

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573		

05.573R14R01R15REEE  
05.573 14 01 15 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 en la qual estàs matriculat.
- Només has d'enganxar una etiqueta d'estudiant a l'espai corresponent d'aquest full.
- No es poden adjuntar fulls addicionals.
- No es pot realitzar la prova en llapis ni en retolador gruixut.
- Temps total: 2 h.
- En cas que els estudiants puguin consultar algun material durant l'examen, quin o quins materials poden consultar?  
No es pot utilitzar calculadora, ni material auxiliar.
- Valor de cada pregunta: Pregunta 1 (20%); Pregunta 2 (40%); Pregunta 3 (40%)
- En cas que hi hagi preguntes tipus test: Descompten les respostes errònies? NO Quant?
- Indicacions específiques per a la realització d'aquest examen:

## 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.

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## 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 (40%)

**2.1: 15%**

**2.2: 15%**

**2.3: 10%**

### Pregunta 3 (40%)

**3.1: 15%**

**3.1.1: 10%**

**3.1.2: 5%**

**3.2: 25%**

**3.2.1: 15%**

**3.2.2: 10%**

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## Pregunta 1

### 1.1 Pràctica – Part obligatòria

Escriure un fragment de codi ensamblador de la subrutina GetSecretCode que inicialitzi el vector secret a 0's. Només heu de fer la part de la subrutina que se us demana.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; GetCodeSecret
; Llegir la combinació secreta.
; Primer netejar l'espai on es llegeix la combinació amb espais en
; blanc, cridant la subrutina clearArea
; Inicialitzar a zeros el vector secret.
; Posar showChar=0, per a indicar que Printch no mostri els
; caràcters llegits i mostri *.
; Llegir la combinació secreta (5 dígit) cridant a getch i
; emmagatzemar-la a secret
;
; Variables utilitzades:
; secret : vector que emmagatzema la combinació secreta
; showChar: 0: mostrar un * per al caràcter llegit
; carac : variable on la subrutina getch deixa el caràcter
; pitjat

; Paràmetres d'entrada :
; Cap
;
; Paràmetres de sortida:
; Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
GetSecretCode:

```

```

    mov esi,0
GSC_iniVec:
    mov byte [secret+esi],0
    inc esi
    cmp esi, 5
    jl GSC_iniVec

```

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573		

## 1.2 Pràctica – Part opcional

Completar el codi de la subrutina PrintHits per a que mostri els encerts fora de lloc.

```

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; PrintHits
; Mostra els encerts a lloc i fora de lloc.
; Mostra els encerts a lloc, mostrant una 'X' a la posició
; corresponent a l'encert. Mostra els encerts fora de lloc,
; mostrant una 'O' a la posició corresponent a l'encert.
; Si hi ha un encert a lloc del mateix dígit, no es mostra
; l'encert fora de lloc. Si el mateix dígit apareix a la jugada
; més d'un cop, sols es mostra un encert fora de lloc
; Per mostrar les X's i les O's, cridar la subrutina printch passant
; el caràcter mitjançant el registre dil
;
; Variables utilitzades:
; hits: Variable de 5 char. Cada posició identifica si hi ha
; hagut encert a lloc (1), fora de lloc (2) o no (0)
;
; Paràmetres d'entrada : Cap
;
; Paràmetres de sortida: Cap
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
PrintHits:
    ...
    mov esi, 0
next_print:
    cmp byte[hits+esi],1
    jne no_hit_place
    mov al,'X'
    mov dil,al
    call printch
    jmp next_hit
no_hit_place:
    cmp byte[hits+esi],2
    jne no_hit2
    mov al,'O'
    mov dil,al
    call printch
    jmp next_hit
no_hit2:
    mov al,' '
    mov dil,al
    call printch
next_hit:
    inc esi
    cmp esi, 5
    jl next_print

PH_end:
    ...
    ret

```

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## Pregunta 2

### 2.1

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

R0 = 00000A10h R1 = 00000B20h R2 = 00000C30h	M(00000A10h) = 0000F00Fh M(00000B20h) = 0000F000h M(00000C30h) = 0000FF00h M(000000F0h) = 00000001h M(000FF0A0h) = 0000000Ah	Z = 0, C = 1, S = 1, V = 0
----------------------------------------------------	------------------------------------------------------------------------------------------------------------------------------------------	----------------------------

L'adreça simbòlica A val 000FF0A0h. Quin serà l'estat del computador després d'executar cada fragment de codi? (només modificacions, excloent-hi el PC).

a)	b)
SUB R0, R1 ADD [R0+200h], R2	MOV R2, [A] MOV [R1], R0
R0 = A10h – B20h = FFFFFFFF0h M(000000F0h) = 00000001h + 00000C30h = 00000C31h  Z = 0, S = 0, C = 0, V = 0	R2 = 0000000Ah M(00000B20) = 00000A10h  Z = 0, S = 1, C = 1, V = 0

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## 2.2

Donat el següent codi en alt nivell:

```
if (A[i] > b) b = A[i]+b;
```

A és un vector de 8 elements de 4 bytes cadascun. Es proposa la següent traducció a CISCA on hem deixat 5 llocs per omplir:

```
MOV R0, [i]
SAL R0, 2
MOV R1, [b]
CMP [A+R0], R1
JLE END
ADD R1, [A+R0]
MOV [b], R1
```

END :

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## 2.3

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

```
MOV R1, A
ADD R2, 6
CMP R1, R2
```

Tradueu-lo a llenguatge màquina i expresseu-lo en la següent taula. Supposeu que la primera instrucció del codi s'assembla a partir de l'adreça 000000C0h (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val 00000400h. 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-lo 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 ajuda les taules de codis:

Taula de codis d'instrucció

B0	Instrucció
20h	ADD
10h	MOV
26h	CMP

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
Nº 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	
000000C0h	MOV R1, A	10	11	00	00	04	00	00					
000000C7h	ADD R2, 6	20	12	00	06	00	00	00					
000000CEh	CMP R1, R2	26	11	12									
000000D1h													

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## Pregunta 3

### 3.1. Memòria cau

#### Memòria cau d'assignació directa

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:

7, 8, 24, 23, 38, 39, 40, 45, 46, 47, 48, 2, 4, 6, 63, 40, 48, 56, 42, 50

**3.1.1** 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	7	8	24	23	38
0	0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	F 4:1 (32 - 39)
1	1:0 (8 - 15)	1:0 (8 - 15)	E 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)	E 2:0 (16 - 23)	2:0 (16 - 23)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	E 3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)

Línia	39	40	45	46	47	48
0	E 4:1 (32 - 39)	4:1 (32 - 39)	4:1 (32 - 39)	4:1 (32 - 39)	4:1 (32 - 39)	4:1 (32 - 39)
1	1:0 (8 - 15)	F 5:1 (40 - 47)	E 5:1 (40 - 47)	E 5:1 (40 - 47)	E 5:1 (40 - 47)	5:1 (40 - 47)
2	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	2:0 (16 - 23)	F 6:1 (48 - 55)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)



# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadores	05.573		

Línia	2	4	6	63	40	48
0	F 0:0 (0 - 7)	E 0:0 (0 - 7)	E 0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)
1	5:1 (40 - 47)	5:1 (40 - 47)	5:1 (40 - 47)	5:1 (40 - 47)	E 5:1 (40 - 47)	5:1 (40 - 47)
2	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	6:1 (48 - 55)	E 6:1 (48 - 55)
3	3:0 (24 - 31)	3:0 (24 - 31)	3:0 (24 - 31)	F 7:1 (56 - 63)	7:1 (56 - 63)	7:1 (56 - 63)

Línia	56	42	50			
0	0:0 (0 - 7)	0:0 (0 - 7)	0:0 (0 - 7)			
1	5:1 (40 - 47)	E 5:1 (40 - 47)	5:1 (40 - 47)			
2	6:1 (48 - 55)	6:1 (48 - 55)	E 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 de fallades ( $T_f$ ) ?

$$T_f = 5 \text{ fallades} / 20 \text{ accessos} = 0,25$$

**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 ( $t_m$ ) ?

$$t_m = T_f \times t_f + (1 - T_f) \times t_e = 0,25 \times 20 \text{ ns} + 0,75 \times 4 \text{ ns} = 5 \text{ ns} + 3 \text{ ns} = 8 \text{ ns}$$

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

## 3.2 Sistema d'E/S

### 3.2.1 E/S programada

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  $v_{\text{transf}} = 1 \text{ MBytes/s} = 1000 \text{ Kbytes/s}$
- Temps de latència mitjà del dispositiu  $t_{\text{latència}} = 0$
- Adreces dels **registres d'estat** i **dades** del controlador d'E/S: 0AF0h i 0AF4h
- El bit del **registre d'estat** que indica que el controlador del port d'E/S està disponible és el bit 2, o el tercer 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 4 instruccions per cicle de rellotge
- Transferència de **escriptura** des de memòria al port d'E/S
- Transferència de  $N_{\text{dades}} = 50000$  dades
- La mida d'una dada és  $m_{\text{dada}} = 4 \text{ bytes}$
- Adreça inicial de memòria on resideixen les dades: A000F000h

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

