

Modificado  
29 JUNIO  
2016

**HILOS PRUEBA SIN LL NI SC y con "LOOPS" CORTOS (pocos saltos con branches)**  
**PROYECTO CI1323-2016-1-SEM**  
**NO TOMAR EN CUENTA LO QUE SE EXPLICA EN CADA HILILLOS SOBRE EL USO DE CANDADOS**

**El hilillo 0** pone ceros en algunas posiciones de memoria. La idea es que servía como liberación de candados. Aún ahora los otros hilillos se quedarán encicladados mientras que el 0 no haya escrito estos 0.

Así, se necesita un 0 en posición 128 para hilillo1  
 un 0 en posición 132 para hilillo 2  
 un 0 en posición 136 para hilillo3  
 un 0 en posición 140 para hilillo 4  
 un 0 en posición 144 para hilillo5  
 un 0 en posición 148 para hilillo6

**El hilillo 1** escribe en memoria, y **el hilillo 3** no lo va a leer, si el hilillo 1 no ha escrito un 0 en la posición 176.

**El hilillo 4** necesita además un 0 en la dirección 260 para comenzar a escribir en memoria el valor 4 varias veces. Pone un 1 en la posición 260 antes ponerse a escribir. Al final escribe un 0 en posición 260, y un 0 en posición 264.

**El hilillo 5** necesita además un 0 en la dirección 260 para comenzar a escribir en memoria el valor 5 varias veces. Pone un 1 en la posición 260 antes ponerse a escribir. Al final Escribe un 0 en posición 260, y un 0 en posición 264.

**El hilillo 6** procede a leer de memoria lo que escribió el hilo 4, o el hilo 5, o ambos, hasta que haya un 0 en la posición 264. (hilillo 4 y el 5 escriben un 0 en esa posición)

**El hilo 0**

	etiq.	instrucción		resultado	H
43 0 0 260		SW	R0, 260(R0)	Libera candado para hilos 4 y 5. El primero que lo toma escribe	H I L O 0
43 0 0 128		SW	R0, 128(R0)	Libera candado hilo 1	
43 0 0 132		SW	R0, 132(R0)	Libera candado hilo 2	
43 0 0 136		SW	R0, 136(R0)	Libera candado hilo 3	
43 0 0 140		SW	R0, 140(R0)	Libera candado hilo 4	
43 0 0 144		SW	R0, 144(R0)	Libera candado hilo 5	
43 0 0 148		SW	R0, 148(R0)	Libera candado hilo 6	
63 0 0 0		FIN		FIN DE HILO 0	

No modifica registros

**El hilo 1 almacena en memoria 4 veces el valor 20 a partir de posición 200 (bloque 12 ) hasta la 212 (bloque 13). Candado para hilo 3 almacenado en posición 176 (bloque 11) .**

codificado	eti.	instrucción		resultado	H	
8 0 1 1	ETIQ1	DADDI	R1, R0, #1	R1 = 1		R1 = 1
35 0 2 128	Allá	LW	R2, 128(R0)	Candado para poder trabajar luego de hilo principal. Una vez que entre, solo escribe.		R2 = 0
43 0 1 128		SW	R1, 128(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	H	R3 = 0
4 1 0 -4		BEQZ	R1, ETIQ1	Si SC falló vaya a intentar hacer la operación atómica de nuevo	I	R4 = 216
5 2 0 -4		BNEZ	R2, Allá	Si R2 no es 0 se mantiene en espera activa	L	R8 = 8
8 0 20 20		DADDI	R20, R0, #20	R20 Contiene valor que se almacenará en memoria	O	R20 = 20
8 0 3 2		DADDI	R3, R0, #2	Va a escribir 4 veces el valor 1 a partir de la posición 840 de 2 en 2 por lo que el ciclo se hace solo 2 veces		
8 0 8 8		DADDI	R8, R0, #8	R8 = 8 para leer 2 posisicones de memoria cada vez		
8 0 4 200		DADDI	R4, R0, #200	Posición inicial de memoria	1	
34 3 1 3	CICLO	DSUB	R3, R3, R1	Se hizo el ciclo 1 vez más		
43 4 20 0		SW	R20, 0(R4)	Escribe valor 20 (primera vez en el ciclo)		
43 4 20 4		SW	R20, 4(R4)	Escribe valor 20 (segunda vez en el ciclo)		
32 4 8 4		DADD	R4, R4, R8	Se corre 8 bytes para escribir 2 siguientes posiciones de memoria		
5 3 0 -5		BNEZ	R3, CICLO	Si no se ha hecho 2 veces el ciclo continúa		
43 0 0 176		SW	R0, 176(R0)	Libera el candado para hilo 3		
63 0 0 0		FIN		FIN DE HILO 1		

