Paměti, aritmetické obvody

Prof. Ing. Ondřej Novák, CSc.
ITE

Rozdělení pamětí

podle možností změny dat:

- **obsah lze libovolně přepisovat ..... RAM (Random Acces Memory)**
  - **SRAM ... statická**
  - **DRAM ... dynamická**
- **permanentní**
  - **obsah určen při výrobě .... ROM (Read Only Memory)**
  - **obsah lze jednorázově naprogramovat .... PROM**
- **semipermanentní**
  - **obsah lze naprogramovat a lze vymazat a přeprogramovat .... EPROM, EEPROM, FLASH**
- **volatilní ... energeticky závislé (uložená informace zanikne po vypnutí napájení) .... SRAM, DRAM**
- **nonvolatilní .... ROM, PROM, EPROM, EEPROM, FLASH**

Moderní paměťové technologie

- **Flash NAND, NOR**
- **MRAM (magnetic RAM), nonvolatilní technologie, omezený počet zápisů**
- **RRAM (Resistive RAM) – změna resistivity vlivem přítomnosti nebo absence vodivých atomů**
- **NRAM (Carbone nanotubes) – změna resistivity po průchodu proudu CN trubičkou**

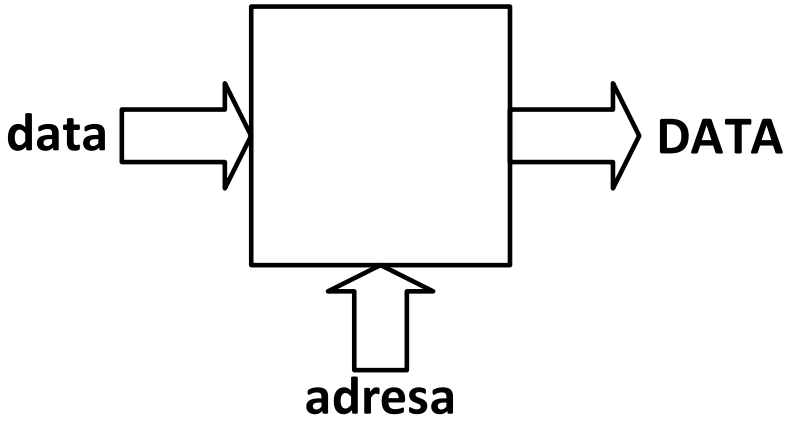
Rozdělení pamětí

podle způsobu výběru položek:

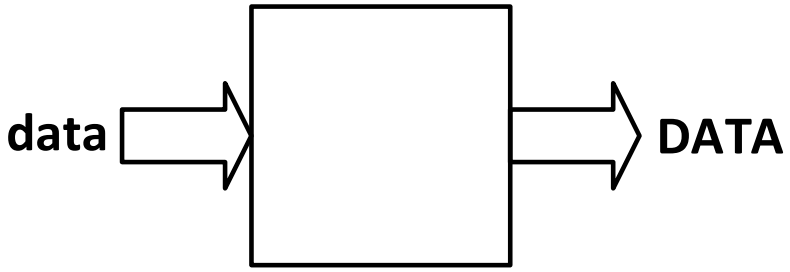
- **s adresovým výběrem (adresové, adresovatelné)**
- **s postupným výběrem (sériové)**
- **asociativní (výběr podle části uložené informace, tzv. klíče ... CAM ... content adressable memory)**
- **zásobník (LIFO ... last in first out)**
- **fronta (FIFO ... first in first out)**

Rozdělení pamětí

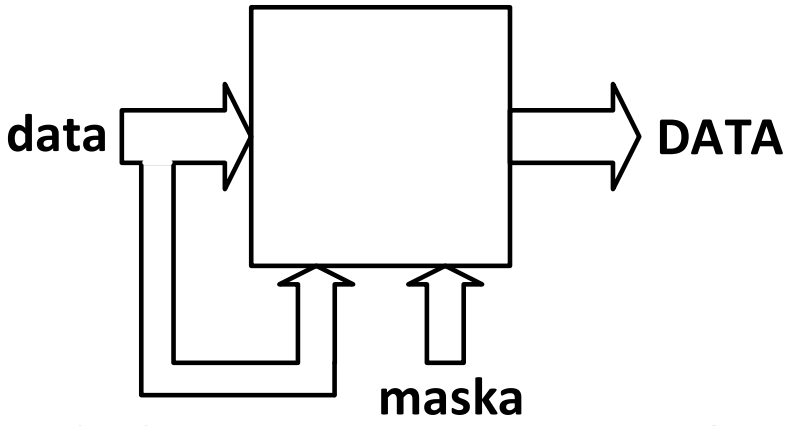
adresovatelná



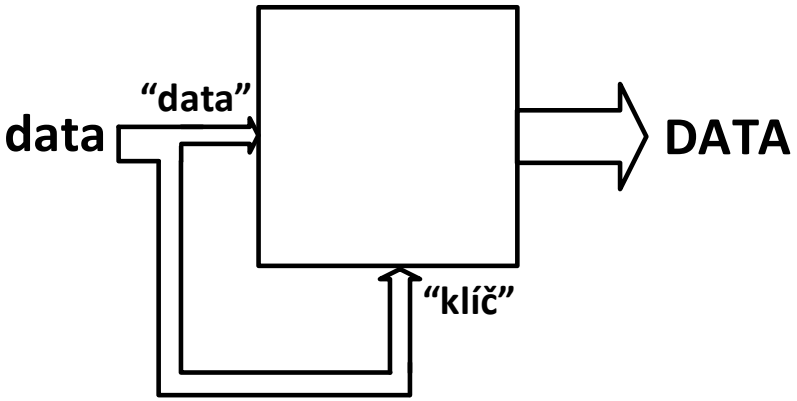
LIFO, FIFO – s postupným výběrem



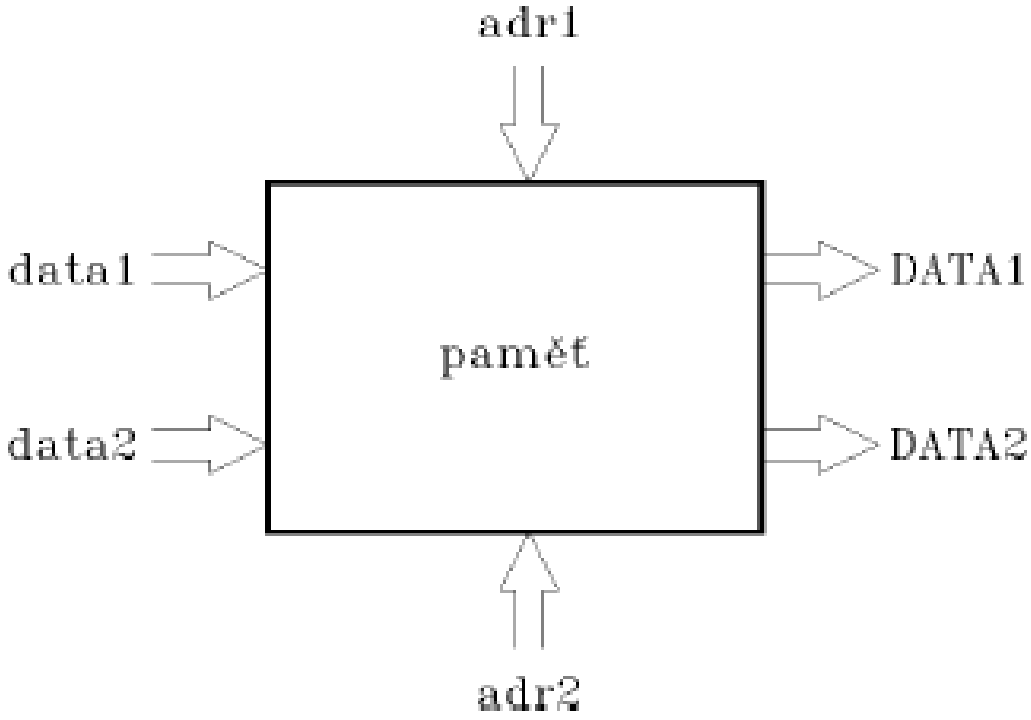
„univerzální“ CAM



„speciální“ CAM



Adresovatelná paměť vícebránová



Základní pojmy

paměťová buňka ... základní stavební blok paměti, slouží k záznamu jednoho bitu

paměťové místo ... skupina paměťových buněk které lze najednou zapisovat nebo číst (šířka slova paměti)

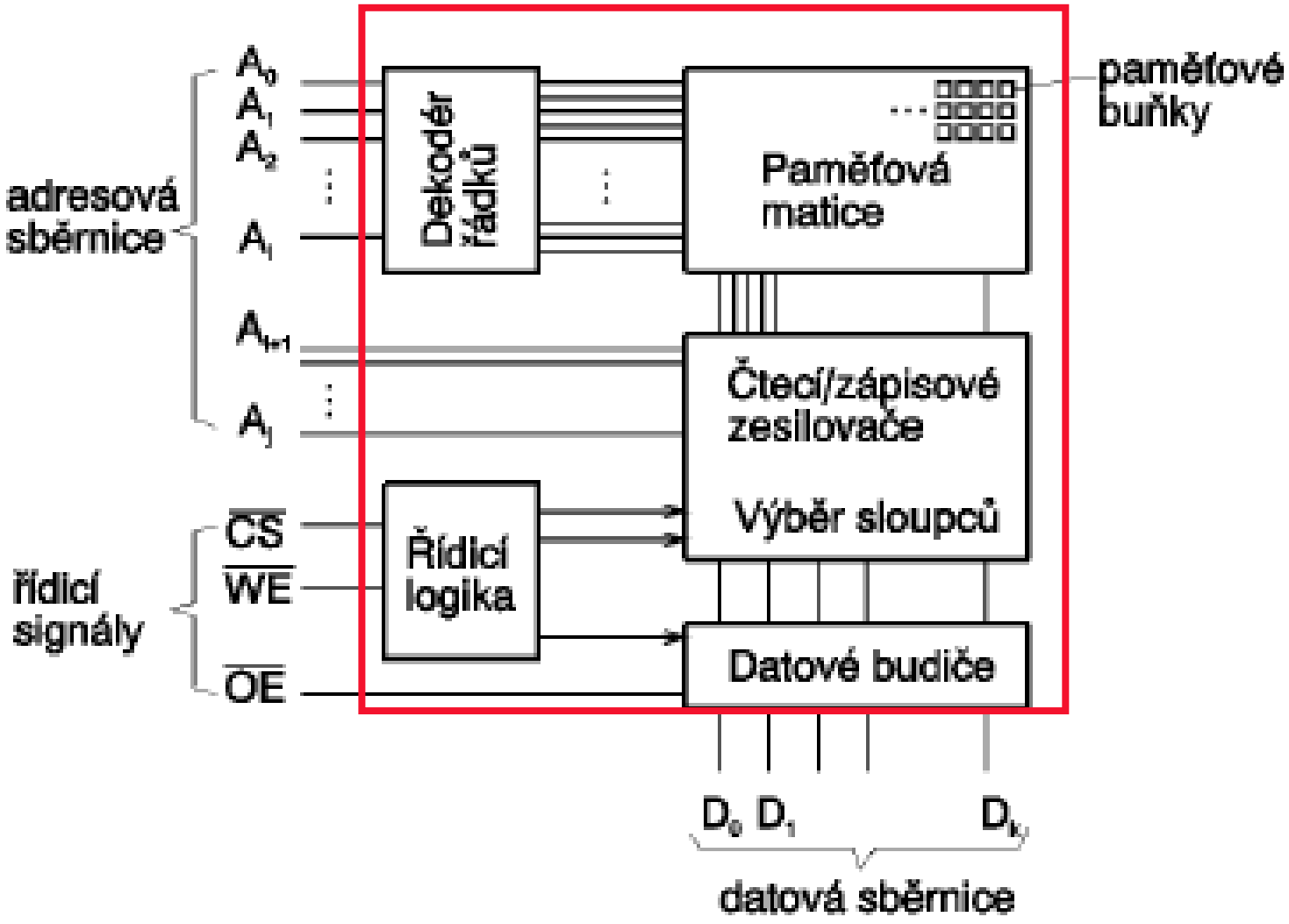
položka ... obsah paměťového místa

adresa ... číselné označení (index) paměťového místa jimž lze vybírat jednotlivé položky

kapacita paměti ... počet položek

paměťová matice ... skupina paměťových míst uspořádaná tak, že je lze vybírat adresou

