

Cognoms:	NIU:
Nom:	Grup:

Responen les preguntes de la prova en l'espai habilitat, no s'acceptaran fulls apart.

L'examen es fa sobre 20 punts.

Llegiu l'enunciat de cada pregunta amb cura i contesteu només el que es pregunta, però justifiqueu sempre la vostra resposta.

Teoria 1. (2,5 punts)

Responen breument les següents preguntes:

- a) Què és Registre Comptador de Programa (PC)?

Registre que conté l'adreça de la següent instrucció a ser executada.

- b) Què és un Mode d'adreçament (enumerar-los **no** és una resposta)

És la funció que genera l'adreça efectiva de la posició de memòria on és l'operand d'una instrucció a partir de l'adreça explícita que apareix a la instrucció.

- c) Què és el Registre d'Estat (SR)?

SR Registre que agrupa bits de caràcter lògic que informen de l'estat de diferents parts de la CPU en particular de la ALU. Bits típics són els de Carry, overflow, signe, etc.

- d) Què vol dir que una Memòria sigui Associativa?

És una memòria que permet accedir la informació que hi ha emmagatzemada en cada posició a partir de buscar una porció del propi contingut de la posició.

- e) Què són el mode usuari i el mode supervisor o sistema?

Estats en que pot operar una CPU amb diferents capacitats. El mode usuari està pensat per executar els programes dels usuaris i el mode supervisor per executar el sistema operatiu.

En mode usuari no es pot accedir a tot el mapa de memòria, ni es pot executar qualsevol instrucció del repertori, n'hi ha de privilegiades, no es pot accedir als ports d'entrada/sortida i no es té accés a determinats registres de la CPU.

En mode supervisor no hi ha cap de les limitacions esmentades en el mode usuari.

Teoria 2. (2,5 punts)

Apliqueu els conceptes estudiats al curs per respondre les preguntes següents:

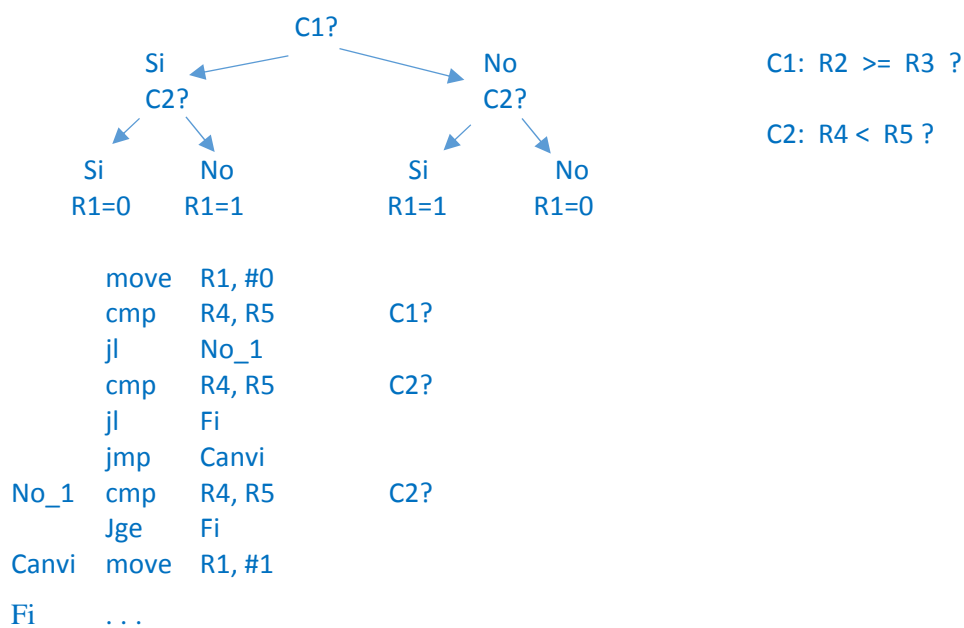
- a) Quin és el nombre màxim (en Megas) de cel·les (posicions) que pot tenir la memòria principal si el bus d'adreces té 24 línies (bits)? **(0,5 punts)**

$$2^{24} = 16 \text{ Megues}$$

- b) Quin és l'efecte d'executar el següent programa? (El PC = 50 inicialment) **(1 punt)** Descriure el programa línia a línia serà considerat incorrecte. Volem l'efecte general. Calcula alguna cosa? Quan acaba? Com canvia la pila?

