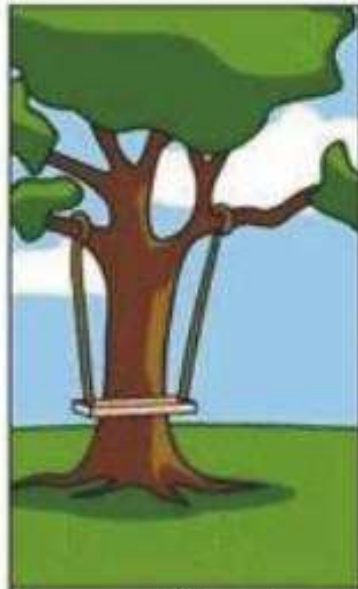




Como o cliente explicou...



Como o líder de projeto entendeu...



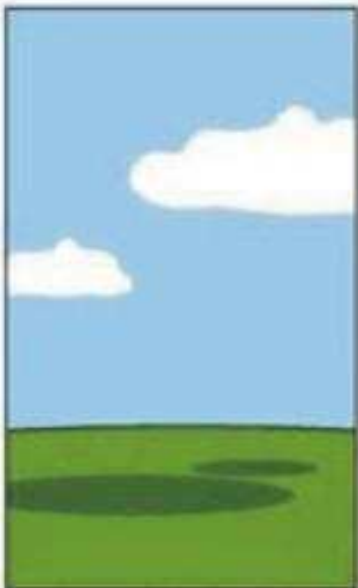
Como o analista projetou...



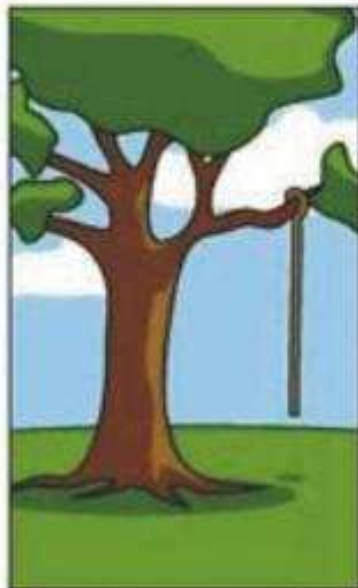
Como o programador construiu...



Como o consultor de negócios descreveu...



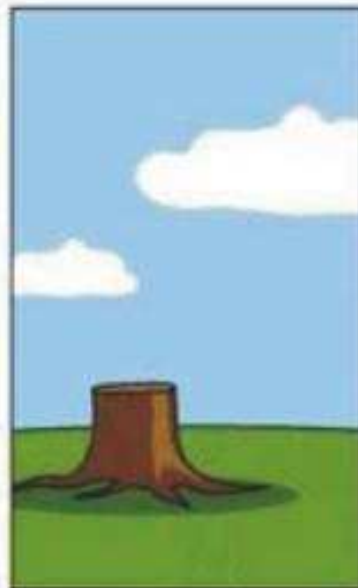
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...

