



GAZİ ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ

BM311 BİLGİSAYAR MİMARİSİ

Gamze Aksu 171180005

ÖDEV-2
INTERRUPTS

ÖZET

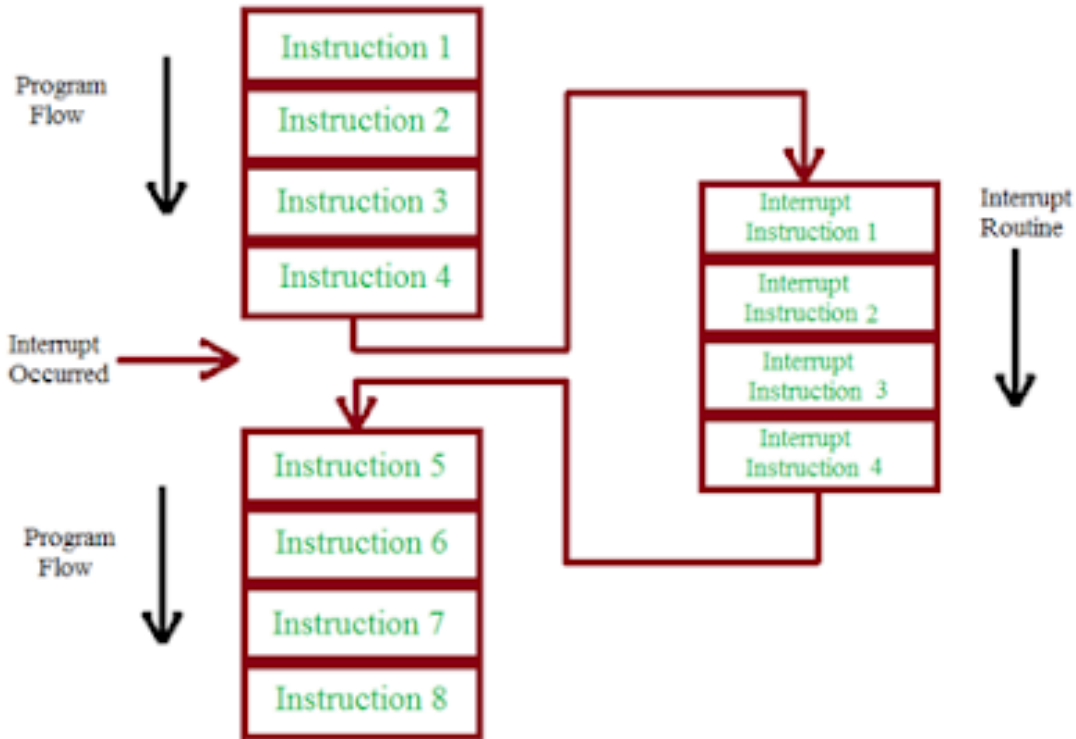
Interrupt işlemi, işlemcinin bir işlem üzerinde çalışırken gelen bir sinyal ile yaptığı ana işlemi duraklatarak başka bir alt işlemi yapmaya başlayıp, alt işlemi bitirdikten sonra ana işine kaldığı yerden devam etmesidir. Bu şekilde işlemci, çevre birimleri ile paralel iletim yollarına sahip olarak eş zamanlı işlem yapabilme özelliği kazanır. Bu işlem ISR (Interrupt Service Routine) diğer adıyla interrupt handler ile gerçekleştirilir. Genel olarak 2 interrupt çeşidi vardır. Bunlar hardware interrupt ve software interrupt'lardır. Hardware interrupt'lar çevre birimlerinden gelen donanımsal interrupt'lardır. Maskelenebilir ve maskelenemez olarak iki çeşittirler. Software interrupt ise işlemcide interrupt'a sebep olan yazılımsal durumlardan kaynaklanmakla birlikte bir program execute edilirken gerçekleşebilecek interrupt durumlarıdır. Interrupt'lar bilgisayar içinde birçok yerde kullanılırlar ve interrupt'ların farklı mimarilerde farklı performansları vardır.

