

**IA039**

**Procesory a paměti**

# Procesory – CISC

## Complex Instruction Set Computer

- Příklady:
  - PDP 11, VAX, IBM 370, Intel 80x86, Motorola 680x0, ...
- Princip:
  - Nedělej programem to, co může udělat hardware

# Důvody existence

- Velikost a rychlost paměti
  - Srovnání s rychlostí samotných procesorů
- Přímá podpora překladačů
- Adresování (přístup k paměti)

# Mikroprogramování

CISC – složité instrukce

- Řídící část procesoru příliš rozsáhlá
  - Mikroinstrukce: Dekompozice na jednodušší instrukce
- Složitá instrukce == mikroprogram

Jednodušší návrh hardware

- Instrukce jsou *emulovány*

Je možno „snadno“ změnit instrukční sadu konkrétního počítače

⇒ *rodina počítačů* (IBM 360, 370, VAX, ...)

**Nevýhody:** příliš složité instrukce, stále složitější analýza instrukcí, zátěž zpětné kompatibility (v rámci rodiny)

# Zvyšování výkonu

- Rychlost hodin udává výkon procesoru
  - Omezeno aktuálními technologickými možnostmi
  - Nelze neomezeně zvyšovat
    - \* Závislosti mezi komponentami
    - \* Rychlost šíření signálu
- Řešení: **paralelizace procesů**

# Pipelining

Překrývání instrukcí *v různých fázích* rozpracovanosti

instrukce → 

1	—	2	—	3	—	4	—	5
---	---	---	---	---	---	---	---	---

 → výsledky

Tři základní oblasti:

1. Zpracování instrukcí
2. Přístupy k paměti
3. Výpočty v pohyblivé řádové čárce

# Pipelining II

Běžný rozklad instrukcí (pětiúrovňový pipelining):

**Instruction Fetch** instrukce je načtena z paměti

**Instruction Decode** instrukce je rozeznána (dekódována)

**Operand Fetch** jsou připraveny operandy (načteny z registrů a/nebo paměti)

**Execute** instrukce je provedena

**Writeback** výsledky jsou zapsány zpět (do registrů a/nebo paměti)

Jednotlivé instrukce jsou zpracovávány paralelně, s posunem o jednu fázi pipeline.

# Pipelining a paměť

- „Neviditelný“ pipelining
  - Předsunutí čtení (zápisu) z (do) paměti před vlastní instrukci pracující s daty
- „Viditelné“ pipelines
  - Explicitní instrukce, s přesně definovaným počtem cyklů do dokončení.
  - Např. Intel 80860



# Procesory – RISC

## Reduced Instruction Set Computer

- První RISC: CDC 6600 (Seymour Cray)
  - První polovina 60. let
- Podmínky vzniku RISC systémů
  - Zavedení vyrovnávacích pamětí (cache)
  - Dramatický pokles ceny a vzrůst velikosti hlavních pamětí
  - Lepší pipelining
  - Kvalitně optimalizující překladače

# RISC podmínky II

- Rychlost přístupu k paměti přestala být (hlavním) úzkým místem
- Velikost programu přestala být podstatná (i rozsáhlé programy se snadno vejdou do paměti)
- Problém: *zadržetí* (stall) při čekání na výsledek předchozí instrukce (v CISC příliš složité vazby)
- Není třeba složitých instrukcí (naopak); čitelnost assembleru přestává být podstatná

# Charakteristiky RISC

- Jednotná délka instrukcí
- Pečlivý výběr skutečně používaných instrukcí
- Jednoduché adresní módy
- Architektura Load/Store
- Dostatek registrů
- „Odložené“ skoky (delayed branches)

# RISC – pokročilý návrh

- Ideál RISC první generace:
  - Jedna instrukce každý tik hodin
- Dnešní realita:
  - Více jak jedna instrukce na tik

# Nové vlastnosti

- Superskalární
- Superpipeline
- (Velmi) dlouhé instrukce ((Very) Long Instruction Word, (V)LIW)

# Superskalární procesory

- Vícenásobné procesní jednotky
  - Aritmetické (ALU), Floating point (FPU) a další
- Příklady:
  - RS/6000, SuperSPARC a vyšší, Motorola 88110, HP PA 7100 a vyšší, DEC Alpha, MIPS R8000 a vyšší, Intel Pentium, IBM P4, P5

