



# Multicore ve Çok İşlemcili Sistemlerde Cache Coherence için Kullanılan Protokoller

## BM311 – BİLGİSAYAR MİMARİSİ

Ekim 2019

ÖDEV – 3

141180046

Can Korkut

GAZİ ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Kapak Sayfası(%4)	İçerik Düzeni(%11)			Referanslar (%15)	Atıf (%5)	İçerik(%65)	NOT
	Özet(%4)	Sonuç(%4)	Alt Başlıklar(%3)				

**İÇİNDEKİLER**

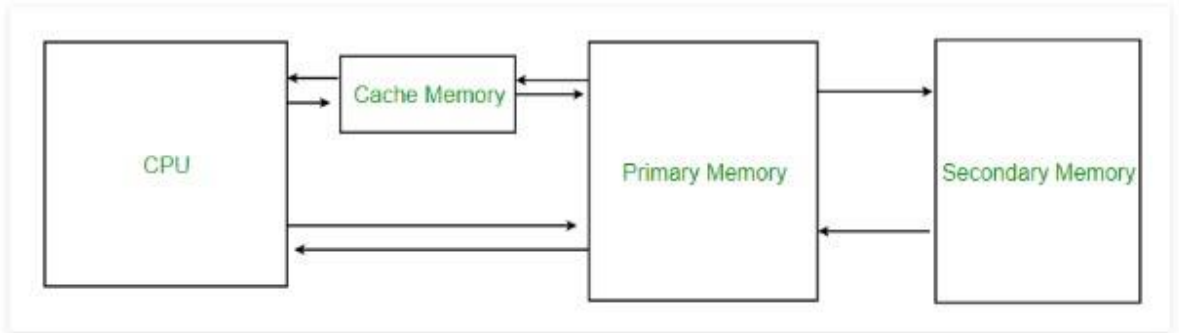
<b>1. Giriş .....</b>	<b>3</b>
<b>2. Ön Bellek Nedir? .....</b>	<b>3</b>
<b>3. Ön Bellek Tutarlılığı Nedir? .....</b>	<b>4</b>
<b>4. Kullanılan Protokoller .....</b>	<b>5</b>
<b>4.1 MSI Protokolü.....</b>	<b>5</b>
<b>4.2 MOSI Protokolü.....</b>	<b>6</b>
<b>4.3 MOESI Protokolü.....</b>	<b>6</b>
<b>5. Sonuç .....</b>	<b>8</b>
<b>6. Kaynakça .....</b>	<b>9</b>

## 1. GİRİŞ

Ön bellek işlemciye yakın konumda bulunur. Yakın konumda bulunmasının sebebi işlemcinin hızlı ulaşmasını sağlamaktır. Ödev kapsamında ön bellek ve ön bellekte kullanılan cache chorency protokolü incelenmiştir. Bu protokol ön bellek tutarlılığından kaynaklı geliştirilmiştir. Ödev kapsamında hem ön bellek hem de ön bellek tutarlılığı örnekler ile açıklanmıştır.

## 2. ÖN BELLEK NEDİR?

CPU belleği olarak da adlandırılan önbellek bir bilgisayarın mikroişlemcisinin normal rasgele erişim belleğine (RAM) erişebildiğinden daha hızlı erişebildiği yüksek hızlı statik rasgele erişim belleğidir (SRAM). Bu bellek tipik olarak doğrudan CPU yongasına bütünleşmiş edilir veya CPU ile ayrı bir veri yolu bağlantısı olan ayrı bir yonga üzerine yerleştirilir. Önbelleklerin amacı, programların çalışmasında tekrar tekrar kullanılan programların ve işlemlerin ve CPU'nun daha sonra ihtiyaç duyacağı bilgilerin depolanmasıdır. Bilgisayar işlemcisi, bu bilgileri bilgisayarın ana belleğinden almak yerine, önbellekten hızla erişebilir. Bu talimatlara hızlı erişim programın genel hızını artırır.



