

Visualizador de Mensajes de una Red de Usuarios con Arquitectura Maestro - Esclavo

Iván Jaramillo, Jorge Avella Castiblanco, Juan Manuel López.

{ijaramilloj, jmlopezl, jeavellac} @unal.edu.co

GMUN – Grupo de Microelectrónica Universidad Nacional

Abstract-- Este artículo presenta la metodología de diseño con Celdas Estándar, empleada para diseñar un circuito integrado de aplicación específica. Se describe cada una de las etapas que se desarrollaron, desde el planteamiento del problema, pasando por un diseño modular en lenguaje de descripción de hardware de alto nivel VHDL, la simulación del sistema en el paquete de software de XILINX ISE y las correspondientes pruebas de funcionamiento de la aplicación en un arreglo de compuertas programables por campo FPGA, hasta el desarrollo del plano final del chip empleando la herramienta CAD L-EDIT de Tanner Tools, aplicando las reglas de diseño para la tecnología CNM25.

Index Terms— ASIC, Celdas Estándar, Circuito Integrado, Metodología de Diseño, Reglas CNM25.

I. INTRODUCCIÓN

LA microelectrónica es sin lugar a dudas un campo de innovación tecnológica, que requiere del conocimiento de los últimos avances en materia. La Universidad Nacional intenta mantenerse al tanto del desarrollo tecnológico en el mundo, capacitando a sus estudiantes ante la demanda del mercado.

Para un diseñador los descubrimientos más interesantes, no son siempre la creciente capacidad de los CIs (Circuitos Integrados), sino las nuevas herramientas para diseñar su propio chip. Los CIs diseñados para un producto o aplicación particular se denominan ASICs. Los ASICs generalmente reducen el costo de fabricación de un producto al minimizar la cantidad de componentes y el consumo de potencia, estos a su vez proporcionan un desempeño más elevado, siempre y cuando el diseño sea óptimo. Los fabricantes de CIs desarrollaron las Celdas Estándar para facilitar el diseño de aplicaciones con base en funciones preestablecidas como decodificadores, registros, flip-flops, compuertas, contadores, etc. De esta manera, el diseñador lógico, debe interconectar estas celdas, para obtener así una determinada aplicación.

En este documento describiremos el proceso que seguimos para diseñar un ASIC mediante celdas estándar, partiendo del problema, luego diseñando los módulos necesarios para realizar la aplicación, posteriormente simular e implementar el diseño en una FPGA a través de código VHDL. A continuación se obtendrá el código Edif, necesario para diseñar las celdas. Finalmente, se diseñan las celdas y, con el código de interconexiones antes mencionado, se obtendrán los patrones geométricos de capas (Layout). Se trabajará con una de las tecnologías empleadas con fines educativos para el diseño de prototipos de circuitos integrados: CNM25;

disponible en la sala blanca del Centro Nacional de Microelectrónica en España, CMOS25.

II. PROCEDIMIENTO

A. Planteamiento del Problema

Se quiere implementar un red de dispositivos, controlada por una unidad maestra a la que están conectados un número fijo n de usuarios, que cuentan con un teclado PS/2 y un LCD de 16 caracteres por 2 líneas cada uno, de tal forma que pueden enviar un mensaje, desplegado para su edición en el LCD, a la unidad maestra que se encarga de visualizarlo en un arreglo de 7×40 LEDs, ubicado en lugar remoto al de los usuarios. Se debe implementar una interfaz amigable que permita introducir el mensaje y elegir entre algunos modos de visualización.

B. Síntesis

El proyecto culminará con el diseño de dos CIs, uno denominado *Master* y otro llamado *Slave*. El primero corresponde a la unidad controladora y visualizadora de mensajes y el segundo corresponde a los dispositivos que estarán a disposición de los usuarios, estos últimos poseen para su identificación, una dirección que puede estar entre 0 y 7, es decir, se podrán conectar al bus hasta 8 *Slave*. La dirección se puede modificar por parte del usuario a través de dos pines del integrado. Básicamente el diseño se divide en tres módulos de trabajo, que son:

1. Módulo Teclado PS/2
2. Módulo Comunicaciones
3. Módulo de Visualización

1) Módulo Teclado PS/2

El primer módulo, ubicado en el *Slave*, es el encargado de recibir la información que el usuario desea visualizar. Se trabaja con un teclado PS/2 para adquirir los datos y una pantalla de cristal líquido LCD, que permite visualizar la información que se va introduciendo. Podemos ver la entidad como una caja negra cuyas entradas serán las líneas del puerto PS/2 y cuyas salidas serán, un arreglo de 32 caracteres ASCII (para ser enviado al módulo de comunicaciones) y las señales que permiten la visualización del mensaje que se está introduciendo, en el LCD (ver Fig. 1)

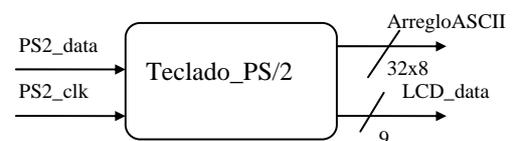


Fig. 1. Caja Negra Módulo Teclado PS/2

El usuario está en la capacidad de editar su mensaje antes de ser enviado al master para su visualización en el arreglo de LEDs. Este módulo prepara el *Arreglo ASCII* cuando el usuario ha terminado de introducir los caracteres y ha pulsado una de las teclas de visualización. Dichas teclas son las de las funciones (de F1 a F12).

2) *Módulo Comunicaciones*

Este módulo se encarga de la comunicación entre *Master* y *Slave*, por tanto se subdivide en dos componentes, localizados cada uno en un CI. La comunicación se baso en el medio RS485 [8], [9], [10], el cual está diseñado para aplicaciones con dispositivos interconectados a través de un bus de datos.

Además en este módulo se establece un protocolo Maestro - Esclavo, estableciendo tramas de comunicación para el master y para los slave, ver figuras 2 y 3.

Inicio de trama (':')	Dirección Slave	Código de Función	Fin de Trama
-----------------------	-----------------	-------------------	--------------

Fig. 2. Trama Dispositivo Master

Inicio de trama (':')	Dirección Slave	Código de Función	Mensaje	Nº de Datos	Datos	CRC	Fin de Trama
-----------------------	-----------------	-------------------	---------	-------------	-------	-----	--------------

Fig. 3. Trama Dispositivo Slave

El contenido de las tramas se describe a continuación:

- Inicio de Trama: Toda trama debe empezar con ':'
- Dirección Slave: Es un byte reservado, debe contener un número entre 00h y 07h. (solo 8 esclavos)
- Código de Función:

Función	Descripción
01h	Pedir Datos: El Master solicita Envío de Datos al Slave
02h	Lectura Correcta: El Master indica al Slave que la lectura de datos fue correcta
03h	Lectura Incorrecta: Los datos recibidos tienen errores.

Tabla 1. Códigos de Función

- Mensaje: Puede contener un 04h, que indica al Master que el dispositivo no posee información para enviar; o puede contener un 05h si hay datos.
- Nº de Datos: Indica la cantidad de Datos que se transmitirán en la sección *Datos*, esto debido a que si el usuario introduce un mensaje con una extensión menor a 32 caracteres, los espacios en blanco luego del fin del mensaje no se transmiten. Esto solo si el espacio mensaje contiene el código 05h, de lo contrario este campo no existe en la trama.
- Datos: Contiene la información que el usuario ha introducido por el Teclado en ASCII, además del modo de visualización. Este campo se transmite solo si el código en el espacio *mensaje* es 05h.
- CRC: Es el código de redundancia cíclica y se compone de dos bytes. Tiene sentido cuando el Slave posee datos para enviar.

Aquellos campos con líneas punteadas pueden o no estar presentes en la trama. Las reglas generales del protocolo son:

- Solo el maestro puede iniciar una sesión de comunicación.
- Existe un único maestro en la red.

- Cada esclavo tiene una dirección única con la cual se identifica.
- Los mensajes que envía el maestro van dirigidos a uno de los esclavos, a pesar de que todos los esclavos lo escuchan.

En el Master, se recibe la información que se desea visualizar, para lo cual se hace un llamado cíclico a cada uno de los Slave, determinando si poseen o no información para visualizar. Si todos los usuarios tienen un mensaje para enviar, el master se encarga de organizar los mensajes y visualizarlos uno a uno. En las figuras 4 y 5 se puede observar este módulo.

