Instruction				Instruction stat	tus			Hardware		
Ins	truction	Iteration	Issue	Execute	Write result		I salto se pred	lice correctamente	como taken.	
1> LDURD DO	), [X1, #0]		<b>✓</b>				Issue = 2 insti	rucciones		
2> FMULD D4	l, D0, D2		~				Load = 6 RS /	1 clk		
3> STURD D4	l, [X1, #0]						Store = 6 RS	/ 1 clk		
4> SUBI X1, X	(1, #8						Alu entera = 3	u entera = 3 RS / 1 clk		
5> CBNZ X1,	L						Alu punto flota	ante = 3 RS / 2 clk		
							Multiplicación	punto flotante = 2 F	RS / 3 clk	
							Branch = 1 RS	S, 1 FU / 1clk		
							ir hasta el 8vo	clock		
Name				_						
		Busy	Ор	Vj R	eservation station	ons Qj				
load 1	- Turne	Busy	load	<u> </u>	VK		<b>Qk</b>	<b>A</b>		
load 2			loau	[X1]	-		) - 	#0		
load 2						1				
store 1										
store 2										
store 3										
Int ALU 1		<del>                                     </del>								
int ALU 2		<del>                                     </del>								
int ALU 3										
FP alu 1								_		
FP alu 2		<del>                                     </del>								
FP mult 1			mult		[D2]	load 1		0 -		
FP mult 1					-					
Branch 1										
				Regist	er Status					
					F .					
0:	D0	D1	D2	D3	D4	D5	D6	D7	D8	
Qi	D0 load 1 X0	D1 X1	D2 X2		D4 FP mult 1 X4	D5 X5	D6	D7 X7	D8	

					Instruction statu	ıs			Hardware				
	Instr	uction	Iteration	Issue	Execute	Write result		I salto se predic	e correctamente	como taken.			
L:	1> LDURD D0, [	X1, #0]		~				Issue = 2 instruc	cciones				
	2> FMULD D4, I	D0, D2		<b>✓</b>				Load = 6 RS / 1	clk				
	3> STURD D4, [	X1, #0]		<b>✓</b>				Store = 6 RS / 1	clk				
	4> SUBI X1, X1,	, #8		<b>✓</b>				Alu entera = 3 F	RS / 1 clk				
	5> CBNZ X1, L							Alu punto flotan	te = 3 RS / 2 clk				
								Multiplicación po	unto flotante = 2	RS / 3 clk			
								Branch = 1 RS,	1 FU / 1clk				
								ir hasta el 8vo c	lock				
				1		servation station							
	Na	ime	Busy	Op	Vj	Vk	Qj	Qk	Α				
	load 1			load	[X1]	-	0	(	) #0 + [x1]	durante este clo	ck calcula la addr		
	load 2												
	load 3												
	store 1			store	[x1]	-	0	FP mult 1	#0	guardo en addr e	el offset, calculo add	dres en el clock	siguiente
	store 2		<u> </u>										
	store 3		<u> </u>										
	Int ALU 1		<u> </u>	sub	[X1]	#8	0	(	0 -				
	int ALU 2		$\vdash$	1									
	int ALU 3		$\vdash$	1									
	FP alu 1		$\vdash$						-				
	FP alu 2		$\vdash$										
	FP mult 1		$\vdash$	mult		[D2]	load 1	(	) -	este se queda es	sperando load 1		
	FP mult 1												
	Branch 1												
				1									
					Dogiota	er Status							
		D0	D1	D2	D3	D4	D5	D6	D7	D8			
	Qi	load 1	וט	DZ	D3	FP mult 1	D5	טט	וט	D0			
	3(1	X0	X1	X2	X3	X4	X5	X6	X7	X8			
	Qi	7.0	int ALU 1	//2	7.0	/ / /	7.0	7.0		7.0			
	Φ.		1		1								

