Arquitectura de Computadores
Problemas (hoja 3). Curso 2015-16

1. Sea un computador superescalar similar a la versión Tomasulo del DLX capaz de lanzar a ejecución dos instrucciones sin dependencias por ciclo de reloj (incluyendo operaciones enteras). El siguiente código implementa la operación DAXPY (Y = a * X + Y) sobre un vector de 100 elementos:

```
foo:    LD      F2, 0(R1)  ; leer X(i)
        MULTD   F4, F2, F0 ; a*X(i)
        LD      F6, 0(R2)  ; leer Y(i)
        ADDD    F6, F4, F6 ; a*X(i)+Y(i)
        SD      0(R2), F6  ; almacenar Y(i)
        ADDI    R1, R1, #8 ; incrementar el índice de X
        ADDI    R2, R2, #8 ; incrementar el índice de Y
        SGTI    R3, R1, done
        BEQZ    R3, foo
```

Las unidades funcionales están segmentadas y se asume que existen suficientes estaciones de reserva.

a) Desenrolla el bucle DAXPY para obtener 4 copias del cuerpo y planificarlo de tal forma que se alcance el mayor IPC posible.

b) ¿Cuántos ciclos emplea cada iteración del bucle original?

c) ¿Cuál es la ganancia producida por el desenrollado del bucle?

2. Sea un computador superescalar similar al del ejercicio anterior, pero además con ejecución especulativa. Las unidades funcionales segmentadas tienen las siguientes características:

Unidad funcional	**Cantidad**	**Latencia**
FP mult	1	7 ciclos
FP add	1	5 ciclos
ALU Int	1	1 ciclos

Se supone que la latencia de memoria es 1 ciclo de reloj.Considerando la misma versión del código DAXPY que en el ejercicio anterior muestra el diagrama instrucción – tiempo para las dos primeras iteraciones del bucle original. Se supone que el salto es correctamente predicho y tomado en la primera iteración.

3. Sea un procesador segmentado con planificación dinámica mediante el algoritmo de Tomasulo

- Los datos que se escriben en la etapa de escritura se pueden usar en la etapa de ejecución de una instrucción en el mismo ciclo.
- Las instrucciones SGTI, la BNEZ y NOP tienen tratamiento de instrucciones enteras.
- Los LOAD Y STORE tienen una latencia de dos ciclos, utilizan su propia unidad funcional no segmentada
- Hay un solo bus de datos común (CDB)
- La estructura del procesador tiene las siguientes características:

UF	CANTIDAD	LATENCIA	SEGMENTADA
FP ADD	1	2	SI
FP DIV	1	8	SI
FP MUL	1	4	SI
INT ALU	1	1	SI
MEMORIA	1	2	NO

ESTACIONES RESERVA	CANTIDAD
FP ADD	2
FP DIV	2
FP MUL	2
INT ALU	3
LOAD	2
STORE	2

Dado el siguiente fragmento de programa:

```
      ADDI R1,R0,#DIR
      LD   F0,0(R7)
LOOP: LD   F4,0(R1)
      DIVD F8,F4,F0
      SUBI R1,R1, #4
      LD   F2,0(R1)
      MULD F8,F2,F8
      SD   0(R1),F8
      SUBI R3,R3,,#8
      SGTI R5,R3,#1000
      BEQZ R5,LOOP
      NOP
```

a) Representa el diagrama instrucción – tiempo para todo el fragmento de programa considerando sólo la primera iteración, indicando en cada caso el tipo de parada que se produce

b) Indica el diagrama instrucción – tiempo para todo el fragmento de programa considerando sólo la primera iteración, suponiendo un superescalar con las mismas unidades funcionales ya vistas, que lanza un par de instrucciones de cualquier tipo en un ciclo de reloj. En caso de dependencia de datos entre dos instrucciones de un par se congela el lanzamiento de la segunda. Se supone que existen dos buses comunes de datos, que el banco de registros puede realizar dos escrituras en cada ciclo de reloj, y que tiene suficientes estaciones de reserva para que no se produzcan paradas.

