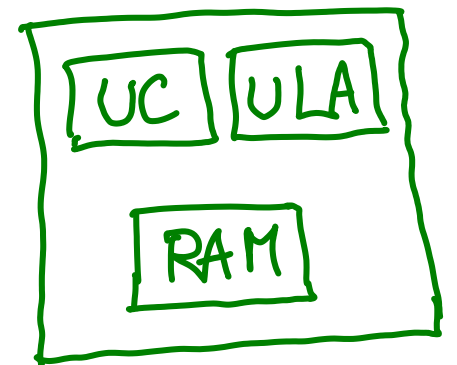
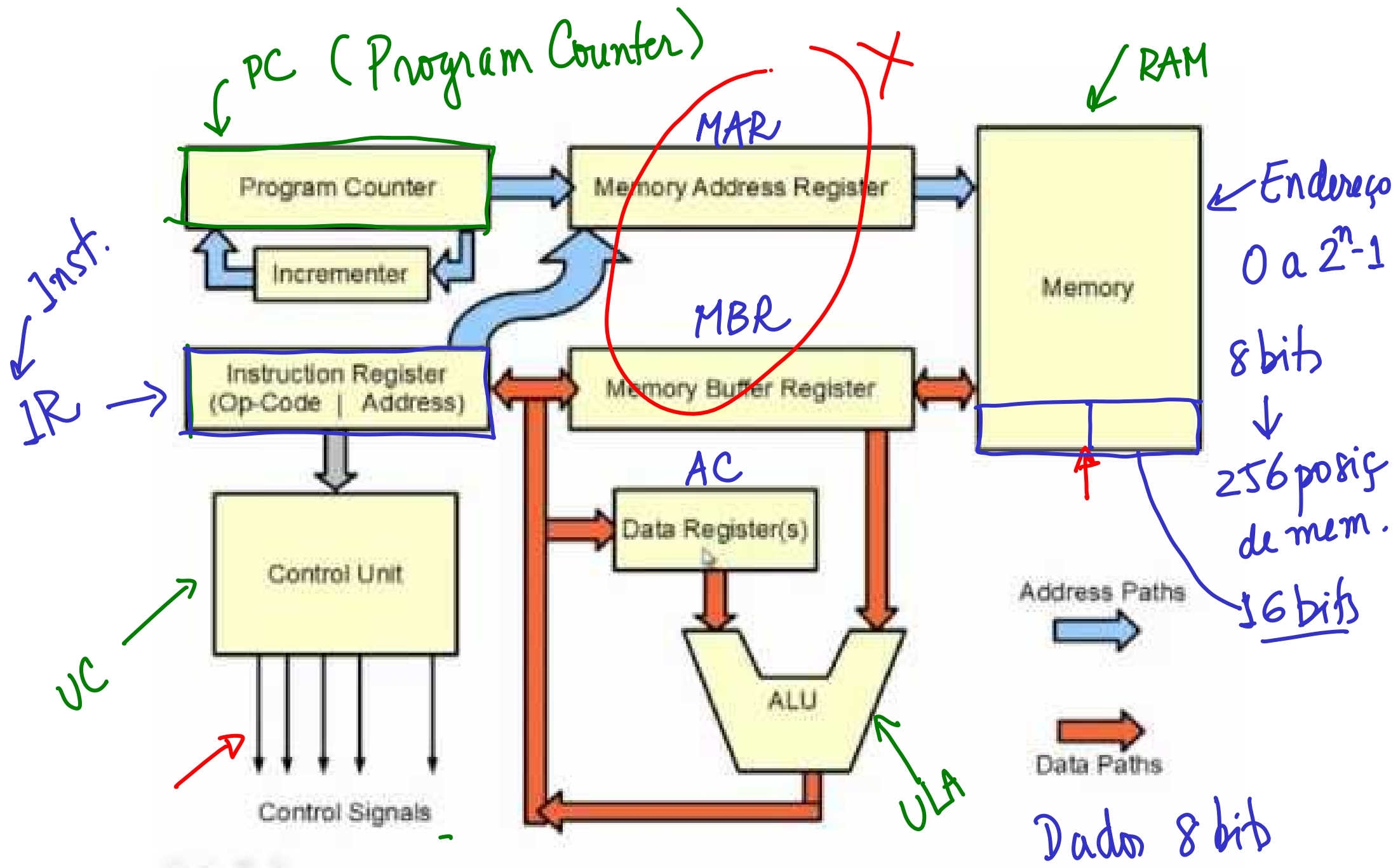


M A C O 3 2 9

06.07.2021

Bom dia !!

