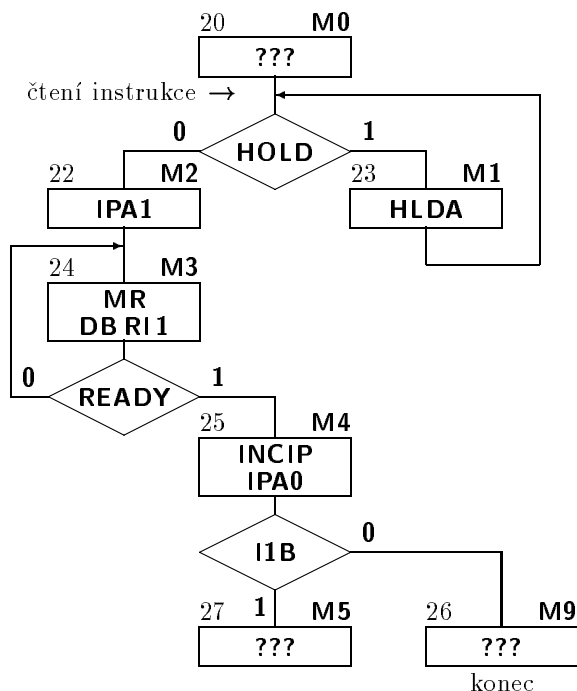
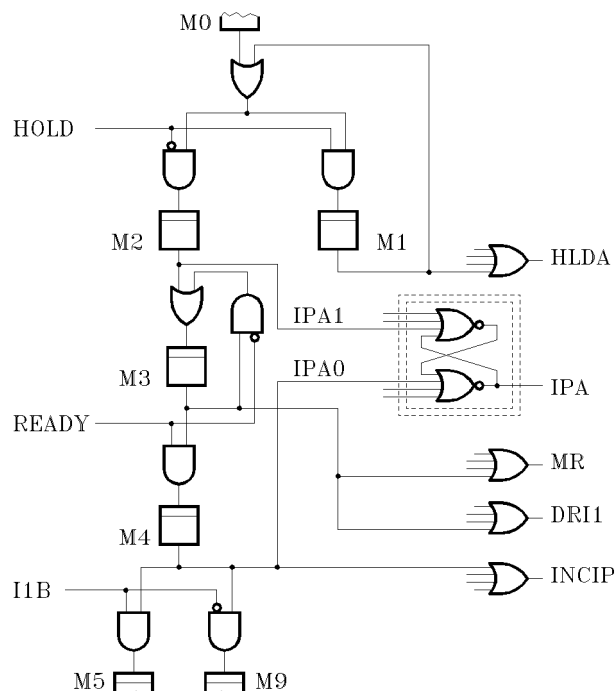


příklad vývojového diagramu řadiče

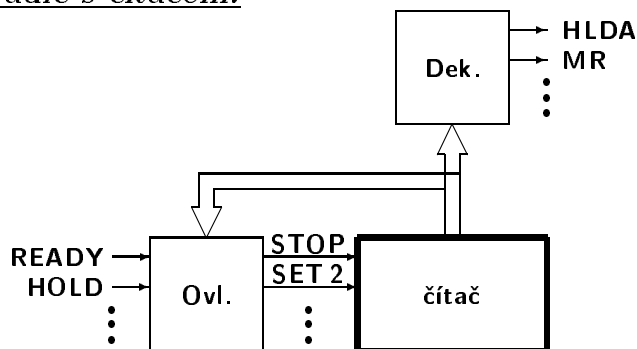
(viz UPS12 – 10)



řadič s řídicími řetězci:



řadič s čítačem:



Dek. ... dekodér Ovl. ... ovládání čítače

Ovládání čítače:

čítač	podmínka	ovládání
0	$\overline{\text{HOLD}}$	SET 2 nastavení na 2
1	HOLD	STOP nečítá
3	$\overline{\text{READY}}$	STOP nečítá
4	$\overline{\text{I1B}}$	SET 9 nastavení na 9
...	...	...
jinak		čítá

mikroprogramovaný řadič — horizontální mikroprogramování:

kódování podmínek:

VP	podmínka
0	adr0
1	READY
2	HOLD
3	I1B
...	...

μprogram

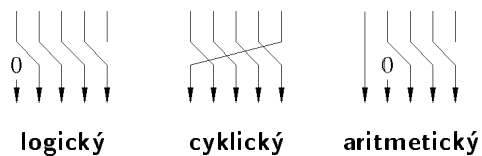
		μOZ							adr	VP
		HLDA	MR	IPA0	IPA1	INCIP	DRI1	...		
M0	20	?	?	?	?	?	?	...	22	2
	21	?	?	?	?	?	?	...	??	?
M2	22	0	0	0	1	0	0	...	24	0
M1	23	1	0	0	0	0	0	...	22	2
M3	24	0	1	0	0	0	1	...	24	1
M4	25	0	0	1	0	1	0	...	26	3
M9	26	?	?	?	?	?	?	...	??	?
M5	27	?	?	?	?	?	?	...	??	?
...	...	...	...	...	...	...	...	...	...	...

Obvody pro posuv

[shifters]

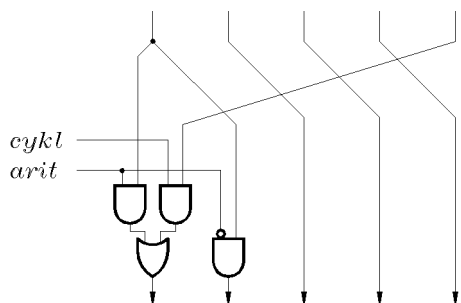
Př.: posuv vpravo o 1 místo

přímý kód — bez detekce ztráty přesnosti



řídící signály:

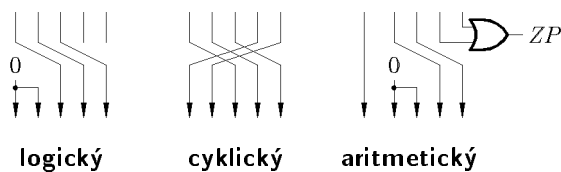
arit	cykl	
0	0	logický
0	1	cyklický
1	0	aritmetický
1	1	???



Obvody pro posuv - II.

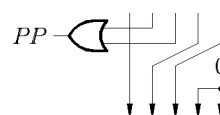
Př.: posuv vpravo o 2 místa

přímý kód — s detekcí ztráty přesnosti (ZP)



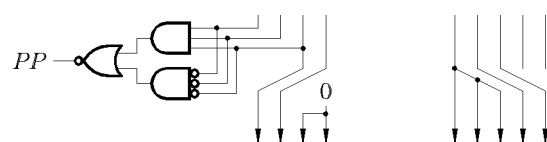
Př.: aritmetický posuv vlevo o 2 místa

přímý kód — s detekcí přeplnění (PP)

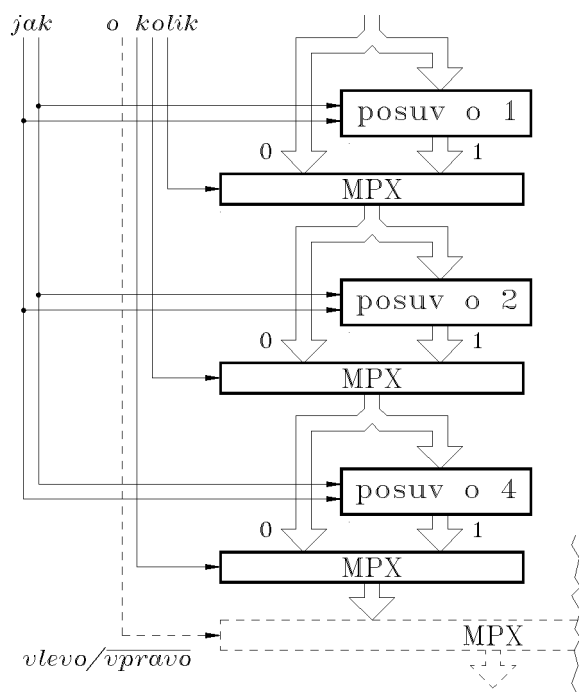


Př.: doplňkový kód — aritmetický posuv o 2 místa

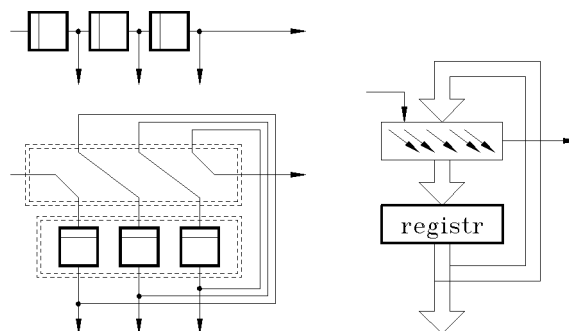
vlevo s detekcí přeplnění (PP) a vpravo bez detekce ztráty přesnosti



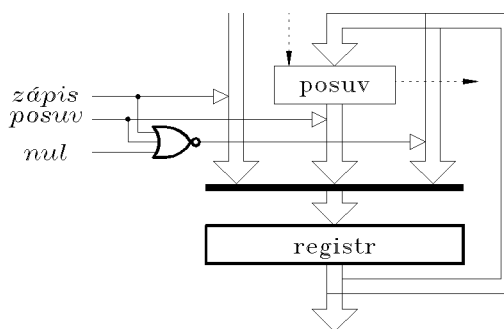
[barrel shifter]



triviální posuvný registr



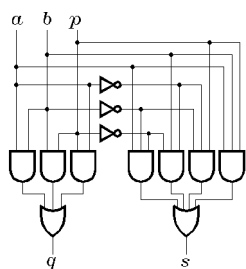
posuvný registr s paralelním vstupem



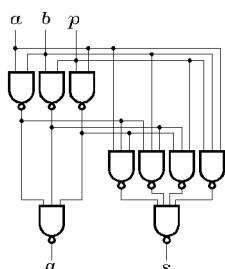
úplná sčítačka

[full adder]

NOT-AND-OR



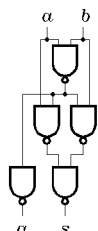
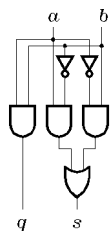
NAND-NAND-NAND



$$s = \bar{a}bp + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp \quad \text{Využívá se např. vztah:}$$

$$q = ab + ap + bp \quad a \cdot \bar{a}b \cdot \bar{a}c = a\bar{b}\bar{c} \text{ apod.}$$

poloviční sčítačka (pulsčítačka) [half adder]

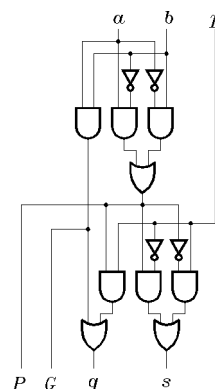


$$s = \bar{a}b + a\bar{b} = a \oplus b \quad \text{Využívá se vztahů}$$

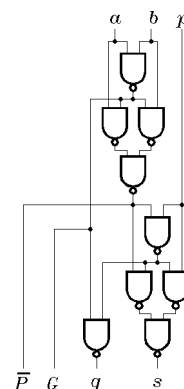
$$q = ab \quad a \cdot \bar{a}b = a\bar{b} \text{ a } \bar{a}b \cdot b = \bar{a}b$$

úplná sčítačka ← pulsčítačky

NOT-AND-OR



NAND-NAND-NAND



$$s = (a \oplus b) \oplus p$$

$$q = ab + p(a \oplus b)$$

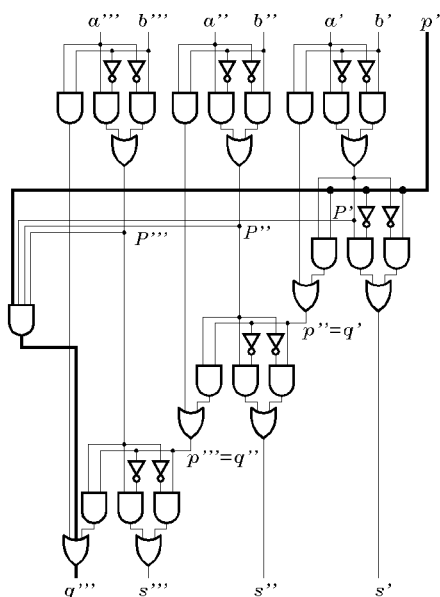
$$G = ab \dots \text{přenos se G eneruje (vzniká)} \quad [\text{Generate}]$$

$$P = a \oplus b \dots \text{přenos P rochází} \quad [\text{Propagate}]$$

Pozn.: Lze použít také $P = a + b$

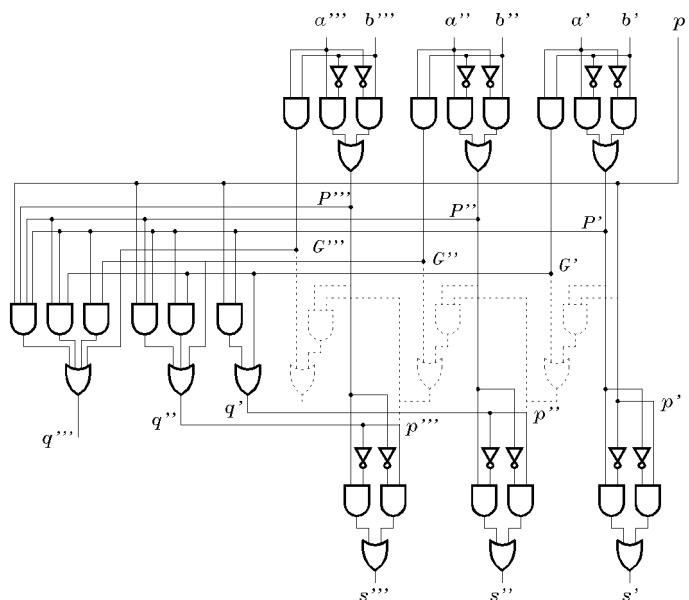
Výhybka

(3bitová sekce)



Predikce přenosů

[carry look-ahead]

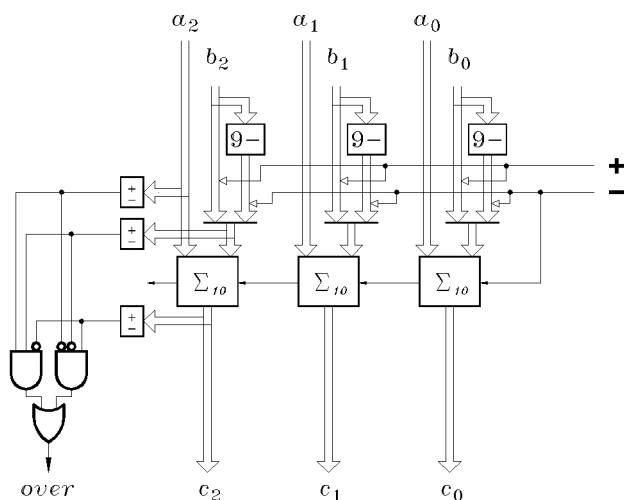


Není nakresleno:

$$G^* = G''' + P''' \cdot G'' + P''' \cdot P'' \cdot G'$$

$$P^* = P''' \cdot P'' \cdot P'$$

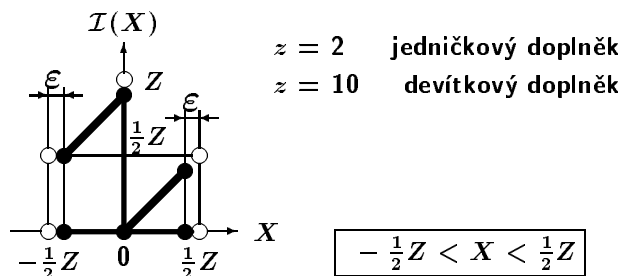
Desítková sčítačka–odčítačka v doplňkovém kódu



Inverzní kód

$$\mathcal{I}(X) = \begin{cases} X, & \text{je-li } X \geq 0, \\ \widetilde{|X|} = Z - \varepsilon + X, & \text{je-li } X \leq 0, \end{cases}$$

$$\text{neboť } \widetilde{|X|} = Z - \varepsilon - |X| = Z - \varepsilon + X$$



rozsah: symetrický

zobrazení: není jednoznačné: kladná a záporná nula

znaménko: číslíce v nejvyšším řádu

— jako u doplňkového kódu, ale:

kladná a záporná nula

(odvození jako u doplňkového kódu)

Inverzní kód — II.

Př.: $z = 2$ $Z = 1000_2$ $\varepsilon = 1$

X	$\mathcal{I}(X)$	
+0	0 0 0	← kladná nula
+1	0 0 1	
+2	0 1 0	
+3	0 1 1	
-3	1 0 0	
-2	1 0 1	← záporná nula
-1	1 1 0	
-0	1 1 1	

Př.: $z = 10$ $Z = 1000_{10}$ $\varepsilon = 1$

X	$\mathcal{I}(X)$	
+0	0 0 0	← kladná nula
+1	0 0 1	
...	...	
+499	4 9 9	
-499	5 0 0	
...	...	
-1	9 9 8	← záporná nula
-0	9 9 9	

Sčítání v inverzním kódu

	$\mathcal{I}(A) + \mathcal{I}(B)$	$\mathcal{I}(A+B)$
$A \geq 0, B \geq 0$	$A+B$	$A+B$
$A \geq 0, B < 0$	$A+B+Z-\varepsilon$	$A+B$
$A < 0, B \geq 0$		$A+B+Z-\varepsilon$
$A < 0, B < 0$	$A+B+Z-\varepsilon+Z-\varepsilon$	$A+B+Z-\varepsilon$

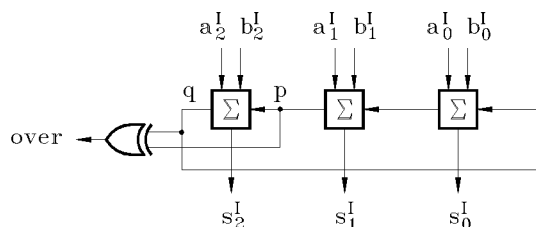
$$\mathcal{I}(A+B) = \begin{cases} \mathcal{I}(A) + \mathcal{I}(B) \\ \mathcal{I}(A) + \mathcal{I}(B) - Z + \varepsilon \end{cases}$$

Sečtou se obrazy a
přenos z nejvyššího řádu
se zavede do nejnižšího řádu

kruhový přenos

přeplnění: jako u doplňkového kódu

Př.: $z = 2$ $Z = 1000_2$ $\varepsilon = 1$



Násobení

$$\begin{array}{lll}
 \text{modul} & \text{jednotka} & \\
 A & Z = z^n + 1 & \varepsilon = z^{-m} \quad A < Z \\
 B & Y = z^\nu + 1 & \zeta = z^{-\mu} \quad B < Y \\
 A^* = A/\varepsilon & & \\
 B^* = B/\zeta & \left. \vphantom{\begin{array}{l} A^* = A/\varepsilon \\ B^* = B/\zeta \end{array}} \right\} \text{ celá čísla}
 \end{array}$$

$$A \cdot B = A^* \cdot B^* \cdot \varepsilon \cdot \zeta$$

$$A \cdot B = A^* \cdot B^* \cdot z^{-m-\mu}$$

$$A \cdot B < Z \cdot Y = z^{n+1+\nu+1}$$

	počty číslic	
	před čárkou	za čárkou
A	$n+1$	m
B	$\nu+1$	μ
$A \cdot B$	$n+1+\nu+1$	$m+\mu$

násobení $\rightarrow$ násobení celých čísel

dále: pouze násobení celých čísel !!!

Násobení nezáporných čísel

$$B = \sum_{i=0}^{\nu} b_i \cdot z^i \implies A \cdot B = \sum_{i=0}^{\nu} A \cdot b_i \cdot z^i$$

$$A \cdot b_i = ? \quad (\text{jinak sčítání a posuvy})$$

$$1 \quad z = 2: \quad A \cdot b_i = \begin{cases} 0 \\ A \end{cases} \quad (\text{UPS8 - 1})$$

$$\begin{array}{l}
 2 \quad z \neq 2: \quad A \cdot 0 = 0 \\
 \quad \quad \quad A \cdot 1 = A \\
 \quad \quad \quad A \cdot 2 = A + A \\
 \quad \quad \quad A \cdot 3 = A + A + A \\
 \quad \quad \quad \vdots
 \end{array}$$

? dvojková $\rightarrow$ desítková násobička ?

- **sčítačka** (desítková)
- **posuvy** (desítkově)
- **registr B: sestupný čítač** (aspoň nejnižší řád)
- **sčítačka, obvod pro posuv a registr C1: o 1 řád širší** (nelze využít přenos — nestačí)
- **posuv až po zpracování 1 číslice** (po vynulování poslední číslice v registru B)

Násobení čísel v přímém kódu

absolutní hodnota:

$$|A \cdot B| = |A| \cdot |B|$$

abs. hodn.: nezáporná čísla

znaménko:

+	+	+	~	0	0	0
+	-	-		0	1	1
-	+	-		1	0	1
-	-	+		1	1	0

$$\text{zn}(A \cdot B) = \text{zn}(A) \oplus \text{zn}(B)$$

Násobení čísel v doplňkovém kódu

$$z=2$$

1 Boothova metoda

čísllice: 0, 1 a -1 — tzv. relativní číslice

konvence: $\hat{1} = -1$ (obecně: $\hat{a} = -a$)

$$\begin{array}{l}
 \text{Př.:} \quad 0 \ 1 \ 0 \ 1 \ 1 = 1 \ \hat{1} \ 1 \ 0 \ \hat{1} \\
 0 \cdot 16 + 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 1 \cdot 16 - 1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 - 1 \cdot 1 \\
 8 + 2 + 1 = 16 - 8 + 4 - 1
 \end{array}$$

i	4	3	2	1	0
b_i	0	1	0	1	1
β_i	1	$\hat{1}$	1	0	$\hat{1}$

$$\beta_i = f(b_i, \dots) = ?$$

$\beta_i \neq f'(b_i) \dots$ konečný automat

$$\begin{array}{lll}
 b_i & \rightarrow & \beta_i \\
 0 & \rightarrow & 0 \quad \& \text{ nic} \quad \dots \text{ stav Q0} \\
 1 & \rightarrow & \hat{1} \quad \& \ b_{i+1} := b_{i+1} + 1 \quad \dots \text{ stav Q1} \\
 & & \dots
 \end{array}$$

	b_i				$\Rightarrow$	b_i		
	0	1	0	1		$b_i - 1$	0	1
Q0	0	-1	Q0	Q1		0	0	-1
Q1	1	0	Q0	Q1		1	1	0

počáteční stav: Q0 $b_{-1} = 0$

Násobení čísel v doplňkovém kódu

I — II.

$$\mathcal{D}(B) = \sum_{i=0}^{\nu} b_i^{\mathcal{D}} \cdot 2^i \quad \text{modul: } Y$$

označme: $\hat{B}$ číslo získané převodem obrazu $\mathcal{D}(B)$

A $b_{\nu}^{\mathcal{D}} = 1 \Rightarrow B < 0$

automat skončí ve stavu Q1 $\Rightarrow$

$\Rightarrow$ nepřičetla se 1 v řádu $\nu+1$

$$\Rightarrow \hat{B} = \mathcal{D}(B) - Y = Y + B - Y$$

$$\Rightarrow \boxed{\hat{B} = B}$$

Př.: celá čísla — 4 bity

$$B = -5 = -101_2$$

$$\mathcal{D}(B) = 1011$$

$$\hat{B} = \hat{1101} = -8 + 4 - 1 = -5$$

B $b_{\nu}^{\mathcal{D}} = 0 \Rightarrow B \geq 0$

automat skončí ve stavu Q0 $\Rightarrow$

$$\Rightarrow \hat{B} = \mathcal{D}(B) = B$$

$$\Rightarrow \boxed{\hat{B} = B}$$

Násobení čísel v doplňkovém kódu

I — př.

Př.: $A = -11_2$ $B = -010_2$ $A \cdot B = 110_2$
 $\mathcal{D}(A) = 101$ $\mathcal{D}(B) = 110$ $\mathcal{D}'(A \cdot B) = 000\ 110$
 $B = -010 = 0\hat{1}0$

$$\begin{array}{r} 1\ 1\ 1\ 1\ 0\ 1\ 0\ \hat{1}\ 0 \\ \hline 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\ \hline 0\ 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1\ 0\ \hat{1} \\ \hline 1 \\ 0\ 0\ 0\ 1\ 1 \\ \hline 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \end{array}$$

negace
horká 1

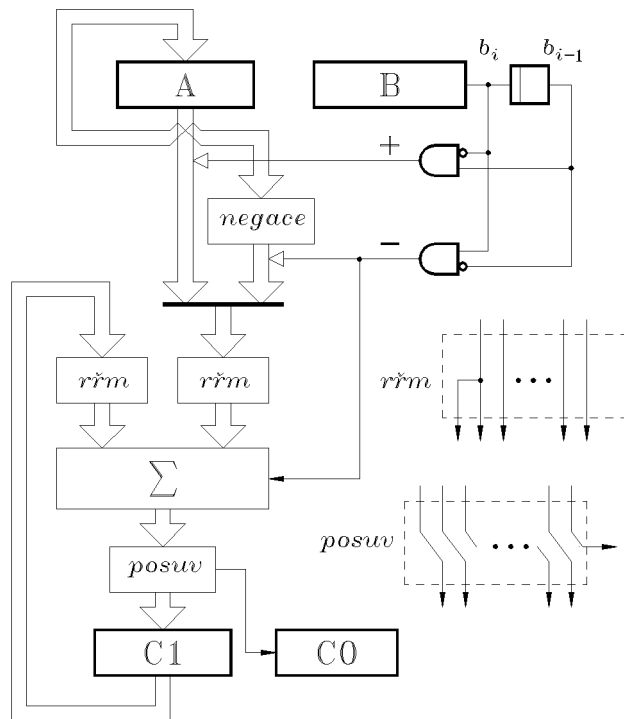
použití 4bitové sčítačky

$$\begin{array}{r} 1\ 1\ 0\ 1\ 0\ \hat{1}\ 0 \\ \hline 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0 \\ \hline 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 1\ 0\ \hat{1} \\ \hline 1 \\ 0\ 0\ 0\ 1\ 1 \\ \hline 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 1 \end{array}$$

negace
horká 1

Násobení čísel v doplňkovém kódu

I — schéma



Násobení čísel v doplňkovém kódu

2

2 jiný způsob:

$$\mathcal{D}(B) = \sum_{i=0}^{\nu} b_i^{\mathcal{D}} \cdot 2^i = B' + B'',$$

kde $B' = b_{\nu}^{\mathcal{D}} \cdot 2^{\nu}$ a $B'' = \sum_{i=0}^{\nu-1} b_i^{\mathcal{D}} \cdot 2^i$

převěde se pouze $B' = \begin{cases} 10 \dots 0 & \rightarrow \hat{1}0 \dots 0 \\ 00 \dots 0 & \rightarrow 00 \dots 0 \end{cases}$

Tedy: První bit $\mathcal{D}(B)$ zleva (tzn. $b_{\nu}^{\mathcal{D}}$) má opačnou váhu, tzn. váhu -2^{ν} (nikoliv váhu $+2^{\nu}$).

Závěr: jako nezáporná čísla, ale v posledním kroku odčítání (0 nebo 1 násobku) místo přičítání.

Př.: $A = -11_2$ $B = -010_2$ $A \cdot B = 110_2$
 $\mathcal{D}(A) = 101$ $\mathcal{D}(B) = 110 \rightarrow \hat{1}10$

$$\begin{array}{r} 1\ 1\ 0\ 1\ 1\ 1\ 0 \\ \hline 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0 \\ \hline 0\ 0\ 0\ 0\ 0 \\ 1\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 1\ 0\ 1 \\ 0\ 0\ 1\ 0\ \hat{1} \\ \hline 1 \\ 0\ 0\ 0\ 1 \end{array}$$

znam. rozšíření
negace
horká 1

Dělení

$$A/B = ?$$

$$B \neq 0 \quad !!!$$

nezáporná čísla — dvojková soustava

I. $Z = 1 \Rightarrow A < 1, B < 1$

— s návratem přes nulu (UPS8 – 2)

— bez návratu přes nulu (UPS8 – 3)

■ sch.

II. $\varepsilon = 1 \Rightarrow$ celá čísla

$$A \div B = ?$$

$$A \% B = ?$$

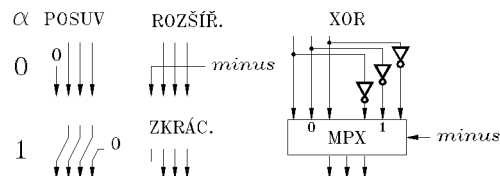
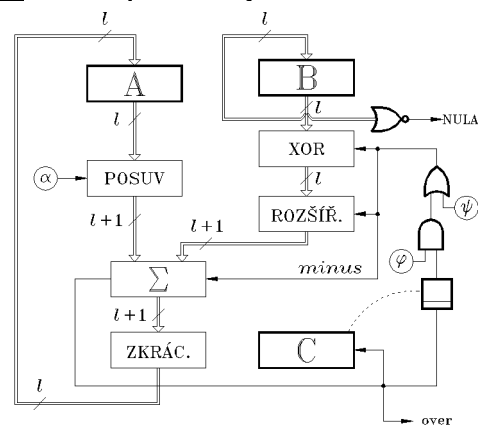
1. • Náhrada dělitele: $B \leftarrow B \cdot z^k = B \ll k$,
kde k je max. takové číslo, že:

$$B \cdot z^k \geq A \text{ nebo } B \ll k \text{ nelze provést}$$

- $0 \rightarrow C$ & $k+1$ kroků podle 1. ■ př. 1

- zbytek := zbytek $\gg k$ ■ př. 2

2. „dlouhý“ registr pro dělence A :

dělenec A : l bitů dělitel B : k bitůdélka registru A : $l+k$ ■ př. 2Dělení — nezáporná dvojková l bitová čísla < 1 

	α	φ	ψ
1. takt	0	—	1
další takty	1	1	0
návrat	0	0	0

$l+1$ (popř. $l+2$) taktů
podíl ... C
zbytek ... A

dělení celých čísel

1 př. 1a

pozorování

$$111 : 011$$

$$A = 111$$

$$B = 011 \Rightarrow B \ll k = 110 \Rightarrow k = 1 \Rightarrow 2 \text{ kroky}$$

$$A = 111 \quad B = 110$$

$$\begin{array}{r} 0111 \\ \square 1001 \text{ negace} \\ \hline 10001 \text{ horká 1} \\ + \Rightarrow 1 \end{array}$$

$$\begin{array}{r} 0010 \\ \square 1001 \\ \hline 01100 - \Rightarrow 0 \end{array}$$

$$\begin{array}{r} 0100 \\ \square 0110 \text{ návrat} \\ \hline 01010 \\ \downarrow \downarrow \downarrow \\ 010 \end{array}$$

01,0 — zbytek 10 — podíl

dělení celých čísel

1 př. 1b

$$A = 111$$

$$B = 011 \Rightarrow B \ll k = 110 \Rightarrow k = 1 \Rightarrow 2 \text{ kroky}$$

$$A = 111 \quad B = 110$$

$$\begin{array}{r} 0111 \\ \square 1001 \text{ negace} \\ \hline 10001 \text{ horká 1} \\ + \Rightarrow 1 \end{array}$$

$$\begin{array}{r} 0010 \\ \square 1001 \\ \hline 01100 - \Rightarrow 0 \end{array}$$

$$\begin{array}{r} 0100 \\ \square 0110 \text{ návrat} \\ \hline 01010 \\ \downarrow \downarrow \downarrow \\ 010 \end{array}$$

01,0 — zbytek 10 — podíl

dělení celých čísel

[2] př. 2a

pozorování

$$111 : 011$$

$$\begin{array}{r}
 000111 : 0011 \\
 \boxed{-} \quad 1100 : : \text{negace} \\
 \quad \quad 1 : : \text{horká 1} \\
 0 \quad 1110 : : - \Rightarrow 0 \\
 \quad \downarrow \downarrow \downarrow \downarrow : \\
 \quad 1101 : \\
 \boxed{+} \quad 0011 : \\
 1 \quad 00001 : + \Rightarrow 1 \\
 \quad \downarrow \downarrow \downarrow \downarrow : \\
 \quad 0001 : \\
 \boxed{-} \quad 1100 : \\
 \quad \quad 1 : \\
 0 \quad 1110 : - \Rightarrow 0 \\
 \boxed{\boxtimes} \quad 0011 : \text{návrat} \\
 1 \quad 0001 : \\
 \quad 001 \text{ — zbytek} \quad 010 \text{ — podíl}
 \end{array}$$

dělení celých čísel

[2] př. 2b

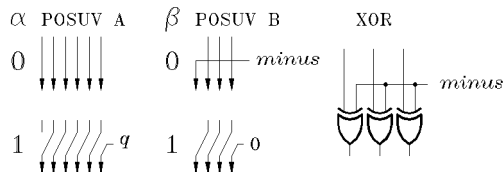
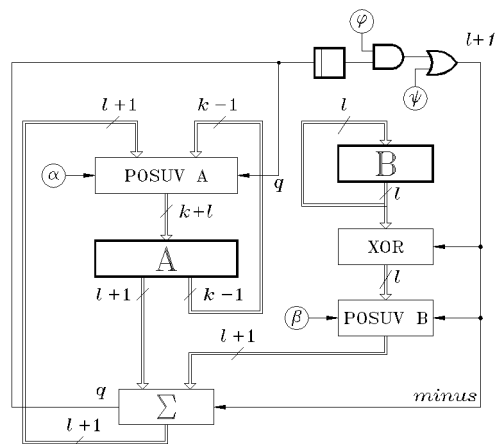
$$A = 111$$

$$B = 011$$

$$\begin{array}{r}
 000111 : 0011 \\
 \boxed{-} \quad 1100 : : \text{negace} \\
 \quad \quad 1 : : \text{horká 1} \\
 0 \quad 1110 : : - \Rightarrow 0 \\
 \quad \swarrow \swarrow \swarrow \swarrow \swarrow \swarrow : \\
 \quad 11011? : \\
 \boxed{+} \quad 0011 : : \\
 1 \quad 00001 : : + \Rightarrow 1 \\
 \quad \swarrow \swarrow \swarrow \swarrow \swarrow \swarrow : \\
 \quad 0001?? : \\
 \boxed{-} \quad 1100 : : \\
 \quad \quad 1 : : \\
 0 \quad 1110 : : - \Rightarrow 0 \\
 \quad \swarrow \swarrow \swarrow \swarrow \swarrow \swarrow : \\
 \quad 110??? : \\
 \boxed{\boxtimes} \quad 0110 : : \text{návrat} \\
 1 \quad 001? : : \\
 \quad \downarrow \downarrow \downarrow \downarrow \downarrow \downarrow : \\
 \quad 001??? : \\
 \quad 001 \text{ — zbytek} \quad 010 \text{ — podíl}
 \end{array}$$

dělení celých čísel

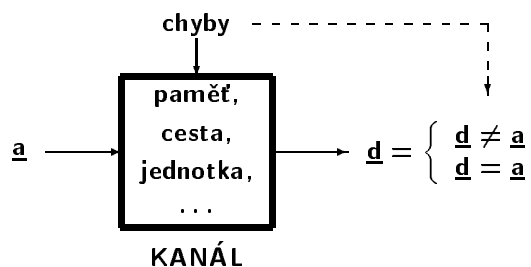
[2]



	α	β	φ	ψ
1. takt	1	0	—	1
další takty	1	0	1	0
návrat	0	1	0	0

l (popř. $l+1$) taktů
 podíl ... nižší řády A
 zbytek ... vyšší řády A

BEZPEČNOSTNÍ KÓDY



modely kanálů (charakter chyb):

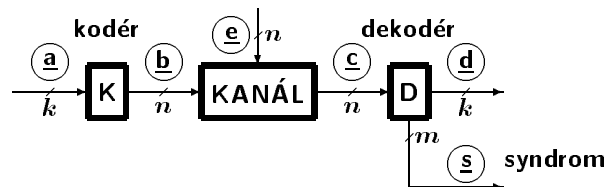
kanál • symetrický: $\text{pravd}(0 \rightarrow 1) = \text{pravd}(1 \rightarrow 0)$

- nesymetrický: $\neq$
- s výmazem

kanál • bez paměti — „nezávislé“ chyby

- s pamětí — shluky chyb

Př.: $\begin{array}{l} \underline{a} = 1110\ 1111 \\ \underline{d} = 1011\ 0111 \\ \quad \downarrow \downarrow \downarrow \\ \quad 0101\ 1000 \\ \underline{a} \oplus \underline{d} \end{array} \left\{ \begin{array}{l} 3 \text{ chyby} \quad \text{nebo} \\ 1 \text{ shluk délky } 4 \quad \text{anebo} \\ 2 \text{ shluky délky } 1 \text{ a } 2 \end{array} \right.$



$\underline{a} = (a_1, \dots, a_k)$

$\underline{b} = (b_1, \dots, b_n)$ slovo, vektor, znak, ...

...

slova $\left\{ \begin{array}{l} \text{kódová,} \\ \text{nekódová,} \end{array} \right.$ nebo znaky $\left\{ \begin{array}{l} \text{přípustné,} \\ \text{nepřípustné.} \end{array} \right.$

ozn.: $B = \{\text{kódová slova}\}$

kód $\mathcal{K}$: $\underline{a} \rightarrow \underline{b}$

!!! někdy pouze B !!!

Př.:

$\underline{a}$	$\underline{b}$			
	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{K}_3$	
0 0	0 0 0	1 1 1	0 0 0 0 0	α
0 1	0 1 1	0 0 1	0 1 0 1 1	β
1 0	1 0 1	0 1 0	1 0 1 0 1	γ
1 1	1 1 0	1 0 0	1 1 1 1 0	δ

dvojkový kód: $b_i \in \{0, 1\}$

blokový kód (n, k) — určeno n a k

k informační (rozhodovací) obsah, $m = n - k$ redundance, nadbytečnost

k/n norma kódu, přenosová rychlost, poměrná (relativní) míra informace

m/n poměrná (relativní) redundance

systematický kód: slovo $\left\{ \begin{array}{l} \text{informační část} \\ \text{kontrolní část} \end{array} \right.$

např. $\mathcal{K}_1, \mathcal{K}_3$ — syst.; $\mathcal{K}_2$ — nesyst.

chyby: vektor chyb $\underline{e} = \underline{b} \oplus \underline{c}$

$\underline{c} = \underline{b} \oplus \underline{e} \iff \underline{b} = \underline{c} \oplus \underline{e}$

zjišťování chyb: $\underline{c} \notin B$

oprava chyb: $\underline{c}$ nahradit „nejpodobnějším“ $\underline{b}$

Př.: $\mathcal{K}_3$ $\underline{c} = 1\ 0\ 1\ 1\ 1 \notin \{\alpha, \beta, \gamma, \delta\}$
 $\underline{b} = 1\ 0\ 1\ 0\ 1 = \gamma \implies \underline{d} = 10$
 $\underline{e} = 0\ 0\ 0\ 1\ 0$

detekční kód — detekce (zjištění) chyb

samoopravný kód — korekce (oprava) chyb

bezpečnostní kód — detekční / samoopravný kód

Hammingova vzdálenost:

[1] slova $\underline{b}'$ a $\underline{b}''$: $\text{vzd}(\underline{b}', \underline{b}'')$ — počet odlišných bitů

Př.: $\mathcal{K}_3$

	β	γ	δ
α	3	3	4
β		4	3
γ			3

[2] kódu — kódová vzdálenost

— $kvzd = \min \text{vzd}(\underline{b}', \underline{b}'')$

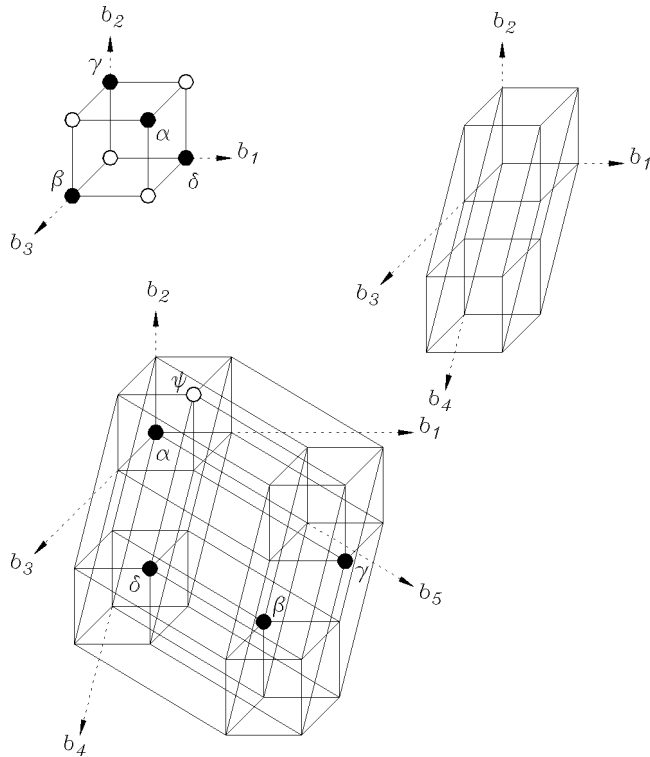
např. pro $\mathcal{K}_3$: $kvzd = 3$

váha slova — počet jedniček

Př.: $\mathcal{K}_3$

α	β	γ	δ
0	3	3	4

$\text{vzd}(\underline{b}', \underline{b}'') = \text{váha}(\underline{b}' \oplus \underline{b}'')$

geometrická interpretace vzdálenosti:

detekce:

$$dch < kvzd$$

 dch ... detekovatelné chyby

oprava:

$$dch + och < kvzd$$

 och ... opravitelné chyby

$$och \leq dch$$

Př.: $kvzd = 4$

dch	3	2
och	0	1
1 ch.	D	K
2 ch.	D	D
3 ch.	D	#
4 ch.	N	N

D — detekce
(bez korekce)

K — korekce

N — nic

— špatná
korekce

S E D
 ↓ ↓ ↓
 Detecting
 Correcting
 Error
 Single
 Double
 Triple
 Quadruple

 $kvzd$

2 SED

3 SEC nebo DED

4 SEC-DED nebo TED

⋮

Jednoduché bezpečnostní kódy**[1] opakovací kód $(n, 1)$:**

$$b_1 = \dots = b_n = a_1 \quad kvzd = n$$

velká redundance $\Rightarrow$ malý informační obsah**[2] „kótvový“ kód (jk, k) :** j krát se opakuje totéž (k bitů)velká redundance $\Rightarrow$ malý informační obsah**[3] parita $(k+1, k)$:**

$$\underline{b} = (a_1, \dots, a_k, p)$$

$$\text{parita} \begin{cases} \text{sudá: } p = a_1 \oplus \dots \oplus a_k \\ \text{lichá: } p = a_1 \oplus \dots \oplus a_k \oplus 1 \end{cases}$$

minimální redundance $\times$ pouze detekce 1 chyby**[4] příčná a podélná parita:**Př.: $k = 4 = 2 \times 2$

a_1	a_2	p_1
a_3	a_4	p_2
p_3	p_4	

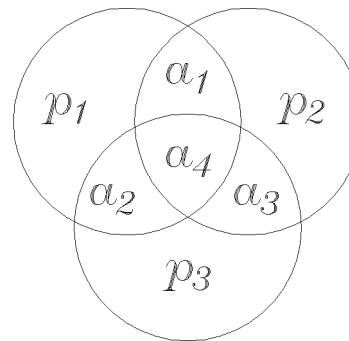
$$kvzd = 3$$

kód (8,4)

a_1	a_2	p_1
a_3	a_4	p_2
p_3	p_4	p_5

$$kvzd = 4$$

kód (9,4)

jeden Hammingův kód — kód (7,4)

$$p_1 = a_1 \oplus a_2 \oplus a_4$$

$$p_2 = a_1 \oplus a_3 \oplus a_4$$

$$p_3 = a_2 \oplus a_3 \oplus a_4$$

 $kvzd = 3 \Rightarrow$ SEC (nebo DED)

b_1	b_2	b_3	b_4	b_5	b_6	b_7
a_1	a_2	a_3	a_4	p_1	p_2	p_3

b_1	b_2	b_3	b_4	b_5	b_6	b_7
p_1	p_2	a_1	p_3	a_2	a_3	a_4
			x	x	x	x
	x	x			x	x
x		x		x		x

$$\begin{aligned}
 b_1 &= a_1 \oplus a_2 \oplus a_4 \\
 b_2 &= a_1 \oplus a_3 \oplus a_4 \\
 b_3 &= a_1 \\
 b_4 &= a_2 \oplus a_3 \oplus a_4 \\
 b_5 &= a_2 \\
 b_6 &= a_3 \\
 b_7 &= a_4
 \end{aligned}$$

$$\begin{aligned}
 b_1 &= b_3 \oplus b_5 \oplus b_7 \\
 b_2 &= b_3 \oplus b_6 \oplus b_7 \\
 b_4 &= b_5 \oplus b_6 \oplus b_7
 \end{aligned}$$

žádná chyba $\implies c_i = b_i$

např.: $c_1 = c_3 \oplus c_5 \oplus c_7$

$\implies c_1 \oplus c_3 \oplus c_5 \oplus c_7 = 0$

$$\begin{aligned}
 s_3 &= c_1 \oplus c_3 \oplus c_5 \oplus c_7 \\
 s_2 &= c_2 \oplus c_3 \oplus c_6 \oplus c_7 \\
 s_1 &= c_4 \oplus c_5 \oplus c_6 \oplus c_7
 \end{aligned}$$

<u>e</u>	<u>s</u>
0000000	000
1000000	001
...	...
0000010	110
0000001	111

$$\underline{b} = \underline{a} \cdot \underline{G}$$

$$\underline{G} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

generovací matice

$$\underline{s} = \underline{c} \cdot \underline{H}^T$$

$$\underline{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

kontrolní matice

Řádky každé z matic tvoří bázi tzv. vektorového neboli lineárního prostoru.

domácí úkol: Najděte generovací a kontrolní matici pro dříve uvedené kódy! Pro lichou paritu a pro kód $\mathcal{K}_2$ by se Vám to však nemělo podařit.

Má vůbec smysl používat bezpečnostní kódy ???

- SED: polovina chyb ???
- SEC: n z $2^n - 1$ chyb ???
- : : : ???

Pravděpodobnost:

P ... pravděpodobnost výskytu chybného bitu

n bitů i chybných: $\binom{n}{i} \cdot P^i \cdot (1 - P)^{n-i}$
jinak: $\sum$

Př.:

$$P = 10^{-9} \quad 32\text{b} / 1\mu\text{sec}$$

nějaká chyba: $32 \cdot 10^{-9} \dots 31 \text{ sec}$

SED (33,32):

nějaká chyba: $33 \cdot 10^{-9}$

sudý počet chyb: $5,3 \cdot 10^{-16} \dots 61,3 \text{ roků}$

SEC (38,32):

nějaká chyba: $38 \cdot 10^{-9}$

neopravitelná chyba: $7 \cdot 10^{-16} \dots 46,0 \text{ roků}$

vektorový / lineární prostor

množina všech slov dvojkového blokového kódu

& operace:

vektorový neboli lineární prostor V
se skalárním součinem
nad tělesem $GT(2)$

- Galoisovo (konečné) 2prvkové těleso [field]

$GT(2)$

$\{0,1\}$	$ \begin{array}{c cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 0 \end{array} $	$ \begin{array}{c cc} \cdot & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array} $
	XOR	AND

$$x+t=y \Leftrightarrow x=y+t \Rightarrow \boxed{\text{odčítání} \equiv \text{sčítání}}$$

- $t \in GT(2) \dots$ skalár
- $\underline{v} = (v_1, \dots, v_n) \in V \dots$ vektor
- operace:
součet vektorů — Abelova (komutativní) grupa
násobení vektoru skalárem: $\underline{0} \cdot \underline{v} = \underline{0}$
 $1 \cdot \underline{v} = \underline{v}$
o asociativní zákon: $(s \cdot t) \cdot \underline{v} = s \cdot (t \cdot \underline{v})$
o distributivní zákony: $(s+t) \cdot \underline{v} = s \cdot \underline{v} + t \cdot \underline{v}$
 $t \cdot (\underline{u} + \underline{v}) = t \cdot \underline{u} + t \cdot \underline{v}$

- skalární součin vektorů $\underline{u}$ a $\underline{v}$:
 $(u_1, \dots, u_n) \cdot (v_1, \dots, v_n) = u_1 v_1 + \dots + u_n v_n$
př: $01011 \cdot 11101 = 0+1+0+0+1 = 0$
- ortogonální vektory: $\underline{u} \perp \underline{v} \Leftrightarrow \underline{u} \cdot \underline{v} = 0$
př: $01011 \perp 11101$

Podprostor W prostoru V :
• $W \subset V$
• uzavřenost

Podprostor W je sám prostorem.

- Všechny lineární kombinace vektorů $\underline{u}, \dots, \underline{v}$ tvoří vektorový (lineární) prostor.
- Vektory lineárně nezávislé $\Rightarrow$ báze (prostoru)
- Stejný počet vektorů každé báze $\rightarrow$ dimenze

ortogonální doplněk U k prostoru V : $U \perp V$

- $\underline{u} \in U \Leftrightarrow (\forall \underline{v} \in U) (\forall \underline{v} \in V) \underline{u} \perp \underline{v}$
- U je prostor, tzv. nulový prostor prostoru V
- $U \perp V \Leftrightarrow V \perp U$
- ortogonalita prostorů $\Leftrightarrow$ ortogonalita bází
- $\underline{v} \in V \Leftrightarrow (\forall \underline{u} \in U) \underline{u} \perp \underline{v}$

Lineární kódy

kódová slova tvoří lineární prostor

popis lineárních kódů: matice

grupové kódy — dvojkové (?) lineární kódy

$\underline{b} = \underline{a} \cdot \underline{G} = a_1 \cdot \underline{g}_1 \oplus \dots \oplus a_k \cdot \underline{g}_k$
 $\underline{b} = \text{lin. komb. } \underline{g}_1, \dots, \underline{g}_k \quad \underline{g}_i = \text{itý řádek } \underline{G}$
všechna $\underline{b} = \underline{a} \cdot \underline{G} \rightarrow$ lineární prostor
báze: $\underline{g}_1, \dots, \underline{g}_k$

řádky $\underline{G}$ lin. nezávislé (Proč?)

$\underline{G} \rightarrow$ prostor V

$\underline{H} \rightarrow$ prostor $U \Leftarrow V \perp U$

řádky $\underline{H}$ lin. nezávislé (Proč?)

$$\underline{G} \cdot \underline{H}^T = \underline{0}$$

$$\underline{G}: k \times n, \underline{H}: m \times n \Rightarrow m + k = n$$

$$\underline{c} \in V \Leftrightarrow (\forall \underline{u} \in U) \underline{c} \cdot \underline{u} = 0$$

$$\underline{c} \in V \Leftrightarrow \underline{c} \cdot \underline{H}^T = \underline{0}$$

$$\underline{b} = \underline{a} \cdot \underline{G} \quad \underline{b} \cdot \underline{H}^T = \underline{a} \cdot \underline{G} \cdot \underline{H}^T = \underline{0}$$

$$\underline{s} = \underline{c} \cdot \underline{H}^T \quad \underline{c} = \underline{b} + \underline{e}$$

$$\underline{s} = \underline{e} \cdot \underline{H}^T \Rightarrow \text{syndrom závisí pouze na chybě}$$

chyba v pozicích	$\underline{e}$	$\underline{s}^T$
i	$0 \dots 010 \dots 0$	i -tý sloupec $\underline{H}$
$i \text{ a } j$	$0 \dots 010 \dots 010 \dots 0$	i -tý + j -tý sloupec $\underline{H}$
$\vdots$	$\vdots$	$\vdots$

$\xi =$ počet lin. nezávislých sloupců matice $\underline{H}$

ξ	dch	kvzd
1	1	2
2	2	3
3	3	4
$\vdots$	$\vdots$	$\vdots$

$$kvzd = \xi + 1$$

$$\underline{G} = (\underline{I}_k, \underline{F})$$

$$\underline{H} = (\underline{F}^T, \underline{I}_m)$$

systematický kód

 $\underline{I}_j \dots$ jednotková matice $j \times j$

elementární operace s řádky:

- záměna řádků
- přičtení lineární kombinace jiných řádků
 - o (násobení řádku nenulovým skalárem)

 $\underline{G} \rightarrow \underline{G}'$ táž množina kódových slov $\underline{H} \rightarrow \underline{H}'$ jiná (vyhovující) kontrolní matice

!!! syndromy mají jiný „význam“ !!!

$$\left. \begin{array}{l} \underline{G} = \dots \\ \underline{H} = \dots \end{array} \right\} \begin{array}{l} \underline{H} = ? \\ \underline{G} = ? \end{array} \left\{ \begin{array}{l} \text{elementární operace} + \\ \text{+ výše uvedené vztahy} \end{array} \right.$$

- Nelze-li $\uparrow$:
- permutace sloupců
 - nalezení matice
 - opačná permutace sloupců



Př.:

oprava 1 chyby $\Rightarrow$ vzájemně různé sloupce $\underline{H}$:

$$\underline{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

elementární operace: $1 + 2 \rightarrow 1'$ $3 + 1' \rightarrow 3'$ $2 + 3' \rightarrow 2'$

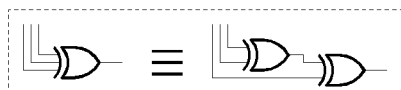
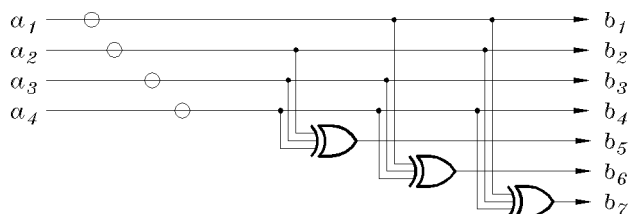
$$\underline{H}' = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$\underline{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Hammingův kód (7,4) — trochu jiný

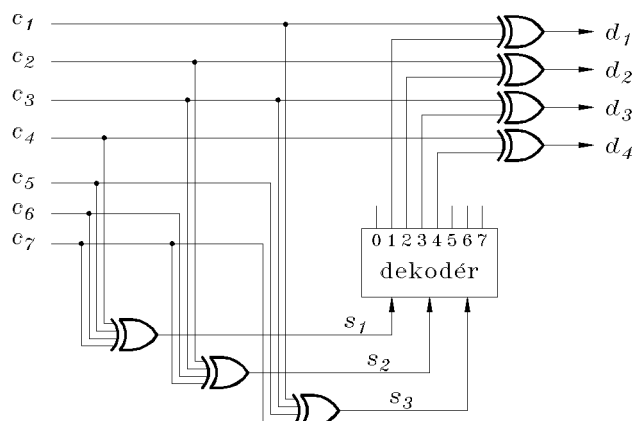
kodér

$$\underline{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$



dekodér

$$\underline{H} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$



kódová vzdálenost:

$$\text{vzd}(\underline{b}', \underline{b}'') = \text{váha}(\underline{b}' \oplus \underline{b}'')$$

lin. kód $\Rightarrow \underline{b}' \oplus \underline{b}''$ je kódové slovo

$$\boxed{\text{lin. kód} \Rightarrow kvzd = \text{minim. váha} \neq 0}$$

lin. kód $\mathcal{K}$ — lichá kódová vzdálenost

lin. kód $\mathcal{K}^*$ — kód $\mathcal{K}$ + parita (sudá)

$$\boxed{kvzd(\mathcal{K}^*) = kvzd(\mathcal{K}) + 1}$$

Př.: Hamm. k.: $kvzd = 3$

Hamm. k. + parita: $kvzd = 4$

↓

rozšířený Hammingův kód

Př.: příčná a podélná parita

analogicky $kvzd: 3 \rightarrow 4$

Hammingovy kódy

perfektní kódy SEC — všechny syndromy „využity“

$$kvzd = 3 \quad n = 2^m - 1 \quad k = n - m$$

m	2	3	4	5	...
kód	(3,1)	(7,4)	(15,11)	(31,26)	...

odvozené kódy: menší $k \Rightarrow n$; např. kódy (12,8), (21,16), (38,32), ...

rozšířené Hammingovy kódy

kvaziperfektní kódy SEC–DED — všechny syndromy „využity“

Př.:

$$\underline{H} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

$$\underline{s} = \text{x x x } 1 \Rightarrow 1 \text{ chyba}$$

$$\underline{s} = \text{x x x } 0 \Rightarrow 2 \text{ chyby} \quad (\text{kromě } \underline{s} = 0000) \\ (\text{žádná chyba})$$

$$\text{obecně: } kvzd = 4 \quad n = 2^m - 1 \quad k = n - m$$

m	3	4	5	6	...
kód	(4,1)	(8,4)	(16,11)	(32,26)	...

kódy RM (Reed – Muller)parametry: ρ a μ jčlenné logické součiny μ proměnných (pomocných)Př.: $\mu = 3$

	j	součiny	
${}^0\underline{\mathbf{G}}$	0	1 1 1 1 1 1 1 1	1
${}^1\underline{\mathbf{G}}$	1	0 0 0 0 1 1 1 1	ζ
		0 0 1 1 0 0 1 1	η
		0 1 0 1 0 1 0 1	ϑ
${}^2\underline{\mathbf{G}}$	2	0 0 0 0 0 0 1 1	$\zeta \cdot \eta$
		0 0 0 0 0 1 0 1	$\zeta \cdot \vartheta$
		0 0 0 1 0 0 0 1	$\eta \cdot \vartheta$
${}^3\underline{\mathbf{G}}$	3	0 0 0 0 0 0 0 1	$\zeta \cdot \eta \cdot \vartheta$

ρ	0	1	2	...
$\underline{\mathbf{G}}$	${}^0\underline{\mathbf{G}}$	$\begin{pmatrix} {}^0\underline{\mathbf{G}} \\ {}^1\underline{\mathbf{G}} \end{pmatrix}$	$\begin{pmatrix} {}^0\underline{\mathbf{G}} \\ {}^1\underline{\mathbf{G}} \\ {}^2\underline{\mathbf{G}} \end{pmatrix}$	...

$$\boxed{kvzd = 2^\mu - \rho} \quad n = 2^\mu$$

$k = \text{počet řádků } \underline{\mathbf{G}}$

Př.: RM (1, 3)

$$\mu = 3, \quad \rho = 1, \quad kvzd = 4$$

$$n = 8, \quad k = 4, \quad \text{kód (8,4) — SEC-DED}$$

$$\underline{\mathbf{G}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$$\begin{aligned} b_1 &= a_1 \\ b_2 &= a_1 + a_4 \\ b_3 &= a_1 + a_3 \\ b_4 &= a_1 + a_3 + a_4 \\ b_5 &= a_1 + a_2 \\ b_6 &= a_1 + a_2 + a_4 \\ b_7 &= a_1 + a_2 + a_3 \\ b_8 &= a_1 + a_2 + a_3 + a_4 \end{aligned}$$

$$\begin{aligned} a_4 &= b_1 + b_2 = b_3 + b_4 = \\ &= b_5 + b_6 = b_7 + b_8 \\ a_3 &= b_1 + b_3 = b_2 + b_4 = \\ &= b_5 + b_7 = b_6 + b_8 \\ a_2 &= b_1 + b_5 = b_2 + b_6 = \\ &= b_3 + b_7 = b_4 + b_8 \end{aligned}$$

příklad – pokračování

$$\underline{\mathbf{a}} = (a_1, a_2, a_3, a_4) = ({}^0\underline{\mathbf{a}}, {}^1\underline{\mathbf{a}}), \text{ kde}$$

$${}^0\underline{\mathbf{a}} = (a_1) \quad \text{a} \quad {}^1\underline{\mathbf{a}} = (a_2, a_3, a_4)$$

$$\underline{\mathbf{b}} = \underline{\mathbf{a}} \cdot \underline{\mathbf{G}} = ({}^0\underline{\mathbf{a}}, {}^1\underline{\mathbf{a}}) \cdot \begin{pmatrix} {}^0\underline{\mathbf{G}} \\ {}^1\underline{\mathbf{G}} \end{pmatrix} = {}^0\underline{\mathbf{a}} \cdot {}^0\underline{\mathbf{G}} + {}^1\underline{\mathbf{a}} \cdot {}^1\underline{\mathbf{G}}$$

$$\underline{\mathbf{b}}' = {}^0\underline{\mathbf{a}} \cdot {}^0\underline{\mathbf{G}} = \underline{\mathbf{b}} - {}^1\underline{\mathbf{a}} \cdot {}^1\underline{\mathbf{G}}$$

$$\underline{\mathbf{b}}' = (b'_1, b'_2, \dots, b'_8) = a_1 \cdot (11\dots 1)$$

$$\underline{\mathbf{b}}' = (b'_1, b'_2, \dots, b'_8) = (a_1, a_1, \dots, a_1)$$

$$a_1 = b'_1 = b'_2 = \dots = b'_8$$

Př.: $\rho = 2$

$$\underline{\mathbf{b}} = \underline{\mathbf{a}} \cdot \underline{\mathbf{G}} = ({}^0\underline{\mathbf{a}}, {}^1\underline{\mathbf{a}}, {}^2\underline{\mathbf{a}}) \cdot \begin{pmatrix} {}^0\underline{\mathbf{G}} \\ {}^1\underline{\mathbf{G}} \\ {}^2\underline{\mathbf{G}} \end{pmatrix} =$$

$$= {}^0\underline{\mathbf{a}} \cdot {}^0\underline{\mathbf{G}} + {}^1\underline{\mathbf{a}} \cdot {}^1\underline{\mathbf{G}} + {}^2\underline{\mathbf{a}} \cdot {}^2\underline{\mathbf{G}}$$

1. Určit ${}^2\underline{\mathbf{a}}$.
2. Určit $\underline{\mathbf{b}}' = \underline{\mathbf{b}} - {}^2\underline{\mathbf{a}} \cdot {}^2\underline{\mathbf{G}}$.
3. Určit ${}^1\underline{\mathbf{a}}$.
4. Určit $\underline{\mathbf{b}}'' = \underline{\mathbf{b}}' - {}^1\underline{\mathbf{a}} \cdot {}^1\underline{\mathbf{G}}$.
5. Určit ${}^0\underline{\mathbf{a}} = a_1$.

Některé kódy RM

ρ	μ	n	k	m	$kvzd$
1	4	16	5	11	8
1	5	32	6	26	16
2	5	32	16	16	8
1	6	64	7	57	32
2	6	64	22	42	16
3	6	64	42	22	8

Kódy BCH

Konečná (Galoisova) tělesa

Pro každé prvočíslo p a pro každé přirozené $j > 0$ existuje právě jedno konečné těleso $GT(p^j)$, které má p^j prvků. !!! Jiná konečná tělesa neexistují.

Př.: $GT(4) = GT(2^2)$

+	♦	♥	♣	♠
♦	♦	♥	♣	♠
♥	♥	♦	♠	♣
♣	♣	♠	♦	♥
♠	♠	♣	♥	♦

·	♦	♥	♣	♠
♦	♦	♦	♦	♦
♥	♥	♦	♣	♠
♣	♣	♠	♣	♥
♠	♠	♣	♥	♣

např.: ♣ + ♥ = ♠ nebo ♣ · ♥ = ♣

♦ ... nula (♦ = 0)

♥ ... jednička (♥ = 1)

V každém konečném tělese existuje primitivní prvek

α , tzn. takový prvek α , že $(\forall \beta \neq 0) (\exists i) \beta = \alpha^i$

Př.: $\alpha' = \clubsuit$

$$\clubsuit = \clubsuit^1 = \clubsuit$$

$$\spadesuit = \clubsuit^2 = \clubsuit \cdot \clubsuit$$

$$\heartsuit = \clubsuit^3 = \clubsuit \cdot \clubsuit \cdot \clubsuit$$

$\alpha'' = \spadesuit$

$$\spadesuit = \spadesuit^1 = \spadesuit$$

$$\clubsuit = \spadesuit^2 = \spadesuit \cdot \spadesuit$$

$$\heartsuit = \spadesuit^3 = \spadesuit \cdot \spadesuit \cdot \spadesuit$$

Př.:

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

·	00	01	10	11
00	00	00	00	00
01	01	00	01	10
10	10	00	10	11
11	11	00	11	10

$\alpha = 10 = \alpha'$ (lze však použít také $\alpha'' = 11$)

$\alpha^1 = 10$ $\alpha^2 = 11$ $\alpha^3 = 01 = \alpha^0$

Sčítání: XOR např.: $10 + 11 = 01$

Násobení: $\alpha^i \cdot \alpha^j = \alpha^{i+j}$, např.

$$10 \cdot 11 = \alpha^1 \cdot \alpha^2 = \alpha^3 = \alpha^0 = 01$$

Př.: $GT(16) = GT(2^4)$

i	α^i	i	α^i	i	α^i	i	α^i
0	0001	4	0011	8	0101	12	1111
1	0010	5	0110	9	1010	13	1101
2	0100	6	1100	10	0111	14	1001
3	1000	7	1011	11	1110	15	0001

kódy BCH

(Bose – Chaudhuri – Hoquenghem)

(zjednodušeno)

kódy (n, k) pro opravu t chyb (tj. kód t EC)

$$n = 2^m - 1 \quad \text{a} \quad m = n - k$$

α je primitivní prvek tělesa $GT(2^m)$

$$\underline{H} = \begin{pmatrix} \alpha^1 & \alpha^2 & \dots & \alpha^n \end{pmatrix} \quad kvzd = 3 \dots \text{SEC}$$

$$\underline{H} = \begin{pmatrix} \alpha^3 & \alpha^6 & \dots & \alpha^{3n} \\ \alpha^1 & \alpha^2 & \dots & \alpha^n \end{pmatrix} \quad kvzd = 5 \dots \text{DEC}$$

$$\underline{H} = \begin{pmatrix} \alpha^5 & \alpha^{10} & \dots & \alpha^{5n} \\ \alpha^3 & \alpha^6 & \dots & \alpha^{3n} \\ \alpha^1 & \alpha^2 & \dots & \alpha^n \end{pmatrix} \quad kvzd = 7 \dots \text{TEC}$$

...

Př.: $n = 31 \Rightarrow GT(32) \Rightarrow$ „5bitové“ prvky

$$kvzd = 3 \Rightarrow m = 5 \Rightarrow \text{kód } (31, 26)$$

$$kvzd = 5 \Rightarrow m = 10 \Rightarrow \text{kód } (31, 21)$$

atd.

Pozn.: Uvedená kódová vzdálenost je tzv. zaručená.

Skutečná kódová vzdálenost může být větší.

Př.: kód(15,7)

$$\underline{H} = \begin{pmatrix} 1111 & 0111 & 1011 & 1110 \\ 0101 & 0010 & 1001 & 0110 \\ 0011 & 0001 & 1000 & 1110 \\ 0001 & 1000 & 1100 & 0111 \\ 0010 & 0110 & 1011 & 1110 \\ 0100 & 1101 & 0111 & 1001 \\ 1001 & 1010 & 1111 & 0001 \\ 0001 & 0011 & 0101 & 1111 \end{pmatrix}$$

Rozšířené kódy BCH

parita navíc $\Rightarrow$ oprava t chyb $\subset$ detekce $t+1$ chyb

(srov. rozšířený Hammingův kód)

Některé rozšířené kódy BCH

kvzd	n	k	n	k	n	k
6	16	7	32	21	64	51
8	16	5 (!)	32	16	64	45
10					64	39
12			32	11 (!)	64	36 (!)
14					64	30 (!)
16			32	6 (!)	64	24 (!)
22					64	18 (!)
24					64	16 (!)
28					64	10 (!)
32					64	7 (!)

Kódy a mnohočleny

mnohočleny nad tělesem GT(2):

Př.: $C(x) = c_{n-1}x^{n-1} + \dots + c_1x + c_0$
 $c_i \in \text{GT}(2) \quad x \in ? \leftarrow !!!$

Mnohočleny stupně $< n$:

- vektorový/lineární prostor
(bez skalárního součinu)
- izomorfní s prostorem uspořádaných n tic:
 $c_{n-1}x^{n-1} + \dots + c_0 \sim (c_{n-1}, \dots, c_0)$
 př.: $n = 7 \quad x^5 + x^3 + x + 1 \sim 0101011$
- izomorfie — jen: součet & násobení skalárem

Mnohočleny:

- násobení a dělení mnohočlenů — obv. postup
- analogie s okruhem celých čísel — dělitelnost

Násobení mnohočlenů

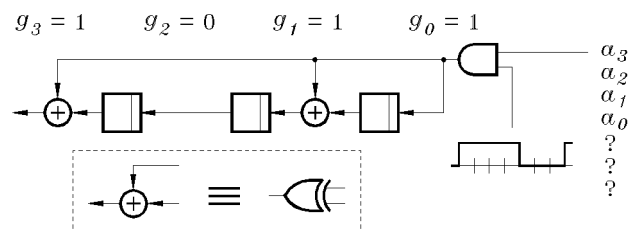
Př.: $B(x) = G(x) \cdot A(x) = (x^3 + x + 1) \cdot (x^3 + x)$

$$\begin{array}{r} 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 \\ 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 \\ 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x \\ 0 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x + 0 \\ \hline 1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x + 0 = B(x) \end{array} \quad \begin{array}{l} G(x) \cdot x^3 \\ G(x) \cdot 0 \\ G(x) \cdot x \\ G(x) \cdot 0 \end{array}$$

„zkrácený zápis“: $1 \ 0 \ 1 \ 1 \cdot 1 \ 0 \ 1 \ 0$

$$\begin{array}{r} 1 \ 0 \ 1 \ 1 \\ 0 \ 0 \ 0 \ 0 \\ 1 \ 0 \ 1 \ 1 \\ 0 \ 0 \ 0 \ 0 \\ \hline 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \end{array} \sim \underline{b} = \underline{g} \cdot \underline{a}$$

(zde $\cdot$ neoznačuje skalární součin!)



Dělení mnohočlenů

Př.: $C(x) = x^6 + x^2 + x \quad G(x) = x^3 + x + 1$

$$\begin{array}{r} 1 \cdot x^6 + 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x + 0 \Rightarrow 1 \cdot x^3 \\ 1 \cdot x^6 + 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 \\ \hline 0 \cdot x^5 + 1 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 \Rightarrow 0 \cdot x^2 \\ 0 \cdot x^5 + 0 \cdot x^4 + 0 \cdot x^3 + 0 \cdot x^2 \\ \hline 1 \cdot x^4 + 1 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x \Rightarrow 1 \cdot x \\ 1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x \\ \hline 1 \cdot x^3 + 0 \cdot x^2 + 0 \cdot x + 0 \Rightarrow 1 \\ 1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 \\ \hline 0 \cdot x^2 + 1 \cdot x + 1 \end{array}$$

zbytek $C(x) \% G(x) = x + 1$

podíl $C(x) \div G(x) = x^3 + x + 1 \leftarrow$

„zkrácený zápis“: $6 \ 5 \ 4 \ 3 \ 2 \ 1 \ 0$

$C(x) \sim 1000110$

$G(x) \sim 1011$

$C(x) \% G(x) \sim 1011$

$C(x) \div G(x) \sim 011$

Dělení mnohočlenů

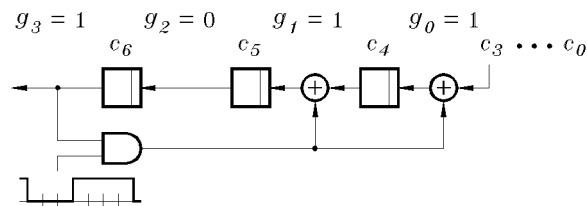
2

Př.: $C(x) : G(x)$

$C(x) = x^6 + x^2 + x = c_6x^6 + \dots + c_0$

$G(x) = x^3 + x + 1 = g_3x^3 + \dots + g_0$

$$\begin{array}{r} 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 : 1 \ 0 \ 1 \ 1 = 1 \ 0 \ 1 \ 1 \\ 1 \ 0 \ 1 \ 1 \downarrow \\ \hline 0 \ 0 \ 1 \ 1 \ 1 \downarrow \\ 0 \ 0 \ 0 \ 0 \downarrow \\ \hline 0 \ 1 \ 1 \ 1 \ 1 \downarrow \\ 1 \ 0 \ 1 \ 1 \downarrow \\ \hline 0 \ 1 \ 0 \ 0 \ 0 \\ 1 \ 0 \ 1 \ 1 \\ \hline 0 \ 0 \ 1 \ 1 \end{array}$$



Dělitelnost mnohočlenůdeg $P(x)$ — stupeň mnohočlenu $P(x)$

deg $0 = -\infty$ (?)

deg $(P(x) \cdot G(x)) = \deg P(x) + \deg G(x)$

$G(x) \neq 0$

deg $(P(x) \% G(x)) < \deg G(x)$

deg $(P(x) \div G(x)) = \deg P(x) - \deg G(x)$

$$P(x) \% G(x) = 0 \iff \boxed{G(x) \mid P(x)}$$

$$\neq \iff \boxed{G(x) \nmid P(x)}$$

$$G(x) \mid P(x) \iff \exists Y(x) \quad P(x) = G(x) \cdot Y(x)$$

$G(x)$ je dělitelem $P(x)$

nerozložitelný (ireducibilní) mnohočlen $P(x)$:

nemá jiné dělitele než 1 a $P(x)$ pro $GT(2)$!
obdobu prvočísel

rozklad na prvočinitele je jednoznačný
prvočinitel = nerozložitelný mnohočlen

Př.: $x^7 + 1 = (x^3 + x^2 + 1) \cdot (x^3 + x + 1) \cdot (x + 1)$

Kódy generované mnohočlenem

$$\boxed{G(x) = x^m + g_{m-1}x^{m-1} + \dots + g_1x + 1}$$

$m = \deg G(x) > 0$

 $G(x)$... generovací mnohočlen

$A(x) \sim \underline{a} \dots k \text{ bitů} \implies k-1 \geq \deg A(x)$

$$\boxed{B(x) = A(x) \cdot G(x)}$$

$B(x) \sim \underline{b} \dots n \text{ bitů} \implies n-1 \geq \deg B(x)$

$$\boxed{n = k + m}$$

 $m = \deg G(x)$... redundance

Př.: $G(x) = x^3 + x + 1$ $\underline{a} = 0101$ $\underline{b} = 0100111$
(viz př. na násobení mnohočlenů)

Pozorování: $x \mid G(x) \implies B(x) = B'(x) \cdot x$
 $\implies \underline{b} = x \dots x0$

Závěr: musí platit $\boxed{x \nmid G(x)}$

Pozorování:

$$A(x) = a_{k-1}x^{k-1} + \dots + a_0$$

$$A(x) \cdot G(x) = a_{k-1} \cdot G(x) \cdot x^{k-1} + \dots + a_0 \cdot G(x)$$

$G(x) \cdot x^{k-1}, \dots, G(x)$ — lineárně nezávislé
— báze lineárního prostoru

 $A(x) \cdot G(x)$ — všechny lineární kombinaceZávěr: Kód generovaný mnohočlenem je lineární

$$\underline{G} = \begin{pmatrix} g_m & g_{m-1} & \dots & 0 \\ 0 & g_m & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & g_0 \end{pmatrix}$$

Př.: $G(x) = x^3 + x + 1$

$$\underline{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Úkol: Ověřte, že kódy generované mnohočlenem $G(x)$ a maticí $\underline{G}$ jsou stejné. Najděte kontrolní matici $\underline{H}$ a všimněte si, že je to matice Hammingova kódu $(7,4)$.

$B(x) \sim \underline{b} \dots$ vyslané slovo

$C(x) \sim \underline{c} \dots$ přijaté slovo

$E(x) \sim \underline{e} \dots$ chybový mnohočlen

$C(x) = B(x) + E(x)$

$$\boxed{S(x) = C(x) \% G(x)} \dots$$
 syndrom

$B(x) \% G(x) = 0$

$S(x) = E(x) \% G(x)$

syndrom závisí pouze na chybách

$E(x) = E'(x) \cdot x^j \quad x \nmid E'(x)$

$E(x) \dots$ shluk chyb délky $\boxed{l = (\deg E'(x)) + 1}$

Př.: $E(x) \sim 0101100 \dots$ shluk chyb délky 4

$E'(x) \sim 01011 \quad j = 2$

$\deg E'(x) = 3$

detekce chyb: $S(x) = E(x) \% G(x) = 0$?

$x \nmid G(x) \implies x^j$ nemá vliv

$E'(x) \% G(x) = 0$?

$$E'(x) \neq 0 \quad \& \quad \deg E'(x) < \deg G(x) \implies$$

$$\implies E'(x) \% G(x) \neq 0, \text{ tedy:}$$

detekce shluků chyb délky $l \leq m = \deg G(x)$

Př.: $G(x) = x^{16} + 1 \implies l \leq 16$

Cyklické kódy

(1) lineární kódy

(2) cyklický posuv kódového slova $\rightarrow$ kódové slovoPř.: $\mathcal{K}_1$ — cyklický (viz B. k. • 2) $\mathcal{K}_2$ — nesplňuje (1) $\mathcal{K}_3$ — nesplňuje (2)dále předp.: $G(x) \mid (x^n - 1)$, tzn.:

$$\exists H(x) \quad \boxed{G(x) \cdot H(x) = x^n - 1}$$

 $H(x)$ — kontrolní mnohočlen

(má podstatně menší význam než kontrolní matice)

Pozorování:

$$B(x) = b_{n-1}x^{n-1} + \dots + b_0 = G(x) \cdot A(x)$$

cykl. pos. vlevo:

$$\begin{aligned} B'(x) &= B(x) \cdot x - b_{n-1} \cdot x^n + b_{n-1} = \\ &= B(x) \cdot x - b_{n-1} \cdot (x^n - 1) = \\ &= A(x) \cdot G(x) \cdot x - b_{n-1} \cdot G(x) \cdot H(x) = \\ &= [A(x) \cdot x + b_{n-1} \cdot H(x)] \cdot G(x) \end{aligned}$$

Závěr: Mnohočlen $G(x)$ generuje cyklický kódPř.: $G(x) = x^3 + x + 1 \quad G(x) \mid (x^7 - 1)$
kód (7,4) generovaný $G(x)$ je cyklický $\mathcal{K}$ — cyklický kód $\implies \exists \mathcal{K}'$:

- kódy $\mathcal{K}$ a $\mathcal{K}'$: stejná množina kódových slov
- kód $\mathcal{K}'$: generován mnohočlenem $G(x)$
- $G(x) \neq 0 \dots$ nejnižší stupeň

Př.: kód $\mathcal{K}_1$ je cyklický (viz B. k. • 2)kód $\mathcal{K}_1'$: $G(x) = x + 1$

táž množina kódových slov:

{ 000, 011, 101, 110 }

Systematické cyklické kódy $B(x) = A(x) \cdot G(x)$ — kód není systematický
(existují výjimky)

Systematický kód:

$$\boxed{B(x) = A(x) \cdot x^m + A(x) \cdot x^{m \% G(x)}},$$

kde $m = \deg G(x)$

Př.: $G(x) = x^3 + x + 1$

$$A(x) \sim 0101$$

$$A(x) \cdot x^3 \sim 0101000$$

$$A(x) \cdot x^3 \% G(x) \sim 100$$

$$B(x) \sim 0101100$$

!!! Množina kódových slov je u obou kódů stejná !!!

Některé cyklické kódy:

- $G(x) = x + 1$

$$B(x) = G(x) \cdot A(x)$$

$$x = 1 \implies G(x) = G(1) = 0 \implies B(x) = 0$$

$$\implies B(x) = b_{n-1} + \dots + b_0 = 0$$

$$\implies \boxed{\text{parita (sudá)}}$$

- $G(x) = x^m + 1$

$$x^m \equiv 1 \pmod{G(x)} \quad \dots \quad \text{kongruence}$$

stejný zbytek po dělení

$$C(x) = C_j(x) \cdot (x^m)^j + \dots + C_0(x) \cdot (x^m)^0$$

$$C(x) \equiv C_j(x) + \dots + C_0(x) \pmod{G(x)}$$

$$\implies \boxed{\text{„podélná“ parita } m\text{tic bitů (sudá)}}$$

- Hammingovy kódy např.:

$$G(x) = x^3 + x + 1 \quad \text{kód (7,4)}$$

$$G(x) = x^4 + x + 1 \quad \text{kód (15,11)}$$

$$G(x) = x^5 + x^2 + 1 \quad \text{kód (31,26)}$$

- BCH kódy (viz lineární kódy)

- RS kódy Reed – Solomon

- Fireovy kódy

řád r mnohočlenu $P(x)$ je nejmenší takové r , že

$$P(x) \mid (x^r - 1)$$

Př.: řád x^3+x+1 je roven 7 (ale stupeň je roven 3)

$$x^i - 1 \mid x^j - 1 \iff i \mid j$$

- shluky chyb $E_1(x)$ a $E_2(x)$
 $E_1(x) = E'(x) \cdot x^i \quad x \nmid E'(x)$
 $E_2(x) = E''(x) \cdot x^j \quad x \nmid E''(x)$
 $\deg E'(x) + \deg E''(x) < q$
 $j \geq i$

otázka: Kdy $E_1(x) \equiv E_2(x) \pmod{x^q+1}$

úvahy: $E_1(x) - E_2(x) \equiv 0 \pmod{x^q+1}$
 $(E'(x) + E''(x) \cdot x^{j-i}) \cdot x^i \equiv 0 \pmod{x^q+1}$
 x^i neobsahuje prvočinitele x^q+1

$$E'(x) + E''(x) \cdot x^{j-i} \equiv 0 \pmod{x^q+1}$$

odpověď: Právě když $E'(x) = E''(x)$ a $q \mid j-i$

- $E^*(x) = E'(x) = E''(x)$
 $\deg E^*(x) < \deg P(x)$ a $P(x)$ má řád r

otázka: Kdy $E^*(x) \cdot (x^{j-i} + 1) \equiv 0 \pmod{P(x)}$

odpověď: Právě když $r \mid j-i$

Fireovy kódy

$P(x)$... nerozložitelný mnohočlen řádu r

$Q(x) = x^q + 1 \rightarrow q$ a r nesoudělné $\leftrightarrow$

$p = \deg P(x) \quad q = \deg Q(x)$

$n = \text{NSN}(q, r)$ nejmenší společný násobek

$$G(x) = P(x) \cdot Q(x)$$

generovací mnohočlen n bitového Fireova kódu

$$l_1 \leq l_2 \begin{cases} l_1 + l_2 \leq q+1 \\ l_1 \leq p \end{cases}$$

$$\begin{array}{l} \text{detekce shluků chyb délky} \leq l_2 \\ \text{oprava shluků chyb délky} \leq l_1 \end{array}$$

Př.: $P(x) = x^3+x+1 \implies p=3 \quad r=7$

$Q(x) = x^5+1 \implies q=5 \quad q \nmid p$

$n = 35 \quad l_1 = l_2 = 3$

nebo $l_1 = 2 \quad l_2 = 4$

anebo $l_1 = 1 \quad l_2 = 5$

Prodloužené Fireovy kódy

Místo jednoho mnohočlenu $P(x)$ se použije několik mnohočlenů $P_1(x), P_2(x), \dots$

Délka kódového slova: $n = \text{NSN}(q, r_1, r_2, \dots)$,

kde $r_1, r_2, \dots$ jsou řády $P_1(x), P_2(x), \dots$

Př.: $P_1(x) = x^{11}+x^7+x^6+x+1$
 $P_2(x) = x^{12}+x^{11}+\dots+x+1$
 $P_3(x) = x^{11}+x^9+x^7+x^6+x^5+x+1$
 $Q(x) = x^{22}+1$
 $G(x) = P_1(x) \cdot P_2(x) \cdot P_3(x) \cdot Q(x)$
kód (558 442, 558 386), tzn. $m=56$
shluky délky ≤ 11

Oprava shluku chyb

(princip Meggittova dekodéru)

cyklický kód: $\exists H(x) \quad G(x) \cdot H(x) = x^n + 1$

$$x^n = G(x) \cdot H(x) + 1$$

syndrom: $S(x) = E(x) \% G(x)$

shluk chyb: $E(x) = E'(x) \cdot x^j \quad x \nmid E'(x)$

$$E(x) \cdot x^{n-j} = E'(x) \cdot x^n$$

$$E(x) \cdot x^{n-j} = E'(x) \cdot G(x) \cdot H(x) + E'(x)$$

$$E(x) \cdot x^{n-j} \equiv E'(x) \pmod{G(x)}$$

$$S(x) \cdot x^{n-j} \equiv E'(x) \pmod{G(x)}$$

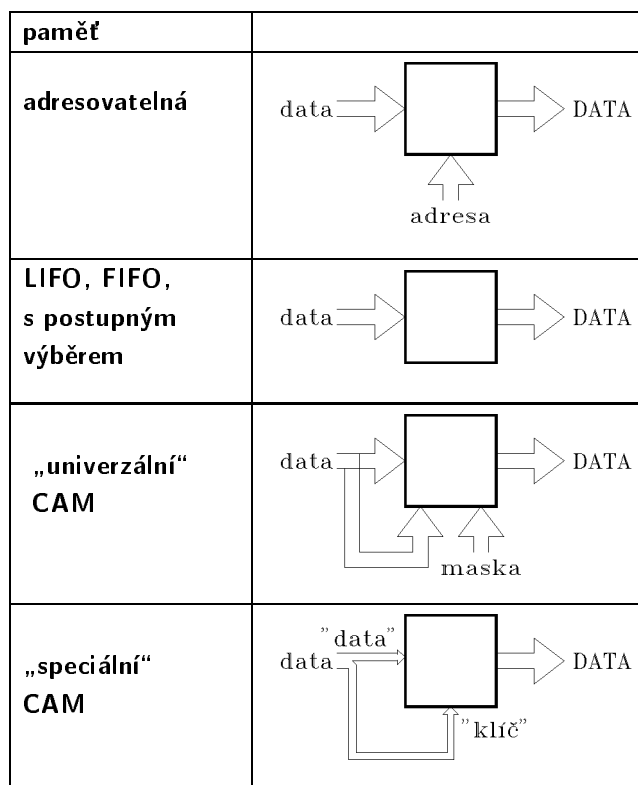
$$\deg E'(x) \leq \deg G(x)$$

$$E'(x) = (S(x) \cdot x^{n-j}) \% G(x)$$

oprava: Počítá se $S(x)$, $S(x) \cdot x$, $S(x) \cdot x^2$, atd., dokud není výsledkem mnohočlen, který přísluší opravitelné chybě. Výsledný mnohočlen je roven $E'(x)$ a počet provedených kroků určuje hodnotu j .

Organizace pamětí z hlediska výběru paměťových míst

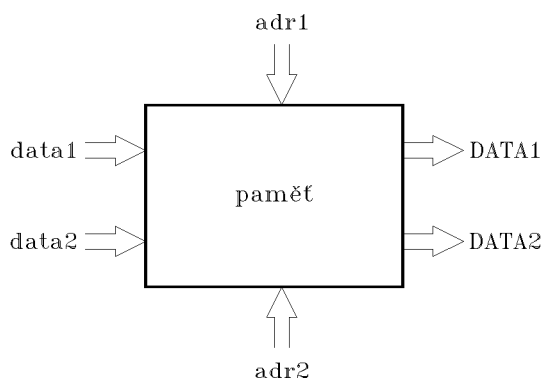
- adresovatelná, s adresním (adresovým) výběrem
 - ◊ paměťové místo určeno adresou
- s postupným výběrem
 - ◊ paměťová místa se vybírají postupně (jak za sebou následují)
- LIFO [Last-In First-Out]
zásobníková, sklípková [stack, push-down]
 - ◊ čte se poslední zapsaná položka
- FIFO [First-In First-Out]
fronta
 - ◊ čte se nejdříve zapsaná položka
- asociativní CAM [Content Addressed Memory]
 - ◊ paměťové místo určeno svým obsahem



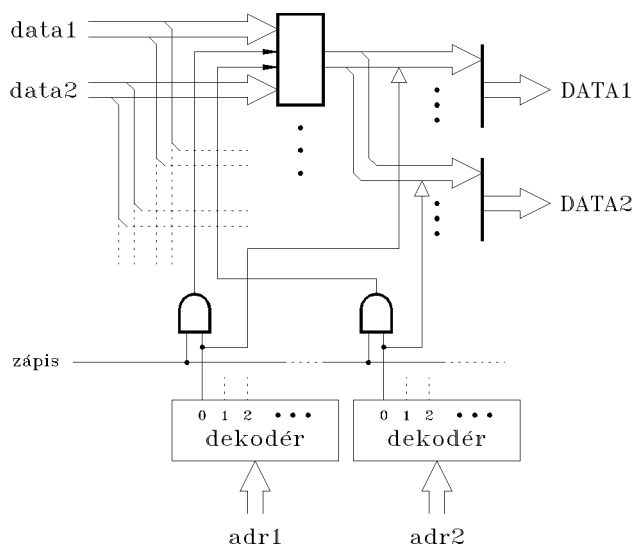
adresovatelné paměti

- jednobránová paměť
 - vícebránová paměť
- n bránová paměť: n adresových vstupů
 n (popř. méně) datových vstupů
 n (popř. méně) datových výstupů

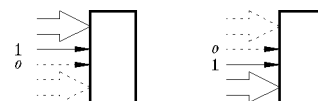
2bránová paměť:



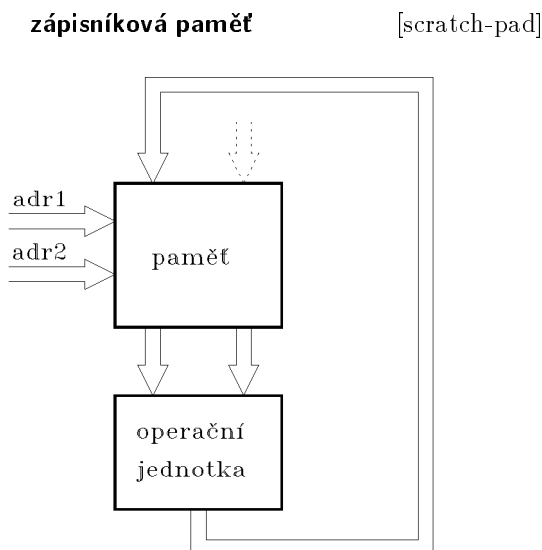
2bránová paměť tvořená registry



vysvětlivky (registry):

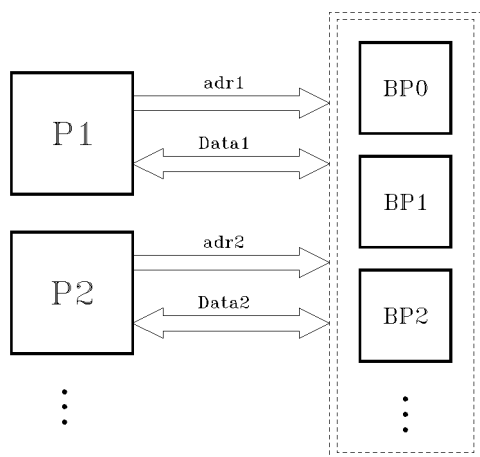


použití 2bránová paměti tvořené registry:



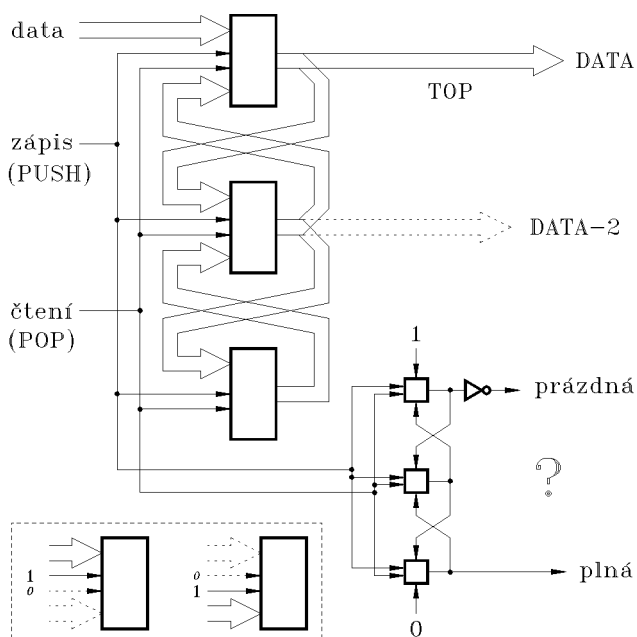
hlavní paměť

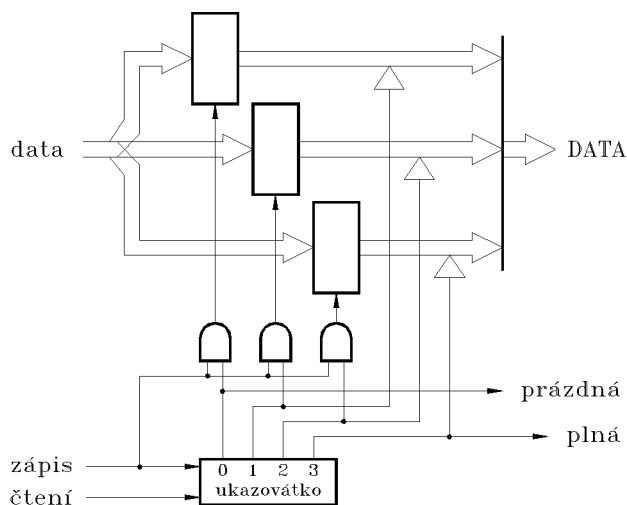
- **vícebránová** — připojení několika procesorů nebo jiných jednotek (např. kanálů pro vstup/výstup)
 - ◊ řízení: řadič hlavní paměti neboli organizátor
 - ◊ kolize požadavků — vyšší priorita V/V
⇒ odebrání cyklu [cycle stealing]
 - ◊ propojení: **přímé sběrnice**
propojovací síť
- **jednobránová**
 - ◊ **registr adres** [MAR = Memory Address Register]
 - ◊ **registr dat** [MDR = Memory Data Register]
 - ◊ **připojena obv. přes sběrnici**
podsběrnice: adresová, datová, řídicí
umožňuje připojení několika jednotek
kolize požadavků — řešena režimem přidělování sběrnice

vícebránová hlavní paměť
s autonomními paměťovými bloky

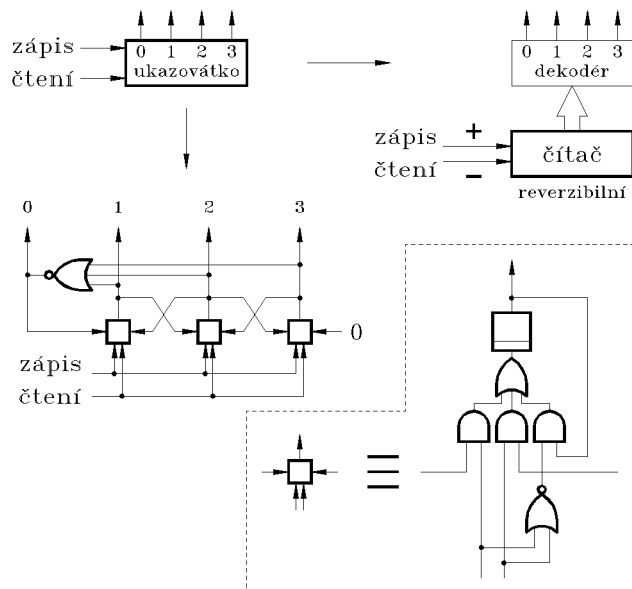
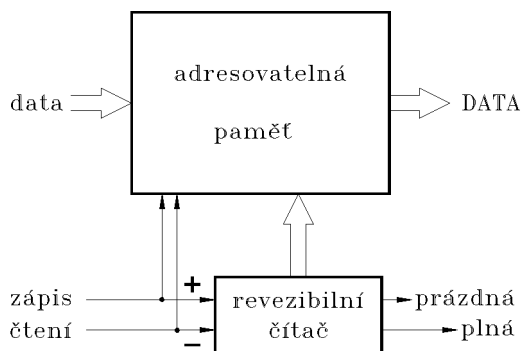
BP0, BP1, BP2, ... — autonomní paměťové bloky
P1, P2, ... — procesory (popř. jiné jednotky)

umožňuje paralelní (současnou) práci různých bloků,
pokud se požadavky týkají různých bloků (týkají se
různých částí paměťového prostoru)

paměť LIFO
sériově řazené registry

registry určené ukazovátkem

registry určené ukazovátkem

realizace ukazovátka:jádro = adresovatelná paměť

zápis (PUSH) — pre-inkrementace

čtení (POP) — post-dekrementace

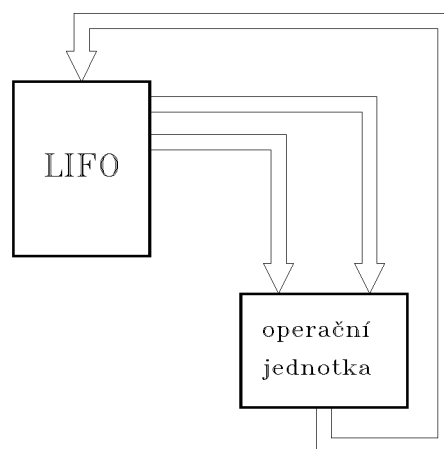
nebo naopak:

zápis (PUSH) — pre-dekrementace

čtení (POP) — post-inkrementace

použití paměti LIFO

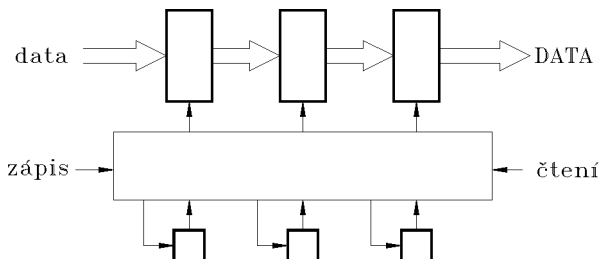
- návratové adresy:
podprogramy a mikropodprogramy
- registry aritmetické jednotky (zápisníková p.)



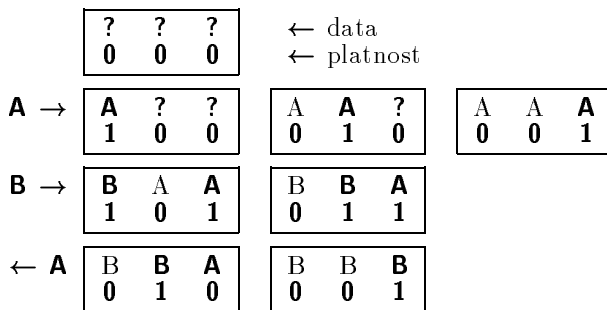
paměť FIFO

použití: vyrovnávací paměť

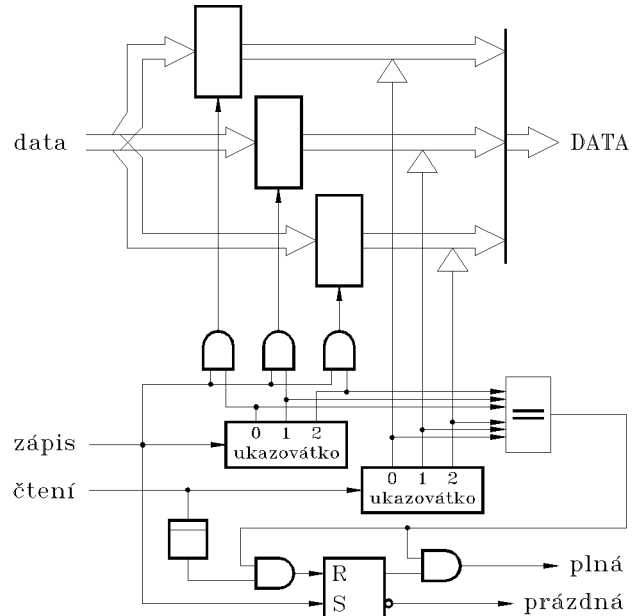
[buffer]

sériově řazené registry

!!! asynchronně !!!



