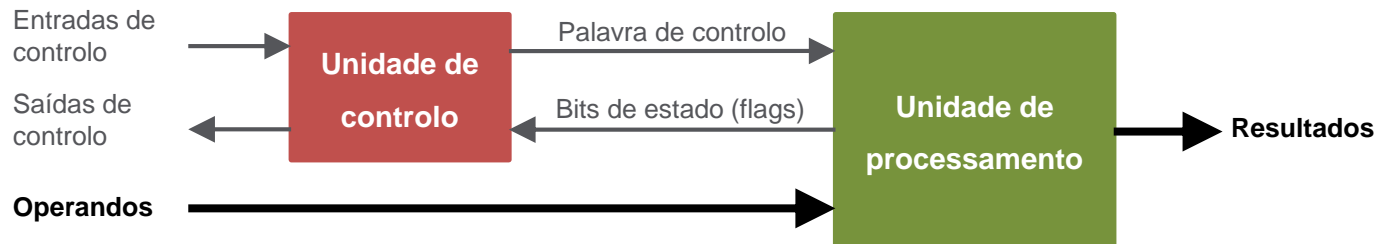


# Sistemas Digitais (SD)

## Máquinas de Estado Microprogramadas: Circuitos de Controlo, Transferência e Processamento de Dados



## ■ Na aula anterior:

- ▶ Projecto de máquinas de estados microprogramadas:
  - com endereçamento explícito
  - com endereçamento implícito
- ▶ Exemplos



SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
17/Fev a 21/Fev	Introdução	Sistemas de Numeração	
24/Fev a 28/Fev	<b>CARNAVAL</b>	Álgebra de Boole	P0
02/Mar a 06/Mar	Elementos de Tecnologia	Funções Lógicas	VHDL
9/Mar a 13/Mar	Minimização de Funções	Minimização de Funções	L0
16/Mar a 20/Mar	Def. Circuito Combinatório; Análise Temporal	Circuitos Combinatórios	P1
23/Mar a 27/Mar	Circuitos Combinatórios	Circuitos Combinatórios	<b>L1</b>
30/Mar a 03/Abr	Circuitos Sequenciais: Latches	Circuitos Sequenciais: Flip-Flops	P2
06/Abr a 10/Abr	<b>FÉRIAS DA PÁSCOA</b>	<b>FÉRIAS DA PÁSCOA</b>	<b>FÉRIAS DA PÁSCOA</b>
13/Abr a 17/Abr	Caracterização Temporal	Registos	L2
20/Abr a 24/Abr	Contadores	Circuitos Sequenciais Síncronos	P3
27/Abr a 01/Mai	Síntese de Circuitos Sequenciais Síncronos	Síntese de Circuitos Sequenciais Síncronos	L3
04/Mai a 08/Mai	Exercícios	Memórias	P4
11/Mai a 15/Mai	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Microprograma	L4
18/Mai a 22/Mai	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	P5
25/Mai a 29/Mai	P6	P6	L5

Teste 1

## ■ Tema da aula de hoje:

- ▶ Circuitos de controlo, transferência e processamento de dados
- ▶ Exemplo de uma arquitectura simples de um processador

## □ Bibliografia:

- M. Mano, C. Kime: Capítulo 7
- G. Arroz, J. Monteiro, A. Oliveira: Secções 8.2 a 8.3

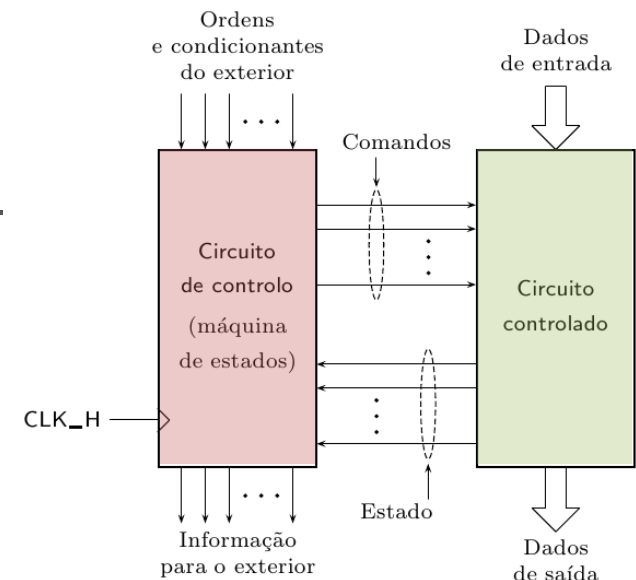
## ■ Circuito de Dados e Circuito de Controlo (Revisão)

► Os sistemas digitais com alguma complexidade tornam-se difíceis de projectar como vulgares máquinas sequenciais síncronas, porque:

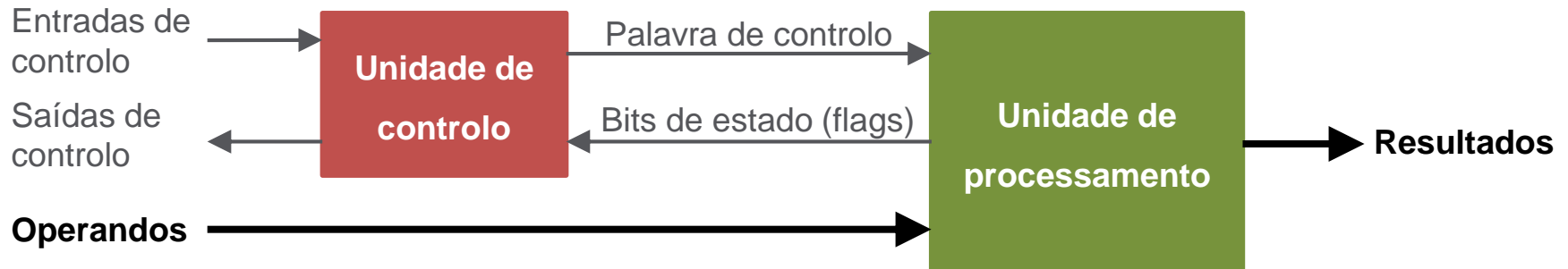
- Diagramas de estados / tabela de estados de grande dimensão
- Elevado número de:
  - Entradas,
  - Saídas,
  - Estados.

**Solução:** organizar esses sistemas hierarquicamente, estabelecendo uma divisão entre:

- **circuito de dados** - dá suporte ao fluxo e à manipulação de dados;
- **circuito de controlo** - controla o circuito de dados.



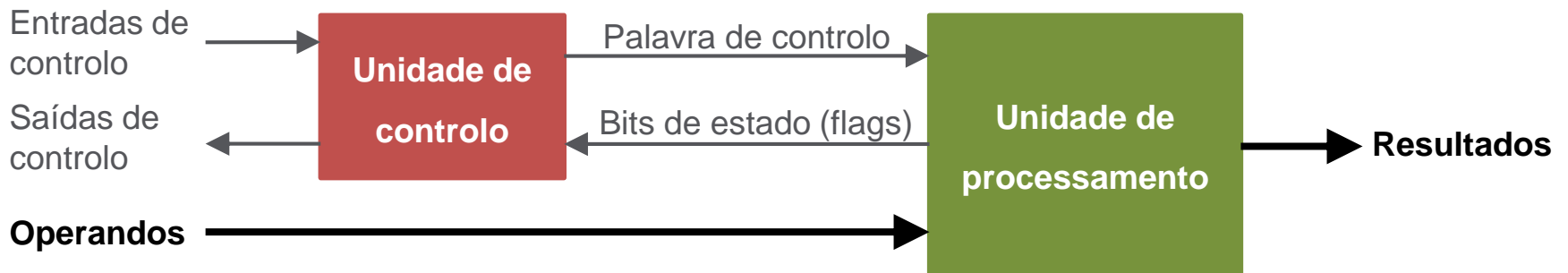
## ■ Unidade de Processamento e Unidade de Controlo



► A partir de um certo nível de complexidade, os circuitos digitais podem ser divididos em dois módulos distintos:

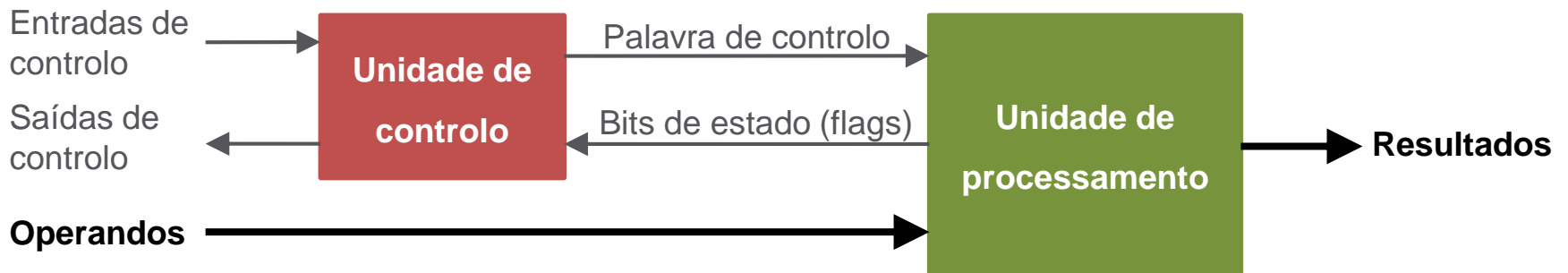
- **Circuito de dados** ou unidade de processamento ou *datapath*;
- **Circuito de controlo** ou unidade de controlo

## ■ Unidade de Processamento e Unidade de Controlo



- ▶ A **unidade de processamento** processa a informação útil do sistema e é tipicamente constituída por um conjunto de **módulos combinatórios** (ex: ALU) e **elementos de memória** (ex: banco de registos, memória RAM, etc.).
- ▶ Sobre a informação contida nos elementos de memória (ex: registo), podem ser realizadas operações cujo resultado pode ser guardado no mesmo registo, noutro registo ou mesmo numa posição de memória.

