Examen final 10 Gener 2005 2 hores

*Es consideraran vàlides les respostes que siguin justificades.*

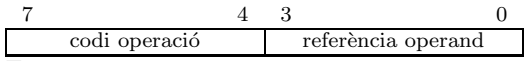
1. Es té un processador, anomenat Bit2005, amb les següents especificacions:

Registres:

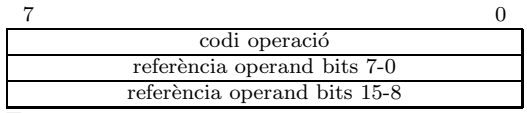
| Nom | No. de bits | Funció |
|---|---|---|
| RI | 8 | Registre d'instrucció |
| PC | 16 | Comptador de programa |
| SP | 16 | Apuntador de pila |
| A | 8 | Acumulador |
| B | 8 | Dades diverses |

Tres formats d'instrucció:

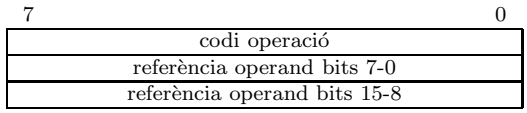
Format 1:




Format 2:




Format 3:




Instruccions (format →codi operació →efecte):

| **Format** | **Codi** | | **Efecte** |
|---|---|---|---|
| 3 | 1 | → | Carrega a l'acumulador l'operand adreçat pels bits 15-0 ($A := Memoria[operand]$) |
| 3 | 2 | → | Escriu contingut de l'acumulador a l'adreça donada pels bits 15-0 ($Memoria[operand] := A$) |
| 2 | 3 | → | Inicialitza l'acumulador en el valor del operand ($A := operand$) |
| 3 | 5 | → | Carrega al registre B l'operand adreçat pels bits 15-0 ($B := Memoria[operand]$) |
| 3 | 6 | → | Escriu contingut del registre B a l'adreça donada pels bits 15-0 ($Memoria[operand] := B$) |
| 3 | 7 | → | Suma a l'acumulador el contingut de l'adreça que hi ha al camp operand. ($A = A + Memoria[operand]$). |
| 1 | 9 | → | Decrementa en 1 el registre referenciat en l'operand (Si l'operand és 0, $A := A - 1$, si és 1, $B := B - 1$). |
| 1 | A | → | Empila (push) el contingut adreçat per l'operand de la instrucció (0 →A, 1 →PC, 2 →B). ($A0 \equiv SP := SP - 1$; $Memoria[SP] := A$) |
| 1 | B | → | Desempila (pop) el contingut adreçat per el SP i el posa en el registre indicat per l'operand de la instrucció (0 →A, 1 →PC, 2 →B). ($B1 \equiv PC := Memoria[SP]$; $SP := SP + 1$) |
| 3 | C | → | Salta a l'adreça indicada pels bits 15-0. ($PC := operand$) |
| 3 | D | → | Si es troba que el contingut del registre $B$ és zero, salta a l'adreça indicada pels bits 15-0. (**si** $B = 0$ →$PC := operand$ **fsi**). En cas contrari, no fa res |

L'estat inicial del processador, el registre $PC$ està a $102D$ i $SP$ a $FFFF$.

a) Indiqueu els paràmetres (capacitat màxima, longitut de paraula) de memòria principal adients al processador Bit2005.

b) A partir de la posició de memòria $102D$ hi ha el següent programa en hexadecimal: 03 00 05 01 10 $0D$ $3C$ 10 07 00 10 91 $0C$ 32 10 02 02 10 i en les adreces 1000, 1001 i 1002 contenen respectivament 5, 2 i 10. Feu i descriviu el seguiment del programa i determineu l'estat de registres del Bit2005 i de les posicions 1000, 1001 i 1002 quan s'ha executat la instrucció que hi ha a a la posició de memòria $103C$. Raoneu com s'ha arribat a aquest estat.

c) Quan s'està executant per primer cop la instrucció que hi ha a la $102F$ de l'anterior programa, arriba una interrupció. El servei d'interrupció comença a l'adreça 20, a partir de la qual, es troba el següent contingut: $A0, 91, 91, B0, B1, B2$. Com afecta la interrupció a l'execució del programa de l'apartat $b$. Raoneu la resposta.

