

Architektura počítačů

Logické obvody

http://d3s.mff.cuni.cz/teaching/computer_architecture/



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Lubomír Bulej

bulej@d3s.mff.cuni.cz

Digitální počítač



● Dvě úrovně napětí

■ Vyšší úroveň

- Logická 1, *high, true, asserted*

■ Nižší úroveň

- Logická 0, *low, false, deasserted*

- Hodnoty jsou vzájemně inverzní a doplňkové

● Kombinační obvody

- Výstup závisí pouze na vstupu (neobsahují paměť)
- Reprezentace logických funkcí

● Sekvenční obvody

- Výstup závisí na vstupu a vnitřním stavu (paměť)
- Umožňují zachytit posloupnost kroků výpočtu



Logické funkce a pravdivostní tabulky

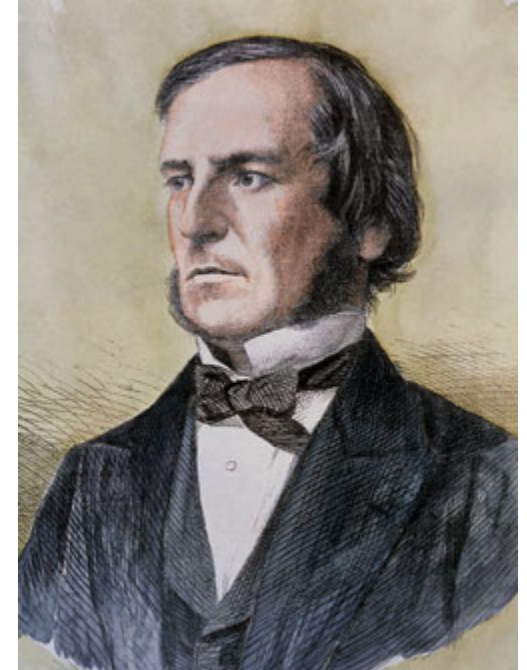
- **Logická funkce (též Booleovská funkce)**
 - Výstupní hodnota je funkcí vstupních hodnot
 - $f: \mathbf{B}^k \rightarrow \mathbf{B}$, kde $\mathbf{B} = \{0, 1\}$ a $k \in \mathbf{N}$ je arita
 - **Pravdivostní tabulka**
 - Definice logické funkce výčtem hodnot vstupů a výstupů (pro k vstupů má tabulka 2^k řádků)

Vstupy		Výstup
a	b	$f(a, b)$
0	0	0
0	1	1
1	0	1
1	1	1



Booleova algebra (1)

- **Geogle Boole (1815 – 1864)**
 - Matematik, filozof, logik
 - Systematizace aristotelské logiky
 - Symbolická logika (algebraická reprezentace logických výroků)
 - V mnoha ohledech předběhl dobu
 - Před axiomatizací teorie množin atd.
 - Retrospektivně považován za jednoho ze zakladatelů informatiky



[1]



Booleova algebra (2)

- **Algebraická reprezentace logických funkcí**
 - Logické proměnné s oborem hodnot $\mathbf{B} = \{0, 1\}$
 - Základní logické operátory – primitivní logické funkce
 - **Negace (NOT):** \bar{x} , $\neg x$, $!x$
 - **Logický součin, konjunkce (AND):** $x \cdot y$, $x \wedge y$, $x \&\& y$
 - **Logický součet, disjunkce (OR):** $x + y$, $x \vee y$, $x || y$
 - Další logické operátory (16 pro 2 proměnné)
 - NAND, NOR, XOR atd.



Booleova algebra (3)



Vstupy		Základní operátory			Univerzální operátory		Další operátory						
a	b	NOT a	a AND b	a OR b	a NAND b	a NOR b	a XOR b	a XNOR b					
		\neg	\wedge	\vee	\uparrow	\downarrow	\oplus	\leftrightarrow	\rightarrow	\leftarrow	\dots	\perp	\top
0	0	1	0	0	1	1	0	1	1	1	...	0	1
0	1	1	0	1	1	0	1	0	1	0	...	0	1
1	0	0	0	1	1	0	1	0	0	1	...	0	1
1	1	0	1	1	0	0	0	1	1	1	...	0	1



Booleova algebra (4)

● Axiomy a odvozené vlastnosti

- **Idempotence:** $a + a = a, a \cdot a = a$
- **Komutativita:** $a + b = b + a, a \cdot b = b \cdot a$
- **Asociativita:** $a + (b + c) = (a + b) + c, a \cdot (b \cdot c) = (a \cdot b) \cdot c$
- **Absorpce:** $a \cdot (a + b) = a, a + (a \cdot b) = a$
- **Distributivita:** $a \cdot (b + c) = (a \cdot b) + (a \cdot c), a + (b \cdot c) = (a + b) \cdot (a + c)$
- **Neutralita 0 a 1:** $a + 0 = a, a \cdot 1 = a$
- **Agresivita 0 a 1:** $a + 1 = 1, a \cdot 0 = 0$
- **Komplementarita:** $a + \neg a = 1, a \cdot \neg a = 0$
- **Absorpce negace:** $a \cdot (\neg a + b) = a \cdot b, a + (\neg a \cdot b) = a + b$
- **De Morganovy zákony:** $\neg(a + b) = \neg a \cdot \neg b, \neg(a \cdot b) = \neg a + \neg b$
- **Dvojitá negace:** $\neg(\neg a) = a$



Logické operace (1)

- **Rozšíření logických funkcí na operace s (konečnými) posloupnostmi bitů**
 - **Slovo** = konečná posloupnost bitů
 - **Délka slova** = počet bitů konečné posloupnosti
 - Výstupní hodnota logické operace je funkce vstupních hodnot
 - $f: (\mathbf{B}^n)^k \rightarrow \mathbf{B}^n$, kde $\mathbf{B} = \{0, 1\}$, $k \in \mathbf{N}$ je arita a $n \in \mathbf{N}$ je délka slova



Logické operace (2)

- **Typické logické operace**

- Logický součin, logický součet, logická negace atd. po bitech slova (*bitwise*)
 - Operátory **&**, **|**, **~** atd. v jazycích C, Java
 - Odpovídající logická funkce aplikována na jednotlivé bity vstupního slova, výsledky po jednotlivých bitech uloženy do výstupního slova
 - Umožňují izolovat (AND), vynulovat (AND, NOR), nastavit (OR), invertovat (XOR) vybrané bity slova, resp. invertovat bity celého slova (NOT)



Logické operace (3)

- **Typické logické operace**

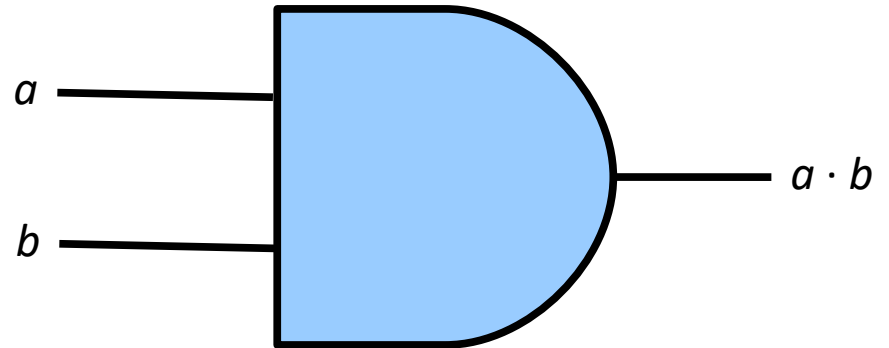
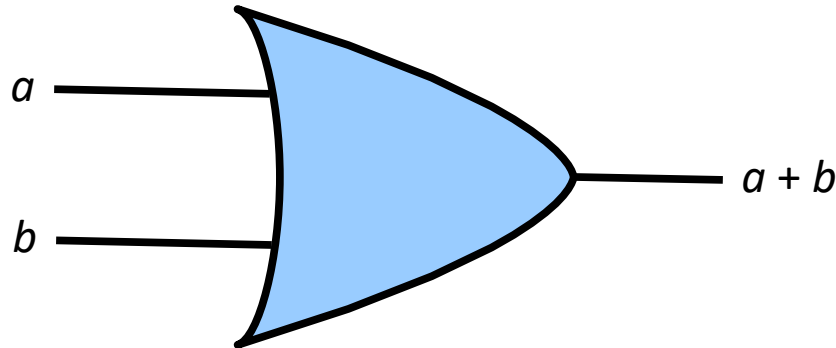
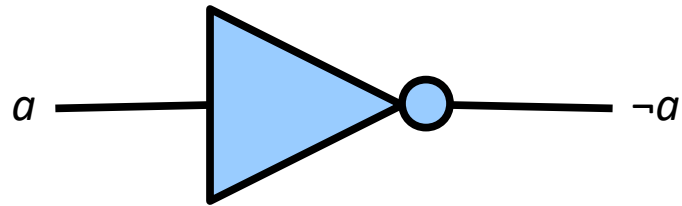
- Logické posuny vlevo a vpravo

