

**Normas del examen:**

- No se podrá emplear documentación adicional a la del examen.
- No se podrán desgrapar las hojas.
- Tiempo del examen 2 horas.
- No se podrá emplear ningún tipo de calculadora, ni teléfono móvil, ni computador portátil o PDA.
- Es necesario obtener al menos 0,6 puntos en la parte de laboratorio para superar la asignatura completa

---

---

**Teoría (4 puntos)**

- ¿Para qué se utiliza la técnica de redondeo al pasar la información de la ALU a los registros o la memoria?. Determinar las tres técnicas utilizadas más comunes. **(1,5 puntos)**

- Calcular el cociente entre  $D: 0100110$  y  $d: 1110$ , ambos representados en binario puro utilizando el algoritmo de división sin restauración **(1,5 puntos)**



**Problema 1 (2 puntos)**

Sea un computador con 8 registros, cuya longitud de palabra es de 2 bytes. Diseñar los formatos para las instrucciones de tipo Registro-Registro, utilizando la técnica de “expansión de código de operación” de modo que permita:

- 127 instrucciones de 3 operandos
- 6 instrucciones de 2 operandos
- 15 instrucciones de 1 operando
- 8 instrucciones de 0 operandos

**Problema 2 (3 puntos)**

Sea un ordenador con un procesador de 8 bits, con un sistema de memoria de las siguientes características:

- Memoria principal:
  - Capacidad 4 GB
  - Entrelazada simple de orden inferior con 16 módulos
  - Tiempo de acceso de 32 ns.
- Memoria caché:
  - Dos módulos de memoria caché independientes (instrucciones y datos)
  - Bloques de 16 bytes.
  - Tiempo de acceso de 4 ns.
  - **Política de ubicación:** asociativa por conjuntos de dos bloques.
  - **Política de actualización:** escritura inmediata sin ubicación.
  - **Política de reemplazo:** FIFO.
  - Capacidad de ambas caches de 64 Kbytes.

En este computador se ejecuta el siguiente código:

```
; Realizar el cálculo del total de una compra almacenando el
; resultado en la posición correspondiente a la variable Total
; mediante el producto de las listas precio y cantidad
WHILE (i ≤ nprod) DO
BEGIN
    Total := Total + precio[i] * cantidad[i];
    i:= i + 1;
END;
```

Si inicialmente, la memoria caché de datos se encuentra vacía:

- a) Indicar la correspondencia entre una dirección de memoria principal y una de memoria cache. **(1 punto)**
- b) Indicar la traza de los 3 primeros accesos, a los datos, realizada por este fragmento de código suponiendo que: **(0,5 puntos)**
  - Las variables  $i$ ,  $nprod$  y  $Total$  se almacenan en dos de los registros del procesador, inicializadas a 0,
  - Las listas  $precio$  y  $cantidad$  se almacenan a partir de las direcciones de memoria principal 80018h y 70018h respectivamente.

- c) Indicar en cuáles de las anteriores referencias a la memoria principal producen fallos en la memoria caché, considerando las 3 iteraciones que realiza el bucle.  
**(0,5 puntos)**
- d) Calcular la tasa de aciertos de la memoria caché de datos para el anterior fragmento de programa.  
**(1 punto)**



**Laboratorio (1 punto)**

Sea el siguiente modelo correspondiente a la arquitectura estructural de un multiplexor de dos entradas y una salida implementado en VHDL

```
ARCHITECTURE estructural OF mux21 IS
COMPONENT inv
    PORT(y:in bit; z:out bit);
END COMPONENT;
COMPONENT and2
    PORT(x:in bit; y:in bit; z:out bit);
END COMPONENT;
COMPONENT or2
    PORT(x:in bit; y:in bit; z:out bit);
END COMPONENT;
-- declaracion de las señales
SIGNAL ctrl-n, n1, n2: bit;
--indicamos ubicación de la arquitectura de los componentes
FOR P1:inv USE ENTITY WORK.inv(func);
FOR P2:and2 USE ENTITY WORK.and2(func);
FOR P3:and2 USE ENTITY WORK.and2(func);
FOR P4:or2 USE ENTITY WORK.or2(func);
--conexiones
P1:inv PORT MAP (ctrl, ctrl-n);
P2:and2 PORT MAP (ctr-n, a, n1);
P3:and2 PORT MAP (ctrl, b, n2);
P4:or2 PORT MAP (n1, n2, z);
END estructural;
```

Se pide:

- Definir la entidad mux21 **(0,4 puntos)**
- Definir la arquitectura de manera comportamental **(0,4 puntos)**
- Definir la arquitectura mediante flujo de datos **(0,2 puntos)**

