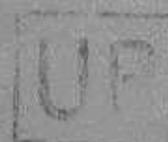


VYSOKÁ ŠKOLA BĀŇSKĀ - TECHNICKĀ UNIVERZITA OSTRAVA



Fakulta elektrotechniky a informatiky

ARCHITEKTURA POČÍTAČŮ I.

Lačezar Ličev



Ostrava

VYSOKÁ ŠKOLA BÁŇSKÁ - TECHNICKÁ UNIVERZITA OSTRAVA

Fakulta elektrotechniky a informatiky

ARCHITEKTURA POČÍTAČŮ I.

Lačezar Ličev

Ostrava

TINLIB

OBSAH

Předmluva.....	2
1. Přehled jednotlivých technologií číslicových obvodů.....	3
1.1 Bipolární technologie.....	3
1.1.1 Technologie TTL, S-TTL a LS-TTL.....	3
1.1.2 Technologie ECL.....	3
1.1.3 Technologie I ² L a I ³ L.....	4
1.2 Unipolární technologie.....	4
1.2.1 Technologie MOS-P a MOS-N.....	4
1.2.2 Technologie HMOS, HMOSII a HMOSIII.....	4
1.2.3 Technologie CMOS CHMOS a SOS.....	5
1.2.4 Technologie FAMOS, FLOTOX a HMOS-E.....	5
1.2.5 Technologie CCD.....	6
1.2.6 Tranzistory typu MOSFET.....	6
1.3 Procesor, mikroprocesor a mikropočítačová sada.....	8
1.3.1 Počítač, minipočítač, mikropočítač a monolitický mikropočítač.....	9
1.3.2 Pseudomikroprocesory a řezové sady.....	10
1.3.3 Přehled vývoje mikropočítačové techniky.....	10
2. Strojová instrukce, pseudoinstrukce, makroinstrukce, porovnání a příklady.....	11
2.1 Základní pojmy.....	11
2.1.1 Bit, šifra toku dat, slabika a slovo.....	11
2.1.2 Řadič.....	11
2.1.3 Doba, fáze, strojový a instrukční cyklus, stavy WAIT a HOLD.....	11
2.1.4 Mikrooperace, mikroinstrukce, mikroprogram a mikroprogramování.....	11
2.1.5 Strojová instrukce, pseudoinstrukce a makroinstrukce.....	12
2.1.6 Podprogramy otevřené a uzavřené.....	12
2.2 Adresy a adresování.....	12
2.2.1 Implicitní adresování a adresy v operačním znaku.....	12
2.2.2 Přímý operand a přímá adresa.....	13
2.2.3 Nepřímé adresy a nepřímé registrové adresování.....	13
2.2.4 Adresování ukazateli.....	13
2.2.5 Adresy relativní, autorelativní, segmentové a indexové.....	13
2.3 Architektura mikroprocesorů a mikropočítačů.....	13
2.3.1 Zápisníková paměť.....	14
2.3.2 Zásobníková paměť LIFO a volání podprogramu.....	14
2.3.3 Fronta (FIFO).....	15
2.3.4 Hierarchie sběrnic.....	15
2.3.5 Von Neumanova a harwardská koncepce procesorů, zásobníkové orientované procesory.....	15
2.4 Pokročilejší architektury.....	16
2.4.1 Přesahování a řetězení.....	16
2.4.2 Multiprocesory a procesorová pole.....	16

3.	Styk s periferním zařízením	17
3.1	Technika V/V bran	17
3.2	Programové řízení styku	18
3.3	Přerušení programu	18
3.4	Přímý přístup do paměti - DMA	21
4.	Mikroprocesory architektury CISC a RISC	23
4.1	Historie vývoje procesorů RISC	24
4.2	Definice architektury RISC	25
4.3	Některé základní pojmy	26
4.4	Zřetěžené zpracování informace	27
4.4.1	Typy zřetěžení	29
4.4.2	Klasifikace zřetěžených systémů	31
4.5	Zřetěžené instrukce procesoru	32
4.5.1	Realizace zřetěžení instrukcí procesoru	32
4.4.2	Problémy zřetěžení instrukcí	33
4.5	Vektorové výpočty	35
4.5.1	Charakteristika vektorového zpracování	35
4.5.2	Architektura vektorového procesoru	36
4.6	Vyrovňovací paměti	37
4.6.1	Charakteristika vyrovnávacích pamětí	38
4.7.2	Algoritmy správy dat	39
4.7.3	Příklad implementace paměti cache	40
5.	Mikroprocesory firmy INTEL	42
5.1	Mikroprocesor 8080A	42
5.1.1	Popis mikroprocesoru 8080A	42
5.2	Mikroprocesory I8086 a I8088	44
5.2.1	Základní vlastnosti mikroprocesorů 8086 a 8088	45
5.2.2	Organizace paměti, segmentace a start mikroprocesoru	47
5.2.2.1	Organizace paměti	47
5.2.2.2	Segmentace paměti	48
5.2.2.3	Start mikroprocesoru a inicializace registrů	49
5.2.3	Vstupy, výstupy a DMA	50
5.2.4	Prizpůsobení pro multiprocesorové systémy a stavové signály	50
5.2.5	Přerušování programu	50
5.2.6	Tabulky elektrických a časových údajů	52
5.2.7	Instrukční soubor mikroprocesorů 8086 a 8088	52
5.2.8	Formáty dat a strojových instrukcí	52
5.2.9	Způsoby adresování a trvání instrukčního cyklu	53
5.2.10	Programová slučitelnost s 8080A a konverzní program CONV86	53
5.3	Mikroprocesory 80186 a 80188	54
5.4	Mikroprocesor 80286	55
5.4.1	Ochrana paměti	55
5.4.2	Popis 80286	57

5.5	Mikroprocesor 80386	59
5.6	Mikroprocesor 80486	61
5.7	Pentium	63
5.8	Procesor PentiumPro	70
5.9	Procesor PENTIUM II	81
6.	Procesory jiných firem	90
6.1	Acorn RISC Machine	90
6.2	RISC procesor Motorola 88000	94
6.3	RISC procesory Intel i860 a i960	96
6.4	Procesory firmy DEC	99
6.4.1	Procesor ALPHA 21064A	99
6.4.2	Procesor ALPHA 21164PC	101
6.4.3	Procesor ALPHA 21164 (266-333 MHz)	104
6.4.4	Procesor ALPHA 21164 (366-600 MHz)	105
6.4.5	Procesor ALPHA 21264	108
6.5	Procesory PowerPC	111
6.5.1	Procesor PowerPC 601	111
6.5.2	Procesor PowerPC 620	114
6.5.3	Procesor PowerPC 603	116
6.5.4	Procesor PowerPC 604	120
6.5.5	Procesor PowerPC 750	123
6.6	Procesory MIPS	126
6.6.1	Procesor MIPS R3000	126
6.6.2	Procesor MIPS R 4000	127
6.6.3	Procesory MIPS R4400, R4600 a R8000	128
6.6.4	Procesor MIPS R10000	148
6.7	Procesory SPARC	152
6.7.1	Procesor UltraSPARC	152
6.7.2	UltraSPARC™ - II	157
6.7.3	Procesor UltraSPARC™ - III	162
6.7.4	Procesor SuperSPARC™ II	167
6.7.5	Procesor UltraSPARC-III	170
6.7.6	Procesor MicroSPARC™ -IIep	170
6.8	Procesory firmy AMD	173
6.8.1	Procesor Am486	174
6.8.2	Procesor Am5x86	175
6.8.3	Procesor AMD-K5	176
6.8.4	Procesor AMD-K6	182
6.9	Procesory firmem Cyrix a IBM	191
6.9.1	Procesor 6x86	191
6.9.2	Procesor Cyrix/IBM 6x86MX (M2)	199

7. Matematické koprocesory	205
7.1 Numerický koprocesor 8087	205
7.1.1 Základní vlastnosti 8087	206
7.1.2 Instrukční soubor a součinnost 8087 a CPU.....	209
7.1.3 Programový emulátor E8087.	210
7.2 Stykový koprocesor 8089.	211
7.3 Numerický koprocesor 80287 a 80387	211
Literatura	212

Skripta "Architektura počítačů I" jsou určena studentům studijního oboru Inženýrská informatika VŠB- Technické univerzity Ostrava, případně dalším zájemcům vysokých škol.

Poskytují čtenáři základní informace o technologiích vyrábění číslicových počítačů, definují základní pojmy jako jsou strojové instrukce a makroinstrukce. Dále poskytují základní informace o způsobu provádění komunikace s periferním zařízením, atd.. Velká pozornost je věnována v těchto skriptech kapitole zabývající se popisem mikroprocesorů. Jsou popsány všechny procesory od Intelu 8080 až po PentiumPro a procesory od MIPS 3000 až po 10000. Dále je věnována pozornost procesorům ze skupiny RISC, kde je popsána jejich definice, vlastnosti a vše je demonstrováno na konkrétní typy (i860-960, Motorola 88000, Power PC 601, 620, atd.). Také je věnována pozornost monolytickým počítačům I8048, TPD 78C06, I8051/52, I8096, atd.

Všechny partie obsažené v těchto skriptech jsou uspokojivým vysvětlením a dávají čtenáři komplexní informaci o architektuře počítačů studované v předmětu Číslicové počítače 2 oboru Inženýrská informatika.

Na tomto místě bych chtěl poděkovat všem, kteří mi byli nápomocní při sběru, utřídění a zpracování textu. Zejména studentům 3. ročníku (Šk. rok 1997-1998) oboru Inženýrská informatika, kolegům Ing. P. Olivkovi a Ing. P. Grygarkovi za věcné poznámky k textu některých kapitol a nakonec kolegům z firem NOVÁ HUTĚ, a.s., a AŘ a.s. OKD, kteří skriptum přečetli a vyjádřili svůj názor a náměty, které jsem pak do konečného textu zahrnul.

Ostrava, červenec 1998

autor

PŘEDMLUVA.

Účel

Výpočetní technika za jeden rok zaznamenává takový rozvoj, který je srovnatelný s rozvojem jiných vědních disciplínách za deset a více let. Z tohoto důvodu jsem se rozhodl vydávat skripta v menším nákladu a častěji. Skripta budou aktuální v době vydání.

Koncepce

Skripta "Architektura počítačů Ia" obsahuje 1. část z přednášek předmětu Číslicové počítače 2.

Skriptum obsahuje následující kapitoly:

1. Přehled jednotlivých technologií číslicových obvodů, základní obvody TTL, LS TTL, ... a jejich vlastnosti.
Základní schéma mikroprogramového řadiče, mikroprogramování, ukázky mikroprogramů.
2. Strojová instrukce, pseudoinstrukce, makroinstrukce: porovnání, příklady použití. Adresace, adresování, typy adres a kódování ve strojovém kódu.
3. Styk s periferními zařízeními: standardní kanál IBM, V/V brány, přímý přístup do paměti - DMA. Řízení styku: programové, pomocí přerušení.
4. Mikroprocesory architektury CISC a RISC. Charakteristika architektury, zřetěžené zpracování informace, vektorové výpočty a vyrovnávací paměti.
5. Univerzální mikroprocesory firmy Intel: I8080, I8086, I8088, I80186, I80188, I80286, I80386, I80486, Pentium, PentiumPro a Pentium II.
6. Procesory jiných firem: Procesor ARM, Motorola 88000, i860 a i960, firmy DEC, PowerPC 601, 620, 603, 604 a MPC 750, MIPS 3000, 4000, 4400, 4600, 8000 10000, Procesory SPARC UltraSPARC, UltraSPARC II, UltraSPARC III, SuperSPARC II, UltraSPARC III a MicroSparc IIep, firmy AMD K4, K5 a K6 a firmy Cyrex a IBMx86 kompatibilní IBM 486 DX4, Cyrex 6x86 a Cyrex/IBM 6x86 MX.
7. Matematické koprocessory: I8087, I80287, I80387. Princip činnosti koprocessoru, styk s procesorem, formáty čísel.

Při studování jednotlivých kapitol těchto skript je nutné si uvědomit, že nejsou nezávislé, ale na sebe navazující.

Tato skripta jsou určena především studentům VŠB-TU FEI, projektantům řídicích a informačních systémů, programátorům a technikům a v neposlední řadě i uživatelům výpočetní techniky.

1. PŘEHLED JEDNOTLIVÝCH TECHNOLOGIÍ ČÍSLICOVÝCH OBVODŮ.

Polovodičové integrované obvody se dělí podle způsobu přenosu elektrického náboje přechodem tranzistorů na dvě základní skupiny - "bipolární a unipolární" [1,2]. V prvním případě se přenosu náboje současně účastní elektrony a díry, v druhém pouze jeden druh z obou typů nosiče náboje (elektrony, díry).

Bipolární obvody jsou ve srovnání s unipolárními obvykle rychlejší, na druhé straně mají větší příkon a nedovolují dosáhnout tak vysokého stupně integrace.

1.1 Bipolární technologie.

1.1.1 Technologie TTL, S-TTL a LS-TTL.

TTL je nejstarší a dosud nejrozšířenější technologie při výrobě obvodů SSI a MSI [1,2] hlavně díky jedinému napájecímu napětí (+5 V), poměrně vysokému log. zisku (větvení) i rychlosti, a také ceně.

Klasická technologie TTL a její varianty L-TTL (pomalejší se sníženým příkonem) a H-TTL (rychlá s větším příkonem) jsou vytlačovány technologiemi S-TTL a LS-TTL.

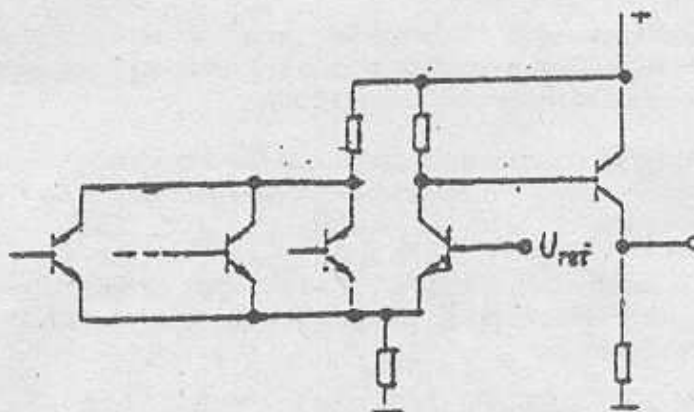
S-TTL je v podstatě varianta technologie TTL, u které se podařilo příznivějšího poměru mezi rychlostí i příkonem. LS-TTL je ve skutečnosti DTL, tj. diodová logika kombinovaná s tranzistorovou výstupní kaskádou z obvodů TTL.

Technologie S-TTL a energeticky úspornější LS-TTL a ALS-TTL jsou využívány při výrobě velmi rychlých malých pamětí (např. MH74S287) a hlavně rychlých a relativně výkonných pomocných obvodů (např. MH3212, 3216, 3226, 8282/83, 8286/87).

Nicméně tyto technologie TTL neumožňují vysoký stupeň integrace.

1.1.2 Technologie ECL.

Paměťové a jiné součástky vyráběné technologií ECL (emitor coupled logic) využívají tzv. proudové spínače, tj. zapojení tranzistorů s emitorovou vazbou dle obr. č. 1.1 [2].



Obr. č. 1.1

Obvody ECL mají malý rozkmit signálu a tím i malou šumovou imunitu a také nejsou přímo slučitelné s obvody TTL.

Technologie ECL se používá pro výrobu mimořádně rychlých pamětí RWM a pro řezové stavebnice (řada M10800-Motorola). Největší uplatnění mají obvody ECL při stavbě velmi rychlých počítačů avšak v mikropočítačové technice se téměř neuplatňují.

1.1.3 Technologie I^2L a I^3L .

Technologie I^2L a I^3L jsou bipolární technologie, které dovolují dosahovat velmi značné hustoty integrace, nízkého příkonu, necitlivosti ke změnám napájecího napětí a současně i rychlostí větších než u běžných unipolárních obvodů [1,2].

Techologie I^2L nepoužívá téměř vůbec rezistory.

Uvedené vlastnosti spolu s dobrou slučitelností s obvody TTL předurčují tyto technologie pro výrobu velkokapacitních RWM i výkonných mikroprocesorů.

Přesto obvody I^2L a I^3L nedosahují předkládaného rozšíření z důvodu poměrně vysokých výrobních nákladů a současná konkurence nových unipolárních technologií.

1.2 Unipolární technologie.

S unipolární technologií je spojen celý dosavadní bouřlivý rozvoj oborů mikropočítačů.

1.2.1 Technologie MOS-P a MOS-N.

U obvodů MOS-P je základním prvkem unipolární tranzistor MOS s kanálem typu P, což např. u paměti MOS-P poskytuje dobu přístupu 1-5 Ts. Vzhledem k nízkým rychlostem spínání a špatné slučitelnosti s log. obvody TTL se technologie MOS-P již nepovažuje za perspektivní [2].

Základním prvkem obvodů MOS-N je unipolární tranzistor s kanálem typu N [2], který je díky pohyblivějším nosičům náboje asi 3x rychlejší než tranzistor MOS-P.

Zvětšování integrace, a s tím spojené zmenšování rozměrů, parazitních kapacit a vzdáleností, velmi brzy vedlo ke zvyšování rychlosti obvodů - 10x.

Uvážíme-li navíc snadnou slučitelnost s obvody TTL a jediné napájecí napětí +5 V, pochopíme, proč je technologie MOS-N dnes základní.

1.2.2 Technologie HMOS, HMOSII a HMOSIII.

Firma INTEL vyvinula dokonalejší výrobní postupy označované HMOS (high performance) [2]. Další firmy pak zavedly obdobné technologie a označují je : X-MOS (National Semiconductor), S-MOS (Texas Instruments), ... a jen Motorola zachovává původní název HMOS .

Všechny jsou založeny na skutečnosti, že součin zpoždění a ztrátového výkonu je přibližně úměrný třetí mocnině rozměru základní struktury. To znamená, že při zachování ztrátového výkonu lze zmenšením o 50% dosáhnout 8x rychlejší činnosti.

Technologie HMOS tedy nevychází ze zcela nových principů a zapojení. Upravují se zejména: délka kanálu, tloušťka oxidové vrstvy řídicí elektrody, hloubka difuze, překrývání elektrod, tloušťky a rozteče spojů a také úroveň dotace příměsími.

Nepříznivé vedlejší efekty: zvyšování intenzity elektrického pole při stejném napájecím napětí, pronikání elektronů do řídicí elektrody, povrchové průrazy atd.

Některé z těchto potíží by bylo možno potlačit snížením napájecího napětí z +5 V na +3 V. To však je nežádoucí, a proto se problémy zdokonalené technologie HMOSII řeší tím, že se některé parametry neupravují faktorem "S", ale faktorem "a.S".

Technologií HMOS a jejími verzemi se vyrábějí monolitické mikroprocesory s až půl miliónem tranzistorů na čip a také nejmodernější paměti RWM-RAM s kapacitou 1 Mbitů atd..

1.2.3 Technologie CMOS CHMOS a SOS.

CMOS (Complementary MOS) je technologie [2], která vychází z použití tranzistoru MOS-N jako základního spínacího prvku s aktivní zátěží, již tvoří tranzistor MOS-P.

Taková struktura dovoluje dosáhnout široké rozmezí napájecího napětí (3-18 V), dobré šumové imunity i slučitelnost s logickými obvody TTL.

Nejdůležitější vlastnosti obvodu CMOS je extrémně nízký příkon, umožňující snadné bateriové zálohování, a tím dosažení energetické nezávislosti u paměti RWM-RAM.

Aplikací principů známých z technologie HMOS vzniká vysoce perspektivní technologie CHMOS.

Na bázi CMOS se vyrábějí logické obvody SSI a MSI, řezové procesorové stavebnice, monolitické mikroprocesory i mikropočítače, paměti i ostatní obvody LSI a VLSI.

SOS (silicon on saphire) - základem čipu je destička syntetického safíru. Na safíru, který má stejnou krystalickou strukturu i teplotní roztažnost jako křemík, se pak nechává epitaxně narůstat tenká vrstva křemíku a na ní se realizují spínací prvky. Hustota je 4x větší než u technologie CMOS.

Paměti RWM na bázi technologií SOS mají vynikající vlastnosti, ale taktéž i vyšší výrobní náklady.

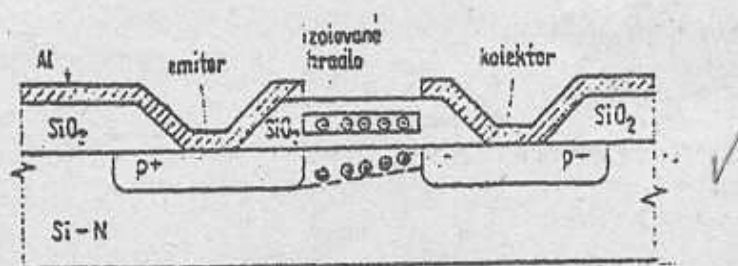
1.2.4 Technologie FAMOS, FLOTOX a HMOS-E.

Technika plavoucího hradla (gate) s lavinovou injekcí nosičů - FAMOS (floating gate avalanche injection) - je nejrozšířenější technologií pro výrobu elektricky programovatelných pevných pamětí EPROM [2].

Při mazání informace z paměti se na čip působí **ionizujícím zářením**. Elektronů v ozářené řídicí elektrodě absorbují fotony záření a získávají dostatečnou energii k překonání bariéry v opačném směru. Potenciály hradla a emitoru se tak vyrovnají a paměť je opět připravena k dalšímu programování.

Při každém mazání informace ozářením dochází k mírné degradaci parametrů paměťové buňky FAMOS. Při šetrném mazání pomocí "studeného ultrafialového záření" parametry paměti EPROM nevybočí z tolerance po desítkách cyklů "mazání-programování".

Na obr. č. 1.2 [2] je znázorněn princip buňky FAMOS paměti EPROM.



Obr. č. 1.2

Pokud hradlo není nabito, kanál neexistuje a dráha "kolektor-emitor" je nevodivá. Je-li izolované hradlo nabito zápornými náboji, vytváří se pod ním vodivý kanál. Modifikací technologie FAMOS je proces FLOTOX (floating-gate tunnel-oxide), který se používá při výrobě elektricky mazatelných pamětí E²ROM.

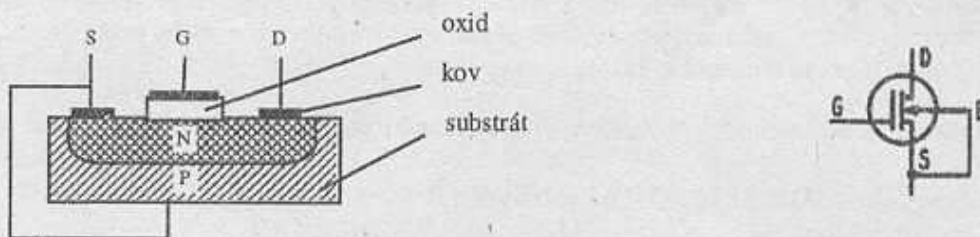
Při dalším rozvoji těchto technologií se uplatňují tytéž trendy zjemňování struktury čipů od MOS-N k HMOS. Výsledkem je technologie HMOS-E, kterou se vyrábějí nejmodernější paměti EPROM a E²ROM firmy INTEL.

1.2.5 Technologie CCD.

Pro součástky CCD (charge coupled devices) je typický přenos náboje na parazitních kapacitách soustavou elektrod vytvořených technologií MOS. To předurčuje technologii CCD k tvorbě velkokapacitních dynamických posuvných registrů, které by mohly konkurovat diskovým pamětím.

1.2.6 Tranzistory typu MOSFET

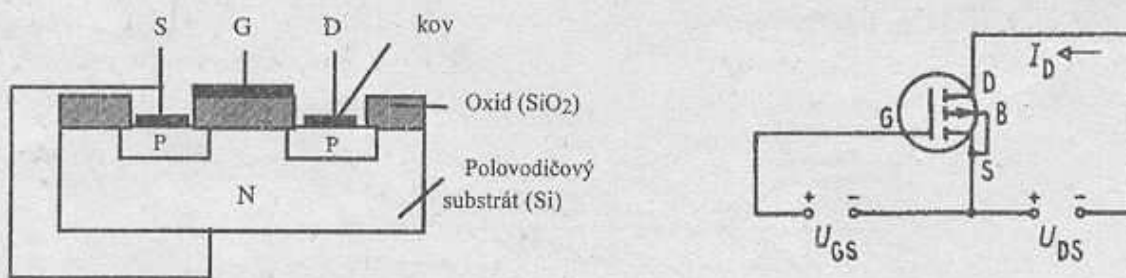
Princip unipolárních tranzistorů [34, 35] spočívá v řízení proudové dráhy pomocí elektrického pole. Podle toho, jak je tato proudová dráha, obecně nazývaná kanál, dotována, hovoříme o tranzistoru s kanálem N nebo P. Technicky jsou dvě možnosti, jak nechat působit elektrické pole na proudový kanál, a to buď přes závěrnou vrstvu přechodu PN, nebo přes zvláštní izolační vrstvu. Proto existují tranzistory typu FET (field effect transistor) s přechodovým hradlem označovaným také JFET (Junction FET) nebo na druhé straně FET s izolovaným hradlem označované IGFET (insulated-gate-FET) nebo MOSFET (metal-oxide-semiconductor FET).



Obr. č. 1.3 Ochuzovací typ IGFET s kanálem N, technologické uspořádání, schématická značka

U tranzistorů s izolovaným hradlem rozlišujeme ochuzovací typ a obohacovací typ. Ochuzovací typ tranzistoru MOSFET se v praxi používá v provedení s kanálem N. Řídicí mechanismus ochuzovacího typu je charakterizován tím, že do kanálu N proniká elektrické pole, které přes velmi tenkou izolační vrstvu a spolu se závěrnou vrstvou protilehlé strany způsobí omezení proudu protékajícího mezi kontakty D a S. Záporné napětí přivedené mezi kontakty GS vypudí další elektrony z kanálu N a tím se zvětší odpor této vrstvy a zmenší se protékající proud.

Pro potřeby číslicové techniky je důležitější obohacovací typ MOSFET s kanálem P nebo N.



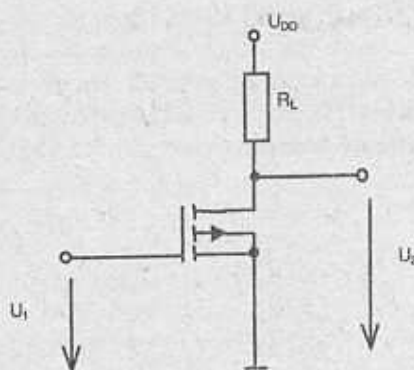
Obr. č. 1.4 Obohacovací typ MOSFET s kanálem P, v zapojení se společným emitorem, technologické uspořádání, schématické znázornění

Z obrázku je patrné, že při napětí $U_{GS} = 0$ nemůže mezi S a D procházet proud. Dosáhne-li U_{GS} záporných hodnot, budou vlivem elektrického pole v tenké oblasti N před hradlem vytlačovány elektrony a na jejich místo přijdou díry. Mezi oběma ostrůvky P vznikne tenký kanál P. Protože tento kanál vznikne obohacením tenké oblasti N pod hradlem děrami, dostal tento MOSFET název obohacovací typ. Jestliže tedy určitým napětím U_{GS} vznikl kanál, může prudce stoupnout s U_{DS} proud I_D . Jak silně stoupne proud I_D , závisí na řídicím napětí U_{GS} , protože jeho velikost určuje hloubku proudového kanálu. Tento MOSFET díky vysokému vstupnímu odporu, který je způsoben izolační vrstvou a není teplotně závislý, a vzhledem k výborným spínacím vlastnostem, hraje významnou úlohu v číslicové technice.

Spínač typu MOSFET

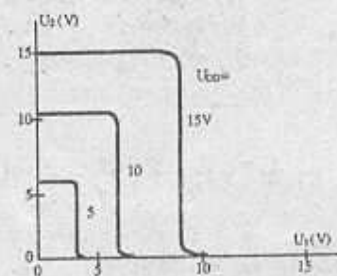
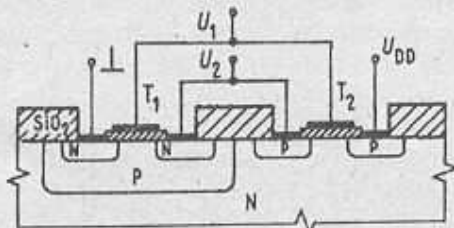
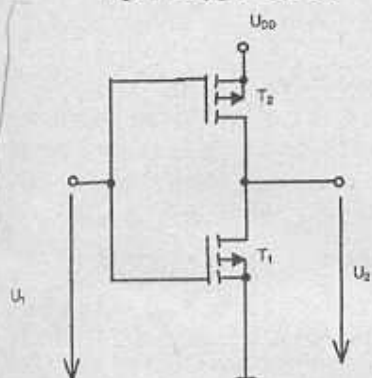
Z tranzistorů řízených polem se jako spínač používá nejčastěji obohacovací typ MOSFET s kanálem P nebo N, který má ze všech tranzistorů řízených polem nejvýhodnější vlastnosti.

Pro $U_1 = 0V$ je FET zavřen: $U_2 = 15V$. Od prahové hodnoty napětí $U_{GS} = -5V$ se začíná otevírat a při $U_1 = -15V$ je plně otevřen: $U_2 = -3$ až $-7V$. Z tohoto poměrně velkého zbytkového napětí vyplývá velký propustný odpor tranzistoru. Proto musí být velký i odpor zatěžovacího rezistoru, aby při otevřeném tranzistoru byl zajištěn příznivý poměr odporu zátěže k odporu tranzistoru, tj. $\min > 100k\Omega$. V integrovaných obvodech se tento rezistor nahrazuje dalším tranzistorem řízeným polem ve funkci rezistoru.



Obr. 1.5 Obecné znázornění IGFET jako spínače

Spínač typu CMOS



Obr. 1.6. Spínací stupeň v technice CMOS, technologické uspořádání a přenosové charakteristiky

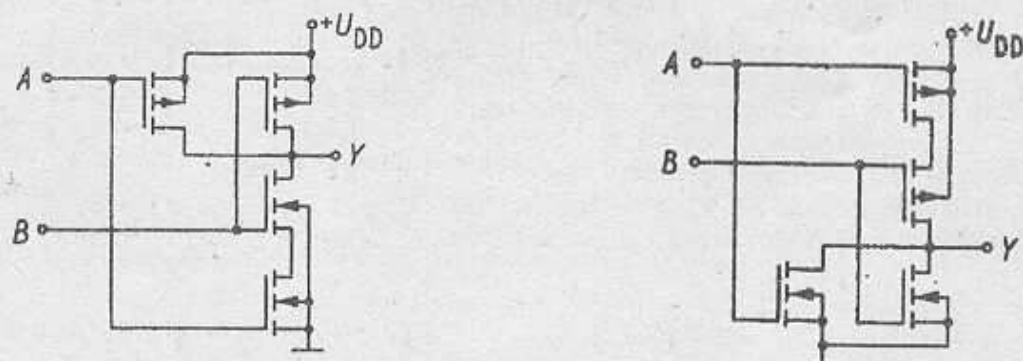
Při hledání spínacích obvodů MOS s vyšší odolností proti rušení, s krátkými spínacími dobami a nízkým příkonem, dojdeme ke komplementární technice CMOS. V ní se kombinuje FET s kanálem P a s kanálem N. Obrázek č. 1.6 přibližuje takto realizovaný spínací stupeň včetně jeho technologického uspořádání a přenosových charakteristik. Jako příklad pro objasnění činnosti stupně vybereme charakteristiku pro $U_{DD} = 10V$. V rozsahu $U_1 = 8-10V$ je T_1 otevřen a T_2 je uzavřen. Ve střední oblasti jsou oba tranzistory otevřeny, tj. jmenovitá hodnota proudu 1 mA prochází jen v okamžiku přepnutí. Další předností této techniky je v tom, že je nezávislá na poměru děliče odporu zátěže k propustnému odporu tranzistoru (není totiž nutný zatěžovací rezistor).

Spínač CMOS má mnoho technologických forem. V tabulce jsou udány nejdůležitější typy, včetně prahových napětí a mezních frekvencí.

Tabulka č. 1.1

Typ	Prahové napětí (V)	Mezní frekvence (Mhz)
Kanál P	2 až 5	2
Kanál N	0,25 až 1,5	5
MNOS (metal nitrid oxide semiconductor)	2	2
Křemíkové hradlo	1,5 až 2	3
Iontová implantační technika	1 až 2	5
CMOS	0,25 až 2	20
SOS (silicon on saphir)	2	30

Spínač CMOS má téměř ideální chování, proto jej lze považovat za ideální negátor. Dále je důležité, že logické obvody založené na členech NAND nebo NOR jsou v technice CMOS rovnocenné, protože vzájemný poměr odporů zde nehraje roli. Na obrázcích je příklad logického členu NAND (sériové zapojení tranzistorů s kanálem N) a členu NOR (paralelní zapojení tranzistorů s kanálem N). Protože napájecí napětí obvodů CMOS má většinou hodnoty $U_{DD} = +5V$ až $+15V$, lze tento druh logických obvodů snadno slučovat s obvody TTL.



Obr. č. 1.7. Logické členy CMOS - NAND a CMOS - NOR

Ve skutečnosti však IO označované jako CMOS nemusí být 100% provedeny v CMOS technologii, tzn. že poměr tranzistorů P:N není 50% : 50%. Od takových obvodů nelze proto očekávat ideální charakteristiky jako od obvodů v klasickém provedení CMOS, výhodou je ale nižší cena a větší rychlost. Početnost obvodů CMOS na čipu závisí na cílech, jakých potřebuje návrhář dosáhnout. V podstatě lze rozlišovat tři skupiny čipů CMOS:

- Klasické čipy CMOS, v nichž návrhář striktně dodržuje počet tranzistorů s kanálem P a s kanálem N v procentním poměru 50 : 50. Takový návrh je nákladnější co do plochy čipu, ale běžně se ho používá ve všech případech, kdy je třeba maximálně využít všech výhod technologie CMOS. Např. v systémech, jež musí pracovat dlouhodobě jen s bateriovým napájením nebo při extrémních teplotách.
- Komerční čipy CMOS pro středně náročné aplikace. Funkčních zlepšení pomocí obvodů CMOS se dosahuje za cenu mírného zvětšení plochy čipu a procentní poměr je 40 : 60 až 20 : 80. Do této skupiny se řadí většina variant 8bitových mikroprocesorů CMOS.
- Speciální čipy CMOS. Jde o čipy, které by se daly normálně realizovat technologií NMOS, vyskytují se v nich ale tzv. "horká" místa. V takových případech lze přidat do určitých míst poměrně malý počet tranzistorů s kanálem P ke splnění určitých cílů návrhu, např. aby se dalo použít levné plastické pouzdro, nebo pro zvětšení teplotního pracovního rozsahu. Na tomto principu lze aplikovat CMOS i v čípech VLSI, např. pro 16/32bitové mikroprocesory a jejich podpůrné obvody.

1.3 Procesor, mikroprocesor a mikropočítačová sada.

V následujících částech si osvětlíme hlavní pojmy jako jsou procesor, mikroprocesor atd [2].

Procesorem se rozumí základní jednotka počítače, tj. logický automat pro zpracování informací, obsahující hlavně arit. jednotku a řadič. "Počítač bez periferních zařízení a bez hlavní paměti".

Mikroprocesor, ozn. CPU (central processor unit), je malý procesor vyráběný technologií velké integrace.

Mikroprocesor má poměrně univerzální strukturu seskupenou kolem jediného výkonného členu. Jeho instrukční soubor je koncipován především na výpočty a logické funkce. Je přednostně orientován na operace nad slovy. Je možné integrovaný programovatelný automat orientovat na ryze logické bitové operace. Pak jde o logický nebo též **booleovský mikroprocesor**.

Každý výrobce mikroprocesorů nabízí i součástky doporučené pro výstavbu zařízení s určitým mikroprocesorem. Jsou to tzv. **mikropočítačové sady** označene MCS (microcomputer set). Každá sada obsahuje další součástky, které mohou být i samostatně použity. Jsou to především:

- Paměťové součástky RWM-RAM, lišící se kapacitou, organizací dat, rychlostí, principem činnosti (statické, dynamické) atd.
- Pevné (permanentní) paměti ozn. ROM (read only memory), programovatelné pevné paměti PROM a paměti, jejichž obsah může uživatel měnit EPROM.
- Pomocné členy. Patří zde generátory hod. signálů, budiče sběrnic, registry, řídicí obvody různé úrovně atd.

MCS neznamena mikropočítač, ale pouze seznam typů součástek doporučených k použití s určitými typy mikroprocesorů.

1.3.1 Počítač, minipočítač, mikropočítač a monolitický mikropočítač.

Definice: Počítač (computer) je stroj na číslicové a log. zpracování dat, grafických údajů nebo údajů z výrobního procesu či experimentu. Pracuje samostatně podle programu předem vloženého do paměti.

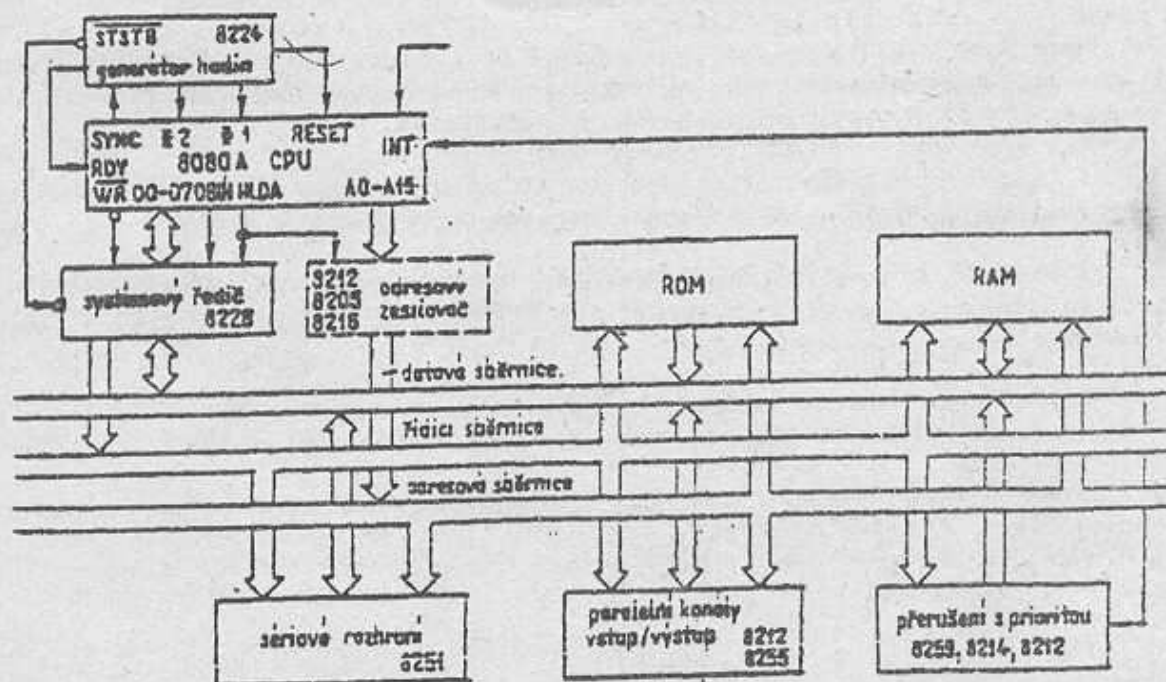
Počítač se skládá z procesoru, hlavní paměti a periferního systému, zahrnujícího: vnější paměť (diskové, páskové ...), terminály, tiskárny, modemy pro styk po telekomunikačních vedeních atd. [2].

Počítače mohou být:

- Stacionární (střediskové) počítače (mainframe). Jsou běžně přístupné uživatelům a jsou obsluhováni specializovaným personálem. Představiteli JSEP, IBM 370... AŘ OKD, NH atd.
- Minipočítače (minicomputer). Jsou proti střediskovým počítačům menší a jednodušší. S tím spojené snížení parametrů je bohatě nahrazeno menší náročností obsluhy, nižší cenou a větší pružností využívání. Jsou proto používány i mimo výp. střediska v laboratořích, jako měřicí a řídicí systémy atd. Přestaviteli SMEP, PDP 11/30, VAX a ...

Z mnoha aplikací jsou již minipočítače vytlačovány vyspělejšími mikropočítači.

- Mikropočítač (personal computer).
Je to sekvenční automat s jedním nebo několika monolitickými mikroprocesory, s paměť a obvody pro vstup a výstup.
Jeho elementární skladba je naznačena na obr. č. 1.8. Zde máme blokové schéma mikropočítače s mikroprocesorem 8080A.



Obr. č. 1.8 Pevná paměť ROM - program, RWM (RAM) - data a zápisník

1.3.2 Pseudomikroprocesory a řezové sady.

Stává se, že určitý typ unipolárního mikroprocesoru plně uspokojuje svými vlastnostmi s výjimkou rychlosti. Pak uživatel může vytvořit s použitím bipolárních obvodů menší integrace procesor, který funkčně kopíruje původní mikroprocesor a je plně slučitelný s jeho programovým vybavením, ale je rychlejší. Ten označujeme pod názvem **pseudomikroprocesor** nebo též **obvodový emulátor** [2].

Řezová sada je název pro soubor obvodů, které vznikají následujícím způsobem: Máme schéma žádoucího procesu. Schéma, hlavně pak arit.log. jednotku myšleně rozdělím na tzv. řezy (slide), tj. opakující se části. Ty se pak vyrobí jako integrované obvody a jejich spojením vzniká úplný procesor. Tak můžeme vytvořit procesory s téměř libovolnou délkou slova (2, 4, 8 bitech).

V ČR se řezů sady MH3000 stavějí především:

- jednodeskový matematický koprocessor do modulární stavebnice MIRIS systému MIKROSAT,
- dvoudeskový řadič pružných disků,
- atd.

1.3.3 Přehled vývoje mikropočítačové techniky.

První integrovaný obvod LSI, který můžeme nazvat mikroprocesor, navrhl pracovník firmy INTEL E. Hoff při práci na vývoji obvodů pro stolní kalkulačtor japonské fy. BUSICOM. Napadlo jej přenést na programovatelnou strukturu univerzálního procesoru řešení úloh. Výsledný obvod byl označen 4004 a byl nazván mikroprocesor, měl šířku dat 4 bity, instr. soubor orientovaný hlavně na práci s čísly v BCD a na log. funkce.

Vývoj však pokračoval dále, firmou ZILOG byla vytvořena mikropočítačová sada MCS o čtyřech členech v čele s mikroprocesorem 4004.

Souběžně s MCS4 byl vyvíjen mikroprocesor 8008 pro inteligentní terminál firmy Computer Terminals Co.

V roce 1972 byl v čele sady MCS8 představen první osmibitový mikroprocesor.

V roce 1974 firma INTEL přišla s typem 8080, který se stal na delší dobu světovým standardem. Asi rok po zavedení typu 8080 uvedla firma MOTOROLA mikroprocesor MC6800 a roku 1976 firma ZILOG mikroprocesor Z80-CPU, který rozvíjí dále architekturu typu 8080A.

Od roku 1978 jsou středem pozornosti 16bitové mikroprocesory. Firmy INTEL, MOTOROLA, ZILOG atd. Architektura zmíněných 16bitových mikroprocesorů dominuje ve světě mikropočítačů.

Koncem 70. let nevznikaly jen 16bitové mikroprocesory, ale i monolitické mikropočítače, které představují samostatnou větev vývoje od poměrně primitivního typu 8048 až k pokročilým typům 8051, 8052 nebo Z8 atd.

2. STROJOVÁ INSTRUKCE, PSEUDOINSTRUKCE, MAKROINSTRUKCE, POROVNÁNÍ A PŘÍKLADY.

V této kapitole se budeme zabývat obecnými vlastnostmi mikroprocesorů i mikropočítačů a vztahy mezi jejich strukturou a vlastnostmi [2].

2.1 Základní pojmy.

2.1.1 Bit, šířka toku dat, slabika a slovo.

- **Bit (binary digit)** je označení pro dvojkovou číslici, která nabývá hodnot 0 nebo 1.
- **Šířka toku dat (šířka sběrnice)** je počet bitů, které se po datové sběrnici přenášejí současně. Některé mikroprocesory pracují s různými šířkami toku dat uvnitř a navenek.
Příklad. Mikroprocesor 8088 má vnitřní datovou sběrnici 16bitovou, ale s vnějším prostředím komunikuje sériově paralelně po 8 bitech.
- **Slabika (byte)** je skupina obvykle 8bitů.
- **Slovo (word)** je skupina slabik, která se v počítači zpracovává jako celek.

2.1.2 Řadič.

Řadič (contoller, contol unit), část procesoru, která řídí vykonávání operace a chod celého procesoru podle instrukce programu. Řadič procesoru obsahuje několik částí. **Registr instrukcí** - uchovává operační znak instrukce po dobu jejího vykonávání, dále **dekodér instrukcí**, který obsah dekóduje a generuje řídicí signály pro procesor.

Řadiče procesorů se budují podle dvou koncepcí.

1. Koncepce - řadič je stavěn jako speciální sekvenční automat, který má obvykle pro každou skupinu podobných operací zvláštní čítač a dekodér. Je drahý a hodí se pro velmi rychlé procesory.
2. Koncepce - pro dekódování operačního znaku se používá tzv. **řídicí paměť**, ve které jsou uloženy mikroprogramy pro řízení operací.

2.1.3 Doba, fáze, strojový a instrukční cyklus, stavy WAIT a HOLD.

Mikroprocesory pracují synchronně, jejich činnost je řízena tzv. **hodinovým signálem**.

- **Doba (time period)** nazýváme časový úsek mezi stejnolehlými body dvou po sobě následujících hodinových impulsů.
- **Fáze (beat)** je označení pro jeden ze stavů řadiče, kterými prochází při uskutečňování strojové instrukce.
- **Strojový cyklus (machine cykle)** představuje jednu dobu nebo častěji několik dob.
- **Čekací stav WAIT** spočívá v tom, že mikroprocesor dočasně odkládá pokračování ve strojovém cyklu (převzetí dat nebo předání dat na datovou sběrnici).
- **Neutrální stav HOLD** je odezvou mikroprocesoru na vnější požadavek, aby mikroprocesor převedl vývody datové i adresové sběrnice i některých řídicích vývodů do tzv. **neutrálního stavu**.

2.1.4 Mikrooperace, mikroinstrukce, mikroprogram a mikroprogramování.

- **Mikrooperace** je činnost procesoru nebo jeho logicky ohraničené části během jedné fáze (beat). Příklad. Posun dat mezi registry.
- **Mikroinstrukce** je kódovaný příkaz pro uskutečňování mikrooperace procesorem. Mikroinstrukce má obvykle dvě části: mikrooperační znak a adresu další mikroinstrukce.
- **Mikroprogramování** je sestavování mikroprogramu z mikroinstrukcí podle algoritmu požadované operace.

2.1.5 Strojová instrukce, pseudoinstrukce a makroinstrukce.

- **Strojová instrukce** je kódovaný příkaz k vykonání strojové operace. Úplný soubor strojových instrukcí tvoří tzv. **strojový jazyk** nebo **strojový kód**. Programování ve strojovém kódu je namáhavé a vzniklé programy jsou nepřehledné. Proto se častěji programuje v **jazyce symbolických adres (assembly language)**.
- **Pseudoinstrukce (direktiva)** je příkaz symbolického programovacího jazyku, který se nepřekládá do cílového programu, ale ovlivňuje překladač.
- **Makroinstrukce** je pokyn pro překladač, aby do cílového programu zařadil předem definovanou posloupnost instrukcí, tzv. **rozvoj makroinstrukce**, což je vlastně otevřený podprogram.

2.1.6 Podprogramy otevřené a uzavřené.

Podprogram (subrotina) je část většího programu.

Rozlišujeme dva druhy:

- **Otevřený podprogram** je posloupnost instrukcí, která se do programu opakovaně vkládá všude, kde se požaduje vykonání příslušných operací.
- **Uzavřený podprogram** je v programu zapsán jen jednou a řízení výpočtu se mu podle potřeby předává instrukcí volání (**call**).

Často používané podprogramy obecného významu se označují názvy **standardní podprogram** nebo **rutina** a bývají k dispozici v **knihovnách podprogramu**.

2.2 Adresy a adresování.

Adresa je označení místa uložení adresovaného slova nebo jeho části.

Adresy dat dělíme na:

- **Adresy zdrojové (source)**, které určují místo, odkud instrukce odebírá data, určená ke zpracování nebo přesunu.
- **Adresy cílové (destination)**, na které instrukce ukládá výsledky operací nebo data alespoň přesouvá.

2.2.1 Implicitní adresování a adresy v operačním znaku.

Pod implicitním adresováním máme na mysli takové označení místa, které buď není obsaženo v adresové části instrukce, nebo není v instrukci vůbec a vyplývá jen z typu instrukce.

Teď si uvedeme tři pojetí implicitního adresování:

- a) V instrukci není explicitně uvedena adresa místa, o které jde.
Příklad.
U mikroprocesoru 8080A instrukce **ADI D8** přičítá číslo **D8** k obsahu střádače. Předpokládá se, že jeden z operandů je ve střádači (**1. implicitní adresa**), výsledek součtu zůstane tamtéž (**2. implicitní adresa**), a že adresa další instrukce je uložena v čítači instrukce (**3. implicitní adresa**)
- b) Formát instrukce připouští explicitní adresu, ale v případě, že se neuvede, mikroprocesor předpokládá využití tzv. **default** adresy předem definované.
- c) Cílové nebo zdrojové místo se identifikuje čísly umístěnými v **operačním znaku**.
Příklad.
Některé instrukce se skládají z operačního znaku a přesto číselně určují místa operandu.
MOV r2,r1
Přesouvá operand z registru **r1** do registru **r2**.

2.2.2 Přímý operand a přímá adresa.

- Přímý operand, nebo adresa nultého řádu je sám operand obsažený v instrukci.

Příklad. **ADI D8**

D8 je přímý operand umístěný v instrukci.

- Přímá adresa (absolutní adresa nebo adresa prvního řádu), je pořadové číslo buňky hlavní paměti.

Příklad. **LDA A16** - přesune do střádače obsah buňky z hlavní paměti s adresou **A16**.

OUT A8 - předá obsah střádače na výstup s adresou **A8**.

2.2.3 Nepřímé adresy a nepřímé registrové adresování.

- Nepřímá adresa, nebo adresa druhého řádu je adresa, na které teprve najdeme adresu přímou.
- Nepřímé registrové adresování. Instrukce s tímto adresováním obsahuje ve svém operačním znaku příznak registru či skupiny registrů, v nichž je uložena adresa přímá nebo další nepřímá adresa operandu.

2.2.4 Adresování ukazateli.

Každý procesor obsahuje speciální registry, které slouží k adresování v hlavní paměti. Jejich obsah "ukazuje" na příslušné místo, a proto se označuje **ukazatel (pointer)**.

- Prvním z nich je čítač instrukcí **PC**, resp. **ukazatel instrukcí IP**.
- Druhým je **ukazatel zásobníkové paměti SP**.

Vedle ukazatelů-registrů se užívají i **ukazatelé v paměti**. Jsou to konstanty nebo proměnné uložené na definovaných místech.

2.2.5 Adresy relativní, autorelativní, segmentové a indexové.

- **Relativní adresa** umístěna v adresové části instrukce představuje tzv. **posunutí (offset, displacement)**, tj. údaj o relativní poloze operandu vzhledem k určitému vztažnému paměťovému místu. Adresa vztažného místa se nazývá **báze** nebo **bázová adresa** a je uložena v **bázovém registru**.
- **Autorelativní adresování**. V tomto případě je bázovou adresou adresa právě vykonávané instrukce a úlohu bázového registru hraje čítač instrukcí. **Autorelativní adresování** se nejčastěji používá u skokových instrukcí.
- **Segmentové adresování**. U mikroprocesoru 8086 se setkáváme s pojmem **segment**. Segmenty jsou souvislé, funkčně samostatné úseky programů nebo polí dat, které uchováváme ve vnější velkokapacitní paměti a jen podle potřeby je přesunujeme do hlavní paměti.
Segmentaci sledujeme:
 1. úsporu hlavní paměti,
 2. možnost dynamického přemístění jednotlivých částí programu nebo dat nezávisle na druhých.
- **Indexové adresování**. Je to velmi užitečná modifikace relativního adresování. Je založeno na použití tzv. **indexového registru**, což je bázový registr upravený tak, že se po použití automaticky inkrementuje nebo dekrementuje.

2.3 Architektura mikroprocesorů a mikropočítačů.

Architekturou procesoru rozumíme ideový návrh, tj. vytyčení hlavních principů jeho výstavby s přihlédnutím k potřebám programování, členění na nejdůležitější části a stanovení jejich vlastností atd.

K charakteristice architektury mikroprocesoru postačí: výčet registrů a jejich funkcí, popis vnitřních a vnějších sběrnic, způsob adresování a instrukční soubor. Analogicky lze chápat i architekturu mikropočítače. Ta je předurčena použitým mikroprocesorem a požadavky na prostředí.

Analogicky lze chápat i architekturu mikropočítače. Ta je předurčena použitým mikroprocesorem a požadavky na prostředí.

2.3.1 Zápisníková paměť.

Časté odkazy do hlavní paměti s využitím přímého adresování pro výběr a ukládání mezivýsledku a dat prodlužují a zpomalují programy. Tento nedostatek se potlačuje dvěma způsoby.

- Použití zkrácených způsobů adresování.
- Použití malé a rychlé pomocné paměti (**zápisníková**) začleněné do procesoru.

Zápisníková paměť (scratch pad memory) je skupina rychlých registrů, které se adresují jen příznaky v operačních znacích instrukcí.

Jeden nebo i více registrů obvykle slouží jako **střádače (accumulator)**, některé jako např. **indexové registry**.

2.3.2 Zásobníková paměť LIFO a volání podprogramu.

- **Zásobníková paměť (stack)** je speciální paměť RWM se sekvenčním přístupem. Charakteristický je pro ni výběr dat v opačném pořadí, než v jakém do ni byla ukládána. Zde je používán tzv. **ukazatel zásobníkové paměti SP**, což je vratný čítač s předvolbou.

Princip činnosti zásobníkové paměti je uveden na následujícím příkladě.

Příklad. Mějme tuto programovou sekvenci:

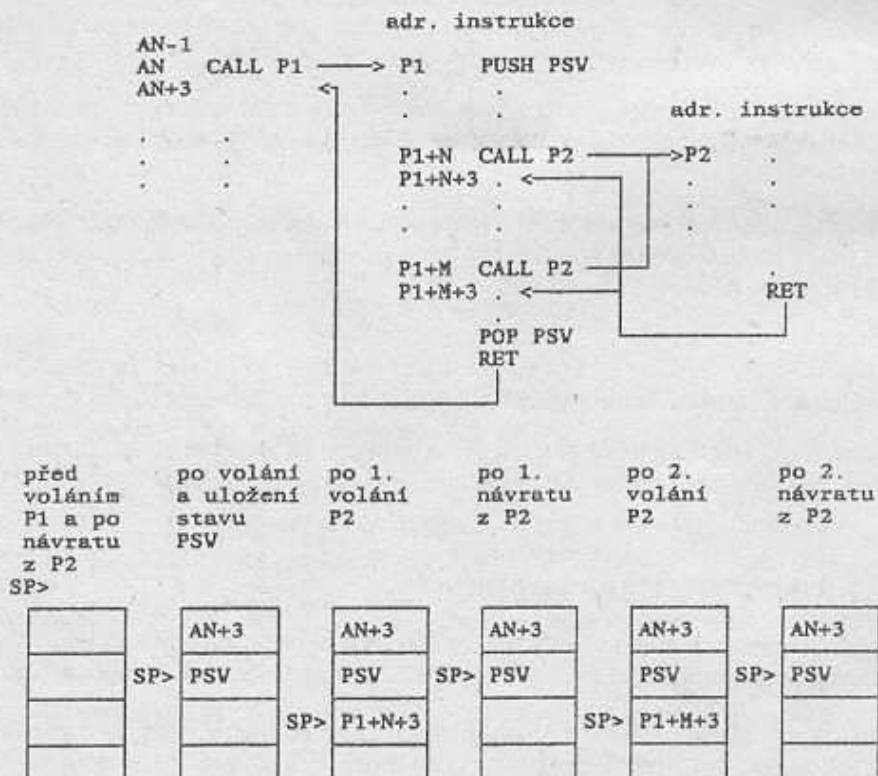
Hlavní program volá podprogram P1 a ten dvakrát volá podprogram P2.

Zásobník roste shora dolů.

Ukazatel zásobníku SP ukazuje na poslední uložené slovo.

V podprogramu P1 se ukládá stav PSW, v P2 nikoliv.

Schématické znázornění je následující:



Je v možnostech programátora vytvářet v hlavní paměti mikropočítače několik zásobníků a přecházet od jednoho k dalšímu podle potřeby.

2.3.3 Fronta (FIFO). ✓

Fronta (queue) nebo též FIFO (first in first out - první dovnitř, první ven) jsou názvy pro paměť RWM se sekvenčním přístupem. Data se vybírají v tom pořadí, v jakém do něj byla vložena.

2.3.4 Hierarchie sběrnic. ✓

Sběrnice (bus) je soustava vodičů, která zajišťuje přenos informace mezi více než dvěma účastníky.

Obecně dělíme sběrnici na 3 dílčí sběrnice:

- datová,
- adresová,
- řídicí.

Máme ještě tzv. sběrnice lokální, které zajišťují spojení v rozsahu konstrukčních modulů (spojových desek) a sběrnici systémovou.

V rámci mikropočítače na jedné desce máme celou hierarchii sběrnic. Nejníže stojí vnitřní sběrnice mikroprocesoru, od kterého pak konstruktér odvozuje lokální sběrnici modulu. Vývody mikroprocesoru však nejsou prostým pokračováním vnitřní sběrnice. Často se tu mění šířka toku dat, multiplexují se data s adresami nebo i stavovou a řídicí informací.

Návrh sběrnice systému patří k nejdůležitějším úkolům a spolurozhoduje o úspěšnosti architektury.

2.3.5 Von Neumanova a harwardská koncepce procesorů, zásobníkové orientované procesory.

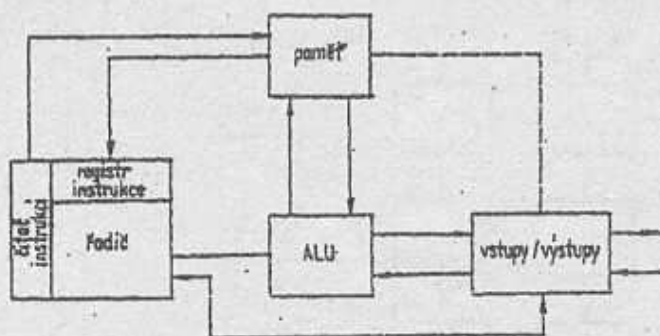
Procesory koncepce Von Neumanovy jsou stavěny dle obr. č. 2.1.

Procesor má k dispozici jen jednu množinu adres a preference adres pro uložení programu je v moci programátora. Tím vzniká možnost zpracovávat instrukce jako data, a tak modifikovat program.

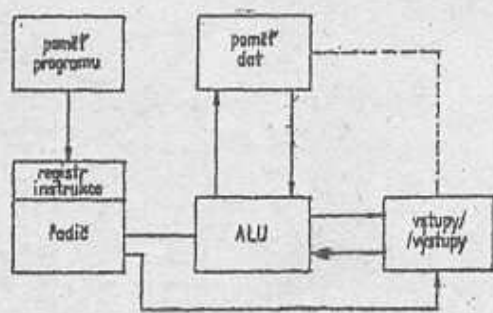
Při daném rozsahu paměti ji lze použít jak pro dlouhé programy s malým objemem dat, tak i pro krátké programy s hromadnými daty.

Podle koncepce Von Neumanovy se stavějí téměř všechny větší počítače, také mikroprocesory a mikropočítače.

Procesory harwardské koncepce jsou stavěny dle obr. č. 2.2.



Obr. č. 2.1 ✓



Obr. č. 2.2. ✓

Harvardská koncepce naopak předpokládá existenci dvou oddělených pamětí. V první paměti jsou uloženy programy a druhé jsou uložena proměnná data. Podle schématu z obrázku se stavějí jednoúčelové programové automaty a některé kapesní kalkulátory.

Zvláštní pozici mezi předešlými koncepcemi zaujímá princip tzv. **zásobníkově orientovaného procesoru**. Je to procesor, který přijímá z vnějšku data v přesně stanoveném pořadí do své zásobníkové paměti a výsledky operací ukládá zase do zásobníku. *- křivka autorů*

2.4 Pokročilejší architektury.

Základním nedostatkem jednoprocessorových počítačů je jejich pomalost, podmíněna sériovým způsobem činnosti. Takový procesor postupně čte instrukci, potom operandy, a pak vykonává příslušnou operaci a ukládá výsledek. Teprve potom čte další instrukci. Odstranění tohoto nedostatku nabízejí různé architektury [3,19] s využitím více procesorů pro jednu úlohu.

Jako takové mohou být:

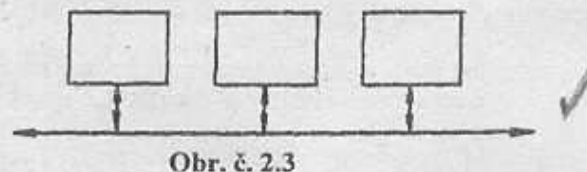
- řetězené procesory,
- paralelní soustava procesorů (multiprocessory),
- procesorová pole (maticové procesory).

2.4.1 Přesahování a řetězení.

- **Přesahování (overlapping)** spočívá v tom, že vykonávání některých instrukcí se v procesoru dokončuje v době náležející již další instrukci.
- **Řetězení (pipelining)** lze chápat jako přesahování v procesoru, který se skládá z řady sériově řazených subprocesorů. Jde o pokročilejší architekturu **řetězených procesorů**.

2.4.2 Multiprocessory a procesorová pole.

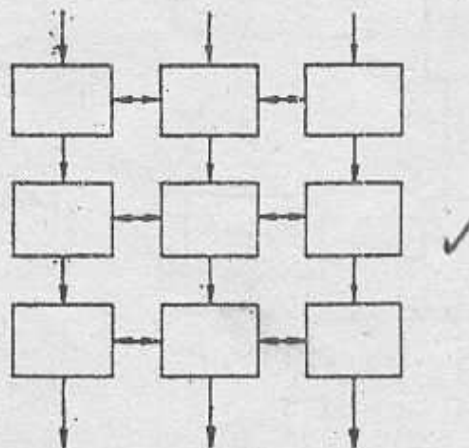
- **Multiprocessor** je centrálně řízený systém s více procesory se společnou hlavní pamětí a společným souborem periferních zařízení viz. obr. č. 2.3.



Obr. č. 2.3

Zde hovoříme o tzv. koprocesorech, např. matematický koprocesor atd.

- **Procesorové pole (maticový procesor)** je sestaveno ze stejných subprocesorů propojených mezi sebou i centrálním řadičem. Každý subprocesor je schopen vykonávat základní operace s uložením v jeho vlastní paměti, ale data se mohou přesouvat mezi sousedy v matici viz. obr. č. 2.4.



Obr. č. 2.4

3. STYK S PERIFERNÍM ZAŘÍZENÍM

Nasazení mikropočítačů má skutečný význam teprve ve spojení s periferními zařízeními, které zprostředkovávají spojení s okolním prostředím.

Součinnost mezi mikroprocesorem a periferním zařízením [2] se realizuje s použitím tří základních principů: **programovým řízením, systémem přerušování programu a metodou přímého přístupu do paměti.**

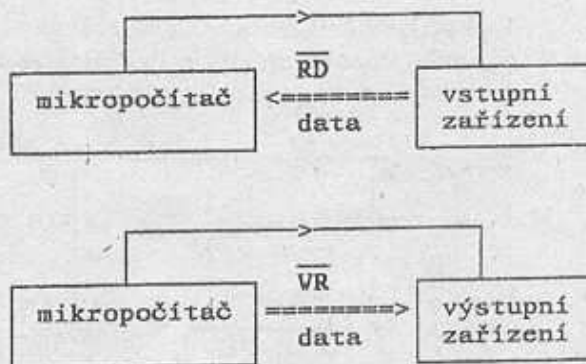
3.1 Technika V/V bran.

Vstupní/výstupní brána (**I/O port**), je obvod, který zprostředkuje předávání dat mezi sběrnici mikropočítače a periferním zařízením.

Používají se buď brány s pamětí, jejichž základem bývá tzv. **záchytný registr (latch)** s tří stavovým výstupem, nebo brány bez paměti, což jsou výkonové zesilovače (**budiče**) jednosměrné nebo řízené obousměrné.

Na obr. č. 3.1 je zobrazena technika podmíněného vstupu a výstupu dat.

V případě vstupu vyšle mikropočítač bitový signál \overline{RD} , čímž přikáže vstupnímu zařízení předat data do vstupní brány mikropočítače.



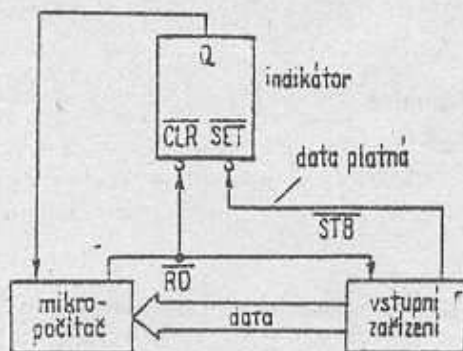
Obr. č. 3.1

Při výstupu mikropočítač vyšle současně data i signál \overline{WR} a výstupní zařízení převezme data.

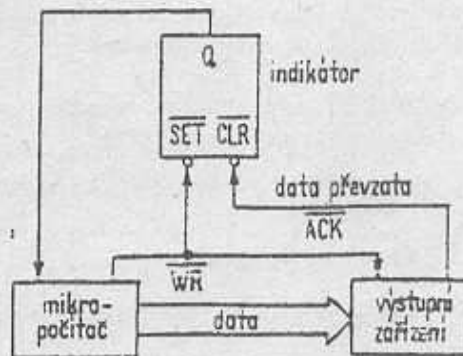
Tento způsob je mimořádně jednoduchý a předpokládá, že periferní zařízení je stále připraveno komunikovat.

Další obrázky znázorňují techniku podmíněného vstupu/výstupu. Na obr. č. 3.2 vstupu probíhá následovně. Jestliže vstupní zařízení poskytuje platná data, strobovacím impulsem \overline{STB} nastaví Q do stavu 1, který pro mikropočítač představuje **indikátor (flag)**. Po zjištění $Q=1$, mikropočítač impulsem \overline{RD} data převezme a nuluje indikátor.

Analogicky při výstupu podle obr. č. 3.3 [2] mikropočítač vyšle impuls \overline{WR} a přepíše data do výstupního zařízení a nastaví indikátor. Výstupní zařízení po převzetí dat impulsem \overline{ACK} indikátor nuluje, a tím mikropočítači sděluje, že může vyslat další data.



Obr. č. 3.2



Obr. č. 3.3

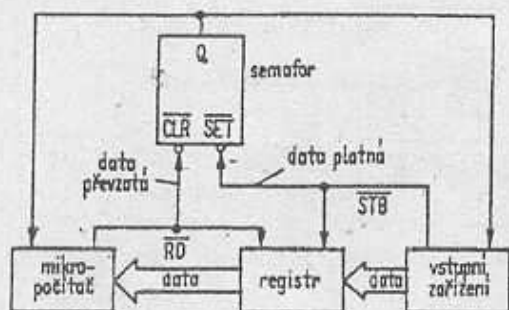
V obou případech jde o **neúplný (jednosměrný korespondenční) režim**, kdy indikátor informuje o zahájení nebo skončení přenosu pouze mikropočítač, a kdy vysílač dat je povinen data udržovat.

Na obr. č. 3.4, 3.5 [2] je znázorněna technika **úplného (obousměrného korespondenčního) režimu**, využívající jednak **vyrovnávací paměť-registr**, **klopný obvod** pracující jako **semafor** (jeho stav je testován vysílačem i přijímačem dat, a tak je možné **vzájemné blokování (interlock)**).

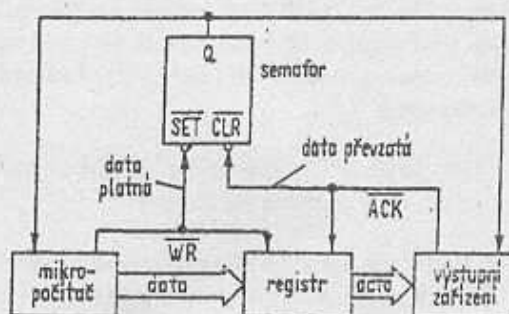
Pokud je registr plný, semafor je ve stavu **1** a vstupní zaří zení další data nevyšle. Je-li registr prázdný, semafor je **nulový** a strobovacím signálem **STB** lze do registru vyslat data a zároveň semafor nastavit.

Mikropočítač testuje semafor a při jeho hodnotě **1** impulsem **RD** převezme data z registru a také semafor nuluje.

V případě výstupu dle obr. 3.5 [2] mikropočítač nejdříve testuje semafor, zda není nulový. V kladém případě mikropočítač impulsem **WR** vyšle data do registru a nastaví semafor do stavu **1**. Výstupní zařízení tak zjistí, že data jsou v registru a může je impulsem **ACK** převzít a semafor vynulovat.



Obr. č. 3.4



Obr. 3.5

3.2 Programové řízení styku.

Při tomto způsobu je styk procesoru s periferním zaří zením řízen programovými prostředky. Program prostřednictvím **stavové vstupní brány** postupně testuje, zda a která vstupní/ výstupní zaří zení jsou připravena vyslat data, a která data přijmout.

Při zjištění připravenosti procesor volá V/V podprogram (**driver**), který pak zajistí vstup popř. výstup dat.

Souběžně podprogram generuje a povelovou bránou vysílá řídicí signály odpovídající typu periferního zaří zení a jemu odpovídajícímu způsobu styku i uspořádaní V/V brány.

Programové řízení je výhodné z hlediska úspory obvodu. Je však pomalé, neboť počítač je příliš zatěžován periodickým testováním stavů, a to i v případech, kdy k přenosům nedochází. Proto se konstruktéři počítačů uchylují k následujícím postupům:

- Přenášejí úlohu styku na pomocný tzv. **komunikační procesor**. Tím se hlavní procesor zbaví větší části zátěže spojené se stykem.
- Přesunou část stykové úlohy z programu na obvody.

3.3 Přerušení programu.

Přerušení (**interruption. trap**) programu je odezva procesoru na požadavek o programové obsluhu.

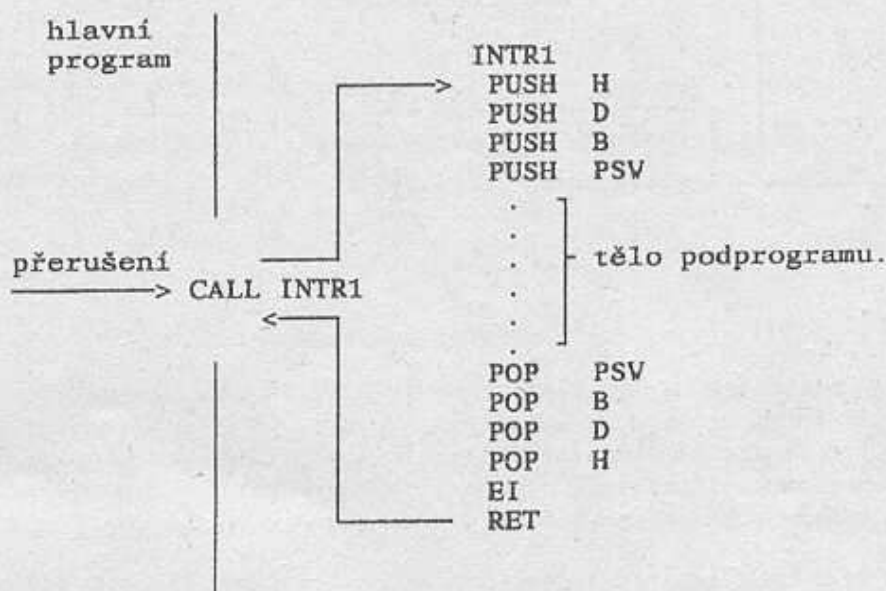
Nechť periferní zaří zení vyžaduje spolupráci s mikropočítačem. Pak zaří zení žádá o obsluhu **přerušovací m signálem**, který se buď přímo, nebo pomocí zprostředkujících obvodů zavádí na bitový vstup mikropro-

cesoru, označovaný **INT**, **INTR**, **TRAP** apod. Mikroprocesor při každé instrukci signál automaticky testuje a v pozitivním případě postupuje takto:

Dokončí rozpracovanou instrukci, znemožní přijetí dalších žádostí o přerušení nulováním interního indikátoru, vyšle signál o akceptování požadavku a převezme z datové sběrnice **instrukci přerušení**. Tato se načte z paměti programu, ale na sběrnici ji dodá přerušující zařízení jako odezvu na signál **INTA**.

Jako přerušovací instrukce se obvykle uplatňují instrukce volání buď typu **CALL**, **RSTn** u 8080 a nebo **INTn** u 8086/88).

Ukázka programu demonstrující přerušení programu.

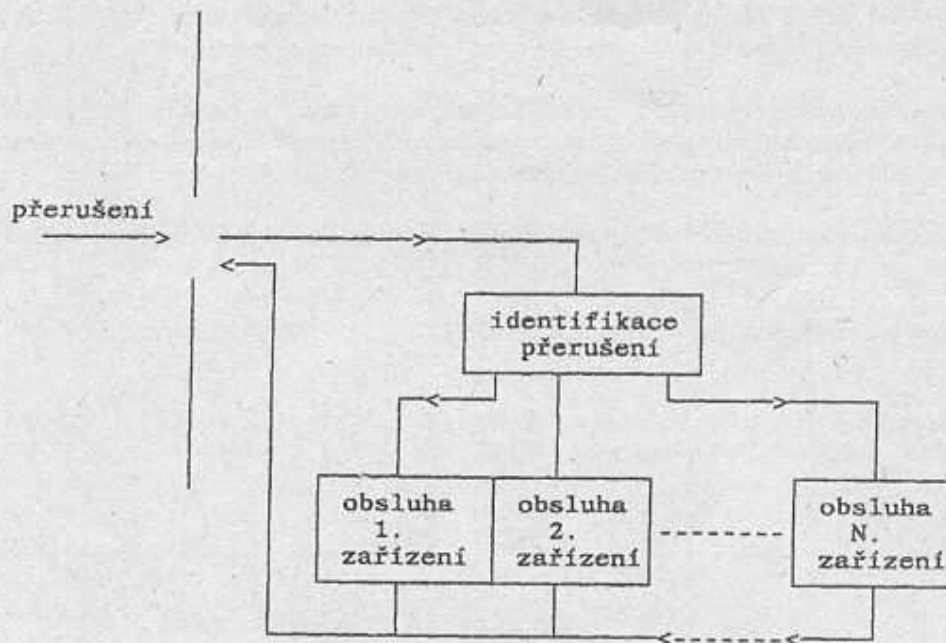


kde: **CALL** - volání přerušení,
INTR1 - podprogram,
CALL INTR1 - volání přerušovacího podprogramu **INTR1**,
PUSH - uložení stavu mikroprocesoru,
POP - obnovení stavu mikroprocesoru,
EI - povolení přerušení,
RET - návrat z poprogramu.

Při větším počtu zdrojů žádosti o přerušení vzniká problém **počtu přerušovacích vstupů a priority jejich obsluhy**. Většina mikroprocesorů má jeden nebo dva takové bitové, vstupy, přičemž druhý bývá nemaskovatelný.

Problém počtu a priority přerušení lze řešit **třemi způsoby**.

- a) S programovou identifikací.
 K přerušovacímu vstupu mikroprocesoru je připojen signál logického součtu externích signálů přerušení, a ty se zároveň přivádějí na **vstupní bránu požadavku**. Po akceptování žádosti o přerušení se vyvolá přerušovací podprogram, který čistě programově podle brány požadavků identifikuje přerušující zařízení a přejde do odpovídající větve viz. obr. č. 3.6.



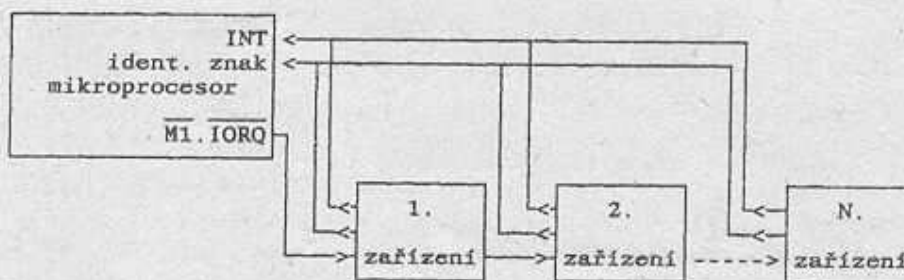
Obr. č. 3.6

Podle pořadí připojení na bráně požadavků lze určit i prioritu. Není to však dokonalé řešení, neboť při nesoučasných požadavcích lze zablokováním dalších přerušení preferovat jen obsluhu přerušení nejvyšší priority. Právě s ohledem na něj však nelze takto chránit ostatní požadavky. Přesto se tento nenáročný způsob uplatňuje v případech, kdy máme jen jedno přerušující zařízení s vysokými nároky na prioritu a rychlost obsluhy, zatímco požadavky ostatních zařízení nejsou tak naléhavé.

Teoreticky je možné i programovým způsobem zajistit úplné prioritní zpracování. Je to však nevýhodné z hlediska času i nároku na program, a tak se dává přednost dalším způsobům.

b) Se sériovou obvodovou identifikací.

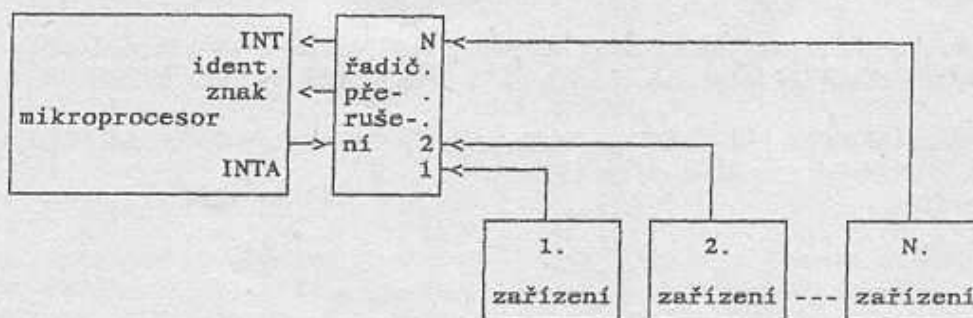
Mikroprocesor po příchodu přerušovacího signálu očekává identifikační znak, generovaný stykovými obvody, který jej nasměruje do žadoucího podprogramu. Stykové obvody jsou však spojeny v pořadí priority podle obr. č. 3.7 a sériově tak, že se nemůže uplatnit další přerušovací signál ani identifikační znak nižší priority.



Obr. č. 3.7

c) S řadičem přerušení.

Specializované pomocné obvody, nazývané **řadiče přerušení**, redukují potřebu aktivní účasti stykových obvodů na fázi identifikace i rozhodování o obsluze podle priority zařízení. Periferní zařízení pouze vysílají požadavky na obsluhu do řadiče přerušení, a ten žádá mikroprocesor o přerušení jen tehdy, je-li prioritní úroveň nového požadavku vyšší než úroveň požadavku právě obsluhovaného viz. obr. č. 3.8.



Obr. č. 3.8

3.4 Přímý přístup do paměti - DMA.

Některá zařízení vyžadují tak vysokou rychlost přenosu dat, že přenos nemůže být řízen mikroprocesorem. Tehdy se používá **přímý přístup do paměti - DMA (direct memory access)**, kdy se data přenášejí mezi periferním zařízením a příslušným polem hlavní paměti bez účasti mikroprocesoru. Přenos kanálem DMA je podmíněn předchozí činností mikroprocesoru, který jej iniciuje, tj. určí směr toku dat, počet slabik v přenášeném bloku v hlavní paměti apod.

Existují vlastně tři způsoby řízení přenosu DMA.

1. způsob - řadič uplatní před zahájením přenosu DMA na vstupu HOLD mikroprocesoru signál, kterým přiměje převést své adresové, datové a řídicí vývody do neutrálního stavu. Tím uvolní sběrnici, jejich řízení převezme řadič DMA. Po ukončení přenosu se řízení sběrnice opět ujímá mikroprocesor a pokračuje v běžné činnosti.
Tento způsob zpomaluje probíhající program.
2. způsob - po ukončení strojového cyklu se zastaví generování hodinových signálů mikroprocesoru a odpojí se jeho budiče sběrnice, nejčastěji na jediný dílčí přenos dat.
Tento způsob zpomaluje probíhající program.
3. způsob - jednotlivé dílčí přenosy se uskutečňují v časových intervalech, v nichž mikroprocesor pracuje, ale nekomunikuje se spolupracujícími zařízeními.
Přenos DMA neovlivňuje rychlost mikroprocesoru. Zde máme přísně synchronní činnost mikroprocesoru a řadiče DMA.

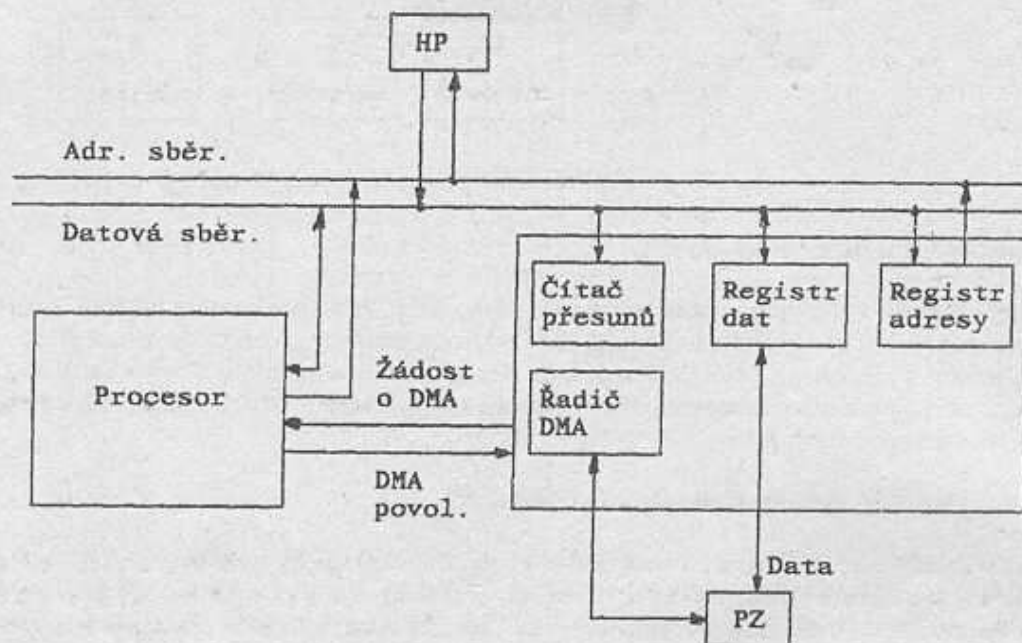
Příklad struktury bloku DMA je uveden na obr. č. 3.9. Pro styk se sběrnicemi jsou určeny tři registry, označené jako **registr dat**, **registr adresy** a **čítač přesunů**:

- **registr dat**: obsahuje slovo, které má být přesunuto z PZ (periferní zařízení) do HP (hlavní paměť) nebo naopak, délka tohoto slova odpovídá šířce toku dat na datové sběrnici systému.
- **registr adresy**: obsahuje adresu v HP, která se má při přesunu zúčastnit, tzn., že na tuto adresu bude zapsáno slovo, obsažené v registru dat v bloku DMA, nebo z ní bude přečteno slovo, které má být do registru dat v bloku DMA zapsáno.
- **čítač přesunů**: obsahuje požadovaný počet slov, která mají být ještě přesunuta v rámci jednoho spojení mezi PZ a HP. Po každém uskutečněném přesunu se obsah tohoto čítače zmenší o jedničku, takže jeho nulová hodnota signalizuje, že spojení skončilo.

Blok DMA může pracovat ve dvou základních režimech. Buď přesouvá data mezi PZ a HP v blocích, nebo jednotlivě. Blokovaný přenos se uskutečňuje tak, že blok DMA nepřetržitě řídí sběrnici po celou dobu, dokud nemá **vynulován čítač přesunů**. Jednotlivé přesuny se uskutečňují v režimu tzv. **odebrání cyklu (cycle stealing)**. V tomto režimu se procesor zastaví pouze na dobu jednoho cyklu HP a po provedení jednoho přesunu slova uvolní blok DMA sběrnici, takže procesor může pokračovat ve své činnosti.

Celá operace přímého přístupu do paměti obvykle probíhá v několika krocích:

1. Před zahájením vlastního přesunu musí procesor "naprogramovat" blok DMA, tzn. , že operacemi V/V zapíše do čítače přesunů požadovaný počet přesouvaných slov a do registru adresy adresu začátku oblasti v HP.
2. Blok DMA spustí PZ a čeká, až bude PZ připraveno přijmout nebo vyslat data. Pro další výklad budeme předpokládat, že DMA řídí operaci vstupu dat z PZ.



Obr. č. 3.9 Blok přímého přístupu do paměti.

3. Procesor obvykle nemůže reagovat na žádost o DMA okamžitě, protože musí nejprve dokončit strojový cyklus. V systémech, které používají přímý přístup do HP, jsou pro procesor přesně definovány okamžiky, kdy může dojít k přímému přístupu.
4. Když procesor rozhodne, že může dojít k přímému přístupu do paměti, vyšle vybrané jednotce signál "DMA povoleno" a uvolní sběrnice. Vybraná jednotka pak vyšle na adresovou sběrnici obsah svého registru adresy, na datovou sběrnici obsah svého registru dat a čeká na provedení jednoho cyklu paměti. Pak zvětší obsah registru adresy o jedničku a současně zmenší obsah čítače přesunů o jedničku. Není-li obsah čítače přesunů dosud nulový, testuje, zda PZ již přesunulo do registru dat nové slovo. Pokud ano, pokračuje rovnou v přesunu, tedy vyšle novou adresu na adresovou sběrnici a na datovou sběrnici nové slovo a dále změní obsahy registru adresy a čítače přesunů. V opačném případě dočasně ukončí přesun dat a předá řízení procesoru.
5. Procesor pokračuje v provádění svého programu až do okamžiku, kdy některý blok DMA vyšle novou žádost o DMA.
6. Jestliže je obsah čítače přenosů nulový, blok DMA končí celý přesun a uvolní sběrnice. Navíc může vyslat žádost o přerušení, čímž si vyžádá programový zásah procesoru, spočívající obvykle v novém naprogramování jeho vnitřních registrů.

Dosud jsme předpokládali, že řadič DMA jen adresuje hlavní paměť a synchronizuje činnost mezi sběrnici a perif. zařízením, zatímco data jdou mimo něj.

Existují však i takové přenosové subsystémy, které jsou založeny na principu DM, ale mají i jistou inteligenci a jsou schopny data i kontrolovat, opakovat a přepracovávat. Jde o kanál, což je speciální procesor určený ke spojení hlavního procesoru s periferními zařízeními. Řídí se **kanálovým programem (kanály: multiplexní a selektorové)**.

4. MIKROPROCESORY ARCHITEKTURY CISC A RISC

Architektura procesoru je výsledkem kompromisu mezi požadovanými parametry a chováním systému vzhledem k programům, které má procesor vykonávat. Jednou z možných cest dosažení vyšší rychlosti zpracování informace je přechod k architektuře RISC (Reduced Instruction Set Computer) [3,19].

Architektura procesoru CISC

Vzrůstající nároky na aplikace výpočetního systému se odrážejí v rostoucí složitosti architektury procesorů, ve změnách jejich instrukčního souboru. Instrukční soubor představuje množinu elementárních funkcí, které může vyžadovat program při realizaci aplikačních problémů.

Instrukce strojového kódu mají pro procesor stejný význam jako klávesy pro kalkulačku. Každá kalkulačka má klávesy pro sčítání, odečítání, násobení a dělení. Některé mají ale i další klávesy, jako např. klávesy pro výpočet procent, druhé odmocniny, goniometrických funkcí atd. Navíc mají některé kalkulátory klávesy (instrukce) použité pro úzce specifické účely, jako např. výpočet střední kvadratické odchylky, výpočet úrokové sazby atd. Tyto složité funkce kalkulátoru jsou přitom realizovatelné určitou posloupností funkcí jednodušších (operacemi součtu, součinu, násobení či dělení).

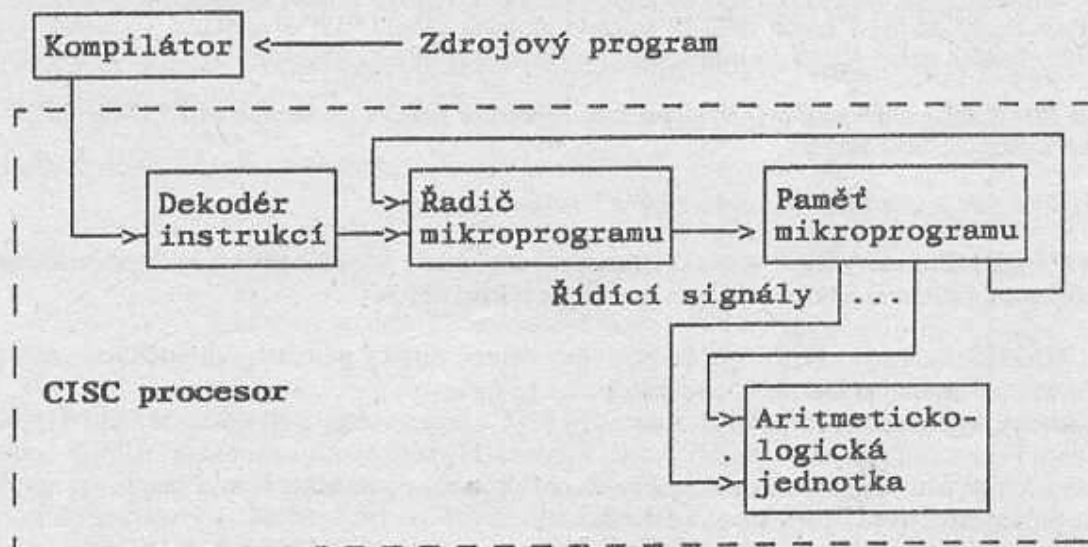
Obečným trendem ve vývoji procesorů byla tvorba stále složitějších a komplexnějších instrukčních souborů s cílem pokrýt a podporovat co nejširší spektrum aplikací.

Instrukce realizují operace bez velkých nároků na přenosy dat mezi procesorem a pamětí a poskytují rozsáhlý repertoár operací s adresami a instrukcí. Tím byly eliminovány nedostatky technických prostředků (drahé a pomalé paměti, sběrnice s nízkou přenosovou rychlostí dat a instrukcí). Počet instrukcí neustále stoupal a jednotlivé instrukce se stávaly stále složitějšími. Složitost instrukcí a bohatost adresovacích módů vedly k nutnosti mikroprogramové implementace instrukčního souboru. S tím ale vzrůstaly problémy jednak s implementací instrukcí na křemíkovém čipu a jednak s překladem programů z jazyka vysoké úrovně do instrukčního souboru.

Dalším problémem se stával také výběr optimálních posloupností strojových instrukcí k realizaci konstrukcí jazyka vysoké úrovně atd.

Procesory této koncepce se nazývají procesory s komplexním instrukčním souborem - CISC (Complex Instruction Set Computer).

Architektura procesoru CISC je uvedena na obr. č. 4.1.



Obr. č. 4.1 Architektura procesoru CISC

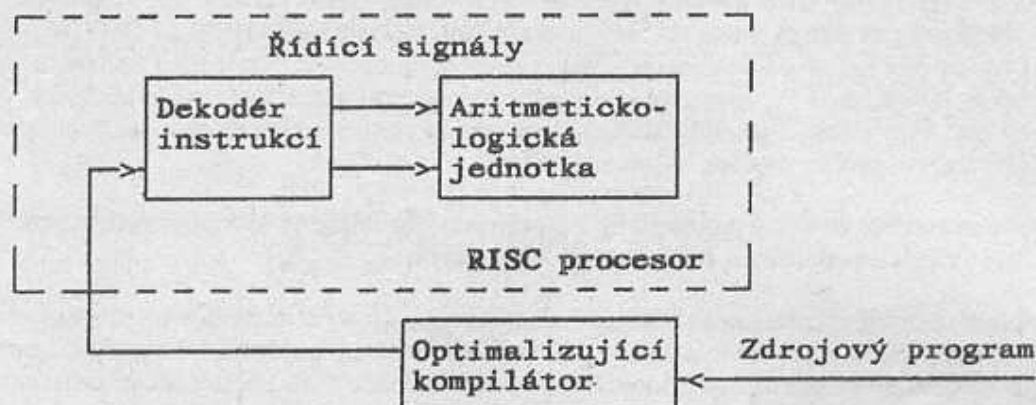
Architektura procesoru RISC

Pokroky technologie v následujícím období umožnily širší sběrnice, rychlejší tok instrukcí do procesoru, rychlejší tok dat, levnější paměti s kratší dobou přístupu a větší kapacitou. Pomocí optimalizujícího překladače je možno přetvářet program napsaný v problémově orientovaném jazyku do jednoduchých strojových instrukcí, které je schopen procesor rychle zpracovat.

Cílem bylo dosáhnout zpracování jedné instrukce během jednoho strojového cyklu procesoru.

Tomu byl podřízen i návrh architektury procesoru. Procesory této koncepce se nazývají procesory s redukovaným instrukčním souborem RISC (Reduced Instruction Set Computer).

Architektura procesoru RISC je uvedena na obr. č. 4.2.



Obr. č. 4.2 Architektura procesoru RISC

4.1 Historie vývoje procesorů RISC

Základní výzkum v oblasti vývoje procesorů RISC je veden již od roku 1975. První procesory RISC, které by bylo možno porovnávat s dnešními výrobky, byly vyvinuty Seymorem Crayem koncem 60. let pro firmu Control Data Corp.

Začátkem 70. let vzniká u firmy IBM podobný projekt. Výsledkem byl minipočítač s typovým označením IBM 801.

K dalším průkopníkům patří Dave Patterson, profesor university Berkeley a John Hennessy z university Stanford. Prof. Patterson byl v letech 1980-82 autorem projektů RISC-I, RISC-II a SOAR (Smalltalk On A RISC), které se později staly výchozí základnou pro vývoj procesorů SPARC (Scalable Processor Architecture).

Na Stanfordské universitě byla vyvíjena řada procesorů RISC s označením MIPS (Microprocessor without Interlocked Pipeline Stages).

Architektura procesorů RISC prošla zhruba 3 vývojovými etapami:

1. Etapa 1975-1982. Procesory RISC mají experimentální charakter a ve velké většině byl realizován pouze samotný procesor (Central Processing Unit). Zde patří počítač IBM 801.
2. Etapa 1982-1985. Zahrnuje vývoj a výrobu první generace prakticky použitelných počítačů osazených procesorem s redukovaným instrukčním souborem.
K charakteristickým vlastnostem počítačů s procesory RISC vývojové etapy patří výkon do 5 MIPS (Million Instructions Per Second), realizace jednotky styku s pamětí (Memory Management Unit - MMU), aplikace vyrovnávacích pamětí, implementace koprocenů pro operace v pohyblivé řádové čáře, pro grafické operace, pro zajištění styku s periferními zařízeními atd.
Procesory druhé etapy vykazují některé rysy architektury procesorů CISC (větší počet instrukcí nebo bohatost adresových režimů).

3. Etapa 1985 - dnes. Zahnuje rozvoj procesorů: Pyramid 9800, T800, T9000, MIPS R3000. Navíc dochází k vývoji nových procesorů, jako jsou např.: Motorola 88000, Intel i860, i960, SPARC, HP Precision Architecture.

Při implementaci procesorů RISC byly použity nové technologie (ECL - Emitter Coupled Logic), ale i materiály (arzenid galia - GaAs). Dochází k rozsáhlému využívání principů **zřetězených a vektorových výpočtů** a výkon okolo **desítek MIPS**.

Porovnání výkonnosti procesorů RISC a CISC tab. č. 4.1.

Tabulka č. 4.1 Porovnání výkonnosti procesorů RISC a CISC

Typ procesoru	Taktovací kmit. v MHz	Poč. strojových cyklů na instr.	Výkon v MIPS
MIPS R3000	25	1.25	20
MIPS R2000	16.7	1.4	12
AM 29000	25	2.0	12.5
SPARC	16.7	2.1	8
IBM RT PC	6	3.0	2
Clipper	33	6.6	5
i80386	20	6.0	3.5
MC 68020	16	7.0	2.3
VAX 8550	22.2	7.0	3.0
☐ VAX-11/780	5	10	0.5

Tabulka uvádí porovnání výkonnosti některých procesorů RISC z počátku třetí etapy a jim historicky odpovídajících procesorů CISC. Porovnání procesorů RISC a CISC není zcela jednoznačné. Procesor RISC může vykonat při realizaci téhož programu o 30 - 100 % instrukcí více než procesor CISC.

Výrazná podpora pro rozvoj technologie RISC a pro její rozšíření se očekávala od dvou vzniklých skupin výrobců výpočetní techniky a programového vybavení.

První skupinu tvořilo sdružení více než 60 výrobců technických i programových prostředků pod názvem **Advanced Computing Environment (ACE)**, založené v dubnu 1991 firmou **Microsoft**.

Druhou skupinou bylo sdružení kolem firmy IBM. Bohužel, následující vývoj v oblasti výpočetní techniky nepotvrdil očekávání původně stanovených cílů.

4.2 Definice architektury RISC

Architektura RISC realizuje větší výpočetní výkon za nižší cenu a s jednodušším instrukčním souborem.

Celkový čas T_c , realizace určitého programu, lze vyjádřit vztahem

$$T_c = T \cdot \sum_{i=1}^N C_i$$

N - počet vykonaných instrukcí procesoru

C_i - počet strojových cyklů potřebných na vykonání i -té instrukce

T - doba jednoho strojového cyklu

Cílem je snížit hodnotu T_c na minimum.

Architektura RISC zjednodušením instrukčního souboru potenciálně zvětšuje hodnotu N , ale zbylé dva členy - C_i a T jsou snižovány, a tím je dosaženo zkrácení doby vykonání aplikačního programu. U procesoru RISC je u většiny $C_i = 1$.

Dalšími možnostmi zvýšení výkonnosti jsou:

- snižování hodnoty T
- redukce hodnoty N při zachování hodnoty T
- redukce C_i na hodnotu menší než 1

Hodnota celkového času realizace určitého programu T_c je snižována takto:

- implementace vyrovnávací paměti pro instrukce, případně čtení několika instrukcí současně
- podpora pevného formátu instrukce (omezuje tím dobu výpočtu adresy následující instrukce)
- jedna instrukce specifikuje dvě a více nezávislých akcí (paralelismus operací)
- jedna instrukce determinuje vykonání sekvence několika elementárních instrukcí

Na základě uvedeného rozboru je možno formulovat následující tři základní principy architektury procesorů RISC.

- regularita a jednoduchost souboru instrukcí, které dovolují opakované použití jednoduchých bloků obvodů pro provádění většiny instrukcí
- v každém strojovém cyklu končí, pokud možno, provedení jedné instrukce
- instrukce mají pevnou délku, formát instrukce je, pokud možno, neměnný

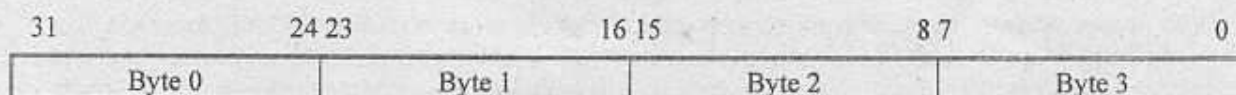
Charakteristické znaky architektury procesorů RISC je možno definovat následovně:

- minimální instrukční soubor (asi 80 až 150 instrukcí)
- jednoduché způsoby adresování
- jeden nebo málo formátů instrukcí
- řízení jednoduchou pevnou logikou
- jednocyklové strojové operace
- styk s pamětí výlučně prostřednictvím instrukcí Load/Store
- datové operace pouze nad registry
- rychlé paměti na uložení programů a operandů (paměť typu Cache, velký počet programově přístupných registrů pro operandy)
- zřetěžená realizace instrukcí
- optimalizující kompilátor

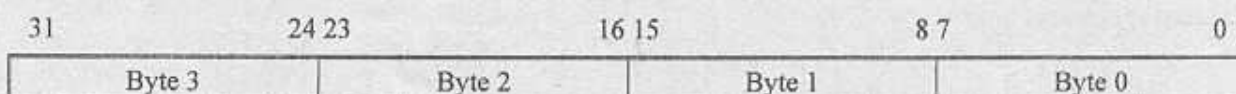
4.3 Některé základní pojmy.

V této kapitole jsou shrnuty některé pojmy, které jsou potřebné pro pochopení terminologie používané při popisu vlastností procesorů RISC.

- **Binární kompatibilita** (Application Binary Interface) definuje standard systémového rozhraní na strojové (binární) úrovni. Dodržení tohoto standardu umožní implementaci programu na různých počítačích se stejným typem procesoru bez nutnosti jeho rekompilace.
- **Atomická instrukce** (Atomic Instruction) zabezpečuje výhradní přístup a možnost modifikace příznaku (např. semaforu) po celou dobu své realizace procesorem. Výhradní použití příznaku zajišťuje nemožnost modifikace tohoto příznaku jakoukoliv jinou instrukcí v době vykonání atomické instrukce.
- **Uspořádání bitů** - pro uložení 32-bitových slov jsou používána dvě základní uspořádání bytů, viz. obr. č. 4.3.



Uspořádání big-endian



Uspořádání little-endian

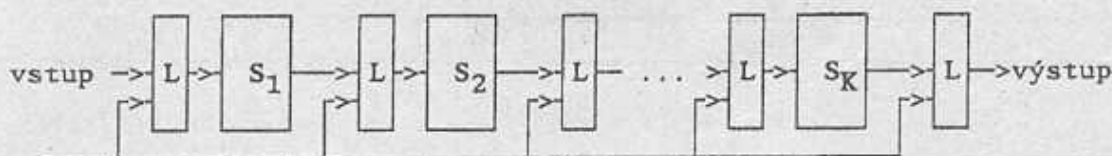
Obr. č. 4.3 Uspořádání bytů v 32 bitovém slově

- **Příznakové bity** procesoru (Condition Codes) umožňují zjistit, zda byl výsledek porovnání kladný, záporný, nulový nebo zda došlo k přetečení atd.
- **Listová procedura** (Leaf Procedure) je taková procedura, která ve svém těle nevolá žádnou jinou proceduru.
- **Jednotka správy paměti** (Memory Management Unit - MMU) transformuje virtuální adresu, která je používána programem, na fyzickou adresu paměti, která je použita technickými prostředky. Výstup z MMU je použit na fyzické adresování paměti.
- **Stránka** (Page) je nejmenší logická jednotka velikosti paměti. Stránka je používána při operacích s virtuálními adresami. Systém realizuje transformaci virtuálních adres na fyzické pro každou stránku.
- **Registrovým oknem** (Register Windowing) je nazývána taková organizace registrů procesoru, která realizuje kruhovou vyrovnávací paměť (Circular Buffer).
- **Semafor** (Semaphore) je návěští (příznak), který je realizován technickými (stavové hradlo) nebo programovými (příznakový bit) prostředky.
- **Příznaková aritmetika** (Tagged Arithmetic) realizuje konzistentní aritmetické operace nad daty, které nemají definovaný typ. U některých prog. jazyků vysoké úrovně (Smalltalk, Lisp) nejsou poskytnuty možnosti deklarace datových typů. V době kompilace programu není možné kontrolovat správnost použití datových typů v arit. operacích. Tato kontrola musí být uskutečněna až v době běhu programu. Příznaková aritmetika je prostředkem uvedené kontroly.
- **Instrukce testuj a nastav** (Test-And-Set) testuje obsah paměťového prvku (příznaku-semaforu) a podle jeho hodnoty ho aktualizuje.
- **Transformační vyrovnávací paměť** (Translation Lookaside Buffer - TLB) je vyrovnávací paměť jednotky MMU, která obsahuje dvojice "virtuální adresa, fyzická adresa", které byly naposledy použity při transformaci virtuální adresy na fyzickou.
- **Součinitel úspěšnosti TLB** (TLB Hit Rate) vyjadřuje poměr úspěšných přístupů k celkovému počtu přístupu k TLB. Vyjadřuje přínos implementace TLB v procesu získávání fyzické adresy z adresy virtuální.
- **Koncepce virtuální paměti** (Virtual Memory) umožňuje považovat paměť výpočetního systému za souvislý, jednotný a rozsáhlý adresový prostor, který není ovlivněn počtem aplikací, množstvím paměti, které tyto aplikace požadují atd. Programátor vytváří program ve virtuální paměti. Transformaci virtuální adresy na fyzickou adresu vykonává jednotka správy paměti MMU. Uvedená transformace je z pohledu programátora plně transparentní.

4.4 Zřetěžené zpracování informace

Koncepce zřetěženého zpracování informace je koncepcí široce používanou i mimo procesory RISC (viz. 180386, 180486). V tomto procesoru je tato koncepce zpracování jednou z nosných technik pro dosažení vyššího výpočetního výkonu procesoru.

Koncepce zřetězeného zpracování ve výpočetním systému je podobná koncepci výrobní linky. Na dosažení zřetězení je nutné rozdělit úlohu do posloupnosti dílčích úloh, z nichž každá může být vykonána samostatnými technickými prostředky. Jednotlivé části pracují souběžně. Algoritmus se vykonává ve zřetězené jednotce složené z lineárního řetězce modulů. Každý modul vykonává určitou podúlohu realizovaného algoritmu, přičemž funkčně závisí na předcházejícím modulu. Podúlohy v modulech jsou realizovány v ideálním případě za stejný časový interval. V opačném případě se nejpomalejší modul stává tzv. "úzkým místem" Princip zřetězeného zpracování ilustruje obr. č. 4.4.



synchronizace

L ... vyrovnávací registr (latch), S_i ... i-tý stupeň řetězce, T_i ... i-tá podúloha v j-tém běhu úlohy.

Obr. č. 4.4 Zřetězený lineární procesor

Obrázek vyjadřuje situaci, kdy je daná úloha T rozdělena na množinu dílčích podúloh:

$$T = (T_1, T_2, \dots, T_m)$$

z nichž daná úloha T_k nemůže začít dřív, než T_{k-1} ukončí svoji činnost.

Jednotlivé stupně jsou odděleny vyrovnávacími registry, které uchovávají mezivýsledky mezi jednotlivými stupni a zároveň kompenzují případné rozdíly v době zpracování podúloh v jednotlivých modulech. Je-li výpočet v modulech vykonán a výsledky jsou uloženy ve vyrovnávacích registrech, potom pod řízením společného časování dojde k přepsání údajů z vyrovnávacího registru do následujícího modulu. Je vhodné, aby doba zpracování v jednotlivých modulech řetězce byla identická, pak řetězec pracuje synchronně s plným využitím výpočetních modulů.

Uvedené řešení lze použít tehdy, když se daný algoritmus dá dekomponovat na množinu za sebou následujících modulů s přibližně stejnou dobou zpracování. Moduly jsou pak realizovány v segmentech zřetězené jednotky technickými prostředky.

Doba realizace jedné podúlohy modulem S_i je určena T_i . Časové zpoždění přenosu informace vyrovnávacím registrem je T_1 . Potom časový interval T k -stupňového řetězce je dán vztahem:

$$T = \max_{1 \leq i \leq k} (T_i) + T_1 = T_{\max} + T_1$$

$f = 1/T$ - frekvence výstupu řetězce, tj. počet výsledků na výstupu za jednotku času.

Jakmile je lineární řetězec naplněn, obdržíme jeden výstup v každém časovém intervalu T , a to nezávisle na počtu stupňů v řetězci.

V ideálním případě vykoná lineární " k "-stupňový řetězec " n " úloh za T_k časových intervalů

$$T_k = k + (n-1)$$

k - časových intervalů je použito na naplnění řetězce nebo na kompletní vykonání první úlohy
 $n-1$ - časových intervalů je zapotřebí pro zbývajících $n-1$ úloh

Tentýž počet úloh je vykonán nezřetězeným procesorem za T_1 časových intervalů

$$T_1 = n \cdot k$$

Na základě dob realizace "n" úloh ve zřetěženém a nezřetěženém procesoru je možno definovat koeficient zrychlení výpočtu (speedup) S_k .

$$S_k = T1/T_k = (n.k)/(k+(n-1))$$

Maximální zrychlení pro velmi velká "n" se blíží "k". To znamená, že lineárně zřetěženým procesorem je možno dosáhnout zrychlení rovnajícího se až počtu stupňů tohoto procesoru. Tohoto maximálního koeficientu zrychlení nebývá v praxi nikdy dosaženo (příčina: nespojitost vstupního toku, přerušení, větvení programu atd.)

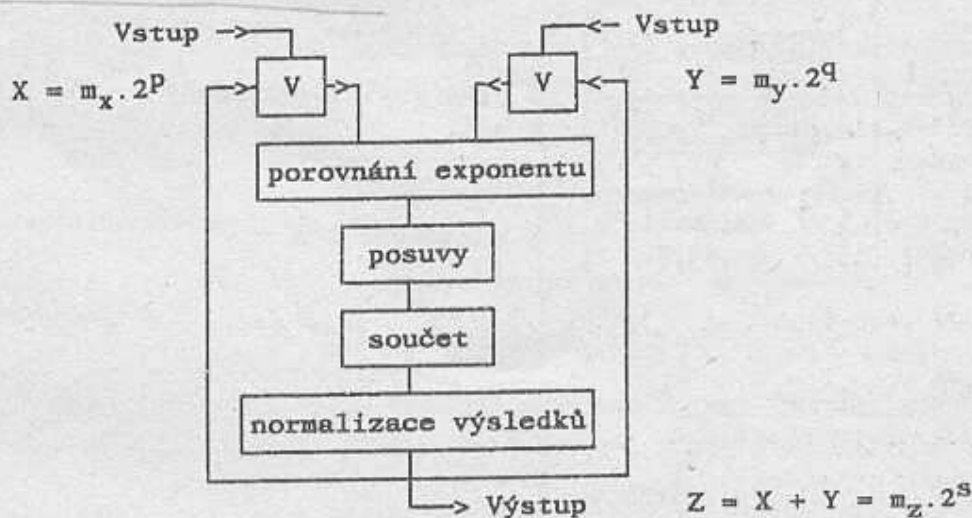
4.4.1 Typy zřetězení

Zřetězení lze rozdělit na:

- zřetězení aritmetické
- zřetězení instrukcí
- zřetězení procesů

Zřetězení aritmetické je dosaženo v aritmeticko-logické jednotce (Arithmetic Logic Unit - ALU).

Příklad zřetěženého sčítání v pohyblivé řádové čárce je uveden na obr. č. 4.5. Na tomto obrázku je čtyřstupňová sčítačka v pohyblivé řádové čárce.



Obr. č. 4.5 Schéma zřetěženého součtu v pohyblivé řádové čárce

Vstupem jsou dvě normovaná čísla "X" a "Y" v pohyblivé řádové čárce.

$$X = m_x \cdot 2^p \text{ a } Y = m_y \cdot 2^q$$

" m_x " a " m_y " jsou normované mantisy a " p ", " q " jsou jejich exponenty.

Úkolem sčítačky je realizace následujícího výpočtu.

$$Z = X + Y = c \cdot 2^t = m_z \cdot 2^s$$

Jednotlivé stupně sčítačky vykonávají následující činnosti:

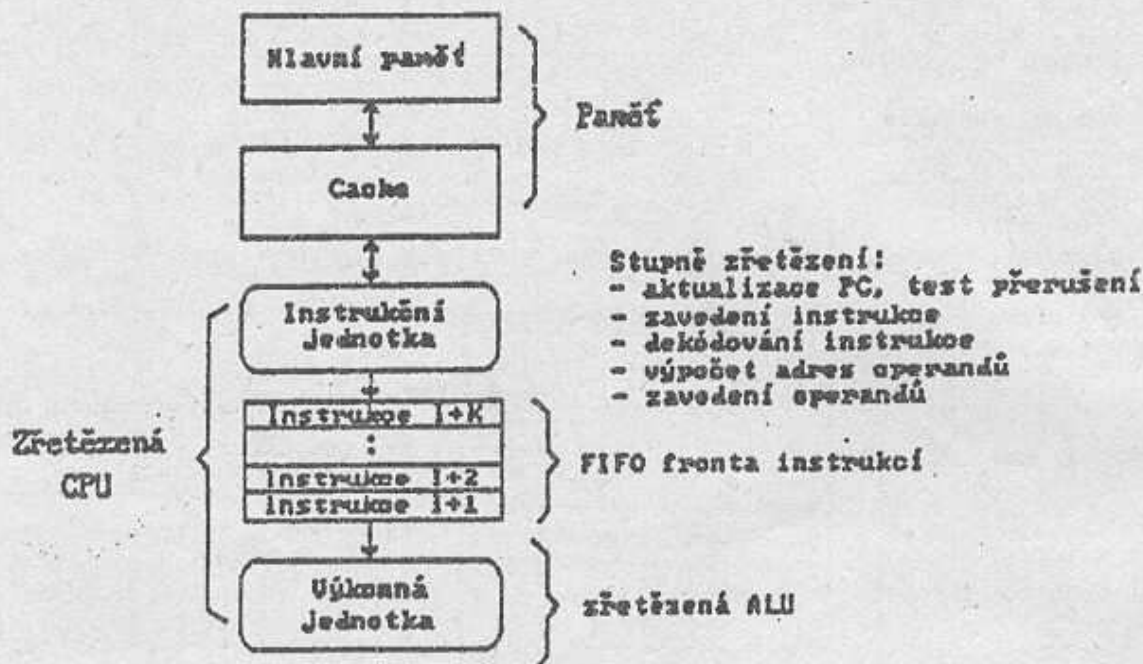
1. porovnání exponentů " p " a " q ", výběr většího z nich $r = \max(p, q)$ a výpočet jejich rozdílu $t = \text{abs}(p - q)$
2. posuv mantisy toho sčítance, jehož exponent je menší (zarovnání před sčítáním mantis) o " t " řádu doprava
3. součet zarovnaných mantis, dosažení mezivýsledku " c ", kde $0 \leq c < 1$

4. posouvání výsledku "c" doleva, dokud není dosaženo, že první cifrou čísla "c" bude "1". Počet posuvů necht' je "u". Výsledek posuvu je "m_z"

Výsledek součtu je potom

$$Z = m \cdot 2^s, \text{ kde } s = r - u$$

Zřetězení instrukcí je dosaženo v zřetězené instrukční jednotce (Instruction Unit), jejíž činnost je rozdělena např. na načítání instrukce z paměti, dekódování instrukce, výběr operandů instrukce z paměti a na vlastní vykonání instrukce [3].



Obr. č. 4.6 Zřetězená struktura CPU

Centrální jednotka procesoru (Central Processing Unit-CPU) moderních číslicových počítačů může být všeobecně dekomponována na tři části:

- instrukční (řídící) jednotka (Instruction Unit)
- fronta připravených instrukcí
- výkonná (aritmicko-logická) jednotka (Execution Unit)

Tyto tři části mohou pracovat zřetězeně, přičemž je možné i vnitřní zřetězení každé části, jak ilustruje obr. č. 4.6. Programy a data jsou umístěny v hlavní paměti.

Často je použito vyrovnávací paměti s cílem překlenout rozdíl v době přístupu k datům v hlavní paměti a v registrech procesoru.

Řetězená Instrukční jednotka (add.a) - Instruction Unit) může být složena z následujících modulů:

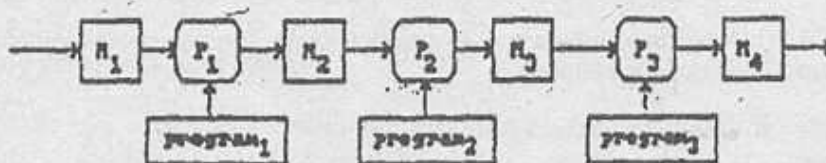
- modul načítání instrukcí z paměti
- modul dekódování instrukce
- modul výpočtu adres operandů
- modul načítání operandů z paměti

Fronta připravených instrukcí (add. b)) tj. instrukcí, které jsou dekodovány a připraveny k realizaci výkonnou jednotkou, je typu **FIFO** (First In, First Out - první dovnitř, první ven) a obsahuje dekodované instrukce a jejich operandy.

Výkonná jednotka (add. c) - Arithmetic Logic Unit, Execution Unit) se může skládat z několika elementárních aritmetických jednotek, funkčně orientovaných na konkrétní aritmetickou nebo logickou operaci, nebo skupinu takovýchto operací.

V okamžiku, kdy výkonná jednotka realizuje I-tou instrukci, jsou ve frontě připravených instrukcí instrukce I+1, I+2, ..., I+K a instrukční jednotka načítá z paměti instrukce I+K+1, viz. obr. č. 4.6.

Zřetězení procesů odpovídá zřetězenému zpracování téhož datového toku kaskádou procesorů, z nichž každý opakovaně řeší specifický program nad různými daty.



Obr. č. 4.7 Zřetězení procesů

Data postupují z paměťového bloku M1, procházejí procesorem P1 a jsou v něm zpracována programem 1. Výsledky jsou uloženy do paměťového bloku M2, který je současně dostupný druhému procesoru P2. Procesor P2 potom zpracuje tato data svým programem a předá je pomocí dalšího paměťového bloku M3 třetímu procesoru P3 atd. (viz obr. č. 4.7,[3]).

4.4.2 Klasifikace zřetězených systémů

Na základě konfigurace řetězce a řízení zřetězeného výpočtu jsou klasifikovány následující typy zřetězených systémů:

- monofunkční, resp. multifunkční řetězec
- statický, resp. dynamický řetězec
- skalární, resp. vektorový řetězec

Monofunkčním řetězcem je nazýván takový, který vykonává jen jednu funkci, jako např. součet v pohyblivé řádové čáře (viz. obr. 4.5).

Multifunkční řetězec je schopen vykonávat různé funkce, a to buď v různém čase nebo v témže časovém okamžiku, prostřednictvím propojení různé podmnožiny modulů zařazených do řetězce.

Statický řetězec předpokládá pouze jednu funkční konfiguraci v daném časovém okamžiku. Statický řetězec může být buď monofunkční nebo multifunkční. Funkce řetězce však musí být nastavena ještě před zahájením výpočtu.

Dynamický řetězec umožňuje současnou realizaci několika funkčních konfigurací. Operandy nepřetržitě vstupují do řetězce, přičemž procházejí pouze těmi jeho moduly, které pro svoje zpracování vyžadují.

Skalární řetězec je schopen zpracovávat sekvenci skalárních operandů.

Vektorový řetězec je speciálně navržen pro zpracování vektorových instrukcí nad vektorovými operandy.

4.5 Zřetězené instrukce procesoru

V této kapitole jsou na příkladu implementace zřetězené instrukční jednotky v procesoru RISC ilustrovány aplikace výše uvedených principů zřetězení.

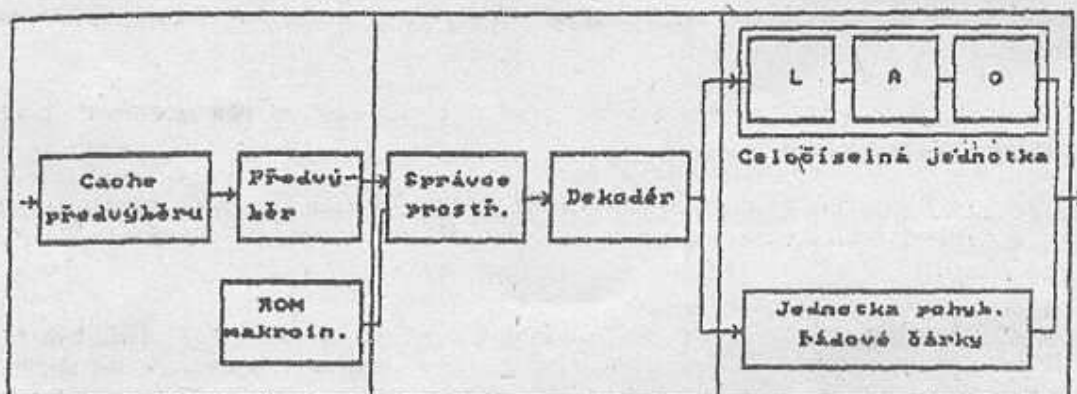
4.5.1 Realizace zřetězení instrukcí procesoru

Zřetězené instrukční jednotky používá řada současných mikroprocesorů. Jednou z prvních implementací byl např. mikroprocesor Intel 8086, který je rozdělen na jednotky BIU (Bus Interface Unit) a EU (Execution Unit). Tyto jednotky tvoří dvoustupňové zřetězení instrukcí. Většina 32bitových procesorů používá techniku zřetězené instrukční jednotky. Jako příklad implementace zřetězení instrukční jednotky v procesorech s redukováným instrukčním souborem je uvedena architektura procesoru Clipper. Tento procesor navíc umocňuje výhody implementace jednoduchého třístupňového zřetězení instrukce zavedením **paralelní činnosti** v jednotlivých modulech řetězce.

Na obr. č. 4.8 [3] je uvedena architektura procesoru Clipper. Procesor je dekomponován do **třístupňového** řetězce složeného z následujících modulů:

- modul zavedení instrukcí (Instruction Fetch)
- modul dekódování instrukcí
- modul výkonné jednotky (Execution Unit)

Každý modul je schopen nezávislé činnosti. Protože Clipper realizuje všechny arit.-logické instrukce pouze nad operandy v registrech, není v procesoru nutný modul výpočtu adres operandů a modul jejich zavedení.



Obr. č. 4.8 Architektura RISC procesoru Clipper

Modul zavedení instrukcí obsahuje dva bloky pro správu instrukčního toku, které pracují také zřetězeně. Clipper se skládá z trojice čipů dedikovaných pro speciální účely. Modul zavedení instrukcí napomáhá CPU zabezpečit nepřetržitý tok instrukcí a zvyšovat úspěšnost nalezení potřebných dat ve vyrovnávací paměti bez potřeby přístupu do hlavní paměti.

Modul dekódování obsahuje dva bloky - správce prostředků a dekodér instrukcí. Správce prostředků eviduje činnost všech komponent procesoru a způsob jejich využití v jednotlivých rozpracovaných instrukcích a tabulku aktuálního přidělení prostředku dané instrukci.

Modul výkonné jednotky procesoru Clipper obsahuje dvě nezávislé jednotky - jednotku pohyblivé řádové čárky a celočíselnou aritmeticko-logickou jednotku.

Celočíselná jednotka vykonává všechny arit.-logické operace a výpočty efektivních adres. Jednotka pohyblivé řádové čárky realizuje výpočty v pohyblivé řádové čárce.

Vlastní celočíselná jednotka je dále rozdělena na tři moduly označené L, A a O. Každý modul může pracovat nezávisle a paralelně s ostatními moduly celočíselné jednotky. **První modul** načítá aktuální operand

(registru) do speciálních registrů označených L. Druhý modul může vykonávat arit.-log. operace, operace posuvu nad operandy v registrech L, nebo nad mezivýsledky předcházejících instrukcí. Poslední modul O přenáší výsledky operací modulu A do:

- jednotky pohyblivé řádové čárky
- všeobecných registrů procesoru
- výstupů z procesoru

Implementace zpětné vazby mezivýsledků z modulu O do modulu L umožňuje použít aktuální data v následující instrukci a vyvarovat se problémů s přerušením plynulosti realizace zřetěženého výpočtu v důsledku tzv. datového konfliktu.

Jednotka pohyblivé řádové čárky vykonává operace v pohyblivé řádové čárce s jednoduchou a dvojnásobnou přesností. Může pracovat souběžně s celočíselnou jednotkou.

4.4.2 Problémy zřetězení instrukcí

Nejvyšší efektivity zřetězení instrukcí je dosaženo, když instrukce spojí proudy do procesoru a jejich tok není narušován.

V reálných programech však dochází k narušení spojitosti toku instrukcí a v důsledku toho k narušení zřetěženého zpracování. Narušení zřetěženého zpracování znamená, že se poruší plynulá návaznost činnosti jednotlivých modulů řetězce, a to jak z hlediska zpracování jedné instrukce, tak z hlediska zpracování celkového toku instrukcí zřetěženým procesorem. Při narušení průběhu zřetěženého zpracování je nutné některé operace zpozdit nebo je třeba rozbíhat zřetěžené zpracování opět od jeho začátku. Každé narušení vede ke ztrátě výkonu procesoru.

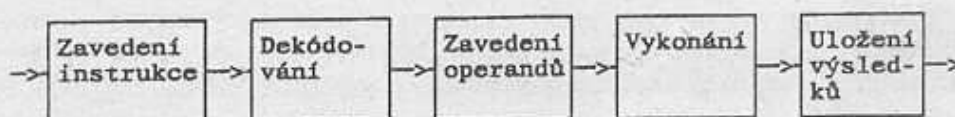
Existují dva druhy konfliktů determinujících narušení průběhu zřetěženého zpracování. Jedná se o datový konflikt a skokový konflikt.

Datový konflikt vzniká v případě, kdy několik instrukcí v řetězci za sebou pracuje se stejným operandem. Např. následující instrukce už vybírá data z registru, aniž by byla ještě dokončena jejich modifikace instrukcí předcházející.

Skokový konflikt může vzniknout na základě použití takové instrukce, která mění obsah čítače instrukcí. K těmto instrukcím patří instrukce podmíněného větvení, v případě naplnění podmínky instrukce nepodmíněného větvení, instrukce volání procedur, volání služeb operačního systému atd. Do zvláštní skupiny narušení spojitosti toku instrukcí je nutno pro specifickou reakce procesoru zařadit všechny druhy přerušení, tj. reakce na vnější podnět, neznámý kód instrukce atd.

Vliv instrukcí větvení a skokového konfliktu na činnost zřetěženého procesoru je ilustrován na příkladu CPU složené z pěti modulů:

- modulu zavedení instrukce
- modulu dekódování
- modulu zavedení operandů
- výkonného modulu
- modulu uložení výsledků

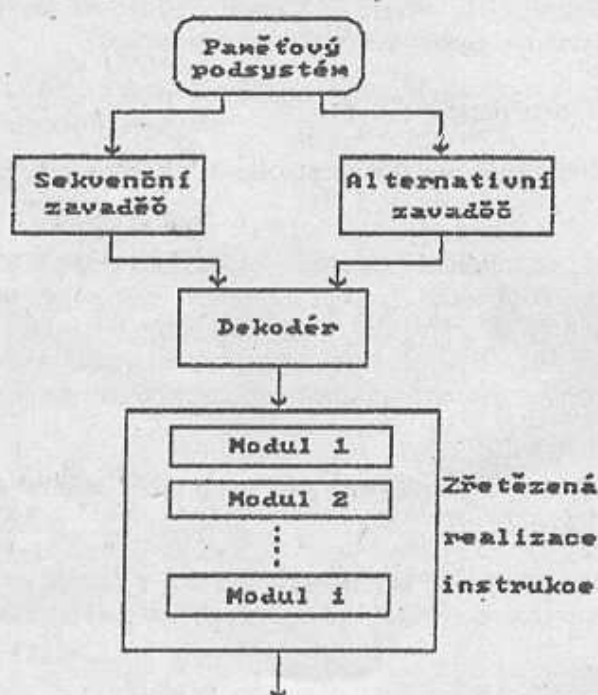


Obr. č. 4.9 Vliv instrukce větvení na spojitost zřetězení instrukcí

V případě, že tok instrukcí je spojitý a neobsahuje žádné instrukce větvení, vykonává zřetězená jednotka po svém naplnění jednu instrukci za jeden časový interval "T". Instrukce větvení je detekována modulem dekódování. Jedná-li se o instrukci podmíněného větvení, je nutno dále zavést operandy podmínky a vypočítat podmínku větvení ve výkonném modulu. Teprve potom je možno rozhodnout o způsobu pokračování.

Každé narušení zřetězeného zpracování se projeví na výkonu počítačů se zřetězeným procesorem. Proto je používána řada metod na potlačování těchto narušení. Tyto metody jsou založeny na spolupráci technických a programových prostředků. Technické prostředky se však používají minimálně, protože ztěžují návrh obvodu a vyžadují dostatečnou plochu čipu atd., což je v rozporu s filosofií návrhu procesoru RISC.

Zabezpečení spojitosti zřetězení instrukcí prostřednictvím **technických prostředků** je uvedeno na obr. č. 4.10 [3].



Obr. č. 4.10 Řešení skokového konfliktu

Modul zavedení instrukcí je nahrazen dvojicí modulů zavádění instrukcí, sekvenčním zavaděčem a alternativním zavaděčem. Každý z nich má nezávislý přístup do operační paměti. Sekvenční zavaděč načítá instrukce v lineární posloupnosti. Je-li dekódována instrukce větvení, je zahájena souběžná činnost alternativního zavaděče. Ten načítá instrukce od adresy dané instrukcí větvení. Podle výsledku dekódování a výsledku podmínky větvení je pro další zpracování vybrána odpovídající fronta instrukcí, přičemž druhá fronta je potlačena.

Zabezpečení spojitosti zřetězení instrukcí **programovými prostředky** vedlo k zavedení techniky tzv. zpožděného skoku. Je-li ve zřetězeném procesoru detekována instrukce větvení, nedojde k její okamžité realizaci. Její vykonání je "zpožděno" o dobu, která je nutná k dokončení instrukcí rozpracovaných za touto instrukcí. Tyto instrukce nesmějí změnit stav procesoru oproti stavu, který by nastal bez použití techniky zpožděného skoku. Potom se pokračuje instrukcí na cílové adrese skoku nebo se skok potlačí a následuje další instrukce v sekvenčním pořadí.

Datový konflikt může vzniknout tehdy, jestliže různé instrukce, které jsou vykonávány v řetězci, pracují nad stejnými datovými objekty, tedy používají operandy z téhož registru nebo téže buňky operační paměti. Na základě způsobu aktualizace a přístupu ke sdíleným datovým objektům jsou rozlišovány následující druhy datových konfliktů:

- zápis po čtení (Write After Read - WAR)
- čtení po zápisu (Read After Write - RAW)

- zápis po zápisu (Write After Write- WAW)

Rozpoznání možných datových konfliktů umožňuje implementovat prostředky, které dovoří detekovat a odstranit skutečné datové konflikty v době běhu programu. Detekce datového konfliktu je obvykle realizována v modulu zavedení instrukce do paměti prostřednictvím testování adres operandů jednotlivých instrukcí. V případě pozitivní detekce datového konfliktu je nutno řešit dočasné "uzamknutí" datového objektu.

4.5 Vektorové výpočty

V této kapitole jsou vysvětleny některé základní pojmy vektorového zpracování dat. Doposud byly vektorové výpočty výhradně aplikovány v superpočítačích. V současnosti je tento způsob výpočtu implementován i v některých procesorech RISC.

4.5.1 Charakteristika vektorového zpracování

Při vektorovém zpracování používá procesor speciální instrukční soubor, který obsahuje instrukce pracující nad vektory. To znamená, že operandy těchto instrukcí jsou vektory. Vektorový operand je definován jako uspořádaná množina "n" prvků, kde "n" je velikost této množiny. Každý prvek vektoru je skalární veličinou, která může být číslem v pohyblivé řádové čárce, celým číslem, logickou hodnotou, znakem či obrazovým prvkem (pixel). Vektorová instrukce realizuje operace nad vektorovými operandy, skalární instrukce nad skalárními operandy. Vektorové instrukce je možno všeobecně klasifikovat do čtyř skupin:

- f1: V - V
- f2: V - S
- f3: V x V - V
- f4: V x S - V

"V" označuje vektorový operand a "S" skalární operand. Vektorové instrukce f1 a f2 jsou unární funkce zobrazující v případě f1 vektorový operand na vektorový operand a v případě f2 vektorový operand na skalární operand. Instrukce f3 transformují dvojici vektorových operandů na vektorový operand a instrukce f4 transformuje vektorový a skalární operand na operand vektorový.

Příklady viz tabulku č. 4.2.

Tabulka č. 4.2 Příklad vektorových instrukcí

Typ	Mnemonika	Popis (I=1 až N)
f1	VSQR	Norma vektoru: $B(I):=A(I)$
	VSIN	Sinus prvků vektoru: $B(I):=\sin(A(I))$
f2	VMAX	Max. prvek vektoru : $S:=\max(A(I))$
f3	VADD	Součet vektorů : $C(I):=A(I)+B(I)$
	VMPY	Skalární součin: $C(I):=A(I)*B(I)$
	VAND	Logický součin : $C(I):=A(I) \text{ and } B(I)$
f4	SADD	Součet vektoru a skaláru: $B(I):=S+A(I)$
	SDIV	Dělení vektoru skalárem : $B(I):=A(I)/S$

Tyto a další vektorové instrukce jsou implementovány ve vektorových procesorech, přičemž při jejich realizaci je často používáno principů zřetězení. Vektorové instrukce vykonávají opakovaně tytéž operace nad různými skupinami datových objektů. Formát vektorové instrukce je uveden na obr. č. 4.11.

Operační kód	Bázová adresa	Inkrement adresy	Relativní adresa	Rozměr vektoru
--------------	---------------	------------------	------------------	----------------

Obr. č. 4.11 Formát vektorové instrukce

Jednou z hlavních výhod vektorových instrukcí oproti instrukcím skalárním je eliminace časových ztrát vznikajících při skalárním zpracování cyklů. Časové ztráty skalárního cyklu vznikají programovou organizací cyklu výpočtu. Popis formátu vektorové instrukce:

1. **Operační kód** - specifikuje druh operace, která zabezpečí vykonání odpovídající vektorové operace
2. **Bázová adresa** - je nutná pro operace s datovými objekty v operační paměti, určuje adresu vstupních vektorových nebo skalárních operandů a výstupních operandů
3. **Inkrement adresy** - definuje velikost jednoho prvku vektoru
4. **Relativní adresa** - na základě této adresy a bázové adresy dochází k výpočtu efektivní adresy vektorového nebo skalárního operandu
5. **Rozměr vektoru** - determinuje ukončení vektorové instrukce a navíc lze použít tzv. vektor masky, který určí, se kterými prvky vektorového operandu bude daná vektorová instrukce provedena

Příklad:

```
for i:= 1 to N do
begin
  A(I) := B(I) + C(I);
  B(I) := 2 * A(I);
end
```

Vektorový procesor může vykonat uvedený program jako posloupnost následujících dvou vektorových instrukcí:

```
A(1:N) = B(1:N) + C(1:N)
B(1:N) = 2 * A(1:N)
```

A(1:N) představuje N prvků vektoru A(1), A(2), ..., A(N)

S cílem zvýšit efektivnost vektorových výpočtů jsou při návrhu vektorových procesorů používány následující přístupy:

- **rozšíření vektorového instrukčního souboru** - rozsáhlejší instrukční soubor zvýší možnost zpracování
- **vhodné kombinování vektorových a skalárních instrukcí**. Při implementaci zřetězené výkonné jednotky (Pipelined Execution Unit) je vhodnější **seskupit skalární instrukce do jedné skupiny a vykonat je najednou, než prokládat realizace vektorových instrukcí realizacemi skalárních instrukcí**
- **volba vhodného algoritmu**. Ve většině případů bude rychlý a efektivní sériový algoritmus
- **použití vektorizujícího kompilátoru**. Pro detekce možných paralelismů je nutno vyvinout speciální překladač. Vektorizující kompilátor by měl regenerovat paralelismus úlohy, který se ztrácí realizací algoritmu sériovým programovacím jazykem. Je vhodné použít jazyk vysoké úrovně

4.5.2 Architektura vektorového procesoru

Na základě umístění operandů vektorového procesoru je možno rozdělit vektorové procesory do dvou skupin architektur:

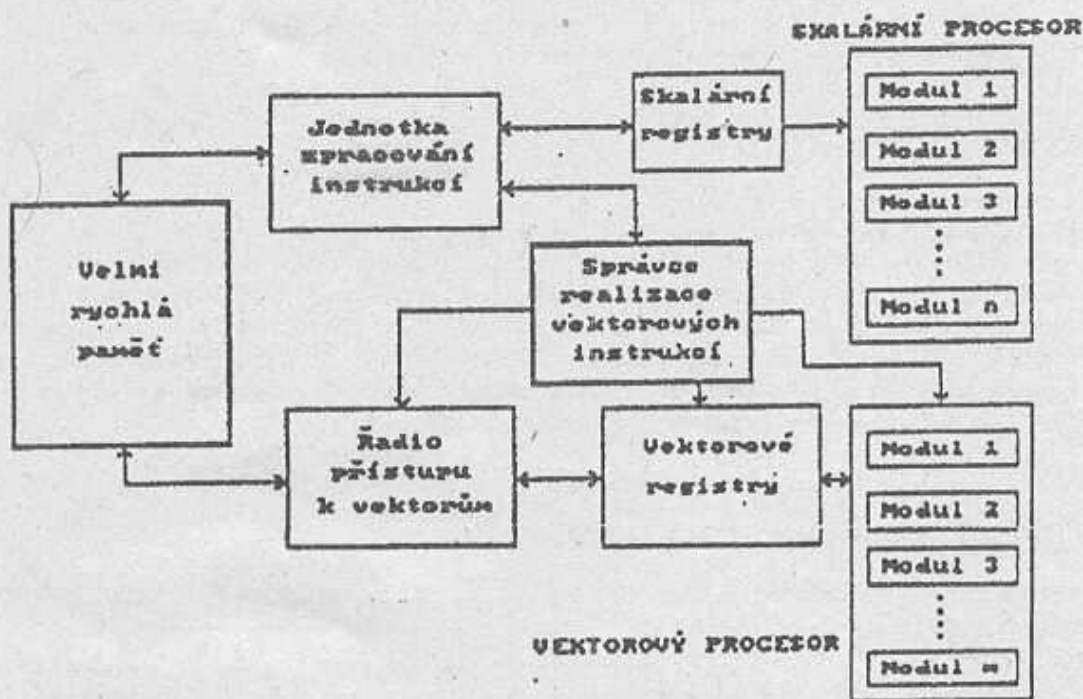
- a) paměťová architektura
- b) registrová architektura

V **paměťové architektuře** vektorového procesoru jsou všechny vektorové operace vykonávány nad vektory v operační paměti. Vstupní vektory, mezivýsledky i výsledky vektorových operací jsou v operační paměti.

V **registrové architektuře** vektorového procesoru jsou vektorové operace vykonávány nad daty v registrech procesoru. Vstupní vektory, skalární veličiny, mezi-výsledky a výsledky jsou ukládány do registrů procesoru. Pro tuto skupinu je charakteristická velká množina vektorových a skalárních registrů procesorů.

Na obr. č. 4.12 [3] je uvedeno blokové schéma vektorového počítače, který používá násobných zřetězených jednotek.

Uvedená architektura je zevšeobecněním současných vektorových počítačů.



Obr. č. 4.12 Architektura vektorového počítače

Instrukce vektorového počítače, jehož schéma je na obr. 4.12, mohou být vektorové i skalární. Jednotka zpracování instrukcí zavádí a dekóduje jak vektorové, tak i skalární instrukce. Všechny skalární instrukce jsou zasilány do skalárního procesoru. Skalární procesor může být dále dekomponován na několik zřetězených jednotek. Jakmile je jednotkou zpracování instrukcí dekódována vektorová instrukce, tj. instrukce nad vektory, veškeré její další zpracování je zabezpečováno správcem realizace vektorových instrukcí. Ten vykonává následující operace:

- dekódování vektorové instrukce
- výpočet efektivní adresy vektorových operandů
- inicializace řadiče přístupu k vektorům
- inicializace a nastavení vektorového procesoru
- monitorování realizace vektorové instrukce

Řadič přístupu k vektorům zavádí vektorové operandy z operační paměti pro další zpracování. Vektorové registry umožňují překlenout rozdíl v rychlosti přístupu k vektorovým operandům umístěným v operační paměti.

4.6 Vyrovnávací paměti

Vyrovnávací paměť (Cache Memory) je velmi rychlá paměť, která je zpravidla umístěna mezi procesorem a hlavní pamětí výpočetního systému. Ve vyrovnávací paměti je uložena ta část hlavní paměti, která je právě procesorem používána. Vyrovnávací paměť může být také umístěna mezi hlavní paměť a velkokapacitní

vnější paměť. Ačkoliv je paměť **cache** zhruba **pětkrát až desetkrát** rychlejší než paměť hlavní. Paměť **cache** má omezenou kapacitu. Jsou v ní uloženy kopie některých dat z hlavní paměti výpočetního systému. Nacházejí-li se požadované údaje, data nebo instrukce v paměti **cache**, jsou přečteny z této rychlé paměti a není uskutečněn přístup do relativně pomalé hlavní paměti.

Použití paměti **cache** je v architekturách RISC velmi rozšířeno. Procesory RISC v řadě implementací používají koprocesor, který přebírá veškeré úlohy spojené s přípravou a aktualizací dat ve vyrovnávací paměti (**cache**).

4.6.1 Charakteristika vyrovnávacích pamětí

Aplikace vyrovnávací paměti jsou známy již z dob návrhu sálových počítačů (Mainframe Computers). Umožňovaly dosáhnout velkého výkonu při současné implementaci pomalejších paměťových podsystémů.

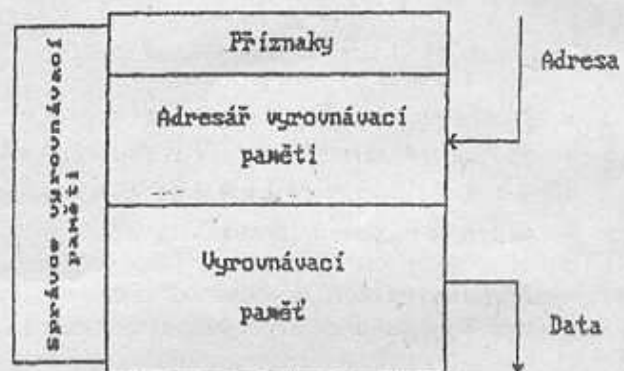
Princip aplikace vyrovnávací paměti spočívá ve využití objemově malé, co do doby přístupu však velmi rychlé, a tedy i drahé vyrovnávací paměti, která je umístěna mezi procesorem (Central Processing Unit) a hlavní pamětí počítače. Rychlý procesor je nucen při čtení pomalé paměti přecházet do tzv. čekacích cyklů (Wait State), kdy čeká na vydání obsahu paměti. Často používaná data a instrukce jsou proto zavedeny do vyrovnávací paměti, ze které mohou být velmi rychle zpřístupněny. K hlavní paměti je uskutečněn přístup pouze při požadavku nové hodnoty dat či instrukcí, které nejsou ve vyrovnávací paměti. Přístup do hlavní paměti je přitom vykonán na procesoru nezávislými technickými prostředky a procesor není přenosem nikterak zatížen.

Vyrovnávací paměť obsahuje následující moduly (viz. obr. č. 4.13, [3]).

- správce vyrovnávací paměti
- adresář vyrovnávací paměti
- vlastní vyrovnávací paměť (typu RAM)

Paměť **cache** je rozdělena do segmentů o stejné velikosti dvou a více bytů. Tyto segmenty jsou obvykle nazývány **bloky** (Block Frames). Vyrovnávací paměť obsahuje dvojice "**adresa bloku, data bloku**". Velikost vyrovnávací paměti je celistvým násobkem velikosti bloku.

Obvykle i hlavní paměť je rozdělena do segmentů, které jsou nazývány **třídy** a jejich velikost je celistvým násobkem velikosti bloků. Třídou je označována skupina bloků v hlavní paměti, z nichž každý může být vybrán pro zápis do jednoho bloku paměti **cache**.



Obr. č. 4.13 Schéma vyrovnávací paměti

Adresa operandu postupující z procesoru je porovnávána s adresami v adresáři paměti cache s cílem zjistit, nachází-li se obsah dané adresy v paměti cache nebo ne. Hledání adresy v adresáři je většinou vykonáno asociativně, takže adresář je realizován jako **asociativní paměť**. V adresáři je shodný počet adres s počtem položek v paměti cache. V paměti cache jsou vždy dvojice "**adresa bloku, data bloku**". Kromě informace o adrese bloku, která je uložena v adresáři vyrovnávací paměti, a jeho obsahu, který je uložen ve vlastní vyrovnávací paměti, může cache paměť obsahovat informaci o aktuálnosti uložených dat v tzv. **poli příznaků** (viz. obr. 4.12). Jedná se např. o příznak **platnosti dat, správnosti dat, blokování přístupu k datům** atd.

Všeobecně platí, že návrh a implementace vyrovnávací paměti vyžaduje náročnější přístupy než návrh standardní paměti.

4.7.2 Algoritmy správy dat

Obsah vyrovnávací paměti je kopií části paměti hlavní. Je nutno rozhodnout, na která místa ve vyrovnávací paměti mají být uložena data z hlavní paměti (problém mapování), které dvojice "adresa bloku, data bloku" z hlavní paměti mají být v paměti vyrovnávací (problém aktualizace dat), a zabezpečit shodu obsahu identických bloků v paměti cache a v hlavní paměti (problém konzistence dat). Universální strategie přenosu položek mezi hlavní a vyrovnávací pamětí neexistuje a dosud známé přístupy jsou i nadále předmětem dalších analýz a dále se vyvíjejí.

Při spuštění programu je vyrovnávací paměť prázdná. Určitá položka hlavní paměti se přesune do paměti vyrovnávací obvykle teprve tehdy, žádá-li procesor obsah některé z adres vyrovnávací paměti (do té doby tedy byly všechny přístupy směřovány do hlavní paměti). Ale již při částečném naplnění vyrovnávací paměti dojde k uplatnění lokality přístupů programů do paměti, a tedy k využití efektu vyrovnávací paměti. Lokality přístupu do paměti znamená, že byla-li určitá adresa volná, pak tato adresa sama, nebo některá z adres v jejím nejbližším okolí, bude po krátké době volána znovu.

V další části budou rozebrány následující tři klíčové problémy:

- způsoby přiřazení bloků vyrovnávací paměti blokům hlavní paměti (problém mapování)
- algoritmy uvolňování dat z vyrovnávací paměti (problém aktualizace dat)
- algoritmy zabezpečení shody obsahu vyrovnávací paměti a hlavní paměti (problém konzistence dat)

1. Organizace paměti cache

Organizace paměti cache je dána zvoleným způsobem mapování (Placement Policy). Mapování určuje pravidla, jakým způsobem jsou přiřazovány bloky v operační paměti blokům v paměti cache.

Máme čtyři možné přístupy k mapování:

- přímé mapování (Direct Mapping)
- plně asociativní mapování (Fully Associative Mapping)
- skupinově asociativní mapování (Set Associative Mapping)
- sektorové mapování (Sector Mapping)

Přímé mapování: Je nejjednodušším mapováním. V tomto případě jsou určité bloky hlavní paměti přiřazeny právě jednomu bloku paměti cache. Paměť cache má velikost jedné paměťové třídy.

Výhodou přímého mapování je současný přístup k datům i adresám tříd. Umožňuje také implementovat jednoduché algoritmy uvolňování dat a údržby jejich konzistence. Ze všech bloků hlavní paměti, mapovaných do jednoho bloku paměti cache, pouze jeden může být v daném časovém okamžiku v paměti cache.

Plně asociativní mapování: Nejlepší, ale i nejdražší organizace cache paměti. Libovolný blok hlavní paměti může být umístěn v libovolném bloku paměti cache. Nastane-li požadavek na zpřístupnění dat z určitého bloku, jsou všechny adresy v adresáři paměti cache porovnávány s požadovanou adresou. Porovnání všech adres je souběžné, asociativní a vyžaduje implementaci speciálních technických prostředků.

Skupinově asociativní mapování: Kompromis mezi jednoduchým přímým mapováním a drahým plně asociativním mapováním. Při implementaci skupinově asociativního mapování je paměť cache rozdělena do S skupin

$$E = M / S$$

paměťovými bloky v jedné skupině, kde:

- M - celkový počet bloků, adresa skupiny je vyhledávána přímo, blok v rámci jedné skupiny asociativně
- S - počet skupin, je obvykle mocninou 2
- E - počet bloků v jedné skupině

Sektorové mapování: Hlavní paměť je rozdělena do několika sektorů, z nichž každý se skládá z několika bloků. Podobně je rozdělena paměť cache do několika sektorů o několika blocích. Nastane-li požadavek na čtení bloku, který není v paměti cache, dojde k jeho zavedení z hlavní paměti a uložení do paměti cache s určitým omezením. Každému bloku paměti cache je přiřazen jeden příznakový bit označující platnost údajů v tomto bloku.

2. Uvolňování dat z paměti cache

Jakmile se vyrovnávací paměť naplní, je nutno rozhodnout o tom, který blok vyrovnávací paměti má uvolnit místo pro nový blok z hlavní paměti. Nejrozšířenějším algoritmem je algoritmus LRU (Least Recently Used = nejdéle nepoužívaný). Vyřazuje se vždy ten blok, který byl nejdéle procesorem nepoužit.

3. Konzistence dat v paměti cache

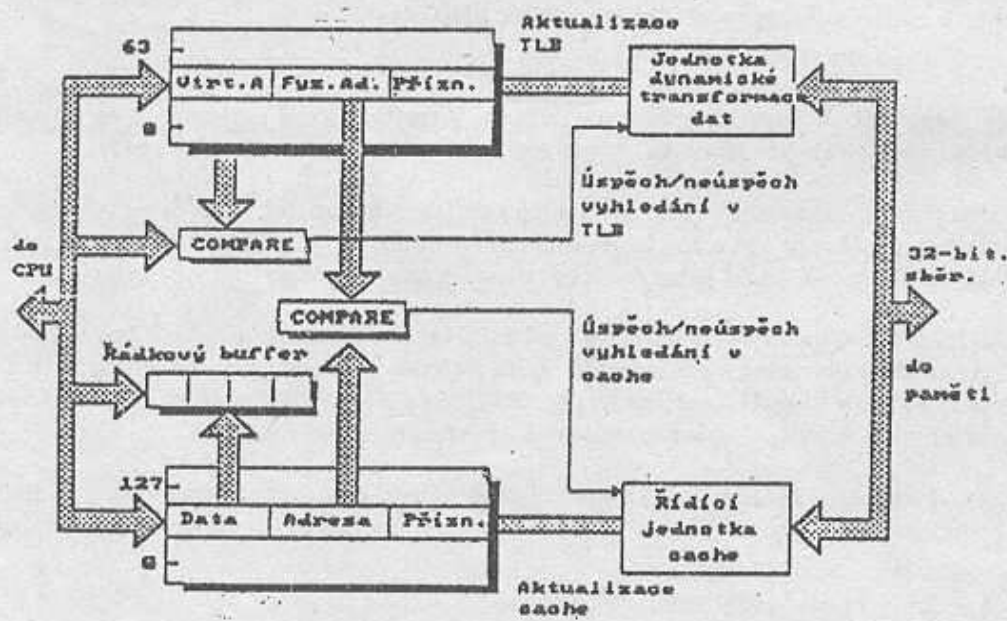
Dalším problémem, který je nutno při aplikaci vyrovnávací paměti řešit, je problém konzistence dat. Vzhledem k tomu, že se některé reálné adresy vyskytují současně v hlavní paměti i v jedné nebo několika vyrovnávacích pamětech, vzniká totiž nebezpečí, že na každou z nich budou zadána jiná data. K neshodám obsahů dochází ze dvou důvodů:

- obsah se změní jen v hlavní paměti
- obsah se změní jen ve vyrovnávací paměti

4.7.3 Příklad implementace paměti cache

V této kapitole jsou ilustrovány vlastnosti vyrovnávací paměti na její konkrétní implementaci v RISC procesoru Clipper, viz. obr. č. 4.14. Procesor Clipper se skládá ze tří prvků - procesoru a dvou vyrovnávacích pamětí. Jedna paměť je určena pro instrukce, druhá pro data. Hierarchie paměti cache procesoru Clipper překonává časový schodek mezi cyklem procesoru (30 ns) a vybavovací dobou vnější diskové paměti (30 ms).

V další části se budeme zabývat pouze cache pamětí umístěnou mezi hlavní pamětí a procesorem [3].



Obr. č. 4.14 Vyrovnávací paměť procesoru Clipper

Paměť cache je rozdělena na dvě části:

- hlavní vyrovnávací paměť (velikost 4 kB)
- velmi rychlá paměť cache nazývána řádkový buffer (16 bytů)

Vyrovnávací paměť se skládá z 256 bloků o velikosti 16 bytů. Při čtení dat je testována jejich přítomnost v bloku hlavní vyrovnávací paměti nebo v řádkovém bufferu. Nachází-li se data v hlavní vyrovnávací paměti, jsou přenesena do procesoru a zároveň je odpovídající blok s požadovanými daty přesunut do řádkového bufferu. Je-li blok již v bufferu, není tento přesun uskutečněn a jsou použita data z tohoto bufferu. V případě nepřítomnosti dat ve vyrovnávací paměti je nutné realizovat čtení z hlavní paměti. Je vždy přenášen jeden blok o velikosti 16 bytů.

Data z řádkového bufferu je možno zpřístupnit ve 30 ns, tj. v jednom cyklu procesoru Clipper. V případě, že data nejsou v řádkovém bufferu, jsou nutné další dva cykly procesoru pro jejich zpřístupnění, tedy 90 ns. Přenos mezi procesorem a vyrovnávací pamětí vyžaduje 15 ns.

Na začátku je paměť cache prázdná. Při čtení dat je uskutečněn přístup do hlavní paměti a 16 bytový blok je uložen do prvního bloku odpovídající skupiny v paměti cache. V případě čtení dat z jiného bloku téže stránky je tento blok zaveden z hlavní paměti a uložen ve druhém bloku skupiny paměti cache. Jsou-li oba bloky skupiny plné, dojde na základě algoritmu LRU k uvolnění jednoho bloku.

Pro vyrovnávací paměť instrukcí je navíc umožněn režim předvýběru instrukcí. V době, kdy procesor vykonává přístup k jednomu bloku, je realizováno načtení následujícího bloku instrukcí.

Problém konzistence dat při zápisu je vyrovnávací paměti procesoru Clipper řešen dvěma způsoby. Jednak je podporován souběžný zápis jak do paměti cache, tak i do hlavní paměti, jednak je podporován zápis do hlavní paměti až při uvolňování bloku z paměti cache.

5. MIKROPROCESORY FIRMY INTEL

První integrovaný obvod, který můžeme nazvat mikroprocesor, navrhl pracovník firmy INTEL E. Hoff při práci na vývoji obvodů pro stolní kalkulačtor japonské firmy BUSICOM. Napadlo jej přenést na programovatelnou strukturu univerzálního procesoru řešení úloh. Výsledný obvod byl označen 4004 a byl nazván mikroprocesor, měl šířku dat 4 bity, instr. soubor orientovaný hlavně na práci s čísly v BCD a na log. funkce. Vývoj však pokračoval dále. Firmou ZILOG byla vytvořena mikropočítačová sada MCS o čtyřech členech v čele s mikroprocesorem 4004. Souběžně s MCS4 byl vyvíjen mikroprocesor 8008 pro inteligentní terminál firmy Computer Terminals Co. V roce 1972 byl v čele sady MCS8 představen první osmibitový mikroprocesor. V roce 1974 firma INTEL přišla s typem 8080, který se stal na delší dobu světovým standardem. Asi rok po zavedení typu 8080 uvedla firma MOTOROLA mikroprocesor MC6800 a roku 1976 firma ZILOG mikroprocesor Z80-CPU, který rozvíjí dále architekturu typu 8080A. Od roku 1978 jsou středem pozornosti 16bitové mikroprocesory firmy INTEL, MOTOROLA, ZILOG atd.

Koncem 70. let nevznikaly jen 16bitové mikroprocesory, ale i monolitické mikropočítače, které představují samostatnou větev vývoje od poměrně primitivního typu 8048 až k pokročilým typům 8051, 8052 nebo Z8 atd.

V následující části této kapitoly si ukážeme jednotlivé typy a jak se procesorová technika vyvíjela. Od jednoduchého procesoru 8080A až ke složitým, teprve dnes zavádějícím se procesorům např. Pentium, PentiumPro a Pentium II. V samotném popisu jednotlivých procesorů si ukážeme, v čem je konkrétní procesor lepší, než jeho předchůdce, a jak jsou konkrétní nedostatky řešené.

5.1 Mikroprocesor 8080A

Prvním monolitickým mikroprocesorem 2.generace byl typ 8080 vyvinutý firmou INTEL v roce 1974 [2,19]. Jeho nedostatkem byla malá zatěžovací schopnost výstupů. Proto již po několika měsících byl nahrazen typem 8080A, který se od původního liší převážně jen výkonnějšími výstupními zesilovači. V 70. letech byl mikroprocesor 8080A pravděpodobně nejrozšířenějším 8bitovým mikroprocesorem na světě.

5.1.1 Popis mikroprocesoru 8080A

Je to úplná univerzální centrální jednotka (CPU) s 8bitovou paralelní strukturou. Je vyroben technologií MOS-N a odpovídá ekvivalentnímu počtu více než 5000 tranzistorů.

Má zápisníkovou paměť a instrukční soubor orientován především pro práci s jejími registry.

Napájení: Nutnost tři napájecích napětí - hlavní nedostatek tohoto procesoru.

Vstupní a výstupní napěťové úrovně s výjimkou vstupů hodinových signálů jsou slučitelné s obvody TTL. Vstupy vyžadují pro log.1 úroveň nejméně 3.3 V.

Hodinový signál: dvoufázový, dvě posloupnosti nepřekrývajících se impulsů.

Délka slova: základní slovo (Byte) je paralelní 8bitové. Zdvojené slovo má 2 slabiky sériově-paralelní. Delší slova zajišťujeme programově.

Sběrnice datová: 8bitová, obousměrná, třístavová.

Sběrnice adresová: šestnáctibitová, jen výstupní, třístavová. Umožňuje adresovat hlavní paměť do 64 K (65 536 Byte), vstupy a výstupy do 256.

Zapouzdření: keramické nebo plastové pouzdro DIL40 resp. DIP40 podle standardu JEDEC.

Aritmetická jednotka ALU: je založena na osmibitové paralelní sčítačce s obvody pro urychlení přenosu. Pracuje se dvěma pomocnými 8bitovými registry, které nejsou programově přístupné.

Indikátory: tzv. příznakové klopné obvody (**condition flip-flops**), jsou jednobitové registry, které se při výkonu některých instrukcí nastaví podle výsledku operace. Mikroprocesor 8080A má 5 indikátorů:

- CY - (carry) nastaví se do stavu log. 1, jestliže nastal přenos z nejvyššího řádu zpracovávaného bytu
- AC - (auxiliary carry-half carry) nastaví se do stavu log. 1, jestliže nastal přenos z nižší tetrády bytu do vyšší
- Z - (zero) nastaví se do stavu 1, jestliže výsledek arit. nebo log. operace je nulový
- S - (sign) nastaví se do stavu 1, jestliže výsledek operace je záporný
- P - (parity) nastaví se do stavu 1, jestliže parita výsledku je sudá

Zápisníková paměť: tvoří ji sedm 8bitových registrů: A,B,C,D,E,H,L. Jsou to registry samostatné a univerzální, dále mají tyto vlastnosti:

- A - (accumulator) střádač pro jednoslabikové operace
- HL - tato dvojice reg. slouží jako střádač pro dvouslabikové sčítání
- BC, DE - tyto dvojice mají rovněž nepřímé registrové adresování
- M,F - formálně je lze k předešlým registrům přiřadit, jsou však fiktivní.
- M je označení místa v paměti s adresou v registrech HL,
- F je označení bytu, který zahrnuje stavy všech 5 indikátorů takto:

S Z 0 A C 0 P 1 C Y

Obsah dvojice A a F představuje tzv. **Programové stavové slovo PSW** (program status word), se kterým operují instrukce:

PUSH PSW a POP PSW.

Zásobníková paměť (zásobník): není součástí mikroprocesoru 8080A, ale vytváří se v hlavní paměti. Pro adresování v zásobníku je mikroprocesor vybaven šestnáctibitovým **ukazatelem zásobníkové paměti - SP**.

Čítač instrukcí PC: slouží k automatickému adresování instrukcí v hlavní paměti a je 16bitový.

Registr instrukcí (řídící registr): je 8bitový. Udržuje operační znak instrukce po dobu jeho dekodování a vykonávání instrukce.

Dekodér instrukcí: řídí činnost mikroprocesoru při výkonu instrukcí.

Časovací a řídící obvody: patří k řadiči a v součinnosti s dekodérem instrukcí a také podle vstupních signálů a dvoufázových hodin $\phi 1$ a $\phi 2$ produkují řídící signály.

Způsoby adresování: kromě adresování čítačem instrukcí PC a ukazatelem zásobníku SP se používají implicitní adresy v operačních znacích, přímé operandy, nepřímé registrové adresování s pomocí dvojice BC, DE a zejména HL, přímé adresy ve skokových a volacích instrukcích i v instrukcích pro přesuny mezi střádači (A, HL) a paměti nebo mezi A a V/V branami.

Formáty dat:

- **Jednoslabiková data:**

Datová slabika má tvar

D7 D6 D5 D4 D3 D2 D1 D0

kde: D7 - nejvyšší řád slova **MSB** (most significant bit)

D0 - nejnižší řád slova **LSB** (least significant bit)

- **Dvouslabiková data:**

Slovo má tvar

D7 D6 D5 D4 D3 D2 D1 D0 D7 D6 D5 D4 D3 D2 D1 D0

Na sběrnici se slabiky slova objevují postupně (sériově paralelní režim), a to napřed nižší slabika, pak teprve vyšší.

– **Tříslabiková a víceslabiková data:**

Tvoří se analogicky jako v předchozím případě postupným řazením slabik v sériově paralelním režimu.

Formáty instrukcí:

– **Jednoslabikové instrukce:**

Operační znak

D7 D6 D5 D4 D3 D2 D1 D0

Především to jsou instrukce: **registrové (posuny a výměny mezi reg.), paměťové s registrovým adresováním, arit. a logické, rotace, návraty atd..**

– **Dvoslabikové instrukce:**

Operační znak Operand (adresa)

D7 D0 D7 D0

Patří zde instrukce s přímým osmibitovým operandem určené především pro **dosazovací, aritmetické a log. operace.**

– **Tříslabikové instrukce:**

Operační znak Dolní část operandu (adresy)

D7 D0 D7 D0

Horní část operandu (adresy)

D7 D0

Jsou to instrukce pracující s šestnáctibitovým operandem.

Počet instrukcí: Podle údajů výrobců má mikroprocesor 8080A celkem 78 instrukcí o délkách 1 nebo 2 nebo 3 bytů.

Přerušeni: Přerušeni probíhajícího programu může být vyvoláno vstupním signálem INT=1, pokud INTE=1.

Po akceptování požadavku signálu INT se automaticky překlápí vnitřní klopný obvod INTE do stavu logické hodnoty "0", takže další přerušeni je nemožné, dokud neuplatníme povolovací instrukci EI (enable interrupt).

Mikroprocesor 8080A má jen vstup maskovatelného přerušeni a nepřítomnost druhého vstupu pro nemaskovatelné přerušeni je jeho nedostatkem.

Start mikroprocesoru: Mikroprocesor je schopen činnosti ihned po ustálení napájecích napětí. Počáteční stavy jeho čítače a dalších registrů však závisejí na náhodných faktorech. Proto je třeba start mikroprocesoru zahájit připojením vnějšího signálu RESET. Tento signál trvá nejméně 3 periody hodin a nastaví nulový počáteční stav čítače instrukcí, registru instrukcí a klopných obvodů INTE a HLDA. Mikroprocesor zahájuje svou činnost od adresy "0", kde zpravidla začíná inicializační program.

5.2 Mikroprocesory I8086 a I8088

Zde se budeme zabývat problematikou 16bitových mikroprocesorů a to mikroprocesory řady I8086, I8088 [2,19].

Mikroprocesor 8086 je úplná 16bitová paralelní univerzální procesorová jednotka (CPU), která navazuje na známé mikroprocesory 8080A a 8085.

K hlavním přednostem 8086 patří:

- delší slovo,
- rozšíření instrukčního souboru zejména o aritmetické operace s 8bitovými i 16bitovými operandy včetně násobení a dělení a také operace s řetězci.

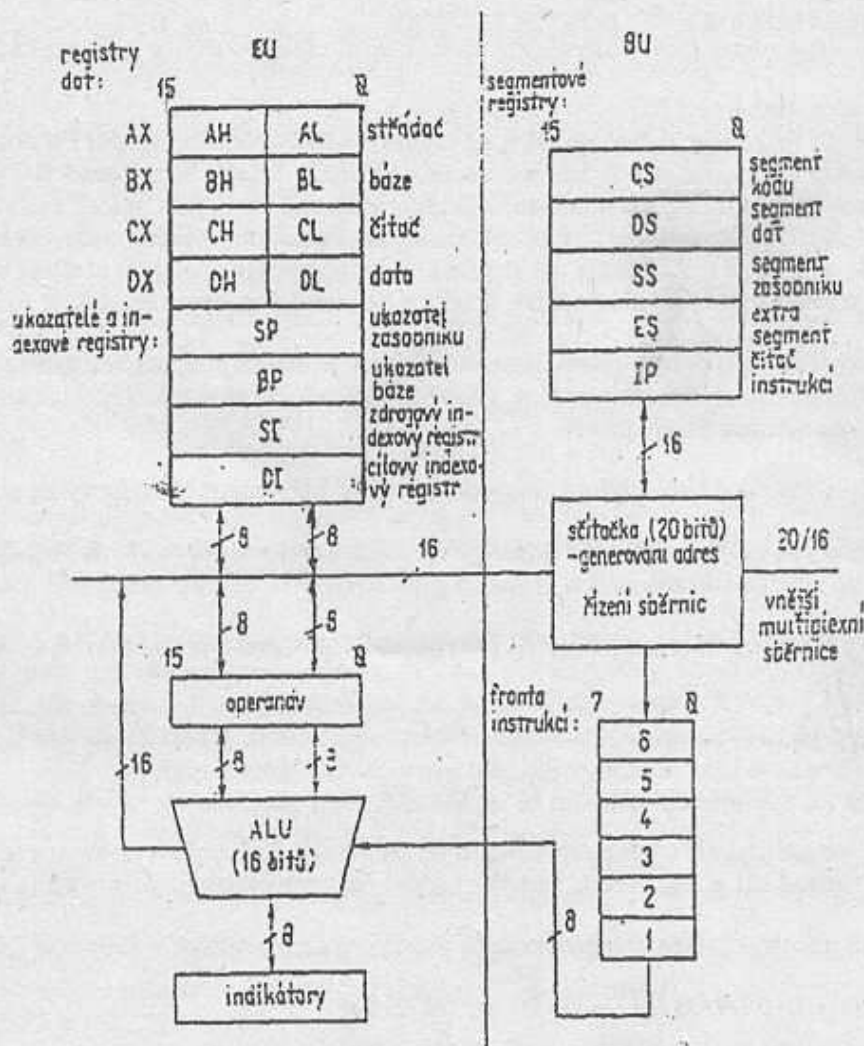
Mikroprocesor 8086 je ryze 16bitový, neboť má nejen vnitřní šířku toku dat 16 bitů, ale i s vnějším světem komunikuje prostřednictvím 16 vývodů datové sběrnice.

Mikroprocesor 8088 budeme probírat společně s typem 8086 (přestože bývá označován za 8bitový). Hlavní rozdíl mezi oběma typy spočívá v tom, že 8088 s vnějším prostředím komunikuje 8bitovou datovou sběrnici. Všechny ostatní odlišnosti 8088 a 8086 jsou méně zřetelné.

5.2.1 Základní vlastnosti mikroprocesorů 8086 a 8088

Mikroprocesor 8086 je vyroben unipolární technologií HMOS ve standardním pouzdře DIL 40 a odpovídá ekvivalentnímu počtu 29 000 tranzistorů.

Na obr. č. 5.1 [2] je uvedené zjednodušené blokové schéma mikroprocesoru 8086.



Obr. č. 5.1 Mikroprocesor 8086

Základní charakteristiky mikroprocesoru jsou:

Napájení: $U_{cc} = +5V \pm 10\%$, 275 mA maximálně.

Vstupní a výstupní napěťové úrovně: s výjimkou vstupů hod. signálu jsou slučitelné s obvody TTL.

Hodinový signál: jednofázový, úrovně napětí (-0.5 až +0.6V) a (3.9 až $U_{cc} + 1V$).

Sběrnice datová: 16bitová, obousměrná, multiplexovaná s částí adresové sběrnice.

Sběrnice adresová: 20bitová jen výstupní, třístavová, zčásti multiplexovaná se stavovou informací.

Stavová informace: na 8 vývodech S7, ..., S0, zčásti multiplexovaných s adresovou sběrnicí, se vydává stavová informace pro řízení součinnosti s dalšími obvody.

Základem architektury 8086 jsou dva zřetěžené subprocesory. Zde máme první příznaky progresivnější technologie vyjádřenou zřetěžením subprocesorů. Tyto subprocesory jsou:

1. **Operační jednotka EU (execution unit)** - vykonává pouze operace předepsané instrukcemi.
2. **Jednotka styku se sběrnicí BU (bus unit)** - označována též BIU (bus interface unit).

add. 1) **Operační jednotka EU.**

Je u 8086 a 8088 stejná. Obsahuje 16bitovou aritmeticko-logickou jednotku ALU, která zpracovává operandy a v souladu s výsledky nastavuje indikátory.

Registry, tok dat a adresy jsou 16bitové. Pro styk s pamětí však BU adresy upravuje na 20bitové.

add. 2) **Jednotka styku se sběrnicí BU**

Je u 8086 a 8088 funkčně schodná, ale liší se rozdílností vnějších sběrnic. BU pro EU obstarává veškerý styk s vnější sběrnicí a navíc v době, kdy subprocesor EU je zaměstnán. Subprocesor BU vybírá z paměti další instrukce nebo zajišťuje čtení operandu či zápis výsledků ve styku s pamětí i se vstupy/výstupy. Instrukce, které BU vybere, ukládá do registrů "**fronty instrukcí**", které představují paměť FIFO. Fronta v 8086 může uchovat 6 slabik (8088 - 4 slabiky). Subprocesor BU usiluje o přečtení další instrukce, jakmile se ve frontě vyprázdní alespoň dvě slabiky a EU nemá požadavek na styk se sběrnicí.

Subprocesory EU a BU pracují relativně nezávisle, a tak je možné překrytí fáze zápisu, čtení i výběru instrukce a fázemi vykonávání předchozí instrukce. Tím získáváme efektivní zvětšení rychlosti mikroprocesoru a také klesají nároky na rychlost hlavní paměti.

Využití fronty instrukcí tedy zvyšuje průchodnost, neboť EU nemusí čekat na výběr příští instrukce.

Omezení fronty na 6 slabik, popř. na 4 slabiky je výsledkem optimalizačních úvah. Je dobré si uvědomit, že v době, kdy BU čte instrukci do zásobníku, případný požadavek EU o přístup na sběrnicí musí počkat atd.

Univerzální registry EU: Mikroprocesor 8086/88 je vybaven 16bitovými registry (obr. č. 5.1 nahoře vlevo), které dělíme na dvě skupiny:

- **Zápisníková paměť, skupina HL** tvořena čtyřmi registry dat. Každý z registrů skupiny HL, lze adresovat a používat jako 16bitový nebo jako dva nezávislé 8bitové registry. Registry dat lze bez omezení používat ve většině arit. a log. operací.
- **Skupina ukazatelů a indexových registrů, neboli skupina PI.** Registry z této skupiny se rovněž mohou podílet na většině arit. a log. operací, "**takže všechny registry vyhovují pojmu střadač**".

Indikátory: 8086/88 má v EU celkem 9 jednobitových indikátorů uspořádaných v 16bitovém slově takto:

D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
- - - - OF DF IF TF SF ZF - AF - PF - CF

Šest z uvedených indikátorů jsou **stavové indikátory**. Stavové indikátory jsou:

- CF** (carry flag) - nastaví se do log. hodnoty 1, jestliže nastal přenos nebo výpůjčka do MSB výsledku operace
- PF** (parity flag) - nastaví se, jestliže parita výsledku je sudá
- AF** (auxiliary carry flag) - nastaví se, jestliže nastal přenos z nižší tetrády slabiky v AL do vyšší nebo výpůjčka z vyšší tetrády AL
- ZF** (zero flag) - nastaví se, jestliže je výsledek operace nulový

- SF** (sign flag) - nastaví se, jestliže výsledek operace je záporný
- OF** (overflow flag) - nastaví se při aritmetickém přeplnění
- DF** (direction flag) - jeho nastavení programem podmiňuje autodekrementaci indexových registrů SI a DI při zpracování řetězců, tj. jejich zpracování od vyšších adres k nižším. Nulování DF umožňuje autoinkrementaci, tj. zpracování řetězců "zdola nahoru"
- IF** (interrupt - enable flag) - nastavený IF umožňuje akceptovat externí maskovatelný požadavek na přerušení
- TF** (trap flag) - po nastavení TF přechází CPU do krokového režimu pro účely ladění

5.2.2 Organizace paměti, segmentace a start mikroprocesoru

5.2.2.1 Organizace paměti

Mikroprocesor 8086 pracuje s 20bitovou adresou pro adresování v hlavní paměti (hexadecimálně: 00000 - FFFFF). Paměť je slabikově (bytově) orientována, tj. CPU adresuje až 1 MByte, kde 1 MB = 1 048 576 Byte.

Instrukce, slabiky nebo 16bitová slova dat lze ukládat na libovolné adresy. Přitom slovo o dvou bytech se ukládá do dvou sousedních buňek tak, že nižší byte slova přijde na specifikovanou adresu a vyšší byte na následující vyšší adresu. Viz. příklad na obr. č. 5.2.

726	727	adresa/číslo		
0	2	8	4	hexadecimálně
0000	0010	1000	0100	binárně

Uložení čísla 8402H=1000010000000010B na sudé adrese 726.

Obr. č. 5.2

Doporučuje se slova dat ukládat přednostně na **sudé adresy**, neboť čtení nebo zápis slova se realizují v **jednom cyklu paměti**, pro **liché adresy** je na tutéž operaci třeba **dvou cyklů**. Jestliže se toto doporučení nedodrží, dojde k jistému zpomalení činnosti.

Architektura 8086/88 vyžaduje, aby programy mikropočítače pokládaly hlavní paměť rozdělenou na **segmenty**. **Segment** je souvislé pole resp. odpovídající část paměti o délce 64 kB a jeho počáteční adresa musí být násobkem 16. Jiná omezení nejsou, takže segmentů může být neomezené množství. Mohou těsně sousedit nebo být odděleny nevyužitými adresami a také se mohou zčásti nebo i úplně překrývat.

Subprocesor **BU** obsahuje kromě čítače instrukcí **IP** (instruction pointer) 4 **segmentové registry** **CS, DS, SS a ES**, z nichž každý udržuje horních 16 bitů skutečné 20bitové adresy počátku jednoho ze segmentů, tj. **bázové adresy segmentu**. Těchto 16 bitů představuje tzv. **segmentovou adresu**. Segmenty, jejichž adresy jsou právě v segmentových registrech, se nazývají **aktuální segmenty (current segments)**.

Pro adresování v aktuálním segmentu postačí 16bitový **offset**, který představuje **intra-segmentovou adresu**. Offset může být zadán buď v instrukci, nebo je v **IP, BU** nebo **EU**, v indexových registrech **SI a DI**, nebo v registrech **SP a BP**.

Intersegmentová adresa: představuje seskupení dvou veličin, segmentové adresy a offsetu. Obě tyto veličiny pojmáme jako 16bitové hodnoty bez znaménka. Nejnižší adrese v segmentu odpovídá nulová hodnota offsetu.

Fyzická adresa (physical address): vztažená k hlavní paměti se generuje v **BU** takto: "do dolních" 16 bitů 20bitové sčítačky se zavede příslušná segmentová adresa a posune se vlevo o 4 bity. Tím se převede na bázovou adresu segmentu, která má vždy 4 nejnižší bity nulové. Pak se přičte **intra-segmentová adresa** resp. **offset** dodaný buď z **IP**, nebo **EU**.

Fyzická adr. = segmentová adr. * 2⁴ + offset

Ukazatel (pointer): se používá při adresování dat mimo aktuální segmenty a představuje tedy intersegmentovou adresu, která je v paměti uložena definovaným způsobem:

- jedno slovo obsahuje hodnotu offsetu a je na dvou nižších adresách,
- další slovo představuje segmentovou adresu a je na dvou vyšších adresách.

Zapisování slova je znázorněno na obr. č. 5.3.

4		5		6		7		adresa/číslo
6	5	0	0	4	C	3	B	hexadecimálně
0110	0101	0000	0000	0100	1100	0011	1011	binárně

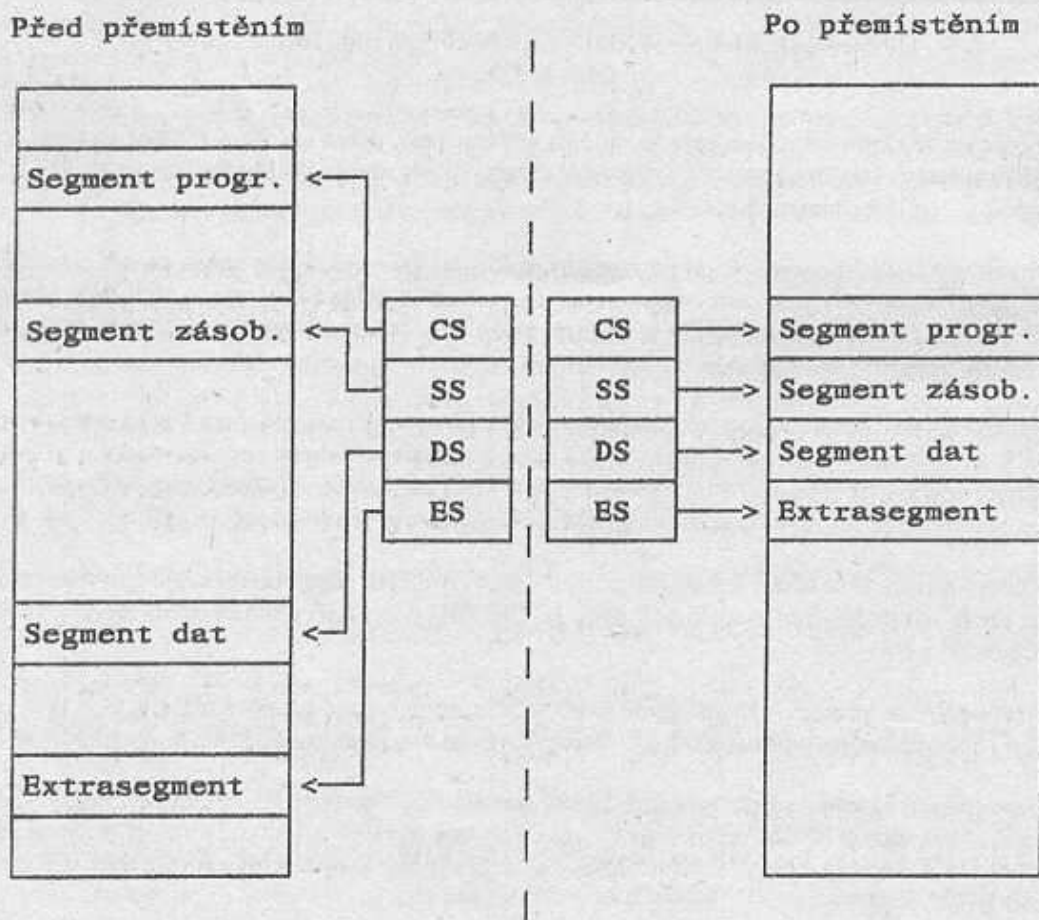
Obr. č. 5.3

Uložení ukazatele na adr. 4, posunutí = 0065H, segmentová adresa = 3B4CH

5.2.2.2 Segmentace paměti

Zde se poprvé setkáváme se segmentovou organizací paměti. Segmentace a relativní adresování s pomocí segmentových registů umožňuje používat úspornou 16bitovou adresu při dostatečně delší fyzické adrese, ale především dovoluje práci s **dynamicky přemístitelnými programy**. Dynamická přemístitelnost dovoluje multiprogramovým a jiným složitým systémům účinně využívat hlavní paměť.

Příklad dynamického přemístění segmentů je znázorněno na obr. č. 5.4.



Obr. č. 5.4

Dynamická přemístitelnost vyžaduje dodržování následujících pravidel:

- program nesmí sám iniciovat nebo měnit svůj segmentový registr,
- nesmí přímo adresovat buňku mimo aktuální segment programu (každý ofset musí být relativní k bázi v segmentovém registru).

Segmentovým registrům v BU jsou přidělena symbolická jména naznačující jejich využití:

Segmentový registr CS:

(code segment) definuje **aktuální segment programu**. K báze adrese v něm obsažené se vztahují všechny paměťové reference při výběru instrukcí z paměti. Obsah registru IP je interpretován jako ofset instrukce. Registr IP plní úlohu čítače instrukcí, ale jen v rámci segmentu určeném obsahem CS.

Segmentový registr SS:

(stack segment) definuje **aktuální segment zásobníku** s délkou 64 kB. Ukazatel zásobníku SP v EU pak obsahuje posunutí vrcholu zásobníku (top of the stack) od báze adresy segmentu. I zde zásobník roste shora dolů, tj. vrchol zásobníku je poslední obsažená adresa.

Šířka dat zásobníku je 16 bitů a data se ukládají a vybírají po slovech tak, že se SP dekrementuje o 2 a pak se uloží, nebo naopak, SP inkrementuje o 2, když se slovo vybere.

Segmentový registr DS (data segment) a ES (extra segment):

určují dva aktuální segmenty dat. Program může při práci s daty orientovat činnost BU kteréhokoliv z aktuálních segmentů, ale předpokládá se, že proměnné jsou většinou v **aktuálním segmentu dat DS**, zatímco **extra-segment ES** je rezervní.

Ofset pro proměnné v segmentu dat vypočítává EU v závislosti na způsobu adresování.

Řetězce:

představují data, která se adresují odlišně od ostatních. Předpokládá se, že zdrojový operand instrukce zpracovávající řetězec je aktuálním segmentu dat. Posunutí zdrojového operandu se získává ze **zdrojového indexového registru SI** v EU. Cílový operand instrukce řetězce je vždy v aktuálním **extra-segmentu** a jeho posunutí se získává z **cílového indexového registru DI** v EU. Instrukce, které řetězce zpracovávají, nastavují SI a DI samočinně a tak operace nad řetězcem probíhají postupně po bytech nebo po slovech.

Z předchozího výkladu plyne, že BU implicitně předpokládá přiřazení jednotlivých segmentových registrů určitým činnostem. Je to pohodlné, ale ne vždy optimální. Proto programátor může příslušné instrukci předřadit **prefix** (segment override prefix), čímž určí BU, který z jiných segmentových registrů instrukce použije při přístupu k proměnné.

Vyhrazené a rezervované oblasti paměti: na obou koncích adresové množiny jsou vyhrazené úseky pro specifikované funkce procesoru a také oblasti rezervované pro účely řízení styku s prostředím. Za **vyhrazené** (dedicated) označujeme adresy, které je krajně nevhodné používat pro jiné účely, než pro účely definované výrobcem mikroprocesoru. Za **rezervované** pokládáme adresy, které se doporučují používat pouze pro určité účely.

5.2.2.3 Start mikroprocesoru a inicializace registrů

Mikroprocesor je schopen činnosti ihned po připojení napájecího napětí, ale s náhodně nastavenými registry. Proto je třeba přivést bitový signál **RESET**. S jeho vzestupnou hranou mikroprocesor ukončí dosavadní činnost a je pasivní, dokud je **RESET=1**. Současně iniciuje některé registry tak, že:

1. fronta instrukce je prázdná
2. indikátor, **DS**, **SS**, **ES** a **IP** jsou vynulovány
3. **CS** obsahuje báze adresu **FFFFH**

Po sestupné hraně signálu **RESET** mikroprocesor zahájí svou činnost vždy od adresy **FFFF0H** (**IP = 0** a **CS = FFFFH**). Minimální délka impulsu **RESET** je na 4 periody hodin.

5.2.3 Vstupy, výstupy a DMA

Obdobně jako 8080A i mikroprocesor 8086 může pro styk s vnějšími zařízeními používat vstupní/výstupní brány adresované odděleně od paměti (isolated IO) nebo situované do hlavní paměti (memory-mapped IO).

Pro oddělené adresování vstupů/výstupů mikroprocesor 8086 používá 8bitovou adresu, a tak instrukcemi typu IN (input) a OUT (output) má přístup k 64 K 8bitových, popř. 32 K 16bitových vstupních bran a ke stejnému počtu výstupních bran.

Instrukce IN a OUT přenášejí data mezi příslušnou branou a mezi střadačem:

AL - při přenosu slabik

AX - při přenosu slov

Rozličné formy instrukcí IN a OUT pracují buď s přímou 8bitovou adresou v instrukci nebo s proměnnou 16bitovou adresou z registru DX.

8086 může s vnějším zařízením komunikovat najednou buď po 8 nebo po 16 bitech. Pro přiřazování vst./výst. adres platí, ve vztahu k sudým a lichým adresám, stejná doporučení a omezení jako při adresování paměti.

Přímý přístup do paměti - DMA:

- V **minimálním módu** mikroprocesor 8086, obdobně jako 8080A, pracuje se signály HOLD a HLDA, které odpovídají tradičním řadičům DMA typu 8257 a 8237.
- V **maximálním módu** je typické použití CPU se stykovým koprocесорem 8089. Jde o lokální víceprocesorový systém typu MULTIMASTER.

5.2.4 Přizpůsobení pro multiprocesorové systémy a stavové signály

1. Přizpůsobení pro multiprocesorové systémy

Vlastnosti mikroprocesoru 8086 usnadňují jeho použití pro výstavbu multiprocesorových systémů. Je zvláště vhodný pro malé systémy s jedním až třemi připojenými pomocnými procesory a pro větší systémy založené na sběrnici MULTIBUS.

2. Stavové signály

8086 má osm vývodů stavových signálů. V maximálním módu signály S2, S1 a S0 identifikují typ strojového cyklu (čtení, zápis, zastavení atd.).

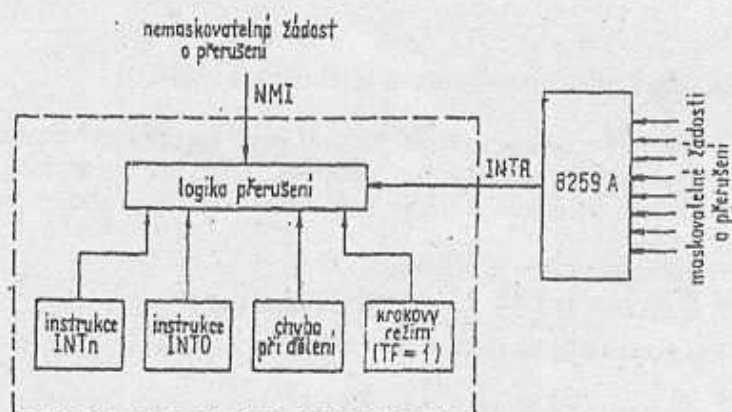
Signály S4 a S3 nezávisle na módu ukazují, který ze segmentových registrů byl použit pro tvorbu fyzické adresy pro stávající cyklus sběrnice.

Signál S5 zobrazuje stav indikátoru přerušení. Signál S6 informuje koprocесоры připojené k CPU o tom, že CPU právě ovládá společnou lokální sběrnici a signál.

Signál S7 není běžně využíván a označuje se za rezervní.

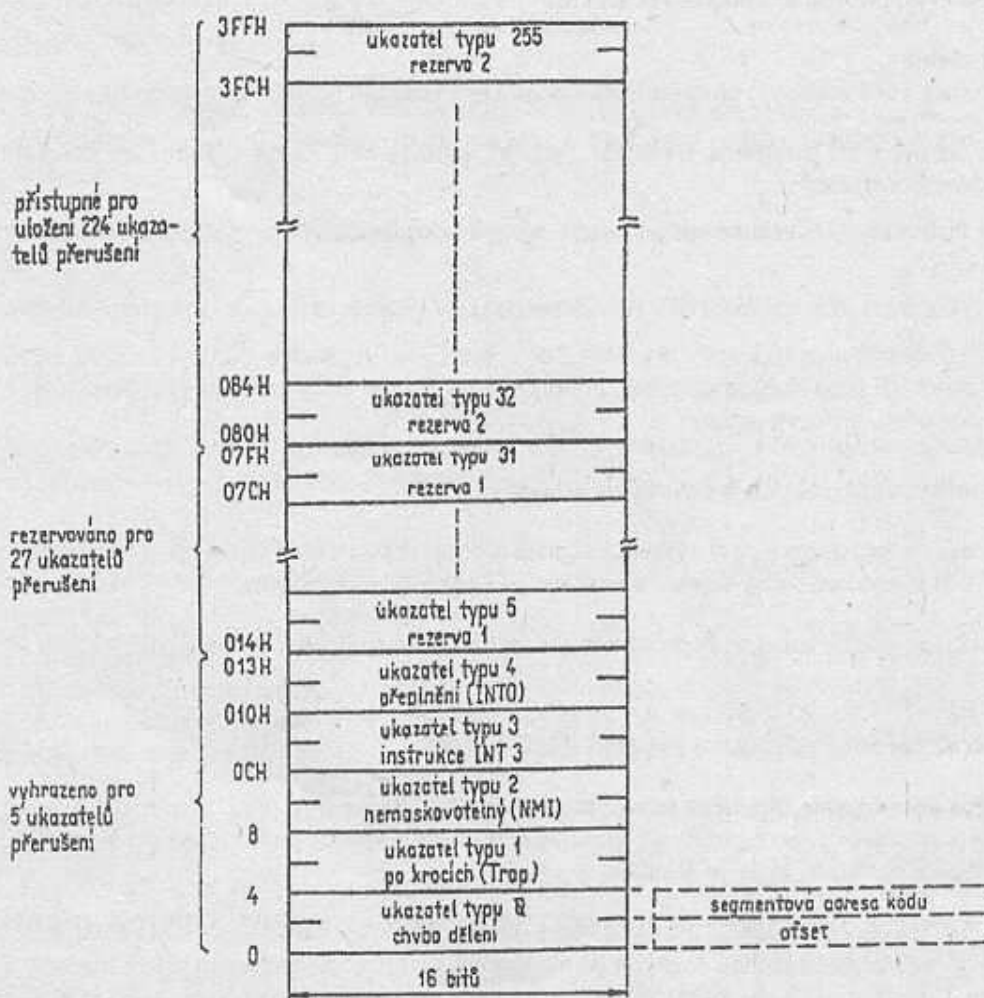
5.2.5 Přerušování programu

Každému původci přerušení je přiřazen číselný typ přerušení (type code), podle kterého jej mikroprocesor identifikuje. 8086 může zpracovat až 256 přerušovacích typů. Na obr. č. 5.5 [2] je to znázorněno.



