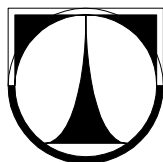


TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií



BAKALÁŘSKÁ PRÁCE

Liberec 2006

Petr Matějka

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika
Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Vývoj procesorů

CPU development

Bakalářská práce

Autor:	Petr Matějka
Vedoucí práce:	Mgr. David Kmoch
Konzultant:	Mgr. Jiří Vraný

Rozsah práce:	48 stran textu
	9 obrázků
	4 tabulky
	1 CD

V Liberci 18. 5. 2006

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 18.5.2006

Podpis:

Petr Matějka

Chtěl bych poděkovat mému kamarádovi Petru Fišerovi za jeho připomínky a návrhy. Rád bych také poděkoval vedoucímu bakalářské práce Mgr. Davidu Kmochovi za poskytnutí cenných poznatků.

Liberec, květen 2006

Autor Petr Matějka

Abstrakt

Tato práce se zabývá vývojem procesorů PC. Je zde stručně popsán vývoj procesorů, s nimiž úzce souvisí i vývoj samotných počítačů, jejichž „srdcem“ je právě procesor. Procesory za dobu své existence prodělaly neuvěřitelný vývoj. Je to až neuvěřitelné, jak se změnil výkon, architektura a hlavně technologie. Vývoj počítačů, a potažmo i procesorů, byl hodně ovlivněn válkou, která vždy uspíší technologický vývoj. Je pravda, že největší vývoj zaznamenaly procesory v posledních deseti letech. Hlavními tahouny se staly firmy AMD a Intel. Těm se snažily konkurovat i jiné firmy, které se však neprosadily, a buď zanikly, nebo byly koupeny silnějšími firmami. Hlavní využití mají procesory v osobních a serverových počítačích nebo noteboocích. Jsou využívány pro práci, hry, zábavu, ale především pro vědu, výzkum a vývoj (rozluštění DNA, vědecké projekty, zdravotnictví). Vlastně je už dnes všechno řízeno procesory. Počítače ovládají naše životy a bez nich by si asi člověk těžko dokázal představit svoji existenci.

Abstract

This work deals with a development of PC processors. The development of the processors, with which is connected the development of computers is here shortly described, whose „ heart" is the processor. CPUs during the time of their existence have gone through the incredible development. It is nearly unbelievable, how the achievement, architecture and mostly technology have switched. The development of computers, and also of the processors, was very influenced by the wars, which always hastens the technological development. It is truth that the biggest development CPUs noted in last ten years. The main draught horses became firms called AMD and Intel. Other firms tried to compete with them which were not successful, and either died out or was bought by stronger firms. Main usage CPUs have in personal, server computers or notebooks. They are used for work, games, entertainment but above all for the science and progress (enucleating DNA, scientific projects, health service). CPUs control our live as a matter of fact already today is everything controlled. Computer control our lives and man can hardly imagine his existence without them.

1 Úvod.....	10
2 ENIAC	12
3 von Neumannova architektura	13
3.1 Pět hlavních částí podle von Neumannovy architektury:.....	13
3.2 Základní odlišnosti dnešních počítačů od von Neumannovy architektury:	14
4 Generace počítačů	15
4.1 0. generace:	15
4.2 1. generace:	16
4.3 2. generace:	16
4.4 3. generace:	17
5 Architektura (struktura) procesorů.....	18
5.1 CISC.....	19
5.2 RISC.....	19
5.3 VLIW	20
5.4 Jak to dnes vypadá	20
6 Jak takový procesor pracuje	21
7 ARCHITEKTURA ČIPU	22
7.1 Pipeline.....	22
7.1.1 Klasické zpracovávání instrukcí	23
7.1.2 Zřetěžené zpracovávání instrukcí.....	23
7.1.3 Nevýhody pipeline	23
7.2 Superskalár.....	24
7.3 Out-of-order a spekulativní vykonávání instrukcí	24
7.4 Vykonávání podmínek	25
8 Cache paměti	26
8.1 Paměť cache L1 (first level cache).....	26
8.2 Paměť cache L2 (second level cache).....	26
9 Instrukční sady	27
9.1 Single Instruction Multiple Data (dále již jen jako SIMD).....	27
9.2 x86.....	27
9.3 x87.....	27
9.4 MultiMedia eXtensions (dále již jen jako MMX).....	27
9.4.1 Enhanced MMX	28
9.5 3DNow!, Enhanced 3DNow!	28
9.6 Streaming SIMD Extensions (dále již jen jako SSE).....	28
9.6.1 SSE2.....	28
9.6.2 SSE3.....	28
9.7 AMD64 (x86-64 nebo x64).....	29
9.7.1 Full support for 64-bit integers	29
9.7.2 Larger virtual address space.....	29
9.7.3 Larger physical address space	29
9.8 EM64T	30
10 Virtualizace	32
10.1 Virtualizace u serverů	32
10.2 Virtualizace pro desktop	32
10.3 Software pro virtualizaci	33
10.4 Jak to funguje ?	33
10.5 Využití.....	34
11 Parametry	35
11.1 HyperThreading (HTT).....	35

11.2 HyperTransport (HT)	35
11.3 Front Side Bus (FSB).....	35
11.4 Base frequency, source clock (základní frekvence)	35
11.5 Latence	36
11.6 Thermal Design Power (TDP)	36
12 Výrobní technologie.....	37
12.1 Používané materiály:.....	37
12.1.1 Bulk Silicon	37
12.1.2 Strained Silicon (SS).....	37
12.1.3 Silicon in Insulator (SOI).....	37
12.1.4 Strained Silicon Directly on Insulator (SSDOI)	37
13 Spotřeba procesorů.....	38
13.1 Vlastnosti redukující spotřebu.....	38
13.1.1 PState (Performance State)	38
13.1.2 Intel SpeedStep.....	38
13.1.3 Enhanced Intel SpeedStep s více PStates (EIST) - Nejnovější způsob...	38
13.1.4 Demand Clock Modulation (ODCM)	38
13.1.5 PowerNow! verze 1.4.....	39
13.1.6 Cool'n'Quiet.....	39
14 DualCore procesory (2 procesory v jednom „kabátě“)	40
14.1 Návrh.....	40
14.2 Co je to Thread.....	41
14.3 Multithreading.....	41
14.4 Problémy	41
14.5 Multitasking	42
5 CELL (konkurence pro DualCore).....	43
16 Mobilní a serverové procesory.....	45
16.1 Pentium M.....	45
16.2 Jádro Banias	45
16.3 Jádro Dothan	46
16.4 Současná nabídka	46
16.5 Serverové procesory.....	46
17 Závěr	47

Seznam obrázků

2	obrázek č.1	12
2	obrázek č.2	12
5.1	obrázek č.3	19
5.2	obrázek č.4	19
14.1	obrázek č.5	40
14.1	obrázek č.6	40
14.4	obrázek č.7	42
15	obrázek č.8	44
15	obrázek č.9	44

