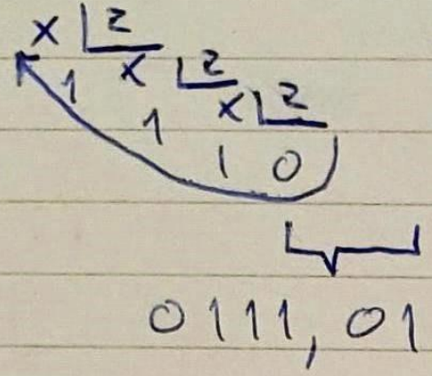
Adriano Augusto Antongiovanni

Transformação:

- ( )10 → ( )2 :



- ( )10 → ( )2 → ( )8

[$\overline{00}$| $\overline{110}$]2 → (xr)8

- ( )10 → ( )2 → ( )16

[$\overline{0000}$| $\overline{1000}$]2 → (xr)16

- **Operações:**
  - → Axiomas: $(a+b).\bar{a} = \bar{a}.b$ $(a.\bar{a}=0)$
  - → Propriedade Absorção: $(y+x).x = x$
    $yx+x = x$
  - → De Morgan: $\overline{x+y} = \bar{x}.\bar{y}$
    $\overline{xy} = \bar{x}+\bar{y}$

Mapa Karnaugh:

Wx \ yz	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

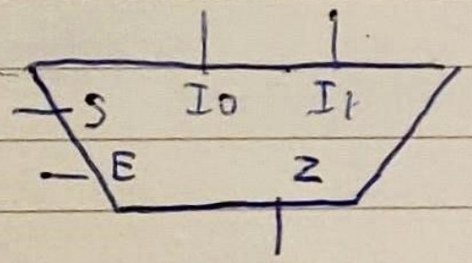
Ficha Resumo

• Complemento de 2 (C2):

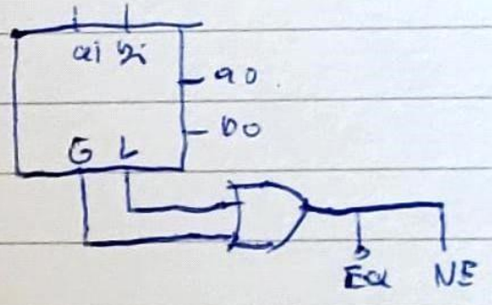
1000100 : 1- Copia até primeiro 1, inclusive
2- inverte demais

Nas operações: • $X - Y = X + (-Y)$
• estender ambos números e quantidade de bits

• Mux:

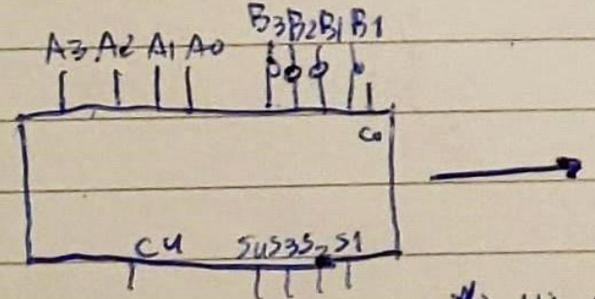


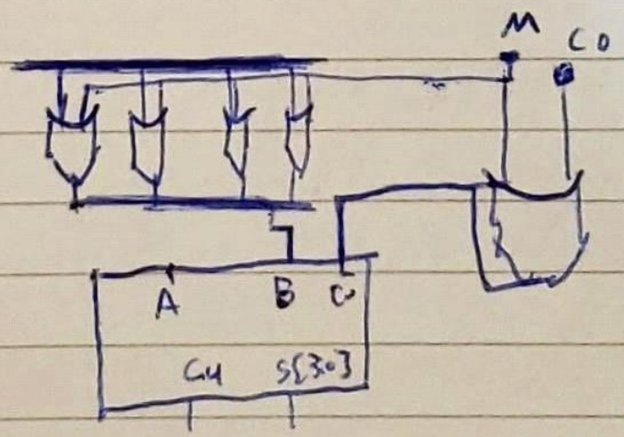
• Comparador Magnitude:



G = 1 → X > Y
G = 0 → X ≤ Y
L = 1 → X < Y
L = 0 → X ≥ Y

• Somador - subtrator:



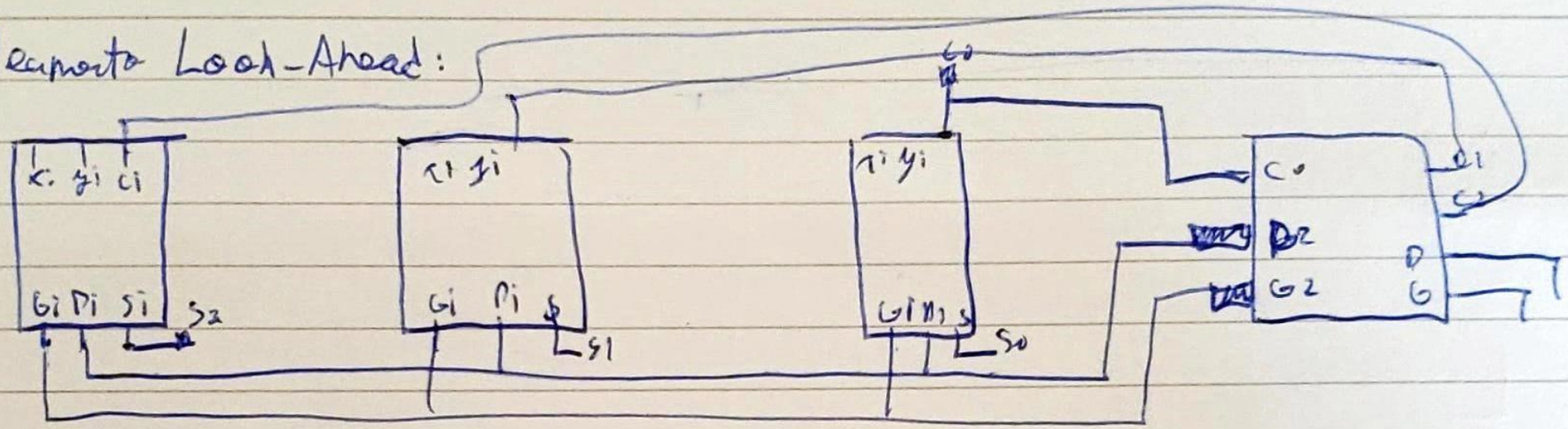


Ai Yi Co

A + B → Ai Bi 0

A - B → Ai $\bar{B}$i 1

Cascatamento Look-Ahead:



ULA:

S1 S0

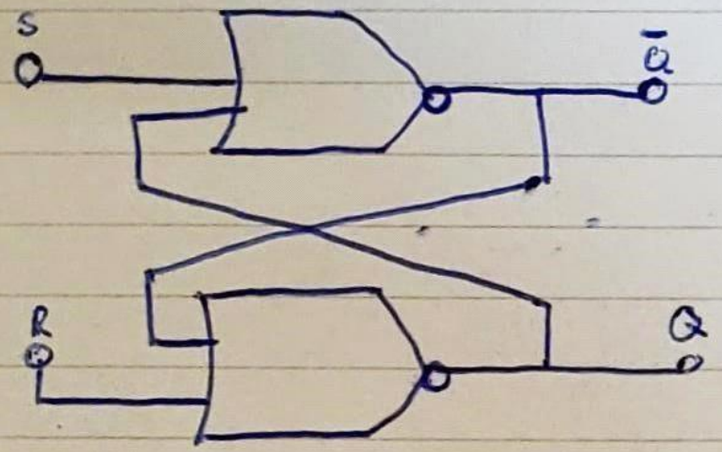
M=0 → Aritmética

S1 S0	Fi	Ai = xi	Bi = yi	XOR(M,C) = z_0	carry-out = C_3
0 0	A+B+c_0	ai	bi	c_0	z_3
0 1	A+1	ai	0	1	z_3
1 0	(A-1)c_2	ai	1	0	$\overline{z_3}$
1 1	(A-B-c	ai	$\overline{bi}$	$\overline{c_0}$	$\overline{z_3}$

M=1 → Lógica

S1 S0	Fi	x_1 y_i z_0 C_5
0 0	ai	ai 0 0 ,
0 1	ai and bi	ai and bi 0 0 0
1 0	ai or bi	ai or bi 0 0 0
1 1	not (ai	not(ai) 0 0 0

Latch SR (pozitiv)



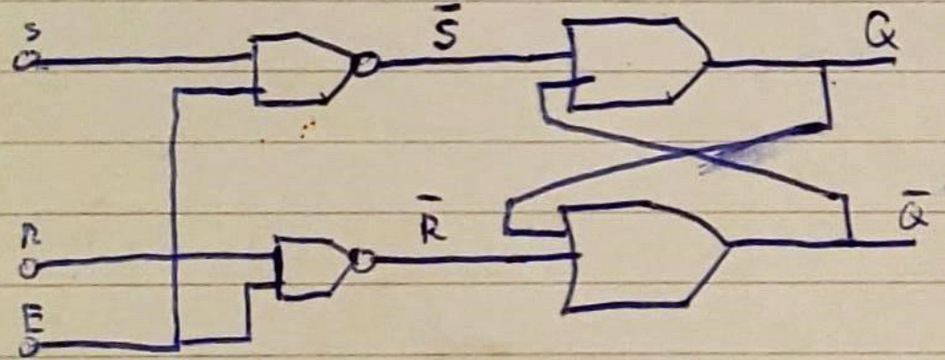
SR	Q^{k+1}	$\bar{Q}^{k+1}$	H
00	Q^k	$\bar{Q}^k$	memorizare
10	1	0	set Q
01	0	1	reset Q
11	0	0	/

