

Complemento de Arquitectura de Computadoras

Solución Examen Julio 2016

- 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)

Suponga un procesador con las siguientes etapas:

1. Instruction Fetch – 50 ns
2. Instruction Decode – 20 ns
3. Memory Read – 50 ns
4. Execute – 30 ns
5. Memory Write – 50 ns

Se quiere implementar el mismo ciclo de CPU pero en pipeline. Indique el ciclo de reloj del nuevo CPU y la frecuencia de ejecución de instrucciones (throughput, o cantidad de instrucciones por segundo) para ambos CPUs. Para la CPU en pipeline considere el caso ideal si hazards.

Pregunta 2 (10 puntos)

Explique qué es un hazard WAW y en que tipo de pipelines se puede producir. Describa una técnica para solucionarlo.

Pregunta 3 (10 puntos)

Explique que es DMA y qué problema intenta resolver. Describa el problema de "robo de ciclos" del DMA a la CPU.

Pregunta 4 (10 puntos)

Explique en qué consiste un "benchmark suite", y comente en particular las especificaciones de SPEC. Razone acerca de la conveniencia de sumarizar los resultados de los benchmarks con medias aritméticas y medias geométricas, cuáles son más adecuadas?

Problema 1 (30 puntos)

Sea la siguiente estructura de datos para nodos de un árbol:

```
struct nodoA{
    int entero;
    unsigned char izq;
    unsigned char der;
}
```

Y considere un árbol definido como:

```
nodoA[256] arbol;
```

- a. Implementar en un lenguaje de alto nivel, la función recursiva

```
int sumarArbol(unsigned char indiceNodo);
```

La cual suma todos los elementos del árbol a partir del nodo pasado como parámetro. Considere que tanto *izq*, *der* e *indiceNodo* son índices en la estructura arbol. El árbol vacío es representado por el valor 0 en estos índices.

- b. Representar gráficamente un nodo del árbol en memoria, y calcular la cantidad de bytes necesaria para almacenar el árbol. Compilar en assembler 8086 sabiendo que el árbol se encuentra almacenado a partir de la dirección 0 del ES, índice nodo se recibe en AL y el resultado se debe devolver en el stack.

Solución:

a)

```
int sumarArbol(unsigned char indiceNodo){
    int suma = 0;
    if (indiceNodo != 0){
        suma += arbol[indiceNodo].entero;
        suma += sumarArbol(arbol[indiceNodo].izq);
        suma += sumarArbol(arbol[indiceNodo].der);
    }
    return suma;
}
```

b)

```
+-----+
| entero high |
+-----+
| entero low  |
+-----+
| izq        |
+-----+
| der        |
+-----+
```

Cada nodo ocupa 4 bytes, luego para almacenar 256 nodos se necesita 1KB.

```
SUMARARBOL PROC
    PUSH BP          ; reservo espacio para el parámetro retorno
    PUSH BP
    MOV BP, SP
    PUSH BX
    PUSH CX

    XOR CX, CX      ; suma = 0
    CMP AL, 0
    JE FIN
    ; si indiceNodo != 0
    XOR AH, AH
    ADD AX, AX
```

```

ADD AX, AX      ; AX = AL * 4, convierte indice en puntero
MOV BX, AX
ADD CX, ES:[BX] ; suma += arbol[indiceNodo].entero
PUSH AX
MOV AL, ES:[BX + 2]
CALL SUMARARBOL
POP DX
ADD CX, DX
POP AX
MOV AL, EX:[BX + 3]
CALL SUMARARBOL
POP DX
ADD CX, DX
FIN:
MOV DX, [BP + 4] ; BX = IP
MOV [BP + 2], DX
MOV [BP + 4], CX
POP CX
POP BX
POP BP
RET
ENDP

```

Problema 2 (30 puntos)

Considere un CPU de 16 bits, que emite direcciones de 16 bits (la cual permite direccionar completamente la memoria principal), donde la memoria es direccionable por bytes. La CPU opera con una memoria DRAM de 32 bloques y una memoria Cache de 8 líneas.

- Indique cómo se interpreta la dirección de memoria en la memoria cache para una función de correspondencia Asociativa de 2 vías.
Sugerencia: Calcular el tamaño de la memoria principal y luego el tamaño del bloque.
- Considere una cierta carga de trabajo donde el *hit* rate de las instrucciones de acceso a memoria resulta de 0.8. Suponga que la ejecución de una instrucción de memoria demora 2 ciclos en caso de que ocurra hit en el cache y 5 ciclos en caso de que ocurra miss. calcule el CPI de las instrucciones de memoria.
- Suponga una carga de trabajo de 2 millones de instrucciones, de las cuales el 50% son instrucciones de memoria y en donde el resto de las instrucciones tiene CPI = 3. La carga de trabajo ejecuta en 1 segundo. Calcule la frecuencia de trabajo de la máquina.

Solución:

a)

Si la caché tiene 8 líneas y los conjuntos son asociativos de dos vías, en total hay 4 conjuntos, por lo que se precisan 2 bits para el conjunto. En total hay 64 KB de memoria principal, y por lo tanto cada bloque tiene $64\text{KB} / 32 = 2 \text{ KB}$. Por lo tanto se precisan 11 bits para direccionar el byte. Como la dirección tiene 16 bits, los restantes 3 bits son para el tag.

b)

$$\text{CPI} = 0.8 * 2 + 0.2 * 5 = 2.6$$

c)

$$t_{\text{cpu}} = \# \text{instrucciones} * \text{CPI} * t_{\text{clk}}$$

$$\text{CPI}_{\text{total}} = 0.5 * 2.6 + 0.5 * 3 = 2.8 \text{ ciclos/seg.}$$

$$1 = 2:000.000 * 2.8 * t_{\text{clk}} \Rightarrow t_{\text{clk}} = 17,8 \text{ us} \Rightarrow f = 5.6 \text{ mhz}$$