
Unidade de Controle - operação e controle microprogramado

Grupo: Daniel Carlos
Everaldo Faustino
Matheus Gameiro

Micro-operações

➤ O QUE SÃO MICRO-OPERAÇÕES?

São instruções detalhadas de baixo nível usadas em alguns projetos para acionar diversos componentes de um sistema, através de sinais de controle.

Micro-operações

➤ ESQUEMA BÁSICO DE EXECUÇÃO DE OPERAÇÃO



Micro-operações

➤ Ciclo de busca

PC: Contador de Programa

MAR: Registrador de endereço memória

MBR: Registrador de buffer de memória

IR: Registrador de instrução

t_1 : $MAR \leftarrow (PC)$
 t_2 : $MBR \leftarrow \text{Memória}$
 $PC \leftarrow (PC) + I$
 t_3 : $IR \leftarrow (MBR)$

MAR	
MBR	
PC	0000000001100100
IR	
AC	

(a) Início (antes de t_1)

MAR	0000000001100100
MBR	
PC	0000000001100100
IR	
AC	

(b) Depois do primeiro passo

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	
AC	

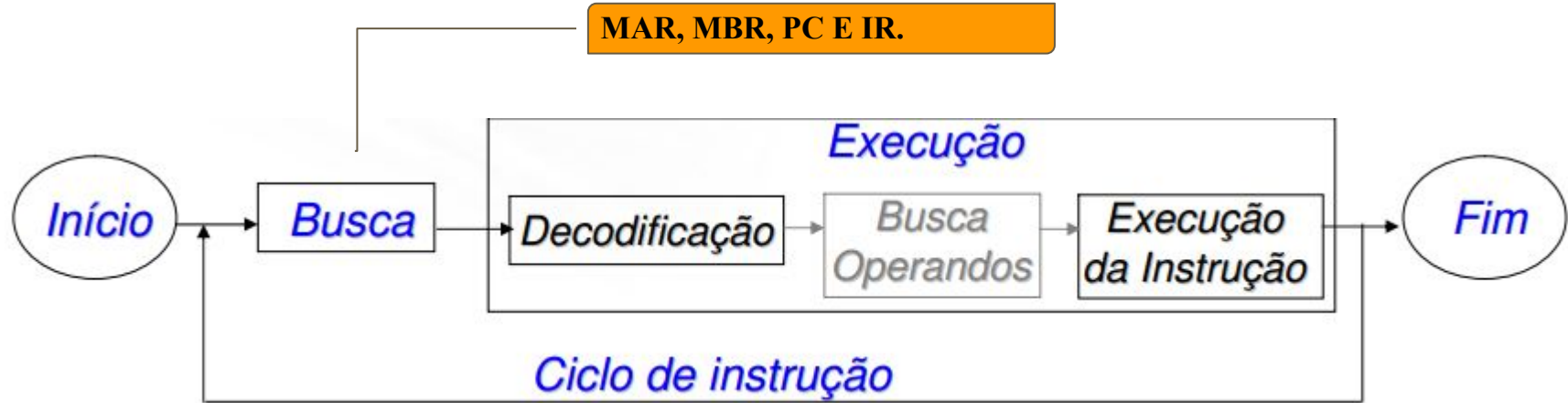
(c) Depois do segundo passo

MAR	0000000001100100
MBR	0001000000100000
PC	0000000001100101
IR	0001000000100000
AC	

(d) Depois do terceiro passo

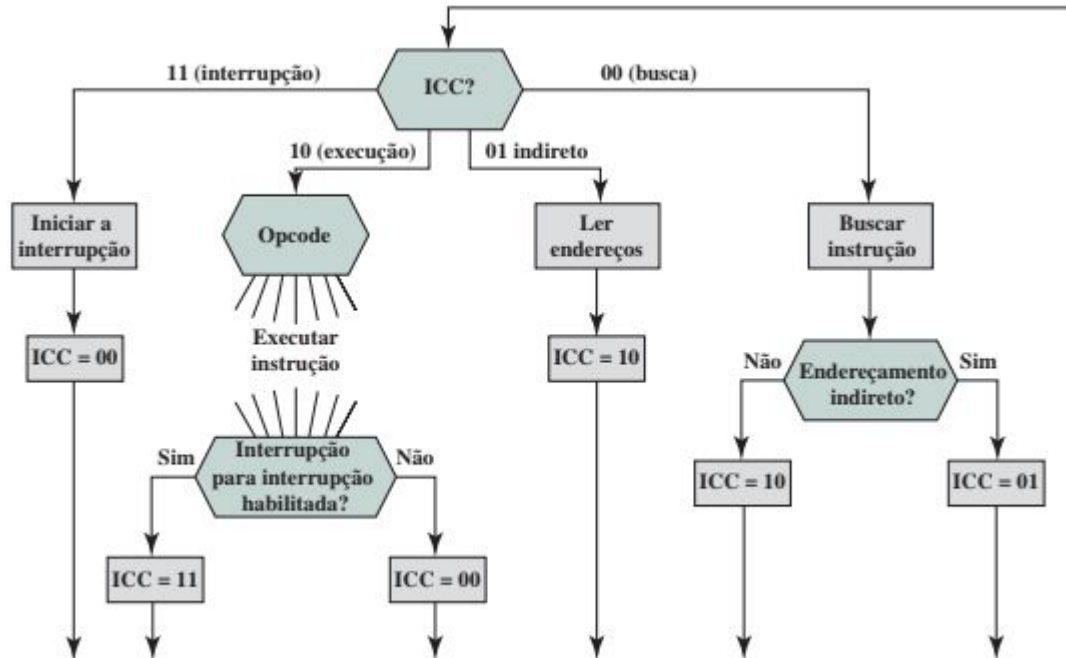
Micro-operações

➤ ESQUEMA BÁSICO DO CICLO DE INSTRUÇÃO - busca



