



**Ali Hadi ALTUNGÖK**

**181180760**

**GAZİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**BİLGİSAYAR MİMARİSİ (BM311 1.Şube)**

**ÖDEV 2 -INTERRUPT**

**M.ALİ AKCAYOL**

**Ders Asistanı : Ayşegül Koçak**

**KASIM 2020**

## ÖZET

Interrupt teknolojisi mikroişlemcinin yapmakta olduğu ana işi, bir Interrupt sinyali geldiğinde bırakarak ve gelen alt işi gerçekleştirip ardından bıraktığı ana işi kaldığı yerden devam ettirerek, çevre cihazların işlemlerini yaparken ve Interrupt gerektiren her türlü durumda bu durumu geçekleştirerek işlemlerin gerçekleştirilmesinde ve eş zamanlı iş yapma anlamında paralelliği arttırarak işlemcinin gerçek kapasitesini ortaya koymasını sağlayan bir teknolojidir. Bu şekilde işlemci çevre birimlerdeki işlerin bitmesini beklemeyerek zaman kaybetmemiş olacaktır. Aynı zamanda aynı anda birden fazla iş gerçekleştirerek gerçek kapasitesini ortaya koymuş olacaktır. Bu araştırma ödevinde bu teknolojinin ne olduğundan, nasıl gerçekleştiğinden, algoritmik olarak işleyişin nasıl meydana geldiğinden bahsedeceğiz. Interrupt çeşitlerini, farklı kaynaklarda Interrupt sınıflandırılmasının nasıl yapıldığını ve bu sınıfların bilgisayar mimarisinde nasıl gerçekleştiğini açıklayıp, bu Interrupt türlerinin örneklerinden bahsedeceğiz. Çalışmanın son bölümlerinde, Interrupt teknolojisinin farklı mikro işlemci mimarilerinde kullanılıp kullanılmadığından, kullanılıyor ise nasıl ve ne şekilde kullanıldığından, hangi mimarilerde daha verimli bir şekilde kullanıldığından bahsederek bu mimarilerin karşılaştırılması yapılarak Interrupt Teknolojisi hakkında kapsamlı çalışma ödevi gerçekleştirilmiş olacağız.

## INTERRUPT

Interrupt mikro işlemcide, işlemler gerçekleşirken beklenmedik başka bir iş ile yapılan işin kesilmesidir. İşlemler gerçekleşirken işleyişi kesen interrupt olarak adlandırılan iş, iş önceliğine sahiptir ve bu iş gerçekleşip bitene kadar yapılır. Bittikten sonra normal işleyişe devam eder. Fakat her iş kesilmesi olan, gerçekleşmesi için başka işlerin gerçekleşmesinin beklendiği durumlar interrupt olmaz. Interrupt olması için gelen işin beklenmedik bir şekilde gelmesi ve öncelik sahibi olması gerekir. İşlemci elindeki işi tamamen bırakıp o işi gerçekleştirdikten sonra kaldığı işe devam eder. Yapılan işte anlaşılabileceği gibi Interrupt türkçeye kesme işlemi olarak geçmiştir.

