

18 NOV 2020

JUEGO 2 - HILOS PRUEBA PROYECTO CI0120-2020-2-SEM RISC-V Ë procesador 1 núcleo caché Inst y Datos con capacidad de 4 bloques de 2 palabras - BUFFER VÍCTIMA 8 BLOQUES

SE TRABAJA CON VALOR 1 COMO VALOR INICIAL EN MEMORIA DE DATOS

El hilo 1 almacena en memoria 64 valores desde 5, y de 5 en 5 hasta el 320 en memoria a partir de posición 0 de memoria hasta posición 252 (bloque 0 hasta bloque 31 inclusive)

El hilo 2 almacena en memoria 68 valores desde -800, y restaándole 2 cada vez (-800 hasta $-800 + (-2 * 67) = -800 + -134 = -934$ en memoria a partir de posición 0 de memoria hasta posición 268 (bloque 0 hasta bloque 33)

NOTA IMPORTANTE: Se incluye la **dirección en la memoria de instrucciones** para cada instrucción de cada hilillo, pero esta no necesariamente coincidirá con la suya.

Bl. Inst	Dir mem Inst	El hilo 1 almacena en memoria 64 valores desde 5, y de 5 en 5 hasta el 320 en memoria a partir de posición 0 de memoria 0 hasta posición 252 (bloque 0 hasta bloque 31 inclusive)					H
		codificado	etiq.	instrucción		resultado	
48	384	19 20 0 5		addi	x20, x0, 5	valor inicial a escribir a memoria (de 5 a 64*5 = 320)	H I L O 1
	388	19 3 0 32		addi	x3, x0, 32	Va a escribir 64 valores seguidos en memoria a partir de posición 0 (ciclo se hace 32 veces, pues escribe de dos en dos)	
49	392	19 8 0 8		addi	x8, x0, 8	x8 = 8 para escribir 2 posiciones de memoria en cada iteración	
	396	19 4 0 0		addi	x4, x0, 0	Posición inicial de memoria	
50	400	83 3 3 1	CICLO	sub	x3, x3,x1	Se hizo el ciclo 1 vez más	
	404	37 4 20 0		sw	x20, 0(x4)	Escribe valor (primera vez en el ciclo)	
51	408	19 20 20 5		addi	x20, x20, 5	aumenta 5 a x20 cada vez	
	412	37 4 20 4		sw	X20, 4(x4)	Escribe valor (segunda vez en el ciclo)	
52	416	19 20 20 5		addi	x20, x20, 5	aumenta 5 a x20 cada vez	
	420	71 4 4 8		add	x4, x4, x8	Se corre 8 bytes para escribir 2 siguientes posiciones de memoria	
53	424	100 3 0 -7		bne	x3, x0, CICLO	Si no se ha hecho 32 veces el ciclo continúa	
	428	999 0 0 0		FIN		FIN DE HILO 1	

X3=0

X4=0 + 32*8 = 256

X8=8

X20 = 325

Bl. Inst	Dir mem Inst	El hilo 2 almacena en memoria 68 valores desde -800, y restaándole 2 cada ve(-800 hasta -800 + (-2 * 67) = -800 + -134 = -934 en memoria a partir de posición 0 de memoria hasta posición 268 (bloque 0 hasta bloque 33)					H
		codificado	etiq.	instrucción		resultado	
54	432	19 20 0 -800		addi	x20, x0, -800	valor inicial a escribir a memoria (de -800 a -928)	H
	436	19 3 0 34		addi	x3, x0, 34	Va a escribir 68 valores seguidos en memoria a partir de posición 0 (ciclo se hace 34 veces, pues escribe de dos en dos)	
55	440	19 8 0 8		addi	x8, x0, 8	x8 = 8 para escribir 2 posiciones de memoria en cada iteración	I
	444	19 4 0 0		addi	x4, x0, 0	Posición inicial de memoria	
56	448	83 3 3 1	Etiqu	sub	x3, x3,x1	Se hizo el ciclo 1 vez más	L
	452	37 4 20 0		sw	x20, 0(x4)	Escribe valor (primera vez en el ciclo)	
57	456	19 20 20 -2		addi	x20, x20, -2	aumenta 5 a x20 cada vez	O
	460	37 4 20 4		sw	X20, 4(x4)	Escribe valor (segunda vez en el ciclo)	
58	464	19 20 20 -2		addi	x20, x20, -2	resta 2 a x20 cada vez	2
	468	71 4 4 8		add	x4, x4, x8	Se corre 8 bytes para escribir 2 siguientes posiciones de memoria	
59	472	100 3 0 -7		bne	x3, x0, Etiqu	Si no se ha hecho 32 veces el ciclo continúa	
	476	999 0 0 0		FIN		FIN DE HILO 2	