Seznam tabulek

4	tabulka č.1	15
8.8	tabulka č.2	26
9.1	tabulka č.3	28
9.1	tabulka č.4	29

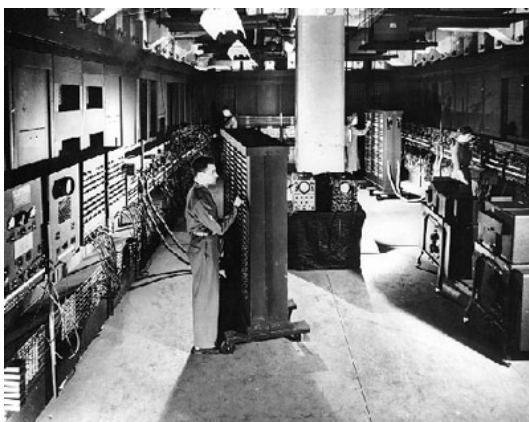
1 Úvod

Vývoj **Central Procesor Unit**, dále jen **CPU** je notně spojen s rozvojem samotných počítačů. Vývoj počítačů začal již počátkem devatenáctého století. Sice se nedá mluvit přímo o počítačích srovnatelných s dnešní dobou, ale i přesto se o nich zmíním. První „počítače“ pracovaly na principu takzvaných děrných štítků. Větší využití přišlo na konci 19.století, kdy za pomoci děrných štítků byl sčítán lid v USA. Poté Němec Konrád Zuse sestrojil nejprve stroje s označením Z1 a Z2 a poté i Z3, který byl z nich nejvyspělejší. Byl založen na principu mechanických relé, kterých bylo použito na 2600 kusů (Z3 byl zničen během bombardování Berlína v roce 1944) . Tento stroj byl používán např. na výpočet doletu raket. V roce 1943 byl vytvořen první elektronkový počítač. Byl nazván **Electronic Numerical Integrator and Computer** dále jen **ENIAC**. Následoval ho britský počítač s názvem **Colossus**, který byl využíván k luštění nacistických šifer během druhé světové války. Jeho nástupcem byl počítač pojmenovaný jako **MARK I**, který byl v provozu roku 1944 a následně byl vytvořen ještě jeho vylepšený typ pod označením **MARK II**. Roku 1957 vznikl programovací jazyk **Fortran**, který je mimochodem používán dodnes. Mezi jeho vlastnosti patřilo např. předávání funkcí jako parametrů, binární přístup k souborům či práce s poli. Koncem roku 1971 firma **Intel** představila svůj první mikroprocesor **4004**, který byl 4 bitový a pracoval na frekvenci 108 kHz a obsahoval 2300 tranzistorů. O rok později **Intel** představil jeho nástupce. Ten byl 8 bitový s frekvencí 0,2 MHz. Využití našly tyto procesory především v kalkulačkách či při řízení světelné signalizace. Teprve procesor řady **8080**, který byl vyroben v roce 1974, byl použit pro osobní počítač. Obsahoval asi 6000 tranzistorů a byl asi desetkrát rychlejší než jeho předchůdce, jelikož pracoval na frekvenci 2 MHz. Roku 1978 byl vydán procesor **8086**, který obsahoval již 26000 tranzistorů a pracoval na frekvenci 8 a 10 MHz. O rok později **Intel** vydal procesor řady **8088**, který byl 16 bitový a pracoval též na frekvenci 8 MHz. Roku 1982 byl **Intelem** představen procesor **80286**, který obsahoval 134 000 tranzistorů. Pracoval na frekvenci od 6 do 12 MHz a jeho rychlost byla asi šestinásobná oproti procesoru **8086**. Následník **80286** byl uveden roku 1985 a byl to 32 bitový procesor **Intel386 DX** s frekvencí od 16 do 33 MHz (nejrychlejší varianta byla uvedena až v roce 1989). Tento procesor byl už schopen adresovat 4 GB paměti, podporoval multitasking, obsahoval 275000 tranzistorů a byl asi 100x rychlejší než původní procesor **4004**. Procesor byl uveden

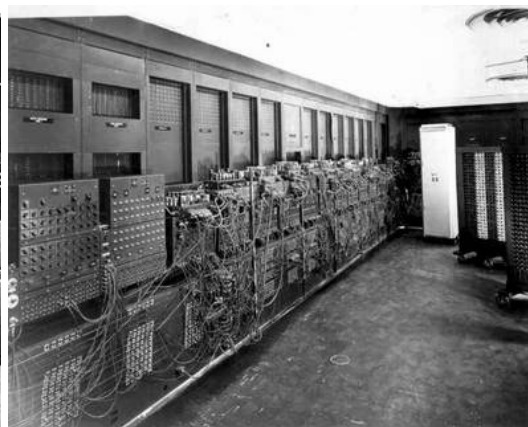
na trh i v jiných levnějších variantách a to buď varianta SX s 16 bitovou externí sběrnici a verze SL, která měla frekvenci od 20 do 25 MHz a byla určená pro přenosné počítače. Roku 1989 uvedla firma Intel na trh procesor s označením Intel486 DX. Ten byl asi 50x rychlejší než 8088 a obsahoval přes jeden milion tranzistorů. Frekvence byla později zvýšena až na 100 MHz, obsahoval také i koprocessor. Hlavními firmami vyvíjejícími procesory PC byly Intel, AMD a Cyrix. Firmu Cyrix později koupila firma IBM.

2 ENIAC

ENIAC (obrázek č.1) byl vytvořen pro potřeby americké armády. Jeho tvůrci jsou J. Presper Eckert a John William Mauchly z University of Pennsylvania. Projekt byl schválen roku 1943 a byl primárně určen pro balistické výpočty raket a pro projekt pod názvem Project PX. Jeho stavba byla zahájena v polovině roku 1944 na Penn's Moore School of Electrical Engineering. Hotový počítač byl pak představen veřejnosti 14. února roku 1946 na University of Pennsylvania. Jen pro informaci, projekt se vyšplhal z původních zhruba 62000 dolarů na neuvěřitelných 500 000 dolarů, což bylo na tehdejší dobu opravdu hodně. ENIAC zabíral plochu o rozměrech 162 metrů čtverečních, vážil 27 tun a spotřeboval ke svému chodu 160 kW. Obsahoval 17 468 elektronek, které byly "mozkem" celého stroje. Dále to bylo například 1500 relé, 70 000 odporů a 10 000 kondenzátorů. Jeho chod byl problematický již od počátku. Teprve později se podařilo docílit toho, že se vadná elektronka vyměňovala zhruba obden a ne jako na začátku, kdy se vlastně nedělalo nic jiného, než že se měnily vadné elektronky. Počítač pracoval v desítkové soustavě, na rozdíl od dnešních počítačů, které pracují v binární soustavě. Počítač neobsahoval uložené programy, ale přeprogramování se realizovalo „předrátováním“ (obrázek č.2). ENIAC dosahoval výpočetní rychlosti 5000 sčítání/odčítání za sekundu nebo 385 násobení za sekundu. ENIAC odstartoval zájem institucí a vědců o počítače a tak se později počítače objevují v dalších institucích či univerzitách. S vynálezem tranzistorů a integrovaných obvodů se pak začal vývoj ubírat k miniaturizaci a výsledkem je to, co teď vidíte před sebou.



obrázek č.1



obrázek č.2

3 von Neumannova architektura

Von Neumannova koncepce (někdy se též můžeme setkat s pojmem von Neumannovo schéma) byla vytvořena americkým vědcem maďarského původu Johnem von Neumannem. Nebyla to jen jeho zásluha. Šlo o týmovou práci. Von Neumannova architektura je představa o tom, jak by měl takový počítač fungovat. Z jakých částí by se měl skládat a co by jednotlivé části měly dělat. Razil zásadu, že vnitřní struktura by se neměla měnit podle zpracovávané úlohy, ale měla by být univerzální. Převratný byl názor, že data i programy by měly být uloženy v jedné paměti a ne odděleně, jak tomu bylo doposud. Ostatní koncepce vycházely totiž z předpokladu, že data i programy nelze slučovat. Ovšem John von Neumann přišel na to, že data i programy jsou v podstatě totéž (jen 0 a 1) a rozdíl mezi nimi je dán jejich významem. Ten jim dává ten, kdo s nimi pracuje. Při uchování dat v jedné paměti rozhoduje probíhající program o tom, jak interpretovat data, která najde v paměti, což má své výhody i nevýhody. Koncepce předpokládá, že každá strojová instrukce, ze kterých se skládají programy, se provádí postupně jedna za druhou. V podstatě se instrukce provede, až na ni přijde řada. Nevýhodou je sekvenční zpracování, což úplně vylučuje paralelismus. Další nevýhodou je i maximální univerzálnost a neměnnost vnitřní struktury, protože speciální počítače dokáží maximalizovat výkon pro danou aplikaci (např. šachové počítače).

3.1 Pět hlavních částí podle von Neumannovy architektury:

1. **Operační paměť:** Slouží k uchování zpracovávaného programu, dat a výsledků výpočtu.
2. **ALU:** Arithmetic-logic Unit (aritmeticko logická jednotka): Jednotka provádějící veškeré aritmetické výpočty a logické operace. Obsahuje sčítačky, násobičky (pro aritmetické výpočty) a komparátory (pro porovnávání).
3. **Řadič:** Řídicí jednotka, která řídí činnost všech částí počítače. Toto řízení je prováděné pomocí řídicích signálů, které jsou zasílány jednotlivým modulům. Reakce na řídicí signály, stavy jednotlivých modulů jsou naopak zasílány zpět řadiči pomocí stavových hlášení.

4. **Vstupní zařízení:** Zařízení určená pro vstup programu a dat.
5. **Výstupní zařízení:** Zařízení určená pro výstup výsledků, které program zpracoval.

3.2 Základní odlišnosti dnešních počítačů od von Neumannovy architektury:

1. Počítač může pracovat i s více než jedním procesorem.
2. Počítač může na rozdíl od von Neumannova schématu pracovat nejen pouze v tzv. diskrétním režimu.
3. Podle von Neumannova schématu počítač pracuje vždy jen s jedním programem. To vede k neefektivnímu využití strojového času. Je tedy obvyklé, že počítač zpracovává paralelně více programů zároveň - tzv. multitasking.
4. V dnešních počítačích existují navíc vstupně / výstupní zařízení která umožňují jak vstup, tak výstup dat (programu).
5. Program se do paměti nemusí zavést celý, ale je možné zavést pouze jeho část a ostatní části zavádět až v případě potřeby.

4 Generace počítačů

Počítače se rozdělují do tzv. **generací**, kde každá generace je charakteristická svou konfigurací, rychlostí počítače a základním stavebním prvkem. Generace počítačů (tabulka č.1):

Generace	Rok	Konfigurace	Rychlost (operací/s)	Součástky
0.	1940	Velký počet skříní	Jednotky	Relé
1.	1950	Desítky skříní	100 - 1000	Elektronky
2.	1958	do 10 skříní	Tisíce	Tranzistory
3.	1964	do 5 skříní	Desetitisíce	Integrované obvody
3. ¹ / ₂	1972	1 skříň	Statisíce	Integrované obvody
4.	1981	1 skříň	desítky milionů	Integrované obvody

Tabulka č.1

4.1 0. generace:

Historie vývoje samočinných počítačů se začíná odvíjet počátkem 40. let 20. století. V roce 1941 konstruuje v Německu Konrad Zuse malý reléový samočinný počítač **Zuse Z3**. Nedaří se mu však vzbudit pozornost armády, proto tento počítač upadá v zapomnění. Později je při jednom z náletů zničen. Rovněž ve Spojených státech se pracovalo na takovémto zařízení. V roce 1943 uvedl Howard Aiken z Harvardské univerzity do provozu svůj reléový počítač **Mark 1** sestrojený za podpory firmy IBM. Tento počítač byl pravděpodobně použit k výpočtům první atomové bomby.

