

DOCUMENTACION EXAMEN

INTRODUCCION A LAS COMPUTADORAS

NOTA IMPORTANTE:

Esta documentación debe ser devuelta al entregar el examen.

No se permite escribir nada sobre la documentación, ni separar las hojas.

Se debe comprobar al recibir la documentación que no tiene nada escrito.

Después de haberla recibido no se admitirán reclamaciones al respecto.

Funcionamiento de los registros

CK	$\overline{\text{LD}}/\text{CNT}$	$\overline{\text{UP}}/\text{DWN}$	Significado
1, 0, ó \downarrow	X	X	No cambia
\uparrow	0	X	Entrada paralelo
\uparrow	1	0	Incrementa registro
\uparrow	1	1	Decrementa registro

Entradas de control de la ALU. Operaciones asociadas a cada combinación

Operación	alu3	alu2	alu1	alu0
ENT2	0	0	0	0
No usada	0	0	0	1
$\text{ENT1} + \text{ENT2} + \text{Fc}$	0	0	1	0
$\text{ENT1} \cap \text{ENT2}$	0	0	1	1
$\text{ENT2} --$	0	1	0	0
$\text{ENT2} ++$	0	1	0	1
$\text{Ca2}(\text{ENT1})$	0	1	1	0
$\text{Ca1}(\text{ENT1})$	0	1	1	1
$\text{ENT1} \cup \text{ENT2}$	1	0	0	0
$\text{ENT1} <_{\text{rc}} <$	1	0	0	1
$\text{ENT1} >_{\text{rc}} >$	1	0	1	0
$\text{ENT1} + \text{Ca2}(\text{ENT2} + \text{Fc})$	1	0	1	1
$\text{ENT1} <_{\text{r}} <$	1	1	0	0
$\text{ENT1} >_{\text{r}} >$	1	1	0	1
$\text{ENT1} >_{\text{sa}} >$	1	1	1	0
$\text{ENT1} \oplus \text{ENT2}$	1	1	1	1

