

Principios de Mecatrónica – SDI-11561

Ingeniería en Mecatrónica

Hugo Rodríguez Cortés

Departamento de Ingeniería Eléctrica y Electrónica

Instituto Tecnológico Autónomo de México

Agosto 2023

- MUL Rd, Rr Multiplicación de números sin signo, el resultado se almacena en R0 byte BAJO y R1 byte ALTO.
- MULS Rd, Rr Multiplicación de números con signo, el resultado se almacena en R0 byte BAJO y R1 byte ALTO.
- MULSU Rd, Rr Multiplicación de números con signo y números sin signo, el resultado se almacena en R0 byte BAJO y R1 byte ALTO.
- Ejemplo

```
LDI    R23, 0x25 ; —
LDI    R24, 0x65 ; —
MUL    R23, R25   ; 0x25 × 0x65 = 0xE99—
                ; R1 = 0x0E, R0 = 0x99—
```

- En 0x310 se encuentra el dato 0xFD, convertirlo a decimal guardando en 0x322, 0x323 y 0x324 los dígitos; en 0x322 el menos significativo.

```

.EQU    HEX_NUM=0x315      ;           —
.EQU    RMND_L=0x322      ;           —
.EQU    RMND_M=0x323      ;           —
.EQU    RMND_H=0x324      ;           —
.DEF    NUM=R20            ;           —
.DEF    DENOMINATOR = R21  ;           —
.DEF    QUOTIENT = R22     ;           —

        LDI    R16, 0xFD   ;           —
        STS    HEX_NUM, R16 ;           —
        LDS    NUM, HEX_NUM ;           —
        LDI    DENOMINATOR, 10 ;       —
        LDI    QUOTIENT, 0 ;           —

L1:     INC    QUOTIENT     ;           —
        SUB    NUM, DENOMINATOR ;       —
        BRCC   L1          ;   salta si C=0—
        DEC    QUOTIENT     ;           —
        ADD    NUM, DENOMINATOR ;       —
        STS    RMND_L, NUM  ;           —
        MOV    NUM, QUOTIENT ;           —
        LDI    QUOTIENT, 0 ;           —

L2:     INC    QUOTIENT     ;           —
        SUB    NUM, DENOMINATOR ;       —
        BRCC   L2          ;           —
        DEC    QUOTIENT     ;           —
        ADD    NUM, DENOMINATOR ;       —
        STS    RMND_M, NUM  ;           —

        STS    RMND_H, QUOTIENT ;       —
    
```

- Para representar el signo de un número con signo se utiliza el bit más significativo

