

Architektura počítačů

Milan Kolář

Ústav mechatroniky a technické informatiky

Literatura

- E-learningový portál FM: <https://elearning.fm.tul.cz>
- Hlavička, J.: Architektura počítačů. ČVUT FEL Praha, 2001.
- Skalický, P.: Přístrojové aplikace mikropočítačů. ČVUT FEL Praha, 2004.
- Pluháček, A.: Projektování logiky počítačů. ČVUT FEL Praha, 2000.
- Pinker, J.: Mikroprocesory a mikropočítače. BEN, Praha, 2004.
- Smékal, Z. – Sysel, P.: Signálové procesory. Sdělovací technika, Praha, 2006.
- Kubátová, H.: Struktura a architektura počítačů s řešenými příklady. ČVUT FIT Praha, 2013.
- Šimeček, I.: Moderní počítačové architektury a optimalizace implementace algoritmů. ČVUT FIT Praha, 2015.

2

Definice architektury

Architektura počítačů – metoda vytváření počítačových systémů z menších celků

Architektura je globální pohled na všechny podstatné vlastnosti počítačů, zahrnuje:

- **strukturu** – popis propojení jednotlivých funkčních bloků
- **organizaci** – popis dynamických interakcí funkčních bloků a řízení styku mezi nimi
- **realizaci** – popis návrhu a vnitřní struktury jednotlivých FB
- **funkci** – popis chování počítače jako celku

Znalost architektury má být prostředkem pro vytváření nových systémů, má být podkladem pro hodnocení kvality výsledku.

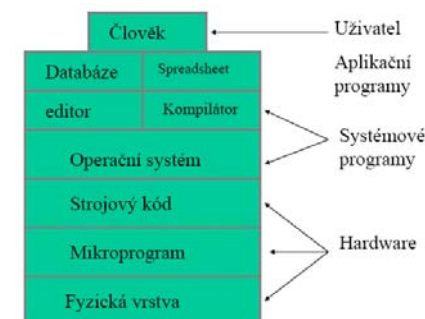
3

Počítač - definice

Stroj na číslicové zpracování informací (dat).

Zařízení, které provádí výpočty nebo řídí operace, které jdou popsat čísly nebo logickými výrazy (Oxfordský slovník)

Virtuální (rozšířený) počítač
– HW + OS



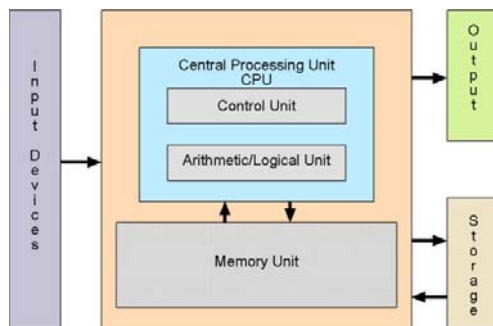
4

Definice architektury

CPU (Central Processing Unit) - procesor

ALU (Arithmetical and Logical Unit) – aritmeticko-logická jednotka

CU (Control / Central Unit) – řadič, řídicí jednotka



5

Počítač (elektronický číslicový)

I/O (Input / Output) Devices – vstupně / výstupní zařízení

Memory – paměť (operační)

Storage – paměť (archivní) – disk, karta (flash), páska

Vše propojují **sběrnice** :

- datové
- adresové
- řídicí

Šířka toku dat (šířka sběrnice) – počet bitů, které se po datové sběrnici přenášejí současně (může být rozdílná uvnitř a vně procesoru)

6

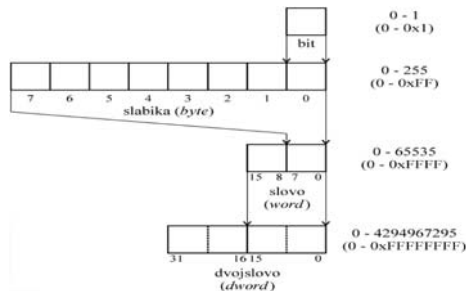
Základní pojmy

Bit (binary digit) – základní jednotka informace (0 nebo 1)

Slabika (byte) – uspořádaná skupina 8 bitů (nibble = 4 bity)

1 kB = 2^{10} B = 1 024 B, 1 MB = 2^{10} kB = 1 048 576 B

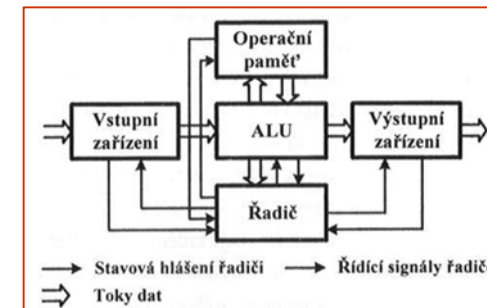
Slovo (word) – skupina slabik, které se v počítači zpracovávají jako celek



7

Von Neumannova architektura

představena v roce 1946



8

Von Neumannova architektura

Charakteristické vlastnosti lze shrnout do následujících bodů:

- 1) struktura počítače je nezávislá na typu řešené úlohy, počítač se programuje obsahem paměti;
- 2) instrukce a operandy jsou v téže paměti;
- 3) paměť je rozdělena do buněk stejné velikosti, jejich pořadová čísla se používají jako adresy;
- 4) program je tvořen posloupností elementárních příkazů (instrukcí), které se provádějí jednotlivě v pořadí, v němž jsou zapsány do paměti;
- 5) změna pořadí provádění instrukcí se vyvolá instrukcí podmíněného nebo nepodmíněného skoku;

9

Von Neumannova architektura

- 6) pro reprezentaci instrukcí i čísel se používají dvojkové signály a dvojková číselná soustava;
- 7) programem řízené zpracování dat probíhá v počítači samočinně (tok dat řídí řadič);
- 8) zpracování dat probíhá v tzv. diskretním režimu (během výpočtu nelze s počítačem komunikovat);
- 9) vstupy (resp. výstupy) jsou koncipovány jako datové zdroje (resp. výsledky) a jsou tedy přímo napojeny na ALU.

Nevýhody:

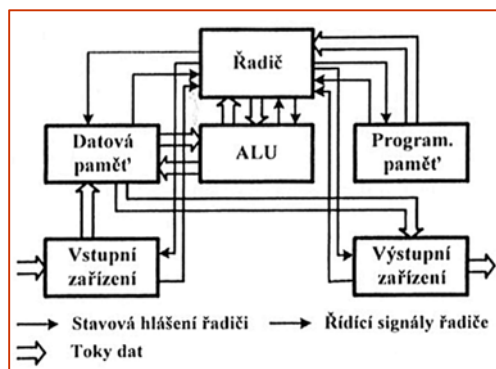
možnost mylně interpretovat data jako program.

10

Harvardská architektura

vznikla v roce 1943 (koncepte IBM Harvard MARK1)

někdy označována jako princetonská architektura



11

Harvardská architektura

Základní principy (rozdíly vůči von Neumannově architektuře):

- 1) paměť programu je oddělena od paměti dat
 - možnost ve stejném okamžiku načítat instrukci a přistupovat k datové paměti
 - datová a programová paměť mohou mít odlišnou organizaci
 - 2) oddělené sběrnice
 - 3) řízení procesoru je odděleno od řízení vstupních a výstupních jednotek (nejsou napojeny přímo na ALU)
- možnost rychlejšího zpracování většího objemu dat

12

Historie počítačů

30. léta 20. stol. – spíše mechanické kalkulátory
- 1938 – **Z1** (Konrád Zuse) elektromechanický
kolíčková paměť na 16 čísel, nespolehlivý
- 1940 – první plně elektronický počítač (releový)
ABC (Atanasoff - Berry Computer)
paměť – 60 slov (50 bitů) v podobě kondenzátorů
taktovací kmitočet 60 Hz, velká chybovost (0,001%)
- 1941 – **Z3** (programovatelný kalkulátor)
pracuje s čísly s plovoucí desetinnou čárkou
(14bitová mantisa, 7bitový exponent, znaménkový bit)
paměť pro 64 čísel, 2600 relé

13

Historie počítačů

- 1943 – **IBM ASCC MARK I** (Harvardská univerzita)
16 m dlouhý, hmotnost 5 tun
800 km drátových spojů
¾ mil. součástek



- 1946 – **ENIAC** (Pennsylvanská univerzita)
hmotnost 30 tun, zabíral 15 m², příkon 174 kW,
17460 elektronek, 1500 relé, 70000 odporů a 10000
kondenzátorů, pracovní frekvence 100 kHz.
Stroj nepracoval binárně, nýbrž dekadicky.

14

Generace počítačů

Každá generace je charakteristická svou konfigurací, rychlostí počítače a základním stavebním prvkem

Generace	0	1	2	3	4
Rok	1940	1951	1957	1964	1981
Prvky	relé	elektronky	tranzistory	SSI	LSI
Hlavní paměť		buben	ferity	ferity	LSI
Kapacita paměti		1 kB	10 kB	1 MB	10 MB
MIPS	0,001	0,01	0,1	1	10
Příklad	Mark I	Univac 1	IBM 7090	IBM 360	Intel 4004

15

Čtvrtá generace

- Její vývoj prožíváme dodnes;
- základem je centrální procesorová jednotka (CPU) označovaná jako mikroprocesor (vesměs z křemíku);
- IO LSI a VLSI (až 10^{18} tranzistorů na čipu);
- malé rozměry (technologie 45 → 32 → 22 nm);
- velká rychlost – využití paralelismu a zavádění programovacích prostředků, které paralelismus podporují;
- Již několik let se mluví o 5. generaci – orientace na využití umělé inteligence, přímý styk s uživatelem na úrovni přirozeného jazyka, textu a obrazů, distribuovaný HW.

16

Instrukční mixy

Nejstarší způsob hodnocení propustnosti číslicových systémů

Instrukční mixy jsou seznamy (tabulky) nejfrekventovanějších instrukcí ohodnocených pravděpodobnostmi jejich výskytu v rámci daného typu zátěže

Nevýhody:

- instrukční mixy jsou zpravidla závislé na konkrétním procesoru a architektuře – obtížná přenositelnost;
- frekvence použití jednotlivých instrukcí závisí subjektivně na programátorech, na druhu zpracovávaných úloh i na souborech vstupních zpracovávaných dat;
- dobu instrukcí ovlivňuje i operační systém.

Nejznámější mixy: Gibson, GPO-WU

17

Četnost výskytu instrukcí

Rozlišujeme mezi statickou a dynamickou četností:

Statická četnost reprezentuje výskyt jednotlivých instrukcí v programu (tak jak je uložen v paměti) – nevystihuje kolik jakých instrukcí procesor skutečně provedl

Dynamická četnost je frekvence instrukcí tak, jak je vykonává procesor

Rozdíl mezi statickou a dynamickou četností je dán zejména existencí příkazu skoku (větvení, cykly) – většinou používáme snadněji zjištělnou statickou četnost, ale vzniklá chyba není příliš velká (max. jednotky %).

18

Zkušební úlohy (benchmarky)

Zkušební úloha je vzorek zátěže, který má ověřit propustnost počítače v rámci určité aplikační oblasti

Výhodou je komplexnost – na jejich běhu se nepodílí jen procesor (jako u mixů), ale jsou ovlivněny i operačním systémem, překladačem, vstupy a výstupy, atd.

19

Skupiny zkušebních úloh

Reálné (přirozené) aplikace – převzaté z běžného provozu počítače (překladač, text. editor, komprimace) - problém s přenositelností (závislost na OS nebo překladači)

Upravené aplikace – úprava z důvodu přenositelnosti nebo zaměření se na určitou část výkonnosti systému

Jádra (kernels) – nejvýznamnější části skutečných programů pro otestování určitých rysů (např. Linpack, BDTI)

Demonstrační zkušební úlohy (toy benchmarks) – krátké programy (nezahrnují vliv paměťového systému – vejdou se do vnitřních pamětí)

Umělé (synthetic benchmarks) – vzniklé pouze za účelem zjišťování výkonnosti (v současnosti nejčastější) např. Whetstone (WHIPS), Dhrystone (DIPS)

20

SPEC

(Standard Performance Evaluation Corporation)

Eliminace slabín jedné zkušební úlohy vedla k vytvoření sad zkušebních úloh => SPEC - skládá se z reálných aplikací z různých vědeckých a inženýrských aplikací:

- pro osobní počítače (desktop benchmarks)
- pro servery (server benchmarks)
- pro vestavěné poč. systémy (embedded benchmarks)
- pro OS MS Windows



Vývoj: **SPEC95, SPEC2000, SPEC2006** (dále se člení na celočíselné výpočty a výpočty v pohyblivé řádové čárce)
více na www.spec.org

21

Systémy pro zpracování transakcí

Transakce je obecně jakýsi zásah do dat (uložených v paměťovém systému) na základě požadavku

Výkonnost systémů pro zpracování transakcí udáván v jednotkách „transakce za sekundu“ (TPS)

Není definována standardní transakce => nejednotné, velmi závisí na architektuře, na typu vstupů a výstupů, aj.

Pokusy o jednotné definování funkcí nad databází, např. simulace reálného bankovního provozu (testovací funkce Scan, Sort, Debit-Credit)

22

Monitorování výkonnosti počítačů

Na rozdíl od zkušebních úloh cílem monitorování je zjištění skutečného chování počítače v čase prostřednictvím hodnot několika předem zvolených stavových proměnných (např. stav procesoru, paměti, sběrnice, V/V zařízení)

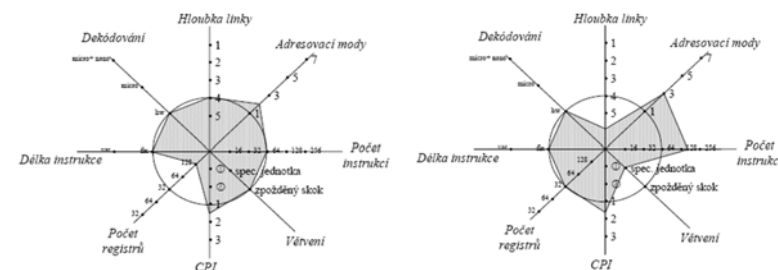
- *programový monitor* – snadná realizace, ale nepatrně zdržuje vlastní činnost počítače
- *obvodový monitor* – samostatný funkční blok, který neovlivňuje počítač a je s ním spojen sondami
- *kombinovaný monitor*

Monitory mohou sledovat nejen činnost OS, ale i požadavky uživatele nebo vyváženost konfigurace počítače.

23

Hodnocení vyváženosti architektur

Kiviatův graf – obecně n radiálních os (n vlastností s různými metrikami) – pro charakterizaci procesorů



Sparc

IBM RS/6000

24

Procesory I.

Milan Kolář
Ústav mechatroniky a technické informatiky

Procesor - definice

Procesorem se rozumí základní jednotka počítače, tj. logický automat pro zpracování informací, obsahující aritmetickou jednotku a řadič (počítač bez periferních zařízení a bez hlavní paměti); chování je definováno programem.

Mikroprocesor je malý procesor vyráběný technologií velké integrace, určen především na výpočty a logické funkce; je přednostně orientován na operace nad slovy.

Je-li mikroprocesor orientován pouze na logické bitové operace, mluvíme o **logickém (booleovském) mikroprocesoru**.

2

Kategorie procesorů

- Univerzální (Intel, AMD, ...)
- Grafické (Nvidia, ATI, ...)
- Signálové (TI, AD, ...)
- Aplikační (pro mobilní telefony, ...)
- Multimediální (TI, Mpack, ...)
- Speciální (šifrovací, kompresní, hrací, ...)

3

Dělení podle platformy (použití)

- Osobní počítače
- Notebooky (mobilní technika)
- Pracovní stanice
- Servery (datové, komunikační, tiskové, databázové)
- Palmtopy, handheldy, pocket PC, PDA
- Grafické karty
- Embedded (vestavěné) aplikace
- Herní konzole
- Automobilová technika
- Telekomunikační technika
- ...

4

Složení CPU

Procesor se skládá z:

- **řídící jednotky** CU (Control Unit) – řadič;
- **aritmeticko-logické jednotky** ALU (Arithmetic Logic Unit)
- **sady registrů** RS (Register Set), které uchovávají různé hodnoty během práce počítače (zápisníková paměť)
ALU + RS je někdy označuje jako **operační jednotka**;
- **programového čítače** PC (Program Counter) – často se uvádí jako jeden registr RS nebo součást řadiče;
- **vnitřní sběrnice** – řeší spojení mezi bloky CPU (typy - datová, adresová, řídící), od každého typu může být v architektuře i více sběrnic.

5

Řadič

Část procesoru řídící vykonávání operace a chod celého procesoru podle instrukcí programu.

Obsahuje **registr instrukcí**, který uchovává operační znak instrukce po dobu jejího vykonávání, a **dekodér instrukcí**, který dekoduje a generuje řídící signály pro procesor.

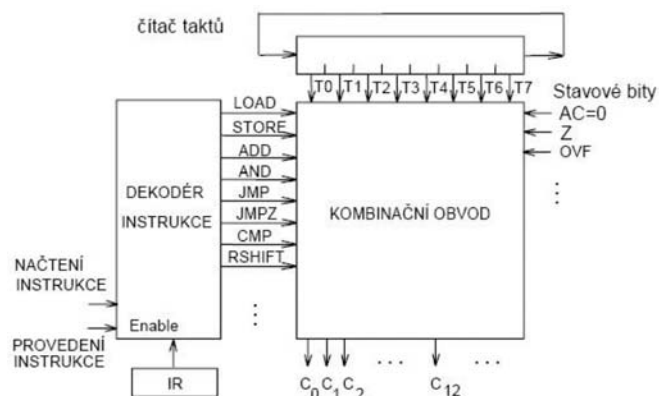
Existují 2 koncepce řadičů:

- 1) řadič je speciální sekvenční automat, který má čítač a dekodér (je dražší, ale rychlejší) – **obvodový řadič** (založen na kombinačních logických obvodech);
- 2) dekodování operačního znaku vykonává řídící paměť, ve které jsou mikroprogramy uloženy – **mikroprogramový řadič** (založen na výběru z paměti ROM).

6

Obvodový řadič

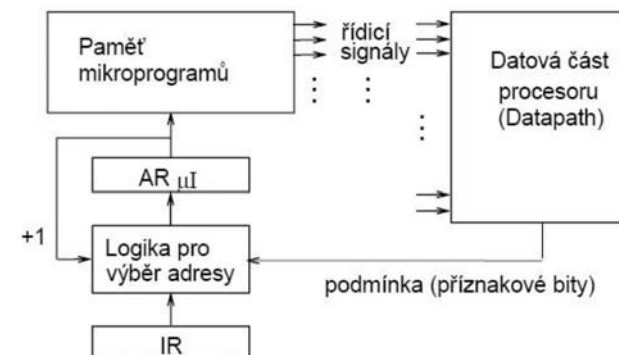
IR – instrukční registr (uchovává adresu další instrukce)



7

Mikroprogramový řadič

AR – adresový registr (uchovává adresu pro čtení/zápis z/do paměti)



8

Řadiče - rozčlenění

Na řadiči lze pohlížet jako na soustavu místních řadičů:

- řadič provedení instrukce ALU s pevnou řádovou čárkou,
- řadič provedení instrukce ALU s pohyblivou řád. čárkou,
- řadič přerušení,
- řadič I/O (řadič kanálu),
- řadič paměti,
- řadič DMA (Direct Memory Access),
- řadiče jednotlivých periférií
- ...

9

Aritmeticko-logická jednotka

ALU je část procesoru, ve které se provádějí všechny aritmetické (např. sčítání, násobení, bitový posuv, ...) a logické (logický součin, negace, ...) výpočty.

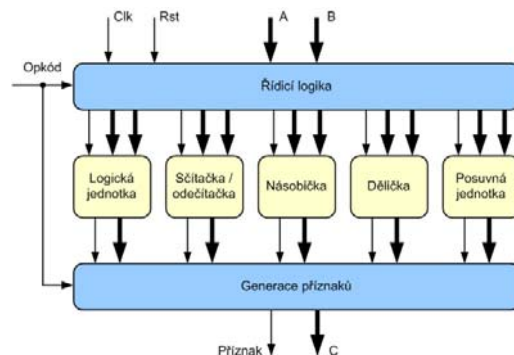
- HW řešen jako číslicový obvod (kombinační či sekv.)
- součástí obvykle bývá *příznakový (status) registr*

Procesor může mít více než jednu ALU - obvykle rozdělené na jednotky pro práci s celočíselnými operandy a jednotky pro práci s operandy v plovoucí řádové čárce (ty se někdy označují jako **FPU**, floating-point unit). Jednotlivé ALU pracují relativně nezávisle, takže procesor může v jednom hodinovém taktu provést více instrukcí současně.

10

Blokové schéma ALU

Liší se zejména možnostmi aritmetických operací, formátem operandů (pevnou a pohyblivou řádovou čárkou), mírou paralelizace, počtem hod. taktů na operaci apod.

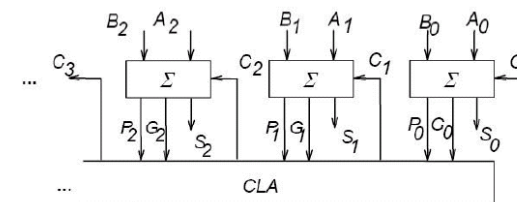
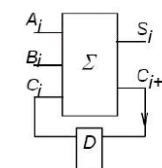


11

HW realizace sčítačky

Sčítačky:

- úplné
- neúplné (polosčítačky)
- sériové (SLO)
- sério-paralelní (SLO)
- paralelní (KLO) – s postupným přenosem
- s rychlým přenosem (CLA)

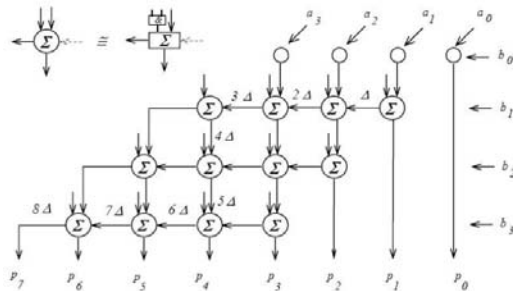


12

HW realizace násobičky

Násobičky:

- sériové (sekvenční) – obsahují SLO a násobení trvá více taktů (typické pro procesory CISC)
- paralelní (kombinační) – obsahují KLO a násobení se provede na jeden takt (typické pro procesory RISC)



13

Aritmetika FP

Sčítání, odečítání, porovnávání

- srovnat exponenty, provést danou operaci s mantisami

Násobení - sečíst exponenty, vynásobit mantisy

Dělení - odečíst exponenty, vydělit mantisy

Posuv - zvětšení/zmenšení exponentu

Normalizace

- posuv mantisy co nejvíce vlevo a úprava exponentu
- normalizace operandů nezaručuje normalizovaný tvar výsledku (nutno provést po každé operaci)

14

Instrukce

Specifikace jednoduché činnosti, kterou má provést technický prostředek (nejčastěji procesor).

Binární tvar instrukce se skládá většinou:

- z operačního kódu (opkód) - určuje typ instrukce
- z parametrů, které mohou být:
 - do instrukce zakódované konstanty,
 - označení registrů, odkud vzít hodnotu, příp. kam zapsat,
 - adresa paměti, odkud načíst hodnotu, příp. kam zapsat,
 - adresa paměti, kam má skočit další provádění programu.

Instrukce lze zapisovat v různých tvarech:

- strojový zápis ve formě binárních čísel (strojový kód),
- častěji zápis v jazyku symbolických adres (mov ah, 56h).

Soubor instrukcí tvoří program.

15

Mikroinstrukce

Jedna instrukce (příkaz) se v moderních procesorech nevykonává přímo, ale nejdříve se rozloží na jednodušší úkony, kterým se říká **mikroinstrukce**,
=> vzniká odlišná externí a interní instrukční sada.

Dochází ke zvýšení výkonu a přitom se zachovává zpětná kompatibilita procesorů.

Mikroinstrukce je nejčastěji uložena v jedné buňce paměti ROM a její vykonávání trvá jeden hodinový takt.

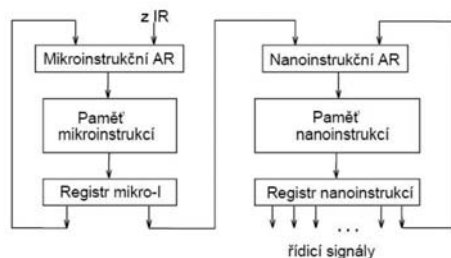
16

Nanoinstrukce

U některých procesorů má mikroinstrukce funkci adresy do sekundární řídicí paměti (nanopaměť), ze které se teprve čtou řídicí signály
⇒ dvouúrovňový paměťový systém.

Obsahy buněk paměti nanoinstrukcí označujeme jako **nanoinstrukce**.

Přínos: - snížení kapacity mikropaměti
- zjednodušení návrhu mikroprocesoru



17

Kódování instrukce

Kódování instrukce je kvalitativním popisem ISA

Každá instrukce musí mít zakódován

- druh operace (nejjednodušší – libovolný jedinečný kód)
- umístění operandů a výsledku (závisí na módech adresování, které ISA podporuje)

Obsahy buněk paměti nanoinstrukcí označujeme jako **nanoinstrukce**.

operace	operand 1	operand2	...	operand n / přímý operand / adresa
---------	-----------	----------	-----	------------------------------------

Operandem bývá zpravidla adresa registru, adresa paměti, nebo hodnota / konstanta.

18

Kódování instrukcí (pokračování)

U více operandů (GPR ISA) je první operand zpravidla cíl, další operandy jsou zdroje operací, poslední segment může být také přímý operand nebo adresa.

Délky jednotlivých segmentů mohou být různé:

- délka segmentu s typem operace musí být schopna zakódovat všechny druhy operací (nejčastěji 8 bitů);
- délka „registrového“ operandu závisí na počtu registrů;
- délka přímého operandu závisí na délce slova procesoru;
- délka adresy či segmentu adresy závisí na šířce adresové sběrnice procesoru a způsobu adresování.

19

Hlavní módy adresování

	Adresování	Příklad	Poznámky
Nultého řádu	Implicitní	OP Rx	Jeden z operandů (v případě ZO ISA všechny) je implicitním cílem, resp. zdrojem operace
	Přímým operandem	OP Rx, Num	Operand (číslo, adresa) je přímo obsažen v instrukci
	Registrové	OP Rx, Ry	Zdrojem operace (u GPR ISA také cílem) jsou registry
Prvního řádu	Nepřímým registrem	OP [Rx]	Hodnota registru Rx je ukazatelem na paměť (mem[Rx])
	Přímou adresou	OP Address	Adresa v instrukci je ukazatelem paměťové buňky (mem[Address])
Druhého řádu	Nepřímou pamětí	OP [Address]	Adresa v instrukci je ukazatelem na ukazatel v paměti (mem[mem[Address]])

20

Formáty kódování instrukcí

Proměnná délka kódu instrukce

- lepší hustota kódování (úspora paměti)
- každý operand může mít svůj adresní mód
- složité dekódování (vyžaduje čas nebo plochu čipu)
- typické pro CISC procesory
(např. délka kódu instrukce IA32 (x86) je od 1B do 13B)

Pevná délka kódu instrukce

- rychlé a jednoduché dekódování
- snazší implementace proudového zpracování (pipeline)
- adresní mód je součástí kódu operace
- omezená implementace adresních módů
- programy zabírají více paměti
- typické pro RISC procesory

21

Instrukční sada

Možno dělit podle různých pohledů:

Podle konceptu architektury – obecný popis organizačních, funkčních a provozních principů procesoru (ISA – Instruction Set Architecture)

Podle rozsahu instrukční sady:

- RISC, CISC

Podle implementace (mikro)architektury:

- x86, IA-32 (i386), IA-64, SPARC, ARM, ...

Podle míry paralelnosti:

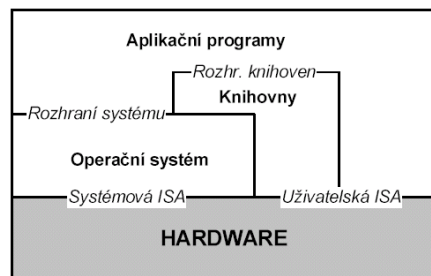
- subskalární, skalární, superskalární (např. VLIW)

22

ISA

Architektura souboru instrukcí – představuje v podstatě rozhraní mezi SW a HW počítače

Souhrn vlastností počítačového systému viděného z pohledu programátora v strojovém jazyce (koncept struktury, funkčního chování).



23

ISA

Musí být definováno:

- způsob kódování instrukcí (pořadí operací, operandů a adres),
- zacházení s operandem (způsoby adresování),
- výčet možných operací - instrukcí (aritmetických, logických, skoky, ...),
- způsob ukládání výsledku (střadač, registr, paměť, zásobník),
- datové typy a velikosti operandů,
- výběr následujících instrukcí (větvení) – varianty podmíněných skoků, volání a návraty z podprogramu, způsoby přerušení.

24

Typy ISA

- *střadačově (akumulátorově) orientovaná ISA* (Accumulator)
- *zásobníkově orientovaná ISA* (Stack)
- *ISA s univerzálními registry (registrová)* (GPR – General Purpose Register)

Díky zpětné binární kompatibilitě nelze ISA často měnit

ISA souvisí s vývojem počítačového HW, s jeho rostoucí hustotou integrace, zrychlováním a zlevňováním.