<pre> 50 mov AX, #25 ; 51 shl AX ; 52 call 1000 ; 53 mov \$100, AX ; 1000 push AX 1001 ret </pre>	<p>Guarda un 25 al registre AX que el converteix en un 50 (SHL) i va a una rutina on posa el 50 a la capçalera de la pila, per sobre del PC guardat pel "CALL". D'aquesta manera quan retorna de la rutina, "RET", en lloc de tornar a la 53 torna a la 50 i això ens porta a un bucle indefinit.</p>
--	---

- c) Escriviu un fragment de codi en llenguatge ensamblador que emmagatzemi en R1 un 1 si es compleix una i només una de les dues condicions següents (R1, R2, etc. són els noms que li hem donat als registres en aquesta màquina): **(1 punt)**
- el contingut de R2 és més gran o igual que el de R3
 - el contingut de R4 és més petit que el de R5.
 - En qualsevol altre cas emmagatzemeu en R1 un 0.



Teoria 3. (2 punts)

Descriviu, a nivell de transferència de registres, l'execució de les instruccions següents (seguir tots els passos del model de màquina de von Neumann).

- a. **mov AX, (1000)** (AX és un registre del processador i (1000) vol dir que es fa servir el mode d'adreçament indirecte, MAR (AR) i MDR (DR) són els registres de transferència amb els busos de memòria)

```
AR ← <PC>
Read + wait memory accés time
Load DR
IR ← <DR> + PC ← <PC> + 1
Cicle de cerca de la instrucció

Decoding

AR ← <IR>Address 1000
Read + wait memory accés time
Load DR
AR ← <DR>
Es resol la indirecció

Read + wait memory accés time
Load DR
AX ← <DR>
Es carrega l'operand a AX
```

- b. **je 2000** (salta si és igual, el bit Z del registre d'estat val 1)

```
AR ← <PC>
Read + wait memory accés time
Load DR
IR ← <DR> + PC ← <PC> + 1
Cicle de cerca de la instrucció

Decoding

PC ← <IR>Address 2000 Només si Z = 1
```

Teoria 4. (2 punts)

En un moment determinat el processador es troba executant la instrucció **cmp R0, R1** que pertany a un cert programa P1. Coneixem les dades següents:

- Aquesta instrucció es troba a la posició 4000 de memòria,
- R0 i R1 són registres de 8 bits i en el moment d'executar-la R0 conté el valor 00110010 i R1 el valor 11100111
- En aquest moment el registre SP conté el valor 1000. En aquesta màquina SP apunta a la primera posició lliure de la pila i la pila creix cap a posicions majors de memòria.
- En el moment en que la ALU (Unitat Aritmètica-Lògica) està fent l'operació entre R0 i R1, s'activa l'únic senyal d'interrupció que té el processador.
- El registre d'estat té 8 bits, però només s'emmagatzemen els bits C, Z, S i O (en aquest ordre i en els bits menys significatius del registre), els altres 4 bits sempre valen 0.
- La rutina de servei que s'executarà en cas d'acceptar la interrupció és a l'adreça 400 de memòria.

Suposant que el processador acceptarà i servirà la interrupció que s'ha produït, responeu les preguntes següents:

- i) El processador acabarà d'executar la instrucció cmp R0, R1 abans d'acceptar la interrupció? (Justifiqueu-ne la resposta)

La instrucció "cmp R0,R1" acabarà d'executar-se fins al final i després atindrà la interrupció. El motiu és per evitar que hi pugui haver inconsistències en l'estat del processador si es deixés la instrucció a mitges.

