

Diseño de módulos I2C en FPGA de la familia Altera

J. A. Cabot¹, D. Criado², V. Escartín³

¹ CIME, Facultad de Eléctrica, CUJAE CORREO

² CIME, Facultad de Eléctrica, CUJAE dilaila.criado@electronica.cujae.edu.cu

³ CIME, Facultad de Eléctrica, CUJAE victor.escartin@electronica.cujae.edu.cu

RESUMEN / ABSTRACT

En este trabajo se describe el diseño de un módulo I2C master utilizando registros del tipo FIFO, y de un módulo I2C esclavo, para implementar el protocolo I2C. Para el diseño se utilizó un FPGA (Field Programmable Gate Array), que mediante un lenguaje de descripción de hardware (HDL), permite realizar diseño de cualquier sistema digital. En esta aplicación se utilizó el lenguaje VHDL (Very High Speed Hardware Description Language) y la plataforma de desarrollo Quartus II versión 5. El módulo I2C master realizado en este trabajo se encarga de la comunicación entre un microcontrolador de la familia MSC51 y el módulo I2C esclavo con velocidades iguales o superiores a la que el estándar impone en los circuitos comerciales. El diseño fue realizado utilizando bibliotecas de módulos parametrizables conocidas por sus siglas en inglés LPM (Library Parametric Module) que ofrece Altera en el software utilizado. Estos módulos permiten disminuir el consumo de celdas optimizando el rendimiento, aunque el pago sea la no universalidad del código empleado. De esta forma se logra un diseño que puede ser utilizado en diferentes aplicaciones. ARREGLAR

Palabras claves: *FPGA, I2C, Módulo IP, VHDL*

Design and implementation of I2C modules using programmable logic from Altera.

Design of I2C modules in FPGA from Altera

Titulo *En ingles FAlta*

In this work one he describes the design of a module I2C master using registers of the type FIFO, and of a module I2C slave, gives birth implement the I2C protocol.

For the design a FPGA was used (Field Programmable Gate Array) that by one language DHL allows to make dise Hardware ño. In this application one the language used VHDL (Description Language High Very) and the Quartus II development platform version 5.

The Module I2C master He takes charge of the communication between a MSC51 family's microcontroller and that Module I2C slave with an above speed between both modules her that the standard imposes in the commercial circuits.

The design was made using LPM (Library Parametric Module) that offers Alters in the used software. These modules allow to reduce the cell optimizing the performance, though the payment is her not consumption code used universality. Of this way I achieve a flexible design and that can be used in different applications.

Key words: *FPGA, I2C, Module IP, VHDL*

INTRODUCCIÓN

En sistemas con microcontroladores (o microprocesadores), en ocasiones se requiere de intercambiar informaciones entre circuitos integrados y el protocolo I2C permite hacerlo utilizando simplemente un bus bidireccional de dos alambres. Este protocolo es confiable y permite realizar sistemas multimaster de una forma muy económica.

Hace algunos años era muy difícil realizar diseños complejos, para volúmenes pequeños o medianos de producción. Sin embargo, con el avance de la tecnología se puede realizar con determinada facilidad diseños complejos y de bajo costo, es decir, es posible diseñar sistemas embebidos totalmente a la medida para una aplicación dada, y aún para volúmenes muy pequeños, usando dispositivos FPGAs.

El objetivo central de este trabajo es diseñar para la familia Cyclone de Altera utilizando VHDL módulos I2C con un Baud Rate máximo no menor de 10Mbps. Para esto se diseñan dos módulos I2C uno en modo master y el otro en modo esclavo. La plataforma de desarrollo para la síntesis y simulación es el software Quartus II versión 5 de Altera.

MATERIALES Y MÉTODOS

Diseño del Módulo I2C master

Este módulo debe cumplir con los siguientes requerimientos:

- Debe operar en modo master.
- Debe trabajar con velocidades no menores de 10 Mb/s.
- Interfaz con microcontroladores de la familia MSC51.
- 4 canales de comunicación I2C.
- Menor consumo de recursos posible.
- FIFO de transmisión y de recepción en el Master.

Arquitectura del Módulo I2C master

El diseño está compuesto por 5 bloques como se muestra en la figura 1. El bloque de puertos de entrada, constituido por un multiplexor que selecciona la información a leer. El bloque de interfaz que se encarga de la comunicación con el microcontrolador. El bloque de interrupción y estado, para generar la interrupción e indicar el estado del módulo. El bloque de control para el funcionamiento del módulo en general. Y el bloque Módulo I2C que se dedica a esta comunicación. En la tabla 1 se presentan los terminales del módulo.

