

Complemento de Arquitectura de Computadoras

Solución Examen 24 de Julio de 2015

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique el total de hojas en la primera. Escriba las hojas de un solo lado.
- Solo se responderán dudas de letra. No se responderán preguntas en los últimos 30 minutos de la prueba.
- La prueba es individual y sin material. Duración de la prueba : 3 horas.
- El puntaje mínimo de aprobación es de 60 puntos.

Pregunta 1 (8 puntos)

En una unidad de control microprogramada, ¿cuáles son las diferencias entre microprogramación horizontal y vertical?

Pregunta 2 (12 puntos)

Enumere técnicas de predicción de saltos, y señale ventajas y desventajas de cada una. En particular compare las técnicas de "saltar siempre" vs. "saltar nunca".

Pregunta 3 (10 puntos)

Comente las razones por las cuales un cache de instrucciones puede ser mas simple que uno de datos.

Pregunta 4 (10 puntos)

Describa la técnica de Register Renaming, y de un ejemplo de aplicación.

Ejercicio 1 (25 puntos)

La siguiente tabla muestra el rendimiento de dos procesadores, el rAlpha y el c86, y de dos compiladores en un benchmark común.

	GHz	Compilador A		Compilador B	
		Instrucciones	CPI Promedio	Instrucciones	CPI Promedio
rAlpha	3,4	7000	1,2	5000	1,5
c86	2,6	1500	2,2	1000	4,0

- (a) ¿Cual es la mejor combinación de procesador-compilador?
- (b) Los dos fabricantes lanzan una nueva versión de sus procesadores con cuatro núcleos (quad-core). En estos nuevos procesadores el benchmark tiene el 80% de las instrucciones paralelizables y el rendimiento es el siguiente:

	GHz	Compilador A			Compilador B		
		Instrucciones	CPI paralelizables	CPI no paralelizables	Instrucciones	CPI paralelizables	CPI no paralelizables
rAlphaX	3,4	9000	1,5	1,2	7000	1,8	1,5
c86x4	2,6	2500	3,2	2,4	2000	5,0	4,5

Calcule la aceleración lograda por cada combinación de compilador y procesador.

- (c) Usando la métrica de MIPS y los datos iniciales ¿cual es la mejor combinación? ¿Es esta medida de utilidad? Justifique.

Solución Ejercicio 1

- (a) Para hallar la mejor combinación usaremos la métrica Tiempo de CPU. Calculamos los tiempos de CPU para cada caso usando la fórmula:

$$\text{Tiempo de CPU} = \frac{\text{Recuento de instrucciones} * CPI}{\text{Frecuencia de reloj}}$$

	Compilador A	Compilador B
rAlpha	$\frac{7000 * 1,2}{3,4 * 10^9} = 2,4 * 10^{-6} \text{ seg}$	$\frac{5000 * 1,5}{3,4 * 10^9} = 2,20 * 10^{-6} \text{ seg}$
c86	$\frac{1500 * 2,2}{2,6 * 10^9} = 1,26 * 10^{-6} \text{ seg}$	$\frac{1000 * 4,0}{2,6 * 10^9} = 1,53 * 10^{-6} \text{ seg}$

Por lo tanto la mejor combinación es la del procesador c86 con el Compilador A.

- (b) Usando la misma métrica de la parte anterior calcule el tiempo con la mejora:

	Compilador A	Compilador B
rAlpha	$\frac{(7200/4) * 1,5 + 1800 * 1,2}{3,4 * 10^9} = 1,43 * 10^{-6} \text{ seg}$	$\frac{(5600/4) * 1,8 + 1400 * 1,5}{3,4 * 10^9} = 1,36 * 10^{-6} \text{ seg}$
c86	$\frac{(2000/4) * 3,2 + 500 * 2,4}{2,6 * 10^9} = 1,08 * 10^{-6} \text{ seg}$	$\frac{(1600/4) * 5,0 + 400 * 4,5}{2,6 * 10^9} = 1,46 * 10^{-6} \text{ seg}$

La aceleración se calcula como $\text{Aceleración} = \frac{\text{Tiempo sin mejora}}{\text{Tiempo con mejora}}$

