

Comic Sans 12 carai!! CHEGA DE ABNT!! HOMENAGEM PARA JOY

OS ASSUNTOS ESTÃO SEPARADOS EM TÍTULOS VERMELHOS, N  
PERDE TEMPO NAS PARADA Q VC MANJA Q TEM COISA PRA K7 AI

Revisão pra prova de arquitetura de fundamentos de análise de pessoas de  
gestão de desenvolvimento de pq vc ainda esta lendo isso vai logo paras  
letras vemelhas e não perca tempo pq to escrevendo aqui só pra  
procrastinar enquanto meu vape carrega, WAAAAAAAAAAAAAAAAAAAAA

parte mais fácil começa por aqui que ta safe

SE POR ACASO VOCÊ FOI EM ALGUMA AULA PODE PULAR PARA  
O "RESUMO HEXA PARA BINÁRIO"

## CONVERSÃO HEXADECIMAL E BINÁRIO

0x3F = 0011 1111

Para converter é comum transformar o hexa em decimal mentalmente<sup>1</sup>, F é  
igual a 15.

Distribuindo em Binário: sempre comece usando os bits mais significativos,  
no caso da conversão, se o numero ultrapassar 8, use o primeiro bit da  
esquerda, 1000, em seguida + 4 seria o segundo bit mais significo, 1100, é  
útil subtrair gradualmente os números da base que quer converter, nesse  
caso,  $15 - (8 + 4) = 3$ , usando mais 1 bit = 1110, que vale 2, agora temos  $3 - 2 = 1$ , o  
primeiro bit é 1, formando 1111 como resultado da conversão do primeiro  
digito do hexadecimal. Duvidas nos valores dos bits? o rodapé<sup>2</sup> ;)

Caso tenha ido na aula, mas se distraiu com o celular ou com pessoas que  
tiram cravos da cara dos colegas durante a aula, sim isso aconteceu! :D

## RESUMO HEXA PARA BINÁRIO

F é 1111 em binario e 3 é 0011 em binário, para converter é só separar em  
grupos DE4 ;) da direita para esquerda (Gatilho de memória, DIREITA DE  
4)

- 1- Os números em hexa não passam de 15 (F) que é o maior número possível, portanto  
é muito fácil converter 1 digito de hexa para decimal.

- 2- lembrete, valores dos binários da direita para esquerda são múltiplos de 2 exceto 1, os primeiros bits são: 1,2,4,8,16,32,64,128,254 foram usados nas aulas até aqui.

## CONVERTER DO BINÁRIO PARA O HEXA

pegando pela mão no limite da ignorância

Separa em grupos de 4 da direita pra esquerda, soma os bits em decimal e converte pra binário

UM NÚMERO ASSIM POR EXEMPLO:

00111100

VOCE FAZ ASSIM

00111100

nessa parte tem gente que confunde, cada grupo de 4 começa os bits do 0, nesse caso o **GRUPO** começa com os valores dos bits como 1,2,4,8 e o segundo **GRUPO** também. olha o exemplo;

PESO DOS BITS: 8 4 2 1      8 4 2 1

VALOR:      0011      1100

Após essa explicação irritantemente clara e concisa voce pode somar os valores dos bits, da direita pra esquerda o primeiro grupo esta ocupando os bits com peso 4 e 8,  $4+8 = 12$  decimal, 12 em HEXA é C, o segundo grupo está ocupando os bits com peso 2 e 1,  $2+1 = 3$  decimal, HEX é 3 mesmo, então a conversão fica 0x3C.

## CÁLCULOS BÁSICOS:

### Soma de Binários

0101 1010 + 0011 1111

1 1 1 1 1 1

01011010

$$\begin{array}{r} + 00111111 \\ 10011001 \\ \hline \end{array}$$

Os <sup>1</sup> mostrando quando subiu um valor para a próxima casa, Combinações que geram esse 1 adicional são: 1+1, o resultado é 0 e vai <sup>1</sup> para próxima casa, e 1+1+<sup>1</sup> que o resultado é 1 e sobe mais <sup>1</sup> para próxima casa.

