

CHARAKTERISTIKA MODERNÍCH PENTIÍ

Úvod:

Flynnova klasifikace paralelních systémů

- Paralelní systémy lze třídit z hlediska počtu toků instrukcí a počtu toků dat:
 - SI** – systém s jedním tokem instrukcí (Single Instruction stream)
 - MI** – systém s několika toky instrukcí (Multiple Instruction stream)
 - SD** – systém s jedním tokem dat (Single Data stream)
 - MD** – systém s několikanásobným tokem dat (Multiple Data stream)
- Použití těchto dvou hledisek vede ke vzniku čtyř základních typů počítačů označovaných zkratkami SISD, SIMD, MISD a MIMD.
- **SISD, SIMD** – klasický jednoprosesorový počítač von Neumannova typu zpracovávající data sériově.
- **SIMD** (Single Instruction stream, Multiple Data stream) – pole procesorů zpracovávající

paralelně pole hodnot podle společného programu.

- **MIMD** (Multiple Instruction stream, Multiple Data stream) – multiprocesorový systém, v němž každý procesor je řízen samostatným programem a pracuje s jinými daty než ostatní procesory.
- **MISD** – soustava procesorů pracujících podle různých programů na společných datech (označováno jako enfant terrible této klasifikace) – v praxi neobsazená alternativa.

Uplatnění SIMD na úrovni instrukcí

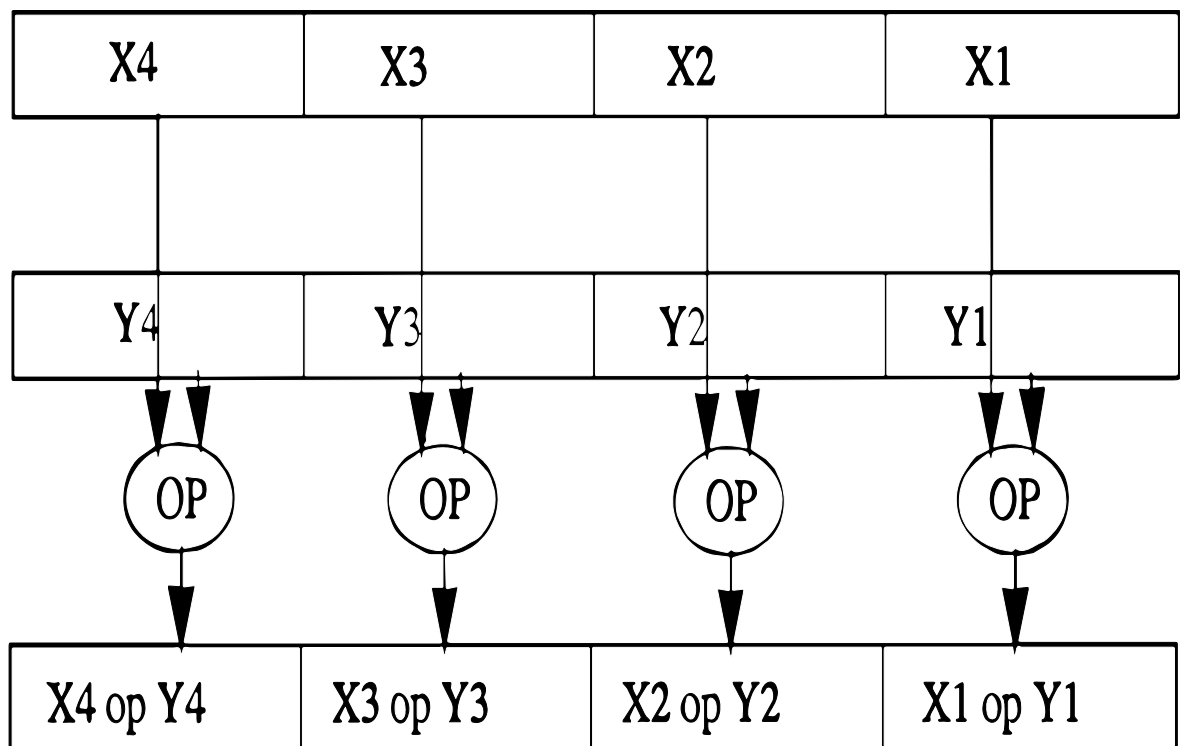
- Typický rys multimediálních aplikací – paralelně prováděné operace na stejných datech.
- Strategie SIMD uplatněná na úrovni instrukcí – jeden tok instrukcí na jednom procesoru, instrukce operuje na násobných ale shodných datech.
- Příklad: technologie MMX, která operuje paralelně s násobnými daty ve zkomprimované formě uloženými v 8/16 bitovém registru.