**El hilo 2** usa JAL y JR , suma, resta, multiplicación y división y guarda en memoria pocos resultados. Así: 3 en 0 y 9 en pos 4 ( bloque 0)

codificado	etiq.	instrucción		resultado	H
8 0 1 1	<b>ETIQ1</b>	DADDI	R1, R0, #1	R1 = 1	R1=1
35 0 2 132	<b>Allá</b>	LW	R2, 132(R0)	<b>Candado para poder trabajar luego de hilo principal.</b> Una vez que entre, hace sus cálculos.	
43 0 1 132		SW	R1, 132(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	R2=2
4 1 0 -4		BEQZ	R1, <b>ETIQ1</b>	Si SC falló vaya a intentar hacer la operación atómica de nuevo-	R4=0
5 2 0 -4		BNEZ	R2, <b>Allá</b>	Si R2 no es 0 se mantiene en espera activa	<b>R5= 1</b>
<b>8 0 4 2</b>		DADDI	R4, R0, #2	<b>CICLO "ATRASO" SE HARÁ 2 VECES</b>	R21=0
34 5 5 5		DSUB	R5, R5, R5	R5 = 0	R22=42
8 4 4 -1	<b>ATRASE</b>	DADDI	R4, R4, # -1	<b>Se hace (R4 - 1) 2 VECES (HASTA QUE VALGA 0)</b>	<b>R23=3</b>
32 5 4 5		DADD	R5, R5, R4	<b>AL FINAL R5 = 1</b>	R31=?
5 4 0 -3		BNEZ	R4, <b>ATRASE</b>	SI R4 = 0 SALE	H El valor del R31, Depende de cómo se guardaron los hilillos en la memoria de instrucciones. Debe dar la posición en la que está la multiplicación, luego del JAL.
3 0 0 16		JAL	16	SE SALTA 4 INST. PARA LLEGAR AL DADDI R21 ... (R31 = pc, pc = pc + 16 = ?)	
12 22 2 22		DMUL	R22, R22, R2	R22 = 9 * 2 = 18	I L O 2
<b>43 0 23 0</b>		SW	R23, 0(R0)	M(0) = R23 = <b>3</b>	
43 0 22 4		SW	R22, 4(R0)	M(4) = R22 = <b>9</b>	
63 0 0 0		FIN		FIN DE HILO 2	
8 0 21 12		DADDI	R21, R0, #12	R21 =12	
34 22 22 22		DSUB	R22, R22, R22	R22 = 0	
8 0 2 2		DADDI	R2, R0, #2	R2 =2	
14 21 2 23	<b>SIGA</b>	DDIV	R23, R21, R2	R23 = R21/R2 = R21/2 R21 SIEMPRE ES PAR	
<b>8 21 21 -6</b>		DADDI	R21, R21, # -6	<b>Se hace el ciclo 1 vez más (SOLO SE HARÁ 2 VECES YA QUE SE DISMINUYE DE 2 EN 2)</b>	
32 22 23 22		DADD	R22, R22, R23	R22 = R22 + R23 (SE ACUMULA ACÁ (12/2 + 6/2 ) = 9 )	
5 21 0 -4		BNEZ	R21, <b>SIGA</b>		
2 31 0 0		JR	31	PC = R31 =? SALTA A LA MULTIPLICACIÓN	