## ■ Unidade de Processamento e Unidade de Controlo



- ▶ A **unidade de controlo** é responsável por gerar os sinais de controlo (**palavra de controlo**) que sequenciam as operações básicas do circuito de dados a cada ciclo de relógio, de modo a que o sistema realize operações complexas. Têm dois tipos de entradas:
  - **Entradas de controlo**, que controlam o funcionamento do sistema;
  - **Bits de estado**, provenientes do circuito de dados, com informação referente à última operação realizada pelo circuito de dados.



## ■ Exemplo: Maior Divisor Comum

### Pseudo-Código

**Maior\_Divisor\_Comum (X,Y)**

Enquanto (Y≠0) {

    se  $X \geq Y$

        então  $X = X - Y$

        se não, troca X com Y

}

Resultado em X

- ▶ Sugestão: verifique, através de um par de inteiros (ex: 54 e 36) que o resultado é o esperado.

## ■ Exemplo: Maior Divisor Comum

### Pseudo-Código

**Maior\_Divisor\_Comum (X,Y)**

Enquanto (Y≠0) {

se  $X \geq Y$

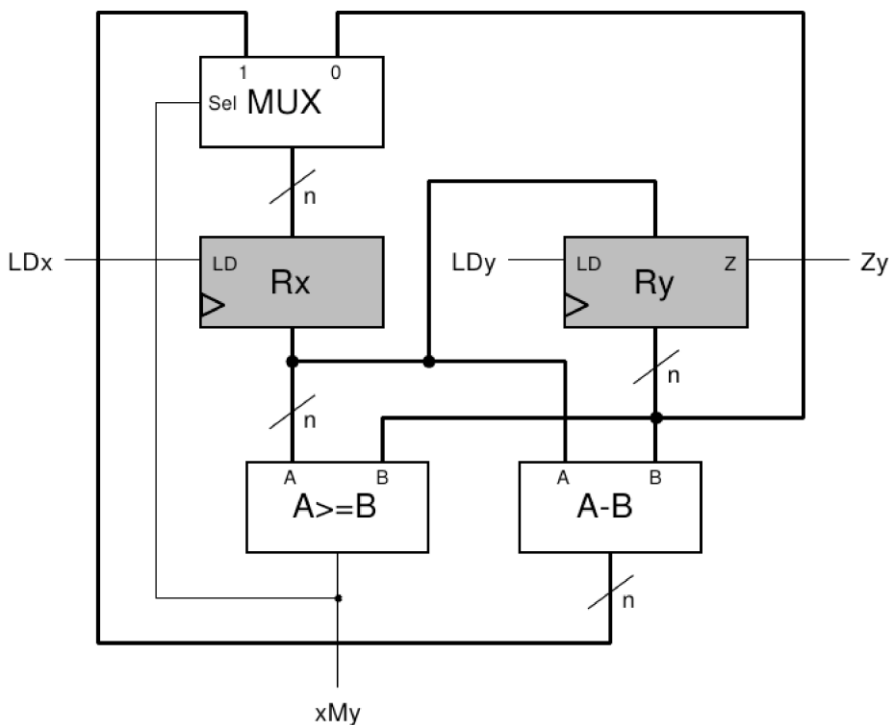
então  $X = X - Y$

