BC-1503

Arquitetura de Computadores

Processamento e UCP

Guiou Kobayashi

guiou.kobayashi@ufabc.edu.br

2º Quadrimestre, 2014

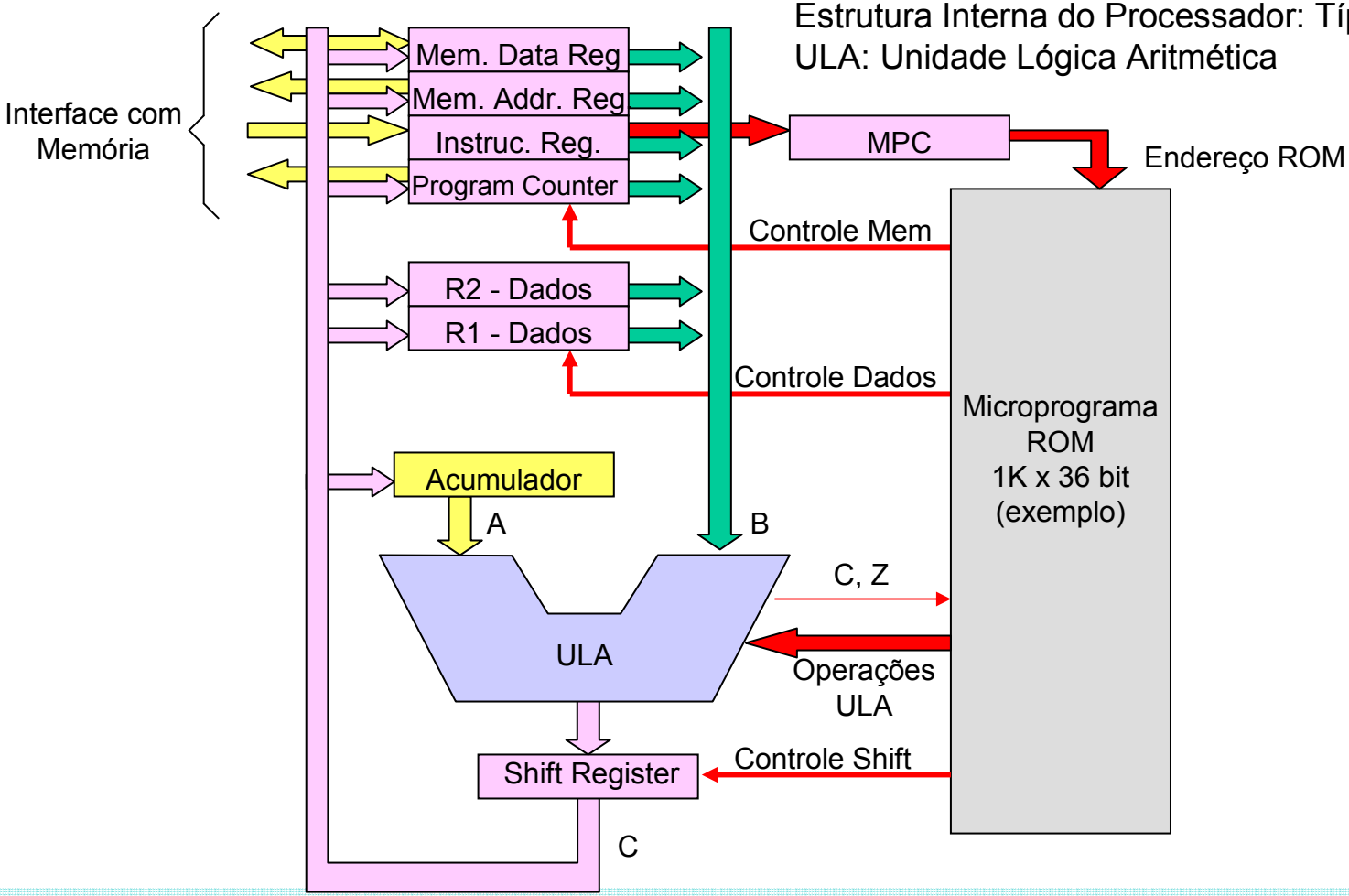
CONTEÚDO PROGRAMÁTICO:

- História e Evolução dos Computadores e Sistemas
- Estrutura de Computadores Digitais
- Lógica Digital Binária
- **Processamento**
- Instruções e linguagem de máquina
- Microprocessadores modernos: pipeline, super escalar, RISC
- Memórias cache e gerenciamento de memórias
- Arquitetura de computadores pessoais
- Arquitetura de Computadores Paralelos
- Sistemas Computacionais: desempenho e confiabilidade

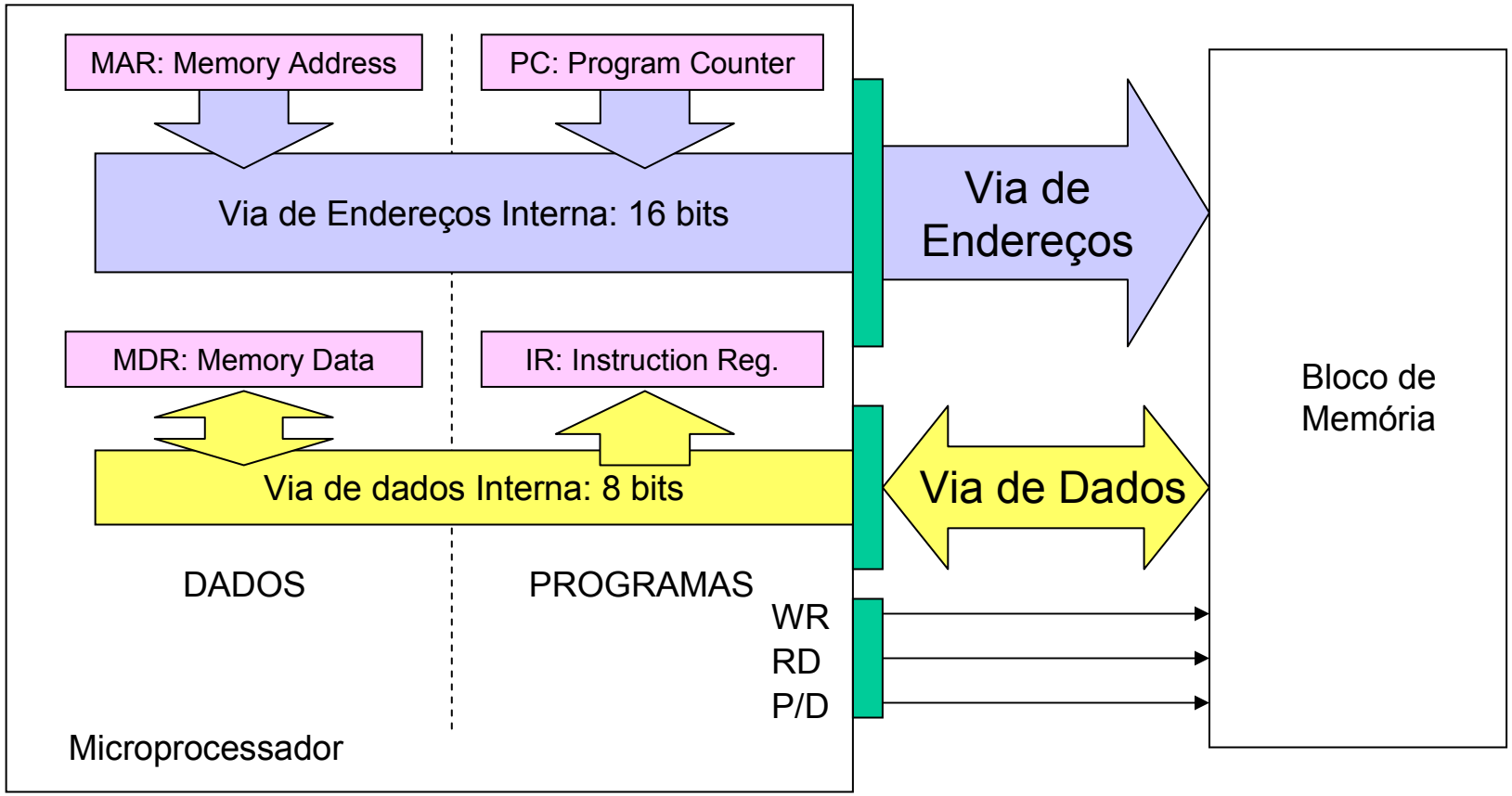
PROCESSAMENTO

UCP: UNIDADE CENTRAL DE PROCESSAMENTO

UCP Didático de 8 bits:
Estrutura Interna do Processador: Típica von Neumann
ULA: Unidade Lógica Aritmética



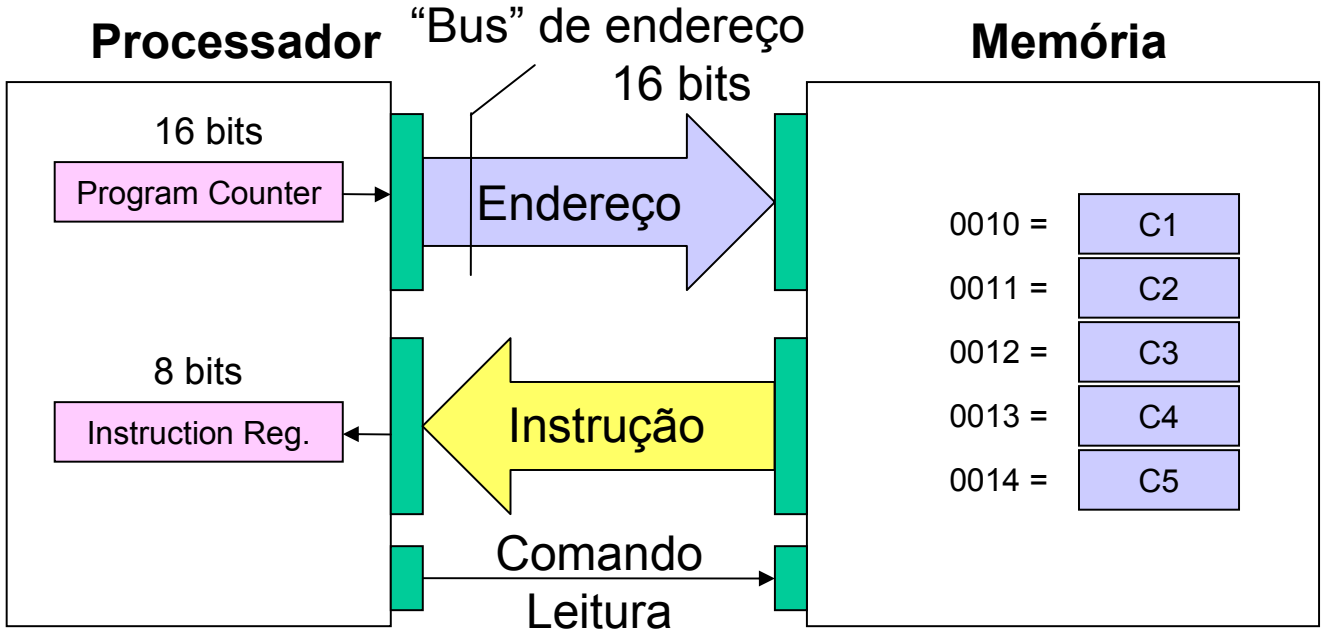
UCP: CONTROLE E ACESSO À MEMÓRIA



Execução da Instrução

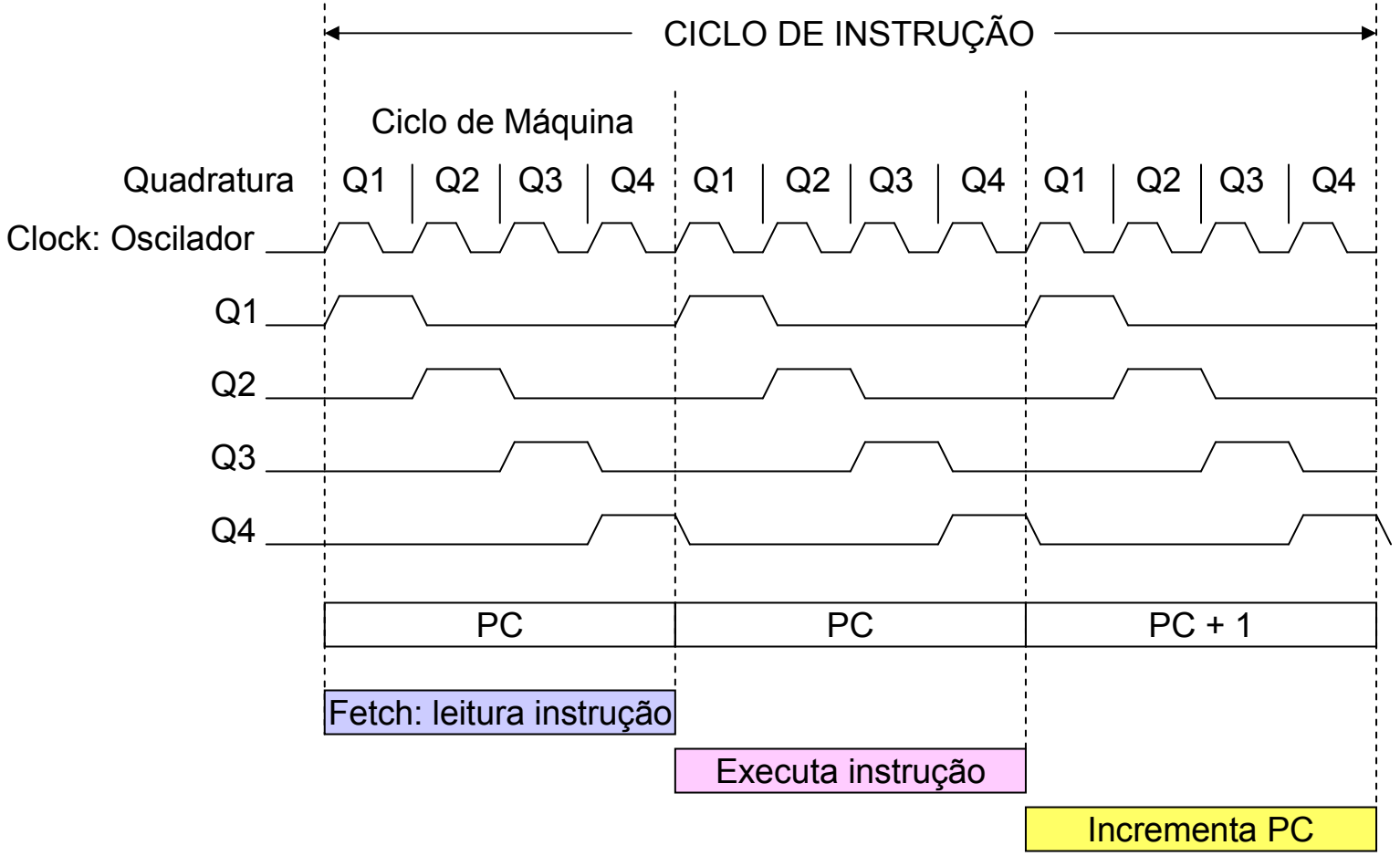
EXECUÇÃO DE UMA INSTRUÇÃO

FETCH: LEITURA DE UMA INSTRUÇÃO DA MEMÓRIA



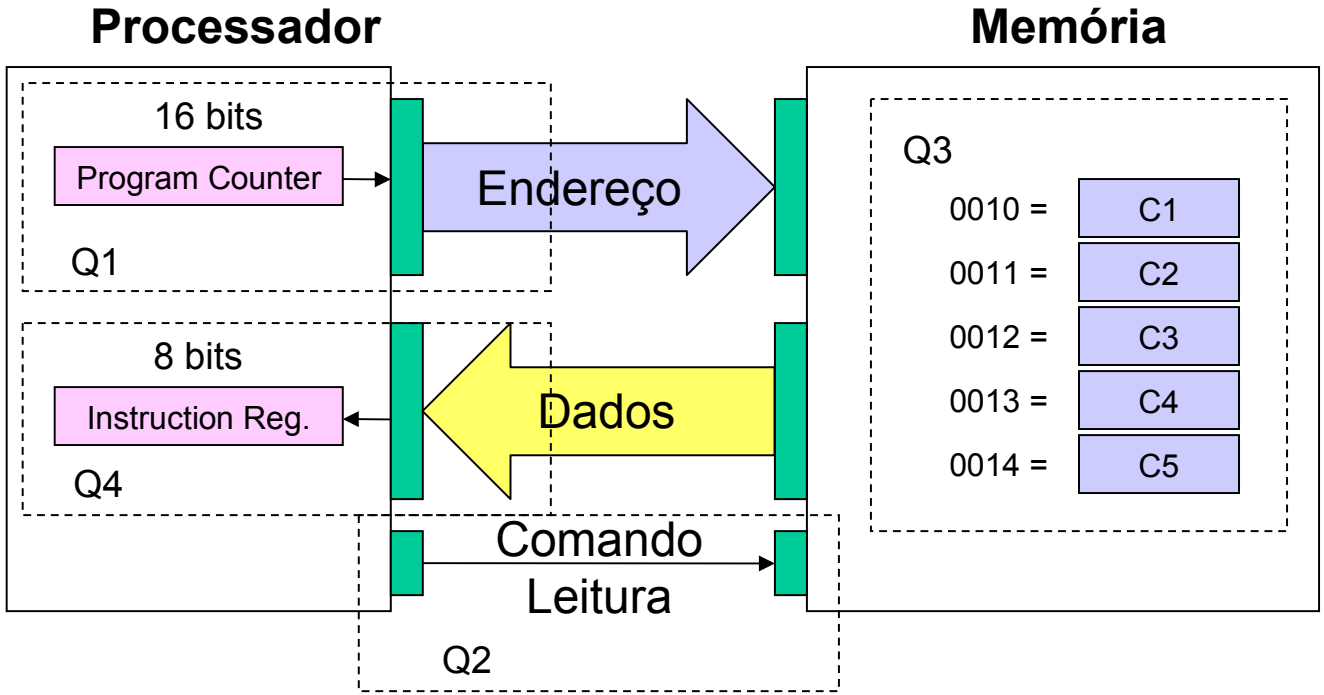
Processador: 8 bits ® dados e instruções 8 bits
Memória: 16 bits de endereço e 8 bits de dados: 64K x 8 = 512K bits