25

Střadačově orientovaná ISA

Implicitním operandem aritmeticko-logických instrukcí (zdrojovým i cílovým) je speciální registr uchovávající hodnotu předchozí operace – **střadač** (akumulátor).

Druhým (explicitním) operandem může být univerzální registr nebo paměťová buňka (\Rightarrow **jednoadresní** architektura).

Charakteristické vlastnosti:

- snadno kódovatelné a krátké instrukce
- rychlé přepínání kontextu (omezený vnitřní stav CPU)
- častý přístup do paměti
- problematická realizace paralelismu mezi instrukcemi

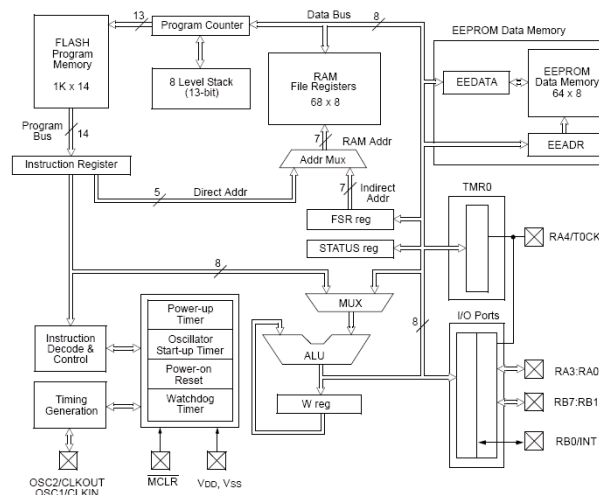
Vede na poměrně jednoduchý hardware (dekodér instrukcí).

Typický představitel – Intel 8080, 8051, 68HC05

26

Střadačově orientovaná ISA

Příklad:
PIC16F84



27

Zásobníkově orientovaná ISA

Zdrojem a cílem všech operací je sekvenční HW zásobník (registrové pole s ukazovátkem vrcholu – top).

Výhody: jednoduché a rychlé instrukce (bez operandů), vysoká hustota kódování (krátké programy), rychlá implementace či emulace.

Nevýhody: komplikovaný přístup k datům v paměti (díky sekvenčnosti zásobníku, obtížný paralelismus operací, nemožnost náhodného přístupu k lokálním proměnným).

Řadič musí řešit přetečení nebo podtečení HW limitovaného zásobníku (přesun dat mezi zásobníkem a hlavní pamětí - stack spilling); nekonečný SW zásobník v hlavní paměti.

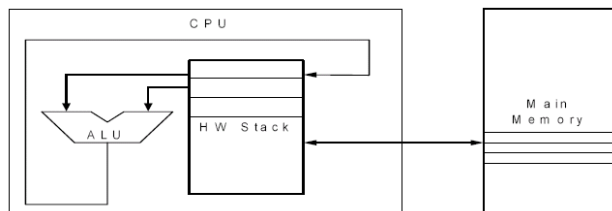
28

Zásobníkově orientovaná ISA

ZO ISA se používá výjimečně (používá se SW implementace - Java Virtual Machine).

Bezadresní - základní instrukce: PUSH (načtení dat z paměti do zásobníku), POP (přesun dat ze zásobníku do paměti) a operace ALU.

Někdy se označuje **MISC** (Minimal Instruction Set Computer).



29

Zásobníkově orientovaná ISA

Příklad – spočítat výraz: $a * a * (b + 4 * a) + 3 * 2$

```
PUSH a
PUSH a
PUSH b
PUSH 4
PUSH a
MUL      ; 4 * a
ADD      ; 4 * a + b
MUL      ; (4 * a + b) * a
MUL      ; (4 * a + b) * a * a
PUSH 3
PUSH 2
MUL      ; 3 * 2
ADD      ; 3 * 2 + (4 * a + b) * a * a
```

Výsledek zůstává na vrcholu zásobníku

30

ISA s univerzálními registry

- nejpoužívanější současná architektura (od 80. let min. století)

Základem je soubor velmi rychlých univerzálních registrů (GPR), které mohou být jak zdroji, tak cíli vykonávaných operací (mohou obsahovat mezivýsledky i proměnné)

Počet registrů se pohybuje od 8 do 128

Instrukce může mít 2–3 operandy (**2 až 3 adresní** architekt.)

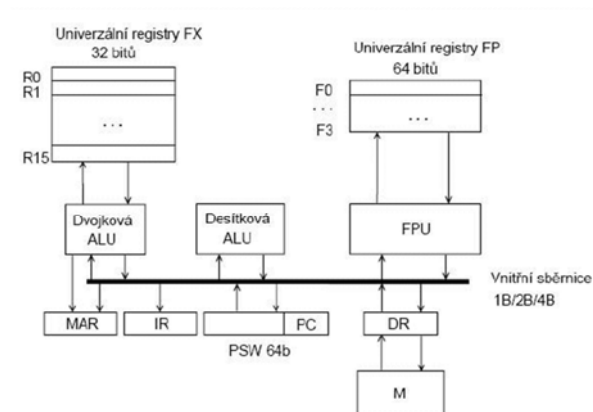
Výhody:

- registry jsou rychlejší než paměť – včetně cache
- přístup k registrům může být náhodný
- méně časté přístupy do paměti → urychlení
- snadná implementace paralelismu

31

ISA s univerzálními registry

Příklad GPR architektury



32

ISA s univerzálními registry

Nevýhody:

- limitovaný počet registrů
- složitý překladač optimalizující využití registrů
- proměnné v registrech nelze adresovat pomocí ukazatelů
- registry nemohou uchovávat složené datové struktury (záznamy, pole ...)
- přepnutí kontextu trvá delší dobu (ukládání více registrů)

Typický představitel – Intel x86 (do i386 označovaná „multiple accumulator ISA“ - víceřadačová)

33

ISA s univerzálními registry

Rozlišujeme varianty:

R-R ... oba operandy musí být v registrech
výhodou je jednoduché kódování instrukčního souboru, pevná délka instrukce, konstantní *CPI*, nevýhodou je vyšší počet instrukcí, typické pro RISC

R-M ... jeden operand může být v paměti
výhodou je přímý přístup k datům bez meziukládání, dobrá hustota kódu, nevýhodou jsou různé *CPI*, typické pro CISC

M-M ... oba operandy mohou být v paměti
úspora registrů, proměnná délka instrukce, paměť může být úzké místo, typické pro kompletní CISC (zastaralé).

34

Srovnání architektur ISA

Příklad výpočtu $C = A + B$

Štřadačová	Zásobníková	Registrová (R-M)	Registrová (R-R)
Load A	Push A	Load R1, A	Load R1, A
Add B	Push B	Add R1, B	Load R2, B
Store C	Add	Store C, R1	Add R3, R1, R2
	Pop C		Store C, R3

35

Srovnání architektur ISA (2)

Příklad výpočtu $R1 = R1^2 + R1^3$

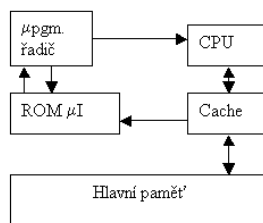
Štřadačová ISA		Registrová ISA	
Instrukce	Význam	Instrukce	Význam
LD R1	$A \leftarrow R1$	MUL R2, R1, R1	$R2 \leftarrow R1 * R1$
MUL R1	$A \leftarrow A * R1$	MUL R1, R2, R1	$R1 \leftarrow R2 * R1$
ST R2	$R2 \leftarrow A$	ADD R1, R1, R2	$R1 \leftarrow R1 + R2$
MUL R1	$A \leftarrow A * R1$		
ADD R2	$A \leftarrow A + R2$		
ST R1	$R1 \leftarrow A$		

36

CISC

Complex Instruction Set Computer

Dříve byly operační paměti výrazně pomalejší než procesory (doba přístupu byla několik taktů CPU), zpomalování výpočtů opakovaným načítáním instrukcí,
⇒ snaha rozšiřovat instrukční soubor,
⇒ mnoho složitých instrukcí používaných jen zřídka.



37

CISC

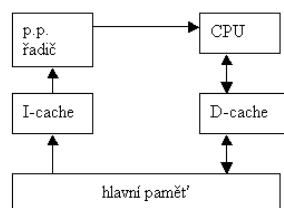
- proměnná délka instrukcí,
- zpracování instrukcí ve více strojových cyklech ($CPI \sim 5-10$),
- velký počet adresovacích módů,
- díky vysoké složitosti byl řadič navržen na principu paměti s mikroprogramy (ROM),
- řídicí obvody zabírají na čipu přibližně 60% místa,
- pro překlad programů bývá zpravidla jednodušší překladač,
- používá se zpravidla GPR ISA (varianta R-M, M-M),
- s postupem doby se začíná používat zřetěžené zpracování (zejména s rozkladem na mikroinstrukce).

38

RISC

Reduced Instruction Set Computer (použit poprvé 1974) počátkem 80.let první RISC procesory.

Snaha přesunout některé složité a zřídka používané CISC instrukce z mikroprogramů do programů (zvětšení počtu instrukcí v programu, ale snížení CPI).



(p.p. – pevně propojený)
(I-cache, D-cache ...
cache pro instrukce a pro data)

39

Základní rysy RISC

- malý počet relativně jednoduchých instrukcí (důležitý není až tak počet, ale jednoduchost) (asi 40–150),
- jednoduché instrukce umožňují vyšší frekvenci,
- implementováno proudové zpracování instrukcí - v každém taktu se většinou dokončí jedna instrukce ($CPI < 1,5$),
- instrukce mají většinou pevnou délku a malý počet formátů,
- řadič s pevnou logikou místo mikroprogramování (rychlé),
- řídicí obvody zabírají pouze 6–10 % místa (obvodový řadič),
- velký počet programově dostupných registrů (32–192),
- operace s daty pouze nad registry (2 zdrojové, 1 cílový),
- registry jsou víceúčelové (jednodušší překladače),

40

Základní rysy RISC (pokračování)

- přístup do paměti pouze pomocí instrukcí přesunu (mluvíme o architektuře L/S – instrukce Load a Store),
- malý počet adresových módů (3–5),
- ortogonální instrukční soubor (ve všech instrukcích, které používají registr procesoru jako zdrojový nebo cílový operand, lze použít libovolný registr),
- pro zvýšení účinnosti pipeline optimalizující kompilátor pro naplňování instrukcí,
- nejčastěji harvardská architektura.

Výkonnostní rovnice CPU: $T_{CPU} = IC \cdot CPI \cdot T_{clk}$
oproti CISC je u programů CPI výrazně nižší než nárůst IC
=> výkon roste, navíc je možné zvýšit T_{clk}

41

Post-RISC

Do této kategorie je možné zařadit většinu současných CPU

- někdy označovány jako CRISC (*Complex RISC*),
- kombinace CISC a RISC (navenek CISCové, ale vnitřní konstrukci mají RISCovou),
- instrukce trvají různě dlouhou dobu,
- instrukce se rozloží na jednoduché mikroinstrukce,
- proudové zpracování mikroinstrukcí,
- větší množství paralelních operací,
- spekulativní provádění instrukcí,
- nadále dochází k dalšímu rozšiřování instrukční sady, zaměřené hlavně do multimediální oblasti a grafiky,
- většinou se zachovává zpětná kompatibilita.

42

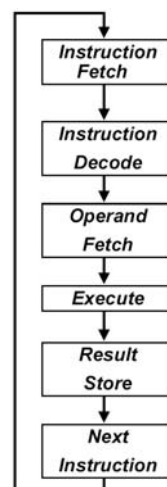
Instrukční cyklus

Definován jako skupina všech dob nezbytných k uskutečnění jedné strojové instrukce.

Periodu instrukčního cyklu lze rozdělit na několik fází:

načtení instrukce (Instruction Fetch – IF)

- CPU pošle na AB (Address Bus) adresu instrukce z PC (Program Counter);
- paměť předá na DB (Data Bus) obsah buněk;
- kód instrukce je z DB zapsán do IR (Instruction Register).



43

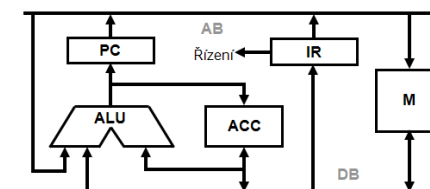
Instrukční cyklus (pokračování)

dekódování instrukce (Instruction Decode – ID)

- instrukční kód je řadičem dekódován a jsou generovány řídicí signály (zejména pro ALU),
- do PC se připraví nová hodnota;

načtení operandů (Operand Fetch – OF), bývá součástí ID

- IR připraví na AB adresu operandu v paměti M,
- M vystaví hodnotu operandu na DB (lze načíst do ACC);



44

Instrukční cyklus (pokračování)

vykonání operace (EXecution – EX)

- podle signálů z IR je v ALU vykonána operace (ACC může být jak zdrojovým, tak cílovým registrem);

obsluha paměti (Memory operation – MEM)

- výsledek ALU je použit jako adresa dat v paměti (nemusí být součástí instrukčního cyklu);

zápis výsledku (Write Back – WB), také Result Store

- IR vystaví na AB adresu M, kam má být zapsán výsledek (výsledek je na DB, může být přechodně v ACC);

Některé instrukce nevyužijí všechny fáze, příp. mohou mít jiné:

např. procesory CISC: IF-ID-AG-CA-EX-WB

(AG – Address Generation, CA – Cache Access, jeden operand v paměti)

Kombinační (jednotaktový) CPU

Model procesoru, který může uskutečnit instrukční cyklus

Model rozděluje paměť na datovou a programovou část

Současně provádí:

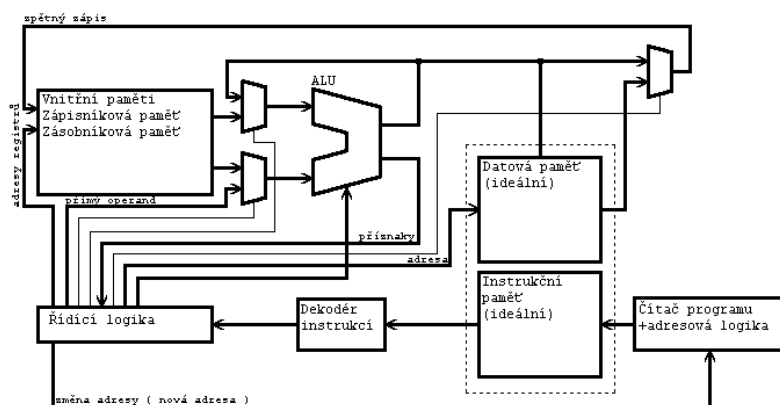
- adresaci zápisníkové paměti (pro čtení i zápis)
- nastavení sběrnice datových zdrojů
- vybuzení obvodů hlavní paměti (adresa, data, čtení, zápis)
- nastavení sběrnice zpětného zápisu
- přípravu nové adresy čítače

Doba vykonávání jedné instrukce je rovna součtu dob ustálení jednotlivých kombinačních obvodů ($CPI = 1$) :

$$T_{clk} = T_{set}^{IF} + T_{set}^{ID} + T_{set}^{EX} + T_{set}^{MEM} + T_{set}^{WB}$$

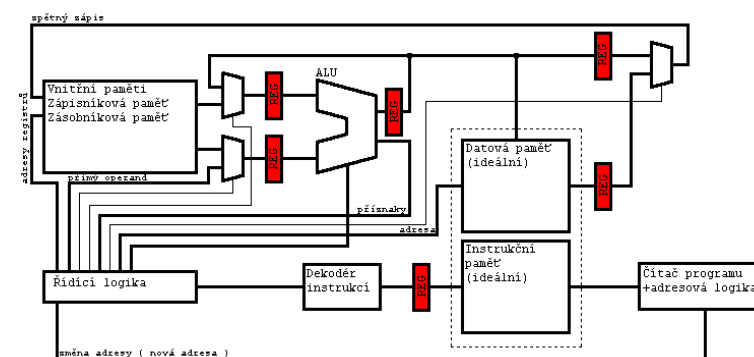
Kombinační (jednotaktový) CPU

Jednoduchý řadič, ale dlouhá doba periody instrukčního cyklu



Vícetaktový procesor

Vložení dodatečných registrů k oddělení jednotlivých fází instrukčního cyklu



Vícetaktový procesor

Rozdělení pomocí registrů umožňuje kontrolovat každou část procesoru odděleně a v důsledku toho některé fáze instrukčního cyklu vynechat

CPI vzroste méněkrát než kolikrát lze zvýšit T_{clk}

⇒ výkon je nepatrně vyšší než u jednotaktového CPU (~ 1,2x)

Vícetaktový procesor je univerzálnější, protože umožňuje implementovat i komplexní funkce např. pomocí mikroprogramování

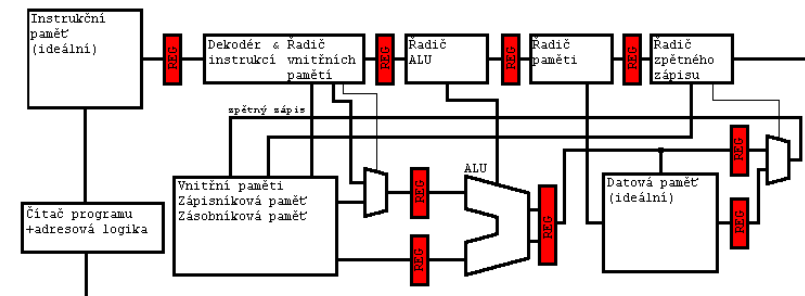
Jednotaktový i vícetaktový CPU jsou **plně sekvenční** (instrukce je dokončena než je započata následující)

Zvýšení výkonu sekvenčních CPU lze dosáhnout **zřetěžením** nebo **replikací** (opakováním) funkčních jednotek

49

Zřetěžený procesor

Řadič rozdělíme na menší řadiče jednotlivých fází instrukčního cyklu a mezi jednotlivé části vložíme registry



50

Zřetěžený procesor

Tím mluvíme o zřetěžené struktuře – *pipeline* a umožníme **proudové zpracování instrukcí (pipelining)**

Oddělením jednotlivých stupňů zvýšíme vytížení jednotlivých komponent CPU (nesníží se čas potřebný k vykonávání jedné instrukce, ani nesnížíme dobu ustalování KLO)

Nejvyšší vytížení je možné získat trvalým přísunem nových instrukcí ⇒ nejlépe instrukcí s jednotnou délkou

Není možné zpracovávat komplexní instrukce (musí se rozložit na sekvenci jednodušších)

T_{clk} je dána dobou ustálení nejpomalejšího KLO v CPU:

$$T_{clk} = \max \{ T_{set}^{IF}, T_{set}^{ID}, T_{set}^{EX}, T_{set}^{MEM}, T_{set}^{WB} \}$$

51

Zřetěžené zpracování informací

Typy zřetězení: zřetězení aritmetické (na úrovni ALU)
zřetězení instrukcí (na úrovni CPU)
zřetězení procesů (na úrovni počítače)

Předpoklady zřetězení:

- nepřetržitý přísun údajů
- operaci lze rozdělit na sekvenci nezávislých kroků (fází)
- trvání jednotlivých kroků by mělo být přibližně stejné

Počet stupňů v řetězu označujeme jako **hloubka zřetězení**

Doba průchodu instrukce všemi stupni řetězu je tzv. **latence řetězu (pipeline)**, tj. hloubka násobená dobou taktu

Zřetěžené zpracování může být **asynchronní** nebo **synchronní**

52

Zřetěžené (proudové) zpracování

Procesory jsou vesměs synchronní, řízeny hodinovým signálem.
 Procesor s jednoduchými fázemi, ale s větším počtem fází, může operovat na vyšší frekvenci.

Procesory s velkým počtem fází (2 až 20) častěji vznikají kvůli různým závislostem prázdná místa (tzv. bubliny), které snižují výkon;

- ⇒ vyšší frekvence CPU nezaručuje větší výkonnost (nelze porovnávat CPU různých architektur podle frekvence);
- ⇒ dlouhá pipeline s vysokou frekvencí může být neefektivní z hlediska spotřeby (Pentium 4 má P_{tot} až 100W).

53

Hazardy zřetěženého zpracování

Hazardy – narušení vykonávání programu (nezaměřovat s hazardy v číslicových obvodech):

- *strukturní hazardy* – kolize sdílených (omezených) prostředků (paměť, registry, funkční jednotky, sběrnice);
- *datové hazardy* – důsledek datových závislostí v programu
 RAW – read after write,
 WAR – write after read,
 WAW – write after write;
- *řídící hazardy* – vznikají při provádění řídicích instrukcí (např. podmíněné skoky – statisticky velmi časté (1/6))
 – čím větší je hloubka zřetězení, tím se zvyšují ztráty.

54

Hazardy zřetěženého zpracování

Řešení:

- *Statické* (ve fázi kompilace) – přeskládáním instrukcí nebo vkládání prázdných instrukcí NOP
- *Dynamické* (za běhu programu) – vkládání prázdných taktů (bublin) řadičem – pozastavení pipeline (stall)

Výkonnost pipeline závisí na počtu fází a četnosti pozastavování

55

Příklad

Určete CPI proudově pracujícího procesoru, kde je dána četnost hazardů a způsobené pozastavení:

- RAW hazardy nastanou v 10 % instrukcí, průměrné pozastavení 2 takty;
- WAW hazardy nastanou v 0,5 % instrukcí, průměrné pozastavení 4 takty;
- řídicí hazardy v důsledku provedených skoků nastanou v 9 % instrukcí, průměrné zdržení jsou 3 takty;
- strukturní hazardy v důsledku nedostupné jednotky či registrového zápisového portu nastanou v 3 % instrukcí, průměrné zdržení jsou 2 takty.

Vycházíme z předpokladu, že pro ideální proudově pracující procesor platí: $CPI = 1$ (1 instrukce za jeden strojový takt).

56

CPU a instrukční cyklus

Vývoj procesorů lze rozdělit do 3 fází:

- 1) **Subskalární (sekvenční) procesory** - tradiční sekvenční provádění instrukcí (celková doba provádění programu je dána aritmetickým součtem časů trvání jednotlivých instrukcí)
- 2) **Skalární procesory** – sekvenční vykonávání nahrazeno paralelním (buď *zřetězené zpracování* nebo *replikace*);
 - v jednom taktu byla k vykonání v některé jednotce dána nejvýš jedna instrukce ($IPC = 1$, Instruction Per Cycle)
 - vykonávání instrukcí probíhá překrytě (i paralelně);
 - dalším vývojem paralelismus navýšen i použitím několika zřetězených funkčních jednotek

57

CPU a instrukční cyklus

- 3) **Superskalární procesory** – paralelní vydávání i paralelní zpracovávání instrukcí ($IPC > 1$)

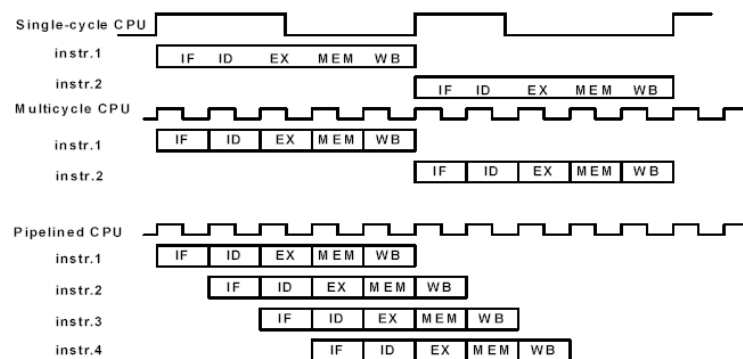
Plánování paralelního zpracovávání rozdělujeme na:

- statické* – paralelní vydávání instrukcí naplánovány kompilátorem (např. architektura VLIW);
- dynamické* – o paralelismu rozhodují technické prostředky za běhu programu (složitější).

Podobné trendy jsou uplatňovány v architekturách počítačů na různých úrovních abstrakce (mikrořadič, procesor, počítačový systém), v architekturách RISC i CISC.

58

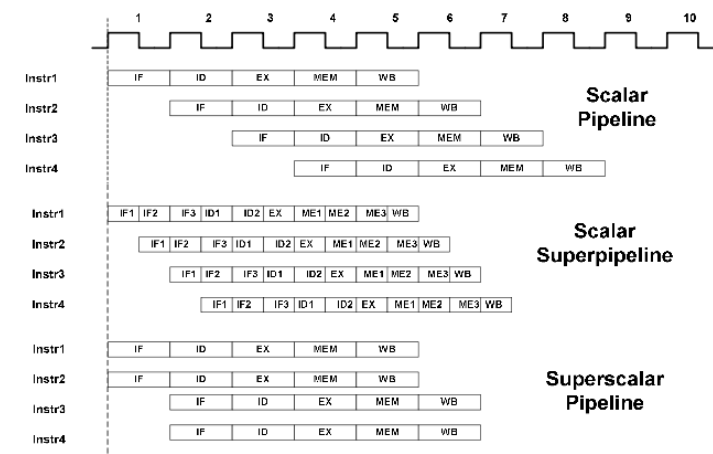
Porovnání typů procesorů



rozdělení i počet jednotlivých fází instrukčního cyklu je u různých procesorů různé

59

Zřetězené procesory



60

Zřetězené zpracování informace

Frekvence výstupu k -stupňového řetězce (počet výsledků za jednotku času):

$$(t_d \dots \text{zpoždění ve vyrovnávacím registru}) \quad f = \frac{1}{\max(t_i)_1^k + t_d}$$

Časový interval na vykonání k -stupňového řetězce n úloh:

$$T_k = [k + (n - 1)](\tau + t_d)$$

$k \cdot (\tau + t_d) \dots$ čas na zaplnění řetězce (kompletní vykonání první úlohy)

$(n - 1) \cdot (\tau + t_d) \dots$ čas na vykonání zbývajících $(n - 1)$ úloh
 $\tau \dots$ max. zpoždění stupně (bez oddělovacího registru)

Čas na vykonání nezřetězené úlohy: $T_s = k \cdot n \cdot \tau$

61

Zrychlení při zřetězení

Zrychlení výpočtu n úloh ve zřetězeném a nezřetězeném procesoru:

$$S_k = \frac{T_s}{T_k} = \frac{n \cdot k \cdot \tau}{[k + (n - 1)](\tau + t_d)}$$

Teoretické maximální zrychlení pro velmi velká n se blíží k , tzn. že lineárně zřetězeným procesorem je možno dosáhnout zrychlení blížící se až počtu stupňů tohoto procesoru (tedy $CPI \rightarrow 1$ pro $n \rightarrow \infty$ a pro $t_d \ll \tau$).

Zvyšování počtu stupňů k zvětšuje zpoždění vstup-výstup (vlivem přidavných registrů) a cenu (větší plocha čipu).

62

Zrychlení při superzřetězení

Superzřetězení (superpipelining) – jednotlivé fáze zpracování instrukce (IF, ID, ...) jsou dále děleny na určitý počet m kratších taktů (hloubka zřetězení se tak zvyšuje na řádově desítky i více)
 \Rightarrow zvýšení využití logiky a možnost zvyšování T_{clk}

Zrychlení vůči normálnímu zřetězení:

$$S_{mk} = \frac{T_k}{T_{mk}} = \frac{(k + n - 1) \cdot (\tau + t_d)}{(m \cdot k + n - 1) \cdot (\tau / m + t_d)}$$

Pro $n \rightarrow \infty$ a pro $t_d \ll \tau$ je $S_{mk} = m$

Je-li zpoždění registrů srovnatelné se zpožděním stupně

$\tau / m \approx t_d$ a pro $n \rightarrow \infty$, pak $S_{mk} = (m + 1) / 2$

63

Zřetězení, superzřetězení

Zřetězená linka nebývá plně využita \Rightarrow průměrné využití stupňů linky (dosažená účinnost):

$$E_k = S_k / k$$

Výsledky úkolu budou k dispozici u nezřetězeného zpracování v intervalech t , zatímco u zřetězené linky v intervalech:

$$T = \tau + t_d = t / k + t_d = T_{clk}$$

Doba zpracování 1 instrukce je:

- pro zřetězené zpracování: $CPI \cdot [t / k + t_d]$
- pro superzřetězené zpracování: $CPI \cdot [t / k \cdot m + t_d]$
- pro superskalární zpracování: $CPI \cdot [t / k + t_d] / IPC$
- pro superzřetězené i superskalární: $CPI \cdot [t / k \cdot m + t_d] / IPC$

64

Superzřetězení vs. superskalárnost

Trendem je kombinace obou technik – nejrychlejší

Při rovnosti $IPC = m$ vychází superskalární řešení rychlejší než superzřetězené.

Počet tranzistorů na čipu se zvyšuje rychleji než jejich rychlost
=> od paralelismu na úrovni bitů (8 až 64bitové CPU)
přecházíme k paralelismu na úrovni instrukcí (IPC se zvyšuje od 1 do 8) a dále k paralelismu na úrovni vláken a procesů.

Zvyšování IPC naráží na paralelizovatelnost algoritmů.

Při zvyšování m se výkon zvyšuje teoreticky lineárně (při $t_d \ll \tau$), ale jsme omezeni technologií.