					Instruction stat	us			Hardware			
	Instru	uction	Iteration	Issue	Execute	Write result		I salto se predice	correctamente	como taken.		
:	1> LDURD D0, [	[X1, #0]		<b>✓</b>	<b>~</b>			Issue = 2 instruc	ciones			
	2> FMULD D4, [	D0, D2		<b>&gt;</b>				Load = 6 RS / 1	clk			
	3> STURD D4, [	[X1, #0]		<b>✓</b>				Store = 6 RS / 1	clk			
	4> SUBI X1, X1,	, #8		<b>✓</b>	<b>✓</b>			Alu entera = 3 R	S / 1 clk			
	5> CBNZ X1, L			<b>✓</b>				Alu punto flotant	e = 3 RS / 2 clk			
								Multiplicación pu	into flotante = 2	RS / 3 clk		
								Branch = 1 RS,	1 FU / 1clk			
								ir hasta el 8vo cl	ock			
				1		eservation stati		_	1			
		ame	Busy	Op	Vj	Vk	Qj	Qk	Α			
	load 1		<u> </u>	load	[X1]	-	(	0	#0 + [x1]	se calcula la add	Ir, ahora pasa a exe	cute -> busy
	load 2		<u> </u>									
	load 3			1.								
	store 1			store	[x1]	-	1	FP mult 1	#0 + [x1]	calculo addr		
	store 2			1				1				
	store 3			ab	IV41	40	ļ ,					
	Int ALU 1			sub	[X1]	#8		0	-	me voy a execut	е	
	int ALU 3											
	FP alu 1								_			
	FP alu 2			1				+				
	FP mult 1			mult		[D2]	load 1	0	_	este se queda es	sperando load 1	
	FP mult 2									4		
	Branch 1			cbnz	-	# -4 (salto)	Int ALU 1	0	-	Hastta que esto	no termina no hago	issue de nada
					Registe	er 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 stat	us			Hardware	1		
	Instru	uction	Iteration	Issue	Execute	Write result		I salto se predio	ce correctament	e como taken.		
1>	> LDURD D0, [	X1, #0]		<b>✓</b>	<b>✓</b>	<b>✓</b>		Issue = 2 instru	ıcciones			
2>	> FMULD D4, [	D0, D2		~				Load = 6 RS /	1 clk			
3>	> STURD D4, [	X1, #0]		<b>✓</b>				Store = 6 RS /	1 clk			
4>	> SUBI X1, X1,	#8		<b>✓</b>	<b>✓</b>	~		Alu entera = 3	RS / 1 clk			
5>	> CBNZ X1, L			~				Alu punto flotar	nte = 3 RS / 2 cll	<		
								Multiplicación p	ounto flotante = 2	2 RS / 3 clk		
								Branch = 1 RS	, 1 FU / 1clk			
								ir hasta el 8vo	clock			
					Re	eservation station	ns					
	Na	ime	Busy	Op	Vj	Vk	Qj	Qk	Α			
lo	oad 1									Termino, ahora	escribo	
lo	oad 2											
lo	oad 3											
st	tore 1			store	[x1]	-	(	FP mult 1	#0 + [x1]	ya calcule addr,	ahora espero a res	sultado
st	tore 2											
st	tore 3											
In	nt ALU 1									termino de ejec	utar	
in	nt ALU 2											
in	nt ALU 3											
FI	P alu 1								-			
FI	P alu 2											
	P mult 1			mult	[D0]	[D2]	(	0	0 -	espero que se t	ermine de escribir	
FI	P mult 2											
Ві	Branch 1			cbnz	[x1]	# -4 (salto)	(	D .	0 -	Hasta que esto	no termina no hago	issue de nada
		1	I			er Status						
		D0	D1	D2	D3	D4	D5	D6	D7	D8		
	Qi					FP mult 1						
		X0	X1	X2	Х3	X4	X5	X6	X7	X8		
	Qi											

				Instruction stat	tus			Hardware	,	
Instr	uction	Iteration	Issue	Execute	Write result		I salto se pred	ice correctament	e como taken.	
1> LDURD D0, [	[X1, #0]		<b>✓</b>	<b>✓</b>	<b>✓</b>		Issue = 2 instr	ucciones		
2> FMULD D4, I	D0, D2		<b>✓</b>	<b>✓</b>			Load = 6 RS /	1 clk		
3> STURD D4, [	[X1, #0]		~				Store = 6 RS /	1 clk		
4> SUBI X1, X1,	, #8		~	<b>✓</b>	~		Alu entera = 3	RS / 1 clk		
5> CBNZ X1, L			~	<b>✓</b>			Alu punto flota	nte = 3 RS / 2 cll	(	
							Multiplicación	punto flotante = 2	2 RS / 3 clk	
							Branch = 1 RS	S, 1 FU / 1clk		
							ir hasta el 8vo	clock		
				R	eservation statio	ns				
Na	ame	Busy	Op	Vj	Vk	Qj	Qk	Α		
load 1									termine de escri	ibir
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 escri	ibir
int ALU 2										
int ALU 3										
FP alu 1								-		
FP alu 2										
FP mult 1		<b>✓</b>	mult	[D0]	[D2]		0	0 -	termine de escri	ibir, puedo ejecutar
FP mult 2										
Branch 1		<b>✓</b>	cbnz	[x1]	# -4 (salto)		0	0 -	Hasta que esto	no termina no hago issue de nada
					ter Status		_		1	
	D0	D1	D2	D3	D4	D5	D6	D7	D8	
Qi					FP mult 1					
	X0	X1	X2	X3	X4	X5	X6	X7	X8	
Qi										

