

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

1. A principal função da ALU é:
  - a) Armazenar dados temporariamente
  - b) Realizar operações aritméticas e lógicas
  - c) Controlar o fluxo de dados entre a ALU e a memória
  - d) Nenhuma das anteriores
  
2. O componente que contém o endereço da próxima instrução a ser executada pelo processador é:
  - a) Instruction Register (IR)
  - b) Program Counter (PC)
  - c) Unidade de Controlo
  - d) Nenhuma das anteriores
  
3. A largura de um barramento é determinada:
  - a) Pela quantidade de memória cache utilizada
  - b) A tecnologia de implementação
  - c) O número de fios no barramento
  - d) Nenhuma das anteriores
  
4. Hierarquia de memória representa:
  - a) A organização dos diferentes níveis de memória num sistema de computação
  - b) A quantidade de memória cache utilizada
  - c) A tecnologia de implementação da memória
  - d) Nenhuma das anteriores
  
5. Num barramento síncrono:
  - a) Todos os eventos ocorrem de forma independente
  - b) Todos os eventos são controlados pelo sinal de relógio
  - c) Os eventos ocorrem apenas quando há uma interrupção
  - d) Nenhuma das anteriores
  
6. A vantagem de um barramento multiplexado é que:
  - a) Requer menos linhas
  - b) Permite operações de armazenamento simultâneas
  - c) Tem um protocolo de comunicação mais simples
  - d) Nenhuma das anteriores


 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

7. A principal função da memória cache é:
  - a) Armazenar dados permanentemente
  - b) Aumentar a velocidade de acesso aos dados frequentemente usados pelo processador
  - c) Armazenar dados temporariamente durante a execução de programas
  - d) Nenhuma das anteriores
  
8. O algoritmo de substituição LRU substitui o bloco:
  - a) Que está na cache há mais tempo
  - b) Que menos vezes foi utilizado
  - c) Que não é usado há mais tempo
  - d) Nenhuma das anteriores
  
9. Numa célula de memória, um erro grave é:
  - a) Um defeito que impede a célula de armazenar dados com segurança
  - b) Um evento aleatório não destrutivo que altera o conteúdo da célula
  - c) Um problema de fornecimento de energia
  - d) Nenhuma das anteriores
  
10. A latência rotacional num disco magnético representa:
  - a) O tempo necessário para mover a cabeça de leitura/escrita para a pista correta
  - b) O tempo gasto enquanto os dados não começam a passar debaixo da cabeça de leitura/escrita
  - c) O tempo necessário para ler ou escrever dados
  - d) Nenhuma das anteriores
  
11. Ache o complemento para um do conteúdo de uma posição de memória (1030H) e guarde o resultado na posição de memória seguinte (1031H).
  
12. Subtraia 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.
  
13. Elabore um programa que leia o conteúdo de uma posição de memória (1050H) e que guarde nas duas posições de memória seguintes (1051H e 1052H) o complemento para um e o complemento para dois do valor lido.

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

Cotação:

1. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
2. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
3. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
4. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
5. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
6. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
7. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
8. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
9. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
10. Pergunta de escolha múltipla:
  - Resposta correta 1.5 valores, resposta errada -0.5 valor, sem resposta 0 valores
11. 1 valor
12. 2 valores
13. 2 valores

 ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 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] ← [[pr]] Load A using implied addressing by BC (pr=B) or DE (pr=D)
STAX	pr						[[pr]] ← [A] Store A using implied addressing by BC (pr=B) or DE (pr=D)
MOV	r,M						[r] ← [[HL]] Load any register using implied addressing by HL
MOV	M,r						[[HL]] ← [r] Store any register using implied addressing by HL
LDA	addr						[A] ← [addr] Load A using direct addressing


<div>P.PORTO</div> <div>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</div>	Tipo de Prova Teste 2		Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores			Hora 14H00
	Unidade Curricular Sistemas digitais e arquitetura de computadores			Duração 00:30 horas

STA	addr	$[addr] \leftarrow [A]$ Store A using direct addressing
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	OPERANDO S	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

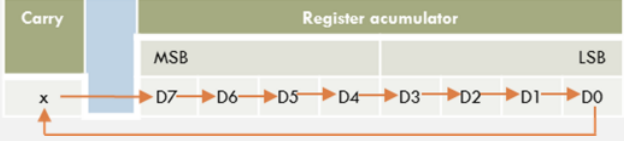
 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Teste 2					Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores					Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores					Duração 00:30 horas	

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]] ← [[HL]] + 1 Increment memory
DCR	M	x	x	x	x	x	[[HL]] ← [[HL]] - 1 Decrement memory
ADI	byte	x	x	x	x	x	[A] ← [A] + byte Add register A with 8 bit immediate data and store the result in register A
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 then the carry flag is set (1) If register A = byte then the zero flag is set (1) If register A > byte then 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