Posición de los flags en el registro F

Posición							
7	6	5	4	3	2	1	0
X	X	I	P	S	O	Z	C

SET DE INSTRUCCIONES DE INDALO 3.0

Instrucción		Descripción	Cod. oper	Flags						Bytes	Cic. Reloj
				C	Z	O	S	P	I		
ADC	A, op8	$A + op8 + Fc \rightarrow A$	1001 0op8	X	X	X	X	X	N	1+A	7+B
ADD	A, op8	$A + op8 \rightarrow A$	1000 1op8	X	X	X	X	X	N	1+A	7+B
AND	A, op8	$A \cap op8 \rightarrow A$	1001 1op8	0	X	0	X	X	N	1+A	7+B
CALL	addr	$SP-2 \rightarrow SP, PC \rightarrow (SP), addr \rightarrow PC$	0000 1011	N	N	N	N	N	N	3	27
CLC		$0 \rightarrow Fc$	0000 1101	0	N	N	N	N	N	1	5
CLI		$0 \rightarrow Fi$	0000 1100	N	N	N	N	N	0	1	5
CMP	A, op8	$A - op8 \rightarrow \text{Act. Flags}$	1000 0op8	X	X	X	X	X	N	1+A	7+B
DEC	reg16	$reg16-- \rightarrow reg16$	0011 00reg16	N	N	N	N	N	N	1	7
DEC	reg8	$reg8-- \rightarrow reg8$	1010 00reg8	N	X	X	X	X	N	1	7
HLT		Paso a Halt	0000 1110	N	N	N	N	N	N	1	3
IN	A, C	(puerto C) $\rightarrow A$	0000 1111	N	N	N	N	N	N	1	8
INC	reg16	$reg16++ \rightarrow reg16$	0011 01reg16	N	N	N	N	N	N	1	7
INC	reg8	$reg8++ \rightarrow reg8$	1010 10reg8	N	X	X	X	X	N	1	7
INT	vector	$SP-2 \rightarrow SP, PC \rightarrow (SP), (V*2) \rightarrow PC$	0001 0000	N	N	N	N	N	N	2	30
JC	addr	Si $Fc=1$ $PC+rel8 \rightarrow PC$	0001 0001	N	N	N	N	N	N	2	7/12
JMP	addr	$addr \rightarrow PC$	0001 0010	N	N	N	N	N	N	3	15
JMP	reg16	$reg16 \rightarrow PC$	0001 11reg16	N	N	N	N	N	N	1	5
JNC	addr	Si $Fc=0$ $PC+rel8 \rightarrow PC$	0001 0011	N	N	N	N	N	N	2	7/12
JNO	addr	Si $Fo=0$ $PC+rel8 \rightarrow PC$	0001 0100	N	N	N	N	N	N	2	7/12
JNP	addr	Si $Fp=0$ $PC+rel8 \rightarrow PC$	0001 0101	N	N	N	N	N	N	2	7/12
JNS	addr	Si $Fs=0$ $PC+rel8 \rightarrow PC$	0001 0110	N	N	N	N	N	N	2	7/12
JNZ	addr	Si $Fz=0$ $PC+rel8 \rightarrow PC$	0001 0111	N	N	N	N	N	N	2	7/12
JO	addr	Si $Fo=1$ $PC+rel8 \rightarrow PC$	0001 1000	N	N	N	N	N	N	2	7/12
JP	addr	Si $Fp=1$ $PC+rel8 \rightarrow PC$	0001 1001	N	N	N	N	N	N	2	7/12
JS	addr	Si $Fs=1$ $PC+rel8 \rightarrow PC$	0001 1010	N	N	N	N	N	N	2	7/12
JZ	addr	Si $Fz=1$ $PC+rel8 \rightarrow PC$	0001 1011	N	N	N	N	N	N	2	7/12
MOV	reg8, op8	$op8 \rightarrow reg8$	011reg8 op8	N	N	N	N	N	N	1+A	7+B
MOV	mem8, reg8	$Reg8 \rightarrow mem8$	0100 mem8reg8	N	N	N	N	N	N	1+A	7+B
MOV	reg16, op16	$op16 \rightarrow reg16$	0101 reg16op16	N	N	N	N	N	N	1+A	7+B
NEG	A	$\bar{A} \rightarrow A$	1011 0000	X	X	X	X	X	N	1	7
NOP		No operar	0000 0000	N	N	N	N	N	N	1	5
NOT	A	$\bar{A} \rightarrow A$	1011 1000	0	X	0	X	X	N	1	7
OR	A, op8	$A \cup op8 \rightarrow A$	1100 0op8	0	X	0	X	X	N	1+A	7+B
OUT	C, A	$A \rightarrow (\text{puerto C})$	0010 0000	N	N	N	N	N	N	1	8
POP	AF	$(SP) \rightarrow AF, SP+2 \rightarrow SP$	0000 0001	X	X	X	X	X	X	1	15
POP	BC	$(SP) \rightarrow BC, SP+2 \rightarrow SP$	0000 0010	N	N	N	N	N	N	1	15
POP	X	$(SP) \rightarrow X, SP+2 \rightarrow SP$	0000 0011	N	N	N	N	N	N	1	15
PUSH	AF	$SP-2 \rightarrow SP, AF \rightarrow (SP)$	0000 0101	N	N	N	N	N	N	1	15
PUSH	BC	$SP-2 \rightarrow SP, BC \rightarrow (SP)$	0000 0110	N	N	N	N	N	N	1	15
PUSH	X	$SP-2 \rightarrow SP, X \rightarrow (SP)$	0000 0111	N	N	N	N	N	N	1	15
RCL	A	$A \ll rc \rightarrow A$	1100 1000	X	X	X	X	X	N	1	7
RCR	A	$A \gg rc \rightarrow A$	1101 0000	X	X	X	X	X	N	1	7
RET		$(SP) \rightarrow PC, SP+2 \rightarrow SP$	0000 1000	N	N	N	N	N	N	1	13
ROL	A	$A \ll 1 \rightarrow A$	1010 1000	X	X	X	X	X	N	1	7
ROR	A	$A \gg 1 \rightarrow A$	1010 0000	X	X	X	X	X	N	1	7
SAR	A	$A \gg sa \rightarrow A$	0100 0000	X	X	0	X	X	N	1	7
SBB	A, op8	$A - (op8 + Fc) \rightarrow A$	1101 1op8	X	X	X	X	X	N	1+A	7+B
SHL	A	$A \ll s \rightarrow A$	1110 0000	X	X	X	X	X	N	1	7
SHR	A	$A \gg s \rightarrow A$	1110 1000	X	X	X	0	X	N	1	7
STI		$1 \rightarrow Fi$	0000 1010	N	N	N	N	N	1	1	5
STC		$1 \rightarrow Fc$	0000 1001	1	N	N	N	N	N	1	5
SUB	A, op8	$A - op8 \rightarrow A$	1111 0op8	X	X	X	X	X	N	1+A	7+B
XOR	A, op8	$A \oplus op8 \rightarrow A$	1111 1op8	0	X	0	X	X	N	1+A	7+B

Para los valores de A y B consultar las tablas op8, op16 y mem8.

