				Control								Control (condiciona	l)		
	Tipo	OPCODE		Detalles cond / no cond			tipo	OPCODE		Reg 1 (Src)	Reg 2 (Src)	dir. ⁻	Tan		
	Про	Condiciona	Instrucción	Reg 1 (Src)	Reg 2 (Src)	Direcció	on del Tag		про	branch	ins	rteg i (ore)	110g Z (010)	dii.	
SCI	00	1	0	4'XXXX (R1)	4'XXXX (R2)	20'XXXXXXXXXXXXXX	XXXXXXXX (Dirección)		31-30	29	28	27-24	23-20	19	-0
SCD	00	1	1	4'XXXX (R1)	4'XXXX (R2)	20'XXXXXXXXXXXXXX	XXXXXXXX (Dirección)								
SI	00	0	0	28'	xxxxxxxxxxxxx	XXXXXXXXXXX (Inmed	diato)			Control (incondicional)					
SIF	00	0	1	28'	xxxxxxxxxxxxxx	XXXXXXXXXXX (Inmed	diato)		tipo	OPC	ODE		Inmed	diata	
									про	branch	ins				
					Memoria				31-30	29	28		27-0)	
	Tipo		ODE			alles									
	·		ucción	Relleno	Reg 1 (Destino)	Reg 2 (Src)	Inmediato			Memoria			Т		
GDR	01		0	3'000	4'XXXX (R1)	4'XXXX (R2)	18'XXXX		tipo	OPC		Reg 1 (Destino)	Reg 2 (Src)	Inme	diato
CRG	01		1	3'000	4'XXXX (R1)	4'XXXX (R2)	18'XXXX			Instrucción	Relleno				
					Datas				31-30	29	000	25-22	21-18	17	-0
		OPC	CODE		Datos	alles						Datos (sin inmediato	. \		
Inst sin Inm	Tipo		Instrucción	Relleno	Reg 1 (Destino)	Reg 2 (Src)	Reg 3 (Src)			OPC		Datos (Sili lillilediate	, 		
SUM	10	0	0000	13'00000000000000	4'XXXX (R1)	4'XXXX (R2)	4'XXXX (R3)		tipo	Flag Inm	ins	Reg 1 (Destino)	Reg 2 (Src)	Reg 3 (Src)	Relleno
RES	10	0	0001	13'00000000000000	4'XXXX (R1)	4'XXXX (R2)	4'XXXX (R3)		31-30	29	28-25	24-21	20-17	16-13	12-0
MUL	10	0	0010	13'00000000000000	4'XXXX (R1)	4'XXXX (R2)	4'XXXX (R3)		0.00	20	20 20	2.2.	20	10.10	
DIV	10	0	0011	13'00000000000000	4'XXXX (R1)	4'XXXX (R2)	4'XXXX (R3)				Г	Datos (con inmediate	n)		
RSD	10	0	0100	13'00000000000000	4'XXXX (R1)	4'XXXX (R2) 4'XXXX (R3)				OPC	ODE		ĺ		
		OPC	CODE			4 *** (R2) 4 *** (R3)			tipo	Flag Inm	ins	Reg 1 (Destino)	Reg 2 (Src)	Inmediato	
Inst con Inm	Tipo	Flag Inm	Instrucción	Reg 1 (Destino)	Reg 2 (Src)	Inme	ediato		31-30	29	28-25	24-21	20-17	16	-0
SUMI	10	1	1000	4'XXXX (R1)	4'XXXX (R2)	17'XXXXXXXXXXXX	XXXXXX (Inmediato)								
RESI	10	1	1001	4'XXXX (R1)	4'XXXX (R2)	17'XXXXXXXXXXXXX	XXXXXX (Inmediato)								
MULI	10	1	1010	4'XXXX (R1)	4'XXXX (R2)	17'XXXXXXXXXXXXX	XXXXXX (Inmediato)								
DIVI	10	1	1011	4'XXXX (R1)	4'XXXX (R2)	17'XXXXXXXXXXXXX	XXXXXX (Inmediato)								
RSDI	10	1	1100	4'XXXX (R1)	4'XXXX (R2)	17'XXXXXXXXXXXXX	XXXXXX (Inmediato)								

Tipo	Nombre	Abreviación	Operación	Ejemplo	Opcode
	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
Control	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
	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
Memoria	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 c	1
	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
Datos	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	Jumpl	JumpCl	JumpCD	MemToReg	MemRead	MemWrite	ALUOp	ALUSrc	RegWrite	ImmSrc	RegDtn	RegSrc2	RegSrc
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	х	0	00
Memoria GDR	0	0	0	х	0	1	000 (operacion de suma)	1	0	01	00	Х	01
Memoria CRG	0	0	0	1	1	0	000 (operacion de la suma)	1	1	01	00	Х	01