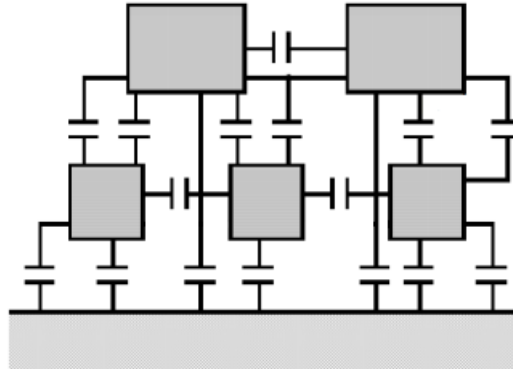


Ruido

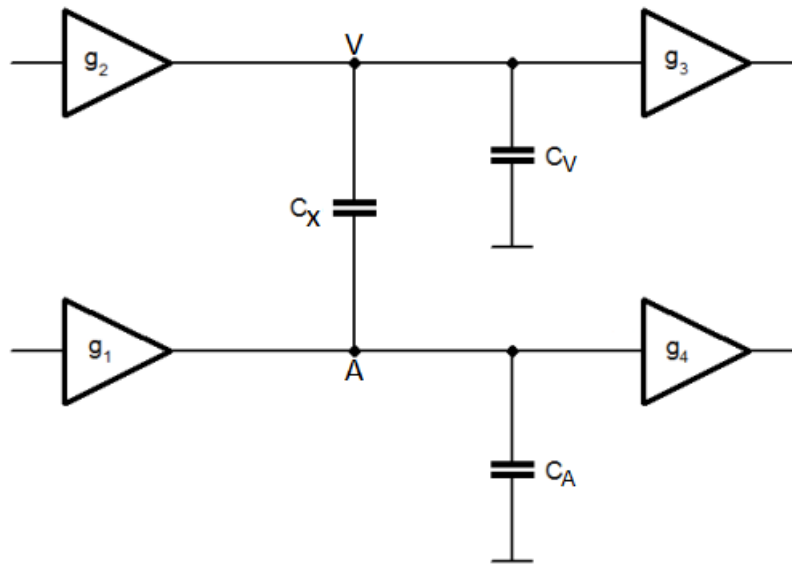
21 de octubre de 2024

1. Introducción

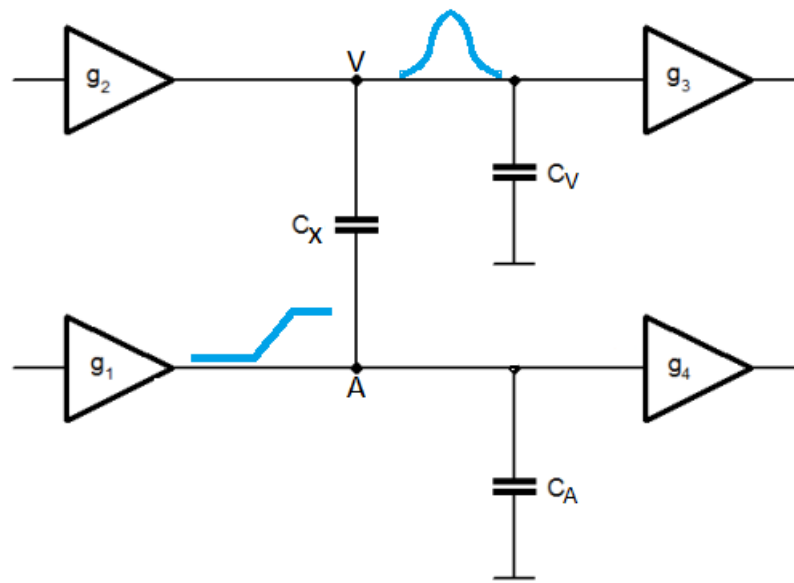
En la clase de interconexiones, se estudió el efecto que las mismas tienen a la propagación de las señales digitales asumiendo que todas las capacidades parásitas existían entre un conductor y GND. Sin embargo, ese caso no es muy realista, más aún para los conductores de las capas superiores para los cuales el sustrato queda en general muy lejos ya además apantallado por los conductores de las capas inferiores.



Por lo tanto, la geometría de los conductores y la proximidad entre los mismos produce un acoplamiento capacitivo. Consideremos un modelo simplificado para iniciar el análisis de la situación. El nodo A se llamará *agresor* y el nodo V se llamará *víctima*. C_A es la capacidad total del conexionado contra GND y de carga del nodo A, mientras que C_V es la capacidad total del conexionado contra GND y de carga del nodo V. C_X es la capacidad de *crosstalk* entre ambos nodos.



