

DISEÑO DE CIRCUITOS INTEGRADOS DIGITALES A NIVEL TRANSISTOR: UN ENTORNO CON DIS_CMOS

M. Bataller, J. Guerrero, A. Rosado, J.R. Magdalena y J. Espí.
Universidad de Valencia. Facultad de Física. Dpto. Informática y Electrónica.
Doctor Moliner 50, 46100 Burjasot (Valencia).
Tfno: 96-3864300-Ext. 3316.
Fax: 96-3864568.
E-mail: Manuel.Bataller@uv.es

RESUMEN.- En este trabajo se describe el entorno DIS_CMOS, herramienta informática que está siendo desarrollada y cuya finalidad es la de servir al alumno como complemento y ayuda en el aprendizaje de los conceptos relacionados con el diseño de circuitos integrados digitales a nivel transistor:

1.- INTRODUCCIÓN

A la hora de realizar el desarrollo de un circuito integrado digital es necesario considerar diferentes modelos de comportamiento que describen en todo momento, desde diferentes ópticas, el proyecto de diseño que se aborda. En efecto, se parte de un modelo de alto nivel (descripción de la función del circuito) y se termina con un modelo de bajo nivel o modelo geométrico, que consiste en la elaboración de los planos geométricos o de máscaras a partir del ensamblado de rectángulos y polígonos respetando determinadas reglas de diseño, pasando por las siguientes etapas, a saber: descomposición del circuito según un modelo de tipo booleano, realización del mismo a partir de redes de transistores de tipo MOS utilizando para éste un modelo de interruptor y, finalmente, dimensionamiento de los transistores, término que afecta a la obtención de la geometría, longitud y ancho del canal, según un modelo eléctrico adecuado.

El presente trabajo se centra en la etapa de síntesis de un circuito mediante redes de transistores MOS y la utilización de su correspondiente modelo de interruptor. Cuando se desea diseñar un circuito digital combinacional o secuencial a nivel de puertas lógicas, se acude a los métodos habituales descritos en la abundante literatura existente sobre circuitos digitales, en los que se utilizan conceptos tan conocidos como minitérminos, maxitérminos, métodos de simplificación de Karnaugh y Quine-McCluskey, etc. Si lo que se pretende es diseñar un circuito digital con transistores MOS pero a nivel de transistor y no de puertas lógicas, o lo que es lo mismo, obtener un diagrama del número de transistores necesarios para para la implementación de un determinado diseño, del tipo de transistores (NMOS o PMOS), de las variables asociadas a los terminales de drenador, puerta y surtidor, así como de las conexiones de los mismos resulta necesario adaptar los métodos utilizados en la síntesis lógica de circuitos. En efecto, los métodos utilizados en este entorno parten de los modos de representación clásicos de los circuitos lógicos y conducen directamente a los esquemas de los transistores sin abordar cuestiones relacionadas con la geometría de los transistores (longitud y ancho del canal) que inciden directamente en la velocidad de funcionamiento [1].

2.- DESCRIPCION DEL ENTORNO

Aún cuando los conceptos relacionados con la síntesis de circuitos MOS a nivel de transistor los adquiere el estudiante a partir de las clases de teoría, es en las clases de laboratorio donde el alumno profundiza en éstos desarrollando los conocimientos adquiridos con la ayuda del entorno DIS_CMOS. Para automatizar la etapa de síntesis se desarrolla el entorno anterior, basado en el lenguaje de programación C [2], que tiene como aplicación principal, tal y como se ha mencionado anteriormente, la de proporcionar soporte al estudiante en el aprendizaje de las técnicas de diseño de circuitos con transistores de efecto de campo y cuyo lugar de utilización, un laboratorio docente, obliga a que reúna, entre otras, las siguientes características:

- Que sea un entorno interactivo y de manejo sencillo.
- Que incorpore una presentación visual en formato de ventanas, un menú de ayuda general sobre diferentes términos relacionados con el diseño de circuitos MOS, incluyendo instrucciones de manejo, y una línea de ayuda situada en la parte inferior de la ventana que ofrezca al usuario comentarios relacionados con las opciones elegidas en cada uno de los menús en los que se ha dividido este entorno.
- Que incorpore un manual de usuario que incluya un tutorial con explicaciones paso a paso de la herramienta y del diagrama de flujo seguido para realizar un diseño.
- Que permita el uso de cursores y/o combinaciones de teclas para el desplazamiento a través de los diferentes menús.
- Que sea fácilmente ampliable, o lo que es lo mismo, que se obtenga un código fuente estructurado mediante la utilización de procedimientos, funciones, etc.

La ventana de trabajo correspondiente a la herramienta informática de diseño DIS_CMOS, tal y como se muestra en la Figura 1, consta de los siguientes menús, que se describen a continuación.

