

BİLGİSAYARDA KESMELER VE KESME YÖNTEMİ

Doç. Dr. Fazıl DEMİRCİ*

Arş. Gör. Cengiz GÖK**

1- GİRİŞ

Kesme (interrupt) işlemi, bilgisayarın merkezi işlem biriminin (CPU) hiç hesapta olmayan bir olayın etkisiyle, normal olarak yapmakta olduğu işi bırakarak, kesmenin gösterdiği olaya geçici olarak yönelmesi (CPU'nun gösterilen yeni işi yapması) için kullanılır.

Kesmeler (Interrupts), CPU'ya içerden veya dışardan olmak üzere çok çeşitli yerlerden gelebilir. CPU, bilgisayarı oluşturan çevre birimleriyle haberleşmek ve çalışmalarını uyumlu bir şekilde sağlamak için kesmeleri kullanır. Kesmeler, CPU'nun çevre birimleriyle haberleşmede kullanılmasının yanısıra, CPU'nun aktif ve etkili bir şekilde kullanılmasını sağlar. Mesela, meydana gelen her hangi bir girdi/çıkı (I/O) olayı, bu girdi/çıkı işini yapacak birimle (printer, keyboard vs.) CPU arasında , kesme kullanılarak düzenli, etken ve yüksek performansta yapılması sağlanır. Şöyle ki: girdi /çıkı (I/O) işlemi sırasında, işlemin düzenli bir şekilde olması için, CPU'nun sürekli I/O birimini kontrol etmesi gerekir. Bu da I/O işleminin sonuna kadar başka işle ilgilenememesi demektir. Bu yüzden de, CPU etkin ve verimli bir şekilde kullanılamamış olur. G/Ç işleminin olmadığı zamanlarda, CPU ile G/Ç birimleri arasında kullanılan kesmeler sayesinde, CPU'ya girdi/çıkı işleminin olmadığı haber verilerek başka işlere

* Gazi Üniversitesi Endüstriyel Sanatlar Eğitimi Fakültesi, Bilgisayar Eğitimi Bölüm Başkanı

**Gazi Üniversitesi Endüstriyel Sanatlar Eğitimi Fakültesi, Bilgisayar Eğitimi Bölümü, Araştırma Görevlisi Bilgisayar Müh.

yönelmesi sağlar. Böylece CPU etkin ve verimli bir şekilde kullanılmış olur (Print işi olurken editörde yazı yazılması veya CPU'nun matematiksel hesaplamalar yapması gibi). CPU'nun çevre birimlerinde meydana gelen hatalardan haberi olması ve ona göre işlem yapması, birtakım tedbirleri alması da yine kesmeler sayesinde olur.

2. KESMELERİN OLUŞMASI VE KULLANILMASI

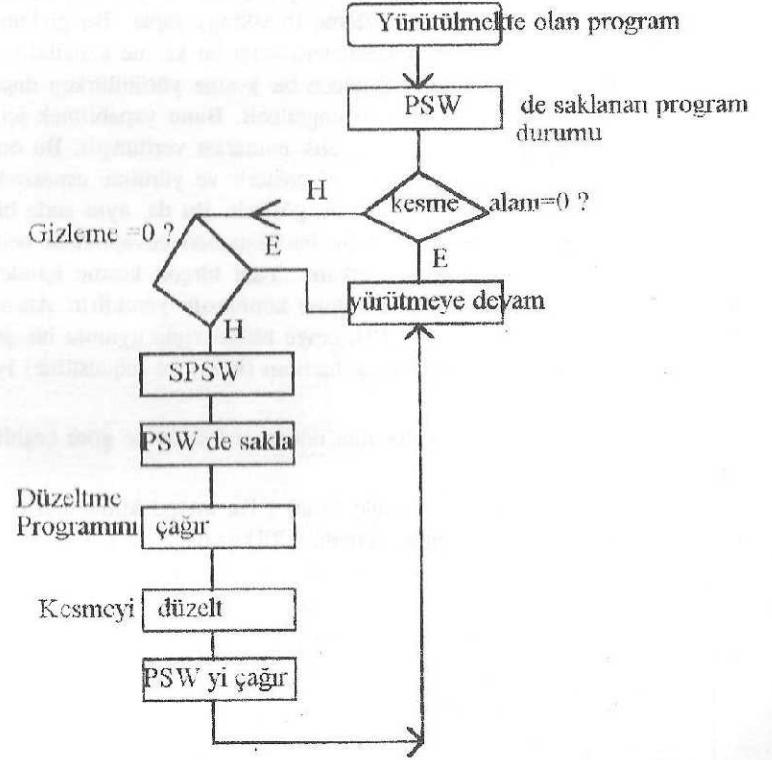
Kesmeler, CPU'yu ilgilendiren her türlü olayda kullanıldığı gibi, çeşitli birimler arasında koordinasyonun sağlanmasında da etkin bir şekilde kullanılır. Kesmeler, herhangi bir I/O olayının başlaması veya herhangi bir HW veya SW hatası durumuna CPU'yu uarmakta kullanılır. Bu tür olaylarda kesme sinyallerinin CPU tarafından kendi içerisinden üretilmesiyle meydana gelen dahili kesmelere "trap" denir. Bu tür kesmeler genelde programlama hatalarından meydana gelir. Mesela, bir matematik işleminde 'sıfıra bölme hatası (divide by zero)' oluşması, veya kullanılan programlama dilinde geçerli olmayan bir komutun kullanılması gibi.

