

1. Zakladna koncepcia pocitacovych systemov

1.1 Informatika a pocitacove inzinierstvo

Informatika – veda o ziskavani, zbere, prenasani, triedeni, ukladani, uchovavani, aktualizovani, spracovani, vyhodnocovani a vyuzivani informacii na urovni signalov, udajov, symbolov, sprav, poznatkov, znalosti

Pocitacove inzinierstvo – zahrna navrh, konstrukciu, implementaciu, udrzbu pocitacov a pocitacmi riadenych zariadeni.

Elektrotechnika a informatika sa potrebujú.

Cislicovy pocitac = digitalny pocitac/system

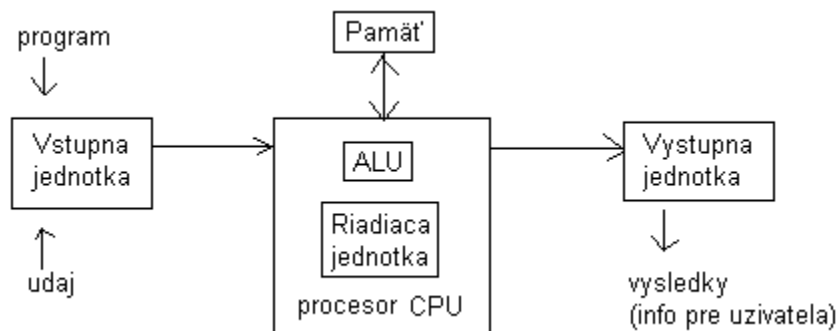
1.2 Principy pocitacov

Pocitac je zlozity, univerzalny system urceny na samocinne vykonavanie operacii nad udajmi zobrazenymi digitalnym kodom na zaklade vopred pripraveneho programu ulozeneho v pamati.

Vypoctovy proces – transformacia vstupnych udajov na vystupne

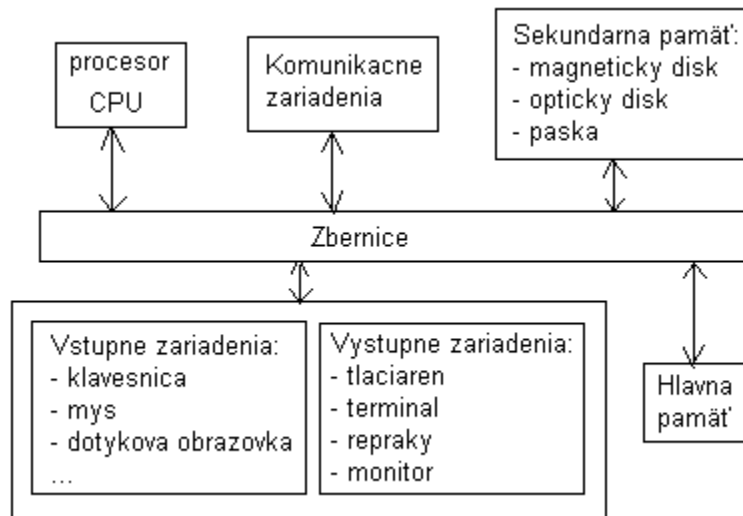
Babbage – 1836 – tkacsky stroj

1946 – Jamos (John) von Neumann



ALU – realizuje operacie

1.3 Počítačový systém v súčasnosti



Cím menej cipov, tým väčšia spoľahlivosť a rýchlosť

1.4 Klasifikácia počítačov

- technické parametre
- aplikácie určenie
- používateľský prístup
- spracovávanie informácie
- architektúra
- spôsob riadenia

Klasifikácia podľa aplikácie určenia

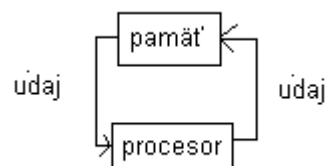


Klasifikácia podľa FLYNNA ('72)

(architektonická koncepcia)

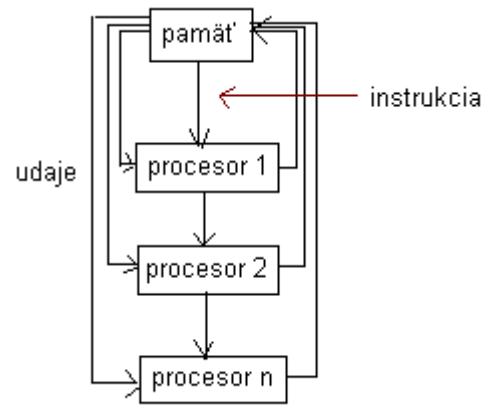
SISD (single instruction stream data)

Von Neumann – 1 procesor, sériový



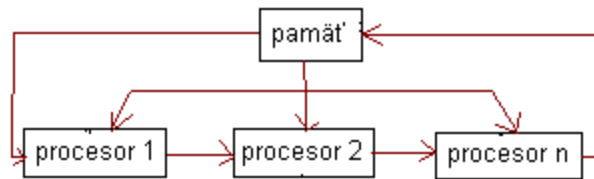
SIMD

- viac procesorov
- paralelny (maticovy, asociativny)



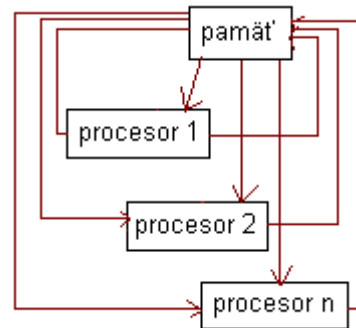
MISD

- viac procesorov
- paralelny
- prúdove spracovanie



MIMD

- viacprocesorovy
- paralelny



Klasifikacia pouzivatel'sko-aplikacna

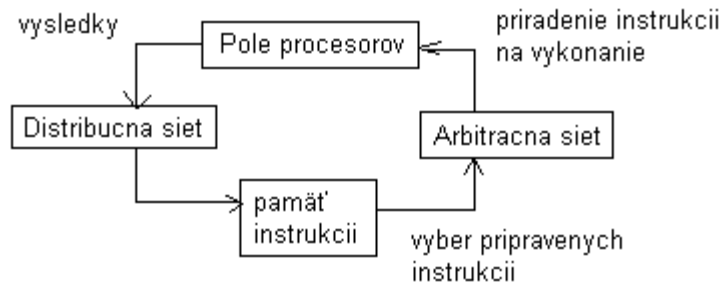
- osobne pocitace – PC (personal computer)
- pracovne stanice – WS (work station)
- specializovane pocitace
- sietove pocitace
- minipocitace (supermini, minisuper)
- superpocitace
- vlozene pocitace (embedded)

Klasifikacia podla spracovavanych informacii

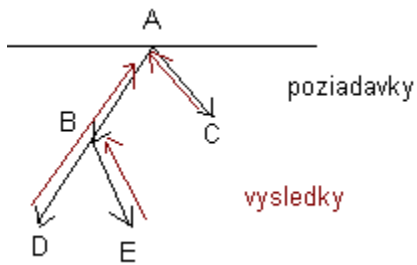
- univerzalne pocitace
- signalove procesory (DSP – digital signal processing)
- jazykove procesory
- databazove pocitacove systemy