DIAGRAMA DE TEMPOS DO PROCESSADOR



EXECUÇÃO DE UMA INSTRUÇÃO

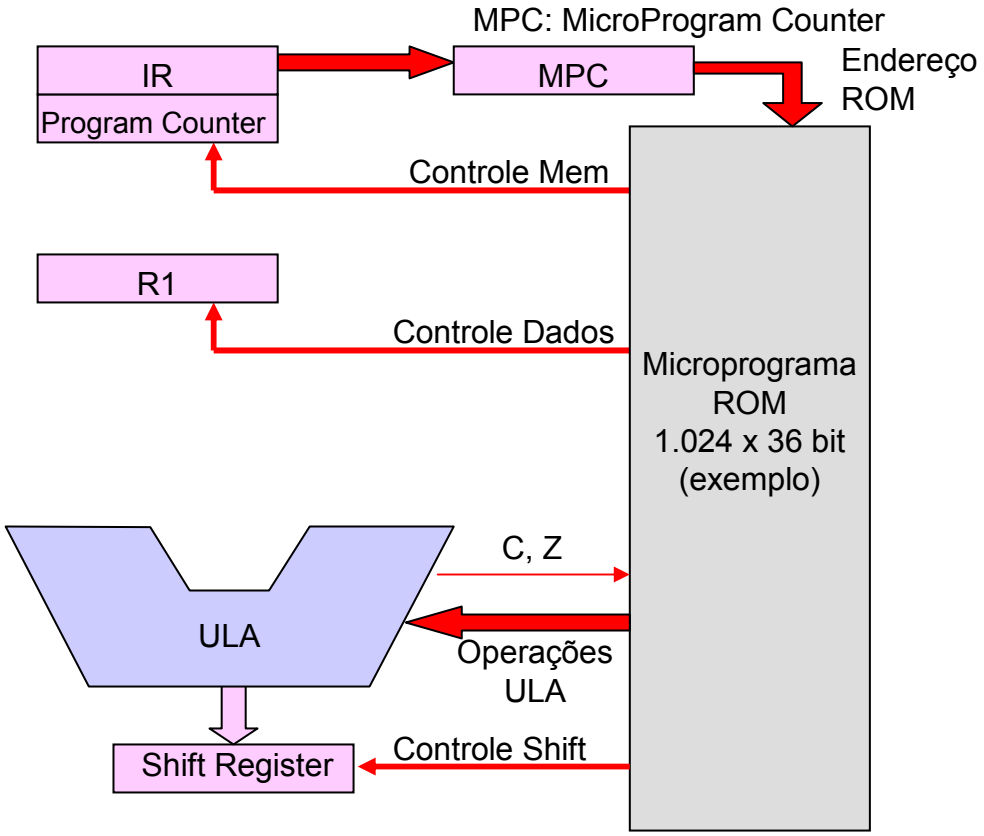
CICLO 1: FETCH



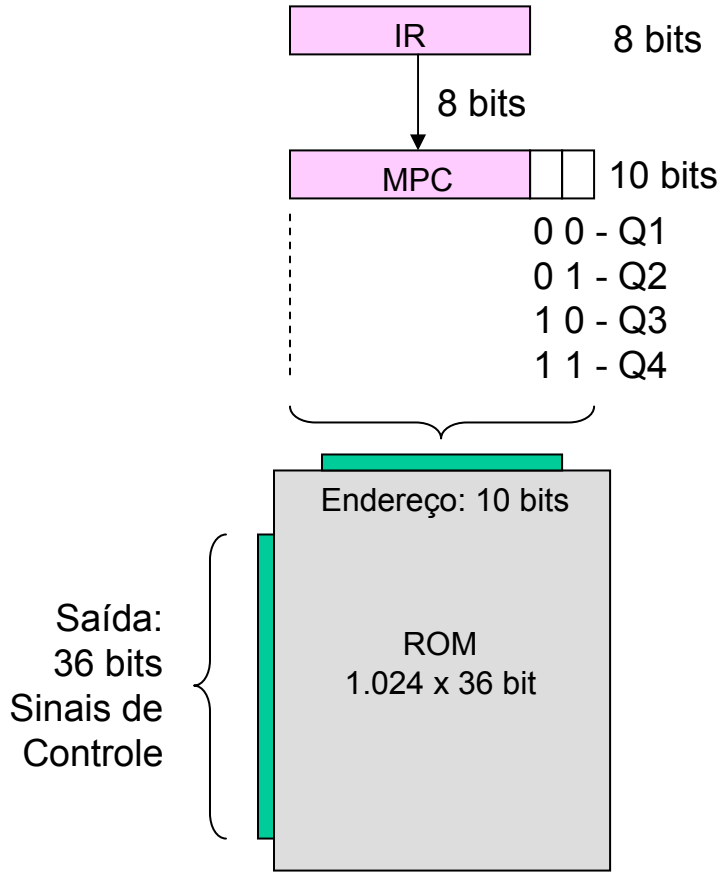
Q1: Processador transfere PC para a Via de Endereços
Q2: Comando de Leitura: Memória decodifica o endereço
Q3: Memória transfere o conteúdo da posição do endereço para Via de Dados
Q4: Processador transfere Bus Dados para IR (registrador de instruções)

EXECUÇÃO DE UMA INSTRUÇÃO

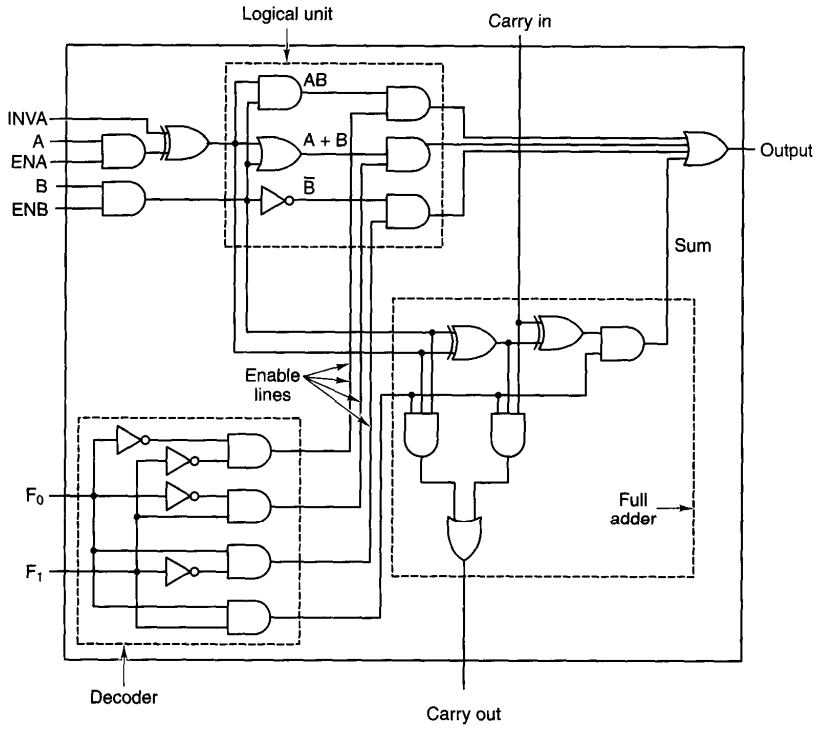
MICROPROGRAMA E SEQUENCIALIZAÇÃO



Exemplo de montagem do MPC



EXECUÇÃO DE UMA INSTRUÇÃO



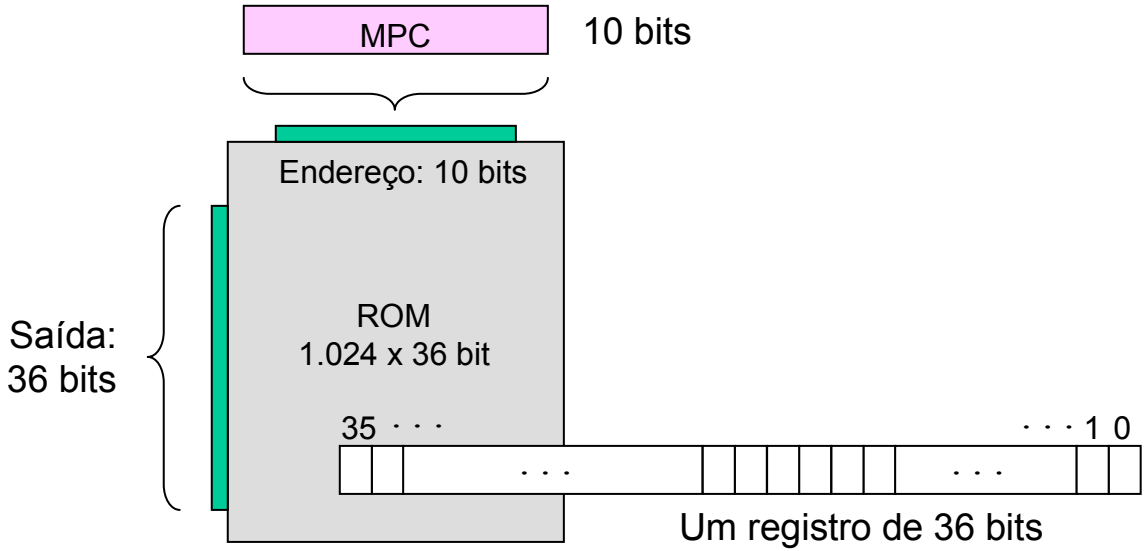
A 1-bit ALU.

EXEMPLOS DE COMANDOS PARA ULA

F_0	F_1	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\overline{A}$
1	0	1	1	0	0	$\overline{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B – A
1	1	0	1	1	0	B – 1
1	1	1	0	1	1	–A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	–1

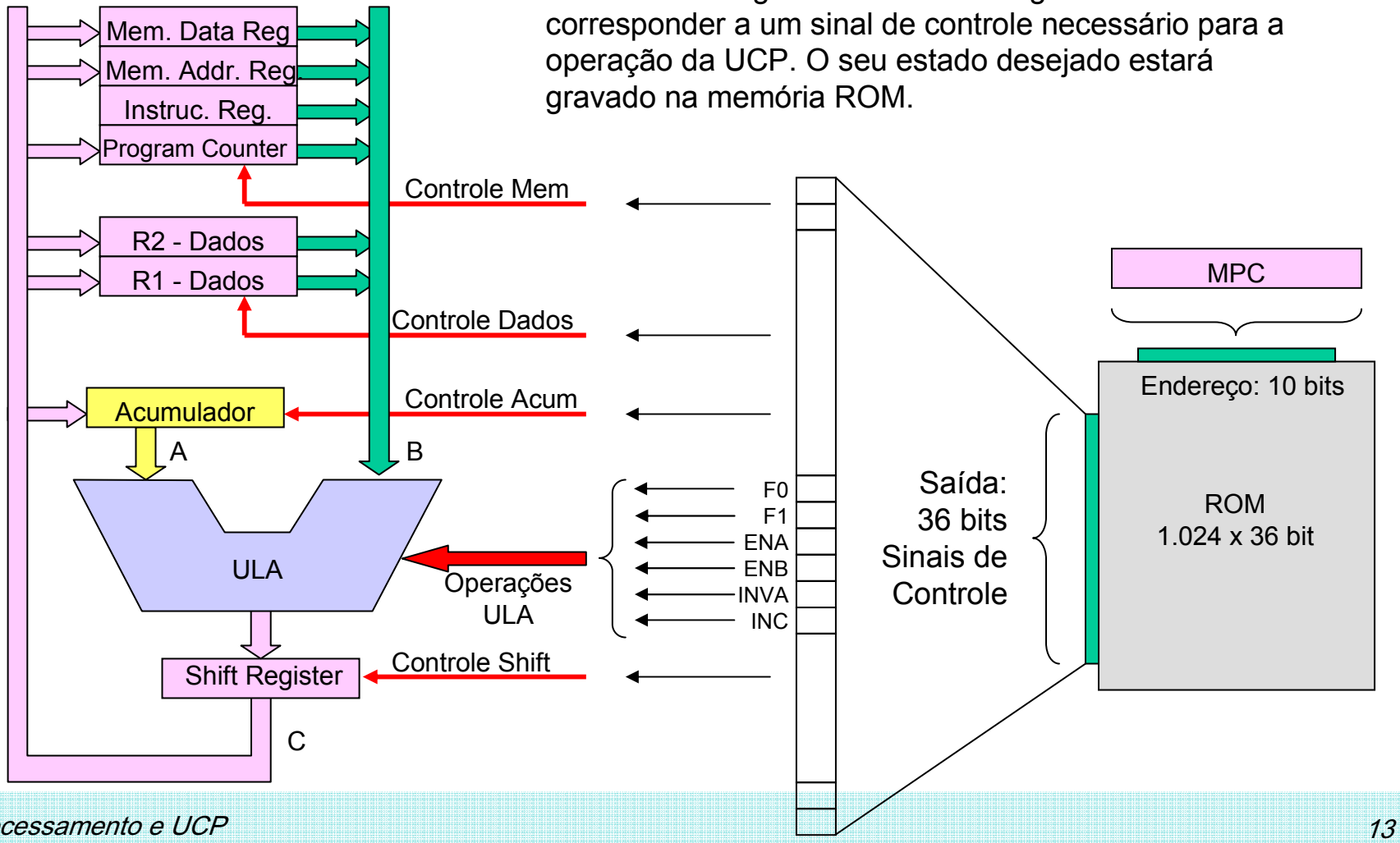
MICROPROGRAMA - 1

- Memória tipo ROM (Read Only Memory), extremamente rápida, com 10 bits de endereço, ou seja 2^{10} ou 1.024 registros.
- Endereço da memória: registrador especial da UCP, chamado de MPC (Micro Program Counter).
- Tamanho do registro: 36 bits.

