

# ARQUITETURA DE COMPUTADORES

Microprocessador INTEL 8085  
(exercícios)

# Conteúdos

2

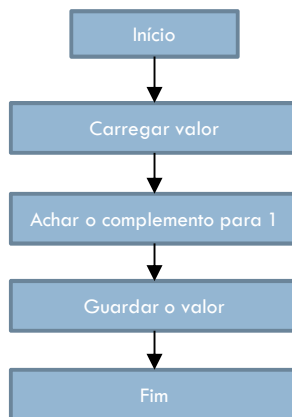
## 1. Programação do microprocessador da INTEL 8085

Nota: as resoluções constantes nos slides seguintes representam propostas de resolução, o que significa que as resoluções apresentadas não são únicas.

# Exercícios

3

- 1. Ache o complemento para um do conteúdo de uma posição de memória (1040H) e guarde o resultado na posição de memória seguinte (1041H).

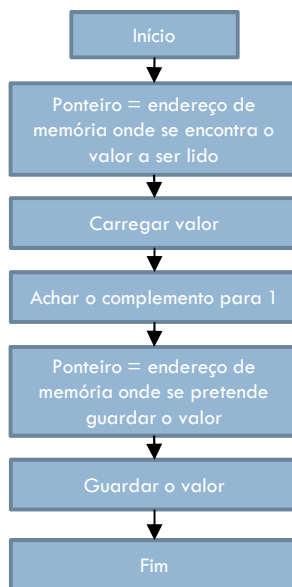


Código	Comentário
LDA 1040H	$[A] \leftarrow [1040H]$
CMA	$[A] \leftarrow [\bar{A}]$
STA 1041H	$[1041H] \leftarrow [A]$
HLT	END

# Exercícios

4

- 2. Ache o complemento para um do conteúdo de uma posição de memória (1040H) e guarde o resultado na posição de memória seguinte (1041H), utilizando o par de registos HL como apontador de memória.

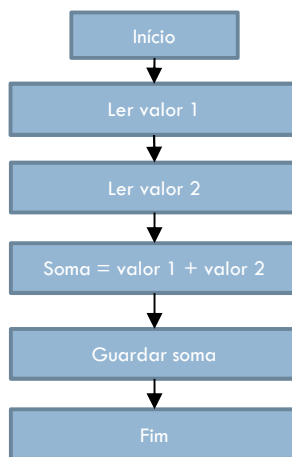


Código	Comentário
LXI H, 1040H	[HL] ← 1040H
MOV A, M	[A] ← [[HL]]
CMA	[A] ← [A̅]
LXI H, 1041H	[HL] ← 1041H
MOV M, A	[1041H] ← [A]
HLT	END

# Exercícios

5

- 3. Adicione o conteúdo de duas posições de memória consecutivas (1040H e 1041H) e guarde o resultado na posição de memória seguinte (1042H).

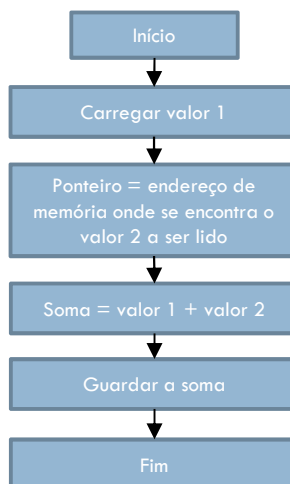


Código	Comentário
LDA 1040H	$[A] \leftarrow [1040H]$
MOV B, A	$[B] \leftarrow [A]$
LDA 1041H	$[A] \leftarrow [1041H]$
ADD B	$[A] \leftarrow [A] + [B]$
STA 1042H	$[1042H] \leftarrow [A]$
HLT	END

# Exercícios

6

- 4. Adicione o conteúdo de duas posições de memória consecutivas (1040H e 1041H) e guarde o resultado na posição de memória seguinte (1042H), utilizando o par de registos HL como apontador da memória.

