Gazi Üniversitesi

Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

BM311 - Bilgisayar Mimarisi

Prof. Dr. M. Ali Akçayol

Ödev 2 - Interrupt
(05.11.2021)

Alper Kürşat Ünver
151180066

Belge İçeriği

1. Giriş

Bilgisayar tasarımında işlemci tasarımı, önemli bir konudur. İşlemcinin tasarımı ve iyileştirmesi sürecinde bir çok sorunla karşılaşılmış ve üretilen çözümler yıllar içinde daha iyiyi amaçlayacak biçimde şekillenmiştir.

Çoklu programlamanın ve çoklu işlemenin sağlanabilmesi için işlemcinin yürütme akışları arasında geçiş yapabilmesi, bir sürecin yürütmesini durdurup diğer bir sürece geçiş yapabilmesi gerekmektedir.

Akışlar arası bu geçişlere **bağlam değişimi** denmektedir. Bağlam değişimini işlemci düzeyinde olanaklı kılan yapıya **kesme** denmektedir.

Amaç ve Kapsam

Bu araştırmanın amacı BM311-Bilgisayar Mimarisi kapsamında "*Kesme çeşitleri, kullanılma amaçları ve farklı mikroişlemci mimarilerinde kesme başarımlarını*" incelemek ve anlamaktır. Konuyu daha iyi kavrayabilmek adına Bilgisayar Tasarımı ile ilgili kaynakların yanı sıra, bir üst katman olan *İşletim Dizgeleri*[1] ile ilgili kaynaklar da incelenmiştir.

2. Kesme

Kesmeler, ana izlenceyi geçici olarak durduran(askıya alan) ve denetimi dış kaynaklara geçirerek görevlerini yürütmesini sağlayan etkinliklerdir. Sonrasında denetimi kaldığı yerden devam edebilmesi için yeniden ana izlenceye verir.[4]

Yürütme Döngüsü tamamlandıktan sonra, etkinleştirilmiş bir kesmenin oluşup oluşmadığına dair bir sınama yapılır. Eğer etkinleştirilmiş kesme gerçekleştiyse Kesme Döngüsü'ne girilir. Bu döngüde gerçekleşecekler tasarımdan tasarıma değişiklik gösterebilir.[3]

Örneğin; 8051'de kesmelerin önceliklendirilebildiği IP(*Kesme Önceliği*[2]) yazmacı[3] vardır. IP yazmacında kesmeye ilişkin tanımlı biti değiştirerek kesmelerin öncelik düzeylerini değiştirebiliriz. *Kesme izlemi*[4] aşağıda anlatıldığı gibi yürütülür.[4]

- Düşük öncelikli bir kesme ancak daha yüksek öncelikli bir kesme tarafından kesilebilir ancak kendinden düşük öncelikli bir kesme tarafından çalışması aksatılamaz.
- Öncelik düzeyleri birbirinden farklı iki ayrı kesme aynı anda geldiği zaman önceliği yüksek olan kesme işleme alınır.
- Eğere öncelik düzeyleri birbirinin aynısı iki kesme aynı anda gelirse, dahili yoklama döngüsü hangi kesmenin işleme alınacağına karar verir.

---

[1] Operating Systems

[2] Interrupt Priority

[3] IP register

[4] strategy

2.1.Kesmelerin Kullanım Amaçları

2.1.1.Çoklu-işlem

Birden fazla işlemi aynı anda gerçekleştirebilmek, birden fazla *izlence*[5]yi aynı anda çalıştırabilmek, bilgisayar tasarımının ilk dönemlerinde en önemli amaçlardandır. Bir diğer izlence çalıştırabilmek için çalışan izlenceyi kapatmak zorunda kalmak, gerek kullanıcı deneyimi gerek kaynak kullanımı açısından sorundur. Günümüz işlemci mimarileri *kesme odaklı*[6]dır ve *bağlam değişimleri*[7]ne olanak tanıyacak biçimde tasarlanmıştır.
Bağlam değişimi; CPU'nun bir *süreç*[8]ten diğer bir sürece geçiş yapabilmesidir.[2]

2.1.2. Eşzamanlama

Eşzamanlama gerektiren uygulama ve dizgelerin tasarımında kesmelerin kullanılması gerekmektedir. Komut düzeyinde koşutluk ve pipelining süreçlerinde geçişler kesmelerle yapılmaktadır.
Aşağıdaki görselde klasik bir RISC pipeline'ına ait tablo gösterilmiştir.

Clock cycle / Instr. No.	1	2	3	4	5	6	7
1	IF	ID	EX	MEM	WB		
2		IF	ID	EX	MEM	WB	
3			IF	ID	EX	MEM	WB
4				IF	ID	EX	MEM
5					IF	ID	EX

Tablo 1- 5 adımlı pipelining [8]

2.1.3. Beklemeden Kaçınma

Süreçler birbirini beklerken de akışında çalışmaya devam edebilmelidir. Örneğin, bir izlence internetten bir dosya indirirken, uzak bilgisayardan gelecek veri bloğunu beklerken bir yandan da yeni verileri beklemeden diske veri yazabilmelidir, aynı şekilde veri yazarken de yazılma sürecini beklemeden indirmeye devam edebilmelidir.

---

[5] program

[6] interrupt-driven

[7] context-switch

[8] process

2.2.Kesme Çeşitleri

2.2.1.Yazılım Kesmeleri

Yazılım kesmeleri, komut kümesindeli özel bir komutun veya işlemcinin kendisindeki istisnai bir durumun neden olduğu kesmelerdir. Yazılım kesmesi, bir donanım kesmesinden farklı olarak yazılım tarafından çağrılır ve özellikle hata veya istisna işleme sırasında, çekirdekle iletişim kurmanın veya dizge çağrılarını çağırmanın yollarından biri olarak kabul edilir.[7]

Yazılım kesmesi, bir uygulama sonlandırıldığında veya işletim dizgesinden bir hizmet beklediğinde meydana gelebilir. Bu kesme donanım düzeyinde meydana gelen kesmeden farklı gerçekleşir. Çünkü uygulamalar doğrudan işlemciyle iletişim kuramaz, arada kernel adı verilen işletim dizgesi çekirdeğiyle iletişime geçmelidir.[7]

Çekirdek, uygulamaya SIGINT, SIGTERM gibi kesme sinyallerini işleyebilmesi için arabirim sağlamaktadır.

Tüm yazılım kesmeleri, alt rutinleri çalıştıracak bir kesme işleyicisi tarafından denetlenmektedir.[7]

2.2.2.Donanım Kesmeleri

Harici bir aygıtlardan gelen kesme sinyalleridir. Klavyeden gelen tuş vuruşları, farenin sürüklenmesi, CPU'nun okuyup işlemesi için donanım kesmeleri üretir. Bu kesme sinyalleri, işlemci herhangi bir komutu işlerken herhangi bir anda **eşzamansız** olarak gelmektedir.

Örneğin, bir disk kesmesi olduğunda işletim dizgesi şu anki sürecin çalışmasını durdurup kesme ile bekletilen disk sürecini çalıştırmak için karar alabilir. Burada alabilir söylemini kullanıyoruz çünkü bu durum aynı zamanda çalışan süreçlerle disk süreçleri arasındaki önceliklendirmesine bağlı olarak değişebilir. [1] Disk üzerinde bir blokluk veri okunduğu zaman, disk verisini bekleyen sürecin engeli kaldırılır ve çalıştırılmaya devam edilebilir.[1]

Donanım kesmeleri, örtülebilir ve örtülemez olarak iki türlü olarak karşımıza çıkar.

2.2.2.1.(MI) Örtülebilir Donanım Kesmeleri

İşlemciler gelen kesmeleri **görmezden gelmesi** için programlanabilir. Her sinyal, *Örtme Yazmacı*[9]'nda bulunan bir bite sahiptir. Bit üzerindeki değere göre görmezden gelme etkin ya da etkisizdir. İşlemcilerin çalışmasını kesebilen bu tür sinyallere örtülü kesmeler denir. [5]

2.2.2.2.(NMI) Örtülemez Donanım Kesmeleri

Örtülemez donanım kesmeleri, işlemci tarafından anında işlenmesi gereken en **yüksek öncelikli** kesme etkinlikleridir. [5] Haddinden uzun süren işlemlerde daha fazla zaman kaybının önüne geçen watchdog sinyali örtülemez donanım kesmesine örnektir.

---

[9] Mask Register

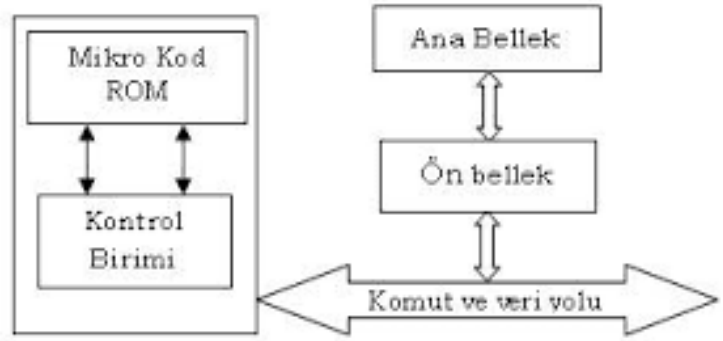
2.3.Farklı Mimarilerde Kesme

2.3.1.CISC[10] (Karmaşık Komut Kümesi Bilgisayarı)