4.2 1. generace:

První generace počítačů přichází s objevem elektronky, jejímž vynálezcem byl Lee De Forest a která dovoluje odstranění pomalých a nespolehlivých mechanických relé. Tyto počítače jsou vybudovány prakticky podle von Neumannova schématu a je pro ně charakteristický diskrétní režim práce. Při tomto zpracování je do paměti počítače zaveden vždy jeden program a data, s kterými se pracuje. Poté je spuštěn výpočet, v jehož průběhu již není možné s počítačem interaktivně komunikovat. Po skončení výpočtu musí operátor do počítače zavést další program a jeho data. Diskrétní režim práce se v budoucnu ukazuje jako nevhodný, protože velmi plýtvá strojovým časem. Důvodem tohoto jevu je "pomalý" operátor, který zavádí do počítače zpracovávané programy a data. V tomto okamžiku počítač nepracuje a čeká na operátora.

V této době neexistují vyšší programovací jazyky, z čehož vyplývá vysoká náročnost při vytváření nových programů. Neexistují ani operační systémy.

4.3 2. generace:

Druhá generace počítačů nastupuje s tranzistorem, jehož objevitelem byl John Barden a který dovolil díky svým vlastnostem zmenšení rozměrů celého počítače, zvýšení jeho rychlosti a spolehlivosti a snížení energetických nároků počítače. Pro tuto generaci je charakteristický dávkový režim práce. Při něm je snaha nahradit pomalého operátora tím, že jednotlivé programy a data, která se budou zpracovávat, jsou umístěna do tzv. dávky a celá tato dávka je dána počítači na zpracování. Počítač po skončení jednoho programu okamžitě z dávky zavádí program další a pokračuje v práci.

V této generaci počítačů také začínají vznikat operační systémy a první programovací jazyky, jako jsou COBOL a FORTRAN.

4.4 3. generace:

Počítače třetí a vyšších generací jsou vybudovány na integrovaných obvodech, které na svých čipech integrují velké množství tranzistorů. U této generace se začíná objevovat paralelní zpracování více programů, které má opět za úkol zvýšit využití strojového času počítače. Je totiž charakteristické, že jeden program při své práci buď intenzivně využívá CPU (provádí složitý výpočet), nebo např. spíše využívá V/V zařízení (zavádí data do operační paměti, popř. provádí tisk výstupních dat). Takové programy pak mohou pracovat na počítači společně, čímž se lépe využije kapacity počítače.

S postupným vývojem integrovaných obvodů se neustále zvyšuje stupeň integrace (počet integrovaných členů na čipu integrovaného obvodu).

5 Architektura (struktura) procesorů

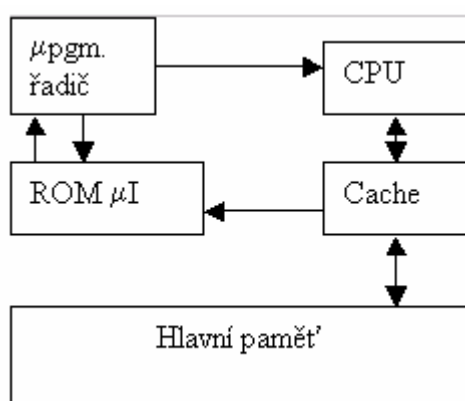
Mezi lidmi, kteří navrhují počítače, byl velmi dlouhou dobu rozšířen názor, že čím bohatší instrukční soubor bude mít procesor počítače, tím bude celý počítač výkonnější. Samotní výrobci se pak v podstatě jen předháněli, komu se podaří implementovat do hardwaru co nejvíce složitých strojových instrukcí a způsobů adresování a tím pádem nabídnout programátorům, co největší nabídku. Při tvorbě těchto instrukcí se přitom vycházelo z předpokladu, že při samotné tvorbě programu budou tyto instrukce využívány. Ale to, co mohlo být využíváno v době, kdy se programovalo převážně v assembleru, nemůže obstát v dnešní době. Z různých testů klasických programů vyplynulo, že domněnky výrobců jsou mylné. Z bohaté instrukční sady se totiž v praxi využívala jen úzká skupina instrukcí. Základní a velmi jednoduché instrukce. Ty složitější se téměř nevyužívaly nebo jen velmi zřídka.

Mít k dispozici silné nástroje, ale využívat je jen minimálně, by samo o sobě nemuselo být na závalu. Otázkou je, zda to není zbytečný luxus. Jestli je vůbec potřebné mít složité instrukce a zda tyto instrukce nekomplikují použití těch jednoduchých, které se opravdu využívají. A právě zde je vidět jádro problému. Procesor, který obsahuje všechny instrukce, jak složité tak i ty jednoduché a umožňuje různé druhy adresování, musí být tedy hodně komplikovaný a ve výsledku i pomalý. Vykonání i těch nejjednodušších instrukcí musí zákonitě u takového procesoru trvat déle než u procesoru, který bude obsahovat mnohem menší (jednodušší) instrukční soubor.

Když si výrobci procesorů uvědomili tuto skutečnost, tak je napadlo, že by se mohly instrukční soubory „ořezat“ o ty složité instrukce a zůstaly by tak pouze jednoduché instrukce s těmi nejnutnějšími způsoby adresování. Tyto instrukce by se pak prováděly co nejrychleji. Toto je základní myšlenka architektury označované jako **Reduced Instruction Set Computer** dále už jen jako **RISC**. Tedy šlo o redukovaný instrukční soubor. Tato koncepce se netýká pouze jen redukovaného instrukčního souboru, ale je to v podstatě představa, jak by měl samotný počítač fungovat. Původní koncepce počítačů se složitým instrukčním souborem se označuje jako **Complex Instruction Set Computer** dále už jen jako **CISC**.

5.1 CISC

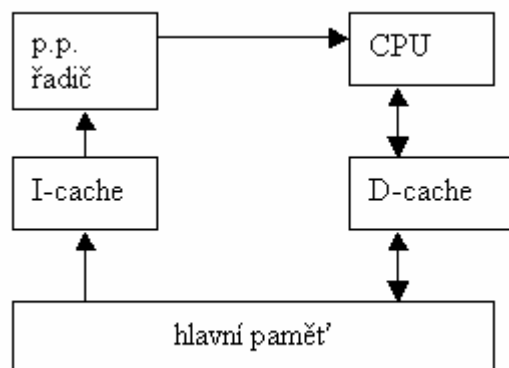
Procesory CISC obsahovaly složité instrukce. Tyto instrukce sloužily k tomu, aby pro celou operaci stačilo načtení jen jedné instrukce. Řadič procesoru je navrhnut na základě řídicí paměti, která uchovává mikroprogramy jednotlivých strojových instrukcí. Vykonání strojové instrukce probíhá jako posloupnost mikrooperací. Posloupnost mikroinstrukcí je dána v řídicí paměti, která je mnohem rychlejší než paměť operační. S postupným rozvojem miniaturizace elektronických obvodů se řídicí paměť integrovala přímo do procesoru spolu s vyrovnávací pamětí (cache).



Obrázek č.3

5.2 RISC

Program se skládá z většího množství jednoduchých instrukcí. Vývoj RISC ukázal, že není důležitý počet instrukcí, ale jejich jednoduchost.



Obrázek č.4

5.3 VLIW

Very Long Instruction Word dále jen jako VLIW. Tato architektura přesouvá většinu problémů z hardware na software. Překladač programu po vygenerování instrukcí typu RISC by vyhledal na sobě nezávislé instrukce a nashromáždil je do bloků. Např. pro 16 bitový procesor VLIW by byl každý blok tvořený 16 instrukcemi. Vykonává se vždy blok za blokem tedy vždy 16 instrukcí najednou. Netestuje se jejich vzájemná závislost, tu už vyřešil překladač.

5.4 Jak to dnes vypadá

Dnes už se rozdíl mezi RISC a CISC začínají vytrácet. V budoucnu můžeme očekávat, že obě architektury budou čím dál více splývat a z každé se vezme to nejlepší.

6 Jak takový procesor pracuje

Procesor nedělá v podstatě nic jiného, než že zpracovává instrukce. Cesta zpracování instrukcí začíná od operační paměti, kam programátor program uloží a poté dá mikroprocesoru příkaz, aby instrukce začal vykonávat. Instrukce se nejdříve přesunou z RAM paměti do L2 cache paměti (u procesorů řady P4 se nejdříve instrukce přesune do L3 cache a poté teprve do L2). Přesun instrukce mezi paměťmi se vykonává přes **Front Side Bus** dále již jen jako FSB (u Athlon 64 je nahrazeno technologií HyperTransport). Poté se instrukce přesune do L1 cache paměti, která je implementována přímo v procesoru a je rozdělena na část pro instrukce a část pro data. Data jsou zpracovávána skrze pipeline, tím je dosaženo vyšší rychlosti zpracování, než kdyby se zpracovávala jen jedna instrukce. Instrukce se dekoduje a určí se její typ. Poté se načtou adresy operandů, se kterými se bude pracovat. Nakonec **Execution Unit** (dále již jen jako EU) vykoná instrukci a výsledek je opět zapsán do L2 cache paměti. Pro urychlení zpracování se využívají již zmíněné pipeline. Dnes již každý procesor obsahuje více těchto pipeline a je proto označován jako superskalární. Procesor dále umí vykonávání instrukcí mimo pořadí a s tím související odhad výsledné větve při vykonávání podmínek-Branch Prediction.

