

Operace ALU

INP 2025

FIT VUT v Brně



Princip ALU (FX)

Požadavky - funkční:

Logické operace

Sčítání (v doplňkovém kódu)

Posuvy/rotace

Násobení

Dělení

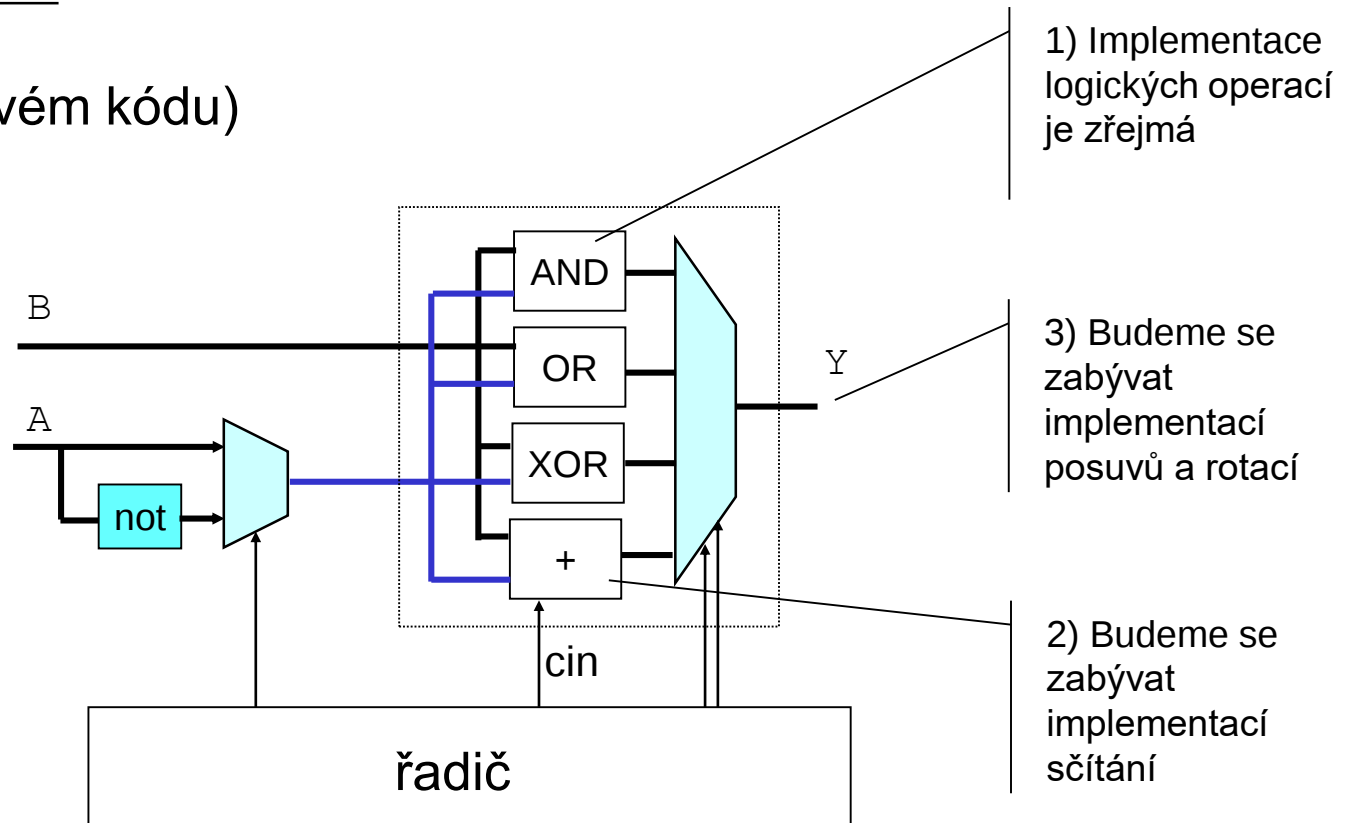
Další požadavky:

zpoždění

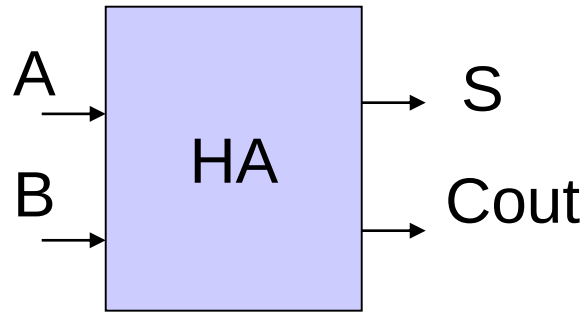
plocha

příkon

atd.



Poloviční sčítačka



S:

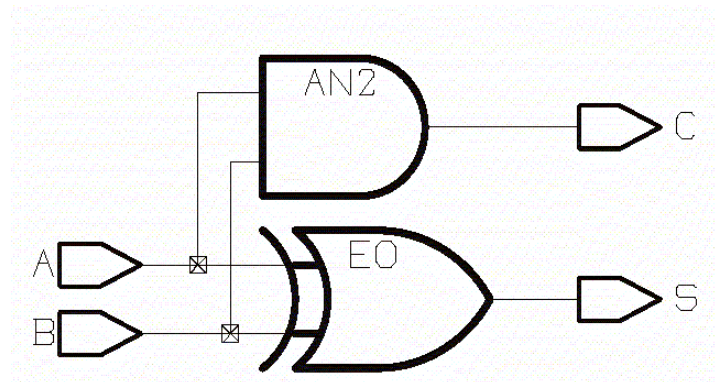
	<u>A</u>
0	1
1	0

| B

C:

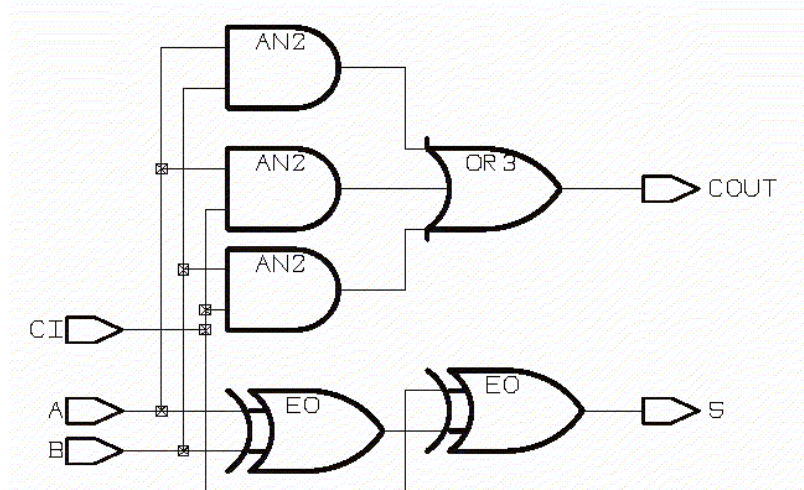
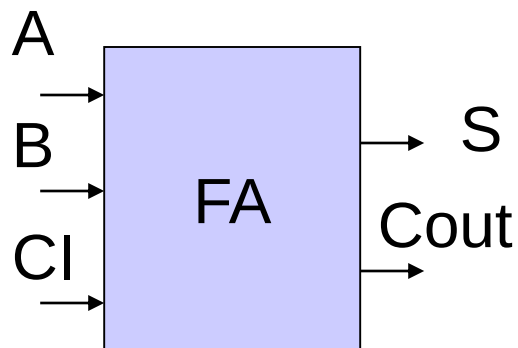
	<u>A</u>
0	0
0	1

| B



Úplná sčítačka

A	B	CI	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



Cout:

			<u>A</u>	
0	0	1	0	
0	1	1	1	B
			<u>CI</u>	

S:

		<u>A</u>		
0	1	0	1	
1	0	1	0	B
		<u>CI</u>		

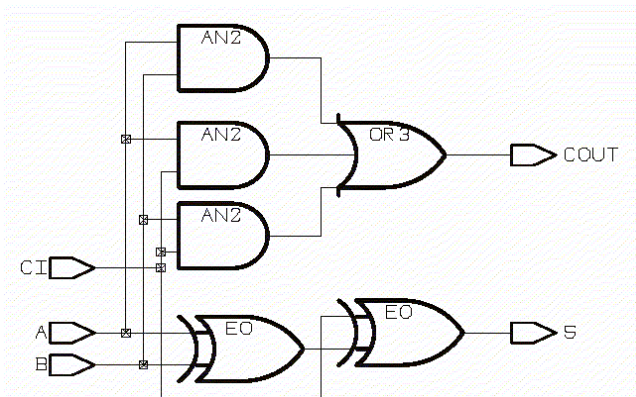
Úplná sčítačka – úrovně popisu

1. Úroveň chování

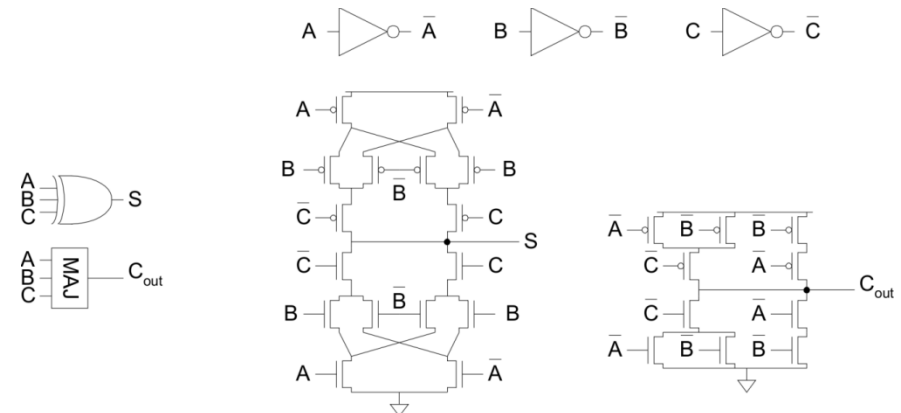
$S \leq A \text{ xor } B \text{ xor } CI;$

$CO_{out} \leq (A \text{ and } B) \text{ or } (A \text{ and } CI) \text{ or } (B \text{ and } CI);$