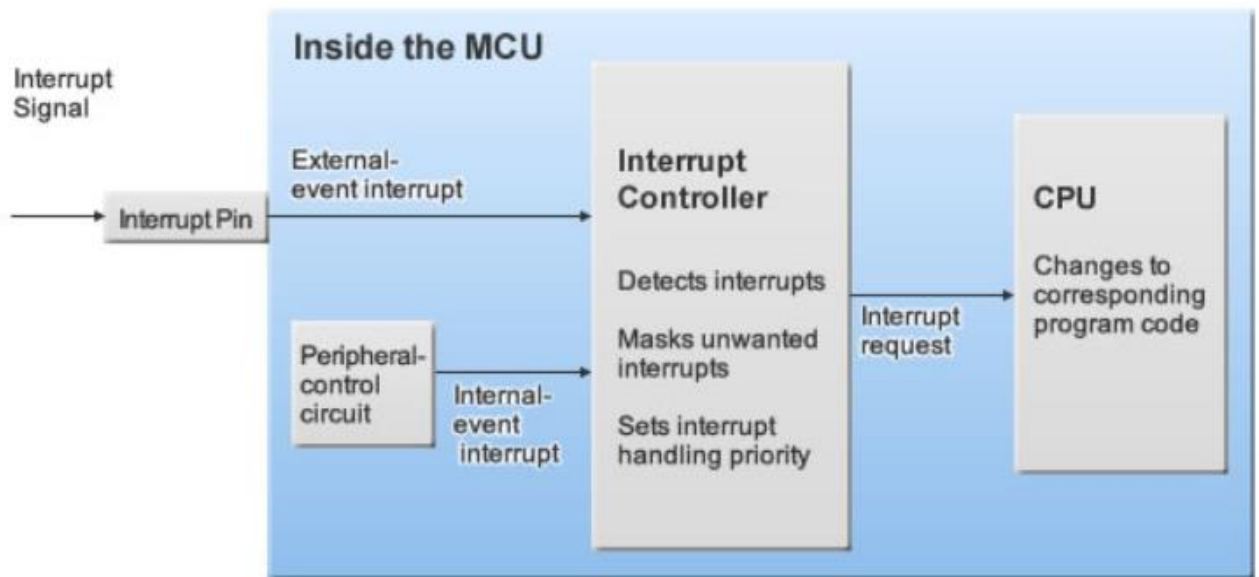
Interrupt olayı ilk bakışta işlemcinin daha yavaş çalışmasına ve işinin ertelenmesine sebep oluyor gibi anlaşılıyor olabilir fakat durum böyle değildir. Interrupt gerçekleşirken alt program bitirilmeden üst programın işine devam edilmez. Alt programlar kullanılarak program içerisinde kullanılacak komut sayısı azalır ve bir sürü mantıksal karışıklıklar önlenmiş olur.

Günlük hayattan örnek verecek olursak trafikte ilerleyen bir araç ışıkları geçmeden aniden kırmızı ışık yanarsa duracaktır fakat bu durum Interrupt olmaz çünkü o an kırmızı ışığın yanması her ne kadar aniden gerçekleşse de hiç beklenmedik bir durum değildir ve trafikte kırmızı ışığın yanması gayet olağandır. Fakat yeşil ışık yanarken araç tam geçecekken yola ambulans çıkması beklenmedik ani bir olaydır. Trafikte Ambulans geçme önceliğine sahip olduğundan ve diğer araçların ona yol vermesi gerektiğinden bu durum Interrupt olarak adlandırılabilir. Ve Interrupt olayına uygun bir örnek olarak verilebilir.

Mikroişlemcide gerçekleşen Interrupt kesilmesi durumundan gerçek anlamda bahsedecek olursak, bilgisayarın port girişlerinden yada donanım içerisindeki herhangi bir sayıcıdan gelen sinyal nedeniyle işlemcinin belleğinde çalışmakta olan programın kesilme olayıdır. Programın kesildiği andan itibaren önceden hazırlanan alt program çalışır. Aynı zamanda çevresel aygıtların mikroişlemciye erişimini yani Interrupt durumunun gerçekleşmesi ile işlemcinin diğer aygıtlardan mouse yazıcı gibi gelen işlemleri gerçekleştirmesidir.

Mikro işlemci Interrupt'ı kısa bir program olan IRS(Interrupt Service Routine) programı ile karşılar ve nasıl işlem yapacağını bilir. Micro işlemcide Interrupt algoritmik olarak şu adımlarla gerçekleşir:

- Ana program çalıştırılmaktadır.
- Bir kesinti sinyali MCU(Micro Controller Unit)'a bir Interrupt olayının meydana geleceğini bildirir.
- MCU kesme sinyalini alır ve ana programın çalıştırmakta olduğu ana işlemini askıya alır.
- MCU mevcut program yürütme durumunu kendi kayıtlarına kaydeer.
- MCU, alınan Interrupt a karşılık gelen Interrupt rutinini yürütür.
- MCU, Interrupt ile genen işi bitirir.
- MCU kaydedilmiş olan ana programın yürütme durumunu geri yükler.
- Ana programın yürütülmesine devam eder.



