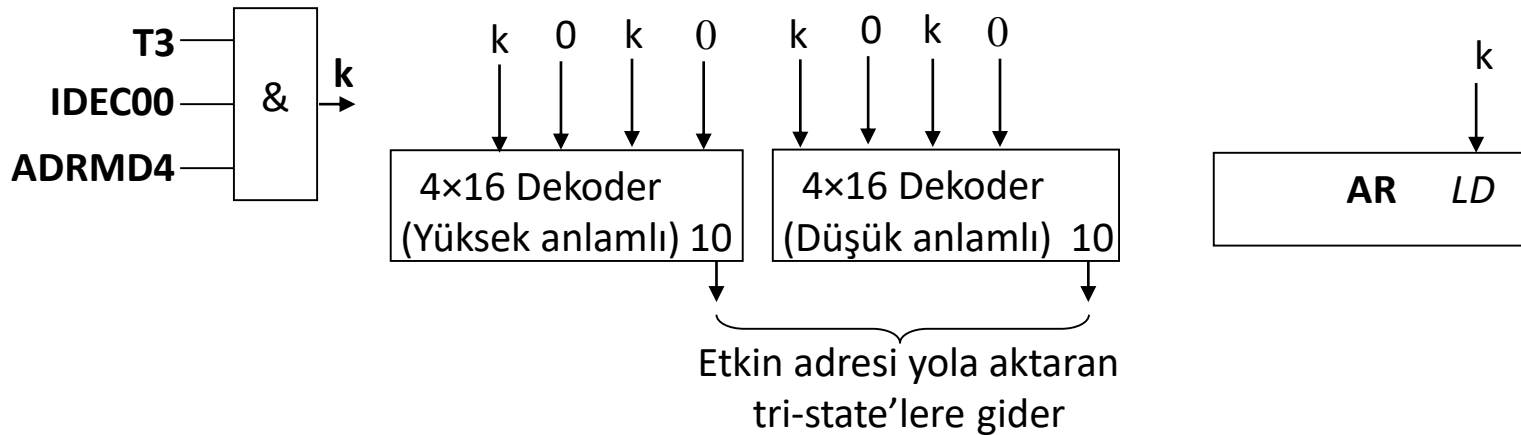
ADD *12h komutu gibi. ‘*’ sembolü indis adreslemeyi temsil etmek için kullanılmıştır. 2 byte’lık bir komuttur. Komuttaki 12h, ofset değeridir. Index kaydedicisinin içeriğine 12h eklenecek ve etkin adres hesaplanacaktır.

İndis mod için ADD komutunun opcode’u $0\ 100\ 0000_2 = 40h$.

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC00.ADRMD4:	$AR \leftarrow \text{Etkin Adres}$
T4. IDEC00.ADRMD4:	$DR_H \leftarrow M[AR], AR \leftarrow AR+1$
T5. IDEC00.ADRMD4:	$DR_L \leftarrow M[AR]$
T6. IDEC00.ADRMD4:	$AC \leftarrow AC+DR, C \leftarrow C_{out}, SC \leftarrow 0$



T3 zamanlama diliminde yapılan mikroişlem adımının donanımsal gerçekleştirimi



Yığın (Stack) ve Yığın Göstericisi (Stack Pointer-SP)

Herhangi bir dallanma olduğunda PC'nin değeri, alt programdan dönüşte kullanılmak üzere yığına atılmalıdır. Kesme geldiğinde ise PC, IX, AC ve CCR kaydedicilerinin içerikleri yığına kaydedilmeli ve kesme programı işletildikten sonra da bu değerler yığından alınarak ilgili kaydedicilere geri yüklenmelidir.

Bu işlem için, hafızada ilk olarak bir yığın bölgesi belirlenir. SP ise verinin koyulacağı adresi gösterir. Veri, yığına eklendikçe SP'nin değeri 1 azaltılır. Benzer şekilde yığından veri alındıkça da 1 arttırılır. Yani temel bilgisayar sistemimizde kullanılan yığın türü LIFO'dur (Last-In First-Out).

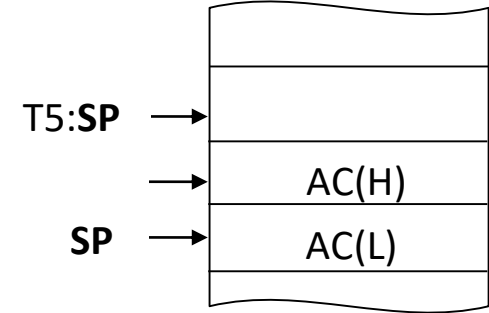
Yüksek seviyeli programların derleyicileri, yerel ve global değişkenleri tutmak için yığını kullanabilirler. Temel bilgisayar sistemimizde bu işleri yerine getirebilmek için PSH (yığına koyma) ve PUL (yığından alma) komutları vardır. Bir de matematiksel işlemlerde, işlem önceliğinin temini için kullanılmaktadır.

Özetle yığın, özellikle çok sayıda alt programa dallanma olanağı sunar.