Blokové schéma typického paměťového obvodu



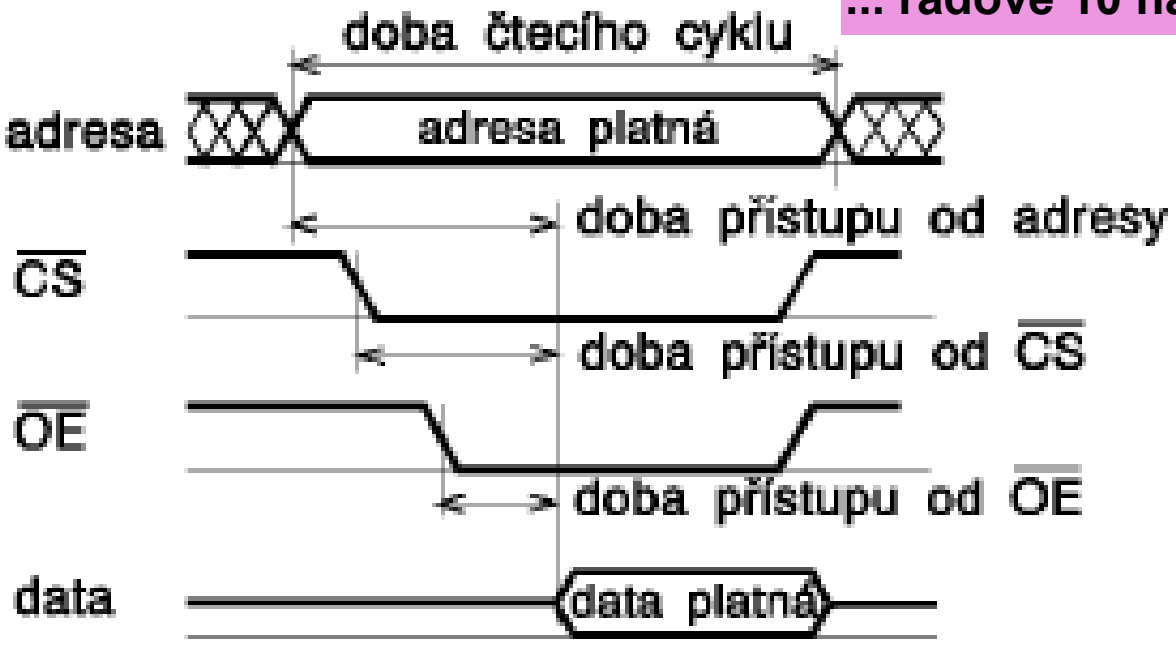
Řídicí signály

- $\overline{\mathbf{OE}}$ - **o**utput **e**nable ... aktivace výstupních třístavových budičů datové sběrnice
- $\overline{\mathbf{WE}}$ - **w**rite **e**nable ... povolení zápisu
- $\overline{\mathbf{CS}}$ - **c**hip **s**elect ... výběr čipu - podmiňuje provedení zápisu nebo čtení

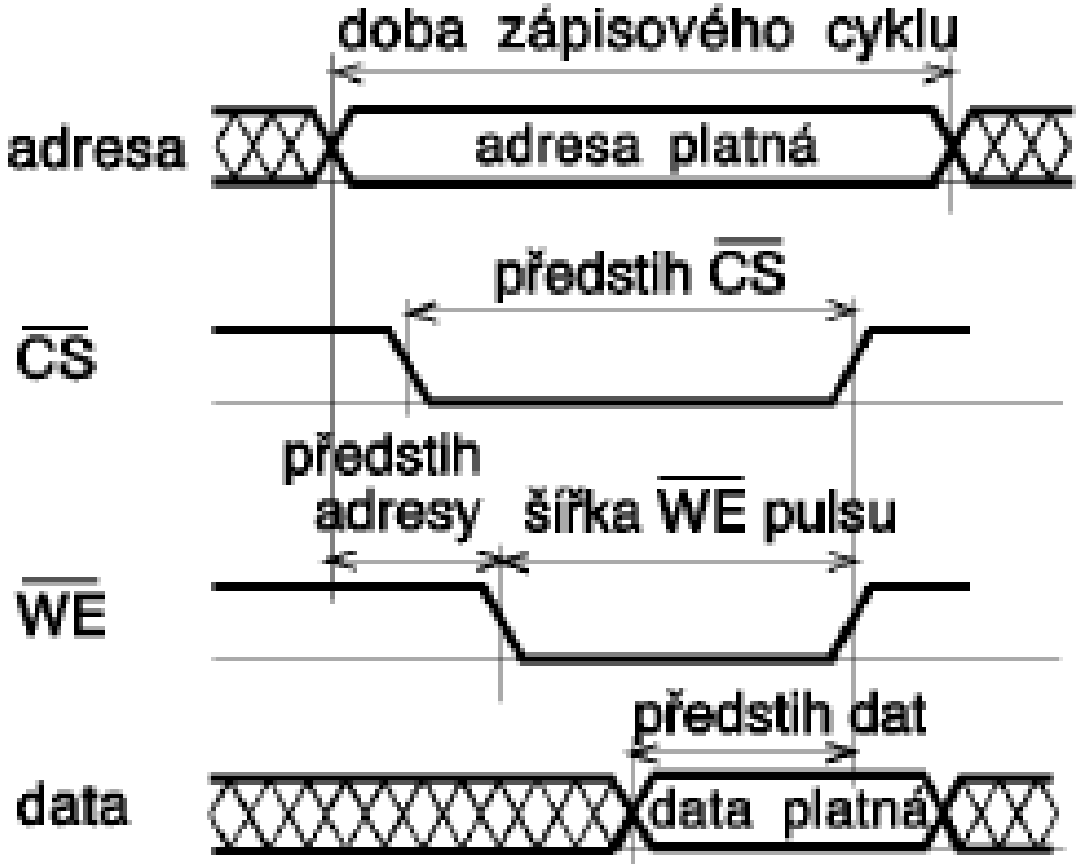
Chování paměťových obvodů, parametry ..... čtení

Typický průběh signálů pro operaci čtení ze statické paměti SRAM (je nutné dodržet minimální nebo i maximální zpoždění mezi aktivací jednotlivých signálů a dalšími událostmi)

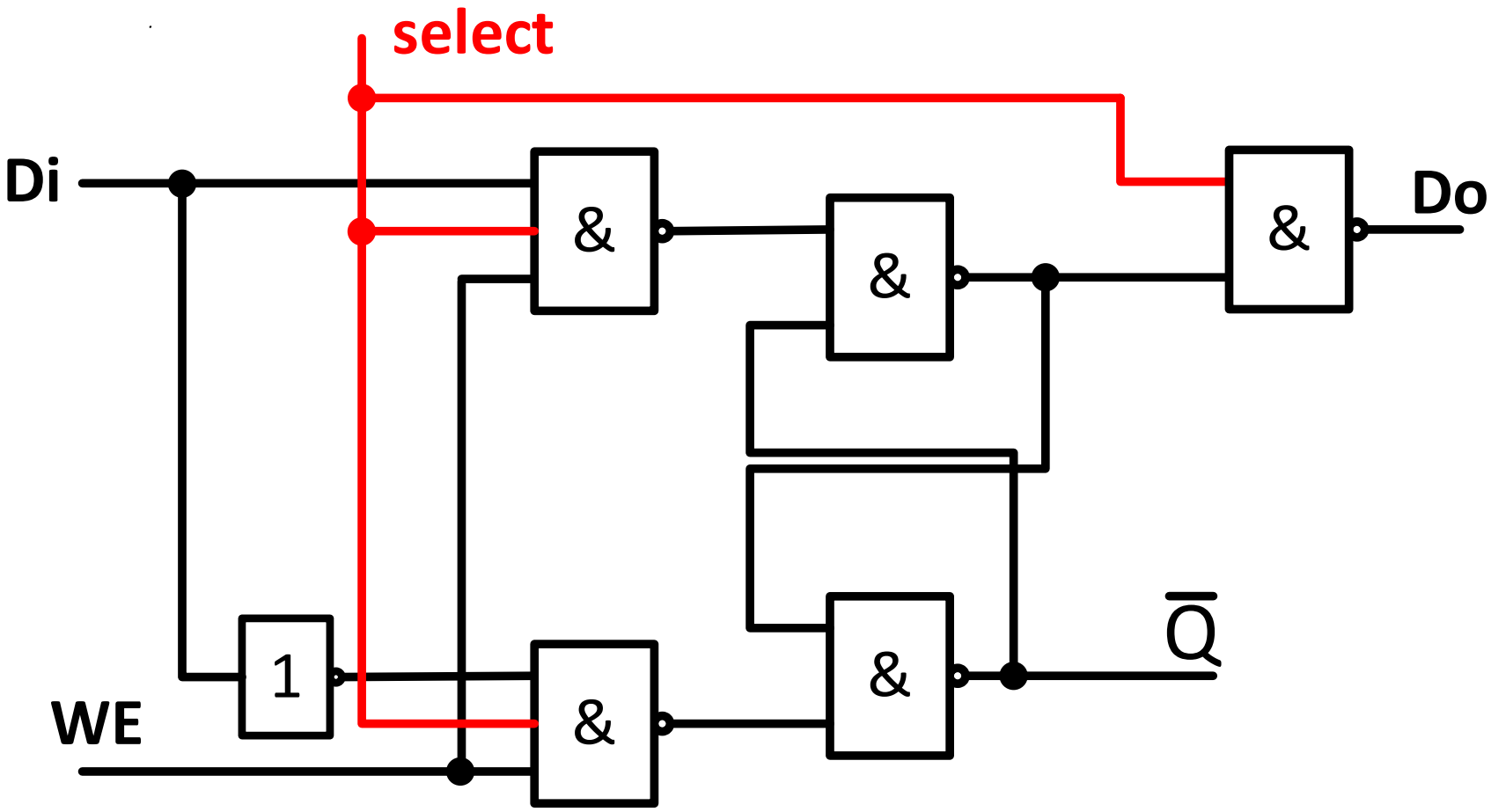
... řádově 10 nanosekund



Chování paměťových obvodů, parametry ..... zápis



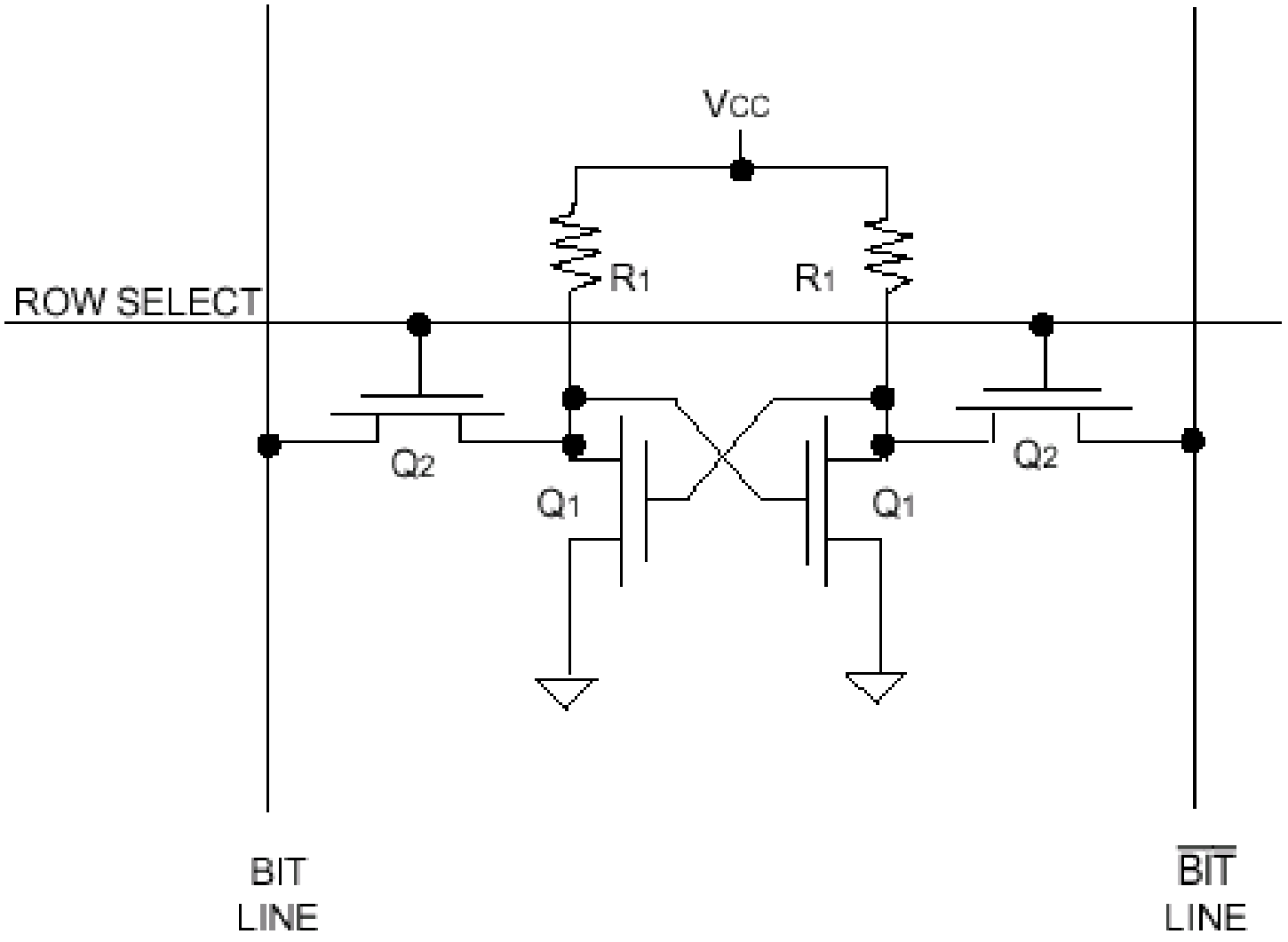
Statická RAM - základní princip



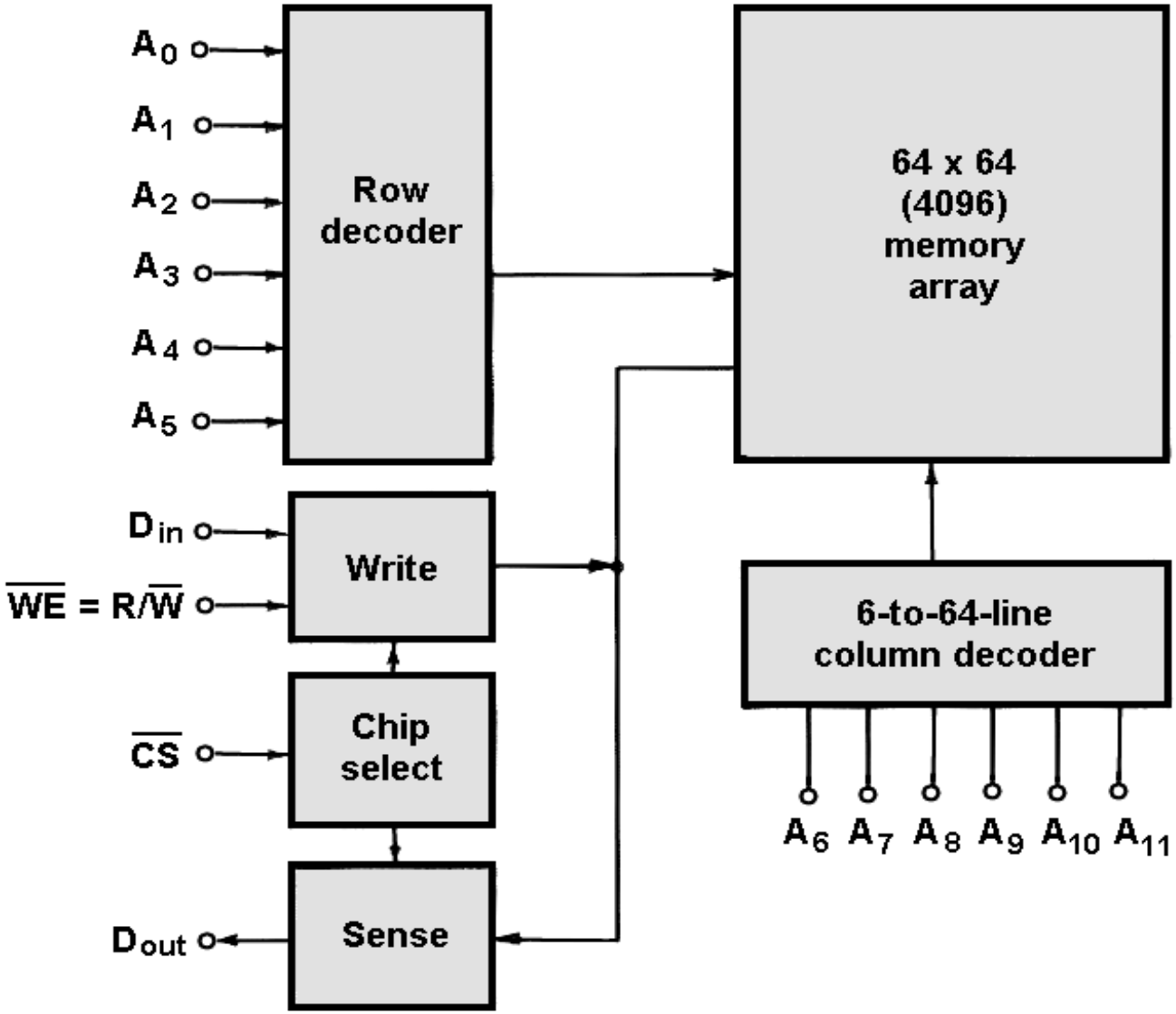
Statické paměti

- Synchronní x asynchronní
- Jednoportové x víceportové

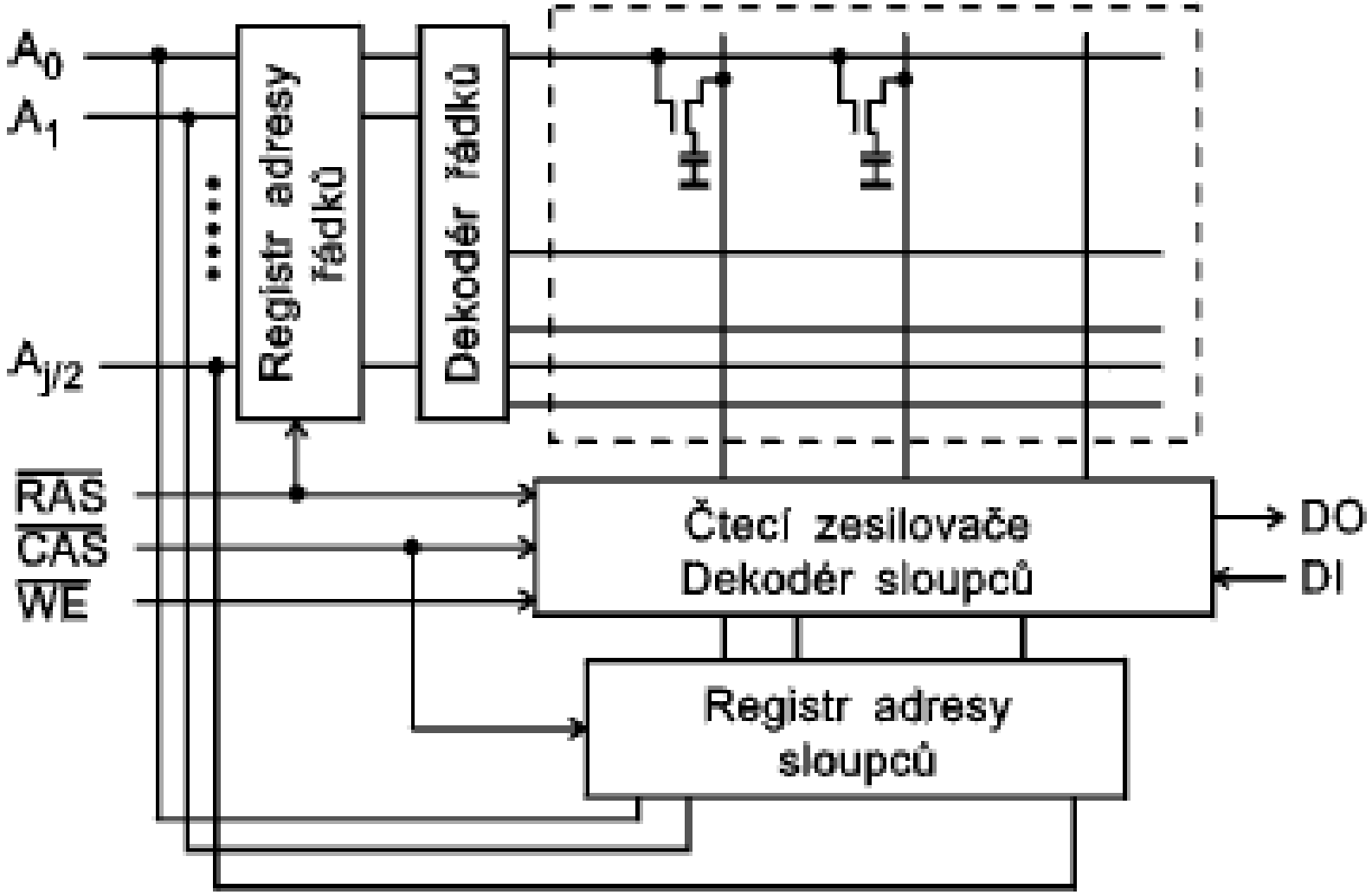
Standardní 2 transistorová statická buňka



Uspořádání paměti



Konstrukční princip DRAM



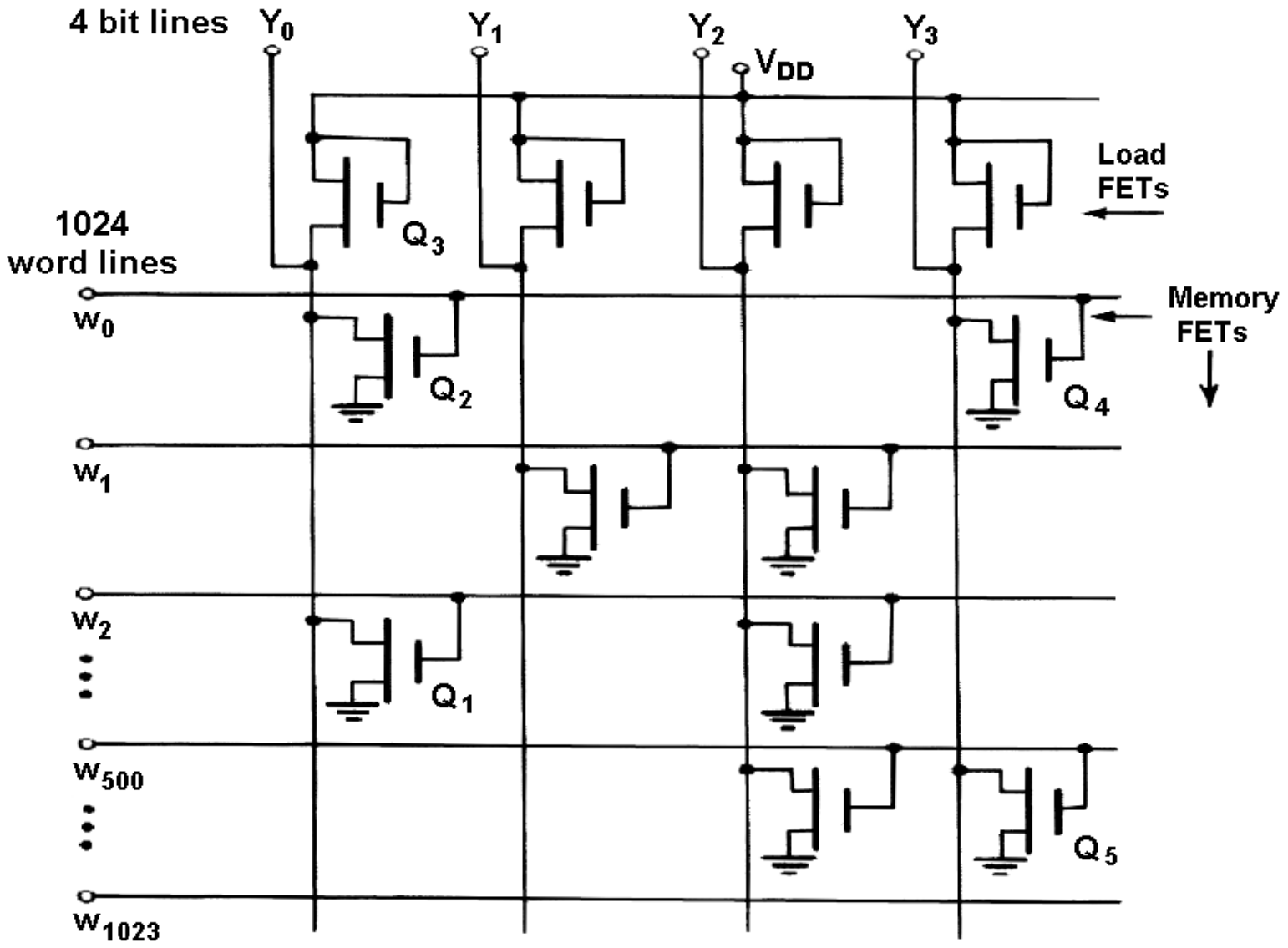
Konstrukční princip DRAM

- **Adresa je časově multiplexována, polovina adresy při $\overline{RAS} = 0$ (řádek), druhá polovina adresy při $\overline{CAS} = 0$ (sloupec)**
- **<u>Zápis</u>: Na datový (sloupcový) vodič se přivede zapisovaná úroveň a aktivuje se zvolený řádek. Paměťový kondenzátor se nabije, nebo vybije.**
- **<u>Čtení</u>: Při výběru řádku se kondenzátory vybíjí do vstupů čtecích zesilovačů (čtení je destruktivní, přečtenou informaci je nutno bezprostředně zapsat zpět).**
- **<u>Obnovení</u>: Stejně jako čtení. Všechny čtecí zesilovače jsou umístěny ve všech sloupcích, obnovují se všechny sloupce jednoho řádku najednou.**

