

---

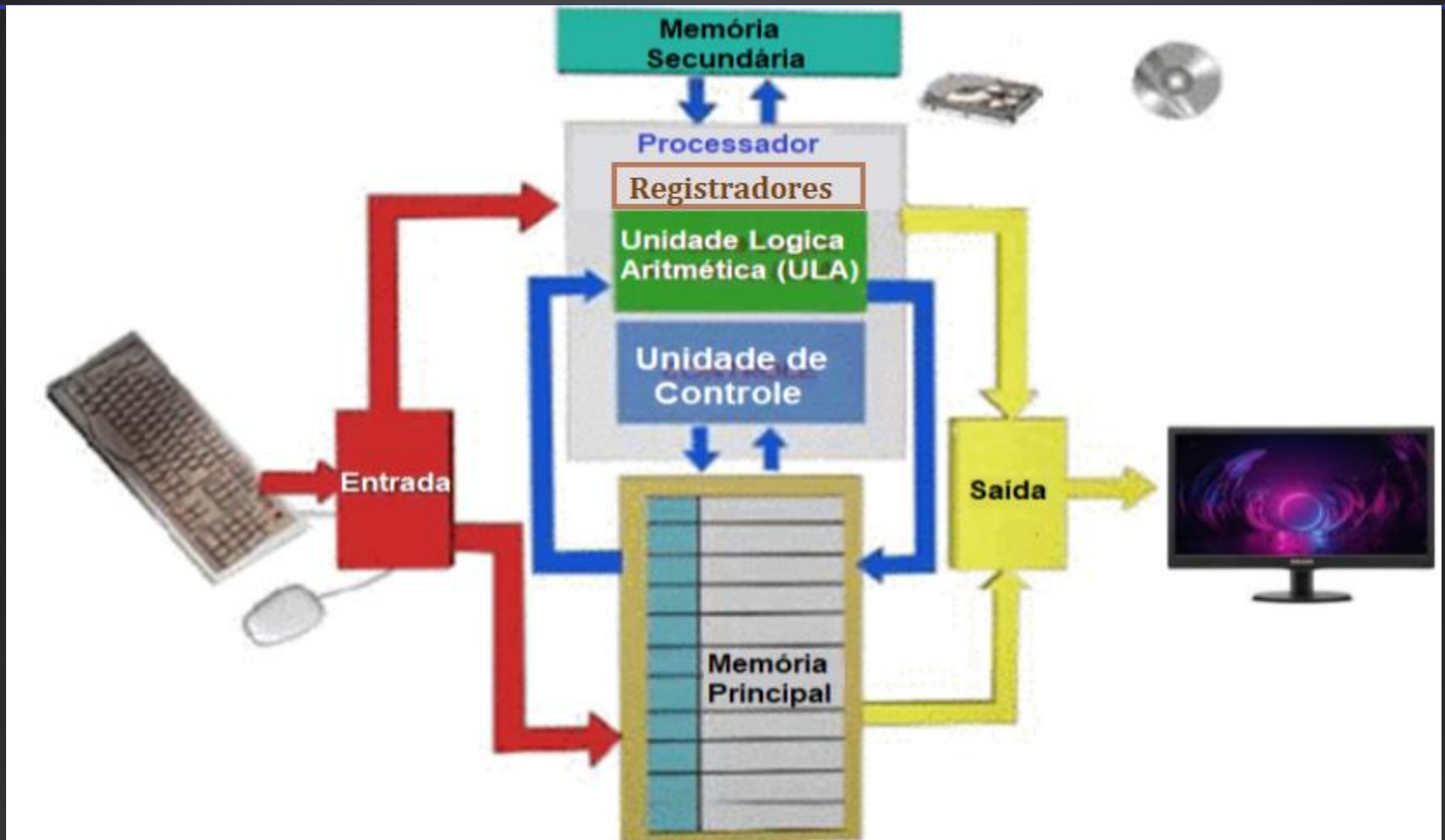
## 6. CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

# MICROPROCESSADOR

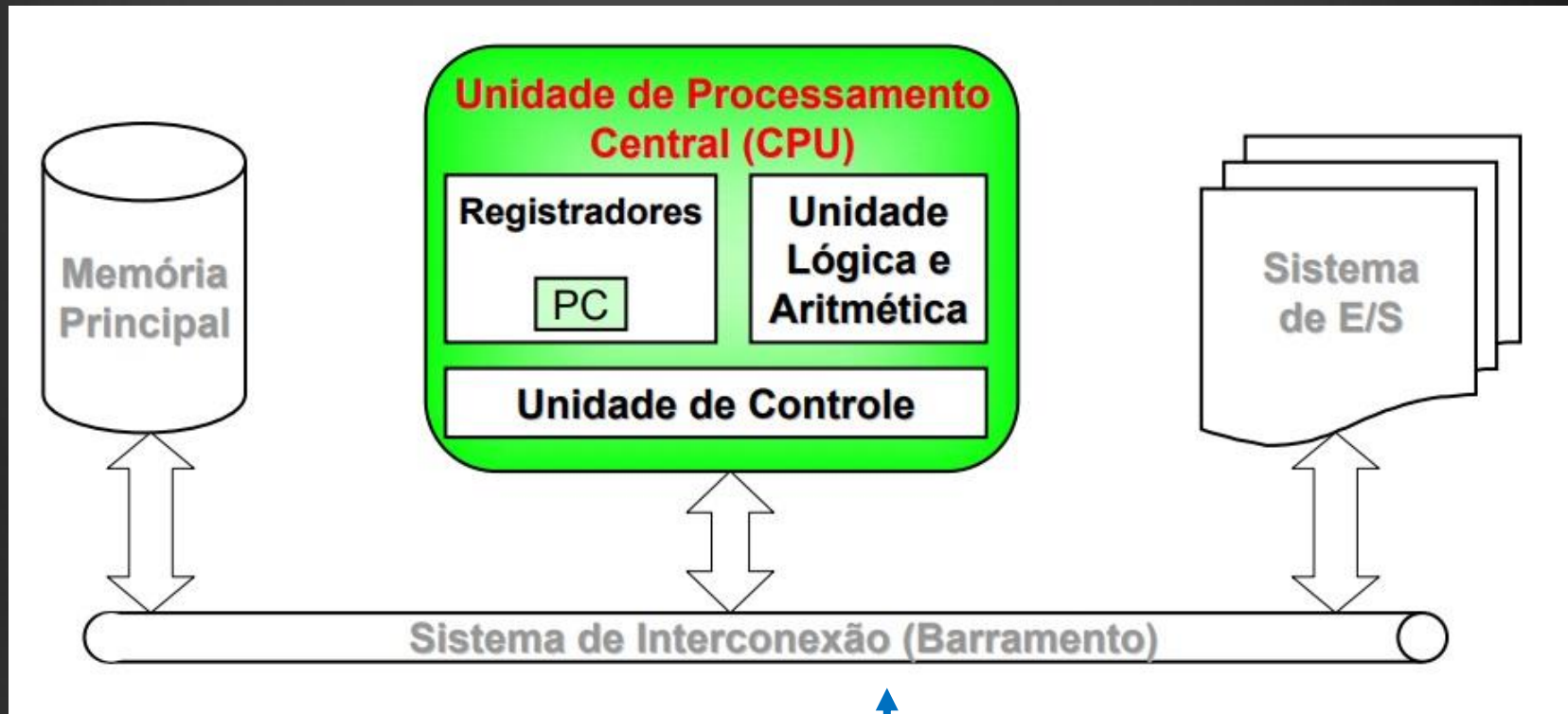
---



# MICROPROCESSADOR



# MICROPROCESSADOR



Composto por pistas de cobre, CIs (chipset) e conectores



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Em termos práticos, a única tarefa realizada pela CPU e pelo computador é a execução de programas.
- As instruções que compõem os programas ficam armazenadas na memória principal e devem ser levadas até a CPU para execução.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Até mesmo para carregar as instruções na memória, em um computador, são necessários programas como o sistema de inicialização (o BIOS) que fica armazenado em memória ROM e tem como função carregar o Sistema Operacional, que por sua vez carregará a memória e gerenciará a execução dos demais programas.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Para que uma instrução seja executada, a função de busca de instrução deve carregar a instrução no Registrador de Instrução.
- A instrução é então interpretada pelo Unidade de Controle
- Só então ela é executada pela ULA
- Este é o chamado de ciclo de instrução.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- CHIPSET

- O chipset é um componente fundamental para o funcionamento do PC. O nome se refere a um conjunto de circuitos integrados que são responsáveis por fazer com que todos os componentes do computador, desde o disco rígido até o processador, possam trocar informações e assim realizar as tarefas que exigimos deles.



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- CHIPSET

- O chipset é dividido em dois componentes principais: ponte norte (northbridge) e ponte sul (southbridge). A ponte norte fica responsável por controlar todos os componentes rápidos do computador, como processador, placa de vídeo (AGP e PCI Express) e memória RAM, fazendo com que eles solicitem informações do disco rígido (que está na ponte sul), as carregue na memória e divida o que será processado entre a CPU e a placa de vídeo, determinando qual será o desempenho final do computador.

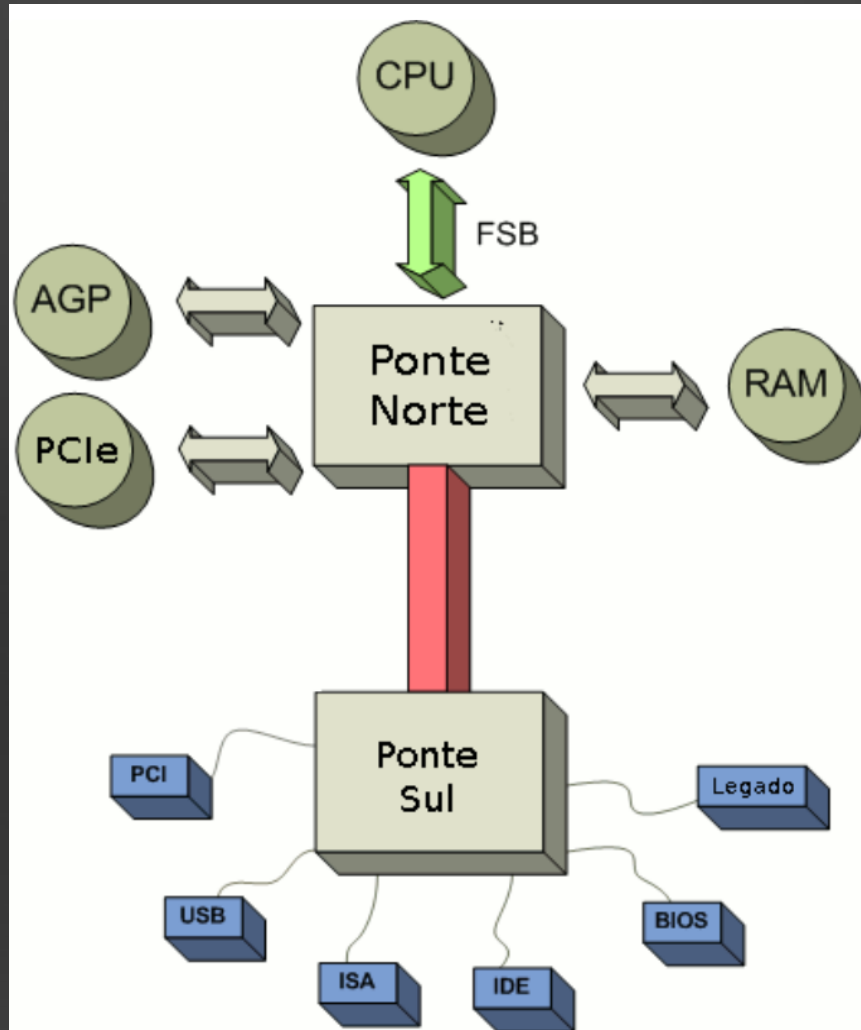
# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- CHIPSET
- A ponte sul fica responsável pelos componentes lentos do PC, também conhecidos como dispositivos de E/S (entrada/saída), o que inclui os discos rígidos (SATA e IDE), portas USB, pararela e PS/2 (utilizada em teclados e mouses antigos), slots PCI e ISA (padrão da IBM, hoje em desuso).

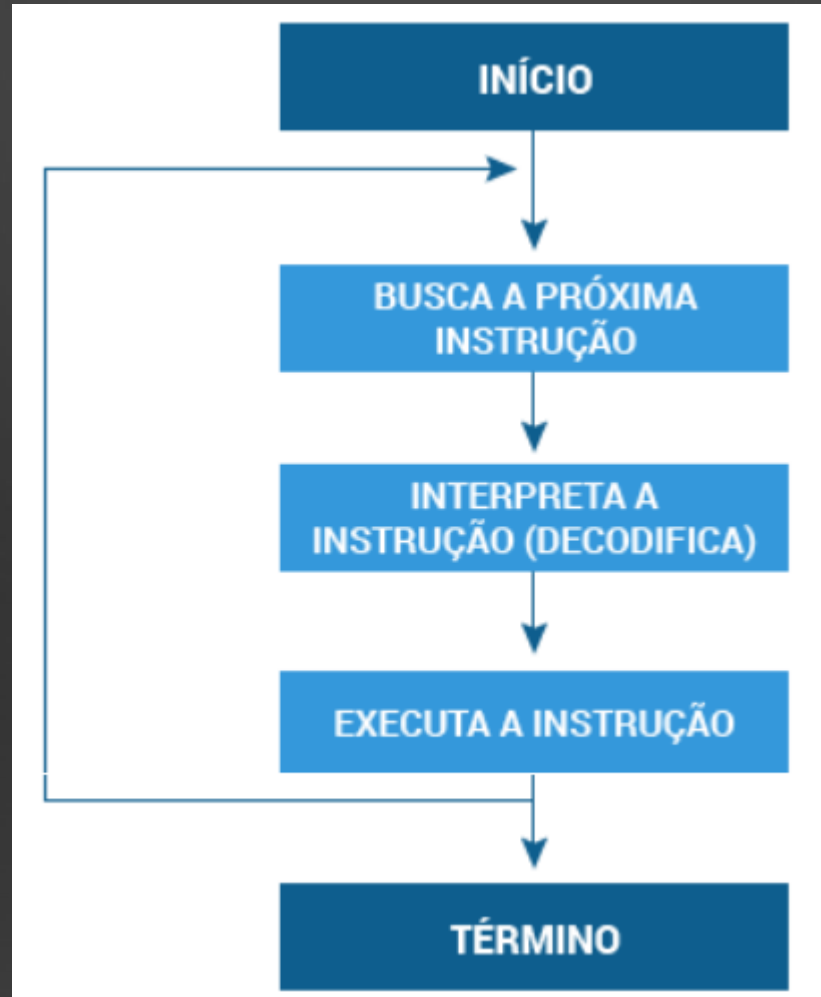
# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

- CHIPSET



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Agora é possível avaliar com mais detalhes o que acontece em cada uma dessas etapas.
- Para buscar a instrução na memória é necessário, primeiramente, saber em que endereço da memória está a instrução a ser executada e, na sequência, trazer a instrução (conteúdo do endereço) para o Microprocessador

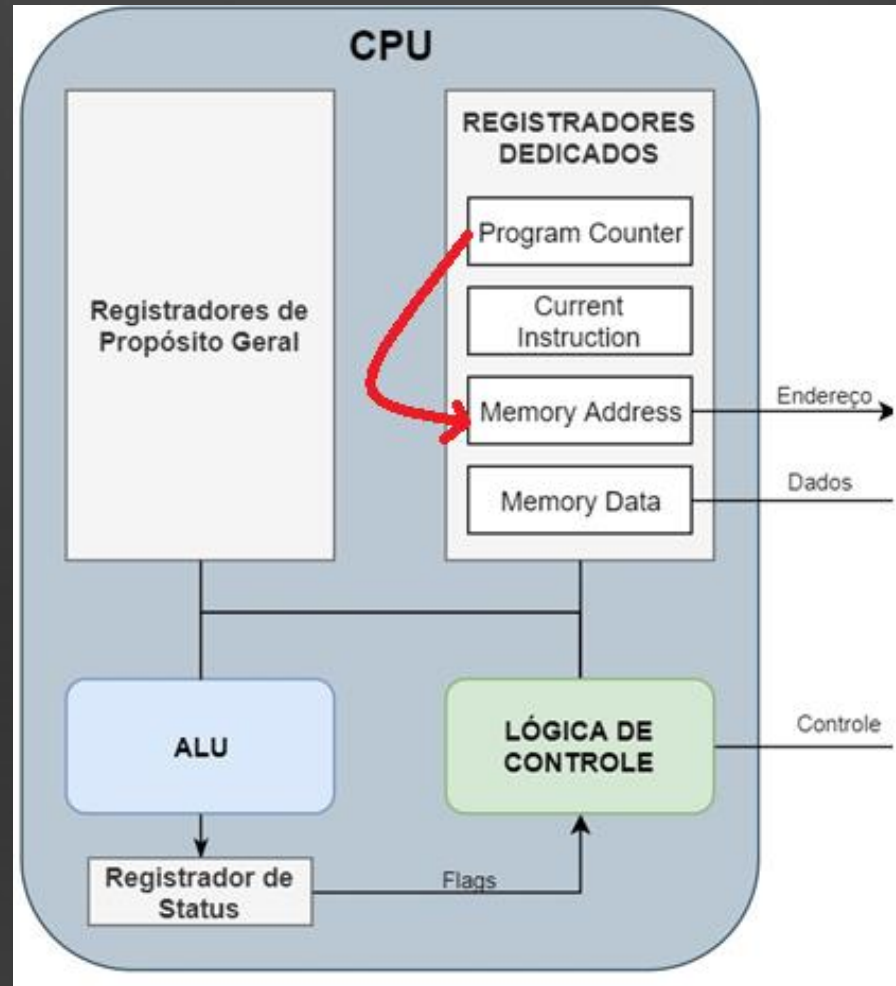


# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Cálculo do endereço da instrução.
  - O conteúdo do registrador Contador de Programa (Program Counter) é copiado para o Registrador de Endereço (Memory Address).

