

Exercicis Tema 14

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 mà 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

Especificad el **camino crítico** (indicando la suma ordenada de los tiempos de propagación de los bloques por los que pasa) y calculad el **tiempo de ciclo mínimo** para que el computador **SISC Von Neumann** pueda ejecutar correctamente el tipo de instrucción SISA que se indica en cada apartado (este sería el tiempo de ciclo mínimo del computador si solo ejecutara instrucciones como la indicada u otras que requieran menor tiempo). No tenéis que añadir ningún porcentaje de seguridad en el cálculo del tiempo de ciclo mínimo. Suponed que los tiempos de propagación de los bloques que forman el computador son los siguientes:

$T_p(\text{ROM_Q+}) = 70 \text{ u.t.}$

$T_p(\text{ROM_OUT}) = 90 \text{ u.t.}$

$T_p(\text{MUX-2-1}) = 60 \text{ u.t.}$

$T_p(\text{MUX-4-1}) = 110 \text{ u.t.}$

$T_p(\text{REG}) = 100 \text{ u.t.}$ // Tiempo de propagación de un registro.

$T_p(\text{REGFILE}) = 200 \text{ u.t.}$ // Tiempo de lectura del banco de registros

$T_p(\text{ALU-slow}) = 900 \text{ u.t.}$ // Tp de la ALU para las operaciones/funciones lentas: ADD, SUB, CMP* .

$T_p(\text{ALU-quick}) = 450 \text{ u.t.}$ // Tp de la ALU para las operaciones/funciones rápidas: cualquier otra distinta de ADD, SUB, CMP* .

$T_{\text{acc}}(\text{MEMORY}) = 1100 \text{ u.t.}$ // Tiempo de acceso (para la lectura o escritura) a la memoria

$T_p(\text{AND-2}) = T_p(\text{OR-2}) = 20 \text{ u.t.}$

$T_p(\text{NOT}) = 10 \text{ u.t.}$

El tiempo de propagación de un bloque combinacional (T_p) y el tiempo de acceso a memoria para realizar una lectura (T_{acc}) es el tiempo desde que están estables todas las entradas necesarias hasta que se estabilizan las salidas requeridas al valor correcto para las entradas aplicadas. Desconocemos como se han implementado internamente los bloques (y podría ser de forma diferente a los vistos en clase). Recordad que un registro con señal de carga (Ld), REGwLd, está construido con un REG y un MUX-2-1 (no os damos el esquema interno del REGwLd, porque lo tenéis que saber).

- T_c correspondiente al nodo de **D** (decode). *1260 ut*
- T_c correspondiente al nodo de **Addi**. *1260 ut*
- T_c correspondiente al nodo de **Ldb**. *1320 ut*
- T_c correspondiente al nodo de **Bnz**. *790 ut*

Exercici 2

El fragmento de programa en ensamblador SISA que se muestra a continuación se ha traducido a lenguaje máquina situando la sección de datos (.data) a partir de la dirección $0x0100$ y la sección de código (.text) a partir de la dirección $0xC000$. Antes de la primera instrucción que se muestra (MOVI R5, 0) hay 150 instrucciones que no se muestran y después de la última instrucción que se muestra (ST 0(R2), R5) hay 20 instrucciones que tampoco se muestran.

```
.data 0x0100
long=6
A: .word 1, -4, 15, 0, 12
B: .word 13, 2, 14, -2, 9
C: .space A
D: .space 1
.text 0xC000
...
MOVI R5, 0
MOVI R1, lo(A)
```

Handwritten annotations in blue:

- Addresses for .data: 0x0100, 0x0102, 0x0104, 0x0106, 0x0108, 0x010A, 0x010C, 0x010E, 0x0110, 0x0112, 0x011D
- Address for .text: 0xC000
- Address for MOVI R5, 0: 0xC000
- Address for MOVI R1, lo(A): 0xC002