 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

1. Qual a diferença fundamental entre um codificador com prioridade e um codificador sem prioridade:
 - a) O codificador sem prioridade não permite mais de uma entrada ativa ao mesmo tempo
 - b) O codificador com prioridade ativa todas as saídas simultaneamente
 - c) O codificador sem prioridade, dá prioridade às entradas de maior peso
 - d) Nenhuma das anteriores


2. Qual o sistema de numeração que utiliza apenas os dígitos 0 e 1:
 - a) Octal
 - b) Hexadecimal
 - c) Binário
 - d) Nenhuma das anteriores

3. Qual é a função principal de um decodificador em circuitos combinatórios:
 - a) Realizar a função inversa de um codificador
 - b) Somar dois números binários
 - c) Detetar e corrigir erros em códigos binários
 - d) Nenhuma das anteriores

4. Qual das seguintes afirmações sobre o código de Hamming é verdadeira:
 - a) O código de Hamming é baseado no código ASCII
 - b) O código de Hamming deteta erros, mas não os corrige
 - c) O código de Hamming identifica e corrige automaticamente o bit incorreto
 - d) Nenhuma das anteriores

5. Qual é a característica principal de um flip-flop edge triggered:
 - a) Reage apenas à transição do sinal de clock
 - b) Reage a qualquer mudança nas entradas
 - c) Não utiliza sinal de clock
 - d) Nenhuma das anteriores

6. Qual é a função dos barramentos num computador:
 - a) Armazenar dados temporariamente
 - b) Interligar o CPU, a memória e as controladoras I/O
 - c) Realizar operações matemáticas e lógicas
 - d) Nenhuma das anteriores

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

7. Qual a função do Instruction Register (IR):
 - a) Armazenar dados temporariamente
 - b) Contém a instrução que está a ser usada pelo processador
 - c) Contém o endereço da próxima instrução a ser executada
 - d) Nenhuma das anteriores

8. O que é um ciclo de instrução:
 - a) O processo de buscar e executar uma instrução armazenada em memória
 - b) O processo de armazenar dados temporariamente
 - c) O processo de controlar o fluxo de dados entre a ALU e a memória
 - d) Nenhuma das anteriores

9. O que é uma interrupção:
 - a) Uma forma de parar um programa em execução para permitir que o sistema operativo faça algo de imediato
 - b) Uma forma de armazenar dados temporariamente
 - c) Uma forma de controlar o fluxo de dados entre a ALU e a memória
 - d) Nenhuma das anteriores


10. O que é um barramento de dados:
 - a) Um conjunto de ligações físicas que interliga o CPU, a memória e as controladoras I/O
 - b) Um conjunto de ligações físicas que transporta dados entre o CPU e a memória
 - c) Um conjunto de ligações físicas que transporta endereços entre o CPU e a memória
 - d) Nenhuma das anteriores

11. Converta o seguinte número $50_{(10)}$ para a base 2.

12. Construa um Código Gray com 3 bits.

13. Projete a seguinte função lógica, $F = X + X \cdot Y \cdot Z$, utilizando um decodificador. Assuma que o valor lógico responsável por ativar as saídas é o valor lógico 1. Utilize a seguinte ordem para as variáveis, X, Y, Z.

14. Implemente a seguinte função lógica, $F = X + X \cdot Y \cdot Z$, utilizando um multiplexer de 2 entradas de controlo. Se necessitar de descartar alguma variável, descarte a variável X. Caso necessite utilize a seguinte ordem para as variáveis, X, Y, Z.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	


15. Complete a seguinte tabela de verdade de um Flip-Flop T Edge Triggered com disparo no flanco descendente.

T	CLOCK	Saída
0	0	?
0	1	?
0	↑	?
0	↓	?
1	0	?
1	1	?
1	↑	?
1	↓	?


16. 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.
17. 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).
18. 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.