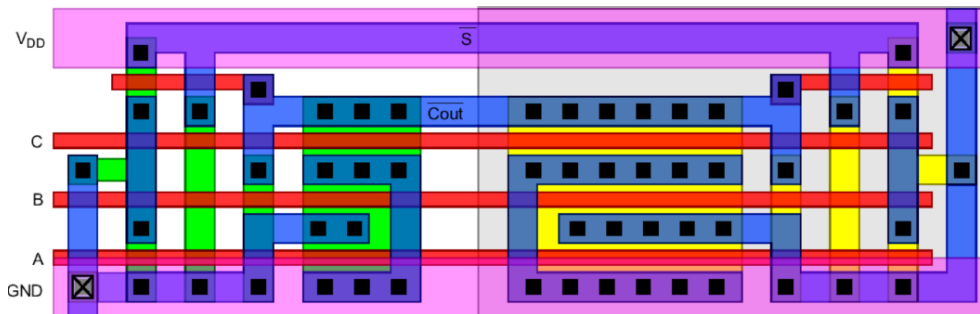
2. Úroveň hradel



3. Úroveň tranzistorů

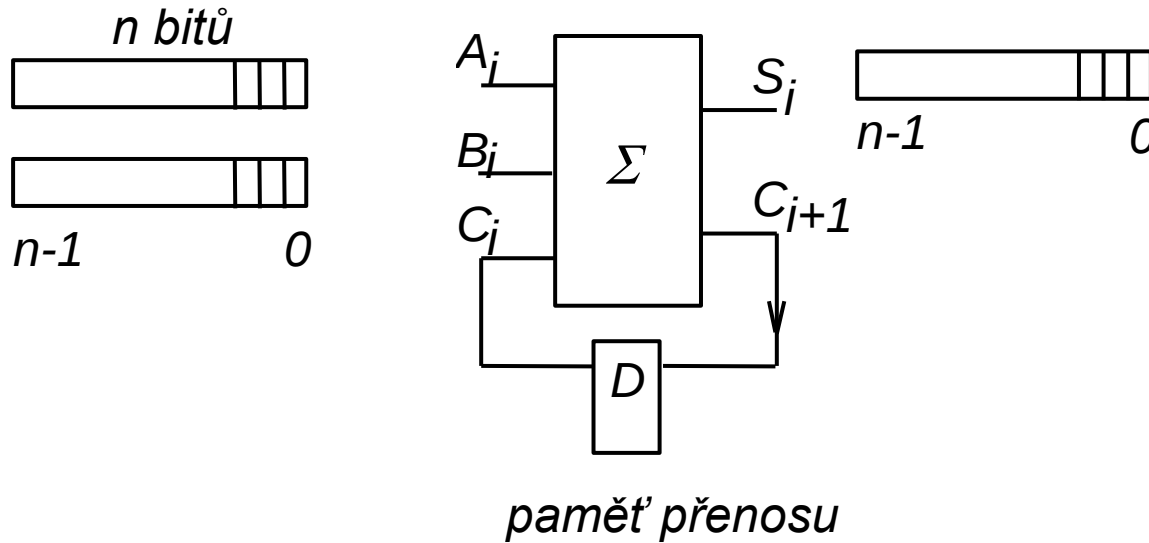


4. Úroveň fyzická (layout)



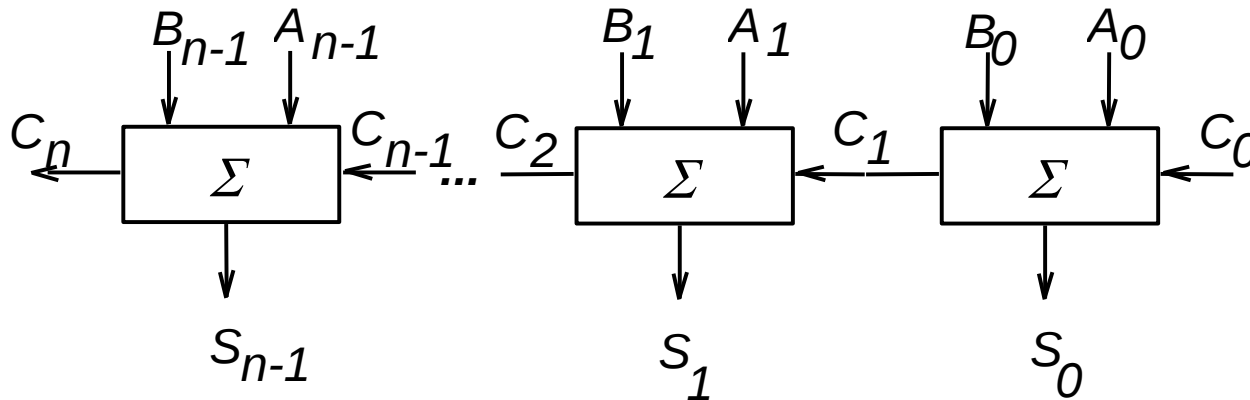
Čím nižší úroveň popisu, tím přesnější získáme znalost skutečných parametrů obvodu (plocha, zpoždění, příkon ...)

Sériová sčítačka



- Jde o jednobitovou sčítačku použitou pro **sériové sčítání** dvou n -bitových čísel, připravených ve vstupních registrech. Na výstupu S_i se postupně objevují bity součtu počínaje nejnižším bitem 0, a výstup přenosu C_{i+1} se zachycuje na dobu jednoho taktu (T_c) v klopném obvodu D (Carry Save). (Předpokládáme, že jde o synchronní sčítačku s taktem T_c).

Paralelní sčítačka s postupným přenosem



- Objeví-li se u první sčítačky výstup přenosu za dobu 2Δ , kde Δ je přenosové zpoždění jednoho logického členu, na výstupu druhé sčítačky je to již 4Δ , atd. Výstup přenosu se u poslední sčítačky objeví za dobu $2n\Delta$.
- Pro prakticky používané šířky sčítaček 32, 64 a 128 bitů je tato doba nepříjemně dlouhá.
- Je proto snaha navrhovat sčítačky s rychlým přenosem.
- Hledáme kompromis mezi zpožděním a počtem hradel (plochou).

Rozšířená sčítačka

C_i	A_i	B_i	S_i	P_i	G_i	C_{i+1}
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	0	0
1	0	1	0	1	0	1
1	1	0	0	1	0	1
1	1	1	1	0	1	1

Zavedeme dva pomocné výstupy:

P_i - **propagate carry** („1“, když přenos sčítačkou prochází, $A_i \neq B_i$)

G_i - **generate carry** (vznik přenosu bez ohledu na hodnotu C_i)

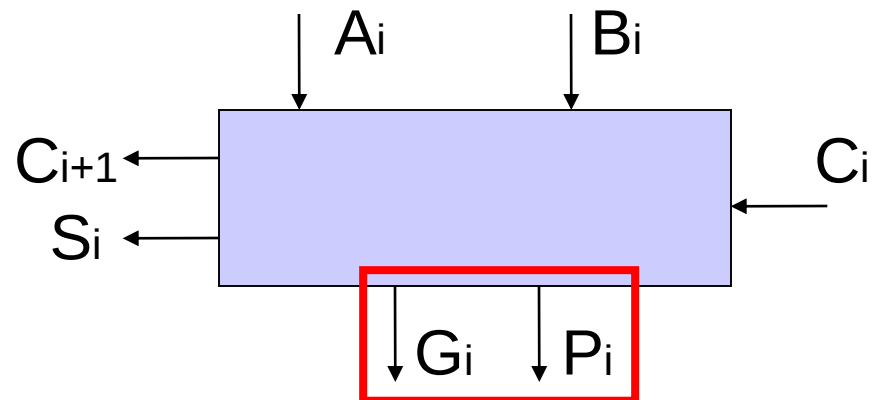
$S_i = A_i \oplus B_i \oplus C_i$ (zpoždění: 2Δ)

$P_i = A_i \oplus B_i$ (Δ)

