

Primer Parcial de Arquitectura de Computadoras (Tema 1)

Apellido: _____ Nombre: _____ Hojas: ____

Práctica

1. Describa a nivel transferencia entre registros la búsqueda, decodificación y ejecución de la instrucción "**saltar y enlazar**" (jal) con modo de direccionamiento relativo en un procesador genérico. (1 punto)

BÚSQUEDA

MAR \leftarrow PC
 READ (MDR \leftarrow MP[MAR])
 IR \leftarrow MDR
 PC \leftarrow PC + 1

DECODIFICACIÓN

μ MAR \leftarrow OPC

EJECUCIÓN

ra \leftarrow PC
 PC \leftarrow PC + desp

2. Haz el diagrama de tiempos de la siguiente secuencia de instrucciones, considerando una segmentación de 5 etapas, tal como se encuentra escrita sin adelantamiento de datos. Encuentra los riesgos y resuélvelos usando reordenamiento y adelantamiento de datos y expresa la mejora relativa como un porcentaje. (2 puntos)

Lw t0,8(t1);
 Lw t2,4(t1);
 Addi t3,t2,0xF5;
 Add t4,t3,t0

Diagrama de tiempos sin adelantamiento de datos

Lw t0, 8(t1)	IF	ID	EX	MEM	WB								
Lw t2, 4(t1)		IF	ID	EX	MEM	WB							
Addi, t3, t2, 0xF5			IF	ID			EX	MEM	WB				
Add t4, t3, t0				IF			ID			EX	MEM	WB	

Riesgos:

t2 es usado por Addi antes de que esté calculado

t3 es usado por Add antes de que esté calculado

Reordenamiento de instrucciones y adelantamiento de datos.

Lw t2, 4(t1)	IF	ID	EX	MEM	WB								
Lw t0, 8(t1)		IF	ID	EX	MEM	WB							
Addi, t3, t2, 0xF5			IF	ID	EX	MEM	WB						
Add t4, t3, t0				IF	ID	EX	MEM	WB					

Mejora:

$$12/8 = 1.5$$

3. Realizar el siguiente producto usando el algoritmo de suma y desplazamiento en una unidad aritmética con registros de 4 bits. Los operandos están expresados en magnitud y signo. Verificar en decimal. $C = X * Y$
 $X = 1\ 0100$ $Y = 0\ 1101$. (2 puntos)

- El 10% son instrucciones de sw.
- El 25% son instrucciones beq, en las que la condición evaluada se cumple en el 70% de los casos.
- El 5% se corresponden con instrucciones jal.
- El 35% restante son instrucciones aritmético-lógicas.

Suponiendo que el procesador funciona a una frecuencia de 1.5 GHz.

- Calcule el CPI obtenido en la ejecución del programa.
- Calcule el tiempo de ejecución de este.

$$a. \text{CPI} = 0,25 \times (0,5 \times 1 + 0,5 \times 2) + 0,10 \times 1 + 0,25 \times (0,70 \times 3 + 0,3 \times 1) + 0,35 \times 1 = 1,425$$

$$b. \text{Tiempo ej.} = \frac{\text{Cant. instr.} \times \text{CPI} \times 10 \text{ ns}}{1.5 \times 10^9} = \frac{500 \times 1,425 \times 10 \text{ ns}}{1.5 \times 10^9}$$

- En el siguiente programa hay errores, encuentrelos, corrija el programa para que funcione correctamente y explique qué hace el programa y los errores presentes. (1 punto)

```
.data
array: .byte 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00, 0xFF
res: .space 10
.text
la a1, array
la a2, res
li a3, 11
loop:
lw a4, 0(a1)
andi a5, a4, 0x04
beqz a5, skip
addi a4, a4, 1
skip:
sb a4, 0(a2)
addi a1, a1, 1
addi a2, a2, 1
bnez a3, loop
addi a3, a3, -1 debería estar antes para decrementar el contador

li a5, 10
ecall
```

El programa recorre un arreglo de 10 elementos de tamaño byte. A los elementos que tienen un 1 en el tercer bit les resta un uno y los almacena en un arreglo Res.

Primer Parcial de Arquitectura de Computadoras (Tema 2)

Apellido: _____ Nombre: _____ Hojas: ____

Práctica

- Describa a nivel transferencia entre registros la búsqueda, decodificación y ejecución de la instrucción "**saltar si igual a cero**" (beqz) con **modo de direccionamiento relativo** en un procesador genérico. (1 punto)

BÚSQUEDA

MAR ← PC

READ (MDR ← MP[MAR])

IR ← MDR

PC ← PC + 1

DECODIFICACIÓN

μMAR ← OPC

EJECUCIÓN

si (Z) entonces PC ← PC + desp

2. Haz el diagrama de tiempos de la siguiente secuencia de instrucciones, considerando una segmentación de 5 etapas, tal como se encuentra escrita sin adelantamiento de datos. Encuentra los riesgos y resuélvelos usando reordenamiento y adelantamiento de datos y expresa la mejora relativa como un porcentaje. (2 puntos)

Lw t0,8(t1);
 Lw t2,4(t1);
 Add t3,t0, t2
 Add t4,t3,t0
 Sw t5, 0(t1)

Diagrama de tiempos sin adelantamiento de datos

Lw t0, 8(t1)	IF	ID	EX	MEM	WB														
Lw t2, 4(t1)		IF	ID	EX	MEM	WB													
Add, t3, t0, t2			IF	ID				EX	MEM	WB									
Add t4, t3, t0				IF				ID				EX	MEM	WB					
Sw t5, 0(t1)								IF				ID	EX	MEM	WB				

Reordenamiento de instrucciones y adelantamiento de datos.

