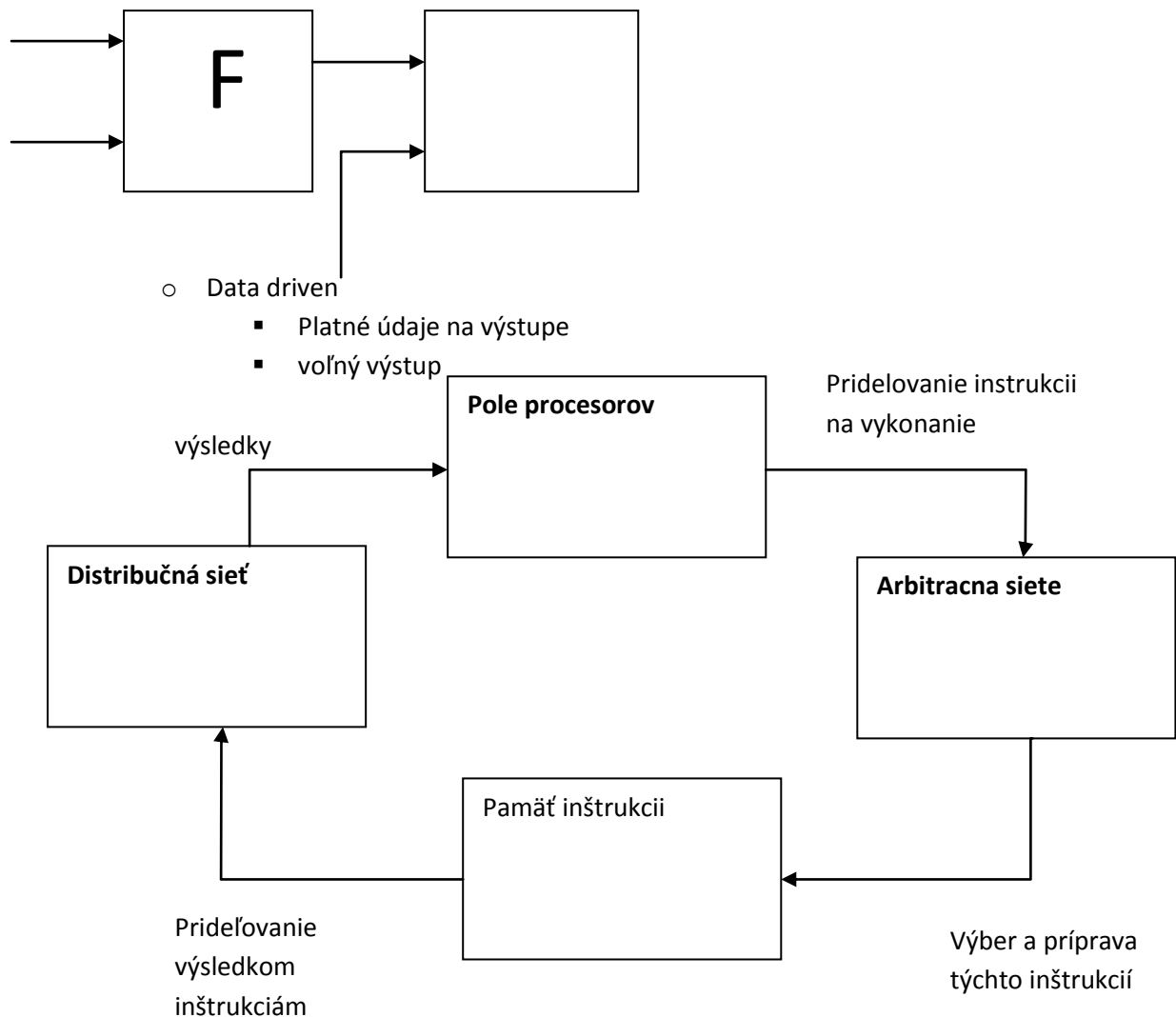


Klasifikácia podľa spracúvaných informácií

- univerzálne počítače
- signálové počítače (DSP digital signal procesing)
- jazykové procesory
- databázové počítačové systémy

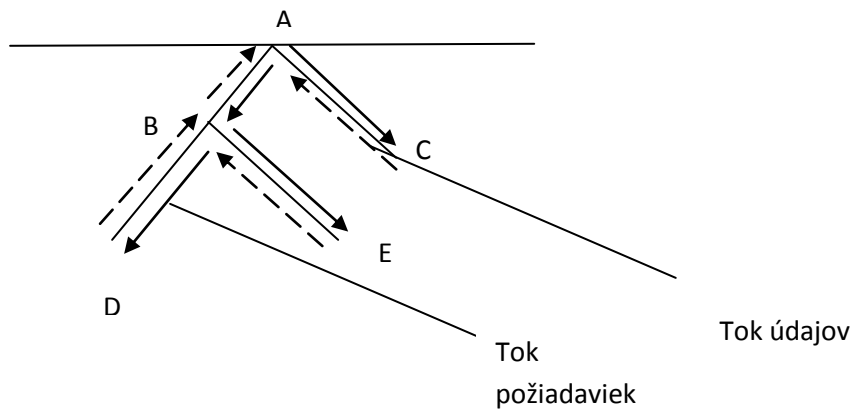
Klasifikácia podľa spôsobu riadenia

- počítače riadené tokom inštrukcií
 - control flow
 - program
- Počítače riadené údajmi
 - data flow



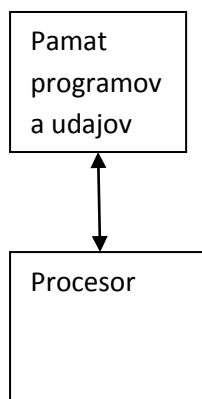
Počítače riadené požiadavkami

- demand driven

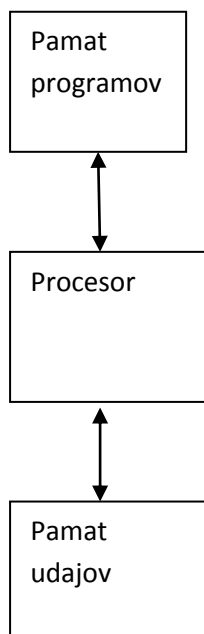


Rozdelenie podľa spôsobu pamatania programov a udajov

Princetonska architektura



Harvadská



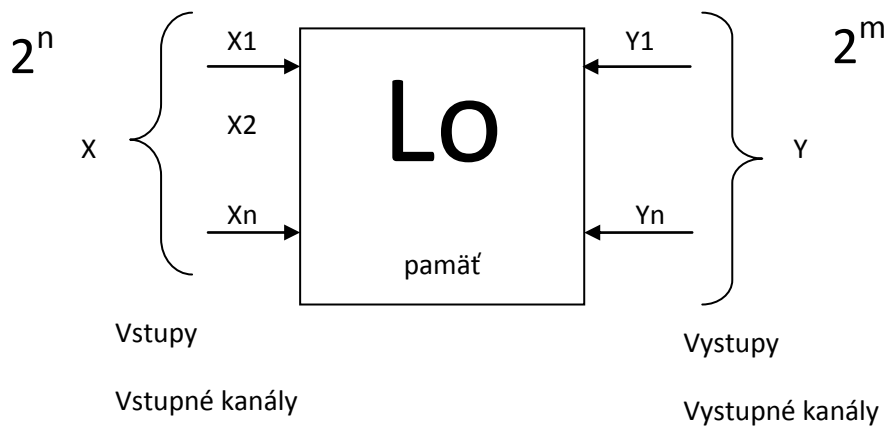
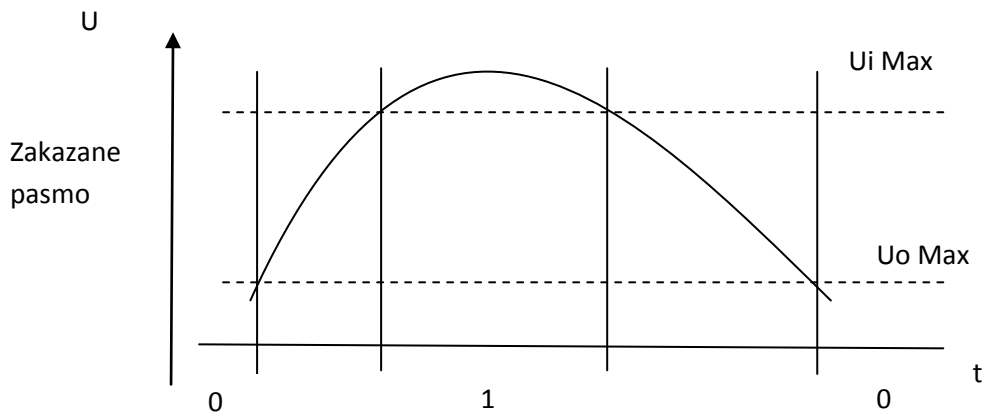
2 Logická úroveň a stavba Počítačových systémov

2.1 Logické obvody

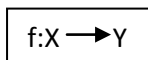
- dvojhodnotové premenné

= Log 0 –fals-low-L

=log 1 – true – high –H



Zobrazenie



$X = \{(000)(001)(010)(011)(100)(101)(110)(111)\}$

$Y = \{(00)(01)(10)(11)\}$

- vstupné slovo
- výstupné slovo
- Správanie obvodov
- Štruktúra obvodov

1. Analýza štruktúra =>správanie

Nejednoznačná

2. Systém –správanie + logické členy

-nejednoznačná

Kritériá optimálnosti

- Rýchlosť
- Cena

3. Simulácia

- Funkčná
- Časová

4. Diagnostika

- Detekcia
- Lokalizácia

Rozdelenie logických obvodov:

- **podľa reakcie**
 - a. kombinačné
 - b. sekundárne
- **podľa činnosti v čase**
 - a. asynchrónne
 - b. synchronné (synchronizačná premenná)
- **podľa spôsobu implementácie:**
 - a. pevná funkcia
 - b. programovateľná funkcia

2.1.1 Kombinované logické obvody

- asynchrónne
- diskretný dynamický systém

k obrazku hore ten s tými kuceravými zatvorkami

$$y_1 = f_1(x_1, x_2, \dots, x_u)$$

$$y_m = f_m(x_1, x_2, \dots, x_u)$$

počet všetkých logických funkcií = 2^{2^n}

neuplne definovaná logická funkcia $\sim \rightarrow X = (0, 1), 0, 1$

počet kombinácií vzrastie 3^{2^n}

$$f: \{0, 1\}^n \rightarrow \{0, 1\}$$

$$f: \{0, 1\}^n \rightarrow \{0, 1, x\}$$

2.1.1.1 Spôsoby zápisu B funkcií

Boole (boolovské premenné)

- Pravdivostná tabuľka

	X ₁	X ₂	X ₃	Y ₁	Y ₂
0	0	0	0	0	X
1	0	0	1	0	0
2	0	1	0	0	0
3	0	1	1	1	1
4	1	0	0	0	0
5	1	0	1	1	1
6	1	1	0	1	1
7	1	1	1	1	X

- Číselný zápis

$$„1“ \rightarrow y_1 = D(3,5,6,7)$$

$$„0“ \rightarrow y_1 = K(0,1,2,4)$$

$$„1“ \rightarrow y_2 = D(3,5,6(0,7))$$

$$„0“ \rightarrow y_2 = K(1,2,4(0,7))$$

- Vektorový zápis

$$y_1 = (00010111)$$

- mapový zápis

- mapa

- Karnaugh

- 2ⁿ štvorčekov
- Gray-ov cyklický kód

⁰ 0	²	³	¹ 0
⁴	⁶	⁷ X	⁵ X

y₁

⁰ 0	² 0	³ 1	¹ 0
⁴ 0	⁶ 1	⁷ 1	⁵ 1

$$y_1 = x_1 \cdot x_2 + x_1 \cdot x_3 + x_2 \cdot x_3$$

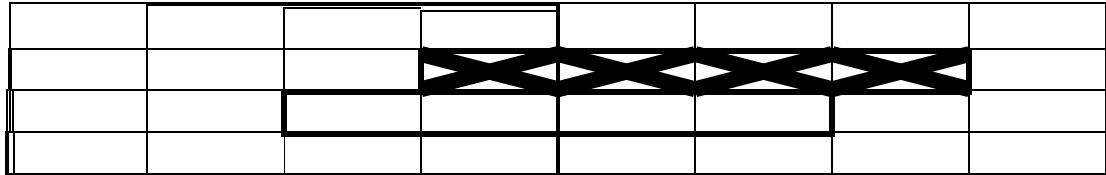
y₂

⁰ X	² 0	³ 1	¹ 0
⁴ 0	⁶ 1	⁷ X	⁵ 1

- Výraz

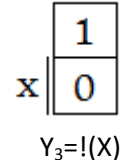
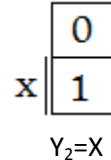
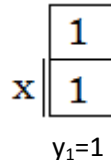
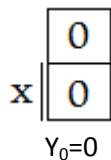
- Logický výraz

- Bulovský výraz[+,·,-]
- Vyjadruje kedy ma hodnotu 1!
- Všeobecne vyjadrenie susedov
- $y_1 = x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot x_2 \cdot x_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3$
- $y_2 = x_1 \cdot x_2 \cdot \bar{x}_3 + x_1 \cdot \bar{x}_2 \cdot x_3 + \bar{x}_1 \cdot x_2 \cdot x_3$
- 2^5 s-stupen konfiguracie



2.1.1.2 B funkcie s jednou Premennou

$$n = 1 \rightarrow 2^{2^n} \rightarrow 4$$



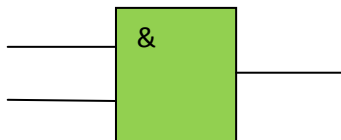
2.1.1.3 B funkcie s doma premennými

$$n = 2 \rightarrow 2^{2^n} \rightarrow 16$$

x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
		$f_0 = 0$			$f_3 = \bar{x}_1$		$f_5 = \bar{x}_2$					$f_{10} = x_2$		$f_{12} = x_1$			$f_{15} = 1$