se não, troca X com Y

}

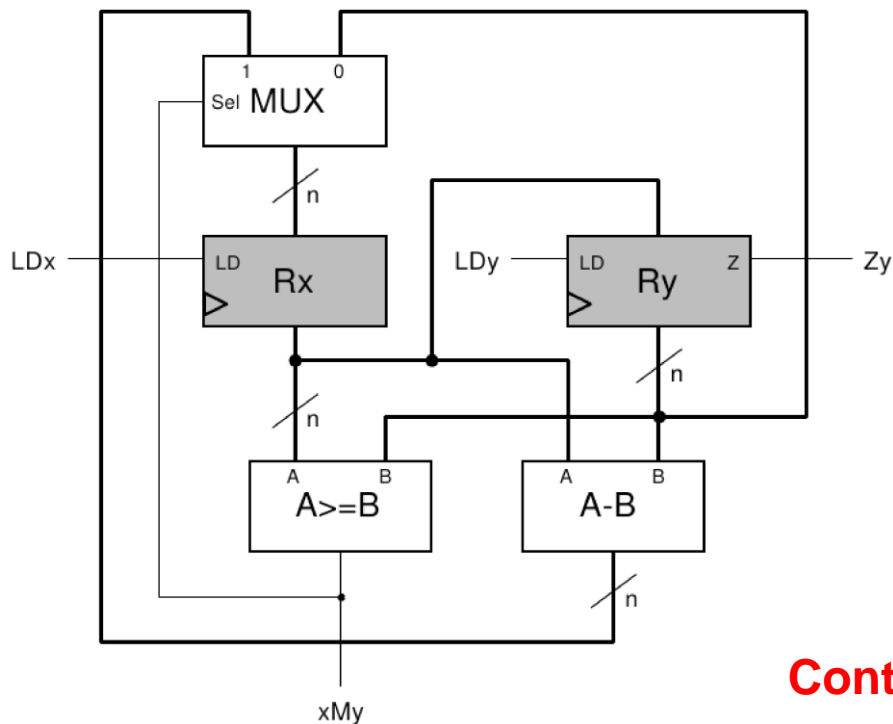
Resultado em X

### Unidade de Processamento



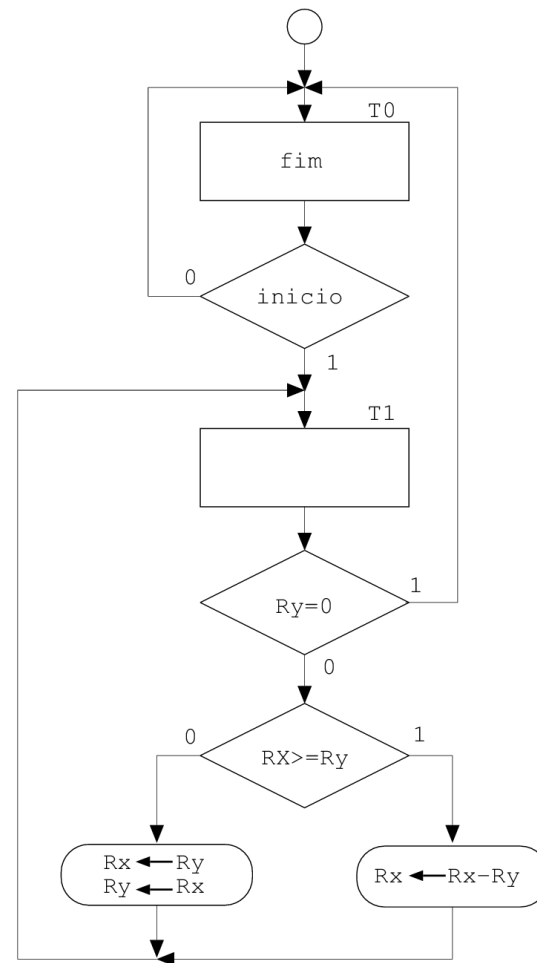
## ■ Exemplo: Maior Divisor Comum

### Unidade de Processamento



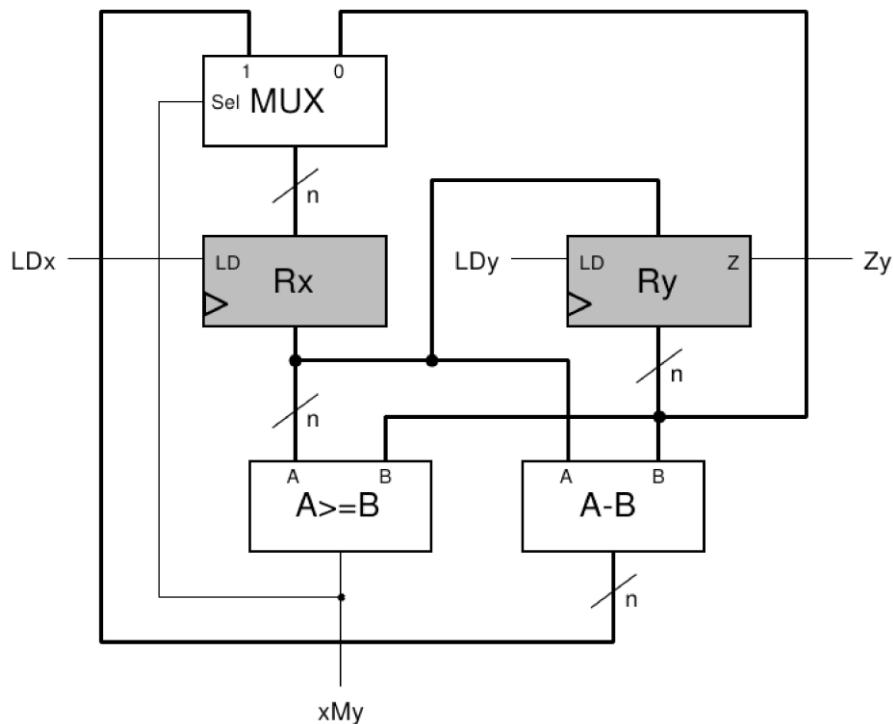
**Controlo?**

### Fluxograma

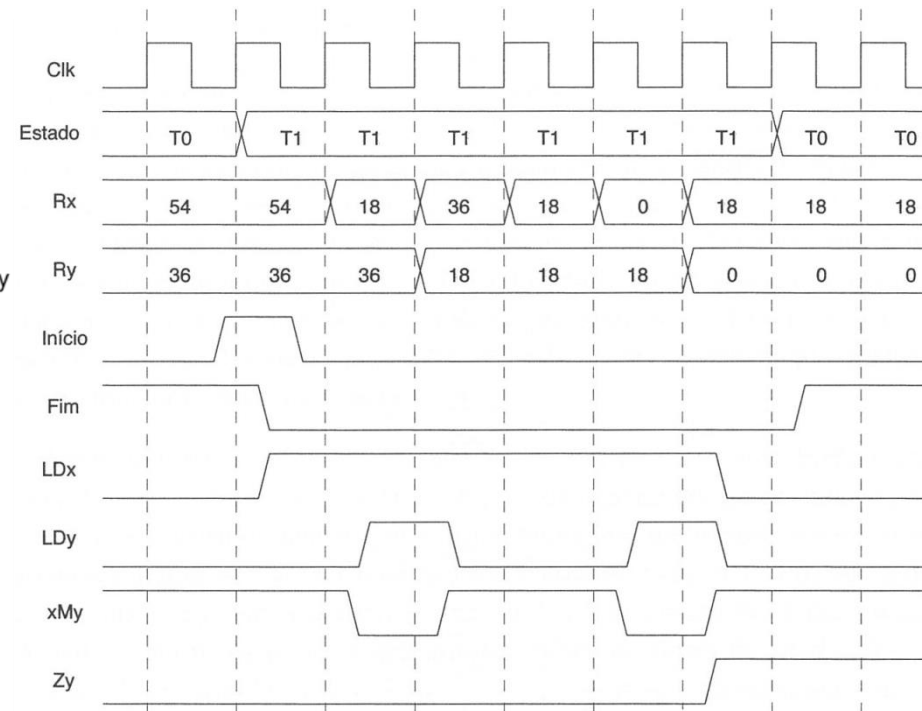


## ■ Exemplo: Maior Divisor Comum

### Unidade de Processamento

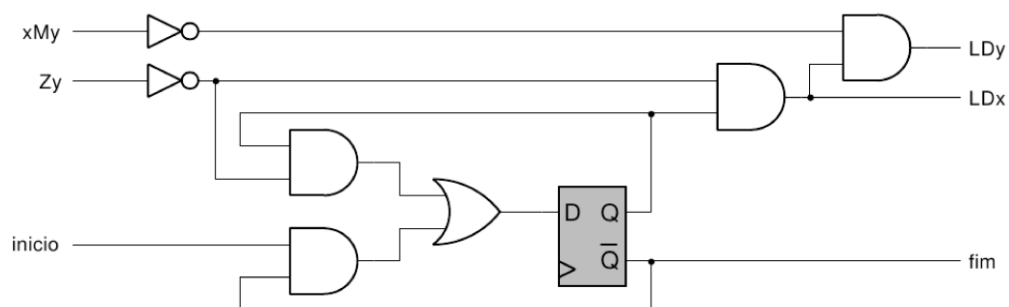


### Exemplo: mdc(54,36)

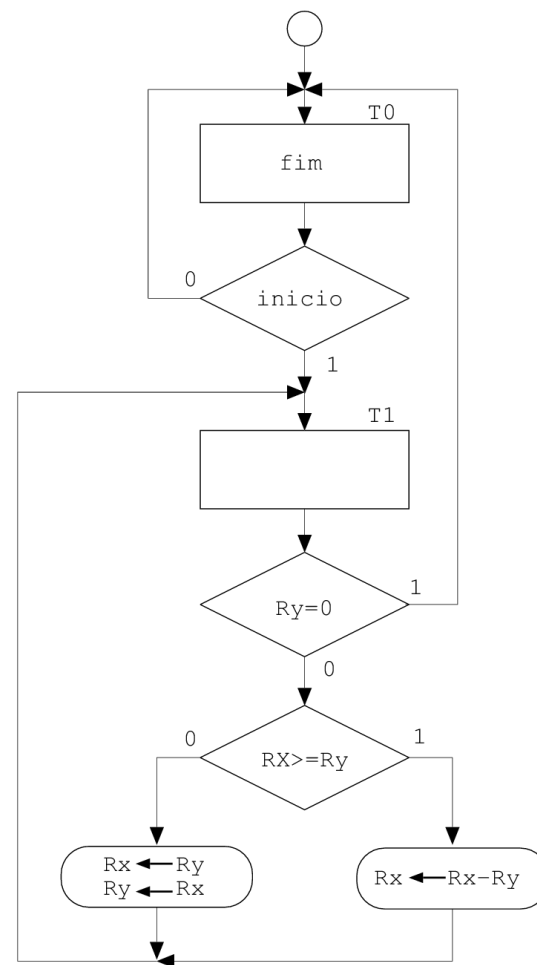


## ■ Exemplo: Maior Divisor Comum

### Unidade de Controlo

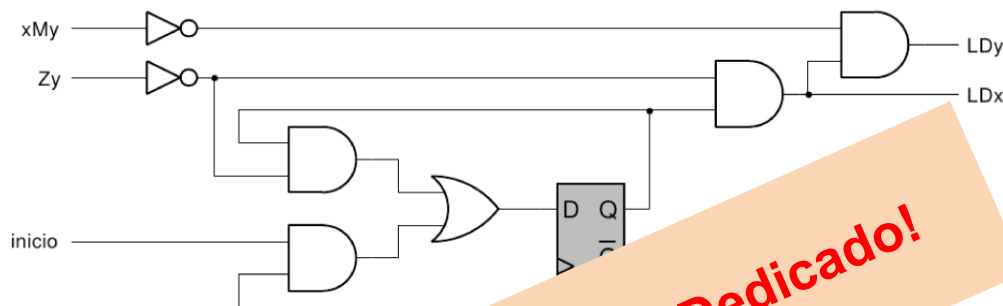


### Fluxograma

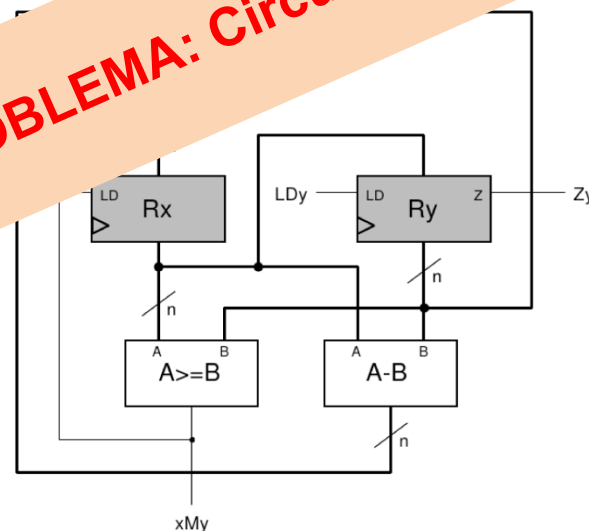


## ■ Exemplo: Maior Divisor Comum

## Unidade de Controlo:



## Unidade de Processamento:



**PROBLEMA: Circuito Dedicado!**

## ■ Problema:

- ▶ A síntese de circuitos de processamento e de controlo para a realização de operações complexas, com muitos estados e variáveis de entrada, torna-se complexa, trabalhosa e pouco eficiente