En saltos condicionales 7/12 quiere decir que la duración es de 7 ciclos de reloj si no se salta y de 12 si se salta

Set de instrucciones por Cod. oper

Cod. Oper	Instrucción		Bytes
0000 0000	NOP		1
0000 0001	POP	AF	1
0000 0010	POP	BC	1
0000 0011	POP	X	1
0000 0101	PUSH	AF	1
0000 0110	PUSH	BC	1
0000 0111	PUSH	X	1
0000 1000	RET		1
0000 1001	STC		1
0000 1010	STI		1
0000 1011	CALL	addr	3
0000 1100	CLI		1
0000 1101	CLC		1
0000 1110	HLT		1
0000 1111	IN	A, C	1
0001 0000	INT	vector	2
0001 0001	JC	addr	2
0001 0010	JMP	addr	3
0001 0011	JNC	addr	2
0001 0100	JNO	addr	2
0001 0101	JNP	addr	2
0001 0110	JNS	addr	2
0001 0111	JNZ	addr	2
0001 1000	JO	addr	2
0001 1001	JP	addr	2
0001 1010	JS	addr	2
0001 1011	JZ	addr	2
0001 11reg16	JMP	reg16	1
0010 0000	OUT	C, A	1
0011 00reg16	DEC	reg16	1
0011 01reg16	INC	reg16	1
0100 0000	SAR	A	1
0100 mem8reg8	MOV	mem8,reg8	1+A
0101 reg16op16	MOV	reg16,op16	1+A
011reg8 op8	MOV	reg8,op8	1+A
1000 0op8	CMP	A,op8	1+A
1000 1op8	ADD	A,op8	1+A
1001 0op8	ADC	A,op8	1+A
1001 1op8	AND	A,op8	1+A
1010 00reg8	DEC	reg8	1
1010 1000	ROL	A	1
1010 10reg8	INC	reg8	1
1011 0000	NEG	A	1
1011 1000	NOT	A	1
1100 0op8	OR	A,op8	1+A
1100 1000	RCL	A	1
1101 0000	RCR	A	1
1101 1op8	SBB	A,op8	1+A
1110 0000	SHL	A	1
1110 1000	SHR	A	1
1111 0op8	SUB	A,op8	1+A
1111 1op8	XOR	A,op8	1+A

op8 → Operando de 8 bits.

Los códigos de operación para las diferentes formas de direccionamiento de op8 son:

Cód.	Mnem.	Dirección del operando	A	B
000	(addr)	La dirección del operando aparece en los 2 bytes siguientes al Cod. oper.	2	11
001	A	Operando en el registro A	0	0
010	B	Operando en el registro B	0	0
011	C	Operando en el registro C	0	0
100	dat8	Valor inmediato viene en el byte siguiente al Cod. Oper	1	3
101	(BC)	Operando en la dirección apuntada por BC	0	1
110	(X+rel8)	Operando en la dirección apuntada por X+ rel8.	1	6

Reg8	
01	A
10	B
11	C

mem8		A	B
00	(addr)	2	11
01	(BC)	0	1
10	(X+rel8)	1	6

op16 → Operando de 16 bits.

Los códigos de operación para las diferentes formas de direccionamiento de op16 son:

Cód.	Mnem.	Dirección del operando	A	B
00	BC	Operando en el registro BC	0	2
01	X	Operando en el registro X	0	2
10	dat16	Valor inmediato viene en los 2 bytes siguientes al Cod.oper	2	8
11	SP	Operando en el registro SP	0	2

Reg16	
00	BC
01	X
10	SP

addr: Direc. (Entero sin signo de 16 bits)

rel8: Entero con signo de 8 bits.

Flags:

0: Flag se pone a 0

1: Flag se pone a 1

X= Flag se actualiza con la ejecución de la inst.

N=Flag no cambia con la ejecución de la inst.

C: Es el acarreo en las sumas, el borrow en las restas, y en desplaz. y rotaciones, es el bit saliente.