Bu araştırma dört bölümden oluşur. İlk bölümde interrupt'ların ne olduğundan bahsedilmiştir. Kısaca interrupt oluşturabilen işlemler ve polling metodu anlatılmıştır ve ISR ne olduğu açıklanmıştır. İkinci bölümde interrupt çeşitleri tanımlanmıştır. Üçüncü bölümde interrupt'ların kullanım amaçlarından bahsedilmiştir. Dördüncü bölümde ise üç farklı mimarideki interrupt performansları açıklanmıştır.

1. INTERRUPT NEDİR?

İşlemci yapısı gereği birden fazla işi aynı anda yapamaz. Bir işlem yürütülürken başka bir işlem geldiğinde sıraya alınır. Sırayla işlemler yapılır. Ama bazı durumlarda işlemcinin anlık yaptığı işlemten daha önemli, daha acil işler ortaya çıkar. Bu gibi durumlarda interrupt işlemi gerçekleştirilir.

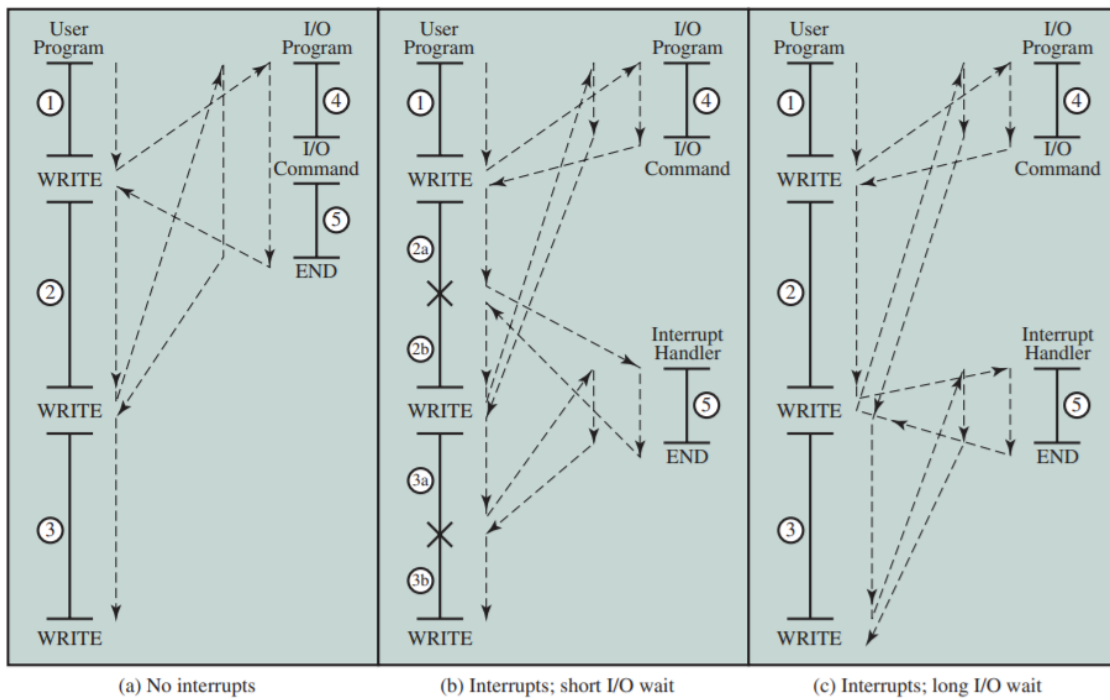
Interrupt (Kesme İşlemi) CPU (Central Processing Unit) bir işlem üzerinde çalışırken hesapta olmayan bir olayın etkisiyle o an yaptığı işi bırakıp başka bir iş yapmaya başlamasıdır. Yani işlemcinin normal işleyişi kesilir ve yeni olay için işlemler yapılmaya başlanır. Bu sırada normalde yapılan işlemin durumu kaydedilerek askıya alınır. Interrupt işlemi bittikten sonra normal işleyişe devam edilir. Daha anlaşılır olması için gerçek hayattaki birkaç interrupt örneğini inceleyelim. Mutfakta yemek yapan bir annenin bebeğinin ağlamasını duyduğunda onun yanına gitmesi bir interrupt işlemidir. Normal işleyişte yemek yapmak için olan talimatlar yapılıyordu. Bebek ağlaması duyulduğunda bebeğin susturulması yemek yapmaktan daha öncelikli bir işlemdir. Yemek yapma işlemi durdurulur. Bebek uyutulur. Bebek uyuduktan sonra da yemek yapma işlemine devam edilir.



Şekil 1: Bir interrupt oluştuğunda işlemcide gerçekleşen olaylar

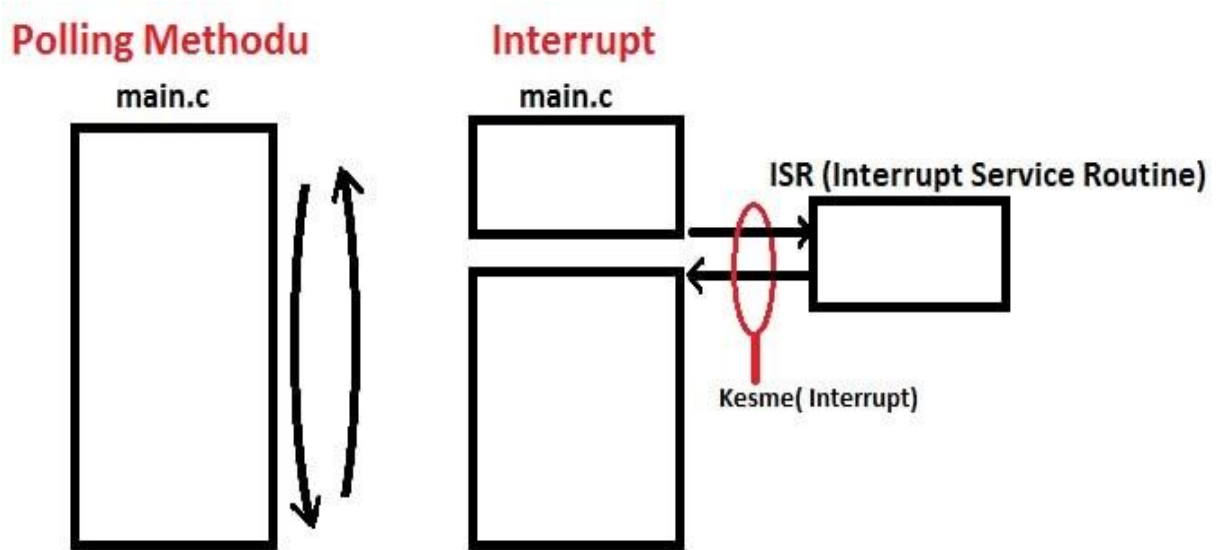
Interrupt oluşturabilen işlemler şunlardır :

- Program: Aritmetik taşma, sıfıra bölme, geçersiz bir makine talimatı yürütme girişimi veya bir kullanıcının izin verilen bellek alanı dışındaki referans gibi talimat yürütmenin bir sonucu olarak ortaya çıkan bazı koşullar tarafından oluşturulur.
- Timer: İşlemci içindeki bir timer tarafından oluşturulur. Bu, işletim sisteminin belirli işlevleri düzenli olarak gerçekleştirmesine izin verir.
- I/O : I/O controller tarafından üretilir. Bir işlemin normal tamamlandığını işaret etmek, işlemciden servis talep etmek veya çeşitli hata durumlarını işaret etmek içindir.
- Donanım: Memory parity hatası, pil uyarısı veya disk okuma hatası tarafından oluşturulur.



✕ = interrupt occurs during course of execution of user program

Interrupt sorununun yazılımsal olarak çözülmesine Polling metodu denir. Çevre birimlerinin kontrol edildiği programın başlangıcında kullanılır. Her zaman sabit bir sırayla işlemleri yürütür. Bu yöntemde hata ayıklamak kolaydır ama işlemler kod çalışırken gerçekleşmez. Bu yöntem CPU kullanımında verimsizdir. Interrupt işleminde polling gibi olayların kaçırılması durumu oluşmaz. Interrupt sadece beklenmedik bir olay olduğunda çalışır bu da işlemciyi yükten kurtarır.



1.2.ISR – Interrupt Service Routine

Genellikle interrupt handler olarak da bilinirler. Belirli bir interrupt koşuluyla ilgili bir kod bloğudur. ISR gelen bir Interrupt isteği tarafından başlatılan bir yazılım işlemidir. ISR ek olarak aygıt sürücülerini veya sistem çağrılarını gibi korumalı işlemler arasındaki geçişleri uygulamak için de kullanılır. Aynı zamanda ISR'nin görevini tamamlama hızı Interrupt'ın neyin tetiklediğine göre değişir. İki tür interrupt işleyicisi vardır: FLIH(First level Interrupt handler) ve SLIH (Second level Interrupt handler).

FLIH daha fazla işlevselliğe sahip olmakla birlikte daha hızlıdır. Ayrıca işlem yürütülürken daha fazla titreşime sahiptir ve bunlar maskelenebilir interruptlardır. SLIH ise işlevselliği daha uzun süreçler gerektiren interruptların işleme görevini yerine getirmesidir. FLIH den farklı olarak SLIH'ler süreçlere ve iş parçalarına benzer şekilde planlanabilir. Nihayetinde FLIH tüm cihaz ve platforma bağlı işlemleri gerçekleştirecekken daha fazla ve uzun vadeli kullanımlar için SLIH kullanılacaktır.

2. INTERRUPT ÇEŞİTLERİ

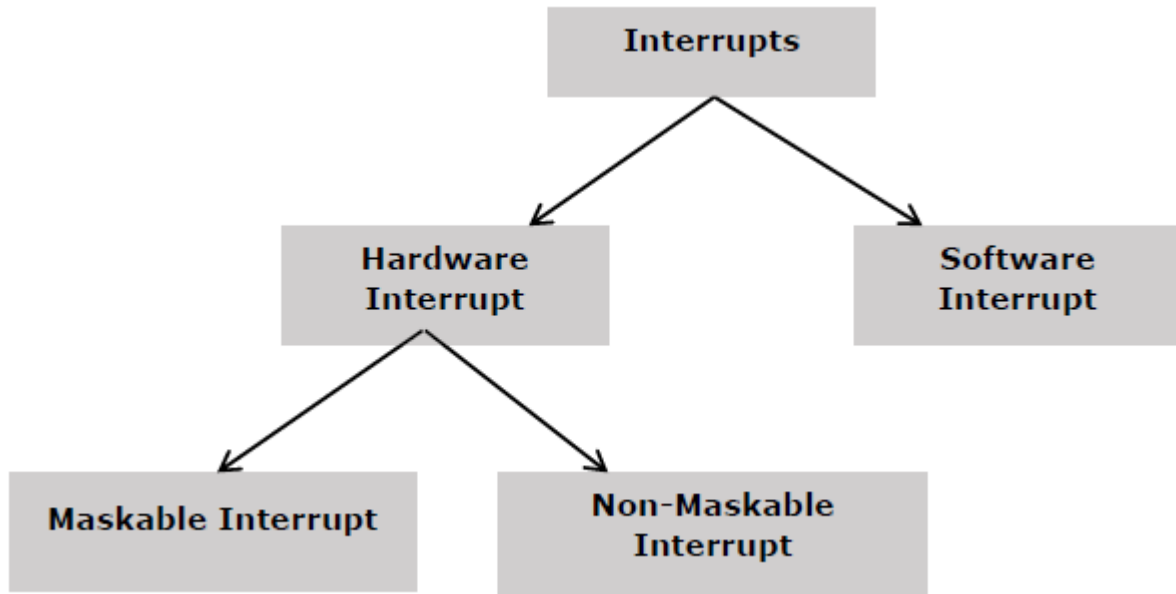
Interruptların kaç çeşit olduğu farklı kaynaklara göre değişkenlik göstermektedir. Bazı kaynaklarda iki çeşit interrupt olduğu söylenir. Bunlar:

- Donanım Kesintileri (Hardware Interrupts)
- Yazılım Kesintileri (Software Interrupts).

Bazı kaynaklarda ise üç çeşit interrupt olduğu söylenmektedir. Bunlar:

- İç Kesmeler (InternalInterrupts)
- Dış Kesmeler (ExternalInterrupts)
- Yazılımsal Kesmeler (Software Interrupts).

Bu konuda kesin bir durum belirtilmemiştir ancak yoğun olarak kullanılan iki türün olduğudur. Bu yüzden bu yazıda bu iki tür anlatılacaktır.



2.1. Hardware Interrupt (Donanımsal Interrupt)

Hardware Interrupt dan bahsedecek olursak tüm cihazlar, IRQ(bir programı geçici olarak durduran ve bunun yerine özel bir programın, bir Interrupt işleyicisinin çalışmasına izin veren bir donanım sinyali) sinyali ile CPU ya bağlanarak CPU timer(zamanlayıcı)'a gömülü cihazlar tarafından algılanır. İşletim sistemi ile çevre cihazlarından Interrupt sinyalleri işlemciye iletilir. Bu sinyali alan işlemci gerekli admları gerçekleştirerek Interrupt işlemini yerine getirir. Hardware Interrupt'a örnek olarak bilgisayarın iç saatinin sayacını gerçekleştiren donanım gibi, klavyeden yazı yazmak istediğimizde bir tuşa basmak veya fareyi hareket ettirmek örnek olarak verilebilir.

2.1.1.Maskelenebilir Interrupt

İşlemciler genellikle bir harici interrupt mask register'a sahiptir. Bu onlara interrupt'ların etkinleştirilmesine ya da devre dışı bırakılmasına izin verir. Her bir interrupt sinyali mask register'daki bir bite bağlıdır. Bit etkin olduğunda interrupt da etkinleştirilir. Bit

etkin olmadığında da interrupt'lar devre dışı bırakılır. Interruptlar devre dışı bırakıldığında ilgili interrupt sinyali dikkate alınmaz normal işleyişe devam edilir. Bu işlemten etkilenen sinyaller maskelenebilir interrupt olarak adlandırılır.

2.1.2.Maskelenemez Interrupt

Maskelenemez interrupt'lar standart interrupt-masking teknikleriyle yok sayılamayacak olan interrupt'lardır. Genellikle geri alınamaz donanım hatalarıyla oluşurlar. Yanıt süresi kritik olduğunda ya da bir interrupt'ın normal çalışma zamanında asla devre dışı bırakılmaması gereken durumlarda kullanılır. Bu kullanımlar arasında genellikle kurtarılamayan donanım hatalarında, sistem debugging işlemlerinde, resetleme gibi özel durumlar vardır.

Modern bilgisayar mimarilerinde hemen ilgilenilmesi gereken işlemler için, mesela kurtarılamayan hataları işlemek, genellikle maskelenemez interrupt kullanılır. Bu yüzden sistemin normal çalışmasında bu tür işlemler maskelenmemelidir. Kurtarılamayan dâhili chipset hatalar, parity gibi sistem hafızasındaki bozulmalar, sistemdeki ve çevresel veri yollarındaki data bozulmaları gibi hatalar örnek gösterilebilir.

Bazı sistemlerde kullanıcılar donanım ve yazılım hata ayıklama ara yüzleri ya da reset butonu ile maskelenemez interrupt'ları tetikleyebilir. Programcılar hata ayıklama işlemi için bazen maskelenemez interrupt'ları kullanabilir. Bu durumlarda işlemlerin monitöre aktarılması için bir interrupt handler çalıştırılır. Bu program ile programcılar hafızayı ve interrupt anındaki iç durumu inceleyebilirler.

2.2. Software Interrupt (Yazılımsal Interrupt)

Software Interrupt, instruction setteki özel bir komutun neden olduğu ya da işlemcinin kendisindeki bir istisnai koşulun neden olduğu interrupt türüdür. Bu, donanım düzeyinde meydana gelen bir hardware interrupt'dan oldukça farklıdır. Hardware interrupt aksine bir yazılım tarafından çağırılır. Bir yazılım kesintisi yalnızca kernel ile iletişim kurar ve dolaylı olarak merkezi işlem biriminin işlerini durdurur. Özellikle hata veya istisna işleme sırasında çekirdekle iletişim kurmanın veya sistem çağrılarını başlatmanın yollarından biri olarak kabul edilir. Her software interrupt sinyali belirli bir interrupt handler ile ilişkilendirilir. Bu aslında bir interrupt olduğunda etkinleştirilen bir rutindir. Bir yazılım kesintisi sırasında yalnızca bir bit bilgi iletilir. Genellikle, bir I/O talebini gerçekleştirmek için bir software interrupt kullanılır. Bu istek, aslında hizmeti gerçekleştiren kernel rutinlerini sırasıyla çağırır.

Bir software interrupt aslında hardware interrupt özelliklerinin çoğuna sahiptir. Hardware interrupt benzer olarak, yalnızca belirli bir interrupt vektör çağırır ve accumulator ve register'larda kayıt tutar. Bir software interrupt bazı hardware interrupt rutinlerini de kullanabilir.

Software interrupt tasarım gereği program derlendiğinde özel bir talimat çalıştırılıp kasıtlı olarak gerçekleştirilebilir. Ayrıca program yürütme hataları nedeniyle Software Interruptlar beklenmedik şekilde tetiklenebilir. Sıfıra bölme gibi ya da integer olarak tanımlı bir değişkenin null olarak gelmesi ya da sonsuz döngülerin kırılması için gerekli olan interruptlar software interrupta örnek olarak verilebilir.

Bunlara ek olarak bazı yerlerde Periodic Interrupt, Aperiodic Interrupt, Synchronous Interrupt ve Asynchronous Interruptlardan da bahsedilmektedir. Periodic Interrupt zaman çizelgesinde belirli aralıklarda periyodik olarak meydana gelen interrupt'lardır. Aperiodic Interruptlarda adından da anlaşılacağı gibi periyodik olmayan interrupt'lardır. Synchronous Interruptlar ise sistem saatine bağlı olarak aynı fazda gerçekleşen interrupt'lardır. Asynchronous Interrupt ise sistem saatinden bağımsız olan ve ona eş fazlı olmayan interrupt'lardır.

3. INTERRUPT KULLANILMA AMAÇLARI

Interruptlar bilgisayarda çok fazla yerde kullanılırlar. Saat ve tarih ayarlamalarında, fare kullanımında, ağ kontrolünde, cd-rom kontrolünde, DOS fonksiyonlarında, klavyede herhangi bir kısa yolda gibi yerlerde kullanılabilirler. Günümüzde tüm bilgisayarlar içinde donanımsal bir interrupt kontrol birimine sahiptir. Buna kısaca PIC (Programmable Interrupt Controler) denir. PIC kullanıcı tarafından interrupt işlemlerini alıp işlemciye sinyal gönderir. Böylece işlemci tarafından çeşitli Interrupt sinyalleri alınarak gerekli şekilde işlemler yapıp Interrupt teknolojisi gerçekleştirilir.