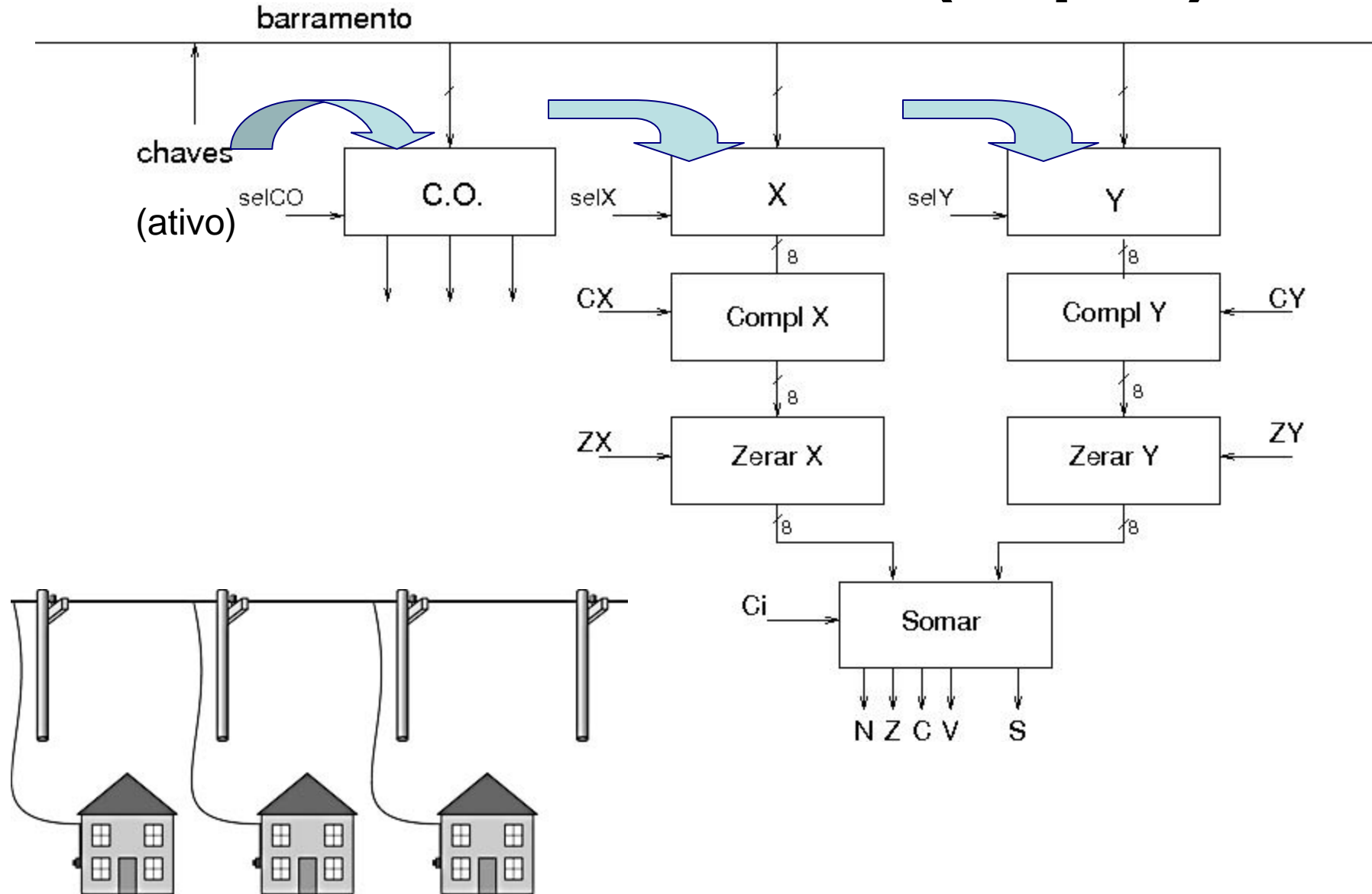


Como foi mantido...

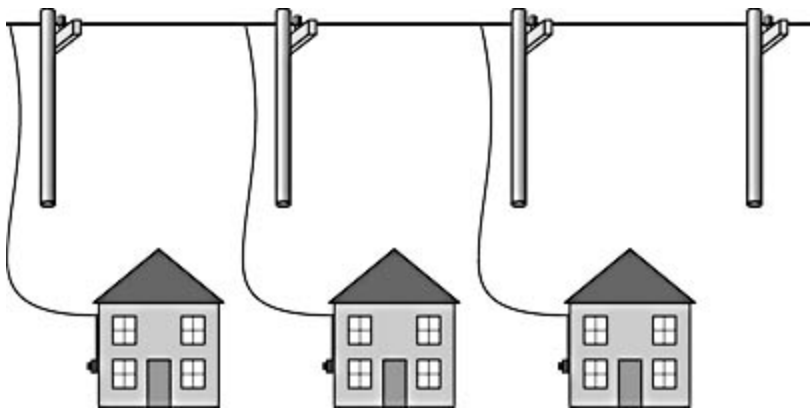
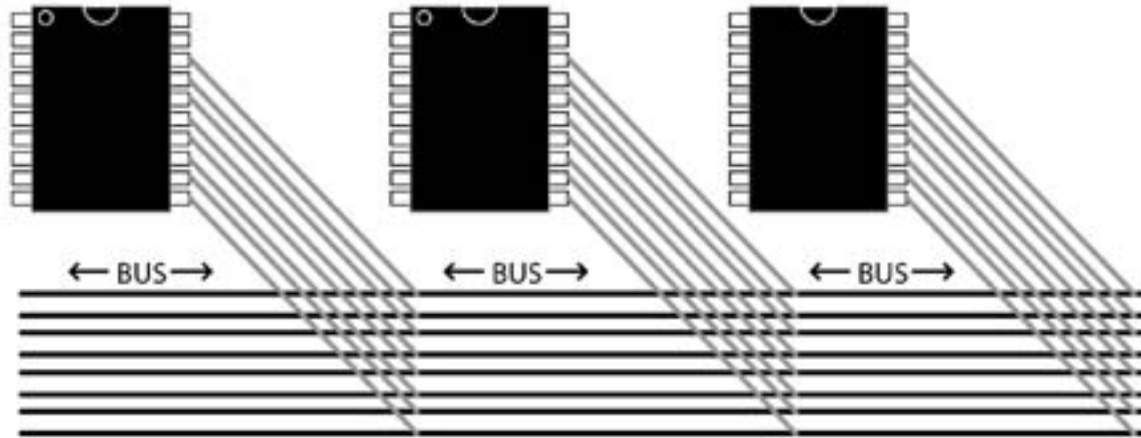


O que o cliente realmente queria...

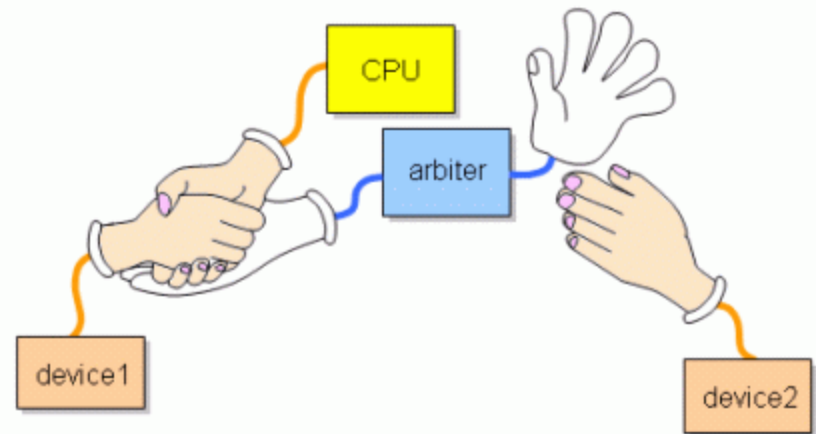
# Unidade Aritmética (Exp. 4)



# Barramento

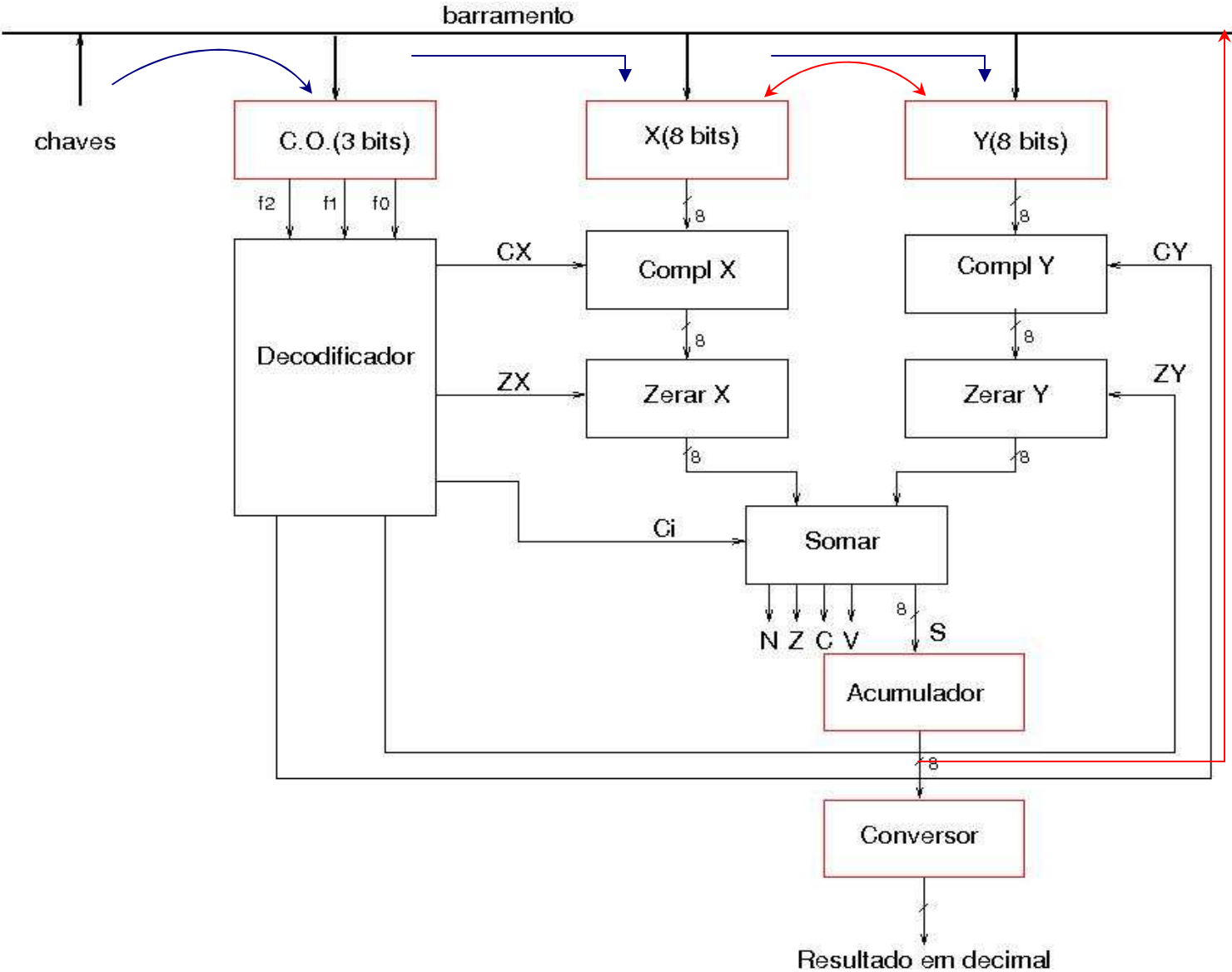


1 dispositivo ativo

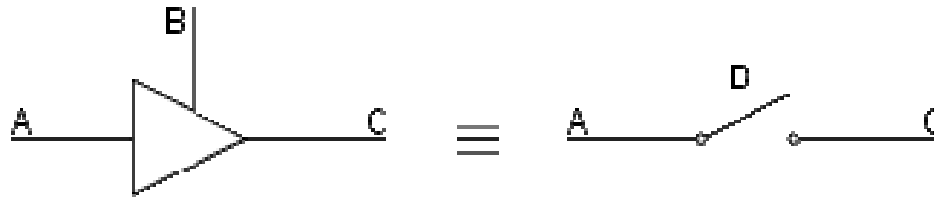


Mais de um dispositivo ativo

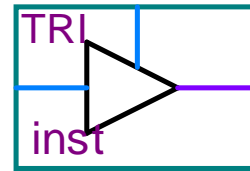
# Calculadora com Memória (Exp.5)



