

---

# Arquitectura de Computadoras

Parcial 2021 – 03/12/2021

---

- ◆ *El parcial consta de 5 preguntas que se deben responder por escrito.*
- ◆ *Se deben completar TODAS las hojas con el nombre y el número de cédula.*
- ◆ *Las hojas deben NUMERARSE y debe indicarse claramente el total de hojas utilizadas en la primer hoja (incluya el texto TOTAL DE HOJAS: [hojas utilizadas]).*
- ◆ *No se puede utilizar material de ningún tipo. Se deben apagar los celulares.*
- ◆ *La duración del parcial es de una hora y media. Se dispone de cartillas Intel 8086.*
- ◆ *Se contestarán preguntas sobre la letra del parcial hasta 20 minutos antes de la hora de finalización.*

## Pregunta 1

Explique qué tipo de circuito es la Unidad de Control de una CPU y qué acciones realiza. Describa los estilos de diseño de Unidad de Control vistos en el curso.

## Pregunta 2

Dados los siguientes códigos, ambos trabajando sobre un arreglo bidimensional definido como:

```
short arr[1024][100];
```

### Código 1:

```
for ( i = 0; i < 100 ; i++ ) {  
    for ( j = 0 ; j < 1024 ; j++ ) {  
        arr [ j ] [ i ] = 2 * arr [ j ] [ i ] ;  
    }  
}
```

### Código 2:

```
for ( i = 0; i < 1024 ; i++ ) {  
    for ( j = 0 ; j < 100 ; j++ ) {  
        arr [ i ] [ j ] = 2 * arr [ i ] [ j ] ;  
    }  
}
```

Y sabiendo que el arreglo se encuentra en memoria y que la caché es de 1KB, ¿cuál código aprovecha mejor la localidad espacial? ¿Por qué?

## Pregunta 3

¿Cuántas ROMs de 2Kx6 se precisan para construir una ROM de 1Kx12 ?  
Dibuje el circuito correspondiente

#### Pregunta 4

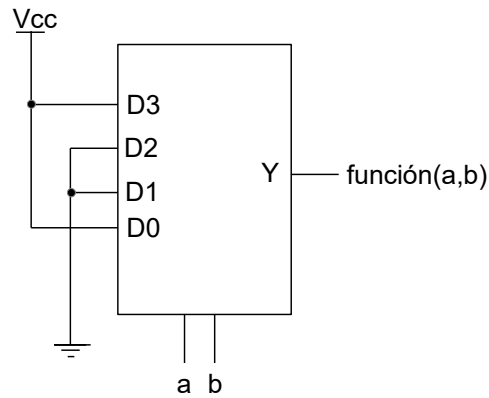
Considere dos procesadores diferentes (CPU1 y CPU2), los cuales implementan una misma caracterización de una arquitectura Von Neumann.

Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique su respuesta.

- Para todo programa en alto nivel que se compila y ensambla para la CPU1, el mismo código binario también puede ser ejecutado en la CPU2.
- Para todo programa en alto nivel, que se compila y ensambla para la CPU1 y además se compila y ensambla para la CPU2, la ejecución del programa en ambas CPUs durará la misma cantidad de ciclos de reloj.

#### Pregunta 5

a) Identificar qué función implementa un multiplexor 4x1 conectado de la siguiente manera:



b) Implemente dicha función en dos niveles con compuertas AND, OR y NOT.

## Solución

### **Respuesta Pregunta 1**

La unidad de control es una máquina secuencial. Su acción es realizar el ciclo de instrucción, un conjunto de acciones ordenado y secuencial para lograr ejecutar cada instrucción, aplicando una operación determinada sobre los operandos correspondientes y almacenando el resultado en un lugar indicado.

Las opciones vistas en el curso para el diseño de la unidad de control son el control cableado o la microprogramación.