**El hilo 3** HACE LA LECTURA DE LO QUE EL HILO 1 ESCRIBIO (CANDADO EN POSICION **176- bloque 11**). lo lee, lo acumula, lo multiplica por 3 y lo escribe en memoria en la **posición 196 (bloque 12)**.

codificado	etiq.	instrucción		resultado	H
8 0 1 1	<b>ETIQ1</b>	DADDI	R1, R0, #1	R1 = 1	<b>R1=1</b> <b>R2= 0</b> <b>R3 = 0</b> <b>R4=216</b>
35 0 2 136	<b>Allá</b>	LW	R2, 136(R0)	<b>Candado para poder trabajar luego de hilo principal.</b> Una vez que entre debe revisar el Candado para poder leer lo de hilo1	
43 0 1 136		SW	R1, 136(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	
4 1 0 -4		BEQZ	R1, <b>ETIQ1</b>	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>	
5 2 0 -4		BNEZ	R2, <b>Allá</b>	Si R2 no es 0 se mantiene en espera activa	
8 0 1 1	<b>¿YA?</b>	DADDI	R1, R0, #1	R1 = 1	<b>R8=8</b> <b>R10=2 ó 1</b> <b>R11=2 ó 1</b> <b>R12=20</b> <b>R13=3</b>
35 0 2 176	<b>AHÍ</b>	LW	R2, 176(R0)	<b>Candado para leer lo escrito por hilo 1</b>	
43 0 1 176		SW	R1, 176(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	
4 1 0 -4		BEQZ	R1, <b>¿YA?</b>	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>	
5 2 0 -4		BNEZ	R2, <b>AHÍ</b>	Si R2 no es 0 se mantiene en espera activa	
<b>8 0 3 2</b>		DADDI	<b>R3, R0, #2</b>	Va a leer <b>4</b> palabras a partir de la posición 200 de memoria, pero lo hace de 2 en 2 por lo que el ciclo se hace solo <b>2</b> veces	<b>I R14= entre 12 y 240</b>
8 0 13 3		DADDI	R13, R0, #3	R13 = 3	<b>O</b> <b>3</b>
34 12 12 12		DSUB	R12, R12, R12	R12 = 0 para acumular valores a leer	
8 0 8 8		DADDI	R8, R0, #8	R8 = 8 para leer 2 posiciones de memoria cada vez	
8 0 4 200		DADDI	R4, R0, #200	Posición inicial de memoria	
34 3 1 3	<b>CICLO</b>	DSUB	R3, R3, R1	Se hizo el ciclo 1 vez más	
35 4 10 0		LW	R10, 0(R4)	Lee valor 20 ó 1 (primera vez en el ciclo)	
35 4 11 4		LW	R11, 4(R4)	Lee valor 20 ó 1 (segunda vez en el ciclo)	
32 12 10 12		DADD	R12, R12, R10	Acumula primer lectura	
32 4 8 4		DADD	R4, R4, R8	Se corre 8 bytes para leer 2 siguientes posiciones de memoria	
32 12 11 12		DADD	R12, R12, R11	Acumula 2da lectura	
5 3 0 -7		BNEZ	R3, <b>CICLO</b>	Si no se ha hecho 2 veces el ciclo continúa	
12 12 13 14		DMUL	R14, R12, R13	Suma total se multiplica por 3 ( <b>entre 4 y 80 ] * 3 = entre 12 y 240</b> )	
43 0 14 196		SW	R14, 196(R0)	Almacena resultado en posición 196 de memoria	
63 0 0 0		FIN		FIN HILO 3	