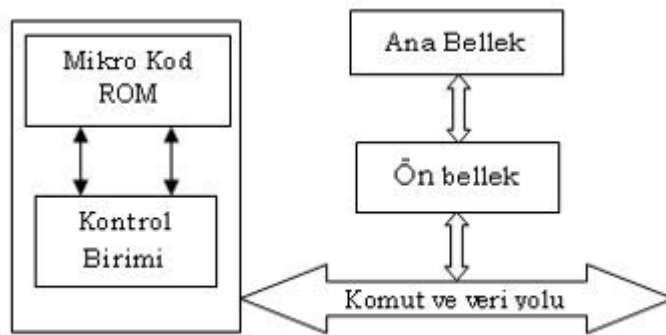
İşlemci yüksek bir hıza sahip olmasına karşın harici cihazlar işlemciye göre yavaştır. Interrupt teknolojisi kullanılmazsa işlemci bu harici cihazları bekleyerek vakit kaybetmiş olacaktır ve bu şekilde CPU verimli şekilde kullanılmamış olur. Bu durumu ortadan kaldırmak için interrupt teknolojisi kullanılmaktadır. Örneğin bir yazıcıdan yazı yazdırma işleminde yazıcı işlemciden daha yavaş çalışır ve işlemci yazıcıyı beklememek için onun işinin bittiği yerde bir Interrupt çalıştırılarak bu sorun çözülmüş olunur. Yazıcı yazma işlemini yaparken CPU da kendi işlerine devam eder ve bu şekilde işlem yapma kapasitesini daha verimli bir şekilde kullanmış olur

4. FARKLI MİMARİLERDE INTERRUPT PERFORMANSLARI

Bu kısımda CISC, RISC ve EPIC mimarilerinin performanslarından bahsedilecektir.

4.1. CISC

Karmaşık komut setli bilgisayar (Complex Instruction Set Computer) olarak bilinen CISC mimarisi en eski mimarilerden biridir. CISC mimarisinin çıktığı zamanlarda hafıza pahalıydı bu yüzden hafızayı ön planda tutan bir mimariydi. Hafızadan tasarruf yapılmak istendiği için karmaşık komut setleri ile işlemciyi yormaya dayalıdır. CISC mimarisinde çok fazla komut sayısının olmasıyla birlikte farklı uzunlukta komut sayıları vardır. Zamanla hafızaların kapasiteleri artıp fiyatları da düştükçe bu mimarinin yerini RISC ve EPIC almıştır.

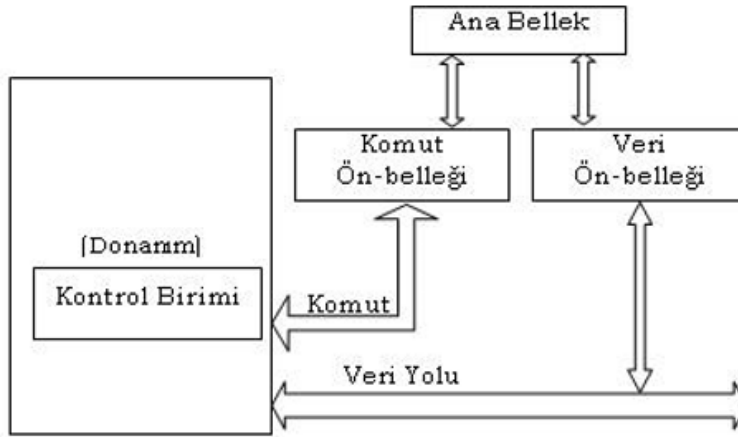


Performansına baktığımızda diğer iki mimariye kıyasla açık ara CISC mimarisin en düşük olduğu görülür. CISC mimarisinin geliştirildiği yıllarda aynı anda birkaç işlemin yapılması önemsenmediği için işlemler paralel olarak gerçekleştirilmemiştir. Bu yüzden interruptların performansı da daha düşüktür. Tabi ki CISC mimarisinin de faydalı yanları vardır. Örneğin interrupt işleminin sık olduğu yoğun ağ trafiğinin yönetildiği bilgisayarlarda sisteme çok yük biner ve bu, sistemin oyalanmasını sağlar. Bu yüzden yoğun ağ trafiğinin yönetildiği sistemlerde CISC mimarisinin olduğu işlemciler kullanılır. Motorola 680000 işlemcisi CISC mimarisine sahiptir. Bu işlemcide komutların paralel çalıştırılması karmaşık donanımsal elemanlar ve işlemler sonucu gerçekleşir.