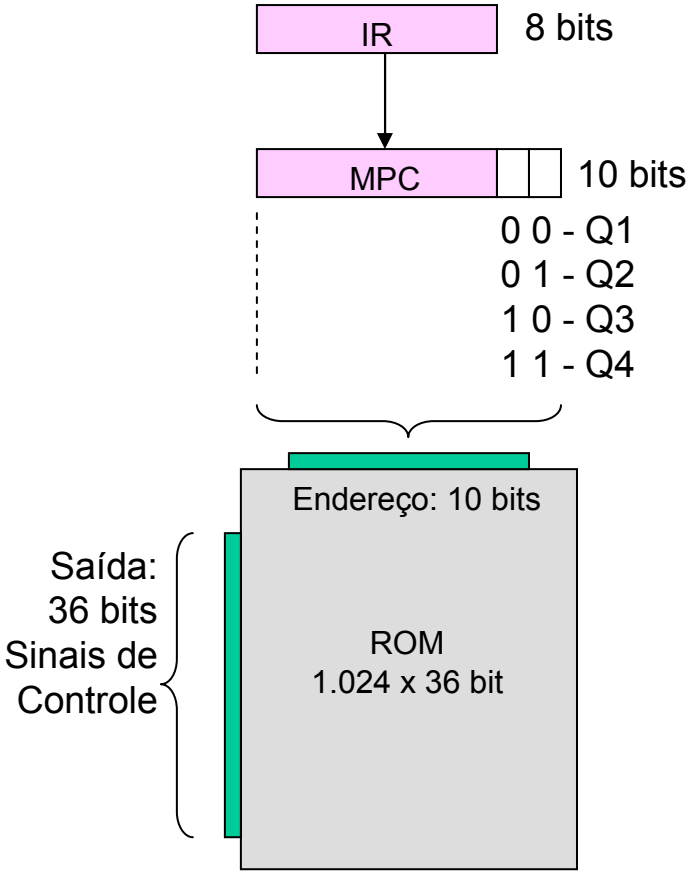


MICROPROGRAMA - 2

Tamanho do registro: cada bit do registro deverá corresponder a um sinal de controle necessário para a operação da UCP. O seu estado desejado estará gravado na memória ROM.



MICROPROGRAMA – 3

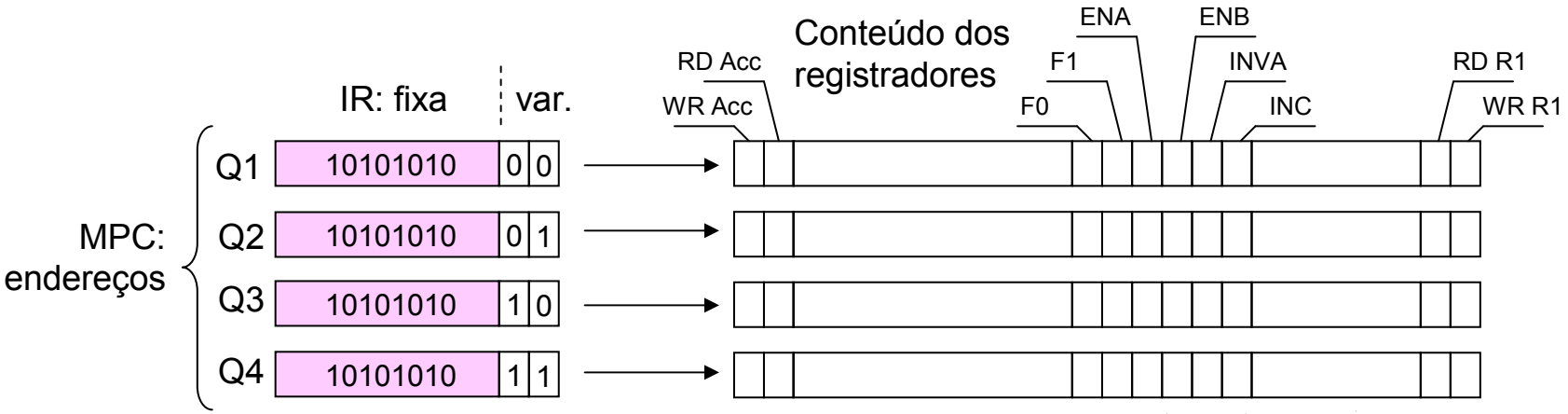


MPC é formado por 10 bits (na nossa UCP didática). Vamos dividir o MPC em duas partes: uma parte fixa formada pelos 8 primeiros bits, e uma parte variável com os dois bits menos significativos. Utilizando um circuito contador de dois bits, e sincronizado pelo relógio da UCP, temos um gerador de endereços do MPC cujos dois bits menos significativos varia de ’00’ a ’11’.

Se a parte fixa de 8 bits for preenchida pelos valores lidos do IR (Instruction Register), temos um endereço de 10 bits que ‘varre’ quatro posições sequenciais, de ‘IR-00’ a ‘IR-11’.

Se para cada uma das quatro posições, atribuirmos ao conteúdo do registrador os sinais necessários para a realização das funções correspondentes aos quatro passos Q1, Q2, Q3 e Q4, implementaremos o mecanismo necessário para a execução do **Ciclo de Máquina da Instrução**.

MICROPROGRAMA – 4



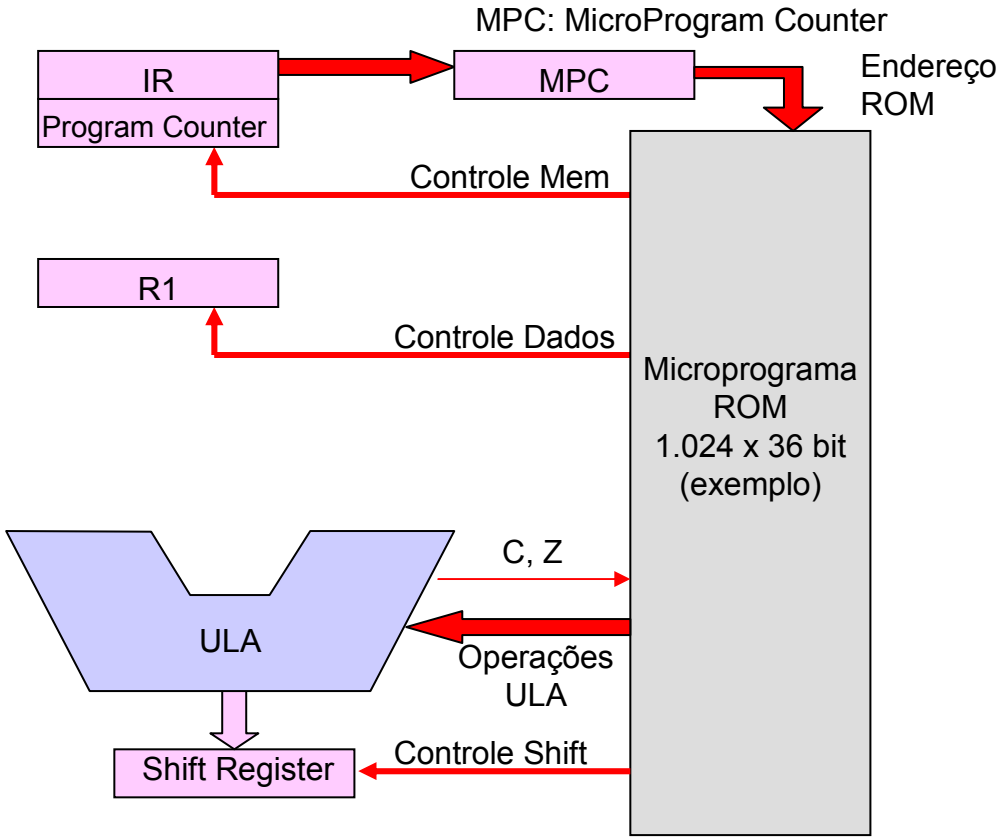
F_0	F_1	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\bar{A}$
1	0	1	1	0	0	$\bar{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B – A
1	1	0	1	1	0	B – 1
1	1	1	0	1	1	–A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
0	1	0	0	0	1	1
0	1	0	0	1	0	–1

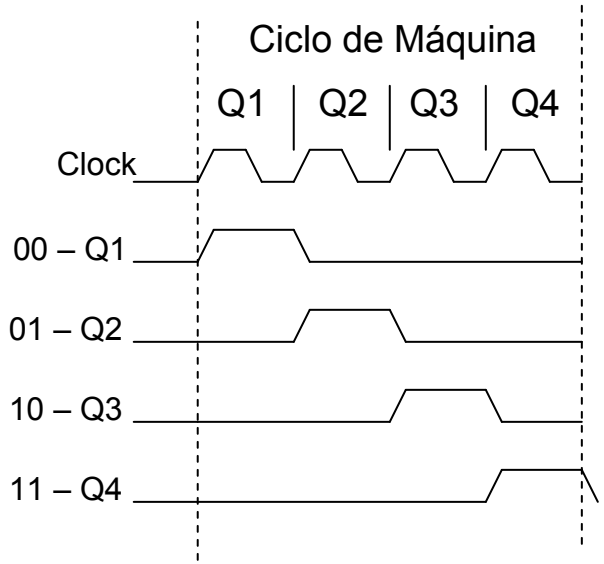
Exemplo:
Acumulador = Acumulador AND R1

Execução:
Q1: R1 → Barramento B
Q2: ULA: Operação AND
Q3: ULA: Retenção Barramento A
Q4: Barramento C → Acumulador

