



Estructura de Computadores

Practicas ensamblador DLX.

Modificar una matriz, obteniendo otra matriz en la que el resultado de cada posición es el menor de sus adyacentes.

Nombre y apellidos: Carlos Guillén Moreno

DNI: 08367636-Y

Grupo de prácticas: 4

Horario de prácticas: Martes / 8:30

ÍNDICE

- 1) Código realizado.
- 2) Cuestión1: Para el código desarrollado, ¿cual será la ganancia que se obtiene con la segmentación con respecto a un procesador sin segmentar?
- 3) Cuestión2: Indicar para el código desarrollado 5 dependencias de datos en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de dependencia de datos, observaciones (opcional).
- 4) Cuestión3: Indicar para el código desarrollado 1 dependencias de control en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, observaciones (opcional).
- 5) Cuestión4: Indicar para el código desarrollado los riesgos de datos para una iteración del bucle sin unidad de anticipación en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de riesgo de datos, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción i y las cuatro siguientes.
- 6) Cuestión5: Indicar para el código desarrollado los riesgos de datos para una iteración del bucle con unidad de anticipación en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de riesgo de datos, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción i y las cuatro siguientes.
- 7) Cuestión6: Indicar para el código desarrollado al menos un riesgo de control en una tabla con los siguientes campos: instrucción de salto o bifurcación, instrucción a ejecutar en caso de salto efectivo, instrucción a ejecutar en caso de salto no efectivo, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción de salto o bifurcación y dos instrucciones siguientes para ambos casos.
- 8) Cuestión7: Calcular el CPI medio que se obtiene en la ejecución del código desarrollado con y sin anticipación. Calcular la mejora obtenida. Indicar para el código desarrollado el número de ciclos de detención debidos a riesgos de control y a riesgos de datos.

