

2010.02.09 vI

1. Uveďte charakteristické vlastnosti procesorů typu 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),
- 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 naplánování instrukcí,
- nejčastěji harvardská architektura.

2. Předpokládejme vylepšení procesoru pro databázové výpočty. Nový procesor je 5krát rychlejší než nynější. Dále víme, že nyní je procesor zaměstnán z 65 % výpočty a 35 % času čeká na vstupně-výstupní operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

$$t_0 = 0,65 + 0,35 = 1$$

$$t_1 = \frac{0,65}{5} + 0,35 = 0,13 + 0,30 = 0,43$$

$$\text{zrychlení} = \frac{t_0}{t_1} = \frac{1}{0,4625} \cong 2,326$$

3. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	BB	ADDWF 14,1
04 (FSR)	AA	BTFSC 15,6
14	DD	DECF 4,1
15	CC	MOVLW 15

W = BB	04 (FSR) = AA	14 = DD	15 = CC
ADDWF 14,1			
14=BB+1=BC			
W = BC	04 (FSR) = AA	14 = BC	15 = CC
BTFSC 15,6			
přeskok se neprovede → následující instrukce se provede			

W = BC	04 (FSR) = AA	14 = DD	15 = CC
DECF 4,1			
04 (FSR)=AA-01=A9			
W = BC	04 (FSR) = A9	14 = DD	15 = CC
MOVLW 15			
W = 15			
W = 15	04 (FSR) = A9	14 = DD	15 = CC

4. Jak bude reprezentováno číslo $(-18)_D$ v sedmimístné celočíselné číslicové formě:

- v doplňkovém kódu
 - v aditivním sudém kódu
- a) $(-18)_D = inv(18_B) + 1 = inv(0010010) + 1 = 1101101 + 1 = 1101110$
b) stačí obrátit první znak $\rightarrow 0101110$

5. Co je to DMA? Naznačte princip činnosti.

Direct Memory Access (přímý přístup do paměti) je metoda kopírování bloků dat mezi pamětí a portem (HDD, pamětí,...) bez průchodu skrz procesor, bez dočasného 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) \Rightarrow řešíme využitím DMA řadičů.

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.

Řadič potřebuje ke své činnosti 3 údaje:

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

6. Jaké znáte hlavní módy adresování? Stručně naznačte principy.

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]])

7. Napište výkonnostní rovnici procesoru bez cache a s cache, popište jednotlivé veličiny.

Výkonnostní rovnice CPU bez cache: $T_{CPU} = IC \cdot CPI \cdot T_{clk}$

T_{CPU} ... doba provádění programu

IC ... počet instrukcí (Instruction Count)

CPI ... průměrný počet taktů na instrukci (Cycles Per Instruction)

T_{clk} ... doba cyklu (taktu)

Výkonnostní rovnice CPU s cache: $T_{CPU} = IC \cdot (CPI + MAPI \cdot MR \cdot MP) \cdot T_{clk}$

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

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

MP ... čas získání dat z nižší úrovně paměťové hierarchie (Miss Penalty)

8. Co jsou to clustery? Jaké znáte typy?

Počítačový cluster je seskupení počítačů, které spolu úzce spolupracují. Cluster se z pohledu uživatele jeví jako jeden velmi výkonný a homogenní počítač. Obvykle jsou PC v clusteru propojeny rychlou datovou sítí (např. Myrinet, cLAN, Gigabitový Ethernet, ATM) pro zasílání zpráv mezi procesy, izolované od vnější síťové infrastruktury. Většinou jeden uzel stojí nad ostatními (master node) a provádí např. rozdělování úkolů. Typický cluster využívá „open source“ OS (např. Linux). Volnější sdružení počítačů se nazývá farma.

Typy clusterů

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

9. Co víte o sběrnici PCI Express (princip, použití, příp. parametry)?

PCI Express je sériová sběrnice složená ze dvou nízkonapěťových diferenciálních párů (vysílacího a přijímacího) – lane LVDS (Low-Voltage Differential Signaling)

- architektura point-to-point (žádné sdílení sběrnice)

- vícecestavová modulace (korekce ztrát)

PCI Express 3.0 - propustnost $16 \cdot 1 \text{ GB/s}$ (16 linek)

2010.01.25 vH

1. Uveďte charakteristické vlastnosti procesorů typu 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,
- s postupem doby se začíná používat zřetěžené zpracování.

2. Mikrořadič pracuje s frekvencí 4 MHz. K provedení jednoho instrukčního cyklu vyžaduje 4 hodinové takty. Program obsahuje 90 % instrukcí jednocyklových a 10 % instrukcí dvoucyklových. Jaký je výkon mikrořadiče v MIPS?

$$P_{MIPS} = \frac{f_{clk}}{CPI} \cdot 10^{-6} = \frac{4 \cdot 10^6}{0,9 \cdot 4 + 0,1 \cdot 2 \cdot 4} \cdot 10^{-6} = \frac{4}{3,6 + 0,8} = \frac{4}{4,4} \cong 0,9 MIPS$$

3. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	F2	MOVF INDF,w
04 (FSR)	0E	ADDWF 14,f
14	AA	DECFSZ 15,f
15	01	MOVLW 04

W = F2	04 (FSR) = 0E	14 = AA	15 = 01
movf INDF,w = movfw INDF (read the contents of the register pointed to by FSR into W) $0E_h = 14_d$			
W = AA			
W = AA	04 (FSR) = 0E	14 = AA	15 = 01
ADDWF 14,f			
14=F2+AA 11110010 +10101010 ----- 10011100			
W = AA	04 (FSR) = 0E	14 = 9C	15 = 01
DECFSZ 15,f			
15=01-01=00 → následující instrukce se přeskočí			
W = AA	04 (FSR) = 0E	14 = 9C	15 = 00
MOVLW 04			
tahle instrukce se neprovede			
W = AA	04 (FSR) = 0E	14 = 9C	15 = 00

4. Číslo zapsané v doplňkovém kódu má tvar „1010101“. Jak bude vypadat zápis stejného čísla v aditivním lichém kódu (zapsaný pomocí stejného počtu bitů)? O jaké číslo v desítkové soustavě se jedná?

