

# **Oganização e Arquitetura de Computadores**

---

**Capítulo 14 e 15**  
**Unidade de Controle**

## **Parte I**

---

**Operação da Unidade de Controle**

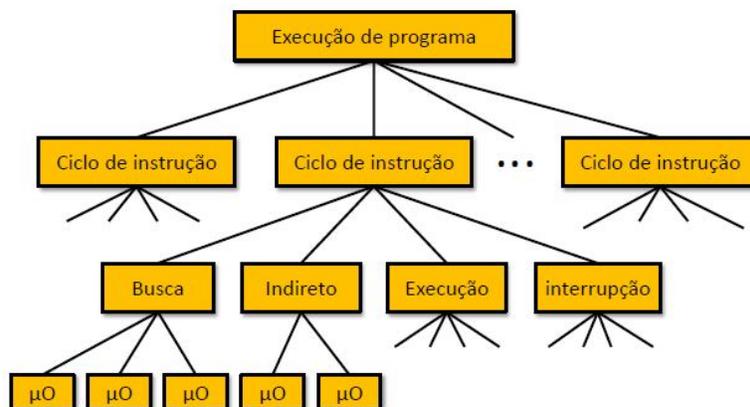
## Micro-Operações

---

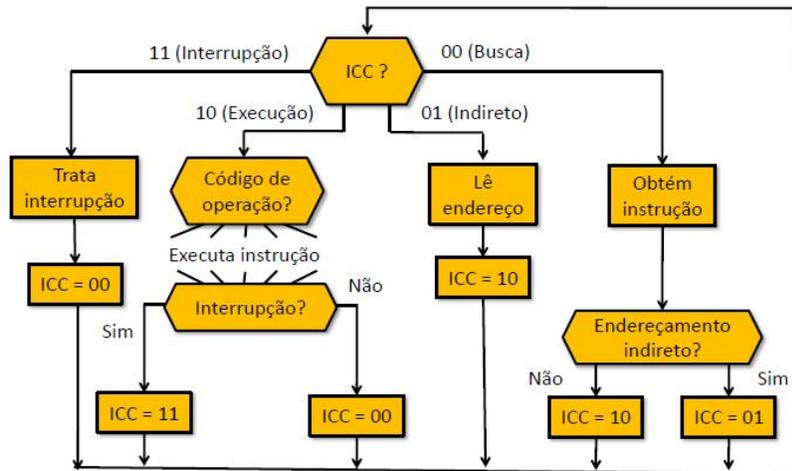
- ❑ Um computador executa um programa
- ❑ Ciclo: Busca/Executa
- ❑ Cada ciclo tem um número de passos
  - pipelining
- ❑ Passos chamados de micro-operações
- ❑ Cada passo faz muito pouco
- ❑ Operações atômicas na CPU

## Elementos constituintes da execução de um programa

---



## Ciclo de Instrução



## Busca - 4 Registradores

- ❑ Relembrando...
- ❑ Memory Address Register (MAR)
  - Conectado ao barramento de endereços
  - Especifica um endereço para ser feita uma leitura ou escrita
- ❑ Memory Buffer Register (MBR)
  - Conectado ao barramento de dados
  - Armazena dados para escrever ou últimos dados para ler
- ❑ Program Counter (PC)
  - Armazena o endereço da próxima instrução a ser executada
- ❑ Instruction Register (IR)
  - Armazena a última instrução que foi buscada

## Seqüência de busca

---

- ❑ Coloca o endereço da próxima instrução no PC
- ❑ O endereço (MAR) é colocado no barramento de endereços
- ❑ A unidade de controle envia um comando de leitura
- ❑ Resultados (dados da memória) aparecem no barramento de dados
- ❑ Dados da memória são copiados para o MBR
- ❑ O PC é incrementado de 1 (em paralelo com dados buscados da memória)
- ❑ Dados (instruções) são movidos de MBR para IR
- ❑ O MBR está livre para novas buscas de dados