Obr. č. 5.5 Přehled zdrojů přerušení u 8086

Přerušení může být vyvoláno nejen signály INTR a NMI, ale i vnitřně instrukcemi INTn a INTO nebo samotným mikroprocesorem.



Obr. č. 5.6

Na obr. č. 5.6 [2] je podrobněji zobrazeno dolních 1024 buněk hlavní paměti mikropočítače s 8086, které jsou rezervovány pro uložení ukazatelů, tj. intersegmentových adres pro obsluhu přerušení. Vidíme, že každému z 256 možných typů přerušení přísluší ukazatel obsahující v dolních dvou bytech ofset a v dalších dvou segmentovou adresu segmentu, ve kterém je uložen podprogram přerušení. Prvních 20 slabik je vyhrazeno pro 5 typů přerušení s pevně přidělenou funkcí, s jejichž obsluhou je třeba vždy počítat. Navazujících 108 slabik (Rezerva 1) se doporučuje rezervovat pro 27 typů přerušení, s jejichž využitím se počítá při spojování 8086 s dalšími výrobky firmy INTEL.

Celkem je 32 typů přerušení, které by neměly být používány libovolně. Zbývajících 896 (Rezerva 2) může být využito pro ostatních 224 typů přerušení bez omezení.

Externí přerušení:

Bitový vstup maskovatelného přerušení INTR (interrupt request) se obvykle budí z programovatelného řadiče přerušení 8259A, ke kterému lze přivést až 8 signálů požadavků z vnějších zařízení.

Prioritu požadavků, typy přerušení a modus činnosti 8259A určuje mikroprocesor a lze je měnit i za chodu programu.

Maskovatelnost funkce signálu INTR spočívá v tom, že akceptování žádosti o přerušení lze programem předem povolit nebo zakázat.

Bitový signál nemaskovatelného přerušení NMI mikroprocesor zachycuje do registru a reaguje na něj na konci probíhající instrukce. NMI je v mikroprocesoru synchronizován hodinovým signálem a požaduje se, aby trval

nejméně 2 periody hodin. Používá se především k signalizaci katastrofických situací (hrozící ztráta napájení, chyba parity v paměti atd.).

Proto MNÍ má vyšší prioritu než požadavek INTR.

Vnitřní přerušení:

U mikroprocesoru 8086 může být přerušení způsobeno i bez vnějšího požadavku a to **čtyřmi** příčinami:

1. Vykonání instrukce INTn vyvolá přerušení typu "n", protože $n=0, \dots, 255$. Takto lze vyvolat kterékoliv přerušovací podprogramy.
2. Vykonání instrukce INTO (interrupt on overflow) způsobí přerušení typu 4, pokud je nastaven indikátor přeplnění OF.
3. Ihned po vykonání instrukce dělení DIV (divide) nebo IDIV (integer divide) dojde k přerušení typu 0, jestliže nastalo dělení nulou.
4. Řídící indikátor TF (trap flag) je nastaven, mikroprocesor automaticky generuje přerušení typu 1 po **každé** instrukci. Jde o tzv. krokový režim.

5.2.6 Tabulky elektrických a časových údajů

Základní informace o funkci vývodů a signálů mikroprocesoru 8086/8088 bývají uvedeny v tabulkách a grafech. V nich jsou soustředěny údaje o elektrických i časových parametrech.

Vystavení obvodu mezním hodnotám může ovlivnit jeho spolehlivost a jejich překročení může obvod zničit.

5.2.7 Instrukční soubor mikroprocesorů 8086 a 8088

Instrukční soubor je zapsán na úrovni jazyka symbolických adres, popř. assembleru ASM86. Pravidla tohoto jazyka a překladače však podrobně probírat nebudeme, zájemci mohou tuto problematiku podrobně studovat v odborné literatuře, které je dostatek.

Soubor instrukcí pro 8086/88 má navíc velmi důležité instrukce násobení, dělení a další. Téměř všechny aritmeticko-logické instrukce mohou používat jako zdrojový a cílový operand nejen obsah registrů, ale i buňky hlavní paměti. Také lze volit mezi 8bitovými a 16bitovými operacemi. Hlavním přínosem je však adresování hlavní paměti.

Důsledně se zde aplikuje relativní adresování s využitím segmentových registrů a to usnadňuje zavádění přemístitelných programových a datových modulů. Tato základní úroveň relativního adresování paměti je implicitní, tj. uplatňuje se vždy, aniž by ji programátor musel specifikovat.

Další významné zdokonalení je ve skupině instrukcí podmíněných skoků, které umožňují realizaci i kombinovaných aritmeticko-logických testů (díky zavedení indikátoru přetečení) a také instrukci tzv. smyčkových skoků (cyklus: "For ... Next").

Koncepce instrukčního souboru mikroprocesoru 8086/88 respektuje potřeby efektivního překladu z vyšších programovacích jazyků a provozu multiprogramových systémů. Je zde vyřešena spolupráce s numerickým koprocesorem 8087.

5.2.8 Formáty dat a strojových instrukcí

Většina instrukcí může pracovat s daty 8bitovými nebo 16bitovými, přičemž **délka slova** je obvykle určena **příznakovým bitem v operačním znaku**. Na úrovni jazyka ASM86 je toto rozlišování určováno deklaracemi nebo pomocí typových identifikátorů.

Veličiny o délce 32 bitů se také vyskytují u operací násobení, avšak mají charakter výsledku a nikoliv operandu.

Aditivní aritmetické operace se realizují ve **dvojkovém doplňkovém kódu**, ale je možné je používat i pro práce s čísly bez znaménka. Pro násobení a dělení existují dva typy instrukce:

- celočíselné hodnoty bez znaménka,
- celočíselné hodnoty se znaménkem.

Formáty instrukcí jsou podstatně rozmanitější, neboť instrukce může obsahovat 1,2,3,4 a 6 bytů, z nichž vždy 1 slabika nebo první 2 slabiky tvoří operační znak a další slabiky představují operandy nebo adresovou informaci.

Uvedu příklad. Běžná přesunová instrukce

MOV d,s

má ve strojovém kódu celkem 11 variant s délkou 2,3,4 nebo 6 bytů v závislosti na typu adresové informace "d" a "s".

Jedna instrukce na úrovni assembleru může reprezentovat více strojových instrukcí, které plní tutéž funkci, ale s rozdílně adresovanými operandy i výsledky.

- **Jednoslabikové instrukce:**

Tyto krátké instrukce obsahují jen 8bitový operační znak a patří k nim jednak řídicí instrukce, jednak instrukce, které využívají implicitního adresování.

- **Víceslabikové instrukce:**

U těchto instrukcí je v prvních dvou slabikách umístěn operační znak, který se skládá z jednoho nebo dvou polí číselně definujících typ instrukce a z jednobitových až třibitových polí příznaků.

Víceslabikové instrukce mohou být:

- a) instrukce se 2 operandy, kdy 2. operand je registr,
- b) instrukce se 2 operandy, kdy 2. operand je přímý,
- c) instrukce s 1 operandem.

5.2.9 Způsoby adresování a trvání instrukčního cyklu

Instrukce mikroprocesorů 8086/88 nejčastěji operují s jedním nebo se dvěma operandy s tím, že výsledek přepíše jeden z operandů. Operandy mohou být přímé nebo mohou být umístěny v registrech, či v paměti. Rozlišujeme tedy adresové reference na registry a na paměť.

Délka instrukčního cyklu v souvislosti s možností využívání různých způsobů adresování způsobuje, že varianty téže instrukce mohou mít nejen různý počet bytů, ale i nestejnou dobu instrukčního cyklu.

Příklad:

ADD r2,r1 ;sečte obsah obou registrů a trvá
;3 periody hodin

ADD r,m;k obsahu registru "r" přičítá obsah ;paměťového místa "m" a trvá celkem
;(9+EA) period hodin, kde ;EA-závisí na na použitém způsobu
;adresování

5.2.10 Programová slučitelnost s 8080A a konverzní program CONV86

Návaznost instrukčního souboru 8086/88 na instrukce 8080A je největší u aritmeticko-logických operací včetně rotací.

Ostatní instrukce jsou odlišné a také je odlišný i způsob jejich použití. Například pro 16bitové aritmetické operace 8080A (DAD,INX a DCX) je nutné použít více instrukcí, což vyplývá z odlišného způsobu ovlivňování indikátorů.

Podobné je to i u instrukcí přesunů dat typu LDAX a STAX mikroprocesoru 8080A. Původní operační znaky je třeba interpretovat dvěma instrukcemi pro 8086, protože příslušné ekvivalenty 16bitových registrů nelze používat jako adresové ukazatele.

Pro snadné převody programů z jazyka ASM80 do ASM86 je určen konverzní program CONV86. Tento program připomíná překladače z vyšších jazyků, ale na rozdíl od nich se výsledný program v jazyku ASM86 neoptimalizuje.

5.3 Mikroprocesory 80186 a 80188

Mikroprocesor 80186 byl zaveden v r.1982 jako úplná 16bitová univerzální procesorová jednotka (CPU), která vznikla inovací mikroprocesoru 8086 [2,19].

Oba typy jsou vyrobeny technologií HMOS II, mají okolo 100 000 tranzistorů, jsou úplně programově slučitelné s 8086/88 a kromě CPU zahrnují:

1. **Programovatelný řadič přerušení** podobný obvodu 8259A. Obslouží 5 vnějších požadavků (z toho 1 nemaskovatelný NMI) a 5 vnitřních požadavků.

2. **Programovatelný řadič DMA pro 2 kanály.**

Umožní přenos typu:

- paměť -- V/V
- V/V -- paměť
- V/V -- V/V
- paměť -- paměť

Řízený přenos dat může probíhat po "Bytech" nebo po "Slozech".

Řadič DMA nemůže plně nahradit stykový mikroprocesor 8089, ale má pokročilejší vlastnosti než obvod 8257.

3. **Programovatelný trojnásobný čítač-časovač.**

Dva z 16bitových čítačů jsou plně použitelné a mají vyvedené čítačí vstupy i výstupy (TMRIN a TMROUT). Třetí z čítačů není přímo vyveden a je určen pro řízení práce mikroprocesoru v reálném čase a pro tvorbu časových zpoždění.

4. **Generátor hodinového signálu.**

Vyžaduje připojení piezoelektrického rezonátoru o dvojnásobku frekvence hodin. Je možné také i připojení vnějších hodin.

5. **Obousměrné budiče datové sběrnice.**

Nahrazují funkci obvodů 8286/87 na lokální sběrnici.

6. **Řadič lokální sběrnice.**

Nahrazuje činnost obvodu 8288 pro lokální sběrnici.

7. **Dekodéry a logika WAIT.**

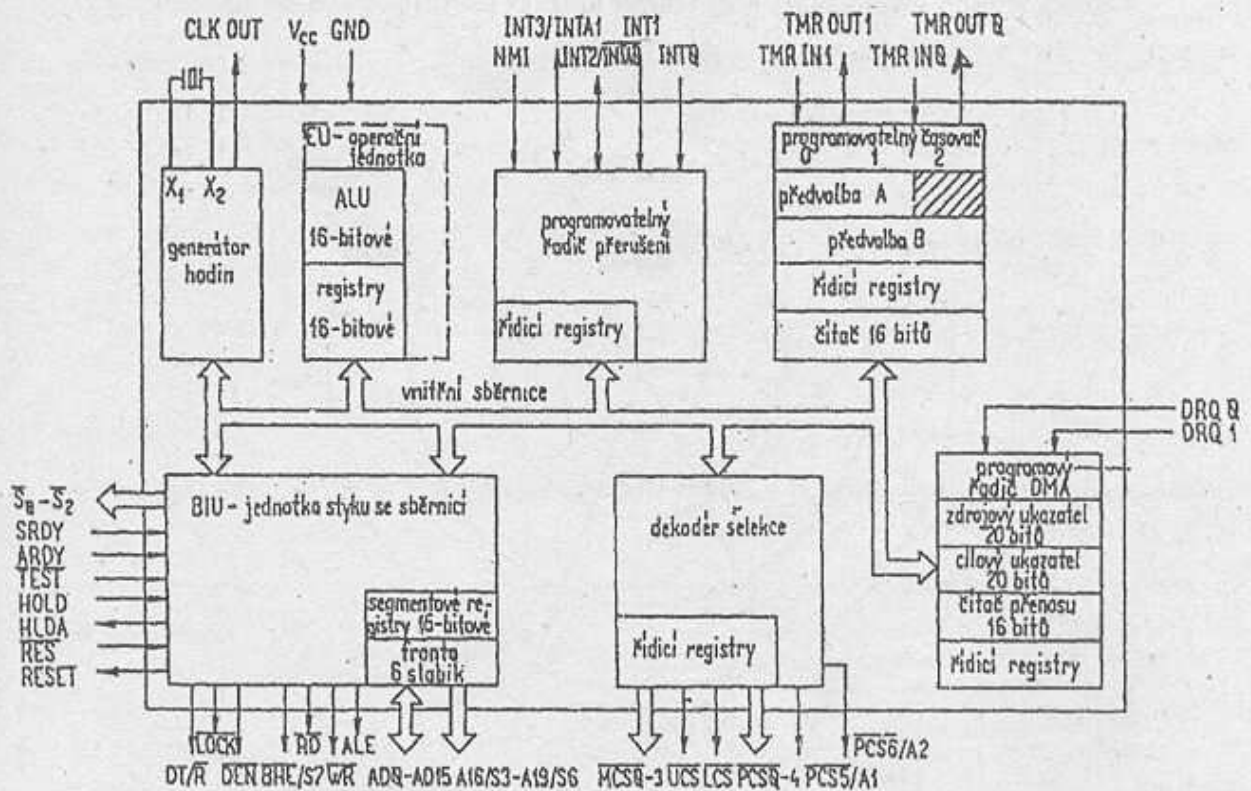
Nahrazuje až 13 obvodů malé a střední integrace, dosud používaných v dekodérech adres pro paměť a pro vstupy/výstupy.

Poskytuje selekční signály pro 6 nezávislých bloků paměti a pro 7 dalších vnějších obvodů.

Všechny tyto signály jsou zavedeny i do programovatelné logiky WAIT.

Souhrnně vzato, 80186/188 nahrazuje ve srovnání s 8086/88 15 až 20 obvodů.

Instrukční soubor i adresovací možnosti 80186 jsou totožné s možnostmi 8086. Pro usnadnění práce se zabudovanými funkcemi, i jako rozšíření stávajících funkcí, bylo navíc přidáno 10 nových instrukcí.



Obr. č. 5.7 Zjednodušené blokové schéma mikroprocesoru 80186

5.4 Mikroprocesor 80286

Mikroprocesor typu 80286 byl zaveden firmou INTEL v r. 1984 [2,19]. Je to 16bitový univerzální mikroprocesor, navržený se zřetelem na potřeby výstavby tzv. velkých osobních počítačů, představovaných zejména typem PC AT firmy IBM.

Tento mikroprocesor vznikl revoluční inovací typu 8086. Vychází z jeho základních vlastností (instrukce, segmentace paměti), ale architektura 80286 se čtyřstupňovým zřetězením je podstatně rozvinutější a poskytuje možnosti, které se s 8086/88 nedají zajistit.

5.4.1 Ochrana paměti

Ochrany virtuální paměti (memory protection) mají zabránit nežádoucímu přístupu do paměti (počítačová kriminalita, neúmyslný zásah atd.).

Jistou ochranu paměti poskytuje její segmentace, neboť zabráňuje přístupu mimo aktuální segmenty. Nechrání však před chybnou segmentací a ani před chybnými zásahy uvnitř aktuálního segmentu.

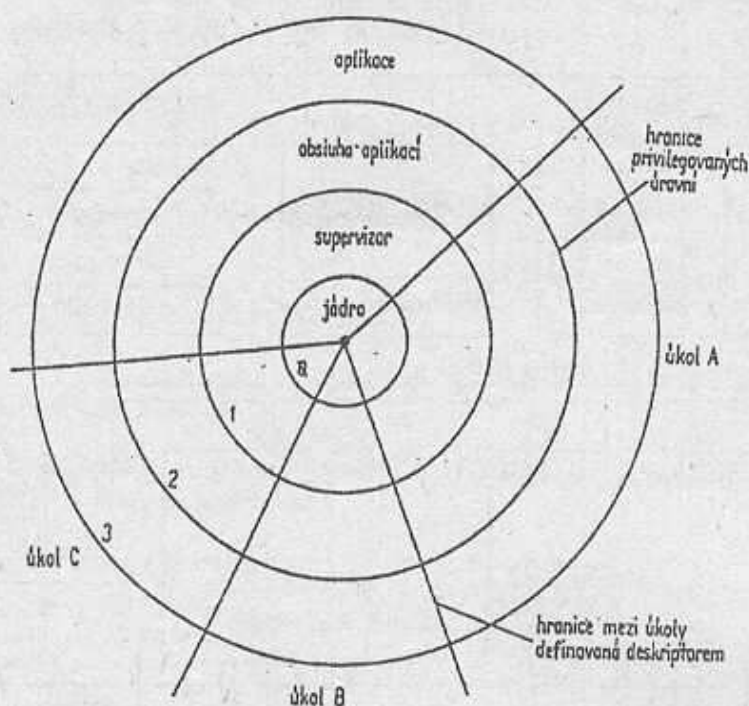
Segmentace se ovšem nezavádí pro ochranu paměti, ale pro přemístitelnost programů a zkrácení adres.

Nicméně segmentace příznivě působí na ochranu.

U dosavadních počítačů je dosti rozšířen jednoduchý způsob dvoustupňové ochrany paměti rozdělením virtuální paměti mezi uživatele a funkce supervizora tak, že nežádoucí přístupy jsou automaticky vyloučeny.

80286 je prvním monolitickým mikroprocesorem, který programem umožňuje volit nejen dvoustupňovou, ale i třístupňovou a čtyřstupňovou hierarchickou ochranou paměti a podporuje ji obvodovými prostředky, což zajišťuje vysokou rychlost, spolehlivost a krátkost příslušných programů.

Základní principy čtyřstupeňové hierarchické ochrany paměti jsou uvedeny na obr. č. 5.8 [2].



Obr. č. 5.8

Rozmístění úkolů ve virtuální paměti a jejich rozložení na privilegované úrovně při čtyřstupeňové ochraně paměti

Programy odpovídající jednotlivým **úkolům** jsou rozprostřeny do 4 privilegovaných úrovní:

- Jádro (kernel):** úroveň 0, má nejvyšší stupeň privilegovanosti. Patří sem programy řízení virtuální paměti, mapování (scheduling tasks), komunikace mezi úkoly apod. Jsou nejtěsněji spjaty s obvodovými funkcemi.
- Supervizor:** úroveň 1. Řídí vstupy/výstupy, umísťuje vyrovnávací paměti dat, plánuje globální úlohy apod. Těsně souvisí s funkcemi jádra, více však závisí na aplikacích, zatímco jádro je spíše programovým prodloužením funkcí řídicích obvodů mikroprocesoru.
- Podpora uživatelských programů:** úroveň 2. Patří sem univerzálně použitelné systémové programy, knihovny podprogramů apod.
- Uživatelské programy:** úroveň 3. Nejnižší úroveň privilegovanosti.

Úkol může být zpracován na kterékoliv úrovni nezávisle na procedúře, která se právě vykonává. To poskytuje možnost strukturování operačního systému na množinu procedur, které mohou být přímo volány, ale jsou před uživatelem chráněny. Každý úkol má řízený přístup do dvou oblastí virtuální paměti, oblasti **veřejné (public)** a oblasti **privátní (private)**. Jsou to:

Tabulka globálních deskriptorů: seznam segmentů přístupných všem systémovým úkolům.

Tabulka lokálních deskriptorů: seznam segmentů příslušných jen jednomu úkolu. Takovou tabulku zahrnuje každý úkol, jako popis svého stavu.

Deskriptor v systémech s 80286 představuje datovou strukturu o 8 Bytech umístěnou ve virtuální paměti.

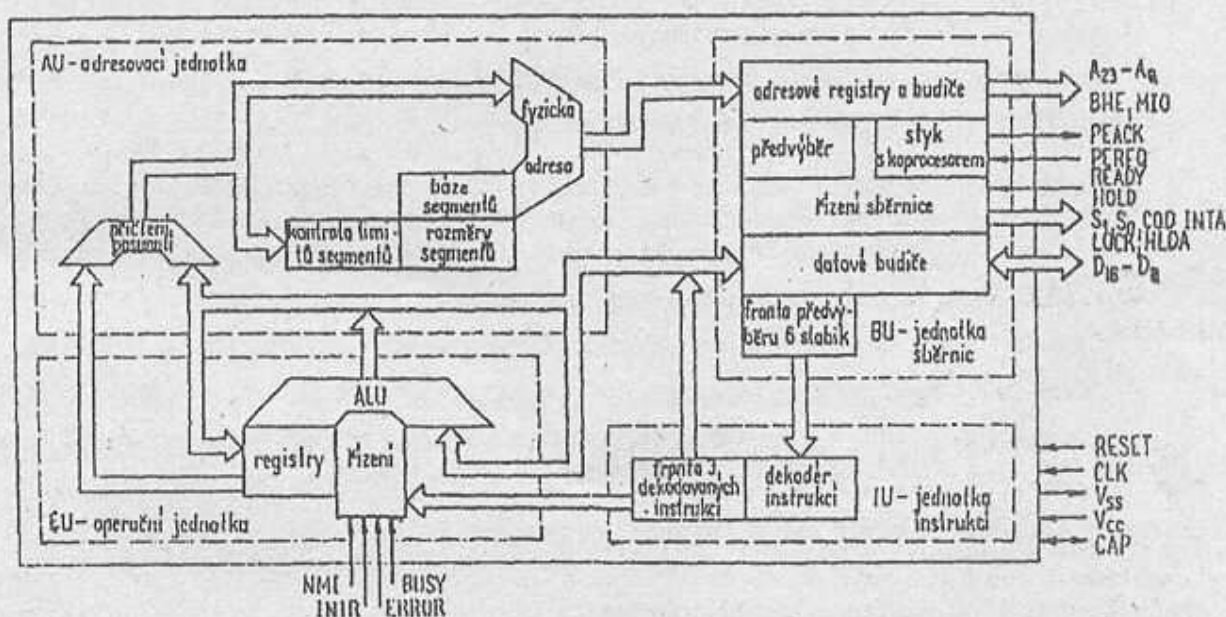
Pro každý segment obsahuje:

- 16 nulových bitů jako rezervu pro zajištění programové slučitelnosti 80286 s 32bitovým mikroprocesorem 80386
- bázovou adresu segmentu (24 bitů)
- rozměr segmentu (16 bitů) nazývaný **limit**
- atributy**, neboli pole řízení přístupu (8 bitů), které definuje, jak může být segment použit (zda platí zákaz čtení nebo zápisu apod.)

Uvedený nástin prostředků řízení a ochrany paměti není úplný: viz. popis funkcí subprocesoru AU (adresovací jednotka).

5.4.2 Popis 80286

Tento mikroprocesor má asi 130 000 tranzistorů a je vyráběn technologií HMOS II ve stejném pouzdře jako 80186.



Obr. č. 5.9 Zjednodušené blokové schéma mikroprocesoru 80286

Vlastnosti:

– **Adresování:**

24 adresovými bity do 16 MB reálné paměti a 30 bity do 1 GB virtuální paměti.

– **Obvodová podpora:**

pro řízení virtuální paměti (memory management), pro ochranu paměti (memory protection) a pro mnohoúkolovost (multitasking) 80286 má tyto 2 volitelné pracovní módy:

- Modus reálné paměti** určený pro přímé vykonávání programů psaných a přeložených pro 8086/88, popř. 80186/188. V tomto módu pracuje bez virtuální paměti a bez funkcí ochrany paměti, ale podstatně rychleji než starší mikroprocesory.
- Modus chráněné virtuální paměti**, ve kterém se plně využívají možnosti nové architektury. I zde je zajištěna slučitelnost zdola s 8086/88, avšak na úrovni zdrojového kódu. To znamená, že veškeré převody související s virtuální pamětí a s funkcemi ochrany paměti, jsou transparentní pro programy 8086/88 po rekompilaci.

Na obr. č. 5.9 [2] je uvedeno blokové schéma mikroprocesoru 80286. U 80286 jsou zřetězeny 4 subprocesory:

- **Jednotka styku se sběrnici BU** obsahující i frontu na 6 Bytů instrukcí.
- **Jednotka dekodéru instrukcí IU** obsahující i frontu 3 dekodovaných instrukcí.
- **Operační jednotka EU** přibližně odpovídá EU u 8086.
- **Adresovací jednotka AU**, která zajišťuje převážnou část funkcí souvisejících s virtuální pamětí. Obsahuje mimo jiné sadu registrů, představující rychlou vyrovnávací paměť (**cache**), udržující **virtuální adresu, limit a atributy** pro všechny segmenty virtuální paměti vybrané pro aktuální úkol (task). Tato paměť dovoluje jednotce AU vytvořit virtuální adresu za 1 periodu hodin.
- **Vyrovnávací paměť tvoří registry:**
 - a) **Čtyři segmentové registry** po 64 bitech. Každý obsahuje 16bitový selector a 48bitový deskriptor aktuálního segmentu. Jen v módu reálné paměti určuje selector básovou adresu segmentu. Jinak představuje pouze index v tabulce deskriptoru v hlavní paměti. Procesor podle selektoru přepíše deskriptor z hlavní paměti do segmentového registru, kde jej pak používá pro rychlou tvorbu adresy a pro operace ochrany paměti.
 - b) **Tři registry** ukazující do 3 tabulek deskriptorů (globální, lokální a přerušení).
 - c) **Registr úkolu**, který ukazuje na segment rezervovaný pro popis stavu aktuálního úkolu.

Je to především subprocesor AU, který v součinnosti s BU-obvodovými prostředky prudce zrychluje funkce související s mnohaúkolovostí (multitasking). Zpracovává totiž automaticky přechody mezi úkoly při tzv. **dispečinku úkolů** a při přerušení. Programově lze přechod mezi úkoly zajistit jedinou instrukcí volání, skoku nebo přerušením.

Pro všechny neaktuální úkoly se ovšem příslušné informace udržují v hlavní paměti v **segmentu stavu úkolů** (task state segment).

Při zavádění úkolu CPU automaticky verifikuje ochranu paměti, pak uloží stav aktuálního úkolu do segmentu stavu úkolu a naplní příslušné registry novým stavem z téhož segmentu.

- **Sběrnice: Adresová (30 bitů) a datová (16 bitů)** jsou oddělené.
- **Rychlost:**

Podle pramenu z vyhodnocení zkušebních programů v jazyce PASCAL plyne, že 80286 při hodinové frekvenci 10 MHz je asi o 30 až 90 % rychlejší, než 32bitový moderní minipočítač VAX11/780 a asi 6 až 7 krát rychlejší, než 8086 s hodinami 5 MHz.

Komunikační úlohy lze rovněž zrychlit využitím stykových koprocesorů. Patří sem typ 82586 pro řízení lokální komunikace a typ 82501 pro sériový styk v sítích typu ETHERNET.
- **Pomocné obvody:**

kromě obecně použitelných pomocných obvodů lze s 80286 spojit generátor hodin 82284 a řadič sběrnice 82288 a arbitr sběrnice 82289.

Tyto speciální obvody jsou funkční obdobou obvodů 8284, 8288 a 8289 používaných s mikroprocesory 8086/88 a 80186/188.
- **Instrukční soubor:**

kromě všech instrukcí 8086 i 10 nových instrukcí pro 80186/188 má mikroprocesor 80286 dalších 16 instrukcí.

Nové instrukce odpovídají potřebám překladu z vyšších programovacích jazyků a tvorby pokročilých operačních systémů včetně funkcí řízení virtuální paměti a její ochrany.

5.5 Mikroprocesor 80386

Prvním 32bitovým mikroprocesorem, který firma INTEL uvedla na trh v roce 1981, byl model iAPX 432. Tento mikroprocesor musel být velmi brzy stažen z výroby, protože nikdy nezískal popularitu srovnatelnou s přecházejícími mikroprocesory.

Měl sice několik významných předností a navíc orientaci strojového kódu na strukturu jazyka ADA, což mělo usnadnit sestavení kompilátoru. Jeho hlavní nevýhodou však byla značná složitost (stavebnice integrovaných obvodů, z nichž 3 tvořily procesor).

Mikroprocesor 80386 [2,19] se vyrábí ve dvou základních provedeních, které se liší šířkou datové sběrnice vyvedené z pouzdra.

- Model 80386 DX z roku 1986 je skutečný 32bitový mikroprocesor, u něhož jak datová, tak i adresová sběrnice má šířku 32bitů.
- Model 80386 SX uveden na trh o tři roky později používá vnitřní reprezentaci dat délky 32bitů, avšak z pouzdra je vyvedena datová sběrnice poloviční šířky, tedy 16b.

Procesor 80386 SX je modifikace procesoru 80386 DX.

Čip procesoru 80386 DX je vyroben technologií CMOS s tloušťkou čar 1.5 μ m, obsahuje 275 000 tranzistorů a příkon 1.8 W. Je umístěn do pouzdra PGA se 132 vývody, což umožňuje vyvést datovou a adresovou sběrnici odděleně. Mikroprocesor 80386 SX je umístěn v pouzdře se 100 vývody.

Mikroprocesory se dodávají v provedení pro hodinovou frekvenci 20, 25 a 33 MHz.

Architektura procesoru 80386 je zjednušeně znázorněna na obr. č. 5.10 [9]. Mezi její významné vlastnosti patří **proudové zpracování instrukcí**. Ve struktuře mikroprocesoru k tomu slouží **dvě fronty instrukcí**:

1. Fronta obsahuje 16 slabik instrukcí tak, jak jsou přečteny z hlavní paměti
2. Fronta obsahuje tři dekodované instrukce připravené k provedení

Vlastní provádění instrukcí je řízeno mikroprogramem.

Aritmetická jednotka ALU kromě běžně používané sčítačky obsahuje též násobičku, děličku a válcový posouvač.

Průměrná délka instrukce je 32 b, takže do fronty instrukcí dlouhé 16 B se umístí průměrně 4 instrukce.

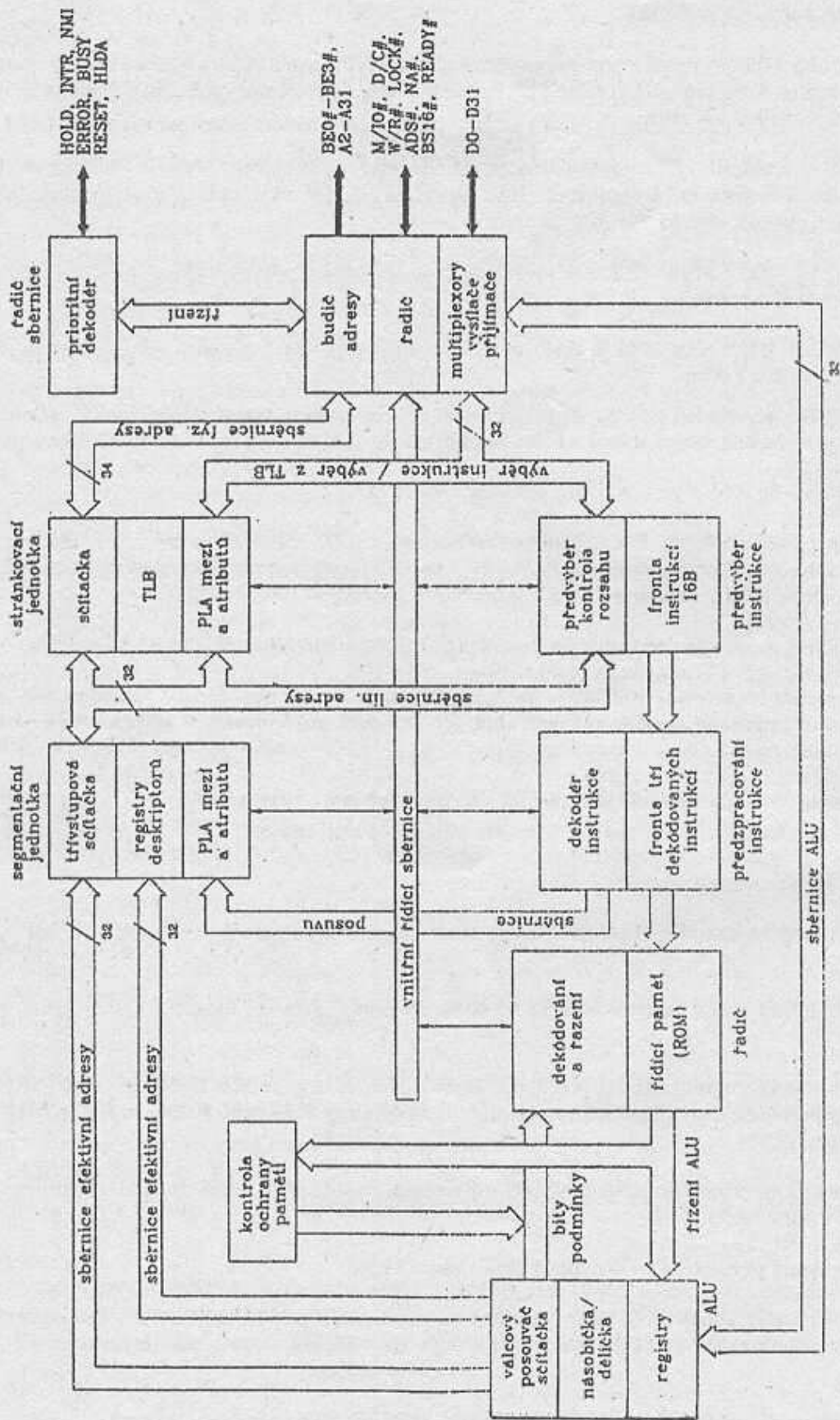
Operandy, s nimiž instrukce pracují, mohou mít délku **8, 16 nebo 32 b**. Délka operandů 16 b se používá především při provádění programů napsaných pro mikroprocesory 8086 nebo 80286, s nimiž je 80386 kompatibilní směrem nahoru.

Na čipu je realizována též **jednotka řízení paměti**, v níž se předkládají virtuální adresy na reálné (viz. obr. č. 5.10 v horní části).

Ochrana paměti je čtyřúrovňová stejně jako u 80286.

Reálné adresy mají 32 b, takže lze přímo adresovat 4 GB paměti, zatímco virtuální adresy při délce 46 bitů umožňují adresovat 64 TB paměti. Paměť je segmentovaná, každý segment může mít maximální velikost 4 GB.

Mikroprocesor 80386 umožňuje simultánní použití několika operačních systémů.



Obr. č. 5.10 Mikroprocesor 80386

Stránkovací jednotka obsažená na čipu mikroprocesoru je určena pro překlad virtuálních adres na reálné. Mikroprocesor používá stránkovaný segmentovaný systém, v němž je při překladu virtuální adresy nejprve vypočtena adresa segmentu a teprve pomocí horních 10 bitů adresy je adresována tabulka zvaná seznam stránek. V této tabulce je nalezena adresa odpovídající tabulce stránek a v této tabulce se vyhledává adresa stránky pomocí dalších 10 nižších bitů adresy. Adresa nalezená v tabulce stránek se pak předřadí posuvu a tím vznikne reálná adresa.

Pro urychlení stránkovacího mechanismu se používá asociativní paměť TLB s 32 adresami a stupněm asociativity $s=4$.

5.6 Mikroprocesor 80486

Mikroprocesor 80486 DX, který byl uveden na trh v roce 1989 [9,19], je kompatibilní směrem nahoru se svým předchůdcem 80386 a navíc, díky vyššímu stupni integrace, je na čipu realizováno více funkčních jednotek než dříve.

Čip vyrobený technologií CHMOS obsahuje cca 1.2 milionu tranzistorů a je umístěn v pouzdře PGA se 168 vývody.

Může pracovat na frekvenci 25, 33, 40 nebo 50 MHz.

Kromě vlastního procesoru a jednotky pro řízení paměti je v modelu 80486 DX integrována i vyrovnávací paměť a aritmetická jednotka pro operace s pohyblivou řádovou čárkou.

Vzhledem k tomu, že instrukce jsou do značné míry prováděny proudově, je tento mikroprocesor schopen provádět některé instrukce v jednom cyklu.

Na obr. č. 5.11 [9] je blokové schéma mikroprocesoru 80486 DX. Datová i adresová sběrnice má šířku 32bitů, aritmetická jednotka a všechny registry mají tutéž šířku. Vzhledem k tomu, že 80486 v sobě spojuje funkce celočíselné aritmetické jednotky a koprocesoru pro pohyblivou řádovou čárku, můžeme v něm najít všechny registry, které používá 80386 a 80387 (mat. koprocesor).

Způsoby adresování paměti a způsoby překladu včetně TLB a ochrany paměti jsou stejné jako v procesoru 80386.

Vyrovnávací paměť na čipu s kapacitou 8 kB je společná pro data i instrukce a může být doplněna vnější vyrovnávací pamětí. Adresář vyrovnávací paměti je částečně asociativní se stupněm asociativity $s=4$.

Vyrovnávací paměť je rozdělena do 128 sad, každá sada obsahuje 4 řádky. Jsou-li všechna místa v určité sadě obsazena, vyřazuje se řádek na základě strategie LRU.

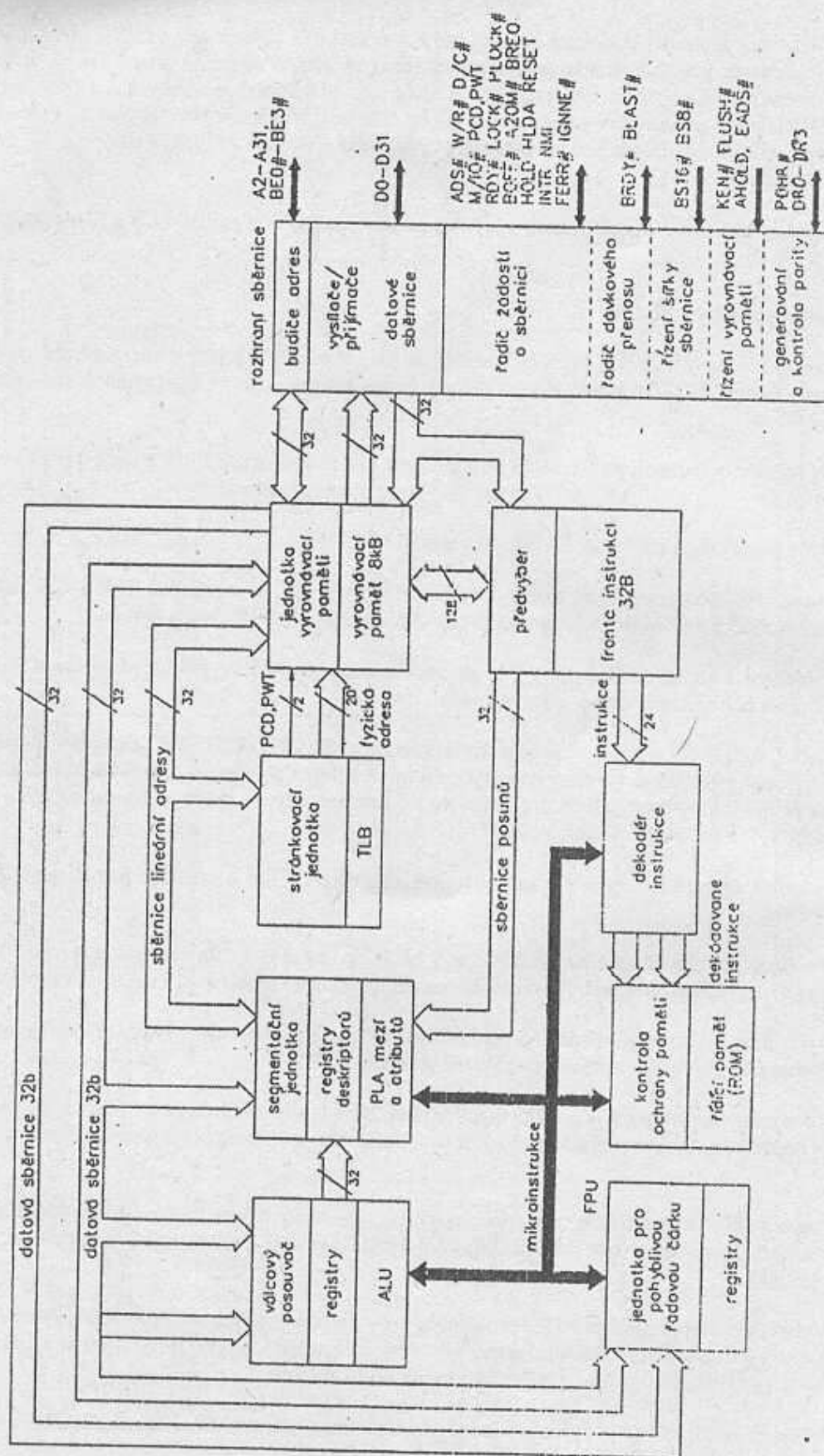
Ve stránkovací jednotce je použita tabulka TLB, která je zdokonalena ve srovnání s procesorem 80386. Algoritmus použitý pro vyřazování adres z TLB je stejný jako alg. vyřazování bloků z paměti cache (vyrovnávací paměť).

Procesor 80486 je vybaven velmi dokonalými diagnostickými prostředky. Jsou to především prostředky pro autonomní diagnostiku (test řídicí ROM, TLB a paměti cache). Pro zbyvající části procesoru se používá vnější diagnostika.

Vybavení mikroprocesoru 80486 pro použití v multiprocesorových systémech se týká operačního kódu. Zde jsou zařazeny instrukce pro multiprocesorový režim, dále prostředků pro zpracování konzistence vyrovnávacích pamětí a prostředků pro připojení paměti cache druhé úrovně.

Mikroprocesor 80486 je určen pro práci s operačními systémy: MS DOS, OS/2, UNIX atd.

Od roku 1991 je nabízena modifikace pod označením 80486 SX. Ta se liší od 80486 DX především tím, že nemá mat. koprocesor. Tento koprocesor je na čipu realizován, avšak není připojen na vývody.



Obr. č. 5.11 Mikroprocesor 80486

V procesorech 80486 se též používá technika zdvojování frekvence označována DX-2 (overdrive). Jedná se o vnitřní zdvojnásobení hodinové frekvence, takže procesory, do kterých je na desce přiváděna frekvence 25 nebo 33 MHz, pracují ve skutečnosti s vnitřní frekvencí 50 nebo 66 MHz.

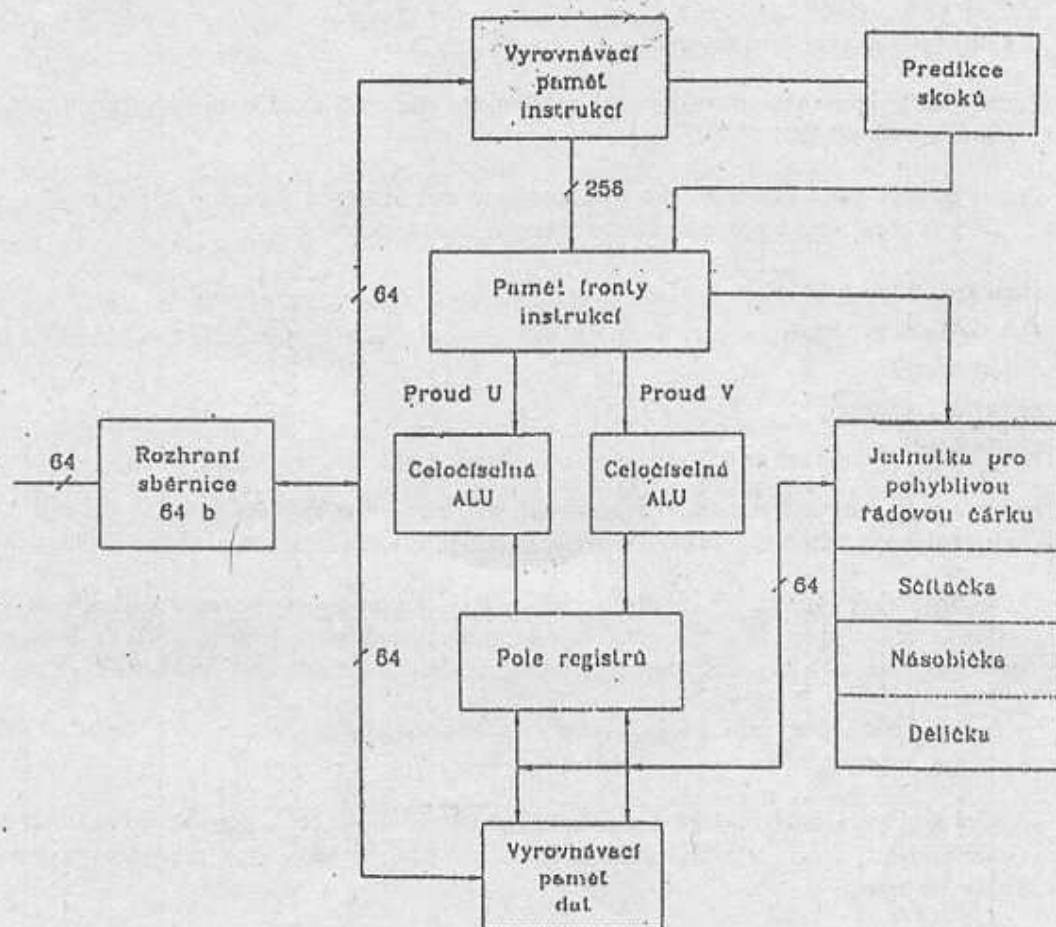
Tím se výkonost systému zvýší nejméně o 50 %.

5.7 Pentium

Od roku 1993 nabízí firma INTEL svůj další mikroprocesor nazvaný Pentium (původně 80586). Jak naznačuje dvojcísle 86 na konci číselného kódu, je tento mikroprocesor kompatibilní směrem nahoru se všemi předcházejícími mikroprocesory, jejichž kód končí číslem 86.

Tento mikroprocesor je jako první navržen kombinací technik CISC a RISC.

Struktura procesoru Pentium je zjednodušeně znázorněna na obr. č. 5.12.



Obr. č. 5.12 Pentium

Procesor Pentium je realizován technologií 0.8 μ m BiCMOS se třemi vrstvami metalizace na čipu o rozměrech 16.6x17.6 mm.

Obsahuje cca 3.1 milionu tranzistorů a při frekvenci hodin 66 MHz má spotřebu 16 W. Více než dvojnásobně převyšuje výkonost zatím nejrychlejšího procesoru INTEL 80486 DX2/66.

K jeho nejdůležitějším vlastnostem patří šířka datové sběrnice 64 b, adresová sběrnice má šířku 32 b jako u předcházejícího modelu.

Šířka toku dat je 64 b a je zachována až ke dvěma odděleným vyrovnávacím pamětem (pro instrukce a pro data), z nichž každá má kapacitu 8 Kb. Každá paměť cache je vybavena adresářem s omezeným stupněm asociativity ($s=2$), přičemž bloky mají velikost 32 B. Pro zajištění konzistence obsahu paměti cache s hlavní pamětí se používá technika úklidu podle příznaku.

Pro zajištění konzistence mezi několika pamětmi cache dat v multiprocessorovém systému se používá protokol MESI (modified, exclusive, shared, invalid), v němž se každá změna obsahu paměti cache dat posílá do paměti cache všech procesorů, které pracují se stejnými daty.

Z paměti cache instrukcí se instrukce přesouvají v šířce 256 b do fronty, jejíž plnění ovlivňuje blok **predikce skoků**. Jeho úkolem je naplnit frontu tak, aby k provedení byly připraveny instrukce, jejichž provedení je nejpravděpodobnější. K tomu se používá paměť cílových adres skoků (branch target buffer-BTB) s kapacitou 256 adres. Každá položka v této paměti má tři části:

- adresu, v níž byla skoková instrukce přečtena
- cílovou adresu skoku
- dva bity popisující historii skoku

Podle nich je jako další instrukce do fronty volena buď instrukce, která následuje v paměti, nebo instrukce z cílové adresy skoku.

Je zde možno číst z fronty během jednoho taktu dvě instrukce současně. Dvě instrukce přečtené současně z fronty se přesouvají k provedení s pěti sekcemi. Jsou to sekce:

- čtení instrukce
- dekódování 1
- dekódování 2
- provedení operace
- zápis výsledku

Během zpracování ve druhé sekci (dekódování) se kontroluje datová závislost mezi instrukcemi, čímž se zaručí, že výsledek výpočtu bude přesně odpovídat pořadí provádění instrukcí předepsané programátorem.

U řady **instrukcí** výrobce opustil dříve používanou mikroprogramovou realizaci, takže se **provádějí v jednom taktu**. U složitějších instrukcí se používá mikroprogramové řízení. Je možné použít současný přístup obou aritmetických jednotek k paměti cache dat (paměť pracuje s osminásobně prokládaným cyklem).

Na čipu je umístěna též aritmetická jednotka pro operace s pohyblivou řádovou čárkou s šířkou toku dat 64 b.

Používá 8 univerzálních registrů a zásobníkovou paměť. Instrukce s pohyblivou řádovou čárkou se zpracovávají proudově v sedmi sekcích, přičemž pouze poslední tři sekce jsou prováděny v jednotce pro pohyblivou řádovou čárku.

Úvodní čtyři fáze zpracování se pro většinu instrukcí s pohyblivou řádovou čárkou provádějí v některé jednotce s pevnou řádovou čárkou a teprve pak je instrukce předána k dalšímu zpracování do jednotky pro pohyblivou řádovou čárku.

Při předávání dat z jednotky s pevnou řádovou čárkou se dva toky dat s šířkou 32 b spojí paralelně a vytvoří tak potřebnou šířku 64 b.

Pro dosažení maximálního výkonu je jednotka s pohyblivou řádovou čárkou vybavena samostatnými bloky, v nichž se operace sčítání, násobení a dělení **provádějí obvodově**.

Operace s pohyblivou řádovou čárkou probíhají 5 až 10x rychleji než u procesoru 486 DX/33.

Pro styk s pamětí se používá virtuální adresování s volitelnou velikostí stránky 4 kB nebo 4 MB. Stránky s větším rozměrem jsou určeny pro grafické systémy a pro jádro operačního systému, kde mají předejít nepřípustně vysoké frekvenci stránkování.

Výrobce věnoval značnou pozornost i zabezpečení systému proti poruchám. V procesoru je použita parita pro kontrolu správnosti funkce všech jednotek a navíc je proces vybaven vnějšími vedeními, která umožňují snadno vytvářet duplexní systémy.

Procesor Pentium je vhodný pro použití pod většinou moderních OS, jako Windows-NT, OS/2, Solaris nebo NEXTstep.

Mikroprocesor PENTIUM s technologií MMX

Procesory [36, 37] jsou založeny na vylepšené 0,35(μm) technologii CMOS, která dovoluje dodat vyšší výkon s nižší spotřebou elektrické energie. Procesor Pentium s technologií MMX obsahuje 4,5 mil. tranzistorů. Kromě posílení o instrukce MMX je obohacen o několik vylepšení vnitřní architektury. Zdvojená vyrovnávací paměť cache má 32 kB, výkonnější je předvídaní větvení instrukcí, které při výkonových testech (benchmark) přináší ve srovnání se standardní centrální procesorovou jednotkou zvýšení výkonu o 10 až 20 %.

Technologie MMX

Společnost Intel vyvinula 57 nových instrukcí, aby zvýšila rychlost některých výpočetně náročných smyček, jež se typicky vyskytují v aplikacích pro zpracování grafiky, zvuku a videa. Technologie MMX si udržuje plnou kompatibilitu s předešlou architekturou procesoru.

Výkon SPEC CPU95 pro 200MHz procesor je 6,41 SPECint95 a 4,66 SPECfp95. Výkon pro 166MHz procesor je 5,59 SPECint95 a 4,30 SPECfp95. Poměry iCOMP (R) Index 2.0 pro tyto procesory pak jsou 182 a 160.

Procesory Pentium s technologií MMX pro stolní počítače s hodinovou frekvencí 166 MHz a 200 MHz se dodávají zapouzdřené v keramice nebo plastu s jehlovými vývody (Plastic Pin Grid Array - PPGA). Typy pro mobilní počítače, s hodinovými frekvencemi 150 MHz a 166 MHz, se pouzdří a dodávají na filmovém pásu (Tape Carrier Packaging) nebo v PPGA.

Tabulka č. 5.1

ČIP	Rychlost (Mhz)	Rychlost sběrnice	L1 Cache Total	L2 Cache	Mikronů	Tranzistorů (mil.)	Soket	Dosažitelnost
Pentium s MMX	133	66	32 kB	-	.35	3.3	Soket 7	nyní
	150	66	32 kB	-	.35	3.3	Soket 7	nyní
	166	66	32 kB	-	.35	3.3	Soket 7	nyní
	200	66	32 kB	-	.35	3.3	Soket 7	nyní
	233	66	32 kB	-	.35	3.3	Soket 7	nyní
Tillamook (low-power Pentium s MMX)	200	66	32 kB	-	.25	3.3	MMO module	nyní
	233	66	32 kB	-	.25	3.3	MMO module	nyní
	266	66	32 kB	-	.25	3.3	MMO module	nyní

CN430TX MOTHERBOARD

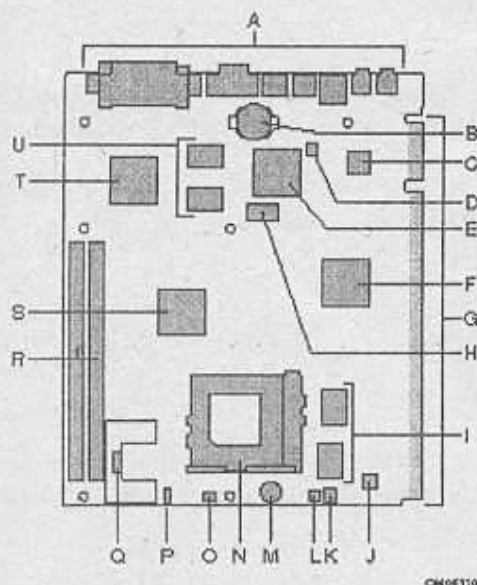
Základní deska CN430TX podporuje procesory Pentium, Pentium Over Drive a Pentium s technologií MMX. Charakteristika základní desky zahrnuje následující:

- velikost odpovídá NLX rámu
- patice na procesor je typ 7 ZIF s 321 piny
- hlavní paměť
 - dva 168 pinové DIMM sokety
 - podpora až 256 MB DRAM
- Second Level Cache paměť
- 256 nebo 512 kB Pipeline Burst SRAM napájena na desce
- Chipset a PCI/IDE rozhraní
 - Intel 82430TX PCIset
 - integrovaný řadič ovládaní PCI sběrnice
- I/O vlastnosti
 - National polovodičový PC87307 Super I/O řadič
 - integruje následující I/O funkce: rozhraní pro FD,
jeden vícemodový paralelní port
dva FIFO seriové porty,
hodiny
řadič klávesnice a myši
- podpora pro rozhraní dvou universálních seriových sběrnic (USB)
- Rozšiřující sloty
 - podpora až na tři PCI a dvě ISA
- Zvukový subsystém
 - Yamaha OPL3-S A3 audio codec
- Grafický subsystém
 - S3 Virge/GX
- Další vlastnosti
 - Plug & Play kompatibilita
 - podpora vylepšené správy napájení (APM)
 - podpora vylepšené konfigurace a rozhraní napájení (ACPI)
 - monitorování hardware (ASIC)
- Softwarové ovladače a utility lze zakoupit od firmy Intel.

Základní deska podporuje následující:

- Pentium s MMX technologií pracující na 166, 200 a 233MHz
- Pentia pracující na 90, 100, 120, 133, 150, 166 a 200MHz

Zabudovaný regulátor napětí omezuje napětí ze zdroje na potřebné napětí. Pro Pentia se musí určit jumperem použití VR- nebo VRE- procesoru. U Pentii s MMX je potřebné napětí automaticky detekováno a není nutné přepínat žádné jumpery.



- A - I/O konektory na zadním panelu
- B - baterie
- C - Audio
- D - jumper na frekvenci sběrnice a obnovování BIOSu
- E - I/O řadič
- F - PIIX4 řadič (PCI ISA IDE)
- G - rozšiřující konektor
- H - 2Mbity přepisovatelné BIOS
- I - cache
- J - sledování hardware ASIC
- K - přípojka na větrák
- L - rámová bezpečnostní přípojka
- M - reproduktor
- N - patice na procesor
- O - jumper na hodiny

Obr. č. 5.13 Motherboard Pentia

Paměť

Hlavní paměť:

Základní deska má dva DIMM sokety (Duální přímozapojitelný paměťový modul). Paměť může být instalována v jednom nebo v obou soketech. Minimálně 8 a maximálně 256MB. Nejsou nutné jumpery, protože BIOS automaticky detekuje typ, velikost a rychlost paměti.

Podporováno je následující:

- 168 pinové DIMM s pozlacenými kontakty a napájením 3.3V
- 60 a 66MHz sběrnice
- nebufferovaná EDO paměť s rychlostí 60ns a napájením 3.3V
- nebufferovaná SDRAM také s rychlostí 60ns a napájením 3.3V, navíc je 4-clock a CAS Latency 2
- cachování pro prvních 64 MB hlavní paměti
- 64 bitová datová cesta
- jedno nebo dvoustranné DIMM velikosti 8, 16, 32, 64 a 128, a to jak EDO, tak CAS Latency 2 SDRAM.

V každém soketu může být jiný typ paměti a nedoporučují se paměti s paritou, protože deska nepodporuje ECC.

EDO DRAM

Zvyšuje čtecí výkon tím, že udržuje data platná do další CAS sestupné hrany. S paměti EDO a CAS přednastavení přesáhne platný čas, který dovolí CAS negovat dříve, zatímco je stále v souladu s oknem platnosti paměťových dat. DIMM musí být podle intelovské specifikace v 1.2

SDRAM (synchronní DRAM)

Zvyšuje celkový paměťový výkon. Oproti EDO je synchronizována s paměťovými hodinami. Ulehčuje to časování a zvyšuje rychlost, protože všechno časování je závislé na počtu paměťových hodinových cyklů. Aby správně fungovala, měla by paměť mít Intelovskou specifikaci v 1.0, která má 4 hodiny, 66MHz a je 64 bitová nebufferována.

Druhoúladinová cache

Má 512KB, je přímo mapovatelná, zpětně prepisovatelná. L2 cache se sestává ze dvou 64Kbitových x 32 PDSRAM povolujících globální zápis (GWE) zařízení napájených na desce. 32Kbitová x 8 vnější SRAM zajišťuje cachování až pro 64MB hlavní paměti.

256KB L2 má jen 32Kbit x 32 PDSRAM a 8Kbit x 8 vnější SRAM

Chipset

se skládá z TX systémového řadiče (MTXC) a PIIX4 zařízení

MTXC

Spojuje v sobě kontrolní funkce cache a hlavní paměti a zajišťuje řízení sběrnice, obsluhu přenosu mezi procesorem, cache, hlavní pamětí a PCI sběrnici. Pro zvýšený výkon systému v sobě spojuje posilací zapisovací a čtecí buffery.

PIIX4

Multifunkční PCI zařízení implementující most mezi PCI a ISA, funkčnost PCI IDE, funkce host/hub USB (universální sériové sběrnice) a vylepšenou správu napájení.

USB

Deska má dva USB porty. Můžeme do nich zapojit buď dvě USB periférie přímo a nebo do jednoho dát externí osu konektoru a USB periférie zapojovat až na ni. Deska plně podporuje standardní universální host řídící rozhraní UHCI a používá standardní softwarové ovladače s ním kompatibilní.

Podpora IDE

Deska má dva nezávislé, sběrnice ovládané PCI IDE rozhraní, které podporuje zařízení PIO3, PIO4, ATA 33 a ATAPI. BIOS podporuje LBA a ECHS překladové módy. Přenosová rychlost a mód IDE zařízení je automaticky detekována BIOSem.

LS-120

LS-120 disketová technologie, která dovoluje uživateli uložit 120MB dat na jedinou 3.5 palcovou disketu, ale odlišnou od klasické diskety. Tato technologie je také zpětně kompatibilní s klasickými formáty disket. Základní deska umožňuje zapojit zařízení kompatibilní s LS-120 a zároveň normální disketovou jednotku. LS-120 lze dokonce v BIOSu nastavit jako bootovatelné.

Super I/O Controller

Tento řadič je ISA plug and play kompatibilní, multifunkční I/O (vstupně/výstupní) zařízení s charakteristikami:

- a) sériové porty
 - dva, UART kompatibilní
 - posílají/přijímají 16 Bytů FIFO
 - čtyři 8bitové DMA volby pro UART s podporou pomalých infračervených zařízení (USI)
 - identifikátor Ring pro oba sériové porty
- b) multimódový, dvousměrový paralelní port
 - Standardní mód, kompatibilní s IBM a Centronics
 - vylepšený paralelní mód (EPP) s podporou BIOSu a ovladačů
 - mód vysoké rychlosti a rozšířených schopností portu (ECP)

- c) řadič disketové jednotky
 - kompatibilní s výrobním standardem 8042A
 - mikrořadič hlavního záměru
 - vnitřní 8bitová datová sběrnice
- d) podpora IrDA a uživatelského infračerveného zařízení

Rozhraní I/O řadiče je automaticky konfigurováno během bootu, ale můžeme ho ručně překonfigurovat v Setupu.

Grafický subsystém

Na desce je S3 Virge/GX grafická karta s vlastnostmi:

- schopnost 3D grafiky
- chip založen na SGRAM, optimalizovaný pro synchronní operace
- 64 bitový grafický engine s urychlovačem jako jádrem
- 170MHz 24 bitový RAMDAC/hodinový syntetizér
- duální programovatelné hodinové generátory
- lineární adresovací schéma založené na DCI
- S3 procesor specializovaný na proudy zaručuje vysokou rychlost přehrávání videa

Zvukový subsystém

Zvuková karta je součástí desky a je to Yamaha OPL3-S A3. Zahrnuje 16 bitový zvukový kóder, OPL3 FM syntézu a DAC, řadič 3D vylepšeného sterea a rozhraní pro MPU-401 a joystick. Chip zaručuje všechny analogové a digitální mixážní funkce potřebné pro záznam a přehrávání zvuku.

Přehled vlastností BIOSu

Základní deska používá Phoenix BIOS, který je uchovávan v přepisovatelné paměti a může být upgradován pomocí jistého programu. Ve flash memory je kromě BIOSu ještě SETUP program, samoprováděcí test při zapnutí počítače (POST), vylepšená správa napájení (APM), utilita pro autokonfiguraci PCI, Plug and Play připravené pro Windows 95 a podpora pro S0, S1 a S5 stavy, zlepšené konfigurace a rozhraní napájení (ACPI). Deska podporuje BIOS shadowing, což znamená nahraní BIOSu do paměti, aby se mohl spouštět rychleji.

Organizace flash paměti

Paměť je od Intelu a má 2Mbity organizované 256KB x 8 bitu a je rozdělena do úseků:

- část, ve které je boot blok
- configurační data
- použité BIOSem
- logo nebo prostor pro flash scanování
- hlavní blok BIOSu

BIOS Setup

Setup je určen pro prohlídnutí a změny nastavení BIOSu. Dostat se do něj můžeme, když zmáčkne F2 poté, co začne POST a před tím, než se začne bootovat operační systém.

Tabulka č. 5.2

Setup Menu Screen	Popis
Maintenance (údržba)	ruší hesla Setupu, přístup jen z módu Configure
Main (hlavní)	zajišťuje a rozděluje zdroje pro hardware
Advanced (rozšířená)	specifikuje rozšířené vlastnosti chipsetu
Security (bezpečnostní)	specifikuje hesla a bezpečnostní vlastnosti
Power (napájení)	vlastnosti správy napájení
Boot (zavádění OS)	možnosti startování a dodávky napájení
Exit (ukončení)	ukládá nebo ruší všechny změny

Přehled obrazovek v Setupu

Setup inicializačně zobrazí Main menu. V každé obrazovce jsou možnosti, jak měnit konfiguraci počítače. Obrazovka se vybírá šipkami vlevo a vpravo. Šipky nahoru a dolů vybírají položku na obrazovce. Enter vybírá submenu. Po výběru položky se pomocí + a - mění její hodnota.

Setup submenu:

Floppy options	- konfigurace disketových jednotek
Primary IDE Master	- konfiguruje primární nadřazené IDE zařízení
Primary IDE Slave	- konfiguruje primární podřazené IDE zařízení
Secondary IDE Master	- konfiguruje sekundární nadřazené IDE zařízení
Secondary IDE Slave	- konfiguruje sekundární podřazené IDE zařízení
Resource configuration	- rezervuje bloky paměti a určité přerušení
Peripheral configuration	- konfiguruje periferní zařízení
Keyboard configuration	- konfiguruje funkce klávesnice
Video configuration	- konfiguruje grafickou kartu na desce
DMI event logging	- ukazuje a modifikuje desktop řídicí rozhraní (DMI) login (souboru) s událostmi.
Hard drive	- vybírá jednotlivé hardisky
Removable devices	- vybírá odstranitelné zařízení

5.8 Procesor PentiumPro

Když bylo dokončeno v Kalifornii Pentium nazývajícím se tehdy P5, měl už další vývojový tým v Portlandu ve státě Oregon hotov procesor P6 nazvaný PentiumPro [23, 19, 36, 37].

P6 je podobně jako jeho předchůdci kompatibilní směrem dolů. 686 rozvíjí dále to, na čem je založeno dnešní Pentium. Je zhotoven technologií 0,6 mikrometru a dosahuje 200 MHz pracovní frekvence. Nyní závod na výrobu čipů s 0,4 mikrometrovou strukturou, která je pro druhou generaci procesorů PentiumPro.

PentiumPro přináší více než dvojnásobný výkon oproti nejrychlejšímu Pentiu. Takový nárůst umožňuje nový design procesoru, který vykazuje **čtyři pipelines na paralelní zpracování příkazů a integrovanou primární i sekundární cache (mezipaměť)**. Kromě toho zavedl Intel **nový sběrníkový systém pro PC**, který byl doposud v podobné formě běžný pouze u velkých počítačů. V jednom počítači mohou lehce **komunikovat až čtyři procesory PentiumPro** - logika multiprocessingu je nasazena do každého čipu.

Vnitřní stavba nového procesoru se skládá ze dvou oddělených čipů. Na jedné ze dvou křemíkových destiček je umístěno vlastní jádro procesoru a primární procesorová cache (L1-cache). Na druhé destičce je umístěna sekundární cache (L2-cache).

Mezipaměť, která doposud u rychlých PC byla umístěna vedle procesoru na základní desce, je nyní přímo integrována v procesoru. Intel tak učinil rozhodující krok a dostal se dál než Nexgen. Tento výrobce sice

představil na CeBITu '94 procesor Nx586 v jednom pouzdře s integrovaným řadičem cache, vlastní paměť je ale ještě umístěna vně. Aritmetický koprocesor je u Intelu také integrován na čipu.

Na obou křemíkových destičkách je integrováno 6 milionů tranzistorů.

Jádro procesoru **PentiumPro** se zásadně odlišuje od jádra Pentia. V Pentiu se začalo s nahrazováním komplexních příkazů jednoduššími instrukcemi RISC (Reduced Instruction Set Computer), počítače s redukováným instrukčním souborem. Tyto jsou rychleji zpracovávány ve dvou paralelních pipelines. V **PentiumPro** jsou nasazeny čtyři takové paralelní pipelines.

Sběrnice připomíná systém, který se (přibližně před deseti lety) objevil ve světě velkých počítačů jako "Future bus".

Elektrické připojení procesoru se už neprovádí přes kontakty se třemi logickými stavy (high, low, velký odpor), nýbrž přes volné kolektory, tzv. GTL buffery, kterými se signální napětí potřebné k přenosu dat krátkodobě stáhne. Tím se umožní provoz sběrnice na relativně vysokém pracovním kmitočtu 75 MHz. Interním zdvojnásobením frekvence, které bylo zavedeno Intelu už u čipů s označením KX2, dosáhne P6 až 150 MHz.

Napětíový rozdíl mezi logickým stavem "0" a "1" obnáší pouhých 1,5 V. Význam ochrany proti rušení v počítači tak roste.

Společně s procesorem **PentiumPro** se objevují nové speciální komponenty, jako např. řadič sběrnice a další čipy. PCI bude představovat most - spojení s dnešními PC. Na PCI Bus mohou být napojena další dnešní rozhraní - periferie ISA a EISA.

Pracovní paměť (RAM) může být zavěšena přes řadič paměti a čtyři multiplexory RAM přímo na procesorové sběrnici. Toto řešení je výhodné hlavně pro velké databankové servery. Pokud je obsazeno všech 16 slotů, pak není třeba vkládat při 75 MHz žádné čekací doby (waitstates) pro paměť. Pro stolní systémy jsou žádoucí jednodušší řešení bez multiplexoru RAM. Adresy budou přenášeny zásadně na šířce 36 bitů, pro data je k dispozici 64-bitová sběrnice. Na rozpoznávání chyb je nasazen kód ECC.

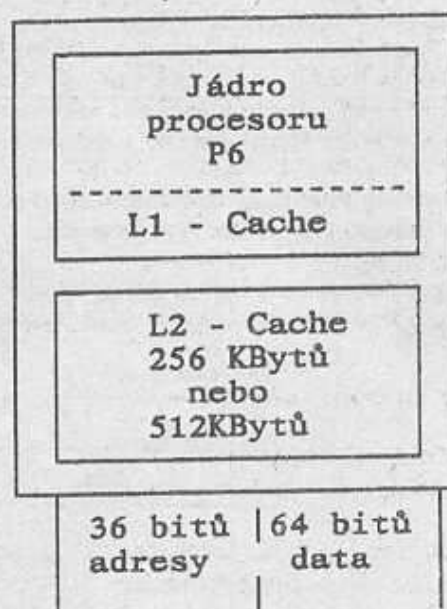
Při paralelním nasazení několika procesorů vyniká další přednost nové sběrnice architektury. Po žádosti o data daný procesor ihned uvolní sběrnici. Zablokuje si ji znovu až tehdy, když jsou rozpoznána data skutečně k dispozici. Při postupech označovaných jako "Split-Cycle Bus" nemusí být data dodávána v požadovaném sledu. To ulehčuje spojení s pomalejšími komponentami, např. ISA kartami, které pak už nebrzdí celý systém.

Snahou Intelu je ještě snížit pracovní napětí procesoru pod hranici nedávno dosažené hodnoty 3,3 V. V současné době se experimentuje s 2,8 V, nicméně tato hranice by se měla ještě snížit.

Při pracovní frekvenci 150 MHz dosahuje **PentiumPro** ztrátového výkonu 28 W. Aby bylo možno pracovat s proudem až 10 A, má u multiprocesorových systémů na základní desce každá CPU vlastní napětíový měnič. I zde je snahou Intelu během vývoje snížit ztrátový výkon pod 20 W.

Snahou Intelu je dosažení početního výkonu 250 až 300 MIPS (milióny instrukcí za sekundu). Multiprocesorový systém se čtyřmi procesory **PentiumPro** má dosáhnout až 1000 MIPS.

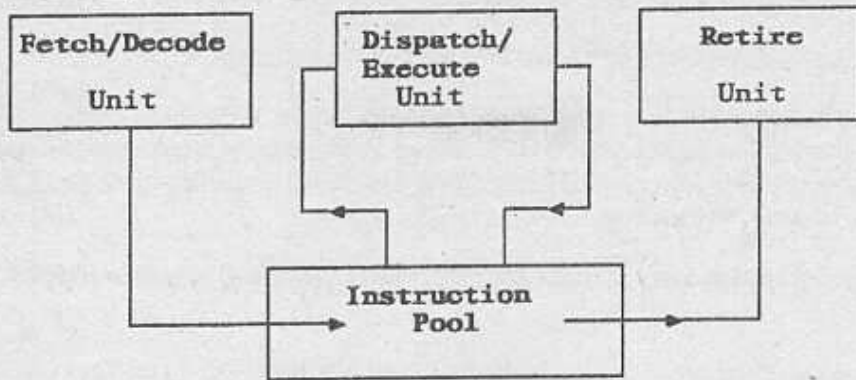
Stavba procesoru PentiumPro:



Obr. č. 5.14

Dva čipy v jednom procesorovém pouzdře: second level cache je v P6 takto integrována

Nyní se zaměříme na popis architektury procesoru PentiumPro. Na obr. č. 5.15 je uvedeno hrubé blokové schéma architektury.



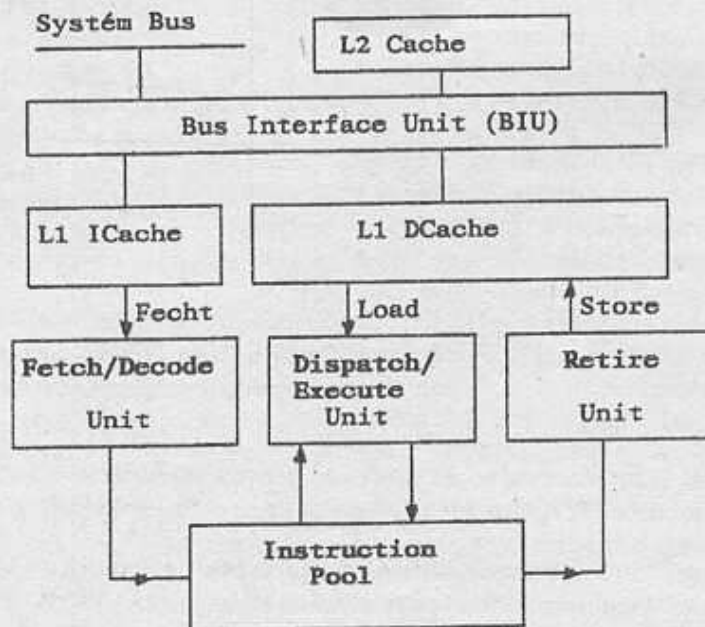
Obr. č. 5.15

- Fetch/Decode Unit** - jednotka na volání instrukcí programu a jejich dekódování
- Dispatch/Execute Unit** - jednotka na přidělování mikrokódu a provedení instrukcí
- RetireP Unit** - návratová jednotka dat
- Instruction Pool** - banka dekódovaných instrukcí

Z uvedeného blokového schématu je zřejmé, že struktura procesoru zahrnuje několik jednotek (Unit). V jednotce **Fetch/Decode** jsou postupně načítány a dekódovány instrukce programu, které pak jsou ukládány v tzv. **Instruction Pool**, tj. v bance dekódovaných instrukcí. V jednotce **Dispatch/Execute** jsou k dekódovaným instrukcím přiřazeny hodnoty operandů a následně provedeny. Po provedení jsou opět vráceny zpět do banky, kde později z jednotky **Retire** jsou nová data vrácena zpět do paměti.

Abychom lépe porozuměli činnosti procesoru, ukážeme si v následující části podrobnější schéma architektury procesoru a jednotlivých jednotek.

Na obr. č. 5.16 je uvedeno podrobnější schéma architektury procesoru. Kromě základní architektury je zde také uvedena návaznost procesoru na jednotlivé paměti cache (L1,L2).



- L1 a L2 - vyrovnávací paměti Cache, ICache - instrukční vyrovnávací paměť
- DCache - datová vyrovnávací paměť, BIU - sběrníková interface jednotka

Obr. č. 5.16

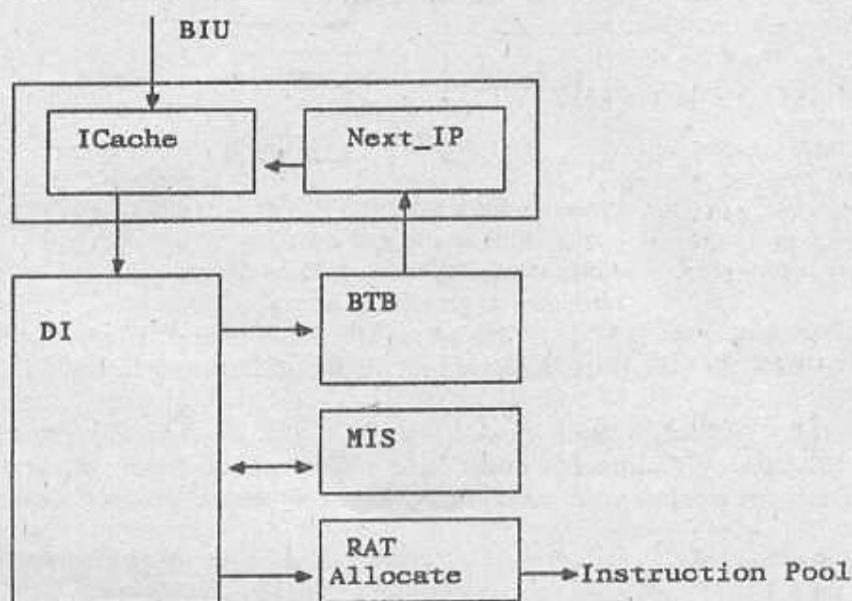
Ze schématu je patrné, že BIU (Sběrniceová interface jednotka) zabezpečuje přenos informací ze systémové sběrnice a L2 Cache (sekundární Cache) na straně jedné a L2 Cache (ICache a DCache) na straně druhé. Tento přenos informací je obousměrný. Je důležité poznamenat, že L2 Cache je rozdělena do dvou částí. Část ICache je používána jako vyrovnávací paměť pro instrukce a část DCache jako vyrovnávací paměť pro data.

Z paměti ICache jsou postupně volány, tj. načítány, jednotlivé instrukce programu do jednotky **Fetch/Decode**, kde jsou dekodovány a následně uloženy do banky **Instruction Pool**. Jednotka **Dispatch/Execute** postupně načítá dekodované instrukce z banky. V této jednotce jsou k načteným dekodovaným instrukcím přidávány operandy z paměti DCache.

Po kompletaci dekodovaných instrukcí je zde plánováno jejich provedení. Výsledkem činnosti této jednotky je provedení instrukce a její zpětné uložení do banky. Pomocí jednotky **Retire** jsou pak nová data vrácena zpět do DCache.

1. Popis jednotky **Fetch/Decode**

1. Popis jednotky **Fetch/Decode**



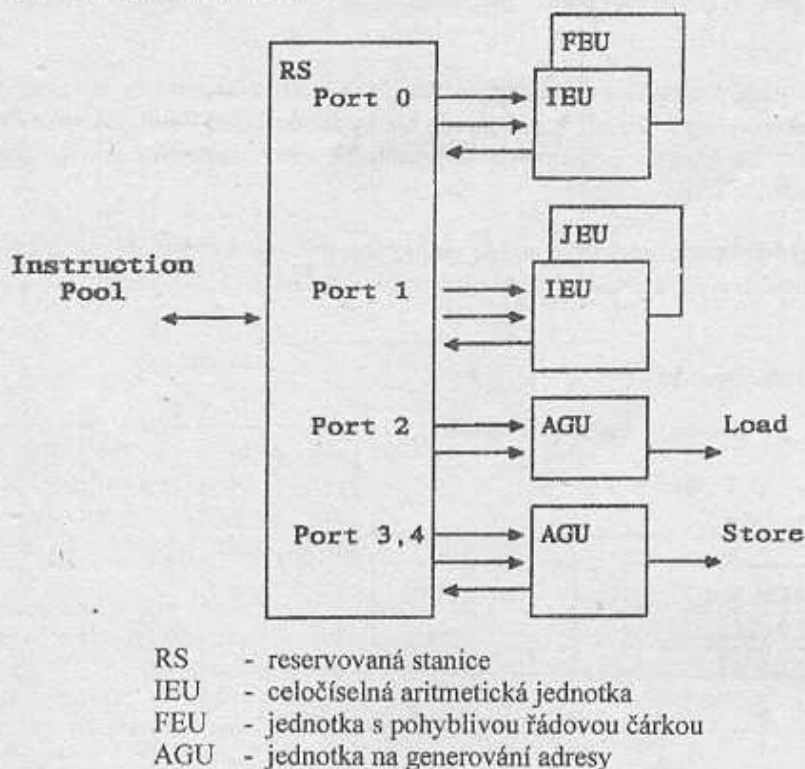
DI	- dekodér instrukcí
BIU	- Bus interface jednotka
ICache	- instrukční vyrovnávací paměť
Next_IP	- následující instrukce
BTB	- větvení v cílovém programu
MIS	- mikrokód instrukce
RAT Allocation	- registr (tabulka)

Obr. č. 5.17 **Fetch/Decode Unit** (Volání instrukce programu/Decode Unit).

Struktura jednotky **Fetch/Decode** je uvedena na obr. č. 5.17. Ze schématu vidíme, že instrukce programu uložené v ICache jsou postupně načítány a v bloku DI je prováděno jejich dekodování. Po provedení dekodování instrukce se v bloku BTB (větvení v cílovém programu) provádí její vyhodnocení za účelem stanovení větvení programu, tj. určení adresy následující instrukce. Po tomto vyhodnocení je předána informace z bloku BTB do bloku Next_IP, kde je určena adresa následující instrukce. Při dekodování je ke každé instrukci přidělena jedna mikroinstrukce. Jsou i případy, kdy na jednu instrukci jsou přiděleny až čtyři mikroinstrukce. V tomto bloku je pak provedena náhrada do tzv. předprogramovaných mikrokódů instrukcí.

V bloku RAT Allocation jsou seřazeny dekódované instrukce, které jsou předány do banky dekódovaných instrukcí.

2. Popis jednotky Dispatch/Execute

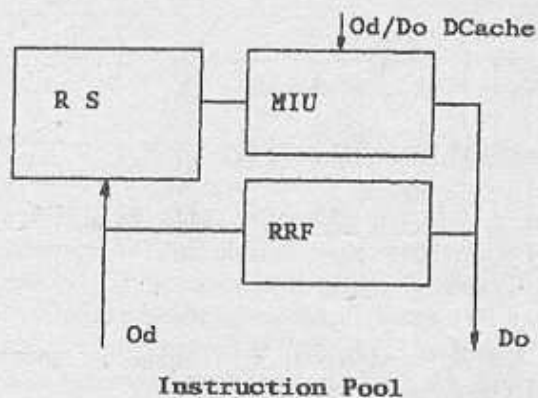


Obr. č. 5.18 Dispatch/ Execute Unit (Přidělování/Exekuce Unit).

Struktura jednotky Dispatch/ Execute je uvedena na obr. č. 5.18. Ze schématu je patrné, že dekódované instrukce programu, nacházející se v Instruction Pool, (Banka dekódovaných instrukcí), jsou postupně načítány a ukládány do tzv. RS. Odtud jsou prostřednictvím jednotlivých portů (0 až 4) přenášeny k následnému zpracování takto:

- Port 0 - provedení aritmetických operací celočíselných nebo s pohyblivou řádovou čárkou
- Port 1 - provedení aritmetických operací celočíselných nebo výpočet skoku
- Port 2 - generování čtecí adresy
- Port 3,4 - generování adresy pro zápis

3. Popis jednotky Retire

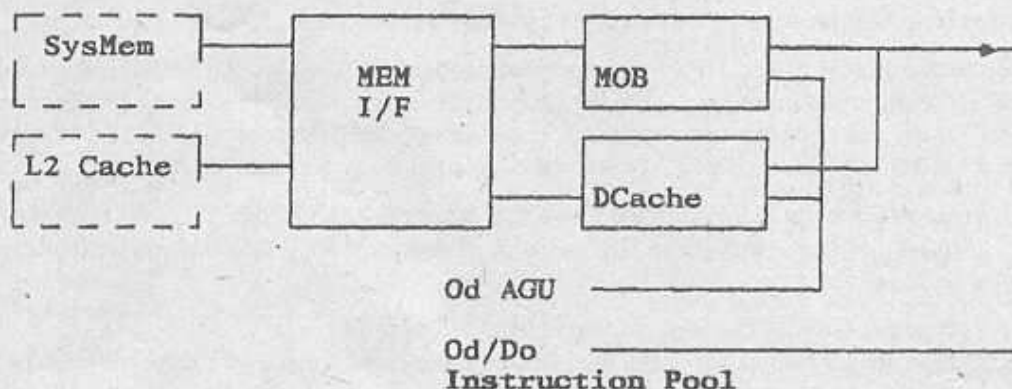


- RS - rezervovaná stanice
- MIU - paměťová interface jednotka
- RRF - soubor návratových registrů

Obr. č. 5.19 Retire Unit (Návratová jednotka dat).

Struktura jednotky **Retire** je uvedena na obr. č. 5.19. Ze schématu je vidět, jak data vzniklá při výpočtu v unitu **Dispatch/Execute** a pak uložená v **Instruction Pool**, jsou odkládána v paměti DCache pomocí unitu **Retire**. V bloku RFF se musí nejdříve identifikovat banka a pak stanovit, komu vrátit data.

4. Popis BUS interface jednotky - BIU



- MOB - seřazená paměť instrukcí
- AGU - jednotka na generování adresy
- MEM I/F - paměť
- SysMem - paměť systému
- L2 Cache - sekundární vyrovnávací paměť

Obr. č. 5.20 BUS interface jednotka

Struktura jednotky **BUS interface** - BIU je uvedena na obr. č. 5.20. Ze schématu je vidět, že tato jednotka zabezpečuje přenos informací z paměti systému a L2 Cache (Sekundární Cache) na jedné straně a L1 Cache (MOB a DCache) na straně druhé. Od L1 Cache (L1 ICache a L1 DCache) je pak vazba na jednotlivé unity procesoru.

Tabulka č. 5.3 Charakteristika procesoru Pentium Pro

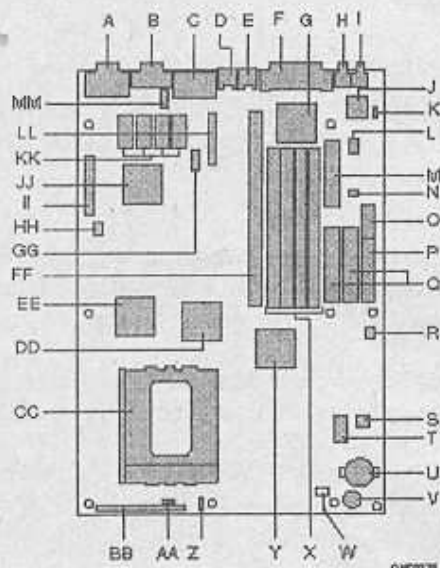
ČIP	Rychlost (MHz)	Rychlost sběrnice	L1 Cache Total	L2 Cache	mikronů	Tranzistorů (miliónů)	Soket	Dosažitelnost
Pentium Pro	150	60	16kB (8kB code, 8kB auto)	256 kB	.35	5.5	P6	nevyrábí se
	166	66	16	256 kB	.35	5.5	P6	nevyrábí se
	166	66	16	512 kB	.35	5.5	P6	nyní
	180	60	16	256 kB	.35	5.5	P6	nyní
	200	66	16	256 kB	.35	5.5	P6	nyní
	200	66	16	512 kB	.35	5.5	P6	nyní
	200	66	16	1 MB	.35	5.5	P6	nyní

AP440FX Motherboard - základní deska pro Pentium Pro

AP440FX je 64 bitová základní deska o velkém výkonu, kombinovaném napájení, energeticky sebevědomá, vysoce integrovaná platforma.

Charakteristika:

- umístění ve standardním 9x13"LPX rámu
- patice na procesor je typ 8 ZIF (nulová zasunovací síla) a poskytuje možnost vylepšení na budoucí OverDrive procesory
- do patice lze použít procesory Pentium Pro s taktovací frekvencí 150, 166, 180 a 200 MHz
- podporuje až 128 MB paměti DRAM zasazené ve standardních 72 pinových SIMM soketech (konektorech). Moduly mohou používat FPM (mód rychlého stránkování) nebo EDO (rozšířený datový výstup) paměť. Jsou podporovány neparitní 32 bitové i paritní 36 bitové SIMMy. S paritními může být deska nakonfigurována na ECC (provedení chyb a jejich oprava) paměťové operace.
- používá chipset (čipovou sadu) od Intelu 82 440FX. Intelovský 82 371SB PCI/ISA Xccelerátor neboli PIIX3 představuje integrovaný řadič IDE sběrnice se dvěma vysoce výkonnými IDE rozhraními až pro 4 zařízení, jako jsou např. harddisky a CD ROMy.
- používá přepisovatelný BIOS s následujícími charakteristikami:
 - a) na ochranu přepisovatelného obsahu před zničením používá hardwarové i softwarové bezpečnostní Flash prostředky
 - b) používá BIOS, který má Desktop Management rozhraní (ovládání z desktopu neboli přímo uživatelem při startu počítače)
- používá I/O řadič, buď PC 87 307 nebo kompatibilní PC 87 308, který v sobě spojuje následující funkce:
 - a) Floppy rozhraní, dva FIFO sériové porty a jeden EPP/ECP paralelní port
 - b) hodiny reálného času
 - c) řadič klávesnice
 - d) podpora IrDA a uživatelského infračerveného zařízení pro malé i střední rychlosti
- má zabudován Crystal zvukový kódovač CS 42 36 zajišťující 16 bitové stereo, kompatibilní se Sound Blasterem Pro. Na desce má taky telefonní konektor k podpoře pozdějších aplikací, využívajících telefon.
- má zabudován S3 Virge/DX grafický řadič k podpoře Super VGA grafiky v rozlišení 1600x1200. Vylepšený LBP konektor s VESA vlastnostmi podporuje vnější multimediální schopnosti.
- hardwarové sledování ASIC zajišťuje následující sledovací funkce:
 - a) zabudovaný teplotní senzor
 - b) sledování rychlosti větráku
 - c) sledování zdroje
 - d) uložení výsledků testů POST (selftestu při zapnutém power) a chybových kódů
- PCI a ISA rozšiřující sloty jsou podporovány konektorem na základní desce, který je navrhnut pro přijetí vyšších karet. Zabudovaný přepínač podporuje vyšší karty s dvěma nebo třemi PCI sloty.



- AA - header keylock (diody na klávesnici, numlock, capslock, scrollock)
- BB - konektor předního panelu
- CC - patice procesoru
- DD - první část čipsetu
- EE - druhá část čipsetu
- FF - soket na přídavné ISA/PCI sloty
- GG - zákaznický header rychle IR
- HH - přepínače ovládající PCI sloty
- II - header LBP VESA vstupu
- JJ - řadič grafické karty
- KK - 2MB video DRAM
- LL - blok přepínačů konfigurace
- MM - header pro dosažení druhého sériového konektoru na COM2

POPIS:

- A - VGA konektor (do monitoru)
- B - sériový port
- C - USB konektory
- D - myš
- E - klávesnice
- F - paralelní port
- G - I/O řadič
- H - mikrofonový vstup (typ jack)
- I - audio výstup (jack)
- J - řadič zvukovky
- K - header CDR0Mky (headery se používají na spojení částí desky, které nejdou dát na plošný spoj)
- L - header Wavetable (část zvukovky, kde se tvoří zvuky)
- M - konektor na disketovku
- N - header telefonního spojení
- O - napájecí konektor 3.3V
- P - konektor hlavního napájení
- Q - IDE konektory
- R - header pro nastavení reakcí zdroje
- S - sledovač hardware ASIC
- T - přepisovatelný BIOS
- U - baterie hodin reálného času
- V - zabudovaný speaker (malý reproduktor)
- W - přepínač na procesory s napájením 3.5V
- X - SIMM s okety na paměť RAM
- Y - řadič IDE

Obr. č. 5.21

MIKROPROCESOR

Základní deska pracuje s procesory Pentium Pro od 2.1 do 3.5 V. Zabudovaný okruh regulátoru napětí zajišťuje požadované napětí z 5 a 3.3 voltových výstupu ze zdroje. Regulátor má schopnost komunikace s VID, což znamená, že automaticky upraví výstupní napětí na napětí požadované instalovaným procesorem. Podporovány jsou Pentia Pro běžící na frekvencích 150, 166, 180 a 200 MHz.

Procesory Pentium Pro v sobě obsahují L2 (druhou hladinovou) cache a řízení cache, což bylo dříve součástí základní desky. Vnitřní (neblokující) L2 cache v procesorech 150, 180 a 200 MHz je 256 kB, zatímco druhá verze 166 a 200 MHz procesoru má 512 kB cache. Pentium Pro má rozšířený číselný koprocesor, který významně zvyšuje rychlost operací s plovoucí desetinnou čárkou a zároveň si uchovává zpětnou kompatibilitu s matematickými koprocesory, které vyhovují ANSI/IEE standardu 754 - 1985.

Poznámka: 200 MHz Pentium Pro s 512 kB cache není touthle deskou podporován.

VÝMĚNA PROCESORU

Patice 8 je 387 pinová, SPGA s ZIF (modifikované rovnoměrně rozložené pinové mřížkové pole s nulovou zasunovací silou) s oketem, spolu s programovatelným regulátorem napětí pro jádro mikroprocesoru zajišťuje uživateli snadnou cestu, jak vyměnit procesor za lepší. Programovatelný regulátor napětí je automatický a je řízený VID piny procesoru.

Chladič a úchytky

Schválený chladič k Pentiu Pro je nutný ke správnému tepelnému rozptylu v LPX rámech. Chladič musí být bezpečně upevněn na patici dvěma úchytkami. Tyto úchytky zachytí chladič a připevní ho za venkovní široké mezery patice Socket 8.

HLAVNÍ SYSTÉMOVÁ PAMĚŤ

Základní deska má čtyři 72 pinové (tin-lead) SIMM sokety, do kterých je možné instalovat až 128 MB RAM. Sokety podporují jednostranné 4 MB moduly a jedno i dvoustranné 8, 16 a 32 MB moduly. Nejmenší možná paměť je 8 MB, největší je při použití čtyř 32 MB modulů, tedy 128 MB.

Časování paměti vyžaduje 60 ns rychlé stránkovací zařízení, nebo pro optimální výkon 60 ns EDO DRAM. Podporovány jsou paritní i neparitní paměti. Pro paritní mohou být nakonfigurovány ECC operace.

Čtyři paměťové sokety jsou rozděleny ve dvou bankách po dvou soketech Bank 1 a Bank 0. Každá banka zajišťuje 64/72 bitovou širokou datovou cestu. Oba SIMM v jedné bance musí být stejné velikosti i typu. V druhé bance může být jiný typ a velikost. Zaplněné mohou být buď obě banky nebo jen jedna, ale vždy oba sokety.

Velikost a typ SIMM je automaticky detekována BIOSem, a proto na desce nejsou žádné přepínače konfigurujiící paměť.

DRAM

EDO (Hyper stránková) - její rys je v tom, že uchovává paměťová data platná až do dalšího cyklu přístupu do paměti.

Parita a ECC

Je podporováno prověřování paměťových chyb a jejich oprava ECC. Paritní SIMM jsou detekovány samy, ale uživatel musí v SETUPu nakonfigurovat buď Paritu nebo ECC operace. Paritní paměť detekuje jednobitové chyby. ECC paměť dvoubitové a navíc opravuje jednobitové chyby.

CHIPSET

Chipset je od Intelu 82 440FX PCI set. Skládá se z řadiče mostu a paměti 82 441FX PCI a z urychlovače datové sběrnice 82 442FX. Další součástí je 82 371 SB, který zajišťuje propojení ISA a PCI sběrnice.