Stačí obrátit první znak a odečíst 1 → 0010100

Číslo začíná 1, takže je záporné. Spočítám jeho absolutní hodnotu.

$$1010101_B = -inv(1010101_B - 1_B) = -inv(1010100_B) = -0101011_B = -43_D$$

5. K čemu slouží aritmeticko-logická jednotka? Z jakých částí se skládá (se může skládat)?

Aritmeticko-logická jednotka ALU (Arithmetic Logic Unit) je součástí 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 sekvenční)

- 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.

6. Jaká je dosažitelná účinnost zřetěžené 5stupňové linky při zpracování 5 instrukcí. Spočtěte dále průměrnou hodnotu CPI.

Čas	1	2	3	4	5	6	7	8	9	10
S1	11	12	13	14	x	15				
S2		11	12	13	x	14	15			
S3			11	12	x	13	14	15		
S4				11	12	-	13	14	15	
S5					11	12	-	13	14	15

(„x“ značí čekání na mezivýsledek; „-“ značí nevyužitelnost)

$$CPI = \frac{10}{5} = 2 \text{ (taktů na 1 instrukci)}$$

$$\text{zrychlení} = \frac{\text{čas bez paralelizace}}{\text{nový čas}} = \frac{S_K}{10} = \frac{5 \cdot 5}{10} = 2,5$$

$$\text{účinnost pipeliningu } E_K = \frac{S_K}{\text{počet stupňů}} = \frac{2,5}{5} = 0,5$$

7. Jaký je principiální rozdíl mezi volně a těsně vázanými systémy ve víceprocesorových systémech? Uveďte výhody a nevýhody obou systémů.

Těsně vázané systémy jsou 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.

Vzniká 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).

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 clustery)

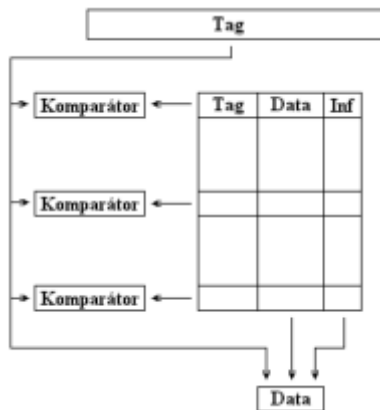
8. V čem se liší plně asociativní cache od n-cestně asociativní cache?

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:

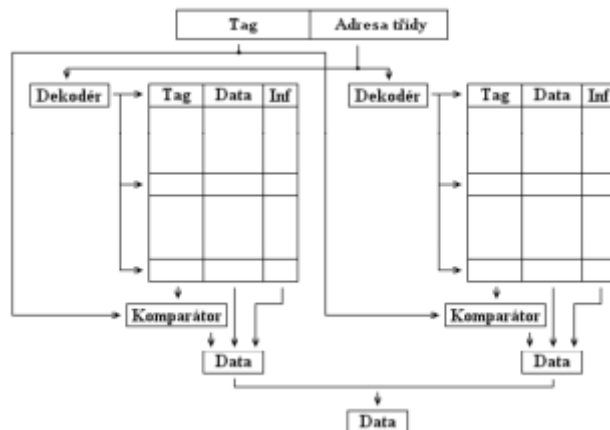
- mnoho komparátorů
- nutná velká kapacita paměti pro tagy



n-cestně asociativní cache

Zadaná adresa se rozdělí na 2 části: tag + adresa třídy. Adresa třídy pomocí dekodérů vybere jeden řádek v každé tabulce, jehož tagy se porovnávají (nejpoužívanější).

- pro velká „n“ se blíží plně asociativní variantě



9. Co víte o sběrnici HyperTransport (princip, topologie). Kde se tato sběrnice používá?

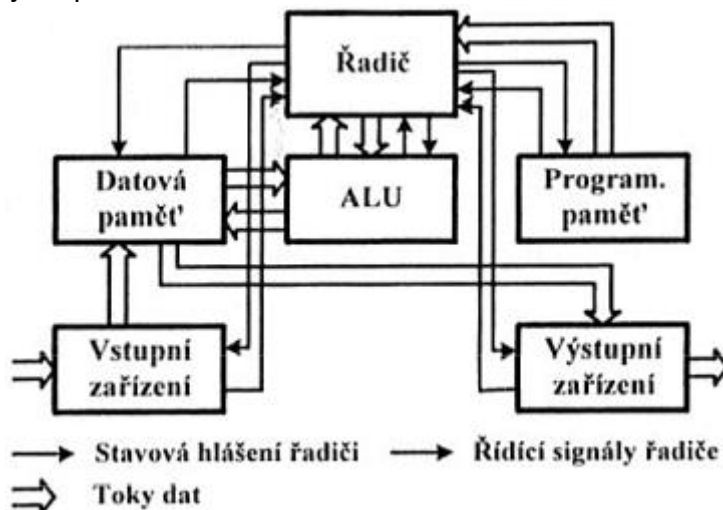
Vylepšená sběrnice FSB – duální jednosměrná sériová sběrnice (pro každý směr komunikace jedna sběrnice) založená na principu paketů (point-to-point)

- 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 Bridge).