Bloque Puertos de Entrada

Este bloque está formado por un multiplexor que selecciona el registro que se desea leer.

Bloque de la Interfaz con el microcontrolador.

Este bloque tiene la función de acoplar el módulo con un microcontrolador de la familia MSC51. Tiene dos modos de trabajo: escritura y lectura. El sentido de la información depende del modo y se realiza mediante las señales *RD*, *WR* y *CS* del microcontrolador. Este bloque está formado por un registro con tres estados realizados con LPM y un circuito de control.

Bloque de Interrupción y Estado.

En este bloque se encuentran el Registro de Interrupción, el Registro de Estado y una máquina de estados para el reset de la interrupción.

La función de este bloque es la de generar la señal de interrupción al microcontrolador. La interrupción se solicita cuando cada FIFO termina de transmitir o recibir la información.

Se encuentra también el Registro de Estado, que es donde el microcontrolador lee el estado en que se encuentra el módulo cada vez que este pide interrupción.

Diseño del Módulo de Control

La función de este módulo es la de generar las señales que permiten el funcionamiento de los diferentes bloques.

Módulo I2C.

Para el diseño de este bloque se utilizó un módulo ya programado en VHDL [1] el cual está formado por 3 bloques funcionales: *timer*, *i2c_byte_controler* y la *interfaz*. Los módulos *timer* e *i2c_byte_controler* son módulos que definen el funcionamiento del protocolo I2C. Y el módulo *interfaz* se encarga de acoplar con el microcontrolador.

Para esta aplicación se modificó la interfaz dando como resultado dos bloques funcionales: *System Interfaces* y *Demux_I2C_Ctrl*.

El primero tiene como función detectar la terminación de cada operación realizada en el bus I2C, los registros de control de la comunicación y de transmisión y recepción del tipo FIFO implementados con LPM, además de las máquinas de estado que permiten el correcto funcionamiento de la comunicación.

El bloque *Demux_I2C_Ctrl* permite 4 canales I2C y un reloj común. Los canales son puertos bidireccionales, ponen en tercer estado el bus I2C cuando no se esté realizando alguna operación. Como puertos de entrada/salida se utilizó una LPM que ofrece Altera. El control fue realizado mediante una máquina de estado, que controla la configuración de cada puerto según sea la operación a realizar.

Síntesis y verificación del módulo I2C master.

La síntesis fue realizada para el dispositivo EP1C3T100C6 de la familia Cyclone [2] de Altera.

El diseño se implementó con un FIFO de transmisión y otro de recepción de 256 palabras de 8 bits cada una. La tabla 2 muestra el resultado de la síntesis y la Figura 3 los resultados de la simulación.

Diseño del Módulo I2C esclavo

Este módulo debe cumplir con los siguientes requerimientos:

- Debe operar en modo esclavo.
- Debe trabajar con velocidades no menores de 10 Mb/s.
- Menor consumo de recursos posible.

Arquitectura del Módulo I2C esclavo

Este módulo está compuesto por dos bloques como se muestra en la figura 4. Un bloque de puertos de entrada y salida, constituido básicamente por registros y registros de desplazamientos, que transforma los datos de forma serie a paralelo y viceversa. Y un bloque de control, formado por

máquinas de estados que garantiza la lógica de trabajo de este módulo.

En la figura 4 se muestra el diagrama general de la arquitectura, y la Tabla 3 describe los terminales de este módulo

Diseño del bloque de Puertos de Entrada y Salida

En este bloque se encuentra todos los puertos de entradas y salida, realizados con registros y registros de desplazamientos utilizando bibliotecas de módulos parametrizados, conocidas por sus siglas en inglés **LPM** (*Library of Parametrized Modules*). Cada registro empleado tiene la siguiente función:

- **Shift_Register_IN**: Registro de desplazamiento donde entran los datos series del bus I2C y salen de forma paralelo a los registros internos del módulo.
- **Shift_Register_OUT**: Registro de desplazamiento donde entran los datos de forma paralelo, y salen de forma serie hacia el bus I2C.
- **Sal_Datos**: Registro donde se transmiten los datos de forma paralelo.

Este módulo tiene un buffer con tercer estado conectado al terminal *SDA* del bus I2C, con la función de generar el bit de reconocimiento cada vez que lleguen los datos correctamente.

Bloque Control I2C

Este bloque se encarga del control de los registros del *bloque de Puertos de Entrada y Salida*, y del proceso de escritura y lectura desde y hacia el *Módulo I2C esclavo*.