Yığınla İlgili Komutlar

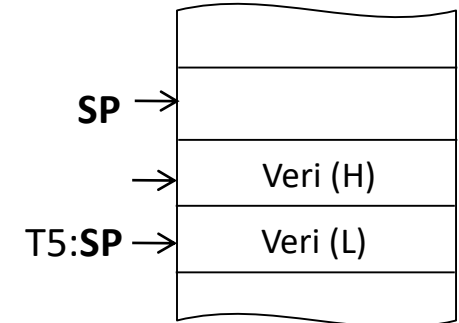
PSH (Push): Akümülatördeki veriyi, yığın göstericisinin gösterdiği yere kaydeder. Doğal adresleme moduna sahiptir. Akünün önce düşük daha sonra da yüksek anlamlı kısmı yığına konulur.

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC06.ADRMD0:	$AR \leftarrow SP$
T4. IDEC06.ADRMD0:	$M[AR] \leftarrow AC_L, SP \leftarrow SP-1, AR \leftarrow AR-1$
T5. IDEC06.ADRMD0:	$M[AR] \leftarrow AC_H, SP \leftarrow SP-1, SC \leftarrow 0$



PUL (Pop): Yığın göstericisinin gösterdiği yerdeki veriyi akümülatöre kopyalar. Doğal adresleme moduna sahiptir. Yığındaki değer, aküye direkt olarak yüklenemeyeceğinden dolayı önce DR'ye, daha sonra da aküye yüklenir.

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC07.ADRMD0:	$SP \leftarrow SP+1$
T4. IDEC07.ADRMD0:	$AR \leftarrow SP$
T5. IDEC07.ADRMD0:	$DR_H \leftarrow M[AR], SP \leftarrow SP+1, AR \leftarrow AR+1$
T6. IDEC07.ADRMD0:	$DR_L \leftarrow M[AR]$
T7. IDEC07.ADRMD0:	$AC \leftarrow DR, SC \leftarrow 0$



Yığınla İlgili Komutlar

BSR (Branch to SubRoutine): Şartsız olarak hesaplanan etkin adresteki alt programa, kısa dallanma sağlar. Göreceli adresleme modunu kullanır. PC'nin içeriğinin yığına atılmasını gerektirir.

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC15.ADRMD5:	$TR \leftarrow AR$ (ofsetin adresi kaydedildi)
T4. IDEC15.ADRMD5:	$AR \leftarrow SP$
T5. IDEC15.ADRMD5:	$M[AR] \leftarrow PC_L$, $AR \leftarrow AR-1$, $SP \leftarrow SP-1$
T6. IDEC15.ADRMD5:	$M[AR] \leftarrow PC_H$, $SP \leftarrow SP-1$
T7. IDEC15.ADRMD5:	$AR \leftarrow TR$ (ofsetin adresi geri yüklendi)
T8. IDEC15.ADRMD5:	$PC \leftarrow$ Etkin adres, $SC \leftarrow 0$

T3 ve T7 adımlarındaki mikroişlemlerin sebebi; etkin adres hesaplama ünitesinin ofset değerine ihtiyaç duymasıdır. AR, direkt olarak belleğe bağlantılı olduğundan ofset değeri, T7 saykılıyla birlikte belleğin çıkışında hazır olacaktır ve bu değer etkin adres hesaplama ünitesine gidecektir.

Yığınla İlgili Komutlar

JSR (Jump to SubRoutine): Şartsız olarak, alt programa uzun dallanma sağlar. Direkt ve indis adresleme modlarını kullanır.

Direkt adresleme modu için mikroişlem adımları;

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC24.ADRMD2:	$TR \leftarrow AR, PC \leftarrow PC+1$ (adres bilgisi kaydedildi)
T4. IDEC24.ADRMD2:	$AR \leftarrow SP$
T5. IDEC24.ADRMD2:	$M[AR] \leftarrow PC_L, AR \leftarrow AR-1, SP \leftarrow SP-1$
T6. IDEC24.ADRMD2:	$M[AR] \leftarrow PC_H, SP \leftarrow SP-1$
T7. IDEC24.ADRMD2:	$AR \leftarrow TR$ (adres bilgisinin başlangıcı geri yüklendi)
T8. IDEC24.ADRMD2:	$TR_H \leftarrow M[AR], AR \leftarrow AR+1$
T9. IDEC24.ADRMD2:	$TR_L \leftarrow M[AR]$
T10. IDEC24.ADRMD2:	$PC \leftarrow TR, SC \leftarrow 0$

İndis adresleme modunu kullanan JSR ile göreceli adresleme modunu kullanan BSR komutlarının mikro işlem adımları aynıdır. Farklılık, etkin adreslerin hesabındadır.

Yığınla İlgili Komutlar

RTS (ReTurn to Subroutine): Alt programdan, program akışının kaldığı yere döner. RTS komutu, PUL komutuna benzemektedir. Aralarındaki fark, yığından okunan bilginin AC yerine PC'ye konulmasıdır.

KOMUTUN MİKRO İŞLEM ADIMLARI	
T3. IDEC26.ADRMD0:	$SP \leftarrow SP+1$
T4. IDEC26.ADRMD0:	$AR \leftarrow SP$
T5. IDEC26.ADRMD0:	$PC_H \leftarrow M[AR], AR \leftarrow AR+1, SP \leftarrow SP+1$
T6. IDEC26.ADRMD0:	$PC_L \leftarrow M[AR], SC \leftarrow 0$