4. Considera la ejecución del código vectorial para la operación DAXPY explicado en clase en el VMIPS.

a) Determina el tiempo de cálculo para vectores de 230 componentes, teniendo en cuenta la penalización de la ejecución por bloques (strip-mining).

b) Deduce, para esta operación el valor de R_∞ (en MFLOPS) y de $N_{1/2}$. Asume un tiempo de ciclo de 1.5 ns.

5. Para estudiar el efecto que produce la adición al VMIPS de dos nuevos pipes de carga/almacenamiento, repite el problema anterior considerando que existen tres pipes de carga/almacenamiento.

6. Se desea realizar el cálculo:

$$A(I) = [B(I)*C(I)] + [D(I)*S], \text{ para } I=1..N$$

donde A, B, C y D son vectores de N componentes residentes en memoria y S es un escalar almacenado en un registro, en una máquina similar al VMIPS, pero con registros vectoriales de longitud p.

a) Suponiendo que p es igual a N, dibujar el diagrama que muestra la realización del anterior cálculo en el menor tiempo posible. Señala los instantes de inicio y final de cada operación y halla el tiempo total del cálculo.

b) Utiliza iterativamente la secuencia de operaciones realizada en el apartado a para hallar el tiempo de cálculo en el caso de que N=265 y p=32. Ten en cuenta la penalización producida por el procesamiento por bloques (strip-minig).

c) ¿Cuál es el rendimiento en MFLOPS para el cálculo efectuado en el apartado b, si el ciclo de reloj es de 2 ns?

d) ¿Para qué valor de N se alcanza el rendimiento mitad del hallado en el apartado c?

7. Dado el fragmento de programa

```
for (i=0; i<50; i++){
    for (j=0; j<50; j++)
        c[i,j])=(a[i,j]*b[i,j])+5;
```

```
}
```

donde a, b y c son matrices de 50x50, almacenadas en memoria por filas,

a) Tradúcelo al correspondiente fragmento de código máquina simbólico del VMIPS sabiendo que los cálculos en cada iteración del bucle interno se implementan mediante instrucciones vectoriales, y los del bucle externo mediante iteraciones sucesivas.

b) Suponiendo que el VMIPS tuviera dos pipes de carga/almacenamiento, determina el tiempo de ejecución del programa obtenido en el apartado anterior, teniendo en cuenta la penalización de la ejecución por bloques.

c) Halla el rendimiento del anterior algoritmo si el nº de filas crece indefinidamente y el nº de columnas se mantiene constante.

8. Dado el bucle

```
for (i=1;i<=265;i++) {
    a[i]=b[i]+c[i];
    if(a[i]==b[i])
            d[i]=a[i]*3;
    b[i]=a[i]-5;
}
```

se pide:

a) Tradúcelo al correspondiente bloque de código vectorial del VMIPS.

b) Determina el tiempo de ejecución, teniendo en cuenta la penalización de la ejecución por bloques. Se supone que las operaciones de comparación se ejecutan en un pipe de cuatro etapas que no puede encadenarse.

c) Si la mitad de las componentes de los vectores a y b son iguales, determina el rendimiento.

9. Supongamos un VMIPS con dos pipes de carga/almacenamiento y otra máquina similar, VMIPS2, que presenta las siguientes diferencias respecto de la primera:

- Mientras que el VMIPS trabaja con un tiempo de ciclo de 1.25 ns, el VMIPS2 lo hace con un tiempo de ciclo de 1 ns.
- Para lograr la mejora del tiempo de ciclo, los pipes del VMIPS2 están segmentados en más etapas. El pipe de suma tiene 8 etapas y el pipe de multiplicación tiene 10 etapas,
- El VMIPS2 tiene 2 pipes de suma y dos de multiplicación. Debido a la complejidad adicional del control de tales pipes, T_{loop} vale 45 ciclos, en lugar de los 15 del VMIPS.

Consideremos la ejecución del siguiente bucle:

```
for (i=0;i<n;i++)
    A[i]=x*A[i]+y*A[i];
```

Se pide:

a) Halla la razón entre los rendimientos asintóticos (en MFLOPS) del VMIPS y el VMIPS2.

b) Estudia si existe algún valor de n tal que por debajo de él es más rápida una de las máquinas, mientras que por encima de él es más rápida la otra.

10. Supongamos dos matrices, A[29,50] y B[50,29], que están almacenadas por filas en la memoria de un computador vectorial con todas las características del VMIPS, pero que posee dos pipes del carga/almacenamiento y dos pipes de suma. Se desea realizar la operación:

```
for (i=1;i<29;i++) {
    for (j=0;j<50;j++)
        C[i,j] = A[i-1,j] + B[j,i-1] * 5;
}
```

Se pide:

a) Traduce el anterior programa al lenguaje máquina simbólico del VMIPS de tal manera que se trate de optimizar el tiempo de ejecución.

b) Dibuja el diagrama de tiempo correspondiente a una ejecución del bucle interno. Determina el tiempo de cálculo y el rendimiento en MFLOPS del programa anterior.

c) ¿Qué porcentaje del rendimiento asintótico se obtiene en la operación?

NOTAS.- Se supone que las direcciones iniciales de las matrices A, B y C están almacenadas en los registros Ra, Rb y Rc, respectivamente. En el enunciado se asume que la notación X[M,N] representa una matriz X de M filas y N columnas. La memoria está entrelazada con 16 bancos. El tiempo de ciclo es 2 ns.

11. Consideremos el vector A, de 37 componentes, y la matriz B, de 37 filas por 29 columnas. Ambas estructuras están almacenadas en la memoria de un VMIPS, a partir de las direcciones apuntadas por los registros Ra y Rb, respectivamente. La matriz B está almacenada por filas. Supongamos que se desea realizar el cálculo:

```
for (i=0;i<37;i++) {
    B[i,2] = A[i] + (B[i,4]*9) + 2.5;
    if ( A[i]< B[i,2])
        C[i] = B[i,2];
}
```

a) Escribe un programa en el leguaje máquina simbólico del VMIPS que ejecute el cálculo en el menor tiempo posible. Se supone que las operaciones de comparación de vectores se ejecutan en un pipe de cuatro etapas que no puede encadenarse.

b) Dibuja el diagrama temporal, determina el tiempo de cálculo del programa y calcula el porcentaje del rendimiento asintótico que se alcanza.

c) Repite los pasos anteriores, pero considerando que en lugar de un sólo pipe de carga almacenamiento existieran dos, así como dos pipes de suma y dos de multiplicación.

12. Supongamos que se desea calcular:

D = f2 (f1 (A,B), f1(A,C))

donde A, B, C y D son vectores de 1000 componentes en punto flotante, y f1 y f2 son operadores vectoriales.

El cálculo se realiza en un computador vectorial con las siguientes características:

- Ciclo de reloj: 2 ns
- 8 registros vectoriales de 128 componentes
- 2 UF segmentadas lineales capaces de implementar f1 y f2, respectivamente. Tiempo de cálculo de f1: 20 ns. Tiempo de cálculo de f2: 16 ns
- 2 pipes de carga/almacenamiento segmentados en 12 etapas
- Instrucciones vectoriales para ejecutar f1 y f2

En caso de necesitar algún parámetro no especificado en el problema puedes definirlo libremente justificando la decisión. Se pide:

a) Construye el diagrama de tiempo para ejecutar el cálculo en el menor tiempo posible. Calcula el rendimiento de la operación y el rendimiento asintótico en MFLOPS.

b) Repite el apartado a, pero suponiendo que existen 2 pipes que implementan f1 y 3 pipes de carga/almacenamiento.