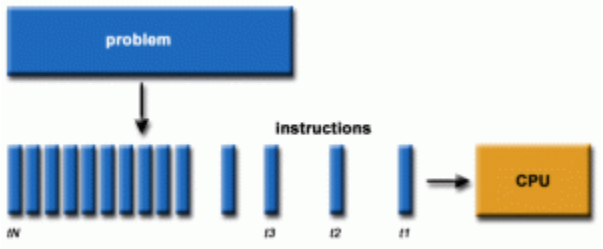
12. Paralel Programlama

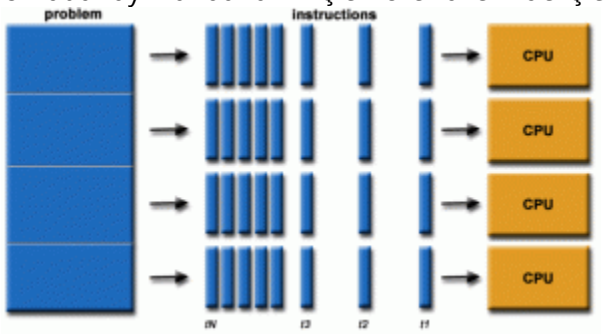
Geleneksel yöntemlerde programlar seri hesaplamalar şeklinde yazılır.

- Program bir bilgisayarda bir işlemci tarafından yürütülür.
- Problem farklı komut serilerine bölünür.
- Her komut bir diğerinden sonra işleme alınır.
- Birim zamanda sadece bir komut işlenir.



Basit şekliyle paralel programla, bir problemin çözümünde birden çok hesaplama kaynağının aynı anda kullanılmasıdır.

- Problemin çözümünde birden çok işlemci kullanılır.
- Problem eşzamanlı çözülmek için farklı parçalara bölünür.
- Her parça farklı komut serilerine bölünür.
- Komutlar aynı anda farklı işlemciler üzerinde işlenir.



Nerelerde kullanilir?

Bilim ve mühendislik alanlarda

- Bioscience, Biotechnology, Genetics
- Chemistry, Molecular Sciences
- Mechanical Engineering
- Computer Science, Mathematics

Endüstriyel ve ticari alanlarda

- Databases, data mining
- Web search engines, web based business services
- Financial and economic modeling

Neden kullanilir?

- Zaman ve para kazanmak
  Teorik olarak bir işe daha fazla kaynak ayrıldığında tamamlanması daha kısa sürer ayrıca paralel bilgisayarların maliyetleri süper ilgisayarlara göre daha ucuz.

- Büyük problemleri çözmek
  Çok büyük yada karmaşık problemlerin tek bilgisayarda çözümü çok zor yada imkansız olmakta.
- Aynı anda kullanımısağlamak
  Tek bilgisayarda birim zamanda bir işlem yapılmakta fakat birden fazla bilgisayarda çok daha fazla iş aynı anda yapılır. Özellikle binlerce kişinin aynı anda kullandığı sistemlerde.
- Lokal olmayan kaynakların kullanımı
  Network üzerinden yada internetten diğer bilgisayarların kaynaklarının kullanılabilir.
- Seri programlamanın limitleri
  İşlemci teknolojisinin küçülme sınırları.
  Ekonomik limitler. Bir işlemcinin daha hızlı yapılabilmesi maliyetli.

Paralel işlem düzeyleri

- **Bit Seviyesinde paralellik**
  - ALU içinde
- **Komut seviyesinde paralellik**
  - Çok sayıda komutun aynı clok da işlenmesi
- **Bellek seviesinde paralellik**
  - Hesaplamarda bellek işlemlerinin örtüşmesi
- **İşletim sistemi seviyesinde paralellik**
  - Birden çok işlemci yer alır
  - Birden fazla iş aynı anda çalışır
    - **Döngü seviyesinde**
    - **Procedur seviyesinde**