### Subtração de Binários:

$$\begin{array}{r} 1011 \\ - 0110 \\ \hline 1 \\ 0101 \end{array}$$

Os casos são:  $1 - 1 = 0$ ,  $1 - 0 = 1$ , e o mais zika é o  $0 - 1$ , nesse caso o resultado é 1 porém tem que colocar **1** para subtração na próxima casa.

### MULTIPLICAÇÃO DE BINÁRIO:

se você leu até aqui consegue deduzir como fazer :D

**ASSEMBLY DO Z80**

**CERTEZA QUE CAI NA PROVA**

**CRITICAL SECTION DON'T FUCKING PANIC**

PEGA NA MINHA MÃO E VEM QUE SE BRILHA ;D

INTRUÇÕES:

LD = LOAD, LE ALGUMA PARADA

ADD = OPERAÇÃO DE ADIÇÃO

ADC = SOMA COM VAI 1, USAR PARA SOMAR VALORES DE 16 BITS

INC = ADICIONA 1 AO VALOR EXISTENTE

SUB = SUBTRAI UM VALOR DO REGISTRADOR A

REGISTRADORES:

A = ACUMULADOR, OS RESULTADOS DE TODAS AS OPERAÇÕES VEM PARA ELE

H = PRIMEIRA PARTE DO BARRAMENTO DE ENDEREÇO, RESPONSÁVEL PELOS 2 PRIMEIROS DÍGITOS.

L = SEGUNDA PARTE DO BARRAMENTO DE ENDEREÇO, RESPONSÁVEL PELOS 2 ÚLTIMOS DÍGITOS.

REGISTRADORES AUXILIARES:

B,C,D = GUARDAM 8 BITS

### ASM DO Z80 SOMA SIMPLES

```
1- LD    H,    20
2- LD    L,    30
3- LD    A,    (HL)
4- INC   L,
5- LD    B,    (HL)
6- ADD   A,    B
7- LD    (HL), A
```

#### LEGENDA:

INSTRUÇÃO: serve pra fazer alguma instrução, tipo, ler, adicionar, subtrair

REGISTRADORES: armazenam dados igual uma variável de linguagem de programação, porém tem usos específicos.

DADOS: São valores, números etc...

- 1- Primeira linha é atribuído o número (dados) 20 ao H
- 2- Segunda linha é atribuído o número (dados) 30 ao L
- 3- Terceira linha é atribuído o valor em memória de (HL) ao registrador A, o (HL) entre parênteses se refere ao que ele aponta em memória<sup>1</sup> e não ao valor 20 e 30 que foi atribuído para ele nas linhas anteriores.
- 4- Linha Quatro, Incrementa o valor de L em 1
- 5- Quinta Linha faz a mesma coisa com que fez com o registrar A, porém com o B.
- 6- Sexta linha faz a soma entre os registradores A e B, O resultado é armazenado em A, **TODOS OS RESULTADOS VÃO PARA A.**
- 7- Escreve o valor de A no endereço de memória de HL.

- 1- Quando é usado o (HL), ele está se referindo ao endereço para qual ele está apontando, se o HL foi carregado com 2030, ele vai buscar um valor que está gravado na memória no endereço 2030.