- ii) Indiqueu els valors emmagatzemats en els registres comptador de programa (PC), estat (SR) i apuntador a la pila (SP) just abans de saltar a la rutina de servei d'interrupció.

R0 00110010

R1 11100111

01001011 Resta

C=0, Z=0, S=0 i O(V)=0 SR = 00000000 PC = 4001 SP = 1000

- iii) Indiqueu els valors emmagatzemats en els registres comptador de programa (PC) i apuntador a la pila (SP) just després de saltar a la rutina de servei d'interrupció.

PC = 400 SP = 1002

- iv) Indiqueu quin valor o valors s'han emmagatzemat en la pila en el moment de fer el salt a la rutina de servei d'interrupció i en quines adreces de memòria s'han emmagatzemat.

<1000> = 4001 (PC de retorn)

<1001> = 00000000 (SR de retorn)

Teoria 5. (1 punt)

Describeu breument els tres mecanismes d'E/S introduïts en el curs (E/S programada amb llaços d'espera, E/S programada amb interrupcions i Accés Directe a Memòria). Indiqueu les raons que justifiquen l'existència dels dos darrers mecanismes.

L'E/S programada de dades (TPD) amb llaços d'espera es fa mitjançant un programa executat per la CPU on:

1 es selecciona el dispositiu

2 es determina si està disponible el dispositiu

3 s'accedeix a l'i/o port i s'envia (escriu) o es rep (llegeix) la dada a transmetre.

4 si no hem acabat tornar a 2

El pas "2" es fa amb un bucle on es llegeix sistemàticament un i/o port d'estat, esperant que estigui disponible el dispositiu. En aquest bucle podem consumir el 99% dels temps de la TPD.

Si canviem aquest bucle pel fet que el controlador envii una interrupció a la CPU quan estigui disponible serà una E/S programada amb interrupcions. Això comporta que a partir del punt 3 la CPU pot anar a executar altres programes i anirà fent el pas 3 cada cop que rebi una interrupció. El pas 3 serà la rutina de tractament de la interrupció. Això evita el bucle del 99% de temps d'espera.

L'Accés Directe a Memòria consisteix en incloure un controlador de DMA. Aquest dispositiu substitueix completament la CPU en fer la transferència, mentre la CPU fa altres tasques. El controlador de DMA quan acabi la feina que se li ha encomanat d'entrada o sortida, interromprà la CPU per a que li programi una altra tasca d'e/s. Per tant la intervenció de la CPU es limita a programar el que ha de fer el DMA i el DMA fa la transferència.

Teoria 6. (1,5 punts)

Durant el curs us hem indicat en múltiples ocasions que un programa desenvolupat per un usuari no pot fer directament E/S.

a) Per quina raó no ho pot fer?

Perquè el programa d'usuari s'executa amb la CPU en mode user. Aquest mode té diverses limitacions entre les que hi ha la de no poder accedir als ports d'E/S. L'objectiu és que només el SO pugui accedir a controlar l'E/S del sistema. El SO s'executa en mode supervisor de la CPU i és l'únic programa que ho pot fer en aquest mode.

b) Quines altres limitacions té un programa d'usuari en executar-se?

Les altres limitacions són les del mode usuari de la CPU: no poder accedir a un conjunt d'adreces de la memòria principal, a un conjunt de registres de la CPU i executar un conjunt d'instruccions que es consideren privilegiades i només poden ser executades en mode supervisor, és a dir pel SO. En resum, no pot accedir directament als recursos, els ha de sol·licitar al SO.

c) Quin és l'únic software que té accés a tots els recursos del computador? Quina és la conseqüència principal d'aquest fet des del punt de vista de la gestió i ús d'aquests recursos?

Com es desprèn de l'esmentat anteriorment, l'únic software que pot tenir accés a tots els recursos del sistema és el SO, amb l'objectiu de gestionar-lo de la manera més eficient i segura possible.