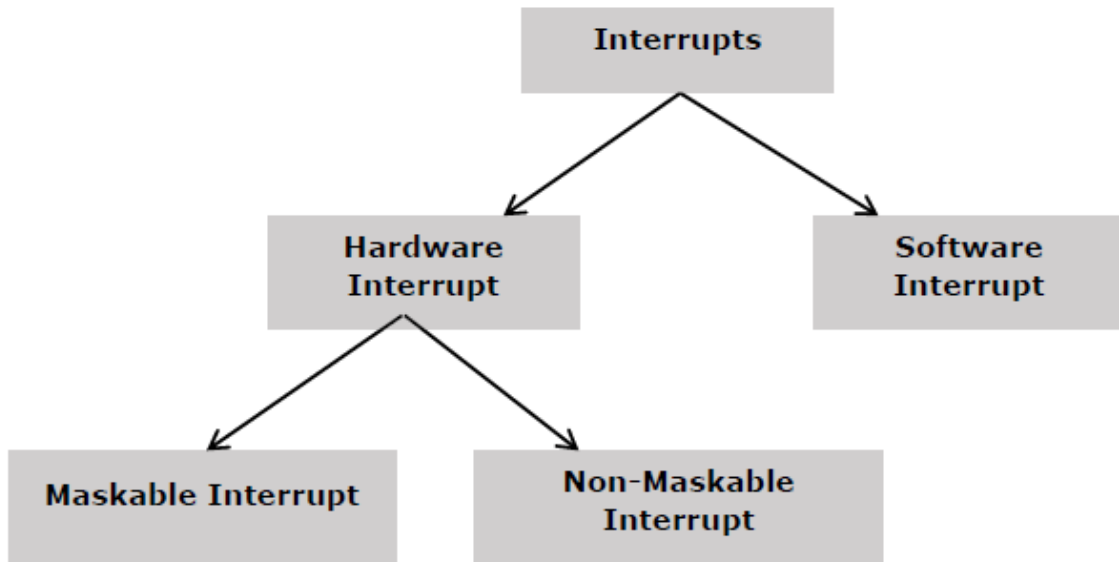
### Interrupt Processing Within the MCU

MCU nun içindeki sürece yakından bakacak olursak MCU ya gönderilen kesme sinyali olay harici bir cihazda meydana gelirse sinyal MCU nun INT pinine gönderilir. Olay MCU nun yonga üstü çevresel bölgelerinde meydana gelirse Interrupt(kesinti) sinyali dahili olarak verilir. Bu sinyalleri MCU nun Interrupt Control(IC) ı tarafından alınır ve işlenir. Birden fazla Interrupt sinyali alınırsa, IC ın mantığı bunların hangi sırayla ele alınacağına, her cihazın öncelik seviyesine göre karar verir. Ardından kesme istek sinyallerini uygun sırada

CPU'ya gönderir. CPU, isteği aldığı anda mevcut program yürütmesini askıya alır ve ardından kesmeye karşılık gelen kesme işleme kodunu yükler ve çalıştırır.