## ARQUITETURA

### COMPONENTES:

memória volátil: todo tipo de memória que apaga com falta de energia

memória permanente: não apaga com falta de energia

ULA = unidade de lógica aritmética, é onde fazem os cálculos

Registradores = onde ficam guardados dados no processador

Unidade de controle

contador de programa = tipo de ponteiro que incrementa 1 toda vez que é usado

registrador de instruções: tipo de registrador específico para instruções

Barramento de dados = por onde passam os dados 8 bits, caso do z80

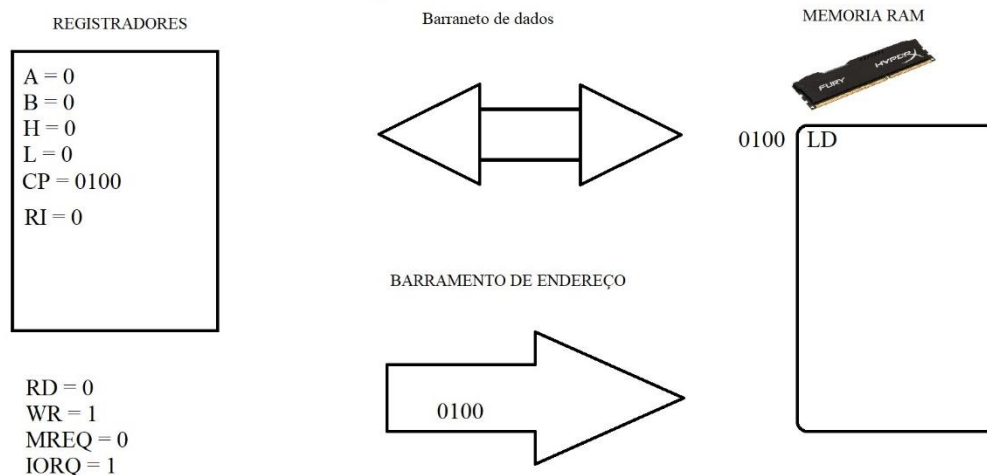
Barramento de endereço = onde passa os endereços, 16bits no caso do z80

## FUNDAMENTOS

ciclo de processamento : Buscar, Decodificar e Executar - Pode ser chamado de Clock

## FLUXO COMPLETO DO ASSEMBY ACIMA

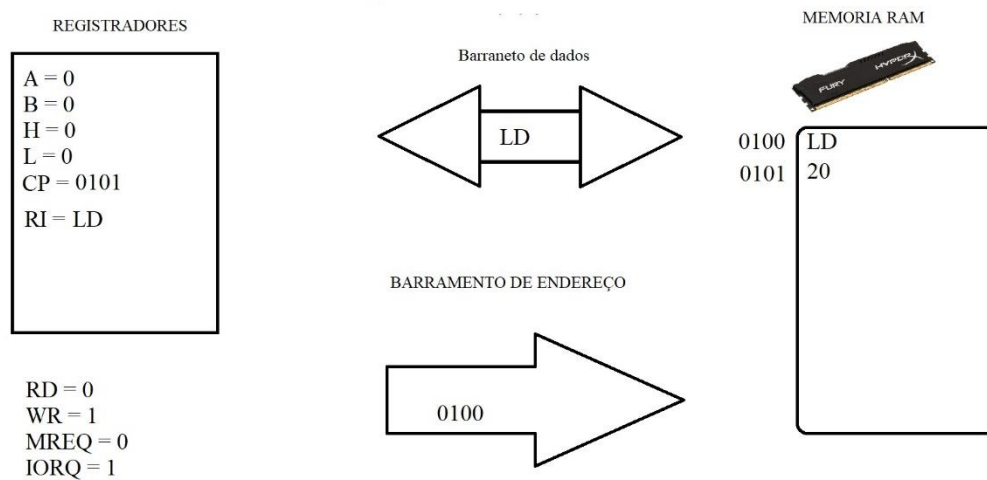
vou considerar que o programa começa no endereço de memória 0x100, que todos os registradores estão zerados.



IMG 1.

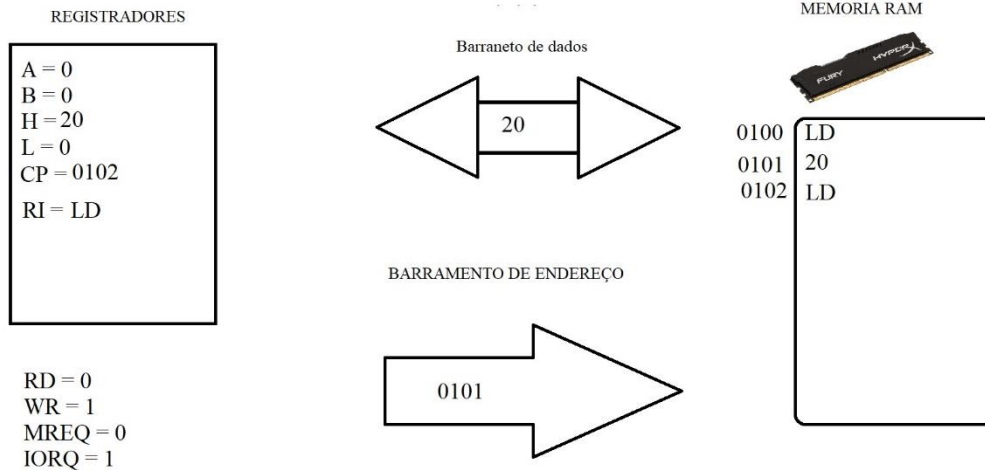
Observação: RD e WR correspondem ao comando de READ e WRITE, eles são barrados, logo se o valor for 1 estão desligados e 0 estão ligados, o mesmo para o MREQ e IORQ que são em qual lugar ele vai ler ou escrever, memória ou dispositivos de entrada e saída. O RI é o Registrador de Instruções.

Na IMG 1 o contador de programa (CP) coloca seu valor no barramento de dados, buscando na memória a primeira instrução, observe que para buscar na memória RD recebe 0 e MREQ recebe 0, pois são barrados o que significa q vão funcionar.



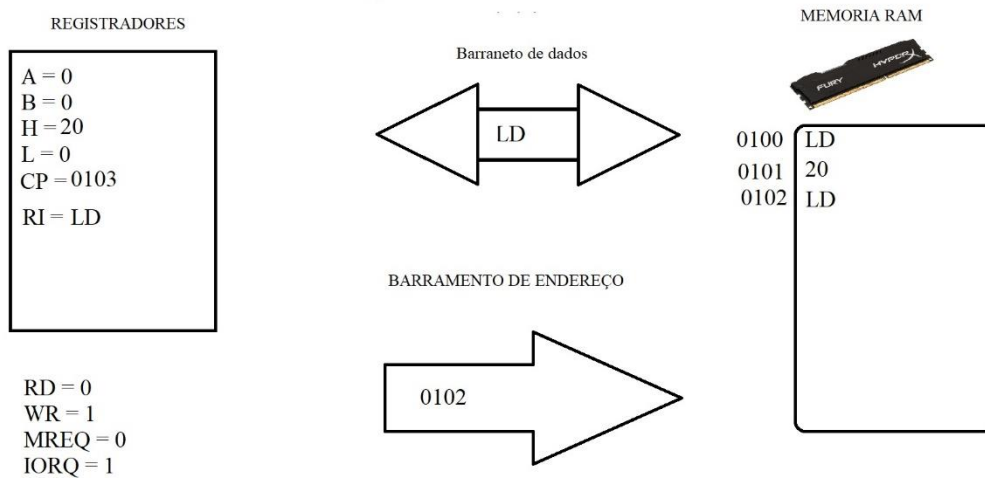
## IMG2

Na IMG2, observe o valor de CP, o contador de programa foi incrementado em 1, o LD que estava na memória passou pelo barramento de dados e foi salvo no Registrador de Instruções.



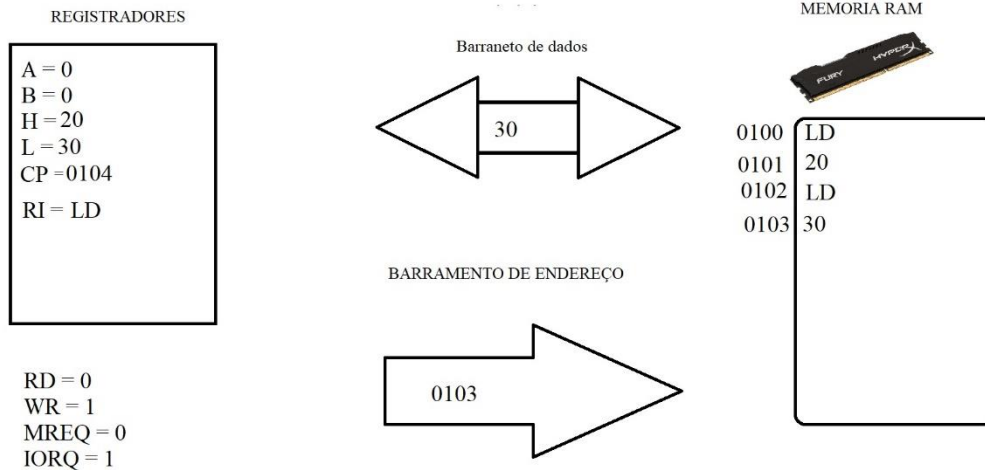
### IMG3

Na IMG3 o valor de 20 passou pelo barramento de dados e foi salvo em H. CP foi incrementado em 1



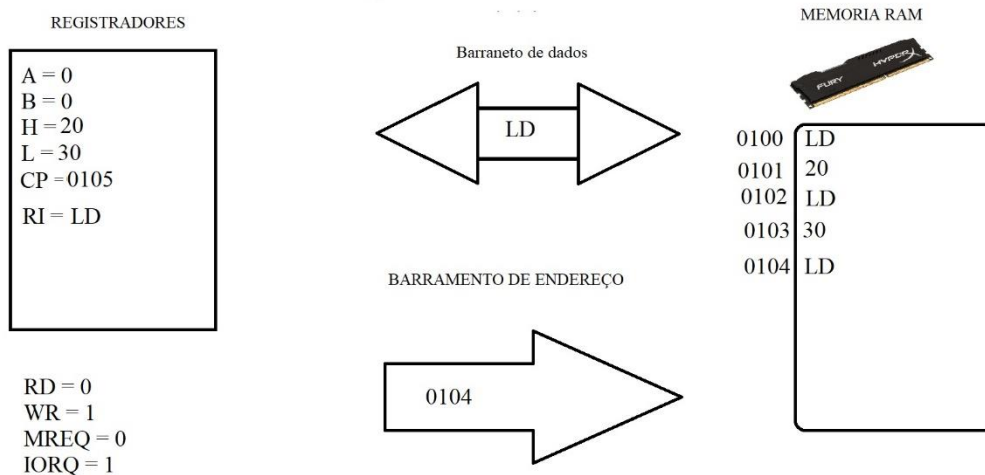
### IMG4

Na IMG4 foi colocado o valor de CP no barramento de endereço e o valor LD da segunda instrução foi salvo no Registrador de instrução. o CP foi incrementado em 1, não vou falar q a cada ciclo o CP é incrementado em 1 pq essas horas vc já sacou isso.



## IMG5

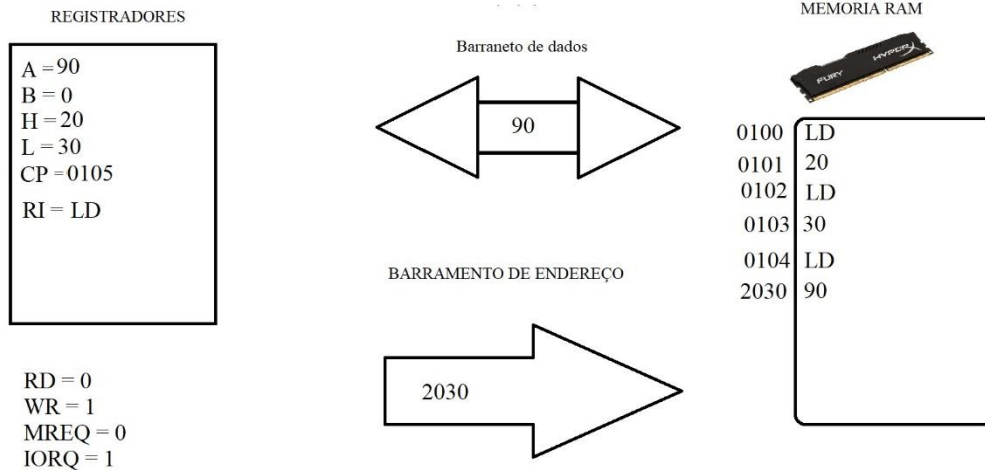
Na IMG5 O valor do endereço 0103, 30, foi movido para o registrador L, AGORA ESTAMOS NA 2 DO NOSSO PROGRAMA DE SOMA SIMPLES. VEM CMG QUE VAI DAR TUDO CERTO :D



## IMG6

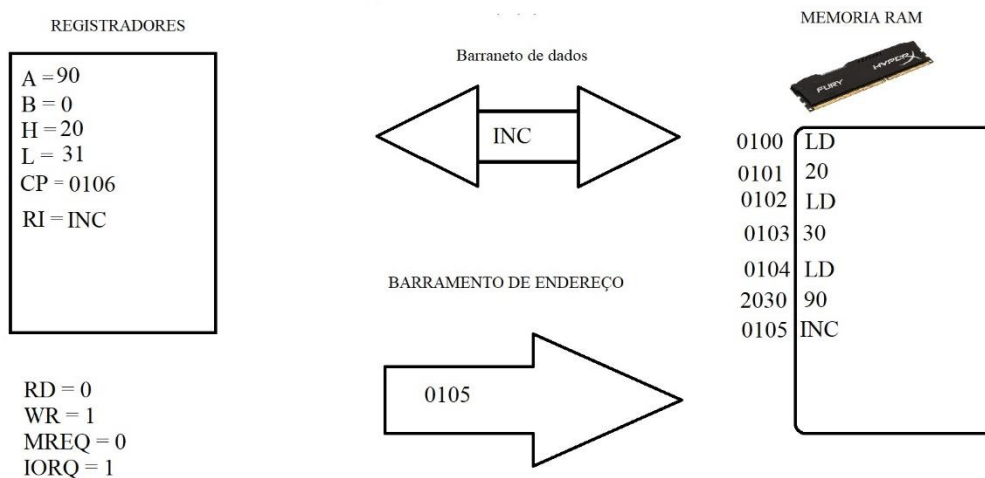
Na IMG6 O FOI COLOCADO O ENDERO 0104 no barramento de endereços, retornou LD para o Registrador de Endereço pelo barramento de dados.





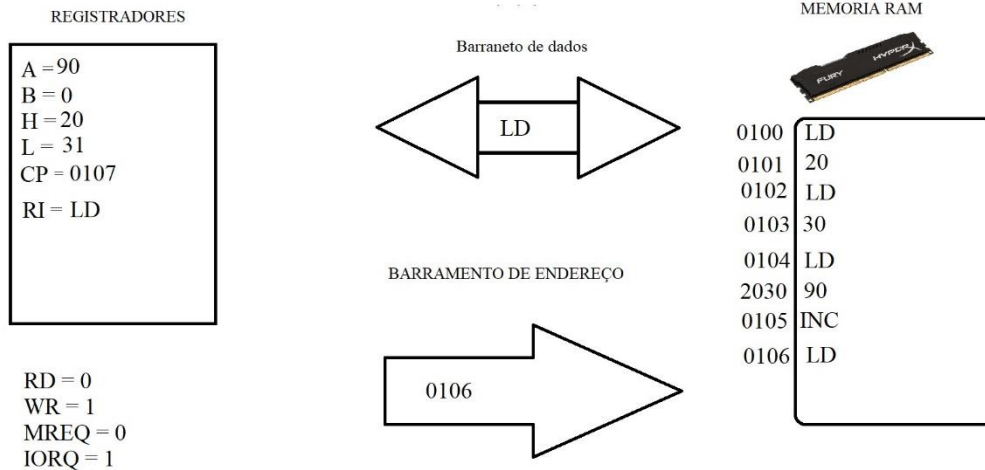
## IMG7

A IMG7 é equivalente a linha 3 do nosso programa, carrega o HL no barramento de endereços, pega o valor da memória e coloca em A, na memória os endereços são sequenciais, nesse caso coloquei na sequência de execução apenas para ser mais compreensível.