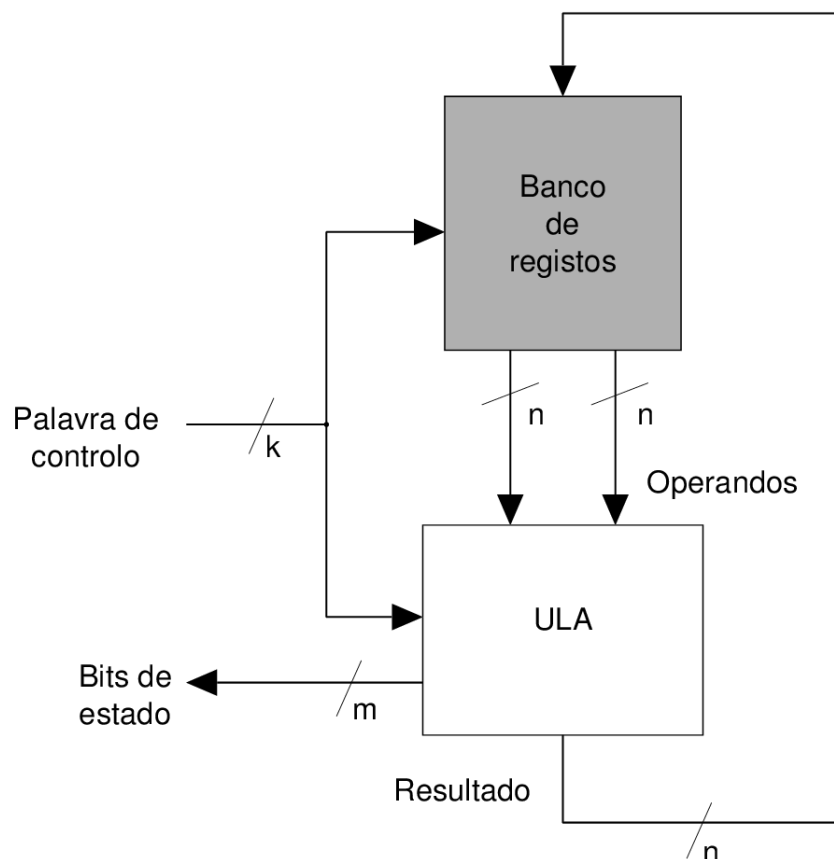


Esforço de desenvolvimento/implementação insustentável

## ■ Alternativa:

- ▶ Utilização de **unidades de processamento** genéricas e não dedicadas à aplicação;
- ▶ Utilização de **unidades de controlo** microprogramadas.

