

# **EXAMEN - DISEÑO DE CIRCUITOS INTEGRADOS DIGITALES**

**ING. MARIANO MOREL**

**MAESTRÍA EN CIENCIAS DE LA INGENIERÍA – FIUBA**

## **Ejercicio 1**

**A - Implemente el código RTL de un sincronizador para N bits de palabra de datos y un largo L de cantidad de flip-flops de resolución. El protocolo de handshaking debe ser de dos fases.**

```
module sincronizador (  
    input wire clk_src,      // Reloj del launching domain  
    input wire clk_dst,      // Reloj del capturing domain  
    input wire [N-1:0] data_in, // Datos de entrada en el launching domain  
    output reg [N-1:0] data_out, // Datos de salida en el capturing domain  
    input wire req,          // Señal de solicitud del protocolo de handshaking  
    output reg ack           // Señal de reconocimiento del protocolo de handshaking  
);  
  
parameter N = 8;           // Número de bits en la palabra de datos  
parameter L = 2;           // Largo de flip-flops de resolución  
  
reg [N-1:0] data_sync [L-1:0]; // Arreglo para los flip-flops en el sincronizador  
integer i;  
  
always @(posedge clk_src) begin  
    if (req) begin  
        data_sync[0] <= data_in; // Primera etapa del sincronizador  
    end  
end  
  
always @(posedge clk_dst) begin  
    for (i = 1; i < L; i = i + 1) begin  
        data_sync[i] <= data_sync[i-1]; // Cadena de flip-flops de sincronización  
    end  
    data_out <= data_sync[L-1];  
    ack <= req; // Protocolo de handshaking  
end  
endmodule
```

El módulo es un **sincronizador** de datos entre dos dominios de reloj diferentes (dos señales de reloj que pueden estar en diferentes frecuencias o fases). Se utiliza para transferir datos de manera segura entre componentes que operan con relojes independientes, evitando problemas de sincronización y asegurando que los datos se capturen correctamente .

**B- Suponga un proceso CMOS con los siguientes parámetros:**

$$TW = 75ns$$

$$\tau_0 = \tau / (A - 1) = 0,23ns$$

$$tsu = 1ns$$

**Si la frecuencia de operación del launching domain es de 20MHz y produce datos al mayor throuput posible, mientras que la frecuencia de operación capturing domain es de 200MHz, dimensione el/los sincronizador/es para un MTBF de 10 años.**

Para calcular el número de etapas de flip-flops necesarias en el sincronizador (L) para obtener el **Mean Time Between Failures (MTBF)** deseado de 10 años en el cruce de dominios de reloj, usamos la fórmula de MTBF para un sincronizador asíncrono en un proceso CMOS.

$$MTBF = \frac{e^{\frac{TW - t_{su}}{\tau}}}{f_{src} \cdot f_{dst}}$$

- TW: Tiempo de ventana (setup + hold time) para la captura
- $\tau$ : Constante de resolución del sincronizador
- $t_{su}$ : Setup time del flip-flop
- $f_{src}$ : Frecuencia del dominio de lanzamiento (20 MHz).
- $f_{dst}$ : Frecuencia del dominio de captura (200 MHz).

$$MTBF \text{ (segundos)} = 315.576.000 \text{ s} \rightarrow \text{incluye un día más por año bisiesto}$$

Haciendo los cálculos correspondientes:

Valor requerido, despejando de la ecuación de MTBF:

$$TW - t_{su} = 12.76ns$$

Valor actual, haciendo la resta directamente de los datos dados:

$$TW - t_{su} = 74 \text{ ns}$$

Dado que el valor actual excede el valor requerido, solo una etapa de flip-flop en el sincronizador es suficiente para garantizar un MTBF de 10 años en este caso.

**C- Como se modificaría el MTBF manteniendo el mismo largo L pero teniendo 4 inversores en los lazos de cada flip-flop de resolución?**

Si agregamos 4 inversores en los lazos de cada flip-flop de resolución, esto afecta la constante de resolución del sincronizador  $\tau$  ya está relacionada con la capacidad del circuito para resolver estados de meta-estabilidad. En un flip-flop, el lazo de realimentación que determina su estabilidad y velocidad de resolución está afectado por la cantidad de inversores, que incrementan la constante de tiempo de  $\tau$

Si consideramos que cada inversor contribuye linealmente al aumento de  $\tau$ , entonces la nueva  $\tau$  puede calcularse en función del número de inversores adicionales.