Kesme işleminde kullanılan en temel metod : Kesmenin geldiği birim ile CPU'yu birbirine bağlayan, kesme istek hattı (Interrupt Request Line) denilen kontrol hattının aktif hale getirilmesidir. Birim tarafından gönderilen kesme sinyalleri, CPU tarafından sürekli test edilen kesme yazmacında (Interrupt Register) saklanır. Kesme yazmacının test edilmesi genelde her komut çevirimi (instruction cycle) sonunda yapılır. Kontroller sonunda herhangi bir kesmenin olduğu görülürse, CPU tarafından özel bir kesme servisi programının bulunduğu adres bulunur, daha sonra o adresteki kesme servisi programı yürütülür. Kısaca kesme işlemini şu şekilde özetleyebiliriz :

- 1- CPU'nun kesme kaynağını tanıması.
- 2- CPU'nun kesme servisi programının bellekteki adresini saptaması.
- 3- Program sayacı (PC) ve diğer CPU durumunu gösteren bilgilerin (kesilen programa ait bilgiler) yığıtta (stack) saklanması.
- 4- PC'nin, kesme servisi programının adresiyle yüklenip, CPU kontrolünün 'kesme servisi programına' geçmesi, kesme servisi programını çalıştırdıktan sonra, 'kesme servisinden dönüş (Return from Interrupt)' ko-

mutunun çalışmasıyla CPU kontrolünün tekrar eski yerine (kesilmiş programa) dönmesi.

Basamaklar şeklinde işleyişi verilen kesme işlemi belli bir düzen içinde işler. Kesmenin geldiği anda, CPU yürüttüğü programın (i) mikro-komutunu (microinstruction) çalıştırıyorsa, önce bu komutun çalışmasını bitirir. PC'nin değeri ((i+1). adres), işlemci durumu (processor status) ve diğer bilgiler kesme yöneticisi tarafından otomatik olarak yığıtta (stack) korunur. Şekil-1'de bir kesmenin çalışma mekanizması gösterilmiştir.



Şekil-1 Kesme işlemini çalışma mekanizması

CPU çevre birimlerinden gelen kesmeleri aldığını ve işleme koyduğunu bildirmek için, kesme gönderen birimi özel bir kontrol sinyali gön-

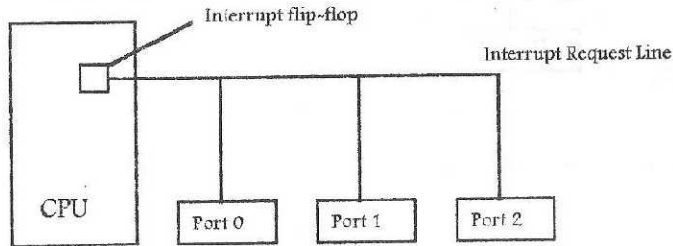
derir. Buna kesme doğrulayıcısı (Interrupt Acknowledge) denir. Bu özel sinyal sayesinde CPU ile çevre birimi arasındaki iletişim sağlanmış olur ve çevre birimi gönderdiği kesmelerin CPU'ya ulaştığını anlamış olur.

