

a) Operador lógico NOT (inversor):

```
LDI R16,!255      ;R16=0. CUALQUIER NÚMERO DIFERENTE DE
                  ;CERO DA CERO
LDI R16,!0         ;R16=1. SIEMPRE QUE SEA !0 EL
                  ;RESULTADO SERÁ 1
```

b) Operador negado:

```
LDI R17,~10        ;R17=245. EL RESULTADO ES 255 MENOS
                  ;10 (complemento)

LDI R17,~0b0101_0101 ;R17=1010_1010. EL RESULTADO ES EL
                  ;COMPLEMENTO

LDI R17,~-2        ;R17=1. EL RESULTADO ES EL COMPLEMENTO
                  ;DE LA DIFERENCIA DE 256-2
                  ;ES DECIR 255-(256-2) =1
                  ;El COMPLEMENTO (~) DE 254
                  ;(complemento de 0b1111_1110=254 en
                  ;BINARIO)es 0b0000_0001
                  ;Dicho de otra forma, el resultado es
                  ;el COMPLEMENTO del COMPLEMENTO-A2 de
                  ;(-2)

NOP                ;ESTE NOP ES NECESARIO PORQUE LA
                  ;LÍNEA ANTERIOR INCREMENTA PC+1
                  ;OJO!!!
```

El valor que asume el AVR de un número *negativo* es el **complemento-A2** de ese número, mientras que el operador negado (~) es solamente el complemento.

En este libro se ha escrito la segunda forma de sacar el **complemento -A2** de un número, esto es (para el número **-10** decimal por ejemplo):

Para 8-bits, el resultado es: $(2^8) - 10 = 256 - 10 = 246$

Para 16-bits, el resultado es: $(2^{16}) - 10 = 65536 - 10 = 65526$

c) Resta de un número negativo con uno positivo:

```
LDI R17,-50        ;R17=256-50=206 (COMPLEMENTO-A2)
LDI R18,100

SUB R17,R18        ;R17=206-100=106
                  ;Se activó bandera de bit de SIGNO
                  ;y bandera de COMPLEMENTO-A2
```

d) Resta de un número mayor con uno menor:

```
LDI R17,-50        ;R17=256-50=206 (COMPLEMENTO-A2)
```

```
LDI R18,100
SUB R18,R17      ;R18=100-(-50)=100+50=150
                  ;R18=100-(256-50)=100-(206)=100+
                  ; (256-206)=150
                  ;SE ACTIVÓ CARRY FLAG
                  ;NEGATIVE FLAG
                  ;COMPLEMENT OVERFLOW FLAG

                  ;y también HALF-CARRY (MEDIO-ACARREO)
```

e) Multiplicación:

```
LDI R18,127*2    ;R18=254
```

```
;LDI R18,128*2   ;ERROR OUT OF RANGE
```

Este es un error que sucede cuando el resultado del operador es mayor de 8-bits

f) División:

```
LDI R18,127/2    ;R18=63...REAL 63.5
LDI R18,127/3    ;R18=42...REAL 42.33
LDI R18,127/128  ;R18=0....REAL 0.99
```

g) Adición:

```
LDI R16,5+6      ;R16=11...EL RESULTADO ES LA SUMA
```

h) Substracción:

```
LDI R16,5-6      ;R16=255...EL RESULTADO ES
                  ;5-6=5+(256-6)=-1+256=255
;o dicho de otro modo, el resultado es 5+(COMPLEMENTO-A2 DE
;-6)
LDI R16,6-5      ;R16=1
```

i) Recorrer a la izquierda (Shift Left):

```
LDI R16,0b0000_0001<<1    ;R16=0b0000_0010...EL 1 se
                           ;recorrió un lugar a la
                           ;izquierda

LDI R16,0b0000_0001<<3    ;R16=0b0000_1000...EL 1 se
                           ;recorrió 3 lugares a la
                           ;izquierda
```

j) Recorrer a la derecha (Shift Right):

```
LDI R16,0b1000_0000>>1    ;R16=0b0100_0000...EL 1 se
                           ;recorrió un lugar a la
                           ;derecha
```

```
LDI R16,0b1000_0000>>3      ;R16=0b0001_0000...EL 1 se
                                ;recorrió 3 lugares a la
                                ;derecha
```

k) Menor que:

```
LDI R16,3<2      ;R16=0...Si el valor de la izquierda es
                  ;mayor que el de la derecha
LDI R16,2<3      ;R16=1...Si el valor de la izquierda es
                  ;menor que el de la derecha
LDI R16,-2<3     ;R16=1...Si el valor de la izquierda es
                  ;menor que el de la derecha
LDI R16,(2<3)+1  ;R16=2...Porque en el paréntesis el valor
                  ;de la izquierda es menor que el de la
                  ;derecha, al resultado se le suma 1
```

l) Menor o igual:

```
LDI R16,3<=2     ;R16=0...Si el valor de la izquierda es
                  ;mayor que el de la derecha
LDI R16,2<=2     ;R16=1...Si el valor de la izquierda es
                  ;menor o igual que el de la derecha
LDI R16,2<=10    ;R16=1...Si el valor de la izquierda es
                  ;menor o igual que el de la derecha
LDI R16,-3<=2    ;R16=1...Si el valor de la izquierda es
                  ;menor o igual que el de la derecha
```

m) Mayor que:

```
LDI R16,3>2      ;R16=1...Si el valor de la izquierda es
                  ;mayor que el de la derecha
LDI R16,3>10     ;R16=0...Si el valor de la izquierda es
                  ;menor que el de la derecha
LDI R16,-5>10    ;R16=0...Si el valor de la izquierda es
                  ;menor que el de la derecha
LDI R16,(-5>10)+8 ;R16=8...Porque en el paréntesis el valor
                  ;de la izquierda es menor que el de la
                  ;derecha, y el resultado es más 8
```

n) Mayor o igual:

```
LDI R16,3>=2     ;R16=1...Si el valor de la izquierda es
                  ;mayor o igual que el de la derecha
LDI R16,10>=10   ;R16=1...Si el valor de la izquierda es
                  ;mayor o igual que el de la derecha
LDI R16,2>=5     ;R16=0...Si el valor de la izquierda es
                  ;menor que el de la derecha
LDI R16,3>=-2    ;R16=1...Si el valor de la izquierda es
                  ;mayor o igual que el de la derecha
```

o) Igual:

```
LDI R16,3==3      ;R16=1...Si el valor de la izquierda es
                  ;igual que el de la derecha
LDI R16,10==3     ;R16=0...Si el valor de la izquierda no
                  ;es igual que el de la derecha
LDI R16,(3==5)+2  ;R16=2...Porque en el paréntesis el valor
                  ;de la izquierda no es igual que el de la
                  ;derecha, y el resultado es más 2
```

p) No-igual:

```
LDI R16,8!=8      ;R16=0...Si el valor de la izquierda no es
                  ;igual que el de la
;derecha
LDI R16,5!=10     ;R16=1...porque el valor de la izquierda no
                  ;es igual al de la derecha
LDI R16,(7!=4)+5  ;R16=6...Porque en el paréntesis el valor
                  ;de la izquierda no es igual que el de la
                  ;derecha, y el resultado es más 5
```

q) Bitwise AND:

```
LDI R16,8&8      ;R16=8...UNO Y OTRO DEBEN SER IGUALES para
                  ;que el resultado sea uno

LDI R16,0b0000_1111&0b1111_0000  ;R16=0b0000_0000...Un BIT
                                  ;y otro deben ser IGUALES
;      0b0000_1111
;      & 0b1111_0000
;      -----
;      0b0000_0000

LDI R16,0b1010_1010&0b1010_1010  ;R16=0b1010_1010...Un BIT
                                  ;y otro deben ser IGUALES
;      0b1010_1010
;      & 0b1010_1010
;      -----
;      0b1010_1010
```