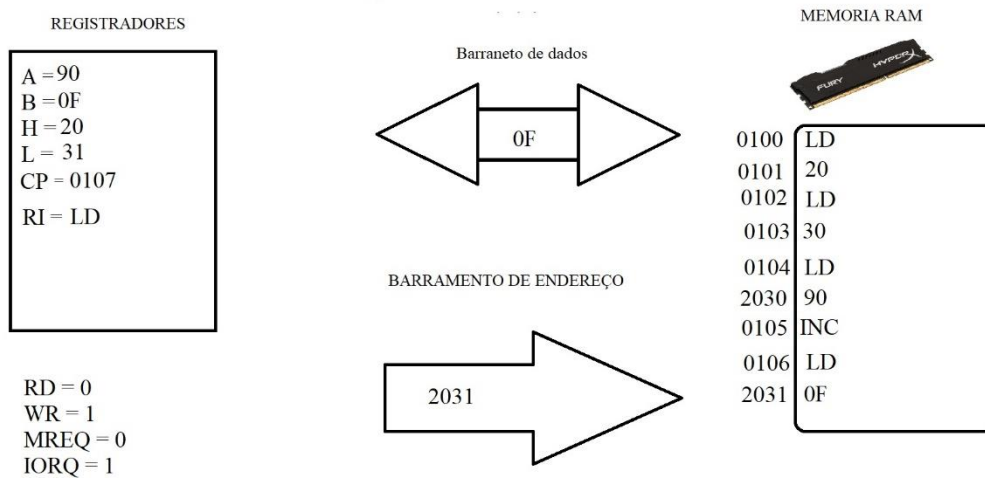
## IMG7

Na IMG7 equivalente a linha 4 do programa coloca o valor de CP no barramento de endereço, encontra a instrução INC em memória e coloca no Registrador de instrução, incrementa o valor de L em 1.



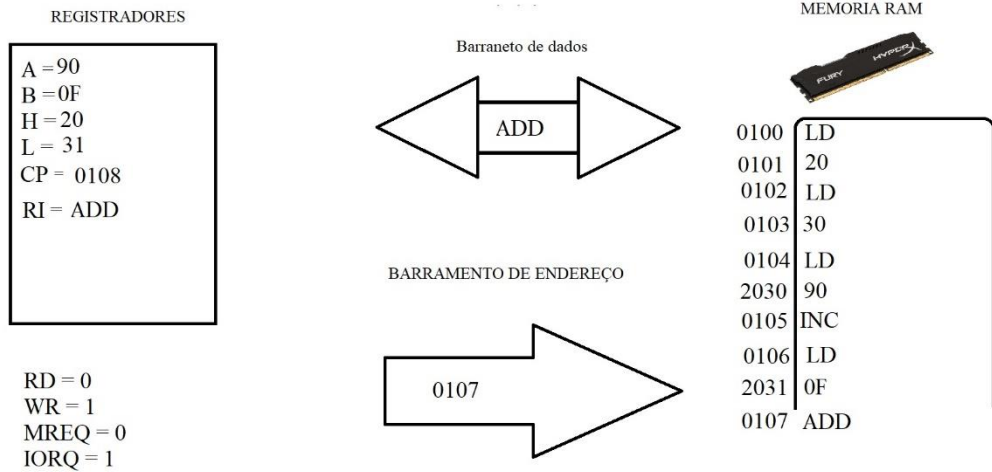
## IMG8

Na IMG8 Coloca o CP no barramento de endereço, O LD no Registrador de instruções.



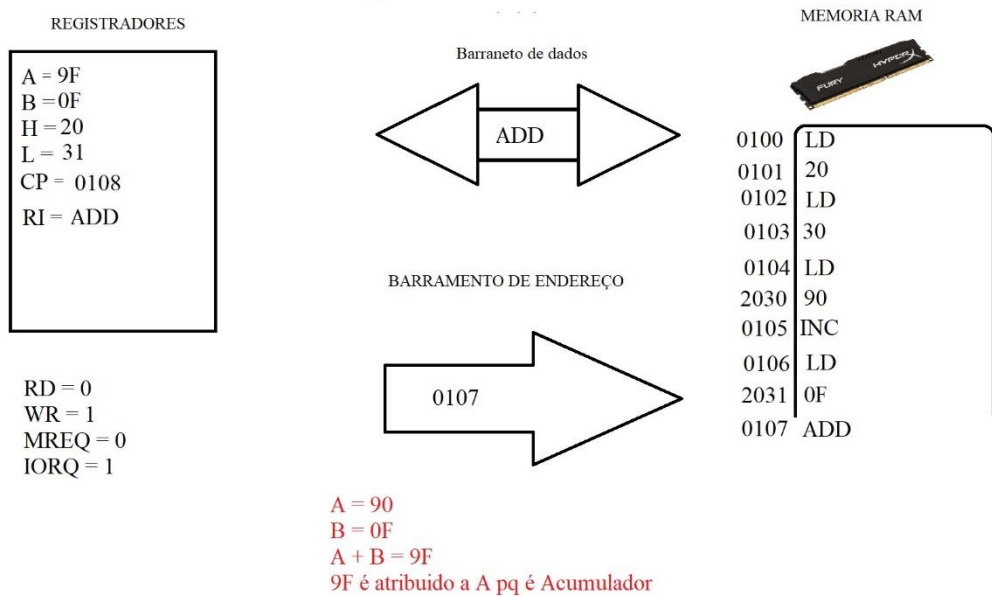
## IMG9

Equivalente a linha 5 do nosso programa ele move o (HL) para B, colocando o valor de HL no barramento de endereço e o valor 0F da memória no barramento de dados.



