Instruction		Iteration	Instruction status		
			Issue	Execute	Write result
L:	1> LDURD D0, [X1, #0]		☑	☐	☐
	2> FMULD D4, D0, D2		☑	☐	☐
	3> STURD D4, [X1, #0]		☐	☐	☐
	4> SUBI X1, X1, #8		☐	☐	☐
	5> CBNZ X1, L		☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Hardware
I salto se predice correctamente como taken.
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk
ir hasta el 8vo clock

Name	Reservation stations						
	Busy	Op	Vj	Vk	Qj	Qk	A
load 1	☐	load	[X1]	-	0	-	#0
load 2	☐						
load 3	☐						
store 1	☐						
store 2	☐						
store 3	☐						
Int ALU 1	☐						
int ALU 2	☐						
int ALU 3	☐						
FP alu 1	☐						-
FP alu 2	☐						
FP mult 1	☐	mult		[D2]	load 1	0	-
FP mult 1	☐						
Branch 1	☐						
	☐						
	☐						

		D0	D1	D2	D3	D4	D5	D6	D7	D8
						Register Status				
Qi		load 1				FP mult 1				
		X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi										

	Instruction	Iteration	Instruction status				Hardware
			Issue	**Execute**	**Write result**		
L:	1> LDURD D0, [X1, #0]		☑	☐	☐		I salto se predice correctamente como taken.
	2> FMULD D4, D0, D2		☑	☐	☐		Issue = 2 instrucciones
	3> STURD D4, [X1, #0]		☑	☐	☐		Load = 6 RS / 1 clk
	4> SUBI X1, X1, #8		☑	☐	☐		Store = 6 RS / 1 clk
	5> CBNZ X1, L		☐	☐	☐		Alu entera = 3 RS / 1 clk
			☐	☐	☐		Alu punto flotante = 3 RS / 2 clk
			☐	☐	☐		Multiplicación punto flotante = 2 RS / 3 clk
			☐	☐	☐		Branch = 1 RS, 1 FU / 1clk
			☐	☐	☐		
			☐	☐	☐		ir hasta el 8vo clock
			☐	☐	☐		

	Name	Reservation stations							
		Busy	**Op**	**Vj**	**Vk**	**Qj**	**Qk**	**A**	
	load 1	☐	load	[X1]	-		0	0	#0 + [x1]	durante este clock calcula la addr
	load 2	☐							
	load 3	☐							
	store 1	☐	store	[x1]	-		0	FP mult 1	#0	guardo en addr el offset, calculo addres en el clock siguiente
	store 2	☐							
	store 3	☐							
	Int ALU 1	☐	sub	[X1]	#8		0	0	-
	int ALU 2	☐							
	int ALU 3	☐							
	FP alu 1	☐						-	
	FP alu 2	☐							
	FP mult 1	☐	mult		[D2]	load 1		0	-	este se queda esperando load 1
	FP mult 1	☐							
	Branch 1	☐							
		☐							
		☐							

Register Status									
		D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi		load 1				FP mult 1				
		X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi			int ALU 1							

Instruction status

Instruction		Iteration	Issue	Execute	Write result
L:	1> LDURD D0, [X1, #0]		☑	☑	☐
	2> FMULD D4, D0, D2		☑	☐	☐
	3> STURD D4, [X1, #0]		☑	☐	☐
	4> SUBI X1, X1, #8		☑	☑	☐
	5> CBNZ X1, L		☑	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Hardware

I salto se predice correctamente como taken.

Issue = 2 instrucciones

Load = 6 RS / 1 clk

Store = 6 RS / 1 clk

Alu entera = 3 RS / 1 clk

Alu punto flotante = 3 RS / 2 clk

Multiplicación punto flotante = 2 RS / 3 clk

Branch = 1 RS, 1 FU / 1clk

ir hasta el 8vo clock

Reservation stations

Name	Busy	Op	Vj	Vk	Qj	Qk	A	
load 1	☑	load	[X1]	-	0	0	#0 + [x1]	se calcula la addr, ahora pasa a execute -> busy
load 2	☐							
load 3	☐							
store 1	☐	store	[x1]	-	0	FP mult 1	#0 + [x1]	calculo addr
store 2	☐							
store 3	☐							
Int ALU 1	☑	sub	[X1]	#8	0	0	-	me voy a execute
int ALU 2	☐							
int ALU 3	☐							
FP alu 1	☐					-		
FP alu 2	☐							
FP mult 1	☐	mult		[D2]	load 1	0	-	este se queda esperando load 1
FP mult 2	☐							
Branch 1	☐	cbnz	-	# -4 (salto)	Int ALU 1	0	-	Hastta que esto no termina no hago issue de nada
	☐							
	☐							

Register Status

	D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi	load 1				FP mult 1				

	X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi		int ALU 1							

Instruction status

	Instruction	Iteration	Issue	Execute	Write result
L:	1> LDURD D0, [X1, #0]		☑	☑	☑
	2> FMULD D4, D0, D2		☑	☐	☐
	3> STURD D4, [X1, #0]		☑	☐	☐
	4> SUBI X1, X1, #8		☑	☑	☑
	5> CBNZ X1, L		☑	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Hardware

Hardware
I salto se predice correctamente como taken.
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk
ir hasta el 8vo clock

Reservation stations

Name	Busy	Op	Vj	Vk	Qj	Qk	A	
load 1	☐							Termino, ahora escribo
load 2	☐							
load 3	☐							
store 1	☐	store	[x1]	-	0	FP mult 1	#0 + [x1]	ya calcule addr, ahora espero a resultado
store 2	☐							
store 3	☐							
Int ALU 1	☐							termino de ejecutar
int ALU 2	☐							
int ALU 3	☐							
FP alu 1	☐						-	
FP alu 2	☐							
FP mult 1	☐	mult	[D0]	[D2]	0	0	-	espero que se termine de escribir
FP mult 2	☐							
Branch 1	☐	cbnz	[x1]	# -4 (salto)	0	0	-	Hasta que esto no termina no hago issue de nada
	☐							
	☐							

Register Status

Qi	D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi					FP mult 1				

	X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi									

Instruction status

	Instruction	Iteration	Issue	Execute	Write result
L:	1> LDURD D0, [X1, #0]		☑	☑	☑
	2> FMULD D4, D0, D2		☑	☑	☐
	3> STURD D4, [X1, #0]		☑	☐	☐
	4> SUBI X1, X1, #8		☑	☑	☑
	5> CBNZ X1, L		☑	☑	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Hardware

I salto se predice correctamente como taken.
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk
ir hasta el 8vo clock

Reservation stations

Name	Busy	Op	Vj	Vk	Qj	Qk	A	
load 1	☐							termine de escribir
load 2	☐							
load 3	☐							
store 1	☐	store	[x1]	-	0	FP mult 1	#0 + [x1]	ya calcule addr, ahora espero a resultado
store 2	☐							
store 3	☐							
Int ALU 1	☐							termine de escribir
int ALU 2	☐							
int ALU 3	☐							
FP alu 1	☐						-	
FP alu 2	☐							
FP mult 1	☑	mult	[D0]	[D2]	0	0	-	termine de escribir, puedo ejecutar
FP mult 2	☐							
Branch 1	☑	cbnz	[x1]	# -4 (salto)	0	0	-	Hasta que esto no termina no hago issue de nada
	☐							
	☐							

Register Status

	D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi					FP mult 1				
	X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi									

Instruction status

	Instruction	Iteration	Issue	Execute	Write result
L:	1> LDURD D0, [X1, #0]		☑	☑	☑
	2> FMULD D4, D0, D2		☑	☑	☐
	3> STURD D4, [X1, #0]		☑	☐	☐
	4> SUBI X1, X1, #8		☑	☑	☑
	5> CBNZ X1, L		☑	☑	☑
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐
			☐	☐	☐

Hardware

Hardware
I salto se predice correctamente como taken.
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk
ir hasta el 8vo clock

Reservation stations

Name	Busy	Op	Vj	Vk	Qj	Qk	A	
load 1	☐							
load 2	☐							
load 3	☐							
store 1	☐	store	[x1]	-	0	FP mult 1	#0 + [x1]	ya calcule addr, ahora espero a resultado
store 2	☐							
store 3	☐							
Int ALU 1	☐							
int ALU 2	☐							
int ALU 3	☐							
FP alu 1	☐							
FP alu 2	☐							
FP mult 1	☑	mult	[D0]	[D2]	0	0	-	2/3 clocks
FP mult 2	☐							
Branch 1	☐							termino, escribo durante un clock
	☐							
	☐							