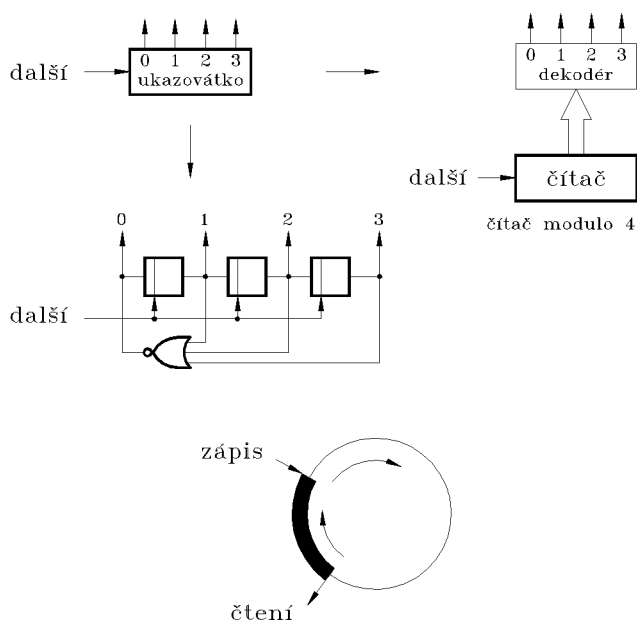
registry určené ukazovátky



pozn.: zde je ukazovátko 3místné (aby se to „někam vešlo“)

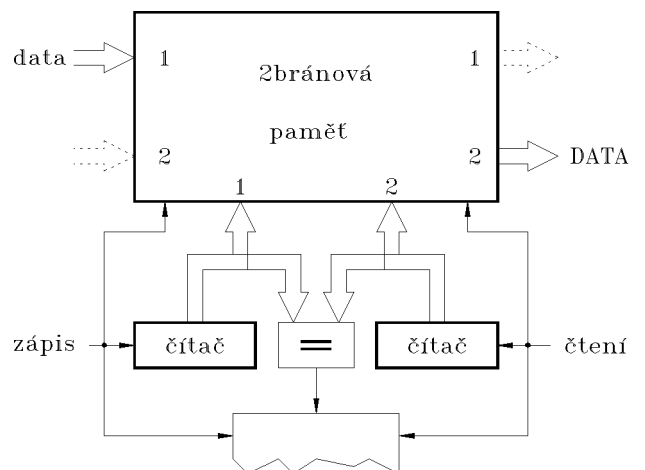
registry určené ukazovátkem

realizace ukazovátka:



pozn.: zde je ukazovátka 4místné (aby bylo „něco vidět“)

jádro = adresovatelná paměť

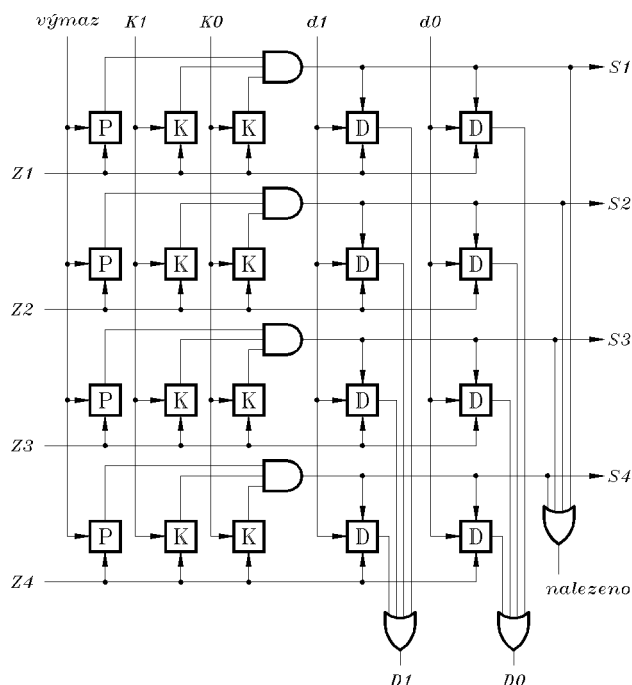


stejné obsahy čítačů $\Rightarrow$ $\begin{cases} \text{paměť prázdná} \\ \text{paměť plná} \end{cases}$

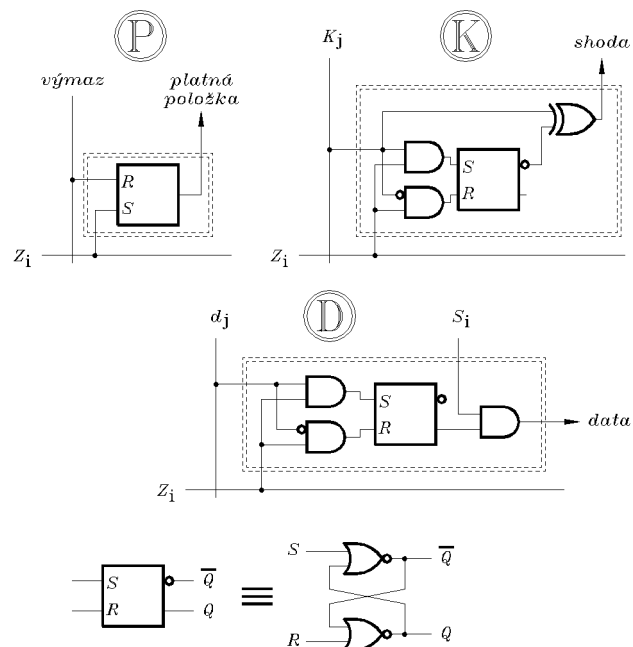
⇒ přidat klopný obvod:

- shoda čítačů po čtení $\Rightarrow$ nulovat
- zápis $\Rightarrow$ nastavit (zapsat 1)

Plně asociativní paměť



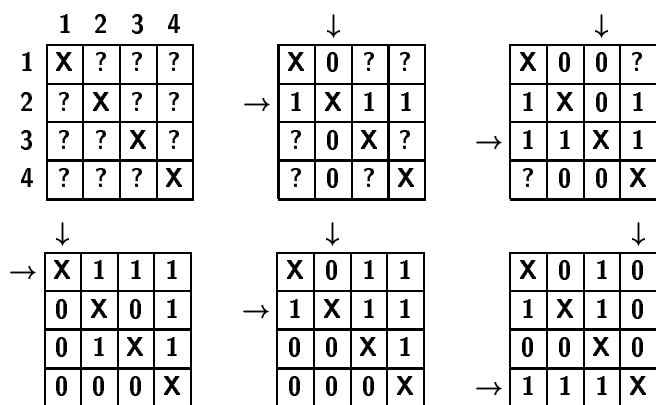
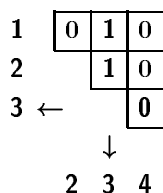
Plně asoc. paměť — vnitřní struktura bloků (princip)

konvence (není-li určeno jinak): **S - J - V - Z**

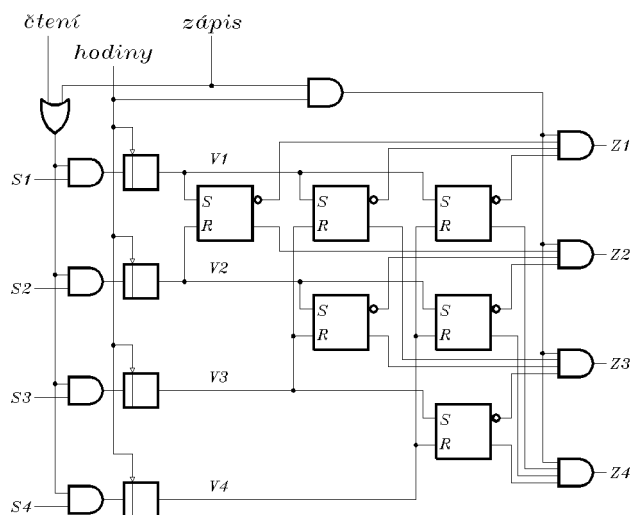
Strategie LRU

[Least Recently Used]

Princip: matice M — použita položka $i \Rightarrow$

$$\Rightarrow 1 \ 1 \ \dots \ 1 \rightarrow i\text{-tý řádek}$$
$$\Rightarrow 0 \ 0 \ \dots \ 0 \rightarrow i\text{-tý sloupec}$$

$$M_{ij} = \overline{M_{ji}} \Rightarrow \text{stačí prvky nad diagonálou:}$$


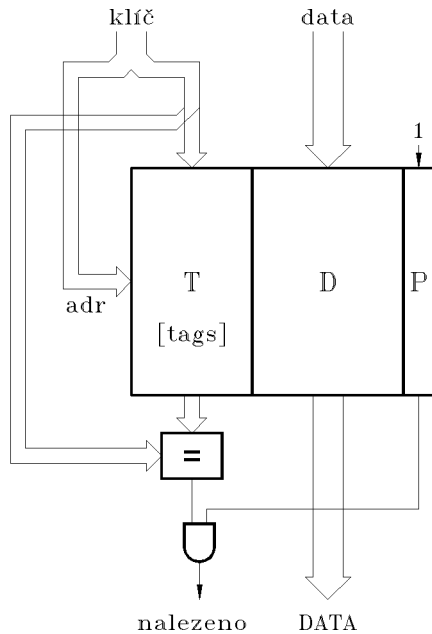
Strategie LRU — schéma



pozn.: Zde se předpokládá, že zápis do synchronních klopných obvodů typu D se provede při závěrné hraně hodin.

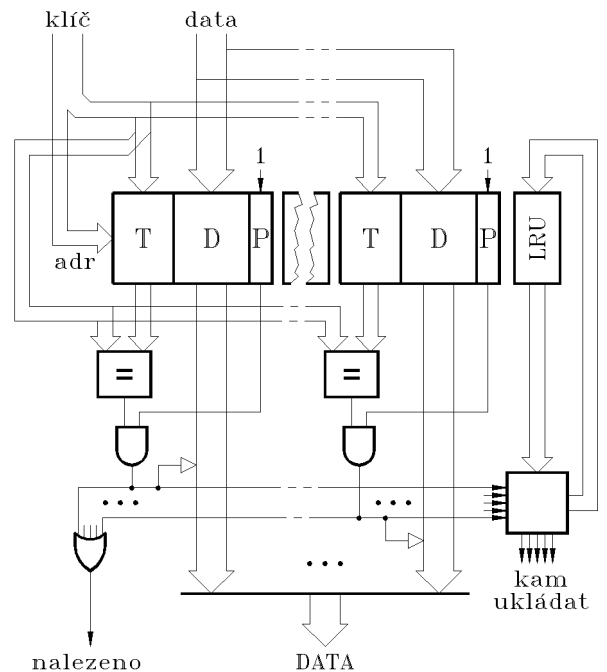
Asociativní paměť s omezeným stupněm asociativity

stupeň asociativity $s = 1$ [direct mapped]



Asociativní paměť s omezeným stupněm asociativity

stupeň asociativity $s > 1$ [s -way set associative]



Asociativní paměť strategie uvolňování položek:

- **LRU** [Least Recently Used]
nejdéle nepoužitá položka
- **FIFO** [First-In First-Out]
nejstarší položka
pole „LRU“ — čítač
- „náhodně“ [random]
např.: společný čítač pro všechny položky
- **pseudo – LRU**
 - **FIFO + ochrana posledně použité položky**
pole „LRU“ — číslo posledně použité položky
 - **nějak „ošizené“ LRU**
př.: cache I486 — 3bitové pole „LRU“ pro $s=4$

PRINCIPY LOKALIT

Lokality dat

- **Časová lokalita** [temporal locality]
Jsou-li použita nějaká data, dá se očekávat, že táž data budou brzy použita znovu.
- **Prostorová lokalita** [spatial locality]
Jsou-li použita nějaká data, dá se očekávat, že brzy budou použita data uložená na blízké adrese.

Lokality instrukcí

- **Časová lokalita** [temporal locality]
Je-li použita nějaká instrukce, dá se očekávat, že táž instrukce bude brzy použita znovu.
- **Prostorová lokalita** [spatial locality]
Je-li použita nějaká instrukce, dá se očekávat, že brzy bude použita instrukce uložená na blízké adrese.

Skrytá paměť [cache]

(Rychlá vyrovnávací paměť) [buffer]

!!! Termín skrytá paměť není běžně používán !!!

skrytá paměť: asociativní paměť (CAM)

HP → { → procesor HP ... hlavní paměť
→ SP SP ... skrytá paměť
data/instrukce ∈ SP ⇒ nečte se z HP, ale ze SP
SP výrazně rychlejší než HP (jinak to nemá smysl)

skrytá paměť: asociativní paměť (CAM)

klíč: adresa v HP

data: příslušné bloky dat/instrukcí
nezapomenout na bit platnosti !

- využívá se principů časové lokality
větší kapacita SP ⇒ častěji se čte ze SP
→ omezený stupeň asociativity
- využívá se principů prostorové lokality:
 - nečtou se jednotlivá slova/slabiky, ale bloky
 - jako „tag“ se použijí vyšší řády

Příklad:

HP: kapacita = 64KB = 2¹⁶B

...	...
0FF0	F0 F1 F2 F3 F4 F5 F6 F7 00 01 02 03 04 05 06 07
1000	08 09 0A 0B 0C 0D 0E 0F FF FE FD FC FE EE ED ED
...	...

SP: 4 adresy stupeň asociativity = 2
délka bloků = 4 kapacita = 32B = 4·4·2 B

0	987	10 20 30 40	P	100	08 09 0A 0B	P	0
1	100	0C 0D 0E 0F	P	0FF	0C 0D 0E 0F	P	0
2	000	11 22 33 44	N	321	FF FF FF FF	N	1
3	100	EF EE ED ED	P	0FF	04 05 06 07	P	0

100	EF	EE	ED	ED	P	0FF	04	05	06	07	P
0FF	≠				P	=	04	05	06	07	P
							0	1	2	3	
							↑↑↑				

Čtení obsahu paměťového místa v HP na adrese
0FFE₁₆ = 0000 1111 1111 11 10₂ ~ 0FF 3 2

ČTENÍ

koeficient úspěšnosti u [hit ratio]
 koeficient výpadku bloku $v = 1 - u$ [miss ratio]

t ... vybavovací doba SP

T ... vybavovací doba HP

τ ... průměrná vybavovací doba

- sériové čtení [look-through]

nejprve čtení ze SP

neúspěšné čtení ze SP $\Rightarrow$ čtení z HP

$$\tau = t + v \cdot T = u \cdot t + v \cdot (t + T)$$

nezatěžuje zbytečně sběrnice

- paralelní čtení [look-aside]

současné čtení ze SP i z HP

úspěšné čtení ze SP $\Rightarrow$ zastaví se čtení z HP

$$\tau = u \cdot t + v \cdot T < t + v \cdot T$$

rychlejší než sériové čtení

ZÁPIS

- průběžný [write-through]

každý zápis se provádí do HP

je zajištěna konzistence dat v HP a ve SP

- s rezervací místa [allocate-write]

$\text{blok} \notin \text{SP} \Rightarrow \begin{cases} \text{čtení bloku z HP do SP} \\ + \text{ zápis dat do SP} \end{cases}$

- bez rezervace místa [non-allocate-write]

$\text{blok} \notin \text{SP} \Rightarrow \text{data ze zápisů pouze do HP}$

Budou-li tato třeba, bude nutné čtení z HP!

- odložený [copy-back]

zápisy se provádějí pouze do SP

$\Rightarrow$ menší zatížení sběrnice

$\Rightarrow$ rychlejší zápis

??? konzistence dat v HP a ve SP ???

příznak modifikace [dirty, modified, write-into]

(vedle bitu platnosti) [present, valid]

Před přepsáním modifikovaného bloku v SP je třeba tento blok uložit do HP.

zápis

- přímo
- přes vyrovnávací paměť (buffer)

Konzistence (neboli koherence) dat (zejm. v multiprocessorových systémech)

- průběžný zápis + zákaz ukládání určených dat (stránek) do SP
 - průběžný zápis + hlídání sběrnice [snooping]
 $\rightarrow$ hlídač [watch dog]
 - odložený zápis ???!!!!???
- protokol **MESI** ... Modify
 Exclusive
 Shared
 Invalid

? 1 nebo 2 SP ? (data + instrukce)

SP by měla být transparentní, a proto:

- 1 SP ... procesory von Neumanova typu
- 2 SP ... procesory harvardského typu („klasické“)

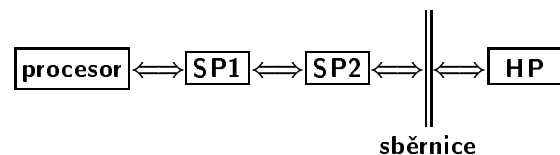
Je vhodné vzít v úvahu:

- 1 SP ... zajišťuje transparentci
 i při přepisování instrukcí
- 2 SP ... nemusí být „stejně“ využívány

? 1 nebo 2 SP ? (jinak)

Hierarchie SP

motivace: větší kapacita $\Leftrightarrow$ menší rychlost



zprav.: SP1 ... průběžný zápis [write-through]

SP2 ... odložený zápis [copy-back]

Virtuální paměť (VP)

pomyslná paměť — využívá se jako „řádná“ paměť
paměťová místa jsou „chvilí tam“ – „chvilí jinde“
hlavní paměť + vnější paměť — adresované „jinak“
HP ... hlavní paměť

- motivace:**
- zvýšení kapacity paměti
 $VP \sim HP$ (ale podstatně větší kapacita)
 - předadresování (nečiní problémy)
 - ochrana paměti (jedna z možností)

- typy:**
- stránkovaná
 - segmentovaná

!!!

Segmentace např. u 8086
reprezentuje zcela jiný problém.

!!!

- kombinovaná
př.: 80x86, kde $x86 \geq 386$ a je použit tzv.
(nikoliv tzv. reálný) režim
virt. a. → **segm** → lin. a. → **str** → fyz. a.

základní principy:

- Procesor akceptuje pouze data/instrukce uložená v HP.
- Proto: Nejsou-li data/instrukce uložena v HP přesunou se do ní z vnější paměti.
- princip časové lokality
⇒ data/instrukce budou brzy asi znovu použita
- princip prostorové lokality:
Když už se provádí přesun, přesune se toho víc:
 - celá stránka (např. 256B, ..., 4KB, ... 4MB)
 - celý segment (např. 1B ÷ 32GB)

- adresy:**
- virtuální (matematické, logické)
 - fyzické — v HP
 - na vnější paměti

překlad adres: • virtuální → fyzické

HW

není-li to možné
přerušeni a potom:

[page fault]

- virtuální → na vnější paměti
- a přesun z vnější paměti do HP

SW

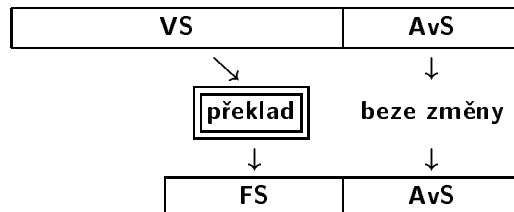
Stránkovaná virtuální paměť

stránka ... množina $K = 2^k$ paměťových míst
začíná na adrese dělitelné K
typická velikost 4kB = 2^{12} B
adresový prostor HP lze rozdělit do rámců [frames]
rámec ~ stránka
analogicky lze rozdělit do rámců adresový prostor VP

Využití HP:

- Některé z rámců vyžívají systémové programy („normální programy“ je tedy využívat nemohou). Systémové programy (jádro operačního systému) zajišťují např. zpracování přerušeni a pod. a obsahují např. tabulku přerušovacích vektorů, (dále diskutovanou) tabulku stránek apod.
- Ostatní rámce jsou vyžívány pro implementaci virtuální paměti (a jsou tedy využívány pro „normální programy“).

Překlad virtuálních adres na fyzické:



VS ... virtuální stránka (číslo rámce)
FS ... fyzická stránka (v HP) (číslo rámce)
AvS ... adresa ve stránce [offset]

Překlad provádí technické vybavení (HW) a vyžaduje:

- tabulku stránek (TS) v HP
 - registr ZTS — registr začátku tabulky stránek
 - registr DTS — registr délky tabulky stránek
- Registry ZTS a DTS jsou programově dostupné.

Nepovede-li se překlad, dojde k přerušeni programu.

⇒ Systémové programy (tedy SW)

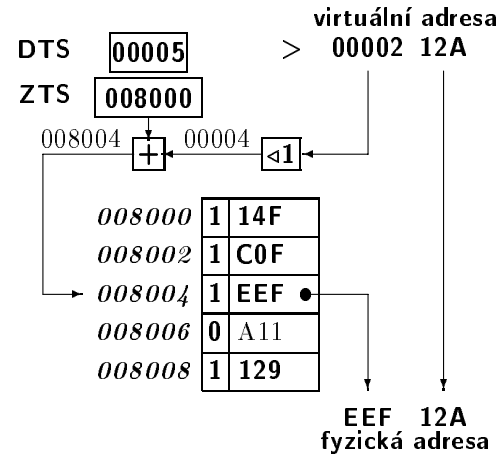
- změní obsazení HP (zavede se do ní VS),
- aktualizuje TS, ZTS a DTS

Potom se bude pokračovat v programu (od místa přerušeni).

