

# PRÁCTICA 5: Puertas lógicas NMOS. Simulación con Pspice.

## Objetivos

Estudiar diferentes aspectos de las puertas lógicas NMOS, basadas en transistores Mosfet de acumulación de canal N y resistencias de carga. Para ello se usará la simulación electrónica con Pspice.

## Índice

1. El inversor NMOS
2. Otras puertas lógicas NMOS

## Material

PC y programa de simulación PSpice para Windows. Existe una versión de estudiante en PoliformaT.

## Desarrollo

Ejecute el programa **VMware** y seleccione la máquina virtual *Virtual XP- TCOMP*. Entonces, pulse la opción “*Play Virtual Machine*” para que nos abra una ventana con la máquina virtual en Windows. En este entorno, ejecute el simulador de circuitos en *Inicio/Todos los Programas/Pspice Student/Schematics*. Los archivos creados se guardan *Mis Documentos\prac5*, y se deben borrar al finalizar.

**MUY IMPORTANTE:** cuando guardéis el circuito, poner un nombre de archivo distinto a schematic1, schematic2, etc.

Si al simular tenemos problemas de librería, comprobad que en el schematics, en **Analysis/Library and Include Files** figura al menos la librería **nom.lib\***

# 1. El inversor NMOS

Vamos a simular un sencillo inversor con resistencia  $R_D$ , como el de la figura 1.

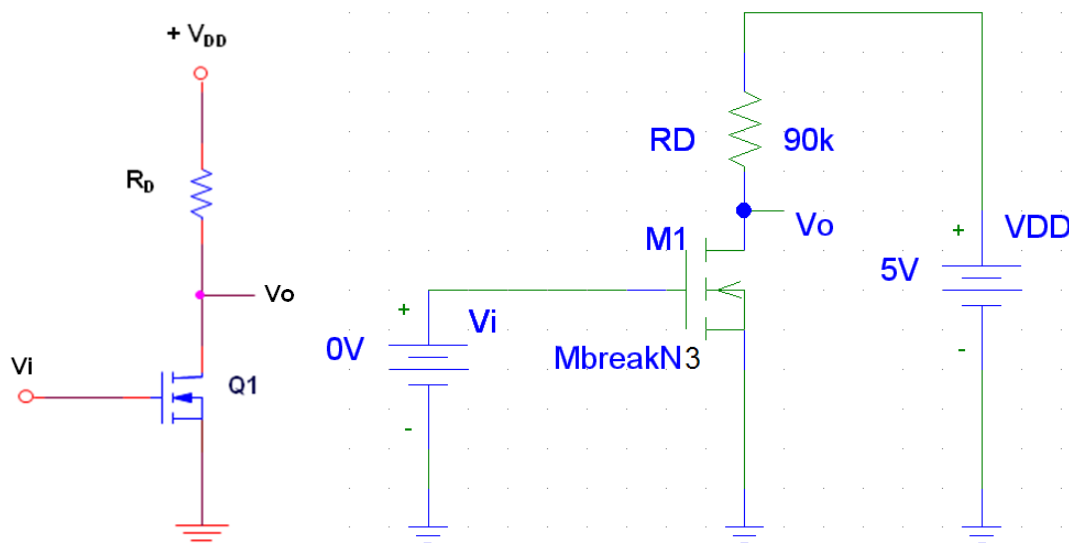


Figura 1. Inversor NMOS con  $R_D$  y su esquema en PSpice.

1.1 Utilizamos el comando **Draw/Get New Part** para obtener los distintos componentes: **VDC** para las fuentes de tensión  $V_i$  y  $V_{DD}$ , **EGND** para la masa y **MbreakN3** para el transistor NMOS. Emplead  $V_{DD}=5V$  y una  $R_D = 90K$ . La salida es sólo un hilo (wire) y una etiqueta de texto **Draw/Text**.

Para especificar los parámetros del transistor, hacemos clic primero en el símbolo del NMOS (se pone de color rojo), y luego usamos el comando **Edit/Model/Edit Instance Model (text)**. Especificad como parámetros:  $K_p = 50u$ ,  $V_{to} = 1V$  en la ventana de edición (ver figura 2). Luego haced doble clic en el símbolo del NMOS y escribid los valores del largo y ancho del canal:  $L = 1u$ ,  $W = 2u$ . Recordemos que la terminación *u* indica *micro* ( $10^{-6}$ ), que  $V_T = V_{to}$  y que  $K = (W/L) K_p / 2$ .

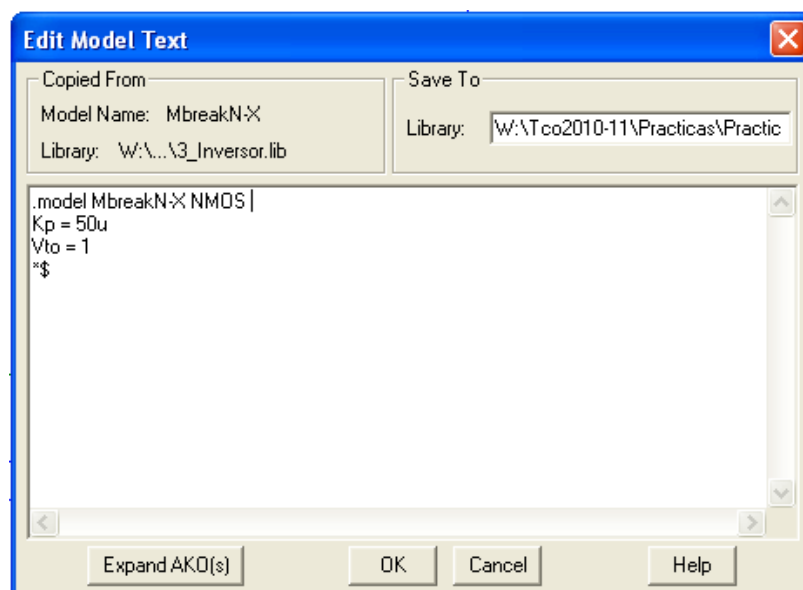



Figura 2. Ventana de edición de modelos.

1.2 Especificad una simulación **DC Sweep** en **Analysis/Setup**. Vamos a generar la **curva de transferencia**  $V_o=f(V_i)$  del inversor. Para ello seleccionaremos  $V_i$  como variable de barrido, y definiremos una variación entre 0 y 5V con incrementos de 0.01V.

1.4 A continuación, utilizad el comando **Analysis/simulate (F11)** para iniciar la simulación. Se activa PROBE y con **Trace/Add trace (Insert)** se selecciona la tensión de salida. La salida está en el drenador del transistor, por lo que una forma de especificarla es  $V(M1:d)$ , donde M1 sería el nombre del transistor (ojo, mirad el nombre del transistor en vuestro esquemático). Debe aparecer la curva de transferencia del inversor, con un aspecto como el que muestra la Figura 3. También podemos verlo poniendo un marcador de tensión  en la D del mosfet.

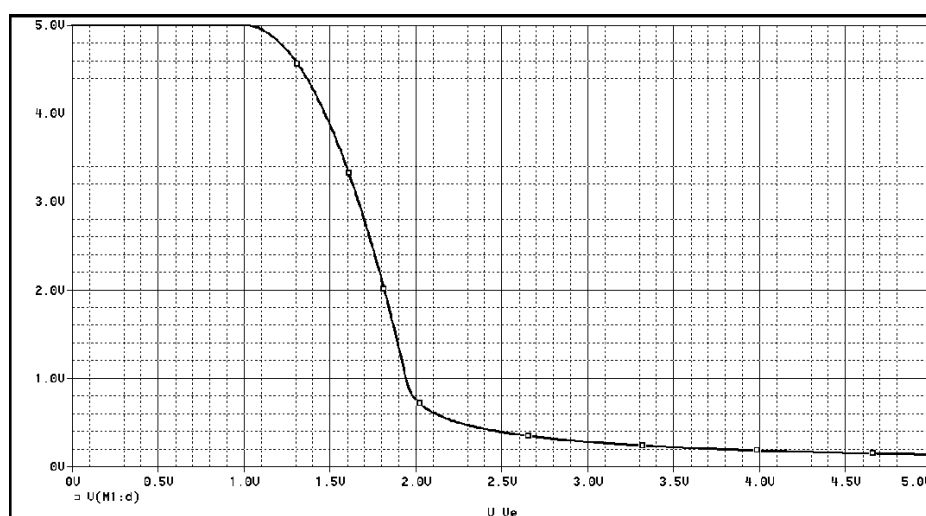


Figura 3. Curva de transferencia del inversor NMOS.

**Cuestión 1:** Medid en la curva de transferencia  $V_{OH}$  y  $V_{OL}$  (es decir, los niveles de tensión de salida a nivel alto y bajo, respectivamente), correspondientes a  $V_i=0V$  y  $V_i=5V$ . Se recomienda usar los **cursores**. La ventana del cursor nos indica estos valores: A1 (valor x) (valor y).

1.4 Representad el consumo de corriente del inversor en función de la tensión de entrada. Para ello, sin salir del PROBE, añadir una nueva gráfica con el comando **Plot / Add Plot to Window**. Seleccionaremos la corriente  $I_D(M1)$  que corresponde a  $I_{DS}$ , y es equivalente a la corriente que suministra la fuente de tensión  $V_{DD}$ . La potencia es el producto  $V_{DD} \times I_D$ , es decir 5V por cada  $I_{DS}$  que se ha medido.

**Cuestión 2:** Medid la corriente con los cursores en los dos estados. ¿En qué estado hay consumo estático de corriente, en el “0” o en el “1” de **salida**? ¿Por qué?

1.5 Volved al *Schematics* y variad el valor de  $R_D$  a 10k y a 200k. Simulad y analizad la influencia en los niveles lógicos de salida y en el consumo de corriente.

**Cuestión 3:** ¿Qué valor de  $R_D$  mejora el nivel lógico  $V_{OL}$ ? Razonad la respuesta a partir del divisor resistivo que se forma entre  $R_D$  y  $R_{ON}$ :

$$V_{OL} = \frac{R_{ON}}{R_{ON} + R_D} \times V_{DD}$$

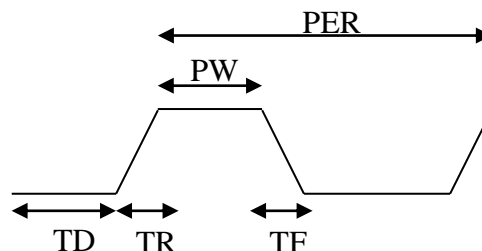
**Cuestión 4:** ¿Qué valor de  $R_D$  reduce (mejora) el consumo? Razonad la respuesta a partir de la corriente que circula por la malla DS:

$$I_{DS} = \frac{V_{DD}}{R_{ON} + R_D}$$


Para visualizar la potencia que disipa el transistor, podemos insertar en **Probe** la siguiente expresión:  $V(M1:d) * ID(M1)$ , que corresponde a  $P = V_{DS} * I_{DS}$ . Se puede observar que entre 1V y 2V la disipación es mayor debido a una  $V_{DS}$  mayor a pesar de una  $I_{DS}$  menor. Esto explica que un transistor Mosfet en conmutación se calentará más en las transiciones.

1.6 Queremos ahora ver la **respuesta en el tiempo** del inversor. Introduciremos una onda cuadrada en la entrada y analizaremos la salida. Sustituiremos la fuente **VDC** de entrada ( $V_i$ ) por una fuente de tipo **VPULSE**. Restablecer  $R_D = 90k$ . Especificad los siguientes parámetros:

- $V1 = 0V$  (nivel bajo)
- $V2 = 5V$  (nivel alto)
- $TD = 10n$  (tiempo inicial)
- $TR = 2n$  (tiempo del flanco de subida)
- $TF = 2n$  (tiempo del flanco de bajada)
- $PW = 40n$  (duración del nivel alto)
- $PER = 100n$  (duración del período)



No cambiar el resto de parámetros.

Situad sendos marcadores de tensión (Voltage/Level Marker)  en la entrada y en la salida.

1.7 Seleccionad una simulación de tipo **Transient** y desactivar la tipo DC Sweep, mediante el comando **Analysis/Setup/Transient**. En la ventana de Transient indicad un **Print step** = 1n, **Final time**=120n. *Print step* y *Final time* especifican la resolución temporal y la duración total de la simulación, respectivamente.

1.8 Simulad con **Analysis/Simulate**. Automáticamente aparecerán las señales de entrada y salida.

**Cuestión 5:** ¿Cómo es la señal de salida respecto a la de entrada? ¿Cuál es la función lógica del circuito?

1.9 Conectad un condensador de **0.1pF** entre la salida y masa (ver Figura 4). Este condensador parásito modeliza las capacidades del transistor y del cableado de salida. Variad  $R_D$  (10k, 90k, 200k) y repetir el análisis Transient anterior, visualizando la señal de entrada y salida. Observad el **retardo** de la señal de salida

respecto de la señal de entrada. La **Figura 5** ilustra la medida del retardo de las dos transiciones en un caso general.

**Cuestión 6:** ¿Qué le ocurre al retardo al aumentar  $R_D$ ? ¿En cuál de los 2 retardos se nota más la influencia de  $R_D$ : en la transición  $L \rightarrow H$  ( $t_{pLH}$ ) o en la transición  $H \rightarrow L$  ( $t_{pHL}$ ) de la señal de **salida**? Mídase con los cursores ambos retardos para 90k.

