MEMORIAS P2 Y P3

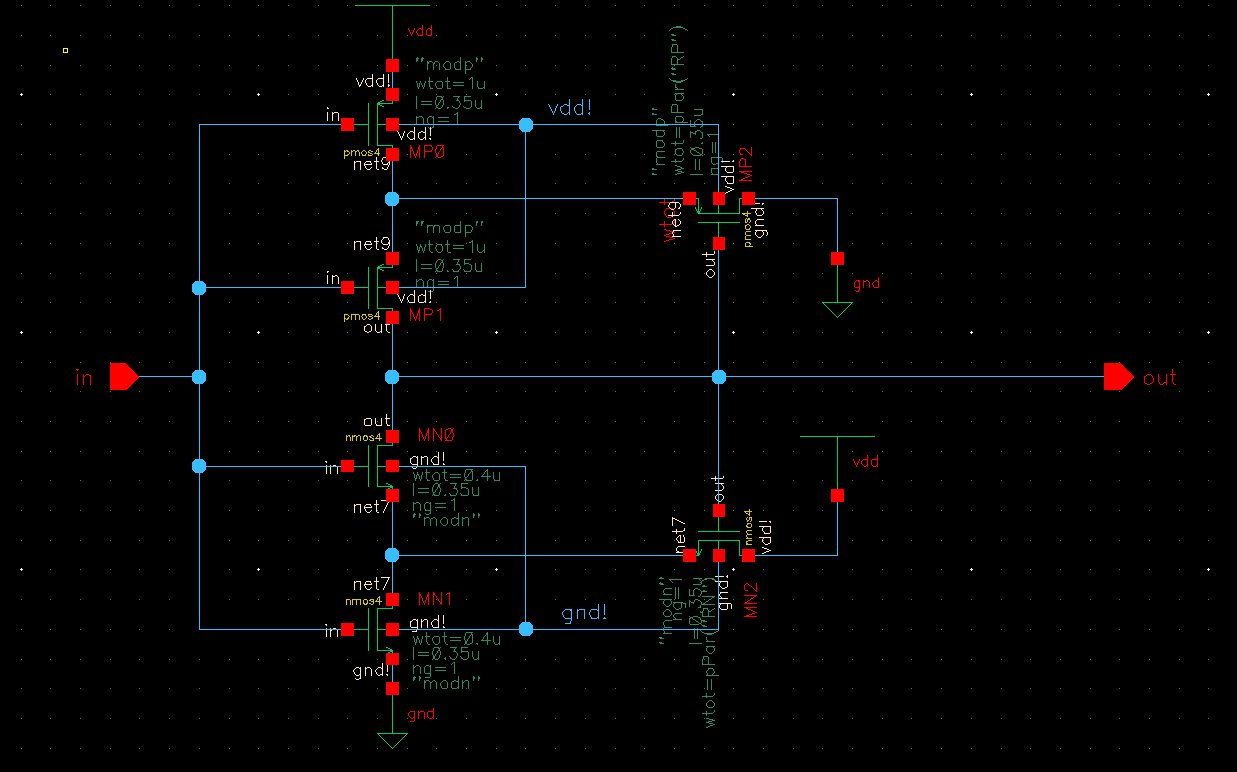
**Jaime Lloret Cuñat**

**Adam Cecetka Ortiz**

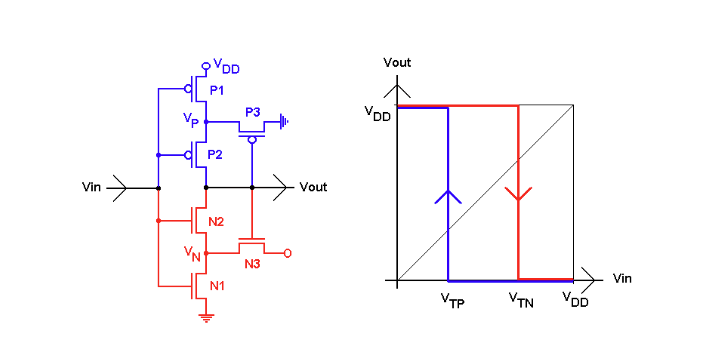
**PRACTICA 2:**

TRIGGER SCHMITT

El circuito inversor *Trigger Schmitt* es un tipo de circuito electrónico diseñado para convertir una señal analógica en una señal digital. Nosotros hemos implementado este circuito en *Cadence* (figura 1), con tres transistores *pmos* y otros tres transistores *nmos*.