# Superskalární procesory – vlastnosti

- Paralelismus v hardware
  - Sekvenční programy
  - „Automatická“ paralelizace technickými prostředky
    - \* Současné načtení více instrukcí
  - Instrukce MADD (Multiply Add)
    - \* Operace  $X*Y+Z$

# Superpipeline

- Další zjednodušení obvodů
  - Rozsáhlejší dekompozice pipeline
  - Rychlejší provádění jednotlivých částí
- Výsledkem rychlejší výpočet
  - Jiná forma paralelismu
- Nazývány též *hluboké* (deep) pipelines



# VLIW

- Obdoba superskalárních (mnoho jednotek)
- Paralelizace pod kontrolou překladače
- Výhody:
  - Jednodušší instrukce
  - Není třeba složitý řídicí hardware
  - Potenciál pro nižší spotřebu energie
- Příklady:
  - Intel i860
  - Crusoe procesory firmy Transmeta
  - Ruské superpočítače (Elbrus)

# RISC – další rysy

- Obcházení registrů
- Přejmenování registrů
- Skoky
  - nulování operace
  - podmíněné přiřazení ( $a = b < c ? d : e ;$ )
  - vícenásobné „předčtení“ z paměti
  - buffer potenciálních cílů skoku
  - předpověď cíle skoku za běhu
    - \* statistická (předem dána)
    - \* dynamická

# ANDES

## Architecture with **N**on-sequential **D**ynamic **E**xecution **S**cheduling

- Východiska
  - Zpomalení způsobeno čekáním na data
  - Dynamický paralelismus
- Příklady
  - HP PA 8000, MIPS R10000, ...

# ANDES – Architektura

- Vícenásobné fronty instrukcí
  - celočíselná fronta pro celočíselné instrukce
  - adresní fronta pro operace Load/Store
  - fronta pohyblivé řádové čárky
- Nezávislá pipeline pro každou frontu
- Vlastnosti
  - Instrukce vybírány podle *připravenosti*
  - Není dodrženo pořadí instrukcí v programu
  - *Dokončení* instrukcí zajišťuje správné uspořádání

# ANDES – Spekulativní výpočet



# ANDES – Další vlastnosti

- Spekulativní skoky:
  - Výpočet pokračuje *předpovězenou* větví
  - Nečeká na výsledek instrukce
- Neblokující Load/Store
- Přejmenování registrů

# Paměť

- Organizace paměti:
  - Řádky a sloupce (matice)
  - Adresa má dvě části
  - *Page mode* – naráz čtena skupina souvisejících bytů

# Vlastnosti paměti

- Přístupová doba (memory access time)
  - Vystav řádek **plus** vystav sloupec **plus** vystav data
- Cyklus paměti (memory cycle time)
  - Určuje, jak často lze data číst
- Obé závisí na typu paměti (dynamická vs. statická)



# Virtuální paměť

- Fyzická vs. logická adresa
  - Více adresních prostorů
- *Translation Lookaside Buffer (TLB)*
  - Překlad logických adres na fyzické
  - Součást hardware
  - TLB výpadky (misses)
- Virtuální (Ne)použití v superpočítačích

# Vyrovnávací paměť

- Hit poměr
- Velikosti 4 KB–16 MB
- Organizace: řádky pevné délky, 16–128 bytů
- Typy:
  - Přímo adresovatelná (direct mapped)
  - Množinově (částečně) asociativní (set-associative)
  - Plně asociativní (fully-associative)

# Architektury

- *Harvard Memory Architecture*
  - oddělení paměti pro data a pro instrukce
- Programově ovládaná vyrovnávací paměť
  - řízení u (některých) superskalárních procesorů (DEC Alpha)

# Přímo adresovatelná vyrovnávací paměť

- Statické mapování
  - Každý řádek vyrovnávací paměti odpovídá předem určeným oblastem hlavní paměti
- Rychlé
- Jednoduché obvody
- Potenciálně neefektivní

# Plně asociativní vyrovnávací paměť

- Dynamické mapování
  - Asociativní paměť
  - Každý řádek vyrovnávací paměti zná adresy „svého“ bloku
  - Současný dotaz na všechny řádky
  - Výběr řádku pro zneplatnění
- Velmi efektivní
- Velmi složité obvody – drahé

# Částečně asociativní vyrovnávací paměť

- Množina přímo adresovatelných vyrovnávacích pamětí
- Kombinace lepších vlastností obou extrémních přístupů
  - Zpravidla 2 a 4 cestné

