

5. VEZJA ZA OPRAVLJANJE ARITMETIČNIH OPERACIJ

5. 1 Enomestni seštevalniki

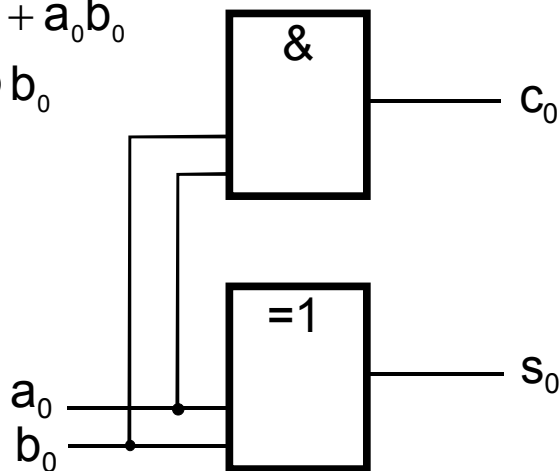
5. 1. 1 Polovični seštevalnik: $a_0 + b_0 \Rightarrow s_0, c_0$

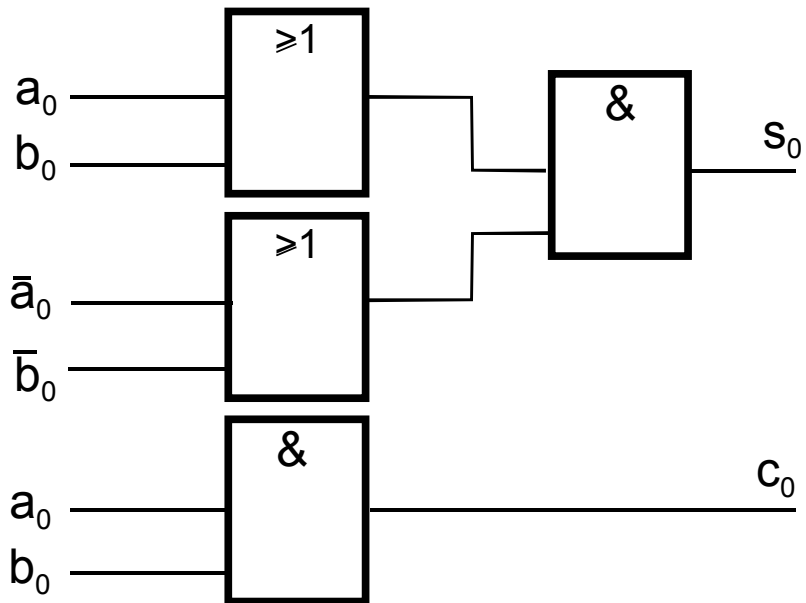
b_0	a_0	s_0	c_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$s_0 = a_0 \bar{b}_0 + \bar{a}_0 b_0$$

$$s_0 = a_0 \oplus b_0$$

$$c_0 = a_0 b_0$$





Če izraz:

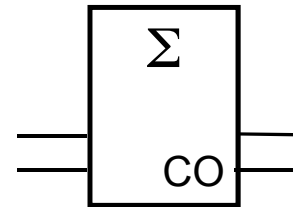
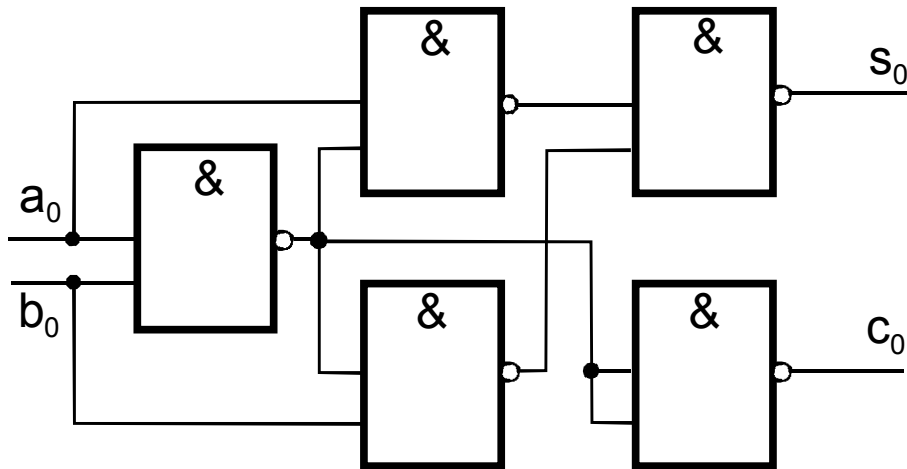
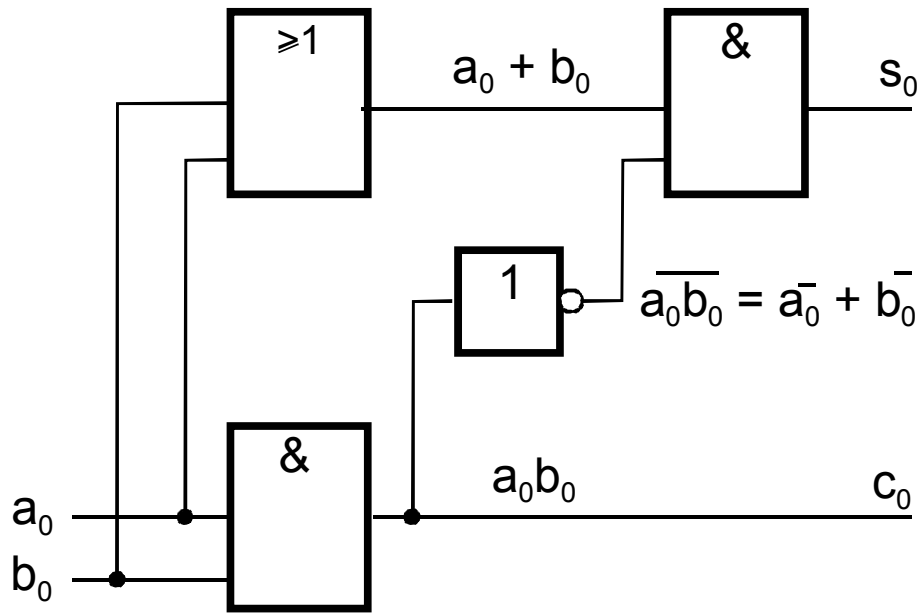
$$s_0 = a_0 \bar{b}_0 + \bar{a}_0 b_0$$

razširimo na:

$$s_0 = a_0 \bar{b}_0 + \bar{a}_0 b_0 + a_0 \bar{a}_0 + b_0 \bar{b}_0$$

$$s_0 = a_0 (\bar{b}_0 + \bar{a}_0) + b_0 (\bar{b}_0 + \bar{a}_0)$$

$$s_0 = (a_0 + b_0) (\bar{b}_0 + \bar{a}_0)$$



5. 1. 2 Popolni seštevalnik

b_i	a_i	c_{i-1}	s_i	c_i
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

$$s_i = a_i \bar{b}_i \bar{c}_{i-1} + \bar{a}_i b_i \bar{c}_{i-1} + \bar{a}_i \bar{b}_i c_{i-1} + a_i b_i c_{i-1}$$

$$c_i = a_i b_i \bar{c}_{i-1} + a_i \bar{b}_i c_{i-1} + \bar{a}_i b_i c_{i-1} + a_i b_i c_{i-1}$$

$$s_i = \bar{c}_{i-1} (a_i \bar{b}_i + \bar{a}_i b_i) + c_{i-1} (\bar{a}_i \bar{b}_i + a_i b_i)$$

$$s_i = \bar{c}_{i-1} (a_i \oplus b_i) + c_{i-1} (\overline{a_i \oplus b_i})$$

$$\overline{a \oplus b} = \overline{a \bar{b} + \bar{a} b} = (\overline{a \bar{b}})(\overline{\bar{a} b}) = (\bar{a} + b)(a + \bar{b}) = (a + \bar{b})(\bar{a} + b) = \bar{a} \bar{b} + a b.$$

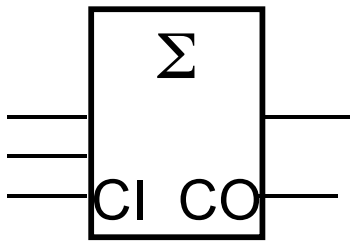
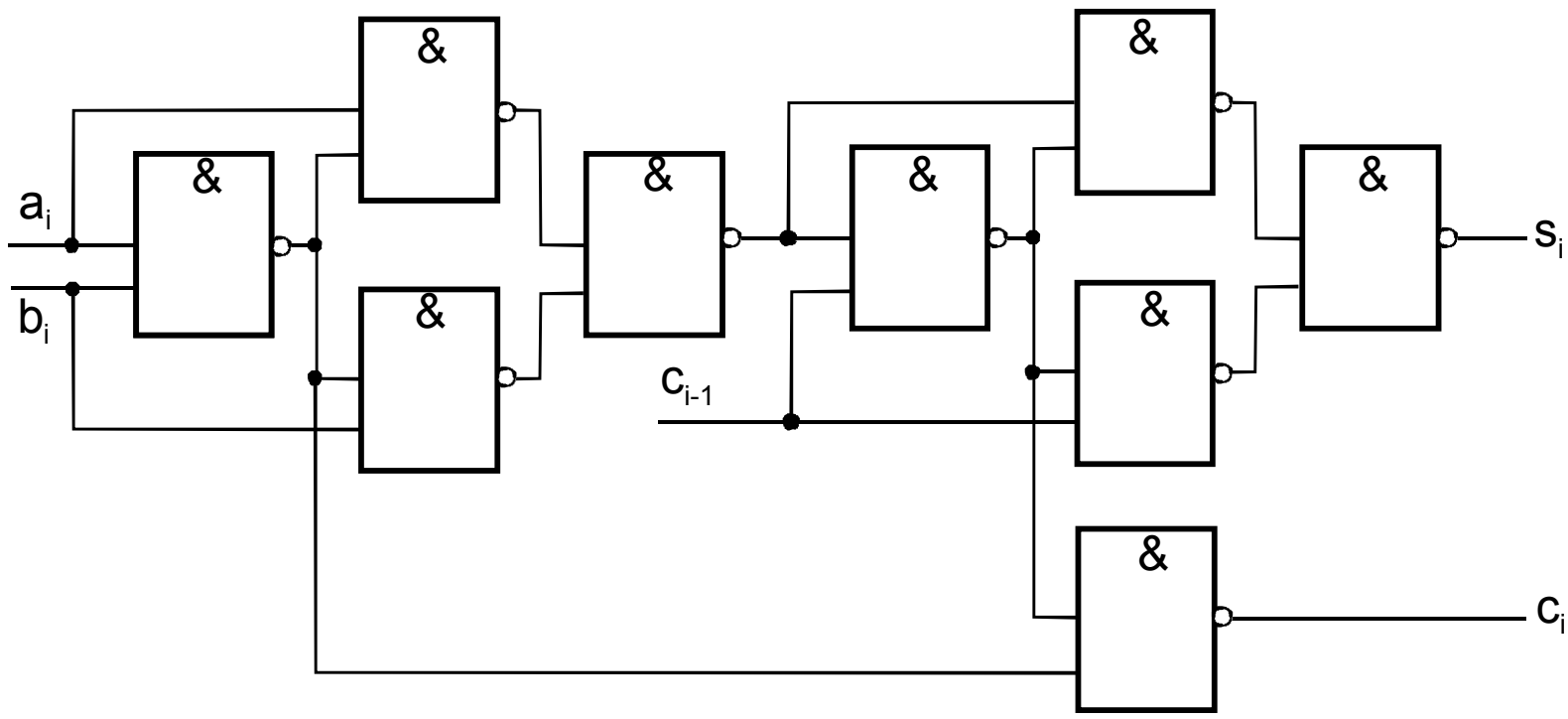
Še enostavnejša oblika za vsoto pa je:

$$s_i = [a_i \oplus b_i] \oplus c_{i-1}$$

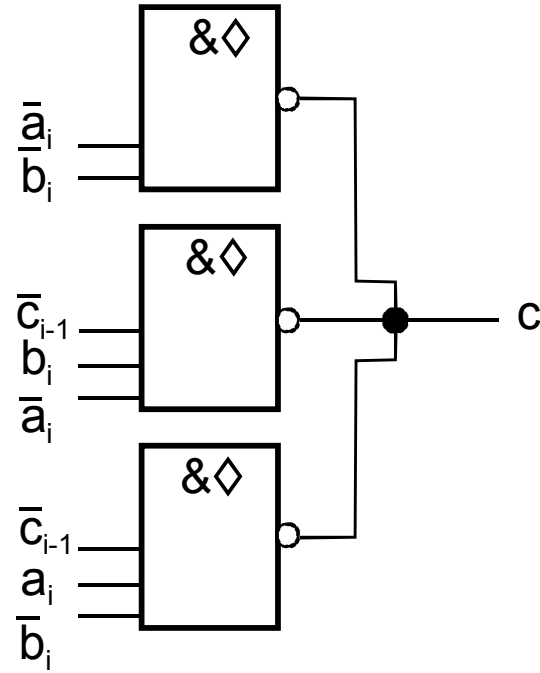
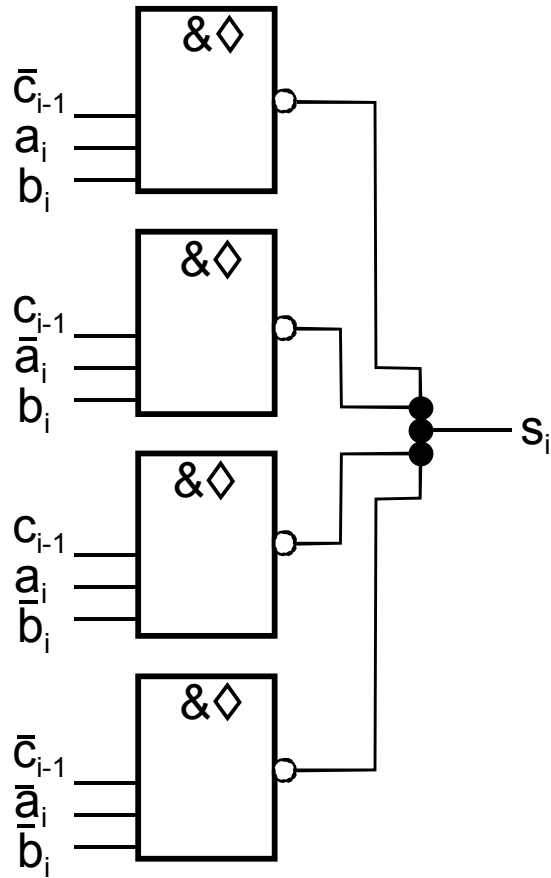
Prenos c_i pa dobimo tako:

$$c_i = a_i b_i (\bar{c}_{i-1} + c_{i-1}) + (a_i \bar{b}_i + \bar{a}_i b_i) c_{i-1}$$

$$c_i = a_i b_i + (a_i \oplus b_i) c_{i-1}$$



Posebne izvedbe enomestne seštevalne enote



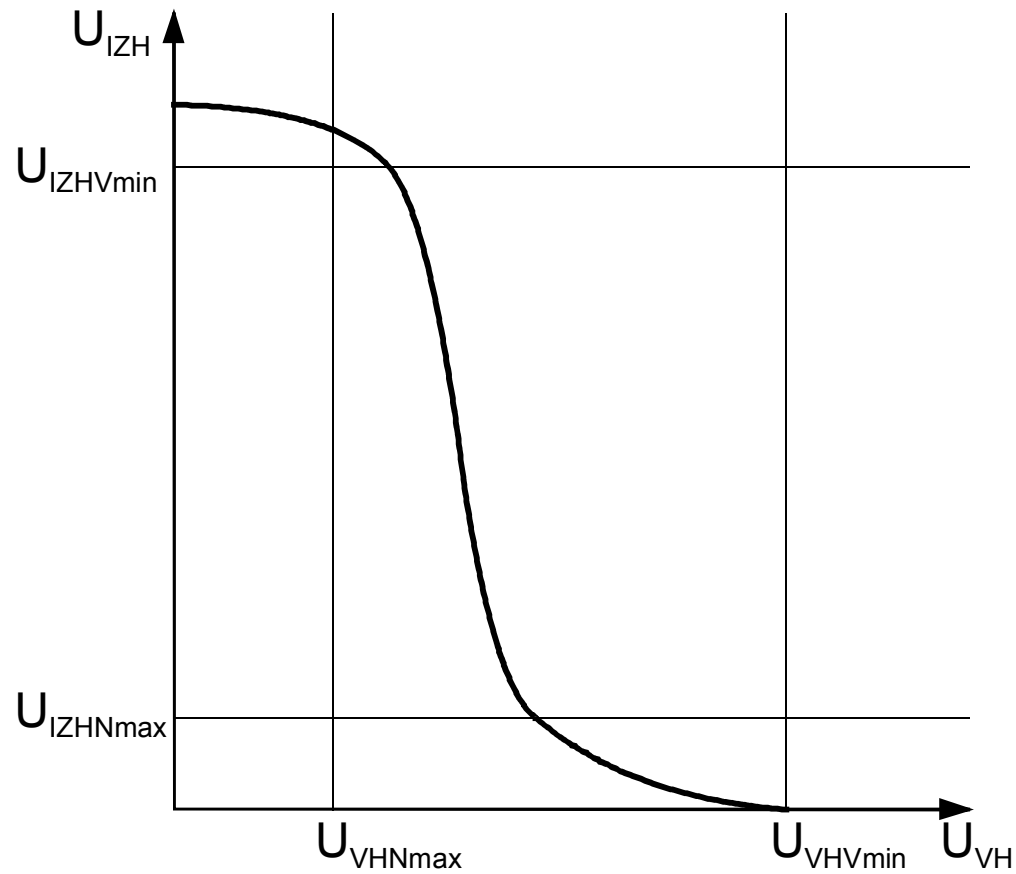
Šumne meje in obremenilni faktorji

Pozitivna logika: visok nivo \equiv logična "1", nizek nivo \equiv logična "0"

Negativna logika: visok nivo \equiv logična "0", nizek nivo \equiv logična "1"

Logična stanja zato popisujemo z vrsto napetostnih nivojev

Napetostni nivo	Opis napetostnega nivoja	Tipična vrednost
U_{VHVmax}	Maksimalna vhodna napetost logične "1"	5,0V
U_{VHVmin}	Minimalna vhodna napetost logične "1"	3,5V
U_{VHNmax}	Maksimalna vhodna napetost logične "0"	1,0V
U_{VHNmin}	Minimalna vhodna napetost logične "0"	0,0V
$U_{IZHVmax}$	Maksimalna izhodna napetost logične "1"	5,0V
$U_{IZHVmin}$	Minimalna izhodna napetost logične "1"	4,9V
$U_{IZHNmax}$	Maksimalna izhodna napetost logične "0"	0,1V
$U_{IZHNmin}$	Minimalna izhodna napetost logične "0"	0,0V

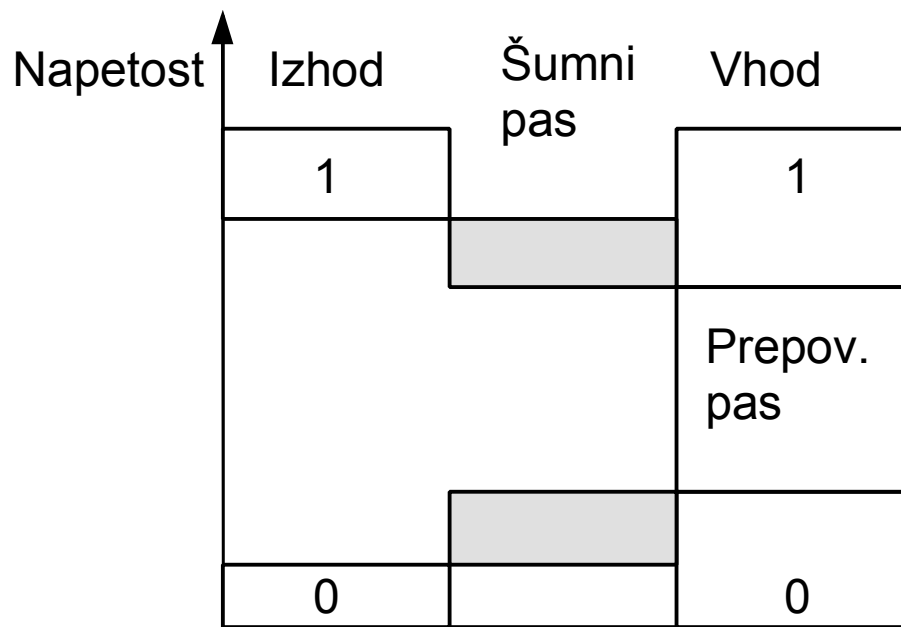


Šumne meje(noise margin)

Te določajo koliko šuma je še lahko prisotnega, preden je nivo napetosti prepoznan za napačnega.

Ločimo dva šumna pasova:

- šumni pas pri visokem nivoju napetosti
- šumni pas pri nizkem nivoju napetosti



$$\check{S}P_{NN} = U_{VHNmax} - U_{IZHNmax}$$

$$\check{S}P_{VN} = U_{IZHVmin} - U_{VHVmin}$$

Večja kot sta šumna pasova, v bolj električno "onesnaženem" okolju lahko vezje obratuje.

Obremenilni faktorji (Fan - out):

Obremenilni faktor izhoda je število, ki pove koliko vhodov lahko neko vezje napaja iz svojega izhoda, ne da bi se pri tem bistveno poslabšala šumna meja.

Pri vezjih, ki imajo uporovni značaj, kot na primer TTL izvedbe, se obremenilni faktor določi kot razmerje izhodnega toka, ki ga je vezje ob zgornjem pogoju sposobno dati, proti toku, ki je za preklop potreben na vhodu v vezje.

Za družino 74ALS veljajo naslednji podatki:

I_{VHVmax}	Maksimalni vhodni tok za "1"	$20 \mu A$
I_{VHNmax}	Maksimalni vhodni tok za "0"	$-100 \mu A$
I_{ZHVmax}	Maksimalni tok izhoda pri "1"	$-400 \mu A$
I_{ZHNmax}	Maksimalni tok izhoda pri "0"	$8mA$

Iz teh podatkov dobimo dva obremenilna faktorja:

$$\text{Obremenilni faktor visokega nivoja: } \frac{I_{ZHVmax}}{I_{VHVmax}} = \frac{400\mu A}{20\mu A} = 20$$

$$\text{Obremenilni faktor nizkega nivoja: } \frac{I_{ZHNmax}}{I_{VHNmax}} = \frac{8mA}{100\mu A} = 80$$

Upoštevati je seveda potrebno manjšega!!

Predpostavimo preklon "1" na "0"

N tranzistor ima takrat zelo nizko upornost in izhodno napetost lahko določimo iz enačbe:

$$U_{IZH} = U_{DD} e^{-t/R_N C_G}$$

Minimalna napetost logične "1" je 3,5 V; maksimalna napetost logične "0" pa 1V

$$U_{DD} = 5V$$

$$R_N = 100 \text{ ohmov}$$

$$C_G = 100 \text{ pF}$$

Čas potreben za upad napetosti iz 3,5V na 1,0V je:

$$t = -100 \times 100 \times 10^{-12} \times \ln \frac{1,0}{5,0} + 100 \times 100 \times 10^{-12} \times \ln \frac{3,5}{5,0} = 12,5 \text{ ns}$$

Če "poganjamo" dvoje vrat hkrati, se kapacitivnost podvoji in z njo se podvoji tudi preklonni čas.

Zato se je potrebno zavedati, da čeprav CMOS lahko napaja praktično neomejeno število vhodov pri fiksnem nivoju izhodne napetosti, je obremenilni faktor omejen z zahtevano hitrostjo preklonov v vezju.

