Arhitectura calculatoarelor 13.11.2024

(curs 7 - S7)

2.3.4. Conditional Sum Adder

→ bazat tot pe același principiu al sumei condiționate prin transport

→ Este o generalizare a CSeA

→ arhitectură multi-nivel

→ Structurat în mai multe etaje

- primul etaj calculează $\left\langle \begin{array}{l} (z_i^0, c_{i+1}^0), \text{dacă } c_i = 0 \\ (z_i^1, c_{i+1}^1), \text{dacă } c_i = 1 \end{array} \right.$
- etajele următoare utilizează biții transport intermediari pentru a selecta ranguri ale sumei

Se consideră operanzii pe 4 biți : X și Y și $c_0 = 0$

$$c_{i+1} = x_i y_i + x_i c_i + y_i c_i =$$

$$= x_i y_i (c_i + \overline{c_i}) + x_i c_i + y_i c_i = \qquad a + \overline{a} = 1$$

$$= x_i y_i c_i + x_i y_i \overline{c_i} + x_i c_i + y_i c_i =$$

$$= x_i y_i \overline{c_i} + x_i c_i (y_i + 1) + y_i c_i = \qquad a + 1 = 1$$

$$= x_i y_i \overline{c_i} + x_i c_i + y_i c_i =$$

$$= \underbrace{x_i y_i}_{g_i} \overline{c_i} + c_i \underbrace{(x_i + y_i)}_{p_i}$$

$$\Rightarrow c_{i+1} = g_i \cdot \overline{c_i} + p_i \cdot c_i$$

și forma complementată:

$$\overline{c_{i+1}} = \overline{g_i \cdot \overline{c_i} + p_i c_i} =$$

$$= \overline{g_i \cdot \overline{c_i}} \cdot \overline{p_i c_i} = \qquad \overline{a + b} = \overline{a} \cdot \overline{b}$$

$$= (\overline{g_i} + c_i)\cdot(\overline{p_i} + \overline{c_i}) = \qquad \overline{a\cdot b} = \overline{a} + \overline{b}$$

$$= \overline{g_i}\cdot\overline{p_i} + \overline{g_i}\cdot\overline{c_i} + c_i\cdot\overline{p_i} + c_i\cdot\overline{c_i} \qquad a\cdot\overline{a} = 0$$

$$= \overline{g_i}\cdot\overline{p_i}(\overline{c_i} + c_i) + \overline{g_i}\cdot\overline{c_i} + c_i\cdot\overline{p_i} =$$

$$= \overline{g_i}\cdot\overline{p_i}\cdot\overline{c_i} + \overline{g_i}\cdot\overline{p_i}\cdot c_i + \overline{g_i}\cdot\overline{c_i} + c_i\cdot\overline{p_i} =$$

$$= \overline{g_i}\,\overline{c_i}(\overline{p_i} + 1) + \overline{p_i}\,c_i(\overline{g_i} + 1) = \qquad a + 1 = 1$$

$$= \overline{g_i}\,\overline{c_i} + \overline{p_i}\cdot c_i$$

$$\Rightarrow \overline{c_{i+1}} = \overline{g_i}\cdot\overline{c_i} + \overline{p_i}\cdot c_i$$

$$c_{i+1} = g_i\,\overline{c_i} + p_i\cdot c_i$$

Biții de sumă $\qquad c_0 = 0 \qquad a\oplus b = \overline{a}b + a\overline{b}$

$$z_0 = x_0 \oplus y_0 \oplus c_0 = x_0 \oplus z_0$$

$$z_1 = x_1 \oplus y_1 \oplus c_1 = \overline{x_1 \oplus y_1}\cdot c_1 + x_1 \oplus y_1 \cdot \overline{c_1}$$

$$c_1 = g_0\cdot\underbrace{\overline{c_0}}_{1} + \underbrace{p_0\cdot c_0}_{0} = g_0$$

$$z_2 = x_2 \oplus y_2 \oplus c_2 = \overline{x_2 \oplus y_2}\cdot c_2 + x_2 \oplus y_2 \cdot \overline{c_2}$$

$$c_2 = g_1\cdot\overline{c_1} + p_1\cdot c_1$$

$$c_3 = g_2\,\overline{c_2} + p_2\,c_2$$

$$\overline{c_3} = \overline{g_2}\,\overline{c_2} + \overline{p_2}\cdot c_2$$

$$z_3 = x_3 \oplus y_3 \oplus c_3 = \overline{x_3 \oplus y_3}\cdot c_3 + x_3 \oplus y_3 \cdot \overline{c_3} =$$

$$= \overline{x_3 \oplus y_3}(g_2\,\overline{c_2} + p_2\,c_2) + x_3 \oplus y_3(\overline{g_2}\,\overline{c_2} + \overline{p_2}\cdot c_2) =$$