Lw t0, 8(t1)	IF	ID	EX	MEM	WB														
Lw t2, 4(t1)		IF	ID	EX	MEM	WB													
Sw t5, 0(t1)			IF	ID	EX	MEM	WB												
Add, t3, t0, t2				IF	ID	EX	MEM	WB											
Add t4, t3, t0					IF	ID	EX	MEM	WB	IF									

Mejora:

$$13/10 = 1.3$$

3. Realizar el siguiente producto usando el algoritmo de suma y desplazamiento en una unidad aritmética con registros de 4 bits. Los operandos están expresados en magnitud y signo. Verificar en decimal. $C = X * Y$
 $X = 0\ 0101$ $Y = 0\ 1011$. (2 puntos)

ACCIÓN	REG. ACUMUL.	REG Q.	REG D.	CONTADOR
carga	0000	0101	1011	$100_2 = 4_{10}$
sumo Reg. Acum + Reg. D	$\begin{array}{r} +1011 \\ 0000 \\ \hline 1011 \end{array}$	$\begin{array}{r} 0101 \\ 0101 \end{array}$		
→	0101	1010		$11_2 = 3_{10}$
→	0010	1101		$10_2 = 2_{10}$
sumo Reg. Acum + Reg. D	$\begin{array}{r} +1011 \\ 0010 \\ \hline 1101 \end{array}$	1101		
→	0110	1110		$01_2 = 1_{10}$
→	0011	0111		0
Resultado	00110111			

El signo se resuelve mediante la operación XOR

$0 \text{ XOR } 0 \rightarrow 0$

El resultado es un número positivo $0 \ 0110111 = 55$

4. Realice la siguiente operación en IEEE 754. Convierta los valores decimales a IEEE 754, realice la operación, convierta nuevamente a decimal y calcule el error relativo cometido. $115.8 - 49.7$. (2 puntos)

$115.8 = 1110011.11001100110011001$

Signo: +

Mantisa normalizada: $1.11001111001100110011001 \times 2^6$

Exp= $6 + 127 = 133$

0	1	0	0	0	0	1	0	1	1	1	0	0	1	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

$49.7 = 110001.101100110011001100$

Signo: +

Mantisa normalizada: $1.10001101100110011001100 \times 2^5$

Exp: $5 + 127 = 132$

0	1	0	0	0	0	1	0	0	1	0	0	0	1	1	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Igualemos el exponente a la mayor potencia:

$1.10001101100110011001100 \times 2^5 = 0.11000110110011001100110 \times 2^6$

$1.11001111001100110011001 \times 2^6$

$0.11000110110011001100110 \times 2^6$

$1.00001000011001100110011 \times 2^6$

No es necesario volver a normalizar

calculo el número obtenido:

$(1 + (1 \times 2^{-5} + 1 \times 2^{-10} + 1 \times 2^{-11} + 1 \times 2^{-14} + 1 \times 2^{-15} + 1 \times 2^{-18} + 1 \times 2^{-19} + 1 \times 2^{-22} + 1 \times 2^{-23})) \times 2^6 = (1 + 0.03281247615) \times 2^6 = 66.0999984741$

Valor real: $115.8 - 49.7 = 66.1$

Valor obtenido=66.0999984741

Error absoluto= $|66.0999984741 - 66.1| = 0.0000015259$

Error relativo: $0.9999997991 / 66.1 = 2.30847201 \times 10^{-8}$

5. Supongamos que sobre el procesador RISC, se ejecuta un programa de 700 instrucciones distribuidas de la siguiente manera: (2 puntos)

El 20% corresponden a instrucciones de lw, y de éstas, en la mitad de los casos vienen seguidas de una instrucción aritmética que lee el registro sobre el que escribe la instrucción lw.

El 15% son instrucciones de sw.

El 30% son instrucciones beq, en las que la condición evaluada se cumple en el 70% de los casos.

El 5% se corresponden con instrucciones jal.

El 25% restante son instrucciones aritmético-lógicas.

Suponiendo que el procesador funciona a una frecuencia de 1.5 GHz.

a. Calcule el CPI obtenido en la ejecución del programa.

b. Calcule el tiempo de ejecución de este.

a. $CPI = 0.2 \times (0.5 \times 1 + 0.5 \times 2) + 0.15 \times 1 + 0.30 \times (0.70 \times 3 + 0.3 \times 1) + 0.05 \times 3 + 0.25 \times 1 = 1.57$

b. $\text{Tiempo ejec} = \frac{\text{Cant. instr.} \times CPI}{\text{Frecuencia}} \times 10^9 \text{ ns} = \frac{700 \times 1.57}{1.5 \times 10^9} \times 10^9 \text{ ns} = 732.66 \text{ ns}$

6. En el siguiente programa hay errores, encuentrellos, corrija el programa para que funcione correctamente y explique qué hace el programa y los errores presentes. (1 punto)

```
.data
array: .byte 0x01, 0x02, 0x04, 0x08, 0x10, 0x20, 0x40, 0x80, 0x00, 0xFF
res: .space 10

.text
la a1, array
la a2, res
li a3, 9
loop:
lb a4, 0(a1)
andi a5, a4, 0x01
beqz a5, skip
addi a4, a4, -3
skip:
sw a4, 0(a2)
addi a1, a1, 1
addi a2, a2, 1
addi a3, a3, 1 nunca decrementa
bnez a3, loop

li a7, 8
ecall
```

El programa recorre un arreglo de 10 elementos de tamaño byte. A los elementos que tienen un 1 en el primer bit les suma un uno y los almacena en un arreglo Res.