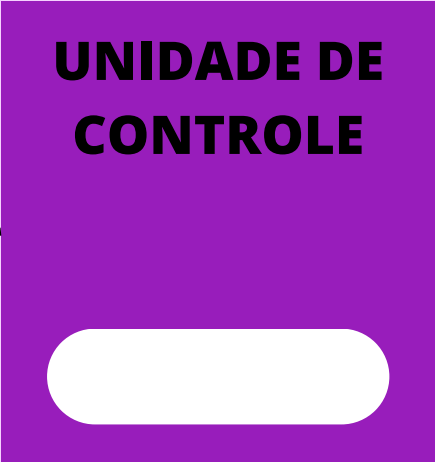
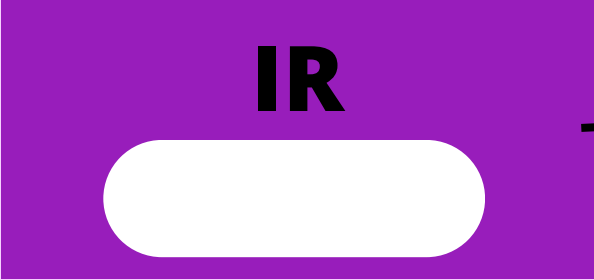
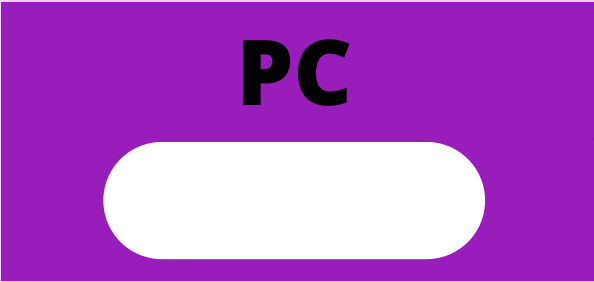


# Ciclo de máquina

Passo a passo de busca e  
execução.



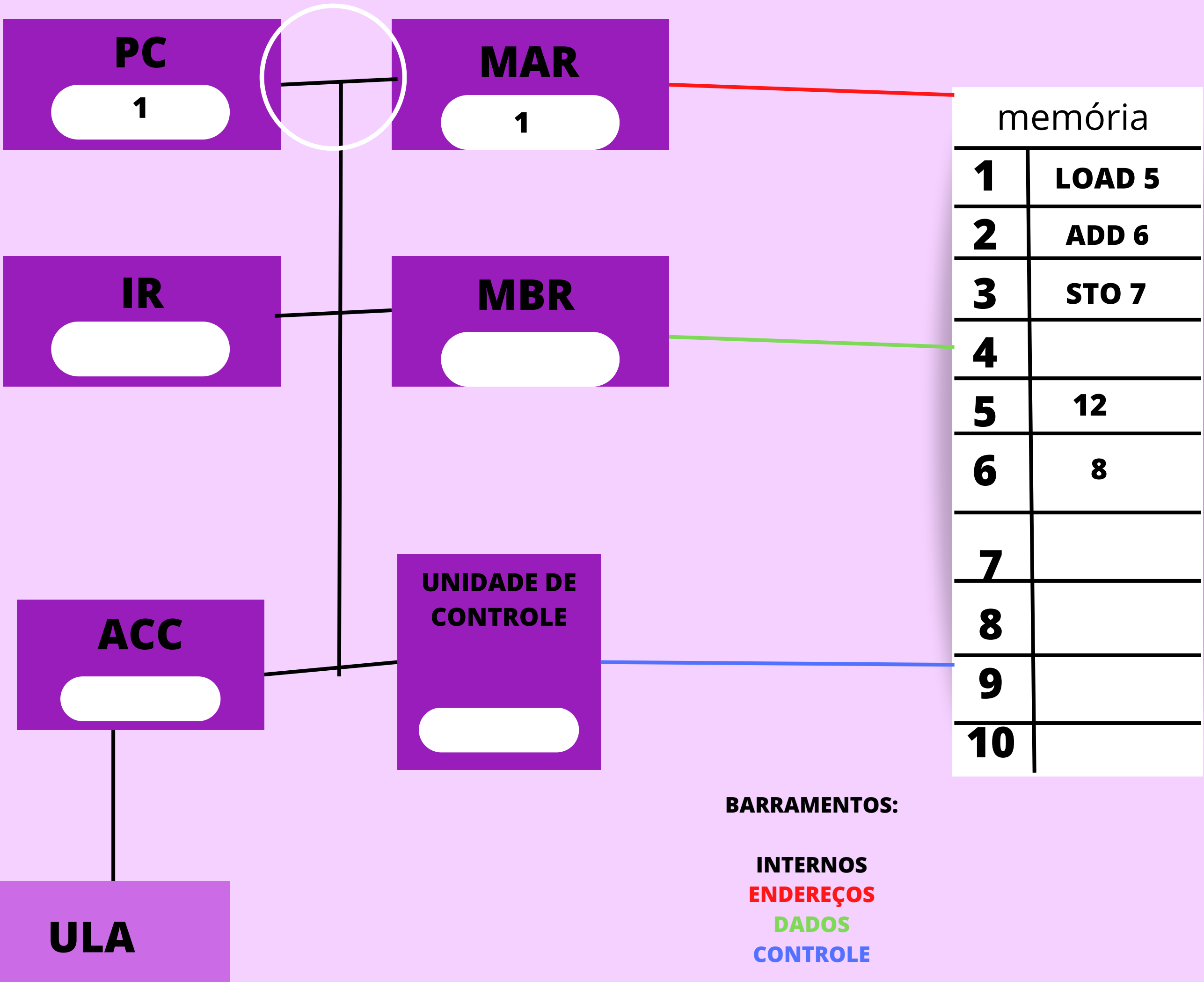
memória	
1	LOAD 5
2	ADD 6
3	STO 7
4	
5	12
6	8
7	
8	
9	
10	

**PC-** armazena o endereço da instrução a ser executada  
**MAR-** armazena um endereço de memória  
**MBR-** armazena uma palavra de dados  
**IR-** contém a última instrução buscada  
**ACC-** acumulador

**BARRAMENTOS:**

**INTERNOS**  
**ENDEREÇOS**  
**DADOS**  
**CONTROLE**

**load-** leitura  
**add-** adicionar  
**sto-** escrita



**PASSO 1:**  
**PC → MAR**

**PC**

1

**MAR**

1

**IR**

**MBR**

**ACC**

**UNIDADE DE  
CONTROLE**

**ULA**

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

**PASSO 2:**  
**MAR** →  
**BARRAMENTO DE  
ENDEREÇO**  
  
**(barramento de  
controle indica  
leitura)**

**BARRAMENTOS:**

**INTERNOS**  
**ENDEREÇOS**  
**DADOS**  
**CONTROLE**

PC

1

MAR

1

IR

MBR

load 5

ACC

UNIDADE DE  
CONTROLE

ULA

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

passo 3:  
A palavra é processada por  
meio do barramento de dados  
e colocada no MBR

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

PC

1

MAR

1

IR

MBR

load 5

ACC

UNIDADE DE  
CONTROLE

ULA

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

MBR → IR  
(deixando o MBR livre para outras  
execuções.

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

PC

1

MAR

1

IR

load 5

MBR

load 5

ACC

UNIDADE DE  
CONTROLE

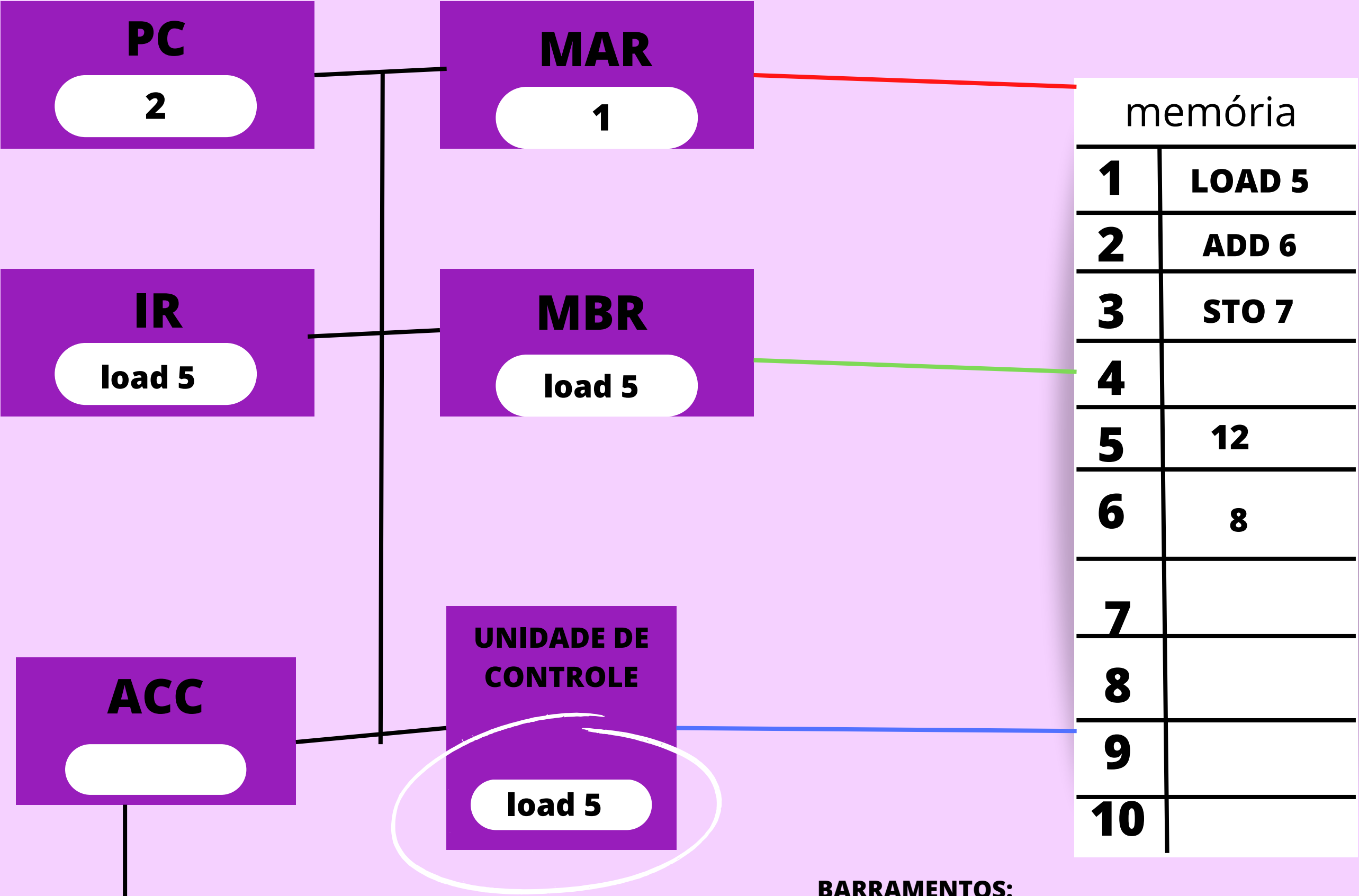
ULA

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

PC++  
(CICLO DE BUSCA SE ENCERRA)

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE



IR → UC  
(início do ciclo de decodificação)

Logo após a CPU identifica a instrução.

