**Organización de Computadores PEP #1, 01/2024**

1. [15% ~ 5% y 10%] Considere que las variables f, g, h e i están almacenadas en los registros $s0, $s1, $s2 y $s3 y que la base de los arreglos A y B están en los registros $s6 y $s7 respectivamente.

Utilizando el mínimo número de instrucciones transforme a lenguaje ensamblador:

# NOTAR, f=$s0, g=$s1, h=$s2, i=$s3, A=$s6, B=$s7

a) f = g + h + B[i]

Respuesta:

sll $t0, $s3, 2 #mult i por 4

add $t0, $t0, $s7 #suma la dirección base

lw $t1, 0($t0) #guardamos en el registro el valor de B[i]

add $t1, $s2, $t1 # sumamos h + B[i]

add $s0, $t1, $s1 # sumamos g + h + B[i]

b) f = g - A[B[i]]

Respuesta:

sll $t0, $s3, 2 #mult i por 4

add $t0, $t0, $s7 #suma la dirección base de B

lw $t1, 0($t0) #guardamos en el registro $t1 el valor de B[i]

sll $t1, $t1, 2 # multiplicamos el valor de B[i] por 4

add $t1, $t1, $s6 # sumamos la dirección base de A al valor de $t1

lw $t2, 0($t1) # guardamos en $t2 el valor de A[B[i]]

sub $s0, $s1, $t2 # en $s0 calculamos g - A[B[i]]

1. [15% ~ 5% c/u] Utilizando el mínimo número de instrucciones construya las siguientes pseudo instrucciones:

a) move $t1, $t2 // $t1 <- $t2

Respuesta:

add $t1, $0, $t2

b) lwx $t5, big($t2) // $t5 <- Memory[$t2 + big]

Respuesta:

lui $t0, upper(big) # guardo con lui la primera parte de big en los bits #mas significativos

ori $t0, $t0 lower(big) # guardo los bits menos signi de big

add $t0, $t0, $t2 # sumo la dirección base

lw $t5, 0($t0) #guardo lo que estaba en la memoria en direccion

c) ble $t3, $t5, L // if ($t3 <= $t5) go to L

Respuesta:

slt $t0, $t5, $t3 #guardo en $t0 el signo de $t5 - $t3,

#dado que busco que $t5 - $t3 >= 0, si este valor es

# es igual o mayor a 0, en $t0 se guarda un 0

beq $t0, $0, L # si $t0 es 0 voy al label L

Hint: big es un número que requiere 32 bits mientras que small es un número que requiere 16 bits.

3. [50%] Considere el siguiente extracto de código:

1. 0xFC00000C X: addi $sp, $sp, -8

2. 0xFC000010 sw $s0, 4($sp)

3. 0xFC000014 sw $ra, 0($sp)

4. 0xFC000018 bne $a0, $0, else

5. 0xFC00001C add $v0, $0, $0

6. 0xFC000020 addi $sp, $sp, 8

7. 0xFC000024 0x03E00008

8. 0xFC000028 else: andi $s0, $a0, 1

9. 0xFC00002C srl $a0, $a0, 1

10. 0xFC000030 jal X

11. 0xFC000034 add $v0, $v0, $s0

12. 0xFC000038 0x8FBF0000

13. 0xFC00003C lw $s0, 4($sp)

14. 0xFC000040 addi $sp, $sp, 8

15. 0xFC000044 jr $ra

a) [5% c/u] Transforme a lenguaje máquina las líneas 3, 8, 9 y 13 (Hexadecimal).

Respuesta:

3- 0xFC000014 sw $ra, 0($sp)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 43 | | | | | | 29 | | | | | 31 | | | | | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | | | | F | | | | B | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| sw $ra, 0($sp) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

8- 0xFC000028 else: andi $s0, $a0, 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 12 | | | | | | 4 | | | | | 16 | | | | | 1 | | | | | | | | | | | | | | | |
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | | | | 0 | | | | 9 | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 1 | | | |
| andi $s0, $a0, 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

9- 0xFC00002C srl $a0, $a0, 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | rd | | | | | shamp | | | | | funct | | | | | |
| 0 | | | | | | 0 | | | | | 4 | | | | | 4 | | | | | 1 | | | | | 2 | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 0 | | | | 0 | | | | 0 | | | | 4 | | | | 2 | | | | 0 | | | | 4 | | | | 2 | | | |
| srl $a0, $a0, 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

13- 0xFC00003C lw $s0, 4($sp)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 35 | | | | | | 29 | | | | | 16 | | | | | 4 | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 8 | | | | F | | | | B | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 4 | | | |
| lw $s0, 4($sp) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

b) [10%] Para la línea 4 determine el BTA y transforme a lenguaje máquina.