Se observa que el retardo  $t_{pLH}$  aumenta al aumentar  $R_D$ . Esto es debido a que en la transición  $L \rightarrow H$  el condensador se carga a través de  $R_D$  y aumenta su tensión exponencialmente hacia  $V_{DD}$ , según la conocida ecuación de carga:  $V_s = V_{DD}(1 - e^{-t/R_D C_1})$ . La constante de tiempo de carga es proporcional al producto  $R_D C_1$ .

Sin embargo, en la transición  $H \rightarrow L$  el condensador se descarga por el transistor Mosfet, que funciona en zona lineal con una resistencia equivalente  $R_{on}$  mucho más pequeña que  $R_D$ . La ecuación de descarga es:  $V_s = V_{DD}e^{-t/R_{on} C_1}$ . La constante temporal de descarga es proporcional al producto  $R_{on} C_1$ . Como  $R_{on} \ll R_D$ , el retardo  $t_{pHL}$  es menor que el  $t_{pLH}$  y no depende de  $R_D$ .

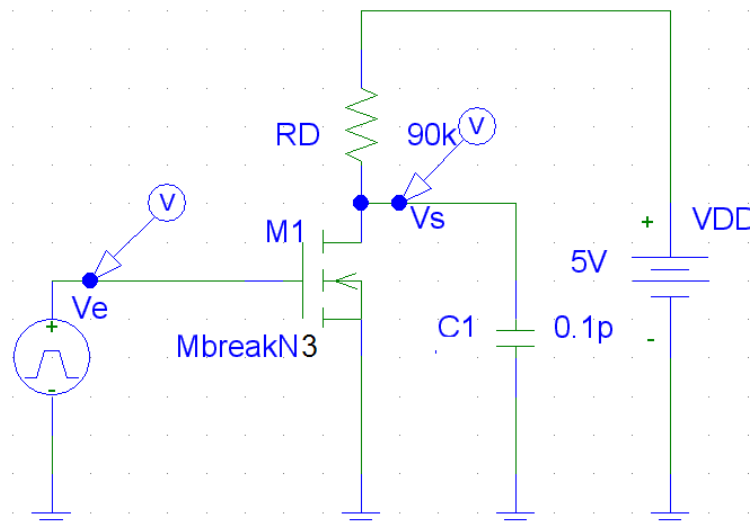


Figura 4. Inversor con condensador parásito.

**Cuestión 7:** ¿Por qué se dice que el valor de  $R_D$  se elige buscando un compromiso entre consumo, niveles lógicos y velocidad?

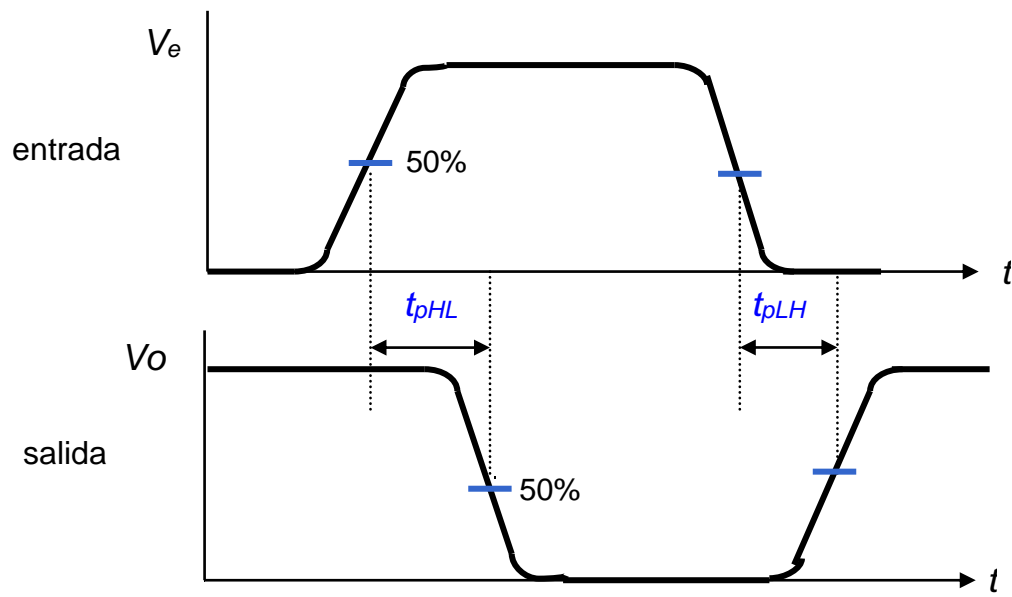


Figura 5. Retardos de la señal de salida respecto a la señal de entrada.