- Operátory \ll a \gg v jazycích C, Java
- Přesun bitů ve slově o i pozic vlevo nebo vpravo
 - „Uvolněné“ bity jsou nahrazeny hodnotou 0
- Pro reprezentaci přirozených čísel jako posloupnost bitů
 - Posun o i bitů vlevo odpovídá násobení číslem 2^i
 - Posun o i bitů vpravo odpovídá dělení číslem 2^i



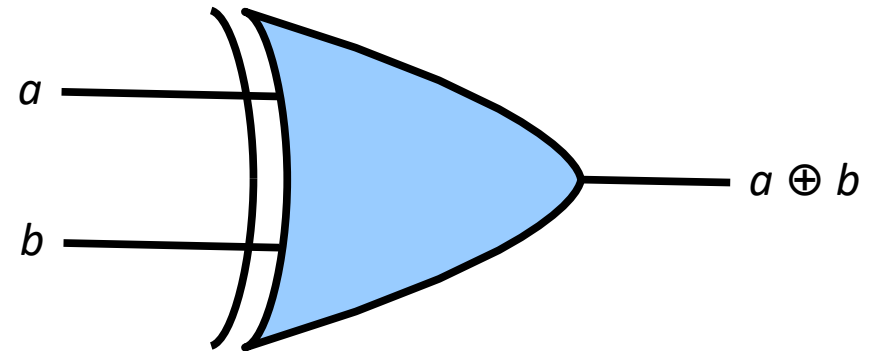
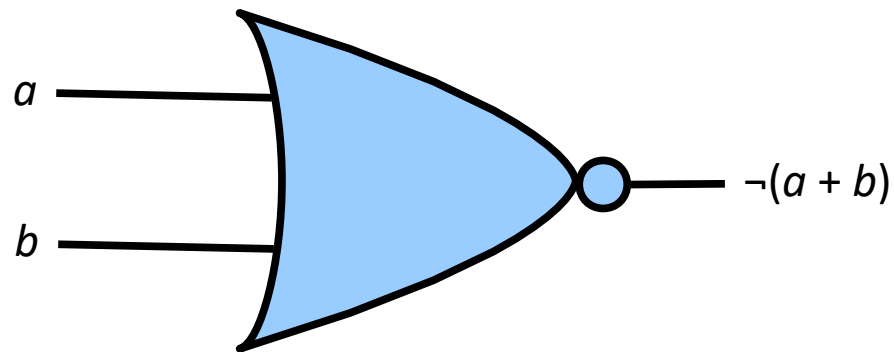
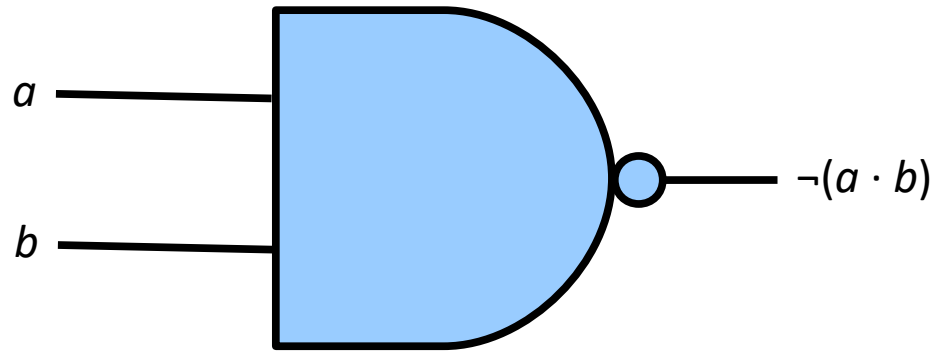
Logická hradla (1)

- Fyzická realizace logických operátorů
 - Základní NOT, OR, AND



Logická hradla (2)

- Fyzická realizace logických operátorů
 - Odvozené NAND, NOR, XOR



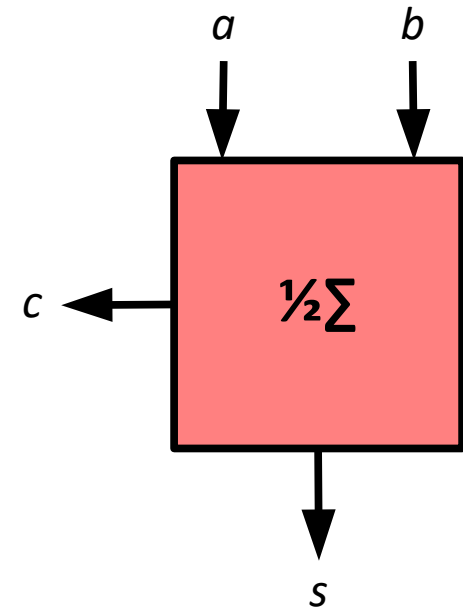
- **Fyzická realizace složitějších logických funkcí a logických operací**
 - Spojení více logických operátorů
 - Logické signály jako proměnné
 - Logická hradla jako operátory
 - V praxi nejčastější kombinace hradel NAND a NOR
- **Funkční blok**
 - Seskupení logických obvodů do větších celků
 - Abstrakce vnitřní struktury



Logický obvod sčítání

● Součet dvou 1-bitových čísel

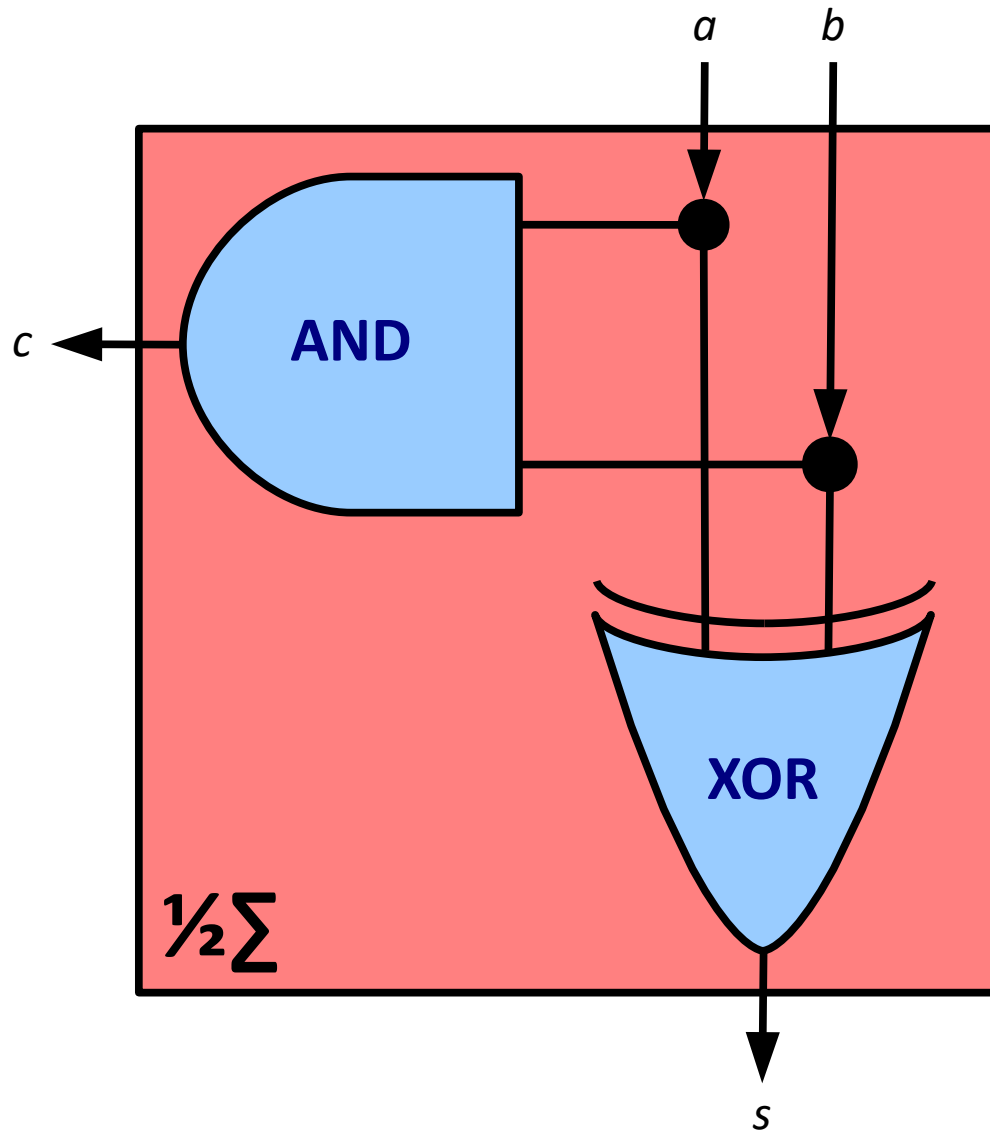
- Nejjednodušší případ
- **Vstup:** operand a , operand b
- **Výstup:** součet s , přenos c
 - $s = a \cdot \neg b + \neg a \cdot b = a \text{ XOR } b$
 - $c = a \cdot b = a \text{ AND } b$