Cotação:

- Pergunta de escolha múltipla:
 - Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
- Pergunta de escolha múltipla:
 - Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
- Pergunta de escolha múltipla:
 - Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
- Pergunta de escolha múltipla:
 - Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
- Pergunta de escolha múltipla:
 - Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
- Pergunta de escolha múltipla:

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

- Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
7. Pergunta de escolha múltipla:
- Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
8. Pergunta de escolha múltipla:
- Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
9. Pergunta de escolha múltipla:
- Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
10. Pergunta de escolha múltipla:
- Resposta correta 1 valor, resposta errada -0.25 valor, sem resposta 0 valores
11. 1 valor
12. 1 valor
13. 2 valores
14. 2valores
15. 1 valor
16. 1 valor
17. 1 valor
18. 1 valor

 ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

Instruções do microprocessador da INTEL 8085


Nomenclatura:

LEGENDA

pr	Par de registos: HL, BC, DE, SP, PC
reg	Registo: A, B, C, D, E, H, L
M	Posição de memória
addr	Endereço de 16 bits de uma posição de memória
x	O bit do registo de flags é afetado
byte	Constante, ou expressão lógica/aritmética que representa um dado de 8 bits
double	Constante, ou expressão lógica/aritmética que representa um dado de 16 bits
[]	Conteúdo do que se encontra dentro de parênteses retos
[[]]	Conteúdo do conteúdo do que se encontra dentro de parênteses retos
CS	Flag de carry
label	Endereço de uma posição de memória
port	Endereço de um dispositivo I/O

Grupo de transferência de dados



INSTRUÇÃO	OPERANDOS	STATUS DO REGISTO DE FLAGS					OPERAÇÃO REALIZADA
		CS	AC	Z	S	P	
LDAX	pr						$[A] \leftarrow [[pr]]$ Load A using implied addressing by BC (pr=B) or DE (pr=D)
STAX	pr						$[[pr]] \leftarrow [A]$ Store A using implied addressing by BC (pr=B) or DE (pr=D)
MOV	r,M						$[r] \leftarrow [[HL]]$ Load any register using implied addressing by HL
MOV	M,r						$[[HL]] \leftarrow [r]$ Store any register using implied addressing by HL
LDA	addr						$[A] \leftarrow [addr]$ Load A using direct addressing
STA	addr						$[addr] \leftarrow [A]$ Store A using direct addressing

 <div> ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO </div>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

LHLD	addr	$[L] \leftarrow [addr]$ and $[H] \leftarrow [addr+1]$ Load H and L registers using direct addressing
SHLD	addr	$[addr] \leftarrow [L]$ and $[addr+1] \leftarrow [H]$ Store H and L registers using direct addressing
MOV	r,r	$[r] \leftarrow [r]$ Move any register to any register
XCHG		$[D] \leftrightarrow [H]$ and $[E] \leftrightarrow [L]$ Exchange DE with HL
SPHL		$[HL] \leftarrow [SP]$ Move HL to SP
LXI	pr,double	$[pr] \leftarrow \text{double}$ Load 16 bits immediate data into BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP)
MVI	M,byte	$[[HL]] \leftarrow \text{byte}$ Load 8 bit immediate data into memory location with implied addressing by HL
MVI	r,byte	$[r] \leftarrow \text{byte}$ Load 8 bit immediate data into any register

Grupo aritmético, lógico e de rotação

INSTRUÇÃO	OPERANDOS	STATUS DO REGISTO DE FLAGS					OPERAÇÃO REALIZADA
		CS	AC	Z	S	P	
ADD	M	x	x	x	x	x	$[A] \leftarrow [A] + [[HL]]$ Add register A with implied addressing by HL and store the result in register A
ADC	M	x	x	x	x	x	$[A] \leftarrow [A] + [[HL]] + [CS]$ Add register A with carry with implied addressing by HL and store the result in register A
SUB	M	x	x	x	x	x	$[A] \leftarrow [A] - [[HL]]$ Subtract register A with implied addressing by HL and store the result in register A
SBB	M	x	x	x	x	x	$[A] \leftarrow [A] - [[HL]] - [CS]$ Subtract register A with carry with implied addressing by HL and store the result in register A
ANA	M	0	1	x	x	x	$[A] \leftarrow [A] \text{ AND } [[HL]]$ AND between register A with implied addressing by HL and store the result in register A
XRA	M	0	0	x	x	x	$[A] \leftarrow [A] \text{ XOR } [[HL]]$ Exclusive-OR between register A with implied addressing by HL and store the result in register A
ORA	M	0	0	x	x	x	$[A] \leftarrow [A] \text{ OR } [[HL]]$ OR between register A with implied addressing by HL and store the result in register A
CMP	M	x	x	x	x	x	$[A] - [[HL]]$ Compare register A with implied addressing by HL If register A < [[HL]] then the carry flag is set (1) If register A = [[HL]] then the zero flag is set (1) If register A > [[HL]] then the carry and zero flags are reset (0)
INR	M	x	x	x	x	x	$[[HL]] \leftarrow [[HL]] + 1$ Increment memory
DCR	M	x	x	x	x	x	$[[HL]] \leftarrow [[HL]] - 1$ Decrement memory
ADI	byte	x	x	x	x	x	$[A] \leftarrow [A] + \text{byte}$ Add register A with 8 bit immediate data and store the result in register A

