3º BIMESTRE

FLIP FLOP

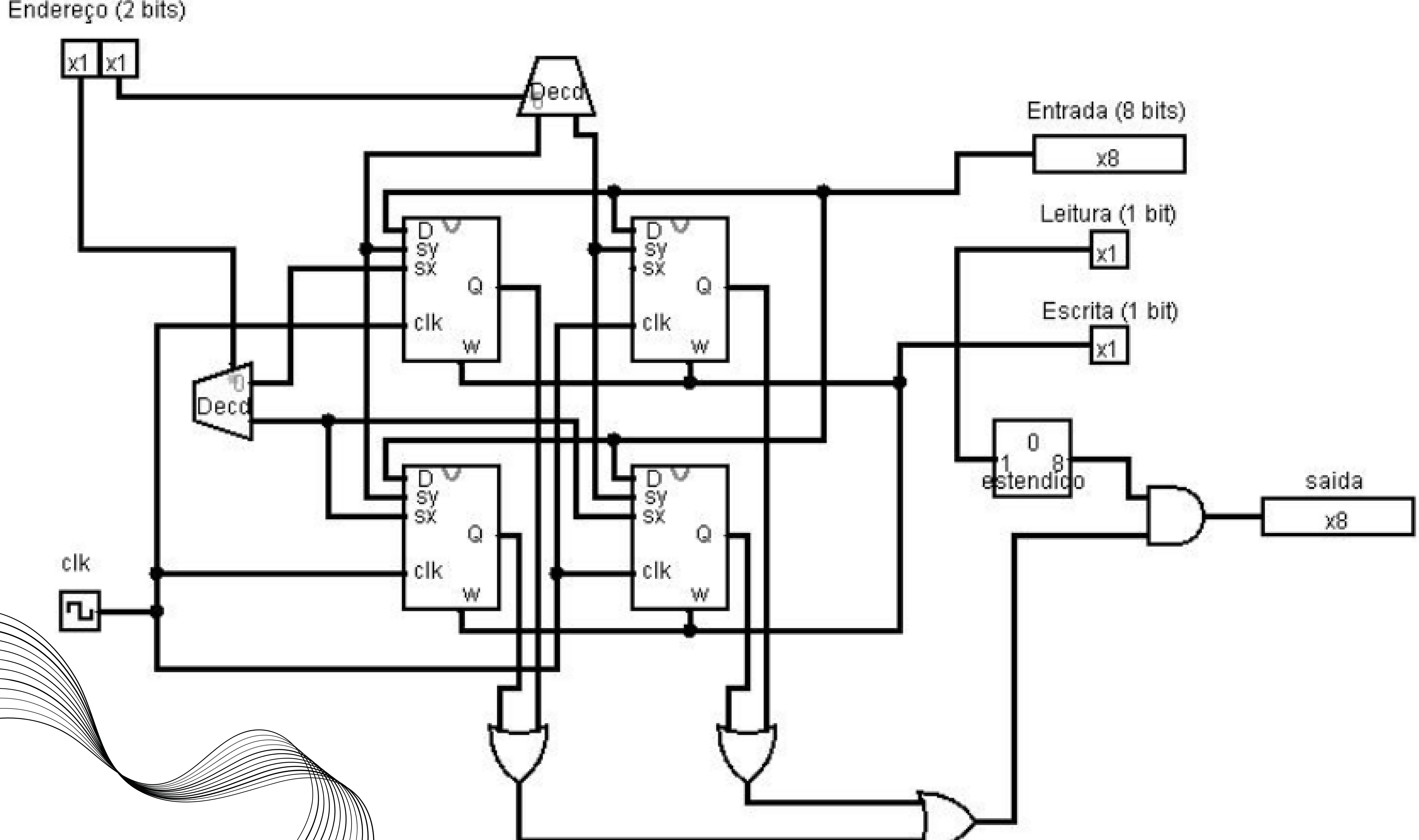
Eduardo Evangelista
Pedro Edom

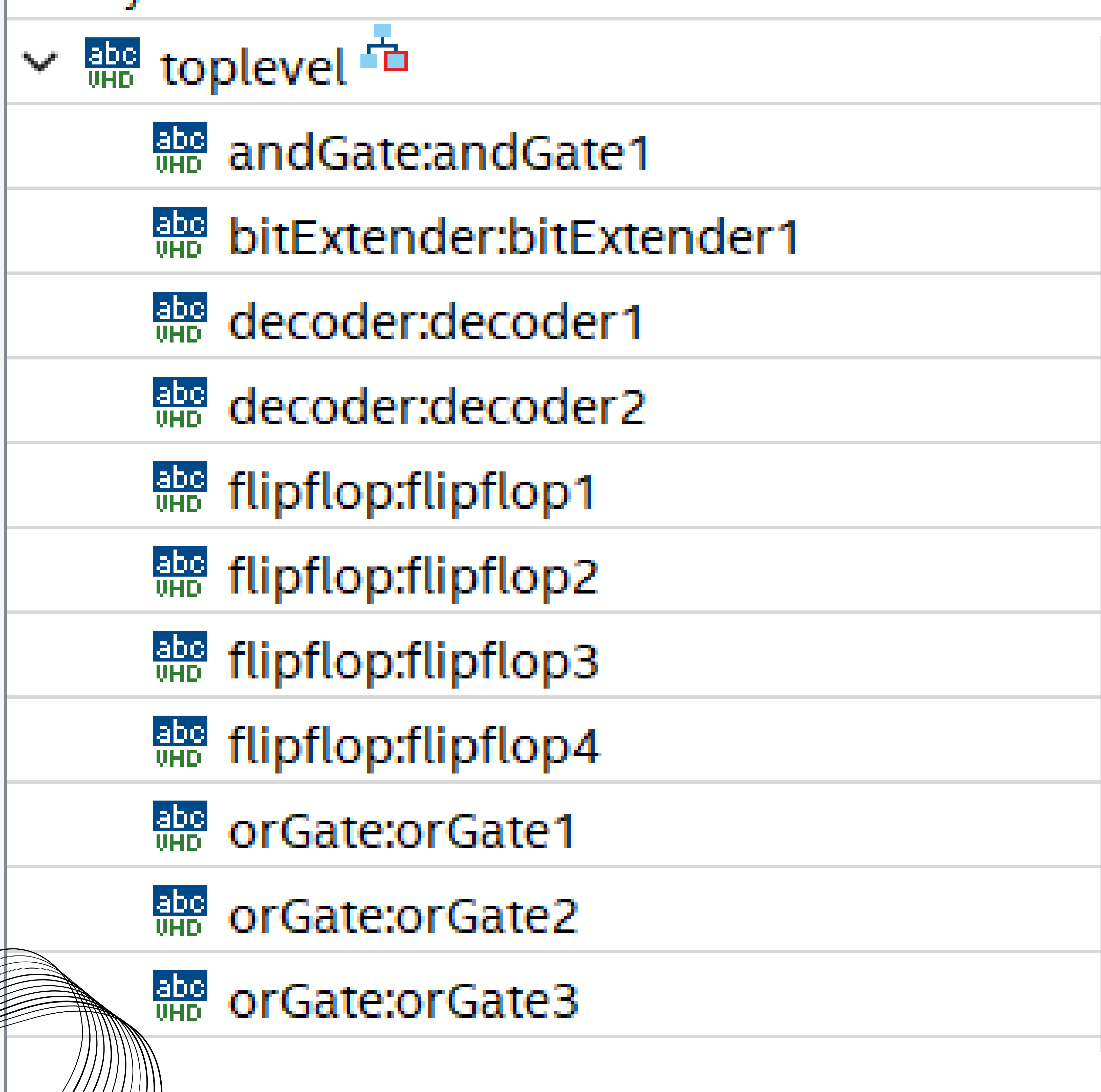
CONTENT

01 ESTRUTURA DO TRABALHO