BARRAMENTOS:

- INTERNOS
- ENDEREÇOS
- DADOS
- CONTROLE

ULA



PC

2

MAR

5

IR

load 5

MBR

load 5

ACC

UNIDADE DE  
CONTROLE

load 5

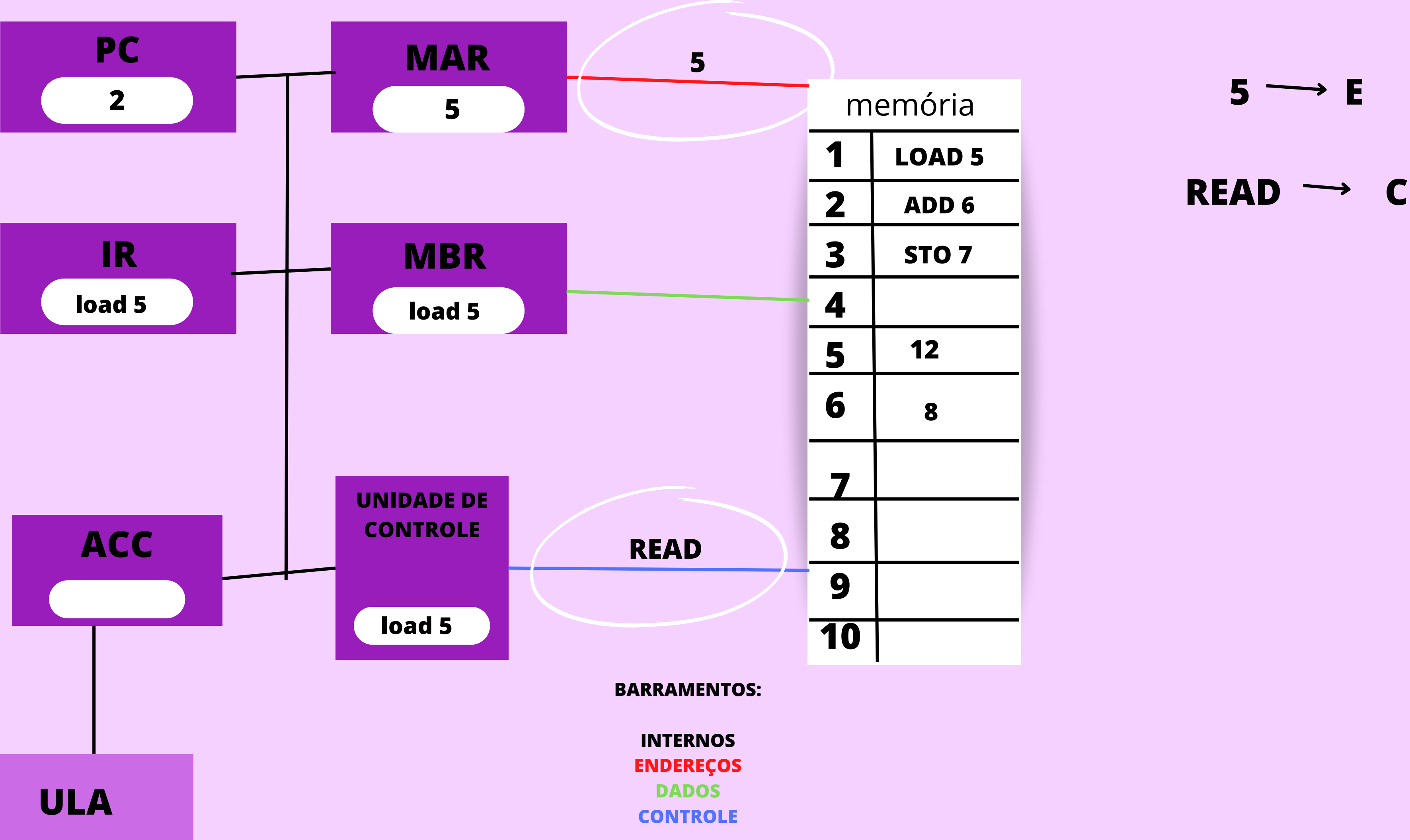
ULA

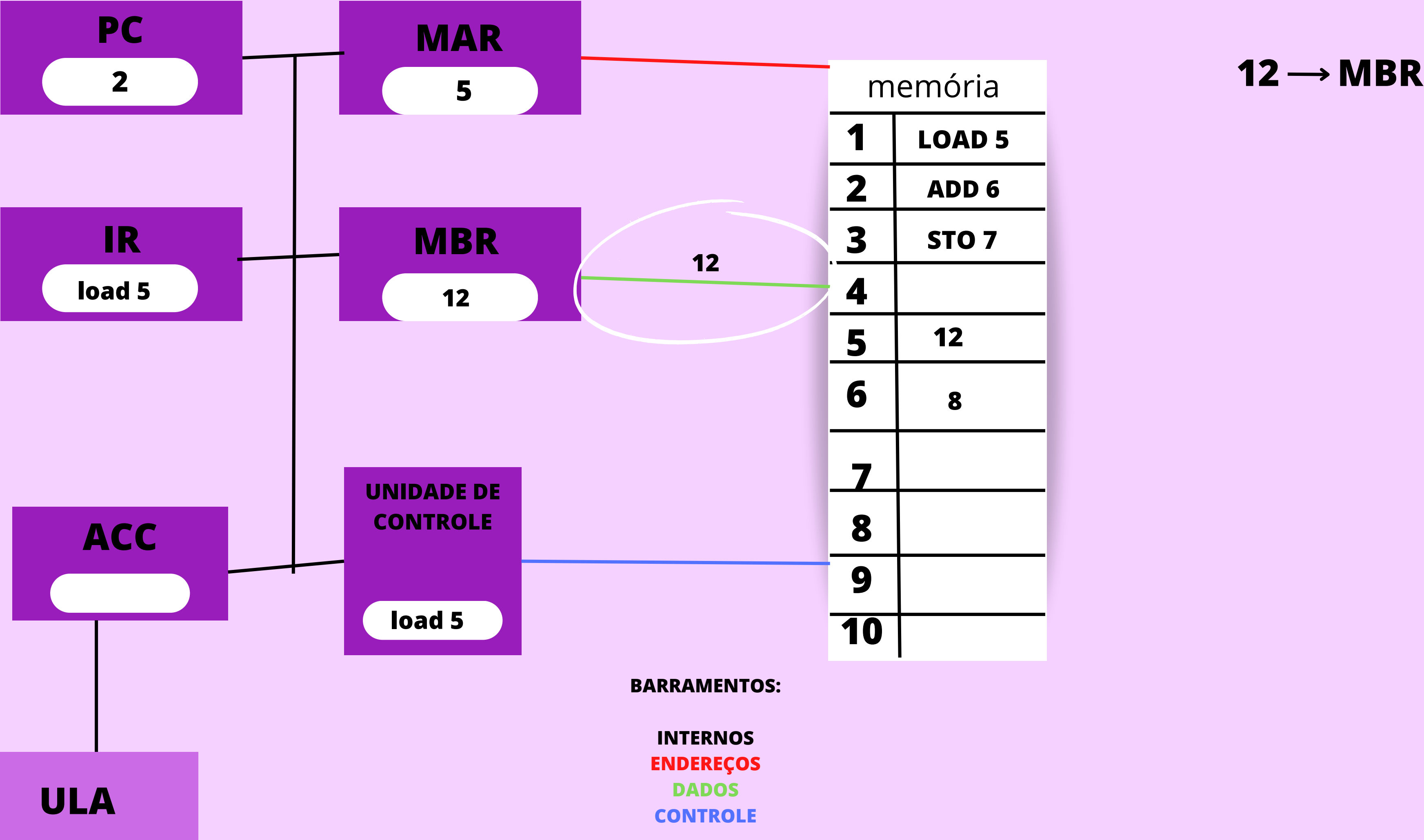
memória	
1	LOAD 5
2	ADD 6
3	STO 7
4	
5	12
6	8
7	
8	
9	
10	