En el caso del control cableado, la UC se diseña como un circuito secuencial utilizando, por ejemplo, la metodología de Máquinas de Estado. En el caso de la microprogramación, la UC se construye en base a una máquina secuencial más simple que ejecuta los microprogramas que contienen la secuencia de valores de las señales internas de control de la CPU para lograr la ejecución de las instrucciones.

### **Respuesta Pregunta 2**

En última instancia en el principio de localidad espacial la “cercanía” es un concepto relativo, determinado por el tamaño de la cache. Dos elementos dados de la estructura estarán “cerca” o “lejos” dependiendo de si su distancia es mayor o menor al tamaño de la cache.

Si la estructura pudiera almacenarse completamente en la memoria cache, entonces todos los elementos estarían “cerca” y no importaría la forma de recorrer la estructura para aprovecharse del principio.

Pero este no es el caso. La estructura propuesta ocupa 200 KBytes (1024 x 100 x 2 bytes) y la cache tiene un tamaño de 1KByte.

Por tanto el aprovechamiento del principio será relativo y dependerá de cómo se recorra la estructura.

Teniendo en cuenta que los arreglos se guardan de forma contigua en memoria y, en el caso de las matrices, por filas, vemos que el segundo código recorre la estructura en la forma en que está almacenada en memoria y por tanto es el que mejor aprovecha el principio.

Una forma alternativa de razonar este resultado es de la siguiente manera. Supongamos por un momento que la línea de la cache es de 16 bytes (el resultado no cambia si consideramos otros valores), con lo que tenemos 64 líneas de 16 bytes.

El segundo código al dar el primer miss accediendo a arreglo[0][0], logra que se traigan a la cache los elementos arreglo[0][1] a arreglo[0][7], produciendo hit en los próximos 7 accesos. El proceso se repite, dando siempre la misma relación de miss y hits.

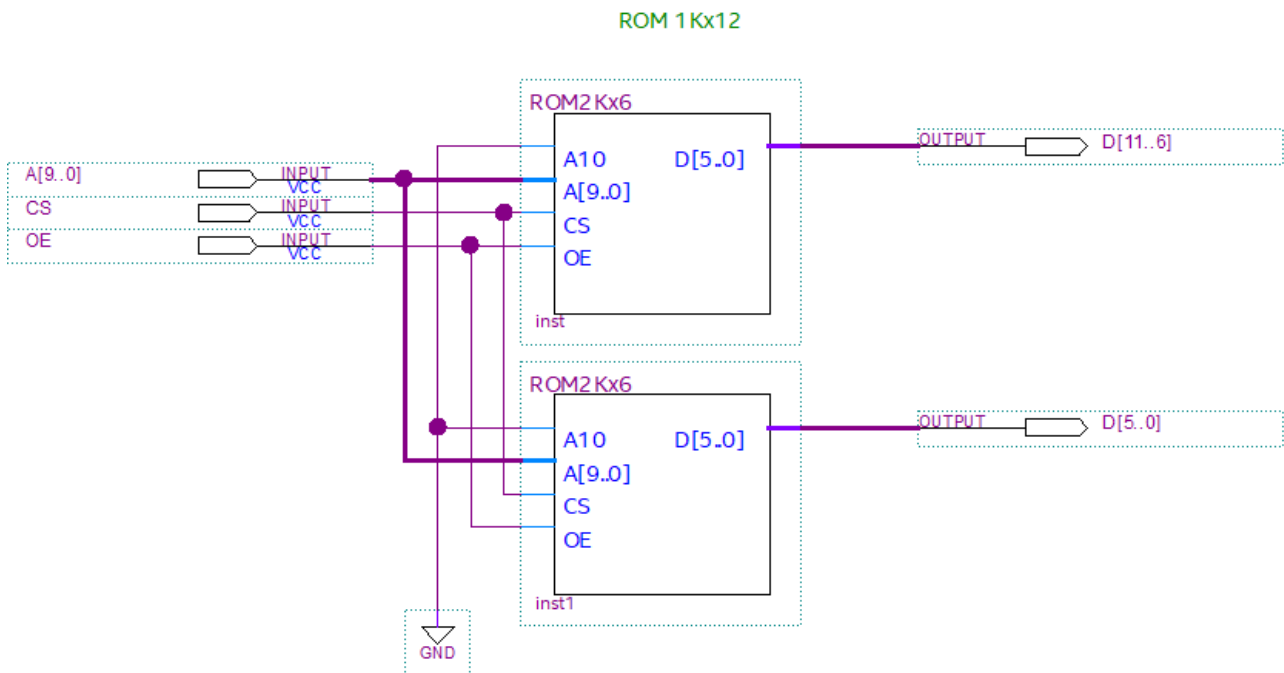
En cambio el primer código al acceder a `arreglo[0][0]` produce el mismo miss y el mismo efecto ya visto. Pero a continuación al acceder al `arreglo[1][0]` también va a dar miss y se van a traer a la cache los elementos `arreglo[1][1]` a `arreglo[1][7]`. Al acceder al `arreglo[2][0]` va a pasar lo mismo y así hasta llegar al `arreglo[63][0]` que provocará que la cache se llene. Por tanto al acceder al `arreglo[64][0]` se producirá el reemplazo del bloque que contiene a los elementos `arreglo[0][0]` a `arreglo[0][7]`. Notemos que estamos suponiendo correspondencia totalmente asociativa para simplificar el razonamiento.