CÓDIGO REALIZADO

```
Guillen Moreno Carlos - Bloc de notas
Archivo Edición Formato Ver Ayuda

.data
MATORIG:      .word 10, 11, 15, 160, 15, 9, 9, 8
               .word 10, 14, 150, 165, 150, 20, 14, 8
               .word 15, 20, 153, 185, 165, 44, 35, 15
               .word 25, 40, 175, 200, 195, 63, 45, 28
               .word 40, 53, 190, 205, 200, 85, 77, 33
               .word 53, 100, 203, 225, 210, 110, 80, 40
               .word 40, 140, 210, 230, 215, 133, 84, 43
               .word 63, 120, 208, 222, 207, 122, 55, 15

MATRESUL:     .space 256
ADYACENTES:    .space 16

.text
main:
    ADDI R28, R0, #4 ;CTE EN 4
    ADDI R29, R0, #1 ;CTE EN 1

    ADDI R17, R0, 0      ;índice fila
    ADDI R18, R0, 0      ;índice columna

    ADDI R26, R0, #32 ;CTE EN 32 (YAM FILA)
    ADD R2, R0, R0 ;índice de la matriz.
    ADD R25, R0, R0

    Estructura:
        AND R6, R0, R0 ;casilla del vector de ocupacion variable.
        OR R5, R0, R0 ;contador de numeros introducidos.
        LW R1, MATORIG(R2)
        ;Comprobar arriba.
        SUB R3, R2, R26
        SGT R4, R0, R3
        BNEZ R4, saltoAbajo
        JAL GuardoElementoEnVector

saltoAbajo:
    ;Comprobar izquierda.
    BEQZ R2, saltoArriba ;si es 0, no tiene izquierda y nos saldriamos de la matriz.
    SUB R3, R2, R0

saltoAbajo:
    ;Comprobar izquierda.
    BEQZ R2, saltoArriba ;si es 0, no tiene izquierda y nos saldriamos de la matriz.
    SUB R3, R2, R0
    ADD R4, R0, R26
    DIV R7, R3, R4
    MULT R7, R7, R4
    SUB R4, R3, R7
    BEQZ R4, saltoArriba
    SUB R3, R2, R28
    JAL GuardoElementoEnVector

saltoArriba:
    ;Comprobar abajo.
    ADD R3, R2, R26
    SLTI R4, R3, #256
    BEQZ R4, saltoIzquierda
    JAL GuardoElementoEnVector

saltoIzquierda:
    ;Comprobar derecha.
    SEQI R3, R2, #252 ;Esta al final, no debe comprobar.
    BNEZ R3, saltoderecha
    ADD R3, R2, R28 ;dividendo
    ADD R4, R0, R26 ;divisor
    DIV R7, R3, R4 ;cociente
    MULT R7, R7, R4
    SUB R4, R3, R7
    BEQZ R4, saltoderecha
    JAL GuardoElementoEnVector

saltoderecha:
    ;Calcular el mínimo con un mejor hasta ahora.
    ADD R6, R0, R0
    LW R12, ADYACENTES(R6)
    OR R10, R0, R29
    J condicionMinimo
    cuerpoMinimo:
        ADD R6, R6, R28
        LW R14, ADYACENTES(R6)
        SLT R13, R14, R12
        BEQZ R13, seguirMinimo
        ADD R12, R14, R0
    seguirMinimo:
        ADD R10, R10, R29
        condicionMinimo:
            SLT R11, R10, R5
            BNEZ R11, cuerpoMinimo
        SW MATRESUL(R2), R12

    OR R10, R0, R29
    J condicionMinimo
    cuerpoMinimo:
        ADD R6, R6, R28
        LW R14, ADYACENTES(R6)
        SLT R13, R14, R12
        BEQZ R13, seguirMinimo
        ADD R12, R14, R0
    seguirMinimo:
        ADD R10, R10, R29
        condicionMinimo:
            SLT R11, R10, R5
            BNEZ R11, cuerpoMinimo
        SW MATRESUL(R2), R12

Fin:
    ADDI R2, R2, 4
    SEQI R23, R18, 7
    BNEZ R23, SALTO2
    ADDI R18, R18, 1
    J SALTO1

SALTO2:
    SUBI R18, R0, 0
    ADDI R17, R17, 1

SALTO1:
    ;Compruebo si he acabado
    SEQI R23, R18, 0
    SEQI R24, R17, 8
    AND R25, R23, R24
    BEQZ R25, Estructura

trap 0
```

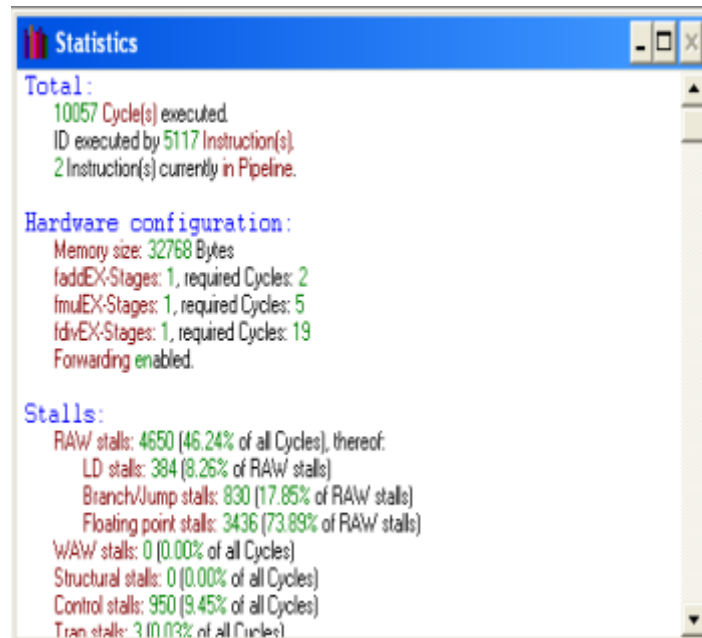
CUESTIÓN 1

Para el código desarrollado, cuál será la ganancia que se obtiene con la segmentación con respecto a un procesador sin segmentar.