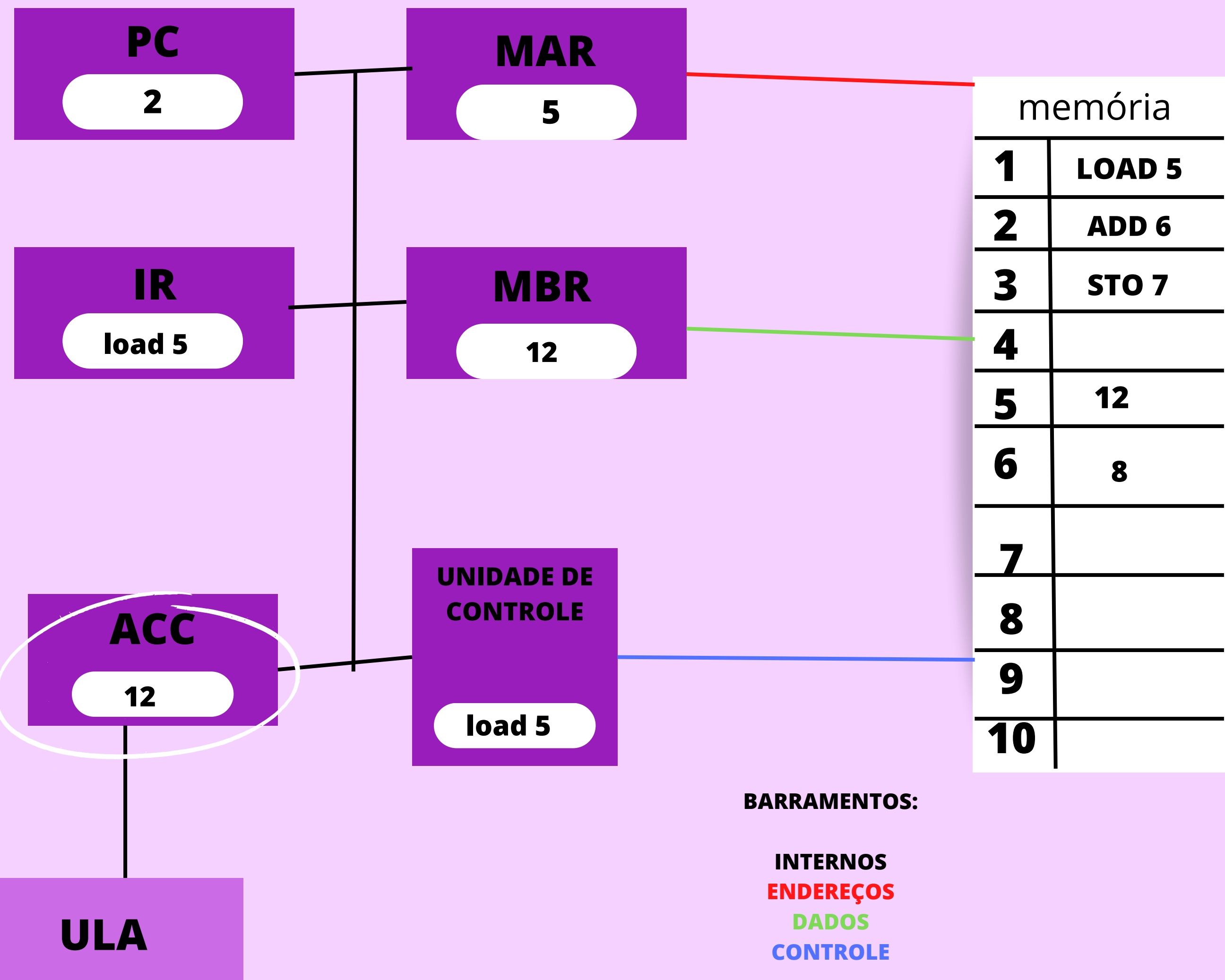
Busca do dado da posição 5, que é um endereço, portanto  
5 → MAR

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

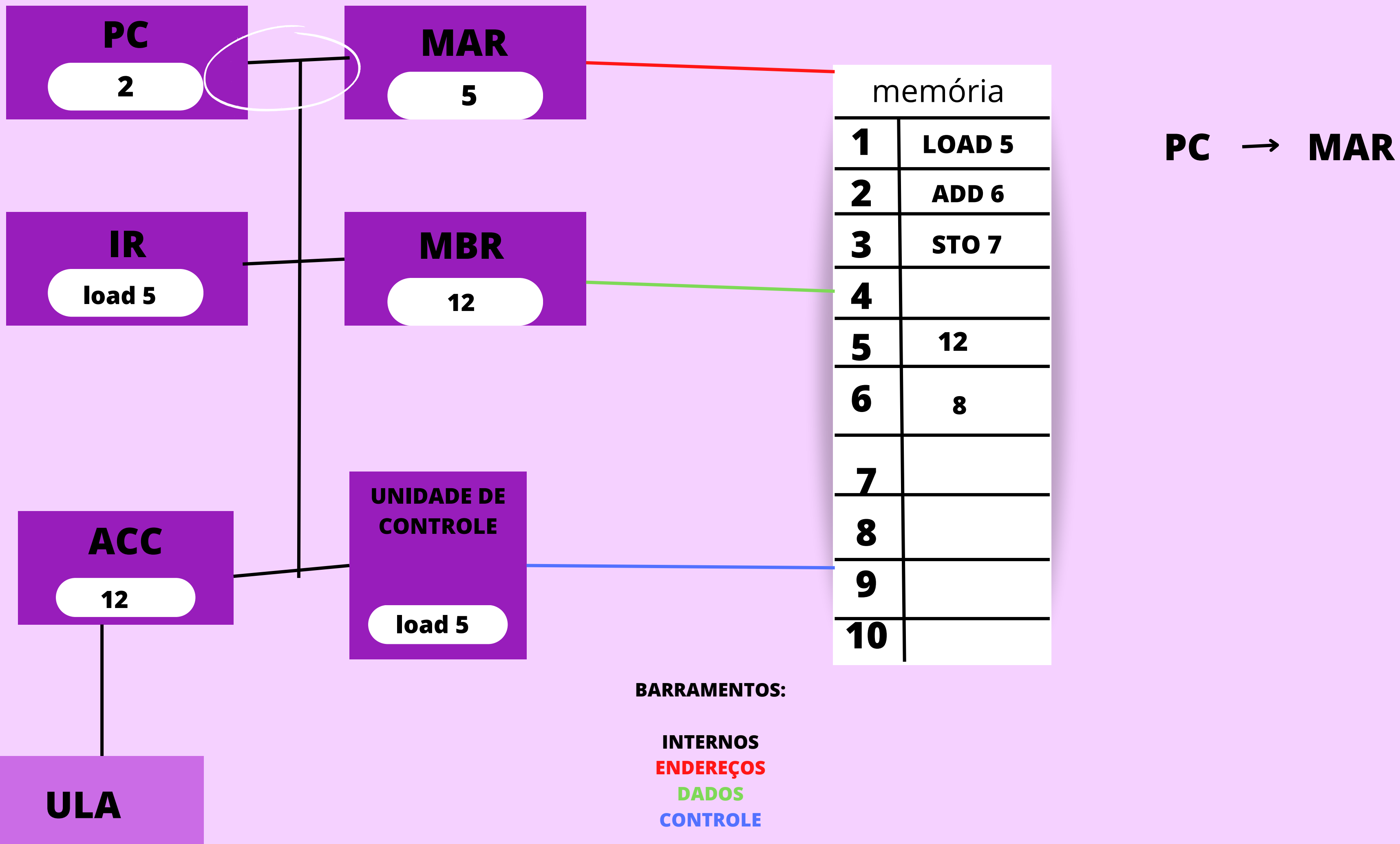


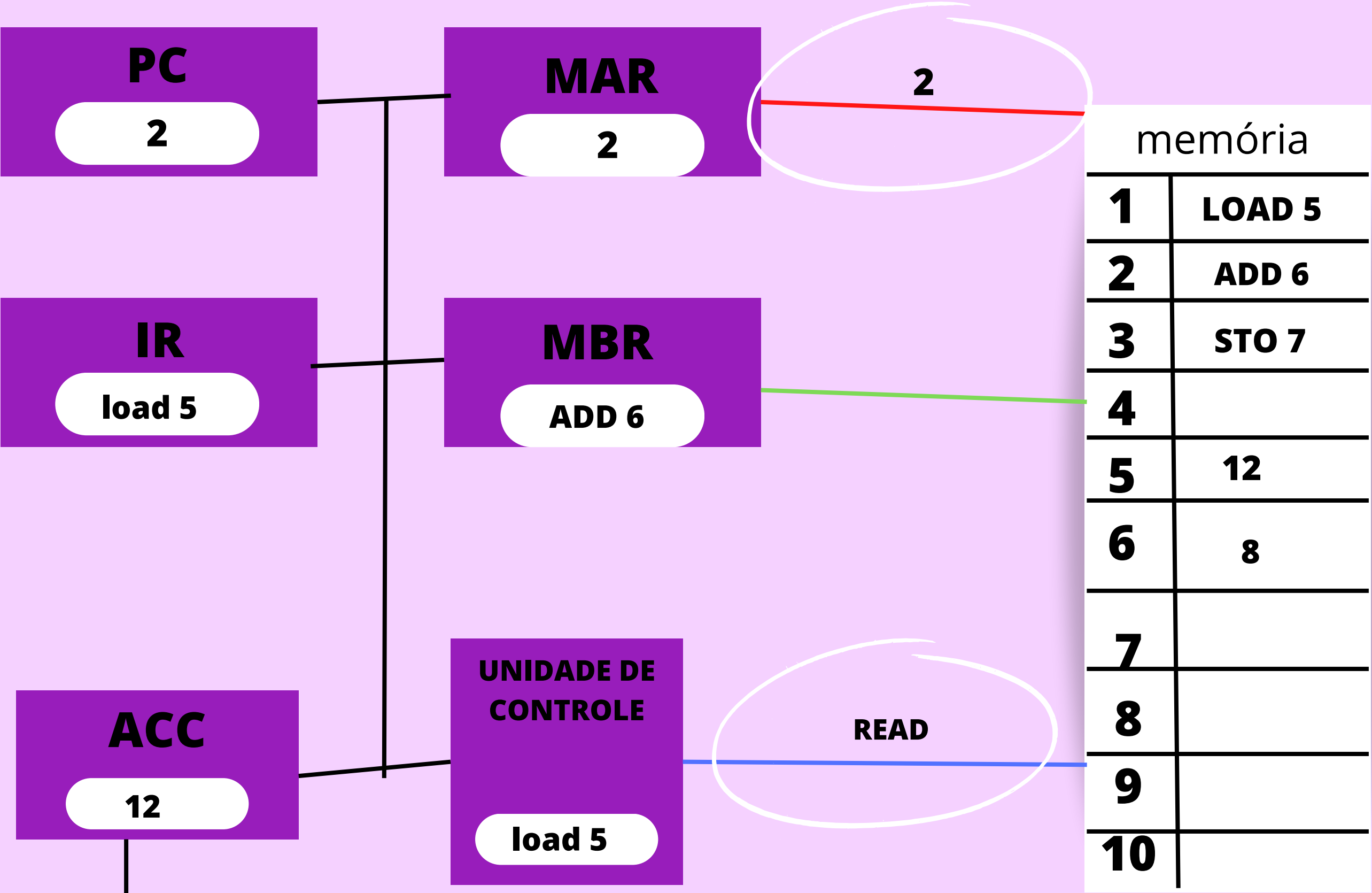




Não estamos lidando com a  
mesma instrução, logo  
MBR → ACC

Encerra ciclo de execução.  
tendo que buscar, decodificar e  
executar a próxima instrução.

