

GAZİ ÜNİVERSİTESİ MÜHENDSİLİK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ



SELİN CANSU AKBAŞ

191180005

Araştırma Ödevi

PROF. DR. M. ALİ AKÇAYOL

Bilgisayar Mimarisi BM 311

Interrupts

## İÇİNDEKİLER

<b>1. ÖZET</b> .....	3
<b>2. INTERRUPTS</b> .....	4
<b>3. SONUÇ</b> .....	9
<b>4. KAYNAKÇA</b> .....	10

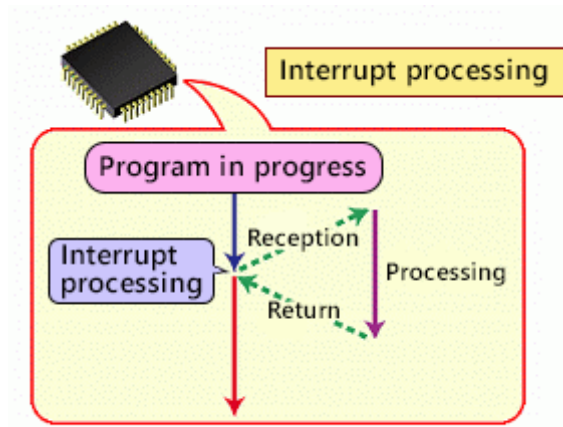
## 1. ÖZET

Interrupt kelimesinin anlamına da bakarsak kesme veya kesinti olarak görürüz. Temel bilgisayar teknolojisinde en önemli özelliklerden biri olarak görüldüğünü söyleyebiliriz. Bütün bilgisayar ve mikrodenetleyicilerde interrupt (kesme) bulunmaktadır. Günümüzde kullandığımız bilgisayarlarda böyle bir imkan olmasaydı yaptığımız işlemleri çok daha yavaş yapardık ve yaparken zorlanırdık. Kesmeler sayesinde CPU meydana gelen hatalardan haberdar olur ve ona göre işlem yapar. Bu kesmeler I/O düzeni ile ilgilidir. Kesmeler CPU'ya içerden veya dışardan olmak üzere çeşitli yerlerden gelebilir. Interrupt işlemcinin yürütmekte olduğu koda ara vererek başka bir kodu çalıştırmasına denir. Bu işlem sadece bilgisayarlarda değil, elektronik donanımsal katmana sahip tüm cihazlarda zorunlu bir durum olarak karşımıza da çıkar. Intel işlemcilerde bunların çeşitleri bulunur. Bu kesmeler farklı mikroişlemci yapılarında farklı performans sergilerler ve bu performanslar mikroişlemciye doğrudan etki eder.

## 2. INTERRUPTS

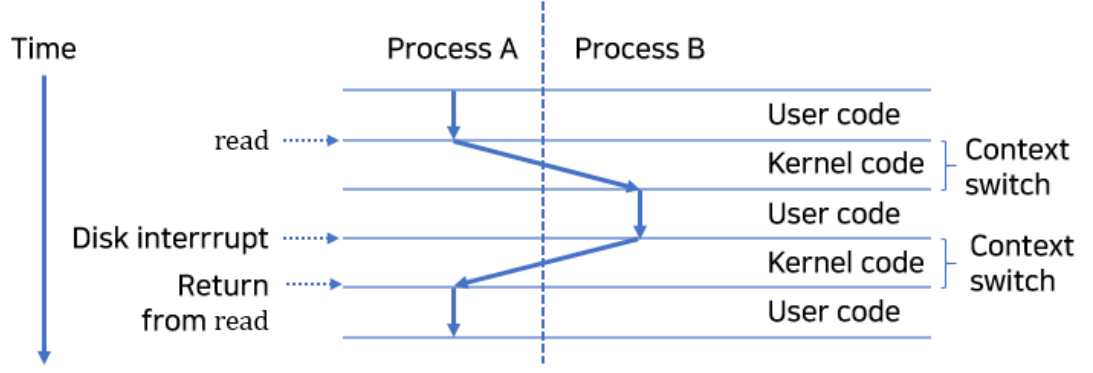
### 2.1. Interrupt Nedir?