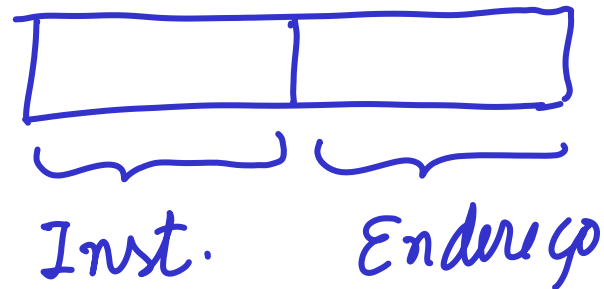




Dados \rightarrow 8 bits

Endereços \rightarrow 8 bits \Rightarrow 256 posições
na RAM

Instruções \rightarrow 16 bits



RAM \rightarrow 16 bits

0000 0000

01 --

| Código | | Descrição |
|---------|---------|---|
| base 10 | base 16 | |
| 00 | 00 | NOP (no operation) |
| 01 | 01 | Copie [EE] para o AC |
| 02 | 02 | Copie [AC] para a posição de endereço EE |
| 03 | 03 | Some [EE] com [AC] e guarde o resultado em AC |
| 04 | 04 | Subtraia [EE] de [AC] e guarde o resultado em AC |
| 07 | 07 | Leia um número e guarde-o na posição de endereço EE |
| 08 | 08 | Imprima [EE] |
| 09 | 09 | Pare |
| 10 | 0A | Desvie para EE (desvio incondicional) |
| 11 | 0B | Desvie para EE se [AC] > 0 |
| 13 | 0D | Desvie para EE se [AC] = 0 |
| 15 | 0F | Desvie para EE se [AC] < 0 |