3. ÇOKLU KESMELER VE KESMELERİN SEÇİLMESİ

Kesme olayı sırasında CPU tarafından kullanılan komutlar genelde CPU'nun komut kümesinde bulunur. Bu komut kümeleri yürütülürken (kesme işlemi konulurken) başka bir kesme tarafından bir daha kesilmesi gerekir. Bunun için CPU, başka kesme kanallarını (interrupt request line) kapatıp, pasif (disable) hale getirir. Veya ikinci bir yol ise; kesme isteklerine (interrupt request) karşı, gizleme (masking) yapar. Bu gizleme sayesinde diğer taraftan gelebilecek kesmelere karşı bu kesme kanallarının bağlantılarının koparılmasını sağlar. Böylece bir kesme yürütülürken dışarıdan başka bir tanesinin araya girmesi engellenir. Bunu yapabilmek için de her kesmeye kendine has belli bir öncelik numarası verilmiştir. Bu öncelik numaraları kesmenin önem derecesini gösterir ve yürütme esnasında ne zaman, hangi derecede önemli olduğunu gösterir. Bu da, aynı anda bir çok kesme etkisi altında kalan CPU'nun, bu kesmeleri cevaplarırken belli bir metodu olması gerektiğini ortaya çıkarır. Yani birçok kesme içinden önemli olan seçilip, öncelikle onların işleme konmasını gerektirir. Ancak böyle bir metodlu seçim sayesinde CPU, çevre birimleriyle uyumlu bir şekilde çalışmış olur. Bu da, kesme istek hattının (interrupt requestline) iyi yönetilmesiyle gerçekleşir.

Kesmelerin seçim işlemi, kullanılan donanım özelliğine göre çeşitlilik gösterir :

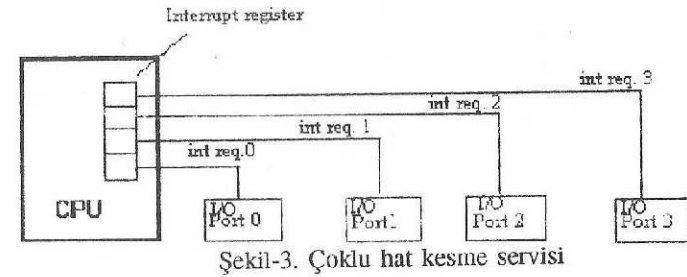
1) Tekli Hat Kullanılması (Single Line) : Bu metod kullanılan metodlardan donanım yönünden en ucuz olanıdır. CPU'ya bağlı bir çok I/O portu, bir tek istek hattını ortak olarak kullanır. Bütün sinyal alış-verişleri bu hat üzerinden yapılır. Şekil-2 de tek hat kullanımının yapısı gösterilmiştir.



Şekil-2. Kesmelerin iletilmesinde tekli hatın kullanılması

Her hangi bir kesme olayının işleme konması sırasında, CPU kendisine bağlı bütün I/O birimlerini teker teker tarar, bu işlem sonucunda kesmenin kaynağını bulur. Kesme kaynağının tespit edilmiş olması, "interrupt acknowledge" hattının aktif edilmesiyle kesme gönderen birime bildirilir. Bütün çevre birimleri sadece bir tek hat üzerinden kesme sinyallerini gönderip alma işlemini yapar. Bütün işlemlerin bir tek hat üzerinden olması hat yöntemini zorlaştırır.

2) Çoklu hattın (Multiple Line) kullanılması : En çok kullanılan bu metod, her çevre biriminin kendine ait özel bağımsız hatlarla CPU'ya bağlanmasıyla oluşur. Her hattın kendine ait bir öncelik sırası vardır. CPU kesme kaynağını bu ayrı özel hatlar sayesinde hemen bulur, bütün birimleri teker teker taramasına gerek yoktur. CPU'nun, diğer birimleri taramadan, kesme kaynağından kesme servisi programının adresini alması gerekir. Bunun için ikinci bir işleme ihtiyaç duyulur. Fakat, kesme servisi programlarının adreslerinin bir vektör yapısında tutulmasıyla bu sorun da ortadan kaldırılmıştır.