4.2. RISC

İndirgenmiş Komut Setli Bilgisayar yani Reduced Instruction Set Computer CISC mimarisinden daha sonra ortaya çıkmıştır. Transistor sayısı arttırılmadan sadece mimaride değişiklikler yapılarak işlemci hızı artmıştır. Bu mimari CISC mimarisinden daha fazla hafıza

kullanır. Aynı zamanda bu mimaride çok daha az komut sayısı vardır ve sabit uzunlukta komutlar kullanılır. Bu şekilde işlemler daha hızlı gerçekleşir. Bununla birlikte işlemler doğrudan hafıza üzerinde yapılmaz, iç saklayıcılarda yapılırlar. Belleğe sadece okuma ve yazma işlemleri için erişilmektedir.



RISC mimarileri iş istasyonlarında ve çok fazla hesaplama işlemlerinin olduğu alanlarda kullanılması performansı artırır. Her iş parçacığı işlenip bittikten sonra interrupt kontrolü yapılır. Interrupt var mı yok mu diye bakılır. RISC mimarisinin performansı CISC mimarisine kıyasla daha yüksektir.

4.3. EPIC

Explicitly Parallel Instruction Computing yani Belirtilmiş Paralel Komutlarla Hesaplama olarak bilinen EPIC diğer mimarilerden de sonra çıkmıştır ve onlardan daha başarılıdır. EPIC mimarisi Intel Itanium'un temelini oluşturur. Bu mimari komutların paralel işletilmesini derleyiciye bırakır. Bu nedenle işlemci içerisindeki karmaşık devrelerin sayısı azalır. Karmaşık devrelerin azlığı da performansı yukarı yönde artırır.

İşlemlerin paralel yürütülmesi ve aynı anda birden fazla işlemin gerçekleştirilmesi performansı çok iyi şekilde etkiler. EPIC mimarisi de bu anlamda geliştirilmiş bir mimaridir. Bu mimaride paralellik kod tarafında paralel işlemler yazılmasa bile paralel olarak işlenir. Örneğin if bloğu paralel bir kod değildir fakat EPIC mimarisinde dallanmalar ile paralel olarak işlem gerçekleştirilir.

Diğer iki mimariye kıyasla EPIC mimarisi paralel işlemleri en çok kullanan ve en çok destekleyen mimaridir. Bu yüzden interrupt işlemleri en çok EPIC mimarisinde gerçekleşir ve interrupt işleminde de en çok verimi ve performansı bu mimari sağlar.

KAYNAKÇA

<https://techterms.com/definition/isr#:~:text=Stands%20for%20%22Interrupt%20Service%20Routine,complete%2C%20the%20process%20is%20resumed>

<https://teachcomputerscience.com/interrupts/>

<https://www.best-microcontroller-projects.com/hardware-interrupt.html>

https://en.wikipedia.org/wiki/Non-maskable_interrupt

<https://en.wikipedia.org/wiki/Interrupt>

<https://dergipark.org.tr/tr/download/article-file/296689>

<https://www.techopedia.com/definition/22195/software-interrupt>

<https://komhedos.com/polling-metodu-ile-interrupt-arasindaki-farklar/>

<https://bilgisayarmuhendisleri.blogspot.com/2020/05/interrupt-kesme-nedir.html>

<https://bilgisayarmuhendisleri.blogspot.com/2020/05/farkli-islemci-mimarilerinde-interrupt.html>

<https://yazilimdnyasi.wordpress.com/2019/10/21/bilgisayar-mimarisi/>

<http://www.mademir.com/2010/09/interrupt-cesitleri-kullanlma-alarlar.html>

<https://bilgisayarmuhendisleri.blogspot.com/2020/05/farkli-islemci-mimarilerinde-interrupt.html>

<http://farabi.sutef.gen.tr/bilorg/1-2.htm>