MICROPROGRAMA – 5

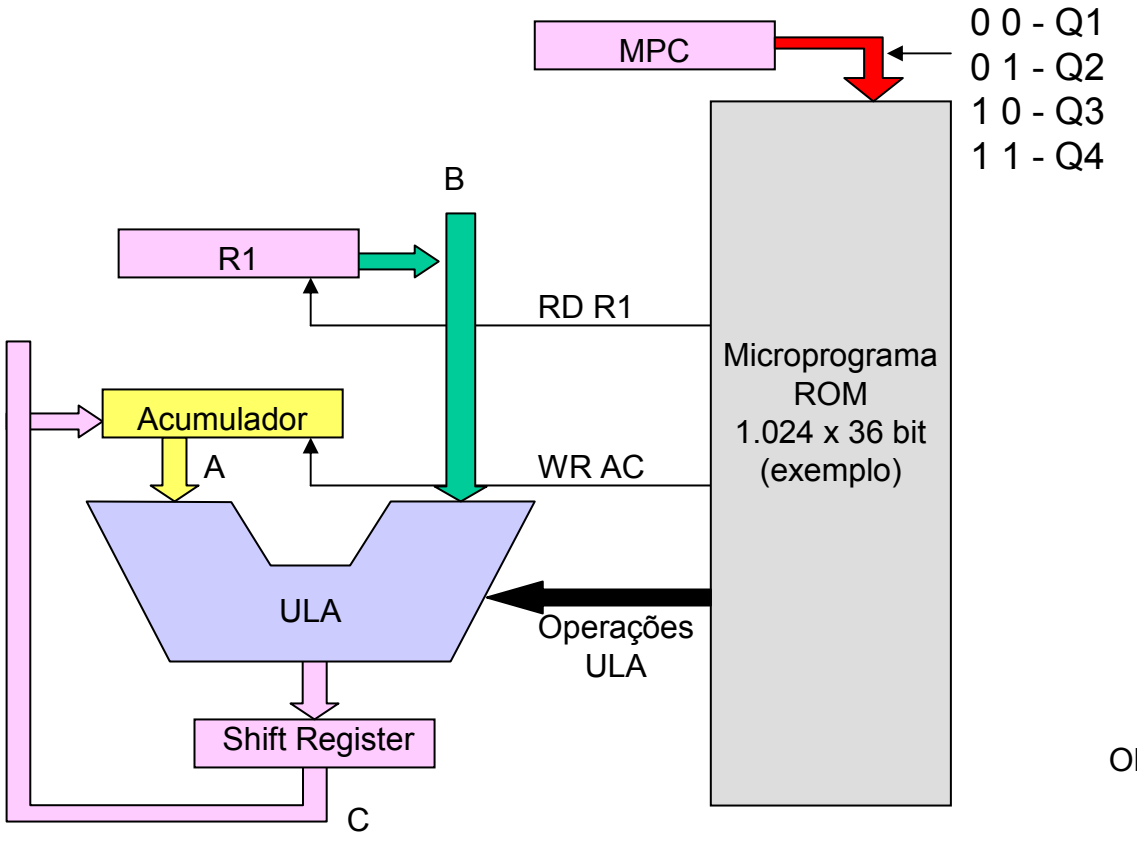
RESUMO





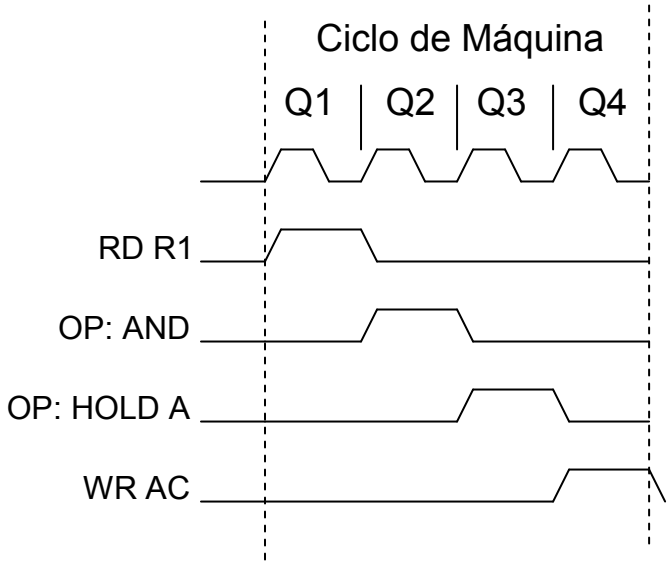
EXECUÇÃO DE UMA INSTRUÇÃO

CICLO 2: EXECUÇÃO DA INSTRUÇÃO:



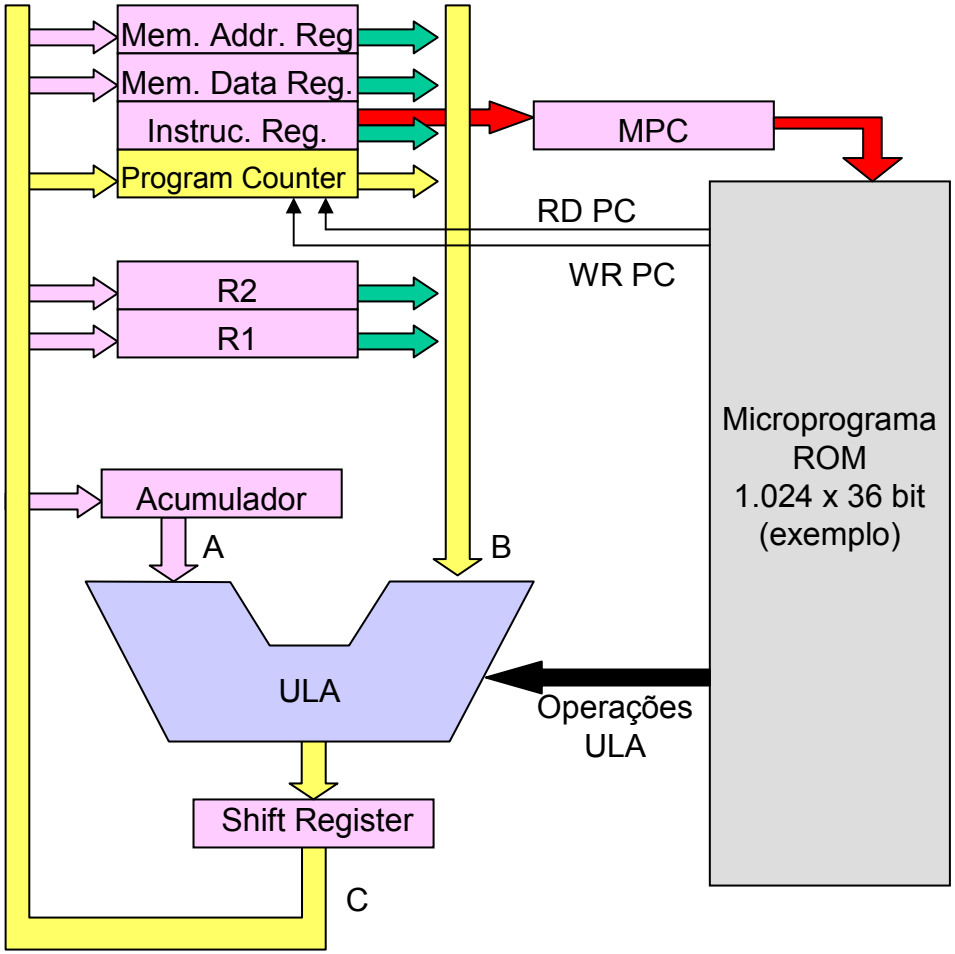
Exemplo:
Acumulador = Acumulador AND R1

Execução:
Q1: R1 → Barramento B
Q2: ULA: Operação AND
Q3: ULA: Retenção Barramento A
Q4: Barramento C → Acumulador



EXECUÇÃO DE UMA INSTRUÇÃO

CICLO 3: INCREMENTA O PROGRAM COUNTER (PC):



PC = PC + 1

Execução:
Q1: ULA: Bloqueia Barramento A
Q2: PC → Barramento B
Q3: ULA: Incrementa Barramento B
Q4: Barramento C → PC

