MEMORIAS P2 Y P3

**Jaime Lloret Cuñat**

**Adam Cecetka Ortiz**

**Estructura de ficheros**

Práctica 2 Práctica 3

****

El proyecto esta dividido en 2 practicas situadas cada una en su librería correspondiente. Arriba podemos observar las distintas carpetas dentro de cada librería.

**P2**

* Trigger
  + Layout
  + Schematic
  + Av-extracted
  + Symbol
* Trigger\_tb
  + Schematic
  + Config
  + State “pruebaconfig”
* Trigger\_tb2
  + Schematic
  + State “Pulsegenerator”

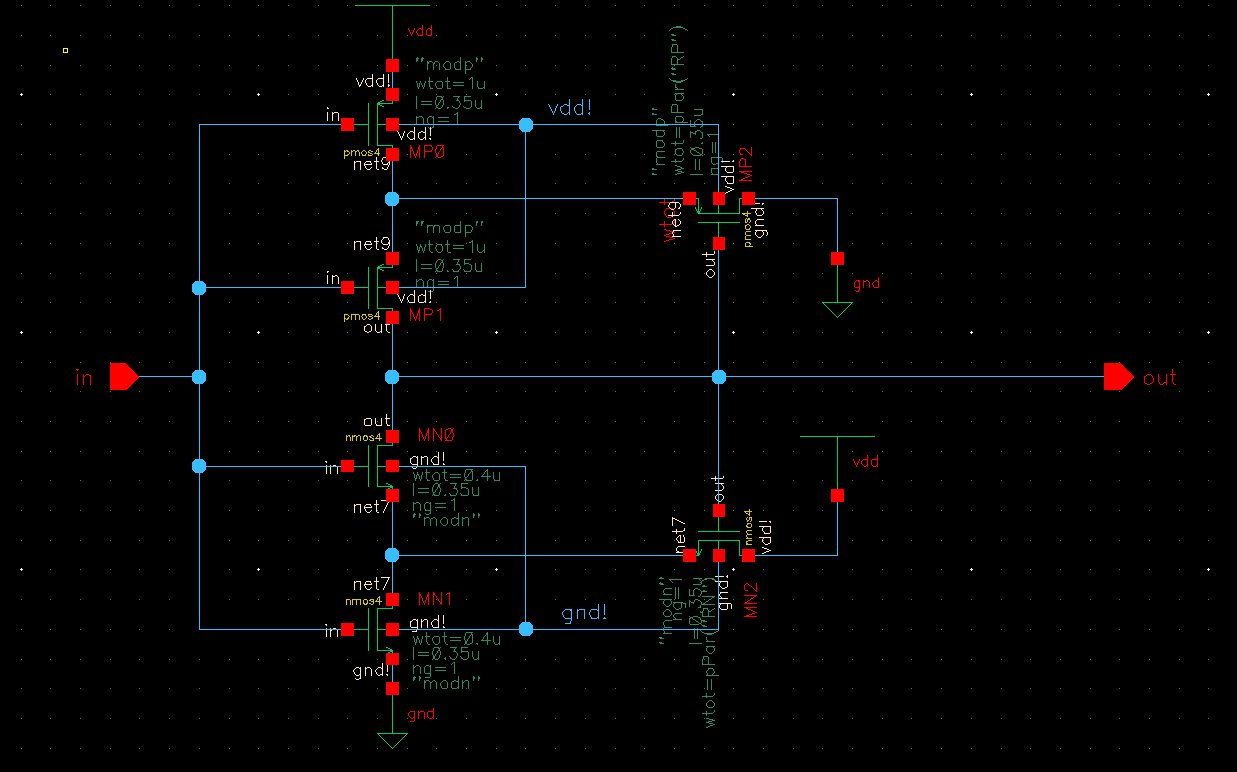
**P3**

* BlockDyn
  + Schematic
  + Symbol
  + Parametric “frecmax”
  + Parametric “withmin”
* BlockDyn\_tb
  + Schematic
  + State “blocktb1”
* BlockDyn\_tb3
  + Schematic
  + State “blocktb3”

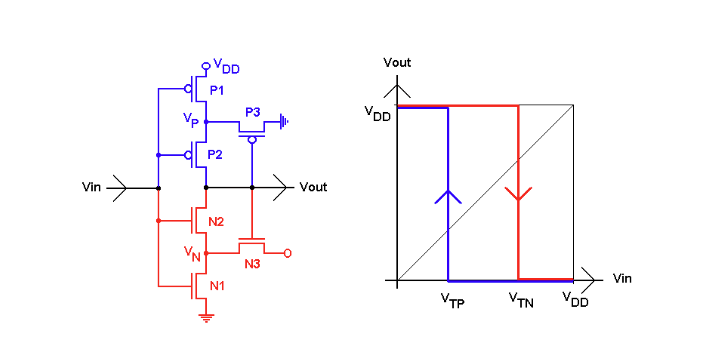
**PRACTICA 2**

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*.

