

Unidad III: Paralelismo en monoprocesadores

Trabajo Práctico N° 2

Dependencias de un programa y Procesadores Superescalares

1.- Estudie el siguiente fragmento de programa y marque los tipos de dependencias que detecte.

```

S1:   A = B * C
S2:   X = Y + V
S3:   V = A + 2
S4:   A = A + 1
S5:   C = B + X
S6:   B = C + A

```

2.- Para el bucle del siguiente fragmento de código, arme los grafos de dependencia de datos que detecte y complete los valores *T* y *C* de los arcos vinculantes.



```

fibonacci[1] = 1;
fibonacci[2] = 2;
for (int i = 3; i <= 1000; i++){
S1: fibonacci[i] = fibonacci[i-1] + fibonacci[i-2];
S2: acum = acum + fibonacci[i];
S3: suma[i] = fibonacci[i] * 100;
S4: prom[i] = suma[i] / count;
S5: data[i] = acum + data[i-1];
S6: total[i] = data[i] / prom[i];
S7: count = total[i];
}

```

Suponga los siguientes parámetros

Suma: 2 ciclos
 Multiplicación: 4 ciclos
 División: 5 ciclos
 Asignación: 1 ciclo

3.- Basándose en el ejercicio anterior, halle:

- los ciclos de dependencia que se forman
- calcule el tiempo de ejecución de cada uno de ellos
- el tiempo de dependencia crítico del loop
- la aceleración total obtenida.

4.- Para el siguiente fragmento de código, arme los grafos de dependencia de datos que detecte muestre todas las trazas existentes y calcule la aceleración que la traza crítica produce por sobre la ejecución secuencial del bucle. En cuántos procesadores podría ejecutarse el loop luego de optimizarlo?

```

for (int i = 3; i < 1003; i++){
S1: A[i] = E[i-2] + D[i-3];
S2: B = B * B + A[i-1];
S3: C[i] = A[i] * B;
S4: D[i] = C[i+2] / 2;
S5: E[i] = C[i-1] * 5;
}

```

Suponga los siguientes parámetros

Suma: 2 ciclos
 Multiplicación: 4 ciclos
 División: 5 ciclos
 Asignación: 1 ciclo

5.- Realice el grafo de dependencias (RAW) del siguiente fragmento de código y averigüe el tiempo que tardará en emitirse el en un procesador con emisión en orden de una instrucción por ciclo y dos unidades de ejecución que pueden ejecutar cualquier tipo de instrucción?

S₁: ADDD r3, r0, r5
 S₂: MULD r2, r2, r5
 S₃: DIVD r1, r5, r0
 S₄: STD (r3), r1
 S₅: SUBD r6, r3, r2
 S₆: LDD r9, (r6)
 S₇: ADDD r2, r6, r3
 S₈: DIVD r10, r3, r1

Suponga los siguientes parámetros

ADDD: 2 ciclos
 MULD: 4 ciclos
 SUBD: 3 ciclos
 DIVD: 7 ciclos
 LDD/STD: 5 ciclos

6.- Elaboración. Intente armar (sin consultar el material bibliográfico) un esquema de bloques de la estructura de un procesador superescalar mostrando todas las etapas de que consiste.

7.- Elaboración. Analice en forma individual o grupal la estructura y la función de cada una de las etapas de un procesador superescalar genérico

8.- a) ¿Cuánto tiempo tardará en emitirse el siguiente fragmento de código en un procesador superescalar de grado 2 con ejecución en orden y 2 unidades de ejecución, en el que todas las instrucciones tienen una latencia de 1 ciclo y cualquier unidad de ejecución puede ejecutar cualquier tipo de instrucción?

b) ¿Cuál es la aceleración obtenida si el mismo fragmento se ejecutara en un procesador escalar (de grado 1)?

S₁: LDD r1, (r2)
 S₂: SUBD r4, r5, r6
 S₃: ADDD r3, r1, r7
 S₄: MULD r8, r3, r3
 S₅: STD (r11), r4
 S₆: STD (r12), r8
 S₇: ADDD r15, r14, r13
 S₈: SUBD r10, r15, r10
 S₉: STD (r9), r10

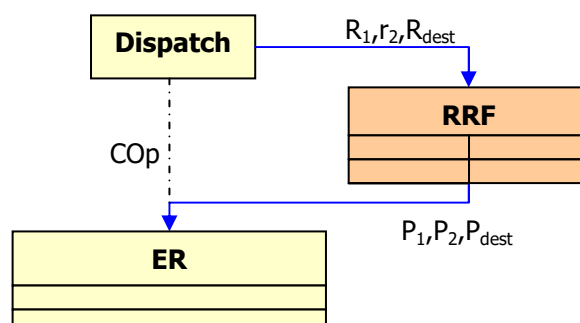
9.- (ejercicio de elaboración)

Analice y discuta los conceptos de emisión en orden y en desorden, emisión con ventana fija y deslizante, ejecución escalar y superescalar y unidades de cálculo genéricas y especializadas.

Proponga un fragmento de código en bajo nivel y pruebe su ejecución en distintas combinaciones de arquitecturas evaluando su tiempo de ejecución en ciclos de reloj.

10.- Emita la secuencia de instrucciones del ejercicio 5 a la estación de reserva (ER) renombrando los registros de arquitectura por registros físicos a partir de la tabla de renombre de registros (RRF) siguiente:

P	valid	Reg des	value	busy	last
1	1	0	100	1	1
2	0	2	1000	1	1
3	1	2	200	1	1
4	1	5	70	1	1
5					
6					
7					
8					
9					
10					
11					



14.- El esquema de la figura representa de manera simplificada la arquitectura de un procesador que planifica la ejecución de instrucciones mediante el algoritmo de *Tomasulo Original*.

RS	Tag	Src1	Tag	Src2	Dst
1					
2					
3					

Adder

RS	Tag	Src1	Tag	Src2	Dst
4					
5					

Mult/Div

FPRF	Busy	Tag	Data
0	0		6.0
2	0		3.5
4	0		10.0
8	0		7.8

S_1 : MULD r4, r0, r8 (3 ciclos)
 S_2 : ADDD r2, r0, r4 (2 ciclos)
 S_3 : ADDD r4, r0, r8 (2 ciclos)
 S_4 : MULD r8, r4, r8 (5 ciclos)
 S_5 : MULD r8, r4, r8 (5 ciclos)
 S_6 : ADDD r8, r4, r8 (5 ciclos) Busy r4 en 0

Se ejecuta la secuencia de instrucciones mostrada en la figura con las latencias de cada UF indicada entre paréntesis. Complete para cada ciclo el estado de las estaciones de reserva (RS) y del banco de registros de punto flotante (FPRF) en los siguientes supuestos:

- Emisión de una instrucción por ciclo
- Emisión múltiple de 2 instrucciones en caso de que se ejecuten en UF diferentes (Adder o Mult/Div).
- Analice el resultado de la ejecución anterior y explique que inconsistencias se presentan.
- ¿Cómo solucionaría dicha inconsistencia? (Tenga en cuenta el tomasulo básico vs optimizado)