

PROBLEMAS DE ESTRUCTURA DE COMPUTADORES

SEGMENTACIÓN

Rendimiento

1. Ciertos evaluadores del rendimiento del procesador MIPS han determinado que la ruta crítica que fija la longitud del ciclo de reloj en la ruta multiciclo no optimizada corresponde a los accesos a memoria para carga y almacenamiento (no para lectura de instrucciones). Esto ha llevado a replantear la implementación para que la frecuencia de reloj sea de 500 MHz en vez de 750 MHz. inicialmente propuesta. Sin embargo, uno de los ingenieros ha propuesto como solución de compromiso que los ciclos de acceso a memoria se subdividan en dos para, de este modo, admitir la frecuencia de reloj inicial (750 MHz). Usando las proporciones de instrucciones que se muestran a continuación, determina la ganancia de velocidad que se obtiene empleando la máquina de 2 ciclos de acceso a memoria con reloj de 750 MHz, frente a la máquina con 1 ciclo de acceso a memoria y reloj de 500 MHz.

Frecuencia de uso de las diferentes instrucciones para un caso típico (gcc):

- Carga: 22%
- Almacenamiento: 11%
- Aritmético-lógicas: 49%
- Salto condicional, realizando el salto: 4%
- Salto condicional, no realizando el salto: 12%
- Salto incondicional: 2%

2. Usando las mismas proporciones de instrucciones del ejercicio anterior

a) determinar cuál de las tres máquinas siguientes es la más rápida y con qué relación:

M1: ruta multiciclo no optimizada y reloj de 500 MHz

M2: ruta multiciclo en donde la actualización del banco de registros se hace en el mismo ciclo de reloj que la lectura de memoria o la operación ALU. Esto disminuye en 2 el número de estados y rebaja la frecuencia de reloj a 400 MHz al aumentar la longitud del camino crítico

M3: ruta multiciclo como la de M2 donde además el cálculo de dirección efectiva se hace en el mismo ciclo que el acceso a memoria. Esto disminuye en 4 el número de estados anterior y rebaja la frecuencia de reloj a 250 MHz, al alargar la longitud del ciclo para el camino crítico.

3. ¿Hay proporciones de instrucciones que hiciesen que fuese otra la máquina más rápida? Si es así, ¿qué proporciones son esas?

Suponiendo los siguientes tiempos de ejecución de los diversos elementos de una ruta de datos para el MIPS:

Lectura de memoria: 5ns; escritura de memoria: 7ns; lógica de control de la ALU: 2ns; lógica de control principal: 2 ns; lectura BR: 1ns; escritura BR: 2ns; operación ALU: 5ns; extensión signo: 1 ns; multiplicación por 4: 1 ns; multiplexores: 1 ns; Tsetup=0,1 ns; Thold=0,1 ns; CLK_to_Q= 0,1 ns.

- a) Calcular cuál es el mejor tiempo de ciclo para una implementación segmentada.
- b) Si el tiempo de escritura en memoria fuese 15 ns, realizar el mismo cálculo.
4. Sea un procesador segmentado multifuncional en el que hay un 35% de riesgos de LDE, un 8% de riesgos de escritura después de escritura. Sabiendo que los riesgos de LDE tienen como media 3 ciclos de penalización y que los riesgos de EDE tienen una penalización media de 4 ciclos
- ¿cuál es el cpi del procesador?
 - Se modifica el procesador de manera que los riesgos de EDE se eliminen inhabilitando la escritura de la primera instrucción. ¿Cuál es el nuevo CPI?
 - ¿Cuál es el speedup obtenido con la modificación?
5. Supongamos un procesador segmentado de cinco etapas en el que se ejecutan 2 aplicaciones diferentes. La aplicación A tiene un 17% de instrucciones con una penalización de 3 ciclos de reloj y un 12% de instrucciones con una penalización de 2 ciclos de reloj. La aplicación B tiene un 25% de instrucciones con 1 ciclo de penalización.
- ¿Cuál es el ciclo por instrucción de cada una de las aplicaciones?
 - ¿Cuál es ciclo por instrucción promedio del procesador?
 - ¿Cuál es el speedup frente al procesador sin segmentar?
 - ¿Cuál es la eficacia frente al procesador segmentado ideal?
6. Sea un procesador segmentado en cinco etapas. Ejecutamos en este procesador una aplicación con las siguientes características:
- El 18% de las veces, las instrucciones I_{i+1} tienen una dependencia de LDE con las instrucciones I_i . El 29 % de éstas corresponden a instrucciones de load.
 - El 6% de las veces las instrucciones I_{i+2} tienen dependencias de LDE con la instrucción I_i
 - El 25% de las instrucciones son de bifurcación condicional, de las cuales se toman el 35%
- Calcular los ciclos por instrucción del procesador para las siguientes características de procesador:
- Procesador sin técnicas para reducir o eliminar paradas. Los saltos se resuelven en ejecución. Se puede escribir y leer el registro en el mismo ciclo de reloj.
 - Procesador con forwarding. Los saltos se resuelven en decodificación.
 - SPEED UP del segundo caso frente al primero
 - eficiencia de ambos casos
7. Sea un procesador segmentado de cinco etapas que tiene el HW para la detección de riesgo y generación de la parada en decodificación. La resolución de los saltos condicionales se realiza en decodificación. Los riesgos de control se resuelven mediante paradas en decodificación, y tiene implementada la técnica de forwarding. Supongamos que en este procesador las etapas de ejecución y memoria se unen. Como consecuencia el ciclo de reloj aumenta un 35%. Hallar la relación de velocidades entre la versión original del procesador y la modificada. Supóngase que en el 25% de los casos existen dependencias LDE entre un load y la instrucción que le sigue y que el 33% son saltos condicionales.

Ruta de datos y cronogramas:

8. Sobre la estructura de del control mult ciclo de nuestro procesador no optimizado, indicar los valores de todas las líneas de datos y los registros, en los siguientes casos:

a.- Ejecución de la instrucción OR \$1, \$2, \$3, al final de cada una de sus cuatro etapas.

Estado actual de la máquina:

\$1=HEX(00000017)

\$2=HEX(001100FF)

\$3=HEX(FF000345)

PC=HEX(00003400)

b.- Ejecución de la instrucción JAL 1024 (1024 en decimal) al final de las tres etapas.

Segmentación:

9. El siguiente fragmento de código se ejecuta en un MIPS con segmentación:

```
sub $1,$2,$3
add $4,$5,$6
sub $5,$4,$8
add $7,$2,$3
add $9,$7,$3
lw $1,10($6)
add $3,$1,$4
sub $6,$7,$8
```

Suponiendo que un dato se puede escribir en un banco de registros y leer su nuevo valor en el mismo ciclo:

a) Calcular el número de ciclos necesarios para ejecutar este código si no existe la posibilidad de anticipar operandos o reordenar el código.

b) Calcular el número de ciclos si existe anticipación de operandos, pero no reordenación de código.

c) Tratar de reordenar el código para conseguir que el número de ciclos sea mínimo, si no hay anticipación de operandos. ¿Cuántos ciclos son necesarios en este caso?

10. Sobre la estructura del computador MIPS segmentado, se ejecuta la siguiente secuencia de instrucciones:

ADD \$1, \$2, \$3

SUB \$4, \$2, \$3

AND \$5, \$2, \$3

OR \$6, \$2, \$3

Si la instrucción ADD está colocada en la dirección de memoria 00002000 (Hex), y el contenido de los registros es:

\$1=00000005

\$2=00000004

\$3=00000002

PC=00002000

- a) Señalar el contenido de los siguientes registros al cabo de 4 ciclos de reloj: IF/ID.pc, ID/EX.pc, EX/MEM.pc, ID/EX.A, ID/EX.B, EX/MEM.ALUout, MEM/WB.ALUout, ID/EX.rd, EX/MEM.rd, MEM/WB.rd.
- b) Indicar también el contenido de cada uno de los registros que almacena el control.

11. Se tiene el siguiente fragmento de código del MIPS.

OR \$4, \$8, \$9

(100 instrucciones con dependencias internas de datos, pero no detención de pipeline)

ADD \$5, \$6, \$7

OR \$4, \$1, \$6

(100 instrucciones con dependencias internas de datos, pero no detención de pipeline)

LW \$10, 20(\$4)

ADD \$8, \$9, \$10

SUB \$6, \$8, \$1

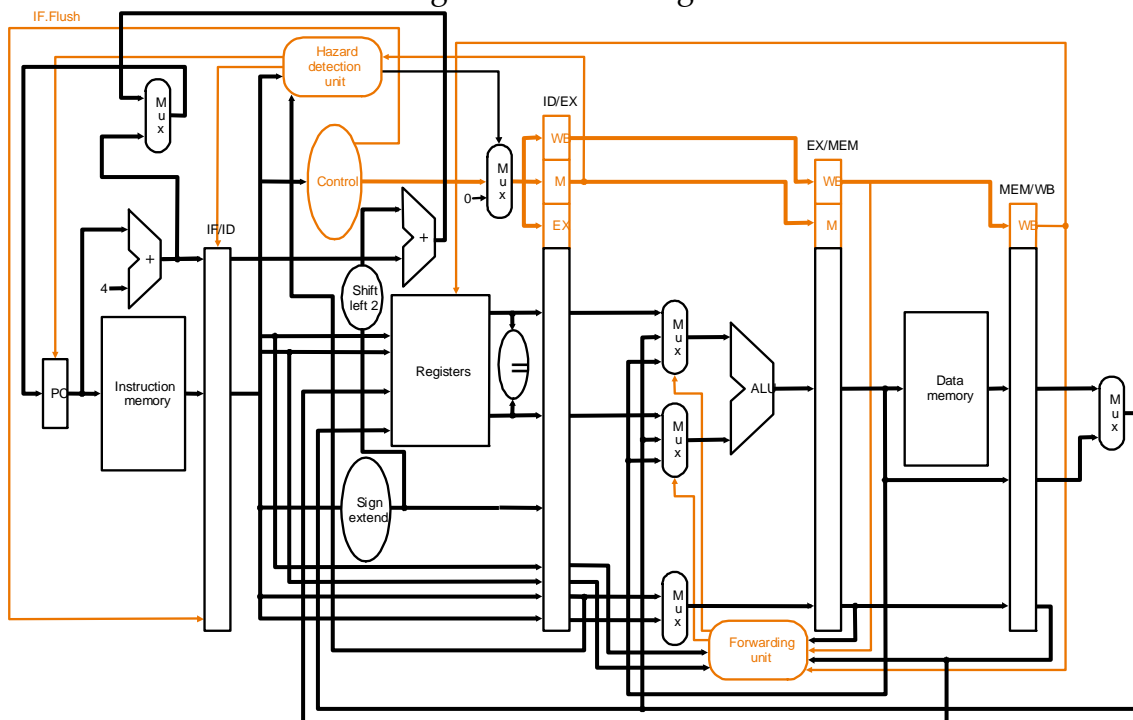
OR \$1, \$3, \$5

BEQ \$1, \$2, 1000

ADD \$1, \$5, \$6

- a) Cuánto tardaría en ejecutarse, si cuando hay un conflicto de control se espera a que se solucione y se supone que el destino de salto se conoce en la fase ID y la comparación de salto también. Suponer que no hay anticipación de operandos.
- b) Si se tuviesen saltos retardados, ¿cómo se podría rellenar el hueco de salto para disminuir el tiempo de ejecución? ¿Cuál sería este tiempo?
- c) ¿y si además de salto retardados tuviésemos anticipación de operandos?
- El destino del salto es la instrucción OR \$4, \$1, \$6, y el salto se repite 100 veces.

12. Considerar la ruta de datos segmentada de la figura.



- a). En esta ruta puede ocurrir simultáneamente un intento de anulación de instrucción (*flush*) debido a un salto y un intento de parada (bloqueo o *stall*) debido a una dependencia

de datos que afecta a una instrucción LW. Mostrar una secuencia de código donde se manifieste la situación descrita.

b). Mostrar cuál es el efecto del bloqueo y de la anulación sobre el *pipeline*, indicando cómo se detectan y a qué señales y registros afectan.

c). Las acciones realizadas en el bloqueo y anulación simultáneos, ¿colaboran o tienen efectos que entran en conflicto? Si hay colaboración, ¿cómo trabajarían juntos? Si hay conflicto, ¿qué acción debe tener prioridad

Ensamblador:

13. El desenrollado de bucles es una técnica que permite elevar la velocidad de ejecución de códigos iterativos. Para el siguiente ejemplo en una arquitectura pipeline,

Código de bucle original	Código de bucle desenrollado una vez
<pre>bucle: lw \$t0, 0(\$s1) add \$t0, \$t0, \$s2 sw \$t0, 0(\$s1) addi \$s1, \$s1, -4 bne \$s1, \$zero, bucle</pre>	<pre>bucle: lw \$t0, 0(\$s1) add \$t0, \$t0, \$s2 sw \$t0, 0(\$s1) lw \$t1, -4(\$s1) add \$t1, \$t0, \$s2 sw \$t1, -4(\$s1) addi \$s1, \$s1, -8 bne \$s1, \$zero, bucle</pre>

comparar la diferencia de rendimiento, mediante contabilización de ciclos de ejecución, para los tres casos siguientes:

- bucle original
- bucle desenrollado
- bucle desenrollado y reordenado para evitar bloqueos si es posible

(Suponer que la instrucción bne toma la decisión de salto en la etapa ID, y se dispone de circuitería de anticipación para resolver dependencias de datos).

14. Supongamos que los dos fragmentos de código ensamblador siguientes han sido generados por sendos compiladores (C1 y C2) para un mismo código fuente.

Compilador 1	Compilador 2
<pre>add \$t0, \$zero, \$zero add \$t1, \$a1, \$zero loop: bge \$t1, \$a2, retorno sll \$t2, \$t1, 2 add \$t2, \$t2, \$a0 lw \$t3, 0(\$t2) add \$t0, \$t0, \$t3 add \$t1, \$t1, \$a3 j loop retorno: add \$v0, \$t0, \$zero jal \$ra</pre>	<pre>add \$v0, \$zero, \$zero sll \$a2, \$a2, 2 sll \$a3, \$a3, 2 add \$t1, \$a1, \$zero bge \$t1, \$a2, retorno loop: add \$t2, \$t1, \$a0 lw \$t3, 0(\$t2) add \$v0, \$v0, \$t3 add \$t1, \$t1, \$a3 blt \$t1, \$a2, loop retorno: jal \$ra</pre>

- Inicialmente \$a1=0, \$a2=5, \$a3=1. Determinar la frecuencia de cada tipo de instrucción rellenando la tabla siguiente:

Tipo de instrucción	CPI	Frecuencia en C1	Frecuencia en C2
ALU Aritmética			
ALU lógica			

Memoria ("load")			
Memoria ("sw")			
Bifurc. Cond. SI			
Bifurc. Cond. NO			
Bifurc. Incond.			
Total			

(Suponemos que la instrucción "sll" - shift logical left - se ejecuta en 3 ciclos. El registro \$zero se corresponde con \$0 y siempre vale 0)

- Calcular el CPI de cada programa usando las frecuencias obtenidas en a)
- Si P1 se ejecuta en una máquina de 500 MHz y P2 en una de 400 MHz ¿qué combinación h/w-s/w es más rápida? ¿Cuál es la ganancia?

Operaciones multiciclo:

15. Sea un procesador DLX segmentado con las siguientes características:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa de decodificación.
- La detección de todo tipo de riesgos y generación de paradas se realiza en la etapa de decodificación.
- En el caso de los riesgos WAW se produce una parada hasta que la instrucción ya lanzada entre en la etapa de memoria.
- Los registros r1 y r2 contienen inicialmente el valor unidad.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	No
FP MUL	1	3	No
FP DIV	1	5	No
Int ALU	1	1	No

En este procesador se ejecuta el siguiente fragmento de código

```

L0:  ADDD F2,F4,F0
      SD F2,0(r1)
      DIVD F4,F4,F0
      SUBI r1,r1,#1
      ADDD F4,F0,F2
      MULD F2,F2,F4
      BEQZ r1,L2
L1:  LD F2,0(r1)
L2:  LD F4,0(r1)
      ADDD F0,F2,F4
      DIVD F4,F2,F0
      ADDD F8,F6,F8
      ADDD F4,F0,F6

```

```

    ADDI r1,r1,#1
    ADDI r2,r2,#1
    BNEZ r1,L3
    ADDD F0,F0,F2
L3:  ADDD F2,F0,F4
end

```

- Representar el diagrama instrucción-tiempo e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- Calcular los CPI (ciclos por instrucción) resultantes de la ejecución del código.

