

BM-311 Bilgisayar Mimarisi

Hazırlayan: M.Ali Akcayol
Gazi Üniversitesi
Bilgisayar Mühendisliği Bölümü

Konular

- Çok işlemcili sistemler
- Simetrik çok işlemciler
- Önbellek tutarlılığı
- MESI protokolü
- Multithreading
- Cluster yapıları

Çok işlemcili sistemler

- Geleneksel olarak **bilgisayar sıralı işlem yapan bir makine olarak düşünülür.**
- Bir çok **programlama dili algoritmaların sıralı komutlar halinde yazılmasını gerektirir.**
- **İşlemciler**, programlardaki **makine komutlarını sıralı çalıştırır.**
- Her **makine komutu**, **sıralı alt işlemler** kullanılarak (fetch instruction, decode instruction, fetch operands, perform operations, store results) çalıştırılır.
- **Mikroişlem düzeyinde birden fazla kontrol sinyali aynı anda oluşturulur.**
- **Superscalar makinelerde** birden fazla işlem birimi vardır ve **bir programın farklı komutları aynı anda çalıştırılır.**

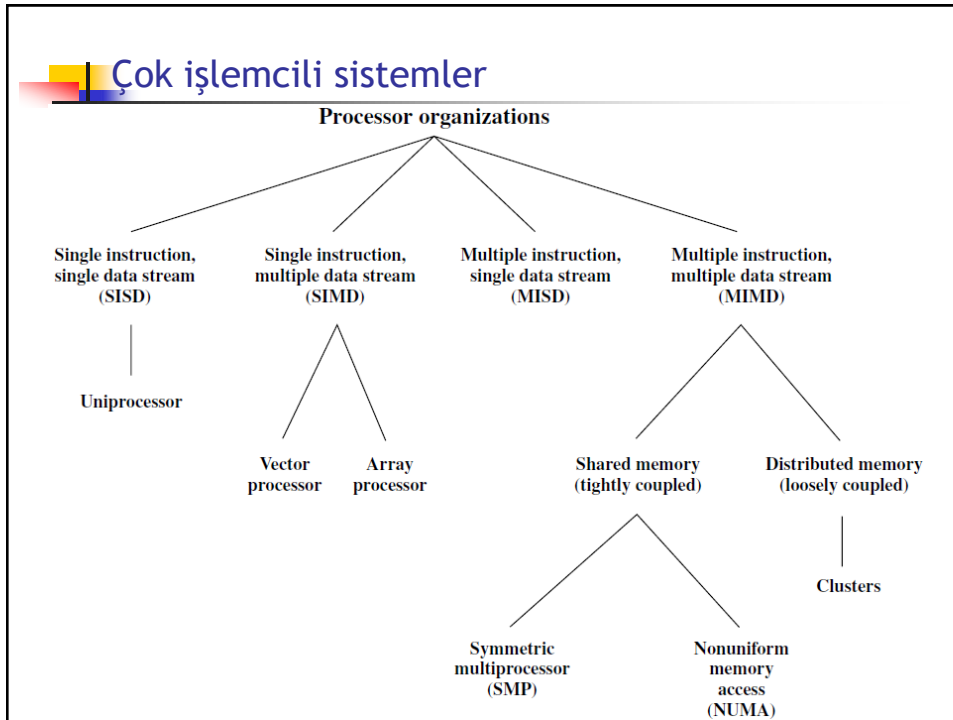
Çok işlemcili sistemler

- Bilgisayar sistemleri dört kategoriye ayrılır:
 - Single instruction, single data (SISD) stream: **Bir işlemci bir bellekte bulunan veri üzerinde işlem** yapmak için **bir komutu çalıştırır** (Tek işlemcili sistemler).
 - Single instruction, multiple data (SIMD) stream: **Bir makine komutu çok sayıda eş zamanlı işlem elemanını** kontrol eder. **Her işlem elemanı kendine ait hafızaya sahiptir** (Vektör ve dizi işlemcileri).
 - Multiple instruction, single data (MISD) stream: **Bir grup işlemciye sıralı veri gönderilir.** Her işlemci farklı komut dizisini çalıştırır (Ticari olarak gerçekleştirilmedi.).
 - Multiple instruction, multiple data (MIMD) stream: **Bir grup işlemci farklı komut dizisini farklı veri kümesi üzerinde eş zamanlı çalıştırır** (Simetrik çok işlemciler, cluster'lar, nonuniform memory access (NUMA) sistemler).