**El hilo 4 escribe 4 veces el valor 4 desde la posición de memoria 304 hasta 316 (bloque 19) . Candado en posición 260 bloque 16. Así que escribe libera candado en pos 264 para que hilo 6 pueda leer. (si hilo 5 escribió primero, ya había un cero acá).**

codificado	etiq.	instrucción		resultado	H
8 0 1 1	ETIQ1	DADDI	R1, RO, #1	R1 = 1	H
35 0 2 140	Allá	LW	R2, 140(R0)	Candado para poder trabajar luego de hilo principal. Una vez que entre debe revisar el Candado en 1476 para saber si hilo 5 no ha comenzado a escribir o si ya finalizó solo	
43 0 1 140		SW	R1, 140(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	
4 1 0 -4		BEQZ	R1, ETIQ1	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>	
5 2 0 -4		BNEZ	R2, Allá	Si R2 no es 0 se mantiene en espera activa	
8 0 1 1	ACÁ	DADDI	R1, RO, #1	R1 = 1	H
35 0 2 260	AHÍ	LW	R2, 260(R0)	Candado en posición 260 de mem (bloque 92) para que hilo 5 no escriba mientras hilo 4 lo hace. Dado que no hay sincronización, esto no se controla y pueden escribir al mismo tiempo.	
43 0 1 260		SW	R1, 260(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1	
4 1 0 -4		BEQZ	R1, ACÁ	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>	
5 2 0 -4		BNEZ	R2, AHÍ	Si R2 no es 0 se mantiene en espera activa	
8 0 14 4		DADDI	R14, R0, #4	R14 = 4 (VALOR QUE ALMACENARÁ EN MEMORIA)	L
8 0 3 2		DADDI	R3, R0, #2	Va a escribir 4 veces el valor 4 a partir de la posición 304 de 2 en 2 por lo que el ciclo se hace solo 2 veces	
8 0 8 8		DADDI	R8, R0, #8	R8 = 8 para leer 2 posiciones de memoria cada vez	
8 0 4 304		DADDI	R4, R0, #304	Posición inicial de memoria	
34 3 1 3	CICLO	DSUB	R3, R3, R1	Se hizo el ciclo 1 vez más	
43 4 14 0		SW	R14, 0(R4)	Escribe valor 4 (primera vez en el ciclo)	O
43 4 14 4		SW	R14, 4(R4)	Escribe valor 4 (segunda vez en el ciclo)	
32 4 8 4		DADD	R4, R4, R8	Se corre 8 bytes para escribir 2 siguientes posiciones de memoria	
5 3 0 -5		BNEZ	R3, CICLO	Si no se ha hecho 2 veces el ciclo continúa	
43 0 0 260		SW	R0, 260(R0)	Libera el candado para hilos 5	
43 0 0 264		SW	R0, 264(R0)	Libera candado para hilo 6 para que pueda leer. POR SI HILO 4 ESCRIBIÓ PRIMERO QUE EL 5. SI VA DE SEGUNDO, YA ACÁ HAY UN 0.	4
63 0 0 0		FIN		FIN DE HILO 4	

R1=1

R2=0

R3=0

R4=320

R8=8

R14=4

**El hilo 5** escribe 4 veces el valor 5 desde la posición de memoria 304 hasta 316 (bloque 19). **Candado en 260**. Así que escribe libera candado en **pos 264** para que hilo 6 pueda leer (si hilo 4 escribió primero, ya había un cero acá).