Entonces la aceleración para cada combinación es:

$$\text{Aceleración rAlpha A} = \frac{2,4}{1,43} = 1,68$$

$$\text{Aceleración rAlpha B} = \frac{2,2}{1,36} = 1,62$$

$$\text{Aceleración c86 A} = \frac{1,26}{1,08} = 1,17$$

$$\text{Aceleración c86 B} = \frac{1,53}{1,46} = 1,05$$

- (c) Calculamos los MIPS para cada caso utilizando la fórmula:

$$\text{MIPS} = \frac{\text{Frecuencia de reloj}}{CPI * 10^6}$$

	Compilador A	Compilador B
rAlpha	$\frac{3,4 * 10^9}{1,2 * 10^6} = 2833$	$\frac{3,4 * 10^9}{1,5 * 10^6} = 2267$
c86	$\frac{2,6 * 10^9}{2,2 * 10^6} = 1182$	$\frac{2,6 * 10^9}{4,0 * 10^6} = 650$

Según esta métrica la mejor combinación es la de mayor MIPS, es decir la que ejecuta mas instrucciones por segundo. En este caso es la combinación del procesador rAlpha con el Compilador A.

Como podemos ver MIPS no es una buena métrica ya que nos dice que el mejor procesador es el que tiene un tiempo de ejecución alto. Esto se debe a que MIPS no es una buena métrica para comparar procesadores con repertorios distintos de instrucciones ya que no tiene en cuenta el trabajo efectivo que realizan las instrucciones.

Ejercicio 2 (35 puntos)

Sea el siguiente algoritmo de búsqueda binaria:

```
short binarySearch(short* array, short size, short buscado){
    short inf = 0;
    short sup = size - 1;
    while (inf < sup){
        short center = (sup - inf) / 2 + inf;
        if (array[center] == buscado){
            return 1;
        }
        if (array[center] > buscado){
            sup = center - 1;
        }else{
            inf = center + 1;
        }
    }
    return 0;
}
```

- Compilar a assembler 8086 sabiendo que los parámetros se pasan en ES:BX, AX y CX respectivamente, y que el resultado se debe devolver en DX.
- Suponga que el código se ejecuta en un microprocesador 8086 y que entre este y la memoria principal se ubica un cache de 2 KB de capacidad, 32 líneas y correspondencia directa. Indique cómo se interpreta la línea de memoria para acceder al caché, halle el valor de todos los parámetros relevantes.
- Suponga que se invoca a la función con un arreglo de 16 posiciones tal que $array[i] = i$, y se busca el número 9. Sabiendo que el arreglo se ubica a partir de la dirección 0x200 de memoria, indique las direcciones accedidas por el programa, si provocan hit o miss en el caché, la dirección de los bloques cargados en cache y la tasa de aciertos en el caché.

Solución:

a)

```
binarySearch PROC
    MOV DI, 0          ; inf
    DEC AX             ; AX = sup
while:
    CMP DI, AX
    JGE fin_0
    MOV SI, AX        ; center = sub
    SUB SI, DI        ; center = sub - inf
    SHR SI, 1        ; center = (sub - inf) / 2
    ADD SI, DI        ; center = (sub - inf) / 2 + inf

    SHL SI, 1        ; convierto índice en desplazamiento a memoria
    CMP ES:[BX + SI], CX
    JE fin_1
    JG restar
    ; else
    SHR SI, 1        ; recupero índice
    MOV DI, SI
    INC DI
    JMP while

restar:
    SHR SI, 1        ; recupero índice
    MOV AX, SI
    DEC AX
    JMP while

fin_0:
    MOV DX, 0
    RET

fin_1:
    MOV DX, 1
    RET

ENDP
```

b) Si el cache es de 2 KB y tiene 32 líneas, entonces cada línea ocupa 64 bytes. Por lo tanto el campo 'Byte' tiene 6 bits. Luego, se precisan 5 bits para el campo línea por tener el cache 32 líneas. Por último, el campo 'tag' ocupa 9 bits por ser de 20 bits la dirección de memoria.

c) El primer acceso es siempre en el índice 7 del arreglo. Como en 0x200 está la posición 0, 0x202 la posición 1, etc, el primer acceso es a la dirección 0x20D (0000 0000 0010 0000 1110). La cual es un miss y carga el bloque de dirección 0x200 (0000 0000 0100 0000 0000) en la línea 01000. Los siguientes accesos son a los índices 11 y 9. Los cuales acceden a las direcciones 0x216 y 0x212 respectivamente. Estos accesos son todos hit ya que todos acceden al mismo bloque (0x200). El hit rate del programa es entonces 66%.