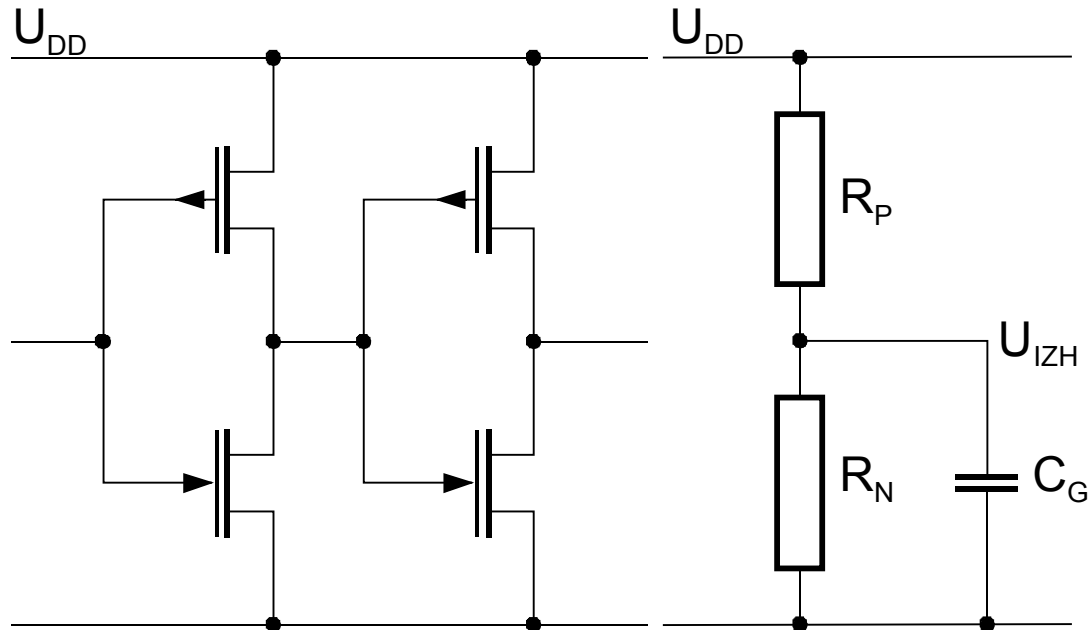
Vhodi CMOS vezij ne zahtevajo toka, ker ni galvanske zveze med vrati (Gate) in izvorom ali ponorom toka v tranzistorju.

Zaradi tega bi pričakovali zelo visok obremenilni faktor, vendar je ta le navidezen.

Vrata in izvor (substrat) toka v tranzistorju tvorita kondenzator, zato je potreben določen čas, da se ta kondenzator napolni oziroma sprazni.

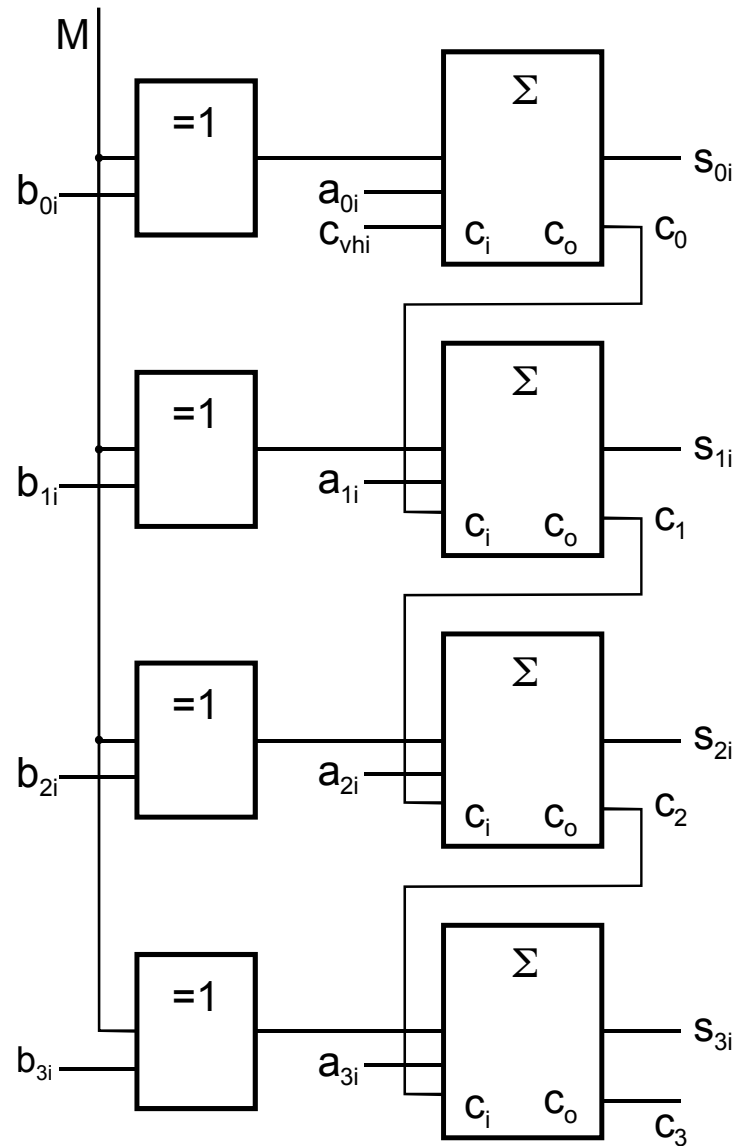
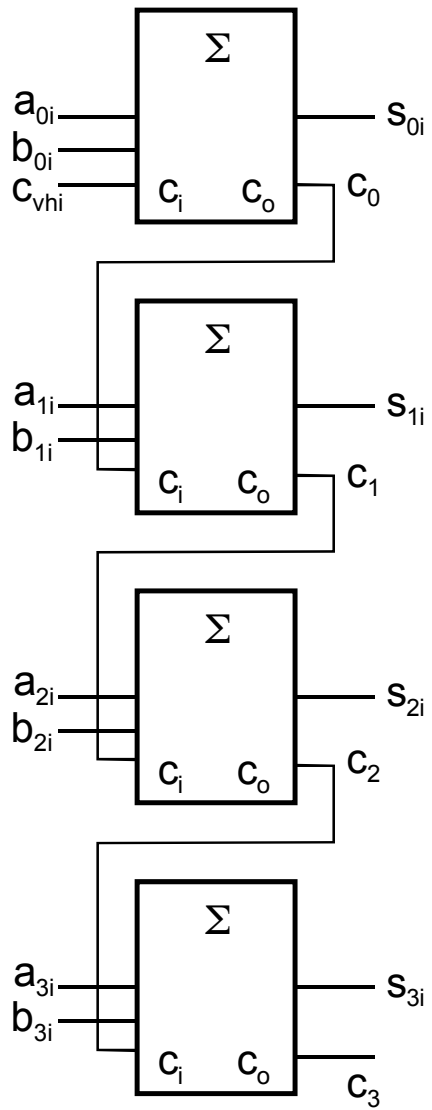
Dodatno k temu ima kapacitivnost tudi povezava me s posameznimi vrati, ki je v zelo hitrih vezjih lahko celo dominantna.

Zato karakteristike teh povezav lahko določimo šele, ko definiramo in načrtamo celotno vezje.



5. 2 Večmestne seštevalno-odštevalne enote

5. 2. 1 Serijski seštevalniki – SS[n] (Ripple Adders)



Osnovni zakasnilni čas je: $2n + 4$, AND - OR – INVERT pa da zakasnilni čas $2n$

5. 2. 2 Paralelni seštevalniki – PS[n] (carry-lookahead adders)

$$s_i = (a_i \oplus b_i) \oplus c_{i-1} \quad c_i = ab_i + c_{i-1}(a_i \oplus b_i)$$

1. G_i naj bo ena vedno, kadar je ena prenos c_i , ne oziraje se na prenos z nižjega mesta c_{i-1} .

Ta funkcija je, izražena z vhodi v seštevalnik, konjunkcija a_i in b_i ; torej $G_i = a_i b_i$.

2. P_i naj bo ena, kadar je prenos c_i enak ena ob pogoju, da je tudi prenos z nižjega mesta enak ena.

Ta funkcija je, izražena z vhodi v seštevalnik, XOR vhodov a_i in b_i ; torej $P_i = a_i \oplus b_i$.

$$s_i = (a_i \oplus b_i) \oplus c_{i-1} = P_i \oplus c_{i-1} \quad c_i = ab_i + (a_i \oplus b_i) c_{i-1} = G_i + P_i c_{i-1}$$

$$c_{i-1} = c_{vh}$$

$$c_0 = G_0 + c_{vh} P_0$$

$$c_1 = G_1 + c_0 P_1$$

$$c_2 = G_2 + c_1 P_2$$

$$c_3 = G_3 + c_2 P_3$$

$$s_0 = P_0 \oplus c_{vh}$$

$$s_1 = P_1 \oplus c_0$$

$$s_2 = P_2 \oplus c_1$$

$$s_3 = P_3 \oplus c_2$$

$$c_1 = G_1 + c_0 P_1 = G_1 + (G_0 + c_{vh} P_0) P_1 =$$

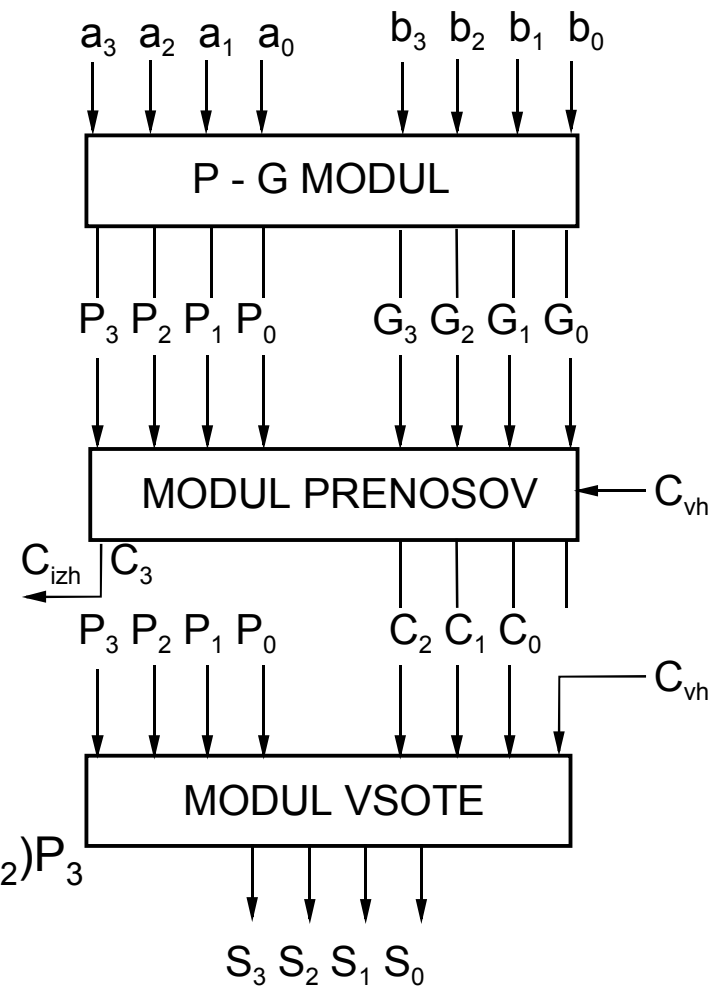
$$= G_1 + G_0 P_1 + c_{vh} P_0 P_1$$

$$c_2 = G_2 + c_1 P_2 = G_2 + (G_1 + G_0 P_1 + c_{vh} P_0 P_1) P_2 =$$

$$= G_2 + G_1 P_2 + G_0 P_1 P_2 + c_{vh} P_0 P_1 P_2$$

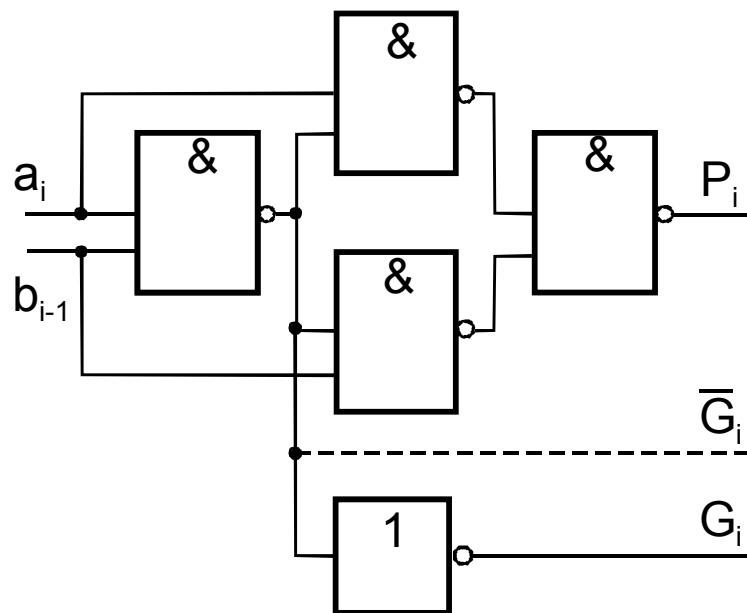
$$c_3 = G_3 + C_2 P_3 = G_3 + (G_2 + G_1 P_2 + G_0 P_1 P_2 + c_{vh} P_0 P_1 P_2) P_3$$

$$c_3 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + c_{vh} P_0 P_1 P_2 P_3$$



P-G Modul

$$P_i = a_i \oplus b_i; G_i = a_i b_i \quad 0 \leq i \leq 3$$



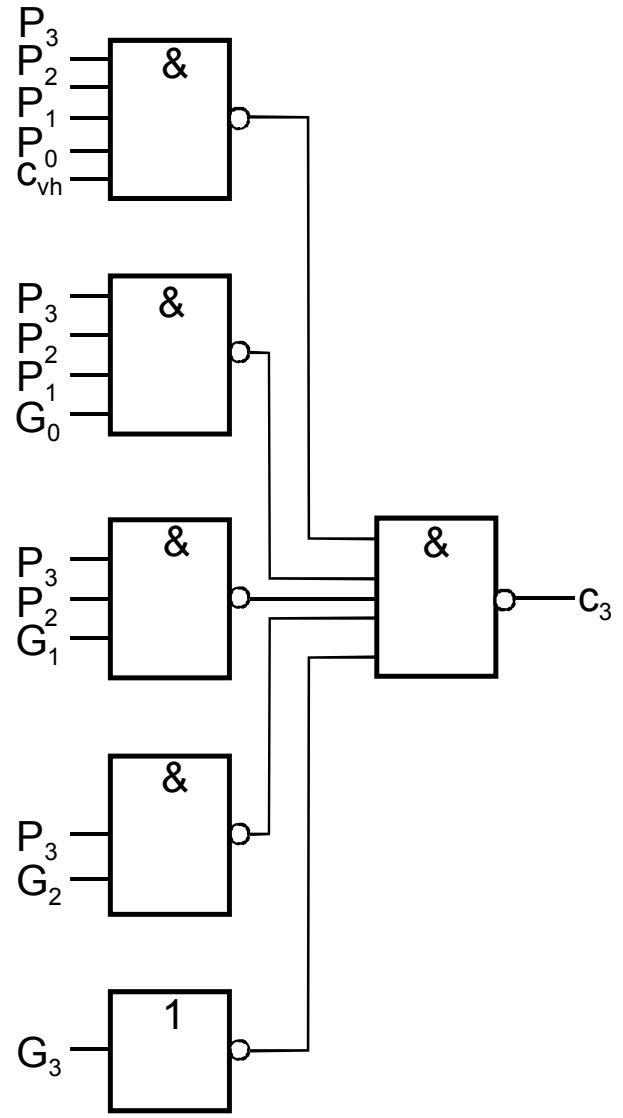
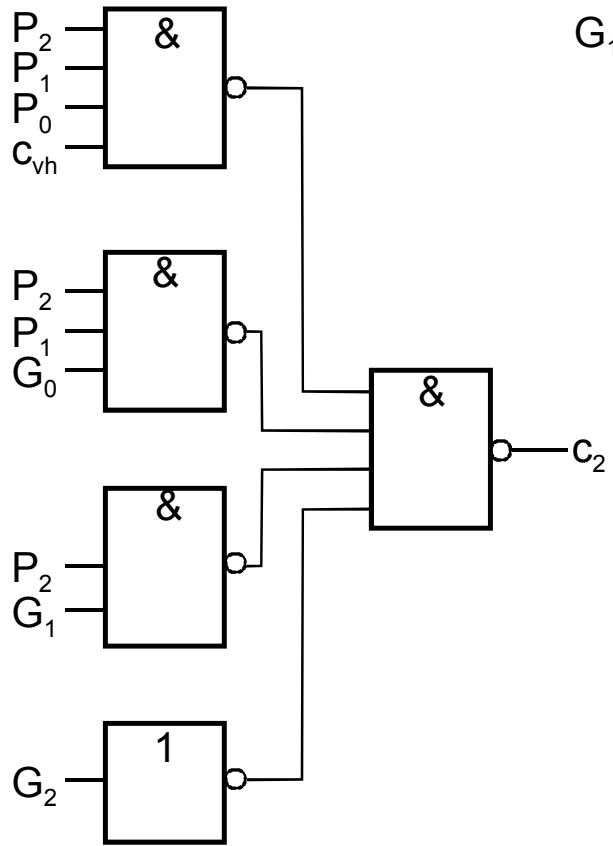
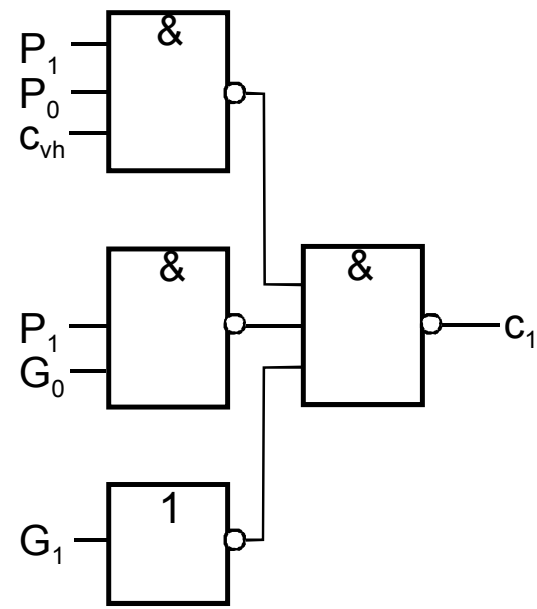
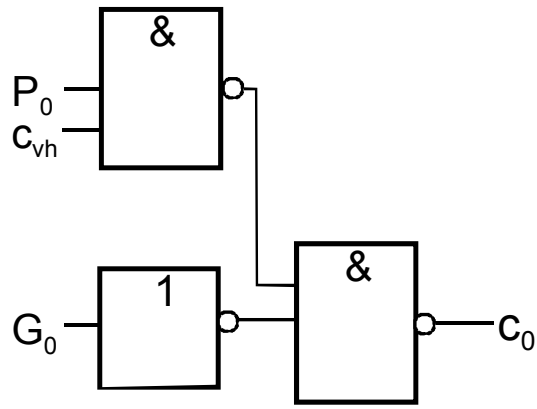
Zakasnilni čas:

VHOD/IZHOD	a_i	b_i
P_i	3	3
G_i	2	2

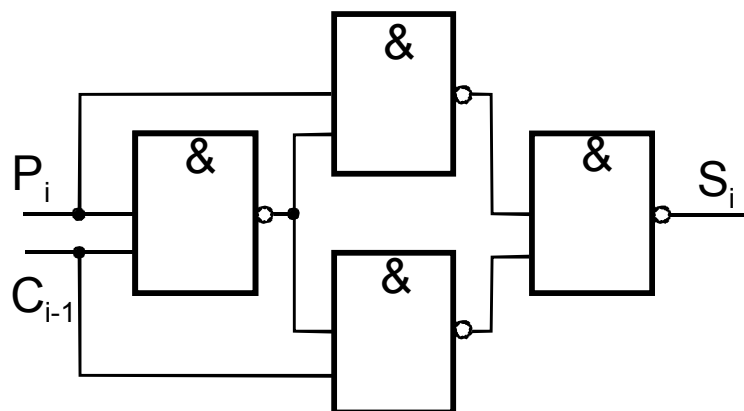
Modul prenosov

Grupa štirih izhodov (c_0, c_1, c_2, c_3).

Zanj potrebujemo 18 NAND elementov, vezje pa ima zakasnitev 2 enot.



Modul vsote



Funkcija	Število enot	Število NAND na enoto	Celotno število NAND	Zakasnilni čas
P-G Modul	4	5	20	3
Prenos	1	18	18	2
Vsota	4	4	16	3
Skupno število NAND elementov 54				
Skupen zakasnilni čas 8 enot				

$$P_i = a_i \oplus b_i \quad G_i = a_i b_i$$

$$s_i = P_i \oplus c_{i-1}$$

Splošni člen v seštevalniku bo:

$$c_k = G_k + c_{k-1} P_k, \quad c_0 = G_0 + c_{vh} P_0$$

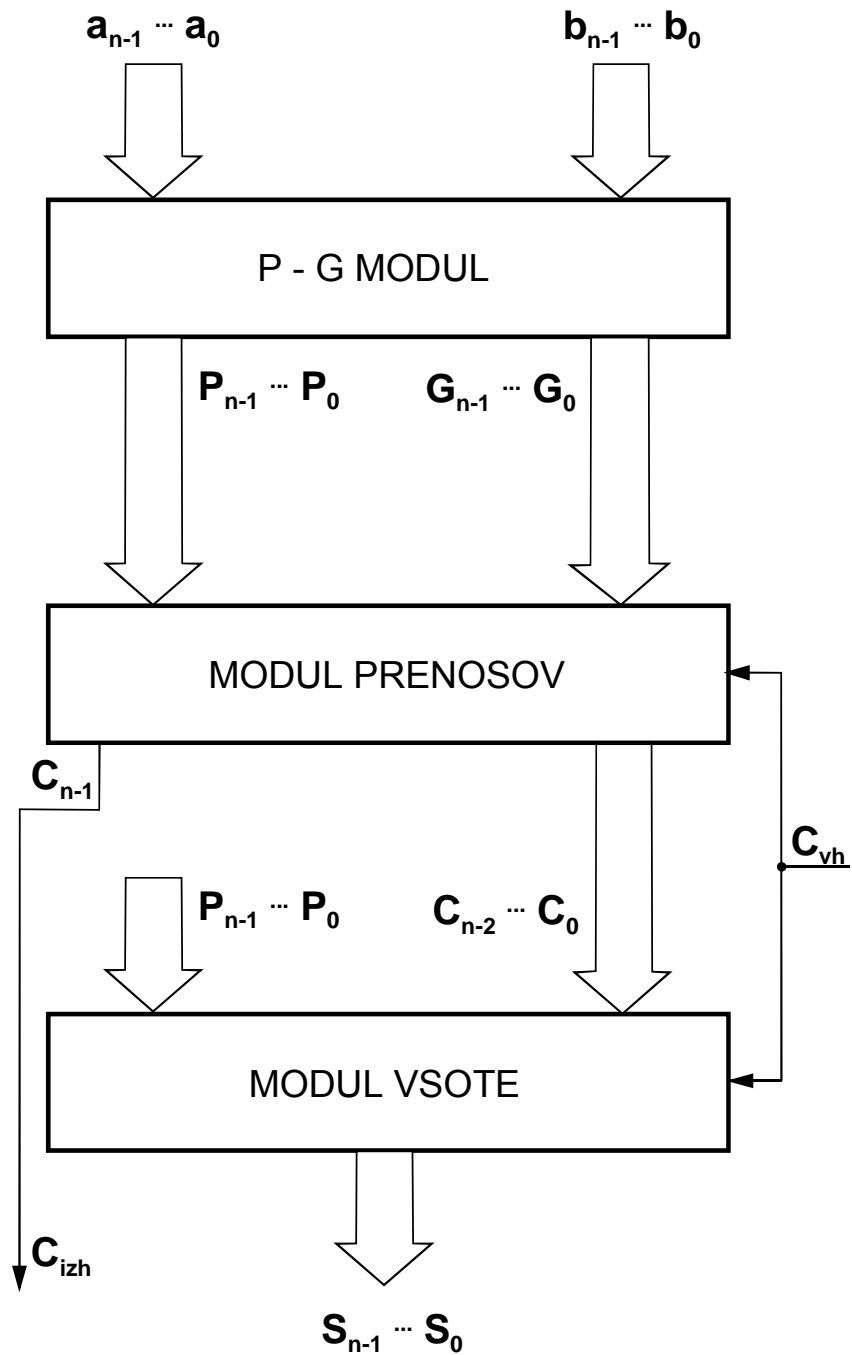
Razširjene enačbe dobijo obliko:

$$c_k = c_{vh} \prod_{i=0}^k P_i + \sum_{j=0}^{k-1} \left\{ G_j \left[\prod_{i=j+1}^k P_i \right] \right\} + G_k$$

$$c_3 = c_{vh} \prod_{i=0}^3 P_i + \sum_{j=0}^2 \left\{ G_j \left[\prod_{i=j+1}^3 P_i \right] \right\} + G_3$$

Torej:
$$c_3 = c_{vh} P_0 P_1 P_2 P_3 + G_0 P_1 P_2 P_3 + G_1 P_2 P_3 + G_2 P_3 + G_3$$

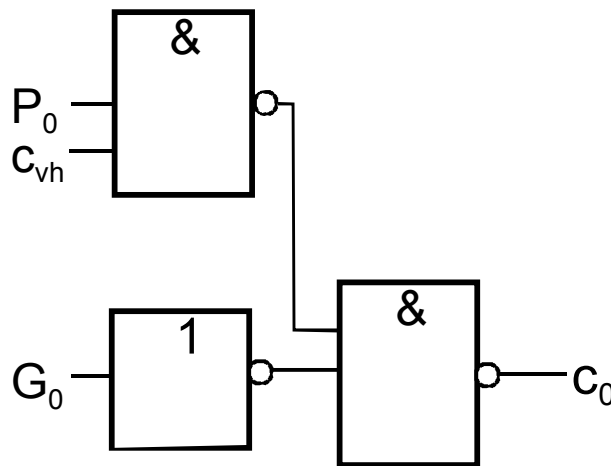
Struktura splošnega seštevalnika je tako:



$$c_k = G_k + c_{k-1} P_k$$

in

$$\begin{aligned} c_{k+1} &= G_{k+1} + c_k P_{k+1} = \\ &= G_{k+1} + [G_k + c_{k-1} P_k] P_{k+1} \end{aligned}$$



Prenos	Število NAND elementov
C_0	3
C_1	4
...	
...	
...	
C_{n-1}	$(n-1)+3 = n+2$

Skupno število potrebnih elementov za vse prenose seštevalnika (N_{EP}) je potem:

$$N_{EP} = \sum_{j=0}^{n-1} (j+3)$$

$$N_{EP} = \sum_{j=3}^{n+2} j = \sum_{j=0}^{n+2} j - \sum_{j=0}^2 j$$

$$\sum_{j=0}^k j = \frac{k(k+1)}{2}$$

$$N_{EP} = \sum_{j=0}^{n+2} j - \sum_{j=0}^2 j = \frac{(n+2)(n+3)}{2} - \frac{2(3)}{2} = \frac{n^2 + 5n}{2} = \frac{n(n+5)}{2}$$

Funkcionalna enota	Število enot	Štev. NAND elem. na enoto	Cel. število NAND elem.	Zakasnilni čas
P - G Modul	n	5	5n	3
Modul prenosov	1	$\frac{n(n+5)}{2}$	$\frac{n(n+5)}{2}$	2
Modul vsote	n	4	4n	3

Skupno število elementov NAND: $\frac{n(n+23)}{2}$

Celotni zakasnilni čas 8 enot.

V gornji tabeli smo predpostavili, da imamo v P-G modulu standardno trinivojsko NAND izvedbo brez invertorja, dvonivojsko NAND izvedbo v modulu prenosov in trinivojsko NAND izvedbo v modulu vsot.

Primerjava s serijsko izvedbo:

Paralelna izvedba: $N_{EPS} = \frac{n(n+23)}{2}$; $t_z = 8$ enot

Serijska izvedba: $N_{ESS} = 9n$; $t_z = 2n + 4$

Pri paralelni izvedbi lahko prihranimo "n" NAND elementov v modulu prenosov, če generiramo \bar{G} v P-G modulu.

Pri serijski vezavi ne pomagajo kaj prida alternativne rešitve, razen če gre za majhno število bitov.

Pri paralelni vezavi pa nam AND-OR-INVERT vezava še dodatno zniža N_{EPS} na:

$$N_{EPS} = \frac{n(n+21)}{2}$$

16 bitov: $N_{EPS} = 2N_{ESS}$ t_{zaks} je približno 4,5 kratni t_{zakp}

64 bitov: $N_{EPS} = 5N_{ESS}$ t_{zaks} je približno 17 kratni t_{zakp}

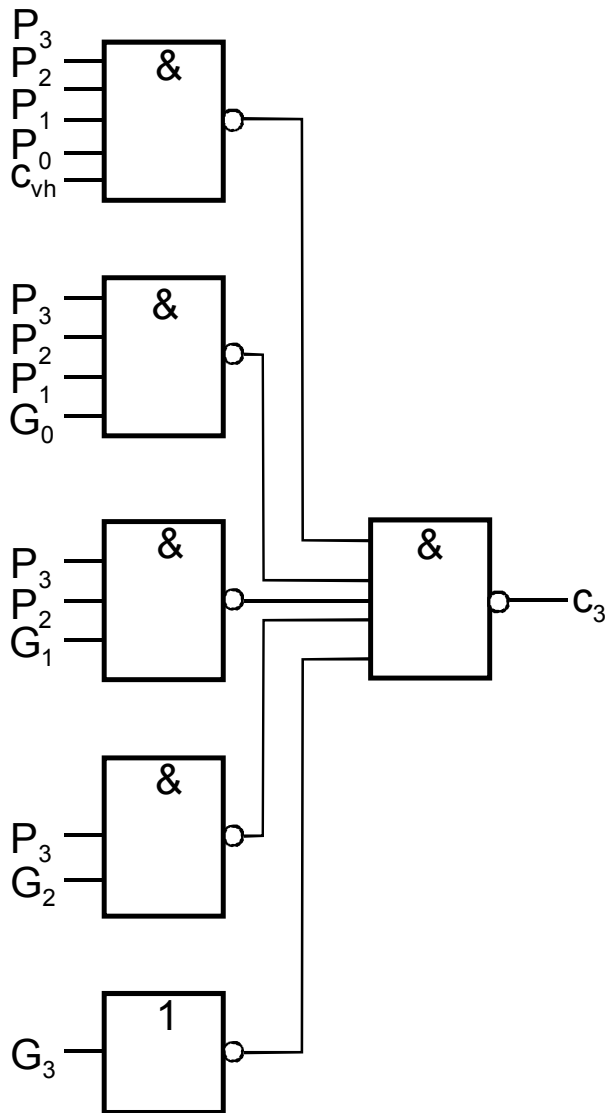
S teoretičnega stališča je za hitre računalnike najbolj primerna paralelna izvedba.

Toda pri praktični izvedbi lahko naletimo na težave pri velikem številu bitov ($n > 16$).

Problem predstavljajo elementi z velikim številom vhodov in zmožnost izhoda, da napaja veliko število vhodov.

Za vse tri dele seštevalnika poiščimo sedaj izraze za število NAND elementov, ki potrebujejo po “j” vhodov.

P – G modul in modul vsote potrebujeta “8n”; (4+4)n dvo-vhodnih in “n” eno-vhodnih NAND elementov.



$$C_3 = C_{vh} P_0 P_1 P_2 P_3 + G_0 P_1 P_2 P_3 + G_1 P_2 P_3 + G_2 P_3 + G_3$$

Če te ugotovitve razvrstimo vidimo, da “n” enačb prenosov zahteva skupaj:

n	1 vhodnih elementov
n+1	2 vhodnih elementov
n	3 vhodnih elementov
n-1	4 vhodnih elementov
...	
n+3-j	j vhodnih elementov
...	
3	n vhodne elemente
2	(n+1) vhodna elementa

Modul prenosov torej zahteva (n+3-j) NAND elementov z “j” vhodi

Tabelarični pregled:

P - G	Prenos	Vsota	Skupaj
n	n	0	$2n - 1$ vhodnih NAND
4n	n+1	4n	$9n+1 - 2$ vhodnih NAND
0	n+3-j	0	$n+3-j - j$ vhodnih NAND

$$3 \leq j \leq n+1$$

Naslednji problem predstavlja zmožnost posameznega izhoda.

Oba vhoda a_i, b_i zahtevata po dva obremenilna faktorja.

c_{vh} mora napajati po en vhod za vsako enačbo prenosa in dva za vsoto $(n+2)$.

Če tega ni zmožen dodamo inverterje; s tem ne pokvarimo zakasnilnega časa.

Vsak izhod modula prenosov napaja le dva vhoda v modulu vsote – ni kritično.

Najbolj neugodne razmere pa se pojavijo v $P - G$ modulu.

Če pazljivo pogledamo enačbe prenosov, bomo ugotovili, da se funkcija G_j pojavlja $(n - j)$ krat.

Uvidimo pa lahko tudi, da se funkcija P_j pojavlja $(j + 1)(n - j)$ krat.

$G_j \rightarrow (n - j)$ krat

$P_j \rightarrow (j + 1)(n - j)$ krat $0 \leq j \leq (n-1)$

P_j je uporabljena natančno $(j+1)$ krat v enačbah prenosov od c_j do c_{n-1} .

Če je "n" liho število, je maksimalno potrebno število enot obremenitve pri funkciji z indeksom $P_{(n-1)/2}$ in je enako:

$$(n + 1)^2 / 4$$

Če pa je "n" sodo število, tedaj je maksimalno potrebno število enot obremenitve pri funkcijah z indeksoma $P_{(n/2)-1}$ in $P_{n/2}$ in je enako:

$$n(n + 2) / 4$$

Vsaka od teh funkcij pa mora napajati še po dva vhoda v modulu vsote !

Skupaj je tako potrebno zagotoviti sledeča števila enot obremenitve.

n – liho število:

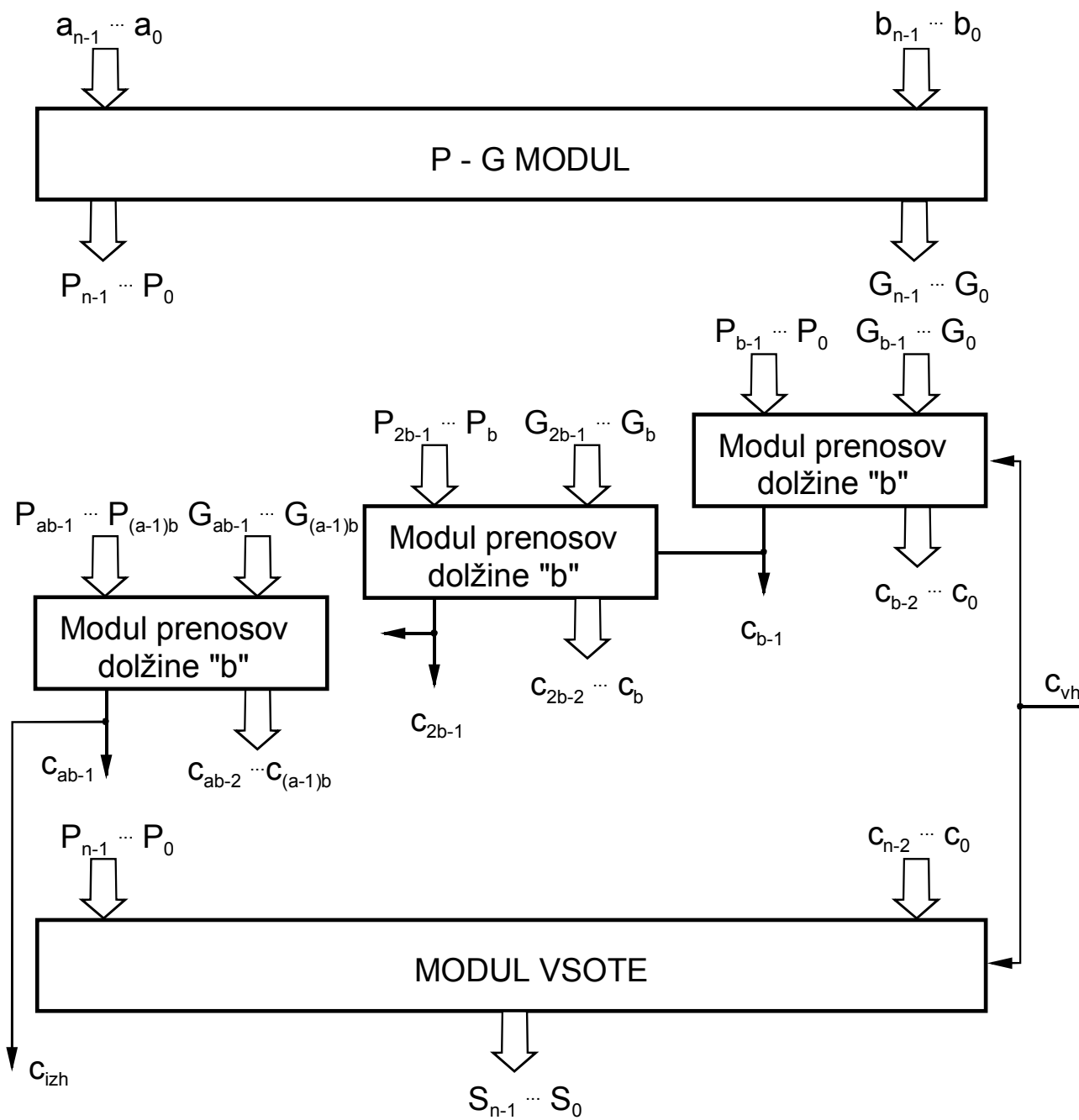
$$P_{(n-1)/2}; \quad O_{\max} = \frac{(n + 1)^2}{4} + 2$$

n – sodo število:

$$P_{(n/2)-1}; \quad O_{\max} = \frac{n(n + 2)}{4} + 2$$

$$P_{(n/2)}; \quad O_{\max} = \frac{n(n + 2)}{4} + 2$$

5. 2. 3 Serijsko paralelni seštevalnik - SPS n: a,b (Ripple-Carry-Lookahead Adders)



Takšna organizacija seštevalnika zmanjša število potrebnih elementov približno za faktor **a** in poveča zakasnilni čas za **2(a-1)** enot.

Nove enačbe za število elementov in zakasnilni čas so sedaj:

$$N_{\text{EPS}(n:a,b)} = \frac{n(b+23)}{2}$$

$$t \text{ zakasnitve SPS } (n:a,b) = 6 + 2a$$

Maksimalna potrebna zmožnost izhoda je tudi znižana na vrednost:

$$\frac{b(b+2)}{4} + 2 \quad \text{za sode } n$$

in

$$\frac{(b+1)^2}{4} + 2 \quad \text{za lihe } n$$

Za primer 16 bitnega SPS (16:4,4) je maksimalno število izhodov znižano iz 74 na 8.

Potrebno število vhodov je prav tako zmanjšano iz (n+1) na (b+1); to je za SPS (16:4,4) od 17 na 5.

Funkcionalna enota	Število potreb. enot	Štev.: NAND elem.na enoto	Skup. število elementov	Zakasnilni čas
P - G	n	5	5n	3
Modul preno. dolžine "b"	$a = \frac{n}{b}$	$\frac{b(b+5)}{2}$	$\frac{n(b+5)}{2}$	2a
Modul vsote	n	4	4n	3

Omejitve: $ab = n$

$$\text{Skupno število elementov: } 9n + \frac{n(b+5)}{2} = \frac{n(b+23)}{2}$$

$$\text{Skupen zakasnilni čas: } 6 + 2a = 6 + \frac{2n}{b}$$

$$O_{\max} = \frac{b(b+2)}{4} + 2;$$

sodi

$$O_{\max} = \frac{(b+1)^2}{4} + 2;$$

lihi

Enačbe prenosov dobimo s sukcesivnim vstavljanjem prenosov v posameznih modulih prenosov dolžine "b".

V primeru 16 bitnega seštevalnika imamo sledeče prenose:

$$c_0 = G_0 + c_{vh} P_0$$

$$c_1 = G_1 + G_0 P_1 + c_{vh} P_0 P_1$$

$$c_2 = G_2 + G_1 P_2 + G_0 P_1 P_2 + c_{vh} P_0 P_1 P_2$$

$$c_3 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3 + c_{vh} P_0 P_1 P_2 P_3$$

$$c_4 = G_4 + c_3 P_4$$

$$c_5 = G_5 + G_4 P_5 + c_3 P_4 P_5$$

$$c_6 = G_6 + G_5 P_6 + G_4 P_5 P_6 + c_3 P_4 P_5 P_6$$

$$c_7 = G_7 + G_6 P_7 + G_5 P_6 P_7 + G_4 P_5 P_6 P_7 + c_3 P_4 P_5 P_6 P_7$$

$$c_8 = G_8 + c_7 P_8$$

$$c_9 = G_9 + G_8 P_9 + c_7 P_8 P_9$$

$$c_{10} = G_{10} + G_9 P_{10} + G_8 P_9 P_{10} + c_7 P_8 P_9 P_{10}$$

$$c_{11} = G_{11} + G_{10} P_{11} + G_9 P_{10} P_{11} + G_8 P_9 P_{10} P_{11} + c_7 P_8 P_9 P_{10} P_{11}$$

$$c_{12} = G_{12} + c_{11} P_{12}$$

$$c_{13} = G_{13} + G_{12} P_{13} + c_{11} P_{12} P_{13}$$

$$c_{14} = G_{14} + G_{13} P_{14} + G_{12} P_{13} P_{14} + c_{11} P_{12} P_{13} P_{14}$$

$$c_{15} = G_{15} + G_{14} P_{15} + G_{13} P_{14} P_{15} + G_{12} P_{13} P_{14} P_{15} + c_{11} P_{12} P_{13} P_{14} P_{15}$$

5. 2. 4 Paralelni seštevalniki 1. reda PS n : a,b

16-bitni seštevalnik - SPS (16: 4,4).

$$c_3 = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3 + c_{vh}P_0P_1P_2P_3$$

$$c_7 = G_7 + G_6P_7 + G_5P_6P_7 + G_4P_5P_6P_7 + c_3P_4P_5P_6P_7$$

$$c_{11} = G_{11} + G_{10}P_{11} + G_9P_{10}P_{11} + G_8P_9P_{10}P_{11} + c_7P_8P_9P_{10}P_{11}$$

$$c_{15} = G_{15} + G_{14}P_{15} + G_{13}P_{14}P_{15} + G_{12}P_{13}P_{14}P_{15} + c_{11}P_{12}P_{13}P_{14}P_{15}$$

Definirajmo sedaj nove P, G funkcije P_j^1 , G_j^1 prvega reda, ki nam bodo omogočile zmanjšati kompleksnost enačb.

$$P_0^1 = P_0P_1P_2P_3$$

$$G_0^1 = G_3 + G_2P_3 + G_1P_2P_3 + G_0P_1P_2P_3$$

$$P_1^1 = P_4P_5P_6P_7$$

$$G_1^1 = G_7 + G_6P_7 + G_5P_6P_7 + G_4P_5P_6P_7$$

$$P_2^1 = P_8P_9P_{10}P_{11}$$

$$G_2^1 = G_{11} + G_{10}P_{11} + G_9P_{10}P_{11} + G_8P_9P_{10}P_{11}$$

$$P_3^1 = P_{12}P_{13}P_{14}P_{15}$$

$$G_3^1 = G_{15} + G_{14}P_{15} + G_{13}P_{14}P_{15} + G_{12}P_{13}P_{14}P_{15}$$

$$c_3 = G_0^1 + c_{vh} P_0^1$$

$$c_7 = G_1^1 + C_3 P_1^1 = G_1^1 + G_0^1 P_1^1 + c_{vh} P_0^1 P_1^1$$

$$c_{11} = G_2^1 + C_7 P_2^1 = G_2^1 + G_1^1 P_2^1 + G_0^1 P_1^1 P_2^1 + c_{vh} P_0^1 P_1^1 P_2^1$$

$$c_{15} = G_3^1 + C_{11} P_3^1 = G_3^1 + G_2^1 P_3^1 + G_1^1 P_2^1 P_3^1 + G_0^1 P_1^1 P_3^1 + c_{vh} P_0^1 P_1^1 P_3^1$$

Popoln sistem enačb 16-bitnega paralelnega seštevalnika 1. reda je naslednji:

P-G modul:

$$P_i = a_i \oplus b_i \quad G_i = a_i b_i; \quad 0 \leq i \leq 11$$

P-G modul prvega reda:

$$P_0^1 = P_0 P_1 P_2 P_3$$

$$P_1^1 = P_4 P_5 P_6 P_7$$

$$P_2^1 = P_8 P_9 P_{10} P_{11}$$

$$P_3^1 = P_{12} P_{13} P_{14} P_{15}$$

$$G_0^1 = G_3 + G_2 P_3 + G_1 P_2 P_3 + G_0 P_1 P_2 P_3$$

$$G_1^1 = G_7 + G_6 P_7 + G_5 P_6 P_7 + G_4 P_5 P_6 P_7$$

$$G_2^1 = G_{11} + G_{10} P_{11} + G_9 P_{10} P_{11} + G_8 P_9 P_{10} P_{11}$$

$$G_3^1 = G_{15} + G_{14} P_{15} + G_{13} P_{14} P_{15} + G_{12} P_{13} P_{14} P_{15}$$

Modul prenosov 1. reda:

$$c_3 = G_0^1 + c_{vh} P_0^1$$

$$c_7 = G_1^1 + G_0^1 P_1^1 + c_{vh} P_0^1 P_1^1$$

$$c_{11} = G_2^1 + G_1^1 P_2^1 + G_0^1 P_1^1 P_2^1 + c_{vh} P_0^1 P_1^1 P_2^1$$

$$c_{15} = G_3^1 + G_2^1 P_3^1 + G_1^1 P_2^1 P_3^1 + G_0^1 P_1^1 P_2^1 P_3^1 + c_{vh} P_0^1 P_1^1 P_2^1 P_3^1$$

Modul prenosov:

$$c_0 = G_0 + c_{vh} P_0$$

$$c_1 = G_1 + G_0 P_1 + c_{vh} P_0 P_1$$

$$c_2 = G_2 + G_1 P_2 + G_0 P_1 P_2 + c_{vh} P_0 P_1 P_2$$

$$c_4 = G_4 + c_3 P_4$$

$$c_5 = G_5 + G_4 P_5 + c_3 P_4 P_5$$

$$c_6 = G_6 + G_5 P_6 + G_4 P_5 P_6 + c_3 P_4 P_5 P_6$$

$$c_8 = G_8 + c_7 P_8$$

$$c_9 = G_9 + G_8 P_9 + c_7 P_8 P_9$$

$$c_{10} = G_{10} + G_9 P_{10} + G_8 P_9 P_{10} + c_7 P_8 P_9 P_{10}$$

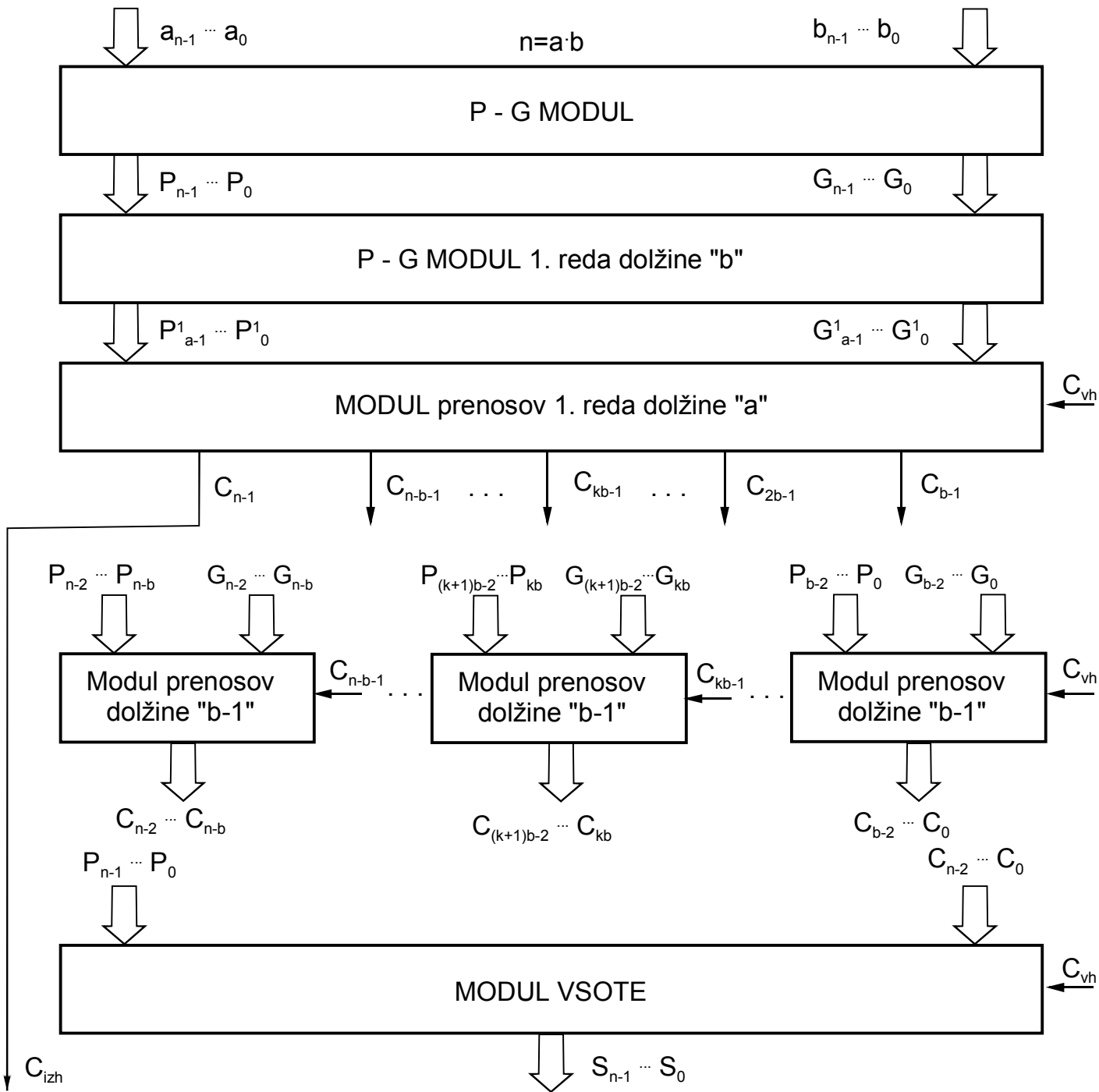
$$c_{12} = G_{12} + c_{11} P_{12}$$

$$c_{13} = G_{13} + G_{12} P_{13} + c_{11} P_{12} P_{13}$$

$$c_{14} = G_{14} + G_{13} P_{14} + G_{12} P_{13} P_{14} + c_{11} P_{12} P_{13} P_{14}$$

Modul vsot:

$$s_0 = P_0 \oplus c_{vh}; \quad s_j = P_j \oplus c_{j-1} \quad ; \quad \text{za} \quad 1 \leq j \leq n-1$$



Parameter b določa velikost grup P-G funkcij 1. reda.

Kvocien $n/b=a$ mora biti celo število in določa število enot v P-G modulu 1. reda.

Funkcija	Štev.enot	Štev. elementov na enoto	Skup. število elementov	Zakasnilni čas
P-G modul	n	5	5n	3
PG modul 1. reda	a	b+3	a(b+3)	2
Modul pre. 1. reda	1	$\frac{a(a+5)}{2}$	$\frac{a(a+5)}{2}$	2
Modul prenosov	a	$\frac{(b-1)(b+4)}{2}$	$\frac{a(b-1)(b+4)}{2}$	2
Modul vsote	n	4	4n	3

Maksimalno število vhodov se pojavlja zopet v modulih prenosov, kjer imamo grupe dveh različnih dolžin (a in b-1).

Zato bo maksimalno število vhodov nastopilo pri tistem, ki je večji $a+1$ ali b .

Če je a večji od b , bo največje število izhodnih enot potrebno v P-G modulu 1. reda

V nasprotnem primeru pa v standardnem P-G modulu.

V obeh primerih pa je število potrebnih izhodnih enot odvisno tudi od tega, kakšni števili sta a in b - sodi ali lihi.

Za te štiri kombinacije dobimo naslednja maksimalna števila potrebnih izhodnih enot:

$a > b$

$$O_{\max} = \frac{a(a+2)}{4}; \quad \text{za sode "a"}$$

$$O_{\max} = \frac{(a+1)^2}{4}; \quad \text{za lihe "a"}$$

$a \leq b$

$$O_{\max} = \frac{b(b+2)}{4} + 2; \quad \text{za sode "b"}$$

$$O_{\max} = \frac{(b+1)^2}{4} + 2; \quad \text{za lihe "b"}$$

Vhodni signal C_{vh} mora napajati " a " vhodov v modulu prenosov 1.reda in $(b-1)$ vhodov v standardnem modulu prenosov ter dva vhoda v modulu vsote; torej $a+b+1$.

Serijski SS (n)			Paralelni PS (n)			Serijsko-paralelni SPS (n:a,b)			Paralelni 1. reda PS (n:a,b)		
n	Št. el.	Zak.	n	Št. el.	Zak.	n:a,b	Št. el.	Zak.	n:a,b	Št. el.	Zak.
12	108	28	12	210	8	12:3,4	162	12	12:3,4	177	12
16	144	36	16	312	8	16:4,4	216	14	16:4,4	238	12
24	216	52	16	564	8	24:3,8	372	12	24:6,4	363	12
32	288	68	32	880	8	32:4,8	496	14	32:8,4	492	12
48	432	100	48	1704	8	48:6,8	744	18	48:8,6	756	12
64	576	132	64	2784	8	64:8,8	992	22	64:8,8	1052	12

Serijsko paralelni in paralelni seštevalniki prvega reda so primerljivi glede števila elementov, števila vhodov in izhodnih enot ter zakasnilnega časa do dolžine 32 bitov.

Za večje seštevalnike ($n > 32$) postane paralelni seštevalnik 1. reda nedvomno najboljša rešitev

Vendar tudi to izvedbo začne omejevati število vhodov in vse ostalo, če binarno besedo razširimo preko 64 bitov.

5. 2. 5. Aritmetično logična enota

Aritmetično logična enota poleg aritmetičnih opravljajo tudi logične operacije.

Teh logičnih operacij doslej nismo posebej izpostavljali, ker jih že poznamo.

Logične operacije namreč ne predstavljajo problematičnega dela enote.

Za ilustracijo vseh operacij aritmetično logične enote si bomo ogledali 4-bitno standardno aritmetično logično enoto, ki v TTL tehnologiji nosi številčno oznako 74181.

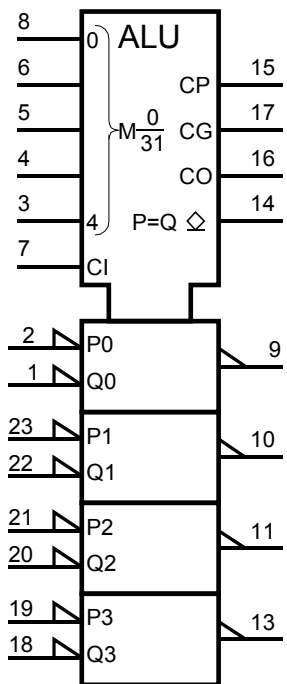
Enota 74181 vsebuje prav takšen paralelni seštevalnik, kot smo ga analizirali v tem poglavju.

Ima pa tudi izhode za P in G funkciji, kar omogoča njeno povezavo v večbitne kombinacije, kot je na primer serijsko paralelno seštevanje.

Glede na to, da je njen notranji ustroj obsežen, ga v simboličnih diagramih ne prikazujemo.

Uporabljamo namreč poseben simbol za celotno enoto.

Po IEC standardu 617-12, katerega uporaba je v dokumentiranju digitalnih sistemov že dolgo obvezna, je za aritmetično logično enoto 74181 predviden naslednji simbol:

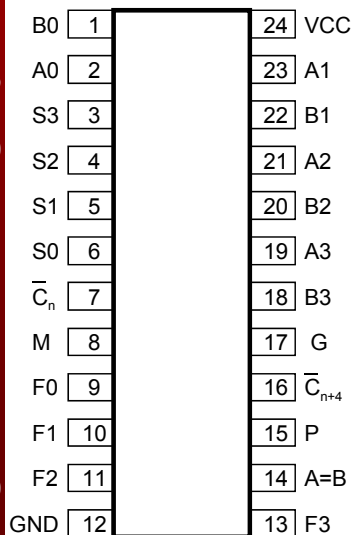


Enota lahko opravlja 16 logičnih in 16 aritmetičnih operacij.

Za izbiranje posameznih operacij potrebujemo 5 bitov

Pri tem je zgornji bit uporabljen za krmiljenje med logičnimi in aritmetičnimi operacijami.

Zgornji bit, ki odloča o tem, ali enota opravlja logične ali aritmetične operacije, je označen z M, ostali štirje pa s s_0 , s_1 , s_2 in s_3 .



Enota tako lahko deluje v dveh logičnih načinih - pozitivnem in negativnem

Pri tem pomeni, da pri pozitivnem logičnem načinu predstavlja logično enico višja napetost, logično ničlo pa nižja napetost; pri negativnem pa obratno.

Pri negativnem načinu delovanja je štiribitni izhod označen z \bar{F}_0 , \bar{F}_1 , \bar{F}_2 , \bar{F}_3 ; negirani pa sta tudi P in G funkciji in vsi vhodi, razen prenosa c_n in c_{n+4} .

S_3	S_2	S_1	S_0	Log. operacije M=1	Arit. operacije M=0
0	0	0	0	\bar{A}	A
0	0	0	1	$\overline{A+B}$	A + B
0	0	1	0	$\bar{A}B$	A + \bar{B}
0	0	1	1	logična 0	minus 1
0	1	0	0	\overline{AB}	A + $A\bar{B}$
0	1	0	1	\bar{B}	(A + B) plus $A\bar{B}$
0	1	1	0	$A \oplus B$	A minus B minus 1
0	1	1	1	$A\bar{B}$	AB minus 1
1	0	0	0	$\bar{A} + B$	A plus AB
1	0	0	1	$\overline{A \oplus B}$	A plus B
1	0	1	0	B	(A + \bar{B}) plus AB
1	0	1	1	AB	AB minus 1
1	1	0	0	logična 1	A plus A*
1	1	0	1	A + \bar{B}	(A + B) plus A
1	1	1	0	A + B	(A + \bar{B}) plus A
1	1	1	1	A	A minus 1

5. 3 Paralelni množilniki

V tem odstavku bomo vpeljali kombinacijska vezja, ki so primerna za binarno množenje.

Obdelali bomo vezje za množenje pozitivnih binarnih števil (positive integers).

5. 3. 1 Produkt dveh binarnih števil

$$A = \sum_{i=0}^{m-1} a_i 2^i$$

in

$$B = \sum_{j=0}^{n-1} b_j 2^j$$

je dan z izrazom:

$$A \cdot B = \sum_{i=0}^{m-1} a_i 2^i \sum_{j=0}^{n-1} b_j 2^j = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} a_i b_j 2^{(i+j)}$$

Števili zapišemo takole:

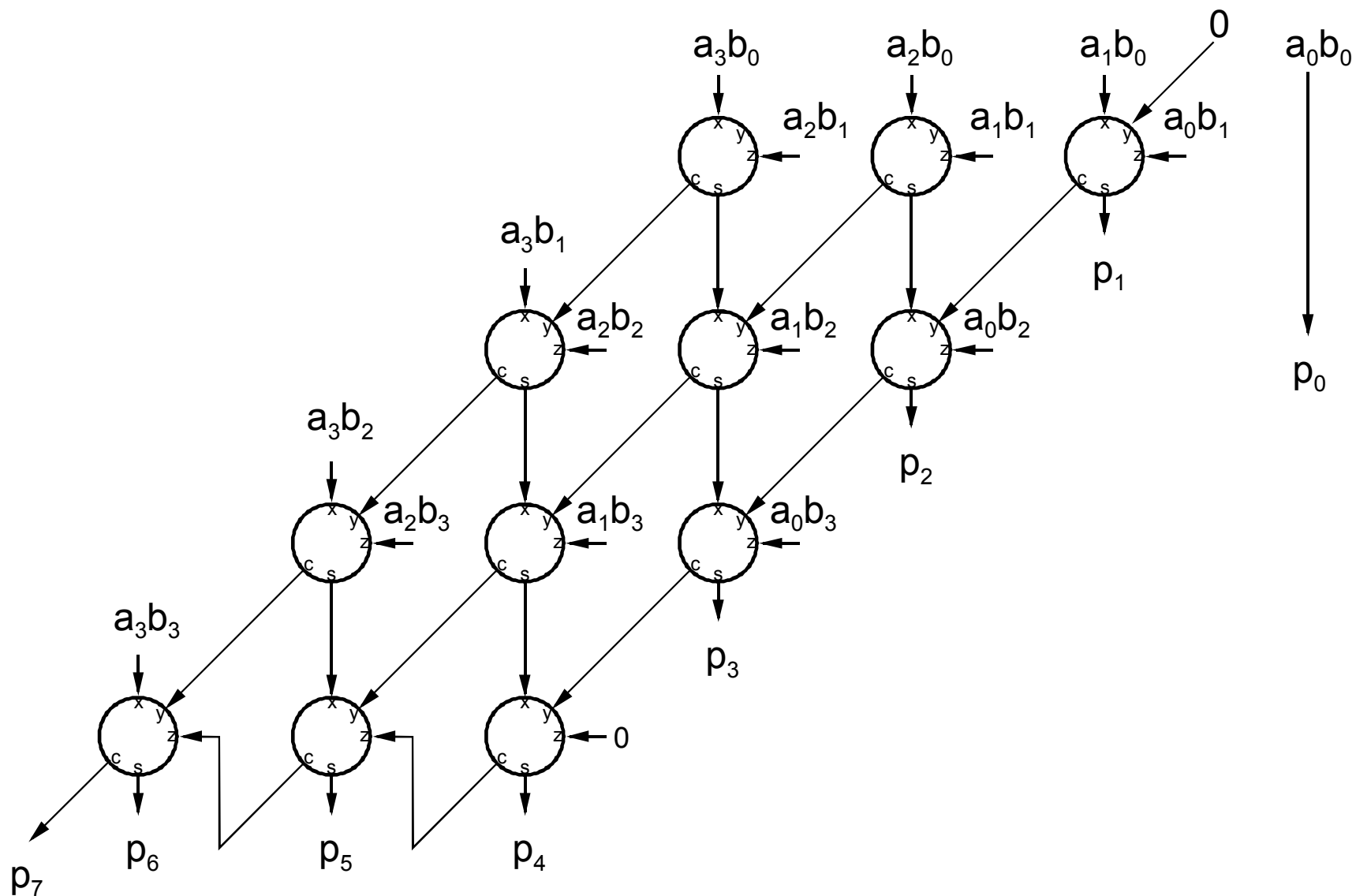
$$A = a_3 a_2 a_1 a_0 ; \quad B = b_3 b_2 b_1 b_0$$

				a_3	a_2	a_1	a_0
				b_3	b_2	b_1	b_0

				$a_3 b_0$	$a_2 b_0$	$a_1 b_0$	$a_0 b_0$
			$a_3 b_1$	$a_2 b_1$	$a_1 b_1$	$a_0 b_1$	
		$a_3 b_2$	$a_2 b_2$	$a_1 b_2$	$a_0 b_2$		
	$a_3 b_3$	$a_2 b_3$	$a_1 b_3$	$a_0 b_3$			

p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

5. 3. 2 Strukturna shema množilnika:



Čas množenja je odvisen od števila bitov!!

Posplošitev:

Vsak dodatni bit pri A zahteva novo diagonalo

Vsak dodatni bit pri B zahteva novo vrstico seštevalnikov

Pri A z "m" biti in B z "n" biti potrebujemo $(m-1)n$ seštevalnikov in mn konjunkcij

Analiza zakasnilnega časa:

Predpostavka: najhitrejši seštevalniki – 2 enoti zakasnitve

4 x 4 množilnik: 12 enot po desni diagonalni in spodnji vrstici

Dodati pa moramo še dve za tvorbo produktov; torej skupno 14 enot zakasnitve

Spolšen primer množilnika $m \times n$:

$$t_{zak} = 2(m - 1 + n - 1) + 2 = 2(m + n - 1)$$

5. 4. Dekadne aritmetične enote

Dekadne aritmetične enote so modificirane binarne enote, ki uporabljajo osnovne binarne strukture za opravljanje aritmetičnih operacij.

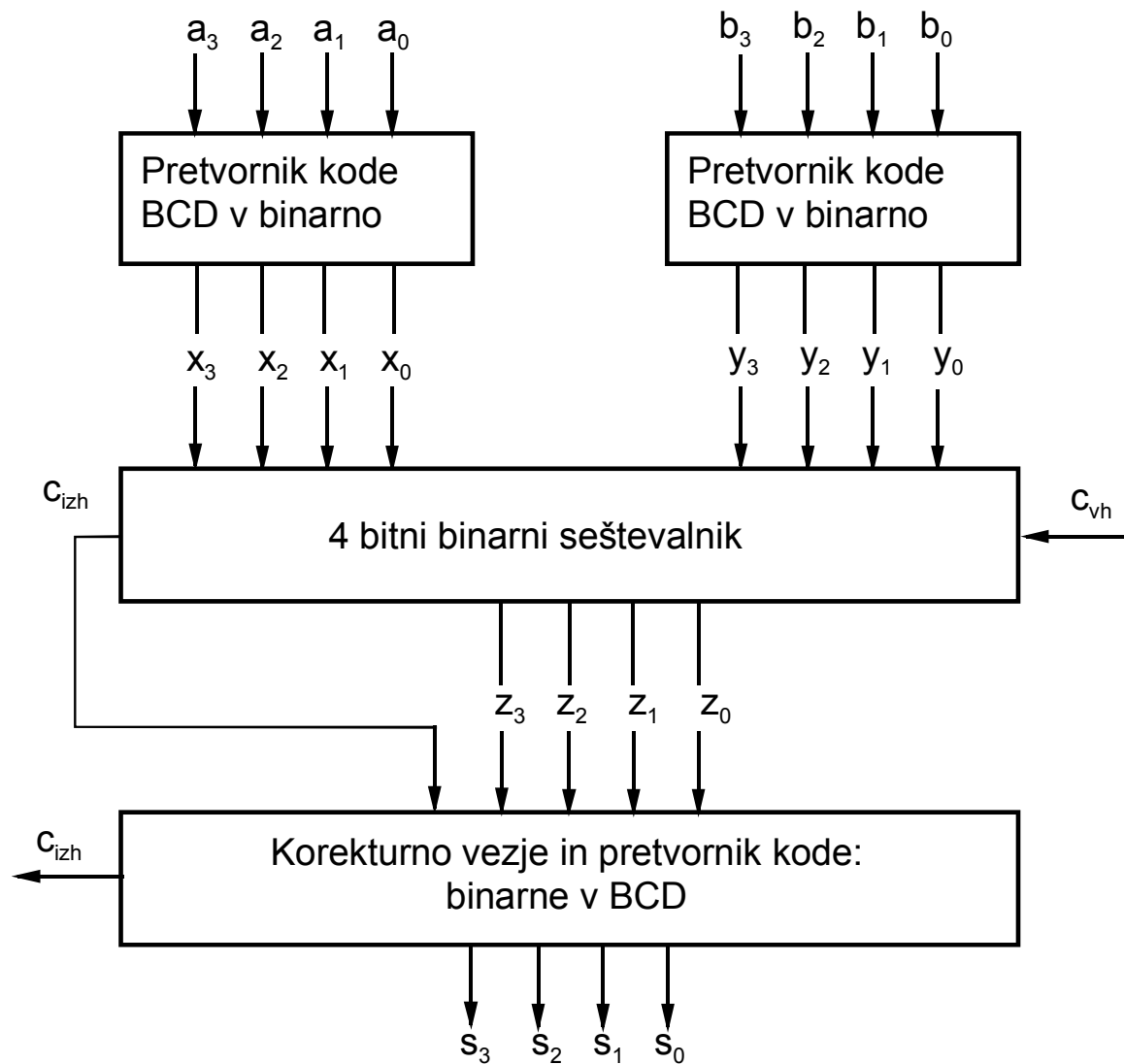
5. 4. 1 Dekadna seštevalna vezja

Vhodi in izhodi dekadnega seštevalnika so štiribitna števila za vsako dekadno mesto.

Prenosi iz nižjega na višje mesto so največkrat le enobitni.

Eno mesto za prenos zadošča za večino uporabnih področij.

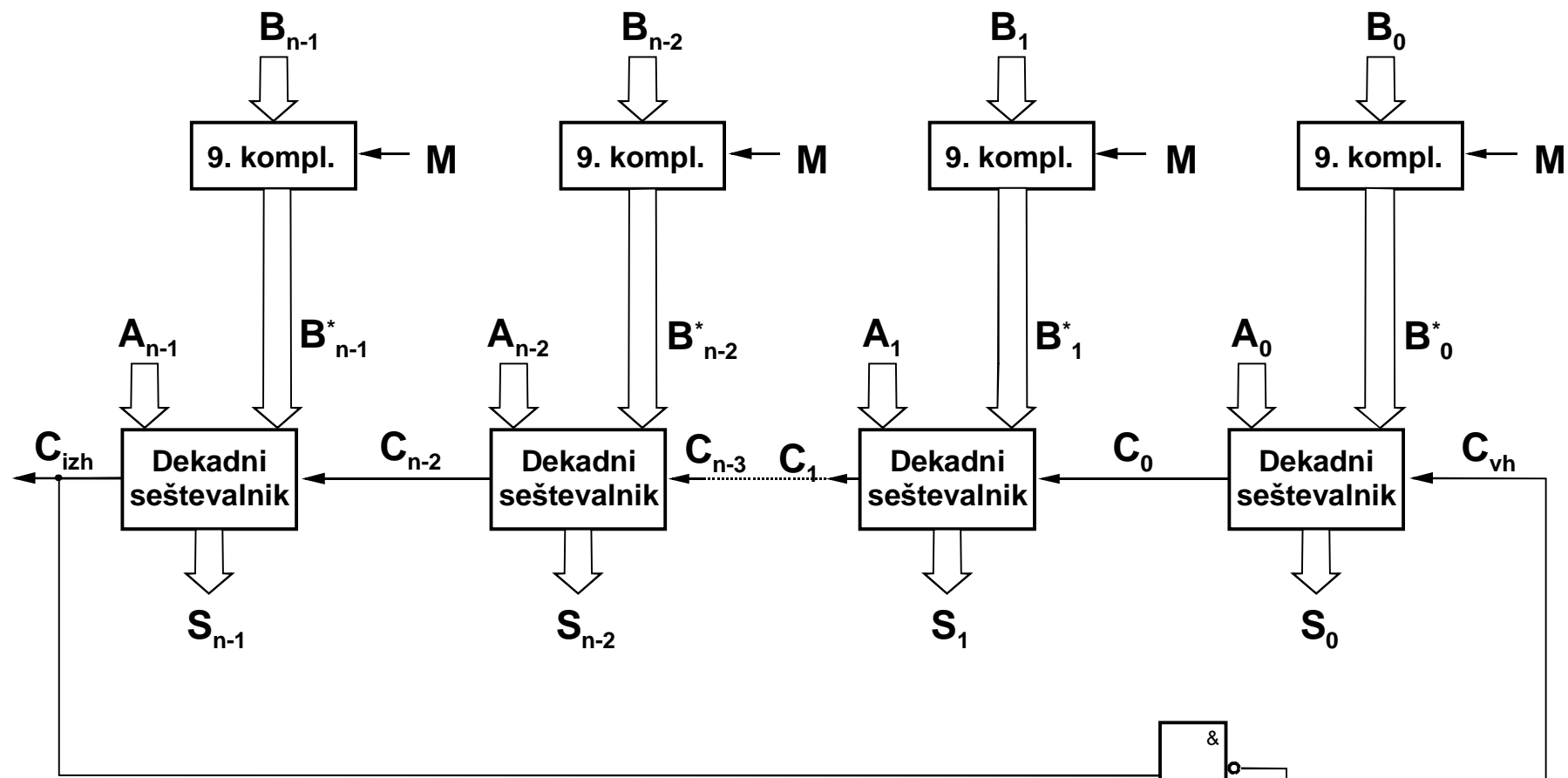
To izdatno poenostavi zgradbo seštevalnika.



A - (a_3, a_2, a_1, a_0) in B - (b_3, b_2, b_1, b_0) v mejah: $0 \leq A, B \leq 9$ ter prenos, ki je lahko le "0" in "1"

Izhod je vsota S (s_3, s_2, s_1, s_0) , ki je zopet v mejah: $0 \leq S \leq 9$; prenos pa le "0" in "1"

5. 4. 2 Dekadni serijski seštevalniki - odštevalniki

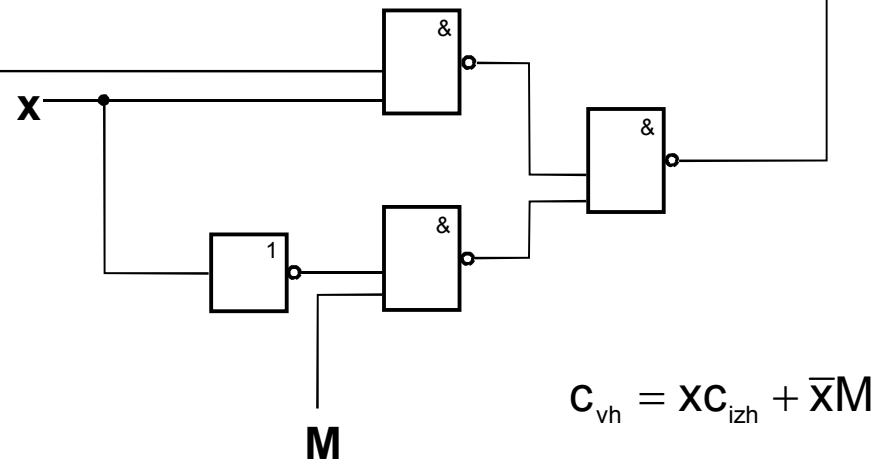


$x = 0$ – desetiški komplement

$x = 1$ – devetiški komplement

$M = 1$ – odštevanje

$M = 0$ – seštevanje



$$C_{vh} = xC_{izh} + \bar{M}$$