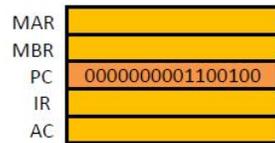
## Seqüência de busca

---

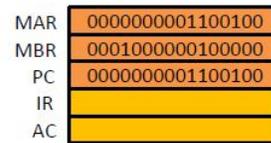
- ❑ t1: MAR  $\leftarrow$  (PC)
- ❑ t2: MBR  $\leftarrow$  (memory)
- ❑ PC  $\leftarrow$  (PC) + 1
- ❑ t3: IR  $\leftarrow$  (MBR)
- ❑ ou
- ❑ t1: MAR  $\leftarrow$  (PC)
- ❑ t2: MBR  $\leftarrow$  (memory)
- ❑ t3: PC  $\leftarrow$  (PC) + 1
- ❑ IR  $\leftarrow$  (MBR)

## Busca – Instrução na posição 0000000001100100

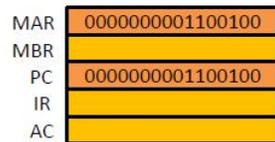
---



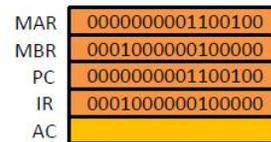
(a) Início



(c) Segundo Passo



(b) Primeiro Passo



(d) Terceiro Passo

## Ciclo indireto

---

- ❑  $MAR \leftarrow (IR_{address})$  - campo de endereço de IR
- ❑  $MBR \leftarrow (memory)$
- ❑  $IR_{address} \leftarrow (MBR_{address})$
  
- ❑ MBR contém um endereço
- ❑ IR está agora no mesmo estado como se um endereçamento direto que tenha sido usado

## Ciclo de interrupção

---

- ❑ t1:  $MBR \leftarrow (PC)$
- ❑ t2:  $MAR \leftarrow$  salva o endereço
- ❑  $PC \leftarrow$  endereço da rotina de tratamento
- ❑ t3:  $memory \leftarrow (MBR)$
- ❑ Mínimo
  - Podem existir micro-operações adicionais para pegar o endereço
  - O salvamento de contexto é feito pela rotina tratadora de interrupção, mas não pelas micro-operações

## Ciclo de execução (ADD)

---

- ❑ Ciclo complexo e imprevisível
  - Em uma máquina com N códigos de operação distintos, poderão existir N seqüência de  $\mu O$
- ❑ e.x.  $ADD R1, X$  – soma o conteúdo da posição X ao Registrador 1 e o resultado em R1
- ❑ t1:  $MAR \leftarrow (IR_{address})$
- ❑ t2:  $MBR \leftarrow (memory)$
- ❑ t3:  $R1 \leftarrow R1 + (MBR)$
- ❑ Não existe a sobreposição de micro-operações

## Ciclo de execução (ISZ)

---

- ISZ X – incremente e salte se 0
  - t1:  $MAR \leftarrow (IR_{address})$
  - t2:  $MBR \leftarrow (memory)$
  - t3:  $MBR \leftarrow (MBR) + 1$
  - t4:  $memory \leftarrow (MBR)$
  - if  $(MBR) == 0$  then  $PC \leftarrow (PC) + 1$
- Notas:
  - Se for uma simples micro-operação
  - Micro-operações feitas durante t4

## Ciclo de execução (BSA)

---

- BSA X – Salte e salve endereço
  - O endereço da instrução seguinte ao BSA é salvo em X
  - Execução continua a partir de X+1
  - t1:  $MAR \leftarrow (IR_{address})$
  - $MBR \leftarrow (PC)$
  - t2:  $PC \leftarrow (IR_{address})$
  - $memory \leftarrow (MBR)$
  - t3:  $PC \leftarrow (PC) + 1$

## **Tipos de microoperações**

---

- ❑ Transferência de dados entre registradores
- ❑ Transferência de dados externa para registradores
- ❑ Transferência de dados dos registradores para o exterior
- ❑ Executar operações lógicas e aritméticas