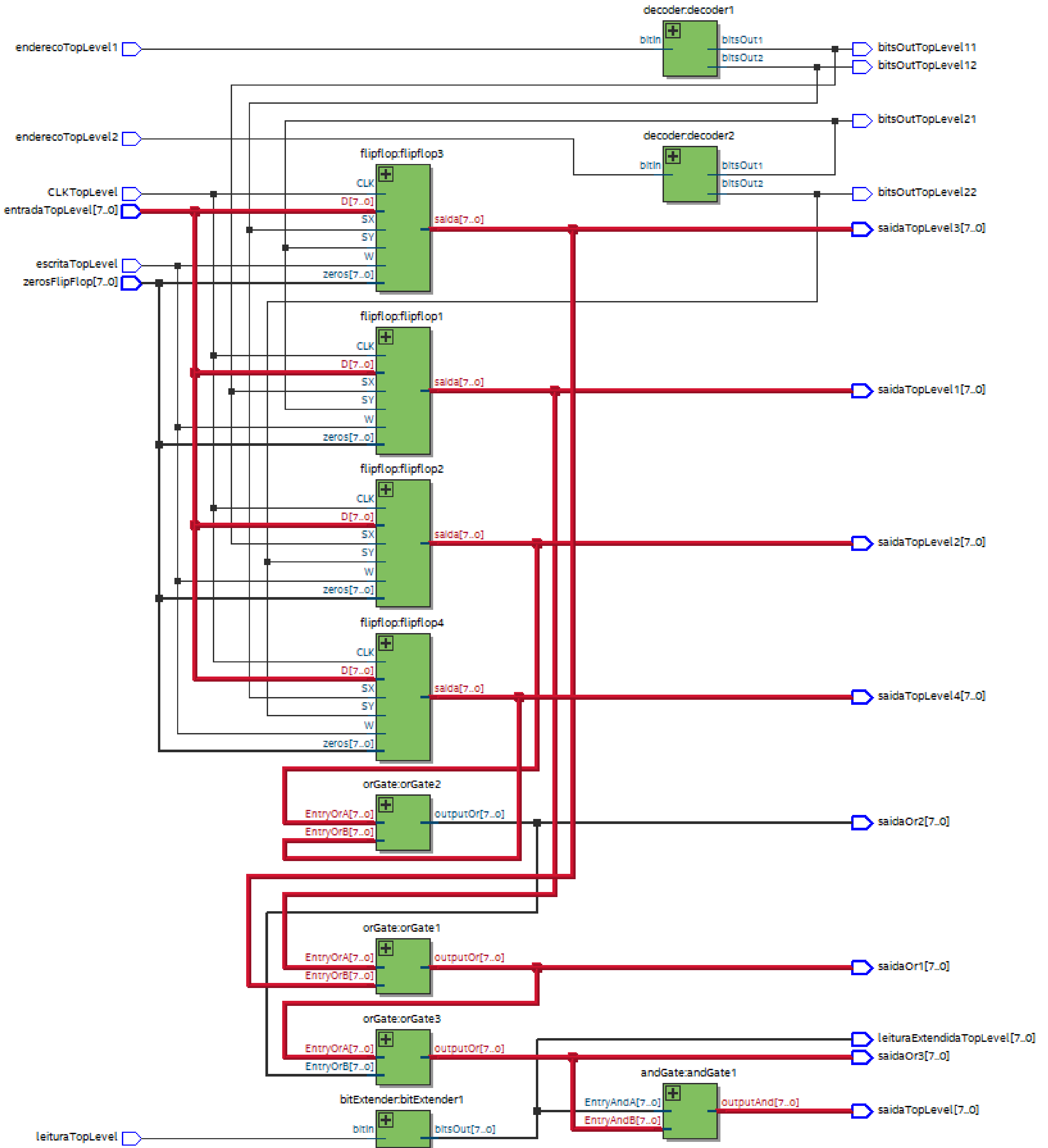
02 RTL VIEW

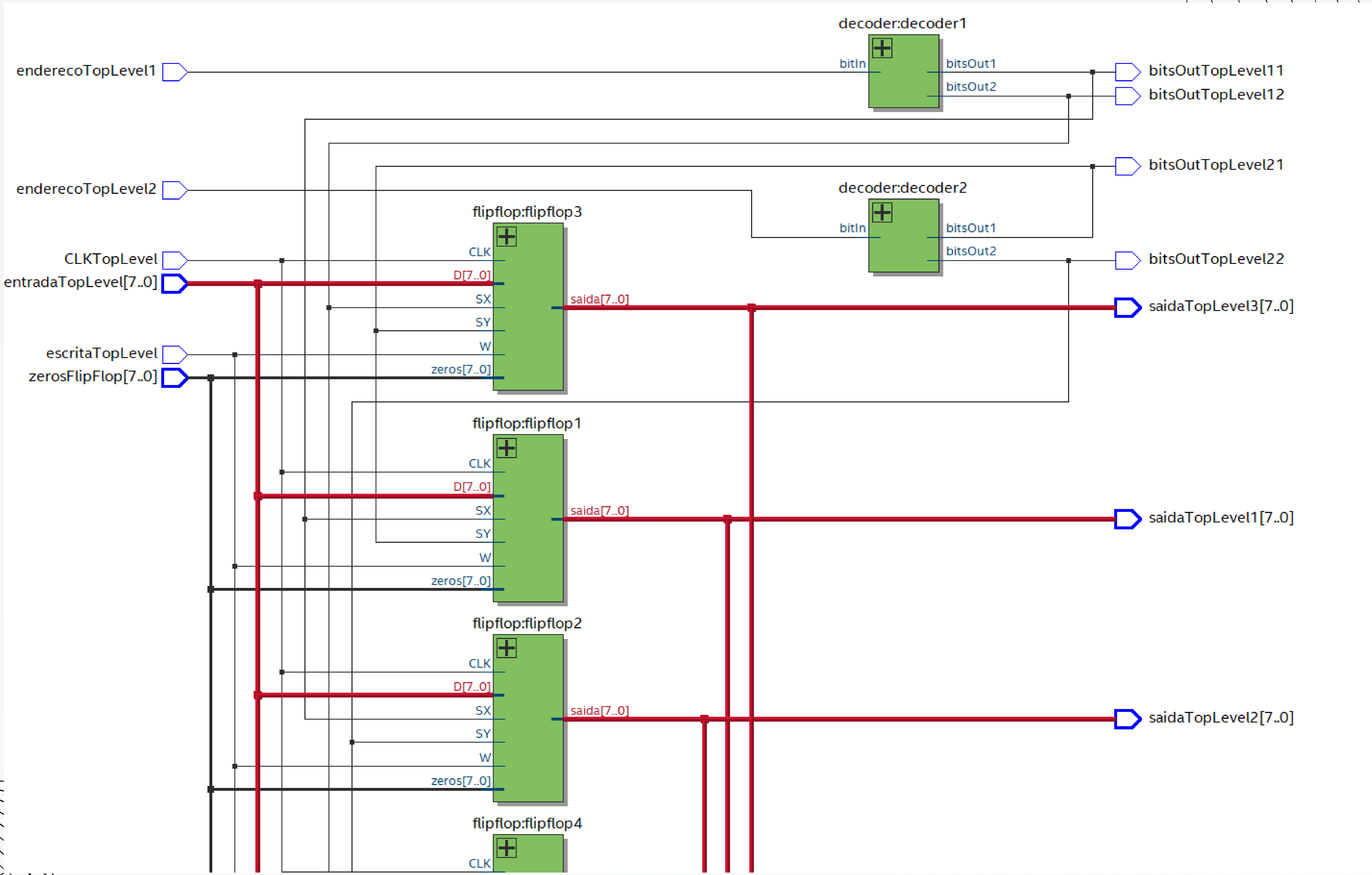
03 CÓDIGO FLIPFLOP

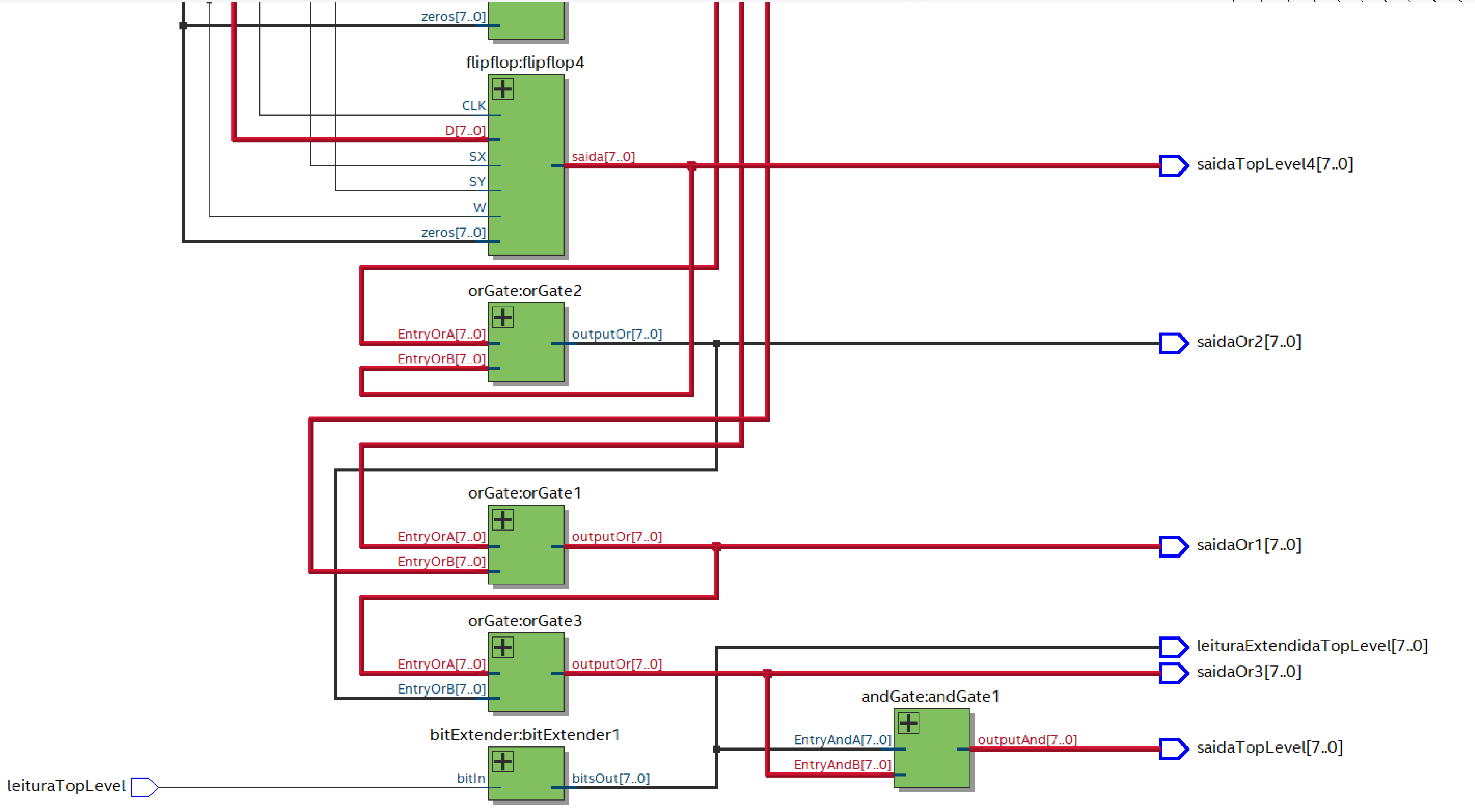