<div>P.PORTO</div> <div>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</div>	Tipo de Prova Teste 2			Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores				Hora 14H00
	Unidade Curricular Sistemas digitais e arquitetura de computadores				Duração 00:30 horas

ORA	r	0	0	x	x	x	$[A] \leftarrow [A] \text{ OR } [r]$ OR between register A with any register and store the result in register A
CMP	r	x	x	x	x	x	$[A] \leftarrow [r]$ 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	$[r] \leftarrow [r] + 1$ Increment any register
DCR	r		x	x	x	x	$[r] \leftarrow [r] - 1$ Decrement any register
CMA							$[A] \leftarrow [\bar{A}]$ 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.
RRC		x					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.
RAL		x					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.
RAR		x					Each binary bit of the register accumulator is rotated right by one position through the carry flag.


<div>P.PORTO</div> <div>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</div>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

			<p>Bit D<sub>0</sub> is placed in the carry flag, and the carry flag is placed in the least significant position D<sub>7</sub>.</p> <p>The carry flag (CS) is modified according to bit D<sub>0</sub>.</p> 
DAD	pr	x	<p><math>[HL] \leftarrow [HL] + [pr]</math></p> <p>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><math>[pr] \leftarrow [pr] + 1</math></p> <p>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><math>[pr] \leftarrow [pr] - 1</math></p> <p>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						<p><math>[PC] \leftarrow \text{label}</math></p> <p>Jump to instruction at address label</p>
PCHL							<p><math>[PC] \leftarrow [HL]</math></p> <p>Jump to instruction at address contained in HL</p>
CALL	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label</p>
CC	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the carry flag (CS) equal to 1</p>
CNC	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the carry flag (CS) equal to 0</p>
CZ	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the zero flag (Z) equal to 1</p>
CNZ	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the zero flag (Z) equal to 0</p>
CP	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the sign flag (S) equal to 0</p>
CM	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the sign flag (S) equal to 1</p>
CPE	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the parity flag (P) equal to 1</p>
CPO	label						<p><math>[[SP]] \leftarrow [PC], [PC] \leftarrow \text{label}, [SP] \leftarrow [SP] - 2</math></p> <p>Jump to subroutine starting at address label if the parity flag (P) equal to 0</p>



 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

RET		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine																		
RC		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the carry flag (CS) equal to 1																		
RNC		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the carry flag (CS) equal to 0																		
RZ		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the zero flag (Z) equal to 1																		
RNZ		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the zero flag (Z) equal to 0																		
RM		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the sign flag (S) equal to 0																		
RP		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the sign flag (S) equal to 1																		
RPE		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the parity flag (P) equal to 1																		
RPO		$[PC] \leftarrow [[SP]], [SP] \leftarrow [SP] + 2$ Return from subroutine if the parity flag (P) equal to 0																		
JC	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the carry flag (CS) equal to 1																		
JNC	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the carry flag (CS) equal to 0																		
JZ	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the zero flag (Z) equal to 1																		
JNZ	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the zero flag (Z) equal to 0																		
JP	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the sign flag (S) equal to 0																		
JM	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the sign flag (S) equal to 1																		
JPE	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the parity flag (P) equal to 1																		
JPO	label	$[PC] \leftarrow \text{label}$ Jump to instruction at address label if the parity flag (P) equal to 0																		
RST	n	<p>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:</p> <p>Instruction Restart Address</p> <table> <tr><td>RST</td><td>0</td><td>0000H</td></tr> <tr><td>RST</td><td>1</td><td>0008H</td></tr> <tr><td>RST</td><td>2</td><td>0010H</td></tr> <tr><td>RST</td><td>3</td><td>0018H</td></tr> <tr><td>RST</td><td>4</td><td>0020H</td></tr> <tr><td>RST</td><td>5</td><td>0028H</td></tr> </table>	RST	0	0000H	RST	1	0008H	RST	2	0010H	RST	3	0018H	RST	4	0020H	RST	5	0028H
RST	0	0000H																		
RST	1	0008H																		
RST	2	0010H																		
RST	3	0018H																		
RST	4	0020H																		
RST	5	0028H																		

 <div> ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO </div>	Tipo de Prova Teste 2	Ano letivo 2024/2025	Data 19-05-2025
	Curso Licenciatura em Engenharia Informática / Licenciatura em Segurança Informática em Redes de Computadores	Hora 14H00	
	Unidade Curricular Sistemas digitais e arquitetura de computadores	Duração 00:30 horas	

RST60030H

RST70038H

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

INSTRUÇÃO	OPERANDOS	STATUS DO REGISTO DE					OPERAÇÃO REALIZADA
		FLAGS					
		CS	AC	Z	S	P	
IN	port						[A] ← [port] Input to register accumulator (A) from I/O port
OUT	port						[port] ← [A] Ouput from register accumulator (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) contentes 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