## 2. Otras puertas lógicas NMOS

Vamos a simular ahora una puerta NMOS, tal como la de la Figura 6.

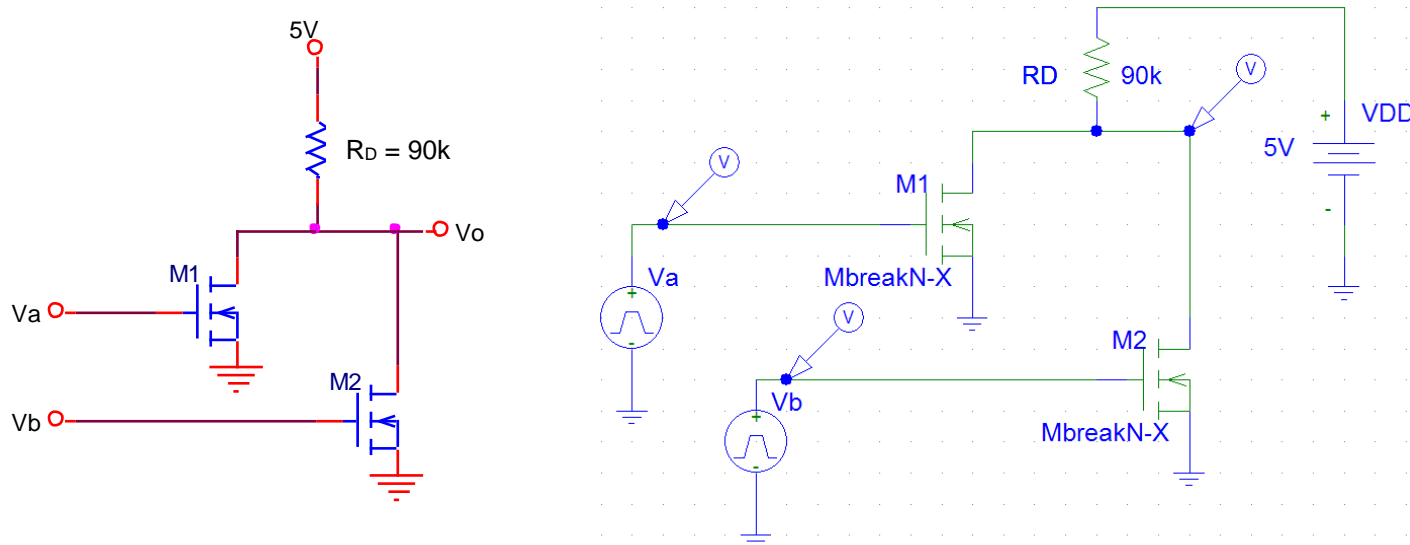


Figura 6. Puerta NMOS de dos entradas.

2.1 Editad el circuito en el *Schematics*. Parámetros de los dos mosfets: **Kp = 50u**, **Vto = 1V**, **L = 1u**, **W = 2u**. En Va y Vb conectad dos fuentes **VPULSE**. Una con los mismos parámetros del circuito inversor (ver punto 1.6). Modificad la otra para conseguir las 4 combinaciones lógicas en Va y Vb: 00, 01, 11, 10. Por ejemplo, desplazándola 20n (los mismos parámetros excepto **TD=30n**).

2.2 Efectuad un análisis tipo **Transient**, con **Print step = 1n**, **Final time=100n**

En Probe, para visualizar las tres señales (dos entradas y una salida) sin solaparse se añadirá una gráfica para cada una de ellas (**Plot/Add plot to Window**) y luego se selecciona en cada una la señal correspondiente: V(Va:+) , V(Vb:+) y V(M1:d).

**Cuestión 8:** ¿Qué función lógica realiza?

**Cuestión 9:** Medid con los cursores el valor de Vo para las combinaciones de entrada 01, 11 y 10. ¿Notáis alguna diferencia? Justificad el resultado teniendo en cuenta que el transistor NMOS funciona como una resistencia  $R_{on}$  cuando la entrada es “1”.

**ATENCIÓN**, antes de apagar el ordenador: borrad cualquier archivo que se haya guardado en local (Mis Documentos, Escritorio, etc.)