Exercicis Processador EduP12

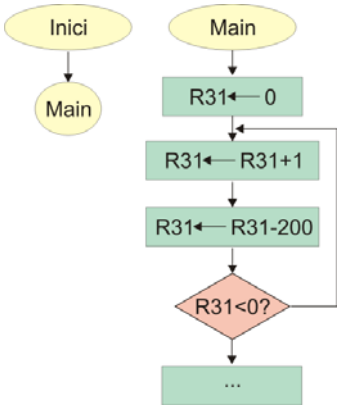
Exercici resolt

La figura 1 correspon a un determinat programa que realitza una rutina de retard. Agafant aquest programa com a base es demana (entre parèntesis s'ha posat, per clarificar l'exercici, la posició de memòria en la que es s'inicia cada tros de programa):

i) Donar un diagrama de flux (en format RTL) del programa.
ii) Trobar el codi màquina del programa. Ajudeu-vos del repertori d'instruccions d'EduP12.
iii) Trobar els cicles de rellotge en els que es descomposa la instrucció SUBI. Donar l'activitat dels senyals de control que s'activen durant la seva execució.
iv) Calcular el retard (nombre de cicles de rellotge) de la subrutina de retard suposant que cada cicle dura exactament un cicle del rellotge base i que la freqüència del rellotge és de 20 MHz.

```
(0)              RJMP main
(32) main:       LDI R31, 0
(33) bucle:      INC R31
(34)             SUBI R31, 200
(35)             BRMI bucle
(36)             ...
```

a)



b)

Figura 1

Apartat i).

La figura 1b mostra el diagrama de flux. S'observa que el programa escrit mostra un salt relatiu inicial abans de passar a executar una altra part. En capítols posteriors s'explica el motiu d'aquest salt.

Apartat ii).

El codi màquina de les instruccions es troba de forma directa analitzant el format de cada instrucció donat en el repertori d'instruccions.

Programa			Format instrucció	Codi màquina	
(0)		RJMP main	0100 kkkkkkkkk	(0)	0100000000100000
(32)	main:	LDI R31, 0	0011 --- ddddd 0000	(32)	0011000111110000
			0	(33)	0000000000000000
	bucle:	INC R31	0011 --- ddddd 1000	(33)	0011000111111000
		SUBI R31, 200	0011 --- ddddd 0100	(33)	0011000111110100
			200	(33)	0000000011001000
		BRMI bucle	1110 kkkkkkkkk 010	(34)	1110111111110010
		...	...	(35)	...

La transformació de totes les instruccions és directa. Com a càlcul a realitzar només s'ha de trobar el salt realtiu des de la instrucció BRMI, que és simple: donat que es troba en la posició 34, que quan s'executa el PC ja ha incrementat (i val PC = 35), i que el salt a fer és a la instrucció 33, el salt relatiu és de -2. Cal recordar que s'ha de posar en complement a la base.

Apartat iii).

- Cicles de rellotge de la instrucció SUBI i activació de senyals de la unitat de procés per a la seva execució.

Cicle	Transferència de dades (en RTL)	Activació de senyals de control
Cicle 1	MAR ← PC	zpMAR←0, ldpMAR←1
Cicle 2	IR ← mem<MAR>, PC ← PC + 1 (Descodificació)	ldIR←1, zA←2, aluOp←INC, zPC←0, ldPC←1 (addRd=17)
Cicle 3	MAR ← PC, PC ← PC + 1	zpMAR←0, ldpMAR←1, zA←2, aluOp←INC, zPC←0, ldPC←1
Cicle 4	R31 ← R31 – mem<MAR>	zA←3, zB←1, aluOp←SUB, zRegs←0, wrRegs←1

Apartat iv).

El càlcul de la duració de la rutina d'espera es troba determinant el nombre de cicles totals de rellotge que triga en executar el bucle. Es troba fàcilment que:

Instrucció	**#cicles**	**#cops que s'executa**	**Total (cicles)**
INC	3	200	600
SUBI	4	200	800
BRMI	3	200	600
		Total	2000

Per tant, la duració total del bucle és de 2000 cicles, que tenint present una freqüència de rellotge de 20MHz implica que triga en executar 2000 / ($2*10^7$) = 100µseg

Exercicis

1. Trobar el codi màquina del conjunt d'instruccions següent:

a. MOV R0, R1
b. CLR R16
c. ANDI Z, 0x88
d. BRID 0x444
e. LPM +X, R16
f. ST +X, R16
g. LDD R24, Y+50

2. Trobar els cicles de rellotge en què es descomposen el conjunt d'instruccions de l'exercici 1.

3. Es té el següent programa (entre parèntesis s'ha posat l'adreça d'inici de cada tros de programa).

```
    ( Suposar que R31 agafa inicialment el valor 1000 )
(32)        main:   RCALL b1
                    RJMP main
                ...
(100)       b1:     RCALL b2
                    RET
(200)       b2:     INC R31
                    RET
```

Es demana:

i) Donar un diagrama de flux del seu funcionament.
ii) Donar un diagrama d'evolució temporal del processador en el que es vegi l'evolució dels registres de la CPU involucrats en l'execució del programa.

