



7. HAFTA

BLM301

BİLGİSAYAR MİMARİSİ

Yrd. Doç. Dr. Salih GÖRGÜNOĞLU

sgorgunoglu@karabuk.edu.tr

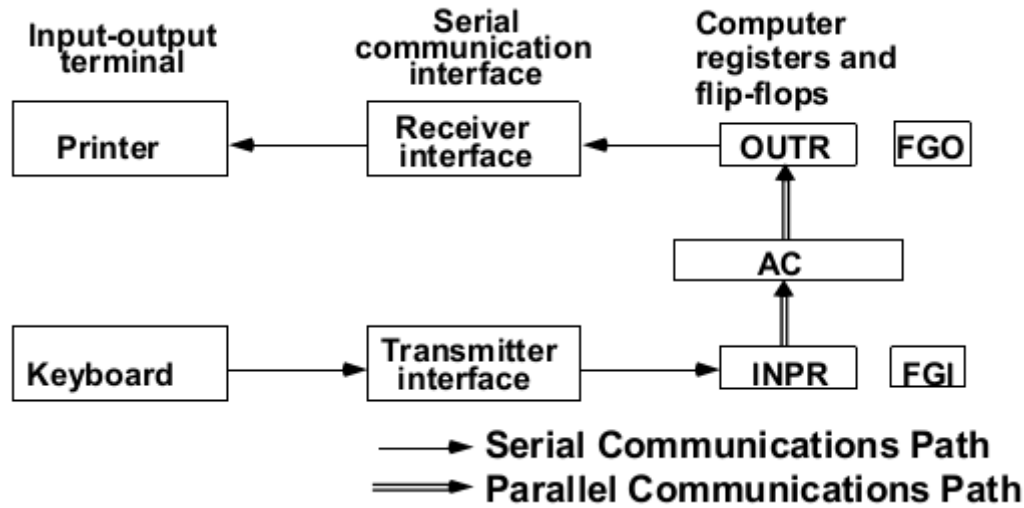
KBUZEM

Karabük Üniversitesi

Uzaktan Eğitim Uygulama ve Araştırma Merkezi

7. Giriş- Çıkış ve Kesmeler

Temel bilgisayar da giriş çıkış işlemleri için bağlantı şekli aşağıdaki şekilde gösterilmiştir. Tek çizgili oklar seri haberleşmenin yapıldığı yerleri, İki çizgili oklar ise paralel haberleşmenin yapıldığı yerleri göstermektedir.



Verilen şekilde kaydedicilerin anlamı şu şekildedir.