No se puede aplicar el análisis de árbol RC porque obviamente no cumple con las hipótesis de dicho modelo. Iniciemos el análisis entonces suponiendo que los conductores no poseen resistencia. Supongamos que el nodo V está a un valor constante 0-lógico (GND) a través de la resistencia de salida de la compuerta g_2 y que el nodo A transiciona de 0-lógico (GN) a 1-lógico (VDD). Entonces, dependiendo de la relación de capacidades, se verá un pulso espúreo (que llamaremos *glitch*) en el nodo V .



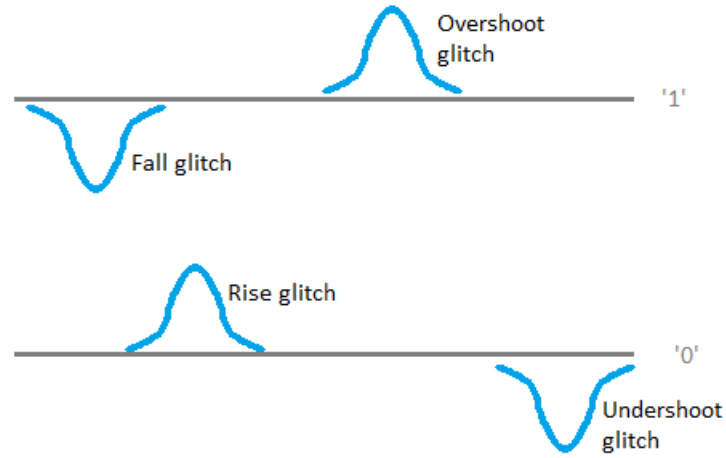
La magnitud del glitch dependerá de los siguientes factores:

- Capacidad de acople C_X : cuanto mayor sea esta capacidad, mayor será la magnitud del glitch.
- Slew del agresor: cuanto más rápida sea la subida o bajada del agresor, mayor será la magnitud del glitch.
- Capacidad a GND de la víctima C_V : Cuanto mayor sea esta capacidad, menor será la magnitud del glitch.
- Driving de la víctima R_{ON_V} : Cuanto mayor sea esta resistencia, mayor será la magnitud del glitch.

Sabemos que para un período de tiempo lo suficientemente grande, el nodo víctima volverá a su estado original. Sin embargo, un mal funcionamiento del circuito puede darse si:

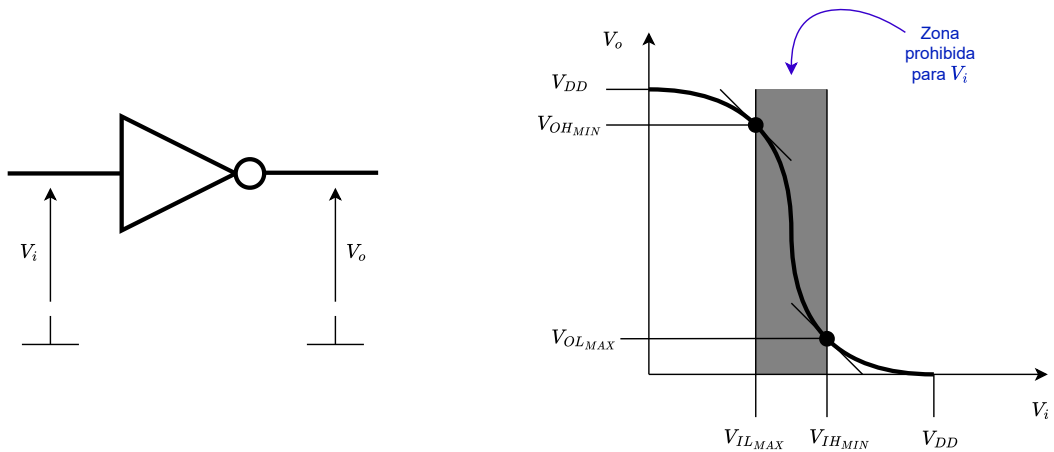
- El glitch se produce sobre una línea de clock o reset y tiene una magnitud y/o duración lo suficientemente grande como para afectar el estado de algún flip-flop.
- El glitch se propaga por la lógica combinacional hasta alcanzar el pin D de un flip-flop en el instante en que aparece el flanco activo de clock y modifica el estado de dicho flip-flop.

2. Tipos de glitches



3. Propagación de un glitch a través de una celda lógica

Consideremos el caso de un inversor:

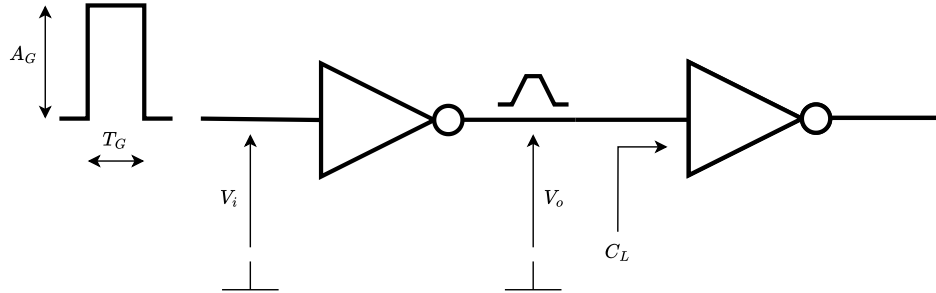


Primer criterio: Si la magnitud del glitch, hace que V_i no ingrese en la zona prohibida, entonces el glitch no se propaga a la salida de la celda.

