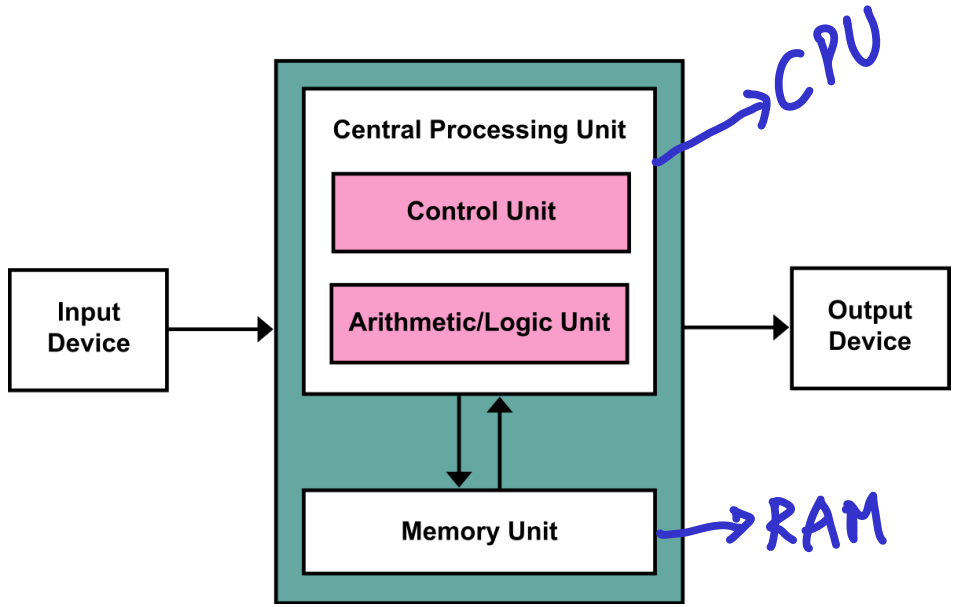