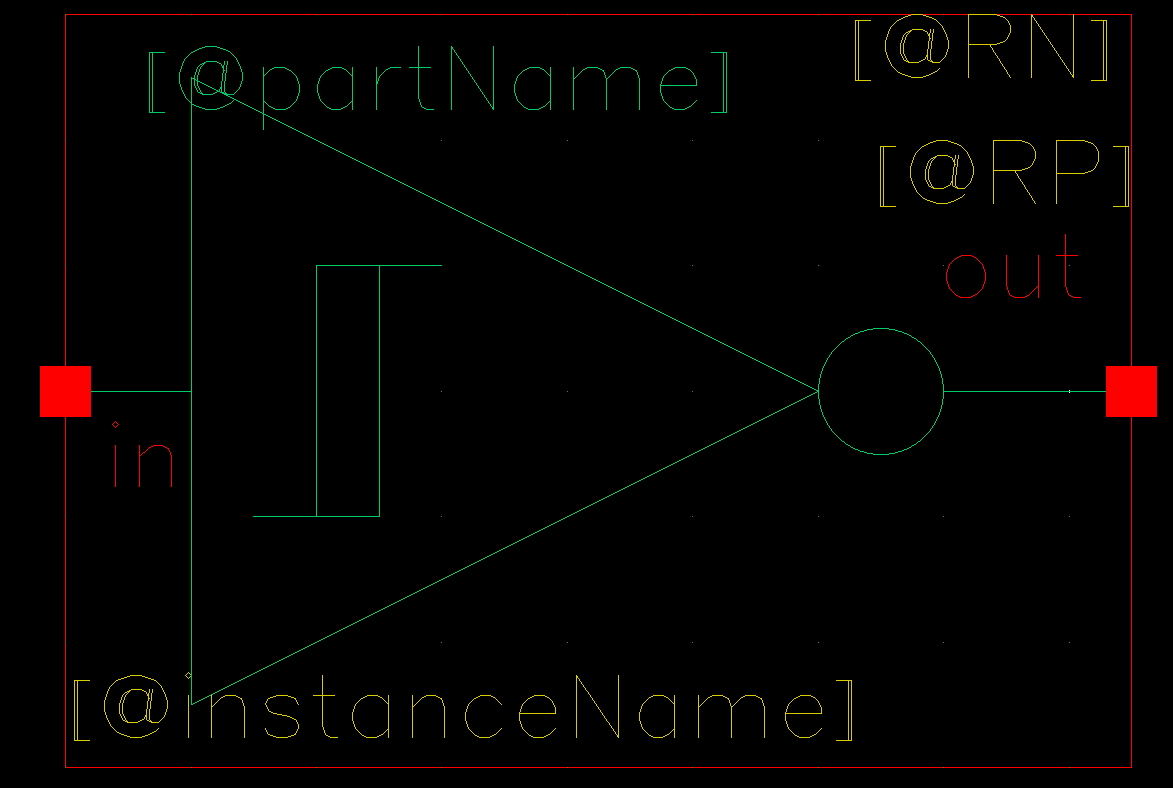


***Figura 1: Esquemático del inversor Trigger Schmitt***



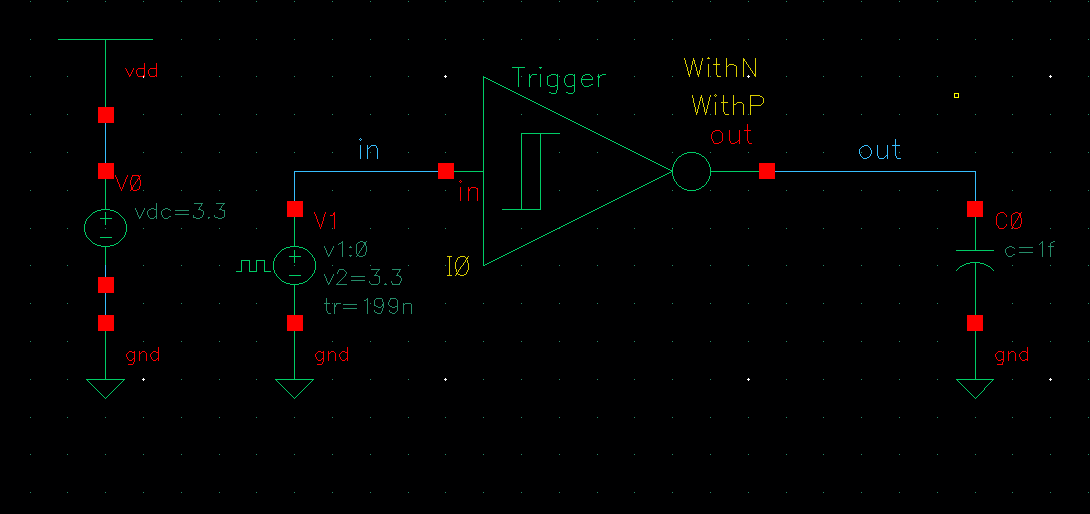
***Figura 2: Dimensionado de un Trigger Schmitt inversor***

A continuación, hicimos el símbolo con las instancias de RP y RM (figura 2).



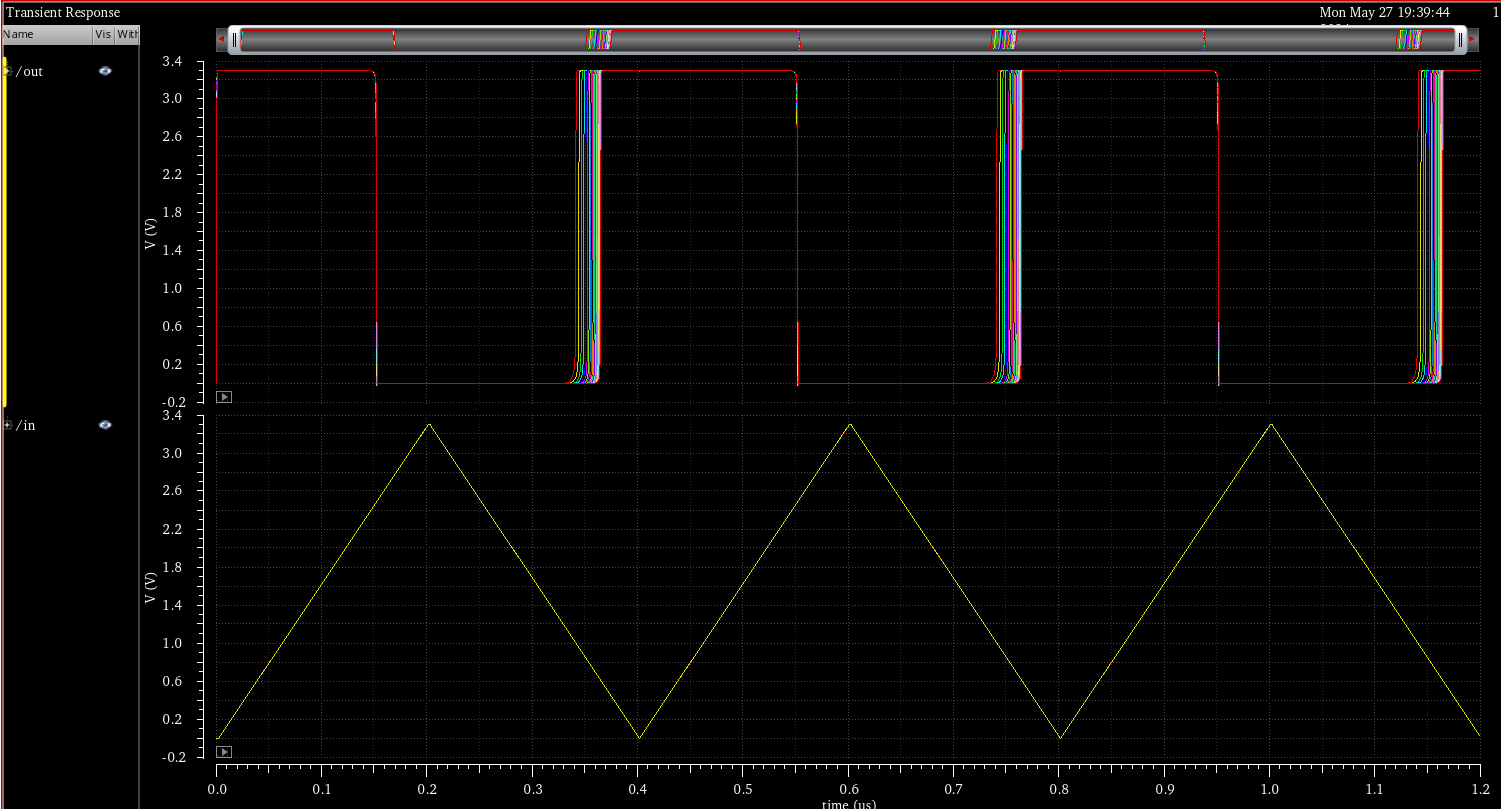
***Figura 3: Símbolo de Trigger Schmitt***

Una vez acabado el símbolo, lo instanciamos en un nuevo diseño llamado *Trigger\_TB*, en el que hicimos un *testbench* para comprobar el correcto funcionamiento del diseño creado. (figura 4)



***Figura 4: Esquemático Trigger\_TB***

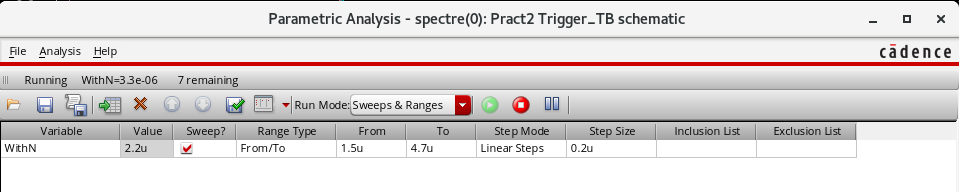
Gracias a este pudimos hacer la siguiente simulación. (figura 5)



***Figura 5: Simulación de Trigger Schmitt***

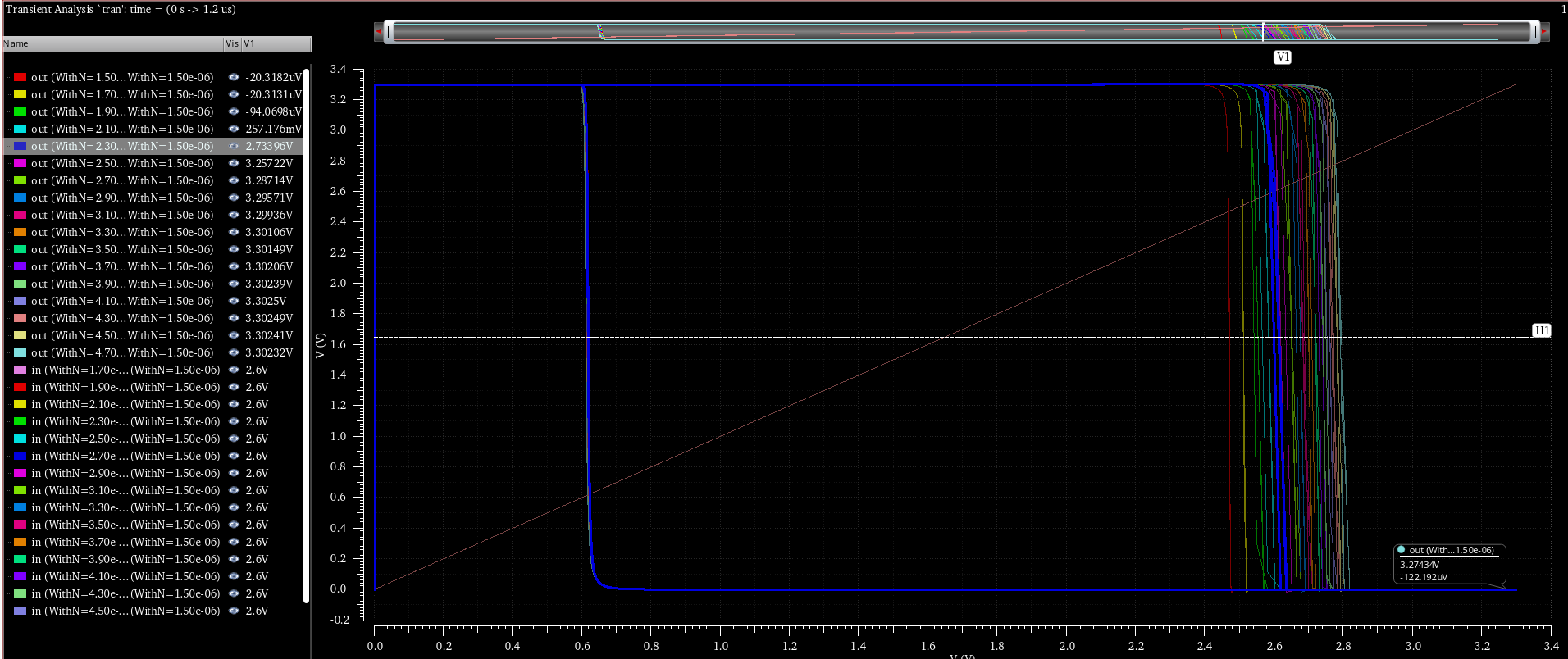
Gracias a esta, con la configuración de ejes de entrada de Y vs Y, pudimos extraer los valores óptimos de *WithN* (figura 6), y de WithP (figura 7).

Para obtener el valor de WithN hemos generado numerosas señales implementadas en un rango de 1.5u a 4.7u con una diferencia entre señal de 0.2u.

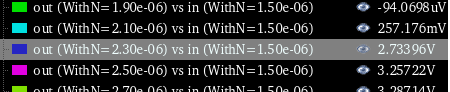


***Figura 6: Configuración del testbench***

Luego hemos generado la gráfica Y vs Y de las señales de entrada. (figura 7) Hemos traqueado la caída a 3 dB con un marcador horizontal situado en 1’65V, y otro marcador a 2.6V (2.6V porque es aproximadamente un 80% de 3.3V). De aquí se obtiene el valor óptimo que en nuestro caso *WithN* vale 2.3e-06. (figura 8)

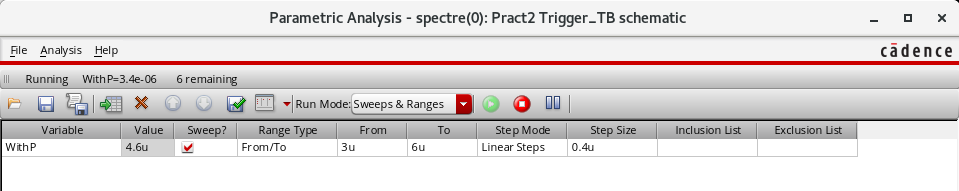
******

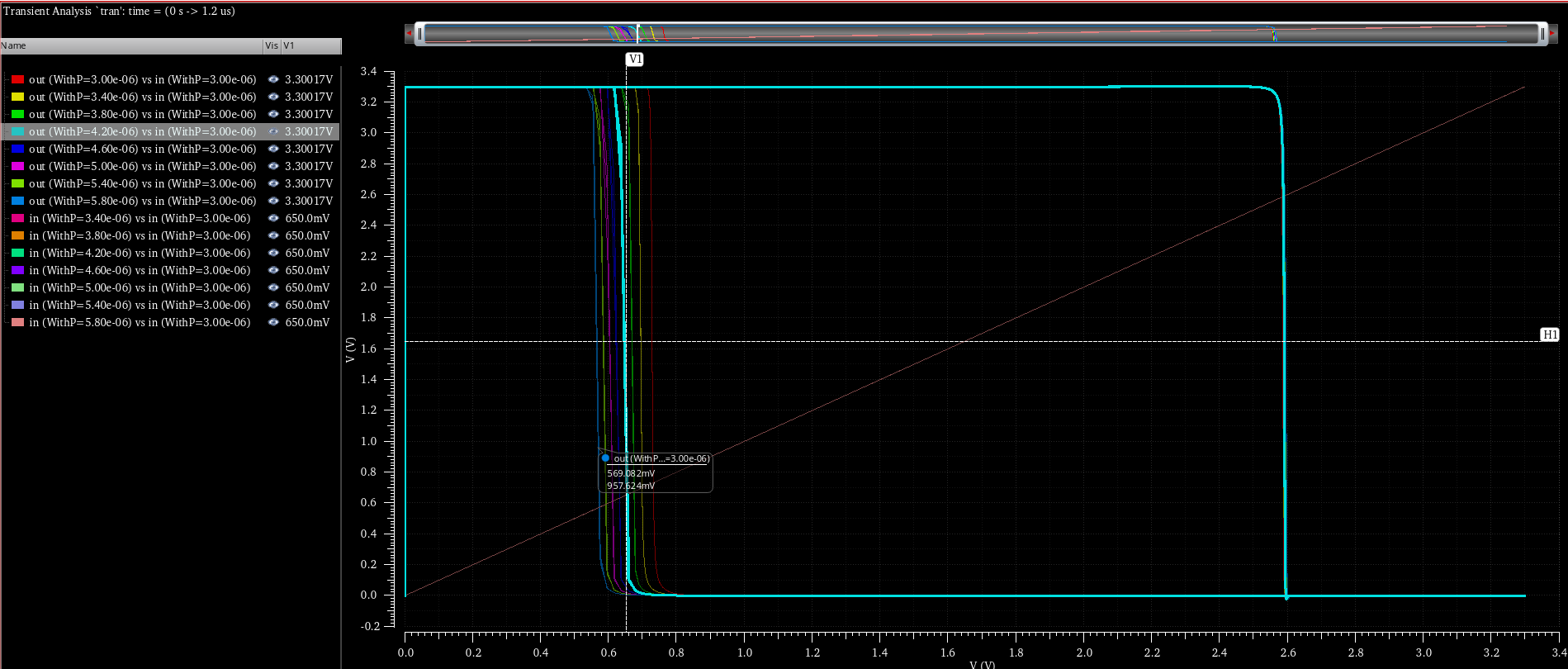
***Figura 7: Simulación Y vs Y del WithN***

