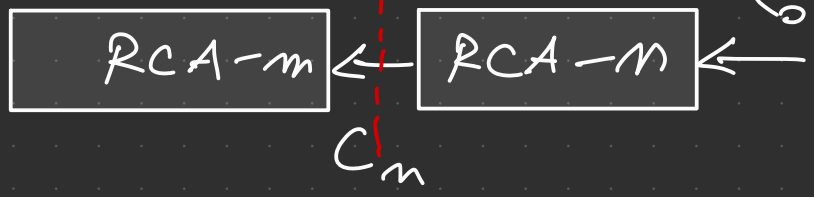
CMOS Pre-Discharge -> Initializare la 0

CSkA Carry Skip Adder

-> transportările sunt pre-discharged

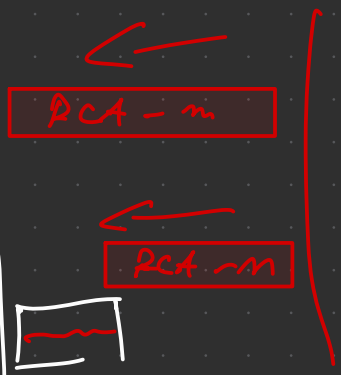
$n + m$ biți

paralel



$C_m == 0 \Rightarrow C_m$ corect 0d

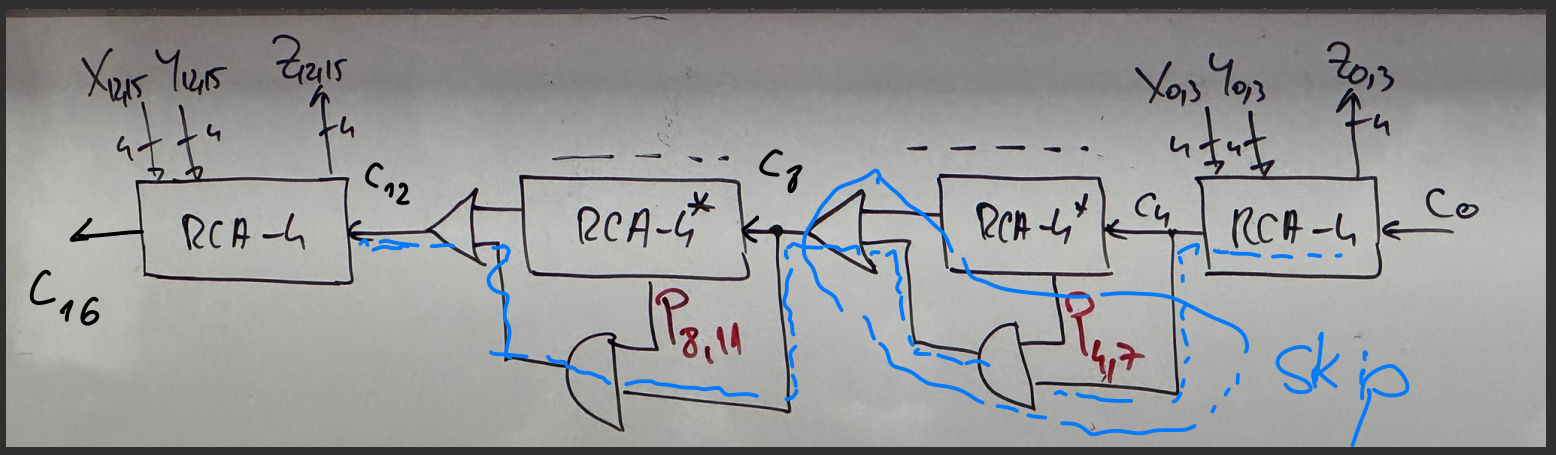
Calea critică RCA

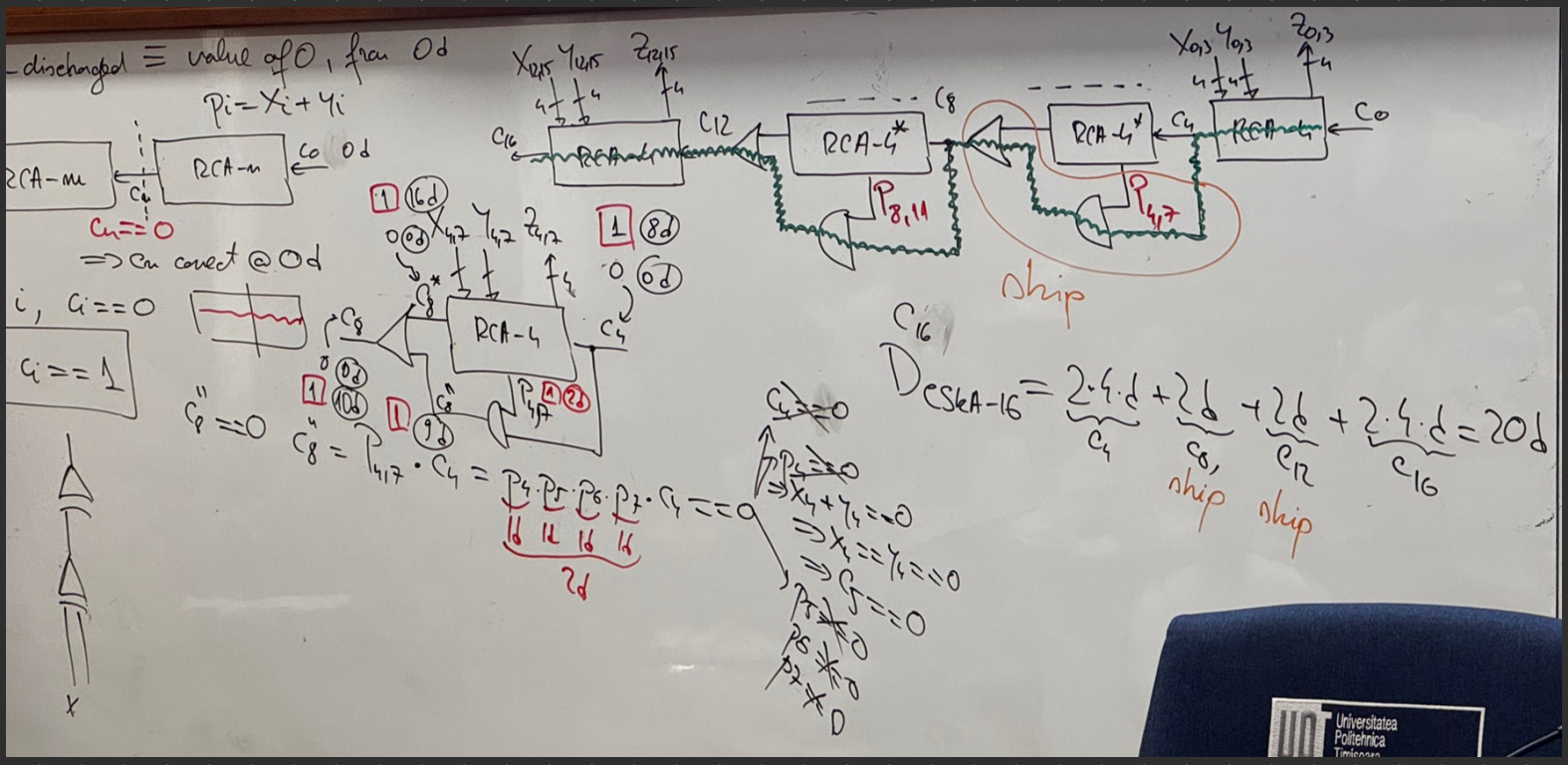


$(\exists)\ i \quad C_i == 0$

$(\forall)\ C_i == 1$ Worst Case

RCA^* -> are ieșirile propagate





Latimea optimă pt. RCA

$$n = b * k$$

b — RCA segment width

$\frac{n}{b}$ RCA $\begin{cases} 2 \text{ ms skip} \begin{cases} \text{first} \\ \text{last} \end{cases} \\ \frac{n}{b} - 2 \text{ with skip} \end{cases}$

$$\Delta_{CSkA-n} = 2bd + 2\left(\frac{n}{b} - 2\right)b + 2bd =$$

$$= \left(4b + \frac{2n}{b} - 4\right)d$$

$$\frac{d\,\Delta_{CSkA-n}^{Z/cout}}{db} = 4 - \frac{2n}{b^2} = 0$$

$$\boxed{b = \sqrt{\frac{n}{2}}}$$

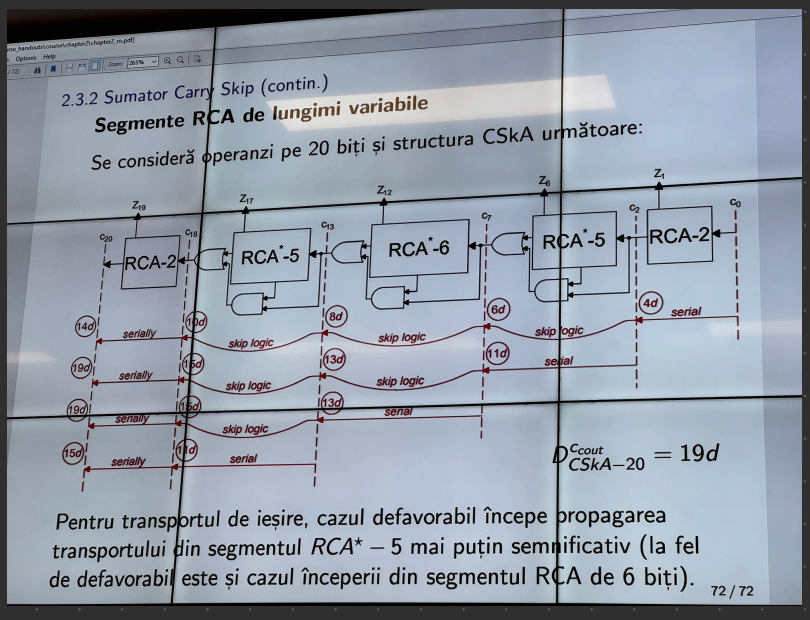
$n = 32 \rightarrow b = 4$ $\quad \Delta_{opt} = 28d$

$n = 126 \rightarrow b = 8$ $\quad \Delta_{opt} = 60d$

$$\Delta_{opt} = 4\left(\sqrt{2n} - 1\right)d$$

$$RCA_{128} : \Delta = 256d$$

Variable-sized RCA segments



Set. Latența Cout

EXAMEN!

↗ bitul transport

↘ bitul sumă

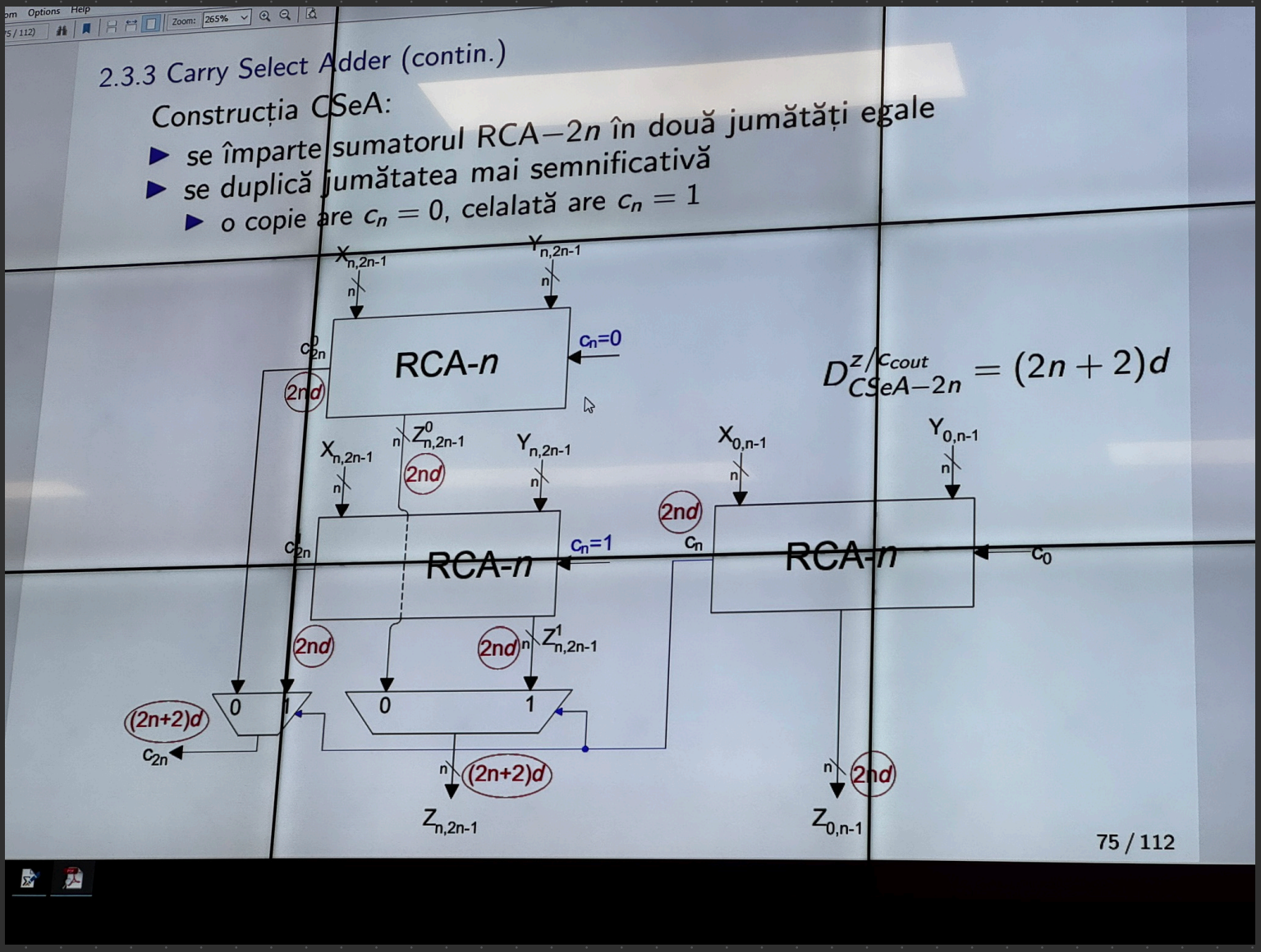
trade-off

Carry Select Adder CSeA

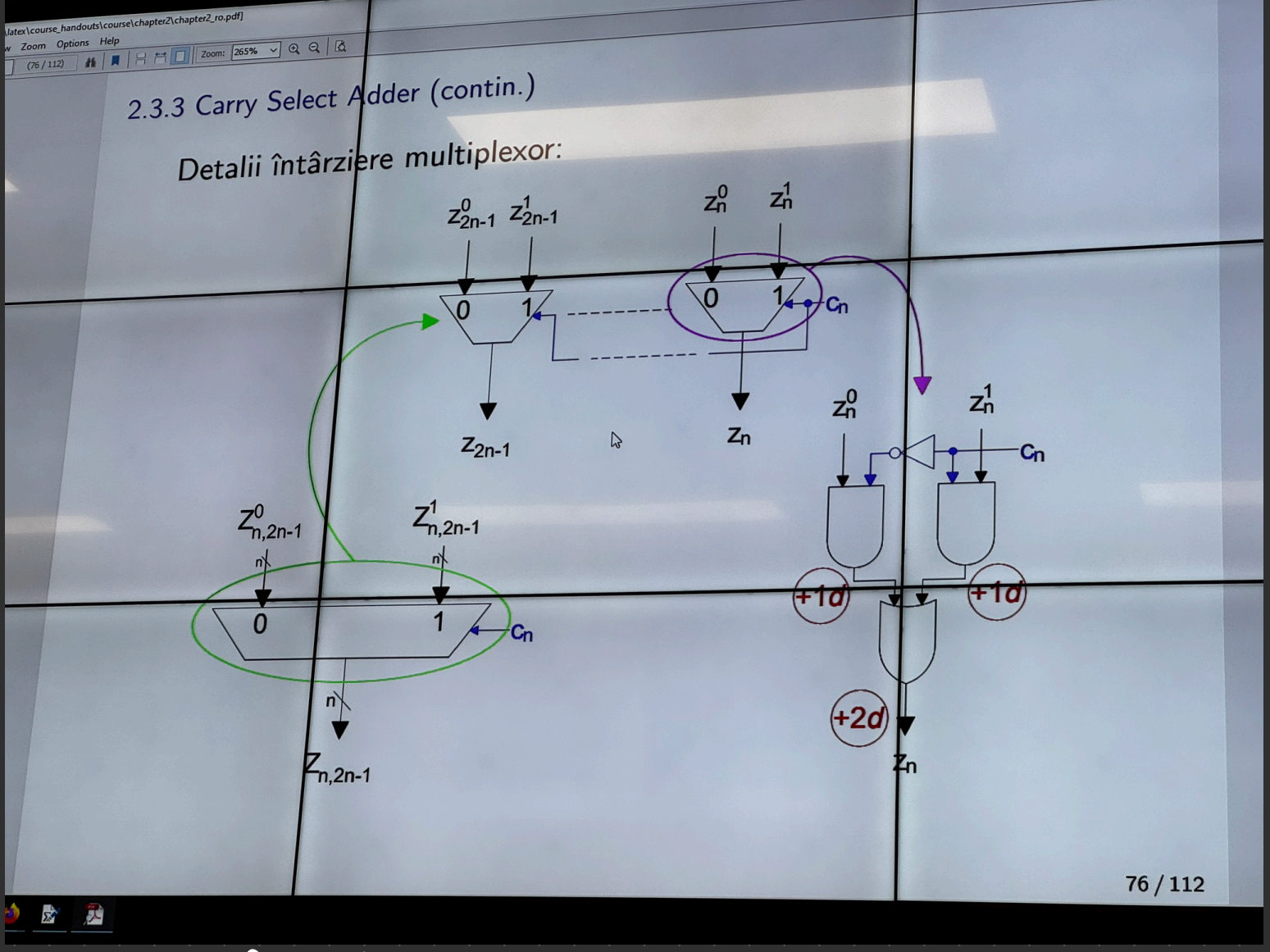
-> principiul sumei condiționate prin transport

$C_i \begin{cases} 0 \Rightarrow z_i^0 \; c_{i+1}^0 & \text{if } c = 0 \\ 1 \Rightarrow z_i^1 \; c_{i+1}^1 & \text{if } c = 1 \end{cases}$

$RCA - 2n \qquad \Delta_{RCA-2n} = 4n\,d$



carry 1 } pt n;2n-1 -> MUX la final
carry 0 }



$$\int^{z/cout}_{CS_e\,A-2n} = (2n+2)\,d$$

↘ duplicare hardware + cost energie

2.3.3 Carry Select Adder (contin.)

Optimizarea ariei CSeA:

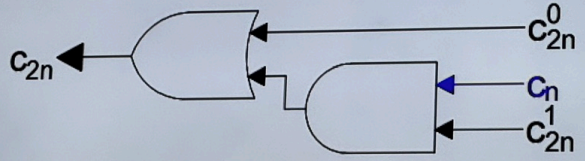
- se pornește de la tabelul de adevăr al semnalului c_{2n}
- **Niciodată** c_{2n}^0 nu poate fi mai mare decât c_{2n}^1!
  - **Întrebare**: de ce?
  - $\Rightarrow$ ~~tabelul de adevăr folosește *don't care* pentru cazurile imposibile~~

Inputs			Output
c_n	c_{2n}^0	c_{2n}^1	c_{2n}
0	0	0	0
0	0	1	0
0	1	0	d
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	d
1	1	1	1

c_{2n} / c_n	c_{2n}^0, c_{2n}^1 = 00	01	11	10
0	0	0	1	d
1	0	1	1	d

$$c_{2n} = c_{2n}^0 + c_{2n}^1 \cdot c_n$$

În consecință, transportul de ieșire c_{2n} va fi generat astfel:



1 s bit -> $2k = 15$ ✗

$k+k+k+1 = 15$ ✗

$3k+1 = 15$

$k+k+k+1+k+2 = 15$

$4k = 12 \quad k = 3$

2.3.3 Carry Select Adder (contin.)

Optimizarea ariei CSeA:

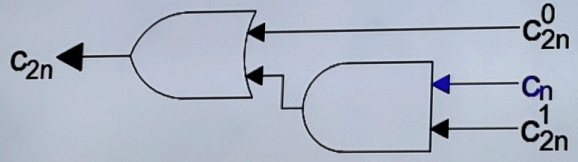
- se pornește de la tabelul de adevăr al semnalului c_{2n}
- **Niciodată** c_{2n}^0 nu poate fi mai mare decât c_{2n}^1!
  - **Întrebare**: de ce?
  - $\Rightarrow$ ~~tabelul de adevăr folosește *don't care* pentru cazurile imposibile~~

Inputs			Output
c_n	c_{2n}^0	c_{2n}^1	c_{2n}
0	0	0	0
0	0	1	0
0	1	0	*d*
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	*d*
1	1	1	1

c_{2n} / c_n	c_{2n}^0, c_{2n}^1 = 00	01	11	10
0	0	0	1	d
1	0	1	1	d

$$c_{2n} = c_{2n}^0 + c_{2n}^1 \cdot c_n$$

În consecință, transportul de ieșire c_{2n} va fi generat astfel:



77 / 1

Subiect -> nr. de biți -> distribuire a segmentelor eficientă

CSkA Multilevel

Conditional Sum Adder

CSuA

→ primul etaj calculează $\begin{cases} z_i^0, c_{i+1}^0 - c_i = 0 \\ z_i^1, c_{i+1}^1 - c_i = 1 \end{cases}$

→ celelalte nivele folosesc biți transport care leagă segmentele

$$c_{i+1} = \underbrace{x_i y_i}_{g_i} \overline{c_i} + \underbrace{(x_i + y_i)}_{p_i} c_i$$

$c_i = 0 \rightarrow c_{i+1} = g_i$

$c_i = 1 \rightarrow c_{i+1} = p_i$

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

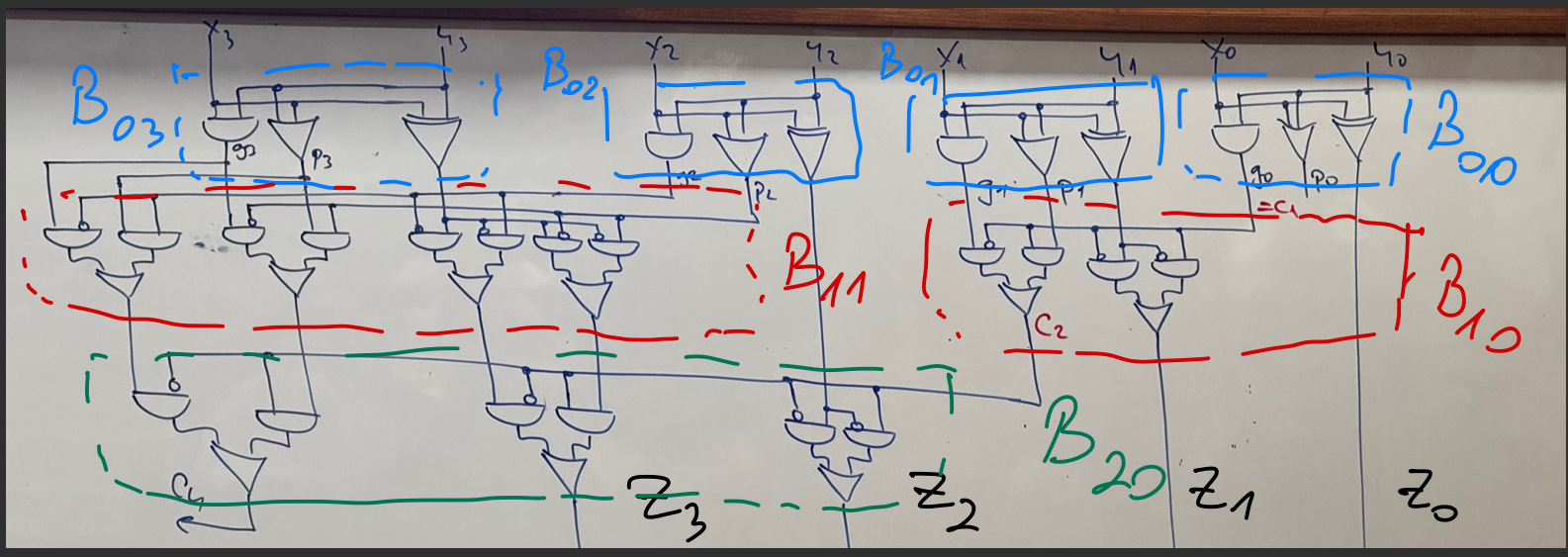
$$= (\overline{g_i} + c_i)(\overline{p_i} + \overline{c_i})$$

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

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

Sum bits	Carry bits
$z_0 = x_0 \oplus y_0 \oplus \cancel{0}$	$c_1 = g_0 \cdot \overline{c_0} + p_0 \quad \begin{matrix} c_0 = g_0 \\ \vdots \\ 0 \end{matrix}$
$z_1 = x_1 \oplus y_1 \oplus c_1 = \overline{x_1 \oplus y_1}\, c_1 + (x_1 \oplus y_1)\overline{c_1}$	$c_2 = g_1 \overline{c_1} + p_1 c_1$
$z_2 = \overline{x_2 \oplus y_2}\, c_2 + (x_2 \oplus y_2)\overline{c_2}$	$c_3 = g_2 \overline{c_2} + p_2 c_2$ $c_3 = \overline{g_2}\, c_2 + \overline{p_2}\, c_2$
$z_3 = \overline{x_2 \oplus y_2}\, c_3 + (x_2 \oplus y_2)\overline{c_3}$ $= \left(\overline{x_2 \oplus y_2}\right)(g_2 \overline{c_2} + p_2 c_2) +$ $+ (x_2 \oplus y_2)(\overline{g_2}\, \overline{c_2} + \overline{p_2}\, c_2)$	

→ Înlocuire Carry cu formula lui



2.3.4 Conditional Sum Adder (contin.)

Arhitectura Conditional Sum Adder (CSuA):

$$D^{z}_{CSuA-4} = 6d$$

$$D^{c_{cout}}_{CSuA-4} = 5d$$