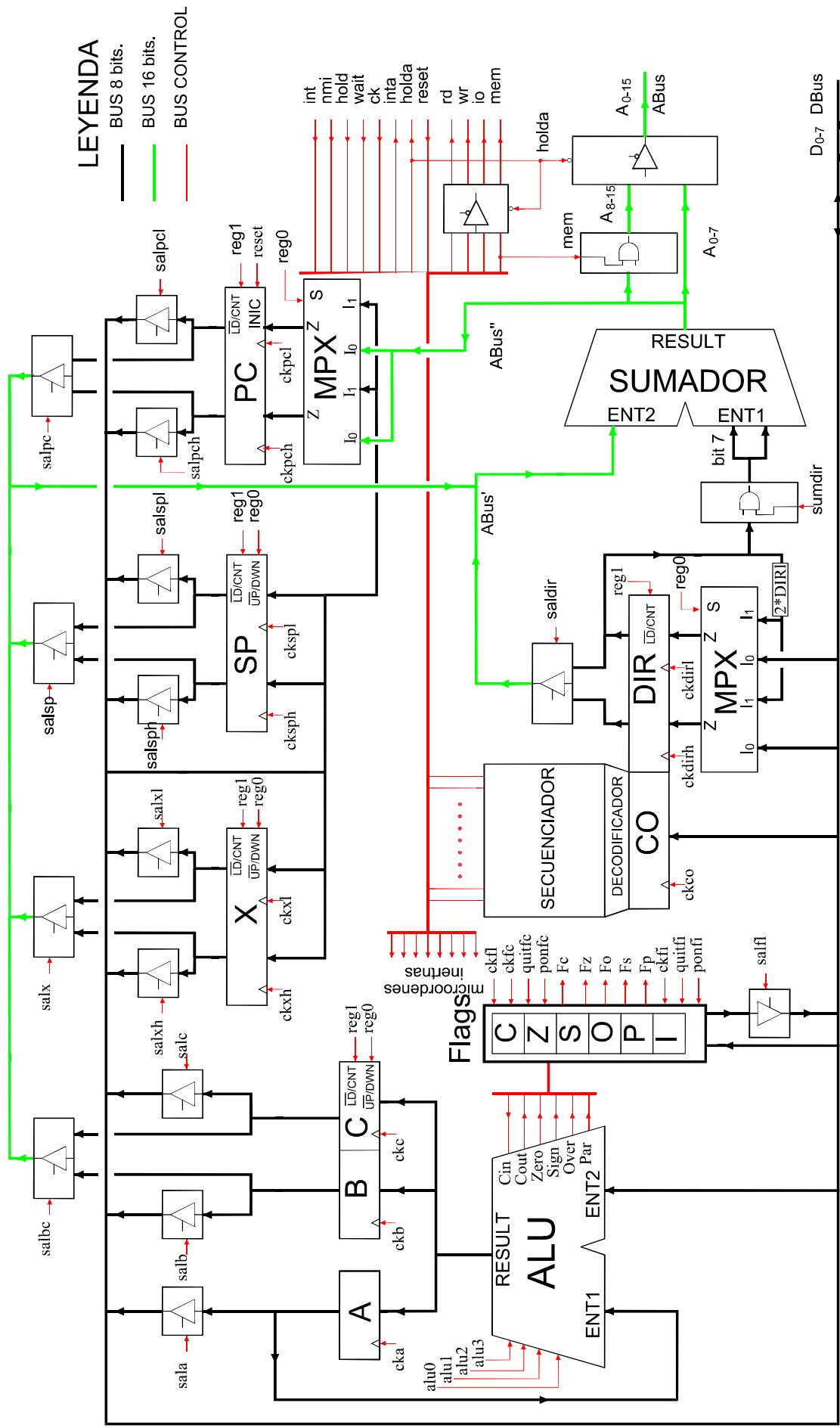
Z: Vale 1 si el contenido del resultado es =0.

O: Realiza una operación XOR entre los acarreos de los 2 bits mas significativos de los operandos. En Shifts y Rotaciones se pone a 1 si el bit de mayor peso cambia como consecuencia de la operación.

S: Es una copia del bit de mayor peso del resultado.

P: Es 0 si el número de unos en el resultado es impar.

