**Тема 2. Процесорни архитектури**

**Main frame –** това е широк архитектурен клас от компютри, които прилагат някой от следните процесорни архитектури: 1) Скаларни процесори CISC (Complex Instruction Set Computer); RISC (Reduced Instruction Set Computer); 2) Суперскаларни CISC; RISC, само че RISC се използва по-често по технологични причини;3) Процесори VLIW – Very Long Instruction Word; 4) Векторни; 5) Суперконвейрни [super piperline];

**Основни характеристики на всички архитектурни класове процесори**: 1) Процесорни цикли CPI; 2) 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: (фиг. 2.1) На диаграмата: колкото сме под правата CPI=1, толкова по-добре. Там са тези процесори, които имат ниво на паралелизъм изпълняват повече от 1 инструкция за 1 цикъл. Ако сме над CPI=1, то трябва повече от 1 цикъл за 1 инструкция. Процесорните архитектури RISC интерпретират прости команди (около 5 процесорни фази), докато CISC интерпретират по-сложни команди (около 7-8 проц. фази).

**Фази на инструкционен конвейер:** Процесорната обработка на типична инструкция реализира MISD паралелизъм на инструкционно ниво и минава през фазите: 1) извличане-fetch (от Instruction cache - обикновено 1 инстр. за цикъл), 2) декодиране-decode (от Instruction cache -установява функцията за изпълнение и необходимите ресурси - регистри, магистрали, устройства), 3) изпълнение - execute (от Data cache - състои се от 7-8 устройства, някой от които са свързани с паметта, в зависимо от операцията се избира устройството), записване – write back (от Data cache – ако има резултат, то той се записва на мястото на генерирания адрес). (фиг. 2.2)

**Времедиаграма на инструкционен конвейер:** 1) Закъснението м/у 2 последователни инструкции е една фаза при скаларните процесори (фиг. 2.3) 2) Поради ресурсен конфликт м/у фазите на извличане и запис по-чести се прилага закъснение на 2 фази м/у инструкциите (фуг. 2.4) на диаграмите: fetch (правоъгълник с ляв диагонал), decode (празен правоъгълник), execute (запълнен правоъгълник), write back (правоъгълник с десен диагонал).

**Синхронни линейни конвейерни процесори:** 1)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→Si+1) към изхода му Sk. Те не са динамично (runtime) настройваеми т.е. са статични. Изпълняват операционни, аритметични и обменни инструкции. 2) Синхронните ЛКП са с интерфейс между фазите, който представлява синхронизиращи буферни ключове (latches) с общ такт. 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 (Ppeak): Ppeak=f =1/τ, τ = τ max+dlach, където τmax е времето на най-бавната фаза, а dlatch е закъснението. Проявява се и фазово отместване s (skew[ing]) на такта при предаване на тактовия сигнал между фазите. Затова се избира τ = τ max+dlach +s. (фиг. 2.5)

**Асинхронни линейни конвейерни процесори:** Те контролират потока данни с “Hand Shaking” протокол - Ready/Ack между Si→Si+1. Подходящи са за комуникационни канали при системи с обмен на съобщения. Производителността на отделните фази може да варира. (фиг. 2.6) **Нелинейни конвейерни процесори НЛКП:** 1)Те са динамични, настройваеми, допуска се разклонение - обратна връзка (feedback) и пре-предаване (feedforward) на данните за обработка. Изходът може да не е от последната фаза. (фиг. 2.7) 2) Карта на резервацията. Тук не е тривиална като при ЛКП. За различните функции може да варира по устройства и време(тактове).Тя се дава и съвместимостта на последователните функции по устройства т.е. зависимостта им по ресурси. (фиг. 2.8)

**Анализ на закъснението при НЛКП:** 1)Закъснението (latency)се представя от броя процесорни тактове *k* между две последователни инициирания на функции.2) Опита за повече от едно инициране едновременно на едно устройство е колизия, която се избягва чрез планиране на последователността от инициирания. 3) 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. 2.9). 4)**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: 1)**ИК е специализиран за обработка на последователните инструкции в машинния код чрез припокриване(overlapping); 2)Типичната инструкция минава през фазите извличане, декодиране, издаване, изпълнение и записване; 3) Архитектурата на процесорния конвейер (фиг. 2.10) 4)Преподреждане на инструкциите за по-голям коефициент

на запълване на цикъла (фиг. 2.11)