2.1.- Menús del entorno DIS_CMOS

El primer menú, que consta de cinco opciones, se llama Archivo (en negrilla la tecla que permite entrar en el mismo), y se encarga de realizar todas las operaciones habituales relacionadas con ficheros, entre las que se pueden citar, junto con la línea de ayuda situada en la parte inferior de la ventana, las siguientes:

- i) **Abrir**: abre el fichero de un diseño a nivel de transistor.
- ii) **Guardar**: guarda el fichero de un diseño a nivel de transistor.
- iii) **Guardar como**: guarda un fichero y le asigna nombre.
- iv) **Imprimir**: imprime un fichero.
- v) **Salir**: finaliza el programa.

El segundo menú llamado **Entrada** dispone de cuatro opciones que permiten al usuario introducir los datos para la síntesis digital del circuito deseado. Estas son las siguientes:

- i) **Minitérminos**: introduce los minitérminos a implementar.
- ii) **Maxitérminos**: introduce los maxitérminos a implementar.
- iii) **Función lógica**: introduce la función lógica a implementar.
- iv) **Tabla de verdad**: introduce la tabla de verdad a implementar.

El tercer menú, **Selección**, contiene diez opciones que se corresponden con las posibles estructuras, [1] y [3], soportadas por el programa DIS_CMOS. En efecto, lo primero que hay que escoger, de entre todas las posibles soluciones existentes, a la hora de realizar el diseño de un circuito digital que implemente una función lógica determinada es la configuración

- básica. El programa DIS_CMOS permite trabajar con las siguientes configuraciones básicas:
- i) Lógica CMOS: obtiene el esquema a transistores con lógica CMOS.
 - ii) Transistores de paso: obtiene el esquema mediante transistores de paso.
 - iii) Circuitos de transmisión: obtiene el esquema utilizando circuitos de transmisión.
 - iv) Lógica Pseudo-nmos: opción de lógica pseudo-nmos.
 - v) Circuitos precargados a 1: esquema mediante el concepto de circuito precargado a 1.
 - vi) Circuitos con ramas de transmisión precargados a 1: esquema utilizando circuitos de transmisión precargados a 1.
 - vii) Circuitos precargados a 0: esquema mediante el concepto de circuito precargado a 0.
 - viii) Circuitos con ramas de transmisión precargados a 0: esquema utilizando circuitos de transmisión precargados a 0.
 - ix) Circuitos secuenciales estáticos: diseño a partir de circuitos secuenciales estáticos.
 - x) Circuitos secuenciales dinámicos: diseño a partir de circuitos secuenciales dinámicos.

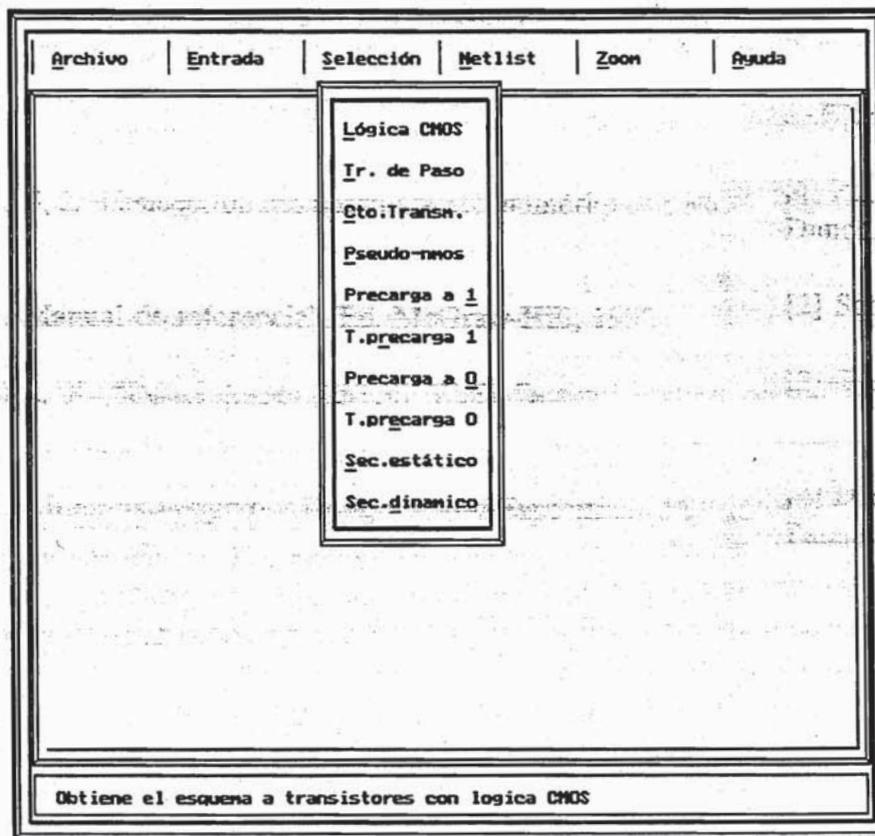


Figura 1.- Menú principal del entorno DIS_CMOS.