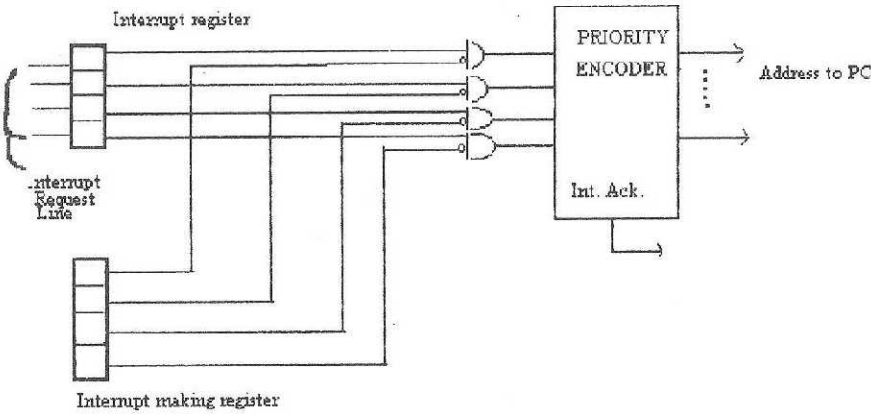
Şekil-3. Çoklu hat kesme servisi

4- INTERRUPT VEKTÖRÜ VE KULLANILMASI

Daha önceki bölümde kesme servisi programının çalıştırılması için servisin bulunduğu adresin getirilmesi işleminin zorluğundan bahsetmiş ve bunun interrupt vektör yapısı kullanılarak çözümleneceğini söylemiştik. Vektör yapısı, belli bir birimden gelen kesme sinyallerine en hızlı ve en uygun bir şekilde cevaplanmak için donanım olarak uyarlanmış bir sistemdir. Vektör yapısı içinde tutulan interrupt handling programının adresi, kesme servisi programının adresi bu uygun donanım yapısı sayesinde kolayca bulunur ve işleme konur. CPU'ya kesme gönderen birim, kesme ser-

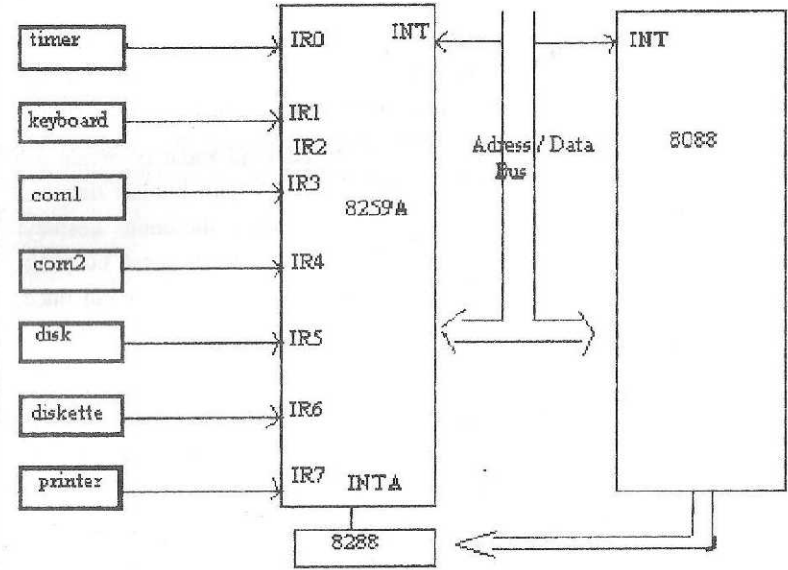
visi programının başlama adresini veya bunu bulmada kullanılacak olan transfer vektörünü CPU'ya bildirir. Böylece CPU'nun servis programını çalıştırıp, kesmeye cevap vermesi daha kolay ve çabuk olur.