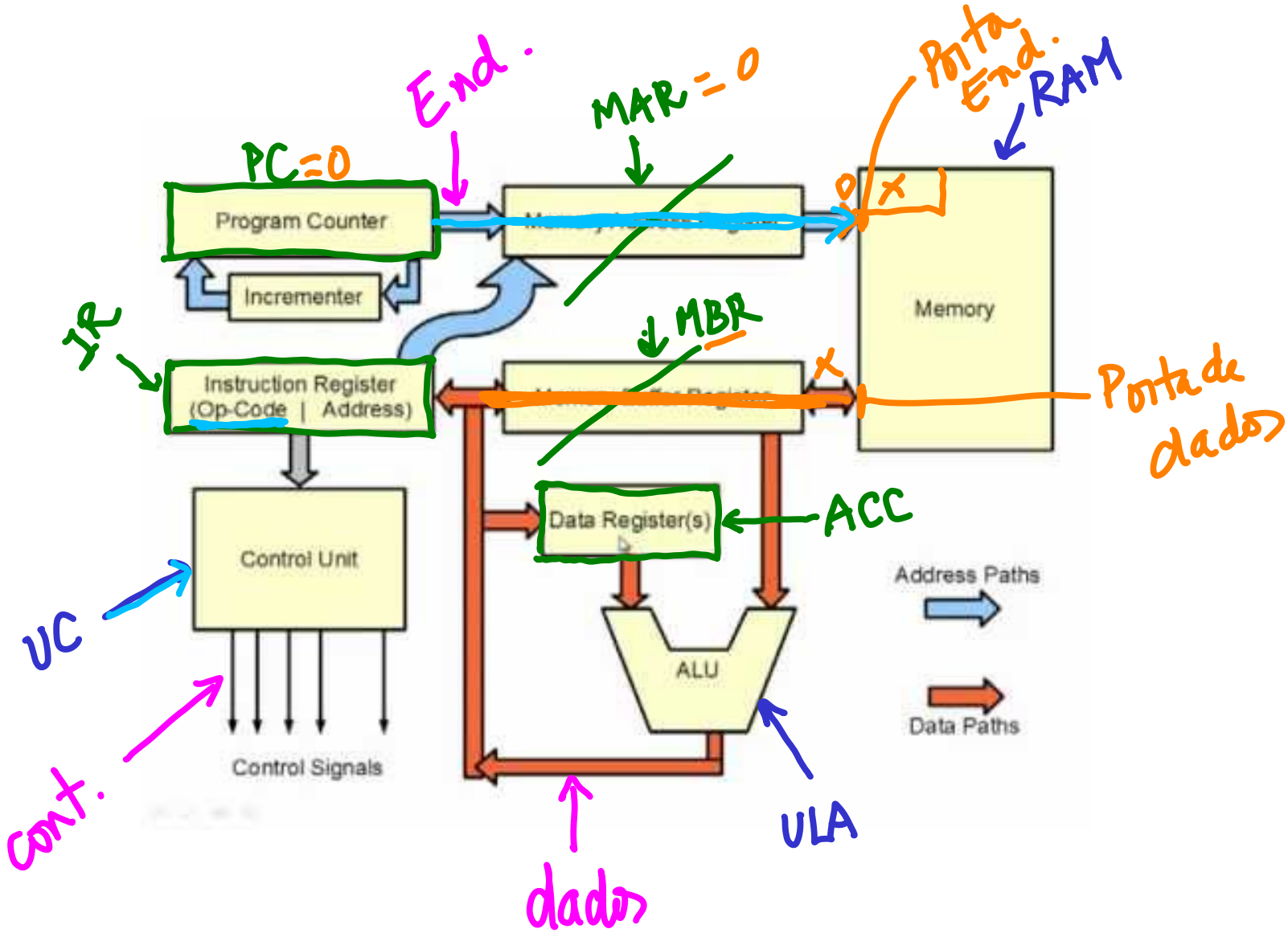
MAC0329 18/06/2020