1. Donat el següent programa determinar:
   i) El codi màquina del programa.
   ii) Determinar la duració de la rutina d'espera .

```
espera: LDI R0, 0xFF
b0: LDI R1, 0xFF
b1: DEC R1
    BRCC b1
    DEC R0
    BRCC b0
    RET
```

Nota: Entre parèntesis s'ha posat l'adreça d'inici de cada tros de programa.

5 Exercicis llenguatge màquina

1. Donats els següents programes, donar el valor del registre d'estat durant l'execució del programa.

.con tot = 0xfff;
.con res = 0x000;
.def a = r31;
.def b = r30;
 	
ldi a, res;
ldi b, res;
tst a, b;
and a, b;
sub a, b;
	.con tot = 0xfff;
	.con res = 0x000;
	.def a = r31;
	.def b = r30;
	
	ldi a, tot;
	ldi b, tot;
	tst a, b;
	and a, b;
	sub a, b;
		.con tot = 0xfff;
		.con res = 0x000;
		.def a = r31;
		.def b = r30;
		
		ldi a, tot;
		ldi b, res;
		tst a, b;
		and a, b;
		sub a, b;
			.con tot = 0xfff;
			.con res = 0x000;
			.def a = r31;
			.def b = r30;
			
			ldi a, res;
			ldi b, tot;
			tst a, b;
			and a, b;
			sub a, b;

2. Donats els següents dos programes es demana:

i) Donar el diagrama de flux del programa.
ii) Trobar el codi màquina del programa.
iii) Per a cada instrucció, indicar quin mode d'adreçament s'empra.

El nombre entre parèntesis al començament s'una instrucció indica la posició en la que comença aquella instrucció.

PROGRAMA 1. Realitzar la suma aritmètica dels 50 primers nombres i donar el valor pel PORTB[1]. En aquests programes és important entendre bé els límits dels índexs per acabar just en el nombre límit que es demana.

```
-- SUMA ARITMÈTICA
(0)     LDI R2,
        0
        MOV R1, R2
loop:   INC R1
        ADD R2, R1
        CPI R1, 50
        BRMI loop (-5)
        OUT PORTB, R2
fi:     RJMP fi (-1)
        NOP
```

PROGRAMA 2: Càlcul del sinus per detecció de valor en taula

```
--Càlcul tabular del sinus
-- La taula del sinus es guarda a partir de la posició 1000, i salta de 10º en 10º
-- Per evitar fraccionaris el valor del sinus es multiplica per 100 i s'agafa l'enter més pròxim
-- Nota: Per simplicitat es mostren els primers i darrers valors de la taula. Acabeu d'omplir-la.
        LDI Z, 0x999
loop:   CPI Z, 1035
```

[1] La instrucció IN, amb format *IN Registre, PORTA* és una instrucció d'entrada que permet agafar dades d'un port extern a la CPU. Respectivament, La instrucció OUT, amb format *OUT PORTB, Registre* és una instrucció de sortida que envia dades cap a un port extern a la CPU. En aquests exemples es fan servir com a ports d'entrada i sortida de dades. En el capítol 8 s'introdueix de manera més formal el treball amb ports d'entrada/sortida.

```
                BRPL fi (+3)
                LPM R0, +Z
                OUT PORTB, R0
                RJMP loop (-6)
fi:             RJMP fi (-1)
                NOP

(1000)          "0000000000000000"          sin(0)
                "0000000000010001"          sin(10)
                "0000000000100010"          sin(20)
                ...
                “1111111111101111"          sin(350)
```

3. Explicar si és coherent i què fa el següent programa:

```
--Programa exemple amb ijmp
.con inici = 10;
.con fi = 20;

.org 0;
    ldi x, inici;
    ldi z, fi;
.org inici;
    out PORTB, x;
    ijmp z;

.org fi;
    out PORTB, z;
    ijmp x;
```

4. Pel PortA s’entren 10 nombres. Treure pel PortB el nombre major.

5. Realitzar un programa que doni la suma dels 25 primers múltiples de 3. Donar el resultat pel PortB.

6. Donat un nombre de 12 bits (entrat pel PortA) treure 1 pel PortB si és múltiple de 100, i si no treure 0.

7. (Problema una mica més llarg). Donat dos nombres A i B de 12 bits (es poden entrar com a immediats en el programa), realitzar l’operació la seva multiplicació i guardar el resultat en els registres R1:R0.