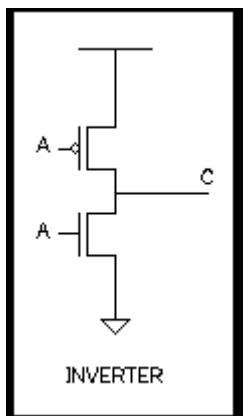


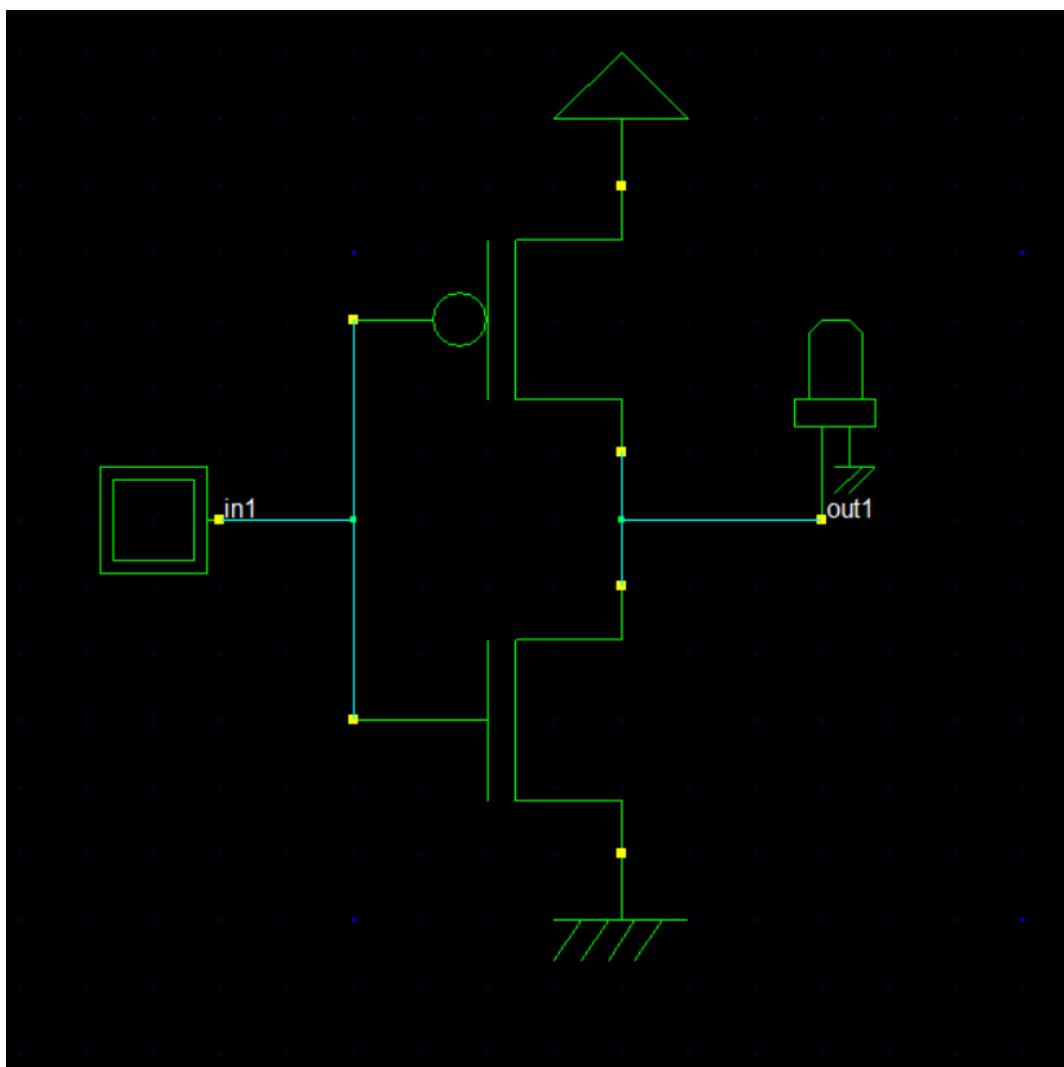
REFERAT VLSI

1. Inversor

- schema circuit

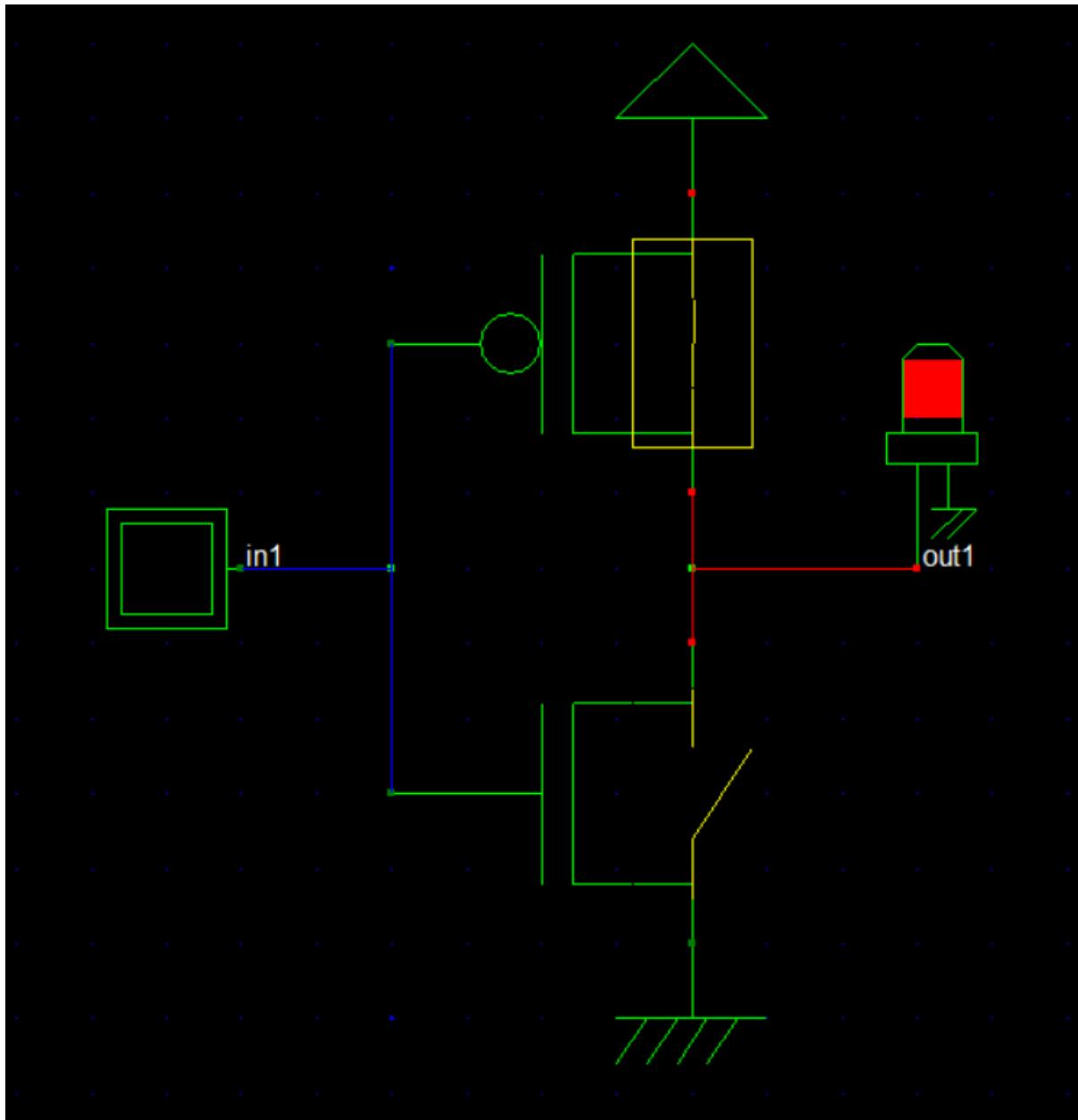


- Schema DSCH

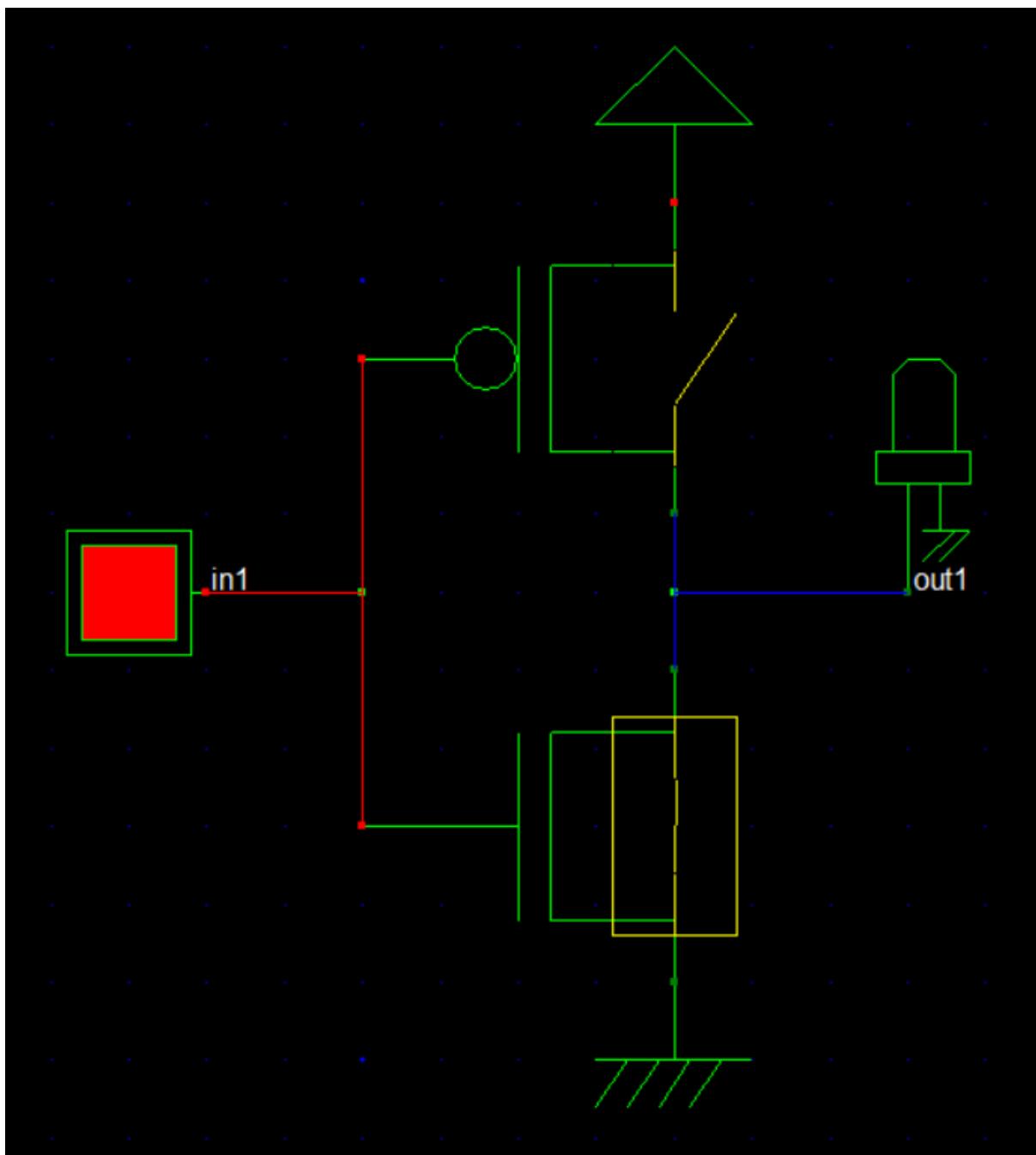


- Simulare DSCH

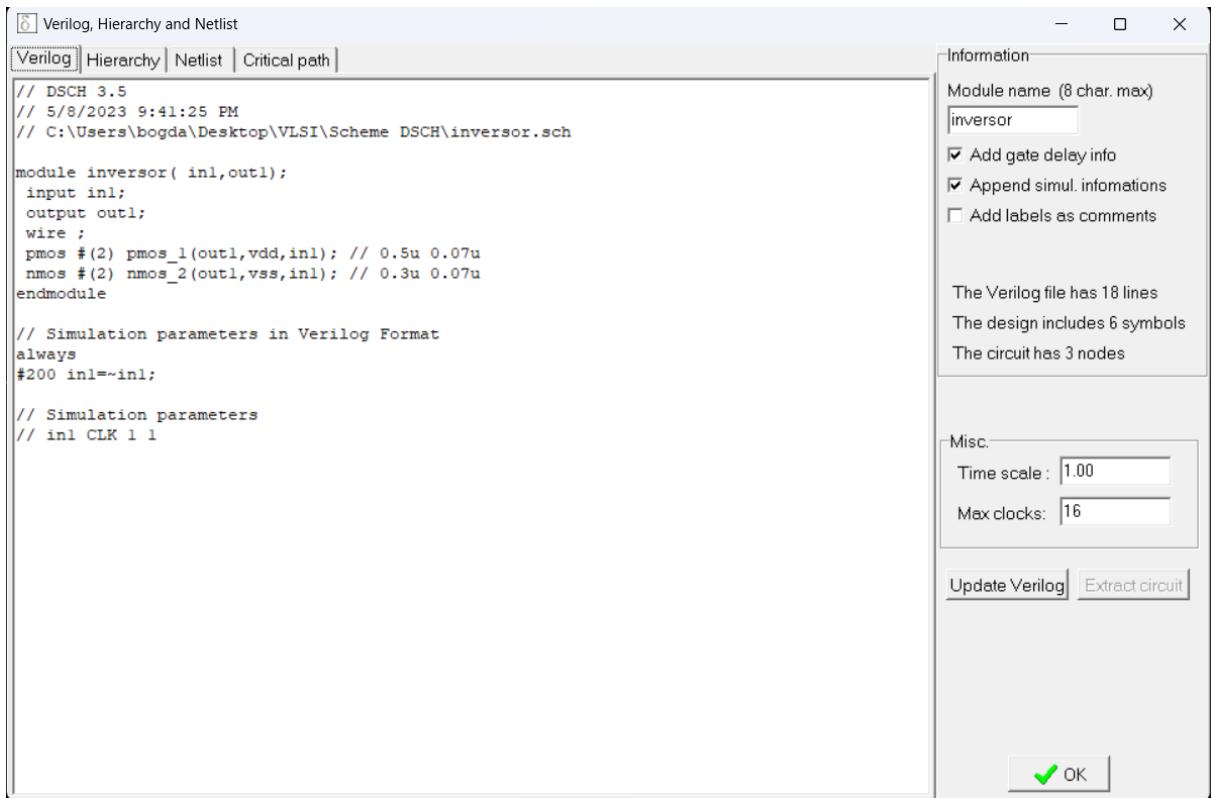
Pentru input = 0 => out = 1



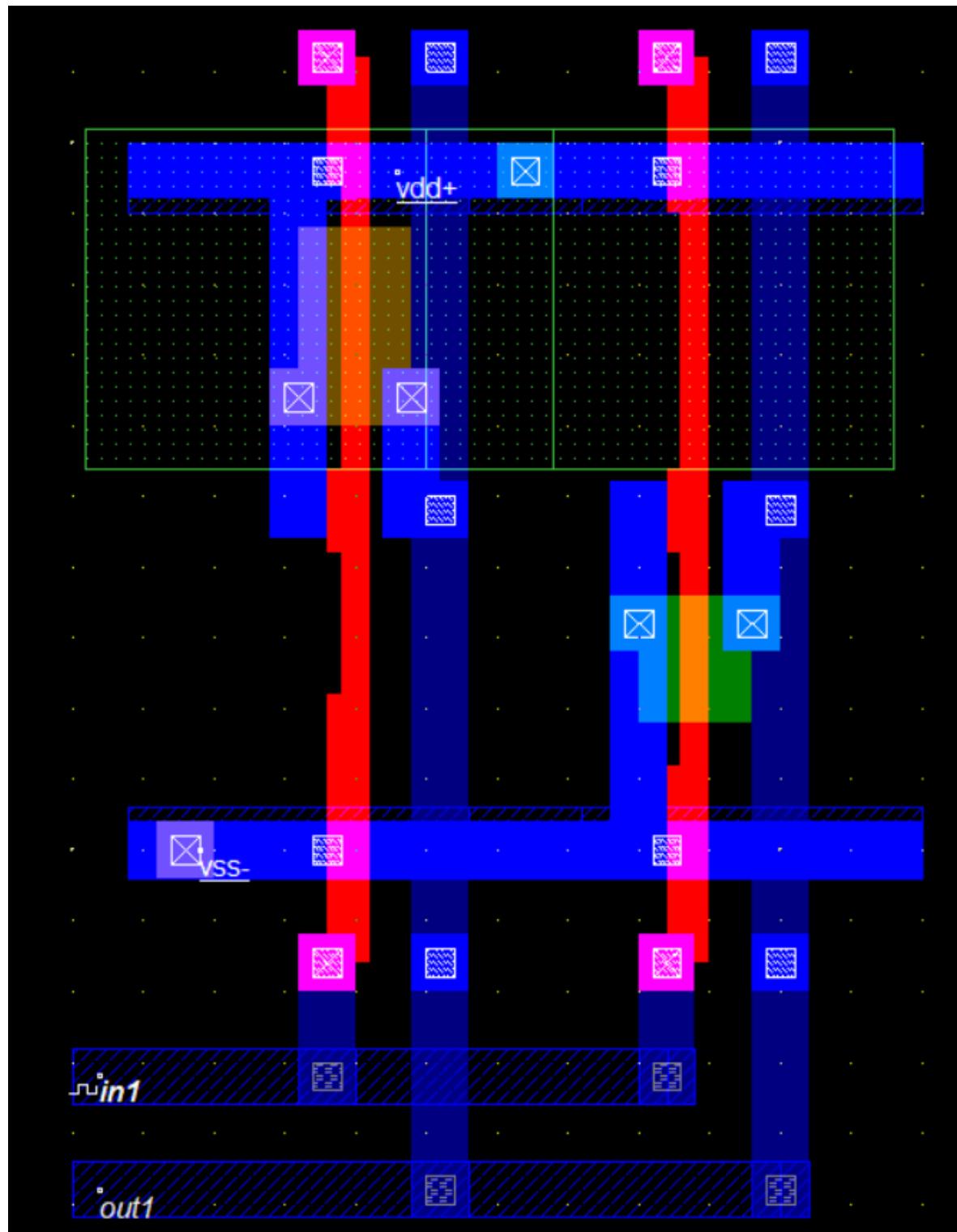
Pentru input = 0 => output = 1



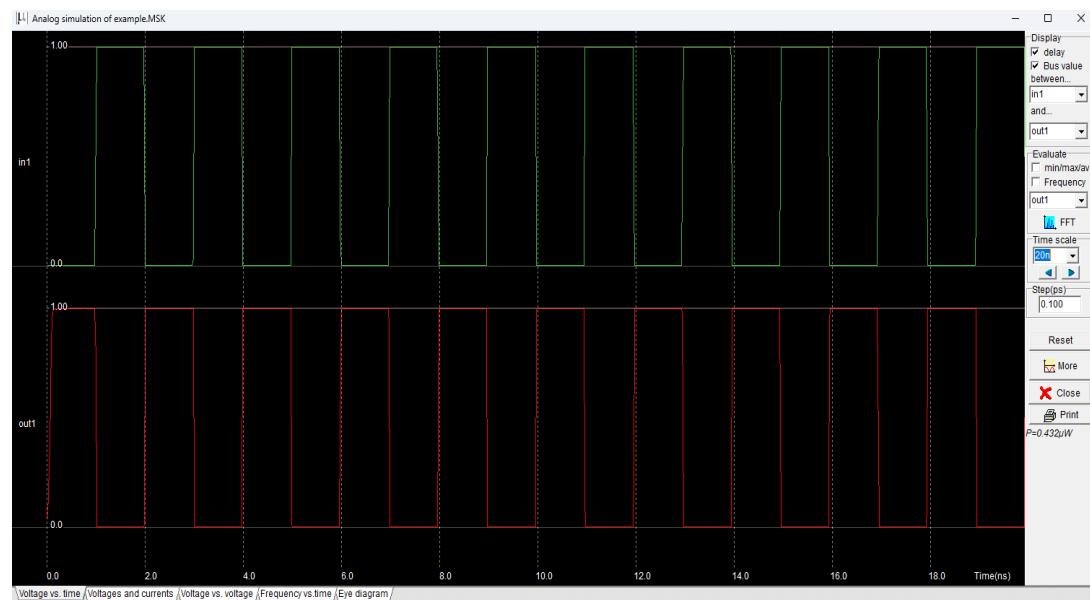
- Cod Verilog



- Microwind

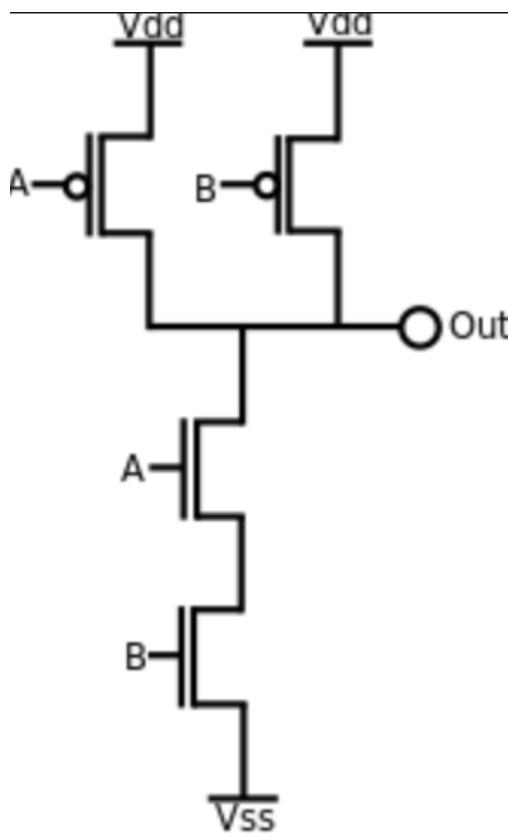


- Simulare Microwind



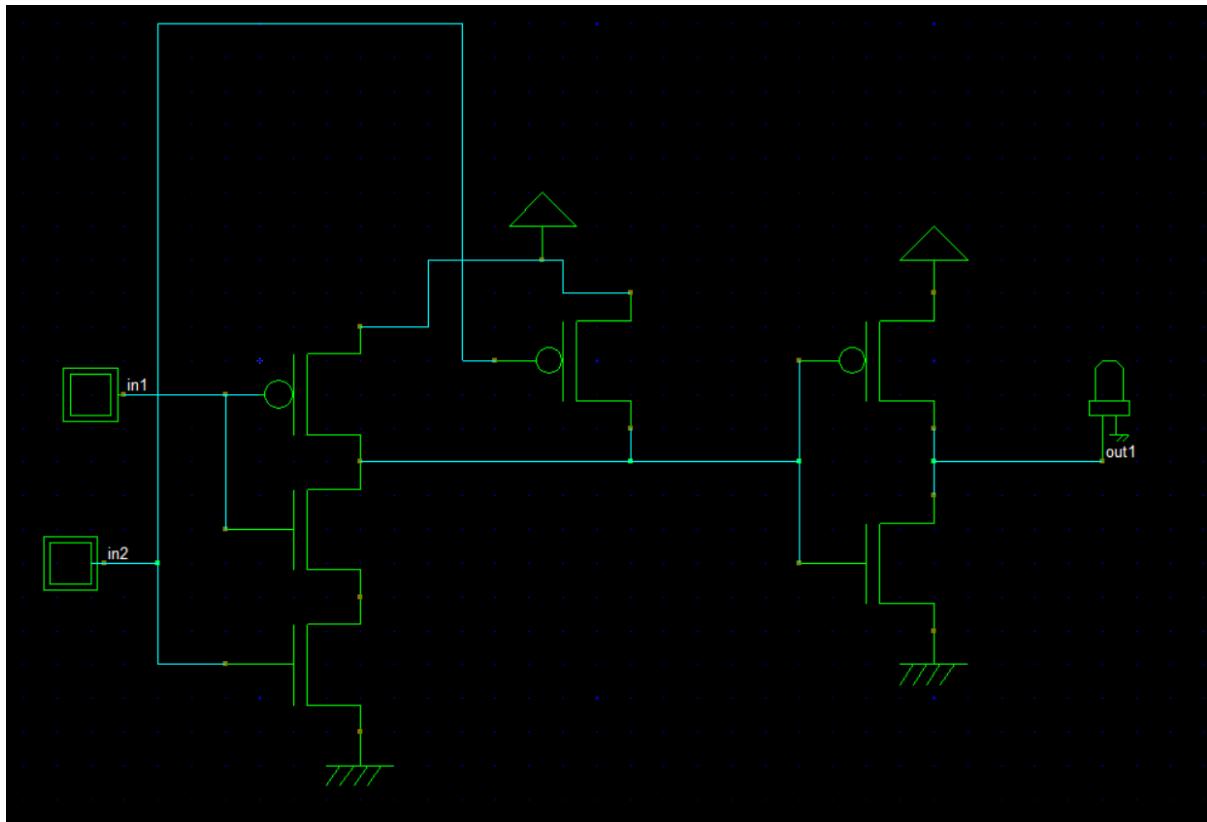
2. AND GATE

- schema circuitului

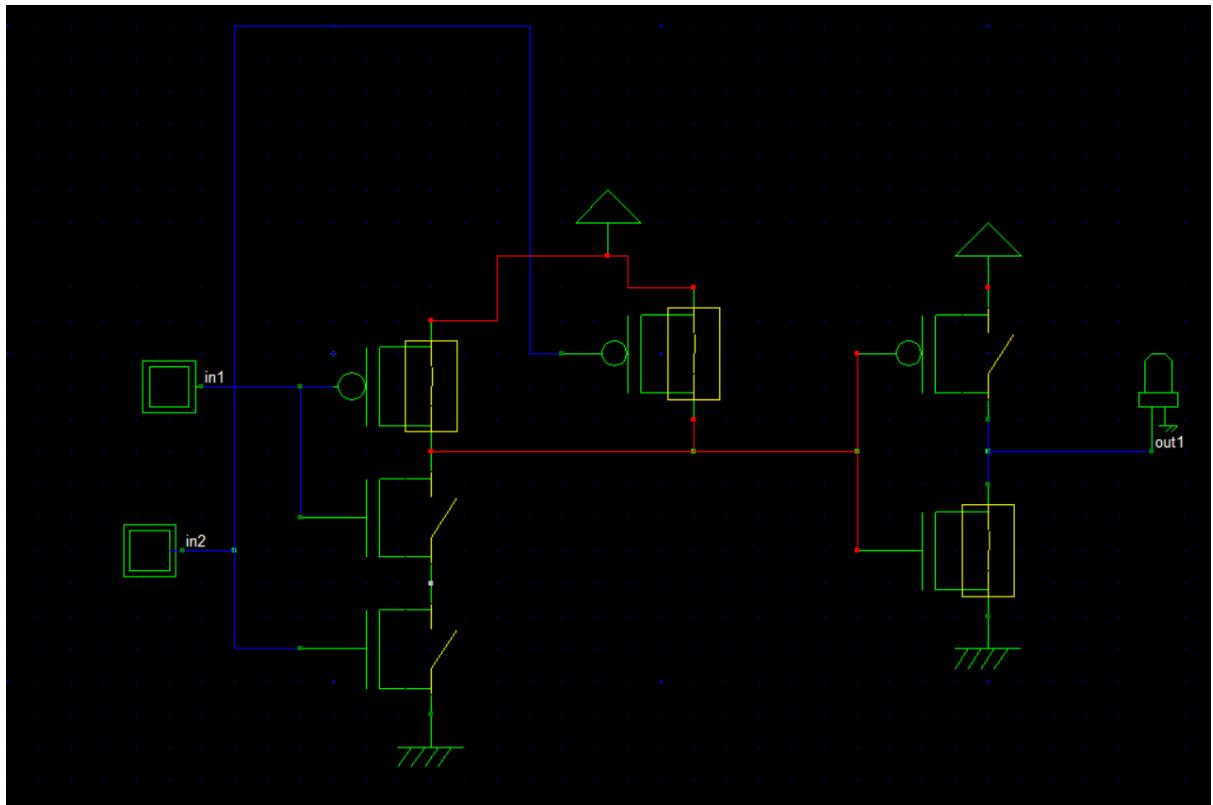


<https://electronics.stackexchange.com/questions/516929/why-is-the-pmos-in-nand-gate-in-p-parallel-and-nmos-series>

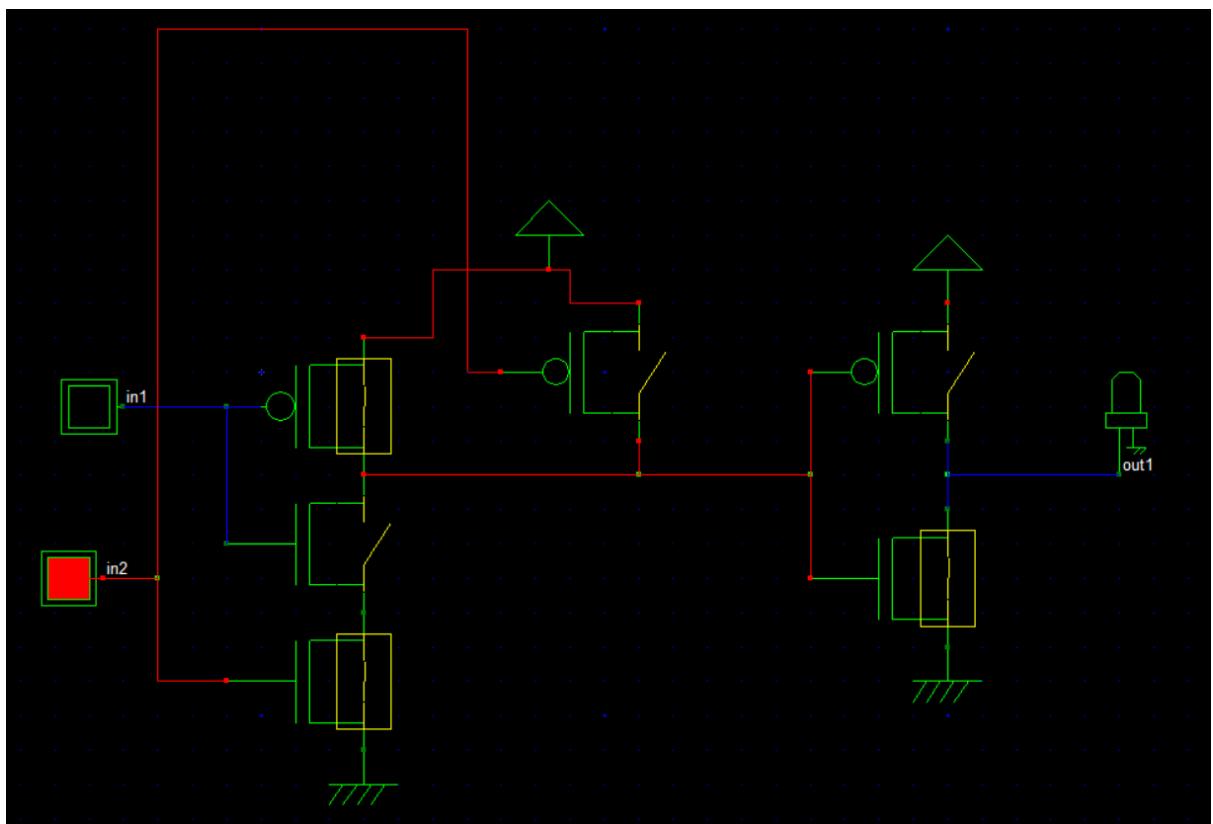
- Schema DSCH



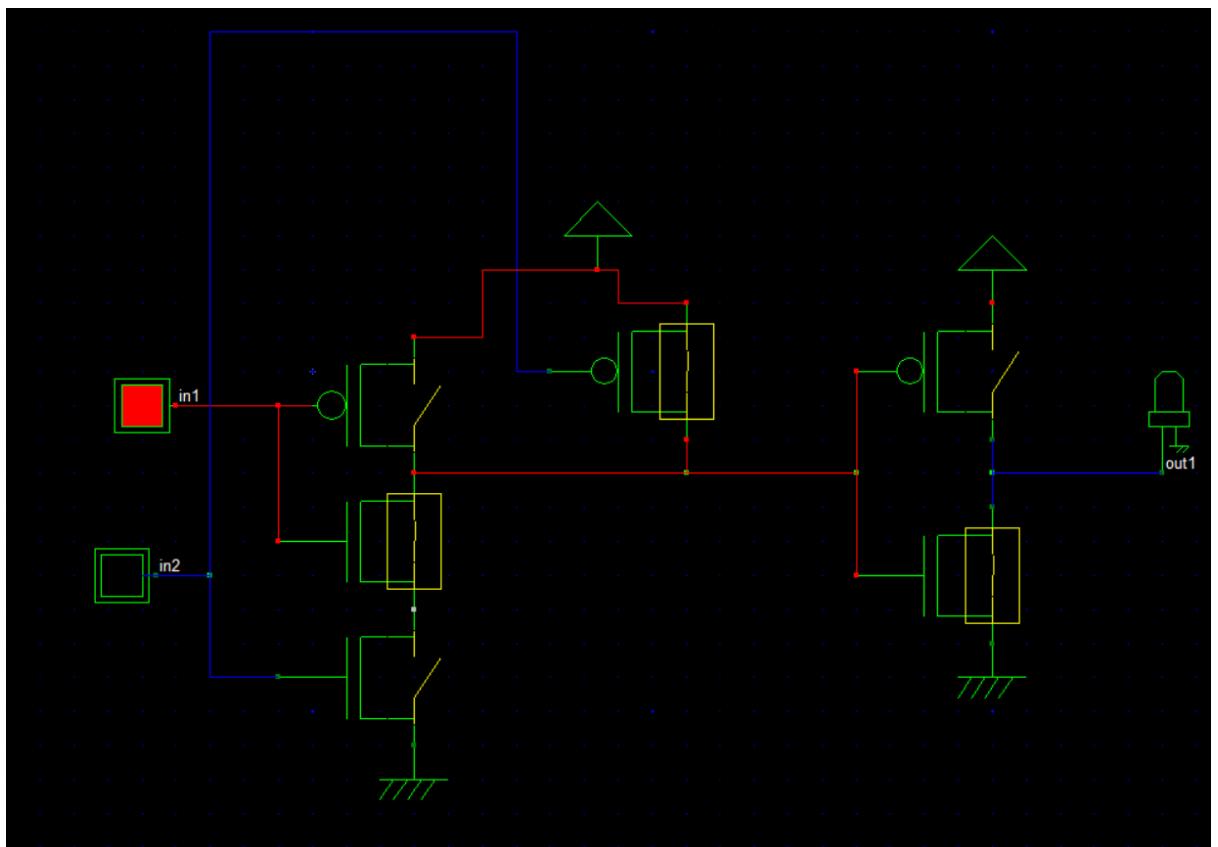
- Simulare DSCH
In 0 0 => out 0



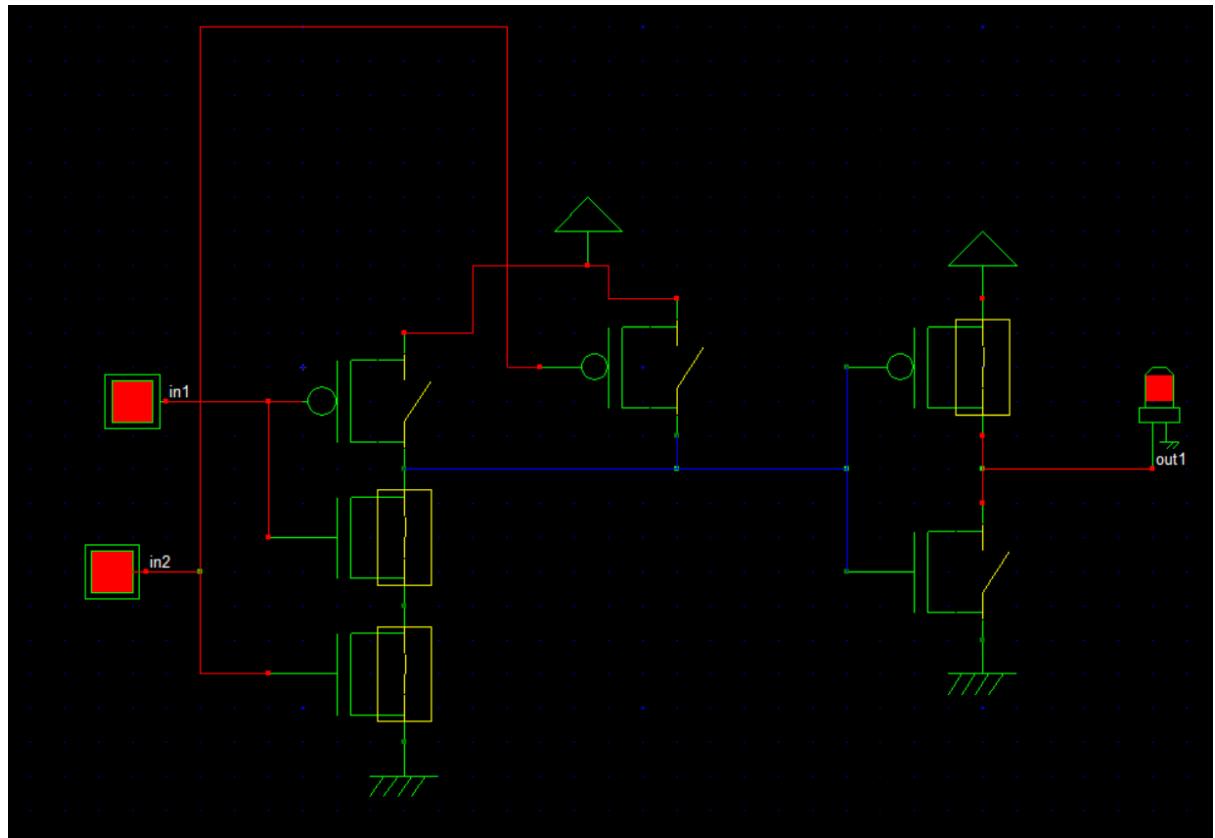
In 0 1 => out 0



In 1 0 => out 0



In 1 1 => out 1



- Cod verilog

Verilog, Hierarchy and Netlist

Verilog | Hierarchy | Netlist | Critical path

```
// DSCH 3.5
// 5/8/2023 10:35:10 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\AND.sch

module AND( in1,in2,out1);
    input in1,in2;
    output out1;
    wire w4,w6;
    pmos #(3) pmos_1(w4,vdd,in1); // 0.5u 0.07u
    pmos #(2) pmos_2(out1,vdd,w4); // 0.5u 0.07u
    nmos #(2) nmos_3(out1,vss,w4); // 0.3u 0.07u
    nmos #(1) nmos_4(w6,vss,in2); // 0.3u 0.07u
    pmos #(3) pmos_5(w4,vdd,in1); // 0.5u 0.07u
    nmos #(3) nmos_6(w4,w6,in1); // 0.3u 0.07u
endmodule

// Simulation parameters in Verilog Format
always
#200 in1=~in1;
#400 in2=~in2;

// Simulation parameters
// in1 CLK 1
// in2 CLK 2
```

Information

Module name (8 char. max)
AND

Add gate delay info
 Append simul. infomations
 Add labels as comments

The Verilog file has 24 lines
The design includes 14 symbols
The circuit has 6 nodes

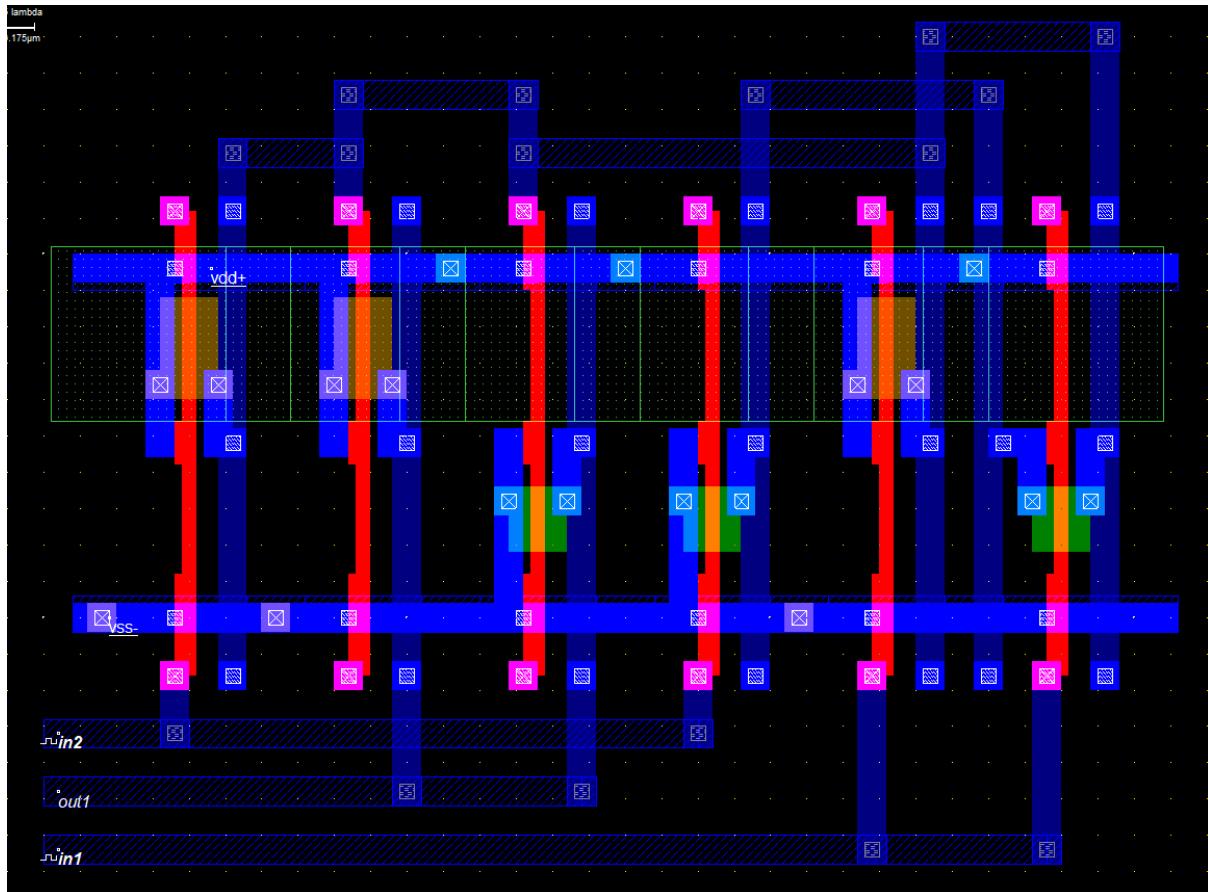
Misc.

Time scale : 1.00
Max clocks: 16

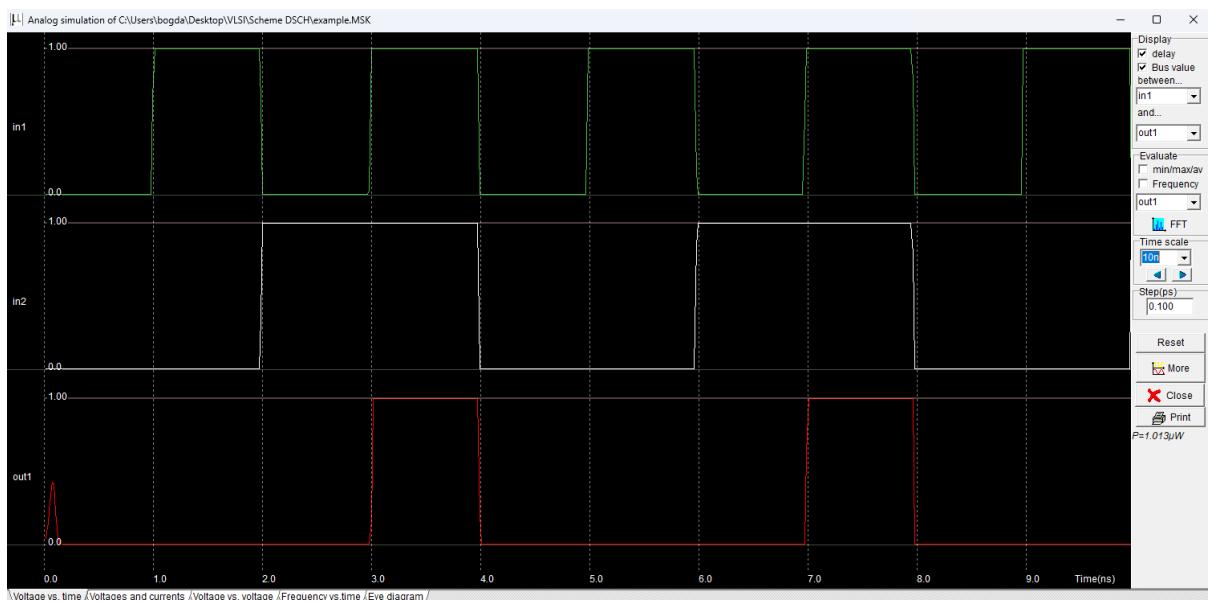
Update Verilog Extract circuit

OK

- Microwind

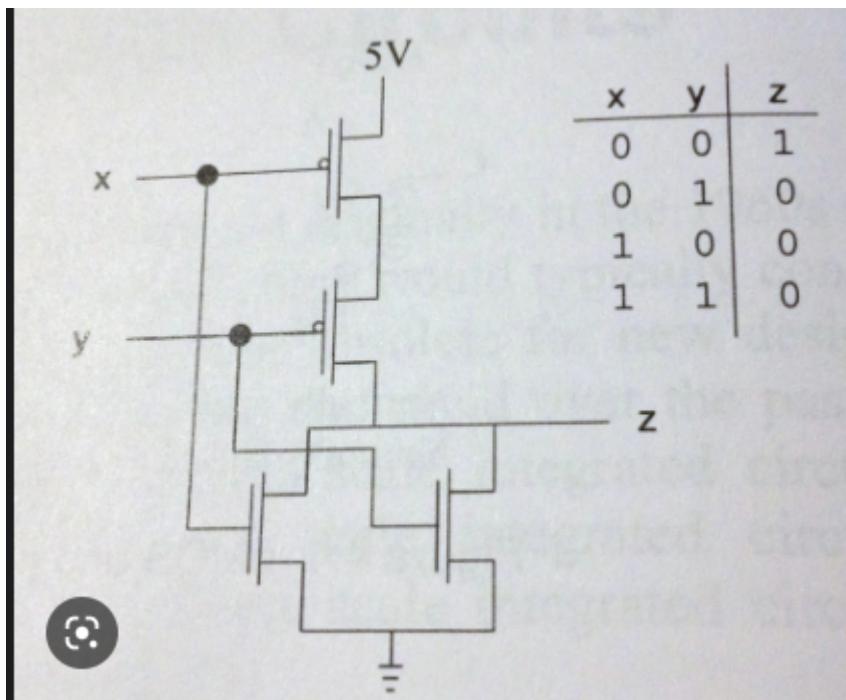


- Simulare Microwind



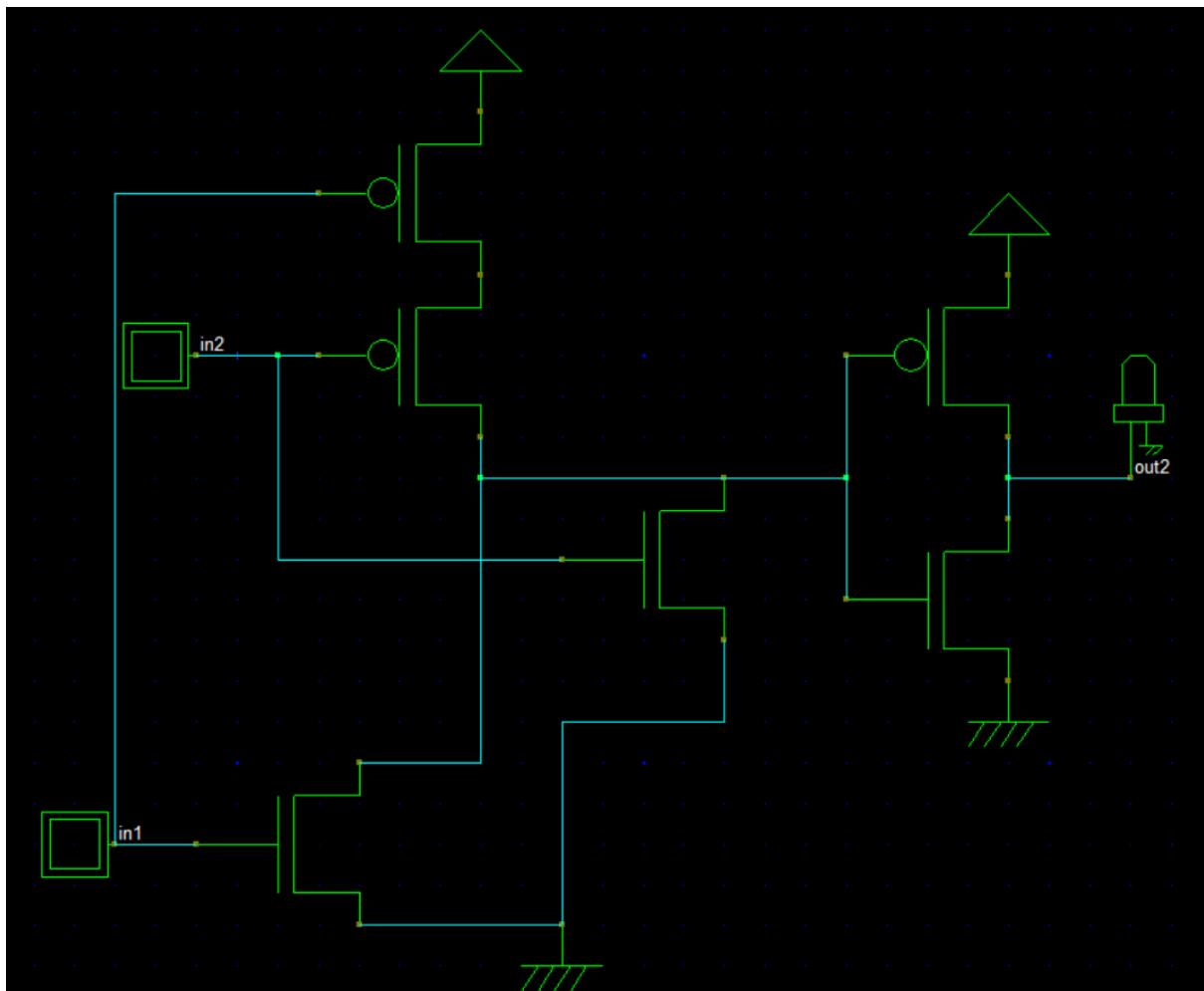
3. OR gate

- schema circuitului

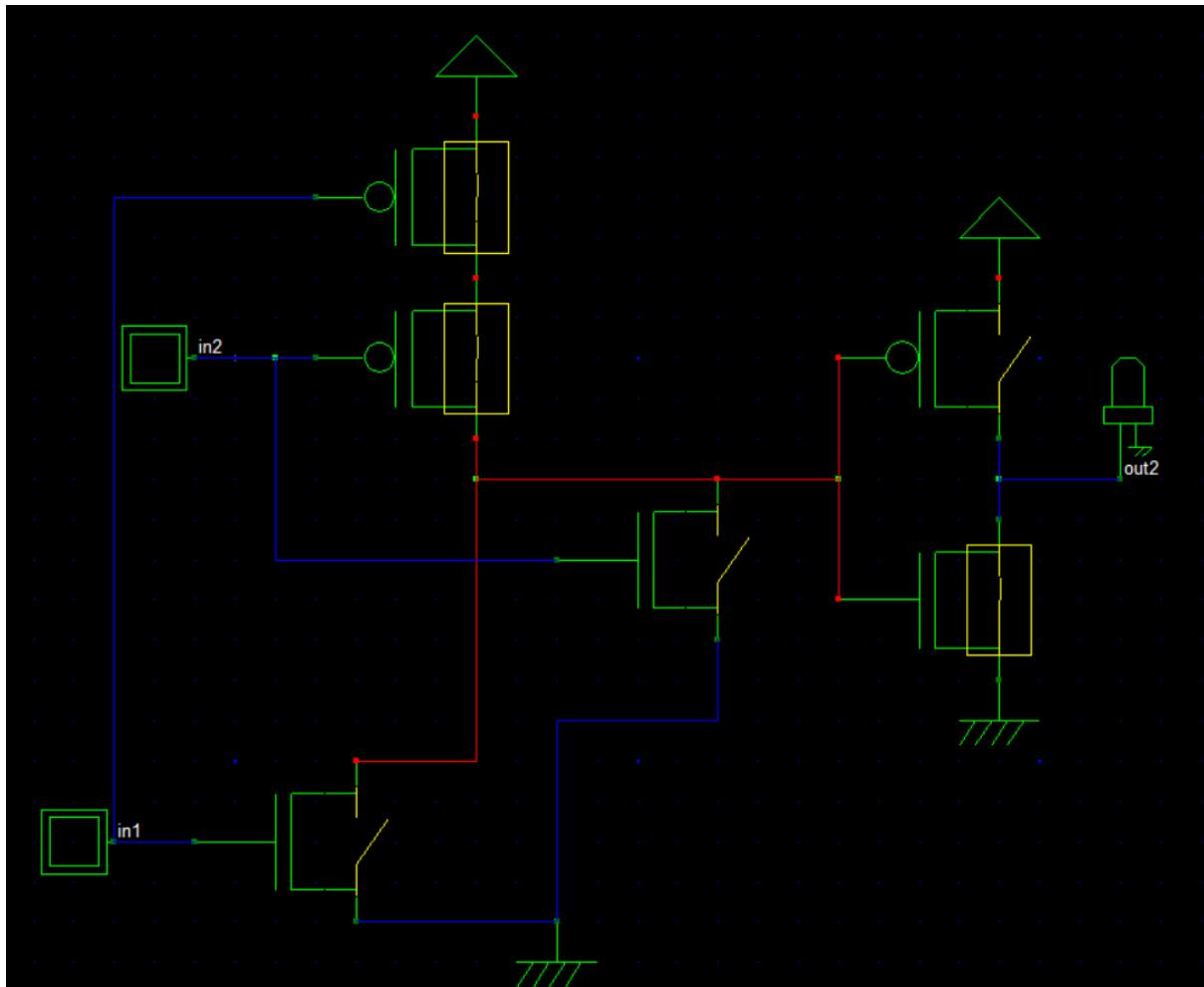


<https://wiresharklabs.wordpress.com/category/digital-design/chapter-4/>

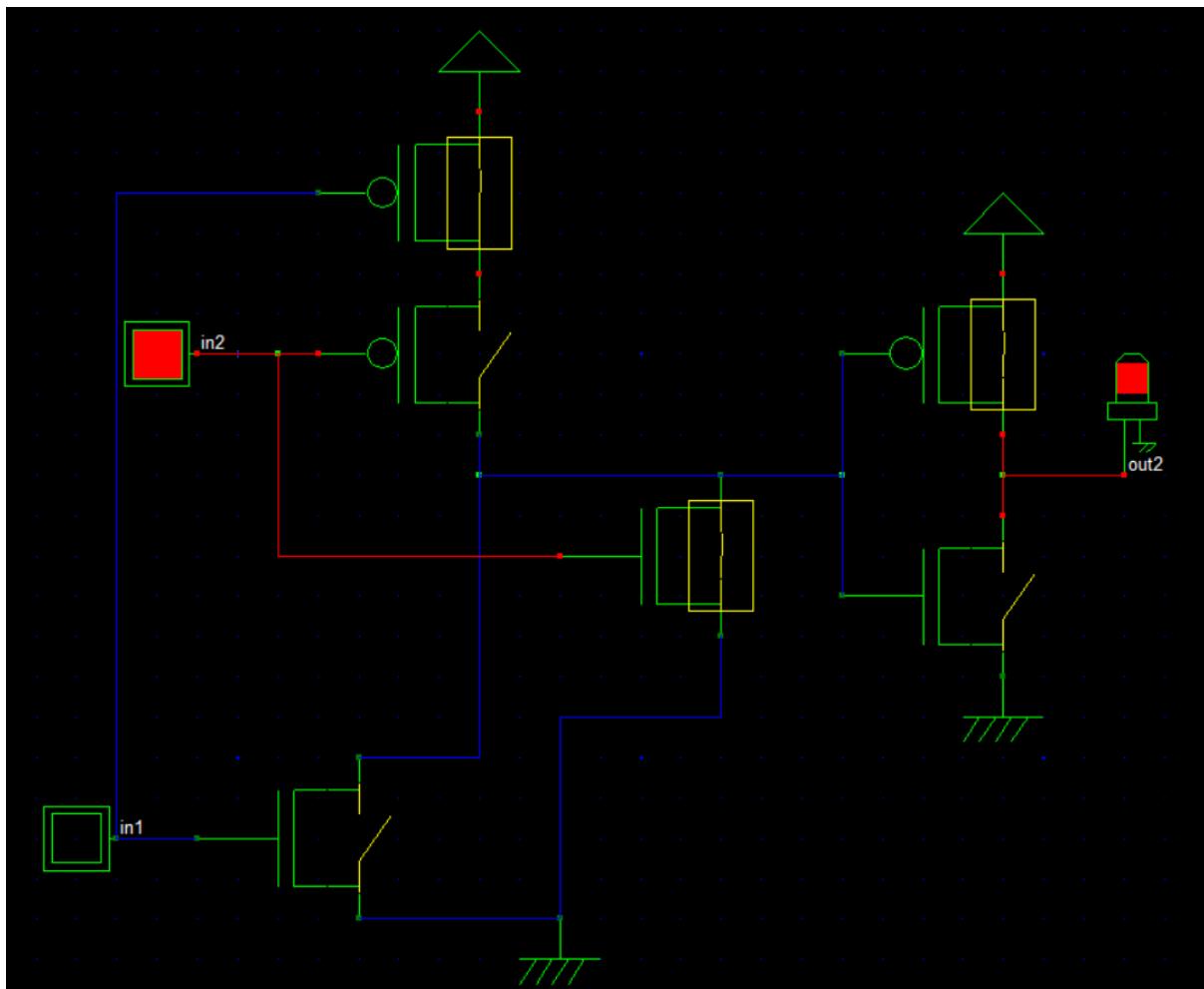
- Schema DSCH



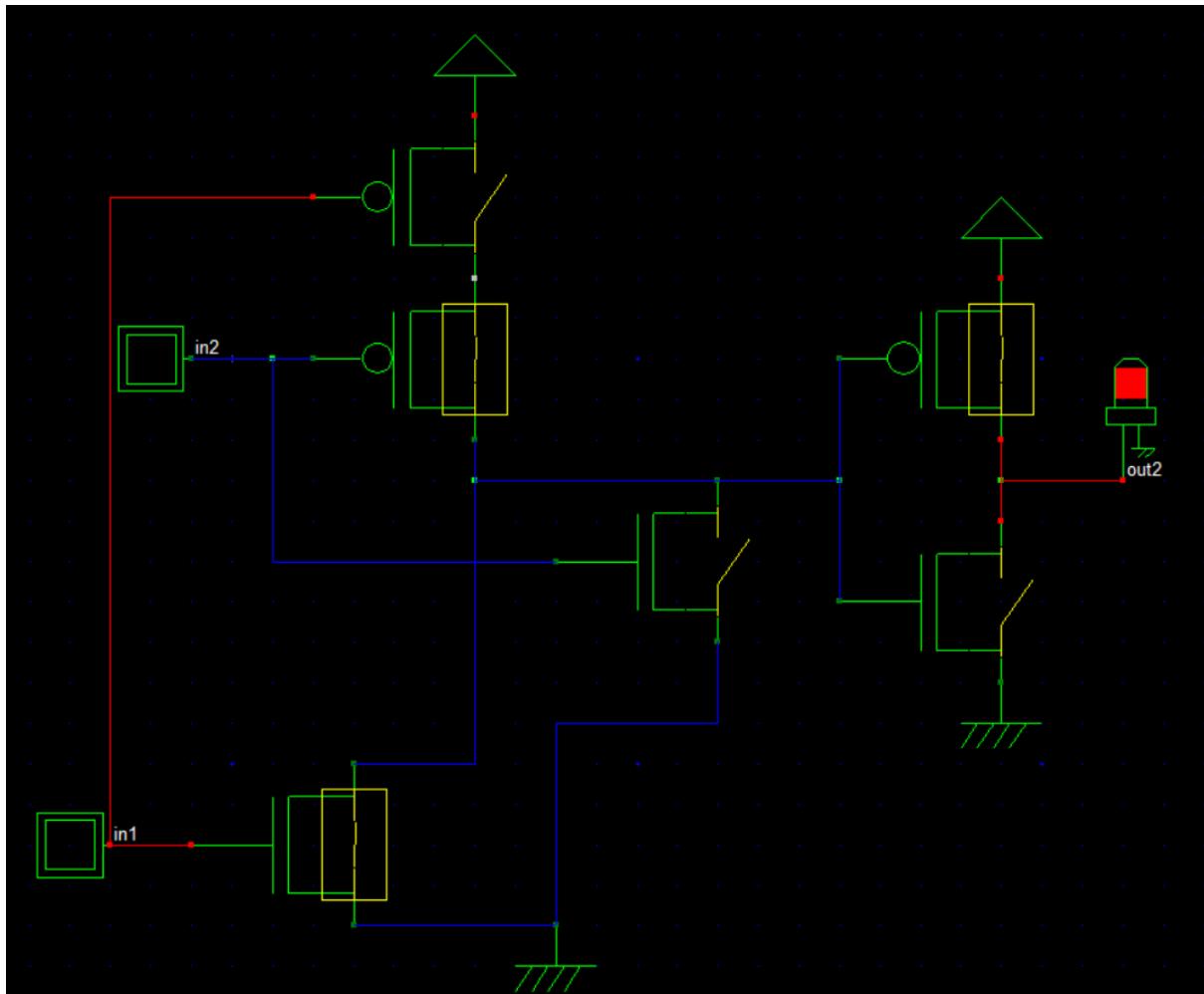
- Simulare DSCH
Input 0 0 => out = 0



Input 0 1 => out = 1

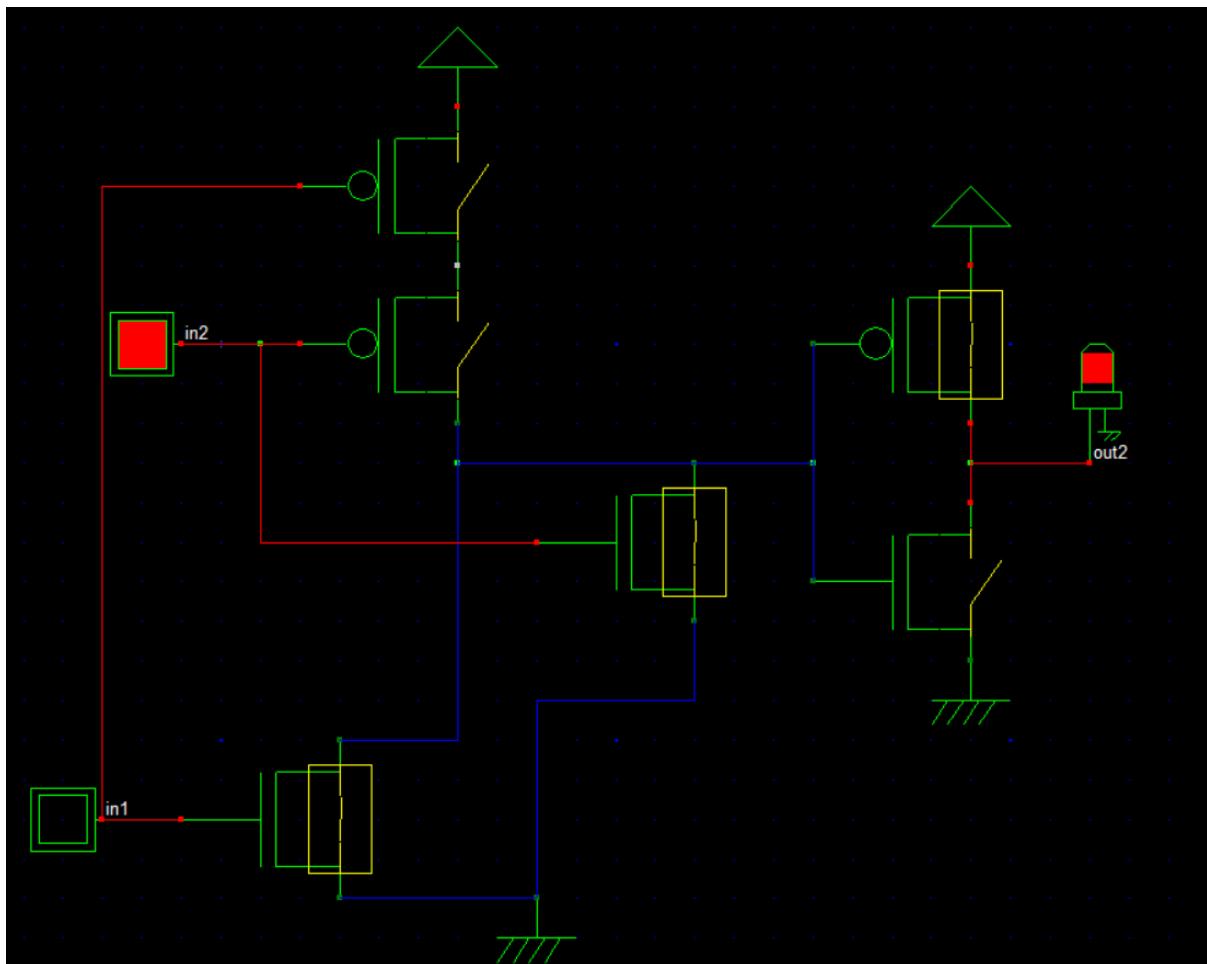


Input 1 0 => out = 1

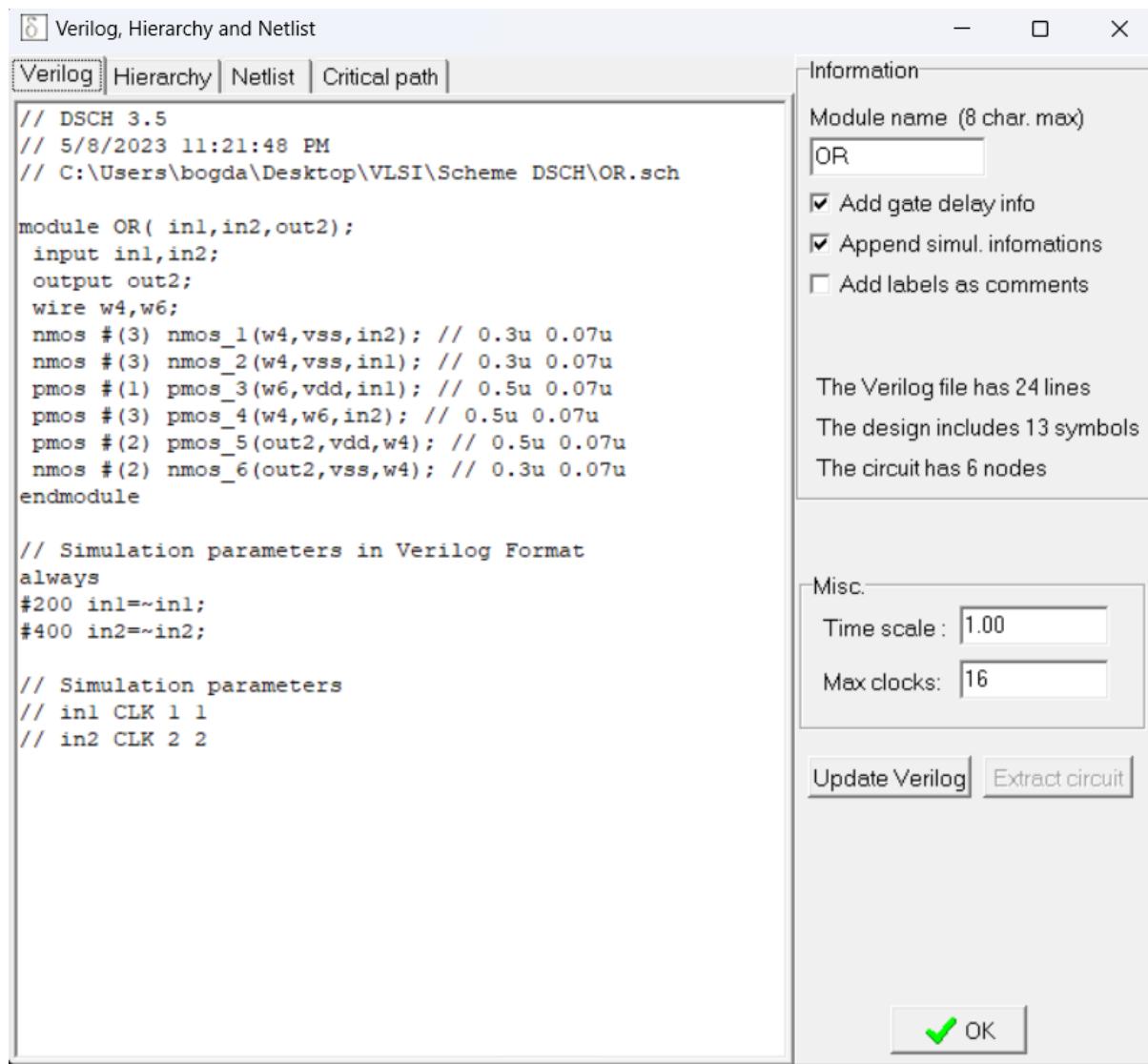


Input 1 1 => out = 1

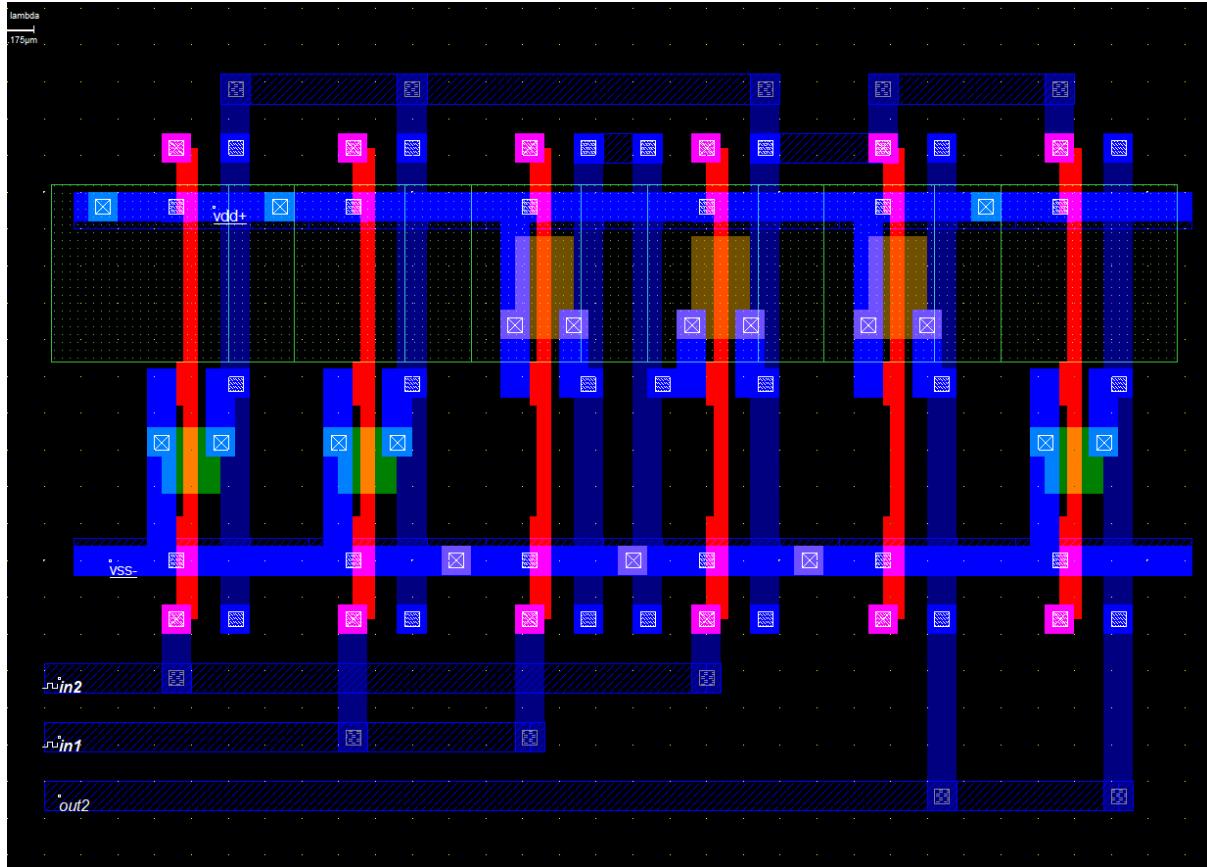
Aici nu stiu de ce nu apare animatia de functionare la Input 1, insa acesta este pornt, deoarece firul este rosu.



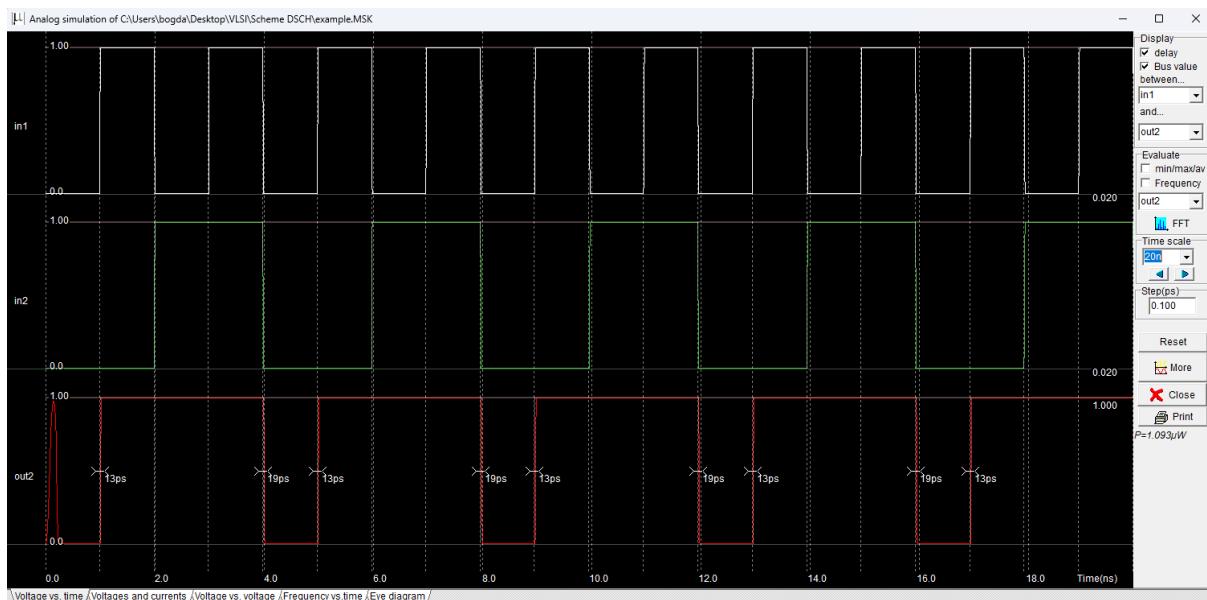
- Cod verilog



- Microwind

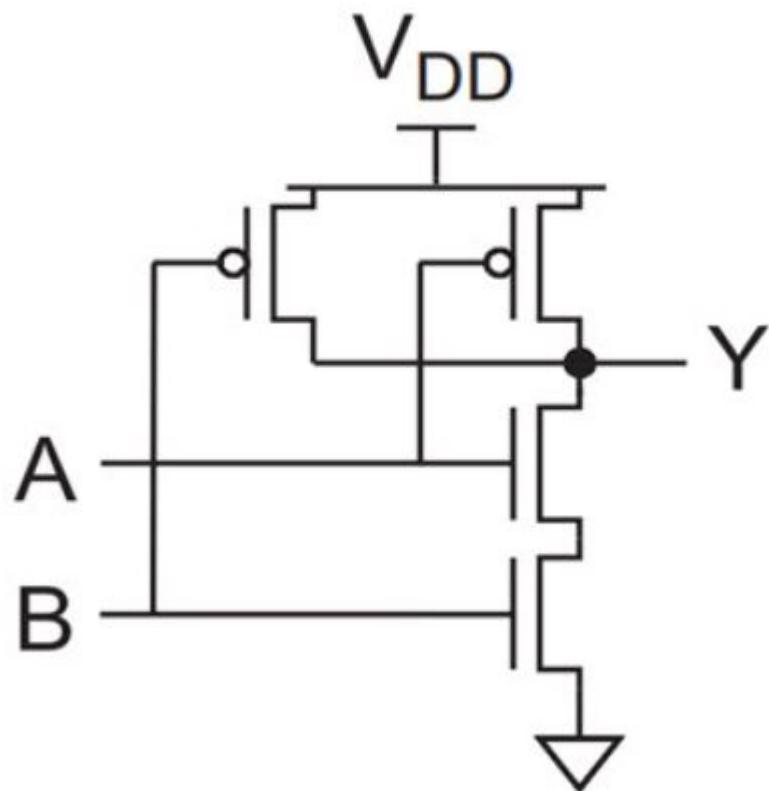


- Simulare Microwind



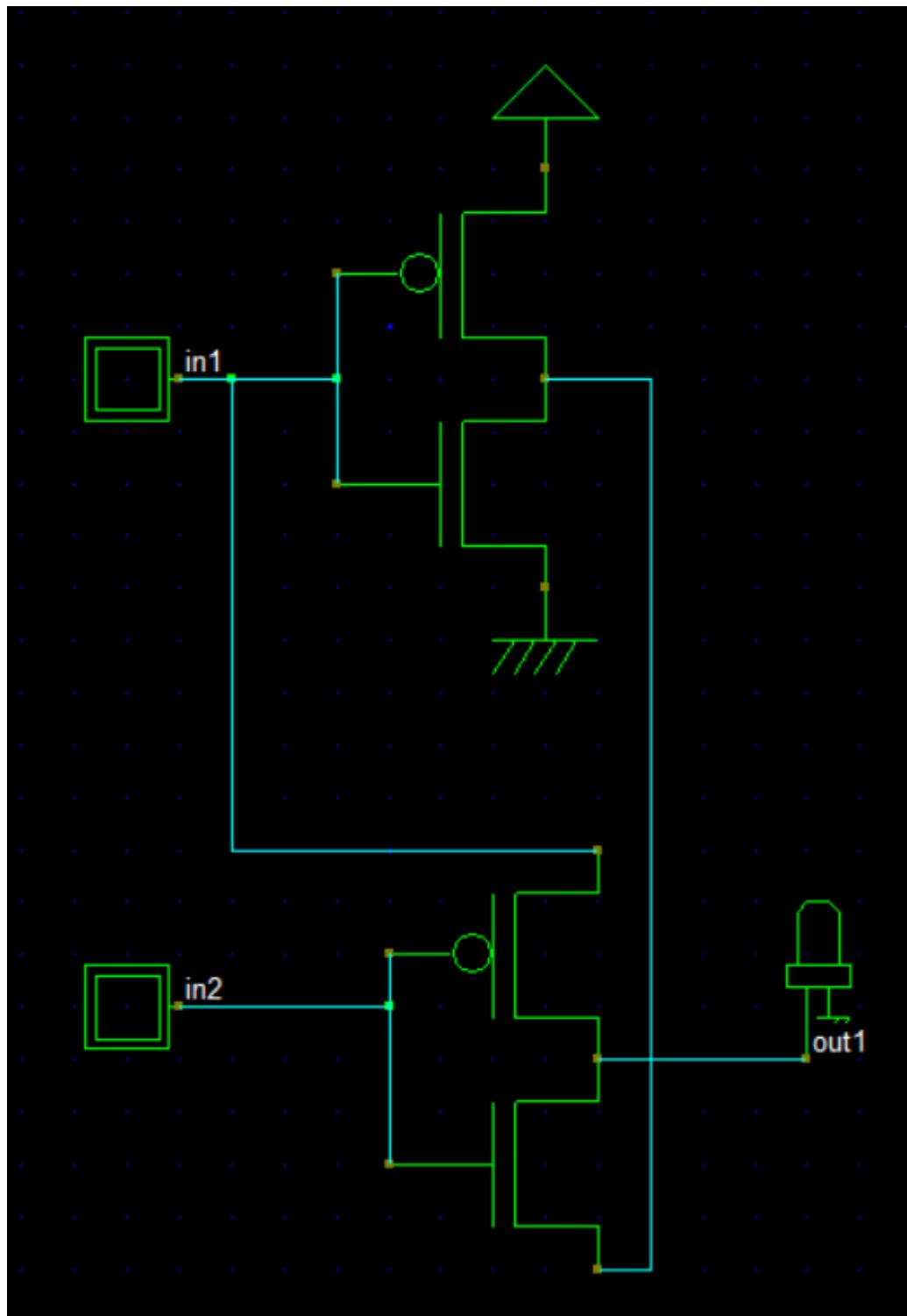
4. XOR GATE

- schema circuitului

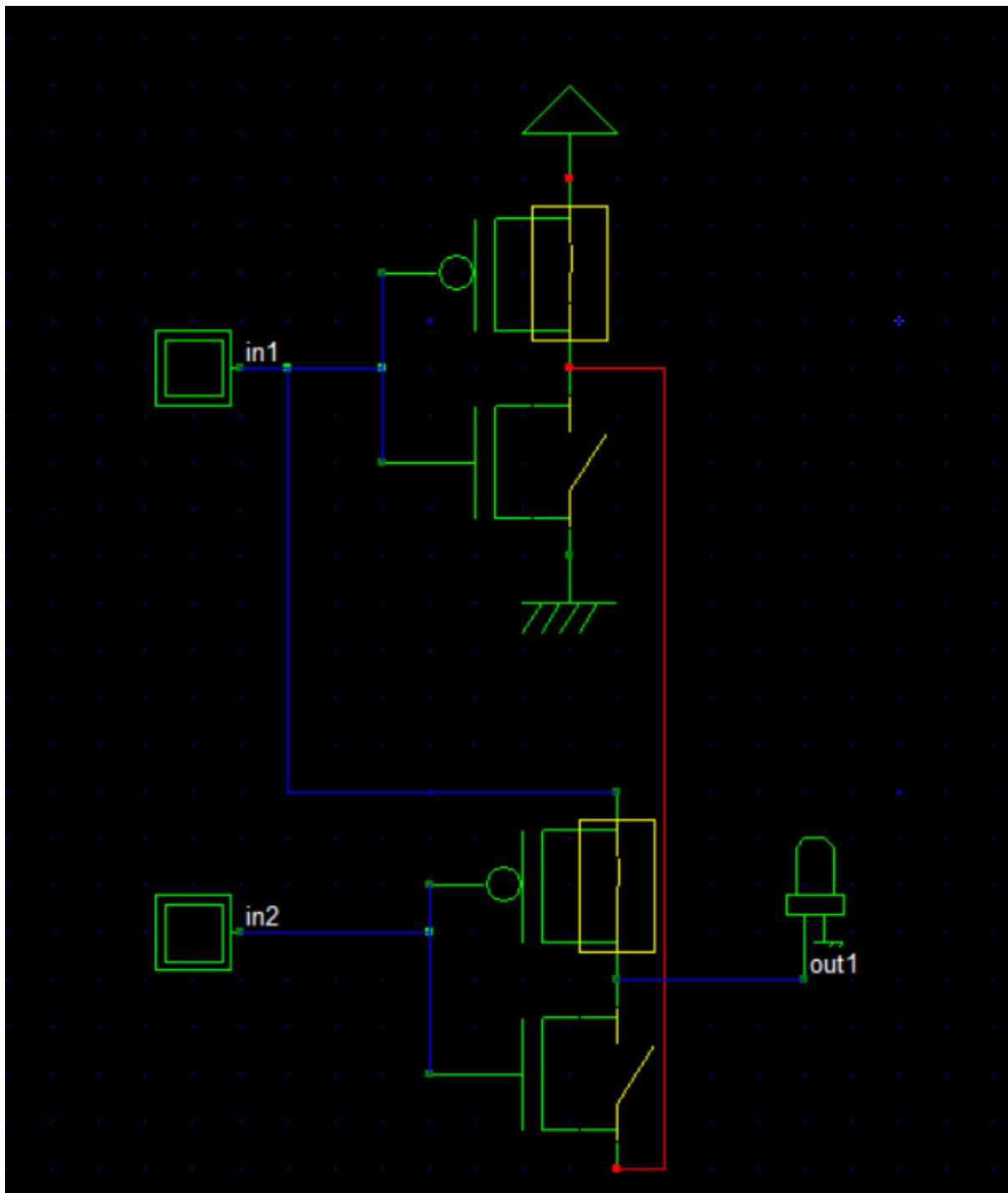


<https://www.chegg.com/homework-help/questions-and-answers/tspice-cmos-xor-gate-cmos-nand-gate-mosfet-s-length-l-018um-case-nmos-pmos-s-width-um-know-q50432658>

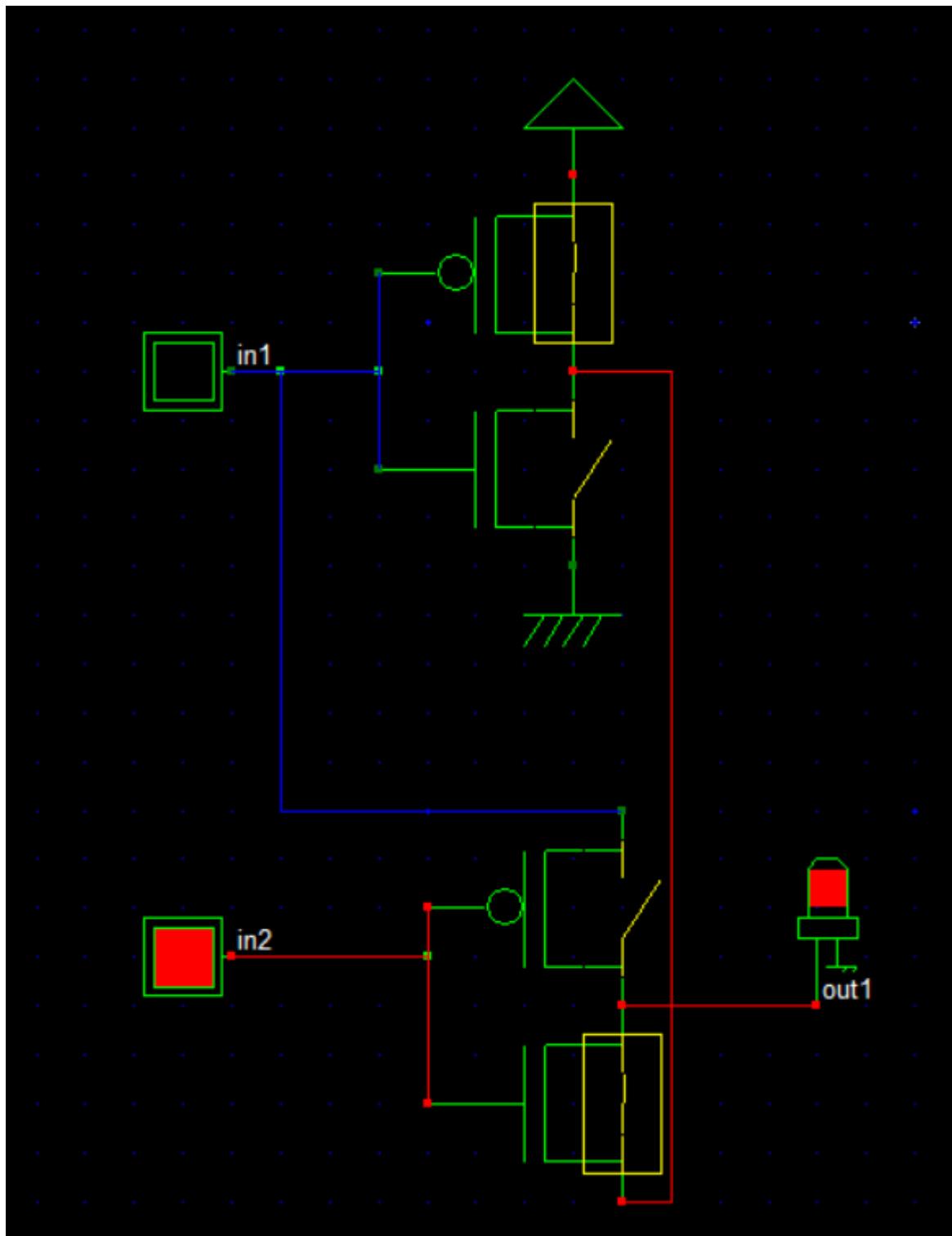
- Schema DSCH



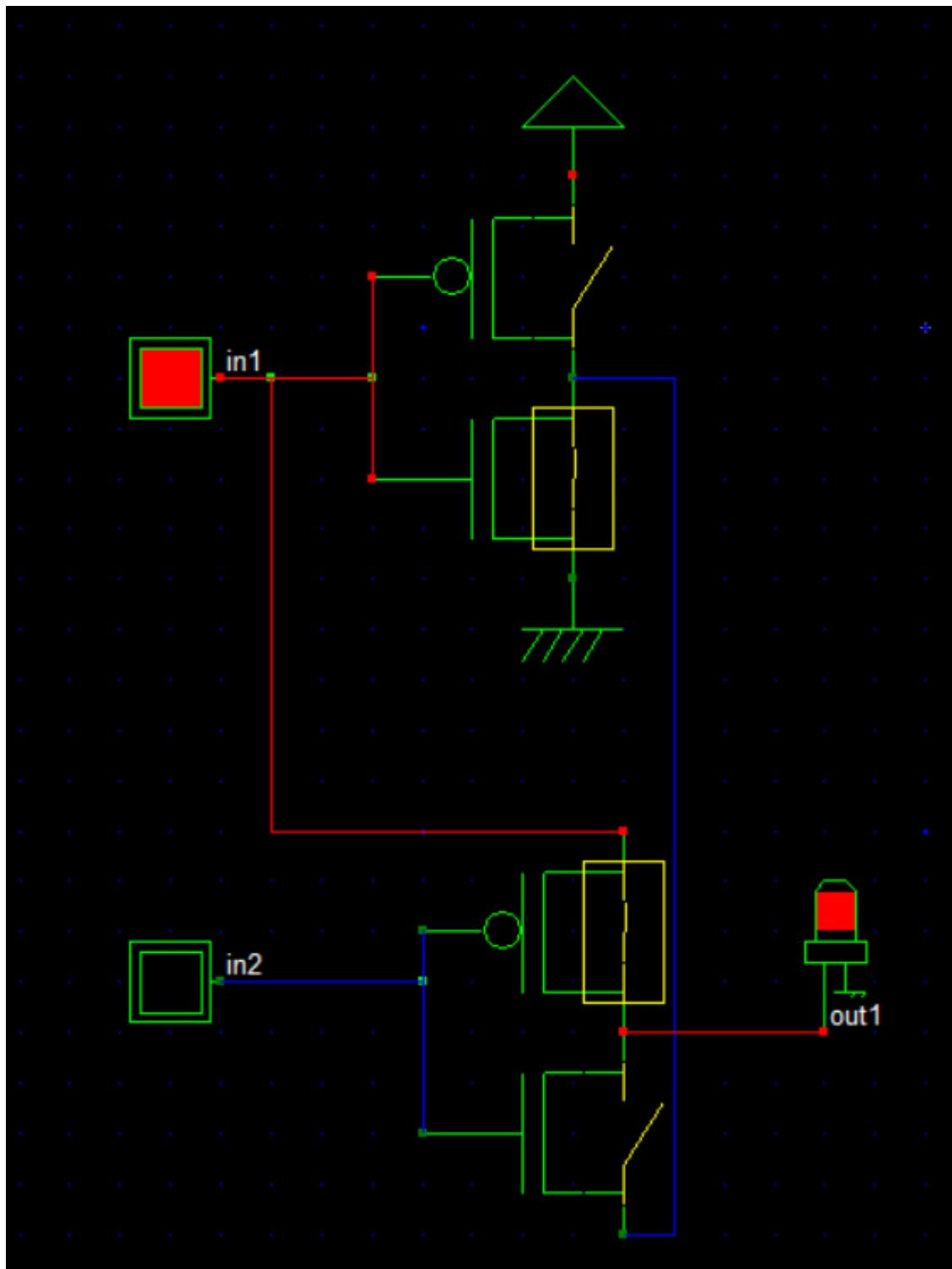
- Simulare DSCH
Input 0 0 => out = 0



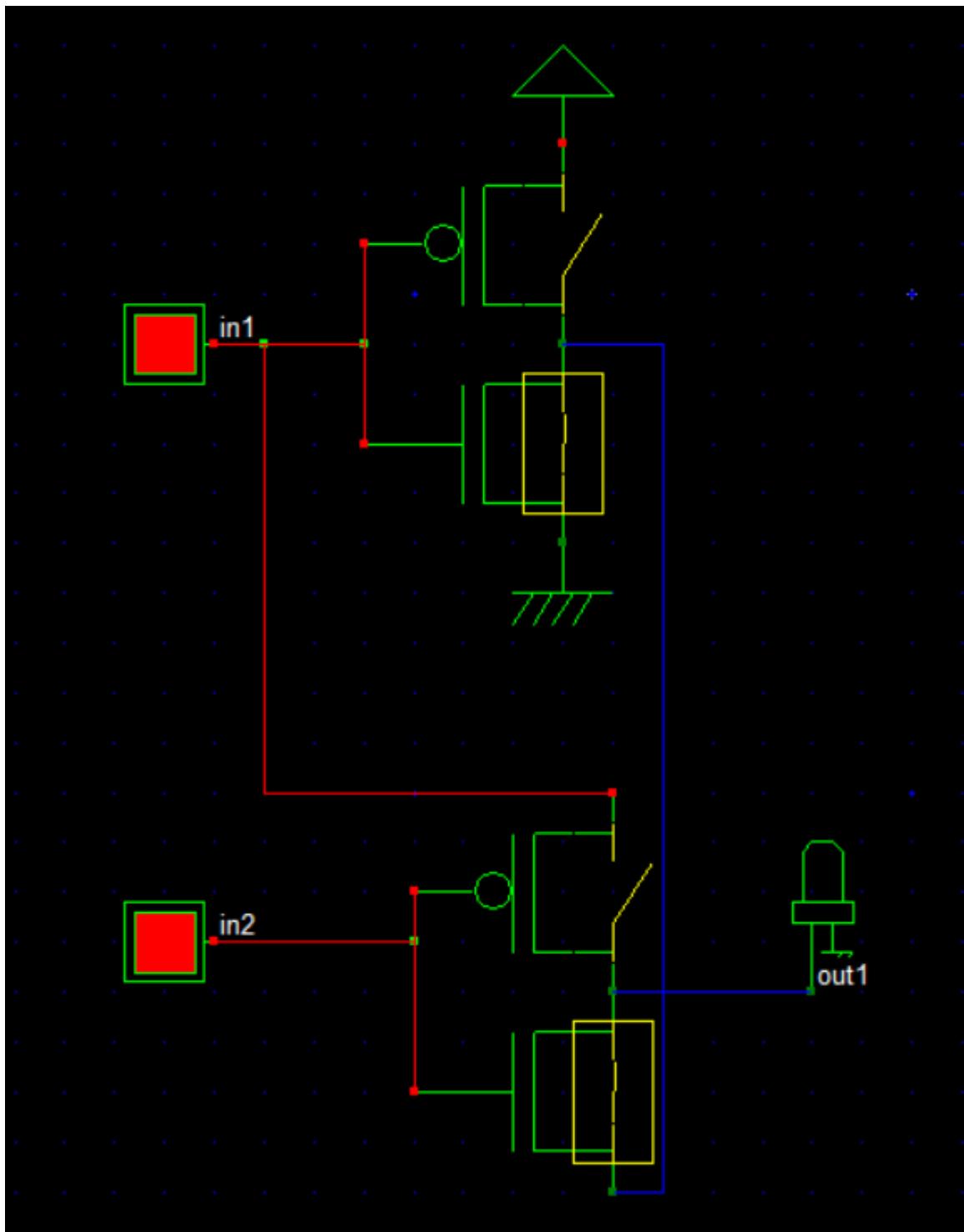
Input 0 1 => out = 1



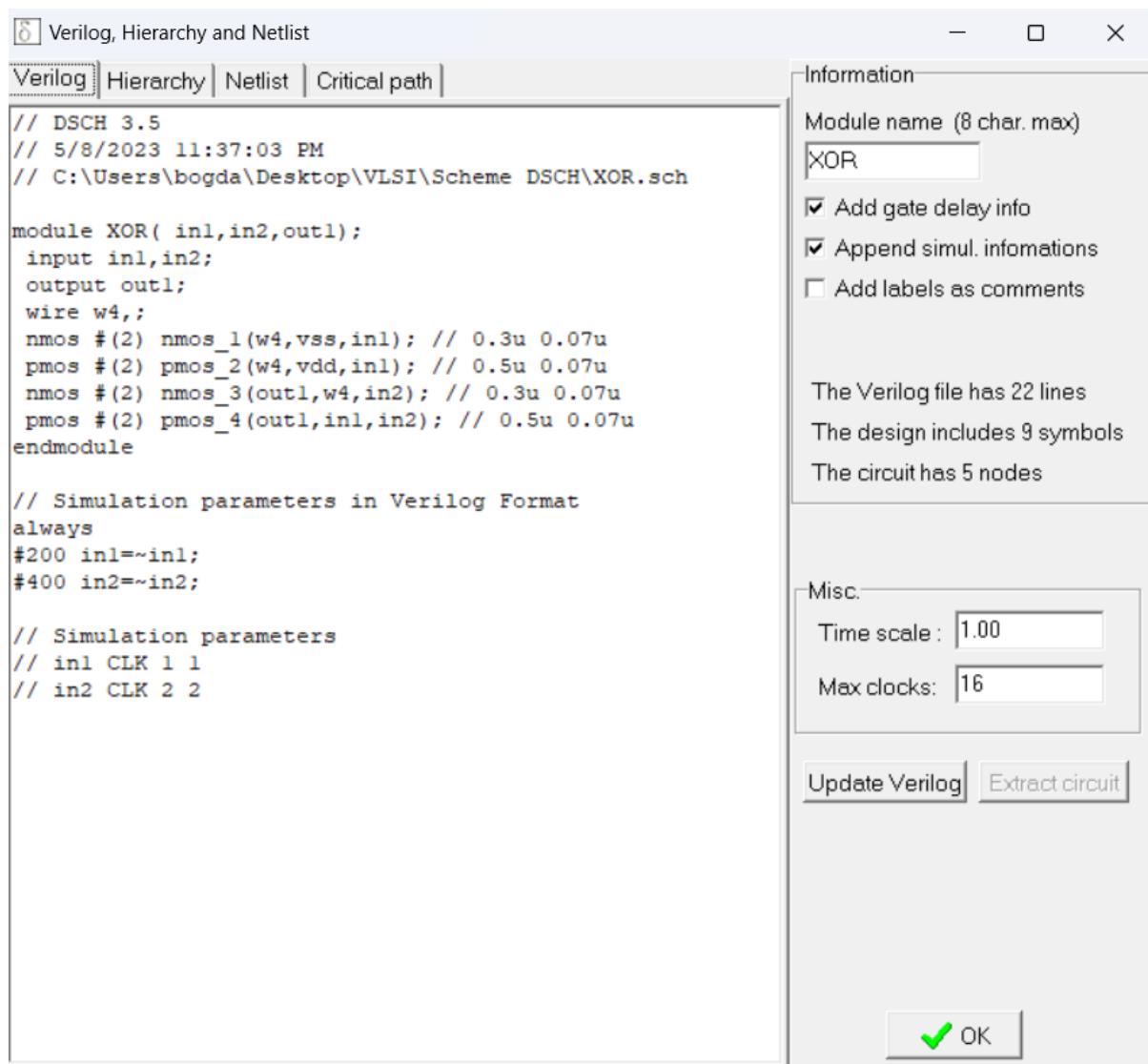
Input 1 0 => out = 1



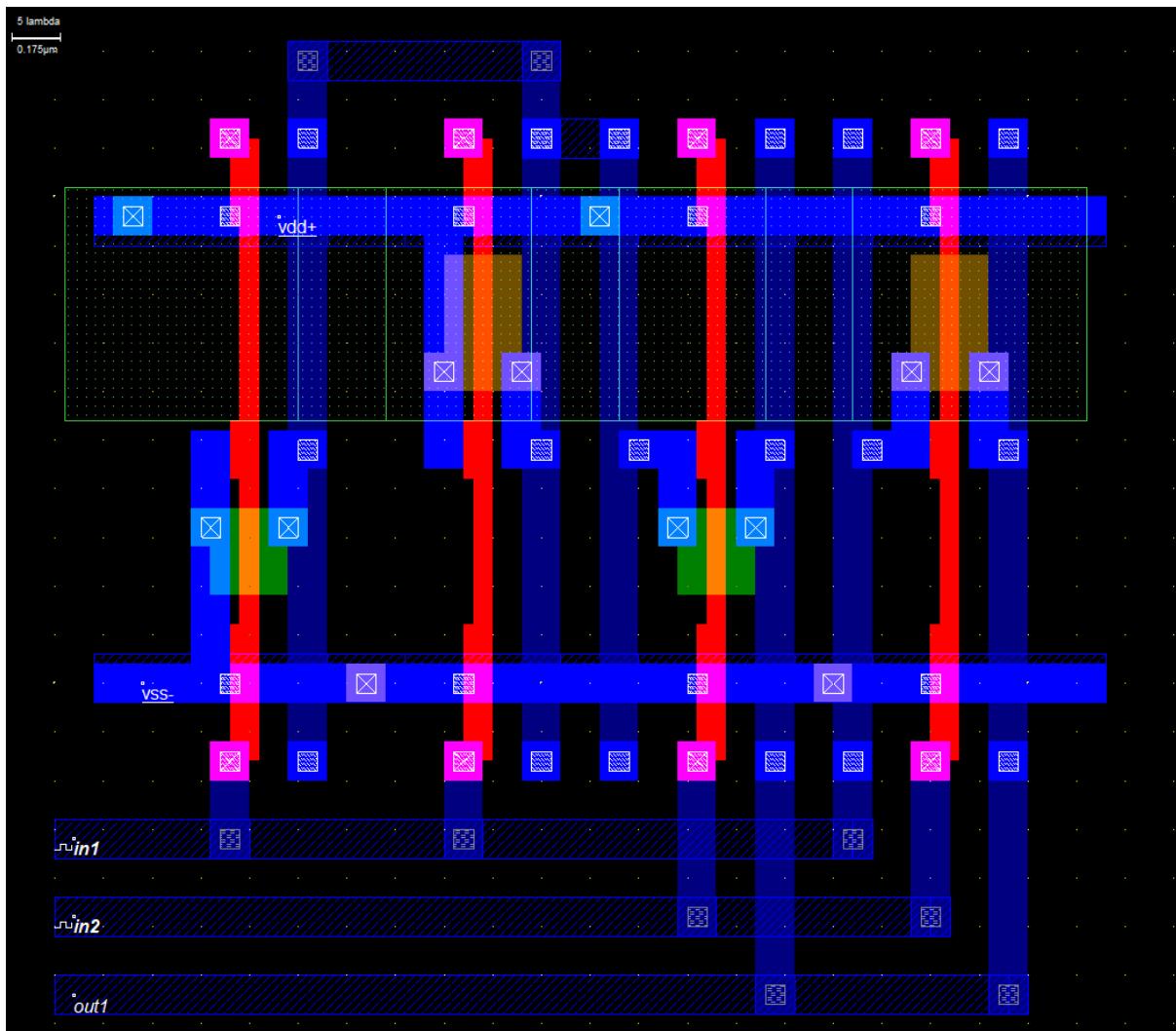
Input 1 1 => out = 0



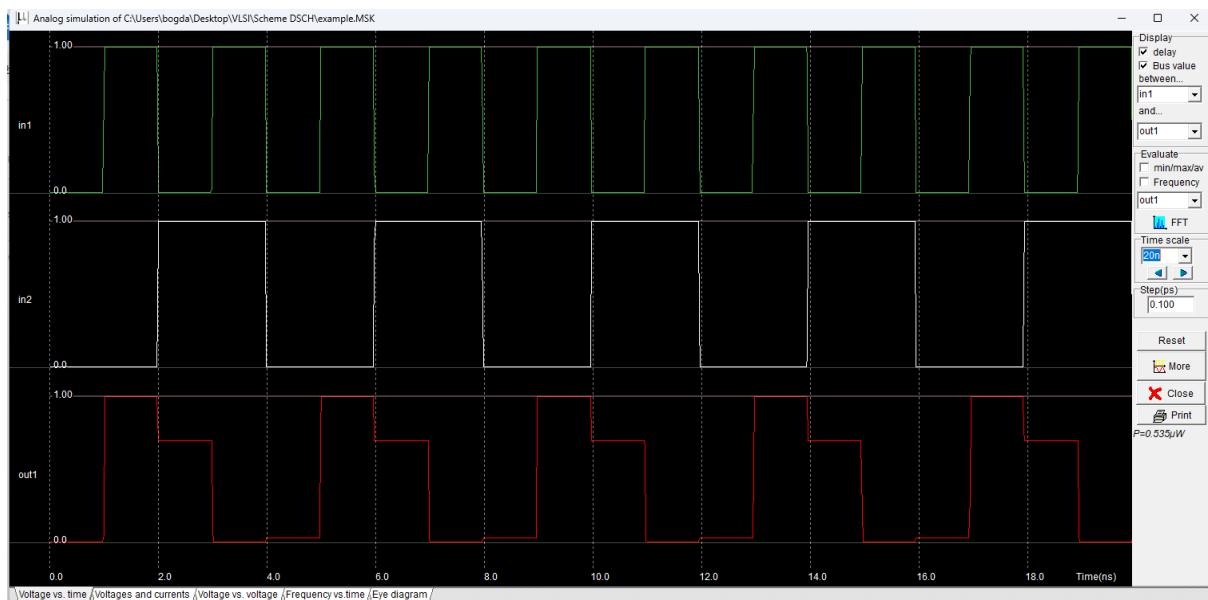
- Cod Verilog



- Microwind

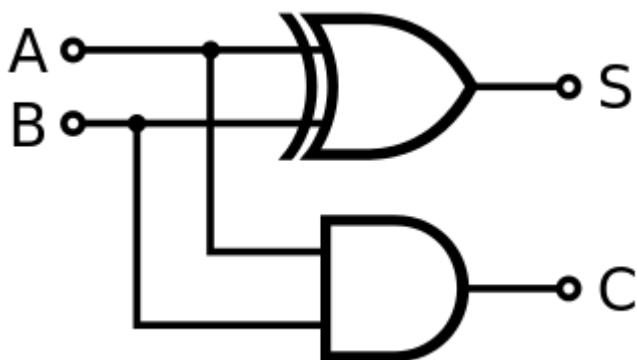


- Simulare Microwind



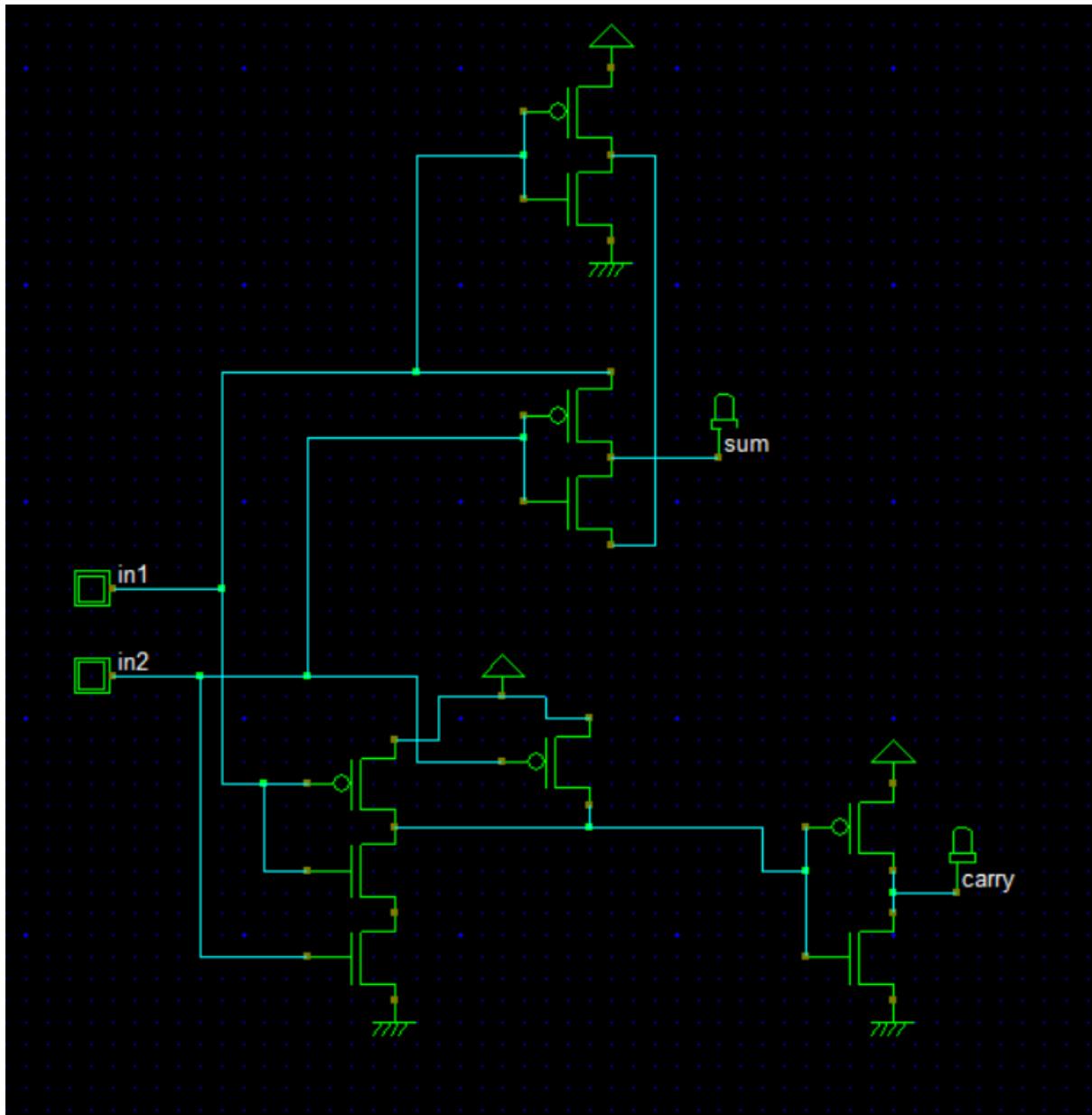
5. SUMATOR

- schema circuitului



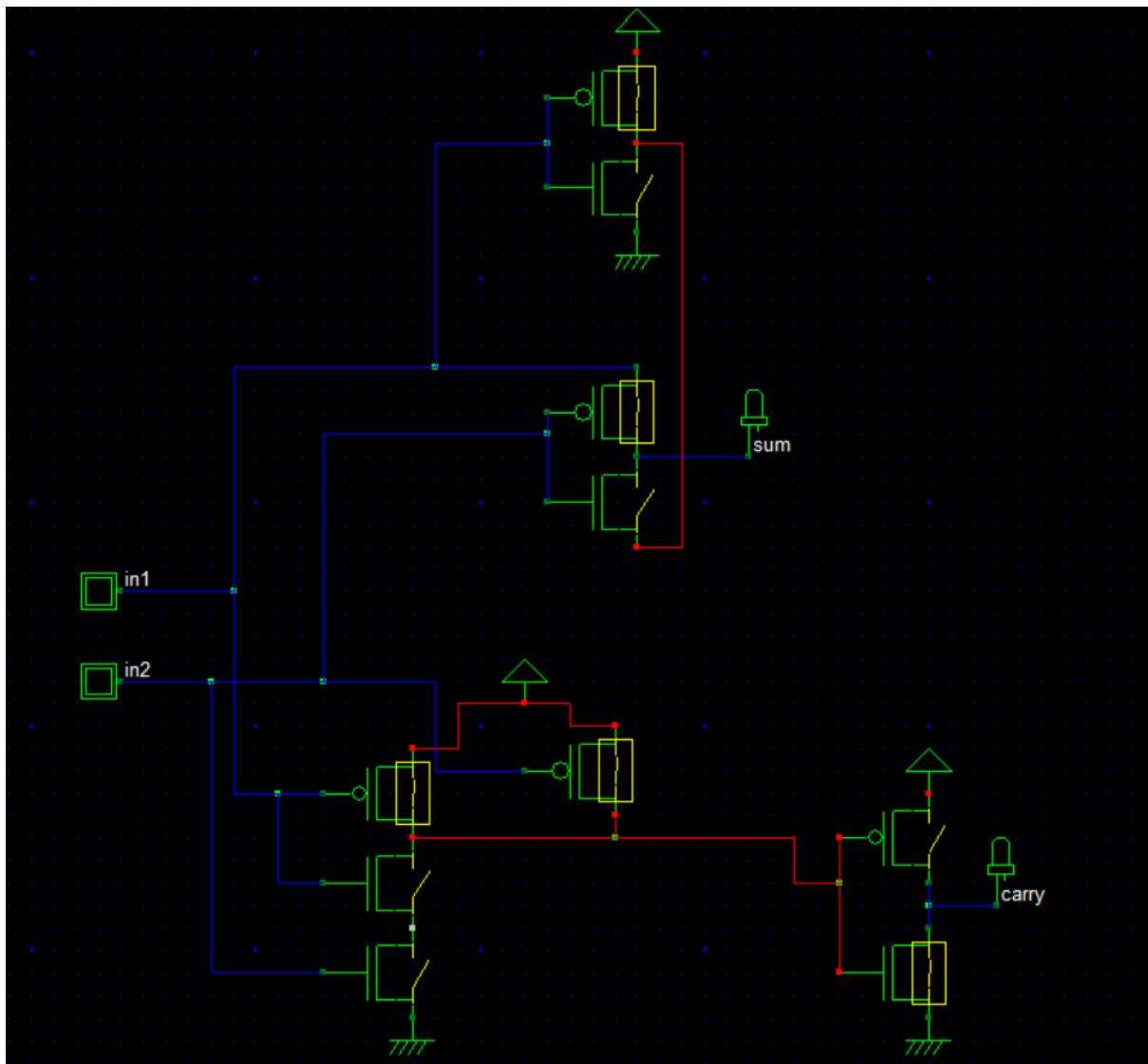
<https://www.techopedia.com/definition/7509/half-adder>

- Schema DSCH

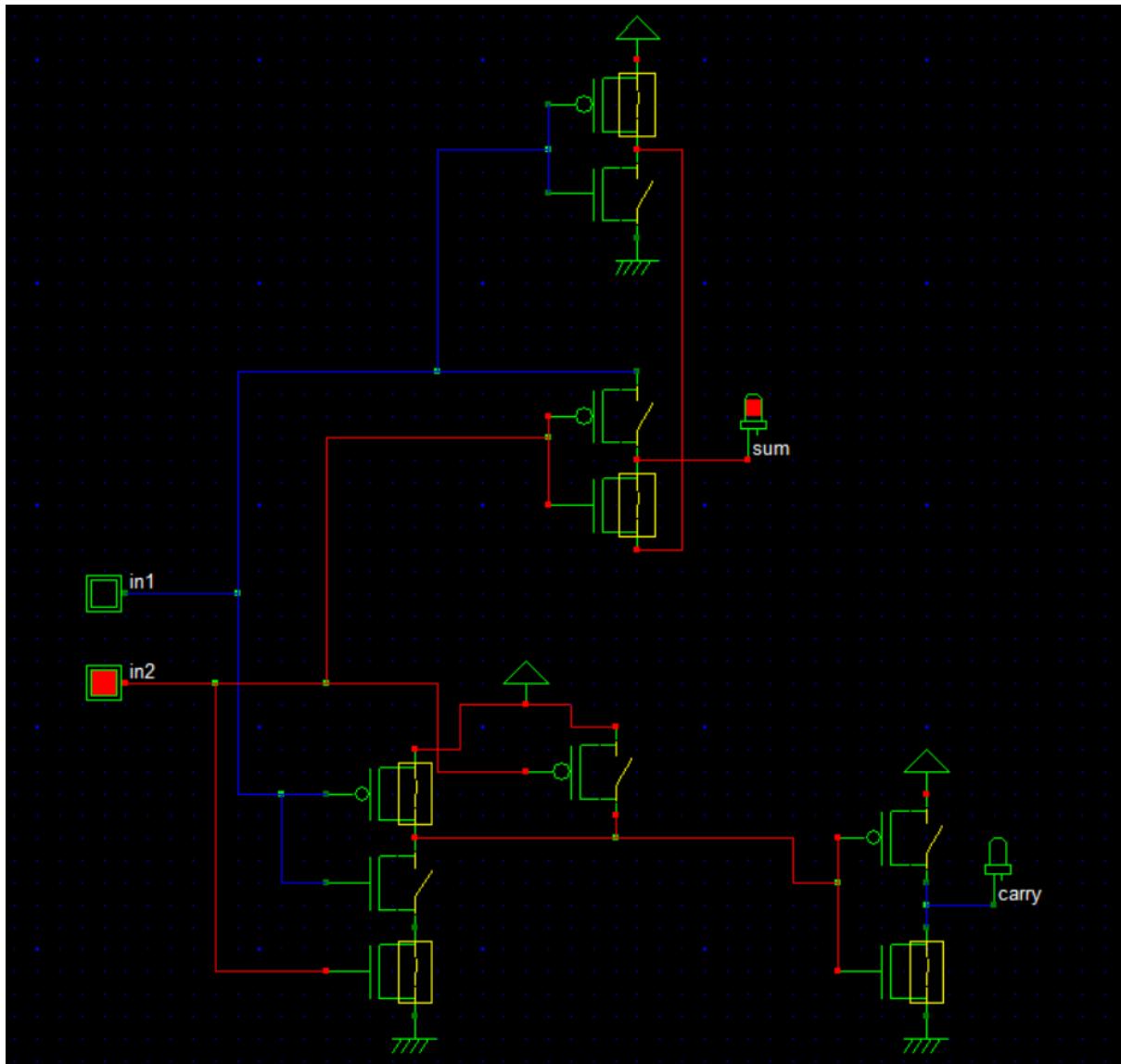


- Simulare DSCH

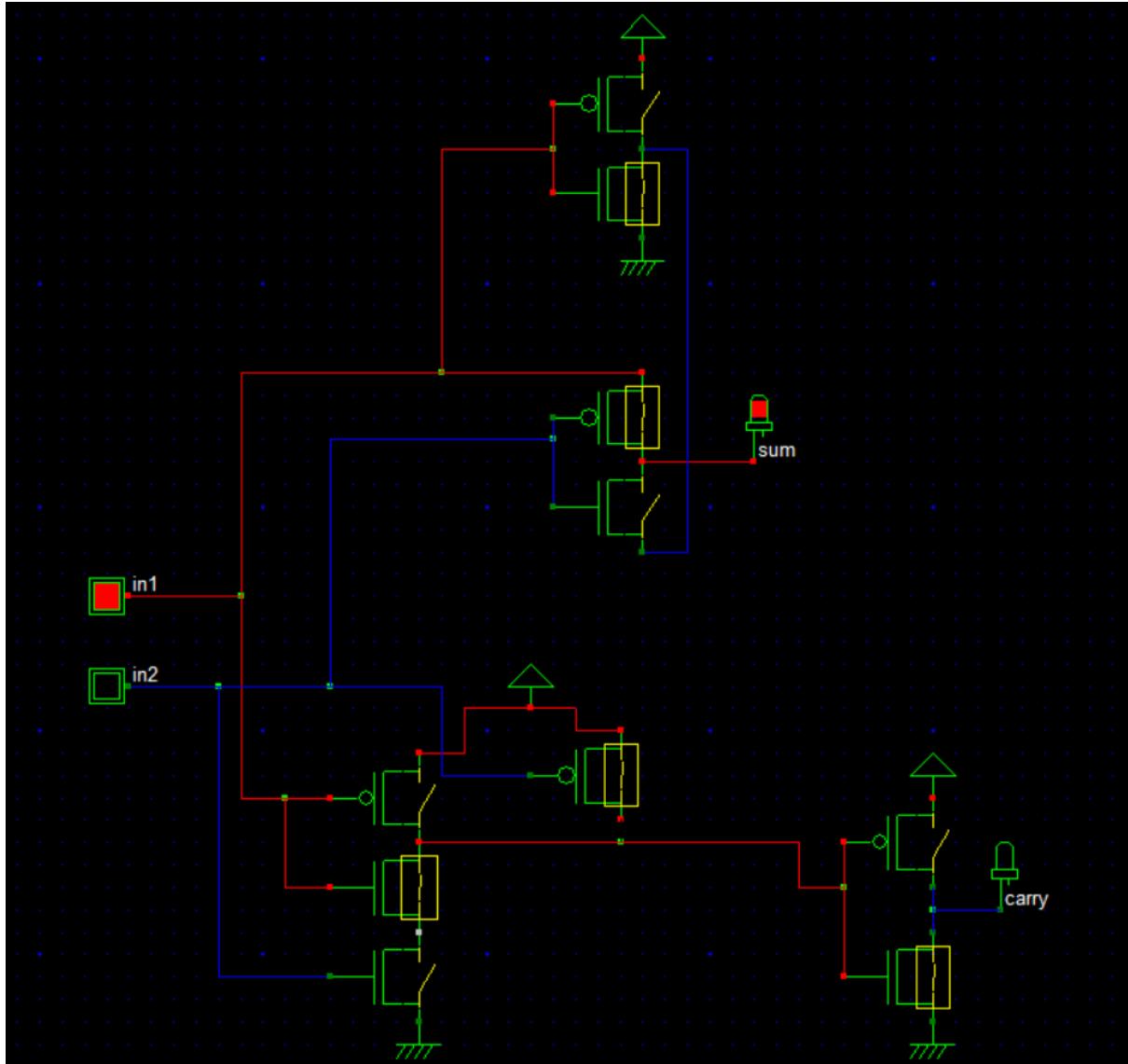
$$a = b = 0 \Rightarrow \text{sum} = \text{cout} = 0$$



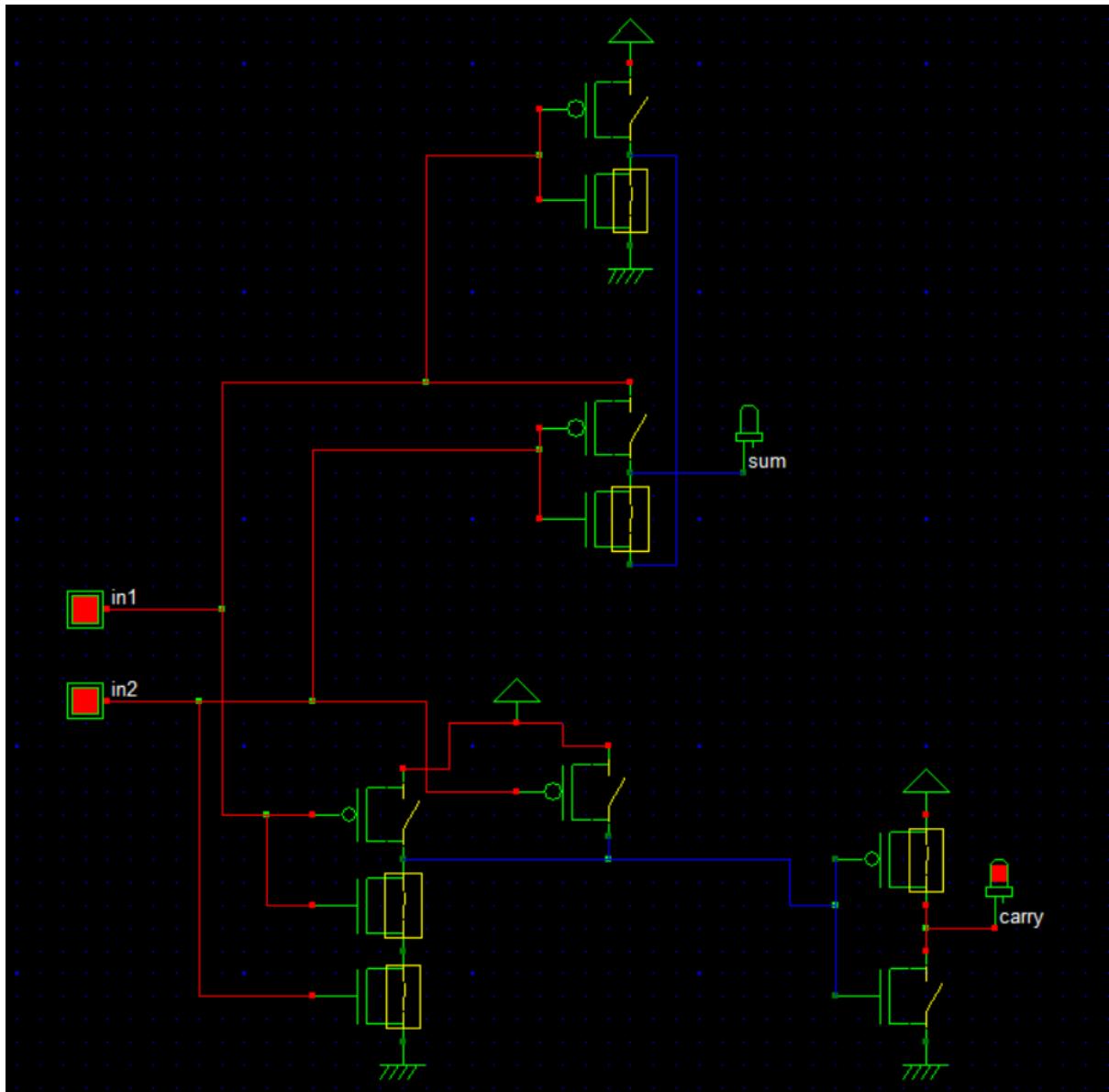
$a = 0$ $b = 1$ $\text{sum} = 1$ $\text{cout} = 0$



$a = 1$ $b = 0$ $\text{sum} = 1$ $\text{cout} = 0$



a = 1 *b* = 1 *sum* = 0 *cout* = 1



- Cod Verilog

Verilog, Hierarchy and Netlist | Critical path |

```
// DSCH 3.5
// 5/9/2023 8:45:25 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\sumator.sch

module sumator( in1,in2,carry,sum);
    input in1,in2;
    output carry,sum;
    wire w4,w6,w7,;
    nmos #(1) nmos_1(w4,vss,in1); // 0.3u 0.07u
    nmos #(3) nmos_2(w6,w4,in1); // 0.3u 0.07u
    pmos #(3) pmos_3(w6,vdd,in1); // 0.5u 0.07u
    pmos #(3) pmos_4(w6,vdd,in2); // 0.5u 0.07u
    pmos #(2) pmos_5(carry,vdd,w6); // 0.5u 0.07u
    nmos #(2) nmos_6(carry,vss,w6); // 0.3u 0.07u
    nmos #(2) nmos_7(sum,w7,in2); // 0.3u 0.07u
    pmos #(2) pmos_8(sum,in1,in2); // 0.5u 0.07u
    nmos #(2) nmos_9(w7,vss,in1); // 0.3u 0.07u
    pmos #(2) pmos_10(w7,vdd,in1); // 0.5u 0.07u
endmodule

// Simulation parameters in Verilog Format
always
#200 in1=~in1;
#400 in2=~in2;

// Simulation parameters
// in1 CLK 1 1
// in2 CLK 2 2
```

Information

Module name (8 char. max)
sumator

Add gate delay info
 Append simul. infomations
 Add labels as comments

The Verilog file has 28 lines
The design includes 20 symbols
The circuit has 8 nodes

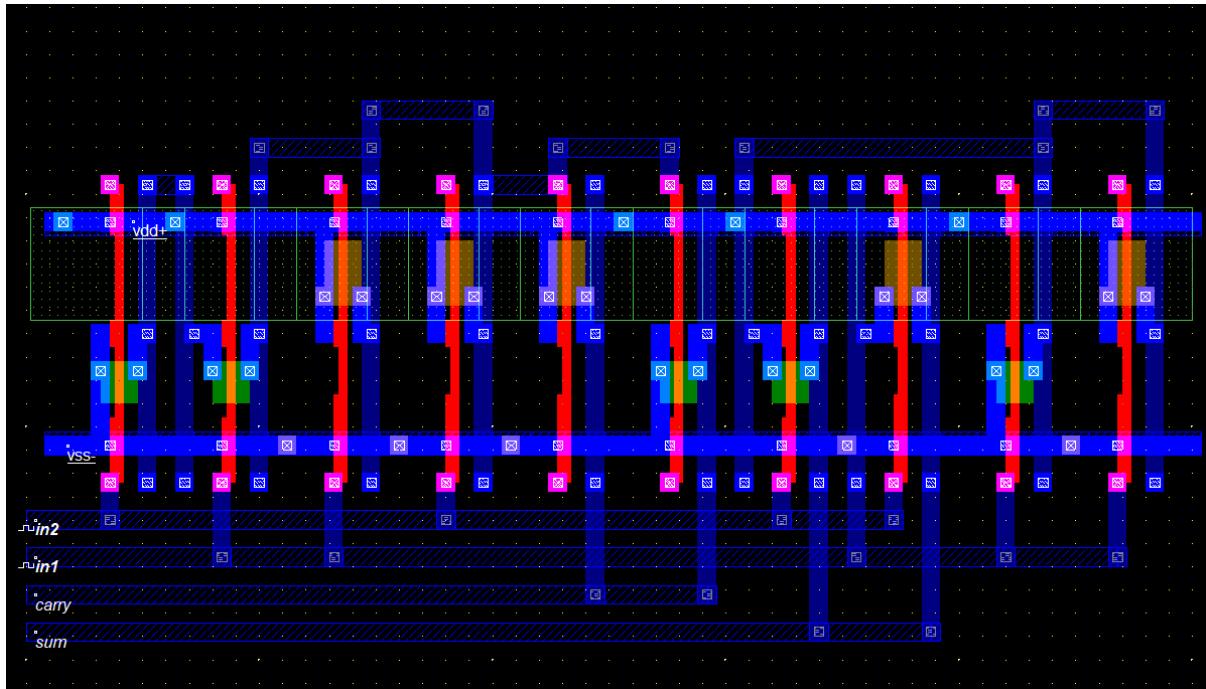
Misc.

Time scale : 1.00
Max clocks: 16

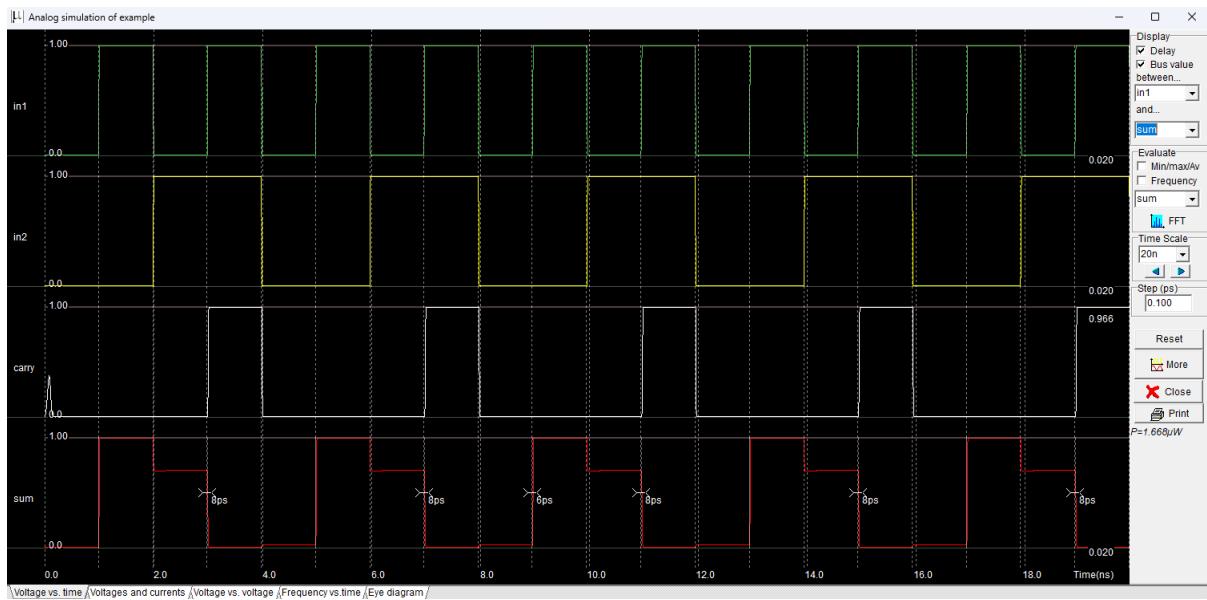
Update Verilog Extract circuit

OK

- Microwind

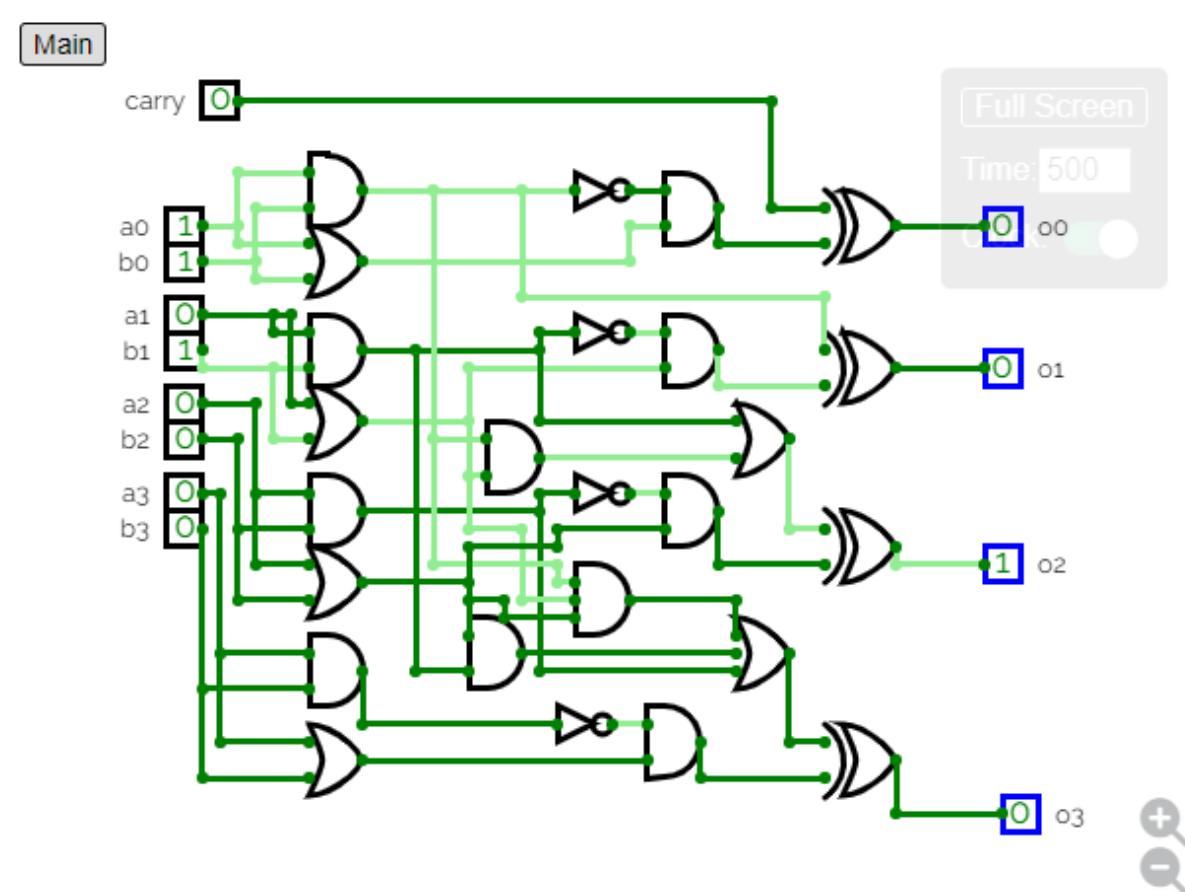


- Simulare Microwind



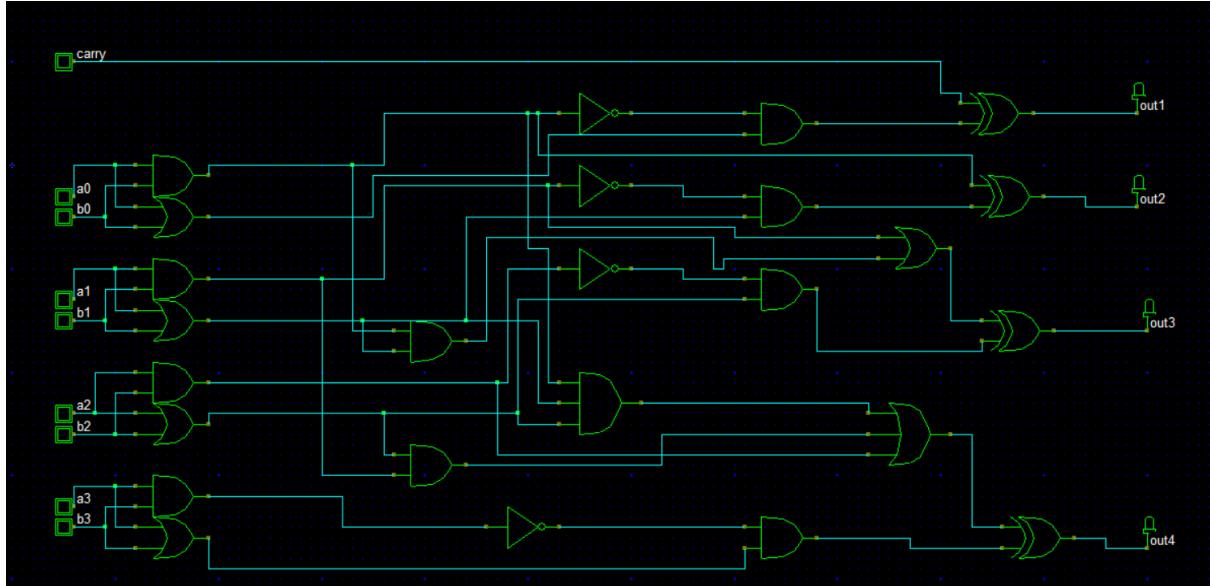
6. Sumator 4 biti Carry Look Ahead

- Schema circuit



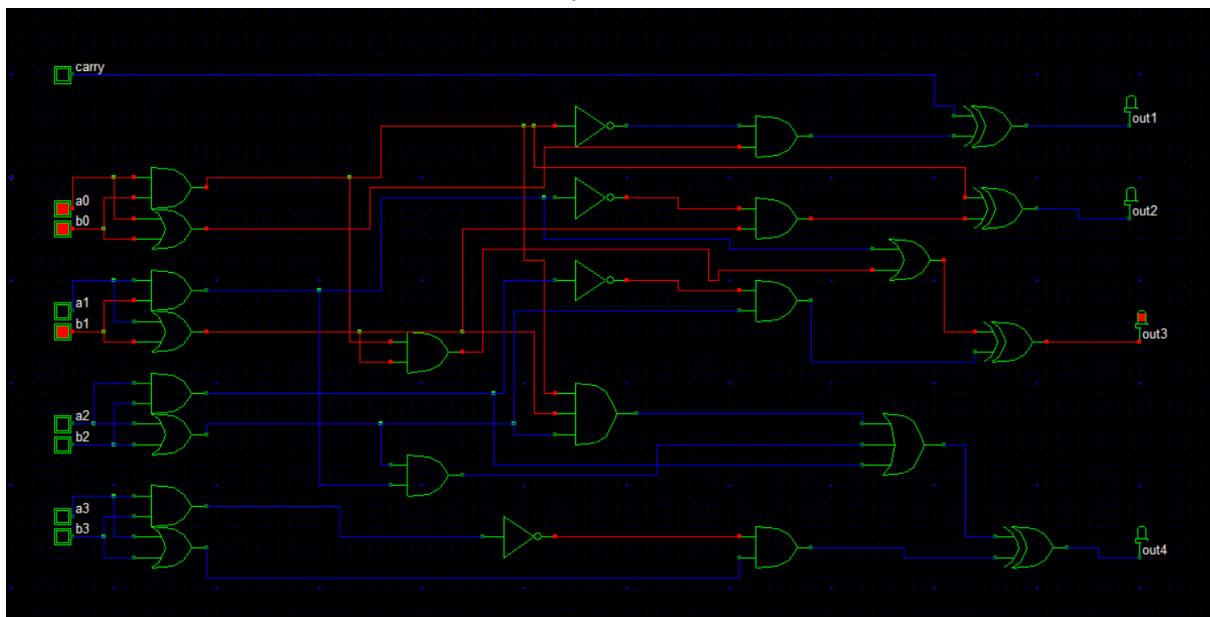
Schema preluata din
<https://circuitverse.org/users/13253/projects/4-bit-carry-lookahead-adder-f091ffb8-59db-4a74-b35f-c5d70c07dfbe>

- Schema DSCH

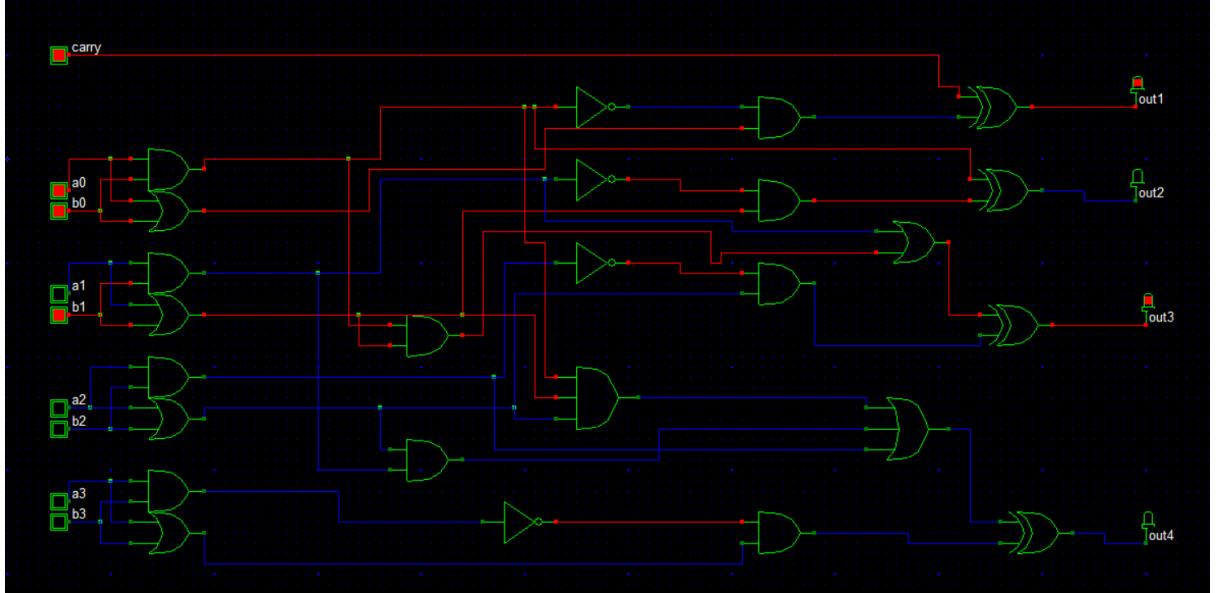


- Simulare DSCH

a0a1a2a3 1000 b0b1b2b3 1100 carry = 0 => out1out2out3out4 0010



a0a1a2a3 1000 b0b1b2b3 1100 carry = 1 => out1out2out3out4 1010



- Cod Verilog

Verilog | Hierarchy | Netlist | Critical path |

```
// Verilog, Hierarchy and Netlist
Verilog | Hierarchy | Netlist | Critical path |
// DSCH 3.5
// 5/9/2023 10:24:07 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\ADDER_CLA_4_BUN.sch

module ADDER_CLA_4_BUN(carry,a0,b0,a1,b1,a2,b2,a3,
b3,out1,out2,out3,out4);
input carry,a0,b0,a1,b1,a2,b2,a3;
input b3;
output out1,out2,out3,out4;
wire w11,w12,w13,w14,w15,w16,w17,w18;
wire w19,w20,w21,w22,w23,w24,w25,w26;
wire w27,w28,w29,w30,w31;
and #(5) and2_1(w11,b0,a0);
and #(4) and2_2(w12,b1,a1);
and #(4) and2_3(w13,b2,a2);
and #(4) and2_4(w14,b3,a3);
or #(3) or2_5(w15,a0,a1);
or #(4) or2_6(w16,a1,b1);
or #(4) or2_7(w17,a2,b2);
or #(3) or2_8(w18,a3,b3);
and #(3) and2_9(w19,w16,w11);
and #(3) and2_10(w20,w12,w17);
and #(3) and2_11(w21,w11,w16,w17);
not #(1) inv_12(w22,w14);
not #(1) inv_13(w23,w13);
not #(1) inv_14(w24,w12);
not #(1) inv_15(w25,w11);
and #(3) and2_16(w26,w15,w25);
and #(3) and2_17(w27,w16,w24);
and #(3) and2_18(w28,w17,w23);
and #(3) and2_19(w29,w18,w22);
or #(4) or3_20(w30,w21,w20,w13);
or #(3) or2_21(w31,w12,w19);
xor #(3) xor2_22(out1,w30,w29);
xor #(3) xor2_23(out2,w31,w28);
xor #(3) xor2_24(out2,w11,w27);
xor #(3) xor2_25(out1,carry,w26);
endmodule

// Simulation parameters in Verilog Format
always
#200 carry=~carry;
#400 a0=a0;
#800 b0=b0;
#1600 a1=a1;
#3200 a2=a2;
#6400 a3=a3;
#12800 b2=b2;
#25600 a3=a3;
#51200 b3=b3;

// Simulation parameters
// carry CLK 1
// a0 CLK 2
// a1 CLK 8
// a2 CLK 16
// a3 CLK 32
// b0 CLK 4
// b1 CLK 16
// b2 CLK 64
// b3 CLK 256
// a0 CLK 128
```

Information

- Module name (8 char max): ADDER_CLA_4
- Add gate delay info:
- Append simul. informations:
- Add labels as comments:

The Verilog file has 61 lines
The design includes 38 symbols
The circuit has 35 nodes

Misc.

- Time scale: 1.00
- Max clocks: 16

Update Verilog Extract circuit

OK

Information

- Module name (8 char max): ADDER_CLA_4
- Add gate delay info:
- Append simul. informations:
- Add labels as comments:

The Verilog file has 61 lines
The design includes 38 symbols
The circuit has 35 nodes

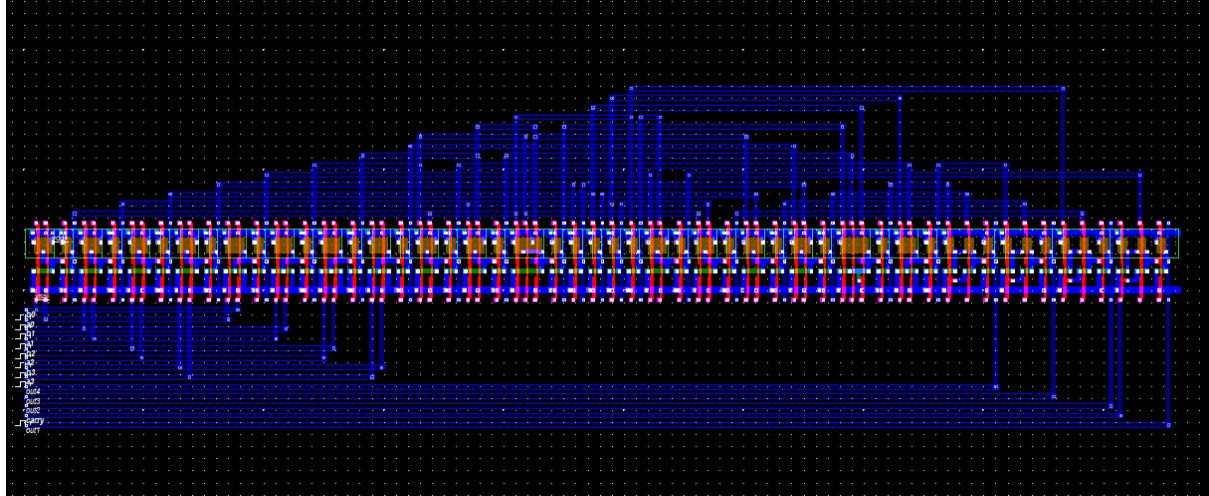
Misc.

- Time scale: 1.00
- Max clocks: 16

Update Verilog Extract circuit

OK

- Microwind

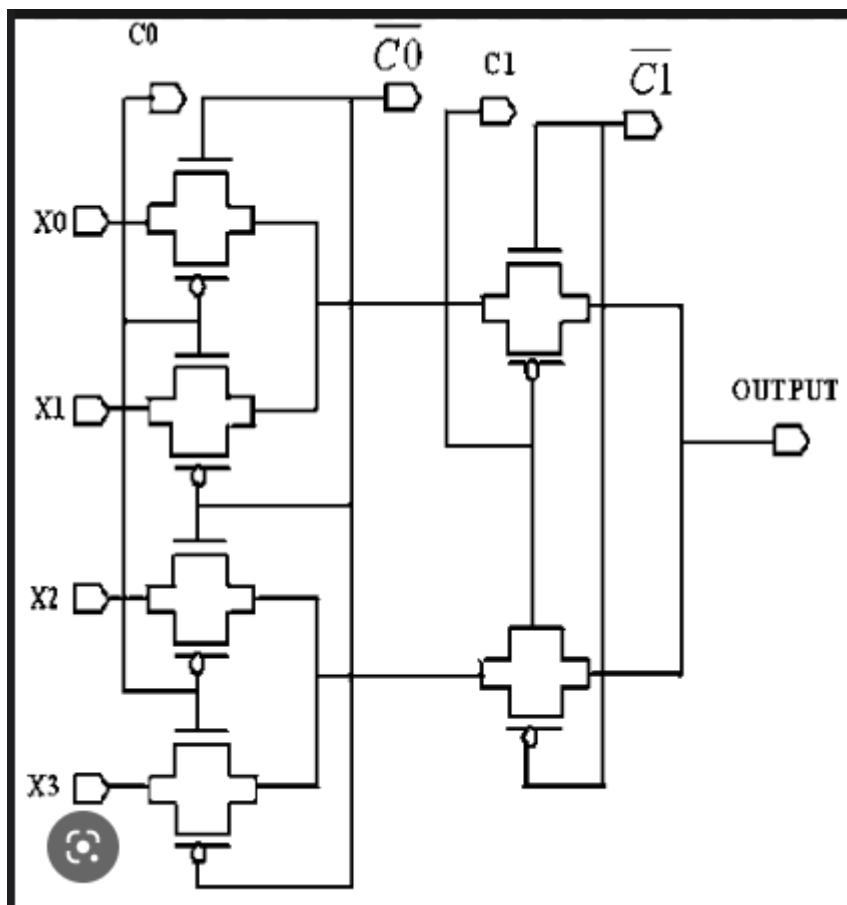


- Simulare Microwind



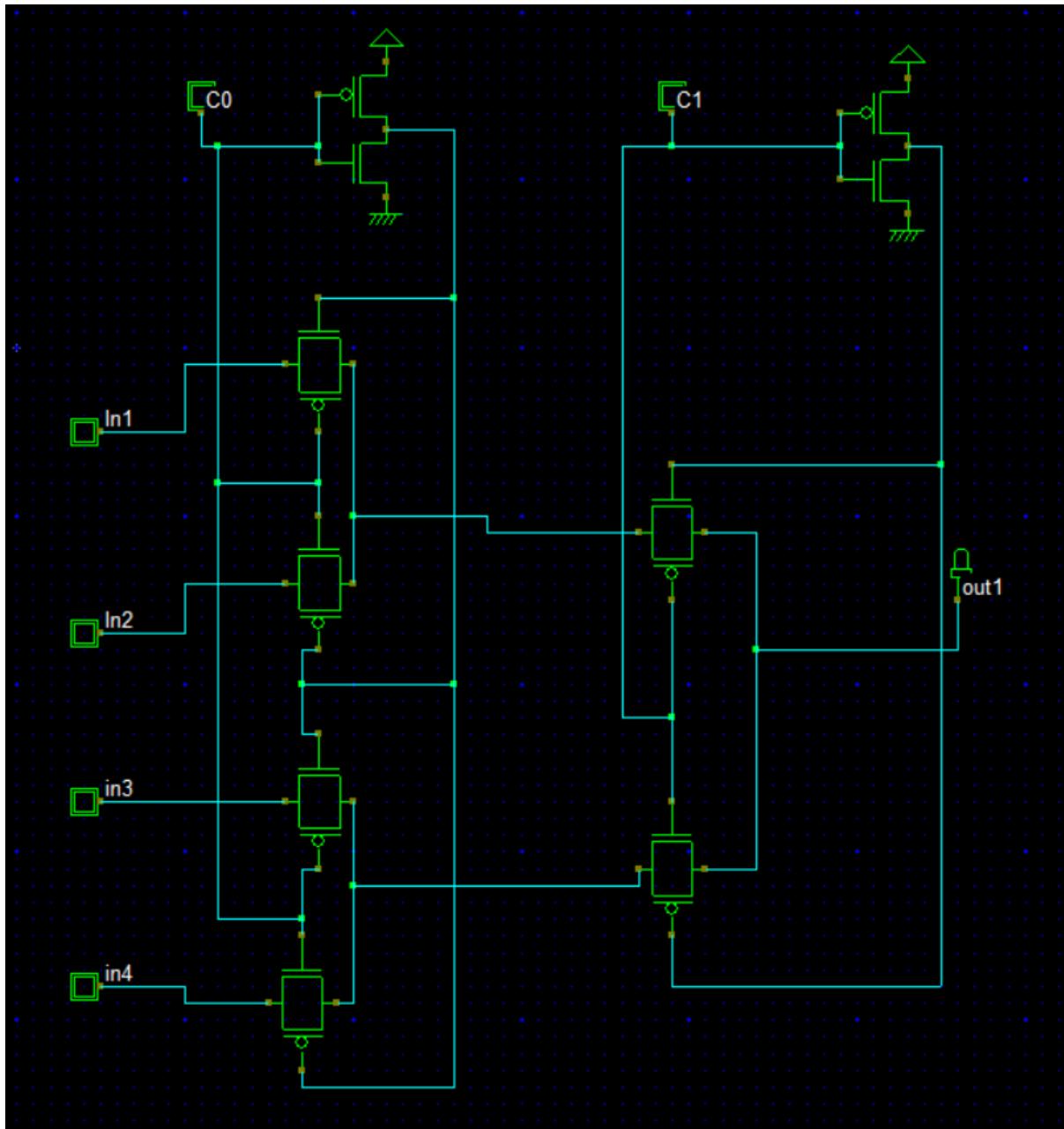
7. MUX 4:1

- schema circuitului

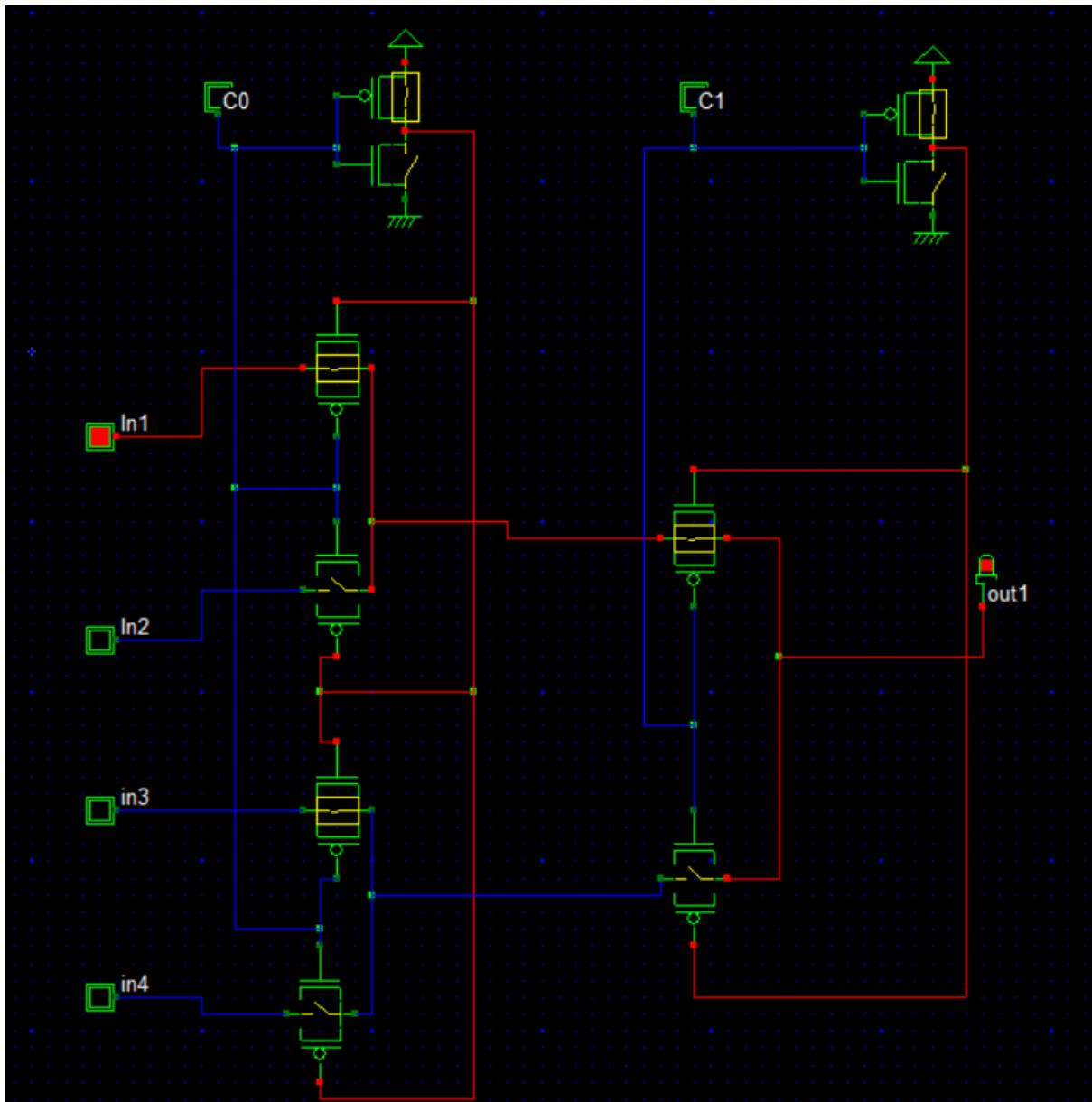


https://www.researchgate.net/figure/Transmission-gate-based-41-MUX_fig5_257799438

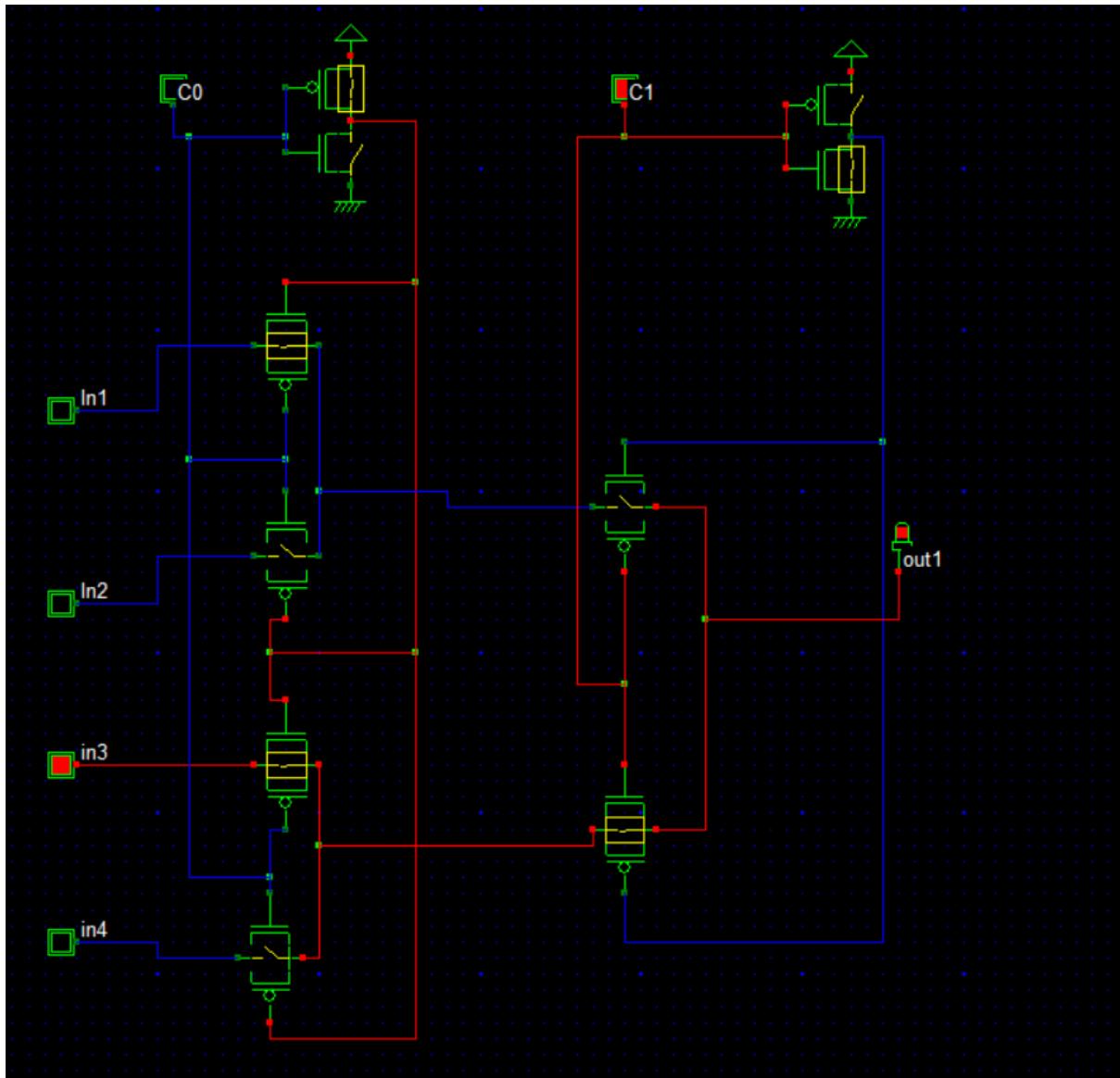
- schema dsch



- simulare circuit
 $in_1 = 1, in_2 = in_3 = in_4 = c_0 = c_1 = 0 \Rightarrow out = 1$



in1 = in2 = in4 = c1 = 0 in3 = 1 c0 = 1 => out = 1



- cod verilog

Verilog, Hierarchy and Netlist

Verilog | Hierarchy | Netlist | Critical path |

```
// DSCH 3.5
// 5/9/2023 11:12:58 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\Mux4-1.sch

module Mux41( In1,In2,in3,in4,C0,C1,out1);
    input In1,In2,in3,in4,C0,C1;
    output out1;
    wire w9,w10,w11,w12;
    nmos #(3) nmos_1(w10,In1,w9); // 0.3u 0.07u
    nmos #(3) nmos_2(w10,In2,C0); // 0.3u 0.07u
    nmos #(3) nmos_3(w11,in3,w9); // 0.3u 0.07u
    nmos #(3) nmos_4(w11,in4,C0); // 0.3u 0.07u
    nmos #(3) nmos_5(out1,w10,w12); // 0.3u 0.07u
    nmos #(3) nmos_6(out1,w11,C1); // 0.3u 0.07u
    pmos #(3) pmos_7(out1,w11,w12); // 0.5u 0.07u
    pmos #(3) pmos_8(out1,w10,C1); // 0.5u 0.07u
    pmos #(3) pmos_9(w10,In1,C0); // 0.5u 0.07u
    pmos #(3) pmos_10(w10,In2,w9); // 0.5u 0.07u
    pmos #(3) pmos_11(w11,in3,C0); // 0.5u 0.07u
    pmos #(3) pmos_12(w11,in4,w9); // 0.5u 0.07u
    pmos #(2) pmos_13(w12,vdd,C1); // 0.5u 0.07u
    pmos #(3) pmos_14(w9,vdd,C0); // 0.5u 0.07u
    nmos #(2) nmos_15(w12,vss,C1); // 0.3u 0.07u
    nmos #(3) nmos_16(w9,vss,C0); // 0.3u 0.07u
endmodule

// Simulation parameters in Verilog Format
always
#200 In1=~In1;
#400 In2=~In2;
#800 in3=~in3;
#1600 in4=~in4;
#3200 C0=~C0;
#6400 C1=~C1;

// Simulation parameters
// In1 CLK 1 1
// In2 CLK 2 2
// in3 CLK 4 4
// in4 CLK 8 8
// C0 CLK 16 16
// C1 CLK 32 32
```

Information

Module name (8 char. max)

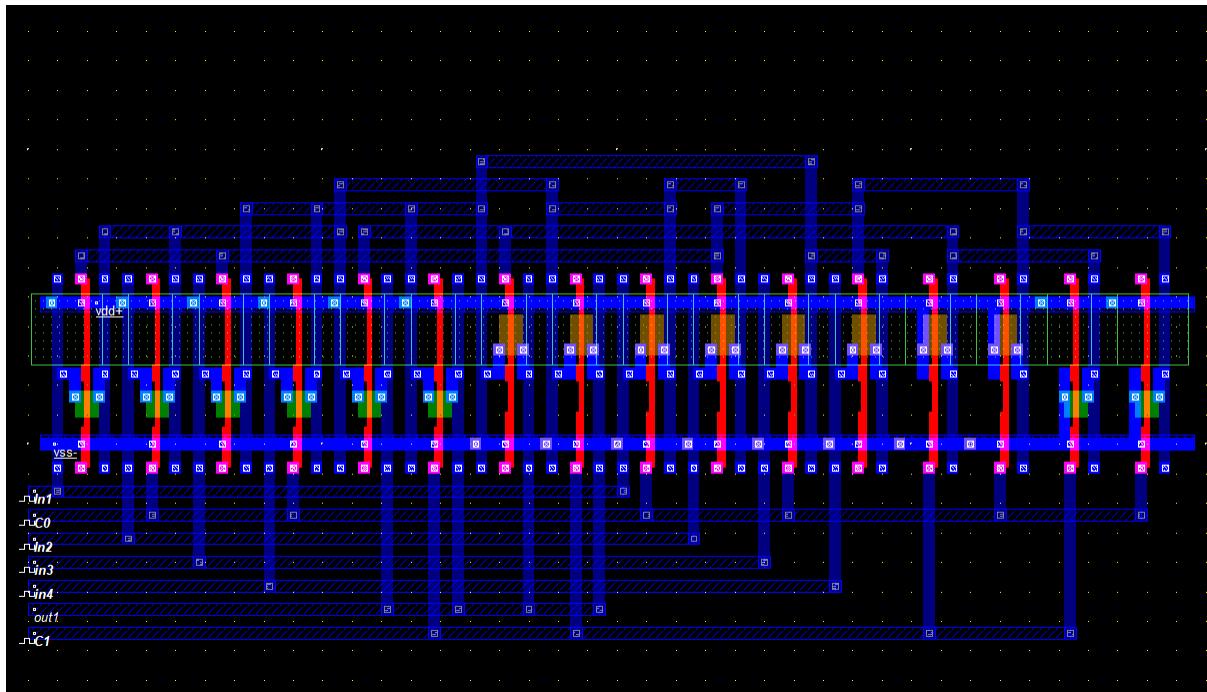
Add gate delay info
 Append simul. infomations
 Add labels as comments

The Verilog file has 42 lines
The design includes 27 symbols
The circuit has 12 nodes

Misc.

Time scale:
Max clocks:

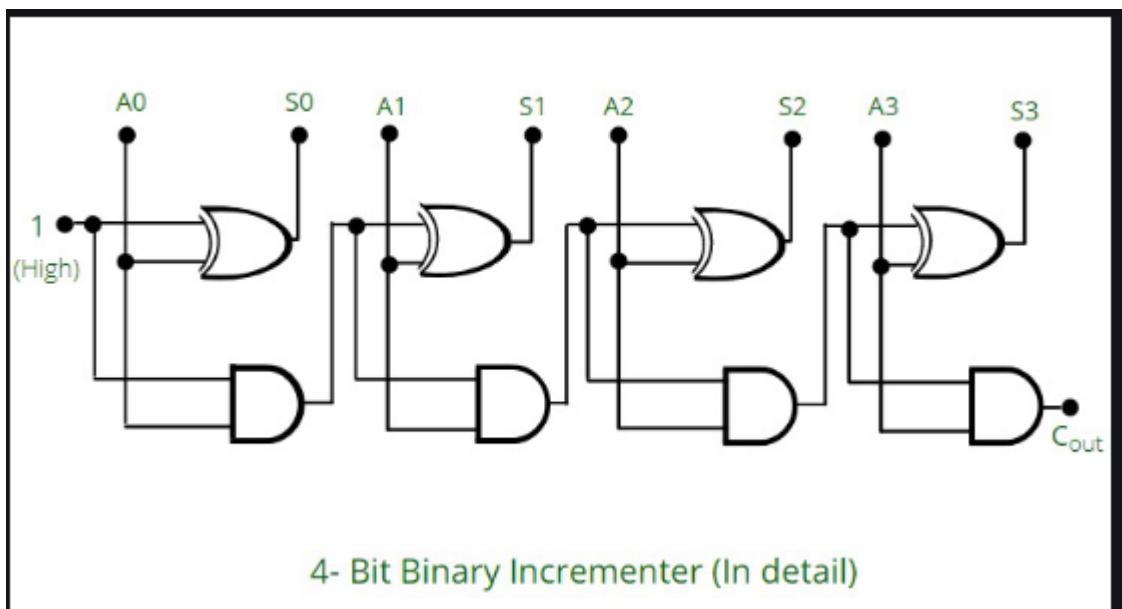
- microwind



- simulare microwind

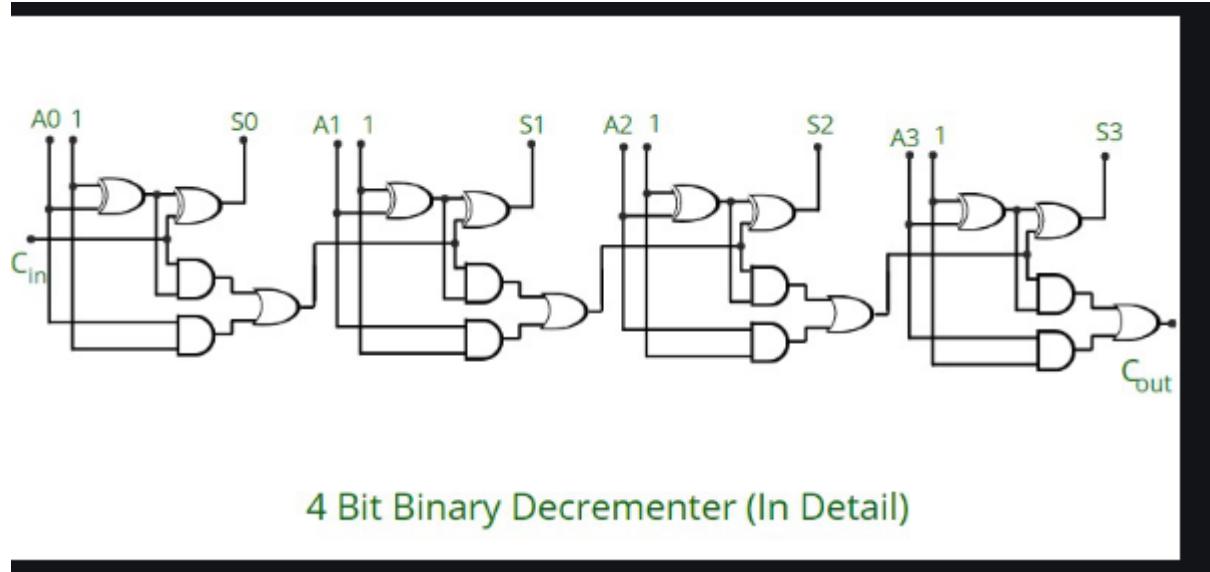
8. Incrementator / Decrementator 4 biti

- schema circuit



4 semi-sumatoare serie

<https://www.geeksforgeeks.org/4-bit-binary-incrementer/>

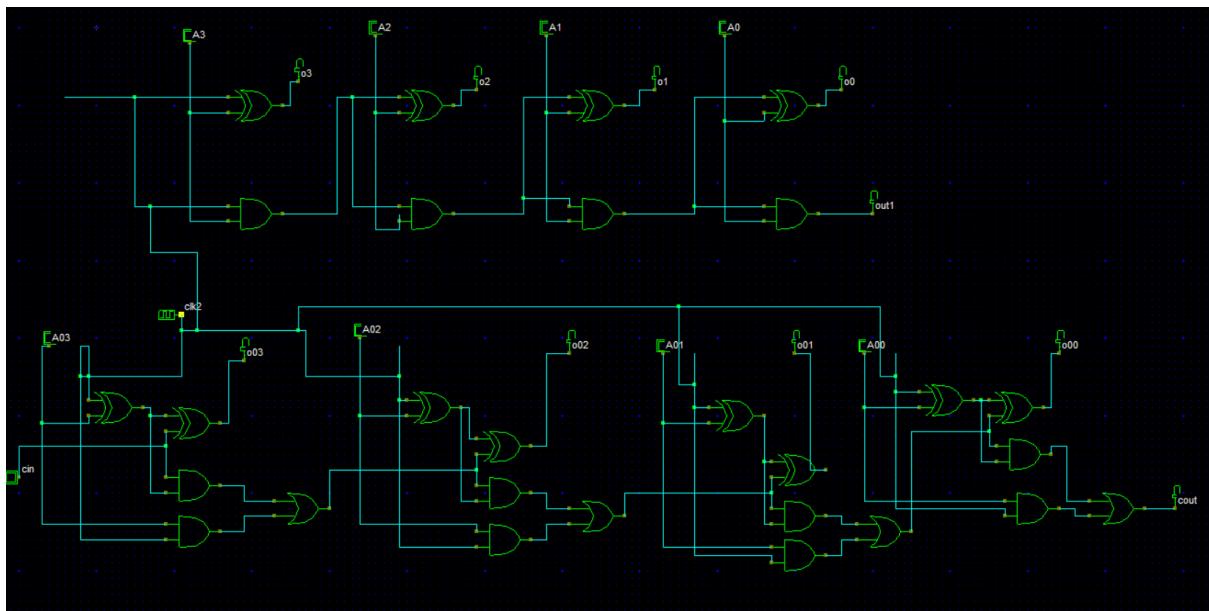


4 Bit Binary Decrementer (In Detail)

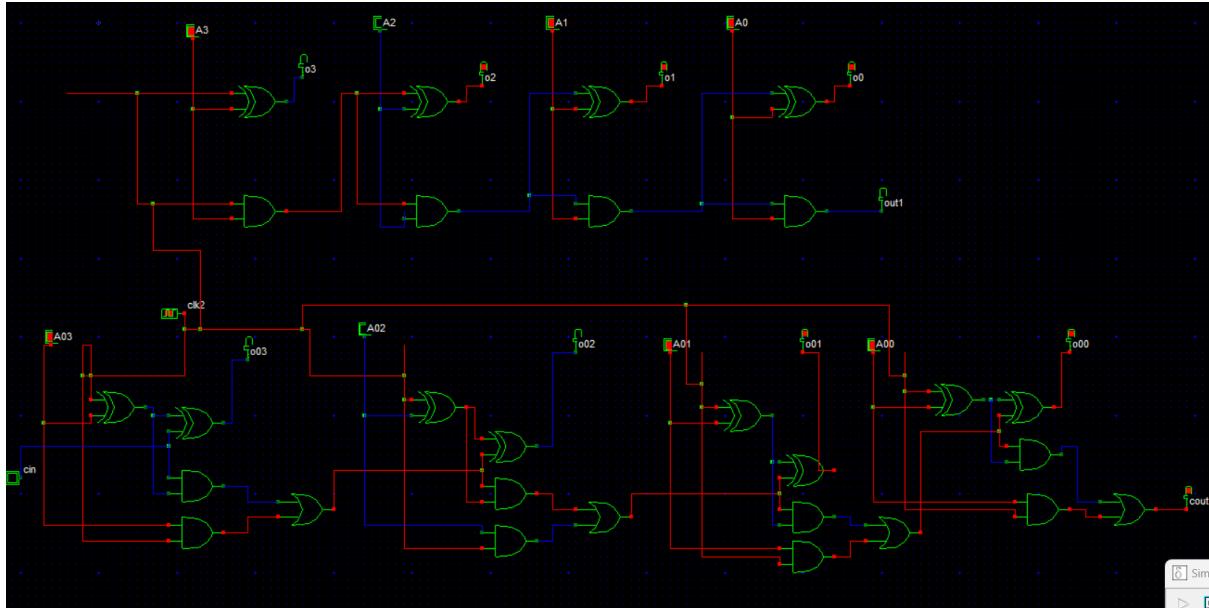
4 sumatoare serie

<https://www.geeksforgeeks.org/4-bit-binary-decrementer/>

- schema dsch



- simulare dsch



la incrementor => de la dreapta la stanga $1101 + 1 = 1110$

la decrementor => de la dreapta la stanga $1101 + 1111 = 1100$

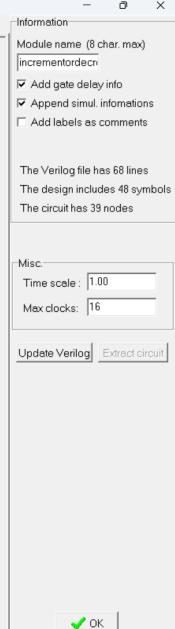
- cod verilog

```

Verilog, Hierarchy and Netlist | Critical path
Verilog | Hierarchy | Netlist | Critical path
// DSCH 3.5
// 5/10/2023 6:33:06 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\incrementor-decrementor.sch
module incrementordecrementor( A3,A0,A1,A2,clk2,cin,A03,A02,
A01,A00,out1,o1,o2,o3,o00,
o01,o02,o03,cout);
input A3,A0,A1,A2,clk2,cin,A03,A02;
input A01,A00;
output out1,o1,o2,o3,o00,o01,o02;
output o03,cout;
wire w5,w8,w11,w17,w21,w22,w29;
wire w30,w31,w32,w33,w34,w35,w36,w37;
wire w39,w40;
xor # (3) xor2_1(w3,clk2,A3);
xor # (3) xor2_2(w3,clk2,A0);
xor # (3) xor2_3(w1,clk2,A1);
xor # (3) xor2_4(w0,w11,A0);
and # (4) and2_5(w5,A3,clk2);
and # (4) and2_6(w6,A2,w5);
and # (4) and2_7(w11,A1,w8);
and # (3) and2_8(out1,A0,w11);
xor # (3) xor2_9(o00,w17,w18);
or # (3) or2_10(cout,w21,w22);
and # (4) and2_11(w22,clk2,A00);
and # (3) and2_12(w22,w23,w28);
xor # (4) xor2_13(w28,clk2,A03);
xor # (4) xor2_14(w17,clk2,A00);
and # (3) and2_15(w30,w29,cin);
and # (3) and2_16(w31,clk2,A03);
or # (4) or2_17(w32,w30,w31);
xor # (4) xor2_18(w33,clk2,A02);
xor # (3) xor2_19(o03,w29,cin);
xor # (3) xor2_20(o02,w33,w32);
and # (4) and2_21(w34,w33,w32);
and # (3) and2_22(w34,w35,w30);
or # (4) or2_23(w36,w34,w55);
xor # (4) xor2_24(w37,clk2,A01);
xor # (3) xor2_25(o01,w37,w36);
and # (3) and2_26(w38,w37,w36);
and # (3) and2_27(w39,clk2,A01);
or # (4) or2_28(w18,w38,w39);
endmodule

// Simulation parameters in Verilog Format
always
#200 A3=A3;
```

```



Verilog, Hierarchy and Netlist | Critical path | Information

```

xor #(3) xor2_9(w00,w17,w18);
or #(4) or2_10(w01,w21,w17);
and #(3) and2_11(w22,w23,w00);
and #(3) and2_12(w21,w17,w18);
xor #(4) xor2_13(w29,clk2,A03);
xor #(4) xor2_14(w17,clk2,A00);
and #(3) and2_15(w30,w29,cin);
and #(3) and2_16(w31,clk2,A03);
or #(4) or2_17(w32,w30,w31);
xor #(4) xor2_18(w33,clk2,A02);
xor #(3) xor2_19(w03,w29,cin);
xor #(3) xor2_20(w34,w35,w32);
and #(3) and2_21(w34,w35,w32);
and #(3) and2_22(w35,clk2,A02);
or #(4) or2_23(w36,w34,w35);
xor #(4) xor2_24(w37,clk2,A01);
xor #(3) xor2_25(w01,w37,w36);
and #(3) and2_26(w38,w37,w36);
and #(3) and2_27(w39,clk2,A01);
or #(4) or2_28(w18,w38,w39);
endmodule

// Simulation parameters in Verilog Format
always
#200 A3=A3;
#400 A0=A0;
#800 A1=A1;
#1600 A2=A2;
#8000 clk2=clk2;
#3200 cin=cin;
#6400 A3=A3;
#12800 A0=A0;
#25600 A1=A1;
#51200 A2=A2;
// Simulation parameters
// A3 CLK 1 1
// A0 CLK 2 2
// A1 CLK 4 4
// P0 CLK 8 8
// CLK CLK 40 40
// Cin CLK 16 16
// P03 CLK 32 32
// P02 CLK 64 64
// A01 CLK 128 128
// A00 CLK 256 256

```

Module name (8 char. max)  
incrementordecn:  
 Add gate delay info  
 Append simul. informations  
 Add labels as comments

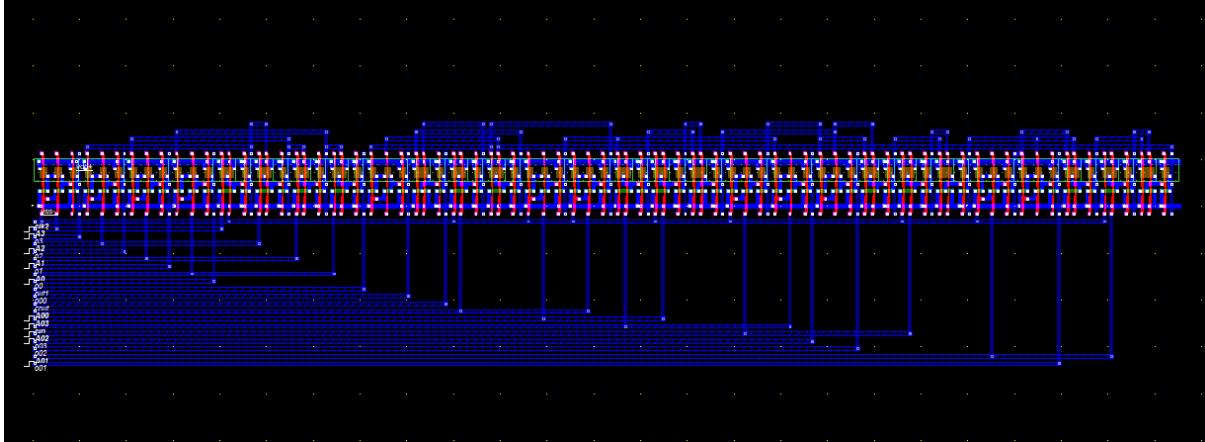
The Verilog file has 68 lines  
The design includes 48 symbols  
The circuit has 39 nodes

Misc.  
Time scale : 1.00  
Max clocks: 16

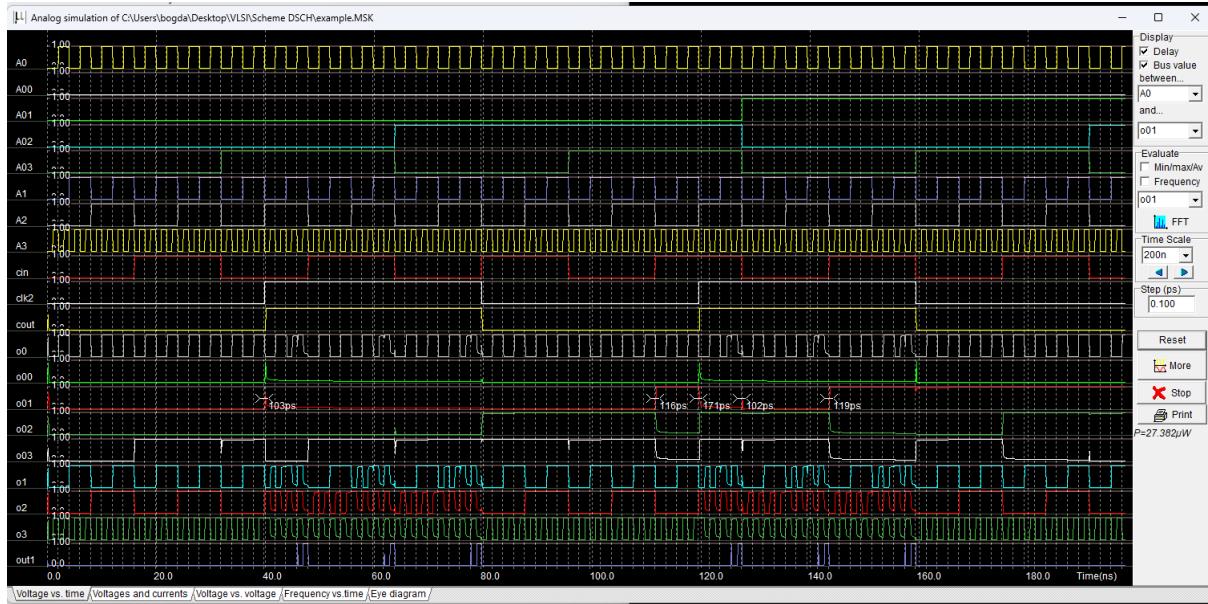
Update Verilog Extract circuit

OK

- microwind

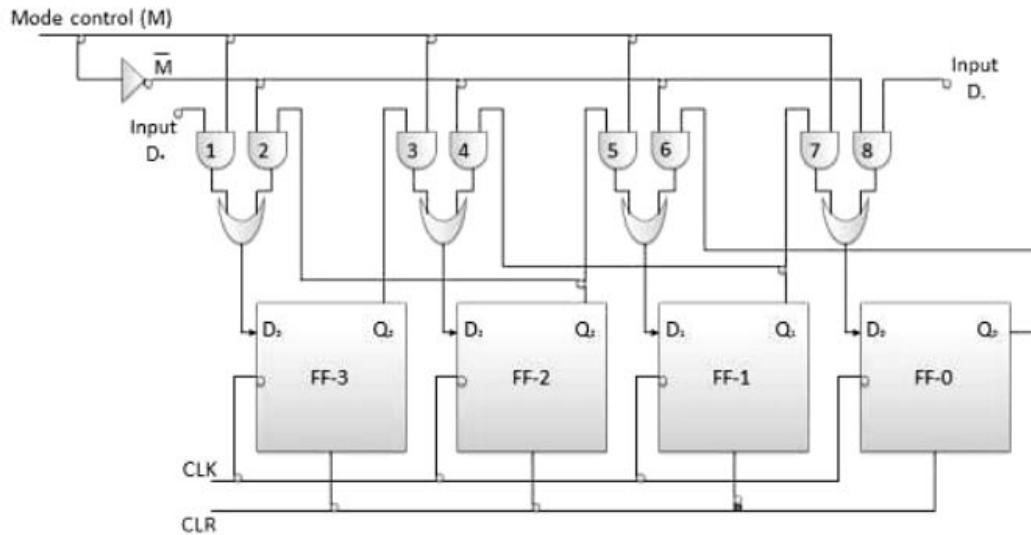


- simulare microwind



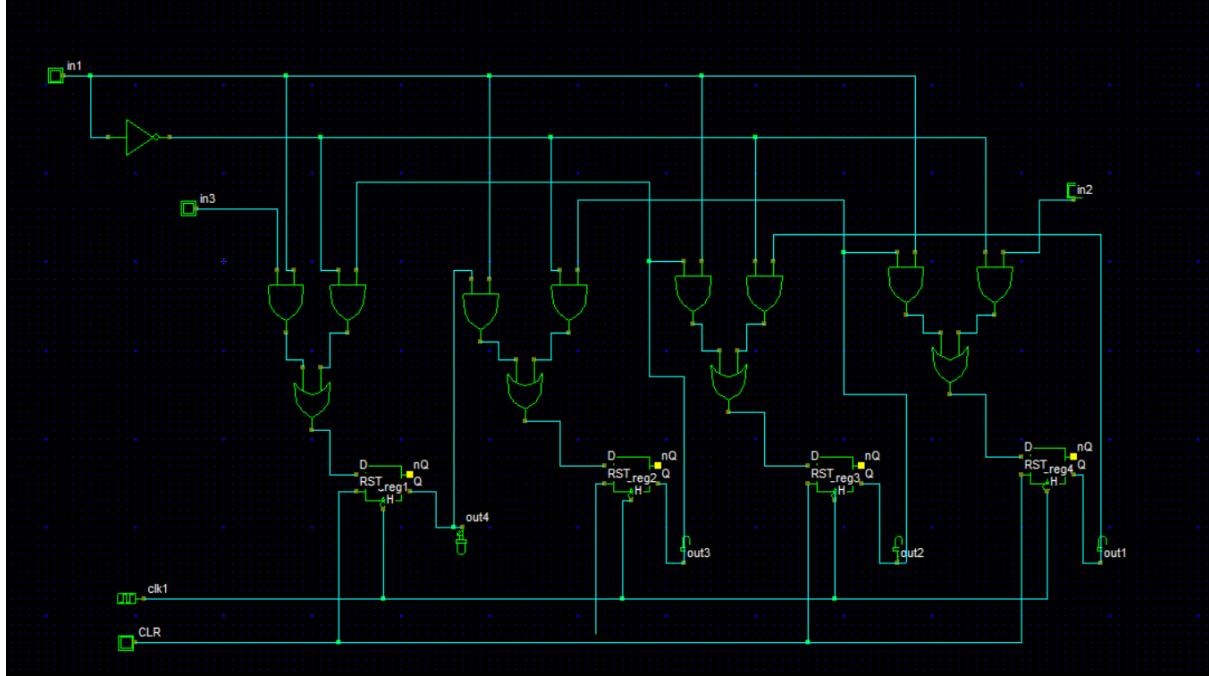
## 9. Registru deplasare stanga/dreapta pe 4 biti

- Schema circuitului



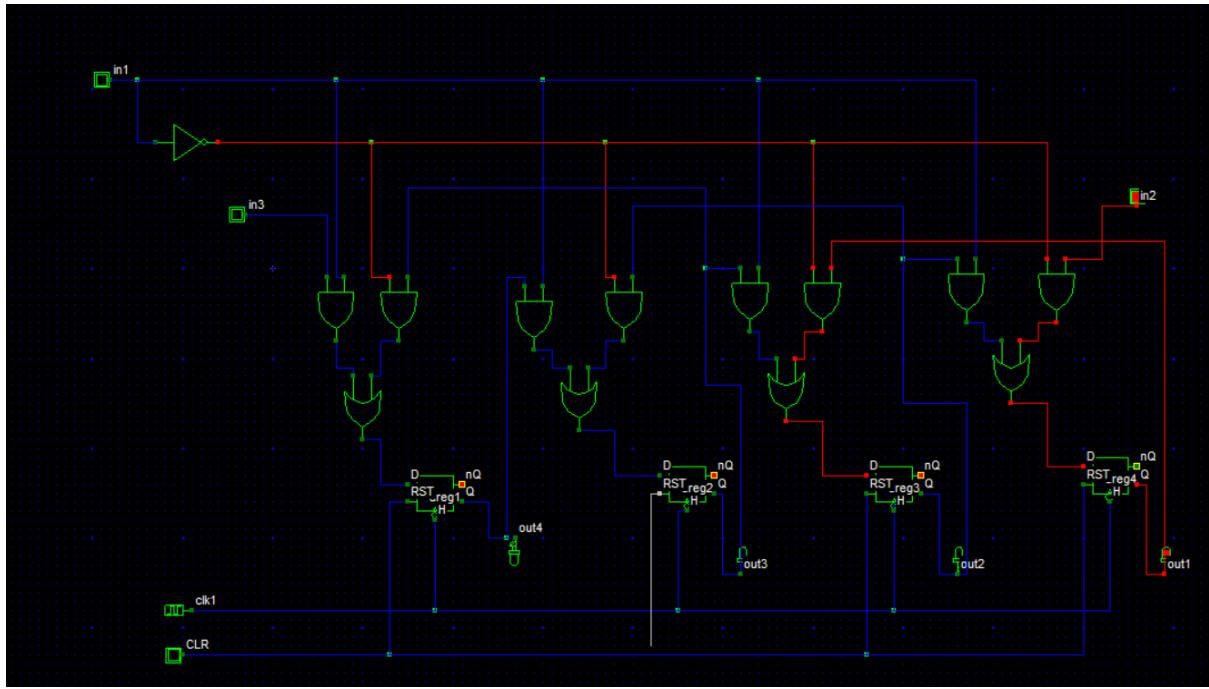
<https://www.ques10.com/p/6634/what-is-shift-register-explain-4-bit-bi-directional/>

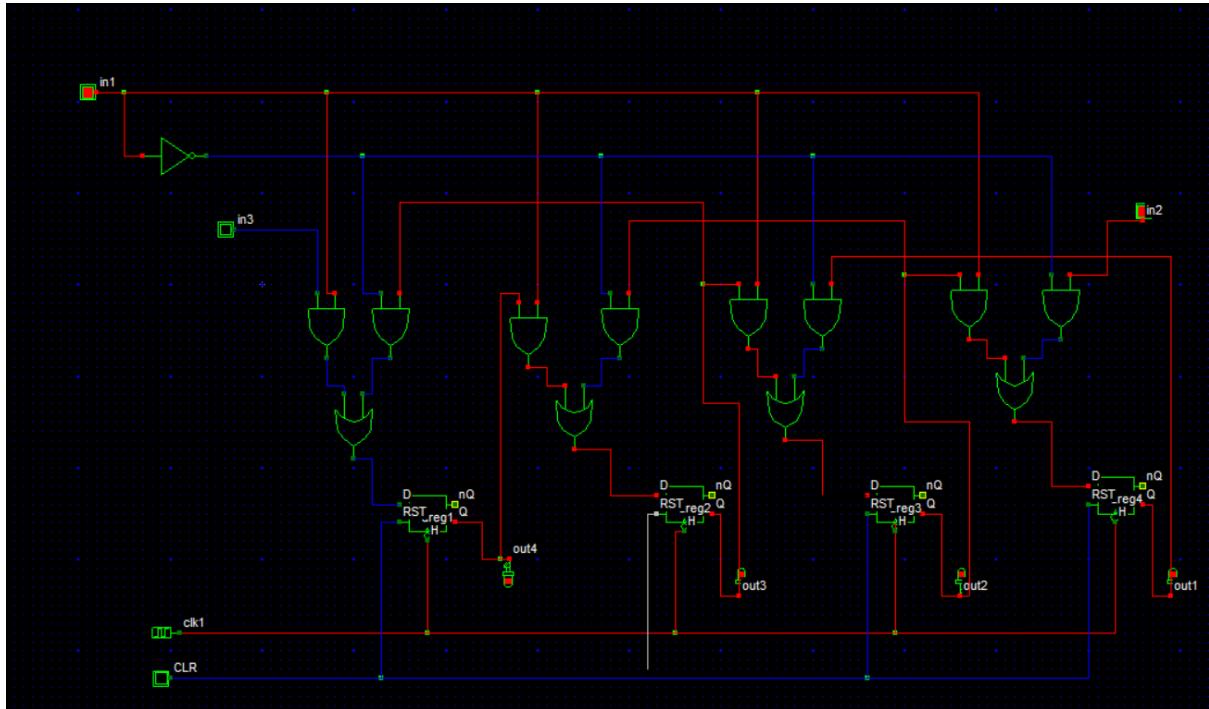
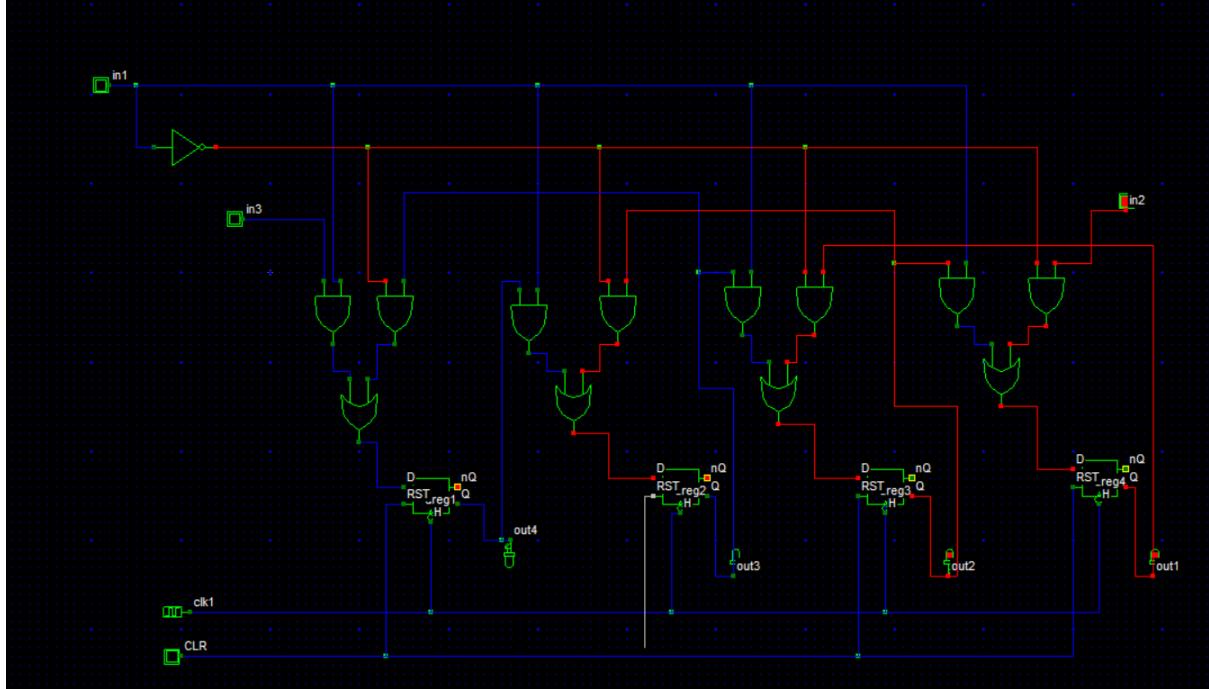
- Schema DSCH

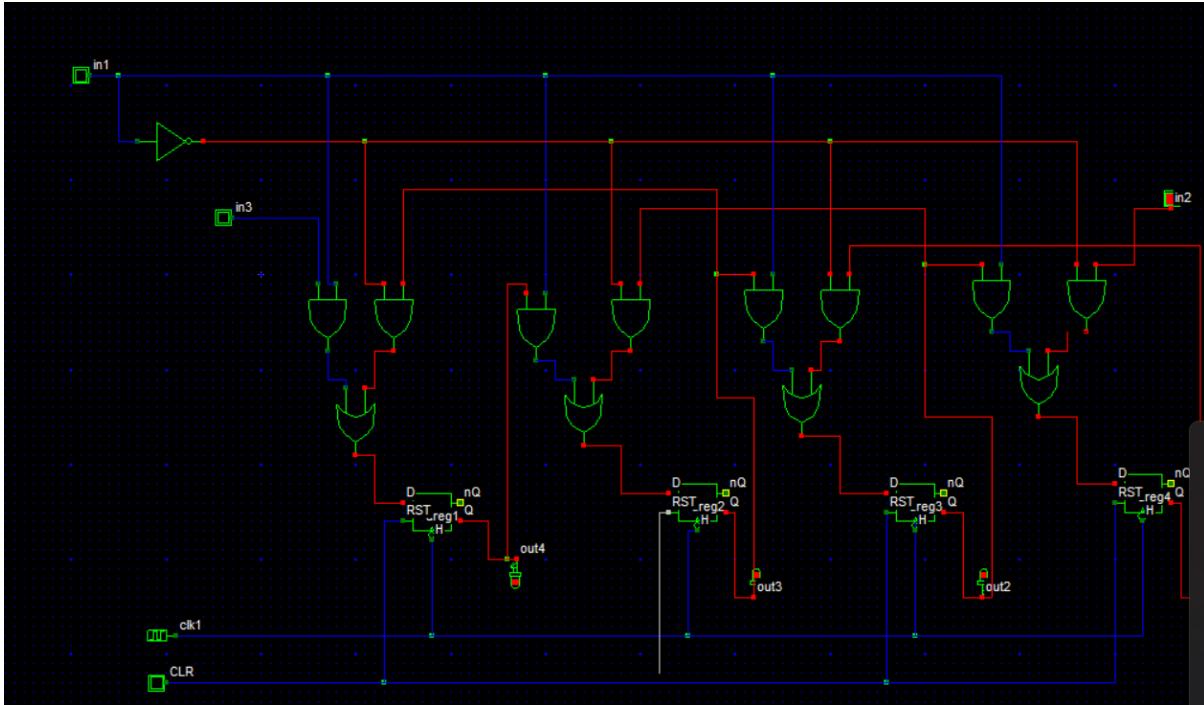


- Simulare DSCH

Pentru a rula aasam pe unul din inputuri. In3 pt dreapta si in2 pt stanga. apoi schimbam din in1 shiftarea.







- cod verilog

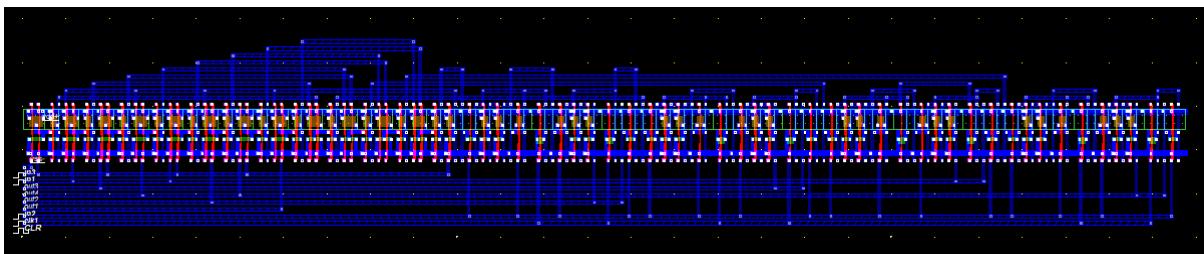
```

Verilog Hierarchy and Netlist
Verilog | Hierarchy | Netlist | Critical path
// DSCH 3.5
// 5/10/2023 7:57:29 PM
// C:\Users\bogda\Desktop\VLSI\Scheme DSCH\shifter.sch
module shifter(in1,in2,in3,clk1,CLR,out1,out2,out3,out4);
input in1,in2,in3,clk1,CLR;
output out1,out2,out3,out4;
wire w4,w5,w7,w9,w11,w12,w14,w15;
wire w17,w18,w19,w20,w21,w24,w25,w26;
wire w27,w28;
and #(3) and2_1(w4,in3,in1);
and #(3) and2_2(w7,w5,out3);
and #(3) and2_3(w9,out4,in1);
and #(3) and2_4(w11,w5,out2);
and #(3) and2_5(w12,out3,in1);
and #(3) and2_6(w14,w5,out1);
and #(3) and2_7(w15,out2,in1);
and #(3) and2_8(w17,w5,in2);
or #(3) or2_9(w17,w4);
or #(3) or2_10(w15,w14,in2);
or #(3) or2_11(w20,w19,w12);
or #(3) or2_12(w21,w17,w15);
not #(3) inv_13(w5,in1);
dreg #(4) dreg_14(out1,w24,w21,CLR,clk1);
dreg #(4) dreg_15(out4,w25,w18,CLR,clk1);
dreg #(4) dreg_16(out3,w27,w19,w26,clk1);
dreg #(4) dreg_17(out2,w28,w20,CLR,clk1);
endmodule

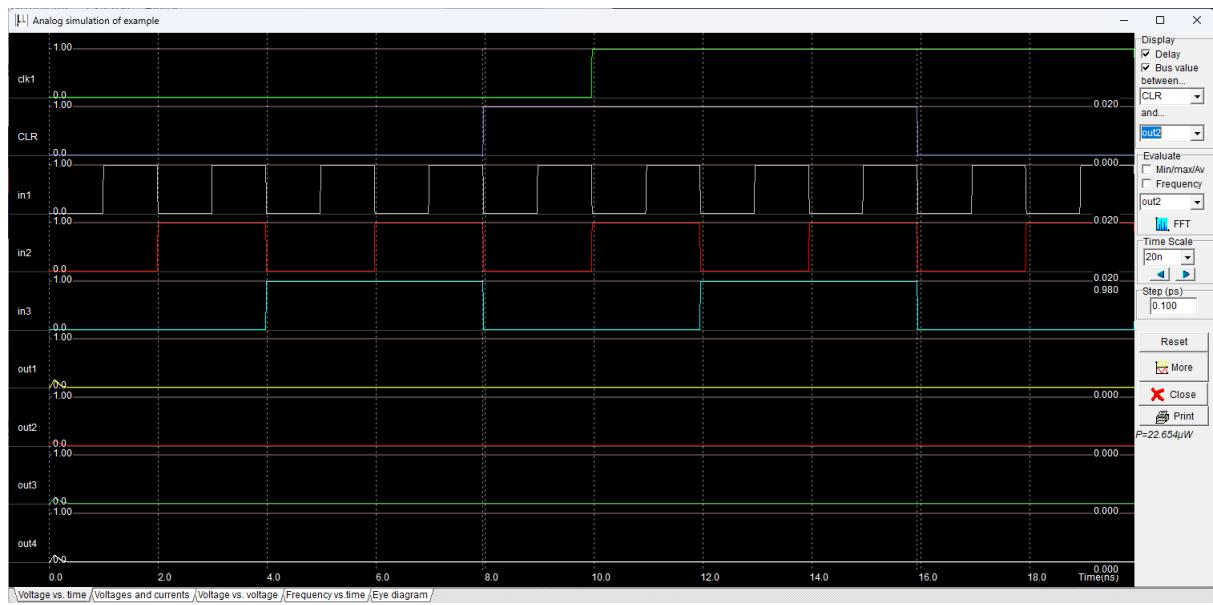
// Simulation parameters in Verilog Format
always
#200 in1=1;
#400 in2=1;
#800 in3=1;
#2000 clk1=1;
#1600 CLR=0;
// Simulation parameters
// in1 CLK 1
// in2 CLK 2
// in3 CLK 4
// clk1 CLK 10
// CLR CLK 8

```

- Microwind

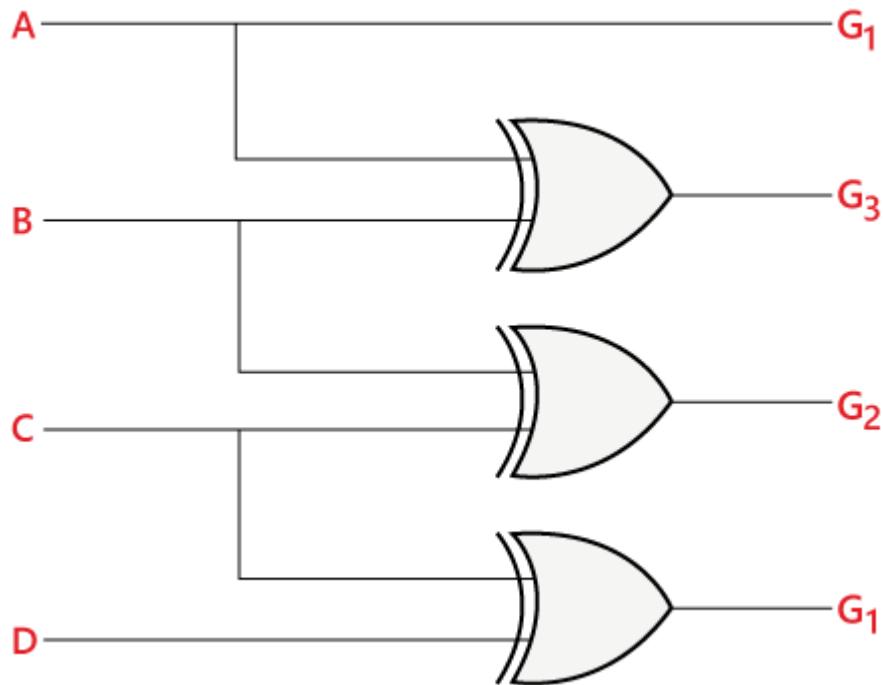


- Simulare Microwind



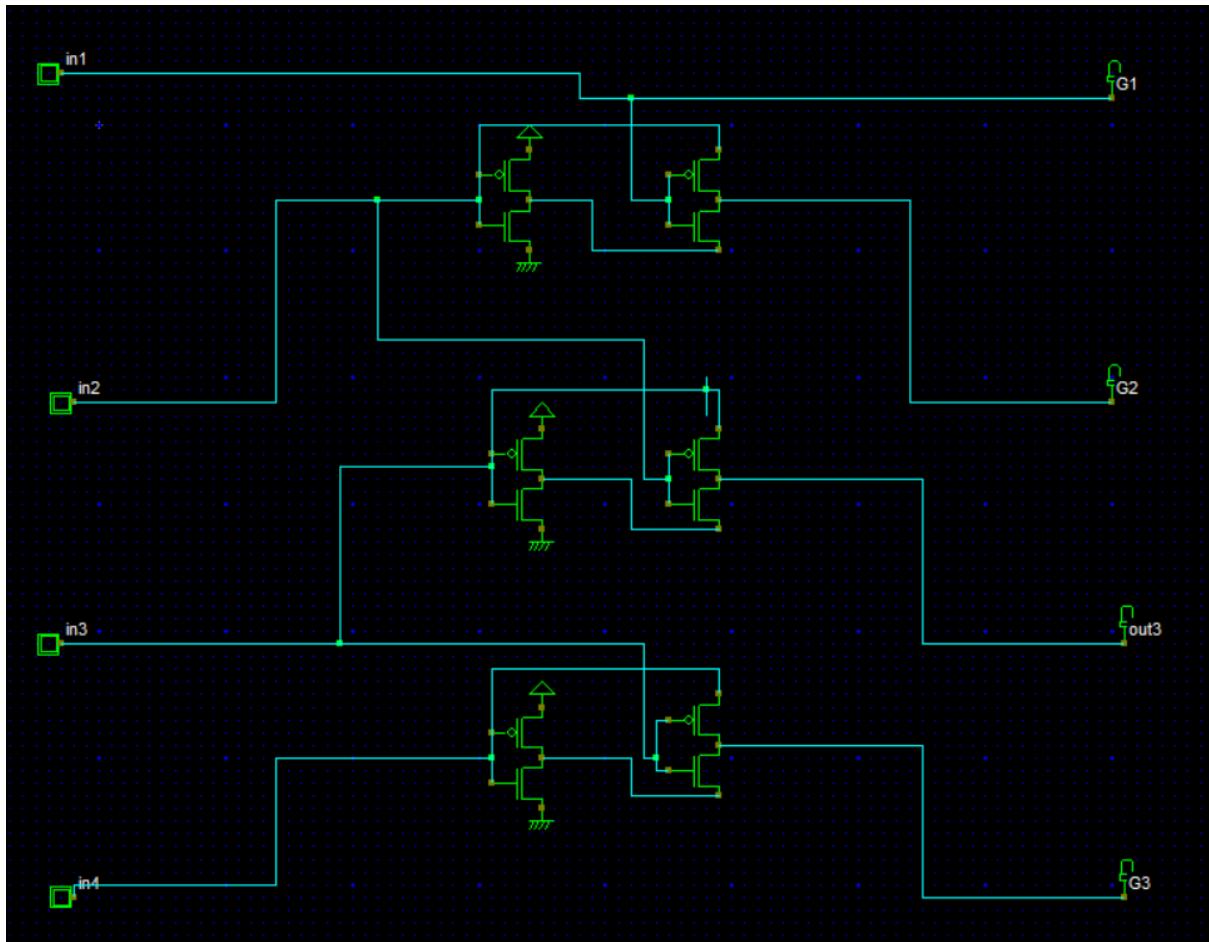
### 10. Convertor cod gray 4 biti

- schema circuit

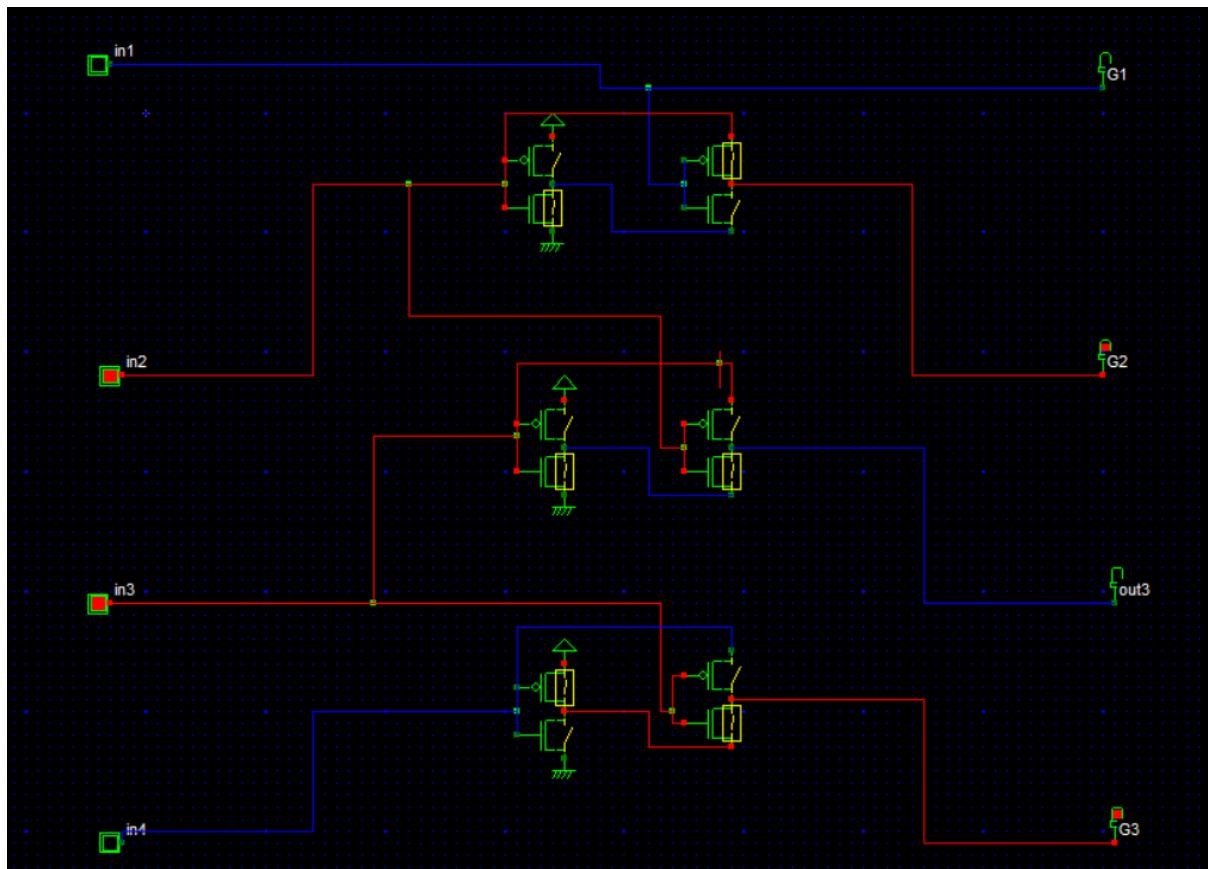


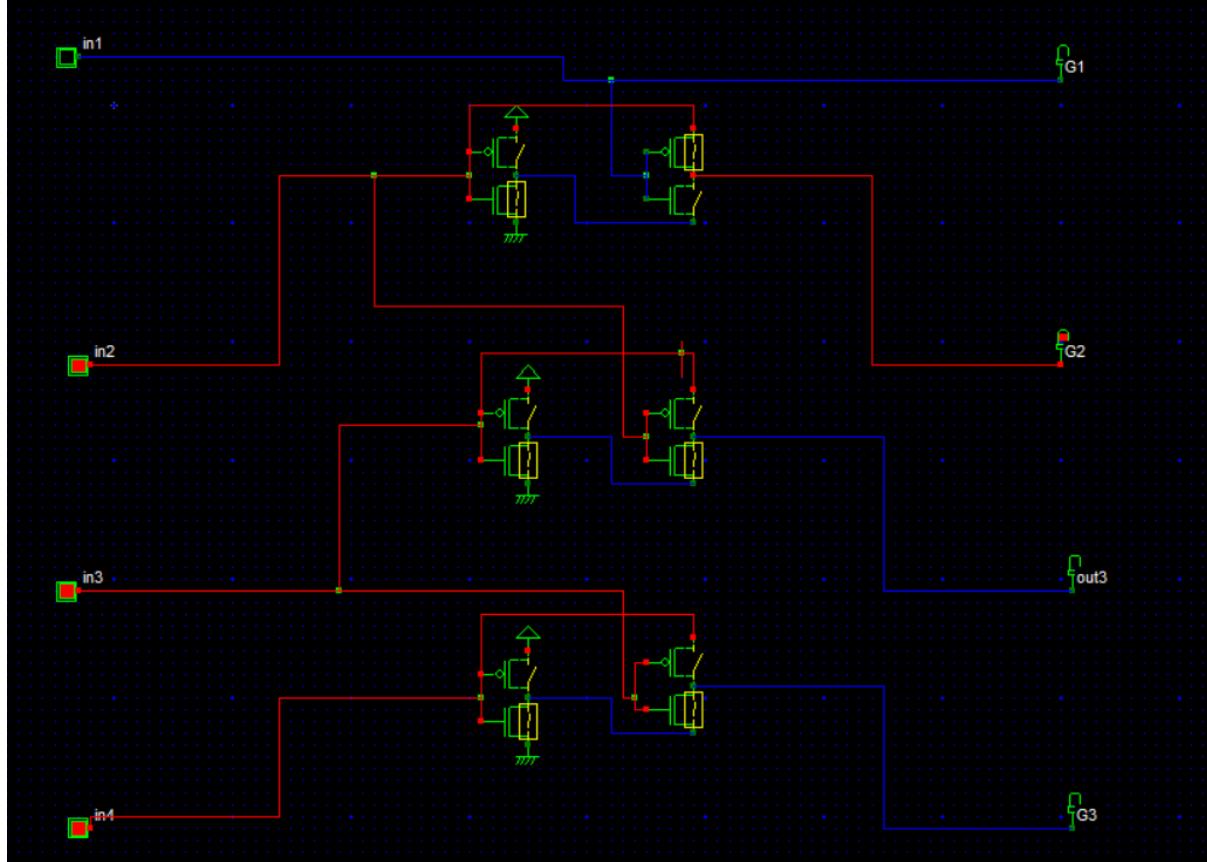
<https://www.javatpoint.com/binary-to-gray-code-conversion-in-digital-electronics>

- schema dsch



- simulare dsch





- cod verilog

```

// Verilog, Hierarchy and Netlist
[Venilog] Hierarchy | Netlist | Critical path |
// DSCH 3.5
// 5/10/2023 8:26:00 PM
// C:\Users\bogda\Desktop\VLISI\Scheme DSCH\convertorGrey.sch

module convertorGrey(in1,in2,in3,in4,G1,G2,out3,G3);
input in1,in2,in3,in4;
output G1,G2,out3,G3;
wire w5,w10,w11;
pmos #(2) pmos_1(w9,vdd,in2); // 0.5u 0.07u
nmos #(2) nmos_2(w9,vss,in2); // 0.3u 0.07u
pmos #(2) pmos_3(w10,vdd,in3); // 0.5u 0.07u
nmos #(2) nmos_4(w10,vss,in3); // 0.3u 0.07u
pmos #(2) pmos_5(w11,vdd,in4); // 0.5u 0.07u
nmos #(2) nmos_6(w11,vss,in4); // 0.3u 0.07u
pmos #(2) pmos_7(G2,in2,in1); // 0.5u 0.07u
nmos #(2) nmos_8(G2,w9,in1); // 0.3u 0.07u
pmos #(2) pmos_9(out3,in3,in2); // 0.5u 0.07u
nmos #(2) nmos_10(out3,w10,in2); // 0.3u 0.07u
pmos #(2) pmos_11(G3,in4,in3); // 0.5u 0.07u
nmos #(2) nmos_12(G3,w11,in3); // 0.3u 0.07u
endmodule

// Simulation parameters in Verilog Format
always
#200 in1=in1;
#400 in2=in2;
#800 in3=in3;
#1600 in4=in4;

// Simulation parameters
// in1 CLK 1
// in2 CLK 2
// in3 CLK 4
// in4 CLK 8

```

Information

- Module name (8 char max)  
convertorGrey
- Add gate delay info
- Append simul. informations
- Add labels as comments

The Verilog file has 34 lines  
The design includes 26 symbols  
The circuit has 11 nodes

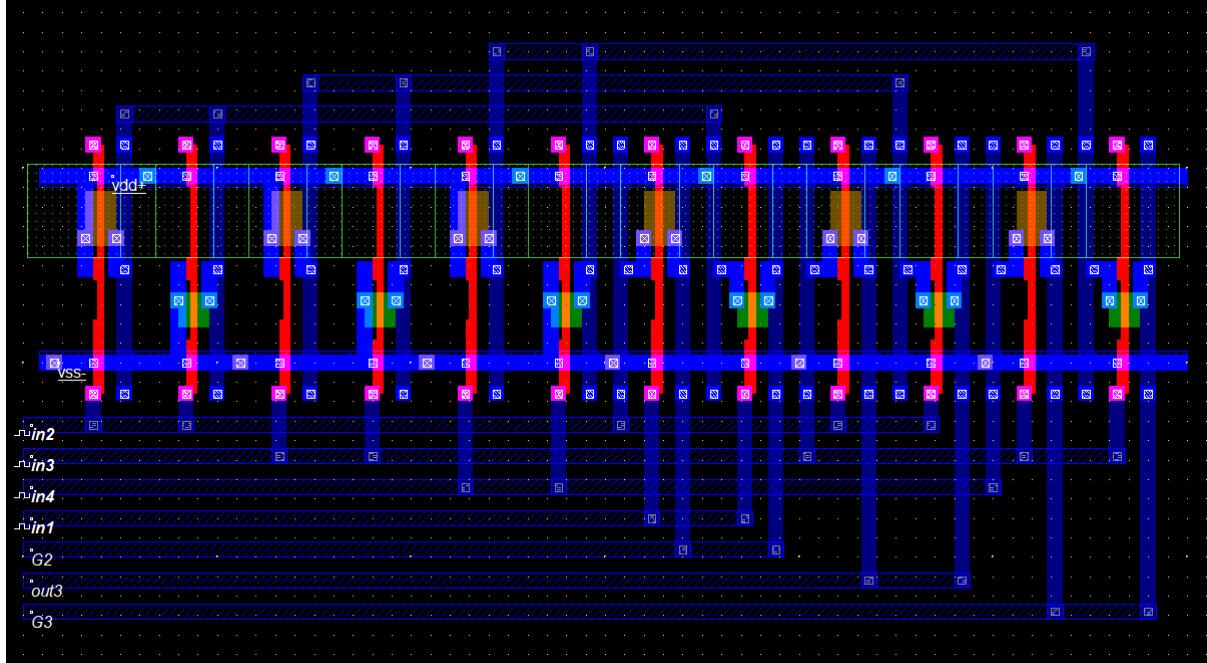
Misc:

Time scale: 1.00  
Max clocks: 16

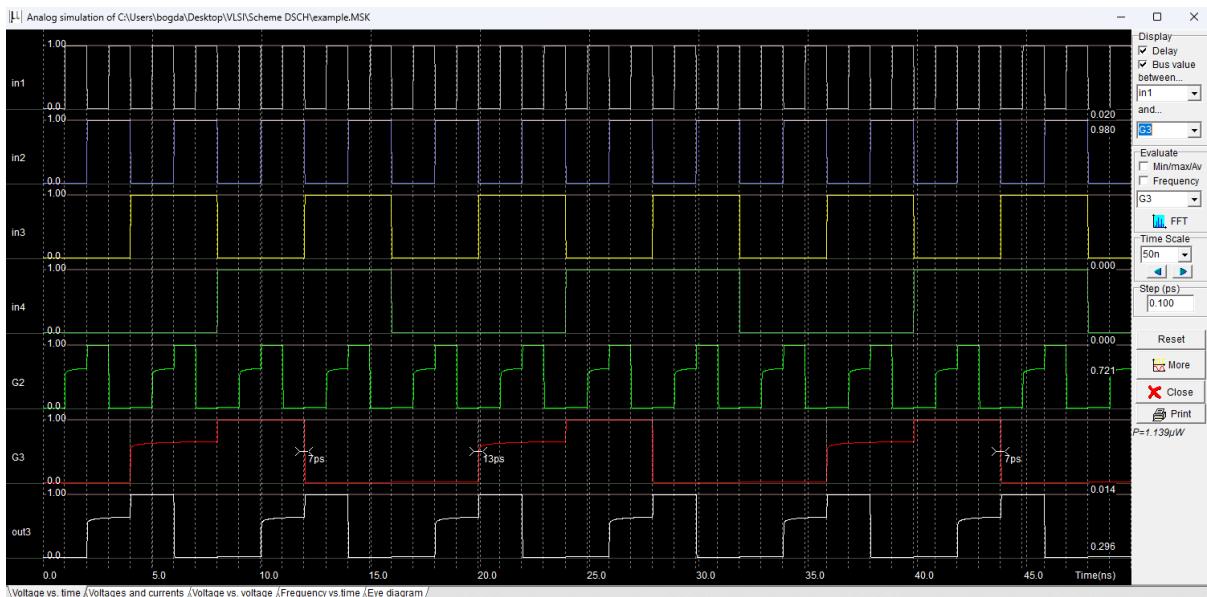
Update Verilog Extract circuit

OK

- microwind



- simulare microwind



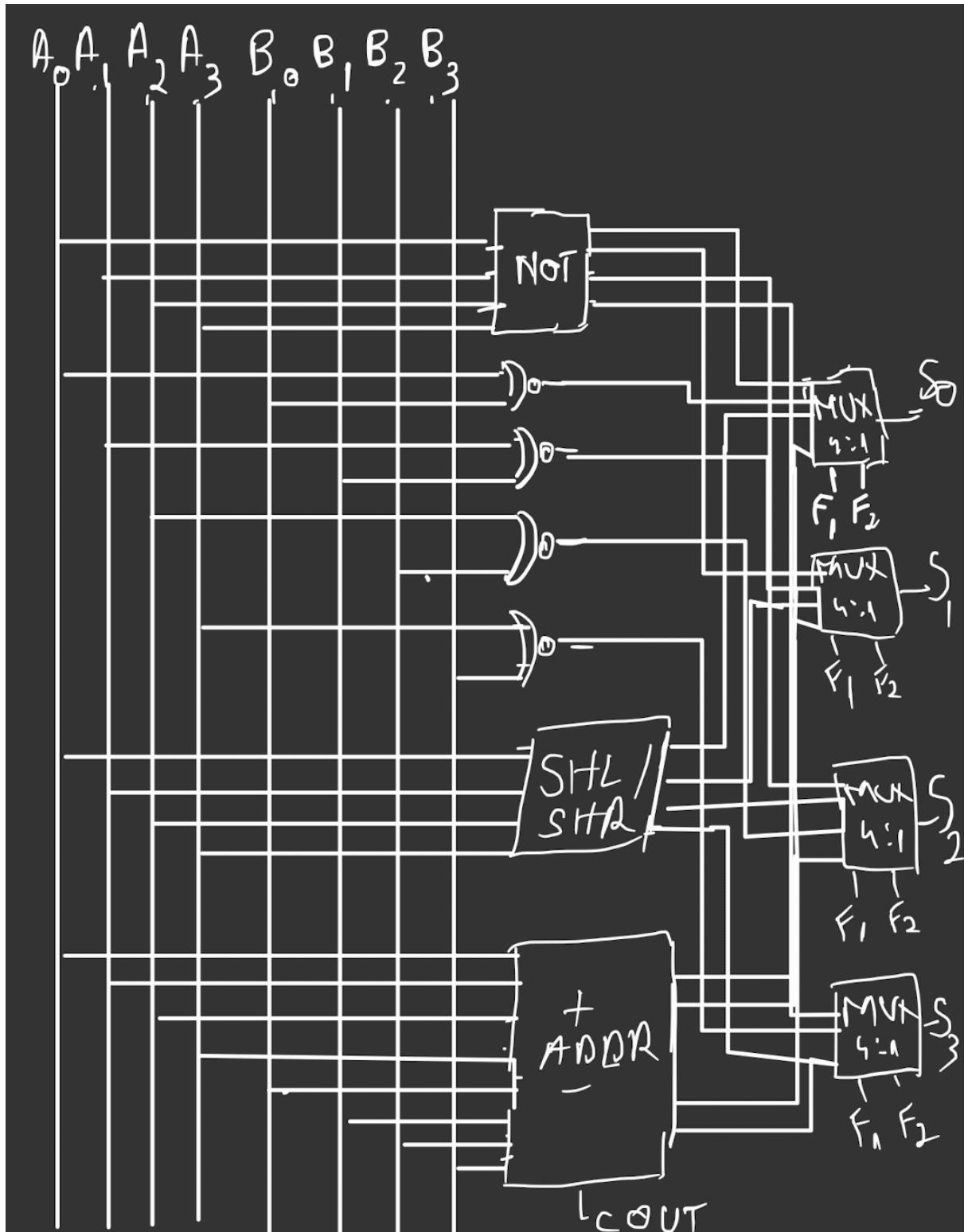
## 11. ALU pe 4 biti

- schema circuit
- A0A1A2A3 - primul numar
- B0B1B2B3 - al doilea numar

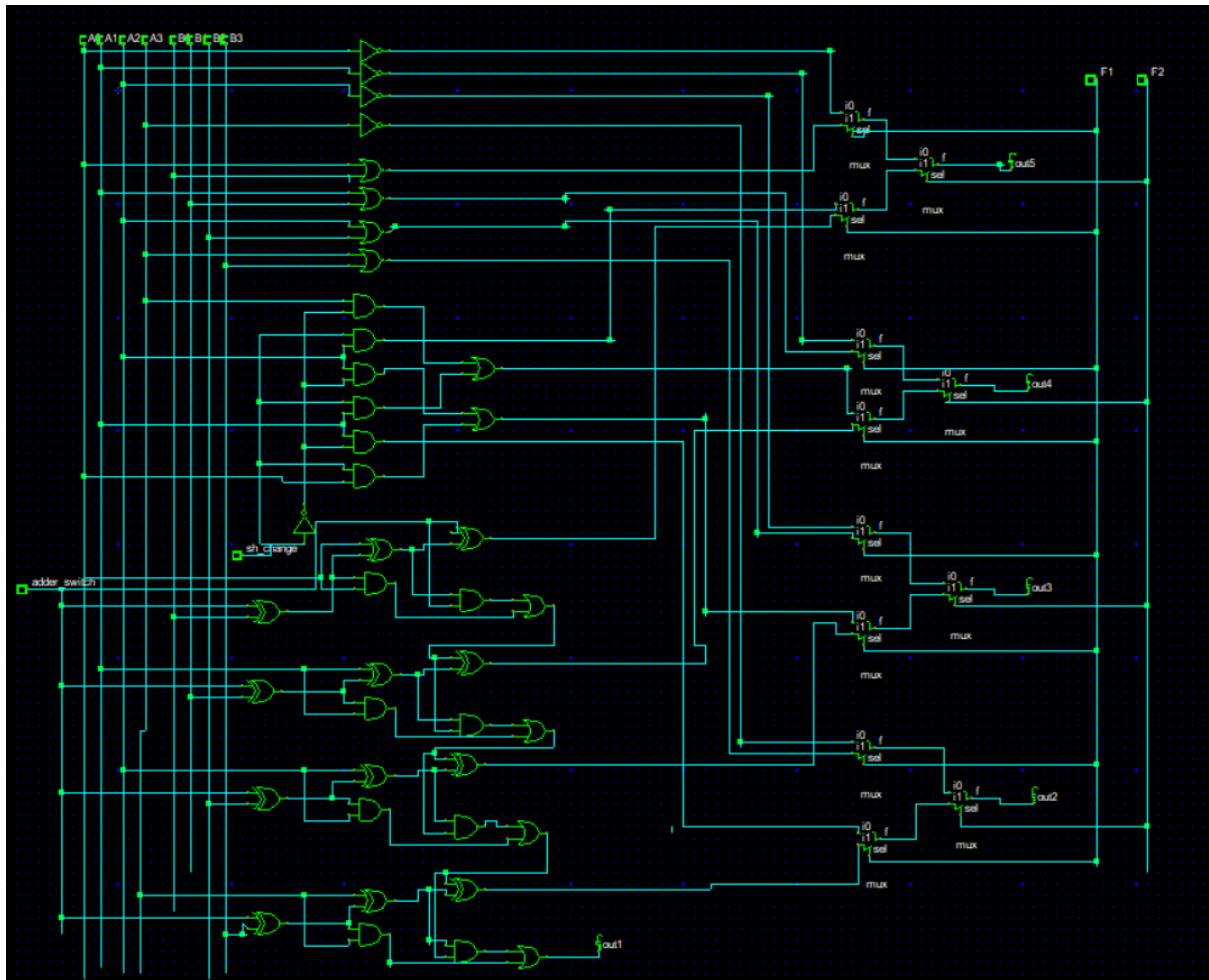
Folosesc Mux 4:1 pentru a activa ce raspuns vreau la iesire ( nor, not, addr sau shifter)

Pentru a selecta daca este adunare sau scadere voi folosi inca un input.

La fel si pentru shifter ( [https://www.researchgate.net/figure/Classical-circuit-for-the-4-bitshift-operator\\_fig2\\_2862072](https://www.researchgate.net/figure/Classical-circuit-for-the-4-bitshift-operator_fig2_2862072))

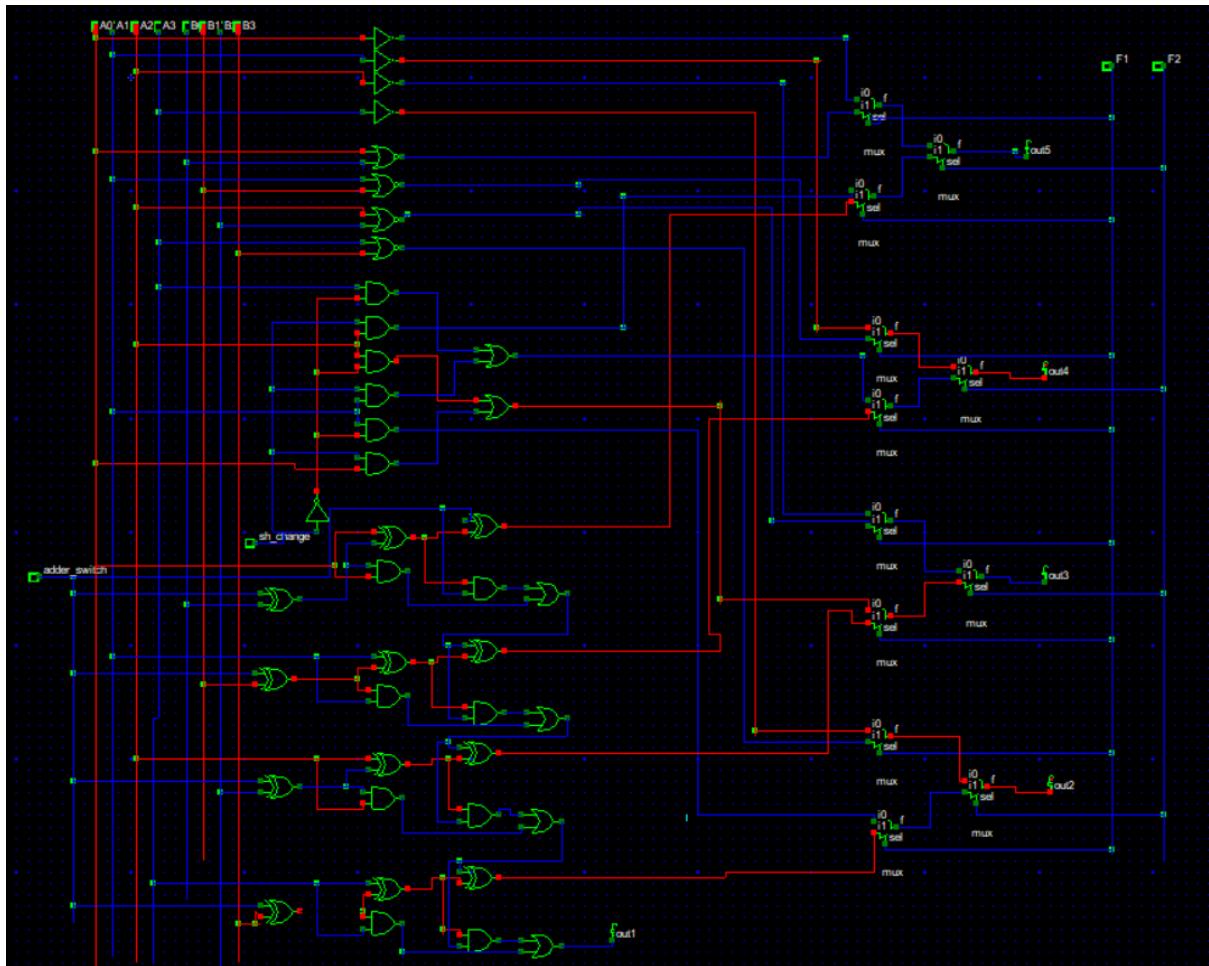


- schema dsch

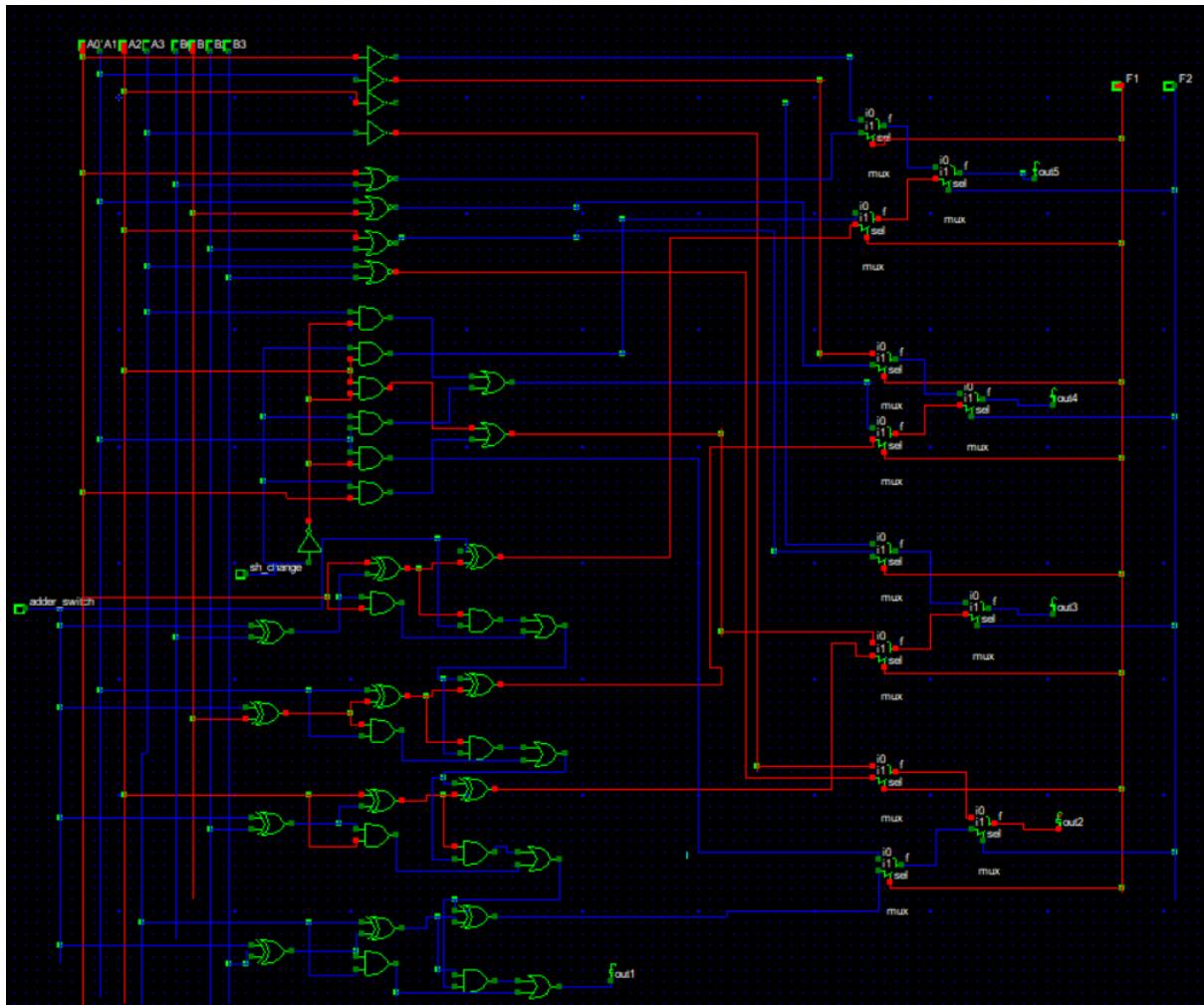


- simulare dsch

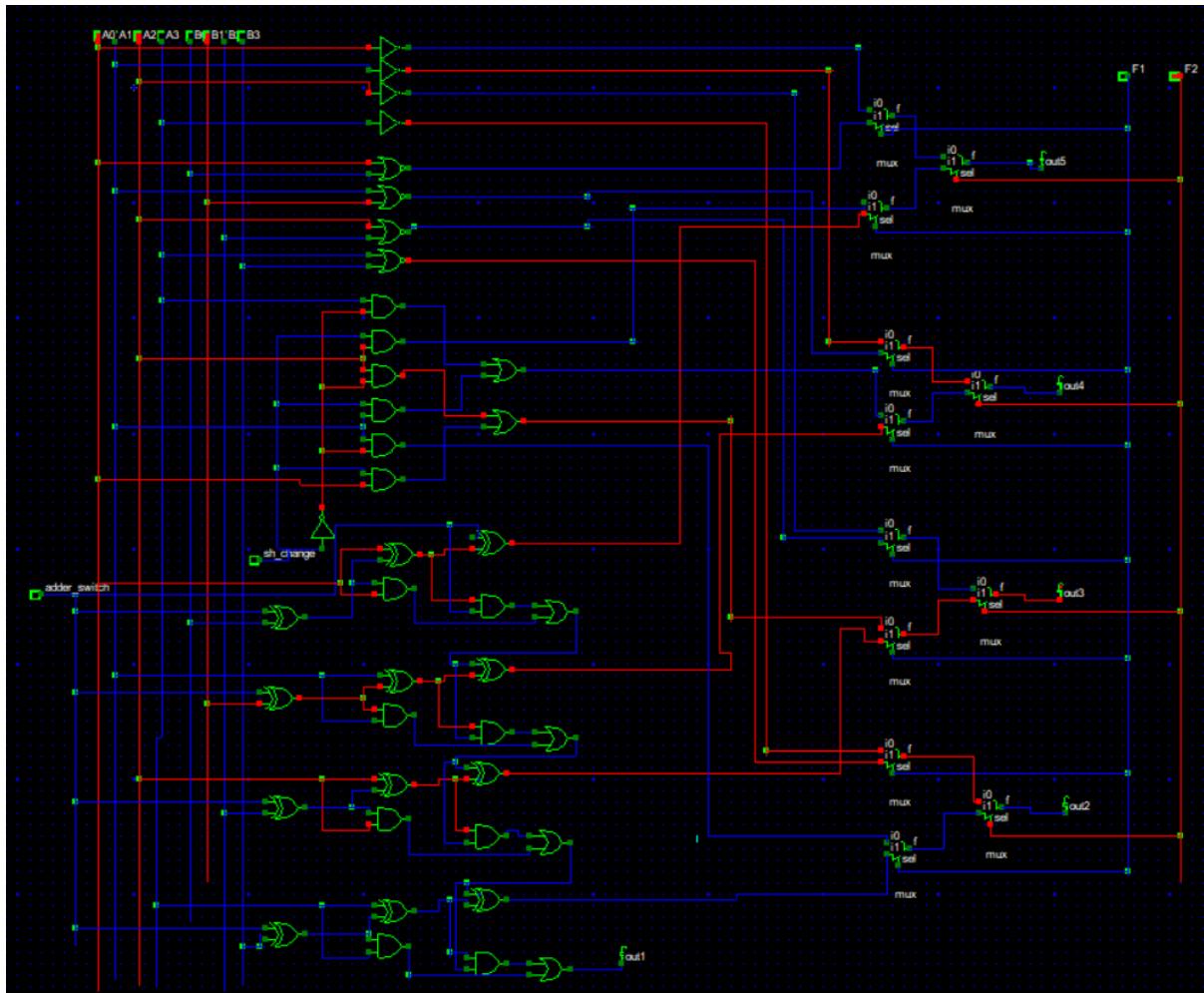
NOT

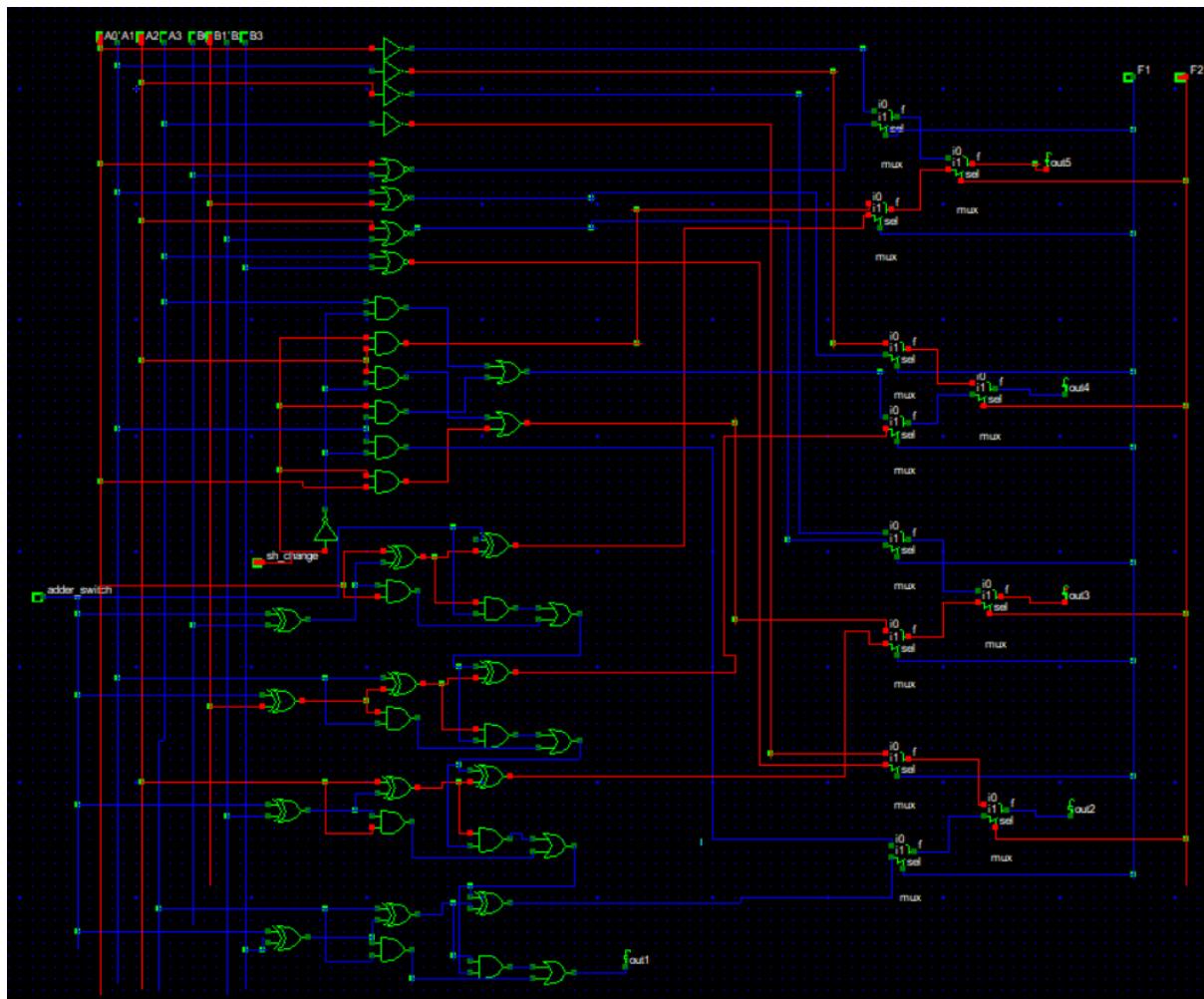


NOR

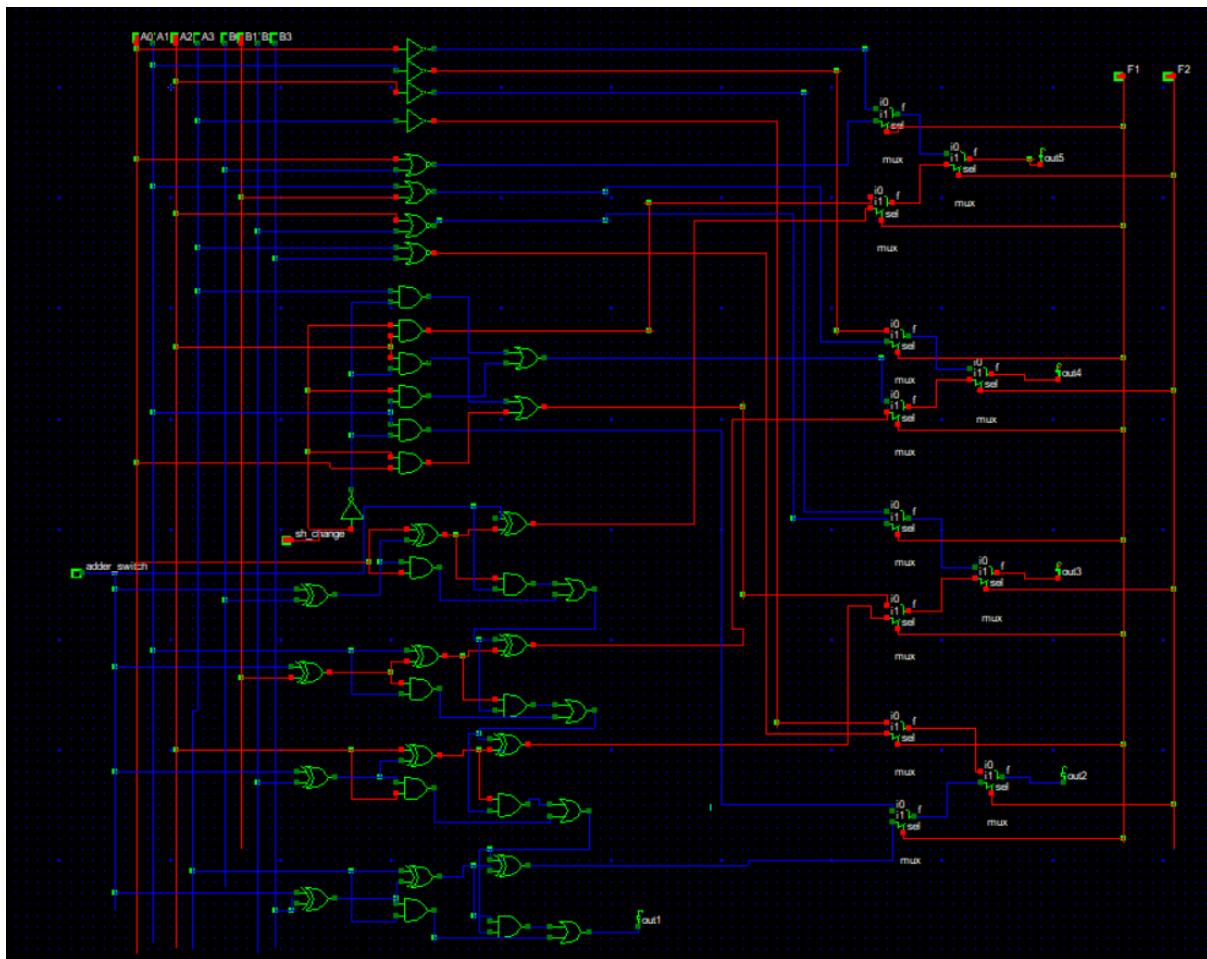


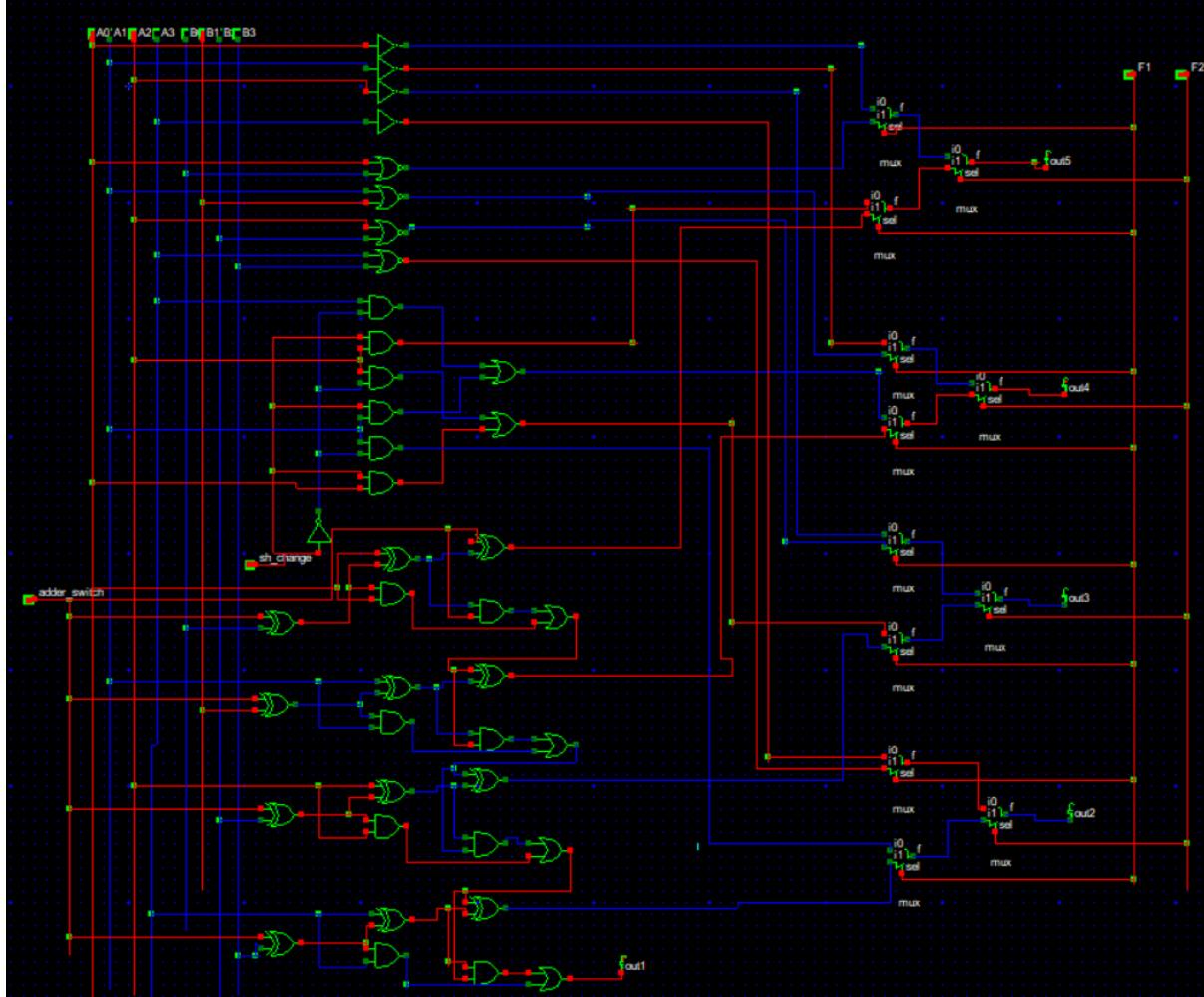
Shifter





ADDER





- cod verilog

```
// Verilog, Hierarchy and Netlist
[Verilog] [Hierarchy] [Netlist] [Critical path]
// DSCH 3.5
// 5/11/2023 10:29:55 PM
// C:\Users\boogda\Desktop\VLSI\Scheme DSCH\alu.sch

module alu(A0,A1,A2,A3,B0,B1,B2,B3,
sh_change,adder_switch,F1,F2,out1,out2,out3,out4,
out5);
input A0,A1,A2,A3,B0,B1,B2,B3;
input sh_change,adder_switch,F1,F2;
output out1,out2,out3,out4,out5,F1,F2;
wire w1,w2,w3,w4,w5,w6,w7,w8,w9,w10,w11,w12,w13,w14,w15;
wire w15,w16,w17,w18,w19,w20,w21,w22,w23,w24,w25,w26;
wire w27,w28,w29,w30,w31,w32,w33,w34,w35;
wire w36,w37,w38,w39,w40,w41,w42,w43,w44;
wire w45,w46,w47,w48,w49,w50,w51,w52;
wire w54,w55,w56,w57,w58,w59,w61,w62,w63,w64,w65;
wire ;
not # (1) inv_1(w10,A3);
not # (1) inv_2(w11,A2);
not # (1) inv_3(w12,A1);
not # (1) inv_4(w13,A0);
nor # (3) nor2_5(w14,B0);
nor # (3) nor2_6(w15,A1,B1);
nor # (3) nor2_7(w16,A2,B2);
nor # (3) nor2_8(w17,A3,B3);
and # (3) and2_9(w19,w18,A3);
and # (3) and2_10(w21,A2,sh_change);
and # (3) and2_11(w22,w18,A2);
and # (3) and2_12(w23,A1,sh_change);
and # (3) and2_13(w24,w18,A1);
or # (3) or2_14(w25,w19,w20);
or # (3) or2_15(w27,w19,w26);
not # (2) inv_16(w18,sh_change);
and # (3) and2_17(w26,A0,sh_change);
and # (3) and2_18(w30,w28,w29);
xor # (3) xor2_19(w32,w28,w29);
and # (3) and2_20(w34,A2,w33);
xor # (4) xor2_21(w29,A2,w33);
or # (4) or2_22(w35,w30,w31);
or # (4) or2_23(w36,w31,w32);
xor # (4) xor2_24(w37,w31,w32);
and # (3) and2_25(w37,A3,w39);
xor # (3) xor2_26(w41,w35,w40);
and # (3) and2_27(w36,w35,w40);
and # (3) and2_28(w44,w42,w43);
xor # (3) xor2_29(w45,w42,w43);
and # (3) and2_30(w47,A1,w46);
```

OK

Verilog, Hierarchy and Netlist | Verilog | Hierarchy | Netlist | Critical path | Information

```

or #(4) xor2_22(w35,w30,w34);
or #(3) xor2_23(w31,w35,w37);
xor #(4) xor2_24(w40,A3,w39);
and #(3) and2_25(w37,A3,w39);
xor #(3) xor2_26(w41,w35,w40);
and #(3) and2_27(w36,w35,w40);
and #(3) and2_28(w44,w42,w43);
xor #(3) xor2_29(w45,w42,w43);
and #(3) and2_30(w47,A1,w46);
xor #(4) xor2_31(w43,A1,w46);
or #(4) or2_32(w28,w32,F1);
or #(4) or2_33(w42,w46,w48);
xor #(4) xor2_34(w51,A0,w50);
and #(3) and2_35(w49,A0,w50);
xor #(3) xor2_36(w52,adder_switch,w51);
and #(3) and2_37(w48,adder_switch,w51);
xor #(4) xor2_38(w39,adder_switch,B3);
xor #(4) xor2_39(w33,adder_switch,B2);
xor #(4) xor2_40(w46,adder_switch,B1);
xor #(4) xor2_41(w50,adder_switch,B0);
mux #(1) mux_42(w54,w13,w14,F1);
mux #(1) mux_43(w55,w21,w52,F1);
mux #(1) mux_44(out5,w54,w55,F2);
mux #(1) mux_45(w56,w12,w15,F1);
mux #(1) mux_46(w59,w25,w45,F1);
mux #(1) mux_47(out4,w58,w59,F2);
mux #(1) mux_48(w61,w11,w16,F1);
mux #(1) mux_49(w62,w27,w32,F1);
mux #(1) mux_50(out3,w61,w62,F2);
mux #(1) mux_51(w64,w10,w17,F1);
mux #(1) mux_52(w65,w24,w41,F1);
mux #(1) mux_53(out2,w64,w65,F2);
endmodule

// Simulation parameters in Verilog Format
always
#200 A0=~A0;
#400 A1=~A1;
#800 A2=~A2;
#1600 A3=~A3;
#3200 B0=~B0;
#6400 B1=~B1;
#12800 B2=~B2;
#25600 B3=~B3;
#51200 sh_change=~sh_change;
#102400 adder_switch=~adder_switch;
#102400 F1=~F1;
#102400 F2=~F2;

```

The Verilog file has 100 lines  
The design includes 70 symbols  
The circuit has 66 nodes

Misc.

Time scale : 1.00  
Max clocks: 16

Update Verilog Extract circuit

OK

Verilog, Hierarchy and Netlist | Verilog | Hierarchy | Netlist | Critical path | Information

```

xor #(4) xor2_39(w33,adder_switch,B2);
xor #(4) xor2_40(w46,adder_switch,B1);
xor #(4) xor2_41(w50,adder_switch,B0);
mux #(1) mux_42(w54,w13,w14,F1);
mux #(1) mux_43(w55,w21,w52,F1);
mux #(1) mux_44(out5,w54,w55,F2);
mux #(1) mux_45(w56,w12,w15,F1);
mux #(1) mux_46(w59,w25,w45,F1);
mux #(1) mux_47(w61,w11,w16,F1);
mux #(1) mux_48(w62,w27,w32,F1);
mux #(1) mux_50(out3,w61,w62,F2);
mux #(1) mux_51(w64,w10,w17,F1);
mux #(1) mux_52(w65,w24,w41,F1);
mux #(1) mux_53(out2,w64,w65,F2);
endmodule

// Simulation parameters in Verilog Format
always
#200 A0=~A0;
#400 A1=~A1;
#800 A2=~A2;
#1600 A3=~A3;
#3200 B0=~B0;
#6400 B1=~B1;
#12800 B2=~B2;
#25600 B3=~B3;
#51200 sh_change=~sh_change;
#102400 adder_switch=~adder_switch;
#102400 F1=~F1;
#102400 F2=~F2;

// Simulation parameters
// A0 CLK 1 1
// A1 CLK 2 2
// A2 CLK 4 4
// A3 CLK 8 8
// B0 CLK 16 16
// B1 CLK 32 32
// B2 CLK 64 64
// B3 CLK 128 128
// sh_change CLK 256 256
// adder_switch CLK 512 512
// F1 CLK 1024 1024
// F2 CLK 2048 2048

```

The Verilog file has 100 lines  
The design includes 70 symbols  
The circuit has 66 nodes

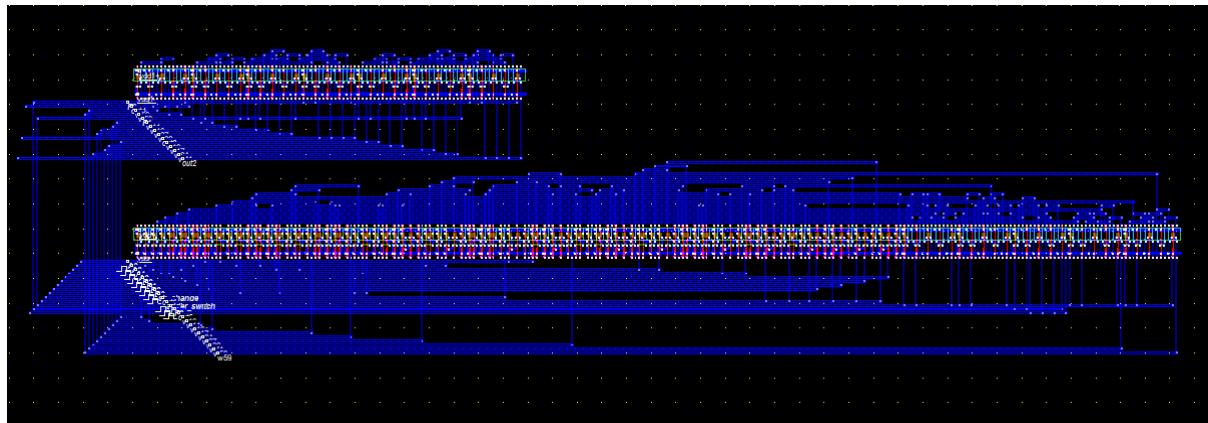
Misc.

Time scale : 1.00  
Max clocks: 16

Update Verilog Extract circuit

OK

- microwind



- simulare microwind