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

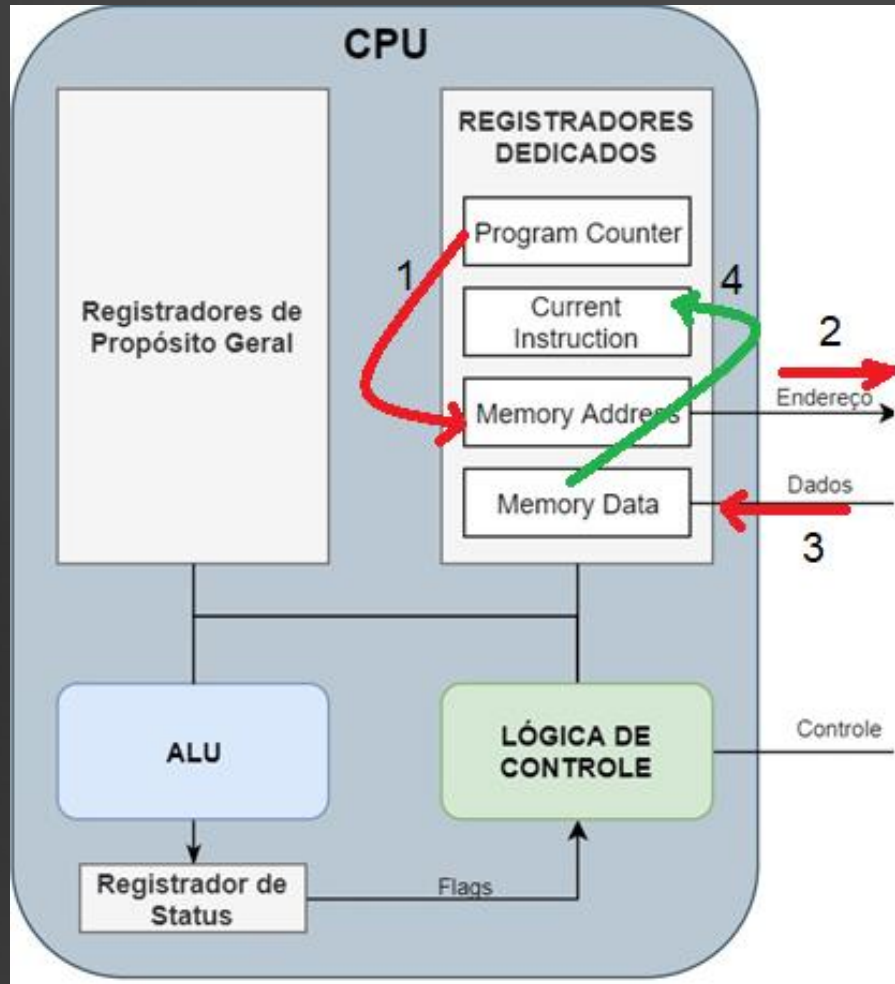


# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Busca da instrução.
  - A Lógica de Controle (Unidade de Controle) gera o sinal de leitura, e o conteúdo da posição solicitada é transmitido para a Memória de Dados do Microprocessador (Memory Data),
  - O conteúdo da Memória de Dados é então enviado para o Registrador de Instruções (Current Instruction).

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Decodificação da Instrução:
  - A Lógica de Controle (Unidade de Controle) decodifica a instrução (código de operação) armazenada no Registrador de Instrução Corrente (Current Instruction) e passa a gerar os sinais de controle na ordem necessária para concluir a operação.



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

O formato da instrução tem, necessariamente, um código (opcode), e pode variar quanto ao número de operandos. Podem existir operações com 1, 2 ou até mais operandos. Existem ainda operações que não possuem operando algum. Ou seja, existem diferentes formas de representar uma instrução no processador.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

Neste ponto o Microprocessador tem a instrução e finalmente "sabe" o que deve ser feito. Podendo proceder a execução da instrução, através dos seguintes passos:

- Cálculo do(s) endereço(s) do(s) operando(s).  
Verifica onde está(ão) o(s) operando(s) da instrução. Os operandos podem estar na Memória Principal, em um registrador, ou em um periférico de entrada.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Busca do(s) operando(s).  
Vai até a posição de memória, registrador, ou periférico, e copia os dados do operando para o local adequado.
- Execução da operação.  
De posse do código de operação e dos operandos é possível executar a instrução, etapa que ocorre na ULA (Unidade Lógica Aritmética).

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Cálculo do(s) endereço(s) do(s) resultado(s).  
Verifica onde deve ser armazenado o resultado (operando destino). Isto pode ocorrer em um registrador, na memória, ou periférico de saída.
- Armazenamento do(s) resultado(s).
  - Escreve o resultado no local adequado.
  - Se a Lógica de Controle não tiver sido modificado nesta etapa (por uma instrução de desvio), ele é incrementado para prosseguir para a próxima instrução.

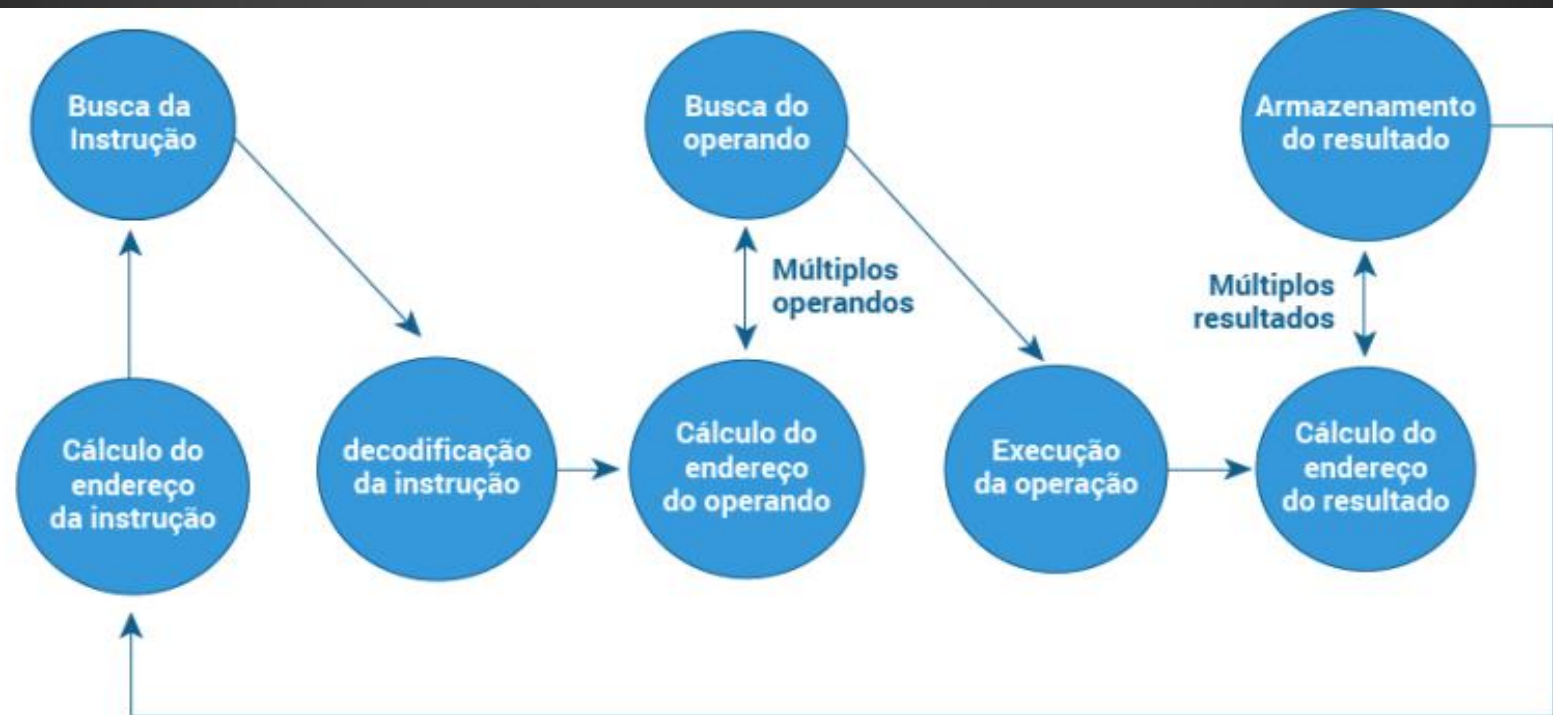
# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Concluído o ciclo de uma única instrução, ele é repetido para a execução da próxima instrução, mas, antes disso, a Unidade de Controle (Lógica de Controle) verifica se há a presença de um sinal de interrupção. Caso tenha ocorrido uma interrupção, ela deve ser atendida, fazendo com que a próxima instrução a ser executada seja a da rotina de tratamento de interrupção.



# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*



Instrução completada:

- Se houve interrupção vai para RTI
- Se não, vai para a próxima instrução

Legenda: CICLO DE INSTRUÇÃO DETALHADO

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- As etapas na parte de baixo do diagrama ocorrem exclusivamente dentro da CPU. Já as etapas na parte superior do diagrama podem envolver acesso externo. Ao menos a instrução estará sempre na memória principal. Os operandos e os resultados poderão estar dentro da CPU nos registradores, ou fora, na memória principal, ou em um periférico ainda mais lento. Criam-se então gargalos de desempenho para o sistema que precisarão ser tratados na implementação da CPU e do sistema.

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Muitas vezes o ciclo de instrução é representado em cinco etapas, forma comumente adotada.
  1. Busca da Instrução (Instruction Fetch);
  2. Decodificação da Instrução;
  3. Busca do(s) operando(s);
  4. Execução da instrução;
  5. Armazenamento do(s) resultado(s).

# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Essas cinco etapas correspondem às colunas do diagrama anterior. Na implementação de uma CPU, para que o ciclo de instrução seja executado adequadamente, os registradores são "zerados", isto é, levados a um estado inicial conhecido. Isso inclui o Contador de Programa, para que o ciclo de instrução comece a ser executado da primeira instrução, que pode corresponder ao Sistema Operacional, ou ao BIOS no Computador.

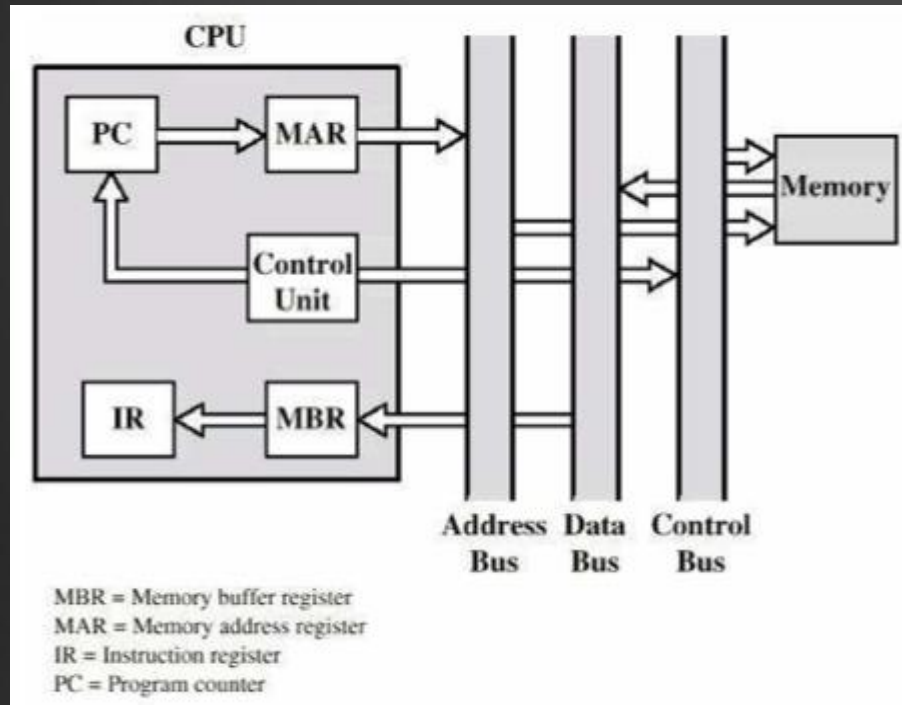
# CICLO DE INSTRUÇÃO DO *MICROPROCESSADOR*

---

- Antes de acompanhar a simulação de algumas instruções do Microprocessador, vamos recordar os registradores do Microprocessador e o Barramento do computador:



# CICLO DE INSTRUÇÃO DO MICROPROCESSADOR



**MAR** (Memory Address Register) - contém o endereço da posição de memória a ser lida / gravada

**PC** (Program Counter) contém o endereço da próxima instrução a ser executada

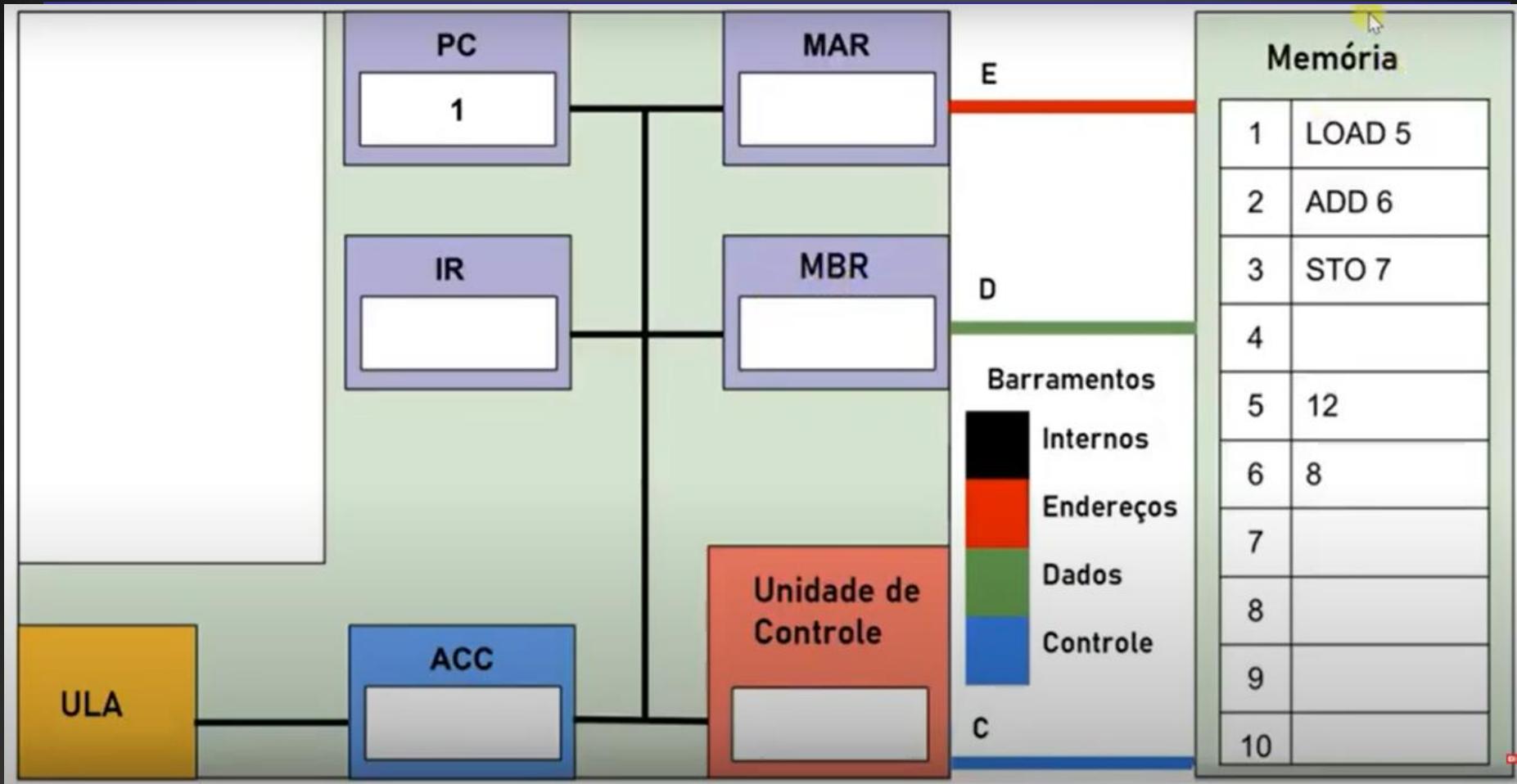
**MBR** - (Memory Buffer Register)- armazena o dado lido ou a ser gravado na memória