Klasifikacia podla sposobu riadenia

- pocitace riadene tokom instrukcii
- pocitace riadene udajmi (data flow DF)

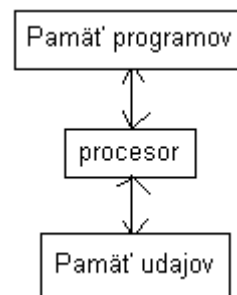
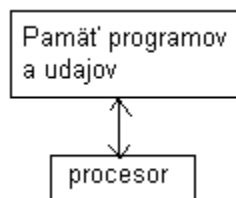


- pocitace riadene poziadavkami (DEMAND DRIVEN)



Klasifikacia podla sposobu pametania udajov a programov

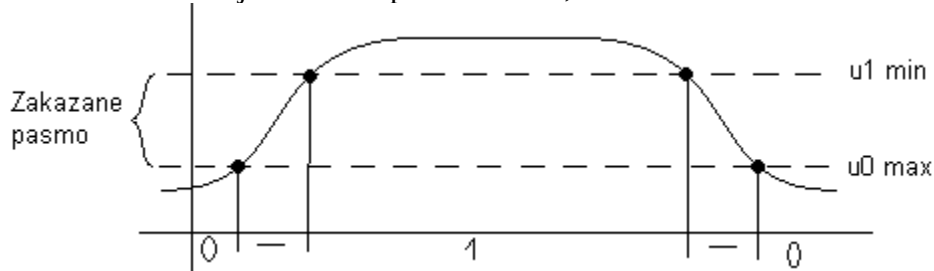
- Princetonska architektura
- Harvardska architektura



2. Logicka uroven a stavba pocitacovych systemov

2.1 Logicke obvody

- Dvojhodnotove premenne – 0, 1



1. analyza
 - struktura → spravanie
 - jednoznacna
2. synteza
 - spravanie + logicke prvky (cleny) => struktura
 - nejednoznacna
 - kriteria optimalnosti – rychlost, cena
3. simulacia
 - funkčna
 - casova
4. diagnostika
 - detekcia
 - lokalizacia

Rozdelenie logickych obvodov

- podľa funkcie:
 - kombinacne
 - sekvenčne
- podľa cinnosti v case:
 - asynchrone
 - synchronne
- podľa sposobu implementacie:
 - s pevnou funkciou
 - s programovatelnu funkciou

2.1.1 Kombinacne logicke obvody

- asynchrone
- diskretny dynamicky system

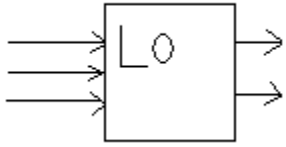
$$\underline{n} \rightsquigarrow 2^{2^n}$$

- neuplne definovana logicka funkcia => x, 0, 1

$$\underline{n} \rightsquigarrow 3^{2^n}$$

2.1.1.1 Spôsoby zapisu Boole-funkcii

- pravdivostna tabulka



	x1,2,3	y1,2
0	000	0x
1	001	00
2	010	00
3	011	11
4	100	00
5	101	11
6	110	11
7	111	1x

Ciselný zapis:

„I“ $\Rightarrow y_1 = D(3,5,6,7)$

„0“ $\Rightarrow y_1 = K(0,1,2,4)$

„I“ $\Rightarrow y_2 = D(3,5,6,0,7)$

„0“ $\Rightarrow y_2 = K(1,2,4,0,7)$

$y_1 = (00010111)$

Mapový zapis:

- Karnaugh (karnauf)

2^n stvorčekov

Vyraz:

- B-vyraz :[+,*, -]

2.1.1.2 B-funkcie s jednou premennou

$n=1 \Rightarrow 2 \text{ na } 2^n = 4$

$y_0=0$

$y_1=1$

$y_2=x$

$y_3=!x$

$x \mid \begin{array}{ c } \hline 0 \\ \hline 0 \\ \hline \end{array}$	$x \mid \begin{array}{ c } \hline 1 \\ \hline 1 \\ \hline \end{array}$	$x \mid \begin{array}{ c } \hline 0 \\ \hline 1 \\ \hline \end{array}$	$x \mid \begin{array}{ c } \hline 1 \\ \hline 0 \\ \hline \end{array}$
y_0	y_1	y_2	y_3

2.1.1.3 B-funkcie s dvomi premennymi

$x_1 x_2$	f															
$x_1 x_2$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
10	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Napr:

$f_0=0$

$f_1 = !(x_1 + x_2) = !f_{14}$

$f_{14} = x_1 + x_2$

$f_{15} = 1$

$n > 2$

speciálne funkcie:

- majoritné
- prahové
- symetrické

2.1.1.4 Boolovský výraz

- retazec:

- premenne a ich negácie
- logické operatory
- zátvorky

Logický výraz: $(a \oplus \bar{b}) \rightarrow (x_1 \bar{x}_2 + x_3)$

Boolovský výraz: $a + [b\bar{c}(\bar{a} + cd) + ce]$ \Rightarrow obsahuje $[+ \cdot \bar{\cdot}]$

Co všetko je výraz:

- premenne x_1, x_2, x_3, x_4, x_n
- ak A je výraz, tak \bar{A} je výraz
- ak A a B sú výrazy, tak aj $A+B$ a $A.B$ sú výrazy

2.1.1.5 Boolovská algebra

$B = \{B^n, +, \cdot, \bar{\cdot}, 0, I\} \Rightarrow$ množina všetkých boolovských výrazov

Pre B – algebru platia tieto ekvivalencie

- | | | |
|---|--|-----------------------------|
| 1. $a+b=b+a$ | $a.b=b.a$ | komutativnosť |
| 2. $a+(b+c)=(a+b)+c$ | $a.(b.c)=(a.b).c$ | asociativnosť |
| 3. $a.(b+c)=a.b+a.c$ | $a+(b.c)=(a+b)(a+c)$ | distributivnosť |
| 4. $a.a.a.a.a...=a$ | $a+a+a+a...=a$ | |
| 5. $\overline{a+b+c} = \bar{a}.\bar{b}.\bar{c}$ | $\overline{a.b.c} = \bar{a} + \bar{b} + \bar{c}$ | de Morganove zákony |
| 6. $\bar{\bar{a}} = a$ | | dvojnásobná negácia |
| 7. $a + \bar{a} = I$ | $a.\bar{a} = 0$ | pravidla komplementa |
| 8. $a+I=I$ | $a.0=0$ | agresivnosť nuly a jednotky |
| 9. $a.I=a$ | $a+0=a$ | |
| 10. $(a+b)(\bar{a}+\bar{b})=b$ | $a.b + \bar{a}.\bar{b} = b$ | spojovanie |
| 11. $a+a.b=a$ | $a.(a+b)=a$ | pohltenie |
| 12. $a.(\bar{a}+b)=a.b$ | $a + \bar{a}.b = a + b$ | |

2.1.1.6 Normálne formy B – výrazov

- Elementárny súčin – $x_1.x_2.x_3$ (rad súčinu - 3)
- Elementárny súčet – $x_1+x_2+x_3$ (rad súčtu - 3)

1. Disjunktívna normálna forma (DNF)
 $f = \sum g_i$
2. Konjunktívna normálna forma (KNF)
 $f = \prod h_j$

- A. Uplná DNF (UDNF)
- Obsahuje elementárne súčiny
 - Jedina

		x2		x3	
		0	1	0	1
x1	0	0	0	1	0
	1	0	1	1	1

$$\text{UDNF: } y = \overline{x_1}x_2x_3 + x_1x_2\overline{x_3} + x_1x_2x_3 + x_1\overline{x_2}x_3$$

$$\text{UKNF: } (\overline{x_1} + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(x_1 + x_2 + \overline{x_3})$$

B. Skratena DNF (SDNF)

- Aplikacia pravidiel spojovania a pohltenia na UDNF

C. Iredundantna DNF (IDNF)

- Neredukovatelna
- Viacero pre jednu funkciu

D. Minimalna DNF (MDNF)

- Najmensi pocet pismen zo vsetkych IDNF
- Moze byt viac pre jednu funkciu

Priklad v poznamkach (9.list)

2.1.2 Analýza logických kombinacných obvodov

- štruktúra + správanie logických členov => vo forme funkcie (mapa, výraz, tabuľka)
- jednoznačné riešenie

2.1.3 Syntéza logických kombinacných obvodov

- správanie sa + súbor typov logických členov => štruktúra (kritéria optimálnosti – rýchlosť, cena)
- jednoznačná úloha

Postup:

- Hľadá sa skupina výrazov zodpovedajúca danej skupine B-funkcií
- DNF/KNF/MDNF (algoritmizovateľne)

2.1.3.1 Vyjadrovanie IDNF (IKNF) z mapy

- implikant funkcie – súčin
- prostý implikant – súčin (najjednoduchší)

		x2		x3	
		0	1	0	1
x1	0	1	0	1	1
	1	1	1	1	0

- implicit funkcie – súčet
- prostý implicit – súčet

Pravidelná konfigurácia:

1. obsahuje 2^s stvorčekov (s – stupeň konfigurácie)
2. každý stvorček v konfigurácii musí mať susedov

Pokrytie:

- najmenej, najväčšie (každý bod pokrytý minimálne raz)

Příklad:

		x3		x4	
x2	x1	0	0	1	1
		1	0	1	X
	x1	X	1	1	0
		0	1	1	0

MDNF:

$$\overline{x_1}x_4 + x_1x_3 + x_2\overline{x_3}x_4$$

MKNF:

$$(\overline{x_1} + x_3)(x_1 + \overline{x_3} + x_4)(x_1 + x_2 + x_4)$$

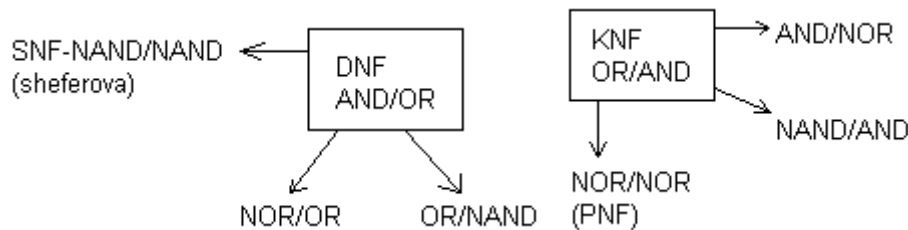
		x3		x4	
x2	x1	0	0	1	1
		1	0	1	X
	x1	X	1	1	0
		0	1	1	0

2.1.3.2 Zmiesane normalne formy

+ . - \uparrow

Dvojstupnové obvody (najrychlejšie)

- neohraniceny pocet logickych clenov
- negacie vstupnych premennych ???



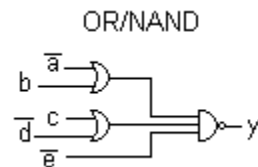
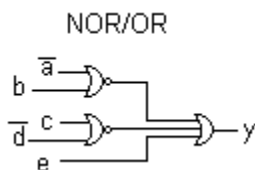
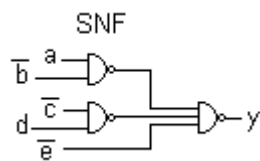
Příklad:

DNF $y = a\overline{b} + \overline{c}d + e$

SNF $y = (a \uparrow \overline{b}) \uparrow (\overline{c} \uparrow d) \uparrow \overline{e}$

NOR/OR $y = (\overline{a} \downarrow b) + (c \downarrow \overline{d}) + e$

OR/NAND $y = (\overline{a} + b) \uparrow (c + \overline{d}) \uparrow \overline{e}$



Příklad2:

KNF OR/AND

$$y = (\overline{a} + \overline{b})(c + \overline{d})e$$

PNF NOR/NOR

$$y = (\overline{a} \downarrow \overline{b}) \downarrow (c + \overline{d}) \downarrow \overline{e}$$

NAND/AND

$$y = (a \uparrow b)(\overline{c} \uparrow d)e$$

AND/NOR

$$y = (a.b) \downarrow (\overline{c}.d) \downarrow \overline{e}$$

2.1.4 Sekvencne obvody

1. synchronne
2. asynchronne
 - A. s pevnou funkciou
 - B. s programovateľnou funkciou
- STAV – historia
- spravenie – konečný deterministický automat

2.1.4.1 Konečný automat

Def: pática (sestica)

$$A = (X, S, Y, p, v, (s_0))$$

X – vstupy

S – stavy

Y – vystupy

Moore $V: S \rightarrow Y$

Nealy $V: S \times X \rightarrow Y$

2.1.4.2 Spôsoby zapisu automatu

1. prechodová tabuľka

Priklad 1: (Nealy)

		x				x			
		x1		x2		x1		x2	
S	1	1	2	2	3	0	1	0	1
	2	1	3	2	2	1	0	1	1
	3	2	3	3	1	0	1	0	1
		p				v			

$p: S \times X \rightarrow S$
 $v: S \times X \rightarrow Y$
 $S = \{1, 2, 3\}$
 $X = \{00, 01, 10, 11\}$
 $Y = \{0, 1\}$

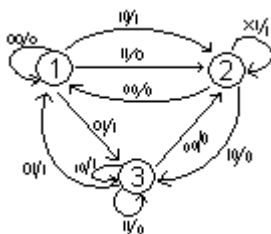
Priklad 2: (Moore)

		x		y	
		x	y	x	y
S	1	1	2	0	
	2	3	2	1	
	3	3	4	1	
	4	1	4	0	
		p		v	

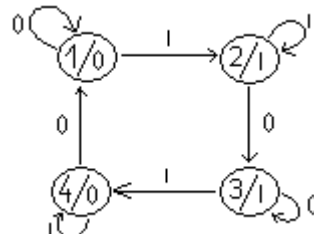
$p: S \times X \rightarrow S$
 $v: S \rightarrow Y$
 $S = \{1, 2, 3, 4\}$
 $X = \{0, 1\}$
 $Y = \{0, 1\}$

2. prechodový graf

Priklad 1:



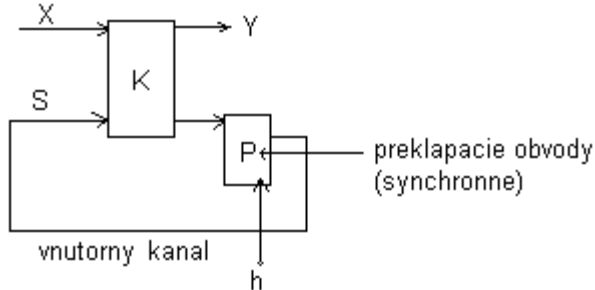
Priklad 2:



2.1.5 Struktúrna syntéza synchronných sekvencných obvodov

- minimálny počet preklapacích obvodov v P
- minimálny počet logických členov v K
- maximálna operačná rýchlosť

Model SSO:

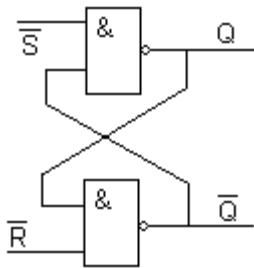


K – kombinačná časť SSO

P – pamäťová časť SSO

2.1.5.1 Synchronné preklapacie obvody

- základ → asynchronný PO-SR

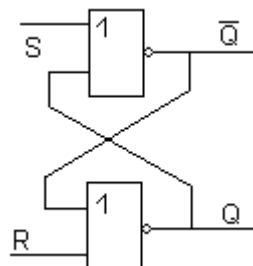


S – set (1)
R – reset (0)
 $PO \rightarrow \overline{S}R$
 $\overline{S} + \overline{R} = 1$

funkcia správania sa

\overline{S}	\overline{R}
X	0
0	P
1	1

S.R=0



S	R	Q
0	0	1
0	1	X
1	0	0
1	1	X

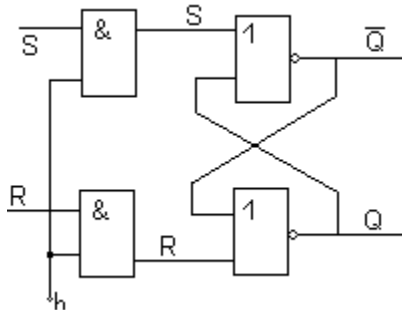
PO-SR

funkcia správania sa

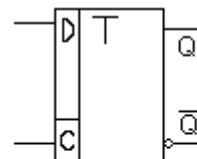
S	R
P	1
1	X
0	0

→ nulové
→ jednotkové
→ pamäťové

Synchronný PO-SR



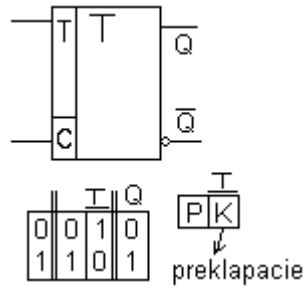
Synchronný PO-D



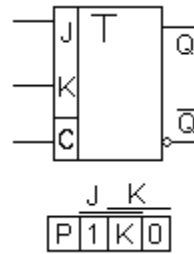
D	Q
0	0
0	1
1	0
1	1

D
0
1

Synchronny PO-T



Synchronny PO-JK



2.1.5.2 Postup pri synteze SSO

PO-SR

PO-JK $\leftarrow P$

PO-T

PO-D

1. Abstraktna synteza – slovny opis spravania (stavy?), automat
2. Strukturna synteza – automat + PO + logicke cleny

2.1.5.3 Priklad

Kontrolor kodu

PO-JK, NAND

Legalne – 0, 1, 2, 3, 4, 5

Nelegalne – 6, 7

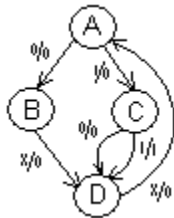
	X_1	X_2	X_3	
0	0	0	0	} $Y=0$
1	0	0	1	
2	0	1	0	
3	0	1	1	
4	1	0	0	
5	1	0	1	} $Y=1$
6	1	1	0	
7	1	1	1	

1. Abstraktna synteza

- Definovanie stavov

- graf

$S=\{A, B, C, D\}$



		x	x
A	B	C	0
B	D	D	0
C	D	D	1
D	A	A	0

p v

2. Strukturna synteza

Kodovanie stavov – pocet vnutornych (stavovych) premennych $|s| \leq 2^{k_{\min}}$

$\varphi: S \rightarrow \{0,1\}^k$

Kodovacia mapa:

	Z_2
Z_1	A D
	B C

φ

A...00
B...10
C...11
D...01

Dvojkový ekvivalent:

z_1	z_2	\overline{x}	x
00	10	11	00
10	01	01	00
11	01	01	01
01	00	00	00

z_2	x
1	1
0	0
0	0
0	0

z_2	x
0	1
0	0
1	1
1	1

k-map
n+k-premenných

Mapa výstupných premenných funkcie:

z_2	x
0	0
0	0
0	1
0	0

m-nap
n+k-premenných
 $y = z_1 \cdot z_2 \cdot x$

Inverzná prechodová funkcia PO

$0 \rightarrow 0 \rightarrow 00$	$0 \rightarrow 1 \rightarrow 10$	$1 \rightarrow 0 \rightarrow 11$	$1 \rightarrow 1 \rightarrow 00$
01	11	01	10
$0X$	$1X$	$X1$	$X0$

z	J	K	Q
0	0	1	0
1	1	0	0

z

$z \rightarrow Z$	J	K
$0 \rightarrow 0$	0	X
$0 \rightarrow 1$	1	X
$1 \rightarrow 0$	X	1
$1 \rightarrow 1$	X	0

z_1	z_2
1	1
0	0
X	X
X	X

J_1 K_1 J_2 K_2

$J_1 = \overline{z_2}$
 $K_1 = I$
 $J_2 = z_1 + x$
 $K_2 = \overline{z_1}$
 $y = z_1 \cdot z_2 \cdot x$

3. Hlavné podsystemy digitalných počítačov

von Neuman – SISD

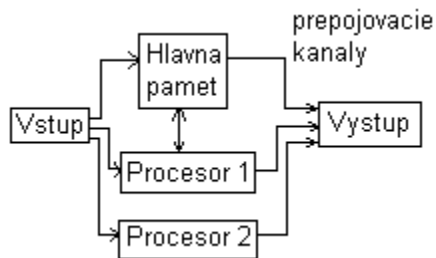
- seriové – riadene tokom instrukcii
- jednoprocesorové

3.1 Prepojovací systém počítača

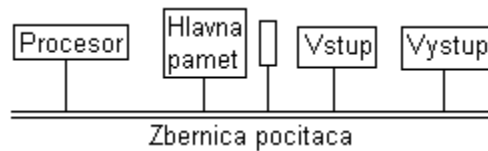
Prepojenie častí počítača

- čítanie instrukcií z pameti
- čítanie údajov z pameti
- zápis údajov do pameti

Prepojovacie kanály:



Zbernice:



Pridelovanie zbernice

Odpojovanie od zbernice

3.1.1 Rozdelenie zberníc

Podľa spôsobu riadenia:

1. Single – master
Master – nadriadený
Slave – podriadený
2. Multi – master
Jedno zariadenie

Podľa synchronizácie prenosu:

1. synchoronne zbernice
spoločný jeden zdroj synchronných impulzov
zariadenia s rovnakou rýchlosťou
2. asynchoronne zbernice
s nerovnakou rýchlosťou → čaka sa na prenos každého údaje

Podľa časového multiplexu:

1. Multiplexované zbernice
- význam/druh prenášanej informácie sa mení s časom
2. Nemultiplexované zbernice

Podľa tvaru prenášaných údajov:

1. Paralelne zbernice
- naraz sa prenáša viacbitový údaj
2. Seriové zbernice
- bit po bite
- šírka zbernice
3. Seriovo – paralelne

3.1.2 Štruktúra typickej počítačovej zbernice

Charakteristika:

- paralelná
- asynchrónna
- nemultiplexovaná

Sekcie:

- adresová
- údajová
- riadiaca

Adresová:

- prenášajú sa adresy
- adresy generuje nadriadený prvok (procesor)
- adresuje sa pamäťová bunka
- vstup-výstup zariadenie

Údajová:

- prenášajú sa inštrukcie – vstupujú do operácie
- nikdy sa neprenášajú naraz

Riadiaca:

- prenáša povely – čítanie, zápis....(doména nadriadeného)
- prenáša ziadosti – priradenie zbernice, ... (podriadený systém)

- signálové sledy → časové priebehy

3.2 Základná koncepcia procesora

Procesor – základná časť počítača (SISD)

- hlavná riadiaca autorita
- interpretuje inštrukcie

HP – hlavná pamäť – program – postupnosť inštrukcií

- vyberá inštrukcie z HP
- vykonáva sa operácia s operandami
- prenos informácie medzi po...ymi časťami počítača

MIMD, SIMD, MISD

Druhy procesorov:

Podľa vykonávanej funkcie:

- univerzálne
- problémovo-orientované

Univerzálne UP – číselné a nečíselné operácie;

- riadenie ostatných častí počítača

→ program (HP)

CISC, RISC, NISC

Problémy procesorov:

- efektívnosť
- rýchlosť, výkonnosť

POP – problémovo orientované

- vykonávajú špecializované funkcie

- programove prostriedky
- vstupno-vystupne procesy
- numericke koprocessory (pracuju s cislami s pohyblivou radovou ciarkou)
- graficke procesory
- logicke procesory
- robia aj to co UP

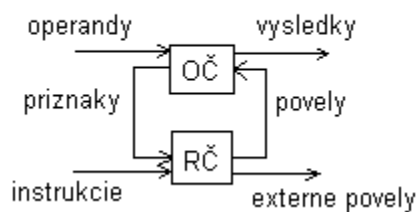
3.2.1 Hlavne casti procesora

Operacna cast – operacie s operandami

Riadiaca cast – vybera a dekoduje instrukcie

- zabezpeci vykonavanie v operacnej casti

PROCESOR

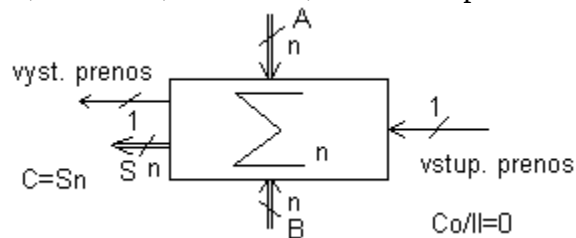


3.2.1.1 Operacna cast

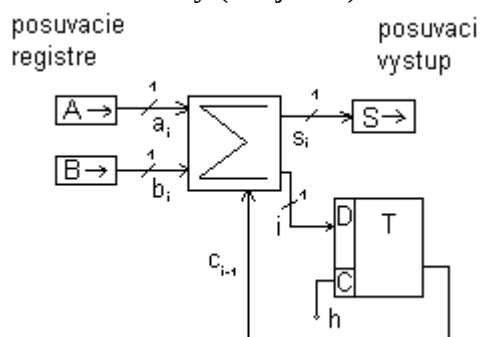
- aritmeticko-logicka jednotka (ALU) – inf. spracuvavaju s operandami
- registre – zapisnikova pamet
 - docasne uchovanie informacie
 - mala pamet, ale najrychlejsia cast v pameti
- komunikacne obvody – medziregistrove prenosy

3.2.1.1.1 ALU

- scitanie, odcitanie, nasobenie, dalenie → paralelna dvojkovala scitacka

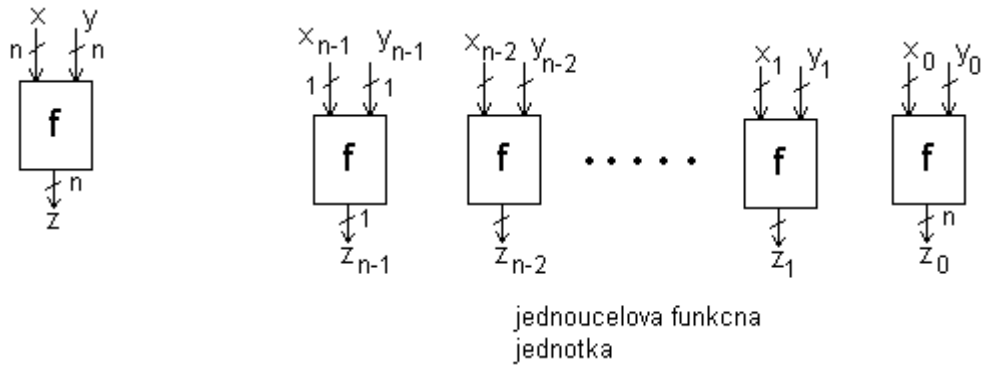


Serieove scitacky (dvojkovala)

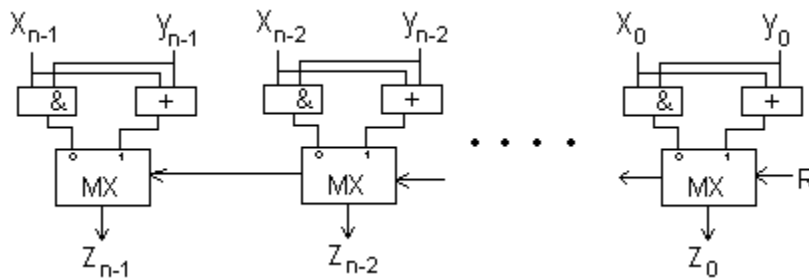


Logicke operacie:

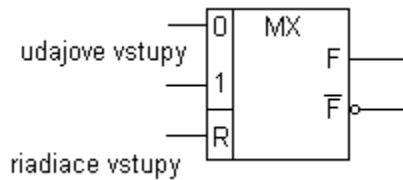
- funkčne jednotky



Dvojfunkčná jednotka (AND, OR)

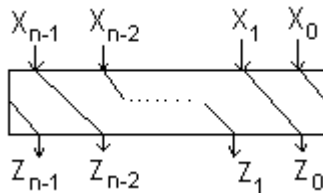


MX – multiplexor – kombinacný obvod

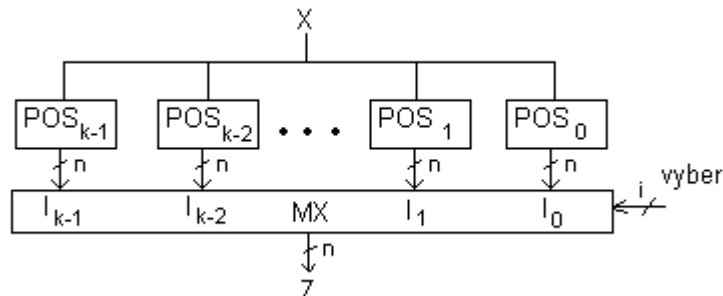


Posuvy:

- aritmetické operácie
- násobenie, delenie s 2^n (posuvacie obvody)



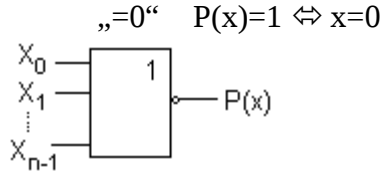
- logický posuv, kruhový posuv, aritmetický posuv (okrem znamienka)



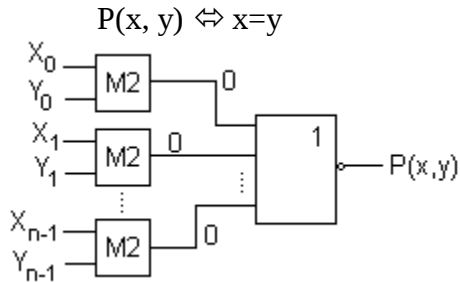
Predikaty (priznaky)

- dvojhodnotove funkcie
- nad jedným operandom [$=0$, <0 , >0]
- nad dvomi operandami [$x < y$, $x > y$, $x \leq y$, ...]

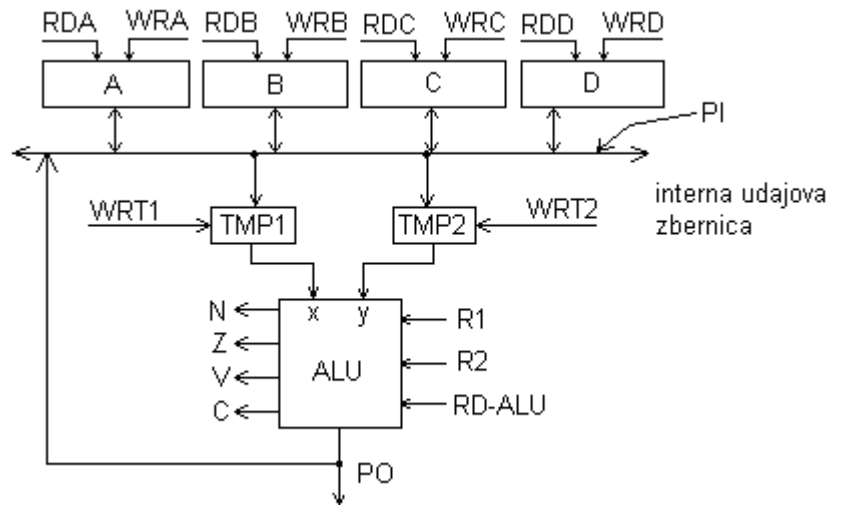
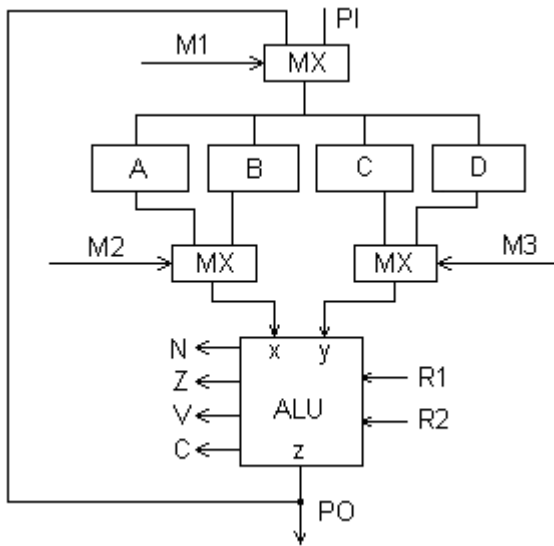
Priklad:



Priklad:



- a) multiplexory a demultiplexory
- b) zbernice



3.2.1.2 Riadiaca časť procesora

- uskutočňuje výber a dekodovanie instrukcií (IR), vykonanie instrukcií, riadenie „okolia“

Instrukcny cyklus (6.faz):

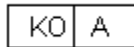
- 1.)
- 2.)
- 3.)
- adresa)
- 4.)
- 5.)
- 6.)
- vystup. zariadenia

vyber instrukcie – IF (instructio fetch)
dekodovanie instrukcie – D (decode)
vypocet adresy operandu – OA (operand

vyber operandov – OF (operand fetch)
vykonanie operacie – E (execution)
zapis vysledku – S (store); do pamete,

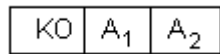
3.2.1.2.1 Format instrukcie

Instrukcia – prikaz pre procesor



KO – kod operacie

A – adresy – 0, 1, 2 (najcastejsie)



3.2.1.2.2 Typy instrukcii

Posunove:

- $R \rightarrow R$
- $R \rightarrow M$
- $M \rightarrow R$
- $R \rightarrow VV$
- $VV \rightarrow R$
- $M \rightarrow M$
- $VV \rightarrow M$

Vypoctove:

- aritmetické
- logické
- ine

Skokove:

- podmienene
- nepodmienene
- skok do podprogramu (FIFO, LIFO)

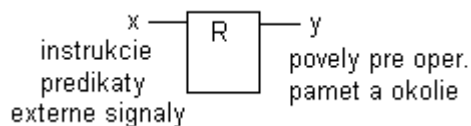
Riadiace (dali by sa zaradit do tych pred tym)

3.2.1.2.3 Sposoby adresovania operandov

		v instr.	v registri	v pameti
impli- citne	1	registrove	operand	
	2	zasobnikove	adresa	operand
	3	nepriame reg.	adresa	operand
expli- citne	1	priame	adresa	operand
	2	nepriame	adresa1	adresa2
	3	bezprostredne	operand	
	4	registrove	adresa reg.	operand
	5	nepriame reg.	adresa reg.	operand
	6	indexove	adresa, index reg.	index
	7	bazovo-index.	baza reg., index reg.	baz.adresa, index

3.2.1.2.4 Riadiaca cast s pevnou funkciou

Synchronny sekvenčný obvod (dana štruktúrou)

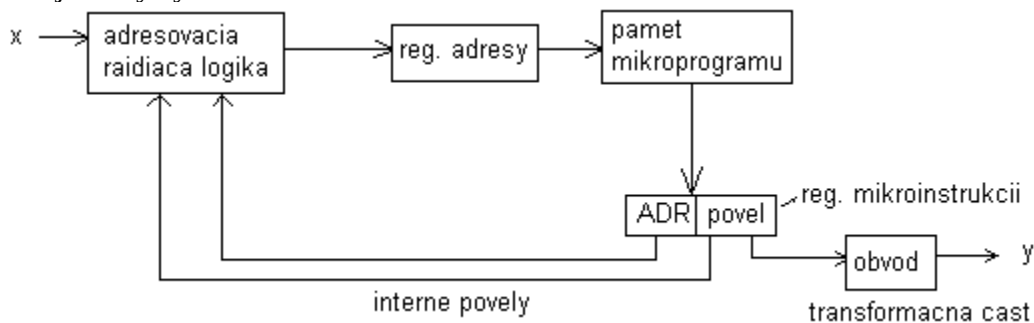


3.2.1.2.5 Mikroprogramovatelna riadiaca cast

Synchronny systém – mikroprogram vykonáva postupnosť mikroištrukcií

(špecializovaný procesor)

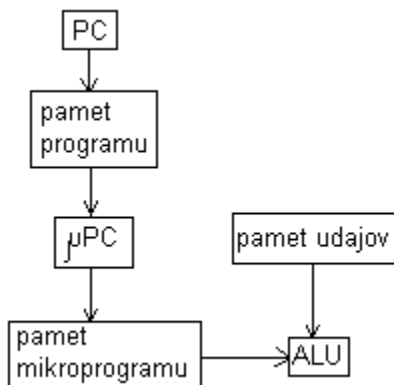
Dvojfazový systém



3.2.2 CISC, RICS, NISC procesory

CISC (complex instruction set computer)

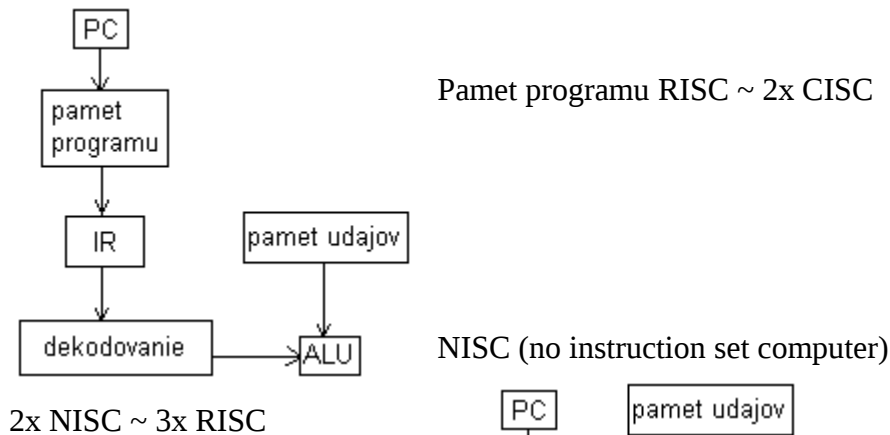
- draha a pomala pamet
- vykonne instrukcie
- princíp realizácie instrukcie:



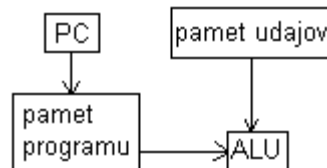
INTEL

1 instrukcia = n mikroištrukcií

RISC (reduced instruction set computer)



NISC (no instruction set computer)



3.2.3 Zvyšovanie vykonnosti procesorov

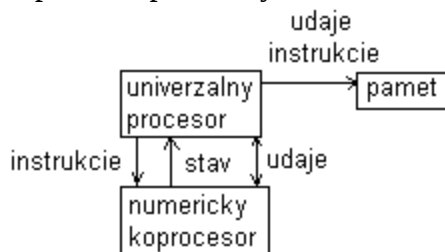
Parameter – počet operácií za sekundu (Hz)

Možnosti zvyšovania:

- 1.) zdokonalovanie technológie – hranica 3,2 GHz
- 2.) zdokonalovanie organizácie spracovania informácie - neohranicena moznost

3.2.3.1 Vypočty v pohyblivej radovej ciarke

- špeciálne procesory – numerické koprocessory (pasívny procesor)



3.2.3.2 Predvyber a predspracovanie instrukcii

Zbernica – využitie → vyber a dekodovanie

- vykonna faza
- vyberova faza

3.2.3.3 Prudove spracovanie instrukcii

IF D OA OF Ex S

1 – instrukcia – 6 taktov (faz)

Priklad: (RISC)

- | | | | | | | |
|---------------|----|----|----|----|----|---------|
| 1. instrukcia | IF | D | OA | OF | Ex | S |
| 2. instrukcia | | IF | D | OA | OF | Ex S |
| 3. instrukcia | | | IF | D | OA | OF Ex S |

3 instrukcie → 8 taktov (operácie trvajú rovnaký čas)

Hazardy, konflikty – moze vzniknut problem s udajmi

- udajova nezávislosť
- riadiaca nezávislosť

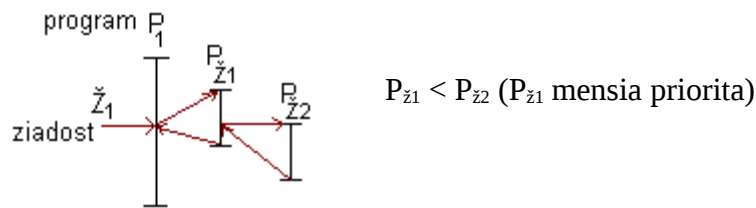
3.2.3.4 Paralelne vykonavanie instrukcii

1. instr. IF D OA OF Ex S

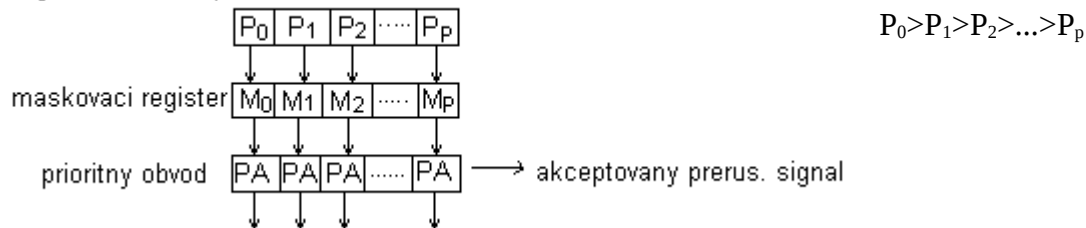
2. instr. IF D OA OF Ex S

→ udajova, riadiaca nezávislosť

3.2.4 Prerušovací systém procesora



register zidosti o prerusenie



3.2.4.1 Klasifikacia preruseni

Podla zdroja prerusenych signalov:

- externe
 - zdroj – okolie procesora alebo pocitaca
 - nesuvisi s prave vykonavanim programom
- interne
 - suvisi s prave vykonavanim programom
 - nemožno ho zakázat

Podla toho kedy prerusenie moze nastat:

- asynchrone
 - nemusí súvisieť s vykonávaným programom
 - môžu nastať kedykoľvek
 - možno zakázat
- synchronne
 - súvisia s vykonávaným programom
 - nemožno zakázat

Podla toho či možno alebo nemožno zakázat:

- maskovateľne

- externe
- asynchrone
- b) nemaskovateľne
 - udalosť sa nemôže maskovať
 - činnosť sa vždy preruší
 - mimoriadne dôležité udalosti

Synchronne prerušenia:

- a) softvérové – špecializuje riadiaca inštrukcia
- b) výnimka (exception) – pri vykonávaní inštrukcie

3.2.4.2 Operácia prerušenia

6 krokov:

1. prijatie požiadavky prerušenia
 - udalosť – dokončenie práve vykonanej inštrukcie; obsluha prerušenia
2. odloženie stavu procesora
 - zásobník – PC (stav programového počítadla)
 - hniezdenie prerušení
 - priorita
3. zistenie zdroja prerušenia
 - prerušovací vektor – adresa začiatkovej inštrukcie obslužného programu
4. vykonanie obslužného programu prerušenia
 - návrat do prerušeneho programu
5. obnovenie pôvodného stavu
 - stav procesora pred prerušením
6. pokračovanie vykonávania prerušenia programu