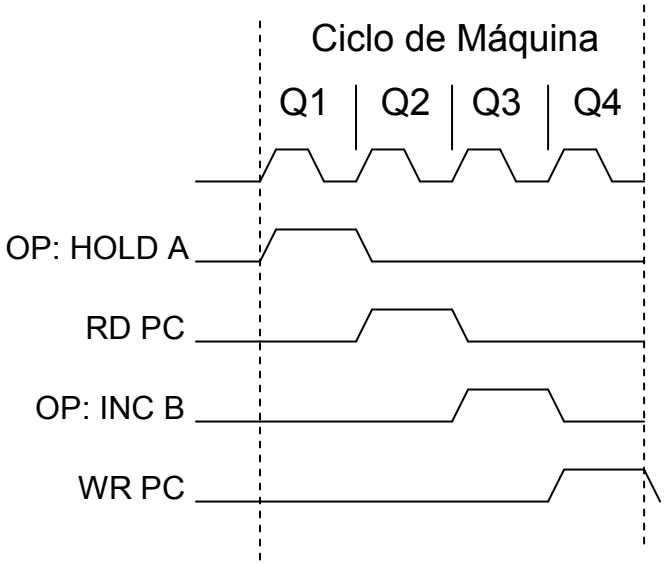
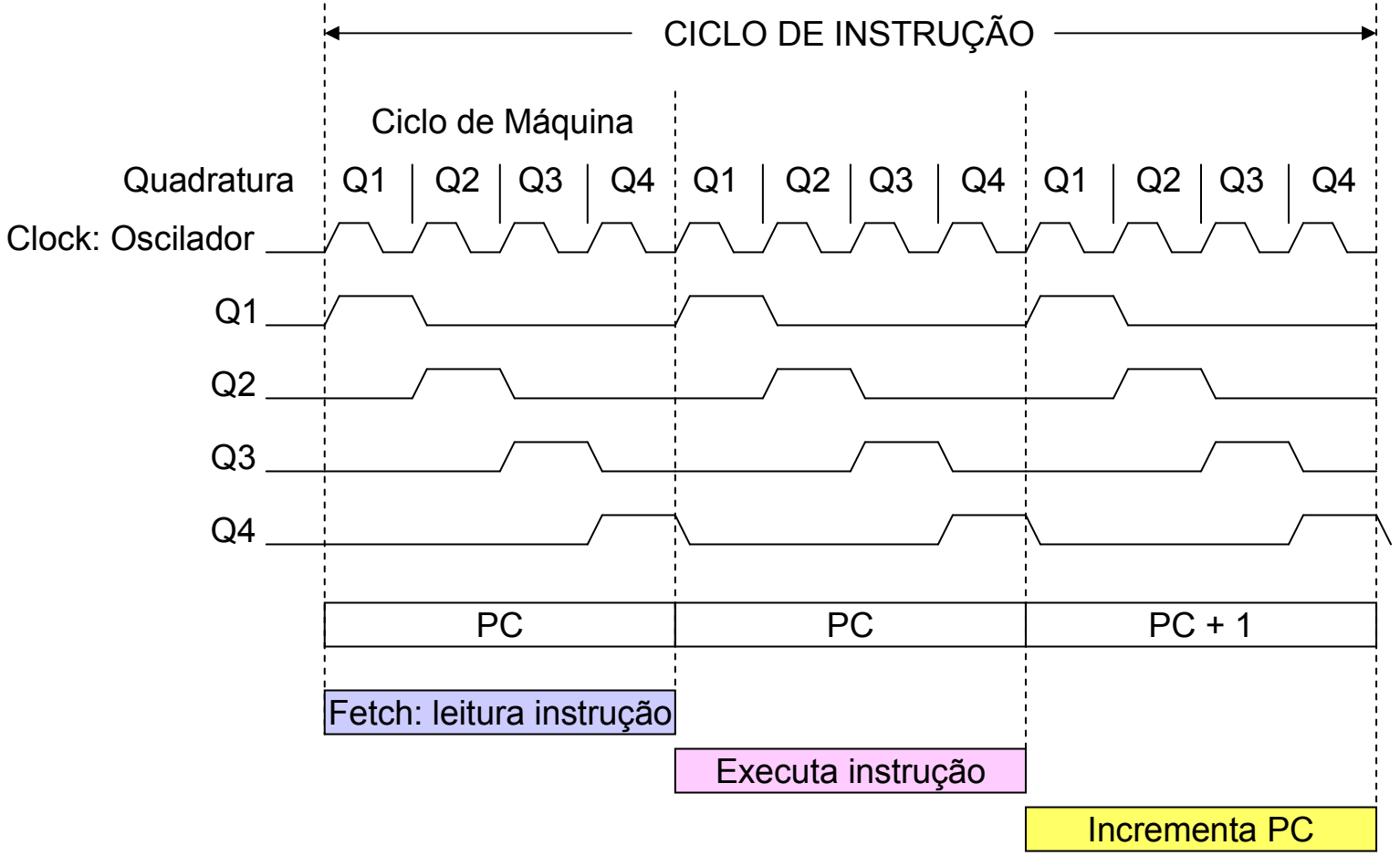
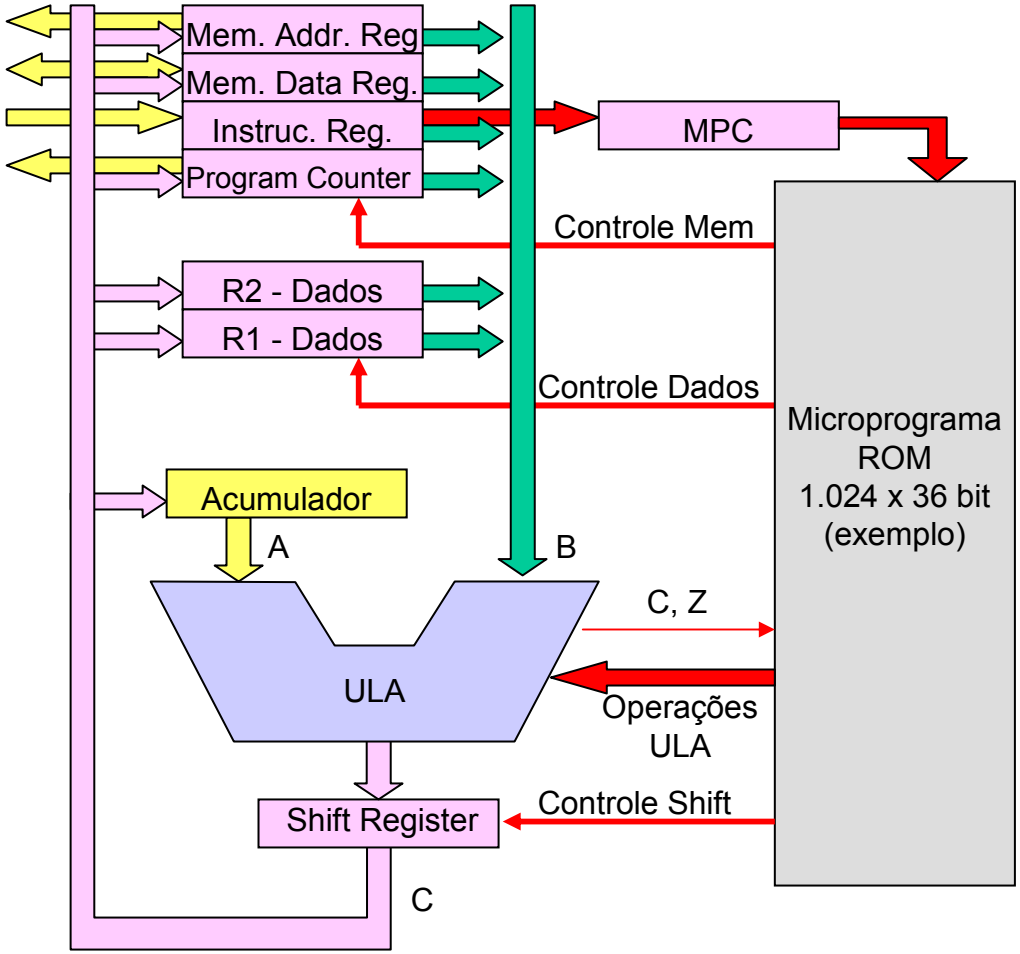


DIAGRAMA DE TEMPOS DO PROCESSADOR



EXECUÇÃO DE UMA INSTRUÇÃO

RESUMO: EXECUÇÃO DA INSTRUÇÃO ACC ← ACC *AND* R1



Um ciclo de Instrução =
Três ciclos de máquina

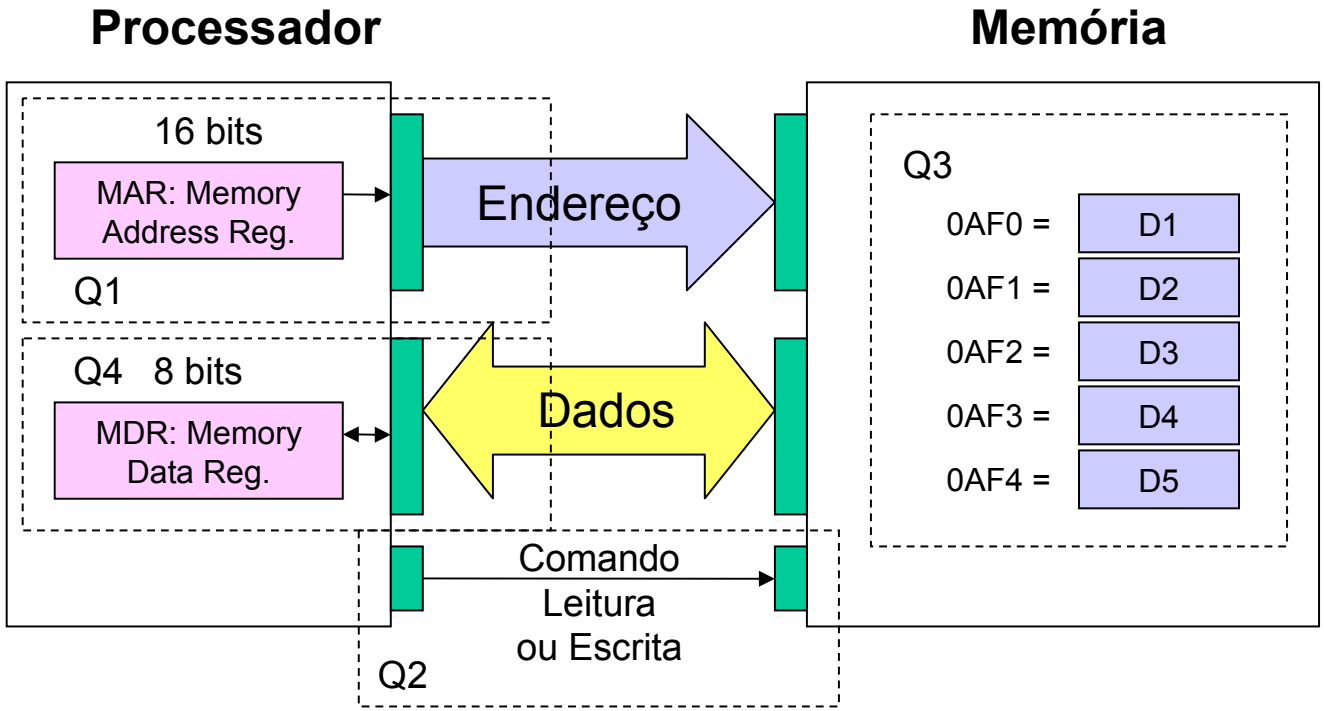
1: FETCH
Q1: PC → Via de Endereços
Q2: Comando de Leitura
Q3: Memória → Via de Dados
Q4: Via de Dados → IR

2: Acumulador = Acumulador AND R1
Q1: R1 → Barramento B
Q2: ULA: Operação AND
Q3: ULA: Retenção Barramento A
Q4: Barramento C → Acumulador

3: PC = PC + 1
Q1: ULA: Bloqueia Barramento A
Q2: PC → Barramento B
Q3: ULA: Incrementa Barramento B
Q4: Barramento C → PC

Mapeamento dos Endereços

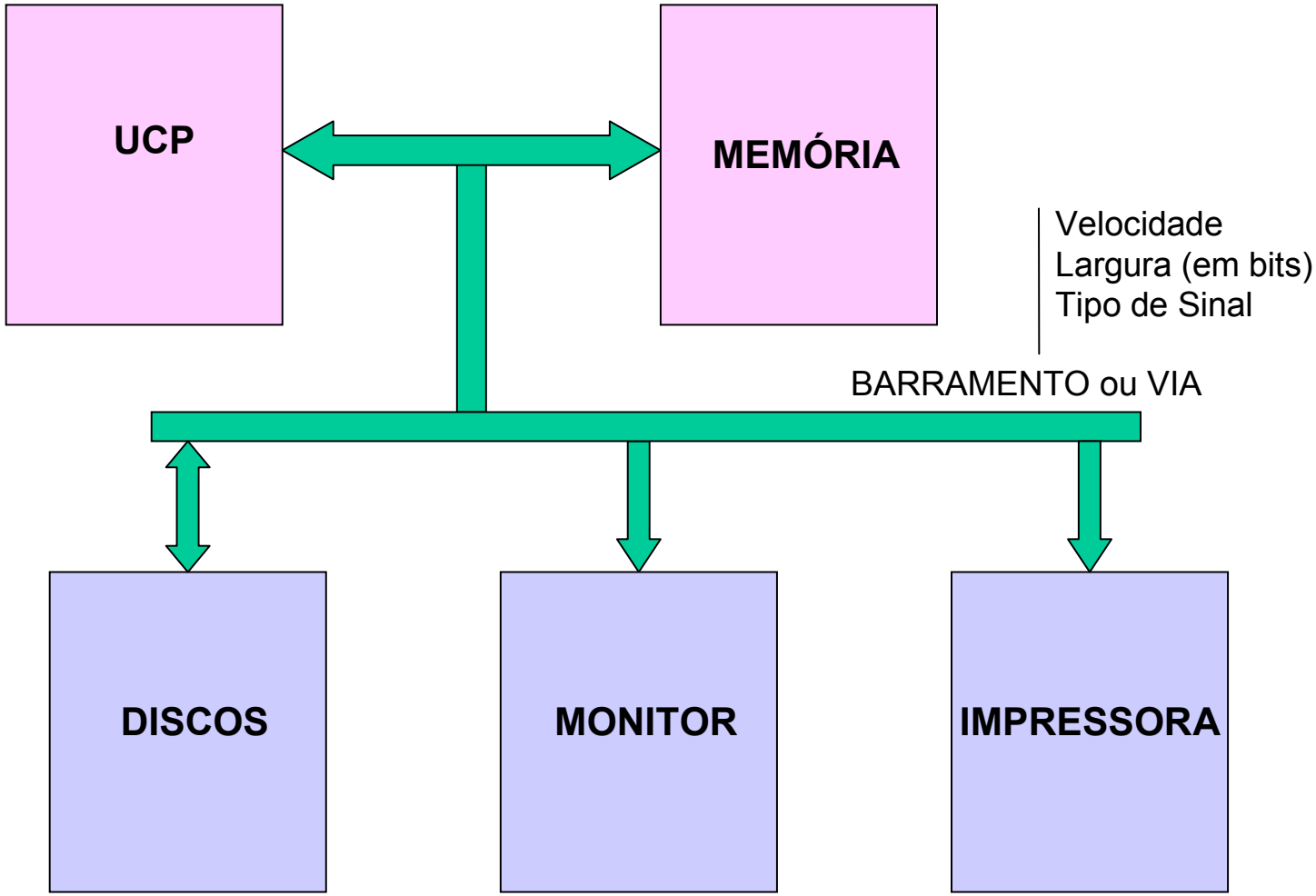
ACESSO À MEMÓRIA DE DADOS: CICLO DE INSTRUÇÕES



