Proyecto Final: Dispositivos Semiconductores

Jose Ricardo Soro Jara B36853
Universidad de Costa Rica, 2018

***Abstract*—Design of a Dual Edge MOS detector in which its priority is speed, second priority power consumption and in third place area. The proposed design is based on Domino Logic because of its speed.**

I. Introducción

Un circuito detector de flancos es aquel que da por salida un pulso en 1 cada vez que la entrada cambia entre 0 y 1 lógico como se muestra en la siguiente figura:

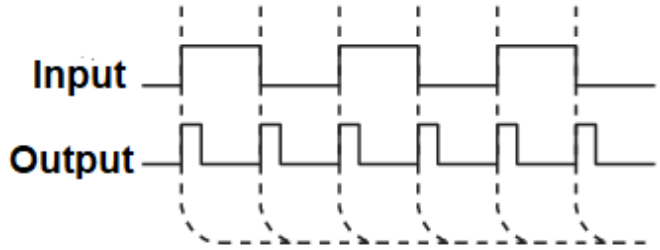

Figure 1. Señales en detector de flancos

Para este se diseño; se utilizará la teoría del esfuerzo lógico para elegir el mejor diseño posible; además de combinarlo con la teoría en familias lógicas para elegir el diseño que mejor cumpla con las condiciones dadas de prioridad; de mayor a menor prioridad:
1. Velocidad
2. Consumo de potencia
3. Área

Por lo cuál se buscará maximizar la velocidad; prescindiendo principalmente de la optimización del área y del consumo de potencia en caso de ser necesario.

II. Diseño Propuesto

La primera consideración que se tuvo a la hora de elegir un diseño, fue el de que familia lógica utilizar; entre las investigadas se encontró que la familia lógica más rápida es ECL (Emitter-Coupled Logic), no obstante, esta fue descartada ya que se busca utilizar transistores MOS, mientras ECL utiliza transistores BJT. Entonces las opciones se redujeron finalmente a lógica pseudo N-Mos o lógica Domino; entre las cuales se eligió la lógica Domino pues más adelante se demostrá que una de sus mayores falencias (lógica incompleta) no afecta al diseño propuesto pues no se necesitaron ni NANDS ni NORS para resolverlo.
Otra de las principales consideraciones fue la de reducir el esfuerzo lógico lo más posible, para esto una de las principales consideraciones fue reducir el Branching effort por ejemplo; además de que el uso de la lógica domino beneficia el calculo del esfuerzo lógico pues en esta lógica, el esfuerzo lógico de cada compuerta disminuye, como se muestra a continuación:

- Inversor (CMOS) –>1
- Inversor (CMOS acoplado a Domino) –>5/6
- NAND (CMOS 2 input) –>4/3
- NAND (Domino 2 input) –>1
- NOR (CMOS 2 input) –>5/3
- NOR (Domino 2 input) –>2/3

Gate type	Clocked evaluation transistor?	Formula	$n=2$	$n=3$	$n=4$
inverter	yes	2/3			
	no	1/3			
NAND	yes	$(n+1)/3$	1	4/3	5/3
	no	$n/3$	2/3	1	4/3
NOR	yes	2/3	2/3	2/3	2/3
	no	1/3	1/3	1/3	1/3
multiplexer	yes	1	1	1	1
	no	2/3	2/3	2/3	2/3

Figure 2. Valores de esfuerzo lógico según Harris

Por lo cuál la logica Domino fue elegida como base de este diseño.

Se tomó la decisión de utilizar circuitos simples para realizar este diseño que encuentren por aparte el rising edge y el falling edge; y una vez encontrados se suman utilizando un OR; a continuación se muestra la forma de encontrar tanto el flanco positivo como el flanco negativo:

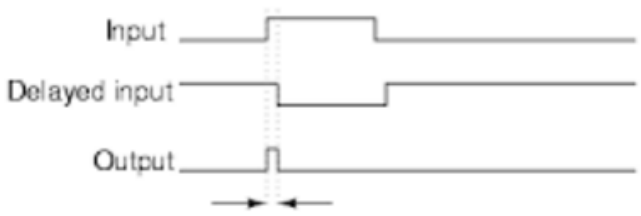

Figure 3. Detección de flanco positivo

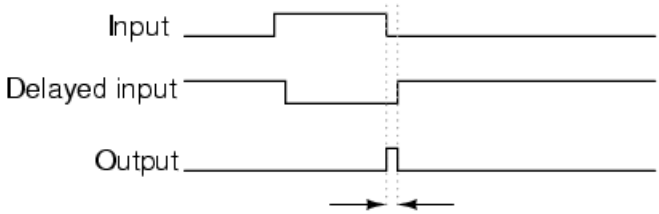

Figure 4. Detección de flanco negativo

Para alcanzar esto; reduciendo al mínimo el branching effort; que según un calculo preliminar podría aumentar el esfuerzo lógico multiplicándolo por 4; se decidió separar el calculo de flancos en dos circuitos completamente aparte (evitando así a toda costa el branching de señales) y teniendo un impacto positivo en el calculo del retraso del circuito, afectando solo el área (se duplica el principio del circuito al duplicar los inversores) pero afectando de manera poco percibible la potencia pues los inversores extra cumplen con la lógica CMOS (famosos por su bajo consumo de potencia), se utilizan CMOS en los inversores porque debido a la monoticidad es imposible encadenar inversores Dinamicos; además para retrasar la señal lo suficiente resulta de mayor utilidad el retraso brindado por la lógica CMOS. Dejando esto claro se muestra entonces el circuito detector de flancos propuesto:

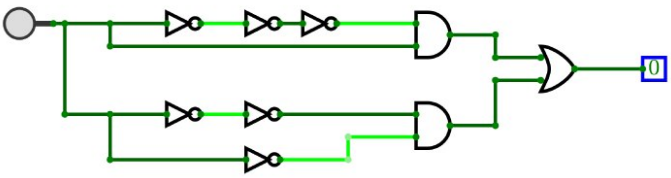

Figure 5. Diagrama de compuertas

Donde se puede notar como se hace un branching de la señal de entrada; para no afectar de esta forma el esfuerzo lógico de ningún camino (marcados con A y B las secciones duplicadas para reducir el branching). Además se nota que esta parte encargada del retardo se desarrolla en CMOS mientras las comparaciones lógicas AND se desarrollan en Domino al igual que la suma de señales. Además se nota en rojo la salida de cada señal. Acto seguido se muestra como teóricamente estos circuitos generan las señales Rising Edge y Falling Edge y como estas no interfieren entre ellas:

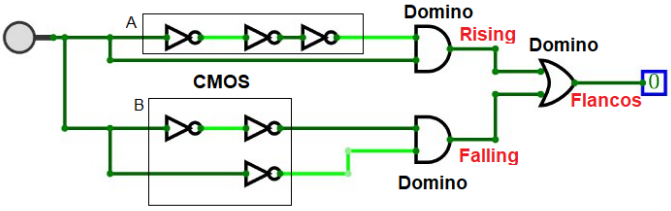

Figure 6. Señales y familias lógicas en diseño propuesto

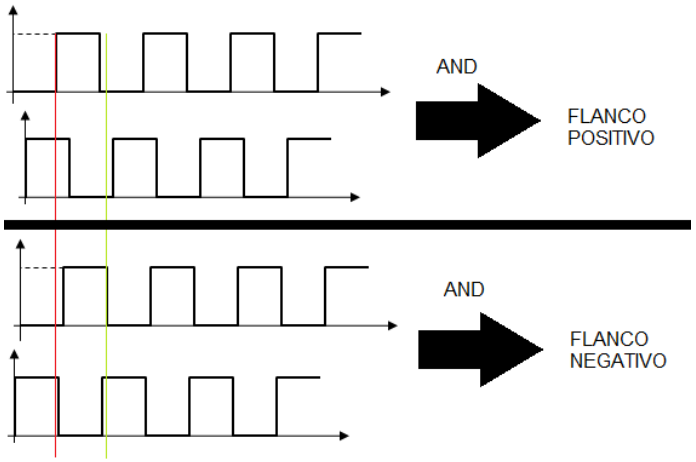

Figure 7. Calculo de Flancos

Donde se muestra con un eje rojo como se calcula en la parte superior el flanco positivo con las señales correctamente retardadas generan un flanco y como este calza con un 0 lógico en la parte inferior. De manera análoga se muestra con un eje verde en la parte inferior el calculo de un flanco negativo y como este calza tambien con un 0 lógico en la parte superior.