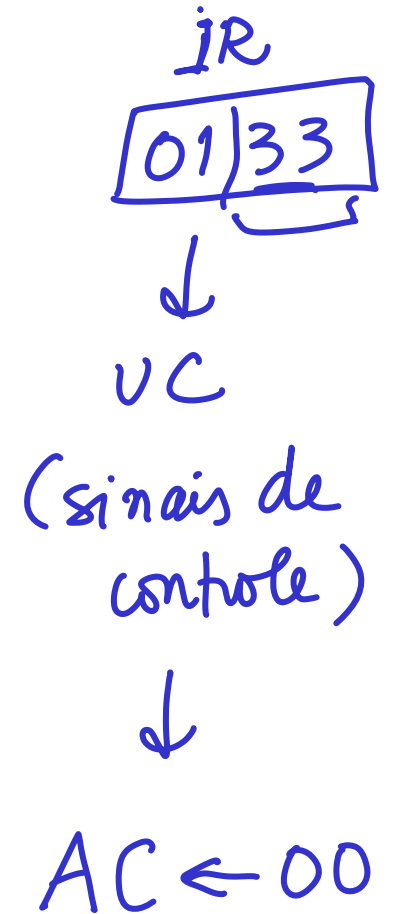
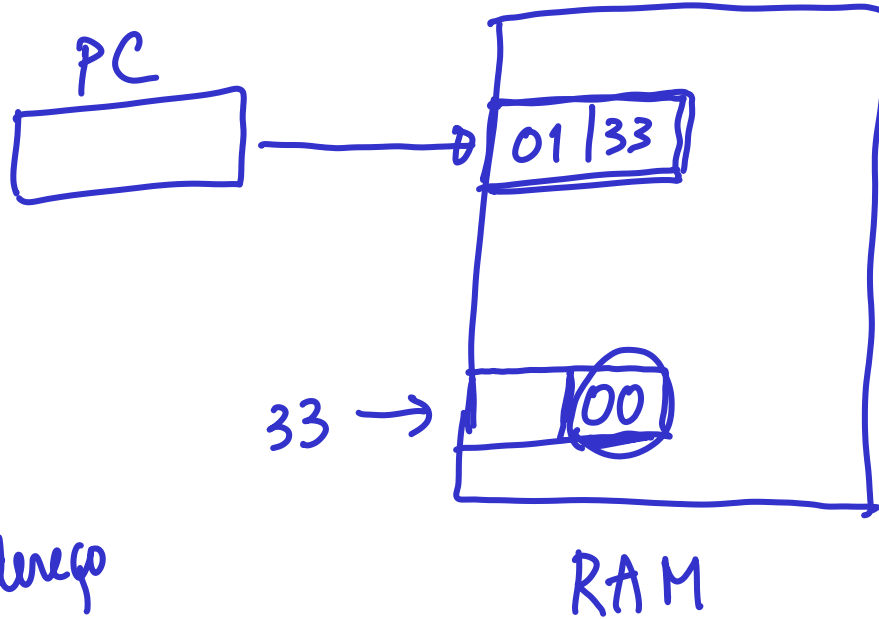
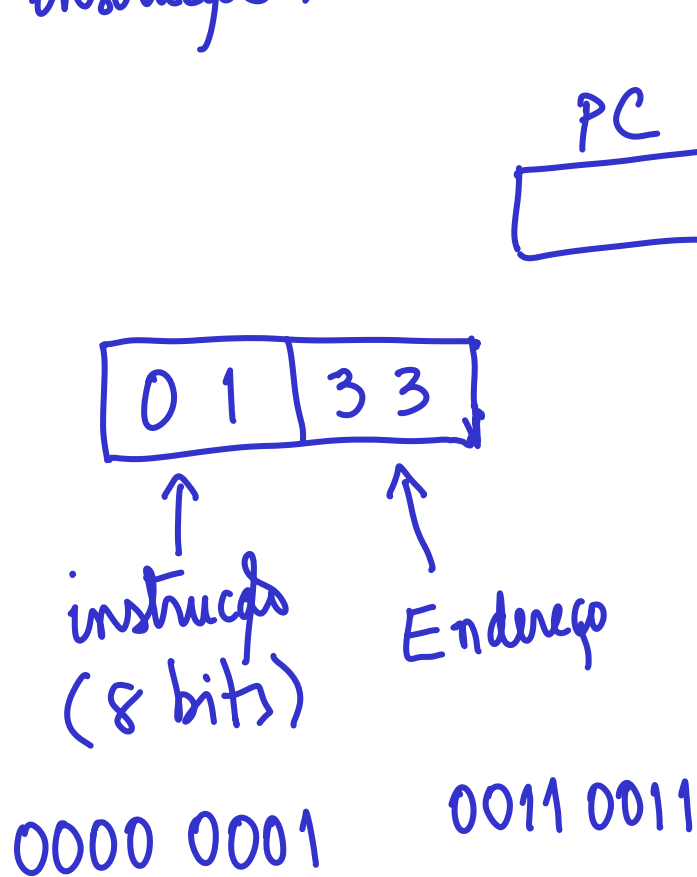
Registrador

]

Código
(instrução)

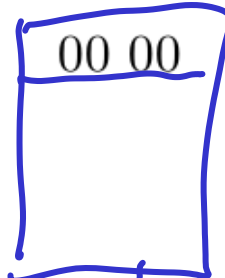
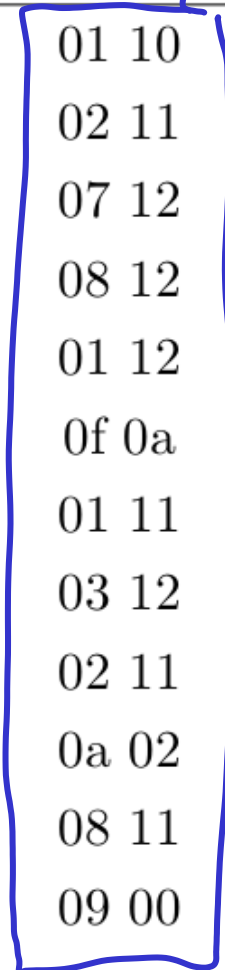
01