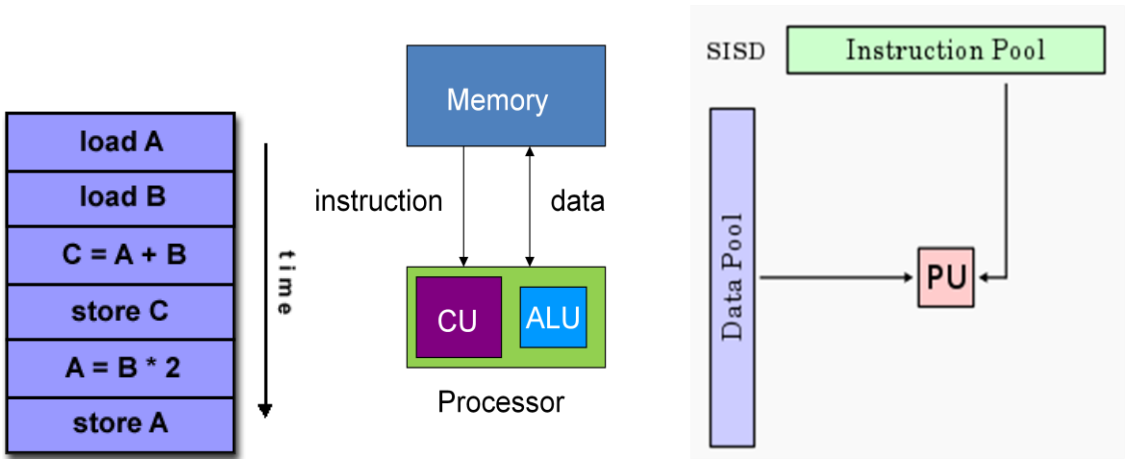
Flynn's Taxonomy

Paralel mimarilerin önemli özelliklerine göre daha detaylı incelenebilmeleri için bir sınıflandırma yapmak faydalı olacaktır . Bu sınıflandırmaya dair basit bir model için Flynn taksonomisi incelenebilir . Bu sınıflandırma metodunda paralel bilgisayarlar bir anda işlenebilen komut sayısı ve veri akışı türlerine göre 4 farklı mimari olarak sınıflandırılmıştır .

- Single Instruction stream - Single Data stream (SISD)
  Tek komut Tek data
- Single Instruction stream - Multiple Data stream (SIMD)
  Tek komut, Çok data
- Multiple Instruction stream - Single Data stream (MISD)
  Çok Komut, Tek Data
- Multiple Instruction stream - Multiple Data stream (MIMD)
  Çok komut, Çok data

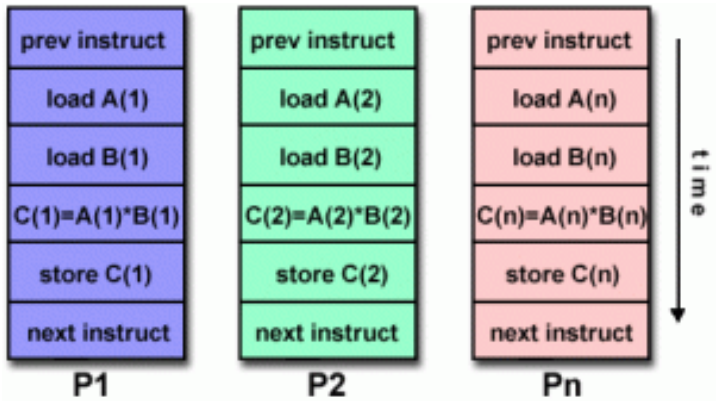
Tek işlem, tek data (Single instruction, single data (SISD))

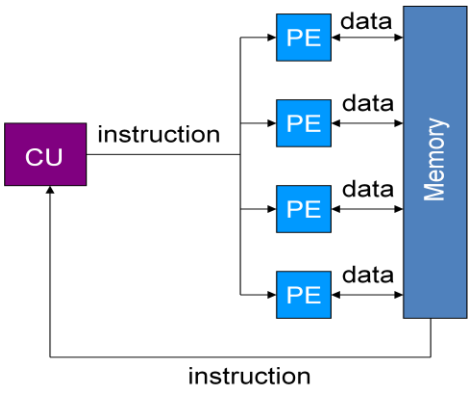
Seri bilgisayarlardır. Birim zamanda sadece tek bir işlem yapılır. Birim zamanda sadece tek bir data akışı kullanılır. SISD sistemlere örnek olarak Von Neumann mimarisini kullanan klasik seri bilgisayarlar verilebilir.

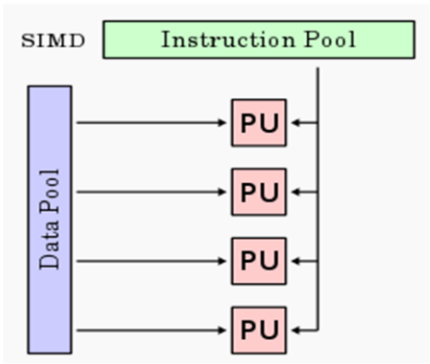


Tek işlem, çoklu data (Single instruction, multiple data (SIMD))

Paralel bilgisayar çeşididir. Birim zamanda sadece tek bir işlem yapılır. Birden çok data akışı kullanılır. Bir komut ile birden çok data üzerinde işlem yapılır

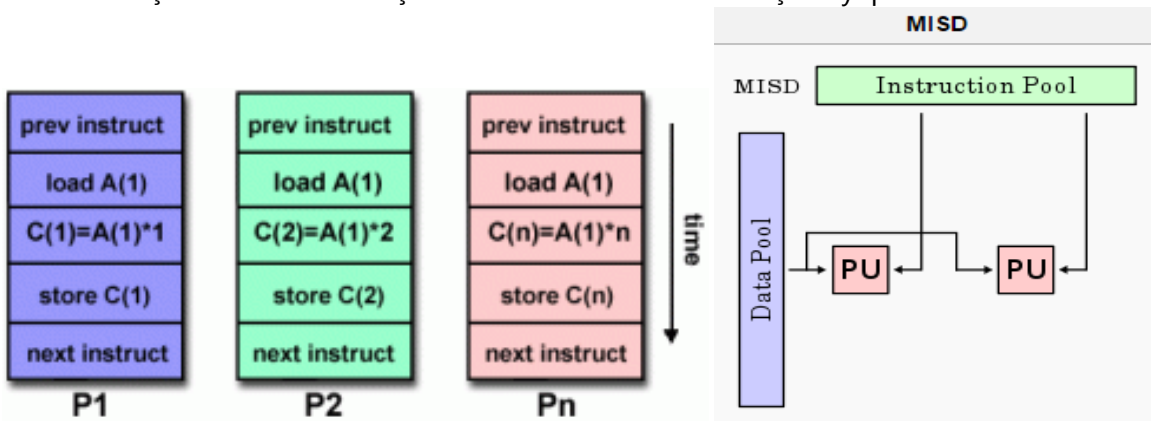






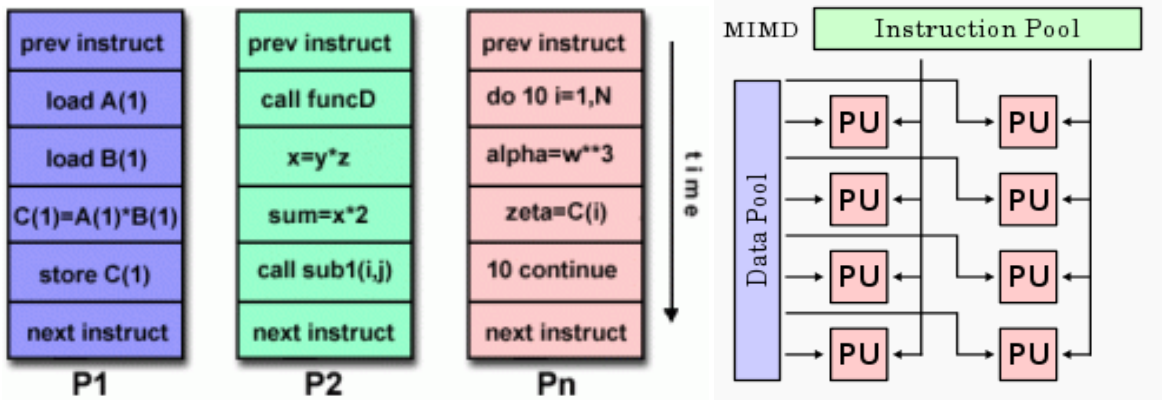
Çoklu işlem, tek data (Multiple instruction, single data (MISD))

Paralel bilgisayar çeşididir. Birim zamanda birden çok işlem yapılır. Birim zamanda sadece tek bir data akışı kullanılır Birden çok komut bir data üzerinde işlem yapar..



Çoklu işlem, çoklu data (Multiple instruction, multiple data (MIMD))

Paralel bilgisayar çeşididir. Birim zamanda birden çok işlem yapılır. Birden çok data akışı kullanılır. Senkron yada asenkron olabilir. Birden fazla komut birden fazla data üzerinde işlem yapar

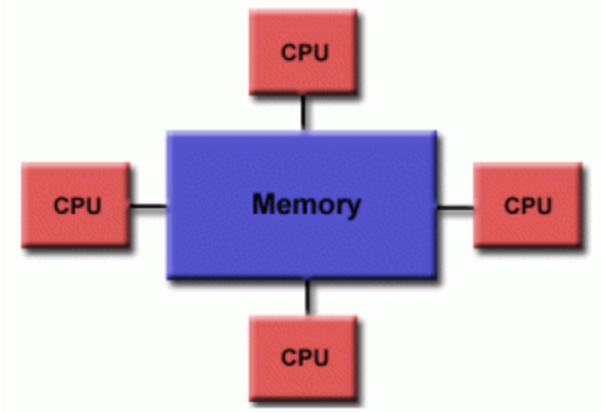


MIMD de kullanılan bellek yapısı

Paylaşımlı bellek(Shared Memory)

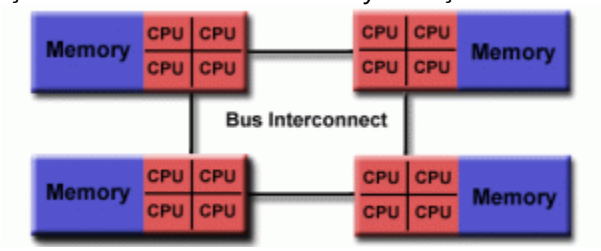
Genel özellikleri

- Paralel bilgisayarlarda çeşitli paylaşımlı bellek çeşitleri vardır fakat genel olarak bütün işlemciler belleğe global adres uzayından erişir.
- Çoklu işlemciler işlemlerini kendi başlarına yaparlar fakat aynı bellek kaynağını paylaşırlar.
- Bir işlemcinin bellek alanında yaptığı değişiklik bütün işlemciler tarafından görülür.
- Paylaşımlı bellek kullanan bilgisayarlar belleğe ulaşım zamanlarına göre iki ana sınıfa ayrılırlar UMA ve NUM
- Uniform Memory Access (UMA):
  - Simetrik çoklu işlemcili (SMP) bilgisayarlarda kullanılır.
  - Belleğe erişim ve erişim hızları eşittir



- Non-Uniform Memory Access (NUMA):

  - Fiziksel olarak birbirlerine bağlı iki yada daha fazla SMP'den oluşur.
  - Bir SMP direk olarak diğer SMP'lerin belleklerine ulaşabilir.
  - Bütün işlemciler bütün belleklere aynı erişim süresine sahip değildir.



Paylaşımlı belleğin avantajları:

- Bellek erişimiiçin global adres uzayının kullanılması programlama açısından kullanıcı dostudur.
- İşlemciler arası data paylaşımı hızlıdır.

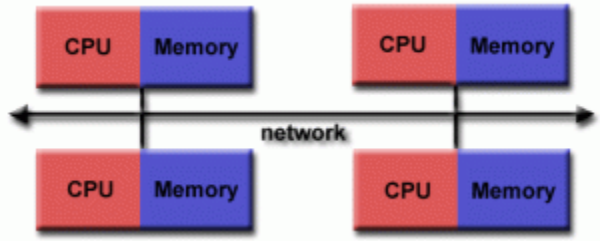
Paylaşımlı belleğin dezavantajları:

- Paylaşımlı belleği kullanan işlemci sayısı artırılırsa işlemci bellek arasındaki trafik artacağından darboğaz oluşturur.
- Belleğin senkronizasyonu ve belleğe doğru şekilde erişilmesi programcının sorumluluğundadır.

Dağıtık bellek (Distributed Memory)

Genel özellikleri

- Dağıtık belleklerde bellekler arası iletişim ağı vardır.

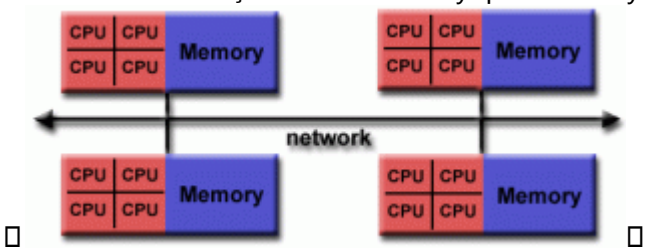


- İşlemcilerin kendi lokal bellekleri vardır. Bir işlemci diğer bellek adreslerini bilmez bu yüzden bellek adresleri için global adres uzayı konsepti yoktur.

- Her işlemci kendi belleğini organize eder ve yaptığı değişiklikler diğer işlemcilerin belleklerini etkilemez.
- Avantajları
  - İşlemci sayısı artarsa orantılı olarak bellekte artar.
  - Her işlemci kendi belleğine hemen erişebilir ağ iletişimi ile uğraşmaz.
- Dezavantajları
  - İşlemciler arası data iletişiminin detaylarından programcı sorunludur.
  - Bellek erişim süreleri farklılık gösterebilir.

Karma bellek (Hybrid Distributed-Shared Memory)

- Günümüzde daha çok karma bellek yapısıkullanılıyor.

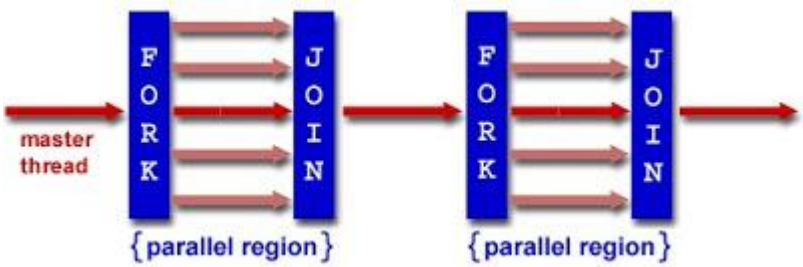


- Her bellek modülünün ait olduğu bir işlemci topluluğu var bu kısım paylaşımlı bellek oluyor.
- Her bellek işlemci topluluğu da birbirleriyle ağ üzerinden bağlı bu kısımda dağıtık bellek oluyor.
- Bu sistem paylaşımlı ve dağıtık bellek sistemlerinin avantaj ve dezavantajlarını hepsini taşıyor.

OpenMP

Çok çekirdekli bir bilgisayarda, paralel olarak komutları çalıştıran bir programlama arayüzüdür. Yazdığınız programları hızlandırmak için, programı parçalara bölüp çekirdeklere dağıtmak gerekir. OpenMp bu işe yaramaktadır. Bir görevi, küçük bir birinden bağımsız çalışmasında mahzuru olmayacak şekilde bölüp çoklu işlemcilerde eş zamanlı olarak çalıştırılır.