r) Lógica AND:

```
;SI UNA DE LAS EXPRESIONES ES CERO ENTONCES EL RESULTADO ES CERO
LDI R16,0b0000_1010&&0b0000_1010  ;R16=0b0000_0001
LDI R16,0b0000_0000&&0b0000_0000  ;R16=0b0000_0000
LDI R16,0b0000_1010&&0b0000_0000  ;R16=0b0000_0000
```

s) Bitwise OR:

```
LDI R16, 8 | 4           ;R16=12
LDI R16, 200 | 200       ;R16=200
```

En el ejemplo anterior no se puede visualizar apropiadamente por tratarse de números decimales, por ello se sugiere que sean manipulados en binario:

```
LDI R16, 0b1000_0000 | 0b0000_0100

;      0b0000_1000      =8
; | 0b0000_0100      =4
; -----
;      0b0000_1100      R16=12      ;El resultado es la suma
;                               ;de la ponderación
```

```
LDI R16, 0b1100_1000 | 0b1100_1000

;      0b1100_1000      =200
; | 0b1100_1000      =200
; -----
;      0b1100_1000      R16=200      ;El resultado es la suma
;                               ;de la ponderación
```

```
LDI R16, 0b0000_1010 | 0b0000_1011      ;R16=0b0000_1011...Un BIT
;u otro deben ser uno
;para que el resultado
;sea UNO
```

```
;      0b0000_1010
; | 0b0000_1011
; -----
;      0b0000_1011      ;El resultado es la suma
;                               ;de la ponderación
```

t) Lógica OR

;SI UNA DE LAS EXPRESIONES NO ES CERO ENTONCES EL RESULTADO ES UNO

```
LDI R16, 0b0000_1010 || 0b0000_1010      ;R16=0b0000_0001
LDI R16, 0b0000_0000 || 0b0000_0000      ;R16=0b0000_0000
LDI R16, 0b0000_1010 || 0b0000_0000      ;R16=0b0000_0001
```

u) XOR:

```
LDI R16, 7 ^ 5           ;R16=2...UNO U OTRO PERO NO AMBOS
LDI R16, 5 ^ 7           ;R16=2...UNO U OTRO PERO NO AMBOS
```

Como en el ejemplo anterior no se puede visualizar apropiadamente por tratarse de números decimales, se sugiere que sean manipulados en binario:

```

;           7           5
;    _____  _____
LDI R16,0b0000_0111^0b0000_0101    ;R16=0b0000_0010...UNO U
;                                           ;OTRO PERO NO AMBOS

;    0b0000_0111
;    ^ 0b0000_0101
;    -----
;    0b0000_0010          R16=2

LDI R16,0b0000_1010^0b0000_1011    ;R16=0b0000_0001...UNO U
;                                           ;OTRO PERO NO AMBOS

;    0b0000_1010
;    ^ 0b0000_1011
;    -----
;    0b0000_0001          R16=1

```

Por ejemplo:

```
LDI R16, EXP2(5)           ;nos dará R16=32 (que sería 25)
```

- **LOG2(expression)**-regresa la parte entera del logaritmo natural de 2:

Por ejemplo:

```
LDI R16, LOG2(10)          ;nos dará R16=3 (el número completo  
                           ;sería 3.3219280948873626)
```

```
;      BYTE4 BYTE3 BYTE2 BYTE1  
; 0X  12    34    56    78
```

```
;                0X56                0X78  
; |-----BYTE2-----| |-----BYTE1-----|
```

```
;                0X56                0X78  
; |-----BYTE2-----| |-----BYTE1-----|  
; |-----HIGH-----| |-----LOW-----|
```

```

; _____
; |7|6|5|4|3|2|1|0| |7|6|5|4|3|2|1|0| |7|6|5|4|3|2|1|0| |7|6|5|4|3|2|1|0|
; |-----|
; |-----| |-----| |-----| |-----|
; 31-24 23-16 15_8 7-0
; |-----|
; 31-16 15-0
; |--- -BYTE4---| |--- -BYTE3---| |--- -BYTE2---| |--- -BYTE1---|