INPR: Giriş (input regidter) Kaydedicisi

OUTR: Çıkış (Output) Kaydedicisi

FGI: Giriş(İnput Flag) Bayrağı

FGO: Çıkış(Output) Bayrağı

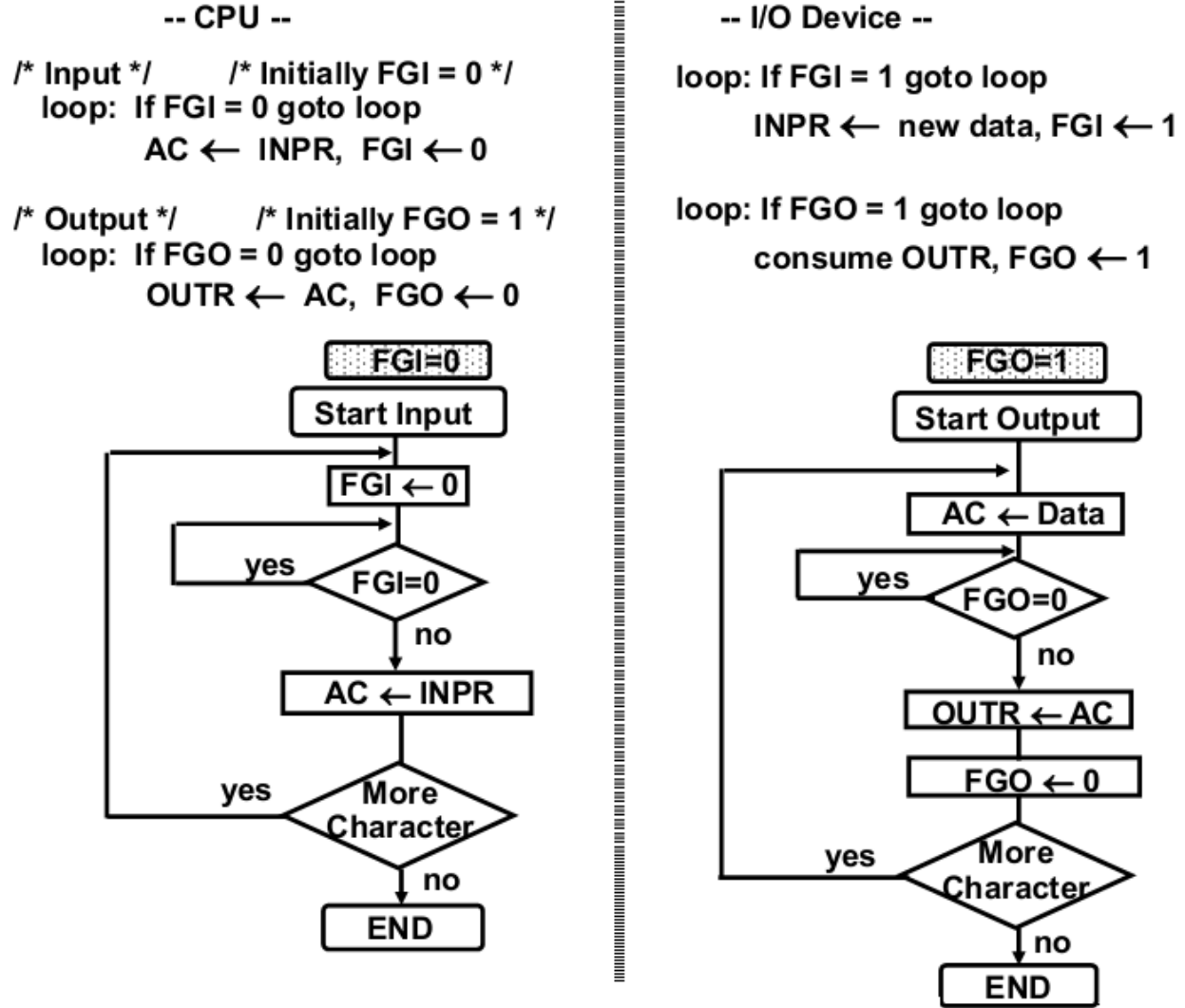
IEN: Kesme yetkilendirme biti(Interrupt enable)

Giriş çıkış işlemlerinde işel akışı şu şekilde gerçekleşmektedir.

- Terminal seri olarak veriyi gönderir veya alır
- klavyeden alınan seri veri INPR kaydedicisine kaydırılır
- Printere gönderilecek veri OUTR Kaydedicisinde tutulur.
- INPR ve OUTR kaydedicileri seri olarak terminalle haberleşir. AC ile paralel olarak haberleşir.
- Bayrak bitleri (FGI ve FGO) Giriş çıkış cihazı ile bilgisayar arasındaki senkronizasyonu sağlamak için kullanılır.

Program kontrollu (Yoklama metodu) giriş çıkış birimleri ile veri transferi

Bir giriş-Çıkış birimi ile temel bilgisayar arasındaki veri transferi iki açıdan düşünülmelidir. Birincisi CPU tarafında verinin alınması veya gönderilmesi, İkinci ise Giriş-Çıkış cihazı tarafından verinin alınması veya gönderilmesi. Bu şekilde veri transferi nin nasıl yapıldığı aşağıdaki şekilde görülmektedir.



Sol taraftaki şekilde CPU tarafından verinin okunması anlatılmaktadır. Eğer FGI biti 0 ise INPR kaydedicisinde veri hazır değildir. Başlangıçta FGI=0 yapılır. FGI sürekli kontrol edilir. Eğer FGI=0 ise veri hazır değildir. Kontrole devam edilir. FGI=1 olduğunda ise veri hazırdır ve INPR kaydedicisinden veri alınarak AC kaydedicisine aktarılır. Bir sonraki okuma işlemi için bu işlem tekrar edilir. Veri göndermek için başlangıçta FGO=1 yapılır FGO=0 olduğu müddetçe çıkış birimi

meşguldur. Veri gönderilmez. FGO kontrol edilmeye devam edilir. Nezaman ki FGO=1 oldu ozaman AC deki değer OUTF kaydedicisine gönderilir. FGO =0 yapılır ve bir sonraki karakter gönderimi için işlem tekrar edilir.