Register Status

	D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi					FP mult 1				
	X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi									

	Instruction	Iteration	Instruction status			Hardware	
			Issue	**Execute**	**Write result**		
L:	1> LDURD D0, [X1, #0]		☑	☑	☑	I salto se predice correctamente como taken.	
	2> FMULD D4, D0, D2		☑	☑	☐	Issue = 2 instrucciones	
	3> STURD D4, [X1, #0]		☑	☐	☐	Load = 6 RS / 1 clk	
	4> SUBI X1, X1, #8		☑	☑	☑	Store = 6 RS / 1 clk	
	5> CBNZ X1, L		☑	☑	☑	Alu entera = 3 RS / 1 clk	
	1> LDURD D0, [X1, #0]	2	☑	☐	☐	Alu punto flotante = 3 RS / 2 clk	
	2> FMULD D4, D0, D2	2	☑	☐	☐	Multiplicación punto flotante = 2 RS / 3 clk	
	3> STURD D4, [X1, #0]	2	☐	☐	☐	Branch = 1 RS, 1 FU / 1clk	
	4> SUBI X1, X1, #8	2	☐	☐	☐	ir hasta el 8vo clock	
	5> CBNZ X1, L	2	☐	☐	☐		

	Name	Reservation stations						
		Busy	**Op**	**Vj**	**Vk**	**Qj**	**Qk**	**A**
	load 1	☐	load	[x1]	-	0	-	#0
	load 2	☐						
	load 3	☐						
	store 1	☐	store	[x1]	-	0	FP mult 1	#0 + [x1]	ya calcule addr, ahora espero a resultado
	store 2	☐						
	store 3	☐						
	Int ALU 1	☐						
	int ALU 2	☐						
	int ALU 3	☐						
	FP alu 1	☐						
	FP alu 2	☐						
	FP mult 1	☑	mult	[D0]	[D2]	0	0	-	3/3 clocks, proximo es WB
	FP mult 2	☐	mult	-	[D2]	load 1	0	-
	Branch 1	☐							termine de escribir, puedo hacer issue
		☐						
		☐						

ya calcule addr, ahora espero a resultado

3/3 clocks, proximo es WB

termine de escribir, puedo hacer issue

	Register Status								
		D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi		load 1				FP mult 2				
		X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi										

Instruction status

Instruction	Iteration	Issue	Execute	Write result
L: 1> LDURD D0, [X1, #0]	1	☑	☑	☑
2> FMULD D4, D0, D2	1	☑	☑	☑
3> STURD D4, [X1, #0]	1	☑	☐	☐
4> SUBI X1, X1, #8	1	☑	☑	☑
5> CBNZ X1, L	1	☑	☑	☑
1> LDURD D0, [X1, #0]	2	☑	☐	☐
2> FMULD D4, D0, D2	2	☑	☐	☐
3> STURD D4, [X1, #0]	2	☑	☐	☐
4> SUBI X1, X1, #8	2	☑	☐	☐
5> CBNZ X1, L	2	☐	☐	☐

Hardware

I salto se predice correctamente como taken.
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk
ir hasta el 8vo clock

Reservation stations

Name	Busy	Op	Vj	Vk	Qj	Qk	A	
load 1	☐	load	[x1]	-	0	-	#0 + [x1]	calculo addr
load 2	☐							
load 3	☐							
store 1	☐	store	[x1]	[D4]	0		#0 + [x1]	ya calcule addr, ahora espero a resultado
store 2	☐	store	[x1]	-	0	FP mult 2	#0	
store 3	☐							
Int ALU 1	☐	sub	[x1]	#8	0	0	-	
int ALU 2	☐							
int ALU 3	☐							
FP alu 1	☐							
FP alu 2	☐							
FP mult 1	☐	mult	[D0]	[D2]	0	0	-	Ahora hago wb
FP mult 2	☐	mult	-	[D2]	load 1	0	-	
Branch 1	☐							
	☐							
	☐							

Register Status

	D0	D1	D2	D3	D4	D5	D6	D7	D8
Qi	load 1				FP mult 2				
	X0	X1	X2	X3	X4	X5	X6	X7	X8
Qi		int ALU 1							