Course 1

6.10.22

Sisteme de reprezentare a numerelor

- baza n → 0, 1, ..., n−1

$$\overset{4}{2}\,\overset{3}{1}\,\overset{2}{3}\,\overset{1}{4}\,\overset{0}{6} = 2\cdot 10^{4} + 1\cdot 10^{3} + 3\cdot 10^{2} + 4\cdot 10^{1} + 6\cdot 10^{0}$$

- adunare în baza 10

$$\begin{array}{r} 7126\,+ \\ 1390 \\ \hline 8516 \end{array}$$

- baza 2 → 0, 1 → bit
- conversie din baza 10 în baza 2

			c	r
143_{10}	: 2	=	71	1
71	: 2	=	35	1
35	: 2	=	17	1
17	: 2	=	8	1

			c	r
8	: 2	=	4	0
4	: 2	=	2	0
2	: 2	=	1	0
1	: 2	=	0	1

invers

$$\overset{7}{1}\,\overset{6}{0}\,\overset{5}{0}\,\overset{4}{0}\,\overset{3}{1}\,\overset{2}{1}\,\overset{1}{1}\,\overset{0}{1}_{\,2} = 1\cdot 2^7 + \cancel{0\cdot 2^6} + \cancel{0\cdot 2^5} + \cancel{0\cdot 2^4} + 1\cdot 2^3 +$$

$$+ 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$$

$$= 128 + 8 + 4 + 2 + 1$$

$$= 143$$

- conversie din baza 2 în baza 10

$2^0 = 1$ $\qquad$ $2^5 = 32$

$2^1 = 2$ $\qquad$ $2^6 = 64$

$2^2 = 4$ $\qquad$ $2^7 = 128$ $\qquad$ obs

$2^3 = 8$ $\qquad$ $2^8 = 256$

$2^4 = 16$

metoda intuitivă

$$98_{10} = 2^6 + 2^5 + 2^1 = 64 + 32 + 2$$

$$\Rightarrow \overset{6}{1}\,\overset{5}{1}\,\overset{4}{0}\,\overset{3}{0}\,\overset{2}{0}\,\overset{1}{1}\,\overset{0}{0}$$

bit = elementul de date cu care reprezentăm o cifră binară → 0, 1

semioctet → 4b $\qquad$ semicuvânt → 16b

octet → 8b (byte) $\qquad$ cuvânt → 32b (word)

4b $2 \times 2 \times 2 \times 2 = 2^4$

□□□□

0 0 0 0 = 0
0 0 0 1 = 1
0 0 1 0 = 2
0 0 1 1 = 3
0 1 0 0 = 4
⋮
1 1 1 1 = 15 = $2^4 - 1$

$$\left.\begin{array}{ll} 8\text{b} & 2^8 - 1 \\ 16\text{b} & 2^{16} - 1 \\ 32\text{b} & 2^{32} - 1 \end{array}\right\} \text{max}$$

- aritmetică în baza 2

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 10_2

ex pe 4b:

$$\begin{array}{r} 1011\,+ \\ 0011 \\ \hline 1110 \end{array} = 14$$

explicație

$1 + 1 = 2_{10} : 2 = 1 \; 0$

$1 : 2 = 0 \; 1$

$\Rightarrow 1 + 1 = 10_2$

• nr negative : bit de semn $\begin{cases} + & 0 \\ - & 1 \end{cases}$

$\boxed{0}1111 = 15$

$\boxed{1}1111 = -15$

$\boxed{0}0000 = 0$

$\boxed{1}0000 = -0$

semn magnitudine

• reprezentarea în complement față de 2:

nr pozitiv → la fel ca în baza 2

nr negativ a → 1) reprez a în baza 2

pași

2) negăm biți

3) adunăm val obținută la 1)

ex: -6

1) $6 = 0110$

2) 1001

3) 1

$$\begin{array}{r} 0110\,+ \\ 1001 \\ 1 \\ \hline 1010_{C2} = 10 \end{array}$$

$-6 \Leftrightarrow 2^4 - |-6|$

10

ex:

$5 = 0101$

$5 - 6 = 5 + (-6) = -1$

$$
\begin{array}{rcccc}
5 & 0 & 1 & 0 & 1 & + \\
-6 & 1 & 0 & 1 & 0 & \\
\hline
\text{cu semn} & 1 & 1 & 1 & 1 & = -1 \\
 & 0 & 0 & 0 & 0 & \\
 & & & & 1 & \\
\hline
 & & & & 1 &
\end{array}
$$

$$
\begin{array}{r}
5 \\
10 \\
\text{fără semn}
\end{array}
$$

11.10.22

Lab 1

Baze de numerație

$(abcdef)_2 = a\cdot 2^6 + b\cdot 2^5 + c\cdot 2^4 + d\cdot 2^3 + e\cdot 2^2 + f\cdot 2^1 + g\cdot 2^0$

$$\begin{array}{r} 567 \\ 183 \\ \hline 750 \end{array} +$$

baza 10: 0 - 9

baza 2: 0 - 1

bit: unitatea atomică de reprezentare a informației

$13 = 1101 = 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0$

$15 = 1111 = 1\cdot 2^3 + 1\cdot 2^2 + 1\cdot 2^1 + 1\cdot 2^0$

$= 1000 + 100 + 10 + 1$

$\text{val} = \sum_{i=0}^{n-1} b_i \cdot 2^i$

$78_{(10)} = ?_{(2)} = 1001110$

LSB

least —"—

MSB

most significant bit

$113_{(10)} = ?_{(2)}$

			c	r
113	: 2	=	56	1
56	: 2	=	28	0
28	: 2	=	14	0
14	: 2	=	7	0
7	: 2	=	3	1
3	: 2	=	1	1
1	: 2	=	0	1

$\Rightarrow 113_{(10)} = 1110001_{(2)}$

baza 16

0 - 0		8 - 8	
1 - 1		9 - 9	
2 - 2		10 - A	
3 - 3		11 - B	
4 - 4		12 - C	
5 - 5		13 - D	
6 - 6		14 - E	
7 - 7		15 - F	

$\underbrace{0(100}_{\substack{4\\4}}\underbrace{1110}_{\substack{14\\E}})_2 = (4E)_{16}$

$$
\begin{array}{l}
5 = 101 \ + \\
6 = 110 \\
\hline
1011_2
\end{array}
\qquad
\begin{array}{r}
11011101 \ + \\
00111010 \\
\hline
100010111_2
\end{array}
\qquad
\begin{array}{r}
1\cancel{1}0\cancel{1}010 \ - \\
101110 \\
\hline
0111100
\end{array}
$$

$$0\boxed{11}_2 = 3$$

$$1\boxed{11}_2 = -3$$

signed magnitude

$C(1) = \sim nr$ complement față de 1

$C(2) = \sim nr + 1$ complement față de 2

$$-17 = \boxed{1}10001 = \begin{array}{r} 001110 \ + \\ 1 \\ \hline 001111 \end{array} = 17$$

ex: $28_{(10)} - 15_{(10)}$

$28 : 2 = 14 \quad 0$

$14 : 2 = 7 \quad 0$

$7 : 2 = 3 \quad 1$

$3 : 2 = 1 \quad 1$

$1 : 2 = 0 \quad 1$

semn

$\boxed{0}11100 = 28_{(10)}$

$\boxed{0}1111 = 15_{(10)}$

$10001 = -15_{C2}$

padding

MSB

$$
\begin{array}{r}
011100 \\
110001 \\
\hline
\cancel{1}001101
\end{array} = 13
$$

overflow

Temă 1

1. 37, 81, 114, 25

a) binary conversion

b) hexa conversion

c) MSB, LSB for all no

2. 100110, 10101, 11110, 01100

3. $\begin{array}{r} 1011 + \\ 110 \end{array}$ $\quad$ $\begin{array}{r} 1101 + \\ 1110 \end{array}$