Porovnání vlastností SRAM a DRAM

- **cena** .... DRAM je při stejné kapacitě levnější než SRAM (1 transistor x 6 transistorů na paměťovou buňku)
- pro refresh (**obnovení**) potřebují DRAM další obvody (periodické generování adres řádků)
- čtení DRAM je **destruktivní**, po čtení musí být informace znovu zapsána ... delší čtecí cyklus
- DRAM mají větší **spotřebu** v klidovém stavu (obnovování)
- (mýtus **rychlosti** ... závisí hlavně na velikosti paměti)

Paměť ROM


4 bit lines
Y0
Y1
Y2
Y3
VDD
Load FETs
1024 word lines
Q3
W0
Q2
Q4
Memory FETs
W1
W2
Q1
W500
Q5
W1023

Polyadické číselné soustavy

z-adické číslo v z-adické soustavě:

$A = a_n z^n + \ldots + a_0 z^0 + a_{-1} z^{-1} + \ldots + a_{-m} z^{-m}$

z je základ polyadické soustavy $z \geq 2$

(nejčastěji z = 2, 8, 10, 16)

a_i je z-adická číslice $0 \leq a_i < z$ („z“ různých a_i)

$a_{min} = 0$ $a_{max} = z - 1$

$A_{min} = 0$ $A_{max} = z^{n+1} - z^{-m}$

mezi řádem „0“ a „-1“ píšeme desetinnou čárku

standardně uvažujeme jen nezáporná čísla

Zobrazování záporných čísel

a) **Přímý kód** – přidáme znaménkový bit
+ … 0 - … 1 (např.: –6 … 1 0110)
složité aritmetické operace – nepoužívá se

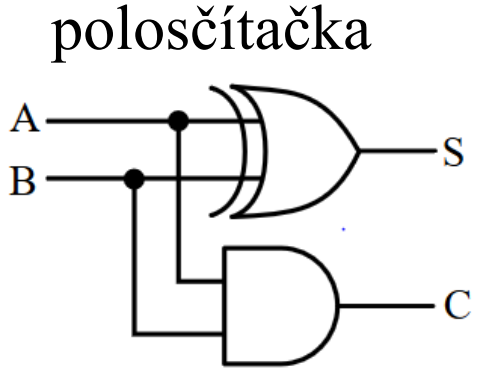
b) **Jednotkový doplněk** (inverzní kód)
$1D(A) = z^n – A –1$
např. A = 10101, z = 2, n = 5, 1D(A) = 01010
problém dvou nul: $0 = (0000)_2$, $-0 = (1111)_{1D}$

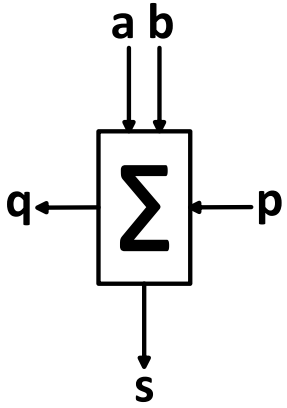
c) **Dvojkový doplněk**
$2D(A) = z^n – A = 1D(A) +1$
např. A = 356, z = 10, n = 4, 2D(A) = 9644

Sčítání a sčítačky

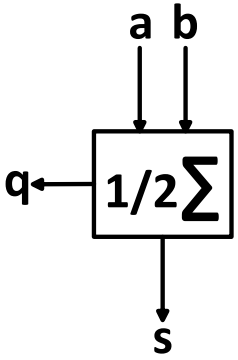
Sčítání:

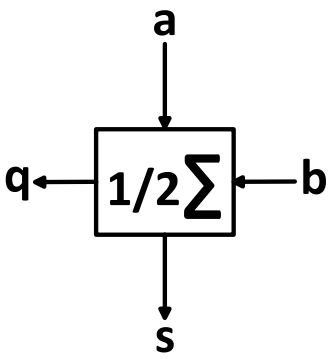
$$\begin{array}{rr} (01011)_2 & (11)_{10} \\ \underline{(00110)_2} & \underline{(\ 6)_{10}} \\ (10001)_2 & (17)_{10} \end{array}$$