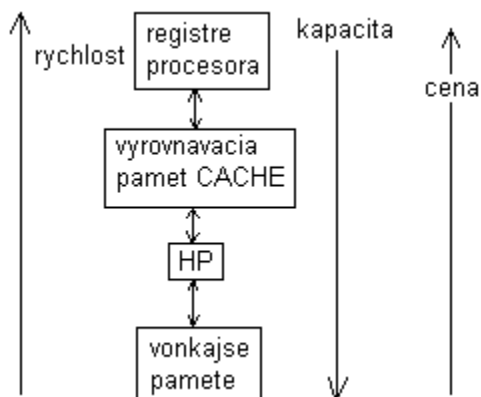
3.3 Pamätový podsystem počítača

Služi na uloženie programov a údajov

Ideál: procesor – pamäť

→ obrovská kapacita, vysoká rýchlosť, nízka cena

3.3.1 Hierarchická organizácia pamätového podsystemu počítača



3.3.2 Rozdelenie pameti

Spôsob prístupu k informácii:

- pamete s priamym pristupom DAM (direct access mem) RAM (random)
- pamete so sekvencym pristupom SAM (sequential access memeory)
- pamete s asociativnym pristupom CAM (countent access memory) - kluc

Moznost citania a zapisu:

- pamete pre citanie a zapis (RWM – read-write-mem)
 - energeticky zavisle; nezavisle
- pamete iba pre citanie (ROM – read-only-mem)
 - pouzivatel, vyrobca

Dĺžka uchovávaného slova:

- pamäte s bitovou organizáciou
- pamäte so slovnou organizáciou (slovo - byte)

Tvar prenášaných údajov:

- sériové pamäte
- paralelne pamäte

Organizácia pametových buniek:

- polovodičové; preklapací obvod; RWM
- statické pamäte – preklapací obvod TTL (SRAM)
- dynamické pamäte – DRAM, parazitná kapacita
- obnovovanie obsahu – refresh

3.3.3 Hlavná pamäť

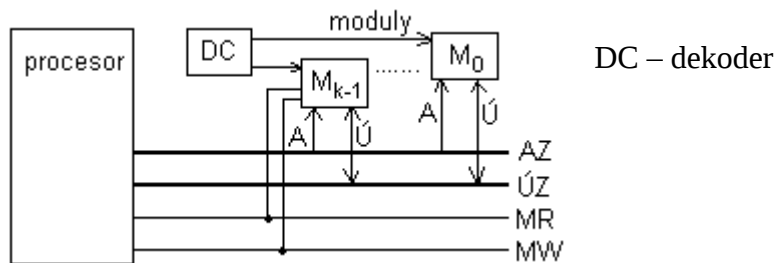
Uchováva práve vykonávaný program, údaje.

DAM, RWM, polovodičová – SRAM, DRAM

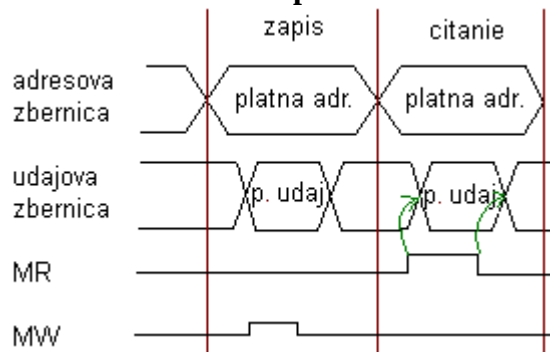
Slovné organizované, paralelne

3.3.3.1 Pripojenie hlavnej pamäte k zbernici počítača (procesora)

- adresová
- údajová
- riadiaca



3.3.3.2 Komunikácia procesora s hlavnou pamäťou



3.3.4 Vyrovnávacia pamet

Procesor \Leftrightarrow VP (CACHE) \Leftrightarrow HP

Nekonzistencia informacii:

- zapis spat (copy-back)
- zapis cez VP (write-through)
- vyber riadku – LRU – najdlhsi cas nepouzity (least-recently used)
LFU – najmenej casto pouzivany riadok (least-frequently used)

3.3.5 Vonkajcie pamete

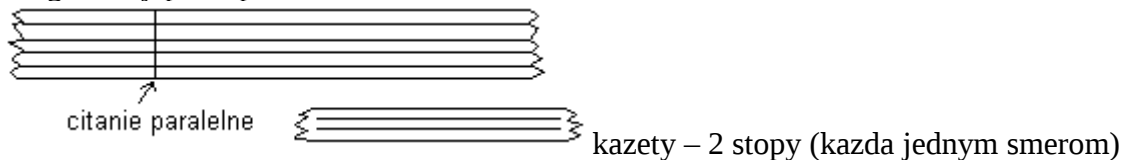
Archivacia, prechodne uchovavanie informacii pocas vypoctu

3.3.5.1 Diskove pamete

- magneticke
- opticke
- opticko-magneticke

3.3.5.2 Paskove pamete

Magneticky princip



- cievkove SAM
- kazetove RWM (archiv)

neparna polarita: 00000000|I 8+1

3.3.6 Sprava a ochrana hlavnej pamete

Absolutna adresa – AZ FPP

Fyzicky pametovy priestor (kapacita HP)

Logicky pametovy priestor (program + udaje) LPP

LPP=FPP \rightarrow ideal

Priciny:

- LPP>FPP (virtualna pamet) – potrebna rychla vonkajsia (hard disk)
- Viac pouzivatelov \rightarrow viac programov v HP; snaha o dynamicku relokovatelnost (rozne absol. adresy)
- Pristupne prava \sim zapis (suvisi aj s ochranou)
Sprava – HW+SW (ideal – spracovavat hardverovo)
MMU – memory manager unit

3.3.6.1 Virtualna pamet

- segmentovanie
- strankovanie

Segmentovanie:

- program a udaje
- rozdelené na bloky – segmenty
- 1. segment – program
- 2. segment – podprogramy
- 3. segment – udaje

Logické adresy



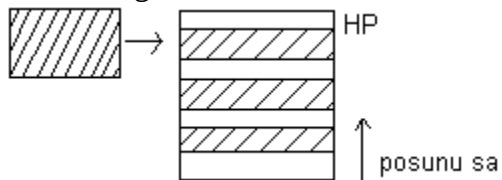
Príznaky (deskriptor):

1. bazová adresa segmentu (zaciatok segmentu)
2. veľkosť segmentu (limit) $\text{offset} < \text{limit}$ (ochrana)
3. vlastnosti (atribúty) segmentu
4. adresa segmentu vo vonkajšej pameti

Absolútna adresa → baza + posunutie

Relatívne adresovanie

Externá fragmentácia



Stránkovanie – HP rozdelená na úseky s rovnakou dĺžkou → stránky

20.1.2009 – 13.30

100min = 1h 40min

Písať tvorivo, neočakáva sa písanie skript (čo sa dá – obrázok, inak heslovite)

PSS – 20 otázok INFO – 16 otázok (po 5 bodoch)

Xena