



**Arellano Granados Angel Mariano**  
**218123444**

**Traductores de Lenguajes I**  
**I7025 D05**

**Actividad 2**

# Uso de las instrucciones lógicas

## NOT

```
01 ;Programa Instruccion NOT
02 .model small
03
04 .code
05 xor ax,ax           ;limpiar registro AX
06 mov ax,3Ah          ;AX = 0011 1010
07 not ax              ;AX = 1100 0101 = FFC5
08
09 ret
```

registers

	H	L
AX	00	3A
BX	00	00
CX	00	08
DX	00	00
CS	0710	
IP	0005	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:0005

Address	Hex	Dec
07100:	33	051
07101:	C0	192
07102:	B8	184
07103:	3A	058
07104:	00	000
07105:	F7	247
07106:	D0	208
07107:	C3	195
07108:	90	144
07109:	90	144
0710A:	90	144
0710B:	90	144
0710C:	90	144
0710D:	90	144
0710E:	90	144
0710F:	90	144

extended value viewer

watch: AX

word

H	L
hex:	00 3A
bin:	00000000 00111010
oct:	000 072

decimal 8 bit

unsigned:	0 58
signed:	0 58
ascii:	:

decimal 16 bit

unsigned:	58
signed:	58

registers

	H	L
AX	FF	C5
BX	00	00
CX	00	08
DX	00	00
CS	0710	
IP	0007	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:0007

Address	Hex	Dec
07100:	33	051
07101:	C0	192
07102:	B8	184
07103:	3A	058
07104:	00	000
07105:	F7	247
07106:	D0	208
07107:	C3	195
07108:	90	144
07109:	90	144
0710A:	90	144
0710B:	90	144
0710C:	90	144
0710D:	90	144
0710E:	90	144
0710F:	90	144

extended value viewer

watch: AX

word

H	L
hex:	FF C5
bin:	11111111 11000101
oct:	377 305

decimal 8 bit

unsigned:	255 197
signed:	-1 -59
ascii:	†

decimal 16 bit

unsigned:	65477
signed:	-59

## NEG

```
01 ;Programa Intruccion NEG
02 .model small
03
04 .code
05 xor ax,ax           ;limpiar registro AX
06 mov ax,99h         ;AX = 1001 1001
07 neg ax              ;AX = 0110 0111 = FF67
08
09 ret
```

emulator: noname.exe\_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	99
BX	00	00
CX	00	08
DX	00	00
CS	0710	
IP	0005	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:0005

07100:	33	051	3
07101:	C0	192	L
07102:	B8	184	0
07103:	99	153	Ö
07104:	00	000	NI
07105:	F7	247	-
07106:	D8	216	i
07107:	C3	195	T
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

extended value viewer

watch: AX

word byte

	H	L
hex:	00	99
bin:	00000000	10011001
oct:	000	231
decimal 8 bit		
unsigned:	0	153
signed:	0	-103
ascii:		0
decimal 16 bit		
unsigned:	153	
signed:	153	

screen source reset aux

registers

	H	L
AX	FF	67
BX	00	00
CX	00	08
DX	00	00
CS	0710	
IP	0007	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:0007

07100:	33	051	3
07101:	C0	192	L
07102:	B8	184	0
07103:	99	153	Ö
07104:	00	000	NI
07105:	F7	247	-
07106:	D8	216	i
07107:	C3	195	T
07108:	90	144	E
07109:	90	144	E
0710A:	90	144	E
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

extended value viewer

watch: AX

word byte

	H	L
hex:	FF	67
bin:	11111111	01100111
oct:	377	147
decimal 8 bit		
unsigned:	255	103
signed:	-1	103
ascii:		9
decimal 16 bit		
unsigned:	65383	
signed:	-153	

screen source reset aux

## AND

```

01 ;Programa Intruccion AND
02 .model small
03 .code
04 xor ax,ax           ;limpiar registro AX
05 xor bx,bx           ;limpiar registro BX
06 mov ax,6Eh          ;AX = 0110 1110
07 mov bx,73h          ;AX = 0111 0011
08 and ax,bx           ;AX = 0110 0010 = 62h
09 ret

```

registers

	H	L
AX	00	6E
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000A	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000A

07100:	33	051	3
07101:	C0	192	L
07102:	33	051	3
07103:	DB	219	0
07104:	B8	184	0
07105:	6E	110	n
07106:	00	000	N
07107:	BB	187	1
07108:	73	115	S
07109:	00	000	N
0710A:	23	035	#
0710B:	C3	195	1
0710C:	C3	195	1
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