El cuarto menú, que consta de cuatro opciones, se llama Netlist y se encarga de realizar operaciones relacionadas con ficheros de listas de conexiones en formato Pspice tales como:

- i) Abrir: abre un fichero de lista de conexiones.
- ii) Guardar: guarda un fichero de lista de conexiones.
- iii) Guardar como: guarda la netlist y le asigna un nombre.
- iv) Imprimir: imprime un fichero de lista de conexiones.

El quinto menú, Zoom, permite realizar operaciones de zoom sobre el esquemático a transistores obtenido a partir de una de las diez opciones anteriores. Sus opciones son las siguientes:

i) **Zoom in:** acerca la zona de visualización.

ii) **Zoom out:** aleja la zona de visualización.

El sexto y último menú, **Ayuda**, es el relacionado con las ayudas del programa al usuario. Contiene dos opciones, a saber:

i) **Contenido:** muestra los temas de ayuda disponibles.

ii) **Acerca de:** indica la versión del programa.

Los resultados de salida procedentes del entorno desarrollado son de tipo gráfico y/o numérico. En efecto, el estudiante dispone de las siguientes opciones:

- Representación gráfica del circuito a nivel de transistor indicando el tipo, P o N, las variables de puerta asociadas a cada transistor, el número total de transistores y el número de transistores asociado al camino de conducción más largo, que permite comparar diferentes estructuras en cuanto a su velocidad de funcionamiento. Un ejemplo en el que se muestra todo lo que se acaba de decir se observa en la figura 2, que corresponde al esquema a transistores de la función AND-OR-INVERT ($f(A,B,C,D)=\overline{(A.B+C.D)}$) utilizando lógica CMOS, o lo que es lo mismo, implementada a partir de la opción i) del menú de Selección.

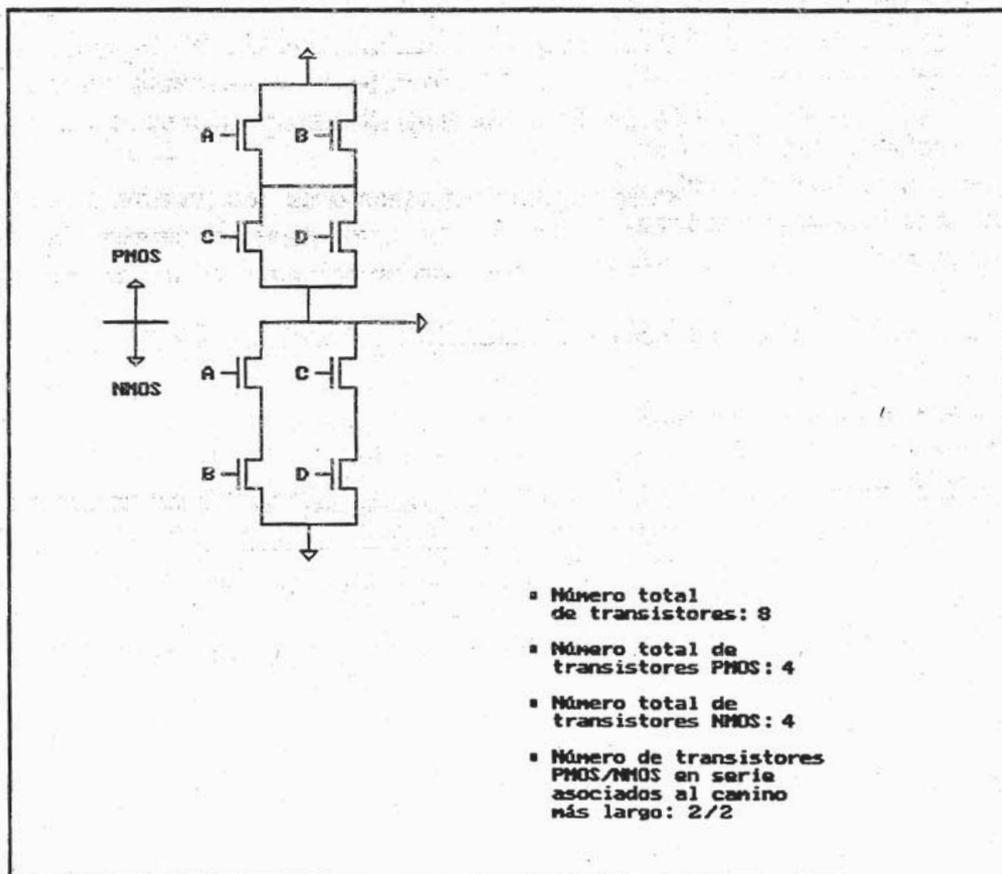


Figura 2.- Esquema a transistores de la función AND-OR-INVERT.