Şekil-4'te çoklu istek (multilevel request) hatından transfer vektör yapısının eldesi gönderilmiştir. Çoklu istek hattını oluşturan her bir hat, kendine özgü sabit adresin eldesinde kullanılır. Bu adres PC'ye atanarak kesme servisinin programının adresi PC'de tutulmuş olur. Birimden gelen kesme istek sinyalleri de CPU'nun kesme yazmacında tutulur. Kesme gizleme yazmacı da, kesme istek hatlarının hepsinin veya bir bölümünün gerekli antarda-pasif (disable) yapılmasında kullanılır. Bunu şu şekilde yapar: Yazmacın (i) bitinin 1 yapılmasıyla (i) kesme hatı pasif (disable); 0 yapılmasıyla da aktif (enable) yapılmış olur. Bunu bir örnekle açıklayalım: Bir bilgisayarın 4 portunu kullanarak 3 tane I/O birimi takıldığını düşünelim, ve i. porttan bir kesme sinyali gelmiş olsun. Dört portun adreslemesinde kullanılan 2 bitlik öncelik (priority) belirleyici encoder sayesinde i. portun adresi sayaca (PC) yüklenir. Böylece kesme gönderen birim ve servis programının adresi öğrenilmiş olur.



Şekil-4 Kesme gizleme yazmacının adres belirlemede kullanılması.

Priority encoder ile kesmelerin önceliğinin ayarlanmasında (PC : 8086/8088 gibi CPU'larda) 8259A işlemci kullanılır. 8259A, birimlerle CPU arasında kesme yöntemi işlemini yapar. Bu yöntemin sırasında 8288 hat kontrolcüsü (bus controller) kullanılır. Şekil-5'te bu işlemciler kullanılarak kesme yönetiminin sağlanması gösterilmiştir.



Şekil-5 Priority encoder 8259A ve hat kontrolcüsü 8288'in kullanılması

5. KESMELERİN AKTİF VE PASİF HALE GETİRİLMESİ

CPU, çevre birimlerinden gelen kesmelere her zaman anında yanıt vermek zorunda değildir. Bazen çevreden gelen bu kesmeleri önemsemeyebilir. Mesela, bir yazdırma olayında yazıcıdan gelen kesme sinyalleri, ancak CPU'da yazdırılacak bir yazı varsa göz önünde tutulur. Eğer yazdırılacak yazı kalmamışsa, yazıcıdan gelen "hazırım" sinyalinin her hangi bir anlamı kalmayacaktır.

CPU, çok önemli bir iş yaparken, çevreden gelen kesmelerle rahatsız edilmesi istenmez. Bu nedenle kesmelerin aktif ve pasif hale getirilmesi gerekir. Bunu yapmanın en basit yolu, "Interrupt enable" ve "Interrupt disable" komutlarının kullanılmasıdır.

Kesme sinyalinin işleyişi biraz daha ayrıntılı incelemeye alınacak olunursa; çevre biriminden gelen kesme sinyali, kesme işleminin "interrupt service routine" tarafından cevaplanana kadar aktif kalır. CPU tarafından yanıt verilir verilmez başka gelebilecek kesmelere karşı önlem alın-

ması gerekir. Bir birimden gelen kesme işlemi cevaplandıktan sonra "pasif" hale getirilmezse, CPU daha sonra aynı yerden gelecek kesmeye bir kez daha yanıt vermek isteyecek ve CPU sonsuz bir döngüye girmiş olacaktır. Bu problemi ortadan kaldırmak için 3 ayrı mekanizma kullanılabilir:

1- Kesme servisi programın ilk komutu çalışana kadar çevreden gelecek kesmelerin "disable:Pasif" edilmesi, veya CPU'nun bunları önemsememesi gerekir. Bu durum ise kesme servisinin ilk komutunun "kesmeyi pasif yap" olmasıyla sağlanır. Kesme programı çalıştıktan sonra bu programın son komutu olan "return from interrupt : kesmeden çıkış"tan önce, kesmeyi yeniden aktif hale getiren "interrupt enable" komutunu kullanmaktadır.

2- CPU'nun otomatik olarak kesme servisi programını çalıştırmadan önce kesmeleri pasif hale getirmesidir. Bu durum ise : Kesme geldiği anda CPU, PC (program counter), PS'u (processor status) istif halde tutarak interrupt disable edilir. Bu da PS yazmacında kesmenin pasif veya aktif olduğunu gösteren bitin kullanılmasıyla olur. CPU kesme rutininden önce bu biti set eder ve 1 yapar. Program çalışıp bittikten sonra, çıkışında "kesmeden çıkış" komutunu çalıştırır ve PS yazmacındaki bit tekrar sıfırlanarak interrupt enable olur.