65

Příklady

1) Jaká je dosažená účinnost zřetězené 4stupňové linky při zpracování 6 instrukcí ($t_d \ll \tau$), je-li diagram vytížení dán tabulkou („x“ značí čekání na mezivýsledek, „–“ značí nevyužitost) :

Čas	1	2	3	4	5	6	7	8	9	10
S1	i1	i2	i3	i4	x	i5	i6			
S2		i1	i2	i3	x	i4	i5	i6		
S3			i1	i2	x	i3	i4	i5	i6	
S4				i1	i2	–	i3	i4	i5	i6

2) Porovnejte zrychlení dvou ideálních instrukčních linek:
a) superzřetězené linky se 4 stupni, každý stupeň se 3 podstupni,
b) superskalární zřetězené linky se 4 stupni s četností vydávání 3 instrukcí/takt,
proti nezřetězenému zpracování $t = 100$ ns na jednu instrukci.
Uvažujeme počet instrukcí $IC \rightarrow \infty$, zpoždění registrů $t_d = 5$ ns.

66

Zrychlení při ztrátových cyklech

Předpokládejme, že $S_k = k$ (pro velký počet instrukcí).

Pokud některé instrukce nelze zřetězit, dostáváme obdobu Amdahlova zákona:

$$S_k = \frac{1}{(1-g) + g/k}$$

k ... počet stupňů řetězce;
 g ... část instrukcí, které lze řetězit.

Příklad:

Máme 6stupňovou zřetězenou linku. Jaké bude zrychlení linky, když 13 % instrukcí bude mít 4 ztrátové (čekací) cykly, 25 % instrukcí bude mít jeden ztrátový cyklus a zbývající počet instrukcí bude bez ztrátových cyklů?

67

Hyperthreading Technology (HT)

Přináší dva programové vstupy, takže je možné najednou zpracovávat dva toky instrukcí (jeden fyzický procesor se tváří jako dva procesory logické).

Ve skutečnosti zdvojen to, co určuje aktuální stav jednoho výpočetního procesu (registry, instrukční čítač)
- lze přepnout jediným hod. taktem,
- dodatečné prostředky zabírají asi jen 5% plochy čipu (ALU i L2 cache jsou sdílené)

Díky HT jsou výkonné jednotky procesoru vytěžovány větší měrou – optimálně lze využít u vícevláknových úloh (výkonový zisk až 30%), náročnější správa

Podporováno MS Windows XP, Linux (od jádra 2.4.17).

68

Hyperthreading Technology

HT byl prvotně implementován u jednoprocessorových systémů (od Pentia 4), u dvoujádrových architektur vynechána, u vícejádrových CPU opět implementována (u některých řad).

Ve víceprocesorových systémech propojuje jednotlivá procesorová jádra vlákny umožňujícími rychlejší paralelní zpracování instrukcí

- např. 4jádrový CPU má 8 logických procesorů.

Zpracování instrukcí mimo pořadí

Out of Order - metoda, při níž se instrukce vykonávají v jiném pořadí, než jak uvádí program uložený v operační paměti. Procesor si sám rozhodne a poskládá instrukce tak, aby byly zpracovány v co nejkratším čase při maximálním využití všech částí procesoru. Toto je důsledek toho, že ne všechny instrukce se dají zpracovávat současně a ne všechny se dají stejně rozpracovat. Kdybychom zpracovávali instrukce v původním pořadí, byly by některé části mikroprocesoru po určitou dobu nevyužity.

Přejmenovávání registrů

Register renaming - technika omezující nebo odstraňující závislosti následujících instrukcí, které pracují se stejnými registry procesoru. Během zpracování instrukcí je k dispozici sada dočasných registrů, které zastupují skutečné registry procesoru.

(Používají-li například dvě následující instrukce stejný registr, je pro každou z nich přidělen dočasný registr s kopií původního registru. Když druhá instrukce změní hodnotu registru, je původní hodnota stále ještě v dočasném registru pro potřeby první instrukce, takže druhá instrukce může být vykonávána mimo pořadí před první instrukcí.)

Spekulativní vykonávání instrukcí

(*Speculative execution*) – spočívá v odhadu vykonávání instrukcí dopředu. Instrukce se vykonávají v době, kdy je mikroprocesor méně vytížen, a jednotlivé výsledky zpracování se ukládají v mikroprocesoru, aby byly v okamžiku potřeby k dispozici.

Problém skokových instrukcí (5–20 % podle typu programu)

- zhruba v 5/6 případů se skok provede,
 - nečeká se na vyhodnocení podmínky,
 - v případě nesprávné predikce se výsledky zahodí,
 - v predikci mohou být další podmíněné skoky,
- => *úroveň (stupeň) spekulace* je omezena, aby „úklid“ po špatné predikci netrval příliš dlouho.

Mikrořadiče

Milan Kolář

Ústav mechatroniky a technické informatiky

Mikropočítač (μP)

Obecný (PC, PDA, server, superpočítač)

- samostatný (sám o sobě) nebo oddělený,
- důraz na HMI (Human-Machine Interface), tj. rozhraní člověk-stroj nebo na komunikační připojení.

Embedded (vestavný, zabudovaný, vložený, řídicí)

- do systému (zařízení, stroje, přístroje), jednoúčelový?
- dedikovaný počítač pracující v reálném čase,
- důraz na M2M (Machine-to-Machine), tj. rozhraní k obsluhovanému objektu (snímače, akční členy),
- pro uživatele není μP přímo viditelný,
- program se většinou vyvíjí na jiném systému.

Mikroprocesor

Jednoobvodový (jednočipový) procesor, vyráběný technologií velké integrace

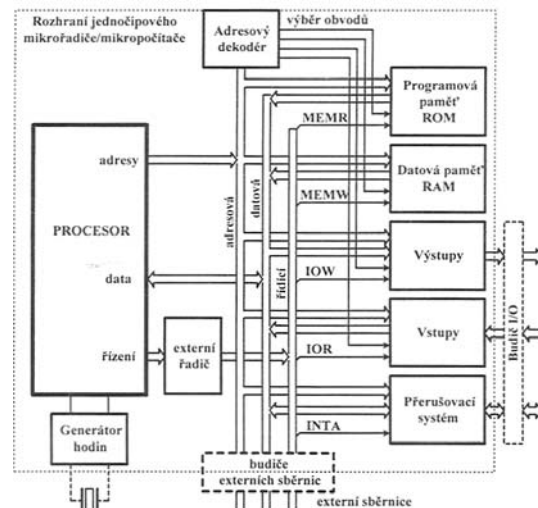
- *univerzální*
 - do osobních počítačů, serverů, pracovních stanic
 - spíše CISC, větší šířka sběrnic (32/64bit.), velké taktovací frekvence (~ GHz), velký příkon;
 - do embedded mikropočítačů
 - spíše RISC, šířka sběrnic 8–32 bitů, integr. periferie, frekvence (stovky MHz), často bateriové napájení;
- *signálový* (DSP – Digital Signal Processor)
 - pro zpracování signálů (filtry, FFT, modulace),
 - speciální architektury i instrukce;
- *grafické, multimediální, ...*

Mikrořadič (microcontroller – μC)

jednoobvodový (jednočipový, monolitický) μP (mikroprocesor doplněný pamětmi a periferiemi),

- RISC architektura (převážně jednocyklové instrukce, vyšší taktovací frekvence - např. 12 až 40 MHz),
- omezený soubor instrukcí (35–130),
- datová sběrnice relativně malá (8, 16, max. 32 bitů),
- relativně nízká cena (cca od 40,- Kč),
- široký rozsah napájecího napětí (od 2.7 V do 6 V),
- technologie CMOS,
- střadačová nebo registrová ISA (příp. kombinace).

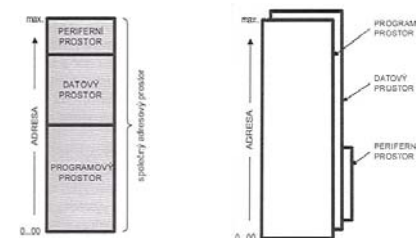
Blokové schéma mikrořadiče



5

Paměťový prostor

- rozsah adres: u 8bit. $\mu C \sim 12-16$ b; u 16bit. $\mu C \sim 16-20$ b
- *vnitřní* nebo *vnější* (jsou-li vyvedeny sběrnice),
- *společný* nebo *vícenásobný* (mohou existovat stejné adresy; přístup je odlišen řídicími signály),
- u jednodušších μP jsou do datové paměti mapovány všeobecné registry.



6

Příznakové bity (Flags)

Vyjadřují výsledky poslední operace, pokud tato operace příznaky mění (zvláštní výstup z ALU)
- využívají se pro větvení programu

Příznakový registr (PSW – Program Status Word) zpravidla obsahuje:

- **C** (CY, carry) – přenos z MSB
někdy zvlášť **B** (BO, borrow) – výpůjčka při odečítání
- **Z** (zero) - nulovost operace
- **S** (sign), **N** (negative) – záporné znaménko
- **V** (OV, overflow) – přetečení čísla se znaménkem, dělení 0
- **H** (half-carry), **AC** (auxiliary carry) – poloviční (pomocný) přenos – při součtu, mezi 3. a 4. bitem
- **P** (parity) – parita, tj. sudý nebo lichý počet jedniček

7

Instrukční sada

Instrukce se skládá z operačního kódu (vždy) a příp. z adres operandů nebo z přímých dat
- může mít různý počet slov

Skupiny instrukcí:

- *přesunů dat* (zdroj → cíl)
- *aritmetické* (+, -, komparace)
- *logické* (NEG, AND, OR, XOR, posuny, rotace)
- *řídící* (nepodmíněné a podmíněné skoky, volání podprogramů)
- *speciální* (I/O, systémové – NOP, HALT, změna stavových bitů)

8

Vstupy a výstupy mikrořadičů

Vstupy a výstupy jsou obecně organizovány do 8bitových bran (portů), některá z bran může být neúplná;

většinou bývají obousměrné – nutno určit směr.

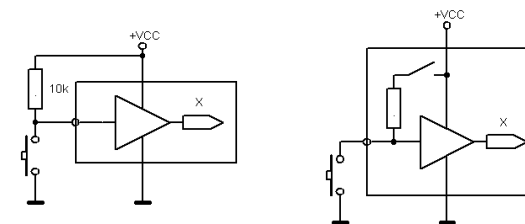
Výstupy:

- jednočinný výstupní budič s otevřeným kolektorem (vstup možný při nastavení log.1 na výstup),
- dvojčinný výstupní budič třístavový.

9

Vstupy mikrořadičů

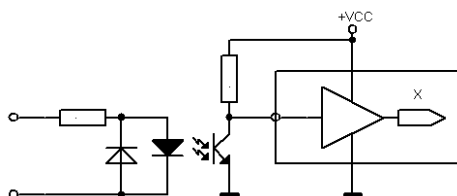
- s pull-up rezistorem (vnější nebo vnitřní)
- s hysterezí (Schmittovy)
- s pull-down rezistorem (příliš se nepoužívá)
- galvanicky oddělený (s optronem)
- nezapojen (NC – no connect), třetí stav



10

Galvanicky oddělený vstup

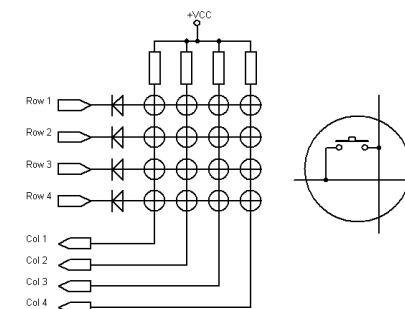
- pro logické (číslicové) signály,
- chrání vstupy mikrořadiče před poškozením,
- může pracovat na jiném potenciálu (rozmezí 4–30 V),
- je možno navrhnut i pro buzení střídavým napětím.



11

Maticová klávesnice

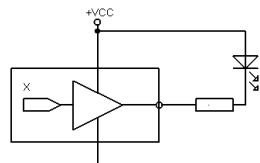
Při připojení více kontaktů – nutno programově scanovat (multiplexovat) – postupně nastavovat řadu po řadě do log.0 (ostatní jsou v log.1) a číst stavy na sloupcích (diody zabraňují zkratu při stisku dvou tlačítek v jednom sloupci).



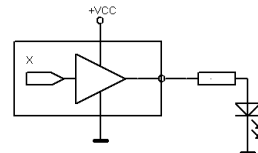
12

Připojení LED

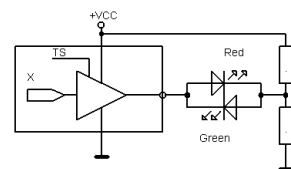
Obdobně je možné zapojit optotriaky, optotriaky apod.



svítí při log.0



svítí při log.1

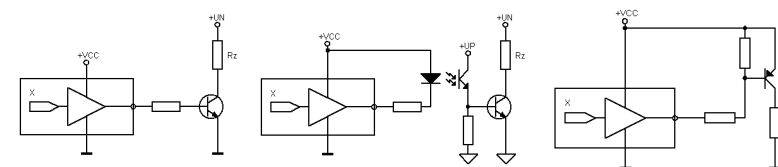


dvojitá LED – při log.0 svítí zelená, při log.1 červená, ve třetím stavu žádná

Připojení větších zátěží

Zátěž vyžadující vyšší proudy nebo napětí, např. relé, stykač, reproduktor, elektromagnet, motorek, odporové topení, žárovka, ...

- tranzistorové spínače (možné i Darlingtonovo zapojení)
- použití optonů (omezení rušení, ochrana μC)
- možnost napájení zátěže z napájecího zdroje μC



Připojení displeje

Jednotlivé segmenty spínáme přímo nebo přes budiče

Většinou se snažíme redukovat počet vývodů

- BCD dekodér (nelze zobrazit libovolné znaky),
- komunikace přes sériové rozhraní,
- multiplexní řízení.



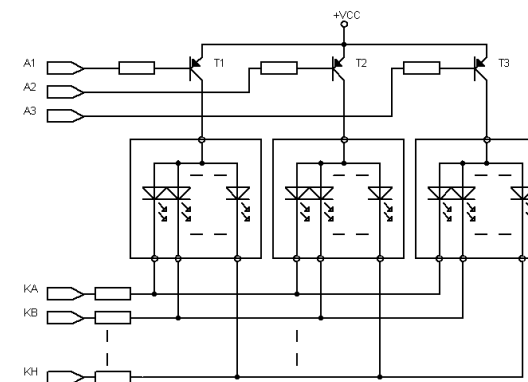
7 segmentů

14 segmentů

matice 5x7

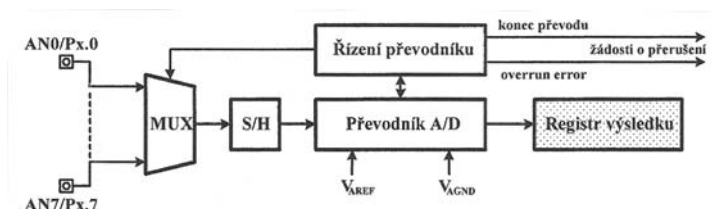
Multiplexní řízení displeje

Cyklicky rozsvěcíme jednotlivé znaky



A/D převodníky (ADC)

- rozlišení 8, 10, výjimečně 12 bitů,
- většinou na principu postupné aproximace,
- rychlost převodu max. 200 kS/s,
- vstupní analogový multiplexer, vzorkovací obvod S/H (Sample/Hold),
- referenční napětí: napájecí, externí, interní obvykle 2,5 V.



17

Další periferie

Analogový komparátor – 2 vstupy přivedeny na interní OZ, výstup čten v SFR, příp. aktivuje přerušení nebo zaveden na výstup
- s vnějšími obvody umožňuje realizovat ADC s dvojitou integrací.

D/A převodník (DAC) – až u novějších typů (dříve jen PWM s dolní propustí nebo externí)
- nejčastěji 8 nebo 12 bitů.

Obvod reálného času (RTC – Real Time Circuit) – obvod se záložním bateriovým napájením a krystalem (32768 Hz)
- z jednotlivých registrů se čtou časové údaje (sec až roky).

18

Přehled mikrořadičů

Mikroprocesory pro embedded systémy lze dělit na 2 kategorie:

- **nezávislé procesorové rodiny** (produkované více než jedním výrobcem) – liší se zejména množstvím periférií, velikostí interní paměti, frekvencí;
– např. řada Intel 8051, řada ARM, řada MIPS aj.
- **procesorové rodiny jednoho výrobce** – závislost na jediném výrobci, většinou mají určitou unikátní vlastnost;
– řada PIC (Microchip), řada MSP430 (TI), řada Freescale, řada Renesas aj.

19

AVR8 (Atmel)

- RISC koncepce, 8bitová DS;
- 90 až 131 instrukcí (i násobení 8x8bitů), 16bitové instrukce;
- programová paměť: Flash 1–128 kB;
- datová paměť: SRAM 128 B – 8 kB, EEPROM 64 B – 4 kB;
- 32 registrů (na zač. SRAM), lze spojovat do dvojic (X,Y,Z);
- 5 adresovacích módů, 10 zdrojů přerušení;
- Power-down mód (odběr 1 až 100 μ A);
- Idle mód (chod naprázdno, odběr 1 až 5 mA) – odpojen CPU, oscilátor a periferie pracují;
- periferie (LCD řadič, JTAG, SPI, UART, PWM, ADC, RTC);
- v současnosti i výkonné AVR32.

20

Řada x51 (Intel)

- navazuje na x48, stále používané s mnoha klony, 8bitové,
- rysy harvardské i von Neumanna (oddělená paměť, ale stejný formát instrukcí a dat, přenos po stejné sběrnici),
- velký počet instrukcí (cca 250) – 58 % jednocyklových, 40 % dvoucyklových, násobení a dělení 4 cykly,
- původně instrukční cyklus $12 T_{CLK}$ (později 6 nebo 4),
- SRAM 128–256 B, paměť kódu 4–64 kB,
- méně univerzálních registrů,
- 2 až 7 portů, 5 zdrojů přerušení (dvě priority),
- mnoho periferních obvodů (čítače, USART, Capture a Compare Unit, PWM, ADC).

21

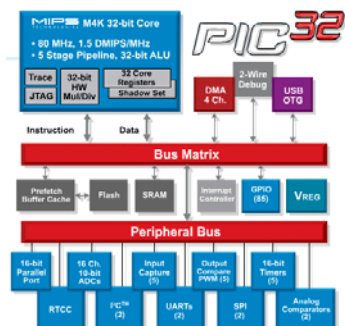
Řada 68HC908x (Motorola)

- navazují na řadu 68HC05/08, velmi rozšířená,
- von Neumannova architektura, 8bitová;
- Flash 2–32 kB, SRAM 32 B – 2 kB, EEPROM až 512 B;
- 134 instrukcí;
- instrukční cyklus trvá 2 až $6 T_{CLK}$;
- málo registrů, mnoho způsobů adresování;
- 5 až 8 portů; propracovaný systém přerušení (ukládají se i systémové registry);
- velká variabilita periférií (i relativně speciální – DTMF, LCD řadič, generátor textu do TV signálu, ADC, ...).

22

PIC32 (Microchip)

- uveden v roce 2007, 32bitový CPU;
- výkon 1,5 DMIPS/MHz, frekvence 80 MHz;
 - 5úrovňová pipeline, 32 registrů CPU;
 - HW násobička/dělička, 4kanálový DMA;
 - Flash 32–512 kB, RAM 8–32 kB.



23

Řada ARM

- vyvinuté firmou ARM Limited (Advanced RISC Machine)
- postupně verze ARM1 (1984) až ARM11, pak Cortex;
 - označování ARM ISA: ARMv1 ... ARMv8 (nepřehledné);
 - 32bitová architektura RISC, objevují se již 64bitové verze ;
 - výborný poměr výkon/spotřeba (pro bateriové aplikace);
 - nyní harvardská architektura (do verze ARM7 s ARMv4 spíše von Neumann – společný adresní prostor, 1 sběrnice);
 - periferie: Ethernet, ADC, UART, CAN, SPI, I2C, PWM,...
 - mnoho klonů (firmy Luminary Micro, ST, Actel, NXP).
 - v současnosti nejpočetnější skupina procesorů (desítky mld. CPU)

24



Řada ARM Cortex-Mx

založena na ISA ARMv7, 32bitové, pro DPS aplikace

ARM Cortex-M3:

- výkon 1,25 DMIPS/MHz;
- 32bitová AS (až 4GB RAM);
- 3úrovňová pipeline s predikcí větvení;
- instrukce 32bitového násobení i dělení (1-12 hod. cyklů);
- spotřeba 0,19 až 0,085 mW/MHz.

ARM Cortex-M8:

- výkon 2,0 DMIPS/MHz, integrovaná L2 cache;
- duální 13úrovňová pipeline (superskalární CPU);
- frekvence 800 – 1000 MHz;
- spotřeba do 300 mW.

25



Řada ARM Cortex-Ax

založena na ISA ARMv7, 32bitové, pro vysoce výkonné aplikace (netbooky, mobilní zařízení, multimédia)

- různé varianty – A5, A8, A9, A15;
- i více CPU jader na čipu (1 – 4);
- frekvence od 0,5 – 2,5 GHz;
- výkon 1,6 – 2,5 DMIPS/MHz na 1 jádro;
- spotřeba 0,12 až 0,08 mW/MHz;
- energetická efektivita 5 – 20 DMIPS/mW;
- až 13úrovňová pipeline;
- podpora množství OS (různé distribuce Linuxu, Symbian, Windows CE).

26



Řada ARM Cortex-A5x

Nová řada založená na ISA ARMv8, 64/32bitová architektura, opět nárůst výkonu oproti řadě Ax, spotřeba se nezvyšuje;

- různé varianty – A53, A57, A72;
- i více CPU jader na čipu (1 – 4);
- frekvence cca 2,0 – 2,5 GHz;
- až 18úrovňová pipeline;
- technologie 28 nm (20 nm).

27

Paměti

Milan Kolář
Ústav mechatroniky a technické informatiky

Pojmy a dělení

Paměť – zařízení, které slouží pro uchování informací (konkrétně binárně kódovaných dat).

Registr – velmi rychlé paměťové místo malé kapacity (jednotky bytů) umístěné většinou uvnitř procesoru počítače.

- *Neadresovatelné paměti* (LIFO, FIFO) – nezádává se u nich, kam zapisovat či odkud číst.
- *Adresovatelné paměti* (RAM, ROM) – paměť bývá rozdělena na **buňky** určité velikosti, z nichž každá je jednoznačně identifikována svým číslem. Toto číslo se nazývá **adresa** paměti a velikost takovéto buňky, která má svou vlastní adresu, se označuje jako nejmenší adresovatelná jednotka.
- *Obsahem adresovatelné paměti* (CAM) – asociativní.

2

Dělení pamětí

Podle materiálu a fyzikálních principů:

- feritové (zastaralé),
- magnetické (informaci uchovává směr magnetizace),
- magnetooptické (světlo mění magn. vlastnosti materiálu),
- polovodičové (nejpoužívanější),
- optické (založeny např. na odrazu světla).

3

Dělení pamětí

Paměť s přímým přístupem: paměť, která dovoluje přistoupit okamžitě k místu s libovolnou adresou
RAM (Random Access Memory);

Paměť se sekvenčním přístupem: paměť, u které je nutné při přístupu k místu s adresou n nejdříve postupně přečíst všechna předcházející místa (0 až $n-1$)
např. magnetické pásky;

Dělení pamětí z hlediska čtení a zápisu:

- *RWM* (Read Write Memory) - RAM: paměť určená ke čtení i zápisu dat;
- *ROM* (Read Only Memory): paměť určená pouze ke čtení dat.

4

Typy RAM

DRAM (Dynamic RAM) – dynamická RAM (nejběžnější);

SDRAM (Synchronous DRAM) – rychlé paměti synchronizované systémovými hodinami;

SRAM (Static RAM) – statická paměť založená na KO (varianty synchronní a asynchr. se systémovými hodinami);

FPM RAM (Fast Page Mode RAM) – logika paměti předpokládá, že další požadavek na čtení bude ležet v sousedství;

EDO ROM (Extended Data Out RAM) – vylepšení FPM;

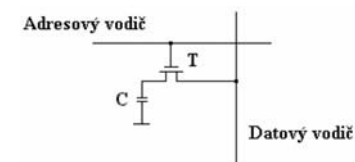
RDRAM (Rambus DRAM) – vylepšení od firmy Rambus.

DRAM

Dynamická RAM – tvořena tranzistorem s parazitním kondenzátorem

- buňka zabírá malou plochu,
- adresa je časově multiplexovaná (nejprve řádek a potom sloupec) \Rightarrow úspora vodičů (menší plocha),
- buňky je třeba každé cca 4 ms obnovovat (refresh).

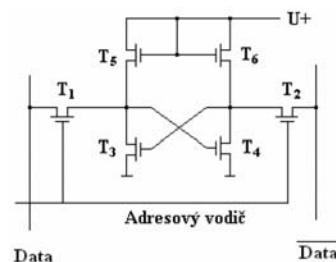
Čtení/zápis probíhá tak, že se přečte a vymaže celý řádek, uloží se do vyrovnávací paměti a pak se zapíše celý řádek z vyrovnávací paměti zpět do paměti.



SRAM

Statická RAM – buňka tvořena bistabilním klopným obvodem, který je relativně složitý \Rightarrow menší kapacita, ale rychlé.

Při zápisu i čtení – na adresový vodič log.1, T1 a T2 se otevrou, zapisuje-li se log.0 (Data = 1), T4 se otevře, T3 zavře; čteme-li pak tuto hodnotu, Data za T2 jsou v log.0 (negace).



Typy ROM

ROM – buňka je představována pojistkou nebo el. odporem (naprogramována výrobcem);

PROM (Programmable ROM) – programuje se v programátoru přepalováním propojek;

WORM (Write Once – Read Many) – optický princip;

EPROM (Erasable PROM) – informace v podobě náboje hradla tranzistoru MOSFET; lze celé smazat (UV zářením) a opět naprogramovat (pouze omezený počet);

EEPROM (Electrically EPROM) – maže se el. impulsy po buňkách;

Flash-EEPROM – buňku tvoří 1 unipolární tranzistor se dvěma vzájemně izolovanými hradly (Control a Floating Gate), programuje se po blocích, levnější než EEPROM.

Rozdělení pamětí

Vnitřní (primární): paměť sloužící pro uchování momentálně zpracovávaných dat a programů. Realizovaná většinou pomocí polovodičových součástek (RAM)

- registry,
- vyrovnávací paměti,
- hlavní paměť (operační).

Vnější (periferní, sekundární): paměť sloužící pro dlouhodobější uchování dat. Realizovaná většinou na principu magnetického (popř. optického) záznamu dat. Ve srovnání s operační pamětí bývá přístup k jejím datům výrazně pomalejší.

Archivní (záložní): pro velké objemy dat, pomalé (mg. pásky).

Rozdělení pamětí

Dělení pamětí z hlediska použité technologie:

- *bipolární* (ECL, TTL) – rychlejší ?
- *unipolární* (CMOS).

Dělení pamětí podle schopnosti uchovávat obsah při odpojení napájení (podle energetické závislosti):

- *volatilní* (nestálé) - ztratí obsah,
- *non-volatilní* (stálé) - drží obsah.

Charakteristické parametry

Vybavovací doba (access time) – udává rychlost s jakou paměť zapíše nebo vyhledá data (ns);

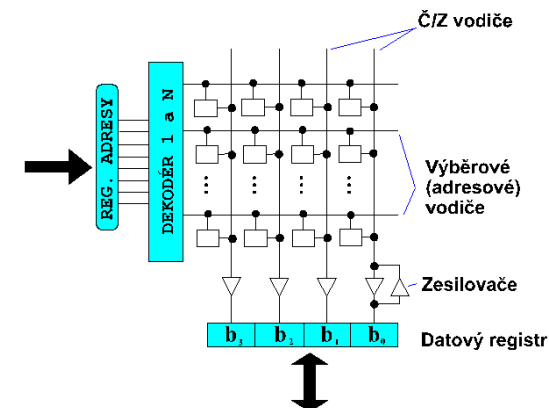
Doba cyklu (cycle time) – minimální časový interval, který musí uplynout mezi dvěma po sobě následujícími požadavky na čtení nebo zápis; je dána vybavovací dobou a časem ustálení přechodových dějů na sběrnicích (AS, DS); stále se zkracuje (ale pomaleji než se zkracují takty procesorů);

Kapacita paměti – jaké množství informace je možné v paměti uchovat (kB, MB, GB, 1kB = 1024 bitů) – vhodné udávat ve tvaru respektujícím organizaci paměti (součin počtu paměť. míst a délky paměťového místa, např. 4Mx4 Byty);

Přenosová rychlost – udává počet datových jednotek přenesených do nebo z paměti za sekundu.

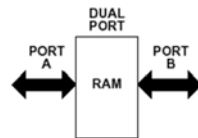
Struktura adresovatelné paměti

Paměťové buňky uspořádány do matice, napojeny na adresové a datové vodiče (sběrnice).



Víceportová paměť RAM

Víceportové (vícebranové) paměti mají obecně N adresových vstupů a N (nebo méně) datových vstupů či výstupů



Nejčastěji 2-portové:

- *simple (pseudo) dual-port* (jeden port pro čtení a druhý pro zápis do rozdílných buněk, často na stejné frekvenci),
- *true dual-port* (dva plnohodnotné porty, ze kterých lze libovolně provádět čtení či zápis při dvou různých frekvencích).