82 441FX

v 281 pinovém provedení zajišťuje následující:

- řízení rozhraní mikroprocesoru
 - a) hostitelskou sběrnici Pentia až na 66 MHz
 - b) 32 bitové adresování
- integrovaný řadič DRAM
 - a) 64/72 cestu k paměti, podpora ECC
 - b) podpora EDO a rychlého stránkování
 - c) od 8 do 256 MB základní paměti (lze využít jen do 128)
- plně synchronizované rozhraní PCI sběrnice
 - a) vyhovuje 2.1 až 5 V
 - b) frekvencím 25/30/33 MHz
 - c) přenos z PCI do DRAM více než 100 MB za sekundu
- bufferování dat
 - a) z Host do DRAM a z PCI do DRAM
 - b) podpora (výbušného) zápisu z Host do DRAM

82 442 FX

připojuje k 64 bitové procesorové datové sběrnici 64/72 bitovou paměťovou datovou sběrnici a 16 bitovou PMC soukromou datovou sběrnici. Pracuje paralelně s PMC, aby počítače s Pentii Pro měly vysoce výkonný paměťový subsystém. Je v 208 pinovém balení.

82 371 SB (PIIX3)

zajišťuje rozhraní mezi PCI a ISA sběrnicemi a zahrnuje dvoukanálové IDE rozhraní podporující až čtyři zařízení. Má čtyři 8 bitové a tři 16 bitové časovače čítače, dva osmikanálové řadiče přerušení, mapování přerušení z PCI na AT, NMI logiku, generování obnovování ISA adresy a PCI/ISA sběrnicové sloučení pro jedno zařízení.

Podpora IDE

Základní deska má dva nezávisle sběrnicově řízené PCI IDE rozhraní schopné obsluhovat zařízení PIO módu 3 a 4. BIOS zajišťuje LBA (adresace logického bloku) a ECHS (rozšířené cylindrové, sektorové nebo hlavové) překladové módy. Detekci přenosové rychlosti a schopnosti překladového módu IDE zařízení zajišťuje BIOS. Normálně programované I/O operace vyžadují nějaký procesorový čas.

Ve skutečných multitaskových systémech se procesorový čas uvolňuje používáním sběrnicově ovládané PCI, IDE může být použito k dokončení dalších úloh, zatímco probíhají diskové přenosy. Když se to zkombinuje s odpovídajícím Windows 95 driverem, IDE rozhraní může fungovat jako sběrnicově řízené PCI schopné obsluhovat PIO zařízení módu 4 s přenosovou rychlostí až 16 MB za sekundu.

Grafický podsystém

Základní desku je možné koupit buď s S3 Virge/DX SVGA grafickou kartou. Obě mají 2 MB EDO SOJ DRAM o rychlosti 50 ns. Dále deska podporuje S3 média channel známý jako LBP.

Zvukový podsystém

Mainboard zahrnuje 16 bitový stereo zvukový podsystém. Je založený na Crystal CS 42 36 multimedialním kodeku. Zajišťuje všechny audio a analogové mixační funkce, požadované pro přehrávání a záznam zvuku na osobním počítači.

BIOS a SETUP

Základní deska používá Intel BIOS, který je umístěn v přepisovatelné paměti, lehce vylepšitelné při používání určitého programu z diskety. Dále paměť kromě BIOSu obsahuje SETUP utilitu, test při startu POST, APM 1.2, autokonfigurační utilitku PCI, a Plug and Play 1.0a. Dále je podporováno zastínění BIOSu, které mu dovoluje startovat z 64 bitové, proti zápisu ochráněné DRAM. BIOS zobrazuje znakovou zprávu v průběhu POST identifikující typ BIOSu a pětičíselný revizní kód.

Organizace přepisovatelné paměti

Přepisovatelní komponent, Intel PA28FB002 BX 2MB, je organizován jako 256 kB.

Tabulka č. 5.4

Adresa		Přepisovatelná paměťová oblast
FFFF0000H	FFFFFFF0H	64 kB hlavní BIOS
FFFE0000H	FFFE0000H	16 kB starovací blok
FFFEA000H	FFFEA000H	8 kB ESCD oblast (k uchování plug&play dat)
FFFE9000H	FFFE9000H	4 kB rezervováno pro BIOS
FFFE8000H	FFFE8000H	4 kB OEM Logo BIOSu
FFFE0000H	FFFE0000H	32 kB rezervováno pro BIOS
FFFD0000H	FFFD0000H	64 kB rezervováno pro BIOS
FFFC0000H	FFFC0000H	64 kB rezervováno pro BIOS

Upgradování BIOSu

pro upgradování BIOSu je potřeba program FMUP.EXE který dokáže:

- nahrát BIOS do počítače ze souboru
- nahrát BIOS z počítače do souboru (jako zálohu)
- porovnat BIOS v počítači s BIOSem v souboru

Přepínač (jumper) na zpřístupnění SETUPu

při nastavení enable je možno při POST stiskem F1 vstoupit do hlavního SETUP menu. Tato výzva se taky při POST vypisuje. Při disable se výzva nevypisuje a nelze spustit SETUP.

Přehled menu obrazovek SETUPu

- při spuštění se ukáže hlavní menu
- šipky vpravo a vlevo mění výběrovou obrazovku
- šipky nahoru a dolů vybírají položku na obrazovce
- "Enter" vybranou položku modifikuje, při tom se zase někdy používají šipky
- někdy se při stisku "Enter" vynoří podobrazovka

Tabulka č. 5.5

Setup Menu Screen	Popis
Main	Nastavení a modifikace základních voleb PC, jako je čas, datum, disketové jednotky, harddisky, ...
Advanced	Modifikace složitějších vlastností PC, jako konfigurace periferií a konfigurace chipsetu
Security	Nastavení hesel k omezení přístupu k počítači
Exit	Ukládání nebo rušení změn
Setup Subscreen	
Floppy Options	Konfigurace disketových jednotek
IDE Device Configuration	Konfigurace IDE zařízení
Boot Options	Modifikace startovacích možností, např. změna boot sekvence
Peripheral Configuration	Modifikace voleb ovlivňující sériové a paralelní porty, rozhraní disketových jednotek
Advanced Chipset Configuration	Modifikace vlastností paměti a systémových sběrnic
Power Management Configuration	Nastavení APM
Plug and Play	Modifikace vlastností Plug&Play
Event Logging Configuration	Nastavení záznamů různých chyb při běhu počítače

Hlavní obrazovka BIOS SETUPu obsahuje:

- a) Systémové datum
 - nastavení aktuálního data, měsíc se vybírá z rolovacího menu
- b) Systémový čas
 - nastavení aktuálního času
- c) Floppy možnosti

- d) Hlavní IDE Master (hlavní hardisk)
 - určuje zda je k počítači připojeno IDE zařízení, při výběru se objeví podobrazovka konfigurace IDE zařízení
- e) Hlavní IDE Slave (podřízený hardisk)
- f) Druhý IDE Master (hlavní hardisk druhého kanálu)
- g) Druhý IDE Slave (zařízení podřízené druhému IDE Master)
- h) Jazyk - výběr jazyku, ve kterém bude zobrazován Setup, pouze z nainstalovaných jazyků
- i) Nastavení Bootu

5.9 Procesor PENTIUM II

Pentium II [36, 37] v sobě spojuje pokrokové technologie procesoru Pentium Pro s možnostmi technologie MMX. Technologie MMX umožňuje posílit výkon při zpracování zvuku, videa a grafických aplikací, stejně jako zrychlit šifrování dat a jejich kompresi. Nová je technologie pouzdrění, tzv. Single Edge Contact (S.E.C.) přináší širokou dostupnost procesoru Pentium II a poskytuje dostatečnou rezervu pro zvyšování výkonu i pro další generace procesoru.

Sběrníková architektura procesoru Pentium II odstraňuje omezení, která existují při použití jediné sběrnice, jak je tomu u procesorů generace s patičkou známou jako "Socket 7" (např. procesor Pentium). Nová architektura nabízí až třikrát vyšší prostupnost dat.

Technologie S.E.C. Cartridge:

S nástupem procesoru Pentium II představila společnost Intel technologii Single Edge Contact cartridge (S.E.C.). Je to nová montážní technologie, jež má nahradit starší technologii Pin Grid Array (PGA). Součásti jsou upevněny na podložku a poté zcela uzavřeny do plastového a kovového krytu, který tvoří vlastní tělo procesoru.

Tabulka č. 5.6

ČIP	Rychlost (MHz)	Rychlost sběrnice	L1 Cache Total	L2 Cache	mikronů	Tranzistorů (mil.)	Soket	Dosažitelnost
Pentium II (Klamath)	233	66	32 kB	modular	.35	7.5	Slot 1	nyní
	266	66	32 kB	modular	.35	7.5	Slot 1	nyní
	300	66	32 kB	modular	.35	7.5	Slot 1	nyní
Deschutes (low-power Klamath)	333	66	32 kB	modular	.35	7.5	Slot 1	nyní
	350	100	32 kB	?	.25	?	Slot 2	nikdy ?
	400	100	32 kB	?	.25	?	Slot 2	únor 98
	450	100	32 kB	?	.25	?	Slot 2	98
Katmai (první MMX-2)	500	100	32 kB	?	.25	?	Slot 2	začátkem 99
Williamette (enhanced Pentium II core)	?	100	?	?	.25	?	Slot 2	99

AL440LX Motherboard - základní deska pro Pentia II

AL440LX má následující charakteristiky:

konstrukce

- do rámu velikosti ATX
- deska má 12x7.75"

procesor

- jeden procesor typu Pentium II
- rychlost sběrnice 66 MHz
- podpora Pentii II všech rychlostí, napájení a frekvencí sběrnice
- 512 kB cache druhé úrovně na cartridzi se Single edge kontaktem(SEC)
- konektor typu Slot 1

hlavní paměť

- tři 168 pinové DIMM sokety
- podpora až 384 MB SDRAM
- může být ECC, ale nemusí

Intel 440LX AGPset(asi chipset) a PCI/IDE rozhraní

- Intel 82443LX PCI/AGP řadič(PAC):
 - a) integrovaný řadič sběrnice řízené PCI
 - b) integrovaný řadič urychleného grafického portu (AGP)
- Intel 82371AB PIIX4:
 - a) podpora až čtyř IDE zařízení
 - b) multifunkční most z PCI do ISA
 - c) řadiče USB a DMA
 - d) dvě rychlé IDE rozhraní
 - e) logika správy napájení
 - f) hodiny reálného času

I/O vlastnosti

- PC97307 Super I/O řadič
 - zahrnuje standartní vstupně/výstupní funkce:
 - rozhraní diskety
 - jeden multimódový
 - paralelní port
 - dva FIFO sériové porty
 - řadič klávesnice a myši
 - rozhraní kompatibilní s IrDA
- dvě USB rozhraní
- pět využitelných rozšiřujících slotů
- jeden ISA
- tři PCI
- jeden PCI/ISA

další vlastnosti:

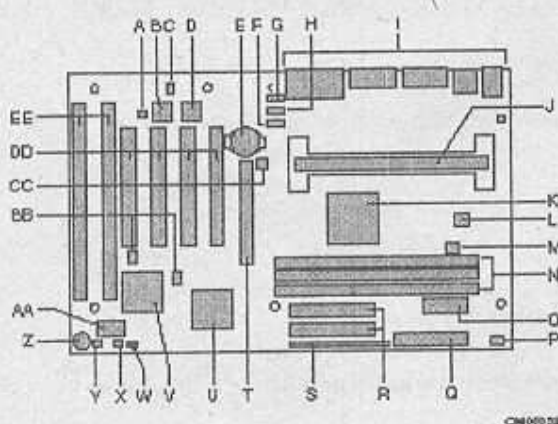
- intel/Phoenix BIOS
- zabudovaný AGP konektor
- plug and play kompatibilita
- jedno-jumperová konfigurace
- vylepšená správa napájení APM
- header pro buzení Ringem

Drivery a utility jsou k dostání od intelu.

Možnosti nastavení

- zvukovka:
 - a) Yamaha OPL3-SA codec
 - b) Yamaha OPL4-ML wavetable syntetizér
 - c) modul pro odkazy a design wavetable od Yamahy
 - d) zvukové konektory na zadním panelu: Line in, Line out a Mikrofon in
 - e) MIDI herní port
 - f) CD-ROM zvukový konektor
- hardware pro rozšíření managementu
- header pro buzení LAN
- bezpečnostní header rámu
- telefonní přípojka
- header pro LED od SCSI harddisku

Komponenty základní desky



- X - LED header pro SCSI hardisky
- Y - header buzení Ringem (telefonem?)
- Z - vnitřní reproduktor
- AA - 2Mbity flash paměti
- BB - headery pro wavetable moduly Yamahy
- CC - header pro větrák 2
- DD - PCI konektory
- EE - ISA konektory

- A - bezpečnostní header rámu
- B - Yamaha OPL4 komponenta
- C - header pro buzení LAN(po síti)
- D - Yamaha OPL3 komponenta
- E - baterie
- F - konektor line-in
- G - konektor CD-ROM audio
- H - telefonní přípojka
- I - I/O konektory na zadním panelu
- J - Patice typu Slot 1 na Pentium 2
- K - PCI/AGP řadič
- L - hardware rozšiřující řízení
- M - header na aktivní chladič na procesor (větrák 3)
- N - DIMM sokety na paměť
- O - hlavní napájecí konektor
- P - header na větrák 1
- Q - konektor na disketovku
- R - IDE konektory
- S - header předního panelu
- T - AGP konektor
- U - PIIX4
- V - vstupně/výstupní řadič
- W - konfigurační header

Obr. č. 5.22

Základní deska podporuje jediný procesor Pentium II. VID piny automaticky programují regulátor napájení na desce na napětí požadované procesorem. V této chvíli jsou podporovány procesory běžící interně na 233, 266 a 300 MHz, které mají 512 kB cache druhé úrovně.

Procesory v sobě mají MMX technologii a udržují si plnou kompatibilitu se svými předchůdci(8086, 286, 386, 486, Pentium a Pentium Pro). U procesoru je číselný koprocessor, který významně zvyšuje rychlost operací s pohyblivou desetinnou čárkou a vyhovuje ANSI/IEEE normě.

Balení procesoru

Procesor je zapakovaný v Cartridge SEC (jedno koncový kontakt). V Cartridge je jádro procesoru, cache druhé úrovně, teplotní destička a zadní kryt. Procesor je připojen k základní desce pomocí konektoru Slot 1, který má 242 pinů. V Slotu 1 je procesor chráněn upevňujícím mechanismem, přidělaným na desce. Chladič je stabilizován podporou chladiče, která je také umístěna na desce.

Cache druhé úrovně

Je umístěna v Cartridge a zahrnuje Burst Pipelined synchronní statickou RAM(BSRAM) a tag RAM. Jsou tam 4 BSRAM o celkové velikosti 512 kB. Cachovat je možné celou hlavní paměť.

Upgrade procesoru

Můžeme měnit za procesory běžící na vyšší rychlosti. Při změně musíme v BIOS konfiguračním módu změnit rychlost procesoru.

Paměť

Hlavní paměť: Deska má 3 DIMM sokety. Minimum paměti je 8MB, maximum 384MB. BIOS automaticky detekuje typ, velikost a rychlost paměti.

Následující vlastnosti jsou podporovány:

- DIMM se 168 piny a pozlacenými kontakty
- jediné 66 MHz SDRAM
- paměť bez ECC(64 bitu) i s ECC(72 bitu)
- pouze paměti na 3.3V
- Jedno i dvoustranné DIMM velikosti 8,16,32,64,128MB

Paměť může být v jednom, dvou nebo všech třech soketech a v každém může být úplně jiná.

Chipset

Intel 440LX chipset je desktop PCset třetí generace a je navrhován přímo pro procesory Pentium II. Skládá se z Intel 82443LX PCI/AGP řadiče (PAC) a PIIX4.

PAC

Zajišťuje sběrnice řídicí signály, adresové cesty a datové cesty pro přenosy mezi procesorovou host sběrnici, PCI sběrnici, AGP a hlavní paměti. PAC je v 492 pinovém provedení a zahrnuje:

- a) řízení procesorového rozhraní
 - procesorová host sběrnice až na 66 MHz
 - 32bitové adresování
 - host sběrnice vyhovuje GTL
- b) řadič DRAM
 - podpora synchronní DRAM(SDRAM)
 - 64/72 bitová cesta k paměti
 - autodetekce typu paměti
 - podpora 4,16,64 MB zařízení DRAM
 - symetrické a asymetrické adresování DRAM
 - podpora pro 3.3V DRAM

- c) rozhraní zrychleného grafického portu (AGP)
 - vyhovuje AGP specifikacím
 - podpora pro 3.3V AGP zařízení s datovým přenosem až 133 MHz
 - synchronní napojení na frekvenci host sběrnice
- d) plně synchronní rozhraní PCI sběrnice
 - vyhovuje PCI specifikacím
 - spojení z PCI do DRAM je lepší než 100 MB/s
 - podpora pěti PCI zařízení připojených k host a PCI-ISA I/O mostu
 - delayed transaction (odkládané transakce)
 - podpora prověřování a generování parity u PCI
- e) bufferování dat
 - bufferování při zápisu mezi host a DRAM, PCI a DRAM, AGP a DRAM
 - při burst (nějakém speciálním) zápisu mezi host a PCI se používá write combining (kombinovaný zápis)
 - podpora zápisu z konkurenční host, PCI a AGP do hlavní paměti
- f) podpora módu řízení systému SMM

PIIX4

Multifunkční PCI zařízení, zahrnující most mezi PCI a ISA, funkčnost PCI IDE, funkce USB host/hub a vylepšenou správu napájení, je v 324 pinovém provedení a obalu MBGA a má tyto vlastnosti:

- a) multifunkční most mezi PCI a ISA
 - podpora PCI sběrnice na 33 MHz
 - vyhovuje PCI specifikaci
 - plná podpora ISA nebo rozšířené I/O sběrnice
- b) řadič USB
 - dva USB porty
 - podpora klávesnice a myši na tomto portu
 - podpora UHCI rozhraní
- c) integrované dvoukanálové vylepšené IDE rozhraní
 - až 4 IDE zařízení
 - přenos v PIO módu 4 až při 16 MB/s
 - podpora Ultra DMA/33 synchronního módu pro přenos až při 33 MB/s
 - sběrnicově ovládaný mód s 8x32 bitovým bufferem pro burst přenosy PCI IDE
- d) vylepšený DMA řadič
 - dva DMA řadiče založené na typu 8237
 - podpora pro PCI DMA se třemi PC/PCI kanály a distribuovanými DMA protokoly
 - rychlý typ-FDMA pro redukované využití PCI sběrnice
- e) řadič přerušení založený na 82C59
 - podpora 15 přerušení
 - programovatelné na hranovou/hladinovou citlivost
- f) logika správy napájení
 - Sleep (spací)/resume (stornovací) logika
 - podpora buzení modemem skrz vstup Ring indikátoru
- g) hodiny reálného času
 - 256 bajtová CMOS RAM napájena z baterie
 - zahrnuje datumový alarm
- h) 16 bitové čítače/časovače založené na 82C54

AGP

Vysoce výkonné spojení pro aplikace založené na grafice, hlavně na 3D grafice. Je nezávislé na PCI sběrnici a je speciálně určeno pro zařízení s grafickým displejem.

Vlastnosti:

- čtení a zápis do paměti je pipelined a skrytý
- demultiplexing adresy a data na sběrnici pro skoro 100% sběrniceovou efektivitu
- AC časování pro 133 MHz přenosy, povoluje datový throughput (vlození přes) 500MB/s

AGP vyhovuje 66 MHz PCI specifikaci

USB

Na desce jsou dva USB porty, na každý lze připojit jedna periferie. Pro více zařízení je nutné připojit na USB venkovní osu s dalšími konektory. Základní deska plně podporuje universální hostem controller rozhraní (UHCI) a používá software a ovladače s ním kompatibilní.

Charakteristiky USB:

- samoidentifikující se periferie, které mohou být zapojené i v době, kdy počítač jede
- automatické mapování funkcí do ovladače a konfigurace
- podpora isochronních a asynchronních přenosu přes stejné dráty
- podpora až 127 zařízení
- zaručena šířka pásma a nízké utajení vhodné pro telefon, zvuk a jiné aplikace
- obsluha chyb a mechanismus oživení po haváriích je zabudován do protokolu

Super I/O Řadič

Tento řadič je ISA plug and play kompatibilní, multifunkční I/O zařízení s těmito charakteristikami:

- a) sériové porty
 - dva 16450/16550A, softwarové UART kompatibilní
 - vnitřně posílají/přijímají 16 bytů FIFO
 - 4 vnitřní osmibitové DMA volby pro UART s podporou SIR(USI) pomalých infra zařízení
- b) multimódový, dvousměrový paralelní port
 - Standardní mód, kompatibilní s IBM a Centronics
 - vylepšený paralelní mód (EPP) s podporou BIOSu a ovládačů
 - mód vysoké rychlosti a rozšířených schopnosti portu (ECP)
- c) řadič diskety
 - kompatibilní s DP8473 a N82077
 - 16 bytů FIFO
 - podpora PS/2 diagnostického registru
 - vysoce výkonný digitální datový oddělovač(DDS)
 - podpora drive módu PC-AT a PS/2
- d) ovladač klávesnice a myši
 - kompatibilní s výrobním standardem 8042A
 - mikrořadič General-purpose(hlavního smyslu?)
 - 8 bitová vnitřní datová sběrnice
- e) podpora IrDA a uživatelského infračerveného rozhraní

Rozhraní I/O řadiče je automaticky konfigurováno během bootu, ale můžeme ho ručně překonfigurovat v Setupu.

Řadič disket

podporuje módy PC-AT i PS/2 a v Setupu může být nakonfigurována na následující kapacity:

- 360 kB 5.25"
- 1.2 MB 5.25"
- 720 kB 3.5"
- 1.2 MB 3.5" (nutný ovladač)
- a.25/1.44 MB 3.5" palce
- 2.88 MB 3.5" palce

Podpora infračervených zařízení

Na I/O konektoru na předním panelu je šest pinů, na které lze připojit IR vysílač nebo přijímač, kompatibilní s Hewlet Packard HSDL-1000. V Setupu lze k připojenému IR zařízení přeměrovat sériový port 2. Toto spojení se používá pro přenos souborů do nebo z přenosných zařízení, jako jsou laptopy, PDA a tiskárny. Specifikace Infračervené Datové Asociace (IrDA) zaručuje datový přenos 115 kBaudů na vzdálenost 1 metr.

Podpora zákaznického IR zařízení

Na I/O konektoru je ještě jeden pin, na který se připojuje zákaznické IR zařízení. To se používá k ovládaní například telefonu, nebo multimedií (jako třeba hlasitosti, nebo změna písničky na CD). K tomu ale potřebujeme dodatečné softwarové a hlavně hardwarové vybavení.

Zvukový subsystém

OPL3-SA3 Zvukový systém

Volitelný zvukový podsystém charakterizuje Yamaha OPL3-SA3 (YMF715) zařízení. Jeho vlastnosti jsou:

- 16 bitový zvukový codec (kódovač)
- OPL3 FM (frekvenčně modulována) syntéza
- integrovaný 3D vylepšený stereo řadič, zahrnující všechny požadované analogové části
- rozhraní pro MPU-401 (asi nějaké midi) a joystick
- stereo převaděče z analogu do digitálu a naopak
- analogové mixování, anti-aliasing a obnovovací filtry
- podpora dekódování 16 bitových adres
- vstupy line, mikrofon
- ADPCM, A-law nebo ulav digitální zvuková komprese a dekomprese
- plně digitální řízení všech mixážních a hlasitostních funkcí
- softwarové přepínání mezi konektory Mic In a Line In
- Plug and Play kompatibilní
- dále je kompatibilní s Pentiem Pro a zvukovým systémem windows

OPL4-ML syntetizér wavetable

Volitelný wavetable syntetizér charakterizuje jediný chip OPL4-ML (YMF704). Ten integruje OPL3 zvukový systém, procesor General MIDI a ROM, ve které je wavetable (tabulka zvuku) do jedné součástky.

Vlastnosti:

- vyhovuje general MIDI systém 1
- rozhraní kompatibilní s MPU-401 UART modem

- FM syntéza je kompatibilní s OPL3
- wavetable syntéza generuje až 24 samostatných kanálů
- balení 100 pinový SQFP

Přehled vlastností BIOSu

Základní deska používá Intel/Phoenix BIOS, který je uchovávan v přepisovatelné paměti a může být upgradován z disku. Paměť zahrnuje ještě Setup program, POST, APM, autokonfigurační utilitku PCI, připravené Plug and Play pro windows95. BIOS může být shadowing. V průběhu POST zobrazuje BIOS zprávu, která identifikuje jeho typ a ověřovací kód.

Organizace flash paměti

Je to PA28FB200BX od Intelu, její 2 Mbyty jsou organizovány jako 256kBx8bitů a je rozdělena do úseků (viz tabulku č. 5.6). Tabulka ukazuje adresy obrazů v ROM v normalním módu (v obnovovacím módu se tyto adresy mění).

Tabulka č. 5.6

Adresa (Hex)	Velikost	Popis
FFFFC000 - FFFFFFFF	16 kB	Bootovací blok (při bootování se spouští jako první)
FFFFA000 - FFFFBFFF	8 kB	Vital Produkt Data (VPD), rozšířené systémové konfigurace (ESCD), konfigurace DMI, Plug & Play
FFFF9000 - FFFF9FFF	4 kB	Užíváno BIOSem (např. pro logy události)
FFFF8000 - FFFF8FFF	4 kB	OEM logo nebo scan flash
FFFFC000 - FFFF7FFF	224 kB	Hlavní blok BIOS

BIOS Setup Program

Setup program je na prohlížení a měnění nastavení BIOS. Dostaneme se do něj stiskem F2 po započeti POST, ale musí to být před tím, než se začne počítač bootovat.

Tabulka č. 5.7 Menu obrazovky

Setup Menu Screen	Popis
Maintenance (obrazovka údržby)	specifikace rychlosti procesoru a rušení hesel Setupu, je přístupná jen v módu Configure
Main (hlavní)	nastavování a zajišťování parametrů hardware
Advanced (rozšířená)	nastavování rozšířených vlastností chipsetu
Security (bezpečnostní)	nastavení úrovně bezpečnosti a zadávání hesel
Power (napájení)	nastavení vlastností power managementu (správy napájení)
Boot (startovací)	změna bootovacích nastavení a řízení dodávky proudu
Exit (ukončovací)	ukončení Setupu a možnost rušení všech změn

Maintenance Menu

rychlost procesoru - 200, 233, 266, 300-specifikuje rychlost v MHz
smaž všechna hesla - smaže uživatelská a správcovská hesla

Main Menu

typ procesoru - zobrazuje typ procesoru
rychlost procesoru - zobrazuje rychlost procesoru

Cache RAM - zobrazuje velikost cache druhé úrovně
Celková paměť - zobrazuje celkové množství RAM na desce
verze BIOS - zobrazuje, ve které verzi je BIOS
Jazyk-jazyky - vybírá, který jazyk bude BIOS používat
Systémový čas - hodina,minuta,sekunda - aktuální čas
Systémové datum - měsíc, den, rok - aktuální datum
Submenu Floppy options
Submenu Primárního IDE nadřazeného
Submenu Primárního IDE podřazeného
Submenu Sekundárního IDE nadřazeného
Submenu Sekundárního IDE podřazeného

Submenu Floppy options

zde se konfigurují disketové jednotky
Diskette A - velikosti - specifikuje kapacitu a velikost drive A
Diskette B - velikosti - specifikuje kapacitu a velikost drive B
Floppy write protect - disabled/enabled - vypíná nebo zapíná ochranu proti zápisu pro obě disketovky

IDE submenu

jsou 4, ale charakteristiky jsou stejné: typ-none/ATAPI/CD-ROM/User/Auto - specifikuje konfigurační mód IDE zařízení, User dovoluje nastavit cylindry, hlavy a sektory, Auto vyplní ty kolonky sám

cylindry - číslo - specifikuje počet cylindrů
hlavy - číslo(1-16) - specifikuje počet diskových hlav
sektory - číslo(1-64) - specifikuje počet diskových sektorů

Maximální kapacita - ukazuje jaká je maximální kapacita harddisku, hodnota je spočítána z předchozích tří údajů
Multi sektors transfer - disabled/2/4/8/16 - zde se nastavuje počet sektorů, které se přenesou v jednom bloku najednou do paměti
kontrola LBA módu - disabled/enabled - vypíná a zapíná LBA (když to změním poté, co byl hardisk zformátovaný, můžeme přijít o data)
Transfer mode - standartní/pio1...pio4 - volba metody přenosu dat mezi harddiskem a systémovou pamětí
Ultra DMA - Disabled/mode0/mode1/mode2 - určuje mód DMA pro harddisk

Advanced menu

Plug&Play O/S-no/yes-volba, jestli je zaveden operační systém, podporující P&P, když dáme ne, BIOS nakonfiguruje všechna zařízení, když ano, nechá BIOS P&P zařízení nakonfigurovat operačním systémem.
Reset configuration data - no/yes - při dalším bootu se smažou BIOS konfigurační data
Memory cache - disabled/enabled - zapíná, nebo vypíná cache
ECC Configuration - non ECC/ECC - určuje ECC operace, jestli ano, nebo ne
Submenu konfigurace zdrojů - konfiguruje paměťové bloky a IRQ pro ISA zařízení
Submenu konfigurace periférií - konfiguruje periferní porty a zařízení
Submenu vlastností klávesnice - nastavuje vlastnosti klávesnice
Submenu konfigurace videa - nastavení video vlastností
Submenu zápisu vyjímek - konfiguruje, jak se to bude zapisovat

6. PROCESORY JINÝCH FIREM.

V této kapitole jsou popsány procesory od jiných firem. Při popisu architektur jsou uváděna jen ta specifika, která daný procesor odlišuje od ostatních nebo která představují vlastnost odlišující se od základních charakteristických vlastností procesoru CISC a RISC.

Poněkud podrobněji, oproti ostatním procesorům RISC, jsou popsány architektury procesorů Acorn RISC Machine, včetně popisu jejich instrukčního souboru.

6.1 Acorn RISC Machine

Acorn RISC Machine (ARM) [3,19] je plně 32 bitový, universální mikroprocesor, který akceptuje principy procesorů s redukováným instrukčním souborem. Procesor je určen pro mikropočítače a pro nejrůznější aplikace v počítačové grafice, či řídicích systémech průmyslových podniků. Může být použit jako základní procesor pracovních stanic (workstations) nebo jako přídatný procesor (akcelerátor).

V roce 1988 byl dán do prodeje osobní počítač **Acorn Archimedes**, který je postaven na základě procesoru ARM.

Charakteristika procesoru ARM

Základní vlastnosti procesoru ARM jsou:

- 32 bitová datová sběrnice s propustností 32 MB/s
- 25 vnitřních 32 bitových registrů
- přístup do paměti pouze instrukcemi Load/Store
- částečné překrývání vnitřních registrů
- nejdelší doba reakce na přerušení 3 s
- hodinová frekvence 8 MHz
- možnost podmíněného vykonání instrukcí
- 26 bitová adresová sběrnice, dostupný lineární adresový prostor 64 MB
- průměrná rychlost vykonání instrukcí 3 až 4 MIPS
- 32 bitová vnitřní architektura
- možnost připojení standardních pamětí DRAM
- technická podpora pro virtuální paměťový prostor
- jednoduchý a výkonný instrukční soubor, jednoduše využitelné kompilátory jazyků vyšších prog. jazyků

Procesor ARM podporuje dva adresové módy. Adresovat lze jednak prostřednictvím čítače instrukcí, jednak prostřednictvím bázové adresy uložené v jednom z vnitřních registrů.

Použití koncepce přístupu do paměti pouze instrukcemi Load/Store (Load-Store Architecture) výrazně zjednodušuje výkonnou jednotku (Execution Unit) procesoru, protože pouze několik instrukcí pracuje přímo s pamětí. Většina instrukcí pracuje s vnitřními registry.

Zvýšení výkonnosti při přístupu do paměti poskytují instrukce násobného přístupu do paměti (Load and Store Multiple Registers), které umožňují rychlé přepínání kontextu, přičemž využívají sekvenčního přístupového režimu do paměti.

Procesor ARM podporuje dva základní typy přerušení, které se navzájem liší **prioritou a použitím registrů**. Nejkratší doba akceptování požadavku na přerušení je poskytována režimem rychlého přerušení FIQ (Fast Interrupt Request). Druhý typ přerušení - IRQ (Interrupt Request) je použit pro obsluhu přerušení nevyžadujících extrémně krátké doby odezvy nebo v případě, že vlastní obsluha přerušení je oproti době reakce procesoru mnohonásobně delší.

Architektura procesoru ARM

Procesor ARM má 44 základních instrukcí a všechny mají jednotnou šířku 32 bitů. Pouze arit.-logické instrukce s registry, případně s přímými operandy, jsou vykonávány v jednom taktu.

V procesoru je realizováno 3-stupňové zřetězení instrukcí.

Procesor je schopen činnosti ve čtyřech režimech (v jednom uživatelském a třech privilegovaných):

- v režimu uživatelskémUSR
- v režimu supervizora SUP
- v režimu přerušení IRQ
- v režimu rychlého přerušení FIQ

Procesor ARM obsahuje množinu 25 částečně se překrývajících 32 bitových registrů, přičemž programově přístupných je v každém režimu činnosti procesoru pouze 16 registrů.

Z množiny 25 registrů procesoru ARM je 15 registrů universálních a zbývajících 10 má speciální funkce (stavové slovo procesoru, registry přiřazené daným režimům činnosti atd.)

Na obr. č. 6.1 je uvedena množina všech registrů procesoru ARM.

Režim	mód	Registry						Poznámka	
Uživatelský	0	R0 až R9	R10	R11	R12	R13	R14	R15 (PSW)	Privilegované
Rychlé přerušení	1		FIQ	FIQ	FIQ	FIQ	FIQ		
Přerušení	2		R10	R11	R12	R13 IRQ	R14 IRQ		
Supervisor	3		R10	R11	R12	R13 SVC	R14 SVC		

Obr. č. 6.1 Množina registrů procesoru ARM

Registry R0 až R13 jsou přístupné v uživatelském režimu pro libovolný účel. Patnáctý registr R14, návratový registr uživatelského režimu (User Mode Return Link), je určen výhradně pro uživatelský režim.

Aktuální režim činnosti procesoru určují nejnižší dva bity stavového slova (PSW).

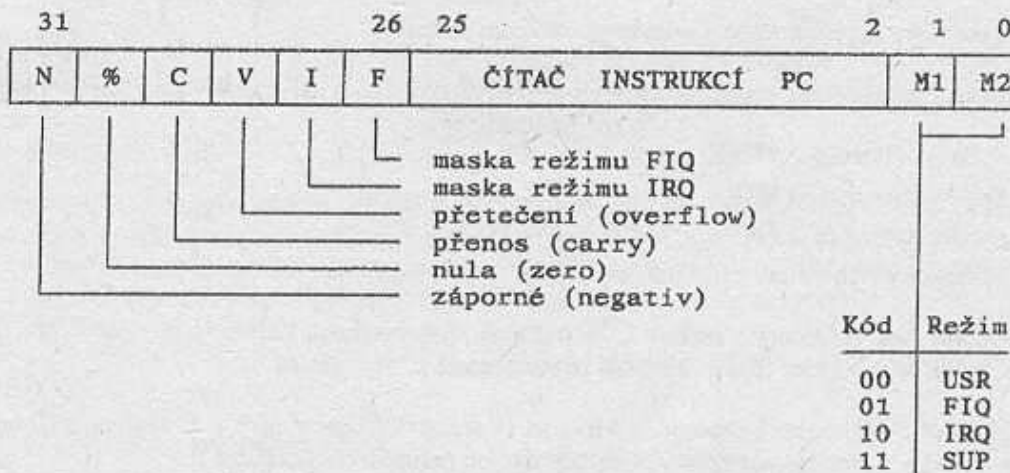
Pět registrů je speciálně určeno pro režim rychlého přerušení (FIQ). Pracuje-li procesor v režimu FIQ, je těchto pět registrů mapováno do registrů R10 až R14 uživatelského režimu. V registru R14 je uložena návratová adresa do přerušeného programu-šetří paměťové cykly v případě uložení návratové adresy do zásobníku v hlavní paměti. Režim FIQ je typicky využíván pro realizaci velmi krátkých programů obsluhy přerušení.

Další dva registry jsou určeny pro režim přerušení (IRQ) a v tomto režimu překrývají registry R13 a R14 uživatelského režimu. Registr R14 je registrem návratové adresy do přerušeného programu. Tohoto režimu se používá pro obsluhu přerušení, u nichž délka obsluhy přerušení výrazně přesahuje čas na "úklid" registrů procesoru.

Další dva registry jsou určeny pro režim supervizora.

V případě aktivace tohoto režimu prostřednictvím instrukce SWI budou těmito registry překryty registry R13 a R14 uživatelského režimu. Určení těchto registrů je identické jako ve výše uvedených režimech.

Poslední registr (R15) obsahuje stavové slovo procesoru a čítač instrukcí (viz. obr. č. 6.2). Tento registr je sdílen všemi režimy činnosti. Významově nejvyšších 6 bitů PSW obsahuje stav procesoru (Význam kombinace: Negative, Zero, Carry a Overflow), dalších 24 bitů představuje čítač instrukcí a významově nejnižší dva bity obsahuje aktuální režim činnosti procesoru.



Obr. č. 6.2 Registr stavového slova procesoru ARM

Specifika procesoru ARM

Jak bylo uvedeno, procesor ARM obsahuje množinu částečně se překrývajících registrů, takže v případě přerušení nemusí být proveden kompletní úklid registrů procesoru.

Zkrácení doby odezvy procesoru je dosaženo, v případě režimu rychlého přerušení FIQ, použitím čtyř lokálních univerzálních registrů a jednoho registru s návratovou adresou. Tyto registry mohou obsahovat všechny ukazatele a různé čítače používané v jednoduchých procedurách obsluhy "Vstupu/Výstupu", takže lze dosáhnout velmi rychlého opakovaného přepínání procesoru mezi režimem "Uživatelským" a režimem "Rychlého přerušení".

V ostatních režimech (režim přerušení IRQ a režim supervizora SUP) je vždy vyčleněn jeden univerzální registr a jeden registr s návratovou adresou.

Dále procesor akceptuje následující přerušení:

- chyba v adresování (Address Exception Trap)
- chyba při čtení nebo zápisu dat do vnější paměti (Data Fetch Cycle Aborts)
- chyba při čtení instrukce z vnější paměti (Instruction Fetch Cycle Aborts)
- přerušení programovými prostředky (instrukce SWI)
- nedefinovaný kód instrukcí (Undefined Instruction Traps),
- reset procesoru

Procesor ARM poskytuje 26 bitovou adresu lineárního paměťového prostoru, což umožňuje adresovat 64 MB fyzického paměťového prostoru. Odkaz na data mimo rozsah způsobí přerušení v důsledku chyby adresování (Address Exception Trap).

Instrukce programového přerušení (SWI) je použita k přechodu z uživatelského režimu do režimu supervizora.

V průběhu uskutečnění této instrukce vykoná procesor následující posloupnost:

- uloží čítač instrukcí R15 do registru R14, který je vyhrazen pro režim SUP
- nastaví dva významově nejnižší bity PSW na indikaci režimu supervizora
- předá řízení na adresu danou vektorem softwarových přerušení

Tabulka č. 6.1 Vektory obsluh přerušení

Adresa	Funkce	Priorita
000 0000	Reset	0
000 0014	Chyba adresace	1
000 0010	Abort (datový)	2
000 001C	Rychlé přerušení (FIQ)	3
000 0018	Přerušení (IRQ)	4
000 000C	Abort (instrukční)	5
000 0004	Nedefinovaný instrukční kód	6
000 0008	Softwarové přerušení	7

0 ... nejvyšší priorita, 7 ... nejnižší priorita

Vektor obsluh jednotlivých přerušení je uveden v tab. 6.1. (Processor Exception Vector Map).

Instrukční soubor procesoru ARM

Instrukční soubor procesoru ARM se skládá z instrukcí, které je možno rozdělit do následujících skupin:

- instrukce zpracování údajů - zpracování registrových operandů, zpracování přímých operandů, nastavení podmínkového kódu a instrukce aritmeticko-logické
- instrukce jednoduchého přenosu údajů (4 instrukce LOAD/STORE - přímé nebo indexové)
- instrukce blokového přenosu údajů - instrukce vykonávají přenos mezi skupinou registrů a spojitou oblastí paměti, přičemž jeden registr je použit jako směrnik
- instrukce větvení a větvení s uchováním návratové adresy (tyto instrukce odkládají PSW do R14)
- instrukce přechodu do režimu supervizora, které zahrnují i programové přerušení

Všechny instrukce obsahují čtyřbitové pole podmínky, které determinuje vykonání této instrukce. **Vykonání všech instrukcí je podmíněno rovností kódu podmínky v instrukci s kódem podmínky, který je odvozen z příznaků v aktuálním PSW.**

Instrukce zpracování údajů pracují pouze nad souborem vnitřních registrů a každá z těchto instrukcí obsahuje **reference na tři operandy: cílový a dva zdrojové.**

Cílovým registrem (Rd) může být kterýkoliv z registrů včetně registru PSW (R15).

Zdrojové registry mohou být zadány dvěma způsoby:

- oba zdrojové operandy (Rm,Rn) jsou registry procesoru
- jeden operand je registr (Rn), druhý je 8bitová konstanta

Instrukce přenosu údajů jsou použity k přenosu dat mezi pamětí a souborem registrů (Load) a naopak (Store).

Efektivní adresa je vypočtena součtem obsahu zdrojového registru (Rn) a posuvu (offset) daného 12 bitovou konstantou nebo obsahem dalšího registru (Rm). Posuv (offset) může být přičítán k indexovému registru nebo od něho odčítán. Indexový registr je dán registrem Rm. Inkrementace a dekrementace může být realizována pomocí indexování nebo postindexování v závislosti na požadovaném adresním módu.

Instrukce přenosu údajů mohou pracovat jak s byty, tak s dlouhým slovem (32 bitů). Byte přečtený z paměti je uložen do významově nejnižších 8 bitů cílového registru a zbytek obsahu je zaplněn nulami. V případě zápisu bytu do paměti je zapsáno 8 významově nejnižších bitů zdrojového registru bez dalších změn v tomto registru.

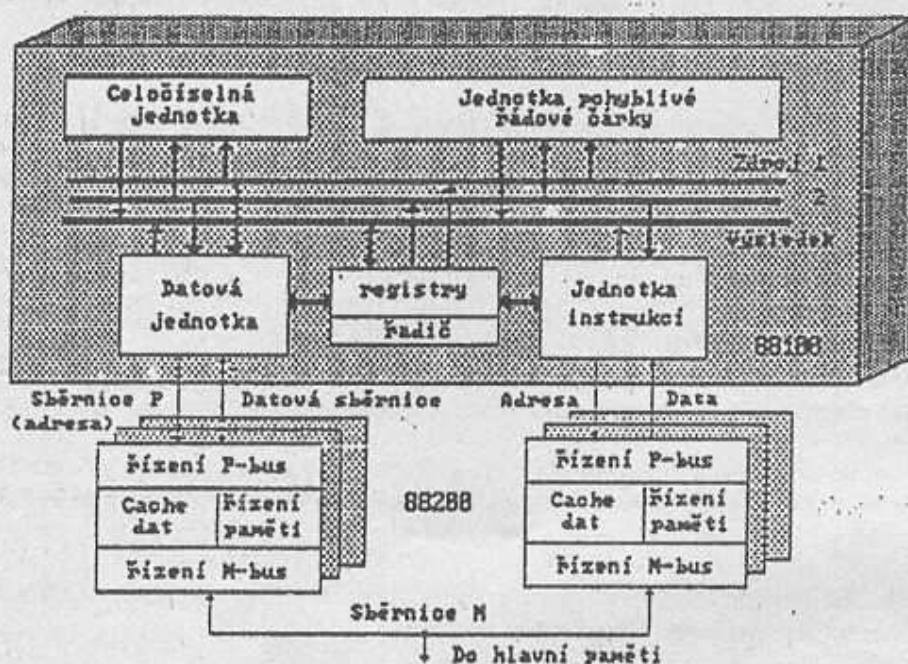
Technickými prostředky procesoru ARM je podporováno adresování fyzické i virtuální paměti. Dva typy instrukcí přenosu obsahují bit T (Translate), který označuje požadavek překladu virtuální adresy na adresu fyzickou. Hodnota bitu T je vyvedena na externí vývod procesoru, který má označení TRANS, čímž je umožněna spolupráce s externí MMU.

Instrukce blokového přenosu zabezpečují přenos několika registrů jednou instrukcí. Instrukce obsahuje pole bitů, z nichž každý odpovídá jednomu registru viditelnému v daném režimu. Bit 0 odpovídá R0, bit 1 odpovídá R1 atd. Je-li bit nastaven (má hodnotu 1), je odpovídající registr zahrnut do přenosové operace.

Instrukce programového přerušení jsou použity především pro přechod do režimu supervizora. PSW je ukládáno do registru R14 režimu supervizora a hodnota čítače instrukcí je dána adresou softwarového přerušení.

6.2 RISC procesor Motorola 88000

Jedním z představitelů 3. generace RISC je sada čipů firmy Motorola s typovým označením 88000 [3,19].



Obr. č. 6.3 Architektura procesoru Motorola 88000

RISC procesor Motorola 88000 se skládá ze dvou základních prvků:

- plně 32 bitového procesoru 88100 s jednotkou pohyblivé řádové čárky
- jednotky (jednotek) 88200, kterou představuje vyrovnávací paměť a obvody správy hlavní paměti MMU. Jednotku 88200 je možno použít separátně jako vyrovnávací paměť instrukcí nebo dat, případně pro data a instrukce současně

Architektura procesoru

32 bitový procesor 88100 obsahuje 32 datových registrů a 32 registrů se speciálními funkcemi. Přímou může adresovat 4 GB paměti.

Instrukční soubor obsahuje 51 instrukcí. Všechny celočíselné, logické a bitově orientované instrukce vykonává procesor v jednom cyklu. Data mohou být uspořádána jak podle definice **big-endian**, tak i podle definice **little-endian**.

Instrukční soubor procesoru 88100 obsahuje speciální instrukce pro operace se skupinami bitů. Jejich prostřednictvím je možno v jednom taktu procesoru realizovat maskování, extrakci, či nastavení skupiny bitů v jednom 32 bitovém slově.

U procesoru Motorola 88000 je také implementována technika zpožděných skoků.

Architektura procesoru Motorola 88000 akceptuje tzv. **harwardskou architekturu** - oddělenou paměť pro data a pro instrukce.

Procesor 88100 obsahuje **čtyři** základní jednotky (viz. obr. č. 6.3).

- jednotka celočíselné aritmetiky, včetně celočíselné násobičky a děličky
- jednotka pohyblivé řádové čárky, sčítačka je realizována pomocí **čtyřstupňového zřetězení**, násobička **pětistupňového aritmetického zřetězení**
- datová jednotka (pomocí **třístupňového zřetězení**) - vykonává nezávislé načítání operandů
- instrukční jednotka (**pomocí dvoustupňového zřetězení**) - vykonává přístup do paměti instrukcí

Všechny 4 jednotky jsou schopné pracovat paralelně a mohou vykonávat současně až 11 různých operací.

Procesor 88100 je vyroben technologií HCMOS (High Performance Complementary Metal-Oxide Semiconductor), ale počítá se s technologií ECL (Emitter Coupled Logic), jejímž použitím pak bude možné procesor taktovat až na 100 MHz.

Vysoké přenosové kapacity se dosahuje díky **čtyřem** navzájem odděleným sběrníkovým systémům. Adresová i datová sběrnice jsou 32 bitové. Při taktu 20 MHz je možno uskutečnit přenos dat až 80 MB/s.

K obvodu 88100 je možné připojit až 8 obvodů 88200, čtyři pro data a čtyři pro instrukce.

Funkční popis a instrukční soubor

V procesoru 88100 je technická podpora maximálního využití registrů řešena dvěma způsoby:

1. Sada registrů je přístupná pomocí tří vnitřních sběrnic, dvou zdrojových a jedné cílové. Po zdrojové sběrnici jsou přenášeny vstupní operandy instrukcí. Výsledek operace je předáván na cílovou sběrnici. Takto jsou umožněny **tři** současné přístupy do registrů, což omezuje možnost vytváření konfliktů na minimum.
2. Mnohonásobné přístupy je nutno synchronizovat. Synchronizace je zabezpečena registrem řídicích bitů (Semafor-Scoreboard). Pracuje-li instrukce s registrem, nastaví se v registru příznaků bit, který náleží cílovému registru, na jedničku. Ostatní instrukce pokračují v činnosti tak dlouho, dokud jejich zdrojové a cílové operandy nemají v registru příznaků nastaven bit. Po ukončení činnosti nastaví instrukce v registru příznaků **nulu**, čímž ho uvolní pro použití dalším instrukcím.

Systém Motorola 88000 může být použit k realizaci multiprocesorových počítačových systémů se sdílenou pamětí. V těchto systémech se vyžaduje důsledná synchronizace přístupu do oblasti společné paměti, aby zjistil, zda v ní již neoperuje jiný procesor. V opačném případě je nucen čekat, případně se zařadit do fronty ostatních čekajících procesorů.

Tato činnost je řízena pomocí semaforů, které jsou využity operačním systémem k řízení přístupu do oblasti společné paměti. Pro testování a modifikaci semaforů obsahuje instrukční soubor 88000 atomickou instrukci **XMEM**.

Začne-li jeden procesor v multiprocesorovém systému vykonávat instrukci **XMEM**, činnost všech ostatních procesorů je přechodně zakázána.

Mikroprocesor i860 tvoří následující bloky (viz. obr. č. 6.4 [3]):

- centrální řídicí jednotka (Core Execution Unit)
- jednotka řízení výpočtů v pohyblivé řádové čárce
- sčítačka v pohyblivé řádové čárce
- násobička v pohyblivé řádové čárce
- grafická jednotka
- jednotka správy paměti MMU a stránkovací jednotka (Memory Mapping Unit)
- vyrovnávací paměť instrukcí
- vyrovnávací paměť dat
- řídicí jednotka sběrnice a vyrovnávacích pamětí

Centrální řídicí jednotka řídí veškerou činnost mikroprocesoru i860. Je dekomponována do **čtyřstupňového řetězce**, který realizuje instrukce přenosové (Load, Store), celočíselné, bitově orientované, instrukce přenosu řízení a vykonává zavedení instrukcí pohyblivé řádové čárky do jednotky pohyblivé řádové čárky.

Pro zpracování celočíselných operandů má k dispozici **třicet dva 32 bitových registrů**. Přenosové instrukce Load a Store přenáší 8 bitové, 16 bitové a 32 bitové operandy z/do těchto registrů.

Data mohou být uspořádána jak podle definice **big-endian**, tak i podle definice **little-endian**.

Procesor poskytuje technickou podporu pro řešení přerušení (vnitřních i vnějších) a pro ladění programů.

Procesor i860 poskytuje ze všech uvedených procesorů nejvšeobecnější koncepci implementace **atomických instrukcí**. Jeho instrukční soubor obsahuje **dvě instrukce Lock a Unlock**. Posloupnost instrukcí v programu umístěná mezi touto dvojicí je procesorem i860 chápána jako **atomická operace**.

Technické prostředky **jednotky pohyblivé řádové čárky** jsou připojeny k množině registrů pohyblivé řádové čárky, které mohou být zpřístupněny v několika režimech (**třicet dva 32 bitových registrů, šestnáct 64 bitových registrů nebo osm 128 bitových registrů**). Vše je podporováno speciálními variantami příkazů Load a Store. **Všechny instrukce pohyblivé řádové čárky používají tyto registry jako zdroje a cíle svých operandů.**

Jednotka řízení výpočtů v pohyblivé řádové čárce řídí činnost sčítačky a násobičky v pohyblivé řádové čárce, realizaci instrukcí, zpracování a reakci na všechna přerušení na základě vstupních a výstupních operandů a aktualizuje stavové bity ve stavovém registru pohyblivé řádové čárky.

Sčítačka a násobička mohou pracovat nezávisle, paralelně, a produkovat tak dva výsledky v jednom časovém okamžiku.

Sčítačka v pohyblivé řádové čárce vykonává **součty, rozdíly, porovnání a konverze 32 bitových nebo 64 bitových operandů**. Je implementována prostřednictvím **třístupňového zřetězení**.

Násobička v pohyblivé řádové čárce vykonává operaci násobení pro celočíselné operandy i pro operandy v pohyblivé řádové čárce.

Násobička je implementována prostřednictvím **třístupňového zřetězení**, přičemž v případě operandů v pohyblivé řádové čárce s dvojnásobnou přesností nepostupuje výsledek v každém taktu, ale v každém druhém.

V **grafické jednotce** je implementována speciální celočíselná logika, která podporuje vykonávání **třírozměrných grafických operací**. Grafická jednotka rozpoznává obrazové prvky (pixel) jako 8 bitové, 16 bitové nebo 32 bitové operandy. Je schopna vypočítat individuální hodnoty barev (červená, zelená, modrá) jednoho obrazového prvku, ale i vykonat operace paralelně nad 64 bitovými operandy. Grafické schopnosti i860 jsou založeny na předpokladu, že povrchy objemových těles jsou kresleny prostřednictvím jejich **aproximace miniaturními mnohoúhelníkovými ploškami**. Intenzity barev vrcholů těchto polygonů a vzdálenosti od pozorovatele jsou známy.

Vzdálenosti a barvy ostatních bodů aproximujících plošek musí být vypočteny interpolací. Instrukční soubor i860 tyto interpolace prostřednictvím grafické jednotky přímo podporuje.

Jednotka správy paměti a stránkovací jednotka implementuje mechanismy ochrany, stránkování a virtualizace paměti. Jednotka stránkování používá TLB k přepočtu virtuální adresy na fyzickou a ke kontrole přístupových práv. Jsou implementovány dvě základní privilegia při operacích s pamětí a to: uživatelská a supervizora.

Uživatelským aplikacím a operačnímu systému je k dispozici 4 GB adresový prostor.

Vyrovňovací paměť instrukcí je dvoucestná asociativní paměť o kapacitě 4 kB postavená ze 32 bitových bloků. Je schopna přenášet 64 bitů v jednom strojovém kroku (tj. 320 MB/s při 40 MHz).

Vyrovňovací paměť dat je dvoucestná asociativní paměť o kapacitě 8 kB postavená ze 32 bitových bloků. Je schopna přenášet 128 bitů v jednom strojovém kroku (tj. 640 MB/s při 40 MHz).

Řídící jednotka sběrnice a vyrovnávacích pamětí zajišťuje přístup dat a instrukcí k centrální řídicí jednotce.

Cyklicky od ní přijímá požadavky a specifikace, vykonává aktualizaci údajů ve vyrovnávacích pamětech dat a instrukcí, řídí TLB transformace a realizuje rozhraní pro vnější sběrnice. Její zřetězená struktura umožňuje až tři současně rozpracované sběrnice cykly.

Obě vyrovnávací paměti implementují skupinově asociativní mapování. Obě mohou být procesorem zpřístupněny současně (tzn. 960 MB/s při 40 MHz).

Mikroprocesor i860 je schopen vykonat v **jednom cyklu jednu nebo dvě instrukce** (single, resp. dual instruction mode).

V **režimu vykonávání jedné instrukce** je možné zpracovat instrukce buď pouze hlavní výkonnou jednotkou nebo jednotkou pohyblivé řádové čárky.

V **duálním režimu** mohou obě jednotky pracovat souběžně.

V tomto režimu může výkonná jednotka zavádět operandy, ukládat výsledky a realizovat řídicí operace cyklu, přičemž vlastní operaci s daty vykonává souběžně jednotka pohyblivé řádové čárky. Pak je možno implementovat knihovnu **vektorových operací** nad vektory s prvky v pohyblivé řádové čárce. Tímto způsobem je možno dosáhnout až 60 MFLOPS výkonu při taktovací frekvenci 40 MHz.

Procesor i960

Dalším procesorem RISC firmy Intel je **mikroprocesor i960** [3,19]. Tento procesor poskytuje sekvenční programové prostředí, které je vhodné pro implementaci **paralelismu** na jednom procesoru. Instrukční soubor i960 je orientován na data v registrech, instrukce pracují se třemi operandy, výměna dat s pamětí je vyhrazena instrukcím Load/Store. Paměť je adresována 32 bitovou adresou.

Na dosažení **paralelismu** poskytuje i960 rozsáhlý instrukční soubor. Několik instrukcí je možno vykonávat souběžně (i operace zápisu a čtení operandů).

Program má přístup ke **třiceti dvěma 32 bitovým** všeobecně určeným **registrům** a **16 globálním registrům**. Tyto registry jsou dostupné v libovolném časovém okamžiku. Právě aktivnímu procesu je dostupno dalších **16 lokálních registrů**. Při každém volání nového procesu přidělí procesor i960 nové lokální registry.

Implementace registrů a strategie jejich zpřístupnění byla navržena s cílem co nejlepšího přizpůsobení konvencím v používání dat v jazycích vysoké úrovně (např. jazyk C).

Parametry mezi procedurami jsou předávány prostřednictvím globálních registrů, lokální data procedur jsou uchovávána v lokálních registrech.

Instrukce procesoru i960 mají pevný formát a jednotnou délku 32 bitů. Většina instrukcí akceptuje zásadu jednoduchosti. Součástí instrukčního souboru jsou i složité instrukce. Jejich realizace je procesorem optimalizována s cílem maximálního využití **paralelismu mezi vnitřními jednotkami procesoru**.

6.4 Procesory firmy DEC

V roce 1992 oznámila firma DEC nový procesor typu RISC, který pracuje s šířkou toku dat 64 bitů. Tři hlavní principy, které byly sledovány při návrhu, jsou **vysoká frekvence hodin, provádění několika instrukcí současně a možnost použití v multiprocesorovém režimu**.

Procesor Alpha [3, 19, 38] je určen pro práce pod operačním systémem VMS i pod systémem OSF/1 (ekvivalent systému UNIX). Architektura procesoru ALPHA sdružuje arit. jednotku pro operace s pevnou řádovou čárkou a arit. jednotkou pro operace s pohyblivou řádovou čárkou na jednom čipu.

Tento čip obsahuje 1,68 miliónu tranzistorů a je vyroben technologií CMOS 0.75 μm . Při hodinové frekvenci 150 MHz dosahuje tento procesor výkonnosti 300 MIPS nebo 150 MFLOPS.

Předpokládá se však i použití na frekvenci 200 MHz.

Na čipu procesoru jsou realizovány dvě vyrovnávací paměti určené pro data a pro instrukce, každá s kapacitou 8 kB.

Pro jejich adresování se používá adresář s přímým mapováním, vyrovnávací paměť dat používá současný zápis do hlavní paměti.

Na čipu je integrována i jednotka pro řízení paměti (MMU), v níž se překládají virtuální adresy délky 64 bitů na reálné adresy délky 34 bitů. Velikost stránky je 8 kB, počet adres v TLB s plně asociativním adresářem je 32. K procesoru lze připojit vnější vyrovnávací paměť až do kapacity 8 MB.

Jednotka pro operace s pohyblivou řádovou čárkou používá jednak formáty podle normy IEEE 754, jednak formáty známé z počítačů řady VAX. Operace s pohyblivou řádovou čárkou probíhají proudově v **deseti** sekcích (stupně), jednotka pro operace s pevnou řádovou čárkou je rozdělena do **sedmi** sekcí (stupně).

6.4.1 Procesor ALPHA 21064A

Základní charakteristika

Mikroprocesor ALPHA 21064A patří do rodiny procesorů Digital, který implementuje architekturu této firmy. Procesor je realizován technologií 0.5 μm . Super-pipeline procesor využívá duální instrukční sady a může pracovat s frekvencí 200, 233, 275 nebo 300 MHz. Procesor je navrhován tak, aby dosáhl co největší rychlosti a přitom poskytoval multiprocesoring a hlavně aby pracoval na všech platformách operačního systému. Procesor běží na všech moderních operačních systémech jako je např. Windows NT, Digital UNIX nebo OpenVMS.

Základní rysy

- 64-bitový se zdokonalenou RISCovou architekturou s podporou multiprocesoringu
- schopnost provádět 400, 466, 550 nebo 600 milionů instrukcí za jednu sekundu
- může mít až 16 GB adresovatelné paměti
- na čipu je 32-bitový zapisovací buffer
- 64 nebo 128-bitová datová sběrnice
- externí cache 256 kB až 16 MB
- 16 kB datová write-through cache s paritou
- 16 kB instrukční write-through cache s paritou
- vylepšena větev předpovědi logických instrukcí
- sériové rozhraní ROM
- 3.3 V chip pro vstupně/výstupní operace s 5 V jádrem

Mikroarchitektura

Procesor obsahuje tyto části:

Instruction Fetch and Decode Unit (IDU) - tato část dále pak obsahuje instrukční překladový buffer (ITBs) a jednotku na předpověď logických instrukcí (**branch unit**). IDU provádí instrukční "fetch", kontrolu prostředků a zajišťuje komunikaci s IEU, LSU, FPU nebo s branch unit. IDU navíc řídí průchody pipeline, restart a selhání.

Integer Execution Unit (IEU) - jednotka pro práci s čísly s pevnou řádovou čárkou - obsahuje běžné použití sčítačky, násobičky, válcového posouvače a bitové masky. Je tvořena 32 vstupními 64 bitovými celočíselnými registry.

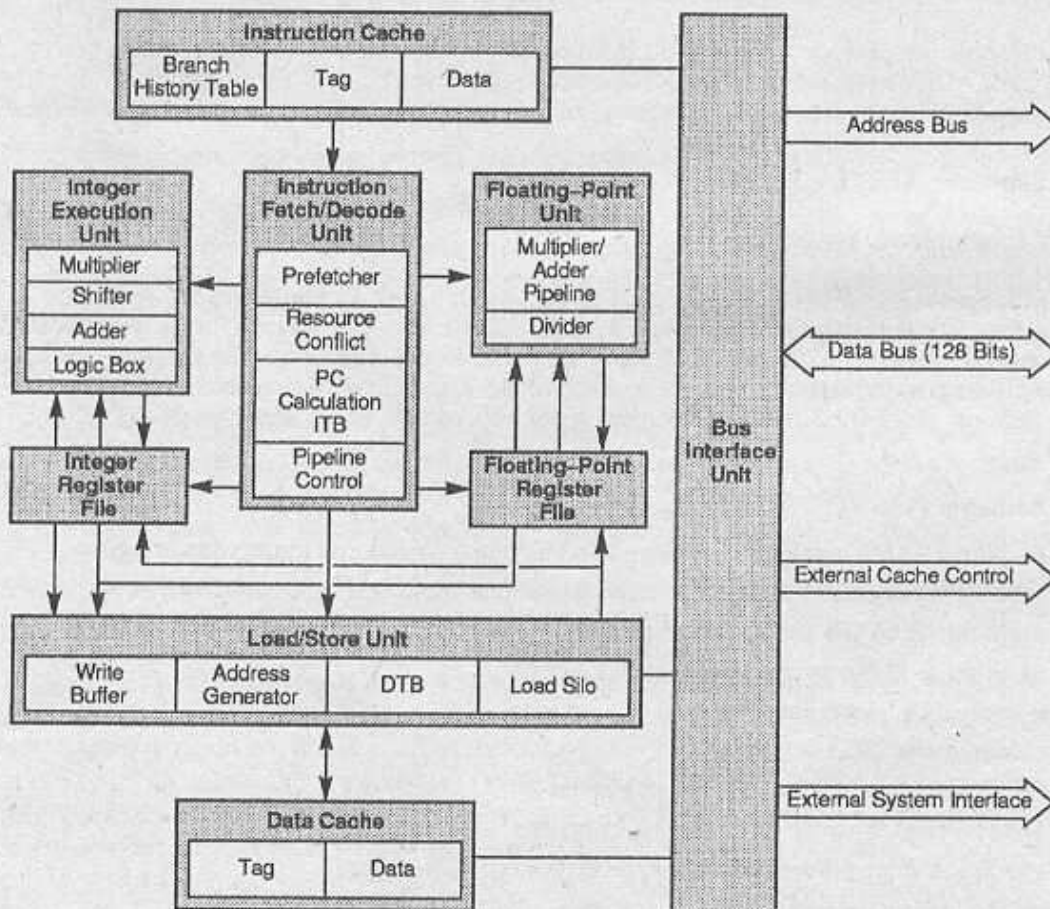
Floating Point Unit (FPU) - provádí se zde nezávislé dělení. Podporuje jednoduchou a dvojitou přesnost. Obsahuje také 32 vstupních 64 bitových registrů s pohyblivou řádovou čárkou.

Load and Store Unit (LSU) - obsahuje 4 hlavní části, a to adresovací část pro datové operace, plnění zásobníku, zapisovací buffer a Data Cache rozhraní. LSU podporuje všechny operace jak celočíselné, tak v pohyblivé řádové čárce.

Instruction Cache a Data Cache (Icahce a DCahee) - jsou 16 kB s paritní ochranou. Adresování se provádí 32-bitovými bloky.

External Cache - většinou je tvořena pamětí typu SRAM. Cache je typu write-back a read-allocate s 8 bitovými bloky. Programovatelný interface podporuje velikosti paměti od 256 kB do 16 MB.

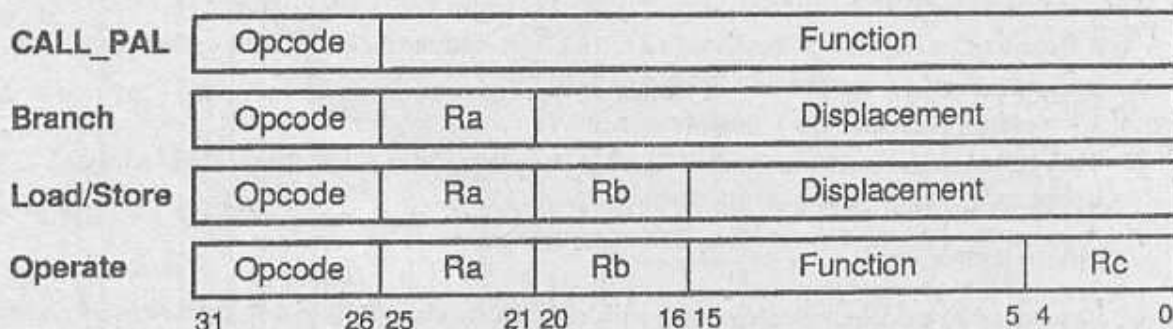
Virtual Address Space - virtuální adresa je 64 bitová celočíselná hodnota bez znaménka, která specifikuje adresu na virtuálním zařízení.



Obr. č. 6.5 Blokové schéma procesoru

Instrukční sada

Všechny instrukce jsou 32 bitové a využívají jeden ze čtyř formátů instrukce. Každý formát používá 6 bitový operační kód a 0, 1, 2 nebo tři 5 bitových položek.



Obr. č. 6.6

CALL_PAL - tento instrukční vektor provádí privilegované a neprivilegované instrukce softwarové úrovně,
Branch - provádí testy registru zda jsou kladné nebo záporné, nulové nebo nenulové, sudé nebo liché,

Load and Store - tato instrukce přesouvá 32 bitů nebo 64 bitů, může načíst nebo uložit 8 nebo 16 bitů,

Operate - instrukce pracuje s celočíselnými hodnotami, je plně 64 bitová a podporuje všechny aritmetické operace jako je sčítání, násobení, posun bitu, porovnání.

Memory manager

Architektura zajišťuje

- větší adresový prostor pro data a instrukce
- vhodně a efektivně poskytuje sdílení data a instrukcí
- nezávislé přístupy pro čtení a zápis

Charakteristika

Napájení	3.3 V 5%
Max. pracovní teplota	90 °C
Úložná teplota	od -55 °C do 125 °C
Ztrátový výkon	
pro 200 MHz	24 W
pro 233 MHz	28 W
pro 275 MHz	33 W
pro 300 MHz	36 W
Pouzdro	431 pinové keramické pouzdro s technologií 0.1 " a s rozšířenou tepelnou technologií

6.4.2 Procesor ALPHA 21164PC

Mikroprocesor ALPHA 21164PC je superscalární procesor, který je založen na úspěšném procesoru ALPHA 21164 firmy Digital. Tato architektura procesu poskytuje real-time video a dekódování MPEG 2 bez potřeby speciálních přídavných zařízení, což zvyšuje současný standart procesoru. Multimediální aplikace, které běží na tomto procesoru mají, vysoce kvalitní obraz a 3D grafiku. Tento procesor obsahuje **motion video instruction (MVI)**, které tento procesor posouvají za hranice tohoto tisíciletí.

Čip je vytvořen technologií 0.35 (m s použitím CMOS).

Základní rysy

- 64 bitová RISCová architektura
- frekvence 400 MHz až 533 MHz
- Superscalární čtyř cestný instrukční tok
- možnost provádět až 2.1 BIPS
- 2 cache implementovány přímo na čipu a to 16 kB L1 instrukční cache a 8 kB datová cache
- podpora L2 cache v rozsahu od 512 kB do 4 MB
- 3.3 V externí rozhraní a 2.5 V interní rozhraní
- 413 výstupů z toho je 264 funkčních, 7 pinů je nevyužito, 46 se používá pro externí napájení, 22 pro interní napájení a 74 je použito pro zemnění.

Mikroarchitektura

Blokové schéma (které je uvedeno níže) se skládá z těchto hlavních jednotek, které řídí celý procesor.

Instruction Fetch/Decode and Branch Unit (IDU) - hlavní funkci této jednotky je řídit a rozepisovat instrukce pro IEU, MTU a FPU. Dále pak musí řídit i instrukcí cache.

Přerušení - IDU rozlišuje tři druhy přerušení

- Hardwarové přerušení - je sedm druhů signálu, které zapříčiňují toto přerušení
- Softwarové přerušení - procesor rozeznává 15 prioritních přerušení, které jsou uloženy v interním registru procesoru (IPR)
- systém asynchronních pastí - (AST) 4 druhy těchto přerušení, které jsou ovládány pomocí interního registru

Integer Execution Unit - jednotka obsahuje dva 64-bitové celočíselné *pipeline*, které obsahují dvě adresy, dva logické boxy, válcový posun a násobení.

Floating-Point Execution Unit - 32 vstupních registrů pro operace s pohyblivou řadovou čárkou. Všechny registry jsou 64 bitové. FPU je schopna zpracovat dvě instrukce v každém cyklu.

Memory Address Translation Unit - jednotka obsahuje 3 hlavní části

- Data translation buffer
- Miss address file
- Write buffer address file

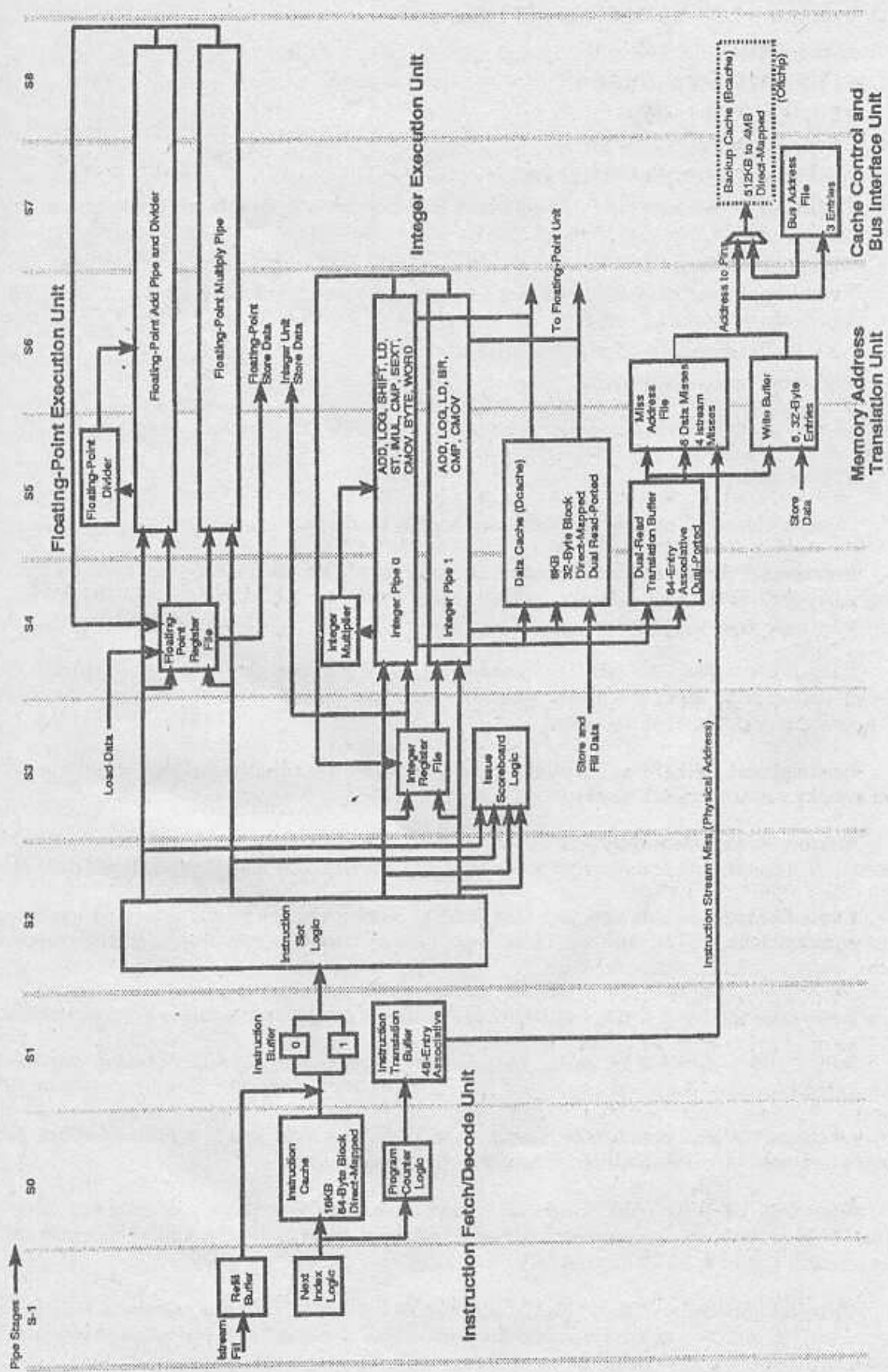
Jednotka přijímá každý cyklus 2 virtuální adresy z IEU. *Translation buffer* pak generuje odpovídající fyzickou adresu a ovládá přístup k informacím pro každou virtuální adresu. Procesor implementuje 43 bitovou virtuální adresu, 40 bitovou necachovatelnou fyzickou adresu a 33 bitovou cachovatelnou adresu.

Cache control and bus interface unit (CBU) - řídí všechny procesy, které jsou vyslány MTU a implementuje všechny paměťové funkce, zvláště pak udržuje protokol funkcí pro *write-back* caching. Ovládá všechny instrukce, primárně pak čtení z data cache a zápis ze čtecího bufferu do spojitě sdílené paměti podsystému. Jednotka také ovládá 128 bitovou obousměrnou datovou sběrnici, adresovou sběrnici a I/O operace.

Organizace paměti cache - procesor obsahuje dvě cache paměti a to primární L1 datou cache a L1 instrukční cache. Obě paměti jsou statické s CMOS strukturou. Procesor ještě může ovládat externí L2 cache, která je umístěna na základní desce.

Data cache - umožňuje duální čtení, zápis pouze jednoduchý. Cache je 8 kB *write-through*, *read-allocate* a přímo mapována pomocí 32 bitových bloků.

Instruction cache - je 16 kB, virtuální, přímo mapovatelná pomocí 64 bitových bloků a 32 bitovou výplni.



Obr. č. 6.7 Blokové schéma:

6.4.3 Procesor ALPHA 21164 (266-333 MHz)

- Základní rysy
- 64 bitová RISCová architektura
- 0.5 μm CMOS technologie
- frekvence 266 MHz až 300 MHz
- Superscalární čtyř cestný instrukční tok
- 9 *pipeline* pro operace s čísly s pohyblivou řadovou čárkou a 7 *pipeline* pro čísla s pevnou řadovou čárkou
- možnost provádět až 1.2 BIPS
- 3 cache implementovány přímo na čipu a to 8 kB L1 instrukční cache, 8 kB datová cache a 96 kB trojcestná *write-back* L2 cache
- 3.3 V externí rozhraní a 2.5 V interní rozhraní
- 499 pinové keramické pouzdro
- procesor tvoří 9.3 miliónu tranzistorů

Mikroarchitektura

Procesor obsahuje 5 nezávislých částí, které řídí běh procesoru.

Instruction Fetch/Decode and Branch Unit - jednotka přijímá, dekóduje a rozepisuje instrukce pro další jednotky jako je IEU, FPU, MTU. Jednotka po přijetí instrukce tuto instrukci dekóduje do 4 paralelních instrukcí a pak otestuje, jestli systém je připraven.

Integer Execution Unit (IEU) - obsahuje dvě 64 bitové celočíselné *pipeline*. Výsledek většiny celočíselných operací je dostupný pro použití následující instrukce. Jednotka také částečně provádí paměťové operace, které souvisí s číselnými operacemi

FloatingPoint Unit (FPU) - jednotka obsahuje *pipeline* pro násobení a sčítání. Dělení se provádí pomocí sčítačky, ale nemá vlastní *pipeline*.

MemoryManagement Unit (MTU) - jednotka provádí všechny operace, které souvisí s *Load/Store* instrukcemi. Dvě *Load* instrukce mohou být obslouženy najednou. Jednotka má na starost i řízení datové cache.

Cache Control and Bus Interface Unit (CBU) - provádí všechny přístupy s MTU a implementuje všechny paměťové funkce. Musí zajišťovat i koherenci s funkcemi cache a zpřístupňuje L2 cache a nepovinnou L3 cache, které je uložena na základní desce.

Instruction Cache - instrukční cache je 8 kB virtuální s přímým mapováním a 32-bitovými bloky.

Data Cache - datová 8 kB cache, která je *write-through, read-allocate* s přímým mapováním s 32-bitovými bloky.

L2 Cache - 96 kB 3 cestná cache, která podporuje oba typy bloků a to 32-bitové i 64-bitové. Je plně zřetězená s podporou *write-through, read-allocate* a přímým mapováním.

Nepovinná L3 cache (není implementována přímo na čipu) - procesor podporuje tuto cache s 32 bitovými i 64 bitovými bloky. Paměti mohou být jak asynchronní, tak synchronní RAM. Velikost této cache je v tomto rozmezí: 1, 2, 4, 8, 16, 32 nebo 64 MB.

Virtuální adresace - velikost virtuální adresy je 64 bitové celé číslo bez znaménka. Mikroprocesor implementuje 43 bitovou adresu, která je podмноžinou této adresy. Procesor přímo podporuje 40 bitovou adresu k adresaci 1 terabytu.

Instrukční sada

Všechny instrukce jsou 32 bitové a využívají jeden ze čtyř formátů instrukce. Každá instrukce je tvořena 6 bitovým operačním kódem a žádným, jedním, dvěma, třemi nebo pěti bitovými položkami.

CALL_PAL - tento instrukční vektor provádí privilegované a neprivilegované instrukce softwarové úrovně.

Branch - provádí testy registru zda jsou kladné nebo záporné, nulové nebo nenulové, sudé nebo liché.

Load and Store - tato instrukce přesouvá 32 bitů nebo 64 bitů, může načíst nebo uložit 8 nebo 16 bitů.

Instrukce podporuje přímo na vstupu *byte* a *word*, které jsou přímo implementovány v hardwaru.

Operate - instrukce pracuje s celočíselnými hodnotami je plně 64 bitová a podporuje všechny aritmetické operace jako je sčítání, násobení, posun bitu, porovnání.

6.4.4 Procesor ALPHA 21164 (366-600 MHz)

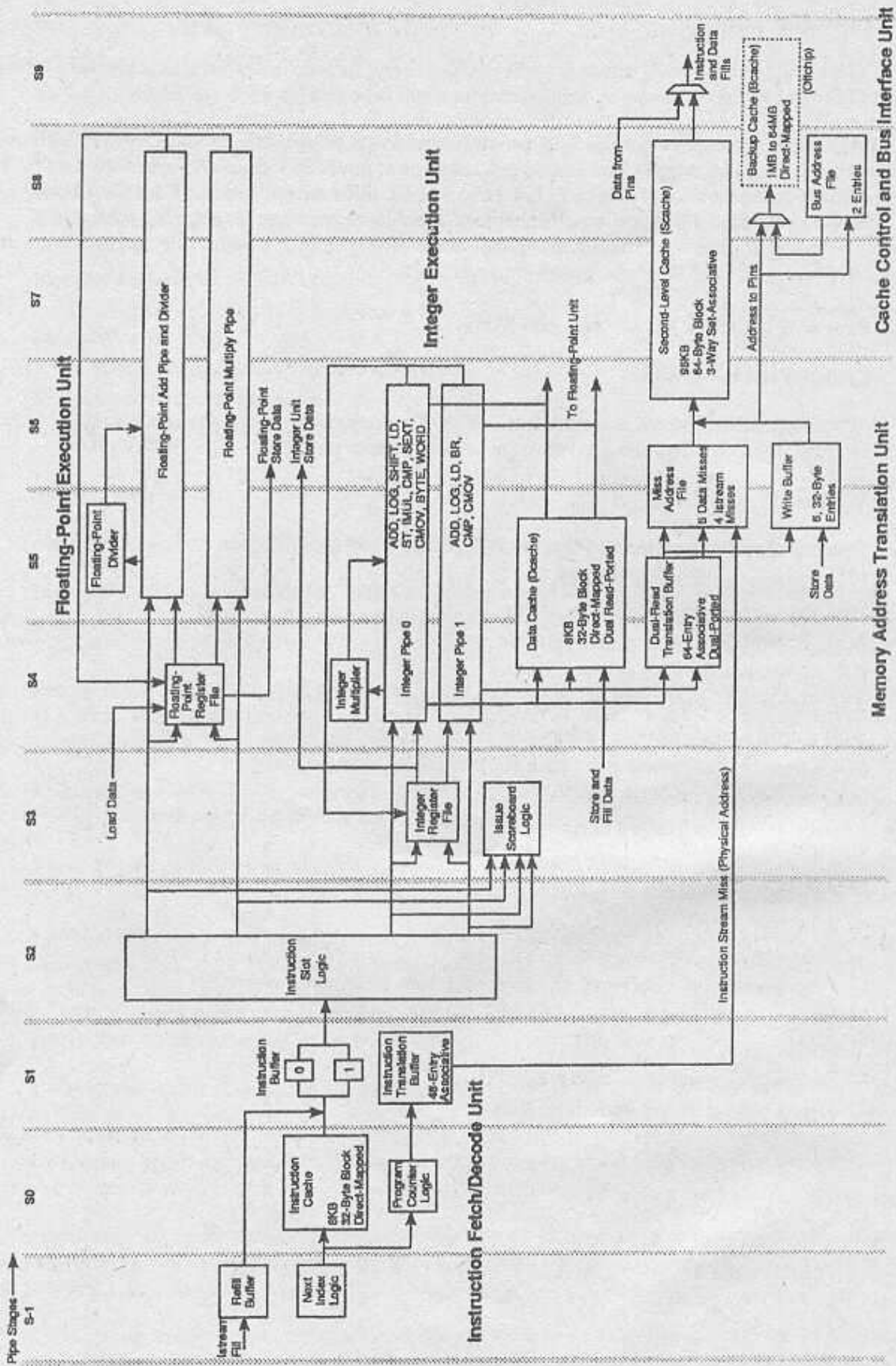
Základní charakteristika

Mikroprocesor 21164 má superskalární design, který umožňuje provádět čtyři instrukce v každém hodinovém cyklu. Integrace instrukční a datové cache na procesoru poskytuje velmi vysoký výkon.

Vlastnosti

- Podpora operačních systémů: Windows NT, Digital Unix, Open VMS
- Pracovní frekvence 366-600 MHz
- Plně zřetěžená 64-bitová RISC architektura
- Superskalární (4 - cestné zpracování instrukcí)
- 0.35 (m technologie CMOS
- 8 kB instrukční cache s přímým mapováním na čipu
- 8 kB datová cache s přímým mapováním (write through) na čipu
- 96 kB třicestná asociativní write back (L2) instrukční a datová cache
- Jednotka pro řízení paměti
- Sériový ROM interface pro inicializaci
- Flexibilní velmi výkonný interface
 - 128-bitová paměťová datová cesta
 - 3.3V I/O
 - Volitelný ECC korekční kód
 - Programovatelný systémový interface; od 1/3 do 1/15 frekvence hodiny
 - Možnost instalace L3 cache od 1 MB do 64 MB, podpora synchronních a asynchronních SRAM
- Rozšíření
 - Podpora datových typů WORD a BYTE
 - Napětí jádra 2.0V pro snížení spotřeby

Dále následuje obrázek č. 6.8



Obr. č. 6.8

1. Instructon Fetch/Decode and Branch Unit (IDU)

Hlavním významem této jednotky rozdělovat tok instrukcí do jednotek IEU, MTU a FEU. Také má na starosti instrukční cache.

1.1 Instruction Prefetch and Decode

IDU zpracovává jen ZAROVNANÉ (naturally aligned) skupiny po čtyřech instrukcích (INT16). Pokud se vyskytne skok doprostřed takto zarovnaných instrukcí, IDU se snaží zpracovat instrukce, na které je odskok prováděn na konci právě zpracovávané skupiny. Proto je velmi důležité mít instrukce předzpracované, aby bylo dosaženo optimálního výkonu.

1.2 Branch Prediction

Tato jednotka umožňuje předpovídat instrukce skoku před jejich načtením a díky tomu upravit další zpracování instrukcí.

1.3 Instruction Translation Buffer

IDU obsahuje 48 vstupních bufferů pro překládání instrukcí (instruction translation buffer - ITB). Buffer ukládá naposledy prováděné instrukce, a chráněné informace pro paměťové stránky od 8 kB do 512 kB.

1.4 Interrupts (přerušení)

- Hardwarová přerušení - zdrojem těchto přerušení jsou signály procesoru
- Softwarová přerušení - existuje 15 priorit pro vykonávání přerušení
- Asynchronní přerušení systému

Většina přerušení je maskovatelná pomocí registrů.

2. Integer Execution Unit

IEU obsahuje dvě 64-bitové prováděcí jednotky E0 a E1, tyto jednotky obsahují:

- Dvě sčítačky
- Dva logické operátory
- Bubnový posunovač (barrel shifter)
- Logiku pro práci s datovým typem BYTE
- Celočíslnou násobičku

3. Floating Point Execution Unit

Tato jednotka zpracovává instrukce s plovoucí desetinnou tečkou. Obsahuje tyto části:

- 64-bitové pole registrů (obsahuje 32 registrů)
- Ovládací registr
- Paralelní násobička pro čísla s plovoucí desetinnou tečkou.
- Paralelní sčítačka, dělička

Jednotka FPU umožňuje zpracovat jen 2 instrukce v každém cyklu, s výjimkou dělení.

4. Memory Adress Translation Unit

Obsahuje tři základní sekce:

- Datový překládací buffer (data translation buffer)
- Seznam chybějících adres (miss address file - MAF)
- Zápisový buffer (write buffer address file)

Jednotka MTU může obdržet až dvě virtuální adresy v jednom cyklu hodiny. Překládací buffer vygeneruje odpovídající fyzickou adresu. Mikroprocesor 21164 implementuje 43-bitové virtuální adresy a 40-bitové fyzické adresy.

5. Cache Control and Bus Interface Unit (CBU)

Zpracovává všechny adresy posílané do MTU a implementuje veškeré paměťové operace. Ovládá L2 cache případně i BackUp cache. CBU také ovládá 128-bitovou obousměrnou datovou sběrnici, adresovou sběrnici a ovládá vstup/výstup.

6. Cache organization

Mikroprocesor 21164 má tři různé cache na čipu - primární L1 datová cache, primární L1 instrukční cache a L2 kombinovaná datová a instrukční cache.

7. Serial Read-Only Memory Interface

SROM poskytuje inicializační služby, např. diagnostiku hardware.

8. Pipeline Organization

21164 má 7-mi stavovou pipeline pro celočíselné operace a instrukce pracující s pamětí, a 9-ti stavovou pipeline pro operace s plovoucí řádovou tečkou.

6.4.5 Procesor ALPHA 21264

Základní charakteristika

Mikroprocesor 21264 je třetí generací 64-bitové architektury Alpha. Obsahuje nejnovější rozšíření (např. instrukce pro práci s obrazovými daty - motion/video instructions).

Aplikace

- Podporuje Windows NT
- Umožňuje spouštět aplikace z Windows 95 a nenativní Windows NT aplikace pomocí emulátoru DIGITAL FX!32
- Podporuje Digital Unix - první 64-bitový unix
- Podporuje operační systémy: OpenVMS, VxWorks a Linux

Vlastnosti

- 15.2 miliónů tranzistorů
- pracovní frekvence 500 MHz a vyšší
- 0.35 (m technologie CMOS
- Plně zřetěžená 64-bitová RISC architektura
- Čtyři jednotky pro zpracování celočíselných instrukcí a dvě paralelní jednotky pro zpracování instrukcí s plovoucí desetinnou tečkou
- 64 kB L1 instrukční cache
- 64 kB L1 datové cache
- Ztrátový výkon 60W
- Implementace Motion Video Instructions pro podporu práce s video daty

Čtyři instrukce v jednom cyklu

Jádro mikroprocesoru 21264 je vysoce paralelní a dokáže zpracovat až šest instrukcí v jednom cyklu hodin, a trvale je schopno zpracovávat čtyři instrukce v jednom cyklu. 21264 má čtyři jednotky pro práci s celočíselnou aritmetikou, z čehož dvě jsou všeobecné jednotky a dvě jsou aritmeticko logické jednotky (ALU). ALU provádí jednoduché aritmetické a logické operace, všeobecná jednotka provádí také aritmetické a logické operace a navíc provádí skoky a posuny. Jedna celočíselná jednotka obsahuje násobičku, ostatní zajišťují zpracování MVI (motion video instructions).

Motion Video Instructions

MVI je speciální sada instrukcí, která umožňuje zpracovávat obrazová data rychleji než konvenční instrukce. Pomocí MVI je mikroprocesor 21264 schopen v reálném čase kódovat nebo dekódovat video formátu MPEG2. Sada instrukcí MVI se odlišuje od ekvivalentních instrukcí jiných výrobců (MMX, VIS) v tom, že je v jistém smyslu minimální a vysoce paralelní.

Pipelined Primary Cache

Architektura cache je v této verzi mikroprocesoru ALPHA přepracována. Primární cache byla zvětšena z původních 8 kB (u 21164) na 64 kB a to jak u datové tak instrukční cache. Díky tomu dosahuje 21264 vyšší rychlosti zpracování (není třeba tak často přistupovat k hlavní paměti).

Advanced Branch Prediction

Při predikci skoků mikroprocesor 21264 používá dvou metod. Mezi těmito metodami si vybírá dynamicky takovou, která více odpovídá danému případu. Díky takto vyspělé predikci skoků dochází k chybnému určení skoku jen v 7-10 případech z tisíce instrukcí. Těto přesnosti se také dosahuje pomocí 35 kB paměti, která obsahuje informace o minulých predikcích.

Double-Speed Data Cache

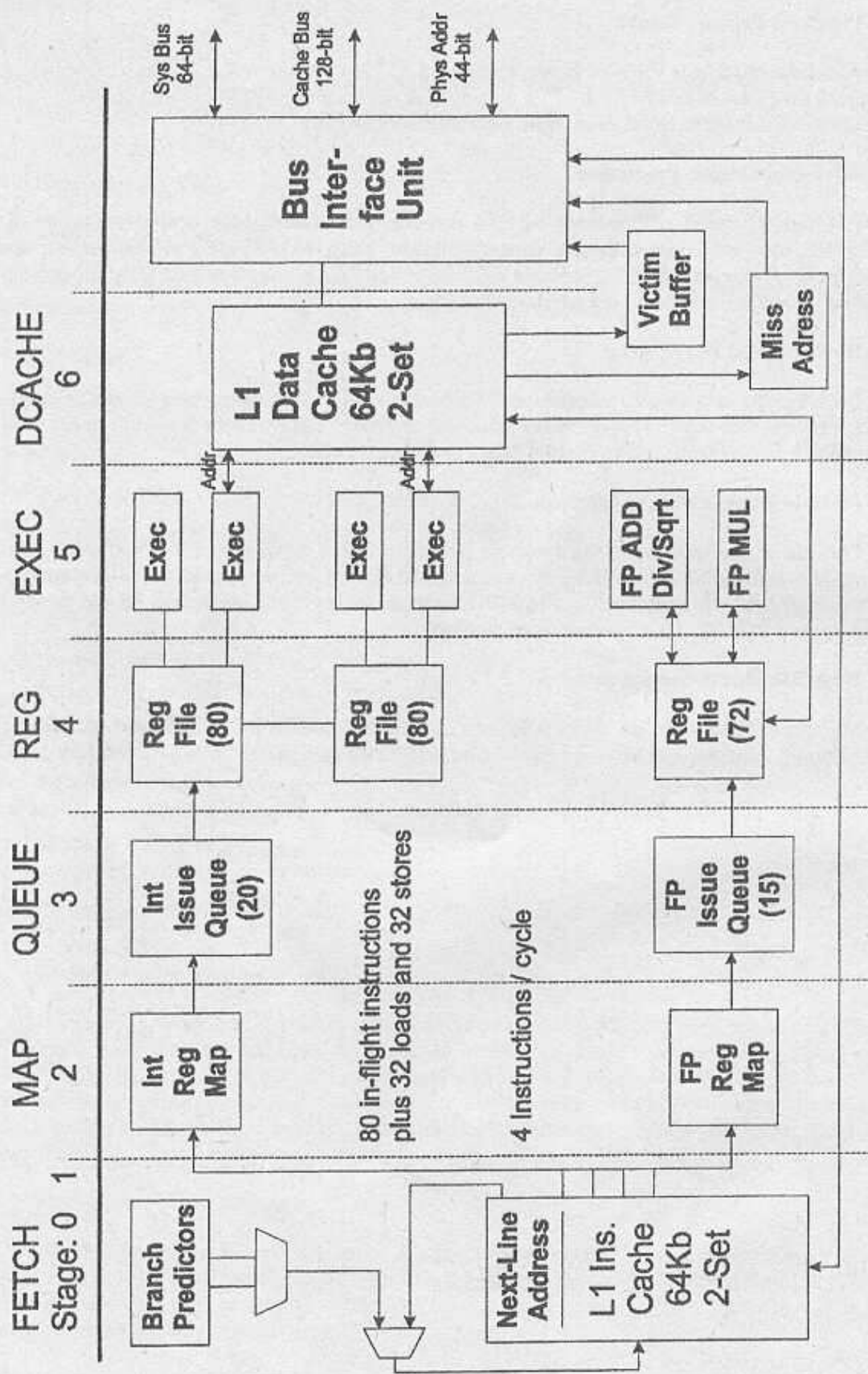
Datová cache na procesoru má přístupovou dobu 2ns, navíc má přístupové porty duální, takže umožňuje načíst dvě nezávislá datová slova (64-bitové) v jednom cyklu hodin. Datová cache umožňuje načíst další adresu v každém taktu hodin dvakrát, tzn. že pracuje s daty na frekvenci 1Ghz.

Flexible External-Cache Interface

Přestože je datová a instrukční cache na procesoru 21264 největší ze všech komerčně vyráběných mikroprocesorů, není nutné pro zvýšení výkonu používat externí cache. Externí cache je přímo ovládána mikroprocesorem a má maximální velikost 16 MB. Externí cache je vlastně synchronní SRAM paměť, pracuje na frekvenci rovné 2/3, 1/2, 1/3 a 1/4 frekvence procesoru.

High-Bandwidth System Bus

Nejjednodušší způsob, jak zvýšit průchodnost sběrnice je udělat ji 128-bitovou, tato sběrnice byla u 21264 použita. Při rychlosti 333 MHz je tato 128-bitová sběrnice schopna přenášet data rychlostí 2.0 GB/s.



Obr. č. 6.9 Blokové schéma mikroprocesoru 21264

6.5 Procesory PowerPC

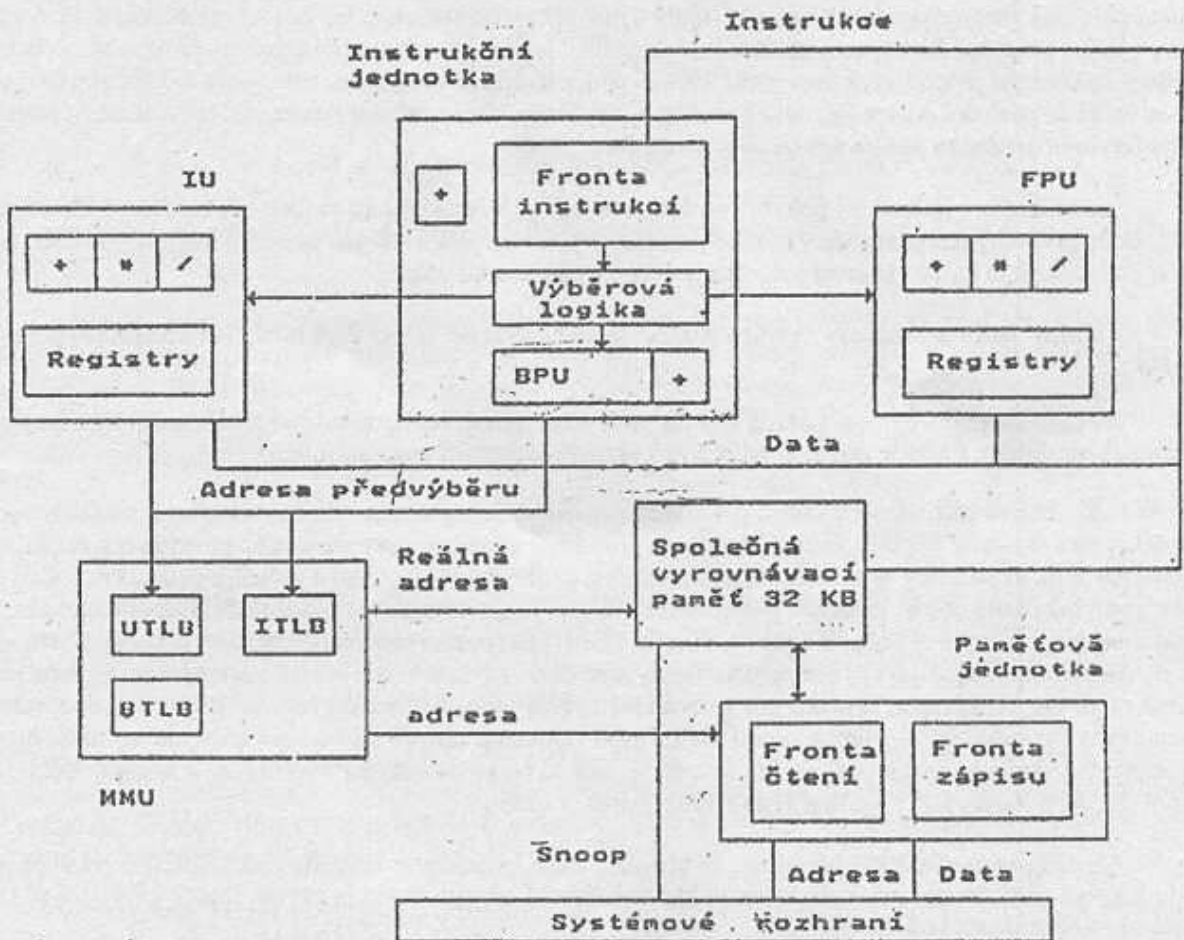
RISC procesory s názvem PowerPC byly uvedeny na trh již od roku 1991 jako výsledek úsilí firem IBM, Apple a Motorola [3, 19, 39].

Jak sám název naznačuje, přebírají tyto procesory základní koncepce předchozího procesoru POWER firmy IBM. Současně využívají některé úspěšné rysy procesoru MC 88100 firmy Motorola. Cílem projektu bylo vytvořit levný, vysoce výkonný procesor.

První model PowerPC 601, který uvedla do výroby firma IBM, používá vnější adresovou sběrnici šířky 32 b a vnější datovou sběrnici šířky 64 b. Vnitřní přenosy dat v procesoru se uskutečňují na šířkách od 32 do 128 b.

6.5.1 Procesor PowerPC 601

Struktura procesoru PowerPC601 je schématicky znázorněna na obr. č. 6.10.



Obr. č. 6.10 PowerPC 601

Máme zde tři výkonové jednotky, které jsou schopny pracovat současně. Jsou to:

- jednotka pro řízení skoků BPU
- celočíselná jednotka IU
- jednotka pro operace s pohyblivou čárkou FPU

Kromě toho jsou na čipu umístěny též:

- instrukční jednotka obsahující frontu instrukcí a BPU
- paměťová jednotka
- vyrovnávací paměť
- jednotka řízení paměti MMU
- systémové rozhraní

Instrukční jednotka má za úkol zajistit, aby všechny tři výkonné jednotky byly stále zaměstnány, protože procesor je schopen **začít zpracovávat až tři instrukce v jednom taktu**.

Instrukce vybírá z vyrovnávací paměti na čipu a zařazuje je do své fronty instrukcí. Tuto frontu udržuje instrukční jednotka neustále naplněnou, doplnění je prováděno průběžně v každém taktu. Není-li požadovaná instrukce přítomna ve vyrovnávací paměti, čeká instrukční jednotka s vysláním požadavku na čtení z hlavní paměti až do chvíle, kdy se fronta instrukcí vyprázdní nejméně do poloviny.

Fronta instrukcí je rozdělena do dvou polovin, které se navzájem liší svými operačními možnostmi. **Horní polovina** funguje pouze jako paměť fronty a instrukce se v ní mohou jen posouvat směrem dolů. Naproti tomu v **dolní polovině fronty** se instrukce mohou jednak posouvat směrem dolů, jednak mohou být vybrány a předány výkonným jednotkám k provedení. Přitom pouze instrukce určené pro celočíselnou jednotku se musejí posouvat až do poslední pozice Q_0 , odkud mohou být vybrány. Tato poslední pozice je totiž současně **vybavena dekodérem operačního znaku** pro tento typ instrukce.

Instrukce pro jednotku s pohyblivou čárkou a jednotku řízení skoků mohou být vybrány z libovolného místa dolní poloviny a předány do výkonné jednotky, vybírá se však vždy jen poslední instrukce daného typu. Další podmínkou je to, že výkonná jednotka je v daném okamžiku volná.

Tímto způsobem lze tedy vybrat v jednom taktu až tři instrukce ze **čtyř míst tvořících dolní polovinu fronty**.

Výběr instrukcí v jiném pořadí, než jak jsou uloženy v paměti, nutně vyvolává riziko nepřipustného zásahu do algoritmu. Proto je procesor MPC610 vybaven pojistným mechanismem.

Jak instrukční jednotka sama, tak i jednotka BPU, je vybavena vlastní sčítačkou, používanou při výpočtu adres skoku. V případě, že BPU dostane k provedení **nepodmíněný skok** nebo **podmíněný skok**, jehož podmínka je již vypočtena, vybere následující instrukci z cílové adresy skoku a zařadí ji do fronty. Pokud se v dolní polovině fronty objeví instrukce podmíněného skoku, je přesunuta do jednotky BPU bez ohledu na to, zda podmínka skoku již byla vypočtena nebo ne. Jednotka BPU pak provádí predikci cílové adresy skoku a rozhoduje tak o tom, která instrukce má být zařazena do fronty jako další. Při tom se řídí jednak statickým pravidlem, podle něhož preferuje skoky zpět, typické pro **provádění cyklů**, jednak **bítem predikce**, jehož hodnotu nastavil kompilátor. Pravidla práce s instrukcemi skoku jsou doplněna tak, že žádná instrukce, která následuje za podmíněným skokem, nesmí psát do registru dříve, než je podmínka skoku vypočtena, a navíc do této doby nesmí být do jednotky BPU vyslána žádná další instrukce skoku.

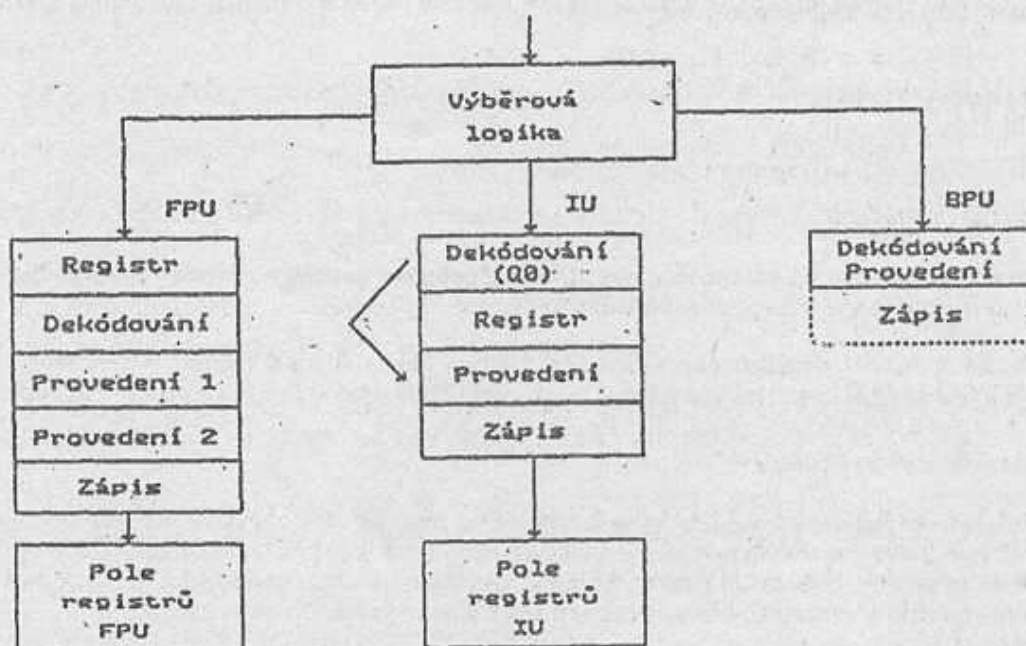
Jestliže jednotka BPU předpoví, že program bude pokračovat lineárně (bez skoku) a po vypočtení podmínky se zjistí, že ke skoku přece jen dojde, znehodnotí se jako obvykle zbytek fronty a nahradí se nově přečtenými instrukcemi z VP.

Pokud naopak BPU rozhodne, že bude následovat skok, zmrazí provádění všech instrukcí, které následují za skokem a přečte z VP instrukce z adresy, na kterou ukazuje skok.

Po vypočtení podmínky skoku se zmrazené instrukce buď vymažou anebo nahradí nově přečtenými instrukcemi.

Jednotka BPU je jednoúrovňová, takže zpracování převážné většiny v ní trvá jeden takt.

Jednotky IU a FPU pracují proudově a jsou proto rozděleny do sekcí, IU do **tří** a FPU do **pěti** (viz obr. č. 6.11.).



Obr. č. 6.11 Činnost tří výkonných jednotek procesoru PowerPC 601

Celočíselná aritmetika provádí všechny celočíselné operace, operace s bitovými poli a zápisy do paměti a čtení z ní.

Moduly zřetěženého zpracování instrukce v celočíselné jednotce jsou: dekodování, provedení a zápis výsledku. Dekodování operačního znaku se provádí v poslední pozici fronty instrukcí. V jednotce IU jsou proto realizovány pouze další dva moduly. Před modulem, v němž se provádí operace, je vložen registr, v němž může být instrukce zdržena v případě, že modul dosud není volný. V jednotce IU je obsažena ALU, násobička, dvoubránové pole 32 univerzálních registrů a registr výjimek.

Do univerzálních registrů mohou mít přístup dvě instrukce současně, např. jedna z jednotky IU a druhá z vyrovnávací paměti.

Jednotka FPU obsahuje sčítačku s dvojnásobnou přesností, násobičku s jednoduchou přesností, děličku a pole registrů. Toto pole registrů je také dvoubránové, avšak není vybaveno schopností přímého předávání operandu. Jednotka FPU je tvořena pěti moduly zřetěženého zpracování. První je čekací registr, pak následuje dekodování, dva prováděcí moduly a zápis výsledku.

Poměrně velká vyrovnávací paměť (32 kB) je určena pro instrukce i pro data. Takové řešení je z hlediska nároků na plochu čipu levnější než dvě oddělené VP a navíc umožňuje dynamicky nastavit rovnováhu mezi rozsahem paměti vyhrazené pro data a pro instrukce. Na druhé straně to však brání procesoru v tom, aby četl instrukce a data v jednom cyklu.

Vyrovnávací paměť je vybavena asociativním adresářem se stupněm asociativity $s=8$. Jeden řádek má délku 64 B a je rozdělen do dvou sektorů po osmi slovech. Do adresáře se zapisují reálné adresy získané překladem v MMU.

Konzistenci obsahu VP a hlavní paměti zajišťují dvě strategie: implicitně se používá úklid při výměně, ale pro jednotlivé řádky lze nařadit společný zápis do VP a HP.

Pro uvolňování bloku se používá strategie LRU.

Jednotka řízení paměti MMU překládá virtuální adresy generované procesorem na fyzické adresy používané při adresování HP. Logický adresový prostor má kapacitu 4 GB a je rozdělen do segmentů po 256 MB. Stránky mají velikost 4 kB a mohou být sdruženy do bloků o velikosti 128 kB - 8 MB.

Stránkovací tabulka je založena jako obvykle v HP, překlad se urychluje pomocí tabulek TLB, které jsou tři.

1. Tabulka UTLB (unified translation lookaside buffer) obsahuje 256 adres stránek programu i dat a je částečně asociativní.
2. Tabulka BTLB (block TLB) zaznamenává čtyři adresy bloků.
3. Tabulka ITLB (instruction TLB - plně asociativní) zaznamenává čtyři posledně překládané adresy instrukcí.

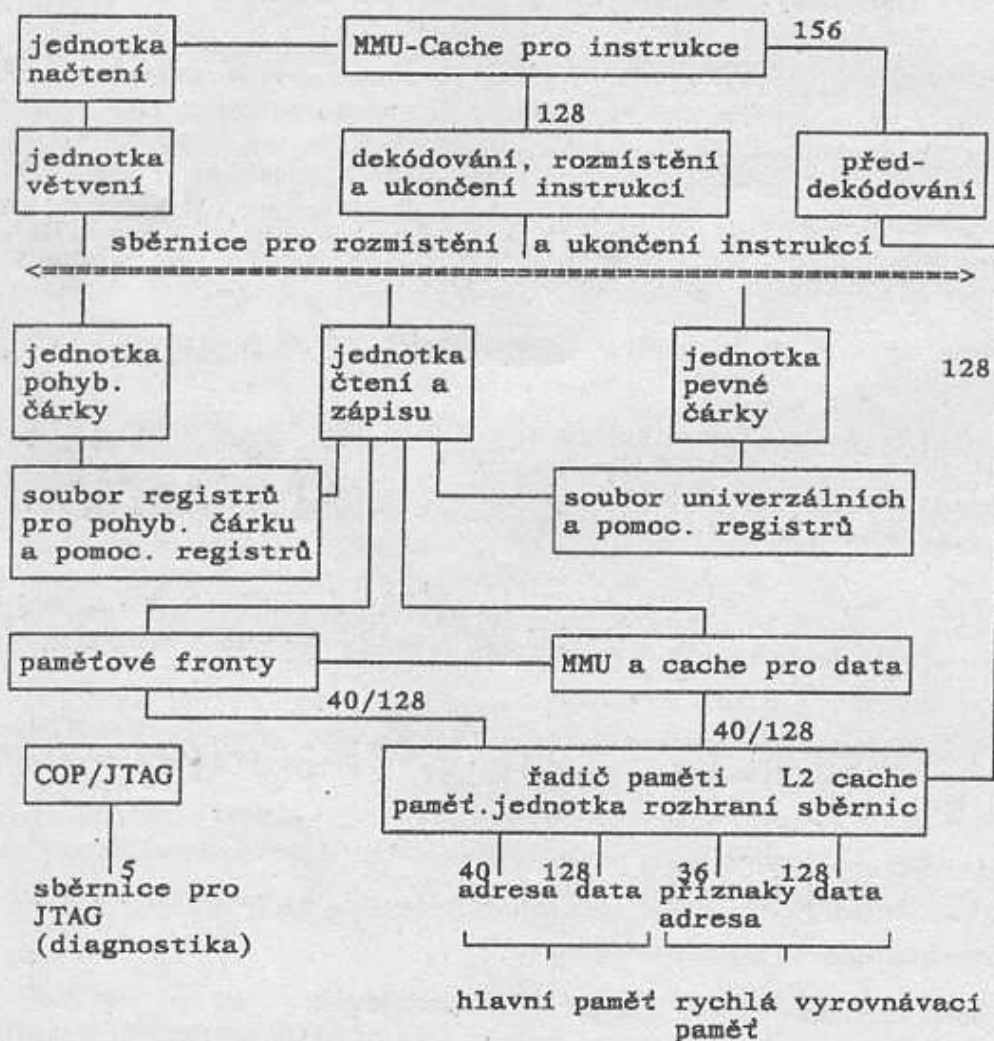
Kromě překladu adres zajišťuje jednotka MMU též ochranu paměti na úrovni stránky, bloku nebo segmentu.

Procesor MPC601 je realizován na čipu 11x11 mm a obsahuje cca 2.8 miliónu tranzistorů. Použitá technologie je 0.65 μm CMOS se čtyřmi vrstvami metalizace. Při hodinové frekvenci 66 MHz má příkon 9 W.

6.5.2 Procesor PowerPC 620

Jedná se o procesor, který bude základem výrobní řady serverů a výkonných pracovních stanic. Mikroprocesor je vybaven rozsáhlým paměťovým subsystémem s vysokou průchodností dat, podporujícím plně symetrický multiprocessing, zpracování transakcí a numericky náročné výpočty. Model 620 je první 64bitovou implementací architektury PowerPC, která dovoluje provoz 32bitových i 64bitových aplikací.

Provedení čipu a topologie uspořádání jednotlivých bloků mikroprocesoru je na obr. č. 6.12.



Obr. č. 6.12 PowerPC 620

Pracovní frekvence čipu je 133 MHz. Mikroprocesor je vybaven prostředky pro redukci příkonu v klidovém stavu.

Počet tranzistorů je cca. 7 miliónů a je použita technologie 0.5 μm CMOS se 4 metalickými vrstvami.

Nárůst počtu tranzistorů je výsledkem zařazení kvalitativně nových komponent:

- 64bitová architektura
- vnitřní rychlé vyrovnávací paměti (cache) pro data a pro instrukce, každá o kapacitě 32 kB a s vlastním řízením MMU
- rozšířené prostředky pro předpověď větvení (branch prediction)
- posílené vstupně-výstupní sběrnice (128 bitů pro data)

Výkonnost procesoru je určena 3 parametry - počtem instrukcí v úloze, průměrným počtem instrukcí, které procesor provádí v jednom cyklu (IPC) a pracovní frekvencí procesoru.

První parametr je minimalizován díky architektuře PowerPC, která je optimální z hlediska použitého strojového kódu. Zvýšení druhého parametru zajišťuje dále popsána vnitřní struktura procesoru.

Třetí parametr je určen vhodně zvolenou technologií.

Důležité je optimálně sladit všechny 3 parametry.

Zřetězené zpracování instrukcí (pipeline) má 5 stupňů: načtení (fetch), rozmístění (dispatch), provedení (execute), ukončení (complete) a zpětný zápis (write back).

V rámci superskalární architektury může současně pracovat 6 nezávislých jednotek: jednotka větvení, dvě jednoduché jednotky pevné čárky, jedna komplexní jednotka pevné čárky, jednotka pohyblivé čárky a jednotka čtení a zápisu viz. obr. č. 6.7.

Aby bylo možno co nejvíce využít současného provádění více instrukcí, používá se předpověď skoků, předčítání instrukcí (prefetching) a jejich předběžné (spekulativní) provádění, instrukce jsou rozmísťovány do jednotlivých jednotek k provádění v pořadí svého umístění v programu. Jejich provádění však probíhá mimo toto pořadí, ale ukončení je opět podle pořadí.

Model 620 používá statickou i dynamickou předpověď skoků. Dynamická předpověď se uplatňuje při načítání a rozmísťování instrukcí.

Podmíněné zpracování instrukcí úzce souvisí s řešením skoků. Instrukce, které navazují na předpovězený výsledek skoku je možno podmíněně provést a výsledky uložit do dočasných pamětí. Po definitivním vyhodnocení skoku se tyto výsledky buďto použijí nebo se zruší. Model 620 dovoluje současně rozpracovat předběžné zpracování až pro 4 nevyřešené skoky.

Charakteristika jednotlivých jednotek mikroprocesoru modelu 620. Dvě jednotky pevné čárky jsou určeny pro instrukce, které jsou hotovy během jediného hodinového cyklu, zatímco třetí slouží pro ty, které vyžadují ke svému provedení více cyklů. Jednotka pohyblivé čárky pracuje podle normy IEEE 754. Všechny uvedené jednotky mají po 32 univerzálních registrech.

Pro podporu pamětí slouží bloky MMU, cache, paměťové fronty a řadič paměti L2 cache. Blok MMU i cache jsou samostatné pro instrukce a pro data. MMU zpracovává 80bitovou virtuální, 64bitovou efektivní a 40bitovou fyzickou adresu. MMU zajišťuje podporu jak segmentace, tak stránkování paměti. TLB je dvoucestná asociativní paměť se 128 položkami.

Rychlé vyrovnávací paměti (cache) pro instrukce a pro data jsou 8cestné asociativní paměti po 32 kB.

Paměťové fronty jsou dvě: pro zápis a pro čtení.

Zápisová fronta je podstatně komplikovanější a má 6 položek.

Řadič L2 cache podporuje externí vyrovnávací paměť s max. kapacitou 128 MB, implementuje MESI-protokol a generuje 9 bitů ECC pro 128 bitů dat.

6.5.3 Procesor PowerPC 603

Hlavní rysy procesoru PowerPC603 jsou tyto:

Vysoce výkonný, superskalární mikroprocesor

- až tři instrukce zpracované za jeden takt
- až pět instrukcí provedených za jeden takt
- jednotaktové provedení většiny instrukcí
- propojené FPU pro všechny operace s jednoduchou přesností a většinu s dvojitou přesností

Pět nezávislých vykonávacích jednotek a dva soubory registrů

- BPU se statickou předpovědí větvení
- 32-bitová IU
- FPU plně podporuje IEEE 754 pro jak jednoduchou tak i dvojitou přesnost operací
- LSU pro přesun dat mezi datovou cache a GPR a FPR
- SRU které provádí instrukce podmínkových registrů (CR) a zvláštních registrů (SPR)
- 32 GPR pro celočíselné operandy
- 32 FPRS pro operandy jednoduchou nebo dvojitou přesností

Vysoký instrukční a datový průtok

- větvení bez cyklů (skládání větví)
- programovatelná statická předpověď větvení na neurčené podmíněné větve
- instrukční výkonná jednotka schopná vykonat dvě instrukce za jeden takt s instrukční cache
- šesti-vstupová instrukční fronta se schopností předpovědi
- nezávislé propojení se zpětným plněním, které redukuje datové závislosti v hardware 8 kB datová cache - dvoucestná množinově asociativní, fyzicky adresovaná, LRU výměnný algoritmus
- 8 kB instrukční cache - dvoucestná množinově asociativní, fyzicky adresovaná, LRU výměnný algoritmus
- operace cache write-back nebo write-through, programovatelné na blokové nebo stránkové bázi
- BPU, které zajišťuje předpovídající operace CR
- adresové překladové zařízení pro 4 kB velikost stránky, proměnná velikost bloku a 256 MB velikost bloku
- 64-vstupová, obousměrná množinově asociativní ITLB
- 64-vstupová, obousměrná množinově asociativní DTLB
- čtyřvstupá datové a instrukční BAT pole zajišťující bloky od 128 kB do 256 MB
- software tabulkové vyhledávací operace podporované rychlým mechanismem pastí
- 52-bitové virtuální adresy; 32-bitové fyzické adresy

Zařízení ke zvýšení výkonu systému

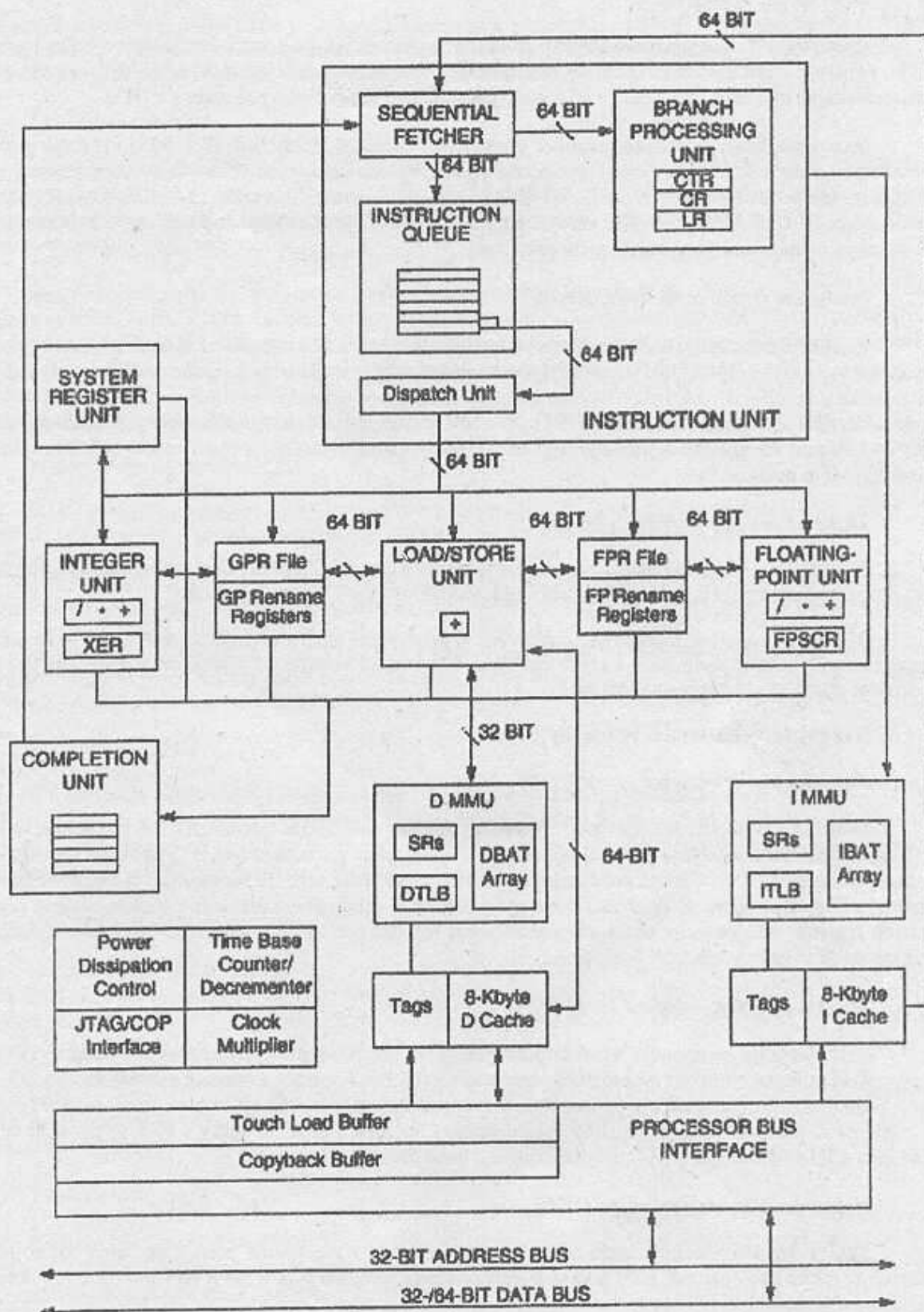
- 32-bitová nebo 64-bitová sdílená vnější sběrnice dat s paralelními přenosy
- podpora pro jednoúrovňové adresy propojení a sběrnice transakce mimo pořadí
- rozšíření sběrnice pro operace rozhraní s přímým ukládáním

Integrovaný Power Managment

- nízkonapěťový 3.3 V design
- vnitřní dělitel hodin procesoru/sběrnice, který poskytuje poměry 1:1, 2:1, 3:1, a 4:1
- tři nízkonapěťové módy : (doze, nap, sleep)
- automatické dynamické snížení odběru, pokud je funkční jednotka v nečinnosti

Vnitřní systémové testovací a ladící schopnosti pomocí JTAG prohlížení hranice.

Blokové schéma je uvedeno na obr. č. 6.13.



PowerPC 603 Microprocessor Block Diagram

Obr. č. 6.13

Instrukční jednotka

Instrukční jednotka procesoru PPC603 se skládá z podávací jednotky, instrukční fronty, odesílací jednotky a BPU. Zajišťuje centralizovanou kontrolu toku instrukcí do vykonávacích jednotek. Instrukční jednotka určuje adresu následující instrukce pro podání z informací, jež obdrží ze sekvenčního podavače a z BPU.

Instrukční jednotka podává instrukce z instrukční cache do fronty instrukcí. BPU vytahuje instrukce větvení z podavače a používá statickou předpověď větvení na neanalyzované podmíněné větve, aby instrukční jednotka mohla podat instrukce z předpokládaného cílového instrukčního toku, až bude podmínková větev vyhodnocena. BPU překládá instrukce větvení pro nepodmíněné větve nebo podmíněné větve nepostihnutelné prováděnými instrukcemi ve vykonávaném propojení.

Instrukční fronta a odesílací jednotka

Instrukční fronta (IQ) udržuje nejméně 6 instrukcí a nahrává až 2 instrukce z instrukční jednotky během jednoho taktu. Instrukční podávací jednotka plynule nahrává tolik instrukcí, kolik jí prostor v IQ dovolí. Instrukce jsou odesílány do příslušných vykonávacích jednotek z odesílací jednotky maximální rychlostí dvě instrukce za takt. Odesílání je usnadňováno do IU, FPU, LSU a SRU rezervační stanicí každé jednotky. Odesílací jednotka zajišťuje kontrolu zdrojového a cílového registru, určuje přesnost odesílání a brání nesekvenčnímu odesílání instrukcí, je-li to požadováno.

Jednotka pro provádění větvení (BPU)

BPU dostává instrukce větvení z podávací jednotky a provádí CR předvidací operace na podmínkových větvích pro rychlou analýzu, dosahující efektu bezcyklové větve v mnoha případech.

BPU používá bit v instrukčním kódování k předvídání směru podmínkové větve. Takže když je nalezena neanalyzovaná podmínková větev instrukcí, 603 podává instrukce z předpokládaného cílového toku, dokud není podmínková větev analyzována.

Nezávislé vykonávací jednotky

Jednotka pro celočíselné operace (IU)

Jednotka pro celočíselné operace (IU) provádí všechny celočíselné instrukce. IU vykonává právě jednu celočíselnou instrukci, zajišťuje počítání její aritmeticko-logickou jednotkou (ALU), registrem pro násobení, dělení a podmínky (XER). Většina celočíselných instrukcí trvá jeden takt. 32 univerzálních registrů slouží pro podpůrné celočíselné operace. Zpoždění skrze spory s GPR je minimalizováno automatickou alokací přejmenovacích registrů. 603 zapisuje obsah přejmenovacích registrů pro zaměstnání GPR, když jsou celočíselné instrukce uvolňovány ukončovací jednotkou.

Jednotka pohyblivé řádové čárky (FPU)

FPU obsahuje jednoduché vícevstupové pole a registr stavu pohyblivé čárky a ovládání (FPSCR). Vícevstupové pole povoluje 603 rychle implementovat vícenásobné operace a operace vícenásobného vkládání. FPU je propojeno, takže instrukce s jednoduchou a dvojitou přesností mohou být vybaveny zároveň. 32 registrů pro pohyblivou řádovou čárku je zajištěno pro podpůrné operace. Zpoždění spory s FPR je minimalizováno stejně jako u IU a rovněž tak i uvolňování instrukcí z ukončovací jednotky.

Nahrávací/Ukládací jednotka (LSU)

LSU vykonává všechny nahrávací a ukládací instrukce a zajišťuje přenos dat mezi GPR, FPR a podsystémy cache nebo paměti. LSU počítá efektivní adresy, provádí řazení dat a sekvencování pro nahrávací/ukládací a vícenásobné instrukce.

Jednotka systémového registru (SRU)

SRU vykonává různé instrukce na úrovni systému včetně logických operací podmínkového registru a instrukce přenosu do/z registrů zvláštního určení. Pro udržování stavu systému, většina instrukcí vykonávaných

v SRU je dokončována postupně, tzn. že vykonání instrukce je pozdrženo do doby, než jsou vykonány všechny prioritní instrukce. Výsledky instrukcí dokončovaných postupně vykonávaných SRU nejsou dostupné nebo navratitelné pro podsekvenční instrukce, dokud není instrukce ukončena.

Ukončovací jednotka

Ukončovací jednotka sleduje instrukce od odeslání přes vykonání a potom je odstraňuje nebo ukončuje podle programu. Ukončení instrukce dává 603 možnost jakékoliv architekturní změny registrů pomocí instrukce. Ukončování podle pořadí zajišťuje správný architekturní stav když 603 musí obnovit špatně předpovězenou větev nebo výjimku.

Stav instrukce nebo jiné informace potřebné pro ukončení jsou uloženy v First- In- First- Out frontě pěti ukončovacích bufferů. Pro každou instrukci, která vstoupí do odesílací jednotky je vyhrazen jeden ukončovací buffer. Volný ukončovací buffer je nutnou podmínkou pro odeslání instrukce není-li k dispozici, odeslání instrukce se zpozdí. Za jeden takt jsou z fronty ukončeny maximálně dvě instrukce.

Podpůrný podsystém paměti

Jednotky řízení paměti (MMU)

MMU PPC603 podporují do 4 Petabytů virtuální paměti a 4 Gigabyty fyzické paměti pro instrukce a data. MMU také kontrolují přístupová práva pro tyto rozsahy blokově nebo stránkově dělené.

LSU počítá efektivní adresy pro nahrávání a ukládání dat, zajišťují řazení dat do a z cache a zajišťuje sekvencování pro n/u řetězec a víceslovné instrukce. Instrukční jednotka počítá efektivní adresy pro podání instrukce.

Potom, co je adresa generována, vyšší bity efektivní adresy jsou přeloženy MMU na fyzické adresové bity. Následně, nižší bity adresy jsou směřovány do čipových cache, kde tvoří index do dvoucestného asociativního pole tagů. Po přeložení adresy MMU posílá vyšší bity adresy do cache a ukončuje obhlídku cache.

Jednotky Cache

603 zajišťuje nezávislé 8 kB, dvoucestné množinově asociativní instrukční a datové cache. Velikost řádky cache je 32 b. Cache jsou vyrobeny tak, aby se držely chování write-back, ale 603 dovoluje kontrolu cacheovatelnosti, chování při zápisu a paměťové koherence na stránkové nebo blokové úrovni. Cache používají metodu nahrazování "Nejméně často použitý"(LRU).

Rozhraní Sběrnice Procesoru

Protože 603 má na čipu write-back cache, nejčastější typ přenosu pro většinu aplikací jsou paměťové operace čtení typu burst, následované výbuchovým zápisem, jednotaktovým čtením nebo zápisem do paměti a operacemi rozhraní s přímým ukládáním. Navíc mohou nastat operace jen adresové, varianty jednotaktových nebo výbuchových operací a aktivity obnovení adresy.

Přístupy k paměti mohou nastat v jednotaktovém (1-8 b) a čtyřtaktovém výbuchu (32B) datového přenosu, když je sběrnice konfigurována jako 64 bitová a v jednotaktovém (1-4B), dvoutaktovém (8 b) a osmitaktovém (32b) datovém přenosu, když je sběrnice konfigurována jako 32 bitová.

Víceúčelové registry (GPR)

Architektura PowerPC definuje 32 uživatelských univerzálních registrů (GPR). Tyto registry jsou buď 32-bitové v 32-bitových PowerPC mikroprocesorech a 64-bitové v 64-bitových GPR a slouží jako zdroj dat nebo cíl pro všechny celočíselné instrukce.

Registry pohyblivé řádové čárky (FPR)

Architektura PowerPC také definuje 32 uživatelských 64-bitových registrů FPR. FPR slouží jako zdroj dat nebo cíl pro operace s pohyblivou řádovou čárkou. Registry mohou obsahovat datové objekty jak s jednoduchou, tak i s dvojitou přesností.

Podmínkový Registr (CR)

CR je 32 bitový uživatelský registr skládající se z osmi 4-bitových polí, které odrážejí výsledky příslušných operací, jako pohyb, celočíselné a floatové porovnání, aritmetické a logické instrukce a zajišťuje mechanismus pro testování a větvení.

Registr pro status a kontrolu pohyblivé řádové čárky (FPSCR)

Tento registr (FPSCR) je uživatelský registr obsahující všechny signální bity pro výjimky, součtové bity výjimek, bity povolení výjimek a přímé kontrolní bity potřebné pro dodržení standardu IEEE 754.

Stavový registr stroje (MSR)

MSR je supervisorský registr definující stav procesoru. Obsah tohoto registru je uložen, je-li zachycena výjimka, a obnoven, když je dokončena. 603 implementuje MSR jako 32-bitový registr, 64-bitové PPC procesory implementují MSR jako 64-bitový.

Segmentové Registry (SR)

Pro správu paměti 32-bitové PPC procesory implementují 16 32-bitových segmentových registrů. Aby urychlila přístup, 603 implementuje SR jako dvě pole (hlavní pole (pro přístup do paměti dat) a stínové pole (pro přístup do paměti instrukcí).

Registry zvláštního určení (SPR)

PowerPC používá mnoho registrů zvláštního určení pro zajišťování kontroly, indikování stavu apod. Program k nim může přistupovat v závislosti na svých přístupových právech. 603 má SPR 32 bitové.

Uživatelské SPR

Uživatel má softwarový přístup k těmto registrům :

- Linkovací registr - používá se k zajištění cílové adresy větvení a k udržení návratové adresy po instrukcích větvení nebo linkovacích instrukcích. Zde jsou 32-bitové.
- Výpočetní registr - CTR je dekrementován a testován automaticky jako výsledek výpočetních instrukcí větvení.
- Celočíselný výjimekový registr-32-bitový XER obsahuje součtový bit přetečení, bit přetečení a pole specifikující počet bytů přesunovaných instrukcemi lswx a stswx.

Supervisorské SPR

Tyto SPR může použít jen supervisor :

- 32bitový DSISR definuje důvod datového přístupu a řadí výjimky.
- Registr adresy dat (DAR) je 32-bitový registr udržující adresu přístupu po seřazení nebo DSI výjimce.
- Dekrementovací registr (DEC) je 32-bitový dekrementující čítač, který zajišťuje mechanismus pro provádění dekrementovací výjimky po programovatelném přerušení.

6.5.4 Procesor PowerPC 604

Hlavní znaky procesoru PowerPC 604 jsou následující :

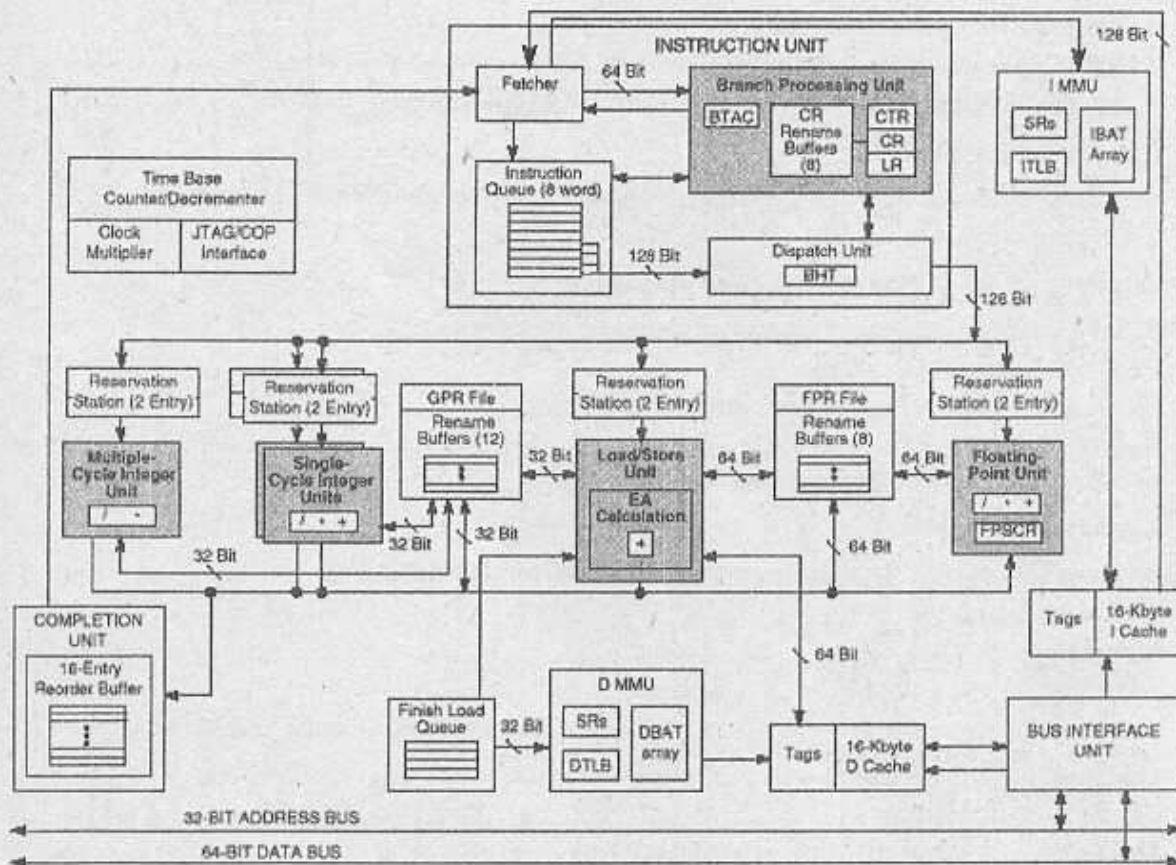
Vysocevýkonový superskalární procesor.

- nejméně čtyři instrukce mohou být vybaveny za jeden takt
- nejméně šest instrukcí se může začít vykonávat v jednom taktu včetně tří celočíselných
- většina instrukcí je vykonávána za jeden takt

Šest nezávislých vykonávacích jednotek a dva soubory registrů

- **BPU charakteristické dynamickou předpovědí větvení**
 - promyšlené větvení
 - 64-vstupová plně asociativní cache cílové adresy větvení (BTAC)
 - 512-vstupová tabulka historie větvení (BHT), se dvěma bity na vstup pro každou úroveň předpovědi
- **dva jednocyklové TU (SCTU) a jeden vícecyklový TU (MCTU)**
 - instrukce vykonávané v SCTU se vykonávají jeden cyklus, většina instrukcí vykonávaných MCTU se vykonává více cyklů
 - každá SCTU má dvouvstupovou rezervační stanici pro minimalizaci prodlev
 - MCTU má dvouvstupovou rezervační stanici a poskytuje předčasné ukončení (tři cykly) pro 16x32-bitové operandy a operandy přetečení
 - 32 GPR pro celočíselné operandy
 - 12 přejmenovacích bufferů pro GPR
- **třístupňová jednotka pohyblivé řádové čárky (FPU)**
 - plně IEEE 754-1985 FPU pro operace s jak jednoduchou, tak i dvojitou přesností
 - podporuje ne-IEEE mód pro časově kritické operace
 - plně propojený jednopřechodový design s dvojitou přesností
 - hardwarová podpora pro nenormalizovaná čísla
 - dvouvstupová rezervační stanice pro minimalizaci zpoždění
 - 32 64-bitových FPR pro operandy s jednoduchou nebo dvojitou přesností
- **vkładací/układací jednotka (LSU)**
 - jednocyklový propojený přístup do cache
 - střádač pro EA výpočty
 - vykonává seřazení a precizní konverzi pro data s pohyblivou řádovou čárkou
 - vykonává seřazení a označuje velikost pro celočíselná data
 - čtyřvstupová fronta s vkládáním na konec FLQ zajišťuje bufferování chybějící náplně
 - šestivstupová ukládací fronta
- **přejmenovatelné buffery**
 - 12 GPR přejmenovatelných bufferů
 - 8 FPR přejmenovatelných bufferů
 - 8 podmínkových registrů pro přejmenovatelné buffery
- **ukončovací jednotka**
 - čtyři instrukce vybavené za takt
 - ukončovací jednotka vybavuje instrukce s 16-vstupového přeorganizovatelného bufferu, když všechny instrukce před ní již byly dokončeny a instrukce dokončila provádění
 - zajišťuje sekvenční programovací model
 - monitoruje všechny odbavené instrukce a vybavuje je dle pořadí
 - sleduje neanalyzované větve a odstraňuje nenormálně provedené odbavené a vykonané instrukce, jestliže je větev vadná
- **oddělené datové a instrukční cache přímo na čipu**
 - 16 kB čtyřcestné množinově asociativní instrukční a datové cache
 - LRU odstraňovací algoritmus

- 32-bitů velikost bloků cache
- fyzicky indexované tagy
- cachové operace write-through nebo write-back programovatelné na stránkové nebo blokové bázi
- instrukční cache může poskytovat čtyři instrukce za takt, datová cache dvě slova za takt
- cache mohou být softwarově vyřazeny, zamčeny
- zkouška na paritu u obou cache
- koherence datové cache MEST, zajišťovaná hardwarově
- zajištěná podpora další datové cache
- koherence instrukční cache, zajišťovaná softwarově
- **oddělené jednotky řízení paměti (MMU) pro instrukce a data**
 - zařízení pro překlady adresy pro 4 kB velikost stránky, velikost bloku proměnných, 256 MB velikost segmentu
 - obě TLB jsou 128-vstupové a dvoucestné asociativní
 - obě TLB jsou hardwarově naplnitelné
 - oddělené instrukční a datové překladové buffery (TLB)
 - oddělené IBAT a DBAT (čtyři každý) také definované jako SPR
 - LRU odstraňovací algoritmus
 - hardwarové hledání v tabulce přes hashované tabulky stránek
 - 52-bitové virtuální adresy, 32-bitové fyzické adresy
- **vlastnosti rozhraní sběrnice obsahují následující:**
 - volitelný poměr procesor versus sběrnice (1:1, 1.5:1, 2:1, 3:1)
 - 64-bitová externí datová sběrnice s burst-transfery
 - podpora pro adresové propojení a limitované mimoprovozní sběrnice transakce
- **víceprocesová podpora obsahuje následující:**
 - hardwarově vnucený, čtyřstavový cache koherentní protokol (MEST) pro datovou cache
 - vkládání/ukládání s rezervovanými páry pro základní paměťové reference, semaforey a jiné multi-procesové operace
- **Power Management**
 - XAP mód podporuje plné vypnutí a podporuje snooping s včasnou indikací
 - operační voltáž 3.3 +/-0.3V



Obr. č. 6.14

6.5.5 Procesor PowerPC 750

Vzhledem k podobnosti s PowerPC 604 se zmíníme jen o odlišných rysech. Vysoce výkonný superskalární procesor - nejméně dvě instrukce mohou být odeslány za jeden takt.

Šest nezávislých vykonávacích jednotek a dva soubory registrů

- BPU charakteristické jak statickou, tak i dynamickou předpovědí větvení
 - instrukce větvení, které nemodifikují registr výpočtu (CTR) nebo registr linku (LR) jsou odstraněny z instrukčního toku
 - 64-bitová (16-ti množinová čtyřcestná množinově asociativní) cílová instrukční cache, cache instrukcí větvení, která byla označena ve větvících/skokových sekvencích kódu. Je-li cílová instrukce v BTIC je vložena do fronty instrukcí dříve, než by bylo možné s instrukční cache, zasahuje-li vkládací přístup BTIC, postihuje první dvě instrukce z cílového toku.
- dvě celočíselné jednotky IU, které sdílejí 32 GPR pro celočíselné operandy
 - IU1 může vykonat jakoukoli celočíselnou instrukci
 - IU2 může vykonat libovolnou celočíselnou instrukci vyjma instrukcí násobení a dělení. Většina instrukcí vykonávaných IU2 je jednocyklových. IU2 má jednovstupovou rezervační stanici.
- třívstupňová FPU
 - jednovstupová rezervační stanice
- dvouvstupová LSU
 - třívstupňová ukládací fronta

Přejmenovatelné buffery

- 6 GPR přejmenovatelných bufferů
- 6 FPR přejmenovatelných bufferů
- bufferování podmínkového registru podporuje dva CR zápisy za takt

Ukončovací jednotka

- ukončovací jednotka vybavuje instrukce ze 6-vstupového přeorganizovatelného bufferu
- dvě instrukce vybaveny za takt

Odděleně přímo na čipu datová a instrukční cache

- 32 kB osmicestná množinově asociativní instrukční a datová cache
- PLRU odstraňovací algoritmus
- cache mohou být zamčeny nebo vyřazeny softwarově
- kritické dvojité slovo je dáno k dispozici žádající jednotce, když se dostane do bufferu. Cache je neblokující, takže může být přístupná během operace.

Level2 rozhraní cache

- přímo na čipu ovladač dvoucestné množinově asociativní L2 cache a tagy
- externí datové SRAM
- podpora pro 256 kB, 512 kB a 1 MB L2 cache
- 64 b (256/512 kB) a 128 b (1 MB) velikost dělené linie
- podporuje flow-through, propojené, se zpožděným zápisem synchronní burst SRAM

Oddělené jednotky řízení paměti pro instrukce a data

- paměti programovatelné jako write-back/write-through, cachovatelné/necachovatelné a koherentně posílené/neposílené na stránkové nebo blokové bázi

Oddělené jednotky rozhraní sběrnice pro systémovou paměť a L2 cache

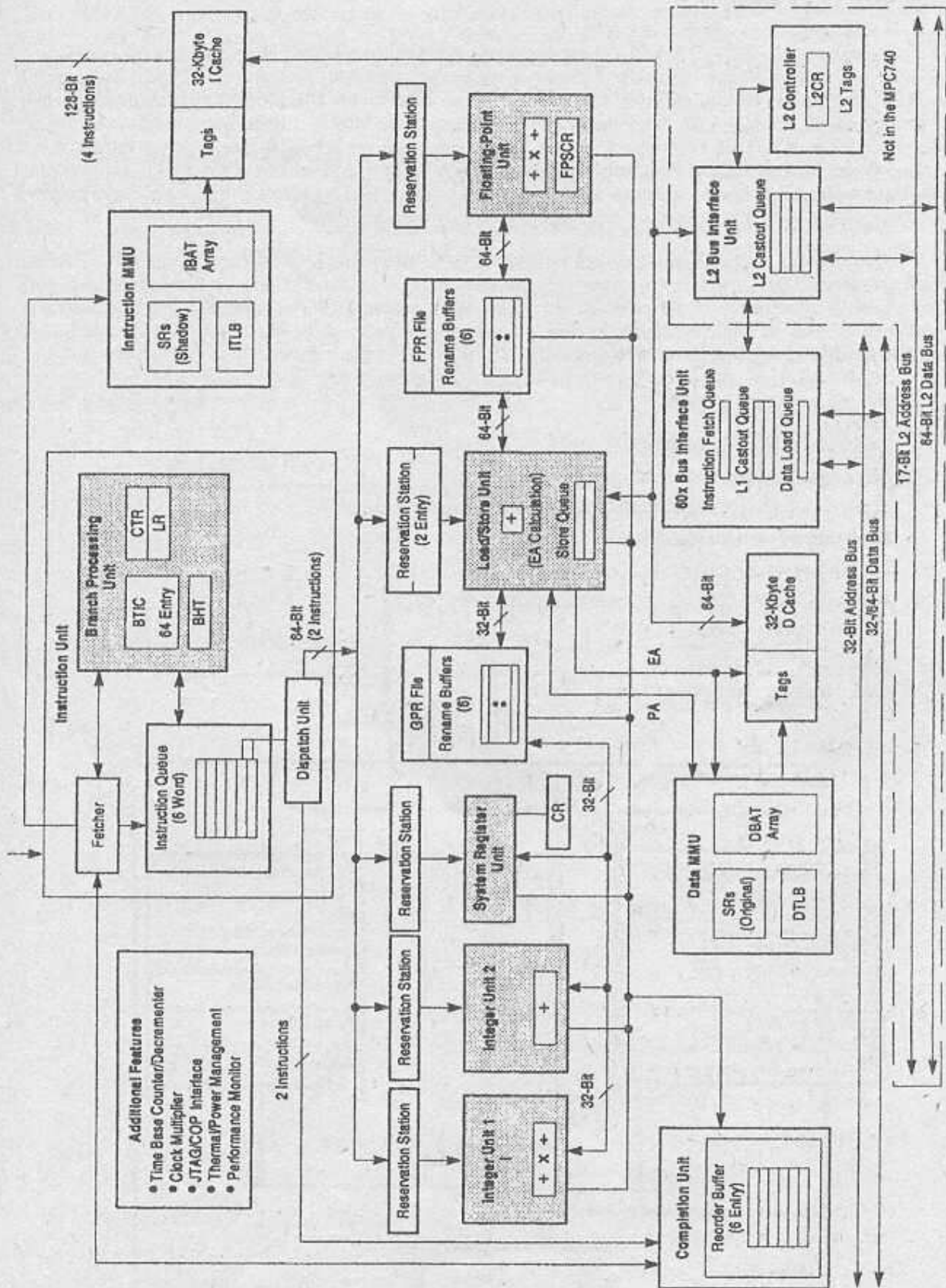
- rozhraní sběrnice je charakteristické následujícím :
 - volitelné poměry frekvence sběrnice k jádru (2x, 2.5x, 3x, 3.5x, 4x, 4.5x až 8x)
 - 64-bitová externí datová sběrnice s burst transfery
 - podpora pro adresové propojení a limitované mimoprovozní sběrnice transakce
 - jednovstupová vkládací fronta
 - jednovstupová instrukční vybavovací fronta
 - dvouvstupová fronta volání cache L1
- rozhraní cache L2 se vyznačuje :
 - jádro k L2 frekvenční dělitele 1, 1.5, 2, 2.5 a 3
 - čtyřvstupová cache L2 vyprazdňující frontu L2 cache BIU
 - 17-bitová adresová sběrnice
 - 64-bitová datová sběrnice

Víceprocesová podpora

- hardwarově posílený třístavový cache koherentní protokol MEI pro datovou cache
- vkládání/ukládání s rezervovanými instrukčními páry pro základní paměťové reference, semaforey a další multiprocesové reference

Power and Termal Managment

- kromě ovládání spotřeby energie procesoru MPC750 umožňuje ovládat i tepelné vyzařování procesoru



MPC750 Microprocessor Block Diagram

Obr. č. 6.15

6.6 Procesory MIPS

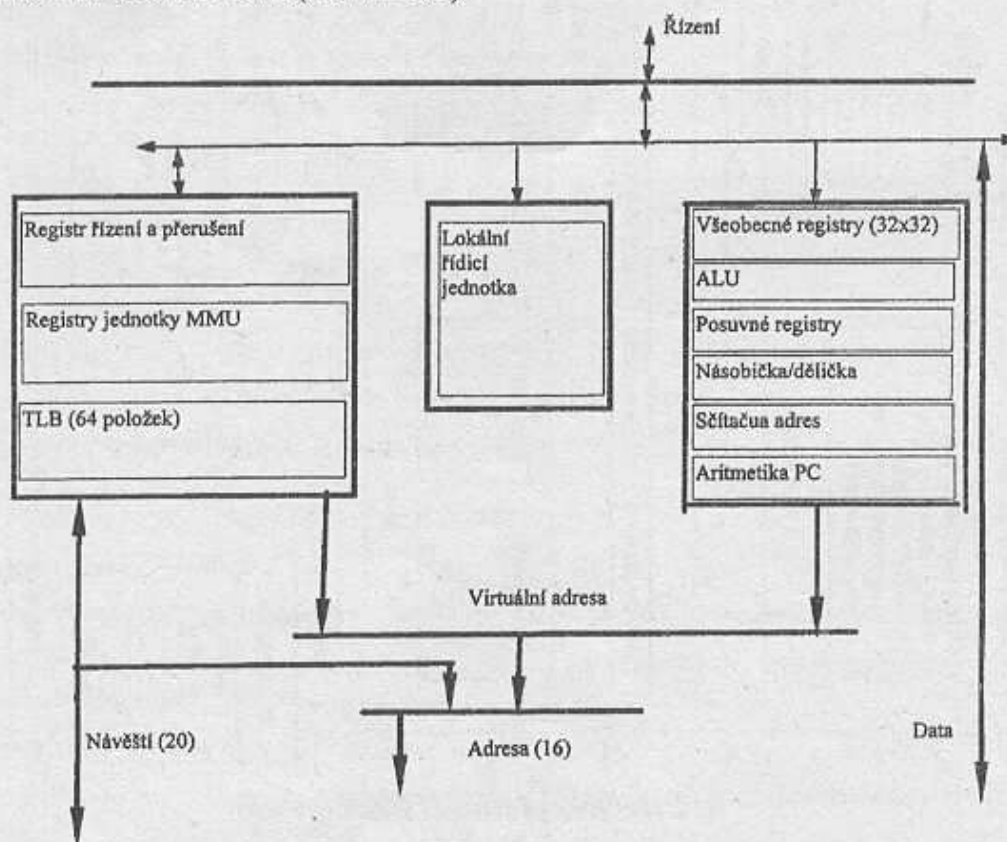
Procesory MIPS [3, 19, 40] (Microprocessor Without Interlocked Pipeline Stages) pocházejí z výzkumné práce Stanfordské university. Zde byly zkoumány možnosti tvorby kompilátorů, které by při kompilaci optimálně využívaly možnosti zřetěženého procesoru. Jako vedlejší produkt byl návrh velmi jednoduchého procesoru, který byl ale extrémně rychlý a výkonný. Zásadním rozdílem oproti ostatním výrobcům procesorů RISC je v případě procesoru MIPS to, že nejprve existoval kompilátor a až potom začala vlastní obvodová implementace. Ještě před existencí procesoru byly ověřeny vlastnosti kompilátoru a tím i instrukčního souboru budoucího mikroprocesoru. Potom došlo k technické implementaci konkrétní instrukce z hlediska maximalizace výkonnosti. Takovýmto způsobem byly navrženy a realizovány základní funkční celky procesoru MIPS.

Místo tvorby rozsáhlé sady registrů (registrová okna /Register Windowing/ - je taková organizace registrů procesoru, která realizuje kruhovou vyrovnávací paměť /Circular Buffer/) a umožňuje redukci doby úklidu a obnovy registrů při volání procedur, je v případě procesoru MIPS, celá problematika přenesena do kompilátoru. V něm je implementována technika meziprocedurálního přiřazení registrů, která zabezpečuje minimalizaci přenosů registrů z/do hlavní paměti.

6.6.1 Procesor MIPS R3000

Procesor R3000 [3] obsahuje:

- třicet dva 32bitových registrů všeobecného určení
- šestnáct 64bitových registrů pro procesor pohyblivé řádové čárky
- aritmeticko-logickou jednotkou
- 32bitový posuvný registr
- autonomní celočíselnou 32bitovou násobičku a děličku
- jednotku správy paměti (MMU)
- jednotku správy přerušeni
- modul transformace adres (tabulka TLB)



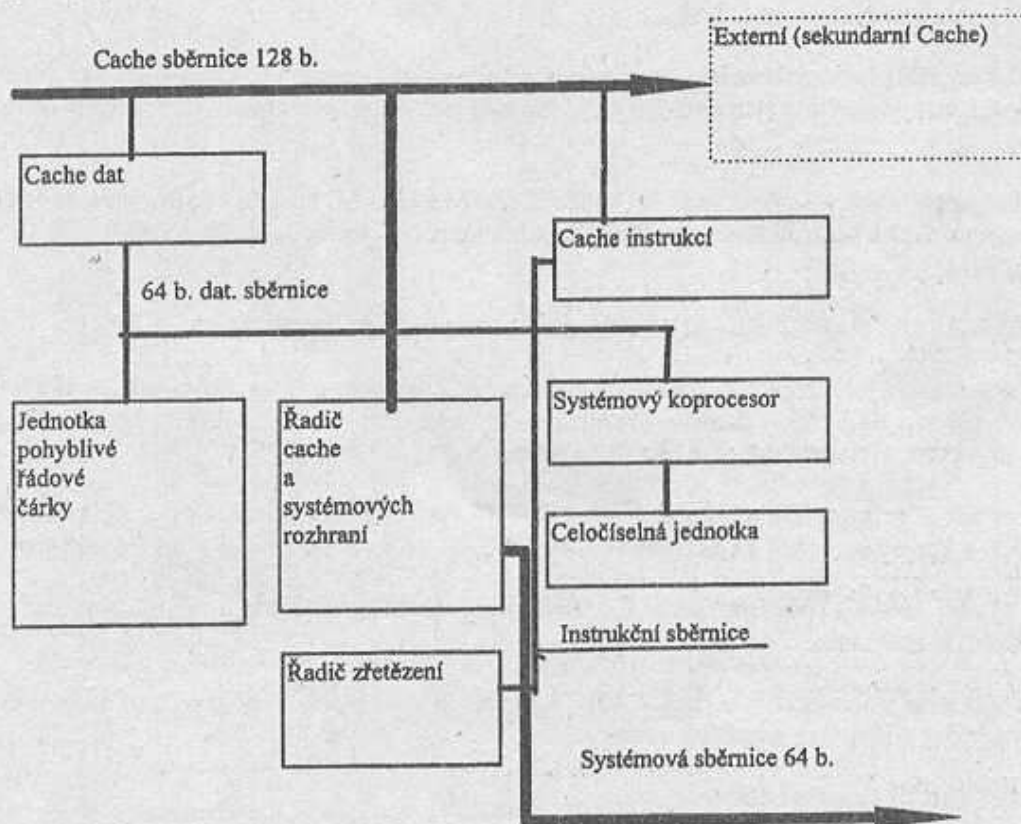
Obr. č. 6.16 Architektura procesoru MIPS R3000

Modul transformace adres obsahuje 64 položek a dokáže díky implementaci asociativního vyhledávání transformovat virtuální adresu do fyzické za 20 ns. Každá položka TLB má identifikační pole procesu (Process Identity Field - PIF), které napomáhá k rychlému přepínání kontextu mezi různými procesy bez nutnosti vždy uvolňovat TLB. Osm položek TLB je možno "uzamknout" a tím blokovat jejich případnou modifikaci a přístup k nim.

Architektura MIPS podporuje širokou škálu typů přerušení. Externí asynchronní přerušení zahrnují 6 maskovatelných přerušení a také přerušení způsobené chybou sběrnice. Modul správy přerušení je v procesoru MIPS velmi dobře propracován. Přerušení je obsluhováno takovým způsobem, že nedojde k narušení zřetězené realizace rozpracovaných instrukcí a zároveň nedojde k rozpracování žádné další instrukce. MIPS procesor je v tomto směru považován za nejpropracovanější RISC procesor v souvislosti s detekcí a obsluhou přerušení.

6.6.2 Procesor MIPS R 4000

Filosofie MIPS procesoru R 4000 [3] realizuje rostoucí požadavky na výpočetní systém současnosti a vyhovuje široké škále aplikací, jako jsou pracovní stanice, DTP (Desktop Publishing), server lokální počítačové sítě, multiprocesory atd.



Obr. č. 6.17 Architektura procesoru MIPS R 4000

Architektura procesoru MIPS R 4000 je uvedena na obr. č. 6.9. Je to plně 64bitový jednočipový procesor, který může pracovat s taktem až 50 MHz. Součástí procesoru je:

- 64bitový procesor pohyblivé řádové čárky, který není součástí R 4000
- jednotka správy paměti MMU
- 8 kB paměti cache pro instrukce a 8 kB paměti cache pro data
- řadič primární a sekundární vyrovnávací paměti
- technická podpora pro implementaci v multiprocesorových systémech

Typický systém postavení na bázi procesoru MIPS vyžaduje minimální nároky na dodatečnou logiku. Vestavěný správce paměti cache dokáže využívat běžné typy paměti SRAM. Je možno volit mezi 4 až 256 kB paměti cache jak pro data, tak i pro instrukce (celkově až 512 kB). Každý blok paměti cache může mít 1, 4, 8, 16 nebo 32 slov, přičemž velikost slov lze separátně volit pro cache oblasti instrukcí a dat. Uvedená organizace poskytuje např. procesoru MIPS R3000 při 25 MHz přenosovou rychlost mezi procesorem a hlavní pamětí až 200 MB/s.

6.6.3 Procesory MIPS R4400, R4600 a R8000

Indigo CPU architektura představuje dva typy CPU: R4400 nebo R4600 [22]. The Power Indigo představuje typ R8000 [22].

Indigo systém zahrnuje:

- zřetěžené GIO64 BUS sloty
- dva kanály SCSI-2, Base-T Ethernet a standardní porty myši a klávesnice
- čtyři EISA BUS konektory
- zdokonalenou architekturu systému

Indigo je příkladem zdokonalení cesty pro nové a rychlejší procesory, CPU a sekundární cache je realizována jednoduchou změnou standardního CPU modulu tak, že spolu komunikují nezávisle na ostatních členech boardu.

Hlavní paměť systémového boardu má kapacitu 384 MB DRAM. Protože systémový board nahrazuje SIMM spojením DRAM s kontrolním čipem DMUX, je hlavní paměť implementována standardně 36 bitovou šířkou DRAM SIMM.

Na následujícím obrázku vidíme INDIGO systém board.

Jádro procesoru obsahuje CPU a sekundární cache. Tato oblast je navržena jak s výhodou velké rychlosti CPU, tak pro 64-bitovou datovou sběrnici. CPU, sekundární cache a EEPROM je v této oblasti realizována samostatně z důvodu jednoduché náhrady nebo výměny.

Hlavní paměť zahrnuje DRAM s podpůrnými obvody. Oblast je opět navržena pro podporu 64-bitového přístupu DRAM a implementována s kontrolním čipem DMUX. To dovoluje použít standardní paměť.

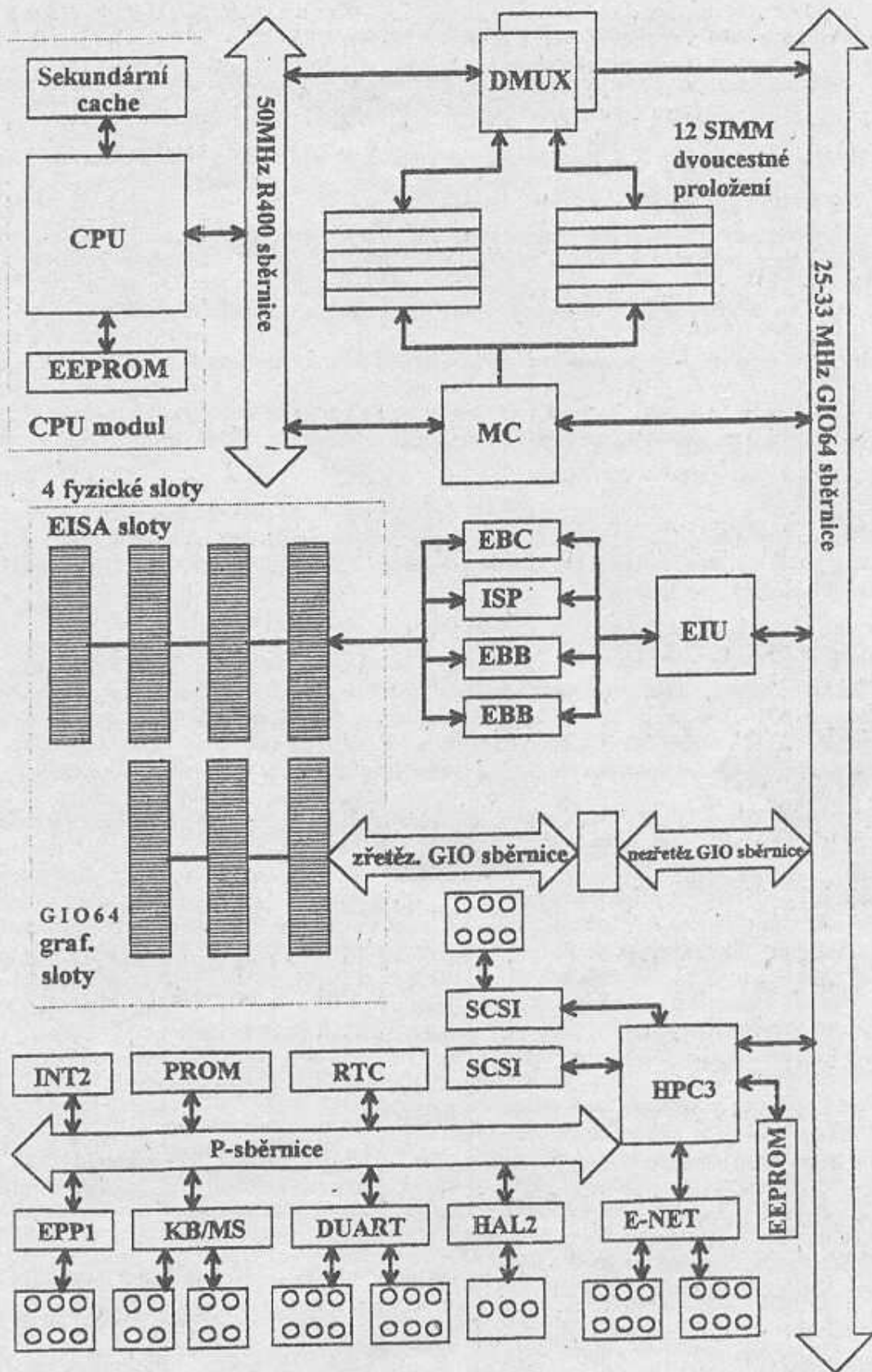
I/O systém zahrnuje periferní porty navržené pro ovládání příchozích a odesílaných dat tak, aby minimálně zatěžoval procesor.

Audio systém s architekturou HAL2 ASIC sestavenou pro využití vykonávaných audio I/O funkcí nahrazujících potřeby digitálních signálů procesoru.

Tři sběrnice propojené části:

- CPU sběrnice, která propojuje CPU s kontrolním hardwarem. Tato sběrnice je plně 64-bitová.
- GIO64 sběrnice, propojuje systém s jádrem procesoru, hlavní pamětí, I/O systémem, rozšiřujícími sloty a grafickým boardem.
- EISA sběrnice je 32-bitová a propojuje GIO64 s EISA sloty.

CPU, GIO64 a EISA sběrnice mají každá samostatně řízený hodinový obvod řízený různou rychlostí podle své způsobilosti. CPU a některé čipy mohou být tedy vylepšovány a měněny nezávisle na systému.



Obr. č. 6.18 Indigo system board

ASIC - podpora komunikace mezi sběrnici a systémem:

MC ASIC provádí mnoho funkcí jako správce GIO64 sběrnice, který představuje samotný interface mezi CPU a GIO64. Je to také tzv. memory controller umožňující DMA (Direct memory access).

DMUX ASIC je sada datových čipů, které jsou řízeny čipem MC. Tyto čipy izolují CPU sběrnici od GIO64 sběrnice.

HPC3 ASIC představuje interface mezi audio systémem, I/O a ostatními periferními ovladači, který jim umožňuje komunikaci s GIO64.

EIU ASIC realizuje interface mezi GIO64 sběrnici a Intel EISA čipovou sadou, řídí EISA sloty a EISA ovladače.

INT2 ASIC je vysoce výkonný obousměrný čip paralelního portu, který posílá data DMA s minimálním ovlivněním CPU.

REX nebo HQ2 ASIC umístěný na grafickém boardu spojuje tento board s GIO64 sběrnici.

1. Jádro procesoru

Procesorový modul obsahuje CPU a jeho sekundární cache, používá tyto tři základní čipy: MC ASIC, který poskytuje CPU přístup do paměti a na GIO64 sběrnici, DMUX ASIC izolující CPU sběrnici od GIO64 a INT2 ASIC pro předání přerušení.

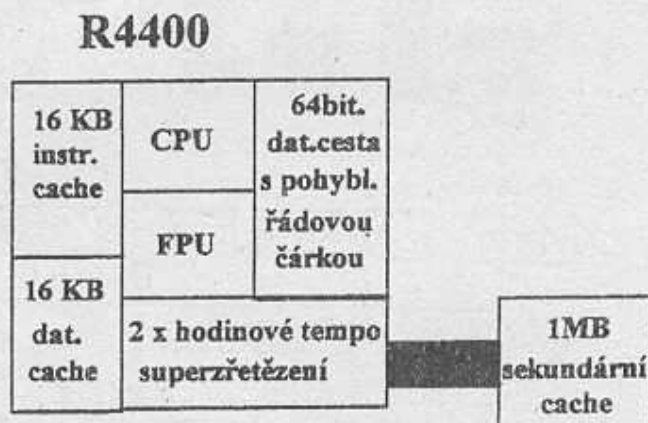
MIPS R4400SC CPU modul:

Tento modul vidíme na následujícím obrázku. Modul obsahuje hlavní procesor, procesor pohyblivé řádové čárky, 16 KB instrukční cache a 16 KB datovou cache. R4400SC má ještě vyhrazený interface pro 1 MB sekundární cache. Vnější rychlost hodinového čipu je 100 MHz, ale vnitřní je 200 MHz. 64-bitové data a instrukce přistupují do cache s takovou rychlostí, že minimalizují ztráty cache a chod CPU naprázdno.

Systémový výkon je 200 MHz u R4400SC a překračuje 180 MIPS (mil. instrukcí za sekundu), 34 MFLOPS, 124 SPECfp92 a 116 SPECint92.

R4400SC používá superzřetězení pro dosažení vysoké vnitřní rychlosti. Při normálním zřetězení CPU rozděljuje všechny instrukce (obvykle čtení, zápis do paměti a zpět, provádění) do jednocyklových kroků a potom provádí instrukce v těchto jednocyklových intervalech.

Při největším CPU hodinovém tempu některé instrukční kroky jako čtení a zápis cache nemohou být prováděny v jednom obyčejném zřetězeném cyklu. Superzřetězení vykoná všechny tyto kriticky pomalé operace v jednom cyklu. A to tak, že jednotlivé kroky rozdělí na podkroky, které provádí normálním zřetězením. Superzřetězení R4400SC je optimalizováno tak, že požaduje malou kontrolu logické a instrukční struktury.



Obr. č. 6.19 Funkční oblast MIPS R4400SC CPU

R4400SC CPU podporuje MIPS 1 instrukční set, MIPS 2 instrukční set a MIPS 3 instrukční set. Data v MIPS 3 jsou 64-bitová a poskytují schopnost systému provádět operace load a store s double slovy s pohyblivou řádovou čárkou v jednom cyklu. MIPS 3 instrukční set také provádí základní kontrolu a synchronizaci cache.

MIPS R4600SC CPU modul:

Tento modul vidíme na obrázku č. 6.20. Modul obsahuje hlavní procesor, procesor pohyblivé řádové čárky, 16 KB instrukční cache a 16 KB datovou cache. Vnější rychlost hodin je 66,67 MHz, ale vnitřní je 133 MHz. 64-bitová data a instrukce přistupují do cache s takovou rychlostí, že minimalizují ztráty cache a chod CPU naprázdno.

Systémový výkon je 133 MHz u R4600SC a překračuje 169 MIPS (mil. instrukcí za sekundu), 20 MFLOPS, 72 SPECfp92 a 109 SPECint92.

R4600 používá 5 stupňové zřetězení podobné R3000. Toto zřetězení je jednodušší než superzřetězení používané u R4400 a je úspornější ve srovnání s ostatními - vzhledem k době ukládání.

R4600

16 KB instr. cache	CPU	64bit. dat. cesta s pohybl. řádovou čárkou
16 KB dat. cache	FPU	
	64 bitový interface	

Obr. č. 6.20 Funkční oblast MIPS R4600SC CPU

R4600SC CPU podporuje MIPS 1 instrukční set, MIPS 2 instrukční set a MIPS 3 instrukční set. Data v MIPS 3 jsou 64-bitová a poskytují schopnost systému provádět operace load a store s double slovy s pohyblivou řádovou čárkou v jednom cyklu. MIPS 3 instrukční set také provádí základní kontrolu a synchronizaci cache.

MIPS TFP R8000 modul:

Procesor MIPS TFP prezentuje zcela novou generaci procesorů architektury RISC. Každý, procesor MIPS TFP poskytuje špičkové vykonávání operací s pohyblivou řádovou čárkou rychlostí 300 MFLOPS - tedy přibližně stejně jako CRAY Y-MP.

Čipová sada procesoru TFP je první 64-bitová superskalární implementace ISA (instrukční architektura) MIPS IV. Jeho superskalární implementace dosahuje extrémních superpočítačových výkonů. Procesor TFP v sobě spojuje značně rychlý přístup, vysoce propustný subsystém cache s velmi výkonným prováděním operací s pohyblivou řádovou čárkou. Jeho rozsah uspokojuje i aplikace s velkým objemem dat. Systém TFP je jako jediný schopen udržovat úroveň výkonu na velice rozsáhlých a nákladných superpočítačových systémech. V minulosti měly procesory typicky omezenou čipovou funkčnost. Malé subsystémy cache a další limitující faktory snižovaly výkon pro rozsáhlé vědecké aplikace. Pro odstranění těchto omezení a pro rozšíření použitelnosti vyvinul Silicon Graphics novou technologii výroby čipů a zdokonalenou architekturu instrukčního souboru. Vysoce integrovaná čipová sada procesoru TFP úspěšně splňuje požadavky numericky náročných aplikací v technické praxi.

Čipová sada podává špičkový výkon 300 MFLOPS s dvojnásobnou přesností a 300 MIPS při taktovací frekvenci 75 MHz. Instrukční soubor TFP je vlastně instrukčním souborem MIPS IV, který je vylepšenou verzí architektury MIPS III a je kompatibilní s předcházejícími verzemi.

Technický přehled

Procesor MIPS TFP je navržen pro extrémně vysoký výkon v oblasti provádění operací s pohyblivou řádovou čárkou. Jeho klíčové rysy jsou:

- Vícesložková čipová sada skládající se z IU, FPU, Tag RAM a 4 MB data-stream cache.
- Řadič cache odpovídající za rozhraní s hlavní pamětí.
- Čtyřcestná superskalární architektura, až 6 operací během jednoho hodinového taktu.
- Plně 64-bitový mikroprocesor s 64-bitovými operacemi s pevnou a pohyblivou řádovou čárkou, s registry a s virtuálním adresováním.
- 3.3 voltová technologie.
- 16 kB instrukční cache v IU, 16 kB double-ported datová cache, 1 kB údajů pro predikci větvení.

- MMU v IU obsahující 384 adres stránek, trojcestnou double-ported asociativní TLB s adresami stránek o proměnné velikosti.
- ANSI/IEEE 754 standardní koprocesor pro operace s pohyblivou řádovou čárkou a s přerušeními.
- 32 64 bitových registrů pro hlavní účely v IU a 32 -64 bitových registrů pro operace s pohyblivou řádovou čárkou v FPU.
- 128 bitová datová sběrnice, která je schopna mapovat až 1 TB fyzické paměti.
- Souvislé rozhraní systému cache s pružnou podporou multiprocessingu.
- Plná kompatibilita s předcházejícími 32-bitovými a 64-bitovými procesory MIPS.

Superskalární implementace

Čipová sada TFP je cenově výhodné řešení pro náročné vědecké výpočty. Superskalární implementace byla vybrána pro:

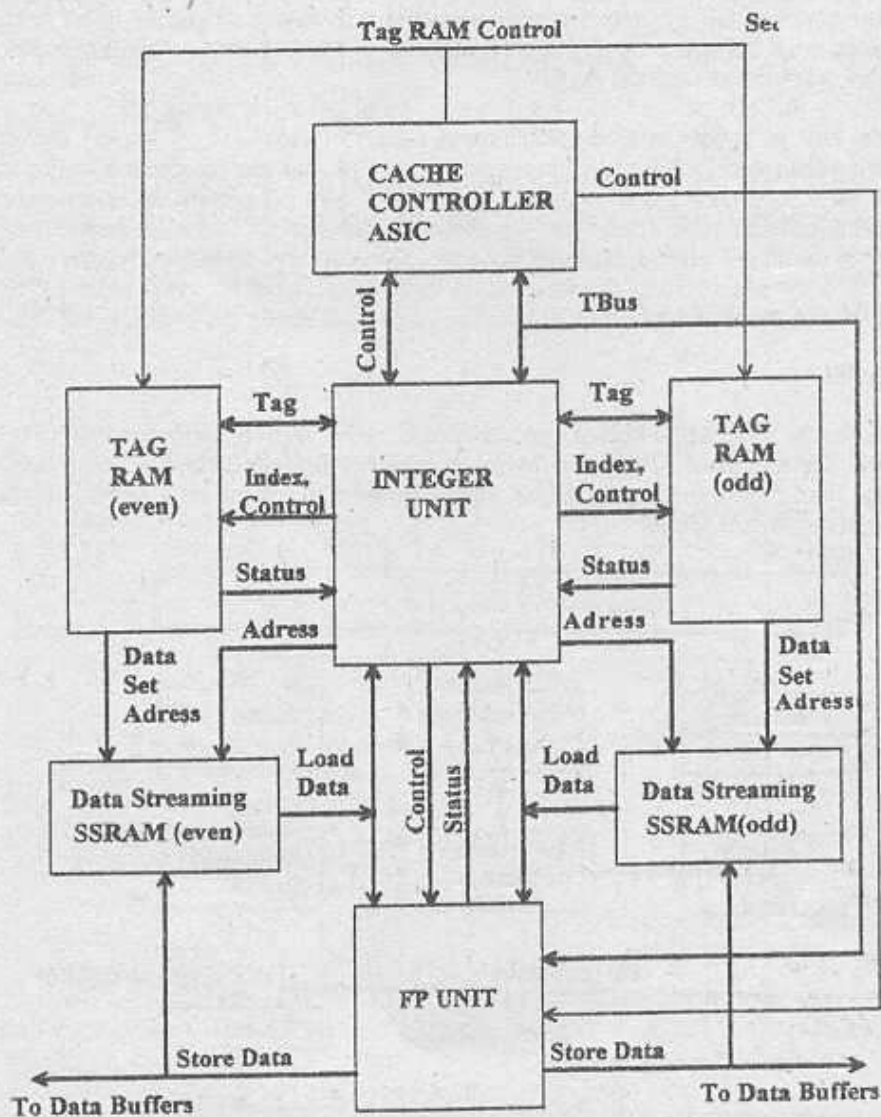
- Podporování výkonu vektorové architektury při vektorizovaném kódu.
- Akcelerování početně náročných kódů.
- Podporu kompatibility s produkty nižší úrovně.

Implementace propustnosti vyrovnávací paměti

Čipová sada procesoru TFP je přizpůsobena k provádění operací s pohyblivou řádovou čárkou. Nicméně obě schopnosti provádění operací s pevnou i s pohyblivou řádovou čárkou jsou v rovnováze s požadovanou paměťovou propustností. Dvě load/store jednotky v IU poskytují funkční jednotky pro Integer a FP s 64-bitovou (Doubleword) informací požadovanou podpůrnými operacemi. Dvě load/store jednotky obstarávají v jednom cyklu nejvýše dva odkazy na paměť. Jsou možné jakékoliv kombinace celočíselných i FP (floating point) instrukcí load a store, vyjma celočíselného store následujícím za druhým celočíselným store ve stejném cyklu. Mimo toho zde nejsou žádná jiná omezení na zřetězení jakýchkoliv jiných kombinací celočíselných a FP load a store.

TFP dokáže zpracovat až čtyři instrukce a dva paměťové přístupy během cyklu. V IU mohou být zpracovávány až dvě celočíselné instrukce (dvě větvení). V FPU (floating point unit) mohou být zpracovávány až dvě FP instrukce. Zdvojené FP jednotky v FPU paralelně ovládají násobení, sčítání, dělení a druhou mocninu. Takto je možno provést až čtyři FP operace během jednoho cyklu. Užitím separovaných celočíselných a FP čipů maximalizuje FP propustnost a připouští současné provádění operací s pevnou a pohyblivou řádovou čárkou.

Charakteristiky paměťového subsystému TFP t.j. počet portů, velikost a algoritmy cache, příznaková RAM, použití buferu doplňuje výkon výpočetních možností TFP a zabezpečuje, že jsou splněny paměťové nároky z FPU a IU.



Obr. č. 6.21 Čipset mikroprocesoru TFP

Instrukční a adresní rozšíření

Několik nových instrukcí zvyšuje výkon při provádění numericky náročných aplikací. Složené FP instrukce pro násobení a sčítání dosahují výkonů srovnatelných se zřetěženými vektorovými operacemi. Vícenásobně FP operace se provádějí v každém strojovém cyklu. Výsledkem těchto vlastností je vyšší výkon a větší přesnost.

Mnoho vědeckých aplikací má oddělené kompilované podprogramy obsahující matice s proměnnou dimenzí jako parametry. Standardní registr-plus-offset adresování vyžaduje dodatečné celočíselné sčítání pro každý přístup do těchto polí. Naproti tomu indexovaný adresní režim TFP (bázový registr plus indexový registr) eliminuje toto dodatečné sčítání pro FP load a store. Pro nejnáročnější numerické procesy musí být účinně vykonávány cykly obsahující příkaz IF. Návrh TFP obsahuje soubor čtyř operátorů podmíněného posuvu, které umožňují použít příkazu IF bez větvení. Těla bloků THEN a ELSE jsou nepodmíněně provedena a výsledky jsou pak uloženy v dočasných registrech. Podmíněné operátory posuvu pak výběrově přenášejí výsledky do příslušných registrů. Souhrnně jsou provedeny obě větve příkazu IF a jedna z nich je vynechána.

Architektura data-stream cache

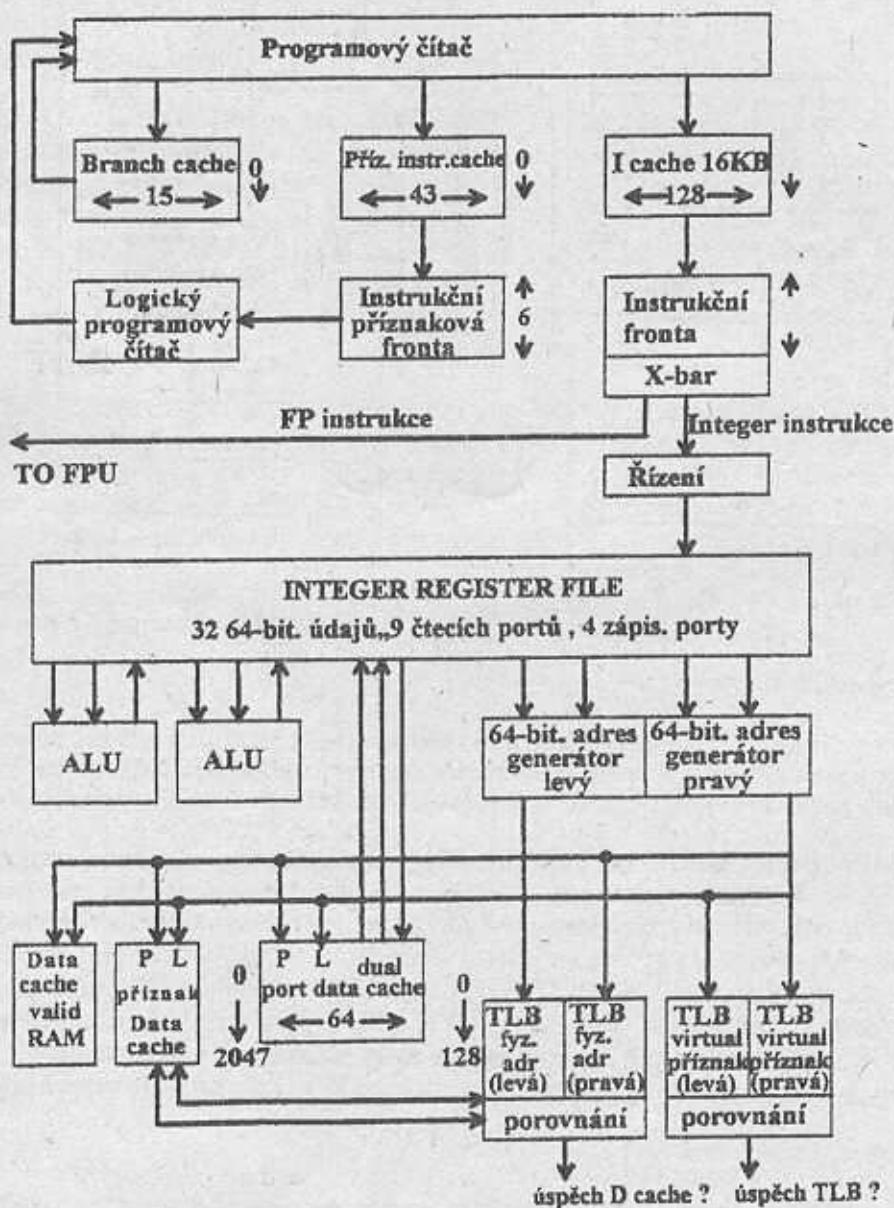
Čipová sada TFP spojuje schéma souvislé cache pro adresování dvou různorodých výpočetních požadavků. Většina programů obsahuje směs celočíselných FP a adresových výpočtů. Celočíselné a adresové

výpočty jsou rychlou odezvou. FP výpočty vyžadují rozsáhlé požadavky na paměť nebo vyžadují rozsáhlý subsystém cache. Cache musí být nejen měněna rychle, ale musí být také značně rozsáhlá pro uchování velkého objemu dat a pro vysokou rychlost toku dat do FPU.

Čipová sada TFP je vybavena ojedinělou úrovní rozdělení cache. První úroveň datové cache je na celočíselném čipu. Její velikost je 16 KB a dovoluje velmi rychlý přístup pro celočíselné load a store. Rozsáhlá externí cache o velikosti 4 MB nazývaná data-stream cache slouží jako sekundární cache pro celočíselná data a instrukce a také jako primární cache pro FP data. Toto uspořádání dovoluje při FP load a store interní cache a umožňuje tak komunikaci přímo s rozsáhlou externí data-stream cache. Tato cache je zřetězena, aby byl dovolen souvislý přístup FPU. Data mohou být přenášena rychlostí dvou 64-bitových slov během cyklu nebo 1,2 GB/s. Rozsáhlé zřetězení data-stream cache zajišťuje kapacitu a dostatečnou šířku potřebnou pro ovládání FP datových objektů.

Organizace IU

Čip IU, 591pinové zařízení, je hlavní výpočetní část čipové sady mikroprocesoru TFP. Její jednotky při vícenásobném vykonávání operací jsou podporovány vyhrazenými řadiči, které dovoluji nezávislé operace této funkční jednotky. Souhrnné sběrníkové schéma snižuje potřebu násobení dat a adres a dovoluje současně FP load a store z a do data-stream cache.



Obr. č. 6.22 Integer Unit procesoru TFP

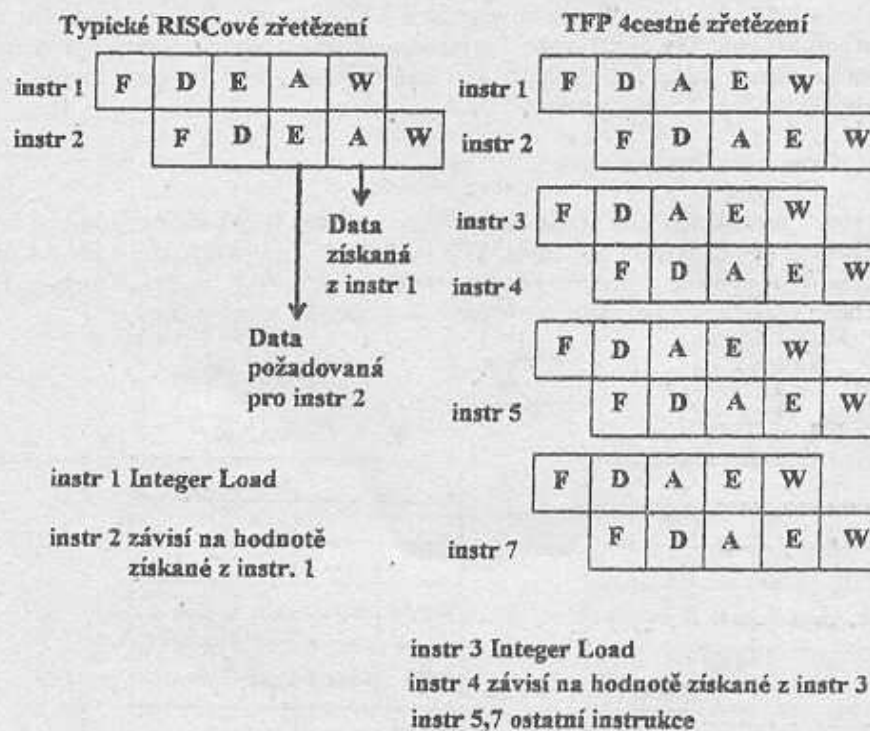
IU obsahuje čtyři druhy cache:

- instrukční cache
- datovou cache
- cache pro predikci větvení
- TLB

Vyhrazený interface dovoluje IU generovat a poskytovat adresové informace do data-stream cache, FPU a příznakové RAM. Přídavná 80-bitová sběrnice umožňuje komunikaci IU a FPU. IU také obsahuje dvě celočíselné aritmeticko-logické jednotky (ALU) a dvě jednotky generující adresy. Tyto funkční jednotky umožňují vykonávání až čtyř instrukcí během jednoho cyklu t.j. dvou celočíselných instrukcí a dvou paměťových přístupů.

Celočíselné zřetězení FDAEW

Čipová sada mikroprocesoru TFP má celočíselné zřetězení se stupni podobnými se stupňům u R3000. Hlavním rozdílem je to, že prováděcí stupeň byl posunut do následujícího stupně zřetězení.



Obr. č. 6.23 Porovnání zřetězení

To pak dovoluje čipové sadě TFP co nejlepší využití doby cyklu. Při jistých podmínkách může mít typické RISC zřetězení vliv na superskalární implementaci. Zřetězení zahrnuje pět stupňů: přivedení (F), čtení a dekódování (D), vykonávání v ALU (E), odkaz na paměťové místo (A) a zápis výsledku do příslušného registru (W). Při FDAEW zřetězení jsou instrukce následující load a instrukce, která je závislá na výsledku tohoto load prováděny v jednom cyklu. Toto prodloužení cyklu kriticky ovlivňuje výkon při superskalárních implementacích, neboť kompilátor musí nyní nalézt čtyři instrukce store pro uložení v odkládacím slotu z důvodu plného využití výpočetních schopností procesoru.

TFP začleňuje nové zřetězené sekvence FDAEW:

- Přivedení a částečné dekódování instrukcí predikce větvení.
- Dekódování instrukcí, čtení ze souboru registrů, vykonání scoreboardu a kontrol závislosti.
- Instrukce load a store a generování požadovaných adres.
- Vykonávání operací v ALU, přístup do datové cache, vyhledávání v TLB, detekce výjimek.
- Zpětný zápis výsledků do souboru registrů.