Respuesta:

4. 0xFC000018 bne $a0, $0, else

8. 0xFC000028 else: andi $s0, $a0, 1

Notamos que la etiqueta else está en la instrucción 8. Luego, entre el PC+4 y la instrucción 8 hay 3 instrucciones,

BTA = PC+4 + 3\*4 = 0xFC00001C + 0xC = 0xFC000028

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 5 | | | | | | 4 | | | | | 0 | | | | | 3 | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | | | | 4 | | | | 8 | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 3 | | | |
| bne $a0, $0, else | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

c) [10%] Para la línea 10 determine el JTA y transforme a lenguaje máquina.

Respuesta:

10. 0xFC000030 jal X

1. 0xFC00000C X: addi $sp, $sp, -8

Notamos que la instruccion X está en la dirección 0xFC000030, luego debemos quitar los 4 bits mas significativos y los 2 menos significativos. Luego:

JTA = 0x3000003 = 11000000000000000000000011

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| F | | | | C | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | C | | | |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
|  |  |  |  | 3 | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 3 | | | |  |  |

Entonces, la instrucción es:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | adress | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | | 0x3000003 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 0 | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 3 | | | |
| jal X | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

d) [5% c/u] Transforme a lenguaje ensamblador las líneas 7 y 12.

7. 0xFC000024 0x03E00008

Notar que es Tipo R, además jr es funct= 1000, y el registro N 15 es $t7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | rd | | | | | shamp | | | | | funct | | | | | |
| 0 | | | | | | F | | | | | 0 | | | | | 0 | | | | | 0 | | | | | 8 | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 0 | | | | 3 | | | | E | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | | 8 | | | |
| jr $t7 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

12. 0xFC000038 0x8FBF0000

Tiene pinta de tipo I, luego lw tiene op=0x23=35, además $sp=N29 y $ra=N31

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 35 | | | | | | 29 | | | | | 31 | | | | | 0 | | | | | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 8 | | | | F | | | | B | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | 0 | | | |
| lw $ra, 0($sp) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

4. [20%] Considere dos implementaciones distintas del mismo conjunto de instrucciones (ISA). Las instrucciones se dividen en cuatro clases según su CPI. Las clases corresponden a U, V, W y X. La primera implementación (P1) tiene una tasa de reloj de 3 Ghz y CPI’s de 1, 2, 3 y 3 mientras que la segunda implementación tiene una tasa de reloj de 3,5 Ghz con CPI’s de 2, 3, 2 y 3 respectivamente. Dado un programa con un total de 1 millón de instrucciones divididas por clases según lo siguiente: 10% tipo U, 30% tipo V, 40% tipo W y 20% tipo X, responda:

a. ¿Qué implementación es más rápida? (¿P1 o P2?)

Deberíamos comparar los Tiempos de ejecución, y considerando el curso, generalmente tomamos el tiempo de la CPU como el tiempo de ejucicón. Luego

Tcpu = Sumatoria(CPI \* #instruccionesTipo)/Tasa

Total instrucciones = 1 000 000

Entonces:

Tcpu1= (1\*0,1+2\*0,3+3\*0,4+3\*0,2)\*10^6 / (3\*10^9) [hz] = 0,8333 \*10^-3 s

Tcpu2= (2\*0,1+3\*0,3+2\*0,4+3\*0,2)\*10^6 / (3,5\*10^9) [hz] = 0,7142\*10^-3 s

La implementación más rápida es P2

b. Determinar el CPI para ambas implementaciones.

Notamos: CPI=(Tcpu\*tasa)/#instrucciones

Luego:

CPI1= (0,8333 \*10^-3 \* 3\*10^9)/( 10^6 ) = 2,499999

CPI2= (0,7142\*10^-3 \* 3,5\*10^9)/( 10^6 ) = 2,4997

c. Determine la cantidad de ciclos de reloj que requieren las implementaciones para ejecutar el programa.

Recordar: CiclosRelog= Tcpu\*tasa

Luego:

Ciclos1= 0,7142\*10^-3 [s] \* 3,5\*10^9 [1/s] = 2, 4997\*10^6

Ciclos2= 0,8333 \*10^-3 [s] \* 3 \*10^9 [1/s] = 2,499999 \*10^6

Tipo R

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | rd | | | | | shamp | | | | | funct | | | | | |
|  | | | | | |  | | | | |  | | | | |  | | | | |  | | | | |  | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tipo I

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
|  | | | | | |  | | | | |  | | | | |  | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Tipo J

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | adress | | | | | | | | | | | | | | | | | | | | | | | | | |
|  | | | | | |  | | | | | | | | | | | | | | | | | | | | | | | | | |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |  | | | |
|  | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |