



Estructura de Computadores - Examen Final - 22/01/2020 - CAT

Estructura de computadores (Universitat Oberta de Catalunya)

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

05.573 22 01 20 EX

Enganxeu en aquest espai una etiqueta identificativa amb el vostre codi personal Examen

Fitxa tècnica de l'examen

- Comprova que el codi i el nom de l'assignatura corresponen a l'assignatura matriculada.
- Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.
- No es poden adjuntar fulls addicionals, ni realitzar l'examen en llapis o retolador gruixut.
- Temps total: **2 hores** Valor de cada pregunta: **S'indica a l'enunciat**
- En cas que els estudiants puguin consultar algun material durant l'examen, quins són?
CAP En cas de poder fer servir calculadora, de quin tipus? **CAP**
- Si hi ha preguntes tipus test: Descompten les respostes errònies? **NO** Quant?
- Indicacions específiques per a la realització d'aquest examen:

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Enunciats

No es pot utilitzar calculadora. Cal saber interpretar un valor en binari, decimal o hexadecimal per a realitzar l'operació que es demani. I el resultat s'ha d'expressar en el format corresponent.

Valoració de les preguntes de l'examen

Pregunta 1 (20%)

Pregunta sobre la pràctica.

Cal completar les instruccions marcades o afegir el codi que es demana.

Els punts suspensius indiquen que hi ha més codi però no l'heu de completar.

NOTA: En cas que el codi proposat en cada pregunta no es correspongui amb la forma que vosaltres plantejaríeu la resposta, podeu reescriure el codi o part del codi segons el vostre plantejament.

1.1 : 10%

1.2 : 10%

Pregunta 2 (35%)

2.1 : 10%

2.2 : 15%

2.3 : 10%

Pregunta 3 (35%)

3.1: 15%

3.1.1 : 10%

3.1.2 : 5%

3.2: 20%

3.2.1 : 10%

3.2.2 : 5%

3.2.3 : 5%

Pregunta 4 (10%)

4.1 : 5%

4.2 : 5%

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Pregunta 1

1.1 Pràctica – 1a Part

Escriure un fragment de codi assemblador de la subrutina showMinesP1 que obté el valor de les unitats i de les desenes de la variable numMines i els deixa als registres al i dl respectivament. (No s'ha d'escriure el codi de tota la subrutina).

```

;;;;
; Converteix el valor del Número de mines que queden per marcar (numMines)
; (entre 0 i 99) a dos caràcters ASCII.
; S'ha de dividir el valor (numMines) entre 10, el quocient representarà
; les desenes i el residu les unitats, i després s'han de convertir
; a ASCII sumant 48, caràcter '0'.
; Mostra els dígit (caràcter ASCII) de les desenes a la fila 27,
; columna 24 de la pantalla i les unitats a la fila 27, columna 26,
; (la posició s'indica a través de les variables rowScreen i colScreen).
; Per a posicionar el cursor cridar a la subrutina gotoxyP1 i per a mostrar
; els caràcters a la subrutina printchP1.
; Variables globals utilitzades:
; rowScreen: Fila de la pantalla on posicionem el cursor.
; colScreen: Columna de la pantalla on posicionem el cursor.
; numMines : Nombre de mines que queden per marcar.
; charac   : Caràcter a escriure a pantalla.
;;;;
showMinesP1:
    push rbp
    mov  rbp, rsp
    push rax
    push rbx
    push rdx

    mov rax, 0
    mov eax, DWORD[numMines]
    mov edx, 0

    ;calcular unitats i desenes
    mov ebx, 10
    div ebx          ;EAX=EDX:EAX/EBX EDX=EDX:EAX%EBX

    add al, '0'      ;convertim els valors a caràcters ASCII
    add dl, '0'      ;charac = charac + '0';

    ;Posicionar el cursor i mostrar dígit
    ; AQUESTA PART DEL CODI NO S'HA D'IMPLEMENTAR
showMinesP1_End:
    pop rdx
    pop rbx
    pop rax
    mov rsp, rbp
    pop rbp

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

ret

1.2 Pràctica – 2a part

Completar el codi de la subrutina `checkEnd`, per a verificar si hem marcat totes les mines i hem obert totes les altres caselles. (Només completar els espais marcats, no es poden afegir o modificar altres instruccions).

```

; ; ; ;
; Verificar si hem marcat totes les mines (nMines=0), que es rep com
; a paràmetre i hem obert o marcat amb una mina totes les altres
; caselles i no hi ha cap espai en blanc ( ' ' ) a la matriu (marks),
; si és així, canviar l'estat (state) que es rep com a paràmetre, a
; 2 "Guanya". Retornar l'estat del joc actualitzat (status).
;
; Variables globals utilitzades:
; marks : Matriu amb les mines marcades i les mines de les obertes.
; Paràmetres d'entrada : rdi(edi) : (nMines) Mines que queden per marcar.
;                          rsi(esi) : (status) Estat del joc.
; Paràmetres de sortida: rax(eax) : (status) Estat del joc.
; ; ; ;
checkEndP2:
    push rbp
    mov rbp, rsp
    push rsi
    push rdi
    ;Guardem estat del joc per a retornar-lo
    mov eax, __esi__
    ;Mirem si hem marcat totes les mines.
    cmp __edi__, 0

    __jg__ checkEndP2_End

    mov rsi, 0 ;índex per a accedir a la matriu marks.
    ;Iniciem el bucle per a mirar si hi ha espais en blanc.
    checkEndP2_Loop:
    cmp __BYTE[marks+rsi]__, ' '
    je checkEndP2_End ;Si és un espai en blanc no hem acabat.

    ;incrementem l'índex per a accedir a la matriu
    inc rsi
    cmp rsi, __SizeMatrix__ ;DimMatrix*DimMatrix

    jl checkEndP2_Loop
    mov __eax__, 2 ;si hem mirat totes les posicions
                    ;i no hi ha cap espai, vol dir que
                    ;hem marcat totes les mines i
                    ;obert tota la resta de posicions.

    checkEndP2_End:
    pop rdi
    pop rsi
    mov rsp, rbp
    pop rbp
    ret

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Pregunta 2

2.1

L'estat inicial del computador CISCA just abans de començar l'execució de cada fragment de codi (a cada apartat) és el següent:

R0= 100h R1= 200h R2= 300h R3= 400h	M(00000200h)= M(00000300h)=00000600h M(00000400h)=0000FFFFh M(00000500h)=80000000h	Z = 0, C = 0, S = 0, V = 0
--	---	----------------------------

Completeu l'estat del computador després d'executar cada codi (indiqueu els valors dels registres en hexadecimal).

Suposeu que l'adreça simbòlica A val 200h.

a)

```
ADD R3, R1
SUB R3, [A+R1]
MOV R2, 0
JNE END
DEC R1
```

END:

```
R3:= 400h + 200h = 600h
[00000400h]=0000FFFFh
R3= 00000600h - 0000FFFFh=
FFFF0601h
R2:= 0
```

Z= 0 , S= 1 , C= 1 , V= 0

b)

```
MOV R2, [A+R2]
CMP R2, [A]
JNE EXIT
...
```

EXIT:

```
R2 = [00000500h]=80000000h
80000000h-FFFFF600h=80000A00
```

Z=0, C=1, S=1, V=0

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

2.2

Suposem que tenim el vector V de 10 elements de 32 bits. Completar la traducció del programa en ensamblador CISCA perquè executi l'algorisme d'alt nivell mostrat. (Hem deixat 8 espais per omplir)

```

i = 0;
do {
    if (V [i] < i) V[i] = 0;
    else V[i] = V[i]*2;
    i= i+1;
}
while (i<=9)

```

R2 representa l'índex i

```

MOV R1, 0
MOV R2, 0
NEXT: CMP [V+R1], R2
      JL  CERO
      MUL [V+R1], 2
      JMP CONT
CERO: MOV [V+R1], 0
CONT: ADD R1, 4
      ADD R2, 1
      CMP R2, 9
      JLE NEXT
END:

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

2.3

Donat el següent fragment de codi d'un programa en llenguatge ensamblador de CISCA:

ADD [00023C00h+R1], 100h

CMP [R10], R2

MOV R1, [R2+A]

Traduïu-ho a llenguatge màquina i expresseu-ho en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 00023C00h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00004000h. En la següent taula useu una fila per a codificar cada instrucció. Si suposem que la instrucció comença en l'adreça @, el valor Bk de cadascun dels bytes de la instrucció amb adreces @+k per a k=0, 1,... s'ha d'indicar en la taula en hexadecimal en la columna corresponent (recordeu que els camps que codifiquen un desplaçament en 2 bytes o un immediat o una adreça en 4 bytes ho fan en format little endian, això cal tenir-ho en compte escrivint els bytes de menor pes, d'adreça més petita, a l'esquerra i els de major pes, adreça major, a la dreta). Completeu també la columna @ que indica per a cada fila l'adreça de memòria del byte B0 de la instrucció que es codifica en aquesta fila de la taula.

A continuació us donem com a ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
20h	ADD
26h	CMP
10h	MOV

Taula de modes d'adreçament (Bk<7..4>)

Camp mode Bk<7..4>	Mode
0h	Immediat
1h	Registre
2h	Memòria
3h	Indirecte
4h	Relatiu
5h	Indexat
6h	Relatiu a PC

Taula de modes d'adreçament (Bk<3..0>)

Camp mode Bk<3..0>	Significat
Num. registre	Si el mode ha d'especificar un registre
0	No s'especifica registre.

		Bk per a k=0..10											
@	Assemblador	0	1	2	3	4	5	6	7	8	9	10	
00023C00h	ADD [00023C00h+R1], 100h	20	51	00	3C	02	00	00	00	01	00	00	
00023C0Bh	CMP [R10], R2	26	3A	12									
00023C0Eh.	MOV R1, [R2+A]	10	11	42	00	40	00	00					
00023C15h													

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Pregunta 3

3.1. Memòria cau

Tenim un sistema de memòria en el que tots els accessos es fan a paraula (no ens importa quina és la mida d'una paraula). Suposarem que l'espai d'adreces de memòria es descompon en blocs de 8 paraules. Cada bloc comença en una adreça múltiple de 8. Així, el bloc 0 conté les adreces 0, 1, 2, 3, 4, 5, 6, 7, el bloc 1, les adreces 8, 9, 10, 11, 12, 13, 14, 15, i el bloc N les adreces $8*N$, $8*N+1$, $8*N+2$, $8*N+3$, $8*N+4$, $8*N+5$, $8*N+6$, $8*N+7$.

Suposem que el sistema també disposa d'una memòria cau de 4 línies (on cada línia té la mida d'un bloc, es a dir, 8 paraules). Aquestes línies s'identifiquen com a línies 0, 1, 2 i 3. Quan es fa referència a una adreça de memòria principal, si aquesta adreça no es troba a la memòria cau, es porta tot el bloc corresponent des de la memòria principal a una línia de la memòria cau (així si fem referència a l'adreça 2 de memòria principal portarem el bloc format per les paraules 0, 1, 2, 3, 4, 5, 6, 7).

Suposem que el sistema fa servir una **política d'assignació directa**, de manera que cada bloc de la memòria principal només es pot portar a una línia determinada de la memòria cau.

L'execució d'un programa genera la següent llista de lectures a memòria:

0, 1, 2, 12, 61, 62, 63, 64, 17, 18, 19, 32, 4, 6, 65, 66, 20, 56, 42, 50

La següent taula mostra l'estat inicial de la cau, que conté les primeres 32 paraules de la memòria (organitzades en 4 blocs).

Completar la taula per a mostrar l'evolució de la cau durant l'execució del programa. Per a cada accés cal omplir una columna indicant si es tracta d'un encert o una fallada.

Si és un encert escriurem E en la línia corresponent davant de les adreces del bloc, si és una fallada escriurem F i en aquest cas s'indicarà el nou bloc que es porta a la memòria cau en la línia que li correspongui, expressat de la forma b:e ($a_0 - a_7$) on b:número de bloc, e:etiqueta i ($a_0 - a_7$) són les adreces del bloc, on a_0 és la primera adreça del bloc i a_7 és la vuitena (darrera) adreça del bloc.

Línia	Estat Inicial	0	1	2	12	61
0	0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	E 1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	F 7:1 (56 - 63)

Línia	62	63	64	17	18	19
0	0:0 (0 - 7)	0:0 (0 - 7)	F 8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)	E 2:0 (16 - 23)	E 2:0 (16 - 23)
3	E 7:1 (56 - 63)	E 7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Línia	32	4	6	65	66	20
0	F 4:1 (32 - 39)	F 0:0 (0 - 7)	E 0:0 (0 - 7)	F 8:2 (64 - 71)	E 8:2 (64 - 71)	8:2 (64 - 71)
1	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)	1:0 (8 - 15)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	E 2:0 (16 - 23)
3	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)

Línia	56	42	50			
0	8:2 (64 - 71)	8:2 (64 - 71)	8:2 (64 - 71)			
1	1:0 (8 - 15)	F 5:1 (40 - 47)	5:1 (40 - 47)			
2	2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)			
3	E 7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)			

3.1.2 a) Quina és la taxa d'encerts (T_e) ?

$$T_e = 13 \text{ encerts} / 20 \text{ accessos} = 0,65$$

3.1.2 b) Suposem que el temps d'accés a la memòria cau, o temps d'accés en cas d'encert (t_e), és de 4 ns i el temps total d'accés en cas de fallada (t_f) és de 20 ns. Considerant la taxa de fallades obtinguda a la pregunta anterior, quin és el temps mitjà d'accés a memòria (t_m) ?

$$t_m = T_e \times t_e + (1 - T_e) \times t_f = 0,65 \times 4 \text{ ns} + 0,35 \times 20 \text{ ns} = 2,6 \text{ ns} + 7 \text{ ns} = 9,6 \text{ ns}$$

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

3.2 Sistema d'E/S

Es vol analitzar el rendiment de la comunicació de dades entre la memòria d'un processador i un port USB, utilitzant E/S programada, amb les següents característiques:

- Velocitat de transferència del dispositiu d'E/S $v_{\text{transf}} = 4 \text{ MBytes/s} = 4000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat i dades** del controlador d'E/S: 0F00h i 0F04h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 4, o sigui el cinqué bit menys significatiu (quan val 1 indica que està disponible)
- Processador amb una freqüència de rellotge de 2 GHz, el temps de cicle $t_{\text{cicle}} = 0,5 \text{ ns}$.
- El processador pot executar 1 instrucció per cicle de rellotge
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de $N_{\text{dades}} = 200000$ dades
- La mida d'una dada és $m_{\text{dada}} = 4 \text{ bytes}$
- Adreça inicial de memòria on resideixen les dades: A0000000h

a) El següent codi realitzat amb el joc d'instruccions CISC realitza la transferència descrita abans mitjançant la tècnica d'E/S programada. Completeu el codi.

```

1.      MOV R3, 200000
2.      MOV R2, A0000000h
3. Bucle: IN R0, [0F00h] ; llegir 4 bytes
4.      AND R0, 00010000b
5.      JE Bucle
6.      MOV R0, [R2] ; llegir 4 bytes
7.      ADD R2, 4
8.      OUT [0F04h], R0 ; escriure 4 bytes
9.      SUB R3, 1
10.     JNE Bucle

```

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

b) Quant temps dura la transferència del bloc de dades $t_{\text{transf_bloc}}$?

$$t_{\text{transf_bloc}} = t_{\text{latència}} + (N_{\text{dades}} * t_{\text{transf_dada}})$$

$$t_{\text{latència}} = 0$$

$$N_{\text{dades}} = 200000$$

$$t_{\text{transf_dada}} = m_{\text{dada}} / v_{\text{transf}} = 4 \text{ Bytes} / 4000 \text{ Kbytes/s} = 0,001 \text{ ms}$$

$$t_{\text{transf_bloc}} = 0 + (200000 * 0,001 \text{ ms}) = 200 \text{ ms} = 0,2 \text{ s}$$

c) Si volguéssim fer servir el mateix processador i el mateix programa però amb un dispositiu d'E/S més ràpid, quina és la màxima taxa o velocitat de transferència del nou dispositiu que es podria suportar sense que el dispositiu s'hagués d'esperar?

$$t_{\text{cicle}} = 0,5 \text{ ns (nanosegons)}$$

$$t_{\text{instr}} = 0,5 \text{ ns} / 1 = 0,50 \text{ ns}$$

El mínim nombre d'instruccions que ha d'executar el programa per a cada dada transferida són les 8 instruccions: 3, 4, 5, 6, 7, 8, 9 i 10. Executar les 8 instruccions requereix $8 * t_{\text{instr}} = 8 * 0,50 \text{ ns} = 4 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 4 ns

Es poden transferir 4 bytes cada 4 ns, es a dir: $4 / 4 * 10^{-9} = 1000 \text{ Mbyte/s} = 1 \text{ Gbytes/s}$

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30

Pregunta 4

4.1

Què és la senyal de rellotge en un processador? En què unitat es mesura?

La senyal de rellotge del processador és una senyal de sincronització que marca el ritme d'execució d'operacions del processador. Es mesura en cicles per segons o hertz i és un dels elements que determina la velocitat d'un processador.

4.2

4.2.1 Un dels factors bàsics que fan que l'esquema de jerarquia de memòries funcioni satisfactòriament és la proximitat referencial. Quins tipus de proximitat referencial podem distingir? Explicar breument en què consisteix cadascun d'ells.

Distingim dos tipus de proximitat referencial:

- 1) **Proximitat temporal.** És quan, en un interval de temps determinat, la probabilitat que un programa accedeixi de manera repetida a les mateixes posicions de memòria és molt gran.
La proximitat temporal és deguda principalment a les estructures iteratives; un bucle executa les mateixes instruccions repetidament, de la mateixa manera que les crides repetitives a subrutines.
- 2) **Proximitat espacial.** És quan, en un interval de temps determinat, la probabilitat que un programa accedeixi a posicions de memòria properes és molt gran.
La proximitat espacial és deguda principalment al fet que l'execució dels programes és seqüencial – s'executa una instrucció darrere l'altra llevat de les bifurcacions –, i també a la utilització d'estructures de dades que estan emmagatzemades en posicions de memòria contigües.

4.2.2 Una manera d'optimitzar les operacions d'E/S per DMA consisteix a reduir el nombre de cessions i recuperacions del bus, mitjançant una modalitat de transferència anomenada mode ràfega. Quin és el funcionament en el cas d'una transferència del dispositiu a la memòria?

Cada cop que el mòdul d'E/S té una dada disponible el controlador de DMA l'emmagatzema en la memòria intermèdia i decrementa el registre comptador. Quan la memòria intermèdia és plena o el comptador ha arribat a zero, sol·licita el bus. Un cop el processador li cedeix el bus, escriu a memòria tot el conjunt de dades emmagatzemades en la memòria intermèdia, i fa tants accessos a memòria com dades tenim i actualitza el registre d'adreces de memòria en cada accés. En acabar la transferència del conjunt de dades allibera el bus.

Un cop acabada una ràfega, si el registre comptador no ha arribat a zero, comença la transferència d'una nova ràfega.

Examen 2019/20-1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573	22/01/2020	15:30