Copie [EE] para o AC



| Endereço | Instrução | Tradução |
|----------|-----------|---|
| 00 | 01 10 | Copie [10] para o AC |
| 01 | 02 11 | Copie [AC] para a posição de endereço 11 |
| 02 | 07 12 | Leia um número e guarde-o na posição de endereço 12 |
| 03 | 08 12 | Imprima [12] |
| 04 | 01 12 | Copie [12] para o AC |
| 05 | 0f 0a | desvie para 0a se [AC] < 0 |
| 06 | 01 11 | Copie [11] para o AC |
| 07 | 03 12 | Some [AC] com [12] e guarde o resultado em AC |
| 08 | 02 11 | Copie [AC] para a posição de endereço 11 |
| 09 | 0a 02 | Desvie para 02 |
| 0a | 08 11 | Imprima [11] |
| 0b | 09 00 | Pare |
| ... | ... | ... |
| → 10 | 00 00 | dado (número 0) |
| 11 | | espaço para armazenar dado |
| 12 | | espaço para armazenar dado |

→ programa



o Dados

| Endereço | Instrução | Tradução |
|----------|--------------|--|
| 00 | <u>01 10</u> | <u>Copie [10] para o AC</u> |
| 01 | <u>02 11</u> | Copie [AC] para a posição de endereço 11 |
| 02 | <u>07 12</u> | <u>Leia um número e guarde-o na posição de endereço 12</u> |
| 03 | 08 12 | Imprima [12] |
| 04 | 01 12 | Copie [12] para o AC |
| 05 | 0f 0a | desvie para 0a se [AC] < 0 |
| 06 | 01 11 | Copie [11] para o AC |
| 07 | 03 12 | Some [AC] com [12] e guarde o resultado em AC |
| 08 | 02 11 | Copie [AC] para a posição de endereço 11 |
| 09 | 0a 02 | Desvie para 02 |
| 0a | 08 11 | Imprima [11] |
| 0b | 09 00 | Pare |
| ... | ... | ... |
| 10 | <u>00 00</u> | dado (número 0) |
| 11 | | espaço para armazenar dado |
| 12 | | espaço para armazenar dado |

10 ~ Zero
 11 ~ Soma
 12 ~ Num

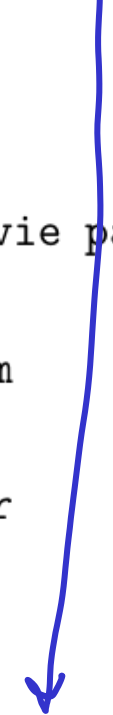
AC ←-- Zero
 Soma ←-- AC

Ler: Leia Num
 Imprima Num
 AC ←-- Num
 Se AC < 0, desvie para Fim

AC ←-- Soma
 AC ←-- AC + Num
 Soma ←-- AC
 Desvie para Ler

Fim: Imprima Soma
 Pare

$Soma \leftarrow Soma + Num$
 $Soma \leftarrow Zero$



Execução de instruções.