```

1.      MOV R3, 50000
2.      MOV R2, A000F000h
3. Bucle: IN  R0, [0AF0h]      ; llegir 4 bytes
4.      AND R0, 00000100b
5.      JE  Bucle
6.      MOV R0, [R2]           ; llegir 4 bytes
7.      ADD R2, 4
8.      OUT [0AF4h], R0       ; escriure 4 bytes
9.      SUB R3, 1
10.     JNE Bucle

```

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

$$\begin{aligned}
 t_{\text{transf\_bloc}} &= t_{\text{latència}} + (N_{\text{dades}} * t_{\text{transf\_dada}}) \\
 t_{\text{latència}} &= 0 \\
 N_{\text{dades}} &= 50000 \\
 t_{\text{transf\_dada}} &= m_{\text{dada}} / v_{\text{transf}} = 4 \text{ Bytes} / 1000 \text{ Kbytes/s} = 0,004 \text{ ms} \\
 t_{\text{transf\_bloc}} &= 0 + (50000 * 0,004 \text{ ms}) = 200 \text{ ms} = 0,2 \text{ s}
 \end{aligned}$$

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

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)

$t_{\text{instr}} = 0,5 \text{ ns} / 4 = 0,125 \text{ ns}$

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  $8 * t_{\text{instr}} = 8 * 0,125 \text{ ns} = 1 \text{ ns}$

Per tant, el temps mínim per a transferir una dada és: 1 ns

Es poden transferir 4 bytes cada 1 ns, es a dir:  $4 / 1 * 10^{-9} = 4000 \text{ Mbyte/s} = 4 \text{ Gbytes/s}$

## 3.2.2 Qüestions sobre E/S

a) En un sistema amb un mapa independent de memòria d'E/S, ens calen instruccions específiques per accedir-hi? Si calen, quines són? i si no calen, explica breument perquè.

Mapa independent d'E/S. Hi ha distinció entre adreces de memòria i registres d'E/S. Les línies d'adreces s'acostumen a compartir, però hi cal afegir algunes línies de control per a distingir si un accés és a memòria o a un port d'E/S. També són necessàries instruccions específiques d'E/S. Les instruccions utilitzades habitualment són IN (per a llegir del port d'E/S) i OUT (per a escriure en el port d'E/S).

Aquest sistema té l'avantatge que la memòria disposa de tot el rang d'adreces i el clar desavantatge que disposa d'un reduït nombre d'instruccions específiques d'E/S que només disposen dels modes d'adreçament més bàsics per a accedir als ports d'E/S.

# Examen 1

Assignatura	Codi	Data	Hora inici
Estructura de computadors	05.573		

**b)** Si un perifèric ha enviat una dada al mòdul d'E/S per a ser transferida al processador, que ha de fer el mòdul d'E/S si utilitza E/S programada? I si utilitza E/S per interrupcions?

En E/S programada, simplement actualitzar el registre d'estat del mòdul d'E/S per a indicar que té la dada disponible. En E/S per interrupcions activar el senyal INT. (La lectura de la dada, per a portar-la a memòria la fa el processador, no el mòdul d'E/S).

**c)** En un sistema d'E/S gestionat per DMA. Quina funció tenen les senyals BUSREQ i BUSACK? Qui les activa i perquè?

Per a controlar l'accés al bus són necessaris dos senyals, BUSREQ i BUSACK (semblants als senyals INT i INTA utilitzats en interrupcions). Amb el senyal BUSREQ el controlador de DMA sol·licita el control del bus i el processador cedeix el bus activant el senyal BUSACK.