

## Documentación del procesador

### Información general del procesador

- Palabra de control de 16 bits.
- 3 bits de dirección en el banco de registros (8 registros).
- Memoria de 16 bits de dirección.
- Memoria de instrucciones de 16 bits de dirección.
- Los datos se guardan a nivel de registro y memoria en 16 bits.
- El secuenciamiento de los saltos es implícito, es decir, a la hora de usar las instrucciones BZ o BNZ el salto (valor que se escribe en hexadecimal) es relativo al punto en que se encuentra, por lo que el valor se interpreta como complemento a 2.

### Binario de las instrucciones

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
0	0	0	0	d	d	d	a	a	a	f	f	f	b	b	b	Op. Lógicas y Aritméticas	AND, OR, XOR, NOT ADD, SUB, SHA, SHL
0	0	0	1	d	d	d	a	a	a	f	f	f	b	b	b	Comparación con y sin signo	CMPLT, CMPLT, -, CMPEQ CMPLTU, CMPLTU, -, -
0	0	1	0	d	d	d	a	a	a	n	n	n	n	n	n	Add inmediato	ADDI
0	0	1	1	d	d	d	a	a	a	n	n	n	n	n	n	Load	LD
0	1	0	0	b	b	b	a	a	a	n	n	n	n	n	n	Store	ST
0	1	0	1	d	d	d	0	n	n	n	n	n	n	n	n	Mover Inmediato	MOVI
							1	n	n	n	n	n	n	n	n		MOVHI
0	1	1	0	b	b	b	0	n	n	n	n	n	n	n	n	Salto condicional modo relativo al PC	BZ
							1	n	n	n	n	n	n	n	n		BNZ
0	1	1	1	d	d	d	0	n	n	n	n	n	n	n	n	Input	IN
				b	b	b	1	n	n	n	n	n	n	n	n	Output	OUT
1	x	x	x													No usado	8 códigos de operación

### Gestión de registros

- El registro R0 (@R0 = 000) está reservado a nivel de hardware y es de sólo lectura. Su valor es permanentemente 0x0000. Al intentar escribir sobre el registro R0 se escribirá sobre el registro R1.
- Los registros R1 (@R1 = 001), R2 (@R2 = 010), R3 (@R3 = 011), R4 (@R4 = 100), R5 (@R5 = 101), R6 (@R6 = 110) y R7 (@R7 = 111) son registros que se pueden utilizar (registros útiles).

### Sobre el ensamblador

El ensamblador diferencia dos tipos de datos principalmente: **valor** (en hexadecimal) y **registro**. Los registros útiles son 7 y se pueden usar bajo la sintaxis R[número del registro].

A continuación, se encuentra una tabla con todos los nemónicos del ensamblador y parámetros que toma:

Nemónico	Argumento 1	Argumento 2	Argumento 3
<b>Operaciones aritméticas</b>			
AND	Rd	Ra	Rb
OR	Rd	Ra	Rb
XOR	Rd	Ra	Rb
NOT	Rd	Ra	
ADD	Rd	Ra	Rb
SUB	Rd	Ra	Rb
SHA	Rd	Ra	Rb
SHL	Rd	Ra	Rb
ADDI	Rd	Ra	0xIMMED [Ca2]
<b>Operaciones lógicas</b>			
CMPLT	Rd	Ra	Rb
CMPLE	Rd	Ra	Rb
CMPEQ	Rd	Ra	Rb
CMPLTU	Rd	Ra	Rb
CMPLEU	Rd	Ra	Rb
<b>Operaciones de memoria</b>			
LD	Rd	Ra	0xVALUE [U]
ST	Rb	Ra	0xVALUE [U]
<b>Movimientos inmediatos de datos a registros</b>			
MOVI	Rd	0xVALUE [Ca2]	
MOVHI	Rd	0xVALUE [Ca2]	
<b>Salto condicionales</b>			
BZ	Rb	0xJUMP [Ca2]	
BNZ	Rb	0xJUMP [Ca2]	
<b>I/O</b>			
IN	Rd	0xPORT [U]	
OUT	Rb	0xPORT [U]	

Valores de 3 bits	Valores de 6 bits	Valores de 8 bits
-------------------	-------------------	-------------------

- [U]: Unsigned, representación en binario natural.
- [Ca2]: Representación en complemento a 2.
- Todos los registros se representan a nivel interno como binario natural.

Se debe tener en cuenta que tanto los valores de 6 bits como de ocho bits serán pasados por el IMMED. Por esta razón se extenderán en signo a 16 bits. Por ejemplo, en la instrucción:

MOVI R1 0xAB
--------------

El valor 0xAB se procesa como Ca2 de 8 bits, dado que 0xAB = 10101011, entonces extendido a 16 bits sería 1111 1111 1010 1011, que en hexadecimal sería 0xFFAB. Mientras que para MOVI R1 0x1A que sería 0x2A = 0010 1010 extendido en signo a 16 bits sería 0x002A.

Un programa sencillo para probar el procesador:

0000	0111 001 0 00000000	0x7200	IN R1 0x00
0001	0111 000 1 00000000	0x7100	OUT R0 0x00
0002	0111 001 1 00000000	0x7300	OUT R1 0x00
0003	0101 010 0 01011101	0x545D	MOVI R2 0X5D
0004	0000 011 001 100 010	0x0662	ADD R3 R1 R2
0005	0111 011 1 00000000	0x7700	OUT R3 P0
0006	0001 011 001 000 010	0x1642	CMPLT R3 R1 R2
0007	0110 011 1 00000010	0x6702	BNZ R3 0x02
0008	0111 011 1 00000000	0x7700	OUT R3 P0
0009	0100 001 000 000000	0x4200	ST R1 R0 0x00
000a	0000 001 001 010 010	0x0252	XOR R1 R1 R2
000b	0111 001 1 00000000	0x7300	OUT R1 P0
000c	0011 001 000 000000	0x3200	LD R1 R0 0x00
000d	0111 001 1 00000000	0x7300	OUT R1 P0

Programa para calcular el máximo común divisor de dos números entrados por el puerto 0x0000 de la entrada:

0000	0111001000000000	0x7200	IN R1 0x00
0001	0111010000000000	0x7400	IN R2 0x00
0002	0001011001011010	0x165A	CMPEQ R3 R1 R2

0003	0101010001011101	0x6707	BNZ R3 0x07
0004	0001100001001010	0x184A	CMPLE R4 R1 R2
0005	0110100100000011	0x6903	BNZ R4 0x03
0006	0000001001101010	0x026A	SUB R1 R1 R2
0007	0110000000000010	0x6002	BZ R0 0x02
0008	0000010010101001	0x04A9	SUB R2 R2 R1
0009	0110000011111001	0x60F9	BZ R0 0xF9
000a	0111001100000000	0x7300	OUT R1 0x00
000b	0110000011111111	0x60FF	BZ R0 0xFF