Vstupy		Výstupy	
a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



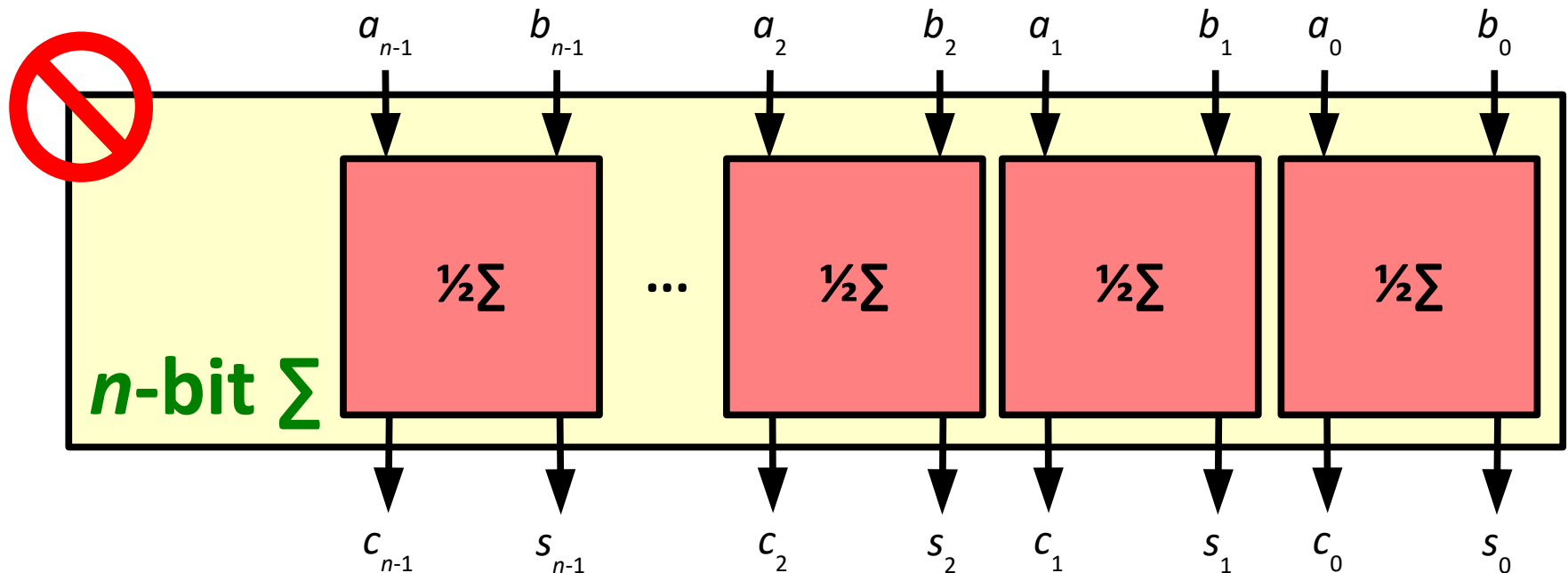
Logický obvod sčítání (2)



Logický obvod sčítání (3)

- Sčítání n -bitových čísel

- Spojení n $\frac{1}{2}$ -sčítaček pro jednotlivé bity?



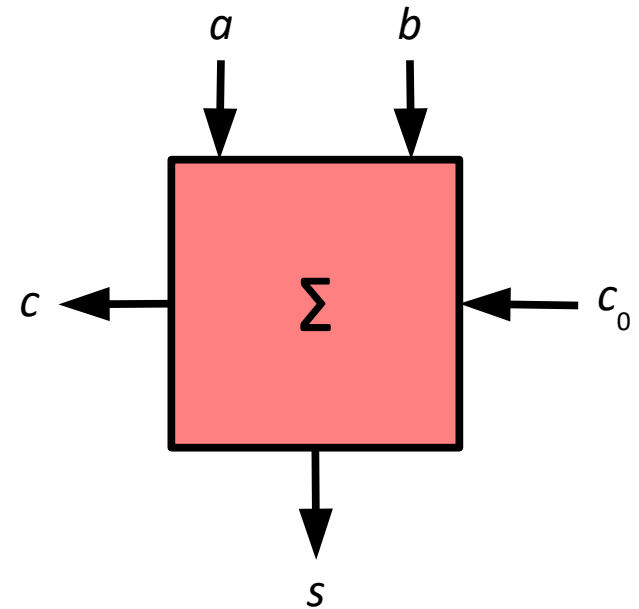
- $\frac{1}{2}$ -sčítačka neumí propagovat přenosy mezi řády (málo vstupů)



Logický obvod sčítání (4)

• Úplná sčítačka

- Součet dvou 1-bitových čísel s přihlédnutím k přenosu z nižšího řádu
- **Vstup:** operand a , operand b , přenos c_0
- **Výstup:** součet s , přenos c



Logický obvod sčítání (5)

