

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

El hilo 0 pone un cero en posición 260 de mem para que hilo 4 ó hilo 5 comience a escribir. Libera candados para que hilos 1,2,3,4,5 y 6 puedan comenzar.

El hilo 1 almacena en memoria 10 veces el valor 2 a partir de posición 200 (bloque 12) hasta la 236 (bloque 14). Libera candado para hilo 3 almacenado en posición 176 (bloque 11) .

El hilo 2 usa jal y jalr , suma, resta, multiplicación y división y guarda en memoria pocos resultados. Así: 4950 en 0 y 42 en pos 4 (bloque 0)

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

El hilo 4 escribe 22 veces el valor 4 desde la posición de memoria 296 (bloque 18) hasta la 380 (bloque 23). 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).

El hilo 5 escribe 22 veces el valor 5 desde la posición de memoria 296 (bloque 18) hasta la 380 (bloque 23) . 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).

El hilo 6 procede a leer desde posición **296 (bloque 18) a la 380 (bloque 23)**, lo acumula, lo MULTIPLICA POR -1 Candado en pos **264** (será liberado por hilo 4 ó 5) y lo almacena en posición 256 de memoria. Note que puede leer lo que escribió el hilo 4 ó lo que escribió el hilo 5, ó una combinación de ellos, pero no 1's.

| Bl. Inst | Dir mem Inst | El hilo 0 pone un cero en posición 260 de mem para que hilo 4 ó hilo 5 comience a escribir. Libera candados para que hilos 1,2,3,4,5 y 6 puedan comenzar. | | | | | H |
|----------|--------------|---|-------|-------------|-------------|---|---|
| | | codificado | etiq. | instrucción | | resultado | |
| 24 | 384 | 37 0 0 260 | | sw | x0, 260(x0) | Libera candado para hilos 4 y 5. El primero que lo toma escribe | |
| | 388 | 37 0 0 128 | | sw | x0, 128(x0) | Libera candado hilo 1 | H |
| | 392 | 37 0 0 132 | | sw | x0, 132(x0) | Libera candado hilo 2 | I |
| | 396 | 37 0 0 136 | | sw | x0, 136(x0) | Libera candado hilo 3 | L |
| 25 | 400 | 37 0 0 140 | | sw | x0, 140(x0) | Libera candado hilo 4 | O |
| | 404 | 37 0 0 144 | | sw | x0, 144(x0) | Libera candado hilo 5 | |
| | 408 | 37 0 0 148 | | sw | x0, 148(x0) | Libera candado hilo 6 | 0 |
| | 412 | 999 0 0 0 | | FIN | | FIN DE HILO 0 | |

No modifica registros

NOTA IMPORTANTE: Se incluye la **dirección en la memoria de instrucciones** para cada instrucción de cada hilillo, pero esta no coincidirá con la suya ya que la de acá se pone como quedaría en memoria de instrucciones compartida centralizada, mientras que en el proyecto los hilillos se distribuirán entre los 3 procesadores y se almacenarán en las memorias de instrucciones propias, las cuales inician en el byte 128.