3- Donanım tarafından ortaya çıkartılabilen çözüm yoludur. Bu çözüm yolunda özel devreler kurularak sadece belli bir yerden veya belirli bir konumdaki kesmeleri yanıtlar.

6. KESME ÇEŞİTLERİ

Kesmelerin sınıflandırılması, gözönünde bulundurulacak kriterlere göre çeşitlilik gösterir : Temelde kesme gelme kaynağına göre 2'ye ayrılır : 1- Hardware kesmeleri, 2- Software kesmeleri. Fakat kesmelerin gözlenme durumları ve önemlilik dereceleri gözönüne alınırsa 3 gruba ayrılırlar : 1- Gizlenebilir kesmeler (Maskable interrupts), 2- Gizlenemez kesmeler (Nonmaskable interrupts), 3- Hiç bir şekilde gizlenemeyen, her zaman önceliği olan kesmeler (elektrik kesmelerine karşı kontrol kesmeleri vs.). Daha ayrıntılı bir sınıflandırma yapacak olursak, aşağıdaki gibi bir sınıflandırma daha uygun olur.

6.1. Program Kesmesi (Program Interrupt)

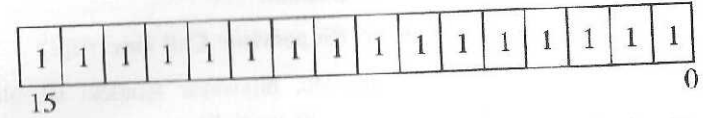
Program kesmesi aşağıdaki sebeplerden herhangi birisinden dolayı ortaya çıkabilir.

a) Geçersiz CPU komutu (Invalid CPU instruction): İşlemci komutlarındaki hatadan ortaya çıkan bir kesme olayıdır. Bu hata yazımdan (dilden) olduğu kadar Syntax (yazılım, söz dizimi) veya verilmesi gereken şifre hatasından da ortaya çıkar.

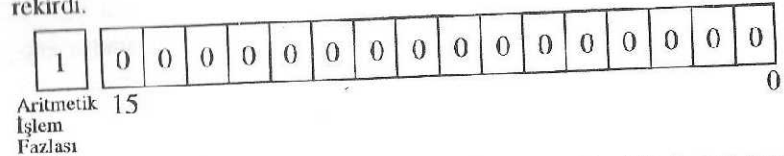
b) Aritmetik işlem fazlasından meydana gelen hata (Overflow Error) : Bilgisayardaki sayısal işlemler (Binary Codes) denilen ikili sayı koduyla temsil edilirler. Bu şekilde temsil edilirken, sayının uzunluğu bilgisayara özgü kelime uzunluğunu geçmemelidir. Eğer bilgisayardaki kelime uzunluğunu geçiyorsa, bu durumda aritmetik işlem fazlası olduğu görülür. Örneğin ;

Bilgisayarda sayıları temsil eden kelime uzunluğu 16 bitten oluşuyor ise, burada tutulabilecek en büyük pozitif sayının uzunluğu

$2.15-1 = 32767$ olur. Bu şekilde açıklanacak olursa ;



Burada + ve - işaretlerine ait olan işaret hücresi ihmal edilmiştir. Eğer 32768 sayısı temsil edilmek isteniyorsa aşağıdaki şekilde olması gerekirdi.



Böyle bir aritmetik işlem fazlası olayı gerçekleştiğinde ani kesme olayı ortaya çıkar. Bundan dolayı program kendiliğinden durur.

c) Korumayı Delme Hatası (Storage Protection Violation): Bu hata, programın herhangi komutunda veya yardımcı bellek üzerinde yazma izni olmayan bir yere (yazılmaması gereken bir yere) ilgi saklamak istenmesi durumunda ortaya çıkar. Bu tür hatalar hem yazma hemde okuma

durumunda ortaya çıkabilir. Eğer CPU, okumanın mümkün olmadığı bir adresten bilgi okumak isterse yine aynı hata görülür.

