

Tarea 6

CC3301

Luciano Márquez

Parte C:

Según el siguiente set de instrucciones Risc-V:

A sub a3,s5,t2
B add a5,t2,s4
C andi a3,a3,255
D addi a3,a3,1
E ori a5,a5,15
F bgt a3,s1,R
G add ...
H sub ...
I xor ...
J andi ...
...
R sub a3,a3,a5
S ori a3,a3,255

Los diagramas que describen los ciclos ejecutados en cada etapa, según la arquitectura, son los siguientes:

- Arquitectura en pipeline, con etapas *fetch*, *decode* y *execute*:

Ciclo	<i>Fetch</i>	<i>Decode</i>	<i>Execute</i>
1	A		
2	B	A	
3	C	B	A
4	D	C	B
5	E	D	C
6	F	E	D
7	G	F	E
8	H	G	F
9	R		
10	S	R	
11		S	R
12			S

Notar que las instrucciones **G** y **H** empiezan su procesamiento en el pipeline, pero debido al salto que produce la instrucción **F**, el trabajo previo hecho para ellas es anulado.

- Arquitectura superescalar, con dos pipelines con etapas *fetch*, *decode* y *execute*:

Ciclo	<i>Fetch</i>	<i>Decode</i>	<i>Execute</i>
1	A B		
2	C D	A B	
3	E F	C D	A B
4	E F	D	C
5	G H	E F	D
6	I J	G H	E F
7	R S		
8		R S	
9		S	R
10			S

Para este tipo de arquitectura es importante que en la etapa de *execute* se tengan en cuenta las dependencias de las instrucciones, por ejemplo, en la instrucción C cambia el valor del registro *a3* y el resultado de la instrucción D depende de este registro, dado esta dependencia estas no pueden ser ejecutadas al mismo tiempo, por lo que la operación D se “queda atrás” en el pipeline, bloqueándolo hasta que se puede ejecutar sin problemas. Esto también ocurre con las instrucciones R y S. También dado el salto que se realiza en la instrucción F, se pierden el trabajo anteriormente hecho para las instrucciones G, H, I y J.