



BİLGİSAYAR MİMARİSİ

Çok İşlemcili Mimariler Araştırma Ödevi

Ayben GÜLNAR-191180041

DECEMBER 2022

İçindekiler

1.ÖZET.....	3
2. MULTİPROCESSORS TARİHÇESİ	3
3. GELİŞMİŞ BİLGİSAYAR MİMARİSİNE GİRİŞ	4
3.1 Four Decades Of Computing.....	5
3.2 Flynn’s Taxonomy Of Computer Architecture	5
3.3 Kuck Classification Scheme.....	8
3.4 Skillicorn Classification Scheme.....	8
3.5 Interconnection Networks	9
3.5.1 Mode of Operation	9
3.5.2 Control Strategy	9
3.5.3 Switching Techniques	9
3.5.4 Topology	9
3.6 Analysis And Performance Metrics.....	10
4. MULTİPROCESSORS SİSTEMLERİN AVANTAJ VE DEZAVANTAJLARI	11
5. MULTİPROCESSOR SİSTEMLERDE GELECEKTEKİ GELİŞİMİ TRENDLERİ.....	12
6. MULTİPROCESSORS PROGRAMMING MODELS	12
7.SONUÇ	14
8. KAYNAKÇA	15

ŞEKİLLER TABLOSU

Şekil 3-0-1	5
Şekil 3-2-1	6
Şekil 3-2-2	7
Şekil 3-2-3	7
Şekil 3-2-4	7
Şekil 3-4-1	108
Şekil 3-6-1	10

1.ÖZET

Çok işlemcili mimariler (multi-processor architectures), bir bilgisayar sisteminde birden fazla işlemci kullanarak paralel işlemeyi sağlayan mimariler olarak tanımlanabilir. Bu mimarilerin gelişimi, bilgisayarların işlem güçlerinin artması ve işlemlerin daha hızlı tamamlanması için bir gereksinim olmuştur.

Çok işlemcili mimariler, ilk olarak 1960'larda paralel işlemleri gerçekleştirebilen ilk bilgisayarların ortaya çıkışıyla başlamıştır. Bu dönemde, çok işlemcili mimariler daha çok araştırma ve ileri teknoloji üzerine odaklanmıştır. Ancak, 1970'lerin ortalarından itibaren çok işlemcili mimariler kullanımının yaygınlaşmasıyla birlikte, bu mimariler daha çok verimliliği ve hızı artırmak için kullanılmaya başlamıştır.

Günümüzde ise çok işlemcili mimariler, çok sayıda işlemcinin bir arada çalışmasını sağlayarak, çok büyük veri setlerini daha hızlı işleyebilme ve bu verilerden daha hızlı sonuçlar çıkarabilme gibi avantajlar sağlamaktadır. Ayrıca, bu mimariler sayesinde, çok büyük boyutta ve karmaşık olmayan işlemler de daha hızlı tamamlanabilmektedir[1].

Kısaca, çok işlemcili mimarilerin tarihçesi, bilgisayar teknolojisinin gelişimine paralel olarak gelişmiş ve günümüzde çok işlemcinin bir arada çalışmasını sağlayarak verimliliği ve hızı artırmak için kullanılmaktadır.

Bu çalışmada ise çok işlemcili mimarilerin kısaca tarihçesi, detaylıca tanımı ve özelliklerine yer verilmiştir.

2. MULTİPROCESSORS TARİHÇESİ

Çok işlemcili bilgisayarlar yeni değil, ancak ilkleri süper Bilgisayar topluluğu tarafından inşa edilmiş ve programlanmıştır. SoC teknolojisi, MPSoC'ler için çok işlemci çekirdekleriyle birlikte, tedarikçileri gelişmiş sistemleri standart ürünlere dönüştürmeye olanak sağlar. Çok sayıda insan paralelleşmenin standartlaşmasının sorunlara neden olabileceğini endişe etti - haklıydılar. İlk olarak Intel mikroişlemciler üretti, ardından diğerleri. 2000'lerin başında, Intel ve AMD 1 GHz engelini aşmaya çalıştılar; bu, 2 GHz ve üzerine yükseldi, güç tüketimi problemi ortaya çıktı. Transmeta'dan Crusoe işlemcisi duyurusu, işlemcilerini optimize etmelerine yol açtı ve Centrino gibi mobil işlemci aileleri ortaya çıktı. Centrino'nun düşük güç profili, daha önce imkansız gibi görünen uygulamaları gerçek kıldı ve müşteriler dizüstü bilgisayarları toplu olarak satın aldı. Mobil bilgisayarlar bir standart haline gelmeye başladı. O

