

# Proyecto Final

Gabriel Araya Mora B80525

*Escuela de Ingeniería Eléctrica, Universidad de Costa Rica*

*Dispositivos Semiconductores (IE-0311)*

---

---

## 1. Construcción de un Flip Flop con transistores MOSFET

### 1.1. Latches

El latch es el elemento más básico de memoria digital. El latch CMOS consiste en dos inversores conectados de forma que haya realimentación positiva entre ellos. El Latch es dispado por nivel, es decir, en los semi-ciclos positivos del reloj está transparente, y deja pasar cualquier dato que haya en su entrada, pero cuando semi-ciclo de reloj es negativo, el latch se encuentra en memoria, y no importa el valor de entrada va a tirar en la salida el valor que había antes de la transición. [1]

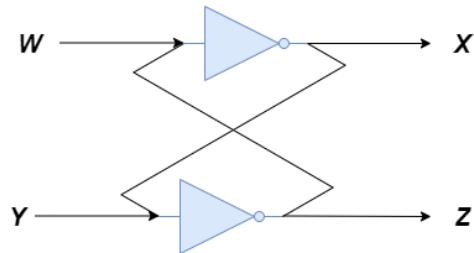


Figura 1: LATCH Básico. (Creación Propia)

### 1.2. D FLIP-FLOP

Para la construcción más básica de un flip flop, pero, un poco complicada, consta en conectar dos latches con una topología de Maestro-Esclavo, como se muestra a continuación:

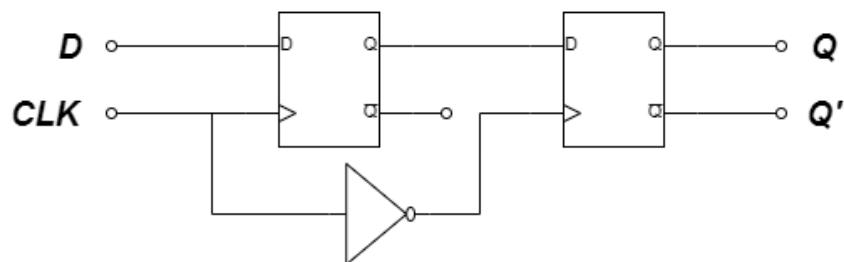


Figura 2: Topología Maestro-Esclavo. (Creación Propia)

El Flip-Flop de tipo D, consta de dos entradas y dos salidas en su forma más básica. La primera entrada, es la entrada de datos ( $D$ ), y la otra entrada es el reloj ( $clk$  o  $\phi$ ). La construcción es casi la misma que la del latch presentado en la figura anterior; sin embargo, ahora el bucle de realimentación solo va a estar activo una parte del tiempo. De hecho el lazo de realimentación solo estará cerrado cuando el reloj esté bajo o en un cero lógico. Además es necesario “aislar” la entrada del conjunto de inversores de forma que si por alguna razón esta cambiara, no afecte los valores en el lazo de realimentación. Para esto se pone un switch que se cierra cuando el reloj está en alto. Cuando el reloj pasa de bajo a alto, el flip-flop adquiere el nivel de lógica que venía entrando por la línea de datos  $D$  un pequeño tiempo antes del flanco de reloj. Esto dice que el flip-flop es disparado por flancos. [2]

De lo anterior se puede llegar a la conclusión, de que cuando el reloj está en un uno lógico, el lazo de realimentación está abierto y la entrada de datos pasa directamente a la entrada del primer inversor. Ahora cuando el reloj pasa de alto a bajo, la entrada de datos se separa por completo del circuito y el lazo de realimentación se cierra para “guardar” el valor anterior de que el ciclo de reloj pasara a bajo. El flanco de reloj en el cual se dispara el flip-flop, se puede modificar sencillamente cambiando las entradas de reloj y el opuesto al reloj de la siguiente figura. [3]

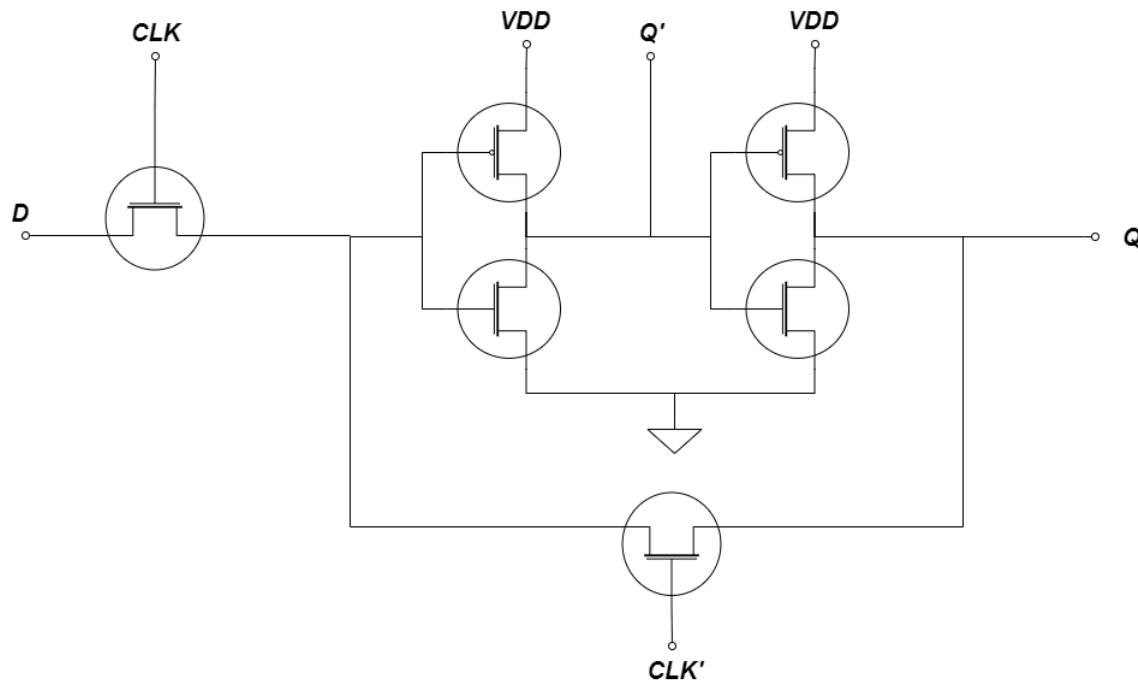


Figura 3: Flip-Flop Básico con topología CMOS. (Creación Propia)

El Flip-Flop, al ser un circuito secuencial, cuando se “arranca” en la salida se puede iniciar con un 1 o un cero con un 50 % de probabilidad, esto debido al fenómeno de power-up/Reset. Sin embargo esto no afecta, ya que cuando se empiezan a recibir datos, los estados comienzan a cambiar y el flop “memoriza” el estado anterior.

## 2. Dimensionamiento de transistores MOSFET mediante el método de esfuerzo lógico

Es pertinente empezar hablando de qué es el esfuerzo lógico. Bueno el esfuerzo lógico viene a responder de una forma rápida y eficaz varias preguntas.

- ¿Cuál es la mejor topología para construir un circuito lógico?
- ¿Cuál es el número de etapas que debe tener el circuito para tener el menor retardo posible?
- ¿Qué tan grandes deben ser los transistores para tener el menor retardo posible?

La teoría del esfuerzo lógico va a resolver de una forma tan simple estas tres preguntas, que los cálculos se pueden hacer en una servilleta tomando el café de la mañana. Sin embargo el esfuerzo lógico solo sirve para optimizar el retardo y no el área o el consumo de potencia del circuito. [4]

Se define el esfuerzo lógico como la razón de la capacitancia de entrada de la compuerta entre la capacitancia de entrada de un inversor de referencia o unitario capaz de entregar la misma corriente en la salida. [5]

Esto permite calcular el parámetro  $g$  de una manera muy sencilla, se cuentan los anchos que ve cada entrada para el  $C_{in}$  y para obtener la relación se toma un  $C_{in}$  del inversor de referencia, depende del parámetro  $\gamma$  o la movilidad de los electrones. ( $g$  se cuenta para cada entrada como individual.) Existe un parámetro importante más antes de pasar al dimensionamiento usando esta técnica. Se define  $h$  como el esfuerzo eléctrico y se obtiene como la capacitancia de salida entre la capacitancia de entrada. (Cuantas veces puede manejar su propio peso la compuerta)

Para dimensionar transistores utilizando el esfuerzo lógico, se debe repartir el esfuerzo entre todas las compuertas equitativamente. De hecho el retardo va a ser el más pequeño posible cuando cada etapa maneje el mismo peso. [5]

$$\hat{f} = g_i h_i = F^{\frac{1}{N}} \quad (1)$$

y el mínimo retardo de cada etapa del circuito es:

$$D = NF^{\frac{1}{N}} + P \quad (2)$$

Donde N es el número de etapas del circuito y P las capacitancias parásitas. Ahora teniendo este resultado se obtiene un criterio para dimensionar los transistores.

$$\hat{f} = gh = g \cdot \frac{C_{out}}{C_{in}} \Rightarrow C_{in_i} = \frac{g_i C_{out_i}}{\hat{f}} \quad (3)$$