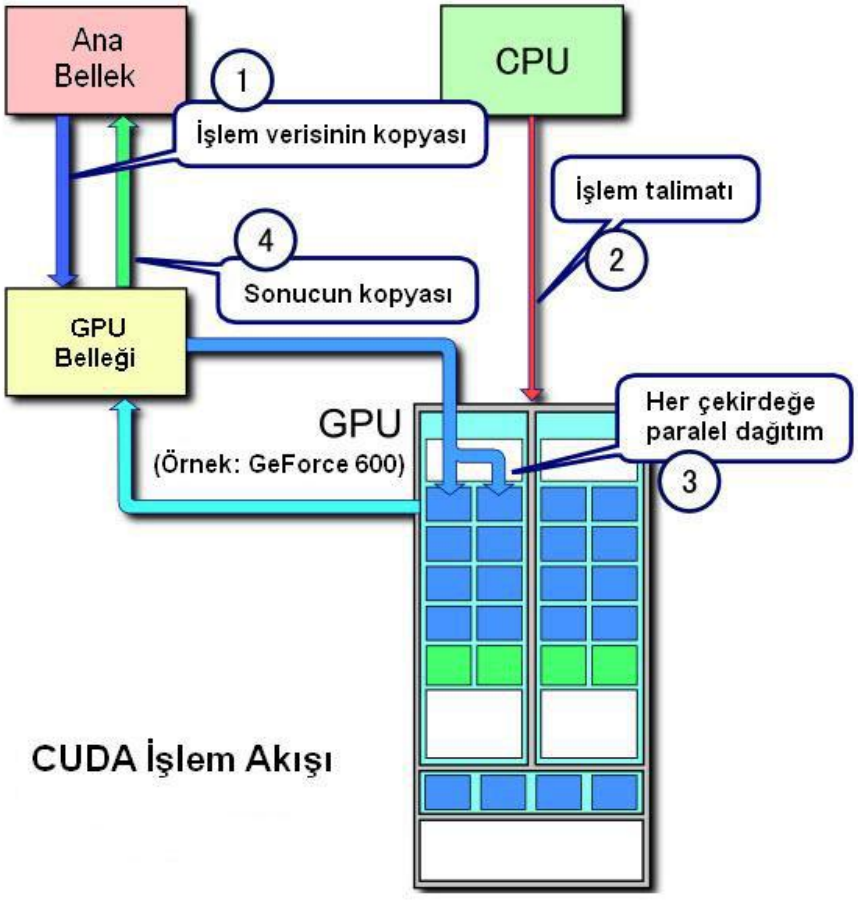
OpenMP bu işi kolaylaştırmak için yapılmış API dır. Pek çok donanım ve işletim siteminde çalışan ve basit derleyici ön işlem direktifleriyle kullanacağınız bir sistemdir. OpenMP nin çalışma sistemi program paralel işletilmesi gereken kısımlarda dallanır paralel çalışıp işlem bitince tekrar birleşmesi şeklinde olur. (Fork - Join Model)



CUDA

CUDA (**C**ompute **U**nified **D**evice **A**rchitecture), GPU (**G**raphics **P**rocessing **U**nit) için NVIDIA'nın sunduğu C programlama dili üzerinde eklenti olarak kullanıma sunulan bir paralel programlama mimarisi ve teknolojidir. CUDA, GPU üzerindeki yüzlerce(ilerde binlerce) çekirdeği kullanarak genel amaçlı matematik işlemleri yapmaya imkan sağlayan bir GPU mimarisi, yazılımı ve programlama platformudur. Programcılar C ve C++ programlama dillerine yapacakları bir ekleme sayesinde CUDA'yı edinebilir ve kullanabilirler.

CUDA Nasıl İşler?
Genellikle video işleme ve dönüştürme konusunda kullanılan CUDA'nın direkt olarak bu işe yönelmediğini en başından belirtmek gerek. Birbiri ile veri paylaşımı yapabilen paralel dizilime sahip çekirdekler, CPU'nun tek düzen şeklinde yapacağı işi yayarak gerçekleştirir. Farklı hatlara yüklenen işlemler yavaş gerçekleşir fakat tek yolda yapılabilecek süreden daha kısa sürede işlem sonuçlanır. Bunu daha iyi anlamak için aşağıdaki görseli inceleyebilirsiniz.



Yani kısacası konu kodlara ve CPU'nun tek başına kaldıramayacağı ağırlıktaki yüklere CUDA göğüs gerer, işi paylaşır ve kısa süre içerisinde bitmesini sağlar. Yalnız bu noktada bir şeyi de atlamamak gerek. CUDA'dan faydalanmak için öncelikle bu özelliği açmak gerekiyor. Yukarıda daha önce belirttiğimiz gibi kullanılan programlama diline yapılan ufak bir ek ile CUDA aktif hale getirilebiliyor