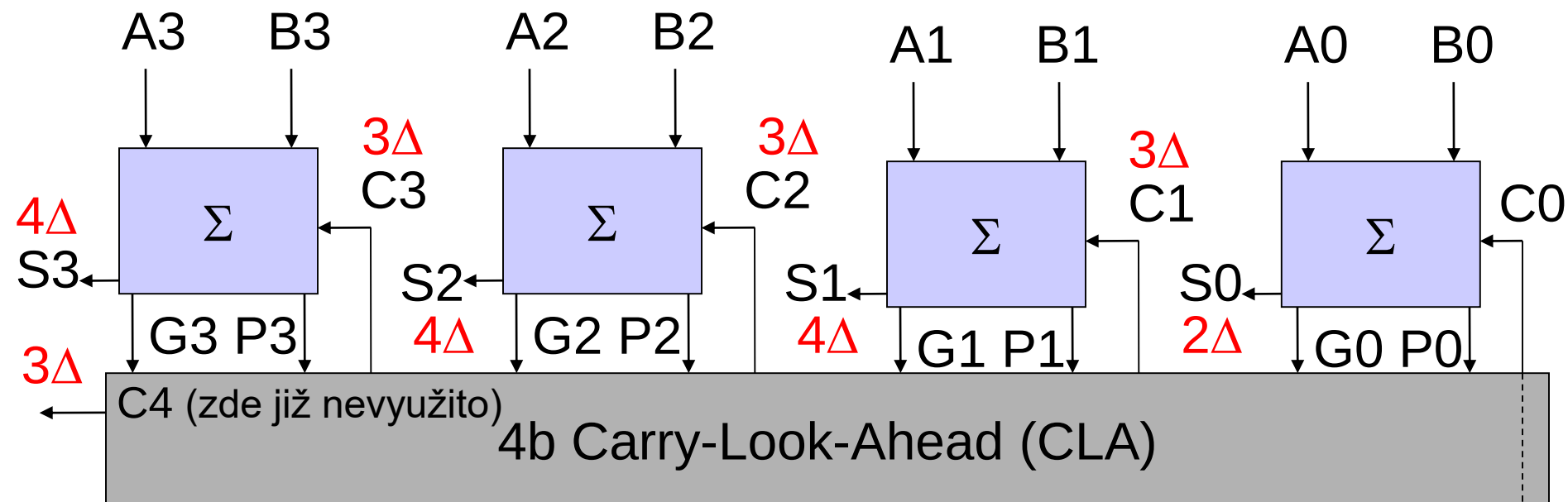
$G_i = A_i \cdot B_i$ (Δ)

$C_{i+1} = P_i \cdot C_i + G_i$ (3Δ)

Rozšířená sčítačka:



4b sčítačka s CLA – zpoždění 4Δ



CLA má
zpoždění
jen 2Δ .

$$C_0 = CI$$

$$C_1 = C_0 P_0 + G_0$$

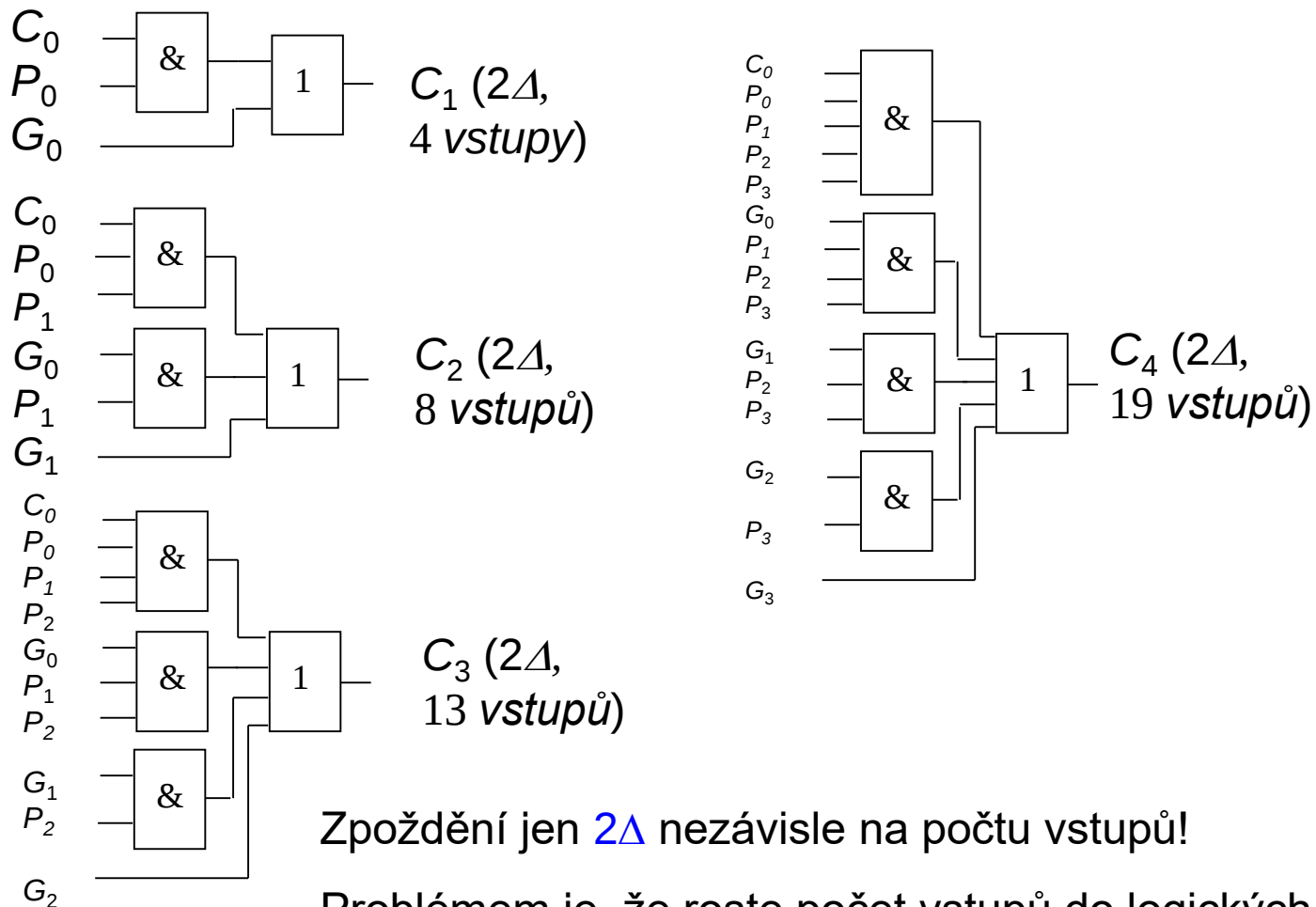
$$C_2 = C_1 P_1 + G_1 = C_0 P_0 P_1 + G_0 P_1 + G_1$$