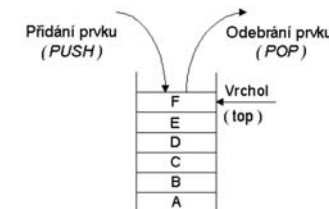
U víceportových pamětí je třeba zjistit chování při současném zápisu a čtení z téže adresy (buňky).

13

LIFO

Last In First Out – zásobníková paměť (stack)

- čte se poslední zapsaná položka – ovládání instrukcemi PUSH (zápis) a POP (čtení)
- pro manipulaci s uloženými daty se udržuje tzv. ukazatel zásobníku (vrchol zásobníku), který udává relativní adresu poslední přidané položky
- obsahem zásobníku mohou být jakékoliv datové struktury

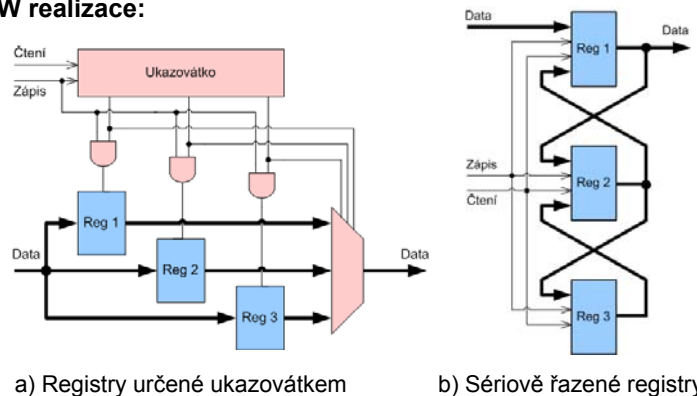


14

LIFO - realizace

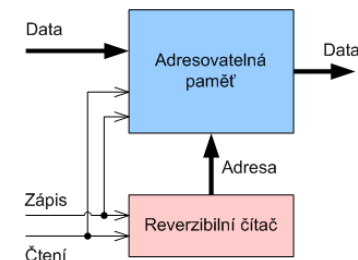
Realizace může být softwarová i hardwarová;

HW realizace:



15

LIFO – realizace (pokračování)



c) Adresovatelná paměť s reverzibilním čítačem

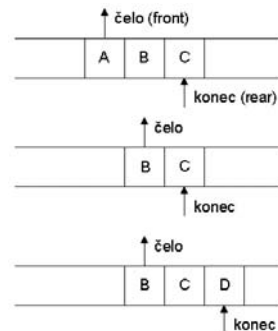
Nejčastější využití v procesoru – do zásobníku jsou ukládány návratové adresy a příznaky stavu procesoru při přerušeních a skocích do podprogramů.

16

FIFO

First In First Out – fronta (queue, pipe), která se rozšiřuje přidáním prvku na konec fronty a zmenšuje odebráním prvku z čela fronty;

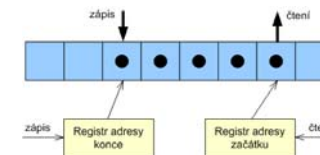
- registr s adresou začátku a konce,
- signály o stavu fronty (prázdná, plná),
- využití při vyrovnávání rychlostí.



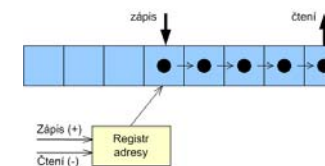
17

FIFO – principy funkce

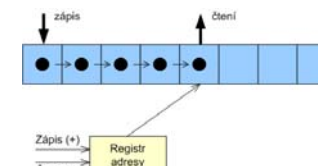
- bez přesouvání obsahu:



- s přesouváním obsahu:



a) přesouvání při čtení

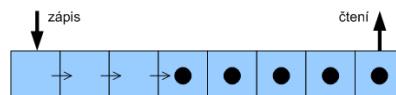


b) přesouvání při zápisu

18

FIFO – principy funkce

- c) s přesouváním obsahu při zápisu i při čtení (fronta s probubláváním – posouvá každou položku po zápisu asynchronně až do posledního volného místa) u každého paměťového místa musí existovat indikátor obsazenosti (klopný obvod).



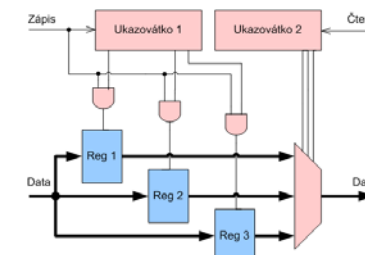
V principu je možné zápisy a čtení provádět jediným hodinovým signálem (Single Clock FIFO) nebo dvěma (Dual Clock).

19

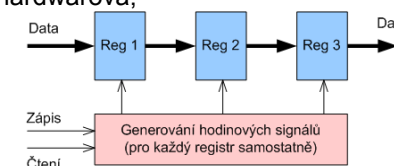
FIFO - realizace

Realizace může být softwarová i hardwarová;

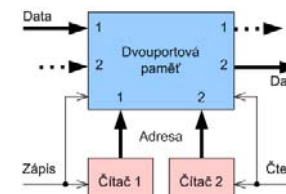
HW realizace:



a) Registry určené ukazovátkem



b) Sériově řazené registry



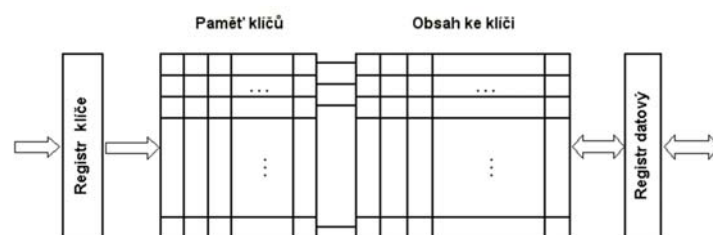
c) Dvouportová adresovatelná paměť

20

Asociativní paměť

CAM (Content Addressable Memory) – obsahem adresovatelná paměť

- tvořeny tabulkou, která obsahuje sloupec s klíči (tagy), podle kterých se v asociativní paměti vyhledává;
- v další tabulce jsou uchovávána data, popř. další informace nutné ke správné funkci (platnost dat aj.);
- použití ve vyrovnávacích a virtuálních pamětech.



21

Vyrovnávací paměť (cache, buffer)

Obečně rychlá polovodičová paměť umístěná mezi rychlým zařízením (procesorem) a zařízením pomalejším (např. operační paměť, HDD), která vyrovnává rozdíly v rychlosti

- zvyšuje výkon systému;
- obvykle se dělí na část pro data a pro instrukce;
- pohyb v paměti cache řídí *řadič cache* - umístěn přímo v CPU (L1, L2) nebo je součástí čipové sady (L2, L3);
- cache je založena na pamětech typu SRAM;
- Vyrovnávací paměť může být i softwarová (SMARTDRV, VCACHE), zvl. vyrovnávací paměť pro pomalé periferie (HDD, vypalovací mechaniky, síťové adaptéry) – lépe označovat jako buffer.

22

Cache v procesorových systémech

Nejčastěji rozdělujeme cache na:

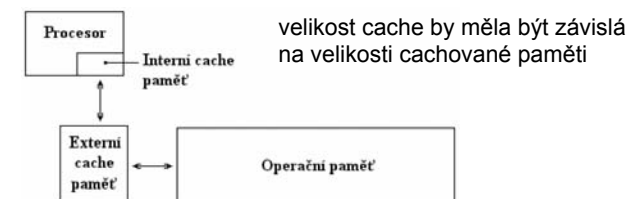
- *cache první úrovně L1* (first level, primární)
 - co nejbližší ALU (součástí CPU),
 - velmi rychlá,
 - relativně malá (velikost cca 32 kB – 128 kB);
- *cache druhé úrovně L2* (second level, sekundární)
 - pomalejší vůči L1,
 - velikost cca 256kB – 1MB,
 - u vícejádrových CPU každé jádro samostatná L2;
- *cache třetí úrovně L3*
 - u vícejádrových CPU sdílená všemi jádry,
 - velikost jednotky MB.

23

Cache v procesorových systémech

Rozdělení podle polohy vůči procesoru:

- *interní* – co nejbližší procesoru (L1, příp. L2);
- *externí* – většinou vně vlastního procesoru (L2 a vyšší), většinou vedle operační paměti na základní desce.



24

Rozdělení cache

Podle způsobu práce při zapisování dat:

- 1) *Write-Through* – (zápis skrz cache, přímý zápis) k zápisu do operační paměti dochází ihned po zápisu CPU do cache (o další osud dat se stará cache).
- 2) *Write-Back* – (opožděný zápis) data jsou zapisována do operační paměti až když je cache zaplněna – vyšší výkon (větší pravděpodobnost, že v cache jsou potřebná data).
- 3) *Pipelined Burst* – nejnovější, provede více operací zřetězeně (např. pokud čte data z nějaké adresy, přečte data i z následujících adres – což by pravděpodobně dělal za chvíli).

25

Implementace cache

Na principu asociativních pamětí; při přístupu do cache je nutné zadat adresu, z níž data požadujeme – tato adresa (buď celá nebo její část) je považována za tag

Druhy: přímo mapované
n-cestně asociativní
plně asociativní

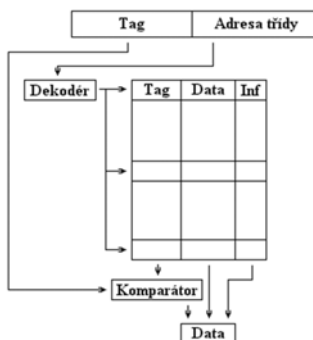
26

Přímo mapovaná (1cestná) cache

Zadaná adresa se rozdělí na 2 části: tag + adresa třídy;

Dekodér vybere podle adresy třídy jeden tag (řádek), jeho tag je následně porovnán s tagem adresy; není-li shoda, je z OP natažen celý řádek dat (více, např. 64 B dat).

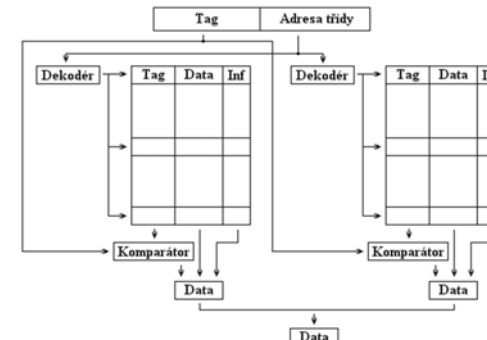
Jednoduché, ale málo efektivní.



27

n-cestně asociativní cache

Adresa třídy pomocí dekodérů vybere jeden řádek v každé tabulce, jehož tagy se porovnávají (nejpoužívanější);
- pro velká „n“ se blíží plně asociativní variantě



28

Plně asociativní cache

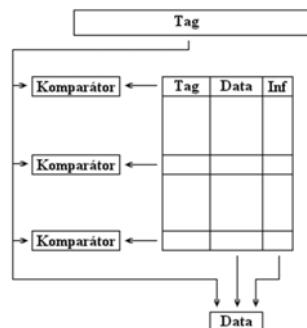
Celá adresa (ze které se budou číst data, resp. na kterou se budou zapisovat) je brána jako tag (porovnává se s tagy v cache – shoda znamená přítomnost dat v paměti).

Nevýhody:

- mnoho komparátorů
- nutná velká kapacita paměti pro tagy

Výhody:

- velká rychlost



29

Parametry cache

Parametry vypovídající o kvalitě cache

- pravděpodobnost úspěchu (*hit ratio*)
- pravděpodobnost neúspěchu (*miss rate*), neboli pravděpodobnost výpadku bloku (*fault*).

- lze definovat zvlášť pro čtení a zápis, pro data a instrukce.

Doba nalezení dat je v případě úspěchu přístupová doba cache, v případě neúspěchu se přičítá ztrátová doba (*miss penalty*), což je doba potřebná na přisunutí bloku, tj. doba potřebná na uvolnění místa + přístupová doba k OP + doba přesunu bloku.

Výpadky cache lze minimalizovat i softwarově (překladačem).

30

Velikost bloků cache

Vlastnosti větších bloků:

- větší blok obsahuje více slov v blízkosti požadovaného slova, více instrukcí jdoucích po sobě \Rightarrow způsobuje méně výpadků
- vzrůstá miss penalty (hlavně na přenos bloků)
- při velkých blocích a dané velikosti cache je v cache málo bloků \Rightarrow roste miss rate



31

Strategie řízení přesunu dat

Použití v cache a při přesunu stránek/segmentů virtuální paměti;

Dojde-li k zaplnění (a je třeba zavést další blok), který z bloků má paměť opustit:

- *LRU* (Least Recently Used) – nejdéle nepoužívaný blok; nejčastější algoritmus
- *NFU* (Not Frequently Used) – blok nejméně používaný
- *FIFO* (Aging - stárnutí) – nejstarší blok
- *RAND* (Random) – náhodně vybraný blok

Pro správu algoritmů jsou nutné další obvodové prvky, např. čítače pro udržování času použití.

32

Efektivita cache

AMAT (Average Memory Access Time) – průměrná doba přístupu do paměti (cache + OP):

$$AMAT = HT + MR \cdot MP$$

HT ... Hit Time – čas pro nalezení a získání položky v cache

MP ... Miss Penalty – čas získání dat z nižší úrovně paměť. hierarchie

MR ... Miss Rate – pravděpodobnost neúspěchu (výpadku bloku)

$$HT = (1 - MR) \cdot T_C = HR \cdot T_C \quad T_C \dots \text{vybavovací doba cache}$$

$$MR = 1 - HR$$

HR ... Hit Rate – poměrná úspěšnost nalezení dat v cache k celkovému počtu přístupů do paměti

Např. $T_C = 10 \text{ ns}$, $MP = 180 \text{ ns}$, $MR = 0,05 \Rightarrow AMAT = 18,5 \text{ ns}$
zlepšení vůči MP (úměrné vybavovací době OP) je cca 10x

33

Výkonnostní rovnice CPU s cache

$$T_{CPU}(prg) = (cyc_{CPU} + cyc_{MEM}) \cdot T_{clk} =$$

$$= IC \cdot (CPI + MAPI \cdot MR \cdot MP) \cdot T_{clk}$$

kde: $T_{CPU}(prg)$... doba provádění programu prg

cyc_{CPU} ... počet taktů, které procesor stráví výpočtem

cyc_{MEM} ... počet taktů, které zabere práce s pamětí

T_{clk} ... perioda hodin procesoru

IC ... počet instrukcí programu prg (Instruction Count)

CPI ... (průměrný) počet hod. taktů pro zpracování jedné instrukce (Cycles Per Instruction)

MAPI ... (průměrný) počet přístupů do paměti na jednu instrukci (Memory Access Per Instruction)

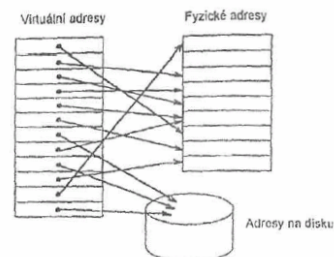
34

Virtuální paměť (adresace)

Zvláštní způsob správy operační paměti (OP), umožňuje oper. systému využívat vnitřní paměť, která je větší než skutečná fyzická velikost paměti. V danou chvíli nepotřebná paměť se odkládá na pevný disk. Virtuální adresní prostor je tedy realizován jak „rychlou“ OP, tak diskem.

Účelem je:

- vytvořit možnost efektivního sdílení paměti mnoha programy i více uživatelů;
- odstranění omezení fyzické velikosti paměti (OP je dražší než prostor na disku).



35

Virtuální paměť - charakteristika

- pracuje se s fyzickým a logickým adresním prostorem;
- o všechny adresy se stará správa virtuální paměti, která rozhoduje o tom, která část bude zavedena do OP a která bude odložena do odkládacího prostoru (swap);
- běžící procesy mohou využívat větší paměť;
- každý běžící proces má k dispozici vlastní paměťovou oblast (pouze on sám);
- paměť se z hlediska procesu jeví jako lineární (i když fyzicky je na různých místech paměti i odkládacího prostoru);
- může dojít ke ztrátě výpočetního výkonu (častou výměnou dat mezi pamětí a diskem).

36

Virtuální paměť - implementace

Existují 2 metody implementace:

- **stránkování** (paměť je rozdělena na relativně velké bloky stejné velikosti, např. 4 až 64 kB; horší využití místa);
- **segmentace** (paměť je rozdělena na bloky různé velikosti, větší volnost při programování, netřeba např. přesouvat celou stránku, pokud stačí pár bytů);
segment je skupina po sobě následujících paměť. míst, jejíž velikost lze měnit i během provádění programu a má jedinou počáteční adresu; obtížné umístění segmentů do OP – musí zůstat souvislý.

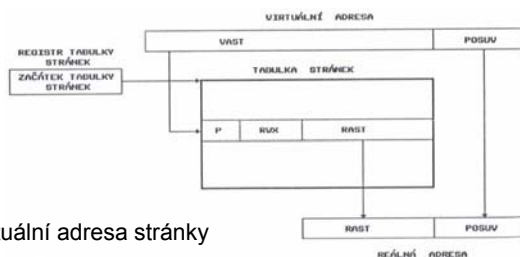
37

Překlad adres

- Překlad adres* se realizuje tabulkou (uloženou také paměti); aby tabulka nebyla příliš rozsáhlá, nepřekládá se celá virtuální adresa na fyzickou, ale jen část;
- při překladu se kontroluje i oprávnění přístupu (autorizace) vyjádřený znakem RWX (read/write/execute) – odvozuje se z kódu instrukce a z režimu práce (čtení, zápis, spuštění programu);
 - jestliže hledaná adresa stránky/segmentu není v OP, vyvolá se přerušení, uvolní se stránka/segment v OP a načte se z odkládacího prostoru požadovaná stránka/segment (mezitím může běžet jiná úloha).

38

Překlad adres stránkové paměti



VAST ... virtuální adresa stránky

RAST ... reálná adresa stránky

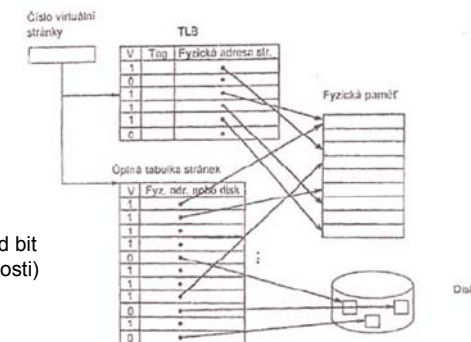
P ... příznak určující, zda je stránka v operační paměti

Překlad virtuálních adres u segmentů je analogický, pouze v tabulce segmentů je třeba mít informaci o jeho velikosti

39

Tabulka přeložených adres

TLB (Translation Look-Aside Buffer) – používá se pro zrychlení překladu virtuálních adres pomocí asociativní paměti (pokud se vstupní adresa nenalezne v TLB, hledá se v úplné tabulce stránek; v TLB jsou nejčastěji volané str.)



V ... valid bit
(bit platnosti)

40

ECC

(Error Correction Code, Error Checking and Correcting) - systém nadbytečné informace, která slouží k objevení a opravě vzniklých chyb – najde a opraví chybu v 1 bitu, detekuje chybu ve 2 bitech (kontrola v reálném čase), nutná podpora řadiče paměti (čipsetu); např. při 64 bitech dat je ECC 8 bitů.

Pokročilejší technologie:
multiple-bit error correction, Chipkill, memory scrubbing, Intel Single Device Data Correction (SDDC).

41

Počítačové paměti DRAM

SIMM (Single Inline Memory Module) 30 pinů, DRAM, 8-bitové, 60-70 ns



SIMM 72 pinů, FPM RAM nebo novější EDO RAM, 32-bitové



DIMM SDRAM 168 pinů, 66-166 MHz, 64-bitové



RIMM RDRAM 184 pinů, 16-bitové, 600-1333 MHz (jednokanálové)



RIMM RDRAM 232 pinů, 32-bitové, 800-1600 MHz (dvoukanálové)



42

Počítačové paměti DDRx DRAM

DIMM DDR SDRAM 184 pinů, 2,5 V, 266-433 MHz



DIMM DDR2 SDRAM 240 pinů, 1,8 V, 533-1066 MHz



DIMM DDR3 SDRAM 240 pinů, 1,5-1,9 V, 1066-1600 MHz



DIMM DDR4 SDRAM 284 pinů, 1,0-1,2 V, 2133-2800 MHz



43

Počítačové paměti DDRx DRAM

DDR (Double Data Rate) – synchronní dynamické paměti, u kterých dochází k přenosu dat na náběžnou i sestupnou hranu hodin.

Při zápisu/čtení dat se nejprve zadá adresa řádku (RAS - Row Access Strobe) a pak sloupce (CAS - Column Access Strobe).

Časování paměti: tCL-tRCD-tRP-tRAS, např. CL9-10-9-27

tCL (CAS Latency) – zpoždění výstupu dat po adresaci sloupce;
tRCD (RAS to CAS Delay) – zpoždění mezi výběrem řádku a adresací sloupce;

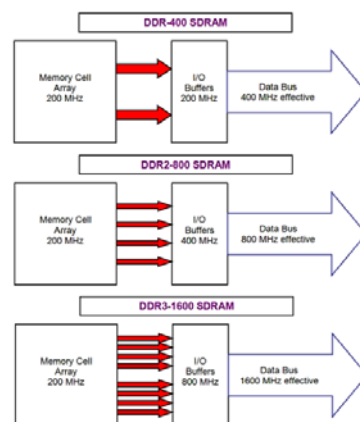
tRP (RAS Precharge) – zpoždění po výběru paměti do adresace řádku;

tRAS (Row Active Time) – doba, jak dlouho musí zůstat RAS aktivní;

Časy se udávají v počtu hodinových taktů => čím nižší, tím lepší.

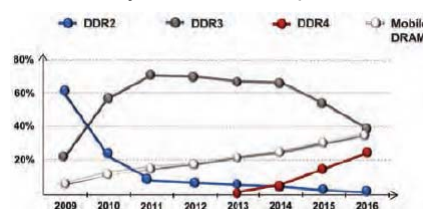
44

Paměti DDR DRAM



Rychlost samotných paměťových čipů se v podstatě nemění (DDR-DDR3), rychlejší jsou I/O buffery.

Trendy na trhu DDR pamětí

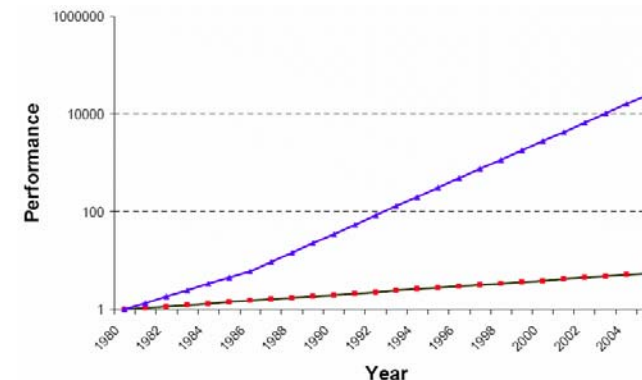


45

Vývoj výkonnosti paměti

mikroprocesory – zvyšování výkonnosti 60%/rok (2x/1,5 roku)

paměti – zvyšování výkonnosti 9%/rok (2x/10 let)



46

SSD disky

Solid State Drive

Disk na principu převážně non-volatilních NAND flash pamětí (výjimečně SRAM či DRAM => RAM-drive),
SLC (Single Level Cell) – v jedné buňce 2 úrovně (1 bit),
MLC (Multi Level Cell) – v jedné buňce 4 úrovně (2 bity)
=> větší kapacita, nižší rychlost, menší počet přepisů;
nejčastěji rozhraní SATA III nebo PCI Express.

Výhody: velká přenosová rychlost (150–1800 MB/s),
nízká spotřeba energie (malý ztrátový výkon),
minimální přístupová doba (0,1 ms),
mechanicky odolné, tichý provoz,
nízká hmotnost i rozměry,
není nutné defragmentovat.



47

SSD disky (pokračování)

Nevýhody: omezený počet přepisů flash pamětí (řádově 10^5 až 10^6 cyklů) – nutné rovnoměrné rozložení zápisů (stárnutí) - v důsledku opotřebení se degraduje výkon;
zápis nových dat je možné pouze do „vyčištěných buněk“
=> nutný cyklus čtení-smazání-upravení-zápis
=> pro udržení výkonu probíhá čištění uvolněných buněk na pozadí;
nižší kapacity disků (řádově 10^1 – 10^3 GB),
vyšší cena na jednotku kapacity (cca 10 Kč/GB).

SSHD (Solid State hybrid drive)
hybridní disk – běžný pevný disk
(500-1000 GB) doplněný SSD
malým diskem (cca 8 GB)



48

RAID pole

Redundant Array of Independent Disks

více pevných disků propojeno (převážně prostřednictvím HW řadiče)
za účelem zvýšení rychlosti a spolehlivosti:

RAID 0 (striping) – data se ukládají prokládaně po blocích (sudé a liché bloky) na různé disky, žádný redundantní disk, zvýší se rychlost (ne spolehlivost), min. 2 disky.

RAID 1 (mirroring) – data se ukládají duplicitně na dva disky, je třeba dvojnásobný počet disků, zvýší se spolehlivost (rychlost se může i snížit).

RAID 2 (error detection and correction code) – obdoba detekce chyb u pamětí, nutná velká redundance.

RAID 3 (Bit-interleaved parity) – 1 disk paritní, je silně vytížen, data se dají z parity zrekonstruovat.

RAID pole (pokračování)

RAID 4 (Block-interleaving parity) – obdoba RAID3, jiný princip výpočtu parity, při zápis po blocích, paritní disk silně vytížen.

RAID 5 (Distributed block-interleaved parity) – obdoba RAID 4, paritní bloky jsou na všech discích, 1 disk redundantní, potřeba min. 3 disky.

RAID 6 (P+Q redundancy) – obdoba RAID 5, dva paritní disky (pomalejší zápis), dovoluje obnovu po 2 chybách.

RAID 0+1 – dvě pole RAID 0 se spojí do RAID 1 (min. 4 disky), vyšší propustnost i spolehlivost.

RAID 10 – pole RAID 1 se spojí do RAID 0 (min. 4 disky).

Obdobně **RAID 50** (min. 6 disků) nebo **RAID 60** (8 disků).

Komunikace procesoru se vstupně-výstupními zařízeními

Milan Kolář

Ústav mechatroniky a technické informatiky

Komunikace CPU s I/O zařízeními

Vstupně výstupní brána (I/O port, V/V řadič) – zprostředkovává předávání dat mezi sběrnici počítače a periferním zařízením.

V/V periferie jsou obecně pomalejší oproti CPU, součinnost se realizuje těmito základními principy:

- 1) *přímou programovou obsluhou*
- 2) *obsluhou s přerušením* (interrupt, trap)
- 3) *komunikací prostřednictvím DMA* (kanály)
- 4) *specializovanými V/V procesory* (kanály)

Kanály - schopné realizovat velké množství V/V operací (odlehčení procesoru).

2

Rozdělení V/V bran

Z hlediska programátora:

- *jen pro čtení*
- *jen pro zápis*
- *pro čtení i zápis* (obousměrné)

Z hlediska architektury:

- *paměťově mapované* (adresovány jako paměť, přístupné pomocí běžných operací čtení a zápisu do paměti)
- *izolované (porty)* – registry jsou přístupné pomocí speciálních instrukcí – IN, OUT)
adresní prostory paměti a V/V bran jsou oddělené

3

1) Přímá programová obsluha

Styk CPU s periferiemi řízen programovými prostředky. Program prostřednictvím stavové vstupní brány postupně testuje, zda a která V/V zařízení jsou připravena vyslat, příp. přijmout data.

Při zjištění připravenosti procesor vyvolá V/V podprogram (*ovladač – driver*), který zajistí vstup nebo výstup dat; ovladač zajišťuje řídicí signály pro danou periferii.

Ve většině případů však používání přímé programové obsluhy počítač zdržuje, neboť je zatěžován periodickým testováním stavů (i když nedochází k přenosům dat).

Vhodná pro relativně rychlá zařízení.

4

Přímá programová obsluha

V programu jsou instrukce vstupu nebo výstupu a komunikace s periferním zařízením synchronizovány:

- vkládáním *čekacích taktů*
- pomocí tzv. *handshake* (před instrukcí V/V je vložena smyčka zjišťující, zda je zařízení schopno přijmout, resp. poskytnout informaci).

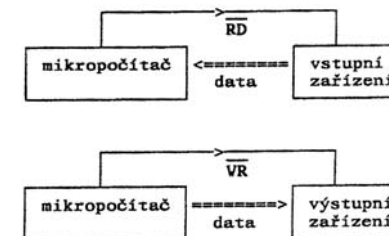
Programové řízení je výhodné z hlediska obvodového řešení. Pro zmenšení zátěže CPU se část stykové úlohy přenáší z programu na obvody, příp. komunikační procesor.

5

Technika nepodmíněného V/V

Jednoduché, ale předpokládá, že periferní zařízení je stále připraveno komunikovat (pro rychlé periferie).

Při vstupu vyšle procesor bitový signál RD (Read), čímž vstupní zařízení předá data, při výstupu vyšle procesor signál WR (Write) a výstupní zařízení převezme data.