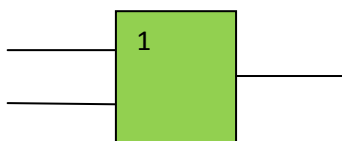
Logický súčin [And,A,I,AI]

$$f_8 = x_1 \cdot x_2$$



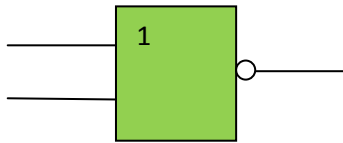
Logický súčet [Or,alebo]

$$f_{14} = x_1 + x_2$$



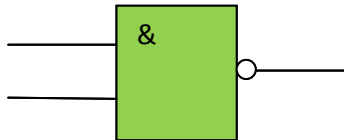
Negácia logického súčtu [Nor,ani] (Peircenova)

$$f_1 = \overline{x_1 \cdot x_2} = \overline{x_1 + x_2} = x_1 \downarrow x_2$$



Negácia logického súčinu [Nand](Seferova)

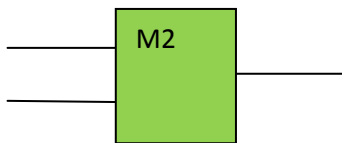
$$f_7 = \overline{x_1} + \overline{x_2} = \overline{x_1 \cdot x_2} = x_1 \uparrow x_2$$



Neekvivalentná (nerovnoznačná)[XOR]

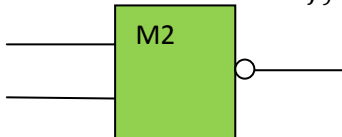
$$f_6 = x_1 \cdot \overline{x_2} + \overline{x_1} \cdot x_2 = x_1 + x_2$$

- Súčet Modulo 2
- Suma modulo 2
- Exclusive OR
- Len dve vstupné premenné



Ekvivalencia (rovnosť)[xnor]

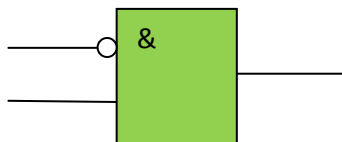
$$f_9 = x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} = x_1 \equiv x_2$$



Inhibícia

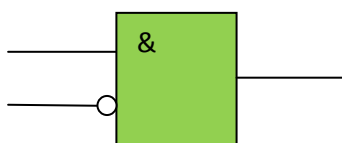
$$f_2 = \overline{x_1} \cdot x_2 \text{ -znabrana}$$

$$x_1 | \rightarrow x_2$$



$$f_4 = x_1 \cdot \overline{x_2} \text{ -znabrana}$$

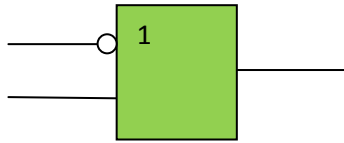
$$x_2 | \rightarrow x_1$$



Implikacia

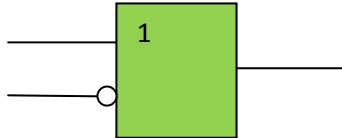
$$f_{11} = \overline{x_1} + x_2$$

$$x_1 \rightarrow x_2$$



$$f_{13} = x_1 + \overline{x_2}$$

$$x_2 \rightarrow x_1$$



$n > 2$

- Špeciálne funkcie
 - o Majoritná (n=5 Apollo aspoň 3 počítače museli mať rovnaký výsledok)
 - o Prahová (Prah)
 - o Symetrická

2.1.1.4 Boolovský výraz (b-výraz)

- Retazec
 - o premenné a ich negácie
 - o logické operátory
 - o zátvorky
- Logický výraz – $(a + \bar{b}) \rightarrow (x_1 \cdot \bar{x}_2 + x_3)$
- Booleanský výraz – $a + [b\bar{c}(\bar{a} + cd) + ce][+, \cdot, -]$
- Čo je všetko výraz
 1. Premenné
 2. Ak A je výraz tak aj \bar{A} je výraz
 3. Ak A a B sú výrazy tak aj A+B je výraz

2.1.1.5 Boolovská algebra

Def: Booleanský výraz je šestica

$$B = \{B^{(n)}, +, \cdot, -, 0, I\}$$

$+, \cdot$ bitové operátory

$B^{(n)}$ množina všetkých B-výrazov

$-$ Unarna operacia

0,1 – unárne operatory

Pre Booleansku algebru platia tieto ekvivalencie

1. $a + b = b + a$ komutatívnosť
 $a \cdot b = b \cdot a$
2. $a + (b + c) = (a + b) + c$ – asociatívnosť
 $a \cdot (b \cdot c) = (a \cdot b) \cdot c$
3. $a \cdot (b + c) = a \cdot b + a \cdot c$ – distributívnosť
 $a + b \cdot c = (a + b) \cdot (a + c)$!!!
4. $a \cdot a \cdot a \dots a = a$
 $a + a + a \dots a = a$
5. $\overline{a + b + c} = \bar{a} \cdot \bar{b} \cdot \bar{c}$ – demorganove pravidlá
 $\overline{a \cdot b \cdot c} = \bar{a} + \bar{b} + \bar{c}$
6. $\bar{\bar{a}} = a$ – pravidlo o zdvojennej negácii
7. $a + \bar{a} = 1$ – pravidlo o komplemente
 $a \cdot \bar{a} = 0$
8. $a + 1 = 1$ – agresívnosť
 $a + 0 = a$
9. $a \cdot 1 = a$
 $a \cdot 0 = 0$
10. $(a + b) \cdot (\bar{a} + b) = b$ – spojovanie !!!
 $(a \cdot b) + (\bar{a} \cdot b) = b$
11. $a + a \cdot b = a$ – pohltenie
 $a \cdot (a + b) = a$
12. $a + \bar{a} \cdot b = a + b$!!!
 $a \cdot (\bar{a} + b) = a \cdot b$

2.1.1.6 Normálne formy Boolovských výrazov

- elementárny súčin - $x^1 \cdot x^2 \cdot \bar{x}^3$
 - o rád – počet počet písmen 3
- Elementárny súčet - $x^1 + x^2 + \bar{x}^3$
 - o Rád – 3

1. Disjunktívna (súčtová) normálna forma (DNF)

$$f = \sum_i q_i$$

Logický súčet

2. Konjunktívna normálna forma (KNF)

$$f = \prod_j h_j$$

Logický súčin

A. Úplná DNF (ÚDNF)-ÚKNF

- Obsahuje súčiny (súčty) s $r = n$
- JEDINÁ

		X_2	X_3
		0	0
		1	0
X_1	0	1	1
	1	1	1

- ÚDNF
- $y_1 = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$
- ÚKNF
- $y_1 = (\bar{x}_1 + x_2 + x_3)(x_1 + \bar{x}_2 + x_3)(x_1 + x_2 + \bar{x}_3)(x_1 + x_2 + x_3)$

ÚDNF $\bar{f} \rightarrow \text{demorgan}$

B. Skrátene formy Booleanských výrazov SDNF SKNF

- JEDINÁ
- Aplikácia pravidiel spojovania a pohltenia (všetkých možností) na ÚDNF ÚKNF
- Vynímanie pred zatvorku z ÚDNF a ÚKND vzorcov
- SDNF: $y_1 = x_1 x_2 + x_2 x_3 + x_1 x_3$
- SKNF: $y_1 = x_1 + x_3)(x_2 + x_3)(x_1 + x_2)$

C. Iredundantná DNF (IDNF) -IKNF

- Nedá sa redukovať počas písania
- Viacero IDNF (IKDF)

D. Minimálna DNF (MDNF) MKDF

- Najmenší počet písmen zo všetkých
- viacero

Príklad:

		X_2	X_3
		1	0
		1	1
X_1	1	1	1
	1	1	0

ÚDNF: $g = \bar{x}_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 x_3 + x_1 \bar{x}_2 \bar{x}_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 x_3$

18

SDNF: $g = \bar{x}_2 \bar{x}_3 + x_1 x_2 + \bar{x}_1 x_3 + x_2 x_3 + \bar{x}_1 \bar{x}_2 + x_1 \bar{x}_3$

12

IDNF: $g = \bar{x}_2 \bar{x}_3 + x_1 x_2 + \bar{x}_1 x_3$

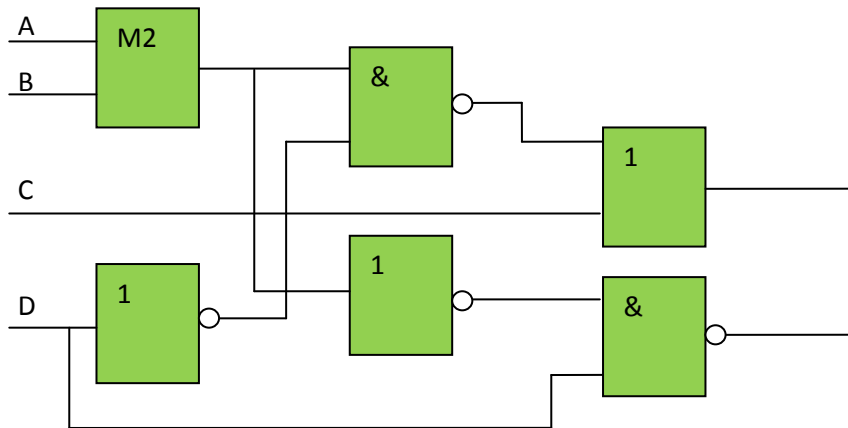
6

$g = x_2 x_3 + \bar{x}_1 \bar{x}_2 + x_1 \bar{x}_3$

6

2.1.2 Analýza logických obvodov

- správanie + štruktúra logických členov
 - o FUNKCIE Mapy výraz...
- JEDNOZNAČNÉ!!!



2.1.3 Syntéza logických kombinačných obvodov

- Správanie sa + súbor typov logických členov (kritéria optimálnosti - (rýchlosť, cena, diagnostikovateľnosť))= **štruktúra**
- Nejednoznačnosť (možnosť vytvorenia veľkého množstva)

Postup:

- hľadá sa skupina výrazov, ktoré zodpovedajú danej skupine B-funkcií $m > 1$
- DNF(disjunktívne normálové formy) alebo KNF
- ↓ MDNF !!
- Algoritmizovateľné

2.1.3.1 vyjadrovania IDNF (IKNF) z mapy

- Implikant funkcie → súčin (ma hodnotu 1 tam kde ma aj funkcia hodnotu 1)
- Prostý implikant → súčin

		X_1	X_2
		0	1
X_3	1	1	0

$\overline{x_1 x_2 x_3}$ – nie je prostý implikant
 $\overline{x_2 x_3}$ – je prostý implikant

- Implikant funkcie → súčet
- Protý implikant → súčet
- SDNF-Všetky PI
- SKNF všetky PI

IDNF ← MDF

Pravidelná konfigurácia

1. Obsahuje 2^s štvorčekov s – stupeň konfigurácie
2. Každý štvorček v konfigurácii musí mať práve \leq susedných štvorčekov

Pokritie

- Najmenej
- Najväčšie
- Každý bod pokryje aspoň jedenkrát

		X_3	X_4
		0	1
X_1	1	0	x
X_2	x	1	1
	0	1	0

Príklad:

MDNF:

$$f = \overline{x_1}x_4 + x_1x_3 + x_2\overline{x_3}x_4 \quad (7)$$

$$f = \overline{x_1}x_4 + x_1x_3 + x_2\overline{x_1}x_3 \quad (7)$$

		X_3	X_4	
		0	0	1
X_1		1	0	1
		x	1	1
X_2		0	1	1

MKNF:

$$f = (\overline{x_1} + x_3)(x_1 + \overline{x_3} + x_4)(x_1 + x_2 + x_4) \quad (8)$$

$$f = (\overline{x_1} + x_3)(x_1 + \overline{x_3} + x_4)(x_2 + x_3 + x_4) \quad (8)$$

Hľadanie minimálnej skupiny DNF:

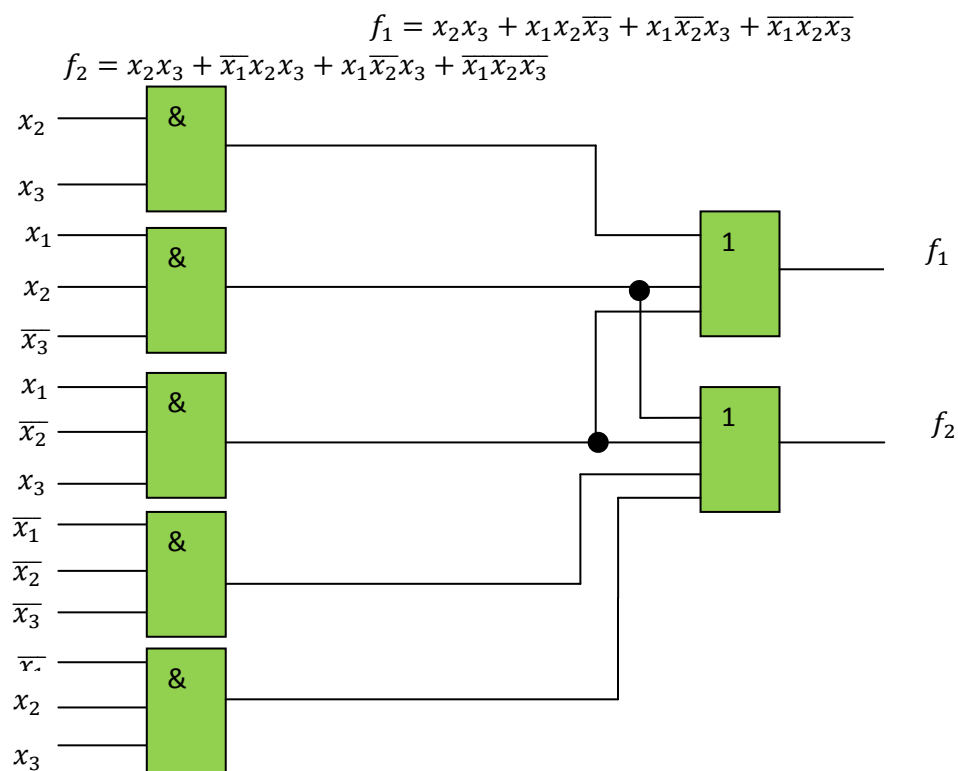
- Minimálny počet rôznych súčinov

		X_2	X_3	
	0	0	1	0
X_1	0	1	1	1

f_1

	X_2		X_3	
	1	0	1	0
X_1	0	1	0	1

f_2



2.1.3.2 Zmiešané normálové formy

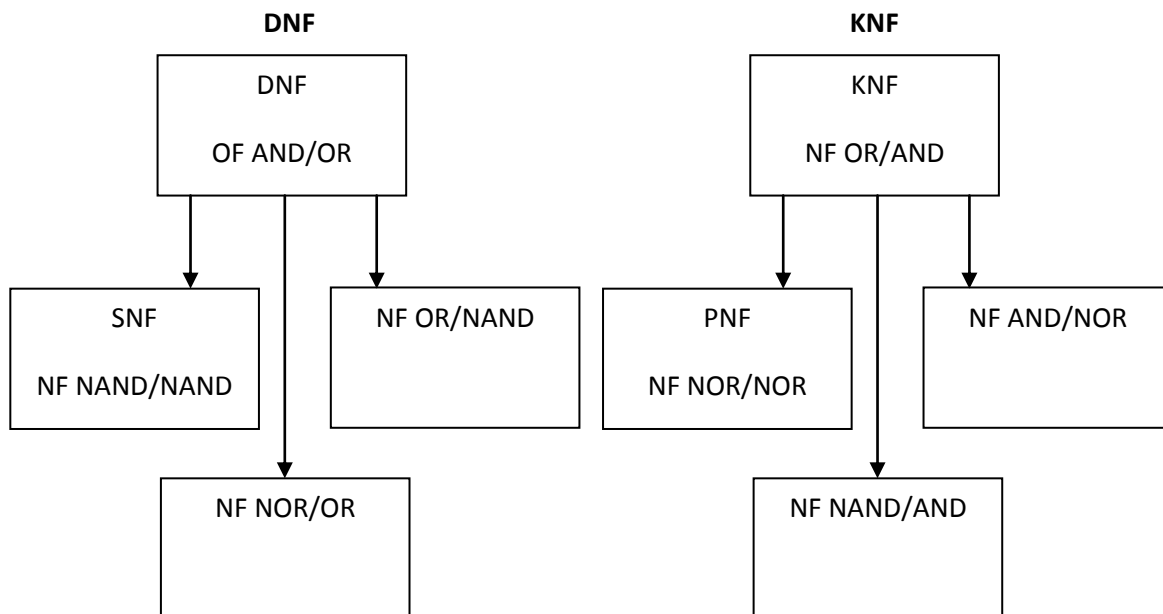
+ · − ↓ ↑

Or and not NOR NAND

Dvojstupnové obvody (najrýchlejšie)

= negácia vstupných premenných

= neohraničený počet vstupov logických členov !!!



Príklad:

Poznámka: $\bar{A} = \downarrow A = \uparrow A$

$$DNF: y = a\bar{b} + \bar{c}d + e \rightarrow NF\ AND/OR$$

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

$$\overline{a\bar{b} + \bar{c}d + e} = \overline{a\bar{b}} \cdot \overline{\bar{c}d} \cdot \bar{e}$$

NF NOR/OR:

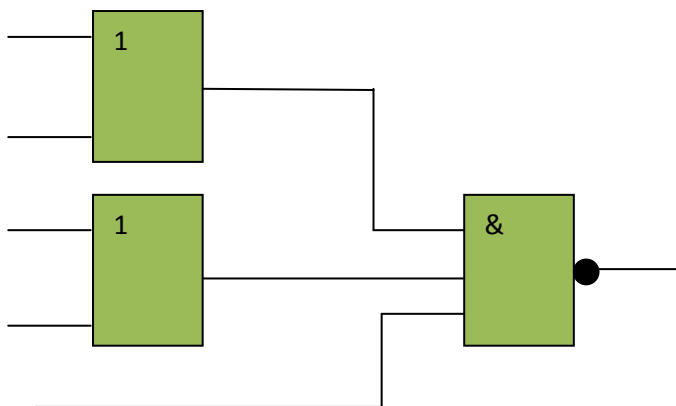
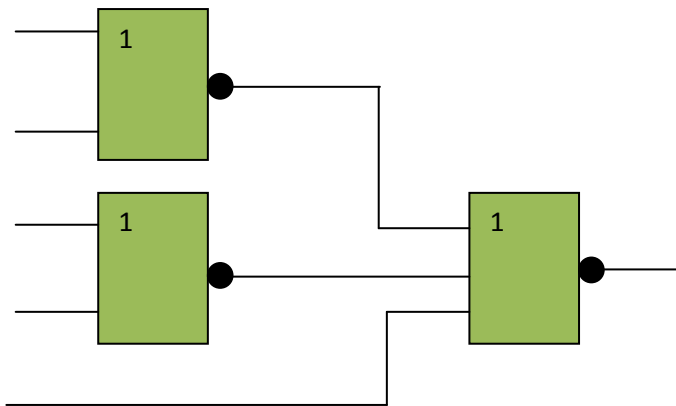
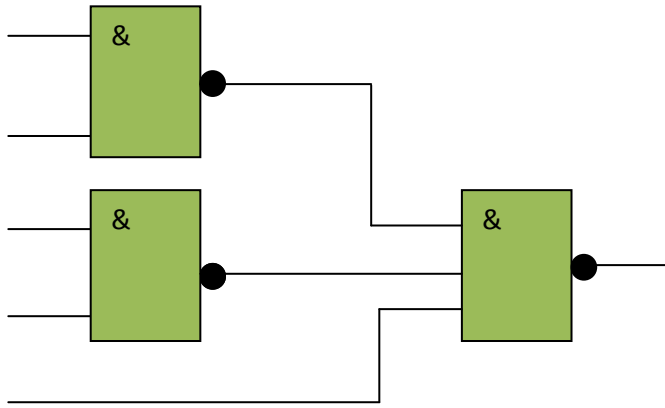
Poznámka: $\overline{A + B} = A \downarrow B = \bar{A} \cdot \bar{B}$

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

NF OR/NAND:

Poznámka: $A \cdot B = A \uparrow B = \bar{A} + \bar{B}$

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



KNF: $y = (\bar{a} + \bar{b})(c + \bar{d})e$ NF OR/AND

DNF: $y = (\bar{a} \downarrow \bar{b}) \downarrow (c \downarrow \bar{d}) \downarrow \bar{e}$

ND NAND/AND: $y = (a \downarrow b) \downarrow (\bar{c} \downarrow d)e$

NF AND/NOR: $y = \overline{(a \cdot b)} \downarrow \overline{(\bar{c} \cdot d)} \downarrow \bar{e}$

2.1.4 Sekvenčné logické obvody

- Synchronne (h)
- Asynchronne
- S pevnou funkciou
- S programovateľnou funkciou

➤ STAV ← história

Správanie

Konečný →
Determinovaný
Automat

2.1.4.1 Konečný automat

Def: Päťica(šestica)

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

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

X – množina vstupov

Y – množina výstupov

S – množina stavov

p – prechod funkcie

$$p: S \times X \rightarrow S$$

$$S = (S_0, S_1)$$

$$X = (X_0, X_1)$$

$$S \times X (\{S_0, X_0\} \{S_0, X_1\} \{S_1, X_0\} \{S_1, X_1\})$$

v – výstupná funkcia

Moore $v: S \rightarrow Y$

Mealy $v: X \times S \rightarrow Y$

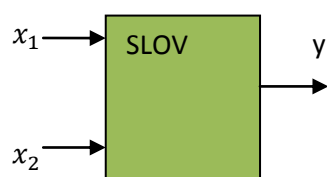
s_0 – výstupná funkcia

2.1.4.2 Spôsoby zápisu automatov

1. Prechová tabuľka

Príklad 1:

	<u>X₁</u>		<u>X₂</u>			<u>X₁</u>		<u>X₂</u>	
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	



$$S = \{1,2,3\}$$

$$X = \{00,01,10,11\}$$

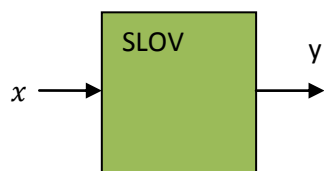
$$Y = \{0,1\}$$

$$p: S \times X \rightarrow S$$

$$v: S \times X \rightarrow S$$

Príklad 2: (Moore) – pomalší o krok

	\overline{x}		y
1	1	2	0
2	3	2	1
3	3	4	1
4	1	4	0



$$p: S \times X \rightarrow S$$

$$v: S \rightarrow Y$$

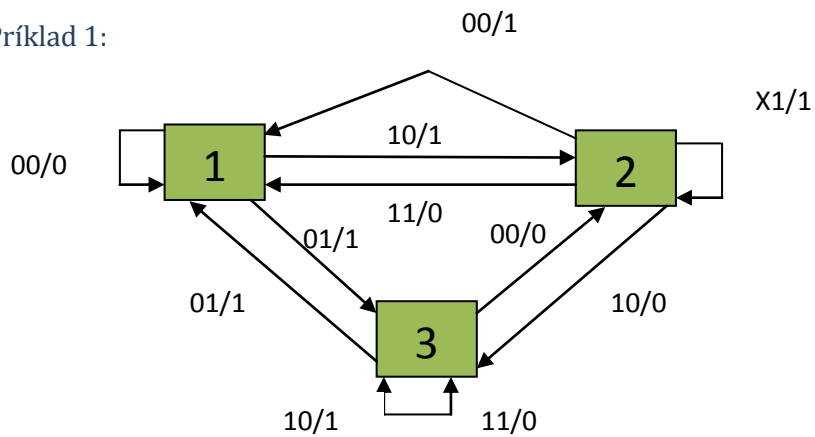
$$S = \{1,2,3,4\}$$

$$X = \{0,1\}$$

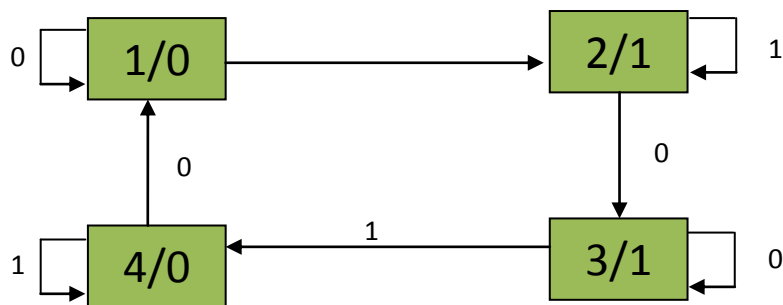
$$Y = \{0,1\}$$

2. Prechodové grafy

Príklad 1:



Príklad 2.:

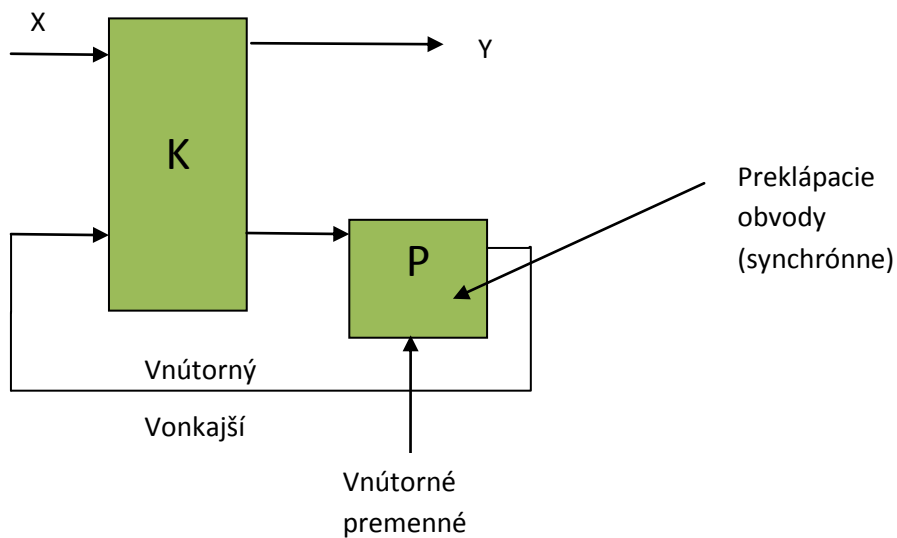


2.1.5 štruktúra syntéza synchronných sekvenčných obvodov

Model SSO:

K- kombinačná časť SSO

P - pamäťová časť SSO



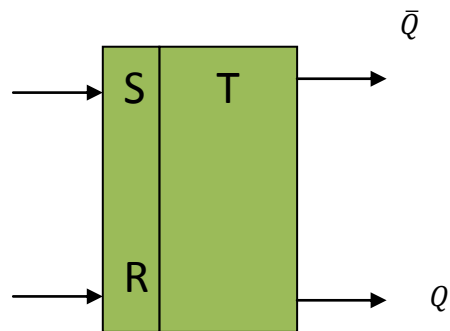
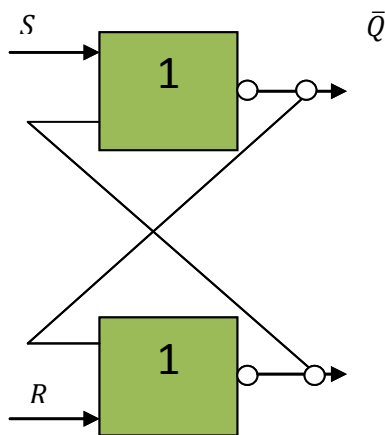
- Minimálny počet Prekladacích obvodov v „P“
- Minimálny počet logických členov v „K“
- Maximálna operačná rýchlosť

2.1.5.1 Synchronne preklápacie obvody

-základ → asynchróny PO-SR

S - set

R - reset



	S	R	Q
0	0	1	x
1	1	1	x

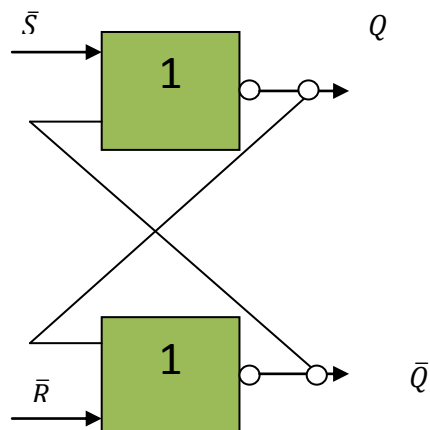
	S	R	Q
P	1	x	0

↑
pamätové

funkcia správania sa

$$S \cdot R \stackrel{!}{=} Q$$

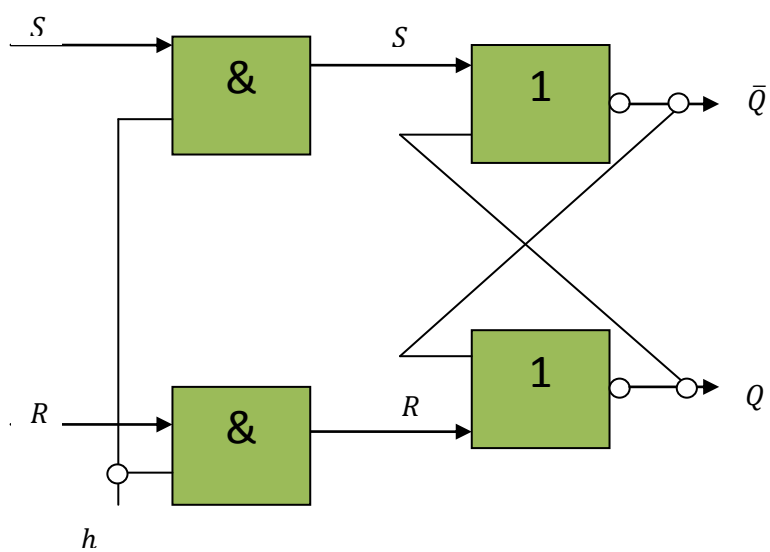
PO – $\bar{S}\bar{R}$



	\bar{S}	\bar{R}
x	0	P

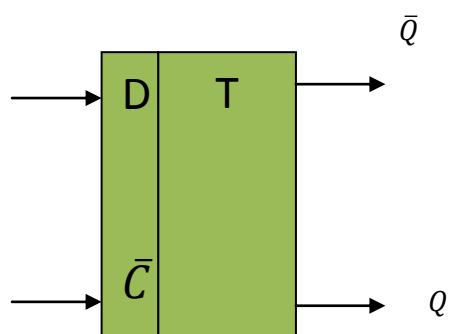
2.1.4.2 Spôsob zápisu automatov

Synchrónny PO-SR



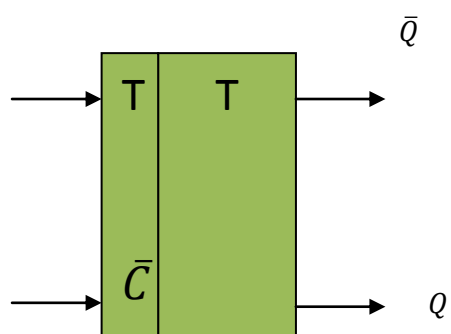
synchrónny PO-D

D	
0	1

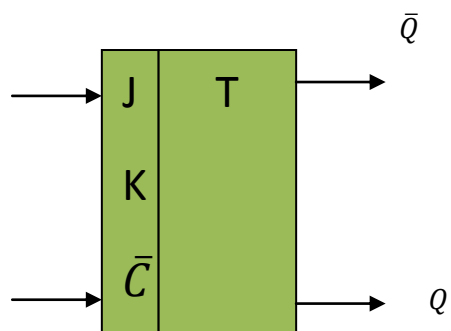


Synchrónny PO-T

T	
P	K



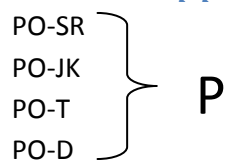
Synchrónny PO-JK



	J	K
P	1	0

Univerzálny

2.1.5.2 Postup pri syntéze SSO



A- Abstraktná syntéza!!!

- Slovný opis správania sa

↓ ← Stavy

AUTOMAT

B- Štruktúrny syntéza

Automat +po+ Logické členy

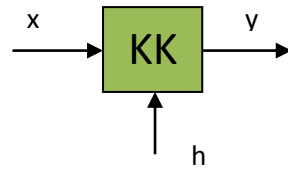
↓

Štruktúra K

2.1.5.3 Príklad

Kontrolór kódu

PO-JK, NAND



Legálne 0,1,2,3,4,5

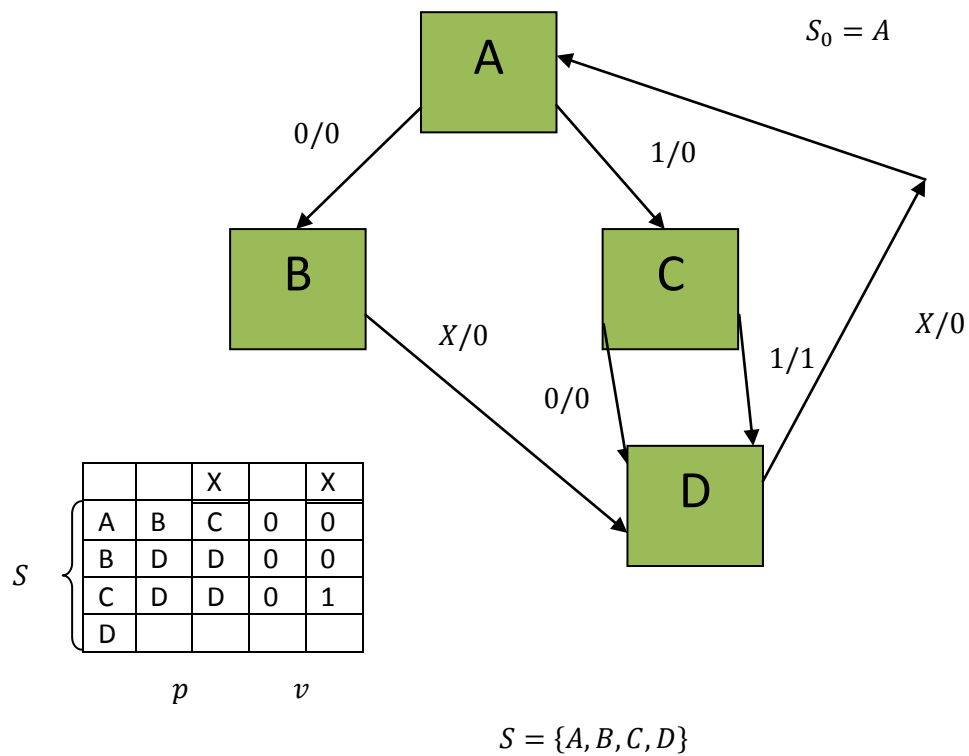
Nelegálne 6,4

	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	

A. Abstraktný systém

- Definovanie stavov

- graf \leftarrow Rozhodovací automat



B. Štruktúra syntézy

a) Kódovanie stavov

- kódovanie nutných (stavových) premených (=PO)

$$|s| \leq 2^{kmin}$$

$$kmin = ?$$

$$|s| = 4 \rightarrow kmin \stackrel{!}{=} 2$$

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

- Kódovacia mapa:

	$\overline{Z_1}$	
	A	B
Z_2	C	D

A ... 00

B ... 10

C ... 11

D ... 01

Dvojitý ekvivalent:

KK_B

		X		X
00	10	11	0	0
01	01	01	0	0
11	01	01	0	1
01	00	00	0	0

Mapy budiacich funkcií

k - map

$n + k$ – premenných

	\overline{x}			\overline{x}	
	1	1		0	1
Z_1	0	0		0	0
Z_2	0	0		1	1
	0	0		1	1
	Z_1			Z_2	
	D_1			D_2	
				... A	
				... D	
				... C	
				... B	

Mapy výstupných funkcií F

k - map

$n + k$ – premenných

		\overline{x}	
Z_1	Z_2	0	1
		0	0
		0	0
		0	1
		0	0

$$y = z_1 Z_2 x$$

Mapy budiacich funkcií

		\overline{x}	
Z_1	Z_2	1	1
		0	0
		0	0
		0	0
		Z_1	

		<u>X</u>	
Z_1 Z_2		0	1
	0	0	0
	1	1	1
	1	1	1
		Z_2	

		<u>x</u>	
Z_1	Z_2	0	0
		0	0
		0	1
		0	0
		<u>y</u>	

$$Z_1 = D_1 =$$

$$Z_2 = D_2 =$$

$$y =$$

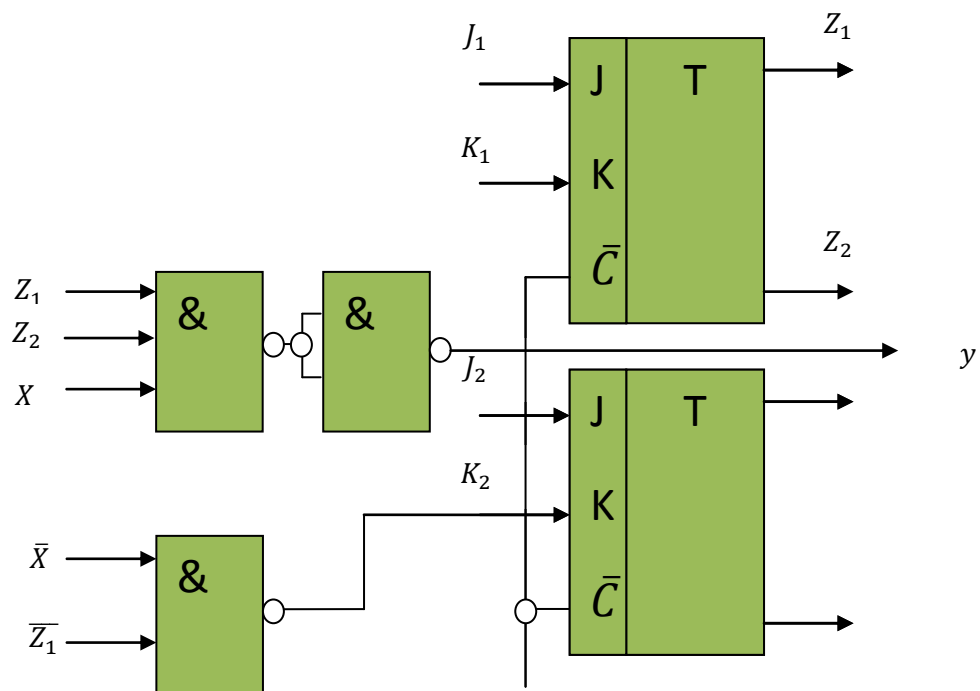
PO-JK

Mapy budiacich funkcií PO:

Inverzná prechodová funkcia PO

$2k$ máp $n + k$ premenných

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



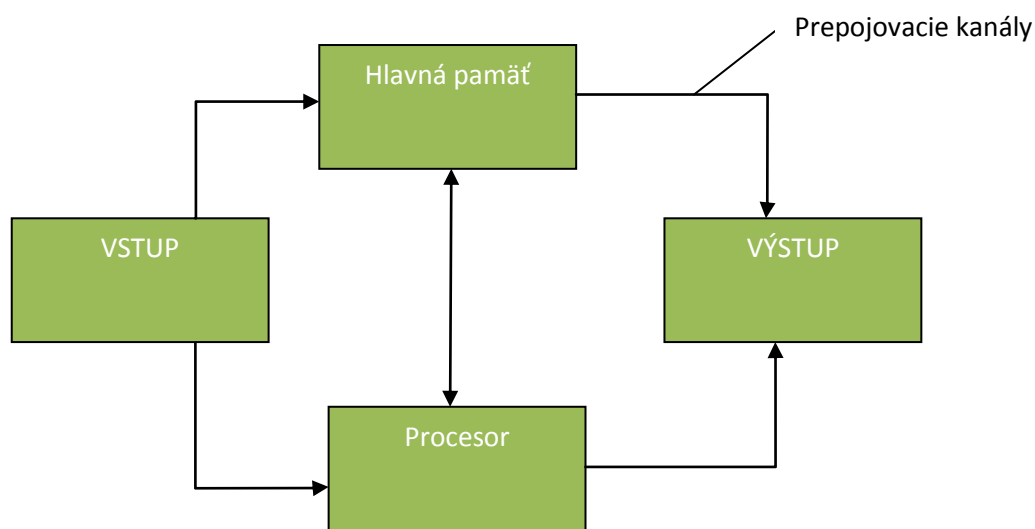
3. Hlavné podsystemy digitálnych počítačov

- Von Neuman – SISD
 - o Sériové – riadené tokom inštrukcií
 - o Jednoprocesorové počítače

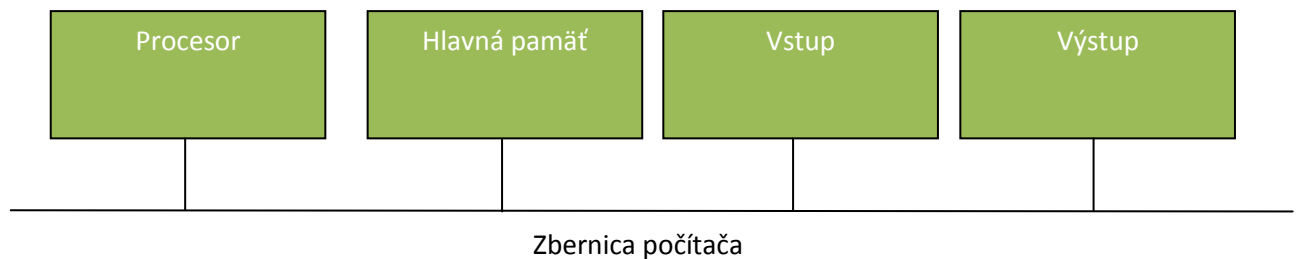
3.1. Prepojovací systém počítača

- Prepojenie jednotlivých častí počítača
 - o čítanie inštrukcií z pamäti
 - o čítanie údajov z pamäti
 - o zápis do pamätí
 - o ...

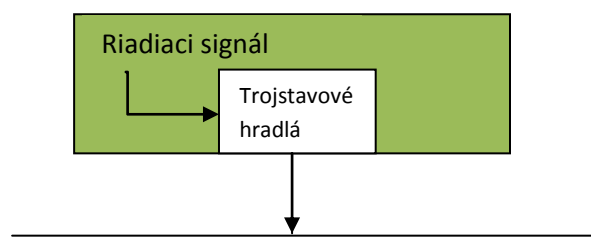
Prepojovacie kanály



Zbernica



- Nevýhoda naraz môžu komunikovať iba dvaja. Zbernicu dostáva ten s vyššou prioritou
- Zariadenie musí byť pripravené na odpojenie zo zbernice
- Vysielač – prijímač (master slave)



3.1.1. Rozdelenia zberníc

- Rozdelenie podľa spôsobu riadenia

A. Single – master

B. Multi-master

⇒ V každom čase zbernicu riadi iba jedno zariadenie

- Podľa synchronizácie prenosu

A. Synchronizované zbernice

⇒ Spoločný jeden zdroj synchronných impulzov

⇒ Zariadenie s rovnakou rýchlosťou

B. Asynchrónne zbernice

⇒ Zariadenia s nerovnakou rýchlosťou

⇒ Čaká sa na prenos (ďalší údaj sa posiela až po dostaní správy o prijatí predošlého)

- Rozdelenie podľa časového multiplexu

A. Multiplexové zbernice

Význam, druh prenášanej informácie sa mení s časom

Údaj

Adresa

Riadiaca

B. Nemultiplexované zbernice

- Rozdelenie podľa tvaru prenášaných údajov

A. Paralelné zbernice

⇒ Naraz sa prenáša viacbitový výraz

⇒ Viac vodičov pravdepodobnosť poruchy

⇒ Vysoká rýchlosť

B. Sériové zbernice

⇒ Bit po bite

⇒ Nižšia rýchlosť

⇒ Vyššia spoľahlivosť

C. Sériovo paralelné zbernice

⇒ Posiela viacero malých sérií, ktoré sa na konci spájajú

3.1.2. Štruktúra typickej počítačovej zbernice

- Charakteristika:
 - Paralelná
 - Asynchrónna
 - Nemultiplexovaná
- Sekvencie
 - adresová
 - údajová
 - riadiaca

Adresová – adresy

- Nadriadený generuje adresy
 - procesor
 - DMA
- Pamäťová bunka
- Vstupno-výstupné zariadenia

Údajová

- Inštrukcie (HP→P)
- Údaje (HP→P; P→HP)

Riadiaca

- Povely – nadriadený
 - (čítanie zápis)
- Žiadosti – podriadený
 - (pridelenie zbernice)

Signálové sledy

- Časové priebehy
 - Dodržať
 - P – HP
 - P – VV

3.2. Základné koncepcie procesora

- Procesor
 - o základná časť počítača (SISD)
 - o Hlavná riadiaca autorita
- Inštrukcia
 - o Kód operácie
 - o adresa operácie

KO	A
----	---

HP- Hlavná Pamäť

- program – postupnosť inštrukcií
- výber inštrukcie z HP [PC] P-rogram C-ounter (programové počítadlo)
- prenos informácie medzi podradenými časťami počítača

Podľa vykonávanej funkcie

- univerzálne procesory
- problémovo orientované procesory

UP

- číselné a nečíselné operácie
- riadenie ostatných častí počítača
- to tvorí program HP

inštrukčný súbor – „ÚPLNÝ“ – „BOHATÝ“

- CISC
- RISC
- NISC – bez inštrukcií riadiace slová NOVÉ

Dôležité činitele

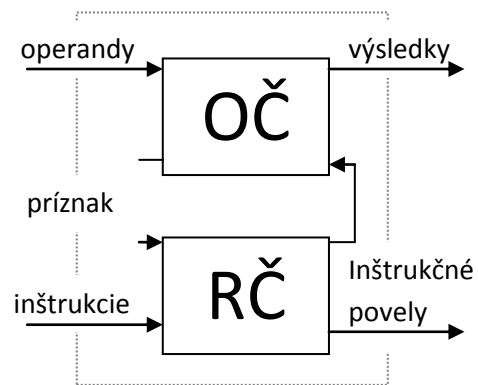
- efektívnosť
- rýchlosť

POP:

- Vykonávajú špeciálne funkcie
- Programové prostriedky
- Vstupno-výstupné procesory
- Numerické koprocessory (386, 486)
- Grafické procesory
- Logické procesory

3.2.1. Hlavné časti procesora

- operačná časť – operácie s operandmi
- riadiaca časť
 - o vyber inštrukcie
 - o dekodovanie inštrukcie
 - o zabezpečenie vzkonania RČ



3.2.1.1. Operačná časť procesora

a. Aritmeticko-logická jednotka (ALJ, ALU)

- Aritmetické
 - Logické
 - Iné
- } operandmi

Sčítanie

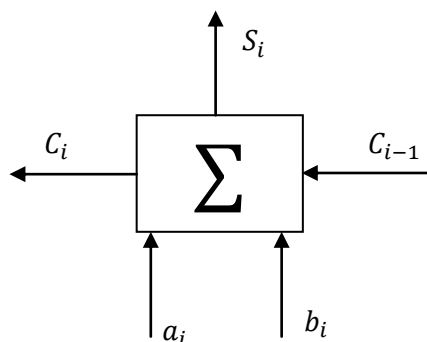
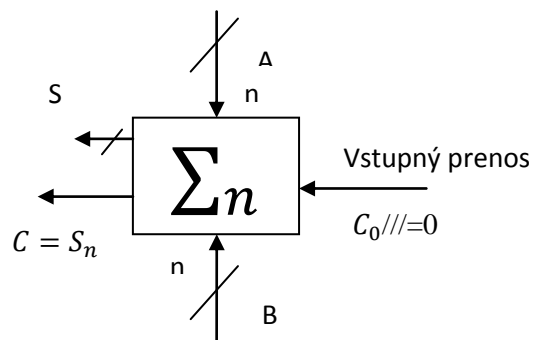
b. Registre zápisníková pamäť

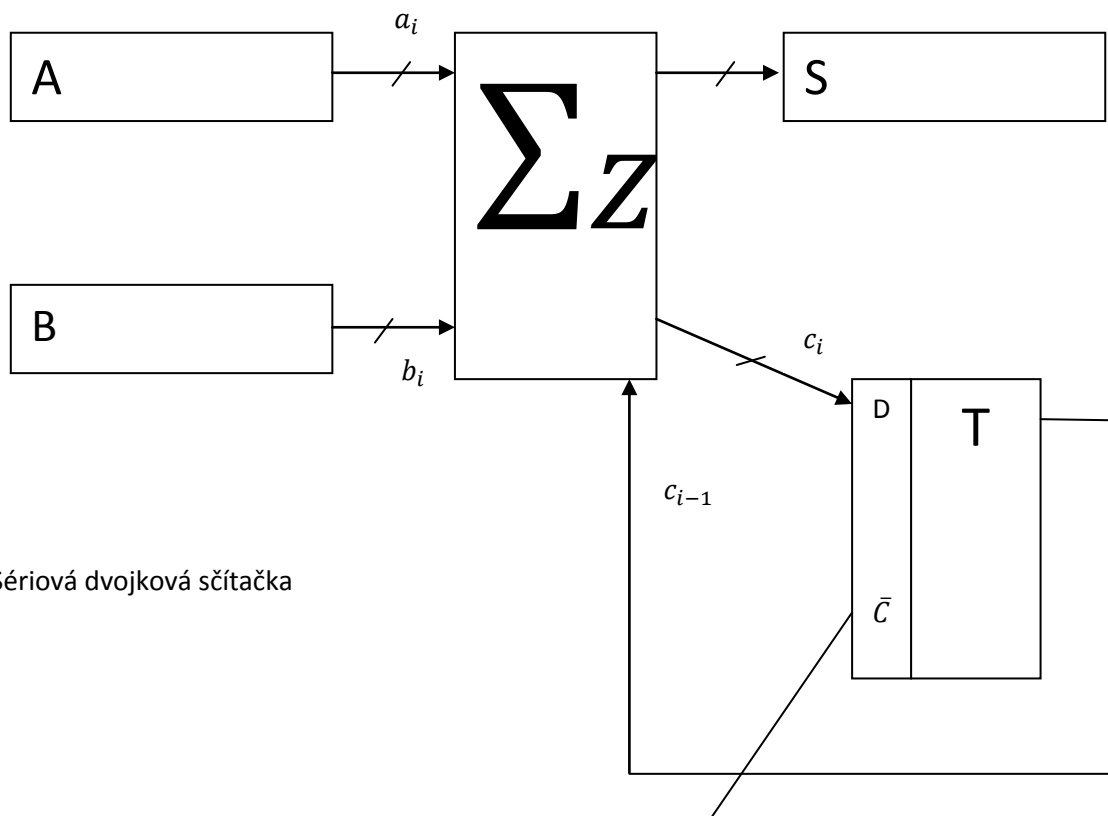
c. Komunikačné obvody

⇒ Medzi registrové obvody

3.2.1.1.1. Aritmeticko-logická jednotka

- Operácia sčítanie
 - Operácia odčítanie
 - Operácia násobenie
 - Operácia delenie
- } Paralelná dvojková
ščítačka !!!

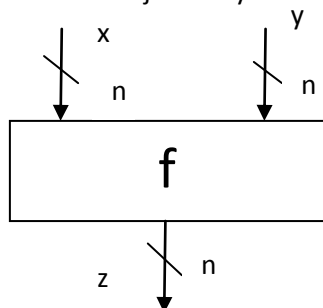




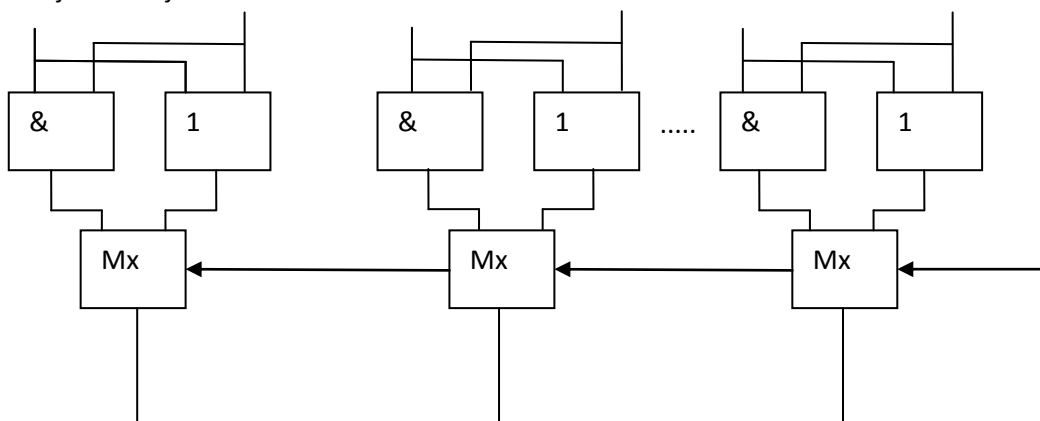
Sériová dvojková sčítačka

Logické operácie

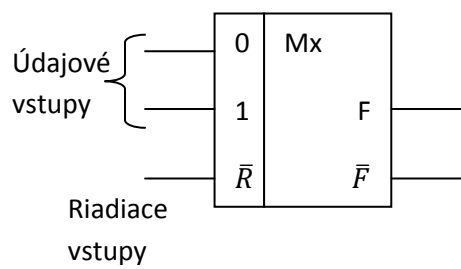
- Funkčné jednotky



- Dvojfunkčná jednotka

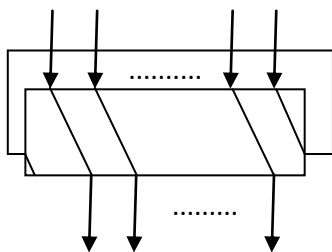


Multiplexor

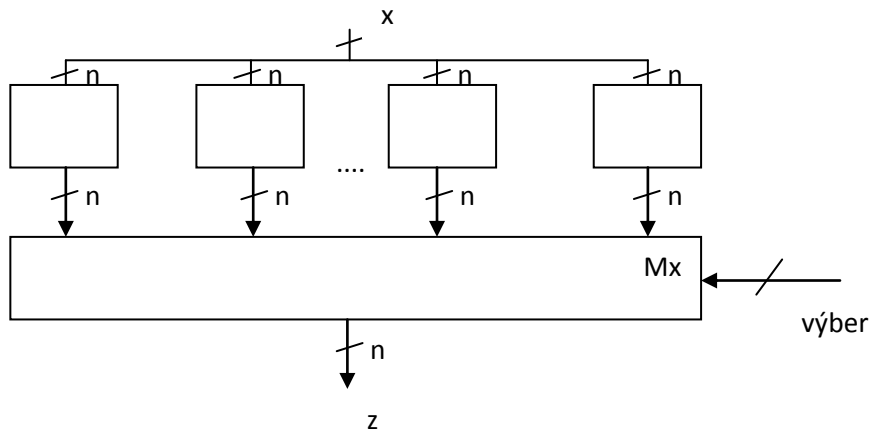


Postupy

- Aritmetické operácie
- Násobenie a delenie 2^n
- Posúvacie obvody
 - logický posun
 - Vľavo
 - vpravo



- kruhový posun
 - Vľavo
 - Vpravo
- aritmetický posun
 - Vľavo
 - Vpravo
 - Okrem znamienka



Predikáty(príkazy)

- Dvojhodnotové funkcie
 - o logické kombinačné obvody
 - Nad jedným operandom
 - $[=0;>0;<0; \dots]$