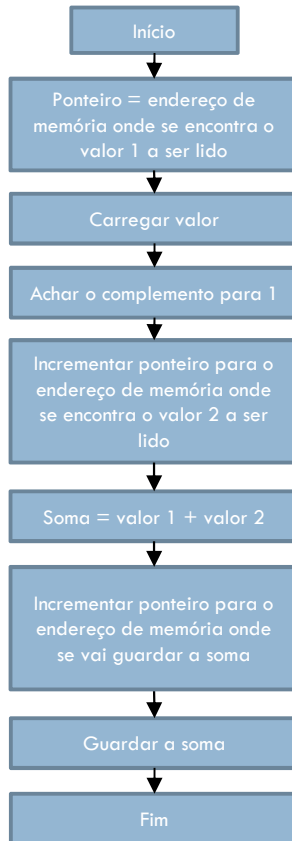


Código	Comentário
LDA 1040H	$[A] \leftarrow [1040H]$
LXI H, 1041H	$[HL] \leftarrow 1041H$
MOV B, M	$[B] \leftarrow [[HL]]$
ADD B	$[A] \leftarrow [A] + [B]$
STA 1042H	$[1042H] \leftarrow [A]$
HLT	END

# Exercícios

7

- 5. Adicione o conteúdo de duas posições de memória consecutivas (1040H e 1041H) e guarde o resultado na posição de memória seguinte (1042H), utilizando o par de registos HL como apontador da memória e a instrução de incrementação (INX)

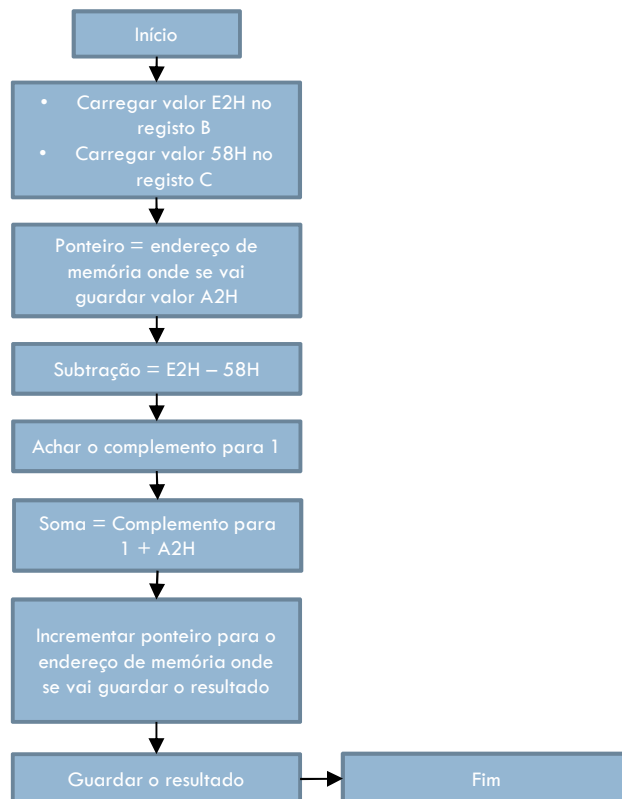


Código	Comentário
LXI H, 1040H	$[HL] \leftarrow 1040H$
MOV A, M	$[A] \leftarrow [[HL]]$
INX H	$[HL] \leftarrow [HL] + 1$
ADD M	$[A] \leftarrow [A] + [[HL]]$
INX H	$[HL] \leftarrow [HL] + 1$
MOV M, A	$[[HL]] \leftarrow [A]$
HLT	END

# Exercícios

8

6. Elabore um programa que carregue os números E2H e 58H nos registros B e C, respectivamente. Guarde o número A2H na posição de memória 1065H, usando o par de registros HL como apontador da memória. Realize a subtração entre E2H – 58H. Ache o complemento do resultado da subtração obtida anteriormente, e some esse valor com o número A2H contido na posição de memória 1065H. Finalmente guarde o resultado na posição de memória 1066H.