Está formado por 5 máquinas de estados y cada una tiene la función de controlar o generar alguna señal interna o hacia el bus I2C. La señal *SDA* del bus I2C es bidireccional por lo que se utilizaron puertos con tres estados. A continuación se describe la función de cada máquina de estado.

- **FSM_START**: Se encarga de seguir el estado del bus I2C y generar una señal interna que indica a las otras máquinas de estado que se va a comenzar una comunicación entre el dispositivo master y el dispositivo esclavo.
- **FSM_STOP**: Se encarga de seguir el estado del bus I2C y generar una señal interna que indica a las otras máquinas de estado que terminó la comunicación entre el dispositivo master y el dispositivo esclavo.
- **FSM_SCL**: Esta máquina de estado sigue los pulsos del reloj de la I2C (*terminal SCL*) generado por el dispositivo master, y genera una señal interna hacia las otras máquinas de estados que indican la cantidad de pulsos llegados.
- **FSM_ACK**: Esta máquina de estado genera el habilitamiento para dar la señal de reconocimiento cada vez que lleguen los datos correctamente al *Módulo I2C esclavo*.
- **FSM_R_W**: Se encarga del proceso de escritura y lectura entre el dispositivo master y el dispositivo esclavo.

Genera las señales de habilitamiento y selección de los bloques.

La conexión entre las 5 máquinas de estado y las señales que se intercambian entre ellas se muestra en la Figura 4

Síntesis y verificación del diseño

La arquitectura de este módulo fue sintetizada y verificada utilizando el software Quartus II (versión 5.0), sobre el dispositivo EP1C3T140C6 de Altera, de la familia Cyclone. Los resultados de la síntesis se muestran en la Tabla 4.

En la tabla 5 se muestra la comparación del módulo I2C esclavo diseñado en este trabajo y un módulo IP (*por sus siglas en inglés Intellectual Property*) I2C esclavo. Este módulo IP es comercializado por la compañía Digital Core Design. En su documentación aparece la síntesis realizada a dispositivos de las compañías Altera, Lattice y Xilinx. En el caso de los dispositivos de Altera, se encuentra la síntesis realizada a un dispositivo de la familia Cyclone de grado 6. En la figura 5 se muestran los resultados de la simulación de este módulo

CONCLUSIONES

Se realizó el diseño de un módulo I2C master utilizando registros del tipo FIFO, permitiendo ganar en velocidad los sistemas donde se utilice este módulo. Se diseñó un módulo I2C esclavo con la posibilidad de intercambiar información entre ambos módulos a una velocidad mayor que la admitida por los circuitos comerciales.

Ambos diseños se realizaron utilizando VHDL sobre dispositivos de ALTERA y con el empleo de LPM, que posibilita un mejor aprovechamiento del uso de los recursos de los dispositivos

REFERENCIAS

Las referencias dentro del texto se harán por orden numérico ascendente consignándose así en el acápite referencias, en el mismo orden que aparecieron en el texto..

1. **VALDESPINO NUÑEZ, P., CRIADO CRUZ, D., ESCARTÍN FERNÁNDEZ VÍCTOR;** "Diseño e implementación de un módulo I2C con VHDL"; presentado en *Congreso Internacional de Telemática y Telecomunicaciones*; La Habana, Noviembre 28 – Diciembre 1, 2006
2. Altera Corporation, Section 1. Cyclone FPGA Family Data Sheet, January 2007.
3. Digital Core Design, "I2C Bus Interface - Slave", April 2008.

AUTORES

Jorge Alberto Cabot López . Profesor asistente Graduado en 2008 de Ingeniería en Telecomunicación y Electrónica

Dilaila Criado Cruz. Investigador Agregado y Master en Electrónica, Graduado en 1992 de Ingeniería Física Nuclear.

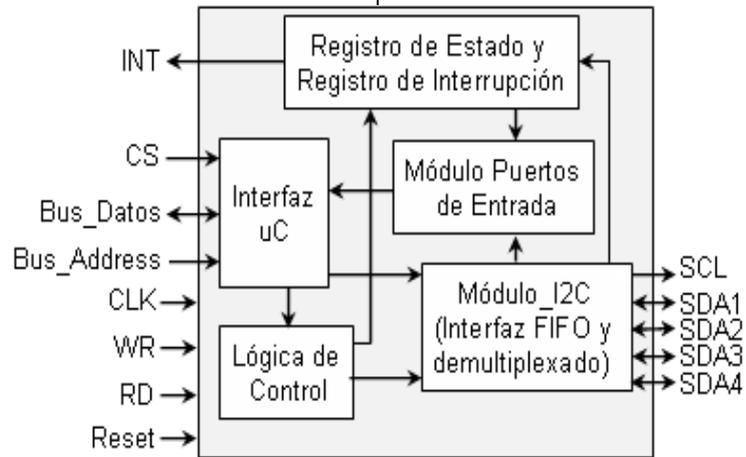


Figura 1. Arquitectura General



Figura 2 Resultado de la simulación del Módulo I2C master.