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 32bitová (pro každý směr), frekvence 3,2 GHz, \Rightarrow propustnost 51,2 GB/s

2010.01.19 vG

1. Uvedte charakteristické vlastnosti Harwardské architektury počítačů.
vznikla v roce 1943 (koncepte IBM Harvard MARK1)
někdy označována jako princetonská 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

2. Vypočtete průměrnou dobu přístupu do paměti (systém složený z cache a operační paměti), je-li vybavovací doba cache 12 ns, čas získání dat z operační paměti 160 ns a pravděpodobnost neúspěchu je 10%.

$$t = 12\text{ns} + 160\text{ns} \cdot 0,1 = 28\text{ns}$$

3. Uvedte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	F5	MOVLW 14
04 (FSR)	10	MOVWF 4
14	BB	DECF 14,1
15	0F	BSF 14,1

W = F5	04 (FSR) = 10	14 = BB	15 = 0F
MOVLW 14			
W = 14			
W = 14	04 (FSR) = 10	14 = BB	15 = 0F
MOVWF 4			

04 (FSR) = 14			
W = 14	04 (FSR) = 14	14 = BB	15 = OF
DECF 14,1			
14=BB-01=BA			
W = 14	04 (FSR) = 14	14 = BA	15 = OF
BSF 14,1			
nastavit bit f 00001010→00001010			
W = 14	04 (FSR) = 14	14 = BA	15 = OF

4. Číslo zapsané v aditivním sudém kódu má tvar „0110010“. Jak bude vypadat zápis stejného čísla v přímém kódu se znaménkem (zapsaný pomocí stejného počtu bitů)? Stačí obrátit první znak a dostaneme číslo v doplňkovém kódu → 1110010 → Číslo je záporné Spočítáme absolutní hodnotu

$$abs(1110010) = inv(1110010 - 1) = inv(1110001) = 0001110$$

Protože je číslo záporné, první znak bude 1 → 1001110

5. Popište výhodu technologie spekulativního provádění instrukcí (speculative execution). Kde se používá?

Spekulativní provádění instrukcí se využívá v Post-RISC procesorech. (Do této kategorie je možné zařadit většinu současných CPU.)

Spekulativní provádění instrukcí 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.

6. Jaká je dosažitelná účinnost zřetězené 4stupňové linky při zpracování 5 instrukcí. Spočtěte dále průměrnou hodnotu CPI.

Čas	1	2	3	4	5	6	7	8	9
S1	11	12	13	14	x	15			
S2		11	12	13	x	14	15		
S3			11	12	x	13	14	15	
S4				11	12	-	13	14	15

(„x“ značí čekání na mezivýsledek; „-“ značí nevyužitelnost)

$$CPI = \frac{9}{5} = 1,8 \text{ (taktů na 1 instrukci)}$$

$$\text{zrychlení} = \frac{\text{čas bez paralelizace}}{\text{nový čas}} = \frac{4 \cdot 5}{9} \cong 2,2$$

$$\text{účinnost pipeliningu } E_K = \frac{S_K}{\text{počet stupňů}} \cong \frac{2,2}{4} \cong 0,55$$

7. Na jakých principech je založena funkce řadiče? Uveďte výhody a nevýhody jednotlivých koncepcí.

Řadič je část procesoru, která řídí vykonávání operace a chod celého procesoru podle instrukcí programu. **Obsahuje registr instrukcí**, který uchovává operační znak instrukce po

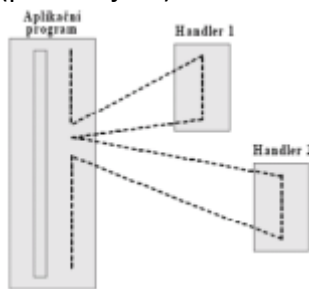
dobu jejího vykonávání, a **dekodér instrukcí**, který dekoduje a generuje řídicí signály pro procesor.

Existují 2 koncepce řadičů:

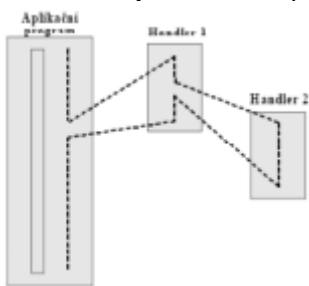
- 1) řadič je speciální sekvenční automat, který má čítač a dekodér (je **dražší**, ale **rychlejší**) – **obvodový řadič** (založen na kombinačních logických obvodech)
- 2) dekódování operačního znaku vykonává řídicí paměť, ve které jsou mikroprogramy uloženy – **mikroprogramový řadič** (založen na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

8. Jaké jsou principiální možnosti řešení priorit při více zdrojích přerušení?

Sekvenční zpracování - během obsluhy jednoho přerušení se další požadavky nepřijímají (pozdržují se). Sekvenční zpracování je jednoduché, ale ne vždy vhodné.



Vnořené zpracování - přijímají se přerušení s prioritou vyšší než je obsluhovaná priorita.