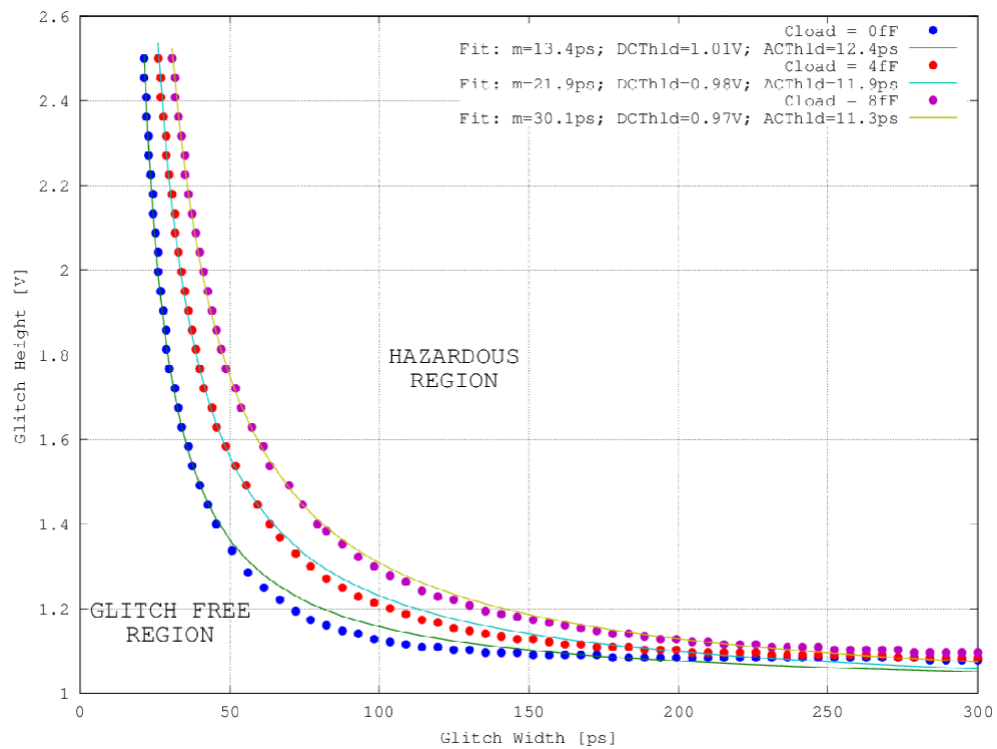
Se define:

$$DC_threshold = \min\{V_{IL_{max}}, V_{DD} - V_{IH_{min}}\}$$

Se observa además que no todos los glitches que ingresan a la zona prohibida (amplitud mayor al DC_threshold) generan una variación a la salida de la celda tal que ingresa en la zona prohibida de la celda siguiente. Para ello, estos deben tener una duración mayor al “retardo inercial” de la celda que es función de la amplitud del glitch y de la capacidad de carga.



A continuación se muestra un conjunto de simulaciones SPICE en función de la amplitud del glitch, la duración del mismo y la capacidad de carga para un glitch rectangular:



Observemos que a medida que aumenta la capacidad de carga, la zona peligrosa se desplaza hacia arriba y el circuito se vuelve más robusto a ruido. Observemos también

que en general, las curvas son aproximables por hipérbolas donde la asíntota horizontal es el `DC.threshold` y la asíntota vertical se denomina `AC.threshold`:

$$A_G = \frac{m}{T_G - AC_threshold} + DC_threshold$$

donde m es una constante que determina la concavidad de la hipérbola.

Criterio general (que incluye al primer criterio): Decimos que un glitch se propaga a través de una celda lógica si la tensión de salida de la misma entra en la zona prohibida de la celda siguiente.

Un enfoque más realista podría ser modelar el glitch por un trapecio de diferentes rise/fall times y diferentes anchos/altos en lugar de un rectángulo. De cualquier forma, un conjunto de simulaciones SPICE será utilizadas para determinar las tablas para los distintos valores de capacidad de salida, tipo de celda, rise/fall times, amplitud y duración de los glitches.