| Bl. Inst | Dir mem Inst | El hilo 1 almacena en memoria 10 veces el valor 2 a partir de posición 200 (bloque 12) hasta la 236 (bloque 14). Candado para hilo 3 almacenado en posición 176 (bloque 11) . | | | | | H |
|-------------|-----------------|--|-------|-------------|---------------|---|---|
| | | codificado | etiq. | instrucción | | resultado | |
| 26 | 416 | 19 9 0 128 | | addi | x9, x0, 128 | x9=128 | H X1=1 I X2=0 L X3=0 O X4=240 X8=8 X9=128 X20 = 2 |
| | 420 | 51 2 9 0 | Acá | lr | x2, x9 | Candado para poder trabajar luego de hilo principal. Una vez que entre, hace sus cálculos. | |
| | 424 | 100 2 0 -2 | | bne | x2, x0, Acá | Si el candado está ocupado lo vuelve a leer | |
| | 428 | 19 1 0 1 | | addi | x1, x0, 1 | x1 = 1 | |
| 27 | 432 | 52 9 1 0 | | sc | x1, 0(x9) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | 1 |
| | 436 | 99 1 0 -5 | | beq | x1, x0, Acá | Si sc falló vaya a intentar hacer la operación atómica de nuevo | |
| | 440 | 19 20 0 2 | | addi | x20, x0, 2 | | |
| | 444 | 19 3 0 5 | | addi | x3, x0, 5 | Va a escribir 10 veces el valor 1 a partir de la posición 200 de 2 en 2 por lo que el ciclo se hace solo 5 veces | |
| 28 | 448 | 19 8 0 8 | | addi | x8, x0, 8 | x8 = 8 para escribir 2 posiciones de memoria en cada iteración | 1 |
| | 452 | 19 4 0 200 | | addi | x4, x0, 200 | Posición inicial de memoria | |
| | 456 | 83 3 3 1 | CICLO | sub | x3, x3,x1 | Se hizo el ciclo 1 vez más | |
| | 460 | 37 4 20 0 | | sw | x20, 0(x4) | Escribe valor 2 (primera vez en el ciclo) | |
| 29 | 464 | 37 4 20 4 | | sw | X20, 4(x4) | Escribe valor 2 (segunda vez en el ciclo) | 1 |
| | 468 | 71 4 4 8 | | add | x4, x4, x8 | Se corre 8 bytes para escribir 2 siguientes posiciones de memoria | |
| | 472 | 100 3 0 -5 | | bne | x3, x0, CICLO | Si no se ha hecho 5 veces el ciclo continúa | |
| | 476 | 37 0 0 176 | | sw | x0, 176(x0) | Libera el candado para hilo 3 | |
| 30 | 480 | 999 0 0 0 | | FIN | | FIN DE HILO 1 | |

| Bl. Inst | Dir mem Inst | El hilo 2 usa jal y jalr, suma, resta, multiplicación y división y guarda en memoria 2 resultados: 4950 en 0 y 42 en pos 4 (bloque 0) | | | | | H |
|-------------|-----------------|--|--------|-------------|----------------|---|--|
