

**Grado Ingeniería Informática. Tecnologías Informáticas.**  
**Estructura de Computadores. Examen final.**  
**1ª Convocatoria. Junio 2011.**

1. Se propone la arquitectura modificada que se muestra en la figura 1 para el CS2010.

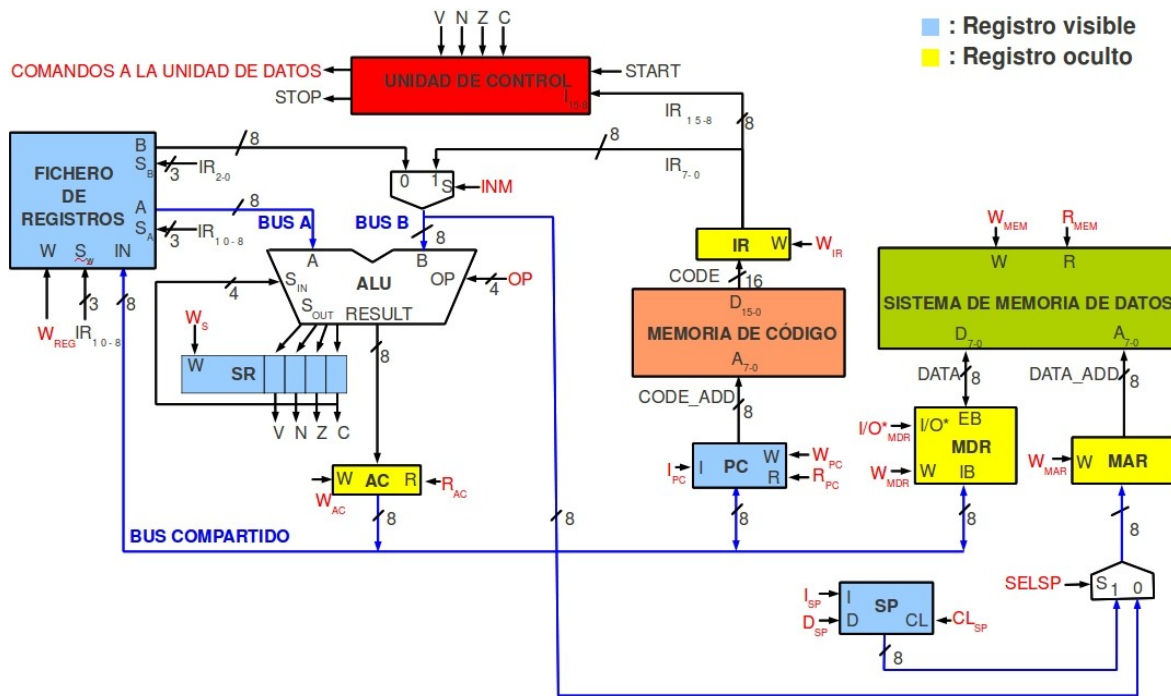


Figura 1

Para la nueva arquitectura indique la descomposición en microoperaciones de las siguientes instrucciones: CALL, ST, LD, STS, LDS.

2. Para la unidad de datos de la figura 2. Diseñe un sistema digital que realice las operaciones que se muestran en la tabla. No considere problemas de desbordamiento.