Çok işlemcili sistemler

- **MIMD organizasyonunda işlemciler genel amaçlıdır ve** yapılacak iş için gerekli tüm komutları çalıştırır.
- MIMD iki gruba ayrılır:
 - **Shared memory:**
 - Tüm işlemciler aynı belleği paylaşır.
 - Tüm program ve veriler aynı bellektedir.
 - **İşlemciler paylaşılan bellek üzerinden iletişim yapar.**
 - **Simetrik çok işlemciler** bu gruptadır.
 - **Distributed memory:**
 - Belleğin farklı bölümlerine farklı işlemcilerin erişim süreleri farklıdır.
 - **Cluster sistemleri** bu gruptadır.
- **Bağımsız tek işlemciler** veya **simetrik çok işlemcili sistemler** birbirine bağlanarak **cluster yapısı oluşturulur.**
- Bilgisayarlar arasındaki bağlantı doğrudan veya ağ üzerinden yapılabilir.

Çok işlemcili sistemler



Çok işlemcili sistemler

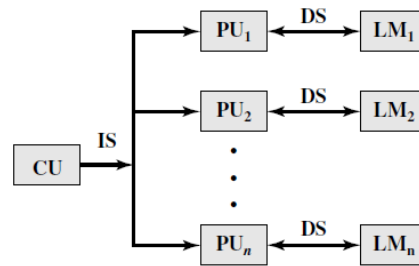
- **SISD** yapısında, bir **kontrol birimi (CU)** tarafından **komut dizisi (IS)** bir **işlem birimine (PU)** sağlanır.
- **İşlem birimi bir bellek birimi (MU)** üzerinde işlem yapar.
- **SIMD** yapısında, bir **kontrol birimi** tarafından **komut dizisi birden çok işlem birimine sağlanır.**
- **Her işlem birimi** kendine ait **lokal belleğe (LM)** sahiptir.



(a) SISD

CU = Control unit
IS = Instruction stream
PU = Processing unit
DS = Data stream
MU = Memory unit
LM = Local memory

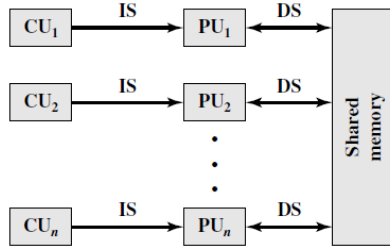
SISD = Single instruction,
= single data stream
SIMD = Single instruction,
multiple data stream
MIMD = Multiple instruction,
multiple data stream



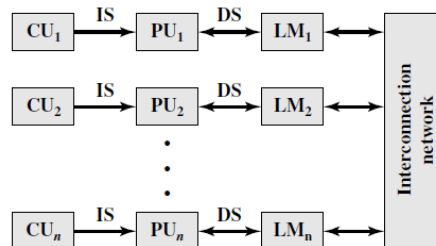
(b) SIMD (with distributed memory)

Çok işlemcili sistemler

- **MIMD** yapısında, **birden fazla kontrol birimi** tarafından **farklı komut dizileri birden fazla işlem birimine** sağlanır.
- Paylaşılmış belleğe veya dağıtık belleğe sahip olabilir.



(c) MIMD (with shared memory)



(d) MIMD (with distributed memory)

CU = Control unit
IS = Instruction stream
PU = Processing unit
DS = Data stream
MU = Memory unit
LM = Local memory

SISD = Single instruction,
= single data stream
SIMD = Single instruction,
multiple data stream
MIMD = Multiple instruction,
multiple data stream

Konular

- Çok işlemcili sistemler
- **Simetrik çok işlemciler**
- Önbellek tutarlılığı
- MESI protokolü
- Multithreading
- Cluster yapıları

Simetrik çok işlemciler

- **Son yıllara kadar** tek kullanıcıli kişisel bilgisayarlar ve iş istasyonların tümü genel amaçlı **tek mikroişlemciye** sahipti.
- Performans gereksinimlerini karşılamak üzere **SMP (Symmetric Multiprocessors)** mimarisi geliştirilmiştir.
- **SMP mimarisine sahip bir bilgisayar:**
 - **İki veya daha fazla benzer işlemciye sahiptir.**
 - İşlemciler, **aynı belleği paylaşır, birbirine bus ile bağlıdır** ve **bellek erişim süreleri yaklaşık aynıdır.**
 - **Tüm işlemciler I/O cihazlarını** aynı iletim kanalları veya farklı iletim kanalları üzerinden **paylaşır.**
 - Tüm işlemciler **aynı işlevleri** yerine getirir (**simetrik**).
 - **Sistem bir işletim sistemi tarafından kontrol edilir.**