7 ARCHITEKTURA ČIPU

7.1 Pipeline

Jde o zřetěžené zpracování instrukcí vycházející ze skutečnosti, že zpracování každé instrukce procesorem lze rozdělit do fází. Vysvětlím na původním procesoru Pentium. Jeho pipeline měla pět částí:

1. Prefetch (PF) – výběr instrukce, další zpracovávaná instrukce se bere buď z RAM paměti nebo z vyrovnávací paměti cache
2. Decode 1 (D1) – dekódování instrukce, určí se typ instrukce a její délka
3. Decode 2 (D2) – výpočet adresy, na které jsou operandy, s nimiž se bude pracovat
4. Exekute (EX) – vlastní provedení instrukce
5. Write Back (WB) – zapíše se výsledky zpracované instrukce

Každou z těchto fází může provádět samostatně pracující jednotka a v okamžiku, kdy je tato jednotka se svou prací hotova, předá svůj výsledek jednotce provádějící následující fázi zpracování a pokračuje ve své práci nad další instrukcí (tabulka č.3).

	Prefetch	Decode 1	Decode 2	Execute	Write Back
Cyklus 1	Instrukce 1				
Cyklus 2		Instrukce 1			
Cyklus 3			Instrukce 1		
Cyklus 4				Instrukce 1	
Cyklus5					Instrukce 1
Cyklus 6	Instrukce 2				
Cyklus 7		Instrukce 2			
Cyklus 8			Instrukce 2		
Cyklus 9				Instrukce 2	
Cyklus10					Instrukce 2

Tabulka č.3

Když je pipeline zpracování programu spuštěno, průběh vypadá takto (tabulka č.4):

	Prefetch	Decode 1	Decode 2	Execute	Write Back
Cyklus 1	Instrukce č.1				
Cyklus 2	Instrukce č.2	Instrukce č.1			
Cyklus 3	Instrukce č.3	Instrukce č.2	Instrukce č.1		
Cyklus 4	Instrukce č.4	Instrukce č.3	Instrukce č.2	Instrukce č.1	
Cyklus 5	Instrukce č.5	Instrukce č.4	Instrukce č.3	Instrukce č.2	Instrukce č.1

Tabulka č.4

7.1.1 Klasické zpracovávání instrukcí

Při klasickém zpracování dat je vidět, že se v prvních pěti cyklech (taktech) zpracuje pouze jedna instrukce. Po jejím úplném zpracování začne procesor zpracovávat druhou instrukci, což mu bude opět trvat pět strojových cyklů.

7.1.2 Zřetěžené zpracovávání instrukcí

Při zřetěženém zpracování dat je sice v prvních pěti cyklech zpracována kompletně jen jedna instrukce, ale druhá instrukce je jeden cyklus před dokončením, třetí instrukce potřebuje ještě dva cykly atd. Takže v každém dalším taktu je zpracována vždy jedna instrukce. Po deseti cyklech je tedy zpracováno kompletně šest instrukcí oproti dvěma při klasickém zpracování.

7.1.3 Nevýhody pipeline

Pipelining má i své nevýhody. Pokud je jedna instrukce datově závislá na jiné instrukci (jedna instrukce má k sobě data, která jsou výsledkem instrukce jiné), pak tuto instrukci nelze vykonat do té doby, dokud nebude znám výsledek první instrukce, což ovšem při tomto způsobu zpracování dat může trvat i několik cyklů, takže čím delší pipeline je, tím častěji se tyto problémy mohou vyskytnout. Tím pádem vznikají v pipeline volná místa což má za následek snížení výkonu.

I když delší pipeline přináší problémy, tak bezesporu má i své výhody. Platí totiž: čím delší pipeline, tím vyšší frekvence lze dosáhnout, proto firma Intel při dosažení vysokých frekvencí má delší pipeline než AMD. Procesor Pentium 4 má 27 stupňovou, ale Athlon 64 jen 12 stupňovou pipeline. Opět platí, že ne vždy vyšší frekvence znamená i vyšší výkon procesoru.

7.2 Superskalár

Neustálé zvyšování frekvence bylo při 486kovém způsobu vykonávání instrukcí příliš neefektivní. Při zvyšující se frekvenci se zvyšovaly požadavky na přísun dat. Výrobní technologie procesorů byla drahá a tak mívaly malou vyrovnávací cache paměť. To způsobovalo, že i při zvýšení frekvence na dvojnásobek nemuselo dojít ke zvýšení výkonu, ale pouze k tomu, že procesor musel čekat více hodinových cyklů na pomalou RAM paměť. Ovšem byly i případy, kdy zvýšení frekvence přineslo i razantní zvýšení výkonu, frekvenci nešlo ale zvyšovat donekonečna.

Přišlo se na to, že kdyby se do procesoru přidala další ALU, mohly by se výpočty začít provádět paralelně. Tím by se výkon v některých aplikacích značně zvýšil. Poprvé to bylo vyzkoušeno u procesoru Intel Pentium Pro (obsahoval dvě ALU). Instrukce byly i tak vykonávány v předem daném pořadí, ale občas se vykonaly paralelně. Dnešní procesory obsahují také více ALU jednotek.

7.3 Out-of-order a spekulativní vykonávání instrukcí

Vykonávání instrukcí v určitém pořadí má své nevýhody. Firmou Intel byla navržena architektura P6 pro procesor Intel Pentium Pro. V procesoru byla také integrovaná cache paměť přímo k výpočetní části a uměl také out-of-order execution neboli zpracování instrukcí mimo pořadí. Procesor pracuje s malým počtem instrukcí zvaných microOps (někdy jen Ops). Vykonávání instrukcí v určitém pořadí má své nevýhody. Na začátku pipeline jsou dekodéry, které instrukce přeloží na microOps. Jádro procesoru si tyto microOps organizuje dle svých potřeb a vykonává je mimo pořadí stanovené programátorem. Jádro procesoru si tyto microOps organizuje dle svých potřeb a vykonává je mimo pořadí stanovené programátorem. Nárůst výkonu Out-of-order proti in-order je velmi značný.

7.4 Vykonávání podmínek

Pentium Pro bylo též vybaveno vlastností vykonání instrukce po podmínce. Podmínky v programech mají tendenci snižovat výkon. V případě splnění podmínky se provádí kód A. V případě nesplnění podmínky se vykoná kód B. Výsledek podmínky bývá většinou řešen těsně před samotnou podmínkou a procesor její výsledek tudíž nezná. Dochází tak k brždění superskalárního vykonávání instrukcí. Pentium Pro ale umí z 90 % odhadnout výsledek podmínky (Branch Prediction) a tak si již předem zpracovává příslušnou část kódu a zároveň je zpracováván kód před podmínkou. V případě, že je podmínka odhadnuta správně, procesor má již většinu kódu. Pokud se splete musí již spočtené výsledky vymazat a začít počítat kód nový. Než se tak stane, vznikne časová prodleva, jejíž velikost je dána délkou pipeline. Tato časová prodleva se nazývá misprediction penalty. Některé procesory jsou navrženy tak, aby vykonávaly obě větve po podmínce s tím, že jednu z nich po vyhodnocení výsledku podmínky „zapomenou“.

8 Cache paměti

S cache pamětí se v počítačové architektuře můžeme setkat poměrně často. Je to v podstatě mezisklad dat mezi komponentami počítače, jejichž účelem je kompenzace rychlostí mezi nimi. Rychlejší komponenta si data načte z cache namísto toho, aby čekala na pomalejší zařízení.

8.1 Paměť cache L1 (first level cache)

Přímo do samotného procesoru jsou dnes implementovány malé paměti označované jako L1 cache. Je využívána k zásobování jednotek procesoru daty ze sběrnice. V podstatě jde o to, že cache si načte více dat ze sběrnice, které pak uchovává. Jakmile procesor tyto data potřebuje vytáhne si je, a protože cache je nesrovnatelně rychlejší než např. sběrnice nemusí procesor na data dlouze čekat.

8.2 Paměť cache L2 (second level cache)

U většiny dnešních procesorů je implementována přímo v procesoru. Dříve byla umístěna přímo na desce. L2 cache slouží jako paměť mezi operační pamětí počítače a pamětí L1. L2 má velikost od 128kB do 2MB u nejmodernějších procesorů. Je pomalejší než L1.

9 Instrukční sady

Jde o soubor instrukcí za pomoci kterých se vytváří programy. CPU vykoná pouze příkazy, které spadají do jemu známých instrukčních sad, pokud je použita instrukce, která není součástí instrukční sady CPU, dojde k chybě.

9.1 Single Instruction Multiple Data (dále již jen jako SIMD)

Používá se pokud máme provést jeden výpočet na více datech pomocí jedné instrukce. Využití např. u multimediálních aplikací, kde je možné zpracovávat několik částí audia současně nebo uvedu praktičtější příklad. Pokud má procesor přičíst k pěti číslům jedničku má možnost to provést s každým číslem zvlášť a výsledek uložit zpět do paměti. To by bylo ale velice zdlouhavé a neefektivní. Může ale využít speciální vektorovou jednotku, která v jednom okamžiku naplní čísla do registrů a v jednom cyklu k nim přičte jedničku.

9.2 x86

Základní celočíselná sada procesorů pro počítače PC.

9.3 x87

Sada pro operaci s plovoucí desetinnou čárkou (Floating point - FP). Určená pro matematické koprocesory. Starší procesory 486SX či 386 vyžadovaly pro podporu x87 pořízení dalšího čipu (externího matematického koprocesoru). Všechny procesory (kompatibilní s x86) počínaje Pentiem mají koprocesor integrovaný a sadu x87 podporují.

9.4 MultiMedia eXtensions (dále již jen jako MMX)

Tato instrukční sada se objevila v roce 1995 a přinesla rozvoj procesorů. Tyto instrukce byly poprvé využity v procesoru Pentium MMX 166MHz. Procesory s MMX obsahují navíc 57 instrukcí. Jak již název vypovídá byly využity především v multimediálních aplikacích jako např. komprese a dekomprese zvukových i obrazových dat, syntéza hlasu či výpočet 3D scény. Dříve tyto instrukce obsahovaly pouze procesory značky Intel, ale dnes tyto instrukce nechybí u žádného procesoru.

9.4.1 Enhanced MMX

Rozšíření MMX zavedené společností Cyrix. Podporuje ho většina výrobců kromě Intelu.

9.5 3DNow!, Enhanced 3DNow!

Po MMX přišla firma AMD s instrukcemi 3DNow! (obsahují 21 nových instrukcí). Slouží pro urychlení 3D grafiky, převážně ve hrách. Je ale nutné, aby hra tyto instrukce podporovala. Kromě 3DNow! ještě existuje enhanced 3DNow! (rozšířené 3DNow!).

9.6 Streaming SIMD Extensions (dále již jen jako SSE)

Instrukční sada představená Intelem roku 1999 jako odpověď na 3DNow!. Poprvé využita procesorem Pentium III. Původně bylo SSE známo jako **Katmai New Instructions** (dále již jen jako KNI). Katmai bylo původně krycí jméno pro revizi jádra Pentia III. I AMD nakonec přidalo SSE do procesorů počínaje Athlon XP. SSE přidává osm nových 128 bitových registrů označovaných jako XM0 – XM7. Každý registr je schopen pojmout čtyři 32 bitová čísla typu single s pohyblivou řadovou čárkou.

9.6.1 SSE2

Je prvně představen Intelem roku 2001 pro procesor Pentium 4. SSE2 je v podstatě rozšířená SSE a má za úkol plně nahradit MMX. Mnoho programů je schopno využít potenciál SSE2 na maximum. Po MMX se jedná o druhou nejpoužívanější instukční sadu. Firma AMD přidala podporu SSE2 do svých procesorů počínaje Opteronem a Athlonem 64.

9.6.2 SSE3

Malé rozšíření SSE2 asi o 10 instrukcí. Bylo známo pod krycím jménem **Prescottt New Instructions** (dále jen jako PNI). SSE3 bylo využito především v Pentium 4 s jádrem Prescottt.

9.7 AMD64 (x86-64 nebo x64)

Je 64 – bitová mikroprocesorová architektura s odpovídající instrukční sadou vytvořenou firmou Advanced Micro Device (dále již jen jako AMD). Jde tak trochu o vylepšenou x64 architekturu. AMD64 je podporován procesory Athlon 64, Athlon 64 FX, Athlon 64 X2, Turion, Opteron a v poslední době i procesory Sempron 64.

AMD x86 – 64 bitová instrukční sada (dále jen jako AMD64) je vlastně rozšíření x86 – 32 bitové architektury. Hlavní charakter AMD64 je podpora 64 bitových registrů, aritmeticko logických operací s celými 64 bitovými čísly a 64 bitovými virtuálními adresami. Byly vylepšeny i další věci jako např. :

9.7.1 Full support for 64-bit integers

Plná podpora pro 64 bitová celá čísla. Všechny univerzální registry (GPRs) jsou rozšířené z 32 na 64 bitů.

9.7.2 Larger virtual address space

Současné procesory využívající AMD64 mohou adresovat až 2^{48} bytů virtuálního adresového prostoru. V budoucnu může být tento limit zvýšen až na 2^{64} bytů. Pro srovnání 32 – bitový x86 umí adresovat jen 2^{32} bytů.

9.7.3 Larger physical address space

AMD64 je v současné době schopno adresovat až 2^{40} bytů. V budoucnu budou možné až 2^{52} bytů. Aktuální 32 bitové adresování umožňuje jen přístup maximálně k 2^{36} bytů.

9.8 EM64T

Procesory Intel Xeon s jádry Nacona a Irwindale přinesly další inovativní technologii, spočívající v rozšíření adresace na 64 bitů. Šlo přitom o podobný proces jako když se přecházelo z 16-bitové na 32-bitovou adresaci, tj. prodloužením délky registrů. Stávající architektura je totiž schopná přímo adresovat pouze 4GB. Přitom je ale využíváno nejvyššího bitu, který je obecně chápán jako znaménkový a tudíž se pak pracuje s rozsahem -2 GB až +2 GB. To neznámá, že nelze adresovat prostor větší (v minulosti byly 16-ti bitové procesory schopny adresovat více než jen 64kB paměti).

Jestliže máme procesor, schopný vykonávat bez úpravy běžný 32-bitový kód, a přitom používat 64-bitové rozšíření, můžeme pracovat ve třech režimech (Tabulka č.2):

Legacy režim	Režim kompatibility	64-bitový režim
Kompatibilita existujících 32-bitových aplikací, nárůst výkonu	Nárůst výkonu pro vybrané 32-bitové aplikace díky rozšířené adresovatelnosti paměti (až 4GB na aplikaci)	Nárůst výkonu pro paměťově a výpočetně náročné aplikace, pokud jsou optimalizovány pro 64-bitovou platformu
Požadavky	Požadavky	Požadavky
• 32-bitový OS • 32-bitové aplikace • 32-bitové ovladače	• 64-bitový OS • 32-bitové aplikace • 64-bitové ovladače	• 64-bitový OS • 64-bitové aplikace • 64-bitové ovladače

Tabulka č.2

Tyto tři režimy umožňují přechod 32 bitů k 64 bitům a přitom není nutná žádná investice do dodatečného HW. V 32-bitovém režimu není nijak degradován výkon současných aplikací, po přechodu na 64-bitový operační systém není nutné upgradovat najednou všechny aplikace a přechod samotný už přináší více prostoru pro jednotlivé 32-bitové aplikace, a dokončení přechodu na 64-bitové aplikace zhodnotí plný potenciál platformy.

EM64T byla poprvé využita procesorem Intel Pentium s jádrem Prescott v revizi E, který byl podporován čipovou sadou i915P (Grantsdale) a i925X (Alderwood). EM64T byl odpovědí firmy Intel na konkurenční AMD64, i když ve skutečnosti šlo v podstatě o kopii, která měla pouze malé úpravy. OD března roku 2006 podporuje každý procesor firmy Intel EM64T. První mobilní procesor, který toto bude podporovat bude Core Duo s jádrem Merom, které má být vydáno v srpnu 2006. I při využití 64 bitového systému je možné samozřejmě využívat i 32 bitové aplikace.

10 Virtualizace

Tento pojem je v oblasti IT známý již dlouhou dobu. Své využití nachází především v oblasti počítačového HW. Konkrétně jde o oblast serverů a úložišť dat. Dnes se tato technologie uplatňuje i v procesorech, kde firmy jako Intel či AMD představují virtualizační řešení již přímo na úrovni procesorů.

10.1 Virtualizace u serverů

Virtualizace serverů přišla v době sálových počítačů. Podstatou bylo, co nejlepší rozložení využití všech serverů. Koncem roku 1990 se objevila virtualizace i na poli operačních systému Windows a UNIX. Vznikla tak možnost provozovat více „virtuálních počítačů“ s různými operačními systémy na jednom počítači. Pozdější snaha o dosažení vyššího výkonu a využití HW prostředků u serverů vedla k vývoji technologií a nástrojů jako např. partition manager, virtual machine, virtual partition nebo logical partition.

Dnešní servery disponují velmi vysokým výkonem, ale ne vždy jsou plně vytiženy, nabízí se možnost sdílení HW. Více virtuálních strojů a někdy i celá virtuální síť může být realizována jedním serverem.

Možným problémem jsou bezpečnostní aspekty. Existují totiž obavy, že pád jedné aplikace může způsobit zhroucení celého systému, a tak bývá jedna aplikace spuštěna jen na jednom serveru, což je ale zbytečné plýtvání. Tyto obavy mají své opodstatnění, ale každá technologie má své „mouchy“.