zaman bile, gömülü işlemci pazarı, masaüstü ve dizüstü pazardan çok daha büyüktü, ancak seçenekler sınırlıydı ve ARM neredeyse bir de facto standarttı. Aynı çip üzerinde çoklu çekirdek odaklanması IBM, Sun, Intel ve AMD tarafından duyurulan çoklu çekirdek ürünleri ile başladı. Bu, güç tavanını kırmadan performans savaşını kazanma paradigmasıydı. MPSoC artık sadece bir akademisyenin konferans konusu değildi ve masaüstü paralel bilgisayarlaşma ulaşılabilir hale geldi. Nvidia ve AMD (ATI) de GPU tabanlı paralel bilgisayar kullanmaya başladı. Nvidia ve AMD (ATI) de GPU tabanlı paralel bilgisayar kullanmaya başladı. Daha yeni, Nvidia Tegra ve Intel Atom ile rekabet daha da artmıştır. IBM, 2005 yılında Cell işlemcisini duyurdu. Cell, oyun konsolu için bir heterojen çok işlemeli çözüm, ucuz süperbilgisayar hayalini tuttu. İnsanlar PlayStation 3'lerin onlarcasını yüzlercesini bir araya getirdiler ve kendilerine karmaşık araştırmalar için laboratuvar yetiştirilmiş süperbilgisayarlar geliştirdiler. Bu olurken, SystemC ve SystemVerilog gibi sistem seviyesi tasarım dilleri, işlem seviyesi modelleme, doğrulama ortamları ve araçlarının kavram ve tanımları ile baş etmeye çalıştı. Ancak, çoklu çekirdekli işlemciler yeni zorluklar ve EDA şirketleri için fırsatlar doğurdu. Anakütle işlemci dünyasının yüksek seviyeli modelleme, benzetim ve doğrulama teknolojisi talep etmesiyle, gömülü dünya çoklu çekirdekli ve heterojen çok işlemcili platformları (örneğin Motorola Zoom, Apple'ın iPad2, Samsung'un Galaxy) ortaya çıkardı. Donanım inovasyon alanını hayal dünyasını aşan bu platformlar ayrıca multimedya işlemede, coprocessor tabanlı hızlandırma ve dokunma ve el işareti tanıma yazılımında sofistikasyon gerektirdi. Bu, doğrulama ve doğrulama için zorlukları çözmek için aşırı inovasyona zorladı. IEEE Yüksek Seviyeli Tasarım Doğrulama ve Test (HLDVT) çalıştayı 16. yılına girdi. Şimdi, daha önce hiç olmadığı kadar, yüksek seviyeli modelleme, doğrulama ve test çözümlerine ihtiyacımız var. HLDVT, geçmiş 15 toplantıda bu nedenle adanmıştır. Mevcut çoklu çekirdek ve heterojenite patlaması ve platform inovasyonlarını sürükleyen kitle pazarı ekonomisi ile, ilgisi daha belirgin hale gelmiştir [2].

3. GELİŞMİŞ BİLGİSAYAR MİMARİSİNE GİRİŞ

Bilgisayar mimarları her zaman bilgisayar mimarilerinin performansını artırmaya çalışmışlardır. Yüksek performans, geçmişteki yoğun devrelerden, paketleme teknolojilerinden ve paralellikten gelebilir. Tek işlemcili süper bilgisayarlar duyulmamış hızlara ulaştı ve donanım teknolojisini çip üretiminin fiziksel sınırına kadar zorluyor. Ancak bu eğilim yakında sona erecek çünkü tek işlemcili bir sistemle elde edilebilecek hesaplama gücünü sınırlayan fiziksel ve mimari sınırlar vardır. Mesaj iletme sistemlerinde, ara bağlantı ağı statik ve dinamik olarak ikiye ayrılır. Statik bağlantılar, programlar çalışırken değişmeyen sabit bir topolojiye

sahiptir. Dinamik bağlantılar, program yürütülürken anında bağlantılar oluşturur. Çok işlemcili kullanmanın ana argümanı, birden çok işlemciyi basitçe bağlayarak güçlü bilgisayarlar oluşturmaktır. Çok işlemcili bir sistemin, en hızlı tek işlemcili sistemden daha hızlı hıza ulaşması beklenir. Ek olarak, birkaç tek işlemciden oluşan birçok işlemcinin, yüksek performanslı tek bir işlemci oluşturmaktan daha uygun maliyetli olması beklenir. Çok işlemcili bir başka avantajı hata toleransıdır. Bir işlemci arızalanırsa, geri kalan işlemciler düşük performansla da olsa sürekli hizmet sunabilmelidir.

3.1 Four Decades Of Computing

Çoğu bilgisayar bilimcisi, bilgi işlemin dört farklı paradigması veya dönemi olduğu konusunda hemfikir. Bunlar: toplu iş, zaman paylaşımı, masaüstü ve ağ. Tablo 1.1, Lawrence Tesler tarafından önerilen bir tablodan değiştirilmiştir. Bu tabloda, farklı bilgi işlem paradigmalarının ana özellikleri, 1960'tan başlayarak her on yıllık bilgi işlemle ilişkilendirilir.