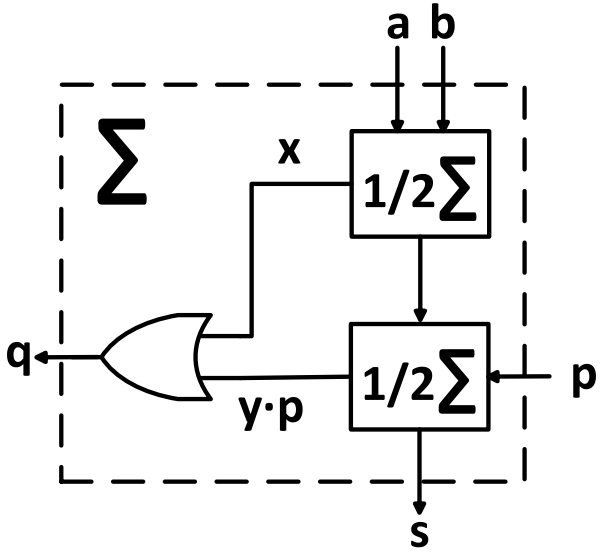


Úplná sčítačka



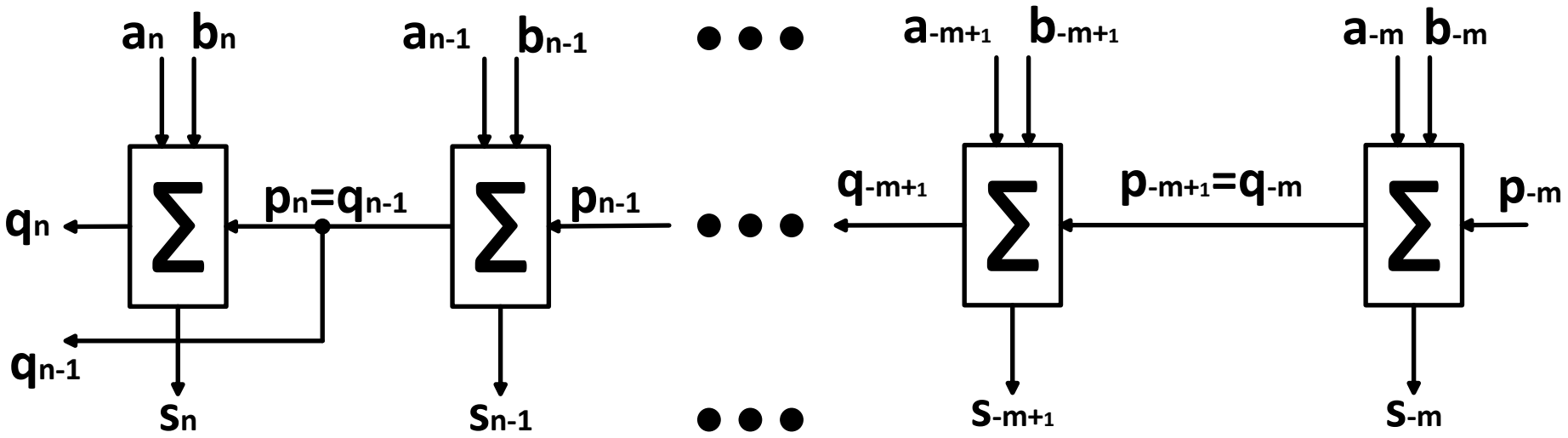


Polosčítačka

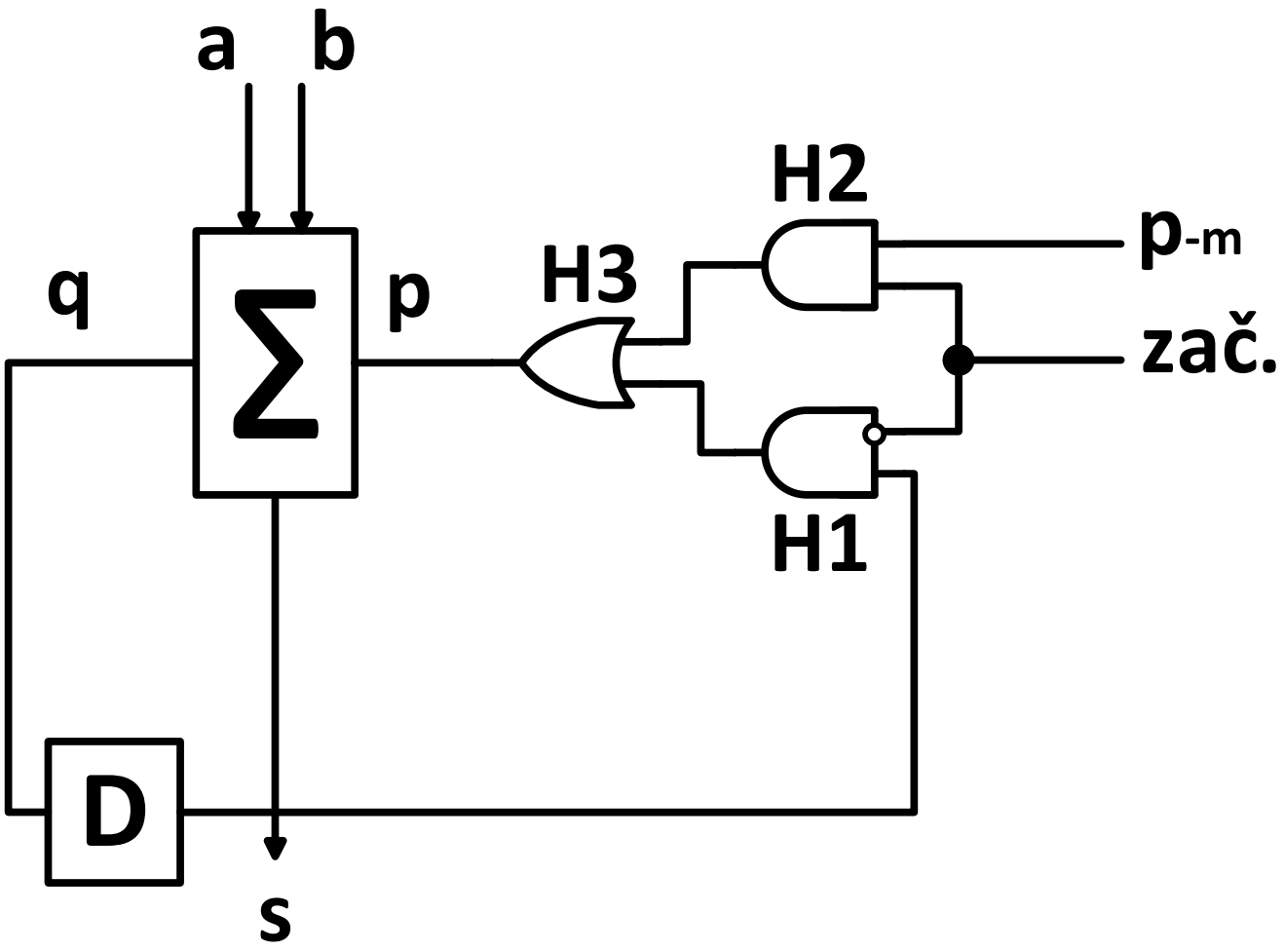


Realizace úplné sčítačky z polosčítaček

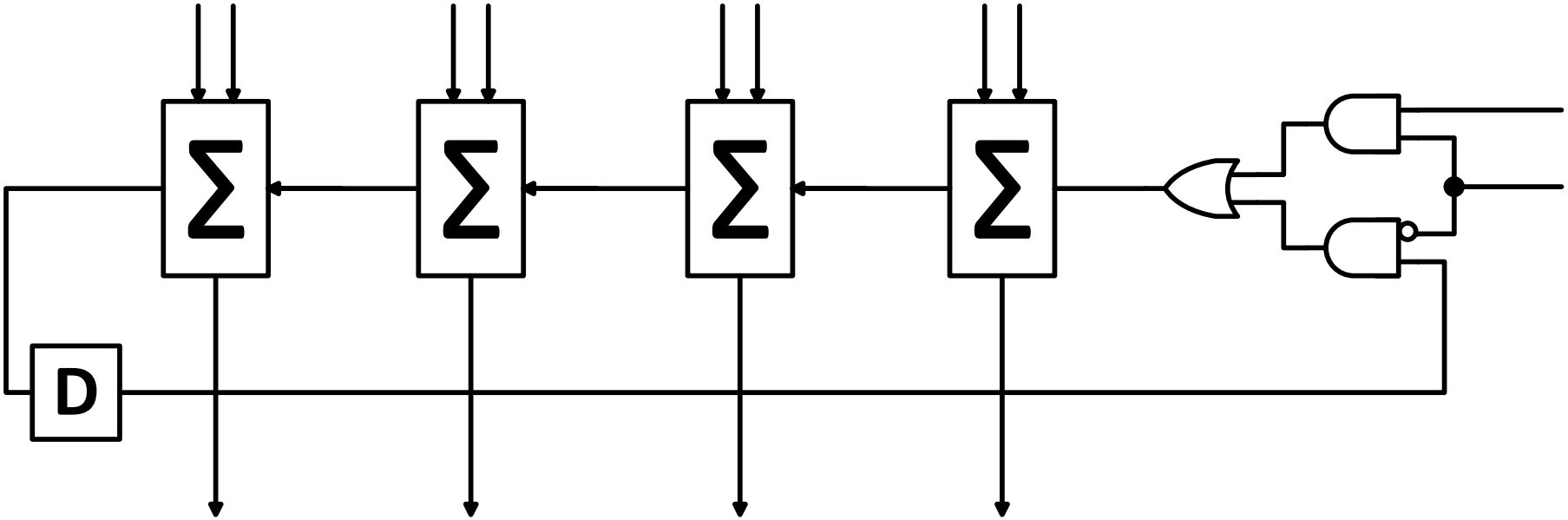
Paralelní dvojková sčítačka



Sériová dvojková sčítačka



Sério-paralelní dvojková sčítačka



Sčítačka pro nezáporná čísla

- p_i .... přenos do řádu i .....*carry*
- q_i .... přenos z řádu i
- $M = 8 = 2^3 = 1000_2 = (111 + 1)_2$

$S = A + B$ *je-li* $q^* = 0$

$S = A + B - M$ *je-li* $q^* = 1$

- *zde:* q^* .... přeplnění .....*overflow*

sčítačka s predikcí přenosů (angl. carry look-ahead adder)

i

$q = ab + ap + bp$

$a = 1$ a $b = 0$ nebo
$a = 0$ a $b = 1$
$\Rightarrow q = p$! v ustáleném stavu !

zpoždění se kumulují $\Rightarrow$ frekvence hodinových pulsů

$P_i = a_i \oplus b_i$ přenos „prochází“ řádem i

$G_i = a_i \cdot b_i$ přenos v řádu i vzniká — generuje se v něm

srov. výstupy půlsčítačky !

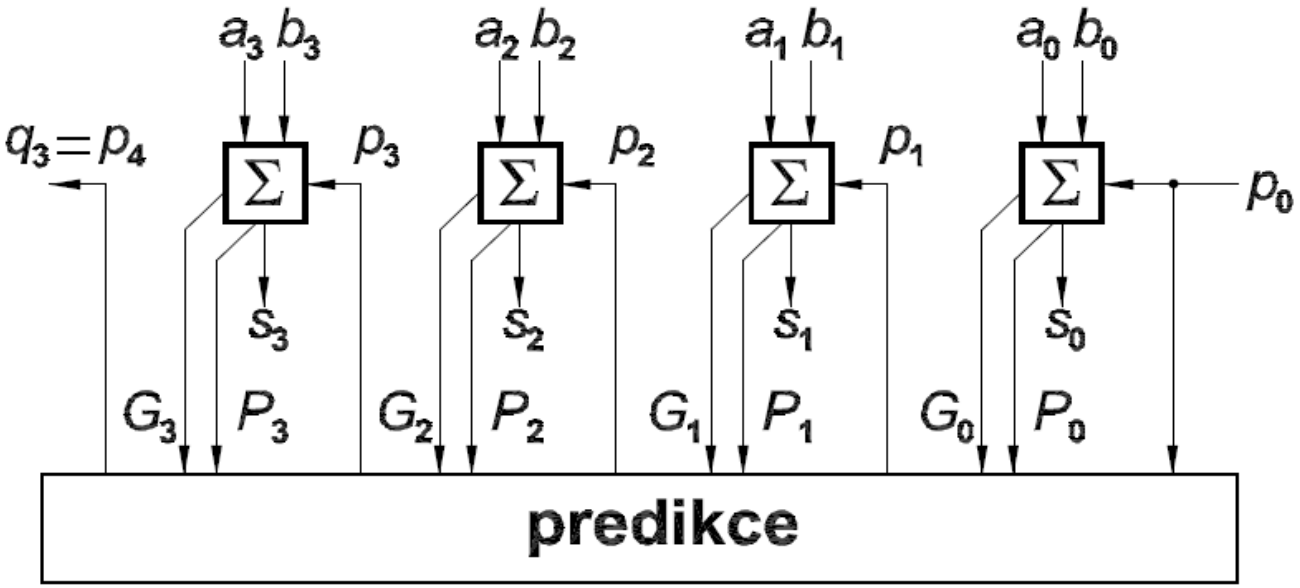
$q_0 = p_1 = G_0 + P_0 \cdot p_0$ $\qquad$ $q_1 = p_2 = G_1 + P_1 \cdot p_1$

$q_1 = p_2 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot p_0$

$q_2 = p_3 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot p_0$

atd.

pozn.: alternativně lze použít $P_i = a_i + b_i$



predikce:

$$p_1 = G_0 + P_0 \cdot p_0$$
$$p_2 = G_1 + P_1 \cdot G_0 + P_1 \cdot P_0 \cdot p_0$$
$$p_3 = G_2 + P_2 \cdot G_1 + P_2 \cdot P_1 \cdot G_0 + P_2 \cdot P_1 \cdot P_0 \cdot p_0$$
$$p_4 = G_3 + P_3 \cdot G_2 + P_3 \cdot P_2 \cdot G_1 + P_3 \cdot P_2 \cdot P_1 \cdot G_0 + P_3 \cdot P_2 \cdot P_1 \cdot P_0 \cdot p_0$$

Odečítání

Použitím dvojkového doplňku:

změníme nuly na jedničky (a naopak) a přičteme jedničku, přenos se ztrácí

	$(01011)_2$	$(11)_{10}$
	$\underline{(11010)_{2D}}$	$\underline{(-\ 6)_{10}}$
1	$(00101)_2$	$(\ \ 5)_{10}$

Je-li výsledek záporný, jeho absolutní hodnotu získáme opět dvojkovým doplňkem

Odečítání

- sčítání .... přenos q^* .....carry
- odčítání .... výpůjčka v^* .....borow

$$\overline{q^*} = v^*$$

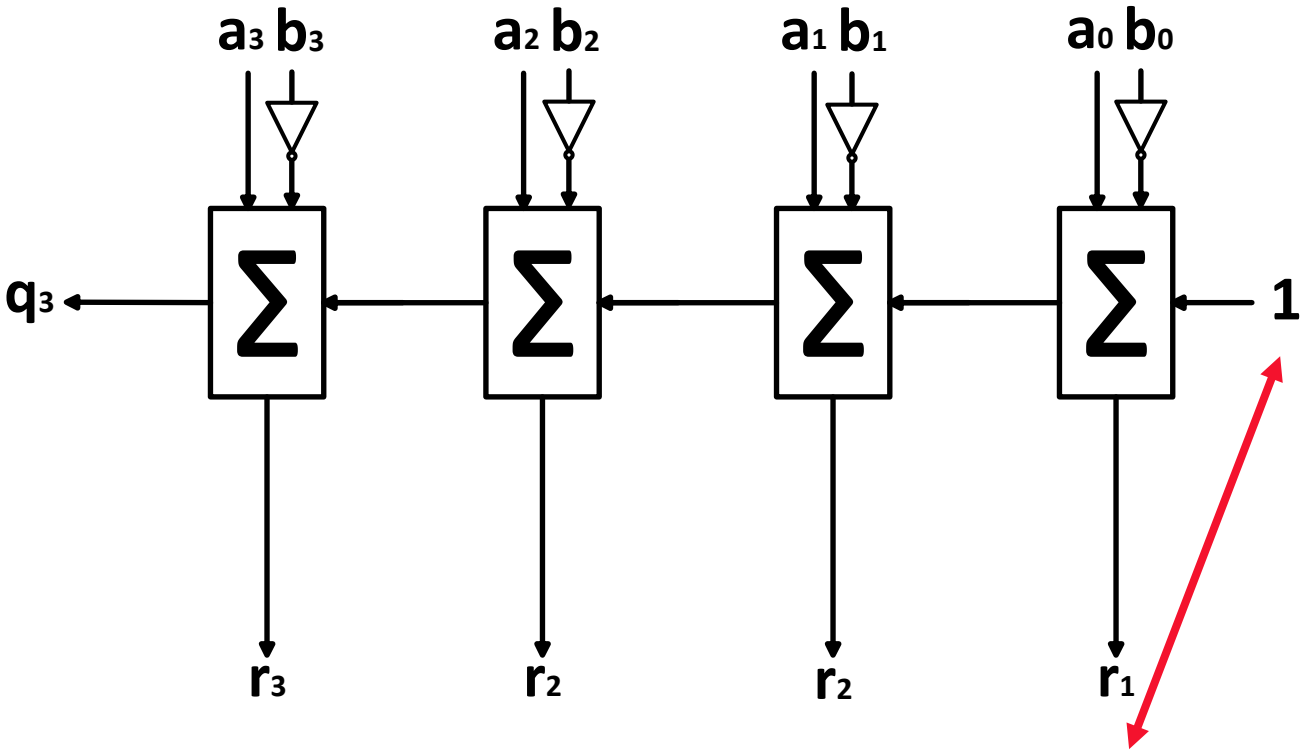
$$q^* = 1 \leftrightarrow v^* = 0 \leftrightarrow A - B \geq 0$$

$$q^* = 0 \leftrightarrow v^* = 1 \leftrightarrow A - B < 0$$

Sčítačka-odčítačka pro nezáporná čísla

- složitější řízení, protože detekce přeplnění záleží na operaci
- sčítání ... přeplnění (zde přenos) pro $q^*=1$
- odčítání .... přeplnění (zde výpujčka) pro $v^*=1$, $q^*=0$