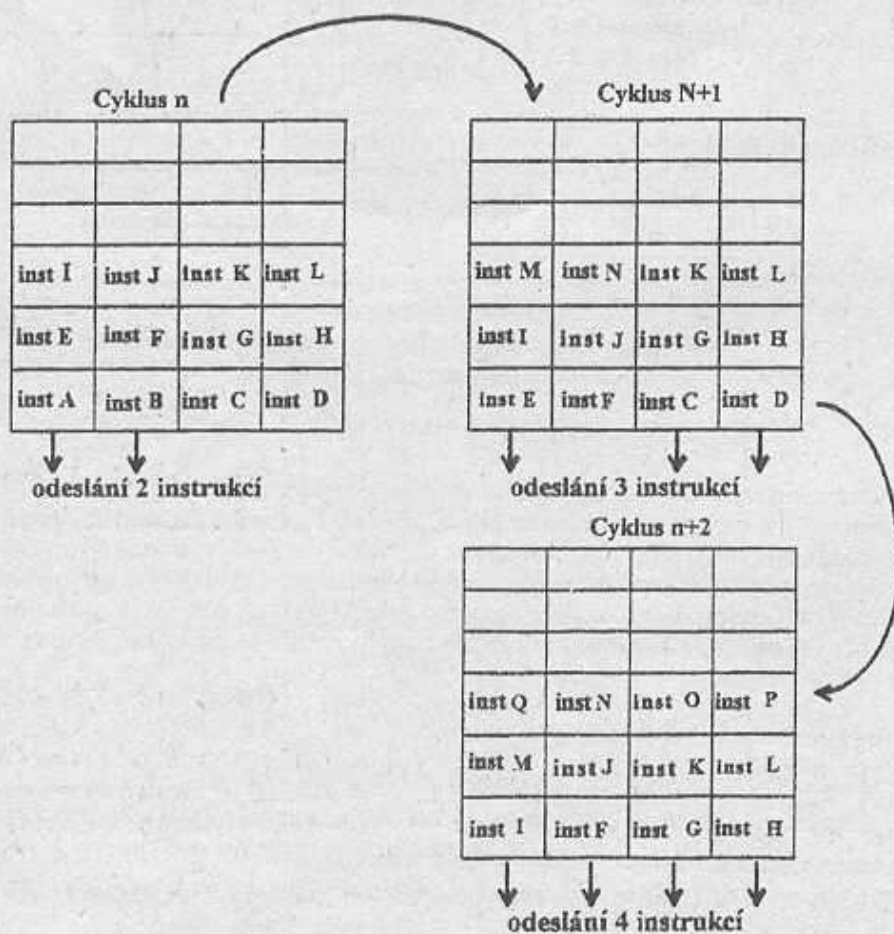
S tímto FDAEW zřetěžením je nyní ovládací slot umístěn před instrukcí load. Tím je umožněno, aby instrukce load byla okamžitě následována podmíněným použitím load. Toto schéma posunuje rozvětvení o jeden cyklus dál a zvyšuje se tak možnost ztráty při větvení, což je závažná ztráta u všech tradičních RISC systémů. K překonání tohoto problému TFP ve skutečnosti předpokládá větvení už ve stupni převedení (F) a snižuje tak pravděpodobnost zpoždění vykonávacího stupně. Zřetězení FDAEW dále umožňuje, aby superskalární procesor RFP zpracoval vícenásobné instrukce okamžitě po instrukci load, a to včetně instrukcí obvykle uvedených v odkládacím load slotu.

Instrukční cache a instrukční příznaková RAM

Instrukční cache obsahuje instrukce, které mají být právě vykonány. Instrukční cache v IU je 16 kB s přímým adresováním. Je uspořádána jako 1024 údajů po 128 bitech. Každý údaj obsahuje čtyři 32-bitové instrukce, jinak každým přístupem na cache se přivedou čtyři instrukce. I-cache je virtuálně indexována s virtuálními příznaky z důvodů snížení požadavků na dekódování adres a v přístupech na cache. Příznaková RAM instrukční cache je používána na zjišťování, zde existuje v I-cache platná instrukce. Příznaková RAM obsahuje 512 údajů t.j. jeden příznak pro každé dva údaje instrukční cache, neboli jeden příznak pro každý řádek I-cache. Každý údaj v příznakové RAM obsahuje 34-bitový příznak a 8-bitový identifikátor adresového místa (ASID), bit platnosti příznaku a dva region bity. ASID rozlišuje instrukce mezi procesy a při multiprocessingu umožňuje instrukcím sdílet I-cache v téže časovém okamžiku. To také zajišťuje, že si dva procesy přistupující na totéž paměťové místo nepřepíší navzájem své instrukce.

Instrukční/příznakové fronty a řízení.

Efektivní využití superskalárního procesoru vyžaduje přístup k několika instrukcím během jednoho cyklu a schopnost je během tohoto cyklu i zpracovat. TFP využívá tvořivého přístupu k přivádění a zpracování instrukcí. IU procesoru TFP přivádí z I-cache během jednoho cyklu čtyři instrukce neboli 128 bitů. Tyto instrukce jsou částečně dekódovány a poté jsou umístěny v 6-stupňové dočasné frontě.



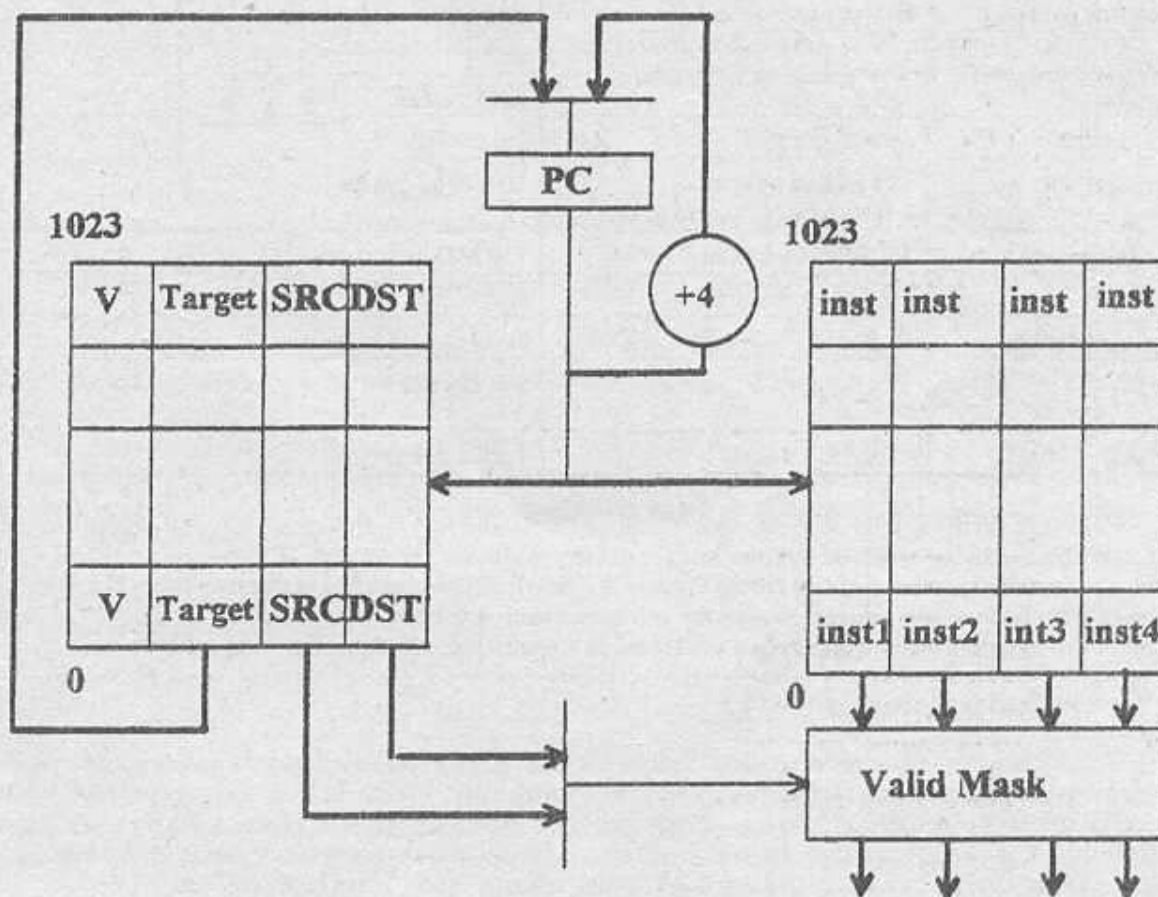
Obr. č. 6.24 Instrukční řízení

Všechny instrukce i instrukce příznakové fronty jsou ukládány do bufferu instrukcí, kde čekají na vykonání. Existence této fronty je nezbytná, neboť některé závislosti mohou zabránit správnému vykonání instrukcí. Částečné dekódování zase redukuje dobu provádění instrukcí ve stupni vykonávání.

X-bar může každý cyklus zpracovat až čtyři instrukce. X-bar se používá k určování, kdy jsou instrukce připraveny k vykonání. Kontroluje stav každé vykonávací jednotky z cyklu do cyklu a určuje v dané posloupnosti vzájemné závislosti mezi těmito čtyřmi instrukcemi.

Cache predikce větvení a predikce větvení

Při zjištění větvení je obsah programového čítače změněn. To ve zřetězené implementaci procesoru způsobí plýtvání jedním nebo i více cykly, a dochází tak ke snižování výkonu procesoru. Výzkum ukázal, že během provádění typické aplikace se příkaz větvení objevuje každou šestou až osmou instrukcí. Proto se u superskalárního procesoru, který vykonává každým cyklem až čtyři instrukce, očekává větvení každý druhý cyklus. Predikce větvení je proto rozhodující při provádění zřetězení a pro zaručení souvislého vykonávání.



Obr. č. 6.25 Cache predikce větvení

Cache predikce větvení procesoru TFP je používána k modifikaci programového čítače (umístění cílové instrukce) v tom případě, kdy se procesor setká s instrukcí větvení. Tato cache pracuje ve spojení s I-cache zahrnující mechanismus predikce větvení prvního stupně. Cache predikce větvení obsahuje 1024 údajů - jeden údaj pro každý údaj v interní cache. Každý údaj je 15-bitový. 10 z těchto bitů je používáno jako cílová adresa predikce větvení na jakýkoliv z 1024 údajů v interní cache. Se zbývajících pěti bitů se první nazývá bit predikce. Jestliže je bit predikce nastaven, tak je odpovídající 10-bitová cílová adresa vložena do programového čítače. V I-cache se skočí na nové místo indexu a instrukce se začnou vykonávat z tohoto místa. Správnost predikce větvení bývá známa až tři cykly po skutečném provedení instrukce větvení.

Tabulka č. 6.2 Parametry pro TFP instrukční cache a cache predikce větvení

Parametr	Instrukční cache	Cache predikce větvení
Umístění	IU	IU
Vstupy /obsah/	128 bitů /4 instrukce/	16 bitů 1 bit Predikce existence instr. větvení 2 bity Umístění čekací instr. 10 bitů Umístění dalšího čtyřnásobného slova 2 bity Zarovnání cílové instr.
Velikost	16 kB	2 kB
Mapování	Přímé Bez hašování	Přímé Bez hašování
Index	Virtuální adresy	Virtuální adresy
Příznak	Virtuální adresy	N/A
Ochrana paměti	Bez parity	Bez parity
Porty	1	1
Prokládání	Jednocestné	Jednocestné
Přístup dat	Jednocyklový	Jednocyklový
Transfer dat	128 bitů za cyklus 4 instrukce (slova) za cyklus	16 bitů za cyklus
cache f.pasmo	1.2 GB za sekundu	150 MB za sekundu
Celkově	1.2 GB za sekundu	N/A
Velikost řádku	32 Bytů nebo 8 slov	N/A
Ztráta	11 cyklů pro DS cache	3 cykly pro aktuální výsledek větvení
Přístup. vzor	1 load za cyklus	1 load za cyklus

Jestliže predikce byla správná, potom jsou ve frontě všechny instrukce za instrukcí větvení správné a není tedy ztracen žádný pracovní cyklus. Jestliže ovšem predikce správná nebyla, potom je zřetězení vyprázdněno a je vypočítávána nová cílová adresa větvení. Ztráta při chybné predikci je tři pracovní cykly. Další čtyři bity v každém údaji cache větvení jsou určeny pro specifikaci, která ze čtyř instrukcí v bloku l-cache obsahuje instrukci větvení, dále jestli je daná větev přijata a kde v těchto čtyřech slovech provádění instrukcí začíná.

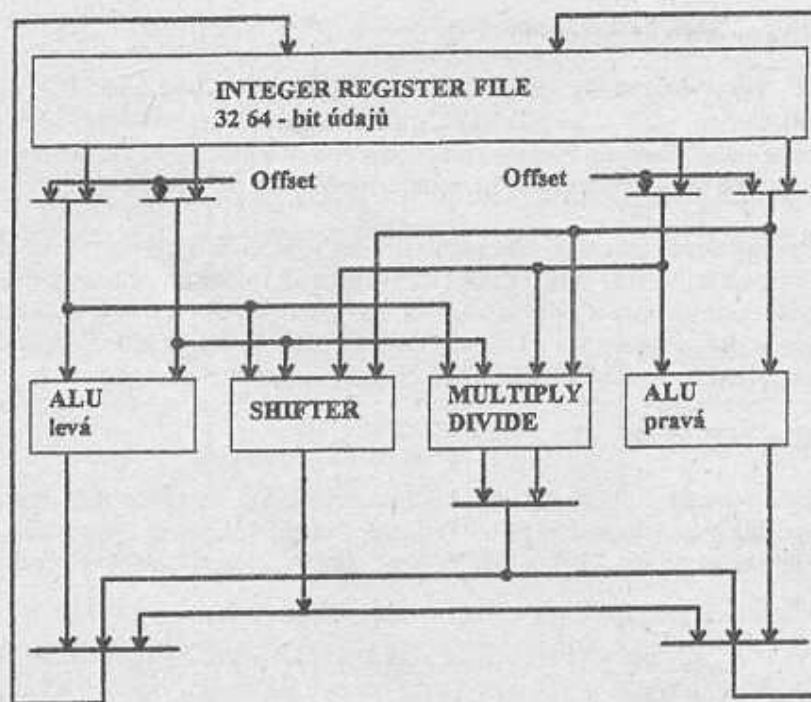
Celočíselný soubor registrů

Celočíselný soubor registrů se skládá z 32 64-bitových údajů a obsahuje data pro operace s pevnou řádovou čárkou a data pro generování adres. Celočíselný soubor registrů používá 9 portů pro čtení a 4 porty pro zápis z důvodů umožnění paralelního vykonávání operací s pevnou řádovou čárkou a pro paralelní přístup k datové cache umístěné na téže čipu. Tyto 4 porty pro zápis definují počet operací, které mohou být prováděny paralelně. Z tohoto důvodu mohou být každý cyklus provedeny dvě operace s pevnou řádovou čárkou a dvě instrukce load.

Čtyři z portů pro čtení a dva z portů pro zápis jsou vyhrazeny k simultánnímu užívání celočíselnou ALU. Čtyři další porty pro čtení jsou použity k zavádění dat z datové cache do souboru registrů. Devátý port pro čtení, port vynechávání, je využit datovou cache pro store a pro nesetříděné load. Při ukládání do datové cache pak existence devátého portu redukuje počet sdílení, která mohou být požadována porty vyhrazenými pro generování adres během operací paměti.

Aritmeticko-logická jednotka

Dvě 64-bitové aritmeticko-logické jednotky, jeden shifter a jedna jednotka pro násobení/dělení vykonávají všechny aritmetické operace s pevnou řádovou čárkou procesoru TFP. Všechny aritmetické operace se vyskytují ve vykonávacím stupni (E) celočíselného zřetězení. Výsledky těchto operací jsou pak zpětně zapisovány do souboru registrů.



Obr. č. 6.26 Aritmeticko logická jednotka

Organizace TLB (Translation lookaside buffer)

TLB převádí virtuální adresy na adresy fyzické. Samostatná TLB v IU ovládá reference instrukcí při nenalezení požadovaných instrukcí v I-cache a ovládá také všechny datové odkazy. TLB obsahuje 384 položek pro snížení pravděpodobnosti nenalezení odkazů v TLB např. při výpočtech rozsáhlých matic. TLB je trojcestně nastavena pro udržování vysokého výkonu. Proměnné délky stránek jsou podporovány jak v uživatelském, tak i v kernel režimu. TLB je double-ported pro umožnění paralelního odkazování. Je rozdělena do dvou částí. Jedna část obsahuje virtuální příznaky (VTAGS), druhá část potom obsahuje aktuální fyzickou adresu korespondující s každým virtuálním příznakem.

Tabulka č. 6.3 Srovnání TLB

Parametr	Power challenge TLB-TFP	Callenge TLB-R4400
Umístění	IU	IU
Obsah	Překlad virtuální adresy na fyzickou	Překlad virtuální adresy na fyzickou
Velikost	384 údajů 1 překlad-údaj	48 údajů 2 překlady /sudé a liché stránky/
Mapování	Třicestné náhodné rozmístění	Plně asociativní náhodné rozmístění
Index	Virtuální adresy	Virtuální adresy
Ochrana dat	Bez parity	Bez parity
Porty	2	2
Přístup	Jednocyklový	Jednocyklový
Čekací doba	Softwarová náplň	Softwarová náplň
Atributy stránek	Nezávislé	Nezávislé
Velikost stránky Kernel	16 kB doplňková velikost TBD	4 kB
Velikost stránky User	16 kBdoplňková velikost TBD	4 kB

Datová cache a příznak datové cache

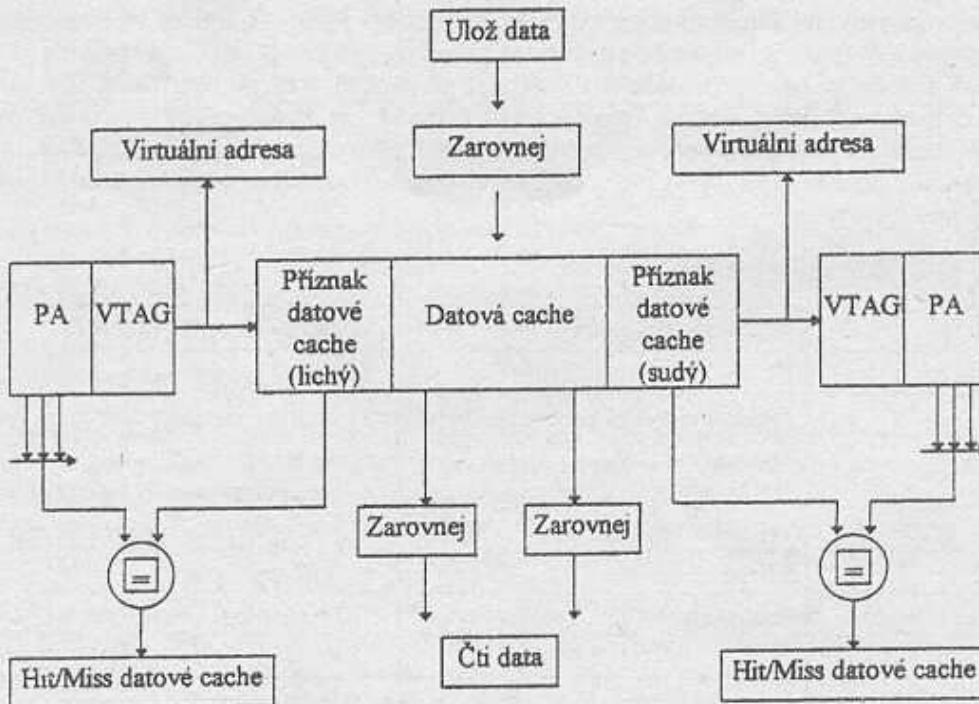
Datová cache v IU o velikosti 16 kB je double-ported. Je uspořádána jako 2048 údajů po 64 bitech. D-cache je virtuálně indexována, má fyzické příznaky a zpracovává přímo zapisovatelný protokol pro udržování konzistence s data-stream cache. D-cache současně umožňuje buď dvě instrukce load nebo jednu instrukci load a jednu instrukci store. Je zde také podporováno zavedení a uložení dat po jednom bytu (osmi bitech).

Příznaková RAM datové cache v IU obsahuje 512 údajů - jeden údaj pro každý 32-bitový údaj datové cache. Všechny položky mají velikost 28-bitů. Fyzická adresa příznaku uložení v příznakové RAM datové cache je porovnávána s překládanou adresou z TLB pro určení, zda jsou požadovaná data v D-cache. Jestliže jsou příznaky shodné, potom se požadovaná data v D-cache nachází. Jestliže ovšem příznaky shodné nejsou, ohlásí se nedostatek požadovaných dat a je zahájen paměťový cyklus do data-stream cache.

Datová cache a TLB

Přístup do datové cache a prohledávání TLB jsou prováděny ve výkonném stupni zřetězení. Tato souběžná činnost určuje, zda požadovaná data jsou v D-cache. Pro zjištění, zda existuje platný překlad virtuální na fyzickou adresu, je zkontrolován TLB a pro zjištění, zda rezidentní data v D-cache jsou platná, je zkontrolována D-cache.

VTAG část TLB určuje, zda v TLB existuje platný překlad virtuální na fyzickou adresu (ve fyzické adresové části). Jestliže virtuální adresa požadovaných dat odpovídá příznaku virtuální adresy v VTAG, pak se jedná o TLB hit (úspěch). V případě neplatného porovnání se jedná o TLB miss (neúspěch). Pokud je určeno, že se překlad virtuální adresy nachází v TLB, je odpovídající část fyzické adresy srovnána s údajem v příznakové RAM datové cache, na základě čehož je pak určeno, zda se jedná o úspěch nebo neúspěch datové cache. Při neúspěchu TLB i datové cache je zahájen externí paměťový cyklus.



Obr. č. 6.27 Datová cache s duálními porty a TLB

Integer operace

IU provádí několik typů integer operací. Každá operace vyžaduje funkční jednotku pro určitý počet cyklů (označované jako staging) a výsledky jsou k dispozici za určitý pevně stanovený čas (označovaný jako latency). Většina integer operací je prováděna jen v jednom cyklu. Některé operace jako například integer

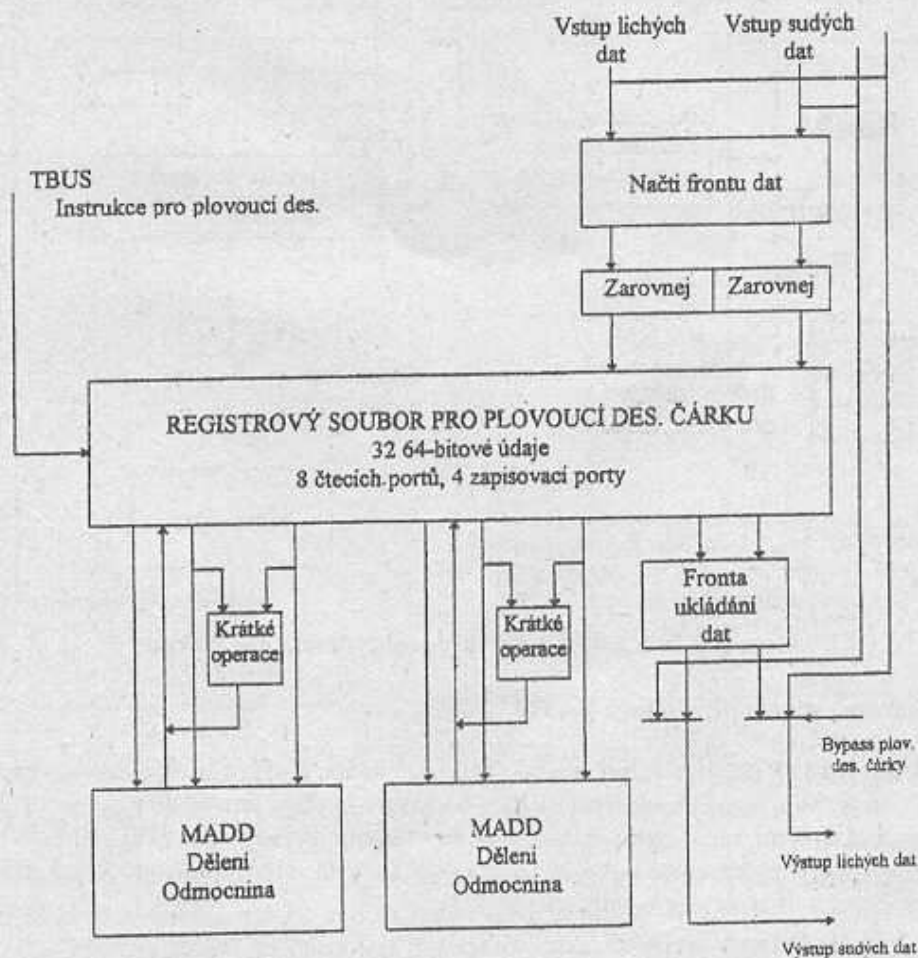
násobení a dělení vyžadují pro vytvoření výsledku dodatečné cykly. Operace TFP integer násobení je jedna z nejrychlejších dostupných implementací. Toto rychlé integer násobení drasticky zlepšuje provádění smyček, které vyžadují integer násobení. Pro integer dělení byl použit rozumný přístup: čas, vyžadovaný operací integer dělení, je funkcí velikosti podílu.

Tabulka 6.4 TFP Integer Latencies

Operace integer jednotky	Latency (počet cyklů)	Staging (počet cyklů)
Přičtení, posun, logické	1	1
Načtení, uložení	1	1
Násobení	4 (32-bitové operandy) 6 (64-bitové operandy)	TBD (32-bitové operandy) TBD (64-bitové operandy)
Dělení	21 (podíl 15 bitů) 39 (podíl 16-61 bitů) 73 (podíl 32-64 bitů)	TBD

Organizace jednotky TFP pro plovoucí desetinnou čárku

FPU čip, 591pinová součástka, provádí všechny funkce s plovoucí desetinnou čárkou pro mikroprocesorovou čipovou sestavu TFP. FPU má dvě plně zřetězené výkonné jednotky (s bypass logikou) dovolující dvě matematické a dvě paměťové operace s plovoucí desetinnou čárkou během každého cyklu. Registrový soubor FPU obsahuje 32 64-bitových údajů a má 8 čtecích a 4 zapisovací porty. Fronty pro čtení a zápis dat (každá pro 32 údajů) poskytují zřetězené rozhraní mezi IU a FPU, usměrňují tok dat a minimalizují nepoužité cykly. FPU nabízí nejvyšší výkon 300 MFLOPS s dvojitou přesností s hodinovou frekvencí 75 Mhz.



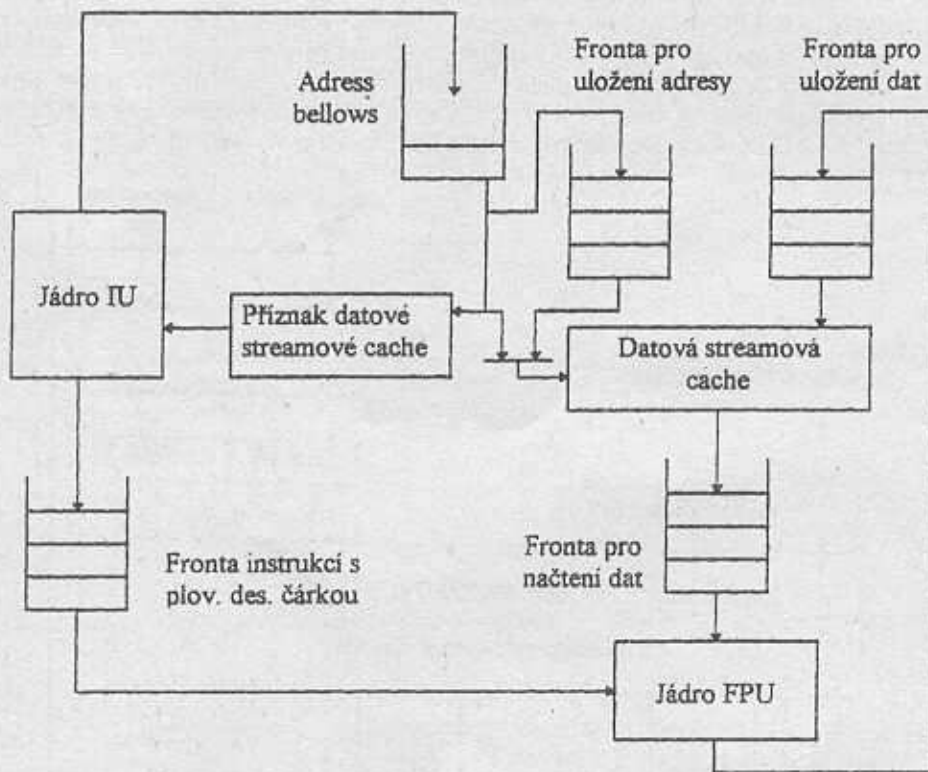
Obr. č. 6.28 Blokový diagram jednotky TFP pro plovoucí desetinnou čárku

Cesta dat s plovoucí desetinnou čárkou

IU umísťuje instrukce s plovoucí desetinnou čárkou ve frontě instrukcí s plovoucí desetinnou čárkou umístěnou v IU. Každý údaj v této frontě je upraven jako čtyřnásobné (quad) slovo. Vyřizovací (dispečerský) mechanismus je podobný jako u IU: v každém cyklu může být IU vyřizeno nula až čtyři instrukce s plovoucí desetinnou čárkou do FPU. Instrukce jsou vyřizovány jen když jsou k dispozici příslušné zdroje v FPU k vykonání instrukcí. Fronta instrukcí s plovoucí desetinnou čárkou poskytuje dočasné uložení pro instrukce s plovoucí desetinnou čárkou do té doby, dokud jakékoliv závislosti, které mohou zabránit jejich provedení (například čekání na dokončení závislého načítání, apod.), nejsou odstraněny.

Vyřizování instrukcí s plovoucí desetinnou čárkou a načítání a ukládání s plovoucí desetinnou čárkou do data-stream cache je řízeno IU. IU je zodpovědná za generování všech adres a kontrolních signálů pro načítání a ukládání s plovoucí desetinnou čárkou do data-stream cache. S přístupy, které selžou v data-stream cache a vyžadují rozhraní s hlavní pamětí, je manipulováno pomocí externího kontroleru cache. Během těchto ztrátových cyklů je FPU používána jen k přesunu dat z načítací datové sběrnice do ukládací datové sběrnice.

Při načítání dat z data-stream cache jsou data s plovoucí desetinnou čárkou umístěna do načítací datové fronty. Při ukládacích operacích jsou výsledky uloženy do ukládací datové fronty. V okamžiku, kdy je k dispozici odpovídající informace o adrese z příznakové RAM, jsou data zapsána do data-stream cache.



Obr. č. 6.29 Cesta dat TFP s plovoucí desetinnou čárkou

Data-stream cache a příznaková RAM

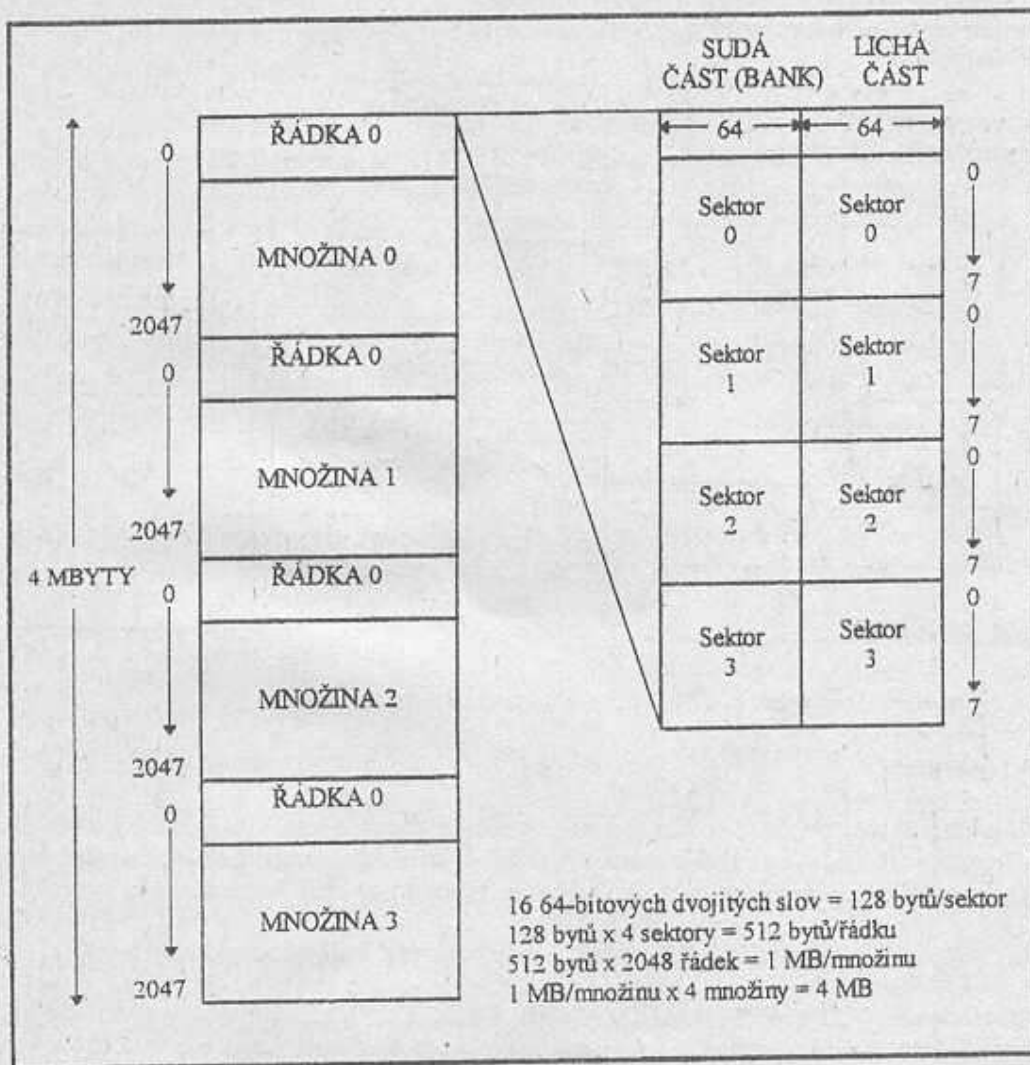
Čipová sestava TFP používá velmi velkou 4MB, asociativní čtyřcestně množinovou cache s externím řetězením dat. Velká cache spojená s blokovými daty podstatně zlepšuje provádění inženýrských a vědeckých aplikací s velkými datovými množinami. Nastavená asociativita zvyšuje efektivní velikost cache. Studie ukázaly, že asociativní čtyřcestně množinové cache vykazují četnosti selhání podobné těm, které byly zjištěny u neasociativních cache s dvojnásobnou velikostí.

Data-stream cache je implementována se čtyřmi 1MB, nezásvkové SIMM moduly. Každý SIMM modul má samostatné sběrnice pro načtení a uložení dat, což eliminuje čas pro průchod sběrnice a dovoluje

současné načítání a ukládání dat do data-stream cache. Oddělené sběrnice pro načtení a uložení dat dovolují, aby načítací a ukládací operace byly plně zřetězeny, a aby bylo možno efektivně číst a zapisovat do data-stream cache každý cyklus. 4MB data-stream cache je rozdělena mezi sudé a liché 2MB části (banks). Každá část se skládá ze dvou SIMM modulů.

Příznaková RAM data-stream cache je asociativní čtyřcestně množinová a je rozčleněna na 2048 údajů po 128 bitech. Každý 128-bitový údaj obsahuje 32 bitů informací pro každou ze čtyř množin. 32 bitů je dále rozděleno na 20-bitovou adresu příznaku, 4 virtuální synonymní bity a 8 stavových bitů.

Každá část (bank) data-stream cache používá uživatelskou příznakovou RAM pro adresování cache. Každá identická příznaková RAM poskytuje 2-bitovou hodnotu adresy k synchronním statickým RAM (SSRAM) data-stream cache. Tyto dva bity jsou používány k určení toho, která množina asociativní čtyřcestně množinové data-stream cache obsahuje rezidentní data. Duální příznakové RAM dovolují také souběžnou a nezávislou činnost částí (banks) data-stream cache v každém cyklu.



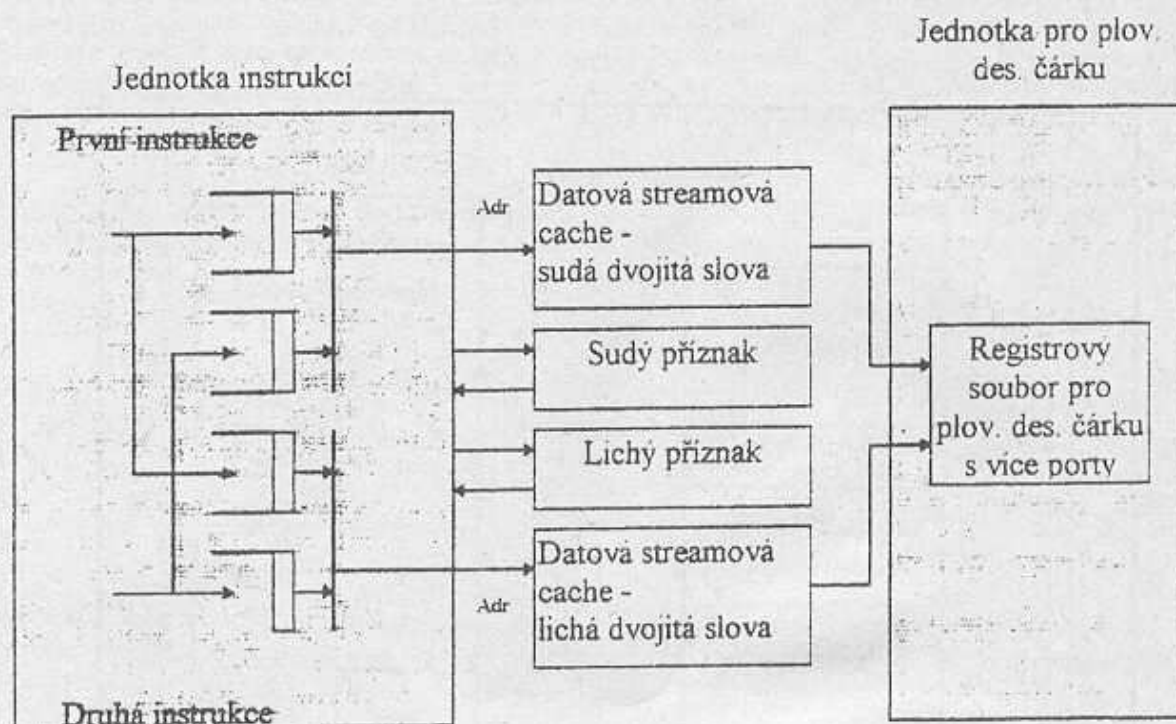
Obr. č. 6.30 Organizace TFP data-stream cache

Návrh prokládání

Kromě toho, že je množinově asociativní, je data-stream cache také dvoucestně prokládaná. Toto poskytuje minimální šířku pásma paměti vyžadovanou k efektivnímu využití obou funkčních jednotek pro plovoucí desetinnou čárku souběžně každý cyklus. Toto také umožňuje skrytí přístupových časů do data-stream cache.

Návrh poskytuje dva 64-bitové operandy (jedno 64-bitové načtení nebo uložení pro každou funkční jednotku pro plovoucí desetinnou čárku) pro FPU každý cyklus s efektivní přenosovou rychlostí 1,2 GB za sekundu. Toto umožňuje každé funkční jednotce provést násobení/přičtení v každém cyklu.

2x2 crossbar řadí odkazy vstupní paměti do lichých a sudých dvojitých slov tak, aby operace mohla být provedena paralelně. Ideálně sudý a lichý odkaz paměti může být dokončen v jednom cyklu. Pro všechny ostatní situace paměťových odkazů on-deep fronta (the address belows) pomáhá v řešení konfliktů paměťové části (bank). Samostatný registr je používán k buferování paměťových operací a k přeuspořádání adresových odkazů jedním cyklem tak, aby liché a sudé paměťové operace co nejlépe využívaly dostupné cykly. Tento jednoduchý rys, doplněný schopností compileru rozpoznat lichou/sudou polaritu adresy, značně omezuje konflikty částí (banks).



Obr. č. 6.31 TFP prokládaná data-stream cache

FPU operace

FPU provádí tři typy aritmetických operací pro plovoucí desetinnou čárku. Tyto jsou označovány jako krátké (short), řádné (regular) a dlouhé (long). Každá operace vyžaduje funkční jednotku pro určitý počet cyklů (označovaný jako staging) a má určitý pevně stanovený čas (označovaný jako latency), za který jsou k dispozici výsledky.

Krátké operace (MOV, NEG, ABS) vyžadují jeden cyklus a mají jednocyklovou latenci.

Řádné operace (ADD, SUB, MADD, MSUB, MUL, CONVERT) vyžadují jeden cyklus a mají 4cyklovou latenci. Řádné operace vyžadují jeden, dva nebo tři zdrojové operandy z registrového souboru FP a dokončení zabírá čtyři cykly. Příštím cyklem je výsledek zapsán zpět do registrového souboru FP. Bypass logika, která je na čipu, usnadňuje provedení datově závislých instrukcí s plovoucí desetinnou čárkou.

Dlouhé operace mají proměnlivé časy latence a stagingu.

Kromě toho jsou definovány načítací a ukládací operace s plovoucí desetinnou čárkou. Pokud jsou zřetězeny, mají tyto operace zdánlivou jednocyklovou latenci, přestože zabírají 5 cyklů při procházení přes data-stream cache.

Podpora multiprocessingu

Procesory MIPS R4400 i TFP implementují komplexní cache, schopné podpory mnoha s cache souvisejících schémat a synchronizaci pro účinnou integraci těsně spojených mikroprocesorových systémů.

Řádka primární cache a Řádka sekundární cache mohou mít jeden z těchto čtyř stavů:

- a) neplatný
- b) sdílený
- c) čistě exkluzivní
- d) částečně (dirty) exkluzivní

Na procesoru MIPS R4400 primární i sekundární datové cache používají strategii zápis zpět (write back). Na základě této strategie instrukce store zapisují data do cache, nicméně nezapisují modifikovanou řádku sekundární cache zpět dokud není řádka sekundární cache nahrazena. Obsah primární cache je udržován jako řádná podmnožina obsahu sekundární cache.

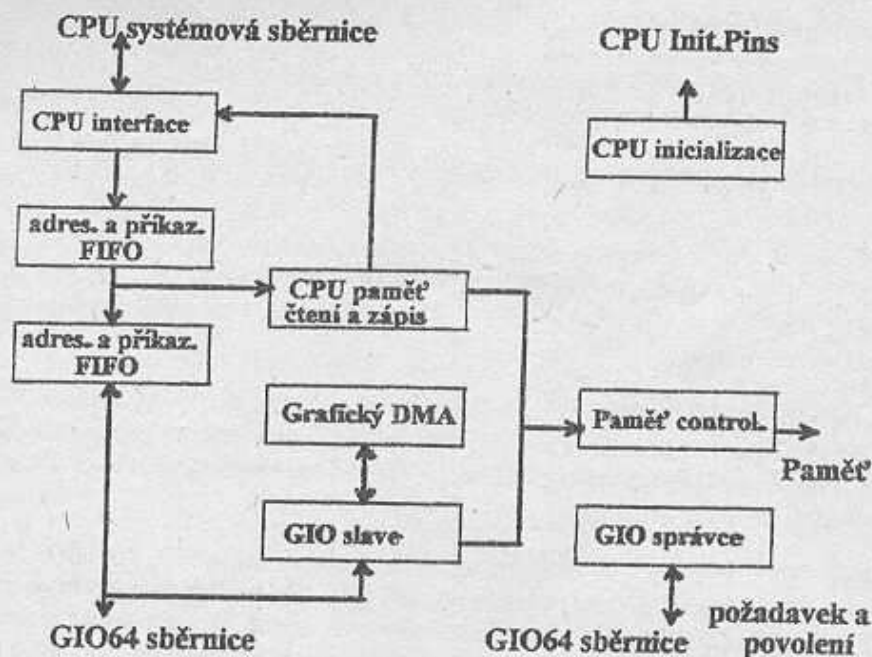
Na procesoru MIPS TFP používá data-stream cache strategii zápis zpět, tak jak bylo popsáno výše. Primární datová cache používá strategii zápis přes (write through). Na základě této strategie integer instrukce store zapisují data do primární datové cache i do data-stream cache. Stejně jako u strategie zápisu zpět, ukládací instrukce nezapisují modifikovanou řádku data-stream cache dokud není řádka cache nahrazena. Obsah primární datové cache je udržován jako řádná podmnožina data-stream cache.

Podpora synchronizace procesoru je poskytována pomocí instrukcí Load Linked a Store Conditional, což poskytuje jednoduchý mechanismus pro generování všech společných synchronizačních částí (test-and-set otestuj a nastav, bit-level locks uzamčení bitových úrovní, semaforey, počítadla a sequencery) - všechno bez dodatečného hardwaru.

2. MC ASIC

Memory controller (obr. č. 6.21) je standardní čip připojený na CPU modul pomocí CPU sběrnice, dále je propojený s GIO64 sběrnicí a má adresové a řídicí vazby připojené na hlavní paměť. Má tyto funkce:

- Tvoří interface mezi hlavní pamětí a CPU jádrem.
- Jako DMA řadič obsluhuje všechny požadavky paměti z grafického systému nebo jiných zařízení napojených na sběrnicí GIO64.
- Chová se jako správce systému pro sběrnicí GIO64.
- Jedním slovem zpřístupňuje CPU na zařízení sběrnice GIO64 a do grafického systému.
- Při obsluze předává přerušení z INT2 ASIC do CPU.
- Inicializuje CPU, vykonává požadavky CPU, ošetřuje paměť a kontroluje parity dat v paměti.
- Zvláštní schopností MC je podpora virtuálních požadavků DMA na procesy běžící v CPU. Proces může také nastavit deskriptor v MC a potom požádá mechanismus DMA o start. Paměťové adresy použité v DMA deskriptoru jsou virtuální paměťové adresy, které MC překládá pomocí TLB a unixových stránkových tabulek na fyzické paměťové adresy. Když je DMA operace ukončena, může být generováno přerušení (které je umožněno v MC) a informován CPU.



Obr. č. 6.32 MC ASIC

3. SBĚRNICE

IO64 sběrnice.

GIO64 sběrnice, hlavní systémové sběrnice pracují s plně 64 bitovými daty. GIO64 je pro Indigo workstation hlavní sběrnici a je navržena pro velkou rychlost přenosu dat. Je spojena s hlavním systémem: Jádro procesoru, hlavní paměť, I/O systém, grafický systém a ostatní boardy připojené na GIO64 expanzní sloty. Je to synchronní, multiplexovaná address/data sběrnice v Burst módu a je časována nezávisle na CPU.

EISA sběrnice

Konverzní protokol sběrnice EISA subsystém je složen z EISA interface jednotky, běžného ASIC čipu a Intel 82350DT EISA čipsetu. EISA interface jednotka obstarává interface pro vyšší výkon GIO64 sběrnice a Intel EISA čipsetu. EISA interface jednotka vykonává následující:

- výběrový byte swapping
- správce EISA
- ovládání sběrnice EISA ovladači

EISA interface jednotka má schopnost převodu Burst pro data v následujících frekvencích:

- 25 MB/sec. pro GIO 25 MHz
- 33,3 MB/sec. pro GIO 33 MHz

Intel 82350DT čipset je sestaven z EISA kontroleru sběrnice (EBC), ISP (Periferní integrovaný systém) a ze dvou bufferů EISA 82350DT čipset vykonává následující:

a) EBC

- podpora EISA/ISA paměťových cyklů
- podpora EISA/ISA wait/nowait cyklů
- podpora EISA Burst cyklů
- generace EISA signálů pro ISA

- podpora A,B,C DMA cyklů
- podpora kompatibilních DMA cyklů
- kontrola bufferů EISA

b) ISP

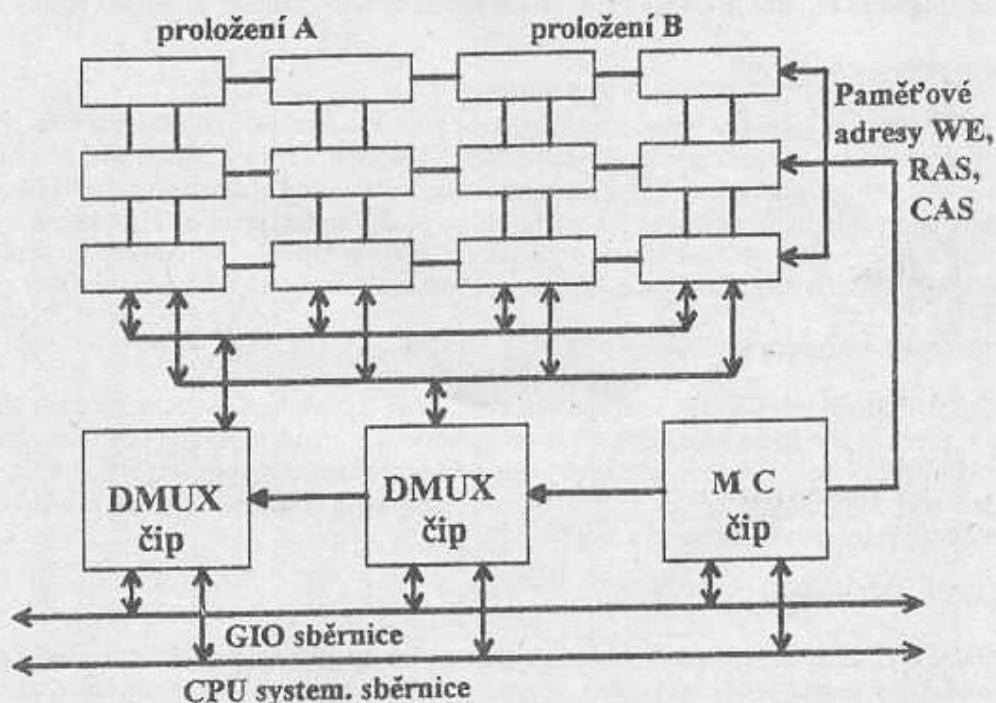
- obstarává vyšší DMA funkce
- obstarává EISA kontrolu přerušení
- obstarává EISA čítač/časovač
- obstarává EISA timeout logic

c) EISA buffery

- obstarávají datové bufferování a swappovací funkce
- obstarávají adresové bufferování a latch funkce

Hlavní paměť

Hlavní paměť je řízena dvěma čipy DMUX ASIC a poskytuje systémový přístup k větší DRAM. Paměť obsahuje standardní off-the-shelf 36 bitů širokou DRAM SIMM (která musí mít 80 ns RAS a schopnost rychlého stránkování). DMUX čipy prokládají SIMM a vytvářejí tak 64 bitů široký obousměrný prokládaný paměťový systém, čímž podporují zvětšené frekv. pásmo R4400SC/R8000 CPU.



Obr. č. 6.33 Hlavní paměť

Hlavní paměť má velikost od 32 MB do 640 MB. R4400/R4600 CPU boardy mají 12 SIMM vstupy realizované ve třech skupinách po čtyřech. Každá skupina vstupu musí používat SIMM stejné velikosti, ale SIMM velikosti se mohou lišit pro různé skupiny, takže je umožněna konfigurace různé velikosti paměti.

Maximální kapacita je:

- 96 MB s použitím všech 8 MB SIMM
- 192 MB s použitím všech 16 MB SIMM
- 384 MB s použitím všech 32 MB SIMM
- 640 MB s použitím všech 64 MB a 32 MB SIMM

6.6.4 Procesor MIPS R10000

Zde se zaměříme na procesor MIPS R10000, který je v dnešní době nejvýkonější a s kterým pracují dnes už všechny řady počítačů Silicon Graphics.

Když počátkem roku 1995 ohlásila firma MIPS předpokládané výkonové parametry svého zcela nového procesoru pro nejnáročnější aplikace (označovaného kodovým znakem T5). Dnes již můžeme s potěšením konstatovat, že firma MIPS jako jedna z mála firem produkujících nové procesory svá prohlášení bezesbytku splnila a v oblasti výkonu dokonce i značně překonala.

Parametry procesoru

Výsledné hodnoty výkonu procesoru na úvodní frekvenci 200 MHz jsou 9 SPECint95 a 19 SPECfp95 (peak), což překonává jak původně udávané minimální hodnoty, tak i výkonové parametry všech v tomto okamžiku dodávaných konkurenčních procesorů. Kromě toho firma NEC, která je prvním výrobcem procesoru R10000, koncem ledna 96 ohlásila první zvýšení taktovací frekvence na 275 MHz, což posunuje výkonové parametry do oblasti 12 SPECint95 a 24 SPECfp95 (peak). Doposud vedoucí procesory Alpha 21164 na frekvenci 333 MHz poskytují výkon 9 SPECint95 a 13 SPECfp95, HP s novým procesorem PA8000 výsledky ještě neohlásilo. Kromě procesoru R10000, uvedla firma MIPS v lednu 96 ještě nový procesor R5000, který sice využívá rysy z vývoje procesoru R10000, ale je zaměřen spíše do oblasti menších a středních pracovních stanic, případně i high-end PC, kde je kladen důraz především na solidní výkon za minimální cenu.

Využití procesoru R10000

Procesor R10000 je určen pro nejnáročnější aplikace jak v oblasti zpracování dat (databáze, fileservery), tak i pro výpočetně velmi náročné aplikace (superpočítače). Díky tradici firmy Silicon Graphics bude tento procesor velmi silný i v oblasti grafických aplikací, kde jeho velký výkon umožní provozování ještě náročnějších aplikací. Procesor podporuje multiprocessing, takže jeho hlavní použití bude v první řadě pro datové, výpočetní a grafické servery a superpočítače. Nárůst výkonu procesoru R10000 ve srovnání s dřívějšími procesory MIPS je pro R4400/250MHz zhruba dvojnásobek v celočíselných operacích a trojnásobek v plovoucí řadové čáře.

Architektura procesoru R10000

R10000 je moderní jednočipový mikroprocesor vyráběný standardní CMOS technologií (0,35 μ m) s napájením 3,3 V. Přestože je umístěn na jediném relativně malém plátku křemíku (zhruba 18 x 18 mm), obsahuje zhruba pětikrát více prvků a funkčních jednotek, než by bylo nezbytné ke stejně dobrému výkonu v testech SPEC. Tyto funkční jednotky zajišťují implementaci algoritmů a technik, které se starají o plynulé zásobování všech výpočetních jednotek procesoru instrukcemi a daty.

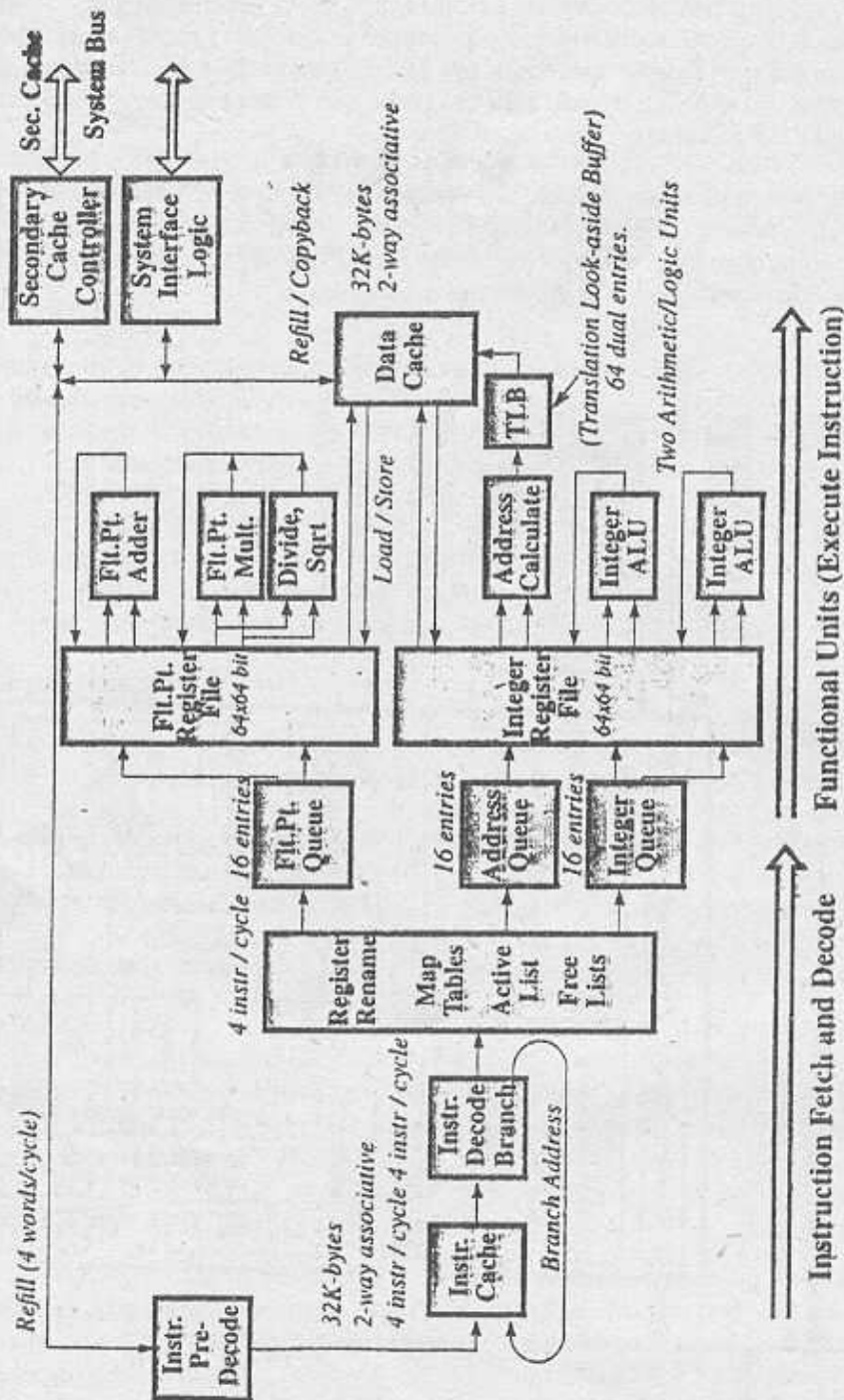
Celková stavba R10000

R10000 je čtyřcestný superskalární procesor s plně 64-bitovou RISC architekturou, který je schopen dekódovat a provést čtyři instrukce v jediném taktu. Je vybaven čtyřmi aritmetickými jednotkami (2 pro výpočty v plovoucí čáře a 2 pro celočíselné operace) a jednou jednotkou Load/Store pro přesuny dat. Interně je tedy schopen provádění až pěti instrukcí současně. Zpracovává RISC instrukční sadu MIPS IV, která je rozšířením starších sad a je k nim zpětně kompatibilní. Procesor poskytuje špičkový výkon 88 MIPS (400 MFLOPS). Mezi nejvýznamnější prvky architektury procesoru z hlediska výkonu patří vykonávání instrukcí mimo pořadí (out-of-order execution), přejmenování registrů (register renaming), predikace skokových instrukcí (branch prediction), spekulativní provádění instrukcí, předvýběr instrukcí (prefetch), neblokující cache (non-block cache), velká propustnost cache paměti a jejich asociativita a další. Všechny tyto rysy napomáhají plynulému zásobování procesoru instrukcemi a daty a zvyšují jeho odolnost proti nepříznivému rozložení instrukcí a operandů ve velkých programech. Součástí procesoru je i interní systémový řadič pro podporu multiprocessingu.

Primární a sekundární Cache

R10000 obsahuje přímo na čipu procesoru relativně velkou primární cache o velikosti 32 kB pro instrukce a 32 kB pro data. Cache pro data je dvoucestně prokládaná a každá prokládaná část je dvoucestně asociativní, což umožňuje současný výběr více operandů současně. Cache je též virtuálně indexována a fyzicky

R10000 - Block Diagram



Obr. č. 6.35 Blokové schéma procesoru MIPS R10000

Vykonávání instrukcí mimo pořadí

Každá dekodovaná instrukce je řazena do jedné ze tří front, z nichž každá může dynamicky plánovat pořadí provedení instrukcí. Plánování je založeno na dostupnosti požadované operační jednotky a na dostupnosti dat pro provádění operací. Instrukce jsou do procesoru čteny v pořadí jak jsou uloženy v programu, uvnitř procesoru však jsou instrukce prováděny a ukončovány v pořadí závislém na obsahu vnitřních front v procesoru. Procesor zvládne až 32 současně rozpracovaných instrukcí v různém stavu dokončení a přitom zjišťuje, aby na výstupu výsledky prováděných operací byly opět "srovnány" do správného pořadí. Další mechanismy procesoru se starají o to, aby rozpracované instrukce mohly být co nejdříve dokončeny, nicméně nedostupnost některého operandu nezabraňuje procesoru v práci na jiných instrukcích.

Uvnitř procesoru jsou instrukce rozdělovány do tří front, jedna na aritmetické jednotky pro operace v plovoucí řadové čárce, druhá na aritmetické jednotky pro celočíselné operace a třetí fronta tzv. adresní, která uchovává instrukce a adresy dat pro přesuny (pro instrukce Load/Store). Každá fronta je organizována jako fronta FIFO a má kapacitu 16 instrukcí. V jednom taktu je každá fronta schopna pojmout až čtyři instrukce a provést dvě operace.

Výsledkem vykonávání instrukcí mimo pořadí na tak velkém počtu instrukcí (spolu s dalšími rysy uvedenými dále) je, že procesor poskytuje poměrně stabilní výkonost velmi málo závislou na typu aplikace, na její velikosti a na konkrétních podmínkách provozování, což až donedávna vůbec neplatilo. Další velkou výhodou je, že i bez přeložení či ladění kódu pro tento procesor lze aplikace vytvořené pro R4400 nebo R8000 provozovat s velmi dobrým využitím procesoru, protože procesor je schopen si za běhu programu přehodit pořadí instrukcí tak, aby mu vyhovovalo a nahradit tak částečně rekompilaci. Vzhledem k tomu, že procesor podporuje instrukční sadu předchozí generace procesorů MIPS. Existující programy mohou být přímo spuštěny a i bez jakékoliv úpravy využít výhod procesoru R10000. V neposlední řadě tento rys odbourává nutnost provést některá rozhodnutí o nejlepší variantě provedení až do doby běhu programu, kdy lze udělat rozhodnutí v řadě případů lépe.

Předvýběr instrukcí

Procesor R10000 umí provádět předvýběr instrukcí. Používá se pro preventivní ošetření situace, která by nastala po provedení skokové instrukce (ať už podmíněné nebo nepodmíněné) na tak vzdálené místo programu, že by další instrukce nebyly k dispozici v primární cache a procesor by musel čekat na jejich zpřístupnění ze sekundární cache nebo dokonce z hlavní paměti. Předvýběr instrukcí zajišťuje, že procesor má prakticky vždy v primární cache k dispozici instrukce pro pokračování systému.

Predikace skoků a spekulativní vykonávání instrukcí

Ztráty výkonu u současných mikroprocesorů spojené s vykonáváním podmíněného příkazu nebo podmíněného skoku by bez ošetření byly neúměrně vysoké, protože rozhodnutí o pokračování programu bude známo až během několika dalších cyklů procesoru a mezi tím by bylo možno vykonat velké množství instrukcí. Různí výrobci tento problém různě řeší. Poměrně častou metodou je predikace skoků na základě nějaké historie programu (hledá se větší pravděpodobnost správné cesty), ale tato metoda je schopna dosáhnout podstatného zlepšení účinnosti jen u některých typů kódů. Proto je procesor R10000 vybaven kromě predikace skoku i mechanismem pro spekulativní vykonávání instrukcí.

Algoritmus predikace skoků začíná už v okamžiku uložení skokové instrukce do primární cache paměti, kdy je každá skoková instrukce vybavena jednobitovým příznakem indukujícím, že jde o skokovou instrukci. Historie všech skoků je v procesoru ukládána do paměti provedených skoků (branch history RAM), která má pro každou vykonanou skokovou instrukci 2 bity. Tato paměť je po provedení každého skoku aktualizována. Na jejím základě je prováděno rozhodnutí o tom, které větvi programu bude přidělen příznak vyšší priority zpracování dalších instrukcí (tuto větev budeme nadále označovat jako hlavní).

Na predikaci skoků bezprostředně navazuje spekulativní vykonávání instrukcí, což je mechanismus, umožňující zpracovávat obě varianty programu (hlavní i vedlejší) nezávisle na sobě a teprve v okamžiku rozhodnutí o správné cestě buď pokračovat v provádění hlavní větve a výsledky vedlejší zrušit nebo naopak udělat korekci a výsledky hlavní větve nahradit výsledky vedlejší větve a nadále pokračovat v jejím provádění. Mechanismus je schopen v jednom okamžiku mít rozpracovány maximálně 4 nezávislé větve programu, což pro většinu situací je naprosto dostatečné. Díky spolupráci obou uvedených mechanismů jsou ztráty výkonu procesoru spojené s podmíněnými příkazy minimální.

Přejmenování registrů

Procesor R10000 obsahuje 64 fyzických registrů, ale programy pracují jen s 32 logickými registry. Důvodem k tomu je to, že procesor sám hospodaří s volnými registry a přiřazuje jim logická jména, s nimiž pracují instrukce. To umožňuje například elegantní provádění spekulativního vykonávání instrukcí, kdy obě větve programu mohou pracovat nad stejnými logickými registry, ale každý se svými hodnotami, závislými na provedené činnosti. Překlopení nesprávné větve programu pak znamená v podstatě jen uvolnění registrů

příslušných k nesprávné větvi programu v mapě registrů bez nutnosti fyzického přesunu dat do správných registrů (a tím ztráty cyklu). Přejmenování registrů se používá i pro ošetření vzájemných závislostí mezi operandy současně prováděných instrukcí a při potřebě uložení mezivýsledků rozpracovaných instrukcí.

Tabulka č. 6.5 Porovnání procesorů MIPS a INTEL:

Parametr	R5000-200	Pentium-133	Pentium Pro-200	R10000-200	R10000-275
Primary Data Cache Size	32 kBytes	8 kBytes	8 kBytes	32 kByte	32 kByte
Primary Inst Cache Size	32 kBytes	8 kBytes	8 kBytes	32 kByte	32 kByte
ISA Support	MIPS-IV	x86	x86	MIPS-IV	MIPS-IV
Secondary Cache Support	Yes	No	Yes	Yes	Yes
Multiprocess. Support	Yes	No	Yes	Yes	Yes
Multiple Inst. Issue	Yes 2-way	Yes 2-way	Yes 3-way	Yes 2-way	Yes 2-way
Separate Int. and FP ALUs	Yes	Yes	Yes	Yes	Yes
Size	84 mm ²	93 mm ²	196 mm ²	324 mm ²	324 mm ²
# Number of Transistors	3.6 million	3.3 million	5.5 million	-	-
SpecInt95	5.5	4.1	8.1	9.0	12.0
SpecFP95	5.5	2.5	6.0	19.0	24.0

6.7 Procesory SPARC

Firma Sun [3, 41] nabízí své systémy na dvou řadách procesorů. První je MicroSPARC, určený pro menší systémy a pro notebooky a druhou Sparc (obdoba PowerPC a POWER u IBM nebo Alpha 21066 a Alpha 21064 u DEC). SPARC znamená Scalable Processor Architecture nebo-li přizpůsobivá architektura procesoru. Přizpůsobivost těchto procesorů RISC je dána zejména nezávislostí jejich architektury na použité polovodičové technologii. Firma Sun navíc uvolnila architekturu tohoto procesoru, takže jej v současné době vyrábí několik firem, z nichž nejdůležitější jsou Texas Instruments a Fujitsu. Ve spolupráci s první z nich byl vyvinut první superskalární SPARC V8, jinak též SuperSPARC a nyní i nový SPARC V9, resp. HyperSPARC, jímž firma Sun hodlá konkurovat procesoru Alpha, MIPS a POWER. Pro SPARC V8 se jako I/O sběrnice používá firemní MBus.

Přímým předchůdcem obou uvedených byl SPARC V7, který ještě neobsahoval jednotku pro správu paměti MMU. V kooperaci s Fujitsu byl vyvinut MicroSPARC II, což je implementace SPARC V8 na malý čip pracující s napětím 3,3 V a na frekvenci od 50 do 120 MHz. Čip je plně statický a se správou napájení dovede celkem udatně spořit energii. Jako I/O sběrnice je u procesoru MicroSPARC použit firemní SBus. Předchůdcem procesoru MicroSPARC II byl MicroSPARC I, který se od něj lišil interní datovou a instrukční cache.

6.7.1 Procesor UltraSPARC

UltraSPARC je vysoce výkonný, vysoce integrovaný procesor implementující 64-bitovou SPARC V9 RISC architekturu. Je schopen uržet a realizovat až 4 instrukce za jeden cyklus, souměrně podle požadavků a podmínkách rozvětvení a možnostech cache paměti. To je hlavní způsob z asynchronního hlediska jednotky přivádějící instrukce a data do konce spojení (fronty). Instrukce předpovězené k uskutečnění jsou výsledkem (v programovém pořadí) v násobné funkční jednotce, realizované paralelně a pro přidání paralelismu, může

dokončit mimo pořadí. Podle pořadí dalších instrukcí se zvětšuje počet instrukcí zpracovaných za jeden cyklus, instrukce ze dvou základních bloků (tj. instrukce před a za rozvětvením podmínky) může skončit v jedné skupině (části).

UltraSPARC je plně implementovaný 64 bit sparc v9 architektura. Podporuje 44 bitovou virtuální adresaci a 41 bitovou fyzickou adresaci prostoru. Jádro instrukční sady bylo rozšířeno o grafické instrukce, které přináší většinou běžné operace spojené s dvoudimenzionálním obrazovým zpracováním, 2 a 3 dimenzionální grafiku a obrazové kompresní algoritmy, a paralelní operace na pixel, data s 8 a 16 bitovými komponentami.

Doba provedení aplikace je závislá na třech faktorech:

- počet instrukcí generovaný kompilátorem,
- průměrný počet cyklů potřebný na instrukci,
- doba jednoho cyklu procesoru.

Architektura a implementace ultrasparcu spojena s novými technikami kompilátoru se snaží redukovat každou komponentu, dokud není tato varianta horší.

Průměrný počet cyklů na instrukci závisí na architektuře procesoru a na schopnosti kompilátoru zvýhodnit hlavní HW nabídky. UltraSPARC realizační jednotky (ALU, LD/ST, větve, dvě grafické, dvě float).

Redukování instrukcí závisí na přetížení. Procesory ultrasparc mají krátké latenční doby operací a zajišťují vnější řízení mezi jednotkami nebo mezi jednou jednotkou. Prázdná cache je obvykle zapříčiněna pomalými (dlouhými) spoluproceny v CPI, je redukována symboly při použití propojených jednotek (prefetch unit, vstupní buffer a uchovávací buffer), které pracují asynchroně se zbytkem spojení.

Jiné rysy, jako úplný propojovací interface k externí cache (E cache) a podpora spekulativního nahrávání, spolu s důmyslnými technikami kompilátoru - jako softwarové spojení a plánování křížení bloků.

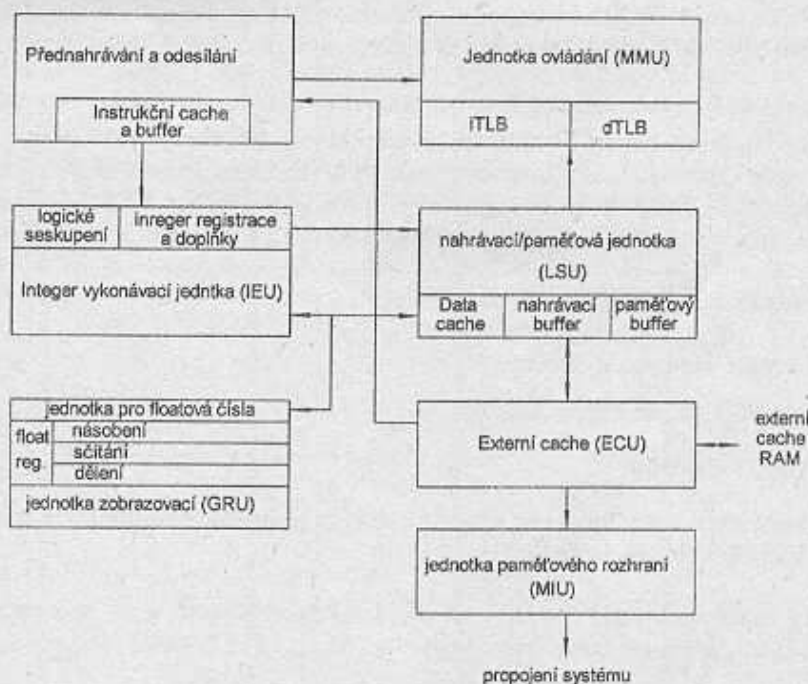
Symetrická architektura musí být schopná poskytnout pomalý CPI bez porušení časového cyklu. Některé z ultrasparcových rysů architektury je, spojování agresivní implementací a state-of-the-art technologii, to umožňuje dokončit instrukce v krátkém cyklu (tab. 6.6) spojení je organizováno tak, že větší část, krátký latenční čas a násobně vnější okruhy nenaruší hodnotově časový cyklus.

Tabulka č. 6.6 Přehled komponent procesoru:

	UltraSPARC I	UltraSPARC II
Technologie	0,5 μ CMOS	0,35 μ CMOS
časový cyklus	7 ns a rychlejší	4 ns a rychlejší

Blokový diagram popisuje následující komponenty

- přednahrávací a odesílací jednotka (PDU), včetně logiky pro předpověď větvení
- 16 kBb instrukční cache (I cache)
- jednotka správy paměti (MMU), obsahující 64 b instrukční překladový buffer okolí (iTLB) a 64 b vstupní datový překladový buffer okolí (dTLB)



Obr. č. 6.36

Přednahrávací a odesílací jednotka (PDU)

zdvojuje instrukce před tím než jsou žádány tak, že provádějící jednotka nemusí čekat na instrukce. Instrukce mohou být dvojeny ze všech úrovní paměťové hierarchie (instrukční cache, externí cache, a s hlavní pamětí) v pořadí zdvojování přes podmíněné větvení, dynamické větvení předpovídané schéma je implementováno v HW. Výsledek větvení je základní bod na 2 bitové historii větvení. Následující položka asociovaná s každou čtvrtou instrukcí v instrukční cache (i cache) ukazuje do následující I cache řádku pro vytvoření dvojníka. Použití další položky umožňuje následovat branou větve a zajistit téměř stejnou instrukční šířku pásma v době běhu sekvenčního kodu. Zdvojené instrukce jsou uchovávány v instrukčním bufferu, dokud nejsou poslány na konec spojení, může se bufferovat až 12 instrukcí.

Instrukční cache (I cache)

je 16 kB dvoucestná skupina asociovaných cache s 32 B bloky. Cache je fyzicky indexována a obsahuje fyzické tagy (značky). Skupina je předpovězena jako část z následující položky, tedy jen index bitů adresy (13 bity, které se rovnají minimální velikosti stránky) jsou potřebné k adresaci cache. I cache vrací až 4 instrukce z 8 instrukcí široké cachovací řádky.

Celočíselná odbavovací jednotka IEU

obsahuje komponenty:

- dvě aritmetickologické jednotky
- multicyklovou celočíselnou násobičku
- multicyklovou celočíselnou děličku
- 8 registrů oken
- 4 skupiny globalních registrů (normal, alternate, MMU, globální přerušení)
- zachytávací registr (podporované úrovně zachytávání v tab. č. 6.7)

Tabulka č. 6.7

	UltraSPARC I	UltraSPARC II
MAXTL	4	4
zachytávací úroveň	5	5

Jednotka pro čísla s desetinou čárkou FPU

je rozdělena do samostatných realizačních jednotek, které dovolují procesoru ultrasparc zpracování a zakončení dvou floatových instrukcí za cyklus. Vstupní a výstupní data jsou uchovávána v 32-vst. registru archivů (pořadač), kde každý vstup může obsahovat 32 bit hodnotu nebo 64 bit hodnotu. Většina instrukcí je plně propojena (s předáváním za jeden cyklus) a mají latenční dobu tři a neovlivňují přesnost operandů (stejný latenční čas pro jednoduchou i dvojitou přesnost). Dělení a mocniny základní instrukce nejsou propojeny a zaberou 12 / 22 cyklů (jedna / dvě) k odbavení, ale nepřetěžují procesor. Jiné instrukce, následující za dělením / mocninou (hlavní) mohou být ukončeny a odloženy do registru archivu před dokončením dělení / mocniny. Precizní model výjimek je udržován synchronizací floatové fronty s frontou integerovou a s předpovídáním nedostatku pro dlouho čekající operace.

Grafická jednotka GRU

ultrasparc zavádí základní sadu grafických instrukcí, které poskytují rychlou HW podporu pro 2 dimenzionální a 3 dimenzionální obrázky a video procesy, obrazovou kompresi, zvukové zpracování... Předpokladem jsou 16 bit a 32 bit částečné sčítání, boolean a porovnávání. 8 bit a 16 bit částečné násobení je podporováno. Jeden cyklus, délka jeden pixel, data -seřazena, pakována a operace spojování jsou všechny podporovány v GRU.

Jednotka správy paměti MMU

zařizuje mapování mezi 44 bitovou virtuální adresací a 41 bitovou fyzickou adresou. To je ustálený proces přes 64 b iTLB pro instrukce a 64 b dTLB pro data, obě TLB jsou plně asociativní. Ultrasparc zprostředkovává HW podporu pro SW TLB nedostatkovou strategii. Sada samostatných globálních registrů je dostupná pro zpracování MMU odchytů. Jsou podporovány velikosti stránky 8 kB (13 bit doplněk), 64 kB (16 bit doplněk), 512 kB (19 bit doplněk) a 4 MB (22 bit doplněk).

Nahrávací / uchování jednotka LSU

je důležitá pro generování všech nahrávaných a uchovávaných virtuálních adres (včetně atomických a ASI nahrávek) pro přístup D cache, pro zdvojování nahrávání při nedostatku ze spojení přes nahrávací buffer a pro zdvojování uchování v ukládacím bufferu. Jedno nahrání, nebo zápis může být udělán za jeden cyklus.

Datová cache D cache

je zapisující průchodně, tedy bez alokací - 16 kB přímé mapování cache s dvěma 16 byte podbloky na řádek. Tato je virtuálně indexována a fyzicky označena (vipt) značkou je pole s dvěma úrovněmi, tedy aktualizace způsobená naplněním řádku není v rozporu s čtením značky pro přicházející nahrávání. Prohledávání D cache používá druhý označený port, tak nezdržují nahrávání.

Externí cache ECU

hlavní role ECU je dosáhnout vhodné minimalizace nedostatků I cache a D cache. ECU může dosáhnout přístup jednou za cyklus do externí cache (E cache), přístup do E cache jsou propojeny efektivně - E cache je část instrukčního spojení. Programy s rozsáhlými datovými skupinami mohou uchovávat data v E cache a mohou plánovat instrukce s nahráváním utajeně v E cache latenční době. Float kód může použít tento rys k efektivnímu skrytí nedostatku pro D cache.

Tab. č. 6.8 ukazuje na E cache velikost, kterou každý ultrasparc model podporuje. E cache velikost řádky je vždy 64 byte. Ultrasparc používá MOESI (modify, own, exclusive, shared, invalid) protokol udržující spojitost systému.

Tabulka č. 6.8

E cache velikost	UltraSPARC I	UltraSPARC II
512 Kb	Ano	Ano
1 Mb	Ano	Ano
2 Mb	Ano	Ano
4 Mb	Ano	Ano
8 Mb		Ano
16 Mb		Ano

ECU zajišťuje přesahy procesů během nahrávacích a uchovávacích nedostatků. Pro situace ukládání, které zasáhne E cache, může pokračovat, dokud je v procesu neúspěch nahrávání. ECU může zpracovávat čtení a psaní tak, že je nerozlišuje, bez nákladného se vracení (2 cykly). Konečně ECU provádí blokové hledání, nahrávání a uchovávání, které nahrává/ukládá 64 Bytově řádek dat z paměti do float registru souborů. ECU zajišťuje vysoký přenos širokého pásma bez znečištění E cache.

E cache SRAM módy

různé ultrasparc modely podporují různé E cache SRAM konfigurace použitím jedné nebo více sram módů. Tab č. 6.9 ukazuje módy, které ultrasparc podporuje.

Tabulka č. 6.9

SRAM Mód	UltraSparc I	UltraSparc II
1-1-1	Ano	Ano
2-2		Ano

1-1-1 mód propojení

E cache sram mají časový cyklus shodný s procesorem. Jméno 1-1-1 indikuje, že to potřebuje jeden procesorový impuls k poslání adres, jeden k přístupu do SRAM pole a jeden k návratu do E cache datové. 1-1-1 mod má 3 cykly pin po pinu prodlevy a poskytuje nejlepší možné E cache předávání.

2-2 register - zavření mod

E cache SRAM má časový cyklus shodný s 1/2 procesorového cyklu. Jméno 2-2 indikuje že potřebuje dva procesorové cykly k poslání adresy a dva k přístupu a návratu do E cache datové. 2-2 mod má 4 cykly prodlev pin po pinu, které přinášejí menší E cache předávání ve snížené ceně.

Memory interface unit MIU

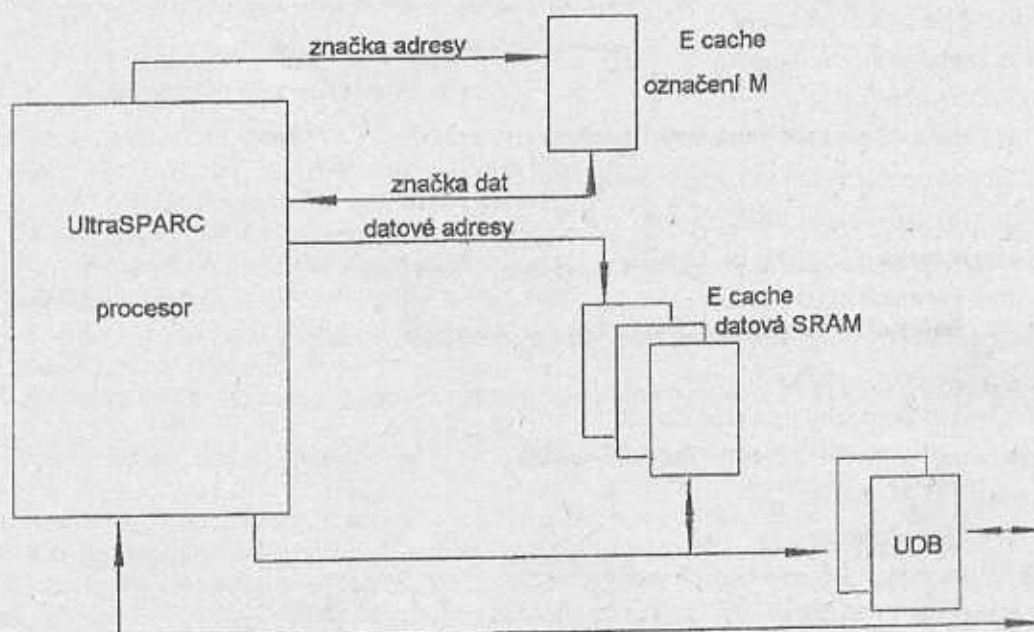
ovládá všechny přenosy do systemového řadiče, např nedostatek externí cache, přerušení, vyhledávání, zpětný zápis... MIU komunikuje se systémem stejným modelem -závislá část ultrasparku frekvence. Tab č. 6.10 ukazuje možné převody mezi procesory a systémovou frekvencí pro každý ultrasparc model.

Tabulka č. 6.10

od frekvence	UltraSparc I	UltraSparc II
2:1	Ano	Ano
3:1		Ano
4:1		

Ultra sparc subsystem

Obr. č. 6.37 je kompletní ultrasparc podsystem, který se skládá z procesoru, synchronní SRAM komponent pro E cache značky a data a dvou čipů ultrasparc datové buffery UDB. UDB oddělují E cache od systému, obstarávají datový buffer pro přicházející a odchozí systémové přenosy a obstarávají ECC generaci a kontrolu.



Obr. č. 6.37

6.7.2 UltraSPARC™ - II

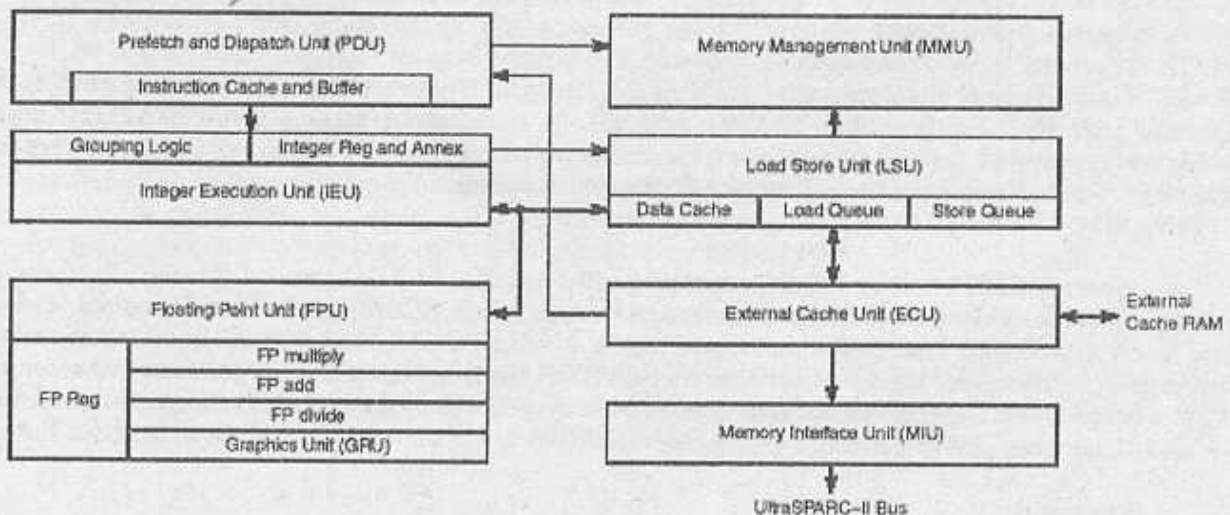
64-bitový mikroprocesor druhé generace architektury SPARC v9. UltraSPARC-II (STP1031) je vysoce výkonný, vysoce integrovaný superskalární procesor postavený na bázi 64-bitové architektury SPARC V9. Tento procesor dokáže dokončit až 4 instrukce za 1 hodinový cyklus (1 tik systémových hodin). UltraSPARC-II jsou procesory druhé generace UltraSPARCů, které používají novou technologii řízení procesů, vyšší hodinové kmitočty, několik režimů SRAM a také několik možných volitelných poměrů frekvencí mezi systémem (systémovou deskou) a procesorem, což jistě nejvíce ocení hlavně vývojáři systémů na bázi tohoto procesoru. Avšak při tom všem tyto procesory stále zůstávají kompatibilní se systémy staršími na bázi procesorů UltraSPARC-I.

Procesor STP1031 taky široce podporuje grafiku jak 2D, tak i 3D, práci s obrazem, kompresi a dekompresi videa a videoefekty. To vše u tohoto procesoru zajišťuje VIS - Visual Instruction Set, což je něco jako MMX u procesorů Intel x86, ale mnohem lepší a výkonnější. VIS (česky - vizuální instrukční sada) poskytuje výkonné funkce hlavně v oblasti multimédií. Je to například komprese a dekomprese videa normy H.261 v reálném čase a rovněž dekomprese jednoduchého proudu videa formátu MPEG-2 v plné kvalitě, opět v reálném čase a bez dalšího potřebného hardwaru.

Vlastnosti

- Vyhovuje architektuře SPARC V9
- Binárně kompatibilní se všemi SPARCOvskými aplikacemi
- Podporuje multimédia pomocí instrukcí VIS
- Podporuje multiprocessing
 - Propojení 4-procesorů
 - Podpora adresářově orientovaného protokolu

- 4-cestný superskalární design s 9-ti prováděcími jednotkami
 - 4 celočíselné jednotky
 - 3 jednotky plovoucí čárky
 - 2 jednotky pro grafiku
- Volitelný formát Bytů Little- nebo Big-Endian
- 64-bitové adresové ukazatele
- 16-kB neblokovaná datová cache
- 16-kB instrukční cache
 - Jednocyklové sledování větvení
 - 2-bitová predikce větvení uvnitř cache
- Integrovaný řadič cache druhé úrovně
- Podporuje cache od 512 kB do 16 MB
- Blokové instrukce Load / Store
- Vysokorychlostní propojení procesoru s pamětí - přenosové rychlosti
 - 1.3 GB/s Procesor/Paměť
 - 600 MB/s Pomocný procesor/Paměť
- Podpora softwarového Prefetch dat do E-Cache
- Zkoumání JTAG hranic
- Technologie / Pouzdro
 - 0.35 μm 5-vrstvá Metal CMOS technologie
 - Napájení 3.3V a 2.6 V
 - 787-pinové keramické pouzdro typu LGA (Land Grid Array)
- Power Management
- Možnost volby taktu procesoru - režimy kompatibilní s UltraSPARC-I



Obr. č. 6.38 Funkční blokový diagram procesoru UltraSPARC-II

Technické parametry

V procesoru UltraSPARC-II jsou na jednom čipu integrovány následující komponenty:

Prefetch, branch prediction and dispatch unit (PDU)

16 kB instrukční cache (I-Cache)

Memory management unit (MMU) obsahující 2 buffery o 64 položkách

- instruction translation lookaside buffer (iTLB)
- data translation lookaside buffer (dTLB)

Integer execution unit (IEU) se dvěma aritmeticko-logickými jednotkami (ALU)

Load & Store unit (LSU) se samostatným adresovým generátorem

Load buffer a Store buffer střídající data z pipeline

16 kB datová cache (D-Cache)

Floating-point unit (FPU) s nezávislými podjednotkami pro sčítání, odčítání, násobení a dělení/odmocňování.

Grafická jednotka (GRU) sestavená ze dvou nezávislých spouštěcích pipeline

Řadič externí cache (E-Cache)

Jednotka pro komunikaci s pamětí - umožňuje přístup do hlavní paměti a k I/O portům

Prefetch and Dispatch Unit

Tato jednotka předřizuje instrukce ještě předtím, než jsou momentálně požadovány v pipeline, takže spouštěcí jednotka nemusí čekat. Instrukce mohou být získávány ze všech možných úrovní paměťové hierarchie, včetně I cache, externí cache a hlavní paměti. V případě požadavku vykonat instrukci, která je až za podmíněným větvením, je hardwarově implementováno schéma tzv. predikce dynamického větvení. Výsledek větvení je založen na 2-bitové historii větvení.

"Následující položka" asociovaná s každou čtvrtou instrukcí v instrukční cache ukazuje na další řádek v instrukční cache, který bude připraven. Použití systému "následující položky" umožňuje následovat větvení takovou rychlostí, jako by byl vykonáván pouze sekvenční kód. Takto zpracované instrukce jsou uloženy do instrukčního bufferu, dokud nejsou poslány na pipeline. Může zde být bufferováno až 12 instrukcí.

Instrukční Cache (I-Cache)

Instrukční cache je 16 kB pseudo-dvoucestná asociativní cache s 32-Bytovými bloky. Je fyzicky indexována a obsahuje fyzické značky. Nastavení je predikováno jako součást tzv. "následující položky", takže k adresování cache jsou potřebné jen indexové bity adresy (13 bitů, které odpovídají minimální velikosti stránky). Instrukční cache vrací až 4 instrukce z řádku o šířce 8 instrukcí.

Memory Management Unit (MMU)

MMU poskytuje mapování mezi 44-bitovou virtuální adresou a 41-bitovou fyzickou adresou. To je prováděno přes 64-položkový lookaside buffer pro překlad instrukcí (iTLB) a 64-položkový lookaside buffer pro překlad dat (dTLB), přičemž oba jsou plně asociativní. UltraSPARC-II poskytuje hardwarovou podporu pro softwarově založenou strategii kontroly chyb překladových lookaside bufferů (TLB). Samostatná sada globálních registrů je přístupná vždy, když dojde ke sporu v práci MMU. Podporované velikosti paměťových stránek jsou 8 kB, 64 kB, 512 kB a 4MB.

Integer Execution Unit (IEU)

Hlavní výpočetní část IEU formují dvě aritmeticko-logické jednotky (ALUs). Dalšími částmi jednotky IEU jsou multicyklová celočíselná násobička a multicyklová celočíselná dělička. Je tu k dispozici také 8 registrových oken a 4 sady globálních registrů (normální, alternativní, registry pro MMU a pro práci s interrupty).

Klapkové registry (trap registers) jsou rovněž součástí IEU.

Load/Store Unit (LSU)

LSU je zodpovědná za generování virtuálních adres pro všechna volání LOAD a STORE, dále za přístup do datové cache, za depárování párovaných chyb vzniklých při LOAD z pipeline přes load buffer a za depárování cyb při STORE při store bufferu. Za jeden cyklus může být provedena 1 akce load nebo 1 akce store.

Data Cache (D-Cache)

Datová cache je typu write-through, nealokující, 16 kB, přímo mapovaná cache se dvěma 16-Bytovými subbloky v každém řádku. Je virtuálně indexována a fyzicky značena. Pole značek je dvouportové, takže update značek nekoliduje se čtením těchto značek.

Floating-Point Unit (FPU)

Oddělení výkonných jednotek v FPU umožňuje procesoru UltraSPARC-II dokončovat 2 instrukce v plovoucí čárce v jednom cyklu. Data parametrů a výsledku jsou uložena do souboru 32-položkových registrů, kde každá položka může obsahovat 32-bitovou nebo 64-bitovou hodnotu. Většina instrukcí je plně pipelineována, trvají 3 cykly, přičemž nejsou ovlivněny přesností operandů (doba vykonávání je stejná jak u jednoduché, tak i dvojitě přesnosti). Instrukce dělení a odmocňování nejsou pipelineovány, takže jejich provedení zabere 12 cyklů (při jednoduché přesnosti) a 22 cyklů (při dvojitě přesnosti), ale nezatěžují procesor. Ostatní instrukce, které následují za dělením nebo odmocňováním, mohou být vykonány a výsledek zapsán do registru ještě dříve, než se dělení nebo odmocňování dokončí. Přesnost je udržována pomocí synchronizace floating-point pipelineu s celočíselným pipelineem a predikcí operací s dlouhou dobou vykonávání.

Graphic Unit (GRU)

UltraSPARC-II uvádí obsáhlou sadu grafických instrukcí, které poskytují rychlou hardwarovou podporu pro práci s dvojrozměrnými i trojrozměrnými obrázky a s videem, kompresí obrazu, práci s audiem a podobné funkce. Jsou zde poskytnuty instrukce pro 16-bitové a 32-bitové (odděleně) sčítání, logické operace a porovnávání, dále je podporováno 8-bitové a 16-bitové násobení. Grafická jednotka ještě obsahuje jednocyklové operace výpočtu vzdáleností pixelů, přízpůsobování rozměrů obrazových dat, operace pakování a spojování atd.

External Cache Unit (ECU)

Hlavní role ECU spočívá v tom, že účinně zachycuje nedostatky I-Cache a D-Cache. ECU může zpracovat jeden přístup do externí cache za cyklus. ECU také poskytuje podporu pro práci externí cache v režimech 1-1-1 a 2-2.

V režimu 1-1-1 je SRAM taktována na frekvenci procesoru.