**Обработка на преходите:** 1)Конвейеризацията се лимитира от зависимостта по данни и от

инструкциите за преход; 2)Производителността при програма с 20%/10% вероятност за условен

преход между последователните инструкции, 50% вероятност за изпълнение на условието и 8-фазен конвейер е 41%/25% по-малка отколкото производителността при програма, в която поне едната вероятност е 0. Затова при конвейерни процесори е желателно алгоритъма да се кодира с минимум условни преходи. 3)Предвиждането на преходите се използва за да се отложи прехода докато се изпълнят определен 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:1)формат на инструкцията и на данните; 2) режими на адресация; 3) регистърно адресиране (регистри с общо назначение); 4) управление на изпълнението на програмата.

**CISC (Complex Instruction Set Computer):** Това е Класическа архитектура (първите процесори са ограничен набор инструкции). Увеличения набор инструкции настъпва с микропрограмирането с промяната на SWcost/HWcost . Параметри:1) 120 – 350 инструкции с няколко формата на инструкциите и данните; 2) 32 – 64 регистъра с общо предназначение 3)4 – 16 режима на адресиране; 5)голяма част от изразите на HLL(high level language) са микрокодирани (т.е. имат съответствие в набора инструкции). Има скаларни CISC процесори – за операции върху скаларни данни и частична конвейеризация поради зависимостта по данни между последователните инструкции и ресурсения конфликт (фиг. 2.12).

**RISC (Reduced Instruction Set Computer)**: 25% от машинните инструкции кодират 90% от HLL програмата и се изпълняват 95% от процесорното време. Подходи за оптимизация:

1)трансформиране на микропоргмна памет в регистърен cache 2)FPU и други специализирани устройства на процесорния чип 3) суперскаларни процесори 4)броя на инструкциите е < 100 – с фиксиран формат (предимно регистър-регистър) 5) до 5 режима на адресиране, инструкциите са предимна от тип load/store 6)“регистърни фалове” – по 32+ вътрешни регистри за бързо превключване между процесите 7) едночипови, затова висока тактова честота CR и нисък CPI т.е. висок MIPS коефициент 8) 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 9)необходимост от ефективен компилатор за постигане на високо ниво конвейризация на ниво инструкция 10)суперскаларна RISC архитектура (фиг. 2.13).

|  |  |  |
| --- | --- | --- |
| Показатели | CISC | RISC скаларен |
| Бр. инструкции | 128-256-300 | 24-32 |
| Формат на инструкции | 16-64 бита, т.е. инструкцията е с плаваща дължина | 32 бита, т.е. инструкцията е с фиксирана дължина |
| Формат на адреси | 8-12 бита, различни начини на адресиране на операционната памет, къси/дълги | Регистър – регистър, 3-4 броя на регистърните формули |
| CPI брой процесорни тактове | 8-20 процесорни такта, т.нар. инструкции с различна степен на сложност | 3-6 процесорни такта, инструкциите са с фиксирана дължина – опростени |
| CM управляващ контролен модул | Базира се на микропрограмиране | С помощта на апаратна логика(АЛ) hardware control |

**Суперскаларни процесори (RISC и CISC) -** Повече от 1 инструкция на такт, поради наличието на няколко инструкционни конвейера – съответно няколко резултата от всеки инструкционен цикъл. Разлика им с векторните процесори е че векторните реализират SIMD на инструкционно ниво, докато суперскаларните реализират MIMD. Паралелизма се реализира на инструкционно ниво – само между логически независими инструкции, като кратност на инструкцията е m = 2 до 5, докато при скаларните процесори m = 1. На (фиг. 2.14) е показан суперскаларен RISC процесор.

**VLIW процесори:** Комбинират концепцията за хоризонтално микрокодиране и суперскаларна архитектура: дълги инструкции (стотици битове), които задават по няколко операции над операндите.Те се различават от суперскаларните процесори по: 1)бързото и просто декодиране на инструкциите, понеже една VLIW инструкция замества няколко суперскаларни; 2) имат по-ниска плътност на кода, но по-висок паралелизъм на инструкционно; 3)непреносим обектен код, понеже нивото на паралелизма при различните процесори е различно; 4)инструкционния паралелизъм се задава на етапа компилация – т.е. статичен, няма динамична диспечеризация и синхронизация. На (фиг. 2.15). е показан VLIW процесор.