Entonces si se conoce la capacitancia de salida de todo el circuito, se puede trabajar para atrás y obtener las capacitancias de entrada de todo el circuito y por ende se obtienen los anchos de las compuertas que dan el menor retardo posible. Obtenidas las capacitancias de entrada, se deben dividir entre los transistores n-mos y p-mos. Esto es muy sencillo ya que se multiplican los números del dimensionamiento base por un parámetro K, y se suman las capacitancias de entrada para una entrada específica y se despeja el parámetro K como el nuevo W y así queda hecho el dimensionamiento por esfuerzo lógico. (Cabe destacar que siempre se debe hacer el dimensionamiento por simetría, ya que de él salen los parámetros para el esfuerzo lógico) [5]

### 3. Diseño de un flip - flop utilizando $C^2MOS$ :

Después de hacer una extensa investigación, se llega a la conclusión de que la lógica CMOS tradicional, si bien hace su trabajo, tiene varios problemas intrínsecos con su diseño e ideología. Empezando porque la lógica CMOS es redundante en sí misma, se tiene una función efectuada por los transistores N-Mos y otra función igual efectuada por los P-MOS, lo que lleva a pensar que se desperdicia bastante espacio en la oblea. Esto es especialmente cierto cuando se intenta construir funciones lógicas muy grandes y complejas. Además esto hace que debido a que los transistores se deben hacer más grandes, el circuito sea más lento. [6]

Otro problema, el cual es especialmente importante en circuitos secuenciales con mucha lógica en cascada, existe una “carrera” debido al sesgo que existe entre dos relojes invertidos ( $\phi$  y  $\bar{\phi}$ ). No hay forma de asegurar en un 100 % que los relojes están invertidos en todo momento, esto da pie al problema de que pueden estar ambos relojes en 1 al mismo tiempo por una fracción muy pequeña de tiempo lo que puede llevar a una mala operación del circuito en cuestión debido a un flujo ilegal de datos.[7]

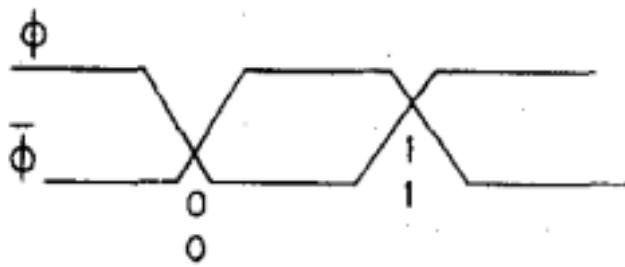


Figura 4: Problema de sesgo entre relojes invertidos. [7]

Para solventar una parte de estos problemas, se propone la lógica CMOS dinámica o Pseudo-NMOS, la cual reduce el área, y elimina el problema de la redundancia entre las funciones del bloque n-mos y p-mos. Sin embargo la desventaja principal de la lógica dinámica es que es imposible conectar circuitos en cascada con el fin lograr una lógica más compleja, esto debido a que se pueden dar nuevamente flujos ilegales de datos dependiendo del sesgo de relojes.

Este problema nuevamente se solventa utilizando lógica dominó, la cual se basa en utilizar lógica dinámica y al conectarla en cascada con otro bloque de lógica dinámica, se conecta un inversor en medio lo que evita el flujo ilegal de datos, pero le quita flexibilidad al diseño y aumenta el tamaño del mismo debido a que la suma del bloque lógico más el inversor da como resultado una señal no invertida. [6]

#### 3.1. $C^2MOS$

Debido a que el carnet universitario del estudiante encargado de realizar este proyecto es B80525, se tiene que la prioridad más alta de diseño es la velocidad y a partir de aquí se busca un balance entre área y consumo de potencia donde la segunda prioridad será el área.

La lógica Clocked-CMOS o bien  $C^2MOS$ , es la perfecta para esta tarea. Empezando porque resuelve todos los anteriores problemas de las familias lógicas mencionadas, incluyendo el sesgo de los relojes.

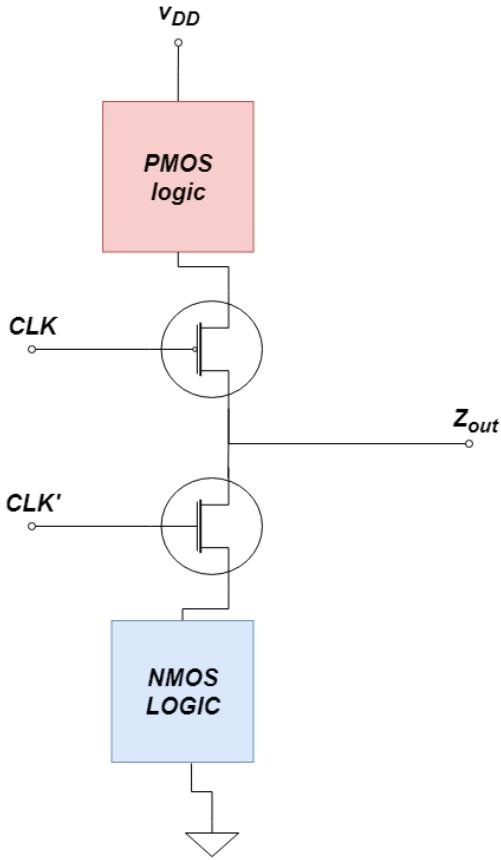


Figura 5: Celda general para la lógica  $C^2MOS$ .(*Creación Propia*)

El funcionamiento del clocked-Cmos es muy interesante y muy simple, la única diferencia física en cuanto a la lógica CMOS tradicional es que ahora los bloques de lógica P-MOS y N-MOS están separados por dos transistores que están controlados por relojes opuestos. Se propone el siguiente ejemplo para comprender el funcionamiento base de esta arquitectura. Cuando se tiene  $\phi = 0$  y  $\bar{\phi} = 1$  ambos transistores de control están encendidos, lo que va a conectar ambas lógicas (n-mos y p-mos) a la salida, y el control de dicha salida va a estar dada por la entrada, lo que a grosso modo hace que funcione exactamente igual a la lógica CMOS tradicional. Sin embargo cuando se tiene el caso opuesto  $\phi = 1$  y  $\bar{\phi} = 0$ , el circuito está apagado y se aísla el nodo de salida de la entrada, lo que hace que esta lógica tenga un comportamiento similar al de un latch. Tomando esto en cuenta y la topología master/slave de la figura 2, se procede a proponer un diseño de flip flop con lógica  $C^2MOS$ . Si bien se puede argumentar que la lógica CMOS tendría dos transistores menos y además consume menos potencia debido a estos mismos dos transistores controlados por el reloj, la lógica CMOS tradicional sufre del sesgo y la “carrera”, de lo cual no sufre  $C^2MOS$ , y como en este proyecto que tiene como prioridad principal la velocidad para esto se usa el dimensionamiento según la teoría del esfuerzo lógico, y la lógica  $C^2MOS$  permite un balance entre consumo de potencia y área, además de hacer el circuito sumamente fiable ya que es anti-carrera y no sufre por el sesgo del reloj. A esto se le llama arquitectura NORA CMOS o bien en inglés No RAce CMOS, y es ampliamente utilizada hasta el día de hoy por compañías como intel o AMD para el diseño de la memoria Caché de sus procesadores, ya que se requiere que estos datos sean lo más precisos posible, además de rápidos. [7]

#### 4. Topología del Flip flop

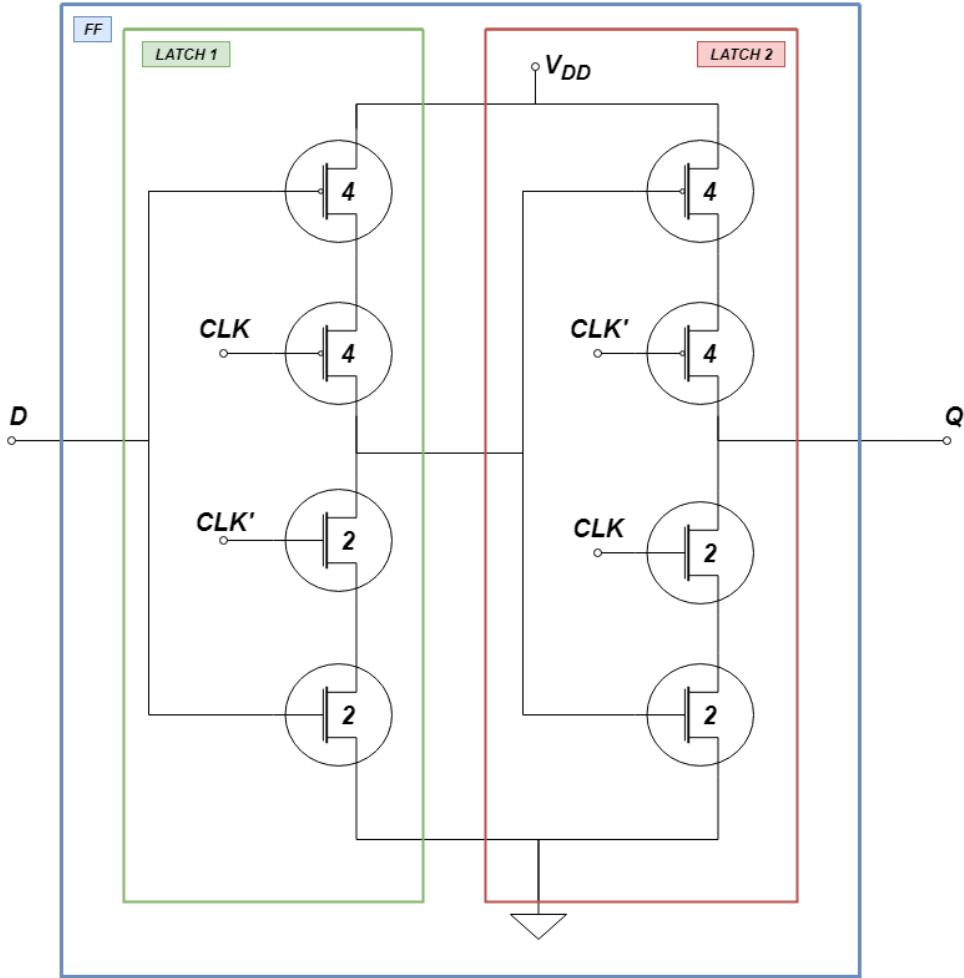


Figura 6: Disposición de dos latches en Master/Slave formados por dos inversores con reloj utilizando lógica  $C^2MOS$  (Creación Propia)