Giriş Çıkış komutlarına ait mikroişlemler aşağıda verilmiştir.

Fetch ve Decode aşamaları tüm komutlar için ortaktır.

T0: AR ← PC (S₀S₁S₂=010, T0=1)
T1: IR ← M [AR], PC ← PC + 1 (S₀S₁S₂=111, T1=1)
T2: D₀, . . . , D₇ ← Decode IR(12-14), AR ← IR(0-11), I ← IR(15)

İndirect aşaması

D₇IT₃: AR ← M[AR]

Giriş Çıkış komutlarının (çalıştır) aşamalarına ait mikroişlemler

D₇IT₃ = p
IR(i) = B_i, i = 6, ..., 11

| | | |
|------------|---|-----------------------------|
| | p: SC ← 0 | Clear SC |
| INP | pB₁₁: AC(0-7) ← INPR, FGI ← 0 | Input char. to AC |
| OUT | pB₁₀: OUTF ← AC(0-7), FGO ← 0 | Output char. from AC |
| SKI | pB₉: if(FGI = 1) then (PC ← PC + 1) | Skip on input flag |
| SKO | pB₈: if(FGO = 1) then (PC ← PC + 1) | Skip on output flag |
| ION | pB₇: IEN ← 1 | Interrupt enable on |
| IOF | pB₆: IEN ← 0 | Interrupt enable off |

Program kontrollu G/Ç

- Sürekli olarak işlemciyi meşgul edecektir.
- CPU işlem hız yavaşlayacaktır.
- Basittir, az donanım bileşeni gerektirir

Temel bilgisayar da giriş ve çıkış işlemleri aşağıdaki program ile gerçekleştirilebilir

Input

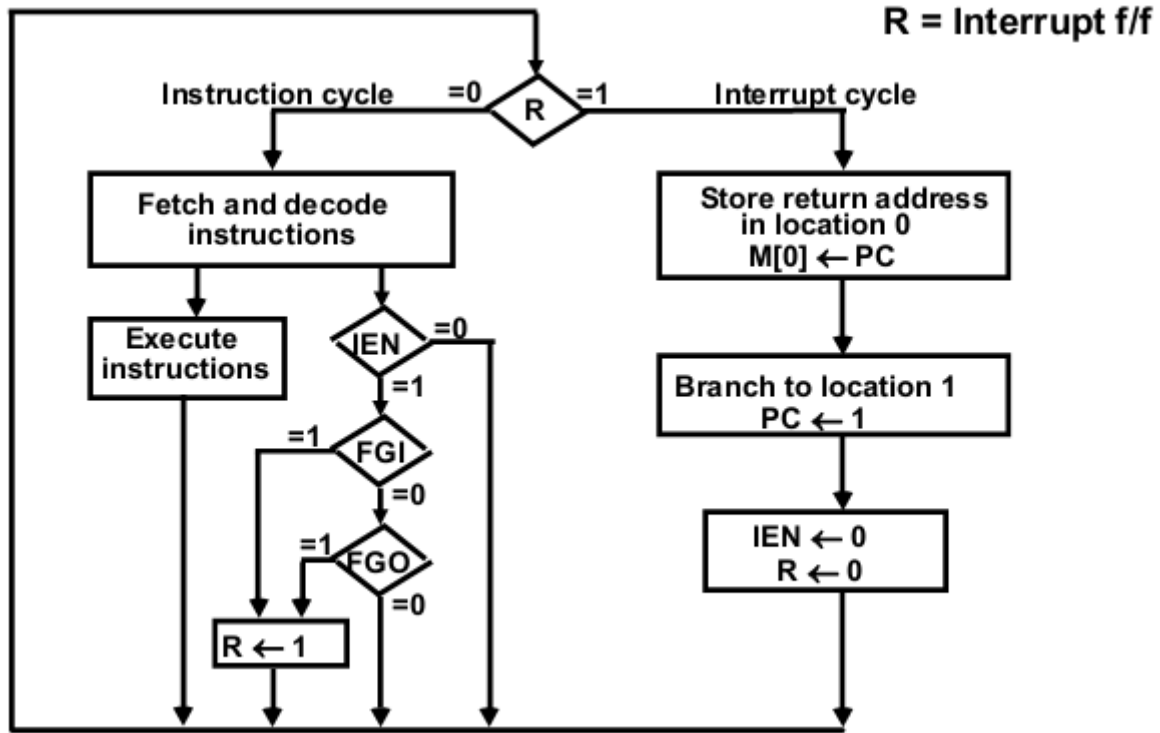
| | | |
|-------|-----|------|
| LOOP, | SKI | DEV |
| | BUN | LOOP |
| | INP | DEV |