INDALO 3.0



Operaciones básicas de Indalo 3.0

Operación	Nro. ciclos reloj	Transferencias	Instrucciones y notas
F	5	(PC)→CO; PC++	Casi todas. (1)
		(PC)→DIR _l ; PC++	(2)
		(PC)→DIR _h ; PC++	(3)
		(PC)→X _l , X _h , SP _l o SP _h ; PC++	MOV X, <i>dat16</i> ; MOV SP, <i>dat16</i>
F _{alu}	5	A-(PC)→Act. Flags; PC++	CMP A, <i>dat8</i>
		A <i>oper2</i> (PC)→A; PC++	XXX A, <i>dat8</i>
		(PC)→ <i>reg8</i> ; PC++	MOV <i>reg8</i> , <i>dat8</i>
F _{flag}	5	(PC)→CO; PC++; 0→F _C o F _l	CLC; CLI; ADD A, <i>op8</i> ; SUB A, <i>op8</i> ; CMP A, <i>op8</i> ; SHR A; SHL A
		(PC)→CO; PC++; 1→F _C o F _l	STC; STI
F _{noinc}	3	(PC)→CO	JMP <i>reg16</i> ; RET
		(PC)→DIR _h	JMP <i>addr</i>
R	3	(SP)→F, X _l , X _h , PC _l o PC _h	POP AF; POP X; RET
		(DIR)→PC _l ó PC _h	INT <i>vector</i> (6)
R _{alu}	3	A- <i>mem8</i> →Act. Flags	CMP A, <i>mem8</i>
		A <i>oper2 mem8</i> →A	XXX A, <i>mem8</i> (4)
		<i>mem8</i> → <i>reg8</i>	MOV <i>reg8</i> , <i>mem8</i>
		(SP)→ <i>reg8</i>	POP AF; POP BC
W	3	<i>reg8</i> → <i>mem8</i>	MOV <i>mem8</i> , <i>reg8</i>
		<i>reg8</i> →(SP)	PUSH AF; PUSH BC
		F, X _l , X _h , PC _l o PC _h →(SP)	PUSH AF; PUSH X; CALL <i>addr</i> ; INT <i>vector</i> (6)
IP	3	(puerto C)→A	IN A,C
OP	3	A→(puerto C)	OUT C,A
RI	2	<i>Vector</i> →DIR _l	Interrupciones hardware
I ₈	2	<i>reg16l</i> →X _l o SP _l ;	MOV X, <i>reg16</i>
		<i>reg16h</i> →X _h o SP _h	MOV SP, <i>reg16</i>
I _{alu}	2	A- <i>reg8</i> →Act. Flags	CMP A, <i>reg8</i>
		A <i>oper2 reg8</i> →A	XXX A, <i>reg8</i> (4)
		<i>reg8</i> → <i>reg8</i>	MOV <i>reg8</i> , <i>reg8</i>
		A <i>oper1</i> →A	YYY A (5)
		<i>reg8++</i> ó <i>reg8--</i>	INC <i>reg8</i> ; DEC <i>reg8</i>
		<i>reg16l</i> →C o <i>reg16h</i> →B	MOV BC, <i>reg16</i>
I _{t16}	2	PC+DIR _l →PC	Jcond <i>addr</i> (si se cumple la condición)
		DIR→PC	JMP <i>addr</i> ; CALL <i>addr</i>
		<i>reg16</i> →PC	JMP <i>reg16</i>
I _{id16}	2	<i>reg16++</i> o <i>reg16--</i>	INC <i>reg16</i> ; DEC <i>reg16</i>
		PC++	Jcond <i>addr</i> (si no se cumple la condición)
		SP--	PUSH AF; PUSH BC; PUSH X; CALL <i>addr</i> ;INT <i>vector</i> (6)
		SP++	POP AF; POP BC; POP X; RET
		DIR++	INT <i>vector</i> (6)
I _{2dir1}	2	2*DIR _l →DIR	INT <i>vector</i> (6)

NOTAS:

(1) Se exceptúan:

- CLC, CLI, ADD, SUB, CMP, SHR y SHL que leen el código de operación mediante F_{flag} .
- JMP *reg16* y RET que lo hacen por medio de F_{noinc} .

(2) Producen esta transferencia:

- Las instrucciones que contienen el operando *rel8*, excepto *Jcond addr* cuando no se cumple la condición.
- Las instrucciones que contienen el operando *addr*.
- INT *vector*.

(3) Producen esta transferencia las instrucciones que contienen el operando *addr*, excepto JMP *addr* que emplea F_{noinc} . Nótese que el operando de *Jcond addr* no es *addr* sino *rel8*.

(4) *Oper2* es cualquiera de las operaciones binarias que puede realizar la CPU: suma con carry, resta con borrow, AND, OR u OR exclusivo; XXX es cualquiera de los mnemónicos de las instrucciones que realizan operaciones binarias: ADD, ADC, SUB, SBB, AND, OR o XOR.

(5) *Oper1* es cualquiera de las operaciones unarias que puede realizar la CPU: rotaciones (circulares y a través de carry; a derecha y a izquierda); shift aritmético a la derecha, y complementos (a 1 y a 2). YYY es cualquiera de los mnemónicos de las instrucciones que realizan las operaciones anteriores: ROR, ROL, RCR, RCL, SHL, SAR, SAL, NOT y NEG.

(6) También se producen estas operaciones en el proceso de atención a las interrupciones hardware.