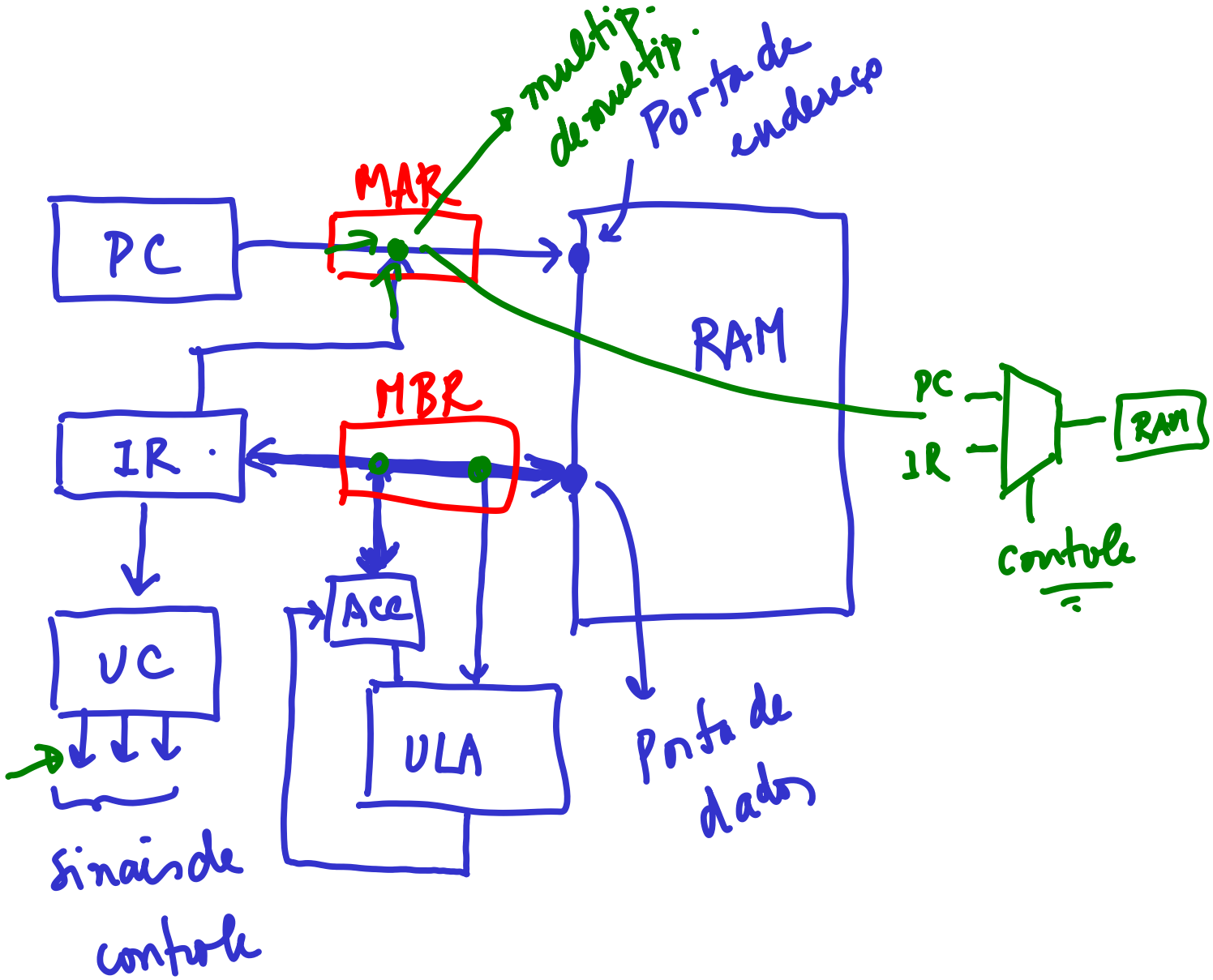
FDX - Fetch, Decode, Execute

Entendendo o funcionamento do processador ...

1. ULA
2. Unidade de controle
3. Memória (RAM)




End.
MAR = 0
Porta End.
RAM
PC = 0
Program Counter
Incrementer
IR
Instruction Register
(Op-Code | Address)
MBR
Memory
Porta de dados
Data Register(s)
ACC
Control Unit
UC
Address Paths
ALU
Data Paths
Control Signals
cont.
dados
ULA


multip.
demultip.
Porta de endereço
MAR
PC
RAM
MBR
IR
PC
IR
RAM
controle
ACC
UC
ULA
Porta de dados
sinais de controle