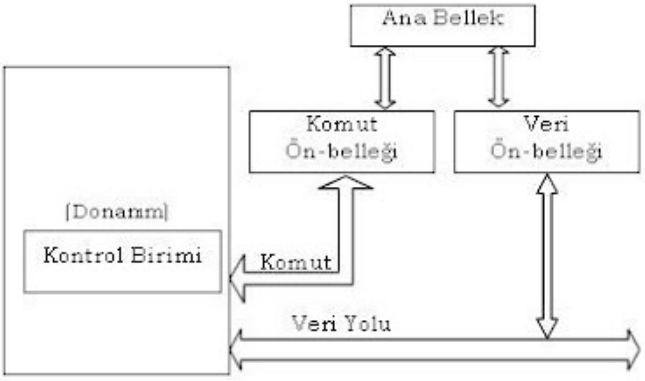
Geliştirilmiş ilk mimarilerden olan CISC, karmaşık komutları işlemek için kullanılan mimaridir. Belleğin maliyetli olduğu dönemde geliştirildiği için CPU'dan ziyade belleğe ilk sıra verilmiştir, CPU üzerindeki iş yükü nispeten daha azdır. Günümüzde değerini kaybetmesinin nedeni oldukça ilkel olması, başarım açısından bilgi-işlem dünyasının isteklerine yanıt verememesi ve bellek giderlerinin de zamanla azalmış olmasıdır.

2.3.2.RISC[11] (İndirgenmiş Komut Kümesi Bilgisayarı)

CISC mimarisindeki bazı eksiklikler giderilerek geliştirilmiş bir mimaridir. Başarım artışını sağlamak için CPU üzerine odaklanılmış ve işlemci gücü artırılmıştır. Komut sayısı azaltılmış, karmaşıklık düşürülmüş ve daha az adresleme kipi kullanılmıştır. Bu da okuma ve yazma işlemlerinde başarım artışı sağlamıştır.



Şekil 1 - CISC mimari yapısı



Şekil 2 - RISC mimari yapısı

2.3.3.EPIC[12] (Açıkça Koşut Komut Bilgisayarı)

EPIC, donanım karmaşıklığını nispeten düşük tutarken, tüm programlarda Komut Düzeyinde Koşutluğu(ILP) özelliklerinin, derleyici geliştirmelerini kolaylaştırmak için sağlandığı mimarileri ifade eder. Derleyici, spekülasyon ve tahmin gibi ILP geliştirme tekniklerini kullanarak, her döngüde paralel olarak yürütülebilecek işlemleri tanımlar ve donanıma bir yürütme izlemi iletir.[6] İşlemleri koşut biçimde yapabildiğinden doğal olarak RISC ve CISC mimarilerine göre daha hızlıdır.

---

[10] Complex Instruction Set Computer

[11] Reduced Instruction Set Computer

[12] Explicitly Parallel Instruction Computer

3. Sonuç ve Kazanımlar

Araştırma kapsamında kesme kavramı detaylandırılmış, çeşitleri incelenmiş ve farklı mimarilerdeki başarımları incelenmiştir.

Başarım açısından en iyi mimari EPIC sonrasında RISC en kötü ise CISC mimarisidir.

Yeni mimariler geliştirildikçe çoklu işleme daha hızlı yapılmaya çalışılmış ve komut işleme başarımında olduğu kadar kesme işleme başarımının da arttığı anlaşılmaktadır.

4. Kaynakça

1. Tannenbaum, A. S. (2006). *Operating Systems: Design and Implementation (Prentice-Hall Software Series) (3rd ed.)*. Prentice Hall.

2. Linfo.org, *Context Switch definition*
   22 Kasım 2021 tarihinde, www.linfo.org/context_switch.html adresinden edinildi.

3. Geeks For Geeks Blog, *Computer Organization | Different Instruction Cycles*
   23 Kasım 2021 tarihinde, https://www.geeksforgeeks.org/different-instruction-cycles adresinden edinildi

4. Tutorialspoint, *Microcontrollers - 8051 Interrupts*
   23 Kasım 2021 tarihinde, https://www.tutorialspoint.com/microprocessor/microcontrollers_8051_interrupts.htm adresinden edinildi.

5. ELPROCUS, *What is an Interrupt: Types and Its Applications*
   23 Kasım 2021 tarihinde, https://www.elprocus.com/basics-of-interrupt-types-and-its-applications adresinden edinildi.

6. August D.I., Raman A. (2011) EPIC Processors. In: Padua D. (eds) Encyclopedia of Parallel Computing. Springer, Boston, MA. https://doi.org/10.1007/978-0-387-09766-4_6

7. Technopedia, *Software Interrupt*
   24 Kasım 2021 tarihinde, https://www.techopedia.com/definition/22195/software-interrupt adresinden edinildi.

8. Tablo 1: https://en.wikipedia.org/wiki/Instruction_pipelining