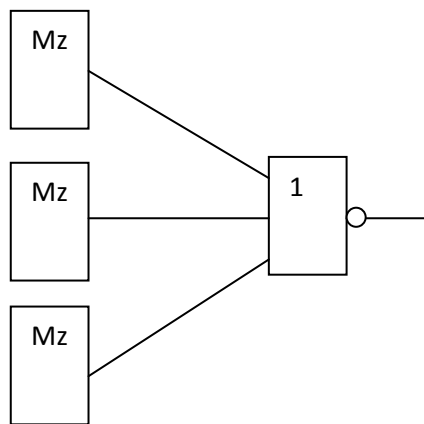
Príklad:

$$\text{„=0“ } P(x) = 1 \leftrightarrow x = 0$$

- Nad dvomi operandmi
 - $[x>y;x<y;x\leq y;x=y;\dots]$

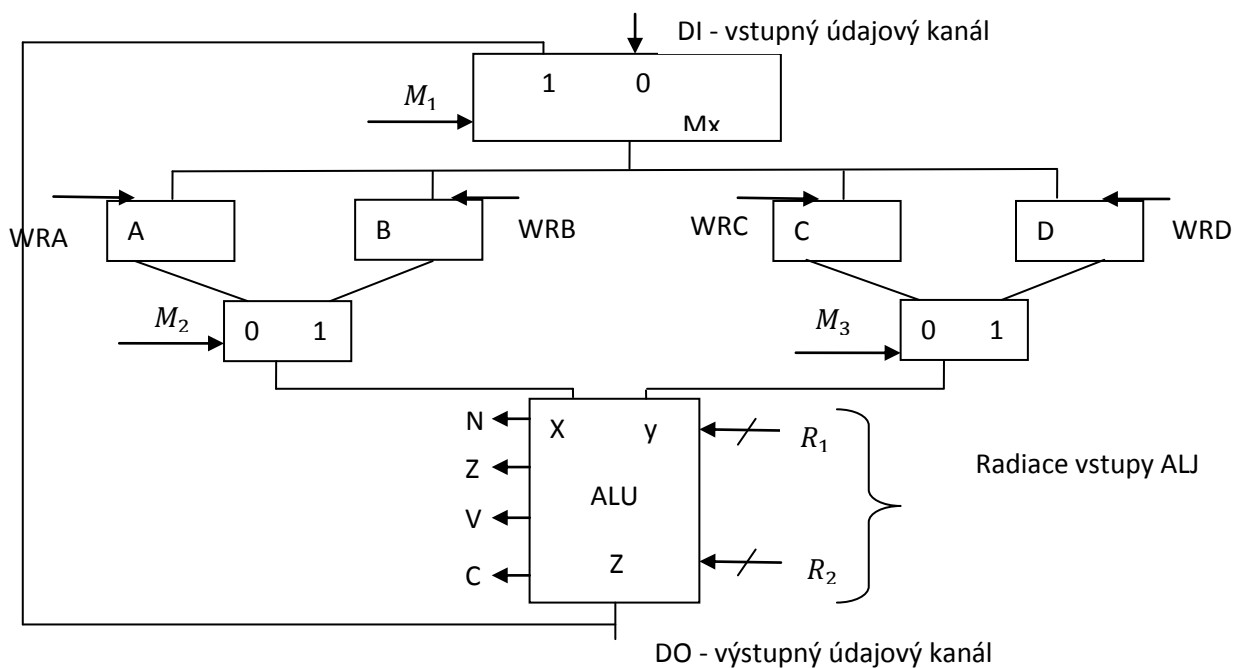
Príklad:

$$P(x, y) = 1 \leftrightarrow x = y$$



Prepojovacie obvody

- Spájajú štrukturálne prvky operačnej časti
 - o medzi registrové prenosy
 - a. Multiplexory a demultiplexory



b. Zbernicové

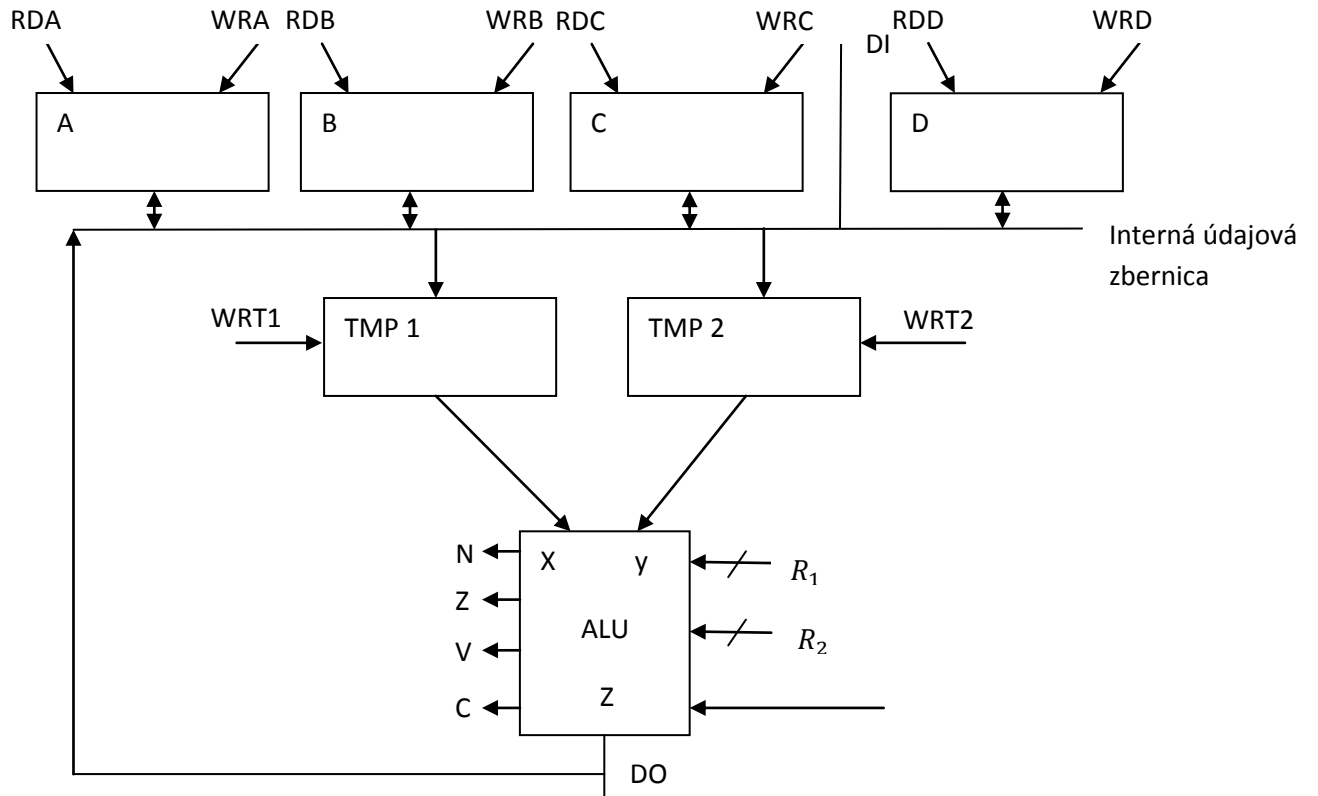
⇒ Vysielač

i. Prijímač 1

ii. Prijímač 2

iii. ...

⇒ Trojstravový radlo



3.2.1.2. Riadiaca časť procesora

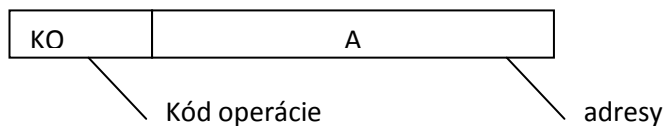
- Uskutočňuje
 - o **výber** inštrukcie → IR
 - o **dekódovanie** inštrukcie
 - o **vykonávanie** inštrukcie (operácia)
 - o *Riadenie okolia*

Inštrukčný cyklus

- 6 fáz
 1. Výber inštrukcie z pamäte – **IF** (instruction fetch)
 2. Dekódovanie inštrukcie – **D** (decode)
 3. Výpočet adresy operandu – **OA** (operand address)
 4. Výber operandov (z pamäte, ...) – **OF** (operand fetch)
 5. Vykonanie operácie – **EX** (execution)
 6. Zápis výsledku (do pamäte, do vstupného zariadenia) – **S** (store)

3.2.1.2.1. Formát inštrukcie

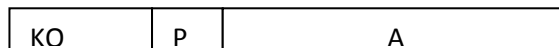
- Inštrukcia → Príkaz pre procesor



Počet adries:	nula adresové	<table><tr><td>KO</td></tr></table>			KO		
KO							
	Jedno adresové	<table><tr><td>KO</td></tr></table>	KO	<table><tr><td>A</td></tr></table>		A	
KO							
A							
	Dvoj adresové	<table><tr><td>KO</td></tr></table>	KO	<table><tr><td>A1</td></tr></table>	A1	<table><tr><td>A2</td></tr></table>	A2
KO							
A1							
A2							

Inštrukčný cyklus

- Procesory
 - o 1 formát (rovnaká dĺžka)
 - o viac formátové



3.2.1.2.2. Typy inštrukcií

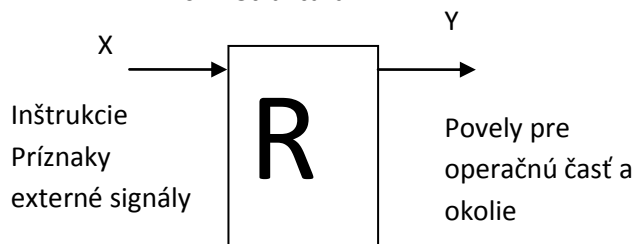
- Presunové inštrukcie
 - presun údajov
 - Register → register
 - Register → memory
 - Memory → register
 - Register → InOut
 - InOut → register
 - Memory → memory
 - InOut → memory
 - Memory → InOut
- Výpočtové inštrukcie
 - aritmetické operácie
 - logické operácie
 - iné
- Skokové inštrukcie
 - podmienené
 - nepodmienené
 - skok do podprogramu
 - návrat z podprogramu
 - Zásobník → LIFO (last in first out) (FIFO opačne)
- Riadiace inštrukcie

3.2.1.2.3. Spôsoby adresovanie operandov

	Adresovanie	V inštrukcií	V registri	V pamäti
implicitné	Registrové		operand	
	Zásobníkové		Adresa	Operand
	Nepriame		Adresa	Operand
explicitné	Priame	Adresa		Operand
	Nepriame	Adresa 1		Adresa 2
	bezprostredné	Operand		
	Registrové	Adresa registra	operand	
	Nepriame registrové	Adresa registra	Adresa	Operand
	Indexové	Adresa index. registra	Index	operand
	Bázovo-registrové	Bázový register Indexový register	Bázová adresa index	operand

3.2.1.2.4. Riadiaca časť pevnou funkciou

- Synchronný sekvenčný obvod
 - o Štruktúra



3.2.1.2.5. Mikroprogramovateľná riadiaca jednotka

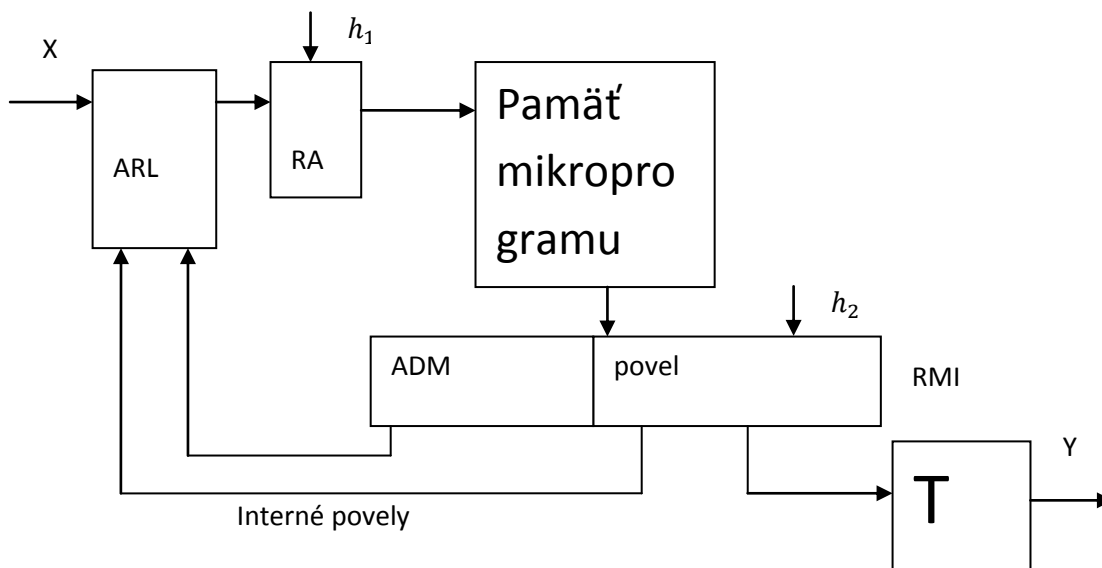
- synchronný systém – vykonáva mikroprogram

Špecializovaný procesor



Postupnosť mikroinštrukcií

- zmeny mikroprogramu
 - o mikroprogramové emulácie



RA - register adresy mikroprogramu

RMI – register m inštrukcií

ARL – adresovacia a riadiaca logika

T – transformačná časť

Prízny	Adresa	Y_I	Y
--------	--------	-------	---

Prízny - Kód logickej podmienky

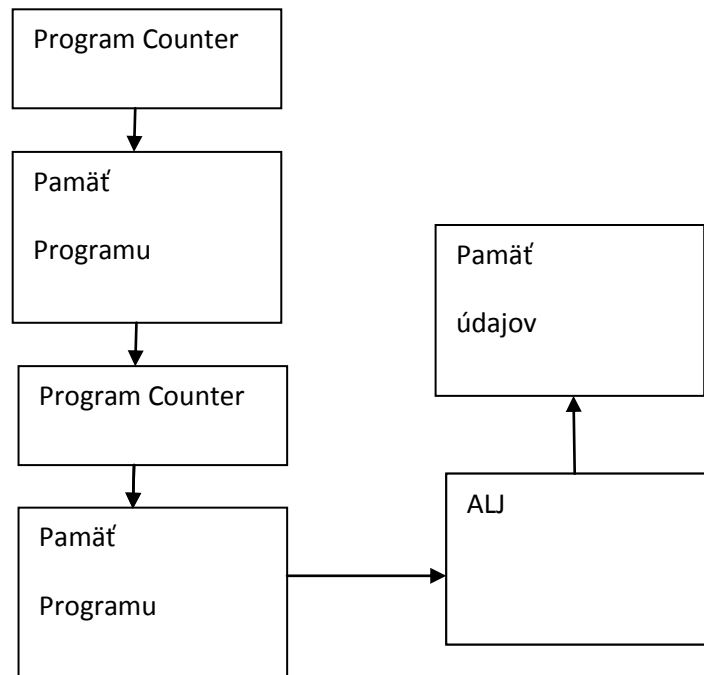
príznak	Y
---------	---

Príznak	Adresa 1	Adresa 2	Y_I	Y
---------	----------	----------	-------	---

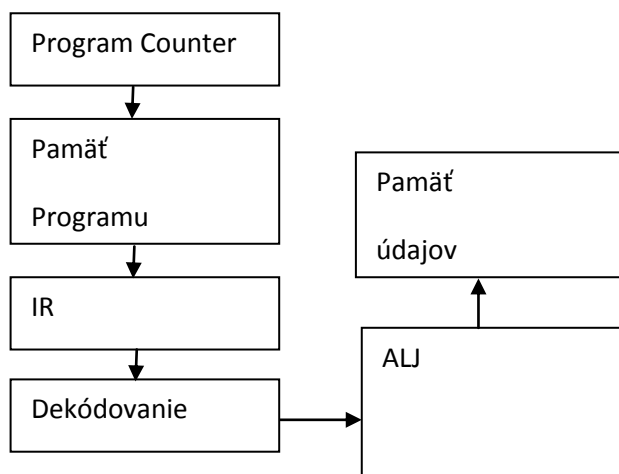
Príznak	H	DAdresa 1	DAdresa 2	Y_I	Y
---------	---	-----------	-----------	-------	---

3.2.2. CISC, RISC a NISC procesory

- Klasifikácia podľa inštrukčných súborov
 - o Procesory CISC (complex instruction set computer)
 - 1970 – 1980 – drahá a pomalá pamäť
 - Výkonné inštrukcie
 - Princípy realizácie inštrukcie

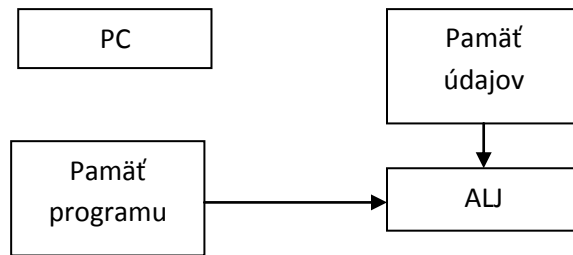


- 1 inštrukcia = n mikroinštrukcií
- Procesory RISC (reduced instruction set computer)
 - o do konca 80 rokov
 - o operácie



Pamäť programu RISC $\cong 2 \times$ pamäť programu CISC

- Procesory NISC
 - o riadiace slovo → bez dekódovania



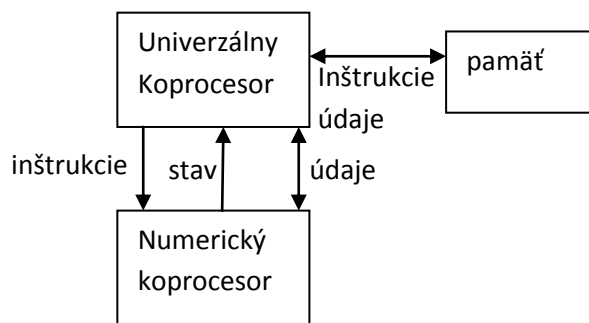
Riadiace slová – 2÷3 x dlhšie ako inštrukcie ale každé riadiace slovo 2÷3 inštrukcií RISC

3.2.3. Zvyšovanie výkonnosti Procesorov

- Parameter
- Možnosti
 - o zdokonalenie technológie do 3,2GHz
 - o zdokonalenie organizácie spracovania informácií

3.2.3.1. Výpočty v pohyblivej rádovej čiarke

- Špeciálne procesory → **Numerické koprocesory**

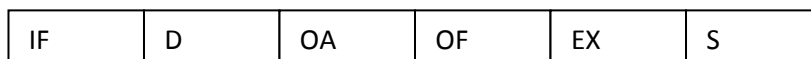


3.2.3.2. Predvýber a predspracovanie inštrukcií

- Zbernica – využitie
 - výber a dekódovanie

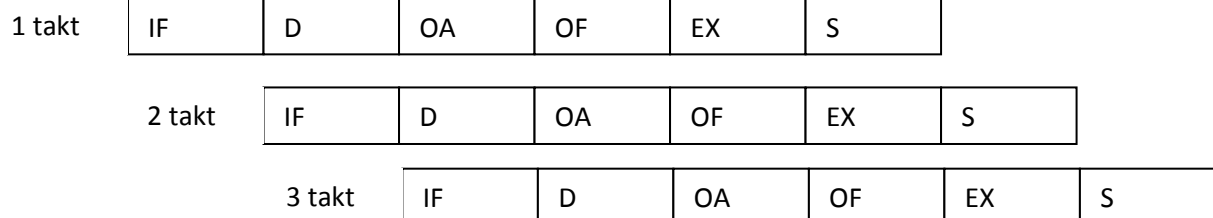
VYKONANÁ }
 VÝBEROVÁ } FÁZA

3.2.3.3 Prúdové spracovanie Inštrukcií



6 taktov spracúvania

$$3 \times 6 = 8$$

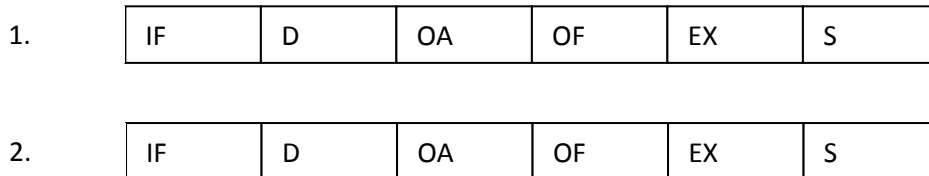


HAZARDY

HONFLIKTY

- Údajová nezávislosť
- Riadiaca nezávislosť

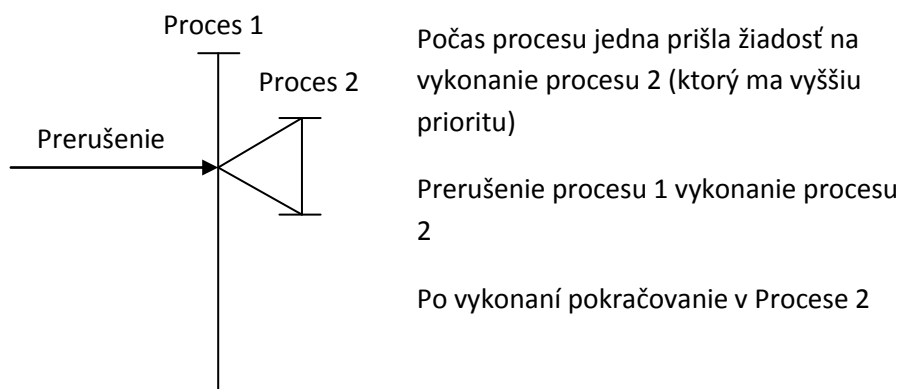
3.2.3.3. Paralelné vykonávanie inštrukcií



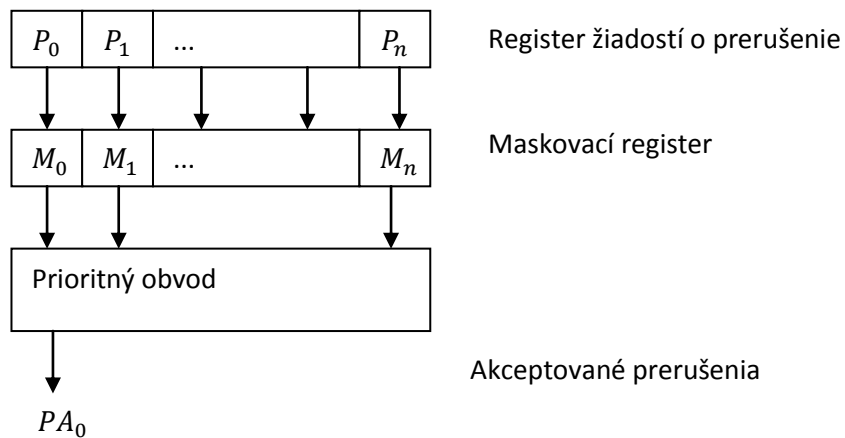
- Údajová nezávislosť
- Riadiaca nezávislosť

3.2.4. Prerušovací systém procesora

- Prioritný → Externá udalosť
- -operačný systém

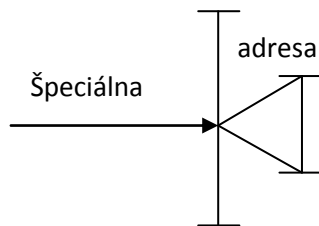


Register žiadostí o prenesenie



3.2.4.1. Klasifikácia prerušení

- **Podľa zdroja maskovacích signálov**
 - A. *Externé* – „okolie“ procesora alebo počítača nesúvisia s práve vykonávaným programom
 - B. *Interné* – súvisia s prácou vykonávaným programom
- **Podľa toho, kedy prerušenie môže nastať**
 - A. *Asynchrónne*
 - a. Súvisiace s vykonávanými inštrukciami
 - b. Kedykoľvek
 - c. Možno zakázať
 - B. *Synchrónne*
 - a. Súvisia s vykonávaným programom
 - b. Nemožno zakázať
- **Podľa toho či preruš možno alebo nemožno zakázať**
 - A. *Maskovateľné*
 - a. *Externé*
 - b. *Asynchrónne*
 - B. *Nemaskovateľné*
 - a. Mimoriadne dôležité udalosti
- **Pri synchrónnych prerušeníach**
 - A. *Softvérové* – špeciálne riadiace inštrukcie



- B. *Výnimka (exemption)*
Pri vykonávaní inštrukcií
IF → „KO neznámy“

3.2.4.2. Operácie prerušenia

6 krokov

1. Prijatie požiadavky na prerušenie

- *Udalosť*

→ dokončenie práve vykonávanej inštrukcie

→ obsluha prerušenia

2. Odloženie stavu procesora

- *Zásobník* – PC
- Hniezdenie prerušaní

3. Zisťovanie zdrojov prerušenia

- Prerušovací vektor adresa začiatkovej inštrukcie obslužného programu

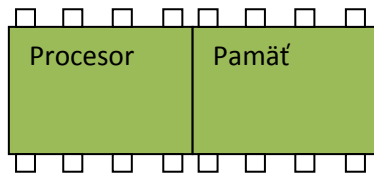
4. Vykonanie obslužného programu prerušenia

5. Obnovenie pôvodného stavu procesora

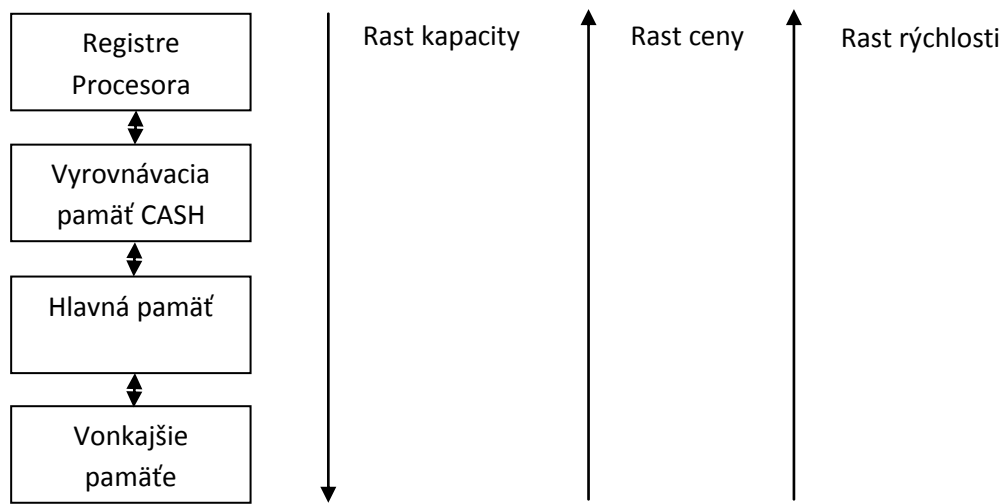
6. Pokračovanie vykonávania prerušeného programu

3.3. Pamäťový podsystem počítača

- Uloženie programov a údajov (práve používané i archivácie)



3.3.1. Hierarchická organizácia pamäťového podsystemu počítača



3.3.2. Rozdelenie pamätí

Spôsob prístupu k informácií

1. **Pamäte s primárnym prístupom** – DAM (direct access memory) (RAM - Random Access Memory)
2. **Pamäte so sekvenčným prístupom** – SAM (Sequentila Access Memory)
3. **Pamäte s asociatívnym prístupom** – CAM (Counter Access Memory)

- **klúč**



- Využíva sa v **cash**

Možnosti čítania a zápisu

1. **Pamäte pre čítanie a zápis** – RWM (Read Write Memory)
 - Energeticky závislé
 - Energeticky nezávislé
2. **Pamäte iba pre čítanie** – ROM (Read Only Memory)
 - Používateľ

- Výrobca

3. Dĺžka uchovaného slova

- Pamäte s bitovou organizáciou
- Pamäte so slovnou organizáciou
 - o slabika (byte)

4. Tvar prenášaných údajov

- Sériové pamäte
- Paralelné pamäte (číta sa celé slovo naraz)