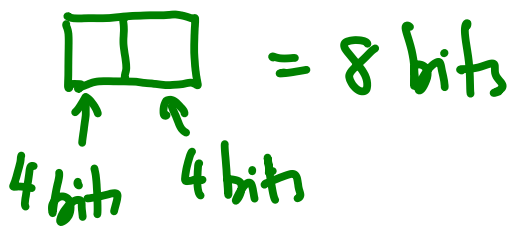
FDX (Fetch-Decode-Execute) Cycle

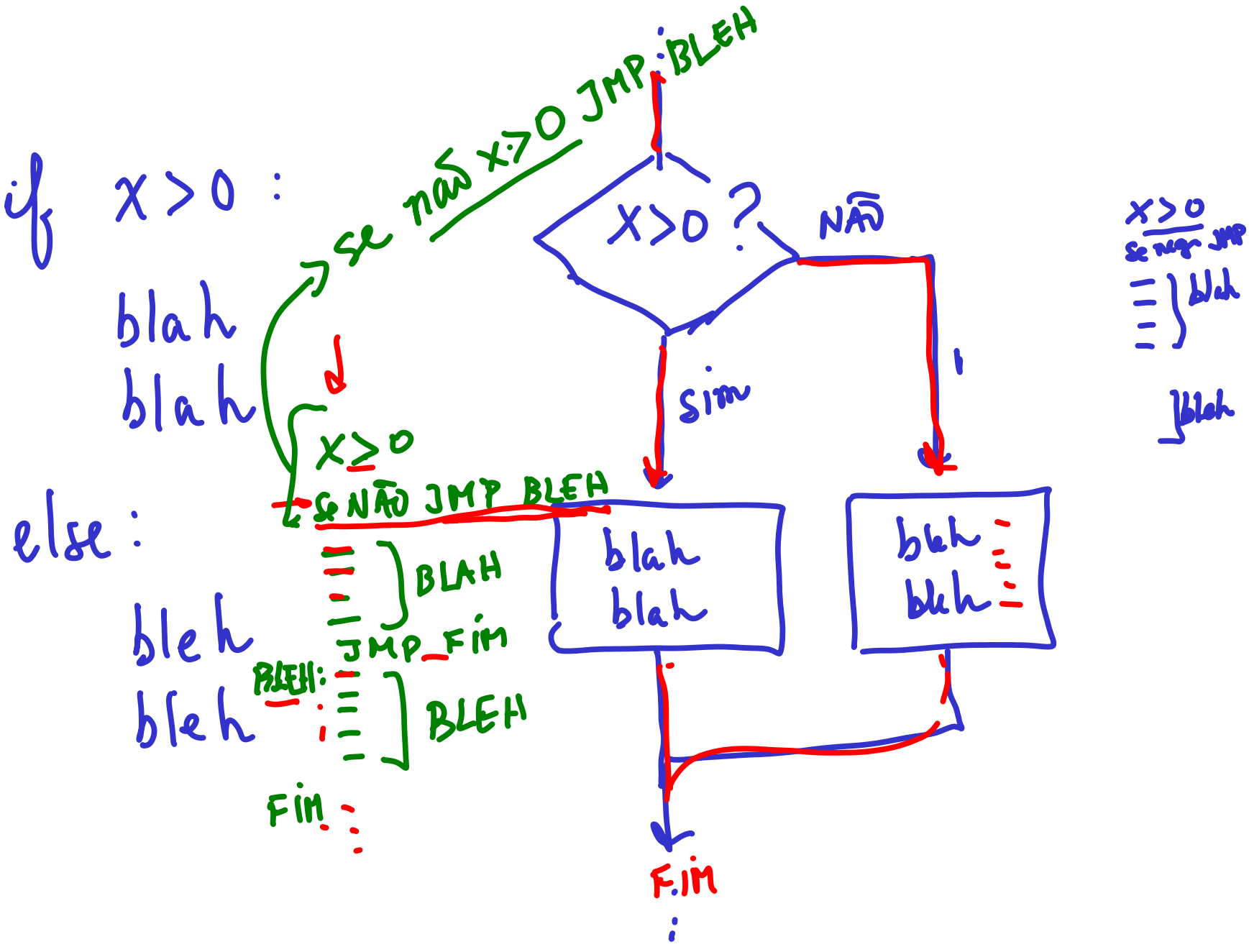
1. *Fetch*: busca-se uma instrução que está na memória RAM
   - a instrução na posição apontada pelo PC deve ser lida da memória e carregada no IR. Além disso, o valor do PC deve ser incrementado em 1.

2. *Decode*: decodifica-se a instrução (determina-se as ações exigidas pela mesma)
   - o valor dos diversos sinais de controle deve ser ajustado conforme a instrução a ser executada (por exemplo, pode ser necessário definir o modo de operação – leitura/escrita – da memória e dos registradores, os sinais que controlam os pinos seletores dos MUXes, etc).

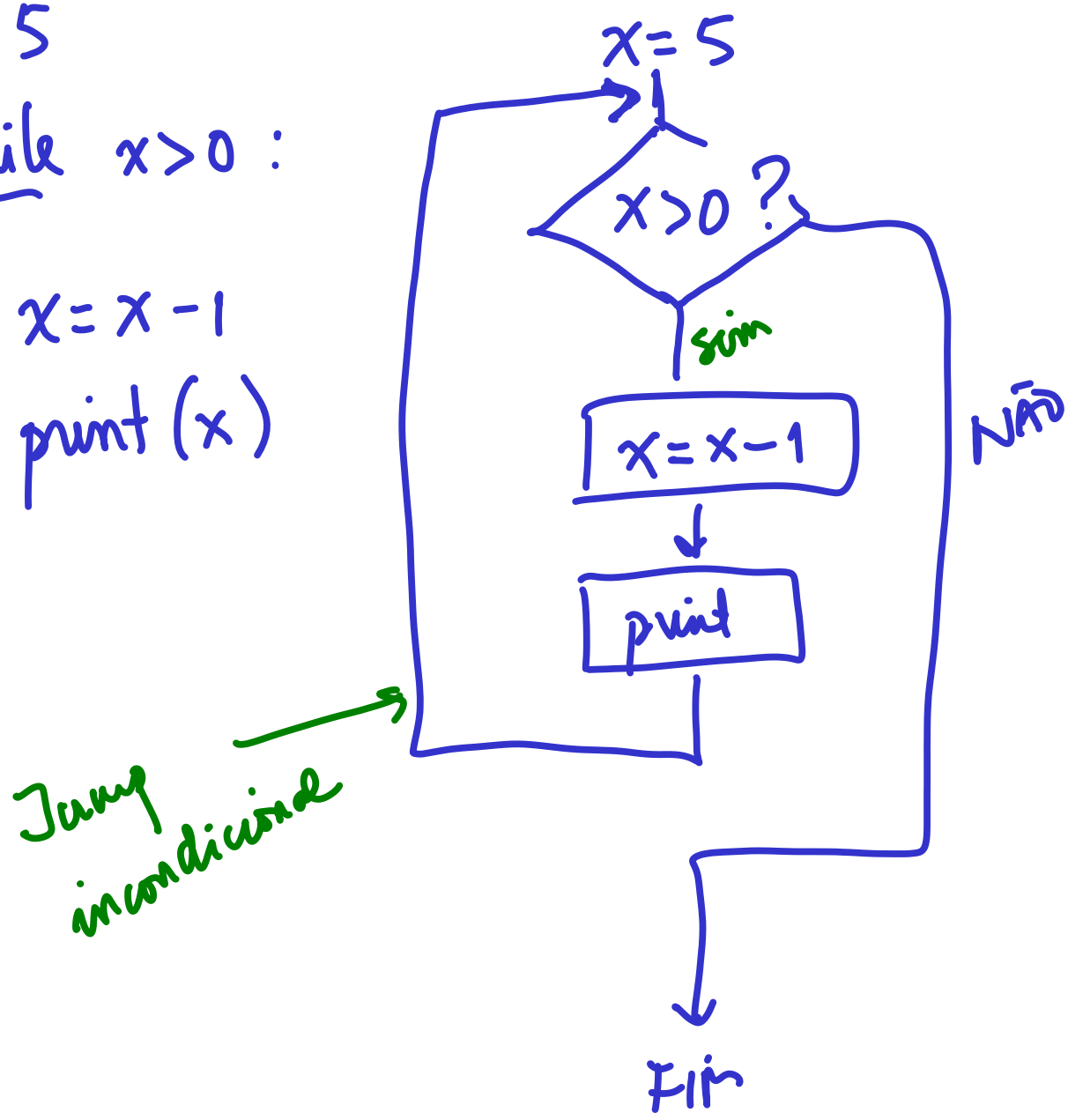
3. *Execute*: executa-se as ações determinadas pela instrução
   - Além disso, o ciclo deve ser "resetado".