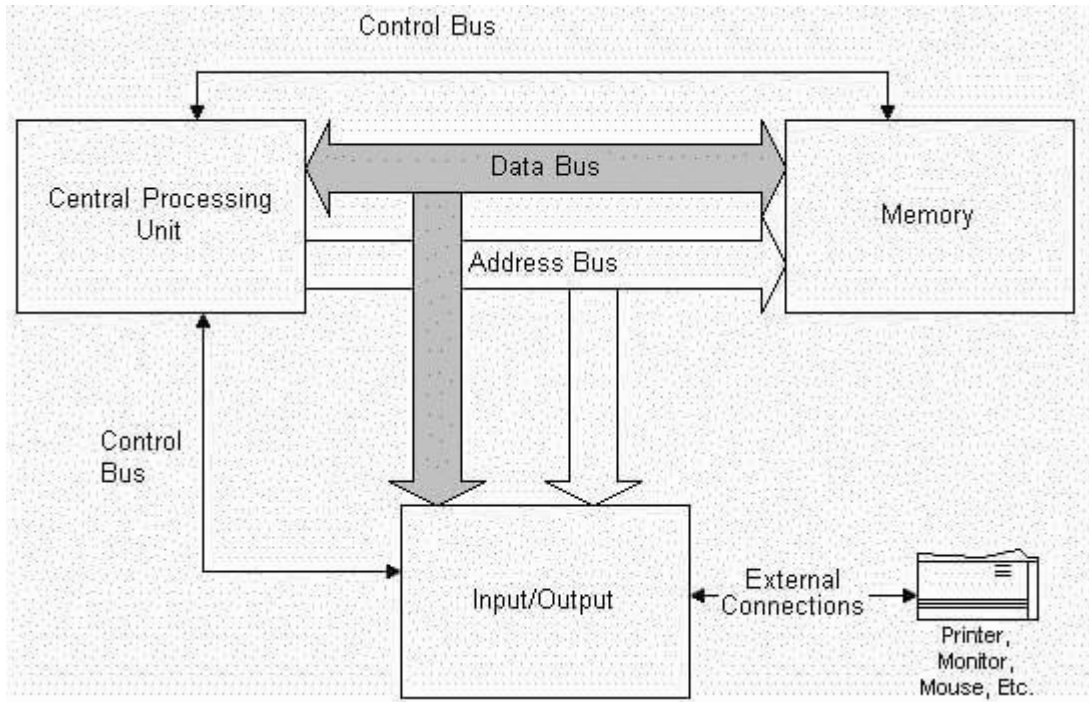
### **Interrupt Çeşitleri:**

Bazı kaynaklarda Interrupt ikiye ayrılmış. Hardware Interrupt ve Software Interrupt olarak sınıflandırılmıştır. Ek olarak Maskable Interrupt Non-Maskable Interrupt olarak da ikiye ayrılmakla birlikte Spurious Interrupt, Periodic Interrupt (Periyodik), Aperiodic Interrupt (Aperiyodik) , Synchronous Interrupt(Senkron) ve Asynchronous(Asenkron) Interruptları da ayrıntıya inilerek kaynaklarda mevcuttur.



Hardware Interruptlar donanımsal Interruptlardır. Donanımsal Interruptlar genellikle fare veya klavye gibi bir giriş aygıtı tarafından oluşturulur. Örneğin klavye kullanılarak bir şey yazılacak olursa, bir tuşa basacak olursak işlemci girdiyi hemen işlemesi gerekir. Benzer şekilde bir fare hareket ettirildiğinde yada tuşuna basıldığında yada dokunmatik ekrana sahip olan bir aygıtın ekranına dokunulduğunda, işlemin işlemci tarafından hemen gerçekleştirilmesi Hardware Interrupt'a örnek olarak verilebilir. Her Interrupt isteği gönderen aygıt kendisine özel bir IRQ( Çalışan ana programı geçici olarak durduran ve bunun yerine özel bir programın , bir Interrupt işleyicisinin çalışmasına izin veren bir donanım

sinyali) sinyaline sahiptir. Böylelikle Interrupt cihazlarında çakışmayı önlenir ve fare ve klavye gibi cihazlara öncelik verilerek sorunlar ortadan kaldırılır.



### Hardware Interrupt

Software Interruptlar ise bir program çalışırken oluşan hataları ve istisnaları işlemek için kullanılır. Örneğin program bir değişkenin geçerli bir integer olmasını bekliyorsa fakat değişken null değere sahip ise programın çökmesini önlemek için bir Interrupt üretilebilir. Benzer şekilde sonsuz bir döngüyü kırmak içinde Interrupt kullanılabilir. Bu durumlarda yazılımsal Interruptlara örnek olarak verilebilir.