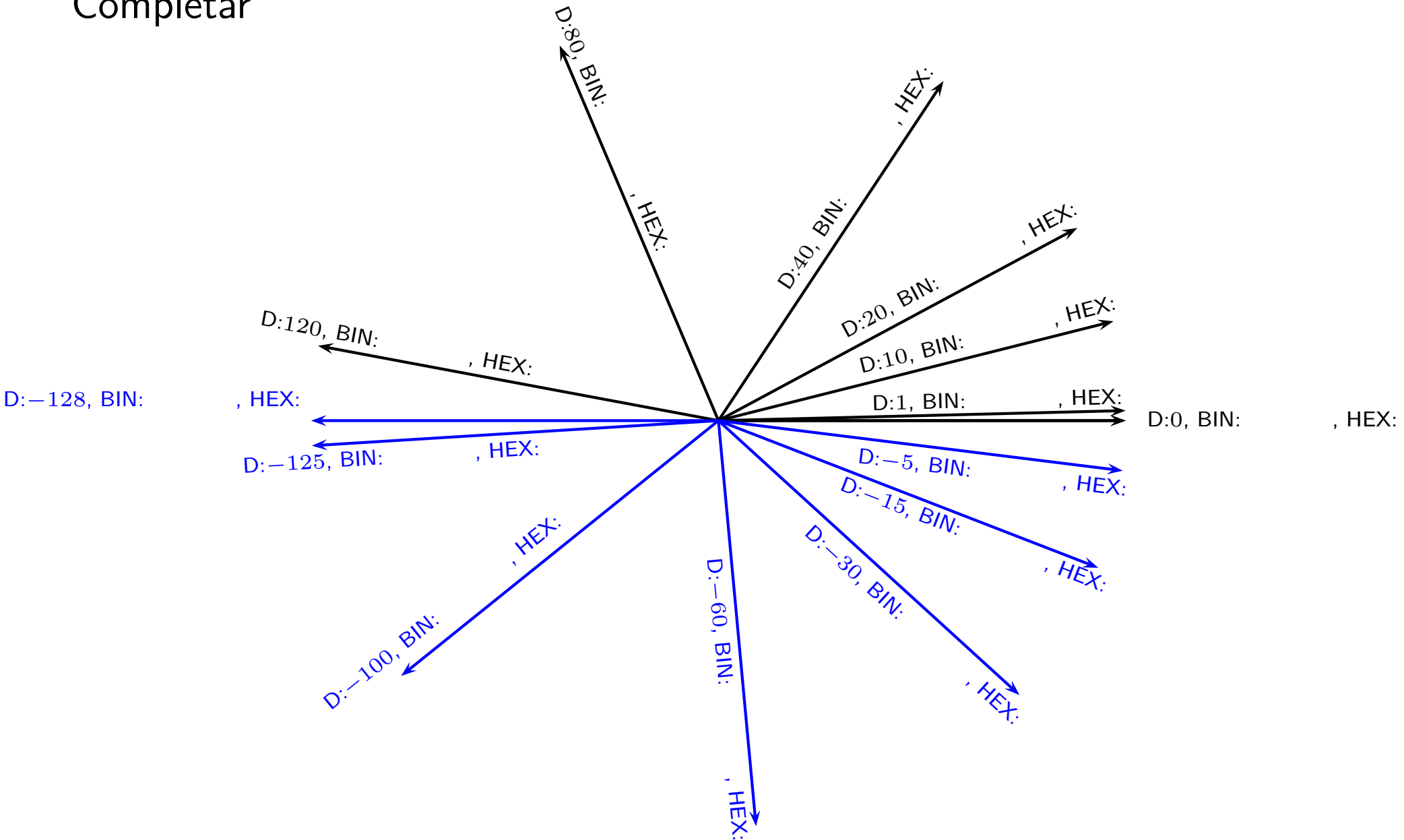
D7	D6	D5	D4	D3	D2	D2	D0
±	-	v	a	l	o	r	-

El rango de números positivos es de 0 a 127, si se requieren valores más grandes se deben utilizar operandos de 16 bits.

- En un número negativo $D7 = 1$ y el valor se representa con el complemento a 2.
- Ejemplo. Representar al número -5

0	0	0	0	0	1	0	1	5 en binario
1	1	1	1	1	0	1	0	complemento a 1
1	1	1	1	1	0	1	1	complemento a 2

Completar



- En operaciones con número con signo puede ocurrir un desbordamiento, el CPU indicará esta situación con la bandera V.
- Ejemplo.

```
LDI    R20, 0x60    ;    —  
LDI    R21, 0x46    ;    —  
ADD    R20, R21     ;    —
```

- En operaciones con número con signo puede ocurrir un desbordamiento, el CPU indicará esta situación con la bandera V.
- Ejemplo.

```
LDI    R20, 0x60 ;      —
LDI    R21, 0x46 ;      —
ADD    R20, R21   ;      —
```

$$0x60 + 0x46 = 0110\ 0000 + 0100\ 0110 = 1010\ 0110$$

El CPU indicará la situación con $V=1$. Notar que $N=1$ (resultado negativo).

- El rango de operación para números con signo son los números enteros $[-128, 127]$.

- En operaciones de 8 bits de números con signo la bandera V se asigna igual a 1 si alguna de las condiciones siguientes ocurre
 - ◆ Se lleva uno de D6 a D7 y no se lleva de D7 (C=0).
 - ◆ Se lleva uno de D7 (C=1) y no se lleva de D6 a D7
- Determine el valor de las banderas N y V en el siguiente código

```
LDI    R20, 0x80    ;      —  
LDI    R21, 0xFE    ;      —  
ADD    R20, R21     ;      —
```


- En operaciones de 8 bits de números con signo la bandera V se asigna igual a 1 si alguna de las condiciones siguientes ocurre
 - ◆ Se lleva uno de D6 a D7 y no se lleva de D7 (C=0).
 - ◆ Se lleva uno de D7 (C=1) y no se lleva de D6 a D7
- Determine el valor de las banderas N y V en el siguiente código

```
LDI    R20, 0x80    ;      —  
LDI    R21, 0xFE    ;      —  
ADD    R20, R21     ;      —
```

$N = 0, V = 1$

Examine el valor de las banderas V y N

- Sumar -2 con -5.

LDI	R20, -2	;	—
LDI	R21, -5	;	—
ADD	R20, R21	;	—

Examine el valor de las banderas V y N

■ Sumar -2 con -5.

```
LDI    R20, -2    ;      —
LDI    R21, -5    ;      —
ADD    R20, R21   ;      —
```

1	1	1	1	1	1	1	0	-2	
1	1	1	1	1	0	1	1	-5	, V = 0, N = 1
1	1	1	1	1	0	0	1	-7	

- Sumar 7 con 18.

LDI	R20, 7	;	—
LDI	R21, 18	;	—
ADD	R20, R21	;	—

- Sumar 7 con 18.

```
LDI    R20, 7      ; —
LDI    R21, 18     ; —
ADD    R20, R21    ; —
```

```
0 0 0 0 0 1 1 1 7
0 0 0 1 0 0 1 0 18 , V = 0, N = 0
0 0 0 1 1 0 0 1 25
```

- En operaciones sin signo C determina el estado de la operación, mientras que en operaciones con signo lo hace V.

- AND Rd, Rr Aplica el operador lógico AND bit a bit entre los registros Rd y Rr. ANDI Rd, k con k un valor constante.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

Esta operación puede activar las banderas Z, S y N.