codificado	etiq.	instrucción	resultado	H	
8 0 1 1	<b>ETIQ1</b>	DADDI R1, R0, #1	R1 = 1	H	R1=1
35 0 2 144	<b>Allá</b>	LW R2, 144(R0)	<b>Candado para poder trabajar luego de hilo principal.</b> Una vez que entre debe revisar el Candado en 260 para saber si hilo 5 no ha comenzado a escribir o si ya finalizó solo		R2=0
43 0 1 144		SW R1, 144(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1		R3=0
4 1 0 -4		BEQZ R1, <b>ETIQ1</b>	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>		<b>R4=320</b>
5 2 0 -4		BNEZ R2, <b>Allá</b>	Si R2 no es 0 se mantiene en espera activa		R8=8
8 0 1 1	<b>ACÁ</b>	DADDI R1, R0, #1	R1 = 1	H	R14=5
35 0 2 260	<b>AHÍ</b>	LW R2, 260(R0)	<b>Candado en posición 260 de mem para que hilo 4 no escriba mientras hilo 5 lo hace. Dado que no hay sincronización, esto no se controla y pueden escribir al mismo tiempo.</b>		
43 0 1 260		SW R1, 260(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1		
4 1 0 -4		BEQZ R1, <b>ACÁ</b>	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>		
5 2 0 -4		BNEZ R2, <b>AHÍ</b>	Si R2 no es 0 se mantiene en espera activa		
8 0 14 5		DADDI R14, R0, #5	R14 = 5 (VALOR QUE ALMACENARÁ EN MEMORIA)	I L O	
<b>8 0 3 2</b>		<b>DADDI R3, R0, #2</b>	<b>Va a escribir 4 veces el valor 5 a partir de la posición 304 de 2 en 2 por lo que el ciclo se hace solo 2 veces</b>		
8 0 8 8		DADDI R8, R0, #8	R8 = 8 para leer 2 posiciones de memoria cada vez		5
<b>8 0 4 304</b>		<b>DADDI R4, R0, #304</b>	Posición inicial de memoria		
34 3 1 3	<b>CICLO</b>	DSUB R3, R3, R1	Se hizo el ciclo 1 vez más		
43 4 14 0		SW R14, 0(R4)	Escribe valor 5 (primera vez en el ciclo)		
43 4 14 4		SW R14, 4(R4)	Escribe valor 5 (segunda vez en el ciclo)		
32 4 8 4		DADD R4, R4, R8	Se corre 8 bytes para escribir 2 siguientes posiciones de memoria		
5 3 0 -5		BNEZ R3, <b>CICLO</b>	<b>Si no se ha hecho 2 veces el ciclo continúa</b>		
43 0 0 260		SW R0, 260(R0)	Libera el candado para hilo 4		
<b>43 0 0 264</b>		SW <b>R0, 264(R0)</b>	Libera el candado para hilos 6 -POR SI HILO 5 ESCRIBIÓ PRIMERO QUE EL 4. SI VA DE SEGUNDO, YA ACÁ HAY UN 0.		
63 0 0 0		FIN	FIN DE HILO 5		

**El hilo 6** procede a leer desde posición **304 a la 316 (bloque 19)**, lo acumula, lo MULTIPLICA POR -1 Candado en pos **264** (puesto por hilos 4 ó 5) y lo almacena en posición **256 de memoria (bloque 16)**. Note que puede leer 4's, 5's y 1's