Se adjunta además un esquematico a nivel de transistores (el cual se incluye a parte para facilidad de análisis)

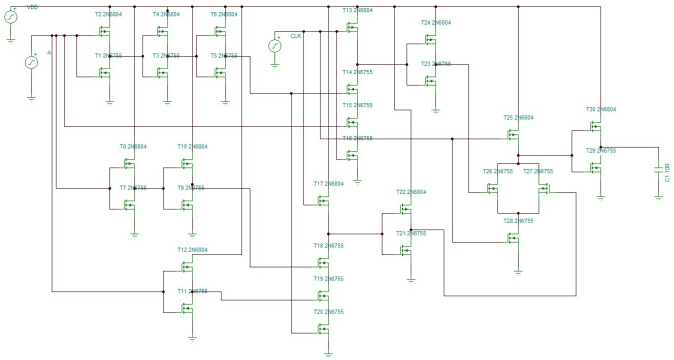

Figure 8. Diagrama a nivel de transistores

Además se adjunta el esquemático de transistores señalizado con rojo las partes implementadas en CMOS (inversores) junto a la señalización en verde de las partes implementadas con lógica domino.

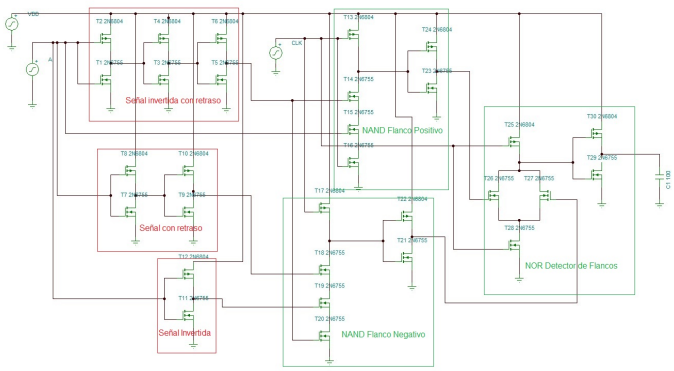

Figure 9. Diagrama a nivel de transistores con designaciones

Entonces se decide a dimensionar el siguiente circuito con los siguientes valores de esfuerzo lógico por celda (extraídos del libro Harris):

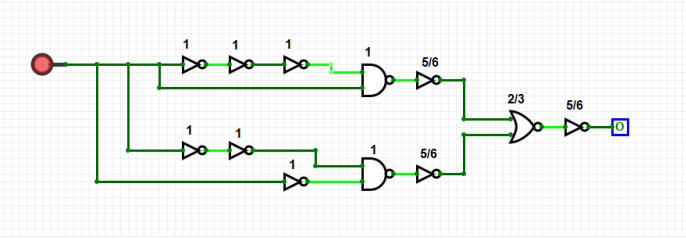

Figure 10. Esfuerzo lógico por celda

Por lo cual se empieza por dimensionar (por teoría de esfuerzo lógico) el camino más largo el cual está marcado en la siguiente figura:

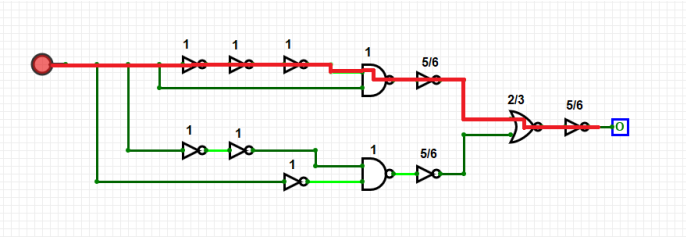

Figure 11. Camino más largo a dimensionar

Donde se sabe que la capacitancia de salida es 100 veces la de entrada (por el enunciado) por tanto el valor de

$$H = Cout/Cin = (100 * Cin)/Cin = 100 \quad (1)$$

Además que el valor B de branching es igual a 1 pues se diseño de esta manera. Por otro lado el valor de G es igual al valor de los g de cada compuerta del camino marcado en rojo donde G=1*1*1*1*(5/6)*(2/3)*(5/6)=0.462962. Por último entonces se despeja F de

$$F = G * B * H = 0.462962 * 1 * 100 = 46.2962 \quad (2)$$

Por su parte se debe calcular el N, que está dado por:

$$N = \log_4 F = \log_4 46.2962 = 2.76 \simeq 3 \quad (3)$$

Donde se redondea a 3 para aprovechar la curva de retardo vs numero de etapas donde tiene un menor efecto sobre el retardo tomar un número de etapas mayor al calculado que un número menor de etapas como se muestra en la siguiente curva de ejemplo:

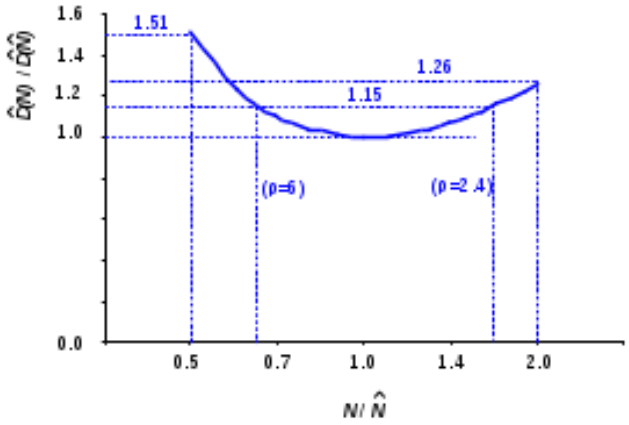

Figure 12. Ejemplo de curva de retardo vs número de Etapas

Se pasa entonces a calcular el mejor esfuerzo que es dado por

$$f = F^{1/N} = 46.2962^{1/3} = 3.60 \quad (4)$$

A partir de este dado se puede dimensionar la capacitancia de cada compuerta de forma que

$$Cin = Cout * g/f \quad (5)$$

Fórmula que se aplica de atrás para adelante para encontrar el valor de cada capacitancia:

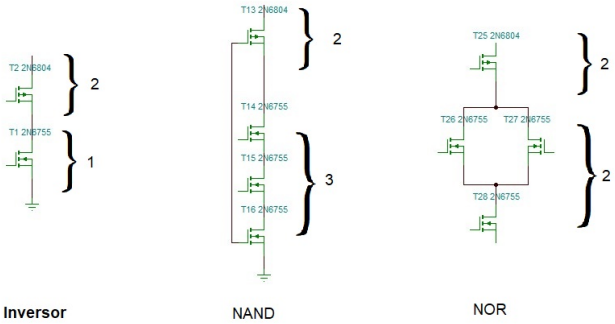

Figure 13. Dimensionamiento de cada compuerta

por lo cual en cada compuerta; habrá una razon 2:1 (pmos:nmos) por cada valor en la capacitancia; en el caso de la NAND hay una razon de 2:3 y en la NOR hay una razon de 1:1. Por ejemplo en el caso de tener una capacitancia de carga de 4.20mF; las capacitancias de atrás para adelante estarían dadas por:

g	Cout	Cout (Cin=4.20mF)	Razon P	Razon N	P	N
Load	100	0,42				
0,83333333	23,1481481	0,097222222	2	1	0,06481481	0,03240741
0,66666667	5,35836763	0,022505144	1	1	0,01125257	0,01125257
0,83333333	0,9922903	0,004167619	2	1	0,00277841	0,00138921
1	0,22969683	0,000964727	2	3	0,00038589	0,00057884
1	0,06380467	0,00026798	2	1	0,00017865	8,9327E-05
1	0,01772352	7,44388E-05	2	1	4,9626E-05	2,4813E-05
1	0,0049232	0,0042	2	1	0,0028	0,0014