Output

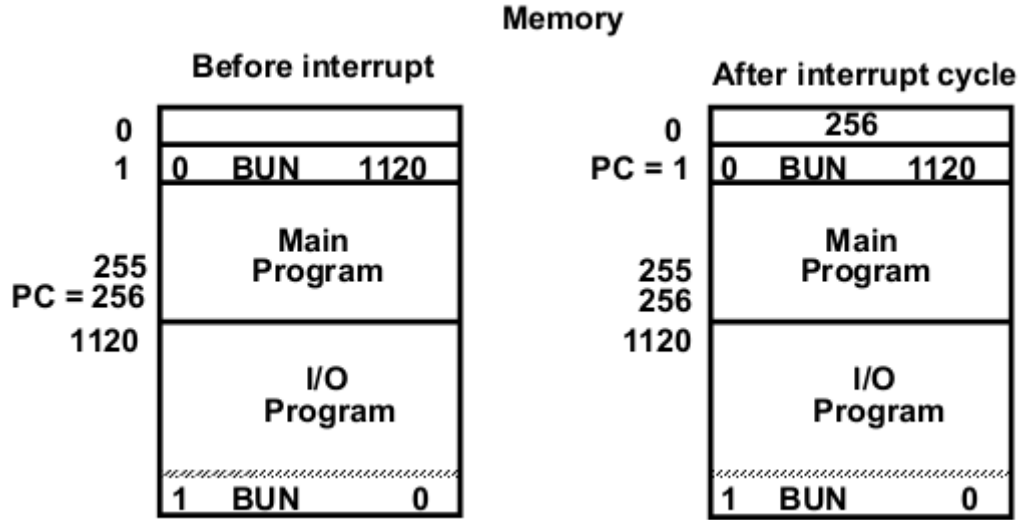
| | | |
|-------|-----|------|
| LOOP, | LDA | DATA |
| LOP, | SKO | DEV |
| | BUN | LOP |
| | OUT | DEV |

Kesmeli Giriş-Çıkış Kullanımı

İşlemciyi daha az meşgul etmek için bu yöntem kullanılır. Bu yöntemde giriş çıkış cihazından veri gelip gelmediği veya gönderilip gönderilmediği G/Ç arabirimi tarafından kontrol edilir. Örneğin G/Ç birimi bir veriyi bilgisayara(CPU tarafına) gönderdiğinde, Giriş arabirim devresi bunu algılar ve Kesme sinyali üreterek bunu işlemci tarafına, işlemci kesme girişine bir sinyal göndererek bildirir. İşlemi bu ana kadar kendi işlemlerinin yapmakla ilgilenir. İşlemci kesme sinyali geldiğini kesme girişlerinden anlar. Kesme sinyali geldiğini işlemci R filip flopunun 1 olmasından anlar. R=1 ise o an yapmakta olduğu komutun çalışması biter bitmez kesme alt programına (0 adresine)dallanır. Ve kesme alt programındaki (1 adresinden itibaren) komutları sırası ile çalıştırır. Buradaki komutlar biter bitmez BUN 0 I Komtu ile (BUN indirect 0 adresi anlamındadır) ana programdan kaldığı yerden devam eder. Temel bilgisayar da bir kesme geldiğinde işlemci otomatik olarak 0 adresine dallanır. 0 adresi bu yüzden kesme vektör adresidir. (Keme vektör adresi kesme geldiğinde dallanılan adrestir). Kesme alt programı 1 adresinden itibaren yazılır. İşlemcinin kesmeyi kabul etmesi için Kesme yetkilendirme (interrupt enable) biti IEN=1 yapılmalıdır. Eğer IEN=0 ise kesme kabul edilmez. Kesme işleminin le alınış biçimi aşağıda verilen akış çizelgesinde görülmektedir.