# Šířka toku dat

- **Bandwidth** = maximální propustnost paměťového systému
  - Měřena v bytech za sekundu

Propustnost není stejná mezi všemi komponentami

- Procesor – vyrovnávací paměť – hlavní paměť – externí paměť
- Zpoždění (Latence)
  - Doba mezi časem požadavku a časem přísunu dat
  - Zvláště významná pro přesun malých objemů dat

# Prokládaná (Interleaved) paměť

- Rozdělení na menší bloky
  - Následující adresy mapovány do různých bloků
  - Umožňuje okamžitý přístup
- Běžné dvou až osminásobně prokládané paměťové subsystémy
  - superpočítače mají vícenásobné prokládání
    - \* Příklad: Convex C3 s 256 násobným prokládáním
    - \* Hodiny 16 ns
    - \* Opakovaný přístup k témuž banku: 300 ns (téměř 20 násobné zrychlení)
- Vyšší latence
  - Odstíněna použitím pipeline



# Přeskládání přístupů k paměti

- Předchůdce ANDES
- Minimalizace následných přístupů do těchž banků paměti
- Kontrola závislostí Load a Store při běhu programu
- Příklad: Motorola 88110

# Procesor MIPS R8000

- Zaveden 1993
- Čtyřnásobná superskalární architektura, max 6 operací/cyklus
  - Zdvojená ALU, zdvojená FPU a dvě Load/Store jednotky
  - FPU s IEEE-754 standardní aritmetikou s nepřesným přerušením
  - 32 registrů (64 bit) pro celočíselné a 32 registrů (64 bit) pro float operandy
  - Podmíněné `move` instrukce (pro IF příkazy)
- Plně 64bitová architektura
  - 128-bit datová sběrnice
  - 40 bitová adresní sběrnice (max 1 TB fyzické paměti)
  - TLB dvoucestný, s 384 položkami

# MIPS R8000 (II)

- Vyrovnávací paměti
  - 16 KB I-cache (instrukce)
  - 16 KB D-cache (dvoucestná, pouze pro celočíselná data)
  - 2 KB branch prediction cache
  - 4 MB streaming cache (výpočty v pohyblivé čárce)

# MIPS R8000 – I-Cache

- Vyrovnávací paměť instrukcí
  - Přímá adresovatelná
  - 1024 položek po 128 bitech
  - Adresována i označena (tagged) virtuální adresou
    - \* Obchází TLB
  - tag RAM – 512 položek (pro každý řádek)
    - \* příznak
    - \* ASID (Address space identifier)
      - ASID rozlišuje shodné virtuální ale různé fyzické adresy
    - \* bit platnosti
    - \* dva bity oblasti

# MIPS R8000 – D-Cache

- Vyrovnávací paměť pro data
  - Přímo adresovaná
  - Dva paralelní přístupy
    - \* 2 load nebo jedna load a jedna store instrukce současně
    - \* Adresována virtuální, označena fyzickou adresou
    - \* Write-through protokol

# MIPS R8000 (IV)

## Srovnání vyrovnávacích pamětí

Parametr	I-cache	Branch	D-Cache	TLB
Umístění	IU	IU	IU	IU
Velikost	16 KB	2 KB	16 KB	
Položka	128 bit	16 bit	64 bit	
Počet položek	1024	1024	2048	384
Počet portů	jeden	jeden	dva	dva
Mapování	přímé	přímé	přímé	3-cestné
Index	Virtuální	Virtuální	Virtuální	Virtuální
Tag	Virtuální	N/A	Fyzická	N/A
Přístup	jeden cyklus	jeden	jeden	jeden
Šířka	128 bit	16 bit	64 bit	
Propustnost	1,2 GB/s	159 MB/s	1,2 GB/s	
Řádek	32 bytů	N/A	32 bytů	
Miss penalty	11 cyklů	3 cykly		

# MIPS R8000 (V)

## Rychlost provádění operací

<b>Celočíselné</b>	<b>Latence</b>	
Add, shift, logical	1	
Load, store	1	
Multiply	4 (6)	
Divide	21	(jmenovatel $\leq$ 15 bitů)
	39	(jmenovatel 16–31 bitů)
	73	(jmenovatel 32–64 bitů)

  

