

Práctico N° 7: Algoritmo de Tomasulo

Ejercicio 1:

Considerando un microprocesador out-of-order execution implementado mediante el algoritmo de Tomasulo, el código en assembler el hardware especificado que se muestra a continuación, muestre el contenido de las tablas de *Status*, las *Reservation stations* y el flujo de ejecución para el 6^{to} clock de ejecución (inclusive).

Código:

1>	LDURD D6, [X2, #32]
2>	LDURD D2, [X3, #44]
3>	FMULD D0, D2, D4
4>	FSUBD D8, D2, D6
5>	FDIVD D0, D0, D6
6>	FADDD D6, D8, D2

Especificaciones de hardware:

Hardware
Issue = 1 instrucción
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Suma punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 6 clk

Ejercicio 2:

Considerando un microprocesador out-of-order execution implementado mediante el algoritmo de Tomasulo, el código en assembler el hardware especificado que se muestra a continuación, muestre el contenido de las tablas de *Status*, las *Reservation stations* y el flujo de ejecución para el 8^{vo} clock de ejecución (inclusive).

Código:

L:	1> LDURD D0, [X1, #0]
	2> FMULD D4, D0, D2
	3> STURD D4, [X1, #0]
	4> SUBI X1, X1, #8
	5> CBNZ X1, L

Arquitectura de Computadoras 2023

Especificaciones de hardware:

Hardware
<i>El salto se predice correctamente como taken.</i>
Issue = 2 instrucciones
Load = 6 RS / 1 clk
Store = 6 RS / 1 clk
Alu entera = 3 RS / 1 clk
Alu punto flotante = 3 RS / 2 clk
Multiplicación punto flotante = 2 RS / 3 clk
Branch = 1 RS, 1 FU / 1clk

Ejercicio 3:

Considerando un microprocesador out-of-order execution implementado mediante el algoritmo de Tomasulo, el código en assembler el hardware especificado que se muestra a continuación, muestre el contenido de las tablas de *Status*, las *Reservation stations* y el flujo de ejecución para el 8^{vo} clock de ejecución (inclusive).

Código:

1> ADDI X2, X0, #50
2> ADDI X3, X0, #70
3> ADDI X4, X0, #40
4> LDURD D0, [X4, #0]
5> LDURD D1, [X2, #0]
6> ADDI X5, X2, #16
7> FADDD D2, D1, D0
8> LDURD D1, [X2, #8]
9> STURD D2, [X3, #0]
10> ADDI X2, X2, #2
11> FADDD D2, D1, D0
12> LDURD D1, [X2, #0]
13> ADDI X2, X2, #1
14> ADDI X3, X3, #1

Especificaciones de hardware:

Hardware
Issue = 4 instrucciones
Load = 4 RS / 2 clock cycle
Store = 4 RS / 2 clock cycle
Suma entera = 2 FU - 4 RS / 1 clock cycles
Multipliación entera = 2 FU - 4 RS / 1 clock cycles
Suma flotante = 2 FU - 4 RS / 2 clock cycles
Multipliación flotante = 2 FU - 4 RS / 4 clock cycles

Ejercicio 4:

Considerando un microprocesador out-of-order implementado mediante el algoritmo de Tomasulo (sin especulación), muestre el contenido de las tablas de Status, Registros, las Reservation stations y el flujo de ejecución para la siguiente secuencia de código para el 7^{mo} ciclo de clk (incluido):

Código:

```

1>> addi  x3,xzr,#80
L: 2>> ldur  D2,[x1]
3>> ldur  D6,[x1,#100]
4>> fmul  D4,D2,D0
5>> fadd  D6,D4,D6
6>> stur  D6,[x1,#100]
7>> subi  x1,x1,#8
8>> cbnz  x1,L
9>> add   x1,xzr,#160

```

Asumir que X1 inicialmente contiene el valor 80. Asumir que el salto fue predicho correctamente como taken. Considerar el siguiente Hardware:

Hardware
Issue = 4 instrucciones
Load = 4 RS / 1 clock cycle
Store = 4 RS / 1 clock cycle
Suma entera = 4 RS, 2 FU / 1 clk
Suma punto flotante = 4 RS, 2 FU / 2 clk
Multipliación punto flotante = 4 RS, 2 FU / 3 clk
Branch = 1 RS, 1 FU / 1clk

Arquitectura de Computadoras 2023

Ejercicio 5:

Considerando un microprocesador out-of-order implementado mediante el algoritmo de Tomasulo (sin especulación), muestre el contenido de las tablas de Status, Registros, las Reservation stations y el flujo de ejecución para la siguiente secuencia de código para el 5^{to} ciclo de clk (incluido):

Código:

```
1>> addi x1,xzr,#80
2>> ldurd D0, [x1]
3>> ldurd D1, [x1, #100]
4>> fmuld D4, D2, D0
5>> fadddd D6, D4, D1
6>> fadddd D3, D5, D7
7>> subi x1,x1,#8
8>> sturd D6, [x1, #100]
9>> andi x4, x2, #0xFF
10>> cbnz x1, L
11>> addi x1, x1, #16
L: 12>> mult x2, x3, x5
```

Especificaciones de hardware:

Hardware
Los saltos son predichos como taken
Issue = 4 instrucciones
Load = 4 RS / 2 clock cycle
Store = 4 RS / 2 clock cycle
Suma entera = 4 RS, 2 FU / 1 clk
Suma punto flotante = 4 RS, 2 FU / 2 clk
Multiplicación punto flotante = 4 RS, 2 FU / 3 clk
Branch = 1 RS, 1 FU / 1clk