**IR** - Instruction Register

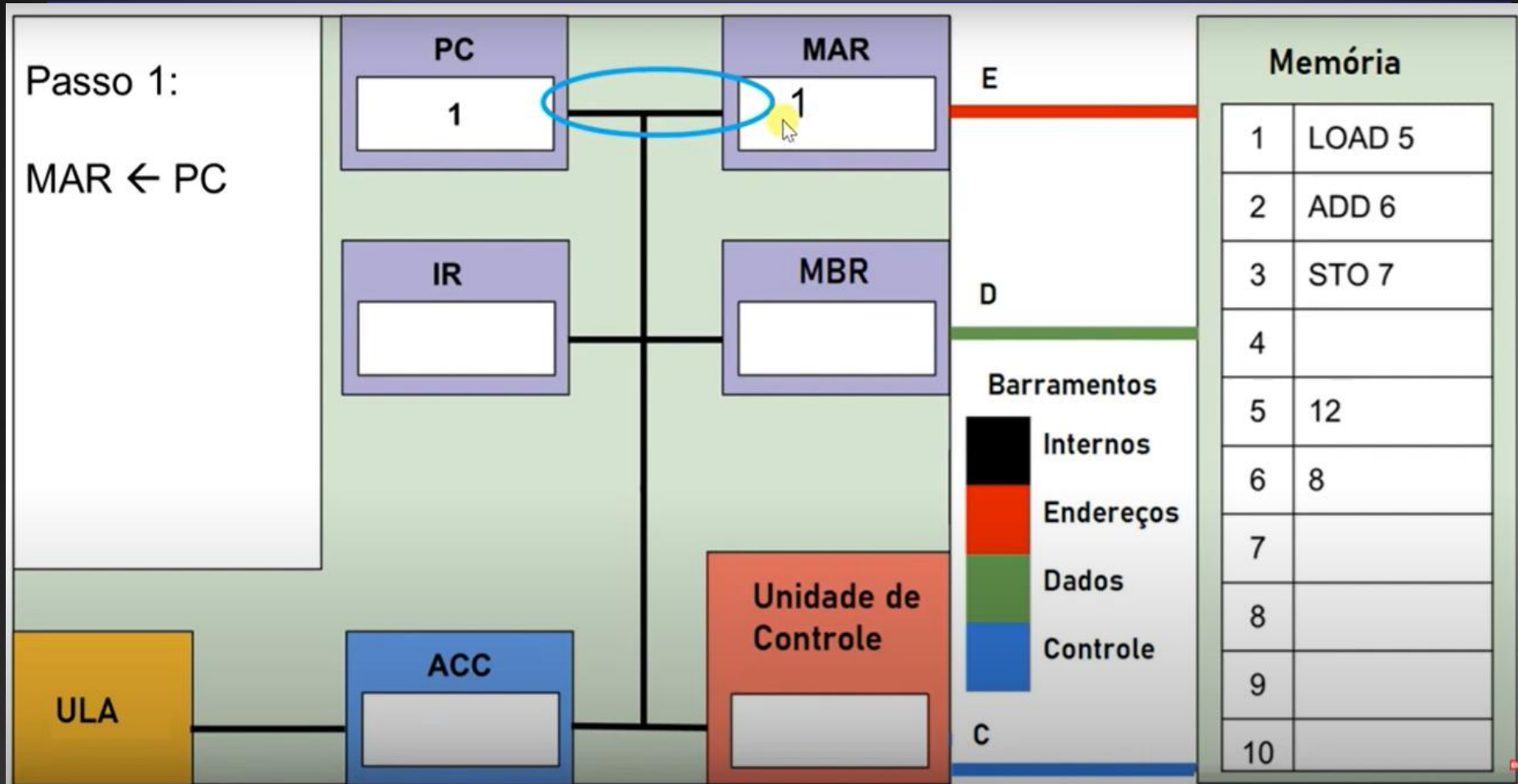
**AP** (Apontador da Pilha): endereço de memória do topo da memória de pilha.