V režimu 2-2 je SRAM taktována na polovinu frekvence procesoru.

Velikost externí cache může být 512 kB, 1 MB, 2 MB, 4 MB, 8 MB a 16 MB, přičemž velikost řádku je vždy 64 Bytů. K udržení spojitosti v systému se používá protokol MOESI (modified, own, exclusive, shared, invalid). Externí cache je fyzicky indexována a fyzicky značena.

Memory Interface Unit (MIU)

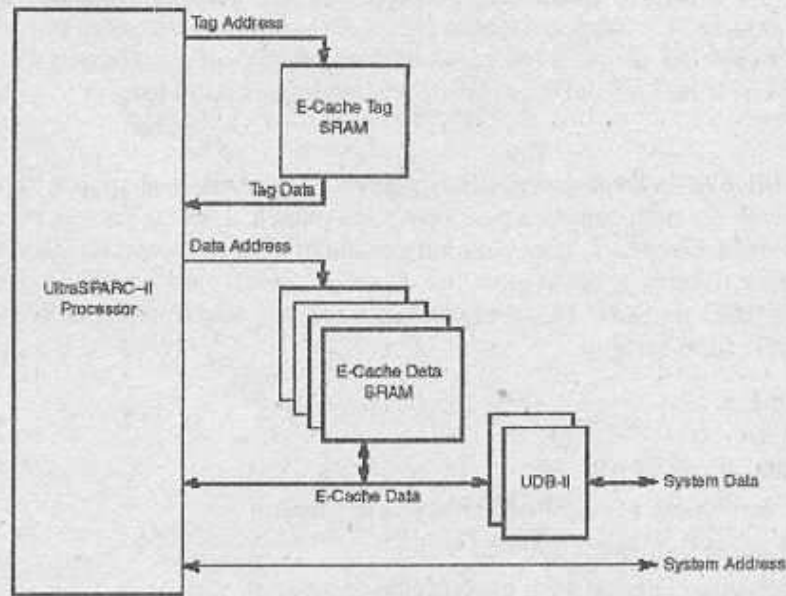
MIU zachycuje všechny transakce v systému, jako chyby externí cache, přerušení, snoops, writebacks, atd. MIU komunikuje se systémem na frekvenci nižší, než je frekvence procesoru, přičemž poměr může být 1/2, 1/3 nebo 1/4.

Subsystémy UltraSPARC - II

V následujícím odstavci bude slovo "systém" znamenat nějakou souvislou lokaci UltraSPARC-II (zde může být zahrnuto několik "počítačů", procesorů zapojených na interconnect, každý včetně paměti cache).

Pojmem "subsystém" rozumíme 1 procesor a elementy jeho rozhraní.

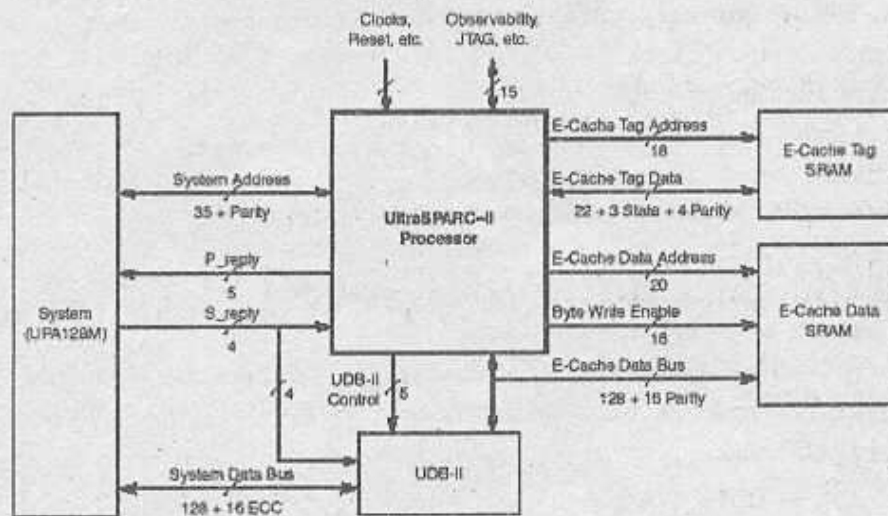
Kompletní subsystém UltraSPARC-II se skládá s procesoru UltraSPARC-II, synchronních SRAM komponent pro externí cache (tags & data) a UltraSPARC-II data bufferu, který má dvě identická zařízení UDB-II. UDB-II izoluje E-Cache od systému a poskytuje datové buffery pro přicházející a odcházející systémové transakce. UDB-II také poskytuje generování a kontrolu ECC.



Obr. č. 6.39

Přehled rozhraní UltraSPARC-II

Zde budou popsány některé hlavní rozhraní UltraSPARC-II. Typický modul obsahuje externí cache složenou ze značkovací (tag) jednotky a datové jednotky. Obě jednotky mohou být implementovány pomocí paměti typu SRAM. Pro přístup do obou jednotek (jak značkovací, tak i datové) jsou poskytnuty oddělené adresové a datové sběrnice, čímž by se měla zvýšit rychlost. Hlavní úloha UDB-II je oddělení procesoru UltraSPARC-II a jeho externí cache od hlavní systémové sběrnice, takže cache může být taktována na stejné frekvenci jako procesor. Datový buffer zároveň poskytuje přeskočení mezi systémovými transakcemi a lokálními transakcemi v E-Cache kdykoli externí cache potřebuje část datového bufferu. Logika řízení UDB-II je obsažena přímo v technologii UltraSPARC-II aby poskytovala rychlé přenosy dat z/do procesoru UltraSPARC-II nebo z/do externí cache a systému. Oddělené adresové sběrnice a řídicí signály jsou určeny pro podporu systémových transakcí. Součástí rozhraní UltraSPARC-II jsou také hodinové signály, resetovací piny, stavové piny a podpora JTAGu.



Obr. č. 6.40

6.7.3 Procesor UltraSPARC™ - III

Vysoce integrovaný 64-bitový procesor RISC s rozhraním PCI. UltraSPARC-III (SME1040) je vysoce integrovaný 64-bitový superskalární procesor architektury SPARC V9. Volitelně obsahuje také APB™ (Advanced PCI Bridge -- SME2411) pro zlepšení připojitelnosti a podporující komunikaci se vstupně/výstupním kanálem PCI. Rozhraní tohoto procesoru bylo optimalizováno pro požadavky typického uniprocessorového systému. To znamená řešení vyváženosti mezi cenou a výkonností, přinášející výkon a možnosti, které velká většina aplikací nejvyšší třídy potřebuje, optimalizující využití výkonu a snadné použití.

UltraSPARC-III a APB dovolují vývojářům přidávat UltraSPARCové, vysoce výkonné a přitom levné, stupně k širokému použití do vnořovaných a desktopových systémů. Tato kombinace procesoru a PCI Bridge nevytěžuje pouze vnořované systémy, které požadují maximum výkonu procesoru, jako jsou telekomunikační zařízení a nejvyšší třídy tiskáren a fotokopírek, ale UltraSPARC-III také urychluje desktopové aplikace a podporuje připojení velkého množství PCI-kompatibilních zařízení taktované na frekvencích až do 66 MHz (přímo) nebo do 33 MHz (přes bridge).

Vlastnosti

- Vyhovuje architektuře SPARC V9
- Binárně kompatibilní se všemi SPARCovskými aplikacemi
- Podporuje multimédia pomocí instrukcí VIS
- 4-cestný superskalární design s 9-ti prováděcími jednotkami
 - 4 celočíselné jednotky
 - 3 jednotky plovoucí čárky
 - 2 jednotky pro grafiku
- Volitelný formát Bytů Little- nebo Big-Endian
- 64-bitové adresové ukazatele
- 16-kB neblokovaná datová cache
- 16-kB instrukční cache
 - Jednocyklové sledování větvení
 - 2-bitová predikce větvení uvnitř cache
- Integrovaný řadič cache druhé úrovně
- Podporuje cache od 256 kB do 2 MB
- Integrovaný podsystém řízení EDO DRAM paměti s propustností až 400 MB/sec
- 64-Bytové blokové instrukce load a store dat do E-Cache
- Podpora softwarového prefetch
- Podporuje až 3 selhání v E-Cache
- Podporuje0 rozhraní UPA64S
 - 800 MB/Sec
 - 64-Bitové slave rozhraní pro grafický nebo podobný podsystém
- Integrované PCI Rev 2.1
 - Mocnější PIO a DMA PCI I/O kanál než konkurenční řešení
 - Read prefetch, hromadný zápis a posílání
- PCI DMA je soudržný s cache
 - Vyhrazená TLB poskytuje mapování a ochranu
- Zkoumání JTAG hranic

- Technologie / Pouzdro
 - 0.35 μm 5-vrstvá Metal CMOS technologie
 - Napájení 2.6 V & 3.3 V (IO Only)
- Volitelná frekvence procesoru (266 MHz a 300 MHz)

UltraSPARC-III (s 512 kB cache) dosahuje v benchmarkcích SPECint95 hodnoty 10 a v SPECfp95 hodnoty 12. Rovněž také disponuje špičkovou propustností a malým zpožděním. UltraSPARCIII na čipu integruje také řadič paměti a I/O rozhraní.

Tento systém se vyrovnává se zúžením mezi pamětí a CPU použitím následujících rychlostí rozhraní:

Hardwarové Rozhraní

CPU

SRAM externí cache (E-Cache)

Datová propustnost v DRAM

PCI

266 MHz CPU

266 MHz

133 MHz

67 MHz

až do 66 MHz

300 MHz CPU

300 MHz

150 MHz

75 MHz

až do 66 MHz

Toto řešení podporuje velké množství aplikací. Každá aplikační oblast procesoru UltraSPARC-III podporuje decentralizované řízení, dovoluje překrývání mnoha činností. Podporuje vykonávání až 4 instrukcí v jednom cyklu (dokonce i při výskytu podmíněného větvení a selhání v cache) a to díky rozdělení prefetch a odbavovací jednotky. Čtecí buffery na vstupní straně prováděcí jednotky, spolu se zápisnými buffery na straně výstupní, rozdělují pipelineové provádění ze selhání datové cache. Předpovídané instrukce jsou vydávány k vykonávání několika funkčním jednotkám v pořadí uvedeném v programu. Takovéto instrukce jsou prováděny paralelně a mohou být dokončeny mimo pořadí. Oddělené řízení paměti a jednotky PCI I/O rozhraní také odděluje jejich příbuzné klíčové činnosti od pipeline instrukcí.

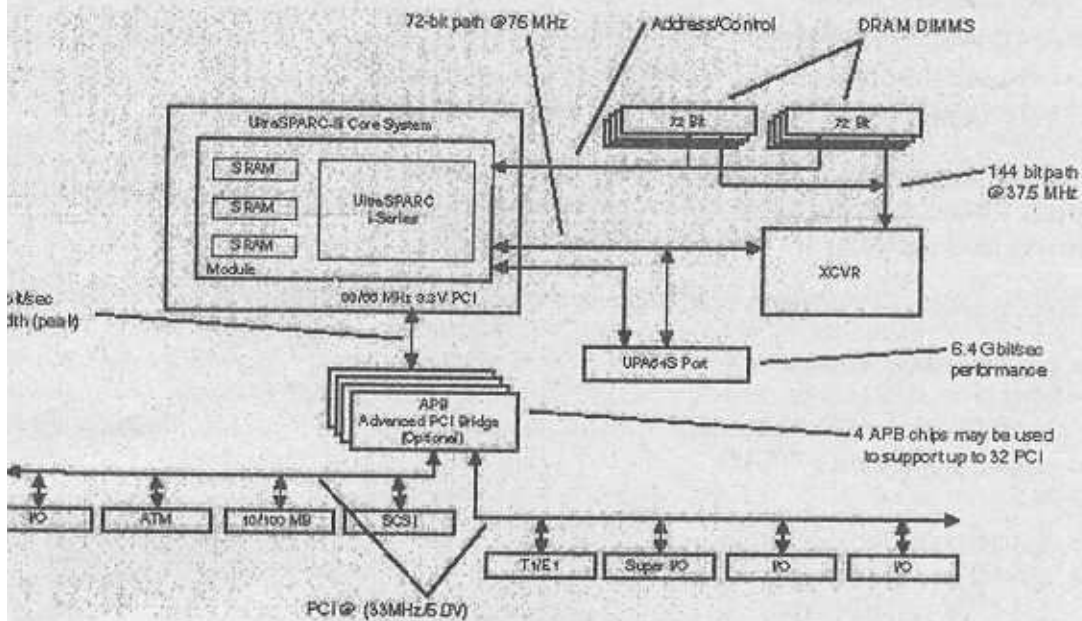
UltraSPARC-III podporuje také množinu rozšířených instrukcí VIS architektury V9. Použití VIS ve spojení s čtyřcestným superskalárním pipeliningem v procesorech série "i" zvyšuje výkon širokého rozsahu síťových aplikací, včetně TCP/IP. Použití instrukcí VIS umožňuje také zrychlení I/O operací. Navíc, instrukce VIS podporují:

- real-timeovou H.261 video kompresi/dekompresi
- dekompresi jednoho proudu MPEG-2 v plné kvalitě bez nutnosti dalšího hardwaru.

Procesor UltraSPARC-III podporuje populární OS Solaris od SUNu a je binárně kompatibilní s širokým spektrem existujících aplikací. Navíc firma SUN plánuje podporu robustního real-timeového operačního systému třetí strany (RTOS), včetně podpůrných vývojových nástrojů pro vestavěné aplikace.

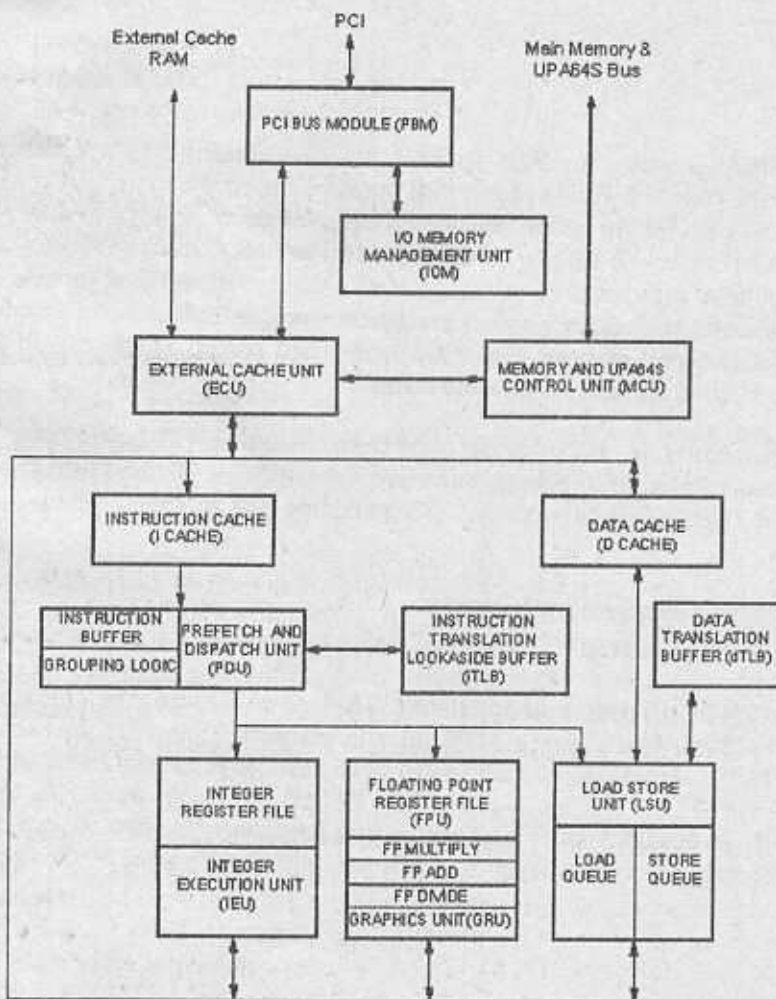
UltraSPARC-III je dostupný buď jako samostatná komponenta, nebo na integrovaných modulech společně s pamětí cache nebo na systémových deskách standardního formátu používaného v průmyslových řešeních.

Příklad systému



Obr. č. 6.41 Příklad implementace systému s UltraSPARC-III

Funkční diagram



Obr.č. 6.42 Funkční blokový diagram procesoru UltraSPARC-III

Přehled komponent

V jednočipovém provedení, UltraSPARC-III integruje následující komponenty:

- Nezávisle časovaná (132 MHz interně, 66 nebo 33 MHz externě) PCI rozhraní, úplně oddělená od hlavního CPU
- Modul sběrnice PCI (PBM)
- Jednotku správy paměti pro PCI I/O (IOM) se 16 vstupy přicházejících I/O pro fyzické mapování/ochranu
- Jednotku řadiče Externí cache (E-Cache) (ECU)
- Jednotku řadiče paměti (MCU), ovládající subsystémy DRAM 144-bitové šířky a rozhraní UPA64S
- 16 kB instrukční cache (I-Cache)
- 16 kB datovou cache (D-Cache)
- Jednotky Prefetch, predikce větvení a rozdělování(dispatch) (PDU) obsahující seskupení logiky a bufferu instrukcí
- 64-vstupý lookaside buffer pro překlad instrukcí (iTLB) a 64-vstupý lookaside buffer pro překlad dat (dTLB)
- Jednotku celočíselných operací (IEU) se dvěma aritmeticko-logickými jednotkami (ALUs)
- Jednotku operací v plovoucí čárce (FPU) s nezávislými částmi pro sčítání, násobení a dělení/mocninu
- Grafickou jednotku (GRU) složenou ze dvou nezávislých vykonávacích pipelines
- Jednotku čtečního a zápisného bufferu (LSU), oddělující přístup k datům od pipeline

PCI Bus Module (PBM) - modul PCI sběrnice

Rozhraní PBM přímo u 32-bit PCI sběrnice, splňuje specifikaci PCI 2.1. Sběrnice PCI běží až do 66 MHz, typicky 33 MHz a 66 MHz. PBM je optimalizován pro 16-, 32- a 64-Bytové přenosy, a může podporovat až 4 mastery. Modul také řadí do fronty nevyřízená přerušení obdržena od čipu soustředujícím přerušení (RIC-SME2210). Celý adresovatelný prostor PCI nelze cacheovat pro odkazy CPU, ale je podporováno souvislé DMA. (To znamená, že všechny zápisy a čtení z/do paměti na PCI jsou v cache souvislé.) Zachytávání přerušení je synchronizováno s dokončením všech prioritních zápisů přes DMA.

I/O Memory Management Unit (IOM)

IOM vykonává překlad adresy z 32-bitových DVMA do 34-bitových fyzických adres, když jsou určeny pro UltraSPARC-III (když je požadován přístup DVMA čtení/zápisu). IOM používá plně asociativní 16-vstupý TLB (lookaside buffer pro překlad). V případě chyby v TLB, IOM vykonává jednoúrovňový hardwarový přechod do velkého zápisného bufferu překladu (TSB) v paměti.

External Cache Control Unit (ECU) - jednotka řadiče externí cache

Hlavní úlohou ECU je účinně zachytávat chyby v I-Cache a D-Cache. ECU může zachytávat v každém cyklu jeden přístup do vnější cache. Čtení dat, která chybí v D-Cache způsobí naplnění 16-Bytové D-Cache použitím 2 následných 8-Bytových přístupů do E-Cache. Zápisy jdou přes E-Cache a jsou plně pipelineovány. Předpříprava instrukce, která chybí v I-Cache, způsobí naplnění 32-Bytové I-Cache použitím 4 následných 8-Bytových přístupů do E-Cache. E-Cache je paritně chráněna. Navíc, ECU podporuje přístup DMA, který přistupuje do externí cache a udržuje souvislost dat mezi externí cache a hlavní paměti.

Externí cache může mít velikost 256 kB, 512 kB, 1 MB, nebo 2 MB (kde velikost řádku je vždy 64 Bytů). Řádky cache mají pouze 3 stavy: změněn, rezervován nebo neplatný.

Kombinace čtečního bufferu a ECU je plně pipelineována. Pro programy s velkým množstvím dat jsou instrukce rozvrženy s ohledem na zpoždění při čtení založeném na zpoždění E-Cache, E-Cache tedy slouží jako velká primární cache. Aplikace nad plovoucí čárkou používají tuto techniku k účinnému "zakrývání" chyb v D-Cache. Souvislost je udržována mezi všemi cachemi a externími odkazy PCI DMA. ECU překrývá zpracování

během chyb při čtení a zápisu. Zápis do E-Cache může být prováděn, zatímco se zpracovává chyba čtení z cache. ECU je také schopna provádět čtení a zápis na obousměrnou datovou sběrnici E-Cache bez zbytečné ztráty času. Blokové čtení a zápis (tj. čtení nebo zápis 64-Bytových řádků dat z paměti nebo E-Cache do registru plovoucí čárky) poskytuje velkou šířku pásma pro přenos. Tím, že se chybějící data nezapišou do E-Cache, je zabráněno tomu, aby se cache zanášela daty, ke kterým je přístupováno pouze jednou. ECU také poskytuje podporu vícenásobných požadavků na přenos vnějších dat do MCU a PBM.

Memory Controller Unit (MCU) - jednotka řadiče paměti

MCU ovládá všechny transakce v DRAM a UPA64S.

Vnější kontakty (piny), řízené MCU, pracují na dělencích taktu procesoru :

- UPA64S běží na 1/3 taktu procesoru
- přenosy dat z/do DRAM se typicky provádí na 1/4 taktu procesoru (nastavení je programovatelné)

Externí zařízení pro přenos dat dovolují, aby data z DRAM měla dvojnásobnou šířku než data z procesoru (piny MEMDATA), tedy cyklus EDO CAS je pouze 26.5 ns při frekvenci procesoru 300 MHz. MCU podporuje také skládanou specifikaci DRAM, která je souhrnem specifikací 60 ns EDO DRAM od hlavních světových výrobců.

Použití rychlejších DRAM dovoluje dosahovat výkonů vyšších než uvedených, protože různé části paměti mají programovatelné zpoždění.

Instruction Cache (I-Cache) - instrukční cache

I-Cache je 16 kB dvoucestná množinově-asociativní cache s 32-Bytovými bloky. Je fyzicky indexována a fyzicky značena. Množina je předvídána jako součást tzv. "následující položky", takže k adresování cache jsou potřebné jen indexové bity adresy (to znamená pouze 13 bitů, které odpovídají minimální velikosti stránky). Instrukční cache vrací až 4 instrukce z řádku o šířce 8 instrukcí.

Data Cache (D-Cache) - datová cache

Datová cache je typu write-through, nealokující, 16 kB přímo mapovaná cache se dvěma 16-byte podbloky na každém řádku. Je virtuálně indexována a fyzicky značena. Pole značek je dvouportové, takže update značek nebrání současnému čtení těchto značek.

Prefetch and Dispatch Unit (PDU)

PDU připravuje instrukce předtím, než se potřebují v pipeline, takže prováděcí jednotka nemusí čekat na instrukce. Instrukce mohou být předpřipraveny ze všech úrovní paměťové hierarchie, včetně instrukční cache, externí cache a hlavní paměti. K předpřipravě podmíněného větvení je hardwarově implementováno schéma dynamické predikce větvení, založené na dvoubitové historii větvení. "Následující položka" asociována s každou čtvrtou instrukcí v I-Cache ukazuje na následující řádek v I-Cache, který bude připraven. To dovoluje následovat dané větvení a poskytuje stejnou rychlost, jako by byl prováděn sekvenční kód. V instrukčním bufferu je uloženo až 12 předpřipravených instrukcí vysílaných postupně do pipeline.

Translation Lookaside Buffers (iTLB and dTLB)

Lookaside buffery překladu poskytují mapování mezi 44-bitovými virtuálními adresami a 34-bitovými fyzickými adresami. 64-položkový iTLB je používán pro instrukce a druhý 64-položkový dTLB pro data, přičemž oba jsou plně asociativní. UltraSPARC-III poskytuje hardwarovou podporu pro softwarovou strategii chyb TLB. Pro malou prodlevu při zachycení chyby je dostupná samostatná sada globálních registrů, kdykoliv je nějaká chyba objevena. Jsou podporovány paměťové stránky o velikostech 8 kB, 64 kB, 512 kB a 4 MB.

Integer Execution Unit (IEU) - výkonná jednotka pro celočíselné operace

Její hlavní výpočetní část tvoří 2 aritmeticko-logické jednotky (ALUs).

IEU také obsahuje vícecyklovou celočíselnou násobičku s detekcí předčasného ukončení a vícecyklovou celočíselnou děličku. Zároveň je zde dostupno také osm registrových oken a čtyři sady globálních registrů (normální, alternativní, MMU a pro přerušení). Součástí IEU jsou také záchytné registry (UltraSPARC-II podporuje pět úrovní zachytávání).

Floating-Point Unit (FPU) - jednotka pro zpracování operací v plovoucí řádové čárce

Oddělení výkonných jednotek v FPU umožňuje procesoru UltraSPARC-II dokončovat 2 instrukce v plovoucí čárce v jednom cyklu. Data parametrů a výsledku jsou uložena do souboru 32-položkových registrů, kde každá položka může obsahovat 32-bitovou nebo 64-bitovou hodnotu. Většina instrukcí je plně pipelinována, trvají 3 cykly, přičemž nejsou ovlivněny přesností operandů (doba vykonávání je stejná jak u jednoduché, tak i dvojitě přesnosti). Instrukce dělení a odmocňování nejsou pipelinovány, takže jejich provedení zabere 12 cyklů (při jednoduché přesnosti) a 22 cyklů (při dvojitě přesnosti), ale nezatěžují procesor. Ostatní instrukce, které následují za dělením nebo odmocňováním, mohou být vykonány a výsledek zapsán do registru ještě dříve, než se dělení nebo odmocňování dokončí. Přesnost je udržována pomocí synchronizace floating-point pipelineu s celočíselným pipelineem a predikcí operací s dlouhou dobou vykonávání.

Graphic Unit (GRU) - grafická jednotka

UltraSPARC-II uvádí obsáhlou sadu grafických instrukcí, které poskytují rychlou hardwarovou podporu pro práci s dvojrozměrnými i trojrozměrnými obrázky a s videem, kompresi obrazu, práci s audiem a podobné funkce. Jsou zde poskytnuty instrukce pro 16-bitové a 32-bitové (odděleně) sčítání, logické operace a porovnávání, dále je podporováno 8-bitové a 16-bitové násobení. Grafická jednotka ještě obsahuje jednocyklové operace výpočtu vzdálenosti pixelů, přizpůsobování rozměrů obrazových dat, operace pakování a spojování atd.

6.7.4 Procesor SuperSPARC™ II

SuperSPARC™II je vysoce integrovaný 32-Bitový RISC mikroprocesor označovaný taky jako STP1021, který patří do rodiny procesorů SuperSPARC II. Obdobně jako jeho předchůdci (STP1020N a STP1020A) i tento následník plně patří ke SPARC verze 8 a je plně zpětně kompatibilní s dřívější verzí implementace SPARC 7, na které běží více než 8500 aplikací a vývojových nástrojů.

Tento vysoce výkonný superskalární mikroprocesor je konstruován BiCMOS technologií. Díky této vysoké integraci se celý procesor skládá s několika subsystémů implementovaných na jednom čipu.

- Celočíselná jednotka
- Jednotka výpočtů v pohyblivé řádové čárce
- Jednotka správy paměti (memory management unit MMU)
- L1 instrukční a datovou CACHE (celkem 56 kB)
- Rozhraní pro podporu sběrnice MBus a VBus

Tato vysoká integrace zvyšuje celkové schopnosti systému při snížení počtu dalších zařízení.

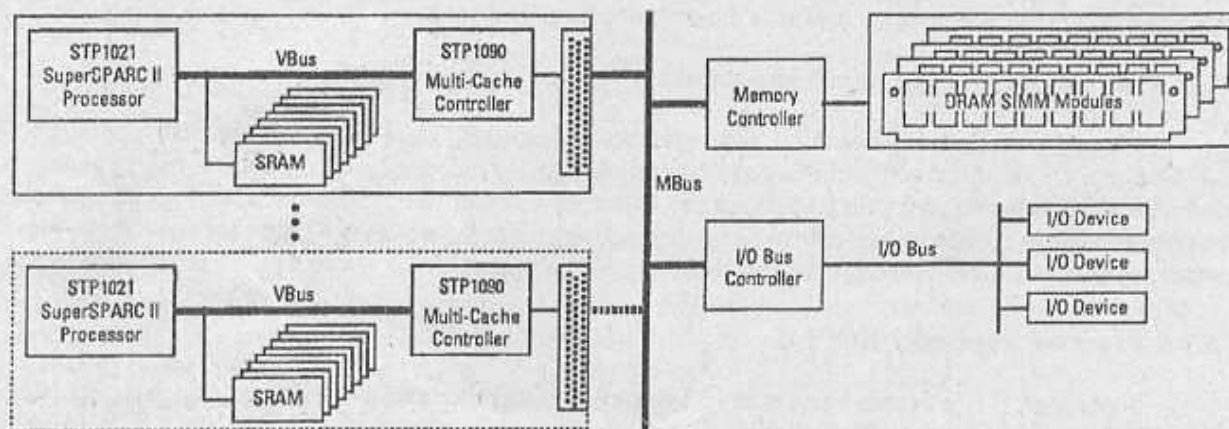
V tomto procesoru můžeme najít spoustu moderních vlastností:

- Superskalární jádro pracující na 75 MHz
- Velké okno registrů (8 oken na 136 registrů)
- Integrovanou jednotku správy paměti
- Velmi výkonnou jednotku pro výpočty v pohyblivé řádové čárce
- Velkou L1 instrukční a datovou CACHE rozdělenou na 20 kB instrukční a na 16 kB datové
- Velké buffery

- Hardwarové násobení a dělení
- Podporu souvislé CACHE pro mutli-processing
- Zabudovanou logiku pro vlastní otestování
- Úplné JTAG rozhraní IEEE1149.1
- Kontrola rozhraní (Boundary scan)
- Rozšířená kontrola rozhraní (Enhanced boundary scan)
- Stop mód
- Kontrola založená na 2 taktovém zpoždovacím testu
- Monitoring výkonu

Typické aplikace SuperSPARC™II

S tímto procesorem se počítá s využitím v mnoha aplikacích jak jednoprocessorových tak i ve velkých multiprocessorových systémech. Jednoprocessorové i multiprocessorové systémy mohou být postaveny se SuperSPARCEm ve VBus módu s použitím externího cache řadiče (MXCC STP1090). Tento externí řadič podporuje multiprocessorové konfigurace použitím MBus, nebo XBus rozhraní s použitím cache do 2 MB (viz obr. č. 6.43)



Obr. č. 6.43 Typický jednoprocessorový / multiprocessorový systém s externím řadičem cache

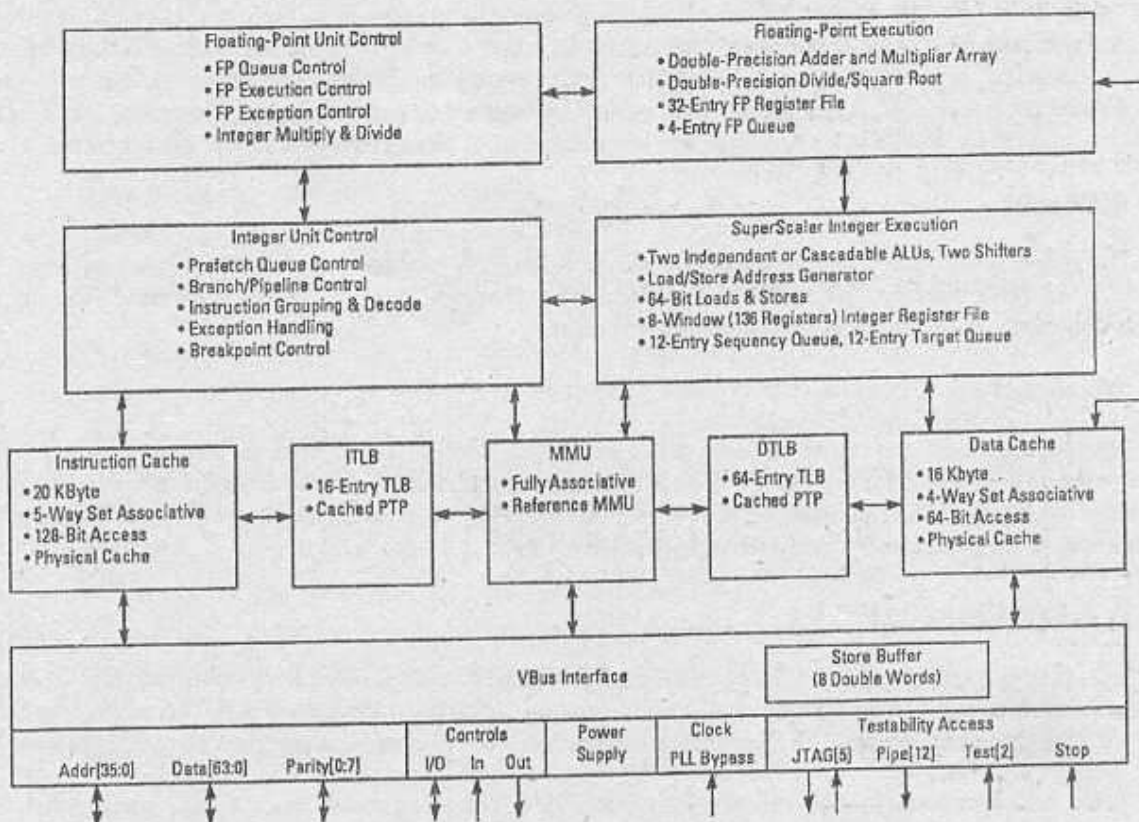
Mikroarchitektura procesoru

Obr. č. 6.44 ukazuje pohled na mikroarchitekturu procesoru. Superskalární celočíselná jednotka provádí seskupení instrukcí a jejich následné dekodování a vykonávání aritmetických, posuvných, větvících a load/store instrukcí. V celočíselné jednotce jsou 3 aritmeticko logické jednotky, které jsou dynamicky konfigurovány jako dvě nezávislé nebo kaskádní aritmeticko logické jednotky, což závisí na instrukčním streamu.

Jednotka výpočtů v pohyblivé řádové čárce se skládá z indexového souboru pohyblivých řádových čárek, polí sčítaček/násobiček s dvojitou přesností a kontrolní logiky. Tato jednotka také vykonává celočíselné násobení a dělení. Jednotka správy paměti vykonává převod adres z virtuální do fyzické podoby. Skládá se z 64 záznamů plně asociativní TLB s hardwarovou pohyblivou tabulkou pro TLB neúspěšné procesy.

SuperSPARC II má oddělenou instrukční a datovou cache, která umožňuje rychlý přístup k instrukcím a datům. Instrukční cache o velikosti 20 kB je 5ti cestná asociativní paměť, která dovoluje vybrat 4 instrukce při každém přístupu, a to při použití 128-bitové sběrnice. Datová cache o velikosti 16 kB je uspořádaná 4 cestná asociativní paměť.

Kromě těchto vlastností je na čipu integrovaná BIST(Built-In-Self-Test) logika, JTAG interface a spousta vlastností, které podporují systémový a softwarový debugging a hardwarové break-pointy.



Obr. č. 6.44 lokový funkční diagram procesoru SuperSPARC™II

Operační módy

VBus je nepřepínaná synchronní sběrnice, která je speciálně navržena pro zajištění velmi výkonného propojení mezi procesorem, externí cache řadičem (STP1090) a externí cache pamětí tvořenou synchronní SRAM pamětí. Skládá se z 36-bitové adresní sběrnice a 64-bitové datové sběrnice. Všechny přenosy prováděné na sběrnici jsou synchronizovány s hodinovým taktmem procesoru. Řídící logika sběrnice VBus je integrována na cache řadiči STP1090, který také podporuje až 2 MB cache a multiprocessing.

MBus je SPARCovská mezinárodně standardizovaná sběrnice navržena jako sběrnice nezávislá na procesoru mezi jedním, nebo více procesory a pamětí. Je to 64-bitová vysoce výkonná přepínaná sběrnice, která je plně synchronní se všemi přenosy a je taktovaná zvláštním hodinovým signálem. Tato sběrnice podporuje blokový přenos o velikostech až 128 B s maximálním výkonem 520 MB/s. Všechny přenosy na této sběrnici jsou řízeny externí logikou integrovanou na cache řadiči STP1090. Řídící algoritmus není definován MBus specifikací, aby dovozoval větší flexibilitu v návrzích různých systémů. Sběrnice je definována jak pro jednoprocetorové systémy nazývaná Level1, tak i pro multiprocetorové systémy nazývaná Level2.

Operace Reset

Procesor SuperSPARC™II podporuje tři druhy resetu

- Hardware reset
- BIST reset
- Watchdog reset

Hardware reset

Hardware reset je vyvolán externě a to signálem RESET. Po tomto signálu následuje v procesoru několik akcí. Nejprve se v několika tisíci cyklech inicializuje řídicí logika. Během této doby jsou také

zkonfigurovány řídící obvody cache. Během této doby procesor nereaguje na žádné požadavky ze sběrnice. Jakmile jsou tyto akce ukončeny, je ihned vyvolán reset trap, což způsobí, že se procesor dostane do boot módu a začne vykonávat kód na virtuální adrese 0xFF000000. Tato operace přinutí sběrnici k jednomu načtení z fyzické adresy 0xFF000000, kde horních 8 bitů je nastaveno výsledkem boot módu a zbytek jsou dvě instrukce. Toto čtení musí pokračovat do doby, kdy je vykonána instrukce kontrolující tento přenos (obvykle asi tak 2-3 instrukce).

BIST reset

Druhý typ resetu, který generuje BIST logika je skoro identický jako předchozí hardwarový reset. S tím rozdílem, že BIST operace mohou být požadovány jak softwarem, tak přes JTAG interfeace. Jakmile je tento reset dokončen, je automaticky generován interní reset.

Watchdog reset

Oproti hardwarovému resetu je tento reset generován interně. Tento reset je generován v případě přechodu procesoru do error módu, což umožňuje zotavení z mnoha chyb. Tímto resetem je dotčen jediný kontrolní bit z MMU (jednotka řízení paměti) a to boot mode bit(BT). Error mode (EM) bit je nastaven k indikaci, zda došlo k watchdog resetu, nebo hardwarovému resetu.

6.7.5 Procesor UltraSPARC-III

Třetí generace rodiny mikroprocesorů UltraSPARC. Tento procesor je projektován pro práci na 600 MHz. Pro splnění velkých požadavků na vzrůstající výkon Internetových serverů bude UltraSPARC-III nabízet vysoce škálovatelné a robustní řešení systémů, které umožňuje multiprocesorovým systémům pracovat až s 1000 procesory.

Tento mikroprocesor reprezentuje krok kupředu ve výkonu procesorů Sparc a oslovuje rapidně se měnící požadavky v síťových prostředích. Škálovatelnost procesoru dovoluje jednoduše podporovat rychle se rozvíjející infrastruktury při zachování 100% binární kompatibility pro OS Solaris.

Použitím výkoné Sunovské VIS instrukční sady pro síť, média, vizualizaci a Javu dostaneme 2 až 3 násobný výkonu oproti UltraSPARC II.

Nejčastější použití nového procesoru je v aplikacích jako je CAD, finanční analýza a datové sklady. První vzorky tohoto procesoru jsou očekávány během léta 1998.

6.7.6 Procesor MicroSPARC™ -IIep

MicroSPARC-IIep 32-bitový microprocessor je vysoce výkonný microprocesor s nejvyšší integrací. Realizovaný SPARC architekturou verze Specification 8, to je ideálně vyhovující pro nenáročné jednoprocso-rové vkládané aplikace.

Je postaven na nejvyšší úrovni CMOS technologie, s jádrem operujícím na nízkém napětí 3.3 V.

MicroSPARC-IIep na čipu obsahuje: integer unit - jednotku pracující s celými čísly (IU), floating-point unit - jednotku pracující s čísly reálnými (FPU), rozsáhlou oddělenou instrukční a datovou cache (vyrovnávací paměť), paměťová řídicí jednotka (MMU) verze Specification 8, programovatelný DRAM ovladač, ovladač PCI, PCI rozhraní, 16-vstupní IOMMU, podporu flash memory rozhraní, ovladač přerušení, 2 hodiny, vnitřní a skokový sken do JTAG ovladače, podporu power managementu a generátoru hodinových impulsů.

Pracovní frekvence je 100 MHz.

Hlavní rysy

- Integrovaný 32-bitový, 33 MHz PCI expansion bus controller
- Integrovaný 256 MB DRAM controller
- Zabudovaný 16 MB flash memory controller

- SPARC velmi výkonná RISC architektura
- 8-oken, 136-slovní registrový soubor
- 16 kB instrukční cache a 8 kB data cache
- Zabudována jednotka pracující s reálnými čísly
- Na čipu MMU
- Operační napětí 3,3V s 5V kompatibilním I/O
- Integrovaný obvod power managementu

Výhody

- Připojitelný do standardních i/o rozšiřujících zařízení
- Rozhraní flash memory spouští real-time operační systémy, které nahrávají a spouštějí kód mimo ROM
- Kompatibilní s více než 10000 aplikacemi a existujícími vývojovými nástroji
- Manipulující přes PCI se zařízeními pro DOS, stejně jako s UNIXovými aplikacemi
- Rychlé odpovědi přerušeni, volání procedur a spouštění programů
- Rozděluje procesorové operace z pomalé externí paměti
- Podporuje konkurenční instrukce s reálnými a celými čísly
- Připraven pro komplikované operační systémy s chráněnou pamětí a virtuální adresací
- Nízké napětí jádra redukuje spotřebu energie
- Minimální spotřeba ve stavu standby
- Vhodný pro vývojové testy.

TECHNICKÉ PARAMETRY

Integer Unit - Celočíslná jednotka (IU)

MicroSPARC-IIep integer unit obsahuje SPARC celočíselné instrukce definované ve SPARC Architecture Manual verze 8. Celočíslná jednotka obsahuje 136 registrů podporující 8-okenních registrů a 8-globálních registrů. Má velký počet velmi výkonných možností obsahujících instruction prefetching, branch folding a 5-stage instrukčních pipeline. Celočíslná jednotka podporuje little- and big-endian byte ordering.

Floating-Point Unit - Jednotka pracující s čísly reálnými (FPU)

Floating-point unit obsahuje všechny single- a double-precision (přesné) floating-point instrukce definované v SPARC Architecture Manual verze 8. FPU na čtyřnásobné přesnosti zachycuje instrukce a tyto potom přesunuje softwaru. FPU obsahuje jádro na Meiko designu, rychlý několikanásobný, 3-úrovňový instrukční dotazovací, a 32 32-bitových reálnočíslných registrů. Reálnočíslné jádro a rychlé, několikanásobné paralelní zpracování v reálnočíslném násobení (FPMUL) a další floating-point instrukce jsou vykonávány do té doby, než je vyčerpána celá dotazovací konkurenční podpora mezi reálnočíslnými a celočíselnými instrukcemi.

Memory Management Unit - Paměťová řídicí jednotka (MMU)

MicroSPARC-IIep memory management unit přenáší 32-bitové virtuální adresy do 32-bitových fyzických adres. To mapuje fyzické adresy do osmi různých adresových míst. MMU poskytuje podporu specifikovanou ve SPARC Reference MMU verze 8 a implementuje hardware table-walk. To implementuje 32-vstupový plně asociativní překladový oboustranný buffer - translation lookaside buffer (TLB) a poskytuje paměťovou ochranu pro 256 souvislostí.

Instruction Cache - Instrukční cache

Instrukční vyrovnávací paměť cache je 16 kB s řízeným mapováním, virtuálním indexováním, virtuálně značená vyrovnávací paměť. Instrukční cache je organizována jako 512 řad skládajících se z 32 Bytů a 32 bitů.

K redukci chyb při čtení instrukční vyrovnávací paměť podporuje vyrovnávací znovunaplnění ve dvou 32 bitových slovech, proudy a odbočení.

Data Cache - Datová cache

Datová vyrovnávací paměť cache je 8 kB s řízeným mapováním, virtuálním indexováním, virtuálně značená vyrovnávací paměť. Hlídní cache při zápisu je podporováno zápisem přes bez zapisovací alokace. Datová cache je organizována jako 512 řad skládajících se z 16 Bytu a 32 bitu. Datová vyrovnávací paměť poskytuje nulovou chybovost datového přístupu k cache.

K redukci chyb při zápisu do paměti datová cache obsahuje 4-úrovňový dvouslovný odkládací buffer. K redukci chyb při čtení obsahuje datová cache podporu znovunaplnění do dvou 32 bitových slov, proudy a odbočení.

DRAM Interface- Ovladač DRAM

Ovladač microSPARC-IIep DRAM podporuje průmyslový standard rychlostránkových modulů DRAM a EDO DRAM, který má podporu rychlostránkových módů. Obsahuje 8 paměťových banků s maximální velikostí do 256 MB systémové paměti.

Jeden paměťový bank může být obsazen 8 MB, 16 MB nebo 32 MB paměti. Ovladač DRAM je programovatelný a podporuje paměti s rozdílnými relativními rychlostmi vůči rychlosti frekvence procesoru.

DRAM data bus je 64 bitu široký se dvěma paritními bity, jeden kryje vždy 32 bitů dat. Paritní bity mohou být odstraněny, zamítnuty.

DRAM ovladač poskytuje programovatelný DRAM obnovovací ovladač, který podporuje CAS-before-RAS obnovování.

PCI Bus Interface - Ovladač PCI Bus

Ovladač PCI bus je založen na průmyslovém standardu PCI Bbus Specification verze 2.1. Poskytuje 32 bitový 33 MHz ovladač. Jeho hlavními rysy obsahují především konverzi pro rozdílné datové typy, PCI Host and Satellite mode, vestavěný 16-vstupový překladový oboustranný buffer (TLB) vestavěný pro i/o přenosy, PCI arbiter, PCI hodiny, držitel resetu a přerušení. Jako PCI Host, microSPARC-IIep plní funkci vykonavatele PCI busu, PCI hodin, držitele resetu a volání přerušení. Jako PCI Satellite, microSPARC-IIep přeměrovává funkce PCI Host na externí zařízení typu PCI Host.

- Ostatní vlastnosti podporovány PCI Bus ovladačem obsahují:
- Programovatelné vstupně/výstupní (PIO) přenosy mezi microSPARC-IIep a externími PCI zařízeními
- PCI Host nebo PCI Satellite mod nastavený pomocí vstupních pinů během chodu
- Programovatelné konfigurace externích PCI zařízení v módu PCI Host
- Programovatelné konfigurace podle externích PCI Host v módu PCI Satellite
- Buffery se srovnatelným plnicím a vyprázdňovacím poměrem dovolujícím rozšířený přenos
- Řízené přenosy mezi PCI masters a PCI slaves
- Dva PCI busy přináší uspokojivý přístup k vykonávaným protokolům: same-level round robin a three-level round robin
- Řízený přístup do virtuální paměti (DVMA) mezi PCI masters a ovladačem paměti microSPARC-IIep slave použitím softwarového ovladače IOTLB ke generování fyzické DRAM adresace
- Bootování z PCI Bus z adresy určené uživatelem
- Dokud je v standby módu k redukci spotřeby, PCI ovladač spotřebovává minimální množství energie k podpoře generátoru hodinových impulsů a čekání na přerušení
- K dispozici jsou dva 32 bitové hodiny nebo jedny 32 bitové hodiny a jeden 64 bitový čítač
- Držitel přerušení podporuje až 8 programovatelných přerušení vstupně/výstupního vlákna

Flash Memory Interface-Ovládač mžikové paměti

Použitím průmyslového standardu programovacího algoritmu je mžiková paměť microSPARC-IIep kompatibilní s 28xxxXX flash memory devices. Ovládač má programovatelnou schopnost přidělit čas přístupu do paměti až 45x vyšší než jsou hodiny procesoru. Po zapnutí je nastaven ovládač na 45 násobek hodin procesoru. Poté je možno ovládač přístupu do paměti programově nastavovat v rozmezí 6 až 45 násobku hodin procesoru přidáváním nebo udiráním vždy 3 násobku hodin procesoru.

Ovládač flash paměti podporuje až 16 MB dat. Podporuje oba jak 32-bitový, tak i 8-bitový přístup při nastavení boot-mode na vstupních pinech. MicroSPARC-IIep je pin-selectable (nastavitelný pomocí pinu) na oba typy z buď flash paměti nebo PCI adresace.

JTAG Test Bus Interface-Tester ovladače

MicroSPARC-IIep poskytuje pět-vláken test access port (TAP) ovladače k podpoře scanování a ovládání hodin.

Tento ovládač je kompatibilní se specifikací IEEE1149.1, IEEE Standard Test Access Port and Boundary Scan Architecture. To zaručuje efektivní přístup k jakýmkoliv samotným čipům.

TAP Cont roller

Ovládač TAP je synchronně sestavený komponent (finite state machine (FSM)), který kontroluje sekvence operací JTAG, v závislosti na změnách v JTAG busu. Ovládač TAP je asynchronní se systémovými hodinami a může být použit ke kontrole hodinové logiky.

TAP FSM rozpoznává diagram rozložení státu podle specifikace IEEE 1149.1.

Power Management

MicroSPARC-IIep může detekovat systémovou inaktivitu a nastavit sám sebe do standby modu pro úsporu energie. Kdykoliv ovládač PCI postrádá aktivitu a nějaká PCI aktivita vyčkává na procesor, je zapnut standby mód.

6.8 Procesory firmy AMD

Společnost **Advanced Micro Devices** si za celou svou dobu existence vybudovala slušnou pozici druhého největšího dodavatele procesorů pro PC. Zaměření firmy je především na trh PC střední a nižší výkonové třídy, kdy se snaží konkurovat ostatním stejným výkonem za nižší cenu.

Po větších problémech v polovině 90. let se ale směle staví svému největšímu konkurentovi na poli procesorů - firmě Intel. Výrobou procesorů, které sice přicházejí na trh se zpožděním oproti konkurenci, se procesory dobře prodávají především kvůli příznivé ceně.

Převážná část výroby a celé vedení firmy je situována do nejmodernější továrny na výrobu procesorů [42] v americkém státě Texas ve městě Austin.

Továrna je nazývána Fab25 a kromě procesorů se zde vyrábí také další polovodičové součástky. Svou kapacitou a rozlohou patří mezi největší továrny.

Na ceně více jak 1,5 miliardy dolarů se největší měrou podílí velkoprostorová místnost "clean room" udržovaná v bezprašném stavu.

Kromě továrny v Americe vlastní také ve s Fujitsu Limited továrnu v Japonsku zaměřenou na výrobu paměťových čipů a karet. V roce 1996 započala výstavba továrny v Evropě a to v německých Drážďanech.

Firma AMD zaměstnává ve všech svých továrnách více jak 17 200 pracovníků.

Na testování, analýzu a výrobu dalších zařízení jsou používány továrny v Thajsku, Malajsií a Singapuru.

Vývoj procesorů od 486 po AMD-K6

Firma AMD uvedla svůj procesor Am486 v dubnu roku 1993 a to procesor **Am486** na 40 MHz a **Am486DX2** na 25/50 MHz. Modely **Am486DX4** využívající ztrojnásobenou vnitřní frekvenci se stali nejspěšnějším procesorem pro desky 486. Na další uvedení vylepšeného procesoru Am5x86 nazývaného též **Am486DX5** si museli zákazníci počkat až do roku 1995. To již byly na trhu procesory Pentium a byl také ohlášen první model procesoru Pentium Pro. **Am486DX5** byl podobný jako Pentium, ale osazoval se do desek pro procesor 486. Jeho vnitřní frekvence byla 5-ti násobná a byl vyráběn 0.35 mikronovou technologií. Teprve v březnu roku 1996 představila firma AMD vlastní klony Pentia, nazvané AMD5K86. Byly ve verzích P75 a P90. V červnu byly nahrazeny procesorem **K5PR100**, kde AMD použilo značení PR (Pentium Rating) značící výkon odpovídající stejnému typu Pentia. Procesor byl také úplně kompatibilní ve všech aplikacích.

V době, kdy procesory **K5** začínaly, byly technologicky na úrovni Pentia, ale výkonově na tom byly o něco hůře, a to především špatně zpracovanému matematickému koprocetoru. Přesto se velmi dobře prodávaly do levnějších sestav. V říjnu roku 1995 padlo rozhodnutí zakoupit firmu NexGen, která měla konkurenční procesor Nx586. Transakce, která stála 863 milionů dolarů, se firmě AMD velmi vyplatila, protože procesor NexGen byl použit pro základ budoucího procesoru **AMD-K6**. Firma AMD dokonce připustila, že jejich návrh nového procesoru nebyl tak dobrý. Tímto procesorem AMD smazala svůj dlouholetý technologický deficit a představila procesor, který je velmi výrazným konkurentem procesorům firmy Intel, a to i ve vyšších výkonnostních třídách sestav. Výkonově vychází procesor **AMD-K6** lépe než procesor Intel Pentium MMX na stejné frekvenci, je však ve verzi 233 MHz levnější téměř o 30%. I když procesor nedosahuje podle benchmarkových testů takového výkonu jako Pentium II, je výborným marketingovým tahem společnosti AMD tím, že **K6** využívá stávající patici Socket 7, která je využívána i u Pentii. To znamená, že výrobci základních desek nemusejí navrhovat nové desky, stačí pouze přepracovat ROM BIOS tak aby akceptoval i procesor K6. Tato situace na trhu vede k cenové válce výrobců, kdy je společnost Intel nucena snížit ceny svých procesorů, a to je samozřejmě velice příjemné pro uživatele.

6.8.1 Procesor Am486

Vlastnosti procesoru:

- write back cache se zpožděným zápisem
- 3V napájecí napětí a zmenšená spotřeba energie při neaktivitě procesoru
- řízení vnitřními hodinami
- ztrojnásobená vnitřní frekvence
- standardní 168 pinová PGA patice



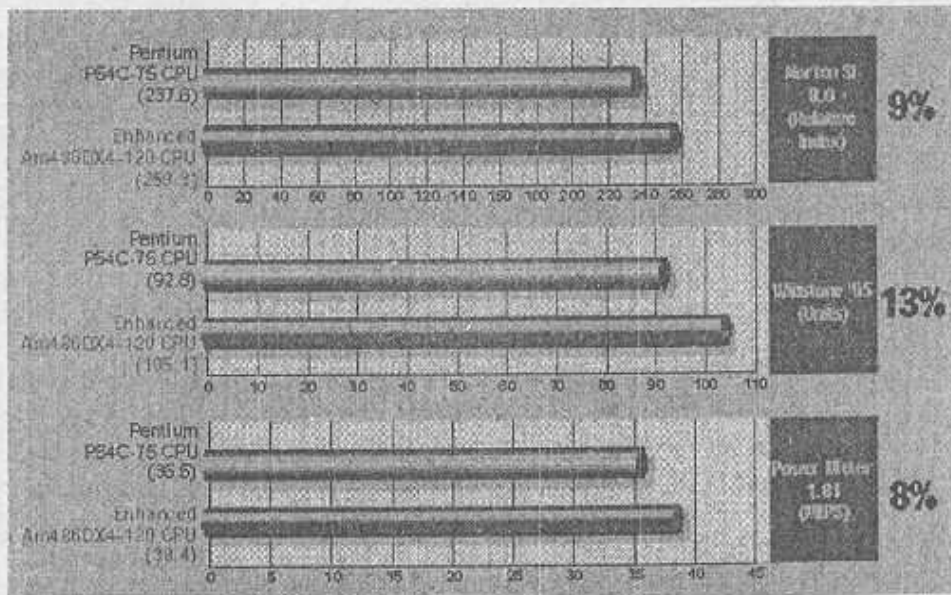
Řady procesorů Am486

Tabulka č. 6.11

	frekvence	označení
DX2	20/40 MHz	Am486DX2-40
	33/66 MHz	Am486DX2-66
	40/80 MHz	Am486DX2-80
DX4	33/100 MHz	Am486DX4-100
	40/120 MHz	Am486DX4-120

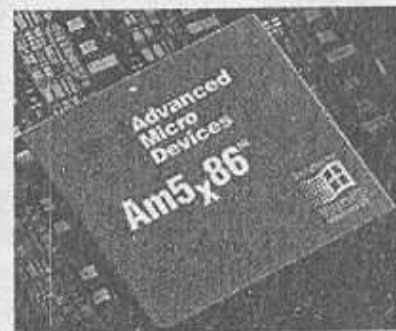
Výkonostní testy

Srovnání AmDX4-120 s Procesorem Intel Pentium 75 Mhz pomocí nejběžnějších testovacích programů (Norton Utils, Winstone 95, Power Meter) s procentuálním vyjádřením zrychlení oproti procesoru Pentium:



6.8.2 Procesor Am5x86

Vlastnosti procesoru Vnitřní rychlost procesoru je 4-násobná a používá 16KB write-back cache. Pinově je kompatibilní s patičkami 168-PGA pro procesory 486 a využívá výhody rychlé sběrnice PCI. Integrovaný power management umožňuje použití napájení 3V. Procesor je vyráběn pokročilou 0,35μm technologií. Je také kompatibilní se všemi nejběžnějšími operačními systémy používanými s současností.

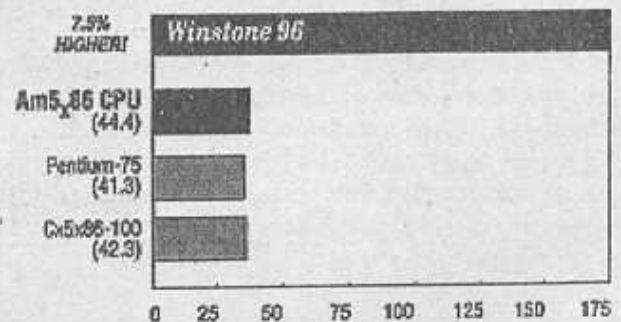


Tabulka č. 6.12 Řady procesoru

Objednací číslo	Typ patice	Napájecí napětí	Teplota
AMD-X5-133ADW	PGA	3,45 V	55°C
AMD-X5-133ADZ	PGA	3,45 V	85°C
AMD-X5-133SFZ	SQFP (notebooky)	3,3 V	85°C
AMD-X5-133SDZ	SQFP (notebooky)	3,45 V	85°C

Výkonostní testy

Výkonostní testy probíhali na stejné standardní sestavě s pomocí testovacího softwaru Winstone 96. Výsledky ukazují, že výkon procesoru Am5x86 je vyšší než výkon procesoru Intel Pentium 75 Mhz a Cyrix 5x86-100 při nižší ceně a při zachování pinové kompatibilitě s patičkami na systémových deskách procesorů třídy 486.



Testovaná konfigurace

PCI Motherboard

- Am5X86 základní deska s UMC UM881F chipsetem
- Cx5X86-100 základní deska s UMC UM881F chipsetem
- Pentium-75 motherboard používá

Intel Triton chipset

PCI Diamond Stealth 64 grafická karta
WD Caviar 2850 hard disk, 850-MB EIDE
256kB L2 Cache (15 ns)
16-MB DRAM (70 ns)
Windows 95

6.8.3 Procesor AMD-K5

Základní charakteristika

Procesor AMD K5 byl představen firmou AMD 17. června 1996.

Jeho hlavními rysy jsou:

- čtyřproudové superskalární jádro s šesti vykonávacími jednotkami uspořádanými do pětistupňové posloupnosti (pětistupňová posloupnost=1. vyzvednutí příkazu z operační paměti nebo cache, 2. dekódování instrukce, 3. zjištění operandové adresy příkazu, 4. zpracování instrukce, 5. uložení výsledku instrukce)
- 16 kBytová čtyřproudová obousměrná asociativní řádková instrukční vyrovnávací paměť
- 8 kBytová obousměrná 2-bránová čtyřproudová asociativní řádková datová vyrovnávací paměť
- techniky pro předpověď větvení a provádění instrukcí mimo pořadí.
- 1 kBytová vyrovnávací paměť využívaná pro techniky předpovědi větvení integrovaná, vysoce výkonná matematická jednotka, pracující s pevnou i pohyblivou čárkou, s velice krátkou dobou na provedení výpočtu
- statické řízení hodin pomocí takzvaného PLL (Phase Lock Loop) schématu
- 3,3 V provoz a SMM systém (System Managment Mode) pro řízení sníženého příkonu
- 64-bitová sběrnice s volitelnou frekvencí
- 296 vývodové SPGA pouzdro
- kompatibilita s existující standardem P54C=standart pro podporu Pentia na jiných infrastrukturách a systémech
- kompatibilita s DOS, Microsoft Windows 95, Microsoft Windows NT, Novell Netware a OS/2 Warp operačními systémy a DOS a Microsoft Windows kompatibilními softwarovými produkty (procesoru AMD-K5-PR100 bylo po testech v laboratořích Microsoft Windows Hardware Quality uděleno logo Windows 95 na důkaz plné kompatibility s tímto operačním systémem).
- plná kompatibilita se všemi předcházejícími x86 procesory
- kompatibilita se standardem pro operace s plovoucí desetinnou čárkou

Procesor AMD K5 využívá nezávisle vytvořený superskalární design založený na RISCovém procesoru vyráběném pomocí moderní 0,35 mikronové Complementary Metal-Oxide Semiconductor (CMOS) technologie. Návrh architektury procesoru vychází z bohatých zkušeností v RISCové a x86 technologii, poskytující pevný základ pro výrobu procesoru v pouzdře SPGA o 296 vývodech a obsahujícím přibližně 4,3 miliónu tranzistorů. Tento procesor je řazen do páté generace. Během jediného cyklu je možné zpracovat až čtyři instrukce v šesti prováděcích jednotkách použitím technik předpovědi větvení a provádění instrukcí mimo pořadí. Oproti svému předchůdci z dílem AMD je procesor AMD-K5 vybaven možností instrukčního předde-

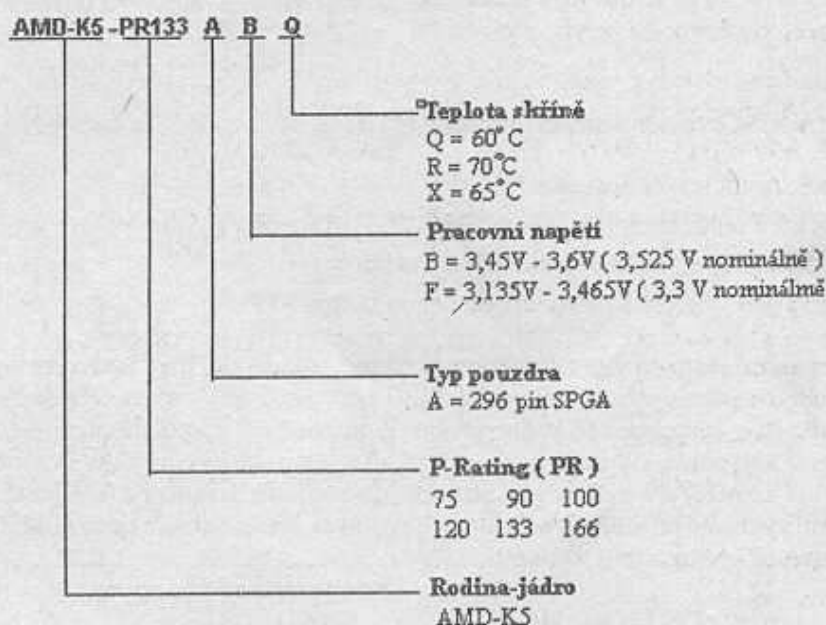
kódování, zlepšenou architekturou vnitřních vyrovnávacím pamětí a rychlejším přístupem k těmto vyrovnávacím pamětím a možností větvení a předpovědi.

Seznam procesorů řady K5

Tabulka č. 6.13

CPU ID				
ID	MODEL ID	CPU Frekvence (MHz)	Rychlost sběrnice	Processor
5	0	75	50	AMD-K5-PR75
		90	60	AMD-K5-PR90
		100	66	AMD-K5-PR100
	1	90	60	AMD-K5-PR120
		100	66	AMD-K5-PR133
		116.7	66	AMD-K5-PR166

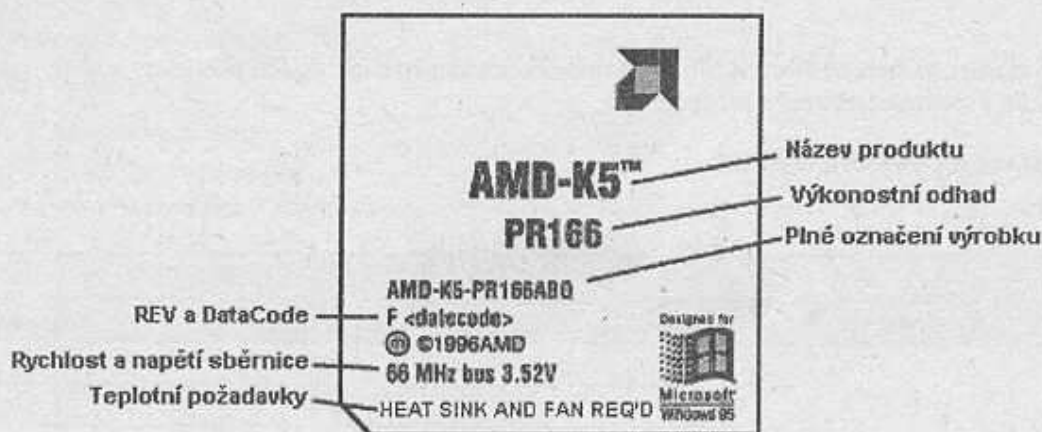
Popis značení procesorů AMD-K5 a možné kombinace



P-Rating - hodnocení procesorů, na kterém se dohodli výrobci čipů kompatibilních s čipy Intelovskými. Toto číslo říká, že např. procesor ohodnocený značkou PR-120 má výkon Pentia 120 nebo vyšší. Výkon se přitom měří uživatelským benchmarkovým testem, ovšem výlučně pod systémem Windows 95.

Tabulka č. 6.14

Processor	Typ pouzdra	Pracovní napětí	Teplota skříně (°C)
AMD-K5-PR166ABX	296-pinové SPGA	3,45 - 3,60	60
AMD-K5-PR133ABR	296-pinové SPGA	3,45 - 3,60	70
AMD-K5-PR133ABQ	296-pinové SPGA	3,45 - 3,60	60
AMD-K5-PR120ABR	296-pinové SPGA	3,45 - 3,60	70
AMD-K5-PR100ABQ	296-pinové SPGA	3,45 - 3,60	60
AMD-K5-PR90ABQ	296-pinové SPGA	3,45 - 3,60	60
AMD-K5-PR75ABR	296-pinové SPGA	3,45 - 3,60	70



Obr. č. 6.45

Superskalární RISCové jádro:

RISCové procesory (neboli procesory s redukováným instrukčním souborem) používají takové koncepcce návrhu architektury procesoru, aby bylo možné dosáhnout zpracování jedné instrukce během jednoho strojového cyklu procesoru, což je obvykle prováděno pomocí optimalizujícího překladače, který program napsaný v problémově orientovaném jazyku přetvoří do jednoduchých strojových instrukcí, které je procesor schopen rychle vykonávat.

Superskalární RISCové jádro samotného procesoru AMD-K5 se skládá ze šesti vykonávacích jednotek:

- dvou aritmeticko-logických jednotek (ALU)
- dvou načítacích a ukládacích jednotek (LOAD / STORE Units)
- jedné jednotky pro předpověď větvení (Branch Unit)
- jedné jednotky pro výpočty s plovoucí desetinnou čárkou (FPU)

Toto superskalární jádro je naprosto nezávislé na instrukcích x86 díky konverzi instrukcí x86, které mají proměnnou délku, na jednodušší stejně dlouhé RISC operace (ROPs), které jsou jednodušší a rychleji zpracovávány. Jakmile jsou instrukce x86 konvertovány, jsou poslány superskalárnímu jádru po 4 ROPsech. Samotné superskalární jádro pak může zpracovat až 6 ROPsů v jeden strojovém cyklu. Superskalární jádro dále podporuje přednačítání instrukcí pro vykonávání po sobě následujících instrukcí a také přeskokování instrukcí pro okamžité urychlení vykonání po sobě nenásledujících instrukcí, čímž se eliminuje zpoždění pro nové načítání instrukcí právě potřebných pro okamžitý výpočet.

Zpracování instrukcí mimo pořadí

Procesor AMD-K5 implementuje tzv. Out of Order Execution, tedy jakési spekulativní provádění instrukcí mimo dané pořadí. Každá vykonávací jednotka má dvě rezervovaná místa (vyjimku tvoří jednotka pro operace s plovoucí desetinnou čárkou FPU, která má pouze jedno rezervované místo), kde jsou udržovány ROPsy přednostně určené pro zpracování. Tyto ROPsy pak mohou být vydány a zpracovány mimo původní pořadí zpracování instrukcí. Pro zajištění správné funkce tohoto postupu slouží šestnácti-vstupový reorder buffer. Tento buffer je využíván pro přejmenování registrů (viz. kapitola 6.), poskytuje informace o mezivýpočtu a udržuje výsledek z x86 instrukce, která by měla být zpracována v okamžiku předpovědi větvení.

Když jsou ROPsy odeslány do jedné z šesti procesorových vykonávacích jednotek, je jejich pořadí uloženo na vrchol reorder bufferu. Reorder buffer sleduje originální posloupnost instrukcí a zajistí, že výsledky po vykonání instrukcí jsou vydány ve správném pořadí, a dále zapisuje výsledky prováděných instrukcí do tzv. Register File (viz kapitola 13). Jestliže je daná větev instrukcí při větvení nepředpovězena, výsledky instrukcí náležících do nepředpovězené větve jsou v reorder bufferu zrušeny předtím, než ovlivní procesorové registry nebo paměťový systém.

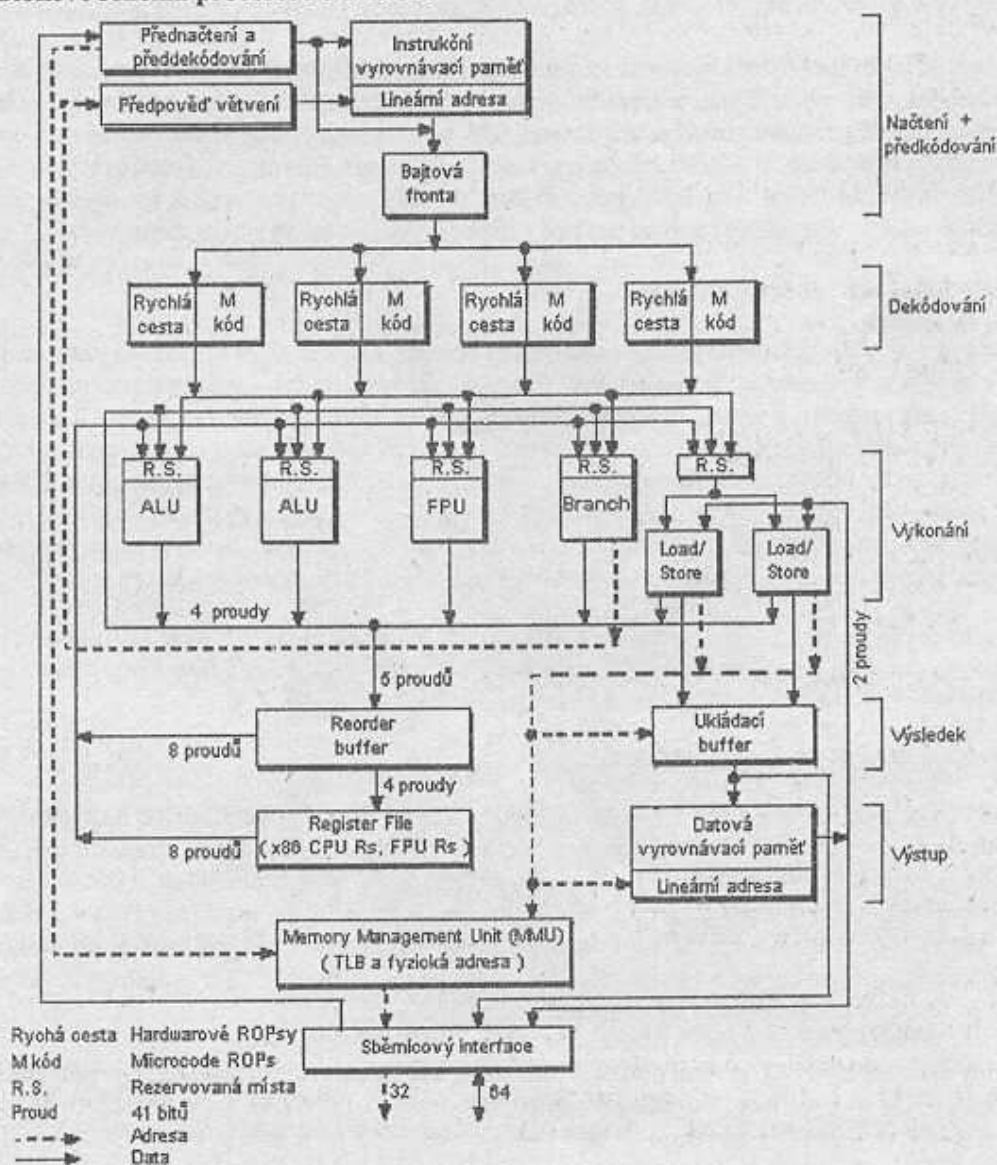
Přejmenování registrů procesoru AMD-K5

Architektura x86 má pouze osm univerzálních registrů, což při použití šesti nezávislých vykonávacích jednotek není mnoho, navíc tyto registry jsou hodně závislé, neboť když je používá jedna jednotka, nemůže jej použít druhá. Procesor AMD-K5 používá jisté přejmenování těchto registrů kvůli odstranění jejich zmíněné závislosti. Jsou tedy definovány pro každý fyzický registr vícenásobné logické registry, které dovolují vykonávacím jednotkám současně použít stejného jména fyzického registru, aniž by došlo ke kolizi.

64-bitová datová sběrnice a rozhraní

Procesor AMD-K5 používá 64-bitovou datovou sběrnice, která poskytuje větší propustnost a zároveň podporu pro 64-bitové přenosové cesty a řádkově naplňovatelnou vyrovnávací paměť pro načítání oddělených interních datových a instrukčních vyrovnávacích paměti. Jakmile data a instrukce vstoupí na rozhraní procesoru, je interní vyrovnávací paměť postupně naplňována rychlostí pěti hodinových cyklů připadajících na naplnění jedné řádky této vyrovnávací paměti. Zvýšená šířka datové sběrnice procesoru a proces naplňování vyrovnávací paměti tak redukuje zpoždění zpracování instrukcí, podporuje nadřazený procesor a celkově zvyšuje systémový výkon.

Blokové schéma procesoru AMD-K5



Obr. č. 46

Předdekódování instrukcí x86

Každý byte kódu, který vstoupí do procesoru je opatřen asociativní informací, která identifikuje hranice jednotlivých x86 instrukcí a umožňuje uspořádání vícenásobných x86 instrukcí (délky od 8 až do 120 bitů). Kromě toho, že předdekódovaná informace oznamuje, kde instrukce x86 začíná a končí, obsahuje navíc počet jednotlivých RISCových operací (tedy ROPsů), které x86 instrukce bude potřebovat pro pozdější překlad. Poté, co jsou instrukce předdekódovány, jsou nataženy do vnitřní instrukční vyrovnávací paměti, pravděpodobně instrukce (x86 instrukce z předpovězené větve) jsou vloženy do Bytové fronty a čekají na pozdější dekodování. Bytová fronta obsahuje nejen x86 instrukce, ale i asociované předdekódované informace, které označují pozici instrukce a typ operace.

Architektura vyrovnávacích pamětí

Procesor AMD-K5 používá oddělenou datovou a instrukční vyrovnávací paměť, a tím jsou eliminovány interní konflikty, které nastávaly současným použitím vyrovnávací paměti jak pro instrukce, tak pro data, jak tomu bývalo u starších procesorů.

Procesorová 16 kBytová instrukční vyrovnávací paměť je obousměrná a poskytuje rychlý převod lineární adresy na adresu fyzickou a má implementovanou čtyřcestnou asociativní strukturu pro dosažení maximálního výkonu této vyrovnávací paměti při dané velikosti (tedy 16 kB).

Procesorová 8 kBytová datová vyrovnávací paměť umožňuje současný přístup ke svým dvěma řádkům dat během jediného hodinového cyklu. Touto schopností dvou přístupů během jediného hodinového cyklu umožňuje datová vyrovnávací paměť překonat překážku zděděnou z x86 architektury. Každý řádek datové paměti je navíc popsán jedním ze čtyřech stavů tohoto MESI protokolu pro identifikaci stavu informace uložené v dané řádce paměti. Díky tomu, že tato paměť má design WriteBack, jsou z ní data do vnější paměti přesouvána, jen když je to nezbytné, čímž udržuje systémovou sběrnici volnou pro jiná zařízení a zvyšuje tak výkon systému.

Předpovídání instrukčních větvení

Větvení se vyskytuje průměrně jednou za sedm x86 operací. Když se nějaké větvení vyskytne, procesor předpovídá, která větev instrukcí bude následovat. Procesor zároveň přidává informace o předpovězeném větvení ke každé řádce instrukcí v interní řádkové instrukční procesorové vyrovnávací paměti. Tato informace indikuje v rámci této řádky v interní instrukční vyrovnávací paměti cílovou adresu první větve instrukcí, u které bylo předpovězeno, že by měla nastat. Tento větvící a předpovídající mechanismus připouští až 1024 větví a 75% správnost předpovědi větvení. I přes to, že při nesprávně předpovězeném větvení následuje trest minimálně 3 cyklů navíc, je tento mechanismus předpovědi větvení výhodný a optimalizuje procesorové vykonávání x86-ého softwaru, jako je např. Microsoft Windows operační systém a přidružené aplikace.

Dynamická předpověď větví procesoru AMD-K5 dále umožňuje to, že jsou instrukce vybírány a posílány vykonávacímu jádru procesoru takovým způsobem, že jsou eliminovány některá nepotřebná větvení a tím dochází k zvýšení výkonu procesoru.

Konverze a dekodování x86 instrukcí

Logický instrukční tok v procesoru AMD-K5 pokračuje po předkódování tak, že 32-Bytové předkódované x86 instrukce jsou vyjmuty z Bytové fronty a posílány postupně do dekodéru. Procesor pak dekoduje tyto složené x86 instrukce do jednotlivých, rychle proveditelných a stejně dlouhých ROPsů. Na začátku dekodovacího procesu dekodér zkoumá složené x86 instrukce a alokuje jednotlivé x86 instrukce v rámci složené x86 instrukce. Tato alokace závisí na 5 bitovém označení, které každá x86 instrukce dostala při překódování. Jakmile složená instrukce projde procesorovým dekodérem, je známo, kolik ROPsů je potřeba na vykonání této instrukce, čímž se šetří čas při následném vykonávání.

Samotný dekodér má 4 dekodovací pozice. Proto x86 instrukce, kterým odpovídají čtyři a méně ROPsů, jsou přímo posílány do dekodéru, a složené x86 instrukce, kterým odpovídá více než čtyři ROPsy jsou přemístěny do MROM (Microcode ROM), kde jsou pak konvertovány na 4 ROPové sekvence.

Po konverzi a dekodování jsou ROPsy paralelně rozepisovány do jednotlivých šesti vykonávacích jednotek.

Memory Managamet Unit (MMU)

Pro podporu standardního x86 požadavku přístupu k virtuální paměti je v procesoru AMD-K5 integrována jednotka Memory Managament Unit, která provádí převod lineární adresy na adresu fyzickou. K zrychlení tohoto převodu jsou nejčastěji prováděné adresové převody uloženy do jednoho ze dvou bufferů (translation lookaside buffer=TLB), jeden pro mapování 4 kBytových stránek a jeden pro mapování 4-MBytových stránek paměti.

Během přístupu do paměti MMU obdrží lineární adresu a prohledává všechny TLB, jestli neobsahují odpovídající fyzickou adresu. Jestliže je nalezena, je poslána pro další zpracování. Jestliže fyzická adresa není nalezena v seznamu TLB, MMU prohledává seznam stránek v paměti. Jestliže je v nich nalezena, MMU natáhne překlad fyzické adresy dané stránky do příslušného TLB. Jestliže není nalezena, procesor generuje chybu.

Čipové sady pro procesor AMD-K5

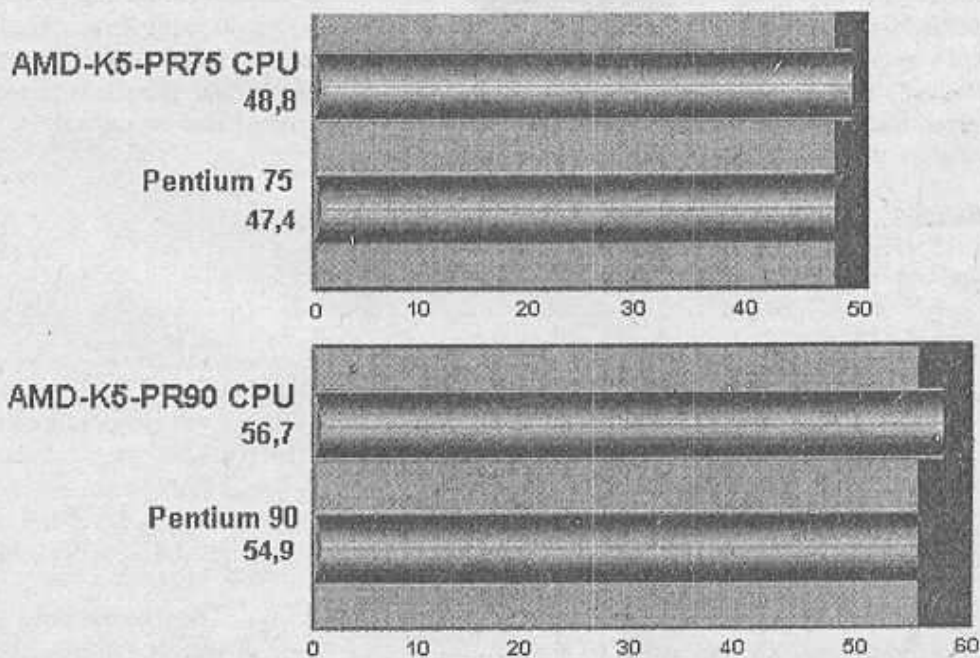
Mezi čipové sady podporující procesor AMD-K5 patří např. Intel 430 VX, SiS 5596, SiS 5571, VIA 580 a mezi výrobce čipových sad podporujících procesor AMD-K5 patří např. VLSI Technology, Opti, VIA, UMC, SiS a další.

Systémové desky

K5 je vývodově kompatibilní s klasickým Pentiem a podle údajů firmy AMD jej stačí pouze osadit do základní desky a AMD-K5 by měla bez problémů pracovat. Bohužel však toto není tak zcela pravda. Například základní desky pocházející přímo z dílen Intelu se rozhodly konkurenční procesor zcela ignorovat a chovají se, jako kdyby v nich žádný procesor nebyl. Nepomůže ani upgrade BIOSU. S procesorem si nerozumějí některé desky ATX, desky typu SOYO 5SA2 a SOYO 5SA5. Deska ASUS typu P/I-P55TP 4XE chybně ohlásila přítomnost procesoru 486DX/66, ale přesto K5 pracovala na plný výkon. U desky FIC PT2003 s čipovou sadou INTEL 430FX byly detekce i výkon procesoru bezchybné.

Srovnání procesorů AMD-K5 s procesory Pentium

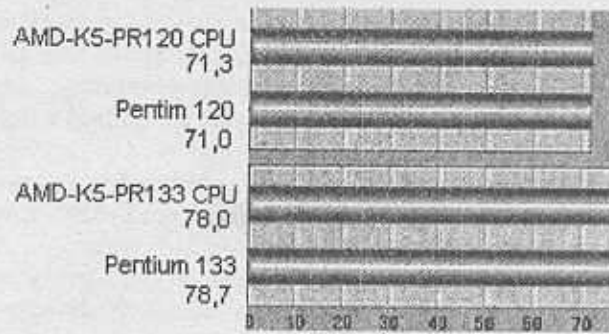
– AMD-K5-PR75 a 90 versus Pentim 75 a 90



AMD-K5-PR75=48,8 x Pentium 75=47,4 AMD-K5-PR90=56,7 x Pentium 90=54,9

Obr. č. 6.47

– AMD-K5-PR120 a 133 versus Pentium 120 a 133



Obr. č. 6.48

AMD-K5-PR120=71,3 x Pentium 120=71,0 AMD-K5-PR133=78,0 x Pentium 90=78,7

Srovnání procesorů bylo prováděno při konfiguraci:

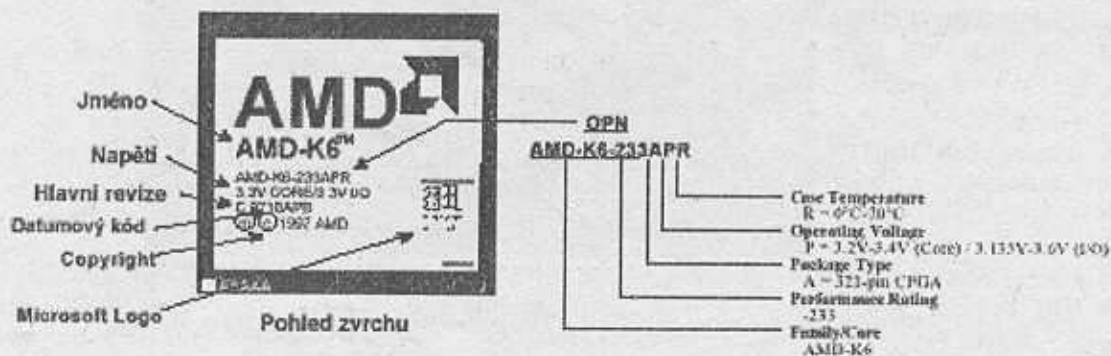
- FIC PA-2005 Motherboard
- VIA Apollo VP1 chipset
- Award BIOS 6.01J900
- 16-MB EDO DRAM (60 ns)
- 256-kB Pipeline Burst SRAM L2 cache
- Quantum Fireball 1.2 GB EIDE harddisk
- Diamond Stealth64 3200 2MB VRAM PCI video kartě
- Na systému Windows 95

6.8.4 Procesor AMD-K6

Procesor šesté generace AMD-K6 využívá superskalární architektury RISC86, takže může vykonávat šest instrukcí současně. K dispozici má sedm paralelních výkonných jednotek, z toho jedna je jednotka MMX a jedna pro počítání v plovoucí čárce. Jeho součástí je rovněž několik sofistikovaných dekodérů instrukcí z instrukční sady x86 do instrukční sady RISC86. Přímou v procesoru jsou obsaženy velké primární vyrovnávací paměti, instrukční a datová, po 32 KB. Součástí procesoru je vysoce výkonná aritmetická jednotka pro výpočty v plovoucí čárce a další jednotka pro provádění instrukcí MMX.

Pohled zvenčí

Výrobu zajišťuje továrna AMD Fab 25, čip je vyráběn v pětivrstvé technologii 0,35 mikronu, obsahuje 8,8 milionů tranzistorů (Pentium MMX jich má 4,5 milionů). Od IBM licencovala AMD technologii "Flip Chip", kdy lze vést vývody nejkratší cestou, nikoli pouze z okraje čipu. Vnitřní napájecí napětí je 2,9V.



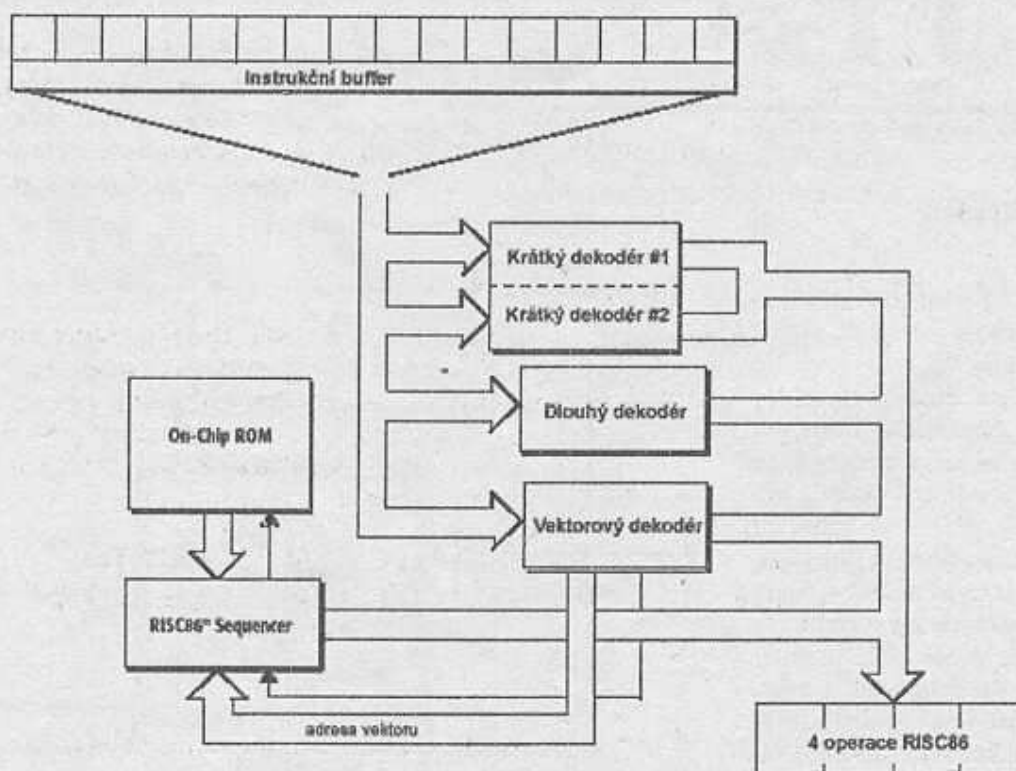
Obr. č. 6.49

Dekodéry Dekódování instrukcí x86 začíná, když je naplněna vnitřní cache v procesoru. Procesor může natáhnout během jednoho hodinového cyklu až 16 Bytů z instrukční cache. Natažené instrukce jsou přesunuty do 16 Bytového instrukčního bufferu, který je posílá přímo dekodéru. Natahovací logika je schopná vytáhnout z 32 Bytového instrukčního rámce 16 za sebou jdoucích instrukčních Bytů. Logika je schopna to zvládnout i v případě, že se těchto 16 Bytů nachází na rozhraní dvou rámců. Ani v takovémto případě se neztratí hodinový cyklus. Poté jsou instrukce přesunuty do instrukčního bufferu a zpracovány dekodéry. Přestože instrukce jsou zarovnávané s Bytovou přesností, instrukční buffer je zarovnávaná na wordy. Proto jsou instrukce z instrukční cache také poskytována po blocích zarovnaných na celé dvou Byty. Když přijde ovládací instrukce (např. skoková - JMP), celý instrukční buffer je vyprázdněn a nahrazen novou sadou 16 instrukčních Bytů.

Dekódovací logika je navržena k tomu, aby uměla zpracovat více x86 instrukcí během jediného hodinového cyklu. Dekodér přijímá Byty x86 instrukcí a jejich predekódované bity z instrukčního bufferu, označuje aktuální hranice instrukcí a generuje z nich operace RISC86. Operace RISC86 jsou vnitřní instrukce s pevným formátem. Mnoho z nich je vykonáno během jediného hodinového cyklu. Z instrukcí RISC86 lze složit každou instrukci x86. Existují instrukce x86, které po dekódování nejsou žádnou RISC86 operací - např. instrukce NOP - nebo pouze jedinou operací - přičítání z registru do registru. Komplexnější instrukce x86 jsou dekódovány do několika RISC86 operací.

K6 obsahuje tři sady dekodérů - dva paralelní krátké dekodéry, jeden dlouhý a jeden vektorový dekodér. Paralelní krátké dekodéry jsou určeny pro překládání nejčastěji užívaných x86 instrukcí (přesuny, rotace, větvení, ALU, MMX, FPU) do žádné, jedné nebo dvou operací RISC86. Krátké dekodéry zpracovávají instrukce x86 pouze tehdy, pokud jsou kratší než 8 Bytů. Navíc jsou konstruovány tak, aby byly schopné přeložit dvě krátké instrukce během jediného hodinového cyklu. Často jsou v kódu zastoupeny instrukce delší než 7 Bytů, ale kratší než 12 a ty jsou zpracovány nepárovým dlouhým dekodérem, který je překládá až do 4 RISC86 operací. Všechny ostatní instrukce jsou zpracovány kombinací vektorového dekodéru a sekvencí RISC86 instrukcí načtenou z paměti ROM čipu.

Přestože všechny tři sady dekodérů načítají instrukci z instrukčního bufferu, během jediného hodinového cyklu může být použit pouze jeden typ dekodéru.



Obr. č. 6.52

Centrální plánovač

Plánovač je srdcem procesoru AMD-K6 MMX. Obsahuje logiku, která je potřebná k provádění instrukcí mimo pořadí, postupování dat, přejmenování registrů a zpracování více RISC86 instrukcí zároveň. Buffer v plánovači má kapacitu 24 RISC86 instrukcí, což odpovídá maximálně 12 x86 instrukcím. Pokud je to možné, plánovač může simultánně rozepisovat operace RISC86 do jakékoliv volné exekutivní jednotky (ukládací, načítací, větvící, celočíselné, multimediální nebo FPU). Celkem může plánovač rozepsat až šest a odebrat 4 RISC86 operace během jediného hodinového cyklu. Výhodou plánovače je to, že si může až 12 operací přeorganizovat tak, aby optimalizoval jejich provádění. To je způsobeno paralelním řízením toku operací RISC86. Přestože plánovač rozepisuje RISC86 i mimo pořadí, z dekodéru přijímá dekodované x86 instrukce sekvenčně.

Exekutivní jednotky

Procesor AMD-K6 MMX obsahuje šest základních exekutivních jednotek - ukládací, načítací, větvící, celočíselnou X, celočíselnou Y, multimediální a jednotku pro počítání v plovoucí čáře.

Předvídání skoků

Logika předvídání skoků v AMD-K6 je implementována proto, aby zamezila zbytečným prodáváním při řešení podmínečných skokových instrukcí. Dvouúrovňový algoritmus na zpracování historie skoků posílá své výsledky do výsledkové tabulky o 8 192 položkách. Aby procesor nemusel mít příliš velkou tabulku a mohl disponovat spíše více položkami, do tabulky se neukládají předpovězené cílové adresy. Místo toho jsou plynule dopočítávány v případě potřeby speciální aritmetickou jednotkou během doby, kdy jsou instrukce v dekodéru. Jednotka spočítá všechny možné adresy a procesor si po dokončení vybere tu adresu, která platí.

K tomu, aby nedošlo ke ztrátě hodinového cyklu, pokud by byla použita předpovězená větev programu, vestavěná cache obsahuje 16 Bytů instrukcí, které pak okamžitě posílá do instrukčního bufferu u dekodéru. Celková úspěšnost předvídání skoků je vyšší než 95%.

Zásobník návratových adres

Zásobník návratových adres je speciální zařízení určené k tomu, aby optimalizoval zpracování párových instrukcí CALL a RET. Programy jsou běžně kompilovány s rutinami, které jsou často volány pomocí instrukce CALL. V tomto okamžiku procesor odloží aktuální adresu do zásobníku a skočí do rutiny, kterou začne provádět. Na konci je instrukce RET, při které procesor ze zásobníku opět natáhne adresu, z níž byla rutina volána a vrátí se na ni. Aby K6 zkrátil čas natahování návratové adresy ze zásobníku, udržuje si ji v rychlé paměti, odkud ji použije, když přijde instrukce RET.

Jednotka provádění skoků

V této jednotce nastává spekulativní provádění instrukcí. Tato jednotka dává procesoru možnost řešit instrukce za podmíněnými skoky ještě předtím, než se k těmto instrukcím program dostane, a dokonce ještě před tím, než je jisté, že tato větev programu bude použita. Procesor totiž průběžně neobnovuje x86 registry, ale uchovává si výsledky ve svých vnitřních registrech. Pokud je předpověď nesprávná, všechny použité registry se vynulují a procesor se chová, jako by k předvídání vůbec nedošlo. AMD-K6 může takto zpracovat až sedm nezávislých větví.

AMD-K6 MMX je pakován do keramické PGA patice podle specifikace Socket 7. Tím je pro návrháře desek jednodušší jeho podporu integrovat do svých výrobků, protože stačí pouze změnit ROM BIOS a povolit svoje hodnoty napájení.

Programátorský pohled

Z hlediska programátora je v první řadě pro potřeby softwaru určení typu a vlastností procesoru. K tomu slouží instrukce CPUID. Ta vrací informace o výrobci, typu a dalších parametrech procesoru. Na základě těchto informací se může software optimalizovat pro daný procesor. Příkladem může být herní program, který s pomocí instrukce CPUID detekuje výkonný procesor a zpřístupní volby v ovládacím menu hry, které jsou

náročnější. V poslední době je další nejčastější využití instrukce CPUID v detekci podpory MMX. Důležitá je také správná detekce procesoru při zapnutí počítače, což zabezpečuje program umístěný v ROM BIOS. Při detekci procesoru v BIOSU se dále používají rutiny na výpočet vnitřní rychlosti procesoru a rychlosti sběrnice procesoru, které slouží jako identifikační string, který je viditelný při bootovacím procesu.

Tabulka výsledků instrukce CPUID a inicializační string vypisovaný BIOSem počítače pro AMD-K6

Tabulka č. 6.15

Typ procesoru	Model procesoru	Rychlost procesoru (MHz)	Rychlost sběrnice procesoru (MHz)	Inicializační string
5 (AMD-K6 procesor)	6	166	66	AMD-K6™ /166
		200	66	AMD-K6™ /200
		233	66	AMD-K6™ /233
		266	66	AMD-K6™ /266
	7	166	66	AMD-K6™ /166
		200	66	AMD-K6™ /200
		233	66	AMD-K6™ /233
		266	66	AMD-K6™ /266
		300	66	AMD-K6™ /300

Výpis 1: Detekování přítomnosti instrukce CPUID

Tabulka č. 6.16

pushfd	; uschování obsahu EFLAGS
pop eax	; uložení EFLAGS na EAX
mov ebx, eax	; uložení v EBX pro další použití
xor eax, 0020000h	; zamaskování 21 bitu
push eax	; uložení na zásobník
popfd	; uschování změněný EAX do EFLAGS
pushfd	; uložení EFLAGS na TOS
pop eax	; uložení EFLAGS na EAX
cmp eax, ebx	; test, jestli se 21 bit změnil
jz NO_CPUID	; když není změněn, není podpora pro CPUID

Technologie MMX

Analýza multimediálních a komunikačních aplikací, jako grafika, video MPEG, syntéza řeči a hudby, rozpoznávání řeči, zpracování obrazu, hry, videokonference, komprese a dekomprese dat aj. ukázala, že tyto aplikace jsou typické následujícími příznaky:

- pracují převážně s 8bitovými (obrazové body, pixely) a 16bitovými (zvuk) daty,
- pravidelné a opakované přístupy do paměti (smyčky),
- často se opakující stejné operace s daty (sčítání, násobení),
- jsou výpočetně náročné,
- lze je poměrně snadno paralelizovat.

Pomocí technologie SIMD (Single Instruction, Multiple Data), charakterizované paralelním prováděním jednoduché operace na více datových položkách, což dnešní superskalární procesory umožňují, bylo možné vytvořit architekturu MMX. Ta se skládá s osmi nových 64bitových registrů, čtyř nových datových typů a 57 nových (celočíselných) instrukcí. Při práci s MMX registry je vše řešeno tak, že jsou 64bitové registry "namapovány" na registry zpracovávající reálná čísla (FP) a jsou od nich odlišena příznakem. Takto je také

zajištěna zpětná kompatibilita. Do 64bitového slova se tak vejde osm pixelů, které se v registru MMX zpracují najednou. Tabulka uvádí, kolik je potřeba vykonat instrukci pro dané operace:

Tabulka č. 6.17

operace	počet instrukcí bez MMX	počet instrukcí s MMX
načtení (load)	32	6
násobení	16	4
sčítání	12	2
jiné	8	12
uložení (store)	4	4
celkem	72	28

Aplikace bez využívání MMX spuštěné na procesoru s MMX poběží rychleji, podle měření o 10-20%. Při provozování nových aplikací, využívajících MMX je ve srovnání s procesorem bez podpory MMX nárůst výkonu o 70% u videa, přes 100% u audia a až 400% u zpracování obrazu. Taky počet instrukcí k manipulaci s třírozměrnými objekty (otáčení, změna velikosti, posun a aktualizace), která se děje pomocí násobení matice 4x4 vektorem, klesl o 61%.

Při psaní nových aplikací, které by měli využívat technologie MMX, je potřeba zjistit, zda procesor podporu MMX, což je možné použitím instrukce CPUID. Při instalaci aplikace je pak možné buď nainstalovat jen tu část, která odpovídá vlastnostem daného procesu nebo nainstalovat aplikaci celou a další funkce programu zpřístupnit jen po zjištění podpory MMX.

Aplikace napsané pro MMX na procesorech bez podpory MMX poběží ovšem bez možnosti využití vlastností technologie MMX.

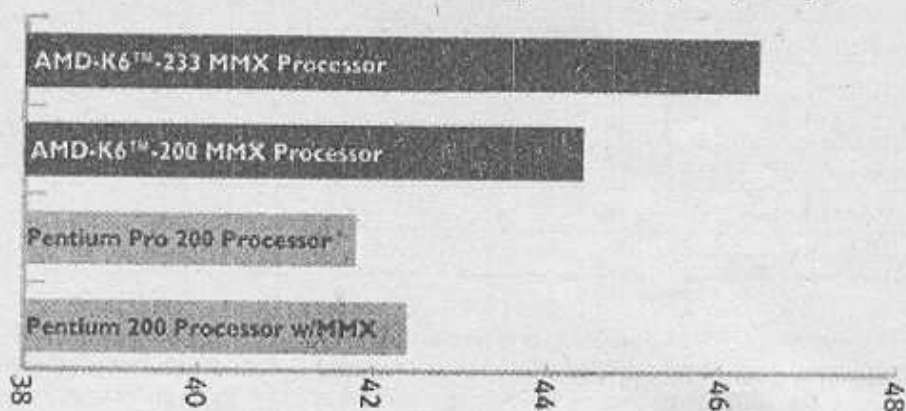
Výkonnostní testy a srovnání s konkurenčními procesory

Srovnání vlastností architektury procesorů AMD a Intel Pentium II, Intel Pentium Pro a Intel Pentium MMX.

Tabulka č. 6.18

Vlastnosti	AMD-K6	Pentium II	Pentium Pro	Pentium MMX
Superskalární architektura	X	X	X	X
RISC architektura	Ano 6 instrukcí	Ano 5 instrukcí	Ano 5 instrukcí	Ne
Spekulativní provádění	X	X	X	-
Provádění mimo pořadí	X	X	X	-
MMX	X	X	-	X
L1 instrukční a datová cache	32K + 32K	16K + 16K	8K + 8K	16K + 16K
Předvídací logika	X	X	X	X
Úspěšnost předpovědi	95%	90%	90%	75-80%
Výpočet v plovoucí řádové čáře	X	X	X	X
Typ sběrnice	Socket 7 66 MHz	Pentium Pro 66 MHz	Pentium Pro 66 MHz	Socket 7 66 MHz

Pro měření výkonu procesorů se používají uznávané benchmarkové testy od firmy ZDNet Winstone 97. Představují kolekci testů všech možných druhů softwaru kancelářského typu, který může být na těchto procesorech provozován. Testy jsou zaměřeny na výkonnost pro vstupně/výstupní operace, počítání v řádové čárce, grafické operace a další. Pro výsledky testů je vždy nezbytné uvedení sestavy, na které konkrétní testy probíhali. Pro prostředí operačního systému Windows95 byly naměřeny tyto výsledky:



Obr. č. 6.53

Z výsledků testů je patrné, že volba procesoru AMD-K6 je velice výhodná právě pro výborný výkon pro běžné kancelářské aplikace.

V dnešní době stále populárnější a náročnější 3D hry mají taky svoje benchmarky, kterými se měří rychlost počtu zobrazených snímků za sekundu (fps). Na tomto výkonu se také velkou měrou právě podílí výkon procesoru. Většinou tyto testy probíhají na grafických urychlovačích. Příkladem je grafický akcelerator firmy 3Dfx Voodoo a hra Quake2, kdy je na přesně dané mapě testována rychlost. Výsledky měření:

- AMD-K6 233 Mhz
Quake2, rozlišení 640*480 výsledek 19,8 fps
- Intel Pentium MMX 233 Mhz
Quake2, rozlišení 640*480 výsledek 23,7 fps
- Intel Pentium II 266 Mhz
Quake2, rozlišení 640*480 výsledek 39,4 fps

Z těchto výsledků je vidět, že slabší multimediální a část pro výpočet v plovoucí čárce je příčinou horších výsledků pro náročné 3D hry.

Základní deska

Typ desky:

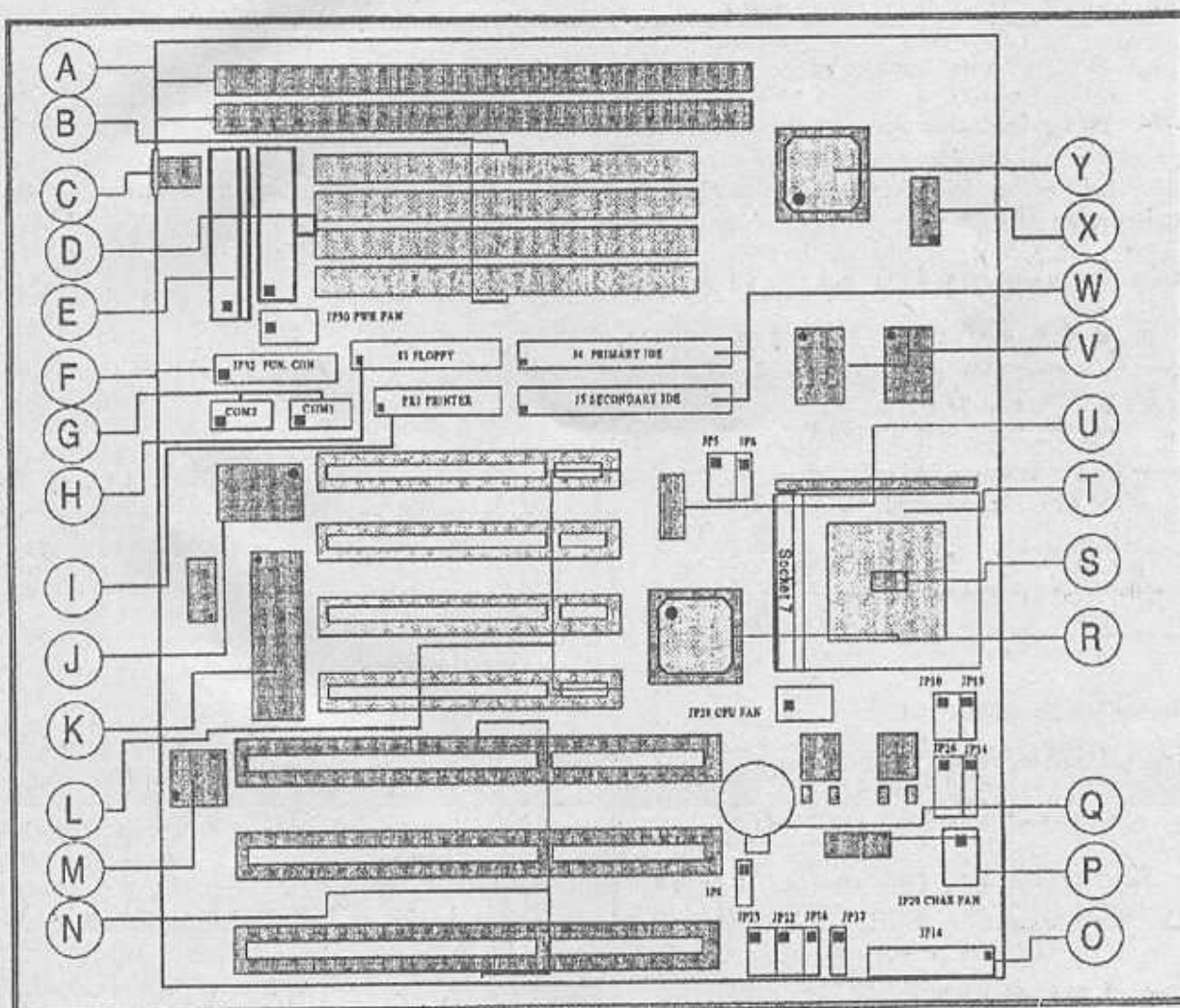
5DTX-TC série Tomato

Vlastnosti základní desky

- Podpora procesoru Intel Pentium do 233 MHz, s podporou nejnovějších procesorů MMX
- Intel 430TX PCIset, obsahující 82371AB PCI ISA IDE Akcelerator (PIIX4)
- Podpora "Auto Jumper YES" pro automatickou detekci všech typů procesorů
- Monitorování teploty CPU
- Monitorování teploty okolí, úroveň napájecího napětí a rychlosti otáčení větrání (jen lepší model)
- Při neaktivitě procesoru dojde k zastavení ventilátoru, což je signalizováno blikající LED diodou
- PnP BIOS s podporou pro power management, rozšířené IDE/SCSI a podpora desktop managementu (DMI), která umožňuje zlepšit přehled o hardwaru

- Flash BIOS s možností snadného upgradu
- 4 SIMM sokety, umožňující osazení až 256 MB DRAM paměti vždy po dvou bankách (možno osadit FPM, EDO RAM paměti)
- Dva velmi rychlé synchronní dynamické sokety (DIMM) paměti
- Tři 16-bitové ISA sloty a čtyři 32-bitové PCI sloty
- Integrovaný PCI bus master IDE kontrolér s dvěma konektory pro připojení až čtyř IDE zařízení jako Hard Disky, CD-ROM, zálohovací pásky. Podporuje "Ultra 33" mod pro přenos dat až o rychlosti 33 MB/s
- Integrovaná I/O kontrolér s jedním floppy kontrolérem, dvěma sériovými porty typu 16550 UART, jedním rychlým paralelním portem typu EPP/ECP a jedním portem pro Infra-Red přenos
- Podpora pro napájení z ATX zdroje
- Dva USB porty pro sériovou komunikaci rychlostí až 12 Mbitů/s. Zahrnuta podpora pro myš a klávesnici.

Zobrazení systémové desky



Popis jednotlivých částí systémové desky

A	3,3V DIMM sokety	N	ISA Bus konektory
B	SIMM RAM sokety	O	Konektory pro ovládací tlačítka z předního panelu
C	AT konektor klávesnice	P	Regulátor napětí
D	ATX napájecí konektor	Q	3V Lithiová baterie
E	AT napájecí konektor	R	Intel 82430TX PCIset-PIIX4
F	USB konektor	S	LM75 pro monitorování teploty CPU
G	Sériové porty (COM1 a COM2)	T	Pentium Socket 7
H	Floppy konektor	U	Systémové hodiny
I	Konektor na paralelní port	V	256/512K Cache RAM
J	Integrovaný I/O konektor	W	IDE konektory
K	Flash BIOS EEPROM	X	TAG RAM
L	PCI Bus konektory	Y	Intel 430TX PCIset system kontrolér
M	LM87 pro systémové monitorování		

Funkce "Auto Jumper YES"

Při zapojení tohoto jumperu dojde k vyřazení funkce všech přepínačů, které mají vliv na nastavení typu napájecího napětí a rychlosti procesoru z činnosti a BIOS desky se snaží detekovat vložený procesor a správně nastavit napájecí napětí a rychlost. Vše se řídí s menu BIOSu, takže odpadá nutnost při změně procesoru nastavovat jumpery pro nový typ.

Nastavení typu CPU

Tabulka č. 6.19 AMD-K5 CPU

Interní rychlost CPU (Mhz)	Externí rychlost CPU (Mhz)	JP5	JP6	Násobek frekvence	JP10
PR90	60	[3-5], [2-4]	[1-2]	1.5x	[1-3], [2-4]
PR100	66	[1-3], [2-4]	[1-2]	1.5x	[1-3], [2-4]
PR133	66	[1-3], [2-4]	[1-2]	1.5x	[1-3], [2-4]
PR150	60	[3-5], [2-4]	[1-2]	1.75x	[3-5], [4-6]
PR166	66	[1-3], [2-4]	[1-2]	1.75x	[3-5], [4-6]

Tabulka č. 6.20 AMD-K6 CPU

Interní rychlost CPU (Mhz)	Externí rychlost CPU (Mhz)	JP5	JP6	Násobek frekvence	JP10
PR166	66	[1-3], [2-4]	[1-2]	2.5x	[3-5], [4-6]
PR200	66	[1-3], [2-4]	[1-2]	3x	[3-5], [2-4]
PR233	66	[1-3], [2-4]	[1-2]	3.5x	[1-3], [2-4]

Tabulka č. 6.21 Nastavení napětí pro CPU

CPU Typ	Vyžadované napětí
Intel Pentium // Cyrix/IBM 6x86 // AMD-K5	3,4V (STD) / 3,5V (VRE)

Intel Pentium MMX	2,8V
Cyrix / IBM 6x86L / M2	2,8V
AMD-K6-PR166, PR200	2,9V
AMD-K6-PR233	3,2V

Tabulka č. 6.22

JP23	JP22	JP36	Napětí
1-3,2-4	1-3,4-6	1-2	2,8V
3-5,2-4	1-3,2-4	1-2	2,9V
1-3,2-4	3-5,4-6	1-2	3,2V
1-3,4-6	3-5,4-6	1-2	3,4V
3-5,4-6	3-5,4-6	1-2	3,5V

Budoucnost

Procesor **AMD-K6 MMX** již byl ohlášěn ve verzi postavené na 0,25mikronové technologii, což umožní snížit velikost čipu z 162 mm² na 68 mm². Také se zvýší hodinové frekvence procesoru a sníží příkon. Verze na 300 MHz bude

možné provozovat na základních deskách, které mají napájecí napětí 2,5 V. Během prvního čtvrtletí roku 1998 byla v prodeji nová verze K6 označená K6 3D. Hlavní změnou proti klasické K6 je hlavně technologie 3D, která bude pomocí nových instrukcí urychlovat 3D grafiku, audio a ostatní multimediální aplikace. Akcelerace se bude týkat především 3D her a programů využívajících Microsoft DirectX. Dále budou procesory **K6 3D** vybaveny hardwarovou podporou dekomprese videa MPEG-2 a zvuku AC-3, což jsou formáty užívané na DVD. Posílena bude i jednotka MMX, která bude plně zřetězena a zdvojnásobena, což by mělo svým výkonem překonat procesory Intel. Vnitřní sběrnice bude urychlena na 100 MHz, což urychlí průtok dat z L2 cache a na lokální sběrnici o 50%. Vše bude postaveno na patičce Socket 7, což sníží náklady na vývoj nových desek. Dalším krokem ve vývojové řadě bude **K6+ 3D**, což je opět vylepšené jádro, kterému bude na čipu přidána cache o kapacitě 256 KB, čímž se cache na základní desce stane L3. Procesor bude využívat novou patičku Super 7, což vlastně ani taková novinka není, protože půjde o patičku Socket 7 na 100 MHz s podporou AGP. Procesor bude k dispozici na 350 až 400 MHz ve druhém pololetí roku 1998. **AMD K7**, s kódovým označením **Argon** je zatím posledním projektem AMD. Procesor bude zřejmě pinově kompatibilní s Pentiem II.

6.9 Procesory firmem Cyrix a IBM

Popis procesorů firem Cyrix a IBM je uveden v [39, 43].

6.9.1 Procesor 6x86

Tento procesor byl vyvinut firmou CYRIX, je výkonově srovnatelný s procesorem Intel Pentium. Procesor se označuje jako CYRIX 6x86, který je vyráběn přímo CYRIXem a IBM 6x86 vyráběn společností IBM, dále jen 6x86. CYRIX a IBM tvrdí, že tento procesor je plně kompatibilních. Existují některé starší aplikace, které nejsou plně korektně na tomto procesoru díky spekulativnímu provádění instrukcí. Vypnutím interní cache se tento nedostatek dá odstranit.

Interní kmitočty je vynásoben 2x nebo 3x. Tento procesor je dodáván ve variantách.

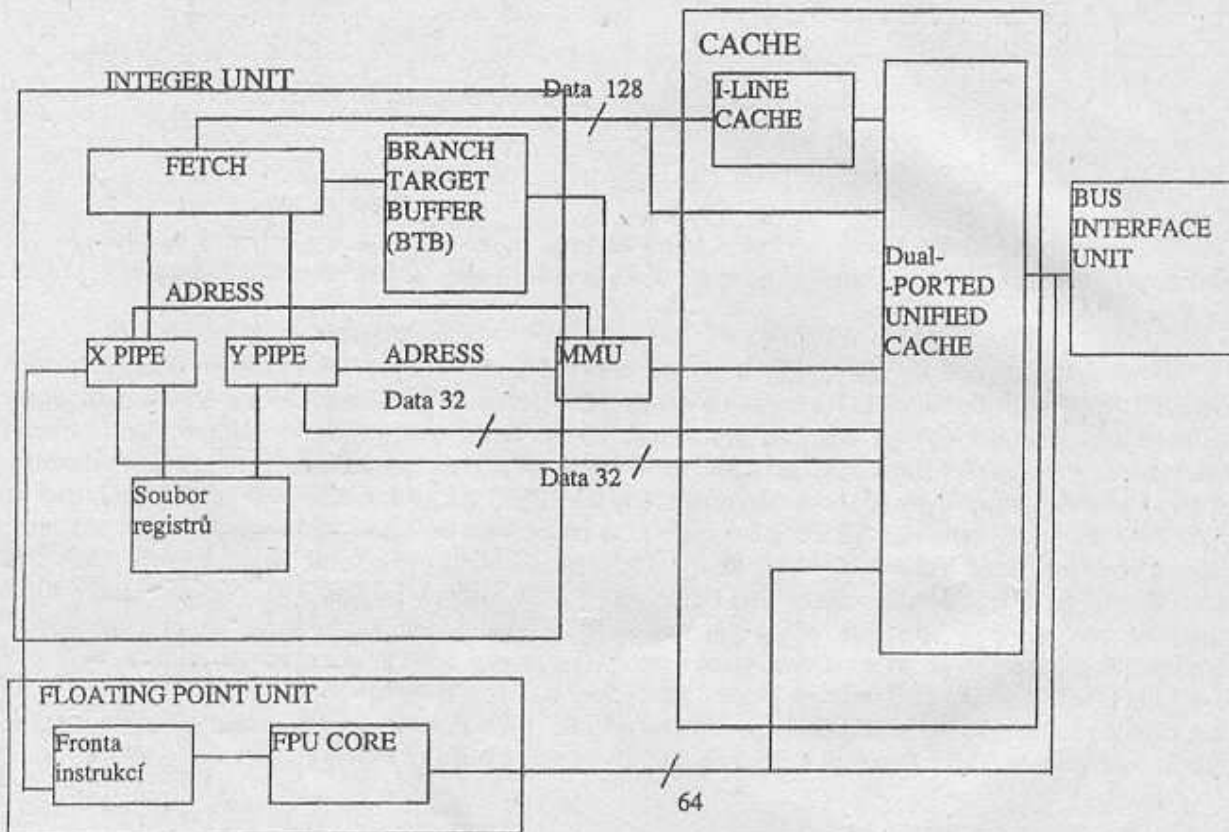
100MHz	P133+
120MHz	P150+ (srovnatelný s procesorem Intel Pentium)
133MHz	P166+
150MHz	P200+

Nevýhoda tohoto procesoru, je že potřebuje kvalitní aktivní chlazení, protože pracuje s napětím 3,2 V. Ztrátový tepelný výkon může dosahovat až 28W. Procesor je umístěn v PGA pouzdře a je kompatibilní s patičkou P54C (Socket 7), tzn. že funguje u většiny základních desek pro Pentium. U některých starších desek se musí provést update FLASH biosu, protože nekorektně detekují procesor.

Procesor s označením 6x86L má vstupní napětí separované do dvou kategorií :

- 1) CPU Core voltage : $2,8V \pm 170 \text{ mV}$
- 2) CPU IO voltage : $3,3V \pm 150 \text{ mV}$

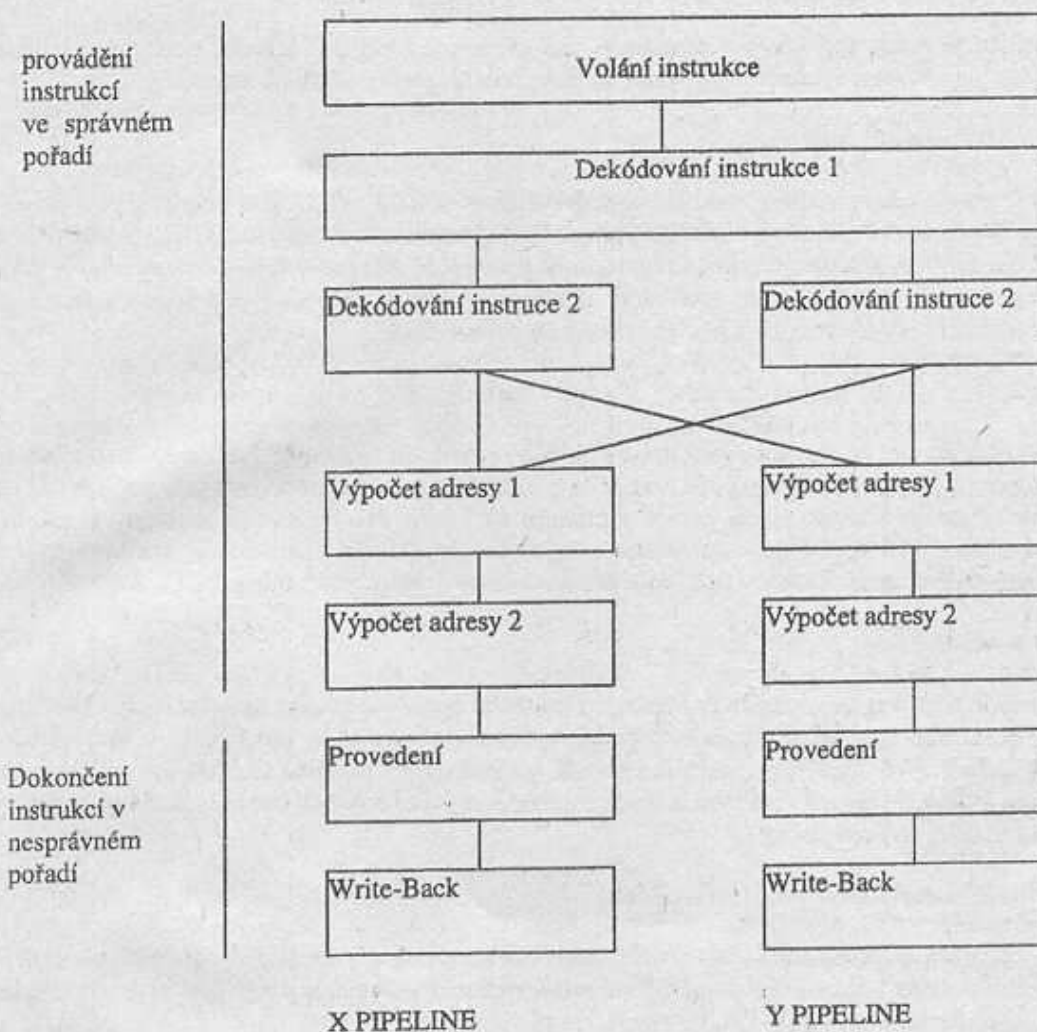
To znamená že je kompatibilní s patičkou P55C.



Obr. č. 6.53 Blokové schéma 6x86

Integer unit

Blokové schéma X a Y pipelines



Obr. č. 6.54 Integer unit.

Popis zřetězení

IU obsahuje dvě 7-stupňové celočíselné zřetězení, pojmenované X a Y pipelines a jsou schopny paralelně provádět 2 instrukce. Jednotlivé stavy jsou:

prefetch - volání instrukcí (PF),
dekódování instrukcí 1 (ID1),
dekódování instrukcí 2 (ID2),
výpočet adresy 1 (AC1),
výpočet adresy 2 (AC2),
provádění (EX),
write-back (WB).

– PF - volání instrukce

PF je společný pro X a Y pipeline. Během tohoto stupně je 16 bytů kódu vykonáno za jeden takt z paměti subsystému. Dále tok kódu je kontrolován ke zjištění výskytu instrukcí, které modifikují normální sekvenční průběh programu. Tyto instrukce se vztahují k větvení instrukcí.

Existují dva typy větvení instrukcí:

- 1) nepřímé,
- 2) přímé.

Jestliže je jeden typ větvení detekován, tak předpověď větvení logicky poskytne předpověděnou cílovou adresu pro větvení. Prefetch stupeň pak začíná přenášet předpověděnou adresu.

– ID - dekódování instrukce

ID stupeň je superpipelined a skládá se ze dvou substupňů ID1 a ID2. ID1 ohodnotí poskytnutí pomocí prefetch stupeňů a rozhodne počet bytů instrukce pro provedení dvou instrukcí na clock. ID2 stupeň pak dekóduje dvě instrukce a zvolí buď X nebo Y pipeline pro další provádění. Zavedení seřazení algoritmu je použito pro zvolenou pipeline. Tento algoritmus rozhodne, u které pipeline je nejméně pravděpodobné na zpoždění dokončení instrukcí díky interakcím s předchozí odeslanou instrukcí.

– AC 1,2

Výpočet adresy je také superpipelined a skládá se ze dvou substupňů AC1 a AC2. Jestliže současná instrukce požaduje paměťové operandy, AC1 spočítá dvě lineární paměťové adresy na clock (jeden na pipeline), AC2 provede přiřazení činnosti řízení paměti a přístupu ke cache. Pro registrové operandy se přejmenování registru stane během AC1 a AC2, pak se přistoupí k souboru registrů. Další floating point instrukce jsou odeslány do FPU během AC2 stupně. Všechny instrukce jsou potlačeny v programu během AC1 a AC2.

– Prováděcí

Prováděcí stupeň provádí operace instrukcí použitím operandů poskytnutým stupněm výpočtu adresy. Výsledky operací jsou zapsány do souboru registrů, vyrovnávací paměť se zapíše během Write-back stupně. Jakmile jsou instrukce vloženy do prováděcího stupně, tak instrukce v pipeline se může vykonávat nezávisle na druhé pipeline. V jiných slovech se mohou instrukce vykonávat v jiném pořadí než byly poslány, avšak výsledky jsou na výstupu ve správném pořadí.

Optimalizované využití zřetězení

Procesorová architektura 6x86 je optimalizována paralelně užitím X a Y pipeline dovolující většinu instrukcí odesílat v párech a dovoluje poměrně nezávislé zpracování dvou pipeline. Tato technika maximalizuje výkon redukcí počtu hodinových cyklů, ve kterých je pipeline nečinný.

– Posílání -Instruction dispatch

Běžné instrukce v x86 mohou být poslány v páru (z ID2) do obou pipeline, aniž by měli ohled na závislost, která je mezi instrukcemi. Instrukcí, které nelze párovat, je velice málo. Existují tři kategorie instrukcí, které musí být poslány pouze do X pipeline:

- větvící instrukce
- instrukce, které operují s plovoucí čárkou
- exklusivní instrukce

První dva (X-pipeline pouze) typy mohou být párovány s dalšími instrukcemi v Y pipe. Exklusivní instrukce nemohou být párovány. Instrukce jsou klasifikovány jako exklusivní, jestliže způsobí v EX stupni (u pipeline) chybu (požadují většinou několikanásobný přístup do paměti. Ačkoli exklusivní instrukce nemohou být párovány, prostředky z obou pipeline jsou použity na dokončení instrukce. Exklusivní instrukce jsou vypsány níže:

- nastavení segmentů v protected mode přístup do speciálních registrů (řídicí, ladící, testovací)
- řetězcové instrukce

- násobení a dělení
- přístup do I/O portů
- PUSHA a POPA (uložení a načtení všech registrů na zásobník)
- přepínání tásků

- Dokončení instrukce v jiném pořadí

Dokončení instrukcí v jiném pořadí se vyskytuje u EX a WB stupně, když je instrukce dříve dokončena než předcházející poslaná instrukce ve vedlejší pipeline, která požaduje mnohonásobně více hodinových taktů k dokončení. Tento typ zpracování je v prvořadě použit, když instrukce čeká na přístup do paměti. Současně následující instrukce v EX stavu vedlejší pipeline může být provedena bez čekání na nevyřízený přístup k dokončení.

Samozřejmě je to možné pouze u nezávislých instrukcí. 6x86 vždy provádí instrukce ve správném pořadí, až do EX stavu. Dovoluje instrukci dokončit v jiném pořadí pouze od tohoto místa. Také zapsání výsledku instrukcí se provede ve správném pořadí, to zajišťuje kompatibilitu x86.

- Odstranění datové závislosti

Do procesoru je začleněna architektura klíčů, která eliminuje nečinné stupně pipeline z vnitřní datové závislosti instrukcí. Kombinace technik: přejmenování registrů, urychlení dat, vynechání dat jsou použity k eliminaci:

- Zápis po zápisu (WAW - write after write)
- Zápis po čtení (WAR)
- Čtení po zápisu (RAW)

- Přejmenování registrů

Procesor obsahuje 32 hlavních fyzických registrů. Tyto registry jsou mapovány nebo přejmenovány do 8 logických registrů x86 (EAX, EBX, ECX, EDX, ESI, EDI, EBP, ESP). Toto přejmenování je zcela řízeno hardwarově. Při každém výskytu zápisu do logického registru je k němu přidělen nový fyzický. Toto zabrání přepsání předchozích dat v logickém registru a také odstraní WAW a WAR závislost viz. následující příklad.

Odstranění WAR

tyto instrukce jsou prováděny současně v X a Y pipeline

- (1) MOV BX,AX
- (2) ADD AX,CX

X PIPE	Y PIPE
(1) BX ← AX	(2) AX ← AX+CX

WAR závislost existuje u AX registru, protože Z pipeline musí čekat na X pipeline pro čtení AX před součtem instrukcí v Y. Toto způsobí, že Y pipeline zůstává stát, když se nepoužívá přejmenování a dokončení v nepořádku je dovoleno. U 6x86 a 6x86L se provede substituce fyzického registru za logický. Tato operace je provedena paralelně a Y pipeline nestojí jako bez přejmenování.

Počáteční přiřazení:

AX = reg0
 BX = reg1
 CX = reg2

X PIPE	Y PIPE
(1) reg3 ← g0	(2) reg4 ← reg0 + reg1

Konečné přiřazení:

AX = reg4

BX = reg3

CX = reg2

Odstranění WAW

tyto instrukce jsou prováděny současně v X a Y pipelines

(1) MOV AX,[mem]

(2) ADD AX, BX

X PIPE Y PIPE

(1) AX ← mem (2) AX ← AX + BX

Použitím techniky Data Forwarding, je paměťový operand přístupný pro obě pipeline současně. Tím pádem nemusí Y-PIPE čekat na X-PIPE.

Počáteční přiřazení:

AX = reg0 BX = reg1

(1) reg2 ← mem (2) reg3 ← mem + reg1

Koncové přiřazení:

AX = reg3

BX = reg1

- Data Forwarding

Odstraňuje závislost RAW registrů a paměti. Tím umožňuje dvěma instrukcím zároveň přistupovat k paměťové buňce. 6x86 implementuje dva typy data forwarding: operand forwarding a result forwarding.

Operand forwarding se vyskytuje u instrukce MOV při načtení dat do registru nebo paměťové buňky. Tyto registry nebo paměťové buňky jsou použity v následující instrukci jako operand, tím se ztváří RAW závislost. Použitím operand forwarding se data načtou do druhé instrukce, aniž by čekali na ukončení MOV (1. Instrukce)

Result forwarding se vyskytuje například, kdy provádíme nějakou operaci a následující instrukce manipuluje s výsledkem operace. Výsledek se současně uloží paralelně do obou operandů.

Příklad:

ADD AX,BX ;sčítání

MOV [mem],AX ;hodnota součtu se paralelně zapíše do AX a paměťové buňky mem

- Obcházení dat

Tato technika se používá v případě, kdy v 1. instrukci provádíme nějakou operaci a výsledek ukládáme do paměti, následující instrukce používá výsledek předešlé instrukce. Tím vzniká RAW závislost. Obcházení dat umožní okamžitý přístup do výsledku operace (fyzický registr).

Příklad:

ADD [mem], AX

SUB BX, [mem]

X PIPELINE

Y PIPELINE

mem ← mem + reg0 (2) reg2 ← reg1 - (mem + reg0)

Větvící instrukce

Větvící instrukce se přibližně objeví každou 4. až 5. instrukci u architektury x86. Větvící instrukce změni normální sekvenční běh programu. To může způsobit pipeline stupňům čekání, než CPU vytvoří instrukční proud. 6x86 minimalizuje tyto vlivy předpovědí větvení a spekulativním vykonáváním .

Předpověď větvení: 6x86 používá větvící cílovou vyrovnávací paměť (BTB - branch target buffer) k uložení větví cílových adres a větví předpověděných adres. Během pefetch stupně je instrukční tok kontrolován pro výskyt větvících instrukcí. Jestliže je nalezen skok bez podmínky, CPU se pokusí nalézt cílovou adresu v BTB. Pokud ji nalezne, tak procesor začne načítat v cílové adrese určené v BTB.

V případě podmínečného větvení BTB poskytuje také adresu, která je pravděpodobnější. Rozhodnutí pro načtení cílové adresy je založeno na čtyřstupňovém algoritmu předpovědi, který dosahuje přibližně 90 % úspěšnosti.

Větvící instrukce se posílá do X pipeline, kde v EX stupni čeká, než je dokončena instrukce, která určuje podmínku větvení. V případě, že adresa odpovídá předpověděné, tak se větev provede během jednoho taktu. Jinak se musí vyprázdnit X pipeline a načíst ze správné adresy.

Spekulativní provádění: CPU je schopen spekulativně provádět další předpověděné větvení nebo FPU instrukce. Spekulativní provádění dovoluje spouštění instrukcí v další větví bez čekání na druhou pipeline. Ten samý mechanismus je použit na spouštění FPU instrukcí paralelně s IU instrukcemi. Procesor má 4 úrovně spekulací pro generování načítací adresy. Používá se kontrolní bod zdroje, který uchovává registry, flags a stav prostředí procesoru. Při další větví se zvýší čítač spekulativní úrovně a začíná provádět operace na předpověděném instrukčním toku. Jakmile jsou větvící instrukce rozhodnuté, CPU sníží spekulativní složitost. Pro správně předpověděnou větev je stav kontrolního bodu zdroje vymazán. Při špatném odhadu větvení CPU vygeneruje správnou načítací adresu a použije hodnotu kontrolního bodu zdroje na znovuoobnovení stavu procesoru v jednom taktu. Pro kompatibilitu není dovolen zápis spekulativních výsledků do cache, dokud není větvící adresa vypočtena. Spekulativní provádění pokračuje, dokud není splněna jedna z následujících podmínek:

- 1) větvící a FPU instrukce je dekodována a spekulativní úroveň má hodnotu 4,
- 2) výjimka, nebo nastala chyba,
- 3) zápisový buffer je plný,
- 4) pokus o modifikaci zdroje, který není uložen v kontrolním bodu (segmentový registr, systémový registr).

Jednotka pro operace v plovoucí čárce

Je vzájemně propojena s celočíselnou jednotkou a s cache přes 64 bitovou sběrnici. FPU je kompatibilní s instrukcemi x87, to znamená také s IEEE-754 standartem. Protože se instrukce FPU a IU vzájemně doplňují, tak tyto jednotky pracují paralelně.

FPU paralelní provádění:

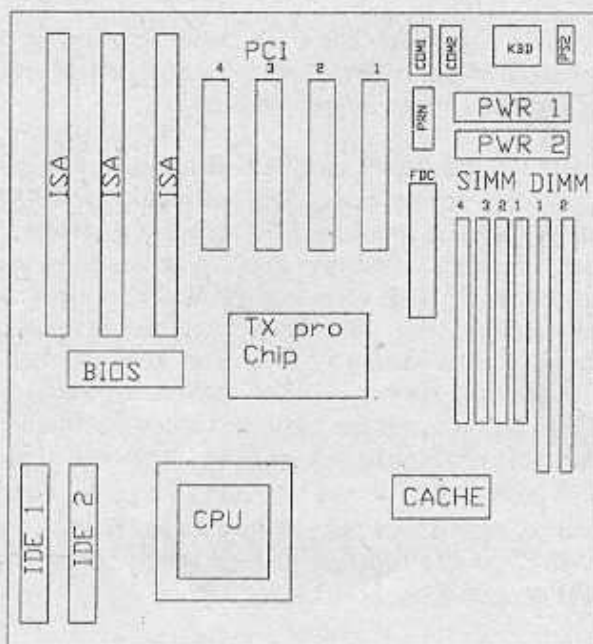
6x86 provádí instrukce FPU a IU paralelně. Celočíselné instrukce mohou být dokončeny v jiném pořadí vzhledem k FPU instrukcím. Procesor si zajišťuje kompatibilitu s x86 signalizováním výjimek a zápisovými cykly v programovém pořadí. FPU instrukce jsou pouze posílány do X pipeline u IU. Stupeň výpočtu adresy X pipeline kontroluje u správy paměti výjimky a přístup k paměťovým operandům FPU. V případě, že výjimka nebyla detekována, posílají se FPU instrukce do FPU instrukční fronty (během AC2 stupně). Procesor může následující celočíselnou instrukci dokončit spekulativně, v nepořádku vzhledem k FPU instrukci a vzhledem k FPU výjimce, která se může přihodit. Procesor je schopen posílat 4 FPU instrukce do FPU instrukční fronty. Jakmile se dokončí FPU instrukce, spekulativní složitost se sníží a kontrolní bod zdroje je volný pro znovupoužití v následující operaci. FPU používá také 4 zápisové buffery, které zabraňují čekání během spekulativního zápisu.

Cache

Procesor obsahuje dvoubránovou cache pro instrukce a data, plně asociativní 256 byte dlouhou instrukční řádku (I-Line), která obsahuje nejvíce používaná paměťová místa. Sjedenocaná cache je primárně datová a sekundárně instrukční. I-Line cache je primárně instrukční. I-Line cache používá LRU algoritmus pro vyjmutí řádků. Sjedenocaná cache umožňuje paralelně provádět dvě operace z těchto:

- načítání kódu
- čtení dat (X-pipeline, Y-pipeline nebo FPU)
- zápis dat (X-pipeline, Y-pipeline nebo FPU)

Příklad základní desky, obr. č. 6.55.



Obr. č. 6.55

Většina dnešních základních desek pro procesor 6x86, Pentium, K5, K6, M2 má:

- sloty sběrnice ISA
- sloty sběrnice PCI
- sloty SIMM 72 pinové - každý má 32bitovou sběrnici, používají se párovane 1,2 a 3,4. Velikost jednoho SIMMu je 4, 16, 36, 64, 128 MB a jsou typu DRAM, EDO DRAM, obvykle mají přístupovou dobu 60 ns.
- sloty DIMM jsou modernější, vytlačují klasické SIMM, 64 bitová datová sběrnice, jsou typu EDO DRAM, SDRAM (10-15ns)
- Řadič: obsahuje dva kanály EIDE po dvou discích master-slave popř. CD-ROM a jiné diskové zařízení, FDC - diskové mechaniky 3,5" ;5,25". COM1 a COM2 - sériové porty, PRN - paralelní port (LPT1). Řadič je připojen na PCI.
- Socket 7 pro P54C nebo P55C procesory
- Cache: 512 kB synchronní SRAM (některé novější mají 1024 kB)
- BIOS: většinou AWARD nebo AMI, jsou srovnatelné
- USB: Universal Serial Bus - většinou dnes nemá využití, navržen pro budoucí použití např. u nastavení parametrů monitorů

- PS/2: konektor pro myš
- PWR1: klasický konektor pro napájení 12V a 5V
- PWR2: novější napájení ATX 3,3V;5V;12V
- KBD: klávesnice
- Přepínače pro nastavení napájení procesoru a taktovací frekvence
- Čipové sady jsou firmy Intel -VX, TX, VX-pro, TX-pro a F.I.C VIA APOLLO 1-3

6.9.2 Procesor Cyrix/IBM 6x86MX (M2)

Tato část se zabývá procesorem, který byl navrhnout firmou CYRIX Corp. a nazývá se **Cyrix 6x86MX**, dříve označený kódovým jménem **M2**. Z důvodu nedostatečných technologických a především výrobních kapacit se k firmě Cyrix připojila firma **IBM Microelectronics**. Ta začala tento čip vyrábět a prodávat pod vlastní značkou a pojmenovala jej **IBM 6x86MX**. Poněvadž byl k dispozici procesor firmy IBM, budou se informace zde uvedené většinou týkat mikroprocesoru právě této značky.

Charakteristika:

Procesor 6x86MX je založen na technologii procesoru 6x86 (viz IBM/Cyrix 6x86) a stejně jako jeho konkurenti Pentium II(Intel) a K6(Advanced Micro Devices, dále jen AMD), má multimediální rozšíření MMX. Stejně jako firma AMD se i společnost Cyrix se nesnaží prosadit nové standardy a její procesory MX lze instalovat do patice Socket 7. Dalšími hlavními atributy jsou superzřetězení (superpipelining), předpovídání skoků, spekulativní provádění, data forwarding a vykonávání instrukcí mimo pořadí. Procesor také obsahuje 64 kB unifikované L1 cache. Čip je umístěn v keramickém pouzdru CPGA a používá zdvojené napájení. 6x86MX je vyráběn ve verzích PR166, PR200, PR233, PR266 a PR300, za použití technologie 0,35 mm.

Popis atributů:

- Multimediální rozšíření MMX (Multi Media eXtension)
Sada 57 nových instrukcí, urychlující multimediální operace (grafika, video, audio). Společnost Cyrix na rozdíl od AMD, která technologii MMX odkoupila od firmy Intel, vyvinula technologii vlastní, kompatibilní s Intel MMX. Při použití softwaru podporující MMX se udává zrychlení 50 - 200 % oproti procesoru běžící na stejné frekvenci bez MMX instrukcí.
- Patice Socket 7
321 pinová patice pro procesory P54C (Intel Pentium, Cyrix/IBM 6x86 a K5 firmy AMD) a P55C (Intel Pentium MMX, Cyrix/IBM 6x86MX, 6x86L a AMD K6), typu ZIF (Zero Insert Force), nebo-li patice s možností snadné výměny procesoru.
- First Level Cache (L1 cache)
Cache je rychlá vyrovnávací paměť, která obsahuje většinu naposledy používaných dat a instrukcí, které budou zpracovávány různými exekutivními jednotkami. Cache první úrovně je cache umístěná přímo na čipu. U procesoru 6x86MX má velikost 64 kB a je rozdělena do dvou částí, na datovou a instrukční cache.
- Pouzdro CPGA
Pouzdro, ve kterém je samotný čip umístěn se nazývá Ceramic Pin Grip Array (keramické pinové síťové pole). Tento standardní "obal" má 291 pinů a rozměry 50 x 50 mm. Toto keramické pouzdro je integrováno s měď-wolframovou slitinou pro kvalitnější odvod tepla.
- Duální napájení
Jak již bylo uvedeno, 6x86MX je typu P55C, a používá tedy tzv. duální napájení. Toto napájení definuje napětí 2,8 V pro jádro procesoru (core) a 3,3 V pro vstupní/výstupní obvody CPU (I/O). Tento nový standard, poprvé použitý u procesoru Intel Pentium MMX, byl zaveden hlavně kvůli snížení množství vyzařovaného tepla.
- Verze 6x86MX
Firmy IBM a Cyrix uvádějí u svých procesorů tzv. Performance Rating (PR), který udává, jaký Intelovský procesor je s ním výkonově srovnatelný (stejně rychlý nebo rychlejší). U čipů 6x86 Performance Rating udával srovnatelné Intel Pentium a u 6x86MX Intel Pentium II. Toto bylo zavedeno

společně s firmou AMD (platí pouze pro K5) pro snadnější orientaci v procesorech, protože 6x86 a 6x86MX jsou na stejné frekvenci podstatně rychlejší než konkurenční Intel Pentium a Pentium II. Např. IBM 6x86MX PR233 má být minimálně stejně rychlé jako Intel Pentium II 233 MHz, i když pracuje na frekvenci 188 MHz (viz tabulka č. 6.22). Pro toto porovnání byly oběma firmami vybrány testy, které ukazují výkon procesoru v reálných komerčních a kancelářských aplikacích pod operačním systémem Windows 95.

Tabulka č. 6.22

CPU	Performance Rating	Frekvence sběrnice	Multiplikátor	Interní frekvence
IBM 6x86MX	PR166	66 MHz	2 x	133 MHz
IBM 6x86MX	PR200	66 MHz	2,5 x	166 MHz
IBM 6x86MX	PR233	75 MHz	2,5 x	188 MHz
IBM 6x86MX	PR266	75 MHz	3 x	225 MHz
IBM 6x86MX	PR300	83 MHz	3 x	250 MHz

Tabulka č. 6.23

Atributes	IBM 6x86MX	IBM 6x86
Pinout	P55C	P54C
Supply Voltage	2,8 V Core 3,3 V I/O	6x86: 3.3 nebo 3.52 V 6x86L: 2.8 V Core, 3.3 V I/O
CPU Primary Cache	64 kB	16 kB
Translation L. Bufffer (TLB)	L1: 16 entry L2: 384 entry	L1: 128 entry Victim TLB: 8 entry
Branch Prediction	512 entry branch target cache 1024 entry branch history table	256 entry branch target cache 512 entry branch history table
MMX	Yes	No
Performance Monitor including Time Stamp Counter and Model Specific Registers	Yes	No
Scratchpad RAM in Primary Cache	Yes	No
Cacheable SMI Code/Data	Yes	No
Clock Modes	2x, 2.5x, 3x, 3.5x	2x, 3x

Hlavní funkční bloky

Procesor 6x86MX se skládá ze 4 hlavních funkčních bloků:

- Memory Managment Unit, MMU (Jednotka pro správu paměti)
- CPU Core (Jádro CPU)
- Cache Unit, CU (Jednotka vyrovnávací paměti)
- Bus Interface Unit, BIU (Jednotka rozhraní sběrnice).

CPU obsahuje superpipelined Integer Unit (IU), Branch Target Buffer (BTB, Paměť adres skoků) a Floating Point Unit (FPU).

BIU představuje rozhraní mezi externí systémovou deskou a vnitřními exekutivními jednotkami procesoru. Při práci s pamětí je pozice v paměti udávána přes Address lines (adresní linky) a data jsou z/do paměti přesouvána po Data lines (datové linky). Každá instrukce je načtena do 256 Bytové Instrukční line cache. CU shromažďuje nejvíce používaná data a instrukce, aby FPU a IU umožnila rychlý přístup k informacím. Když si pak jádro CPU vyžádá instrukci z CU, jsou obdržené instrukce dekódovány jednou ze dvou (X nebo Y) processing pipelines (pracovní linky) v IU. Pokud je to instrukce MMX nebo instrukce FPU, je zaslána do FPU pro zpracování.

Jestliže požadována data nejsou v cache, pak se k nim přistupuje přes BIU do hlavní paměti. Jednotka pro správu paměti vypočítává fyzickou adresu, včetně adresace založené na stránkování. Fyzická adresa je vypočtena MMU a předána do CU a BIU.

Integer Unit

IU obstarává paralelní instrukční provádění, přičemž používá dvě sedmi-stavové integer pipelines (linky). Každá z těchto dvou linek X a Y může provádět několik instrukcí současně.

- **Out-of-Order Processing (Dokončení instrukcí mimo pořadí)**
- **Pipeline Selection (Výběr pracovní linky)**
- **Data Dependency Solutions (Řešení datových závislostí)**

- Register Renaming (Přejmenovávání registrů)
- Data forwarding (Přesun dat)
- Data Bypassing (Obcházení dat)

- **Branch Control (Řízení větvení)**

Větvící (skoková) instrukce se objeví přibližně každou 4. až 6. instrukci u architektury x86. Skoková instrukce změní normální sekvenční běh programu. To může způsobit pipeline stupňům čekání, než CPU vytvoří instrukční proud. 6x86MX minimalizuje tyto vlivy (stejně jako 6x86) předpovědí skoků a spekulativním vykonáváním

- **Branch Prediction (Předpovídání skoků)**

Algoritmus dynamického předvídání skoků, je založen na jednoduché myšlence - je-li v programu podmíněný skok, bylo by dobré odhadnout, jaký bude výsledek podmínky ještě před tím, než se vyhodnotí, aby byly ve vyrovnávací paměti instrukce připravené na zpracování bez zbytečné prodlevy. Technika je opět stejná jako u 6x86, ale jsou rozdílné parametry Branch Target Bufferu, který má 512 položek a tabulka posledních skoků (Branch target history table) maximálně 1024 položek.

- **Speculative Execution (Spekulativní provádění)**

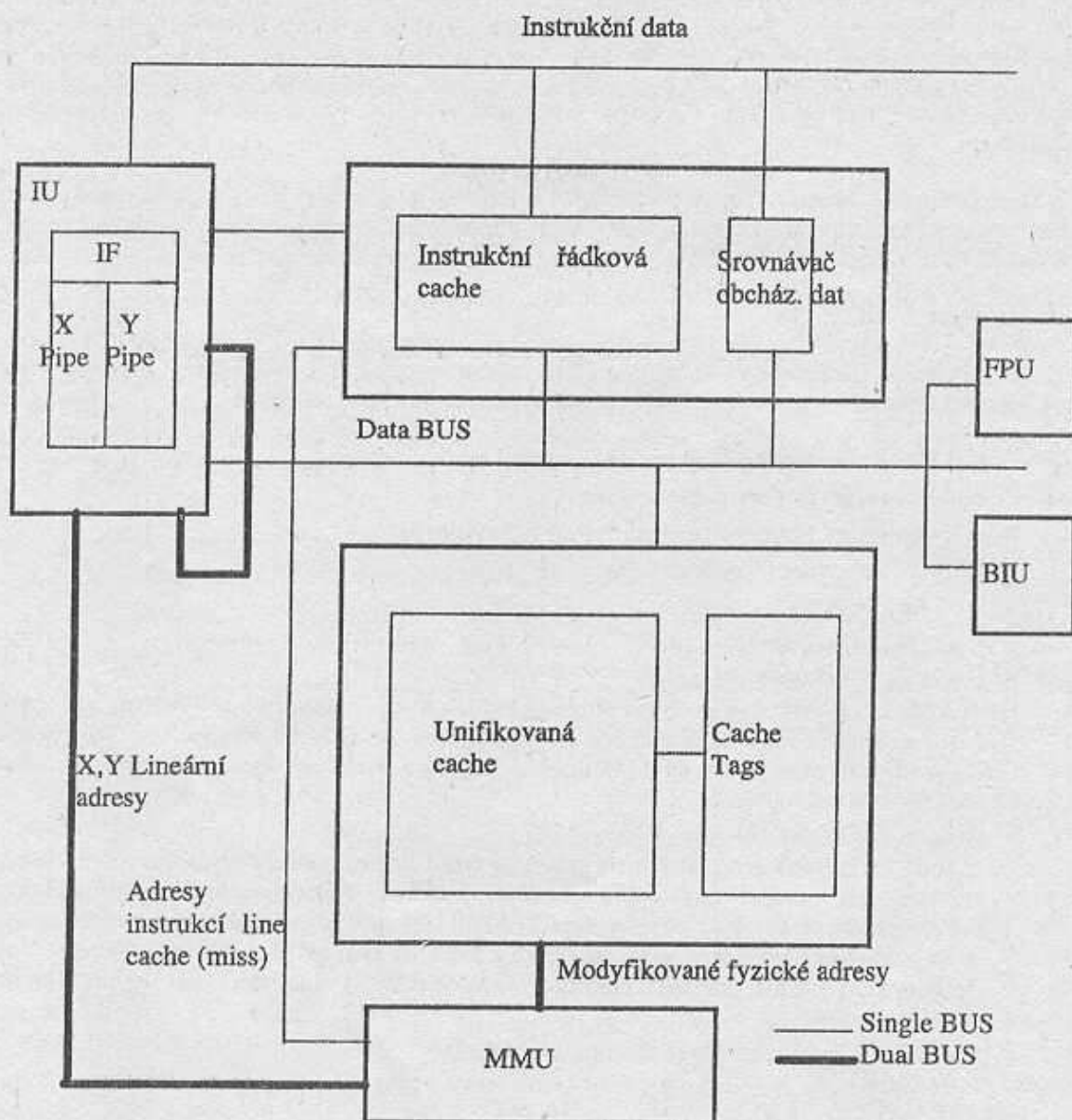
Cache Units

CPU 6x86MX používá dvě cache, unifikovanou a tzv. instrukční řádku.

Hlavní cache je 4-cestná, 64 kB velká set-associative unifikovaná cache. Unifikovaná cache zajišťuje vysoké procento zásahů (hits) tím, že je dynamicky (podle potřeby) rozdělována na datovou a instrukční cache.

Instrukční řádková cache je plně asociativní 256-Bytová cache. Tato část vyrovnávací paměti se zabráňuje nadměrným konfliktům mezi kódovým a datovým přístupem do unifikované cache.

– Blokové schéma a operace Cache Unit



Obr. č. 6.56

– Unified Cache (Unifikovaná cache)

64 kBytová unifikovaná write-back cache pracuje jako primární datová cache a jako sekundární instrukční cache. Je konfigurována jako 4-cestná asociativní a může shromažďovat až 64 kB kódu a dat ve 2048 řádcích. Cache je rozdělená a umožňuje jakýmkoliv dvou z následujících operací, aby proběhly paralelně:

- Code fetch (vykonání kódu)
- Data read (X pipe, Y pipeline nebo FPU)
- Data write (X pipe, Y pipeline nebo FPU)

Unifikovaná cache používá pseudo-LRU výměnný algoritmus a může být konfigurována pro alokování nových řádků jen při neúspěšném čtení z cache (miss) nebo při neúspěšném čtení a zápisu.

- Instruction Line Cache (Instrukční řádková cache)

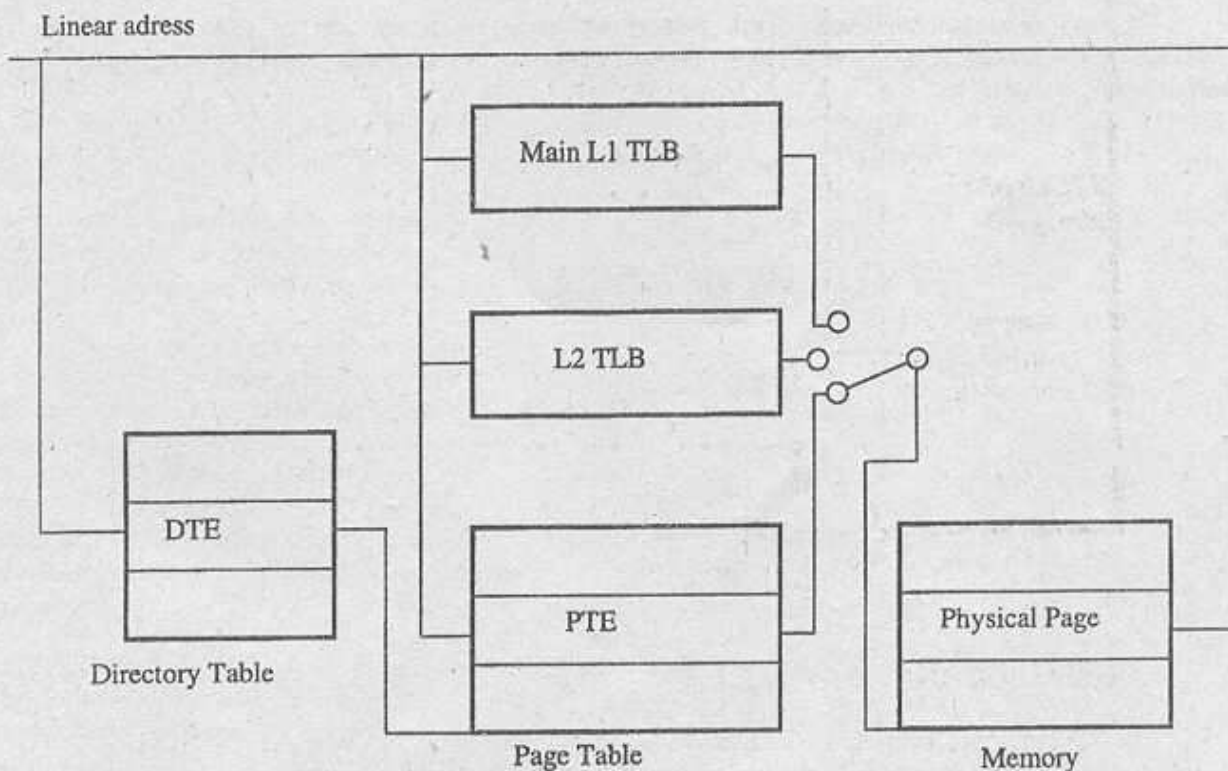
Plně asociativní 256 Bytová instrukční řádková cache je používána jako primární instrukční cache. Tato cache je plněna z unifikované cache přes Data BUS (Datová sběrnice). Požadavky z IU, které jsou nalezeny v instrukční line cache (hits) znamenají, že není již třeba přistupovat do unifikované cache. Jestliže se tam ale nenacházejí (miss), instrukční line data jsou z unifikované cache přesunuta současně do instrukční line cache a Integer Unit.

Instruction Line Cache taktéž používá pseudo-LRU výměnný algoritmus. Pro zajištění řádné operace v případě vlastní změny kódu jsou všechny zápisy do unifikované cache testovány, zda se již tyto data nenacházejí v instruction line cache. V případě úspěchu (hit) je příslušná řádka zrušena.

Memory Management Unit (Jednotka správy paměti)

Memory Management Unit (MMU) překládá lineární adresu, získanou od IU, na fyzickou adresu, která je pak použita v unifikované cache a Bus Interface Unit (BIU). Funkce MMU jsou x86 kompatibilní, držící se standardního stránkovacího mechanismu. V procesoru M2 se nacházejí dva TLB (Translation Lookaside Buffer - Překladový buffer), hlavní L1 TLB a větší L2 TLB. L1 TLB je 16-položkový a je přímo mapován, obsahuje 42 řádků. L2 TLB, obsahující 384 položek, je 6-cestný asociativní a obsahuje 384 řádků. DTE (Directory Table - adresářová tabulka) je umístěn v paměti.

Schéma MMU



Obr. č. 6.57

- Floating Point Unit (Jednotka pro práci s čísly s pohyblivou desetinnou čárkou)

Floating Point Unit (FPU) procesoru 6x86MX zajišťuje zpracovávání floatong point a MMX instrukce. FPU vytváří rozhraní mezi Integer Unit a Cache Unit přes 64-bitovou sběrnice. M2 FPU je kompatibilní s instrukční sadou x87 a drží se standardu IEEE-754. Protože většina aplikací obsahuje FPU instrukce smíchané s instrukcemi integer, procesor M2 dosahuje vysokého výkonu tím, že dokončuje FPU a integer operace paralelně.

- FPU Parallel Execution (Paralelní vykonávání v FPU)

6x86MX dokončuje FPU a integer instrukce paralelně. Integer instrukce mohou být dokončeny v jiném pořadí s ohledem na FPU instrukce. Procesor dodržuje kompatibilitu s x86 tím, že signalizuje výjimky a provádí zápisové cykly v řádném pořadí.

Jak již bylo výše uvedeno, FPU instrukce jsou vždy zasílány do X pipeline v IU. Stav (stupeň), kdy se vypočítává adresa, testuje, zda nedošlo v MMU k výjimce a přistupuje k paměťovým operandům přes FPU. Jestliže není detekována žádná výjimka, CPU zkontroluje stav procesoru a ve stupni AC2 odešle floating point instrukce do FPU instrukční fronty. 6x86MX pak může dokončit následující Integer instrukci spekulativně a mimo pořadí vzhledem k FPU instrukci a relativně k potenciální FPU výjimce, která by mohla nastat.

Když jsou další FPU instrukce vloženy do pipeline, M2 může do FPU instrukční fronty odeslat až 4 FPU instrukce. Procesor pak dál může provádět instrukce spekulativně mimo pořadí vzhledem k FPU frontě, dokud procesor nenarazí na situaci, která způsobí zastavení spekulativního provádění.

M2 FPU taky používá sadu 6 zápisových bufferů proti zahlcení při spekulativních zápisech.

Bus Interface Unit (Jednotka rozhraní sběrnice)

Bus Interface Unit (BIU) obstarává signály a časování vyžadované externím systémem.

Základní deska pro CPU 6x86MX

Co se týče základních desek a jejich podpory tohoto procesoru, tak snad jen to, že tento procesor lze zasunout do většiny desek, podporující 6x86 (včetně chipsetů). Samozřejmě je nutnost podpory BIOSu a duálního napájení (P55C).

7. MATEMATICKÉ KOPROCESORY.

Univerzálnost mikroprocesoru [2,19] spočívá v tom, že pro ně můžeme psát programy pro velice různorodé úlohy. Tato zásadní možnost neznamena vždy, že jde o řešení optimální nebo přijatelné. Existuje mnoho úloh, jejichž realizace univerzálními mikroprocesory je příliš pomalá.

Příklad:

1. Výpočet transcendentní funkce $tg(x)$ ve formátu s pohyblivou řádovou čárkou.
2. Převzetí bloku dat z periferního zařízení spojené s ověřením přenosu podle nějakého protokolu.

Takové příklady navozují myšlenku uvolnit univerzální mikroprocesor od některých úkolů a předat je tzv. **koprocesorům**, tj. spolupracujícím specializovaným mikroprocesorům spojeným s CPU.

Jako koprocesory pro numerické výpočty se zpočátku používaly obvody kapesních kalkulatorů. Pak vznikly numerické koprocesory APU 8231 a FPU 8232 pro 8bitové mikroprocesory, které pracují až 50krát rychleji než CPU. Jejich vysoké rychlosti se však málokdy plně využije. Způsobují to časové ztráty při předávání instrukcí, operandů i výsledků mezi koprocesorem a CPU. Mezi výhody patří nepříliš velká závislost na specifickém CPU, jak je tomu u pozdějších koprocesorů.

Časové ztráty při styku s CPU byly výrazně potlačeny u novějších obvodů. Patří sem typy 8087 a 8089 a na ně navazující koprocesory 80287 a 80387, určené k pokročilejším univerzálním mikroprocesorům 80286 a 80386.

7.1 Numerický koprocesor 8087.

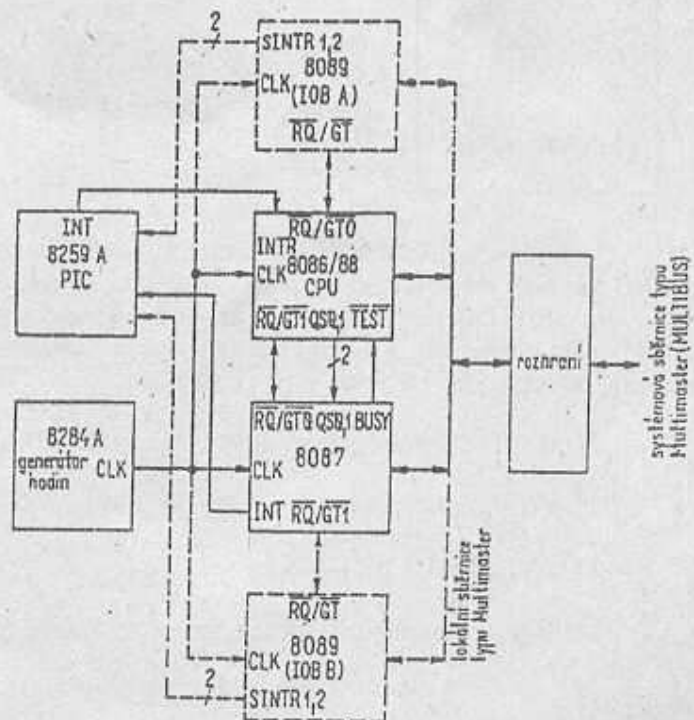
Je vyroben technologií HMOS v pouzdře DIL40. Jeho architektura, funkční možnosti i rozložení vývodů pouzdra jsou maximálně přizpůsobeny pro spojení s mikroprocesory 8086 a 8088.

Jde o vysoce specializovaný na numerické výpočty koprocesor, používaný ve dvojici s 8086, 8088, 80186 a 80188.

Vzájemné vztahy v kombinaci CPU s 8087 jsou tak těsné, že se na ni hledí jako na jeden numericky orientovaný mikroprocesor ve 2 pouzdech a označuje se zkratkou NDP (Numeric Data Processor). V tomto smyslu lze chápat 8087 jako rozšíření univerzálního mikroprocesoru o další instrukce i s nimi související typy dat a registry.

Doplnění CPU o 8087 přináší téměř úplnou úsporu paměti dosud zabírané numerickými podprogramy. Hlavním přínosem je v průměru 100 násobné zrychlení matematických operací.

Přímé spojení koprocesoru 8087 s CPU je zjednodušeně naznačeno na obr. č. 7.1 [2].



Obr. č. 7.1 Spojení univerzálního mikroprocesoru s 8087 i s jedním nebo dvěma stykovými mikroprocesory.

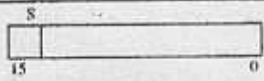
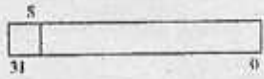
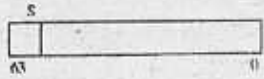
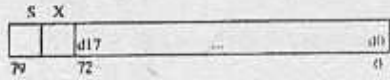
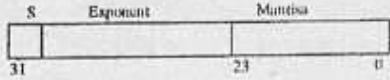
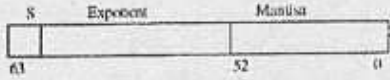
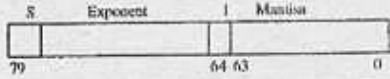
Zde je čárkovaně znázorněno i připojení jednoho až dvou stykových koprocesorů 8089. CPU s koprocesory tak vytváří jádro dvouprocesorového až čtyřprocesorového mikropočítače, založeného na lokální sběrnici typu MULTIMASTER.

K řízení činnosti na této sběrnici postačí signály produkované i vyhodnocované samotnými mikroprocesory.

7.1.1 Základní vlastnosti 8087

Koprocesor 8087 zpracovává 7 typů dat podle tabulky č.8.1.

Tabulka č. 7.1

Formát a typ	Počet slabik	Rozsah (přibližně)	Zobrazení
Slovo integer	2	-32767 až 32768	
Krátký integer	4	$-2 \cdot 10^9$ až $2 \cdot 10^9$	
Dlouhý integer	8	$-2 \cdot 10^{18}$ až $2 \cdot 10^{18}$	
BCD zhuštěný	10	$-10^{18} + 1$ až $10^{18} - 1$	
Krátký real	4	$\pm 1 \cdot 10^{38}$ až $\pm 3 \cdot 10^{38}$	
Dlouhý real	8	$\pm 10^{308}$ až $\pm 10^{308}$	
Pomocný real	10	$\pm 10^{4932}$ až $\pm 10^{4932}$	

Nejvýznamnější bit (MSB) v každém formátu je znaménkový (S-signum). Zhuštěný formát BCD (packed decimal) má absolutní hodnotu vyjádřenou 18 dekadickými číslicemi po 4 bitech a 7 bitů nejvyšší slabiky nevyužívá (X), resp. koprocesor při převzetí operandu tyto bity ignoruje a ve výsledku do nich dosazuje nuly. Formáty typu krátký a dlouhý real (s pohyblivou řádovou čárkou) odpovídají standardu IEEE 754. To znamená, že odpovídají nárokům běžných překladačů z vyšších prog. jazyků.

Exponenty čísel real jsou vyjádřeny v kódu (E+BIAS), kde BIAS je posunutí o hodnotě:

- 127 (7FH) - pro každý real.
- 1 023 (3FFH) - pro dlouhý real.
- 16 383 (3FFFH) - pro pomocný real.

Mantisy jsou normalizovány tak, že vyhoví vztahu:

$$2 > M \geq 1.$$

U formátů krátký real a dlouhý real se pro zdvojnásobení přesnosti využívá skutečnosti, že u nenulových mantis je vždy nejvyšší bit roven 1. Tento bit se proto nepřenáší a je implicitní (hidden bit). Formát pomocný real však jednotkový nejvyšší bit mantisy obsahuje explicitně.

Nehledě na výběr formátů a typů uvnitř koprocessoru 8087 se všechny operace vykonávají v **pomocném formátu** (temporary real), tj. s největší možnou přesností i rozsahem hodnot, přičemž převody do pomocného formátu i zpět jsou samočinné. Hlavním důvodem pro toto řešení je snaha redukovat vliv přetečení a podtečení v posloupnosti dílčích výpočtů s čísly kratších formátů, kdy konečný výsledek sice formátu vyhovuje, ale mezivýsledek nikoliv.

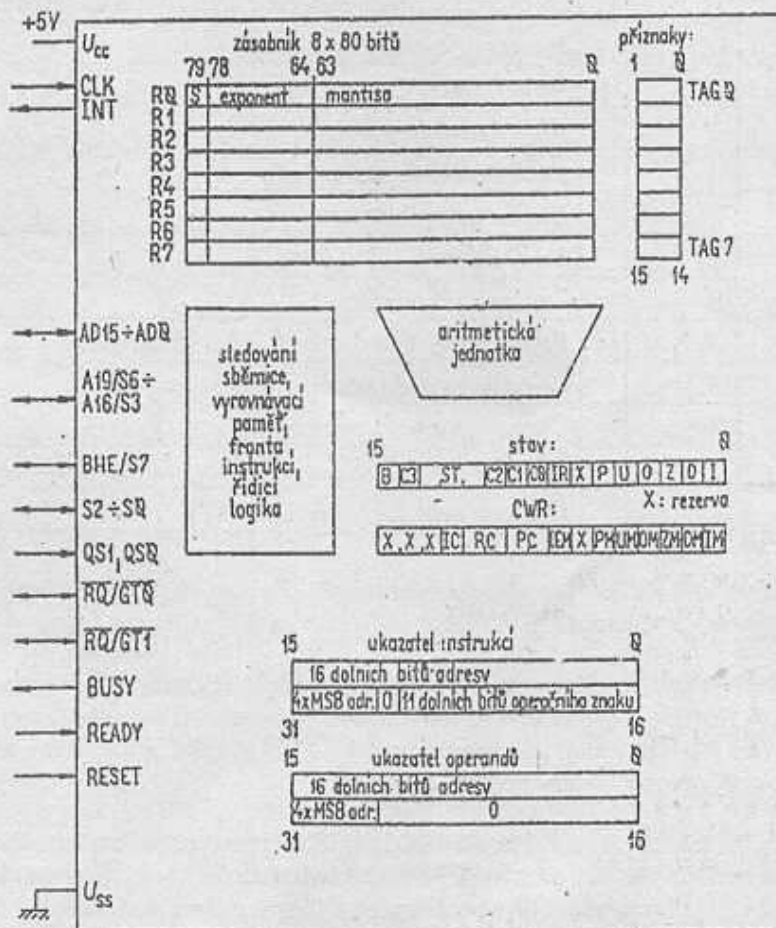
Na podporu formátů dat má jazyk ASM86, kromě deklaračních pseudoinstrukcí **DB** a **DW** i direktivy **DD** (define doubleword), **DQ** (define quadword) a **DT** (define tenbyte) pro definování dat a přidělování paměti. Jejich používání je zřejmé z příkladu:

```

SHORT_REAL DD 1.125           ;ulož jako krátký real
SHORT_INT DD 12345678H        ;ulož jako krátký integer
LONG_REAL DQ -1.9             ;ulož jako dlouhý formát real
LONG_INT DQ 123456789ABCDH    ;ulož jako dlouhý integer
BCD DT -91567                 ;ulož jako zhuštěný BCD
REAL_CONST DT -1.3            ;ulož jako pomocný real

```

Architektura 8087 viz obr. č. 7.2 [2] je rozdělena do **dvou základních částí**:



Obr. č. 7.2 Zjednodušené blokové schéma numerického koprocessoru 8087.

Z uvedeného schématu lze konstatovat, že architektura 8087 je **dvoustupňová**. Jsou zde tyto subprocesory.

- **Řídící jednotka CU** (control unit), která přijímá a dekóduje instrukce, čte a zapisuje operandy do paměti a vykonává řídicí instrukce. Součástí CU je i fronta instrukcí stejná jako v CPU. Délka fronty se nastavuje automaticky na 6 slabik nebo na 4 slabiky (zda jde o součinnost s 8086 nebo 8088)

Řídící logika je připojena k datové i adresové sběrnici a sleduje i stavové signály z CPU. Na instrukce čtené z hlavní paměti nereaguje, dokud nepřijde na řadu instrukce ESC.

- **Numerická operační jednotka NEU** (numeric execution unit), která má s CU společnou frontu operandů. Vlastní numerické výpočty zajišťuje aritmetická jednotka, která mantisy čísel zpracovává s přesností 68 bitů, tj. s přesností ještě o 4 bity větší, než má pomocný formát. Přidané bity umožňují rozhodování o zakrouhlování a o výjimečných stavech. NEU má pro data v pomocném formátu zásobníkovou paměť z osmi 80bitových registrů doplněných 2bitovými značkami. Do zásobníku je přístup sekvenční (LIFO) i libovolný.

Vrchol zásobníkové paměti je adresován třemi bity v poli **ST** 16bitového **stavového registru**. Při sekvenčním, ukládání se napřed dekrementuje pole **ST** a pak se data zapíše. Při sekvenčním čtení se napřed data přečtou, a pak **ST** inkrementuje.

Při libovolném přístupu může být každý registr zásobníku adresován indexem relativně k vrcholu zásobníku.

Tabulka č. 7.2 Význam bitů stavového registru koprocessorů 8087

Číslo	Označení	Význam
0	I	Nevhodná činnost-přetečení, podtečení zásobníku, neplatný
1	D	Nenormalizovaný operand
2	Z	Dělení nulou
3	O	Přetečení exponentu.
4	U	Podtečení exponentu.
5	P	Chyba zakrouhlení... .
6	X	Nevyužívá se.
7	IR	Žádost o přerušení.
8, 9	C0, C1	Podmínkový kód nastavovaný instrukcemi
10, 14	C2, C3	komparace a testovacích instrukcí.
13, 12	ST	Ukazatel vrcholu zásobníku.
11		
15	B	Operace není dokončena (BUSY).

Ostatní bity stavového registru představují informace o chybových stavech, o podmínkách pro některé instrukce a stav rozpracovanosti operace (BUSY).

Každý z **šesti** typů chyb, indikovaných bity 5 až 0 stavového registru, může být maskován dosazením "1" do příslušného bitu v **řídícím registru CWR**. Masku tvoří bity 5 až 0 CWR, které jsou stejnohlé k bitům indikací chyb ve stavovém registru a mají i obdobné označení. Maskováním určité chyby se znemožní žádost o přerušení a 8087 pokračuje v činnosti standardním způsobem.

Přerušení však lze maskovat i nastavením 1 do bitu IEM (interrupt enable mask bit) registru CWR. V tomto případě jsou znemožněna všechna přerušení. Při IEM=0 může každá nemaskovaná chyba vyvolat žádost o přerušení a orientovat CPU do přerušovacího podprogramu. Tento podprogram zkoumá aktuální obsah dvou 32bitových registrů (**ukazatele instrukce a ukazatele operandů**) v 8087, které pro tento účel přepíše do hlavní paměti. Oba ukazatele identifikují **instrukci i operand**, u kterých nastala hlášená chyba.

Ukazatel instrukcí udržuje jen 11 dolních bitů kódu instrukce. **Horních 5 bitů je vždy 11011B**, neboť může jít jen o **instrukci ESC**.

Zbývající bity usnadňují řízení přesnosti (**PC**), zaokrouhlování (**RC**) a odezvu na nekonečno.

Význam:

PC: 00 - přesnost 24 bitů.
01 - rezervováno.
10 - přesnost 53 bitů.
11 - přesnost 64 bitů.

RC: 00 - zaokrouhlení k nejbližšímu číslu.
01 - zaokrouhlení směrem k -.
10 - zaokrouhlení směrem k +.
11 - zanedbání zbytku.

IC: 0 - nekonečná veličina bez znaménka (od nekonečna do nekonečna).
1 - dvě různé veličiny se znaménkem (od nekonečna do nekonečna).

Při každé inicializaci nebo RESETu 8087 se v CWR nastavuje:

PC = 11, RC = 00, IC = 0, IEM = 0 a všechny bity masky chyb = 1.

Ve stavovém registru jsou všechny bity s výjimkou podmínkového kódu nulovány.

7.1.2 Instrukční soubor a součinnost 8087 a CPU.

Jazyk symbolických adres ASM86 zahrnuje kromě instrukcí pro 8086/88 i 69 instrukcí pro 8087. Nejde však o instrukce navíc, ale o doplňkovou symboliku, která sjednocuje a usnadňuje pro dvojici CPU + 8087 zadávání různých variant instrukce ESC. Instrukce pro 8087 tvoří samostatné programy, ale umísťují se do programu CPU.

Pro každou mnemonickou instrukci koprocesoru, která ovlivňuje NEU, generuje assembler dvě strojové instrukce:

- kód instrukce WAIT,
- kód příslušné varianty ESC.

Příklad.

Máme direktivou DD definováno pole začínající adresou SHORTREAL. Pak instrukce

FST SHORTREAL,

kteřá konvertuje obsah vrcholu zásobníku v 8087 do krátkého typu real a výsledek ukládá na SHORTREAL, bude assemblerem přeložena takto:

10011 011 - WAIT

11011 001 00 010 110 DISPL DISPH

ESC OZ OD OZ R/M

Dvě dvojice číslic 001 a 010 představují externí OZ pro FST.

Instrukce WAIT zkoumá signál TEST, ten se předřazuje, aby se koprocesoru umožnilo dokončit rozpracovanou operaci dříve, než dekóduje ESC. Je-li

TEST

aktivní, ESC se dekóduje a CPU s 8087 pokračují v souběžné činnosti. Instrukce ESC mají vždy 2,3 nebo 4 slabiky. Viz. předchozí příklad, vyšších 5 bitů prvního slova je operační znak ESC, 6 bitů je tzv. externí operační znak, který určuje typ operace, a ostatní bity jsou modifikační. Třetí a čtvrtá slabika představují 8bitový nebo 16bitový offset.

Za provozu je řídicí jednotka CPU připojena k datové i adresové sběrnici, sleduje i stavové signály:

S0, ST, S2, S6, QS1 a QS0 z CPU a na instrukce čtené z paměti nereaguje, dokud nepřijde na řadu instrukce ESC. Tu pak zpracovávají CPU i koprocesor rozdílně. CPU rozliší, zda ESC odkazuje na paměť, nebo nikoliv.

Při odkazu na paměť CPU obvyklým způsobem vypočte adresu a vykoná tzv. **prázdný cyklus čtení**. Při něm CPU přečtené slovo na datové sběrnici ignoruje a přejde k další instrukci programu. V cyklu bez odkazu na paměť CPU prostě přejde na další instrukci.

Rozlišují se tři možné odkazy instrukcí ESC k paměti:

1. bez přístupu k paměti,
2. převzetí operandu z paměti do 8087,
3. zápis operandu z 8087 do paměti.

V prvním případě 8087 operaci předepsanou v ESC pouze vykoná.

V dalších obou případech, kdy ESC odkazuje na paměť, koprocessor využívá prázdný cyklus čtení CPU k převzetí adresy, kterou CPU vysílá na adresovou sběrnici. Pokud ESC předepisuje převzetí operandu, přijme 8087 navíc přečtené slovo z datové sběrnice. Jestliže však operand má být delší než jedno slovo, 8087 s využitím signálů

RQ/GT

převzme od CPU řízení sběrnice a samostatně čte ostatní data z navazujících adres. Pak koprocessor sběrnici opět uvolní a přistoupí k vykonání žádané operace.

V případech, kdy ESC předepisuje zápis výsledku do paměti, koprocessor ignoruje data čtená CPU v prázdném cyklu a čeká, až bude informace z NEU připravena k zápisu. Pak CPU převzme řízení sběrnice od CPU a data zapíše na zachycenou adresu do paměti. Z toho plyne, že předávání parametrů i výsledku výpočtu mezi CPU a 8087 se realizuje pouze přes hlavní paměť.

Instrukce lze rozdělit do 6 skupin:

- přesuny dat: operand z paměti o libovolném formátu může být přiveden do pomocného formátu a zaveden do registrového zásobníku s posunem jeho vrcholu (PUSH). Analogicky může být zaveden obsah registru. Opačně vrchol zásobníku lze převést do žádaného formátu a uložit do paměti. Také lze posouvat data v zásobníku a vzájemně vyměnit obsah registru s vrcholem. Po každém přesunu se aktualizují příznaky u registrů.
- aritmetiku: Výsledek sčítání, odčítání, násobení nebo dělení přijde na vrchol zásobníku nebo do specifikovaného registru. Jeden ze zdrojových operandů je vždy na vrcholu zásobníku a pro formáty REAL může být druhý operand v registru nebo v paměti. Některé instrukce také posouvají vrchol zásobníku po dokončení arit. operace. Kromě základních operací skupina obsahuje také instrukce pro odmocninu, posun měřítka mocninou dvou, dělení modulo, zaokrouhlení čísla real atd.
- komparace: Porovnávají vrchol zásobníku se zdrojovým operandem, existují podmínkové bity ve stavovém registru.
- transcendentny: Patří sem funkce: $\text{tg}()$, $\text{arctg}()$, $\log()$ atd.
- konstanty: S přesností do 19 dekadických míst lze na vrchol zásobníku zavést konstanty: 0, 1, $\log_2 1$, $\log_{10} 2$ a $\ln 2$.
- instrukce řízení: Většinou se používají pro iniciaci, zpracování chybových atd.

7.1.3 Programový emulátor E8087.

Mnohdy není předem jasné, zda bude nutné spojit s univerzálním mikroprocesorem numerický koprocessor 8087 nebo zda mikroprocesor stačí na výpočty sám. I v tomto případě lze psát programy s obsáhlými numerickými částmi co nejstručněji - předpokládá se tedy použití 8087.

Existuje totiž programový emulátor E8087 [2,19], což je v podstatě soubor programů (16 Kslibik) pro CPU, které mohou nahradit funkce 8087.

Zdrojové programy v jazycích ASM86, PLM86 ... jsou pak na použití koprocessoru nezávislé a teprve v etapě spojování přeložených modulů programem LINK86 se rozhoduje o ponechání kódu instrukcí pro 8087 nebo o jejich nahrazení instrukcemi, které zajistí vyvolání podprogramů.

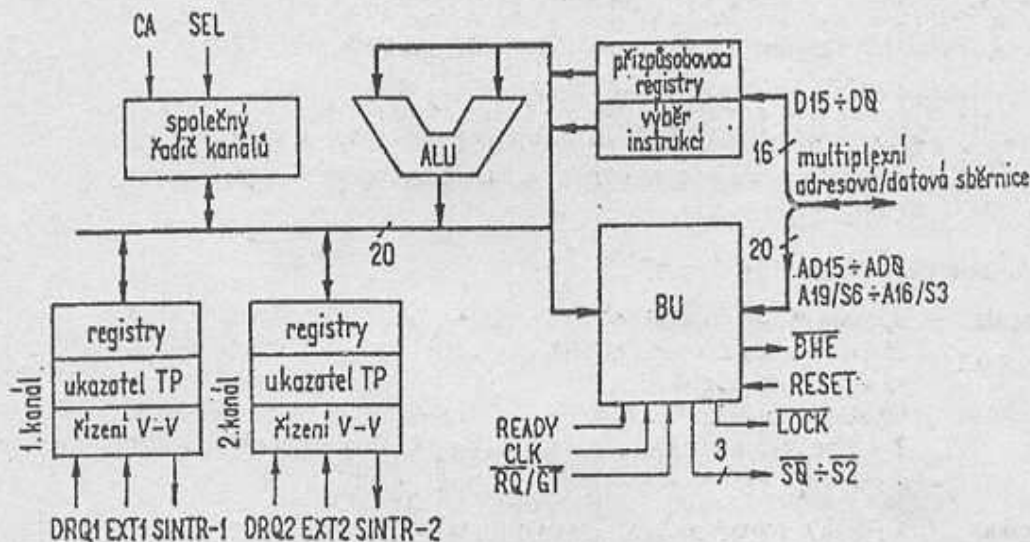
7.2 Stykový koprocesor 8089.

Jde o specializovaný mikroprocesor 8089 (IOP-Input Output Processor) [2,19], který bývá označován za kombinaci nezávislého mikroprocesoru s dvojitým řadičem DMA.

Koprocesor 8089 (obr. č. 7.3 [2]) je určen pro práci se soubory a vyrovnávacími paměťmi při řízení diskových jednotek, pro realizaci opravných podprogramů při chybách atd.

Data vybraná koprocesorem z adresovatelného zdroje přecházejí do 8089, kde s nimi může být nakládáno předem naprogramovaným způsobem. Dále jsou data přenášena do cílového místa, které je rovněž adresovatelné.

Pokud se při takové operaci (čtení-modifikace-zápis) adresa vztahuje k paměti, pak se po dílčích přenosech inkrementuje. Adresa vstupní/výstupní brány se nemění.



Obr. č. 7.3 Zjednodušené blokové schéma stykového mikroprocesoru 8089.

Každý dílčí přenos probíhá ve 2 cyklech:

1. cyklus pro výběr,
2. cyklus pro přenos.

Oba cykly se mohou odehrávat na téže sběrnici nebo různých sběrnicích.

Architektura i instrukční soubor 8089 jsou orientovány hlavně na řešení stykových úloh. Rozumějí se tím přesuny dat s detekcí a opravami chyb při nastavitelných podmínkách ukončení, opakování přenosů, transformace dat a řízení periferních zařízení.

Použitím 8089 se pro CPU komplexní úloha přenosu dat redukuje na předání požadavku a parametrů přenosu.

7.3 Numerický koprocesor 80287 a 80387.

Koprocesor 80287 vznikl jako modifikace mat. koprocesoru 8087, s tím že je zvláště přizpůsobený modernějšímu 16bitovému mikroprocesoru 80286. Z hlediska rychlosti i matematických možností je stejný jako 8087. Vývody 80287 jsou však přizpůsobeny přímému spojení s 80286 na lokální sběrnici, tj. většímu adresovanému prostoru, odlišné funkci některých vývodů a složitějšímu algoritmu styku se sběrnicí, kdy BU s jednoduchou frontou instrukcí v CPU 8086 je u 80286 nahrazena dvěma subprocessory BU a IU.

Analogicky vznikl numerický koprocesor 80387, určený k novému 32bitovému univerzálnímu mikroprocesoru typu 80386.

LITERATURA

- [1] Katalog obvodů TTL fy Texas Instruments
- [2] Valášek: Monolitické mikroprocesory a mikropočítače, SNTL, Praha 1989.
- [3] Strelec, Líška: Architektury procesorů RISK, Grada, Praha 1992.
- [4] Operační systém MS-DOS, ČSVTS FEL ČVUT 1989.
- [5] Kunzel,Žáček: Mikroprocesorová technika, ČVUT.
- [6] Bílek, Šnorek, Záček: Mikroprocesorová technika, skriptum FEL ČVUT.
- [7] Šnorek: Číslicové počítače I, cvičení, skriptum FEL ČVUT.
- [8] Hlavička: Číslicové počítače II, skriptum FEL ČVUT 1992.
- [9] Hlavička: Architektura počítačů, skriptum FEL ČVUT 1994.
- [10] Blatný J., Dábek V.: Standardizace číslicových rozhraní, skriptum, Brno 1990.
- [11] Michal Brandejs: Mikroprocesory INTEL, Grada, Praha 1994.
- [12] Němec F.: Počítačové systémy, VŠB FH, skriptum, Ostrava 1989.
- [13] Sloup V., Rozehnal Z.: Jednočipové mikropočítače, skriptum ČVUT, Praha 1992.
- [14] Lokální sběrnice PCI,Chip 12/1993.
- [15] Procesory, Chip 9/1994.
- [16] Vývoj grafických zobrazovačů, CW 35/94.
- [17] Super VGA, PC WORLD 2/93.
- [18] Grafické karty, Chip 9/92.
- [19] Ličev L.: Architektura počítačů, skriptum FEI VŠB-TUO Ostrava, 1996.
- [20] Ličev L.: Základy výpočetní techniky, skriptum FEI VŠB -TUO Ostrava, 1996.
- [21] Minasi M.: Pevné disky od A do Z, Grada, Praha 1992.
- [22] IRIS Indigo2 & IRIS POWER Indigo2, Technical Report, Copyring 1993, 1994 Silicon Graphics.
- [23] A Tour of the Pentium Pro Processor Microarchitecture, October 1995, Copyright Intel Comporation
- [24] CHIP 3/93, 2/95, 4/96, 5/96
- [25] Softwarové noviny 1/93
- [26] Bajt 1/90, 5/92, 7/92, 1/94
- [27] Minasi M.: Velký průvodce hardware, Grada, Praha 1992.
- [28] Firemní materiály: TGNumeric, IBM, Microsoft, HP, S3, ATI a řada dalších
- [29] Scannery - Jaroslav T. Hyan, Grada 1993
- [30] CHIP, leden 1993
- [31] CHIP, červen 1994
- [32] CAD & GRAPHICS - červen 1995
- [33] Kolektiv autorů: Jak publikovat na počítači, SCIENCE, 1996
- [34] Frisch H.: Zákaly elektroniky a elektronických obvodů, SNTL
- [35] Smutný L.: Mikroelektronické a optoelektronické systémy, VŠB
- [36] <http://www.Questlink.com>, 1998
- [37] <http://www.Intel.com>, 1998
- [38] <http://www.DEC.com>, 1998

- [39] <http://www.IBM.com>, 1998
- [40] <http://www.SGI.com>, 1998
- [41] <http://www.SUN.com>, 1998
- [42] <http://www.AMD.com>, 1998
- [43] <http://www.Cyrix.com>, 1998
- [44] Horák J.: Hardware pro pokročilé, Computer press, 1997
- [45] Drábek V.: Výstavba počítačů, PC-DIR s r.o., 1995
- [46] <http://www.HP.com>, 1998

Ústřední knihovna VŠB-TU Ostrava

II-251469-3352/99

Číslo skladové:	1909	400
Určeno pro posluchače:	3. ročníku FEI	
Autor:	Ing. Lačezar Ličev, CSc.	
Katedra, institut:	informatiky	456
Název:	Architektura počítačů I.	
Místo, rok, vydání:	Ostrava, 1999, 1. vydání	
Počet stran:	220	
Vydání:	VŠB - TECHNICKÁ UNIVERZITA OSTRAVA	
Titul:	Lučevní střediska VŠB	
Náklad:	200 ks	
Tematická skupina:	17	
Hovorenia MK ČSR č. j. 21.514/79 ze dne 4. 12. 1979		

ISBN 80 - 7078 - 631 - 0