16. Sea un DLX segmentado con las siguientes características:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa de decodificación.
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDD	1	3	Sí
FP SUBD	1	3	Sí
FP MULD	1	4	Sí
FP DIVD	1	5	No
INT ALU	1	1	No

```

    ADDI r3,r0,#3
L1 : SUBD F2,F6,F8
    SUBD F4,F8,F6
    SD 0(r3),F4
    DIVD F2,F4,F8
    ADDD F2,F8,F8
    SUBI r3,r3,#1
    DIVD F6,F4,F8
    MULD F4,F2,F6
    SUBD F10,F2,F6
    LD F4,0(r3)
    ADDD F0,F4,F2
    BNEZ r3,L1
    MULD F4,F2,F2
end

```

- Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.

- b) A la vista del diagrama obtenido, y sin necesidad de desarrollar todas las iteraciones, calcular el número de ciclos que toma la ejecución completa del código.

17. El siguiente fragmento de código se ejecuta en un DLX con segmentación:

```

LOOP: LD F2, 0(R1)
      MULTD    F4, F2, F0
      LD F6, 0(R2)
      ADDD     F6, F4, F6
      SD      0(R2), F6
      ADDI    R1, R1, 8
      ADDI    R2, R2, 8
      SGTI    R3, R1, DONE
      BEQZ    R3, LOOP
      NOP

```

Suponiendo que:

- un dato se puede escribir en un registro y leer su valor en el mismo ciclo
- se dispone de lógica de cortocircuito
- permite la coexistencia del store y de las operaciones aritméticas en las etapas de M y W
- los saltos se resuelven en la etapa de decodificación
- se emplea un *branchdelay slot* de una instrucción
- las unidades funcionales tienen las siguientes características:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	4	No
FP MUL	1	5	No
Int ALU	1	1	No

- a) Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- b) Determinar el CPI teniendo en cuenta que `done=0x1000` y `r1=0x0100` al inicio de la ejecución.

18. dlx segmentado de cinco etapas con las siguientes características:

- un dato se puede leer y se escribir en el mismo ciclo de reloj
- Existe anticipación de operandos
- Las dependencias de LDE se detectan en la primera etapa de cada unidad funcional y se resuelven mediante cortocircuito cuando los operandos están disponibles. Esta es la característica del DLX que lo hace diferente a aquellos en los que las dependencias de LDE se detectan en decodificación.
- Los riesgos estructurales se detectan y se resuelven mediante espera en la última etapa de cada unidad funcional
- Los riesgos de EDE entre dos instrucciones A y B tal que A precede a B se resuelven mediante inhibición de escritura de la instrucción A.
- Las unidades funcionales del procesador son:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	2	si
FP MUL	1	5	si

Int ALU	1	1	No
---------	---	---	----

Sabiendo que el siguiente fragmento de código se ejecuta sobre dicho procesador

```
LD F10,0(R1)
MULD F4,F0,F10
LD F12,0(R2)
ADD F2,F12,F4
LD F4 8(R1)
MULD F12,F4,F12
LD F14,168R1)
```

- Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas.
- Determinar el CPI

19. Supongamos un computador como el DLX básico segmentado en cinco etapas que posee anticipación de operandos y utiliza saltos retardados con un "delay-slot" de un ciclo, pero que posee una sola memoria cache para el almacenamiento de instrucciones y datos. Por esta razón no resulta posible leer una instrucción y realizar la lectura o escritura de un dato en el mismo ciclo de reloj. Las escrituras en el banco de registros se hacen en la primera mitad de la fase WB, mientras que las lecturas se hacen en la segunda mitad de la fase WB.

Supongamos que este computador ejecuta el siguiente programa:

```
BUCLE:  LW R2,X(R6)
        LW R3,Y(R6)
        SUB  R2,R2,R3
        ADD  R2,R2,R1
        SUB  R6,R6,#4
        SW Z(R6),R2
        BNEZ R6,BUCLE
        ADD  R1,R1,#1
        SUB  R3,R3,R7
```

Se supone que el valor inicial de R6 es 2000.