Masking(Maskeleme) den bahsedecek olursak işlemciler Hardware Interruptların seçici olarak etkinleştirilmesine ve devre dışı bırakılmasına izin veren dahili bir Interrupt Mask Register(Yazmaç)'ına sahiptir. Her Interrupt Mask Registerinde bir bit ile bağlantılıdır. Interruptın etkinleştirilmesi ve devre dışı bırakılması bu bitin aktifleştirilmesi ile gerçekleştirilir. Bazı durumlarda Interrupt sinyali Interrupt Making den etkilenmez. Bu nedenle devre dışı bırakılamazlar. Bu durumlarda Masking Register ile engellenemeyen Interruptlar Non-Masking Interrupt olarak adlandırılır.

Spurious Interrupt(Sahte Interrupt) dan bahsedecek olursak Interrupt girişinde geçersiz kısa süreli sinyaller olarak sınıflandırılmıştır. Bu tür Interruptlar elektriksel parazit ile arızalı cihazlarla birlikte meydana gelen aksaklık durumlarında gerçekleşir. Aynı zamanda bir cihaz ve sistemin aynı anda iki veya daha fazla işlemi gerçekleştirmeye çalıştığında ortaya çıkan istenmeyen durumlarda gerçekleşir. Bu işlemlerin düzgün bir şekilde gerçekleştirilmesi için uygun bir sıra ile yapılması gerekir.

### **Interrupt Kullanım Amacı**

Harici cihazların hızı CPU dan daha yavaştır. Dolayısıyla harici cihazlarda işlem yapılacağı zaman herhangi bir kesinti olmadığı durumda CPU harici cihazların hızının kendisiyle eşleşmesini bekleyerek çok fazla zaman kaybeder. Bu durum CPU nun verimliliğini azaltır. Bu nedenle bu sınırlamaları ortadan kaldırmak için Interrupt yapılması gerekir. Bu durumu daha iyi açıklamak için şöyle anlatabiliriz. Yazıcıya CPU nun talimat verdiğini sayalım. CPU yazıcı bu talimatları yerine getirirken beklemek yerine yapmakta olduğu işi devam ettirir ve yazıcı işini yerine getirince işini yaptığını CPU ya söyler. Bu şekilde CPU nun bekleme süresi kısaltılmış olup kapasitesini daha aktif bir şekilde kullanarak daha verimli bir işlemci elde etme amacı gerçekleştirilmiş olunur.

### **Mikroişlemci Mimarilerinde Interrupt Performansı**

Complex Instruction Set Computer (CISC) mimarisinin çıktığı yıllarda hafıza elemanları düşük kapasiteye sahip olduğundan hafızanın doğru ve verimli bir şekilde kullanılması yönünde oluşmuş bir mimaridir. Endüstride kullanılan en eski mimaridir. Hafıza birimlerinin zamanla gelişmesi ile bu mimarinin yerini RISC ve EPIC mimarileri almıştır.

CISC mimarisi oldukça uzun ve karmaşık kod bloklarına sahiptir. Bu mimarinin kullanıldığı dönemlerde hafıza birimlerinin kapasiteleri zayıf olduğundan bu şekilde tasarlanmışlardır işlemciler. Dolayısıyla işlemlerin paralel olarak gerçekleştirilmesi önemli ölçüde baz alınmadığından Interrupt teknolojisi bu mimaride gelişmemiştir. Motorola 6800 , 6809 ve 68000, Intel 8080 ,iAPX432 ve x86 ailesi , Zilog Z80, Z8, Z8000 aileleri , MOS Technology 6502 ailesi işlemcileri CISC mimarisine sahip olan işlemcilerdir.