10.2 Virtualizace pro desktop

Firmy AMD i Intel přišly s vlastní koncepcí virtualizace. Firma AMD ji dala kódové jméno Silverdale, nyní Pacifica. Výrobce používá zkratku SVM (Security and Virtual Machine). Firma Intel nazvala svou technologii Vanderpool, zkráceně VT. Na rozdíl od AMD již procesory firmy Intel tuto technologii podporují. Jde např. o procesory modelových řad 662 a 672, Pentia D a Pentia EE. V těchto procesorech se nalézá pod názvem VT-x (x je podle x86).

OS má za úkol oddělit aplikaci od HW. Při využití více OS je nutné jim zabránit v komunikaci s HW. Přitom je ale nutné, aby si OS myslel, že danou instrukci vykoná. Tu vykoná hypervisor (nadřazený systém) a provede příslušné akce. V klasické intelovské architektuře běží aplikace v uživatelském režimu a

operační systém v privilegovaném režimu. Existují privilegované instrukce, které se smí provádět pouze v privilegovaném režimu. Při pokusu vykonání na nižší úrovni dojde buď k ukončení aplikace nebo k vykonání na vyšší úrovni a poté dojde k předání řízení zpět aplikaci.

10.3 Software pro virtualizaci

Při vytvoření několika virtuálních serverů na jednom počítači je potřeba nainstalovat potřebný software, který ošetří výše zmiňovaný problém s pádem aplikace. SW musí zajistit, aby při pádu byl zasažen pouze daný virtuální server. Ostatní servery by neměly být ovlivněny.

Virtualizační technologie vytvoří virtuální hardwarové prostředí, ve kterém pak může běžet další operační systém zcela nezávisle na hostujícím systému.

Firmy zabývající se virtualizační technologií:

- **VMware** - tuto technologii lze provozovat na platformách Windows dostupný i ve verzi pro Linux a obsahuje různé virtualizační prostředky
- **Microsoft Virtual Server** - zástupce této kategorie virtualizační technologie.

10.4 Jak to funguje ?

Pokud chceme zavést koncept virtuálního stroje, musíme dosáhnout toho, že privilegovaný kód operačního systému poběží ve skutečnosti v uživatelském režimu, privilegované instrukce jsou zachyceny a emulovány. O emulaci HW prostředí a komunikaci s hypervisorem se stará VMM (**V**irtual **M**achine **M**onitor). Architektura x86 u některých instrukcí ale používá v různých režimech nevirtualizovatelné instrukce. To lze vyřešit pomocí přístupu ke zdrojovému kódu. Tuto možnost nám dovolují pouze některé OS a navíc úpravy jsou složité.

Společnost Intel pro usnadnění uvedení virtualizace zavedla technologii označenou jako Vanderpool, nyní její oficiální označení zní Intel Virtualization Technology. Tato HW virtualizace zavádí novou úroveň, která je nadřazená privilegované úrovni. Tím je umožněna jednodušší implementace virtuálního stroje a jejich bezpečnější.

10.5 Využití

Např. u administrátora, který na počítačích uživatelů sítě bude moci provozovat virtuální počítač s jehož pomocí se dostane k počítači uživatele, který bude mít zhroucený systém. Je-li potřeba vyzkoušet nějakou aplikaci na různých platformách, je možnost to vyzkoušet na jednom počítači s využitím virtualizace. Bude-li počítač (server) dostatečně výkonný můžeme na něm, jak vyvíjet samotnou aplikaci tak ji i zkoušet. Je možné testovat i síťové aplikace s využitím jediného počítače.

11 Parametry

11.1 HyperThreading (HTT)

Koncept HTT vychází z toho, aby se jeden fyzický procesor byl schopen tvářit a pracovat jako dva logické procesory. Toho dosáhneme když „zdvojíme“ to, co určuje aktuální stav jednoho výpočetního procesu. Jedná se o registry a stavový vektor. Poté je možné využít toho, že když jedno vlákno čeká na data (řekněme z hlavní paměti) tak se procesor přepne během jednoho taktu hodin na vlákno druhé a pracuje dál. To umožňuje zlepšit využití výpočetních částí jádra, což přináší vyšší výkon. Při využívání více vláken najednou může také nastat problém s prioritou vláken, což může mít za následek pokles výkonu, protože méně důležitá aplikace pracuje na úkor důležitější aplikace. Pro lepší využití této technologie je nutná i SW podpora, kterou ne všechny programy nabízejí. Pokud jsou aplikace speciálně přizpůsobené, fungují samozřejmě i na procesoru bez podpory HTT.

V dnešní době ale tato technologie už ztrácí na významu, protože nastupují dvou jádrové procesory a ty již obsahují dva fyzické procesory .

11.2 HyperTransport (HT)

Oproti procesorům firmy Intel, které komunikují s NorthBridge přes FSB (jde o paralelní přenos dat) využívá firma AMD HyperTransport. Má jednoduchý návrh, vysokou propustnost a nízké latence. Základní frekvence sběrnice je 200 MHz. Tato frekvence se ale ještě zvyšuje pomocí HT násobiče. Šířka sběrnice je 2, 4, 8, 16 nebo 32 bitů.

11.3 Front Side Bus (FSB)

Obousměrná sběrnice spojující jádro procesoru se zařízeními systému jako např. RAM pamětí atd. Je velmi rychlá (např. 533, 800, 1066 MHz) a bývá velmi vytížená.

11.4 Base frequency, source clock (základní frekvence)

Frekvence generovaná frekvenčním generátorem. Slouží jako základ pro získávání dalších frekvencí.

11.5 Latence

Doba, která uplyne mezi vyvoláním požadavku a jeho splněním. Tato doba je velmi krátká. Udává se v nanosekundách či hodinových cyklech. Čím je tato hodnota nižší, tím procesor může rychleji zpracovávat instrukce.

11.6 Thermal Design Power (TDP)

Udává hodnotu tepelného výkonu. Čím vyšší tato hodnota je, tím vyšší jsou nároky na spotřebu a nároky na chlazení. TDP udává každá společnost jinak. Např. firma AMD udává maximální dosažitelný tepelný výkon, ale naproti tomu firma Intel uvádí TDP jako typicky dosažitelný tepelný výkon. Rozdíl mezi oběma značeními je velmi velký. U firmy AMD není takřka možné hodnoty TDP dosáhnout kdežto u procesů Intel je TDP voleno jako cílový chladičí výkon. Pokud dojde k jeho překročení pak je aktivována ochrana Thermal Monitor a ta sníží výkon.

12 Výrobní technologie

Tzv. "feature size", tj. velikost transistoru. Čím menší je tranzistor, tím rychleji dokáže přepínat, ale také potřebuje menší napětí, což má velký vliv na spotřebu. Čím menší je použitá technologie, tím více se začíná projevovat leakage current, což je stálá (statická) neovlivnitelná spotřeba. Větší odolnost proti leakage current mají pomalejší tranzistory což má ale za následek snížení výkonu.

12.1 Používané materiály:

12.1.1 Bulk Silicon – jde o klasický křemík s extrémní čistotou. Je určen pro výrobu polovodičů

12.1.2 Strained Silicon (SS) – klasický křemík obohacený o germanium. To snižuje odpor. Oproti Bulk Silicon umožňuje rychlejší přepínání tranzistorů (při stejném napětí) a tím pádem dosažení vyšších frekvencí.

12.1.3 Silicon in Insulator (SOI) – na vrstvě oxidu je navařen klasický křemík. Oxid slouží jako izolant. Díky SOI je možné snížit spotřebu a leakage current. Je dosahováno i vyšších frekvencí.

12.1.4 Strained Silicon Directly on Insulator (SSDOI) – doposud nejlepší výrobní technologie, která byla vyvinuta firmou AMD ve spolupráci s firmou IBM

13 Spotřeba procesorů

Spotřeba procesorů byla vždy problém. S rostoucím výkonem rostla i spotřeba energie. Např. u zastaralého Pentia Pro v porovnání s 486 DX2 66MHz byla spotřeba 2,5 krát vyšší, ale bylo dosaženo dvounásobného zvýšení výkonu.

Dalo by se předpokládat, že nejvíc energie spotřebují výpočetní části (ALU,FPU). Opak je ale pravdou. ALU brala 6% a FPU jen 5% celé spotřeby procesoru. Naopak predikce větvení a out-of-order vykonávání instrukcí byly energeticky dost náročné. Největší spotřebu měl přísun instrukcí a jejich dekódování, datová cache, buffery pro určení vhodné sekvence zpracování instrukcí a paměťová pole registrů.

Spotřebu se dařilo redukovat díky stále se zmenšující výrobní technologii (při přechodu z 0,18 μm na 0,13 μm klesla spotřeba na zhruba 70% původní hodnoty). Čím menší výrobní technologie, tím menší byly i tranzistory (na stejnou plochu tak bylo možné umístit více tranzistorů) a potřebovaly i menší napětí (snížení spotřeby).

13.1 Vlastnosti redukující spotřebu

13.1.1 PState (Performance State) – Redukce spotřeby za pomoci snížení frekvence a napětí jádra. Snížení napětí má za následek snížení výkonu, ale také frekvence.