# Portas com saída *Tri-State*



**Símbolo:**



## Tabela-verdade

**Entrada Saída**

A	B	C
0	0	Z
1	0	Z
0	1	0
1	1	1

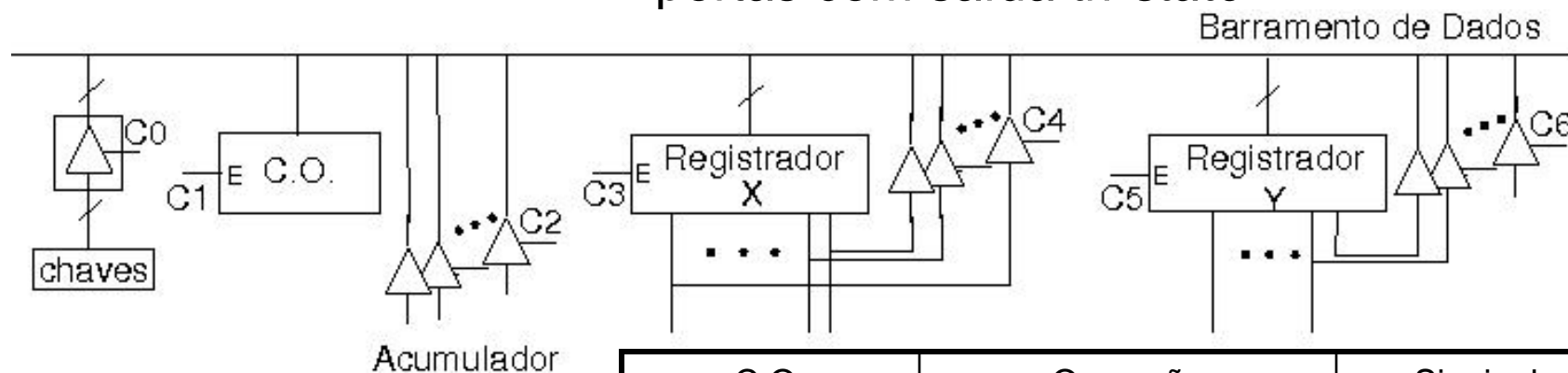
**VHDL:**

```
library altera;
use altera.altera_primitives_components.all;

-- Instantiating TRI
<instance_name> : TRI
-- <data_out> may feed an inout pin
port map (a_in => <data_in>, oe =>
          <enable_signal>, a_out => <data_out>);
```

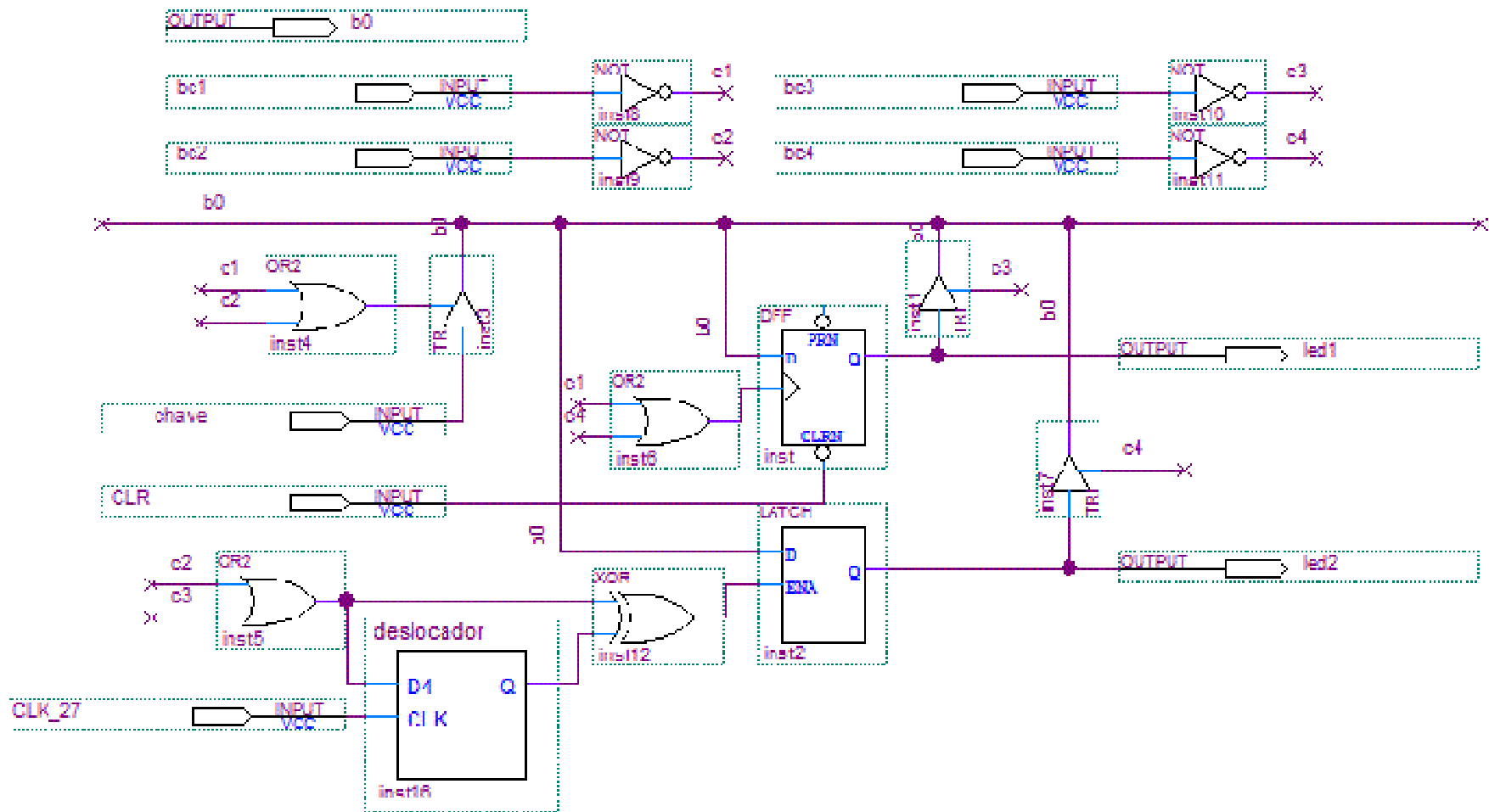
# 1º. Experimento

Projete um circuito que sincronize acessos “ativos” ao barramento por chaves, X, Y e Acc, com uso de portas com saída *tri-state*



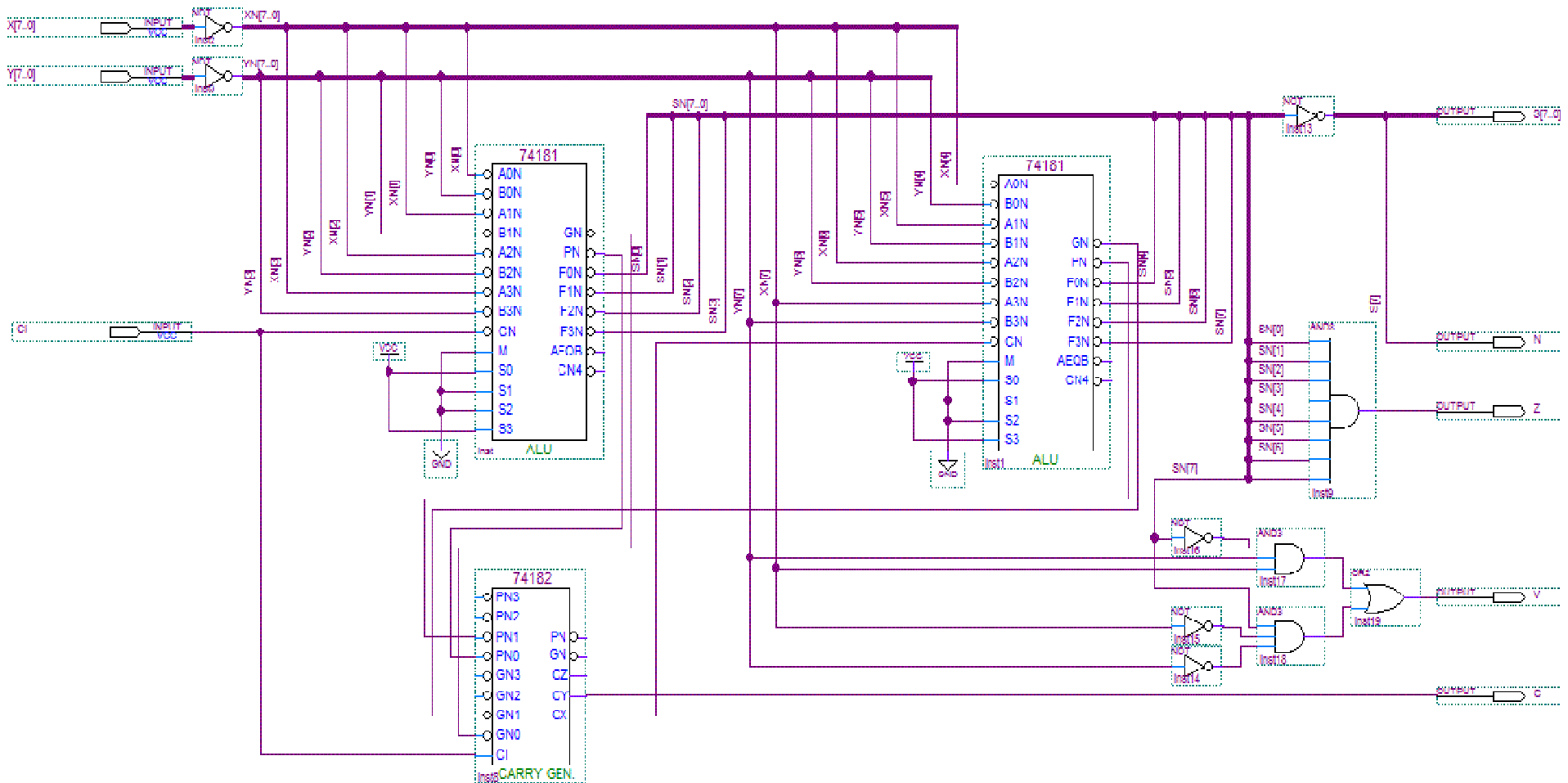
C.O.	Operação	Sinais de controle
(No.+7)%8	MXY $Y \leftarrow (X)$	$C0=C1=C2=C3=C6=0$ $C4=1; C5=1$
(No.+6)%8	MYX $X \leftarrow (Y)$	$C0=C1=C2=C4=C5=0$ $C3=1; C6=1$
(No.+5)%8	MAY $Y \leftarrow (\text{Acc})$	$C0=C1=C3=C4=C6=0$ $C2=1; C5=1$
	$Y \leftarrow \text{chaves}$	$C1=C2=C3=C4=C6=0$ $C0=1; C5=1$
	$\text{C.O.} \leftarrow \text{chaves}$	$C2=C3=C4=C5=C6=0$ $C0=1; C1=1$

# 1º. Experimento



Como deve ser o sincronismo dos sinais de controle que atuam no par de componentes: (chave → ffD), (chave → latch), (ffD → latch), (latch → ffD)?

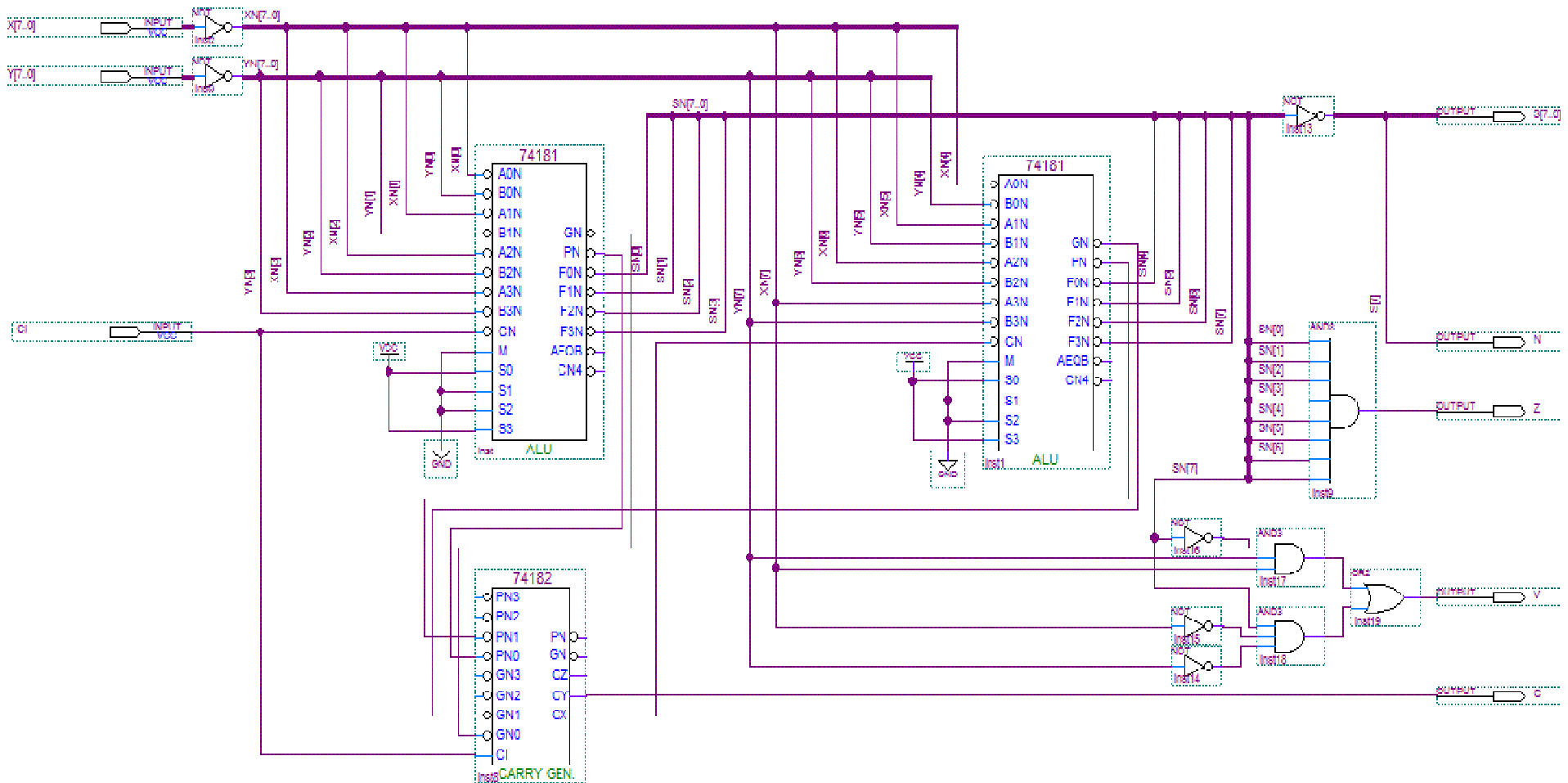
# 2º. Experimento: Bloco “Somar”



G → gerar  
P → propagar

Utilizar componentes 74181 e 74182

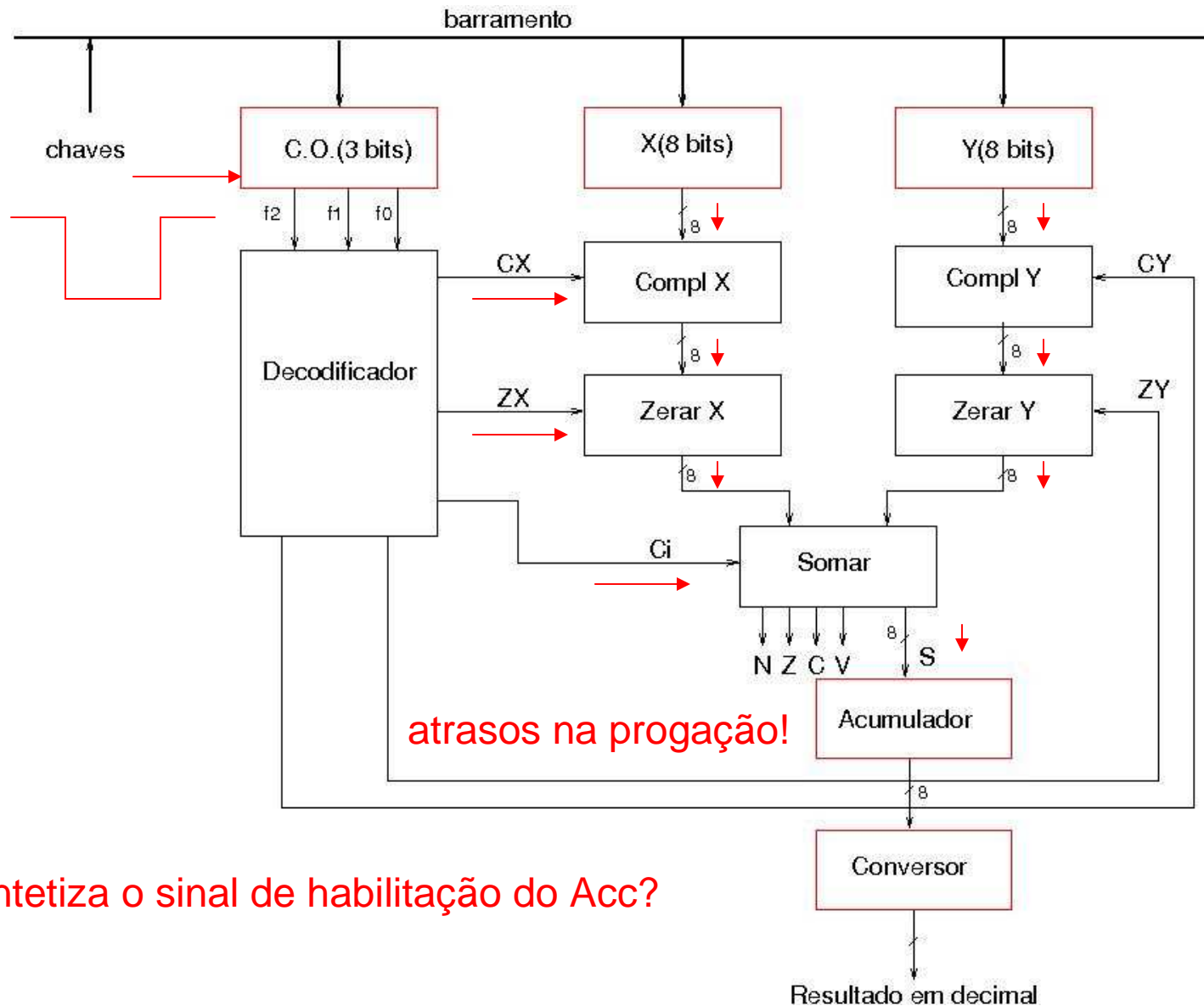
# 2º. Experimento: Bloco “Somar”



G → gerar  
P → propagar

Utilizar componentes 74181 e 74182

# 2º. Experimento: Acumulador

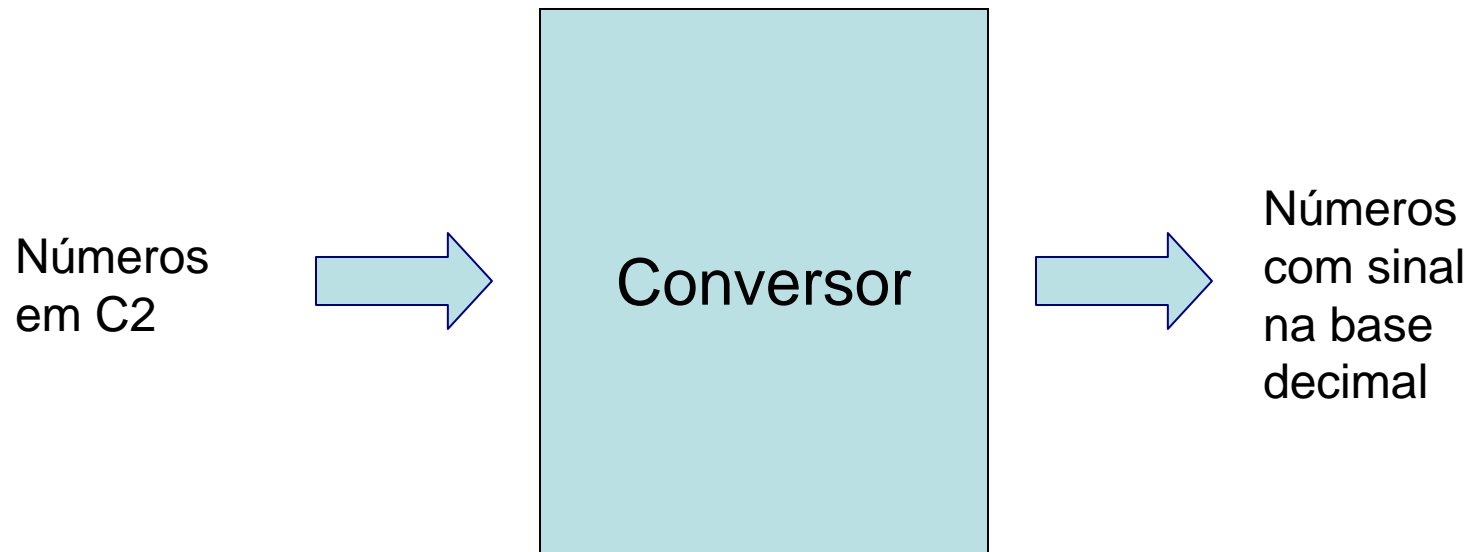


Como sintetiza o sinal de habilitação do Acc?

# 3º. Experimento: Decodificador

C.O.	Operação		Sinais de controle
(No.+7)%8	MXY	$Y \leftarrow (X)$	$C_0=C_1=C_2=C_3=C_6=0$ $C_4=1; C_5=1$
(No.+6)%8	MYX	$X \leftarrow (Y)$	$C_0=C_1=C_2=C_4=C_5=0$ $C_3=1; C_6=1$
(No.+5)%8	MAY	$Y \leftarrow (\text{Acc})$	$C_0=C_1=C_3=C_4=C_6=0$ $C_2=1; C_5=1$
		$Y \leftarrow \text{chaves}$	$C_1=C_2=C_3=C_4=C_6=0$ $C_0=1; C_5=1$
		$\text{C.O.} \leftarrow \text{chaves}$	$C_2=C_3=C_4=C_5=C_6=0$ $C_0=1; C_1=1$
(No.+3)%8	ADD	$\text{Acc} \leftarrow X+Y$	
(No.+1)%8	SUB	$\text{Acc} \leftarrow -X+Y$	
(No.+2)%8	CMY	$\text{Acc} \leftarrow Y'$	
(No.+0)%8	CSX	$\text{Acc} \leftarrow -X$	
(No.+4)%8	NOP		

# 4º. Experimento: Conversor



## Alternativas:

1. Tabela-verdade: Dígitos Binários  $\leftrightarrow$  Códigos Decimais
2. Contadores (binários – 74161, 74163; e decimais – 74160, 74162)  $\rightarrow$  Exp.2
3. Algoritmo de conversão:
  1. Extrair o valor absoluto  $x$  do número
  2. **Dividir** sucessivamente  $x$  por 10 e concatenar os dígitos dos restos.

# VHDL: Operações Lógico-Aritméticas

```
library ieee;use
ieee.std_logic_1164.all;
use ieee.numeric_std.all;
```

**and**  
**or**  
**nand**  
**nor**  
**xor**  
**xnor**

**\*\*** exponenciação;  
**abs** valor absoluto;  
**not** complemento;  
**\*** multiplicação;  
**/** divisão;  
**mod** módulo;  
**rem** resto da divisão;  
**+** soma ou identidade;  
**-** Subtração ou negação;

**Alguns tipos de sinais:**

**bit** → '0', '1'

**std\_logic** → '0', '1', 'X', 'Z'

**unsigned** → inteiro sem sinal

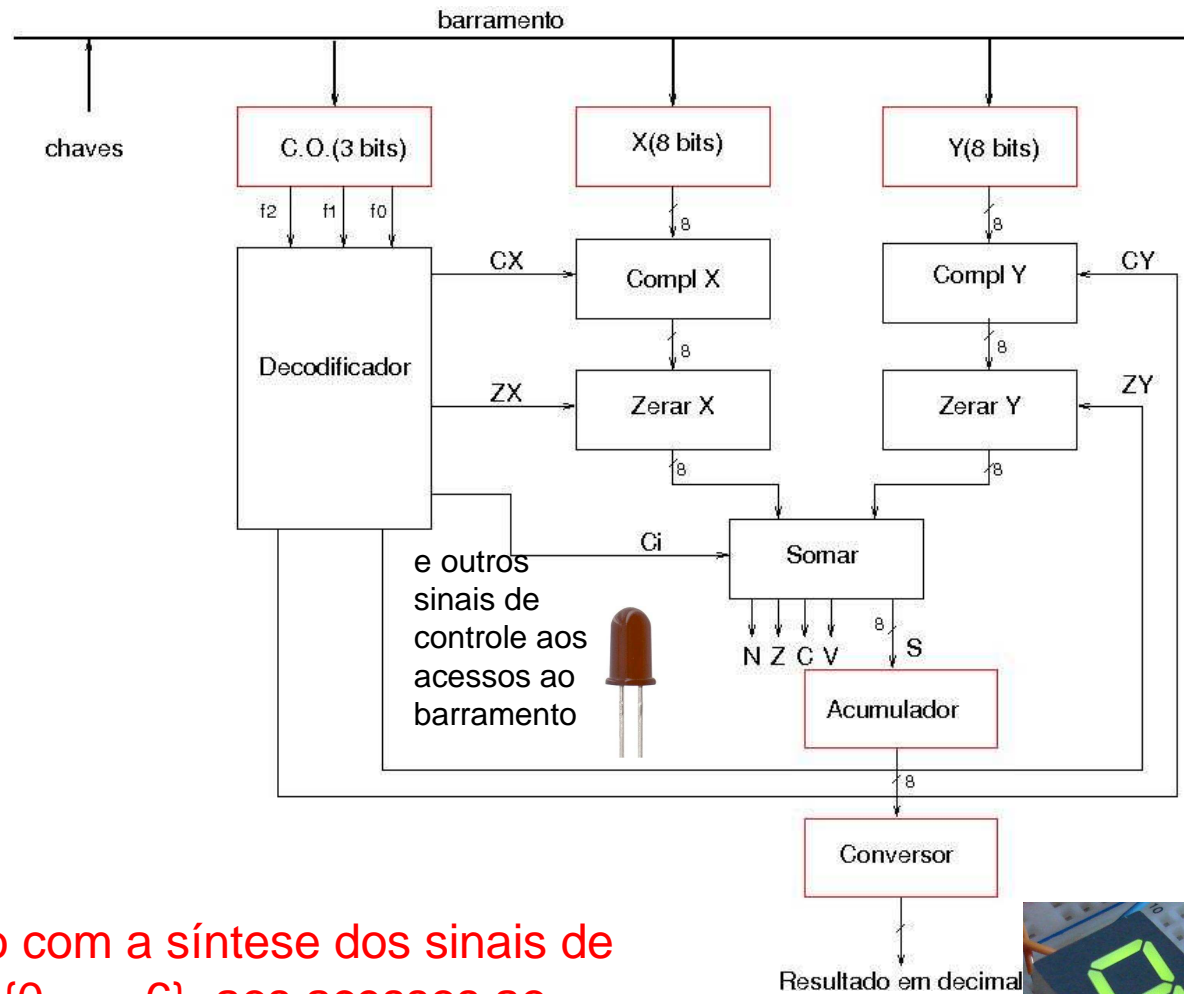
**signed** → complemento 2

**integer** → **signed** em, no mínimo, 32 *bits*

**Conversão entre os tipos de dados:**

1. **Unsigned, signed** → **std\_logic\_vector**  
std\_logic\_vector(v)
2. **Std\_logic\_vector** → **unsigned, signed**  
unsigned(v)  
signed(v)
3. **Unsigned, signed** → **integer**  
to\_integer(v)
4. **Integer** → **unsigned, signed**  
to\_unsigned(v, v'length)  
to\_signed(v, v'length)

# 5º. Experimento: Integração



Muito cuidado com a síntese dos sinais de controle  $C_j$ ,  $j=\{0, \dots, 6\}$ , aos acessos ao barramento!!!