$$C_3 = C_2 P_2 + G_2 = C_0 P_0 P_1 P_2 + G_0 P_1 P_2 + G_1 P_2 + G_2$$

$$C_4 = C_3 P_3 + G_3 = C_0 P_0 P_1 P_2 P_3 + G_0 P_1 P_2 P_3 + G_1 P_2 P_3 + G_2 P_3 + G_3$$

CI

Logický obvod CLA



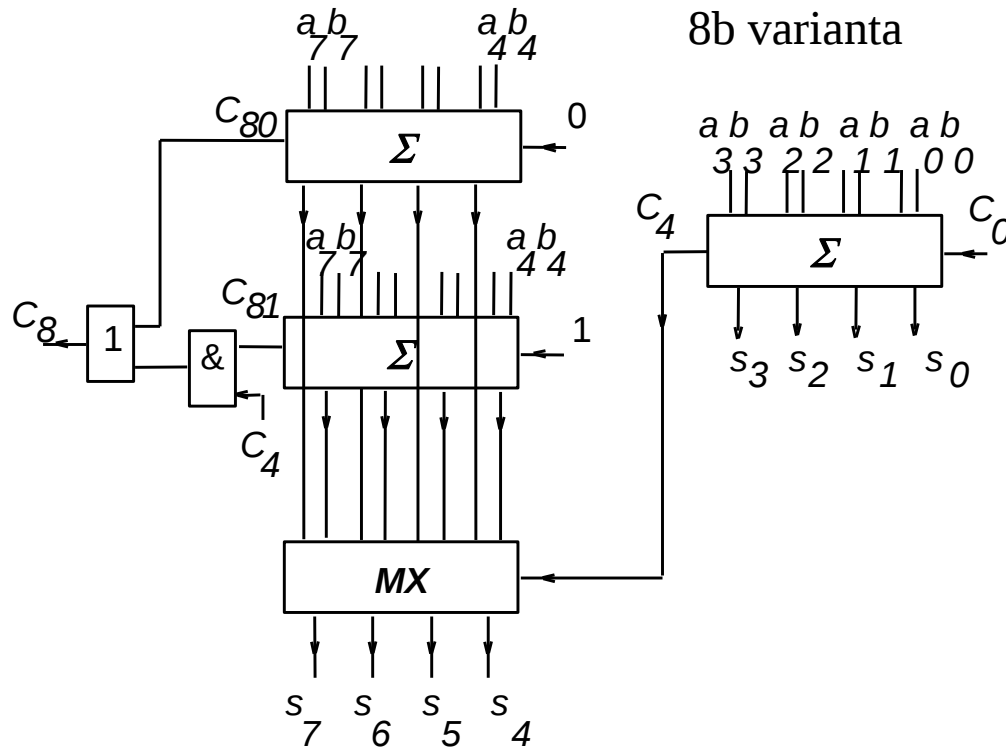
Zpoždění jen 2Δ nezávisle na počtu vstupů!

Problémem je, že roste počet vstupů do logických členů AND a OR. Logický člen s k vstupy lze fyzicky realizovat jen pro malé hodnoty k .

n-bitová sčítačka s CLA – konst. zpoždění 4Δ

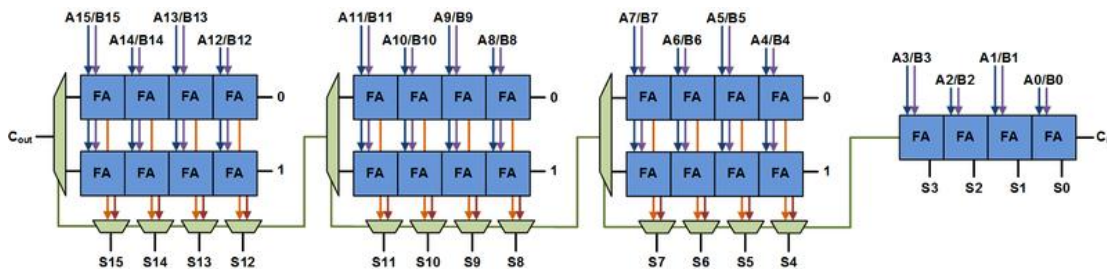
- Cesty pro šíření postupného přenosu jsou zrušeny a na vstupy přenosu všech sčítaček se přivádějí příslušné výstupy **generátoru přenosu (obvodu CLA)**.
- Funkce P_i , G_i se tvoří se zpožděním Δ , v čase 3Δ jsou k dispozici všechny rychlé přenosy, a součet je tedy vytvořen v čase 4Δ .
- Popsané uspořádání **n-bitové** sčítačky je **nejrychlejší možné řešení**.
- Složitost (zejména počet vstupů u log. členů) dvoustupňového generátoru přenosu však roste pro rostoucí šířku sčítačky s **druhou mocninou šířky**. Pro šířky 32 a 64 bitů je toto řešení již technologicky nepřijatelné.
- Byla proto navržena řešení umožňující za cenu nárůstu zpoždění zmenšit potřebnou plochu na čipu:
 - stromový generátor přenosu s CLA – např. pomocí několika 4b obvodů CLA uspořádaných do stromu
 - **výběr přenosu – viz příklad**
 - přeskakování přenosu atd.

Sčítačka s výběrem přenosu



- 8-bitová sčítačka je rozdělena na dvě 4-bitové sčítačky.
- Horní sčítačka je zdvojená, přičemž jedna má vstup přenosu 0, a druhá 1.
- Obě připraví výsledek, a 4-vstupový multiplexor pak vybere jednu z nich podle hodnoty přenosu C_4 .
- Obvodové řešení se tedy prodražilo o více než 50% vzhledem ke sčítačce s postupným přenosem.

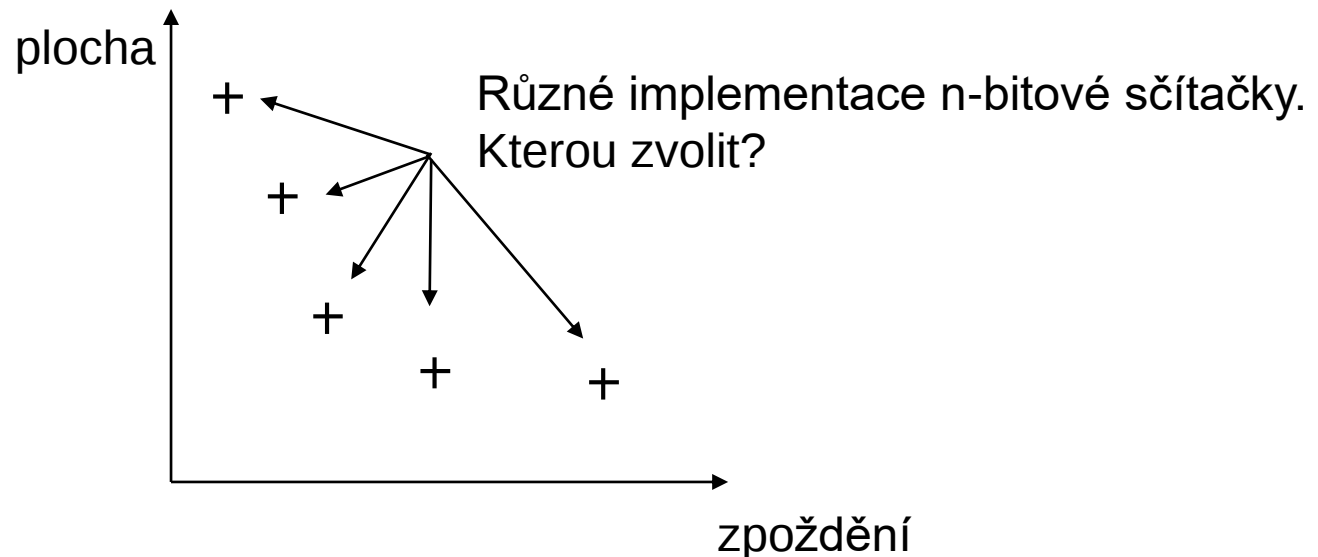
16b varianta



- Pro větší šířky sčítaček se postupuje obdobně; každý blok je zdvojen a je přidán výběrový multiplexor.

Hodnocení složitosti logických obvodů

- Jde o nalezení **kompromisního řešení** mezi **cenou** a **výkonností**. V poslední době se navíc hledá kompromis s příkonem.
- **Cena** – popíše se např. součtem počtu vstupů všech použitých logických členů, součtem počtu logických členů, plochou na čipu apod.
- **Výkonnost** – popíše se hodnotou nejdelšího přenosového zpoždění daného obvodu (které následně určuje $f_{\max}$).



Operační rychlost a plocha různých typů sčítaček šířky n

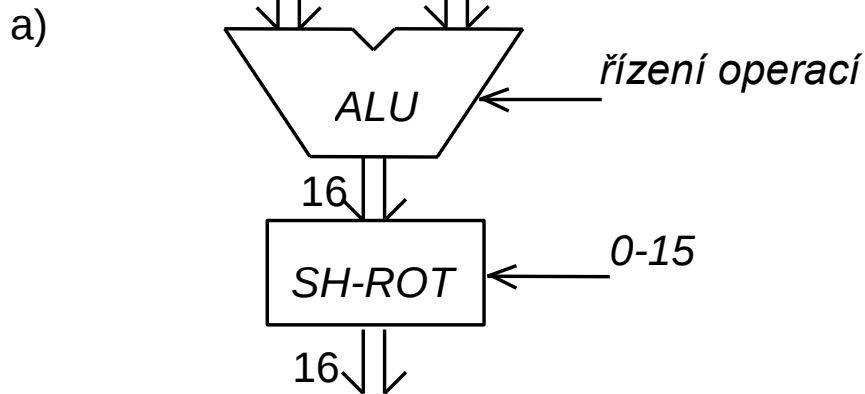
Typ	doba výpočtu	plocha
Postupný přenos	$O(n)$	$O(n)$
2 – stupňový CLA	4	$O(n^2)$
Stromový CLA k -nární	$O(\log_k n)$	$O(n \log_k n)$
Přeskakování přenosu	$O(\sqrt{n})$	$O(n)$
Výběr přenosu	$O(\sqrt{n})$	$O(n)$

kde k je počet dílčích přenosů, které se skládají v jednom uzlu stromu.