Este circuito funciona muy en el fondo como un flip flop formado por dos latches conectados en Master/slave. Ahora se plantea un ejemplo con el fin de explicar el funcionamiento del flip flop propuesto. Si se tiene que en el primer instante de tiempo  $D = 1$  y  $\phi = 0$  y  $\bar{\phi} = 1$ , el bloque está encendido y como tengo un 1 en la entrada de datos, se enciende el arreglo n-mos y me pone un cero en la salida del master (cuadro verde). Pero como los relojes del slave están invertidos el bloque rojo está apagado y se tiene un estado indeterminado en la salida. Esto es de esperar siempre en un flip flop al encenderse en los primeros ciclos debido a que la lógica secuencial necesita el “pasado” para funcionar. Ahora cuando el reloj cambia  $\phi = 1$  y  $\bar{\phi} = 0$  se apaga el master y se enciende el slave, y como había un cero en el nodo de salida del master esto enciende la red p-mos y pasa un 1. Es decir cuando se tiene un flanco positivo se da la transición y se cumple el funcionamiento esperado del flip flop tipo D.

## 5. Dimensionamiento por esfuerzo lógico:

Como la compuerta en cuestión es asimétrica, no todas las entradas tienen la misma capacitancia de entrada y por ende se debe escoger el camino con el retardo más grande para optimizar mediante la teoría de esfuerzo lógico. En este caso es muy sencillo ya que se trata de una compuerta **not** controlada por relojes. Como es de esperar y a simple vista se puede observar que independientemente de cual arreglo esté encendido el camino más largo es cuando los transistores de los relojes están encendidos y se tiene ya sea un cero o un uno en la entrada de datos. Como se observa al tener un dato en cero el  $v_{dd}$  debe pasar por dos transistores en serie para llegar a la salida. Por lo que es este camino el que se va a optimizar.

1. **Esfuerzo lógico:** Como ambas compuertas son iguales, se encuentra el esfuerzo lógico de la entrada D para una sola y se tiene el de las dos. Como se ve en el esquemático, la entrada D tiene 4 anchos del transistor p-mos y 2 anchos del transistor n-mos, para una capacitancia de entrada en la compuerta igual a 6, ahora hay que dividir entre la capacitancia de entrada de un inversor base, y tomando la movilidad como 2, se sabe de la teoría que el inversor base tiene una capacitancia de entrada igual a 3, por lo que el esfuerzo lógico de una sola compuerta y para la entrada es igual a  $\frac{6}{3} = 2$ . Ahora para el esfuerzo lógico del camino total, se multiplican los esfuerzos lógicos de cada etapa para un total de 4.
2. **Esfuerzo Eléctrico:** Este es muy sencillo, ya que del enunciado se sabe que la capacitancia de salida del camino completo es cien veces mayor a la de entrada lo que deja un valor de  $H=100$ .
3. **Branching Effort:** En este caso no hay ramas, ya que se trata de dos inversores en serie, por lo que la rama es 1, como se observa en la siguiente tabla, la capacitancia  $C_{off, path}$  es cero y se obtiene un único 1 lo cual corresponde al numero de ramas en el circuito en cuestión.
4. **Effort Delay:** Teniendo los resultados anteriores el effort delay de todo el camino se obtiene fácilmente, multiplicando G, B y H. De aquí se obtiene un 400 de effort delay.
5. **Número de etapas:** Este parámetro se puede ver directamente con solo contar la cantidad de compuertas entre la entrada aplicada y la salida deseada.
6. **Esfuerzo mínimo:** Teniendo todo lo anterior basta con aplicar una formula para obtener el esfuerzo mínimo que debe hacer cada etapa para minimizar el retardo.
7. **Capacitancias parásitas:** Este parámetro también es fácil de ver con el circuito, ya que hay una capacitancia parásita en cada difusión en el caso de que hayan dos transistores en serie. En la sección de anexos se muestra en el esquemático todas las capacitancias parásitas y el procedimiento manual.
8. **Path Delay:** Este es el último de los parámetros que se obtienen, debido a que es una fórmula, la más importante, la cual resume todos los resultados anteriores.

En la siguiente tabla se muestra un resumen de todos los los parámetros del esfuerzo lógico resumidos. Además en anexos se puede ver el procedimiento hecho a lapiz.

<b>G</b>	$g_{D1} \cdot g_{D2}$	$2 \cdot 2 = 4$
<b>H</b>	$\frac{C_{OutPath}}{C_{InPath}}$	$\frac{100C_{in}}{C_{in}} = 100$
<b>B</b>	$b_i = \frac{C_{OnPath} + C_{OffPath}}{C_{OnPath}}$	$b_1 \cdot b_2 = 1 \cdot 1 = 1$
<b>F</b>	$GBH$	$4 \cdot 1 \cdot 100 = 400$
<b>N</b>	# of stages	2
$\hat{f}$	$F^{\frac{1}{N}}$	$\sqrt{400} = 20$
<b>P</b>	$p_i + p_{i+1} + \dots + pn$	$3 + 3 = 6$
<b>D</b>	$NF^{\frac{1}{N}} + P$	46

Cuadro 1: Tabla resumen de los parámetros de esfuerzo lógico. (Creación Propia)

Obtenidos los resultados anteriores, para obtener el dimensionamiento, a partir de la fórmula de  $f$  óptimo se despeja la capacitancia de entrada para cada etapa.

$$C_{in} = \frac{g_i C_{cout}}{\hat{f}} \quad (4)$$

y se trabaja de atrás para adelante, ya que solo se conoce la capacitancia de salida.

Entonces para la capacitancia de entrada de la segunda etapa:

$$Y = \frac{2 \cdot 100}{20} = 10$$

A partir de la capacitancia anterior se obtiene la capacitancia de entrada de todo el circuito.

$$X = \frac{2 \cdot 10}{20} = 1$$

Hecho esto se procede a multiplicar los números de dimensionamiento base por un parámetro K, y con las capacitancias anteriores se despeja el nuevo valor del ancho del transistor.

$$\begin{aligned} C_{in_y} &= 4k + 2k \implies 10 = 6k \implies k = \frac{5}{3} \\ C_{in_x} &= 4k + 2k \implies 1 = 6k \implies k = \frac{1}{6} \end{aligned}$$

Con esto se obtiene el dimensionamiento que da el menor retardo para este flip flop por si solo. Aún el circuito se puede optimizar más usando la teoría de esfuerzo lógico, poniendo más etapas para dividir aún más el peso y aumentar la velocidad.

### 5.1. Encontrando el N óptimo:

De la teoría, y de redondeos un poco groseros se obtiene que la expresión para obtener el número óptimo de etapas es la siguiente:

$$N = \log_4 F \quad (5)$$

Sustituyendo los valores de la tabla anterior se obtiene:

$$N = \log_4 400 = 4.32$$

Como no se puede tener una fracción de etapa, se redondea al número par más cercano, el que en este caso es 4.

Teniendo en cuenta que ahora N es 4, se procede a hacer el mismo procedimiento que cuando se diseñó con dos etapas, de hecho los parámetros G,H, B y F se van a mantener iguales, lo que va a cambiar el esfuerzo óptimo debido al aumento en etapas, y por consiguiente también va a cambiar el retardo general de la compuerta ya que ahora se tiene una compuerta más rápida. Además debido a que se están agregando dos inversores las capacitancias parásitas aumentan en dos.

<b>G</b>	$g_{D1} \cdot g_{D2}$	$2 \cdot 2 = 4$
<b>H</b>	$\frac{C_{OutPath}}{C_{InPath}}$	$\frac{100C_{in}}{C_{in}} = 100$
<b>B</b>	$b_i = \frac{C_{OnPath} + C_{OffPath}}{C_{OnPath}}$	$b_1 \cdot b_2 = 1 \cdot 1 = 1$
<b>F</b>	$GBH$	$4 \cdot 1 \cdot 100 = 400$
<b>N</b>	#of stages	4
$\hat{f}$	$F^{\frac{1}{N}}$	$400^{\frac{1}{4}} = 4.47$
<b>P</b>	$p_i + p_{i+1} + \dots + p_n$	$3 + 3 + 1 + 1 = 8$
<b>D</b>	$NF^{\frac{1}{N}} + P$	25.88

Cuadro 2: Tabla resumen para los parámetros de esfuerzo lógico para el mejor diseño. (Creación propia)

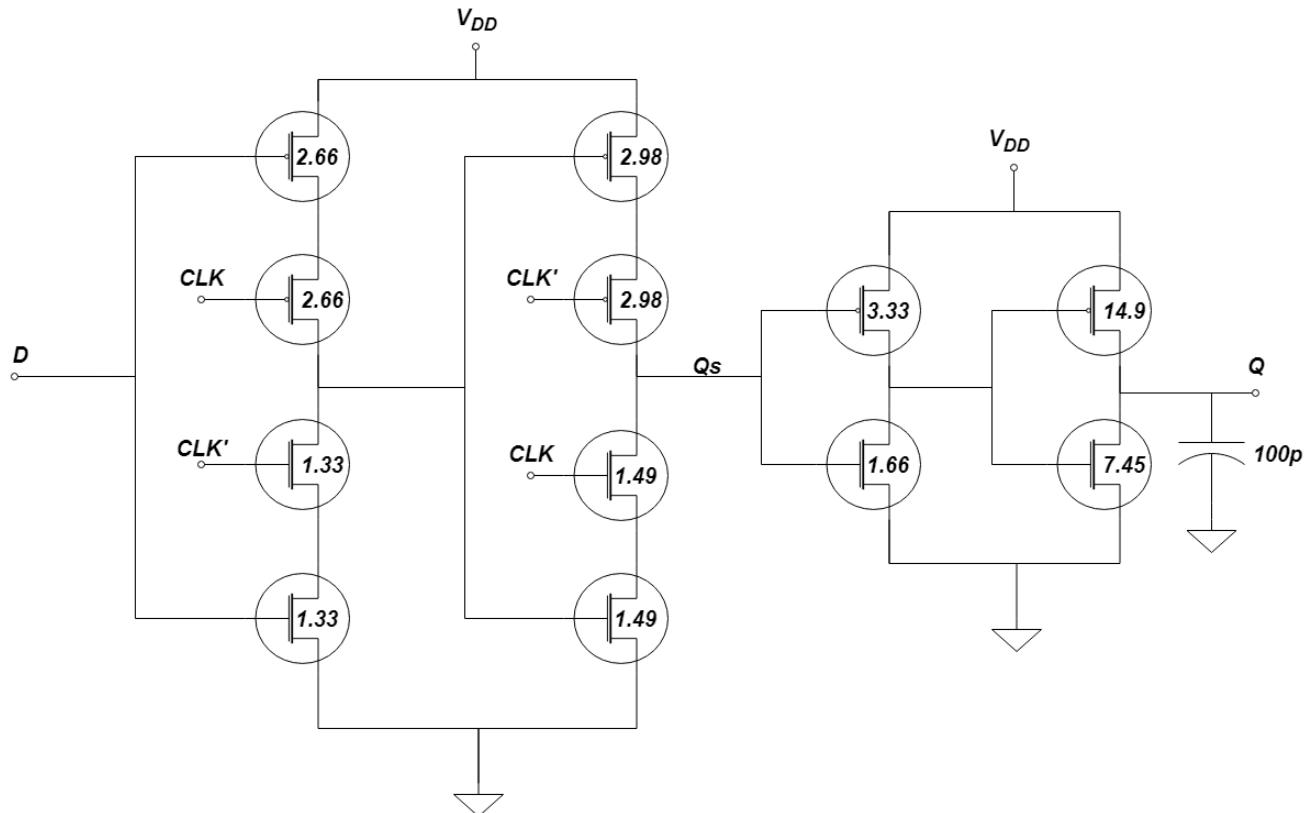


Figura 7: Flip-Flop con óptimo retardo. (Creación Propia)

Ahora se hace la primer simulación con el fin de mostrar que el funcionamiento de ambos flops es correcto y además comprobar que al agregar ambas etapas el circuito se hace más rápido.

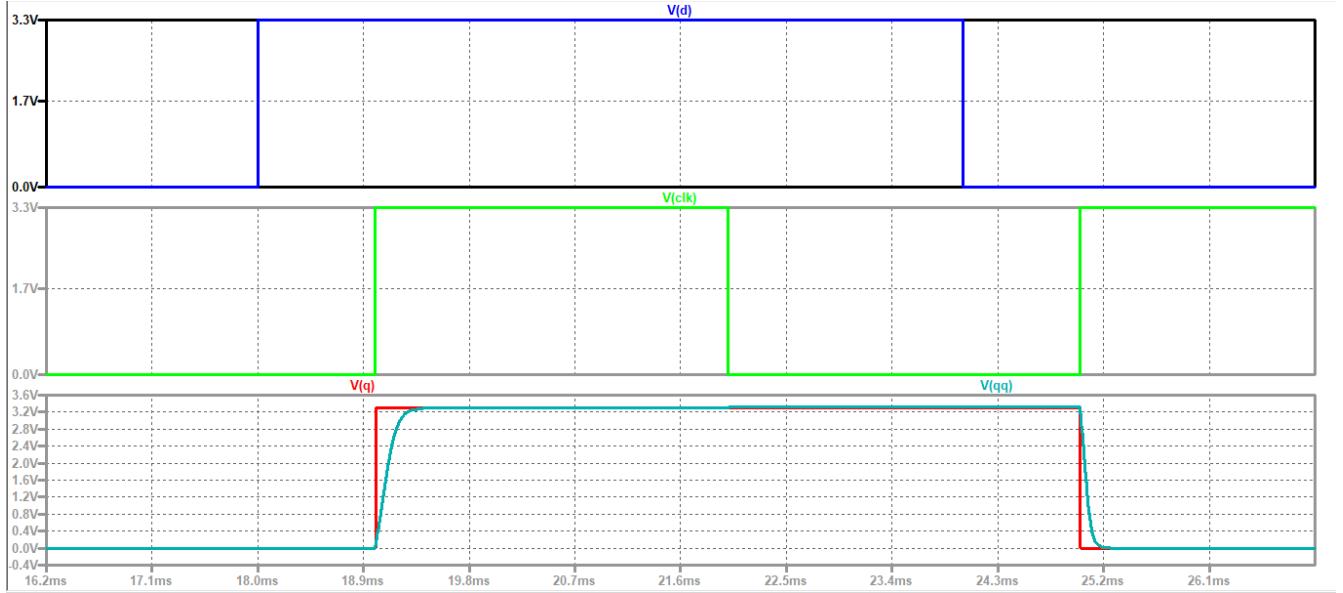


Figura 8: Comparación de las respuestas de ambos flops. (Creación Propia)

En la figura anterior se observa como efectivamente se comporta como un flop controlado por flanco positivo de reloj, y además se comprueba la teoría en la práctica ya que la línea roja se ve casi perfecta a esta frecuencia mientras que la línea celeste la que equivale al circuito de dos etapas, es mucho más lenta.

Si bien para hacer el circuito más rápido se sacrifica un poco el área debido a que se agregaron dos etapas extra, la prioridad de este diseño era la velocidad y se optimizó eso, mientras se mantuvo el área lo más pequeña posible sin que este parámetro afectara la velocidad. Contrastando el diseño propuesto con otros en línea, aún agregando estas dos etapas el circuito sigue siendo bastante pequeño en comparación, además de que si se hubiera tomado un diseño más grande es probable que para optimizar la velocidad del mismo usando esfuerzo lógico, hubiera tenido que agregar etapas haciendo aún más grande el circuito. Por lo que entonces el diseño propuesto sirve bien ante la lista de prioridades, se optimizó la velocidad al máximo utilizando esfuerzo lógico, y el diseño propuesto toma también la cantidad mínima de transistores posible en comparación a otros diseños en línea. Por último el consumo de potencia se deja como última prioridad.

Todos los cálculos realizados para llegar a esta conclusión se discuten en anexos.

## 6. Determinación de los tiempos de reacción del flip flop:

### 6.1. Tiempo de carga:

El tiempo de carga hace referencia al tiempo que le toma a la red p-mos para cargar el capacitor de salida. Para esto se tiene un modelo de retardo simplificado por medio de resistencias y capacitancias.

Primeramente se sabe que la ganancia del transistor está dada por la siguiente expresión:

$$\beta = K_p \cdot \frac{W}{L} \quad (6)$$

Ya todos estos son parámetros conocidos, debido a que la constante del proceso ( $K_p$ ) viene dada en las especificaciones físicas para los parámetros dados por el enunciado, al igual que el largo de los transistores. Para el ancho de los transistores se tomó el valor obtenido durante el dimensionamiento usando esfuerzo lógico.

Entonces como se toman los resultados para el PMOS:

$$\beta = 2.9352 \times 10^{-5} \cdot \frac{14.91\mu}{1.2\mu} = 3.64 \times 10^{-4}$$

Ahora se hace una aproximación un tanto grosera; sin embargo, sirve bien para aproximar el valor de la resistencia equivalente de un transistor.

$$R_{eq} = \frac{1}{\beta}$$

Ahora se sustituyen los valores:

$$R_{eq} = \frac{1}{3.64 \times 10^{-4}} = 2741.9\Omega$$

y ahora bajo el modelo de constantes de tiempo, como se sabe que la capacitancia de salida es de 100pF, se procede a obtener la aproximación del tiempo de carga.

$$\tau_{carga} = 3 \cdot R_{eq} \cdot C_L \quad (7)$$

$$\tau_{carga} = 3 \cdot 2741.9 \cdot 100 \times 10^{-12} = 0.82\mu s$$

### 6.2. Tiempo de descarga:

Ahora para descargar el capacitor de carga, se debe encender la red n-mos, y básicamente se hacen los mismos cálculos anteriores salvo que la constante del proceso y el ancho de los transistores cambia.

$$\beta = 9.6379 \times 10^{-5} \cdot \frac{7.45\mu}{1.2\mu} = 5.983 \times 10^{-4}$$

$$R_{eq} = \frac{1}{5.983 \times 10^{-4}} = 1671.25\Omega$$

$$\tau_{descarga} = 3 \cdot 1671.25 \cdot 100 \times 10^{-12} = 0.5013\mu s$$

## 7. Determinación de la frecuencia de operación máxima:

### 7.1. Estimado de la frecuencia máxima:

Como no se puede obtener directamente la frecuencia máxima de operación, se debe obtener el período primero. Para esto se suman ambos tiempos obtenidos en la sección anterior.

$$T = t_{carga} + t_{descarga} \quad (8)$$

$$T_{min} = 0.82\mu + 0.5\mu = 1.32\mu s$$

Ahora de la relación de período y frecuencia:

$$f = \frac{1}{T} \quad (9)$$

Sustituyendo valores:

$$f = \frac{1}{1.32\mu} = 0.75 Mhz$$

### 7.2. Estimado del consumo de potencia:

Se sabe de la teoría de circuitos lineales que la potencia de consumo se define como:  $P = I \cdot V$ ; sin embargo, por ley de ohm esta expresión tiene otras formas y colores, en este caso como no se trabajan con resistencias, se necesita un modelo de impedancia equivalente, el cual, según la teoría de clase se define con la siguiente expresión:

$$P_{dinámica} = C \cdot V_{DD}^2 \cdot f \quad (10)$$

Ahora sustituyendo los valores de capacitancia, asumiendo una alimentación de 2.5v y usando la frecuencia máxima teórica de operación:

$$P_{dinámica} = 100 \times 10^{-12} \cdot (2.5)^2 \cdot 0.75 \times 10^6 = 4.68 \times 10^{-4} watts$$

Cabe mencionar que aunque los relojes están constantemente cambiando, esto no implica un consumo extra de parte de esos transistores al encenderse ya que no se cierra el camino a tierra o a  $v_{dd}$ . Como se dijo en la sección de diseño, la lógica  $C^2MOS$  solo pone o un 0 o un 1 lógico cuando  $\phi = 0$  y solo en este punto estaría consumiendo potencia dicha etapa. cuando el reloj para a un 1 se apaga la etapa anterior y se enciende la siguiente. Esto en cuanto a los inversores controlados por reloj, el buffer de salida va a estar consumiendo en cada transición dada por el flip flop.

Estos resultados se comprueban en la siguiente sección.

## 8. Comprobación de los resultados obtenidos :

En la siguiente figura se muestra el esquemático del circuito del flop montado en el simulador LTspice.

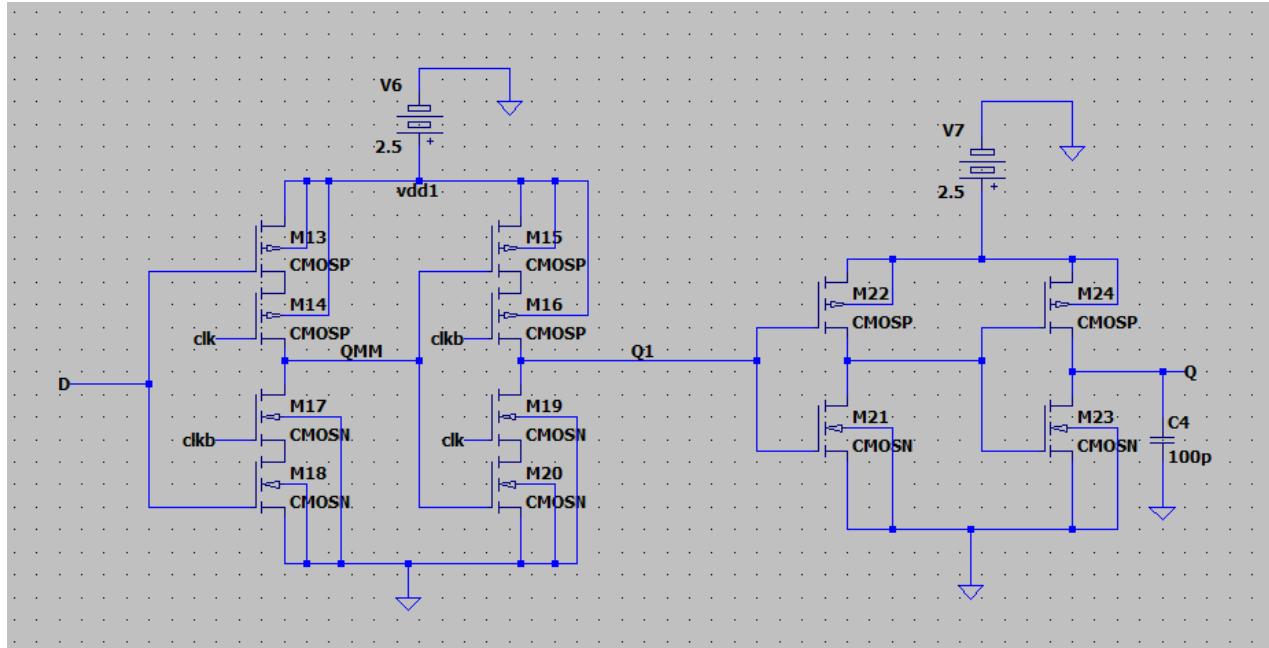


Figura 9: Circuito montado en el simulador LTSPICE.(Creación Propia)

### 8.1. Tiempo de subida:

En la figura se muestra que la diferencia entre ambos punteros, y dice que es un valor de 801ns o bien  $0.801\mu s$ , lo que da como resultado un 2.31 % de error respecto al valor teórico obtenido en la sección anterior.

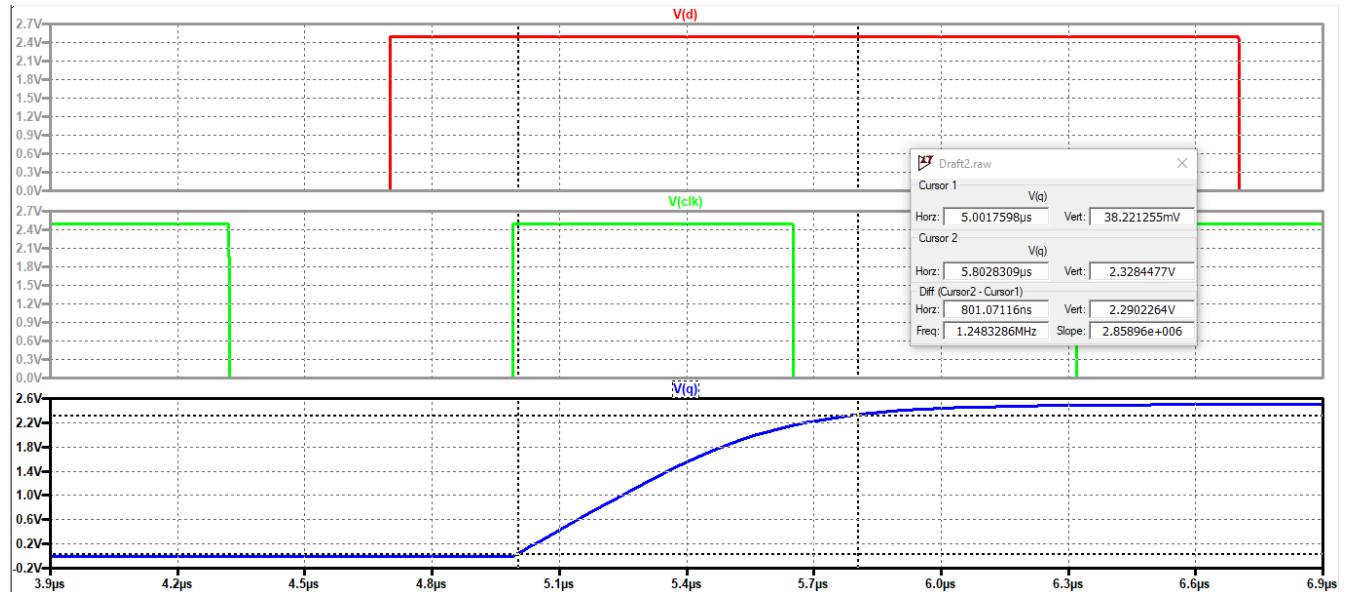


Figura 10: Obtención del tiempo de subida a partir de la respuesta del flip flop. (Creación Propia)

### 8.2. Tiempo de bajada:

En la figura se muestra que la diferencia entre ambos punteros, y dice que es un valor de 507ns o bien  $0.507\mu s$ , lo que da como resultado un 1.13% de error respecto al valor teórico obtenido en la sección anterior.

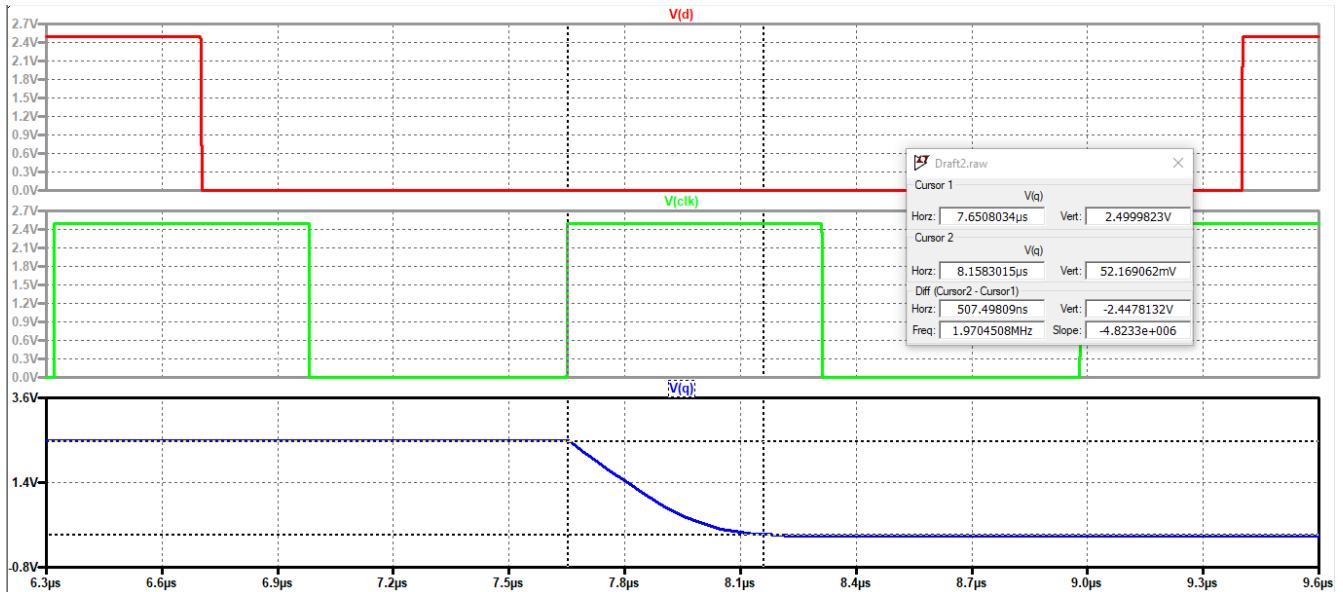


Figura 11: Obtención del tiempo de bajada a partir de la respuesta del flip flop. (Creación Propia)

### 8.3. Frecuencia máxima:

En la siguiente figura se observa como el flip-flop puede operar sin problema a una frecuencia de 0.75mHz, lo cual fue la frecuencia máxima teórica.

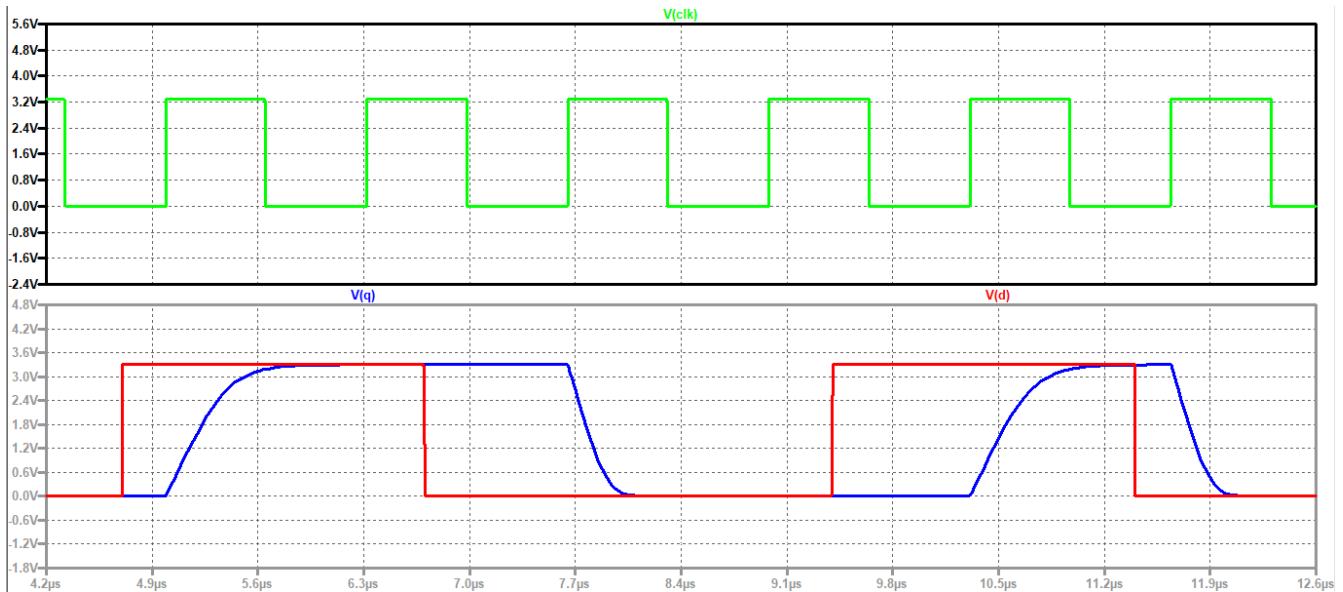


Figura 12: Flop operando en la frecuencia máxima estimada. (0.75MGz) (Creación Propia)

Se hizo una prueba con el fin de confirmar que a una frecuencia de 1mHz, el flop ya no opera de buena forma.

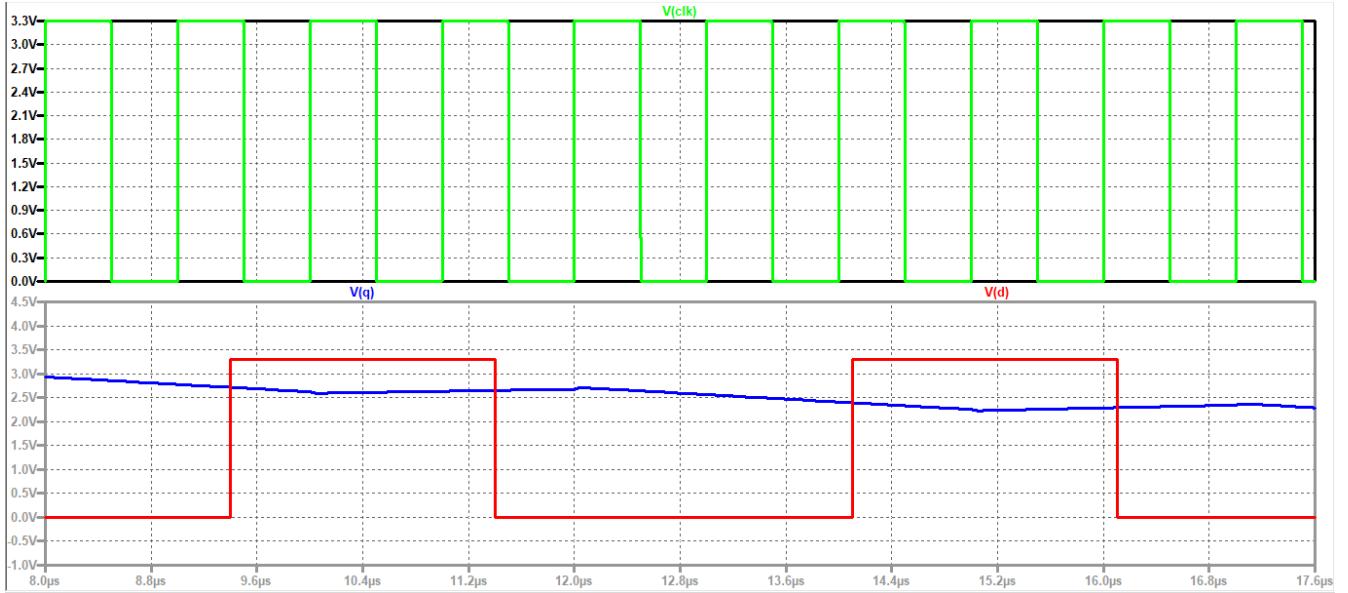


Figura 13: Flop operando a una frecuencia de 1mHz. (Creación Propia)

#### 8.4. Consumo de potencia:

Como se observa en la figura, cuando se da una transición, comienza a fluir corriente para cargar/descargar el capacitor, y esto se toma como un consumo. Como se observa el consumo de corriente en el capacitor es de  $402.29\mu A$ , y la potencia de consumo es de  $505.62\mu W$  lo que al compararlo con el resultado teórico obtenido en la sección anterior, se obtiene un porcentaje de error de 7.66 %.

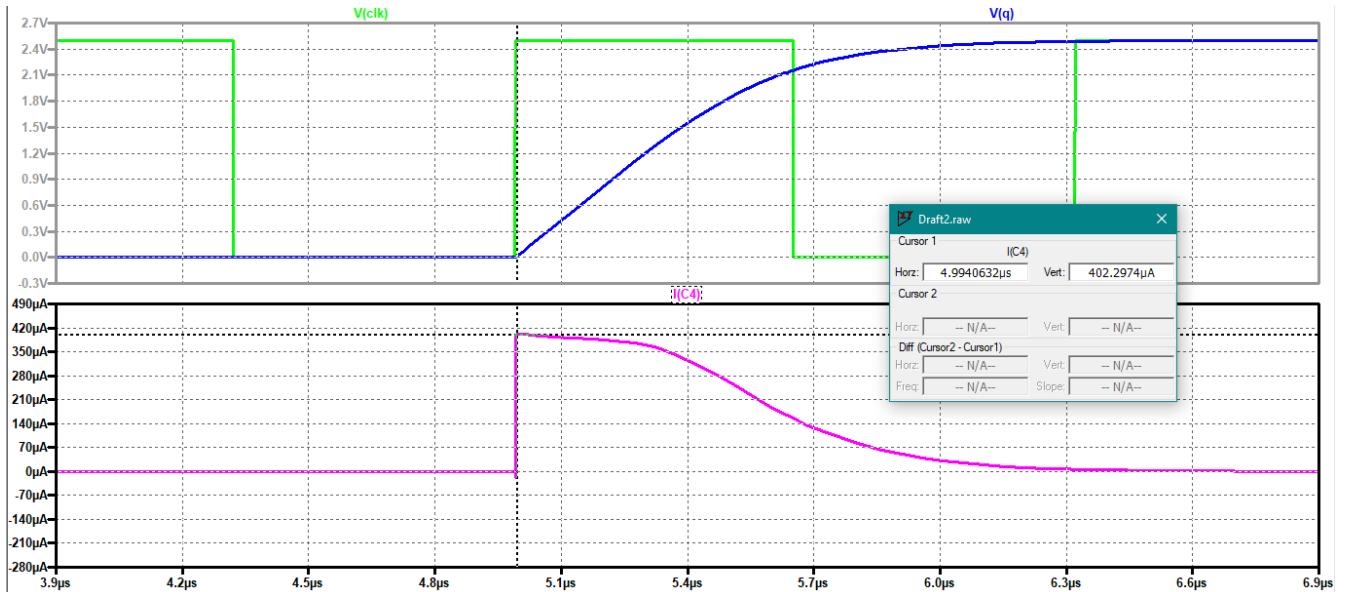


Figura 14: Consumo de corriente para el flip flop operando a 0.75MHz (Creación Propia)

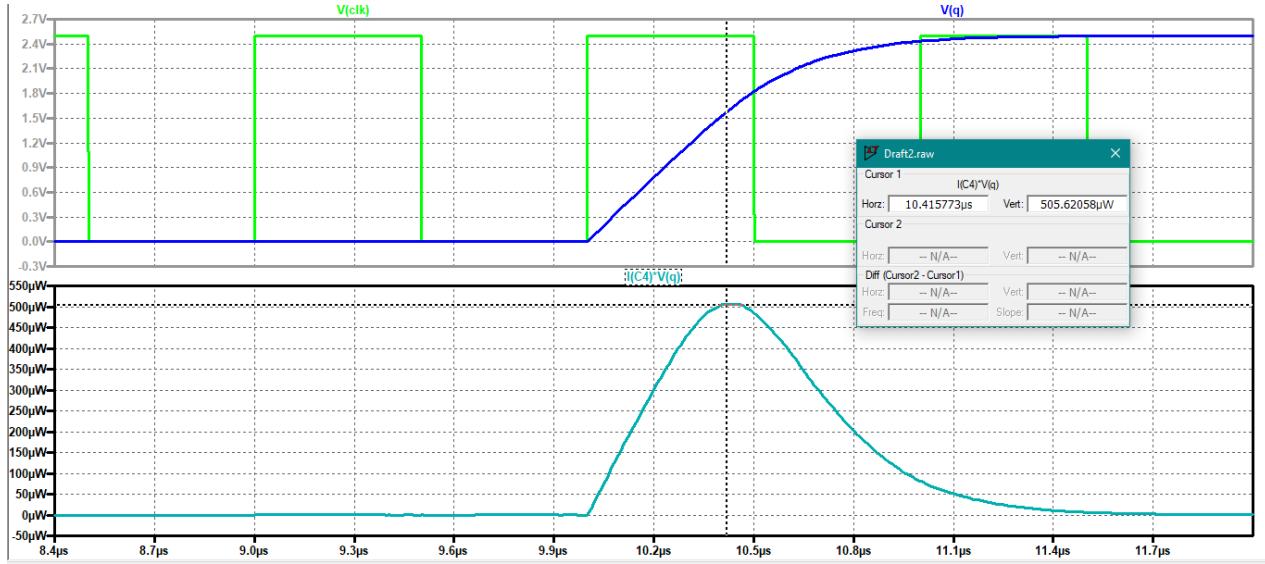


Figura 15: Consumo de potencia para el flip flop operando a 0.75MHz (Creación Propia)

Además queda comprobado que el diseño no consume potencia estática, por lo que si bien la potencia no era una prioridad de optimización, por daño colateral y el diseño escogido, se consume la menor potencia posible.

## 9. Determinación del $\tau$ :

Se sabe de la teoría de esfuerzo lógico que:

$$D = \frac{d_{abs}}{\tau} \quad (11)$$

Ya de la quinta sección, se obtuvo un valor para el parámetro de retardo dado por la teoría de esfuerzo lógico. ( $D = 25.88$ )

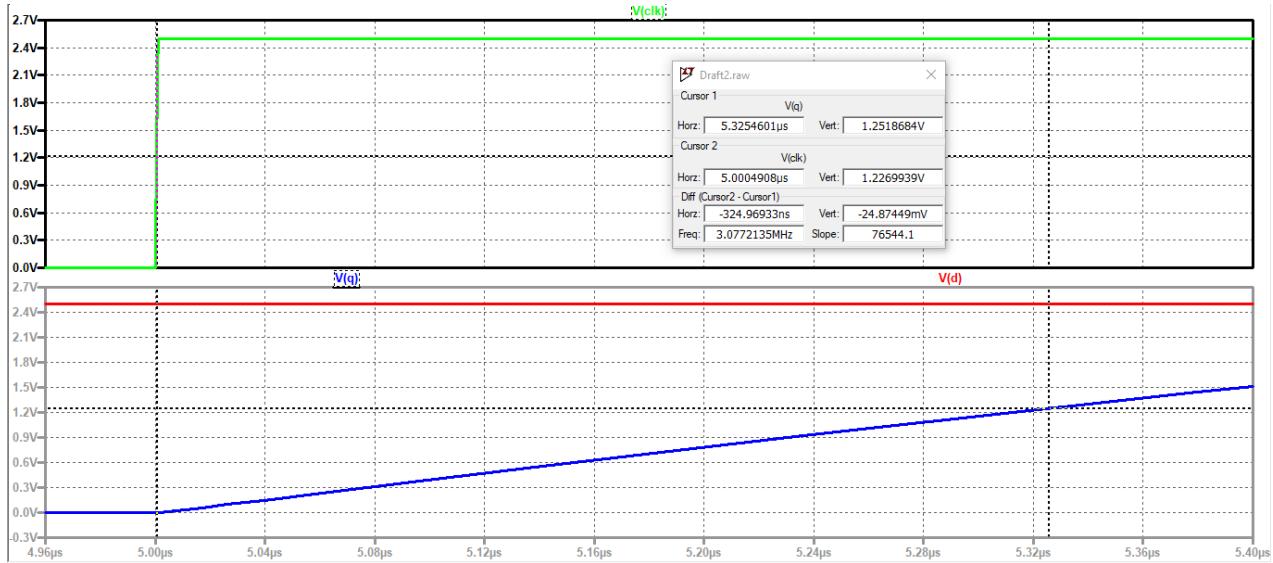


Figura 16: Determinación del  $d_{abs}$  a partir de la simulación. (Creación Propia)

Ahora de la simulación se puede obtener un valor estimado para  $d_{abs}$ .

Se toma el  $d_{abs}$  como el valor de tiempo desde que la entrada cambia y va por un 50 % de su valor final, hasta el tiempo en que la salida va por un 50 % de su valor final.

Ahora se debe analizar, ya que como el flip flop es disparado por flanco positivo de reloj, aunque la entrada de datos D se ponga en 1 una hora antes del flanco, la salida no va a cambiar, por lo que, aunque se optimizó el retardo en el camino de datos, la única forma de que los caminos de datos estén completos y el circuito cierre para que cambie la salida se debe esperar al flanco y a partir de aquí se mide el retardo absoluto.

De la figura anterior, se ve que ambos cursores se encuentran a un 50 % del valor final de 2.5V, y la diferencia entre ambos punteros, es de 324.96ns.

Ahora se hacen los cálculos sustituyendo datos en la fórmula anterior:

$$\tau = \frac{d_{abs}}{D} = \frac{324.96 \times 10^{-9}}{25.88} = 1.2556 \times 10^{-8} s$$

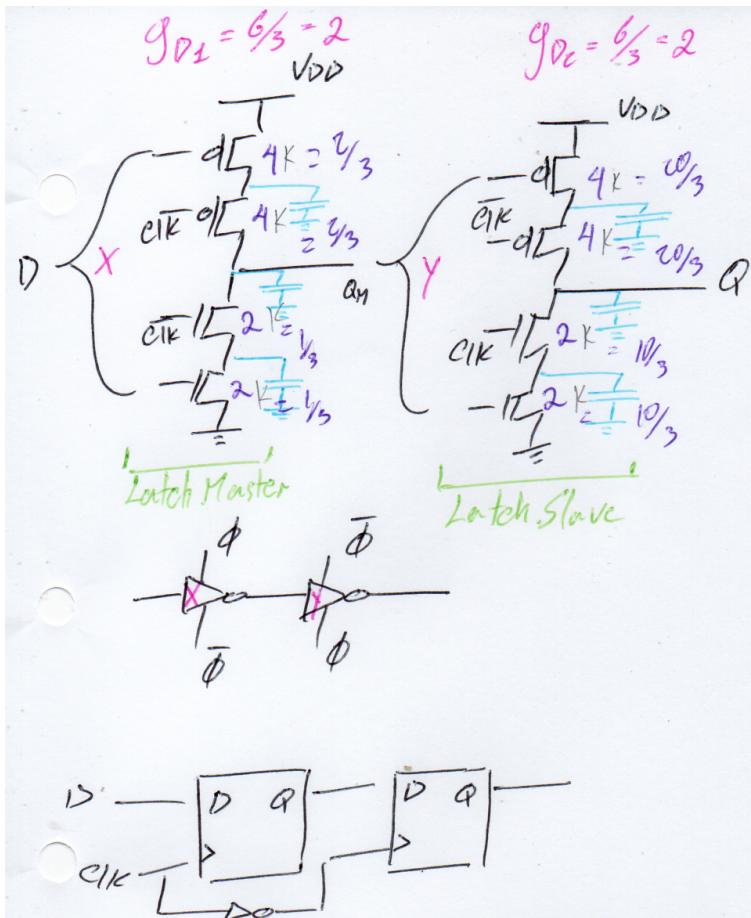
Por lo que entonces se concluye que la constante de tiempo  $\tau$  tiene un valor de:

$$\boxed{\tau = 12.556 ns}$$

## Referencias

- [1] N. Weste and D. Harris. *CMOS VLSI Design*. Addison-Wesley, 4 edition, 2010.
- [2] R. Boylestad and L. Nashelsky. *Electrónica: Teoría de Circuitos y Dispositivos Electrónicos*. Pearson Educación, 10 edition, 2009.
- [3] Adel S Sedra, Dean Emeritus Adel S Sedra, Kenneth Carless Smith, et al. *Microelectronic circuits*. New York: Oxford University Press, 1998.
- [4] Enrique Coen. Videos de las clases impartidas por el profesor este semestre, 2021.
- [5] Ivan Sutherland, Robert F Sproull, Bob Sproull, and David Harris. *Logical effort: designing fast CMOS circuits*. Morgan Kaufmann, 1999.
- [6] A Gago, R Escano, and JA Hidalgo. Reduced implementation of d-type det flip-flops. *IEEE Journal of Solid-State Circuits*, 28(3):400–402, 1993.
- [7] N.F. Goncalves and H. De Man. Nora: a racefree dynamic cmos technique for pipelined logic structures. *IEEE Journal of Solid-State Circuits*, 18(3):261–266, 1983. doi: 10.1109/JSSC.1983.1051937.

10. Anexos



$$b_2 = \frac{\text{Conpath} + \text{Coffpath}}{\text{Conpath}} \rightarrow \text{Solo horay} \quad \pm \text{Amino}$$

$N = 2 \rightarrow$  Dos etapas.

$$C_{in} = \frac{g_i C_{out}}{f}$$

$$X = \frac{2 \cdot 10}{20} = 1$$

$$C_{iny} = 4K + 2K \Rightarrow 10 = 6K \rightarrow K = \underline{\underline{5/3}}$$

$$C_{inx} = 4K + 2K \Rightarrow 1 = 6K \rightarrow K = \underline{\underline{1/6}}$$

$$G = g_{D2} \cdot g_{D2} = 2 \cdot 2 = 4$$

$$H = \frac{\text{Cout path}}{\text{Cin Path}} = \frac{100 \text{ Cin}}{\text{Cin}} = 100$$

$$B = 1$$

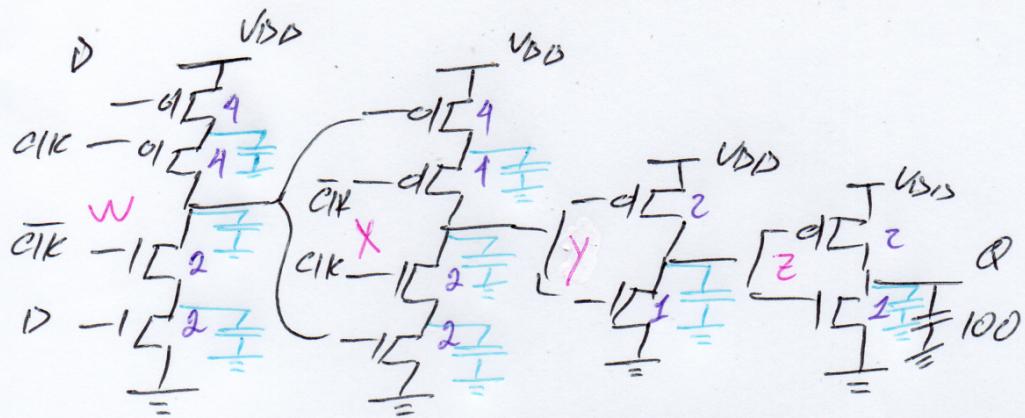
$$F = G B H = 4 \cdot 1 \cdot 100 = 400$$

$$f = F^{\frac{1}{N}} = \sqrt[4]{400} = 20$$

$$P = 3 + 3 = 6$$

$$D = N f^{\frac{1}{N}} + P = 46$$

$$Y = \frac{2 \cdot 100}{20} = 10$$



$$g_{D1} = \frac{6}{3} = 2 \quad g_{D2} = \frac{6}{3} = 2 \quad g_{D3} = \frac{3}{3} = 1 \quad g_{D4} = \frac{3}{3} = 1$$

Dimensionamiento:

$$Y = \frac{1 \cdot 22,37}{4,47} = 5$$

$$C_{in} = \frac{g_i C_{out}}{f}$$

$$X = \frac{4 \cdot 5}{4,47} = 4,47$$

$$Z = \frac{1 \cdot 100}{4,47} = 22,37$$

$$W = \frac{4 \cdot 4,47}{4,47} = 4$$

$\Phi = g_{D1} \cdot g_{D2} \cdot g_{D3} \cdot g_{D4} = 4$
$H = 100$
$B = 1$
$F = \Phi B H = 400$
$f = \sqrt[4]{400} = 4,47$
$P = 8$
$D = 25,88$

$$C_{in Z} = 2K + K \rightarrow \frac{22,37}{3} = K$$

$$C_{in Y} = 2K + K \rightarrow \frac{5}{3} = K$$

$$C_{in X} = 4K + 2K \rightarrow \frac{4,47}{6} = K$$

$$C_{in W} = 4K + 2K \rightarrow \frac{4}{6} = K$$

10.1. NETLIST de Spice:

```

V2 D 0 PULSE(0 2.5 1ns 1ns 1ns 2us 4.7us)
V3 clk 0 PULSE(0 2.5 1u 1n 1n 0.5u 1u)
V4 clk 0 PULSE(2.5 0 1u 1n 1n 0.66u 1u)
M13 vdd1 D P001 vdd1 CMOSP l=1.2u w=2.66u
M14 P001 clk QMM vdd1 CMOSP l=1.2u w=2.66u
M15 vdd1 QMM P002 vdd1 CMOSP l=1.2u w=2.98u
M16 P002 clk 0 Q1 vdd1 CMOSP l=1.2u w=2.98u
M17 QMM clk 0 P003 CMOSN l=1.2u w=1.33u
M18 P003 D 0 0 CMOSN l=1.2u w=1.33u
M19 Q1 clk P004 0 CMOSN l=1.2u w=1.49u
M20 P004 QMM 0 0 CMOSN l=1.2u w=1.49u
V6 vdd1 0 2.5
M21 N002 Q1 0 0 CMOSN l=1.2u w=1.66u
M22 N001 Q1 N002 N001 CMOSP l=1.2u w=3.33u
M23 Q N002 0 0 CMOSN l=1.2u w=7.45u
M24 N001 N002 Q N001 CMOSP l=1.2u w=14.91u
V7 N001 0 2.5
C4 Q 0 100p
.model NMOS NMOS
.model PMOS PMOS
.tran 50u
.MODEL CMOSN NMOS LEVEL=3 PHI=0.600000 TOX=2.1200E-08 XJ=0.200000U
+TPG=1 VTO=0.7860 DELTA=6.9670E-01 LD=1.6470E-07 KP=9.6379E-05
+U0=591.7 THETA=8.1220E-02 RSH=8.5450E+01 GAMMA=0.5863
+NSUB=2.7470E+16 NFS=1.98E+12 VMAX=1.7330E+05 ETA=4.3680E-02
+KAPPA=1.3960E-01 CGD0=4.0241E-10 CGS0=4.0241E-10
+CGB0=3.6144E-10 CJ=3.8541E-04 MJ=1.1854 CJSW=1.3940E-10
+MJSW=0.125195 PB=0.800000
.MODEL CMOSP PMOS LEVEL=3 PHI=0.600000 TOX=2.1200E-08 XJ=0.200000U
+TPG=-1 VTO=-0.9056 DELTA=1.5200E+00 LD=2.2000E-08 KP=2.9352E-05
+U0=180.2 THETA=1.2480E-01 RSH=1.0470E+02 GAMMA=0.4863
+NSUB=1.8900E+16 NFS=3.46E+12 VMAX=3.7320E+05 ETA=1.6410E-01
+KAPPA=9.6940E+00 CGD0=5.3752E-11 CGS0=5.3752E-11
+CGB0=3.3650E-10 CJ=4.8447E-04 MJ=0.5027 CJSW=1.6457E-10
+MJSW=0.217168 PB=0.850000
.backanno
.end

```