**PSW** (Program Status Word): armazena informações sobre execução de instruções, como a ocorrência de overflow.

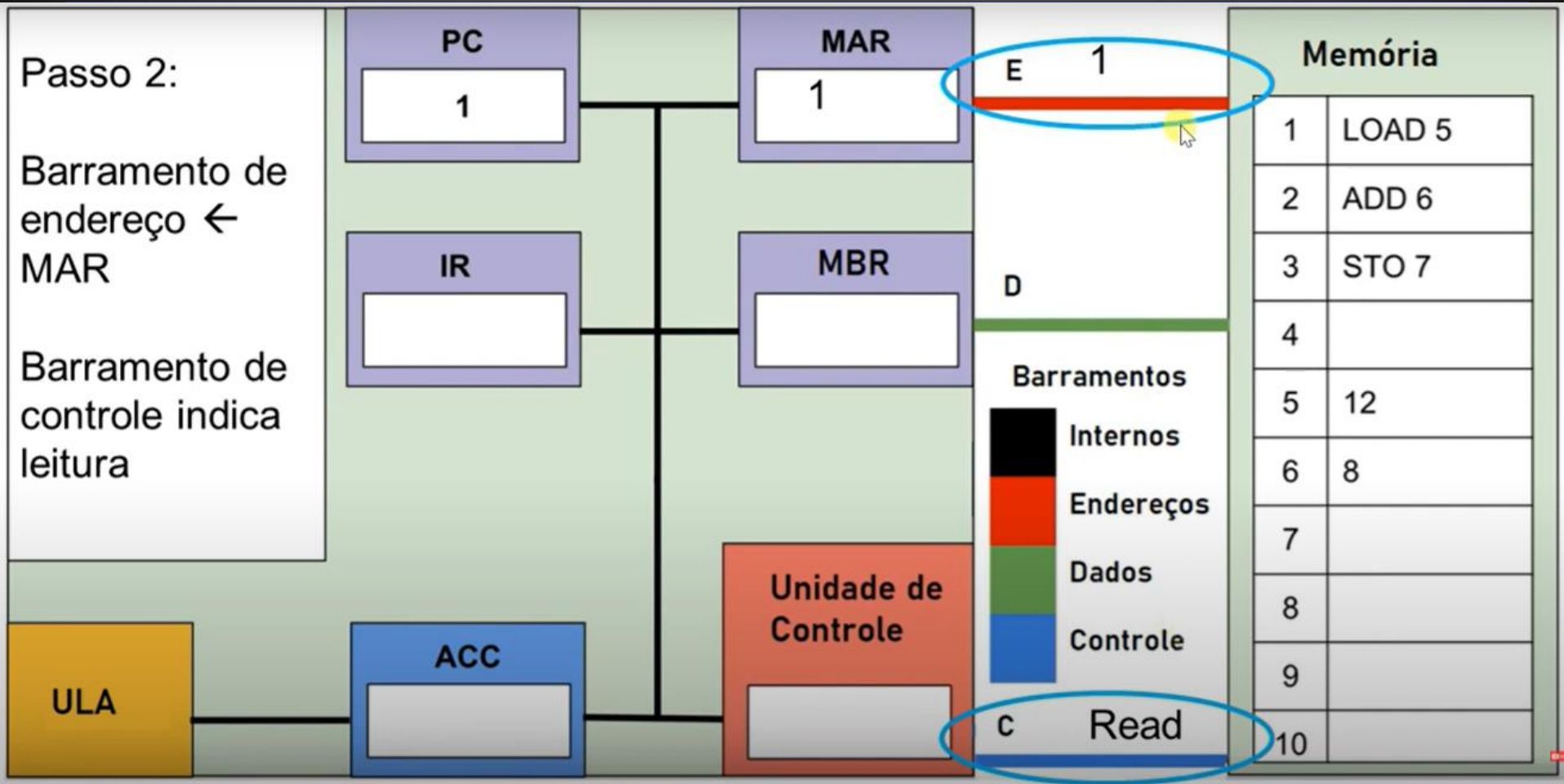
# CICLO DE BUSCA DO *MICROPROCESSADOR*



# CICLO DE BUSCA DO MICROPROCESSADOR

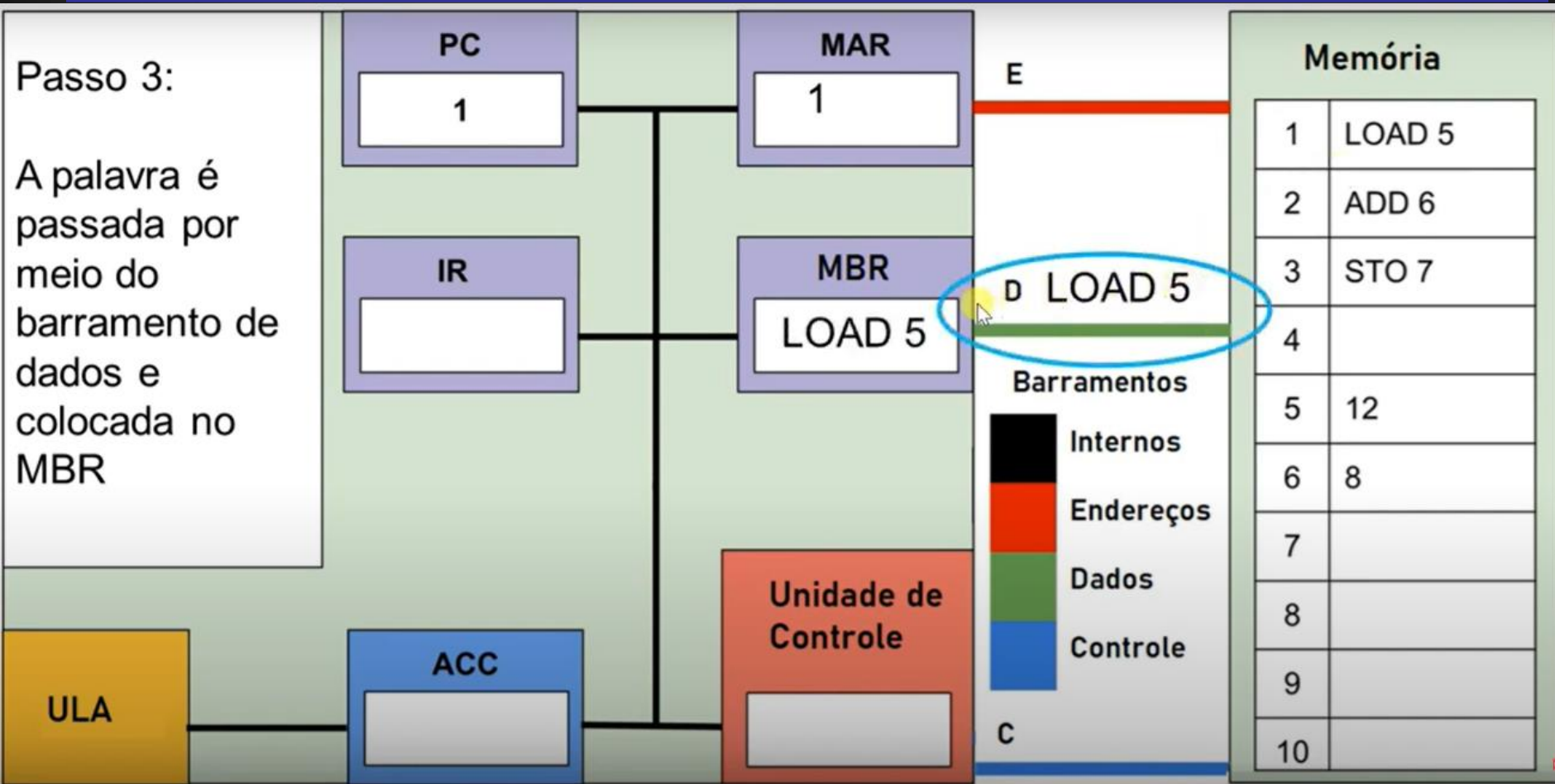


# CICLO DE BUSCA DO MICROPROCESSADOR

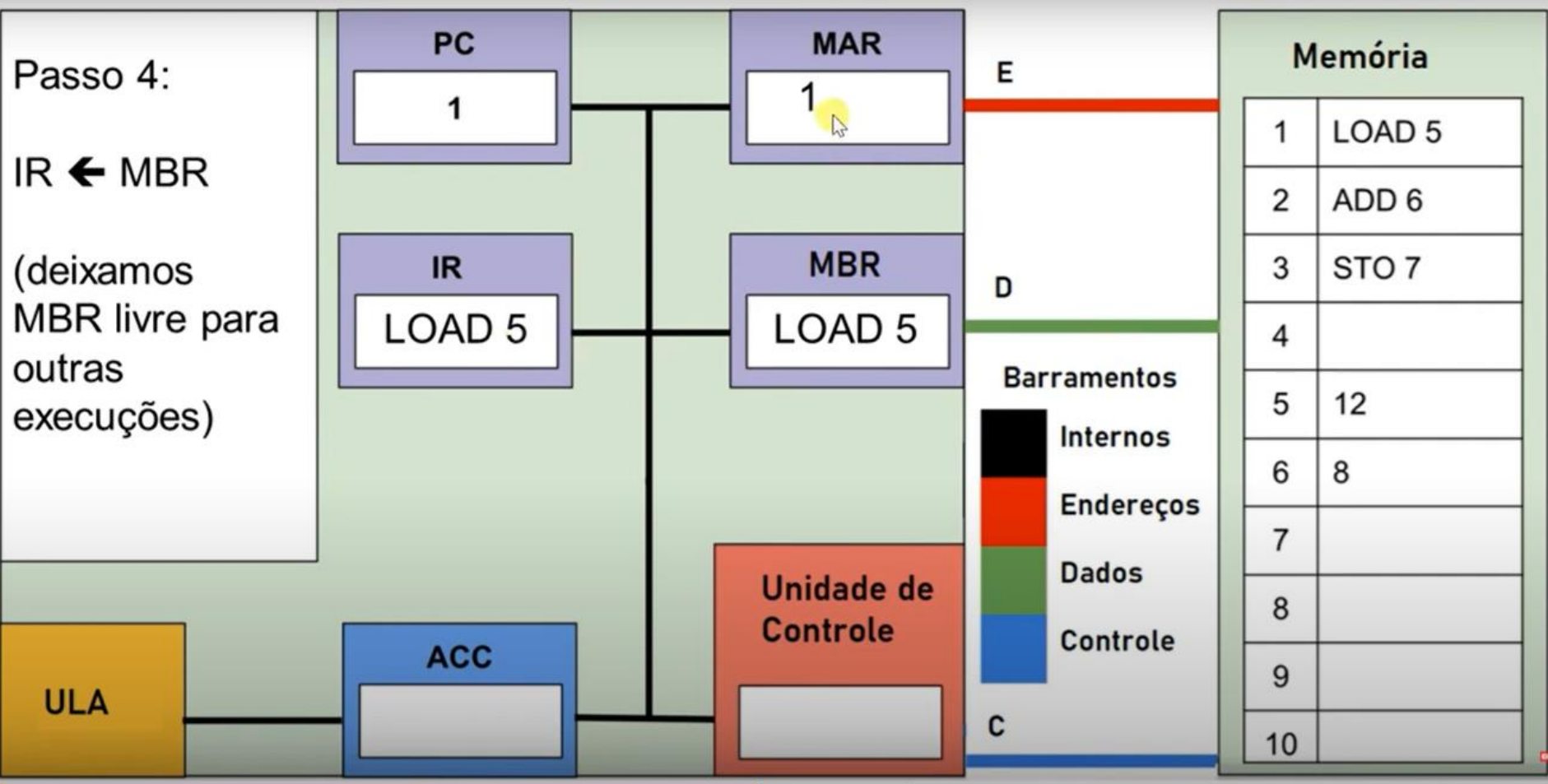




# CICLO DE BUSCA DO MICROPROCESSADOR

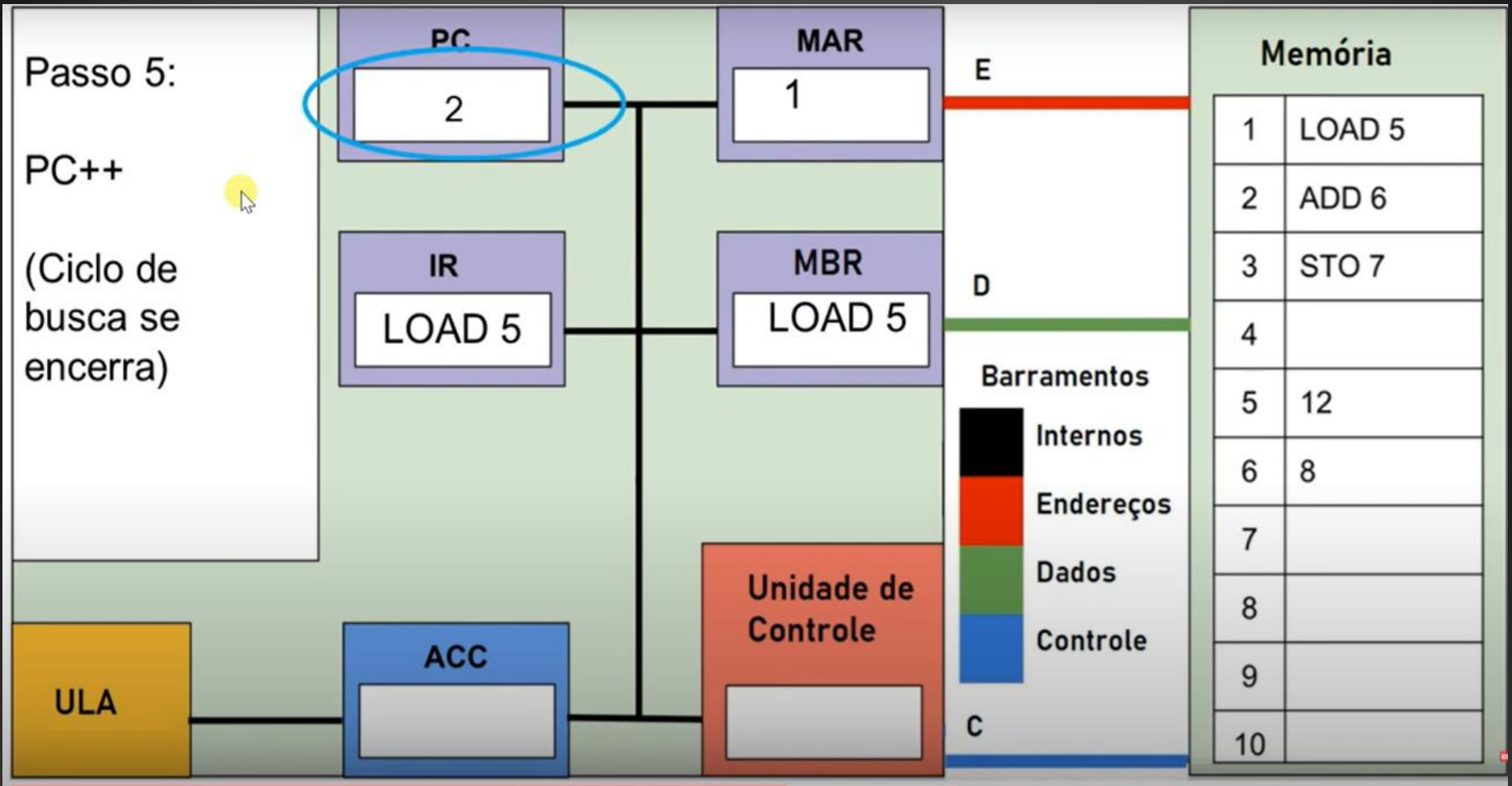


# CICLO DE BUSCA DO MICROPROCESSADOR

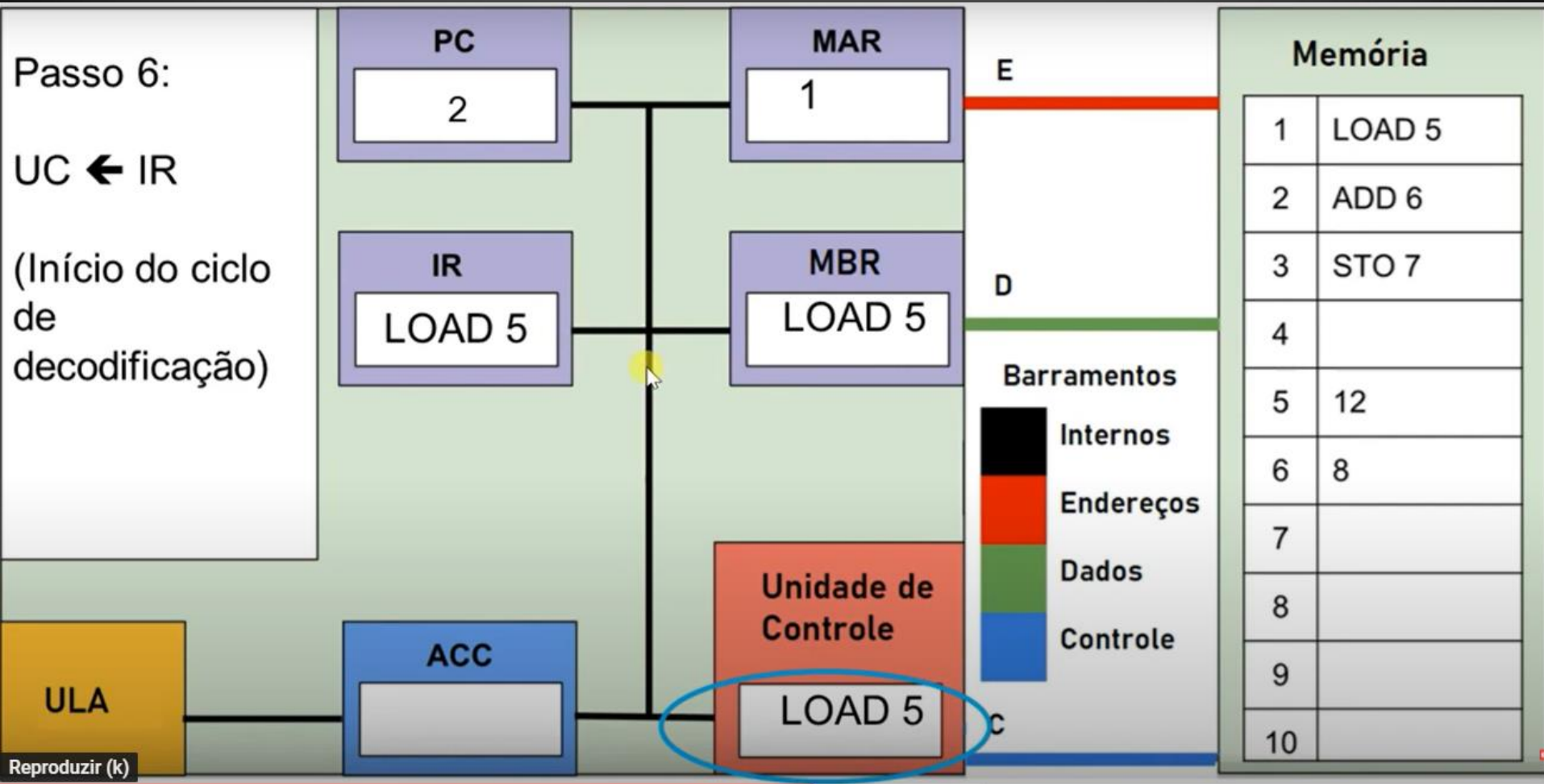




# CICLO DE BUSCA DO MICROPROCESSADOR



# CICLO DE DECODIFICAÇÃO DO MICROPROCESSADOR



# CICLO DE DECODIFICAÇÃO DO *MICROPROCESSADOR*

## PASSO 7

CPU identifica a instrução

00100110 ↑ 000000000000000000000000101

OPCODE

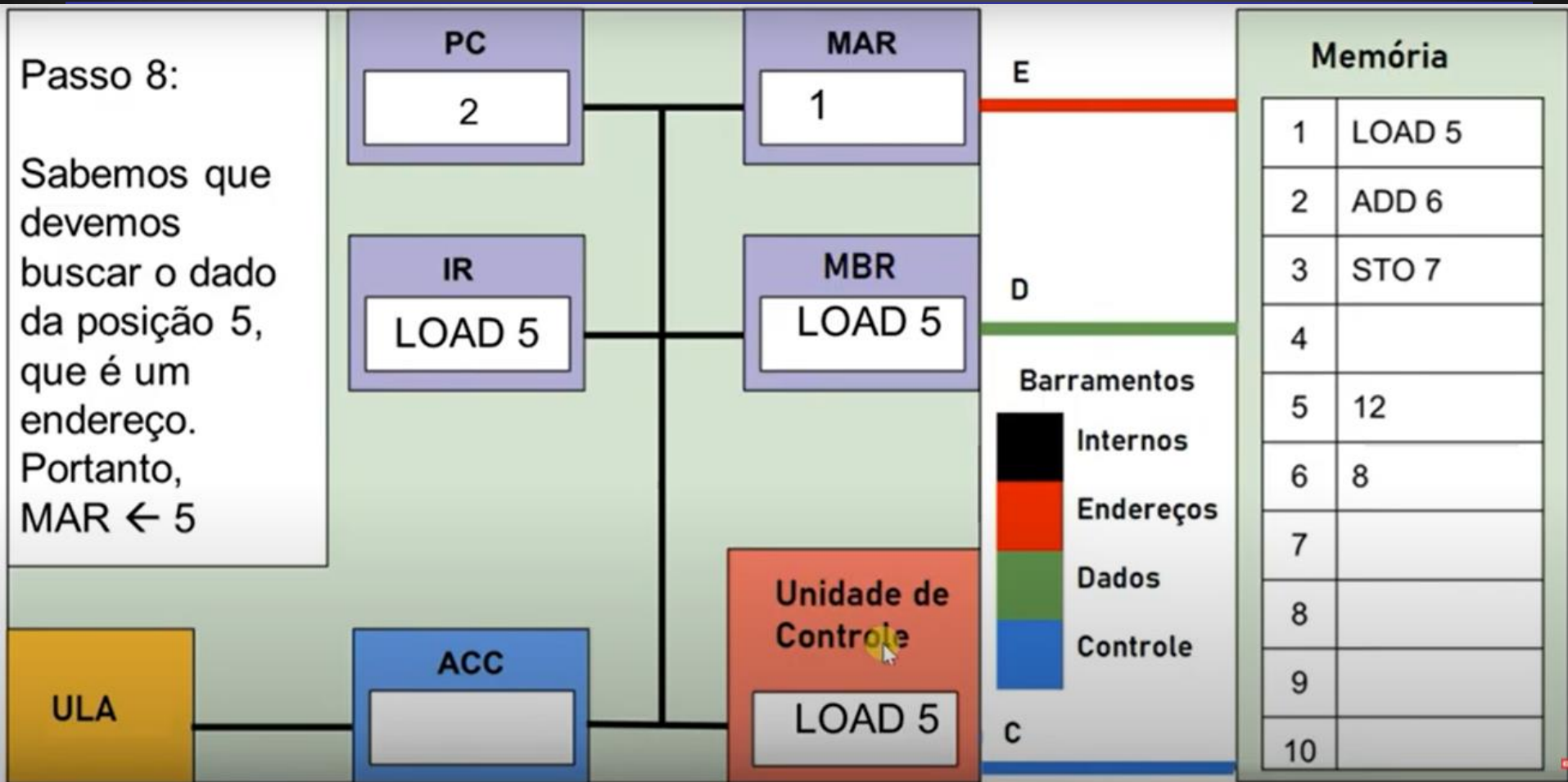
MODO DE ENDEREÇAMENTO

OPERANDO

Dado pronto para uso

Endereço

# CICLO DE DECODIFICAÇÃO DO MICROPROCESSADOR





# CICLO DE DECODIFICAÇÃO DO MICROPROCESSADOR

Passo 9:

$E \leftarrow 5$   
 $C \leftarrow \text{Read}$

PC  
2

MAR  
5

IR  
LOAD 5

MBR  
LOAD 5

E 5

D

Barramentos

Internos  
Endereços  
Dados  
Controle

Unidade de Controle

c Read

Memória

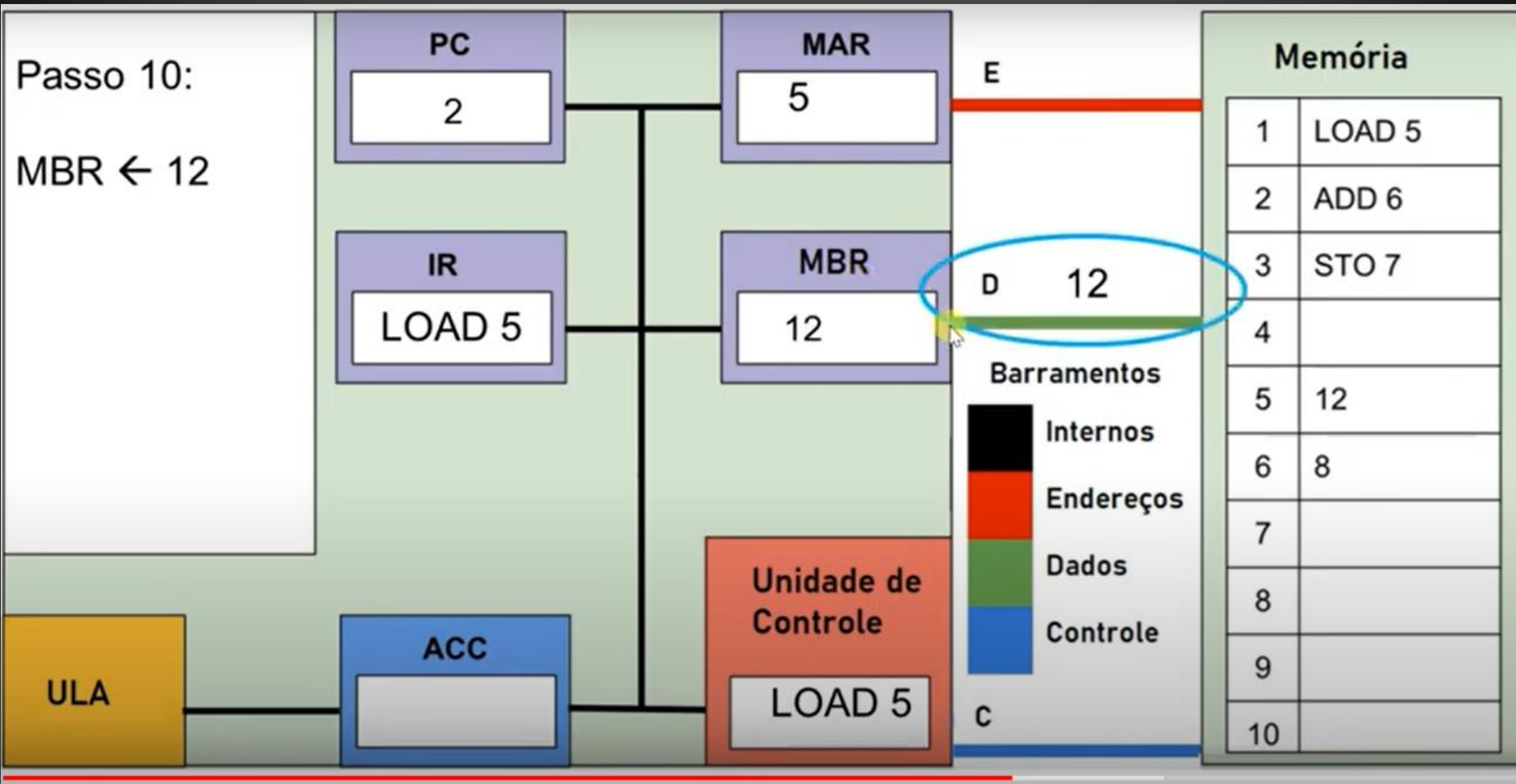
1	LOAD 5
2	ADD 6
3	STO 7
4	
5	12
6	8
7	
8	
9	
10	

ULA

ACC

LOAD 5

# CICLO DE DECODIFICAÇÃO DO MICROPROCESSADOR



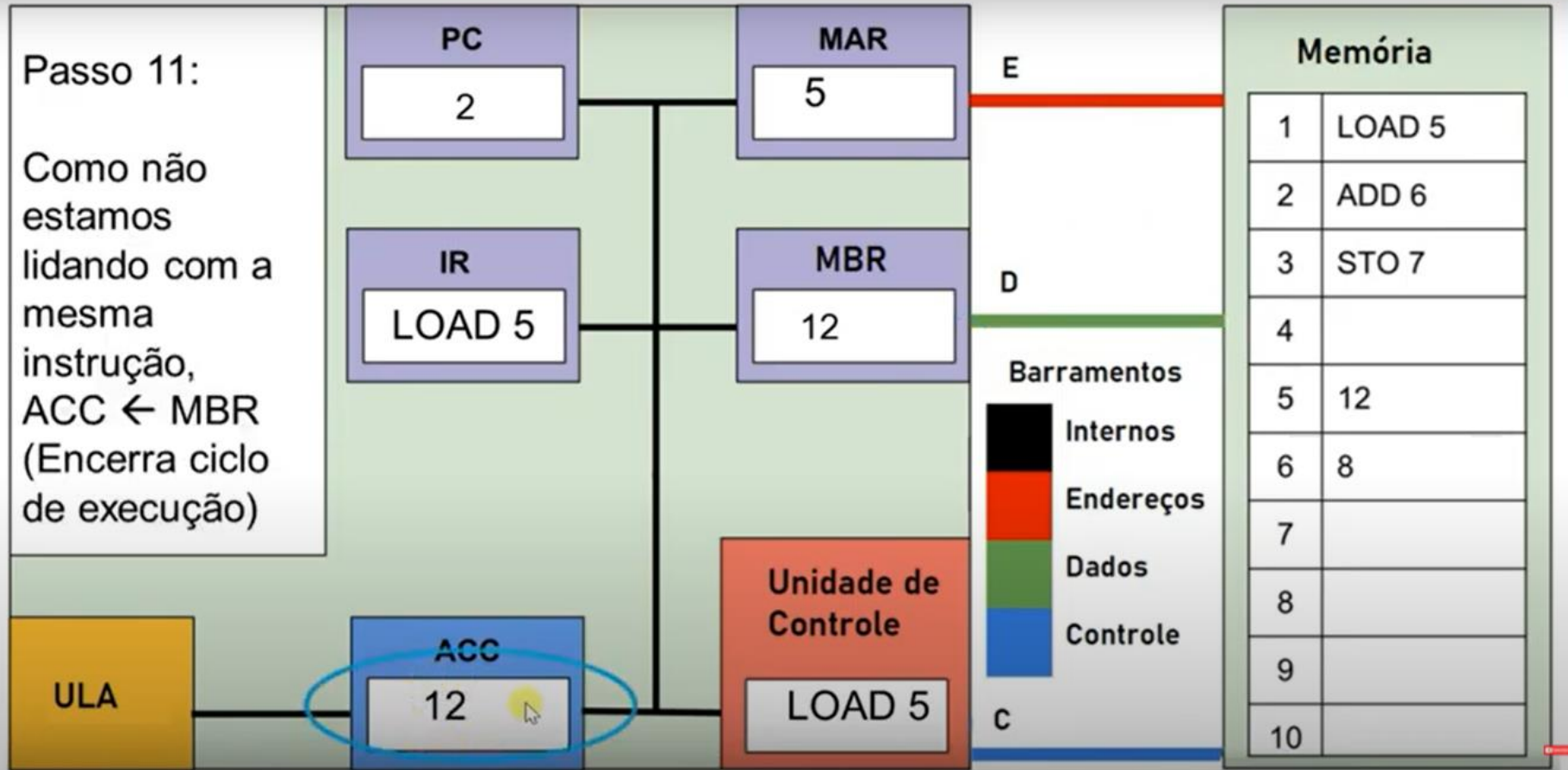


# CICLO DE DECODIFICAÇÃO DO MICROPROCESSADOR

Ciclo de Busca, Decodificação e Execução de Instruções pela CPU - em Detalhes - PC, IR, MAR e MBR

Passo 11:

Como não estamos lidando com a mesma instrução,  $ACC \leftarrow MBR$  (Encerra ciclo de execução)