9. Na jakém principu pracuje sběrnice I²C?

Sběrnice 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

- 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 přenese 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

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

2008.01.13 vF

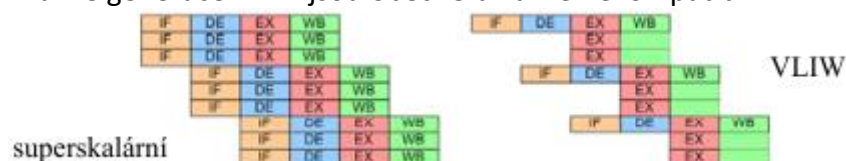
1. Uveďte charakteristické vlastnosti architektury VLIW počítačů; kde se v současnosti nejčastěji používá?

VLIW (Very Long Instruction Word) – architektura se čtením s více přístupy – delší instrukce sestavené z dílčích částí, které mohou být provedeny paralelně (instrukční packet) - např. 8 dílčích 32bitových 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.



Architektura VLIW počítačů se používá u paralelního zpracovávání procesů (superskalární procesory, statické plánování).

2. Předpokládejme vylepšení procesoru pro databázové výpočty. Nový procesor je 4krát rychlejší než nynější. Dále víme, že nyní je procesor zaměstnán ze 70 % výpočty a 30 % času čeká na vstupně-výstupní operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

$$t_0 = 0,70 + 0,30 = 1$$

$$t_1 = \frac{0,70}{4} + 0,30 = 0,175 + 0,30 = 0,475$$

$$\text{zrychlení} = \frac{t_1}{t_0} = \frac{1}{0,475} \cong 2,105$$

3. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	BB	ADDWF 15,1
04 (FSR)	DD	BTFSC 14,6
14	AA	INCF 4,1
15	CC	MOVLW 15

4. Jak bude reprezentováno číslo $(-11)_D$ v šestimístné celočíselné číslicové formě:

a) v doplňkovém kódu

b) v aditivním sudém kódu

a) $(-11)_D = \text{inv}(11_D) + 1 = \text{inv}(001011) + 1 = 110100 + 1 = 110101$

b) stačí změnit první znak doplňkového kódu $\rightarrow 010101$

5. Co jsou to clustery? Jaké znáte typy?

Počítačový cluster je seskupení počítačů, které spolu úzce spolupracují. Cluster se z pohledu uživatele jeví jako jeden velmi výkonný a homogenní počítač. Obvykle jsou PC v clusteru propojeny rychlou datovou sítí (např. Myrinet, cLAN, Gigabitový Ethernet, ATM) pro zasílání

zpráv mezi procesy, izolované od vnější sítě infrastruktury. Většinou jeden uzel stojí nad ostatními (master node) a provádí např. rozdělování úkolů. Typický cluster využívá „open source“ OS (např. Linux). Volnější sdružení počítačů se nazývá farma.

Typy clusterů

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).

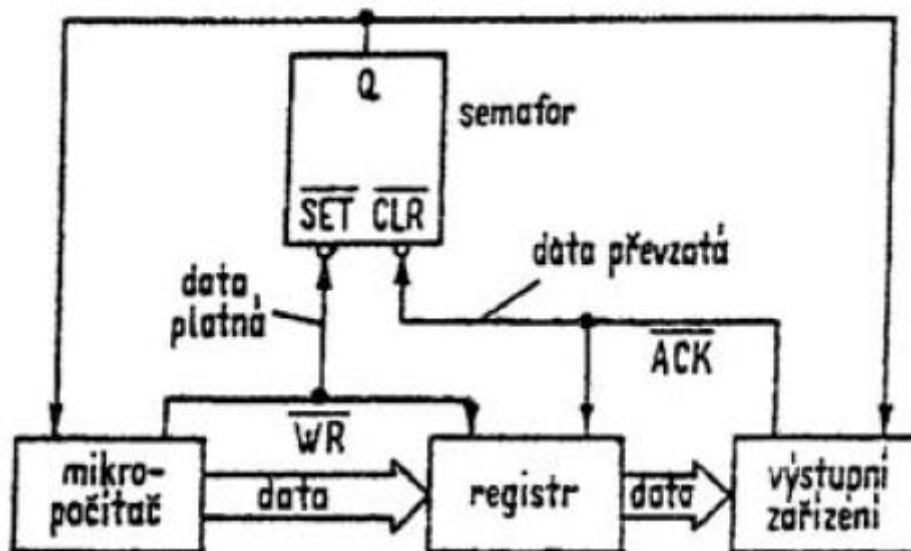
Cluster s rozložením zátěže (load ballancing, scalable)

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

6. Co představují bloky na obrázku? Stručně popište funkci.



Komunikace procesoru se vstupně-výstupními zařízeními - V/V periferie jsou obecně pomalé oproti CPU.

Na obrázku je **Úplný podmíněný výstup**

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.

7. Jaké znáte typy neadresovatelných pamětí? Jednotlivé typy stručně charakterizujte.

Neadresovatelné paměti (LIFO - zásobník, FIFO - fronta) – nezadává se u nich, kam zapisovat či odkud číst.

8. Co je architektura souboru instrukcí, co určuje? Jaké znáte typy?

Architektura souboru instrukcí (Instruction Set Architecture, ISA) představuje v podstatě rozhraní mezi SW a HW počítače.

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.

Typy ISA

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

9. Co víte o sběrnici PCI Express (princip, použití, příp. parametry)?

PCI Express je sériová sběrnice složená ze dvou nízkonapěťových diferenciálních párů (vysílacího a přijímacího) – lane LVDS (Low-Voltage Differential Signaling)

- architektura point-to-point (žádné sdílení sběrnice)
- více stavová modulace (korekce ztrát)

PCI Express 3.0 - propustnost $16 \cdot 1 \text{ GB/s}$ (16 linek)

2008.02.17 vE

1. Jaký je rozdíl mezi subskalárními, skalárními a superskalárními procesory?

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 zřetěžené zpracování, nebo replikace). V jednom taktu byla k vykonání v některé jednotce dána nevíš jedna instrukce (IPC = 1, Instruction Per Cycle). Vykonávání instrukcí probíhá překrytě (i paralelně). Dalším vývojem paralelismus navýšen i použitím několika zřetěžených funkčních jednotek.

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)

2. Výpočetní úloha je rozdělena na 3 části, z nichž každá trvá daný čas ($P_1 = 20\%$, $P_2 = 30\%$, $P_3 = 50\%$). Jaké je celkové zrychlení, jestliže část P_1 zrychlíme 5krát, část P_2 nezrychlíme a část P_3 zrychlíme 10krát?

$$t_0 = 0,20 + 0,30 + 0,50 = 1$$

$$t_1 = \frac{0,20}{5} + 0,30 + \frac{0,50}{10} = 0,04 + 0,30 + 0,05 = 0,39$$

$$\text{zrychlení} = \frac{t_0}{t_1} = \frac{1}{0,39} \cong 2,5641$$

3. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	F3	ANDLW F4h
04 (FSR)	14	MOVF 14,1
14	FF	INCF 14,1
15	00	BSF 14,7

4. Jak bude reprezentováno číslo $(-13)_D$ v šestimístné celočíselné číslicové formě:

a) v doplňkovém kódu

b) v aditivním sudém kódu

a) $(-13)_D = \text{inv}(13_D) + 1 = \text{inv}(001101) + 1 = 110010 + 1 = 110011$

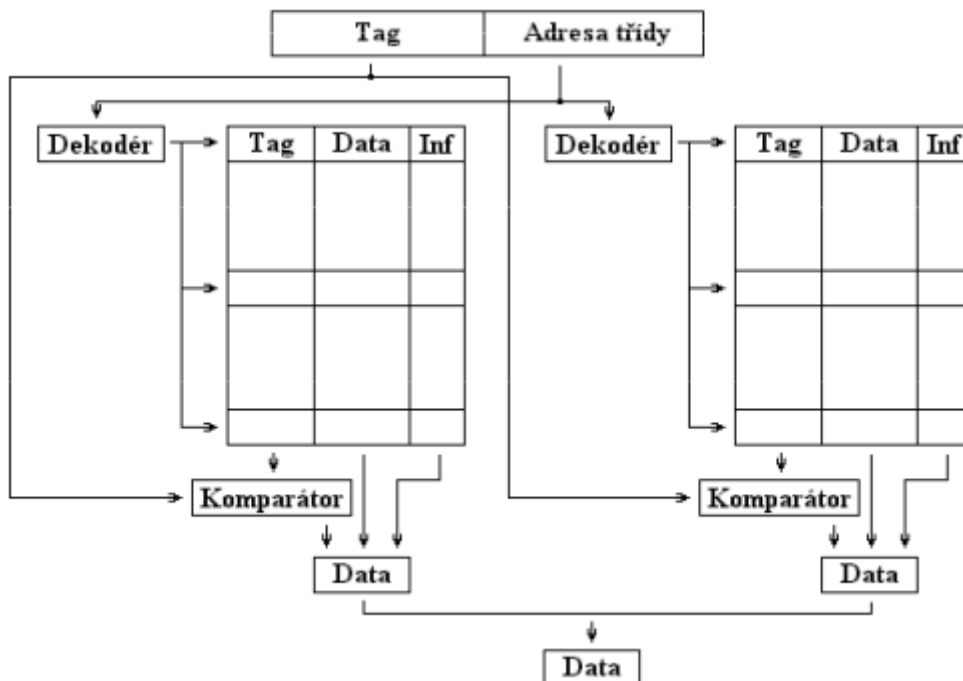
b) stačí změnit první znak doplňkového kódu $\rightarrow 010011$

5. Popište výhodu technologie zpracování instrukcí mimo pořadí (out-of-order). Kde se používá?

Out of Order je 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.

6. Jaké zařízení je na obrázku? Stručně popište jeho funkci.



Na obrázku je n-cestně asociativní cache.

Zadaná adresa se rozdělí na 2 části: tag + adresa třídy. Adresa třídy pomocí dekodérů vybere jeden řádek v každé tabulce, jehož tagy se porovnávají (nejpoužívanější). Pro velká „n“ se blíží plně asociativní variantě.

7. Čím se liší aritmetická jednotka běžných signálových procesorů od aritmeticko-logické jednotky běžných procesorů?

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

- HW řešení jako číslicový obvod (kombinační či sekvenční)

- součástí obvykle bývá příznakový (status) registr

Typický DSP má kromě aritmeticko-logické jednotky (ALU) navíc rychlou násobičku, která dokáže operaci násobení s přičítáním $A \leftarrow A + B \cdot k$. Tato operace je základní operací většiny algoritmů digitálního zpracování signálu.