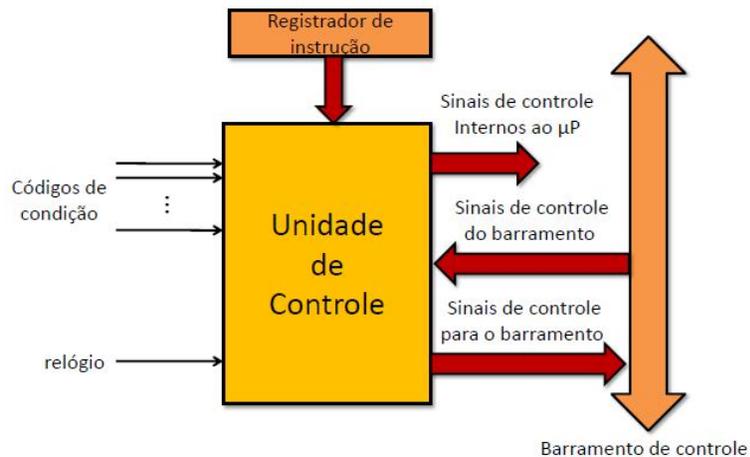
## **Funções da unidade de controle**

---

- ❑ Seqüenciamento
  - Faz com que a CPU execute todos os passos relativos às micro-operações
- ❑ Execução
  - Determina a performance de cada micro-operação
- ❑ É feita através de **sinais de controle**

## Sinais de controle

---



## Sinais de controle - entrada

---

- ❑ Clock
  - Uma microoperação (ou conjunto de micro-instruções paralelas) por ciclo de clock
- ❑ Registrador de instruções
  - Código de operação da instrução corrente
  - Determina quais as micro-instruções são realizadas
- ❑ Flags
  - Estado da CPU
  - Resultados das operações prévias
- ❑ Do barramento de controle
  - Interrupções
  - Sinais de confirmação e resposta

## Sinais de controle - saída

---

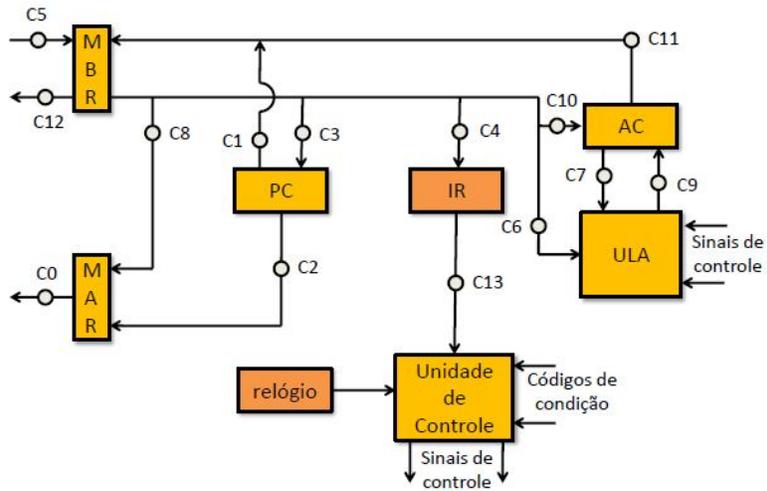
- ❑ Dentro da CPU
  - Causam movimento de dados
  - Ativam funções específicas
  
- ❑ Via controle do barramento
  - Para a memória
  - Para os módulos de I/O

## Exemplo de Seqüência do Sinal de Controle - Busca

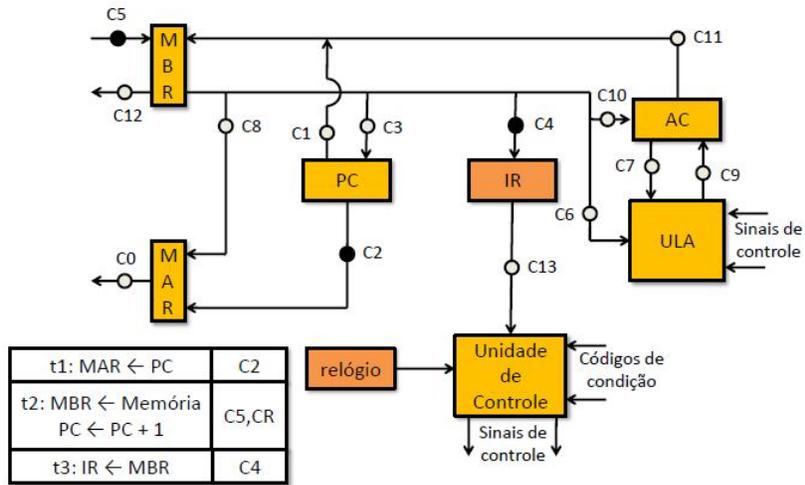
---

- ❑ MAR ← (PC)
  - A unidade de controle ativa um sinal para abrir as portas entre o PC e o MAR
  