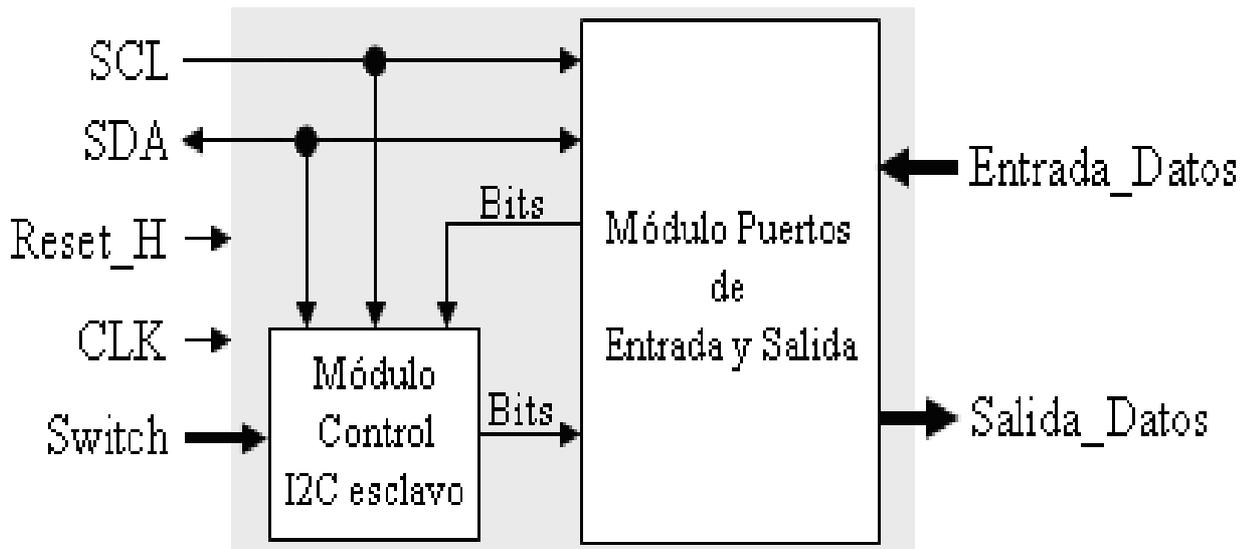


Figura 3: Diagrama en bloques del Módulo_I2C_esclavo.

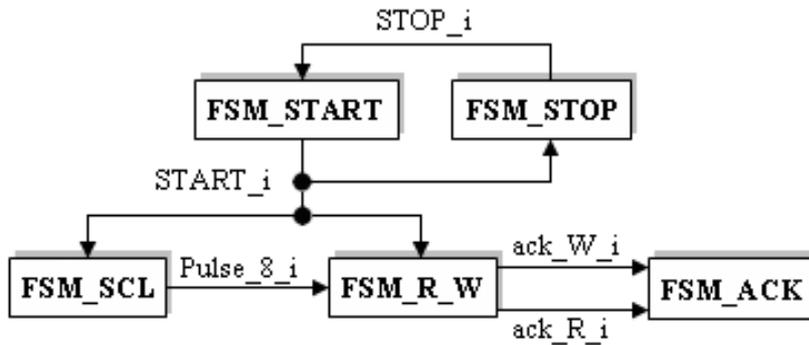


Figura 4: Diagrama de conexión entre las máquinas de estados.