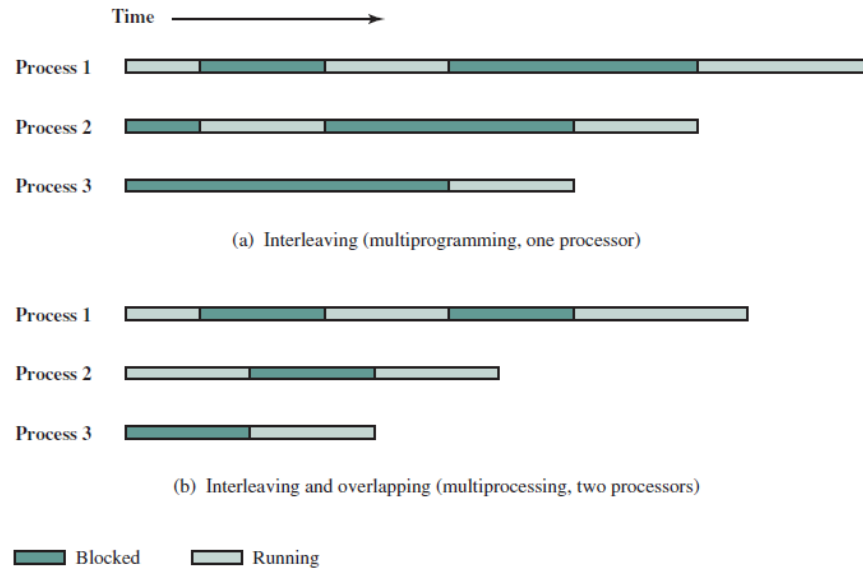
Simetrik çok işlemciler

- SMP mimarisindeki **işletim sistemi, process'lerin veya thread'lerin çalışmasını planlar.**
- **SMP mimarisi** tek işlemcili mimariye göre, **performance, availability, incremental growth** ve **scaling** avantajlarına sahiptir.
- SMP mimarisinde **paralel çalışmanın etkisi işletim sisteminin sağladığı araçlara bağlıdır** (işlemcilerin senkronizasyonu, process ve thread'lerin planlanması).

Performance

- İşlerin bazı kısımları **aynı anda çalıştırılabilir.**
- **Çok işlemcili bir sistem** tek işlemcili sisteme göre **önemli oranda performans artışı sağlar.**

Simetrik çok işlemciler



Simetrik çok işlemciler

Availability

- SMP'de tüm işlemciler aynı işleve sahip olduğundan, **bir işlemci çalışmazsa makine** diğer işlemcilerle daha düşük performansla **çalışmasına devam eder.**

Incremental growth

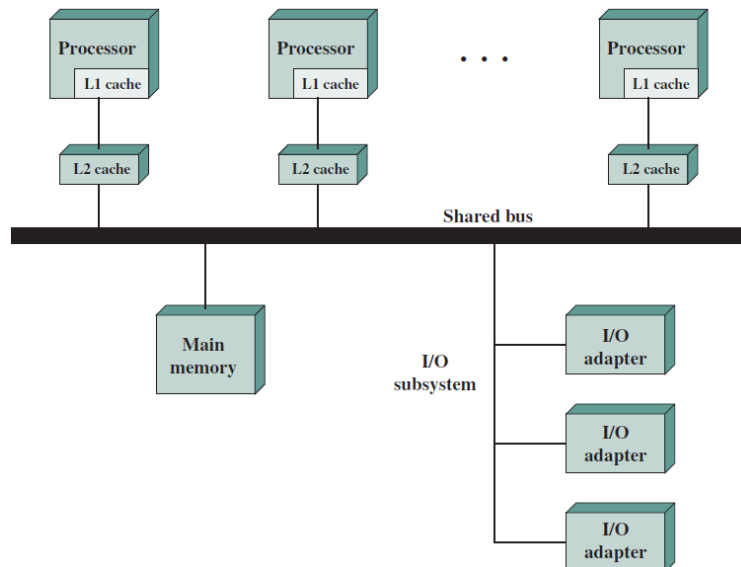
- Sistem performansı işlemci eklenerek artırılabilir.

Scaling

- Üreticiler **işlemci sayısına bağlı** olarak **farklı performans ve farklı fiyata sahip ürün** aralığı **sağlayabilir.**

Simetrik çok işlemciler

- Çok işlemcili sistemler zaman paylaşımli bus kullanır.



Konular

- Çok işlemcili sistemler
- Simetrik çok işlemciler
- **Önbellek tutarlılığı**
- MESI protokolü
- Multithreading
- Cluster yapıları

Önbellek tutarlılığı

- Modern çok işlemcili sistemlerde, **önbelleğin bir veya iki seviyesi** her işlemci için **ayrı oluşturulur**.
- Farklı işlemcilerde **ayrı önbellek bulunması** önbellek tutarlılığı (**cache coherence**) açısından **problem oluşturur**.
- Aynı verinin birden fazla kopyası üzerinde farklı önbelleklerde işlem yapılabilir.
- **İşlemciler aynı verinin farklı önbelleklerdeki kopyalarını aynı anda değiştirirse tutarsızlık ortaya çıkar**.
- Veri üzerindeki değişikliklerin tutarlılığı bozmayacak şekilde yapılması gerekir.