8. Co je to cache a k čemu slouží? Jaké znáte typy?

Vyrovňovací paměť (cache, buffer) je obecně rychlá polovodičová paměť umístěná mezi rychlým zařízením (procesorem) a zařízením pomalejším (např. operační paměť, HDD), která **vyrovňuje rozdíly v rychlosti**. Zvyšuje výkon systému, obvykle se dělí na část pro data a pro instrukce. Pohyb v paměti cache řídí řadič cache, který je umístěn přímo v CPU (L1, L2) nebo je součástí čipové sady (L2, L3). Cache je založena na pamětech typu SRAM. Vyrovňovací paměť může být i softwarová (SMARTDRV, VCACHE). Zvláštní vyrovňovací paměť pro pomalé periferie (HDD, vypalovací mechaniky, síťové adaptéry) – lépe označovat jako buffer.

Nejčastěji rozdělujeme cache na:

- cache první úrovně L1 (first level, primární)

- co nejblíže 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.

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)

9. Co víte o sběrnici SPI? Na jakém principu pracuje?

SPI (Serial Peripheral Interface) byla navržena 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

Zařízení lze k SPI připojovat paralelně nebo sériově do řetězce (aktivní všechna zařízení).

Komunikace na sběrnici mezi zařízeními Master a Slave:

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

Konfigurační módy:

CPOL – určuje polaritu SCK

CPHA – posun datového signálu vůči hodinám

2008.01.XX vC

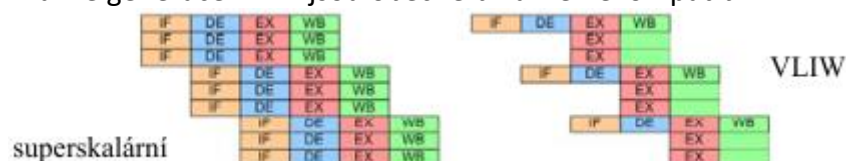
1. Uveďte charakteristické vlastnosti architektury VLIW počítačů; kde se v současnosti nejčastěji používá?

VLIW (Very Long Instruction Word) – architektura se čtením s více přístupy – delší instrukce sestavené z dílčích částí, které mohou být provedeny paralelně (instrukční packet) - např. 8 dílčích 32bitových 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.



Architektura VLIW počítačů se používá u paralelního zpracovávání procesů (superskalární procesory, statické plánování).

2. Předpokládejme vylepšení procesoru pro databázové výpočty. Nový procesor je 5krát rychlejší než nynější. Dále víme, že nyní je procesor zaměstnán z 65 % výpočty a 35 % času čeká na vstupně-výstupní operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

$$t_0 = 0,65 + 0,35 = 1$$

$$t_1 = \frac{0,65}{5} + 0,35 = 0,13 + 0,35 = 0,48$$

$$\text{zrychlení} = \frac{t_1}{t_0} = \frac{1}{0,48} = 2,08\bar{3}$$

3. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

Adresa (dec)	Hodnota (hex)	
W	AA	ADDWF 15,1
04 (FSR)	20	BTFSC 14,6
14	A0	INCF 15,1
15	FF	MOVLW 15

4. Jak bude reprezentováno číslo $(-10)_D$ v šestimístné celočíselné číslicové formě:

a) v doplňkovém kódu

b) v aditivním sudém kódu

a) $(-10)_D = \text{inv}(10_D) + 1 = \text{inv}(001010) + 1 = 110101 + 1 = 110110$

b) stačí změnit první znak doplňkového kódu $\rightarrow 010110$

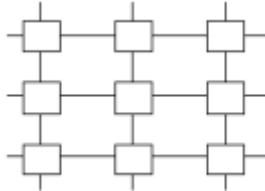
5. Charakterizujte systolická pole. Kde se používají?

Multiprocesorový systém MISD (několik procesorů zpracovává jedna 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

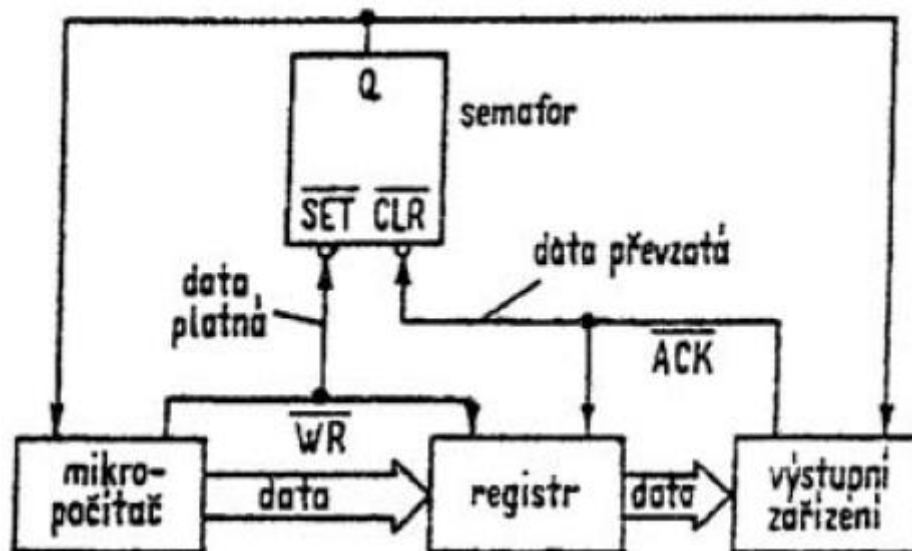


Systolická pole

- 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.

