

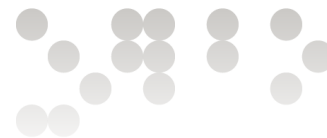
PAC1

Estructura de computadors

Programa

set15-feb16

Estudis d'informàtica, multimèdia i comunicació



Presentació

La present PAC1 conté 4 preguntes i representa el 50% de la nota de l'avaluació contínua.

Com podreu veure, els exercicis són molt semblats als quals heu fet durant aquests dies, en els quals a més heu pogut donar les solucions, comentar-les i plantejar dubtes en el fòrum. Aquesta PAC és **individual**, **avaluable** i per tant no pot comentar-se.

Competències

Les competències específiques que persegueix la PAC1 són:

- [13] Capacitat per identificar els elements de l'estructura i els principis de funcionament d'un ordinador.
- [14] Capacitat per analitzar l'arquitectura i organització dels sistemes i aplicacions informàtics en xarxa.
- [15] Conèixer les tecnologies de comunicacions actuals i emergents i saber-les aplicar convenientment per dissenyar i desenvolupar solucions basades en sistemes i tecnologies de la informació.

Objectius

Els objectius de la següent PAC són:

- Conèixer el joc d'instruccions de la màquina CISCA.
- Conèixer els modes d'adreçament de la màquina CISCA.
- Traduir petits programes a ensamblador.
- Comprendre la codificació interna de les instruccions d'ensamblador.
- Descriure les microoperacions involucrades en l'execució d'una instrucció d'ensamblador.

Enunciat

Respondre cada pregunta o apartat en el requadre corresponent.

Recursos

Podeu consultar els recursos disponibles a l'aula, però no fer ús del fòrum.

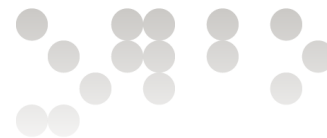
Criteris de valoració

La **puntuació** de cada pregunta i els **criteris d'avaluació** els trobareu a cada pregunta.



Format i data de lliurament

- La PAC1 podeu lliurar-la a l'apartat de **lliurament d'activitats** amb el nom **cognom1_cognom2_nom_PAC1** (pdf / odt / doc / docx).
- La data **límit** de lliurament és el **30/10/2015**.



Enunciat

Pregunta 1 (2 punts)

Suposeu que l'estat inicial del computador (el valor que contenen els registres, posicions de memòria i bits de resultat just abans de començar l'execució de cada fragment de codi, de cada apartat) és el següent:

R0= 100h	M(00000200h)=FFFFFF600h
R1= 200h	M(00000300h)=00000600h
R2= 300h	M(00000400h)=0000FFFFh
R3= 400h	M(00000500h)=60000000h

- Bits de resultat del registre d'estat: Z=0, S=0, C=0, V=0
- Registres especials: suposem que el PC apunta a l'inici del fragment de codi de cada apartat.

Quin serà l'estat del computador després d'executar cadascun dels següents fragments de codi? Indiqueu solament el contingut (**en hexadecimal**) dels registres i posicions de memòria que s'hagin modificat com a resultat de l'execució del codi. Indiqueu el valor final de tots els bits de resultat. (No us demanem que indiqueu el valor del PC després d'executar el codi i per això no us hem donat el valor inicial del PC, on comença cada fragment de codi).

Suposeu que l'adreça simbòlica A val 200h.



a)

```
MOV  R3, FFFFFFFF0h
ADD  R3, [A+R0]
MOV  [A], R0
```

Solució:

$$R3 := \text{FFFFFFF0h}$$

$$R3 := \text{FFFFFFF0h} + [\text{00000300h}] = \text{FFFFFFF0h} + \text{00000600h} = \text{000005F0h}$$

$$[A] = [\text{00000200h}] = 100\text{h}$$

$$Z = \underline{0} \text{ , } S = \underline{0} \text{ , } C = \underline{1} \text{ , } V = \underline{0}$$

b)

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

Solució:

$$R3 := 400\text{h} + 200\text{h} = 600\text{h}$$

$$R3 := R3 - [\text{00000400h}] = \text{00000600h} - \text{0000FFFFh} = \text{FFFF0601h}$$

$$R2 := 0$$

$$Z = \underline{0} \text{ , } S = \underline{1} \text{ , } C = \underline{1} \text{ , } V = \underline{0}$$



c)

```
        MOV    R0, [R0+100h]
PLUS:   CMP    R0, 0
        JGE    ELSE
        ADD    R0, 1
        JMP    PLUS
ELSE:   MOV    R0, R2
END:
```

Solució:

```
R0:= [00000200h] = FFFFFFF600h
CMP FFFFFFF600h, 0h
R0:= FFFFFFF600h + 1 = FFFFFFF601h
...
R0:= 0
R0:= 300h

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

Criteris de valoració. Els apartats a) i b) valen 0,5 punts cadascun i el c) val 1 punt. La valoració de cada apartat és del 100% si no hi ha cap error en la solució de l'apartat, és del 50% si el contingut de les posicions de memòria i registres modificats és correcte però hi ha algun error en el contingut de un o varis dels bits de resultat, i és del 0% si hi ha algun error en alguna posició de memòria o registre modificat.



Pregunta 2 (2 punts)

En el codi d'alt nivell M és una variable de tipus vector de dues dimensions (matriu) de 10 per 10 elements. Cada element de la matriu és un enter de 32 bits. En el programa ensamblador la matriu es troba emmagatzemada a partir de l'adreça simbòlica M , per files i en posicions consecutives ($M[0][0]$, en l'adreça simbòlica M , el següent, $M[0][1]$, en l'adreça $M+4$, ..., $M[1][0]$ en l'adreça $M+40$, $M[1][1]$ en l'adreça $M+44$, etc.).

El programa en cada fila, actualitza els valors de manera que cada posició de columna J contindrà la suma dels elements $J+1$ a $N-1$ (N en aquest cas seria 10).

Al programa ensamblador, $R1$ s'utilitza per adreçar l'element I , $R2$ per adreçar J , i $R4$ per adreçar AUX .

```
I:= 0;

J:= 0;

WHILE I<10 {

    J:= 8;

    WHILE J>=0 {

        AUX:= M[I][J+1];

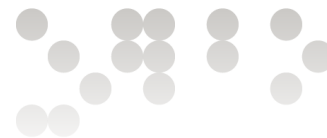
        M[I][J]:= M[I][J]+AUX;

        J:= J-1;

    }

    I:= I+1;

}
```

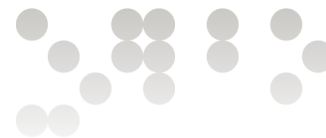


Solució:

```
                MOV R1, 0
                MOV R2, 0
WI:             CMP R1, 400
                JGE ENDWI
                MOV R2, 32
WJ:             CMP R2, 0
                JL  ENDWJ
                MOV R3, R1
                ADD R3, R2
                ADD R3, 4
                MOV R4, [M+R3]
                SUB R3, 4
                ADD [M+R3], R4
                SUB R2, 4
                JMP WJ
ENDWJ:          ADD R1, 40
                JMP WI
ENDWI:
```

Criteris de valoració.

2 punts. Es perden 0,5 punts per cada instrucció incorrecta.



Pregunta 3 (3 punts)

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

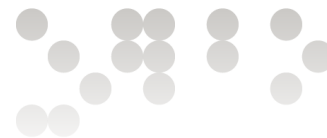
```

ADD    R3, R1
SUB    [A+R1], R5
MOV    R2, 0
JNE    END
DEC    R1
END:   XOR [A] , R3

```

Traduiu-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 **003FC000h** (que és el valor del PC abans de començar l'execució del fragment de codi). Supposeu que l'adreça simbòlica A val **00003E80h**. En la següent taula utilitzeu una fila per 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 'Adreça' 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.

Adreça	Assemblador	Bk per a k=0..10										
		0	1	2	3	4	5	6	7	8	9	10
003FC000	ADD R3, R1	20	13	11								
003FC003	SUB [A +R1], R5	21	51	80	3E	00	00	15				
003FC00A	MOV R2,0	10	12	00	00	00	00	00				
003FC011	JNE END	42	60	02	00							
003FC015	DEC R1	25	11									
003FC017	XOR [A] , R3	32	20	80	3E	00	00	13				
003FC01E												



Criteris de valoració. Cada instrucció assemblada incorrectament resta 0.5 punts. Una instrucció està incorrectament assemblada si no s'escriu el valor o s'escriu un valor incorrecte de un o varis dels dígit hexadecimale que codifiquen la instrucció en llenguatge màquina. A més es resta 0.5 punts si hi ha algun error en la columna que indica les adreces de memòria en les quals comença cada instrucció.

Pregunta 4 (2 punts)

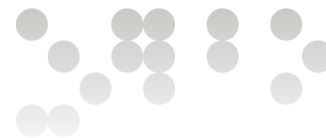
El *cicle d'execució* de una instrucció es divideix en 3 fases principals

- 1)** Lectura de la instrucció
- 2)** Lectura dels operands font
- 3)** Execució de la instrucció i emmagatzematge de l'operant destinació

Donar la seqüència de micro-operacions que cal executar en cada fase per a les següents instruccions del codi codificat en la pregunta anterior.

JNE END

Fase	Micro-operacions
1	$(MAR=003FC011h) \leftarrow (PC=003FC011h), \text{ read ;PC al MAR}$ $(MBR=00026042h) \leftarrow \text{Memòria} \quad ;\text{llegim instrucció}$ $(PC=003FC015) \leftarrow (PC=003FC011h) + 4 \quad ;\text{incrementem PC}$ $(IR=00026042h) \leftarrow (MBR=00026042h) \quad ;\text{carreguem instr. a IR}$
2	Relatiu a PC. No cal fer cap operació per obtenir l'operand.
3	Si Z=0 $(PC=003FC017h) \leftarrow (PC=003FC015h) + IR(\text{desplaç}=0002h)$

**SUB [A+R1], R5**

Fase	Micro-operacions
1	$(MAR=0003FC003h) \leftarrow (PC=003FC003h)$, read ;PC al MAR $(MBR=1500003I805121h) \leftarrow \text{Memòria}$;llegim instrucció $(PC=003FC00Ah) \leftarrow (PC=003FC003h) + 7$;incrementem PC $(IR=1500003I805121h) \leftarrow (MBR=1500003I805121h)$;carreguem ;instr. a IR
2	$MAR \leftarrow IR(\text{Adreça}=03E80h) + IR(\text{Registre índex } R1)$, read $MBR \leftarrow \text{Memòria}$
3	$MBR \leftarrow MBR - R5$, write memòria $\leftarrow MBR$

Criteris de valoració. Si tot està correcte s'obtenen 2 punts. Cada instrucció és independent i val 1 punt. Es resta 0.5 per cada fallada dins de la mateixa instrucció.



Pregunta 5 (1 punt)

Respon a les següents preguntes:

Pregunta 1. La unitat de control dels processadors pot ser microprogramada o cablejada. Quins avantatges té una unitat de control microprogramada?

Els avantatges són:

- Simplifica el disseny.
- És més econòmica.
- És més flexible:
 - Adaptable a la incorporació de noves instruccions.
 - Adaptable a canvis d'organització, tecnològics, etc.
- Permet disposar d'un joc d'instruccions amb gran nombre d'instruccions. Solament cal posseir una memòria de control de gran capacitat.
- Permet disposar de diversos jocs d'instruccions en una mateixa màquina (emulació de màquines).
- Permet ser compatible amb màquines anteriors de la mateixa família.
- Permet construir màquines amb diferents organitzacions però amb un mateix joc d'instruccions.

Pregunta 2. Si parlem de formats de les instruccions, quins són els elements que componen una instrucció?

- Codi d'operació: especifica l'operació que fa la instrucció.
- Operant font: per fer l'operació poden ser necessaris un o més operands font; un o més d'aquests operands poden ser implícits.
- Operant destinació: emmagatzema el resultat de l'operació realitzada. Pot estar explícit o implícit. Un dels operands font es pot utilitzar també com operant destinació.
- Adreça de la instrucció següent: especifica on està la instrucció següent que s'ha d'executar; sol ser una informació implícita, ja que el processador va a buscar automàticament la instrucció que es troba a continuació de l'última instrucció executada. Solament les instruccions de ruptura de seqüència especifiquen una adreça alternativa.



Criteris de valoració. Cada pregunta individual val 0,5 punts si és correcta. 0 si no ho és o és incompleta.