<b>Reálné</b>	<b>Latence</b>	<b>Zdržení</b>
Move, negate, abs value	1	1
Add, Multiply, MADD	4	1
Load, Store	1	1
Compare, cond. move	1	1
Divide	14 (20)	11 (17)
Square root	14 (23)	11 (20)
Reciprocal	8 (14)	5 (11)
Reciprocal sq. root	8 (17)	5 (14)

# Procesor MIPS R10000

- Zaveden 1996
- ANDES architektura, tři fronty
- Superskalární, 4 instrukce současně
  - 2 ALU a 2 FPU (neekvivalentní)
  - FPU s IEEE-754 standardní aritmetikou a přesným přerušením
  - 32 (64 fyzických) registrů (64 bit) pro celočíselné operandy,
  - 32 (64 fyzických) registrů pro float operandy
  - přejmenování registrů
- Plně 64 bitová architektura
  - 128 bit datová sběrnice, 40 bitová adresní sběrnice
  - TLB plně asociativní, 64 položek (zdvojených) velikost stránky 4 KB–16 MB



# MIPS R10000 (II)

- Vyrovnávací paměti
  - 32 KB I-cache (2-set associative)
  - 32 KB D-cache (dvoucestná, 2-set associative)
  - předpověď skoků (4 úrovně)
  - 1 MB L2 cache
- Neblokující instrukce load a store

# MIPS R10000 (III)

- Výpočetní jednotky
- 2 ALU
  - Společně
    - \* Součet, Rozdíl a Logické operace
  - Rozdílné
    - \* ALU1: skoky a operace posunu
    - \* ALU2: násobení a dělení (iteračně)
- 2 FPU (Další dvě jednotky (bez pipeline) pro dělení a odmocninu (iteračně))
  - \* FPU1: sčítačka
  - \* FPU2: násobička

# MIPS R10000 – Fronty

- **Celočíselná**

- 16 položek
- až 4 instrukce současně zapsány

- **Float**

- 16 položek
- až 4 instrukce současně zapsány
- nelze současně zahájit Divide a Square root instrukce
- MADD instrukce projde oběma FPU

# MIPS R10000 -Fronty (II)

## ▪ Adresní

- 16 položek (FIFO)
- instrukce spustitelné v libovolném pořadí
- zápis a vyjmutí musí být sekvenční (zajištěno FIFO bufferem)
- znovuspuštění instrukce při neúspěchu (cache miss, konflikt, závislost)

# MIPS R10000 (V)

## Rychlost provádění operací

<b>Celočíselné</b>	<b>Latence</b>	<b>Zdržení</b>
Add, shift, logical, branch	1	1
Load, store	2	1
Multiply (32 bit)	5–6	6
Multiply (64 bit)	9–10	10
Divide (32 bit)	34–35	35
Divide (64 bit)	66–67	67
Int to Float (32 bit)	4	1

---

<b>Reálné</b>	<b>Latence</b>	<b>Zdržení</b>
Move, negate, abs value	1	1
Add, Conversion	2	1
Load, Store	3	1
Multiply (64 bit)	2	1
MADD	4	1
Divide	12 (19)	14 (21)
Square root	18 (33)	20 (35)
Reciprocal sq. root	30 (52)	20 (35)

# Procesor UltraSPARC-I

- Zaveden 1987 (Sparc V9)
- Čtyřnásobná superskalární architektura
  - 2 ALU, FPU (2 instrukce), GRU (Grafika)
  - 32 FPU (64 bit) registrů
- 64bitová architektura; možnost volby little a big endianu
  - 128 bitová datová sběrnice, 41 bitů fyzická adresa, 44 virtuální adresa
  - 64 položek v TLB, stránky s 8 K, 64 K, 512 K nebo 4 MB
- Visual Instruction Set

# UltraSPARC-I (II)

- Vyrovnávací paměti
  - 16 KB neblokující D-cache
  - 16 KB I-cache (s predikcí skoku)
  - 0,5–4 MB L2 cache (propustnost 3,2 GB/s)
- Blokující load/store instrukce

# UltraSPARC-I – výpočetní jednotky

## ■ FPU

- Dělení a odmocnina samostatné (mimo FPU pipeline)
- 12 (22) cyklů pro jednoduchou (dvojnásobnou) přesnost
- neblokují pipelinované FPU instrukce
- přesná přerušení

## ■ GRU

- 16 a 32 bitové shlukované sčítání a boolovské instrukce
- 8 a 16 bitové násobení
- skládání a rozbor dat
- přímý přístup k (grafické) paměti obcházející D-cache
- přímá podpora „motion compensation“.