LEITURA DA MEMÓRIA
Q1: MAR → Via de Endereços
Q2: Comando de Leitura: Memória decodifica
Q3: Memória → Via de Dados
Q4: Via de Dados → MDR

ESCRITA NA MEMÓRIA
Q1: MAR → Via de Endereços
MDR → Via de Dados
Q2: Comando de Escrita: Memória decodifica
Q3: Via de Dados → Memória
Q4: Estabilização Memória

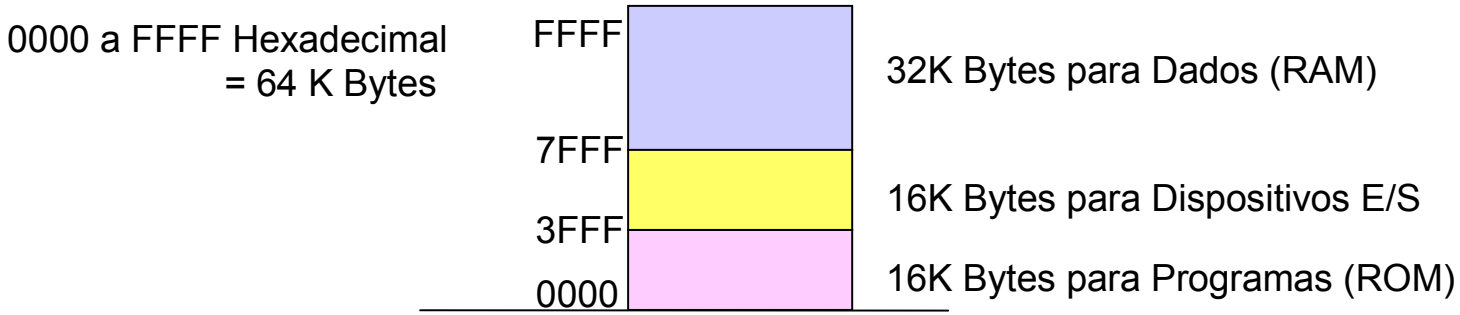
UCP E ACESSO A OUTROS DISPOSITIVOS



ENDEREÇAMENTO DE DISPOSITIVOS E/S

EXEMPLO PARA DISPOSITIVOS MAPEADOS EM MEMÓRIA

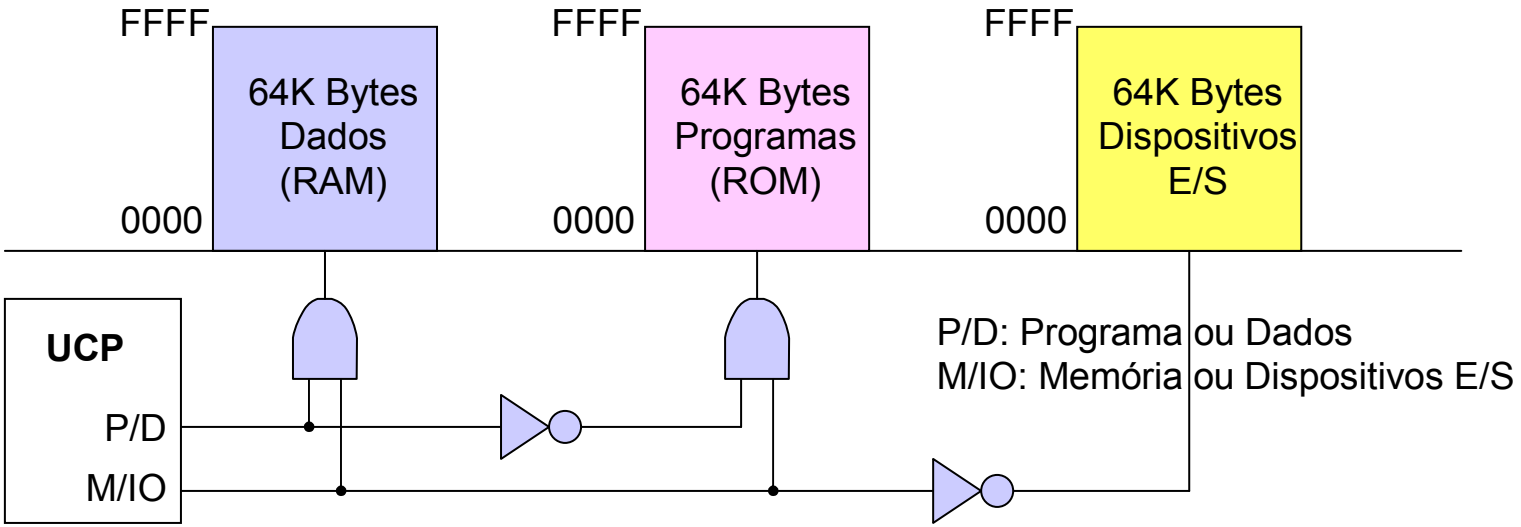
Área Total de 64K Bytes: Divididos com Programas, Dados e Dispositivos E/S



ENDEREÇAMENTO DE DISPOSITIVOS E/S

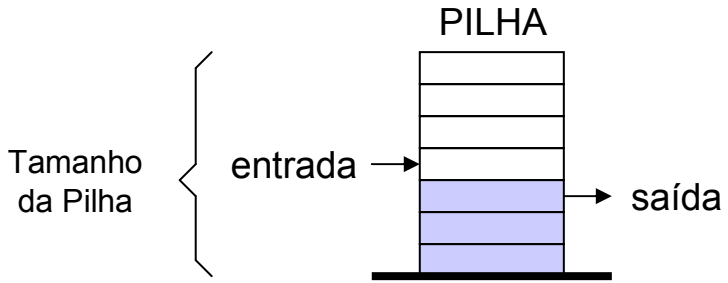
EXEMPLO PARA ÁREA DE MEMÓRIA DE 64K Bytes:

Áreas Separadas de 64K Bytes para Programas, Dados e Dispositivos E/S (uso dos sinais M/IO e P/D)



Pilhas e Stack Pointer

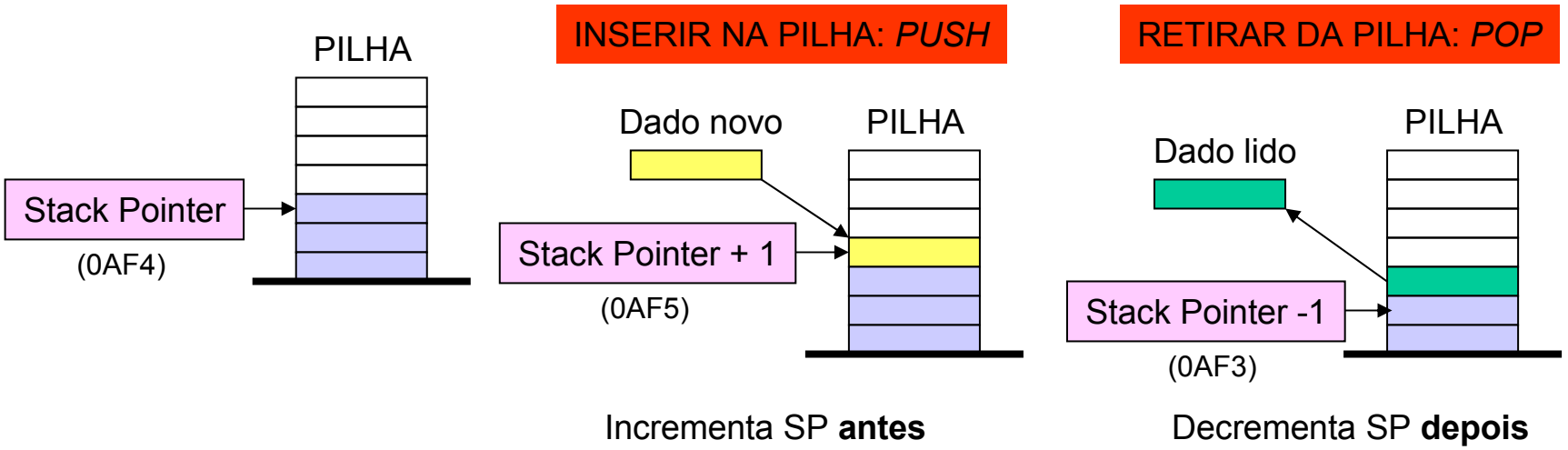
STACK-POINTER: PONTEIRO DE PILHA



Pilha:
Estrutura de Dados do tipo FILO
(First In Last Out)

Stack Pointer: contém o endereço da memória (ponteiro) onde está localizado o último dado escrito (inserido) na pilha.