TABLE 1.1 Four Decades of Computing

Feature	Batch	Time-Sharing	Desktop	Network
Decade	1960s	1970s	1980s	1990s
Location	Computer room	Terminal room	Desktop	Mobile
Users	Experts	Specialists	Individuals	Groups
Data	Alphanumeric	Text, numbers	Fonts, graphs	Multimedia
Objective	Calculate	Access	Present	Communicate
Interface	Punched card	Keyboard and CRT	See and point	Ask and tell
Operation	Process	Edit	Layout	Orchestrate
Connectivity	None	Peripheral cable	LAN	Internet
Owners	Corporate computer centers	Divisional IS shops	Departmental end-users	Everyone

LAN, local area network.

Şekil 3-0-1

Bilgi işlemdeki açık eğilimlerden biri, pahalı ve uzmanlaşmış paralel makinelerin daha uygun maliyetli iş istasyonları kümeleriyle değiştirilmesidir. Bir küme, bazı ara bağlantı ağları kullanılarak bağlanan bağımsız bilgisayarların bir koleksiyonudur. İnternetin yaygınlığı, ağ hesaplamaya ve son zamanlarda grid hesaplamaya ilgi yaratmıştır. Grid'ler coğrafi olarak dağıtılmış hesaplama platformlarıdır. Üst düzey hesaplama tesislerine güvenilir, tutarlı, yaygın ve ucuz erişim sağlamalıdır.

3.2 Flynn's Taxonomy Of Computer Architecture

Bilgisayar mimarisinin en popüler sınıflandırması, 1966'da Flynn tarafından tanımlandı. Flynn'in sınıflandırma şeması, bir bilgi akışı kavramına dayanmaktadır. Bir işlemciye iki tür bilgi akar: talimatlar ve veriler. Talimat akışı, işlem birimi tarafından gerçekleştirilen talimat

dizisi olarak tanımlanır. Veri akışı, bellek ile işlem birimi arasında değiş tokuş edilen veri trafiği olarak tanımlanır. Flynn'in sınıflandırmasına göre, talimat veya veri akışlarından biri tekli veya çoklu olabilir. Bilgisayar mimarisi aşağıdaki dört farklı kategoride sınıflandırılabilir:

- . Single-instruction single-data streams (SISD)
- . Single-instruction multiple-data streams (SIMD)
- . Multiple-instruction single-data streams (MISD)
- . Multiple-instruction multiple-data streams (MIMD)

Geleneksel tek işlemcili von Neumann bilgisayarları, SISD sistemleri olarak sınıflandırılır. Paralel bilgisayarlar SIMD veya MIMD'dir. Yalnızca bir kontrol birimi olduğunda ve tüm işlemciler aynı komutu senkronize bir şekilde yürüttüğünde, paralel makine SIMD olarak sınıflandırılır. Bir MIMD makinesinde, her işlemcinin kendi kontrol birimi vardır ve farklı veriler üzerinde farklı komutlar yürütebilir. MISD kategorisinde, aynı veri akışı, farklı talimat akışlarını yürüten doğrusal bir işlemci dizisi boyunca akar. Pratikte geçerli bir MISD makinesi yoktur; ancak, bazı yazarlar ardışık düzendeki makineleri (ve belki de sistolik dizili bilgisayarları) MISD örnekleri olarak kabul etmişlerdir. Şekil 1.1, 1.2 ve 1.3, sırasıyla SISD, SIMD ve MIMD'nin blok diyagramlarını göstermektedir [3].

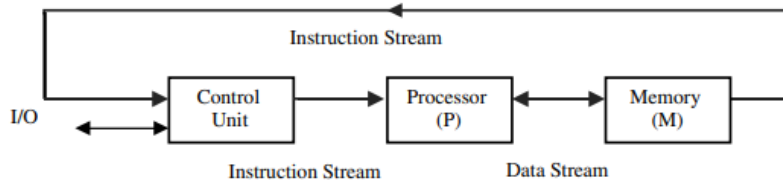


Figure 1.1 SISD architecture.

Şekil 3-2-1

Paralel hesaplamının SIMD modeli iki bölümden oluşur: olağan von Neumann stilinde bir ön uç bilgisayar ve Şekil 1.4'te gösterilen bir işlemci dizisi. İşlemci dizisi, farklı veriler üzerinde aynı işlemi aynı anda gerçekleştirebilen bir dizi özdeş senkronize işleme elemanıdır. Dizideki her işlemci, paralel olarak işlenirken dağıtılan verilerin bulunduğu küçük bir yerel belleğe sahiptir. İşlemci dizisi, ön ucun bellek veri yoluna bağlıdır, böylece ön uç, başka bir bellekmiş gibi yerel işlemci belleklerine rasgele erişebilir [3].

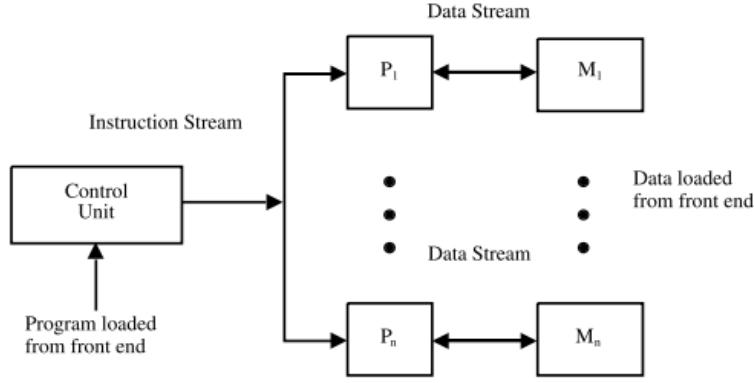


Figure 1.2 SIMD architecture.

Şekil 3-2-2

Çoklu talimat çoklu veri akışları (MIMD) paralel mimarileri, bazı ara bağlantı ağları aracılığıyla birbirine bağlanan çoklu işlemcilerden ve çoklu bellek modüllerinden yapılıdır. İki geniş kategoriye ayrılırlar: paylaşılan hafıza veya mesaj iletme. Şekil 1.6, bu iki kategorinin genel mimarisini göstermektedir. İşlemciler, paylaşılan bellek sistemlerinde merkezi paylaşılan bellekleri aracılığıyla bilgi alışverişinde bulunurlar ve mesaj geçiş sistemlerinde ara bağlantı ağları aracılığıyla bilgi alışverişinde bulunurlar [3].

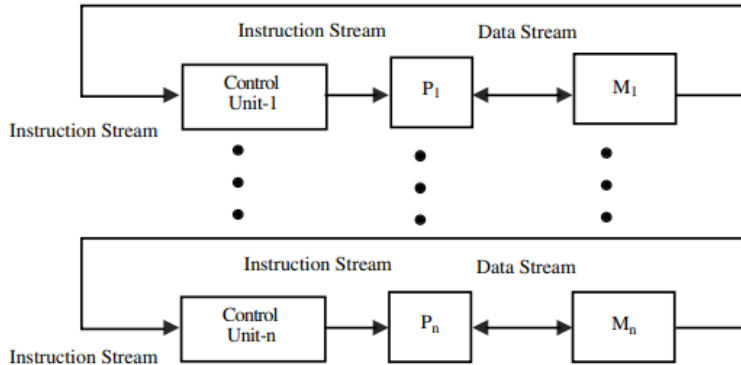


Figure 1.3 MIMD architecture.

Şekil 3-2-3

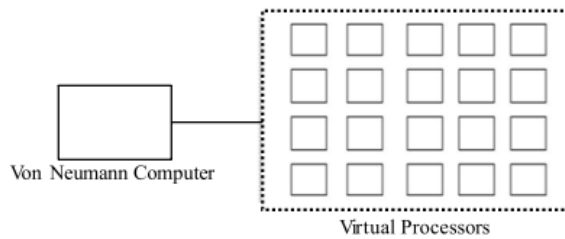


Figure 1.4 SIMD architecture model.

Şekil 3-2-4

3.3 Kuck Classification Scheme

Flynn'in vergi sınıflandırması, birçok bilgisayar mimarı tarafından genişletilen genel bir sınıflandırma olarak düşünülebilir. Böyle bir genişleme, D. J. Kuck tarafından 1978'de tanıtılan sınıflandırmadır. Sınıflandırmasında, Kuck, tek (skaler ve dizi) ve çoklu (skaler ve dizi) akışlı emirleri genişletti. Kuck'ın sınıflandırmasındaki veri akışı, çalıştırma akışı olarak adlandırılır ve ayrıca tek (skaler ve dizi) ve çoklu (skaler ve dizi) akışları içerecek şekilde genişletilmiştir. Bu akışların kombinasyonu, Tablo 11.3'de gösterildiği gibi mimari toplam 16 kategorisi sonucunu verir. Anahtar gözlemimiz, Flynn'in ve Kuck'ın sınıflandırmalarının mimari alanının tamamını kapsadığıdır. Ancak Flynn'in sınıflandırması emir kümesi seviyesinde mimari tanımlarını vurgularken, Kuck'ın sınıflandırması donanım seviyesinde mimari tanımlarını vurgular [4].