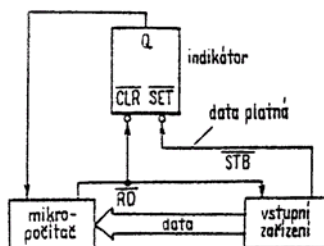


6

Technika podmíněného vstupu

Jednosměrný korespondenční (neúplný) režim – o zahájení nebo ukončení přenosu je informován pouze μP a vysílač dat je povinen data udržovat.

Má-li vstupní zařízení platná data, nastaví pomocí signálu STB (Strobe) výstup klopného obvodu ($Q = 1$); μP následně impulsem RD převezme data a nuluje indikátor Q.

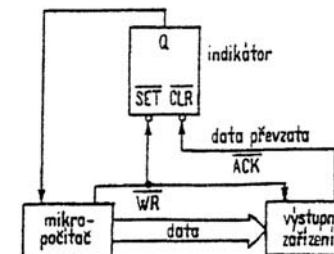


7

Technika podmíněného výstupu

Opět *jednosměrný korespondenční (neúplný) režim*

Má-li μP platná výstupní data, vyšle impuls WR, který současně nastaví indikátor Q. Výstupní zařízení po převzetí dat impulsem ACK (Acknowledge) nuluje indikátor Q. Tím může μP vyslat další data.

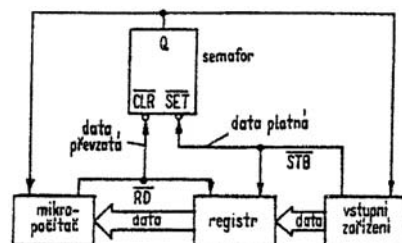


8

Úplný podmíněný vstup

Obousměrný korespondenční (úplný) režim – využívá registr (vyrovnávací paměť) a klopný obvod pracující jako semafor, který je testován vysílačem i přijímačem dat.

Pokud je registr plný, semafor je ve stavu 1 a vstupní zařízení další data nevyšle. Je-li prázdný ($Q = 0$), lze signálem STB do registru vyslat data, čímž se semafor nastaví do 1. Při $Q = 1$ může μP impulsem RD data převzít a vynulovat Q

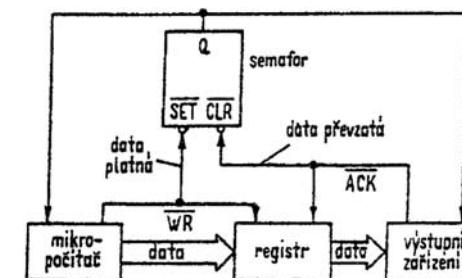


9

Úplný podmíněný výstup

Opět obousměrný korespondenční (úplný) režim

Při výstupu μP testuje semafor. Je-li nulový, impulsem WR vyšle data do registru a nastaví semafor do stavu 1. Výstupní zařízení tak zjistí, že v registru jsou data a může je impulsem ACK převzít a semafor vynulovat.



10

2) Přerušení

„nestandardní“ událost v systému (vně i uvnitř procesoru), která způsobí přerušení právě probíhajícího programu (bez ohledu na právě prováděné místo v programu se dokončí pouze právě prováděná instrukce); procesor začne zpracovávat program obsluhy přerušení a po jeho dokončení se vrátí na přerušené místo.

Vhodné pro zařízení (resp. události), jejichž rychlost (resp. četnost) je výrazně nižší než rychlost procesoru (počítač nemusí jednotlivá zařízení, resp. události, periodicky testovat).

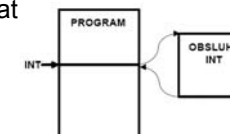
(IRQ – Interrupt Request – žádost o přerušení)

11

Činnost při přerušení

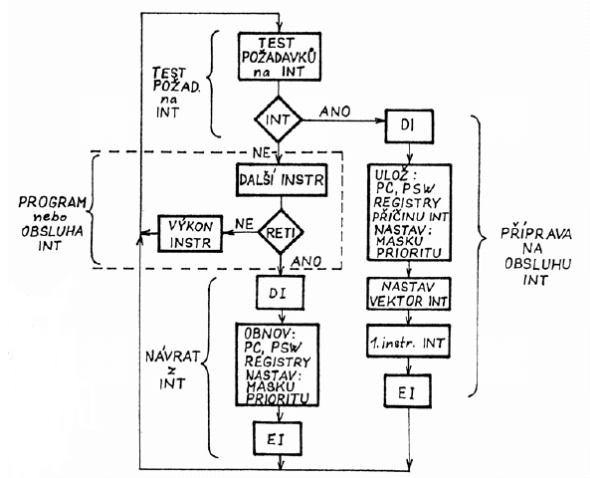
Objevili-li se žádost o přerušení, tak procesor:

- dokončí rozpracovanou instrukci
- znemožní přijetí dalších žádostí o přerušení (např. nulováním interního indikátoru, DI – Disable Interrupt)
- vyšle signál o akceptování požadavků (INTA)
- určí, který podprogram se bude vykonávat
 - a) vnucení adresy
 - b) vnucení instrukce
 - c) vnucení přerušovacího vektoru
- zpracuje se podprogram přerušení
- v podprogramu přerušení se opět povolí přerušení (instrukce EI – Enable Interrupt)
- nakonec se vrátíme na přerušené místo hlavního programu.



12

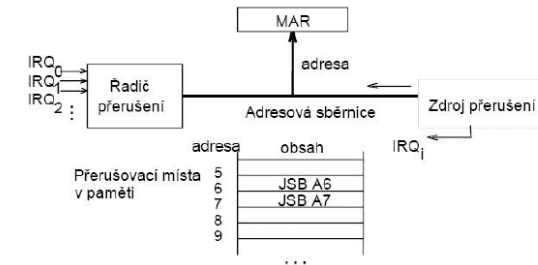
Obsluha přerušení



13

a) Vnucení adresy

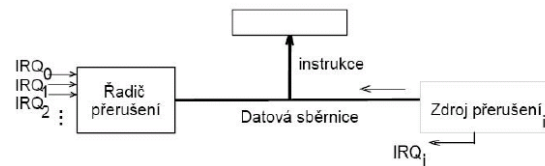
Každý zdroj je identifikován adresou, kterou je schopen při přerušení zaslat procesoru – do registru MAR (Memory Address Register) - následující instrukce pak bude načtena z této adresy



14

b) Vnucení instrukce

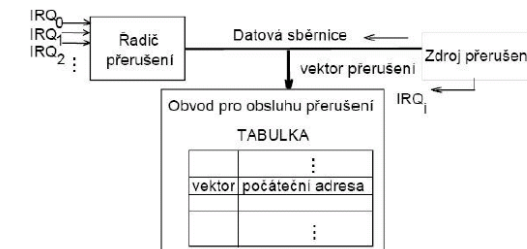
Jednotka vyžadující přerušení (příp. řadič přerušení) zašle procesoru po datové sběrnici instrukci, která se má provést (např. RST n, CALL n)



15

c) Vnucení přerušovacího vektoru

Každý zdroj přerušení generuje přerušovací číslo, zvané *vektor přerušení*, které je ukazatelem do tabulky počátečních adres přerušovacích podprogramů (*tabulka vektorů přerušení*); nejčastější (většinou od adresy 0 v OP, např. 256 x 4B = 1kB).



16

Typy přerušení

a) Hardwarové

- *vnitřní* (od procesoru) – je vyvoláno chybou při provádění strojové instrukce (dělení nulou, přetečení, výpadek stránky cache, neplatná instrukce, apod.)
- *vnější* (od periférií vně procesoru) – nejčastěji od řadiče přerušení, V/V zařízení (IRQ n se označují linky od periférií k procesoru), od časovače aj. Procesor tyto vstupy neustále testuje (při každé instrukci).

Podle reakce na vnější přerušení:

- *maskovatelné* (INTR) – je možné dočasně zakázat (např. nastavením bitu IF – Interrupt Flag);
- *nemaskovatelné* (NMI – Non-Maskable Interrupt) zejména pro havarijní situace (výpadek napájení).

17

Typy přerušení (pokračování)

b) Softwarové (programové)

- přerušení je vyvoláno instrukcí volání přerušení umístěnou přímo v programu (volání služeb OS), (např. INT n – volá přerušovací podprogram n).
Vesměs jde o nepřímé volání podprogramů; má vysokou (nejvyšší) prioritu.

Z hlediska časování:

- *asynchronní* přerušení (nejčastěji hardwarová vnější);
- *synchronní* přerušení (s během programu) – nejčastěji softwarová a hardwarová vnitřní) např. INT, TRAP.

18

Řešení priorit

Při větším počtu zdrojů žádostí o přerušení vzniká problém:

- počtu přerušovacích vstupů (zdrojů),
- priorit jejich obsluhy.

Více zdrojů přerušení lze řešit třemi způsoby:

- programovou identifikací,
- sériovou obvodovou identifikací,
- řadičem přerušení (interrupt controller).

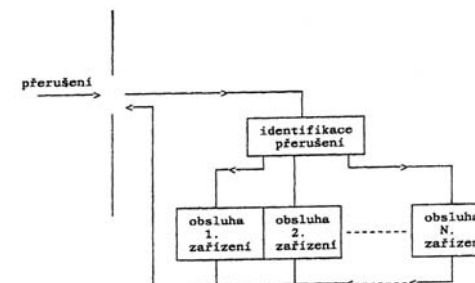
Každé přerušení má stanovenou prioritu - pro případ současné žádosti více komponent;

- standardní přerušení (řeší se po dokončení instrukce, během níž vznikl požadavek);
- výjimka vysoké úrovně (řeší se během provádění instrukce – nelze ji dokončit).

19

a) Programová identifikace

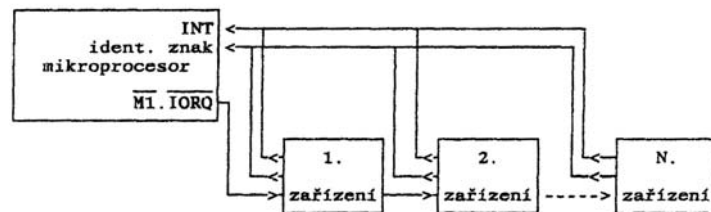
K přerušovacímu vstupu procesoru je připojen signál logického součtu externích signálů přerušení; po akceptování žádosti o přerušení se vyvolá přerušovací podprogram, který programově identifikuje přerušující zařízení (*cyklické výzvy – pooling*); podle pořadí lze určit i prioritu.



20

b) Sériová obvodová identifikace

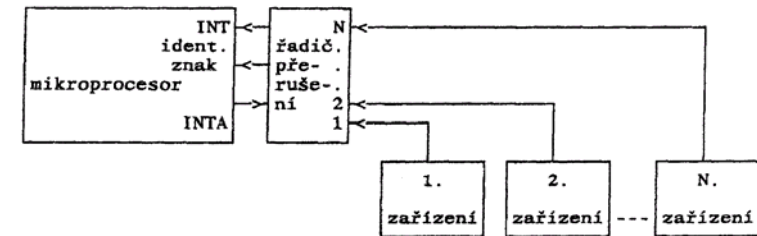
Procesor po příchodu přerušovacího signálu očekává identifikační znak generovaný stykovými obvody, který jej nasměruje do žádoucího podprogramu. Stykové obvody jsou spojeny sériově v pořadí priority (neuplatní se přerušení nižší priority) – *zřetězení* (Daisy Chain).



21

c) Řadič přerušení

Specializovaný stykový obvod, který soustřeďuje všechny žádosti ze všech přerušovacích zdrojů; periferní zařízení vysílají pouze požadavky na obsluhu do řadiče přerušování. Řadič požádá o přerušení i v případě, že nový požadavek má vyšší úroveň než právě probíhající.



22

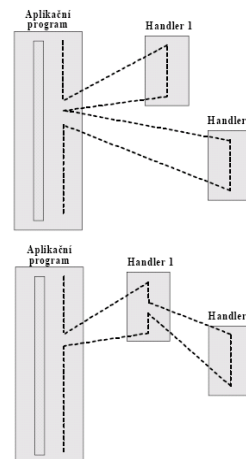
Vícenásobná přerušování

Sekvenční zpracování

- během obsluhy jednoho přerušování se další požadavky nepřijímají (pozdržují se);
- jednoduché, ale ne vždy vhodné.

Vnořené zpracování

- přijímají se přerušování s prioritou vyšší než je obsluhovaná priorita



23

3) DMA

Direct Memory Access – *přímý přístup do paměti*

Metoda kopírování bloků dat mezi pamětí a portem (HDD, pamětí, ...) bez průchodu skrz procesor, bez dočasněho ukládání těchto dat v pomocných registrech

- rychlejší,
- méně náročné na výkon.

Teoreticky přenos dat např. ze V/V portu do paměti nelze provést přímo (kvůli jedné adresové sběrnici) - je třeba nejprve data načíst do pomocného registru CPU a odtud (třeba ještě v rámci téže instrukce) do paměti (třeba min. dva cykly sběrnice)
⇒ řešíme využitím DMA řadičů.

24

Řadič DMA

Musí umět:

- generovat adresy pro paměť,
- generovat příslušné řídicí signály,
- žádat CPU o uvolnění paměti (zabránění kolize v paměti),
- po ukončení činnosti vrací řízení sběrnic procesoru.

V praxi většinou DMA řadiče tvořeny několika samostatnými kanály vyhrazenými určitému zařízení, které umí adresovat.

Řadič potřebuje ke své činnosti 3 údaje:

- kolik dat má přenést,
- kterým směrem,
- ze/do kterého místa paměti (tj. od jaké počáteční adresy).

25

Činnost řadiče DMA

- Postup: 1) zařízení požádá o přímý přenos dat svůj řadič (při výstupu dat požádá DMA řadič sám procesor);
2) DMA řadič požádá procesor o přidělení sběrnic;
3) po akceptování CPU vyšle DMA řadič adresu paměti na adresovou sběrnici a vygeneruje příslušné řídicí signály - pro paměť (R/W) i zařízení;
5) zařízení vyšle data na datovou sběrnici, resp. začne číst data z této sběrnice;
6) po ukončení přenosu to zařízení oznámí řadiči;
7) DMA řadič vrátí sběrnice procesoru.

DMA řadič není řízen instrukcemi – je to jednoúčelové HW zařízení.

26

Způsoby řízení přenosu DMA

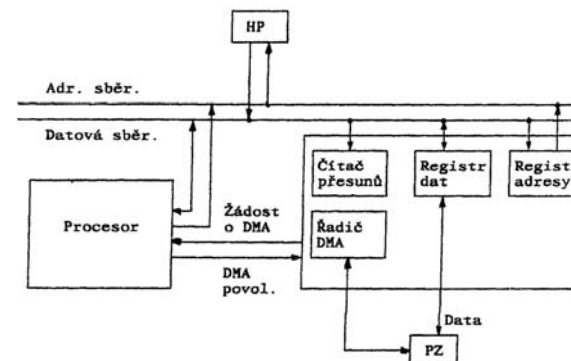
- 1) CPU převede své sběrnice do neutrálního stavu, jejich řízení převezme řadič DMA, procesor nepokračuje v běžné činnosti.
- 2) Odpojí se generování hodinových signálů pro CPU, odpojí se jeho budiče sběrnic (nejčastěji na jediný dílčí přenos).
- 3) Dílčí přenosy se uskutečňují v časových intervalech, v nichž procesor pracuje, ale nekomunikuje po sběrnicích (neovlivňuje rychlost CPU); je třeba přísně synchronní činnost CPU a řadiče DMA.

27

Spolupráce DMA - CPU

HP – hlavní paměť

PZ – periferní zařízení



28

4) Specializované V/V procesory

Mezi procesor a řadič periferie se vkládá další jednotka (kanál), což je specializovaný V/V procesor schopný samostatně řídit V/V operace.

Používá se u rozsáhlejších počítačových systémů, které mohou mít i více kanálů (i více CPU).

Kanál je obdobou DMA, ale je více samostatný;

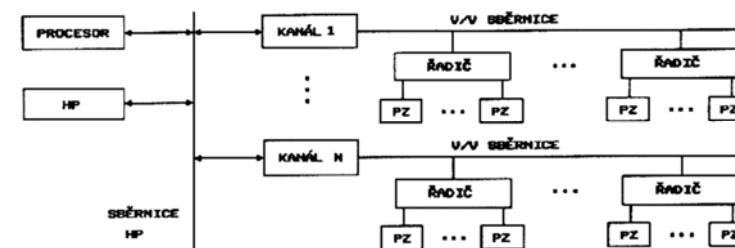
je řízen posloupností vlastních instrukcí (tzv. kanálovým programem)

- *selektorový kanál* (v jednom okamžiku může obsluhovat jen jedno zařízení, vhodné pro rychlé PZ);
- *multiplexní kanál* (může obsluhovat několik PZ současně; rozdělen na podkanály).

Samostatné kanály

HP – hlavní paměť

PZ – periferní zařízení



Sběrnice

Milan Kolář
Ústav mechatroniky a technické informatiky

Co je sběrnice

Informační cesta umožňující přenos informace mezi jednotlivými funkčními bloky systému nebo mezi systémem a okolím.

Soustava vodičů, které přenáší data nebo signál stejného charakteru.

Sběrnice má obvykle řadič (master), kterému jsou ostatní části podřízené (slave).

Sběrnici lze definovat z hlediska:

- mechanického (tvar a rozměry mechanických dílů, konektorů)
- elektrického (úroveň U a I, přiřazení logických stavů)
- funkčního (význam jednotlivých signálů a jejich časové průběhy)
- operačního (informační kódy a formáty přenášených zpráv)

Dělení sběrnic

Podle topologie:

- dvoubodovým spojem
- vícebodovým spojem

Podle výstupního obvodu:

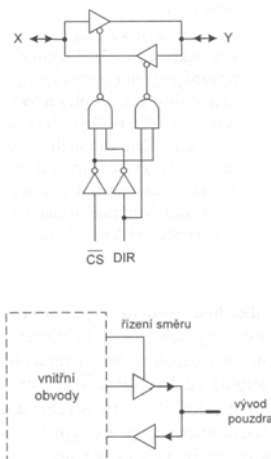
- třístavové
- s otevřeným kolektorem

Podle umístění:

- vnitřní (krátké, velký počet vodičů)
- vnější (pro malé i velké vzdálenosti)

Podle směru přenosu:

- jednosměrné
- obousměrné



Dělení sběrnic

Podle úrovně signálu:

- asymetrické (SE – Single Ended)
- symetrické (LVD – Low Voltage Differential)

Podle způsobu přenosu:

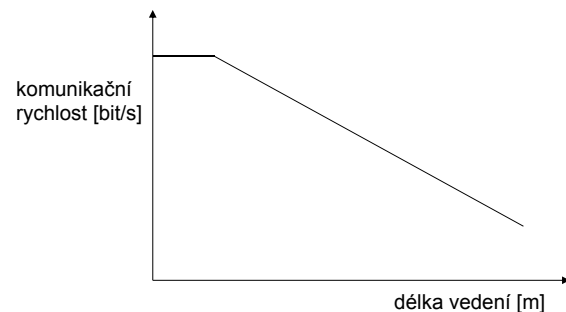
- sériové
- paralelní
- kombinované (sério-paralelní)

Podle funkce:

- datové
- adresové
- řídicí

Technické parametry sběrnic

- maximální délka sběrnice,
- počet přípojných míst (zatížení),
- přenosová rychlost (kapacita),
- impedance sběrnice (odpor),
- napěťové úrovně,
- konektory ...



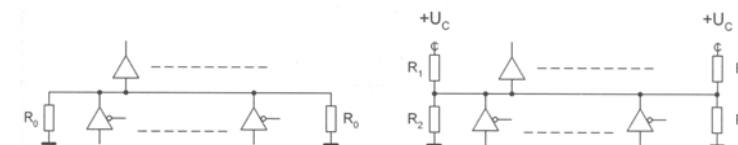
5

Impedanční zakončení sběrnice

Na sběrnici je třeba pohlížet jako na **dlouhé vedení**, které je třeba impedance zakončit **terminátory** – „právě jen“ na obou koncích (ideálně $R_0 = Z_0 = R_1 || R_2$).

U sběrnic s otevřeným kolektorem je třeba mít pull-up rezistory rovněž na obou koncích vedení (jako terminátory).

Terminátory zatěžují napájecí zdroj (nevhodné zejména u vícevodičových sběrnic nebo u bateriového napájení).

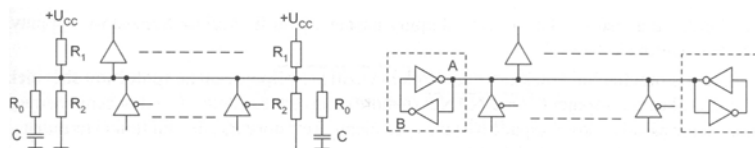


6

Terminátory s malým příkonem

Pro snížení příkonu se používají sériové rezistory nebo doplňkové kondenzátory (pro střídavé signály zkrat) – časová konstanta R_0C se volí min. 4x větší než zpoždění signálu na vedení (R_1 a R_2 mají velkou hodnotu a zajišťují pouze definovanou úroveň při odpojení všech budičů).

Lze použít i **aktivní terminátory** („bus hold“) – princip bistabilního klopného obvodu (odčerpává energii přicházející vlně a již nedojde k odrazu).



7

Synchronní a asynchronní přenos

Synchronní přenos – společně s daty se vysílají synchronizační impulsy, příp. speciální synchronizační vodič

- signál obsahuje jen celistvé násobky nejkratších charakteristických intervalů,
- výhodné pro velké objemy dat,
- v každém okamžiku konstantní přenosová rychlost.

Asynchronní přenos - dávkový přenos dat

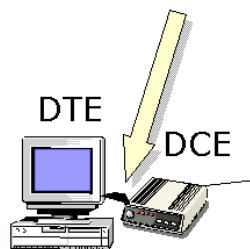
- přenosový rámec obsahuje synchronizační informace,
- vhodné při nepravidelných požadavcích na přenos,
- odolnější proti rušení.

Arytmický přenos – kombinace asynchronního a synchronního přenosu (v rámci jedné značky se pracuje synchronně, značky se přenáší asynchronně).

8

RS-232

- sériová komunikace arytickým přenosem;
- úrovně datových signálů v negativní logice;
- předpokládá dva typy zařízení:
 - DTE (Data Terminal Equipment) – koncové zařízení (např. počítač);
 - DCE (Data Circuit Terminating Equipment) – zařízení ukončující okruh (např. modem).



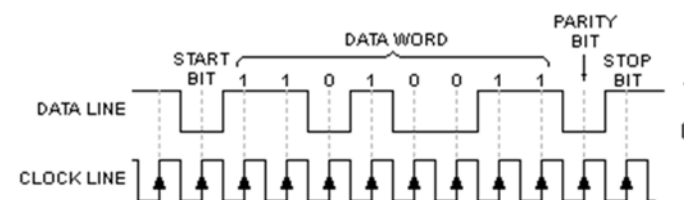
Datové signály		
Úroveň	Vysílač	Příjímáč
Log. L	+5 V to +15 V	+3 V to +25 V
Log. H	-5 V to -15 V	-3 V to -25 V
Nedefinovaný	-3 V to +3 V	

Řídící signály		
Signál	Driver	Terminátor
"Off"	-5 V to -15 V	-3 V to -25 V
"On"	5 V to 15 V	3 V to 25 V

9

RS-232

- maximální rychlosti do 19200 bit/s (115,2 kbit/s);
- maximální délka sběrnice normována na 15 m nebo kapacitu 2,5 nF (možno dosáhnout až 50 m);
- přenos po rámcích (start bit, 5-8 datových bitů, paritní bit, 1-2 stop bity).



10

RS-232

Významné signály rozhraní:

- RxD (Read Data) – vstupní data
- TxD (Transmit Data) – výstupní data
- DTR (Data Terminal Ready) – počítač je připraven
- DSR (Data Set Ready) – modem je připraven
- RTS (Request To Send) – žádost o možnost vyslat data
- CTS (Clear To Send) – povolení možnosti vyslat data



11

I²C (Inter Integrated Circuit)

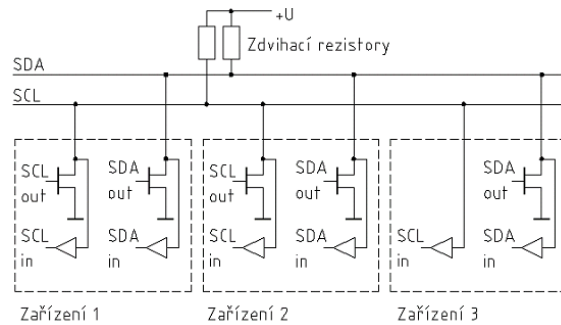
- navržen firmou Philips pro komunikaci obvodů (paměti, porty, A/D a D/A převodníky, snímače, budiče displeje, ...);
- dva typy zařízení – master a slave;
- sběrnice typu multi-master s detekcí kolizí (v každém okamžiku jen jediný master);
- každé zařízení je sw adresovatelné (má jedinečnou adresu);
- přenos je sériový, osmibitový, synchronní:
 - Standard mode (100 kbit/s),
 - Fast mode (400 kbit/s),
 - High-speed mode (3,4 Mbit/s);
- maximální počet zařízení limitováno celkovou kapacitou sběrnice (nesmí překročit 400 pF).



12

I²C

- používá dva obousměrné vodiče – sériovou datovou linku SDA (Serial Data Line) a linku hodinového signálu SCL (Serial Clock Line);
- na sběrnici jsou budiče s otevřeným kolektorem + pull-up odpory.

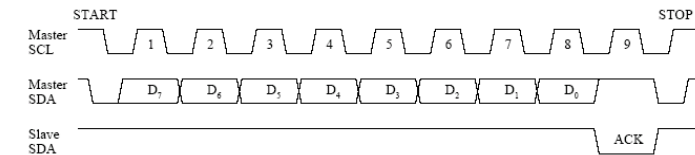


13

I²C

Definované situace:

- *klidový stav* – SDA i SCL jsou v log.1 (neaktivní)
- *podmínka startu* – SDA je masterem stažena na log.0
- *podmínka stop* – SDA přejde z log.0 na 1, SCL v log.1
- *přenos dat* – daný vysílač přivede na SDA 8 datových bitů (posun v rytmu SCL od mastera)
- *potvrzení (acknowledge)* – přijímač potvrdí příjem log.0 na SDA; může se přijímat další byte



14

I²C

- neaktivní účastníci sběrnice jsou v log.1 a neustále vyhodnocují signály na sběrnici;
- je-li použit jen jeden master, vysílá SCL jen on;
- data může vysílat jak master, tak slave;
- přenos a potvrzování adres je stejné jako u dat;
- postup: master vyrobí podmínku startu a pak v bitech 7 až 1 přeneše adresu součástky, v bitu 0 (R/W) je požadovaný směr přenosu; slave potvrdí adresu, pak se posílají data (každých 8 bitů je potvrzeno); přenos ukončí master podmínkou stop.

15

SPI (Serial Peripheral Interface)

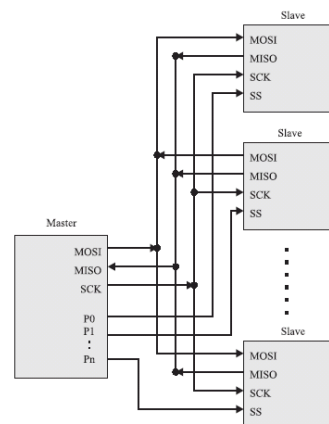
- navrženo firmou Motorola;
- dva typy zařízení – master a slave;
- synchronní sériové rozhraní se čtyřmi druhy vodičů:
 - datový výstup MOSI zařízení master (Master Out Slave In)
 - datový vstup MISO zařízení master (Master In Slave Out)
 - výstup hodinového signálu SCK z masteru
 - výběrový vodič SS (Slave Select) – aktivní v log.0 (z master do každého zařízení slave);
- rychlost SCK je standardně 2 MHz, u variant High-speed SPI (HSSPI) až 20 MHz;

16

SPI – připojení zařízení

Zařízení lze k SPI připojovat:

- *paralelně* (signály MISO musí být s třístavovými budiči, SS je rozveden individuálně)
- *sériově* do řetězce (k-násobné prodloužení datového rámce, aktivní všechna zařízení).

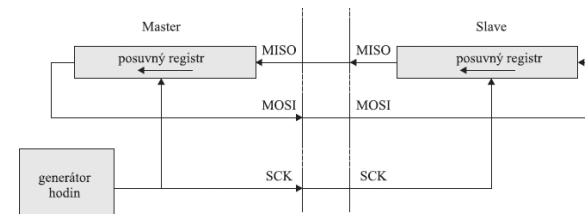


17

SPI

Komunikace na sběrnici mezi zařízeními Master a Slave:

- obě zařízení obsahují 8-bitový posuvný registr, který postupně po 8 taktech čtou nebo do něj zapisují,
- master generuje hodinový signál SCK,
- přenos probíhá od MSB k LSB.



