Pokročilé architektury počítačů

– Podklady pro cvičení –

Paralelný dvojkový sumátor se zrychlenými přenosy a s předvídáním

České vysoké učení technické, Fakulta elektrotechnická

Sumátor CLA (Carry Look-Ahead) - motivace

- paralelní sumátor se zvlněným přenosem je pomalý
- paralelní sumátor s minimálním zpožděním je prakticky nepoužitelný (pro 64-bitový sumátor bychom potřebovali 10^{20} hradel)
- sumátor CLA nabízí dostatečné zrychlení v porovnaní se sumátorem se zvlněným přenosem při přijatelném nárůste ceny hardwaru,
  (pro 64-bitový sumátor se zvýši cena HW o necelých 50% a rychlost se zvýši 9:1, což představuje významné zlepšení *rychlost/cena)*

Sumátor CLA (Carry Look-Ahead)

Mějme dvě n-bitová čísla x a y. Jak vypočteme jejich součet? Víme, že součet v j-tem řádu je dán součtem $x_j + y_j + c_j$, kde c_j je přenos v j-tem řádu. Klíčovým úkolem proto bude co nejrychleji stanovit přenos…

Uvažujme dvě 12-bitové čísla x =1101 0110 1011 a y = 0011 1011 1101.
Z přenosů v jednotlivých řádech můžeme sestavit 12-bitový vektor c.
Součet s je pak dán s = x + y + c po bitech, tj. $s = x \oplus y \oplus c$.

```
x = 1101 0110 1011
y = 0011 1011 1101
c = ???? ???? ????
------------------
s = ???? ???? ????
```

Jak můžeme urychlit výpočet součtu? = Jak můžeme urychlit výpočet přenosu?

Tyto rovnice jsou klíčové pro pochopení principu…

vstupy			výstupy		
x_j	y_j	c_j	s_j	c_{j+1}	
0	0	0	0	0	
0	0	1	1	0	
0	1	0	1	0	$c_{j+1} = c_j$
0	1	1	0	1	$c_{j+1} = c_j$
1	0	0	1	0	$c_{j+1} = c_j$
1	0	1	0	1	$c_{j+1} = c_j$
1	1	0	0	1	
1	1	1	1	1	

x_j	y_j	c_{j+1}	
0	0	0	
0	1	c_j	-> šíření přenosu
1	0	c_j	-> šíření přenosu
1	1	1	-> generování přenosu

$$c_{j+1} = x_j y_j \vee (x_j \oplus y_j) c_j$$

$$s_j = \bar{x}_j \bar{y}_j c_j \vee \bar{x}_j y_j \bar{c}_j \vee x_j \bar{y}_j \bar{c}_j \vee x_j y_j c_j$$

$$= c_j (\bar{x}_j \bar{y}_j \vee x_j y_j) \vee \bar{c}_j (x_j \bar{y}_j \vee \bar{x}_j y_j) = c_j \left(\overline{x_j \oplus y_j}\right) \vee \bar{c}_j (x_j \oplus y_j)$$

Tyto rovnice jsou klíčové pro pochopení principu…

Označme:

- generování přenosu:

$$g_j = x_j y_j$$

- šíření přenosu (propagation):

$$p_j = x_j \oplus y_j = x_j \bar{y}_j \vee \bar{x}_j y_j$$

Pak:

- součet v j-tem řádu:

$$s_j = c_j \left(\overline{x_j \oplus y_j}\right) \vee \bar{c}_j \left(x_j \oplus y_j\right) = c_j \bar{p}_j \vee \bar{c}_j p_j = p_j \oplus c_j$$

- přenos do vyššího (j+1) řádu:

$$c_{j+1} = x_j y_j \vee \left(x_j \oplus y_j\right) c_j = g_j \vee p_j c_j$$

CLA

Takže platí:

$c_1 = g_0 \vee p_0c_0$

$c_2 = g_1 \vee p_1c_1 = g_1 \vee p_1(g_0 \vee p_0c_0) = g_1 \vee p_1g_0 \vee p_1p_0c_0$

$c_3 = g_2 \vee p_2c_2 = g_2 \vee p_2(g_1 \vee p_1g_0 \vee p_1p_0c_0) = g_2 \vee p_2g_1 \vee p_2p_1g_0 \vee p_2p_1p_0c_0$

$c_4 = g_3 \vee p_3c_3 = \ldots = g_3 \vee p_3g_2 \vee p_3p_2g_1 \vee p_3p_2p_1g_0 \vee p_3p_2p_1p_0c_0$

$c_5 = \ldots$

Například rovnici pro c_3 je možné číst následovně:

Přenos do 3. řádu nastane **pokud** přenos byl generován v 2. řádu **nebo** se 2. řádem šíří a byl generován v 1. řádu **nebo** se šíří 2. aj 1. řádem a byl generován 0-tým řádem **nebo** se šíří druhým, prvním aj 0-tým řádem a byl v c_0 (c_0=1).

CLA

Vidíme, že předchozí rovnice jsou iterativního tvaru a nic nám nebrání abychom pokračovali v tomto procesu rozpisování přenosu i nadále (pro c_5, c_6, …). Tyto rovnice jsou zároveň druhého řádu, takže jednotku CLA pro přenos s předvídáním možno rozšířit tak, aby pokrývala více bitů bez zvýšení zpoždění. Avšak, když zvýšíme počet bitů, velkost a počet hradel se taktéž zvýši: c_3 potřebuje 4-vstupové hradla, c_4 by potřebovalo 5-vstupové hradla, c_5 by potřebovalo 6-vstupové hradla atd. Počet bitů, které jednotka CLA může pokrýt je limitovaný počtem vstupů použitých hradel.

CLA – návrh 64-bitového sumátoru

Význam a použití rovnic budeme demonstrovat na 64-bitovém sumátoru. Při návrhu použijeme 4-bitovou jednotku CLA pro přenos s předvídáním, proto 64-bitový sumátor rozdělíme do 4-bitových skupin (grup), kde grupa 0 zahrnuje bity 0 až 3, grupa 1 bity 4 až 7 atd. Potom definujeme grupové generování Gg a grupové šíření Gp, kde pro grupové generování v grupě 0 (bity 0 až 3) platí rovnice:

$$Gg_0 = g_3 \vee p_3 g_2 \vee p_3 p_2 g_1 \vee p_3 p_2 p_1 g_0$$

přičemž přenos byl generován někde v grupě a všechny vyšší bity (p_i) splňují podmínku šíření. Pro grupové šíření grupy 0 bude platit rovnice:

$$Gp_0 = p_3 p_2 p_1 p_0$$

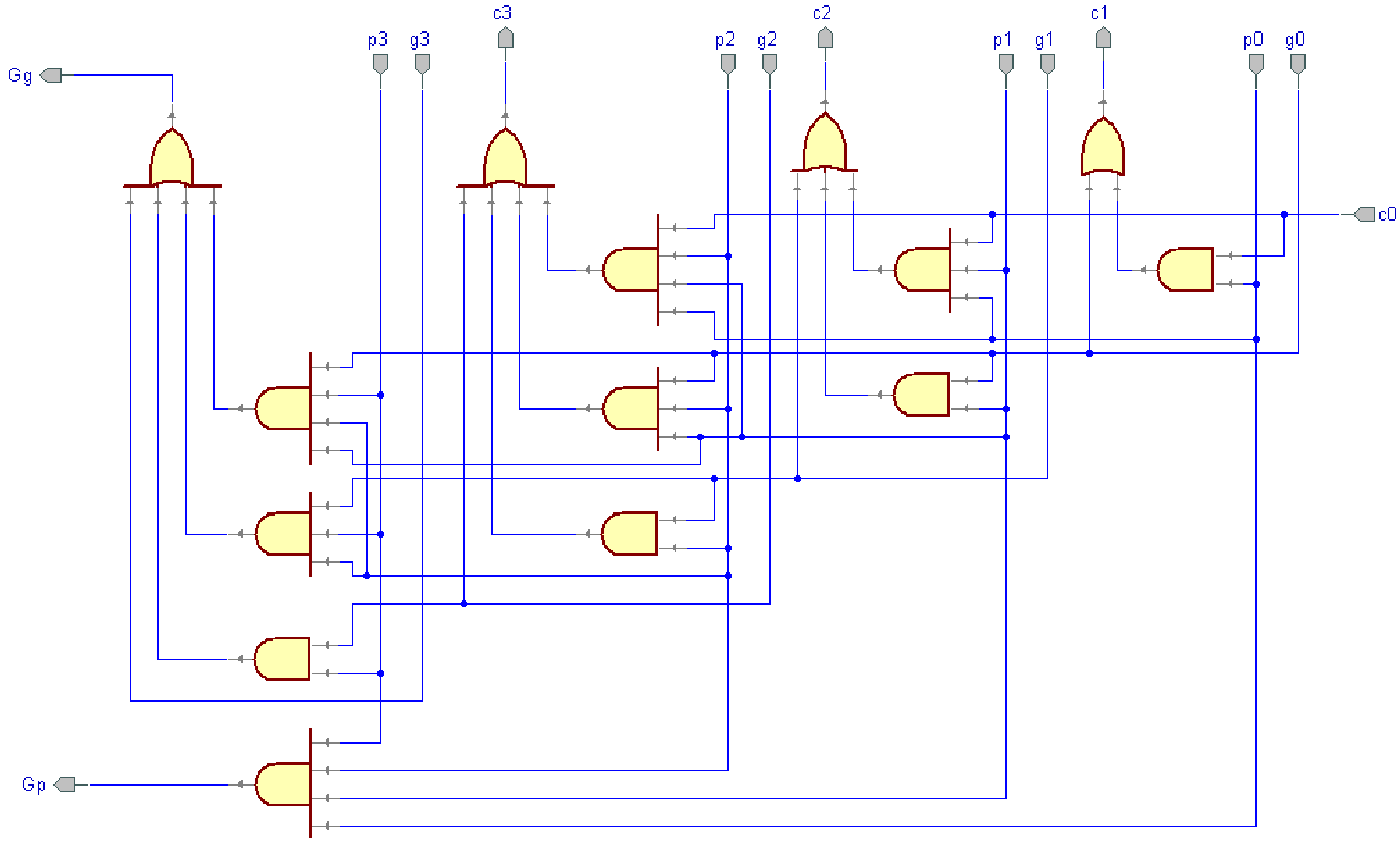
takže přenos do grupy bude procházet celou grupou. Čili přenos z grupy existuje pokud se přenos generuje někde v grupě a šíří se ní, nebo existuje přenos do grupy, který se šíři skrz grupu. Na základe toho můžeme definovat grupový přenos Gc_1, který se rovná přenosu c_4 podle rovnice:

$$c_4 = g_3 \vee p_3 c_3 \quad \text{takže} \quad c_4 = Gc_1 = Gg_0 \vee Gp_0 \, Gc_0$$

kde $Gc_0 = c_0$ je přenos do grupy 0.

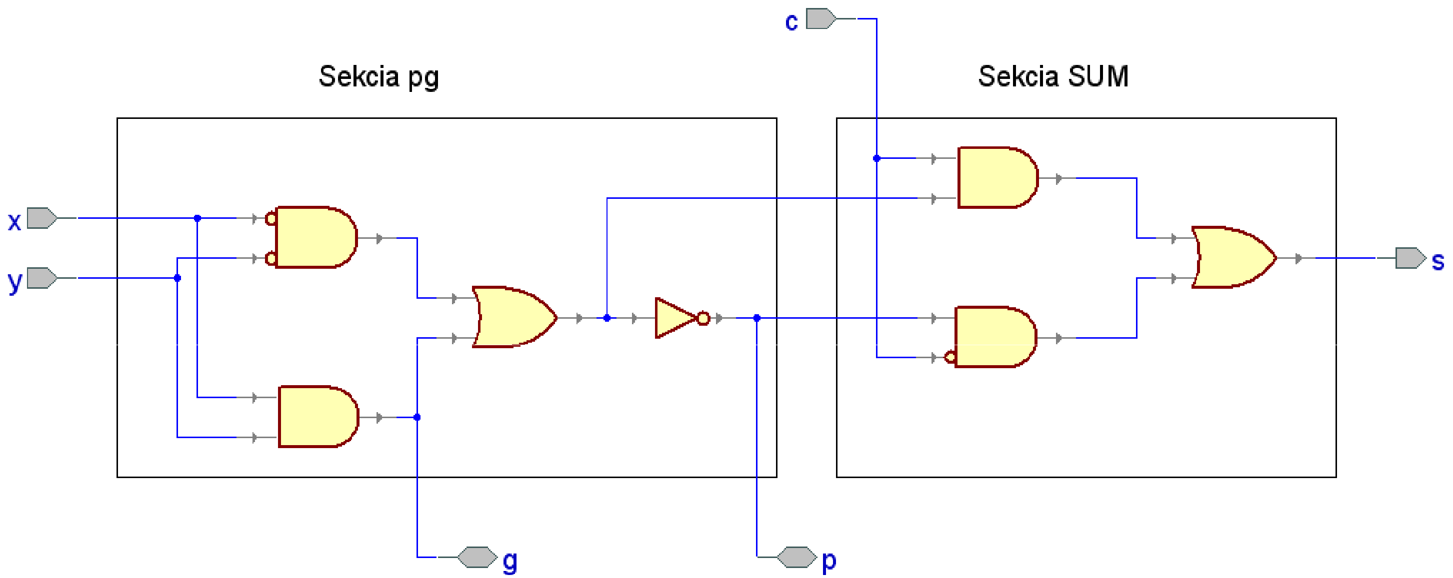
CLA – návrh 64-bitového sumátoru

Vycházejíc z rovnic pro c_1 až c_3 a rovnic pro Gg_0 a Gp_0 můžeme sestrojit jednotu CLA pro přenos s předvídáním následovně:



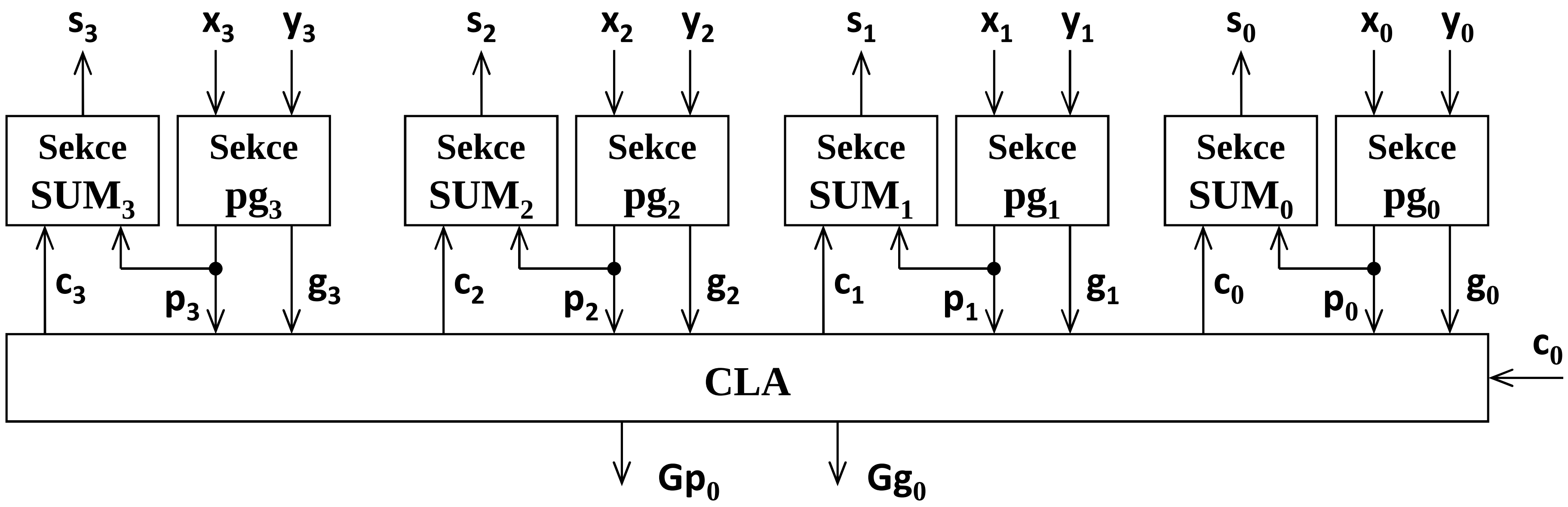
CLA – návrh 64-bitového sumátoru

Dále na základě rovnic pro generování a šíření přenosu a součet v j-tém řádu můžeme sestrojit úplnou sčítačku upravenou pro potřeby sumátoru CLA



CLA – návrh 64-bitového sumátoru

Zapojení sumátoru pro bity 0 až 3 je ukázáno na následujícím obrázku:



CLA – návrh 64-bitového sumátoru

Podobným způsobem jako jsme stanovili c_4 můžeme odvodit rovnice pro grupové přenosy:

$$c_8 = \boldsymbol{Gc_2} = Gg_1 \vee Gp_1 Gc_1 = Gg_1 \vee Gp_1(Gg_0 \vee Gp_0 Gc_0) =$$

$$= Gg_1 \vee Gp_1 Gg_0 \vee Gp_1 Gp_0 Gc_0$$

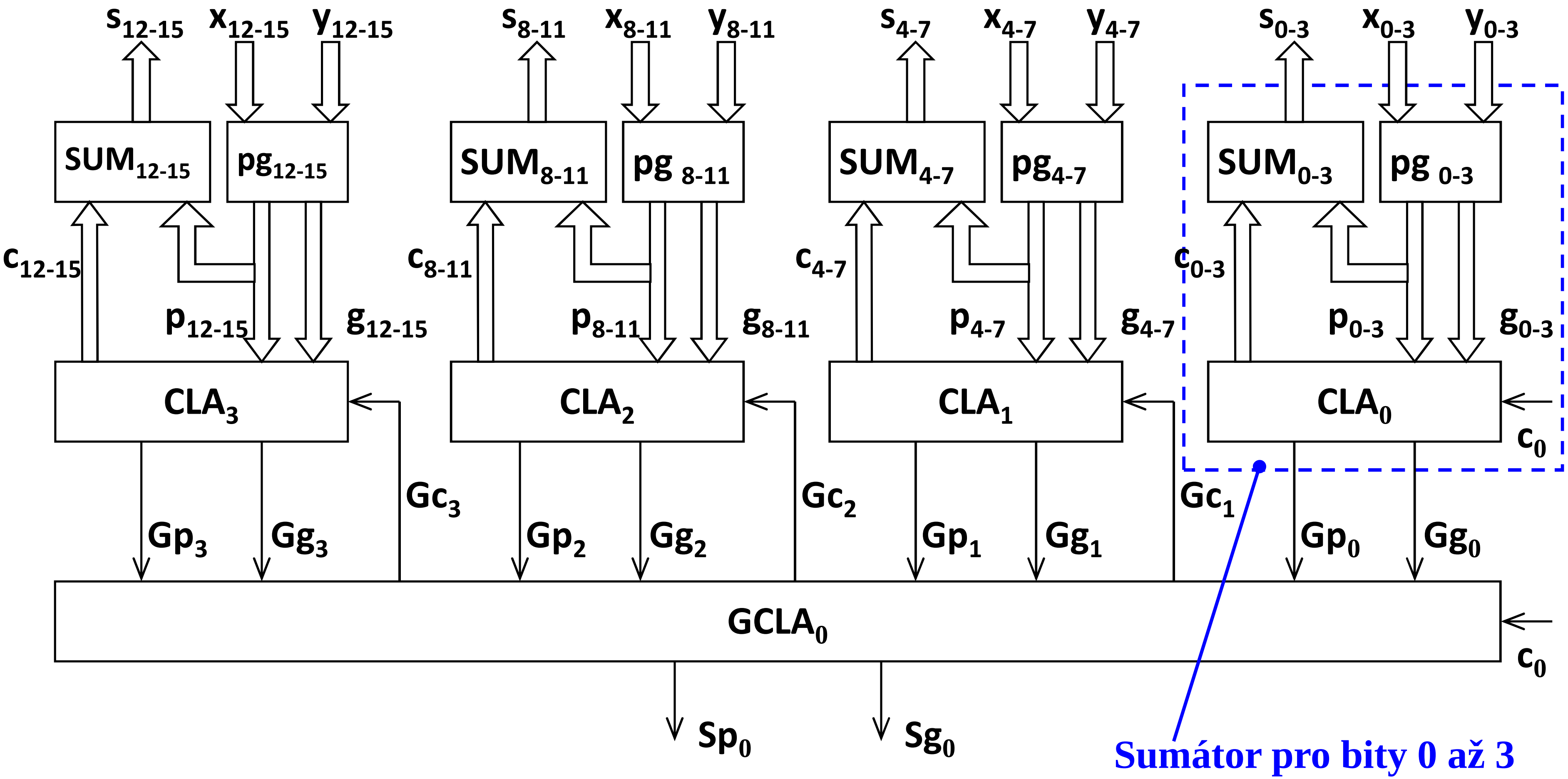
$$c_{12} = \boldsymbol{Gc_3} = Gg_2 \vee Gp_2 Gc_2 = Gg_2 \vee Gp_2(Gg_1 \vee Gp_1 Gg_0 \vee)$$

$$= Gg_2 \vee Gp_2 Gg_1 \vee Gp_2 Gp_1 Gg_0 \vee Gp_2 Gp_1 Gp_0 Gc_0$$

S výjimkou názvů proměnných vidíme, že výše uvedené rovnice pro c_4, c_8 a c_{12} jsou identické s rovnicemi pro c_1 až c_3. Čili, členy grupových přenosů lze vytvářet stejnými obvody CLA, jako pro obyčejné přenosy. Propojení sumátorů a jednotek CLA pro bity 0 až 15 je ukázáno na následujícím obrázku:

CLA – návrh 64-bitového sumátoru

16-bitový sumátor s grupovým předvídáním:



CLA – návrh 64-bitového sumátoru

Seskupením čtyř výše uvedených 16-bitových sumátorů získáme 64-bitový sumátor. Jinak řečeno 64-bitový sumátor je možné rozdělit na 4 sekce po 16 bitech.

Definujme sekciové generování Sg a sekciové šíření Sp analogicky jako při grupových přenosech.

$$Sg_0 = Gg_3 \vee Gp_3\, Gc_3$$

kde

$$Gc_3 = Gg_2 \vee Gp_2\, Gg_1 \vee Gp_2\, Gp_1\, Gg_0 \vee Gp_2\, Gp_1\, Gp_0\, Gc_0$$

potom

$$Sg_0 = Gg_3 \vee Gp_3(Gg_2 \vee Gp_2\, Gg_1 \vee Gp_2\, Gp_1\, Gg_0 \vee Gp_2\, Gp_1\, Gp_0\, Gc_0) =$$

$$= Gg_3 \vee Gp_3\, Gg_2 \vee Gp_3\, Gp_2\, Gg_1 \vee Gp_3\, Gp_2\, Gp_1\, Gg_0 \vee Gp_3\, Gp_2\, Gp_1\, Gp_0\, Gc_0$$

a

$$Sp_0 = Gp_3\, Gp_2\, Gp_1\, Gp_0$$

CLA – návrh 64-bitového sumátoru

Analogicky**:**

$c_{16} = Gc_4 = \boldsymbol{Sc_1} = Sg_0 \vee Sp_0\ Sc_0$

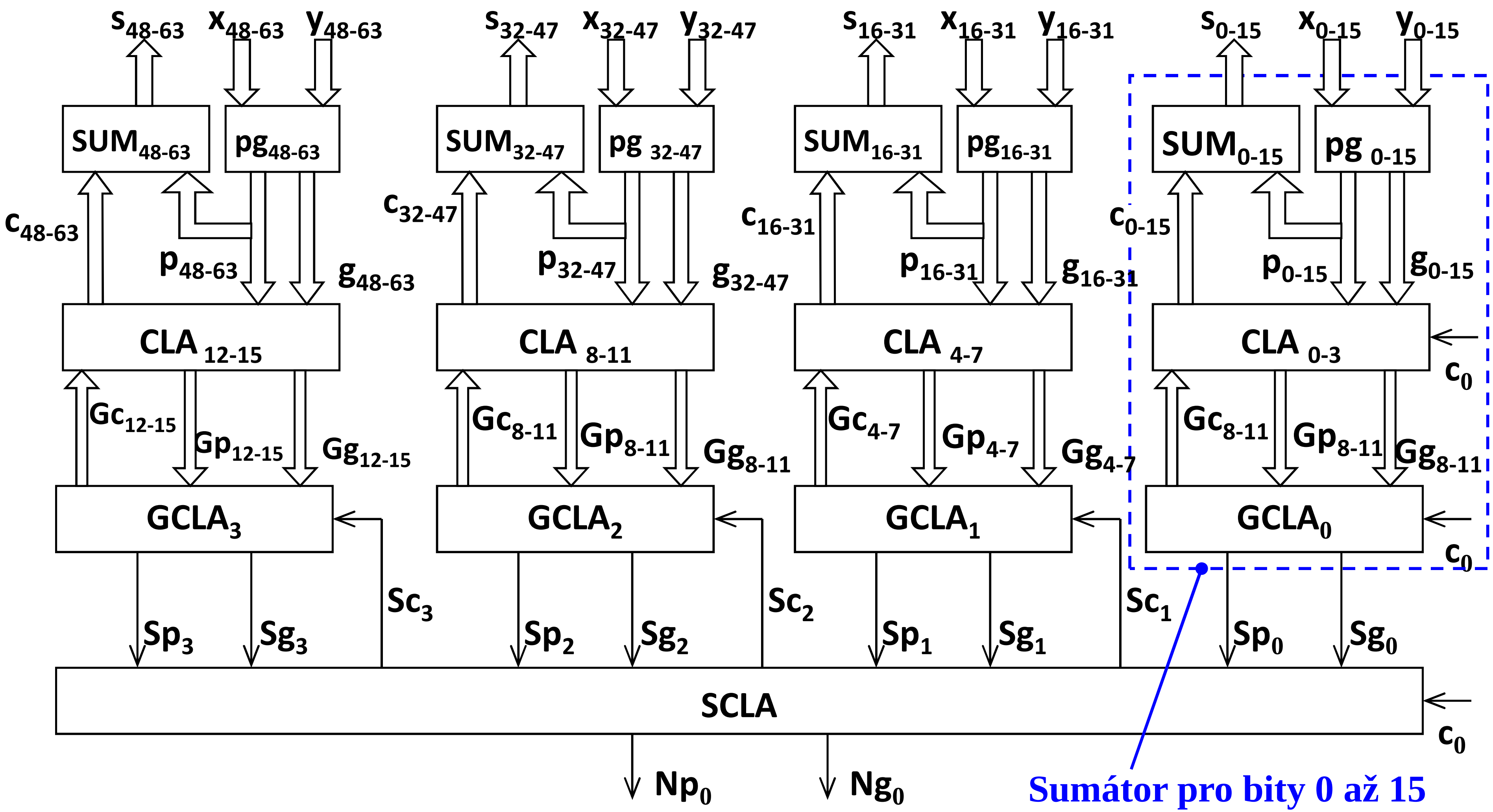
$c_{32} = Gc_8 = \boldsymbol{Sc_2} = Sg_1 \vee Sp_1\ Sc_1 = Sg_1 \vee Sp_1(Sg_0 \vee Sp_0\ Sc_0) =$
$= Sg_1 \vee Sp_1\ Sg_0 \vee Sp_1\ Sp_0\ Sc_0$

$c_{48} = Gc_{12} = \boldsymbol{Sc_3} = Sg_2 \vee Sp_2\ Sc_2 = Sg_2 \vee Sp_2(Sg_1 \vee Sp_1\ Sg_0 \vee Sp_1\ Sp_0\ Sc_0) =$
$= Sg_2 \vee Sp_2\ Sg_1 \vee Sp_2\ Sp_1\ Sg_0 \vee Sp_2\ Sp_1\ Sp_0\ Sc_0$

Uváděné rovnice mají stejný tvar jako rovnice pro původní CLA, lze tedy opět použít stejné obvody.

CLA – návrh 64-bitového sumátoru

64-bitový sumátor se sekciovým předvídáním:



CLA – návrh 64-bitového sumátoru

Jelikož neexistují další úrovně předvídání, třeba odvodit konečný výstupní přenos c_{64}:

$$c_{64} = Gc_{16} = Sc_4 = Sg_3 \vee Sp_3\, Sc_3 =$$
$$= Sg_3 \vee Sp_3(Sg_2 \vee Sp_2\, Sg_1 \vee Sp_2\, Sp_1\, Sg_0 \vee Sp_2\, Sp_1\, Sp_0\, Sc_0) =$$
$$= Sg_3 \vee Sp_3\, Sg_2 \vee Sp_3\, Sp_2\, Sg_1 \vee Sp_3\, Sp_2\, Sp_1\, Sg_0 \vee Sp_3\, Sp_2\, Sp_1\, Sp_0\, Sc_0$$

Abychom dostali tento člen, můžeme použít výstupy SCLA značené jako Np_0 (pro šíření přenosu) a Ng_0 (pro generování přenosu) a $Sc_0 = c_0$.