6. Co představují bloky na obrázku? Stručně popište funkci.



Komunikace procesoru se vstupně-výstupními zařízeními - V/V periferie jsou obecně pomalé oproti CPU.

Na obrázku je **Úplný podmíněný výstup**

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.

7. Jaké znáte typy neadresovatelných pamětí? Jednotlivé typy stručně charakterizujte.

Neadresovatelné paměti (LIFO - zásobník, FIFO - fronta) – nezadává se u nich, kam zapisovat či odkud číst.

8. Co je architektura souboru instrukcí, co určuje? Jaké znáte typy?

Architektura souboru instrukcí (Instruction Set Architecture, **ISA**) představuje v podstatě rozhraní mezi SW a HW počítače.

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.

Typy ISA

- střadačově (akumulátorově) orientovaná ISA (Accumulator)
- zásobníkově orientovaná ISA (Stack)
- ISA s univerzálními registry (registrová) (GPR – General Purpose Register)
 - o 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
 - o 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
 - o M-M ... oba operandy mohou být v paměti, úspora registrů, proměnná délka instrukce, paměť může být úzké místo, typické pro kompletní CISC (zastaralé).

9. Na jakém principu pracuje sběrnice USB?

USB (Universal Serial Bus) - Standard pro sériový přenos dat (4, resp. 8 vodičů – USB 3.0).

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,...).
- 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)



Architektura USB - víceúrovňová hvězdicová struktura, pouze jeden hostitel. Dva typy zařízení: hub a periferie.

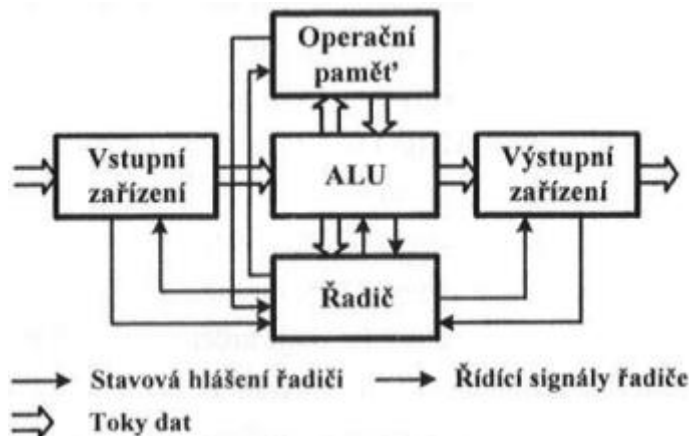
Při startu systému nebo při zapojení zařízení host: přiřadí každému zařízení identifikátor (bus ID) a zjistí od každého zařízení, v jakém režimu chtějí komunikovat:

- řídicí (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)

2008.01.09 vB

1. Uvedte charakteristické vlastnosti Von Neumanovy architektury počítačů.
představena v roce 1946

Nevýhodou je možnost mylně interpretovat data jako program.



- 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
- 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. diskrétním 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

2a. Předpokládejme vylepšení procesoru pro web. Nový procesor je 10krát rychlejší pro webové aplikace než nynější. Dále víme, že nyní je procesor zaměstnán ze 40 % výpočty a 60 % času čeká na vstupně-výstupní operace. Jaké bude celkové zrychlení po plánovaném vylepšení?

$$\begin{aligned}t_0 &= 0,40 + 0,60 = 1 \\t_1 &= \frac{0,40}{10} + 0,60 = 0,04 + 0,60 = 0,64 \\zrychlení &= \frac{t_1}{t_0} = \frac{1}{0,64} \cong 1,5625\end{aligned}$$

2b. Předpokládejme, že při FP výpočtech v počítačové grafice operace odmocniny FPSQRT odpovídá 20 % a všechny FP instrukce odpovídají 50 % doby výpočtu úlohy. Úkolem je rozhodnout, zda je výhodnější 10x zrychlit provádění operace FPSQRT nebo 1,6x zrychlit provádění všech instrukcí.

$$t_0 = 0,20 + 0,30 + 0,50 = 1$$

$$t_{\text{FPSQRT}} = \frac{0,20}{10} + 0,30 + 0,50 = 0,02 + 0,30 + 0,50 = 0,82$$

$$t_{\text{FP}} = \frac{0,20 + 0,30}{1,6} + 0,50 = 0,3125 + 0,50 = 0,8128$$

Nepatrně výhodnější je zrychlit provádění všech instrukcí.

Jiný postup:

a)

Fe = 0,2

Se = 10 Soverall = $1 / ((1-0,2) + 0,2/10) = 1 / (0,8 + 0,02) = 1,22$

b)

Fe = 0,5

Se = 1,6 Soverall = $1 / ((1-0,5) + 0,5/1,6) = 1 / (0,5 + 0,31) = 1,23$

3. Jak bude reprezentováno číslo -12 v pětimístné celočíselné číslicové formě:

a) v doplňkovém kódu

b) v aditivním lichém kódu

a) $(-12)_D = \text{inv}(12_D) + 1 = \text{inv}(01100) + 1 = 10011 + 1 = 10100$

b) stačí změnit první znak doplňkového kódu $\rightarrow 00100$ a odečíst 1 $\rightarrow 00011$