6.2. Giriş/Çıkış Kesmeleri (I/O Interrupt)

Bu tip kesmeler giriş/çıkış kanallarında ortaya çıkarlar. Bunun nedeni ise aşağıdakilerden herhangi birisi olabilir :

a) Giriş çıkış komutundan ortaya çıkan hatalar sonucu : Örneğin : Kullanılan dilde girdi veya çıktı komutu olarak geçerli olmayan bir komutun kullanılması; ikili sayı şifresinde meydana gelen hatadan; Girdi/Çıktı ve kanal numaralarının yanlış veya tespit edilememesinden ortaya çıkan hatalar sonucunda bu tür kesmeler gelebilir.

b) Girdi/Çıktı programının yürütülmesinin bitmesiyle ortaya çıkan kesmeler : Kanal programında kanal komut kelimesinin son deyimini yürütüldükten sonra işletim sistemleri kanal programını durdurmak için (H I/O) ile emir verir ve ani kesme olur. Bilgisayarın diğer kısımları çalışmasına devam eder.

c) Girdi/Çıktı ünitelerinin verileri geçiş yolundan taşınması sonucunda ani kesme olayı sonucunda ortaya çıkabilir.

6.3. İşletim Sistemi Kesmeleri (Supervisor Call Interrupt)

Kullanıcının hiçbir ilişkisi olmadan, bilgisayar içindeki işlemlerin yürütülmesi için işletim sistemi kesme emri verebilir.

6.4. Dıştan Kesme (External Interrupt)

Aşağıdaki iki sebepten dolayı ortaya çıkabilir.

a) Kullanıcının zorunlu olarak çalışmakta olan bir programdan çıkmak için yapılan kesme olayıdır. Örneğin : kısır döngü durumlarının meydana gelmesi durumlarında; kullanıcının makinayı kiraladığı sürenin dolması durumlarında meydana gelebilir. Elde olmayan nedenlerden dolayı makinenin durdurulmasının gerektiği durumlarda yapılan kesmelerdir.

b) Zamanlayıcılardaki bozukluktan dolayı ortaya çıkan kesmelerdir. Zaman paylaşımına bağlı çalışmalarda veya zamanlayıcının (+) değerinin (-) değere dönüşmesi durumunda ortaya çıkan kesmelerdir. Bu kesmeler bilgisayar aksamının birbirleriyle bağlanması sırasında ortaya çıkabilir. Mesela, işlemci işlemciyle veya ana belleğin kanallara bağlanması durumlarında ortaya çıkabilir.

6.5. Makinenin Test Kesmesi (Machine Check Interrupt)

Bilgisayarın ilk açıldığında, işletim sistemi bilgisayara bağlı olan üniteleri kendiliğinden test eder. Bu test yapıp doğruluğu kontrol edilmeden bilgisayar çevre birimlerini tanıyamaz ve işlevini yerine getiremez.

7. SONUÇ

Kesmelerin bilgisayar içinde bu şekilde kullanılması, genelde sisteme zarar verici, sistemin normal çalışmasını engelleyici bir etki olarak düşünülür. Gerçekte bu böyle değildir. Her ne kadar normal bir şekilde yürütmekte olan bir işi engelleyici olarak görülse de, aslında kesme bilgisayar ortamında işleri düzenleyen, sıraya koyan bir mekanizmadır. Bu mekanizmanın düzgün bir şekilde çalışması, kesme gönderen çevre birimlerinin yoğunluğuna ve kesme gönderme sıklığına bağlıdır. Çok sayıda çevre birimi olan ve hepsi de asenkron olarak (rastgele) kesme gönderen bir sistemde, sistem bütün çevre birimlerine kararlı bir şekilde cevap veremez, mutlaka ister istemez birkaç istek cevapsız kalacaktır. Ama kesmelerin CPU ve kendi aralarındaki asenkron haberleşmede yüklendiği görev göz önüne alınacak olursa, bilgisayarda iletişim konusunda kesme tekniğinin önemi bir kez daha ortaya çıkar. Bazı kesmelerin olmadığı durumlarda CPU çevre birimleriyle haberleşemeyecektir ve görevini tam olarak yerine getiremeyecektir. Eğer bütün kesmeler pasif duruma gelirse (örneğin bir kesme servisi programı içinde gelebilecek bütün kesmeler pasif duruma getirildi ve kesme servisi programından çıkarken tekrar aktif hale getirilmedi ise) işletim sistemi fonksiyonunu yerine getiremeyecek ve ortada sistem diye birşey kalmayacaktır.