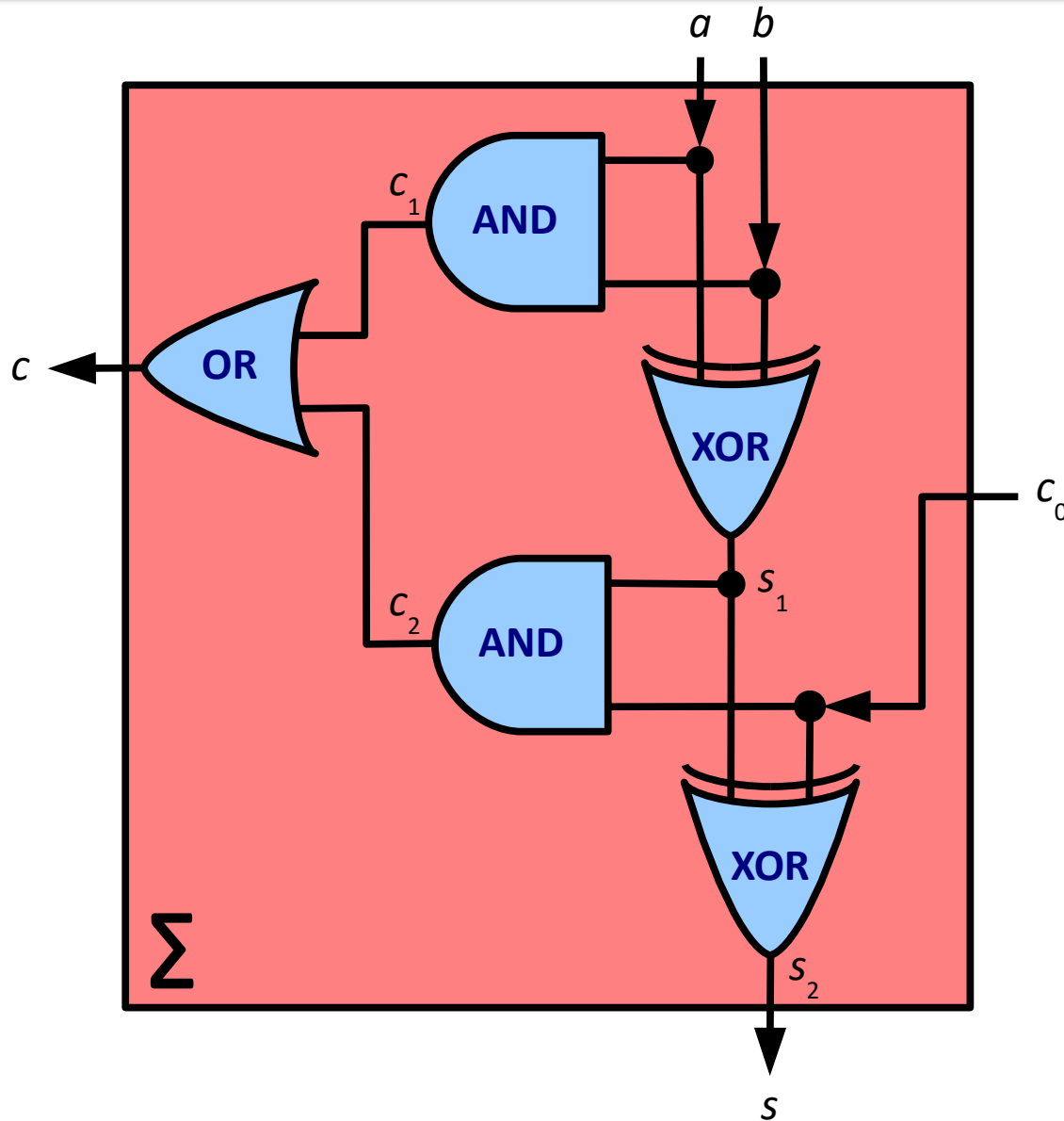
● Úplná sčítačka

- Součet dvou 1-bitových čísel s přihlédnutím k přenosu z nižšího řádu
- **Vstup:** operand a , operand b , přenos c_0
- **Výstup:** součet s , přenos c
 - $s = \neg c_0 \cdot (a \cdot \neg b + \neg a \cdot b) + c_0 \cdot (a \cdot b + \neg a \cdot \neg b)$
 $s = \dots$
 $s = c_0 \text{ XOR } (a \text{ XOR } b)$
 - $c = a \cdot b + c_0 \cdot (a \cdot \neg b + \neg a \cdot b)$
 $c = (a \text{ AND } b) \text{ OR } (c_0 \text{ AND } (a \text{ XOR } b))$

c_0	a	b	c	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Logický obvod sčítání (6)

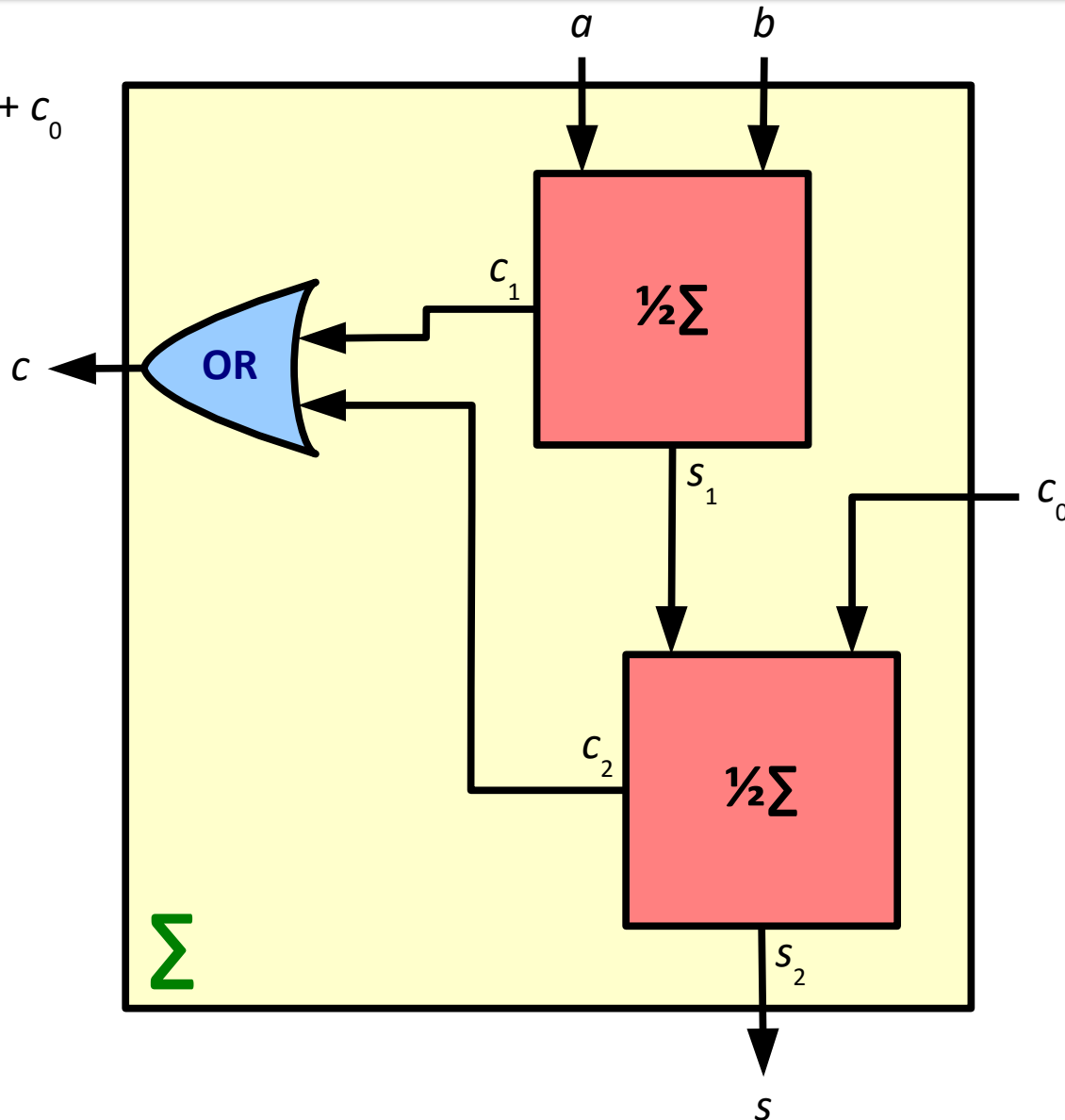


Logický obvod sčítání (7)

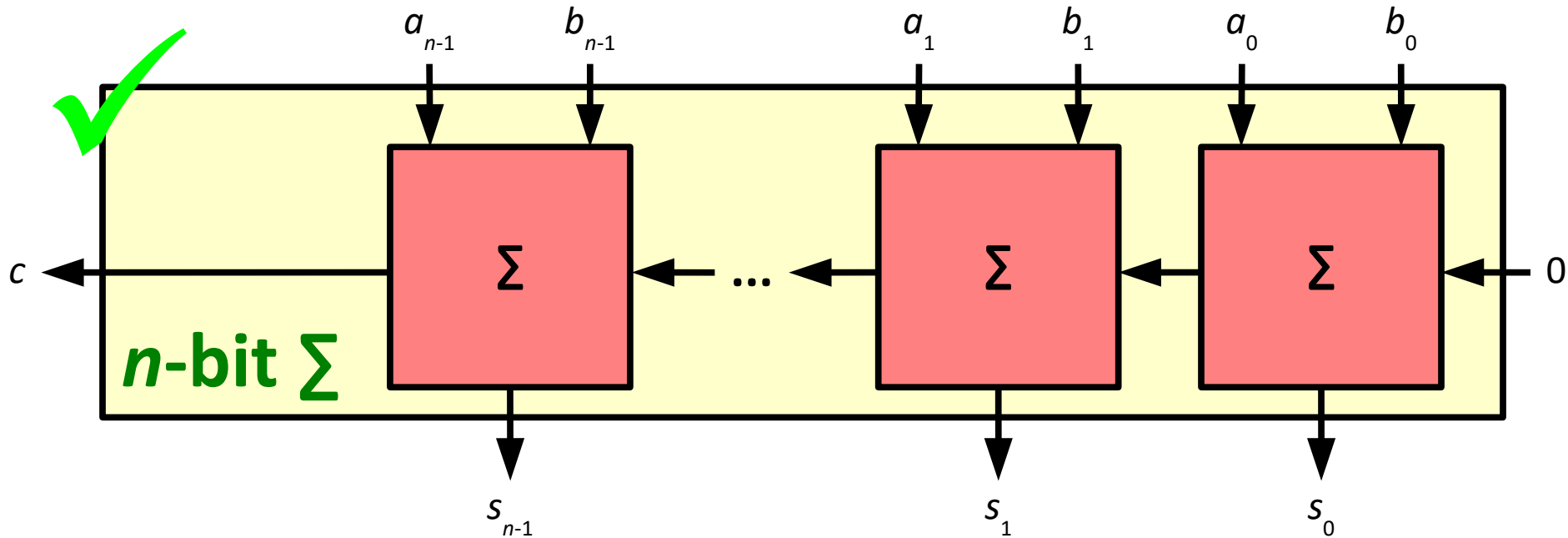


Abstrakce:

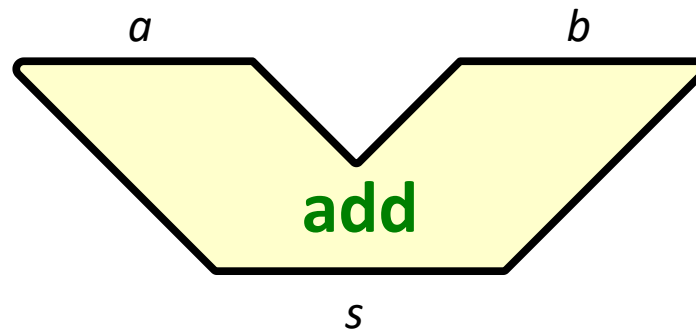
$$a + b + c_0 = (a + b) + c_0$$



Logický obvod sčítání (8)



Funkční blok:

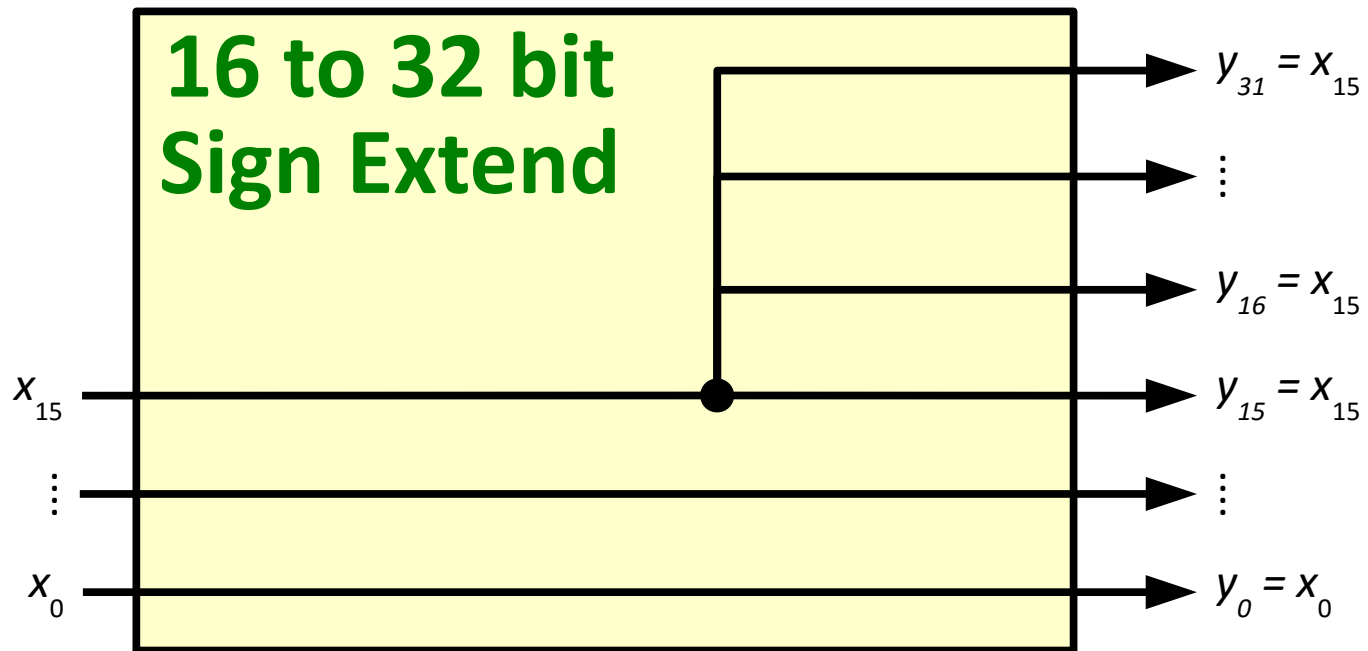


Logický obvod odčítání

- **Využití vlastností dvojkového doplňku**
 - Základní stavební blok: Sčítačka
 - Použití XOR hradla jako řízeného invertoru
 - **Příklad:** 2-bitová ALU s podporou sčítání a odčítání
 - **Vstup:** operand a_1a_0 , operand b_1b_0
 - **Řídící vstup:** určení typu operace SUB
 - $SUB = 0 \rightarrow$ sčítání
 - $SUB = 1 \rightarrow$ odčítání
 - **Výstup:** součet/rozdíl s_1s_0 , přenos c

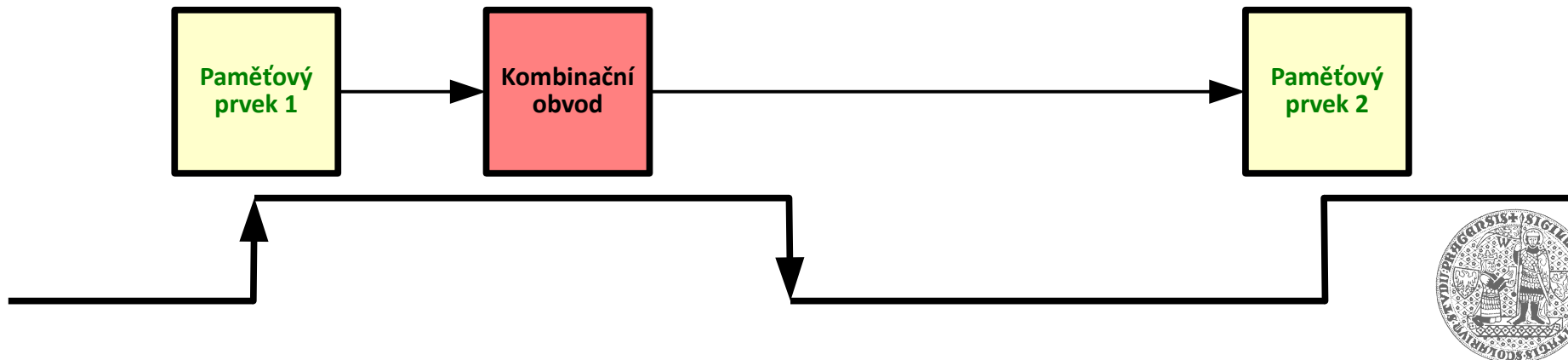


Logický obvod znaménkového rozšíření



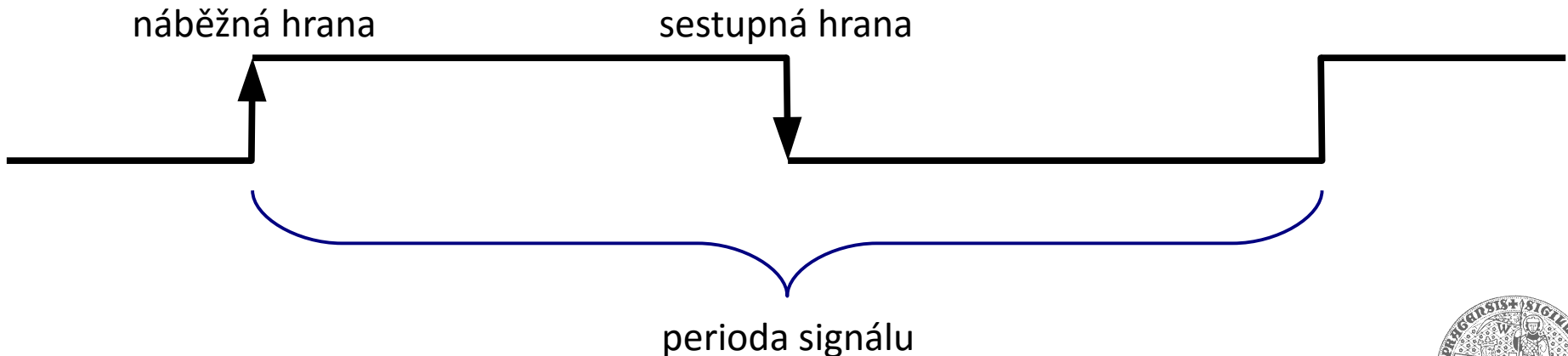
Sekvenční obvody

- **Kombinační obvody + paměťové prvky**
 - Paměťové prvky udržují vnitřní stav
 - Vstup a obsah paměťových prvků určuje výstup a následující stav (nový obsah paměťových prvků)
 - Synchronní vs. asynchronní sekvenční obvody
 - Způsob a okamžik změny stavu
 - Nutnost docílit stability vstupních hodnot (vstupní hodnoty se nemění)