| | | codificado | etiq. | instrucción | | resultado | |
| 30 | 484 | 19 8 0 132 | | addi | x8, x0, 132 | x8=132 | x1=1 x2= 2 x4 =0 x5= 4950 X8= 132 x21=0 al final |
| | 488 | 51 2 8 0 | Allá | lr | x2, x8 | Candado para poder trabajar luego de hilo principal. Una vez que entre, hace sus cálculos. | |
| | 492 | 100 2 0 -2 | | bne | x2, x0, Allá | Si el candado está ocupado lo vuelve a leer | |
| 31 | 496 | 19 1 0 1 | | addi | x1, x0, 1 | x1 = 1 | x22 = 42 X23= 1 x30= dir de sig instr ¿580? x31= dir de la mult ¿532? |
| | 500 | 52 8 1 0 | | sc | x1, 0(x8) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | |
| | 504 | 99 1 0 -5 | | beq | x1, x0, Allá | Si sc falló vaya a intentar hacer la operación atómica de nuevo | |
| | 508 | 19 4 0 100 | | addi | x4, x0, 100 | CICLO "ATRASO" SE HARÁ 100 VECES | |
| 32 | 512 | 83 5 5 5 | | sub | x5, x5, x5 | x5 = 0 | H I L O 2 |
| | 516 | 19 4 4 -1 | ATRASE | addi | x4, x4, -1 | Se hace (x4 - 1) 100 VECES (HASTA QUE VALGA 0) | |
| | 520 | 71 5 5 4 | | add | x5, x5, x4 | AL FINAL X5 = 4950 | |
| | 524 | 100 4 0 -3 | | bne | x4, x0, ATRASE | SI x4 = 0 SALE | |
| 33 | 528 | 111 31 0 16 | | jal | x31, 16 | SE SALTA 4 INST. PARA LLEGAR AL addi x21 ... (x31 = pc, pc = pc + 16 = ?) | |
| | 532 | 72 22 22 2 | | mul | x22,x22, x2 | X22 = 21 *2 = 42 | |
| | 536 | 37 0 5 0 | | sw | x5, 0(x0) | M(0) = x5 = 4950 | |
| | 540 | 37 0 22 4 | | sw | x22, 4(x0) | M(4) = x22 = 42 | |
| 34 | 544 | 999 0 0 0 | | FIN | | FIN DE HILO 2 | |
| | 548 | 19 21 0 12 | | addi | x21, x0, 12 | x21 =12 | |
| | 552 | 83 22 22 22 | | sub | x22, x22, x22 | x22 = 0 | |
| | 556 | 19 2 0 2 | | addi | x2, x0, 2 | x2 =2 | |
| 35 | 560 | 56 23 21 2 | SIGA | div | x23, x21, x2 | x23 = x21/x2 = x21/2 x21 SIEMPRE ES PAR | |
| | 564 | 19 21 21 -2 | | addi | x21,x21, -2 | Se hace el ciclo 1 vez más (SOLO SE HARÁ 6 VECES YA QUE SE DISMINUYE DE 2 EN 2) | |
| | 568 | 71 22 22 23 | | add | x22, x22, x23 | x22 = x22 + x23 (SE ACUMULA ACÁ (12/2 + 10/2 + 8/2 + ...+2/2) = 21) | |
| | 572 | 100 21 0 -4 | | bne | x21, x0, SIGA | Salta 5 veces (en total se hace 6 veces) | |
| 36 | 576 | 103 30 31 0 | | jalr | x30, x31, 0 | X30 = PC, PC = x31 + 0 =? SALTA A LA MULTIPLICACIÓN | |