Klíčové znaky moderních procesorů Pentium

1. Podpora Streaming SIMD Extensions 2 (SSE2).
2. Intel Netburst micro-architecture.
3. Implementace vlastností architektury Intel Netburst micro-architecture do procesoru Pentium.

Vývoj technologie SIMD na úrovni instrukcí

- SIMD – Single Instruction Multiple Data, jeden typ instrukce se provádí na více datech.
- Typická operace SIMD – obr. 1.
 - Dvě skupiny operandů, každá obsahuje 4 zkomprimované datové prvky (X1, X2, X3, X4 a Y1, Y2, Y3, Y4).



Obr. 1 Operace typu SIMD

- Takto to bylo u technologie MMX.
- Technologie MMX umožňovala realizovat operace SIMD na zkomprimovaných slabikách, slovech, dvouslovech (vše celá čísla) uložených v osmi 64 bitových registrech.
- Speciální instrukce byly šity na míru pro multimediální a komunikační aplikace.

Registry MMX, 64 bitů		Registry XMM, 128 bitů
MM7		XMM7
MM6		XMM6
MM5		XMM5
MM4		XMM4
MM3		XMM3
MM2		XMM2
MM1		XMM1
MM0		XMM0

- **Pentium III** – model SIMD byl změněn na **Streaming SIMD Extensions (SSE)**.
- Rozdíl oproti klasickému SIMD:
 - Operace se provádějí na operandech obsahujících čtyři komprimovaná čísla v **pohyblivé řádové čárce**.
 - Operandy jsou uloženy buď v paměti nebo v **osmi 128 bitových registrech** (registry XMM).
 - Speciální množina instrukcí vytvořená pro procesory MMX byla rozšířena o **dalších 64 instrukcí**.

- **Pentium 4** – byl použit model **Streaming SIMD Extensions 2 (SSE2)** s těmito vlastnostmi:
 - Operace se provádějí na těchto operandech:
 - dvě čísla v pohyblivé řádové čárce,
 - dvojnásobná přesnost,
 - 16 zkomprimovaných slabik,
 - 8 zkomprimovaných slov,
 - 4 zkomprimovaná dvouslova,
 - 2 zkomprimovaná čtyřslova (čtyři poslední alternativy – čísla typu integer).
 - Operandy mohou být v paměti nebo registrech.
 - Podpora aritmetiky SIMD pro práci se 64 bitovými operandy typu integer.
 - Instrukce pro konverzi mezi datovými typy (původními a novými).
- Trend: **SIMD na úrovni instrukcí, rozšiřování množiny datových typů.**

V čem spočívá implementace vlastností Netburst micro-architecture v Pentiu 4

A. Rychlé vyrovnávací paměti integrované do čipu (on-chip cache)

- Na rozdíl od dřívějších typů to není pouze klasická rychlá vyrovnávací paměť L1, ale:
 - rychlá vyrovnávací paměť L1 pro data kapacity 8 kB,
 - rychlá vyrovnávací paměť typu Execution Trace Cache kapacity 12K μ op,
 - rychlá vyrovnávací paměť L2, 8 way, kapacity 256 kB typu Advanced Transfer Cache.
- Pojem Advanced Transfer Cache – vyrovnávací paměť L2 (integrována do čipu procesoru), pracuje na stejném kmitočtu jako procesor).
- Trend: **výrazná snaha o rozšíření pojmu „rychlá vyrovnávací paměť“ a hledání nových možností uplatnění v architektuře PC.**