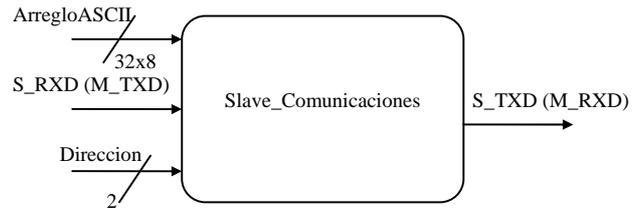


Fig. 4. Caja Negra Módulo de Comunicaciones para el Slave

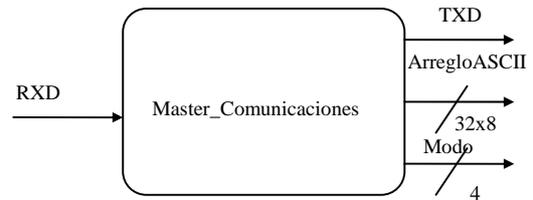


Fig. 5. Caja Negra Módulo de Comunicaciones para el Master

3) *Módulo de Visualización*

Es el encargado de mostrar la información deseada en una matriz de 7 x 40 LEDs. El manejo se hace de forma dinámica, por medio de 47 pines conectados al arreglo. Cada caracter que se visualiza ocupa un espacio de 7 filas por 8 columnas en la matriz. El usuario puede escoger entre 12 modos de visualización diferentes, por medio de las teclas de función, de la siguiente manera:

Tecla	Modo
F1	Modo Normal: Los LEDs correspondientes a cada caracter se encienden sobre un fondo apagado.
F2	Barrido Izquierda – Derecha: El Mensaje aparece por el lado izquierdo del arreglo de LEDs y se desplaza hacia la derecha a una velocidad de 1 caracter por segundo.
F3	Barrido Derecha – Izquierda: El Mensaje aparece por el lado derecho del arreglo y se desplaza hacia la izquierda a una velocidad de 1 caracter por segundo
F4	Barrido Arriba hacia Abajo: El mensaje aparece por secciones de 8 caracteres, las cuales se desplazan de arriba hacia abajo, a una velocidad de 1 fila cada 0.5 segundos.
F5	Barrido Abajo hacia Arriba: Igual que el modo anterior, solo que la sección parte desde la fila inferior del arreglo y se desplaza hacia arriba.
F6	Parpadeo: El mensaje se despliega como en el modo normal, solo que durante los tres segundo de cada sección se apaga y se vuelve a encender 3 veces cada segundo.
F7	Modo Normal Invertido: Igual que el modo Normal, con la diferencia de que el fondo de los caracteres es ahora lo que se enciende.
F8	Barrido Izquierda – Derecha Invertido.
F9	Barrido Derecha – Izquierda Invertido.

F10	Barrido Arriba hacia Abajo Invertido.
F11	Barrido Abajo hacia Arriba Invertido.
F12	Parpadeo Invertido.

Tabla 2. Modos de Visualización

En caso de que no haya un mensaje nuevo, se sigue mostrando el mensaje anterior. En la Fig. 6. podemos ver la caja negra de este módulo, cuyas entradas serán el arreglo de caracteres recibido por el módulo de comunicaciones en el maestro y el modo de visualización y cuyas salidas son las líneas que controlan la matriz de LEDs.

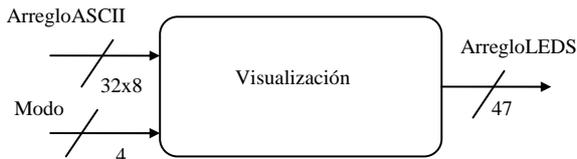


Fig. 6. Caja Negra Módulo de Visualización

4) Controladores

Los módulos descritos anteriormente requieren de entes que controlen su funcionamiento, estos son básicamente máquinas de estados finitos que habilitan a uno u otro módulo, en los momentos adecuados.

5) Consideraciones

Los módulos se presentan de manera muy general. Una descripción extensa está fuera del propósito de este documento. El esquema del proyecto se presenta en la Fig. 11.

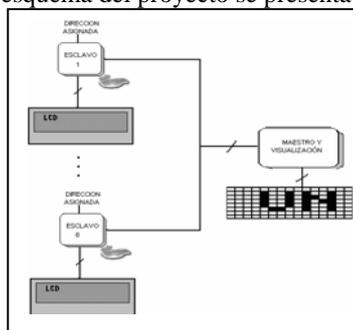


Fig. 11. Esquema general del Proyecto

Empleando el gráfico Y de Gajski y Kuhn, se parte entonces del dominio funcional, en el cual se tiene por cada módulo una caja negra donde se especifican sus entradas y salidas y la relación entre ellas. De esta forma, cada módulo es diseñado en lenguaje VHDL [11]. Finalmente se implementa y simula cada módulo usando la herramienta computacional MODELSIM para finalmente implementar el diseño obtenido en una FPGA SPARTAN3, como herramienta de verificación del correcto funcionamiento del sistema [7].

C. Pruebas de Funcionamiento del Código

En la sesión de pruebas se utilizaron dos esclavos programados con direcciones distintas en dos FPGAs, con sus respectivos teclados y LCDs. Se probaron los 12 modos de visualización con todos los caracteres ASCII, y se obtuvo un resultado satisfactorio. Luego se pasó al dominio estructural, en donde el diseño se ve como una caja transparente, definida en términos de componentes sencillos y de sus

interconexiones; se creó el archivo .edif, el cual puede ser generado a partir de un archivo de XILINX ISE. El archivo contiene una descripción textual del diseño esquemático y en él se definen las celdas requeridas. El archivo Edif. depende del dispositivo en el que se quiera implementar el diseño, en nuestro caso se probó en una FPGA, sin embargo, las celdas estándar resultantes eran bastante complejas, ya que era necesario crear los Layout para memorias RAM, ROM, LUT, entre otras. Por lo tanto, se generó el archivo Edif para una CPLD XC9500XV, de tal forma que las celdas requeridas eran compuertas sencillas [10].

Finalmente para diseñar el Layout, se usó la herramienta CAD L-Edit [2], que necesita una plantilla para el diseño y que contiene las reglas con las que se va a trabajar, (CMOS CNM25 [3], [4] para nuestro caso), de esta forma, se crea una grilla de 0.25 μm . CNM25, es una tecnología de pozo doble, con aislamiento por oxidación local LOCOS; posee dos niveles de polisilicio y dos de metal, esta técnica permite tanto la integración de circuitos digitales como analógicos y mixtos debido a la posibilidad de implementar capacitancias con los dos niveles de polisilicio. En las reglas de diseño CNM25 [6] se especifican parámetros tales como el ancho y la longitud mínima de los canales de los transistores, los cuales deben ser de $W = 2.5 \mu\text{m}$ y $L = 3 \mu\text{m}$ respectivamente; además, se establecen las distancias de separación, solapamiento, desbordamiento, margen, los anchos de pistas, las dimensiones fijas, los niveles de corriente y los tipos de cubrimientos permitidos sobre zonas activas, pozos, niveles de polisilicio, metales, implantaciones, pasivaciones, vías y contactos.

D. Diseño del Layout

1) Diseño de una Celda Estándar:

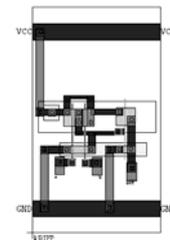


Fig. 12. Layout de Compuerta NAND de dos Entradas.

Para cada celda se debe colocar un puerto de empalme. La dimensión y posición de estos puertos corresponden a los límites de la celda, este puerto debe tener la misma altura en todas las celdas que se utilicen. Los buses de alimentación entran y salen al final de las filas de las celdas estándar y las demás señales se enrutan por el lado superior o inferior de la celda, por lo demás, se deben cumplir las normas de diseño de la tecnología CNM25. En la Fig. 12 se presenta el layout de una compuerta NAND de 2 entradas diseñada en L-Edit con las reglas CNM25.

El siguiente paso fue el enrutamiento de las celdas, el cual se puede hacer de manera automática usando el paquete de herramientas SPR (*Standard Place and Route*). Configurando esta herramienta con las reglas CNM25, el archivo Edif. y las celdas estándar se obtiene la geometría interna del Padframe

de nuestro diseño, es decir el esquemático hasta el borde interno del circuito.

Para poder asignar los Pads, previamente se debió diseñar las celdas con cuatro puertos de señales, dos de ellos para la alimentación y los otros para las señales globales que se tengan en el circuito (entradas y salidas del sistema). De esta manera, es posible hacer el enrutamiento de Pads, ya que L-Edit conecta las señales globales al puerto apropiado en el Padframe. Finalmente se hacen los terminales de conexión del circuito, es decir los pines.

Así, se diseñaron dos integrados, el *Master* tenía finalmente 51 pines, dos de ellos para la alimentación, 2 para la comunicación con los *Slave* y 47 pines para controlar la matriz de LEDs empleada. El segundo circuito denominado *Slave*, tenía 15 pines, 9 de estos para manejar el LCD, 2 para la comunicación con el maestro, dos para el manejo del teclado ps/2 y dos de alimentación, los diagramas finales de los circuitos integrados se muestran en las Figuras 14 y 15.

Se simularon los transistores empleados para el diseño y se observó que el voltaje umbral obtenido fue cercano al voltio. Por otra parte, con una alimentación de 5V, una capacitancia de carga de 1pF a una frecuencia de 4MHz, a temperatura ambiente (25 °C), se obtuvieron tiempos de subida y bajada en los transistores, del orden de las décimas de nano segundos, los cuales eran resultados cercanos a los que estábamos esperando según (1) y (2). Asimismo, se verificó el correcto funcionamiento de cada celda, usando la herramienta de simulación ORCAD PSPICE. [13]

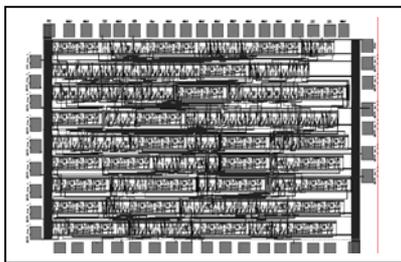


Fig. 14. Chip MASTER, este chip contenía el módulo maestro y de visualización del proyecto.

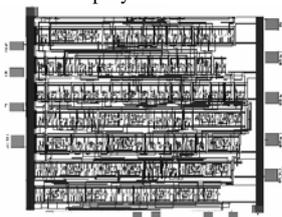


Fig. 15. Chip SLAVE, este chip contenía el módulo maestro y de visualización del proyecto.

III. CONCLUSIONES

La metodología empleada permite diseñar circuitos de aplicación específica con un alto nivel de confiabilidad, ya que la mayoría de los pasos a realizar se llevan a cabo de manera automática mediante software. El punto más vulnerable de la técnica empleada es tal vez el diseño de las celdas estándar, ya que del buen diseño del Layout final del integrado depende que funcione o no la aplicación a la hora de implementarlo físicamente en un chip. Por lo tanto, las

herramientas CAD constituyen una ayuda importante para el diseño de circuitos integrados, actividad que en décadas pasadas se efectuaba de manera gráfica, lo que consistía una limitación para el tamaño de los diseños. El Software ha permitido a los ingenieros trabajar en un nivel de abstracción más alto, de modo que los sistemas que se diseñan son más complejos y tienen un mejor desempeño.

Otro punto en el cual se debe hacer énfasis es en la preparación del diseñador lógico, ya que de no lograrse una descripción óptima de un circuito, se sobredimensionaría el problema, lo cual conllevaría a que se ocupe más espacio en un CI, que a su vez se traduce en más costos de fabricación para el diseñador.

IV. REFERENCIAS

- [1] Jan Rabaey. "Digital Integrate circuits A Design Perspective". Agosto 17,2000.
- [2] Andrés Falcony, Carolina Albis, Iván Díaz, Pedro Duarte, Andrés Fernández, Guillermo García, Héctor Gratz, Sandra Gómez, Tatiana sarmiento, Mauricio morato, Iván Jaramillo profesor a cargo. "Manual L-Edit". Universidad Nacional de Colombia.
- [3] www.cnm.es
- [4] Joaquín Santander, Manuel Lozano. "Motivos de test para el test paramétrico de las tecnologías CNM25 y CNM175". Nota técnica. Octubre de 1996.
- [5] J.A. Plaza, M. J. Lopez, I. gràcia, C. Cane, J. Wöllestein, G. Kühner, G. plescher, H. Böttner. "A glass/ silicon technology for low power robust gas sensor". IEEE Sensor Journal.
- [6] Normas de Diseño para la tecnología CNM25. Versión No1. 4 de Noviembre de 1999.
- [7] Spartan-3. Starter Kit Board. User Guide. <http://www.digilentinc.com/Data/Products/S3BOARD/S3BOARD-rm.pdf>
- [8] The art and Science of RS485. Julio 1999 <http://www.circuitcellar.com>
- [9] Jan Axelson. "Serial port complete Programming and circuits for RS232 and RS485 Links and Networks". Lakeview research. Usa 1998. <http://www.lvr.com>
- [10] Mark Balch. "Complete digital Design". Mc Graw Hill. New York 2003.
- [11] Douglas L. Perry. "Programming by example". Mc Graw Hill. Cuarta Edición. New York 2002.
- [12] Sedra/ Smith. Circuitos Microelectronicos.Oxford University Press. Cuarta Edicion. Mexico D.F 1999.
- [13] Grupo de Técnicas de integración I-2006. Documento final "Diseño de un ASIC aplicando la metodología de celdas estándar"