

Examen de Arquitectura de Computadoras

22 de julio del 2016

Instrucciones:

- Indique su nombre, apellido y número de cédula en todas las hojas que entregue.
- Escriba las hojas de un solo lado.
- Las hojas deben estar numeradas y en la primer hoja debe escribirse el total.
- Empiece cada ejercicio en una hoja nueva.
- No puede utilizar material ni calculadora.
- **Apague su celular.**
- La duración del examen es de tres horas. **En dicho tiempo debe también completar sus datos.** Solo se contestarán dudas de letra. No se aceptan preguntas en los últimos 30 minutos del examen.
- Para aprobar debe contestar correctamente un ejercicio entero y dos preguntas.

Pregunta 1

Indique signo, exponente, mantisa y valor real del mínimo número positivo normalizado en punto flotante IEEE 754 de precisión simple.

Pregunta 2

Explique los modos de direccionamiento: inmediato, directo a registro y directo a memoria.

Pregunta 3

Minimice la siguiente expresión booleana utilizando mapas de Karnaugh $F(A,B,C,D) = \Sigma(0,2,4,7)$

Pregunta 4

Considere una CPU de 16 bits con memoria principal de 64 KB y una caché de 32KB con correspondencia asociativa de 2 vías. El tamaño de bloque es de 16 bytes.

Indique como se interpretan las direcciones de memoria, desde el punto de vista del sistema de memoria caché, especificando cuantos bits corresponden a cada campo.

Problema 1

La empresa **Medi2** quiere desarrollar, en base a un procesador dedicado, un contador digital de picos de consumo eléctrico. El contador cuenta con un display y un botón para encender/apagar el display.

El medidor debe tomar el consumo instantáneo 32 veces por segundo y cada un segundo, mostrar el máximo leído en el último segundo.

Para obtener el consumo instantáneo es necesario leer el puerto de E/S de sólo lectura de 16 bits en la dirección MEDIDOR. El valor leído consiste en 3 dígitos BCD y un nibble de paridad bit a bit impar.

valor + paridad = 1010 0101 0011 0011 => **VÁLIDO**

Si la paridad es incorrecta no debe considerarse dicha lectura para calcular el máximo. En caso de descartar todas las medidas, se debe mostrar todos los dígitos en cero.

La pantalla led cuenta con 4 dígitos y para mostrar un valor se debe escribir en el puerto de E/S

de sólo escritura LEDS, dónde cada dígito se expresa en BCD.

Para apagar o prender la pantalla el dispositivo posee un botón que invoca a la rutina de interrupción `toggle()`, la cual prende la pantalla si está apagada o la apaga en caso contrario. Para apagar la pantalla es necesario escribir 0xF para cada dígito.

A modo de ahorrar energía, **no se deben tomar consumos instantáneos mientras esté apagada la pantalla.** Al encender, la pantalla demora 1 segundo en mostrar la primer medida.

Se cuenta con una rutina de interrupción llamada `timer()`, la cual es invocada con una frecuencia de 64 Hz.

Se pide:

Escribir en un lenguaje de alto nivel (preferentemente C) todas las rutinas necesarias para implementar el sistema.

Problema 2

Se cuenta con el siguiente código que calcula la suma de los enteros impares en un árbol.

```
typedef struct {
    nodo* izq, der;
    short valor;
} nodo;

short sumarImpares(nodo* arbol){
    if(arbol == NULL){
        return 0;
    }
    int suma = 0;
    if (arbol->valor % 2 == 1){
        suma = arbol->valor;
    }
    suma += sumarImpares(arbol->izq);
    suma += sumarImpares(arbol->der);
    return suma;
}
```

Se pide:

Parte a) Compilar la rutina en Intel 8086 sabiendo que el parámetro árbol se recibe en el stack. Todos los punteros son representados como desplazamientos con respecto a ES. Debe conservarse los registros.

Parte b) Respecto al uso de memoria en el problema planteado indique:

- i) El tamaño máximo posible del árbol en cantidad de nodos.
- ii) El máximo consumo de stack para un árbol cualquiera de N nodos.
- iii) El árbol más grande que puede ejecutar su solución.

Respuesta 1

Indique signo, exponente, mantisa y valor real del mínimo número positivo normalizado en punto flotante IEEE 754 de precisión simple.

El normalizado más chico (mayor que cero) es el siguiente:

signo = 0
 exponente = 00000001
 mantisa = 00...00 (23 ceros)

El valor real de ese número es:

$$r = (-1)^0 * 2^{-127+1} * 1,0 = 2^{-126}$$

Respuesta 2

Explique los modos de direccionamiento: inmediato, directo a registro y directo a memoria.

Inmediato: El operando se guarda directamente en la instrucción.