Princip komunikace po SPI

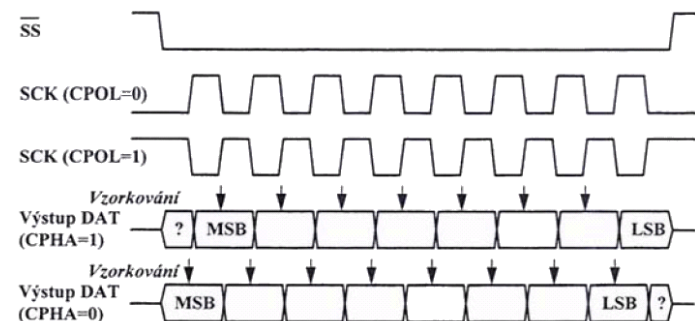
18

SPI - módy

Konfigurační módy:

CPOL – určuje polaritu SCK

CPHA – posun datového signálu vůči hodinám



19

USB (Universal Serial Bus)

Standard pro sériový přenos dat (4, resp. 8 vodičů – USB 3).

Charakteristické parametry:

- komunikační rychlost
 - Super Speed - 4.8 Gbits/s (v.3.0),
 - High Speed - 480 Mbits/s (v.2.0),
 - Full Speed - 12 Mbits/s (v.1.1),
 - Low Speed - 1.5 Mbits/s (v.1.1),
- komunikační vzdálenost do 5 m (se zesílením až 30 m),
- lze připojit až 127 zařízení, zpětná kompatibilita,
- log.0 ... 0-0.3 V, log.1 ... 2.8-3.6 V (diferenciální),
- zajišťuje správné přidělení prostředků (IRQ, DMA, ...).



Pin	Jméno	Barva	Popis
1	VBus	Red	+5 VDC
2	D-	White	Data -
3	D+	Green	Data +
4	GND	Black	Ground

20

USB

- podpora ovladačů ve stávajících operačních systémech;
- autoidentifikace periférií a překonfigurovatelné periferie;
- HotSwap – připojení/odpojení za chodu počítače;
- možnost využití celé šířky pásma jedním zařízením;
- rozhraní obsahuje 5V napájení
 - zařízení mohou být napájena přímo ze sběrnice (do 100 mA, příp. 500 mA pro jedno zařízení, 900 mA pro v.3.0)
- pro vyšší rychlosti třeba stíněný kabel;
- každé USB zařízení má svoji adresu;
- dva typy konektorů – typ A (USB Host) a B (USB Device).

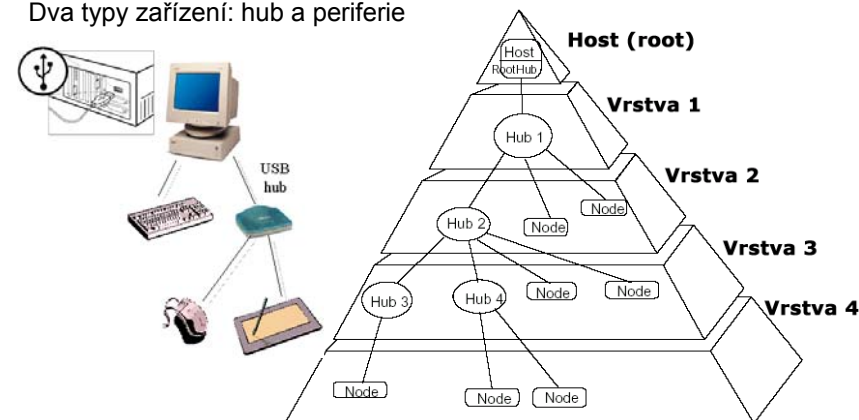


21

Architektura USB

Víceúrovňová hvězdicová struktura, pouze jeden hostitel

Dva typy zařízení: hub a periferie



22

Režimy komunikace na USB

- *řídící (control mode)* – používá hlavně PC k inicializaci;
- *režim přerušení (interrupt mode)* – používají zařízení, která mají k přenosu malé objemy dat (např. myši, klávesnice);
- *dávkový režim (bulk mode)* – pro zařízení, která potřebují přenášet velké objemy dat se 100% spolehlivostí (disky);
- *izochronní režim* – pro zařízení, která potřebují komunikovat v reálném čase, ale dokáží tolerovat určité ztráty dat (např. audio zařízení);

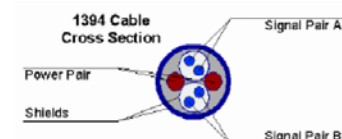
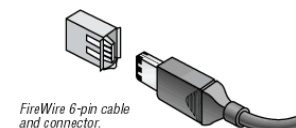
Při startu systému nebo při zapojení zařízení host:

- přiřadí každému zařízení identifikátor (bus ID);
- zjistí od každého zařízení, v jakém režimu chtějí komunikovat.

23

IEEE 1394 (FireWire, i-Link)

- sériové rozhraní pro velké objemy dat,
- rychlost 400 Mb/s (FW400, 1394a – 1995),
- rychlost až 800 Mb/s (FW800, 1394b – 2003),
- peer-to-peer topologie (síť může ovládat každé způsobilé zařízení),
- více zařízení lze propojit kaskádně nebo větvením,
- 6 vodičový kabel (4 datové).



24

Druhy paralelních portů

- Standard Parallel Port (SPP) (Centronics)
- Enhanced Parallel Port (EPP)
- Extended Capabilities Port (ECP)

Centronics:

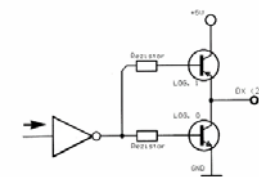
- napěťové úrovně TTL,
- doporučená maximální délka kabelu 3 až 5 m,
- přenos na principu Handshake,
- osmibitový výstup pro tiskárnu,
- omezená možnost monitorování stavu tiskárny,
- rychlost 100 - 200 kB/s.

25

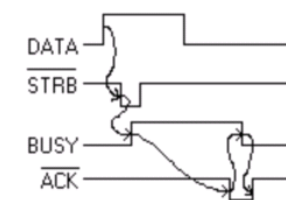
Centronics (SPP)

Hlavní signály:

datové vodiče (standardně 8) – každý stíněný GND vodičem (z PC),
Strobe – řídicí signál oznamující platnost dat (z PC),
Busy – oznamuje, že zařízení není připraveno k přenosu (z periferie),
Acknowledge – předávání stavových informací (z periferie),
+ další (Error, PE, Select, Init, ...).



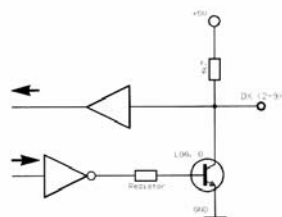
Datový pin SPP



26

EPP

- od roku 1991 (Intel, Xircom, Zenith DS),
- používá se pro interaktivní komunikaci s adaptérem LAN, mechanikou CD ROM, diskem, páskovou jednotkou
- 8-bitový vstup a 8-bitový výstup,
- rychlost 500 - 2 000 kB/s.



Datový pin EPP a ECP

27

ECP

- Hewlett-Packard LaserJet 4, Microsoft,
- obousměrný osmibitový přenos,
- vysokorychlostní přenos bloků dat,
- podpora datové komprese, kompresní poměr je 64:1,
- rychlejší než EPP (využívá DMA),
- rychlost přenosu > 2 MB/s.

28

Čipová sada (chipset)

Sestava integrovaných obvodů na základní desce - vlastnosti:

- řídí komunikaci na základní desce;
- určuje typy, příp. počet procesorů;
- určuje kapacitu a typy pamětí;
- určuje typ a počet rozšiřujících slotů, popř. integrované prvky (grafická, síťová nebo zvuková karta);
- obsahuje rozhraní pro FDD a HDD (P-ATA, S-ATA);
- podpora Plug and Play, USB, příp. IEEE1394;
- funguje jako řadič operační i vyrovnávací paměti;
- definuje rozhraní (systémovou sběrnici) mezi CPU a okolím;
- může řídit Power Management.

29

Čipová sada - architektura

Nejrozšířenější: Intel, VIA, nVidia (nForce, GeForce), ATI (Radeon, CrossFire) - vzájemně nekompatibilní.

Architektura: **North Bridge – South Bridge – Super I/O**

North Bridge – rozbočovač řadiče pamětí (MCH – Memory Controller Hub) je přes FSB (Front Side Bus) spojen s CPU;

- propojuje paměťovou a grafickou sběrnici;
- pracuje na frekvenci FSB, pro níž je řadičem;
- zajišťuje rychlý přesun velkého množství dat (\Rightarrow nutné chlazení);

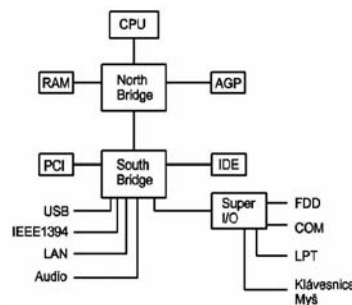
South Bridge – rozbočovač řadiče vstupů a výstupů (ICH – I/O Controller Hub), spojuje středně rychlé sběrnice PCI, USB, IDE, LAN, audio, zajišťuje funkce BIOSu.

30

Blokové schéma chipsetu

Super I/O – pro práci s pomalejšími vstupy a výstupy
(v současnosti jeho funkci přebírá South Bridge)

Trendem se přesouvání řadiče paměti DDRx a řadiče sběrnice PCI-e přímo do CPU (mizí North Bridge?)



31

Sběrnice FSB (Front Side Bus)

Systémová sběrnice pro komunikaci mezi CPU a základní deskou (North Bridge, respektive s pamětmi) – pomalejší než interní frekvence jádra procesoru, aby periferie zvládaly komunikaci.

Násobitel (Multiplier) - udává, kolikanásobně je procesor rychlejší než externí sběrnice FSB (roste po hodnotách 0,5). Pokud chceme procesor přetaktovat (overclocking), musíme změnit buď frekvenci FSB nebo násobitel.

FSB přenáší data na sestupné i vzestupné hraně hodinového signálu, rychlosti se postupně zvyšují (do 1600 MHz).

Propustnost až 12,8 GB/s ($1,6\text{GHz} \times 2\text{B} \times 2 \left\lfloor \frac{1}{2} \right\rfloor \times 2\text{směry}$).

32

HyperTransport

Vylepšená sběrnice FSB – duální jednosměrná sériová sběrnice (pro každý směr komunikace jedna sběrnice) založená na principu paketů (point-to-point)

- vysoká propustnost (mezi CPU a North Bridge),
- nízké zpoždění (latence),
- podpora více procesorů,
- nízké napájení (méně ztrátového tepla),
- může mít obecnější použití (nejen mezi CPU a North B.).

Použita poprvé v procesorech AMD v roce 2001, různé verze: 1.x (0,8GHz), 2.0 (1,4GHz), 3.0 (2,6GHz), 3.1 (3,2GHz).

V současnosti 32-bitová (pro každý směr), frekvence 3,2 GHz, \Rightarrow propustnost 51,2 GB/s ($4B \times 3,2GHz \times 2 \left(\downarrow \uparrow \right) \times 2směry$).

33

QuickPath Interconnect (QPI)

Náhrada FSB od fy Intel (2008), obdoba HyperTransportu

Skládá se ze dvou jednosměrných sběrnic point-to-point
2 x 21 signálů (16 data + 4 řídicí + 1 hodinový),
všechny signály jako diferenciální páry \Rightarrow 84 vodičů

Přenos dat na obě strany hod. signálu.

Rychlosti: 2,4 GHz, 2,93 GHz, 3,2 GHz
propustnost 25,6 GB/s ($2B \times 3,2GHz \times 2 \left(\downarrow \uparrow \right) \times 2směry$).

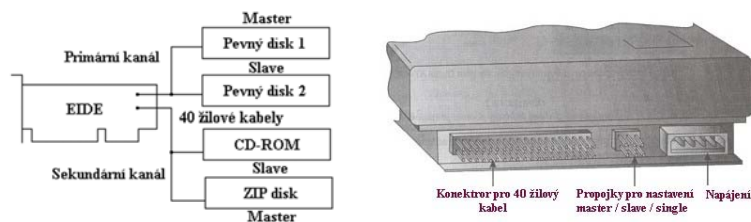
QPI sběrnic může být v CPU i více (zejména u vícejádrových)
- distribuce původní systémové sběrnice

34

P-ATA (IDE, ATA)

Integrated Drive Electronics,
(Parallel Advanced Technology Attachment)

- 40-žilový kabel (80-žilový), max. 45cm;
- na každý kanál lze připojit až 2 zařízení (master – slave);
- ATAPI (ATA Packet Interface) – pro CD-ROM, ZIP, LS-120.



35

Přenos dat P-ATA

Přenos dat mezi zařízeními (HDD, CD-ROM, ZIP, LS-120) a operační pamětí může být realizován dvěma režimy:

- PIO (Processor Input Output)** - přenos dat je řízen procesorem (veškerá data procházejí přes procesor);
- DMA (Direct Memory Access)** – speciální obvod řídí přenos dat bez účasti procesoru (varianty DMA: Single-word, Multi-word, Ultra).

36

P-ATA - přehled

Specifikace	ATA-1 IDE	ATA-2 E-IDE	ATA-3 Fast ATA	ATA-4 UltraATA	ATA-5 UltraDMA	ATA-6	ATA-7
Přenosové módy	PIO1	PIO4 DMA2	PIO4 DMA2	PIO4 DMA2	PIO4 DMA3 UDMA4	PIO4 DMA3 UDMA5	PIO4 DMA3 UDMA6
Max. přenosová rychlost (MB/s)	8,3	16,7	16,7	33	66	100	133
Rok uvedení	1986	1994	1997	1998	1999	2000	2002
Charakteristické vlastnosti	Asyn-chronní přenos, 512 MB	Syn-chronní přenos, 4 zařízení	Techno-logie SMART	Přenos na obě hrany, CRC, 80 vodičů	80 vodičů	Nap. 3,3 V	

37

S-ATA

Serial Advanced Technology Attachment

- přenosová rychlost S-ATA – 150 MB/s (1,5 Gb/s),
S-ATA II – 300 MB/s (3 Gb/s),
S-ATA III – 600 MB/s (6 Gb/s)
- na jednom kabelu jedno zařízení (point-to-point)
- délka kabelu až 1 m, diferenciální přenos signálů, dva datové páry a 3 vodiče na stínění
- napájení cca 0,5 – 0,6 V (menší spotřeba)



38

Propustnost P-ATA vs. S-ATA

Ultra ATA 133:

16-bitová sběrnice (2 byty), 2 transakce za 1 periodu
 $33,3 \text{ MHz} \times 2 \times 2 = 133 \text{ MB/s}$

Serial ATA:

sériový přenos bit po bitu, kódování 8b/10b
 $1500 \text{ MHz} \times 0,8 / 8 = 150 \text{ MB/s}$
 (S-ATA III má rychlost 4násobnou)

39

PCI

Peripheral Component Interconnect (1992)

- lokální sběrnice, komunikace již neprobíhá po I/O sběrnici (např. jako u sběrnice ISA);
- od systémové sběrnice oddělena řadičem s akcelerátorem;
- kmitočet na sběrnici 66MHz (dříve 33 MHz);
- datová sběrnice 32 nebo 64 bitů (paralelní);
- propustnost od 132MB/s do 528 MB/s (8B x 66 MHz);
- napájení 5V/3.3V/1.5V;
- dvoustavová modulace (log.0 – log.1).

40

AGP (Accelerated Graphics Port)

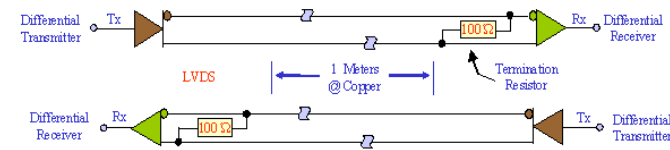
- vychází z PCI, dokáže však přenos zrychlit
- režimy 1x – 2x – 4x – 8x (násobky frekvence PCI sběrnice)
- point-to-point sběrnice
- díky technologii DiME (Direct Memory Execute) může přistupovat přímo k operační paměti bez zásahu procesoru.

Typ	Frekvence	Napětí	Propustnost
AGP (1x)	66 MHz	3,3 V	266 MB/s
AGP 2x	133 MHz	3,3 V	533 MB/s
AGP 4x	266 MHz	1,5 V	1066 MB/s
AGP 8x	533 MHz	0,8 V	2,1 GB/s

41

PCI Express (PCI-e)

- Sériová sběrnice složená ze dvou nízkonapěťových diferenciálních párů (vysílacího a přijímacího) - lane
LVDS (Low-Voltage Differential Signaling);
- propustnost 2,5 Gb/s na 1 lane, kódování 8b/10b;
 - architektura point-to-point (žádné sdílení sběrnice);
 - vícetavová modulace (korekce ztrát).



42

PCI Express - přehled

sdužování lane do linků – snadné zvyšování propustnosti

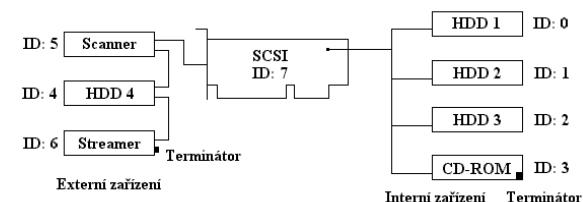
Počet lanů v linku	x1	x2	x4	x8	x12	x16	x32
Rychlost [Gb/s]	2,5	5	10	20	30	40	80
Propustnost [MB/s]	250	500	1000	2000	3000	4000	8000

43

SCSI

Small Computer Systems Interface – sběrniceová topologie

- rozhraní pro připojování diskových a jiných zařízení k počítači (scanner, CD drive, Zip);
- zařízení se propojují do řetězce, řetězec musí obsahovat řadič, připojení až 15 zařízení na jeden kanál;
- provedení Wide používá dvoubytový přenos.



44



SCSI - přehled

Standard	Frekvence [MHz]	Šířka sběrnice [bit]	Přenosová rychlost [MB/s]	Max. počet zařízení
SCSI-1	5	8	5	8
Fast SCSI-2	10	8	10	8
Wide SCSI-2	5	16	10	16
Fast Wide SCSI-2	10	16	20	16
Ultra SCSI-3	20	8	20	8
Ultra Wide SCSI-3	20	16	40	16
Ultra 2 SCSI	40	8	40	8
Ultra 2 Wide SCSI	40	16	80	16
Ultra 3 SCSI	80	8	80	8
Ultra 3 Wide SCSI	80	16	160	16
Ultra 320 SCSI	160	16	320	16



Rozhraní SAS

Serial Attached SCSI – nástupce SCSI rozhraní (topologie „point-to-point“), použití v serverech a diskových polích;

- rychlost SAS: 3 Gb/s, SAS 2.0 – 6 Gb/s;
- lze připojovat i odpojovat za provozu;
- každý disk má dva nezávislé komunikační kanály (vhodné pro redundanci např. v diskových polích – možné připojit na záložní řadič);
- rozšiřitelnost pomocí externích expandérů (stromová topologie);
- kompatibilní s normou SATA (lze kombinovat i v rámci jednoho řadiče) – disky SAS nelze připojit k řadiči SATA;
- HDD SAS mají větší otáčky (10-15 tis. ot/min), ale menší velikosti (stovky GB) vůči HDD SATA.

Speciální procesory (signálové, grafické, mobilní)

Milan Kolář
Ústav mechatroniky a technické informatiky

Signálový procesor (DSP)

Mikroprocesor nebo mikroprocesorový systém, který je přizpůsoben pro efektivní realizaci algoritmů číslicového zpracování signálu v reálném čase s co nejmenšími vynaloženými náklady.

Architektury signálových procesorů mají různý stupeň paralelizmu zpracování instrukcí (závisí na množství instrukcí, které lze provést během jednoho cyklu, a na množství provedených operací v každé instrukci) a paralelního zpracování dat.

- úroveň paralelizmu omezuje technologie výroby IO.

Předpokládá se průběžné zpracování velkého množství dat „protékajících“ procesorem (na zpracování dat se aplikují různé algoritmy DSP).

2

Požadavky

- zpracování toku dat v reálném čase (vysoký pracovní kmitočet, využití co nejvyššího stupně paralelizmu),
- velký matematický výkon,
- malé nároky na paměť (a tím malá plocha čipu a nízký příkon) u zpracovávaných algoritmů,
- efektivní instrukční sada (úsporné programy),
- výkonné kompilační programy z vyšších programovacích jazyků,
- nízká cena, dostupnost na trhu,
- přenositelnost programů mezi generacemi mikroprocesorů,
- snadná modifikace zpracování toku dat.

3

Operace signálových procesorů

Nejčastější algoritmy číslicového zpracování signálů:

- konvoluce
- číslicová filtrace (IIR, FIR)
- diskrétní transformace
- korelace
- práce s maticemi

$$y_n = \sum_{i=0}^M h_i \cdot x_{n-i}$$

$$y_n = \sum_{i=0}^M a_i \cdot x_{n-i} - \sum_{i=1}^L b_i \cdot y_{n-i}$$

$$r_n = \frac{1}{N} \sum_{i=0}^{N-1} x_i \cdot x_{n+i}$$

Důraz je kladen na násobení hodnot uložených v datové paměti konstantou nebo proměnnou a na akumulaci těchto dílčích součinů, dále aritmetické a logické posuny.

4

Signálové procesory

1979 – první signálový procesor I2920 (Intel)
výrobci – Texas Instruments, Analog Devices, Freescale (dříve Motorola).

Výpočetní výkon – hodnocen MIPS, MFLOPS, MOPS, MMACS.

Porovnání výkonnosti architektur signálových procesorů:

- množství instrukcí, které lze provést během jednoho hod. cyklu,
- počet instrukcí provedených paralelně,
- množství provedených operací v každé instrukci.

Signálové procesory - dělení

Dělení podle (dynamického) rozsahu: 16, 24, 32 bitové
(závisí na použití – zpracování hlasu v telefonii, zvuku v HIFI technice, obrazu, 3D grafiky).

Dělení podle periférií:

DSP mikroprocesory (neobsahují paměť ani další periferie),
DSP mikrokontroléry (obsahují rozličné periferie – např. A/D a D/A převodníky).

Částečně dochází k prolínání mikrořadičů a signálových procesorů
=> hybridní mikrokontroléry.

Dělení podle aritmetiky

Rozdělení podle typu zpracovávaných dat:

- *procesory v celočíselné aritmetice*
levné, ale náročnější vývoj algoritmů, nutná tzv. normalizace;
- *procesory s pevnou řádovou čárkou*
rychlejší, levnější, menší přesnost a rozsah (snadné přetečení), kompromis mezi celočíselnou a FP;
- *procesory s pohyblivou řádovou čárkou* (32-bitové)
složitější struktura procesoru, větší spotřeba energie, malý kvantizační šum, větší rozsah, programování převážně ve vyšších programovacích jazycích, jednodušší algoritmy.

Architektury signálových procesorů

DSP má většinou:

- harvardskou architekturu (zejména proto, že kód i data mají zvláštní sběrnice, možnost uchovávat data i v paměti programu);
- omezenou instrukční sadu (RISC);
- zřetěžené zpracování dat (pipeline) => zvyšování kmitočtu;
- kapacitní přenos operandů k ALU nebo MAC
=> větší počet adresových a datových sběrnic (2 až 4)
vně procesoru vyvedena většinou jen jedna AS a DS.
- vysoký vnitřní paralelismus (superskalární, speciální typy LIW a VLIW), někdy se využívají architektury paralelních systémů, příp. shlukování DSP do výpočetních sítí.

Hlavní části běžných DSP

Hlavní aritmetická jednotka (jednotlivé části nezávislé):

- jednotka MAC (Multiplier and Accumulator) – paralelní násobička a sčítačka (na 1 strojový cyklus) – velikost střadače dána součtem velikostí operandů + 8 bitů;
- jednotka ALU (+, -, *, /, inkrementace, dekrementace, log. fce);
- jednotka podporující paralelní posuny (barell-shifter).

Generátory adres DAG (Data Address Generator) - správa adres pro čtená data a konstanty, nejčastěji dva (pro konstanty a pro data) – podporují různé druhy adresování (inkrementace, dekrementace, reverzní bitové adresování, adresování v kruhovém zásobníku apod.).

Čítač instrukcí – může opakovat jednu či více instrukcí.

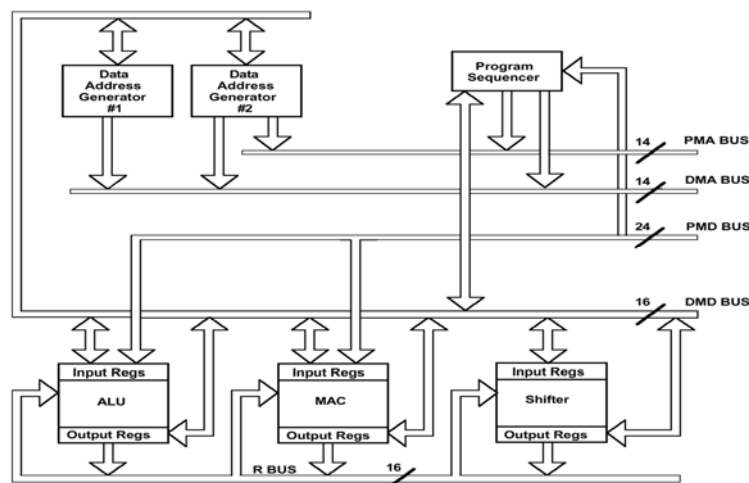
9

Další možné části DSP

- nezávislé sběrnice (PMA, PMD, DMA, DMD, R bus);
- jedno- nebo dvouportová paměť RAM pro data, příp. několik nezávislých paměťových bank;
- rychlé sériové kanály (převážně synchronní);
- programová paměť – nejčastěji Flash, ale i ROM;
- obvodová zásobníková paměť, množství pomocných registrů;
- DMA kanály se speciálními sběrnicemi (i vícenásobné);
- přerušovací systém (+ zrcadlové sady registrů);
- čítače/časovače, A/D převodníky;
- další standardní rozhraní (RS-232, SPI, CAN, I2C, USB, Ethernet);
- generátor hodin (s externím krystalem), fázový závěs.

10

Architektura běžných DSP



11

Paralelní systémy

Třídění paralelních systémů podle počtu programů:

- **SI** (Single Instruction Stream) – v čase řešení problému běží jeden program;
- **MI** (Multiple Instruction Stream) – během řešení běží více programů paralelně.

Třídění podle toků dat (podle počtu zpracovávaných dat. souborů):

- **SD** (Single Data Stream) – jeden zpracovávaný tok dat;
- **MD** (Multiple Data Stream) – více zpracovávaných toků dat.

12

Čtyři kategorie paralelních systémů

- SISD** – chápán jako klasický von Neumannův počítač s jedním programem a jedním sériově přiváděným tokem dat;
- MISD** – hypotetická kombinace několika programů zpracovávajících jeden tok dat (nezaměňovat se zřetězeným zpracováním instrukcí, kde je vždy aktivní pouze jeden program), lze uvažovat vektorové počítače;
- SIMD** – větší počet funkčních jednotek pracujících na řešení téhož programu (všechny jednotky provádí současně tutéž instrukci, ale každá s jinými daty);
- MIMD** – obecný typ paralelního systému, který obsahuje jednotky již tak samostatné, že každá z nich řeší samostatný program a zpracovávají jiná data.

13

Paralelizmus zpracování instrukcí

Čtení s jedním přístupem (Single Issue Processing)

řídící jednotka čte pouze jednu instrukci během jednoho hod. taktu, ale paralelizmus je realizován sdružením několika operací do jedné instrukce (harvardská architektura)
=> SIMD – paralelní zpracování dat

Tím je zahušťován programový kód a šetřena programová paměť, ale znamená to řadu omezení (které registry se musí použít pro které operace a které operace lze sdružit do jedné instrukce) – toto rovněž ztěžuje vytváření kompilátorů.

14

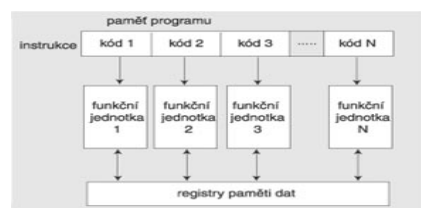
Paralelizmus zpracování instrukcí

Metoda čtení s více přístupy (Multi-Issue Processing)

provádí se jedna operace v jednom instrukčním cyklu, přitom se čte skupina instrukcí paralelně (vícenásobné čtení)
– paralelní zpracování instrukcí

Jedna instrukce je složena z několika částí, které současně ovládají skupinu funkčních jednotek

- Superskalární architektura
- Architektura VLIW



15

Superskalární architektura

spíše typická u procesorů pro všeobecné použití

závislost registrů a dat na instrukcích řeší za běhu programu sám procesor – *jednotka Schedule Unit*
=> jednodušší na programování, větší hustota kódu;

instrukce pro dílčí jednotky jsou jednoduché
=> jednodušší kompilátory, větší rychlost;

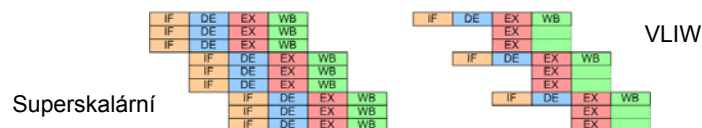
umožněna binární kompatibilita mezi generacemi procesorů (menší závislost na konkrétní architektuře procesoru);

programátor přesně neví, které instrukce procesor seskupí pro paralelní zpracování => problematický odhad celkového počtu instrukčních cyklů (problematická optimalizace).

