

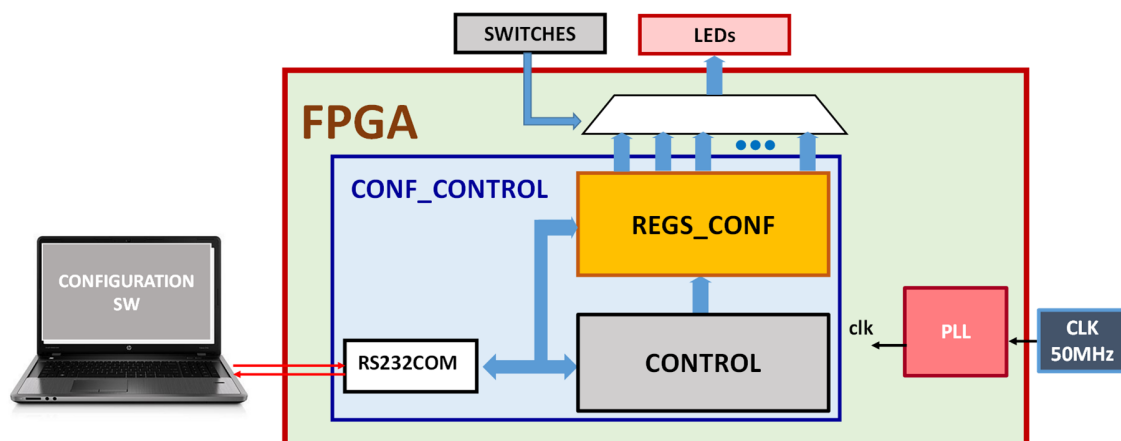
# E5

## Escritura y lectura de registros de configuración

El **objetivo** de esta práctica es implementar el bloque de comunicación entre el PC y el dispositivo FPGA, que nos permitirá escribir y leer desde el PC, a través de un puerto serie RS232, los registros de configuración del modulador de FM/AM. El subsistema implementado deberá funcionar a una frecuencia de reloj de 25 MHz y se verificará utilizando la tarjeta DE2-115 escribiendo y leyendo los registros desde un PC.

A continuación, se muestra el esquema que se utilizará en esta práctica para validar el subsistema de comunicación con el PC. Está compuesto de los siguientes bloques:

- **RS232COM**: este módulo (cuyo código HDL se proporciona) se encarga de realizar la capa física de la comunicación RS232.
- **REGS\_CONF**: módulo que incluye los registros de configuración del modulador AM/FM y los registros necesarios para implementar la comunicación.
- **CONTROL**: bloque encargado de gestionar la comunicación
- **PLL**: Phase Locked-Loop del dispositivo Cyclone IV configurado para generar un reloj de  $f_{clk}=230400$  kHz a partir del reloj de 50 MHz de la tarjeta DE2-115.
- **Interruptores y LEDs** para visualizar los valores almacenados y facilitar la depuración.



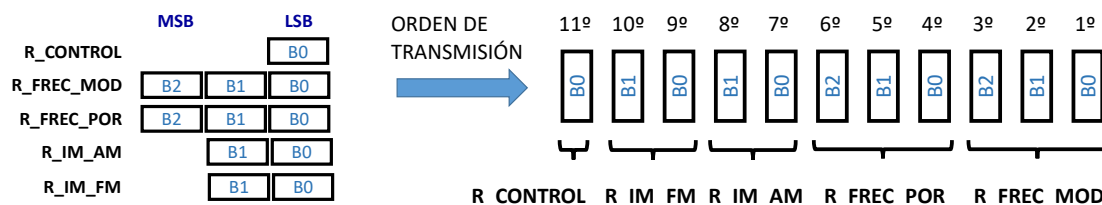
### Registros de configuración del modulador AM/FM

El funcionamiento del modulador se controla configurando 5 registros. Sus nombres, tamaño y la función que desempeñan se exponen en la siguiente tabla.

REGISTRO	TAMAÑO	FUNCIÓN
R_CONTROL	1 byte	Registro de control
R_FREQ_MOD	3 bytes	Paso del DDS para generar la frecuencia de las señales de test
R_FREQ_POR	3 bytes	Paso del DDS para generar la frecuencia portadora
R_IM_AM	2 bytes	Índice de modulación de AM
R_IM_FM	2 bytes	Índice de modulación de FM

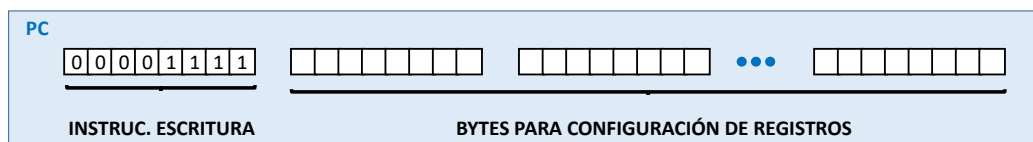
### Procedimiento de comunicación

El puerto serie nos permite transmitir datos de tamaño byte. Por tanto, los datos de configuración de cada registro se tienen que descomponer en bytes para transmitirlos de forma independiente respetando el orden de transmisión indicado en la siguiente figura.