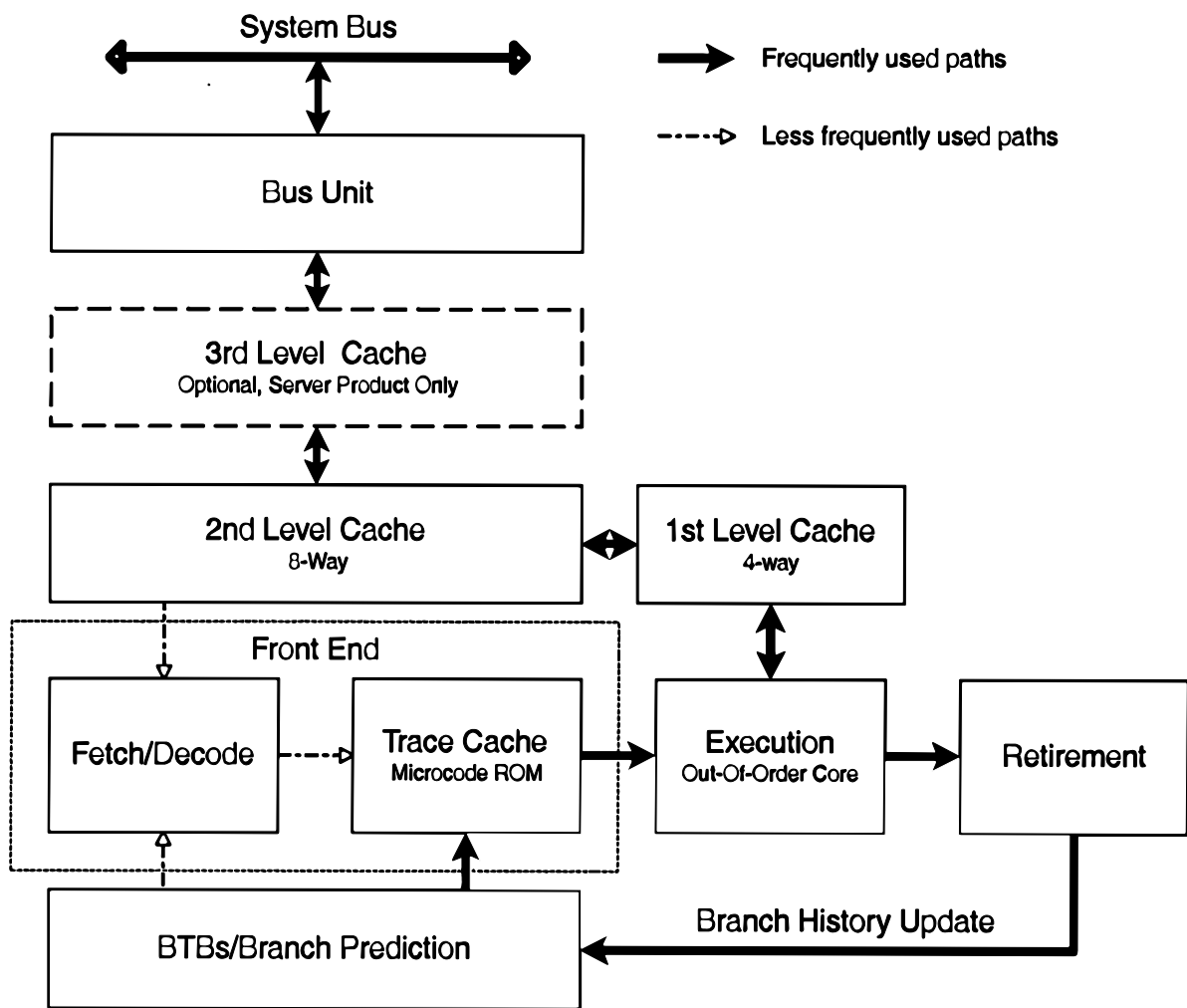
B. Rychlost komunikace procesoru s okolím

- Pentium III komunikovalo s okolím na frekvenci 100/133 MHz, Pentium 4 má komunikaci s okolím synchronizovanu kmitočtem 800 MHz.
- Trend: **zvyšování rychlosti komunikace procesoru s okolím.**

Další vlastnosti Intel NetBurst Micro-Architecture

- Hyperskalární (zřetězení instrukcí – více jak 10 jednotek) a superskalární architektura (dvě fronty).
- Dokonalé uplatnění principů zřetězení, přičemž různé komponenty jsou synchronizovány různými frekvencemi, některé vyššími, některé nižšími frekvencemi než je frekvence procesoru.
- Synchronizace procesoru frekvencí vyšší než 3 Ghz (stav v r. 2003).
- Zvyšování kmitočtu procesoru – výrazná snaha o zvýšení počtu jednotek (kvůli zvyšování kmitočtu je to nutnost).

- Komunikace s okolím – 800 Mhz (systémová sběrnice).
- Provádění instrukcí mimo pořadí.
- Konstrukce obvodů tak, aby se nejčastěji prováděné instrukce realizovaly rychleji.