16

Architektura VLIW

- VLIW (Very Long Instruction Word) – architektura se čtením s více přístupů (obdoba předchozí architektury pro signálové procesory) – delší instrukce sestavené z dílčích částí, které mohou být provedeny paralelně (instrukční packet) - např. 8 dílčích 32-bitových subinstrukcí (256 bitů)
- o paralelizaci instrukcí rozhoduje překladač, případně samotný programátor (HW nekontroluje hazardy);
 - větší náročnost na programovou paměť (větší programy se synchronizačními NOPy);
 - různé generace VLIW jsou obecně binárně nekompatibilní.



17

Vícejádrové DSP procesory

- Signálové procesory s velkým výkonem (jednojádrové):
frekvence 200 až 1200 MHz, výkon řádově jednotky GIPS
- Trend k vícejádrovým CPU (jako u univerzálních CPU)
- Např. TMS320C6472 od firmy Texas Instruments
- CPU s pevnou řádovou čárkou, 6jádrový DSP, 32bitový
 - frekvence až 700 MHz
 - 8 instrukcí/cyklus v jádře → $6.8.700 = 33600$ MIPS
 - 8 MAC 16×16 v jádře → $6.8.700 = 33600$ MMACS
 - cache L1 6 x 64kB, L2 4416 kB (768 kB sdílená)
 - energetická spotřeba 0,15 mW/MIPS
 - napětí jádra 1.2 V, I/O buňky 1.2 V, 1.5 V, 1.8 V, 3.3 V
 - I2C, Ethernet 10/100/1000, 12 x 64bit. časovače,
 - Serial RapidIO Link, DDR2 memory controller, JTAG

18

Grafické procesory (GPU)

- Speciální řídicí procesory zajišťující vykreslování dat (nejčastěji data uložená v grafické paměti a jsou obrazována na monitoru) – zpracovává 3D geometrii na 2D obraz.
- Grafický procesor může být mnohdy výrazně výkonnější než hlavní CPU počítače – architektura se přizpůsobuje charakteru obrazových dat – stream processing (SIMD – jeden postup nad velkým množstvím dat) – paralelní architektura.
- Bývá na grafické kartě, v severním můstku nebo přímo v CPU.
- Nejčastěji pracují s vlastní grafickou pamětí – velká a rychlá.
- Nejvýznamnější výrobci GPU:
- nVidia, AMD (ATI), Intel, VIA Technologies, Matrox.

19

GPU – hlavní části

- Shadery** – plně programovatelné části GPU (speciálními programovacími jazyky) - spíše superskalární zřetěžené architektury, operace nejčastěji v pohyblivé řádové čarce; někdy jsou shadery chápány jen jako programy řídicí části GPU;
- Řadič paměti** – stará se o komunikaci grafickou pamětí a GPU;
- Jednotka TMU** (Texture Mapping Unit) – nanáší textury na objekty;
- Jednotka ROP** (Render Output Unit) – stará se o konečný výstup dat z grafické karty (2D).



20

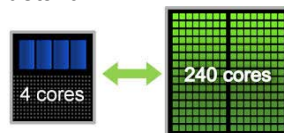
Shadery

Vývoj z 2D a 3D grafických akceleratorů,
po roce 2000 se objevují shadery – zprvu samostatné specializované vertex a pixel, později geometry shadery:

- *Vertex shader* – provede danou transformaci na každém polygonu (vrcholu trojúhelníků), vytvoří 3D scénu, odstraní neviditelné prvky, nasvítí scénu;
- *Pixel shader* – provede operaci na každém pixelu (potahování texturou, stínování, obarvení, odlesky);
- *Geometry shader* – umožňuje přidávat a odebírat vrcholy (doplňování detailů existujícího modelu);
- *Shader pro teselaci* – přidávání dalších detailů.

V současné době tzv. **unifikované shadery**

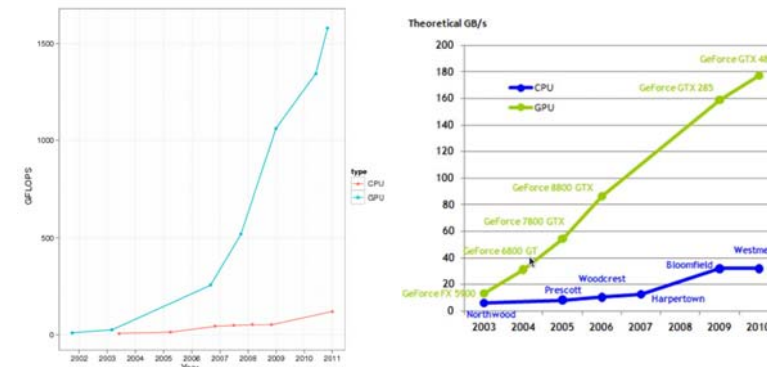
- zamezení přelévání dat mezi specializovanými shadery;
- velké množství jednodušších jader.



21

CPU vs. GPU

Srovnání výkonu a propustnosti



22

Mobilní procesory

Významný podíl procesorů ARM – velký výkon, nízká spotřeba, nízká cena (cca 98% mobilních zařízení – zejména mobilní telefony, navigace, tablety, ...)

Firma ARM procesory pouze vyvíjí, pak prodává licence => řada výrobců: největší asi Qualcomm – např. generace Snapdragon, Samsung (řada Exynos), Apple, Texas Instruments, nVidia, Intel (Atom), ...

Podpora v OS (Android, Windows CE, Symbian, ...).

Významnou částí mobilních procesorů jsou grafické čipy (např. řada Mali u ARM, řada Adreno u Qualcomm) nebo DSP jádra.

2011 – první mobilní čip se dvěma jádry (nVidia Tegra 2);

2012 – první 4jádrový CPU (nVidia Tegra 3).

V současnosti až 8jádrové, 32/64bitové, 1,0-2,5 GHz.

23

Mobilní procesory (pokrač.)

Největší důraz kladen na úsporu energie:

Architektura typu „big.little“ - kombinace různých jader s různou výkonností (není-li potřeba výkon, zapnou se pouze pomalejší a úspornější jádra) – přepínání jádra však vyžaduje množství hod. cyklů (řádově 10^4), což je energeticky a časově náročné.

Power gating – nevypínají se celá jádra, ale dochází ke snížení napětí nepoužívaných logických obvodů (až na nulu) nebo se vypínají jen určité části – pak se rychleji „probouzejí“.

Clock gating - snižování pracovní frekvence procesoru při nečinnosti.

Stále větší důraz na úsporu energie periferií (displej, SRAM, Wi-Fi adaptér, grafika, ...).

24

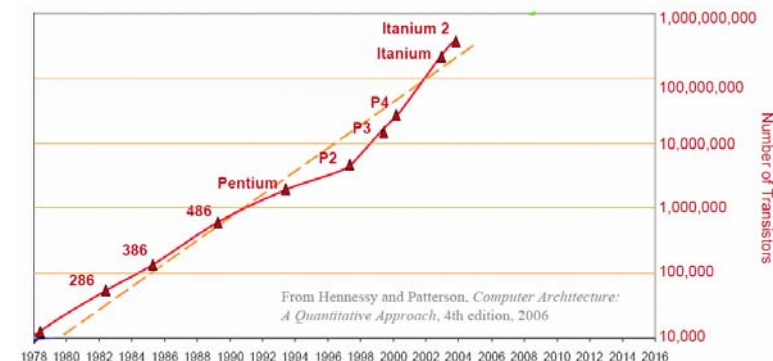
Procesory II.

Milan Kolář

Ústav mechatroniky a technické informatiky

Moorův zákon

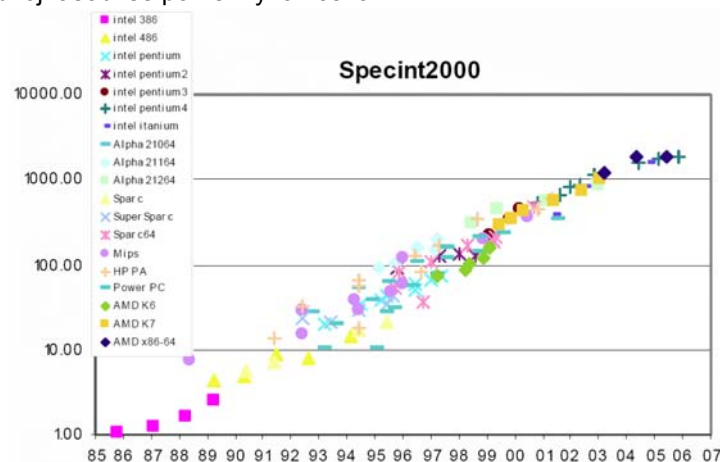
1965 – Gordon Moore – počet tranzistorů na čipu se bude exponenciálně zvyšovat - zdvojnásobení každých cca 18 až 24 měsíců (a cena se sníží na polovinu) - stále platný



2

Moorův zákon (modifikovaný)

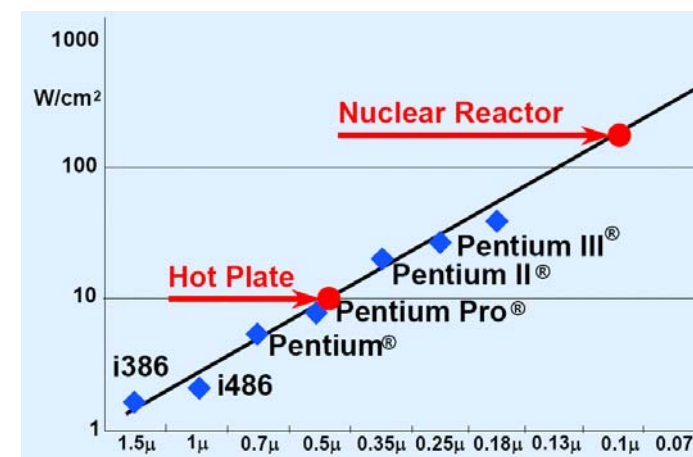
zdvojnásobí se poměr výkon/cena



3

Hustota výkonu

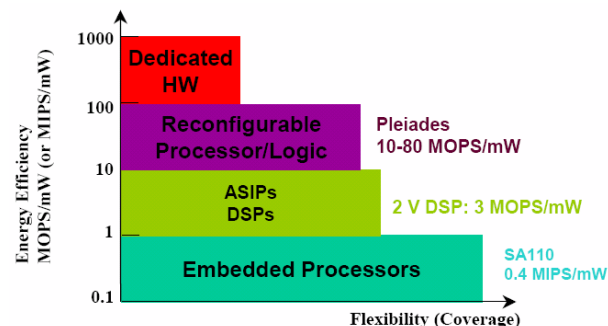
hlavní problém při zvyšování hustoty funkcí na čipu



4

Výpočetní (energetická) účinnost

Computational (energy) efficiency – počet operací (resp. instrukcí) za sekundu vztažený ke ztrátovému výkonu na daném čipu (energetická účinnost)



5

Procesy a vlákna

Proces i vlákno (thread) – posloupnost instrukcí viděná z hlediska jejich vykonávání, řízená operačním systémem.

Charakterizují se identifikací, vlastníkem, stavem, adresovým prostorem, sadou registrů, zásobníkem, čítačem instrukcí,...

Vlákna jemněji dělí proces – rychleji se synchronizují a přepínají (přepínání kontextu) – nutné pro zajištění multitaskingu.

Multitasking - schopnost operačního systému (resp. počítače) zpracovávat několik úloh (procesů) najednou v multiplexním provozu (alespoň zdánlivě).



6

Multitasking

Dělení z hlediska realizace:

- skutečný** – běží více úloh současně, např. víceprocesorové systémy, ne všechny úlohy musí běžet současně; (multiprocesový vs. multivláknový multitasking),
- zdánlivý** – běh úloh se rychle střídá; rychlejší přepínání kontextu, úspornější z hlediska paměti (dnes častější).

Dělení podle způsobu přidělování a odebrání časových kvant zdánlivého multitaskingu:

- kooperativní,**
- preemptivní.**

7

Kooperativní multitasking

též **nepreemptivní** – vyžaduje aktivní spoluúčast běžících úloh (každá je povinná dostatečně často voláním systémové funkce předat řízení zpět operačnímu systému, který díky tomu může spustit jinou úlohu);

příliš se již nepoužívá (např. MS Windows 3.1 nebo Mac OS);

výhodou je jednodušší implementace v operačním systému;

nevýhodou je, že chybně naprogramovaná úloha, která nevrátí řízení OS, způsobí zastavení ostatních úloh.

8

Preemptivní multitasking

O přidělování a odebírání procesoru jednotlivým úlohám plně rozhoduje operační systém - v pravidelných intervalech (typicky zhruba po 10 ms) přeruší provádění běžícího programu, vyhodnotí aktuální situaci (které úlohy žádají o přidělení procesoru, jejich priority atd.) a nechá běžet buď opět původní nebo jinou úlohu;

běžící úloha může dobrovolně požádat o přepnutí kontextu (vzdát se zbytku svého kvanta), pokud chce na něco čekat;

používá se ve MS Windows od verze 95, Linux

výhodou je, že nedochází k „zatužení“ počítače;

nevýhodou je složitější implementace OS a vyšší nároky na HW.

9

Multiprocessing

Využití více procesorů pro provádění počítačových úkolů.

Rozdělení úloh může být:

symetrické – právě volný procesor lze zapojit do řešení probíhající úlohy;

asymetrické – každý procesor má pevně přidělenou práci (nelze jej používat jinde).

Nutno používat „MP-ready“ aplikace a operační systémy.

10

Technologie Turbo Boost

Technologie umožňující jednotlivým jádrům na vyšší frekvenci, pokud to okolnosti dovolují – závisí na počtu aktivních jader, na očekávané spotřebě a na teplotě procesoru.

Aktuální zejména při nerovnoměrném vytížení jednotlivých jader CPU; automaticky se zvyšuje násobí u jednoho či více jader (po 133 MHz).



11

Technologie Anti-Theft (Intel)

HW podpora ochrany v CPU (+ placená služba) – nutná registrace; zablokování počítače při krádeži (uživatel musí aktivovat přes internet); sledování pohybu na základě lokace IP adresy; automatické zablokování, nedojde-li do určité doby k synchronizaci se serverem.

Po registraci si nainstalujeme SW (aktivace)

U CPU Core i3/i5/i7 (od Sandy Bridge), podpora v OS od Win 7



12

IA-64 (Intel Architecture)

Nová „architektura“ speciálně určená pro 64bitové procesory

(Intel + HP → CISC + RISC) – post-RISC architektura;

- EPIC (*Explicitly Parallel Instruction Computing*);
- paralelní zpracování instrukcí (superskalární CPU);
- detekce paralelních instrukcí je plně na kompilátoru (CPU neztrácí čas vyhodnocováním instrukcí);
- používá „svazky instrukcí“ - Bundled Instructions (společně prováděné instrukce jsou pro rychlejší zpracování vloženy do jednoho dlouhého slova – tzv. VLIW kódování (*Very Long Instruction Word*));
- většinou se zachovává kompatibilita s instrukcemi IA-32 (díky překladači) za cenu nižšího výkonu těchto instrukcí.

13

Významné parametry procesorů

- velikost datové sběrnice ... 16, 32, 64 bitů;
- velikost cache L1 ... 8 – 128 kB (vztaženo na 1 jádro);
- velikost cache L2 ... 64 kB – 3 MB; 1 MB/jádro;
- frekvence FSB (Front Side Bus) ... 200 – 1600 MHz;
HyperTransport 3.2 GHz, QPI 3.2 GHz;
- frekvence jádra ... 1500 – 4000 MHz;
- tepelný výkon ... 30 – 130 W;
- napětí jádra procesoru ... 0,9 – 2,5 V;
- maximální teplota jádra ... 70 – 95 °C;
- technologie výroby ... 90 nm, 65 nm, 45 nm, 32 nm, 22 nm, 14 nm;
- pouzdro (socket) ... 478, 754, 775, 939, 1156, 1366, 2011.

14

Vývoj procesorů Intel

Intel 8086 (1978), IBM PC XT, 16bitový, 2stupňová pipeline, skalární, CISC, 29 tisíc tranzistorů, 5-10 MHz, 0,33-0,66 MIPS, 3 μm, 20bitová AS (OP max. 1 MB), architektura „x86“.

Intel 80286 (1982), IBM PC AT, 16bitový, 3stupňová pipeline, 134 tisíc tranzistorů, 8-25 MHz, 1,2-3,75 MIPS, 24bitová AS (OP max. 16 MB).

Intel 80386 (1985), 32bitový, 3stupňová pipeline, 275 tisíc tranzistorů, 19-40 MHz, 6-15 MIPS, 1,5 μm, 32bitová AS (OP 4 GB), IA-32.

Intel 80486 (1989), 32bitový, 5stupňová pipeline, 1,25 mil. tranzistorů, 25-50 MHz, 22-44 MIPS, 0,8 μm, 32bitová AS (OP 4 GB), sdružuje FPU, prvky RISC (nejčastější instrukce má přímo zahrnuty v log. obvodech - nepotřebuje pro ně mikroprogram)

15

Intel Pentium 80586 + PRO

1993 – 3.1 mil. tranzistorů, 0,8 μm, 5 V, 60-66 MHz, 102-112 MIPS, L1 2x8 kB, dvoucestná, možnost multiprocessingu (2 CPU).

2 instrukce za takt (první superskalární procesor Intel); 32bitová AS, vnější datová sběrnice 64 bitů (vnitřně 32bitový), jednotka předvídání skoků.

1994 – 3.3 mil. tranzistorů, 0,6 μm, 3,3 V, 75-100 MHz, 120-160 MIPS.

1997 – 166-233 MHz, 0,28 μm, 4.5 mil. tranzistorů, L1 2x16 kB čtyřcestná.

1995 – **Pentium PRO**: 14násobná pipeline; 36bitová AS, 0,5-0,35 μm, 21 až 67 mil. tranzistorů, až 232 MIPS, provádění instrukcí mimo pořadí; spekulativní provádění instrukcí; přejmenovávání registrů; SMP max. 4 CPU,

16



Intel Pentium II + III + 4

Pentium II (1997) – 233-450 MHz, 0.25 μm , 650 MIPS, L1 2x16kB, L2 256-512 kB;

Pentium III (1999) – 0.25-0.13 μm , 1.5-2 V, 450-1133 MHz, vícenásobná predikace větvení;

Pentium 4 (2000) – 180-65 nm, 1.4-3.8 GHz, 42 mil. tranzistorů; 32bitový CPU, L1 8-16 kB, L2 256 kB až 1 MB; označení **NetBurst**; hyper zřetězení (20-31stupňová pipeline); L1 již neuchovává instrukce, ale mikroinstrukce (při opakování instrukce odpadá dekódování); mikrojádru s ALU pracuje na dvojnásobku nominální frekvence (velký TDP); technologie HT (Hyper-Threading) – u pozdějších variant P4.

17



Další vývoj CPU Intel

Intel Itanium (2001) – plně 64bitový CPU; 10stupňová pipeline, technologie 0,18 μm , 800 MHz (málo vůči P4); výkon 13 Gflops; ztráta kompatibility z předešlými programy (nemůže nativně vykonávat instrukce x86, x87, MMX apod.) – velká instrukční sada by snižovala výkon; přidán překladač 32bitových instrukcí IA-32 na IA-64.

Pentium Extreme Edition (2005) – první dvoujádrový CPU Intel - v podstatě se tváří jako dva nezávislé procesory (je schopen provádět paralelně dvě sekvence instrukcí); 3.2-3.73 GHz, L1 2x16 kB, L2 2x1-2x2 MB, 90-65 nm, 230 mil. tranzistorů, 1.25 V, TDP 130 W; aplikace a OS musí dokázat využívat (vícevláknové aplikace).

18



Mikro-architektury procesorů Intel

Postupné zvyšování výkonu a snižování spotřeby, podpora nových rozhraní, zvyšování frekvence pro RAM, zvyšování cache

Core 2 (od 2006) – 64bitový, 65 nm, FSB, 1-2-4-6 jader, zkrácen pipelining (12-14 stupňový), snížení frekvence, snížení ztrátového výkonu (spojování řad pro PC a NB).

Nehalem (2008) – 45-32 nm, 750 mil. tranzistorů, nově QPI, DDR3, 2-4-8 jader, L1 2x32 kB, L2 256 kB/jádru, nově sdílená L3 8 MB, nově technologie Turbo Boost, integrace řadiče DDR3 a PCI-e do CPU, GPU na jiném čipu (1 pouzdro), řady i3/i5/i7, návrat k HyperThreadingu.

Sandy Bridge (2011) – 32 nm, výrazné vylepšení GPU (na 1 čipu), 2. generace i3/i5/i7; nově System Agent – obdoba North Bridge přímo v CPU (řadiče PCI-e 3.0, DMI, DDR3), Media Engine

19



Mikro-architektury procesorů Intel

Ivy Bridge (2012) – 22 nm, 1,4 mld. tri-gate (3D) tranzistorů – snížení spotřeby (až 50 %), TDP 14-87 W, podpora PCI-e 3.0, DDR3L, USB 3.0, zvýšení výkonu GPU, 2-8 jader, 1.7-3.9 GHz.

Haswell (2013) – 22 nm, důraz na úsporu energie (TDP od 10 W), L3 2-20 MB, některé typy i L4 128 MB, 14-16stupňová pipeline, 2-18 jader, 4. generace i3/i5/i7.

Broadwell (2014) – 14 nm, důraz na nízkou spotřebu (pro mobilní zařízení – TDP od 3,5 W), zvýšení výkonu zejména u GPU, 5. generace i3/i5/i7.

Skylake (2015) – 14 nm, podpora DD4, 6. generace i3/i5/i7

20

Core i7-5960X (Intel)

Jeden z nejvýkonnějších CPU Intel (pro PC)

- 8jádrový, 64bitová architektura, frekvence 3.0/3,5 GHz; Haswell-E;
- technologie Hyper-Threading (až 16 vláken současně);
- technologie Enhanced Intel Speedstep (úspora energie);
- technologie Turbo Boost 2.0 (přetaktování jednotl. jader);
- sběrnice QPI (QuickPath Interconnect);
- L2 cache 0,25 MB/jádro, L3 cache 20 MB, HW virtualizace;
- integrovaná PCI Express 3.0;
- integrovaný 4kanálový řadič DDR4;
- výrobní technologie 22 nm;
- TDP 140 W, socket 2011-3.



21

Vývoj procesorů AMD

Am286 (1984) – licence od Intel (požadavek IBM), kompatibilní s 80286

Am386 (1991) – konkurence Intel, levnější

Am486 (1993) – snaha o výkonnější varianty vůči Intel, později snížení nap. napětí na 3.3 V a zvýšení frekvence (ztráta kompatibility, rozdílné základní desky)

K5 (1996) - superskalární procesor (konkurent Intel Pentium), 0.5-0.35 μm , 4.3 mil. tranzistorů, až 120 MIPS, 75-166 MHz

K6 (1997) – 0.35-0.25 μm , 8.8 mil. tranzistorů, 166-300 MHz, až 290 MIPS, FPU 8-17 MFlops, FSB 60-66 MHz;

K6-2 (1998) – 266-550 MHz, 0.25 μm , 9.3 mil. tranz., FSB až 100 MHz; 3DNow! – rozšíření o 21 instrukcí zrychlujících vytváření 3D scén.

K6-3 (1999) - 21.3 mil. tranzistorů, 4cestná L2 256 kB, až 490 MIPS.

22

Vývoj procesorů AMD

K7 (1999) – Athlon, Athlon XP, Athlon MP, Duron, Sempron

K8 (2003) – nová technologie x86-64 – plně kompatibilní s 32bitovou IA32 i 64bitovou x86, ale méně výkonná oproti IA-64; Athlon 64, Athlon 64 X2 (první vícejádrový CPU od AMD – 2005), Turion 64, Turion 64 X2, Sempron;

K10 (2007) – Phenom, Opteron (první 4jádrové procesory)

Bulldozer (2011) – 32 nm, L2 3-8 MB, L3 8MB, 4-16 jader, frekvence až 3.6/4.2 GHz, TDP 10-100 W,

FX-8350 (AMD)

Jeden z nejvýkonnějších CPU AMD (pro PC)

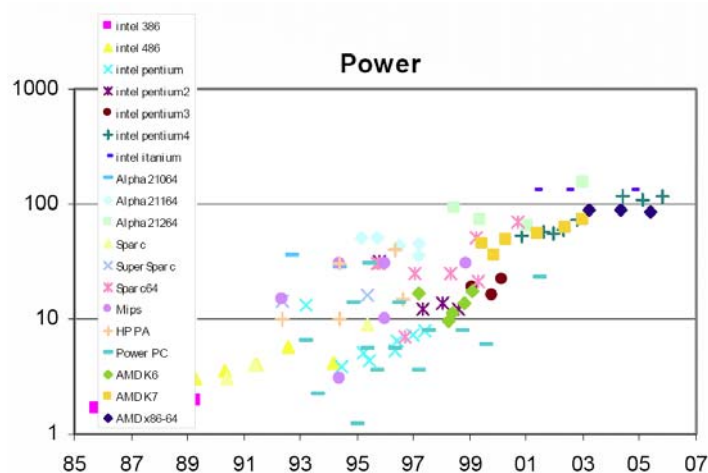
- 8jádrový (4 dvoujádrové bloky), frekvence až 4,2 GHz;
- 64bitový CPU;
- technologie Turbo Core 2.0 (přetaktování jednotl. jader);
- L2 cache 1 MB/jádro, L3 cache 8 MB (4 x 2 MB);
- integrovaný 2kanálový řadič paměti DDR3;
- výrobní technologie 32 nm;
- TDP 125 W, socket AM3+ (942 pinů);
- vůči Intel Core i7 lepší poměr cena-výkon.



23

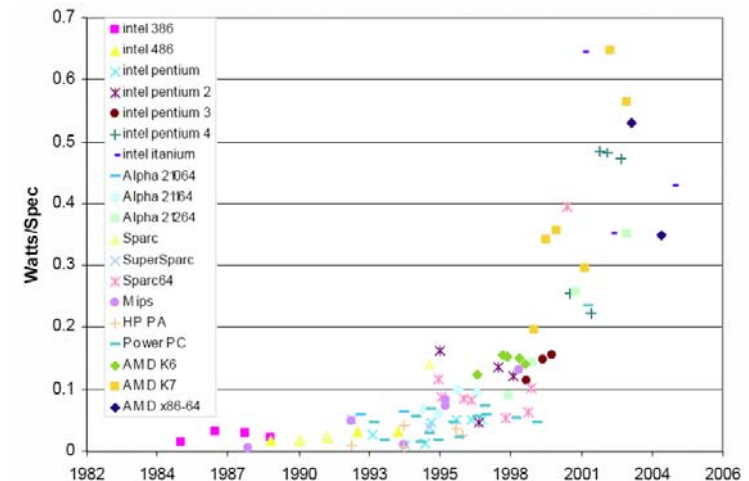
24

Spotřeba napájení (W)



25

Efektivita napájení



26

Trendy ve vývoji - shrnutí

- 80. léta
superskalární procesory - zřetězení (10 CPI → 1 CPI);
- 90. léta
zlepšování cache (2- až 6-cestné), out of order,
branch prediction (1 CPI → 0,5 CPI);
- 00. léta
vícejádrové procesory;
SoC (System on Chip), kombinace HW procesorů s hradlovými poli (embedded design);
softwarové procesory v FPGA (architekturu lze přizpůsobit, možné i víceprocesorové systémy).

27

Trendy ve vývoji - pokračování

- integrace grafických procesorů s vysokým rozlišením (4K);
- důraz na energetickou efektivitu (v závislosti na výkonu), podpora mobilních procesorů (pro ultrabooky a tablety);
- hardwarové funkce pro ochranu dat (Intel Anti-Theft);
- hardwarová podpora vzdálené administrace (bez nutné podpory OS) – Intel KVM Remote Control (propojeno s integrovanou grafickou kartou);
- nové instrukce pro standard pokročilého šifrování dat (Advanced Encryption Standard) – včetně dešifrování.

28

Paralelní multiprocessorové systémy

Milan Kolář

Ústav mechatroniky a technické informatiky

Paralelní multiprocessorové systémy

Jako paralelní označujeme systém, v němž může probíhat několik procesů současně (paralelně)

- snaha zvyšovat výkonnost nad hranici danou technologií výroby součástek (mikroprocesorů);
- rozměry, cena a energetická náročnost elektronických prvků klesá rychleji než roste jejich rychlost \Rightarrow vytváření paralelních systémů.

Výkonnost většinou neroste lineárně s počtem procesorů (má spíše logaritmický průběh) – vlivem komunikace CPU, synchronizace, nedokonalým vytížením, nevhodnými algoritmy, ...
- od jisté hranice je přidávání procesorů nerentabilní.

2

Paralelismus

Paralelní systémy dále členíme podle formy (granularity, zrnitosti) paralelního procesu – zajímají nás procesy od určitého stupně složitosti (např. systémy pracují paralelně s celými slabikami obvykle jako paralelní neoznačujeme).

Pro paralelní systémy je charakteristické, že procesory jsou obvykle soustředěny v poměrně malém prostoru (min. časové ztráty při přenosu informací) – v opačném případě mluvíme o distribuovaném (rozloženém) systému - počítačová síť.

