

Cognoms i Nom:

NIU:

Responen les preguntes de la prova en l'espai habilitat, no s'acceptaran fulls apart. Llegiu l'enunciat de cada pregunta amb cura i contesteu només el que es pregunta, però justifiqueu sempre la vostra resposta.

Teoria 1. (1,5 punts) Responen breument les següents preguntes:

a) Què és els registres Stack Pointer (SP) MAR i MBR? (0,5 punts)

Els tres registres són de la CPU. El Stack Pointer conté l'adreça de la capçalera (Top) de la pila (Stack). Pot estar apuntant la primera posició lliure o la darrera ocupada, depèn del processador. MAR conté l'adreça de memòria on estem accedint (en lectura o escriptura) i garanteix l'estabilitat del valor durant tot el temps d'accés. El MBR conté el valor que volem escriure en una posició de memòria o recull el valor llegit de la memòria en la posició adreçada per l'MBR.

b) Què és un Mode d'adreçament (enumerar-los **no** és una resposta) (0,5 punts)

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.

e) Què són el mode usuari i el mode supervisor? Relacioneu-ho amb el Sistema Operatiu (0,5 punts)

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. (0,75 punts)

a) Què és el DMA (Accés Directe a Memòria) i com funciona? (0,5 punts)

Consisteix en un controlador (dispositiu) que substitueix la CPU en l'operació d'entrada/sortida amb els perifèrics i la memòria principal. Aquest controlador està connectat directament al bus intern del sistema i conté un registre que indica la posició de memòria on s'ha de fer la transferència, un altre amb la quantitat de dades a transferir i que porta el compte de la quantitat de dades transferides, un altre que conté l'adreça del port del perifèric amb el que s'ha de fer la transferència i un altre on s'indica el tipus d'operació que s'ha de fer.

La CPU posarà els continguts adequats en cada registre per a que es faci l'operació d'entrada sortida que calgui. El controlador de DMA actuarà automàticament i independent de la CPU fins finalitzar la transferència completa de totes les dades. En acabar l'operació el controlador de DMA genera una interrupció a la CPU per indicar el final de l'operació requerida.

Cal tenir en compte que el controlador i la CPU competiran en l'ús del bus per accedir a la memòria i al perifèric. Tot i així la velocitat de transferència millora en evitar que les dades hagin de passar per la CPU abans de ser transferides.

b) Quin és l'efecte d'executar el següent programa? (El PC = 50 inicial) (1 punt). Si descriu el programa línia per línia serà considerat incorrecte. Volem l'efecte general, global. Calcula alguna cosa? Quan acaba? Com canvia la pila? (0.25 punts)

```
50    MOV RPG, #25    ;
51    SHL RPG
52    PUSH RPG        ;
53    CALL 1000        ;
54    MOV 100, RPG     ;

1000  PULL RPG
```

1001	RTN	(RETORN)
------	-----	----------

Guarda un 25 al registre RPG que el converteix en un 50 (SHL) i el guarda a la pila. Salta a una rutina guardant el PC de retorn ala pila, a sobre del 50.

A la rutina extreu el capçal de la pila (el PC guardat) i el deixa a RPG. Al retornar de la rutina agafa com adreça de retorn (PC) el 50 i això ens porta a un bucle indefinit.

Teoria 3. (1 punt) Descriu, a nivell de microoperacions de transferència de registres, l'execució de la instrucció següent (seguir tots els passos del model de màquina de von Neumann).

- a. **MOV RPG , 3000 (X)** (RPG és un registre de propòsit general del processador i 3000 vol dir que es fa servir el mode d'adreçament indexat, MAR i MBR són els registres de transferència amb els busos de memòria. El contingut del registre X és 25.