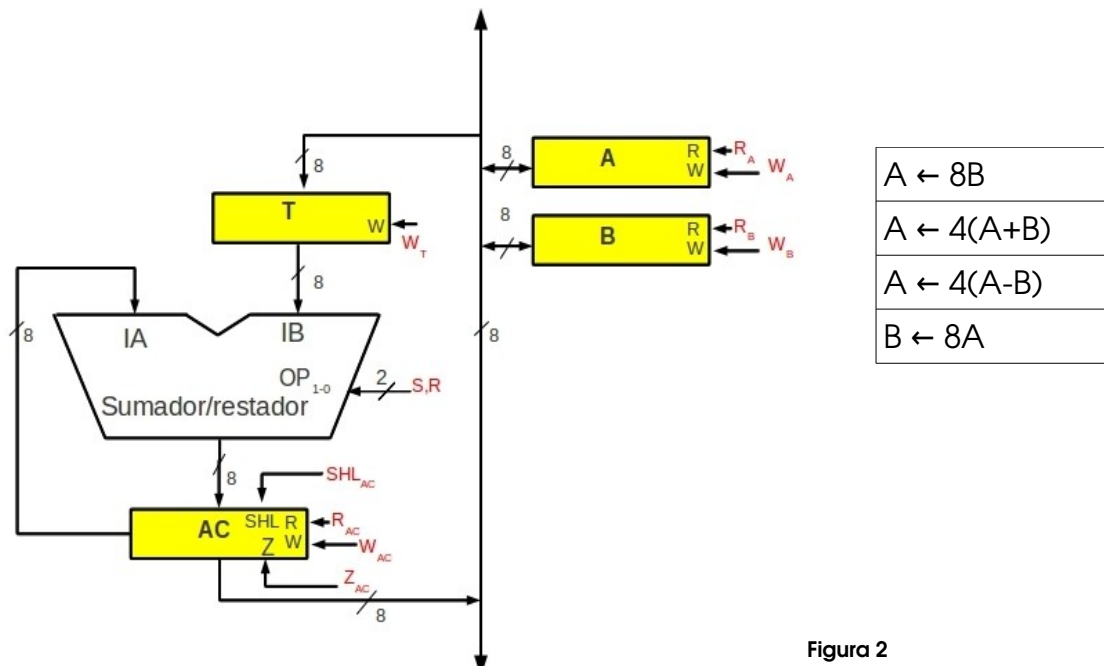


Figura 2

Especifique:

a) Diagrama de bloques del sistema digital distinguiendo unidad de datos y de control y señales que las comunican.

b) Carta ASM con indicación tanto de las transferencias entre registros como de las señales que se activan.

3. A partir de las direcciones \$100 y \$150 de la memoria de datos del ATmega328pa, se encuentran dos vectores, A y B, de 32 elementos con signo de tamaño byte. Se desea diseñar una subrutina que permita generar, con los vectores anteriores, un nuevo vector, C, que se almacenará a partir de la dirección \$200. En general, el elemento  $C(i) = \text{mayor}(A(i), B(i))$  (para  $i=0..31$ ) siempre que  $|A(i)| < 50$ . En caso contrario  $C(i) = B(i)/2$ . El pseudocódigo que permite realizar la operación que se describe es el que se muestra a continuación:

```
subr Calculo
X <- VectorA
Y <- VectorB
Z <- VectorC
i <- 0
hacer
    A <- (X), X <- X+1
    B <- (Y), Y <- Y+1
    si(abs(A) < 50)
        si (A > B)
            (Z) <- A
        sino
            (Z) <- B
        fsi
        Z <- Z + 1
    sino
        (Z) <- B/2, Z <- Z+1
    fsi
    i <- i+1
mientras(i < 32)
fin Calculo

subr abs
si (R20[7] == 1)
    R20 <- -R20
fsi
fin abs
```

Realice el programa en ensamblador.

**Nota: todos los ejercicios puntúan igual.**

# ANEXO 1: INFORMACIÓN CS2010

Descripción ALU	OP <sub>3</sub>	OP <sub>2</sub>	OP <sub>1</sub>	OP <sub>0</sub>	RESULT=	V <sub>OUT</sub> =	N <sub>OUT</sub> =	Z <sub>OUT</sub> =	C <sub>OUT</sub> =
	0	0	-	0	-	V <sub>IN</sub>	N <sub>IN</sub>	Z <sub>IN</sub>	0
	0	0	1	1	-	V <sub>IN</sub>	N <sub>IN</sub>	Z <sub>IN</sub>	1
	0	1	0	0	SHR(A, C <sub>IN</sub> )	c <sub>IN</sub> EXOR A <sub>0</sub>	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	A <sub>0</sub>
	0	1	0	1	SHL(A, C <sub>IN</sub> )	A <sub>7</sub> EXOR A <sub>6</sub>	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	A <sub>7</sub>
	0	1	1	-	A	-	-	-	-
	1	0	0	-	(A + B) mod 2 <sup>8</sup>	overflow(A+B)	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	carry(A+B)
	1	0	1	-	(A - B) mod 2 <sup>8</sup>	underflow(A-B)	RESULT <sub>7</sub>	NOT OR <sub>i=0</sub> <sup>7</sup> (RESULT <sub>i</sub> )	borrow(A-B)
	1	1	-	-	B	-	-	-	-

## Descripción registro MDR

W	I/O*	MDR←	IB=	EB=
0	0	[MDR]	H.I.	[MDR]
0	1	[MDR]	[MDR]	H.I.
1	0	IB	H.I.	H.I.
1	1	EB	H.I.	H.I.