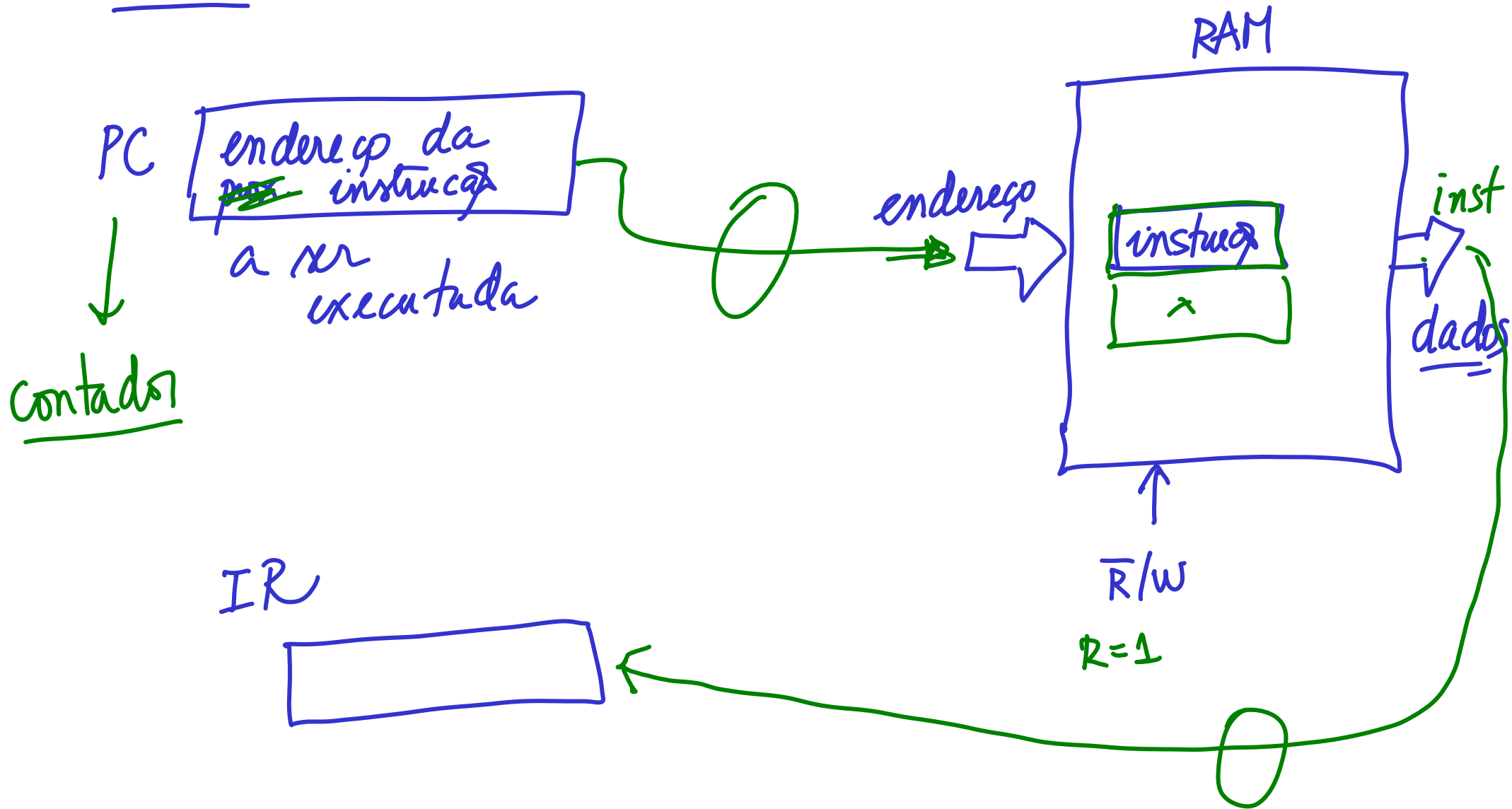
FDX cycle (Fetch-Decode-Execute)

1. Fetch \Rightarrow buscar a próxima instrução
Atualizar o PC

2. Decode \Rightarrow preparar a execução.

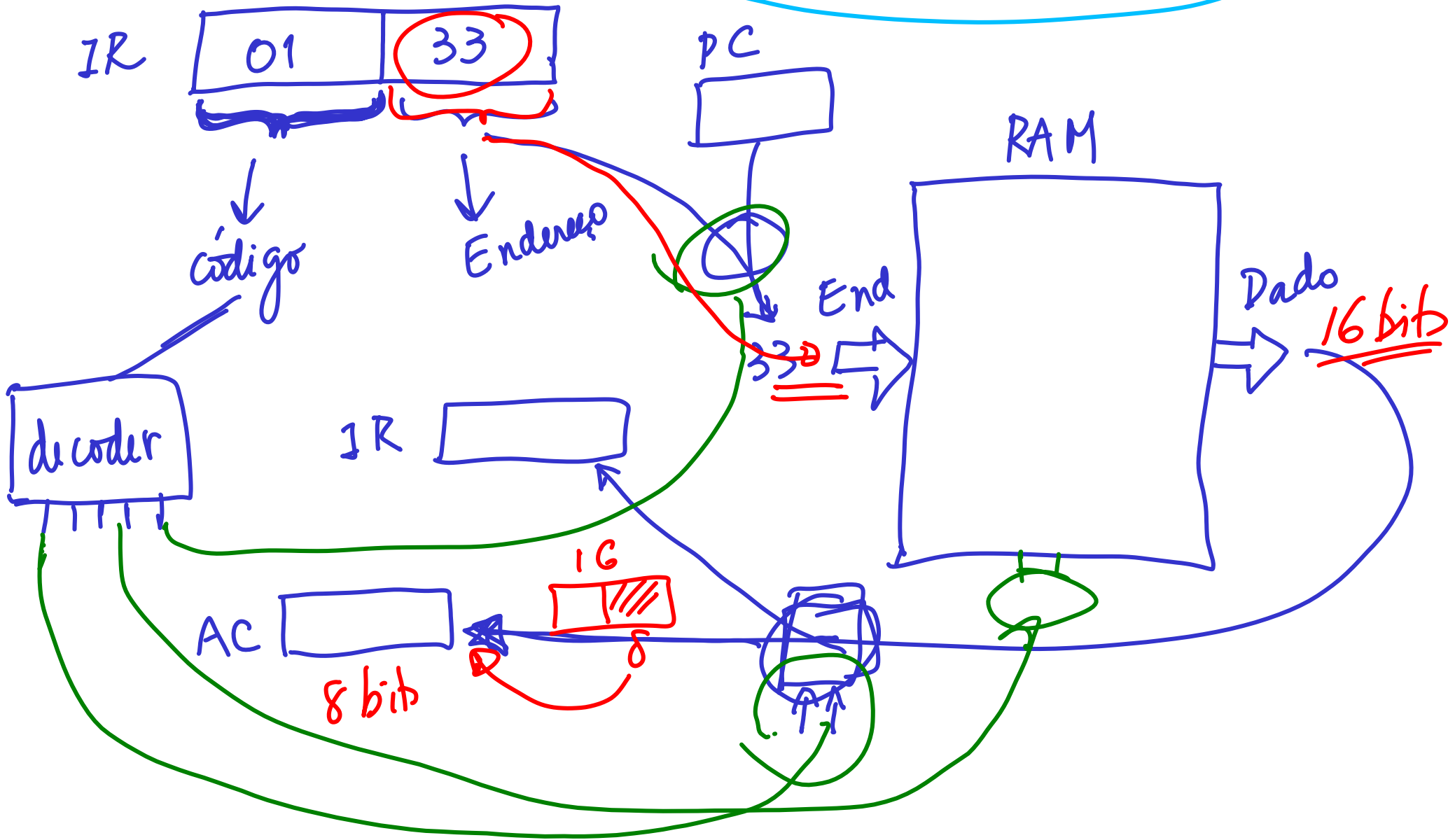
3. Execute \Rightarrow execução da instrução + next cycle.

1. Fetch (Buscar a instrução)



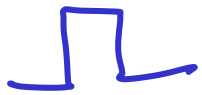
2. De code

01 33
Copie [33] p/σ AC



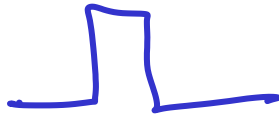
3. Execute 

Reset o ciclo



Fetch

Decode



Execute

