

Complemento de Arquitectura de Computadoras

Solución Examen 19 de Febrero de 2018

(ref: ecac20180219.odt)

- 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.
- Justifique todas sus respuestas.

Pregunta 1 (10 puntos)

Describa los buses típicos que aparecen en un sistema basado en microprocesador. Describa los contenidos del bus de control cuando se realiza un acceso de E/S y cuando se realiza un acceso a memoria, suponiendo un espacio de direccionamiento de E/S aislado.

Pregunta 2 (10 puntos)

Explique la técnica de register renaming. ¿Qué tipos de hazards soluciona? ¿Ese tipo de hazards, ocurre en el pipeline MIPS de 5 etapas? Justifique.

Pregunta 3 (10 puntos)

Explique el mecanismo de detección de interrupciones en un procesador 8086. En particular, explique cómo se obtiene la dirección de memoria de la rutina a ejecutar.

Pregunta 4 (10 puntos)

En una arquitectura con pipeline, explique la técnica de predicción de saltos. Explique qué tipos de hazards soluciona y por qué.

Ejercicio 1 (30 puntos)

Consideramos un computador equipado con un procesador Intel 8086 y una memoria caché de 1Kb. La cache está organizada en líneas de 4 bytes con función de correspondencia directa. Definimos una lista encadenada de enteros mediante las siguientes declaraciones

```
struct nodo {
    int dato;
    int sig // índice en arreglo del siguiente elemento de la lista (cero indica NULL)
};
var lista : array[N] of nodo;
```

El primer nodo del arreglo no se utiliza porque el índice 0 indica el final de la lista.

En esta computadora se ejecuta el siguiente fragmento de código:

```
código:    push bx      ; parámetro: índice dentro del arreglo lista.
           call incrementa,
```

donde la rutina `incrementa` suma uno a todos los campos `dato` de la lista recibida por parámetro.

```
void incrementa (int índice) {
    while (índice != 0) {
        lista[índice].dato++;
        índice = lista[índice].sig;
    }
}
```

1. Compile la rutina `incrementa`. La variable `lista` se encuentra a partir del desplazamiento cero con respecto a `ES`. El parámetro se recibe en el `stack` y debe ser retirado. No es necesario conservar el valor de los registros.
2. Calcule la tasa de aciertos en caché en función del largo de la lista, cuando se ejecuta el fragmento de código ubicado en la etiqueta `código`. Inicialmente la caché está vacía. El tamaño `N` del arreglo es 256. Si el resultado depende del contenido de la lista, realice el cálculo para el peor caso (mínima tasa de aciertos).

Solución:

Parte A

```

incrementa proc
    pop ax
    pop bx
    push ax
loop:
    or bx, bx
    jz fin
    shl bx, 1
    shl bx, 1
    mov cx, es:[bx]
    inc cx
    mov es:[bx], cx
    mov bx, es:[bx+2]
    jmp loop
fin:
    ret
incrementa endp

```

Cada nodo de la lista ocupa 4 bytes, que es exactamente el tamaño de las líneas de caché.

Como la lista comienza en una dirección múltiplo de 4 (de hecho múltiplo de 16 de por tener desplazamiento 0 con respecto al inicio de un segmento), cada nodo de la lista se almacena completamente en una línea de cache. Como hay $1024/4 = 256$ líneas de caché y hay $N = 256$ elementos en el arreglo `lista`, toda la lista cabe completamente en la caché.

Por cada nodo de la lista se generan 3 accesos a memoria, 2 para la instrucción `inc` y uno para el acceso al campo `sig`, de los cuales el primer acceso al bloque genera un fallo y los dos siguientes son aciertos.

En resumen, la tasa de aciertos es $2/3$.

Ejercicio 2 (30 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) Calcule los MIPS de cada combinación.

Solución ejercicio 2:

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

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

	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 (basado en los datos de la parte B):

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

	Compilador A	Compilador B
rAlpha	$\frac{9000}{1.42 * 10^{-6} * 10^6} = 6338$	$\frac{7000}{1.36 * 10^{-6} * 10^6} = 5158$
c86	$\frac{2500}{1.07 * 10^{-6} * 10^6} = 2336$	$\frac{2000}{1.46 * 10^{-6} * 10^6} = 1369$

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.