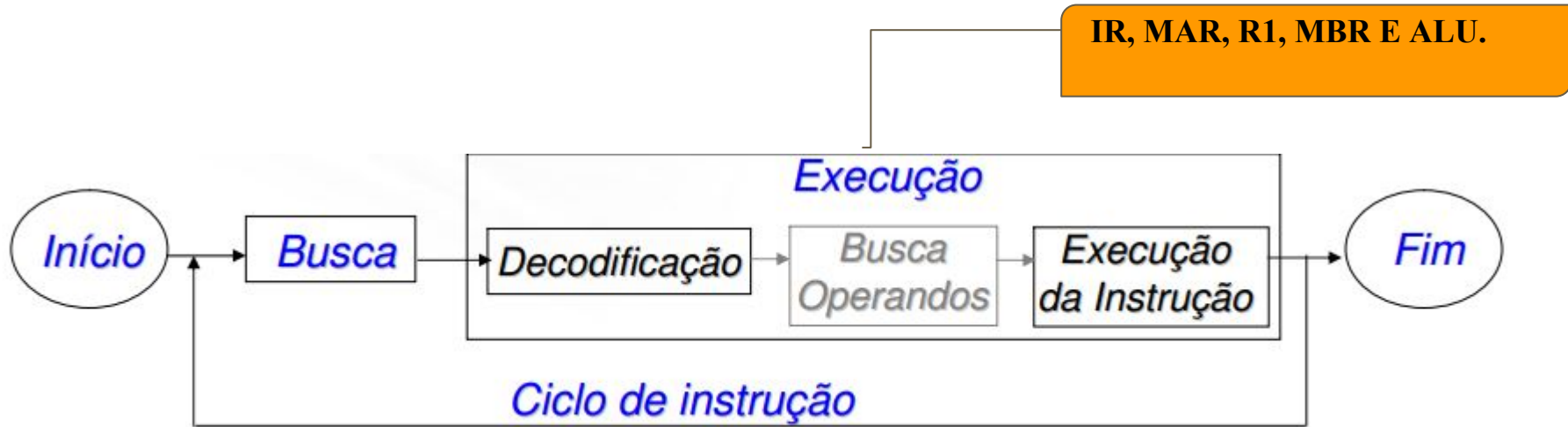
Micro-operações

➤ FLUXOGRAMA DETALHADO



Micro-operações

➤ ESQUEMA BÁSICO DO CICLO DE INSTRUÇÃO - execução



Controle do processador

➤ REQUISITOS FUNCIONAIS

São funções capazes de definir o que exatamente a unidade de controle deve fazer acontecer. Uma definição desses requisitos funcionais é a base para o projeto e a implementação da unidade de controle.

1. ALU.
2. Registradores.
3. Caminhos de dados internos.
4. Caminhos de dados externos.
5. Unidade de controle.

Controle do processador

➤ REQUISITOS FUNCIONAIS

Sequenciamento: a unidade de controle faz com que o processador siga uma série de micro-operações na sequência correta, com base no programa que está sendo executado.

Execução: a unidade de controle faz cada micro-operação ser executada.

Controle do processador

➤ SINAIS DE CONTROLE

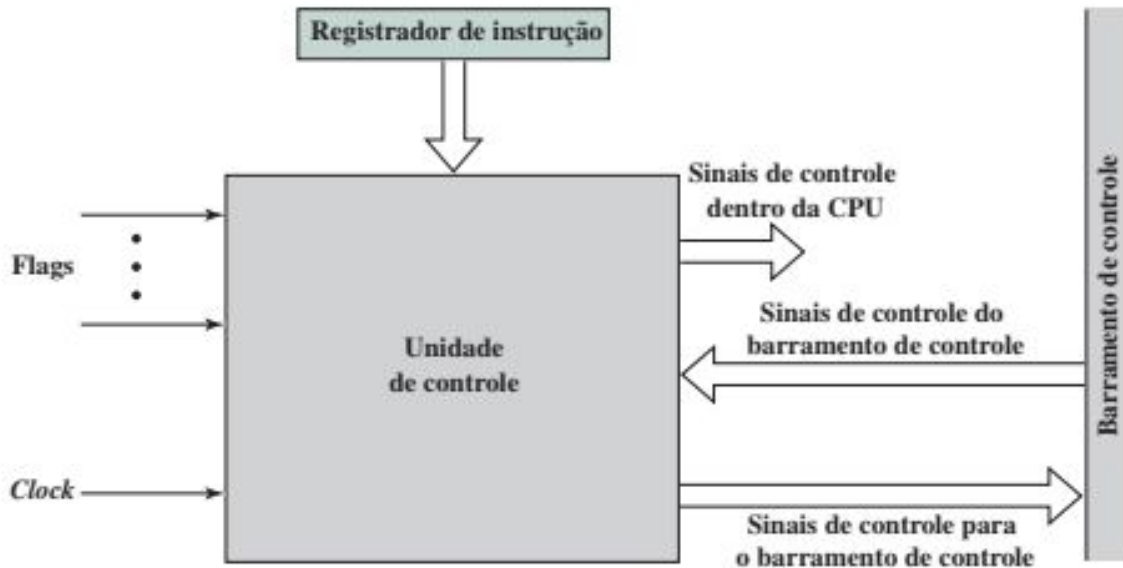
Podemos macro definir como três tipos: ativador de função, ativador do caminho de dados e barramento externo.

ONDE:

- Clock
- Flags
- Sinais do barramento de controle
- Sinais dentro da CPU
- Sinais para o barramento de controle