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




if x>0 :
blah
blah
else :
bleh
bleh
se não x>0 JMP BLEH
X>0 ?
NÃO
SIM
X≥0
SE NÃO JMP BLEH
BLAH
JMP FIM
BLEH:
BLEH
FIM
blah
blah
bleh
bleh
FIM
X≥0
SE NÃO JMP
blah

```
x = 5
while x > 0 :
    x = x - 1
    print(x)
```

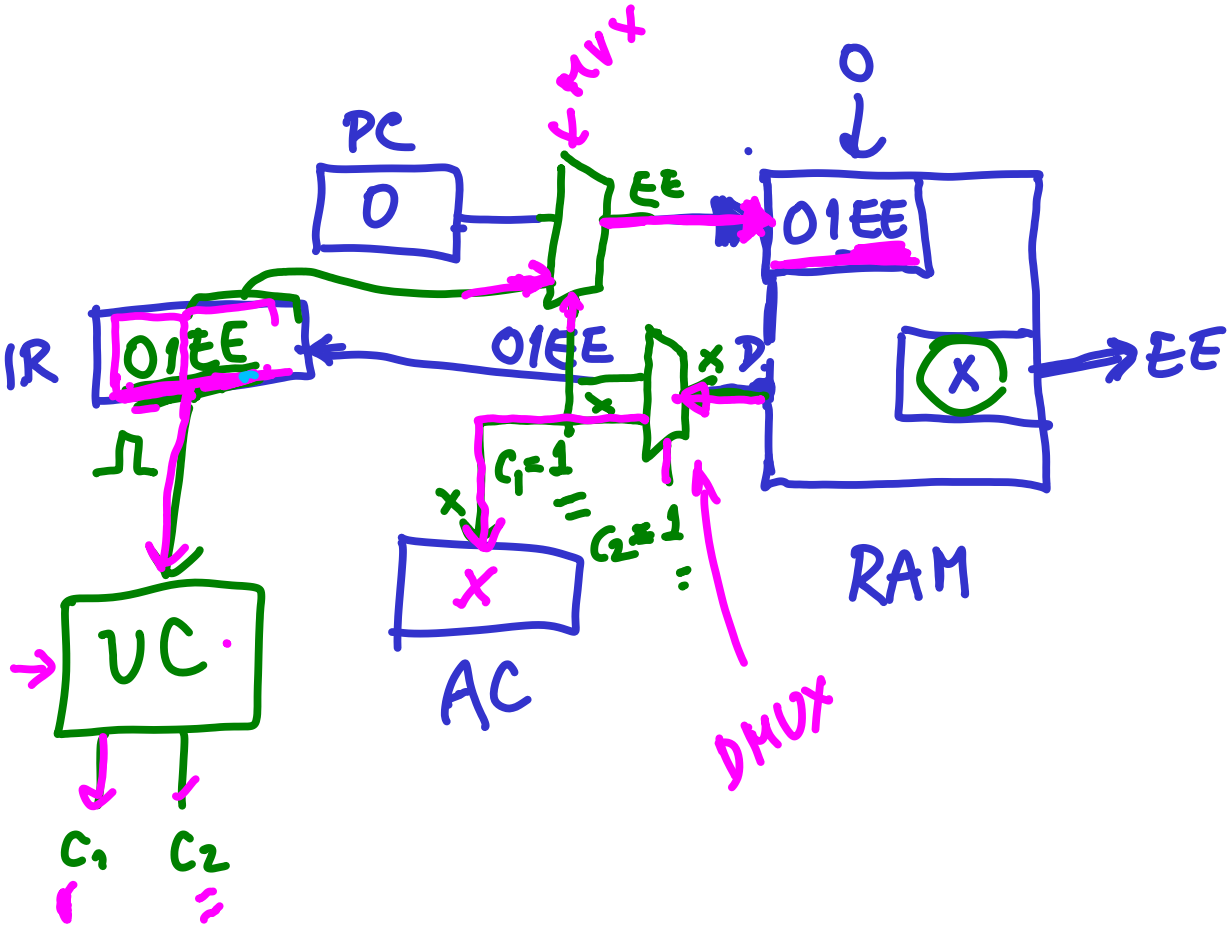


0→ Copie [EE] p/ o AC

01EE

Início do ciclo

1. Fetch
2. Decode
3. Execute



02 → Desvie p/ EE (OAEE)

Antes

PC = 02

Porta End RAM ← 02

Porta de dados RAM = OAEE

1. Fetch ⊓ + incremento do PC

IR ← OAEE

2. Decode →

3. Executa ⊓

PC = EE