

Tarea 7

P1

a) La tabla completa quedaría:

Ciclo	Fet	Dec	An	Exe	Mem	Sto
1	ab					
2	cd	ab				
3	ef	cd	ab			
4	%	%	%	a		
5	%	%	%	a		
6	%	%	%	a		
7	%	%	%	a		
8	%	%	%	a		
9	%	%	%	a		
10	%	%	%	a		
11	%	%	%	a		
12	a'b'	ef	cd	b		a
13	%	%	%	c		b
14	c'd'	a'b'	ef	d		
15	%	%	%	e		d
16	e'f'	c'd'	a'b'	f		e
17	%	%	%	a'		
18	%	%	%	a'		
19	%	%	%	a'		
20	%	%	%	a'		
21	%	%	%	a'		
22	%	%	%	a'		
23	%	%	%	a'		
24	%	%	%	a'		
25		e'f'	c'd'	b'		a'
26		%	%	c'		b'
27			e'f'	d'		
28			%	e'		d'
29				f'		e'

Respecto a la efectividad de esta arquitectura versus la de un solo pipeline, podemos ver se ejecutan las instrucciones una a la vez, por lo que se requieren la misma cantidad de ciclos para ejecutar las dos iteraciones en cualquiera de las dos arquitecturas. Esto se debe a las dependencias que existen entre cada parte del código.

b) Ya que g necesita R2 para ser ejecutada y h modifica el valor de R2, se genera una dependencia que hará que la cantidad de ciclos para ejecutar las dos iteraciones sea nuevamente 29 ya que g se deberá hacer antes que h. Aún así, si h se ejecutara en el último

ciclo de g, o si g necesitara R2 sólo en el primer ciclo, se podría reducir en 1 la cantidad de ciclos por iteración al ejecutar g y h en paralelo, es decir, las dos iteraciones se harían en 27 ciclos. La tabla del primer supuesto es prácticamente idéntica a la de la parte a), mientras que la tabla del segundo supuesto quedaría:

Ciclo	Fet	Dec	An	Exe	Mem	Sto
1	gh					
2	ij					
3	kl	ij	gh			
4	%	%	%	g		
5	%	%	%	g		
6	%	%	%	g		
7	%	%	%	g		
8	%	%	%	g		
9	%	%	%	g		
10	%	%	%	g		
11	g'h'	kl	ij	gh		
12	%	%	%	i		gh
13	i'j'	g'h'	kl	j		i
14	%	%	%	k		
15	k'l'	i'j'	g'h'	l		k
16	%	%	%	g'		
17	%	%	%	g'		
18	%	%	%	g'		
19	%	%	%	g'		
20	%	%	%	g'		
21	%	%	%	g'		
22	%	%	%	g'		
23		k'l'	i'j'	g'h'		
24		%	%	i'		g'h'
25			k'l'	j'		i'
26			%	k'		
27				l'		k'

P2

La tabla sería:

Ciclo	Fet	Dec	An	Exe	Mem	Sto	Ret
1	ad						
2	ef	ad					
3	a'b'	ef	ad				
4	e'f'	a'b'	ef	a			
5		e'f'	a'b'	a			
6			e'f'	a			
7				a			
8				a			
9				a			
10				a			
11				ad			
12				e		ad	
13				f		e	ad
14				a'			ef
15				a'			
16				a'			
17				a'			
18				a'			
19				a'			
20				a'			
21				a'd'			
22				e'		a'd'	
23				f'		e'	a'd'
24							e'f'

El cuello de botella se encuentra al sumarle 1 a k, ya que k se almacena en R2 y es ocupado en la instrucción a, por lo que no se puede avanzar a e hasta que se termine a. Si se solucionara este tema, se podrían ejecutar e y f mientras se termina a.