0710:000A

```

XOR AX, AX
XOR BX, BX
MOV AX, 0006Eh
MOV BX, 00073h
AND AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

screen source reset aux vars debug stack flags

registers

	H	L
AX	00	62
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000C	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000C

07100:	33	051	3
07101:	C0	192	L
07102:	33	051	3
07103:	DB	219	0
07104:	B8	184	0
07105:	6E	110	0
07106:	00	000	N
07107:	BB	187	1
07108:	73	115	S
07109:	00	000	N
0710A:	23	035	#
0710B:	C3	195	1
0710C:	C3	195	1
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

extended value viewer

watch: AX

word byte

	H	L
hex:	00	62
bin:	00000000	01100010
oct:	000	142

decimal 8 bit

unsigned:	0	98
signed:	0	98
ascii:		b

decimal 16 bit

unsigned:	98
signed:	98

screen source reset aux

## OR

```

01 ;Programa Intruccion OR
02 .model small
03 .code
04 xor ax,ax           ;limpiar registro AX
05 xor bx,bx           ;limpiar registro BX
06 mov ax,6Eh          ;AX = 0110 1110
07 mov bx,73h          ;AX = 0111 0011
08 or ax,bx            ;AX = 0111 1111 = 7Fh
09 ret

```

registers

	H	L
AX	00	6E
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000A	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000A

Address	Hex	Disasm
07100:	33	051 3
07101:	C0	192 3
07102:	33	051 3
07103:	DB	219 3
07104:	B8	184 3
07105:	6E	110 3
07106:	00	000 3
07107:	BB	187 3
07108:	73	115 3
07109:	00	000 3
0710A:	0B	011 3
0710B:	C3	195 3
0710C:	C3	195 3
0710D:	90	144 3
0710E:	90	144 3
0710F:	90	144 3

0710:000A

```

XOR AX, AX
XOR BX, BX
MOV AX, 0006Eh
MOV BX, 00073h
OR AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

screen source reset aux vars debug stack flags

registers

	H	L
AX	00	7F
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000C	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000C

Address	Hex	Disasm
07100:	33	051 3
07101:	C0	192 3
07102:	33	051 3
07103:	DB	219 3
07104:	B8	184 3
07105:	6E	110 3
07106:	00	000 3
07107:	BB	187 3
07108:	73	115 3
07109:	00	000 3
0710A:	0B	011 3
0710B:	C3	195 3
0710C:	C3	195 3
0710D:	90	144 3
0710E:	90	144 3
0710F:	90	144 3

extended value viewer

watch: AX

word

	H	L
hex:	00	7F
bin:	00000000	01111111
oct:	000	177

decimal 8 bit

	H	L
unsigned:	0	127
signed:	0	127
ascii:		â

decimal 16 bit

	H	L
unsigned:		127
signed:		127

screen source reset aux

# XOR

```

01 ;Programa Instruccion XOR
02 .model small
03 .code
04 xor ax,ax           ;limpiar registro AX
05 xor bx,bx           ;limpiar registro BX
06 mov ax,6Eh          ;AX = 0110 1110
07 mov bx,73h          ;AX = 0111 0011
08 xor ax,bx           ;AX = 0001 1101 = 1Dh
09 ret

```

registers

	H	L
AX	00	6E
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000A	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000A

Address	Hex	Dec	Symbol
07100:	33	051	3
07101:	C0	192	L
07102:	33	051	3
07103:	DB	219	0
07104:	B8	184	0
07105:	6E	110	n
07106:	00	000	N
07107:	BB	187	1
07108:	73	115	S
07109:	00	000	N
0710A:	33	051	3
0710B:	C3	195	F
0710C:	C3	195	F
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

0710:000A

```

XOR AX, AX
XOR BX, BX
MOV AX, 0006Eh
MOV BX, 00073h
XOR AX, BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...

```

screen source reset aux vars debug stack flags

registers

	H	L
AX	00	1D
BX	00	73
CX	00	00
DX	00	00
CS	0710	
IP	000C	
SS	0710	
SP	0000	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

0710:000C

Address	Hex	Dec	Symbol
07100:	33	051	3
07101:	C0	192	L
07102:	33	051	3
07103:	DB	219	0
07104:	B8	184	0
07105:	6E	110	n
07106:	00	000	N
07107:	BB	187	1
07108:	73	115	S
07109:	00	000	N
0710A:	33	051	3
0710B:	C3	195	F
0710C:	C3	195	F
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E

extended value viewer

watch: AX

word byte

	H	L
hex:	00	1D
bin:	00000000	00011101
oct:	000	035

decimal 8 bit

	H	L
unsigned:	0	29
signed:	0	29
ascii:		*

decimal 16 bit

	H	L
unsigned:	29	
signed:	29	

screen source reset aux

## **Conclusiones**

Esta no fue mi primera vez programando en ensamblador, pero si en la arquitectura del 8086 aun así me pareció una excelente introducción al tema y aunque solo sean operaciones lógicas simples no dejo de pensar en las infinitas maneras en las que las podría implementar en futuros programas.

## **Biografía**

- Microprocesadores de Intel, séptima edición de Barry B. Brey.