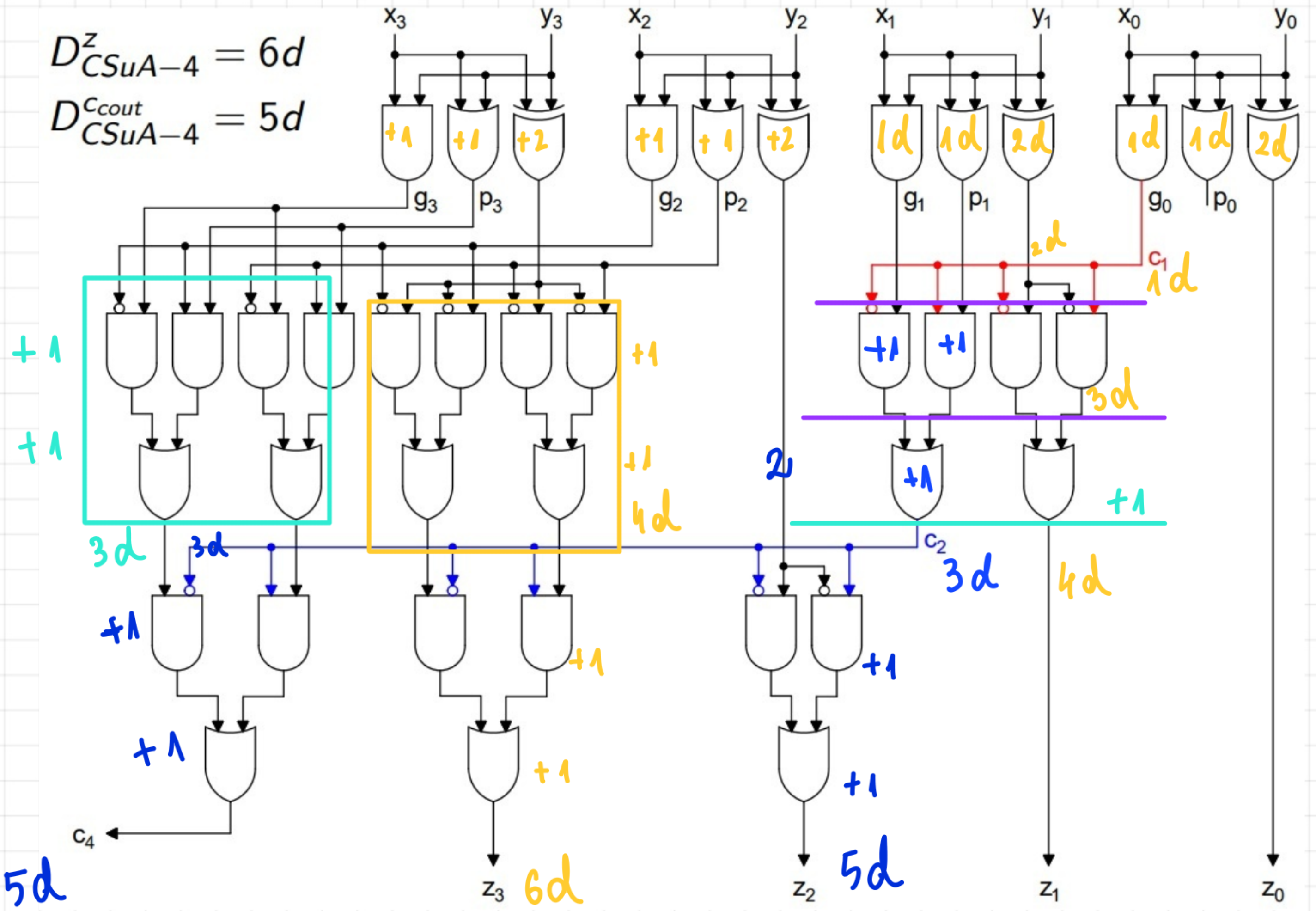
$$= \left[\overline{x_3 \oplus y_3}\cdot g_2 + (x_3 \oplus y_3)\cdot\overline{g_2}\right]\cdot\overline{c_2} \; +$$

$$+\left[\overline{x_3 \oplus y_3}\cdot p_2 + (x_3 \oplus y_3)\cdot \overline{p_2}\right] c_2$$

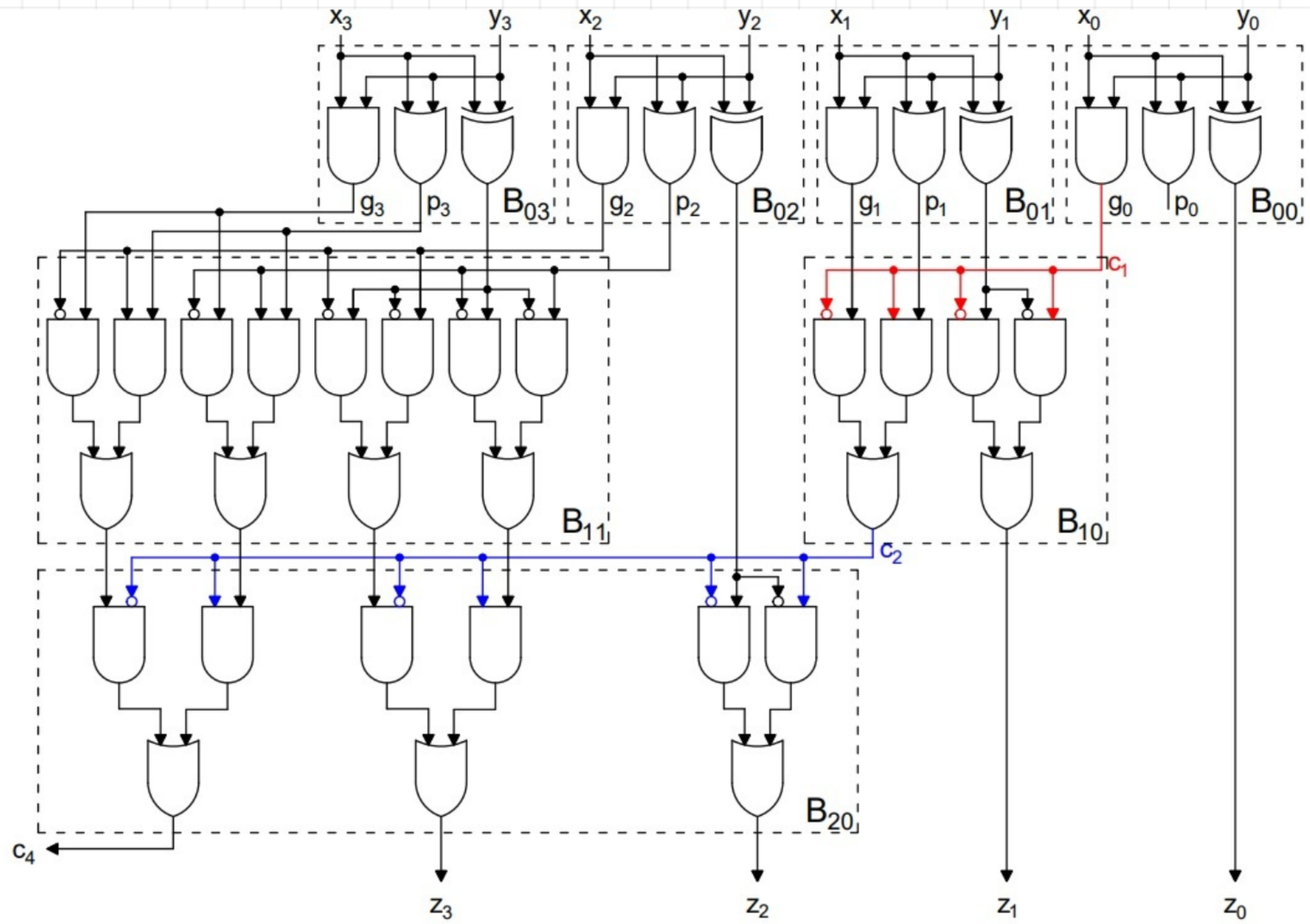
$$c_4 = g_3 \cdot \overline{c_3} + p_3 \cdot c_3 = g_3\left(\overline{g_2}\,\overline{c_2} + \overline{p_2}\, c_2\right) + p_3\left(g_2\,\overline{c_2} + p_2 \cdot c_2\right) =$$

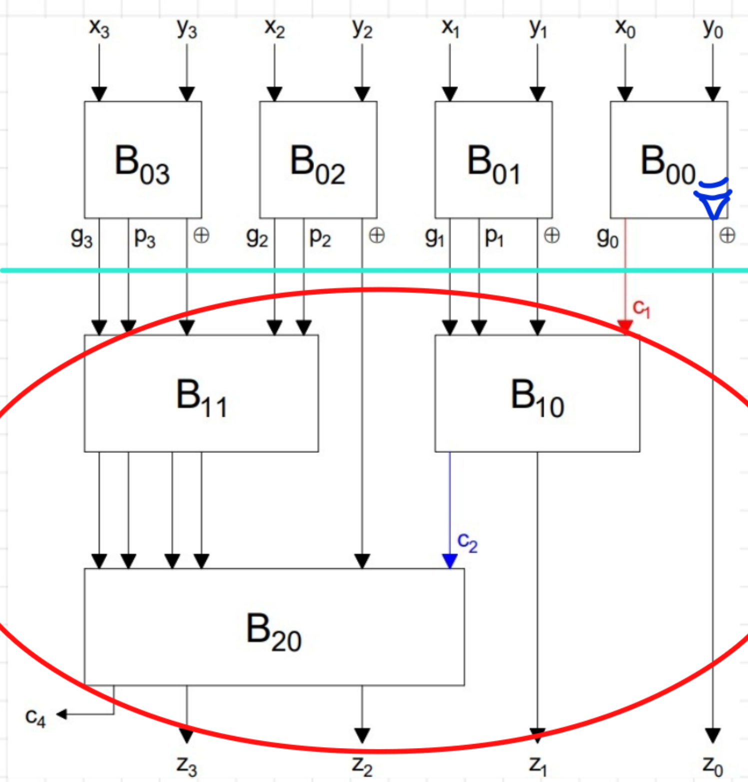
$$= \left[g_3\overline{g_2} + p_3 g_2\right]\overline{c_2} + \left[g_3\overline{p_2} + p_3 p_2\right] c_2$$

Arhitectura Conditional Sum Adder (CSuA)



Cu celule





nivel 0

n biți

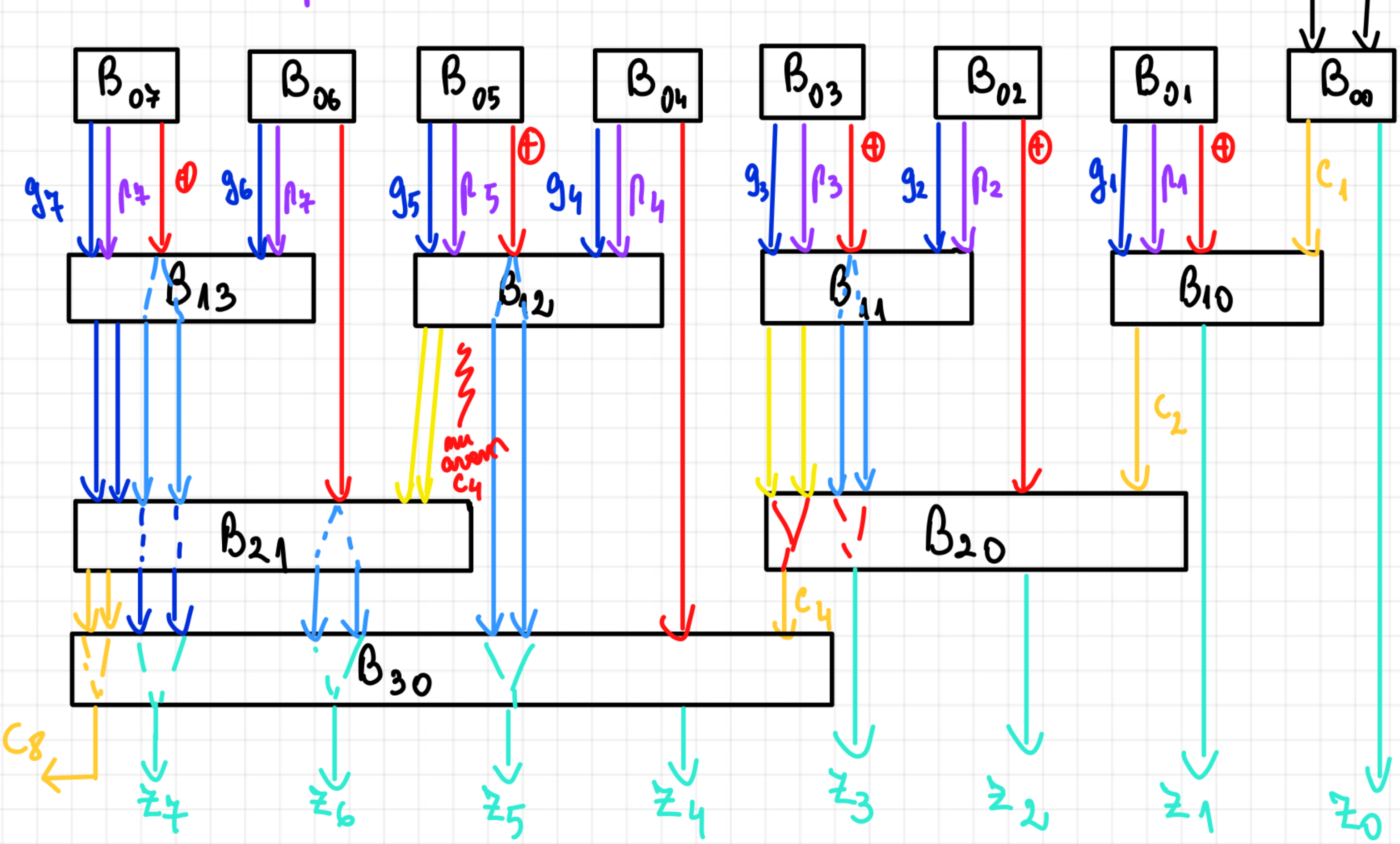
=> $\log_2 n$ nivele

arbore binar

4 biți ⟶ 3 nivele

Pentru evaluarea latenței CSuA în cazul general, se consideră operanzi pe n biți.

CSuA pe 8 biți



În cartea lui Vlăduțiu

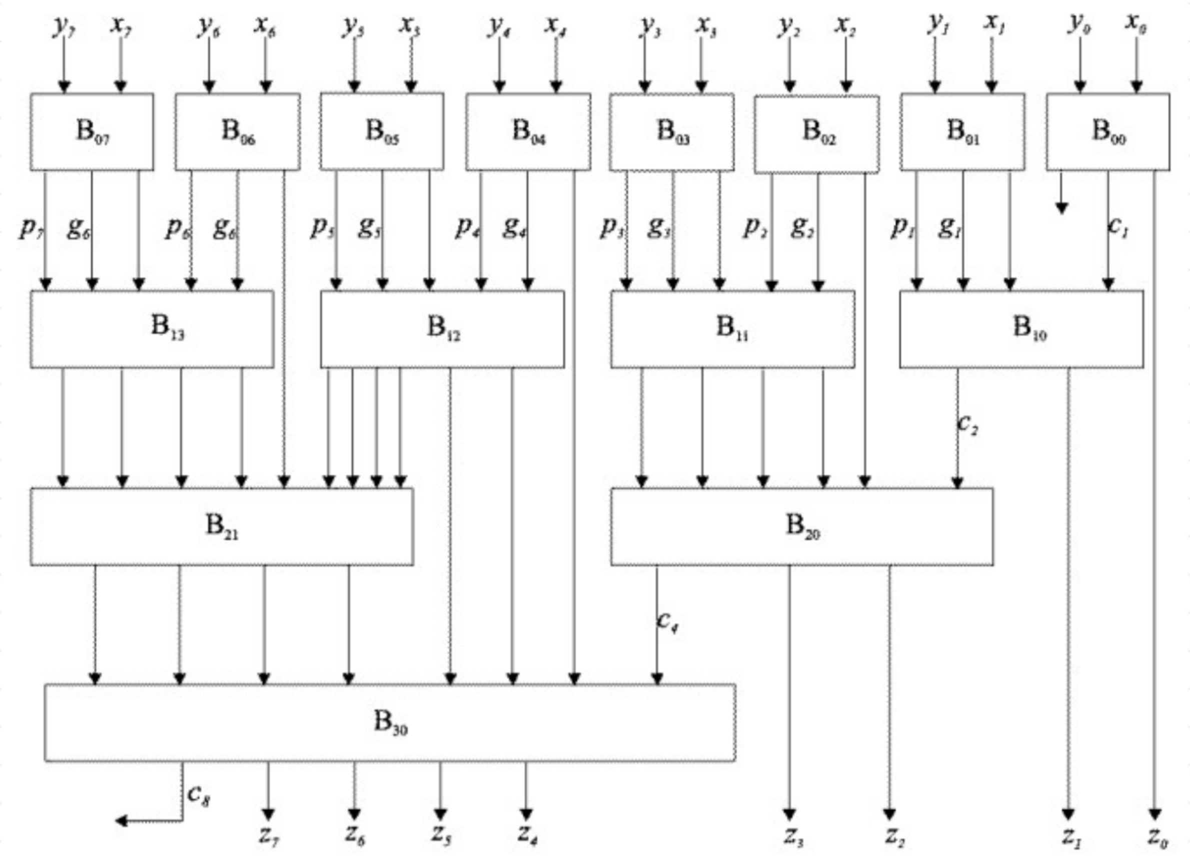
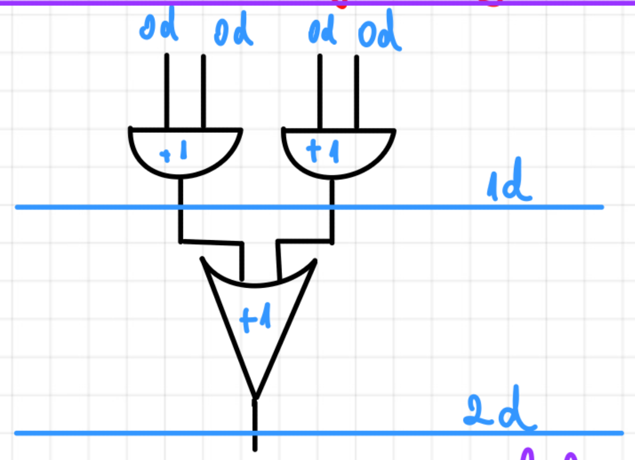


Fig. 2.29 Block diagram of an 8-bit CSuA

! Exceptând primul nivel, $\exists \lceil \log_2 n \rceil$ alte nivele în arhitectura CSuA.
Aceste nivele corespund celulelor B_{ij}, cu $j \geq 1$, și sunt similare nivelelor B+C din structura CLA multi-nivel.

$\longrightarrow$ fiecare nivel B_{ij} (cu $j \geq 1$) întârzie semnalele cu $2d$:



celulă $B_{ij}\ (j \geq 1)$

$$b^{z}_{CSuA-n} = \underbrace{2d}_{\substack{\text{poartă EXOR} \\ \text{din } B_{0j}}} + \underbrace{2}_{\substack{\text{întârziere} \\ B_{ij}(j\geq 1)}} \cdot \underbrace{\lceil \log_2 n \rceil d}_{\substack{\text{nr nivele} \\ \text{(fără nivel 0)}}} = 2(\lceil \log_2 n \rceil + 1)d$$

Cout pornește de la o întârziere de 1d

$$\Rightarrow b^{Cout}_{CSuA-n} = \underbrace{1d}_{g_i, p_i} + 2 \cdot \lceil \log_2 n \rceil d = (2\lceil \log_2 n \rceil + 1)d$$

Scurtă comparație a latențelor pe $n = 64$ de biți:

CSuA	CSeA	ML-CLA	RCA
$b^{z} = 2(6+1) = 14d$	$b^{z} = 2(\frac{n}{2}+1) = 66d$	$b^{z} = 4\lceil \log_2 n \rceil + 1 = 25d$	$b^{z} = 2n = 128d$
$b^{Cout} = 13d$	$b^{Cout} = 66d$	$b^{Cout} = 2\lceil \log_2 n \rceil + 3 = 15d$	$b^{Cout} = 2n = 128d$
	fără seg. variabile		

ex: Însumați operanzii pe 8 biți X și Y după metoda CSuA.

$X = 1\,0\,0\,0\,1\,1\,0\,1 \quad (=128+13=141)$

$Y = 0\,1\,1\,1\,0\,1\,0\,1 \quad (=64+32+16+5=117)$

Primul nivel : calcularea perechilor (z_i^0, c_{i+1}^0) și (z_i^1, c_{i+1}^1) pt. fiecare rang

Operand \ Rank		7		6		5		4		3		2		1		0	
X		1		0		0		0		1		1		0		1	
Y		0		1		1		1		0		1		0		1	
Block level	Carry in	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
I. i=0	$c_{in}=0$	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	0
	$c_{in}=1$	1	0	1	0	1	0	1	0	1	0	1	1	0	1		
i=1	$c_{in}=0$																
	$c_{in}=1$																
i=2	$c_{in}=0$																
	$c_{in}=1$																
i=3	$c_{in}=0$																
	$c_{in}=1$																

⟶ buffer overflow

I. $i=0$, $c_{in}=0$ rank 0 $x=1$ $y=1$ $\Rightarrow \underbrace{S}_{\text{sumă}}=0$ $\quad \underbrace{C}_{\text{carry out}}=1$

$c_{in}=1$ nu se poate

⟶ se continuă sumele

Nivelul al doilea: determinarea variantelor de sume corecte pentru blocuri de câte 2 ranguri binare.

Operand \ Rank		7		6		5		4		3		2		1		0	
X		1		0		0		0		1		1		0		1	
Y		0		1		1		1		0		1		0		1	
Block level	Carry in	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
i=0	$c_{in}=0$	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	0
	$c_{in}=1$	1	0	1	0	1	0	1	0	1	0	1	1	0	1		
i=1	$c_{in}=0$	0	1		1	0	1		1	1	0		0	0	1		0
	$c_{in}=1$	1	0		0	1	0		0	1	0		1				
i=2	$c_{in}=0$																
	$c_{in}=1$																
i=3	$c_{in}=0$																
	$c_{in}=1$																

! val. încercuită (carry rang curent) ne indică ce transport va fi trimis către rangul următor.

$i = 1$, rang 1 => vom avea (0 1) ⇔ $C_1 = 1$ pe aceeași linie

Nivelul al treilea: determinarea variantelor de sume corecte pentru blocuri de câte 4 ranguri binare

2^2 | carry de la mijloc folosit

2^1

Operand \ Rank		7		6		5		4		3		2		1		0	
X		1		0		0		0		1		1		0		1	
Y		0		1		1		1		0		1		0		1	
Block level	Carry in	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
i=0	$c_{in}=0$	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	0
	$c_{in}=1$	1	0	1	0	1	0	1	0	1	0	1	1	0	1		
i=1	$c_{in}=0$	0	1		1	0	1		1	1	0		0	0	1		0
	$c_{in}=1$	1	0		0	1	0		0	1	0		1				
i=2	$c_{in}=0$	0	1		1		1		1	1	0		0		1		0
	$c_{in}=1$	1	0		0		0		0								
i=3	$c_{in}=0$																
	$c_{in}=1$																

2^0

2^1 $c_{in} = c_0 = 0$ (nu poate fi 1)

2^2

Nivelul al patrulea: determinarea sumei corecte pentru întreg setul de ranguri al celor 2 operanzi.

Operand \ Rank		7		6		5		4		3		2		1		0	
X		1		0		0		0		1		1		0		1	
Y		0		1		1		1		0		1		0		1	
Block level	Carry in	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
i=0	c_{in}=0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	0
	c_{in}=1	1	0	1	0	1	0	1	0	1	0	1	1	0	1		
i=1	c_{in}=0	0	1		1	0	1		1	1	0		0	0	1		0
	c_{in}=1	1	0		0	1	0		0	1	0		1				
i=2	c_{in}=0	0	1		1		1		1	1	0		0		1		0
	c_{in}=1	1	0		0		0		0								
i=3	c_{in}=0	1	0		0		0		0		0		0		1		0
	c_{in}=1																

rezultat

ex curs:

Operand \ Rank		9		8		7		6		5		4		3		2		1		0	
X		0		1		1		0		0		1		0		0		1		1	
Y		1		1		0		1		1		0		1		1		1		0	
BL.	C_{in}	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
i=0	0	0	1	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	0	0	1
	1	1	0	1	1	1	0	1	0	1	0	1	0	1	0	1	0	1	1		
i=1	0	1	0		0	0	1		1	0	1		1	0	1		1	1	0		1
	1	1	0		1	1	0		0	1	0		0	1	0		0				
i=2	0	1	0		0	0	1		1		1		1	1	0		0		0		1
	1	1	0		1	1	0		0		0		0								
i=3	0	1	0		0	1	0		0		0		0		0		0		0		1
	1	1	0		1																
i=4	0	1	0		1		0		0		0		0		0		0		0		1
	1																				

Dacă nu e multiplu de 4, copiem

Ex Vlăduțiu

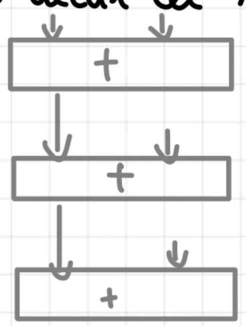
Range		15		14		13		12		11		10		9		8		7		6		5		4		3		2		1		0	
X		1		0		1		0		1		1		0		1		0		0		0		1		0		1		0		1	
Y		0		0		1		1		0		1		1		0		1		0		1		0		1		1		0		1	
Block level	Carry in	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S	C	S
i=0	$c_{in}=0$	0	1	0	0	1	0	0	1	0	1	1	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	1	0	0	0	1	0
	$c_{in}=1$	1	0	0	1	1	1	1	0	1	0	1	1	1	0	1	0	1	0	0	1	1	0	1	0	1	0	1	1	0	1		
i=1	$c_{in}=0$	0	1		0	1	0		1	1	0		0	0	1		1	0	1		0	0	1		1	1	0		0	0	1		0
	$c_{in}=1$	0	1		1	1	1		0	1	0		1	1	0		0	0	1		1	1	0		0	1	0		1				
i=2	$c_{in}=0$	0	1		1		0		1	1	0		0		1		1	0	1		0		1		1	1	0		0		1		0
	$c_{in}=1$	0	1		1		1		0	1	0		1		0		0	0	1		1		0		0								
i=3	$c_{in}=0$	0	1		1		1		0		0		0		1		1	0	1		1		0		0		0		0		1		0
	$c_{in}=1$	0	1		1		1		0		0		1		0		0																
i=4	$c_{in}=0$	0	1		1		1		0		0		0		1		1		1		1		0		0		0		0		1		0

2.3.5. Carry Save Adder (CSA)

→ sumă în format redundant < sumă / transport

! Dacă nu ar fi impedimentul legat de formatul redundant, acest sumator ar fi cel mai performant, deoarece întârzierea pe un astfel de sumator este constantă.

Vectorul transport este cu o poziție mai semnificativ decât cel sumă

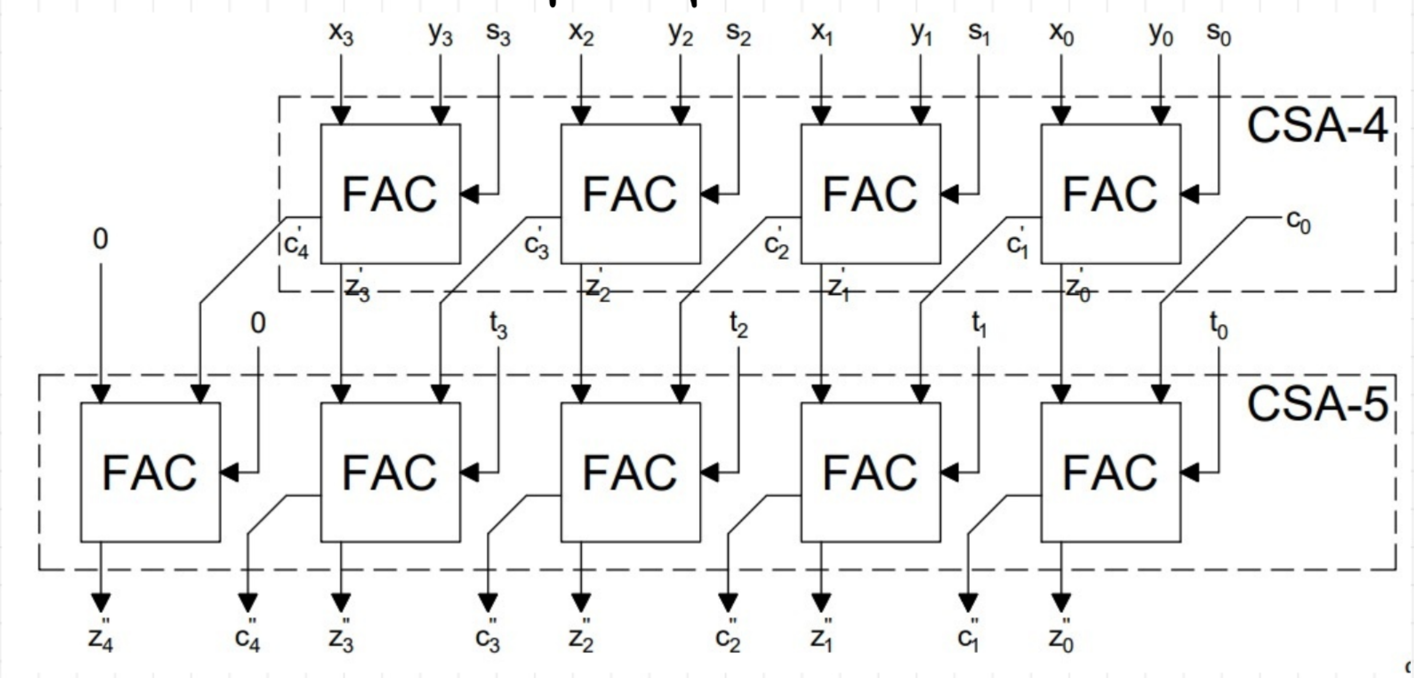


! Permite adunarea multi-operand.

→ celulele FAC sunt independente.

X, Y, S, T - 4 biți

Suma $Z = X+Y+Z+T$ poate fi calculată:



! bitul transport este adunat cu un rang mai semnificativ decât bitul sumă înlănțuind astfel nivele CSA.

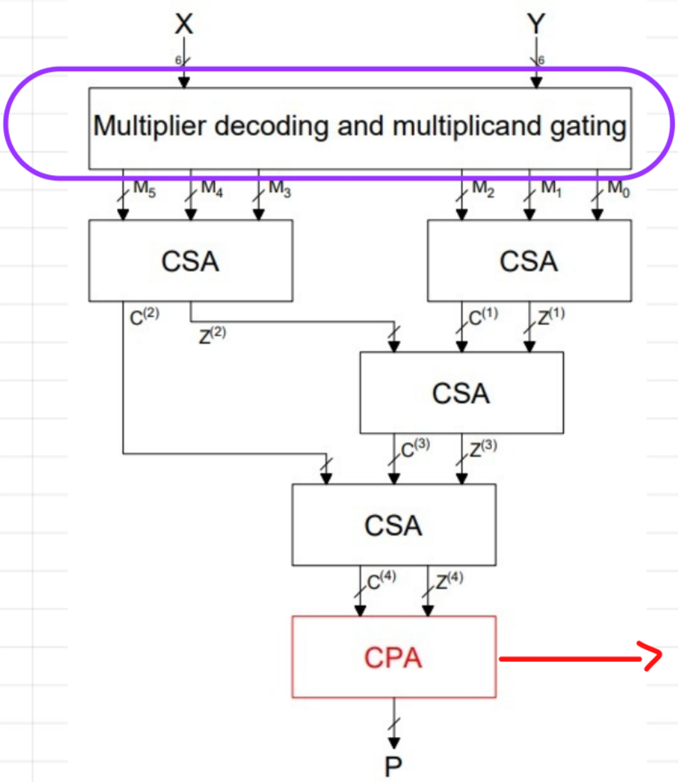
⟶ facilitează realizarea operației de înmulțire (combinațional)