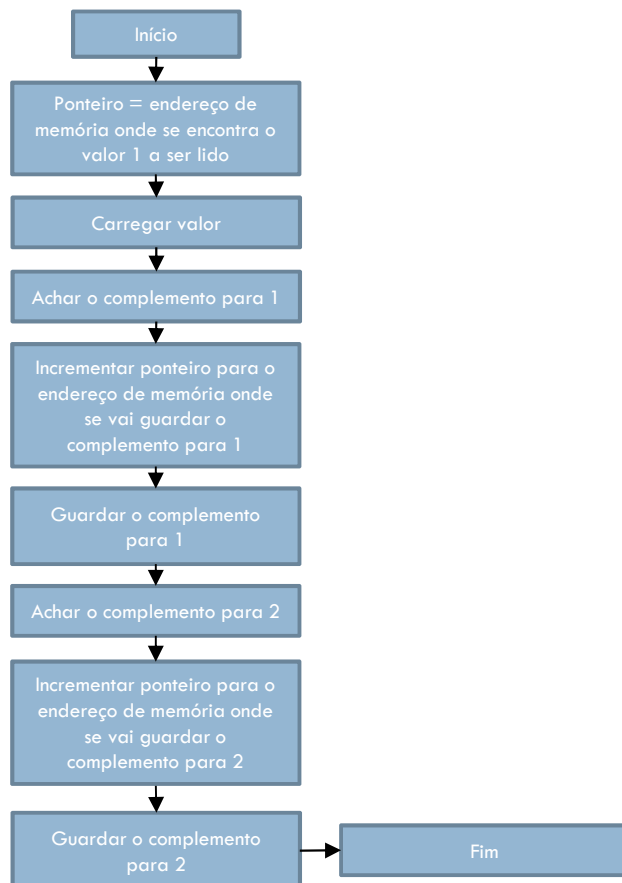
Código	Comentário
LXI B, E258H	$[BC] \leftarrow E258H; [B] \leftarrow E2H; [C] \leftarrow 58H$
LXI H, 1065H	$[HL] \leftarrow 1065H$
MVI M, A2H	$[[HL]] \leftarrow A2H$
MOV A, B	$[A] \leftarrow [B]$
SUB C	$[A] \leftarrow [A] - [C]$
CMA	$[A] \leftarrow \overline{[A]}$
ADD M	$[A] \leftarrow [A] + [[HL]]$
INX H	$[HL] \leftarrow [HL] + 1$
MOV M, A	$[[HL]] \leftarrow [A]$
HLT	END



# Exercícios

9

7. Elabore um programa que leia o conteúdo de uma posição de memória (1040H) e que guarde nas duas posições de memória seguintes (1041H e 1042H) o complemento para um e o complemento para dois do valor lido.

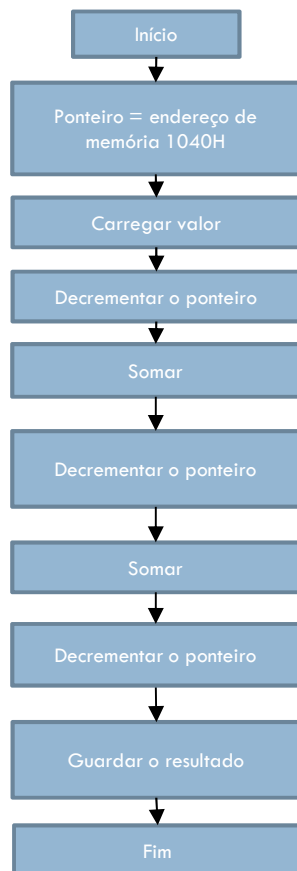


Código	Comentário
LXI H, 1040H	$[HL] \leftarrow 1040H$
MOV A, M	$[A] \leftarrow [[HL]]$
CMA	$[A] \leftarrow [\bar{A}]$
INX H	$[HL] \leftarrow [HL] + 1$
MOV M, A	$[[HL]] \leftarrow [A]$
INR A	$[A] \leftarrow [A] + 1$
INX H	$[HL] \leftarrow [HL] + 1$
MOV M, A	$[[HL]] \leftarrow [A]$
HLT	END

# Exercícios

10

8. Elabore um programa que adicione os conteúdos de três posições de memória consecutivas (1038H, 1039H e 1040H), guarde o resultado da soma na posição de memória anterior à ocupada pelas parcelas a adicionar.

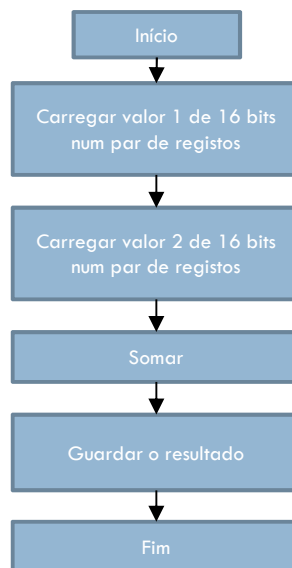


Código	Comentário
LXI H, 1040H	[HL] ← 1040H
MOV A, M	[A] ← [[HL]]
DCX H	[HL] ← [HL] - 1
ADD M	[A] ← [A] + [[HL]]
DCX H	[HL] ← [HL] - 1
ADD M	[A] ← [A] + [[HL]]
DCX H	[HL] ← [HL] - 1
MOV M, A	[[HL]] ← [A]
HLT	END

# Exercícios

11

9. Elabore um programa que adicione dois números de 16 bits colocados cada um em duas posições sucessivas de memória (1040H e 1042H), sabendo que o byte mais significativo está na posição de memória com endereço superior. Guarde o resultado nas duas posições de memória seguintes.

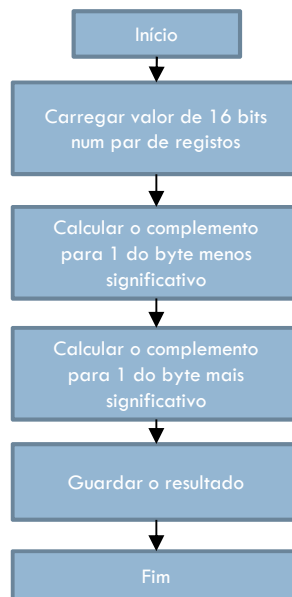


Código	Comentário
LHLD 1040H	[L] ← [1040H]; [H] ← [1041H]
XCHG	[D] ↔ [H]; [E] ↔ [L]
LHLD 1042H	[L] ← [1042H]; [H] ← [1043H]
DAD D	[HL] ← [HL] + [DE]
SHLD 1044H	[1044H] ← [L]; [1045H] ← [H]
HLT	END

# Exercícios

12

10. Elabore um programa que calcule o complemento para um de um número de 16 bits, contido em duas posições de memória consecutivas (1040H e 1041H) e coloque o resultado nas duas posições de memória seguintes (1042H e 1043H). O byte mais significativo está e deve ficar na posição de memória mais elevada.

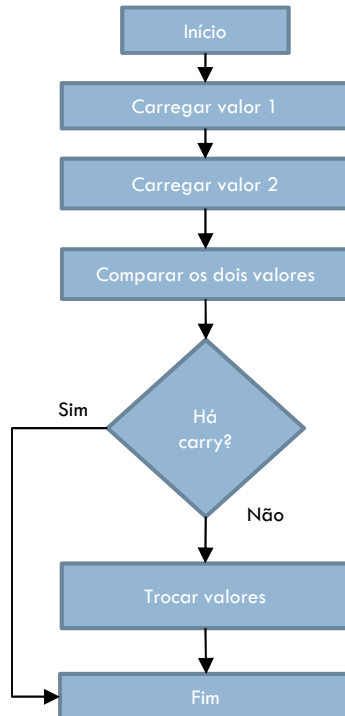


Código	Comentário
LHLD 1040H	$[L] \leftarrow [1040H]; [H] \leftarrow [1041H]$
MOV A, L	$[A] \leftarrow [L]$
CMA	$[A] \leftarrow [\bar{A}]$
MOV L, A	$[L] \leftarrow [A]$
MOV A, H	$[A] \leftarrow [H]$
CMA	$[A] \leftarrow [\bar{A}]$
MOV H, A	$[H] \leftarrow [A]$
SHLD 1042H	$[1042H] \leftarrow [L]; [1043H] \leftarrow [H]$
HLT	END

# Exercícios

13

11. Elabore um programa que coloque os conteúdos de duas posições de memória (1040H e 1041H) por ordem decrescente, ou seja, o maior valor em primeiro lugar.