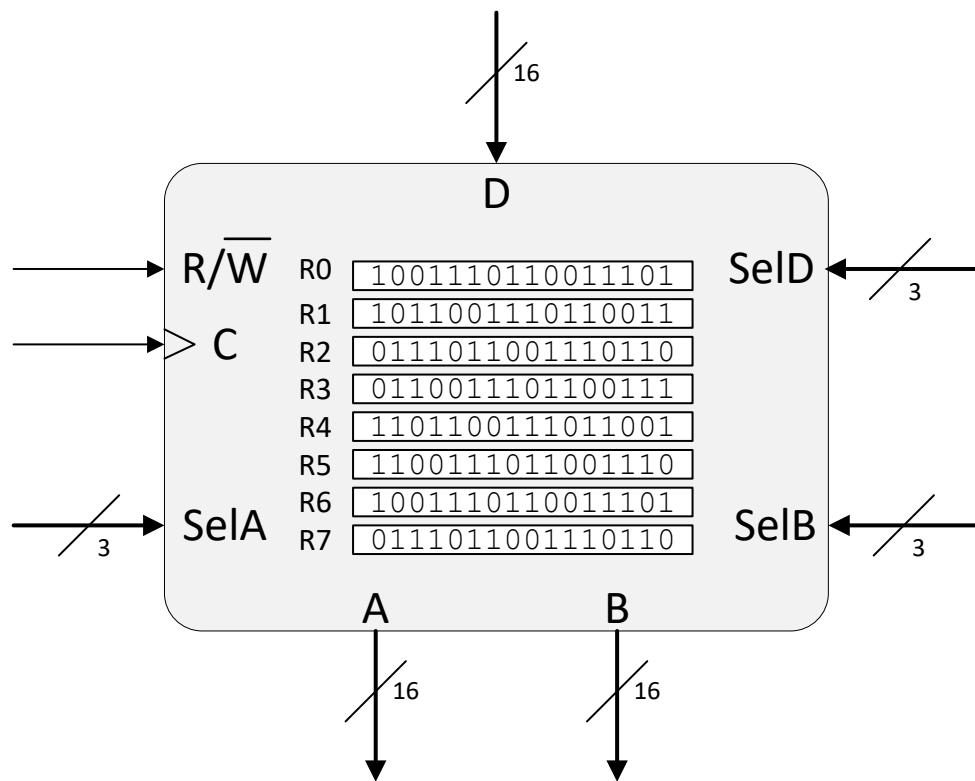
## ■ Unidade de Processamento Genérica





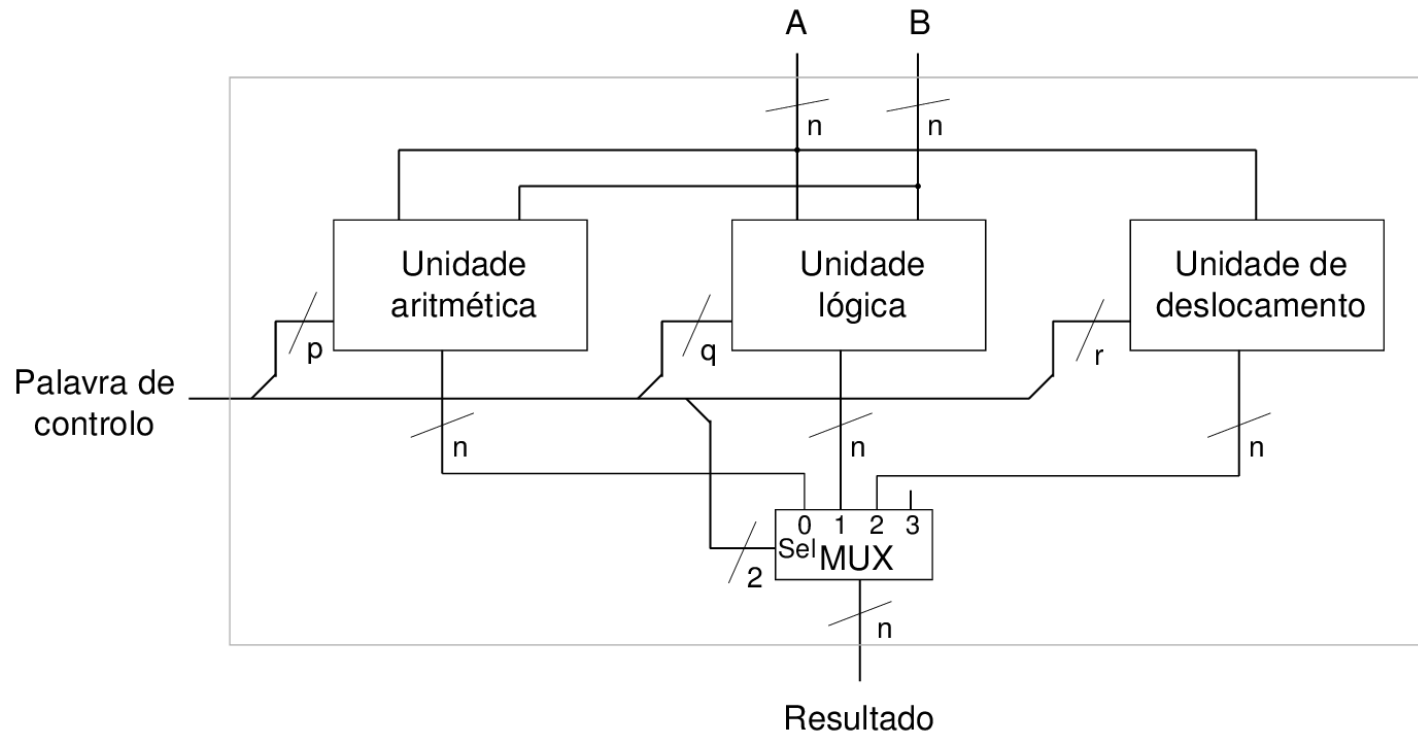
## ■ Unidade de Processamento Genérica

### Banco de Registos

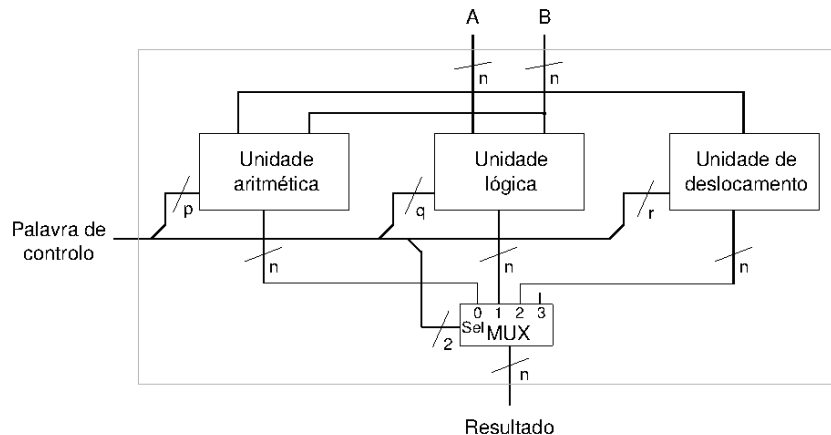


## ■ Unidade de Processamento Genérica

### Unidade Lógica e Aritmética (ULA)



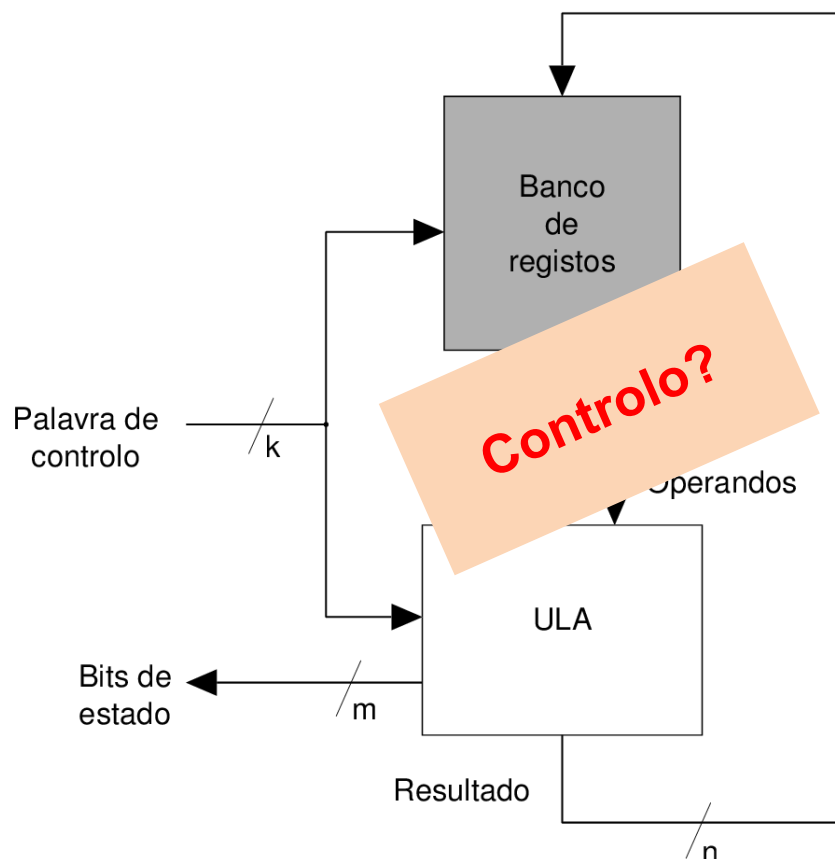
## ■ Operações da unidade lógica e aritmética



- ▶ A função realizada é definida por uma palavra de comando;
- ▶ Várias codificações possíveis.

$S_4S_3S_2S_1S_0$		Operação
00000	$R \leftarrow A + B$	Soma
00001	$R \leftarrow A - B$	Subtração
00010	$R \leftarrow A + B + C$	Soma com bit de transporte
00011	$R \leftarrow A - B - \overline{C}$	Subtração com transporte negado
00100	$R \leftarrow A - 1$	Decremento
00101	$R \leftarrow A + 1$	Incremento
00110	$R \leftarrow A - \overline{C}$	Decremento, se $C=0$
00111	$R \leftarrow A + C$	Incremento, se $C=1$
01-00	$R \leftarrow \overline{A}$	Complemento
01-01	$R \leftarrow A \wedge B$	Conjunção
01-10	$R \leftarrow A \vee B$	Disjunção
01-11	$R \leftarrow A \oplus B$	Disjunção exclusiva
10000	$R \leftarrow \text{SHR } A$	Deslocamento lógico à direita
10001	$R \leftarrow \text{SHL } A$	Deslocamento lógico à esquerda
10010	$R \leftarrow \text{SHRA } A$	Deslocamento aritmético à direita
10011	$R \leftarrow \text{SHLA } A$	Deslocamento aritmético à esquerda
10100	$R \leftarrow \text{ROR } A$	Rotação à direita
10101	$R \leftarrow \text{ROL } A$	Rotação à esquerda
10110	$R \leftarrow \text{RORC } A$	Rotação à direita com transporte
10111	$R \leftarrow \text{RORL } A$	Rotação à esquerda com transporte
11---	$R \leftarrow A$	Transferência

## ■ Unidade de Processamento Genérica



## ■ Unidade de Controlo Microprogramada

### ► Características:

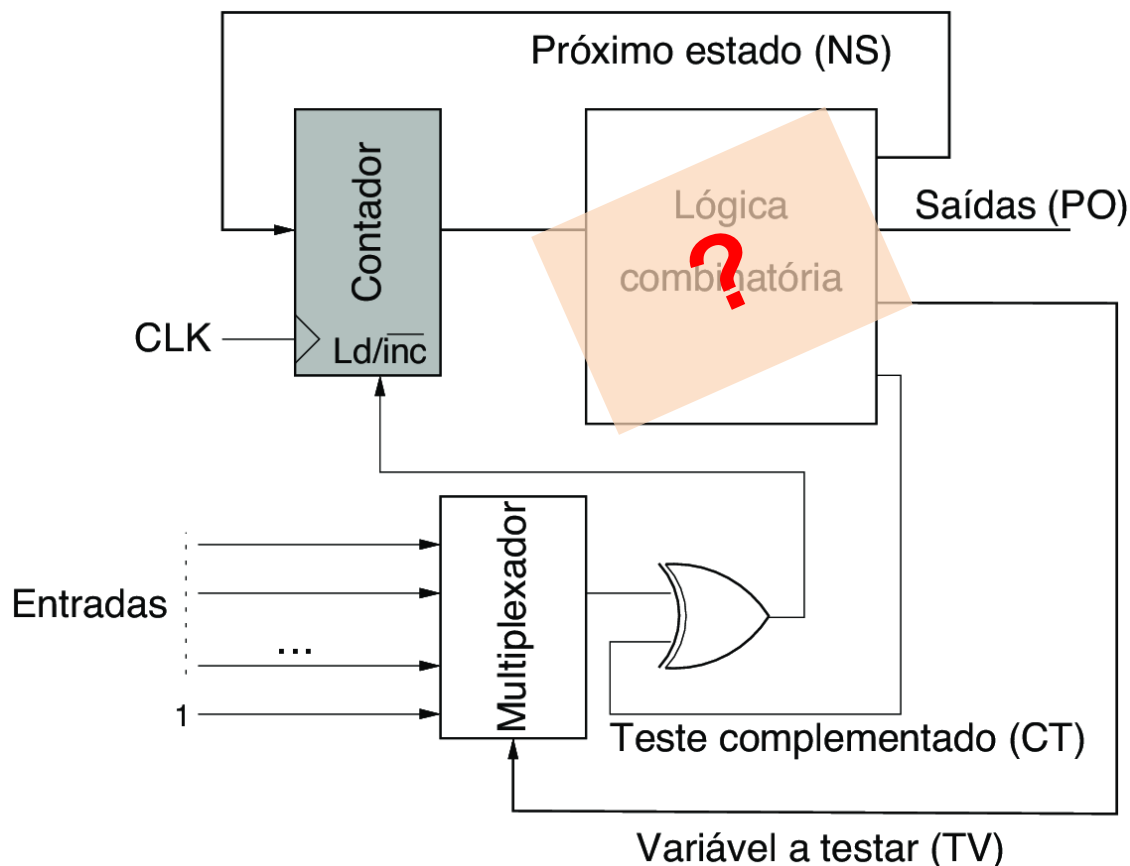
- As saídas dependem **apenas** do estado actual



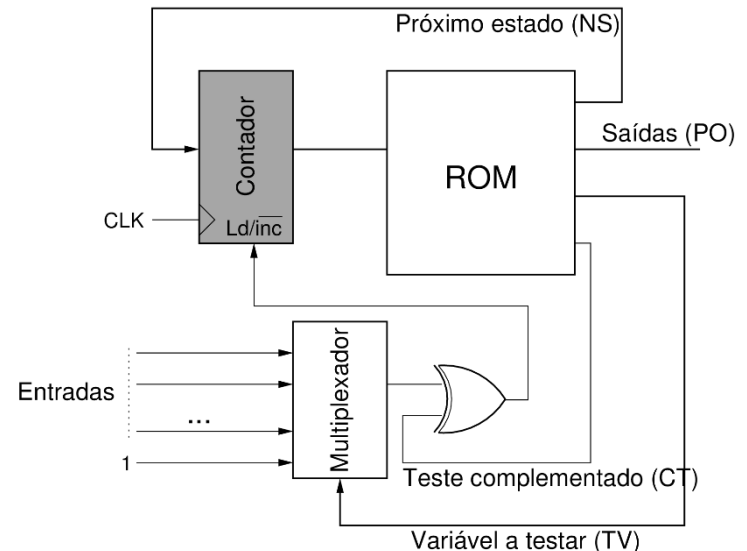
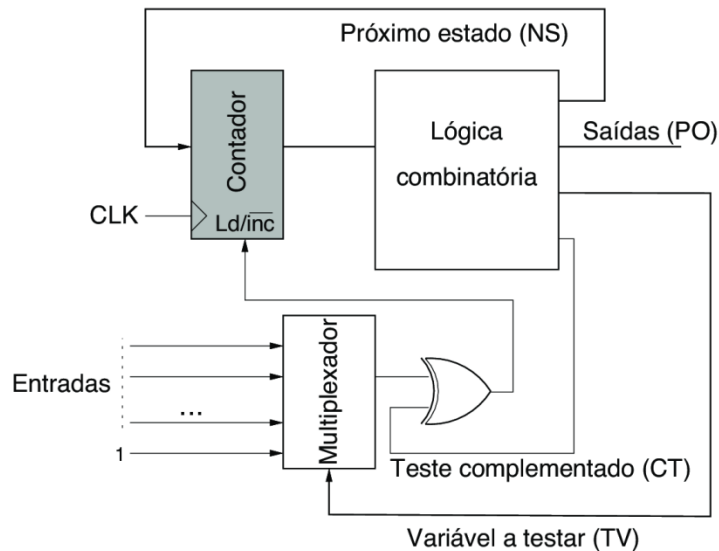
**Máquina de Moore**

- Em cada estado **apenas** é testada uma variável de entrada;
- Como resultado de cada teste numa variável de entrada, o controlador poderá saltar para um estado arbitrário (se o teste for verdadeiro) ou transitar para o estado seguinte (se o teste for falso)

## ■ Unidade de Controlo Microprogramada

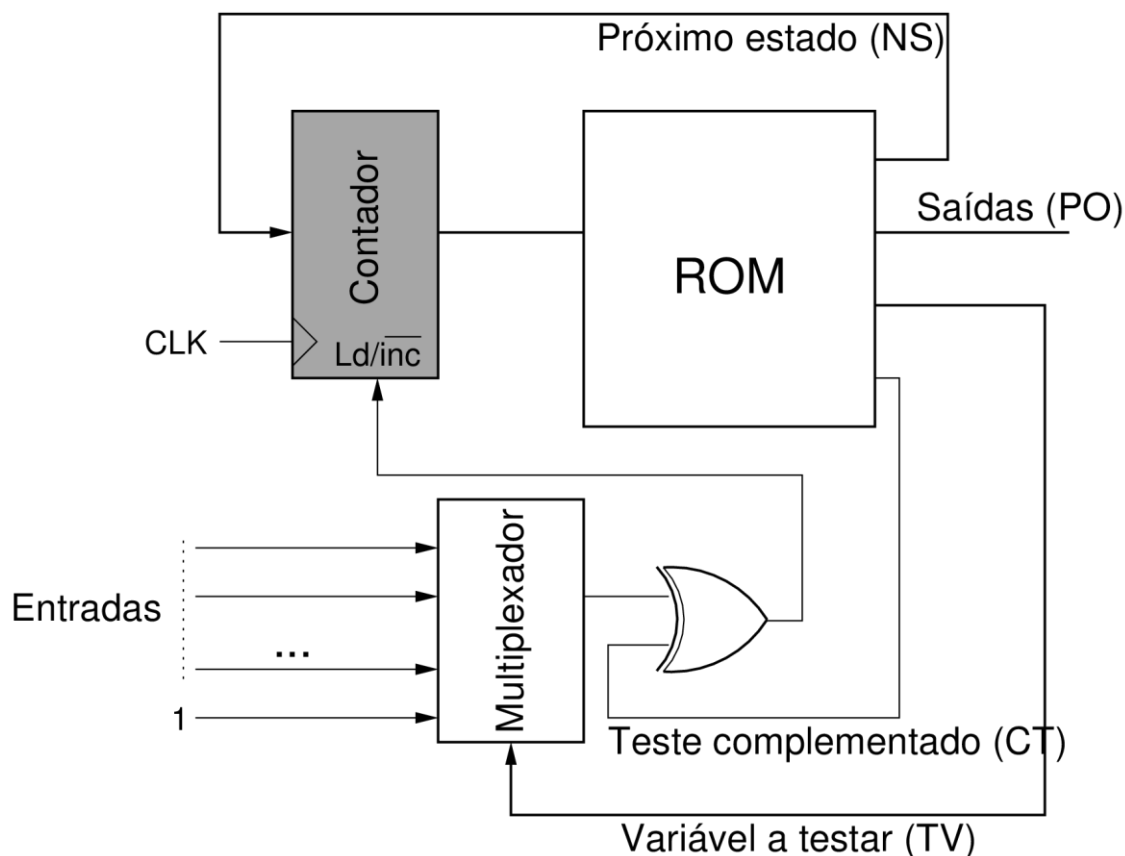


## ■ Unidade de Controlo Microprogramada



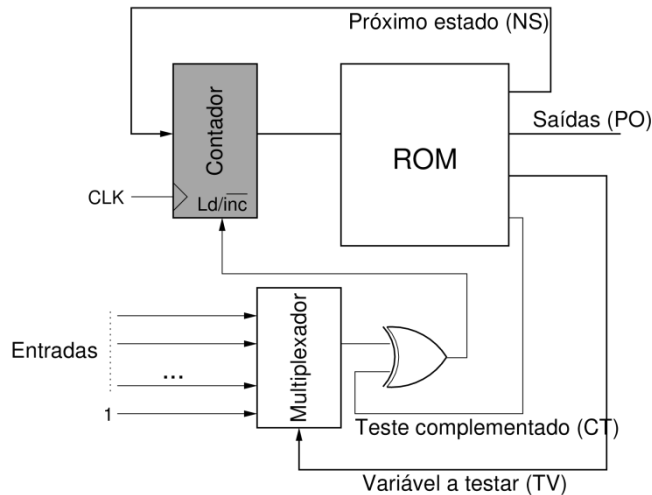
- O bloco combinatório pode ser realizado utilizando uma ROM com um número de linhas igual ao número de estados e com tantas saídas quantas as necessárias para gerar as variáveis de saída do bloco combinatório.

## ■ Unidade de Controlo Microprogramada





## ■ Unidade de Controlo Microprogramada



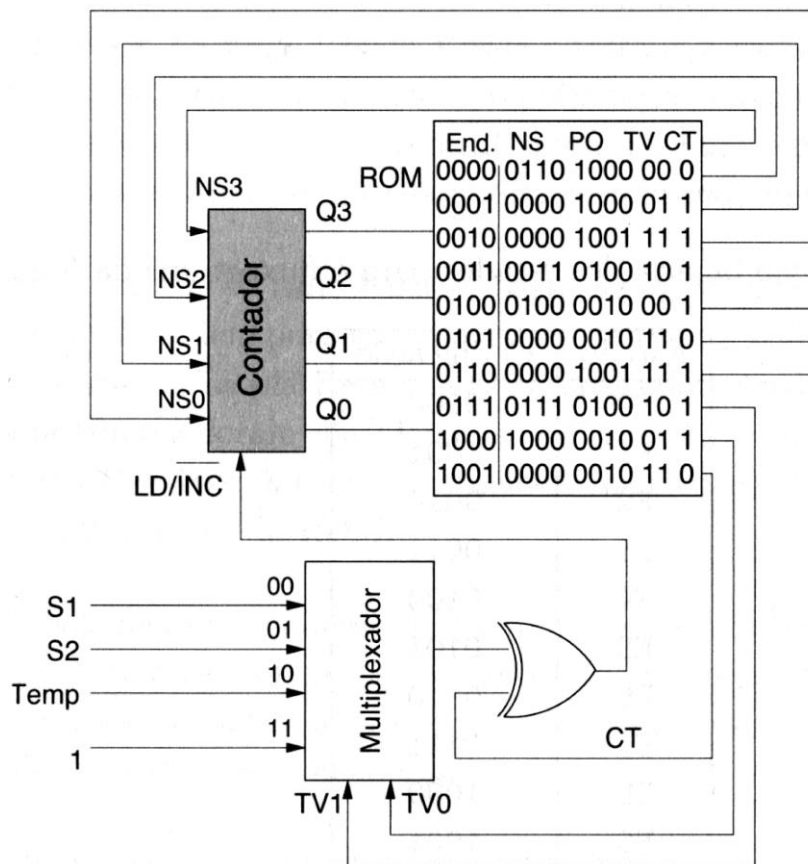
Cada palavra da ROM deverá conter os seguintes campos:

NS	PO	TV	CT

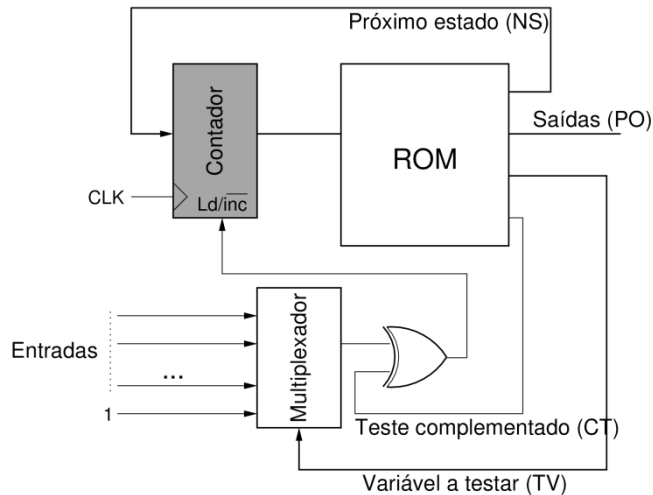
- **Next State (NS)** - especifica o próximo estado, para onde o controlador deverá saltar se o teste efectuado tiver resultado positivo;
- **Primary Output (PO)** - valores pretendidos para as variáveis de saída;
- **Test Variable (TV)** - indica a variável que deverá ser testada;
- **Complement Test (CT)** - indica se o salto deverá ocorrer quando a variável de teste está a 1 ou a 0.

## ■ Unidade de Controlo Microprogramada

► Exemplo:



## ■ Unidade de Controlo Microprogramada



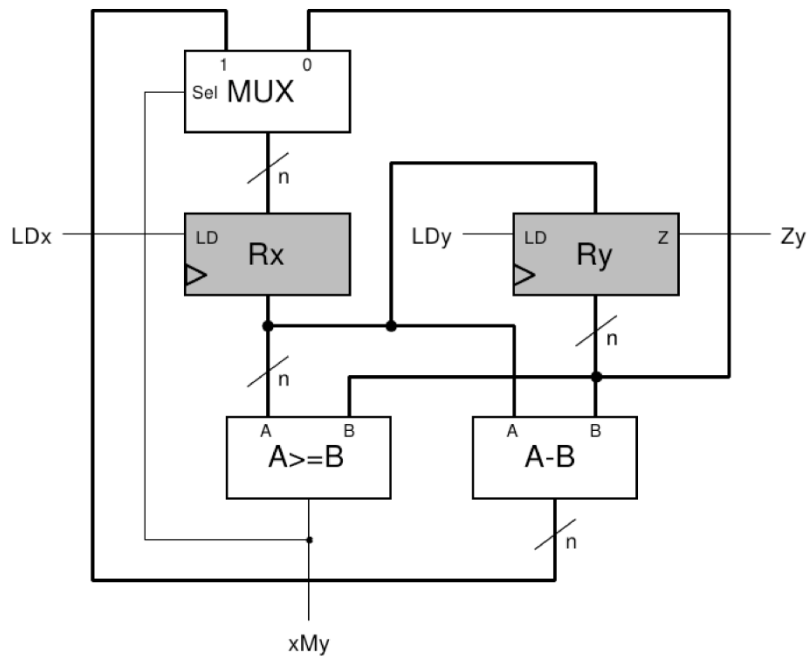
Cada palavra da ROM deverá conter os seguintes campos:

NS	PO	TV	CT

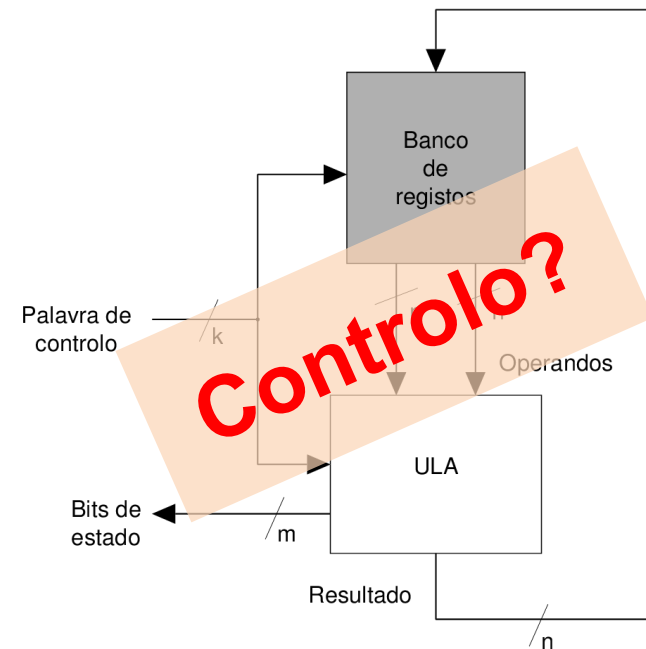
- ▶ Ao conjunto de campos que especificam o funcionamento do controlador chama-se **microinstrução**;
- ▶ Ao conjunto de microinstruções chama-se **microprograma**.

## ■ Exemplo: Maior Divisor Comum

## Unidade de Processamento Dedicada



## Unidade de Processamento Genérica



## ■ Exemplo: Maior Divisor Comum

### Descrição em linguagem RTL:

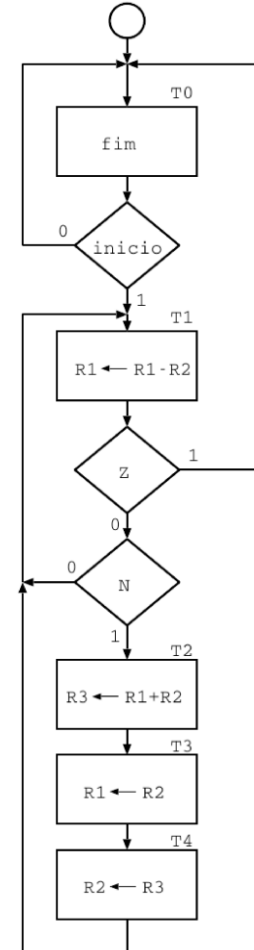
$T0: fim \leftarrow 1$   
 $T0.inicio: T0 \leftarrow 0, T1 \leftarrow 1$   
 $T1: R1 \leftarrow R1 - R2, fim \leftarrow 0$   
 $T1.Z: T1 \leftarrow 0, T0 \leftarrow 1$   
 $T1.N: T1 \leftarrow 0, T2 \leftarrow 1$   
 $T2: R3 \leftarrow R1 + R2, T2 \leftarrow 0, T3 \leftarrow 1, fim \leftarrow 0$   
 $T3: R1 \leftarrow R2, T3 \leftarrow 0, T4 \leftarrow 1, fim \leftarrow 0$   
 $T4: R2 \leftarrow R3, T4 \leftarrow 0, T1 \leftarrow 1, fim \leftarrow 0$

**Dados de entrada:** R1, R2

**Resultado:** R2

(R3 = registo temporário)

### Fluxograma

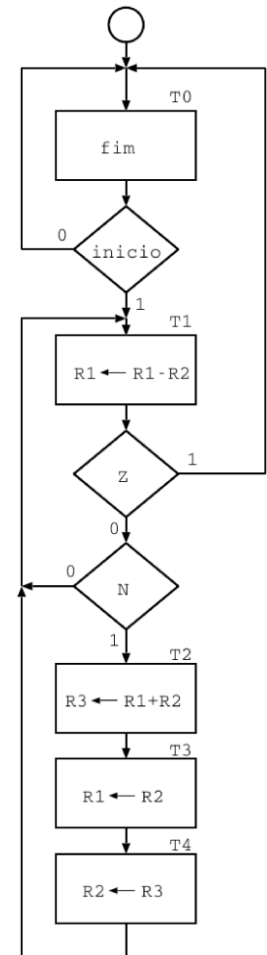


## ■ Exemplo: Maior Divisor Comum

### Microprogramação:

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES0				ES1				SEL	<i>fim</i>	operação ULA				reg A		reg B		dest	