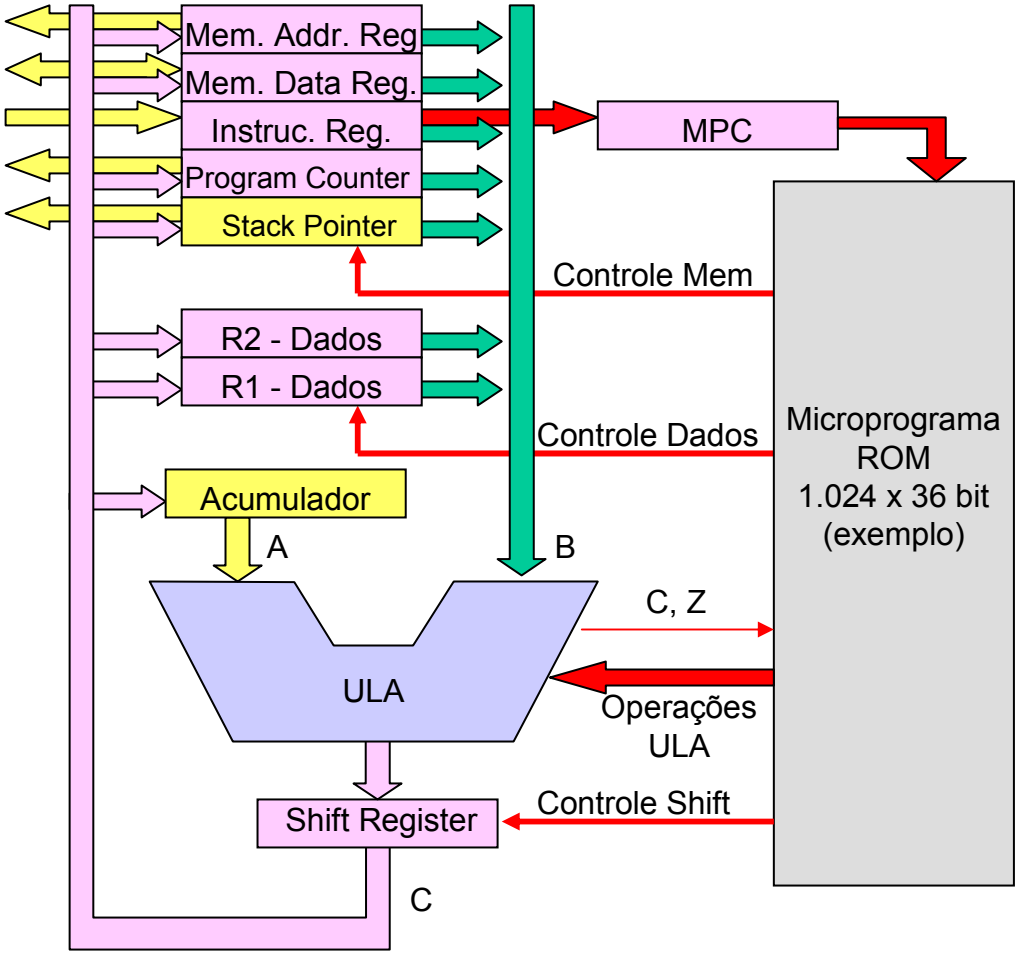
Exemplo: Se o último dado da pilha estiver gravado no endereço 0AF4 (em hexadecimal), o ponteiro irá conter o valor 0AF4.



Incrementa SP **antes**

Decrementa SP **depois**

INSTRUÇÃO INSERÇÃO NA PILHA: PUSH ACC



Quatro ciclos de máquina

1: FETCH da Instrução PUSH
Q1: PC → Via de Endereços
Q2: Comando de Leitura
Q3: Memória → Via de Dados
Q4: Via de Dados → IR

2: SP = SP + 1
Q1: ULA: Bloqueia Barramento A
Q2: SP → Barramento B
Q3: ULA: Incrementa Barramento B
Q4: Barramento C → SP

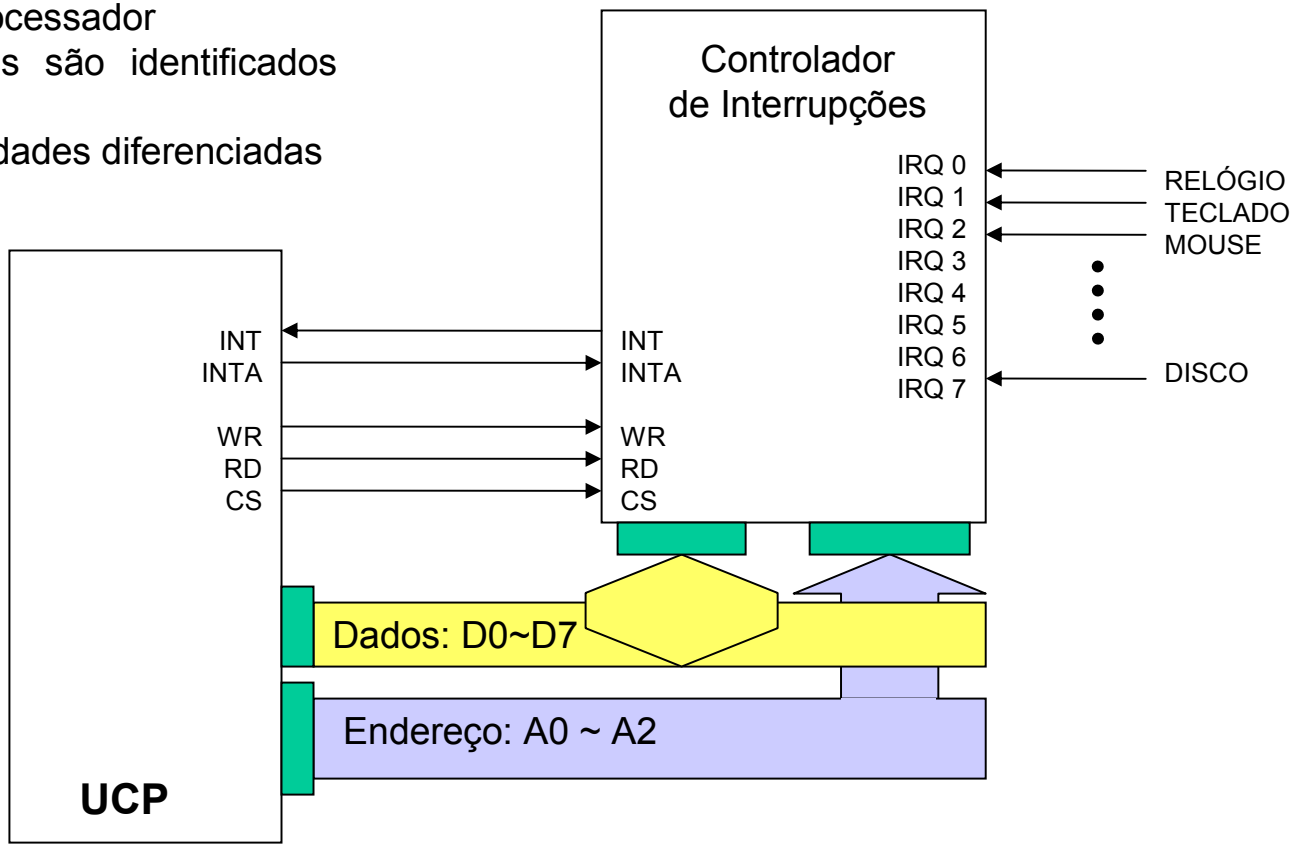
3: ESCRITA NA MEMÓRIA
Q1: SP → Via de Endereços
ACC → Via de Dados
Q2: Comando de Escrita
Q3: Via de Dados → Memória
Q4: Estabilização Memória

4: PC = PC + 1
Q1: ULA: Bloqueia Barramento A
Q2: PC → Barramento B
Q3: ULA: Incrementa Barramento B
Q4: Barramento C → PC

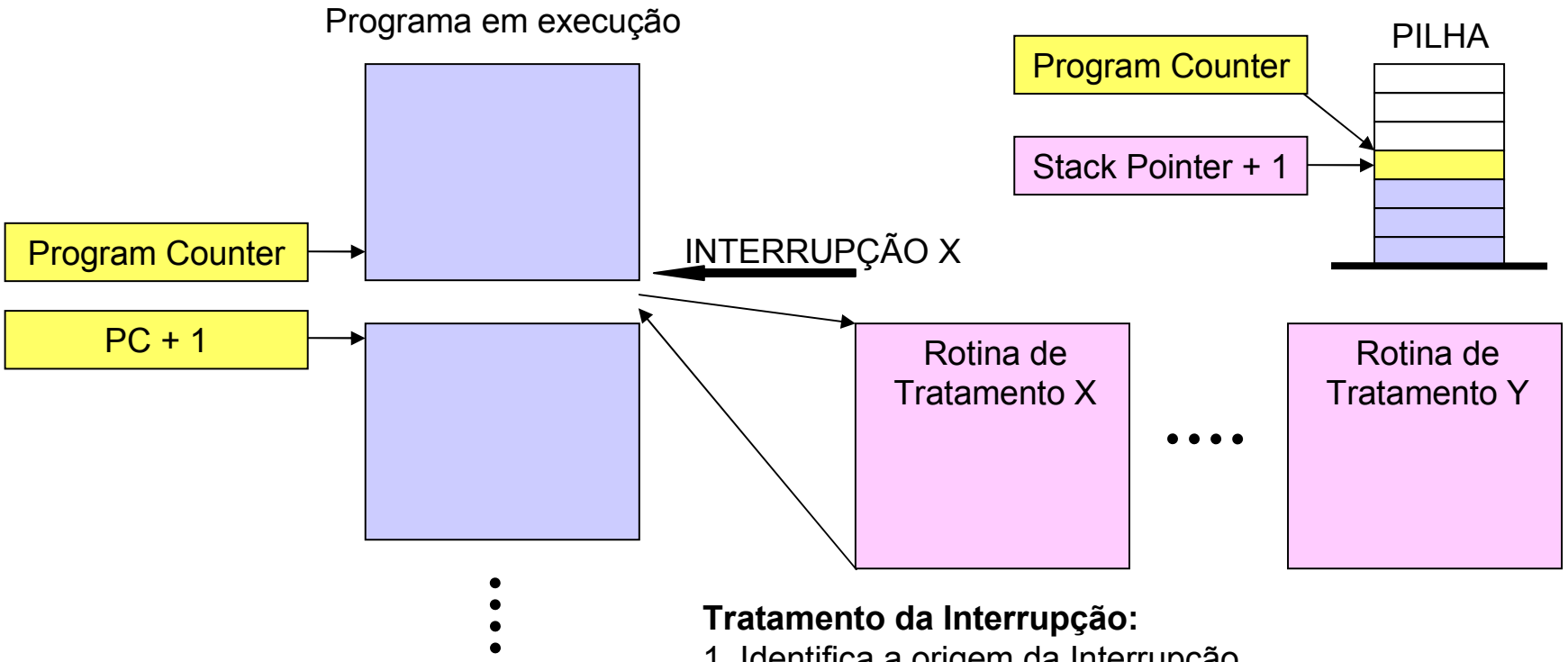
Interrupção

INTERRUPÇÕES

SOLICITAÇÃO DE INTERRUPÇÃO
- Enviados pelos dispositivos que necessitam de um atendimento imediato pelo processador
- Interrupções são identificados individualmente
- Possuem prioridades diferenciadas



SOFTWARE: TRATAMENTO DE INTERRUPÇÕES



Tratamento da Interrupção:

1. Identifica a origem da Interrupção
2. Salva o Program Counter (PC) na pilha
3. Carrega o PC com o endereço da rotina de tratamento correspondente à Interrupção
4. Executa a rotina de tratamento
5. Retorna ao ponto antes da Interrupção: (retira o valor do PC da pilha e restaura)