Sin segmentación:

1 instrucción → 5 ciclos.

Numero de ciclos sin segmentar = nº de instrucciones * nº de ciclos por instrucción.



Como podemos observar se ejecutan un total de 5117 instrucciones, a 5 ciclos por instrucción suman un total de 25.585 ciclos.

En nuestra ejecución nuestro pipeline ejecuta el programa aprovechando la técnica de segmentación y ejecuta 10.057 ciclos.

En conclusión el segmentado obtiene una ganancia de 15.528 ciclos, siendo por lo tanto un 54, 39% más rápido.

CUESTIÓN 2

Indicar para el código desarrollado 5 dependencias de datos en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de dependencia de datos, observaciones (opcional).

Instrucción i y j	Registro	Tipo Dependencia Datos	Observaciones
Seqi r23, r18, #7 Bnez r23, SALTO2	R23	Dependencia verdadera	La instrucción j intenta leer r23 antes de que a este se le asigne un valor (0 o 1 en este caso), se produce un riesgo RAW y una dependencia de datos.
And r25,r23,r24 Beqz r25, Estructura	R25	Dependencia verdadera	La instrucción j intenta leer r25 antes de que a este se le asigne un valor (0 o 1 en este caso), se produce un riesgo RAW y una dependencia de datos.
Slt r11,r10,r5 Bnez r11,cuerpoMinimo	R11	Dependencia verdadera	La instrucción j intenta leer r11 antes de que a este se le asigne un valor (0 o 1 en este caso), se produce un riesgo RAW y una dependencia de datos.
Lw r14,ADYACENTES(r6) Slt r13, r14, r12	R14	Dependencia verdadera	La instrucción j intenta leer r14 antes de que a este se le asigne un valor (0 o 1 en este caso), se produce un riesgo RAW y una dependencia de datos.
Slt r13,r14, r12 Beqz r13, seguirMinimo	R13	Dependencia verdadera	La instrucción j intenta leer r13 antes de que a este se le asigne un valor (0 o 1 en este caso), se produce un riesgo RAW y una dependencia de datos.

CUESTIÓN 3

Indicar para el código desarrollado 1 dependencias de control en una tabla con los siguientes campos: instrucción i y j, registro para el cálculo del salto efectivo o no, observaciones (opcional).

