

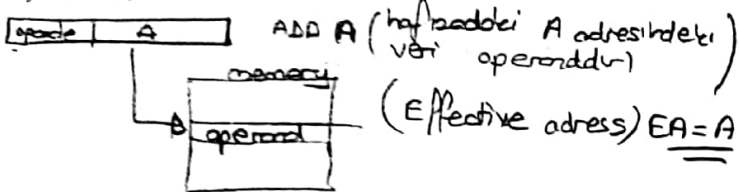
Adresleme Modları ve Formatları

ADD 3 * holder
*

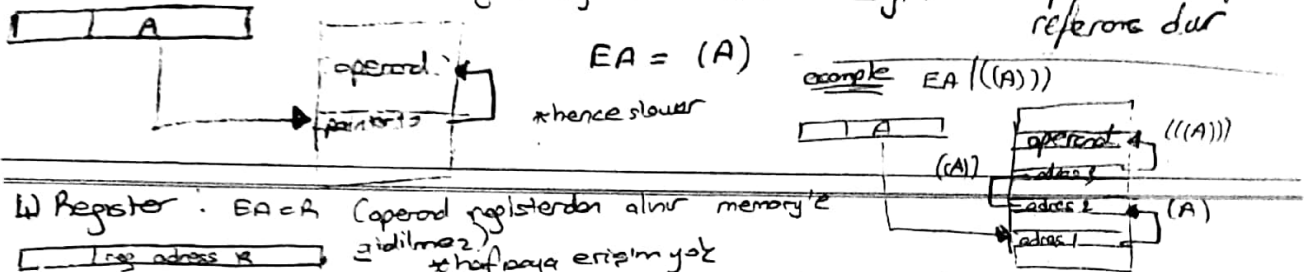
→ konat var ve operand belli, operand tespithne gerek yok

⇒ Veriyi alıp getirmek için hafıza refleksine gerek yok

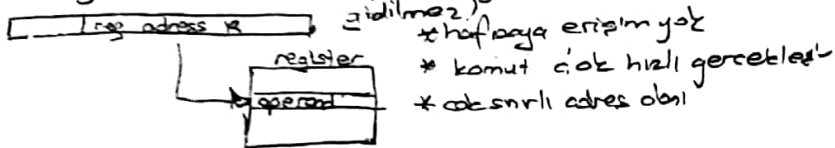
2) Direct



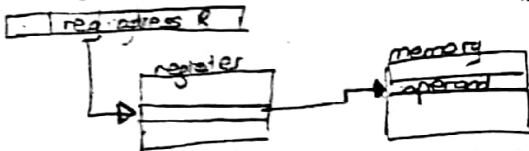
3) Indirect ADD (A) "A'ın geçirdiği adresleri; adrese girer" "ofıza bir den fazla referans olur"



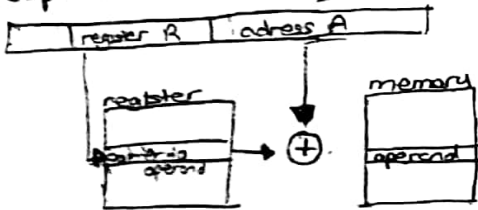
4) Register : EA = R (operand register dan alamat memory)



5) Register Indirect * EA = (R)
* hem register hem hafıza kullanılır
* Büyük adres dani vardır (2n)



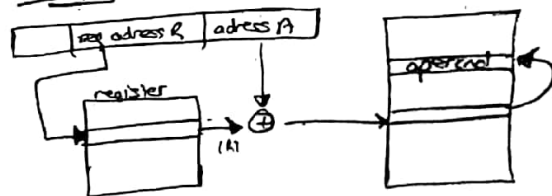
6) Displacement Addressing $EA = A + (LR)$



7) Relative $EA = A + (PC)$

8) Indexed $EA = A + R$

Example $EA = (A + (R))$

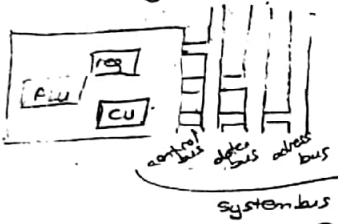


Bölüm 14 İşlemci Yapısı ve Fonksiyonları

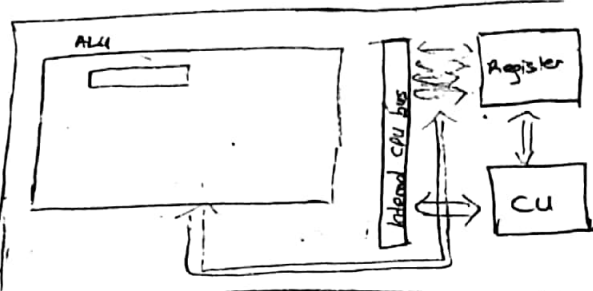
İşlemci Gereksinimleri

- ③ Fetch: İşlemci bir komutu hafızadan okur
- ③ Interpret Instruction: Komut hangi işlemi gerektirdiğini belirlemek için çalışır
- ③ Fetch data: Bir komutun yürütülmesi, bellekten veya I/O modülündeki verileri demanı gerektirebilir
- ③ Process data:
- ③ Write data:

System veri yolu ile birlikte CPU



CPU'nun iç yapısı



Saklayıcı Organizasyonu

- ③ İşlemci içerisinde ana bellekten bir üst düzey bellek ve hiyerarşide bir bellek olarak bir dizi saklayıcı (2 rolü vardır)
- ③ Saklayıcıların durumunu optimize ederek ana bellek kullanımını en aza indirmek için
- user-visible saklayıcılar
- ③ Makine veya assembly dili programını etkileşim

- Kontrol ve durum saklayıcıları
- ③ İşleminin çalışmasını kontrol etmekten CU tarafından kullanılır

- ③ Programların yürütülmesi için ayrıcalıklı işletim sist. programları tarafından kullanılır
- USER-VISIBLE SAKLAYICILAR → işletimin uyguladığı makine diline referans edilir

→

→

→

→

Önemli

KONTROL VE DURUM SAKLAYICILARI

Komutların yürütülmesi için 4 saklayıcı gerekli

PC (program counter) = alıp getirilecek komutun adresini içerir

IR (instruction reg) en son getirilen komutu içerir

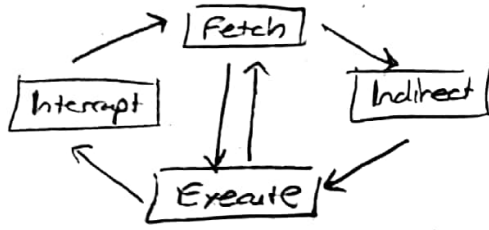
MAR = bellekteki bir konunun adresini içerir

MBR → Belleğe yazılmak üzere bir veri sözcüğü veya en son okunan sözcüğü içerir

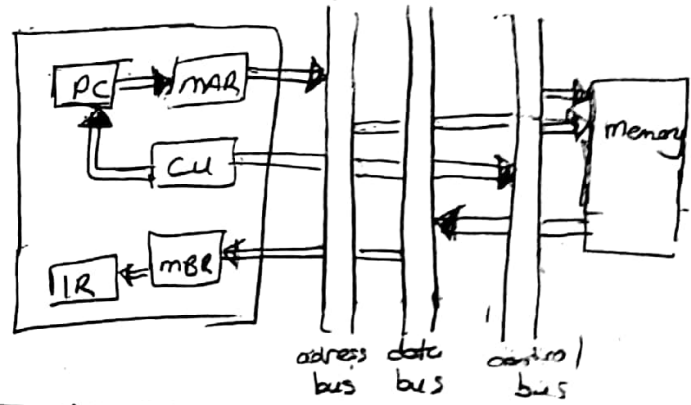
③ Program status word (PSW) → durum bilgisi içeren saklayıcı veya saklayıcı

③ Durum bayrakları → sign equal zero overflow

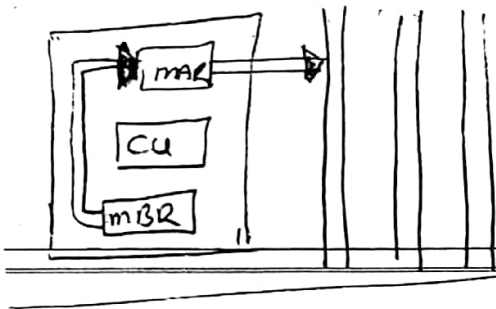
Komut Geçirimi



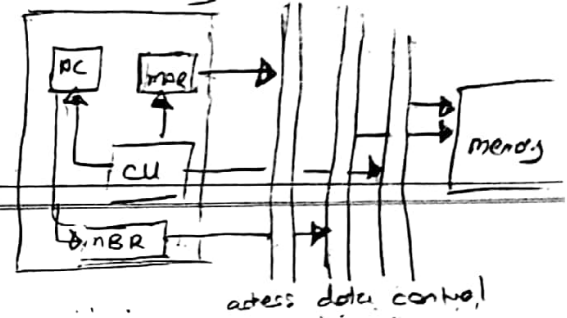
Veri-Alma, Al-Getir Geçirimi



Veri Akışı - Delayli Geçirimi



Veri Akışı - Kesme Geçirimi



Sıra Sıra Ek adımlar

- Fetch Instruction (FI) : bir sonraki beklenen talimatı bir arabelleğe koyun
- Decode Instruction : opcode ve operand belirteçlerini belirleme
- Calculate operand : her kaynak operandın fetch adresini hesaplayın
- Fetch operand : her operand bellekten getir (Kayıtlardaki operandların getirilme siye gerek yok)
- Execute instruction : belirtilen işlem yapılır ve hedef operand konumunda sonucu saklayın
- Write operand : sonucu bellekte saklayın

⊗ pipelining mantığıdır ⇒ senkron düzey işlemlerin gerçekleştirilmesi (artarda)