X3=0

X4=0 + 32*8 = 272

X8=8

X20 = -936

VALORES FINALES EN MEM DATOS COMPARTIDA

Si no coinciden los valores para algún bloque, buscarlo en la caché de datos L1, pudo quedar ahí como modificado- o en el buffer víctima

BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR
0	0	5	8	64	85	16	128	165	24	192	245	32	256	1	40	320	1
	4	10		68	90		132	170		196	250		260	1		324	1
1	8	15	9	72	95	17	136	175	25	200	255	33	264	1	41	328	1
	12	20		76	100		140	180		204	260		268	1		332	1
2	16	25	10	80	105	18	144	185	26	208	265	34	272	1	42	336	1
	20	30		84	110		148	190		212	270		276	1		340	1
3	24	35	11	88	115	19	152	195	27	216	275	35	280	1	43	344	1
	28	40		92	120		156	200		220	280		284	1		348	1
4	32	45	12	96	125	20	160	205	28	224	285	36	288	1	44	352	1
	36	50		100	130		164	210		228	290		292	1		356	1
5	40	55	13	104	135	21	168	215	29	232	295	37	296	1	45	360	1
	44	60		108	140		172	220		236	300		300	1		364	1
6	48	65	14	112	145	22	176	225	30	240	305	38	304	1	46	368	1
	52	70		116	150		180	230		244	310		308	1		372	1
7	56	75	15	120	155	23	184	235	31	248	315	39	312	1	47	376	1
	60	80		124	160		188	240		252	320		316	1		380	1

CACHE DATOS			
			0
			1

BUFFER VÍCTIMA		

REGISTROS
X3=0
X4= 256
X8=8
X20=325

VALORES FINALES EN MEM DATOS COMPARTIDA

Si no coinciden los valores para algún bloque, buscarlo en la caché de datos L1, pudo quedar ahí como modificado- o en el buffer víctima

BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR
0	0	-800	8	64	-832	16	128	-864	24	192	-896	32	256	-928	40	320	1
	4	-802		68	-834		132	-866		196	-898		260	-930		324	1
1	8	-804	9	72	-836	17	136	-868	25	200	-900	33	264	-932	41	328	1
	12	-806		76	-838		140	-870		204	-902		268	-934		332	1
2	16	-808	10	80	-840	18	144	-872	26	208	-904	34	272	1	42	336	1
	20	-810		84	-842		148	-874		212	-906		276	1		340	1
3	24	-812	11	88	-844	19	152	-876	27	216	-908	35	280	1	43	344	1
	28	-814		92	-846		156	-878		220	-910		284	1		348	1
4	32	-816	12	96	-848	20	160	-880	28	224	-912	36	288	1	44	352	1
	36	-818		100	-850		164	-882		228	-914		292	1		356	1
5	40	-820	13	104	-852	21	168	-884	29	232	-916	37	296	1	45	360	1
	44	-822		108	-854		172	-886		236	-918		300	1		364	1
6	48	-824	14	112	-856	22	176	-888	30	240	-920	38	304	1	46	368	1
	52	-826		116	-858		180	-890		244	-922		308	1		372	1
7	56	-828	15	120	-860	23	184	-892	31	248	-924	39	312	1	47	376	1
	60	-830		124	-862		188	-894		252	-926		316	1		380	1

CACHE DATOS			
			0
			1

BUFFER VÍCTIMA		

REGISTROS
X3=0
X4=0 + 32*8 = 272
X8=8
X20 = -936