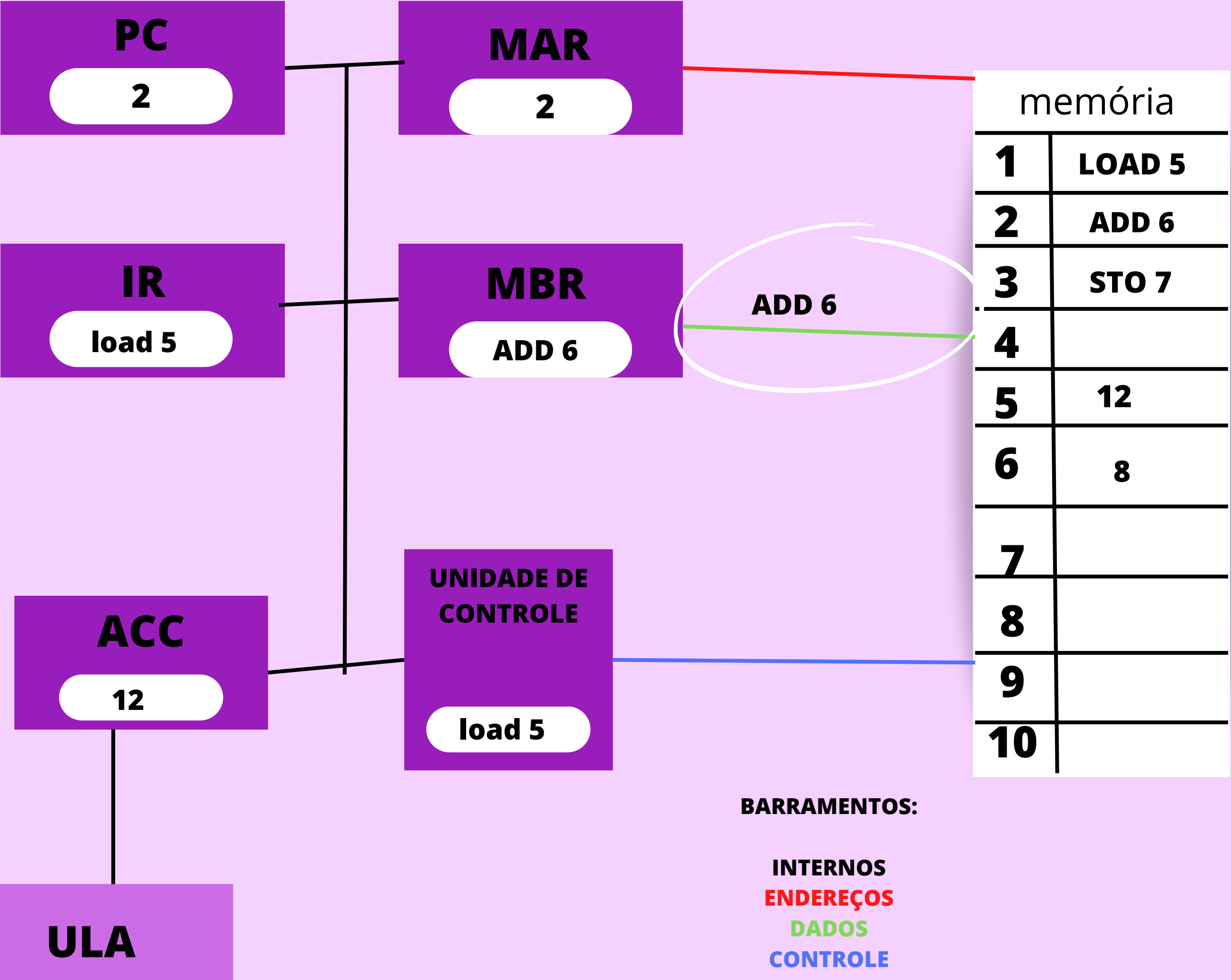




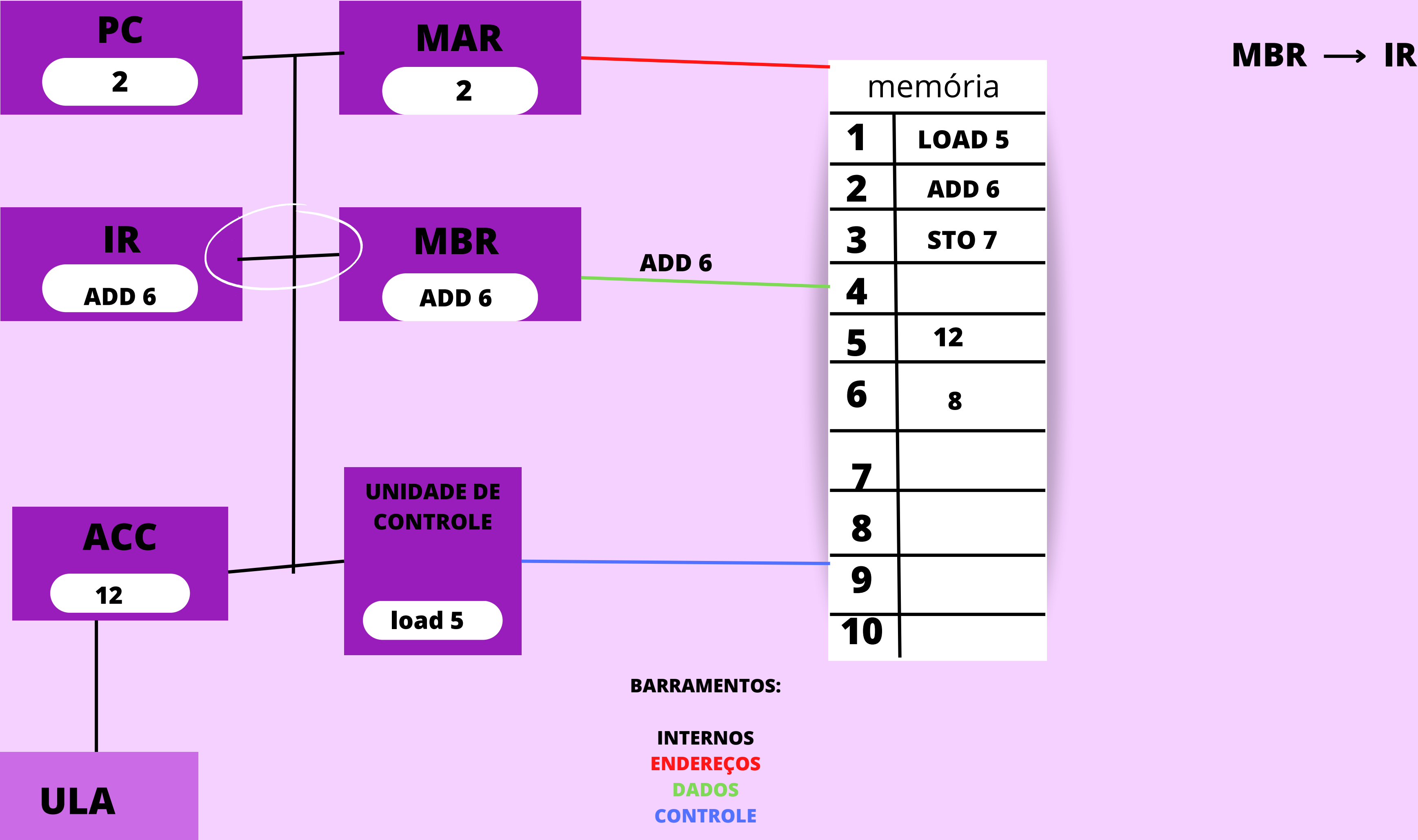
Valor em MAR passado pela linha de endereço.  
Linha de controle como leitura

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE



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





PC

3

MAR

2

IR

ADD 6

MBR

ADD 6

ACC

12

UNIDADE DE  
CONTROLE

load 5

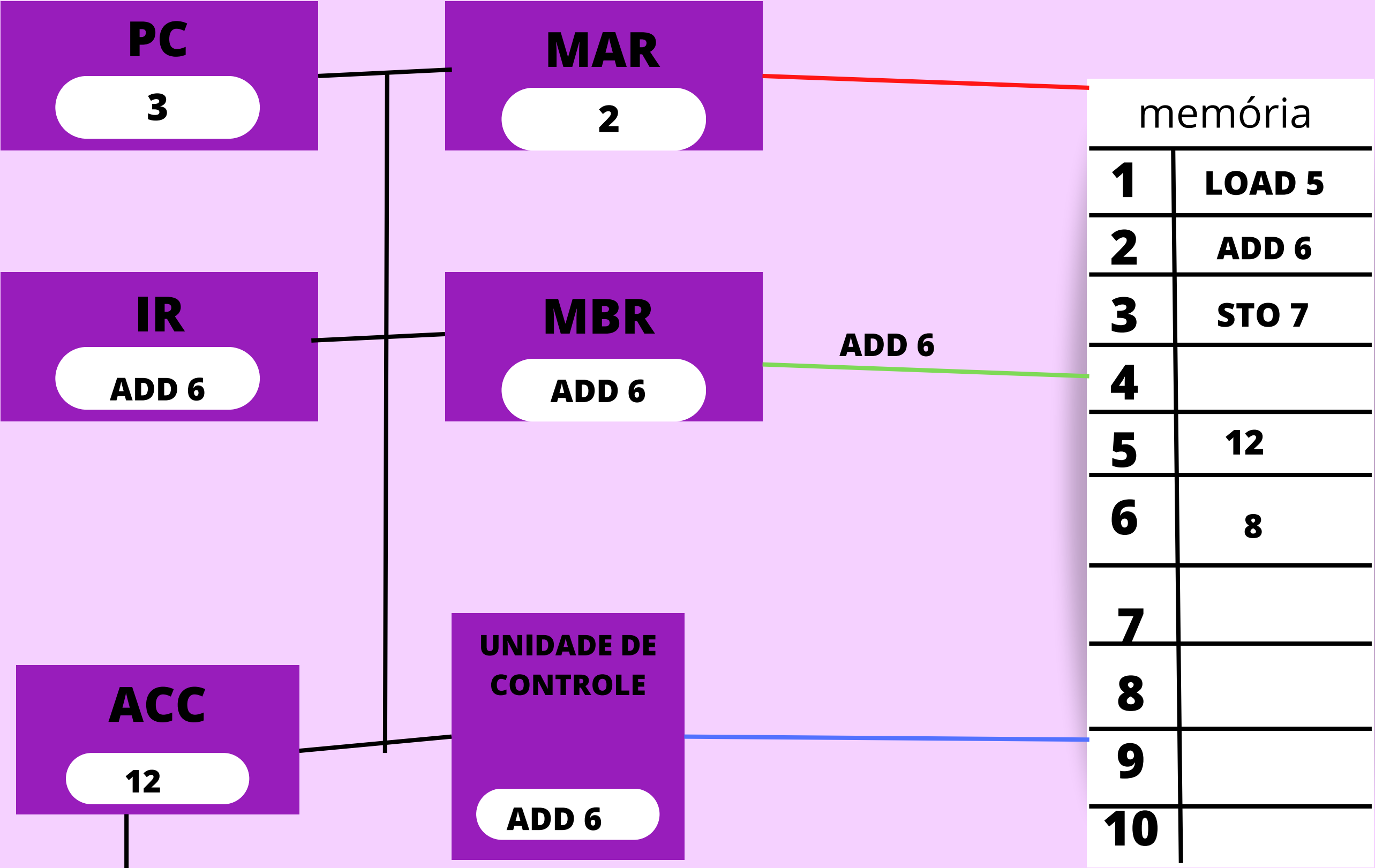
ULA

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

PCC++

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

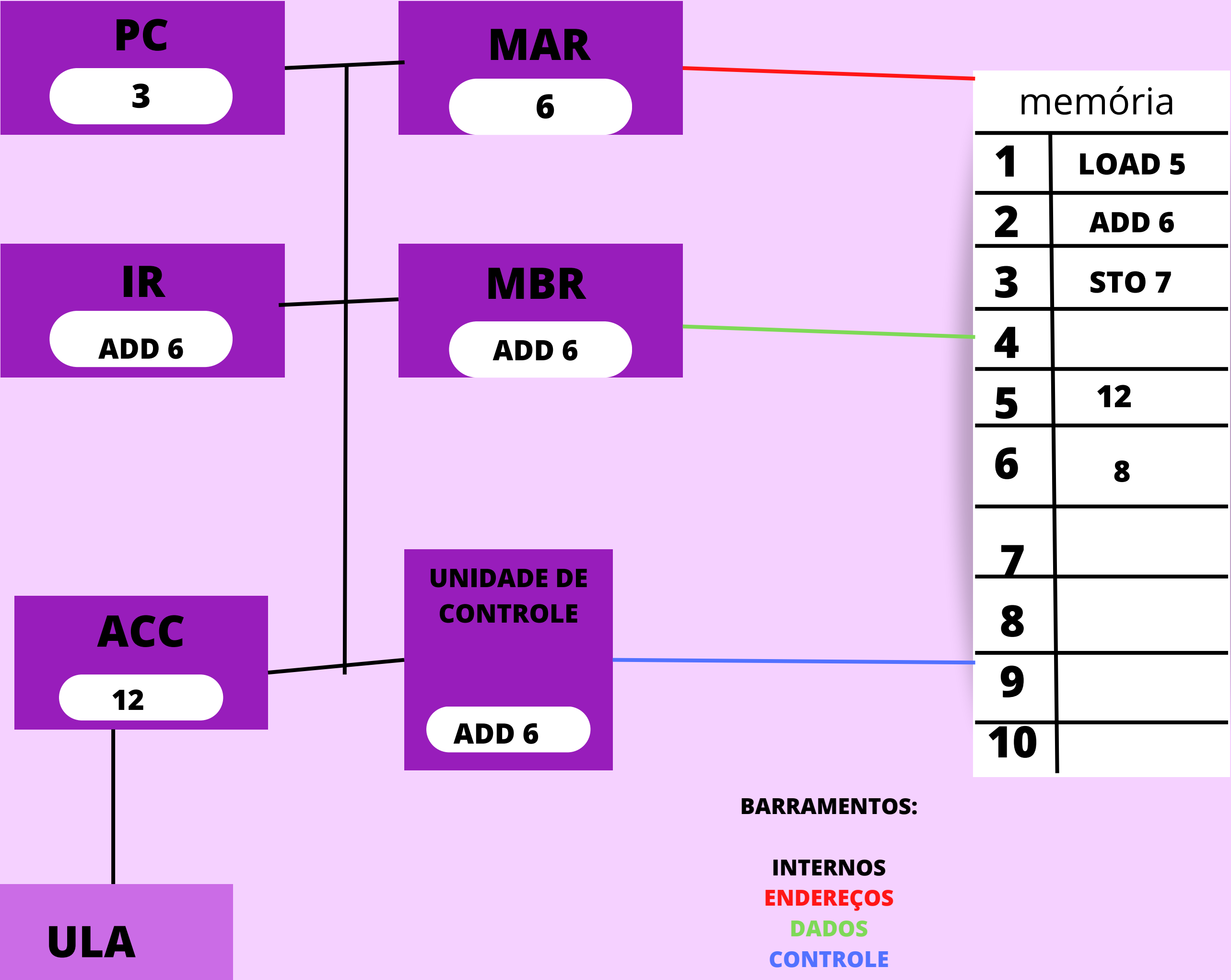


IR → UC  
Logo após o CPU identifica a instrução.

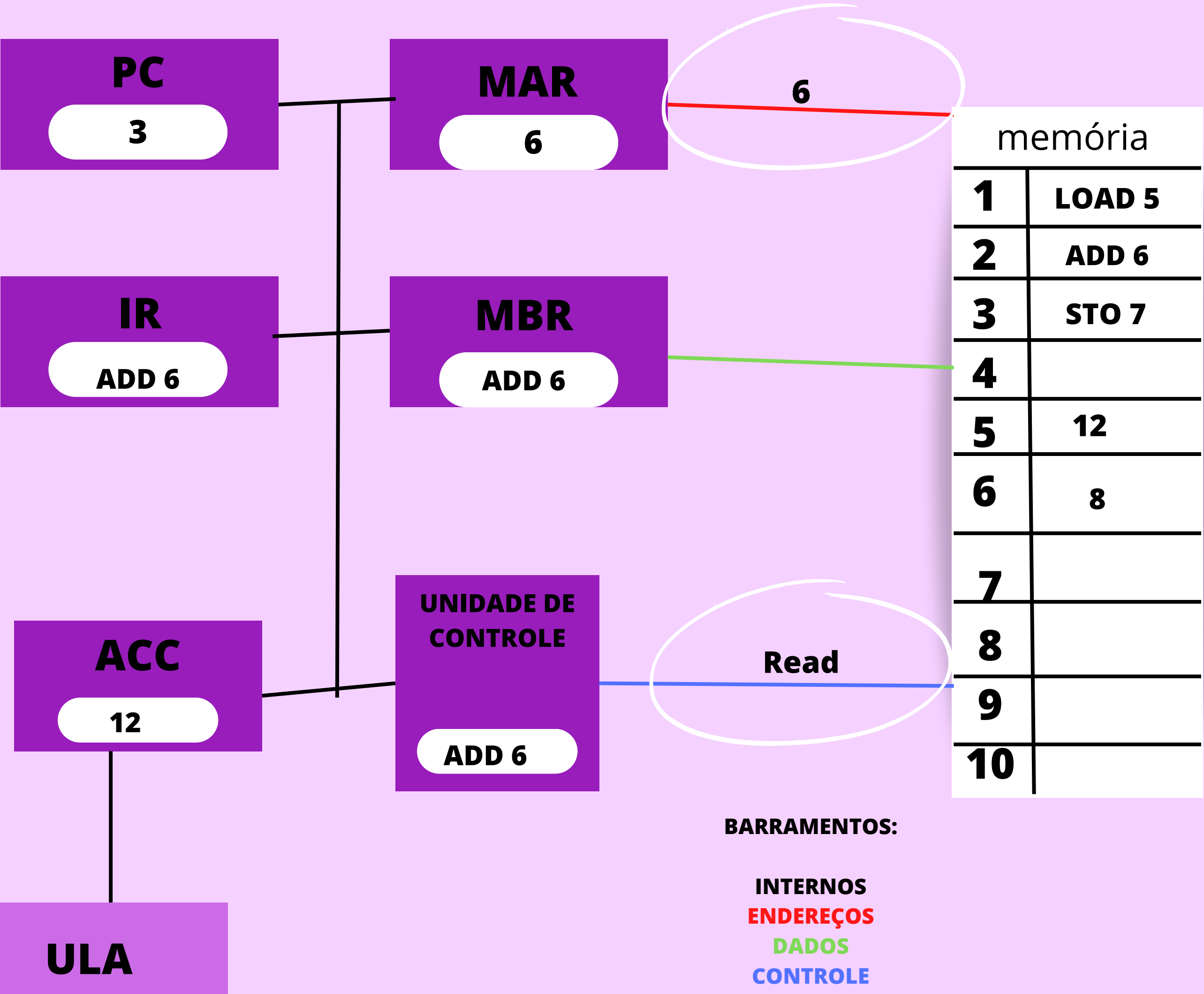
BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

ULA



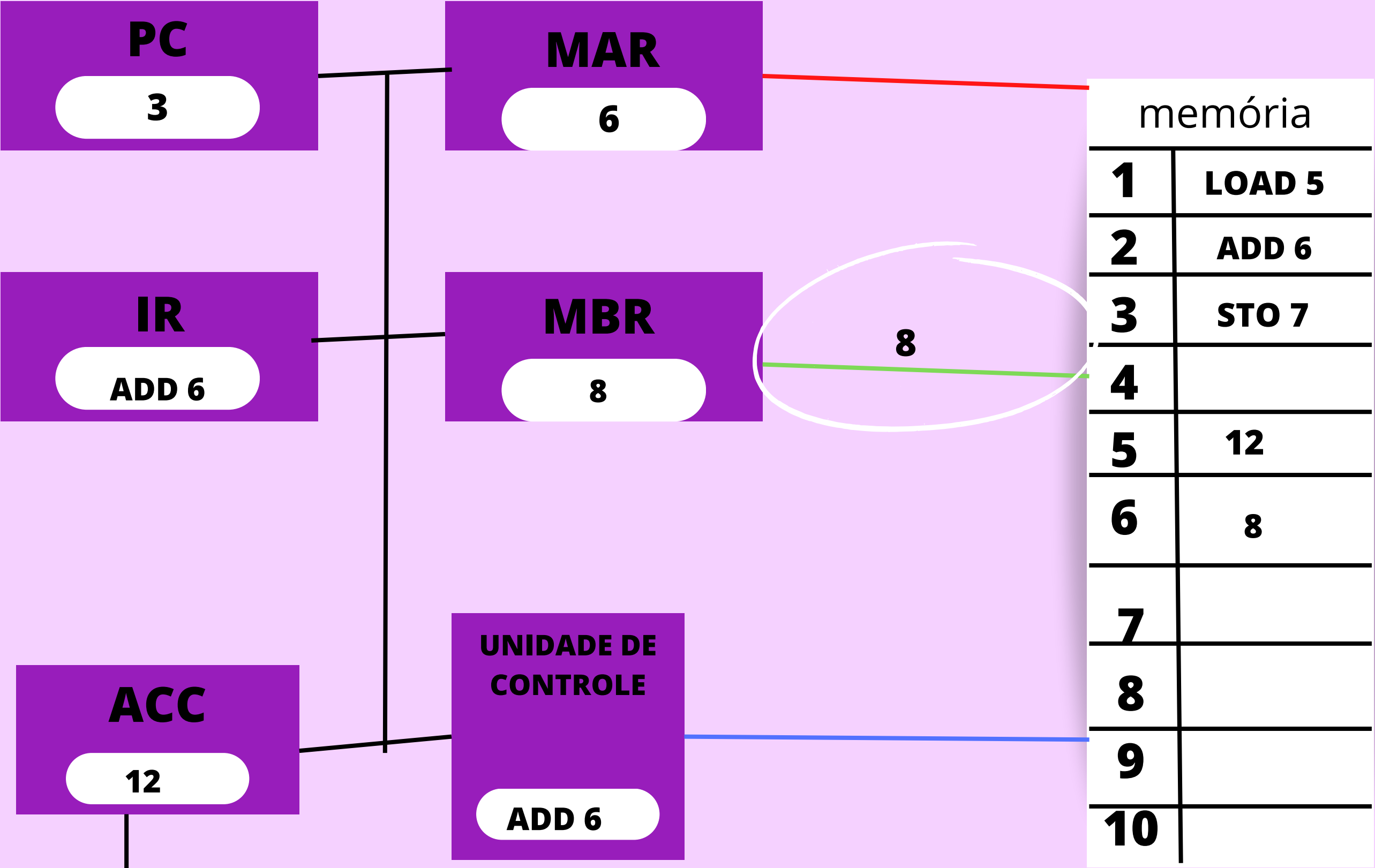
Logo, deverá buscar no endereço 6.  
Portanto 6 → MAR



O endereço é passado e lemos da memória

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

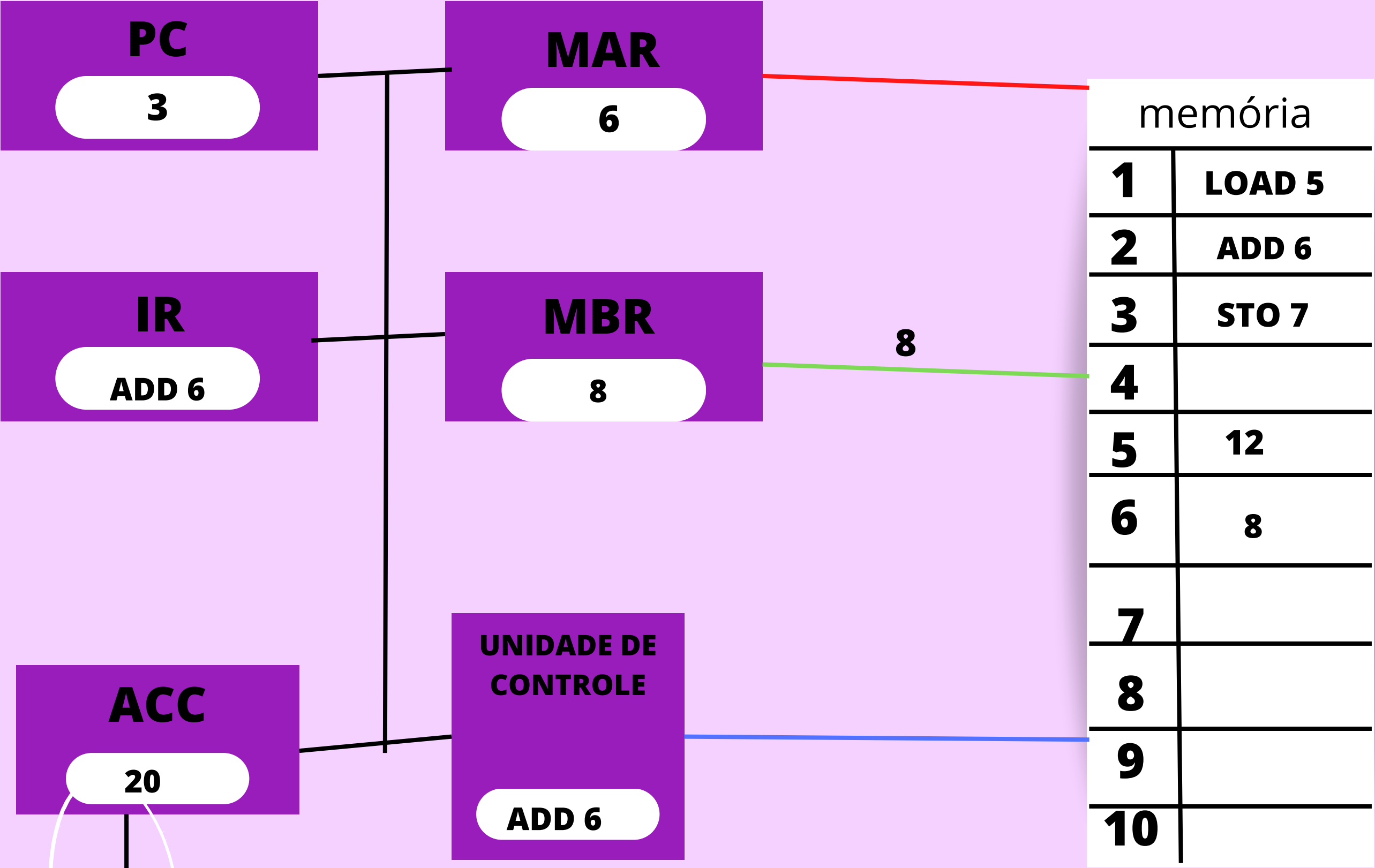


O dado é passado pela linha de dados e armazenado no MBR

BARRAMENTOS:

- INTERNOS
- ENDEREÇOS
- DADOS
- CONTROLE

ULA

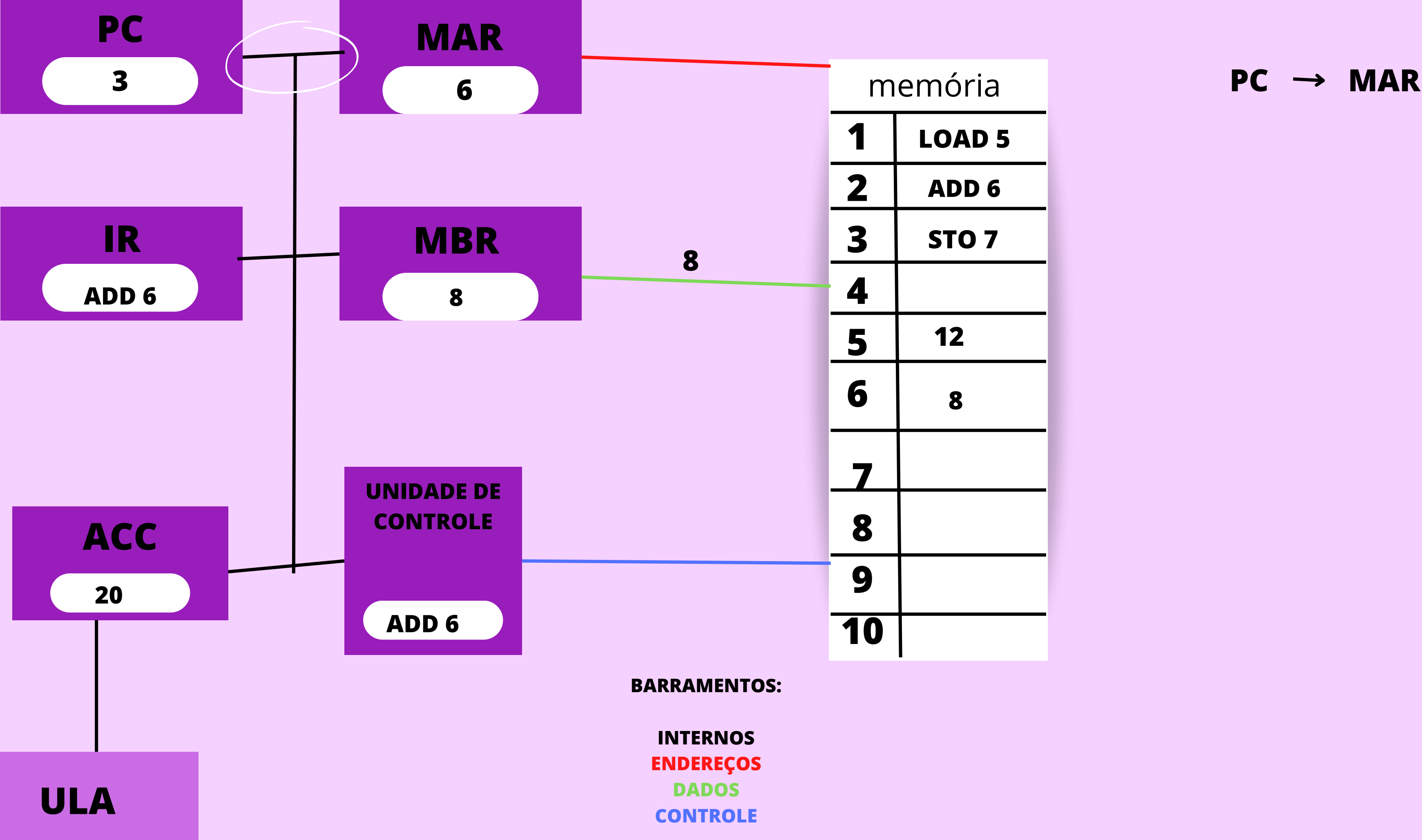


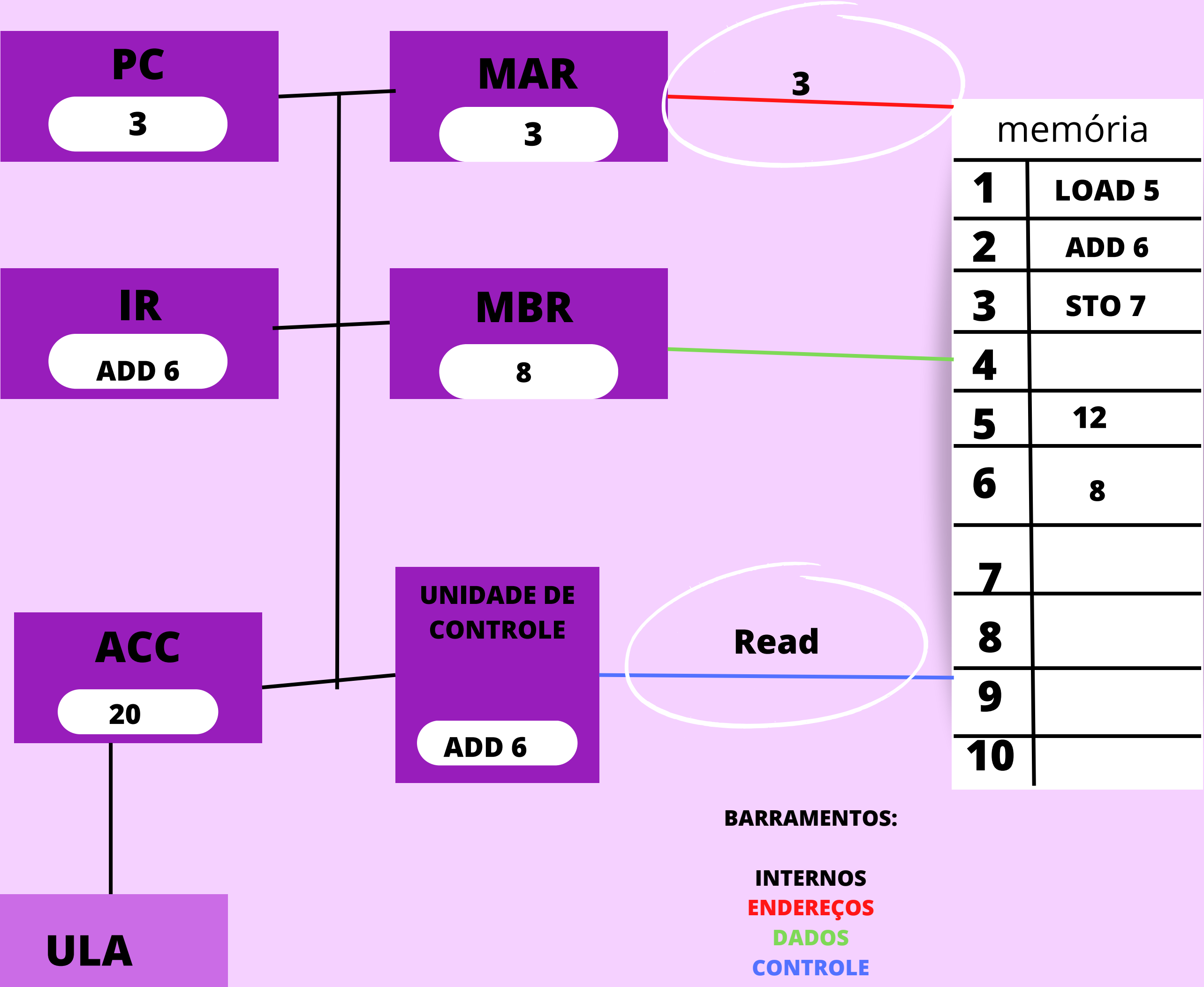
A soma é realizada pela ULA e salva no acumulador  
 $12+8 \rightarrow \text{ACC}$

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

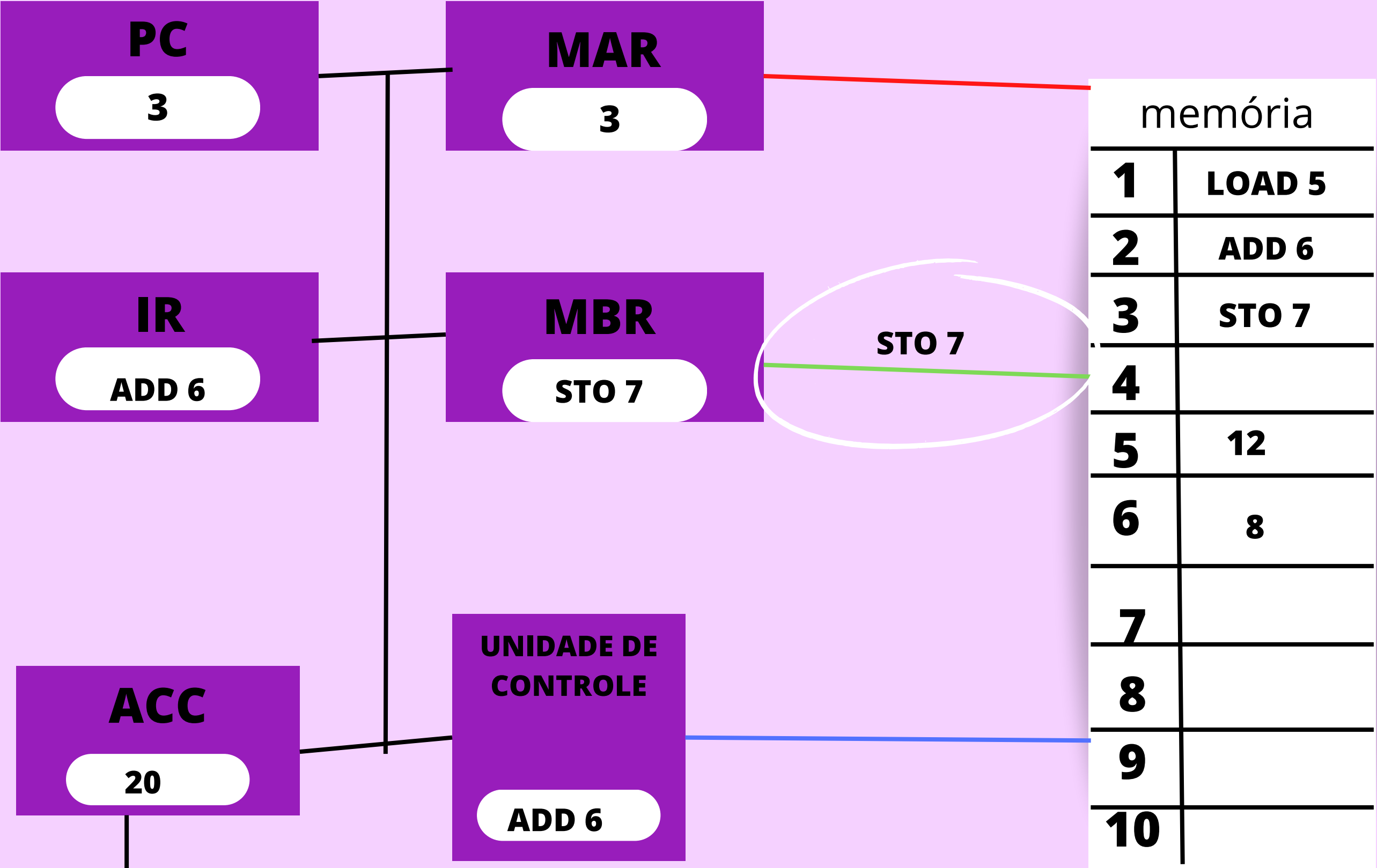






O endereço é passado pela linha de dados e controle indica leitura (read)





Instrução é passada pela  
linha de dados e  
armazenada em MBR

PC

3

MAR

3

IR

STO 7

MBR

STO 7

ACC

20

UNIDADE DE  
CONTROLE

ADD 6

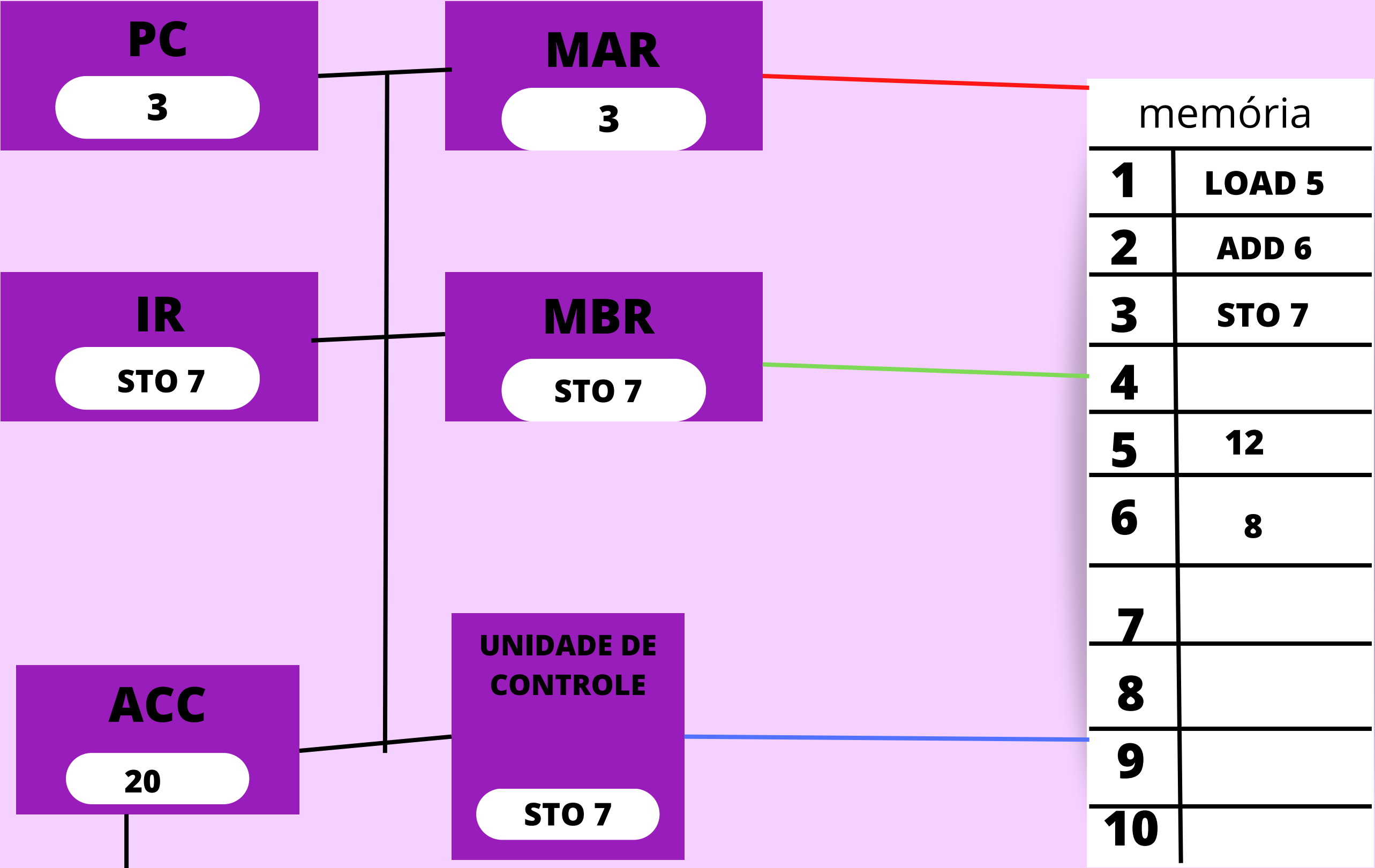
ULA

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

Por ser uma nova  
instrução MBR → IR

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE



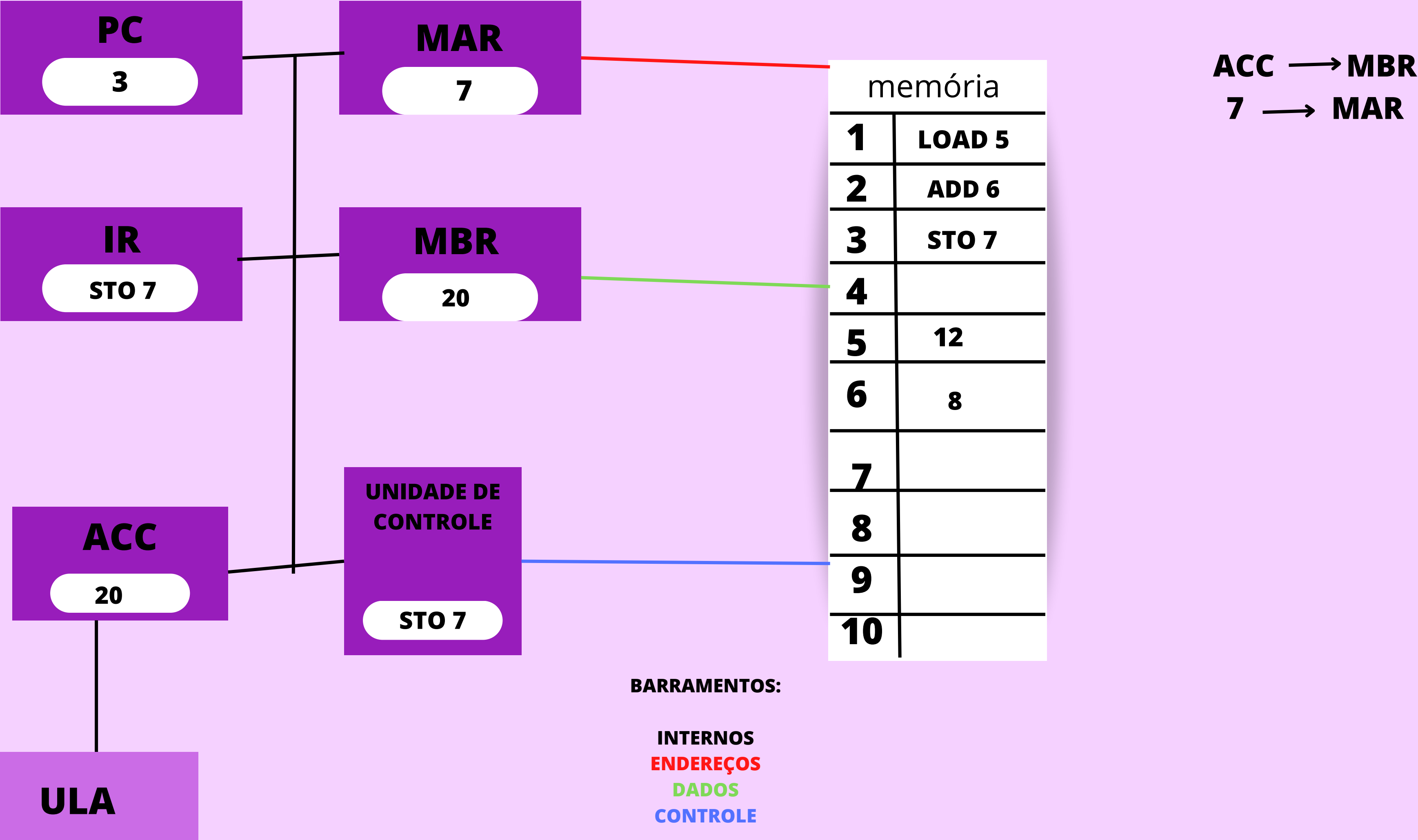
**IR → UC**  
(o endereço da última instrução a ser executada)

( logo em seguida a CPU identifica a instrução)

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE

ULA



PC

3

MAR

7

IR

STO 7

MBR

20

ACC

20

UNIDADE DE  
CONTROLE

STO 7

ULA

7

20

Write

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

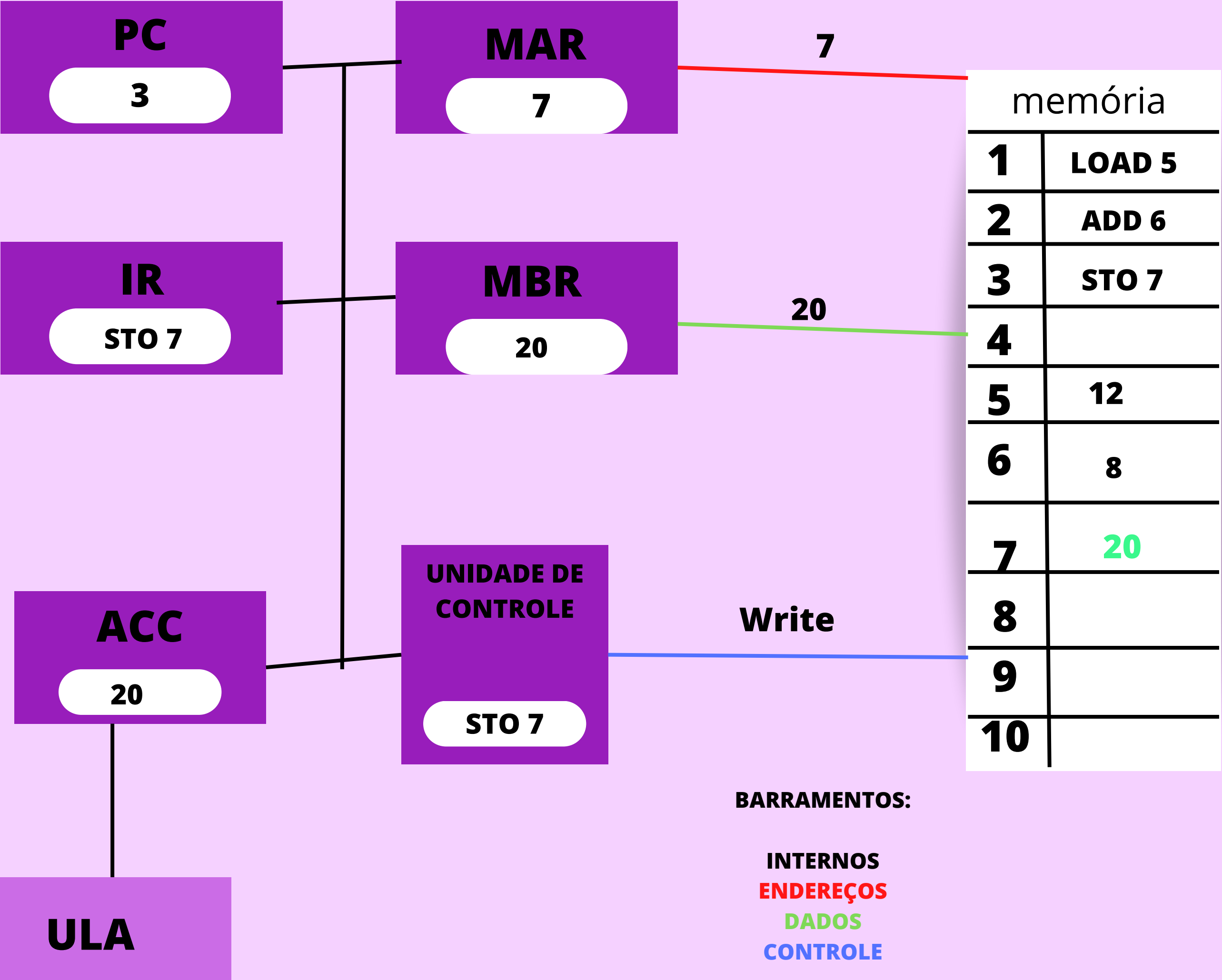
conteúdo de MAR passado  
pela linha de endereço;

conteúdo de MBR passado  
pela linha de dados;

controle de escrita.

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE



Com isso a informação é passada e armazenada na memória.

BARRAMENTOS:

INTERNOS  
ENDEREÇOS  
DADOS  
CONTROLE