Önbellek tutarlılığı

- İki write policy yaygın kullanılır: **write back** ve **write through**.
- **Write back**
 - Değişiklikler önbellekte yapılır.
 - **Önbellekten atılacağı zaman ana hafızada güncelleme yapılır.**
 - Ana hafızadaki veri yazılana kadar güncel değildir.
- **Write through**
 - **Değişiklikler önbellekle aynı anda hafızada da yapılır.**
 - Ana hafızaki veri sürekli günceldir.
- **Write back kullanılıyorsa** ve iki önbellek aynı veriye sahipse, birisinde yapılacak değişiklikte **tutarsızlık oluşur**.
- Önbellek tutarlılığı **yazılım** ve **donanım** çözümleriyle sağlanır.

Önbellek tutarlılığı

Yazılım çözümleri

- **Önbellek tutarlılığını derleyici veya işletim sistemi sağlar.**
- **Derleyici** tabanlı çözümlerde, **kod analiz edilir** ve önbellek için **güvenilir olmayan veriler işaretlenir**.
- **İşletim sistemi** veya **donanım** önbelleklenemez olarak işaretlenen **bu verileri önbelleğe almaz**.
- Basit bir yaklaşımla **paylaşılan tüm veriler önbelleklenemez yapılır**.

Önbellek tutarlılığı

Donanım çözümleri

- Donanımsal çözümlerde önbellek tutarlılığı için **protokol geliştirilir.**
- **Run time'da** potansiyel **tutarsızlıklar dinamik olarak algılanır.**
- **Donanımsal çözümlerde** problem ortaya çıktığı anda giderildiğinden, **yazılımsal çözümlere göre daha iyi performans sağlar.**
- Donanımsal çözümler iki gruba ayrılır:
 - **Directory protokolleri**
 - **Snoopy protokolleri**

Önbellek tutarlılığı

Donanım çözümleri – directory protokolleri

- Directory protokolleri, **verilerin kopyasının nerede olduğuna dair bilgi toplar ve bellekte saklar.**
- Genellikle ana hafızaya bütünleşik **merkezi bir denetleyici**, önbelleklerden **gelen istekleri izler.**
- Herhangi bir önbellekteki bir **veri üzerinde yapılacak lokal işlem, merkezi denetleyiciye bildirilir.**
- Merkezi denetleyici verinin hangi CPU'da olduğunu saklar.
- Herhangi bir önbellek **veriye yazma** yapacağı zaman **merkezi denetleyiciden izin ister**, diğer önbellekler **invalidate** yapar.
- **Başka önbellek aynı veriyi istediğinde**, merkezi denetleyici güncel halini tutan önbellekten **write back yapmasını ister.**
- Directory protokollerde, **merkezi bileşende yığılma** ve haberleşmeden dolayı **overhead** oluşur.

Önbellek tutarlılığı

Donanım çözümleri – snoopy protokolleri

- Önbellek tutarlılığı **dağıtık olarak** işlemcilerde bulunan **önbellek denetleyiciler tarafından yapılır.**
- Her önbellek **diğer önbelleklerle paylaştığı verileri bilmek zorundadır.**
- **Paylaşılan bir veride** bir önbellek tarafından **güncelleme yapıldığında, tüm önbelleklere broadcast bildirim yapılır.**
- Her önbellek denetleyici **broadcast mesajları izler.**
- Snoopy protokoller **bus trafiğini artırır.**

Konular

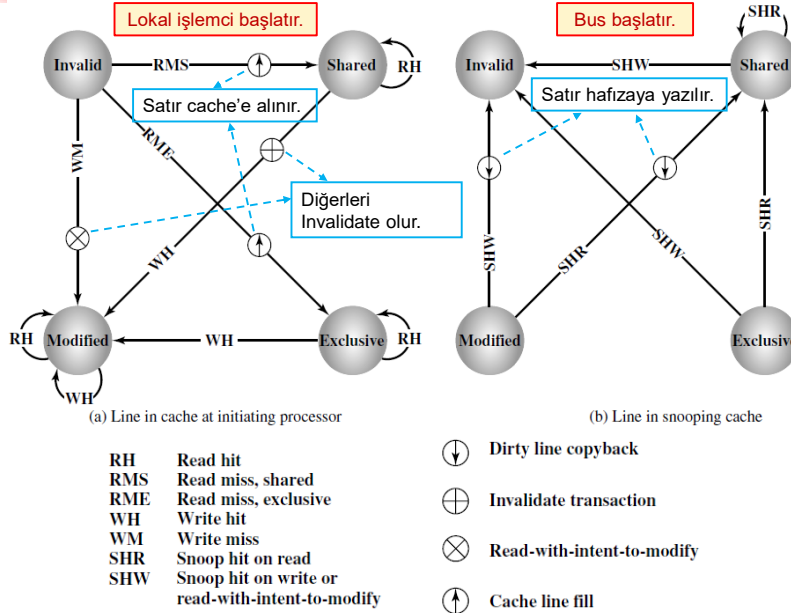
- Çok işlemcili sistemler
- Simetrik çok işlemciler
- Önbellek tutarlılığı
- **MESI protokolü**
- Multithreading
- Cluster yapıları