Reduced Instruction Set Computer(RISC) mimarisi incelendiğinde CISC mimarisine kıyasla çok daha basit ve sabit uzunlukta komutlarla işlem görülmektedir. İşlemler doğrudan bellek üzerinde yapılmamaktadır. Hafıza birimine sadece okuma ve yazma işlemleri yapmak için gidilmektedir. Komutlar daha basit ve kısa olduğundan daha çok işlem görülmektedir. CISC mimarisinden daha sonra ortaya çıkmıştır. Hafıza kullanımı CISC mimarisi ile karşılaştırıldığında daha fazla olduğu görülmektedir. Ancak gelişen hafıza birimleri kapasiteleri ile birlikte RISC mimarisinin kapasitesindeki artış bu durumu fazlasıyla tolare etmektedir.

RISC mimarili işlemcide mikroişlemci üzerinde gerçekleşen her işlemden sonra Interrupt olup olmadığına bakılmaktadır. Aynı zamanda işlemci hafıza birimleride sadece veriyi yazmak ve okumak için gider böylelikle paralel anlamda iş yolları oluşturulmuş olur ve Interrupt teknolojisindeki kesme sinyallaeri sayısında işlemci gerçekte sahip olduğu kapasiteyi daha verimli kullanarak işlemcinin performansı arttırılmış olur.

Explicitly Parallel Instruction Computing(EPIC) mimarili mikro işlemciler RISC ve CISC mimarili mikro işlemcilere göre daha yüksek kapasiteye sahiptirler. Intel Itanium mikro işlemcileri temelde bu mikroişlemci mimarisine sahiptir. Bu mimari paralelliği en çok destekleyen mimari olmakla birlikte yüksek kapasiteli olarak paralelliği gerçekleştirebilen mimaridir. Bu mimaride işlemci, işlemci üzerindeki komutların paralel bir şekilde etkileşimini derleyicilere bırakmaktadır. Bu şekilde kendi yüksek kapasitesini daha verimli bir şekilde gerçekleştirmektedir. Paralelliğin artması ile birlikte eş zamanlı olarak daha fazla işlem gerçekleştirildiği ortadadır. Bu düşünülecek olursa ve Interruptların, paralelliği ve eşzamanlı iş yapma kapasitesini arttırdığını da ele alacak olursak, işlem kapasitesinin arttırılmasının ve paralel iş kapasitesinin arttırılmasının diğer mimarilerdeki işlemcilere göre daha kaliteli ve yüksek teknolojili Interrupt teknolojisi kullanımı ile gerçekleştirildiği görülmektedir.



**Kaynakça:**

- <https://en.wikipedia.org/wiki/Interrupt>
- <https://www.renesas.com/us/en/support/technical-resources/engineer-school/mcu-05.html>
- [https://www.tutorialspoint.com/microprocessor/microprocessor\\_8086\\_interrupts.htm](https://www.tutorialspoint.com/microprocessor/microprocessor_8086_interrupts.htm)
- <https://www.oreilly.com/library/view/understanding-the-linux/0596005652/ch04s06.html>
- <https://www.researchgate.net/post/What-is-the-difference-between-an-internal-interrupt-and-software-interrupt>
- <https://teachcomputerscience.com/interrupts/>
- [https://en.wikipedia.org/wiki/Interrupt\\_request\\_\(PC\\_architecture\)#:~:text=In%20a%20computer%2C%20an%20interrupt,interrupt%20handle%2C%20to%20run%20instead.](https://en.wikipedia.org/wiki/Interrupt_request_(PC_architecture)#:~:text=In%20a%20computer%2C%20an%20interrupt,interrupt%20handle%2C%20to%20run%20instead.)
- <http://www.onlinecmag.com/information-interrupts-software-hardware/>
- <https://www.geeksforgeeks.org/purpose-of-an-interrupt-in-computer-organization/>
- <https://bilgisayarmuhendisleri.blogspot.com/2020/05/farkli-islemci-mimarilerinde-interrupt.html>
- <http://valhalla.altium.com/Learning-Guides/Legacy/CR0121%20TSK3000A%2032%20bit%20RISC%20Processor.pdf>