******

***Figura 8: Valor óptimo de WithN (2.3e-06)***

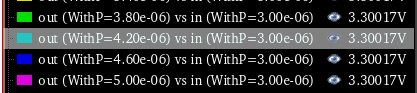
Posteriormente, hicimos algo parecido para el *WithP*, únicamente cambia que esta vez, el valor que se tenía que proyectar era 0.65, ya que es un 20% de 3.3V. (figura 9)

******

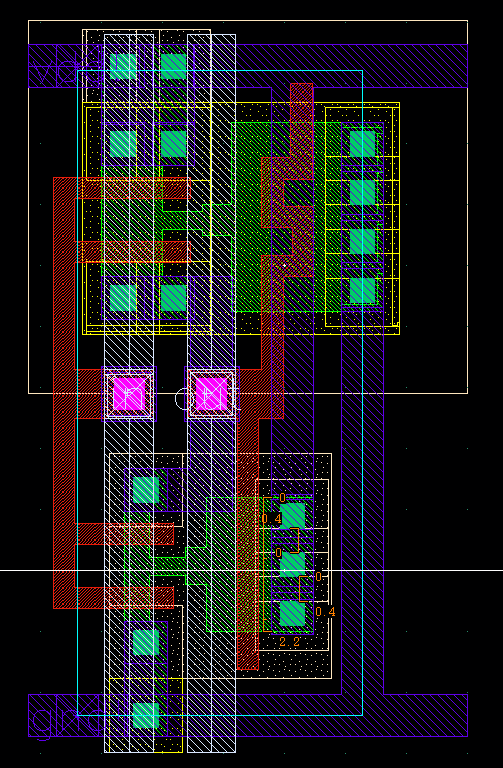


***Figura 9: Simulación Y vs Y del WithP***

Y de aquí se ve claramente que el valor óptimo de WithP es 4.2e-06

******

***Figura 10: Valor óptimo de WithN (4.2e-06)***

Paralelamente diseñamos el *layout*. (figura 11)

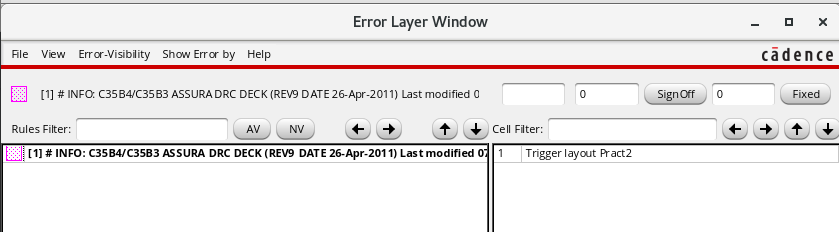
El *layout* está compuesto por 3 transistores *pmos* situados en la parte superior, y tres transistores *nmos* situados en la parte inferior.

El polisilicio del *PPLUS* está en forma de serpiente para cumplir con el ancho de transistor, que tiene que ser igual a 4.6u.

Las conexiones de *In* y *Out* están estratégicamente ubicadas en el centro del diseño para optimizar el espacio y para que se reduzca lo máximo posible el tiempo de retardo.

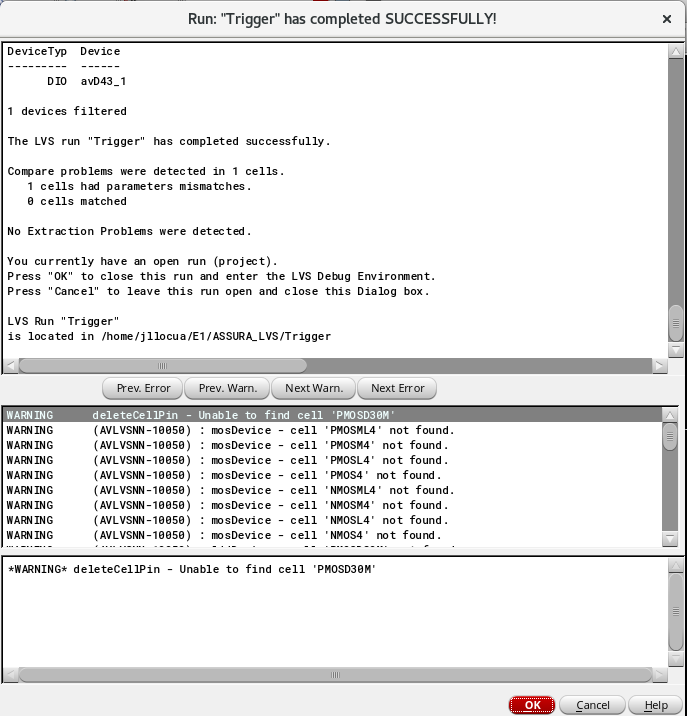
***Figura 11: Layout de Trigger Schmitt***

Una vez terminado el *layout* comprobamos que no se estuviera incumpliendo ninguna regla de diseño con un Run Asura DRC. (figura 12)



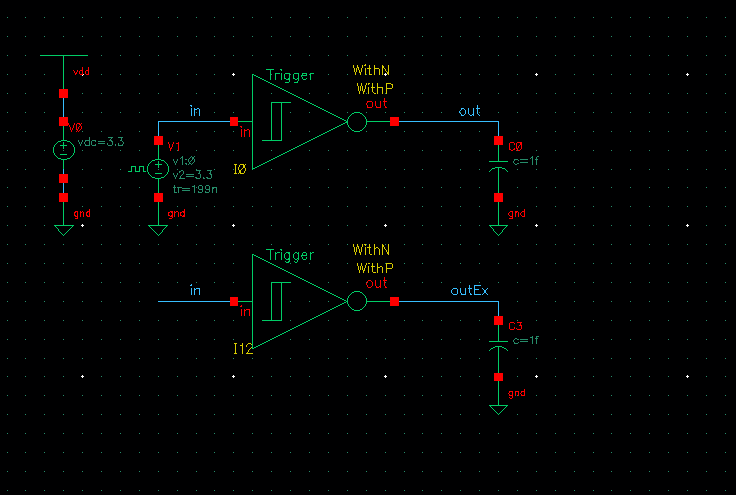
***Figura 12: Run Asura DRC del Trigger Schmitt***