Fie X, Y — nr. fără semn pe 6 biți

Produsul $P = X \cdot Y$ este obținut prin adunarea produselor de 1 bit

$M_i = x_i \cdot Y \cdot 2^i$

8 un produs = o sumă multi-operandă



⟶ $X = \sum x_i \cdot 2^i$

$\Rightarrow P = \sum_i x_i \cdot Y \cdot 2^i$

⟶ carry propagate adder

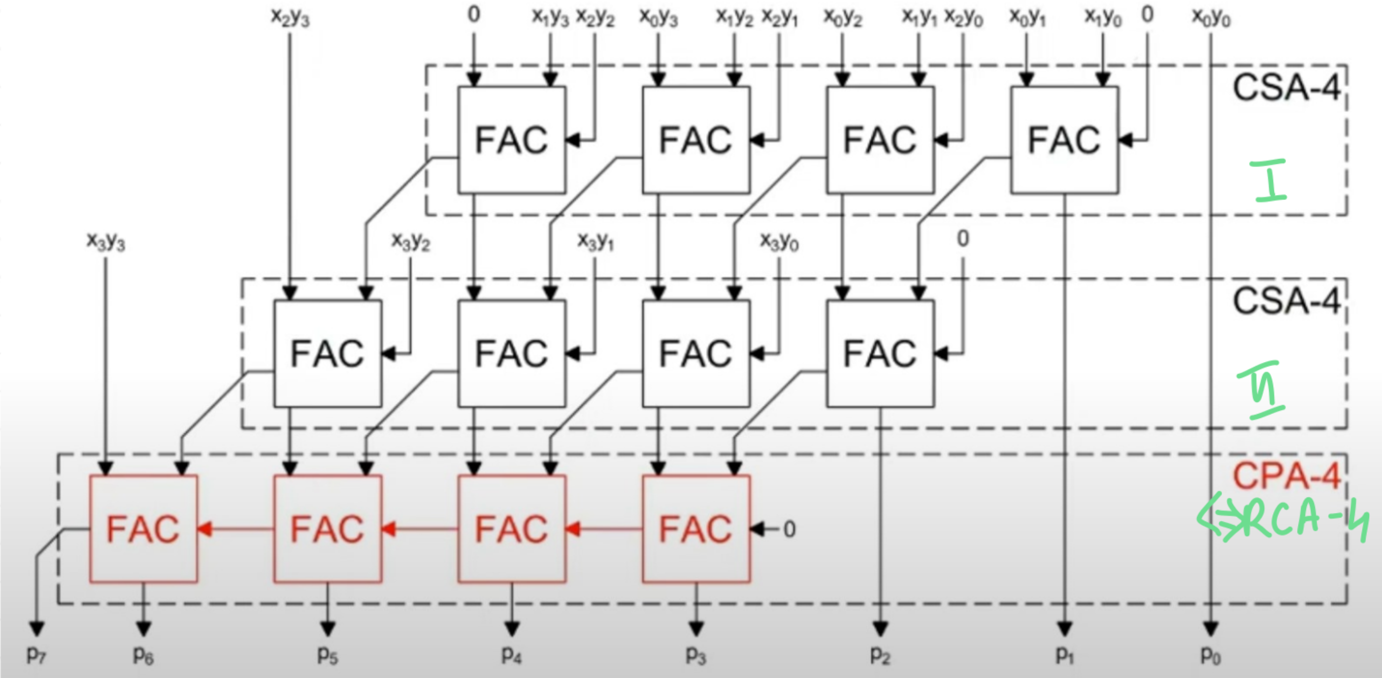
Nivel mov: iau câte un bit din X și M_i — rămâne la fel (Y)
— va fi adus la 0, dacă $x_i = 0$

$M_0 = 6b$ $M_1 = 7b$ $M_2 = 8b$ $M_3 = 9b$ $M_4 = 10b$ $M_5 = 11b$

CSA ⟶ generez perechi

CPA - sumator binar care propagă de la un rang la altul transportul $\Rightarrow$ produs

Structura de mai jos permite calcularea produsului a două numere fără semn pe 4 biți: $X = x_3x_2x_1x_0$ $\quad Y = y_3y_2y_1y_0$ $\Rightarrow$ 4 P de 1 bit



I, II $\longrightarrow$ îmi permit să adun 4 vectori

Ex. de pe curs

4 biți $\Rightarrow M_i,\ i \in [0,3]$

$x = 15$ $\qquad$ 1 1 1 1

$y = 13$ $\qquad$ 1 1 0 1

$M_0 = x_0 \cdot y \cdot 2^0$ $\qquad$ 1 1 0 1 $\qquad$ $1 \cdot 13 \cdot 1 = 13$

$M_1 = x_1 \cdot y \cdot 2^1$ $\qquad$ 1 1 0 1 0 $\qquad$ $1 \cdot 13 \cdot 2 = 26$ $\qquad$ sau muți la stânga

$M_2 = x_2 \cdot y \cdot 2^2$ $\qquad$ 1 1 0 1 0 0 $\qquad$ +CSA $\qquad$ $1 \cdot 13 \cdot 4 = 52$

$z^{(1)}$ $\qquad$ 1 0 0 0 1 1

$c^{(1)}$ $\qquad$ 0 1 1 1 0 0

$2c^{(1)}$ $\qquad$ 0 1 1 1 0 0 0

$z^{(1)}$ $\qquad$ 1 0 0 0 1 1

$M_3 = x_3 y \cdot 2^3$ $\qquad$ 1 1 0 1 0 0 0 $\qquad$ +CSA

$c^{(1)}$ e cu o poziție mai semnificativă decât $z^{(1)}$

$z^{(2)}$ $\qquad$ 1 1 1 0 0 1 1

$c^{(2)}$ $\qquad$ 0 1 0 1 0 0 0

$2c^{(2)}$ 01010000 (carries: 1 1)

$z^{(2)}$ 1110011 — +CPA

P 11000011 = 195

ex de la cours

$X =$ 101101 = 45

$Y =$ 1001 = 9

$M_0 = x_0 Y \cdot 2^0$ 1001

$M_1 =$ 00000

$M_2 =$ 100100 — +CSA

$z^{(1)}$ 101101

$c^{(1)}$ 000000

$M_3 = x_3 Y \cdot 2^3$ 1001000

M_4 000000000

M_5 100100000

$z^{(2)}$ 101101000

$c^{(2)}$ 000000000

$z^{(1)}$ 101101

$2 \cdot c^{(1)}$ 0000000

$z^{(2)}$ 101101000

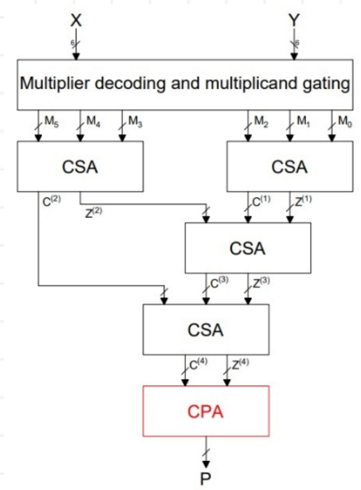
$z^{(3)}$ 101000101

$c^{(3)}$ 000101000

$z^{(3)}$ 101000101

$2c^{(3)}$ 0001010000

$2c^{(2)}$ 0000000000 — +CSA



$z^{(4)}$	1 0 0 0 1 0 1 0 1
$c^{(4)}$	0 0 1 0 0 0 0 0 0
$z^{(4)}$	1 0 0 0 1 0 1 0 1
$2 \cdot c^{(4)}$	0 0 1 0 0 0 0 0 0 0
$p.$	0 1 1 0 0 1 0 1 0 1

$(1+4+16+128+256 = 405)$ ✓