3.4 Skillicorn Classification Scheme

İşsel doğası gereği, Flynn'in sınıflandırması, benzer mimari özelliklere sahip ancak farklı işlevlere sahip bilgisayar sistemlerini tek bir sınıfta gruplamakla sonuçlanabilir. Aynı gözlem, 1988'de tanıtılan Skillicorn sınıflandırmasının ana nedeni olmuştur. Bu sınıflandırmaya göre, Şekil 11.5'te gösterildiği gibi soyut bir von Neumann makinesi modellenmiştir.

Görüldüğü gibi soyut model, komut işlemcisi (IP) ve veri işlemcisine (DP) ek olarak, komut belleği (IM) ve veri belleği (DM) olmak üzere iki bellek alt bölümü içerir. Sınıflandırma şemasını geliştirirken, aşağıdaki olası ara bağlantı ilişkileri dikkate alınmıştır: (IP – DP), (IP – IM), (DP – DM) ve (DP – IP). Ara bağlantı şeması, veri işlemcileri, veri hafızaları, komut işlemcileri ve talimat hafızaları arasındaki bağlantıların tipini ve sayısını dikkate alır. Hiç, birden çoğa ve çoktan çoğa bu tür bağlantılar olmayabilir [4].

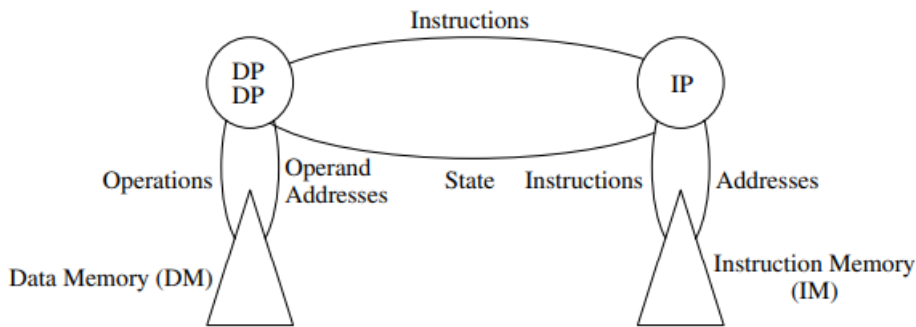


Figure 11.5 Abstract model of a simple machine

3.5 Interconnection Networks

Çok işlemcili ara bağlantı ağları (IN'ler), bir dizi kritere göre sınıflandırılabilir. Bunlar, (1) çalışma modunu (eşzamanlıya karşı eşzamansız), (2) kontrol stratejisini (merkeziye karşı merkezi olmayan), (3) anahtarlama tekniklerini (devreye karşı paket) ve (4) topolojiyi (statike karşı dinamik) içerir [3].

3.5.1 Mode of Operation

İşletim moduna göre, IN'ler senkron ve asenkron olarak sınıflandırılır. Senkron işletim modunda, sistemdeki tüm bileşenler tarafından tek bir küresel saat kullanılır ve böylece tüm sistem bir kilitlenme şeklinde çalışır. Asenkron işletim modu ise, global bir saat gerektirmez. Bunun yerine, asenkron sistemlerin işletimini koordine etmek için kolaylaştırıcı işaretler kullanılır. Senkron sistemler asenkron sistemlere göre yavaş olabilirler ancak yarış ve tehlikelerden arındırılmışlardır [4].

3.5.2 Control Strategy

Kontrol stratejisine göre, IN'ler merkezi ve dağıtık olarak sınıflandırılabilir. Merkezi kontrol sistemlerinde, sistemin bileşenlerinin işletimini denetlemek ve kontrol etmek için tek bir merkezi kontrol birimi kullanılır. Dağıtık kontrolde ise, kontrol işlevi sistemdeki farklı bileşenler arasında dağıtılmıştır. Merkezi kontrol biriminin işlevi ve güvenilirliği merkezi bir kontrol sisteminde engelleyici bir etken olabilir. Çapraz çubuk bir merkezi sistemdir ancak çok aşamalı bağlantı ağları dağıtıktır [4].

3.5.3 Switching Techniques

Bağlantı ağları, anahtarlamanın mekanizmasına göre devre ve paket anahtarlama ağları olarak sınıflandırılabilir. Devre anahtarlama mekanizmasında, bir kaynak ve hedef arasındaki iletişimin başlamasından önce tam bir yol oluşturulmalıdır. Oluşturulan yol, iletişimin tamamı boyunca varlığını korur. Paket anahtarlama mekanizmasında, bir kaynak ve hedef arasındaki iletişim, daha küçük birimlere ayrılmış mesajlar aracılığıyla gerçekleşir. Hedeflerine ulaşana kadar, paketler bir düğümden diğerine depolama ve ileme yöntemiyle gönderilebilir. Paket anahtarlama, devre anahtarlama göre ağ kaynaklarını daha verimli kullanmayı hedeflerken, değişken paket gecikmeleriyle karşı karşıya kalır [4].

3.5.4 Topology

Topolojilerine göre, IN'ler statik ve dinamik ağlar olarak sınıflandırılır. Dinamik ağlarda, girişler ve çıkışlar arasındaki bağlantılar anahtarlama elemanları kullanılarak yapılır.

Anahtarlama ayarlarına göre farklı bağlantılar oluşturulabilir. Statik ağlarda ise düğümler arasında doğrudan sabit yollar vardır. Statik ağlarda anahtarlama elemanları (düğümler) yoktur.

Bağlantı ağlarının sınıflandırılması için genel kriterler tanıttıktan sonra, topolojisine dayalı bir IN vergi sistemini tanıtmak mümkündür.

Şekil 11.12'de bu tür bir vergi sistemi verilmektedir.

Gösterilen vergi sistemine göre, IN'ler ya statik ya da dinamik olarak sınıflandırılır. Statik ağlar daha ileri bir biçimde, bağlantı desenlerine göre tek boyutlu (1D), iki boyutlu (2D) veya hiperküpler (HC) olarak sınıflandırılabilir. Dinamik ağlar ise, bağlantı şemasına göre otobüs tabanlı veya anahtarlama tabanlı olarak sınıflandırılabilir. Otobüs tabanlı IN'ler tek otobüs veya çoklu otobüs olarak sınıflandırılır. Anahtarlama tabanlı dinamik ağlar, bağlantı ağı yapısına göre tek aşamalı (SS), çok aşamalı (MS) veya çapraz çubuk ağları olarak daha ileri bir biçimde sınıflandırılabilir [4].

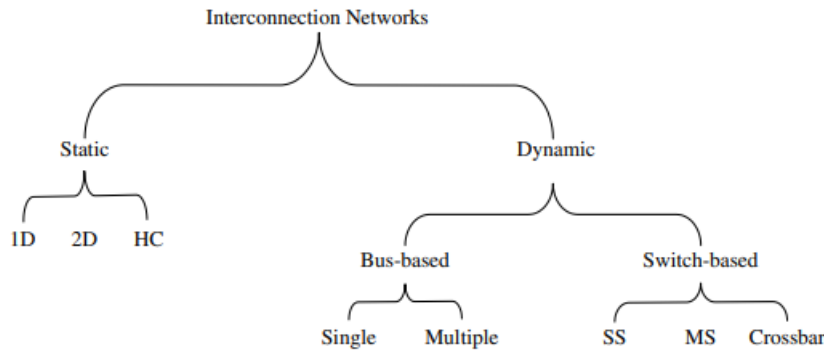


Figure 11.12 A topology-based taxonomy for interconnection networks

Şekil 3-6-1

3.6 Analysis And Performance Metrics

Multiprocessor mimarisi hakkında bir giriş yaptıktan sonra, multiprocessorlarda performans konuları hakkında bazı temel fikirler sunuyoruz. Sorulan en temel soru, bir verilen problemi multiprocessorlar kullanarak ne kadar hızlı çözebileceğimizdir? Bu soru aşağıda tanımlanan hızlandırma faktörü ile ifade edilebilir.

$S(n)$ = speed-up factor

= Increase in speed due to the use of a multiprocessor system consisting of n processors

= $\frac{\text{Execution time using a single processor}}{\text{Execution time using } n \text{ processors}}$

N işlemcinin her birinin ne kadar verimli kullanıldığıdır. Soru, aşağıda tanımlanan verimlilik şeklinde formüle edilebilir.

$$E(n) = \text{Efficiency}$$

$$= \frac{S(n)}{n} \times 100\%$$

Bir çoklu işlemci kullanarak görevleri (programları) yürütürken, belirli bir görevin her biri bir işlemci tarafından yürütülebilen n eşit alt göreve bölünebileceği varsayılabilir. Bu nedenle, beklenen hızlanma $S(n) = n$ tarafından verilirken, verimlilik $E(n) = \%100$ olacaktır. Belirli bir görevin, her biri bir işlemci tarafından yürütülen n eşit alt göreve bölünebileceği varsayımı gerçekçi değildir. Aynı bölümde bir dizi başka performans ölçütü de tanıtılmış ve analiz edilmiştir [4].

4. MULTİPROCESSORS SİSTEMLERİN AVANTAJ VE DEZAVANTAJLARI

Yüksek Verim: Verim, CPU tarafından belirli bir zamanda yürütülen işlemlerin sayısıdır, bu nedenle bu tür bir sistem daha yüksek verime sahiptir.

Paralel işleme türü: Paralel işleme, aynı anda birden fazla işlemin (thread olarak da bilinir) yürütülmesi anlamına gelir.

Daha az elektrik kullanımı: Tek bir işlemci sisteminde, aynı anda birçok işlemin yürütülmesi gerektiğinden daha fazla yük vardır. Ancak çok işlemcili sistemlerde birden fazla işlemin yürütülmesi birkaç defada yapılır. Bu, çok işlemcili CPU'ların tek bir işlemciden daha az elektrik tükettiği anlamına gelir.

Yüksek Güvenilirlik: Birden fazla işlemci işini aralarında paylaştığından, iş işbirliği ile tamamlanır. Bu, bu sistemlerin güvenilir olduğu anlamına gelir.

Ekonomik: CPU'lar tarafından daha fazla iş tamamlandıkça bu sistemler ekonomik olarak da iyidir.

İletişim: Birden çok işlemci birbiriyle iletişim kurduğundan, işletim sistemi uygulamasının işlenmesi karmaşıktır.

Daha fazla bellek gerekli: Birbiriyle çalışan çok işlemcili olduğundan, her işlemcinin bellek alanına ihtiyacı vardır.

Kilitlenme: Herhangi bir işlemci zaten G/Ç cihazını kullanıyorsa, diğer işlemciler kilitlenme yaratan aynı G/Ç cihazını kullanamaz.

Performans: Herhangi bir işlemci çalışmazsa, iş diğer işlemcilere bölünür. Kötü etki, işin yüksek sürede tamamlanması ve sistemin performansının etkilenmesi olacaktır.

Pahalılık: Bu tür sistemlerin satın alınması pahalıdır [5].

5. MULTIPROCESSOR SİSTEMLERDE GELECEKTEKİ GELİŞİMİ TRENDLERİ

Moore'nın Kanunu'nun gelecek on yıla kadar uzatılması için hiçbir temel engele rastlanmıyor. Özellik boyutu her işlem jenerasyonunda %30 azalarak, yüksek performanslı mikroişlemcide transistor sayısı ikiye katlanır. Bu mikroişlemcideki transistor sayısının büyük artışı, kritik işlevlerin entegrasyonunu ve mikroişlemcinin performansını büyük ölçüde artırır. Mikroişlemci ile bellek arasındaki boşluk, mikroişlemci hızının ana bellekten daha hızlı artması nedeniyle genişlemeye devam eder. Çok düzeyli cache belleğinin entegrasyonu, yavaş belleğin etkisini azaltır. Daha büyük cache boyutları, coherency misses'den daha fazla conflict ve capacity misslerini azaltır. Mikroişlemcide bellek ve I/O denetleyicilerini entegre etmek, bellek erişim gecikmesini azaltarak ve bus bant genişliği gereksinimlerini azaltarak. Mikroişlemcideki kötüleşen küresel bağlantılar, frekans skalasına ve daha fazla entegrasyona önemli bir zorluk oluşturur. İyileştirilmiş metallizasyon ve düşük-k malzeme orta vadeli bir çözümdür. Uzun vadeli çözüm, mikroişlemci ve sistem mimarisini yeniden yapılandırarak, bir plakadaki bileşenler ve büyük bir sistemdeki bileşenler arasındaki iletişim maliyetini minimuma indirmeyi hedefleyebilir. Yüksek güç tüketimi, bir mikroişlemcinin frekans ve performans skalasının kritik bir engelli hale gelmesine neden olur. Depleted Substrate Transistor ve gelişmiş güç yönetimi, mikroişlemcinin güç tüketiminin hızlı artışını kısıtlamak için vaadedici yollardandır [6].

6. MULTIPROCESSORS PROGRAMMING MODELS

Programlama modeli geliştirme, devrimci olmaktan çok evrimsel olacak ve eğilim, tamamen yeni programlama paradigmalarının geliştirilmesinden ziyade yerleşik yazılım dillerini ve teknolojilerini desteklemek olacaktır. Şu anda ve öngörülebilir gelecekte, büyük sistemler çoğunlukla C++, Java veya C# gibi Microsoft Ortak Dil Çalışma Zamanı (CLR) tarafından desteklenen dillerde yazılacaktır. Java ve CLR destekli diller hem sıkıca bağlı hem de gevşek bağlı programlama için programlama modelleri oluşturmuştur. Kısaca ifade edilirse, sıkıca

bağlı bilgi işlem, bir SMP modelindeki bazı varyantlarla (yani, iş parçacıkları, monitörler, koşullar ve sinyaller) yapılır ve heterojen dağıtılmış bilgi işlem, bir bileşen nesne modelindeki (örn. CORBA, Enterprise Java Fasulyeleri) bazı varyantlarla gerçekleştirilir. , Microsoft DCOM6 ve gelişmeleri). C++ evrimi7 için son teklifler ayrıca C++ standart kitaplık belirtimi içinde SMP ve dağıtılmış nesne modelleri için çağrıda bulundu.

MultiFlex sisteminde kullanılan iki SoC paralel programlama modeli, büyük sistem geliştirme için ana akım yaklaşımlardan esinlenmiştir, ancak SoC alanı için uyarlanmış ve sınırlandırılmıştır.

- DSOC modeli: Bu model, CORBA ve Microsoft DCOM dağıtılmış bileşen nesne modellerini anımsatan, heterojen dağıtılmış bilgi işlemi destekler. Bu bir mesaj iletme modelidir ve çok basit bir CORBA benzeri sistem arayüzü tanımlama dilini (sistemimizde SIDL olarak adlandırılır) destekler.

- Paylaşılan belleğe erişen eşzamanlı iş parçacıklarını destekleyen SMP: Burada kullanılan SMP programlama kavramları, Java ve Microsoft C#'ta yer alanlara benzer. Uygulama, zamanlamayı gerçekleştirir ve iş parçacıkları, monitörler, koşullar ve semaforlar için destek içerir. Her iki programlama modelinin de uygulamaya bağlı olarak güçlü ve zayıf yönleri vardır. MultiFlex sisteminde trafik yöneticisi ve MPEG4 video kodlayıcı uygulamalarında daha ayrıntılı olarak gösterileceği gibi, her ikisi de birlikte çalışabilir bir şekilde birleştirilebilir. Bir yandan, bu yaklaşım Java veya C# ile aşina olan programcılar için doğal olmalı, ancak diğer yandan, gelişmekte olan SoC cihazlarında kullanım için yeterince verimli (hem yürütme verimliliği hem de kaynak gereksinimleri açısından) olmalıdır [7].

7.SONUÇ

Çok işlemcili mimariler, bir işlemcinin bir tek işi yerine birden fazla işlemci kullanarak işlem yapmayı amaçlar. Bu mimariler, bir işlemcinin tek başına sağlayabileceği performansı artırmak için birden fazla işlemci kullanarak paralel işlem yapmayı sağlar. Bunun sonucunda, bir işlemcinin tek başına yapabileceği işlemler daha hızlı ve verimli bir şekilde yapılabilir. Bu mimariler ayrıca birden fazla işlemcinin bir arada kullanılmasının avantajlarından yararlanarak daha fazla güç tasarrufu sağlar. Genel olarak, çok işlemcili mimariler daha hızlı ve verimli işlem yapma kabiliyetine sahiptir ve bu nedenle birçok uygulama için tercih edilir. Çok işlemcili mimarilerin birçok avantajı vardır. Örneğin, bu mimariler işlemcinin tek başına sağlayabileceği performansı artırabilir ve bu da daha hızlı işlem yapmayı mümkün kılar. Ayrıca, bu mimariler birden fazla işlemci kullanarak paralel işlem yaparak işlemleri daha verimli bir şekilde gerçekleştirebilir. Bu da işlemlerin daha hızlı tamamlanmasını sağlar ve bu da sistemlerin daha hızlı çalışmasına yardımcı olur. Çok işlemcili mimariler ayrıca birden fazla işlemcinin bir arada kullanılmasının avantajlarından yararlanarak daha fazla güç tasarrufu sağlar. Bu, işlemcinin tek başına kullanılmasına göre daha verimli bir şekilde çalışmasını sağlar ve bu da sistemlerin daha uzun süre kullanılabilmesine yardımcı olur. Son olarak, çok işlemcili mimariler daha esnek bir mimari sağlar. Bu mimariler, işlemcileri kolaylıkla ekleyebilir veya çıkarabilirler.

Özetlemek gerekirse bu araştırmamda çok işlemcili mimarileri çok detaylıca her yönüyle ele aldık.

8. KAYNAKÇA

- [1] Bilgisayar Donanım Tarihi", Wikipedia, Erişim: 12 Aralık 2022, https://tr.wikipedia.org/wiki/Bilgisayar_donanim_tarihi
- [2] Shukla, Sandeep & Mishra, Prabhat & Zilic, Zeljko. (2011). A Brief History of Multiprocessors and EDA. IEEE Design & Test of Computers. 28. 96. 10.1109/MDT.2011.50.
- [3] Hesham El-Rewini and Mostafa Abd-El-Barr. 2005. Advanced Computer Architecture and Parallel Processing. Wiley-Interscience, USA.
- [4] S. K. Rajput, "Fundamentals of Computer Organization and Architecture," New Age International, 2010.
- [5] Advantages and Disadvantages of Multiprocessor Systems," IT Release, 2020. [Online]. Available: <https://www.itrelease.com/2020/06/advantages-and-disadvantages-of-multiprocessor-systems/>. [Accessed: 12-Dec-2022].
- [6] Yung, R. & Rusu, S. & Shoemaker, K.. (2002). Future trend of microprocessor design. 43-46.
- [7] Jerraya, Ahmed & Bouchhima, Aimen & Pétrot, Frédéric. (2006). Programming models and HW-SW Interfaces Abstraction for Multi-Processor SoC. 280-285. 10.1109/DAC.2006.229246.