A su vez, también nos aseguramos de que no hubiera ningún problema con las conexiones del *layout* al esquemático con un Run Asura LVS. (figura 13)



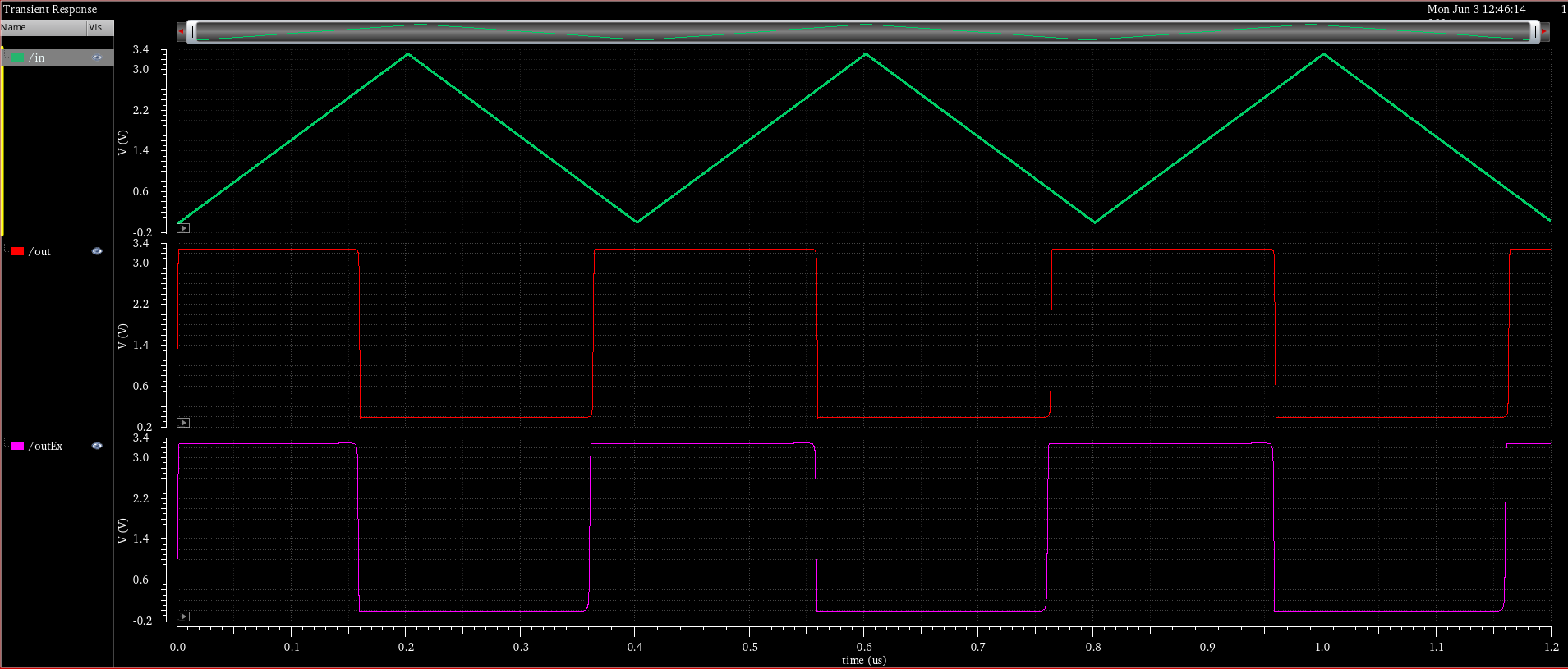
***Figura 13: Run Asura LVS del layout***

Para asegurarnos de que los valores del esquemático coincidan con los del *av\_exctracted* hicimos un testbench con las instancias de ambos módulos, y conectándolos a la misma señal de entrada. Los valores de *WithN* y *WithP* también son los mismos. (figura 14).



***Figura 14: Esquemático del testbench dual***

Ejecutamos la simulación y se ve claramente que las señales del esquemático del diseño y del *av\_exctracted* coinciden.

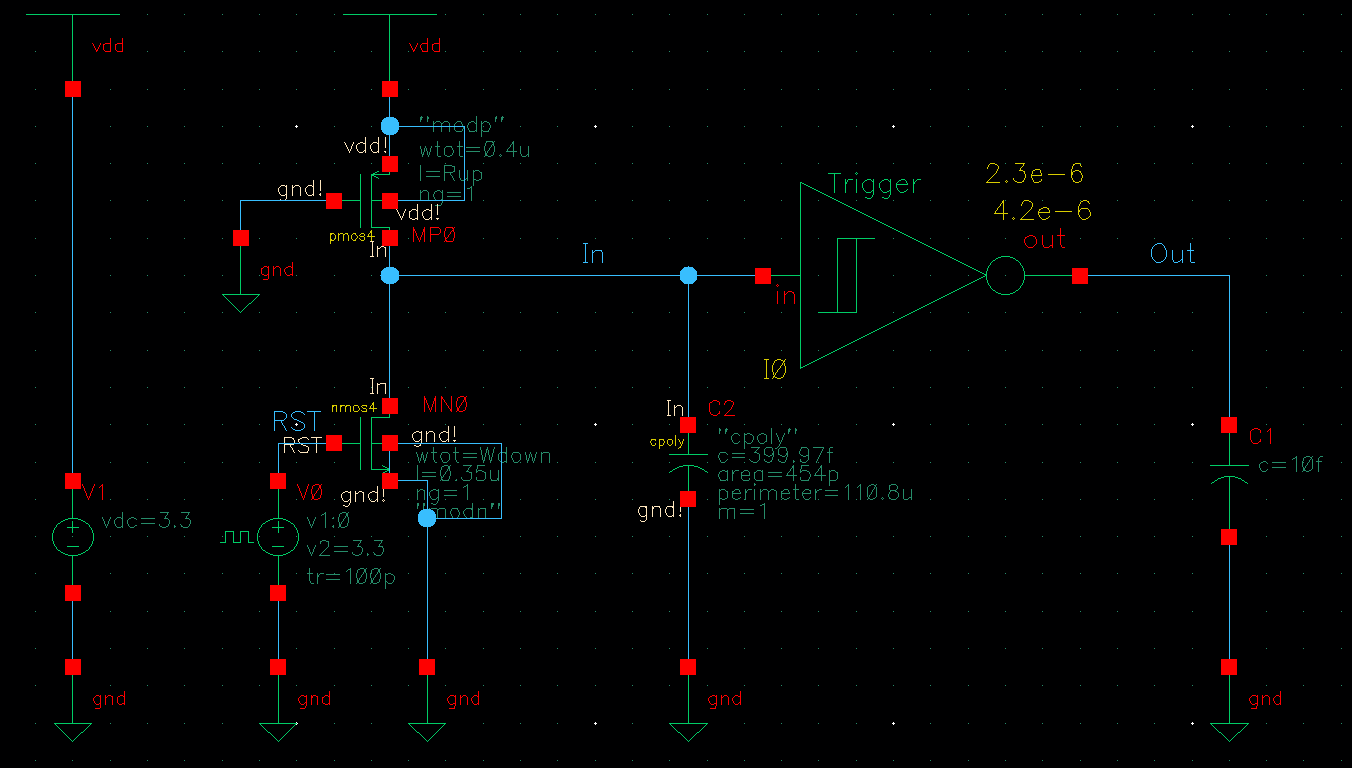


***Figura 15: Resultados del testbench***

GENERADOR DE PULSOS

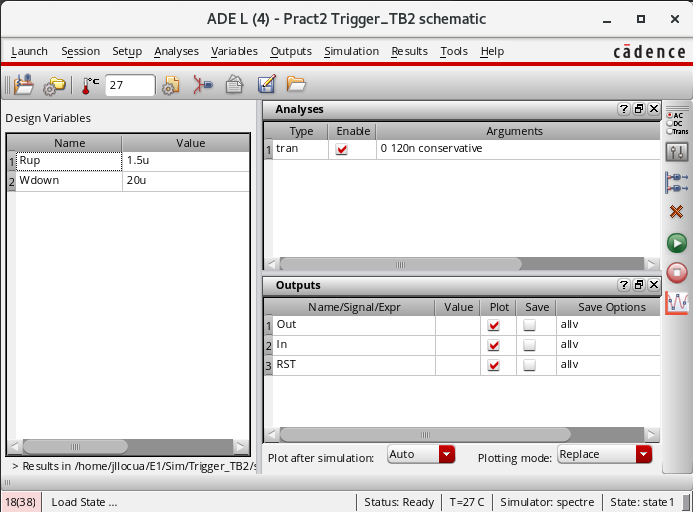
Por último, se nos pedía utilizar los diseños empleados anteriormente para la aplicación de un generador de pulsos basado en el *Trigger Schmitt*.

Este es el diseño que se nos proponía realizar: (figura 16)



***Figura 16: Esquemático del generador de pulsos***

Posteriormente configuramos las variables de *Rup* con 1.5u y *Wdown* con 20u. (figura 17)



***Figura 17: Configuración de la simulación del generador de pulsos***

Y estos son los resultados obtenidos de las señales *In*, *Out* y *Reset* del generador de pulsos: (figura 18)



***Figura 18: Simulación del generador de pulsos***

**PRACTICA 3:**

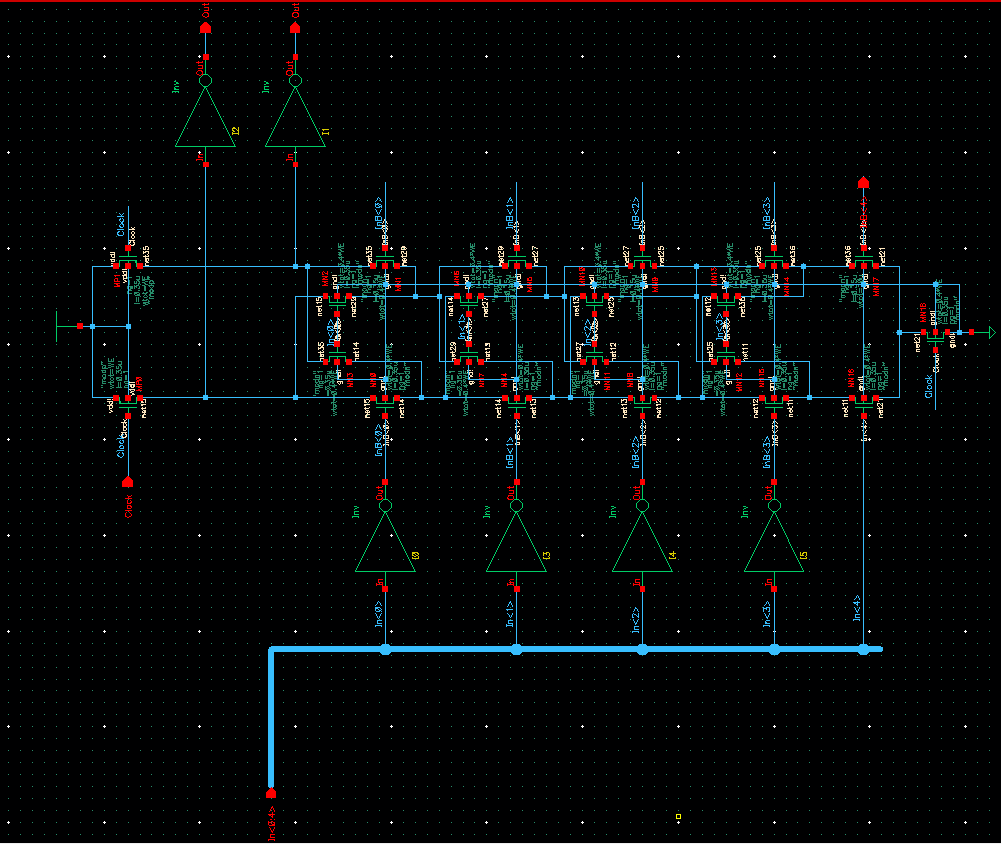
CIRCUITO CMOS DINÁMICO

Esta práctica consiste en la creación de una etapa CMOS dinámica.

Un circuito CMOS dinámico es una variante de los circuitos CMOS que utiliza un enfoque de pre-carga y evaluación para mejorar la velocidad de conmutación y reducir el número de transistores necesarios.

Este tipo de circuitos tiene varias ventajas, como un máximo swing de voltaje, menos carga de capacitancia y no hay corriente en estado estacionario. Sin embargo, también tiene desventajas como una alta impedancia de salida, la necesidad de pre-carga, problemas de cascada y problemas de carrera.

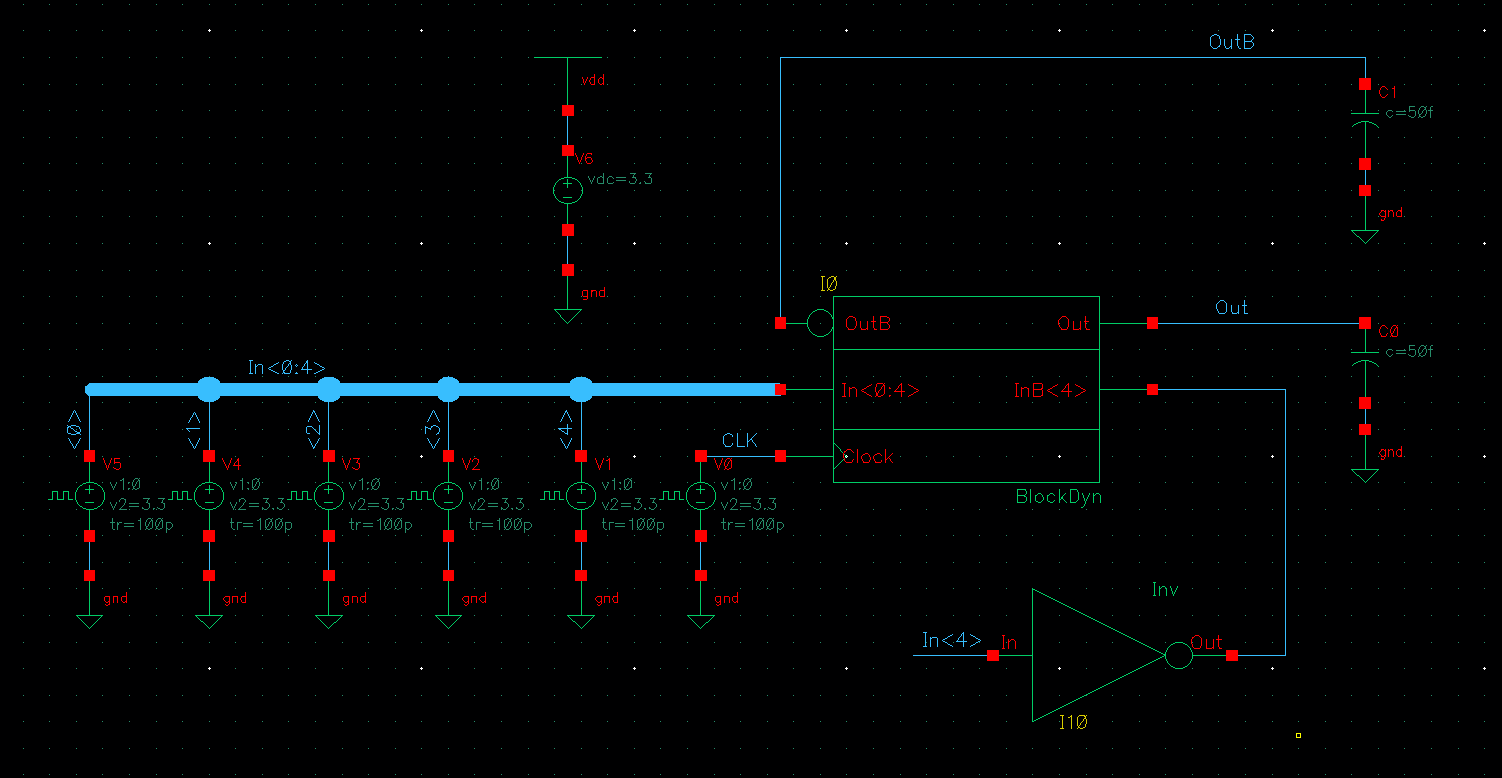
Nosotros hemos implementado el siguiente circuito en cadence del circuito CMOS dinámico, el cual hemos metido dentro de un módulo llamado BlockDyn (figura 19)



***Figura 19: Esquemático del BlockDyn***

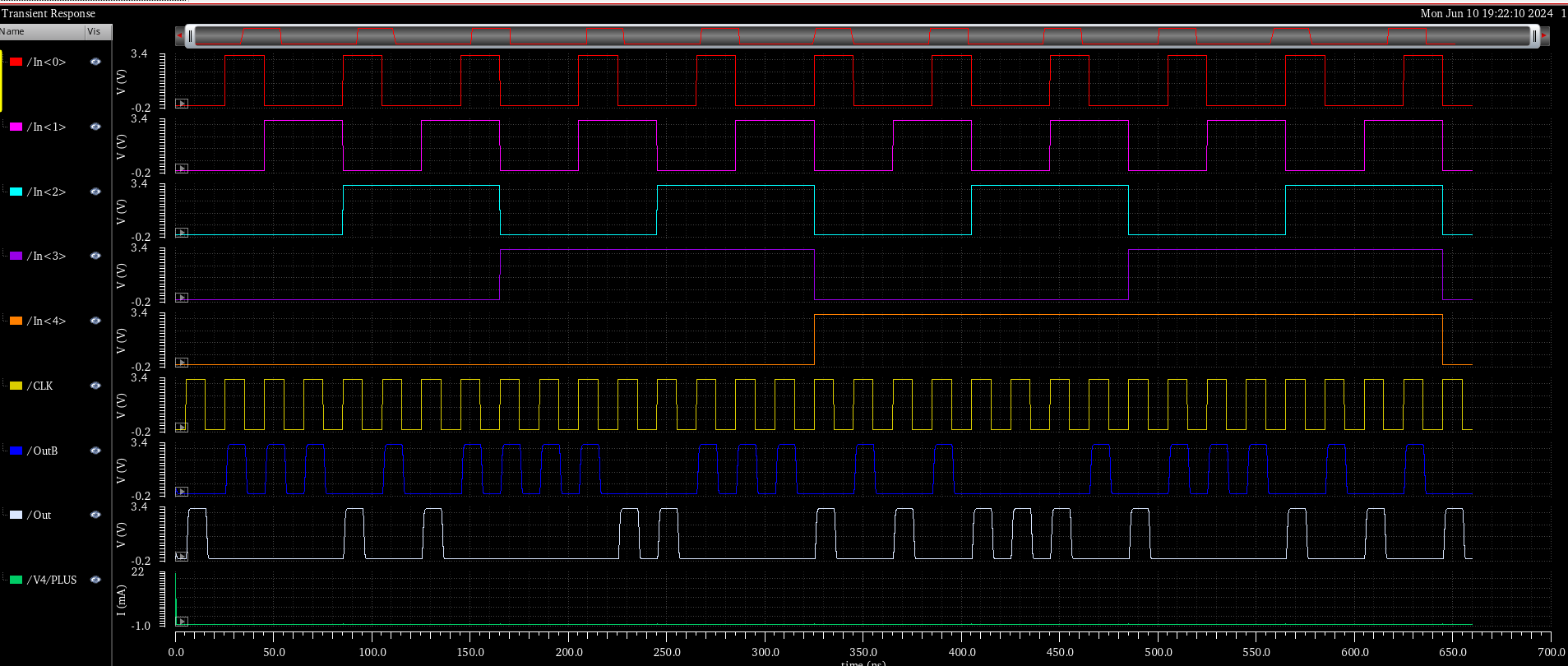
Este circuito tiene una entrada de 5 bits y está compuesto por 19 transistores nmos con una anchura de canal de 0.4u, y 2 transistores pmos ubicados en la parte superior con una anchura de 0.4\*WE, este WE es una variable que típicamente le pondremos el valor de 1u. El nmos del reloj tiene una anchura de 1\*WE.

