

1. Considereu la funció:

```

for (i = 0; i < len; i++) {
    y[i] = a * x[i] + y[i]
}

```

Estudiarem el rendiment d'un RISC-V de 32 bits amb pipeline de 5 etapes davant d'aquest codi. La versió en assembler de la funció és la següent:


```

// x1 manté el punter a x
// x2 manté el punter a y
// x3 conté el valor a
// x4 conté el valor len
add x5, x0, x0
LOOP:    bge x5, x4, FET
         lw x6, 0(x1)
         mul x6, x6, x3
         lw x7, 0(x2)
         add x6, x6, x7
         sw x6, 0(x2)
         addi x1, x1, # 4
         addi x2, x2, # 4
         addi x5, x5, # 1
         j LOOP
FET:

```


FET:

a) Feu un diagrama multicicle per a l'execució de les primeres 12 instruccions dinàmiques, considerant que el nostre RISC-V té 5 etapes de pipeline i forwarding datapaths. Supposeu que no tenim problemes a l'obtenir dades de memòria i que $\text{len} > 2$. Considereu, també, que els salts incondicionals es resolen a l'etapa D i els salts condicionals a l'etapa EX.



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	add x5, x0, x0	IF	ID	EX	M	W															
2	bge x5, x4, FET		IF	ID	EX	M	W														
3	lw x6, 0(x1)			IF	ID	EX	M	W													
4	mul x6, x6, x3				-	IF	ID	EX	M	W											
5	lw x7, 0(x2)					IF	ID	EX	M	W											
6	add x6, x6, x7						-	IF	ID	EX	M	W									
7	sw x6, 0(x2)							IF	ID	EX	M	W									
8	addi x1, x1, 4								IF	ID	EX	M	W								
9	addi x2, x2, 4									IF	ID	EX	M	W							
10	addi x5, x5, 1										IF	ID	EX	M	W						
11	j LOOP											IF	ID	EX	M	W					
12	bge x5, x4, FET												-	IF	ID	EX	M	W			

b) Té sentit reordenar el codi ? si en té, torneu a omplir la taula multicicle amb les primeres 12 instruccions dinàmiques, aquest cop amb el codi reordenat.



		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	add x5, x0, x0	IF	ID	EX	M	W														
2	bge x5, x4, FET		IF	ID	EX	M	W													
3	lw x6, 0(x1)			IF	ID	EX	M	W												
4	lw x7, 0(x2)				IF	ID	EX	M	W											
5	mul x6, x6, x3					IF	ID	EX	M	W										
6	add x6, x6, x7						IF	ID	EX	M	W									
7	sw x6, 0(x2)							IF	ID	EX	M	W								
8	addi x1, x1, 4								IF	ID	EX	M	W							
9	addi x2, x2, 4									IF	ID	EX	M	W						
10	addi x5, x5, 1										IF	ID	EX	M	W					
11	j LOOP											IF	ID	EX	M	W				
12	bge x5, x4, FET												-	IF	ID	EX	M	W		