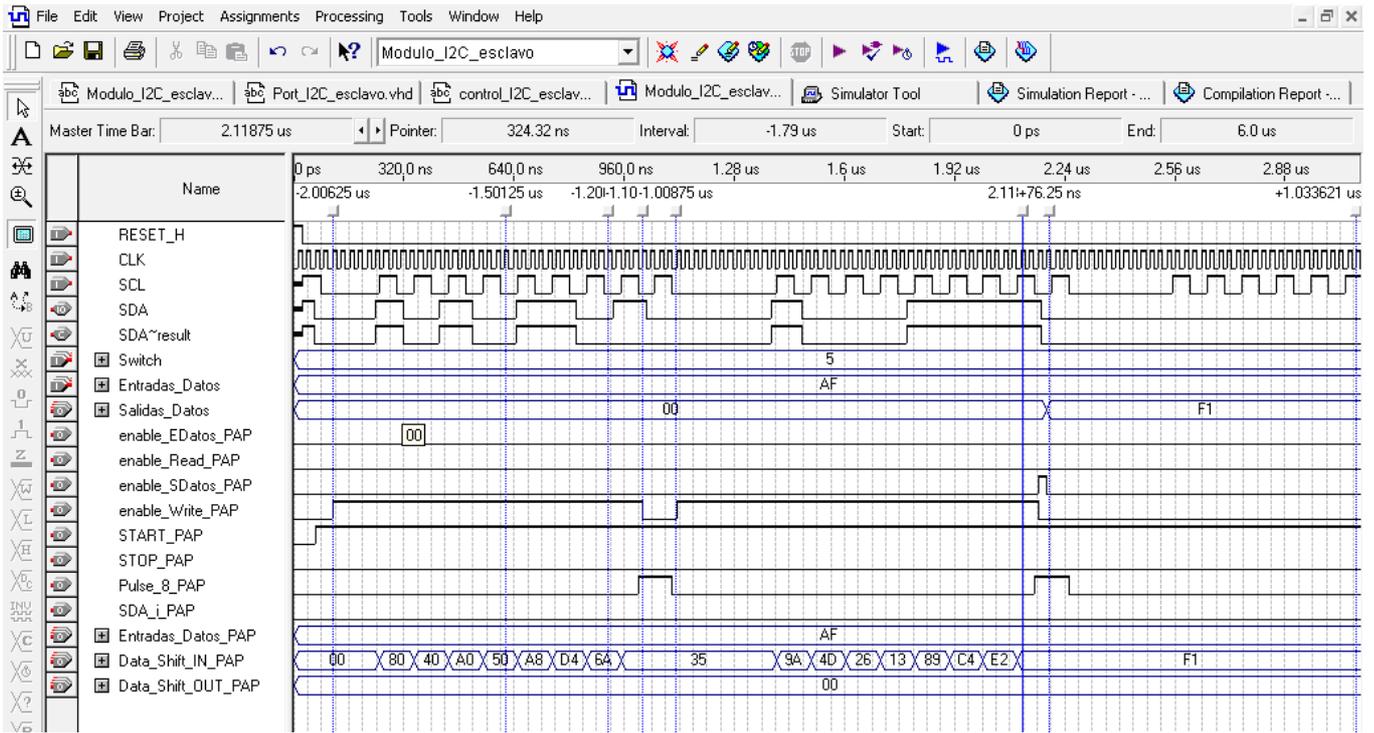


Figura 5: Resultado de la simulación del Módulo I2C esclavo.

Tabla 1. Terminales del módulo

Terminal	Tipo	Descripción
CLK	Entrada	Reloj general
Reset	Entrada	Reset general.
WR	Entrada	Señal de escritura de los registros internos.
RD	Entrada	Señal de lectura de los registros internos.
CS	Entrada	Señal para activar el módulo.
Bus_Address	Entrada	Bus de dirección.
Bus_Datos	Bidireccional	Bus de datos.
INT	Entrada	Señal de Interrupción.
SDA1...SDA4	Bidireccional	Datos para la I2C
SCL	Salida	Reloj de la I2C

Tabla 2 Síntesis del diseño en el FPGA.

Familia	Dispositivo	EL	Bits de Memoria	Fmáx. (MHz)
Cyclone	EP1C3T100C6	306	4096	167.98

Tabla 3 Terminales del Módulo_I2C_esclavo.

Terminal	Tipo	Descripción
CLK	Entrada	Reloj General
Reset_H	Entrada	Reset General
SCL	Entrada	Reloj del bus I2C
SDA	Birideccional	Datos del bus I2C
Switch	Entrada	Señal de dirección del módulo
Entrada_Datos	Entrada	Entrada paralelo
Salida_Datos	Salida	Salida paralelo

Tabla 4: Síntesis del Módulo I2C esclavo.

Familia	Dispositivo	Elementos Lógicos (LE)	Bits de Memoria	Fmáx (MHz)
Cyclone	EP1C3T140C6	88	0	278.32

Tabla 5: Comparación de resultados.

Dispositivo	Familia	LE	Grado de Velocidad	Fmáx (MHz)
Digital core	Cyclone	170	6	220
CIME	Cyclone	88	6	278.32