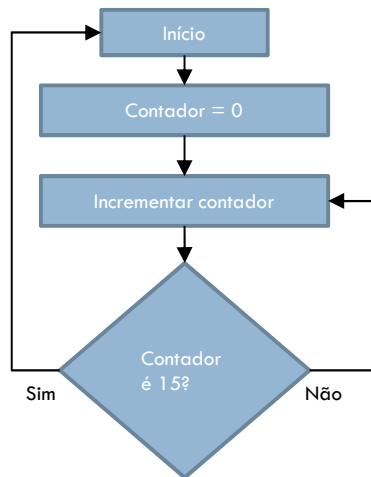


Etiqueta	Código	Comentário
	LDA 1040H	$[A] \leftarrow [1040H]$
	MOV B, A	$[B] \leftarrow [A]$
	LDA 1041H	$[A] \leftarrow [1041H]$
	CMP B	$[A] - [B]$
	JC FIM	$[PC] \leftarrow FIM$ if carry flag (CS) equal to 1
	STA 1040H	$[1040H] \leftarrow [A]$
	MOV A, B	$[A] \leftarrow [B]$
	STA 1041H	$[1041H] \leftarrow [A]$
FIM	HLT	END

# Exercícios

14

- 12. Elabore um programa que efetue a contagem de 0 a 15 e que repita o ciclo de contagem.

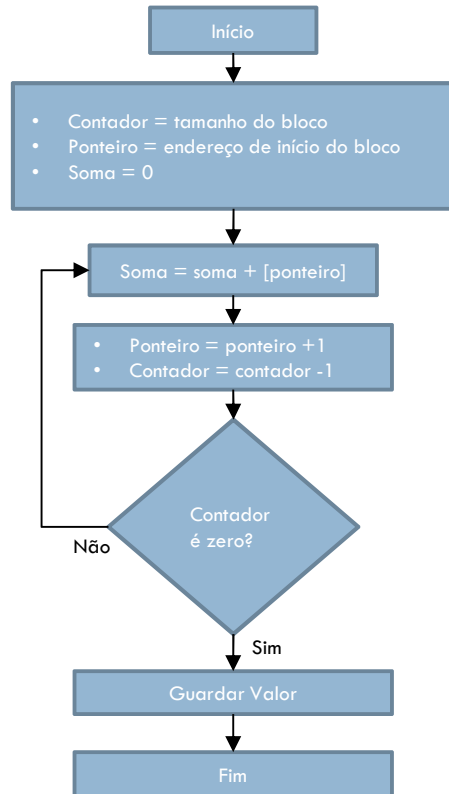


Etiqueta	Código	Comentário
INICIO	SUB A	$[A] \leftarrow [A] - [A]$ Clean the accumulator register (A)
CONT	INR A	$[A] \leftarrow [A] + 1$ Count update
	CPI 0FH	$[A] - 0FH$ Finish counting?
	JNZ CONT	$[PC] \leftarrow CONT$ if zero flag (Z) equal to 0 Finish counting = No $\rightarrow$ continue counting
	JMP INICIO	$[PC] \leftarrow INICIO$ Finish counting = Yes $\rightarrow$ restart counting
	HLT	END

# Exercícios

15

13. Elabore um programa que calcule a soma de um bloco de números. O tamanho do bloco está colocado na posição 1041H. O bloco inicia-se na posição de memória seguinte (1042H). Coloque o resultado da soma na posição de memória 1040H.

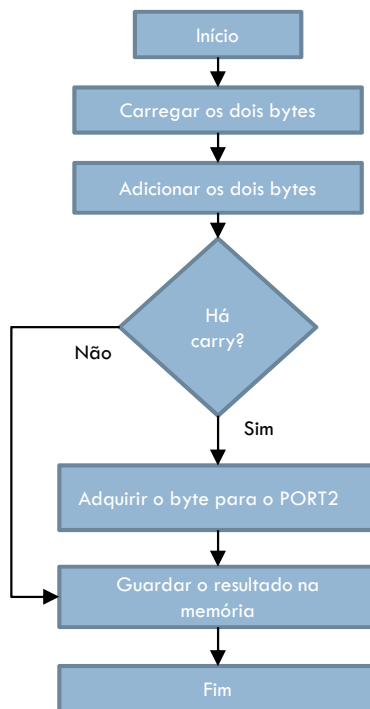


Etiqueta	Código	Comentário
	LXI H, 1041H	[HL] ← 1041H
	MOV B, M	[B] ← [[HL]] Acquires the block size
	SUB A	[A] ← [A] - [A] Clean the accumulator register (A)
CICLO	INX H	[HL] ← [HL] + 1 Register pair HL acquires new position
	ADD M	[A] ← [A] + [[HL]] Sum = Sum + data
	DCR B	[B] ← [B] - 1 Decrement register B
	JNZ CICLO	[PC] ← CICLO if zero flag (Z) equal to 0
	STA 1040H	[1040H] ← [A]
	HLT	END

# Exercícios

16

14. Elabore um programa que coloque dois bytes em hexadecimal, A9H e 7BH, nos registos B e C, e calcule a respetiva soma. Se a soma for maior que 8 bits, ou seja, se produzir carry, então coloque o número 00H no porto PORT2, cujo endereço é 02H, e na posição de memória 1080H. Caso contrário, guarde a soma apenas na posição de memória 1080H.



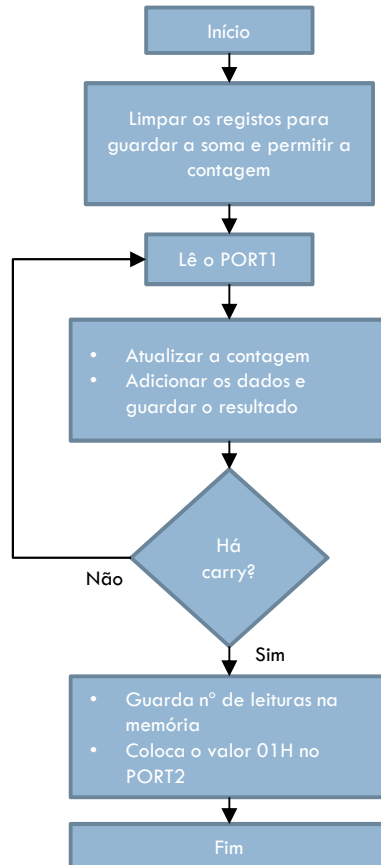
Etiqueta	Código	Comentário
	MVI B, A9H	[B] ← A9H
	MVI C, 7BH	[C] ← 7BH
	MOV A, C	[A] ← [C]
	ADD B	[A] ← [A] + [B]
	JNC GUAR	[PC] ← GUAR if carry flag (CS) equal to 0
	MVI A, 00H	[A] ← 00H
	OUT 02H	[02H] ← [A] Puts the byte in register accumulator (A) into port with address 02H (PORT2)
GUAR	STA 1080H	[1080H] ← [A]
	HLT	END



# Exercícios

17

15. Elabore um programa que efetue a leitura de dados provenientes do porto PORT1, cujo endereço é 01H, que conte o número de leituras realizadas e efetue a soma dos bytes lidos. Quando a soma ultrapassar o valor FFH, pare de efetuar leituras do porto PORT1, coloque o número de leituras realizadas na posição de memória 1080H e coloque no porto PORT2, cujo endereço é 02H, o byte 01H.



Etiqueta	Código	Comentário
	LXI B, 0000H	[BC] ← 0000H; [B] ← 00H; [C] ← 00H
LEDA	IN 01H	[A] ← [01H] Read data from port with address 01H (PORT1)
	INR C	[C] ← [C] + 1 Counting update
	ADD B	[A] ← [A] + [B] Add the new data with previous sum
	MOV B, A	[B] ← [A] Save the sum
	JC GUAR	[PC] ← GUAR if carry flag (CS) equal to 1
	JMP LEDA	[PC] ← LEDA Return to the beginning to read new value
GUAR	LXI H, 1080H	[HL] ← 1080H Register pair HL acquires memory position 1080H
	MOV M, C	[[HL]] ← [C] Save counting in memory
	MVI A, 01H	[A] ← 01H
	OUT 02H	[02H] ← [A] Puts the byte in register accumulator (A) into port with address 02H (PORT2)
	HLT	END