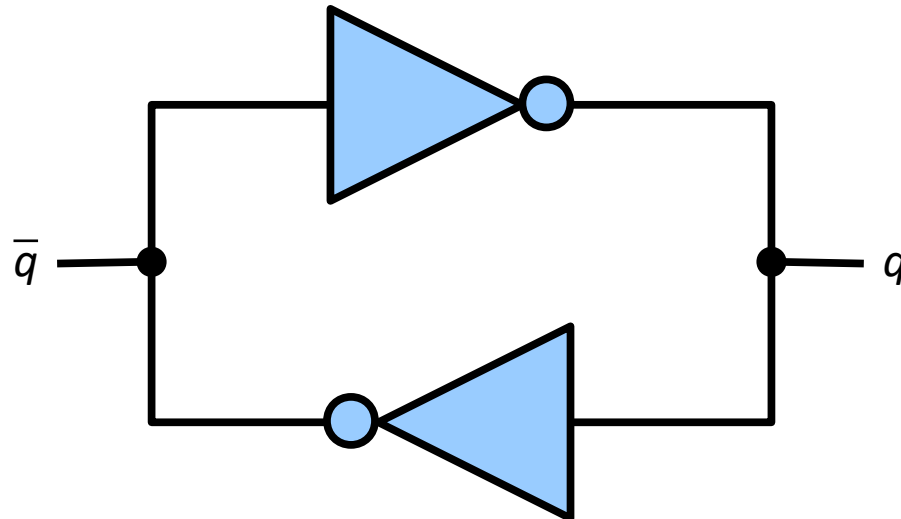


Synchronní sekvenční obvody

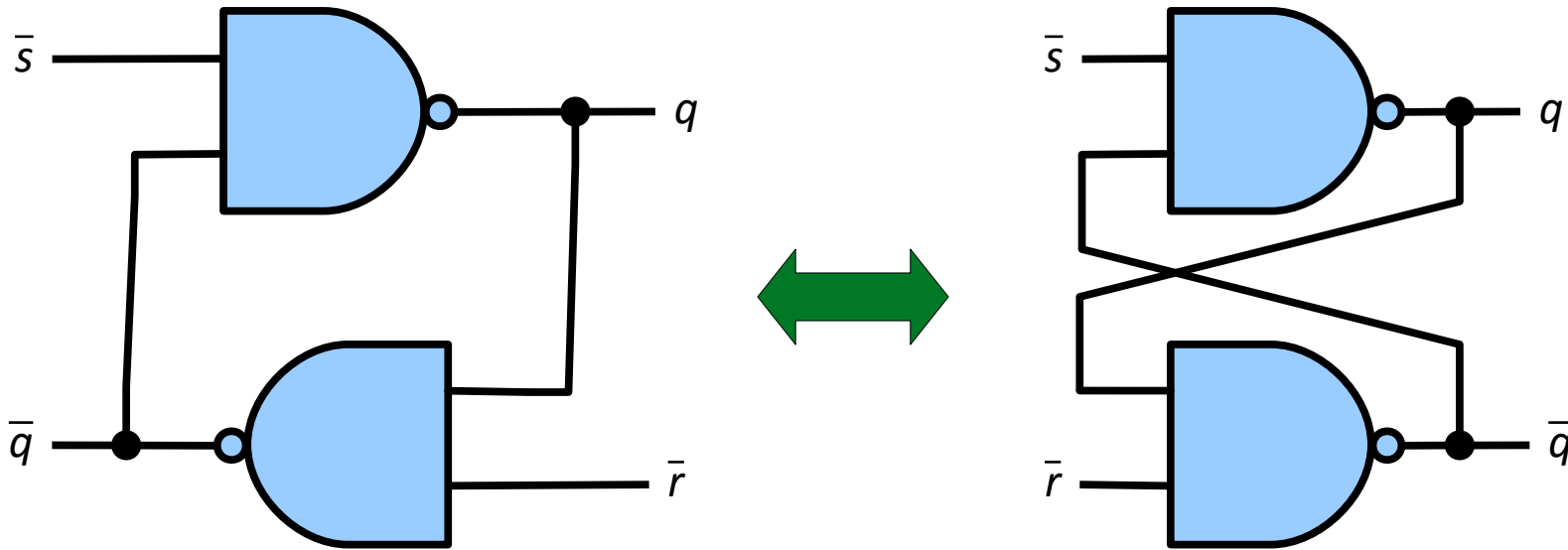
- **Hodinový signál synchronizuje změny stavu**
 - Změna stavu během jednoho cyklu hodin
 - Hodnoty vstupů kombinačních obvodů se během čtení nemění
 - Zápis výstupů do paměťových prvků probíhá s náběžnou/sestupnou hranou hodinového signálu



- **Dvojice invertorů se zpětnou vazbou**
 - Asynchronní obvod se dvěma stabilními stavy
 - Stav slouží pro uložení 1 bitu informace
 - Logický základ paměťových obvodů
 - Potřebujeme stav obvodu nějak řídit...



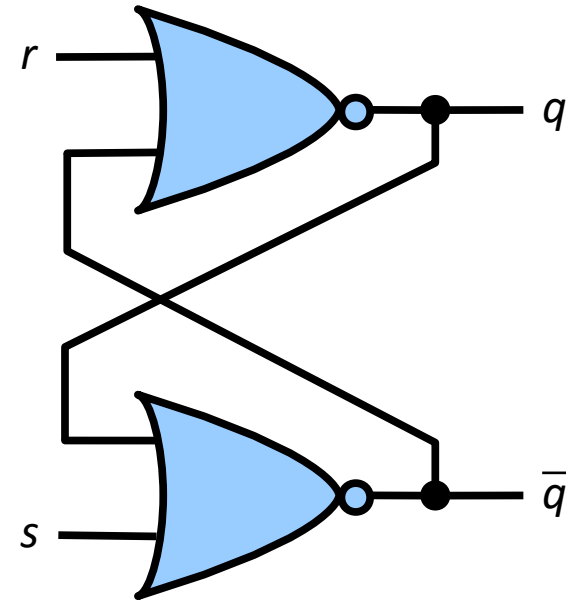
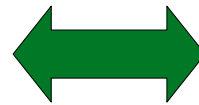
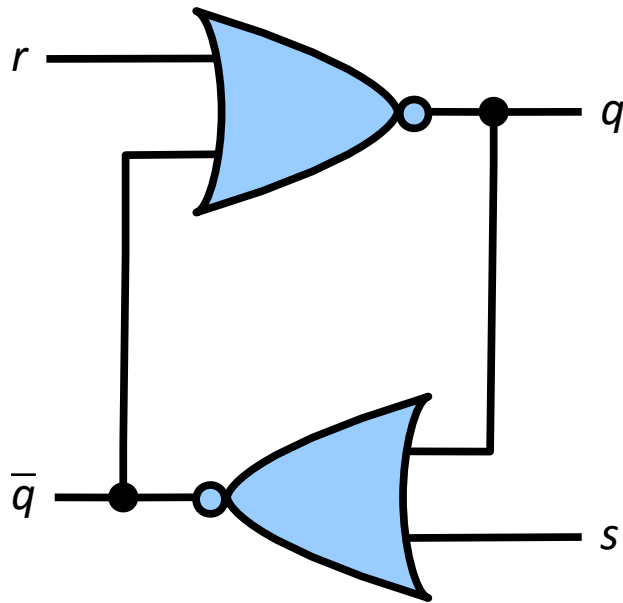
Klopný obvod Set-Reset (R-S, latch)



Vstupy		Výstupy	
\bar{r}	\bar{s}	q_n	\bar{q}_n
0	0	?	?
0	1	0	1
1	0	1	0
1	1	q_{n-1}	$\neg q_{n-1}$



Klopný obvod Set-Reset (R-S, latch)



Vstupy		Výstupy	
r	s	q_n	\bar{q}_n
0	0	q_{n-1}	\bar{q}_{n-1}
0	1	1	0
1	0	0	1
1	1	?	?