				lı lı	nstruction statu	ıs			Hardware			
	Instru	ction	Iteration	Issue	Execute	Write result		I salto se predic	e correctamente	como taken.		
.:	1> LDURD D0, [>	<1, #0 <u>]</u>		<b>✓</b>	<b>✓</b>	<b>~</b>		Issue = 2 instru	cciones			
	2> FMULD D4, D	0, D2		~	~			Load = 6 RS / 1	clk			
	3> STURD D4, [X	(1, #0]		~				Store = 6 RS /	1 clk			
	4> SUBI X1, X1,	#8		<b>✓</b>	<b>✓</b>	<b>✓</b>		Alu entera = 3 F	RS / 1 clk			
	5> CBNZ X1, L			~	<b>✓</b>	~		Alu punto flotan	te = 3 RS / 2 clk			
								Multiplicación p	unto flotante = 2	RS / 3 clk		
								Branch = 1 RS,	1 FU / 1clk			
								ir hasta el 8vo d	clock			
						servation statio			_			
	Na	me	Busy	Op	Vj	Vk	Qj	Qk	Α			
	load 1											
	load 2											
	load 3											
	store 1			store	[x1]	-	0	FP mult 1	#0 + [x1]	ya calcule addr,	ahora espero a res	ultado
	store 2											
	store 3											
	Int ALU 1		Ц									
	int ALU 2		Ц									
	int ALU 3		Ц									
	FP alu 1		Ц									
	FP alu 2											
	FP mult 1		<u> </u>	mult	[D0]	[D2]	0	(	0 -	2/3 clocks		
	FP mult 2											
	Branch 1									termino, escribo	durante un clock	
					Desire.	Otatus						
		D0	D1	D2	D3	er Status D4	D5	D6	D7	D8		
	Qi	טט	וט	D2	DS	FP mult 1	פט	ם סט	D/	Do		
	نو	νο.	V1				VE	l ve	V7			
	Qi	X0	X1	X2	X3	X4	X5	X6	X7	X8		
	Ų								1			

					Instruction stat	us			Hardware			
	Instr	ruction	Iteration	Issue	Execute	Write result		I salto se pred	lice correctamente	e como taken.		
:	1> LDURD D0,	[X1, #0]		~	~	~		Issue = 2 insti	rucciones			
	2> FMULD D4,	D0, D2		~	~			Load = 6 RS	1 clk			
	3> STURD D4,	[X1, #0]		~				Store = 6 RS	/ 1 clk			
	4> SUBI X1, X1	, #8		~	<b>✓</b>	<b>✓</b>		Alu entera = 3	3 RS / 1 clk			
	5> CBNZ X1, L			<b>✓</b>	<b>~</b>	<b>&gt;</b>		Alu punto flota	ante = 3 RS / 2 clk			
	1> LDURD D0,	[X1, #0]	2					Multiplicación	punto flotante = 2	RS / 3 clk		
	2> FMULD D4,	D0, D2	2	2				Branch = 1 R	S, 1 FU / 1clk			
	3> STURD D4,	[X1, #0]	2	2								
	4> SUBI X1, X1	, #8	2					ir hasta el 8vo	clock			
	5> CBNZ X1, L		2									
					Re	eservation stati	ons					
	Na	ame	Busy	Op	Vj	Vk	Qj	Qk	Α			
	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 resul	tado
	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, proxi	mo es WB	
	FP mult 2			mult	-	[D2]	load 1		0 -			
	Branch 1									termine de escrit	oir, puedo hacer issu	е
						er Status	T					
		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 stat	us			Hardware	ı	
Inst	truction	Iteration	Issue	Execute	Write result		I salto se pred	dice correctament	e como taken.	
1> LDURD D0	, [X1, #0]		1	<b>✓</b>	<b>✓</b>		Issue = 2 inst	rucciones		
2> FMULD D4	, D0, D2		1	<b>✓</b>	<b>✓</b>		Load = 6 RS	/ 1 clk		
3> STURD D4	, [X1, #0]		1				Store = 6 RS	/ 1 clk		
4> SUBI X1, X	(1, #8		1	~	~		Alu entera = 3	3 RS / 1 clk		
5> CBNZ X1, I	L		1	~	~		Alu punto flot	ante = 3 RS / 2 cll	(	
1> LDURD D0	, [X1, #0]		2				Multiplicación	punto flotante = 2	2 RS / 3 clk	
2> FMULD D4	, D0, D2		2				Branch = 1 R	S, 1 FU / 1clk		
3> STURD D4	, [X1, #0]		2							
4> SUBI X1, X	(1, #8		2				ir hasta el 8vo	clock		
5> CBNZ X1, I	L		2							
				R	eservation stati					
N	Name	Busy	Op	Vj	Vk	Qj	Qk	Α		
load 1			load	[x1]	-		0 -	#0 + [x1]	calculo addr	
load 2										
load 3										
store 1			store	[x1]	[D4]		0	#0 + [x1]	ya calcule addr, a	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										
	1				er 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								