Directo a registro: El operando está en un registro, el cual está codificado en la instrucción.

Directo a memoria: El operando está en la memoria principal y la dirección de memoria se encuentra en la instrucción.

Respuesta 3

Minimice la siguiente expresión booleana utilizando mapas de Karnaugh $F(A,B,C,D) = \Sigma(0,2,4,7)$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

ab\cd	00	01	11	10
00	1	0	0	1
01	1	0	1	0
10	0	0	0	0
11	0	0	0	0

$$F = !a!c!d + !a!b!d + !a b c d$$

Respuesta 4

Considere una CPU de 16 bits con memoria principal de 64 KB y una caché de 32KB con correspondencia asociativa de 2 vías. El tamaño de bloque es de 16 bytes.

Indique como se interpretan las direcciones de memoria, desde el punto de vista del sistema de memoria caché, especificando cuantos bits corresponden a cada campo.

Para saber la cantidad de conjuntos simplemente se divide el tamaño de la memoria cache entre el tamaño de cada conjunto, siendo este dos veces el tamaño de una línea (equivalente al tamaño de dos bloques).

$$\text{Tamaño de cada conjunto} = 2^4 * 2 = 2^5 \text{ bytes}$$

$$\text{Cantidad de conjuntos} = 32 \text{ kbytes} / 2^5 = 2^{15} / 2^5 = 2^{10}$$

Tag	Conjunto	Byte
2 bits	10 bits	4 bits

Solución Problema 1

```
#define APAGADA 0
#define PRENDIDA 1
#define TRUE 1

int tics, maximo;
unsigned char pantalla;

void interrupt timer() {
    int medida;

    if (pantalla) {
        if ((tics % 2) == 0) {
            medida = in(MEDIDOR);
            if (((medida >> 12) ^ ((medida >> 8) & 0x0F) ^ ((medida >> 4)
& 0x0F) ^ 0x0F) == (medida & 0x0F)) {
                medida = medida >> 4;
                if (medida > maximo) maximo = medida;
            }
        }
        tics++;
        if (tics == 64) {
            tics = 0;
            out(LEDS, maximo);
            maximo = 0;
        }
    }
}

void interrupt toggle() {

    if (pantalla) {
        pantalla = APAGADA;
        out(LEDS, 0xFFFF);
    }
    else {
        pantalla = PRENDIDA;
        tics = 0;
    }
}

void main() {
    pantalla = PRENDIDA;
    tics = 0;
    // instalo rutinas de interrupcion
    enable();
    while(TRUE);
}
```

Solución Problema 2**Parte A)**

```

PROC SUMARIMPARES
    PUSH BP
    MOV BP, SP
    PUSH CX
    PUSH BX
    PUSH AX

    XOR CX, CX                ; suma = resultado = 0

    MOV BX, [BP + 4]         ; BX = arbol
    CMP BX, 0
    JE FIN

    MOV AX, ES:[BX + 4]      ; AX = valor
    AND AX, 1
    JZ RECURSIVO            ; salteo la suma si Z = 0 (es decir, que el bit menos
                            ; significativo de AX vale 0 y por tanto el valor es par)

    ADD CX, ES:[BX + 4]

RECURSIVO:
    PUSH ES:[BX]
    CALL SUMARIMPARES
    POP AX
    ADD CX, AX
    PUSH ES:[BX + 2]
    CALL SUMARIMPARES
    POP AX
    ADD CX, AX

FIN:
    MOV [BP + 4], CX        ; pongo el resultado donde venía el parámetro
    POP AX
    POP BX
    POP CX
    POP BP
    RET

ENDP

```

Parte B)

I) Cada nodo ocupa 6 bytes y se almacena en el segmento extra. La cantidad máxima de nodos para un árbol se calcula como el tamaño del segmento dividido el tamaño del nodo. Teniendo en cuenta el desplazamiento cero no puede utilizarse como un nodo válido del árbol (porque 0 indica el final de una rama).

$$\# \text{nodos máxima} = 64K - 1 / 6 = 65535 / 6 = 10922$$

II) Tanto el paso base como el recursivo consumen 8 bytes de contexto más 4 de la llamada (dos de parámetro y dos de IP).

El peor caso de consumo ocurre cuando el árbol degenera en una lista. Para un árbol con estas características, la función requiere una llamada para el primer nodo y una adicional por cada otro nodo más el nodo vacío del final (paso base), es decir, N pasos recursivos y 1 paso base.

Consumo(arbol N nodos) = $(N+1) * 12$ bytes

III) Tamaño mínimo de stack = $C(10922) = 10922 * 12$ bytes = 131904 bytes. No entra en un segmento. Entonces $(N+1) * 12$ bytes $\leq 64K \implies N_{\max} = 65536/12 - 1 = 5460$.