Supongamos que cada inversor añade una cantidad proporcional a la  $\tau$  original. Entonces, si se agregan 4 inversores, la nueva constante de resolución podría aproximarse como:

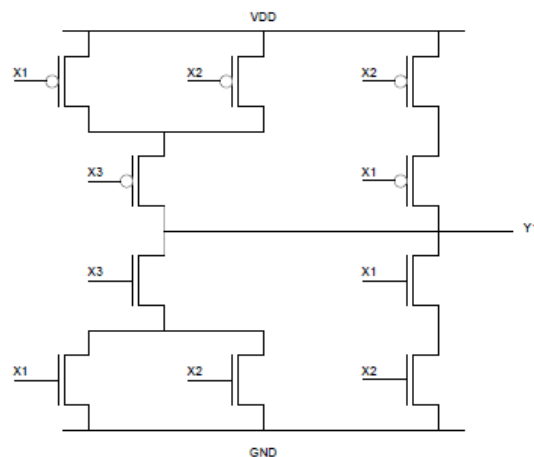
$$\tau_{\text{new}} = \tau (1 + k \times n_{\text{inv}}) \quad k: \text{Factor de aumento de la constante de resolución por cada inversor}$$

Como ejemplo, se asume un incremento del 10% por cada inversor en la constante de resolución. Con 4 inversores adicionales, tenemos un  $\tau_{\text{new}}$  un 40% mayor que la  $\tau$  original.

El MTBF daría mucho mayor a los 10 años, por lo que el sincronizador tendría nuevamente una probabilidad extremadamente baja de fallar en el cruce de dominios de reloj

**Ejercicio 2**

**A- Determine la función lógica del siguiente circuito:**



La tabla de verdad es:

X1	X2	X3	Y1
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Con las simplificaciones correspondientes aplicando algebra de Boole, la función lógica queda:

$$Y1 = \overline{X1} \cdot \overline{X2} + \overline{X3} \cdot (X1 \oplus X2)$$

O lo que es equivalente por De Morgan a:

$$Y1 = \overline{\left( \left( \overline{X1} \cdot \overline{X2} \right) \cdot \left( \overline{X3} \cdot (X1 \oplus X2) \right) \right)}$$

**B- Si para un proceso se cumple que el inversor es simétrico cuando  $W_p = 3W_n$ , dimensione los transistores para que la compuerta posea el mismo logical effort por entrada. Cual es dicho logical effort?**

Se parte del circuito anterior, con la relación del inversor simétrico  $W_p = 3W_n$ .

Por otro lado  $W_p = K \times W_n$  (mismo L), representa la relación de anchos entre NMOS y PMOS, asumiendo un  $K=2$ .

Lo que se necesita es lo siguiente: mirando el circuito, quiero que la resistencia equivalente que presenta el paralelo de los transistores PMOS con entradas X1 y X2, en serie con el transistor con entrada X3, sea igual a su versión NMOS. Para que esto sea posible, las resistencias de los transistores PMOS con entradas X1 y X2, deben ser de valor R y para el transistor con entrada X3, debe ser R/2. Sabiendo que si reduzco la resistencia a la mitad, la capacidad se duplica, para los dos primeros será igual a la  $C_{inv} \times K$  y para el restante  $C_{inv} \times k \times 2$ .

Para la versión pull-down NMOS, será lo mismo sin afectar la capacidad por K.