- Obtención de la Netlist en formato PSPICE que permite realizar la simulación del circuito obtenido. En la figura 3 se muestra la Netlist correspondiente a la función AND-OR-INVERT implementada anteriormente.

```

=====
Netlist (PSPICE)
=====

* Modo 3 es "A"
* Modo 4 es "B"
* Modo 6 es "C"
* Modo 7 es "D"
* Modo 5 es "salida"
MP1 1 3 2 1 TP
MP2 1 4 2 1 TP
MP3 2 6 5 1 TP
MP4 2 7 5 1 TP
MN1 5 3 8 0 TN
MN2 8 4 0 0 TN
MN3 5 6 9 0 TN
MN4 9 7 0 0 TN
.MODEL TN NMOS LEVEL=2 VTO=0.75 KP=72.000E-6 L= 1.00U W= 3.00U
.MODEL TP PMOS LEVEL=2 VTO=-1.10 KP=31.000E-6 L= 1.00U W= 3.00U
.TRAN 10PS 20.00N
.END

```

Figura 3.- Netlist PSPICE de la función AND-OR-INVERT.

Tal y como se ha comentado anteriormente, a partir de los resultados anteriores, el estudiante dispone, también, de la posibilidad de impresión del esquema obtenido, así como del almacenamiento del mismo en unidades de disco.

3.- APLICACIONES

Utilizando como soporte este entorno, el alumno puede experimentar en el laboratorio con las configuraciones básicas a partir de las cuales se generan las células con las que se diseñan los circuitos integrados digitales haciendo hincapié en las estructuras que, con más frecuencia, aparecen en las bibliotecas propuestas por los fabricantes de semiconductores [4]. En efecto, como parte integrante de las prácticas de laboratorio se obtienen diferentes configuraciones básicas de transistores MOS correspondientes a multiplexores, circuitos aritméticos y biestables, estructuras que se simulan a partir de la netlist en formato PSPICE que proporciona el programa. Una vez se ha obtenido el diagrama a nivel de transistor, se ha simulado y se ha obtenido la mejor estructura, se utiliza un programa de diseño microelectrónico que permite al estudiante realizar el layout correspondiente, experimentar con la geometría de los transistores (longitud y ancho del canal) y verificar las reglas de diseño.

Una segunda aplicación del entorno se relaciona con la posibilidad de comprobar la correcta síntesis de circuitos que el alumno previamente ha realizado sobre el papel, sirviéndole, por tanto, como ayuda y soporte a las explicaciones impartidas en las clases de teoría.

4.- CONCLUSIONES

Se ha presentado una herramienta, de uso docente, de diseño de circuitos digitales con transistores MOS cuya finalidad es la de servir al alumno como complemento y ayuda en el aprendizaje de los conceptos relacionados con el diseño de circuitos integrados digitales a nivel transistor.

El entorno, implementado sobre PC, es interactivo, de manejo sencillo, abierto (fácilmente ampliable), con un menú de ayuda general sobre diferentes términos relacionados con el diseño de circuitos MOS, que incluye instrucciones de manejo y una línea de ayuda situada en la parte inferior de la ventana. Asimismo, se ofrece un manual de usuario que incluye un tutorial con explicaciones paso a paso de la herramienta y del diagrama de flujo seguido para realizar un diseño.

Es una herramienta adicional, que se complementa con las existentes en cualquier laboratorio docente de diseño VLSI, que acerca al estudiante los conceptos relacionados con el diseño a nivel transistor de circuitos digitales y facilita la implementación posterior del layout de los mismos, aproximando al estudiante la metodología de diseño VLSI.

5.-REFERENCIAS

[1] Piguét, C., Stauffer, A. y Zahd, J. "Conception des circuits ASIC numériques CMOS". Ed. Dunod, 1990.

[2] Schildt, H. "Turbo C/C ++. Manual de referencia". Ed. McGraw-Hill, 1992.

[3] Weste, N.H.E. y Eshraghian, K. "Principles of CMOS VLSI Design". Ed. Addison-Wesley, 1985.

[4] Deschamps, J.P. "Diseño de Circuitos Integrados de Aplicación Específica (ASICs)". Ed. Paraninfo, 1994.