4. Glitch analysis

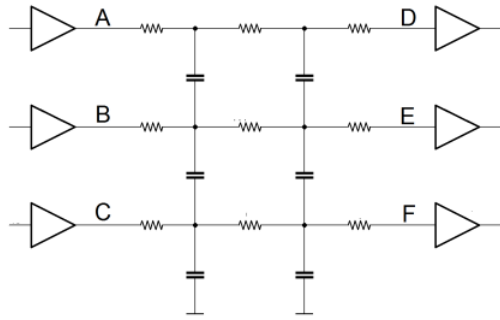
Este procedimiento se utiliza para determinar si los glitches generados por crosstalk afectan el estado de un circuito.

1. Determinar el tipo de glitch en la entrada de la celda. Los overshoot y undershoot glitches no pueden cambiar el estado de un circuito.
2. Determinar la capacidad de carga de la celda y así obtener los valores de `DC.threshold` y `AC.threshold`.
3. Determinar mediante la tabla, si el glitch entra en la zona prohibida de la compuerta siguiente.

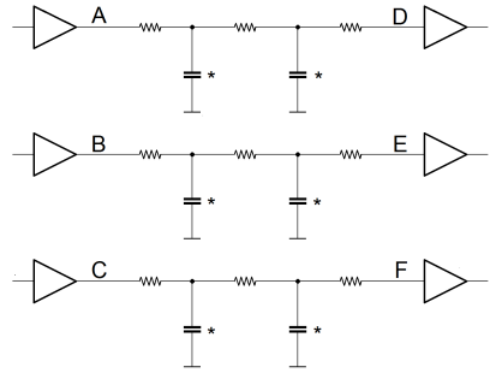
A lo largo de un camino R2R, este procedimiento se continua celda tras celda hasta recorrer todo el path y llegar al pin D del flip-flop destino. Si el instante de arribo al pin D de dicho flip-flop es muestreado por el clock del sistema, entonces se dice que el circuito es afectado por el crosstalk.

5. Delay analysis

Ya se ha comentado que un circuito con acoplamiento capacitivo no puede ser modelado como un árbol RC al no cumplir con las hipótesis del modelo. Lo que se hará en su defecto es desacoplar el circuito de forma tal que sí puede ser analizado:

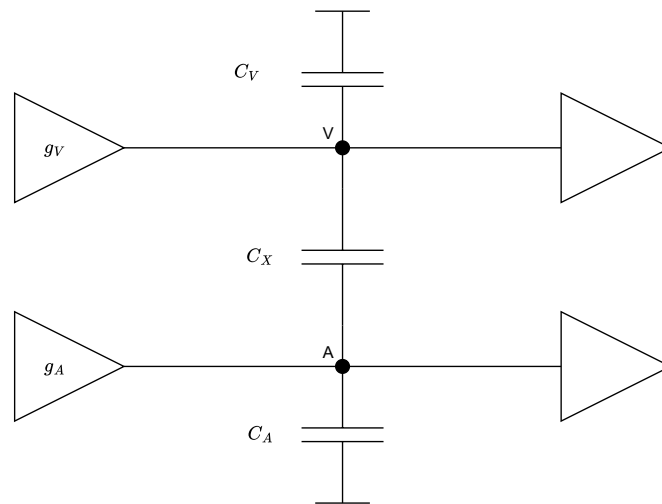


Circuito capacitivamente acoplado.



Circuito capacitivamente desacoplado.

Analicemos el circuito siguiente.

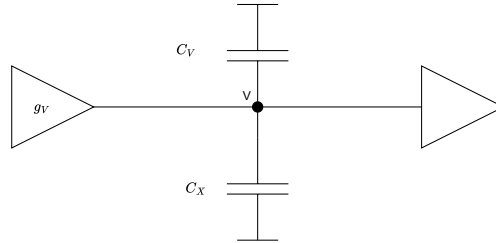


Se definen 8 casos de interacción entre el agresor y la víctima:

| Caso | V_A | V_V |
|------|--------------------|--------------------|
| 1 | $V_A = 0$ (GND) | $V_V = \uparrow$ |
| 2 | $V_A = 0$ (GND) | $V_V = \downarrow$ |
| 3 | $V_A = 1$ (VDD) | $V_V = \uparrow$ |
| 4 | $V_A = 1$ (VDD) | $V_V = \downarrow$ |
| 5 | $V_A = \uparrow$ | $V_V = \uparrow$ |
| 6 | $V_A = \downarrow$ | $V_V = \downarrow$ |
| 7 | $V_A = \uparrow$ | $V_V = \downarrow$ |
| 7 | $V_A = \downarrow$ | $V_V = \uparrow$ |

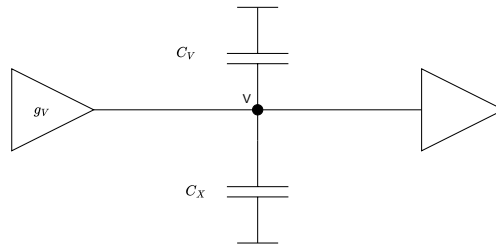
Caso 1: Agresor estable $V_A = 0$ (GND) - Víctima transiciona de 0 a 1 $V_V = \uparrow$

En este caso, la celda g_V debe suministrar una carga total $Q = V_{DD}(C_V + C_X)$. Por lo tanto, la capacidad de carga vista por g_V es $C_{LV} = C_V + C_X$. Se podría pensar en el siguiente circuito equivalente desacoplado:

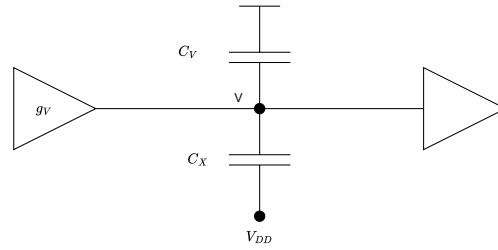


Caso 2: Agresor estable $V_A = 0$ (GND) - Víctima transiciona de 1 a 0 $V_V = \downarrow$

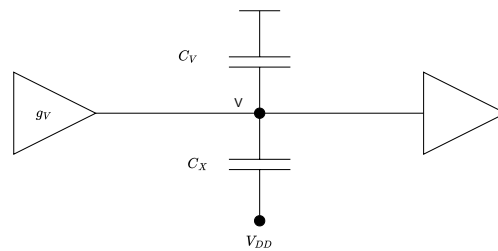
Similar al caso anterior, la celda g_V observa una capacidad de carga $C_{LV} = C_V + C_X$.



Caso 3: Agresor estable $V_A = 1$ (VDD) - Víctima transiciona de 0 a 1 $V_V = \uparrow$



Caso 4: Agresor estable $V_A = 1$ (VDD) - Víctima transiciona de 1 a 0 $V_V = \downarrow$



Caso 5: Agresor V_A y Víctima V_V transicionan ambos de 0 a 1 $V_A = \uparrow$, $V_V = \uparrow$

Hay tres subcasos:

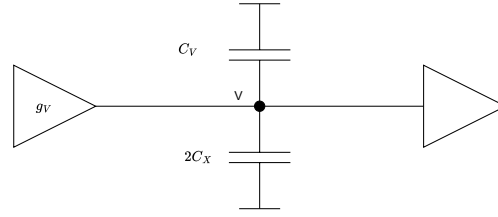
- Si ambos nodos transicionan con el mismo slew (misma pendiente) no hay variación de tensión sobre el capacitor C_X y por lo tanto la celda g_V sólo observa una carga C_V ya que debe proveer una carga $Q = V_{DD}C_V$.
- Si el slew del agresor es más rápida que la de la víctima la carga suministrada por g_V puede ser menor que $V_{DD}C_V$ ya que el agresor puede proveer carga para C_V . Este caso se denomina "negative crosstalk".
- Si el slew del agresor es más lento que el de la víctima, la carga suministrada por g_V puede ser mayor que $V_{DD}C_V$ y menor que $(C_V + C_X)V_{DD}$.

Caso 6: Agresor V_A y Víctima V_V transicionan ambos de 1 a 0 $V_A = \downarrow$, $V_V = \downarrow$

Similar al caso anterior.

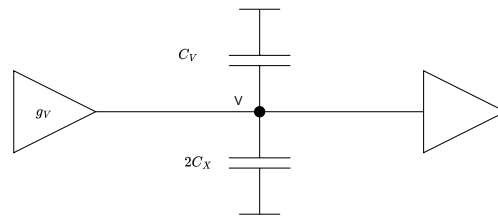
Caso 7: Agresor V_A transiciona de 0 a 1 y Víctima V_V transiciona de 1 a 0 en sentido opuesto $V_A = \uparrow$, $V_V = \downarrow$

Si el slew del agresor y la víctima son exactamente iguales pero de sentido opuesto, entonces la tensión en el capacitor C_X varía desde $+V_{DD}$ a $-V_{DD}$. Por lo tanto g_V suministra una carga total $Q = V_{DD} + (C_V + 2C_X)V_{DD}$.



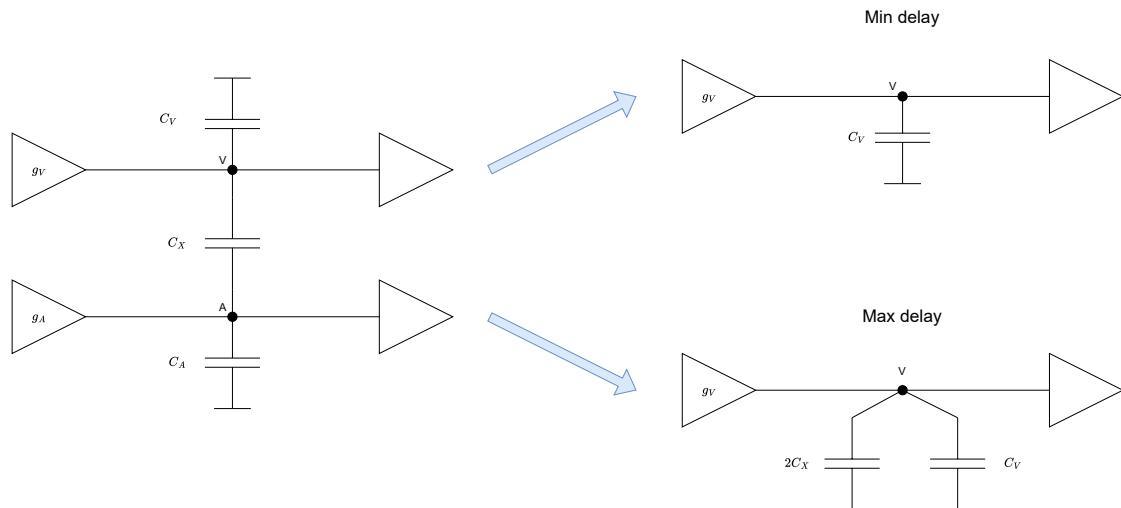
Caso 8: Agresor V_A transiciona de 1 a 0 y Víctima V_V transiciona de 0 a 1 en sentido opuesto $V_A = \downarrow$, $V_V = \uparrow$

Similar al caso anterior.

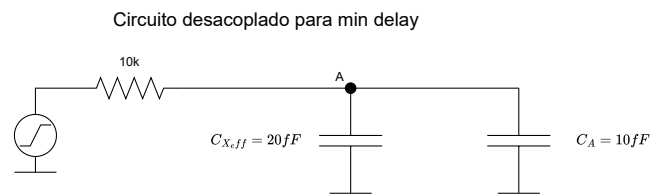
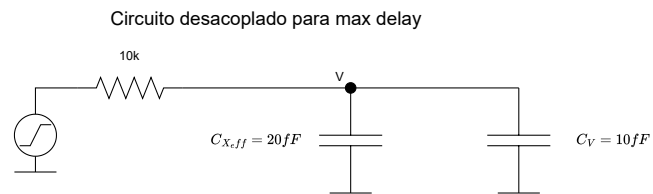
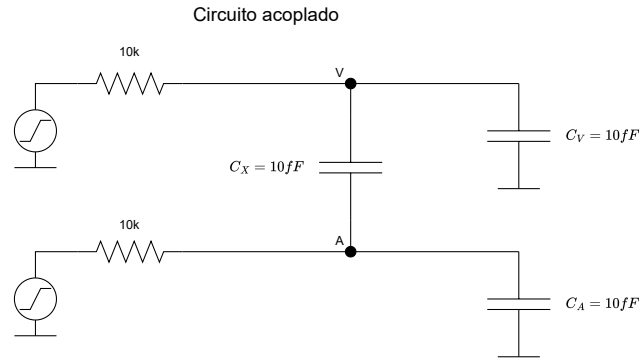


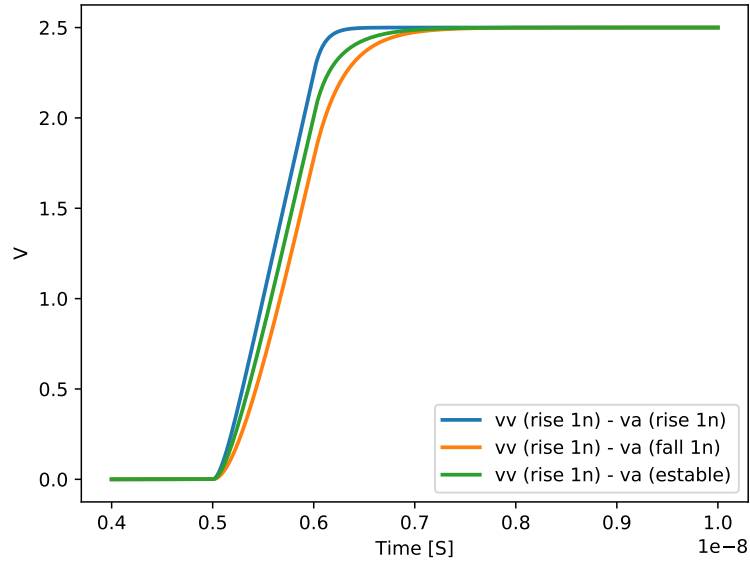
Observación:

Se puede entonces desacoplar el circuito para dos casos extremos de mayor y menor retardo tal como se observa a continuación y aplicar el método de árboles RC al circuito desacoplado.

