

Proyecto Camino de Datos

1.

Valores INICIALES almacenados en los registros

R0 : 1	R1: 2	R2: 0	R3: 0
R4: 0	R5: 3	R6: 0	R7: 3
R8: 2	R9: 0	R10: 0	R11: 10
R12: 0	R13: 5	R14: 0	R15: 0

valores FINALES almacenados en los registros

R0 : 4	R1: 2	R2: 0	R3: 6
R4: 3	R5: 2	R6: 8	R7: 2
R8: 3	R9: 9	R10: 8	R11: 0
R12: 5	R13: 1	R14: 0	R15: 0

Esquema de segmentación

----- tiempo ----->																						
Nº CICLOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
SUB R3, R1, R0	IF	ID	EX	MEM	WB																	
ADD R4, R3, R1		IF	IF	IF	ID	EX	MEM	WB														
BNE R4, R5, Eti					IF	IF	IF	ID	EX	MEM	WB											
ADD R3, R5, R7								IF	ID	EX	MEM	WB										
JUMP Eti2									IF	ID	EX	MEM	WB									
Eti ADD R3, R4, R7																						
Eti2 ADDi R8, R8, 1										IF	ID	EX	MEM	WB								
ADD R9, R3, R8											IF	IF	IF	ID	EX	MEM	WB					
SUB R12, R11, R13														IF	ID	EX	MEM	WB				
AND R14, R12, R11															IF	ID	ID	ID	EX	MEM	WB	

Riesgos

No se produce el salto de la 3 línea. Existen riesgos de datos en la 2, 3, 8 y 10 operación. Riesgos estructurales en la 2, 4 y 10 operación.

El código tarda en ejecutarse 21 ciclos.

2.

Optimizaría el código intercalando operaciones que no estén interrelacionadas con la anterior entre las que sí lo están, ahorrándome así los riesgos de datos, y también eliminando algunos riesgos estructurales. Siempre y cuando no varíe el resultado del programa.

Nº CICLOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
SUB R3, R1, R0	IF	ID	EX	MEM	WB																	
BNE R4, R5, Eti		IF	ID	EX	MEM	WB																
SUB R12, R11, R13			IF	ID	EX	MEM	WB															
ADD R4, R3, R1				IF	ID	ID	EX	MEM	WB													
JUMP Eti2						IF	ID	EX	MEM	WB												
Eti ADD R3, R4, R7																						
Eti2 ADDi R8, R8, 1							IF	ID	EX	MEM	WB											
ADD R9, R3, R8								IF	ID	EX	MEM	WB										
AND R14, R12, R11									IF	ID	EX	MEM	WB									
ADD R3, R5, R7										IF	ID	EX	MEM	WB								

Ahora tarda 14 ciclos y los valores finales son los mismos.

3.

Lo único que cambia es que no se ejecutaría el salto de etiqueta, lo que nos permite optimizar más el código. Ahora tengo una operación más para ejecutar entre las operaciones relacionadas que me quedaban, de esta forma aprovecho los ciclos de ejecución al máximo sin perder tiempo esperando para entrar en un proceso. Nos quedamos con un tiempo de ejecución de **14 ciclos**.

Nº CICLOS	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
SUB R3, R1, R0	IF	ID	EX	MEM	WB																	
ADD R4, R3, R1		IF	IF	IF	ID	EX	MEM	WB														
BNE R4, R5, Eti					IF	IF	IF	ID	EX	MEM	WB											
ADD R3, R5, R7																						
JUMP Eti2																						
Eti ADD R3, R4, R7								IF	IF	IF	ID	EX	MEM	WB								
Eti2 ADDi R8, R8, 1											IF	ID	EX	MEM	WB							
ADD R9, R3, R8												IF	IF	IF	ID	EX	MEM	WB				
SUB R12, R11, R13															IF	ID	EX	MEM	WB			
AND R14, R12, R11																IF	ID	ID	ID	EX	MEM	WB

Sobre el código original, al cambiar los valores iniciales a estos, r4 y r5 dejan de tener el mismo valor y por tanto se produce el salto a eti2, por lo que cambia el proceso anterior y **pasamos a tardar un total de 22 ciclos. No haríamos ninguna optimización con esos valores.**