4. Uveďte nový stav registrů mikrořadiče rodiny PIC16 po provedení dané posloupnosti čtyř instrukcí:

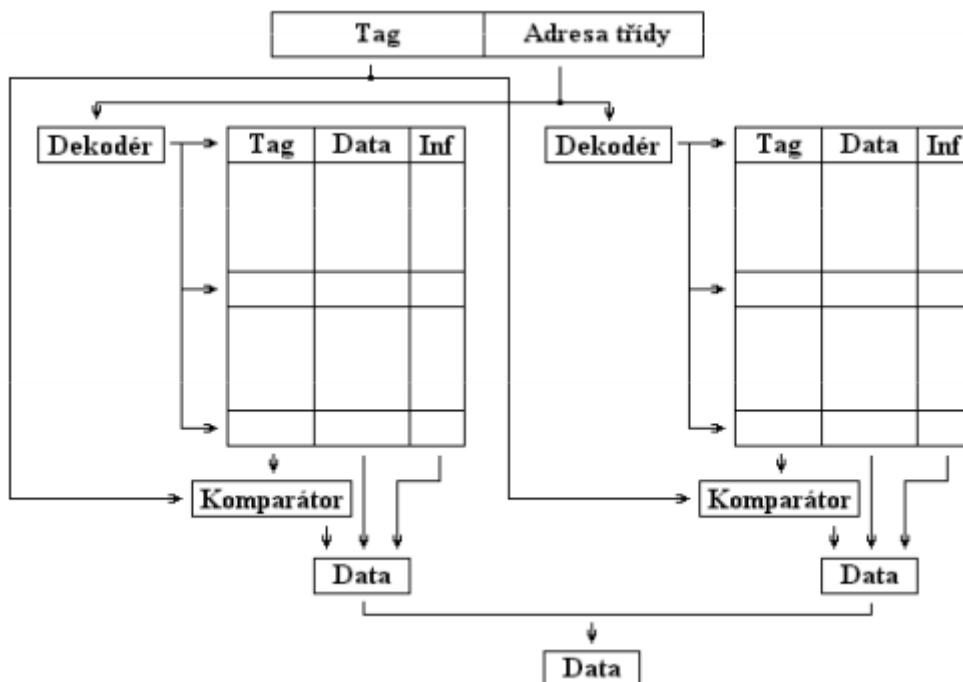
Adresa (dec)	Hodnota (hex)	
W	F2	ANDLW F4h
04 (FSR)	00	MOVF 14,1
14	FF	INCF 14,1
15	00	BCF 14,7

W = F2	04 (FSR) = 00	14 = FF	15 = 00
ANDLW F4h			
$\begin{array}{r} 11110100 \\ \text{and } 11110010 \\ \hline 11110000 \end{array}$			
W = F0	04 (FSR) = 00	14 = FF	15 = 00
MOVF 14,1			
14=01			
W = F0	04 (FSR) = 00	14 = 01	15 = 00
INCF 14,1			
14=01+01=02			
W = F0	04 (FSR) = 00	14 = 02	15 = 00
BCF 14,7			
(nuluje MSB, zde: vynulovaná 0 = 0)			
W = F0	04 (FSR) = 00	14 = 02	15 = 00

5. Popište výhodu technologie zpracování instrukcí mimo pořadí (out-of-order). Kde se používá?

Out of Order je 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.

6. Jaké zařízení je na obrázku? Stručně popište jeho funkci.



Na obrázku je n-cestně asociativní cache.

Zadaná adresa se rozdělí na 2 části: tag + adresa třídy. Adresa třídy pomocí dekodérů vybere jeden řádek v každé tabulce, jehož tagy se porovnávají (nejpoužívanější). Pro velká „n“ se blíží plně asociativní variantě.

7. Co je to cache a k čemu slouží? Jaké znáte typy?

Vyrovňovací paměť (cache, buffer) je obecně 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, který je umístěn přímo v CPU (L1, L2) nebo je součástí čipové sady (L2, L3). Cache je založena na pamětech typu SRAM. Vyrovňovací paměť může být i softwarová (SMARTDRV, VCACHE). Zvláštní vyrovnávací paměť pro pomalé periferie (HDD, vypalovací mechaniky, síťové adaptéry) – lépe označovat jako buffer.

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.

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)

8. Co víte o sběrnici SPI? Na jakém principu pracuje?

SPI (Serial Peripheral Interface) byla navržena 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

Zařízení lze k SPI připojovat paralelně nebo sériově do řetězce (aktivní všechna zařízení).

Komunikace na sběrnici mezi zařízeními Master a Slave:

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

Konfigurační módy:

CPOL – určuje polaritu SCK

CPHA – posun datového signálu vůči hodinám

9. Čím se liší aritmetická jednotka běžných signálových procesorů od aritmeticko-logické jednotky běžných procesorů?

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 sekvenční)

- součástí obvykle bývá příznakový (status) registr

Typický DSP má kromě aritmeticko-logické jednotky (ALU) navíc rychlou násobičku, která dokáže operaci násobení s přičítáním $A \leftarrow A + B \cdot k$. Tato operace je základní operací většiny algoritmů digitálního zpracování signálu.

Převod čísel do různých kódů

X	$P(X)$ přímý kód	$I(X)$ inverzní kód		$D(X)$ doplňkový kód		$A(X)_S$ sudý aditivní kód		$A(X)_L$ lichý aditivní kód
9	01001	01001		01001	inv(MSB)	11001	-1	11000
-9	11001	10110	+1	10111		00111	-1	00110
$\frac{9}{16}$	0,1001	0,1001		0,1001		1,1001	-,1	1,1000
$-\frac{9}{16}$	1,1001	1,0110	+,1	1,0111		0,0111	-,1	0,0110