- Construye el diagrama de tiempo correspondiente a la primera iteración del bucle, indicando sobre el diagrama los cortocircuitos que se activan.
- Calcula el valor del CPI
- Si el computador trabaja con una frecuencia de 1 GHz, determina el rendimiento en MIPS.

20. En un DLX con segmentación ejecutamos el siguiente fragmento de código:

```
LOOP :  DIVD F0,F4,F2
        ADDD F0,F2,F6
        DIVD F8,F8,F2
        ADDI R3,R3,#1
        ADDD F2,F6,F8
        MULD F6,F8,F0
```

```

LD F2, 0(R3)
SD 0(R5), F6
MULD F2,F6,F8
ADDD F6,F8,F0
ADDD F0, F2, F2
SUBI R5,R5,#1
BNEZ R5,LOOP
ADDD F4,F2,F2
SUBD F6,F0,F0
END : SUBI R3, R3, #1

```

Se supone que:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos se resuelven en la etapa EX y se espera a que se resuelvan antes de lanzar la siguiente instrucción.
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- Inicialmente r5=1000.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDD	1	2	Sí
FP SUBD	1	2	Sí
FP MULD	1	3	Sí
FP DIVD	1	4	No
INT ALU	1	1	No

Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados así como las paradas producidas y sus causas. A la vista del diagrama obtenido, indicar el número de ciclos por instrucción (CPI) en régimen estacionario.

21. El siguiente fragmento de código se ejecuta en un DLX con segmentación de 7 etapas:

```

LOOP: LD F2, 0(R1)
      MULTD    F4, F2, F0
      LD F6, 0(R2)
      ADDD    F6, F4, F6
      SD    0(R2), F6
      ADDI   R1, R1, 8
      ADDI   R2, R2, 8
      SGTI   R3, R1, DONE
      BEQZ   R3, LOOP

```

NOP

Suponiendo que:

- Tanto la memoria de datos como la de instrucciones están segmentadas en dos etapas
- un dato se puede escribir en un registro y leer su valor en el mismo ciclo
- se dispone de lógica de cortocircuito tanto secuencial como combinacional
- La detección de todo tipo de riesgos (estructurales y LDE) y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- los saltos se resuelven en la etapa de decodificación
- se emplea un *branchdelay slot* de una instrucción
- una inst.aritmetica y un store pueden coexistir en las etapas de MEM y WB
- las unidades funcionales tienen las siguientes características:

UF	Cantidad	Latencia	Segmentación
FP ADD	1	4	si
FP MUL	1	5	si
Int ALU	1	1	No

Determinar el CPI teniendo en cuenta que `done=0x1000` y `r1=0x0100` al inicio de la ejecución.
Reordenar/modificar el código para que el CPI sea mínimo.

22. Sea un dlx etapa de ejecución multiciclo con las siguientes características:

- Un dato se puede escribir y leer de un registro en el mismo ciclo de reloj.
- Los saltos se resuelven en la etapa de decodificación
- Las detecciones riesgos de datos y generación de paradas se realiza en la etapa de decodificación.
- La detección de riesgos WAW se realiza en la etapa de DEC y se produce una parada hasta que la instrucción ya lanzada entre a la etapa de memoria.
- Forwardingcombinacional
- Salto retardado de 1 instrucción

UF	CANTIDAD	LATENCIA	SEGMENTACIÓN
FPADD	1	2	NO
FP DIV	1	10	NO
FPMUL	1	5	NO
INTALU	1	1	NO

Y el siguiente fragmento de código:

```
LOOP:   LD F6, 0(R2)
        MULF F8,F6,F0
        ADDI R2,R2,#1
        LD F2, 0(R2)
        DIVF F8,F2,F8
        SD 0(R2), F8
        MULF F8,F2,F0
        ADD  F8,F4,F6
```

```
ADDI R3,R3,#8
SGTI R4,R3,DONE
BNEZ R4,LOOP
NOP
```

- a) CPI del pipeline, suponiendo que el bucle se ejecuta muchas veces.
- b) Suponiendo que el hardware de la maquina puede inhibir la escritura en el registro destino de una instrucción ya lanzada. Explica cómo usar esta característica para reducir las penalizaciones debidas a los riesgos EDE. Calcula el nuevo valor de CPI