- ❑ MBR ← (memória)
  - Portas abertas entre o MAR e o barramento de endereços
  - Sinal de controle de **leitura** da memória
  - Portas abertas entre o barramento de dados e MBR

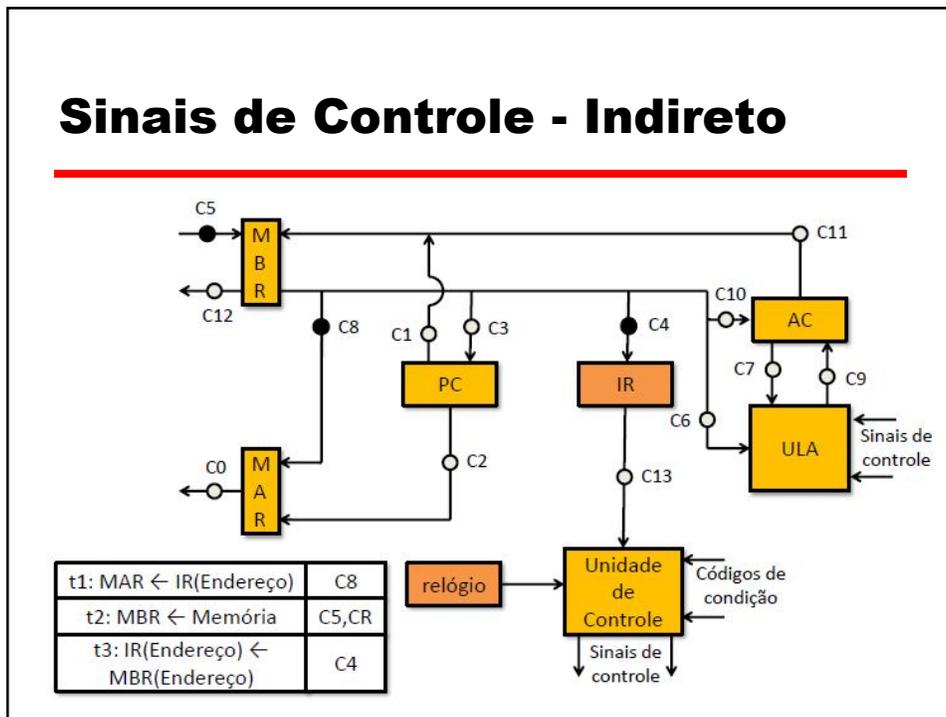
## Caminhos de dados e sinais de controle



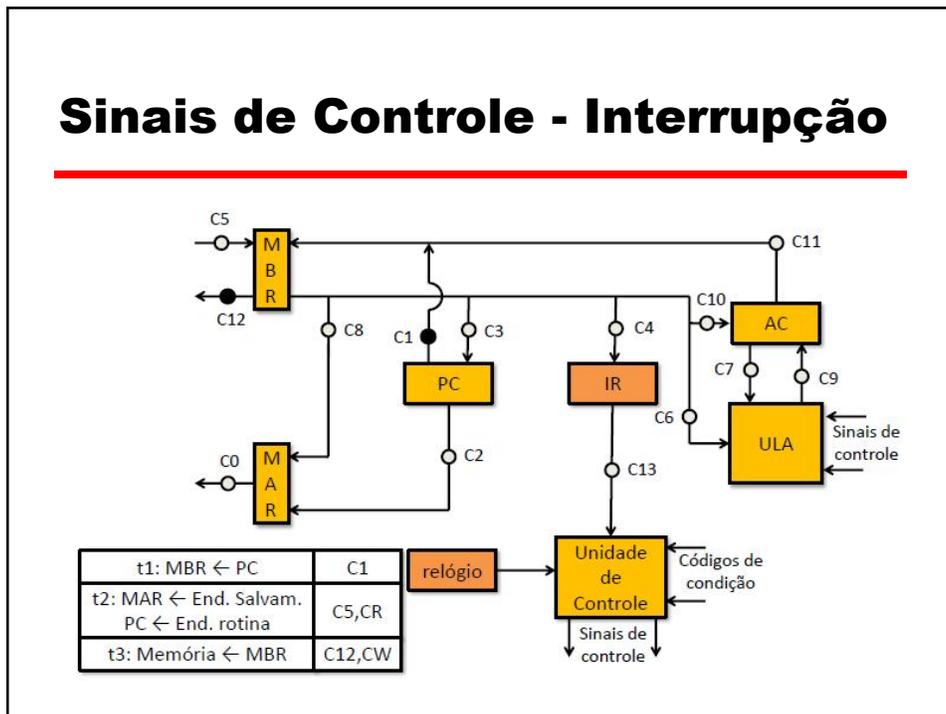
## Sinais de Controle - Busca



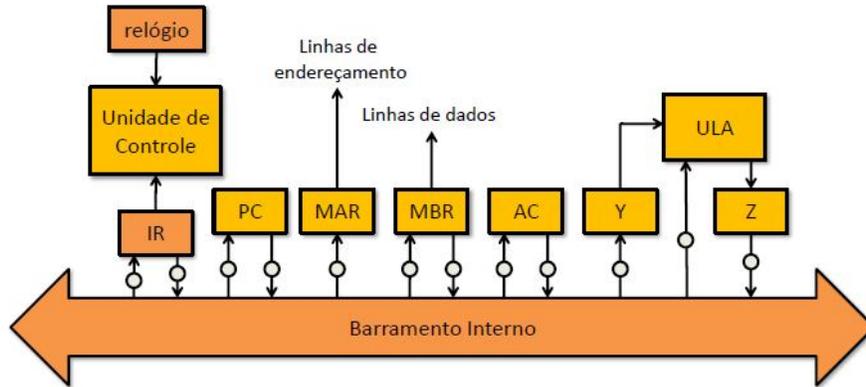
## Sinais de Controle - Indireto



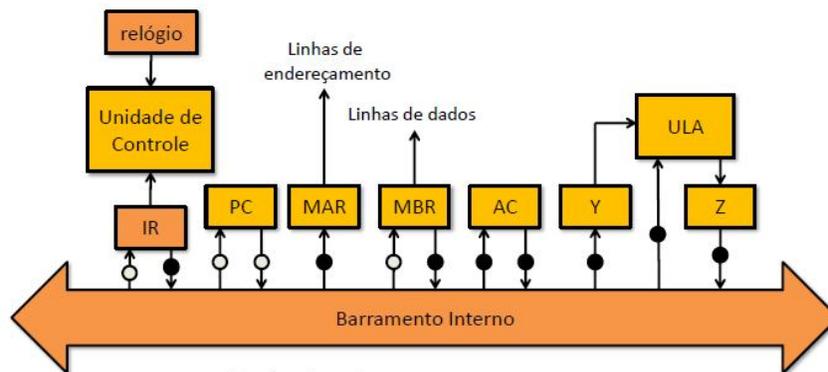
## Sinais de Controle - Interrupção



## Organização interna - barramento



## Organização interna - barramento



- t1: MAR  $\leftarrow$  IR(Endereço)
- t2: MBR  $\leftarrow$  Memória
- t3: Y  $\leftarrow$  MBR
- t4: Z  $\leftarrow$  AC + Y
- t5: AC  $\leftarrow$  Z

## **Parte II**

---

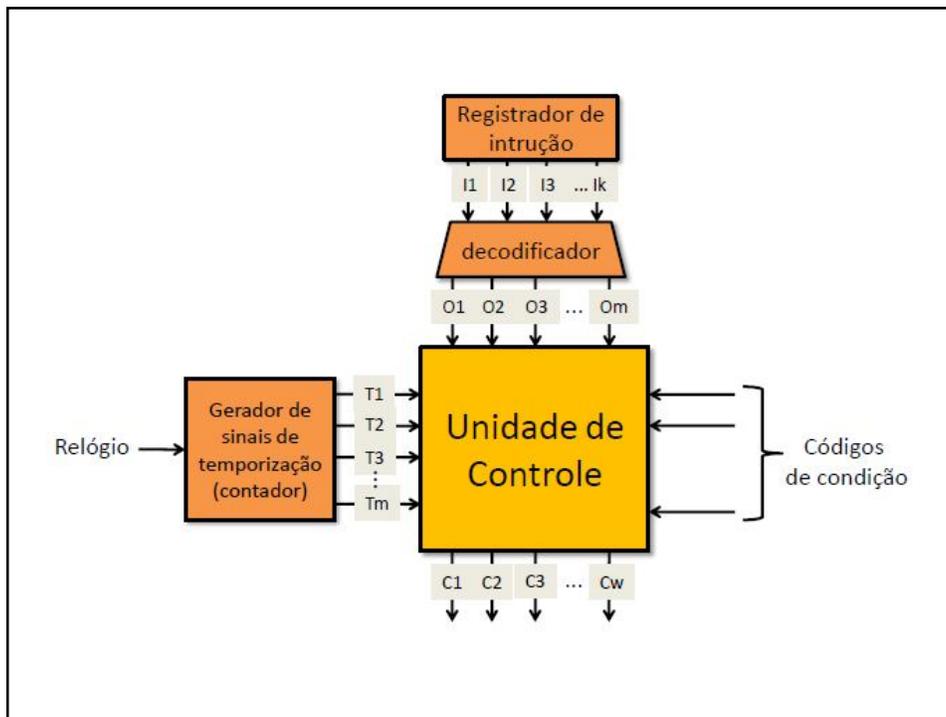
### **Implementações da Unidade de Controle**

- Implementação Hardwired
- Implementação Microprogramada

### **Implementação Hardwired**

---

- ❑ Basicamente um circuito combinatório
- ❑ Entradas da unidade de controle se transformam em sinais de controle
- ❑ Sinais:
  - Flags e barramento de controle
  - Registrador de Instruções
  - Clock



## Lógica da UC

- ❑ Cada sinal: expressão booleana
- ❑ Exemplo: **C5**

t1: MAR ← (PC) C<sub>2</sub>  
 t2: MBR ← Memória C<sub>5</sub>, C<sub>R</sub>  
 PC ← (PC) + 1  
 t3: IR ← (MBR) C<sub>4</sub>

PQ = 00    Ciclo de busca  
 PQ = 01    Ciclo indireto  
 PQ = 10    Ciclo de execução  
 PQ = 11    Ciclo de interrupção

$$C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + \bar{P} \cdot Q \cdot T_2$$

## **Problemas com Projeto Hardwired**

---

- ❑ Seqüenciamento complexo e microoperações lógicas
- ❑ Difícil de projetar e testar
- ❑ Projeto inflexível
- ❑ Difícil de agregar novas instruções

## **Implementação Microprogramada**

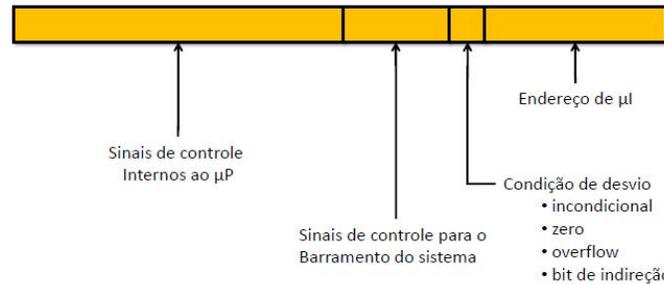
---

- ❑ Proposta por wilkes, 1950
- ❑ Usado pela primeira vez no System 360 da IBM
- ❑ Objetivo:
  - Simplificação do projeto de CPU
- ❑ Utilizado principalmente nas máquinas CISC
- ❑ Idéia:
  - Execução de microprogramas dentro da CPU



## Microinstrução Horizontal

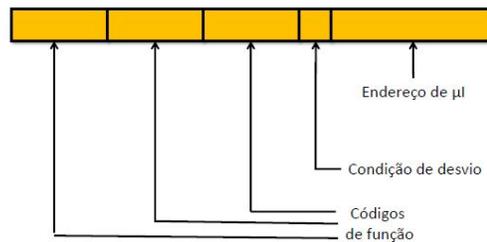
---



- 1 bit para cada linha interna do processador
- 1 bit para cada linha de controle do barramento do sistema
- Campo de condição indica a condição em que o desvio para o endereço da microinstrução deve ser tomado