Para la parte derecha del circuito, el razonamiento es similar, por lo que queda, para los transistores PMOS, dos resistencias en serie R/2 cada una, lo mismo para la versión NMOS. Así las capacidades de entrada serán para cada PMOS,  $C_{inv} \times k \times 2$  y para cada NMOS,  $C_{inv} \times 2$ . Utilizando la relación  $g = C_{in}/C_{inv}$ , la capacidad de entrada, el logical effort de cada entrada y el total queda:

$$C_{in_{X1}} = 2 + 1 + 4 + 2 = 9$$

$$C_{in_{X2}} = 2 + 1 + 4 + 2 = 9$$

$$C_{in_{X3}} = 4 + 2 = 6$$

$$g_{X1} = 9/3 = 3$$

$$g_{X2} = 9/3 = 3$$

$$g_{X3} = 6/3 = 2$$

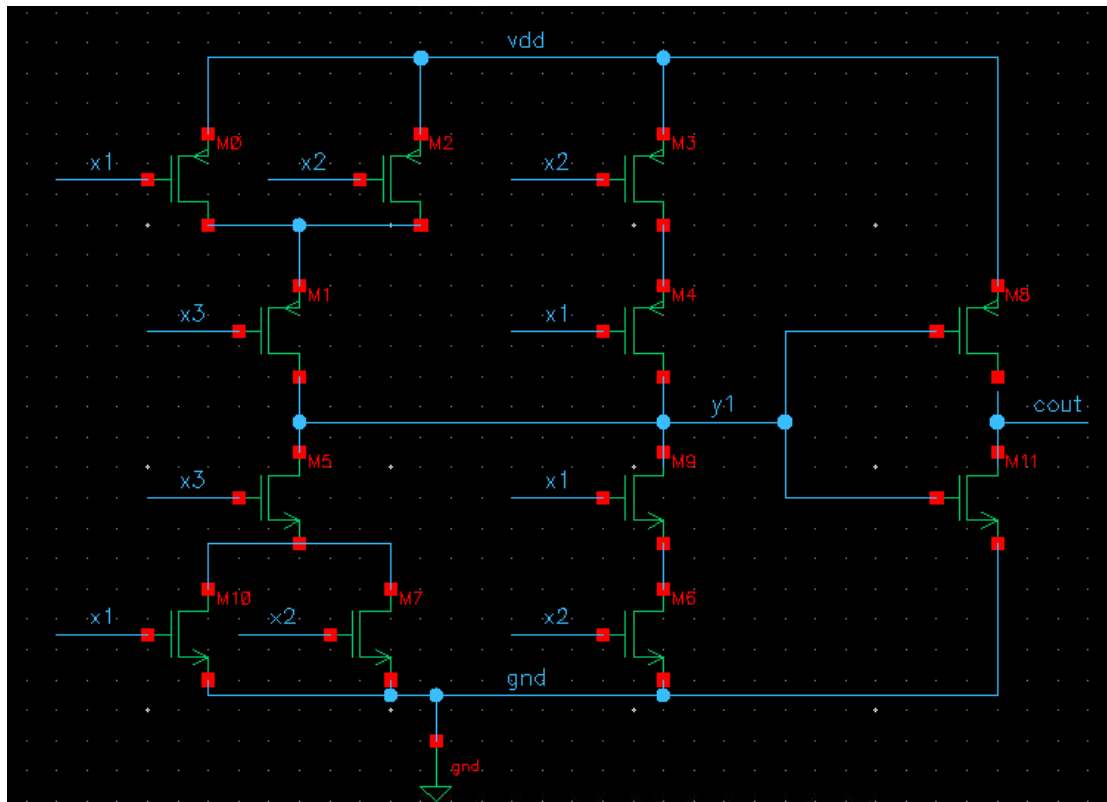
$$g_{total} = 3 + 3 + 2 = 8$$

**C- Cuál es la combinación de entradas que produce máximo delay y cuál produce mínimo delay.**

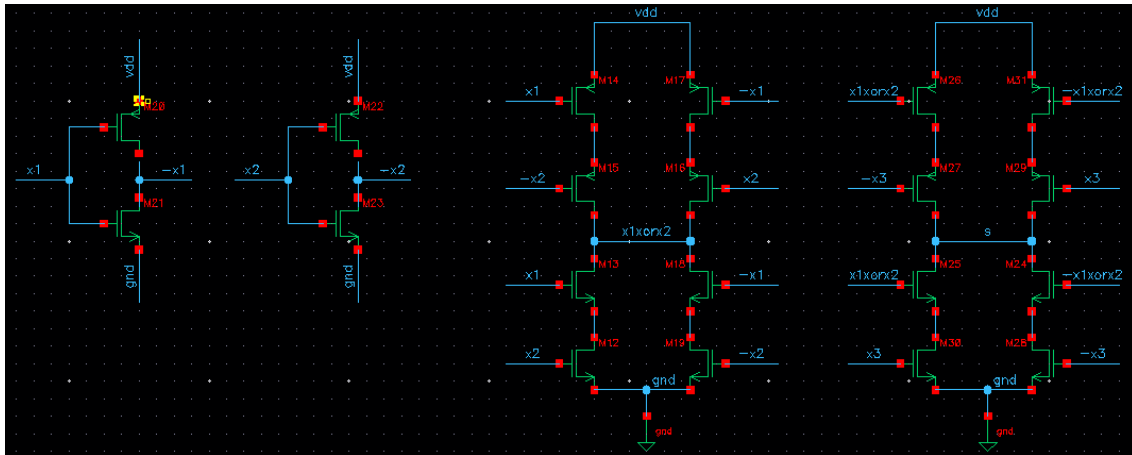