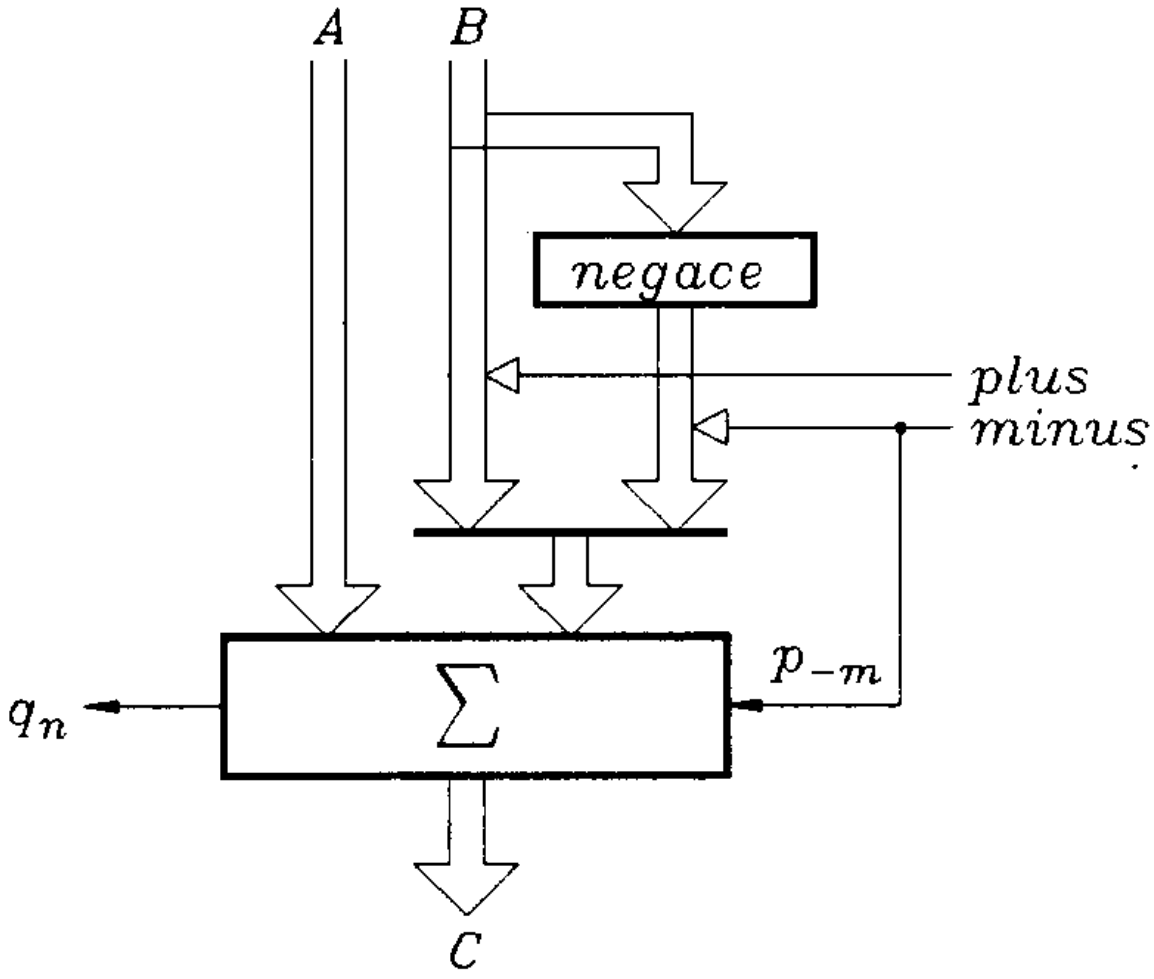
Odčítačka pro nezáporná čísla



p**= *1 .... horká jednička
hot one

R* = *A* - *B, je-li *q* = 1*

Sčítačka - odečítačka



Doplňkový kód

Vychází z dvojkového doplňku:

$D(A) = A$ (pro $A \geq 0$) a $z^n - A$ (pro $A < 0$)

Ve dvojkové soustavě bude mít obraz nezáporného čísla v nejvyšším řádu 0, obraz záporného čísla 1.

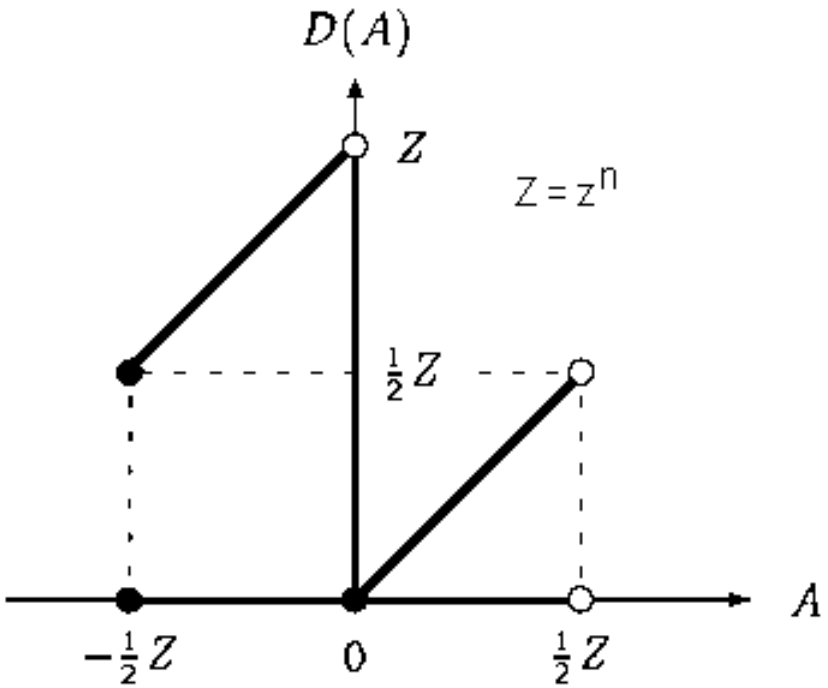
Nutno vyloučit stejné obrazy (dvojznačnost)

Doplňkový kód

Proto předpokládáme, že A splňuje podmínku:

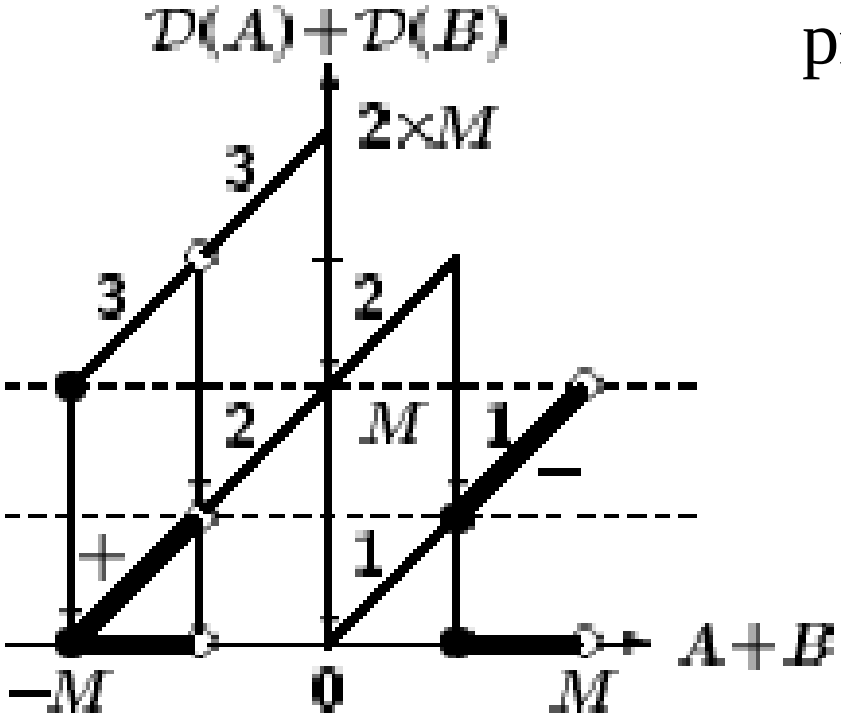
$-0{,}5.z^n \leq A < 0{,}5.z^n$

„z“ uvažujeme sudé (2, 8, 10, 16) $\Rightarrow$ $0{,}5.z^n$ je celé

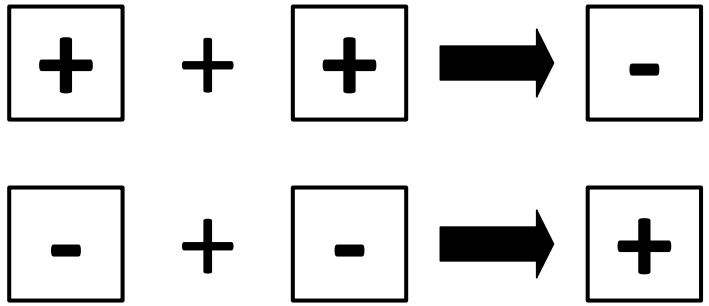


Přeplnění

Přeplnění (overflow) není přenos (carry) !!!!!



přeplnění:



Pro dvojkovou soustavu:
XOR přenosů do a
z nejvyššího řádu

Sčítačka-odčítačka v doplňkovém kódu (včetně detekce přeplnění)

přeplnění: + + + ➡ -

- + - ➡ +

tzn. v nejvyšším řádu sčítačky bude:
a=0 b=0 s=1
nebo
a=1 b=1 s=0

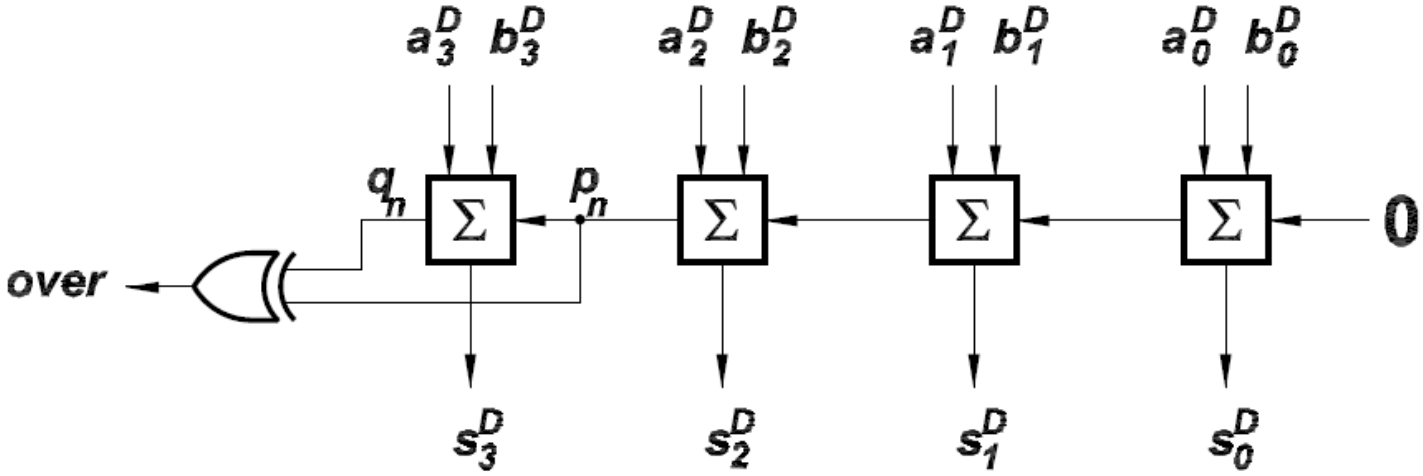
$p \neq q$

$$over = p \oplus q$$

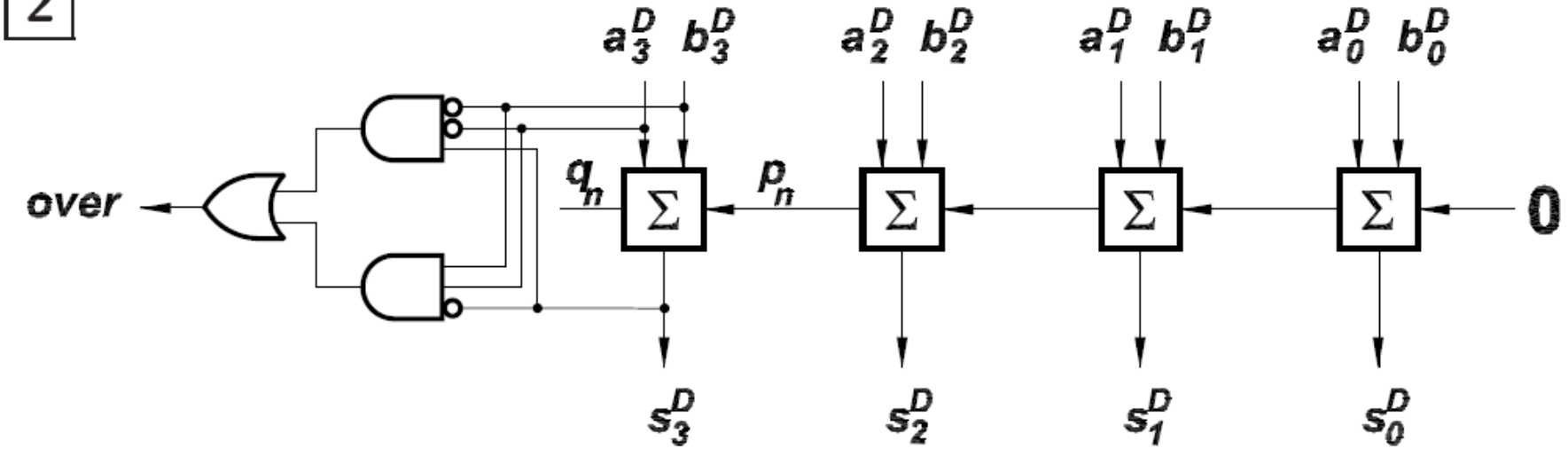
nebo také:

$$over = \overline{a}\,\overline{b}s + ab\overline{s}$$

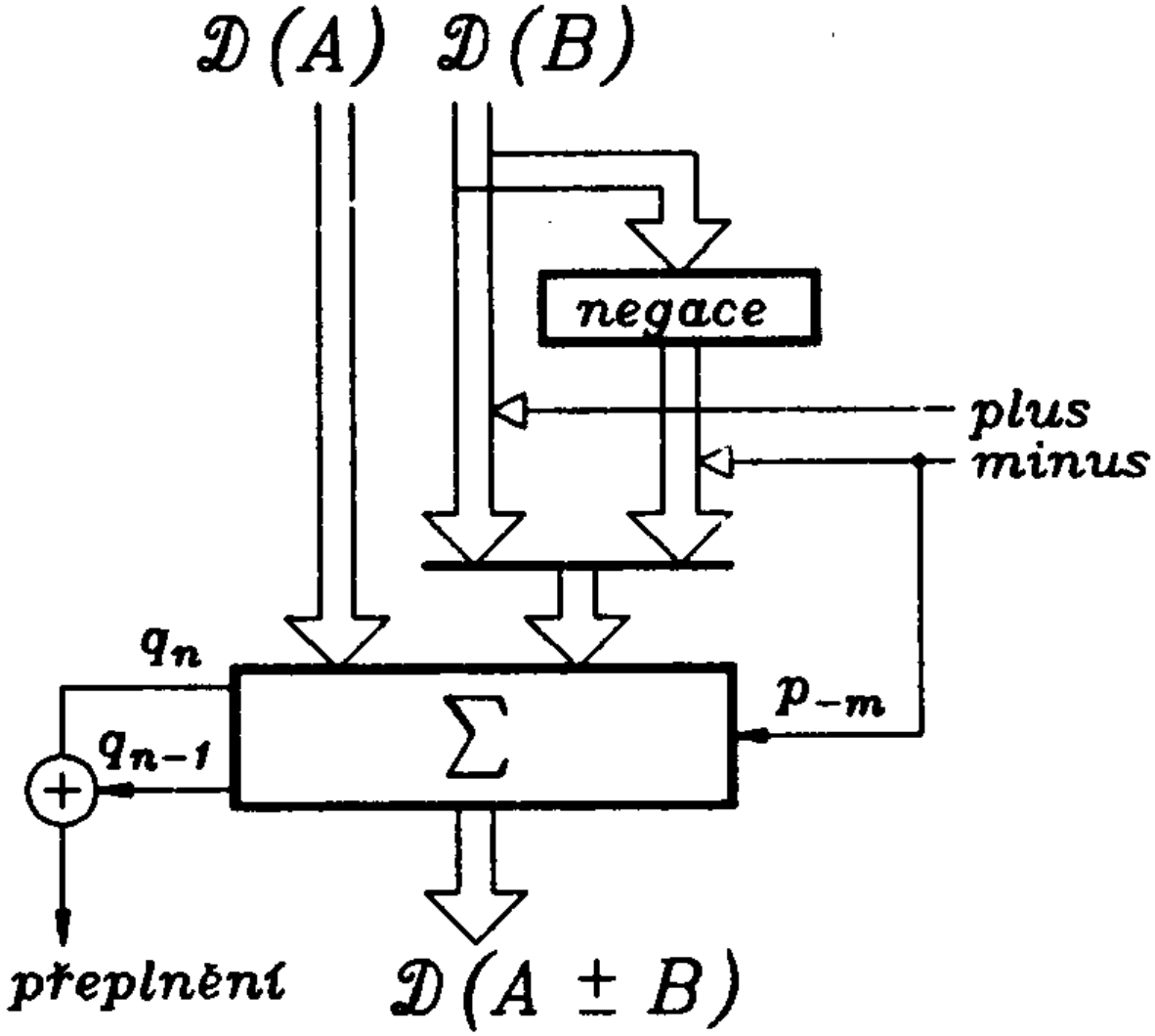
s_i	a	b	p	q	S
0	0	0	0	0	0
1	0	0	**1**	**0**	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	**0**	**1**	0
7	1	1	1	1	1


a_3^D b_3^D
a_2^D b_2^D
a_1^D b_1^D
a_0^D b_0^D
q_n
p_n
Σ
0
over
s_3^D
s_2^D
s_1^D
s_0^D

2


a_3^D b_3^D
a_2^D b_2^D
a_1^D b_1^D
a_0^D b_0^D
q_n
p_n
Σ
0
over
s_3^D
s_2^D
s_1^D
s_0^D

Sčítačka-odčítačka v doplňkovém kódu (včetně detekce přeplnění)



Násobení ve dvojkové soustavě

$$A = \sum_{i=0}^{m} a_i . z^j$$

$$B = \sum_{j=0}^{n} b_j . z^j$$

$$C = AB = \sum_{j=0}^{m+n+1} c_j . z^j$$

$$C = \sum_{j=0}^{n} A b_j . z^j$$

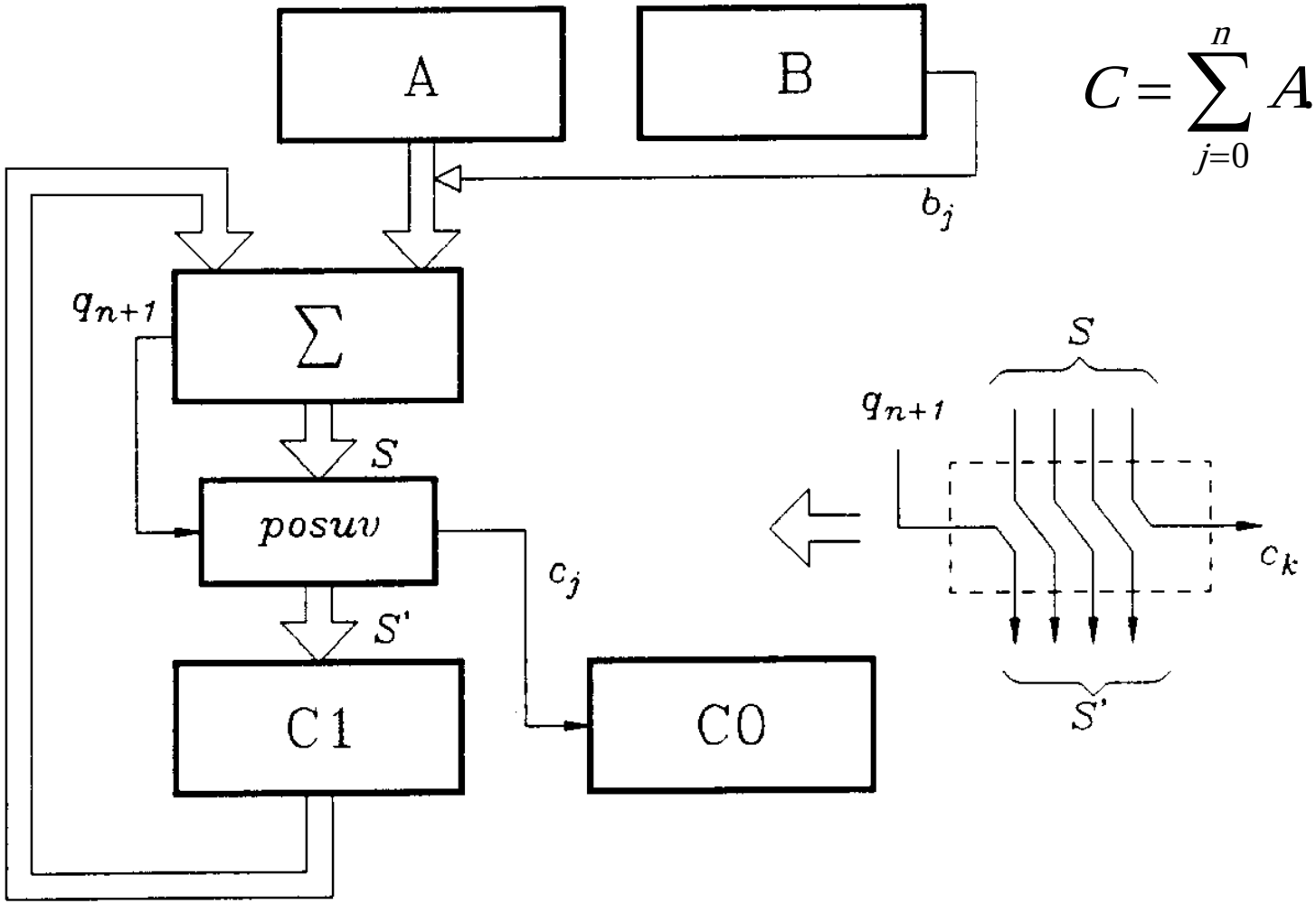
Postupné součty a posuvy

např.

$$\begin{array}{r} (111)_2 \\ * (101)_2 \\ \hline (100011)_2 \end{array} \qquad \begin{array}{r} (7)_{10} \\ * (5)_{10} \\ \hline (35)_{10} \end{array}$$

operace	C1	CO
přičtení 111	0111	000
posuv	0011	100
přičtení 000	0011	100
posuv	0001	110
přičtení 111	1000	110
posun	0100	011

Násobení ve dvojkové soustavě



$$C = \sum_{j=0}^{n} Ab_j.z^j$$

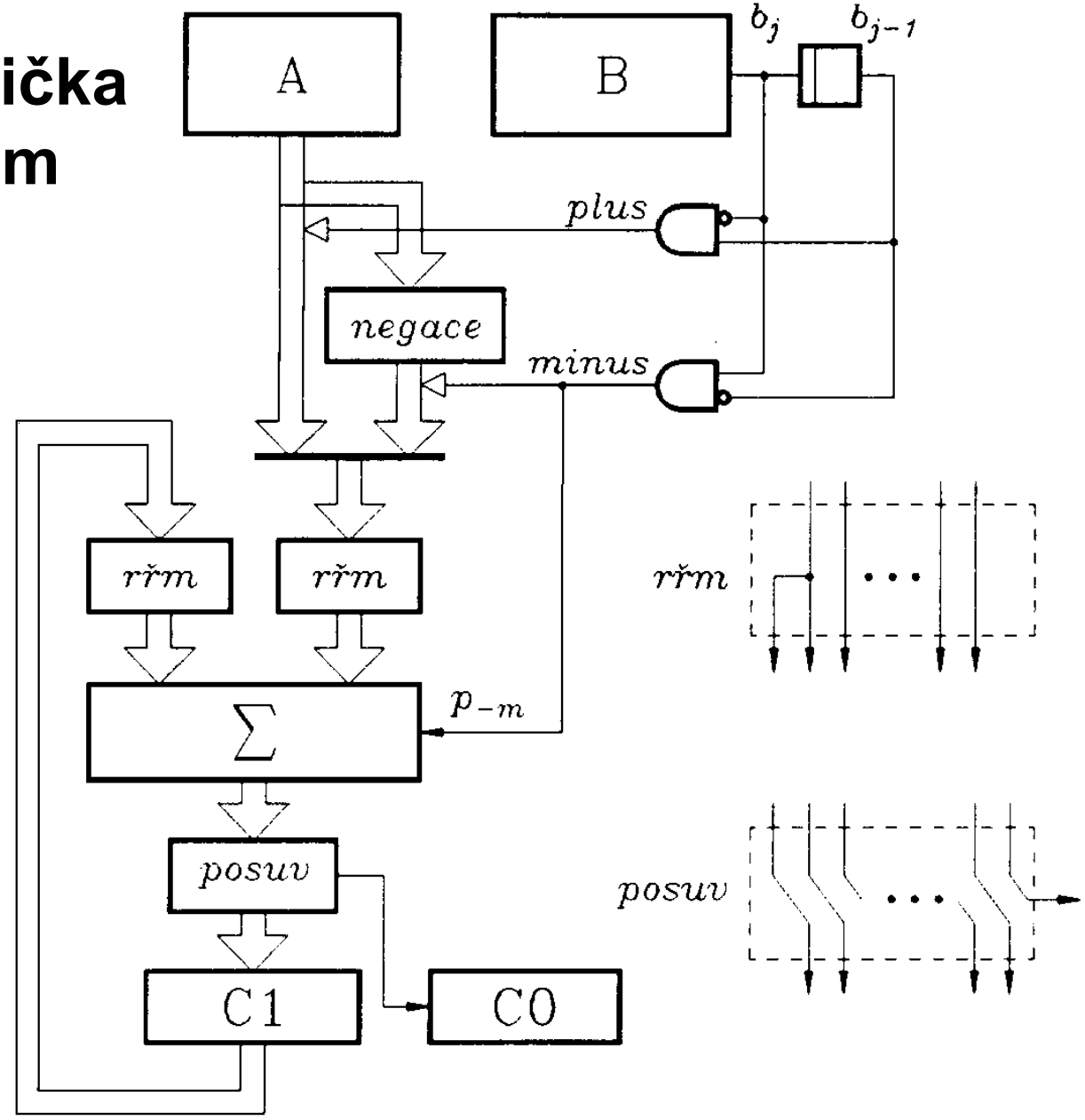
Násobení záporných čísel

Lze násobit absolutní hodnoty a vyhodnotit znaménko.