Další klopné obvody

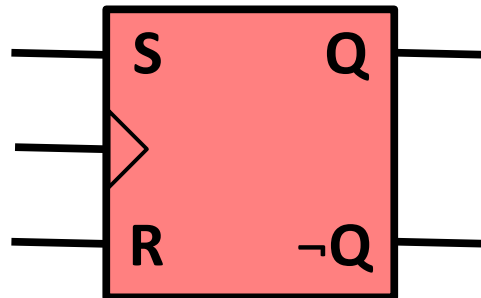
- **Odvozené od R-S**

- *Clocked R-S latch*

- Synchronní varianta R-S
- Mění stav, kdykoliv je nastaveno r , s a hodinový signál je logická jednička

- *R-S master/slave (R-S flip-flop)*

- Mění stav jen s náběžnou/sestupnou hranou hodinového signálu



Další klopné obvody (2)

- **Odvozené od R-S**

- *J-K master/slave (J-K flip-flop)*

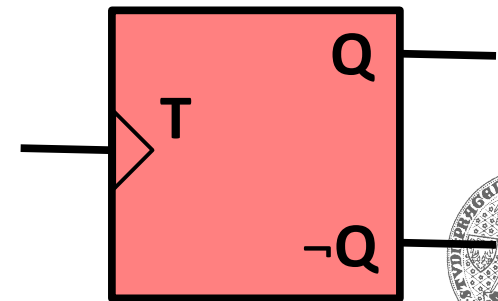
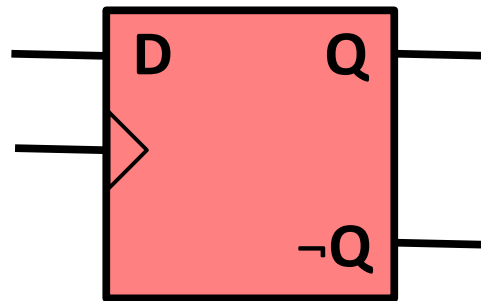
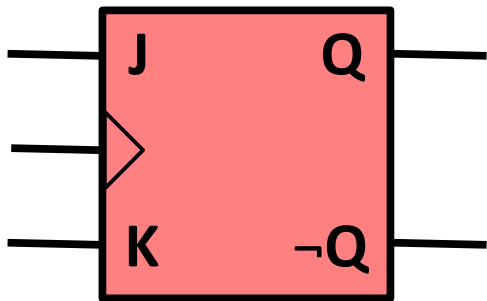
- Rozšíření R-S ($J = S$, $K = R$), pro $J = K = 1$ provede inverzi aktuálního stavu

- *Clocked D latch, D flip-flop*

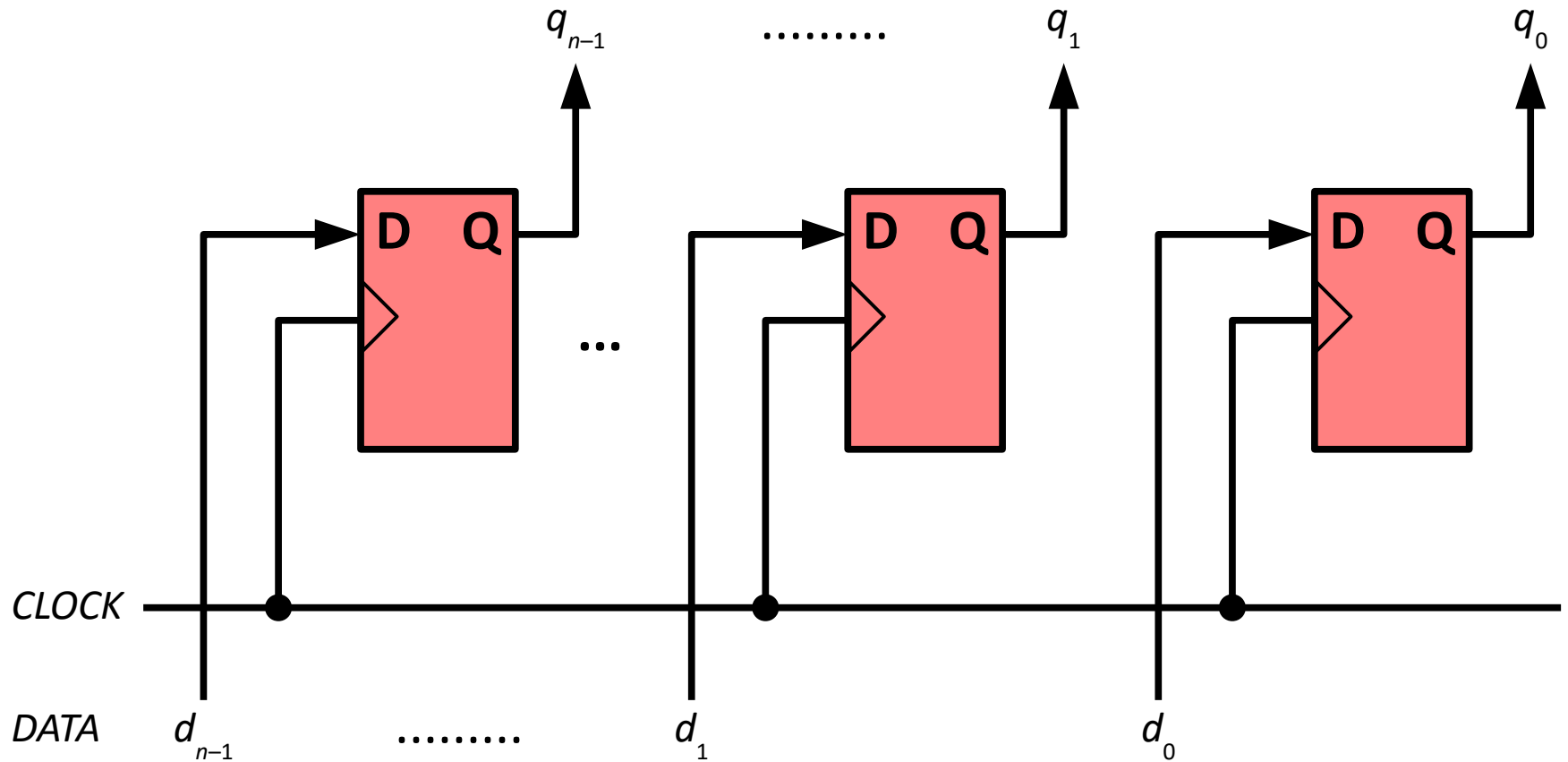
- Jen jedna vstupní hodnota

- *T flip-flop*

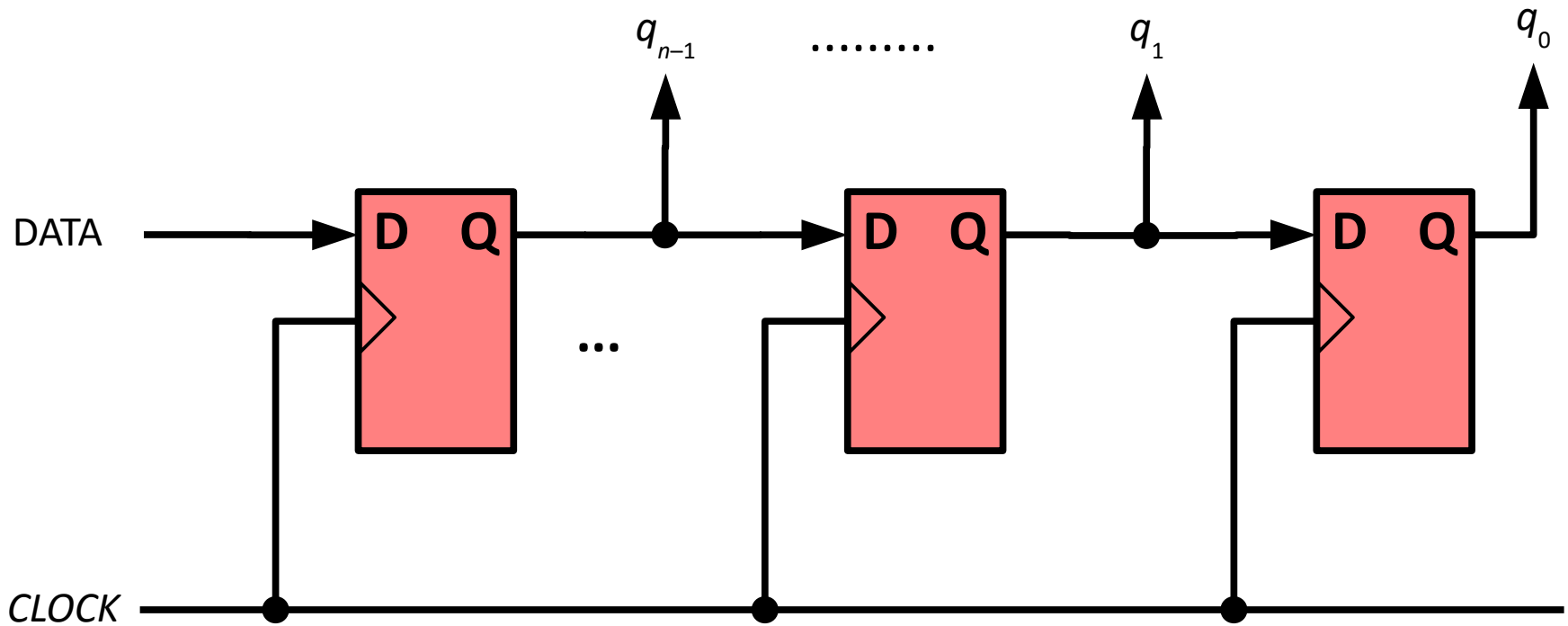
- Dělič vstupního hodinového signálu



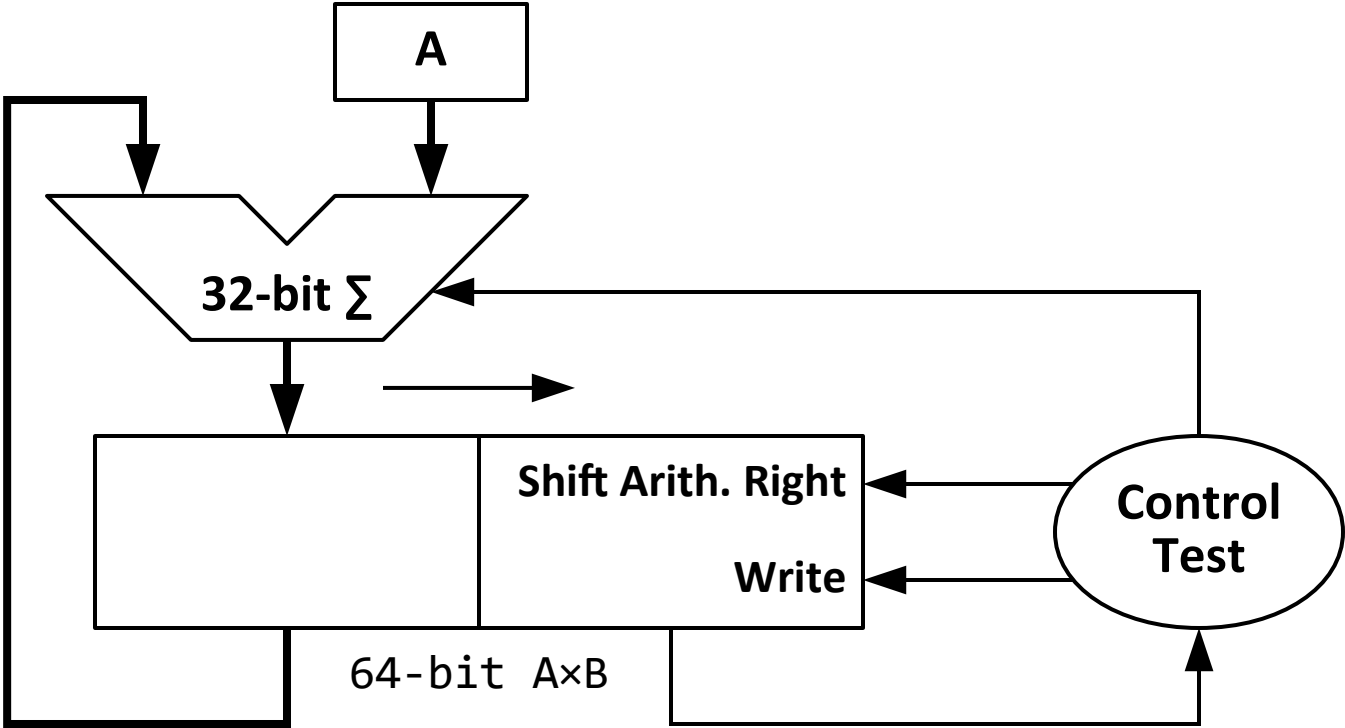
Registr z klopných obvodů



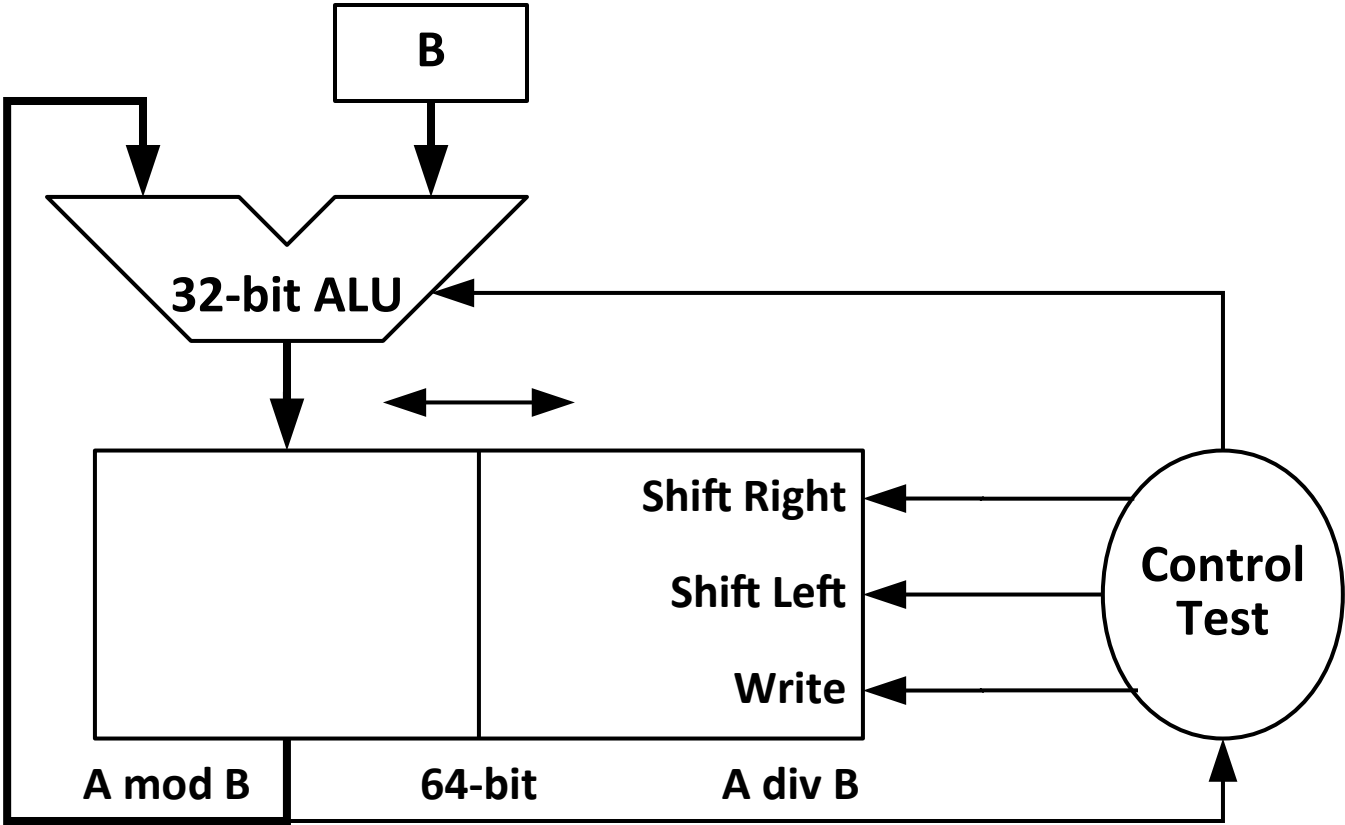
Posuvný registr z klopných obvodů



32-bitová sekvenční násobička



32-bitová sekvenční dělička



Reference

[1] http://upload.wikimedia.org/wikipedia/commons/c/ce/George_Boole_color.jpg