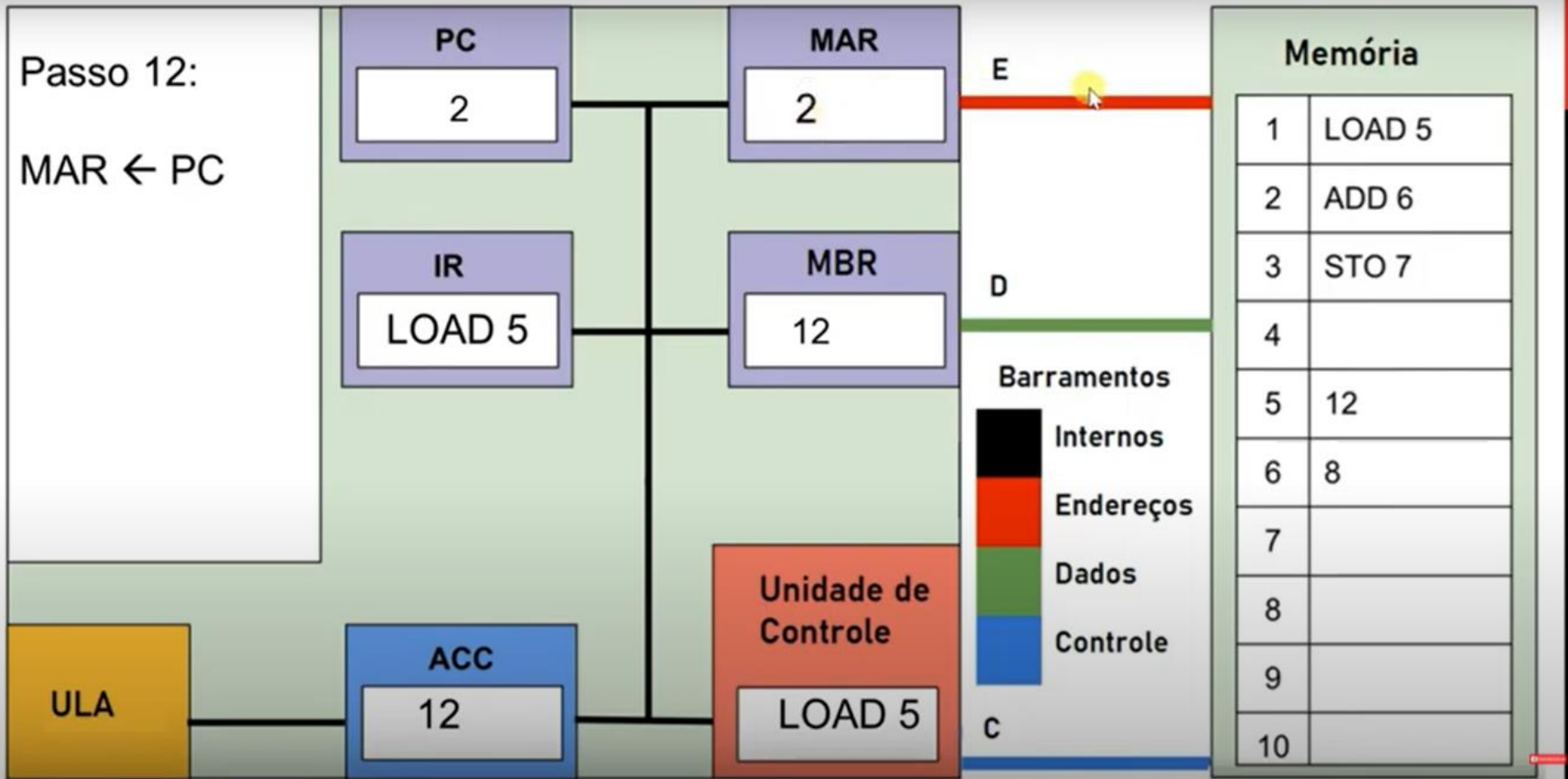


14:16 / 21:03

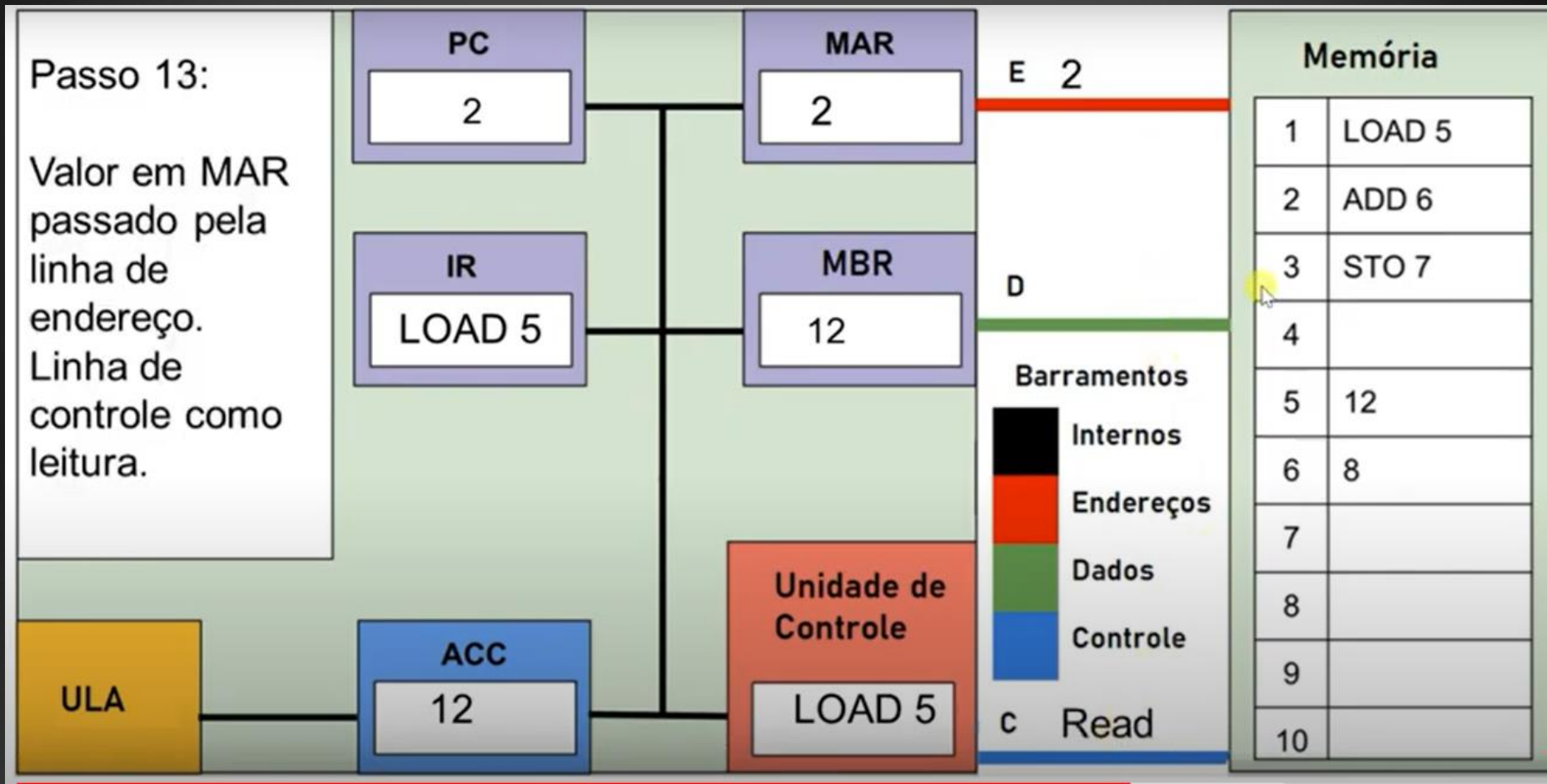
Role para ver detalhes

# CICLO DE BUSCA DO *MICROPROCESSADOR*

Ciclo de Busca, Decodificação e Execução de Instruções pela CPU - em Detalhes - PC, IR, MAR e MBR



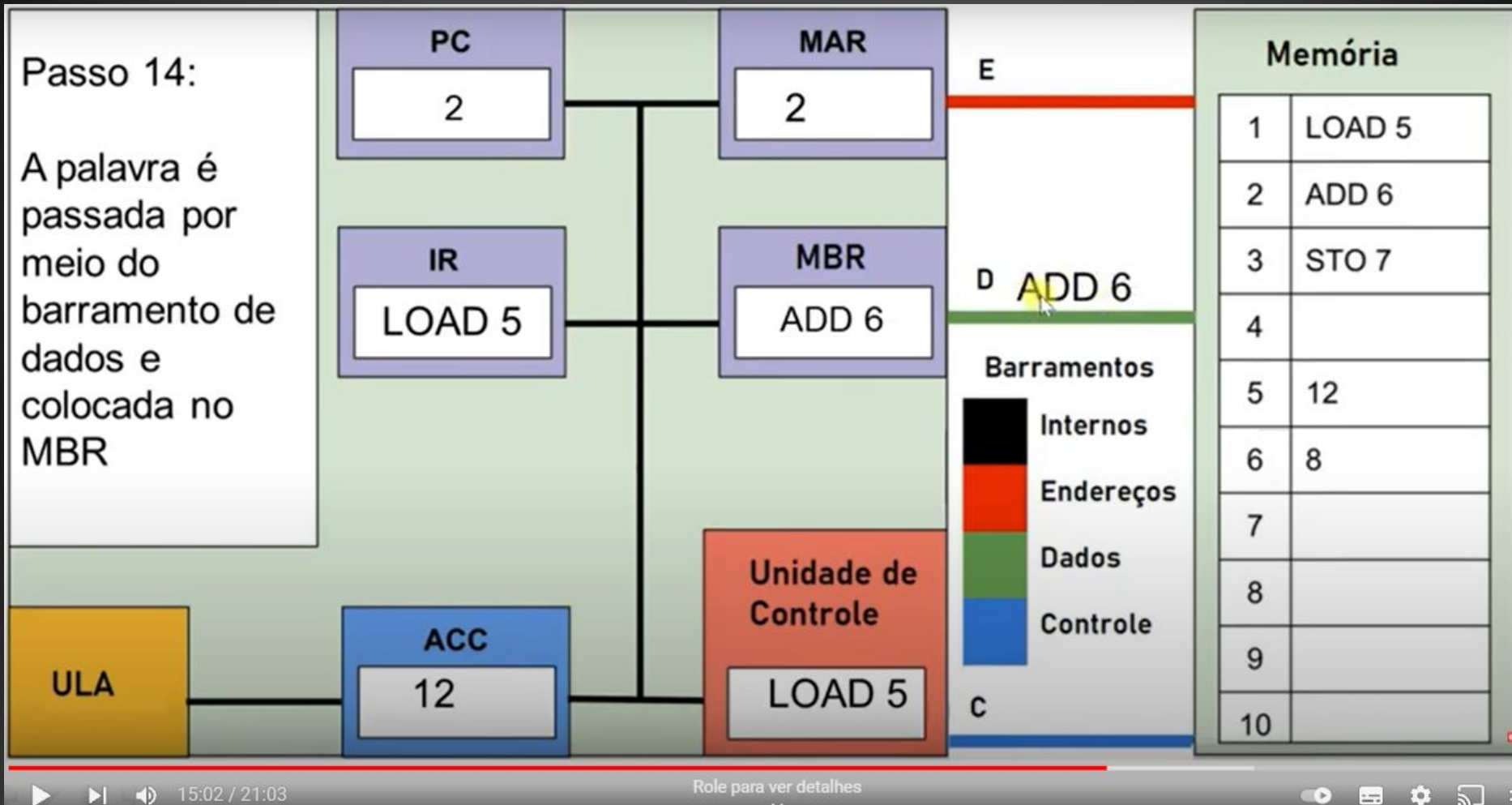
# CICLO DE BUSCA DO MICROPROCESSADOR



# CICLO DE BUSCA DO MICROPROCESSADOR

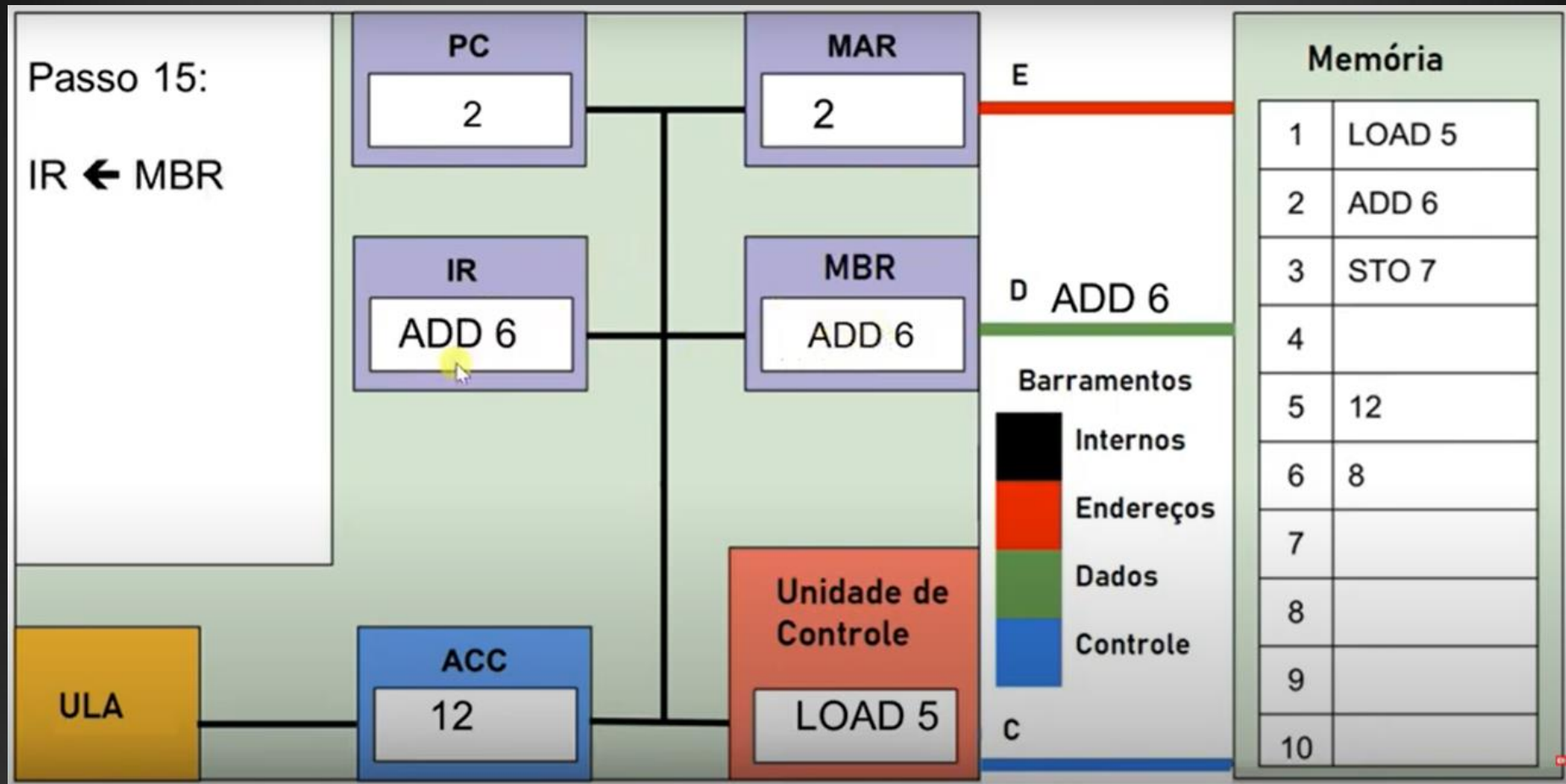
Passo 14:

A palavra é passada por meio do barramento de dados e colocada no MBR

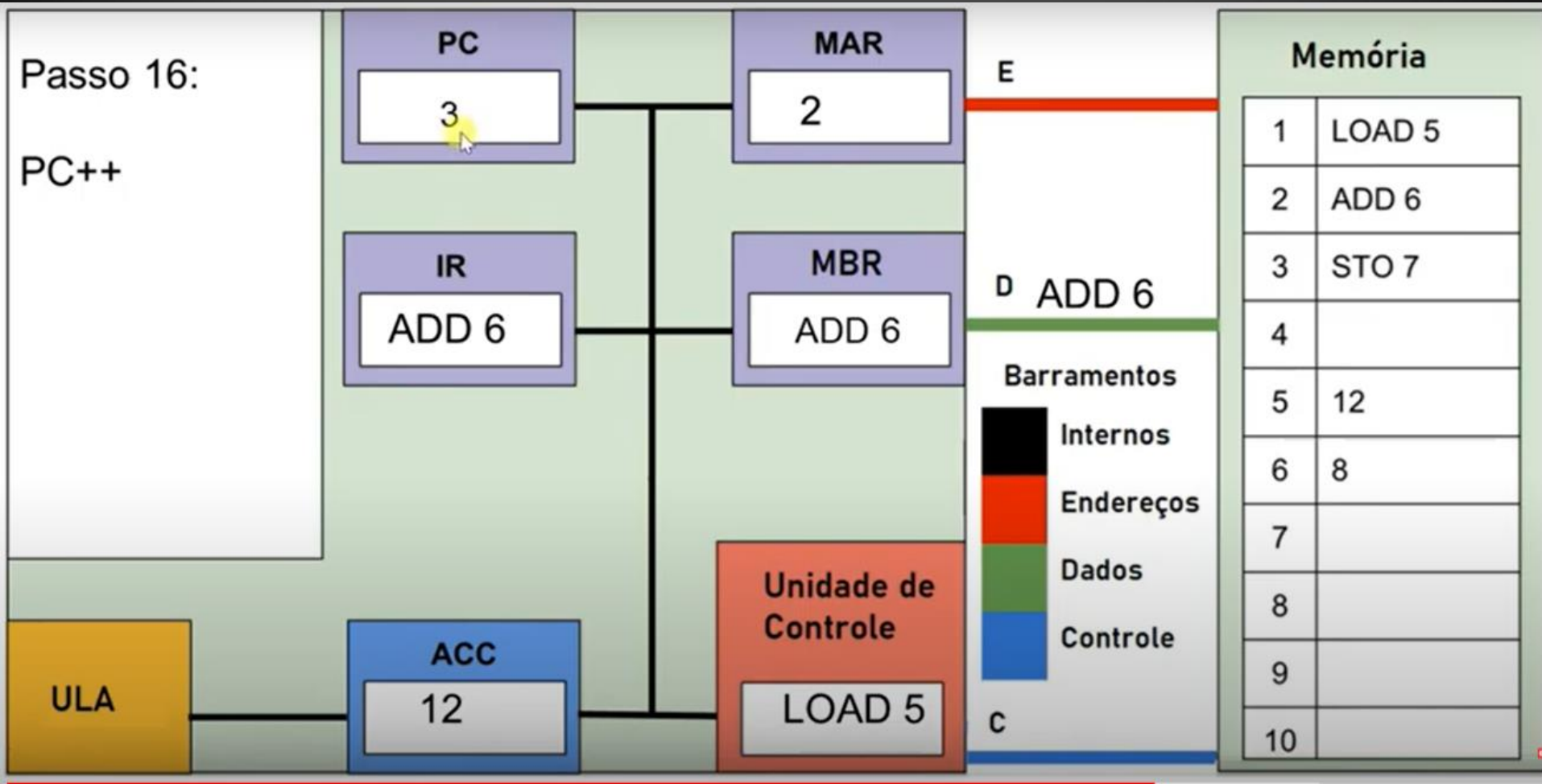




# CICLO DE BUSCA DO *MICROPROCESSADOR*



## ENCERRAMENTO DO 2º CICLO DE BUSCA

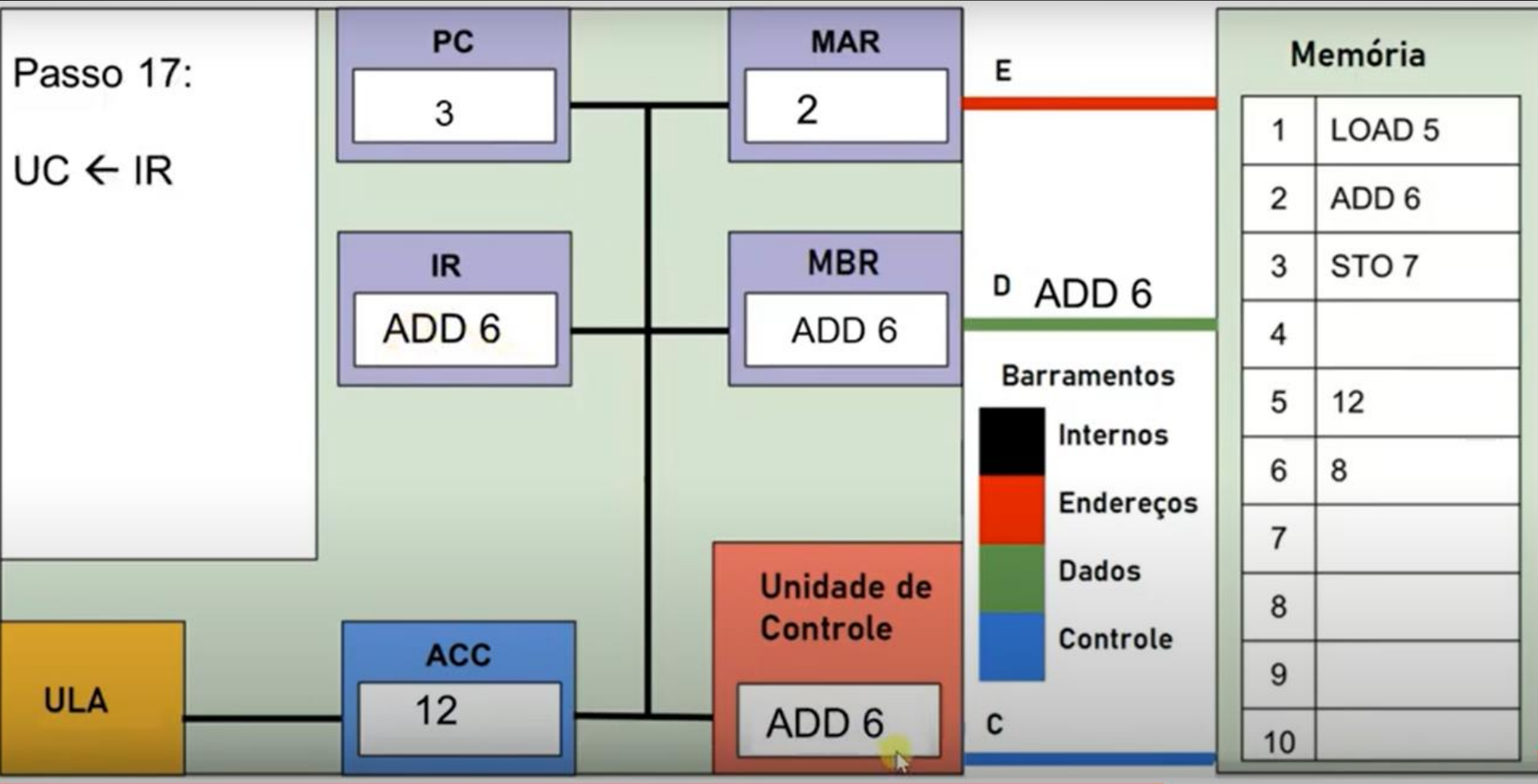




# ENCERRAMENTO DO 2º CICLO DE BUSCA

Passo 17:

UC ← IR



# CICLO DE EXECUÇÃO

## PASSO 18

CPU identifica a instrução

001011101 00000000000000000000110

OPCODE

MODO DE ENDEREÇAMENTO

OPERANDO

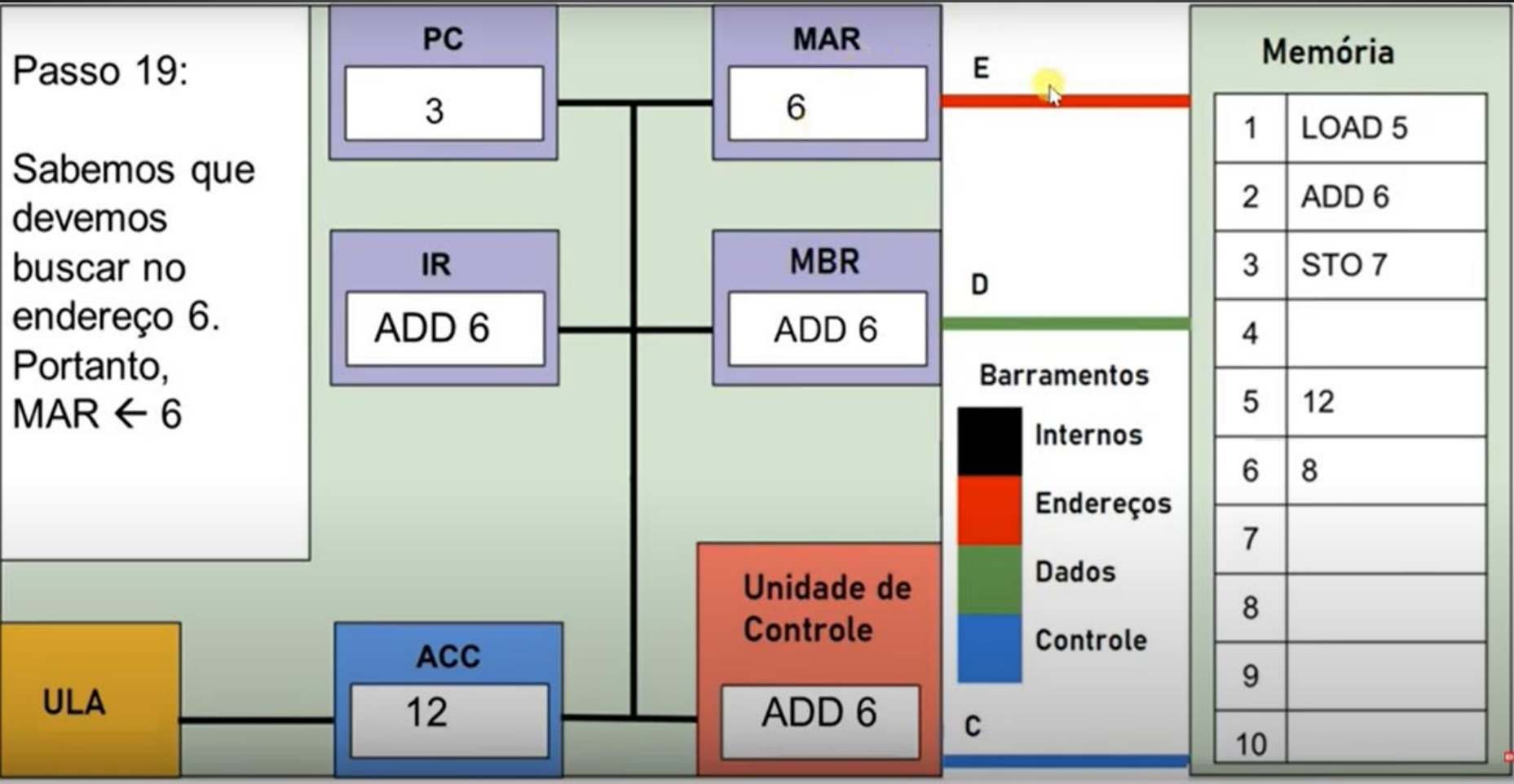
Dado pronto para uso

Endereço

# CICLO DE EXECUÇÃO

Passo 19:

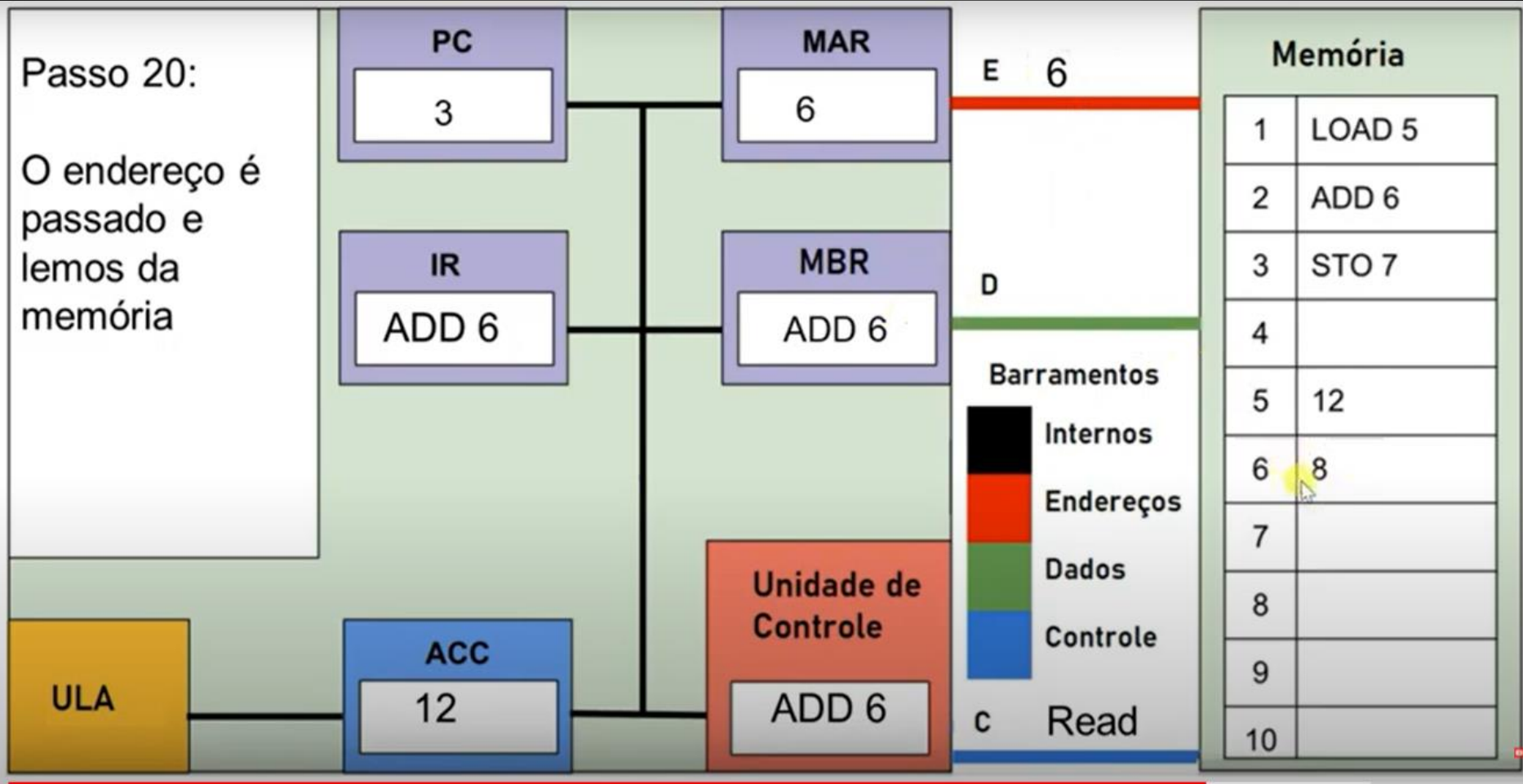
Sabemos que devemos buscar no endereço 6. Portanto,  $MAR \leftarrow 6$



# CICLO DE EXECUÇÃO

Passo 20:

O endereço é passado e lemos da memória

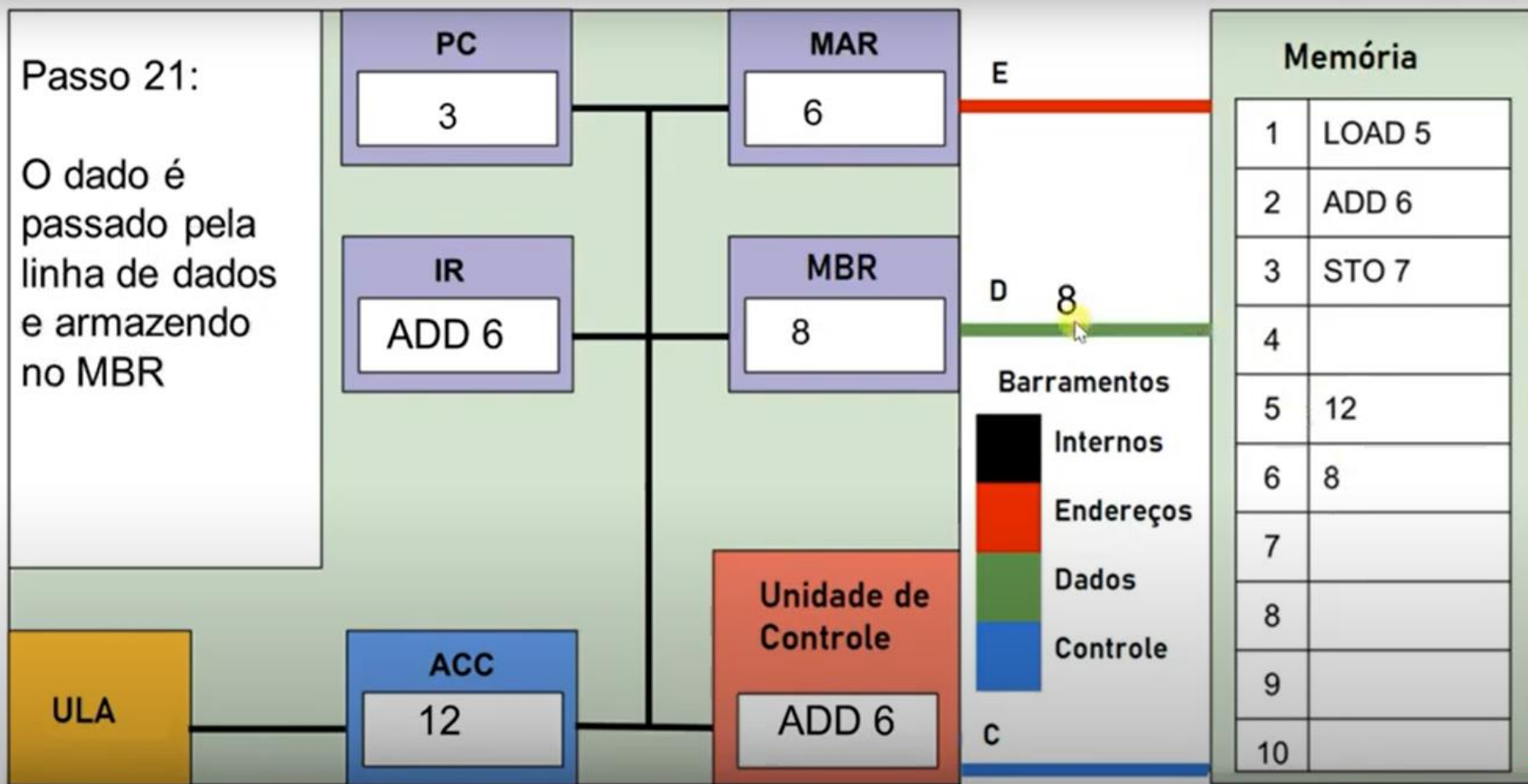




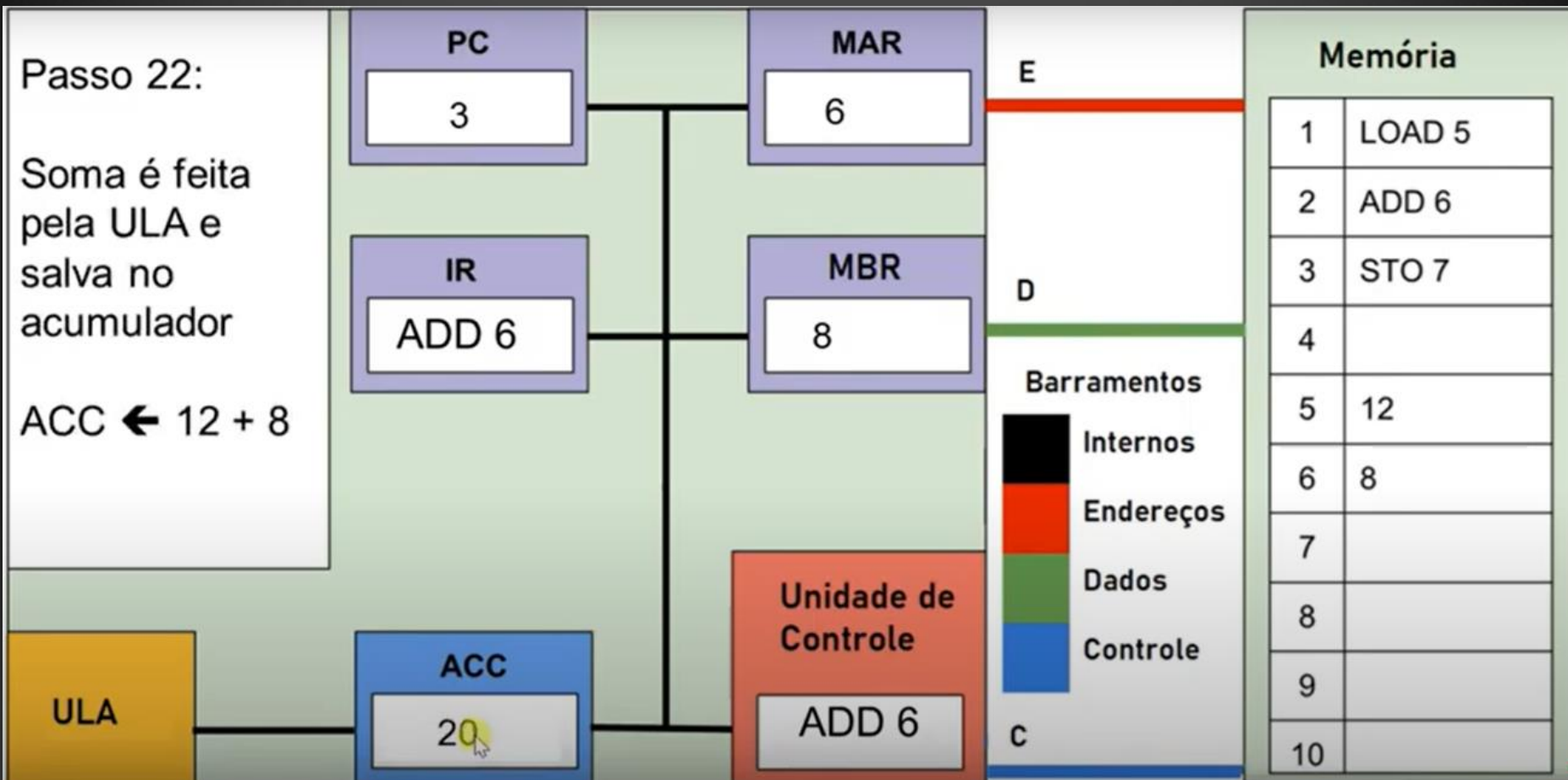
# CICLO DE EXECUÇÃO

Passo 21:

O dado é passado pela linha de dados e armazenado no MBR

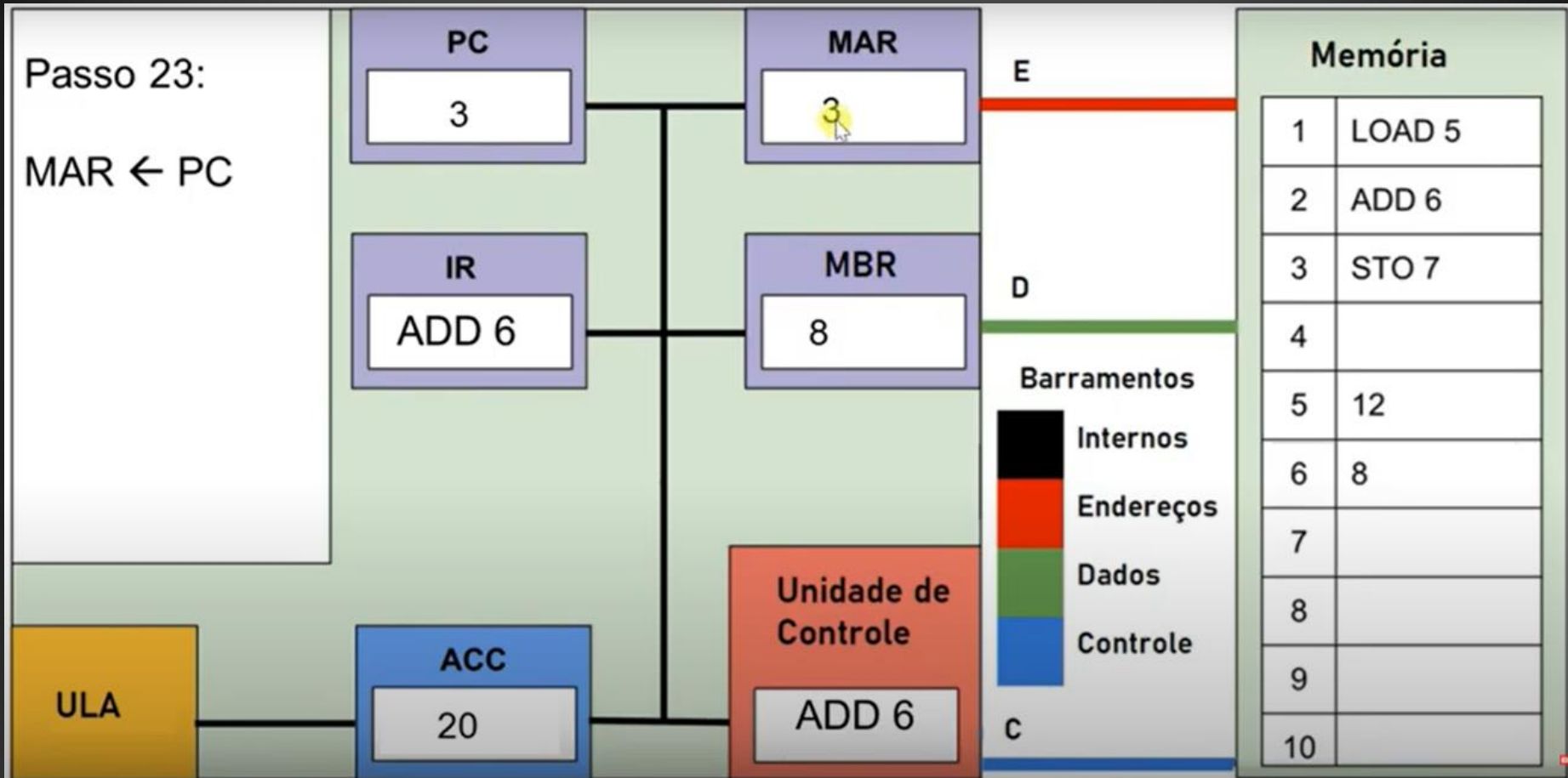


# CICLO DE EXECUÇÃO





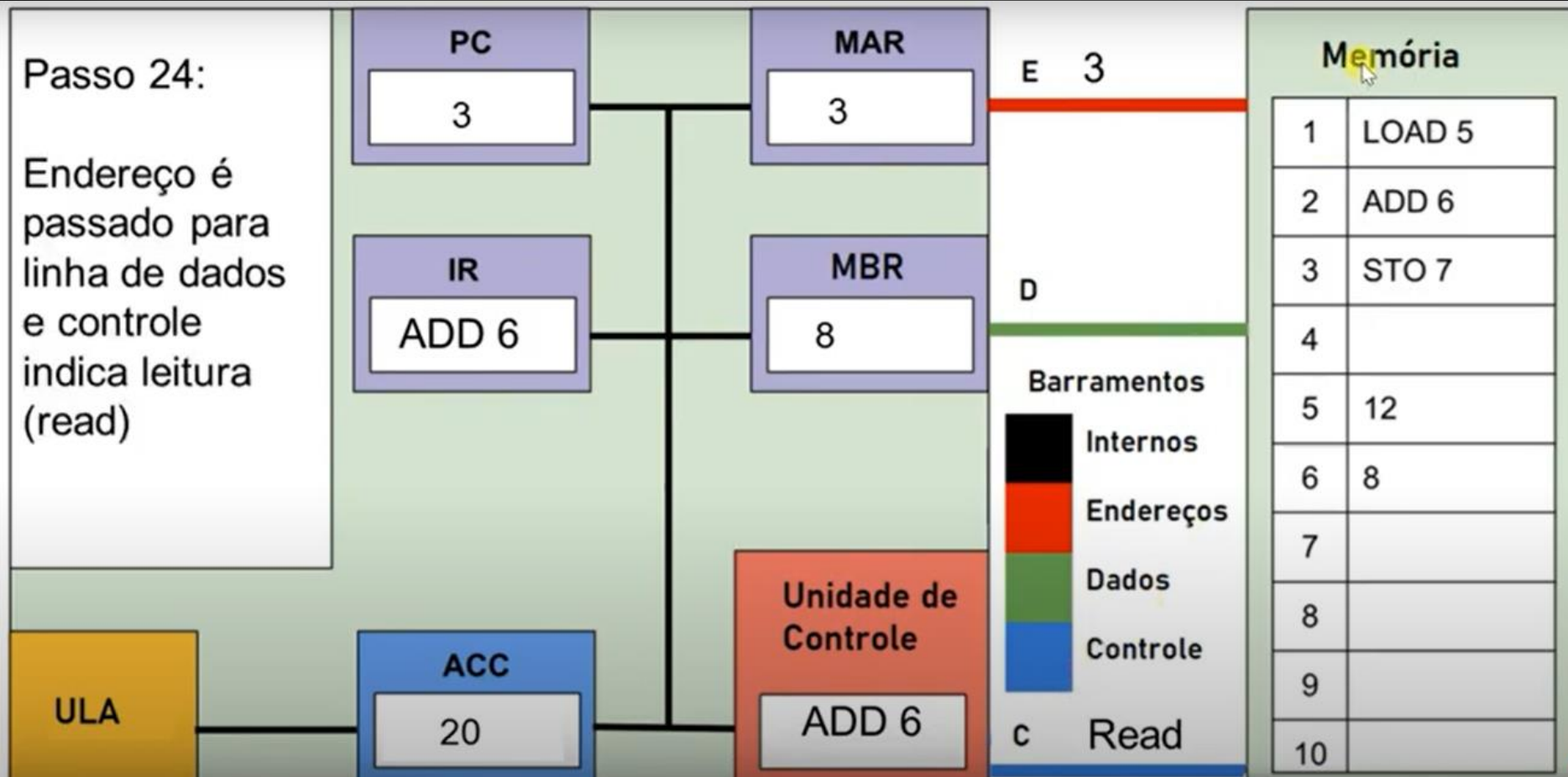
# NOVO CICLO DE BUSCA



# NOVO CICLO DE BUSCA

Passo 24:

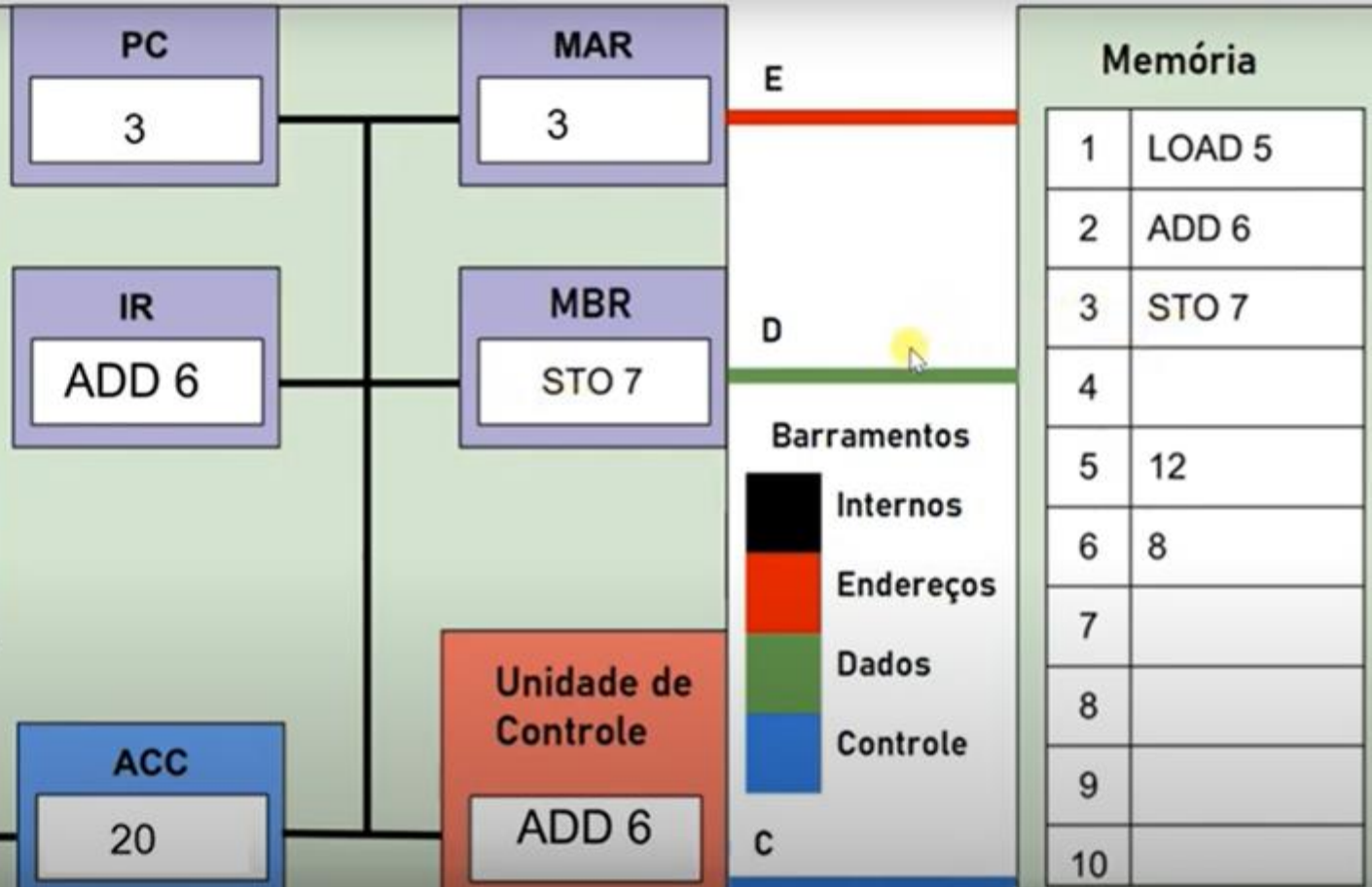
Endereço é passado para linha de dados e controle indica leitura (read)



# NOVO CICLO DE EXECUÇÃO

Passo 25:

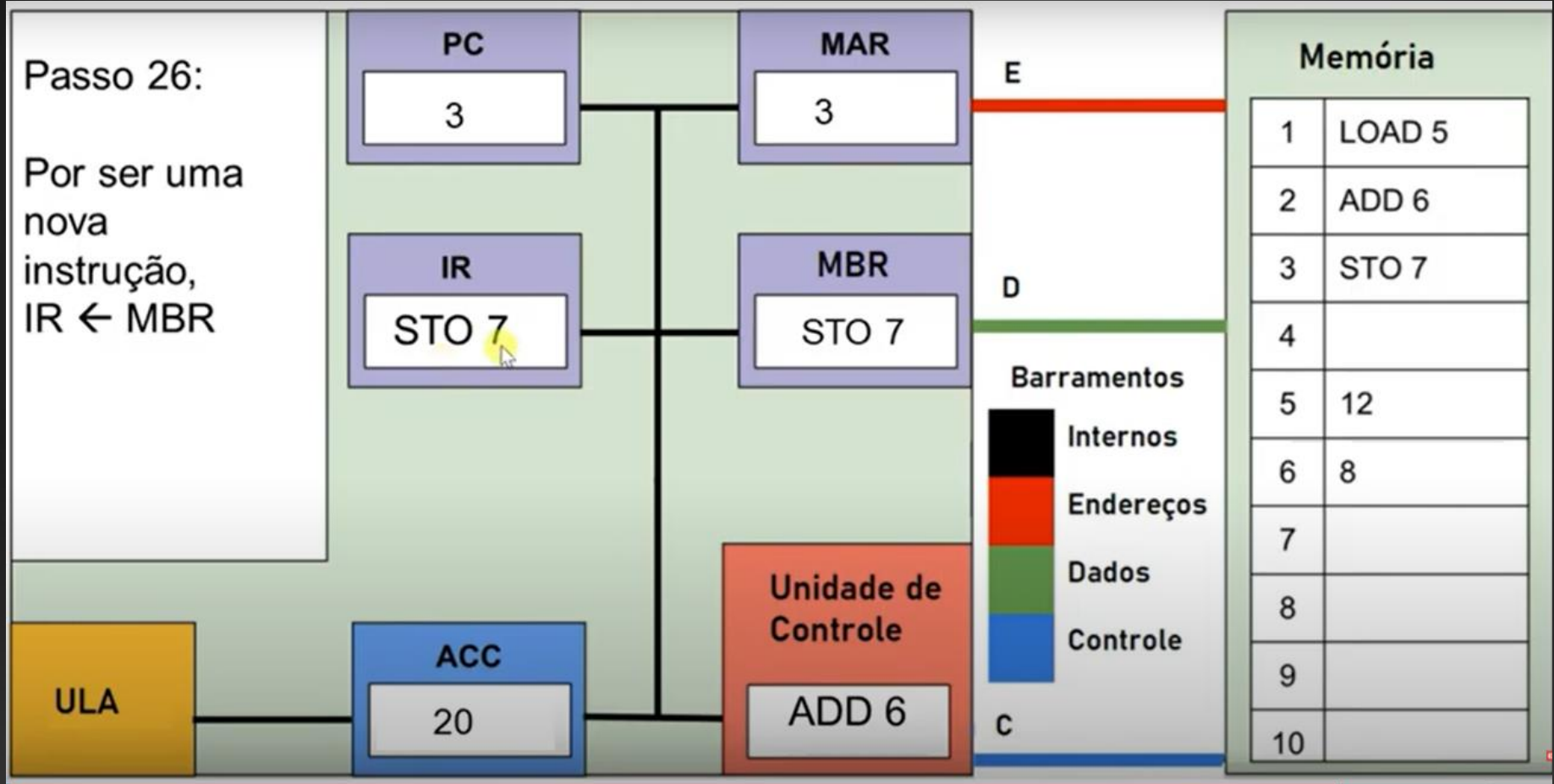
Instrução é passada pela linha de dados e armazenada em MBR



# NOVO CICLO DE EXECUÇÃO

Passo 26:

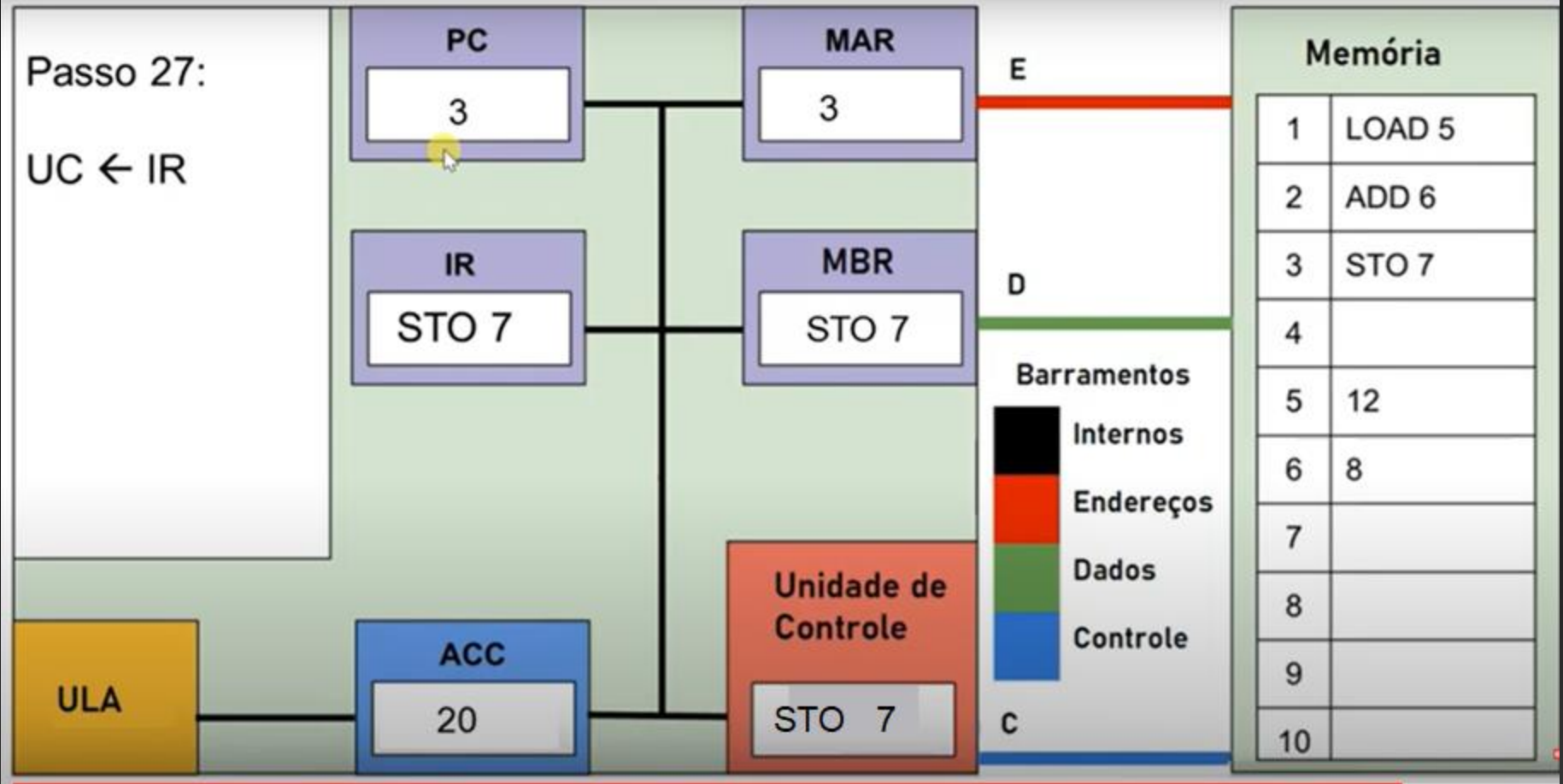
Por ser uma nova instrução,  
 $IR \leftarrow MBR$





# NOVO CICLO DE EXECUÇÃO

Ciclo de Busca, Decodificação e Execução de Instruções pela CPU - em Detalhes - PC, IR, MAR e MBR





# NOVO CICLO DE EXECUÇÃO

## PASSO 28

CPU identifica a instrução

10101010 1 000000000000000000000000111

OPCODE

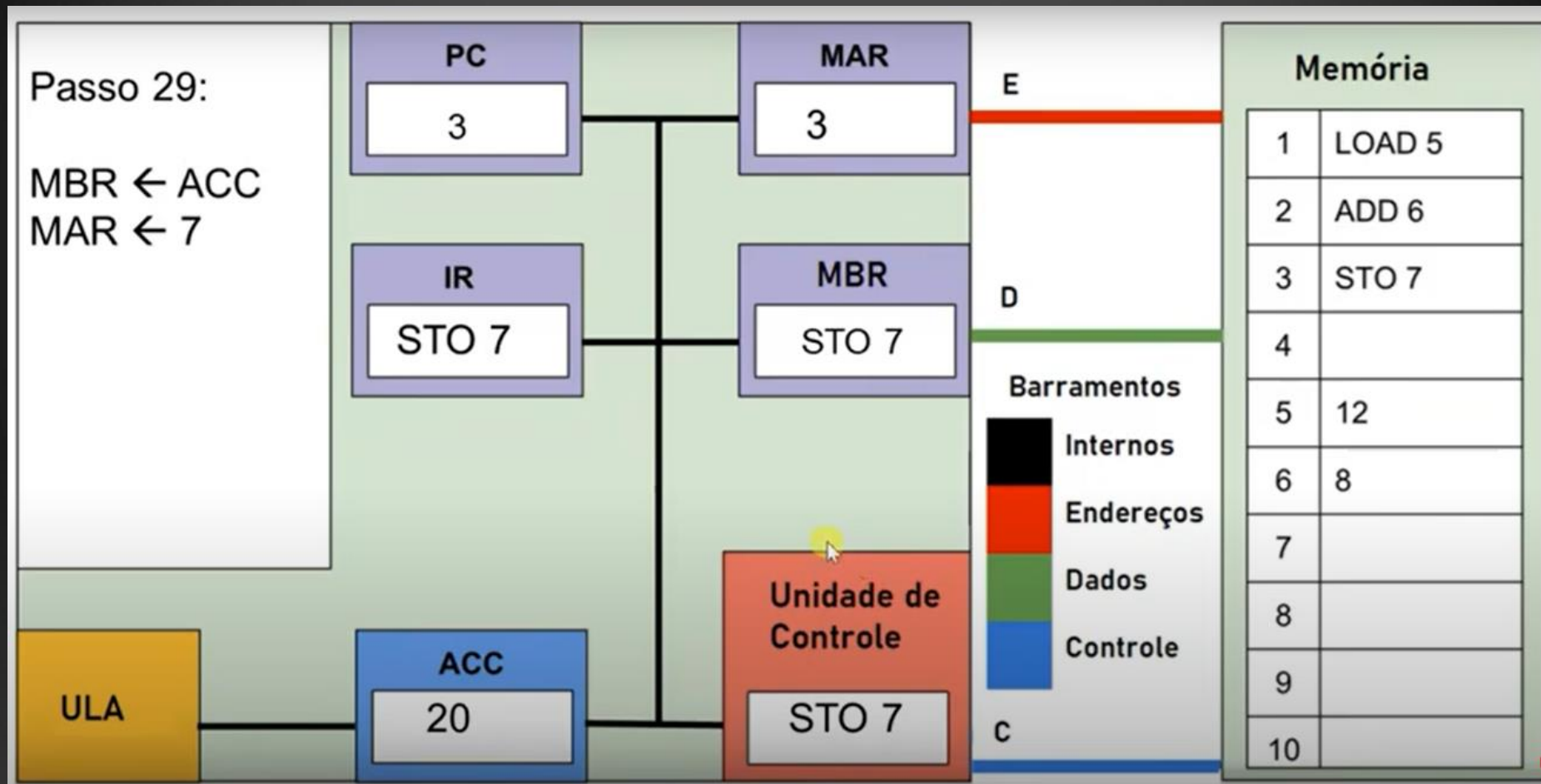
MODO DE ENDEREÇAMENTO

OPERANDO

Dado pronto para uso

Endereço

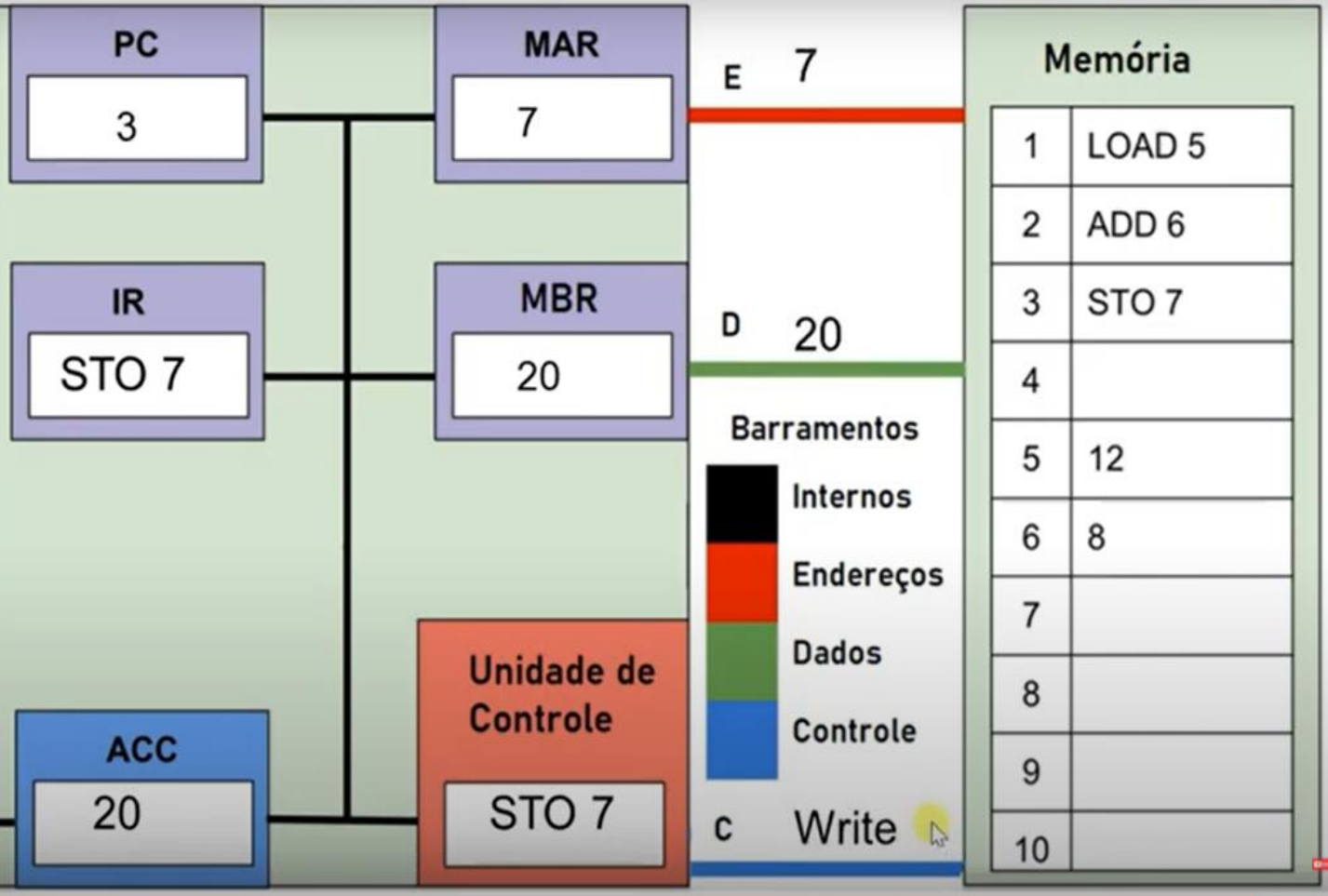
# NOVO CICLO DE EXECUÇÃO



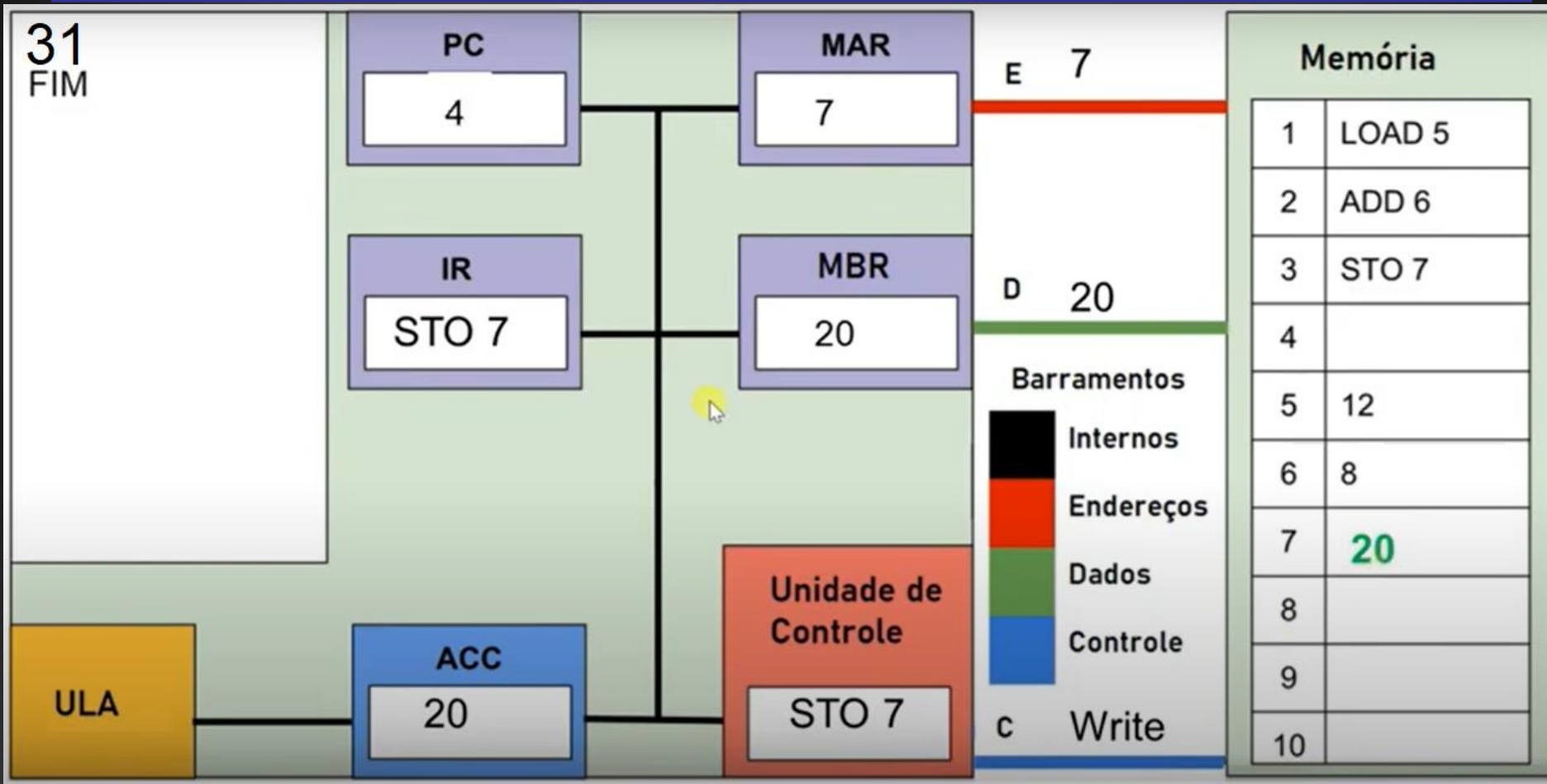
# NOVO CICLO DE EXECUÇÃO

Passo 30:

Conteúdo de MAR passado pela linha de endereço;  
Conteúdo de MBR passado pela linha de dados;  
Controle de escrita.



# NOVO CICLO DE EXECUÇÃO



# CICLOS DO MICROPROCESSADOR

- <https://www.youtube.com/watch?v=kEhztbcUBcg&t=19s>

**CICLO DE  
BUSCA E  
EXECUÇÃO**

**PROF. DR. OLIBÁRIO NETO**