Endereço (2 bits)
x1
x1
Deco
0
Entrada (8 bits)
x8
Leitura (1 bit)
x1
Escrita (1 bit)
x1
D
sy
sx
Q
clk
w
D
sy
sx
Q
clk
w
Deco
0
D
sy
sx
Q
clk
w
D
sy
sx
Q
clk
w
0
1
8
estendido
saida
x8
clk

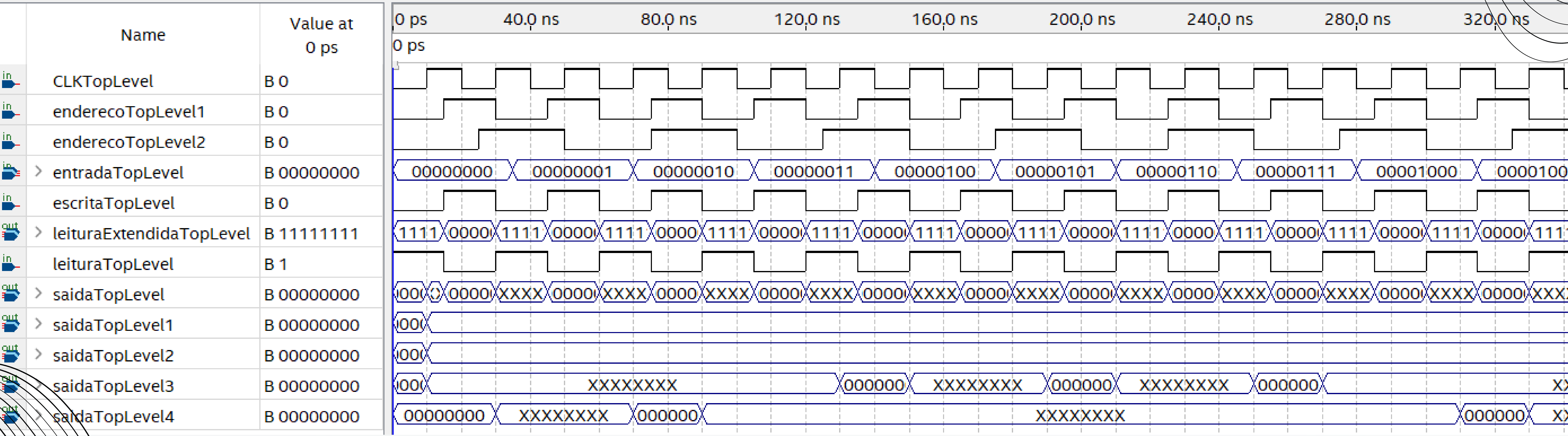


- toplevel
  - andGate:andGate1
  - bitExtender:bitExtender1
  - decoder:decoder1
  - decoder:decoder2
  - flipflop:flipflop1
  - flipflop:flipflop2
  - flipflop:flipflop3
  - flipflop:flipflop4
  - orGate:orGate1
  - orGate:orGate2
  - orGate:orGate3

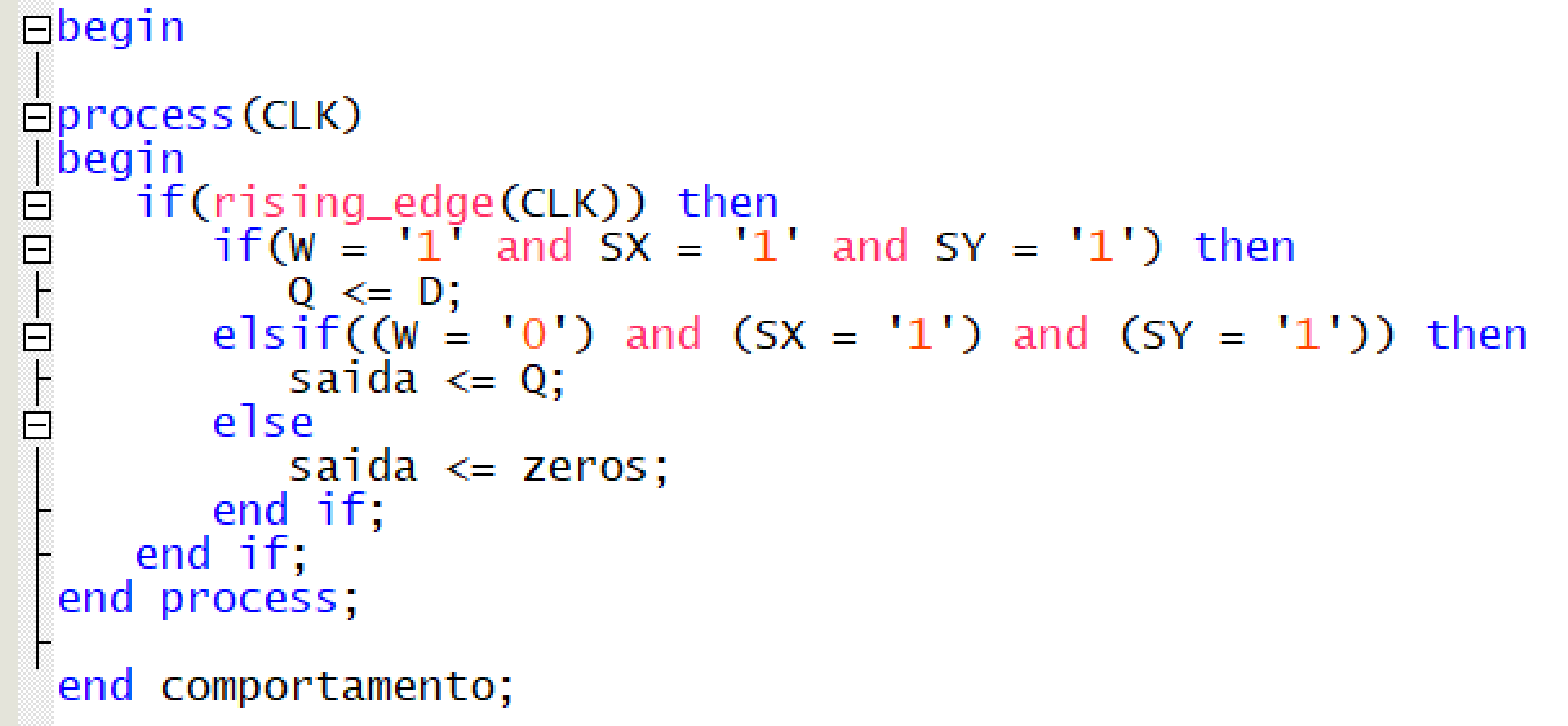

enderecoTopLevel1
decoder:decoder1
bitIn
bitsOut1
bitsOut2
bitsOutTopLevel11
bitsOutTopLevel12
bitsOutTopLevel21
enderecoTopLevel2
decoder:decoder2
bitIn
bitsOut1
bitsOut2
bitsOutTopLevel22
flipflop:flipflop3
CLKTopLevel
entradaTopLevel[7..0]
CLK
D[7..0]
SX
SY
W
zeros[7..0]
saida[7..0]
saidaTopLevel3[7..0]
escritaTopLevel
zerosFlipFlop[7..0]
flipflop:flipflop1
CLK
D[7..0]
SX
SY
W
zeros[7..0]
saida[7..0]
saidaTopLevel1[7..0]
flipflop:flipflop2
CLK
D[7..0]
SX
SY
W
zeros[7..0]
saida[7..0]
saidaTopLevel2[7..0]
flipflop:flipflop4
CLK
D[7..0]
SX
SY
W
zeros[7..0]
saida[7..0]
saidaTopLevel4[7..0]
orGate:orGate2
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
saidaOr2[7..0]
orGate:orGate1
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
saidaOr1[7..0]
orGate:orGate3
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
leituraExtendidaTopLevel[7..0]
saidaOr3[7..0]
andGate:andGate1
EntryAndA[7..0]
EntryAndB[7..0]
outputAnd[7..0]
saidaTopLevel[7..0]
bitExtender:bitExtender1
leituraTopLevel
bitIn
bitsOut[7..0]


decoder:decoder1
enderecoTopLevel1
bitIn
bitsOut1
bitsOut2
bitsOutTopLevel11
bitsOutTopLevel12
bitsOutTopLevel21
enderecoTopLevel2
decoder:decoder2
bitIn
bitsOut1
bitsOut2
flipflop:flipflop3
CLKTopLevel
CLK
bitsOutTopLevel22
entradaTopLevel[7..0]
D[7..0]
SX
saida[7..0]
saidaTopLevel3[7..0]
SY
escritaTopLevel
W
zerosFlipFlop[7..0]
zeros[7..0]
flipflop:flipflop1
CLK
D[7..0]
SX
saida[7..0]
saidaTopLevel1[7..0]
SY
W
zeros[7..0]
flipflop:flipflop2
CLK
D[7..0]
SX
saida[7..0]
saidaTopLevel2[7..0]
SY
W
zeros[7..0]
flipflop:flipflop4
CLK