## Tabla de Instrucciones

Bits del código de operación					NEMÓNICO	FORMATO	TIPO	SINTAXIS	EFECTO <sup>1</sup>	VNZC <sup>2</sup>
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	----

<sup>1</sup> (sin tener en cuenta el registro de estado y el incremento del PC)

<sup>2</sup> 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
<b>A</b> instrucción con operando registro	código de operación					registro destino (fuente en ST)			-	-	-	-	-	registro fuente (registro base en ST/LD)		
<b>B</b> instrucción con operando memoria o inmediato									dato inmediato / dirección del dato							
<b>C</b> instrucción de salto						condición de salto			dirección de salto							

I <sub>10</sub>	I <sub>9</sub>	I <sub>8</sub>	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

## ANEXO II: Información Atmega328pa

Para cada instrucción se presenta la siguiente información:

Mnemónico	Operandos	Descripción	Rango	Operación	Banderines	Ciclos de reloj
-----------	-----------	-------------	-------	-----------	------------	-----------------

### Instrucciones de transferencia de datos

<b>MOV</b>	Rd,Rr	Copiar registro	d,r∈ [0,31]	Rd←Rr	Ninguno	1
<b>MOVW</b>	Rd,Rr	Copiar registro W	d,r∈ [0,30]	Rd+1:Rd←Rr+1:Rr	Ninguno	1
<b>LDI</b>	Rd,k	Cargar dato inmediato	d∈ [16,31] k∈ [0,255]	Rd← k	Ninguno	1
<b>LDS</b>	Rd,k	Cargar dato desde la memoria	d∈ [0,31] k<64K	Rd← (k)	Ninguno	2
<b>LD</b>	Rd,X Rd,X+ Rd,-X Rd,Y Rd,Y+ Rd,-Y Rd,Z Rd,Z+ Rd,-Z	Carga el registro con un dato indirecto	d∈ [0,31]	Rd← (X) Rd← (X);X←X+1 X←X-1,Rd← (X) Rd← (Y) Rd← (Y);Y←Y+1 Y←Y-1;Rd← (Y) Rd← (Z) Rd← (Z);Z←Z+1 Z←Z-1;Rd← (Z)	Ninguno	2
<b>LDD</b>	Rd,Y+q Rd,Z+q	Carga el registro con un dato indirecto con desplazamiento	d∈ [0,31] q∈ [0,63]	Rd← (Y+q) Rd← (Z+q)	Ninguno	2

<b>STS</b>	K, Rr	Almacenar dato en memoria	r∈ [0,31] K∈ [0,64k]	(K)← Rr	Ninguno	2
<b>ST</b>	X,Rr X+,Rr -X,Rr Y,Rr Y+,Rr -Y,Rr Z,Rr Z+,Rr -Z,Rr	Almacenar registro en memoria	r∈ [0,31]	(X)← Rr (X)← Rr;X←X+1 X←X-1,(X)←Rr (Y)←Rr (Y)←Rr;Y←Y+1 Y←Y-1;(Y)←Rr (Z)←Rr (Z)←Rr;Z←Z+1 Z←Z-1;(Z)←Rr	Ninguno	2
<b>STD</b>	Y+q,Rr Z+q,Rr	Almacenar registro en memoria con indirecto con desplazamiento	r∈ [0,31]	(Y+q)←Rr (Z+q)←Rr	Ninguno	2
<b>LPM</b>	Rd,Z Rd,Z+	Carga memoria de programa		R0← (Z) Rd← (Z) Rd← (Z); Z←Z+1	Ninguno	3
<b>SPM</b>		Almacenar en memoria de programa		(Z) ← R1:R0	Ninguno	-