3

Paralelizace algoritmů

Paralelismus je také přirozenou vlastností mnoha algoritmů, které jsou v jednoprocessorovém počítači prováděny neefektivně, např.: operace s vektory, maticemi, grafy, aritmetika vícemístných čísel, ...

Transformace sériového algoritmu na paralelní (automatická nebo ruční) – např. sériové procházení cyklů lze přidělit více procesorům (nejsou-li závislé).

4

Ukazatele paralelizace

1) Časová složitost (time complexity) $O(n, p)$

funkce, jejíž hodnota je pro konkrétní algoritmus úměrná maximální době jeho výpočtu

Např.

a) sekvenční algoritmus pro součet n čísel:

$$O(n) = n \quad (\text{doba výpočtu je lin. závislá na počtu čísel})$$

b) paralelní součet prováděný na $p = n/2$ procesorech

$$O(n, n/2) = \log n$$

Ukazatele paralelizace (pokračování)

2) Zrychlení (speedup) $S(n, p)$

poměr doby výpočtu nejlepšího známého sekvenčního algoritmu a doby výpočtu paralelního algoritmu na téže (paralelním) počítači, využíváme-li p procesorů
(výjimečně *superlineární zrychlení* – lepší než lineární)

Např. předešlý příklad: $S(n, n/2) = n / \log n$

3) Paralelní efektivita (efficiency) $E(n, p)$

jedná se o zrychlení dělené počtem použitých procesorů

Např. předešlý příklad: $E(n, n/2) = 2 / \log n$

Amdahlův zákon

Udává maximální zrychlení pro víceprocesorové systémy:

$$S \leq \frac{T(n)}{T(n, p)} = \frac{1}{f_s + \frac{f_p}{p}} \leq \frac{1}{f_s} \quad f_s = \frac{t_s}{t_s + t_p}$$

f_s ... část výpočtu, která musí být provedena sekvenčně

f_p ... část výpočtu, kterou lze paralelizovat

p ... počet procesorů

Předpokládá se:

- při výpočtu sekvenční části se ostatní CPU nevyužívají,
- velikost řešeného problému se při přidávání procesorů nemění.

Gustafsonův zákon

Udává maximální zrychlení pro paralelní systémy:

$$S \leq \frac{T(n)}{T(n, p)} = f_s + p \cdot (1 - f_s) \quad f_s = \frac{t_s}{t_s + \frac{t_p}{p}}$$

f_s ... část výpočtu, která musí být provedena sekvenčně

f_p ... část výpočtu, kterou lze paralelizovat

p ... počet procesorů

Předpokládá se:

- za konstantní se považuje doba běhu (ne rozsah úlohy)

Zdroje neefektivnosti

- nedostatek užitečné práce pro daný počet procesorů,
- velké komunikační náklady,
- příliš velká režie synchronizace procesorů,
- špatná distribuce práce (nerovnoměrné rozdělení práce) mezi procesory.

Řešení:

- *technologické* (rychlejší komunikační HW a komunikační režie, překrývání komunikačních a výpočetních operací);
- *algoritmické* (dobré mapování algoritmy na paralelní architekturu, volba vhodné zrnitosti úkolů, vhodné předřazování komunikačních operací před výpočty, rovnoměrné rozdělení výpočetní zátěže mezi CPU, ...).

9

Úrovně granularity procesů

- příkazy a instrukce (nejjemnější),
- cykly a iterace,
- podprogramy,
- části úloh a programů,
- nezávislé úlohy a programy (nejhrubší).

Vývoj postupoval od nejnižších úrovní k nejvyšším, např.
paralelismus na 1. úrovni je typický pro superskalární CPU,
paralelismus na 2. úrovni je typický pro vektorové CPU.

10

Klasifikace paralelních systémů

zavedl v roce 1966 J. M. Flynn (podle toku instrukcí a dat):

- SISD (Single Instruction stream, Single Data stream),
- MISD (Multiple Instruction stream, Single Data stream),
- SIMD (Single Instruction stream, Multiple Data stream),
- MIMD (Multiple Instruction stream, Multiple Data stream).

příliš hrubé dělení, někdy se používají varianty:

- **MSIMD** (Multiple SIMD) – systém, v němž pracuje několik podsystémů SIMD nezávisle na sobě;
- **SPMD** (Single Program, Multiple Data stream) – modifikace SIMD, kde všechny procesory sice provádějí stejný program (dělí se o instrukce), ale nezávisle na sobě (bez synchronizace).

11

Klasifikace paralelních systémů

Podle způsobu řízení systému:

- *řízené tokem instrukcí* (dané pořadím instrukcí)
 - SIMD – s lokální nebo globální pamětí,
 - MIMD – volně nebo těsně vázané;
- *řízené tokem událostí* (k provedené operaci dochází v okamžiku, kdy to okolnosti umožňují nebo vyžadují)
 - řízení tokem dat,
 - řízení tokem požadavků;
- *bez centrálního řízení* (např. jednoúčelové systolické systémy, které jsou obvodově přizpůsobeny výpočetnímu algoritmu nebo neuronové sítě), MISD (několik procesů zpracovává jedny data).

12

Těsně vázané systémy

Těsně vázané systémy – paralelní systémy, v nichž procesory nejsou vybaveny lokální pamětí nebo je tato paměť velmi malá – pokud jsou všechny CPU stejného typu a mají rovnocenný přístup k paměti, mluvíme o *symetrickém multiprocesoru* (Symmetric Multi-Processor, SMP).

Propojovací síť lze propojit libovolný CPU s libovolným paměťovým modulem.

Velké nároky na rozsah a rychlost komunikace mezi CPU.

Sdílená paměť

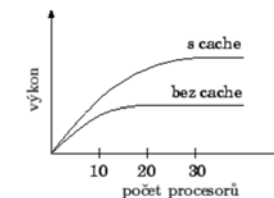


13

Těsně vázané systémy

Při použití většího množství procesorů již nedochází k nárůstu výkonu
⇒ použití cache paměti u procesoru.

Vzniká ale problém konzistence dat uložených v několika cache současně – řeší se např. metodou přímé signalizace změn (zrušení platnosti kopií ve všech cache) nebo rozdělením dat na taková, která lze a která nelze přesouvat do cache (přesouvají se jen data, která se nemění).



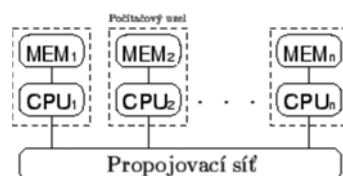
14

Volně vázané systémy

Volně vázané systémy – každý procesor v systémech s distribuovanou pamětí je vybaven velkou lokální pamětí (a často i svými periferními zařízeními)

- každý CPU má značnou autonomii,
- někdy se tyto systémy označují jako *multipočítače*,
- velká rychlost komunikace s lokální pamětí,
- počet CPU není v principu omezen,
- není problém s konzistencí dat v cache,
- komunikace formou zpráv,
- slabá interakce mezi CPU.

(masivně paralelní počítače
clusters)



15

Propojovací síť

Slouží ke komunikaci mezi jednotkami; dělí se na:

- *statické síť* – neměnné, vhodné pro méně složité sítě;
- *dynamické síť* – obsahují spínací prvky - řídí se buď globálně (centralizované řízení, společný řadič, propojování okruhů, vhodné pro občasné velké objemy dat) nebo lokálně (distribuované řízení, propojování paketů, vhodné pro velký počet malých objemů)
 - křížové přepínače,
 - síť typu promíchání s výměnou,
 - válcové posouvače,
 - sběrnice.

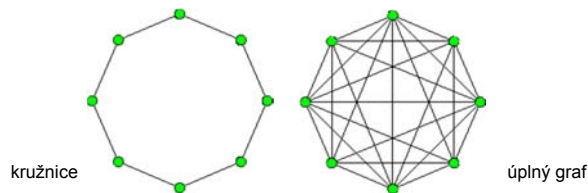
16

Statické propojovací sítě

Strukturu obvykle vyjadřujeme grafem (uzel odpovídá modulu, hrana odpovídá komunikačnímu spoji).

Důležité vlastnosti:

- průměr grafu (největší vzdálenost dvou uzlů v grafu) – má vliv na celkovou rychlost systému;
- stupeň uzlu (počet hran příslušejících uzlu) – určuje počet komunikačních adaptérů procesoru (souvisí s cenou).



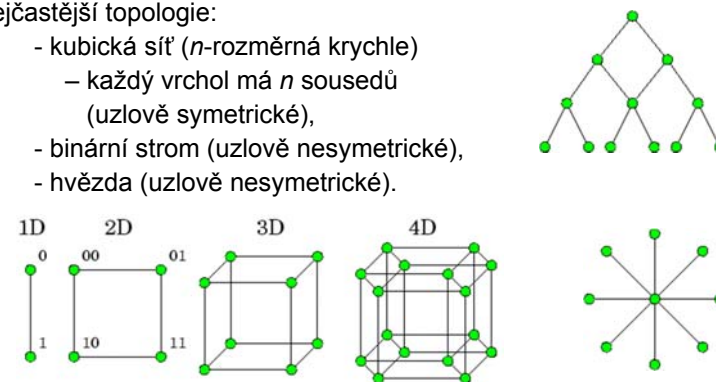
17

Statické propojovací sítě

Scalability (rozšiřitelnost) – vyjadřuje, jak se změní složitost komunikace, přidáme-li další uzly.

Nejčastější topologie:

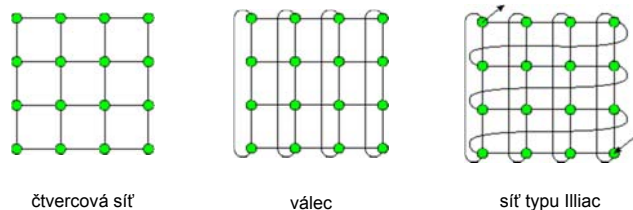
- kubická síť (n -rozměrná krychle) – každý vrchol má n sousedů (uzlově symetrické),
- binární strom (uzlově nesymetrické),
- hvězda (uzlově nesymetrické).



18

Statické propojovací sítě

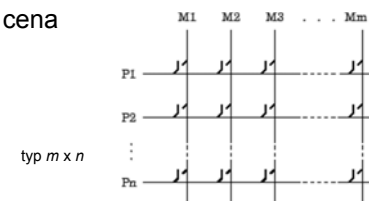
- čtvercová síť
- trojúhelníková síť
- válec
- anuloid, torus, síť typu Illiac



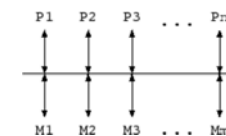
19

Dynamické propojovací sítě

Křížový přepínač – informace prochází pouze jedním přepínačem, počet dvojic vstup-výstup není omezen, nevýhodou je vysoká cena (velký počet spínačů)



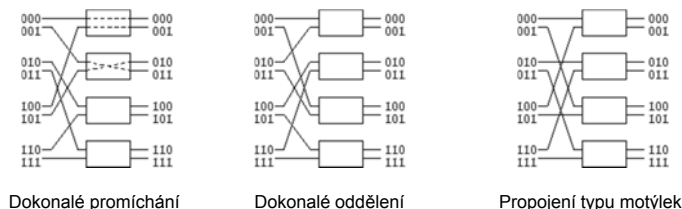
Sběrnice – může komunikovat každý s každým, v jednom okamžiku pouze jeden přenos



20

Dynamické propojovací sítě

Síť typu promíchání výměnou – je tvořena různě propojenými elementárními prepínači, nejčastěji typu 2 x 2 (mají dva pracovní stavy – identita a výměna); spojuje se několik prepínačů za sebou.



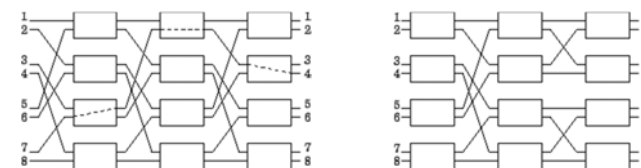
21

Dynamické propojovací sítě

Blokující síť – požadované spojení může být blokováno existujícími spojeními.

Přestavitelné síť – lze vždy realizovat všechna propojení, avšak někdy pouze za cenu přestavění stávajících spojení.

Neblokující síť – umožňují vytvořit spojení libovolného vstupu s libovolným výstupem bez ohledu na to, jaká jiná spojení již jsou uskutečněna.



22

Dynamické propojovací sítě

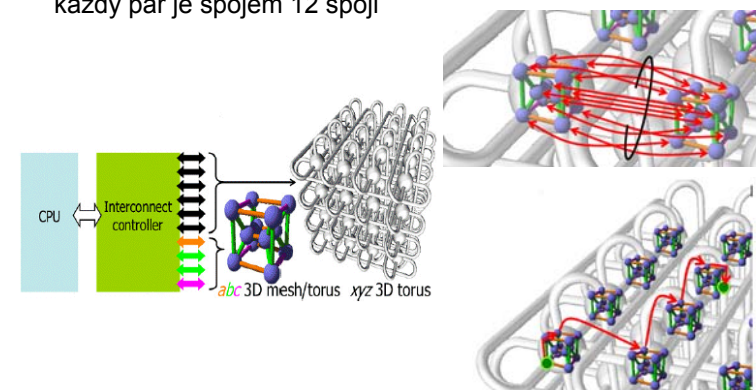
Přepínače lze řídit globálně nebo lokálně ⇒

- *centralizované řízení sítě* (společný řadič)
vede na *propojování okruhů* – vytvoří se přenosová cesta po celou dobu spojení (vhodné pro velké objemy dat);
- *distribuované řízení* (každý prepínač má svůj rozhodovací prvek) – vede na *propojování paketů* (vhodné pro přenos velkého počtu malých zpráv).

23

TOFU propojovací síť

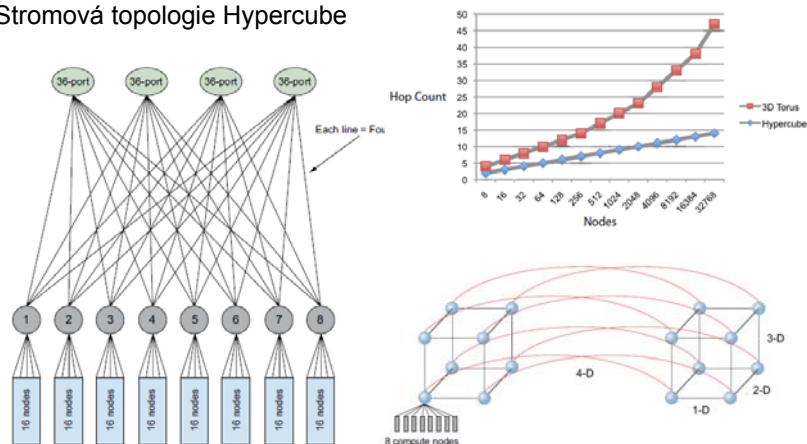
TOFU (torus/fusion) – 6D mesh/torus topologie, každý uzel obsahuje 6+4 spoju (každý o rychlosti 10GB/s), každý pár je spojen 12 spoji



24

Propojovací síť Hypercube

Stromová topologie Hypercube



25

Superskalární procesory

Paralelní zpracovávání procesů; jeho plánování rozdělujeme na:

- *statické* – paralelní vydávání instrukcí naplánovány kompilátorem (např. architektura VLIW) – složitější kompilátory;
- *dynamické* – o paralelismu rozhodují technické prostředky za běhu programu (složitější), obsahuje plánovací jednotku (Schedule Unit), umožněna binární kompatibilita mezi generacemi procesorů.

Programátor přesně neví, které instrukce procesor seskupí pro paralelní zpracování => problematický odhad celkového počtu instrukčních cyklů (problematická optimalizace).

26

Vektorové procesory

Účelem je zrychlit práci s vektory (a tím také s maticemi).

Celá programová smyčka je nahrazena jedinou vektorovou instrukcí a provedena zřetěženě se všemi prvky vektorů.

Výkonnost roste s hloubkou zřetěžení, současně se zvětšuje ale i velikost procesoru.

Odhaduje se, že cca 70% všech algoritmů má vektorový charakter, příp. se na něj dá převést (vektORIZUJÍCÍ kompilátor)

Vektorové procesory obsahují v architektuře i skalární procesory pro nevektorové operace.

Paměťová nebo registrová architektura (záleží, kde jsou umístěna data, se kterými jsou prováděny vektorové operace).

27

Maticové procesory

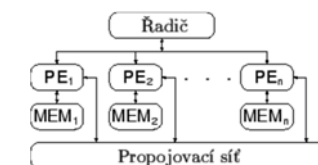
Pro operace s maticemi je charakteristické, že větší počet prvků matice se zpracovává současně a navíc se se všemi provádí stejná operace => vhodné pro SIMD architekturu.

Každá jednotka má svou lokální paměť, komunikace mezi PE zajištěna centrální propojovací sítí.

Maticové procesory bývají součástí univerzálních počítačů.

Někdy se maticové procesory označují jako paralelní vektorové procesory

PE - Processor Element



28

Systolická pole

Multiprocessorový systém MISD (několik procesorů zpracovává jedny data), procesory komunikují vždy pouze s nejbližším sousedem v propojené síti, každý systolický prvek (SP) využívá pouze svou lokální paměť a vykonává stále stejnou úlohu (netřeba řídit);

- SP může obsahovat buď kompletní CPU nebo dekodér instrukcí + procesní jednotku nebo jednoúčelovou pevně zadrátovanou buňku.

Vlastnosti:

- strukturou proudí vstupní data i mezivýsledky (různými směry i rychlostmi) – na rozdíl od zřetězeného zpracování;
- buňky jsou jednoho nebo několika málo typů;
- kombinace intenzivní lokální komunikace a výpočtů;
- nejčastěji mají dimenzi 1 nebo 2.

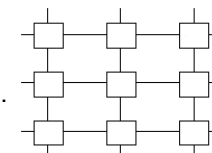
29

Systolická pole

- *Jednoúčelová* – zadrátované systolické prvky ušité na míru specifické aplikaci (může být i na jednom čipu);
- *Univerzální* – mohou se adaptovat na řadu aplikací rekonfigurací;
- *Programovatelná* – každá procesní jednotka je programovat.

Využití: manipulace s datovými strukturami nebo řetězci znaků, matematické operace, maticová aritmetika, zpracování obrazové informace, rozpoznávání řeči nebo obrazu.

Systolickou síť není nutné řídit programem.

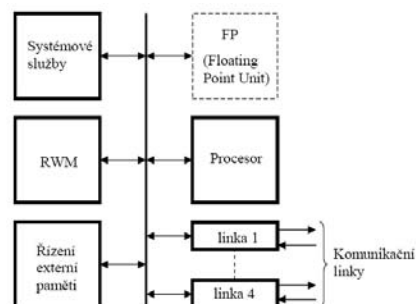


30

Transputer

Jedná se v podstatě o jednočipový mikro počítač, který je základním prvkem výstavby paralelních výpočetních systémů. Skládá se z procesoru, paměti RAM (pro cache, instrukce a data), komunikačních linek, časovače, příp. FPU, obvody přerušení, DMA

- vznikl v roce 1983 (TRANSistor-comPUTER),
- 16 až 64 bitové,
- hodinový takt 20-50 MHz,
- RISC.

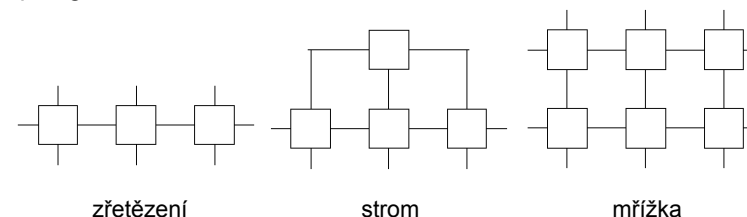


31

Propojování transputerů

Spojování pomocí několika (obvykle čtyř nebo osmi) rychlých sériových kanálů (20-100Mbps), vzájemnou komunikaci podporuje i instrukční sada (zahájí se přenos a ten pak probíhá bez účasti procesoru).

Topologie:



32

Počítačové clustery

Seskupení volně vázaných počítačů, které spolu úzce spolupracují (navenek může pracovat jako jeden počítač)

- obvykle propojeny rychlou datovou sítí (např. Myrinet, cLAN, Gigabitový Ethernet, ATM, InfiniBand) pro zasílání zpráv mezi procesy, izolované od vnější síťové infrastruktury;
- volnější sdružení počítačů se nazývá *farma*.

Většinou jeden uzel stojí nad ostatními (master node) a provádí např. rozdělování úkolů.

Typický cluster využívá „open source“ OS (např. Linux).

Cluster se z hlediska uživatele (aplikačního programátora) jeví jako jeden velmi výkonný a homogenní počítač.

33

Vlastnosti clusterů

- většinou sestaveny ze „standardních“ počítačů (x86);
- příznivý poměr ceny a výkonu;
- snadná rozšiřovatelnost;
- většinou homogenní uzly, mohou být i heterogenní;
- odolnost vůči výpadku, spolehlivost;
- některé uzly mohou být pasivní (záložní);
- nižší využitelnost výkonu (nižší využití procesorů);
- náročnější správa;
- úzká místa ve formě sdílených komponent (např. komunikační přepínač);
- vhodné pro paralelizovatelné slabě vázané procesy, které nepotřebují častou synchronizaci či výměnu výsledků.

34

Typy clusterů (podle funkce)

Výpočetní cluster (high performance computing - HPC)

- pro zvýšení výpočetního výkonu.

Cluster s vysokou dostupností (high availability, failover)

- zajištění nepřetržité poskytování nějaké služby (při výpadku uzlu přebírá práci jiný uzel).

Cluster s rozložením zátěže (load balancing, scalable)

- službu poskytuje několik počítačů, požadavky jsou zasílány na uzly podle jejich aktuálního vytížení.

Úložný cluster (storage) – zprostředkovává přístup k diskové kapacitě (většinou rozložena mezi více počítačů).

Gridové clustery – rozlehlé clustery zabezpečené tak, aby komunikace mohla probíhat v nechráněném prostoru internetu (základem jsou certifikáty uživatelů a počítačů).

Funkce clusterů se ve skutečnosti prolínají.

35

Clustery pracovních stanic

Clustery většinou chápeme lokálně, ale propojovací síť může být např. i internet (COW – Clusters of Workstations, NOW – Network of Workstations).

Na stanicích je spuštěn *démon PVM* (Parallel Virtual Machine), na jednom z počítačů je spuštěn *hlavní proces*, který posílá démonům úkoly

- COW umožňuje heterogenost stanic (HW i SW),
- z hlediska programátora se COW jeví jako paralelní počítač s distribuovanou pamětí,
- minimální pořizovací náklady,
- nevýhodou je zejména nízká komunikační kapacita a její velká režie (časová náročnost),
- obtížné ladění paralelních aplikací,
- malá využitelnost instalovaného výkonu.

36

Nejvýkonnější cluster na světě

Tianhe-2 – (Top500 – první již od června 2013),
3,12 mil. jader, CPU Intel Xeon E5 2,2 GHz, 33 862 TFlops,
1 024 TB RAM, OS Kylin Linux, 17,8 MW; síť TH Express-2;
National Super Computer Center v Guangzhou, Čína



37

Nejvýkonnější clustery v ČR

Národní superpočítačové centrum IT4Innovations, VŠB-TU Ostrava,
www.it4i.cz

Anselm – jaro 2013, 1. etapa
207 serverů – každý dva 8jádrové CPU Intel Sandy Bridge,
RAM 4 GB/jádro, tj. 13 248 GB, HDD 300 GB/server (62,1 TB),
výpočetní výkon 66 TFlops, OS Linux.
Výpočetní vysokorychlostní síť InfiniBand QDR,



38

Nejvýkonnější cluster v ČR

Salomon – uveden do provozu v červenci 2015 (2. etapa)
1008 výpočetních uzlů s 24192 výpočetními jádry Intel Xeon
(Haswell-EP) a 129 TB operační paměti; 864 koprocesorových
karet (Intel Xeon Phi 7120) s 52704 jádry a 13,8 TB paměti RAM;
teoretický výpočetní výkon 1457 TFlop; disková kapacita
2 PB a 3 PB páskové kapacity pro zálohování;
výpočetní síť InfiniBand FDR (rychlost 56 GB/s);
připojení do Internetu 40 GB/s;
operační systém CentOS;

40. nejvýkonnější superpočítač
na světě (žebříček TOP500).



39

Metacentrum

Virtuální organizace sdružující výpočetní a úložné kapacity vědecko-
výzkumných institucí ČR.

Aktuálně připojeno přes 11 tis. CPU jader v 10 institucích
(více na <http://metavo.metacentrum.cz>).

Uživatelé se může stát každý zaměstnanec nebo student
akademické instituce ČR (členství je zdarma).



40

Nejvýkonnější cluster na FM TUL

Hydra – celkem 70 CPU, 116 GB RAM, 2.2TB HDD
13 uzlů *Sun Fire* (2 x AMD Opteron 252, 2600 MHz,
4 GB RAM, 73 GB HDD);
11 uzlů *Dell PowerEdge* (2 x Intel Xeon 5140 dual-core
2330 MHz, 4 GB RAM, 80 GB HDD);
propojení přes Ethernet 1 Gbps,
operační systém Linux CentOS.

Cloud computing

- sdílení hardwarových i softwarových prostředků pomocí sítě (poskytování služeb a programů uložených na serverech na internetu – kancelářské aplikace, operační systémy provozované ve webových prohlížečích, výpočetní systémy, hosting apod.);
- veřejný, soukromý, hybridní, komunitní.

Výhody a nevýhody:

- snížení nákladů na IT operace (HW, SW),
- vysoká škálovatelnost a spolehlivost,
- nebezpečí ztráty soukromí, otázka bezpečnosti dat,
- závislost na poskytovateli.

Instrukční soubor mikrořadičů PIC16

Mnemonika, operandy		Cyklů	Popis	14 bitová instrukce		Nastavuje příznaky
				msb	lsb	
ADDWF	f, d	1	Součet W a f	00 0111 dfff ffff		C, DC, Z
ANDWF	f, d	1	AND W a f	00 0101 dfff ffff		Z
CLRF	f	1	Nuluje f	00 0001 1fff ffff		Z
CLRW		1	Nuluje W	00 0001 dfff ffff		Z
COMF	f, d	1	Doplňěk f	00 0111 0xxx xxxx		Z
DECF	f, d	1	Dekrement f	00 0011 dfff ffff		Z
DECFSZ	f, d	1 (2)	Dekrement f, přeskok při 0	00 1011 dfff ffff		
INCF	f, d	1	Inkrement f	00 1010 dfff ffff		Z
INCFSZ	f, d	1 (2)	Inkrement f, přeskok při 0	00 1111 dfff ffff		
IORWF	f, d	1	OR W a f	00 0100 dfff ffff		Z
MOVF	f, d	1	Přesun f	00 1000 dfff ffff		Z
MOVWF	f	1	Přesun W do f	00 0000 1fff ffff		
NOP		1	Prázdná operace	00 0000 0xx0 0000		
RLF	f, d	1	Rotace vlevo přes C	00 1101 dfff ffff		C
RRF	f, d	1	Rotace vpravo přes C	00 1100 dfff ffff		C
SUBWF	f, d	1	Odečte W od f	00 0100 dfff ffff		C, DC, Z
SWAPF	f, d	1	Zamění nibly f	00 1110 dfff ffff		
XORWF	f, d	1	XOR W a f	00 0110 dfff ffff		Z
Bitové operace						
BCF	f, b	1	Nuluje bit f	01 00bb bfff ffff		
BSF	f, b	1	Nastavit bit f	01 01bb bfff ffff		
BTFSC	f, b	1 (2)	Test bitu f, přeskok je-li 0	01 10bb bfff ffff		
BTFSS	f, b	1 (2)	Test bitu f, přeskok je-li 1	01 11bb bfff ffff		
Operace s literálem a řídicí instrukce						
ADDLW	k	1	Součet literálu a W	11 111x kkkk kkkk		C, DC, Z
ANDLW	k	1	AND literálu a W	11 1001 kkkk kkkk		Z
CALL	k	2	Volání podprogramu	10 0kkk kkkk kkkk		
CLRWDT		1	Nuluje watchdog	00 0000 0110 0100		TO, PD
GOTO	k	2	Nepodmíněný skok	10 1kkk kkkk kkkk		
IORLW	k	1	OR literálu a W	11 1000 kkkk kkkk		Z
MOVLW	k	1	Přesun literálu do W	11 00xx kkkk kkkk		
RETFIE		2	Návrat z přerušení, nastaví bit GIE	00 0000 0000 1001		
RETLW	k	2	Návrat z podprogramu s literálem ve W	11 01xx kkkk kkkk		
RETURN		2	Návrat z podprogramu	00 0000 0000 1000		
SLEEP		1	Přecod od módu se sníženým příkonem	00 0000 0110 0011		TO, PD
SUBLW	k	1	Rozdíl W a literálu	11 110x kkkk kkkk		C, DC, Z
XORLW	k	1	XOR W a literálu	11 1010 kkkk kkkk		Z
Pouze pro slučitelnost s PIC 16C5X. Použití v nových programech není doporučeno.						
OPTION		1	Uloží W do OPTION registru	00 0000 0110 0010		
TRIS	p	1	Uloží W do TRIS registru	00 0000 0110 pppp		