MESI protokolü

- **MESI (Modified Exclusive Shared Invalid)** protokolü **önbellekteki verinin durumunu iki bite tutar.**
- Önbellekteki **her satır dört durumdan birisinde olabilir:**
 - **Modified:** Veri **değişmiştir** (ana hafızadaki farklıdır) ve **sadece bu önbellekte güncel hali vardır.**
 - **Exclusive:** Veri **değişmemiştir** (ana hafızadaki aynıdır) ve **başka önbellekte yoktur.**
 - **Shared:** Veri **değişmemiştir** (ana hafızadaki aynıdır) ve **başka önbellekte olabilir.**
 - **Invalid:** Önbellekteki **veri geçerli değildir.**

	M Modified	E Exclusive	S Shared	I Invalid
This cache line valid?	Yes	Yes	Yes	No
The memory copy is ...	out of date	valid	valid	—
Copies exist in other caches?	No	No	Maybe	Maybe
A write to this line ...	does not go to bus	does not go to bus	goes to bus and updates cache	goes directly to bus

MESI protokolü



MESI protokolü

- **MESI önbellekteki verinin durumunu, ait olduğu işlemci veya bus'tan aldığı girişle değiştirir.**
- **Read Miss**
 - Önbellekte **read miss olursa, işlemci hafızadan okuma yapar.**
 - İşlemci bus'tan **diğer işlemci önbelleklerine mesaj gönderir.**
 - 1) **Başka bir önbellekte güncel kopyası varsa, bu kopya alınır. Gönderendeki durumu Exclusive ise Shared yapılır, alan önbellekteki durumu Invalid ise Shared yapılır.**
 - 2) **Bir veya daha fazla cache'te güncel kopya varsa ve Shared durumunda ise, alan önbellekte Invalid ise Shared yapılır.**
 - 3) **Eğer başka bir önbellekte var ve Modified ise, gönderendeki durumu Shared yapılır, alandaki durumu Shared yapılır.** Ana hafızadaki kopya güncellenir.
 - 4) **Başka önbellekte yoksa ana hafızadan alınır ve Invalid durumu Exclusive yapılır.**

MESI protokolü

- **Write Miss**
 - Önbellekte **write miss olursa, işlemci ana hafızadan veriyi okur.**
 - Diğer belleklere bus üzerinden **read-with-intent-to-modify sinyali gönderir.**
 - Satır yüklenir yüklenmez **durumu Modified yapılır.**
 - Eğer **diğer önbelleklerden birisinde Modified olarak varsa, bus üzerinden hafızada güncelleme yapılır ve durumu Invalid yapılır.**
 - Eğer **diğer önbelleklerde Modified durumunda yoksa, Shared durumunda varsa** hepsi **Invalid durumuna geçer** ve talep eden işlemci ana hafızadan alır.

MESI protokolü

- **Read Hit**
 - **Önbellekte read hit olursa**, işlemci veriyi okur ve **mevcut durumu (Modified, Shared, Exclusive) değişmez.**
- **Write Hit**
 - **Lokal önbellekte bir satırda write hit olursa**, mevcut duruma göre işlem yapılır:
 - 1) **Shared: İşlemci değişiklik yapmadan önce Exclusive erişime sahip olmalıdır.**
Tüm işlemcilere bus'tan sinyal gönderilir ve **Shared durumunda olanlar Invalid yapılır.**
İşlemci **değişikliği yapar** ve **Shared durumundan Modified durumuna alır.**
 - 2) **Exclusive: İşlemci değişikliği yapar** ve **Exclusive durumundan Modified** durumuna alır.
 - 3) **Modified: İşlemci değişikliği yapar** ve satır için **durum değişikliği yapmaz.**

Konular

- Çok işlemcili sistemler
- Simetrik çok işlemciler
- Önbellek tutarlılığı
- MESI protokolü
- **Multithreading**
- Cluster yapıları

Multithreading

- İşlemcilerin en önemli **performans ölçütü, komutları çalıştırma hızıdır.**

$$\text{MIPS rate} = f \times \text{IPC}$$

- f **clock frekans (MHz)** ve **IPC instruction per cycle**'dir.
- **MIPS** saniyedeki komut sayısı (Million Instruction Per Second).
- **IPC, pipelining** ve **superscalar** ile **artırılmaya çalışılır.**
- Pipeline performansı **komutların sırası değiştirilerek** artırılır.
- **Karmaşıklık** ve **enerji** tüketiminden dolayı **belirli bir limite kadar iyileştirme yapılabilir.**
- **Multithreading**, devre karmaşıklığını ve enerji tüketimini artırmadan **instruction level parallelism'i artırır.**
- Instruction stream çok sayıda **küçük stream'e bölünür (threads)** ve paralel çalıştırılır.

Multithreading

- **Process:** Bir bilgisayarda çalışan programın örneğidir.
 - **Resource ownership:** Bir process, bir virtual adres aralığında **program, veri, stack** ve kendi **özelliklerini tutar.**
 - **Scheduling/execution:** Process, işletim sistemi tarafından **running durumuna** geçirilir ve **işlemciye atanır.**
- **Process switch:** İşlemcinin **bir process'ten başka bir process'e geçmesini** ifade eder. Önceki process'in **tüm verileri saklanır.**
- **Thread:** Bir process içerisinde **ayrı çalıştırılabilen kısımdır.** Bir thread sıralı çalıştırılır ve kesilip başka bir thread'e geçilebilir.
- **Thread switch:** İşlemcinin **bir thread'den başka bir thread'e geçmesini ifade eder.** Thread switch, process switch'e göre daha düşük maliyete sahiptir.

Multithreading

- Multithreaded işlemcilerde **her thread için ayrı program counter** olması gereklidir.
- Bir **process içindeki thread'ler aynı kaynağı paylaşır.**
- **Thread switch süresi process switch süresinden daha düşüktür.**
- **User-level thread'ler**, uygulama programları tarafından kullanılır.
- **Kernel-level thread'ler**, işletim sistemi tarafından kullanılır.
- **Implicit multithreading**, **sıralı tek programdan** birden fazla thread' in eşzamanlı çalıştırılmasını ifade eder.
- **Explicit multithreading**, **farklı programlardan** birden fazla thread' in eşzamanlı çalıştırılmasını ifade eder.

Multithreading

Explicit multithreading

- Multithreaded işlemci, **thread'leri eşzamanlı** (concurrently) çalıştırmak için **tek program counter** kullanabilir.
- Multithreading için **dört yaklaşım vardır:**
 - **Interleaved** (fine-grained) multithreading
 - **Blocked** (coarse-grained) multithreading
 - **Simultaneous** multithreading
 - **Chip multiprocessing**

Multithreading

Explicit multithreading

■ Interleaved (fine-grained) multithreading

- CPU **iki veya daha çok thread arasında** her clock cycle'da geçiş yapar.
- Thread **veri bağımlılığı** veya **hafıza gecikmesi** nedeniyle **bloklanırsa** başka thread'e geçilir.

■ Blocked (coarse-grained) multithreading

- Bir thread'deki komutlar **cache miss** veya **delay** oluncaya kadar **sıralı çalıştırılır**. **Cache miss** veya **delay** olunca **başka thread'e geçilir**.

■ Simultaneous multithreading

- **Birden fazla thread'deki komutlar** superscalar işlemciye **aynı anda gönderilir**.

■ Chip multiprocessing

- Bir chip içerisindeki **birden fazla işlemci (multicore)** farklı thread'leri çalıştırır.

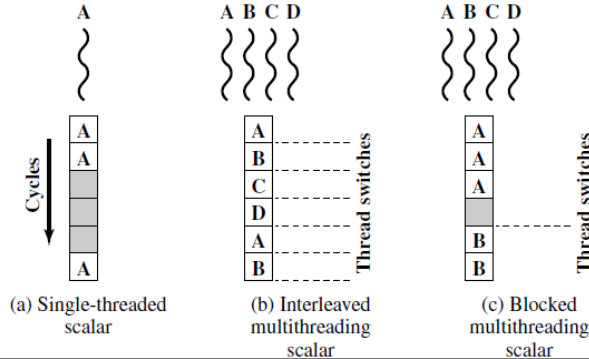
■ İlk iki yaklaşımda, **farklı thread'lerdeki komutlar aynı anda çalışmaz, son ikisinde aynı anda çalışır.**

Multithreading

■ **Single-threaded scalar** çalışmada, **bir pipeline vardır ve multithreading yoktur.**

■ **Interleaved multithreaded scalar** çalışmada, her clock cycle'da **bir thread'den diğerine geçiş yapılır.**

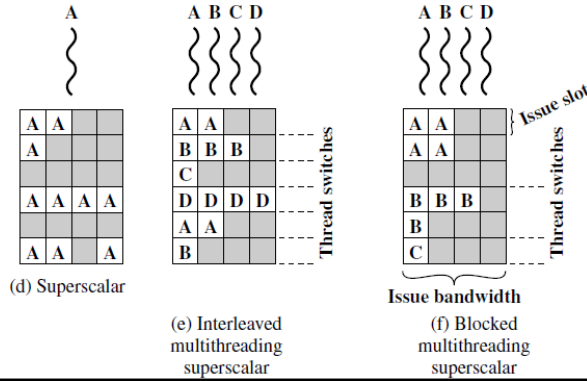
■ **Blocked multithreaded scalar** çalışmada, **delay** veya **cache miss olana kadar thread çalıştırılır**, olunca diğerine geçilir.



Multithreading

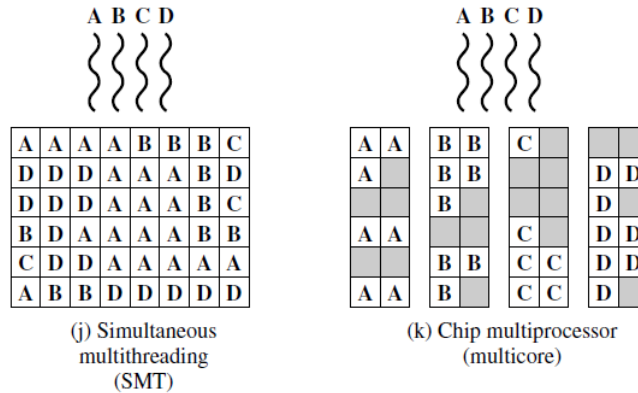
- **Temel superscalar** çalışmada, **multithreading yoktur** ve bazen **maksimum sayıda** (işlem birimi sayısı) **komut alınamayabilir**, bazen **hiç komut alınamayabilir**.
- **Interleaved multithreading superscalar** çalışmada, bir thread'den olabildiği kadar çok komut alınır. **Her clock cycle'da thread'ler arasında geçiş yapılır.**

- **Blocked multithreaded superscalar** çalışmada, bir cycle'da **bir thread'den komutlar alınır** ve blocked multithreading kullanılır.



Multithreading

- **Simultaneous multithreading** çalışmada, **farklı thread'lerden** olabildiği kadar çok komut aynı anda çalıştırılır.
- **Chip multiprocessor (multicore)** çalışmada, her işlemci birden fazla komut alabilen superscalar yapıya sahiptir. **Her işlemci bir thread'e atanmıştır** ve iki komut aynı anda alabilir.



Konular

- Çok işlemcili sistemler
- Simetrik çok işlemciler
- Önbellek tutarlılığı
- MESI protokolü
- Multithreading
- Cluster yapıları

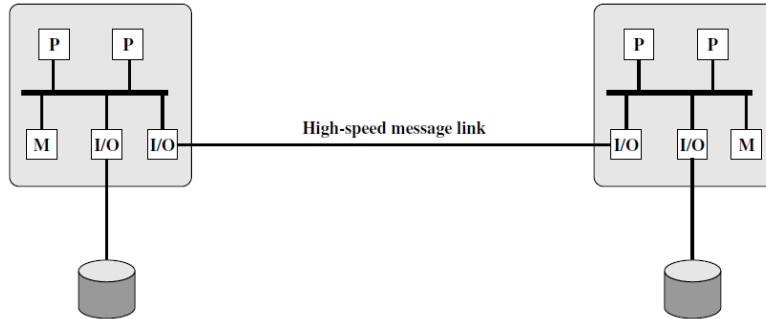
Cluster yapıları

- **Cluster**, birbirine bağlı bir grup bilgisayarın sahip olduğu **kaynakları** birleştirilmiş **bir makine gibi gösterir**.
- Cluster içinde her bilgisayar **düğüm (node)** olarak adlandırılır.
- Cluster yapısının faydaları:
 - **Mutlak ölçeklenebilirlik:** Cluster'a **yeni makineler kolaylıkla eklenebilir** (Onlarca, yüzlerce hatta binlerce makine olabilir).
 - **Artırımlı ölçeklenebilirlik:** Bir cluster'a küçük artırımlarla **yeni sistemler eklenebilir**. Kullanıcı ihtiyaçlarını sistemde büyük değişiklik yapmadan genişletebilir.
 - **Yüksek bulunurluk (high availability):** Cluster'daki **bir düğümdeki hata** sistemin **servis dışı kalmasına neden olmaz**.
 - **Fiyat/performans üstünlüğü:** **Bir grup basit makine** ile büyük bir makineyle **eşdeğer** veya **daha güçlü bir cluster oluşturulabilir**. Bir tane **büyük makineye göre** daha **düşük maliyet ortaya çıkar**.