4. −37, −81, −114, −25 $\quad c(2) = ?$

5. a) 35 − 12

b) 42 − 21

Solutions

1. a) $37 = 2^5 + 2^2 + 2^0 = 100101$

$81 = 2^6 + 2^4 + 2^0 = 1010001$

$114 = 2^6 + 2^5 + 2^4 + 2^1 = 1110010$

$25 = 2^4 + 2^3 + 2^1 = 11001$

b) $00(100101)_2 = (25)_{16}$

$0(1010001)_2 = (51)_{16}$

$0(1110010)_2 = (72)_{16}$

$000(11001)_2 = (10)_{16}$

c) 11, 11, 10, 11

2. a) $100110 = 2^5 + 2^2 + 2^1 = 38$

$10101 = 2^4 + 2^2 + 2^0 = 21$

$11110 = 2^4 + 2^3 + 2^2 + 2^1 = 30$

$101100 = 2^5 + 2^3 + 2^2 = 44$

b) $00(100110)_2 = (26)_{16}$

$000(10101)_2 = (15)_{16}$

$000(11110)_2 = (1E)_{16}$

$00(101100)_2 = (16)_{16}$

c) 10, 11, 10, 10

3. a)
$$\begin{array}{r} 1011 + \\ 110 \\ \hline 010001 \end{array}$$

b)
$$\begin{array}{r} 1101 + \\ 1110 \\ \hline 011011 \end{array}$$

4. $c(2) = \sim nr + 1$

a) -37 $00100101 \sim 11011010 \overset{+1}{=} 11011011$

b) -81 $01010001 \sim 10101110 \overset{+1}{=} 10101111$

c) -114 $01110010 \sim 10001101 \overset{+1}{=} 10001111$

d) -25 $00011001 \sim 11100110 \overset{+1}{=} 11100111$

5. a) $35 = 2^5 + 2^1 + 2^0 = 100011$

- $12 = 2^3 + 2^2 = 1100$
- $001100 \sim 110011 \overset{+1}{=} 110100$
- $\begin{array}{r} 100011 + \\ 110100 \\ \hline \cancel{1}010111 \end{array}$

b) $42 = 2^5 + 2^3 + 2^1 = 101010$

- $21 = 2^4 + 2^2 + 2^1 = 10101$
- $010101 \sim 101010 \overset{+1}{=} 101011$

$\begin{array}{r} 101010 + \\ 101011 \\ \hline \cancel{1}010101 \end{array}$

Course 2

13.10.22

Algebra booleană
Porți logice

- variabile & operații
  - → true : 1
  - → false : 0

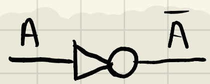
$\underbrace{\qquad\qquad}$

A, a, b, x, a1

PORȚI

a. negare / NOT / inversor : $\bar{A}$

$\bar{A}$

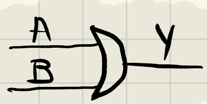


A	$\bar{A}$
0	1
1	0

tabelă adevăr

b. sumă / OR / SAU logic :

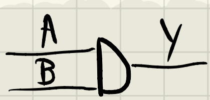
$Y = A + B$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

c. produs / AND / ȘI logic :

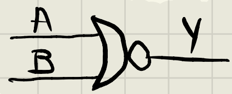
$Y = A \cdot B$



A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

d. SAU negat / NOR

$Y = \overline{A} + \overline{B}$

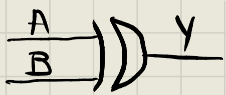


A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

e. SAU exclusiv / XOR

$Y = A \oplus B$

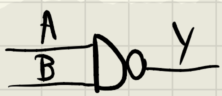
$= A \cdot \overline{B} + \overline{A} \cdot B$



A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

f. SI negat / NAND

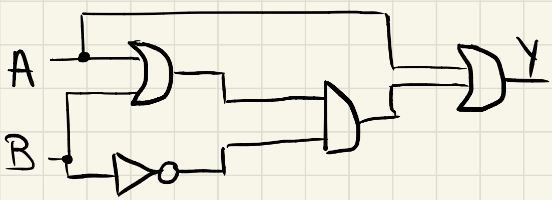
$Y = \overline{A \cdot B}$



A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

ex: $Y = (A+B) \cdot \overline{B} + A$

nu pot exista bucle



$Y = A \cdot \overline{B} + B \cdot \overline{B} + A$

$= A \cdot \overline{B} + 0 + A = A \cdot \overline{B} + A = A \cdot (\overline{B} + 1) =$

$= A \cdot 1 = A$

Teoreme / Legi

1. Identitate

$$A + 0 = A$$
$$A \cdot 1 = A$$

2. Elementul nul

$$A + 1 = 1$$
$$A \cdot 0 = 0$$

3. Idempotența

$$A + A = A$$
$$A \cdot A = A$$

4. Involuția

$$\overline{\overline{A}} = A$$

5. Complement

$$A + \overline{A} = 1$$
$$A \cdot \overline{A} = 0$$

6. Comutativitate

$$A \cdot B = B \cdot A$$
$$A + B = B + A$$

7. Asociativitate

$$A \cdot B + A \cdot C = A \cdot (B + C)$$

8. Distributivitate

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

9. Acoperire

$$A \cdot (A + B) = A$$

10. De Morgan

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$
$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

Tabelă adevăr → Circuit

- reprezentare cu termeni canonic sau sumă de produse

$Y = ?$ 3 intrări A, B, C

$f(A, B, C)$

A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

$\bar{A} \cdot \bar{B} \cdot \bar{C}$

$\bar{A} \cdot \bar{B} \cdot C$

$\bar{A} \cdot B \cdot \bar{C}$

$\bar{A} \cdot B \cdot C$

$A \cdot \bar{B} \cdot \bar{C}$

$A \cdot \bar{B} \cdot C$

$A \cdot B \cdot \bar{C}$

$A \cdot B \cdot C$

termeni canonici

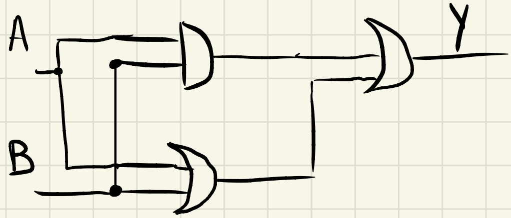
rez = 1

$$\overline{Y} = \bar{A} \cdot \bar{B} \cdot C + A \cdot \bar{B} \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$$

$$= \bar{B} \cdot (\bar{A} \cdot C + A \cdot \bar{C}) + A \cdot C \cdot (\bar{B} + B)$$

$$= \bar{B} \cdot (A \oplus C) + A \cdot C$$

ex :

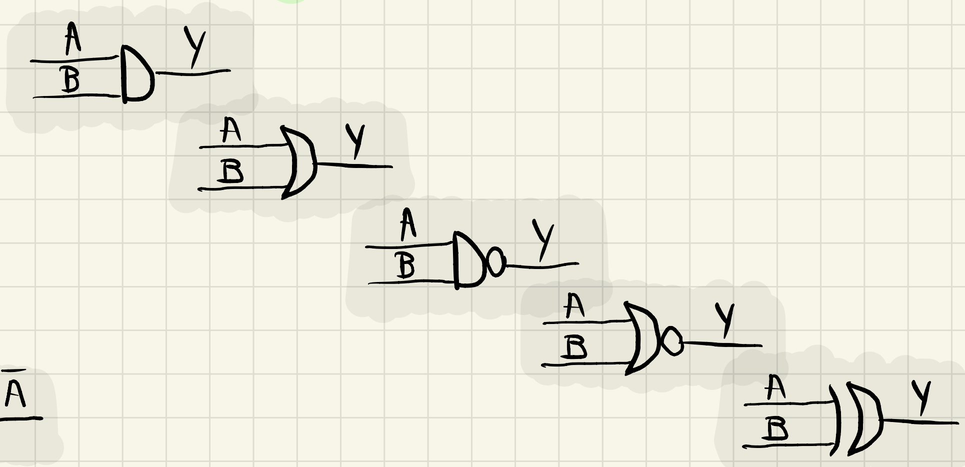


$$Y = (A \cdot B) + (A + B)$$

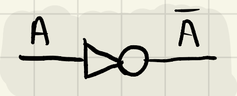
Lab 2

18.10.22

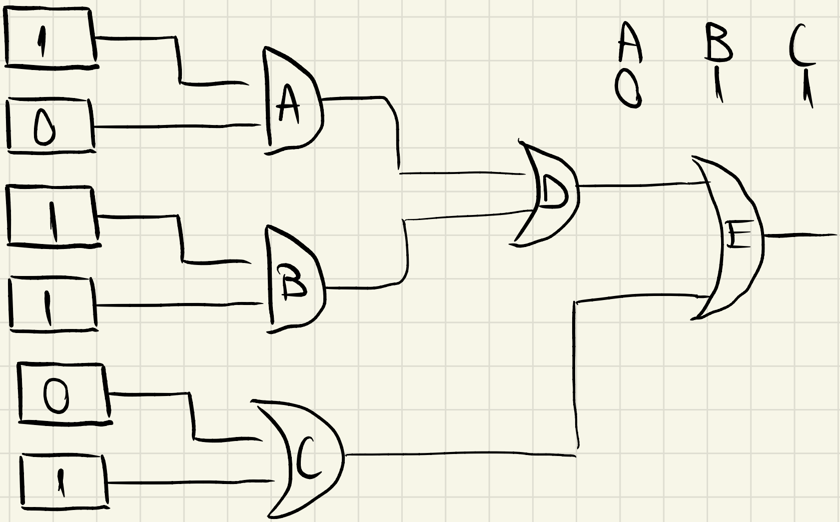
A	B	AND	OR	NAND	NOR	XOR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0