		Tipo de Prova		Ano letivo	Data	
		Exame da época de recurso		2024/2025	02-07-2025	
		Curso			Hora	
		Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores			10:00	
		Unidade Curricular		Duração		
		Sistemas digitais e arquitetura de computadores		02:00 horas		


ACI	byte	x	x	x	x	x	[A] ← [A] + byte + [CS] Add register A with 8 bit immediate data with carry and store the result in register A
SUI	byte	x	x	x	x	x	[A] ← [A] – byte Subtract register A with 8 bit immediate data and store the result in register A
SBI	byte	x	x	x	x	x	[A] ← [A] – byte – [CS] Subtract register A with 8 bit immediate data with carry and store the result in register A
ANI	byte	0	1	x	x	x	[A] ← [A] AND byte AND between register A with 8 bit immediate data and store the result in register A
XRI	byte	0	0	x	x	x	[A] ← [A] XOR byte Exclusive-OR between register A with 8 bit immediate data and store the result in register A
ORI	byte	0	0	x	x	x	[A] ← [A] OR byte OR between register A with 8 bit immediate data and store the result in register A
CPI	byte	x	x	x	x	x	[A] – byte Compare register A with 8 bit immediate data If register A < byte than the carry flag is set (1) If register A = byte than the zero flag is set (1) If register A > byte than the carry and zero flags are reset (0)
ADD	r	x	x	x	x	x	[A] ← [A] + [r] Add register A with any register and store the result in register A
ADC	r	x	x	x	x	x	[A] ← [A] + [r] + [CS] Add register A with any register with carry and store the result in register A
SUB	r	x	x	x	x	x	[A] ← [A] – [r] Subtract register A with any register and store the result in register A
SBB	r	x	x	x	x	x	[A] ← [A] – [r] – [CS] Subtract register A with any register with carry and store the result in register A
ANA	r	0	1	x	x	x	[A] ← [A] AND [r] AND between register A with any register and store the result in register A
XRA	r	0	0	x	x	x	[A] ← [A] XOR [r] Exclusive-OR between register A with any register and store the result in register A
ORA	r	0	0	x	x	x	[A] ← [A] OR [r] OR between register A with any register and store the result in register A
CMP	r	x	x	x	x	x	[A] – [r] Compare register A with any register If register A < r than the carry flag is set (1) If register A = r than the zero flag is set (1) If register A > r than the carry and zero flags are reset (0)
INR	r		x	x	x	x	[r] ← [r] + 1 Increment any register
DCR	r		x	x	x	x	[r] ← [r] – 1 Decrement any register
CMA							[A] ← [A] Complement register A
DAA		x	x	x	x	x	The contents of the accumulator are changed from a binary value to two 4-bit binary coded decimal (BCD) digits. This is the only instruction that uses the auxiliary flag (AC) to perform the binary to BCD conversion, and the conversion procedure is described below. If the value of the low-order 4-bits in the accumulator is greater than 9 or if AC flag is set (1), the instruction adds 6 to the low-order four bits. If the value of the high-order 4-bits in the accumulator is greater than 9 or if the Carry flag (CS) is set (1), the instruction adds 6 to the high-order four bits.
RLC		x					Each binary bit of the register accumulator is rotated left by one position. Bit D7 is placed in the position of D0 as well as in the Carry flag. The carry flag (CS) is modified according to bit D7.

P.PORTO <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	


			
RRC		x	<p>Each binary bit of the register accumulator is rotated right by one position. Bit D₀ is placed in the position of D₇ as well as in the Carry flag. The carry flag (CS) is modified according to bit D₀.</p>
			
RAL		X	<p>Each binary bit of the register accumulator is rotated left by one position through the carry flag. Bit D₇ is placed in the carry flag, and the carry flag is placed in the least significant position D₀. The carry flag (CS) is modified according to bit D₇.</p>
			
RAR		x	<p>Each binary bit of the register accumulator is rotated right by one position through the carry flag. Bit D₀ is placed in the carry flag, and the carry flag is placed in the least significant position D₇. The carry flag (CS) is modified according to bit D₀.</p>
			