LDI	R20, 0x35	;	—
ANDI	R20, 0x0F	;	—

- AND Rd, Rr Aplica el operador lógico AND bit a bit entre los registros Rd y Rr. ANDI Rd, k con k un valor constante.

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

Esta operación puede activar las banderas Z, S y N.

```
LDI    R20, 0x35 ; —
ANDI   R20, 0x0F ; —
```

```
0 0 1 1 0 1 0 1 0x35
0 0 0 0 1 1 1 1 0x0F , Z = 0, N = 0
0 0 0 0 0 1 0 1 0x05
```

- OR Rd, Rr Aplica el operador lógico OR bit a bit entre los registros Rd y Rr. ORI Rd, k con k un valor constante.

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

Esta operación puede activar las banderas Z, S y N.

```
LDI  R20, 0x04 ; —
ORI  R20, 0x30 ; —
```


- OR Rd, Rr Aplica el operador lógico OR bit a bit entre los registros Rd y Rr. ORI Rd, k con k un valor constante.

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

Esta operación puede activar las banderas Z, S y N.

```
LDI  R20, 0x04 ; —
ORI  R20, 0x30 ; —
```

```
0 0 0 0 0 1 0 0 0x04
0 0 1 1 0 0 0 0 0x30 , Z = 0, N = 0
0 0 1 1 0 1 0 0 0x34
```

- EXOR Rd, Rr Aplica el operador lógico XOR bit a bit entre los registros Rd y Rr.

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

Esta operación puede activar las banderas Z, S y N.

LDI	R20, 0x54	;	—
LDI	R20, 0x78	;	—
EOR	R20, R21	;	—

- EXOR Rd, Rr Aplica el operador lógico XOR bit a bit entre los registros Rd y Rr.

X	Y	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

Esta operación puede activar las banderas Z, S y N.

```
LDI    R20, 0x54 ;      —
LDI    R20, 0x78 ;      —
EOR    R20, R21  ;      —
```

```
0 1 0 1 0 1 0 0 0x54
0 1 1 1 1 0 0 0 0x78 , Z = 0, N = 0
0 0 1 0 1 1 0 0 0x2C
```

- Analice el siguiente código

```
LDI    R20, 0xFF    ;    —
OUT    DDRC, R20    ;    —
LDI    R20, 0x00    ;    —
OUT    DDRB, R20    ;    —
OUT    PORTC, R20   ;    —
LDI    R21, 0x45    ;    —
HERE:
IN      R20, PINB    ;    —
EOR     R20, R21     ;    —
BRNE    HERE        ;    —
LDI     R20, 0x99    ;    —
OUT     PORTC, R20   ;    —
```

- COM Rd Esta instrucción complementa bit a bit el contenido de un registro. Se conoce también como el complemento a uno.
- NEG Rd Esta instrucción calcula el complemento a dos de contenido del registro.
- CP Rd, Rr Esta instrucción es una resta entre Rd y Rr excepto que el valor de Rd no cambia.
- CPI Rd, k comparación con una constante. CP y CPI se unen a operadores de saltos condicionales.

- BCD (binary coded decimal) es la representación binaria de los números del 0 al 9. Se asocia a dos conceptos BCD sin comprimir y BCD comprimido.
- En un BCD sin comprimir el nibble bajo representa al BCD y el nibble alto se llena con ceros. Por ejemplo, $9 = 00001001$, $5 = 00000101$. Se utiliza un byte de memoria para almacenarlo.
- En un BCD comprimido. Los nibbles alto y bajo contienen un número. Por ejemplo, $0x59 = 01011001$. Se requiere la misma cantidad de memoria para almacenarlo, un byte
- En teclados ASCII cuando se activa la tecla '0' se envía $0110000 = 0x30$ al CPU. De forma similar '2' $= 0110010 = 0x32$.

- Para convertir de BCD comprimido a ASCII. Primero el BCD se descomprime y luego se combina con 0x30.

BCD comprimido	BCD sin comprimir		ASCII	
0x29	0x02	0x09	0x32	0x39
0010 1001	0000 0010	0000 1001	0011 0010	0011 1001

- Código

```

LDI    R20, 0x29 ;
MOV    R21, R20  ;
ANDI   R21, 0x0F ;
ORI    R21, 0x30 ;

MOV    R22, R20  ;
SWAP   R22       ; intercambia nibbles—
ANDI   R22, 0x0F ;
ORI    R22, 0x30 ;
    
```

- Para convertir de ASCII a BCD comprimido. Primero el BCD se descomprime y luego se combina para comprimirlo.

Tecla	ASCII	BCD sin comprimir	BCD comprimido
'4'	0x34	00000100	
'7'	0x37	00000111	01000111=0x47

- Código

```

LDI    R21, '4'    ;    —
LDI    R22, '7'    ;    —
ANDI   R21, 0x0F   ;    —
SWAP   R21         ;    —

ANDI   R22, 0x0F   ;    —
OR     R22, R21    ;    —
MOV    R20, R22    ;    —
    
```