Interrupt en genel tanımıyla işlemcide yürütülmekte olan bir olayı keserek başka bir iş yapılmasına denir. Yani önceliği yüksek işlerin mikrodenetleyici tarafından ana programın akışının kesilerek yapılması işlemidir. [1]



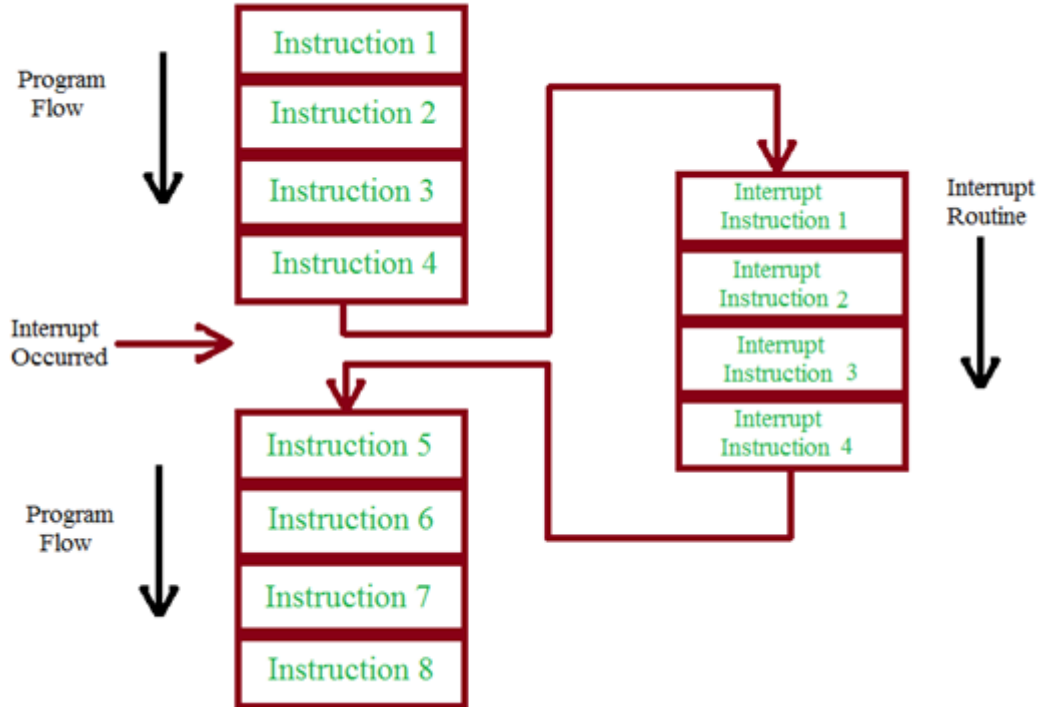
Şekil 2.1. Interrupt işlem süreci

Interruptlara yazılım veya donanım kaynaklı gerçekleşen bir olayın işlemciye acil bir şekilde bildirilerek işlenmesi gereken durumlarda karşılaşıyoruz. Bu durumlar işlemci üzerinde işlenmekte olan processin bitmesini bekleyemeyecek kadar acil oldukları için bekletilmezler ve işlenirler. İşlemci bir interrupt ile karşılaştığı zaman yapmakta olduğu işi o an için bırakır. Context Switch ile işlemci kaldığı işin yerini işaretler. Context Switch bir işlemin veya iş parçacığının durumunu saklama işlemidir. Böylece geri yüklenilebilir ve bırakılan işe dönülebilir. Daha sonrasında yürütmeye devam edebilir. Gerekli işleme gerçekleştirildikten sonra interrupt bırakılır ve işlemci önceden yaptığı işe geri döner. [2,3]



Şekil 2.2. Context Switch kavramı

Interrupt, yazılımın sistem çağrılarını kullanması veya kernel ile haberleşmesine olanak sunan bir mekanizmadır. Yani user space uygulamasının kernel space çağrıları ile gerçekleştirmesi için kullanılır. [4]



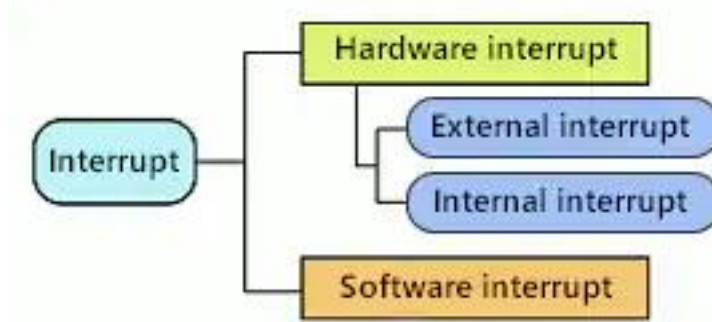
Şekil 2.3. Interrupt işleyişi

## 2.2. Interrupt Kullanım Amacı Nedir?

Interrupt genellikle çok hızlı yapılması gereken işlemlerde kullanılır. Bir programda hata meydana geldiğinde sistemin kitlenmemesi için kullanılır. İşlem bekletmemek ve birçok işlemi aynı anda gerçekleştirebilmek için kullanılır. Senkronizasyon gerektiren uygulama ve sistemleri tasarlamak için kullanılır. Örnek olarak alarm sistemleri, zamanlayıcılar verilebilir. [5]

## 2.3. Interrupt Çeşitleri

Interrupt çeşitlerine gelecek olursak interruptları 2'ye ayırabiliriz. Bunlar Software Interrupt ve Hardware Interrupt olarak geçer.



Şekil 2.4. Interrupt çeşitleri

Donanım kesintisi, çevre birimdeki devrelerden gelen kesme isteği sinyali ile gerçekleşir. Yazılım kesintisi ise özel bir talimat yürüterek gerçekleşir. [6]

*Yazılım Kesmeleri (Software Interrupts):* Bu kesme türü yazılımsal olayları içerir. Özel bir komuttan veya işlemcideki istisna olan durumdan kaynaklanan bir kesme türüdür. Sadece çekirdek ile iletişim kurar ve CPU ile olan iletişimi dolaylı olarak keser. Yazılım kesmesi sırasında sadece bir bit bilgi iletimi olur. Genellikle bir I/O isteği gerçekleştirmek için kullanılır. [7]

*Donanım Kesmeleri (Hardware Interrupts):* Donanım kesintisi, klavye, yazıcı gibi çeşitli donanım aygıtlarının herhangi birinden alınan ve programın, işlemcilerin o anda yürütülen işleminden daha öncelikli olarak ilgilenmesi gerektiğini program tarafından alarak bilmesini sağlayan ve CPU'ya gönderilen sinyaldir. Örneğin tuş vuruşları ve fare hareketleri buna neden olur. [8]

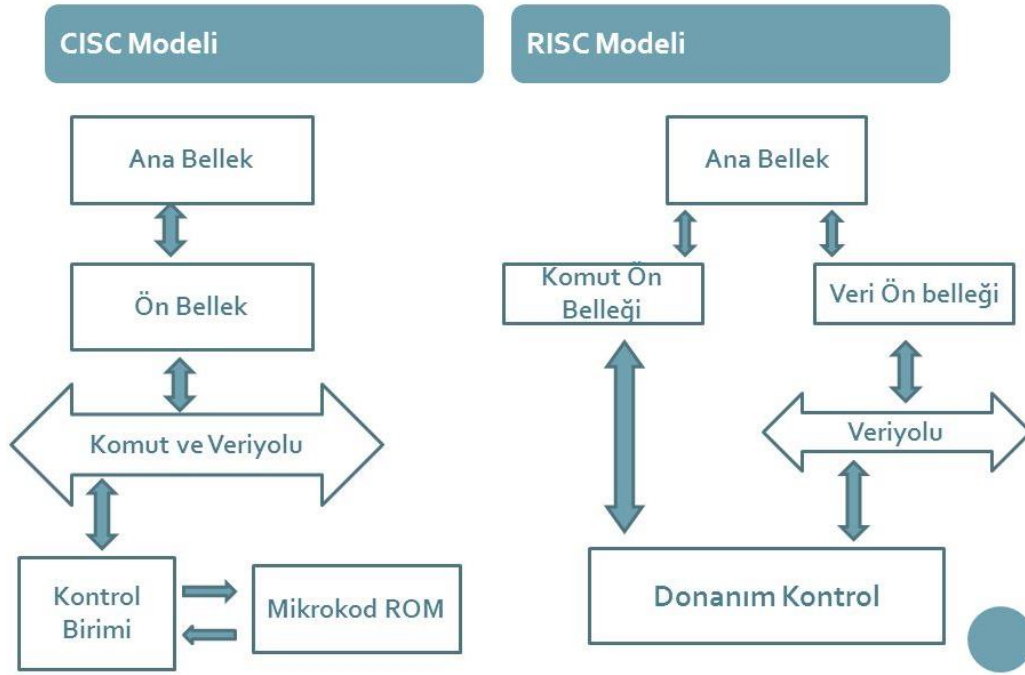
Donanım kesmesinin bir harici kesmesi (external interrupt) ve dahili kesmesi (internal interrupt) vardır. External interrupt, harici kesme terminaline belirli bir sinyal girildiğinde gerçekleşir. Internal interrupt, mikrodenetleyicide yerleşik bir çevresel devreden gelen bir kesme isteği sinyali ile gerçekleşir. Ayrıca donanım kesmeleri, kesme isteği sinyalinin nasıl alındığına bağlı olarak maskelenebilir ve maskelenemez kesmelere sahiptir. [6]

## **2.4. Farklı İşlemci Mimarilerinde Interrupt Performansları**

Farklı işlemci mimarilerindeki interrupt performanslarını incelemek için öncelikle 3 farklı mimariye bakacak olursak bunlar CISC, RISC ve EPIC mimarileridir.

