

# Exercicis a paper Tema 11

## Edgar Aguadé Nadal - Grup 62

S'ha d'entregar un únic fitxer PDF que inclogui la solució que vosaltres proposeu als problemes plantejats. El fitxer PDF no te que ser necessàriament una solució feta per ordinador, pot ser una solució escrita a ma i digitalitzada. El PDF ha d'incloure una capçalera on s'indiqui el vostre nom i cognoms, i l'enunciat de cada pregunta abans de la vostra resposta.

### Exercici 1

Dado el sistema formado por la UPG, el subsistema de entrada/salida (IO) con teclado e impresora (con efecto lateral) y memoria de datos, escribid un código ensamblador SISA que lea un dato (que se interpreta como un número natural) del teclado, lo divida entre 2 elevado a  $Y < 4$  (siendo  $Y < 4$  el valor de los 5 bits de menor peso de la posición de memoria cuya dirección esta almacenada en el registro R7, interpretados en complemento a dos) y lo escriba en la impresora. Debe realizar este proceso indefinidamente. El registro (puerto) de datos del controlador del teclado tiene la dirección 0 y el de estado la dirección 1, ambas del espacio de entrada.

```
// Supposem STATUS-IMPRESORA: PORT 3, DATA-IMPRESORA: PORT 2

IN R0,1
BZ R0,-2
IN R1,0

LD R2,0(R7)
NOT R2,R2
ADDI R2,R2,1
SHL R1,R1,R2

IN R0,3
BZ R0,-2
OUT 2,R1
BNZ R0,-11
```

### Exercici 2

Dado el sistema formado por la UPG, el subsistema de entrada/salida (IO) con teclado e impresora (con efecto lateral) y memoria de datos, escribid un código ensamblador SISA que invierta las posiciones de un vector de 100 elementos de tipo byte. El vector A[0..99] está almacenado a partir de la dirección 0x2000 de la memoria de datos. El elemento A[0] está en la posición 0x2000, el A[1] está en la posición 0x2001, etc. Después de ejecutar la el programa el orden de los elementos debe estar invertido: A[99] deberá estar en la posición 0x2000, el A[98] deberá estar en la posición 0x2001, etc.

```
MOVI R0,0x00
MOVHI R0,0x20
ADDI R1,R0,99
MOVI R7,50

LDB R2,0(R0)
LDB R3,0(R1)
STB 0(R0),R3
STB 0(R1),R2

ADDI R0,1
ADDI R1,-1
ADDI R7,-1
BNZ R7,-8
```

## Exercici 3

Dado el sistema formado por la UPG, el subsistema de entrada/salida (IO) con teclado e impresora (con efecto lateral) y memoria de datos, escribid un código ensamblador SISA que lea 50 datos (que se interpretan como números enteros en Ca2) de un teclado y que calcule cuantas operaciones de resta darían overflow si a cada dato le restásemos el valor 25. El resultado de esta cuenta debe dejarse en el registro R4. Los puertos de datos y de estado de los controladores del teclado y la impresora son los que figuran en el esquema de la entrada/salida de la documentación. En el teclado e impresora el acceso al puerto de datos tiene un efecto lateral sobre su registro de estado

```
// Supposem STATUS-TECLAT: PORT 0, DATA-TECLAT: PORT 1
// Supposem STATUS-IMPRESORA: PORT 2, DATA-IMPRESORA: PORT 3

MOVI R7,50
MOVI R4,0
MOVI R6,0x00
MOVHI R6 0x80

IN R0,0
BZ R0,-2
IN R1,1

AND R2,R1,R6
BZ R2,4

ADDI R3,R1,-25
AND R3,R3,R6
BNZ R3,2

ADDI R4,R4,1

ADDI R7,R7,-1
BZ R7,-15

IN R0,2
BZ R0,-2
OUT 3,R4
```

## Exercici 4

Dado el sistema formado por la UPG, el subsistema de entrada/salida (IO) con teclado e impresora (con efecto lateral) y memoria de datos, escribid un código ensamblador SISA que convierta un vector de 100 elementos de tipo word almacenados en formato littleendian a un formato big-endian. El vector A[0..99] está almacenado a partir de la dirección 0x3000 de la memoria de datos.

```
MOVI R0,0x00
MOVHI R0,30
MOVI R7,100

LDB R1,0(R0)
LDB R2,1(R0)
STB 1(R0),R1
STB 0(R0),R2

ADDI R0,2
ADDI R7,-1
BNZ R7,-7
```