Entre una memòria de 64K paraules i Bit2005, tenim una caché associativa per conjunts de 2 conjunts de 2 linies de 4 paraules cadascuna. Es demana que

d) Dissenyeu el format de l'adreça per gestionar la caché.

e) Sense tenir en compte la interrupció de l'apartat 1.c, calculeu el percentatge de fallides que tindrà l'execució d'un tros del programa de l'apartat 1.b, sabent que la caché aplica la substitució LRU. El tros del programa a exaeminar és el que va des de l'inici a $102D$ fins que torna a la instrucció 1032 per segon cop (Aquesta darrera no cal incloure-la).

**MÉS EXERCICIS AL DORS**

2. Es tenen els següents programes corresponents a quatre processos residents en la cua de preparats en ordre d'arribada A, B, C i D.

A:
```
var T: taula[1..10, 1..10] de enter
i, j: enter;

per i := 1 fins 10 fer
   per j := 1 fins 10 fer
      T[i, j] := 0
   fper
fper
```

B:
```
var T: taula[1..10, 1..10] de enter
i, j: enter;

per j := 1 fins 10 fer
   per i := 1 fins 10 fer
      T[i, j] := abs(T[i, j]) − 10 ∗ T[j, i]
   fper
fper
```

C:
```
var T: taula[1..10, 1..10] de enter
i, j: enter; f: FST

per j := 1 fins 10 fer
   per i := 1 fins 10 fer
      T[i, j] := T[i, j]/8 + 6 ∗ T[i, j]
      escriureEnterFST(f,T[i, j])
   fper
fper
```

D:
```
var T: taula[1..10, 1..10] de enter
i, j: enter;f: FST

per i := 1 fins 10 fer
   per j := 1 fins 10 fer
      llegirEnterFST(f,T[i, j])
      T[i, j] := 4 ∗ T[i, j] + 3
   fper
fper
```

Es té que el cost del cos central de les iteracions pel procés $A$ és 1 unitat de temps, 2 unitats pel procés $B$, 14 pel procés $C$, i 7 pel procés $D$. Pel cas dels processos $C$ i $D$ el cost donat no té en compte el temps d'accés al disc. El cost de l'increment i inicialització de les variables d'iteració és zero (despreciable). D'acord als costos donats,

a) Calculeu el temps de tongada de cada procés i il·lustreu l'execució dels processos en els casos en que el planificador usat és Primer en Entrar Primer en ser Servit (FCFS), torn rotatori (Round Robin, quantum=6) i Primer el Treball Més Curt (SJF).

b) Calculeu els temps de retorn i temps d'espera de cada procés per cada algorisme de planificació esmentat abans.

Els processos s'executaran en un sistema de memòria paginada de marcs de 10 paraules. La mida d'un enter és d'una paraula. El codi de cada procés ocupa una pàgina. El sistema operatiu assigna 3 pàgines per cada procés i aplica l'algorisme de substitució de pàgines LRU. La taula es guarda a memoria seguint l'ordre $T[1,1], T[1,2], \ldots, T[1,N], T[2,1], T[2,2], \ldots$ El compilador ha decidit a més que les variables $i$ i $j$ siguin registres del processador i per tant no fan cap accés a memòria.

c) Determineu el nombre de fallides de pàgina pel programa A i pel programa B.

d) Si s'augmentés el nombre de pàgines per programa, s'obtindria alguna millora en algun cas pels programes A i B?.

e) Tenint en compte el sistema de memòria paginada esmentat i i amb la hipòtesi de que les pàgines no actives dels programes estan en la memòria d'intercanvi, creieu que cal fer canvis en l'apartat 2.$a$?. En cas afirmatiu, indiqueu els canvis per tornar a plantejar l'apartat 2.$a$.