Cuando se termine la primer iteración del bucle interno y se cambie de índice para empezar por `arreglo[0][1]`, ese elemento ya no estará en la cache por haber sido reemplazado y dará miss nuevamente.

Es fácil ver que como este proceso se repite continuamente, el primer código solo produce miss y ningún hit, por lo que claramente no aprovecha en nada el principio de localidad.

### Respuesta Pregunta 3

Se precisan 2 ROMs de 2Kx6, cada una de las cuales aportará la parte baja o alta de cada palabra de la ROM de 1Kx12. Notar que solamente se utiliza la mitad de capacidad de cada una de la ROMs individuales por lo que la entrada de dirección más significativa se conecta a una constante conocida (tierra = 0).



### Respuesta Pregunta 4

1) Verdadero. La caracterización de la arquitectura Von Neumann está compuesta por los distintos elementos señalados en el curso: set de instrucciones, formato de

instrucción, set de registros, modos de direccionamiento, manejo de entrada/salida e interrupciones.

En la literatura se la describe como la interfaz entre el hardware y el software, es decir, brinda la información necesaria para escribir programas en lenguaje de máquina.

Decir que dos CPUs poseen una misma caracterización de la arquitectura Von Neumann implica que pueden ejecutar todos los programas que cumplan la especificación, sin importar en cuál se ellas se haya compilado el mismo.

2) Falso. Si bien ambas CPUs pueden ejecutar el programa (por la parte anterior se desprende que el código binario es el mismo), no se especifica *cómo* lo hacen. Por ejemplo una de las CPUs podría incluir o no un *pipeline* en su organización, lo cual va a provocar que los programas ejecuten en una menor cantidad de ciclos de reloj. Otros ejemplos podrían ser que implementaran ciclos de instrucción distintos, o que una CPU tuviera su Unidad de Control diseñada por control cableado y la otra por microprograma.

### Respuesta Pregunta 5

a) La tabla de verdad que resulta del multiplexor conectado de esa manera es:

a	b	función(a,b)
0	0	1
0	1	0
1	0	0
1	1	1

Esta función es la negación del XOR (XNOR), que en definitiva es la función que indica si “a” es igual a “b”.

b) La mínima implementación de la función en dos niveles coincide con su suma de productos canónicos, al igual que el XOR. Por tanto la expresión queda:

$$\text{función}(a,b) = a.b + !a.!b$$

El circuito es:

