

# Taller

## Preparación del Segundo Parcial

### **Pregunta 4 (Diciembre 2014)**

En la dirección de memoria ES:[DI] está guardado el entero -15.

- Sabiendo que el valor de ES es 0x1030 y el de DI 0x22AA, ¿en qué posición de memoria física está guardado el entero?
- Especifique el valor de DS para que el acceso a DS:[0x1A] coincida con el entero guardado.
- Explique por qué no es posible especificar un valor para DS de forma tal que DS:[0x5] coincida con la dirección física del dato guardado.

### **Pregunta 3 (Febrero 2015)**

¿Qué son los hazards de control? Explique por qué la predicción de saltos ayuda a mitigar los hazards de control.

### **Pregunta 3 (Diciembre 2016)**

Considere una CPU de 32 bits con memoria totalmente direccionable de a byte y una cache de 16KB con líneas de 32 bytes de largo con una función de correspondencia directa. Considere que la cache tiene guardada la dirección 0x4C3D2E1F (y su bloque asociado).

Indique 2 nuevas direcciones de forma tal que la primera resulte en un cache hit y la segunda resulte en un remplazo de la dirección 0x4C3D2E1F en la cache.

### **Pregunta 4**

Explique qué es una dependencia de tipo WAW y en que tipo de procesadores aparece.

### **Pregunta 5**

Explique los conceptos de ILP, MLP y cómo se relacionan.

### **Pregunta 6**

Describa la etapa commit en procesadores superescalares. Explique cuál es su utilidad.

### **Pregunta 7**

Describa la Ventana de Instrucciones (Instruction Window). Explique cómo ayuda a la explotación del ILP.

## **Problema 1 (Diciembre 2015)**

Con la cercanía del verano **Cabañas Sécheresse** desea instalar un tanque de agua en cada una de las cabañas del complejo turístico para mejorar la disponibilidad del líquido elemento para los huéspedes, pero al mismo tiempo debe facturar el consumo de agua de cada una de las cabañas de forma independiente.

Para esto se instalará un detector de nivel mínimo de agua (DNM) y una electroválvula (EV) en cada tanque, y un único medidor de caudal (MC) en la entrada general de agua, la que luego llega a los 8 tanques disponibles, junto a un microcontrolador dedicado que comandará estos elementos a los efectos de cumplir el objetivo.

El sistema deberá mantener el agua por encima del nivel mínimo en todos los tanques y contabilizar el consumo de agua de cada uno. Se asume que el caudal de agua que ingresa se reparte en partes iguales entre todos los tanques que tengan la electroválvula abierta.

Además, deben controlarse los siguientes aspectos:

- Una vez alcanzado el nivel mínimo de agua en un tanque debe mantenerse abierta la electroválvula durante 20 minutos más.
- Las electroválvulas no pueden permanecer abiertas por más de 2 horas. Si se alcanza este tiempo deben cerrarse y activarse la alarma.

La situación de alarma debe indicarse manteniendo encendida una sirena durante 1 minuto.

Se dispone de los siguientes puertos de E/S:

- Byte de sólo escritura para abrir (1) o cerrar (0) una EV en la dirección CONTROL\_EV. El bit *i* controla la *i*-ésima EV (numeradas de la 0 a la 7).
- Byte de sólo lectura que permite conocer si el nivel de un DNM está por encima del mínimo (1) o no (0) en la dirección ESTADO\_DNM. El bit *i* consulta el estado del *i*-ésimo DNM (numerados del 0 al 7).
- Byte de sólo escritura en la dirección SIRENA, cuyo bit menos significativo es utilizado para encender (1) o apagar (0) la sirena.
- Palabras de 16 bits de sólo escritura donde se debe registrar el valor actual del consumo de agua (en litros) de cada tanque, en las direcciones CONSUMO+*i*, donde *i* es el número del tanque (numerados del 0 al 7), es decir las direcciones CONSUMO, CONSUMO+1, CONSUMO+2, etc

Se dispone de un *timer* que interrumpe a una frecuencia de 5Hz invocando a la rutina **tiempo()**. El sensor de caudal genera una interrupción por cada litro de agua que pasa por él, que invoca a la rutina **caudal()**.

**Se pide:**

Implementar todas las rutinas necesarias en alto nivel (preferentemente en C).

## Problema 2 (Diciembre 2014)

Se considera la siguiente estructura de árbol binario:

```
struct
{
    unsigned char hijoIzq;
    unsigned char hijoDer;
    unsigned short dato;
} nodo;
nodo arbol[256]
```

En donde hijoIzq e hijoDer son los índices a los subárboles izquierdo y derecho respectivamente para un nodo dado.

El valor **0** para un índice indica que no existe nodo asociado. El árbol tiene por lo menos un elemento y no hay nodos con **dato** repetido en el mismo.

### Se pide:

**a)** Escribir en alto nivel un procedimiento recursivo que retorna el dato mayor que se encuentra en el árbol y su índice asociado en el arreglo.

El procedimiento tiene la siguiente firma:

```
void buscoMayor(unsigned char indice, out short mayor, out char indiceMayor)
```

Compilar la rutina en assembler 8086 sabiendo que la variable **arbol** se encuentra a partir de la dirección segmentada **ES:0000**. El parámetro se recibe en el stack y el resultado también se debe retornar por el stack. La rutina debe conservar todos los registros de propósito general.

**b)** Calcular el tamaño mínimo que debe tener el stack para que la función pueda ser ejecutada en todos los casos, cualquiera sea el tamaño y topología del árbol.