

Nom i cognoms:.....

PROVA DE PROBLEMES

Temps: 1h 15m. Obtenint 3 o més punts sobre 10, la prova puntuarà el 50% del total de l'examen. La puntuació està posada al costat de cada exercici. La revisió de l'examen es farà el dia 14 a l'hora de classe (15:00h).

1.- MÀQUINA SENZILLA – SOFTWARE: La sèrie de Fibonacci. (5 Punts)

a.- Escriu un programa en ensamblador de la Màquina Senzilla que calculi l'*n*-èssim valor de la sèrie de *Fibonacci*. (4 Punts).

Recorda que la sèrie de *Fibonacci* comença per 0 i 1 i el següent element amb posició *n* s'obté a partir de la suma dels dos elements de les posicions anteriors *n - 1* i *n - 2*. Una definició formal de la sèrie pot ser la següent:

$$Fib(n) = \begin{cases} n & n \geq 0 \wedge n \leq 1 \\ Fib(n-1) + Fib(n-2) & \forall n > 1 \end{cases}$$

Per exemple, els primers termes de la sèrie són:

<i>n</i>	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
<i>Fib(n)</i>	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	610	987	1597	2584	4181	6765	...

NOTA: Per implementar el programa, es poden utilitzar, de forma orientativa, les següents dades:

<i>Etiqueta</i>	@	<i>Contingut</i>
zero:	100	0
un:	101	1
n:	102	<i>n</i>
i:	103	X
Fib0:	104	X
Fib1:	105	X
resultat:	106	X

També pot ser d'ajuda el següent algorisme iteratiu en *pseudocodi* que soluciona el problema:

```

i := 0; // Inicialitzar comptador
resultat := 0; // Calcular Fib(0)
Fib0 := 1; // Calcular Fib(1)
Mentre i <> n Fer // Mentre no s'arribi a l'n-èssim
    Fib1 := resultat + Fib0; // Calcular el següent element
    Fib0 := resultat; // Guardar l'anterior
    resultat := Fib1; // A resultat posar-hi el següent
    i := i + 1; // Incrementar el comptador
F Mentre
    
```

b.- Si s'interpreten sense signe els valors emmagatzemats a la memòria de la MS1, quin és el màxim valor d'*n* perquè no hi hagi cap desbordament (*overflow*) i, per tant, el resultat de l'execució del programa implementat sigui correcte? (1 Punt)

2.- MÀQUINA SENZILLA – HARDWARE: Comparació amb Prefech. (5 Punts)

Es vol modificar la versió de la Màquina Senzilla original amb l'autòmat d'estats de la Unitat de Control sense simplificar per tal d'optimitzar una mica (en temps) l'execució dels programes.

Durant la fase d'execució de la instrucció **CMP** (estat S_9 a les transparències de classe) **no** s'està utilitzant la memòria. Donat que després de l'execució d'un **CMP** sempre cal executar la instrucció de l'adreça següent (que està apuntada pel PC), es pot avançar el Fetch de la següent instrucció i fer-lo a l'estat d'execució del **CMP**.

Es demana:

- a.- Definir quines són **les accions** que s'han de dur a terme a cada una de les Fases (*Fetch*, Descodificació, Càrrega d'Operands i Execució) de la nova versió de la instrucció **CMP**. **(1.5 Punts)**
- b.- Modificar, si cal, el format d'instrucció. **(0.25 Punts)**
- c.- Modificar la Unitat de Procés, si cal, per executar la nova versió de **CMP**. **(0.25 Punts)**
- d.- Modificar l'autòmat d'estats de la Unitat de Control, si cal, per executar la nova versió de **CMP**. **(1.25 Punts)**
- e.- Definir el vector de sortides de la Unitat de Control pels nous estats que hagin aparegut i redefinir el vector pels estats que hagin canviat. **(0.75 Punts)**
- f.- La modificació realitzada estalvia cicles de rellotge en l'execució d'un sol **CMP**? En cas afirmatiu, quants? I en l'execució d'un **CMP** al mig d'altres instruccions? En cas afirmatiu, quants?. **(1 Punt)**

NOTA: Cal raonar adequadament cada una de les respostes.