| Bl. Inst | Dir mem Inst | 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). | | | | | H |
|-------------|-----------------|---|-------|-------------|---------------|---|-----------|
| | | codificado | etiq. | instrucción | | resultado | |
| 36 | 580 | 19 30 0 136 | | addi | x30, x0, 136 | x30=136 | X1=1 |
| | 584 | 51 2 30 0 | Allá | lr | x2, x30 | Candado para poder trabajar luego de hilo principal. Una vez que entre, hace sus cálculos. | |
| | 588 | 100 2 0 -2 | | bne | x2, x0, Allá | Si el candado está ocupado lo vuelve a leer | |
| 37 | 592 | 19 1 0 1 | | addi | x1, x0, 1 | x1 = 1 | X2=0 |
| | 596 | 52 30 1 0 | | sc | x1, 0(x30) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1. Si no, no escribe y devuelve un 0 en x1 | X3=0 |
| | 600 | 99 1 0 -5 | | beq | x1, x0, Allá | Si sc falló vaya a intentar hacer la operación atómica de nuevo | X4 = 240 |
| | 604 | 19 17 0 176 | | addi | x17, x0, 176 | x17 = 176 | X8=8 |
| 38 | 608 | 19 1 0 1 | mal | addi | x1, x0, 1 | x1 = 1 | X10=2 |
| | 612 | 51 2 17 0 | AHÍ | lr | x2, x17 | Candado para leer lo escrito por hilo 1 | X11=2 |
| | 616 | 52 17 1 0 | | sc | x1, 0(x17) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1. Si no, no escribe y devuelve un 0 en x1 | X12= 20 |
| | 620 | 99 1 0 -4 | | beq | x1,x0, mal | Si sc falló vaya a intentar hacer la operación atómica de nuevo | H X13=3 |
| 39 | 624 | 100 2 0 -4 | | bne | x2,x0, AHÍ | Si x2 no es 0 se mantiene en espera activa | I X14=60 |
| | 628 | 19 3 0 5 | | addi | x3,x0,5 | Va a leer 10 palabras a partir de la posición 200 de memoria, pero lo hace de 2 en 2 por lo que el ciclo se hace solo 5 veces | L X17=176 |
| | 632 | 19 13 0 3 | | addi | x13, x0, 3 | x13 = 3 | X30=136 |
| | 636 | 83 12 12 12 | | sub | x12, x12,x12 | x12 = 0 para acumular valores a leer | O |
| 40 | 640 | 19 8 0 8 | | addi | x8,x0,8 | x8 = 8 para leer 2 posiciones de memoria cada vez | 3 |
| | 644 | 19 4 0 200 | | addi | x4, x0,200 | Posición inicial de memoria | |
| | 648 | 83 3 3 1 | CICLO | sub | x3, x3, x1 | Se hizo el ciclo 1 vez más | |
| | 652 | 5 10 4 0 | | lw | x10,0(x4) | Lee valor 2 (primera vez en el ciclo) | |
| 41 | 656 | 5 11 4 4 | | lw | x11, 4(x4) | Lee valor 2 (segunda vez en el ciclo) | |
| | 660 | 71 12 12 10 | | add | x12, x12, x10 | Acumula primer lectura | |
| | 664 | 71 4 4 8 | | add | x4, x4, x8 | Se corre 8 bytes para leer 2 siguientes posiciones de memoria | |
| | 668 | 71 12 12 11 | | add | x12, x12, x11 | Acumula 2da lectura | |
| 42 | 672 | 100 3 0 -7 | | bne | x3, x0, CICLO | Si no se ha hecho 5 veces el ciclo continúa | |
| | 676 | 72 14 12 13 | | mul | x14, x12, x13 | Suma total se multiplica por 3 (20 * 3 =60) | |
| | 680 | 37 0 14 196 | | sw | x14, 196(x0) | Almacena resultado en posición 196 de memoria | |
| | 684 | 999 0 0 0 | | FIN | | FIN HILO 3 | |

| Bl. Inst | Dir mem Inst | El hilo 4 escribe 22 veces el valor 4 desde la posición de memoria 296 (bloque 18) hasta la 380 (bloque 23). 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). | | | | | H |
|-------------|-----------------|---|-------|-------------|---------------|---|-----------|
| | | codificado | etiq. | instrucción | | resultado | |
| 43 | 688 | 19 30 0 140 | | addi | x30, x0, 140 | x30=140 | X1=1 |
| | 692 | 51 2 30 0 | Allá | lr | x2, x30 | Candado para poder trabajar luego de hilo principal. Una vez que entre, hace sus cálculos. | X2=0 |
| | 696 | 100 2 0 -2 | | bne | x2, x0, Allá | Si el candado está ocupado lo vuelve a leer | X3 = 0 |
| | 700 | 19 1 0 1 | | addi | x1, x0, 1 | x1 = 1 | X4 = 384 |
| 44 | 704 | 52 30 1 0 | | sc | x1, 0(x30) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1. Si no, no escribe y devuelve un 0 en x1 | X5 = 260 |
| | 708 | 99 1 0 -5 | | beq | x1, x0, Allá | Si sc falló vaya a intentar hacer la operación atómica de nuevo | X8 = 8 |
| | 712 | 19 5 0 260 | | addi | x5, x0, 260 | dir del candado . para poder escribir en memoria | X14= 4 |
| | 716 | 19 1 0 1 | ACÁ | addi | x1,x0,1 | x1 = 1 | X30 = 140 |
| 45 | 720 | 51 2 5 0 | AHÍ | lr | x2, x5 | Candado en posición 260 de mem (bloque 92) para que hilo 5 no escriba mientras hilo 4 lo hace | H |
| | 724 | 52 5 1 0 | | sc | x1, 0(x5) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | I |
| | 728 | 99 1 0 -4 | | beq | x1, x0, ACÁ | Si SC falló vaya a intentar hacer la operación atómica de nuevo | L |
| | 732 | 100 2 0 -4 | | bne | x2, x0, Ahí | Si x2 no es 0 se mantiene en espera activa | O |
| 46 | 736 | 19 14 0 4 | | addi | x14, x0, 4 | x14 = 4 (VALOR QUE ALMACENARÁ EN MEMORIA) | 4 |
| | 740 | 19 3 0 11 | | addi | x3, x0, 11 | Va a escribir 22 veces el valor 4 a partir de la posición 296 de 2 en 2 por lo que el ciclo se hace solo 11 veces | |
| | 744 | 19 8 0 8 | | addi | x8, x0, 8 | x8 = 8 para leer 2 posiciones de memoria cada vez | |
| | 748 | 19 4 0 296 | | addi | x4, x0, 296 | Posición inicial de memoria | |
| 47 | 752 | 83 3 3 1 | CICLO | sub | x3, x3, x1 | Se hizo el ciclo 1 vez más | |
| | 756 | 37 4 14 0 | | sw | x14, 0(x4) | Escribe valor 4 (primera vez en el ciclo) | |
| | 760 | 37 4 14 4 | | sw | x14, 4(x4) | Escribe valor 4 (segunda vez en el ciclo) | |
| | 764 | 71 4 4 8 | | add | x4, x4, x8 | Se corre 8 bytes para escribir 2 siguientes posiciones de memoria | |
| 48 | 768 | 100 3 0 -5 | | bne | x3, x0, CICLO | Si no se ha hecho 11 veces el ciclo continúa | |
| | 772 | 37 0 0 260 | | sw | x0, 260(x0) | Libera el candado para hilos 5 | |
| | 776 | 37 0 0 264 | | sw | x0, 264(x0) | 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. | |
| | 780 | 999 0 0 0 | | FIN | | FIN DE HILO 4 | |

| Bl. Inst | Dir mem Inst | El hilo 5 escribe 22 veces el valor 5 desde la posición de memoria 296 (bloque 18) hasta la 380 (bloque 23) . 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). | | | | | H |
|----------|--------------|--|-------|-------------|---------------|--|--|
| | | codificado | etiq. | instrucción | | resultado | |
| 49 | 784 | 19 30 0 144 | | addi | x30, x0, 144 | | X1= 1 X2 = 0 X3 = 0 X4 = 384 X8 = 8 X9= 260 X14 = 5 X30 = 144 |
| | 788 | 19 1 0 1 | ETIQ1 | addi | x1, x0, 1 | x1 = 1 | |
| | 792 | 51 2 30 0 | Allá | lr | x2, x30 | Candado en 144 para poder trabajar luego de hilo principal. 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 | |
| | 796 | 52 30 1 0 | | sr | x1, 0(x30) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | |
| 50 | 800 | 99 1 0 -4 | | beq | x1, x0, ETIQ1 | Si SC falló vaya a intentar hacer la operación atómica de nuevo | H I L O 5 |
| | 804 | 100 2 0 -4 | | bne | x2, x0, Allá | Si x2 no es 0 se mantiene en espera activa | |
| | 808 | 19 9 0 260 | | addi | x9, x0, 260 | X9= 260 | |
| | 812 | 19 1 0 1 | ACÁ | addi | x1, x0, 1 | x1=1 | |
| 51 | 816 | 51 2 9 0 | AHÍ | lr | x2, x9 | Candado en posición 260 de mem para que hilo 4 no escriba mientras hilo 5 lo hace | H I L O 5 |
| | 820 | 52 9 1 0 | | sc | x1, 0(x9) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | |
| | 824 | 99 1 0 -4 | | beq | x1,x0, ACÁ | Si SC falló vaya a intentar hacer la operación atómica de nuevo | |
| | 828 | 100 2 0 -4 | | bne | x2, x0, AHÍ | Si x2 no es 0 se mantiene en espera activa | |
| 52 | 832 | 19 14 0 5 | | addi | x14, x0, 5 | x14 = 5 (VALOR QUE ALMACENARÁ EN MEMORIA) | 5 |
| | 836 | 19 3 0 11 | | addi | x3, x0, 11 | Va a escribir 22 veces el valor 5 a partir de la posición 296 de 2 en 2 por lo que el ciclo se hace solo 11 veces | |
| | 840 | 19 8 0 8 | | addi | x8, x0, 8 | x8 = 8 para leer 2 posiciones de memoria cada vez | |
| | 844 | 19 4 0 296 | | addi | x4, x0, 296 | Posición inicial de memoria | |
| 53 | 848 | 83 3 3 1 | CICLO | sub | x3, x3, x1 | Se hizo el ciclo 1 vez más | 5 |
| | 852 | 37 4 14 0 | | sw | x14, 0(x4) | Escribe valor 5 (primera vez en el ciclo) | |
| | 856 | 37 4 14 4 | | sw | x14, 4(x4) | Escribe valor 5 (segunda vez en el ciclo) | |
| | 860 | 71 4 4 8 | | add | x4, x4, x8 | Se corre 8 bytes para escribir 2 siguientes posiciones de memoria | |
| 54 | 864 | 100 3 0 -5 | | bne | x3, x0, CICLO | Si no se ha hecho 11 veces el ciclo continúa | 5 |
| | 868 | 37 0 0 260 | | sw | x0, 260(x0) | Libera el candado para hilo 4 | |
| | 872 | 37 0 0 264 | | sw | x0, 264(x0) | Libera el candado para hilos 6 -POR SI HILO 5 ESCRIBIÓ PRIMERO QUE EL 4. SI VA DE SEGUNDO, YA ACÁ HAY UN 0. | |
| | 876 | 999 0 0 0 | | FIN | | FIN DE HILO 5 | |

| Bl. Inst | Dir mem Inst | El hilo 6 procede a leer desde posición 296 (bloque 18) a la 380 (bloque 23), 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 lo que escribió el hilo 4 ó lo que escribió el hilo 5, pero no 1's. | | | | | H |
|-------------|-----------------|--|-------|-------------|---------------|---|---|
| | | codificado | etiq. | instrucción | | resultado | |
| 55 | 880 | 19 9 0 148 | | addi | x9, x0, 148 | X9= 148 | X1 = 1 X2 = 0 X3 = 0 |
| | 884 | 19 1 0 1 | ETIQ1 | addi | x1, x0, 1 | x1 = 1 | |
| | 888 | 51 2 9 0 | Allá | lr | x2, x9 | Candado en 148 para poder trabajar luego de hilo principal. Una vez que entre debe revisar el Candado en 264 para saber si hilo 4 ó 5 ya escribió | |
| | 892 | 52 9 1 0 | | sc | x1, 0(x9) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | |
| 56 | 896 | 99 1 0 -4 | | beq | x1, x0, ETIQ1 | Si sc falló vaya a intentar hacer la operación atómica de nuevo | X4= 384 X8 = 8 X9= 148 X10= 4 ó 5 X11 = 4 ó 5 |
| | 900 | 100 2 0 -4 | | bne | x2, x0, Allá | Si x2 no es 0 se mantiene en espera activa | |
| | 904 | 19 16 0 264 | | addi | x16, x0, 264 | x16= 264 | |
| | 908 | 19 1 0 1 | ALLÁ | addi | x1, x0, 1 | x1 = 1 | |
| 57 | 912 | 51 2 16 0 | AQUÍ | lr | x2, x16 | Candado para poder leer lo que hilo 4 ó 5 escribió. No se necesita que sea un candado. Solo este hilo va a leer | I x12= ENTRE 88 y 110 x14= ENTRE -110 y -88 |
| | 916 | 52 16 1 0 | | sc | x1, 0(x16) | Si puede hacer la op atómica escribe x1(que vale 1) en lock y devuelve un 1 en x1 . Si no, no escribe y devuelve un 0 en x1 | |
| | 920 | 99 1 0 -4 | | beq | x1, x0, ALLÁ | Si sc falló vaya a intentar hacer la operación atómica de nuevo | |
| | 924 | 100 2 0 -4 | | bne | x2, x0, AQUÍ | Si x2 no es 0 se mantiene en espera activa | |
| 58 | 928 | 19 3 0 11 | | addi | x3, x0, 11 | Va a leer 22 palabras a partir de la posición 296 de memoria, pero lo hace de 2 en 2 por lo que el ciclo se hace solo 11 veces | L O 6 X15=-1 X16 = 264 |
| | 932 | 83 12 12 12 | | sub | x12, x12, x12 | x12 = 0 para acumular valores a leer | |
| | 936 | 19 8 0 8 | | addi | x8, x0, 8 | x8 = 8 para leer 2 posisicones de memoria cada vez | |
| | 940 | 19 4 0 296 | | addi | x4, x0, 296 | Posición inicial de memoria | |
| 59 | 944 | 19 15 0 -1 | | addi | x15, x0, -1 | x15= -1 | |
| | 948 | 83 3 3 1 | CICLO | sub | x3, x3, x1 | Se hizo el ciclo 1 vez más | |
| | 952 | 5 10 4 0 | | lw | x10, 0(x4) | Lee valor 4 ó 5 (primera vez en el ciclo) | |
| | 956 | 5 11 4 4 | | lw | x11, 4(x4) | Lee valor 4 ó 5 (segunda vez en el ciclo) | |
| 60 | 960 | 71 12 12 10 | | add | x12, x12, x10 | Acumula primer lectura | |
| | 964 | 71 4 4 8 | | add | x4, x4, x8 | Se corre 8 bytes para leer 2 siguientes posiciones de memoria | |
| | 968 | 71 12 12 11 | | add | x12, x12, x11 | Acumula 2da lectura | |
| | 972 | 100 3 0 -7 | | bne | x3, x0, CICLO | Si no se ha hecho 11 veces el ciclo continúa | |
| 61 | 976 | 72 14 12 15 | | mul | x14, x12, x15 | Suma total se multiplica por -1 (valdrá entre -110 y -88) | |
| | 980 | 37 0 14 256 | | sw | x14, 256(x0) | Almacena resultado en posición 256 de memoria | |
| | 984 | 999 0 0 0 | | FIN | | FIN DE HILO 6 | |

VALORES FINALES EN MEM DATOS COMPARTIDA

Si no coinciden los valores para algún bloque, buscarlo en las cachés de datos L1, pudo quedar ahí como modificado

| BL | Pos | VALOR | BL | Pos | VALOR | BL | Pos | VALOR | BL | Pos | VALOR | BL | Pos | VALOR | BL | Pos | VALOR |
|----|-----|-------|----|-----|-------|----|-----|-----------|----|-----|-------|----|-----|------------------|----|-----|-------|
| 0 | 0 | 4950 | 4 | 64 | 1 | 8 | 128 | 0 luego 1 | 12 | 192 | 1 | 16 | 256 | Entre -110 y -88 | 20 | 320 | 4 ó 5 |
| | 4 | 42 | | 68 | 1 | | 132 | 0 luego 1 | | 196 | 60 | | 260 | 0 | | 324 | 4 ó 5 |
| | 8 | 1 | | 72 | 1 | | 136 | 0 luego 1 | | 200 | 2 | | 264 | 0 | | 328 | 4 ó 5 |
| | 12 | 1 | | 76 | 1 | | 140 | 0 luego 1 | | 204 | 2 | | 268 | 1 | | 332 | 4 ó 5 |
| 1 | 16 | 1 | 5 | 80 | 1 | 9 | 144 | 0 luego 1 | 13 | 208 | 2 | 17 | 272 | 1 | 21 | 336 | 4 ó 5 |
| | 20 | 1 | | 84 | 1 | | 148 | 0 luego 1 | | 212 | 2 | | 276 | 1 | | 340 | 4 ó 5 |
| | 24 | 1 | | 88 | 1 | | 152 | 1 | | 216 | 2 | | 280 | 1 | | 344 | 4 ó 5 |
| | 28 | 1 | | 92 | 1 | | 156 | 1 | | 220 | 2 | | 284 | 1 | | 348 | 4 ó 5 |
| 2 | 32 | 1 | 6 | 96 | 1 | 10 | 160 | 1 | 14 | 224 | 2 | 18 | 288 | 1 | 22 | 352 | 4 ó 5 |
| | 36 | 1 | | 100 | 1 | | 164 | 1 | | 228 | 2 | | 292 | 1 | | 356 | 4 ó 5 |
| | 40 | 1 | | 104 | 1 | | 168 | 1 | | 232 | 2 | | 296 | 4 ó 5 | | 360 | 4 ó 5 |
| | 44 | 1 | | 108 | 1 | | 172 | 1 | | 236 | 2 | | 300 | 4 ó 5 | | 364 | 4 ó 5 |
| 3 | 48 | 1 | 7 | 112 | 1 | 11 | 176 | 1 | 15 | 240 | 1 | 19 | 304 | 4 ó 5 | 23 | 368 | 4 ó 5 |
| | 52 | 1 | | 116 | 1 | | 180 | 1 | | 244 | 1 | | 308 | 4 ó 5 | | 372 | 4 ó 5 |
| | 56 | 1 | | 120 | 1 | | 184 | 1 | | 248 | 1 | | 312 | 4 ó 5 | | 376 | 4 ó 5 |
| | 60 | 1 | | 124 | 1 | | 188 | 1 | | 252 | 1 | | 316 | 4 ó 5 | | 380 | 4 ó 5 |

Resultados en registros para cada hilo (HILILLO 0 NO MODIFICA REGISTROS)

Hilillo 1

| | |
|-----|-----|
| x1 | 1 |
| x2 | 0 |
| x3 | 0 |
| x4 | 240 |
| x8 | 8 |
| x9 | 128 |
| x20 | 2 |

2

| | |
|-----|-------|
| x1 | 1 |
| x2 | 2 |
| x4 | 0 |
| x5 | 4950 |
| x8 | 132 |
| x21 | 0 |
| x22 | 42 |
| x23 | 1 |
| x30 | 580¿? |
| x31 | 532¿? |

El valor del x31 y el de x30 depende de cómo se guardaron los hilillos en las memorias de instrucciones. X31 tendrá dir de la multiplicación luego del jal. Y x30 la dirección que sigue luego de la última instrucción

3

| | |
|-----|-----|
| x1 | 1 |
| x2 | 0 |
| x3 | 0 |
| x4 | 240 |
| x8 | 8 |
| x10 | 2 |
| x11 | 2 |
| x12 | 20 |
| x13 | 3 |
| x14 | 60 |
| x17 | 176 |
| x30 | 136 |

4

| | |
|-----|-----|
| x1 | 1 |
| x2 | 0 |
| x3 | 0 |
| x4 | 384 |
| x5 | 260 |
| x8 | 8 |
| x14 | 4 |
| x30 | 140 |

5

| | |
|-----|-----|
| x1 | 1 |
| x2 | 0 |
| x3 | 0 |
| x4 | 384 |
| x8 | 8 |
| x9 | 260 |
| x14 | 5 |
| x30 | 144 |

6

| | |
|-----|------------------|
| x1 | 1 |
| x2 | 0 |
| x3 | 0 |
| x4 | 384 |
| x8 | 8 |
| x9 | 148 |
| x10 | 4 ó 5 |
| x11 | 4 ó 5 |
| x12 | entre 88 y 110 |
| x14 | entre -110 y -88 |
| x15 | -1 |