

	Tipo	Nombre	Abreviación	Operación	Ejemplo	Opcode
	Control	salto condicional en igualdad	SCI	if (R[reg1] == R[reg2]): PC = PC + dir	SCI R1, R2, salto donde: -R1 y R2 son los registros a comparar - salto es la etiqueta en el código a la que se quiere llegar.	10
		salto condicional en desigualdad	SCD	if (R[reg1] != R[reg2]): PC = PC + dir	SCD R1, R2, salto donde: -R1 y R2 son los registros a comparar - salto es la etiqueta en el código a la que se quiere llegar.	11
		salto incondicional	SI	PC = PC + dir	SI salto donde: - salto es la etiqueta a donde se quiere ir	00
		salto incondicional para función	SIF	RR = PC + 1 PC = dir	SIF salto donde: - salto es la etiqueta a donde se quiere ir - En RR se guarda la dirección de la siguiente instrucción de la que hizo el salto	01
	Memoria	guardar en memoria	GDR	Mem[R[reg2] + inm] = R[reg1] * solo se cambian los bits indicados por la sección (ver ejemplos)	GDR R1, 2 (R2) - en esta instrucción se guarda el valor de todos los bits del registro R1 en la dirección de memoria 1 + valor de R2	0
		cargar de memoria	CRG	R[reg1] = Mem[R[reg2] + inm] * solo se cambian los bits indicados por la sección (ver ejemplos)	CRG R1, 1 (R2) - en esta instrucción se carga en el registro R1 el valor contenido en la dirección de memoria 1 + valor de R2	1
	Datos	suma	SUM	R[reg1] = R[reg2] + R[reg3]	SUM R1, R2, R3 - en R1 se guardará el valor de la suma de R2 con R3	0000
		resta	RES	R[reg1] = R[reg2] - R[reg3]	RES R1, R2, R3 - en R1 se guardará el valor de la resta de R2 con R3	0001
		multiplicación	MUL	R[reg1] = R[reg2] * R[reg3]	MUL R1, R2, R3 - en R1 se guardará el valor de la multiplicación de R2 con R3	0010
		división entera	DIV	R[reg1] = R[reg2] // R[reg3]	DIV R1, R2, R3 - en R1 se guardará el valor de la división entera de R2 con R3	0011
		residuo	RSD	R[reg1] = R[reg2] % R[reg3]	RSD R1, R2, R3 - en R1 se guardará el valor de la residuo de R2 con R3	0100
		suma con inmediato	SUMI	R[reg1] = R[reg1] + inm	SUMI R1, R2, inm - en R1 se guardará el valor de la suma de R2 con el inm	1000
		resta con inmediato	RESI	R[reg1] = R[reg1] - inm	RESI R1, R2, inm - en R1 se guardará el valor de la resta de R2 con el inm	1001
		multiplicación con inmediato	MULI	R[reg1] = R[reg1] * inm	MULI R1, R2, inm - en R1 se guardará el valor de la mult de R2 con el inm	1010
		división entera con inmediato	DIVI	R[reg1] = R[reg1] // inm	DIVI R1, R2, inm - en R1 se guardará el valor de la divi de R2 con el inm	1011
		residuo con inmediato	RSDI	R[reg1] = R[reg1] % inm	RSDI R1, R2, inm - en R1 se guardará el valor del residuo de R2 con el inmediato	1100

	Operación	JumpI	JumpCI	JumpCD	MemToReg	MemRead	MemWrite	ALUOp	ALUSrc	RegWrite	ImmSrc	RegDtn	RegSrc2	RegSrc1
	Datos con inmediato	0	0	0	0	0	0	* depende de la operación, ver tabla ALU	1	1	10	01	X	10
	Datos sin inmediato	0	0	0	0	0	0	* depende de la operación, ver tabla ALU	0	1	XX	01	1	10
	Control	1 si es SI, 0 si no	1 si es SCI, 0 si no	1 si es SCD, 0 si no	0	0	0	001 (operacion de resta)	0	0	00	X	0	00
	Memoria GDR	0	0	0	X	0	1	000 (operacion de suma)	1	0	01	00	X	01
	Memoria CRG	0	0	0	1	1	0	000 (operacion de la suma)	1	1	01	00	X	01