Es porta el contingut de la posició 3025 (3000+25) al registre de la CPU RPG

MAR \leftarrow <PC>

Read

wait memory accés time

Cicle de cerca de la instrucció

Load MBR

IR \leftarrow <MBR> + PC \leftarrow <PC> + 1

Decoding

MAR \leftarrow <IR>Address + <X> 3025

Read

Cicle d'operand

wait memory accés time

Load MBR

RPG \leftarrow <MBR>

Es carrega l'operand a RPG

Teoria 4. (0,75 punts)

En un moment determinat el processador es troba executant la instrucció SUB RPG0, RPG1 que pertany a un cert programa P1. Coneixem les dades següents:

- Aquesta instrucció es troba a la posició 2500 de memòria,
- RPG0 i RPG1 són registres de 8 bits i en el moment d'executar-la RPG0 conté el valor 00110010 i RPG1 el valor 11100111
- En aquest moment el registre SP conté el valor 1555. En aquesta màquina SP apunta a la primera posició ocupada de la pila i la pila creix cap a posicions menors de la memòria.
- En el moment en que la ALU (Unitat Aritmètica-Lògica) està fent l'operació (en complement a dos) entre RPG0 i RPG1, 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, N i V (en aquest ordre i en els 4 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) 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. (0,25 punts)

R0 00110010

R1 11100111

01001011 Resta

C=0, Z=0, S=0 i O(V)=0 SR = 00000000 PC = 2501 SP = 1555

- ii) 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. (0,25 punts)

PC = 400 SP = 1553

- iii) 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. (0,25 punts)

<1554> = 2501 (PC de retorn)

<1553> = 00000000 (SR de retorn)

Teoria 5. (1 punt)

Descriviu el procés que es desencadena quan es produeix una interrupció. Relacioneu-ho amb la instrucció “retorn d'interrupció”. Quins tipus d'interrupció es poden produir en un computador? Descriviu-los.

Un interrupció és un senyal asíncron que en arribar a la CPU fa que aquesta deixi d'executar el programa en curs per executar un altre programa amb la idea que aquest segon resolgui una situació que requereix l'atenció immediata per posteriorment retornar i continuar l'execució del programa inicial.

En arribar el senyal s'acaba d'executar la instrucció en curs.

Es produeix, a la CPU, un canvi de mode User a Supervisor.

Es salva el contingut del registre d'estat i del PC a la pila

Es passa a executar la Rutina de Tractament de la Interrupció, l'adreça d'inici de la qual és normalment proporcionada pel propi perifèric. Per tant es carrega aquesta adreça al Comptador de Programa i l'execució anirà a fer la rutina d'interrupció.

La rutina d'interrupció ha d'acabar amb una instrucció de Retorn d'Interrupció que canvia la CPU de mode Supervisor a User, restaura el registre d'estat i el PC (des de la pila) i així es recupera el fil d'execució del programa que s'estava executant en el moment que va arribar la interrupció.

La interrupció hardware és la que es produeix per l'activació d'un senyal hardware indicant que s'ha produït un esdeveniment exterior a la CPU que requereix la seva atenció. Normalment és generat per un dispositiu com una impressora, un disc, un motor, un rellotge, etc.

L'excepció desencadena el mateix procés que l'anterior però la causa que el produeix és interna a la CPU, divisió per zero en una operació, intent d'executar una instrucció privilegiada o una d'il·legal (inexistent), un overflow, etc.

La interrupció software (TRAP) és una instrucció del repertori de la CPU i que pot ser inclosa al nostre programa d'usuari, que desencadena exactament el mateix procés que les anteriors. Està pensada per invocar el Sistema Operatiu a través de les anomenades crides al sistema.

Problema 1. (1 punt) Donat el fragment de codi següent, en llenguatge C, escriviu el fragment de codi equivalent en llenguatge màquina. Cas que sigui necessari podeu assignar a cada variable l'adreça que considereu adient.

```
#define N 1000
int i, A[N], B[N], C[N];
...
for ( i = 1; i < N; i++ )
    if ( A[i] < B[i] ) {C[i] = A[i]}; else {C[i] = B[i]}
```

“A[1]” situat a 3000, “B[1] situat a 2000” i “C[1]” situat a 1000

```

MOV X, #1
for: MOV RPG, 3000 (X)
      CMP RPG, 2000 (X)
      BNN else
      MOV 1000 (X), RPG
      JMP ffor
else: MOV RPG, 2000 (X)
      MOV 1000 (X), RPG
ffor: INC X

```

“X” s'utilitza per mantenir l'índex “i” del for.

Problema 2. (1 punt) Mostreu el procediment per calcular-ho

Quin és el valor en decimal (base 10) de les seqüències de bits següents i) 01010011 i 11010110 si les interpretem com:

a) Números naturals en binari (0,25 punts)

$$01010011 = 2^6 + 2^4 + 2^1 + 2^0 = 64 + 16 + 2 + 1 = 83_{(10)}$$

$$11010110 = 2^7 + 2^6 + 2^4 + 2^2 + 2^1 = 128 + 64 + 16 + 4 + 2 = 214_{(10)}$$

b) Enters representats en signe i magnitud i 8 bits (SM) Posteriorment amplieu-los a 12 bits (0,25 punts)

$$01010011 = 2^6 + 2^4 + 2^1 + 2^0 = 64 + 16 + 2 + 1 = +83_{(10)} \text{ en 12 bits : } 000001010011_{(2)}$$

$$11010110 = 2^6 + 2^4 + 2^2 + 2^1 = 128 + 64 + 16 + 4 + 2 = -214_{(10)} \text{ en 12 bits : } 100001010110_{(2)}$$

c) Números enters representats en complement a 2 i 8 bits (C2) Posteriorment amplieu-los a 12 bits (0,25 punts)

$$01010011 = 2^6 + 2^4 + 2^1 + 2^0 = 64 + 16 + 2 + 1 = +83_{(10)} \text{ en 12 bits : } 000001010011_{(2)}$$

$$11010110 = -(00101010) = -(2^5 + 2^3 + 2^1) = -(32+8+2) = -42_{(10)} \text{ en 12 bits : } 111110101110_{(2)}$$

Problema 3. (0,5 punts)

a) Quin és el valor en binari (12 bits) de la seqüència de dígit 235 si la interpretem com un número natural en base 16 (és a dir, $235_{(16)}$) (0,25 punts)

$$0010 \text{ (2)} \quad 0011 \text{ (3)} \quad 0101 \text{ (5)} \quad 001000110101_{(2)}$$

b) Un número natural en base 8 (és a dir, $235_{(8)}$) (0,25 punts)

$$010 \text{ (2)} \quad 011 \text{ (3)} \quad 101 \text{ (5)} \quad 0000010011101_{(2)}$$

c) Un número natural en base 10 (és a dir, $235_{(10)}$) (0,25 punts)

$$235 \text{ div } 2 = 117, \text{ Residu } = 1$$

$$117 \text{ div } 2 = 58, \text{ Residu } = 1$$

$$58 \text{ div } 2 = 29, \text{ Residu } = 0$$

$$29 \text{ div } 2 = 14, \text{ Residu } 1 \quad . . . \quad 000011101011_{(2)}$$

d) Calculeu en binari (4 bits fraccionaris) el número $235,82_{(10)}$ (0,25 punts)

$$0,82 \times 2 = 1,64 \quad 1$$

$$0,64 \times 2 = 1,28 \quad 1$$

$$0,28 \times 2 = 0,56 \quad 0$$

$$0,56 \times 2 = 1,12 \quad 1$$

$$235,82_{(10)} = 000011101011,1101_{(2)}$$

Problema 4. (0,5 punts)

Donats els valors $A = 11110101$ i $B = 11100001$ que representen dos números enters binaris en complement a 2 (C2) i 8 bits. (Opereu sempre en C2)

a) Quin és el valor de $A + B$? Es produeix desbordament (overflow)? (expliqueu el perquè)

b) 11110101

c) 11100001

d) 11010110 $C=1$ **Overflow = 0** en aquest cas perquè el signe dels operands és igual al del resultat, tot i que en podria haver-hi si sumem dos números del mateix signe

Problema 5 (2 punts) Els modes d'adreçament utilitzats són:

n Directe	#n immediat	n[X] Indexat	(n) Indirecte
-----------	-------------	--------------	---------------

L'SP apunta a la 1a posició ocupada de la pila i aquesta creix cap a posicions de menor numeració de memòria.

A partir del següent fragment de programa

1.-	MOV X, #1	50.-	PUSH R0
2.-	MOV R0, 200[X]	51.-	INC R0
3.-	MOV (301), R0	52.-	INC X
4.-	CALL 50	53.-	PULL R0
5.-	CMP X, #3	54.-	RTN
6.-	BNE 2		

Mostreu l'evolució dels registres, la memòria i la pila. Poseu només el que canvia.

[illegible]