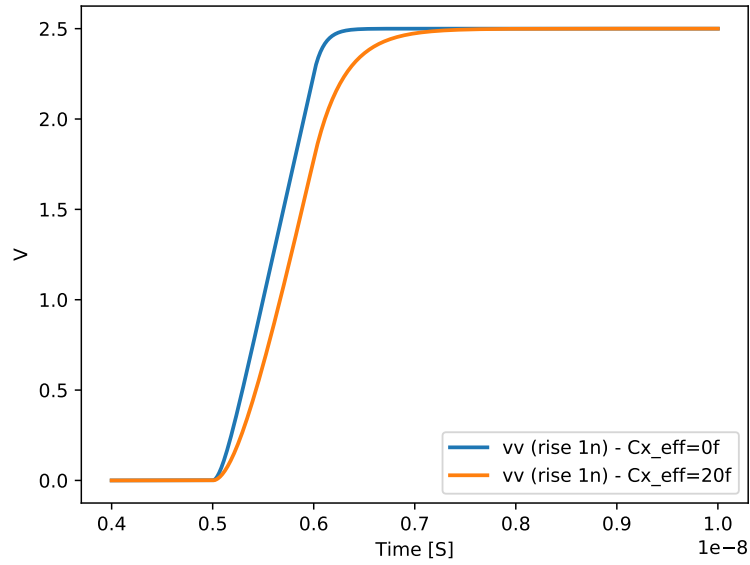


Ejemplo: Simulación SPICE para Slew=1ns, $C_X = 10fF$.





Circuito acoplado.



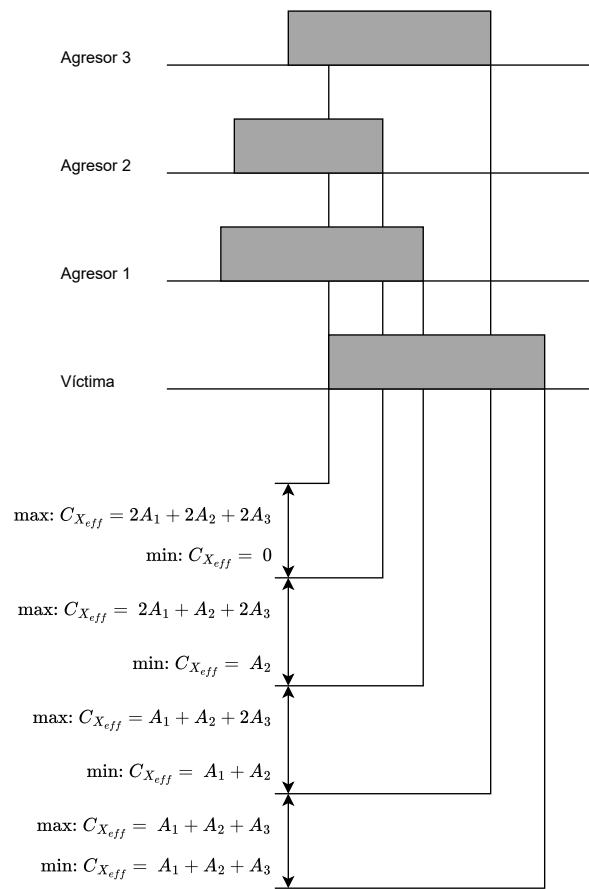
Circuito desacoplado para max y min.

Observación: Si el agresor se mantiene estable durante la transición de la víctima, entonces el circuito se desacopla refiriendo la capacidad C_X a GND.

6. Timing windows

Cuando sobre una víctima afectan uno o más agresores, se deben considerar en el momento de transición de la víctima cuáles de los agresores también transicionan para desacoplar para max/min delay las capacidades de crosstalk. Todo agresor que no transiciona solapadamente con la víctima es desacoplado con un factor de desacople de 1.

Debido a la incertidumbre PVT de la víctima y de los agregresores, se define como *timing windows* a los instantes temprano y tardío en que dichas señales pueden transicionar.



Ejemplo de cómo desacoplar las capacidades para el caso de 3 agresores sobre una víctima.

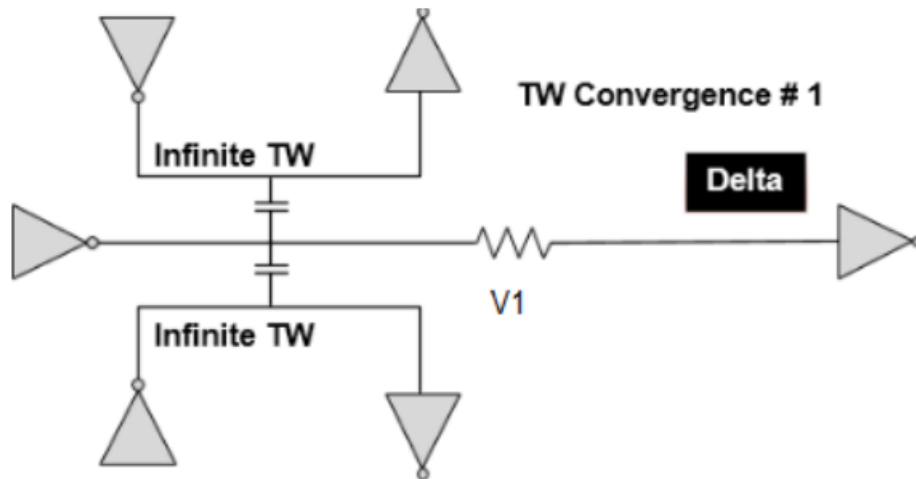
Observar que fuera del timing window de la víctima, sólo interesa hacer glitch análisis y

estimar si el glitch puede propagarse.