En/clk	C	D	Q^{k+1} $\bar{Q}^{k+1}$
0	0	X	Q^k $\bar{Q}^k$
0	↑	0	0 1
0	↑	1	1 0
1	X	X	0 1

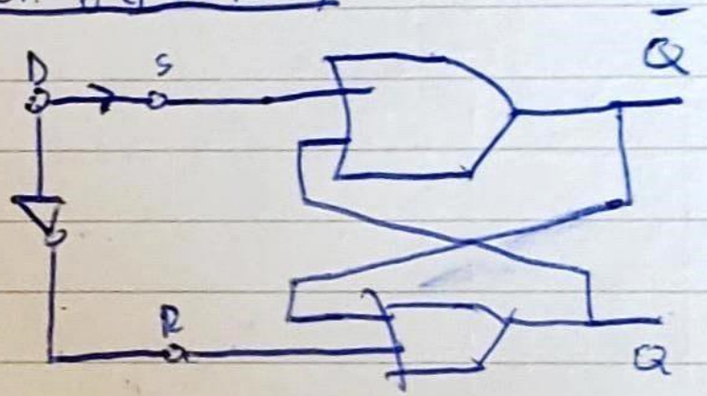
↳ clk ↳ copiază D

↳ mută în

Latch controlat (negativ):



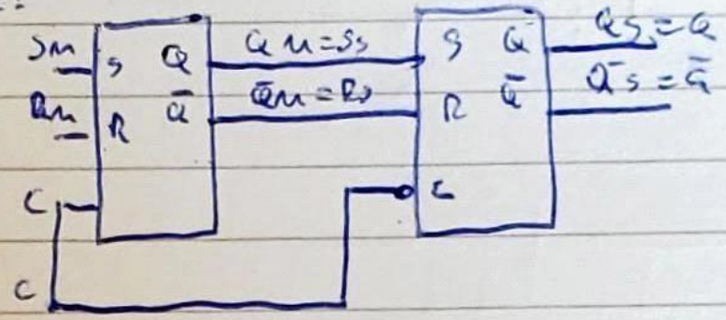
Latch D (pozitiv):



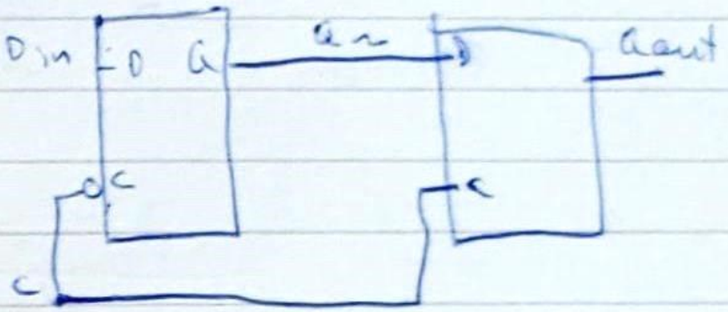
Flip-Flop: = latch, mas em tempo discreto

Mestre-Escrava:

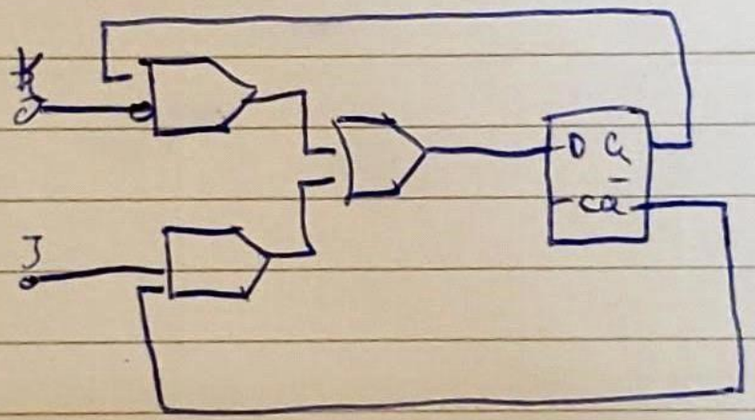
SR:



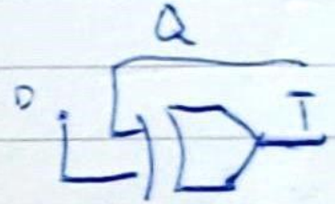
D:



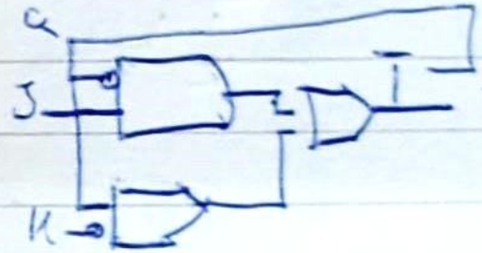
JK → D



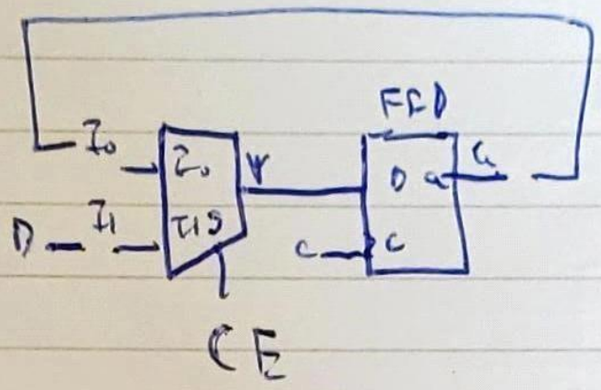
T → D:



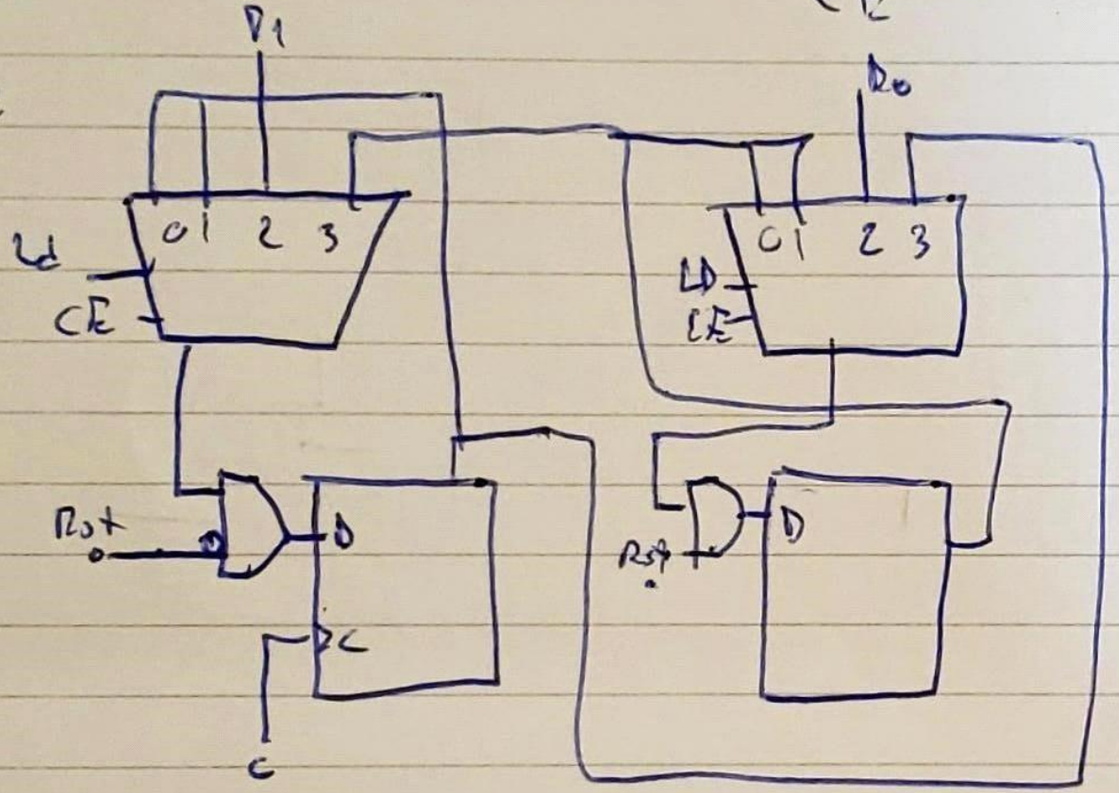
T → JK



Flip-Flop : Entradas Multiplexadas



Confusión :



Verilog

```
module (input , x, y, output, z);
  reg  z_reg;
  assing z = z_reg
  always (@ posedge clk or negedge clk)
          negedge clk - Switch
          posedge clk → Desicion
                                begin
    begin
      case (@ x, y)
          2'b00:
              z_reg =  ;
          ...
      endcase
      if (x == 1) begin
              z_reg = ;
      end
endmodule
```