Posteriormente hicimos un test bench con un generador de pulsos con periodos dependientes 2 a 2, para comprobar que funcionara correctamente. (figura 20)



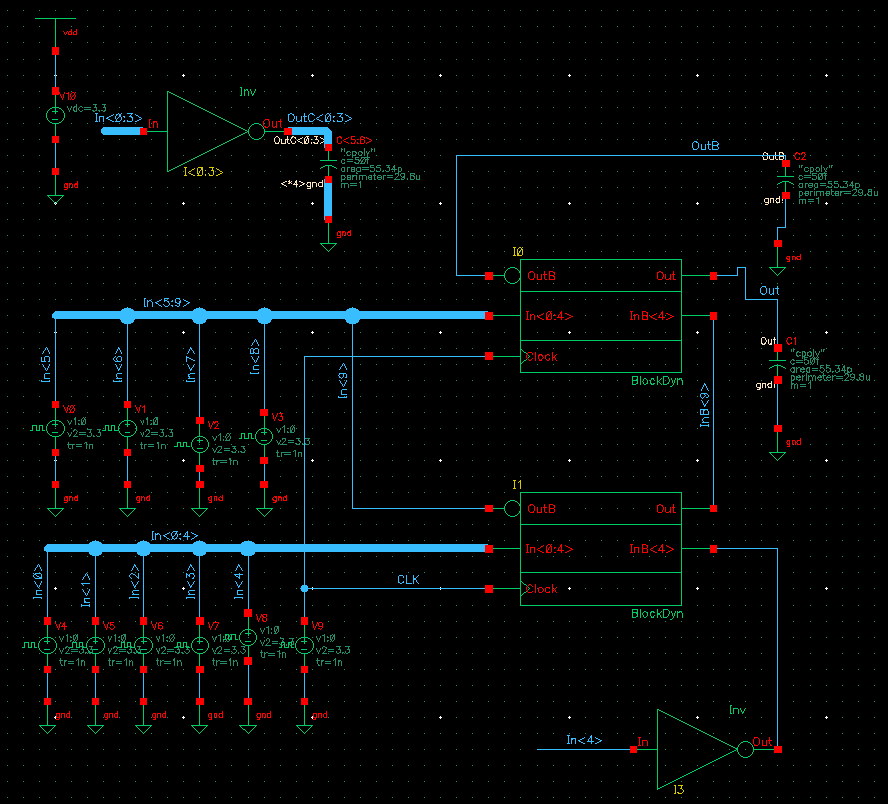
***Figura 20: Test bench del BlockDyn***

Una vez hecho las conexiones pertinentes lo simulamos y los resultados obtenidos fueron los siguientes: (figura 21)



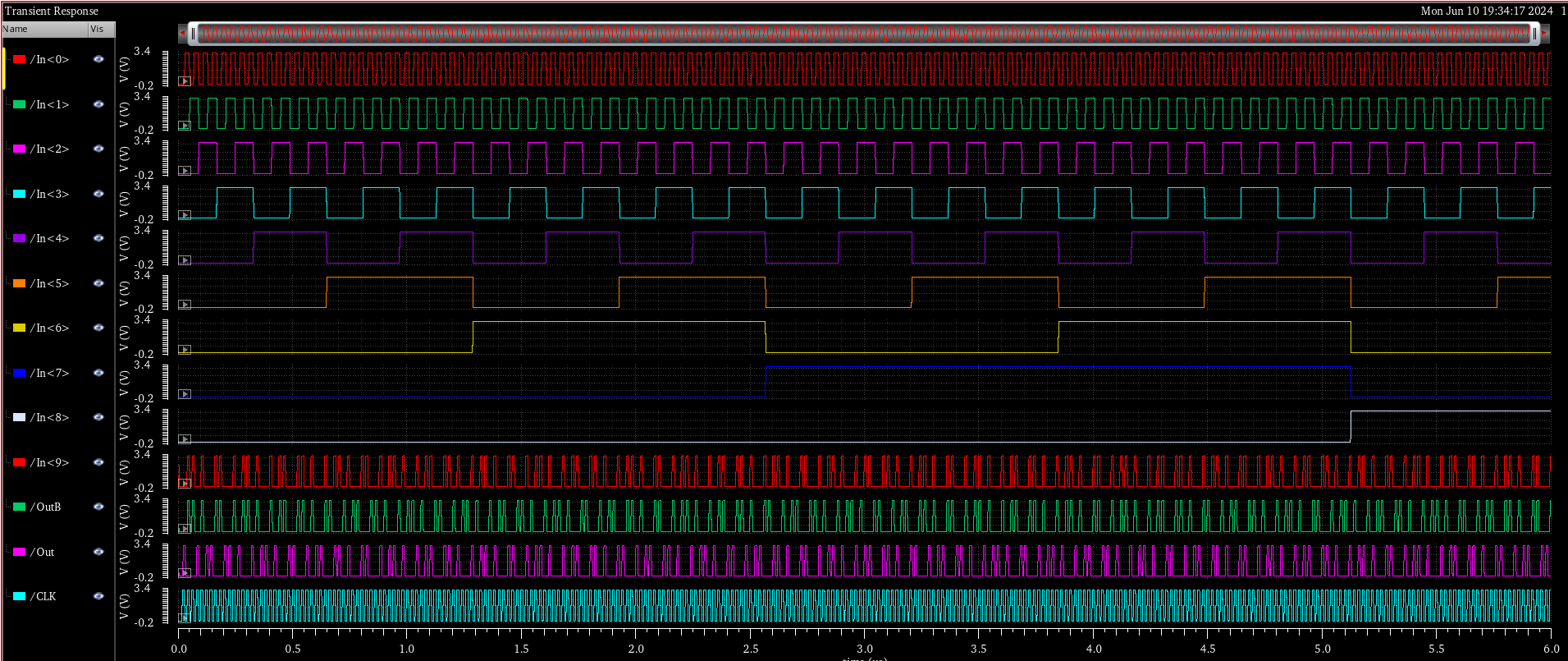
***Figura 21: Waveform del BlockDyn***

De esta forma, se puede visulizar la señal de Out como va dependiendo de las señales de entrada. Aparte, se nos pedía implementar un nuevo módulo que consistía en dos BlockDyn en cascacada. (figura 22) El cual nosotros hemos llamdo BlockDyn\_TB3.



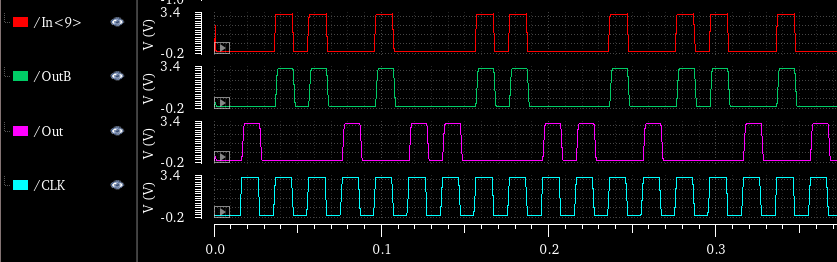
***Figura 22: Testbench del BlockDyn\_TB3***

Una vez simulado estos han sido los resultados obtenidos de la simulación.



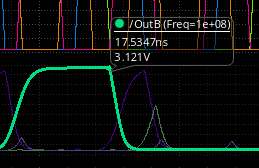
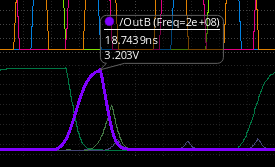
***Figura 23: Waveform del BlockDyn\_TB3***

Si hacemos *zoom* podemos observar el comportamiento de las señales que nos interesan. (figura 24)



***Figura 24: Señales ampliadas de la simulación***

Se aprecia claramente como las señales de Out y OutB están invertidas y el In<9> sigue el mismo comportamiento que la señal de OutB.



BlockDyn cas

Cascada el de arriba

Cascada comentas la señal