*CISC Mimarisi:* Complex Instruction Set Computer yani Karmaşık Komut Setli Bilgisayar olarak adlandırılır. Endüstride kullanılan eski bir mimaridir. Diğer mimarilere kıyasla daha fazla bileşen ve komut içerir yani daha fazla işlem yapma kapasitesindedir. Fakat bu mimaride aynı anda birden fazla işlerin yürütülmesi önem taşımadığından interruptların performansı düşüktür.

*RISC Mimarisi:* Reduced Instruction Set Computer yani İndirgenmiş Komut Setli Bilgisayar olarak adlandırılır. CISC mimarisi ile kıyasladığımızda daha basit ve sabit uzunlukta komutlarla işlem yapıldığını görürüz. Böylece de işlemler daha hızlı gerçekleşebilir. İşlemci üzerinde her iş parçacığının işlenmesi bittikten sonra interrupt olup olmadığı kontrol edilir. CISC mimarisine göre daha yüksek performans sağlar.



Şekil 2.5. CISC ve RISC mimarileri şeması

*EPIC Mimarisi:* Explicitly Parallel Instruction Computing yani Belirtilmiş Paralel Komutlarla Hesaplama olarak adlandırılır. Bu mimari diğer 2 mimariden sonra ortaya çıkmıştır. Diğerlerine göre daha iyi bir yapıya sahiptir. Intel Itanium'un temelini oluşturur. Bu işlemcide paralel işlemleri derleyen bir derleyici vardır. İşlemler bu derleyiciye bırakılır. Böylece diğer mimarilerde olduğu gibi karışık uzun komutlar çalıştırılmaz. Bu da performansı önemli ölçüde artırır. Interrupt bu mimaride diğer mimarilere göre daha etkilidir.

Bilgisayarda aynı anda gerçekleşen işlem sayısı arttıkça interrupt sayısı da artar. Interrupt performansının en yüksek olduğu mimari de EPIC mimarisidir. Çünkü paralel iş yürütebilmeyi en çok mümkün kılan ve kullanan mimari budur. [9]



### 3. SONUÇ

Bilgisayar mimarisinde kesmelerin büyük bir önemi vardır. Kesmeler olmadan iki birim arasındaki iletişimi sağlıklı bir şekilde oluşturamayız. Bu iletişim oluşturulamadığında da bilgisayar performansı düşer. CPU'nun çalıştığı her an kesme olayını görebiliriz. Kesmelerin bilgisayarda bu şekilde kullanılması genelde sisteme zarar veren bir şey gibi görünüyorsa da aslında böyle değildir. Bilgisayardaki işleri düzenler ve sıraya koyar. Kesmeler CPU'nun çevre birimler ile haberleşmede kullanılabilmesinin yanında, CPU'yu aktif ve etkin şekilde kullanmayı da sağlar. CPU aktif olarak çalışmasa bile birçok kesme gelir. CPU da uygun kesme yöntemi ile hangisinin alınıp alınmayacağına karar verir ve buna göre işlem yapar. Ayrıca farklı mikroişlemci mimarilerinin çalışma yapılarını da burada incelemiş olduk. Interruptın en yavaş çalışan mimari yapıda bile daha çok performans sağladığını görmüş olduk.

#### 4. KAYNAKÇA

1. <https://elektrokod.wordpress.com/2013/10/07/pic16f628-mikrodenetleyicinde-rb0-kesmesi-interrupt/>
2. <https://www.youtube.com/watch?v=HIdH5ofXuuo>
3. <https://www.mshowto.org/interrupt-nedir-hardware-interrupts-software-interrupts.html>
4. <https://www.milliyet.com.tr/egitim/sozluk/interrupt-ne-demek-turkcesi-nedir-interrupt-ne-ise-yarar-6796136>
5. <http://kursatcakal.azurewebsites.net/Makale/Detay/69>
6. <https://toshiba.semicon-storage.com/ap-en/semiconductor/knowledge/e-learning/micro-intro/chapter4/interrupt-processing-types-interrupts.html>
7. <https://tr.theastrologypage.com/software-interrupt>
8. <https://il.farnell.com/hardware-interrupt-definition>
9. <https://bilgisayarmuhendisleri.blogspot.com/2020/05/farkli-islemci-mimarilerinde-interrupt.html>