NOT



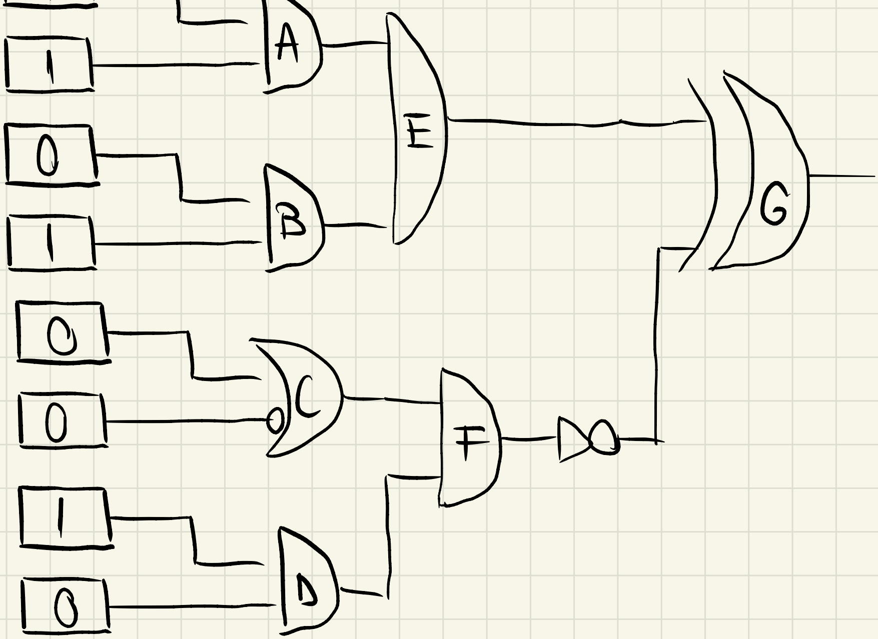
1.



A	B	C	D	F
0	1	1	1	1

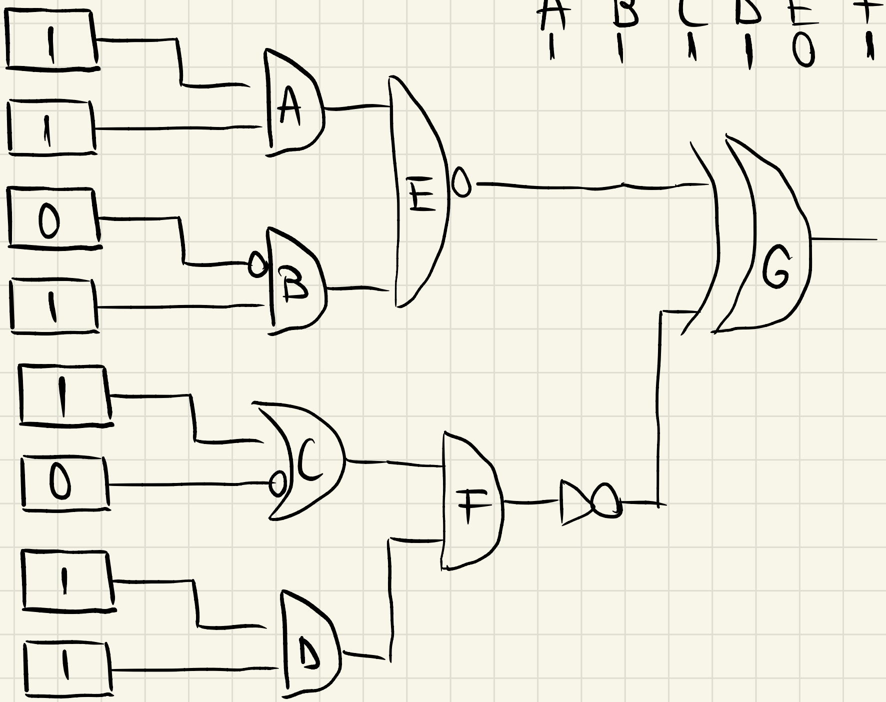
2.

A	B	C	D	E	F	G
1	0	1	0	0	0	1

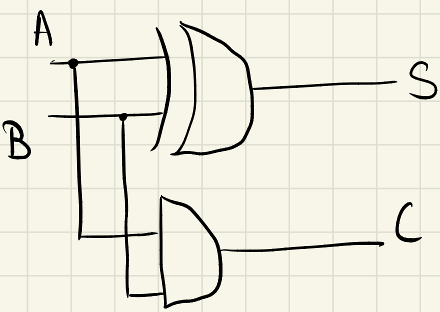


3.

A	B	C	D	E	F	G
1	1	1	1	0	1	0

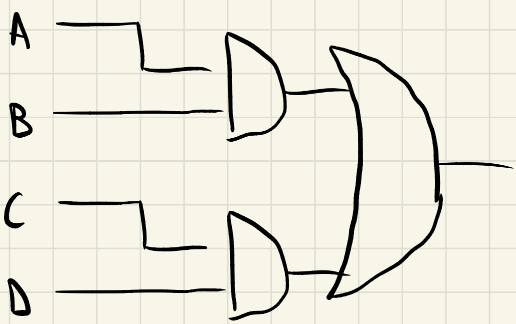


4.

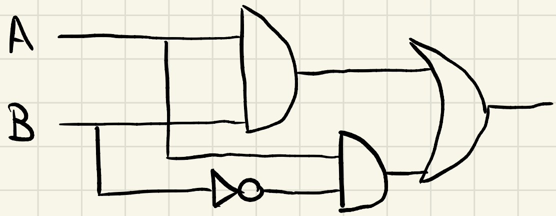


A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

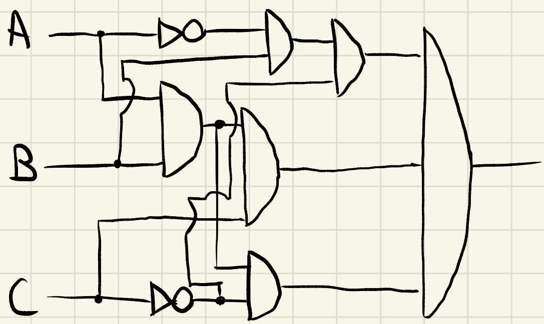
5. $F = A \cdot B + C \cdot D$



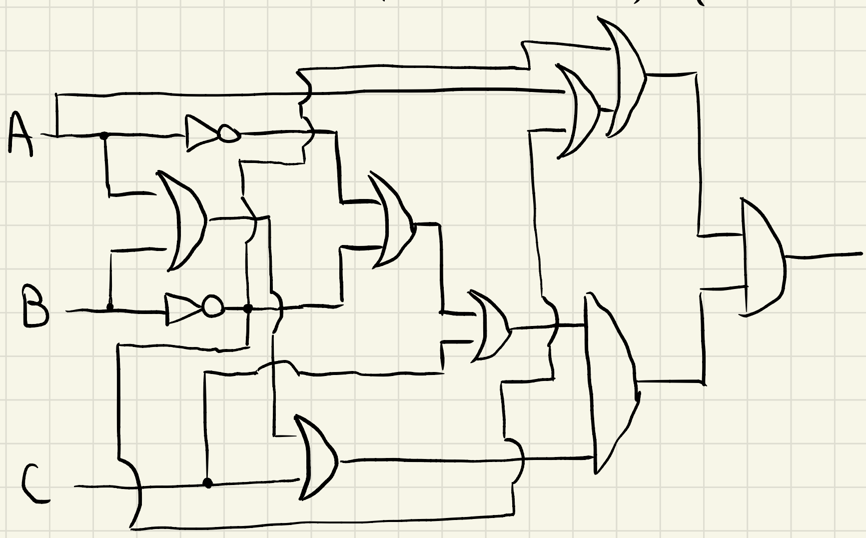
6. $F = A \cdot B + A \cdot \overline{B} = A \cdot (B + \overline{B}) = A$



7. $F = A \cdot B \cdot C + A \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot \overline{C}$



8. $F = (A + B + C) \cdot (\overline{A} + \overline{B} + C) \cdot (A + \overline{B} + C)$



9. $\sum_{A,B,C}(0,2,3,5,7)$

from here we take the ones

	A	B	C	Y	
0	0	0	0	1	$\bar{A}\cdot\bar{B}\cdot\bar{C} \rightarrow \bar{0}s$
1	0	0	1	0	
2	0	1	0	1	$\bar{A}\cdot B\cdot\bar{C}$
3	0	1	1	1	$\bar{A}\cdot B\cdot C$
4	1	0	0	0	
5	1	0	1	1	$A\cdot\bar{B}\cdot C$
6	1	1	0	0	
7	1	1	1	1	$A\cdot B\cdot C$

$$F = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot B\cdot\bar{C} + \bar{A}\cdot B\cdot C + A\cdot\bar{B}\cdot C + A\cdot B\cdot C$$

10. $F(A,B,C) = \sum(3,5,6,7)$

	A	B	C	Y	
0	0	0	0	0	
1	0	0	1	0	
2	0	1	0	0	
3	0	1	1	1	$\bar{A}\cdot B\cdot C$
4	1	0	0	0	
5	1	0	1	1	$A\cdot\bar{B}\cdot C$
6	1	1	0	1	$A\cdot B\cdot\bar{C}$
7	1	1	1	1	$A\cdot B\cdot C$

$$F = \bar{A}\cdot B\cdot C + A\cdot\bar{B}\cdot C + A\cdot B\cdot\bar{C} + A\cdot B\cdot C$$

11. $F(A,B,C) = \sum(0,1,2,4)$

	A	B	C	Y
0	0	0	0	1
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

$\bar{A}\cdot\bar{B}\cdot\bar{C}$

$\bar{A}\cdot\bar{B}\cdot C$

$\bar{A}\cdot B\cdot\bar{C}$

$A\cdot\bar{B}\cdot\bar{C}$

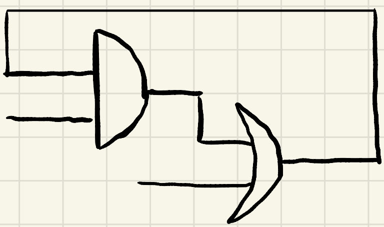
$$F = \bar{A}\cdot\bar{B}\cdot\bar{C} + \bar{A}\cdot\bar{B}\cdot C + \bar{A}\cdot B\cdot\bar{C} + A\cdot\bar{B}\cdot\bar{C}$$

Course 3

19.10.22

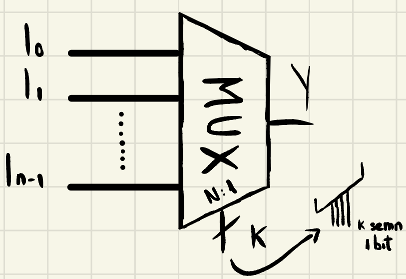
Circuite combinaționale

- semnal ieșire depinde de valorile curent de pe intrări, ex: porți logice
- nu există / sunt permise bucle combinaționale



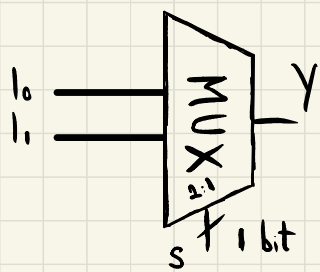
Multiplexor N : 1

- n intrări, 1 ieșire, semnal selecție s
  ↳ reprezentat pe k bits



$Y = I_s \quad N = 2^k$

ex: MUX 2:1



obs: MUX 8:1 ⇒ 3 bits

ex: MUX 2:1 cu porți logice

	I_0	I_1	S	Y	
0	0	0	0	0	I_0
1	0	0	1	0	I_0
2	0	1	0	0	I_0
3	0	1	1	1	I_1
4	1	0	0	1	I_0
5	1	0	1	0	I_0
6	1	1	0	1	I_1
7	1	1	1	1	I_1

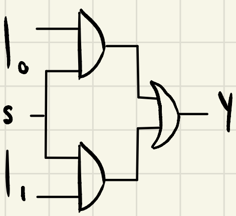
alegem "I_s"

suma de produse

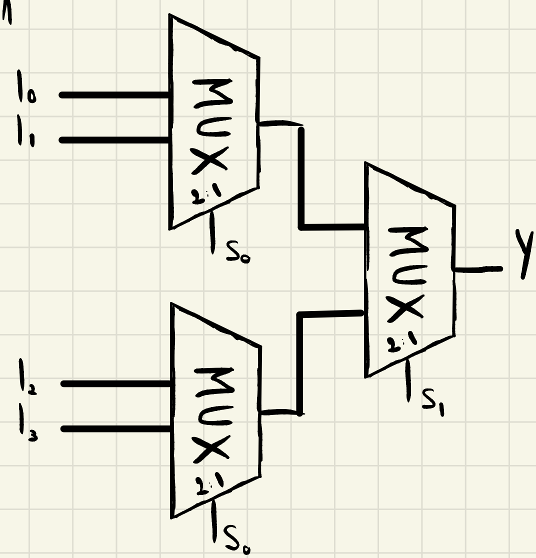
$$Y = \bar{I_0} I_1 s + I_0 \bar{I_1} \bar{s} + I_0 I_1 \bar{s} + I_0 I_1 s$$

$$= I_1 \cdot S \cdot \underbrace{(\bar{I_0} + I_0)}_{1} + I_0 \cdot \bar{S} \cdot \underbrace{(\bar{I_1} + I_1)}_{1}$$

$$= I_1 s + I_0 \bar{s}$$



ex: MUX 4:1



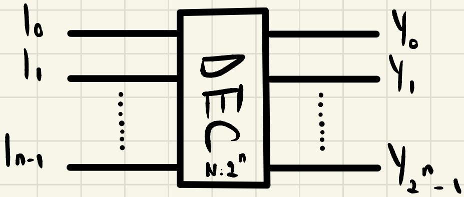
Extra

S_0	S_1	I_0	I_1	I_2	I_3	Y	
0	0	0	0	0	0	0	I_0
0	0	0	0	0	1	0	
0	0	0	0	1	0	0	
0	0	0	0	1	1	0	
0	0	0	1	0	0	0	
0	0	0	1	0	1	0	
0	0	0	1	1	0	0	
0	0	0	1	1	1	0	
0	0	1	0	0	0	1	
0	0	1	0	0	1	1	
0	0	1	0	1	0	1	
0	0	1	0	1	1	1	
0	0	1	1	0	0	1	
0	0	1	1	0	1	1	
0	0	1	1	1	0	1	
0	0	1	1	1	1	1	
0	1	0	0	0	0	0	I_1
0	1	0	0	0	1	0	
0	1	0	0	1	0	0	
0	1	0	0	1	1	0	
0	1	0	1	0	0	1	
0	1	0	1	0	1	1	
0	1	0	1	1	0	1	
0	1	0	1	1	1	1	
0	1	1	0	0	0	0	
0	1	1	0	0	1	0	
0	1	1	0	1	0	0	
0	1	1	0	1	1	0	
0	1	1	1	0	0	1	
0	1	1	1	0	1	1	
0	1	1	1	1	0	1	
0	1	1	1	1	1	1	
1	0	0	0	0	0	0	I_2
1	0	0	0	0	1	0	
1	0	0	0	1	0	1	
1	0	0	0	1	1	1	
1	0	0	1	0	0	0	
1	0	0	1	0	1	0	
1	0	0	1	1	0	1	
1	0	0	1	1	1	1	
1	0	1	0	0	0	0	
1	0	1	0	0	1	0	
1	0	1	0	1	0	1	
1	0	1	0	1	1	1	
1	0	1	1	0	0	0	
1	0	1	1	0	1	0	
1	0	1	1	1	0	1	
1	0	1	1	1	1	1	

Decodificator

- n intrări, 2^n ieșiri

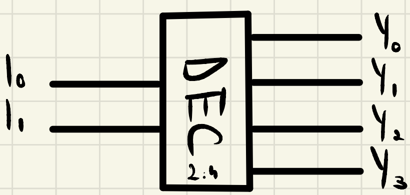
Def: activează ieșirea cu indicele = valoarea zecimală de pe cele n intrări



$Y_X = 1$, $X = I_{n-1} \dots I_1 I_0$

$Y_{K \neq X} = 0$

ex: DEC 2:4



	I_0	I_1	Y_0	Y_1	Y_2	Y_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

Temă: prob. DEC material

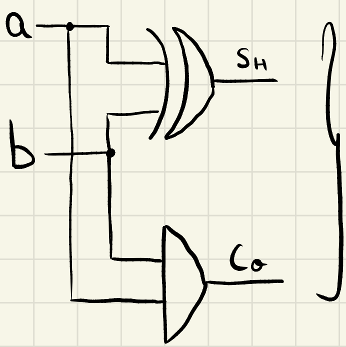
Sumator

- 1 bit
  - intrările a, b
  - ieșirea S_H
  - cât C_o
  - carry in C_{in}

	a	b	S_H	$C_o = Y$
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

$$S_H = \bar{a}b + a\bar{b} = a \oplus b$$

$$C_o = a \cdot b$$



semisumator

a	b	C_{in}	S_H	S	$C_o = Y$
0	0	0	0	0	0
0	1	0	1	1	0
1	0	0	1	1	0
1	1	0	0	0	1
0	0	1	0	1	0
0	1	1	1	0	1
1	0	1	1	0	1
1	1	1	0	1	1

$$S = C_{in} \oplus S_H$$

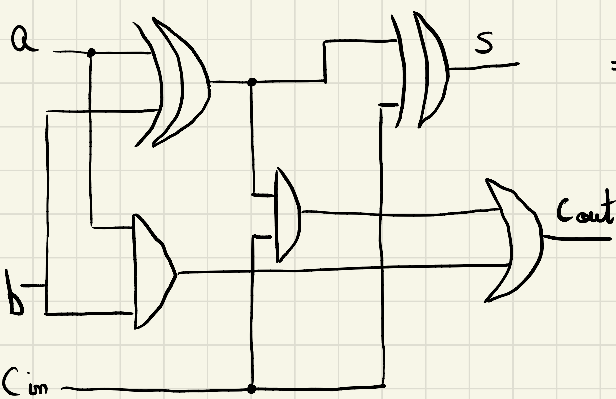
$$= C_{in} \oplus a \oplus b$$

$$C_o = ab\bar{c}_{in} + \bar{a}bc_{in} +$$

$$+ a\bar{b}c_{in} + abc_{in}$$

$$= ab\cancel{(c_{in} + \bar{c}_{in})} + c_{in}(\bar{a}b + \bar{b}a)$$

$$= ab + c_{in}(a \oplus b)$$



Temă

DEC 2:4

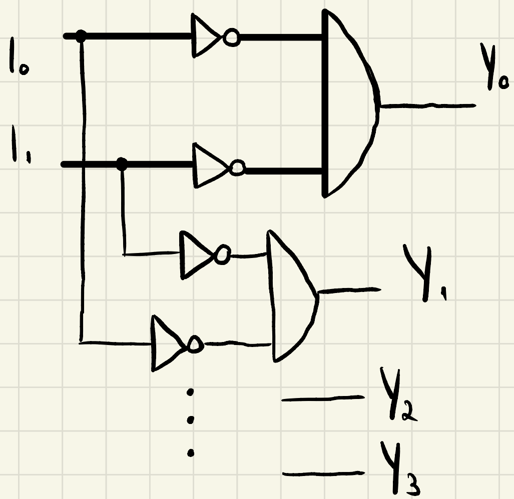
	I_0	I_1	val index	Y_0	Y_1	Y_2	Y_3
0	0	0	0	1	0	0	0
1	0	1	1	0	1	0	0
2	1	0	2	0	0	1	0
3	1	1	3	0	0	0	1

$Y_0 = \bar{I_0} \cdot \bar{I_1}$

$Y_1 = \bar{I_0} \cdot I_1$

$Y_2 = I_0 \cdot \bar{I_1}$

$Y_3 = I_0 \cdot I_1$



Lab 3

25.10.22

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
:
entity top is
  Port ( btn : in STD_LOGIC;
         sw : in STD_LOGIC_VECTOR (15 down to 0);
         led : out STD_LOGIC_VECTOR (15 down to 0);
end top;
architecture Behavorial of top is
```

bus de semnale

= declarăm semnale / variabile intermediare

```
SIGNAL ISA1: STD_LOGIC;
SIGNAL ISA2: STD_LOGIC;
begin
```

= descriem comportamentul circuitului prezent

```
ISA1 <= sw(1) and sw(0);
ISA2 <= sw(2) and sw(3);
led(1) <= ISA1 or ISA2 ;
```

Temă:

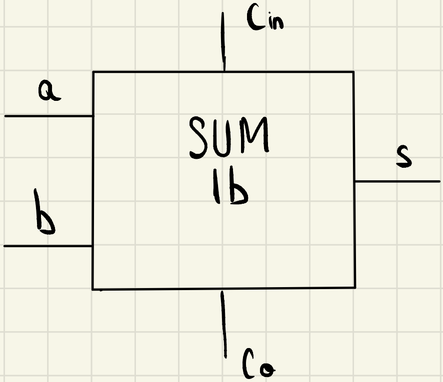
carte FPGA, p. 1-47
anexa A, p. 225-253
DOCX

steps

1. generate bitstream
2. open target
3. program device

Course 4

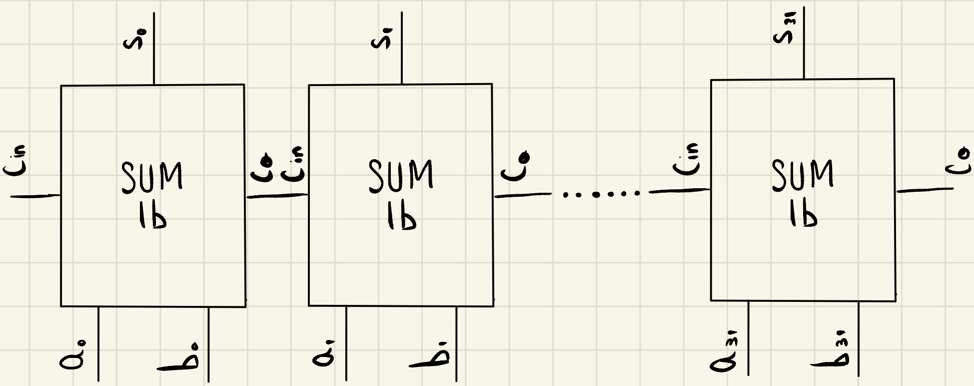
27.10.22



$$\begin{array}{cccccc} a_{31} & a_{30} & \dots & a_1 & a_0 & + \\ b_{31} & b_{30} & \dots & b_1 & b_0 & \\ \hline s_{31} & s_{30} & \dots & s_1 & s_0 & \end{array}$$

n biti -> n sumatoare

Sumator complet

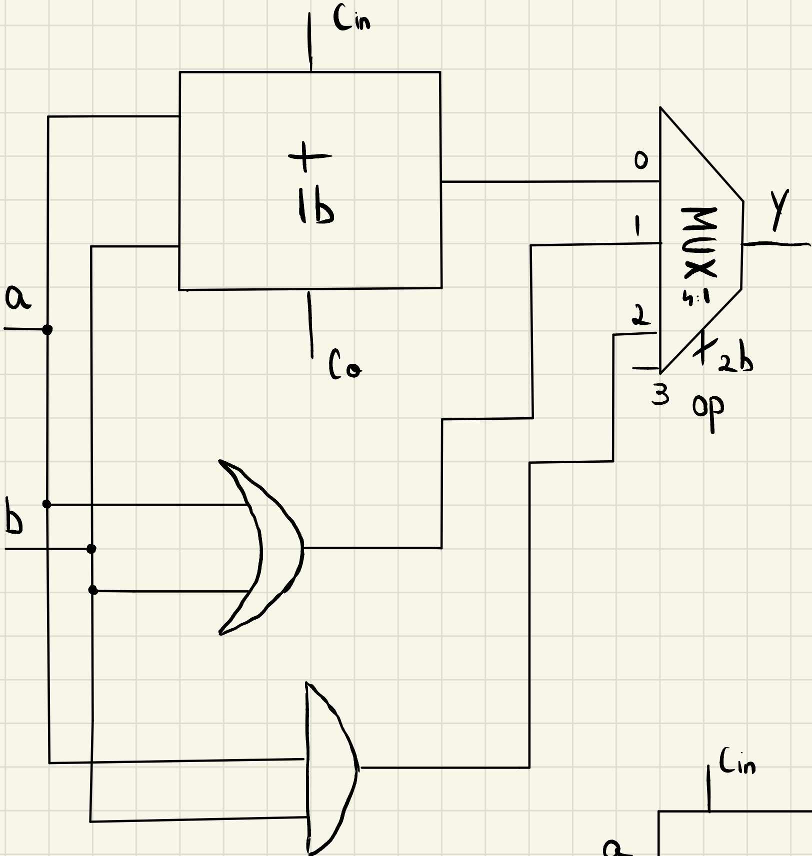


Unitate aritmetică - logică

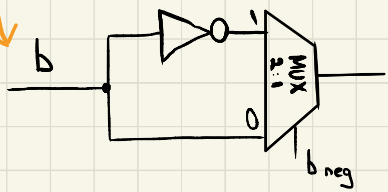
+, -   SAU, ȘI

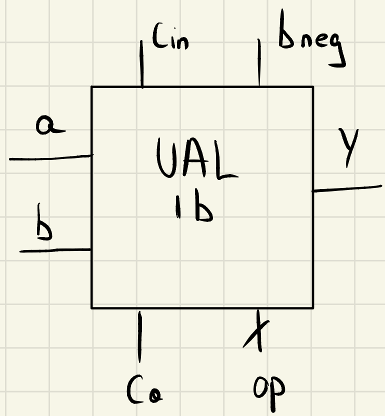
- combinațională
- 1 UAL pe 1bit cu 2 intrări a, b

  +, SAU, și   1 ieșire y



op	
0	+
1	SAU
2	SI
3	





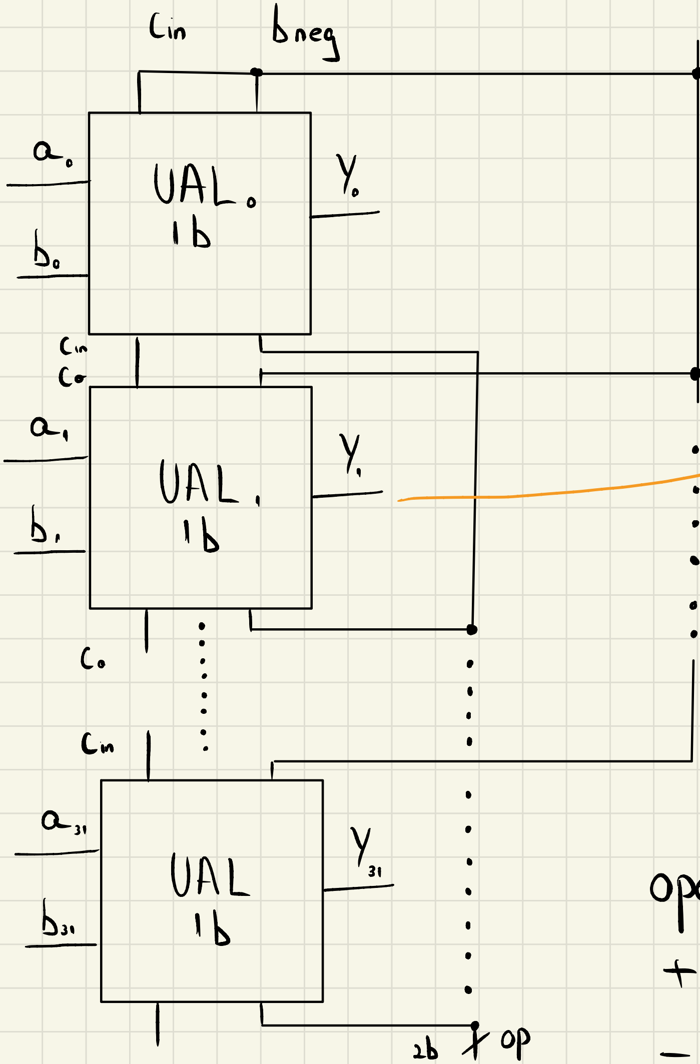
- 1 UAL → +, −, SAU, SI

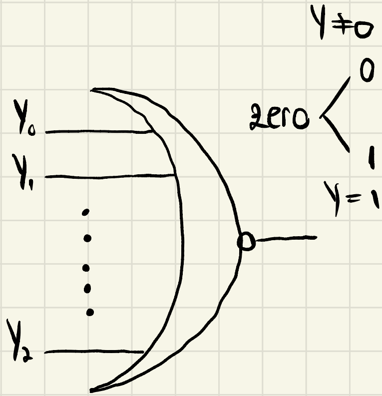
$$a-b = a+(-b) \overset{+1}{=} a+\bar{b}+1$$

	1 bit	
fără semn	0	1
cu semn	0	1 $\Leftrightarrow -1_{(10)}$

• UAL 32b

+, −, SAU, ŞI



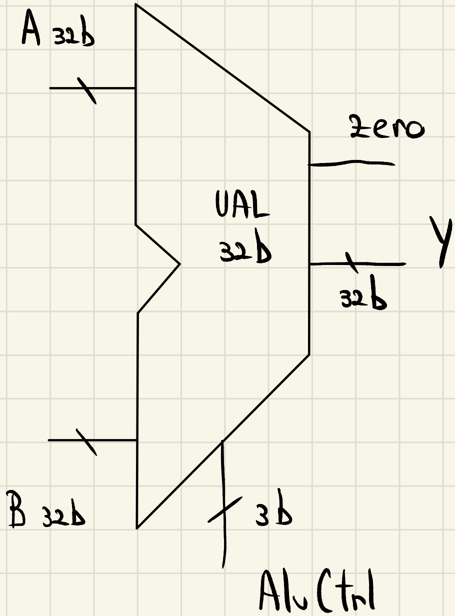


Detecție de zero

$$Op = Op_1 \, Op_0$$

operație (UAL)	bneg	Op_{10}
+	0	00
−	1	00
SAU	0	01
ŞI	0	10

$AluCtrl_{2,1,0}$

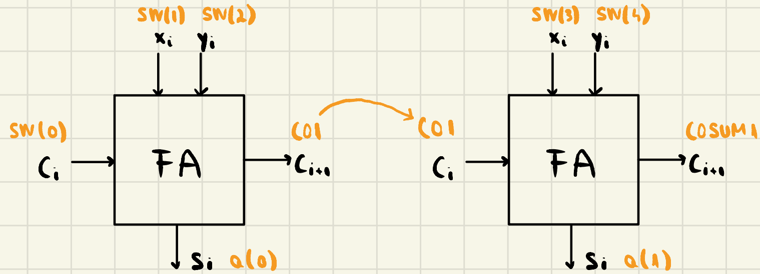


O_p = reprezentare binară pe biți a indexului MUX

Lab 4

1.11.22

1.



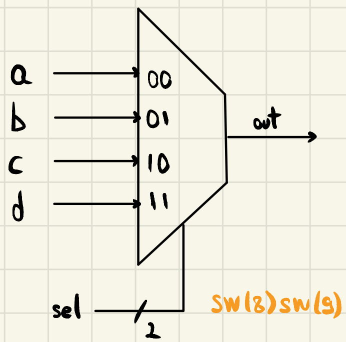
$S_i = x_i \text{ XOR } y_i \text{ XOR } c_i$

$C_{i+1} = x_i \text{ AND } y_i \text{ OR } (x_i \text{ OR } y_i) \text{ AND } c_i$

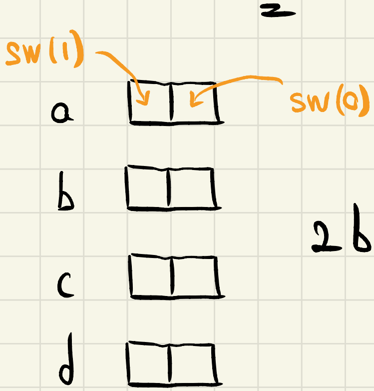
see code_notes for FA

2. MUX : a. SUM c. SUM constant

b. SR d. NAND



see code_notes for MUX



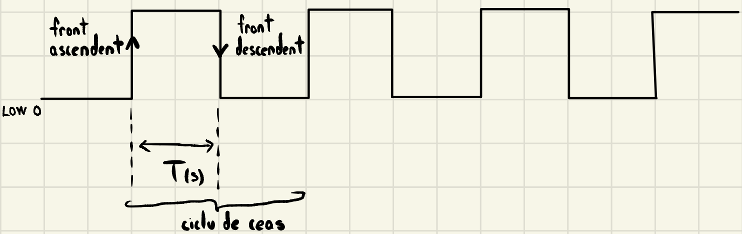
2 bits → 1 DOWNTO 0

Course 5

3.11.2022

Circuite secvențiale
(sincron)

- semnal de ceas
- semnal de ieșire <=> stare
- modificarea stării e dată de semnale de ceas

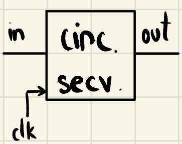


$$\text{frecv} = \frac{1}{T} \quad \text{u.m. Hz}$$

$$1\ \text{GHz} = 10^{9}\ \text{Hz}$$

$$T = \frac{1}{10^{9}\ \text{Hz}} = 10^{-9}\ \text{s}$$
$$= 1\ \text{ns}$$

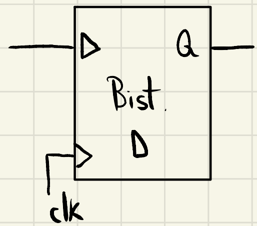
- regula de propagare



- se memorează pe fr. asc.
- val stabilă de pe intrare înainte de fr. cresc. se propagă pe semnalul de ieșire

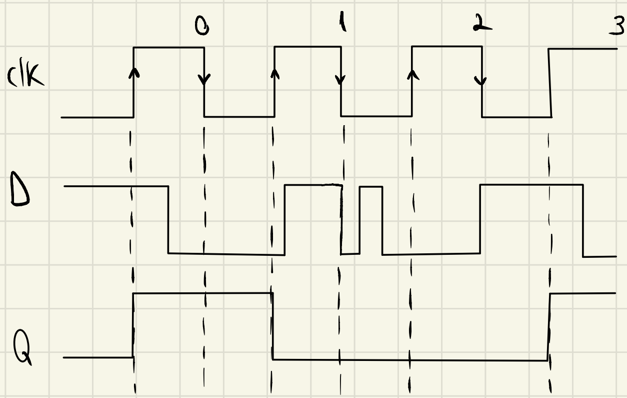
Bistabil D sincron cu activare pe ↑
(scriere)

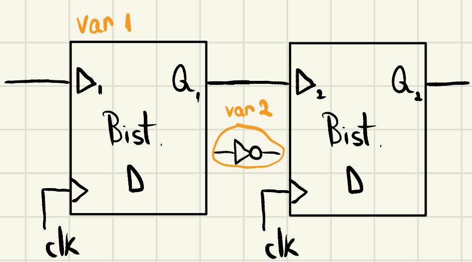
- 1 bit

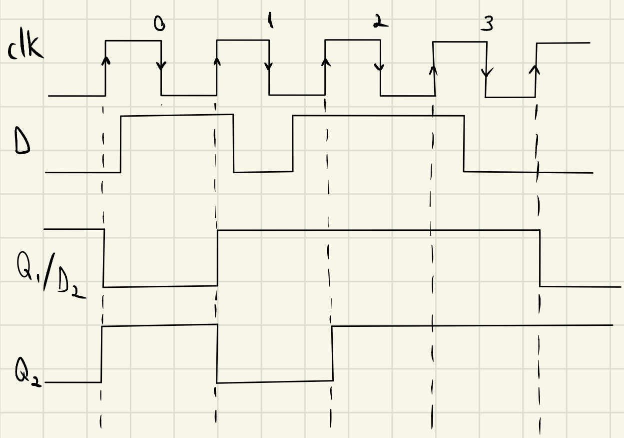


D	clk	Q_{in}	Q
x	0	nu se mod.	
x	1	nu se mod.	
0	↑	x	0
1	↑	x	1

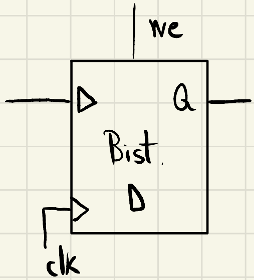
x - nu contează





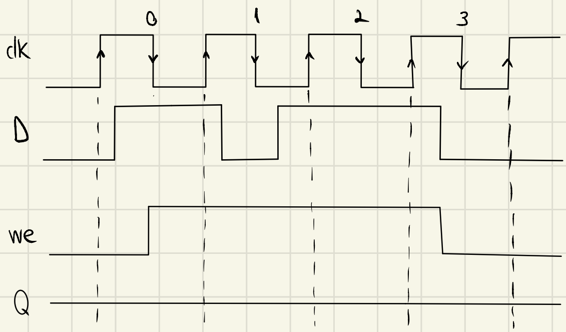


Bistabil D sincron cu semnal de validare a scrierii

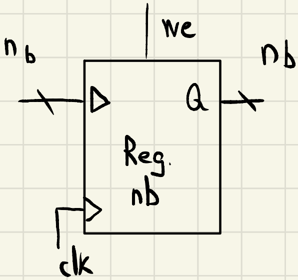


we - write enable

scrierea are loc ↑ și we = 1



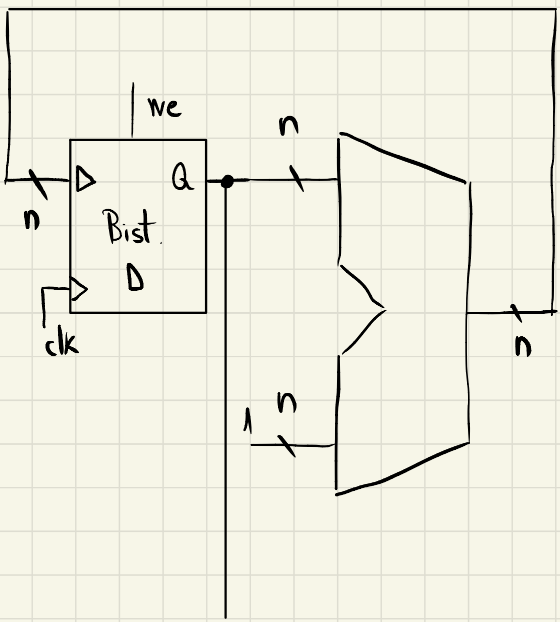
=> Registru pe n biți format din n bistabile D



Course 6

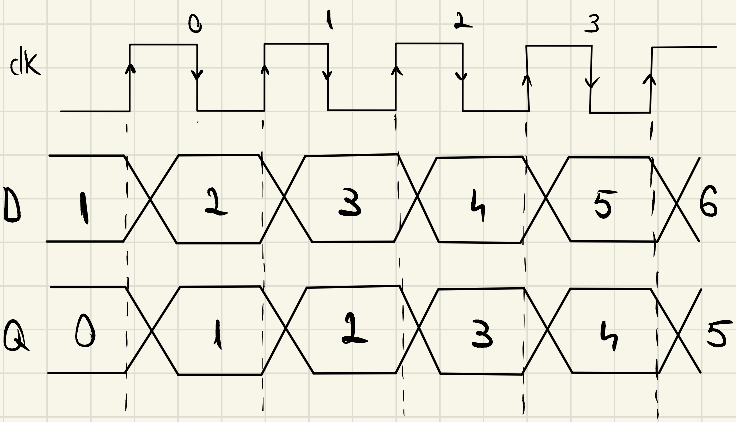
10.11.22

Un circuit numărător



pp. initial

$T_0 \quad Q = 0$



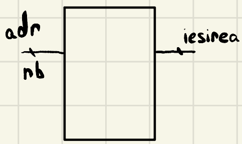
Circuite de memorie

- RAM: random access memory → volatilă
- ROM: read-only memory → non-volatilă, pt. citire

RAM → acces aleatoriu

SRAM, DRAM

elemente (date)
adrese



n biți ⇒ 2^n elem
cu index $0, \dots, 2^n-1$

1 bit adresă ⇒ 2^1 el
2 — " — ⇒ 4 el
32 — " — ⇒ 2^{32} el

elementul — octet

- 1024 octeți (10 adrese)

0 8b
1 8b = 1 kilobyte

1023 8b

1 octet = 1 byte
1 KB = 2^{10} bites / B
1 MB = 2^{10} KB = 2^{20} B
1 GB = 1024 MB = 2^{30} B
4GB = $2^2 \cdot 2^{30}$ B

4 GB adresabilă pe octet

organizată pe cuvânt 4B

adresă date

0 000 ... 0000 8b 8b 8b 8b

4 000 ... 0100 8b 8b 8b 8b $= 4$ GB

8 000 ... 1000 ⋮ $= 1$ Giga cuvinte

$= 2^{30}$ cuvinte

$2^{32}-4$ 111 ... 1100 8b 8b 8b 8b

Course 7

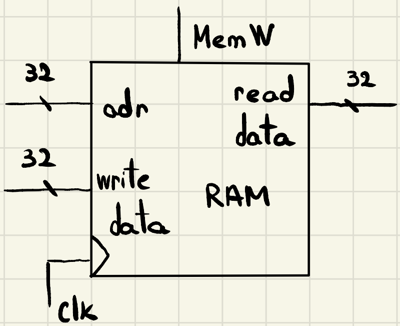
17.11.22

Ex. de circuite folosite în proc. MIPS 32

1. Memorie RAM

4 GB, 32 b adresă

32 b elementul citit din memorie

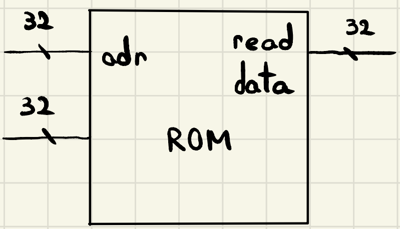


a. citirea : asincronă / combin.

b. scrierea: sincronă ↑ & MemW = 1

WD se memorează în RAM la indexul de pe semnalul adr.

2. Memorie ROM

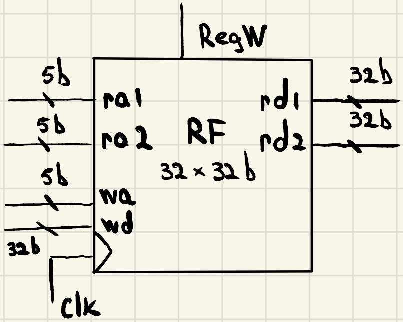


4 GB, 32 b adresă

a. citire : comb.

3. Bloc de Reg. RF (Reg. File)

32 reg. × 32 b (32 variabile posibile)



↳ 2 citiri simultane

1 scriere sincronă ↑ & RegW = 1 la adresa wa

RF[wa] ⇐ wd

32 registre: $r_0, r_1, \dots, r_{31}$ (RF[0], RF[1], RF[31])

Spre conceptul de proc.

- instrucțiuni / set instrucțiuni
  - etichetă, operanzi → limbaj de ansamblare
  - cod mașină / binar

- 32 reg, 32 b → RF 32 × 32 b, $r_0, r_1, \dots, r_{31}$
- 4 instrucțiuni +, −, *, / (ADD, SUB, MUL, DIV)
- nume_inst r_d, r_{s1}, r_{s2} ← registrii

1. $Y = a * b + c$

presupunem
$a \leftrightarrow r_5$
$b \leftrightarrow r_6$
$c \leftrightarrow r_{31}$
$Y \leftrightarrow r_5$

mul r_5, r_5, r_6

add r_5, r_5, r_{31}

2.

operație	rd	rs1	rs2		
2b (4 operații)	5b	5b	5b	$\rightarrow$	17b

	opcod
add	00
sub	01
mul	10
div	11

Course 8

24.11.2022

Analiză & cale de date

program => cod binar / mașină — 1 instr./ciclu de ceas

↳ memorie de instrucțiuni

ROM ? elem (256) ? dimensiune element (17 b)

? cum se parcurge prog < 1 reg.-contor progr (PC) 8b; sumator

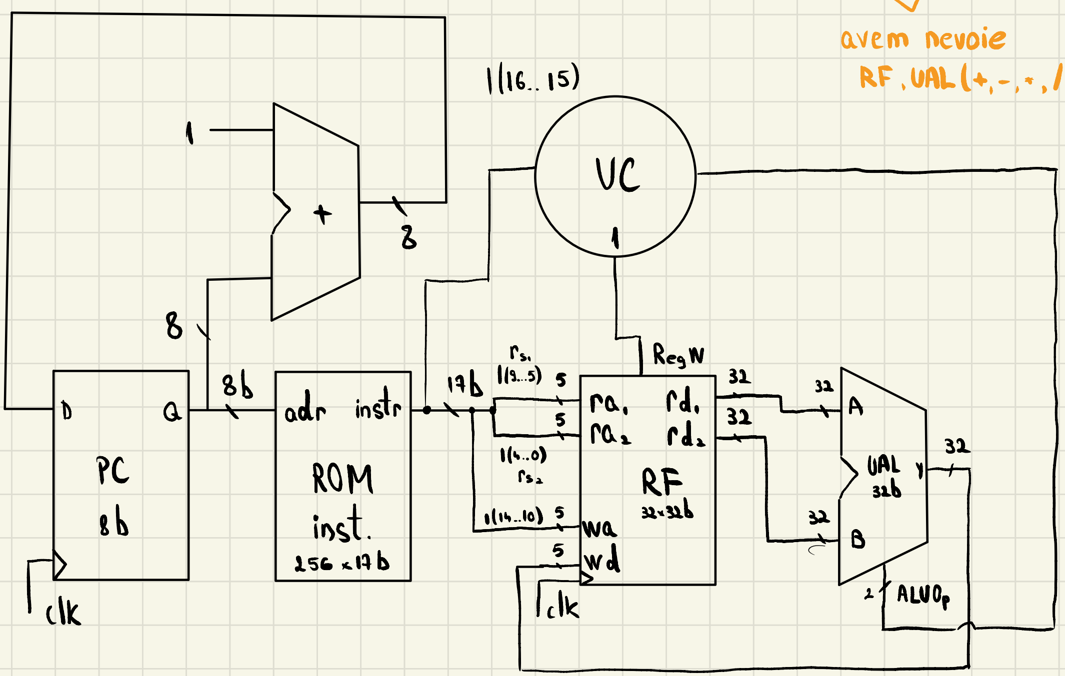
op rd, rs_1, rs_2

RF [rd] ← RF [rs_1] op RF[rs_2] → RTL (register transfer level)

2b	5b	5b	5b
opcod	rd	rs_1	rs_2
I(16...15)	I(14...10)	I(9...5)	I(4...0)

⇓

avem nevoie RF, UAL (+, -, *, /)



instr	opcode	UAL	AluOp
add	00	+	00
sub	01	−	01
mul	10	*	10
div	11	/	11

instr	AluOp	RegW
add	00	1
sub	01	1
mul	10	1
div	11	1

MIPS → RF[0] = 0   ex : add r_0, r_0, r_0
NOOP = no operation