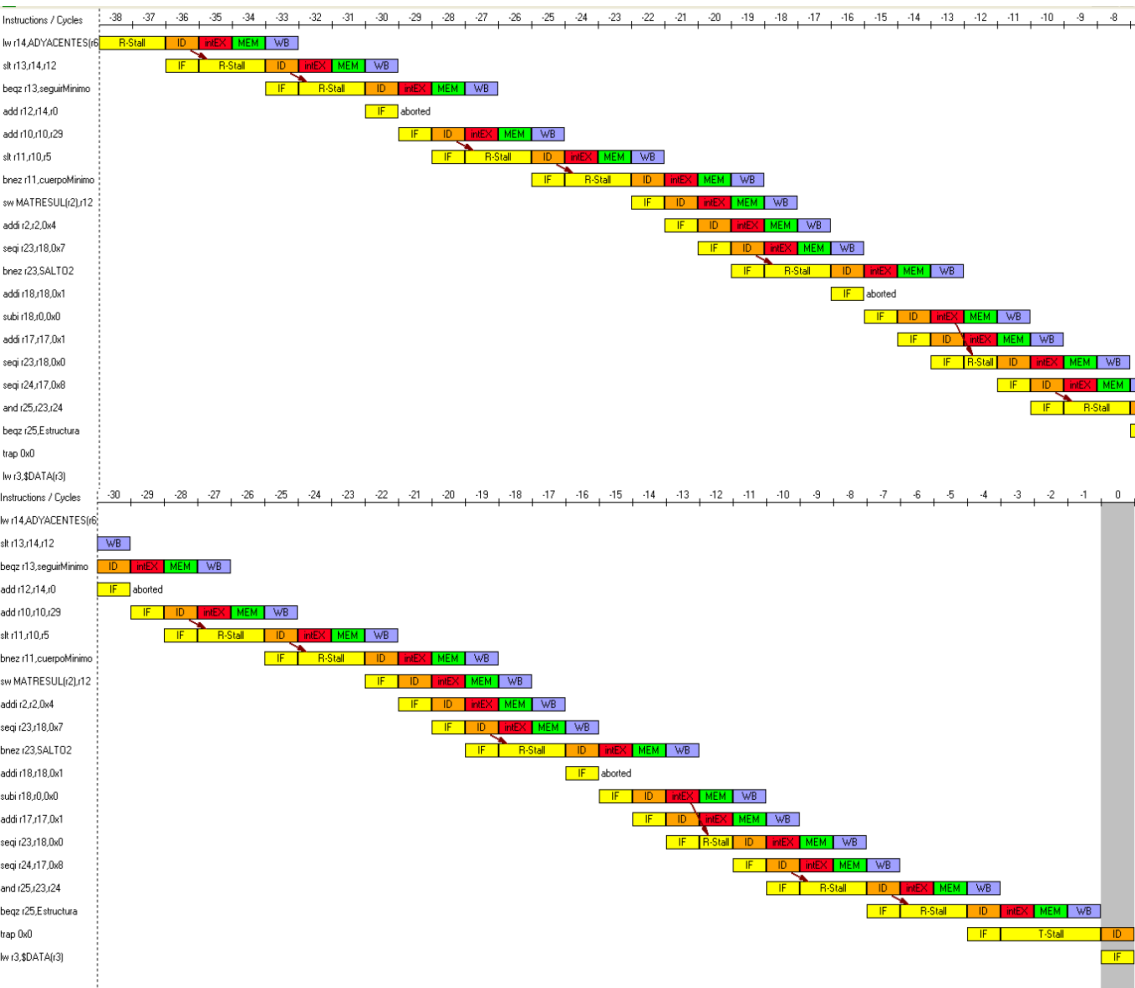
Instrucciones i y j	Registro afectado	Observaciones
Beqz r2, saltoArriba	R2	Si el salto se hace siempre efectivo, las siguientes instrucciones no llegarían a ejecutarse: SUB R3, R2, R0 ADD R4, R0, R26 DIV R7, R3, R4 MULT R7, R7, R4 SUB R4, R3, R7 JAL GuardoElementoEnVector

CUESTIÓN 4

Indicar para el código desarrollado los riesgos de datos para una iteración del bucle principal sin unidad de anticipación en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de riesgo de datos, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción i y las cuatro siguientes.

Instrucciones	Registro	Tipo
Lw r14, ADYACENTES(r6)// Slt r13,r14,r12	R14	RAW
Slt r13, r14, r12// Beqz r13, seguirMinimo	R13	RAW
Add r10, r10, r29// Slt r11, r10, r5	R10	RAW
Slt r11, r10, r5// Bnez r11,cuerpoMinimo	R11	RAW
Seqi r23, r18, #7// Bnez r23, SALTO2	R23	RAW
Subi r18, r0, #0 // Seqi r23, r18, #0	R18	RAW
Seqi r24, r17, #8// And r25, r23, r24	R24	RAW
And r25, r23, r24// Beqz r25, Estructura	R25	RAW

Pantallazo cronograma de WinDLX:

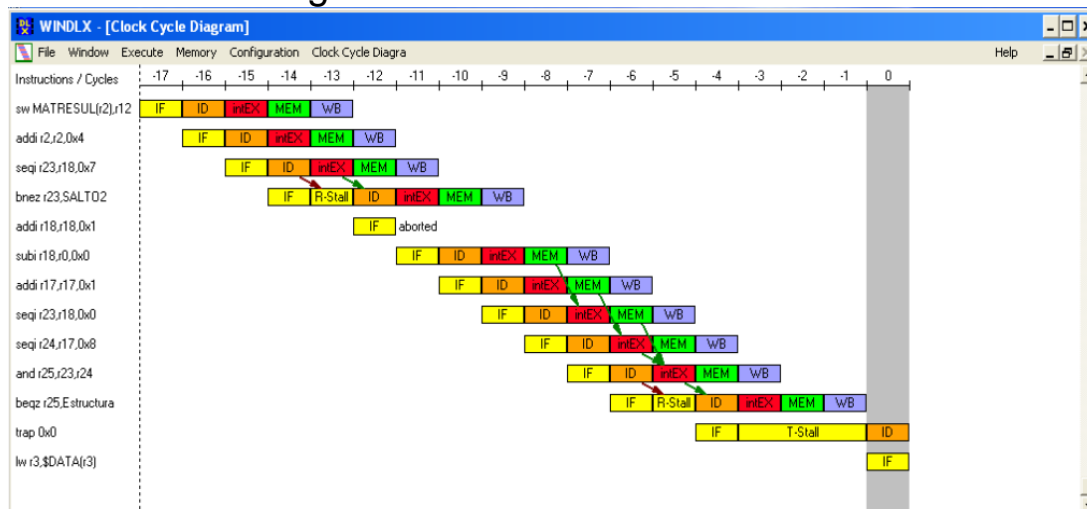


CUESTIÓN 5

Indicar para el código desarrollado los riesgos de datos para una iteración del bucle con unidad de anticipación en una tabla con los siguientes campos: instrucción i y j, registro o zona de memoria, tipo de riesgo de datos, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción i **y las cuatro siguientes**.

Instrucciones	Registro	Tipo
Seqi r23, r18, #7 // Bnez r23, Salto2	R23	RAW
And r25, r23, r24 // Beaz r25. Estructura	R25	RAW

Pantallazo cronograma de WinDLX:

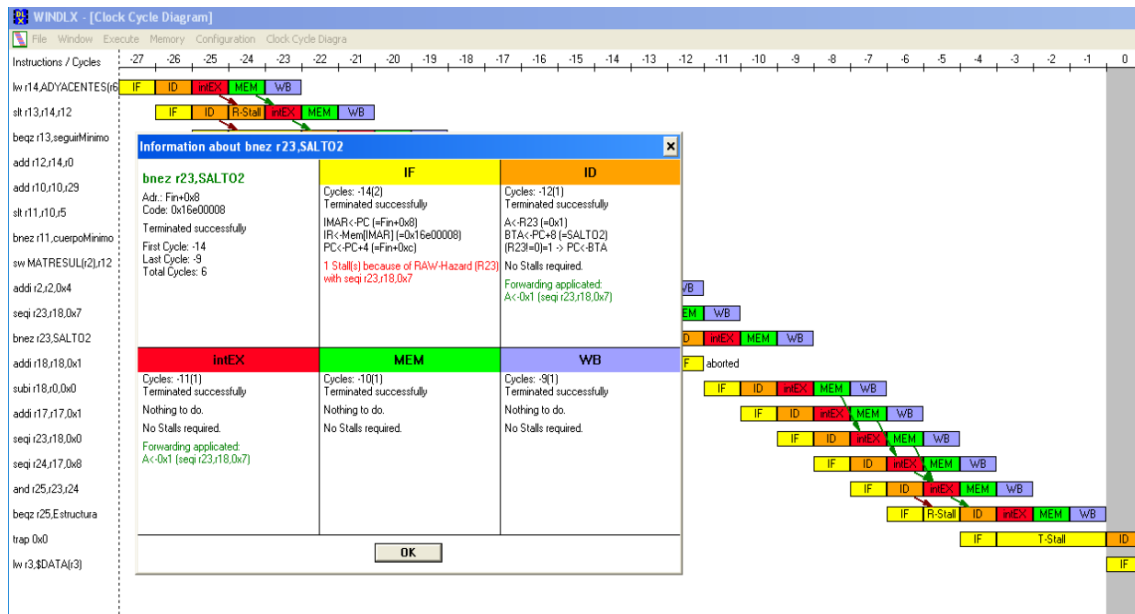


CUESTIÓN 6

Indicar para el código desarrollado al menos un riesgo de control en una tabla con los siguientes campos: instrucción de salto o bifurcación, instrucción a ejecutar en caso de salto efectivo, instrucción a ejecutar en caso de salto no efectivo, captura de pantalla de la ventana de cronograma de WinDLX con la instrucción de salto o bifurcación **y dos instrucciones siguientes para ambos casos.**

Instrucción	Instrucción Salto Efectivo	Instrucción Salto No Efectivo
Bnez R23. SAL TO2	Subi R18. R0. #0	Addi R18. R18. #1

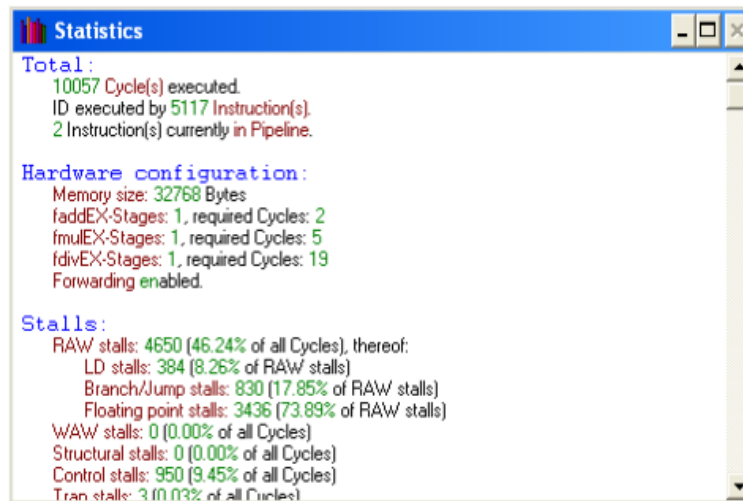
Pantallazo cronograma de WinDLX:



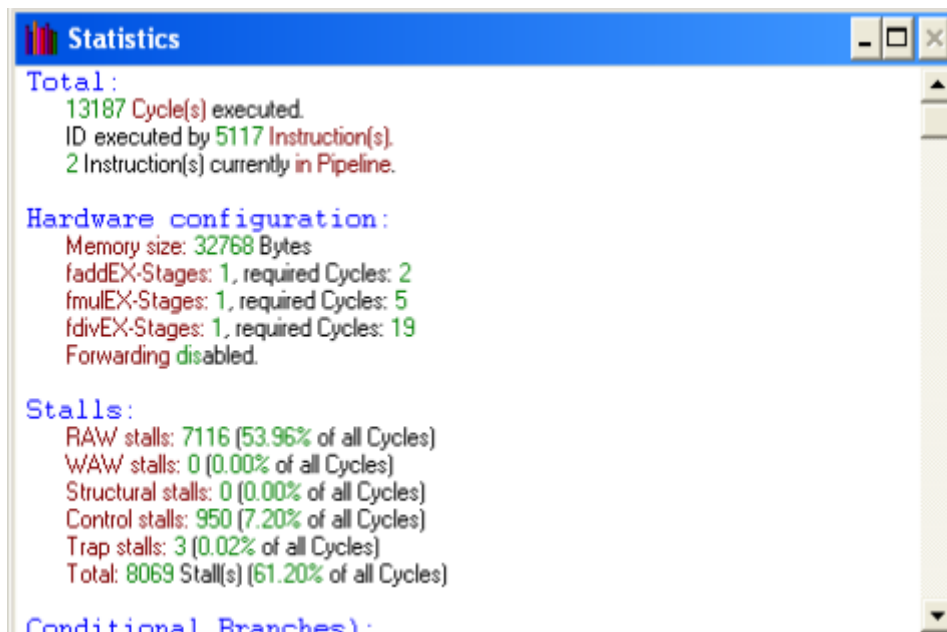
CUESTIÓN 7

Calcular el CPI medio que se obtiene en la ejecución del código desarrollado con y sin anticipación. Calcular la mejora obtenida. Indicar para el código desarrollado el número de ciclos de detención debidos a riesgos de control y a riesgos de datos.

CPI = nº ciclos / nº de instrucciones



CPI con anticipación = $10.057 / 5.117 = 1.96$ ciclos por instrucción



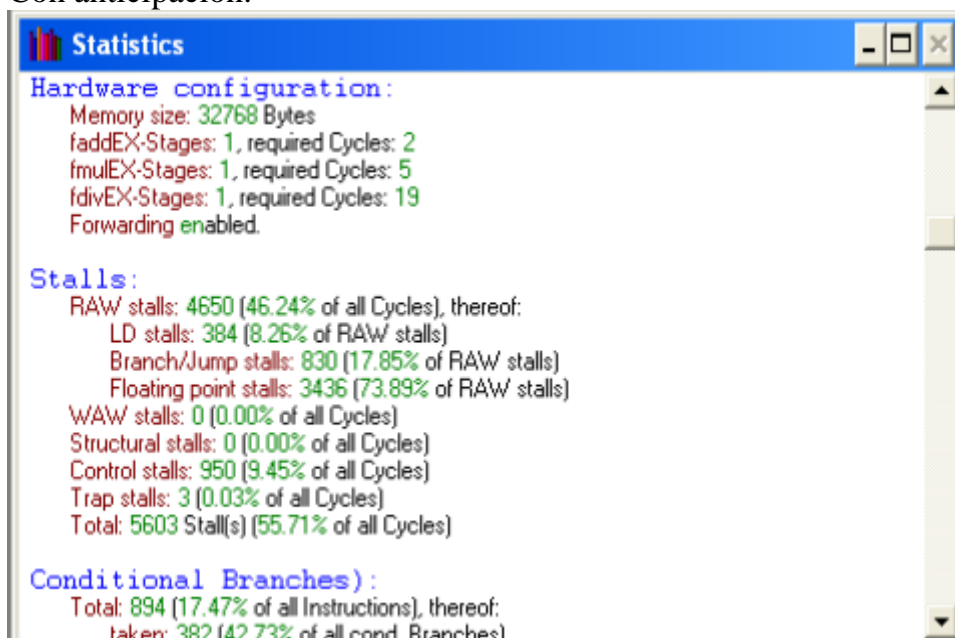
CPI sin anticipación = $13.187 / 5.117 = 2.57$ ciclos por instrucción

Mejora obtenida = (CPI sin Anticipación / CPI con Anticipación -1)*100

Mejora obtenida = $(2.57 / 1.96 - 1) * 100 = 31.12\%$ de mejora.

Ciclos de detención:

Con anticipación.



Ciclos de detención por riesgos de datos: 4650.

Ciclos de detención por riesgos de control: 950.

Ciclos de detención:

Sin anticipación.

```
Stalls:
RAW stalls: 7116 (53.96% of all Cycles)
WAW stalls: 0 (0.00% of all Cycles)
Structural stalls: 0 (0.00% of all Cycles)
Control stalls: 950 (7.20% of all Cycles)
Trap stalls: 3 (0.02% of all Cycles)
Total: 8069 Stall(s) (61.20% of all Cycles)
```

Ciclos de detención por riesgos de datos: 7116.

Ciclos de detención por riesgos de control: 950.