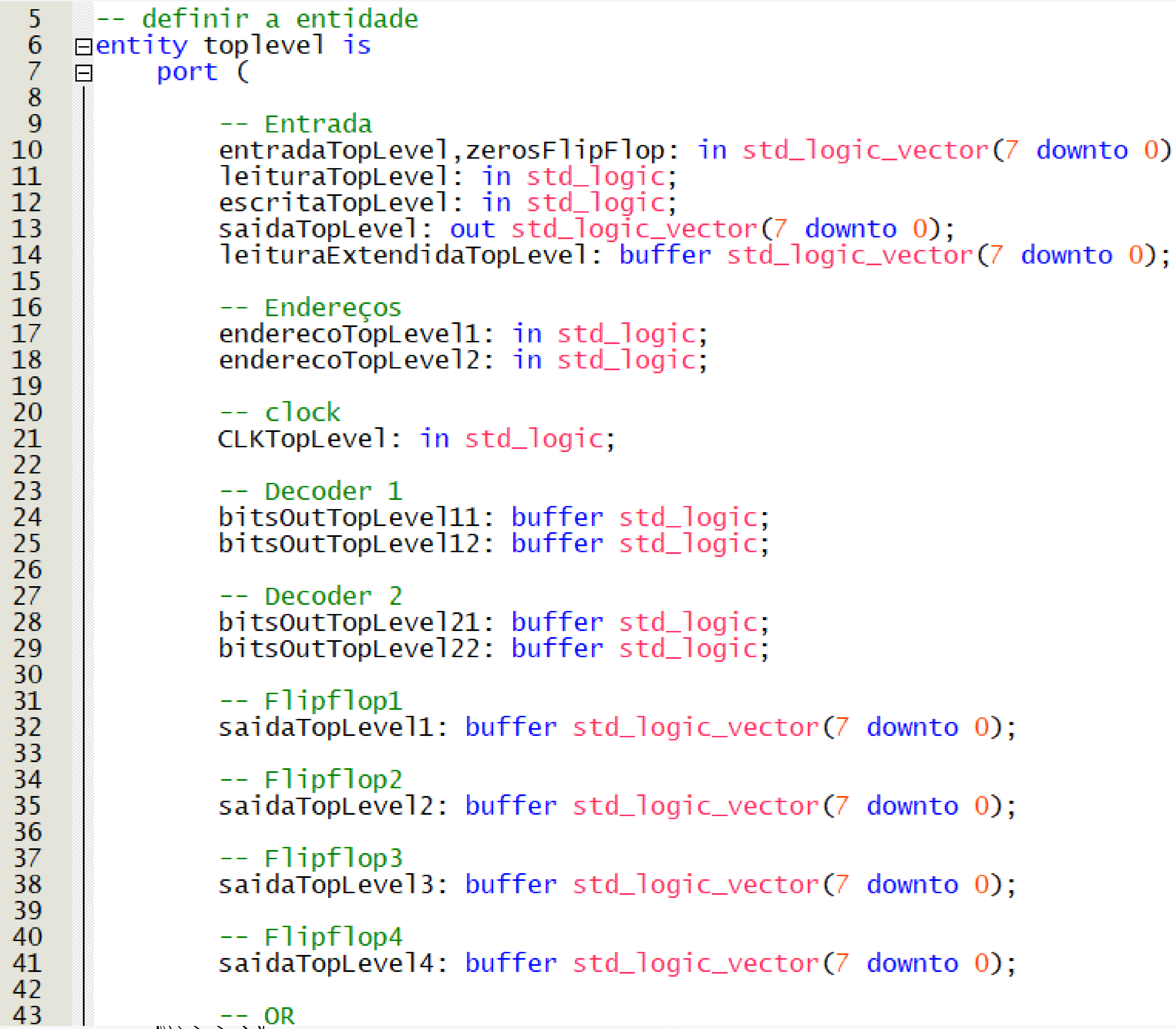

zeros[7..0]
flipflop:flipflop4
CLK
D[7..0]
SX
SY
W
zeros[7..0]
saida[7..0]
saidaTopLevel4[7..0]
orGate:orGate2
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
saidaOr2[7..0]
orGate:orGate1
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
saidaOr1[7..0]
orGate:orGate3
EntryOrA[7..0]
EntryOrB[7..0]
outputOr[7..0]
leituraExtendidaTopLevel[7..0]
saidaOr3[7..0]
andGate:andGate1
EntryAndA[7..0]
EntryAndB[7..0]
outputAnd[7..0]
saidaTopLevel[7..0]
bitExtender:bitExtender1
bitIn
bitsOut[7..0]
leituraTopLevel