Si pensamos a la función lógica dividada en dos ramas, tendríamos la rama de la NAND con  $(\neg X1) \cdot (\neg X2)$  y la otra rama como otra NAND compuesta por  $(\neg X3) \cdot (X1 \oplus X2)$ . El máximo delay se puede conseguir cuando ambas ramas cambian a activo alto. Esto sucede por ejemplo cuando  $X1=X2=1$  y  $X3=0$ . En el caso del mínimo delay, deberíamos tener una rama en activo alto, cuya combinación puede ser  $X1=X2=0$  y  $X3=1$

**D- Utilice el circuito anterior en la implementación de un Full-Adder.**

El siguiente circuito representa el Cout de un F-A

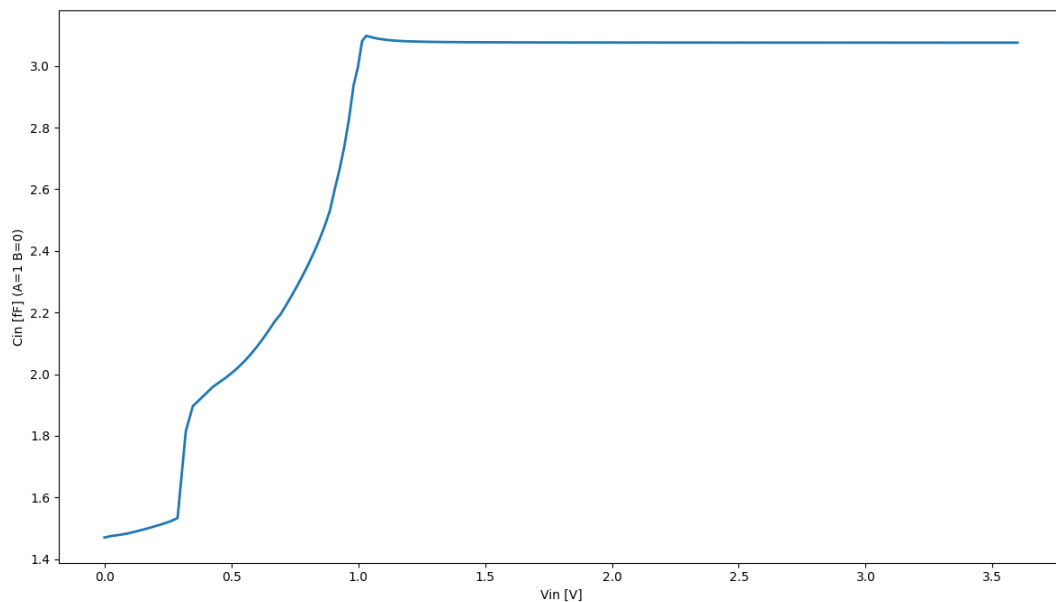


El siguiente circuito representa S de un F-A



### Ejercicio 3

A- Para una compuerta NAND simétrica de 2 entradas y tamaño mínimo, determine la capacidad de entrada mediante simulación.



Se utilizó una entrada con un generador de corriente, y la otra entrada y demás conexiones a GND. Además una relación de anchos nmos-pmos de 3 veces. En la parte constante se obtuvo una capacidad de 3.076fF.