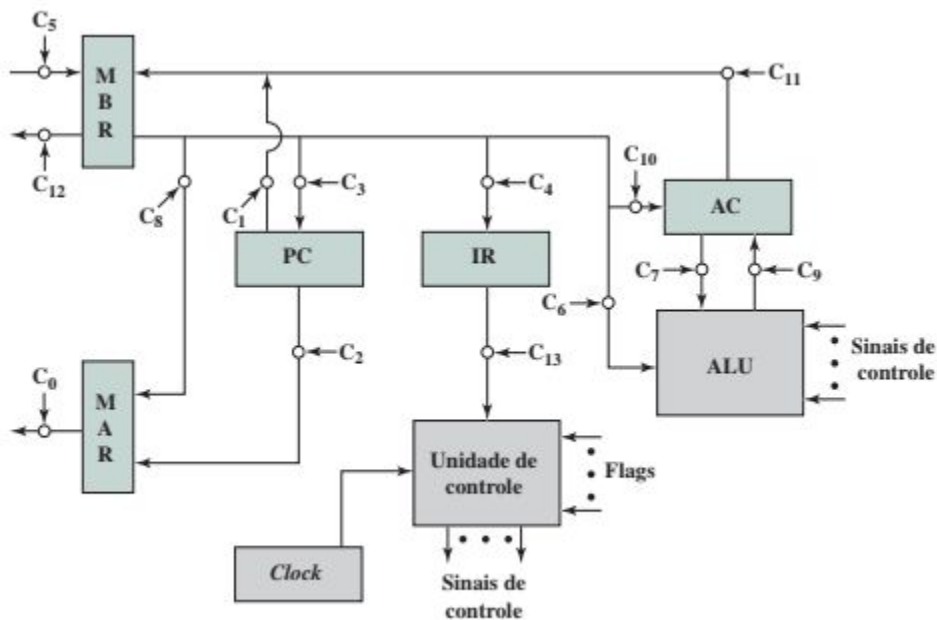
Controle do processador

➤ SINAIS DE CONTROLE



Controle do processador

➤ SINAIS DE CONTROLE



Controle do processador

➤ MICRO-OPERAÇÕES E SINAIS DE CONTROLE

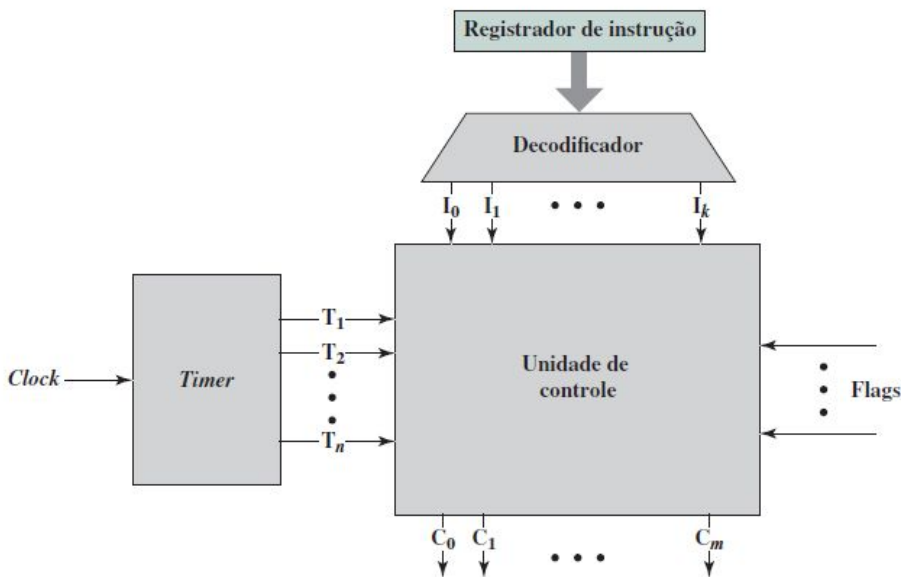
	Micro-operações	Sinais de controle ativos
Busca:	$t_1: MAR \leftarrow (PC)$	C_2
	$t_2: MBR \leftarrow \text{Memória}$ $PC \leftarrow (PC) + 1$	C_5, C_R
	$t_3: IR \leftarrow (MBR)$	C_4
Indireto:	$t_1: MAR \leftarrow (IR(\text{Endereço}))$	C_8
	$t_2: MBR \leftarrow \text{Memória}$	C_5, C_R
	$t_3: IR(\text{Endereço}) \leftarrow (MBR(\text{Endereço}))$	C_4
Interrupção:	$t_1: MBR \leftarrow (PC)$	C_1
	$t_2: MAR \leftarrow \text{Endereço-salvar}$ $PC \leftarrow \text{Endereço-rotina}$	
	$t_3: \text{Memória} \leftarrow (MBR)$	C_{12}, C_W

C_R = Sinal de controle de leitura para o barramento do sistema.

C_W = Sinal de controle de escrita para o barramento do sistema.

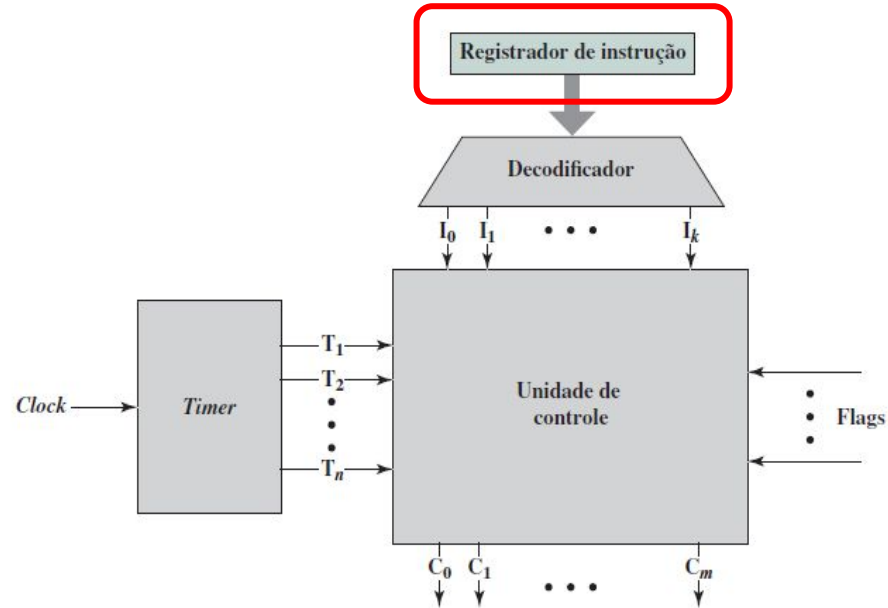
Implementação em Hardware

- Nesta implementação, a unidade de controle é um circuito combinatório. Seus sinais lógicos de entrada são transformados em um conjunto de sinais lógicos de saída, que são os sinais de controle.



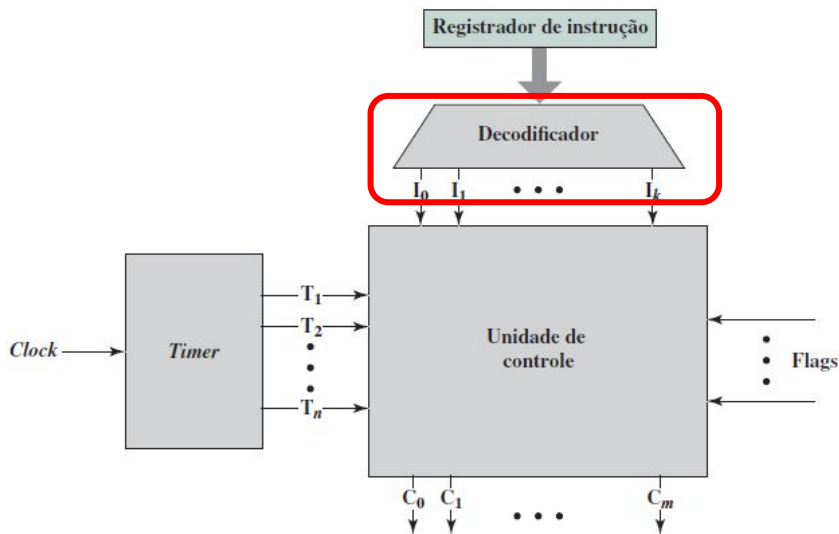
Registrador de Instrução

- A unidade de controle faz uso do opcode e vai efetuar diferentes ações (gera uma combinação diferente de sinais de controle) para instruções diferentes.



Decodificador

- Para simplificar a lógica da unidade de controle, deveria haver uma única entrada lógica para cada opcode. Essa função pode ser executada por um decodificador, o qual recebe uma entrada codificada e produz uma única saída.



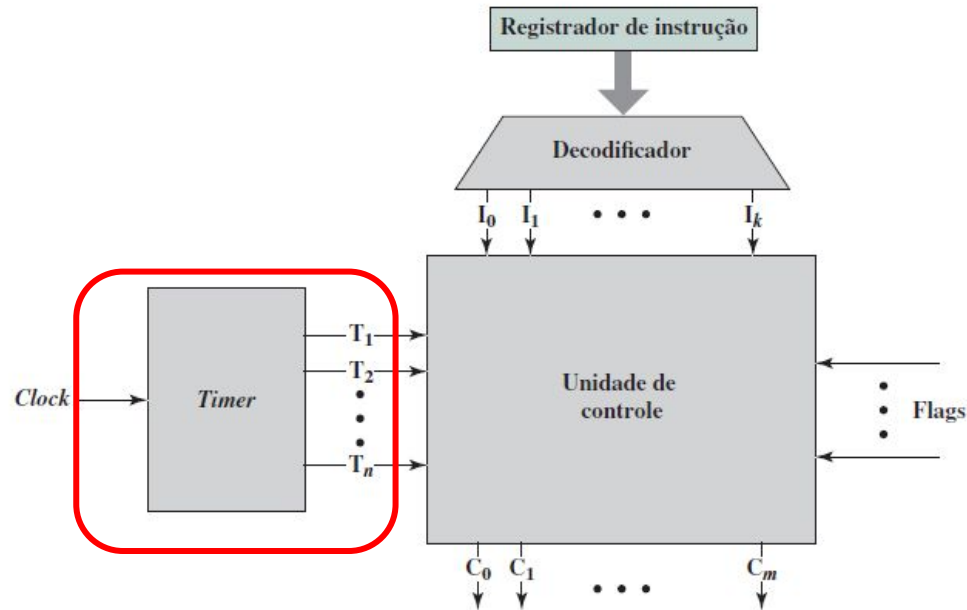
Decodificador

- Um decodificador com quatro entradas e 16 saídas.

[illegible]

Clock

- A unidade de controle envia diferentes sinais de controle em unidades diferentes de tempo dentro de um mesmo ciclo de instrução.



Exemplo de Unidade de Controle

- A UC gera uma sequência de sinais de controle, que causam a execução das microoperações. A tabela indica os sinais de controle requeridas para algumas execução das microoperações

	Micro-operações	Sinais de controle ativos
Busca:	$t_1: \text{MAR} \leftarrow (\text{PC})$	C_2
	$t_2: \text{MBR} \leftarrow \text{Memória}$ $\text{PC} \leftarrow (\text{PC}) + 1$	C_5, C_R
	$t_3: \text{IR} \leftarrow (\text{MBR})$	C_4
Indireto:	$t_1: \text{MAR} \leftarrow (\text{IR}(\text{Endereço}))$	C_8
	$t_2: \text{MBR} \leftarrow \text{Memória}$	C_5, C_R
	$t_3: \text{IR}(\text{Endereço}) \leftarrow (\text{MBR}(\text{Endereço}))$	C_4
Interrupção:	$t_1: \text{MBR} \leftarrow (\text{PC})$	C_1
	$t_2: \text{MAR} \leftarrow \text{Endereço-salvar}$ $\text{PC} \leftarrow \text{Endereço-rotina}$	
	$t_3: \text{Memória} \leftarrow (\text{MBR})$	C_{12}, C_W

C_R = Sinal de controle de leitura para o barramento do sistema.

C_W = Sinal de controle de escrita para o barramento do sistema.

Exemplo de Unidade de Controle

- Considere o sinal de controle C5 (que faz com que um dado seja lido do barramento externo para o REM).
- Vamos definir dois sinais P e Q que possui a seguinte interpretação.

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

- O sinal de controle C5 é ativado durante o segundo intervalo de tempo t2, tanto no ciclo de busca quanto no de busca do operando(endereçamento indireto). Portanto:

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

Exemplo de Unidade de Controle

- A expressão não está completa. C5 é também necessária durante o ciclo de execução de uma instrução.
- Supondo que apenas as instruções LDA, ADD e AND efetuem leitura na memória, temos:

$$C_5 = \bar{P} \cdot \bar{Q} \cdot T_2 + \bar{P} \cdot Q \cdot T_2 + P \cdot \bar{Q} \cdot (LDA + ADD + AND) \cdot T_2$$

- Esse mesmo processo poderia ser repetido para todo sinal de controle, resultando em um conjunto de equações booleanas que definem o comportamento da UC e portanto de UCP.
- A tarefa de implementar um circuito combinatório que satisfaça todas essas equações se torna extremamente difícil.
- Uma abordagem bem mais simples, conhecida como microprogramação, normalmente é usada.

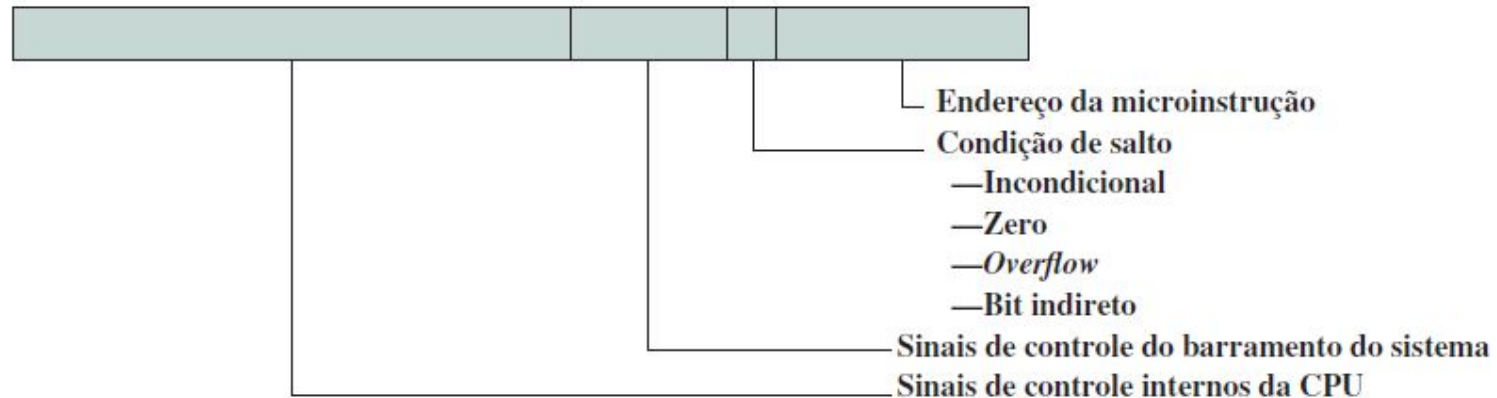
Microinstruções

- Uma alternativa usada em vários processadores CISC é implementar uma unidade de controle microprogramada.
- Uma sequência de instruções é conhecida como um microprograma, ou firmware.

Ordem	Efeito da ordem
$A\ n$	$C(Acc) + C(n)$ para Acc_1
$S\ n$	$C(Acc) - C(n)$ para Acc_1
$H\ n$	$C(n)$ para Acc_2
$V\ n$	$C(Acc_2) \times C(n)$ para Acc , onde $C(n) \geq 0$
$T\ n$	$C(Acc_1)$ para n , 0 para Acc
$U\ n$	$C(Acc_1)$ para n
$R\ n$	$C(Acc) \times 2^{(n+1)}$ para Acc
$L\ n$	$C(Acc) \times 2^{n+1}$ para Acc
$G\ n$	IF $C(Acc) < 0$, transferir o controle para n ; se $C(Acc) \geq 0$, ignorar (isto é, proceder serialmente)
$I\ n$	Ler próximo caractere do mecanismo de entrada para n
$O\ n$	Enviar $C(n)$ para mecanismo de saída

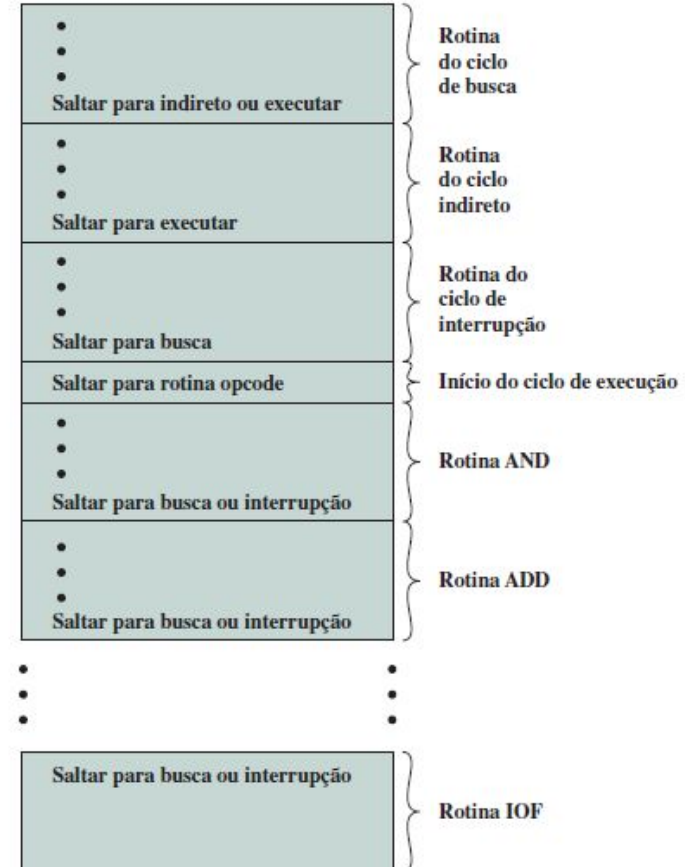
Microinstrução Horizontal

- Existe um bit para cada linha de controle interna do processador e um bit para cada linha de controle do barramento do sistema. Há um campo de condição indicando a condição em que deve haver um desvio e há um campo com o endereço da microinstrução a ser executada depois que um desvio é tomado.

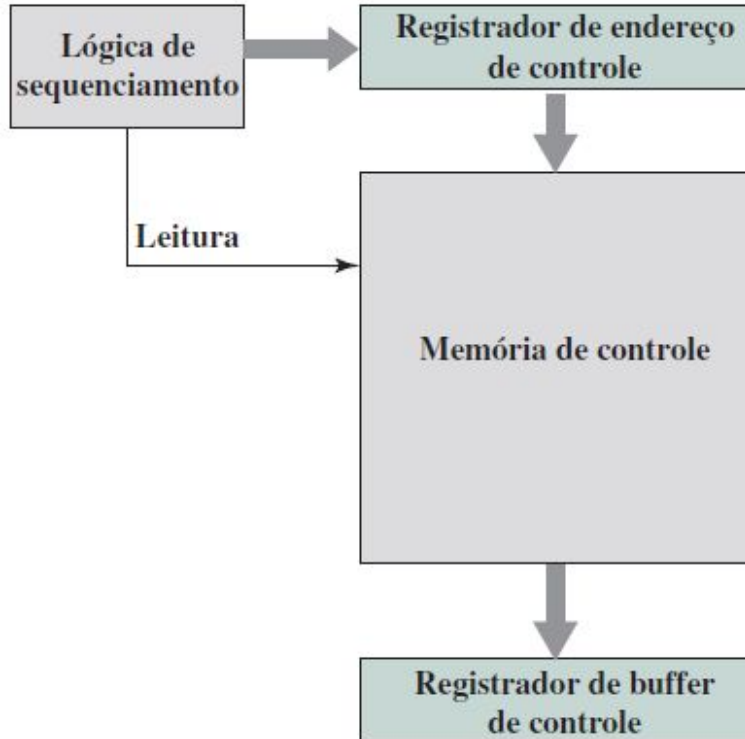


Memória de Controle

- Ela define a sequência de micro-operações para serem executadas durante cada ciclo (de busca, indireto, de execução, de interrupção) e especifica o sequenciamento desses ciclos.



Microarquitetura da Unidade de Controle.

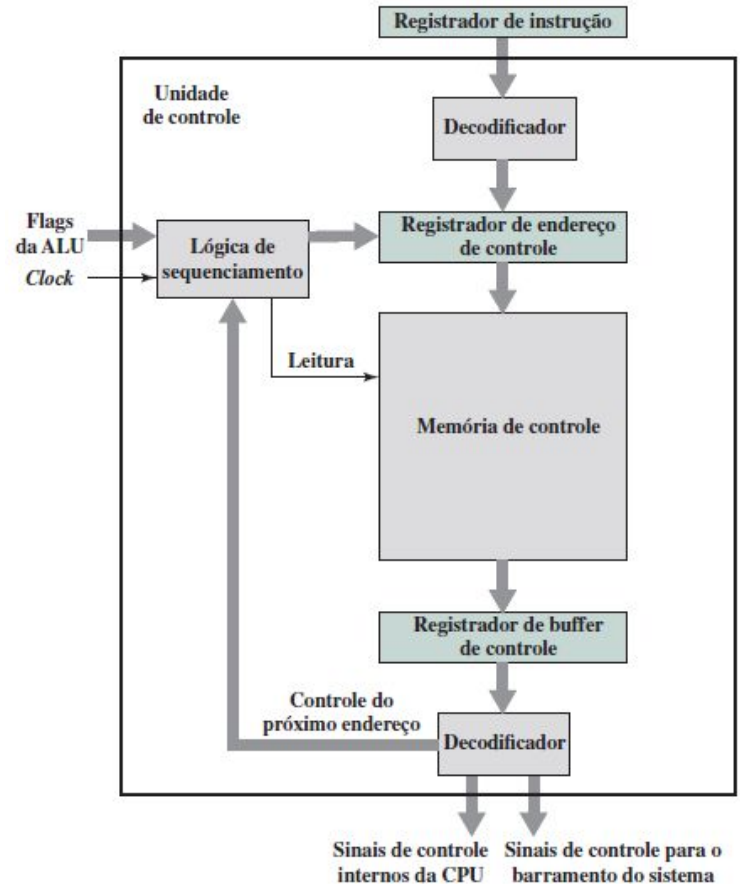


- O conjunto de microinstruções é armazenado na memória de controle. O registrador de endereço de controle contém o endereço da próxima microinstrução a ser lida.
- Quando uma microinstrução é lida a partir da memória de controle, ela é transferida para um registrador de buffer de controle.

Funcionamento da unidade de controle microprogramada

➤ A unidade de controle funciona desta forma:

- Para executar uma instrução, a unidade lógica de sequenciamento emite um comando READ para a memória de controle.
- A palavra cujo endereço é especificado no registrador de endereço de controle é lida para dentro do registrador de buffer de controle.
- O conteúdo do registrador de buffer de controle gera sinais de controle e a informação do próximo endereço para a unidade lógica de sequenciamento.
- A unidade lógica de sequenciamento carrega um novo endereço no registrador de endereço de controle com base na informação do próximo endereço.

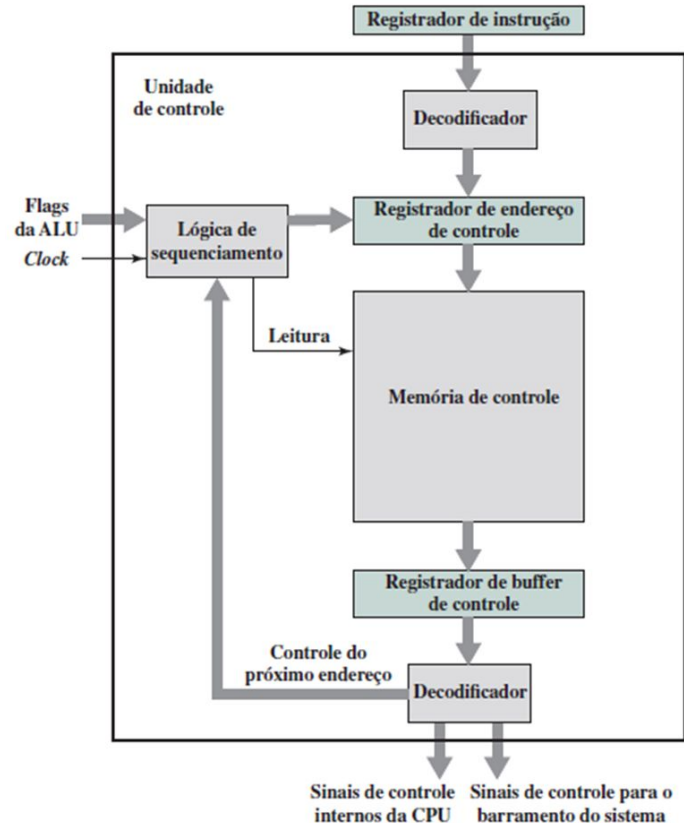
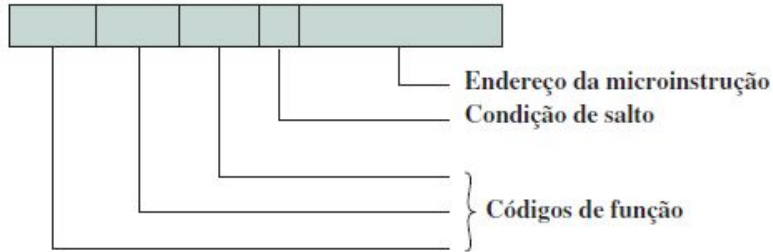


Funcionamento da unidade de controle microprogramada

- Dependendo do valor das flags da ALU e do registrador de buffer de controle, uma das três decisões é tomada:
 - Obter a próxima instrução: adiciona 1 ao registrador de endereço de controle.
 - Saltar para uma nova rotina com base em uma microinstrução de salto: carrega o campo de endereço do registrador de buffer de controle no registrador de endereço de controle.
 - Saltar para uma rotina de instrução de máquina: carrega o registrador de endereço de controle com base no opcode que está em IR.

Funcionamento da unidade de controle microprogramada

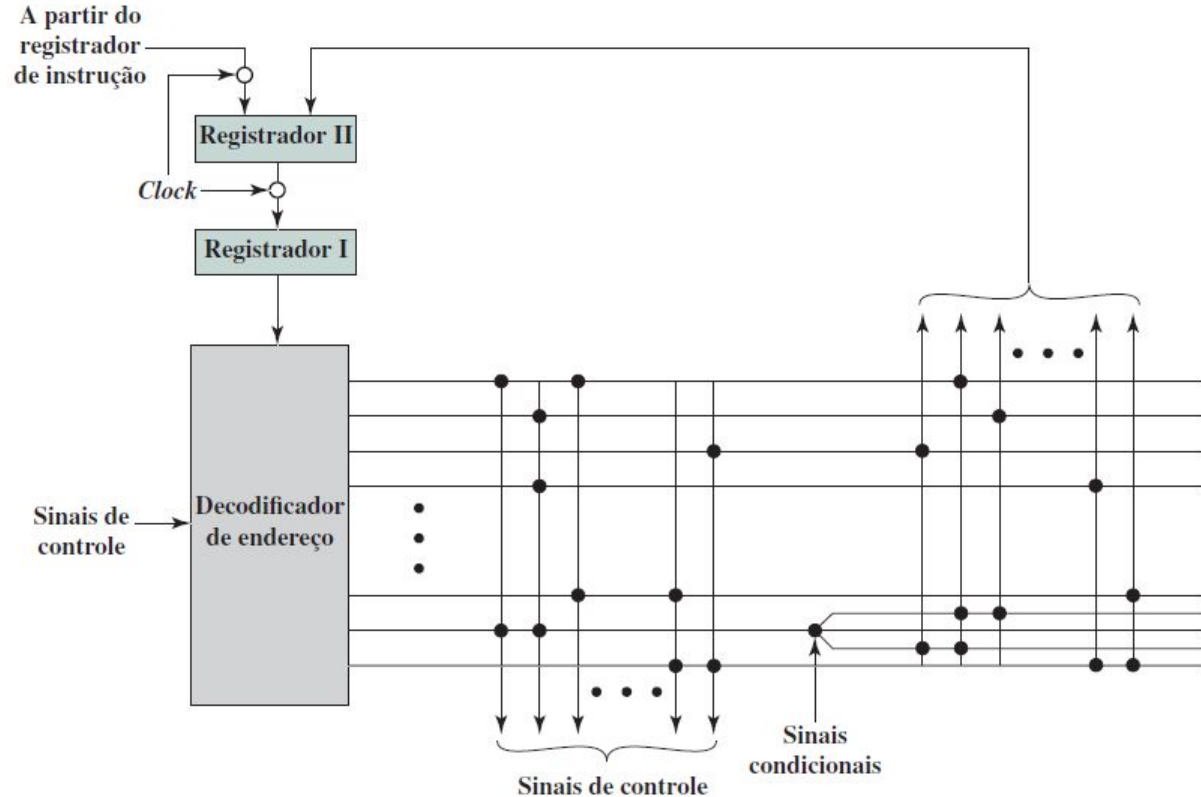
- Em uma microinstrução vertical, um código é usado para cada ação a ser efetuada, e o decodificador traduz esse código em sinais de controle individuais.



Controle de Wilkes

A	Multiplicando
B	Acumulador (metade menos significativa)
C	Acumulador (metade mais significativa)
D	Registrador de deslocamento

E	Serve tanto como um registrador de endereço de memória (MAR) quanto como um registrador de armazenamento temporário
F	Contador do programa
G	Outro registrador temporário; usado para contagem



Controle de Wilkes

		Unidade aritmética	Unidade de controle de registradores	Flip-flop condicional		Próxima microinstrução	
				Definir	Usar	0	1
	0		F para G e E			1	
	1		(G para "1") para F			2	
	2		Armazenamento para G			3	
	3		G para E			4	
	4		E para decodificar			—	
A	5	C para D				16	
S	6	C para D				17	
H	7	Armazenamento para B				0	
V	8	Armazenamento para A				27	
T	9	C para armazenamento				25	
U	10	C para armazenamento				0	
R	11	B para D	E para G			19	

Vantagens e desvantagens

- A principal vantagem do uso da microprogramação para implementar uma unidade de controle é que ela simplifica o projeto da unidade de controle. Assim a implementação fica mais barata e menos propensa a erros.
- A principal desvantagem de uma unidade microprogramada é que ela será um pouco mais lenta do que uma unidade por hardware de tecnologia comparável.

Sequenciamento de microinstruções

- Preocupações: o tamanho da microinstrução e o tempo de geração do endereço.
- Ao executar um microprograma, o endereço da próxima microinstrução a ser executada se encaixa em uma dessas categorias:
 - Determinado pelo registrador de instrução.
 - Próximo endereço sequencial.
 - Lógica de desvio.

Técnicas de sequenciamento

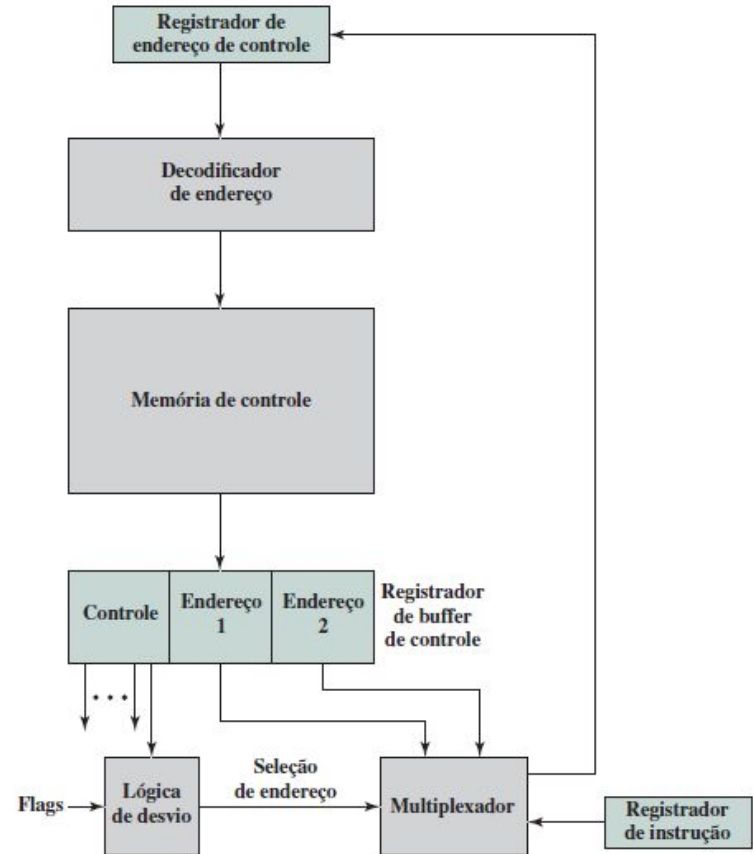
□ Dois campos de endereço.

□ Campo de endereço único.

