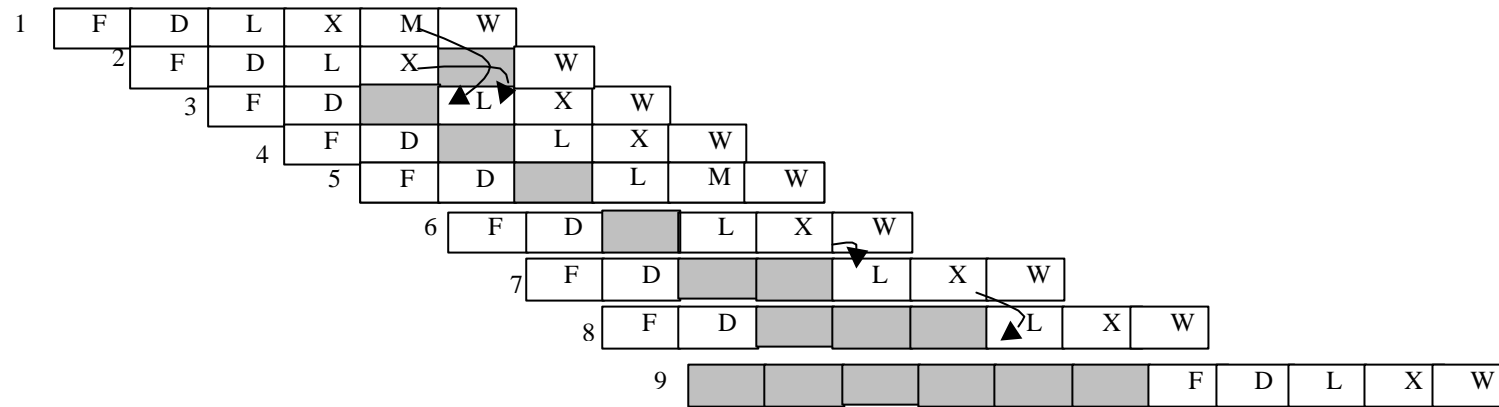


PROBLEMA 1

a)



b) RAW: Instrucció 1 i instrucció 3 (R1).

Instrucció 2 i instrucció 3 (R4).

Instrucció 6 i instrucció 7 (R5).

Instrucció 7 i instrucció 8 (R8).

Conflictes estructurals instrucció 1 i instrucció 2 en la escriptura (W).

c) 1 load r1(r2)

2 add r4,r2,r3

3 nop

4 sub r5,r1,r4

5 add r3,r2,r1
6 store r4,(r2)
7 add r5,r4,r2
8 nop
9 mult r8,r5,r7
10 nop
11 bne r8,salt
12 nop
13 nop
14 nop
15 sub r1,r7,r8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
MULT S1, S2, S3	F	D	X	X	X	X	W																	
MULT S1, S1, S3	F	D					X	X	X	X	W													
SUB S2, S4, S5		F	D	X	X	W																		
CMP S6, S2, 0		F	D			X	W																	
BNE S6,LL1			F	D			X	W			H	H	H	H										
ADD S1, S5, S2			F	D		X	X	W																
SUB S3, S4, S7				F	D		X	X	W															
MULT S2, S2, S3				F				D	X	X	X													
LL1: SUB S3, S2, S3								F	D							F	D	X	X	W				
MULT S4, S3, S1								F								F	D			X	X	X	X	W

Estacions de reserva i banc de registres el cicle:6

NOM	OCUPAT	OPER	VJ	QJ	VK	QK
+1	(3)	SUB	S4		S5	
+2	(6)	ADD	S5		S2	(+1)
+3	(7)	SUB	S4		S7	
*1	(1)	MULT	S2		S3	
*2	(2)	MULT	S1	(*1)	S3	
SEN1	(4)	CMP	S2	(+1)		
SEN2	(5)	BNE	S6	(SEN1)		

	S1	S2	S3	S4	S5	S6	S7
Vi							
Qi	(+2)	(+1)	(+3)			(SEN1)	

Estacions de reserva i banc de registres els cicle:8

NOM	OCUPAT	OPER	VJ	QJ	VK	QK
+1						
+2	(6)	ADD	S5		S2	
+3	(7)	SUB	S4		S7	
*1	(8)	MULT	S2		S3	(+3)
*2	(2)	MULT	S1	(*1)		
SEN1						
SEN2	(5)	BNE	S6			

	S1	S2	S3	S4	S5	S6	S7
Vi		(+1)					
Qi	(+2)	(*1)	(+3)			(SEN1)	

Estacions de reserves i banc de registres els cicles:11

NOM	OCUPAT	OPER	VJ	QJ	VK	QK
+1	(9)	SUB	S2	(*1)	S3	
+2						
+3						
*1	(8)	MULT	S2		S3	(+3)
*2	(2)	MULT	S1	(*1)		
SEN1						
SEN2						

	S1	S2	S3	S4	S5	S6	S7
Vi	(+2)	(+1)				(SEN1)	
Qi		(*1)					

Estacions de reserva i banc de registres els cicles:20

NOM	OCUPAT	OPER	VJ	QJ	VK	QK
+1	(9)	SUB	S2		S3	
+2						
+3						
*1	(10)	MULT	S3		S1	
*2						
SEN1						
SEN2						

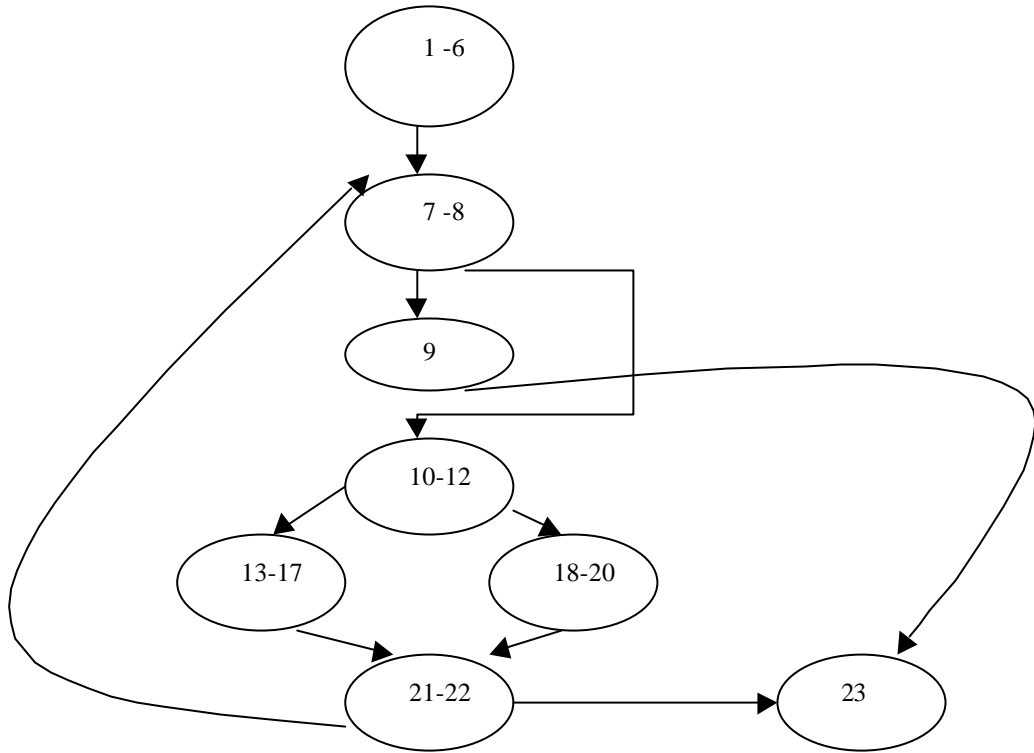
	S1	S2	S3	S4	S5	S6	S7
Vi							
Qi			(+1)	(*1)			

B) Quan realitza el SALT els registres es troben en estat arquitectònic mentre quan recupera després del SALT es trobarà en estat en ordre.

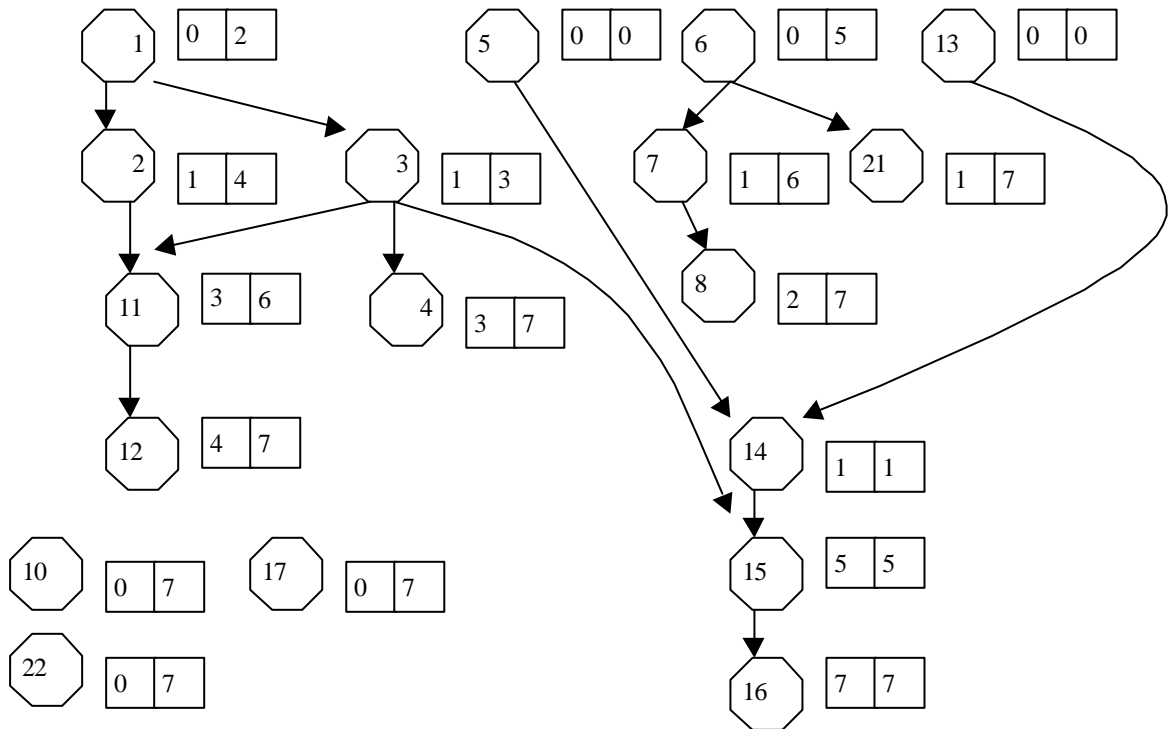
3) a)

```
1    load r1,(r12)0
2    add r0,r1,r2
3    add r1,r2,r1
4    store r1,(r12)0
5    mov r8, var_S3
6    mov r4,0
7    LL2: cmp r5,r4,5
8    ble r5,LL5
9    b LL3
10   LL5: load r13, (r12)0
11    cmp r7,r1,r0
12    beq r7,LL6
13    load r9, (r11)0
14    mult r3,r8,r9
15    add r6,r1,r3
16    store r6, (r12)0
17    b LL7
18   LL6: mult r6,r13,r0
19    add r13,r2,r13
20    store r13, (r12)0
21   LL7: load r13, (r12)0
22    cmp r7,r1,r0
23    beq r7,LL9
24    load r9, (r11)0
25    mult r3,r8,r9
26    add r6,r1,r3
27    store r6, (r12)0
28    b LL8
29   LL9: mult r6,r13,r0
30    add r13,r2,r13
31    store r13, (r12)0
32   LL8: add r4,1,r4
33    b LL2
34   LL3:
```

b)



c)



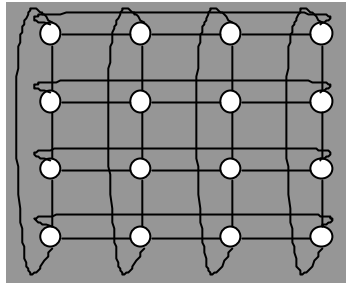
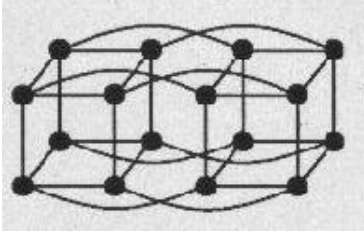
cicle	Llista	SEN	SALT	MUL	MEM	COM
1	1(2),5(0),6(5),10(7),13(0),17(7),22(7)		17		13	5
2	1(2),6(5),10(7),22(7),14(1)		22	14	1	6
3	10(7),2(4),3(3),7(6),21(7)	3			10	7
4	2(4),15(5),8(7),21(7)	2	8			
5	15(5),4(7),11(6),21(7)	21			4	
6	15(5),4(7),11(6)	15				11
7	16(7),12(7)		12			
8	16(7)				16	

1.- Volem construir un multicomputador amb 16 nodes. Per fer aquest disseny volem comparar dues topologies de xarxa d'interconnexió. Per una banda un hipercub i per l'altra un torus

(1,5 punts).

a) Fer el disseny de la xarxa hipercub i la xarxa torus 2D.

Solució :



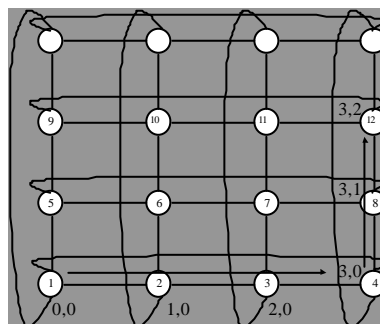
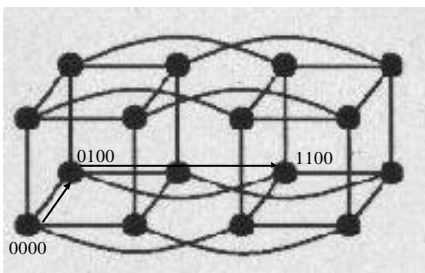
b) Calcular el grau (d), ampla de bisecció (b) i el diàmetre (d) per cadascuna de les xarxes.

Solució:

Tipus Xarxa	Grau (d)	Diàmetre (d)	Ampla Bisecció (b)
Hipercub	3	L 3	8
Torus (r*r)	4	2 (r / 2)	2r

c) Executar l'algorisme d'encaminament Dimension Order i XY (hipercub i xarxa torus respectivament) per trobar la ruta entre els nodes 1 i 12. Fer l'etiquetatge dels nodes i calcular la ruta.

Solució:



d) Si disposem d'enllaços de 2 GB, de quin ampla de banda disposem cada node. (suposem que cada node està connectat a un router)

Solució:

$$\text{Hipercub} = b * 2 = 8 * 2 = 16 \text{ GB} / 16 \text{ nodes} = 1 \text{GB} / \text{node}$$

$$\text{Torus 2D} = b * 2 = 8 * 2 = 16 \text{ GB} / 16 \text{ nodes} = 1 \text{GB} / \text{node}$$

2.- Disposem del següent codi :

(1 punt)

```

for(j=1;j<1000;j++)
{
    for(i=1;i<1000;i++)
    {
        for (k=1;k<1000;k++)
        {
            c[i][j] = ADD (a[i][k],b[k][j]);
            t = ADD (c[i][k],a[i]);
        }
    }
}
    
```

Indicar com afectaria l'aplicació de les següents tècniques d'optimització. Justificar quines es podem aplicar i quines no, i quines podrien arribar a millorar el rendiment. Al efectuar un perfex, tenim el següent perfil en les caches:

% perfex -e 25 multmat <- Per obtenir el número de falles de cahcé de L1

0 Cycles.....59002735555
 25 Primary data caché misses.....1338524786

% perfex -e 26 multmat <- Per obtenir les falles de caché de L2

0 Cycles.....59018761168
 26 Secondary data caché misses.....54515279

Temps d'execució 351 segons

Tècnica d'optimització	Justificació
SW PIPELINING	No és pot aplicar ja que tenim una dependència de dades al bucle.
1) Cache Blocking	OK pot millorar el rendiment. Especialment amb mides de bloc grans.
2) Loop fusion	No podem aplicar-la ja que no tenim bucles per fusionar.
3) Inlining	OK. Podem aplicar-la sobre la funció suma (ADD).
4) Loop Interchange	OK. Pot funcionar ja que tenim un bucle amb índexs que podem accedir de forma ineficient a les dades.

3.- Respon, justificant la resposta, les següents preguntes sobre PVM :

(1 punt)

a) Quina instrucció fem servir per inicialitzar un buffer (canal) per enviar dades. De forma que les dades es codifiquin en un format estàndar a dues arquitectures diferents ?. Com s'anomena el protocol que efectua la conversió de dades per tal de fer-la transparent al programador ?

Solució :

```
pvm_initsend (PvmDataDefault);  
XDR
```

b) Quin missatge d'error hauria d'aparèixer al 'printf' en el següent fragment de codi PVM:

```
int FILLS[2];  
pvm_spawn("TASCAFILLA", NULL, PvmTaskDefault, 0, 2,  
FILLS);  
  
if (pvm_send(FILLS[1], 2) < 0) {  
    printf("????? \n");  
    pvm_exit();  
}
```

Solució:

Problemes en la tasca PARE al enviar el buffer al FILL 2