DAD	pr	x	<p>[HL] ← [HL] + [pr] Add HL to a register pair BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP) and store the result in HL</p>
INX	pr		<p>[pr] ← [pr] + 1 Increment register pair BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP) and store the result in a register pair BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP)</p>
DCX	pr		<p>[pr] ← [pr] - 1 Decrement register pair BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP) and store the result in a register pair BC (pr=B), DE (pr=D), HL (pr=H), SP (pr=SP)</p>

Grupo de controlo e de salto

INSTRUÇÃO	OPERANDOS	STATUS DO REGISTO DE FLAGS					OPERAÇÃO REALIZADA
		CS	AC	Z	S	P	
JMP	label						[PC] ← label Jump to instruction at address label
PCHL							[PC] ← [HL] Jump to instruction at address contained in HL
CALL	label						[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] - 2 Jump to subroutine starting at address label
CC	label						[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] - 2 Jump to subroutine starting at address label if the carry flag (CS) equal to 1
CNC	label						[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] - 2 Jump to subroutine starting at address label if the carry flag (CS) equal to 0

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

CZ	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the zero flag (Z) equal to 1
CNZ	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the zero flag (Z) equal to 0
CP	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the sign flag (S) equal to 0
CM	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the sign flag (S) equal to 1
CPE	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the parity flag (P) equal to 1
CPO	label	[[SP]] ← [PC], [PC] ← label, [SP] ← [SP] – 2 Jump to subroutine starting at address label if the parity flag (P) equal to 0
RET		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine
RC		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the carry flag (CS) equal to 1
RNC		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the carry flag (CS) equal to 0
RZ		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the zero flag (Z) equal to 1
RNZ		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the zero flag (Z) equal to 0
RM		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the sign flag (S) equal to 0
RP		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the sign flag (S) equal to 1
RPE		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the parity flag (P) equal to 1
RPO		[PC] ← [[SP]], [SP] ← [SP] + 2 Return from subroutine if the parity flag (P) equal to 0
JC	label	[PC] ← label Jump to instruction at address label if the carry flag (CS) equal to 1
JNC	label	[PC] ← label Jump to instruction at address label if the carry flag (CS) equal to 0
JZ	label	[PC] ← label Jump to instruction at address label if the zero flag (Z) equal to 1
JNZ	label	[PC] ← label Jump to instruction at address label if the zero flag (Z) equal to 0
JP	label	[PC] ← label Jump to instruction at address label if the sign flag (S) equal to 0
JM	label	[PC] ← label Jump to instruction at address label if the sign flag (S) equal to 1
JPE	label	[PC] ← label Jump to instruction at address label if the parity flag (P) equal to 1
JPO	label	[PC] ← label Jump to instruction at address label if the parity flag (P) equal to 0

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Exame da época de recurso	Ano letivo 2024/2025	Data 02-07-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 10:00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 02:00 horas	

RST n

The RST instruction is equivalent to a 1-byte call instruction to one of eight memory locations depending upon the number. The instructions are generally used in conjunction with interrupts and inserted using external hardware. However these can be used as software instructions in a program to transfer program execution to one of the eight locations. The addresses are:
Instruction Restart Address

RST	0	0000H
RST	1	0008H
RST	2	0010H
RST	3	0018H
RST	4	0020H
RST	5	0028H
RST	6	0030H
RST	7	0038H

Grupo de controlo do CPU, I/O e da Pilha

INSTRUÇÃO	OPERANDOS	STATUS DO REGISTO DE FLAGS					OPERAÇÃO REALIZADA
		CS	AC	Z	S	P	
IN	port						[A] ← [port] Input to register acumulator (A) from I/O port
OUT	port						[port] ← [A] Output from register acumulator (A) to I/O port
PUSH	pr						[[SP]] ← [pr], [SP] ← [SP] – 2 Push register pair BC (pr=B), DE (pr=D), H (pr=HL), PSW (pr=PSW) contents onto stack
POP	pr						[pr] ← [[SP]], [SP] ← [SP] + 2 Pop stack into register pair BC (pr=B), DE (pr=D), H (pr=HL), PSW (pr=PSW)
XTHL							[HL] ← [[SP]] Exchange HL with top of stack
EI							Enable interrupts following execution of next instruction
DI							Disable interrupts
SIM							Set interrupt mask
RIM							Read interrupt mask
NOP							[PC] ← [PC] +1 No operation but program counter (PC) is incremented
HLT							HALT Stop CPU operation