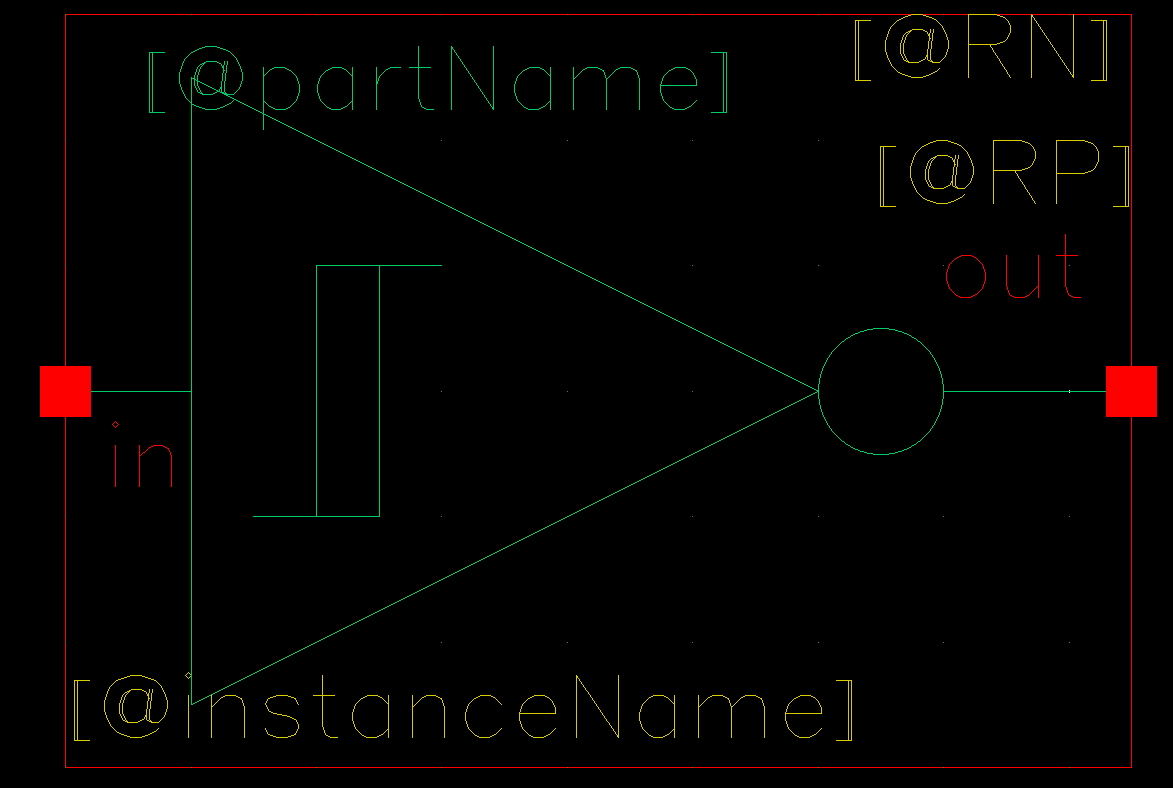


***Figura 1: Esquemático del inversor Trigger Schmitt***



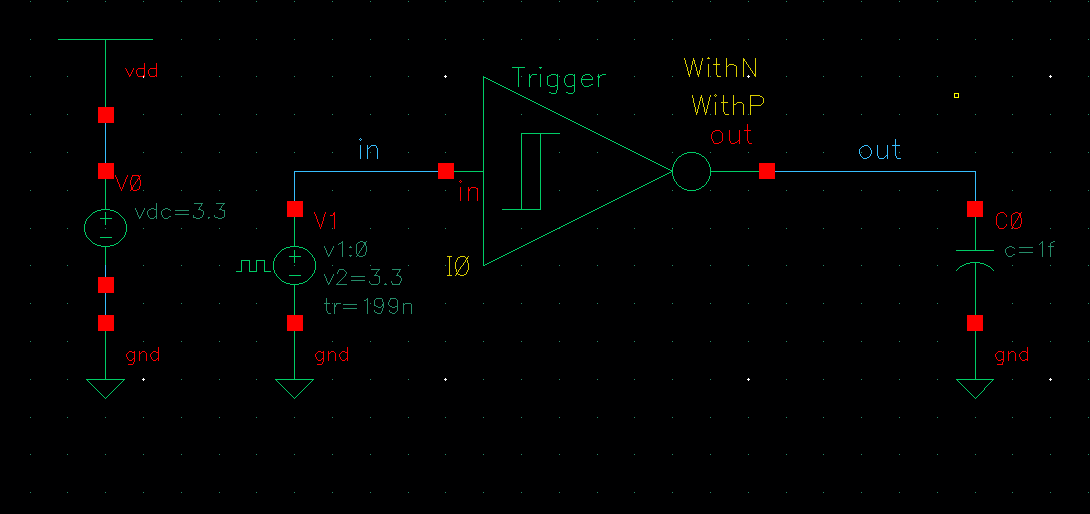
***Figura 2: Dimensionado de un Trigger Schmitt inversor***

A continuación, hicimos el símbolo con las instancias de RP y RM (figura 3).



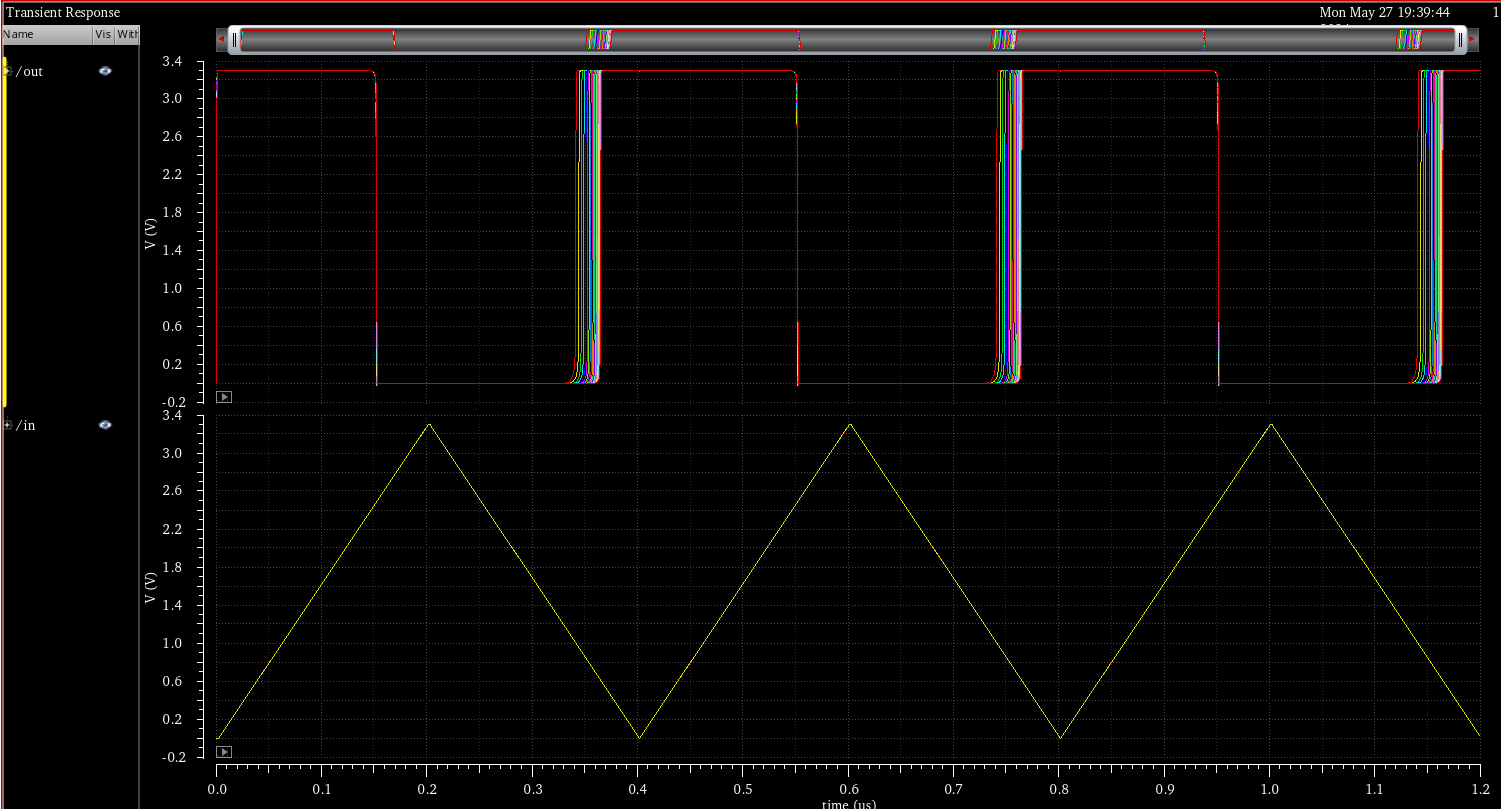
***Figura 3: Símbolo de Trigger Schmitt***

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***

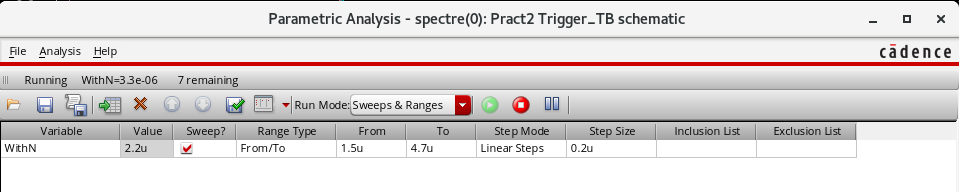
Gracias a este pudimos hacer la siguiente simulación. (figura 5)



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

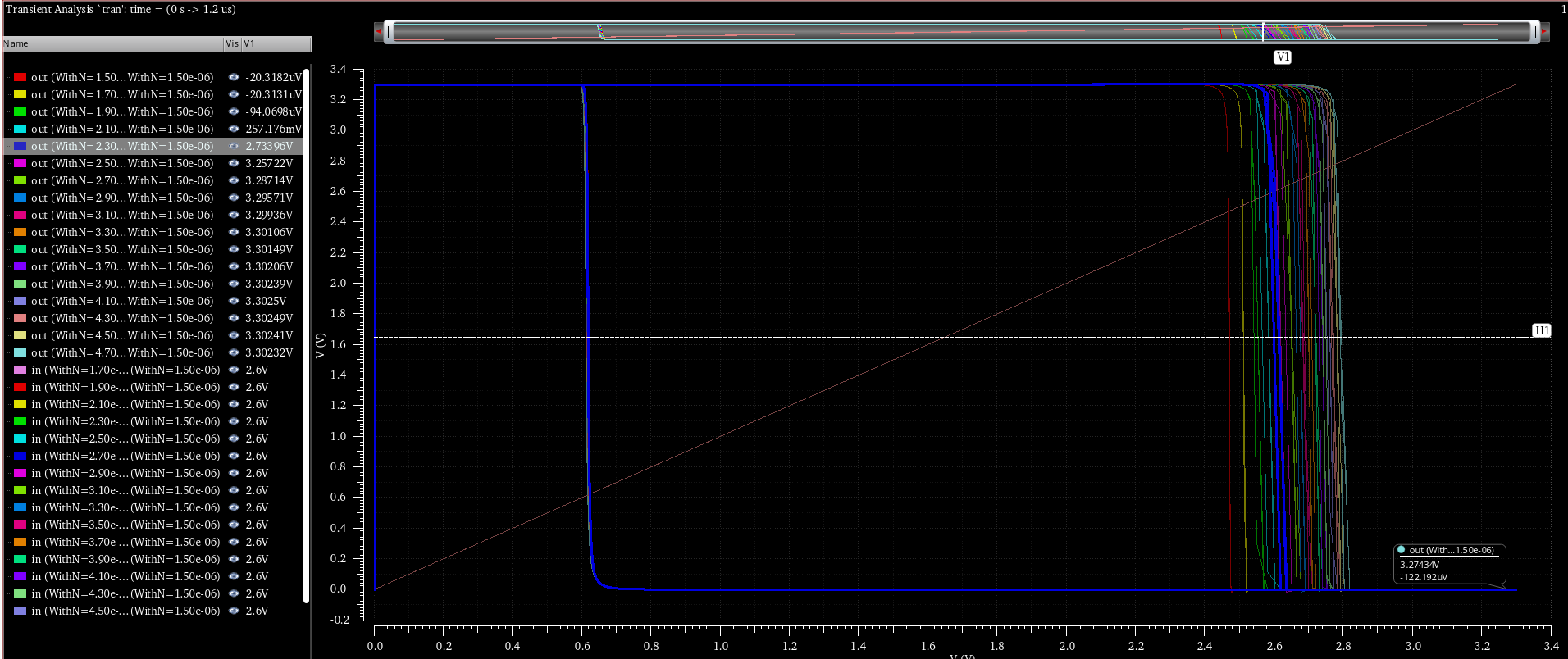
Gracias a esta, con la configuración de ejes de entrada de Y vs Y, pudimos extraer los valores óptimos de *WithN* (figura 6), y de WithP (figura 7).

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.

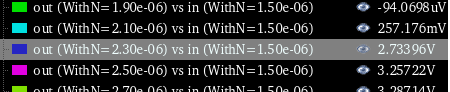


***Figura 6: Configuración del testbench***

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)

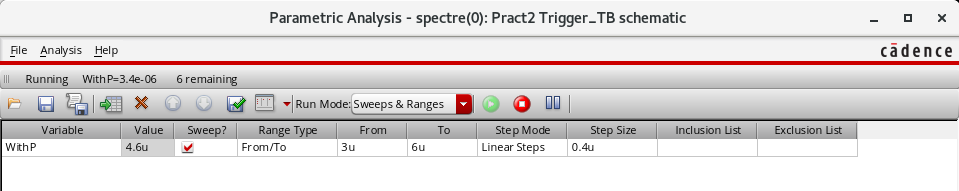
******

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

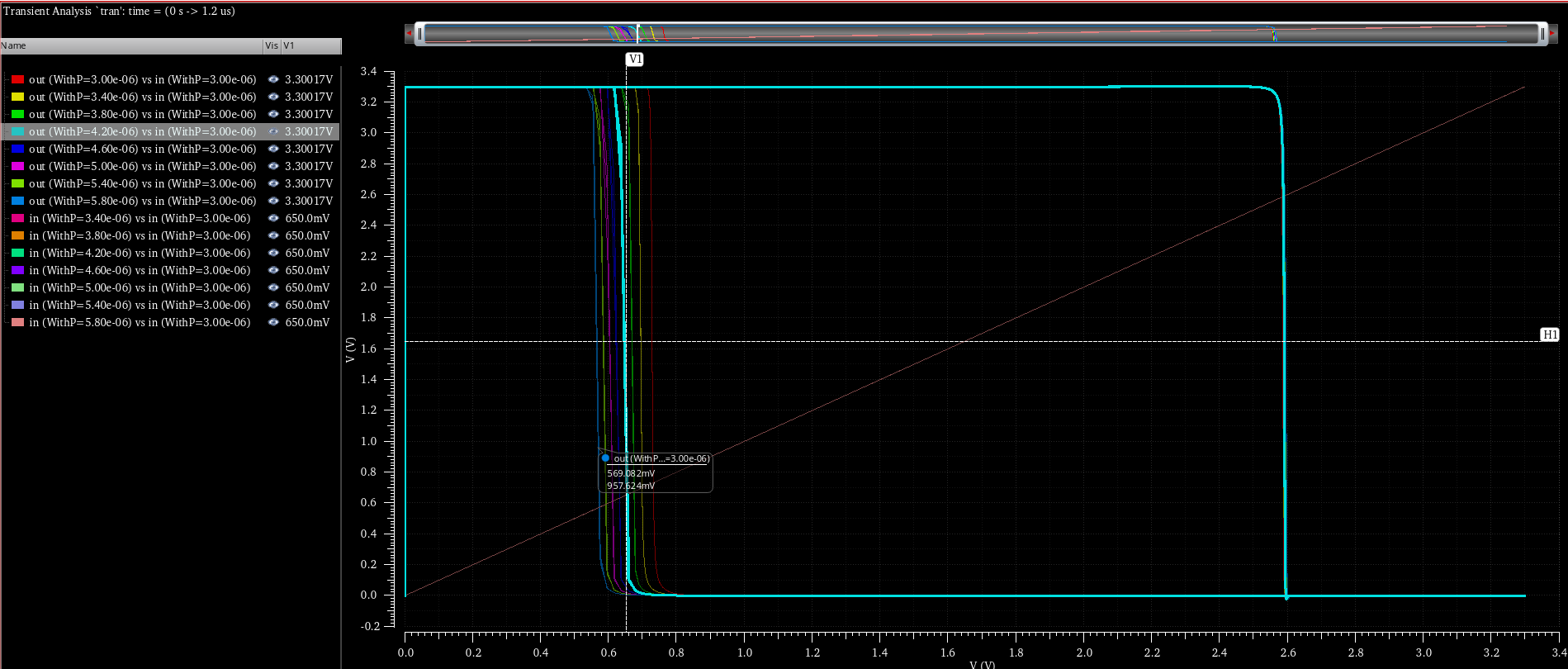
******

***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)

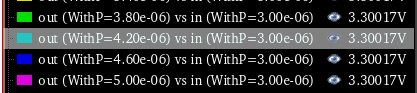
******

***Figura 9: Valores introducidos***



***Figura 10: Simulación Y vs Y del WithP***

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

******

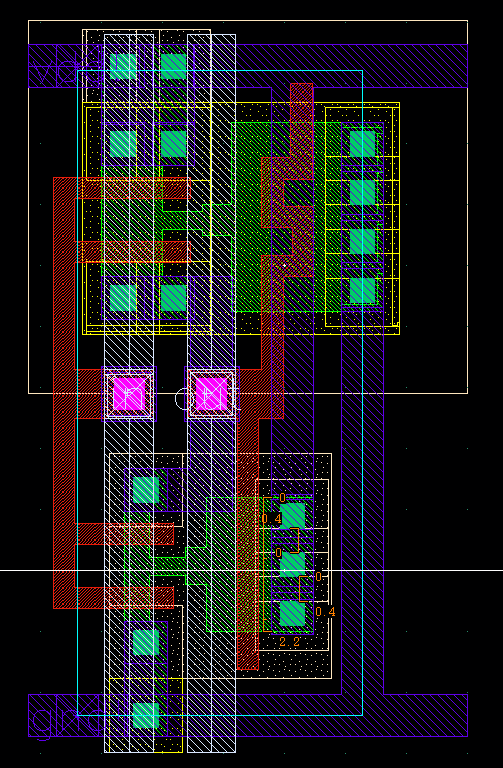
***Figura 11: Valor óptimo de WithN (4.2e-06)***

Paralelamente diseñamos el *layout*. (figura 12)

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

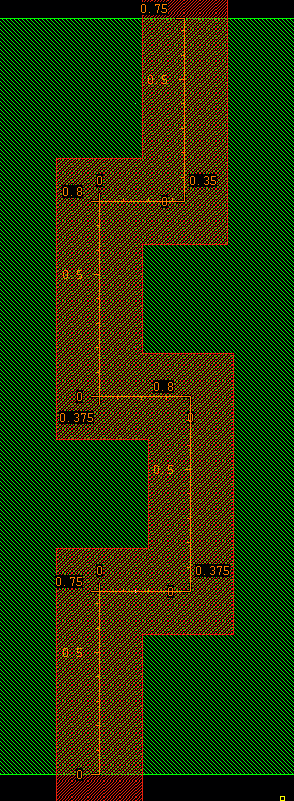
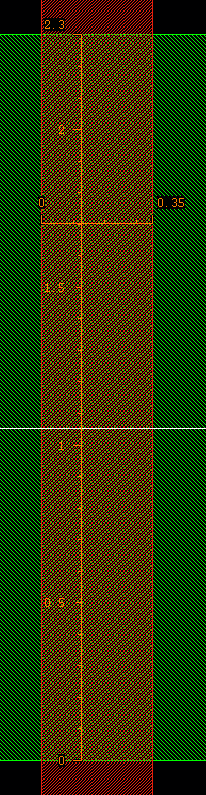
El polisilicio del *PPLUS* está en forma de serpiente para cumplir con el ancho de transistor, que tiene que ser igual a 4.6u.

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.



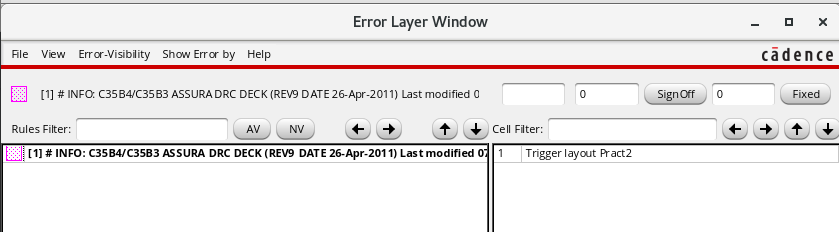
***Figura 12: Layout de Trigger Schmitt***

Para el diseño del *layout* hemos utilizado las medidas de los WithN y WithP para diseñar la anchura del transistor. En el pmos tuvimos que alargar el polisilicio para cuadrar las medidas. En nuestro caso se ve claramente en la figura 13 como 0.75+0.375+0.8+0.375+0.8+0.35+0.75 = 4.2 que era nuestro valor óptimo del WithN previamente obtenido. Para el nmos no tuvimos que hacer ningún apaño, pudimos implementar el valor de WithP, (figura 14) que en nuestro caso era 2.3, para la anchura del transistor. Aparte, longitud de canal la pusimos en 0.35u como nos indica en las reglas de diseño. Se puede apreciar en la figura 14.

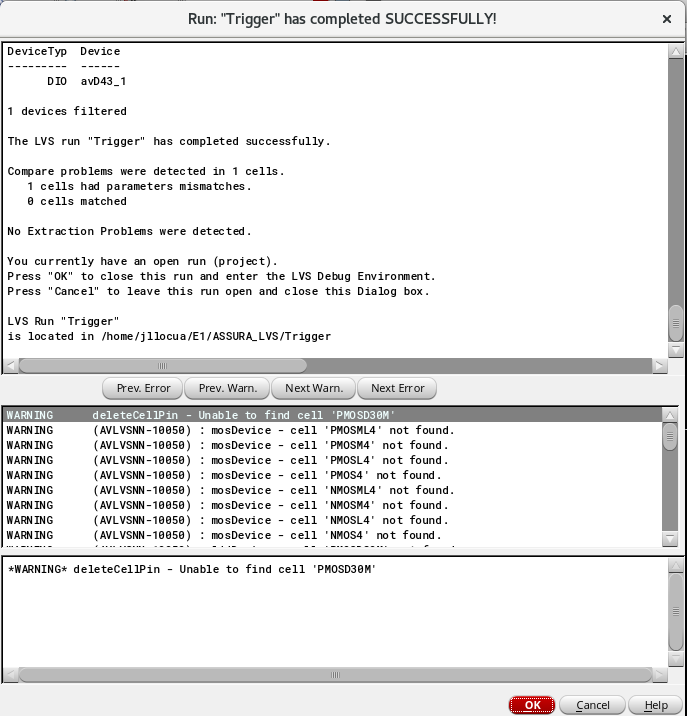
 

***Figura 13: Canal transistor pmos Figura 14: Canal transistor nmos***

Una vez terminado el *layout* comprobamos que no se estuviera incumpliendo ninguna regla de diseño con un Run Asura DRC. (figura 15)



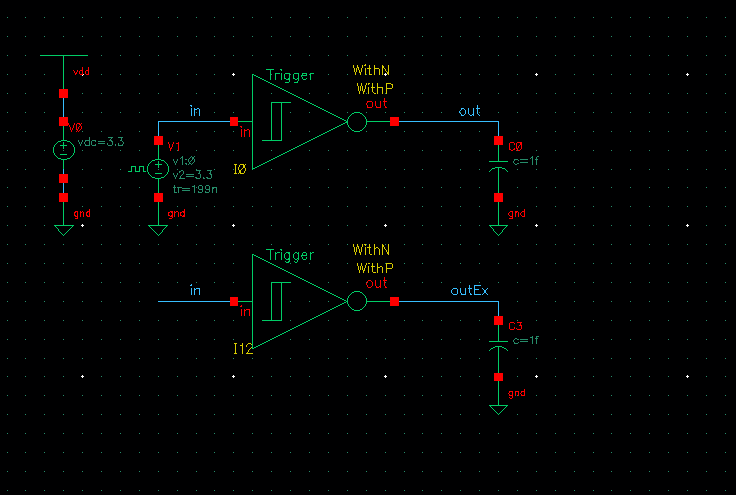
***Figura 15: Run Asura DRC del Trigger Schmitt***



***Figura 16: Run Asura LVS del layout***

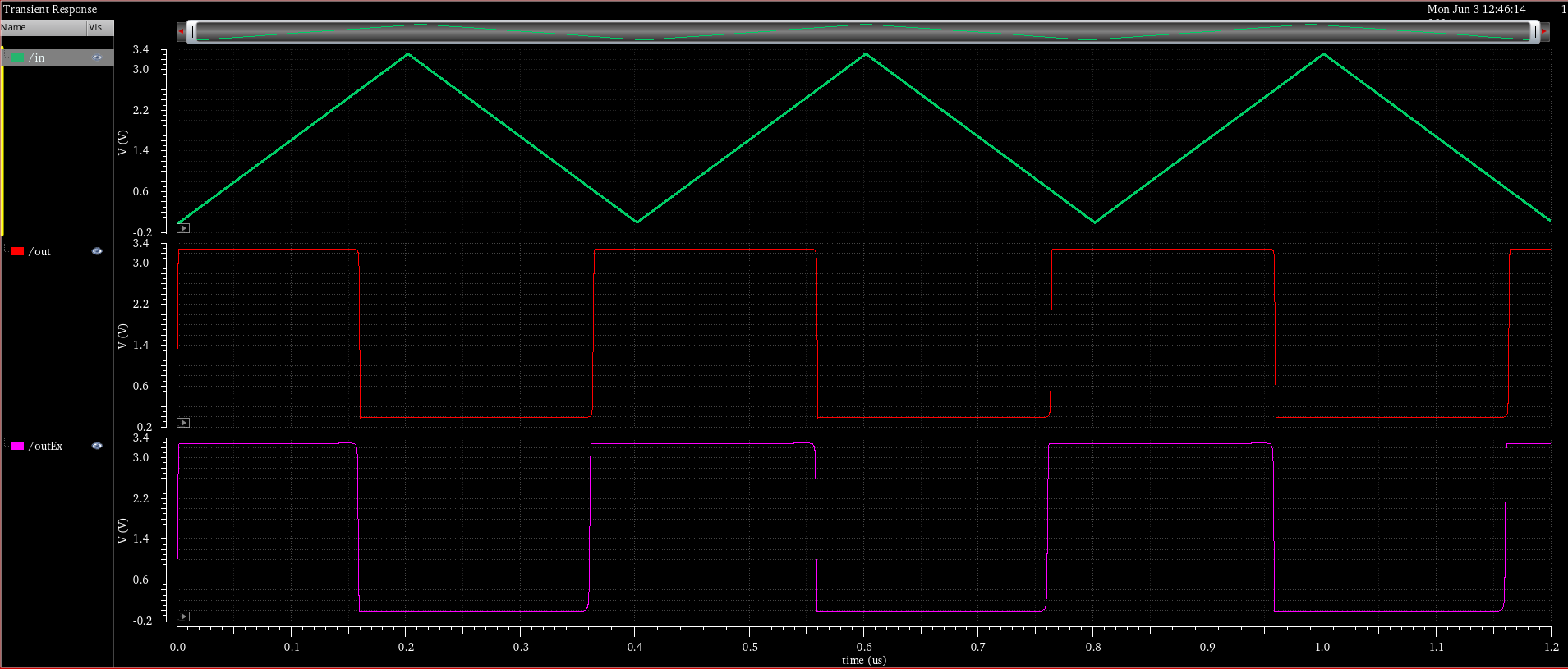
A su vez, también nos aseguramos de que no hubiera ningún problema con las conexiones del *layout* al esquemático con un Run Asura LVS. (figura 16)

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 18). La anchura del transistor NMOS



***Figura 17: Esquemático del testbench dual***

Ejecutamos la simulación y se ve claramente que las señales del esquemático del diseño y del *av\_exctracted* coinciden. (figura 18)

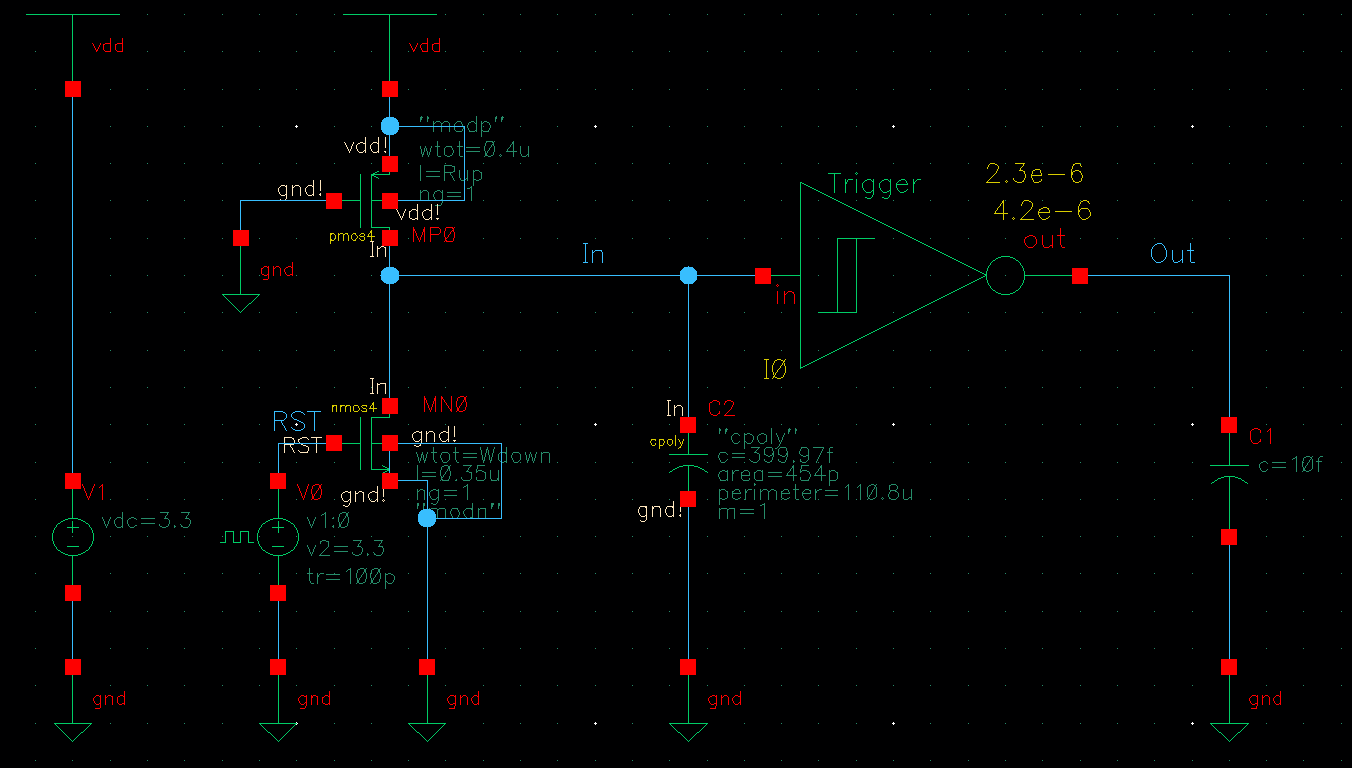


***Figura 18: Resultados del testbench***

**GENERADOR DE PULSOS**

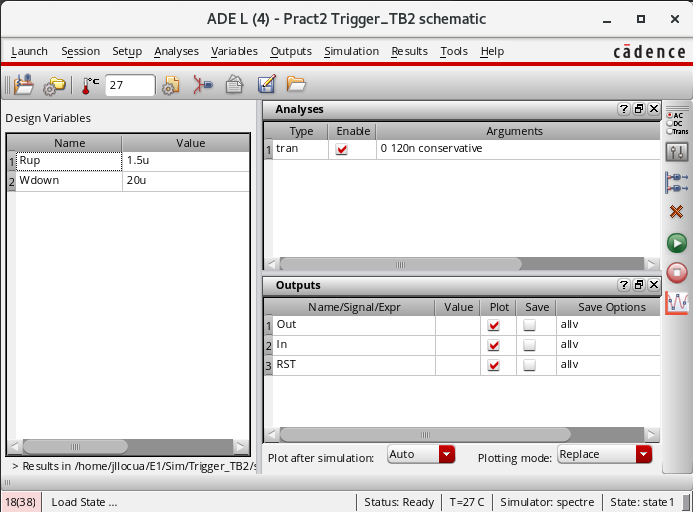
Por último, se nos pedía utilizar los diseños empleados anteriormente para la aplicación de un generador de pulsos basado en el *Trigger Schmitt*.

Este es el diseño que se nos proponía realizar: (figura 19)



***Figura 19: Esquemático del generador de pulsos***

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



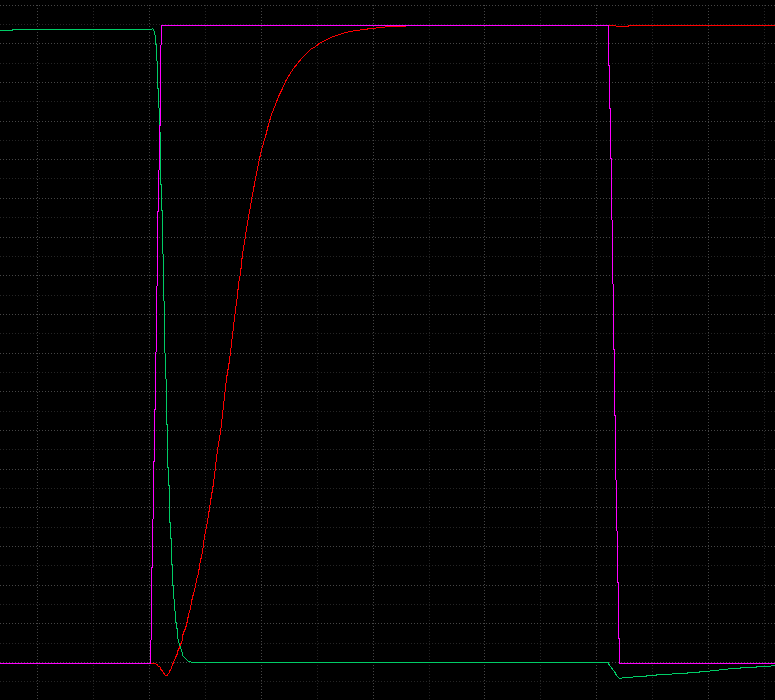
***Figura 20: Configuración de la simulación del generador de pulsos***

Y estos son los resultados obtenidos de las señales *In*, *Out* y *Reset* del generador de pulsos: (figura 21)



***Figura 21: Simulación del generador de pulsos***

Esta es una simulación de 160ns. Hemos introducido un delay al reset de 140ns para estabilizar la señal de In. También hemos configurado un ancho de pulso de salida de 60 ns. (Se nos exigía que fuera mayor o igual a 40ns). Aparte también hemos puesto una anchura de reset de 4 ns como se nos indicaba. (figura 21)

Si hacemos un poco de zoom vemos como funciona el generador de pulsos. (figura 21)

IN -> VERDE

OUT -> ROJO

RESET -> ROSA

Se ve claramente en la figura 21 como el reset (señal rosa) se activa y consecuentemente baja la señal de In (señal verde). Y cuando esto pasa, al tratarse de un Trigger Schmitt, la señal de Out (señal roja) empieza con su flanco de subida. (figura 21)

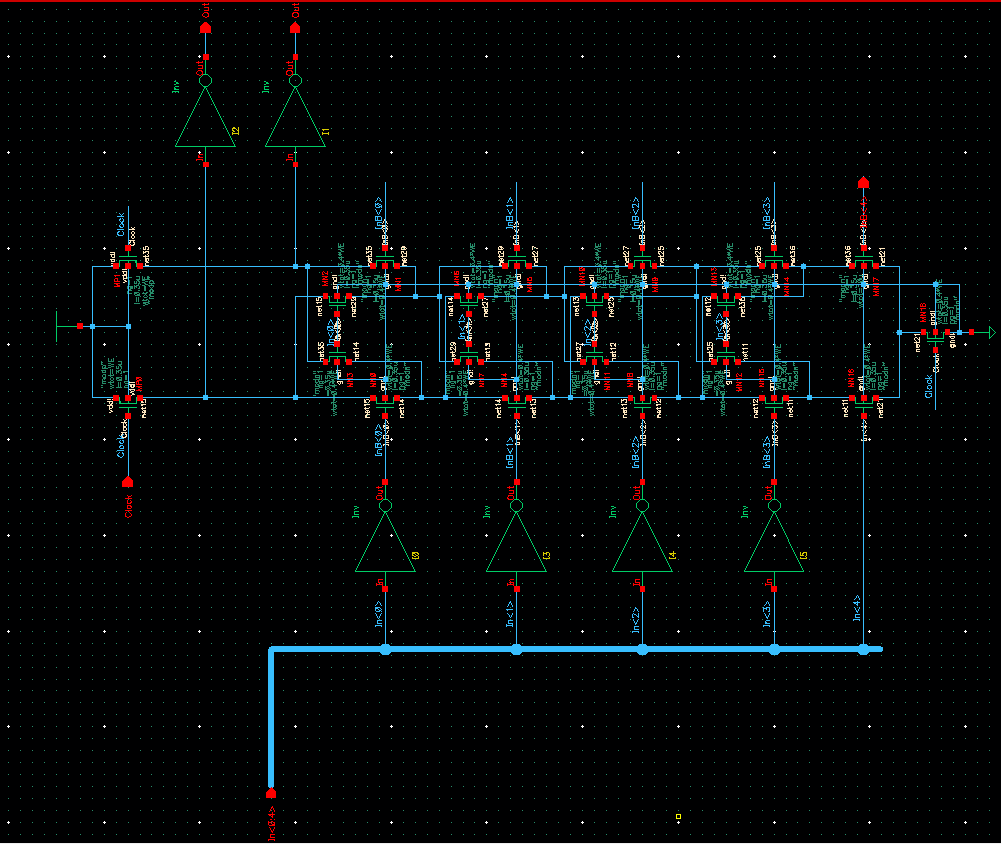
***Figura 21: Ampliación de la simulación.***

**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.



***Figura 22: Esquemático del BlockDyn***

**1. ¿Qué Tipo de Lógica Dinámica es y qué Función Combinacional realiza?**

La función que realiza es una XOR de las entradas. El tipo de lógica empleada es de tipo dominó diferencial debido a la incorporación de inversores y a que tenemos además de la salida XOR también la XNOR.

**2. ¿Qué porcentajes del Periodo de Reloj asignaría, en principio, a la Precarga y a la Evaluación?**

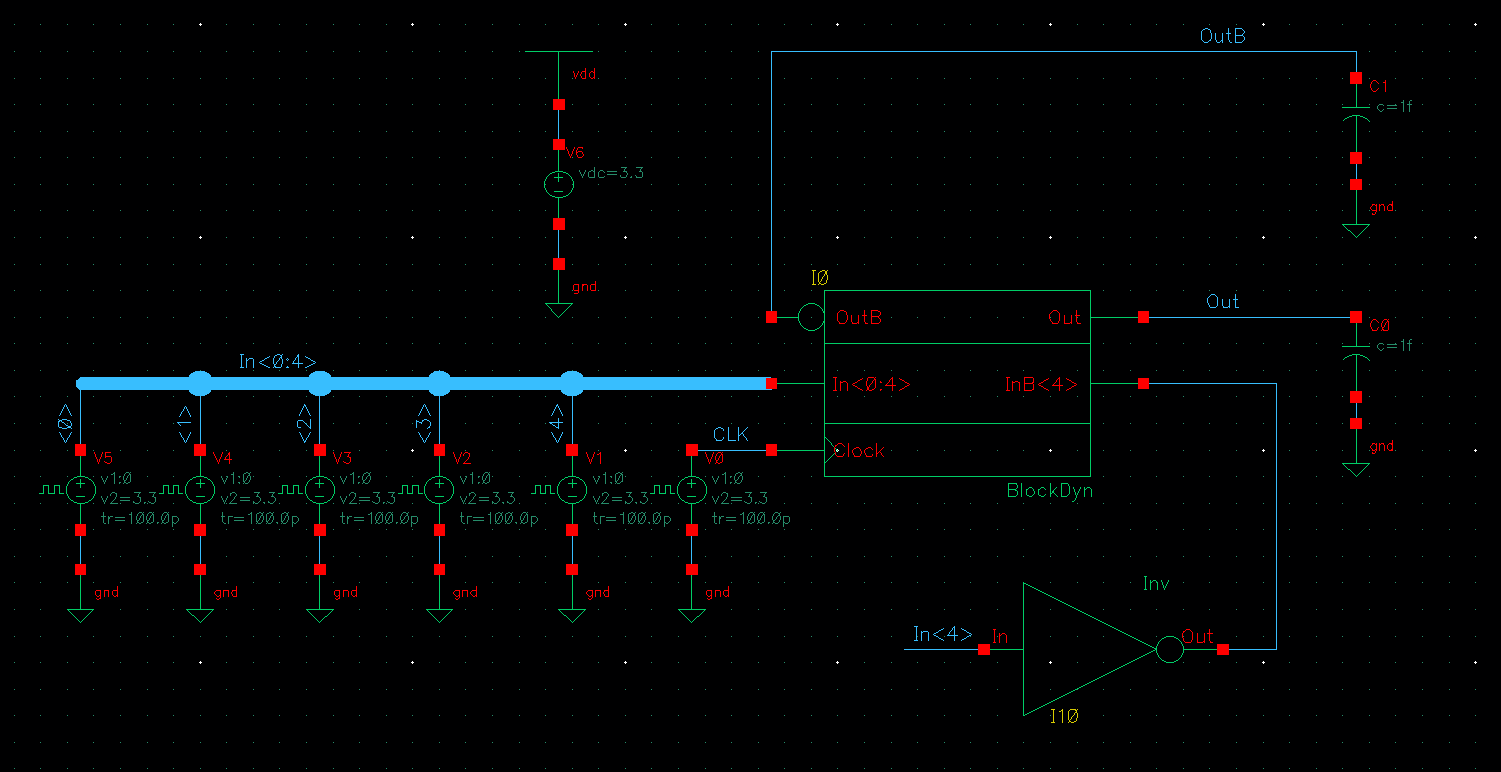
Ya que la fase crítica es la fase de evaluación que es donde se determina la lógica del circuito, asignaremos un mayor porcentaje del periodo del reloj. Emplearíamos 30-40% del periodo para la precarga y 60-70% para la evaluación. Pese a esto, en la práctica se ha decidido emplear unos porcentajes 50-50%.

Nosotros hemos implementado el siguiente circuito en *cadence* del circuito CMOS dinámico, el cual hemos metido dentro de un módulo llamado *BlockDyn* (figura 22)

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.

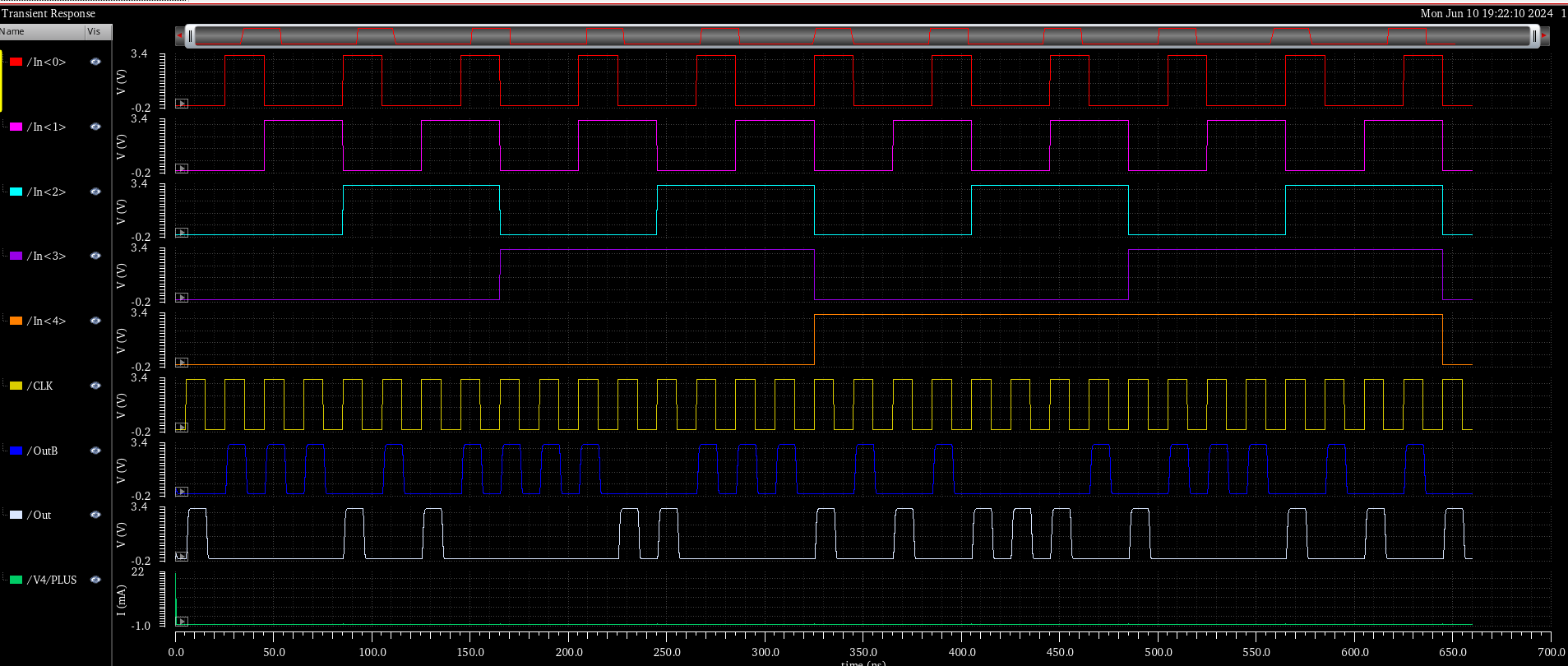
Este circuito tiene una entrada de 5 bits y está compuesto por 19 transistores nmos con una anchura de canal de 0.4u, y 2 transistores pmos ubicados en la parte superior con una anchura de 0.4\*WE, este WE es una variable que típicamente le pondremos el valor de 1u. El nmos del reloj tiene una anchura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

Posteriormente hicimos un testbench con un generador de pulsos con periodos dependientes 2 a 2, para comprobar que funcionara correctamente. (figura 23)



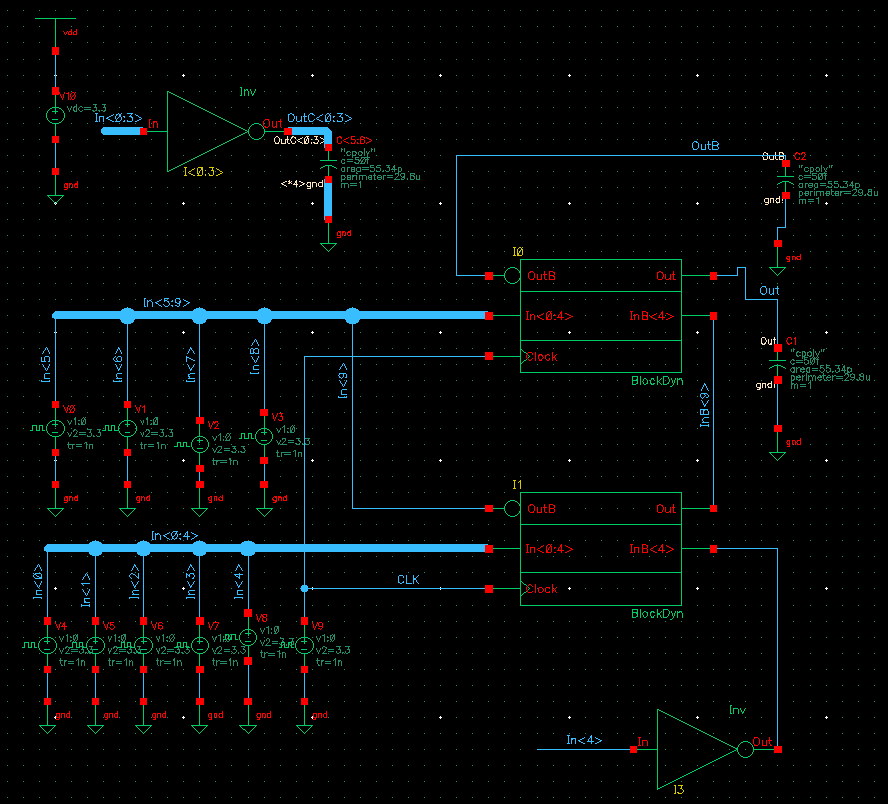
***Figura 22: Testbench del BlockDyn***

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



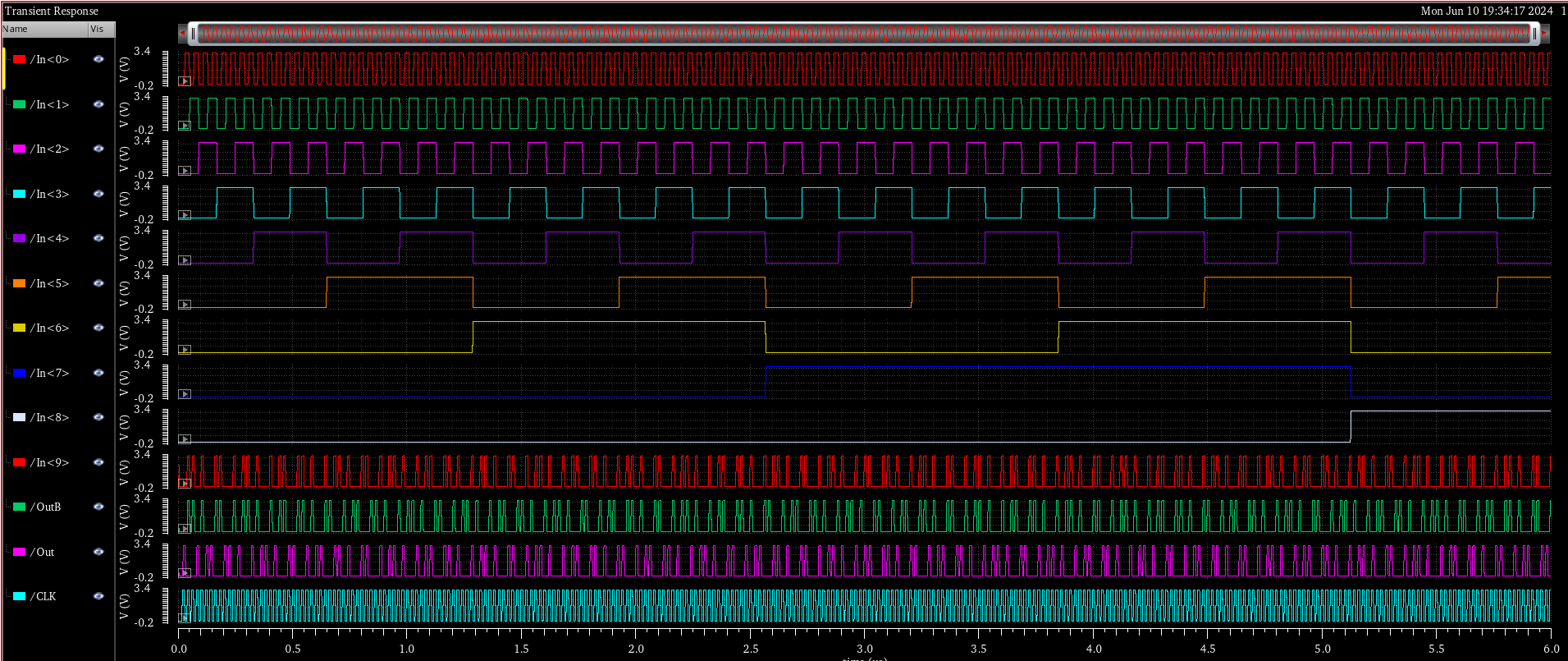
***Figura 23: Waveform del BlockDyn***

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 cascada. (figura 22) El cual nosotros hemos llamdo BlockDyn\_TB3.



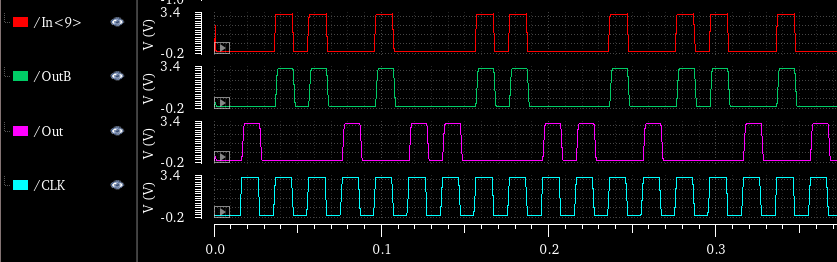
***Figura 24: Testbench del BlockDyn\_TB3***

Una vez simulado estos han sido los resultados obtenidos de la simulación. (figura 25)



***Figura 25: Waveform del BlockDyn\_TB3***

Si hacemos *zoom* podemos observar el comportamiento de las señales que nos interesan. (figura 26)



***Figura 26: 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. (figura 26)

En el guión se formula la siguiente pregunta sobres esto:

**1.Advierta la asignación, de izquierda a derecha de las señales del bus In<5:9> a los pines In<0:4> de I15. Explique por qué las salidas Out y OutB de la 1ª etapa (I16) deben actuar a través de las In<4> e InB<4> en la etapa I15 y no sobre cualquiera de sus In<0:3>.**

Esto es debido a que In<4> se encarga de la separación entre las dos etapas y conectando la siguiente etapa en esta entrada las conseguimos mantener independientes la una de la otra.

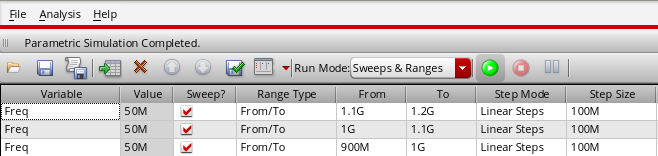
**2. ¿Explique por qué, justamente, Out y OutB deben actuar, respectivamente, sobre InB4 e In4 en ese orden y no al revés?**

Usamos este orden porque sinó no obtendríamos salida. Para hacerlo en otro orden habría que invertir todas las señales de entrada y ponerlas al revés

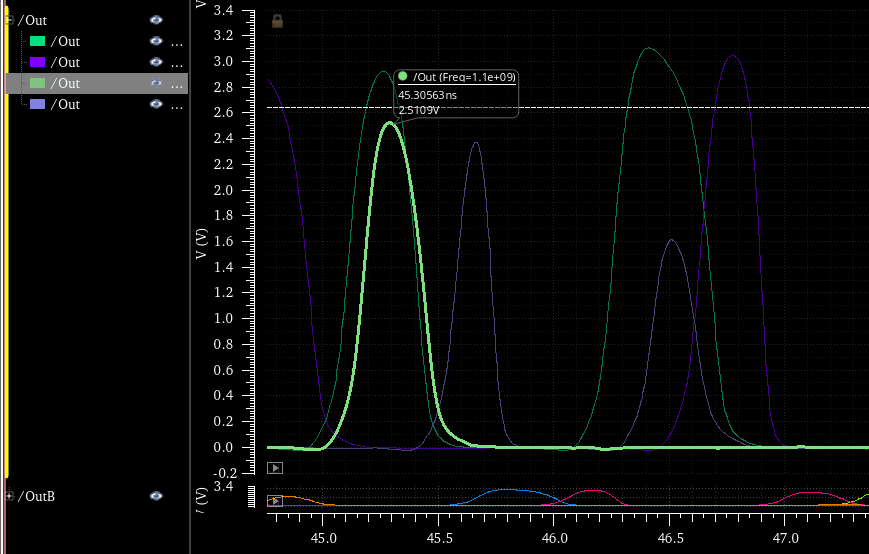
Análisis Paramétrico

A continuación, se responderá a las siguientes cuestiones realizando un análisis paramétrico.

**3. Con los valores iniciales de WP y WE…. ¿Cuál es el valor máximo de la frecuencia Freq de Clock, supuesto el reloj simétrico, con tiempos fijos trise = tfall = 0,1 ns? Para ello adopte las variables de simulación WE y Freq y realice un análisis paramétrico sobre Freq.**



***Figura 26***



Considerando 2.65V el umbral para la tecnología observamos en el análisis paramétrico como a partir de 1.1 GHz (la señal de color amarillo) la señal empieza a estar por debajo de ese valor de forma considerable por lo que diremos que ese es nuestro valor de frecuencia máxima.

**4. Tomando WE como variable. ¿Cuál es su valor mínimo para operar a 500 MHz, con trise = tfall = 0,1 ns?**

Decidimos barrer en un rango de 0.1 a 0.3 micras y podemos observar como para el valor de 0.2u obtenemos una señal por debajo del umbral común para la tecnología empleada, 2.65u.