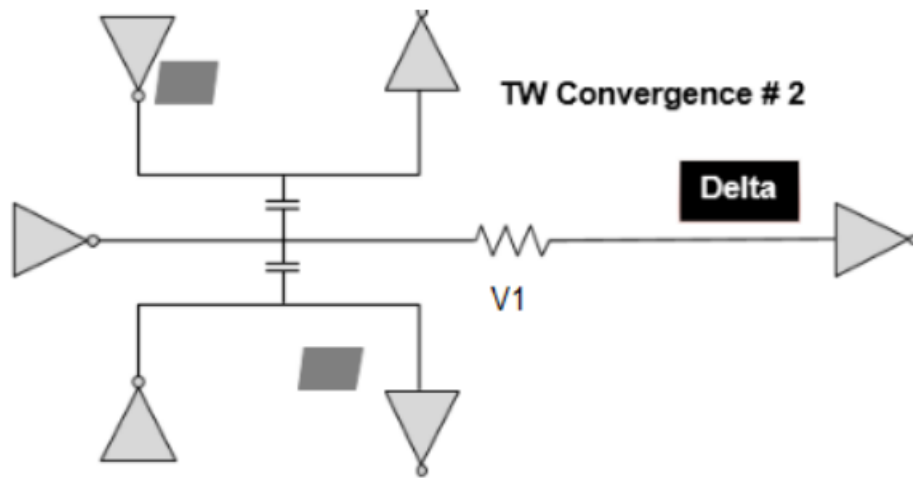
6.1. Timing windows iterations

Dados que los timing windows afectan el crosstalk delay y el crosstalk delay afectan los timing windows, entonces las herramientas de timing analysis requieren el uso de procedimientos iterativos.

Hay dos enfoques: comenzar por un timing window de duración infinita y con eso reducir paso a paso los timing windows o comenzar con un timing windows no afectado por el crosstalk y luego modificar dichos timing windows iterativamente. El número de iteraciones puede ser fijo o puede que la iteración continúe hasta que se alcance convergencia.



Inicio de la iteración con TW de infinita duración.



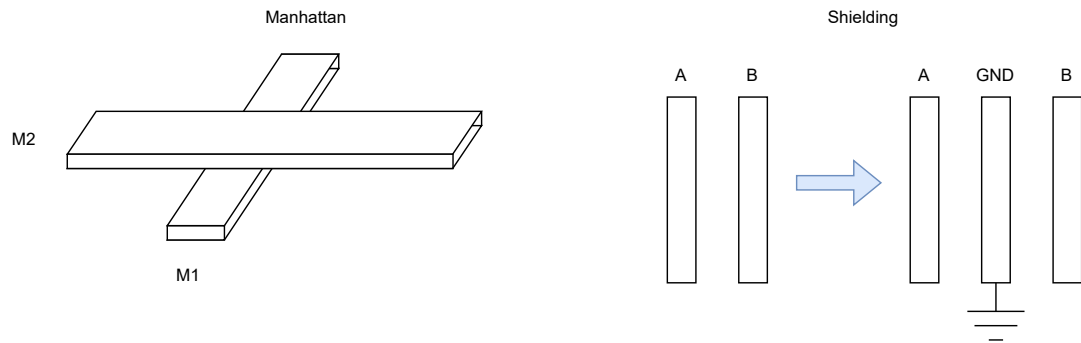
TW segunda iteración.

7. Reduciendo la complejidad computacional

- Si la capacidad de acoplamiento es menor a cierto valor se la considera despreciable.
- Si la capacidad de acoplamiento dividida la capacidad contra GND es menor a cierto valor se la considera despreciable.

8. Técnicas de layout para reducir crosstalk

- Metales de capas consecutivas se las rutea de forma ortogonal (ruteo Manhattan).
- Conductores paralelos:
 - Se aumenta la separación entre ellos.
 - Se aumenta la capacidad relativa a GND intercalando un conductor a GND entre ellos (shielding).



TW segunda iteración.

9. Preguntas para pensar

- 1- Cómo afecta el glitch noise a la funcionalidad del circuito? Influye T_{clk} ?
- 2- Cómo influye el crosstalk delay a la funcionalidad del circuito? Influye T_{clk} ?