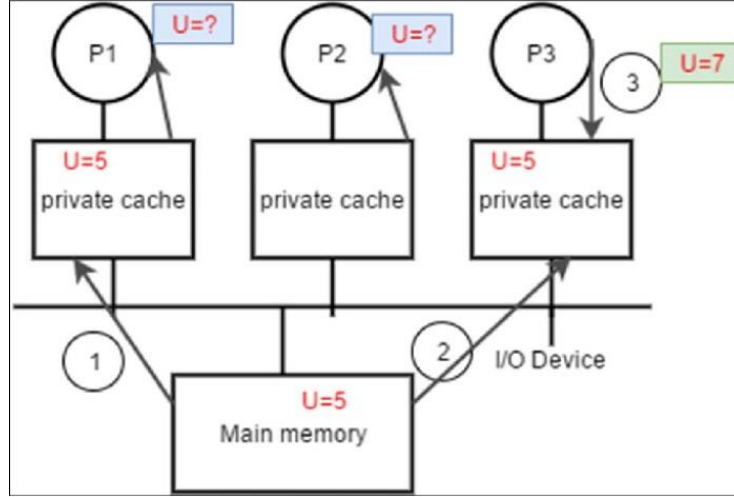
Şekil 1: CPU – Hafıza Birimleri [2]

Önbellek, Ana bellekten verilere erişmek için ortalama süreyi azaltmak için kullanılır. Önbellek, sık kullanılan ana bellek konumlarından verilerin kopyalarını saklayan daha küçük ve daha hızlı bir bellektir. Talimatları ve verileri depolayan bir CPU'da çeşitli farklı bağımsız önbellekler vardır. L1, L2 ve L3 önbellekleri, bilgisayardaki RAM'a benzer farklı bellek

havuzlarıdır. İşlemci tarafından verilere erişmek için harcanan zamanı azaltmak için yapılmıştır. Bu süreye gecikme denir. Yapıldıkları mimari yapı da oldukça farklı. Örneğin L1 önbellek, daha büyük transistörler ve daha geniş metal izler kullanılarak üretildi, yer kaplıyor ve hız için güç sağlıyor. Daha yüksek seviye önbellekleri daha sıkı paketlenir ve daha küçük transistörler kullanır.

### 3. ÖN BELLEK TUTARLILIĞI NEDİR?

Önbellek tutarlılığı, L1 ve L2 önbellekleri dağıtılmış çok çekirdekli bir sistemde ortaya çıkan bir problemdir. Her çekirdeğin kendi L1 ve L2 önbellekleri vardır ve verilerin en güncel sürümüne sahip olmak için her zaman birbirleriyle senkronize olmaları gerekir. Önbellek Tutarlılığı Sorunu, işlemcilerden biri birden fazla önbellek arasında paylaşılan yerel veri kopyasını güncellediğinde birden fazla yerel önbelleği senkronize tutma zorluğudur. Aynı verilerin birden fazla kopyasının aynı anda farklı önbellekte bulunduğu bir senaryo düşünün ve işlemcilerin kendi kopyalarını serbestçe güncellemelerine izin verilirse, tutarsız bir bellek görüntüsü ortaya çıkabilir.



Şekil 2: Ön Bellek Tutarsızlık Problemi [3]

Paylaşılan hafıza çok işlemcileri, artan işlem gücü ve paylaşılan hafıza modelinin programlanabilirliğini sunar. Ancak, işlemciler arasında hafıza paylaşımı, hafıza erişimini geciktiren çekişmelere yol açmaktadır. Her işlemci için bir önbellek eklemek, ortalama erişim süresini azaltır, ancak önbellekteki kopyalar arasında tutarsızlık olasılığı yaratır. Önbellek tutarlılık

sorunu, aynı bellek konumunun tüm önbelleğe alınmış kopyalarını aynı tutmaktır. Bu tez, önbellek tutarlılığı sorununa olası çözümler araştırıyor ve çekici bir alternatif olarak önbellek tutarlılığı protokollerini - tamamen donanımda uygulanan çözümleri - çekici bir alternatif olarak tanımlamaktadır. Paylaşılan veri yolu sistemleri için yöntemlerin ilginç bir özel durum olduğu gösterilmiştir. Önceden önerilen paylaşılmış veri yolu protokolleri tek tip terminoloji kullanılarak tanımlanmıştır ve bunların iki kategoriye ayrıldığı gösterilmiştir: geçersiz kılma ve dağıtılmış yazma. Geçersiz kılma protokollerinde, diğer tüm önbelleğe alınmış kopyalar herhangi bir kopya değiştirilmeden önce geçersiz kılınmalıdır; Dağıtılmış yazma protokollerinde, paylaşılan bir blok her değiştirildiğinde tüm kopyaların güncellenmesi gerekir. Her kategoride, simülasyon sonuçlarına dayanarak önceki şemalardan daha iyi performans gösteren yeni bir protokol sunulmaktadır. Simülasyon modeli ve parametreleri ayrıntılı olarak açıklanmıştır. Genel bağlantı ağları için önceki protokollerin kusur içerdiği ve uygulanmasının maliyetli olduğu gösterilmiştir. Bir çapraz çubuk geçişi kullanılarak yapılan simülasyon sonuçlarında gösterildiği gibi, yüksek bir performans seviyesini korurken, daha düşük uygulama maliyeti ve genişletilebilirliği sunan yeni bir protokol sınıfı sunulmuştur. Tüm yeni protokollerin doğru olduğu kanıtlandı; ispatlardan biri dâhil edilmiştir. Önbellek tutarlılığının önceden tanımları yetersizdir ve yeni bir tanım sunulmuştur. Tutarlılık, aynı zamanda tanımlanmış olan diğer tutarlılık seviyeleri ile karşılaştırılır ve karşılaştırılır. Paylaşılan veri yolu protokollerinin tutarlılığının, veri yolu olmayan protokollerinkinden doğal olarak daha güçlü olduğu gösterilmiştir. Türünün ilk protokolü, her kümedeki veri yolu tabanlı bir protokol ve genel bir protokol kullanarak, büyük bir hiyerarşik çoklu işlemciye sunulur.

## 4. KULLANILAN PROTOKOLLER

### **4.1 MSI Protokolü:**

Bu çok işlemcili bir sistemde kullanılan temel bir önbellek tutarlılık protokolüdür. Protokol adının harfleri, bir önbelleğin olabileceği olası durumları tanımlar. Bu nedenle, MSI için her blok aşağıdaki olası durumlardan birine sahip olabilir:

- Modified -  
Blok önbellekte değiştirildi, yani önbellekteki veriler destek deposuyla (bellek) tutarsız. Bu nedenle, “M” durumunda bir bloğu olan bir önbellek, tahliye edildiğinde bloğu destek deposuna yazma sorumluluğuna sahiptir.
- Shared –

Bu blok değiştirilmedi ve en az bir önbellekte mevcut. Önbellek, verileri destek deposuna yazmadan çıkarabilir.

- Invalid -  
Bu blok geçersiz ve eğer bu önbellekte depolanacaksa bellekten veya başka bir önbellekten alınmalıdır.

#### **4.2 MOSI Protokolü:**

Bu protokol MSI protokolünün bir uzantısıdır. MSI protokolünde aşağıdaki durumu ekler:

- Owned -  
Mevcut işlemcinin bu bloğa sahip olduğunu ve blok için diğer işlemcilerden gelen istekleri karşılayacağını belirtir. MESI Protokolü -
- En yaygın kullanılan önbellek tutarlılık protokolüdür. Her önbellek satırı aşağıdaki durumlardan biriyle işaretlenmiştir:
- Modified -  
Bu, önbellek satırının yalnızca geçerli önbellekte bulunduğunu ve kirli olduğunu gösterir, yani değeri ana bellekten farklıdır. Önbellek, verileri geçersiz ana bellek durumlarını okumaya izin vermeden önce, gelecekte ana belleğe geri yazmak için gereklidir.
- Exclusive -  
Bu, önbellek satırının yalnızca geçerli önbellekte bulunduğunu ve temiz olduğunu, yani değerinin ana bellek değeriyle aynı olduğunu gösterir.
- Shared -  
Bu önbellek hattının makinenin diğer önbelleklerinde saklanabileceğini gösterir.
- Geçersiz -  
Bu önbellek satırının geçersiz olduğunu gösterir.

#### **4.3 MOESI Protokolü:**

Bu, diğer protokollerde yaygın olarak kullanılan tüm olası durumları kapsayan tam bir önbellek tutarlılık protokolüdür. Her önbellek satırı aşağıdaki durumlardan birindedir: □

Modified-

Bu durumdaki önbellek satırı, ana bellekteki kopya yanlıştır ve başka hiçbir işlemcinin bir kopyası bulunmazken, verilerin en son ve doğru kopyasını tutar.

- Owned-  
Bu durumdaki bir önbellek satırı, verilerin en son ve doğru kopyasını tutar. Diğer işlemcilerin en son, doğru verilerin bir kopyasını tutabileceği ve paylaşılan durumun aksine paylaşılan duruma benzer, ancak ana bellekteki kopya yanlış olabilir. Yalnızca

bir işlemci verileri ait durumda tutabilirken, diğer tüm işlemciler verileri paylaşılan durumda tutmalıdır.

- Exclusive -  
Bu durumdaki bir önbellek satırı, verilerin en son ve doğru kopyasını tutar. Ana hafıza kopyası ayrıca en yeni, doğru veri kopyası iken, hiçbir veri kopyasına sahip değildir.
- Shared -  
Bu durumdaki bir önbellek satırı, verilerin en son ve doğru kopyasını tutar. Sistemdeki diğer işlemciler de paylaşılan durumda veri kopyalarını alabilir. Ana bellek kopyası, başka hiçbir işlemci ait durumda değilse, verilerin en son ve doğru kopyasıdır.
- Invalid -  
Bu durumdaki bir önbellek satırı geçerli bir veri kopyasını içermiyor. Verilerin geçerli kopyaları ana bellekte veya başka bir işlemci önbelleğinde olabilir.

## 5. SONUÇ

Bu çalışmada öncelikle ön bellek kavramı incelenmiştir incelenme sonucu ön belleğin mikroişlemciye fiziksel olarak yakın bir konumda bulunan, RAM alanından daha hızlı erişime sahip bir bellek türü olduğu saptanmıştır. Ön bellek işlemciye yakın konumda bulunur. Yakın konumda bulunmasının sebebi işlemcinin hızlı ulaşmasını sağlamaktır. Çalışma kapsamında ön bellek ve ön bellekte kullanılan cache chorency protokolü incelenmiştir. Bu protokol ön bellek tutarlılığından kaynaklı geliştirilmiştir. Ödev kapsamında hem ön bellek hem de ön bellek tutarlılığı örnekler ile açıklanmıştır. Ön bellek üzerinde aynı anda ve aynı veri üzerinde işlem yapmasından dolayı bir tutarsızlık oluşabilir. Bu problem ön bellek tutarsızlığı olarak adlandırılır. Problem çözümü için geliştirilen yöntemlerden kısaca bahsedilmiştir. Çözüm sürecinde kullanılan protokoller maddeler halinde açıklanmıştır.



## 6. KAYNAKÇA

- [1] Yazılı Basın (Kitap): Handy, Jim. The Cache Memory Book. Academic Press, Inc., 1998. ISBN 0-12-322980-4
- [2] "Cache Memory in Computer Organization - GeeksforGeeks", *GeeksforGeeks*, 2019. [Online]. Available: <https://www.geeksforgeeks.org/cache-memory-in-computerorganization/>. [Accessed: 04- Nov- 2019].
- [3] Y. Tian and G. Chen, "A Review of Researches on Cache Coherence Protocols for Multi-Core Processor", *Advanced Materials Research*, vol. 933, pp. 740-743, 2014. Available: 10.4028/www.scientific.net/amr.933.740.
- [4] Cache Coherence Kavramı - mehmetduran.com", *Mehmetduran.com*, 2019. [Online]. Available: <http://www.mehmetduran.com/Blog/Makale.html/Cache-CoherenceKavrami/194>. [Accessed: 04- Nov- 2019].