

# Arquitectura de Computadoras

## Solución Parcial 2017

- El parcial consta de 5 preguntas que se deben responder por escrito en hojas aparte. Completar TODAS las hojas con el nombre y el número de cédula. Numerarlas y escribir el total en la primer hoja.
- Utilizar una única carilla e iniciar cada respuesta en una hoja nueva.
- No se puede utilizar material de ningún tipo. **Apagar celulares.**
- Se dispone de cartillas Intel 8086.
- El parcial dura 1 hora y media.
- Se contestarán preguntas hasta 20 minutos antes de la hora de finalización.

### Pregunta 1

Por un canal con ruido se envía un dígito (0-9) representado en BCD usando códigos de hamming (7,4). Si se sabe que hubo a lo sumo un error y la tira recibida fue: 0000101, ¿Qué dígito BCD fue enviado originalmente ? (  $a_4 a_3 a_2 p_3 a_1 p_2 p_1$  )

**Nota:**  $p_1 = a_4 \otimes a_2 \otimes a_1$  ,  $p_2 = a_4 \otimes a_3 \otimes a_1$  ,  $p_3 = a_4 \otimes a_3 \otimes a_2$  .

**Solución:**

$$s_0 = p_1 \otimes a_4 \otimes a_2 \otimes a_1 = 1 \otimes 0 \otimes 0 \otimes 1 = 0$$

$$s_1 = p_2 \otimes a_4 \otimes a_3 \otimes a_1 = 0 \otimes 0 \otimes 0 \otimes 1 = 1$$

$$s_2 = p_3 \otimes a_4 \otimes a_3 \otimes a_2 = 0 \otimes 0 \otimes 0 \otimes 0 = 0$$

Por lo tanto, el bit de error es el 010b = 2, es decir, el bit p2, por lo tanto el código BCD enviado no cambió, es: 0001 y por tanto el dígito enviado es el "1".

### Pregunta 2

¿Cuál es la función de las entradas CS y OE en una ROM? Indique qué se obtiene en la salida para cada una de las combinaciones de CS y OE.

**Solución:**

La función de la entrada CS (chip select) es "habilitar" la ROM desde el punto de vista lógico. Si CS = 0, la salida es 0, independientemente del contenido.

La función de la entrada OE (output enable) es "habilitar" la ROM desde el punto de vista eléctrico. OE = 0 poner la salida en tercer estado, mientras que si es 1 entonces la memoria tendrá el nivel lógico que corresponda.

CS = 1 y OE = 1: Salida de la ROM (valor almacenado en la dirección indicada)

CS = 1 y OE = 0: Tercer estado

CS = 0 y OE = 1: Cero

CS = 0 y OE = 0: Tercer estado

### Pregunta 3

En jerarquía de memoria, explique por qué interpretar una dirección de memoria como:

| **LÍNEA** | **TAG** | BYTE |

no aprovecharía de la mejor forma el principio de localidad.

**Solución:**

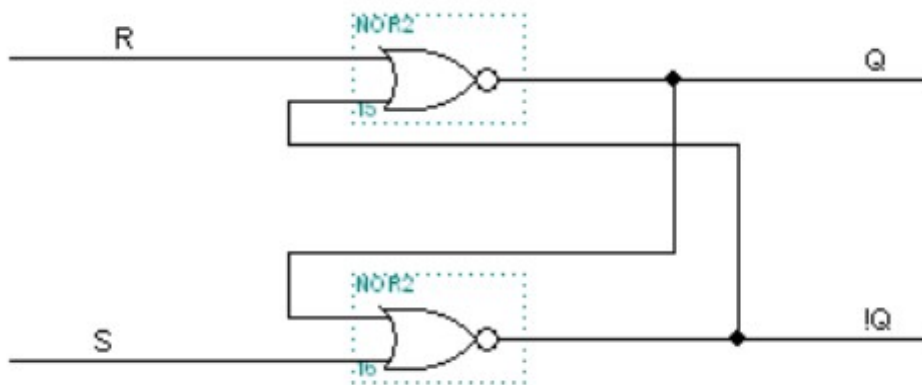
En caso de interpretarse la dirección de memoria de esa forma, bloques contiguos de memoria se mapearían a la misma línea del caché, provocando que al accederlos de forma alternada se desplacen unos a otros, generando más misses por conflicto que con la interpretación TAG | LINEA | BYTE.

Ejemplo: Supongamos una dirección de 16 bits, y una caché tal que la sección TAG ocupa 8 bits, la LINEA 4 bits y la sección BYTE otros 4 bits. Si se acceden a las direcciones 0x0010, 0x0020 y 0x0010 nuevamente, como los bloques de las tres direcciones van mapeadas a la misma línea (con distinto TAG cada vez), se sobre-escribirían, logrando hit rate = 0. Sin embargo, con la interpretación TAG | LINEA | BYTE, los bloques se almacenarían en líneas diferentes, logrando que el tercer acceso sea hit (y por tanto hit rate = 1/3).

**Pregunta 4**

Dibuje un flip-flop RS y justifique mediante ejemplos por qué es un circuito secuencial.

**Solución:**



Es un circuito secuencial porque su salida depende no solo de las entradas actuales, sino de las anteriores. Ejemplo, si en tiempo t la entrada R = S = 0, el valor de Q está determinado por el valor de su última entrada. Si R(t-1) = 0 y S(t-1) = 1, el valor de Q(t) es 1. Si R(t-1) = 1 y S(t-1) = 0, el valor de Q(t) es 0. Esto es porque esos son las dos situaciones de estabilidad del circuito realimentado, ambas posibles, por lo que para determinar cual de las dos posibles se da, se debe recurrir a la historia previa de los valores de entrada.

**Pregunta 5**

¿Cuál es el valor del registro CX luego de ejecutar el siguiente fragmento de assembler x86?

Justifique

<pre> xor cx,cx mov ah, 2 mov al, 1 salto: shr ax,1 inc ax ; sigue a la derecha </pre>	<pre> mov bx, ax and bx, 1 jz fin inc cx jmp salto fin: ... </pre>
--	--

***Solución:***

El programa se puede ejecutar a mano y llegar a la conclusión de que  $CX = 8$ .

Se puede deducir que el patrón del programa consiste en shiftear el bit 2 de AH, una vez en cada iteración, hasta que en la última este bit se encuentre en la posición 0 de AL (es decir  $AX = 0000\ 0000\ 0000\ 0001$  luego de ejecutar el SHR). De este modo, al ejecutar  $INC\ AX$ , el valor de este registro será  $0000\ 0000\ 0000\ 0010$ , luego el AND con 1 da como resultado  $AX = 0$  y se salta hacia fin. Esto requiere nueve iteraciones, salteando el incremento en la última, lo cual da como resultado  $CX = 8$ .