- ▶ 5 estados → 3 bits para codificar os endereços de controlo
  - ▶ Como T1 pode ter 3 estados seguintes diferentes (um deles é incremental):
    - 2 campos para endereço seguinte ES0 e ES1
    - 2 bits para decidir entre os 3 estados seguintes possíveis
  - ▶ Condições de salto:
    - *inicio*, em T0
    - flags Z e N da ALU, em T1
    - *estado seguinte*, em T2 e T3
    - salto incondicional, em T4
- ↓
- 2 bits de controlo (Sel)
- ▶ 1 bit para saída de controlo (*fim*)



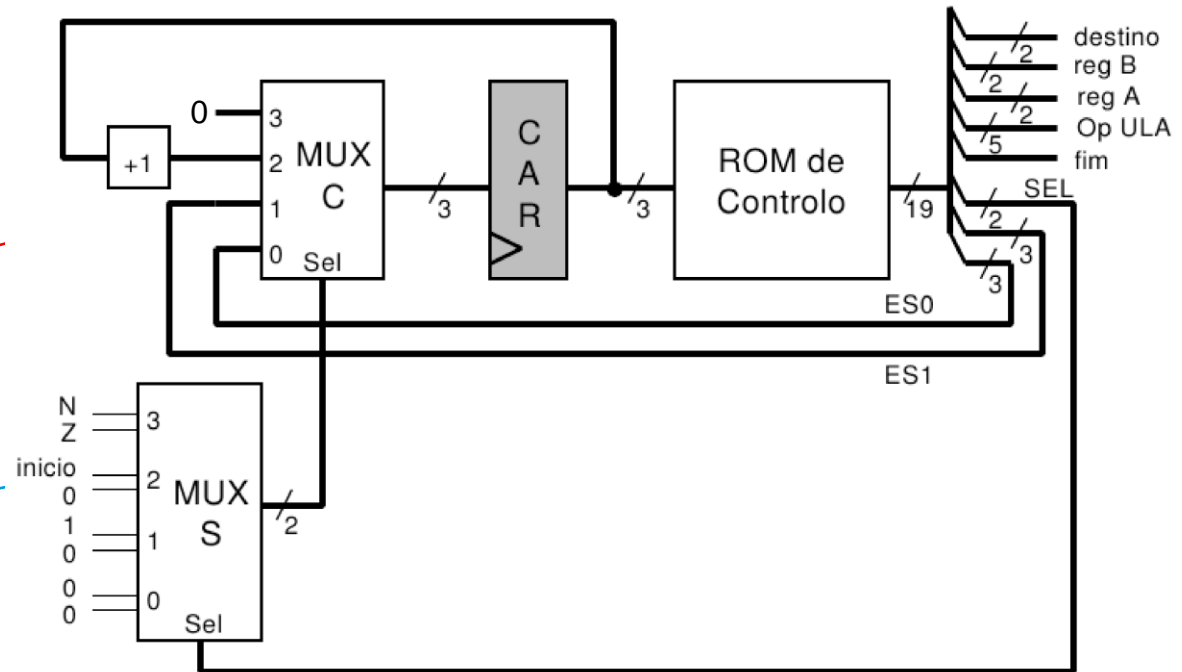
## ■ Exemplo: Maior Divisor Comum

### Unidade de Controlo:

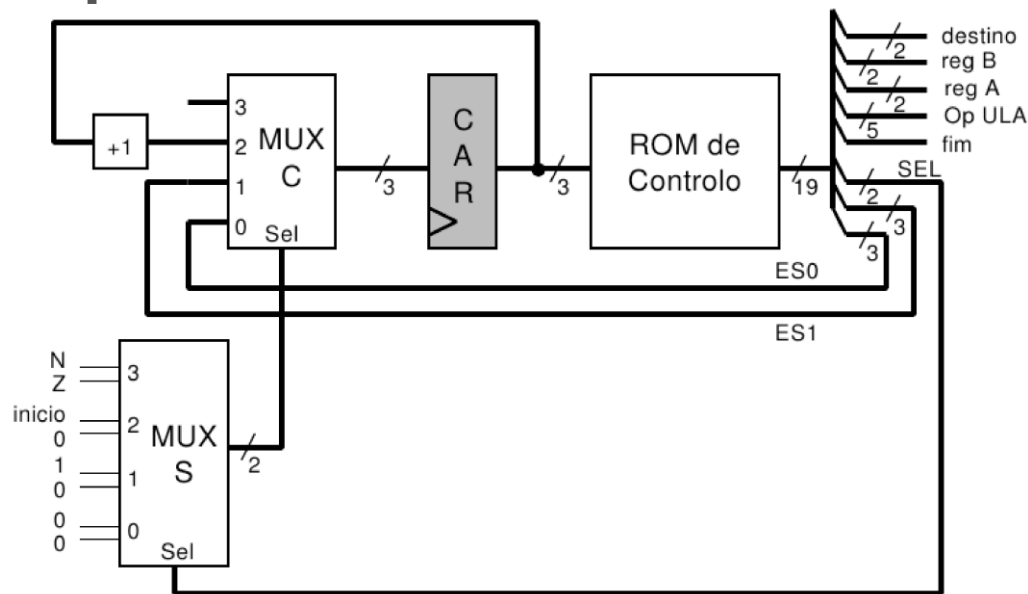
**Control Address Register (CAR)**

### MUX S:

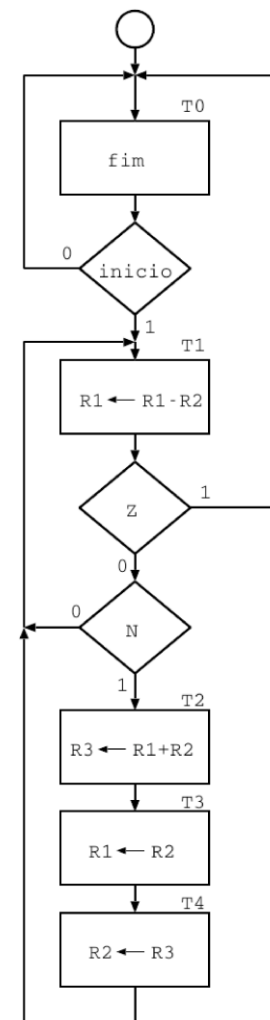
- Sel = 00  $\Rightarrow$  CAR  $\leftarrow$  ES0
- Sel = 01  $\Rightarrow$  CAR  $\leftarrow$  CAR + 1
- Sel = 10  $\Rightarrow$  CAR  $\leftarrow$  ES0, se início = 0  
CAR  $\leftarrow$  CAR + 1, se início = 1
- Sel = 11  $\Rightarrow$  CAR  $\leftarrow$  ES0, se NZ = 00  
CAR  $\leftarrow$  ES1, se NZ = 01  
CAR  $\leftarrow$  CAR + 1, se NZ = 10



## ■ Exemplo: Maior Divisor Comum



End.	ES0	ES1	SEL	fim	Op. ULA	reg. A	reg. B	reg-dest.
0	000	XXX	10	1	XXXXXX	XX	XX	XX
1	001	000	11	0	00001	01	10	01
2	XXX	XXX	01	0	00000	01	10	11
3	XXX	XXX	01	0	11000	10	XX	01
4	001	XXX	00	0	11000	11	XX	10





## ■ Exemplo: Maior Divisor Comum

### ► Especificação da ROM:

End.	ES0	ES1	SEL	fim	Op. ULA	reg. A	reg. B	reg. dest.
0	000	XXX	10	1	XXXXX	XX	XX	XX
1	001	000	11	0	00001	01	10	01
2	XXX	XXX	01	0	00000	01	10	11
3	XXX	XXX	01	0	11000	10	XX	01
4	001	XXX	00	0	11000	11	XX	10

End.	ROM
0h	00000010100000000000
1h	00100011000001011001
2h	00000001000000011011
3h	00000001011000100001
4h	00100000011000110010

### ► Dimensão da ROM:

- 5 endereços, palavras de 20 bits → 100 bits

## ■ Tema da Próxima Aula:

- ▶ Lógica programável:
  - ROM
  - PLA
  - PAL
  - FPGA
- ▶ Linguagens de descrição de hardware
  - VHDL

## Agradecimentos

Algumas páginas desta apresentação resultam da compilação de várias contribuições produzidas por:

- Nuno Roma
- Guilherme Arroz
- Horácio Neto
- Nuno Horta
- Pedro Tomás