Tabulka stránek TS (uložená v HP):

- Položky TS[i] jsou indexovány $i = 0, 1, 2, 3, \dots$; položka TS[i] přísluší VS (virtuální stránce) i.
- Každá položka musí obsahovat:
 - číslo rámce HP příslušejícího dané VS (pokud je VS vůbec v HP — jinak cokoliv)
 - bit platnosti ($1 \iff VS$ je v HP)
- Položky jsou stejně dlouhé (a to dostatečně).
- Délka položky se volí tak, aby mohla být uložena v $L = 2^l$ paměťových (adresovatelných) místech (např. 4B).
 $\implies \text{adresa}(\text{TS}[i]) = \text{ZTS} + i \cdot L$, neboli
 $\text{adresa}(\text{TS}[i]) = \text{ZTS} + i \ll l$, kde
 ZTS je obsah registru ZTS a
 $i \ll l$ označuje posuv i o l míst vlevo.
- Položka TS[i] přísluší VS uložené v HP právě tehdy, když bit platnosti je roven 1 a $i < DTS$, kde DTS je obsah registru DTS.

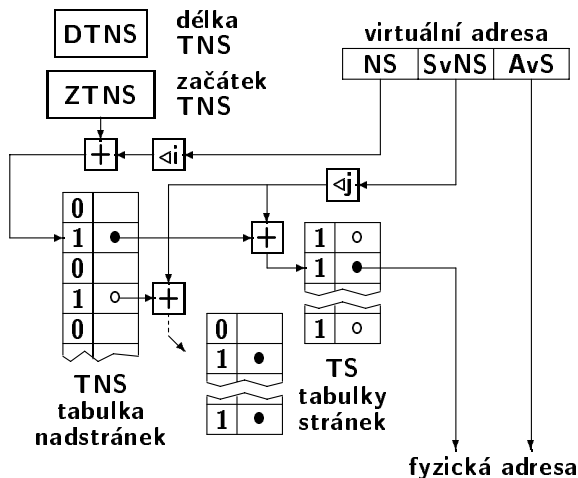
Př.: 1 stránka $\sim 4\text{KB} = 2^{12}\text{B} \sim 1$ rámec
 VP ... $4\text{GB} = 2^{32}\text{B} \sim 1\text{M} = 2^{20}$ rámců
 HP ... $16\text{MB} = 2^{24}\text{B} \sim 4\text{K} = 2^{12}$ rámců
 položka TS: $l \geq 12\text{b} + 1\text{b} \implies L = 2\text{B}$
 max. velikost TS = 2MB

**2úrovňová tabulka stránek**

virtuální adresa:

NS	SvNS	AvS
----	------	-----

 NS ... nadstránka
 SvNS ... stránka v nadstránce
 AvS ... adresa ve stránce



Pozn.: položka TNS může obsahovat také délku TS
víceúrovňová tabulka stránek — analogicky

Příklady HP: $2^{24}\text{B} = 16\text{MB}$

$$2^{12} = 4\text{K rámců} \quad \text{a} \quad 2^{12}\text{B} = 4\text{KB}$$

20b	12b
VS	AvS

TS: 1M položek à 2B = 2 MB

14b	6b	12b
NS	SvNS	AvS

TNS: 16K položek à 4B = 64 KB

TS: max. 64 položek à 2B = 128B

$$\text{max.} < 4\text{K TS} = \frac{512\text{ KB}}{576\text{ KB}}$$

10b	10b	12b
NS	SvNS	AvS

TNS: 1K položek à 4B = 4 KB

TS: max. 1K položek à 2B = 2KB

$$\text{max.} < 1\text{K TS} = \frac{2000\text{ KB}}{2004\text{ KB}}$$

ad 3.:

- Byl uvažován horší případ než „nejhorší možný“!
- Stěží budou poslední „nejpoužívanější“ stránky v příliš mnoha nadstránkách $\Leftarrow$ prostorová lokalita

1 úrovně tabulky stránek:

- při každém přístupu do VP 1 čtení z HP „navíc“
- tedy: VP je $2 \times$ pomalejší než HP

více úrovně tabulky stránek:

- méně paměti
- další čtení z HP $\Rightarrow$ delší cyklus paměti

??? řešení ???

TLB : [Transfer Look-aside Buffer]

- asociativní paměť (CAM)
— velmi často plně asociativní (!!!)
- klíč: virtuální stránka
- data: fyzická stránka + ...
- kapacita < max. počet fyzických stránek (obvykle)
typicky 64 slov
- transparentní (obvykle)
- není paměť „funkčně nezbytná“ (obvykle)
- kopie položek jsou v TS (popř. v TNS)
(jejichž čtení však trvá déle)

Výměna stránek v HP

VS $\notin$ HP $\Rightarrow$ přerušení $\Rightarrow$ je třeba: VS $\rightarrow$ HP

otázky: 1. Je místo v HP?

2. Co vyhodit?

3. Jak vyhodit?

ad 1.: „ano“ $\Rightarrow$ Není o čem diskutovat!

ad 2.: řeší se prostředky SW

$\Rightarrow$ lze použít i složitější algoritmy

$\Rightarrow$ HW tomu musí poskytnout podporu

ad 3.: řeší se prostředky SW:

Došlo ke změně VS umístění v HP

(došlo k zápisu)?

ANO $\Rightarrow$ před přepsáním stránky v HP

je nutné zajistit konzistenci

(uložit přepisovanou stránku)

NE $\Rightarrow$ nic

HW: je třeba uložit programově dostupný

příznak změny [modify, dirty]

vhodné místo (programově dostupné)

— položka TS

Výměna stránek v HP

ad 2. (ještě jednou)

Jak vybrat „tu správnou“ stránku „k vyhození“?

nejlépe: Vyhodit stránku, která
bude(!) potřeba nejpозději. ? prognóza?

Řeší se prostředky SW. To vyžaduje „podporu HW“:

příznak použití stránky [used, referenced]

vhodné místo (programově dostupné) — položka TS

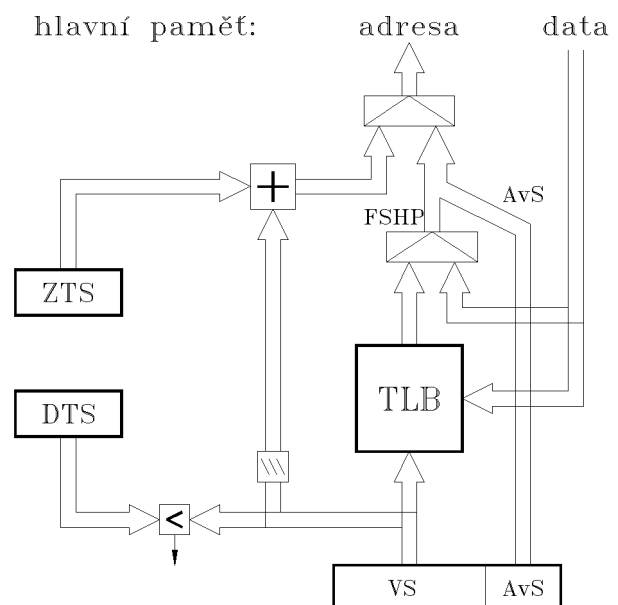
Používá se režim tzv. „krájení času“. [time slicing]

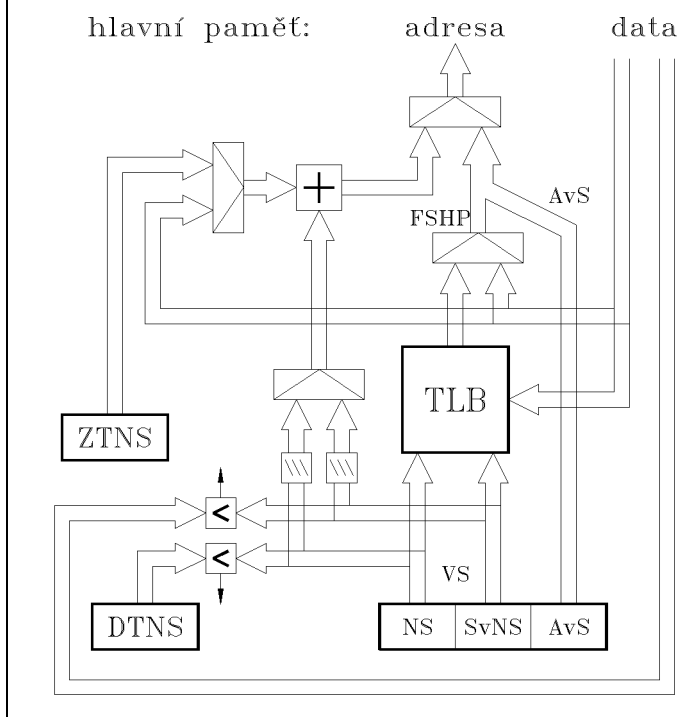
Program je přerušován časovačem. (např. po msec)

Při každém přerušení dojde

— k vyhodnocení příznaků

— k vymazání (teď již „starých“) příznaků

1 úrovně tabulka stránek

2 úroňová tabulka stránek**Výměna stránek v HP**

ad 2. (do třetice)

Jak vybrat „tu správnou“ stránku „k vyhození“?

Řeší se prostředky SW !!!

- **LRU** [Least Recently Used]
 - **matice ?** (viz asociativní paměti) **? rozměry ?**
 - **čítače ?** — pro každou fyzickou stránku jeden
 - poněkud nestandardní čítač
 - počet bitů = počet stránek
 - **čítání počtu přerušení způsobených časovačem**
 t = číslo takového přerušení
 stránka byla použita $\Rightarrow$ přidělit jí hodnotu t
 minimální t ... nejdéle nepoužito
- **FIFO** [First-In First-Out]
 - Vyhodí se to nejstarší.
 - modif.: ochrana posledně použitých stránek
- **LFU** [Least Frequently Used]
 - počet intervalů od poslední výměny stránek, v nichž byla stránka použita
 - Vyhodí se stránka s minimálním ohodnocením.

Tabulka stránek TS (znovu a úplněji):

Položka TS by měla (může) obsahovat:

- číslo rámce HP (číslo fyzické stránky)
- bit platnosti [present, valid]
- bit změny (zápis do stránky) [dirty, modified]
- bit „použito“ [used, referenced]
- jiné tzv. řídicí bity, např.
 - zákaz ukládání do SP (skrytá paměť — cache)
 - zákaz zápisu do stránky (tzv. ochrana paměti — ze stránky lze číst, ale nelze do ní zapisovat)

Segmentovaná virtuální paměť

segment ... množina paměťových míst

velikost je omezena

může být umístěn kdekoli v HP

př.: $1\text{B} \div 4\text{GB} \in \text{max. } 4\text{GB HP}$

využití HP: Část využívají systémové programy.

Zbytek využívají „normální“ programy.

Obdobné principy jako u stránkované VP.

??? Čím se liší ???

- velikost stránky / segmentu (stejná / různá)
- umístění stránky / segmentu ($i \cdot K$ / kdekoli)
- tabulka stránek / tabulka segmentů
- položka tabulky segmentů, tzv. popisovač segmentu [segment descriptor], musí obsahovat informaci o délce segmentu
- pro určení fyzické adresy nutná sčítačka

SBĚRNICE

[BUS]

soubor vodičů a pravidel určený
k propojení jednotek počítače

styl = propojení = rozhraní [interface]

podsběrnice — obvyklé též nazývané sběrnice:

- adresová [Address Bus]
- datová [Data Bus]
- řídicí ... řídicí a stavové signály [Control Bus]

adresy a data mohou být multiplexovány (časově):

⇒ stačí jediná sběrnice pro adresy i data

⇒ je třeba řídicí signál „adresa / data“

připojení jednotek ke sběrnici — výstupy:

- otevřený kolektor — několik jednotek může vysílat současně signál (nulu) na týž vodič
- 3stavový výstup — „vysílat“ může pouze ta jednotka, která to má povoleno

jednotky { aktivní ... právě komunikují
pasivní

aktivní jednotka:

- ♠ řídicí (nadřízená, řadič) [master]
- ♠ řízená (podřízená) [slave]

aktivní jednotka:

- ♡ vysílající (vysílač, zdroj) [source, talker]
- ♡ přijímající (přijímač, cíl) [destination, listener]

operace (typické):

- čtení [read]
- zápis [write]
- čtení – změna – zápis [read – modify – write]
- zápis – čtení [read after write]
- přenos bloku [block data transfer]

- ⊗ broadcast (oběžník?): příkaz všem jednotkám
- ⊗ broadcast (???): informace ode všech jednotek (popř. od určených jednotek)

sběrnice:

- ◇ jednoúčelové („soukromé“) [dedicated]
- ◇ sdílené [shared, non-dedicated]
 - systémové
 - lokální
 - V/V

sběrnice:

- ♡ paralelní
- ♡ sériové ... záložní a pro systémové účely

sběrnice:

- ♣ synchronní ← hodiny
jednosměrně řízené [one-way control]
- ♣ asynchronní
zprav. obousměrně řízené [two-way control]
⇒ korespondenční režim
- ♣ pseudosynchronní (semisynchronní, polosynchr.)
hodiny + signál(y) „čekání“ [WAIT]
(obousměrně řízené)

korespondenční režim

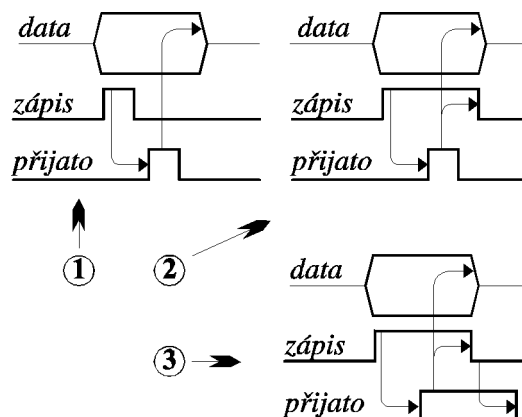
[hand shake]

- ① jednoduše vázaný [non-interlocked]
- ② středně vázaný [half interlocked]
- ③ plně vázaný [fully interlocked]

	①	②	③
operace	←	←	←
požadavek	↘	↘	↘
potvrzení	↘	↘	↘

↘ trvá stanovenou dobu
↘ ukončeno po přijetí „odpovědi“

Př.: zápis — adresa: zároveň s daty (nezakreleno)



přidělování sběrnic:

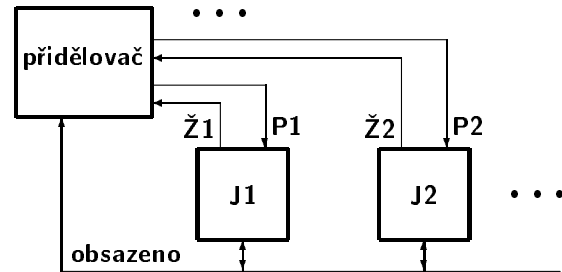
- **centralizované: přidělovač sběrnice** [bus arbiter]
(funkci přidělovače někdy zastává procesor)
 - **paralelní = nezávislé žádosti a potvrzení** [independent requests and grants]
 - **sériové** [daisy chain]
 - **kombinované — paralelní a sériové**
 - **cyklické výzvy** [pooling]
žádost o přidělení sběrnice $\Rightarrow$
 $\Rightarrow$ **přidělovač „nabízí“ postupně a adresně sběrnici jednotlivým jednotkám**
- **distribuované ... bez přidělovače**
 - **prioritní**
při větším zatížení sběrnice:
starvation („vyhladovění“): jednotky s nízkou prioritou „nemají šanci“
řešení: fairness („korektnost“, „slušnost“)
 - **round-robin: jednotky si „cyklicky“ předávají právo přístupu na sběrnici** (srov.: token ring)

pozn.: přidělování lze provádět „s předstihem“

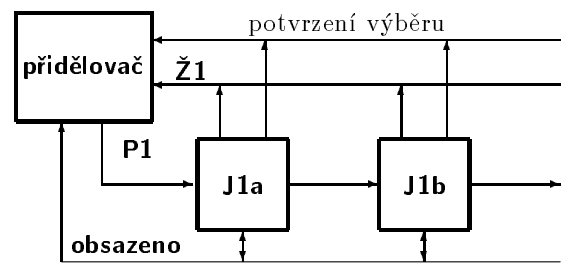
Centralizované přidělování:

J_i jednotka P_i přiděleno $\bar{Z}_i$ žádost

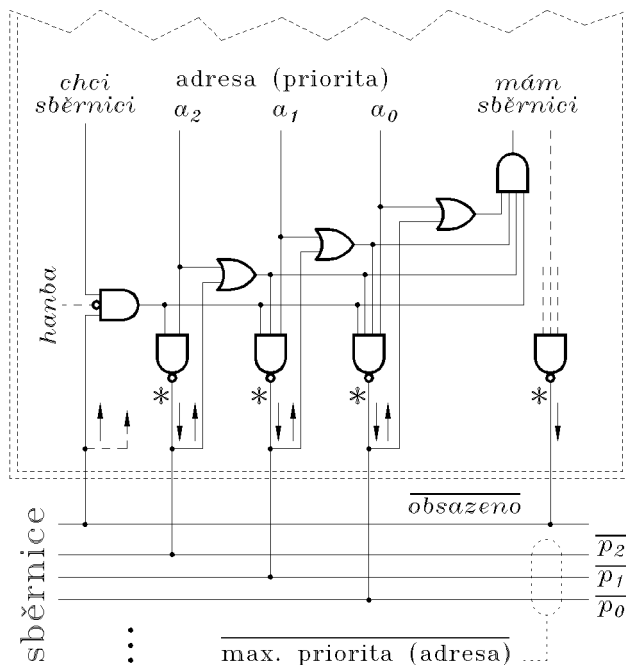
paralelní:



sériové:

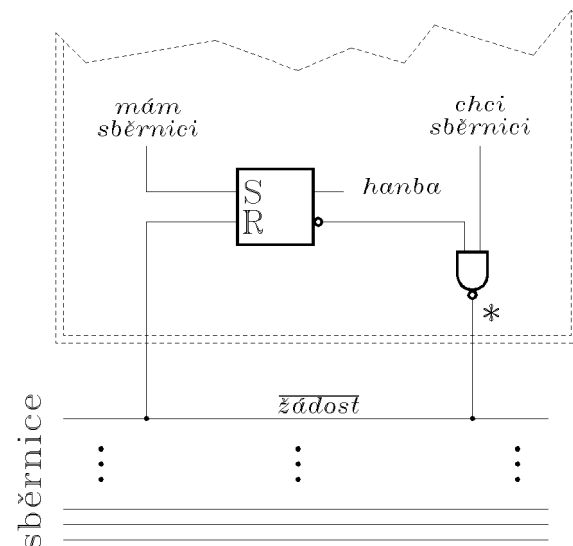


Distribuované přidělování (prioritní)

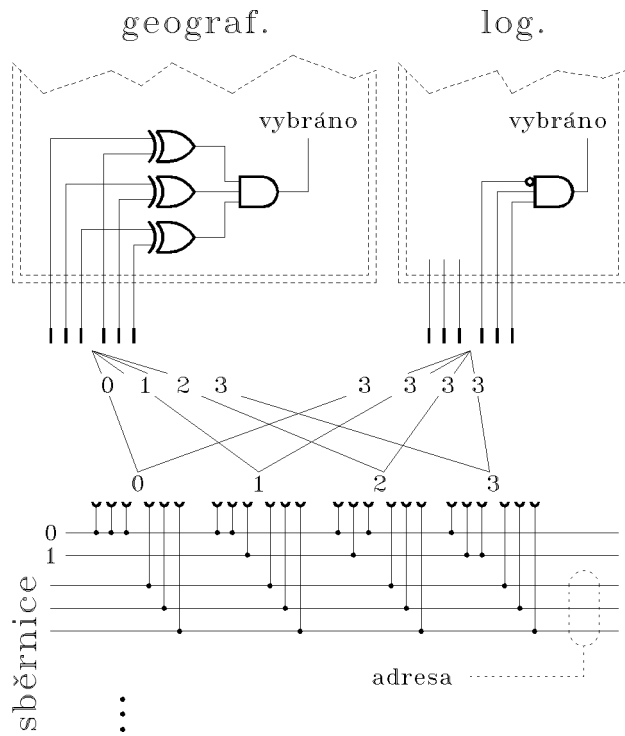


* ... otevřený kolektor

Fairness:



Logické a geografické adresování:



Některé sběrnice

UNIBUS	DEC	
MUBUS	Švýc. feder. technol. ústav	1975
S-100	MITS	1976 IEEE 696
MULTIBUS	Intel	1974 IEEE 796
STD-BUS	Pro-log	1981 IEEE 961
VME	Motorola, Mostek,	IEEE 1096,
	Philips-Signetics,	IEEE 1132,
	Thomson-CSF	1981 ...
NUBUS	MIT	1979 ANSI/IEEE
	(Western Digital,	1196/1987
	Texas Instruments)	
MULTIBUS II.	Intel	1983 IEEE 1296
FUTUREBUS+	1983	IEEE
		896.1-1987
V/V:	Centronics	
	SCSI	ANSI X3.131-1986
PC:	ISA, EISA, MCA, VL-BUS, PCI	
SPARC:	SBUS, MBUS	

Komprese dat (Komprimace dat)

Př.: zakódovat slovo ARARAUNA

	$\mathcal{K}_1$	$\mathcal{K}_2$	četnost	
			absolutní	relativní
A	00	0	4	0,500
N	01	111	1	0,125
R	10	10	2	0,250
U	11	110	1	0,125

kód $\mathcal{K}_1$: 00 10 00 10 00 11 01 00 ... 16 bitů

kód $\mathcal{K}_2$: 0 10 0 10 0 110 1110 ... 14 bitů

prefixový kód: žádné kódové slovo není prefixem
(začátkem) jiného kódového slova

Př.: $\mathcal{K}_1$ — blokový kód (viz též bezpečnostní kódy)

$\mathcal{K}_2$ — prefixový kód

postfixový kód: žádné kódové slovo není postfixem
(koncovkou) jiného kódového slova

afixový kód: kód, který je prefixový i postfixový

komprese (též komprimace): kódování (přesněji: překódování) — snížení počtu bitů

dekódování: dekomprese (dekomprimace)

kompresní poměr $KP = \frac{\text{nová délka}}{\text{původní délka}}$

Př.: předp.: $\mathcal{K}_1$: původní kód

$\mathcal{K}_2$: nový kód — komprimovaná data

$KP = 14/16 = 87,5\%$ (pro „ARARAUNA“)

zpráva = řetěz dílčích zpráv

např.: text = řetěz písmen a dalších znaků

zdrojová jednotka S_j — (původní) dílčí zpráva

kódová jednotka C_j — zakódovaná dílčí zpráva

informační entropie

Čím menší je pravděpodobnost p zprávy, tím je zpráva cennější — čím větší je její informační obsah E , tzn.:

$\frac{1}{p}$ roste $\implies$ roste E

logaritmická míra: $E = \log_2 \frac{1}{p} = -\log_2 p$

$E = -\log_2 p$ (shannon)

E ... informační entropie, informační obsah, míra množství informace aj. (viz též bezpečnostní kódy)

stejně pravděpodobná n bitová čísla:

počet čísel: $N = 2^n$

pravděpodobnost: $p = \frac{1}{N} = 2^{-n}$

entropie: $E = n$ (shannon $\sim$ bit)

v n bitovém čísle může být n shannonů

p_i ... pravděpodobnost i té z k možných zpráv

průměrná entropie

$$H = - \sum_{i=1}^k (p_i \cdot \log_2 p_i)$$

Př.: příkl. „ARARAUNA“;

předp.: pravděpodobnost = relativní četnost

entropie písmen A, N, R, U: 1, 3, 2, 3

$\frac{4}{8} \frac{1}{8} \frac{2}{8} \frac{1}{8}$
průměrná entropie 1 písmene:

$H = 0,5 \cdot 1 + 0,125 \cdot 3 + 0,25 \cdot 2 + 0,125 \cdot 3 = 1,75$

p_j ... pravděpodobnost zdrojové jednotky S_j

E_j ... entropie zdrojové jednotky S_j

H_j ... průměrná entropie j té dílčí zprávy

zpráva $S_1, S_2, \dots, S_m$:

pravděpodobnost $p = p_1 \cdot p_2 \cdot \dots \cdot p_m$

entropie $E = -\log_2 p$, tzn.:

$$E = \sum_{j=1}^m E_j$$

průměrná entropie (analogicky):

$$H = \sum_{j=1}^m H_j$$

Př.: entropie zprávy ARARAUNA:

$E = 1+2+1+2+1+3+3+1 = 14$ (shannon)

průměrná entropie 8písmenového slova:

$H = 8 \cdot 1,75 = 14$ (shannon)

redundance (nadbytečnost)

E ... entropie zprávy

L ... délka zápisu = max. možná entropie

redundance: $R = L - E$

p_i ... pravděpodobnost i té z k možných zpráv

E_i ... entropie i té zprávy

L_i ... délka zápisu i té zprávy

H ... průměrná entropie

průměrná redundance: $Q = \sum_{i=1}^k p_i \cdot (L_i - E_i)$

$$Q = \sum_{i=1}^k p_i \cdot L_i - H$$

Př.: příkl. „ARARAUNA“; kód $\mathcal{K}_1$

předp.: pravděpodobnost = relativní četnost

$R = Q = 16 - 14 = 2$ (shannon)

komprese dat: snížit redundanci na minimum!

Metody komprese (vybrané:)

• speciální

- ◊ čísla
 - Elias
 - Fibonacci
- ◊ obrazové informace (přírůstky)
- ◊ databáze (jména, data narození apod.)
- ◊ ...

• obecné I.

- ◊ statické — týž kód pro různá data
 - Shannon – Fano
 - Huffman
 - aritmetické k.
- ◊ hybridní = semiadaptivní — kód $\in$ data
- ◊ dynamické = adaptivní — týž algoritmus pro kompresi i dekompresi
 - FGK (Faller – Gallager – Knuth)
 - V (Vitter)

• obecné II. — slovníkové metody

- ◊ statické
- ◊ hybridní = semiadaptivní
- ◊ dynamické = adaptivní
 - Lempel – Ziv (LZ77, LZ78)
 - BSTW (Bentley – Sleator – Tarjan – Wei)

Eliasův kód I.

kódování přirozených čísel (větších než 0)

předpoklad: menší čísla — častější výskyt

princip: n bitový zápis čísla, začínající 1

před tento zápis umístit $n-1$ nul

$\Rightarrow$ prefixový kód

1	1
2	0 1 0
3	0 1 1
4	0 0 1 0 0
5	0 0 1 0 1
6	0 0 1 1 0
7	0 0 1 1 1
8	0 0 0 1 0 0 0
9	0 0 0 1 0 0 1
...	...

Eliasův kód II.

před n bitový zápis čísla umístit n v Eliasově kódu I.
z n bitového zápisu čísla vypustit první bit (jedničku)

Př.: $386_{10} = 11000\ 0010_2 \Rightarrow n = 9 \sim 000\ 1001$

výsledné kódování: $000\ 1001\ 1000\ 0010$

Fibonacciho kód

kódování přirozených čísel (větších než 0)

předpoklad: menší čísla — častější výskyt

Fibonacciho čísla (řádu 2): $F_{-1} = F_0 = 1$,
 $F_i = F_{i-1} + F_{i-2}$ pro $i = 1, 2, 3, \dots$,

$F_0, F_1, F_2, \dots$ použijeme jako „váhy“ zápisu čísel:

i	...	5	4	3	2	1	0	-1
F_i	...	13	8	5	3	2	1	1
1							1	
2						1	0	
3					1	0	0	
4					1	0	1	
5			1	0	0	0	0	
6			1	0	0	1		
7				1	0	1	0	
8			1	0	0	0	0	
9			1	0	0	0	1	
...	...							

v zápisech nejsou 2 jedničky vedle sebe

Fibonacciho kód (řádu 2):

- obrátit pořadí bitů (v předchozích zápisech)
- přidat 1 na konec $\Rightarrow$ prefixový kód

	1	2	3	5	8	...
1	1	1				
2	0	1	1			
3	0	0	1	1		
4	1	0	1	1		
5	0	0	0	1	1	
6	1	0	0	1	1	
7	0	1	0	1	1	
8	0	0	0	0	1	1
9	1	0	0	0	1	1
...	...	...	...	...	...	...

Shannon – Fanoův kód

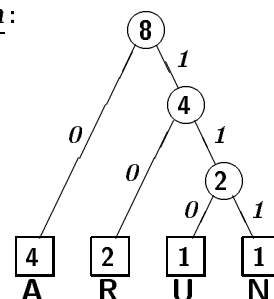
- seřadit znaky podle četnosti (abs. nebo rel.)
- dělit podle četnosti na poloviny (pokud možno)
- 0 $\leftarrow$ $\rightarrow$ 1 (nebo naopak)

Př.: příkl. „ARARAUNA“

A	R	U	N
4	2	1	1
1/2	1/4	1/8	1/8
0	1..		
	10	11.	
		110	111

A	0
R	10
U	110
N	111

tzv. kódový strom:

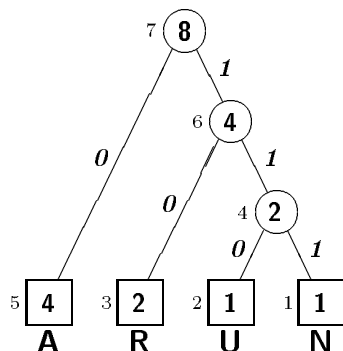


Huffmanův kód

kódový strom:

- seřadit znaky podle četnosti (abs. nebo rel.)
- znaky $\rightarrow$ uzly ohodnocené četností
- nové uzly:
dvojice minimálně ohodnocených uzlů
ohodnotit součtem ohodnocení

Př.: příkl. „ARARAUNA“



A	0
R	10
U	110
N	111

tzv. sourozenecká vlastnost

Kódy Shannon – Fanoův a Huffmanův

představitelé statických metod

- velmi malá až minimální redundance

Př.:

znak	četnost	S-F	H
A	11	00	0
B	5	01	100
C	5	10	101
D	4	110	110
E	3	111	111

28 znaků: Shannon – Fano 63 bitů
Huffman 62 bitů

semiadaptivní (hybridní) kódování

- dva průchody:
1. určení četností dílčích zpráv $\rightarrow$ kód
2. vlastní kódování
- s daty je nutno přenést/zapsat
i kódovací tabulku

metoda FGK

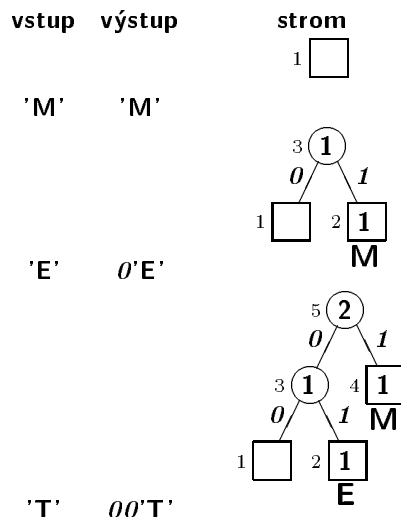
(Faller – Gallager – Knuth)

postupné vytváření a modifikace kódového stromu:

- na počátku tvoří strom jen „zvláštní“ uzel — uzel (list) ohodnocený 0: *nulový list*
- nový znak na vstupu:
 - ◊ vyšle se kód nulového listu (poprvé nic)
 - ◊ vyšle se vstupující znak (např. v kódu ASCII)
 - ◊ nulový list se nahradí binárním podstromem:
 - o kořen ohodnocen 1
 - o levý list: nulový list
 - o pravý list přísluší znaku — ohodnocen 1
- „známý“ znak na vstupu:
 - ◊ vyšle se kód znaku
 - ◊ pro list, který znaku přísluší, a postupně pro všechny jeho předchůdce (směrem ke kořenu) se zamění uzel s podstromem (pokud existuje):
 - o jehož kořen je stejně ohodnocený uzel
 - o má vyšší číslo — uzly jsou číslovány zdola nahoru a zleva doprava
 - ◊ ohodnocení uzlu a všech jeho předchůdců se zvýší o 1

metoda FGP — příklad I.

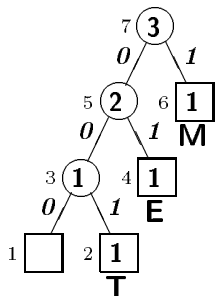
zakódovat řetěz znaků 'METEME_'



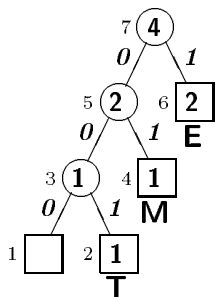
metoda FGP — příklad II.

vstup výstup strom

'T' 00'T'



'E' 01

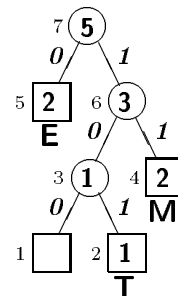


'M' 01

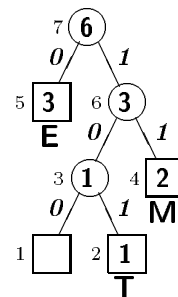
metoda FGP — příklad III.

vstup výstup strom

'M' 01



'E' 0



'_' 100'_'

Slovníkové metody

- orientovány na opakující se podřetězce, tzv. fráze, popř. digramy, trigramy, ...
- vytváří se „slovníky“ frází, ...

metoda LZ78

- ◇ položky ve slovníku číslovány od 1
- ◇ 0 ~ prázdný řetěz
- ◇ do slovníku se ukládá výstup ~ podřetěz
- ◇ ve slovníku se hledá nejdelší podřetěz na vstupu
- ◇ výstup: číslo položky & následující znak

Př.: zakódovat: **ARA_ARARAUNA_** (13 znaků)

	vstup	výstup
1	A	0 A
2	R	0 R
3	A_	1 _
4	AR	1 R
5	ARA	4 A
6	U	0 U
7	N	0 N
8	A_	3

Grupa

Grupa je množina G , nad jejímiž prvky je definována operace splňující dále uvedené axiomy. Pro tuto operaci je zde použita znaménko $\diamond$, např. $a \diamond b$.

$$\text{uzavřenost:} \quad (\forall a, b) \quad a \diamond b \in G \quad (\text{G1})$$

$$\text{asociativní zákon:} \quad (\forall a, b, c) \quad (a \diamond b) \diamond c = a \diamond (b \diamond c) \quad (\text{G2})$$

$$\text{existuje neutrální prvek:} \quad (\exists e) (\forall a) \quad a \diamond e = e \diamond a = a \quad (\text{G3})$$

$$\text{existuje opačný či inverzní prvek:} \quad (\forall a) (\exists a^*) \quad a \diamond a^* = a^* \diamond a = e \quad (\text{G4})$$

Abelova neboli **komutativní grupa** splňuje kromě toho ještě další axiom:

$$\text{komutativní zákon:} \quad (\forall a, b) \quad a \diamond b = b \diamond a \quad (\text{G5})$$

Pro označení operace se (místo znaménka $\diamond$) často používá stejné znaménko jako pro sčítání, resp. násobení, např. $a + b$, resp. $a \cdot b$. Pak bývá zvykem zapisovat neutrální prvek e jako 0, resp. 1, a opačný či inverzní prvek a^* jako $-a$, resp. a^{-1} . Nemusí jít přitom o čísla ani o sčítání či násobení,

Těleso

Těleso je množina T , na níž jsou definovány dvě operace, které jsou zde označovány stejně jako sčítání a násobení.

$$\text{uzavřenost sčítání:} \quad (\forall a, b) \quad a + b \in T \quad (\text{T1})$$

$$\text{asociativní zákon pro sčítání:} \quad (\forall a, b, c) \quad (a + b) + c = a + (b + c) \quad (\text{T2})$$

$$\text{komutativní zákon pro sčítání:} \quad (\forall a, b) \quad a + b = b + a \quad (\text{T3})$$

$$\text{existuje nulový prvek:} \quad (\exists 0) (\forall a) \quad a + 0 = 0 + a = a \quad (\text{T4})$$

$$\text{existuje opačný prvek:} \quad (\forall a) (\exists (-a)) \quad a + (-a) = (-a) + a = 0 \quad (\text{T5})$$

$$\text{uzavřenost násobení:} \quad (\forall a, b) \quad a \cdot b \in T \quad (\text{T6})$$

$$\text{asociativní zákon pro násobení:} \quad (\forall a, b, c) \quad (a \cdot b) \cdot c = a \cdot (b \cdot c) \quad (\text{T7})$$

$$\text{existuje jednotkový prvek:} \quad (\exists 1) (\forall a) \quad a \cdot 1 = 1 \cdot a = a \quad (\text{T8})$$

$$\text{existuje inverzní prvek:} \quad (\forall a \neq 0) (\exists a^{-1}) \quad a \cdot a^{-1} = a^{-1} \cdot a = 1 \quad (\text{T9})$$

$$\text{distributivní zákony:} \quad (\forall a, b, c) \quad a \cdot (b + c) = a \cdot b + a \cdot c \quad (\text{T10})$$

$$(\forall a, b, c) \quad (a + b) \cdot c = a \cdot c + b \cdot c \quad (\text{T11})$$

Komutativní těleso splňuje kromě toho ještě další axiom:

$$\text{komutativní zákon pro násobení:} \quad (\forall a, b) \quad a \cdot b = b \cdot a \quad (\text{T12})$$

Množina T spolu s operací označovanou znaménkem sčítání tvoří tedy komutativní grupu a bývá nazývána **aditivní grupou** tělesa T a množina T bez nulového prvku spolu s operací označovanou znaménkem násobení tvoří tedy jinou grupu a bývá nazývána **multiplikativní grupou** tělesa T .

Vektorový (lineární) prostor

Vektorový (neboli lineární) prostor V nad tělesem S je komutativní grupa V , na níž je navíc definována operace nazývaná násobení vektoru skalárem, která splňuje dále uvedené axiomy. Prvkům množiny V se říká **vektory**, prvkům tělesa S se říká **skaláry**. V tělese S existuje (musí existovat) nulový prvek 0 a jednotkový prvek 1 — viz (T4) a (T8). Operaci, která spolu s množinou V tvoří grupu, se říká **sčítání vektorů** a je označována jako sčítání. Operace **násobení vektoru skalárem** je označována jako násobení. Sčítání vektorů tedy splňuje axiomy (G1) až (G5) s tím, že znaménko $\diamond$ je nahrazeno znaménkem $+$. Kromě toho vektorový prostor splňuje tyto axiomy:

$$\text{asociativní zákon:} \quad (\forall s \in S)(\forall t \in S)(\forall v \in V) \quad s \cdot (t \cdot \mathbf{v}) = (s \cdot t) \cdot \mathbf{v} \quad (\text{V1})$$

$$\text{násobení jednotkovým skalárem:} \quad (\forall \mathbf{v} \in V) \quad 1 \cdot \mathbf{v} = \mathbf{v} \quad (\text{V2})$$

$$\text{distributivní zákon I.:} \quad (\forall s \in S)(\forall u \in V)(\forall v \in V) \quad s \cdot (\mathbf{u} + \mathbf{v}) = s \cdot \mathbf{u} + s \cdot \mathbf{v} \quad (\text{V3})$$

$$\text{distributivní zákon II.:} \quad (\forall s \in S)(\forall t \in S)(\forall v \in V) \quad (s + t) \cdot \mathbf{v} = s \cdot \mathbf{v} + t \cdot \mathbf{v} \quad (\text{V4})$$

Z těchto axiomů vyplývá mj. zákon o **násobení nulovým skalárem**

$$(\forall \mathbf{v} \in V) \quad 0 \cdot \mathbf{v} = \mathbf{0},$$

kde $\mathbf{0}$ je nulový vektor (tzn. nulový prvek grupy V).

Množiny V a S nemusí mít nic společného. Někdy však bývá vektorem $\mathbf{v}$ uspořádaná n -tice skalárů $\mathbf{v} = (v_1, \dots, v_n)$. Pak bývá definován tzv. **skalární součin** $\mathbf{u} \cdot \mathbf{v}$ vektorů $\mathbf{u} = (u_1, \dots, u_n)$ a $\mathbf{v} = (v_1, \dots, v_n)$ takto:

$$\mathbf{u} \cdot \mathbf{v} = u_1 \cdot v_1 + \dots + u_n \cdot v_n.$$

(Znaménka na pravé straně označují příslušné operace v tělese S .)