Aşağıda kesme kullanarak bir giriş çıkış ileminin nasıl yapıldığı görülmektedir. /Ç biriminden bir kesme kelmeden önce 0 adresi boştur ve 1 adresinde BUN 1120 yazılıdır. İndirect biti I=0 dir. PC=256 yer almaktadır. Ana program(main program) 256 adresinde yer alan komutu çalıştırmaktadır. Bu esnada G/Ç biriminde kesme gelmiş ve R =1 olmuştur. Donanımsal olarak bu komutun çalışması biter bitmez 0 adresine gidilmiş buraya ana programa dönüş adresi 256 atılmıştır. Daha sonra PC=1 yapılarak 1 adresindeki komut çalıştırılmıştır. 1 adresinde BUN 1120 komutu vardır ve I=0 dır. BUN 1120 komutu çalışınca PC=1120 olmaktadır. Bunu anlamı 1120 adresindeki komutu çalıştır demektir. 1120 adresinden itibaren G/Ç işlemlerini yapan altprogram yer almaktadır. Buradak komutların çalışması sırası ile yapıldıktan sonra ensonda yer alan BUN 0 I komutu çalıştırılmaktadır. = adresşne indirect olatrak dallan demektir. Bu komut çalışınca 0 adresinin içindeki adrese(256 adresine) dallanma gerçekleştirilir. 0 adresinin içinde yer alan 256 adres değerinin ana program dönüş adresi olduğuna dikkat ediniz. Böylece ana programa kaldığı yerden devam edilir.



Kesme geldiğinde G/Ç (I/O) işlemlerinin yapılması ile ilgili mikroişlemler aşağıda yer almaktadır.

- $R \leftarrow 1$ if $IEN (FGI + FGO)T_0'T_1'T_2'$
 $\Leftrightarrow T_0'T_1'T_2' (IEN)(FGI + FGO): R \leftarrow 1$

Fetch ve decode aşamaları kesme dikkate alınarak aşağıdaki şekilde olmalıdır. Bura R' keme yoksa, kesme sinyali gelmemişse anlamındadır. $R'T_0$ kontrol sinyalinin anlamı Kesme yoksa ve T_0 zamanında ise dir. Diğerleri de bu şekilde düşünülmelidir.

| | | |
|--------|----------------------------------|--|
| Fetch | $R'T_0:$ $R'T_1:$ $R'T_2:$ | $AR \leftarrow PC$ $IR \leftarrow M[AR], PC \leftarrow PC + 1$ $D_0, \dots, D_7 \leftarrow \text{Decode } IR(12 \sim 14),$ $AR \leftarrow IR(0 \sim 11), I \leftarrow IR(15)$ |
| Decode | | |

Kesme gelmiş ise bu taktirde T_0, T_1, T_2 zamanlarında kesme sinyalinin geldiği düşünülerek fetch ve decode aşağıdaki gibi olur. Indirect aşaması değişmez.

| | | |
|--------------------|---|---|
| Indirect Interrupt | $D_7'IT_3:$ $T_0'T_1'T_2'(IEN)(FGI + FGO):$ $RT_0:$ $RT_1:$ $RT_2:$ | $AR \leftarrow M[AR]$ $R \leftarrow 1$ $AR \leftarrow 0, TR \leftarrow PC$ $M[AR] \leftarrow TR, PC \leftarrow 0$ $PC \leftarrow PC + 1, IEN \leftarrow 0, R \leftarrow 0, SC \leftarrow 0$ |
|--------------------|---|---|

Burada şu sorulara cevap verilmesi gerekmektedir.

- CPU cihazların kesme isteğini nasıl tanımaktadır.

- Birden fazla farklı cihaz kesme isteğinde bulunursa, hangi cihaz için hangi alt program(kesme hizmet programı – interrupt service routine –ISR) çalıştırılacaktır.
- Bir kesme çalışırken, bir başka kesme sinyali gelirse ne olacaktır.
- Aynı anda birden fazla birim kesme üretirse ne olacaktır.