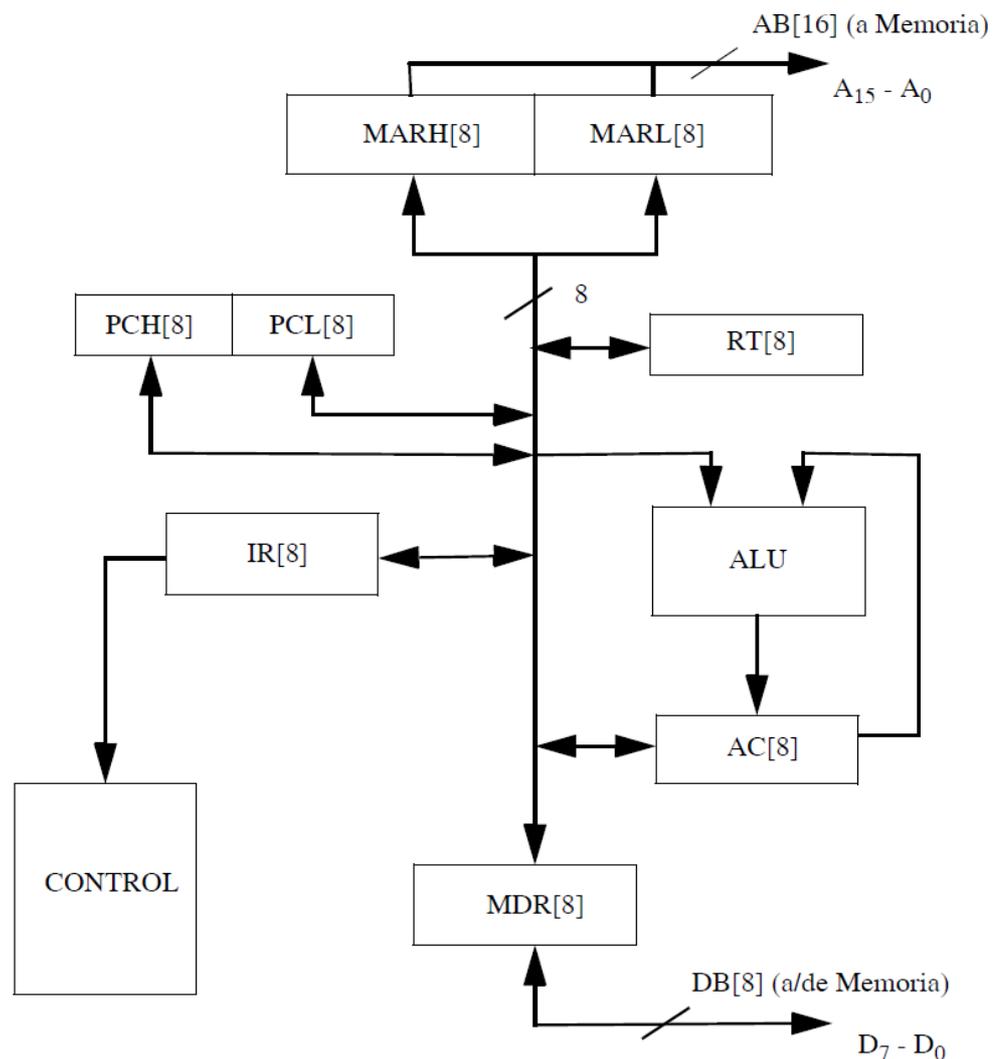


Problema 1.- La siguiente figura muestra la Unidad de Procesado de Datos de un microprocesador con arquitectura von Neumann. Como puede observarse, el bus de datos (D7-D0) es de 8 bits y el de direcciones (A15-A0) de 16 bits. Los registros visibles son los siguientes:

- PC (Program Counter): es de 16 bits y está formado por la concatenación de dos registros de 8 bits (PCH y PCL).
- AC (Acumulador): es de 8 bits.

Además, la unidad de procesamiento dispone de los siguientes registros ocultos:

- MAR (Memory Address Register): de 16 bits, está formado por la concatenación de dos registros de 8 bits: MARH y MARL.
- MDR (Memory Data Register): de 8 bits, su finalidad es servir como registro intermedio entre el procesador y la memoria externa. Todo dato que entre o salga del procesador deberá ser almacenado previamente en MDR.
- IR (Instruction Register): es de 8 bits.
- RT (Registro Temporal): es de 8 bits y es utilizado para el cálculo de operaciones intermedias.



Las posiciones de memoria tienen un ancho de 8 bits, y cada instrucción ocupa tres posiciones consecutivas: la primera indica el código de operación, la segunda contiene los 8 bits más significativos del operando y la tercera los 8 bits menos significativos. Consideremos por ejemplo la instrucción LDA. Dicha instrucción escribe el contenido de la posición de memoria indicada por el operando en AC. La instrucción 'LDA \$B043' ocuparía 3 posiciones: en la primera se encontraría el código de operación correspondiente a LDA, en la segunda habría escrito \$B0 y en la tercera \$43. Indique la secuencia de microoperaciones necesarias para realizar dicha instrucción y diga cuáles corresponden al ciclo de FETCH y cuáles al de EXECUTE.

Problema 2.- Describa en Verilog la unidad de control del CS1 (ver Figura 1).

Problema 3.- Se desea cambiar la semántica de las instrucciones del CS1. La siguiente tabla muestra la semántica de las nuevas instrucciones junto a su código de operación:

Código de operación ($IR_{7,6}$)	Semántica
00	$Rd \leftarrow Rd + 2 Rf$
10	$Rd \leftarrow Rd - 2 Rf$
01	$Rd \leftarrow 2 Rf$
11	$Rd \leftarrow 0$

Dibuje las cartas ASM de datos y control modificadas para que se realicen las nuevas instrucciones (véase Figura 1 y Figura 2).

Problema 4.- Describa en Verilog la unidad de control del CS2. Las cartas ASM del CS2 se ilustran en la Figura 3 (la primera parte coincide con la carta de la Figura 1).

Problema 5.- Se desea cambiar la semántica de los dos primeros códigos de operación del CS2. Dichos códigos pasaran a codificar una no-operación y una instrucción de carga de operando inmediato tal y como se ilustra en la siguiente tabla:

Código de operación (IR_{13-11})	Semántica
000	NOP
001	$Rd \leftarrow \text{operando}$

Dibuje las cartas ASM de datos y control modificadas para que se realicen las nuevas instrucciones (véase Figura 3 y Figura 4).

Problema 6.- Se desean añadir al CS2010 dos instrucciones que permiten la inserción (PUSH Ri) y extracción (POP Ri) de datos en la pila. Asigne un código de operación adecuado e indique sus microoperaciones (véase Figura 5 y Figura 6).

Problema 7.- Se desea cambiar la semántica de las instrucciones ST y LD del CS2010 para que usen el modo de direccionamiento indirecto con postincremento. Su semántica es idéntica a la del modo indirecto normal salvo que, tras realizarse el acceso a memoria, el registro base se incrementa en una unidad.

- Indique que cambios habría que realizar a la unidad de datos del sistema para que pudiera implementarse dicho modo (véase Figura 5).
- Descomponga las nuevas instrucciones en microoperaciones.

Problema 8.- Diseñe una subrutina para el CS2010 que calcule la división de dos números sin signo. El dividendo está almacenado en el registro R0 y el divisor en R1. El cociente debe quedar guardado en R2 y el resto en R3.

Problema 9.- Se tiene una tabla de 10 números con signo almacenados a partir de la dirección \$34 de la memoria. Diseñe una subrutina para el CS2010 que los ordene de mayor a menor.

Problema 10.- Observe la descripción a nivel RT de la ALU usada en la implementación propuesta de la arquitectura del CS2010. Discuta que utilidad puede tener que las salidas de estado tomen los valores que aparecen en la tabla.

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 ₀	RESULT ₇	NOT OR _{i=0} ⁷ (RESULT _i)	A ₀
0	1	0	1	SHL(A, C _{IN})	A ₇ EXOR A ₆	RESULT ₇	NOT OR _{i=0} ⁷ (RESULT _i)	A ₇
0	1	1	-	A	-	-	-	-
1	0	0	-	(A + B) mod 2 ⁸	overflow(A+B)	RESULT ₇	NOT OR _{i=0} ⁷ (RESULT _i)	carry(A+B)
1	0	1	-	(A - B) mod 2 ⁸	underflow(A-B)	RESULT ₇	NOT OR _{i=0} ⁷ (RESULT _i)	borrow(A-B)
1	1	-	-	B	-	-	-	-

Problema 11.- Describa en Verilog los registros usados en la implementación propuesta de la arquitectura del CS2010.

Problema 12.- En el CS2010 la condición a testear en los saltos condicionales se codifica en los bit 8, 9 y 10 de la instrucción (véase Figura 6). Hay cuatro códigos de condición no utilizados a los que se desea asociar las siguientes condiciones de salto:

I ₁₀	I ₉	I ₈	CONDICIÓN	nmónico(s) de la condición	notas
1	0	0	NOT Z	ZC, NE	será cierta justo tras realizar la resta A-B si y solo si A≠B
1	0	1	NOT C	CC, SH	será cierta justo tras realizar la resta A-B si y solo si A≥B asumiendo notación base 2 sin signo
1	1	0	NOT V	VC	será cierta si y solo si el dato recién calculado es representable asumiendo notación complemento a 2
1	1	1	N nexor V	GE	será cierta justo tras realizar la resta A-B si y solo si A≥B en notación complemento a 2

Indique los cambios que habría que realizar sobre las cartas ASM del CS2010 para implementar las nuevas condiciones de salto.

Problema 13.- Se desea cambiar la semántica de las instrucciones ST y LD del CS2010 para que usen direccionamiento indirecto de registro con desplazamiento. En este modo de direccionamiento el dato se encuentra en una posición de memoria cuya dirección se obtiene sumando el contenido de un registro (registro base) y una constante codificada en la propia instrucción (desplazamiento). Su sintaxis ensamblador será: <desplazamiento> (<registro base>)

A modo de ejemplo, al ejecutarse las siguientes instrucciones el registro R0 se escribiría con el valor 23 y el registro R1 se escribiría con el contenido de la posición de memoria 33.

```
LDI R0,23
LD R1,10(R0)
```

El desplazamiento se codificará en los bits no usados del formato A (véase Figura 6) en notación complemento a dos.

- Indique como habría que modificar la unidad de datos de la implementación propuesta para hacer esto posible (véase Figura 5).
- Indique como habría que modificar las carta ASM de datos y control.

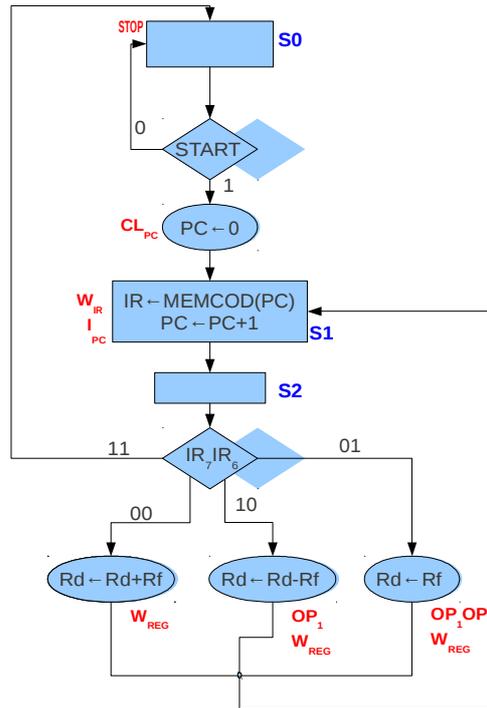


Figura 1: Carta ASM de datos y control del CS1

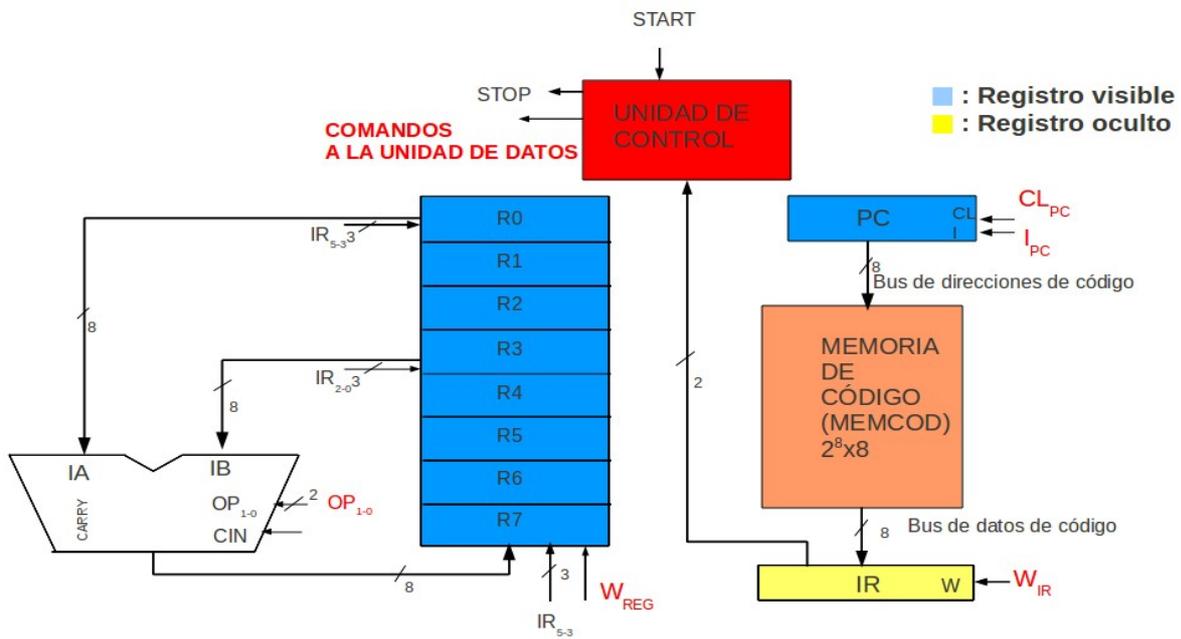


Figura 2: Unidad de datos del CS1

CO: IR7 IR6	SINTAXIS	FUNCIÓN
00	ADD Rd,Rf	Rd←Rd+Rf
10	SUB Rd,Rf	Rd←Rd-Rf
01	MOV Rd,Rf	Rd←Rf
11	STOP	NOP

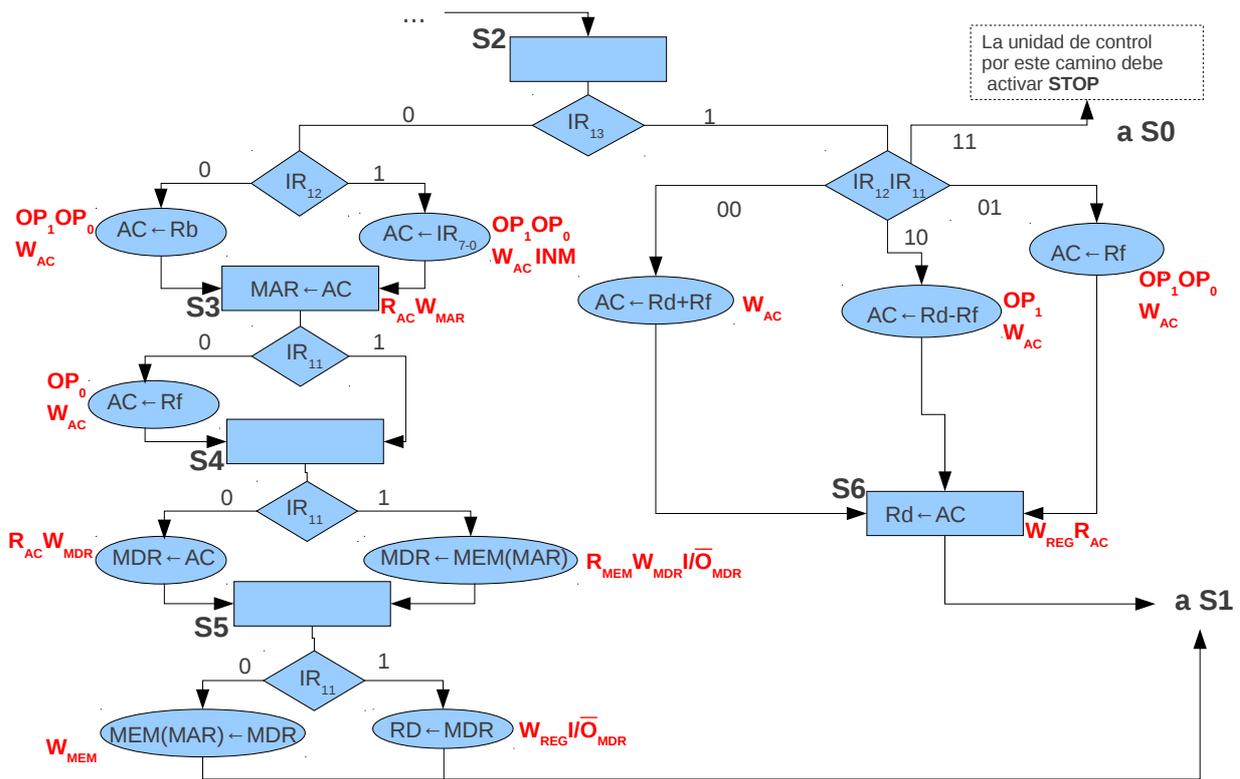


Figura 3: Carta ASM de datos y control del CS2

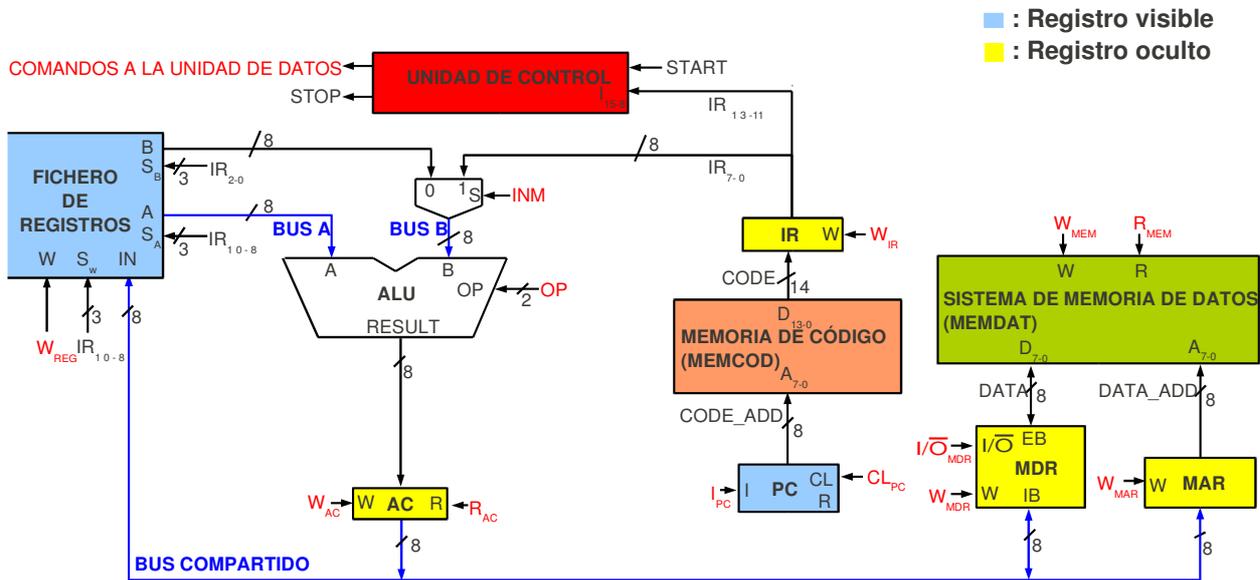


Figura 4: Unidad de datos del CS2

CO	SINTAXIS	FUNCIÓN
000	ST (Rb),Rf	MEMDAT(Rb)←Rf
001	LD Rd, (Rb)	Rd←MEMDAT(Rb)
010	STS dir, Rf	MEMDAT(dir)←Rf
011	LDS Rd,dir	Rd←MEMDAT(dir)
100	ADD Rd,Rf	Rd←Rd+Rf
110	SUB Rd,Rf	Rd←Rd-Rf
101	MOV Rd,Rf	Rd←Rf
111	STOP	NOP

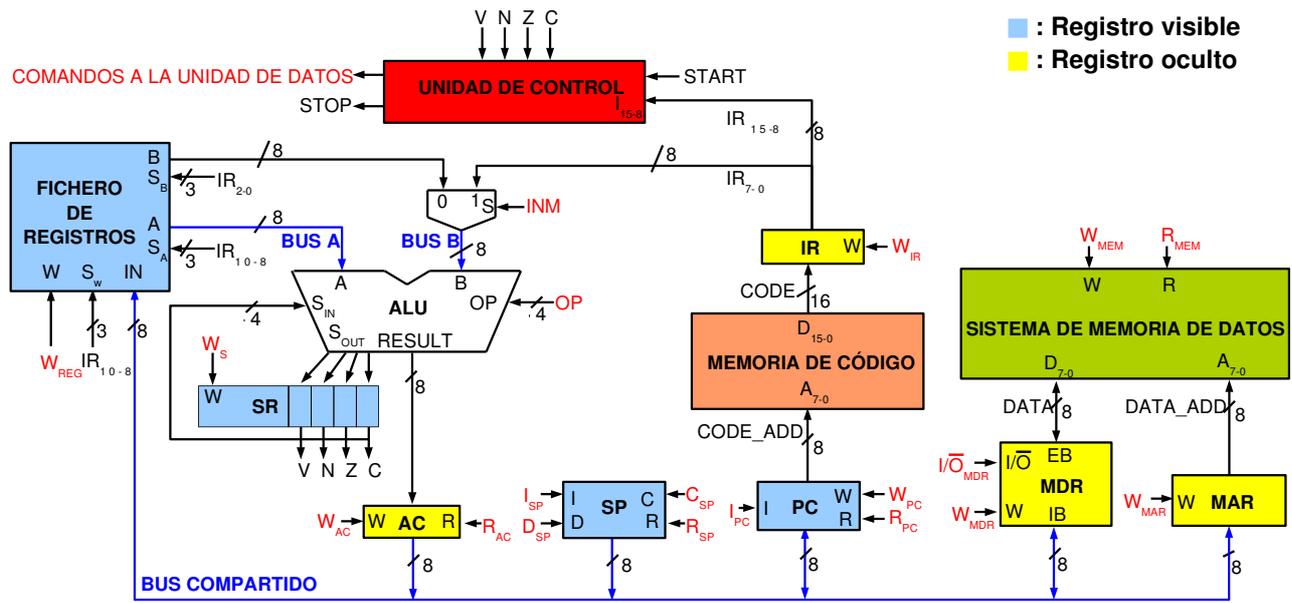


Figura 5: Unidad de datos del CS2010

Bits del código de operación					NEMÓNICO	FORMATO	TIPO	SINTAXIS	EFECTO 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)	Rfuente←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	Rfuente←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←Rdestino	----
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"

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	nemónico(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

Figura 6: Formatos y códigos de operación de las instrucciones del CS2010