<b>IN</b>	Rd,P	Entrada del puerto	d∈ [0,31] P∈ [0,63]	Rd← P	Ninguno	1
<b>OUT</b>	P,Rr	Salida hacia el puerto	r∈ [0,31] P∈ [0,63]	P← Rr	Ninguno	1
<b>PUSH</b>	Rr	Empujar en pila	r∈ [0,31]	STACK←Rr	Ninguno	2
<b>POP</b>	Rd	Sacar de pila	d∈ [0,31]	Rd← STACK	Ninguno	2

### Instrucciones aritmético-lógicas

<b>ADD</b>	Rd,Rr	Suma sin carry	d,r∈ [0,31]	Rd← Rd+Rr	Z,N,V,C,H	1
<b>ADC</b>	Rd,Rr	Suma con carry	d,r∈ [0,31]	Rd← Rd+Rr+C	Z,N,V,C,H	1
<b>ADIW</b>	Rd,K	Suma inmediato con palabra	d∈ [24,26,28,30] K∈ [0,63]	Rd+1:Rd← Rd+1:Rd + K	Z,N,V,C	2
<b>SUB</b>	Rd,Rr	Resta sin carry	d,r∈ [0,31]	Rd←Rd-Rr	Z,N,V,C,H	1
<b>SUBI</b>	Rd,K	Resta inmediato	d∈ [16,31] K∈ [0,255]	Rd←Rd-K	Z,N,V,C,H	1
<b>SBC</b>	Rd,Rr	Resta con carry	d,r∈ [0,31]	Rd←Rd-Rr-C	Z,N,V,C,H	1
<b>SBCI</b>	Rd,K	Resta inmediato con carry	d∈ [16,31] K∈ [0,255]	Rd←Rd-K-C	Z,N,V,C,H	1
<b>SBIW</b>	Rd,K	Resta inmediato con palabra	d∈ [24,26,28,30] K∈ [0,63]	Rd+1:Rd← Rd+1:Rd - K	Z,N,V,C	2
<b>AND</b>	Rd,Rr	And lógica	d,r∈ [0,31]	Rd←Rd∧Rr	Z,N,V	1
<b>ANDI</b>	Rd,K	And lógica con dato inmediato	d∈ [16,31] K∈ [0,255]	Rd←Rd∧K	Z,N,V	1

<b>OR</b>	Rd,Rr	Or lógica	d,r∈ [0,31]	Rd←Rd∨Rr	Z,N,V	1
<b>ORI</b>	Rd,K	Or lógica con dato inmediato	d∈ [16,31] K∈ [0,255]	Rd←Rd∨K	Z,N,V	1
<b>EOR</b>	Rd,Rr	Exclusive or	d,r∈ [0,31]	Rd←Rd⊕Rr	Z,N,V	1
<b>COM</b>	Rd	Complemento a 1	d,r∈ [0,31]	Rd←\$FF-Rd	Z,N,V,C	1
<b>NEG</b>	Rd	Complemento a 2	d,r∈ [0,31]	Rd←\$00-Rd	Z,N,V,C	1
<b>INC</b>	Rd	Incrementa	d,r∈ [0,31]	Rd←Rd+1	Z,N,V	1
<b>DEC</b>	Rd	Decrementa	d,r∈ [0,31]	Rd←Rd-1	Z,N,V	1

<b>CLR</b>	Rd	Poner a cero	d,r∈ [0,31]	Rd←0	Z,N,V	1
<b>SER</b>	Rd	Poner todo a 1	d,r∈ [0,31]	Rd←\$FF	Z,N,V	1
<b>CP</b>	Rd,Rr	Compara	d,r∈ [0,31]	Rd-Rr	Z,N,V,C,H	1
<b>CPC</b>	Rd,Rr	Compara con carry	d,r∈ [0,31]	Rd-Rr-C	Z,N,V,C,H	1
<b>CPI</b>	Rd,K	Compara inmediato	d∈ [16,31] K∈ [0,255]	Rd-K	Z,N,V,C,H	1
<b>MUL</b>	Rd,Rr	Multiplica sin signo	d,r∈ [0,31]	R1:R0←Rd x Rr	Z,C	2
<b>MULS</b>	Rd,Rr	Multiplica con signo	d,r∈ [0,31]	R1:R0←Rd x Rr	Z,C	2
<b>MULSU</b>	Rd,Rr	Multiplica signo con sin signo	d,r∈ [0,31]	R1:R0←Rd x Rr (Rd signed Rr unsigned)	Z,C	2

### Instrucciones de salto

<b>RJMP</b>	Etiqueta	Salto relativo	-2k< Etiqueta <2k	PC←PC+ Etiqueta +1	Ninguno	2
<b>JMP (1)</b>	Etiqueta	Salto	0<Etiqueta<4M	PC← Etiqueta	Ninguno	2
<b>IJMP</b>		Salto indirecto		PC ← (Z)	Ninguno	3
<b>RCALL</b>	Etiqueta	Llamada a subrutina relativa	-2k< Etiqueta <2k	STACK ← PC PC←PC+ Etiqueta +1	Ninguno	3
<b>CALL (1)</b>	Etiqueta	Llamada a subrutina	0<Etiqueta<4M	STACK ← PC PC← Etiqueta	Ninguno	3
<b>ICALL</b>		Llamada a subrutina indirecta		STACK ← PC PC ← (Z)	Ninguno	4
<b>RET</b>		Regreso de subrutina		PC← STACK	Ninguno	4
<b>RETI</b>		Regreso de interrup.		PC← STACK	I	4
<b>CPSE</b>	Rd,Rr	Compara, esquivo si iguales	d,r∈ [0,31]	Si Rd=Rr PC← PC+2 (ó 3)	Ninguno	01/02/03
<b>SBRC</b>	Rr,b	Esquiva si el bit está a cero	r∈ [0,31] b∈ [0,7]	Si (Rd(b)=0) PC← PC+2 (ó 3)	Ninguno	01/02/03
<b>SBRS</b>	Rr,b	Esquiva si el bit está a uno	r∈ [0,31] b∈ [0,7]	Si (Rd(b)=1) PC← PC+2 (ó 3)	Ninguno	01/02/03

<b>SBIC</b>	P,b	Esquiva si el bit del puerto está a 0	P∈ [0,31] b∈ [0,7]	Si (I/O(P,b)=0) PC← PC+2 (ó 3)	Ninguno	1 o 2 o 3
<b>SBIS</b>	P,b	Esquiva si el bit del puerto está a 1	P∈ [0,31] b∈ [0,7]	Si (I/O(P,b)=1) PC← PC+2 (ó 3)	Ninguno	1 o 2 o 3
<b>BREQ</b>	Etiqueta	Salta si iguales	Etiqueta ∈ [-64,63]	Si (Z=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRNE</b>	Etiqueta	Salta si distintos	Etiqueta ∈ [-64,63]	Si (Z=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRCS</b>	Etiqueta	Salta si C está a 1	Etiqueta ∈ [-64,63]	Si (C=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRCC</b>	Etiqueta	Salta si C está a 0	Etiqueta ∈ [-64,63]	Si (C=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRSH</b>	Etiqueta	Salta si igual o mayor	Etiqueta ∈ [-64,63]	Si (C=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2

<b>BRLO</b>	Etiqueta	Salta si menor	Etiqueta ∈ [-64,63]	Si (C=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRMI</b>	Etiqueta	Salta si negativo	Etiqueta ∈ [-64,63]	Si (N=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRPL</b>	Etiqueta	Salta si positivo	Etiqueta ∈ [-64,63]	Si (N=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRHS</b>	Etiqueta	Salta si H está a 1	Etiqueta ∈ [-64,63]	Si (H=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRHC</b>	Etiqueta	Salta si H está a 0	Etiqueta ∈ [-64,63]	Si (H=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRTS</b>	Etiqueta	Salta si T está a 1	Etiqueta ∈ [-64,63]	Si (T=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRTC</b>	Etiqueta	Salta si T está a 0	Etiqueta ∈ [-64,63]	Si (T=0) PC← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRVS</b>	Etiqueta	Salta si V está a 1	Etiqueta ∈ [-64,63]	Si (V=1) PC← PC+ Etiqueta +1	Ninguno	1 o 2

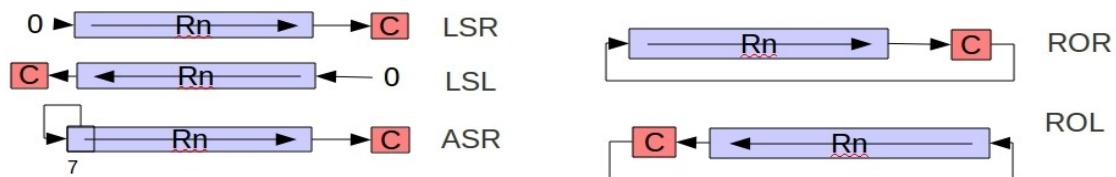
<b>BRVC</b>	Etiqueta	Salta si V está a 0	Etiqueta ∈ [-64,63]	Si (V=0) PC ← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRIE</b>	Etiqueta	Salta si I está a 1	Etiqueta ∈ [-64,63]	Si (I=1) PC ← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRID</b>	Etiqueta	Salta si I está a 0	Etiqueta ∈ [-64,63]	Si (I=0) PC ← PC+ Etiqueta +1	Ninguno	1 o 2

<b>BRGE</b>	Etiqueta	Salta si mayor o igual, (signo)	Etiqueta ∈ [-64,63]	Si (N⊕V=0) PC ← PC+ Etiqueta +1	Ninguno	1 o 2
<b>BRLT</b>	Etiqueta	Salta si menor (signo)	Etiqueta ∈ [-64,63]	Si (N⊕V=1) PC ← PC+ Etiqueta +1	Ninguno	1 o 2

Test (CP Rd,Rr)	Booleana	Mnemonico	Comentario
Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Signo
Rd < Rr	(N ⊕ V) = 1	BRLT	Signo
Rd = Rr	Z = 1	BREQ	Signo/Sin signo
Rd ≠ Rr	Z = 0	BRNE	Signo/Sin signo
Rd ≥ Rr	C = 0	BRCC/BRSH	Sin signo
Rd < Rr	C = 1	BRCS/BRLO	Sin signo
Carry	C=1	BRCS	Simple
Sin carry	C=0	BRCC	Simple
Negativo	N=1	BRMI	Simple
Positivo	N=0	BRPL	Simple
Overflow	V=1	BRVS	Simple
Sin overflow	V=0	BRVC	Simple
Cero	Z=1	BREQ	Simple
No cero	Z=0	BRNE	Simple

### Instrucciones de bit y de bit test

<b>LSL</b>	Rd	Desplazamiento a la izquierda	d ∈ [0,31]	Rd(n+1) ← Rd(n), Rd(0) ← 0, C ← Rd(7)	Z,C,N,V,H	1
<b>LSR</b>	Rd	Desplazamiento a la derecha	d ∈ [0,31]	Rd(n) ← Rd(n+1), Rd(7) ← 0, C ← Rd(0)	Z,C,N,V	1
<b>ROL</b>	Rd	Rotación a la izquierda	d ∈ [0,31]	Rd(n+1) ← Rd(n), Rd(0) ← C, C ← Rd(7)	Z,C,N,V,H	1
<b>ROR</b>	Rd	Rotación a la derecha	d ∈ [0,31]	Rd(n) ← Rd(n+1), Rd(7) ← C, C ← Rd(0)	Z,C,N,V,	1
<b>ASR</b>	Rd	Desplazamiento aritmético a la derecha	d ∈ [0,31]	Rd(n) ← Rd(n+1), Rd(7) ← Rd(7), C ← Rd(0)	Z,C,N,V,	1



<b>SWAP</b>	Rd	Intercambia nibbles	d ∈ [0,31]	Rd(3..0) ← Rd(7.4)	Ninguno	1
<b>SBI</b>	P,b	Poner a 1 el bit b del puerto IO	b ∈ [0,7] P ∈ [0,31]	IO(P,b) ← 1	Ninguno	2
<b>CBI</b>	P,b	Poner a 0 el bit b del puerto IO	b ∈ [0,7] P ∈ [0,31]	IO(P,b) ← 0	Ninguno	2
<b>SEcc</b>		Poner a 1 el bit cc del registro de estado			cc	1
<b>CLcc</b>		Poner a 0 el bit cc del registro de estado			cc	1

cc= C,N,T,Z,I,V,H,S

### Instrucciones de control

<b>NOP</b>		Nada			Ninguno	1
<b>BREAK</b>		Para depuración			Ninguno	N/A
<b>WDR</b>		Reinicia el temporizador del perro guardián			Ninguno	1
<b>SLEEP</b>		Dormir			Ninguno	1