Figure 14. Capacitancias de ejemplo

Dado que el valor de F se mantiene en las demás branches del circuito; es justo decir que el dimensionamiento de los demás inversores es igual al del branch principal que tenga la misma profundidad que ellos. De modo que el dimensionamiento quedaría de la siguiente forma:

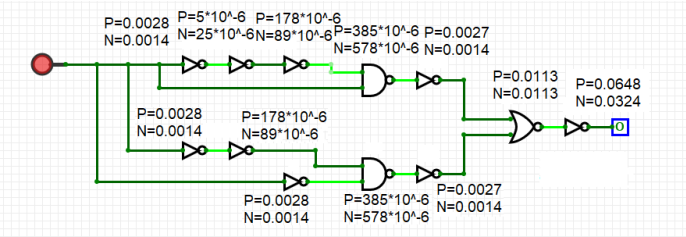

Figure 15. Capacitancias de ejemplo

III. Tiempos

El calculo del tiempo de retardo está dado por

$$D = \sum di = Df + P \tag{6}$$

donde Df = F = 46.2692 y P por su parte:

$$P = \sum p1 = 1 + 1 + 1 + 2 + 1 + 2 + 1 = 9 \tag{7}$$

por lo cual:

$$D = (46.2692 + 9)\tau = 55.2692\tau \tag{8}$$

Donde se calcula frecuencia de conmutación como la máxima frecuencia a la que el circuito puede funcionar; esto se da cuando el retardo de la señal no es mayor que el periodo de la señal de entrada; por ende la frecuencia de conmutación viene dada por

$$f_{conmutacion} = 1/Dmin = 0.018/\tau \tag{9}$$

IV. Simulación en Spice

Para comprobar los resultados y el funcionamiento del circuito; se hizo una simulación programada en Spice y ejecutada en el simulador de consola NgSpice se obtuvo los siguientes resultados ante una entrada:

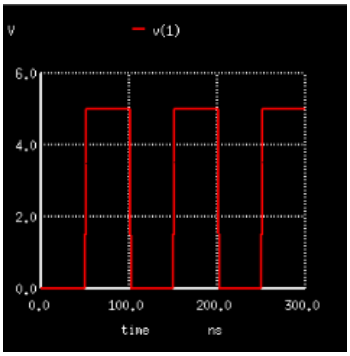

Figure 16. Entrada del circuito

En la cual se detectan tanto los flancos positivos como negativos como se muestra a continuación:

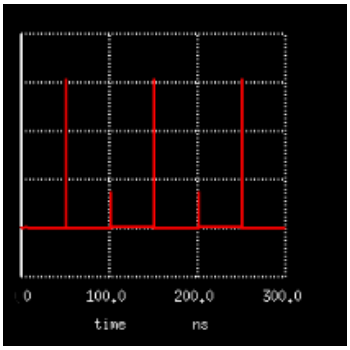

Figure 17. Detección de Flanco Positivo

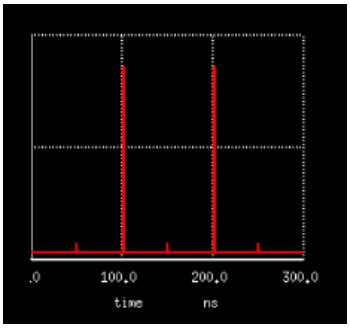

Figure 18. Detección de Flanco Negativo

Y a continuación se muestra la detección de ambos flancos en forma comparativa con la entrada:

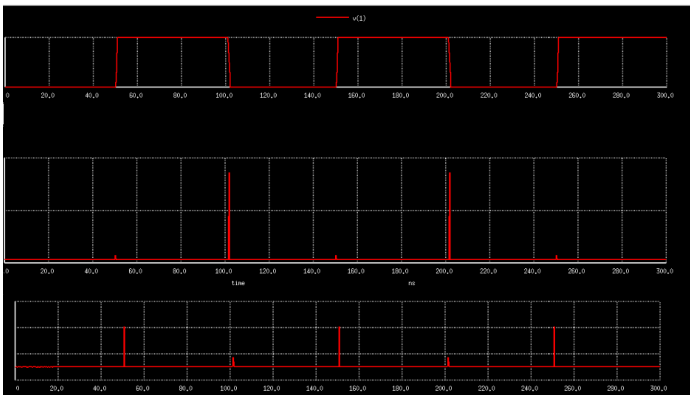

Figure 19. Detección de Flancos vs Entrada

Estas señales se suman en la compuerta OR la cual da una salida:

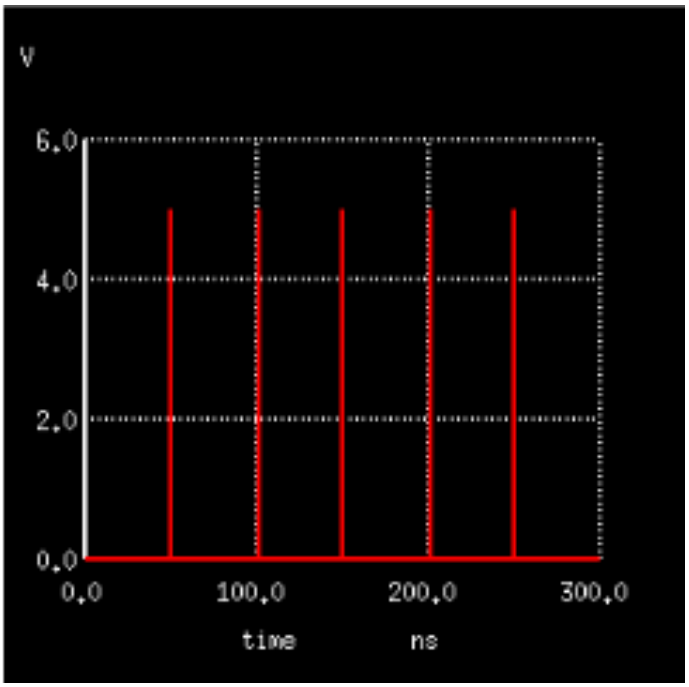

Figure 20. Suma de flancos

Terminando por mostrar la comparación entre la entrada y la salida del circuito detector de flancos:

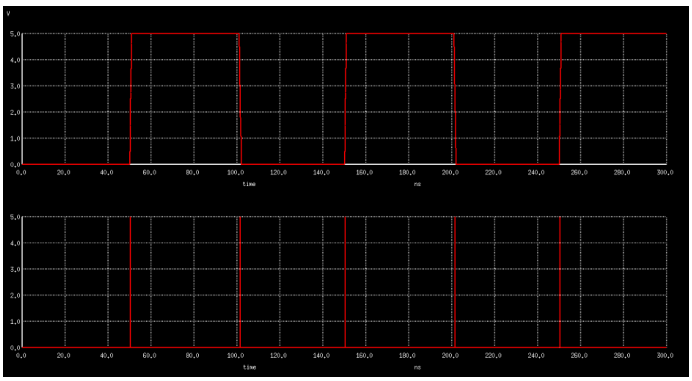

Figure 21. Entrada vs Salida

V. Conclusión

Se mostró como se puede utilizar tanto la teoría de esfuerzo lógico junto a la teoría de familias lógicas para conseguir un circuito detector de flancos rápido; especialmente si se compara con un circuito CMOS por ejemplo; además como se aprovechó una desventaja de la lógica domino (carencia de señales negadas) a favor del circuito al utilizar solo compuertas AND y OR; mientras se dejó los inversores en CMOS para evitar problemas de monotonicidad y utilizar el retardo que proveen estas compuertas a favor del circuito (haciéndolo un poco más robusto ante entradas CLK muy grandes)