**Organización de Computadores PEP #1, 02/2023 Diurno**

1. Historia, Hardware y Rendimiento (30%)

Se ha instalado Arch Linux en un computador que cuenta con una tasa de reloj de 2Ghz. La ejecución de un programa X (400 mil millones de instrucciones, o bien 4\*10^11) demora 13 minutos y 20 segundos (800 s).

(a) Determine el CPI del programa X.

Respuesta:

Notar: CPI= (Tcpu\*Tasa)/#instrucciones

Luego:

CPI = (800 [s] \* 2\*10^9 [1/s]) / 4\*10^11 = 400 \*10^-2 = 4

(b) Calcule el tiempo de CPU para un programa Y que tiene un 20% más de instrucciones que X y el CPI corresponde al 75% del calculado en el punto (a).

Respuesta:

Notar: 20% mas de Instrucciones = 1,2\*4\*10^11 = 4,8\*10^11,

y 75% del CPI = 0,75\*4 = 3

Luego, considerando la tasa constante:

Tcpu = (#instruc \*CPI) /Tasa = (4,8\*10^11 \* 3) /2\*10^9 [1/s]) = 7,2 \* 10^2 s

(c) Si otro computador obtiene un CPI de 3,5 para el programa X, ¿cuál es más rápido y por cuánto?

Respuesta:

Considerando otro computador con CPI 3,5 para el programa X y con la tasa cte.

Tcpu=(#insutrucc\*CPI)/tasa = (4\*10^11 \* 3,5 )/ 2\*10^9 [1/s] = 700 [s] = 11 min 40s

Comparamos tiempos, 800s/700s = 1,14 veces más rápido

Luego, este nuevo computador es más rápido

(d) En el contexto de esta pregunta: ¿cómo podemos mejorar el rendimiento del sistema al ejecutar X?

Considerando los ciclos constantes, podemos aumentar la frecuencia, y esto se hace reduciendo el periodo, es decir, disminuyendo el tiempo de un ciclo de relog.

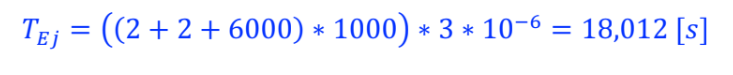
Otra forma sería aumentando los ciclos de relog, algunos computadores implementan esta funcionalidad, ejemplo TurboBoost

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Determine el tiempo de uso de procesador: COPIA DE LA PAUTA

Respuesta: El ciclo interno requiere 2 accesos a memoria, 2 operaciones aritméticas y 2 direccionamientos) para mantenerse dentro del ciclo y salir). El ciclo externo requiere 2 operaciones aritméticas (una de ellas para resetear j) y 2 de tipo direccionamiento.



Texto

Descripción generada automáticamente

Respuesta:

#suponiendo que $s0=i, $s1=s, y $s3=uwu (dirección base)

add $s0, $0, $0 #inicializamos i

add $s1, $0, $0 #inicializamos s

for: slti $t0, $s0, 200 #en t0 se guarda el signo de i-200

beq $t0, $0, salida #condicion salida, reutilizamos el $t0 para lw

sll $t0, $t0, 2 #mult por 4 la copia del i

add $t0, $t0, $s3 # a la copia anterior sumamos la d.base, es d.elem

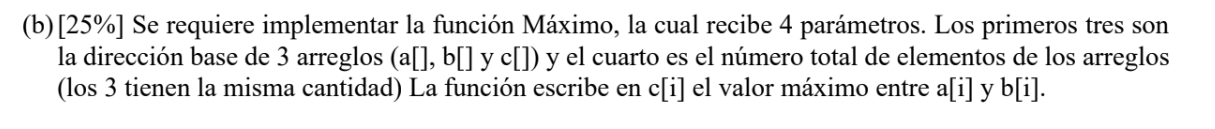
lw $t1, 0($t0) # accedemos a uwu[i] y lo guardamos en $t1

add $s1, $s1, $t1 # s = s + uwu[i]

addi $s0, $s0, 1 # i++

j for #salto al inicio del loop

salida:



Es sacable pero es lenta, IDEA: comparo con el slt los valores de cada elemento de cada array respectivamente (después de calcular el i\*4+DB), luego veo el mayor y lo escribo en el array C con sw, luego vuelvo al loop, y así hasta llegar al final (4to parametro)

Supone que en $a0 = DB a, $a1= DB b, $a2= DB c, y $a3 = tamaño

Máximo: #inicializo parámetros

add $s0, $0, $a0 #el profe los copio en los registros, aunque no es

add $s1, $0, $a1 #obligatorio

add $s2, $0, $a2

add $s3, $0, $a3

add $t0, $0, $0 #definimos el $t0 como el índicede las iteraciones

While:

slt $t1, $t0, $s3 #considera i-len(array)

beq $t1, $0, salida #salida cuando i-len(array) sea 0

sll $t1, $t1, 2 #mult por 4 la copia del i en el mismo $t1

add $t2, $t1, $s0 # Direc elemento A[i]

add $t3, $t1, $s0 # Direc elemento B[i]

lw $t2, 0($t2) # en $t2 guardo lo que tenía A[i]

lw $t3, 0($t3) # en $t3 guardo lo que tenía B[i]

slt $t4, $t2, $t3 # resto el A[i] – B[i] y guardo el signo

#si 1(neg), el B[i] es mayor, sino son = o A[i] es mayor

Beq $t4, $0, aEsMayor

add $t4, $t1, $s2 #sumo la DB de c al i\*4

sw $t3, 0($t4) #escribo el C[i] el valor de B[i] que es mayor

j volverCiclo #salto al final para volver a iterar sin entrar en al #caso de A[i] es mayor

aEsMayor:

add $t4, $t1, $s2 #sumo la DB de c al i\*4 .SE REPITE, NO OPTIMO

sw $t2, 0($t4) #escribo el C[i] el valor de A[i] que es mayor

j volverCiclo

volverCiclo:

addi $t0, $t0, 1 # i++

j while

salida: jr $ra #volvemos a donde se llamo la función

Mi dif con el código del profe es que el después de direccionar que numero es mayor y escribirle, salta al inicio del bucle, porque ya hizo el i++.

Texto, Carta

Descripción generada automáticamente Texto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

1. Convierta a lenguaje máquina las líneas 2, 5 y 10

2. uwu: sw $ra, 0($sp), notamos tipo I

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
|  | | | | | | 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 | | | |
| 0xAFBF0000 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

5. beq $t0, $0, uwu

Notamos que entre PC+4 y uwu hay -4 instrucciones.

Entonces C2 de -4 ----> 0100 ---- 1011----1100

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 4 | | | | | | 8 | | | | | 0 | | | | | -4 | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 1 | | | | 1 | | | | 0 | | | | 0 | | | | F | | | | F | | | | F | | | | C | | | |
| 0x1100FFFC | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

10. j uwu

Notamos que la dirección del uwu es 0xAFBF0004. Luego calculamos el addres

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| A | | | | F | | | | B | | | | F | | | | 0 | | | | 0 | | | | 0 | | | | C | | | |
| 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 | 1 | 0 | 0 |
|  |  |  |  | 3 | | E | | | | F | | | | C | | | | 0 | | | | 0 | | | | 1 | | | |  |  |

Entonces, JTA = 0x3EFC001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | adress | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2 | | | | | | 0x3EFC001 | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0 | 0 | 0 | 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 | 1 |
| 0 | | | | B | | | | E | | | | F | | | | C | | | | 0 | | | | 0 | | | | 1 | | | |
| 0x0BEFC001 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

1. Convierte a lenguaje ensamblador las líneas 1, 6 y 9

1. 0xAF848008. No es tipo R. y como opcode de sw es 43, es de tipo I

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | imm | | | | | | | | | | | | | | | |
| 43 | | | | | | 28 | | | | | 4 | | | | | 32776 o 0x8008 | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| A | | | | F | | | | 8 | | | | 4 | | | | 8 | | | | 0 | | | | 0 | | | | 8 | | | |
| sw $a0, 0x8008($gp) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

2. 0x8FBF0000. No es tipo R. opcode 35 o 0x23 es de lw

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

3. 0x03E00008. Es de tipo R, según el funct es un jr al reg 31 que es $ra

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| opcode | | | | | | rs | | | | | rt | | | | | rd | | | | | shamp | | | | | funct | | | | | |
| 0 | | | | | | 31 | | | | | 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 $ra | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |