**2. Процесорни архитектури**

**Процесорни архитектури**

Main frame – широк архитектурен клас от компютри, прилагат се някой от

следните процесорни архитектури:

- Скаларни процесори CISC (Complex Instruction Set Computer); - RISC (Reduced

Instruction Set Computer); - Суперскаларни CISC; RISC, само че RISC се използва по-често по технологични причини;

- Процесори VLIW – Very Long Instruction Word; - Векторни; - Суперконвейрни [super piperline]; Основни характеристики на всички архитектурни класове процесори:

Процесорни цикли CPI; Тактова честота CR.

Тези два параметъра на пръв поглед са независими, но между тях съществува корелация, която може да се представи в диаграма на технологичното пространство:

**(2.4)**

**Фази на инструкционен конвейер**

Процесорната обработка на типична инструкция реализира MISD паралелизъм на инструкционно ниво и минава през фазите извличане (от cache - обикновено 1 инстр. за цикъл), декодиране (установява функцията за изпълнение и необходимите ресурси - регистри, магистрали, устройства), издаване (резервира ресурсите чрез блокиране и извлича операндите от регисрите към устройствата), изпълнение (1 или повече фази), записване (writeback - на резултатите в регистрите).(тук картинка с 4 квадратчета)

 **Времедиаграма на инструкционен конвейер:** Закъснението между две последователни инструкции е една фаза при скаларните процесори(2.6.1) Поради ресурсен конфликт между фазите на извличане и запис по-често се прилага закъснение на две фази (два субтакта) между инструкциите:(2.6.2)

 **Синхронни линейни конвейрни процесори ЛКП:** ЛКП е каскада от k процесорни фази (stages - Si), която изпълнява фиксирана функция върху данните, преминаващи през устройството от входа (S1) през последователните фази (Si→S(i+1)) към изхода му Sk. Те не са динамично (runtime) настройваеми т.е. са статични. Изпълняват операционни, аритметични и обменни инструкции. - Синхронните ЛКП са с интерфейс между фазите, който представлява синхронизиращи буферни ключове (latches) с общ такт. Фиг. 2.7. Ключовете са регистри които изолират входовете от изходите и предават данните синхронно във всички фази. Фазата с най-голямо закъснение определя общия такт и общата производителност Проявява се и фазово отместване s (skew[ing]) на такта при предаване на тактовия сигнал между фазите

**Асинхронни линейни конвейрни процесори ЛКП** Асинхронните ЛКП контролират потока данни с “Hand Shaking” протокол - Ready/Ack между Si→Si+1. Фиг. 2.8. Подходящи за комуникационни канали при системи с обмен на съобщения. Производителността на отделните фази може да варира.

**Нелинейни конвейрни процесори НЛКП** Динамични, настройваеми, допуска се разклонение, обратна връзка (feedback) и пре-предаване (feedforward) на данните за обработка. Фиг. 2.9.1. Изходът може да не е от последната фаза.

 **Карта на резервацията**. Тук не е тривиална като при ЛКП. За различните функции може да **варира по устройства и време** (тактове). Фиг. 2.9.2. Тя се дава и съвместимостта на последователните функции по устройства т.е. зависимостта им по ресурси

**Анализ на закъснението при НЛКП** - -Закъснението **(latency)** се представя от броя процесорни тактове *k* между две последователни инициирания на функции. - Опита за повече от едно инициране едновременно на едно устройство е колизия, която се избягва чрез *планиране* (*диспечеризация*, *scheduling*) на последователността от инициирания. Когато закъснението е такова, че предизвиква колизия, то е *забранено закъснение*; трябва да се избере последователност от закъснения, така че да не предизвиква колизия. Пример за две забранени закъснения с карта на резервацията - фиг.2.10.

- *Цикъл на закъснението* е последователност от закъснения, която се повтаря неопределено дълго. Интервалите между две последователни инициирания на функции в цикъла на закъснението може да са еднакви, (*константен цикъл*), но може и да са различни, при което се изчислява *средно закъснение*. Чрез коефициента на запълване на цикъла се получава ефективността на конвейера. **Инструкционен конвейер** ИК е специализиран за обработка на последователните инструкции в машинния код чрез припокриване (overlapping) Типичната инструкция минава през фазите извличане (от cache - обикновено 1 за цикъл), декодиране (установява функцията за изпълнение и необходимите ресурси - регистри, магистрали, устройства), издаване (резервира ресурсите чрез блокиране и извлича операндите от регисрите към устройствата), изпълнение (1 или повече фази), записване (writeback - на резултатите в регистрите)

 **Обработка на преходите** Конвейризацията се лимитира от зависимостта по данни и от инструкциите за преход Производителността при програма с 20%/10% вероятност за условен преход между последователните инструкции, 50% вероятност за изпълнение на условието (т.е. на прехода; статистически обаче повечето условни преходи - 60% -се изпълняват) и 8-фазен конвей ре 41%/25% по-малка отколкото производителността при програма, в която поне едната вероятност е 0. Затова при конвейрни процесори е желателно алгоритъмът да се кодира с минимум условни преходи. Предвиждането на преходите се използва за да се отложи прехода докато се изпълнят опр. брой инструкции, независими от условието на прехода. То може да бъде базирано на кода на програмата -статично или на историята на изпълнението – динамично.

 **Архитектура на набора инструкции** Разграничават се класовете RISC и CISC по следните параметри: формат на инструкцията и на данните • режими на адресация • регистърно адресиране (регистри с общо назначение) • управление на изпълнението на програмата (flow control instructions)

 **CISC** класическа архитектура (първите процесори са ограничен набор инструкции) уверичения набор инструкции настъпва с микропрограмирането с промянатана SWcost/HWcost (първосигнална реакция на семантичната ножица между HLL и процесорните архитектури/машинните езици)параметри:120 – 350 инструкции с няколко формата на инструкциите и данните 32 – 64 регистъра с общо предназначение 4 – 16 режима на адресиране голяма част от изразите на HLL са микрокодирани (т.е. имат съответствие в набора инструкции скаларни CISC процесори – за операции върху скаларни данни; частична конвейризация поради: зависимост по данни между последователните инструкции ресурсен конфликт

**RISC** 25% от машинните инструкции кодират 90% от HLL програмата и се изпълняват 95% от процесорното време подходи за оптимизация: трансформиране на микропоргмна памет в регистърен cache FPU и други специализирани устройства на процесорния чип суперскаларни процесори броя на инструкциите е < 100 – с фиксиран формат (предимно регистър-регистър) до 5 режима на адресиране, инструкциите са предимна от тип load/store “регистърни фалове” – по 32+ вътрешни регистри за бързо превключване между процесите едночипови, затова висока тактова честота CR и нисък CPI т.е. висок MIP коефициент скаларните RISC процесори са подобни на скаларните CISC но при еднаква тактова честота производителността може да е по-ниска поради по-малката плътност на кода необходимост от ефективен компилатор за постигане на високо ниво конвейризация на ниво инструкция суперскаларна RISC архитектура – фиг. 2.15.

 **Суперскаларни процесори (RISC и CISC)** Повече от 1 инструкция на такт поради наличието на няколко (напр. 3) инструкционни конвейера –съответно няколка резултата от всеки инструкционен цикълмодел MIMD инструкционно ниво разлика от векторните процесори, които реализират SIMD на инструкционно ниво паралелизма се реализира на инструкционно ниво – само между логически независими инструкции кратност на инструкцията m = 2 до 5 (при скаларните процесори m = 1) суперскаларен RISC процесор

 **VLIW процесори** Комбинират концепцията за хоризонтално микрокодиране и суперскаларна архитектура: дълги инструкции (стотици битове), които задават поняколко операции над операндитеразличават се от супескаларните процесори по бързо и просто декодиране на инструкциите понеже една VLIW инструкция замества няколко суперскаларни по-ниска плътност на кода но по-висок паралелизъм на инструкционно ниво (понеже инструкциите са с фиксиран формат който може да включва и ниезпълними операции, а суперскаларните операции са само изпълними) непреносим обектен код понеже нивото на паралелизма при различните процесори е различно а е заложено в самата дълга инструкция (супрескаларните архитектури са портабелни със скаларните) – само за специализирани компютри инструкционния паралелизъм се задава на етапа kомпилация – т.е. статичен; няма динамична диспечеризация и синхронизация VLIW процесор

**Технологии на процесорите: -** Суперскаларните архитектури са по-подходящи за паралелизъм по данни - многократни операции се изпълняват конкурентно на няколко еднотипни устройства (блокове за изпълнение, портове към регистрови файлове …). Затова необходимост от по голяма интеграция в чипа – CMOS технология. Комбинират се с RISC архитектура на процесорната дума. - Суперконвейрите се базират на паралелзъм по управление, поради което съществено при тях е прилагането на устройствас висока тактова честота – т.е. TTL технология.

**Intel Pentium**

С въвеждането на Пентиум архитектурата (1995) Интел прилага предимстват на некласическа паралелна архитектура в производството на процесор,

предназначен типично за масови компютри: - суперскаларен процесор с ниво на инструкционния паралелизъм m=2 (3 заР4) – едновременна обработка на 2 целочислени операнда по модела MIMD (когато последователните инструкции нямат зависимост по данни или управление!); - всеки инструкционен конвейер се състои от 5 фази: извличане, декодиране, адресна генерация (типичен CISC процесор с много режими на адресация нa ОП), изпълнение и запис;

- изпълнението на последователни инструкции от всеки конвейер е със закъснение 1 фаза (извличане): два самостоятелни I cache + D cache по 8 кБ **-** за ефективно съчетаване на работата на конвейрите (т.е. за избягване на някои от случаите на конфликт) работата на двата инструкционни конвейра е «дефазирана» със стъпка от 1 фаза (първата фаза «извличане»);

- със същата цел Dcache е с двупортова организация – по един самостоятелен порт за всеки от инструкционните конвейри - cache буферите са с асоциативна организация на достъпа: асоциативната памет има 32-байтов TLB с последните адреси така че търсенето на зарежданата страница става в 32 адреса (вместо 8К) - планирането на активните страници в cache е по дисциплината LRU - изискването за свързаност (кохерентност) между данните в cache и в ОП се постига чрез специален протокол – MESI – което позволява изграждането на мултипроцесорни архитектури (т. нар. симетричен мултипроцесинг –хомогенна мултипроцесорна архитектора с обща памет между процесорите

**Intel Multicore** Intel Core Microarchitecture технология прилага интегриране намашинната архитектура на симетричния мултипроцесинг в микропорцесор суперконвейрни суперскаларни ядра: 14 фази (коеф. На супреконвейрност 2); по 4 инструкции Две/четири независими ядра – NUMA мултипроцесинг с локализиран L1-cache за всяко ядро и общ L2-cache елементи на VLIW и RISC (едновременно!): поддържа х86 CISC набор инструкции, като декодира част от тях до 2 и повече конвейрни микроинтрукции – RISC т.нар. микрооперации обединяват няколко често срещани последователности от машинни инструкции за изпълнение като една инструкция – напр. проверка на стойност и преход по флаг са обединени в “микрооперацията” условен преход интегриран арбитраж на достъпа до L2-магистралата; разширени възможности на управление на енергийното потребление и версии