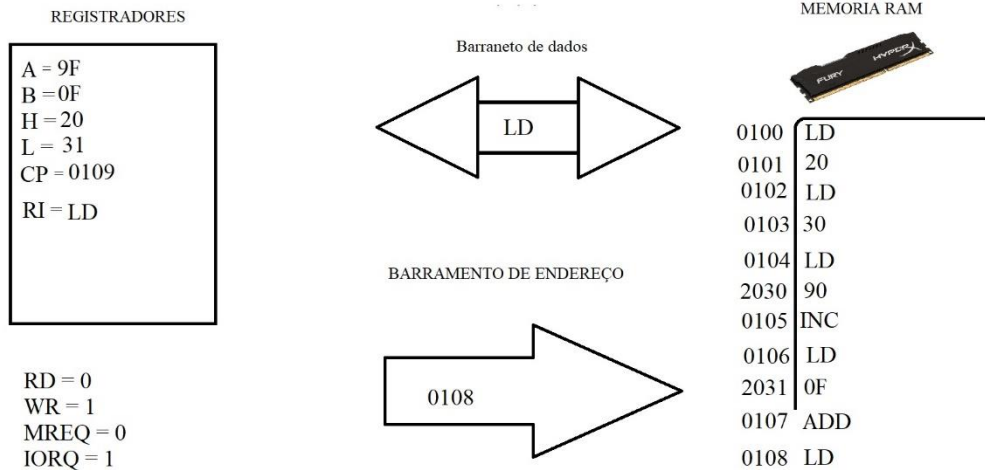
## IMG10

Coloca ADD no registrador de Instrução.



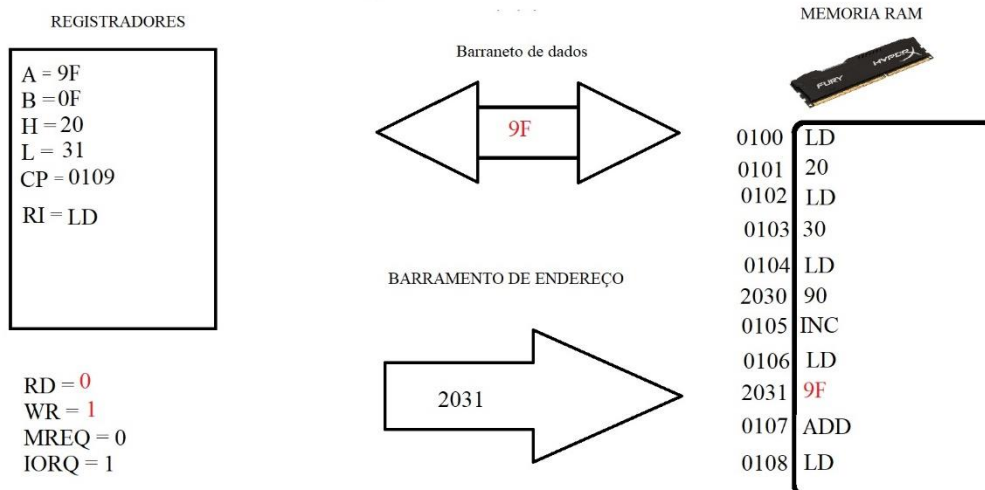
## IMG11

Equivalente a linha 6 do nosso algoritmo, Soma A com B, o resultado é salvo em A.



## IMG12

Salvou LD em RI.



## IMG13

EQUIVALENTE A ÚLTIMA LINHA, SALVA GRAVA O VALOR DE A EM HL, EU SEI FICA BEM MELHOR EM VERMELHO MOSTRANDO O QUE MUDOU EM CADA FIGURA, MAS EU PENSEI NISSO SÓ NA ULTIMA FIGURA AI JA TINHA FEITO N VOU FAZER DNV, mas se prestar atenção da pra sacar sim.

Para você que chegou até aqui boa sorte!!!  
:D