Cluster yapıları

Cluster konfigürasyonları

- En basit cluster yapısında, **cluster içindeki makineler aynı diske paylaşarak erişir.**
- Şekildeki iki düğüme sahip bir cluster **yüksek hızlı bağlantı ile düğümler arasında mesaj aktarmaktadır.**
- Cluster'daki **her bilgisayar çok işlemcildir.**

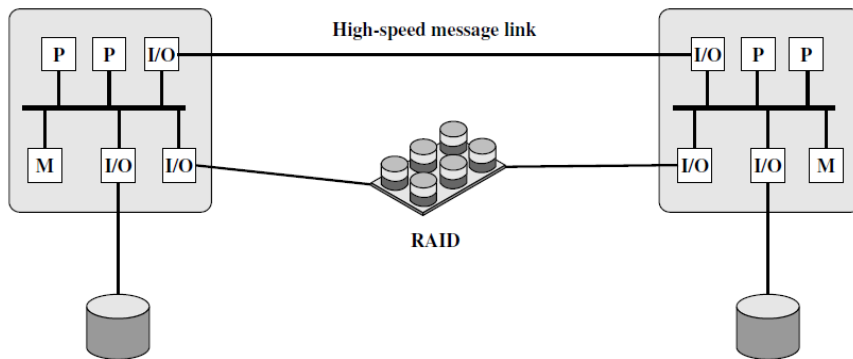


(a) Standby server with no shared disk

Cluster yapıları

Cluster konfigürasyonları

- Bir cluster'daki **makineler aynı diske paylaşarak erişebilir.**
- Düğümler arasında **yüksek hızlı bir link kullanılır.**
- Bir **RAID** (Redundant Array of Independent Disks) disk sistemi ortak **paylaşılan bir disk sistemi** olarak kullanılabilir.



(b) Shared Disk

Cluster yapıları

Passive standby

- Primary makine çalışırken **secondary makine hazır bekler.**
- Primary makine çalışır olduğunu **periyodik olarak secondary makineye bildirir.**
- **Secondary** makine **mesaj alamadığında** otomatik olarak **aktif hale gelir.**
- **Oluşturulması kolaydır.**
- **Maliyeti yüksektir.** Secondary makine başka bir iş için kullanılmaz.
- **Passive standby** bir **cluster yapısı olarak değerlendirilmez.**

Cluster yapıları

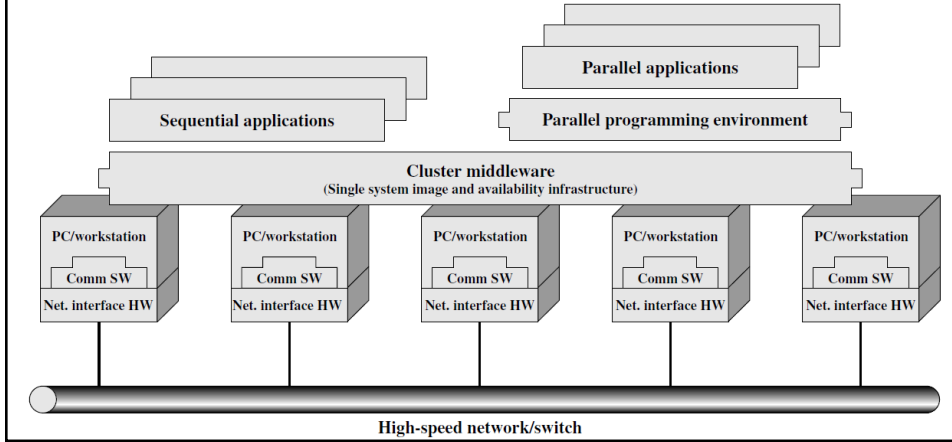
Active secondary

- Ayrı sunucular
 - Tüm **sunucular kendi diskine sahiptir.**
 - Veri **primary** ile **secondary** makine **arasında sürekli kopyalanır.**
 - **High availability** sağlar.
 - Veri **kopyalama** yüksek **overhead oluşturur.**
- Paylaşımsız
 - Ortak **diskler partition'lara bölünür** ve **her partition bir bilgisayara atanır.**
 - **Bir bilgisayar hata verirse,** ona ait volume **başka bilgisayar tarafından yönetilir.**
- Disk paylaşımı
 - **Çok sayıda bilgisayar** aynı anda **aynı diski paylaşır.**
 - Tüm bilgisayarlar **tüm volume'lere erişebilir.**
 - Disklerde **veri tutarlılığı** için **lock mekanizması kullanılır.**

Cluster yapıları

Cluster mimarileri

- Cluster'daki **bilgisayarlar bağımsız işlem yapabilir.**
- Middleware layer** yazılım, **high availability, load balancing** ve **failure management** sağlar.



Cluster yapıları

Blade server mimarisi

- Blade server mimarisinde bir kasada **çok sayıda server** vardır.
- Her blade; işlemci, hafıza ve hard diske sahiptir.**