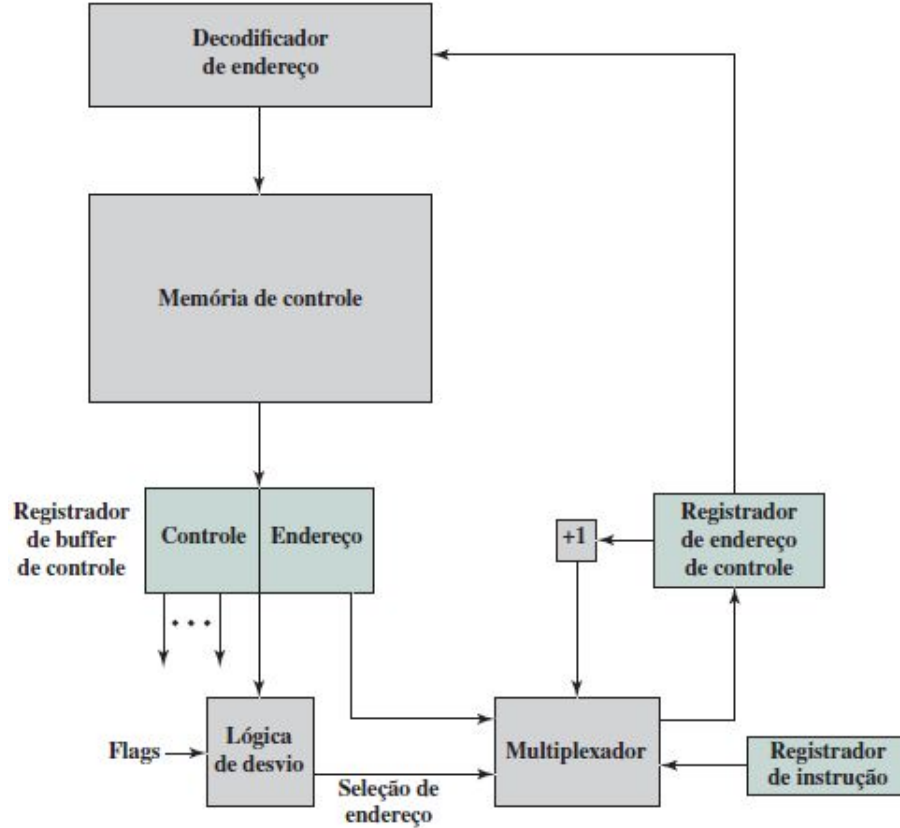
□ Formato variável.

Dois campos de endereço

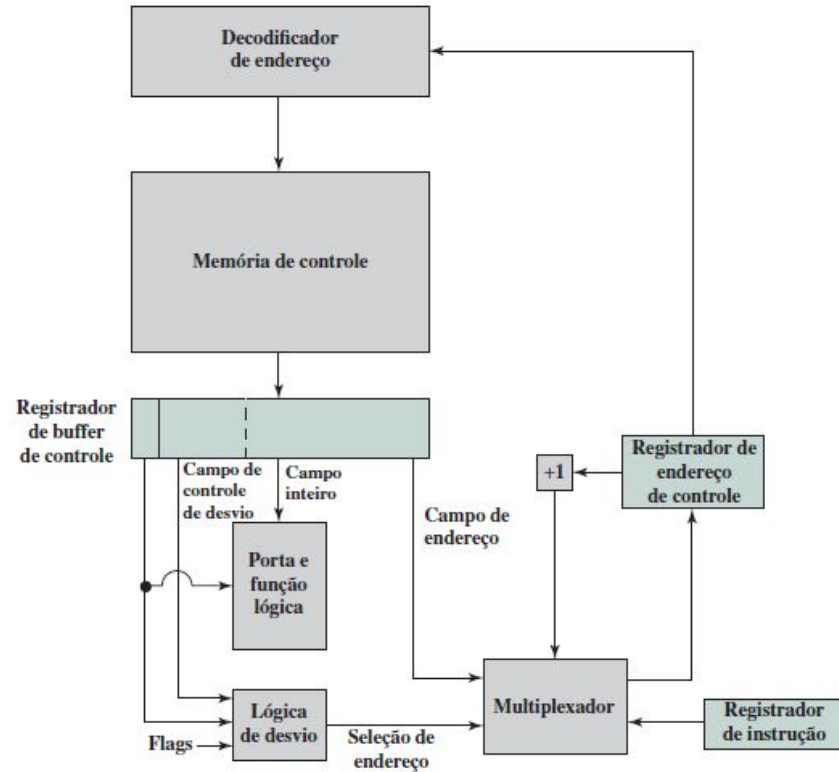
- Um multiplexador existente serve como destino para os campos de endereço e para o registrador de instrução.
- O multiplexador transmite o opcode ou um dos dois endereços para o registrador de endereço de controle.



Campo de endereço único



Formato variável



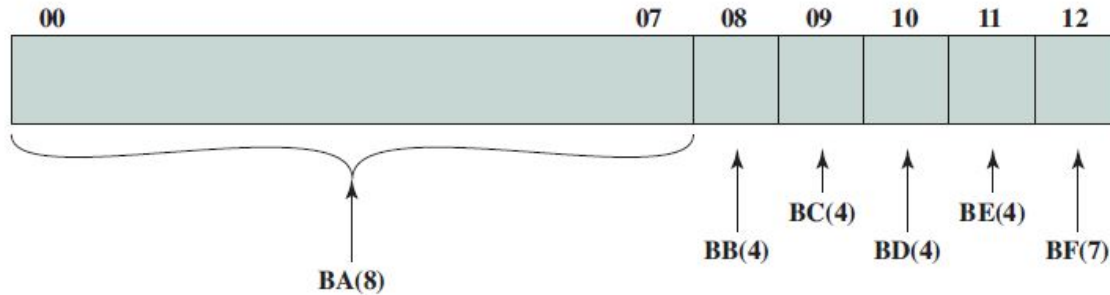
Geração de Endereços

- Podem ser divididas em técnicas explícitas, em que o endereço está disponível explicitamente na microinstrução, e técnicas implícitas, que requerem lógica adicional para gerar o endereço.
- Uma instrução de desvio condicional depende dos seguintes tipos de informação:
 - Flags da ALU.
 - Parte do opcode ou campo de modo de endereço da instrução de máquina.
 - Partes do registrador selecionado, como o bit de sinal.
 - Bits de estado dentro da unidade de controle.

Explícita	Implícita
Dois campos	Mapeamento
Desvio incondicional	Adição
Desvio condicional	Controle residual

Técnica Implícita

- Envolve a combinação ou a adição de duas partes de um endereço para formar o endereço completo.



Questões

1 - Quais os registradores que participam do ciclo de busca? Comente a função de cada um.

2 - Como é interpretada uma microinstrução horizontal?