**Векторни процесори:** Те са специализирани копроцесори за векторни операции – операндите в отделната инструкция са масив[и]. Дългите вектори (надвишаващи дължината на регистърните файлове) се сегментират. Инструкциите са тип: 1)регистър-регистър – кратки (адресират се регистърни файлове); 2)памет-памет – дълги (защото съдържат адреси от основната памет), те могат да обработват по-големи масиви с различна дължина. Типични векторни операции са: 1) зареждане на вектор от паметта на компютъра: V1←Mn; 2)запис: V1→Mn; 3)ескалиране: S1•V1→V2; 4) векторна операция, при която и двете операнди са вектори и резултата е вектор: V1•V2→V3; 5) редукция от векторни операнди и резултат скалар: V1•V2→S1. 6) зареждане вектор-вектор: • V1 →V2. 7) редукция на единичен вектор: • V1 → S1. 8) аналогични инструкции от тип памет-памет – операндите са от вида *Mi*(1 : *n*).

**Суперконвейерна архитектура:** При степен *n* цикъла на суперконвейера е 1/*n* от базовия цикъл на фазите. (фиг. 2.16). Закъснението за една операция е равно на базовия цикъл, но ILP e *n*.

*T*(1, *n*) = *k* + (*N*-1)/*n ; S*(1, *n*) =n\*(N+k-1)/(n\*k+N-1) →*n* за *N*→∞, където k е фазов базов конвейер, а N е последователни независими инструкции ; Cray1: *n*=3.

**Суперконвейрна суперскаларна архитектура:** Степента е (*m*, *n*) като *m* е кратността на едновременно издаваните инструкции (т.е. на супрескаларност), а *n* е кратността на супреконвейера (1/n от кратността на базовия цикъл между групите последователни

инструкции) (фиг. 2.17). Закъснението за една операция е равно на базовия цикъл, но ILP e *n*.

*T*(*m*, *n*) = *k* + (*N*-*m*)/*mn ; S*(*m*, *n*) =m\*n\*(k+N-1)/(m\*n\*k+N-1) →*nm* за *N*→∞, където k е фазов базов конвейер, а N е последователни независими инструкции ; DEC Alpha: *n*=6, *m*=2.

**Intel Pentium:** С въвеждането на Pentium архитектурата,Intel прилага предимствата на некласическа паралелна архитектура в производството на процесор, предназначен типично за масови компютри: 1)суперскаларен процесор с ниво на инструкционния паралелизъм m=2 (3 за

Р4) – едновременна обработка на 2 целочислени операнда по модела MIMD (когато последователните инструкции нямат зависимост по данни или управление!); 2)всеки инструкционен конвейер се състои от 5 фази: извличане, декодиране,адресна генерация (типичен CISC процесор с много режими на адресация на ОП), изпълнение и запис; 3)изпълнението на последователни инструкции от всеки конвейер е със закъснение 1 фаза (извличане): два самостоятелни кеша: Instruction cache и Data cache по 8 KB.4) за ефективно съчетаване на работата на конвейрите се използват двата инструкционни конвейра; 5) със същата цел Dата cache е с двупортова организация – по един самостоятелен порт за всеки от инструкционните конвейeри; 6)cache буферите са с асоциативна организация на достъпа, т.е. асоциативната памет има 32-байтов TLB (Translation Lookup Buffer) с последните адреси, така че търсенето на зарежданата страница става в 32 адреса; 7)планирането на активните страници в cache е по дисциплината LRU(Least Recenty User ); 8)изискването за свързаност (кохерентност) между данните в cache и в ОП се постига чрез специален протокол – MESI – което позволява изграждането на мултипроцесорни; 9)интегрирано FPU(Float Pointing Unit) устройство с 8-фазов конвейер (извличане, декодиране,адресна генерация изпълнение, обработка мантиса, обработка експонента, обработка приближение и запис, който може да изпълнява и две FP

инструкции едновременно (когато едната от тях е присвояване). (фиг. 2.18).

**Intel Multicore:** Intel Core Microarchitecture е технология, която прилага интегриране на машинната архитектура на симетричния мултипроцесинг в микропорцесор. Суперконвейрни суперскаларни ядра имат 14 фази с по 4 инструкции. Има две/четири независими ядра – NUMA мултипроцесинг с локализиран L1-cache за всяко ядро и общ L2-cache. Достъпа до L2-магистралата има интегриран арбитраж.