13.1.2 Intel SpeedStep – Poprvé využita v procesoru Pentium III, jde o první úspornou technologii. Jedná se o softwarově ovládaný způsob přes I/O, což není příliš efektivní. Obsahuje pouze dva PState. Přechod mezi PState je pomalý. Dnes jde již o zastaralý způsob.

13.1.3 Enhanced Intel SpeedStep s více PStates (EIST) - Nejnovější způsob použitý u procesorů Pentium M, Pentium 4 s jádrem Prescottt a Pentium D. Většinou má více Pstate (např. 6). Změna napětí je realizována za chodu. To je velmi rychlé a účinné.

13.1.4 Demand Clock Modulation (ODCM) – Pro snížení spotřeby vkládá systém prázdné takty. Vložením prázdného taktu dojde ke snížení výkonu.

13.1.5 PowerNow! verze 1.4 – změna napětí procesu je realizována za plného chodu. Zatím nejdokonalejší technologie využívaná procesory AMD Athlon 64.

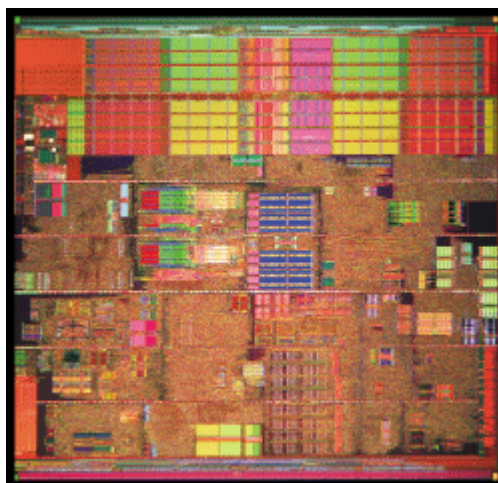
13.1.6 Cool'n'Quiet – jde o regulaci otáček ventilátorů v kombinaci s PowerNow! verze 1.4. Tím je dosaženo jak snížení spotřeby, tak i hluku.

14 DualCore procesory (2 procesory v jednom „kabátě“)

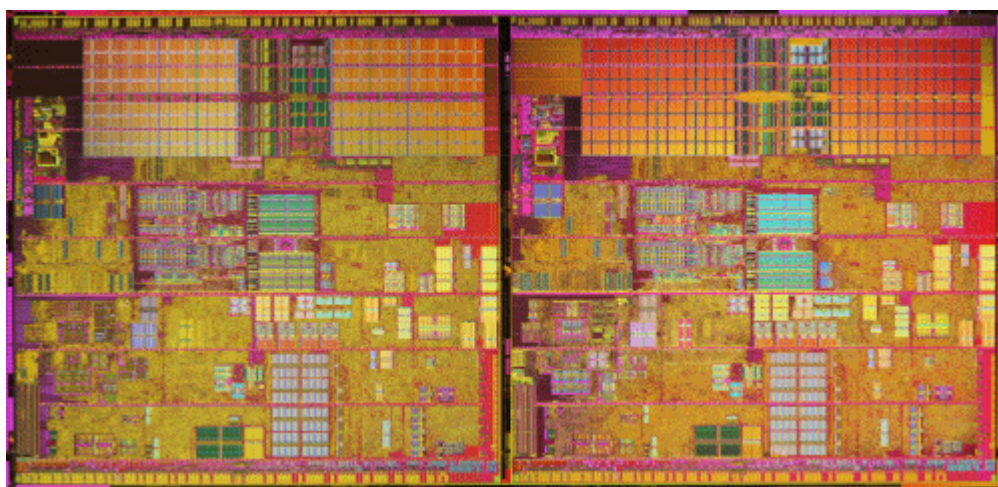
14.1 Návrh

Při návrhu procesorů je stále kladen důraz na zvyšování výkonu, ale zároveň na co nejmenší náklady. Současné jedno jádrové procesory naráží na problém, jak zvyšovat výkon, když už se naráží na frekvenční maximum, proto je potřeba nový pohled na vývoj procesorů. Řešením je umístění dvou čipů (obrázek č.5) do jednoho (obrázek č.6).

Klíčem úspěchu dvou jádrových CPU či dvou procesorových PC jsou aplikace využívající více threadů a multitaskingové operační systémy (např. Windows). Dnešní procesory jsou schopny zpracovávat jen jedno vlákno (vyjma procesorů s HT).



obrázek č.5



obrázek č.6

14.2 Co je to Thread

Např. u počítačové hry počítáme fyzikální model postav a jejich inteligenci, což jsou velmi nezávislé výpočty. V případě superskalárních procesorů je to ideální stav. Teoreticky bychom mohli počítat oboje najednou. Vytížit jednotky a dosáhnout skvělého výkonu. Zde se dá představit, že fyzikální model bude jeden thread (vlákno) a inteligence druhý thread.

14.3 Multithreading

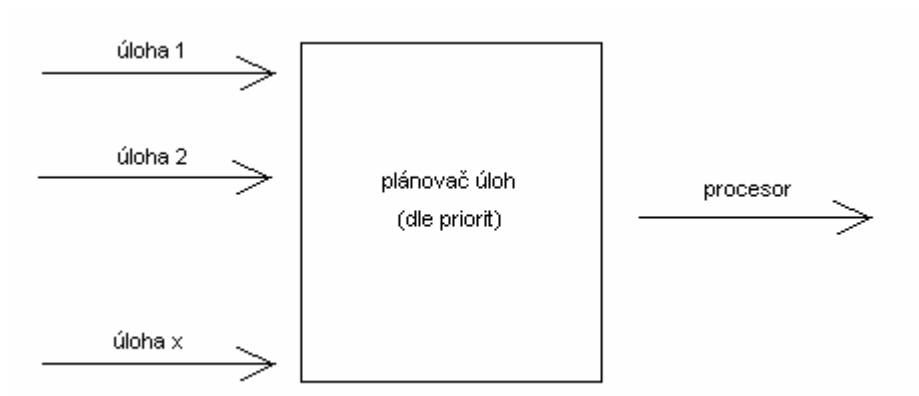
Pointa ve využití threadů je tedy v tom najít nezávislé výpočty, buď v rámci jednoho nebo více programů. Např. když hrajeme hru a máme přitom spuštěný antivirový program. DualCore tak v některých případech může zaznamenat např. 80 procentní zvýšení výkonu. Při využití multithreadingu je tedy zapotřebí synchronizace. Ta ale potřebuje výpočty, které trochu snižují výkon

14.4 Problémy

Využití práce s více thready má své problémy. Už při vytvoření jen dvou vláken se může stát, že první vlákno dokončí svoji úlohu mnohem dříve než vlákno druhé nebo naopak moc pozdě. Vzniká tedy problém se synchronizací. Ta spočívá v tom, že vlákno můžeme občas pozastavit a poté znovu spustit nebo vložit prázdné cykly. Problém je ale v naprogramování synchronizace, protože velice často nastávají neočekávané situace. Programovat s thready není tak jednoduché jako bez nich a navíc narůstá možnost chyb.

14.5 Multitasking

Dnešní OS jsou multitaskingové. To znamená, že v jednom okamžiku běží několik programů (obrázek č.7). Vypadá to, že tyto programy běží současně. Není tomu ale tak.



obrázek č.7

Plánovač přiděluje jednotlivým úlohám výkon procesoru jen na určitý čas. Plánovač pak tyto úlohy posílá procesoru ke zpracování na základě priorit (ty lze zjistit např. ve správci úloh). Výpočty procesorů jsou tak rychlé, že to působí, jakoby úlohy běžely najednou.

5 CELL (konkurence pro DualCore)

Vývojář procesorů má jednoduchý úkol za pomoci jakéhokoli způsobu dostat z procesoru naprosté výpočetní maximum. Na začátku byl samotný procesor. Postupem času se přidal matematický komprocesor a nakonec SIMD pro celá čísla a plovoucí desetinnou čárku. Toto obsahuje v podstatě každý procesor. Jak tedy zvýšit výkon? Přece použít procesory dva. Tak vznikly víceprocesorové počítače. A protože jde v podstatě o dva identické procesory, vzniklo pro ně označení symetrický multiprocessing. Mezi nejznámější patří např. procesor Opteron od firmy AMD či Intel Xeon. Mít ale pro třeba čtyři procesory také čtyři patice ? To není zrovna nejlepší myšlenka. Lepší je, snažit se umístit více procesorů na jeden kousek křemíku. Právě tuto myšlenku využívají např. procesory IBM Power4, IBM PowerPC 970MP, Intel Pentium D nebo AMD Opteron DualCore, protože ani DualCore procesory nestačí, naskytuje se otázka: Je nutné, aby každé jádro mělo všechny součásti? Třeba aby jeden kompletní procesor řídil několik „ořezaných“ jednotek a právě toto je typický znak procesoru CELL.