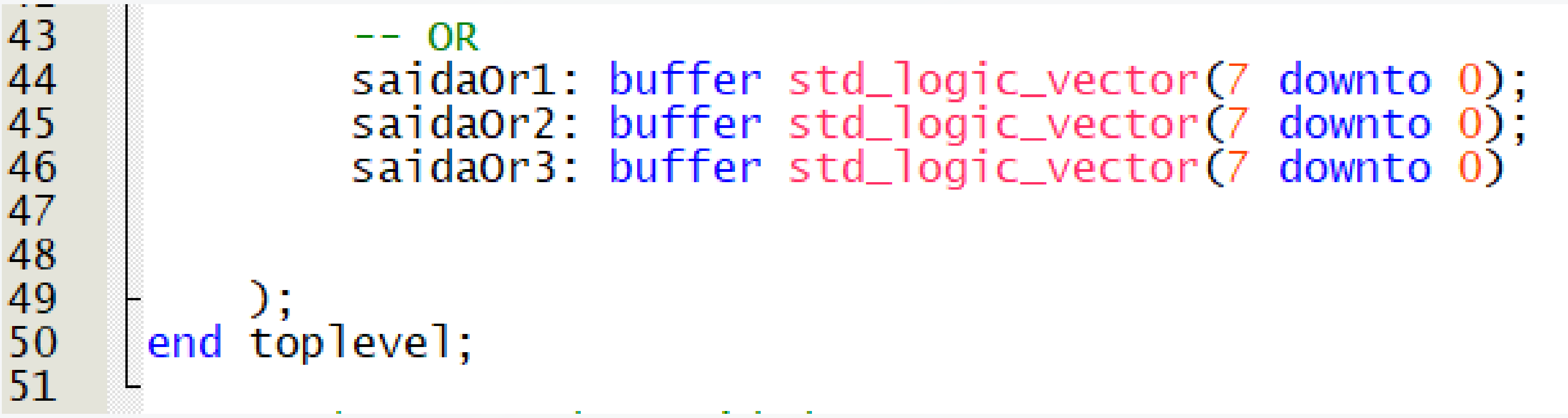
Name	Value at 0 ps
CLKTopLevel	B 0
enderecoTopLevel1	B 0
enderecoTopLevel2	B 0
entradaTopLevel	B 00000000
escritaTopLevel	B 0
leituraExtendidaTopLevel	B 11111111
leituraTopLevel	B 1
saidaTopLevel	B 00000000
saidaTopLevel1	B 00000000
saidaTopLevel2	B 00000000
saidaTopLevel3	B 00000000
saidaTopLevel4	B 00000000



```vhdl
begin

process(CLK)
begin
   if(rising_edge(CLK)) then
      if(W = '1' and SX = '1' and SY = '1') then
         Q <= D;
      elsif((W = '0') and (SX = '1') and (SY = '1')) then
         saida <= Q;
      else
         saida <= zeros;
      end if;
   end if;
end process;

end comportamento;
```



```
-- definir a entidade
entity toplevel is
    port (

        -- Entrada
        entradaTopLevel,zerosFlipFlop: in std_logic_vector(7 downto 0)
        leituraTopLevel: in std_logic;
        escritaTopLevel: in std_logic;
        saidaTopLevel: out std_logic_vector(7 downto 0);
        leituraExtendidaTopLevel: buffer std_logic_vector(7 downto 0);

        -- Endereços
        enderecoTopLevel1: in std_logic;
        enderecoTopLevel2: in std_logic;

        -- clock
        CLKTopLevel: in std_logic;

        -- Decoder 1
        bitsOutTopLevel11: buffer std_logic;
        bitsOutTopLevel12: buffer std_logic;

        -- Decoder 2
        bitsOutTopLevel21: buffer std_logic;
        bitsOutTopLevel22: buffer std_logic;

        -- Flipflop1
        saidaTopLevel1: buffer std_logic_vector(7 downto 0);

        -- Flipflop2
        saidaTopLevel2: buffer std_logic_vector(7 downto 0);

        -- Flipflop3
        saidaTopLevel3: buffer std_logic_vector(7 downto 0);

        -- Flipflop4
        saidaTopLevel4: buffer std_logic_vector(7 downto 0);

        -- OR
```



```
        -- OR
        saidaOr1: buffer std_logic_vector(7 downto 0);
        saidaOr2: buffer std_logic_vector(7 downto 0);
        saidaOr3: buffer std_logic_vector(7 downto 0)


    );
end toplevel;

```

OBRIGADO!