Kesme yönteminin iyi yapılması halinde, bilgisayar içindeki iletişimde ve sistemin görevleri yerine getirmesinde kesmelerin kullanılması avantajları vardır. Yürütülmekte olan bir kesmenin başkaları tarafından yeniden kesilip engellenmemesi için gelebilecek diğer kesmelerin gizlenmesi ne kadar önemliyse, sistem için hayati önem taşıyan gizlenemeyen kesmelerin (nonmaskable interrupts) de engellenmeden çalışmasının sağlanması da o kadar önemlidir. Bu durumu bir örnekle açıklayacak olursak; güç kaynağından CPU'ya gelen ve herhangi bir ani elektrik kesintisini haber veren kesme maskelenemez kesme türündedir ve böyle bir

kesme geldiğinde CPU'nun yapmakta olduğu işi bırakarak bu kesmeye cevap vermesi ve gerekli önlemleri alması gerekir. Böyle bir durumda, sistem, kondansetörlerindeki yük ile bir kaç mikrosaniye daha çalışabilir. Güç kaynağının ilk işaretinde üretilen bu tip bir maskelenemez kesme, bilgisayarın düzenli ve güvenilir bir şekilde kapatılmasını sağlamak için aktif hale getirilir. Bu maskelenemez kesme sayesinde bilgisayar, veri kaybı ve disk dosyalarında bozulma olmadan kapatılmış olur. Kesmelerin özellikle maskelenemez kesmelerin bir uygulama alanı da gerçek zamanlı sistemlerdir (real time systems). Bir fabrika ortamında, bir yerdeki sıcaklık ve basıncın düzgün aralıklarla alınıp kontrol edilmesi gerektiği zaman, bu bilgilerin anında ve doğru olarak alınması ve gerektiğinde ona göre bir aksiyonun anında üretilmesi önemli olduğundan, bu bilgilerin maskelenemez kesmeler kullanılarak alınıp değerlendirilmesi yapılır.

Yukarıdaki açıklamalardan anlaşılacağı üzere; bilgisayar mimarisi içinde CPU'nun çevre birimleri ile iletişimde kesmelerin büyük önemi vardır. Kesmeler kullanılmadan iki birim arasındaki iletişim sağlıklı bir şekilde olmamaktadır. Bu da bilgisayarda performans ve etkin kullanma oranını düşürmektedir. CPU'nun çalışmasının her anında kesme olayını görmek mümkündür. CPU'nun aktif olarak çalışmadığı anlarda bile, çevre birimlerinden birimin durumu bildiren birçok kesme gelir. CPU gelen bu kesmelerden hangisini alıp, hangisini alamayacağını kullanan uygun kesme yöntemi sayesinde karar verir ve ona göre işlem yapar.

KAYNAKLAR

- 1- Biggerstaff, J. Ted : Systems Software Tools, Prentice Hall, 1986.
- 2- Brey, B. Barry : Intel Microprocessor : 8086/8088, 80186, 80286, 80386, 80486 : Architecture, Programming and Interfacing, MacMillan, 1991.
- 3- Clements, Alan : Principles of Computer Hardware, McGRAW Hill, 1989.
- 4- Demirci, Fazıl : İşletim Sistemi Ders Notları, G.Ü. End. San. Eğitim Fakültesi, 1993

- 5- Hayes, P. John : Computer Architecture and Organization, McGRAW Hill, 1988.
- 6- Mano, M Morris : Computer System Architecture, Prentice Hall, 1978.
- 7- Mathur, P. Aditya : Introduction to Microprocessors, McGRAW Hill, 1988
- 8- Mesud, Omar : Nuzum El-Tesgi, Dar el-mustakbal, Ürdün, 1988
- 9- Tanenbaum, S. Andrew : Modern Operating Systems, Prentice Hall, 1992.
- 10- Zaky, G. Safwat, Hamacker, V. Carl : Computer Organization, McGRAW Hill, 1990