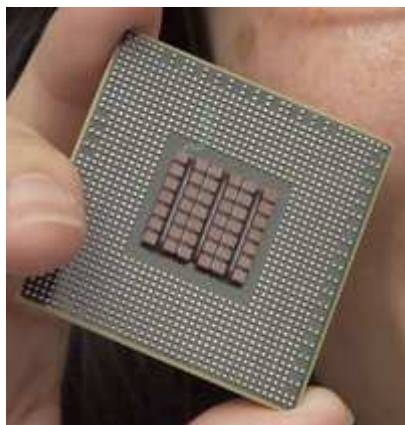
Nejtěžší na používání procesorů s více jádry je to, zda jsme schopni „zaměstnat“ tolik procesorů najednou. Vytvořit procesor s řádově desítky jader není zas až takový problém, horší je ale využití tolika jader najednou. Takovýto obrovský výpočetní výkon, který je vtěsnán do několika čtverečních centimetrů lze využít po málu. Kdo ale přijde na to, jak tento mega výkon ovládnout, může ho využít například v aplikacích, kde je potřeba zpracovávat 3D aplikace v reálném čase (toto bylo demonstrováno na veletrhu CEBIT, kde bylo možné pomocí speciálních brýlí sledovat 3D obraz lidského srdce a to z jakéhokoli úhlu při různé míře transparency) či v leteckých simulátorech.

Projekt firem Sony, Toshiba a IBM dostal název Cell (buňka) podle vnitřní architektury tohoto procesoru. Skládá se z devíti jader PowerPC (obrázek č.9,obrázek č.10), nejedná se ale o symetrický multiprocessing. Jedno jádro je privilegované a ostatním nadřazené s názvem PPE (**P**ower **P**rocessor **E**lement). Ostatní jádra jsou ochuzena o nepotřebné části, označují se jako SPE (**S**ynergistic **P**rocessing **E**lement). PPE má parametry průměrného 64bitového procesoru, velikost cache L1 je 32 + 32 kB, L2 má kapacitu 512 kB a podporuje cosi jako HyperThreading.

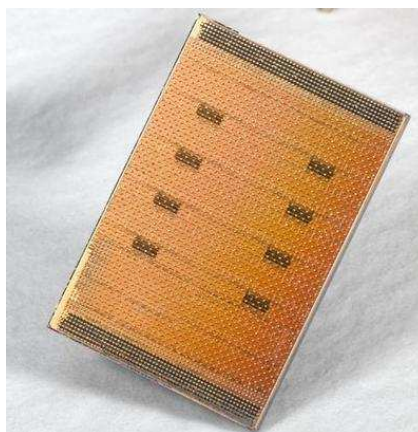
SPE je vlastně vyfiltrovaná součást SIMD, pro plovoucí desetinnou čárku s obvyklou šířkou 128 bitů. Uvádí se, že při frekvenci 3,2 GHz vydá teoreticky výkon

až 26,5 GFLOPS! Vynásobíme-li to sedmi, jelikož jedno SPE je vypnuté, dostaneme obrovský výpočetní výkon. Každé SPE obsahuje 256kB cache L2. Čím více cache paměti má SPE k dispozici, tím jednodušeji ho lze programovat. Cache v Cellu navíc nepracuje jako běžná vyrovnávací paměť procesoru x86. Nejsou v ní tedy uloženy poslední položky čtené z paměti, ale je to spíše rychlá lokální paměť každé vektorové jednotky. Označení cache tedy pro ni ani není přesné.

Klíč k rychlosti je právě ve využití cache pamětí. Toho si bylo IBM vědomo, a proto použilo jednu z nejrychlejších pamětí a to konkrétně XDR DRAM. Jelikož má každé jádro svou vlastní cache nemusí se o ní „hádat“ s ostatními SPE. CELL má na první pohled mnoho jednotek (1+8), ale existují i mnohem složitější. CELL též obsahuje velké množství registrů a programátor tak nemusí pořád něco ukládat do RAM paměti. Po přetaktování 2,4@4GHz lze dosáhnout výkonu až 256 GFLOPS. Pro představu Clearspeed CSX600 má 25 GFLOPS, nVidia GeForce 6800 na 550 MHz má 106 GFLOPS, GeForce FX 5900 má 40 GFLOPS, Riva 128 má 5 GFLOPS, Intel Itanium 2 na 1,6 GHz s 9 MB cache L3 má špičkový výkon 24,2 GFLOPS, Itanium 300 MHz má 13 GFLOPS, Pentium 4 na 3 GHz až 12 GFLOPS, první modely AMD Opteron měly 11,264 GFLOPS, Athlon XP 2200+ má 7,2 GFLOPS, Athlon 600 MHz má 2,4 GFLOPS.



obrázek č.8



obrázek č.9

16 Mobilní a serverové procesory

Mobilní procesory jsou určeny do notebooků (dále již jen jako NB). Nejprve byly do NB dávány klasické desktopové procesory. Později jako první uvedl Intel procesor Pentium M, který byl výhradně určen pro využití v NB. Sice se později objevily možnosti, jak dostat procesor Pentium M do klasických počítačů. Bylo toho dosaženo za pomoci redukce do s478. Posléze bylo možné do redukce umístit mobilní procesor.

16.1 Pentium M

Pentium M bylo uvedeno roku 2003. Šlo o x86 architekturu a vycházelo z platformy Centrino. Písmeno M v názvu je z anglického slova mobile což znamená mobilní (přenosný). Pentium M reprezentovalo nový směr Intelu a rozhodně se nejednalo o slabší verzi desktopového procesoru (Pentium 4). Pentium M nevychází z Pentia 4, ale ze staršího Pentia III Tualatin (ten vychází z Pentia Pro). To je optimalizované pro výkonovou efektivitu a pro prodloužení životnosti baterie. Má mnohem nižší příkon a tepelný výkon (TDP). Pentium M pracuje na nižších taktech než Pentium 4, ale naproti tomu podává obdobné výkony.

Pentium M má s Pentiem 4 a Pentiem III kompatibilní sběrnice rozhraní. Má zlepšené dekódování instrukcí, předpověď větvení, zkrácenou pipeline na 12 kroků (díky nižší frekvenci se zkrátila pipeline a tím pádem vzrostl výkon), podporuje SSE2 a má velkou cache paměť. Díky snížení napětí procesoru v době, kdy není zatíženo razantně snižuje spotřebu. Je zde i vylepšená služba SpeedStep. Díky tomu je dosaženo extrémně nízké spotřeby. V klidu procesor potřebuje pouhých 5 W a v zátěži pak 27 W.

16.2 Jádro Banias

První jádro pro tuto mobilní platformu byl Banias. Vyráběl se 130nm výrobním procesem a pracoval na frekvencích 1,3 až 1,7 GHz. Měl velkou L2 cache (1MB). TDP bylo maximálně 24W.

16.3 Jádro Dothan

Druhé jádro bylo pojmenováno Baniyas. Bylo vyráběno 90nm technologií. Obsahovalo 2MB L2 cache. TDP se snížilo v zátěži na 21W a byla i zvýšena frekvence na maximálních 2,26 GHz. Procesory s frekvencemi kolem 1GHz byly ultra nízko napěťové (0,94V), což maximálně prodlužovalo výdrž baterie.

16.4 Současná nabídka

Firma AMD zpočátku dávala do NB pouze vybrané kusy svých procesorů. Teprve později přišla s platformou Turion. To je ovšem pouze upravená verze Pentia M. V současné době přichází Intel s úplně novou platformou Core (Core Duo). Pro mobilní oblast Intel uvede (třetí kvartál 2006) jádro Merom. AMD odpovídá vydáním DualCore procesoru Turion 64 X2, který ovšem již přichází na trh. Firma AMD tak získává náskok před firmou Intel.

16.5 Serverové procesory

Vždy se lišily od těch desktopových. Tyto procesory se vyznačovaly vylepšenými vlastnostmi jejich desktopových „příbuzných“. Šlo o velikosti cache pamětí, vylepšené zpracování paralelizmu, možnost osadit desku větším množstvím RAM pamětí a hlavně víceprocesorové zapojení (2,4,8 procesorů na jedné základní desce). V současné době jsou serverové procesory zastoupené především výrobky firem Intel a AMD. Firma AMD nabízí serverové procesory s označením Opteron a firma Intel procesory Xeon (někdy i Xeon MP).

17 Závěr

Cílem této práce bylo popsat vývoj Central Processor Unit (CPU). Doufám, že informace obsažené v této práci budou pro spoustu lidí návodem k pochopení, co to je CPU. Toto téma je velice obsáhlé a není možné napsat vše. Snažil jsem se vybrat to nejpodstatnější a také nejzajímavější, co by mohlo případné zájemce zajímat. I když to nebyl zrovna lehký úkol, věřím, že se mi to podařilo. Při získávání informací potřebných pro vypracování jsem se dozvěděl i spoustu věcí kolem . Ty mi pomohly si objasnit dosud nejasné partie tohoto oboru.

Použitá literatura

www.amd.com

www.intel.com

www.theinquirer.net

www.xbitlabs.com

www.hardwaresecrets.com

www.anandtech.com

www.techgage.com

www.wikipedia.org

www.svethardware.cz

www.pentium4.cz

www.pctuning.cz

Lačezar Ličev, David Morkes: Procesory, Computer Press, Brno, 1999

Vondráček Martin: *Procesory*,

<http://fallvonder.wz.cz/work/design/procesory/uvod.html>

Koc Petr: Obecně o procesorech,

http://www.svethardware.cz/branch_sch.jsp?name=Procesory&ga=Procesory%5CObecn%C4%9B