Funkce Front End

- Sestává ze dvou bloků:
 - *Fetch/Decode*
 - *Execution Trace Cache*

- *Front End* má tuto funkci:
 - čtení instrukcí, jejich dekódování a náhrada posloupnostmi mikrooperací (Intel rozlišuje tyto typy instrukcí: *instruction*, *complex instruction*, *special purpose instruction*).
 - přenos dříve dekódovaných instrukcí z *Execution Trace Cache*,
 - predikci výsledků instrukcí skoku
- Architektura moderních mikroprocesorů fy Intel usiluje o originální řešení problémů různých zpoždění, k nimž dochází při provádění instrukcí. Mezi tato zpoždění patří, např.:
 - doba potřebná na dekódování instrukcí
 - doba potřebná na řešení problémů větvení programu (různé typy skoků).
- Redukce doby potřebné na dekódování instrukcí – využití *Execution Trace Cache*.
- Princip:
 - Instrukce jsou rozdekódovány (*Translation Engine*) do posloupnosti mikrooperací (μop).
 - Instrukce jsou reflektovány posloupností mikrooperací – tyto posloupnosti se nazývají *traces* (kopie, obrazy) a jsou uloženy do *Execution Trace Cache*.

- Posloupnosti mikrooperací jsou uloženy tak, jak odpovídá toku programu.
- Instrukce skoku - v *Execution Trace Cache* jsou k těmto mikrooperacím uloženy do stejných řádků výsledky těchto skoků => zvyšuje se pravděpodobnost správné predikce (tvrdí fa Intel), zvyšuje se podíl kódu prováděného z *Execution Trace Cache*.
- Na předvídání výsledku skoku se podílejí *Execution Trace Cache* a *Translation Engine*.
- Díky tomu, že jsou cílové adresy uloženy v *Execution Trace Cache*, mohou se po předpovězení výsledku brát mikroinstrukce z *Execution Trace Cache* (pokud tam je tato část programu zavedena, jinak se musí část programu, kde se bude pokračovat, zavést z paměti – nemusí to pochopitelně být operační paměť).
- Pentium 4 – *Execution Trace Cache* může uchovat až 12K μ operací.
- Procesor Pentium 4 je optimalizován tak, že nejčastěji prováděné instrukce jsou realizovány z *Execution Trace Cache*, pouze malé množství instrukcí se provádí z paměti mikroprogramů (*microcode ROM*).

- Jinak řečeno: většina instrukcí má svou reprezentaci uloženou v *Execution Trace Cache*, pouze pro malou část je využívána paměť mikroprogramů.
- Trend: **zkrátit čas potřebný k zajištění přístupu k mikroprogramům.**

Jednotka Out-of-Order Core

- Jednotka *Out-of-Order Core* umí přeuspořádat provádění instrukcí tak, aby neutrpěla logika programu.
- Princip: pokud nemůže být některá μ op provedena, protože nemá k dispozici data, provede se jiná μ op => tímto způsobem je možné odstranit možná zpoždění, která vzniknou v důsledku nedostupnosti dat.
- V mechanismech rozhodujících o zahájení realizace konkrétní μ operace se bere v úvahu také dostupnost potřebných hardwarových prostředků.
- Pořadí provádění mikrooperací se může modifikovat podle toho, zda jsou k dispozici hodnoty operandů pro patřičné mikrooperace a jsou volné hardwarové prostředky pro její realizaci.

- Trend: **optimalizovat principy realizace instrukcí zpracovávaných ve frontě s cílem zkrátit jejich provádění.**

Zpětné uspořádání posloupnosti instrukcí

- Tuto činnost provádí jednotka, kterou Intel označuje jako *Retirement Section*.
- Za ukončení mikrooperace se považuje stav poté, co je výsledek uložen do *cílového registru*.
- Jednotka *Reorder Buffer (ROB)* provádí zpětné uspořádání výsledků instrukcí tak, aby odpovídaly původní posloupnosti.
- Jednotka *Retirement Section* uchovává informaci o tom, jak dopadly skoky a předává ji do *BTB (Branch Target Buffer)*.
- Podle této informace se obnovuje obsah *BTB*.

Předvídání výsledků skoků

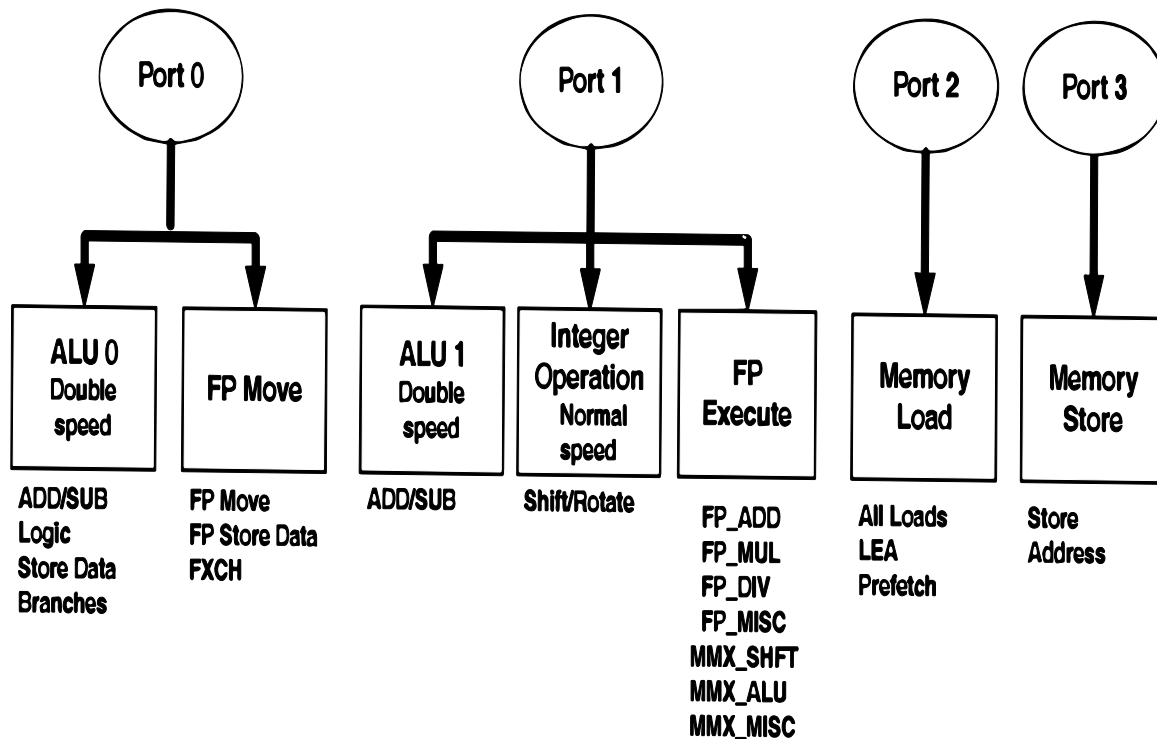
- Předvídání výsledků skoků je velmi důležité pro procesory zpracovávající fronty instrukcí.
- Techniky předvídání skoků (*Branch Prediction*) umožňují pokračovat ve zpracování programu na

správném místě předtím, než je instrukce skoku zpracována a znám výsledek skoku.

- Pojem *Branch Delay* reprezentuje zpoždění, které nastane, pokud dojde při předvídání výsledku skoku k omylu.
- Správná předpověď - *Branch Delay* je nulové.
- Mylná předpověď - *Branch Delay* bude odpovídat délce fronty (do fronty se musí načíst instrukce z jiné adresy než bylo předpověženo a ta se do prováděcí jednotky dostane za takový počet cyklů, který odpovídá počtu komponent podílejících se na zpracování).
- Architektura Intel NetBurst micro-architecture umí předpovídat tzv. *krátké skoky* (podmíněné a nepodmíněné skoky).
- Neumí předpovídat vzdálené skoky (?), instrukce *iret* a programová přerušení.
- Trend: **v oblasti předpovídání výsledků skoků firma Intel usiluje o inovaci technik pro předpovídání výsledků skoků.**

Provádění instrukcí

- Instrukce jsou prováděny formou posloupnosti μ operací čtené buď z paměti μ programů nebo z *Execution Trace Cache*.
- Plánování instrukcí zohledňuje dostupnost operandů – tzn. jsou prováděny instrukce, které mají k dispozici operandy.
- Pro superskalární architekturu uplatňovanou v procesorech firmy Intel je typické to, že obsahuje hardwarové prostředky pro paralelní provádění μ operací, přičemž tyto prostředky rozlišují typ prováděné μ operace.
- Jde o principy provádění μ operací => mluvíme proto o mikro-architektuře.
- Tato mikro-architektura sestává ze čtyř portů – každý pro provádění jiných typů mikrooperací.



- Výsledek: v jednom cyklu se může současně provádět až 6 μ operací, přičemž všechny souvisejí s prováděním instrukcí zařazených do jedné fronty – tzn. jde o paralelismus uplatňovaný na jiné úrovni – paralelismus na úrovni prováděcí jednotky (Execution Unit) – viz přednáška o uplatnění prvků architektury RISC v mikroprocesorech Intel.