Násobení přímo v doplňkovém kódu: např. (-3).(±5) = ±15

1101.1011	1101.0101
00000	00000
11101 +	11101 +
111101	111101
11101 +	00000 +
111011	111110
00000 +	11101 +
111101	111100
00010 -	11111 -
1	1
00001111 ⇒ +15	11110001 ⇒ -15

Sériová násobička v doplňkovém kódu



Paralelní násobičky

Násobení se obvykle provádí během jediného taktu

$$C = \sum_{i=0}^{m} \sum_{j=0}^{n} a_i.b_j.2^{i+j}$$

$a_i.b_j$ lze realizovat ve dvojkové soustavě jako log. součin

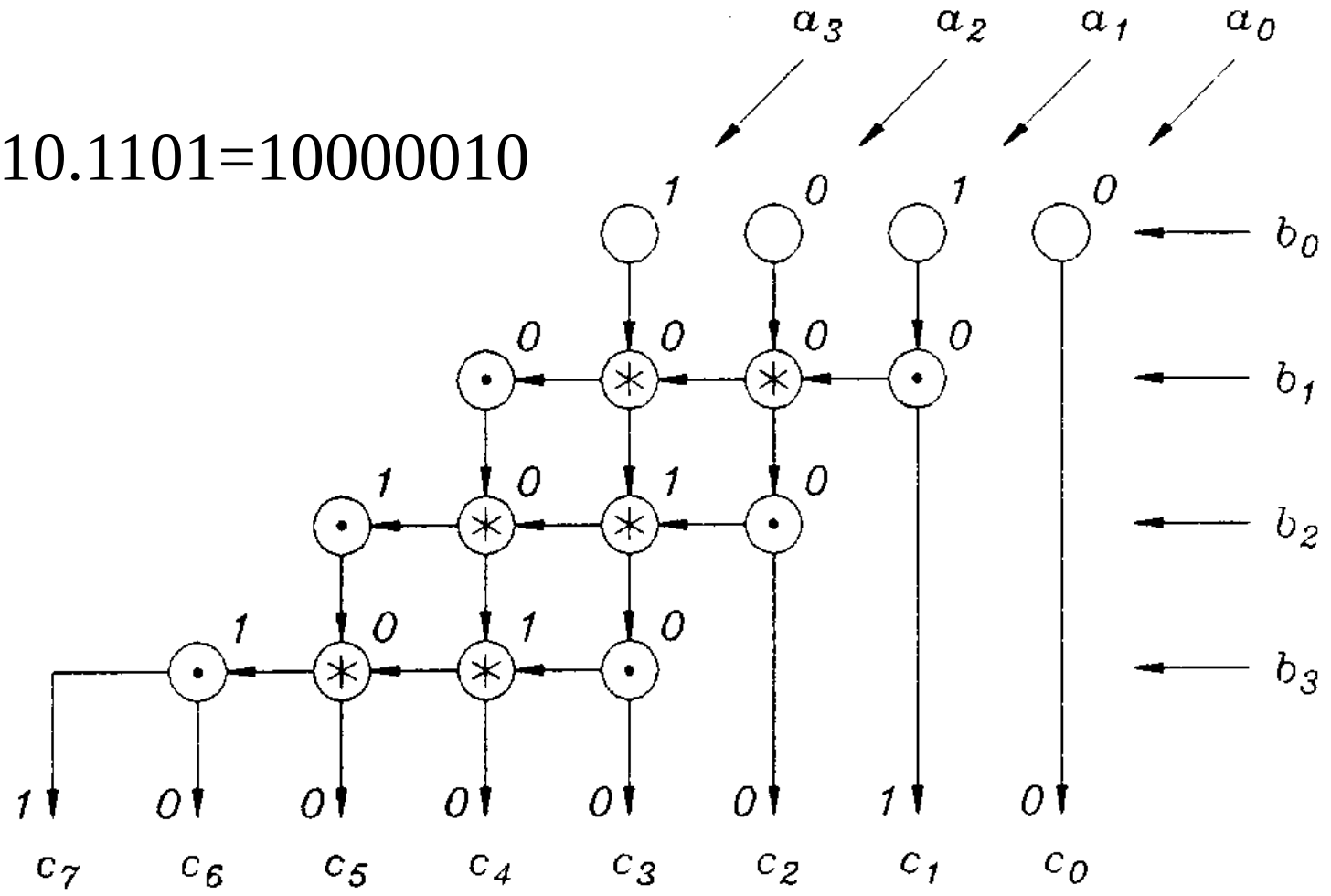
2^{i+j} znamená posuv o i+j míst

Vzniká pravidelná struktura se součinovými hradly (O)

- kroužky označené hvězdičkou, resp. tečkou reprezentují navíc úplnou, resp. poloviční sčítačku

Paralelní násobička

1010.1101=10000010



dělení s návratem přes nulu

```
0 , 1 0 1   :  0 , 1 1 0
            ↓
  1 0 1     :  1 1 0   =  0 , 1 1 0        podíl
−   1 1 0
    − 1                  →   0 ,
+   1 1 0                                  návrat
    1 0 1   0
−     1 1 0
      1 0 0  0           →       1
−       1 1 0
          1 0  0         →         1
−         1 1 0
          − 1 0          →           0
+         1 1 0                        návrat
          1 0 0                        zbytek
```

návrat — pomocná operace obnovující původní dílčí zbytek pro následující odčítání

dělení bez návratu přes nulu

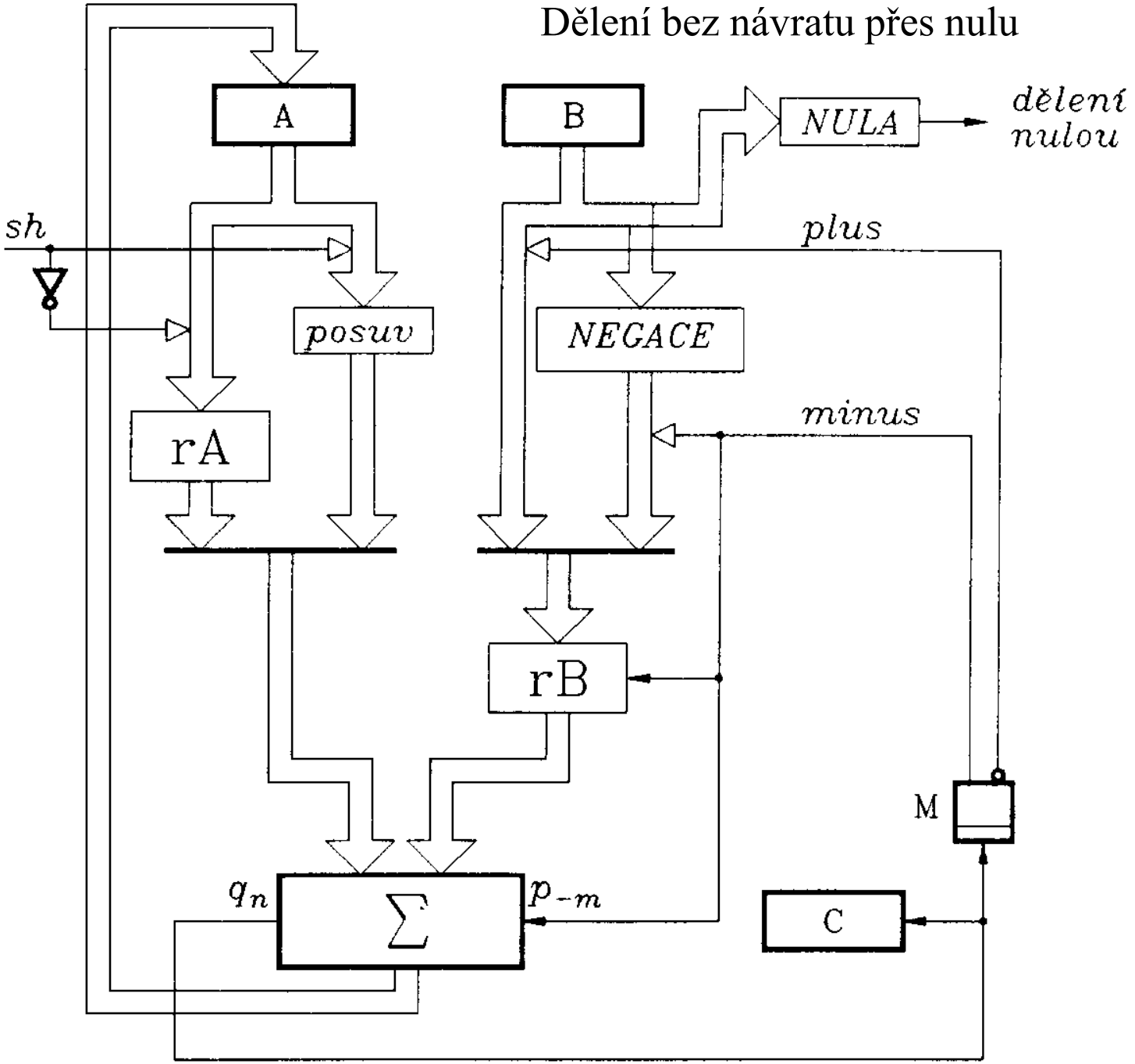
návrat = přičtení jistého čísla $X \rightarrow + \quad X$

následuje odečtení čísla $X \triangleright 1 \rightarrow \dfrac{-X/2}{+X/2}$

	101 : **110** = **0,110**				**podíl**
−	**110**				
	$-$ **1** 0	→	**0,**		
+	**110**				
	100 0	→		**1**	
−	**110**				
	10 0	→		**1**	
−	**110**				
	$-$ **10**	→		**0**	
+	**110**				**návrat**
	100				**zbytek**

návrat — pouze *v posledním kroku*, **je-li zbytek záporný, a to** *jenom* **v případě, že** *třeba získat* **také** *správný zbytek*.

Dělení bez návratu přes nulu



Zvýšení rychlosti dělení

Dělení převádíme na násobení převrácenou hodnotou a k násobení použijeme rychlé paralelní násobičky.

$$C = \frac{A}{B} = A.B^{-1}$$

Pro určení hodnoty B^{-1} lze použít iterační vzorec:

$$x_{i+1} = x_i(2 - B.x_i)$$

pak $\lim_{i\to\infty} x_i = B^{-1}$

Počet iterací závisí na volbě x_0 (každým iteračním krokem získáme dvojnásobný počet platných bitů). Vhodné počáteční hodnoty iteračního algoritmu mohou být tabelovány. Adresa paměťového místa v tabulce může představovat několik prvních bitů čísla *B* (např. je-li adresa 5-bitová, budou pro 32-bitová čísla B a stačit 3 iterace).

Elementární funkce

Postupy vhodné pro řešení programy jsou vzhledem ke své složitosti často nevhodné (Mc. Laurinův rozvoj, Čebyševovy polynomy) – problematické je zejména násobení a dělení, snadný je výběr bitu nebo skupiny bitů.

Výpočet probíhá často v pohyblivé řádové čárce. To se dá využít pro omezení rozsahu.

Např . Ln A = lnM + E.ln2, kde 0,5<M<1

Určit lnM v uvedeném rozsahu je snazší

Druhá odmocnina

Možno nalézt jako numerické řešení rovnice

$$x^2 - A = 0$$

Lépe využít vztahu, že odmocnina je větší než maximální počet nejmenších přirozených lichých čísel, která lze od A odečíst

Postupné odečítání po sobě jdoucích lichých čísel dokud nedostaneme nulu nebo záporný zbytek

$\sqrt{25} = 5$ $\qquad 25 - {}_1 1 - {}_2 3 - {}_3 5 - {}_4 7 - {}_5 9 = 0$

Děkuji za pozornost

Tento materiál vznikl v rámci projektu ESF CZ.1.07/2.2.00/28.0050
Modernizace didaktických metod a inovace výuky technických předmětů,
který je spolufinancován Evropským sociálním fondem a státním rozpočtem ČR.