```

```

; |-----| |-----|
; HWRD LWRD
; |-----|
; PAGE (21-16)

```

```

LDI R18,BYTE1(0X12345678) ;R18=0x78
LDI R18,BYTE2(0X12345678) ;R18=0x56
LDI R18,BYTE3(0X12345678) ;R18=0x34
LDI R18,BYTE4(0X12345678) ;R18=0x12

LDI R18,LOW(0X12345678) ;R18=0x78
LDI R18,HIGH(0X12345678) ;R18=0x56

```



```

LDI R18,LOW(0b0001_0010_0011_0100_0101_0110_0111_1000)
;           1      2      3      4      5      6      7      8  =0x78

LDI R18,HIGH(0b0001_0010_0011_0100_0101_0110_0111_1000)
;           1      2      3      4      5      6      7      8  =0x56

LDI R18,BYTE1(0b0001_0010_0011_0100_0101_0110_0111_1000) ;BYTE1=0x78
;           1      2      3      4      5      6      7      8

LDI R18,BYTE2(0b0001_0010_0011_0100_0101_0110_0111_1000) ;BYTE2=0x56
;           1      2      3      4      5      6      7      8

LDI R18,BYTE3(0b0001_0010_0011_0100_0101_0110_0111_1000)
;           1      2      3      4      5      6      7      8  =0x34

LDI R18,BYTE4(0b0001_0010_0011_0100_0101_0110_0111_1000)
;           1      2      3      4      5      6      7      8  =0x12

```

En otra sintaxis para binario:

```

LDI R18,BYTE4(0b00010010001101000101011001111000)
;           1      2      3      4      5      6      7      8  =0x12

```

```

LDI R18,HIGH(BYTE1(0x12345678)) ;R18=0x00

```

```

LDI R18,LOW(LWRD(0b0001_0010_0011_0100_0101_0110_0111_1000))
;           1      2      3      4      5      6      7      8  =0x78
;                                     |-----LWRD-----|
;                                     |---LOW---|

```

```

LDI R18, HIGH (LWRD (0b0001_0010_0011_0100_0101_0110_0111_1000))
;           1      2      3      4      5      6      7      8      =0x56
;                                     |-----LWRD-----|
;                                     |--HIGH--|

LDI R18, LOW (HWRD (0b0001_0010_0011_0100_0101_0110_0111_1000))
;           1      2      3      4      5      6      7      8      =0x34
;           |-----HWRD-----|
;           |--LOW--|

LDI R18, HIGH (HWRD (0b0001_0010_0011_0100_0101_0110_0111_1000))
;           1      2      3      4      5      6      7      8      =0x12
;           |-----HWRD-----|
;           |--HIGH--|

LDI R18, PAGE (0b0001_0010_0011_0100_0101_0110_0111_1000)
;           1      2      3      4      5      6      7      8      =0b0011_0100=$34
;           |--PAGE--|
;           16-21

LDI R19, LOW (0x1234)      ;R19=0x34
;           --

LDI R19, HIGH (0x1234)     ;R19=0x12
;           --

```

(-\$0F83) = 0b0000_1111_1000_0011
 0 F 8 3

A2:

```

0b0000_1111_1000_0011
      ↓
0b1111_0000_0111_1100
+
-----
0b1111_0000_0111_1101   = $F07D

```

Este es el complemento
A2 del dato (-\$0F83)

Entonces:

```

LDI R17, LOW (-$0F83)      ;R17=0x7D
LDI R18, HIGH (-$0F83)     ;R18=0xF0

```