**B- Si a la salida se la carga con una capacidad C que puede valer entre 10fF y 100fF, determine un modelo de energía consumida en cada transición considere sólo el peor caso de combinación de entradas. Considere que las entradas están alimentadas por una fuente de tensión ideal (escalón) con resistencia equivalente de 10k.**

Para la transición de las dos entradas de alto a bajo (peor caso), el modelo de energía para 10fF y 100fF por simulación es:

Energía entregada por la fuente (C\_L=10fF): 1.7152773405812274e-14

Energía almacenada en el capacitor (C\_L=10fF): 3.125e-14

Energy\_cap / Energy\_supply (C\_L=10fF): 1.8218628125414886

Energía entregada por la fuente (C\_L=100fF): 1.2316166322602024e-14

Energía almacenada en el capacitor (C\_L=100fF): 3.1250000000000003e-13

Energy\_cap / Energy\_supply (C\_L=100fF): 25.373155234719054

**Energía Consumida por Transición:** La energía consumida en cada transición se puede modelar usando la fórmula de energía para cargar y descargar una capacitancia (V es la tensión de la fuente de alimentación de la compuerta):

$$E = \frac{1}{2} C_{\text{load}} V^2$$

**Modelo de Consumo de Energía con Capacitancia de Entrada y Resistencia:** La potencia disipada durante el cambio de estado se puede calcular considerando la corriente que circula a través de los transistores y la resistencia, aunque este último término podría despreciarse.

$$E_{\text{trans}} = \frac{1}{2} C V^2 + I^2 R t_{\text{trans}}$$

#### Ejercicio 4

Para el siguiente circuito, halle los tiempos de propagación para max delay y min delay del path V2-J considerando los timing windows de las señales V1, V2, V3. Considere la resistencia de las conexiones despreciables frente a las resistencias de salida de los inversores

### **NODO V2**

$$\text{SlewIN} = 125\text{ps}$$

$$\text{CIN} = 2\text{fF}$$

$$\text{SlewOUT} = 113\text{ps}$$

$$\text{Delay} = 120.5\text{ps}$$

### **NODO F**

$$\text{Delay} = T_2 = 150\text{ps}$$

### **NODO G**

$$\text{SlewIN} = 113\text{ps}$$

$$\text{CinMAX} = 7\text{fF} = (\text{C}_9 + 2(\text{C}_5 + \text{C}_3))$$

$$\text{CinMIN} = 3\text{fF} = (\text{C}_9)$$

$$\text{SlewOUTMAX} = 114.6\text{ps}$$

$$\text{SlewOUTMIN} = 111.6\text{ps}$$

$$\text{DelayMAX} = 125.12\text{ps}$$

$$\text{DelayMIN} = 120.08\text{ps}$$

### **NODO H**

$$\text{SlewINMAX} = 114.6\text{ps}$$

$$\text{SlewINMIN} = 111.6\text{ps}$$

$$\text{CinMAX} = 7\text{fF} = (\text{C}_{10} + 2(\text{C}_4) + \text{C}_6)$$

$$\text{CinMIN} = 5\text{fF} = (\text{C}_{10} + \text{C}_6)$$

$$\text{SlewOUTMAX} = 114.92\text{ps}$$

$$\text{SlewOUTMIN} = 112.32\text{ps}$$

$$\text{DelayMAX} = 125.5\text{ps}$$

$$\text{DelayMIN} = 122.32\text{ps}$$

### **NODO I**

$$\text{SlewINMAX} = 114.92\text{ps}$$

$$\text{SlewINMIN} = 112.32\text{ps}$$



$$C_{in} = 2fF = C_{INV}$$

$$\text{SlewOUTMAX} = 110.98\text{ps}$$

$$\text{SlewOUTMIN} = 110.46\text{ps}$$

$$\text{DelayMAX} = 119.09\text{ps}$$

$$\text{DelayMIN} = 118.72\text{ps}$$

### **NODO J**

Hago la suma de todos los delays (máximos y mínimos) calculados por tabla e interpolación, y además le sumo el T2 del nodo F.

$$\text{DELAY MAXIMO} = 490.21\text{ps} + T2 = 640.21\text{ps}$$

$$\text{DELAY MINIMO} = 481.62\text{ps} + T2 = 631.62\text{ps}$$