**1. Модели компютърни**

**архитектури**

**Класове компютърни архитектури**

- архитектура – компоненти и организация на системата;

- фон Нойманова (1.3.1) – на възли и мрежи и некласическа организация (систолични, потокови, логически и редукционни модели и невронни мрежи); - класификация на Michаel Flynn (1966) по управление на потока инструкции и потока данни (операнди) – SISD, SIMD, MISD и MIMD архитектури – (1.3.2)

- SIMD – за векторна обработка, фина грануларност; MISD – за конвейрна обработка (обработнащи фази върху вектор) – систолични масиви; MIMD – обикновено с локална и глобална памет; за средна и едра грануларност.

 - технологично-ориентирана таксономия на паралелните архитектури:

мултипроцесори, мултикомпютри, потокови машини, матрични процесори, конвейерни векторни процесори и систолични матрици – частично съответствие с класовете на Флин

**HW/SW паралелизъм** За паралелно изпълнение на програми е небходима едновременно апаратна и програмна поддръжка.

Апаратен паралелизъм. Обуславя се от архитектурата и ресурсите, които са баланс между производителността и цената.

Характеризират се с пикова производителност и средно

натоварване. Той задава зависимостта по ресурси.

Програмен паралелизъм. Обуславя се от зависимостта по данни и по управление. Реализира се като паралелизъм по управление - конвейризация, мултиплициране на функционални възли. Обслужва се паралелно, прозрачно за програмиста. паралелизъм по данни - типичен за SIMD, но и при MIMD.

**Метрика: ускорение и ефективност**

ускорение S(n) = T1/Tn; лимитиращи фактори

ефективност E(n) = S(n)/n

**SIMD -** обобщения модел включва контролно устройство и еднотипни обработващи модули с достъп към обща памет – (1.7.1)

- програмно-апаратна зависимост на паралелизма/ускорението – пример за изпълнение на програма на SIMD машина (1.7.2) -nпроцесорните елементи изпълняват операциите във формат битове или думи;

- локалната памет за данните може да бъде разпределена, обща или йерархична (със свързваща мрежа) (1.7.3)

- особености:

опростена архитектура спрямо MIMD поради общото контролно устройство (за дешифриране

и зареждане на инструкциите) и съответно поддържане само на едно копие от кода за инструкции

скаларните операции (включително контролната логика) се изпълняват от контролното устройство – евентуално конкурентно на паралелната обработка на данни в обработващите устройства

имплицитна синхронизация между отделните обработващи устройства (при MIMD – експлицитна)

**МISD**

това е архитектурния принцип на всички конвейри – вкл. на процесорния конвейер – обработката се разделя на последователни фази; обработката на следващата инструкция (при най-фина грануларност) или на следващия процес започва веднага щом предходния процес освободи първата фаза – (1.8.1) Прилагат се и функционални (или циклични) конвейри, например с фазите (1.8.2):

- четене на инструкциите от обща памет - зареждане в обработващото устройство с евентуално буфериране

-Обработка и пренос на резултата към общата памет (буфериране)

- запис в общата памет

Нива на конвейеризация: инструкционно, субсистемно (обикн при аритметична обработка – нелинейни конвейри с фази add, mul, div, sort…) и системно ниво (процеси, също и програмна организация)

**Систолични матрици** (Systolic Arrays)

Представляват модификация на MISD на субсистемно ниво, специализирана архитектура за определени алгоритми – с многодименсионни конвейри т.е. фиксирана мрежа от обработващи устройства

- Ограничено приложение – ЦОС (цифрова обработка на сигнали – DSP), обработка на образи и др.

- опростени процесорни елементи и комутирацинна съобщителна мрежа с ограничен

набор шаблони;

- управлението е по инструкции (control flow – не data flow) но програмирането е като

при потоковите архитектури

- архитектурата включва обработващ масив (с комутатор) и управляващ модул, който настройва масива, предава данните и извлича резултатите (+ контролен възел – хост) – (1.9.1)

- производителността се понижава значително при интензивен вход/изход

**-** тенденцията е към елемeнти за фина грануларност – на инструкционно ниво – снабдени с няколко високоскоростни дуплексни серийни канали (броя на които определя валентността – коеф. на съседство)

- пример: iWrap серия на Интел и университета Carnegie-Mellon (1.10.1) – процесорната клетка се състои от iWrap компонент с изчислителен и комуникационен агент и страницирана памет с директен интерфейс към компонента

 - пример: умножение на матрици в двумерен систоличен масив с коеф. на съседство 6

(1.10.2)

**MIMD**

това е архитектурния принцип на всички мултипроцесори и мултикомпютри: процесорите са автономни и могат да изпълняват различни програми (вкл. локално копие на ОС!) . Имат общ ресурс с разпределен конкурентен достъп – памет или комуникационна среда

 - обикновено се изграждат с масови процесори (вместо специализирани процесорни елементи с ограничени функции)

- наличието на автономна локална памет ги разделя на: системи с обща памет; синоними: мултипроцесори | [shared-memory | tightly-coupled] systems | Global-Memory MIMD, GM-MIMD | Uniform Memory Access System – UMA

 системи с обмен на съобщения; синоними: мултикомпютри, [distributed- memory | loosely-coupled] systems | Local-Memory MIMD, LM-MIMD | Non- Uniform Memory Access System – NUMA (поради наличието на локална и отдалечена памет)

- глобално и локално адресно пространство; виртуалната памет поддържа глобално адресно пространство на страниците (не на ниво думи), което се управлява от разпределена ОС (РОС) за МП и хомогенните МК. При МК общата виртуална памет се поддържа и с обмен на съобщения

- хетерогенните МК използват мрежови ОС (МОС), при които нивото на достъп е разпределена файлова система (напр. базирана на DNS)

**Мултикомпютри**

 Разпределената обща памет (distributed shared memory DSM): програмната имплементация на обща памет в система с автономни възли (и адресни пространства);

- виртуално общо адресно пространство от страници (не думи) – 4/8 kB – (което позволява програмиране за мултикомпютъра като за виртуален уникомпютър).При отсъствие на страница от локалната памет възниква вътрешно прекъсване (memory trap) и зареждане на страницата в локалната от отдалечената памет; Възможно е репликиране на страници само за четене (read only); Ако страницата е и за запис, се прилагат различни мерки за поддържане на свързаност

**Архитектура с обща памет (мултипроцесори)** – UMA (uniformly shared memory access) - еднакъв достъп на процесорите -

 силносвързани састеми. Архитектура:

- обща шина - разширение от унипроцесинг към мултипроцесинг

- комутируема матрица (crossbar switch) - многоканални мрежи (1.14)

**NUMA и COMA**

NUMA (non-uniformly shared memory access) – йерархия на общата памет - локални, глобални и/или клъстерни памети; COMA (cache only shared memory access) - паметта елакална (cache) но йерархията и позволява част от нея (“директория”) да се адресира отдалечено

**Потокови архитектури (Data Flow):** при класическите фон Нойманови архитектури (вкл. модификациите по Флин) програмата е последователност от инструкции, която се изпълнява от контролно устройство – control flow. При потоковите архитектури операциите се изпълняват веднага при наличие на операндите (и наличие на операционен ресурс) – контрола се осъщесвява чрез планиране на операндите т.е. данните; концептуално всички инструкции с готови операнди могат да се изпълнят паралелно (на практика конкурентно);

 Програмите за потокови архитектури се представят с потокови графи – възлите представят операции, а дъгите –информационните връзки на операндите; Нивото на паралелизъм обикновено е инструкционно (1.17.1)

- Допълнителни особености на потоковите архитектури: реконфигурация, буфериране на данните, комплементиране на операндите

 - Наличие на управляавщ процесор, който пакетира операндите и инструкцията в блок – token – и го предава на някой от обработващите процесори

**Статични потокови архитектури**

 статични – програмния (потоковия) граф е фиксиран. За изпълнение на повече от

 една програма се използват различни варианти на зареждането на данните, които се

 генерират на етапа компилация. Този модел не поддържа процедури, рекурсия и обработка на масиви;- Организация 1.18.1

 Статични с реконфигурация – логическите връзки между процесорните елементи се установяват на етапа зареждане на програмата: топологията на връзките се решава от компилатора и след зареждане на програма остава фиксирана при изпълнението; особености : - физическите канали съществуват, но се комутират;

- броя алокирани (заредени) процесори обикновено е по-малък от инсталираните

процесори поради ограничения в комутацията – логическата връзка между процесорите е дърво, не всички процесори в листата на което се използват

Пример – MIT Data Flow Machine – клетките памет съответстват на информацията във възлите на потоковия граф – т.е. инструкционните блокове (tokens) – когато блока е комплектован с операнди, той се предава като операционен пакет към елемент за обработка; пакета с резултата се връща в клетъчната памет

**Динамични потокови архитектури**

 базират се на логически канали между процесорите, които могат да се реконфигурират по време на изпълнение подобно на система с обмен на съобщения –с маркирани блокове (tagged tokens) дъгите в потоковия граф могат да съдържат повече от един блок едновременно (но с различни марки!)операциите се извършват когато възела получи блокове (с еднакви марки) на всичките си входящи дъги циклични итерации могат да бъдат изпълнявани паралелно: за целта всяка итерация се представя като отделен субграф като маркировката се разширява с номера на итерацията –1.18.1 (само при информационна независимост на итерациите!) пример – Manchester Data Flow Machine MDM: цикличен конвейер, в който блоковете циркулират и се управляват от ключов модул. Компонентите са (1.18.2) Блоков буфер (token queue) – за съхраняване на междинни резултати (ако се произвеждат по-бързо отколкото е последващата им обработка) – капацитет 32К блока и производитилност 2.5 МБлока/Сек Комплементираща памет (matching store) – за комплементиране на блоковете с еднакви марки – процеса е апаратен и поддържа до 1.25 МБлок Памет инструкции (instruction store) – n-торките (обикновено 2ки) операнди-блокове се пакетират с инструкции и адрес (етикет) на резултата и се предават за изпълнение

 обработващ модул (processing unit) – 20 процесора (24-битова дума и 4Кдуми вътрешна памет)

**Мрежи за връзка**

 осъществяват комуникациите между процесорните възли

 при всички видове МП и МК – статични и динамични

 (базират се на [каскади от] комутируми блокове - ключове)

топологии на свързване: пълен граф, линия и пръстен, двудименсинна циклична и ациклична мрежа, хиперкуб (n-куб), двоично дърво

shuffle exchange

**Характеристики на мрежите за връзка**