## Microinstrução

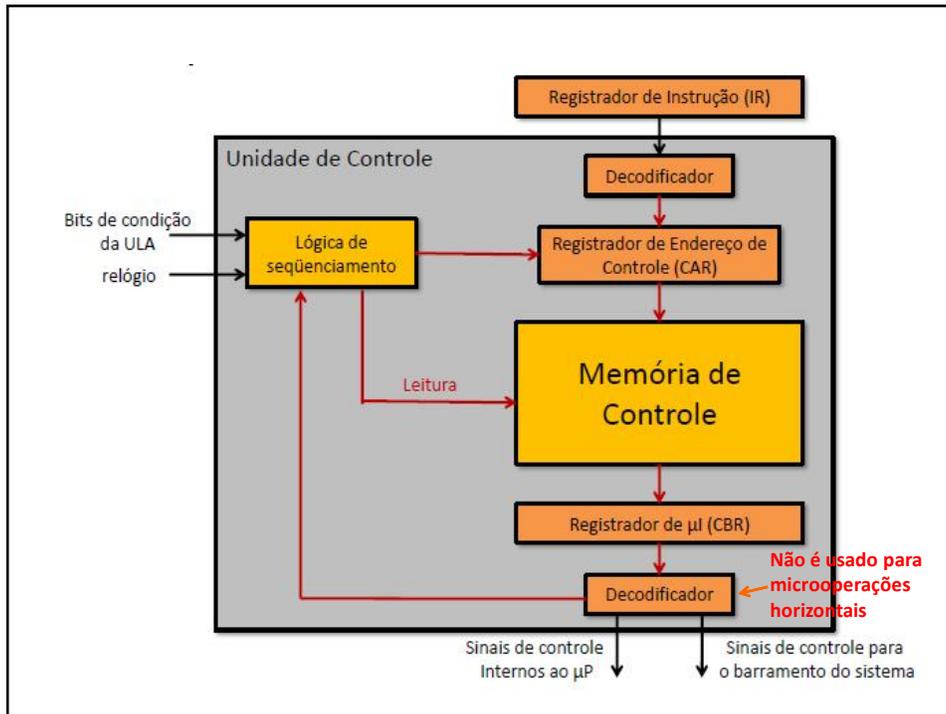
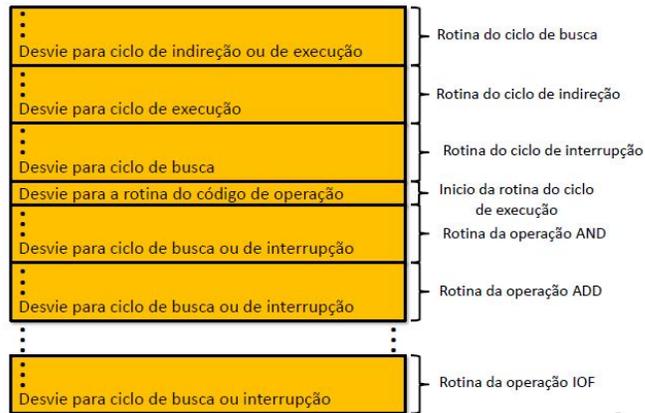
---



- Um código para cada ação a ser efetuada (p. ex.  $MAR \leftarrow (PC)$ )
- O código é decodificado e é traduzido em sinais de controle individuais

# Memória de Controle

- Memória onde as palavras de controle são armazenadas



## Execução de MicroInstruções

---

- ❑ Unidade de sequenciamento envia um comando de leitura para a memória de controle
- ❑ Palavra cujo endereço é especificado no registrador de endereço é lida para o registrador de microinstrução
- ❑ O conteúdo deste registrador gera os sinais de controle e a informação sobre o próximo endereço para a unidade de sequenciamento
- ❑ A unidade de lógica de sequenciamento carrega o novo endereço no registrador de endereço de controle
  
- ❑ Tudo isso ocorre em um ciclo de clock

## Vantagens e Desvantagens

---

- ❑ Vantagens:
  - Simplificação do projeto da UC
  - Testes e Correções mais simples
  - Possibilidade de acrescentar novas instruções
- ❑ Desvantagem:
  - Velocidade

## **Exercícios**

---

- ❑ 14.1
- ❑ 14.2
- ❑ 14.3
- ❑ 14.4
- ❑ 15.3
- ❑ 15.4