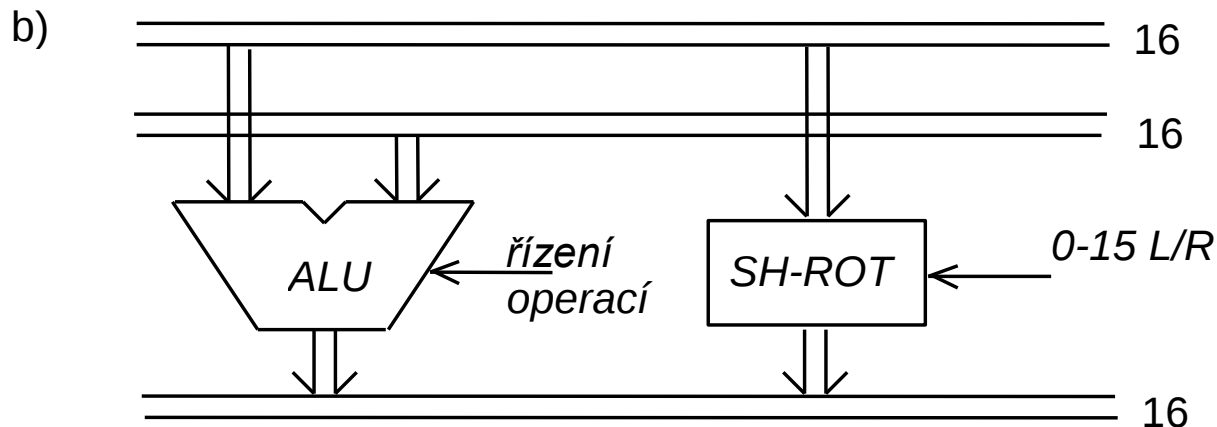
Další operace ALU – posuvy a rotace

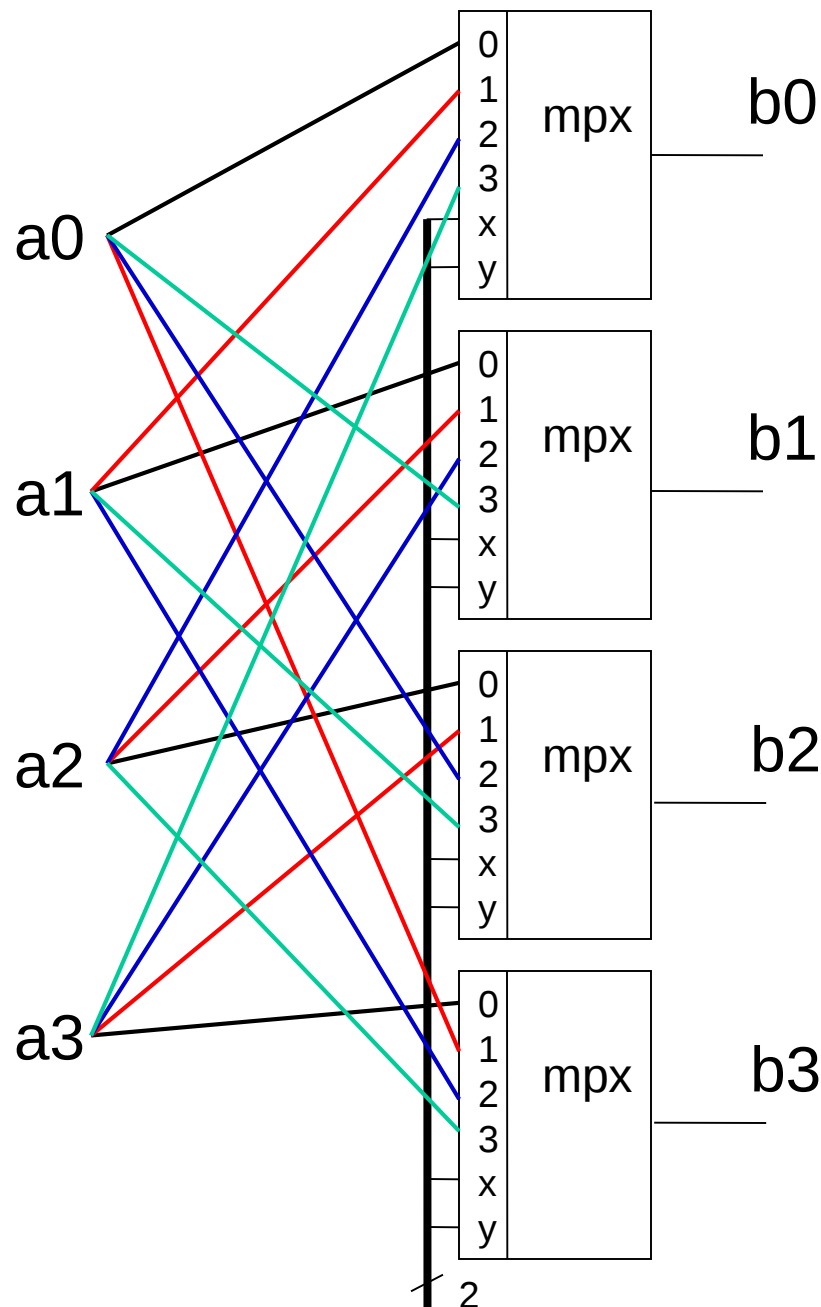
Opakování:

- Rotace
- Logický a aritmetický posuv



- Blok posuvů a rotací je umístěn buď za výstupem ALU, nebo paralelně k bloku ALU.
- **Řídicí vstupy:** směr posuvu, jeho typ (logický nebo aritmetický), a počet bitů, o které se posouvá.
- U každého operačního bloku se musí pamatovat na operaci **identity**.
- Implementace: válcový posouvač (Barrel Shifter) – pomocí multiplexorů, ne posuvný registr!!





4b válcový
posouvač
pro rotace vpravo
(4 x 4-vst. MUX,
zpoždění 2Δ)

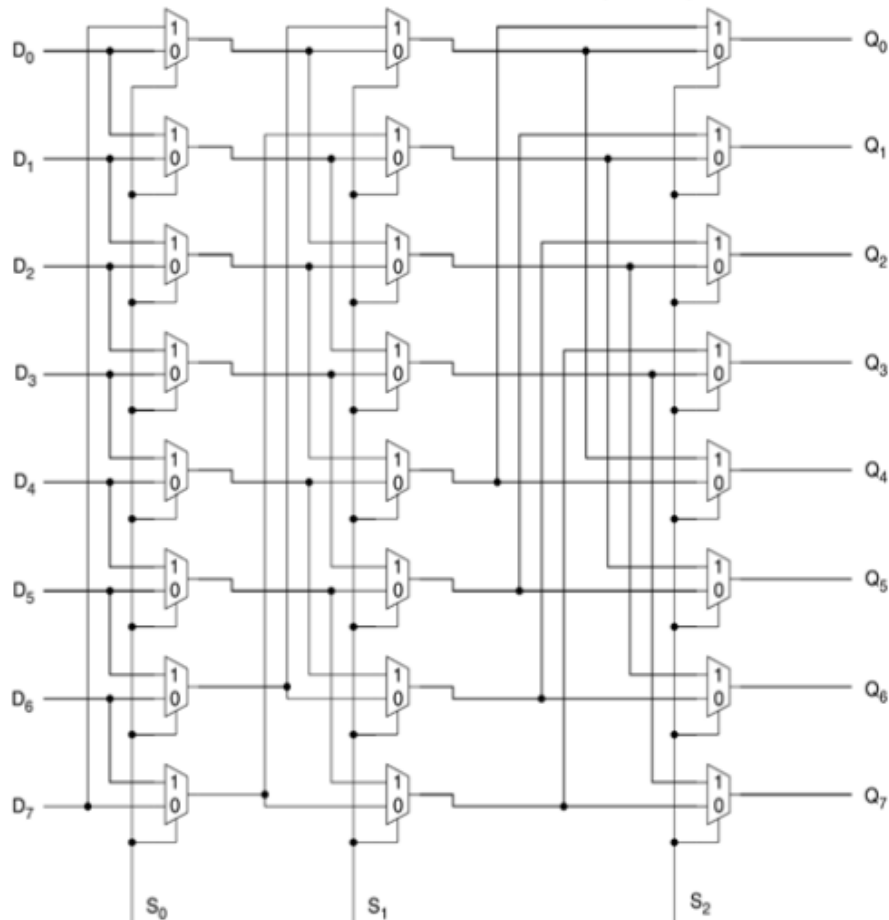
b_3	b_2	b_1	b_0	
↓	↓	↓	↓	
a_3	a_2	a_1	a_0	0 bit
a_0	a_3	a_2	a_1	1 bit
a_1	a_0	a_3	a_2	2 bit
a_2	a_1	a_0	a_3	3 bit

Možnosti realizace 8b válcového posouvače (rotace vlevo)

<http://verilog-code.blogspot.com/2013/09/barrel-shifter-design-using-21-mux.html>

(8) - 8 x 8-vstupový multiplexor: dražší řešení, zpoždění 2Δ

(2, 2, 2) - 24 x 2-MUX, zpoždění 6Δ , levnější řešení, viz obrázek



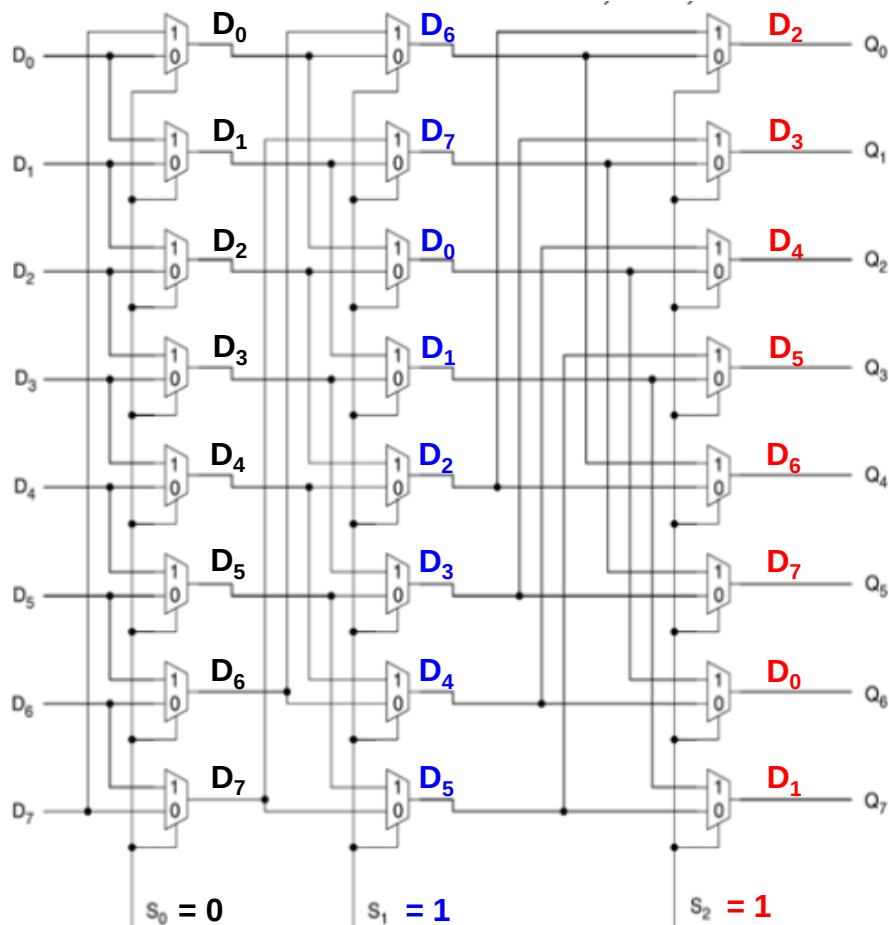
Posun o 0
nebo 1 bit.

Posun o 0
nebo 2 bity.

Posun o 0
nebo 4 bity.

Př. 8b válcový posouvač (rotace vlevo)

<http://verilog-code.blogspot.com/2013/09/barrel-shifter-design-using-21-mux.html>



Př. Rotace o 6b vlevo
 $S_2 S_1 S_0 = 110$

Na výstupech

$Q_7 \ Q_6 \ Q_5 \ Q_4 \ Q_3 \ Q_2 \ Q_1 \ Q_0$
budou vstupní bity:
 $D_1 \ D_0 \ D_7 \ D_6 \ D_5 \ D_4 \ D_3 \ D_2$

Poznámky

- Při optimalizaci vnitřní struktury válcového posouvače se kromě základních kritérií, což je cena (počet logických členů příp. počet vstupů logických členů) a přenosové zpoždění, používá pro každou technologii ještě povolené vstupní a výstupní větvení, příp. ještě další kritéria. Může se pak dospět ke struktuře, která používá v jednotlivých stupních **multiplexory s rozdílným počtem vstupů**.
- Je vhodné pak popisovat struktury symbolicky. Např. jednostupňové uspořádání 16 bit. posouvače s 16-vstupovými multiplexory se zapíše (16), čtyřstupňové uspořádání se 2-vstupovými multiplexory jako (2,2,2,2), se 4-vstupovými MUX (4,4), smíšená struktura např. (2,4,2) atd. Přenosové zpoždění válcového posouvače se strukturou (2,2,2,2) je 8 jednotkových zpoždění Δ .
- Poznámka: Logická struktura multiplexoru je dvoustupňová, s přenosovým zpožděním 2Δ .