codificado	etiq.	instrucción		resultado	H	
8 0 1 1	ETIQ1	DADDI	R1, R0, #1	R1 = 1	H	R1=1
35 0 2 148	Allá	LW	R2, 148(R0)	Candado para poder trabajar luego de hilo principal. Una vez que entre debe revisar el Candado en 1480 para saber si hilo 4 ó 5 ya escribió		R2=0
43 0 1 148		SW	R1, 148(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1		R3=0
4 1 0 -4		BEQZ	R1, ETIQ1	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>		R4=320
5 2 0 -4		BNEZ	R2, Allá	Si R2 no es 0 se mantiene en espera activa		R8=8
8 0 1 1	ALLÁ	DADDI	R1, R0, #1	R1 = 1	H	R10= 1, 4 ó 5
35 0 2 264	AQUÍ	LW	R2, 264(R0)	Candado para poder leer lo que hilo 4 ó 5 escribió. En realidad no se necesita que sea un candado. Solo este hilo va a leer		R11 = 1, 4 ó 5
43 0 1 264		SW	R1, 264(R0)	Si puede hacer la op atómica escribe R1(que vale 1) en lock y devuelve un 1 en R1 . Si no, no escribe y devuelve un 0 en R1		R12= ENTRE 4 y 20
4 1 0 -4		BEQZ	R1, ALLÁ	<del>Si SC falló vaya a intentar hacer la operación atómica de nuevo</del>		R14= ENTRE -4 y -20
5 2 0 -4		BNEZ	R2, AQUÍ	Si R2 no es 0 se mantiene en espera activa		R15=-1
8 0 3 2		DADDI	R3, R0, #2	Va a leer 4 valores partir de la posición 304 de 2 en 2 por lo que el ciclo se hace solo 2 veces	O	
34 12 12 12		DSUB	R12, R12, R12	R12 = 0 para acumular valores a leer		
8 0 8 8		DADDI	R8, R0, #8	R8 = 8 para leer 2 posiciones de memoria cada vez		
8 0 4 304		DADDI	R4, R0, #304	Posición inicial de memoria		
8 0 15 -1		DADDI	R15, R0, #-1	R15= -1		
34 3 1 3	CICLO	DSUB	R3, R3, R1	Se hizo el ciclo 1 vez más	6	
35 4 10 0		LW	R10, 0(R4)	Lee valor 1, 4 ó 5 (primera vez en el ciclo)		
35 4 11 4		LW	R11, 4(R4)	Lee valor 1, 4 ó 5 (segunda vez en el ciclo)		
32 12 10 12		DADD	R12, R12, R10	Acumula primer lectura		
32 4 8 4		DADD	R4, R4, R8	Se corre 8 bytes para leer 2 siguientes posiciones de memoria		
32 12 11 12		DADD	R12, R12, R11	Acumula 2da lectura		
5 3 0 -7		BNEZ	R3, CICLO	Si no se ha hecho 2 veces el ciclo continúa		
12 12 15 14		DMUL	R14, R12, R15	Suma total se multiplica por -1 (valdrá entre -4 y -20 )		
43 0 14 256		SW	R14, 256(R0)	Almacena resultado en posición 256 de memoria		
63 0 0 0		FIN		FIN DE HILO 6		

Memoria Compartida  
Valores Finales  
(algunos pudieron haber quedado como bloques M en alguna caché)

BL	Pos	VALOR	BL	Pos	VALOR	BL	Pos	VALOR
0	0	3	8	128	0 luego 1	16	256	entre -4 y -20
	4	9		132	0 luego 1		260	0
	8	1		136	0 luego 1		264	0
	12	1		140	0 luego 1		268	1
1	16	1	9	144	0 luego 1	17	272	1
	20	1		148	0 luego 1		276	1
	24	1		152	1		280	1
	28	1		156	1		284	1
2	32	1	10	160	1	18	288	1
	36	1		164	1		292	1
	40	1		168	1		296	1
	44	1		172	1		300	1
3	48	1	11	176	1	19	304	4 ó 5
	52	1		180	1		308	4 ó 5
	56	1		184	1		312	4 ó 5
	60	1		188	1		316	4 ó 5
4	64	1	12	192	1	20	320	1
	68	1		196	entre 12 y 240		324	1
	72	1		200	20		328	1
	76	1		204	20		332	1
5	80	1	13	208	20	21	336	1
	84	1		212	20		340	1
	88	1		216	1		344	1
	92	1		220	1		348	1
6	96	1	14	224	1	22	352	1
	100	1		228	1		356	1
	104	1		232	1		360	1
	108	1		236	1		364	1
7	112	1	15	240	1	23	368	1
	116	1		244	1		372	1
	120	1		248	1		376	1
	124	1		252	1		380	1

Puede que queden en caché como bloque M