

CS2010

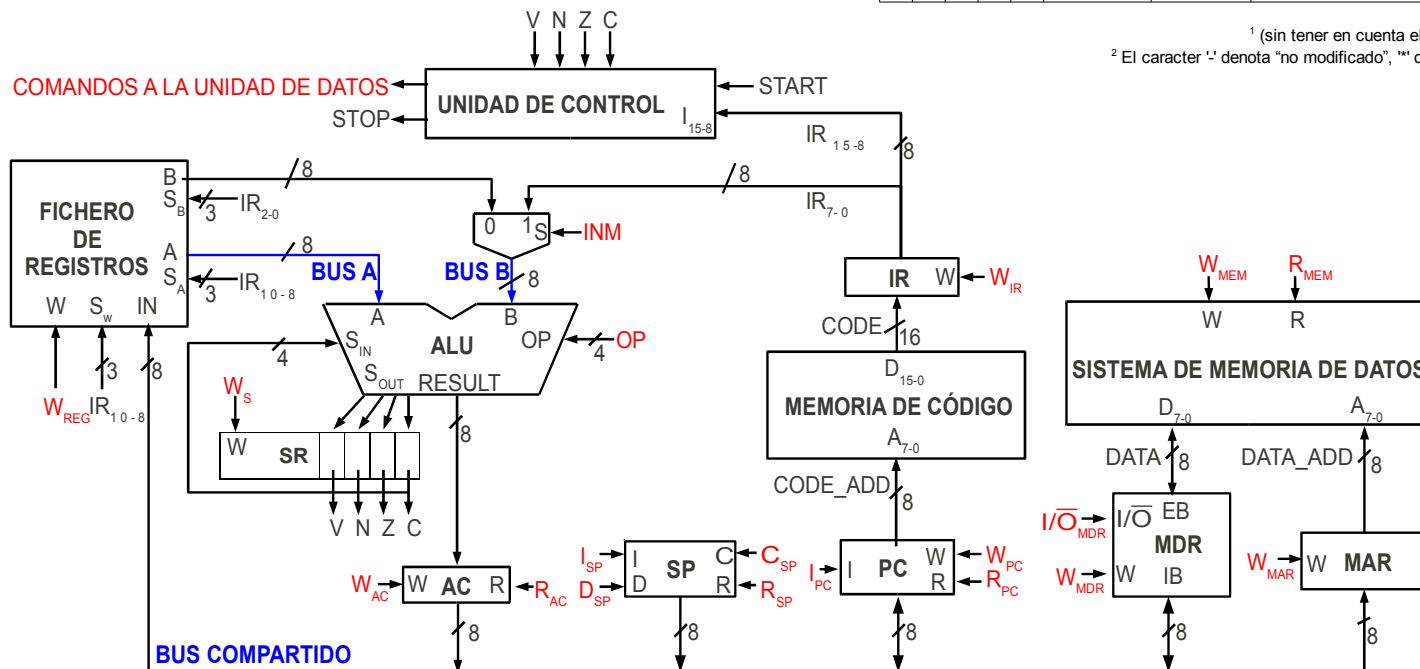
formato	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A instrucción con operando registro	código de operación					registro destino (fuente en ST)			-	-	-	-	-	registro fuente (registro base en ST/LD)		
B instrucción con operando memoria o inmediato									dato inmediato / dirección del dato							
C instrucción de salto						condición de salto			dirección de salto							

I ₁₀	I ₉	I ₈	CONDICIÓN	nmónico(s) de la condición	notas
0	0	0	Z	ZS, EQ	será cierta justo tras realizar la resta A-B si y solo si A=B
0	0	1	C	CS, LO	será cierta justo tras realizar la resta A-B si y solo si A<B asumiendo notación base 2 sin signo
0	1	0	V	VS	será cierta si y solo si el dato recién calculado no es representable en notación complemento a 2
0	1	1	N xor V	LT	será cierta justo tras realizar la resta A-B si y solo si A<B asumiendo notación complemento a 2
1	-	-	?	-	estas condiciones no están definidas y no deben utilizarse

Bits del código de operación					NEMÓNICO	FORMATO	TIPO	SINTAXIS	EFFECTO ¹	VNZC ²
15	14	13	12	11						
0	0	0	0	0	ST	A	memoria	ST (Rbase), Rfuente	MEM[Rbase] ← Rfuente	----
0	0	0	0	1	LD	A	memoria	LD Rdestino, (Rbase)	Rdestino ← MEM[Rbase]	----
0	0	0	1	0	STS	B	memoria	STS dirección, Rfuente	MEM[dirección] ← Rfuente	----
0	0	0	1	1	LDS	B	memoria	LDS Rdestino, dirección	Rdestino ← MEM[dirección]	----
0	0	1	0	0	CALL	C	salto	CALL dirección	MEM[SP] ← PC, SP ← SP-1, PC ← dirección	----
0	0	1	0	1	RET	-	salto	RET	PC ← MEM[SP+1], SP ← SP+1	----
0	0	1	1	0	BRxx	C	salto	BRxx dirección	xx: PC ← dirección	----
0	0	1	1	1	JMP	C	salto	JMP dirección	PC ← dirección	----
0	1	0	0	0	ADD	A	aritmético/lógica	ADD Rdestino, Rfuente	Rdestino ← Rdestino + Rfuente	****
0	1	0	0	1	-	-	-	-	no documentado	UUUU
0	1	0	1	0	SUB	A	aritmético/lógica	SUB Rdestino, Rfuente	Rdestino ← Rdestino - Rfuente	****
0	1	0	1	1	CP	A	estado	CP Rdestino, Rfuente	NOP	****
0	1	1	0	0	-	-	-	-	no documentado	UUUU
0	1	1	0	1	-	-	-	-	no documentado	UUUU
0	1	1	1	0	-	-	-	-	no documentado	UUUU
0	1	1	1	1	MOV	A	movimiento de datos	MOV Rdestino, Rfuente	Rdestino ← Rfuente	----
1	0	0	0	0	-	-	-	-	no documentado	UUUU
1	0	0	0	1	-	-	-	-	no documentado	UUUU
1	0	0	1	0	CLC	-	estado	CLC	NOP	----
1	0	0	1	1	SEC	-	estado	SEC	NOP	----
1	0	1	0	0	ROR	A o B	desplazamiento	ROR Rdestino	Rdestino ← SHR(Rdestino, C)	****
1	0	1	0	1	ROL	A o B	desplazamiento	ROL Rdestino	Rdestino ← SHL(Rdestino, C)	****
1	0	1	1	0	-	-	-	-	no documentado	UUUU
1	0	1	1	1	STOP	-	especial	STOP	lleva el procesador a espera	----
1	1	0	0	0	ADDI	B	aritmético/lógica	ADDI Rdestino, dato	Rdestino ← Rdestino + dato	****
1	1	0	0	1	-	-	-	-	no documentado	UUUU
1	1	0	1	0	SUBI	B	aritmético/lógica	SUBI Rdestino, dato	Rdestino ← Rdestino - dato	****
1	1	0	1	1	CPI	B	estado	CPI Rdestino, dato	NOP	****
1	1	1	0	0	-	-	-	-	no documentado	UUUU
1	1	1	0	1	-	-	-	-	no documentado	UUUU
1	1	1	1	0	-	-	-	-	no documentado	UUUU
1	1	1	1	1	LDI	B	movimiento de datos	LDI Rdestino, dato	Rdestino ← dato	----

¹ (sin tener en cuenta el registro de estado y el incremento del PC)

² El caracter '-' denota "no modificado", '*' denota "modificado de forma definida", 'U' denota "no documentado"



ALU

OP ₃	OP ₂	OP ₁	OP ₀	RESULT =	V _{OUT} =	N _{OUT} =	Z _{OUT} =	C _{OUT} =
0	0	-	0	-	V _{IN}	N _{IN}	Z _{IN}	0
0	0	0	1	-	-	-	-	-
0	0	1	1	-	V _{IN}	N _{IN}	Z _{IN}	1
0	1	0	0	SHR(A, C _{IN})	C _{IN} EXOR A ₃	RESULT ₇	NOT OR ₃₌₀ ⁷ (RESULT ₁)	A ₃
0	1	0	1	SHL(A, C _{IN})	A ₂ EXOR A ₆	RESULT ₇	NOT OR ₃₌₀ ⁷ (RESULT ₁)	A ₂
0	1	1	-	A	-	-	-	-
1	0	0	-	(A + B) mod 2 ⁸	overflow(A+B)	RESULT ₇	NOT OR ₃₌₀ ⁷ (RESULT ₁)	carry(A+B)
1	0	1	-	(A - B) mod 2 ⁸	underflow(A-B)	RESULT ₇	NOT OR ₃₌₀ ⁷ (RESULT ₁)	borrow(A-B)
1	1	-	-	B	-	-	-	-