5. Organizácia pamäťových buniek

- Polovodič
- RWM

I. Statické pamäte –SRAM

- Preklápací obvod
 - o TTL

II. Dynamické pamäte – DRAM

- Parazitná kapacita

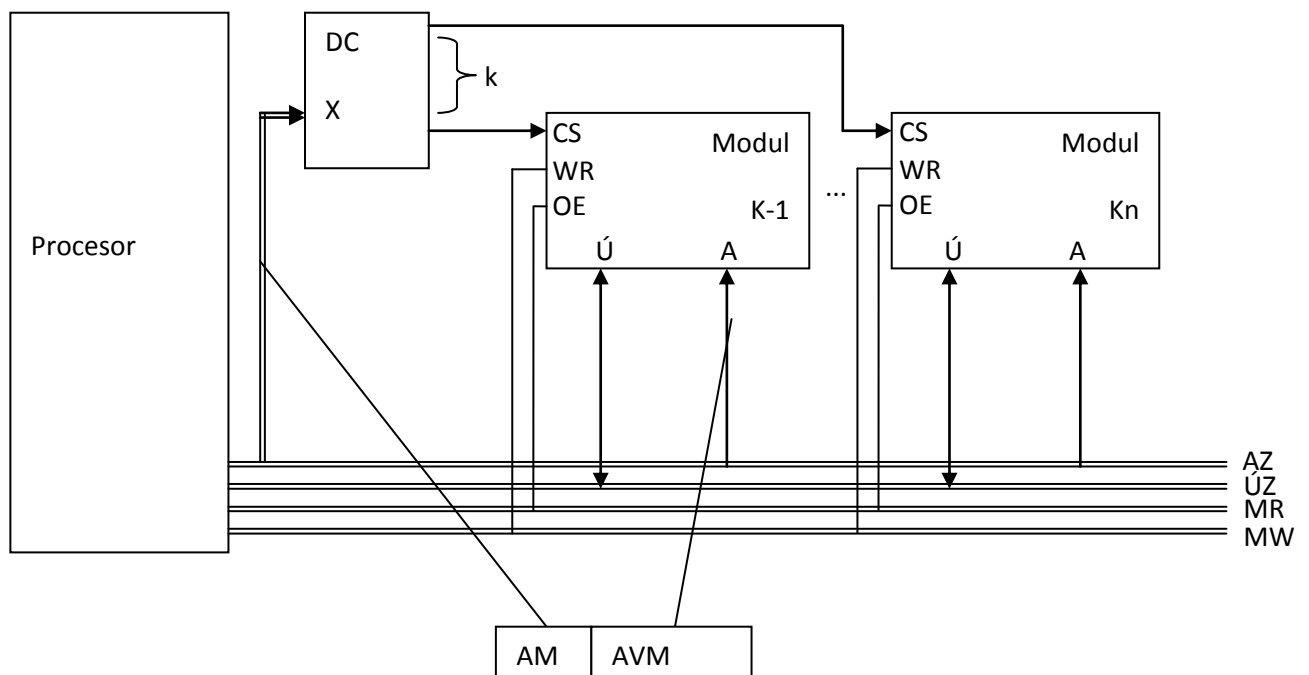
Obnovovanie obsahu (refresh)

3.3.3. Hlavná pamäť

- Program
- Údaje
- DAM
- RWM
- Statické / Dynamické – SRAM, DRAM
- Slovné organizované
- Paralelná

3.3.3.1. Pripojenie hlavnej pamäte k zbernici počítača (počítača)

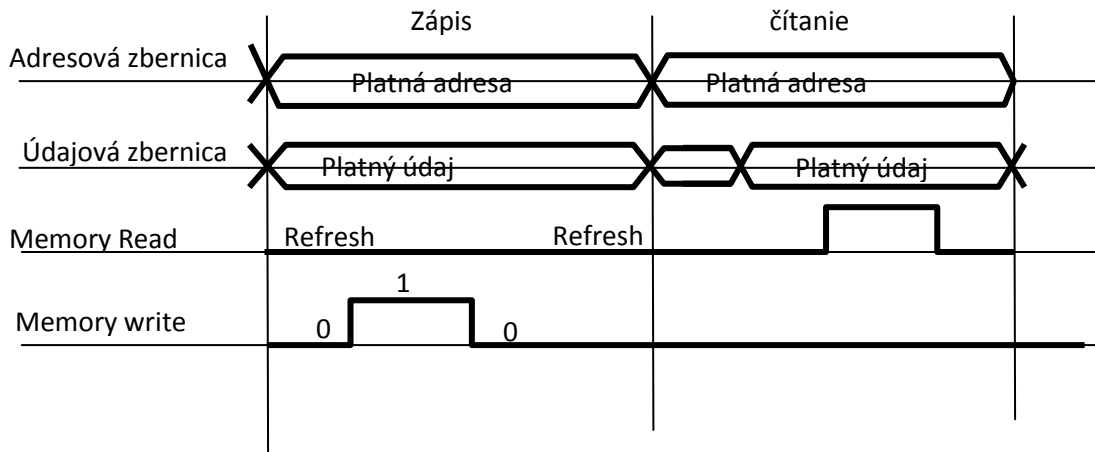
- Adresová
 - Údajová
 - riadiaca
- sekvencia



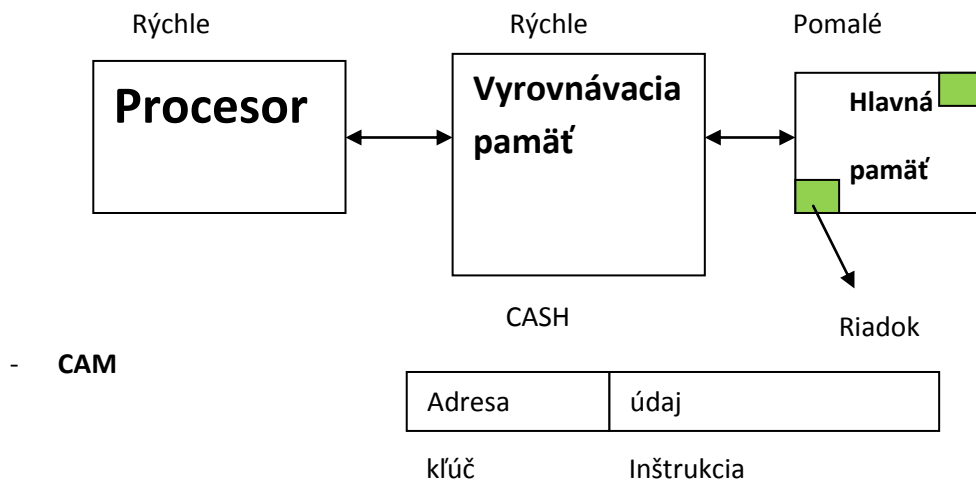
- DC – dekodér
- -čip select
- Output enable

3.3.3.2. Komunikácia procesora s hlavnou pamäťou

- Signálové sledy



3.3.4. Vyrovnávacia pamäť



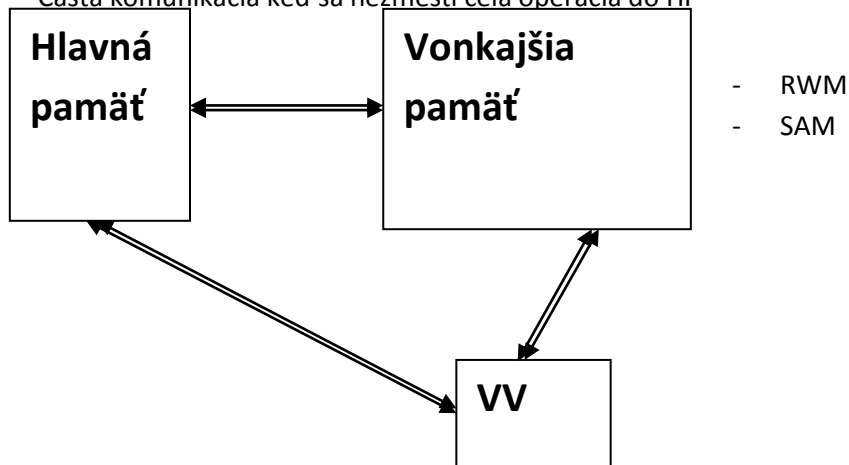
- CAM

Nekonzistencia Informácií (Udaj vo VP kopírovať hneď do HP)

- **Zápis späť (copy-back)** (až po tom ako sa riadok stane neaktuálny)
- **Zápis cez vyrovnávaciu pamäť** (okamžité prepisovanie)
- **Výber riadku**
 - **LRU (least recently Used)** – riadok, ktorý bol najdlhší čas nepoužitý
 - **LFU (least frekvently used)** – najmenej často využívaný riadok

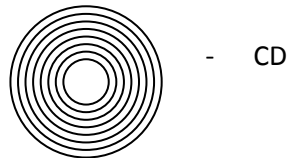
3.3.5. Vonkajšie pamäte

- Archivácia
- Prechodné uchovávanie informácií počas výpočtu
- Častá komunikácia keď sa nezmestí cela operácia do HP

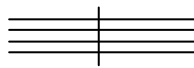


3.3.5.1. Diskové pamäte

- magnetické
- optické
- opto-magnetické



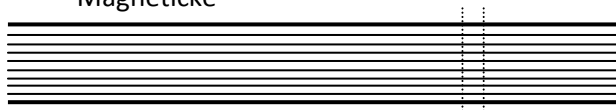
Disky obsahujú aj viac diskov každý disk ma svoju hlavičku ktorá ho číta



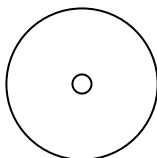
DVD – špirála

3.3.5.2. Páskové pamäte

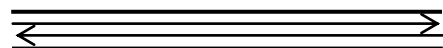
- Magnetické



- Cievkové – SAM
- Kazetové - RAM



Používané v PC 2 vrstvy



8 bytov a 1 paritný

Parita hovorí či je počet 1-notiek párný alebo ne

Pozdĺžna a priečna parita

3.3.6. Správy a ochrana hlavnej pamäte

- Absolútna adresa – AZ
- Fyzický pamäťový priestor – HP → FPP
- Logický pamäťový priestor – Program + údaje → LPP
- Rozpracovanie viacerých programov aby bol procesor stále využívaný
- Potrebná správa pamäte

FPP = LPP

Potrebnosť správy pamäte:

1. LPP > FPP
 - Virtuálna pamäť → rýchla vonkajšia pamäť (pevná – HDD)
 2. Viac používateľov – viac programov v HP
 - Dynamická relokácia (premiestňovanie údajov v čase)
 - Rôzne Absolútne adresy
 3. Prístupové práva – najdôležitejšia operácia zápisu (zápis na nepovolenú časť pamäte)
- OCHRANA** HW+SW jednotka *memory management unit* **MMU**

3.3.6.1. Virtuálna pamäť

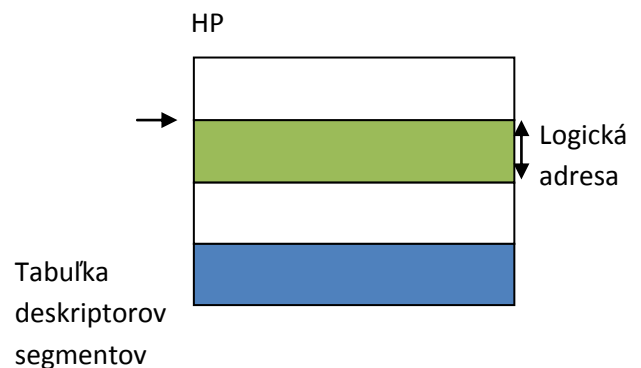
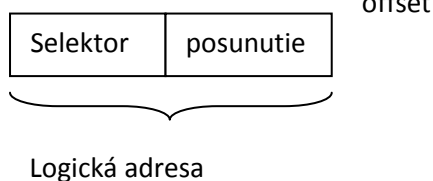
Existencia rýchlej vonkajšej pamäte

- 1) Segmentovanie
- 2) Stránkovanie

1. Segmentovanie

- Program a údaje
 - o rozdelenia bloky → Segmenty
 1. Segment – program
 2. Segment – podprogram
 3. Segment – údaje

Logická adresa ← program



Deskriptor:

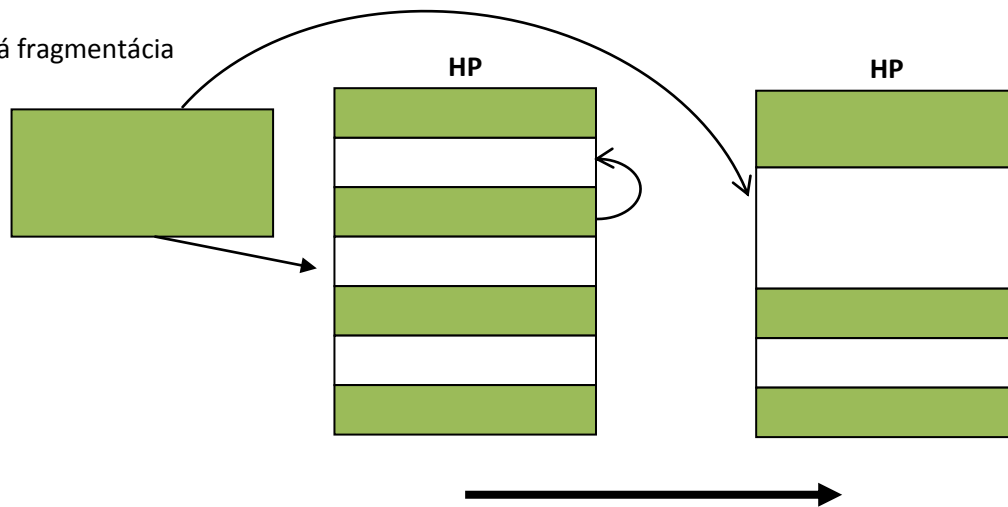
1. **Bázová adresa segmentu (base)**
 - Začiatok segmentu
2. **Veľkosť segmentu (limit)**
 - offset < Limit - ochrana
3. **Atribúty (vlastnosti) segmentu (attributes)**
 - a. Informácia o prítomnosti segmentu v HP
 - b. Informácia o type segmentu (iba čítanie, čítanie a zápis)
 - c. Informácia o privilegovanej úrovni segmentu

4. Adresa segmentu vo vonkajšej pamäti

Absolútna adresa → **báza + posunutie**

Relatívne adresovanie

Externá fragmentácia



2. Stránkovanie

HP – úseky s rovnakou dĺžkou → **stránky** (stránkové rámy)

Výpadok prúdu ☺
KONEC