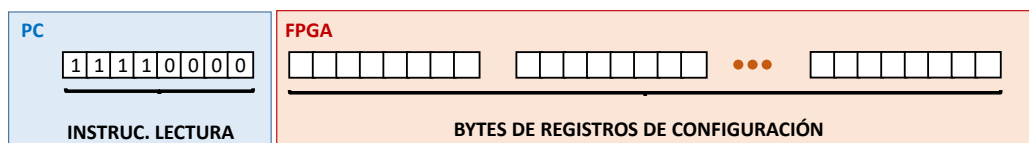


El procedimiento de comunicación es el siguiente:

- Para transmitir los 11 bytes desde el PC al dispositivo FPGA, primero se envía desde el PC un byte con el código "0Fh", que corresponde a la instrucción de escritura, y después, se envían los 11 bytes en el orden indicado anteriormente.

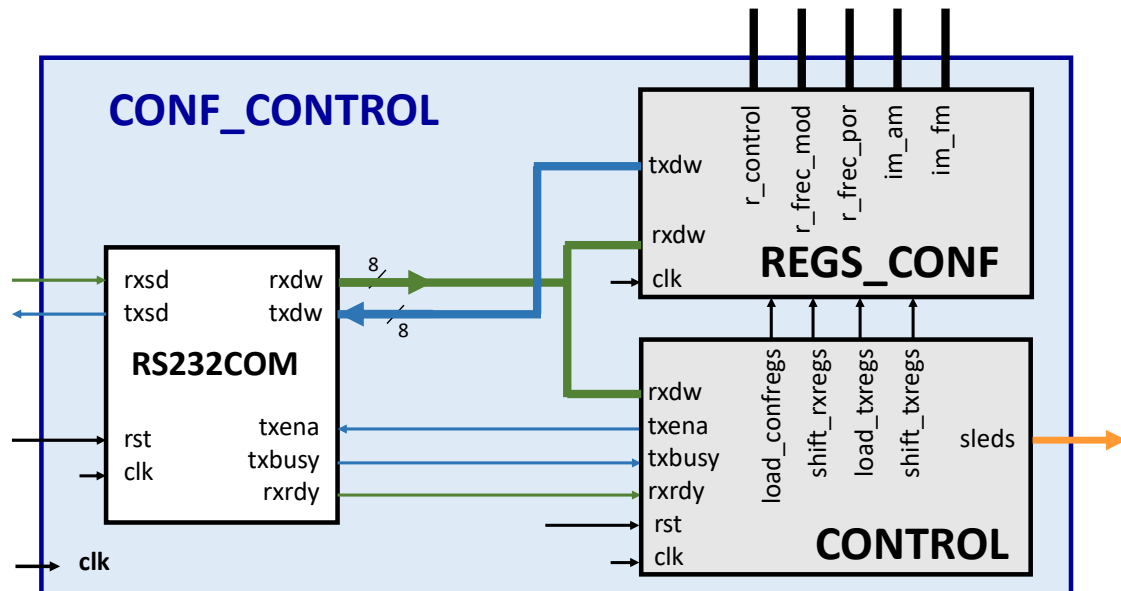


- Para transmitir desde el dispositivo FPGA al PC el contenido de los registros, primero se envía desde el PC un byte con el código "F0h", que corresponde a la instrucción de lectura, y después se envían desde el dispositivo FPGA los 11 bytes manteniendo el orden indicado anteriormente.



## Módulo CONF\_CONTROL

El módulo CONF\_CONTROL es el que se encarga de gestionar la comunicación entre el PC y el dispositivo FPGA. Está formado por 3 bloques (RS232COM, CONTROL y REGS\_CONF), conectados tal y como se indica en la figura. En las salidas del módulo CONF\_CONTROL están accesibles los contenidos de todos los registros de configuración del modulador AM/FM.



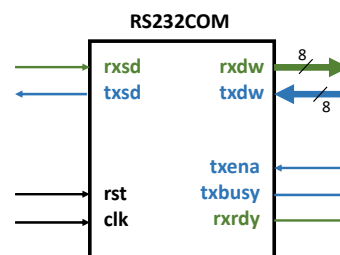
A continuación, se describe el interfaz y comportamiento de cada uno de los bloques que forman el módulo CONF\_CONTROL.

## Módulo RS232

RS232COM es el módulo que se encarga de implementar la capa física de la comunicación entre el PC y el dispositivo FPGA.

Está configurado para transmitir/recibir bytes a 57600 baudios, incluyendo 1 bit de start y 1 bit de stop.

Su interfaz se describe en la siguiente tabla:

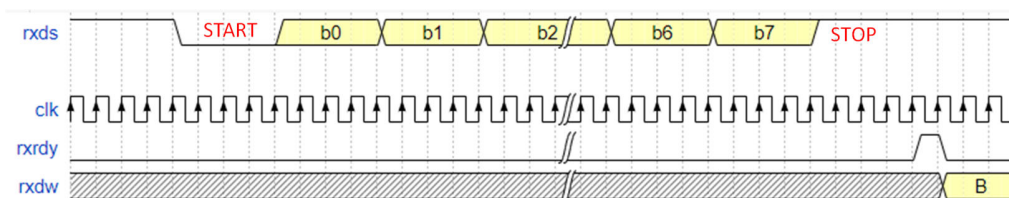


Módulo RS232			
Nombre	Tipo	Formato	Descripción
clk	In	Bit	Entrada de reloj
rst	In	Bit	Entrada de reset, activa a nivel alto
rxsd	In	Bit	Dato serie recibido en el puerto RS232
txsd	out	Bit	Dato serie transmitido por el puerto RS232
rxdw	out	8 bits	Dato (byte) recibido
txdw	in	8 bits	Dato (byte) a transmitir

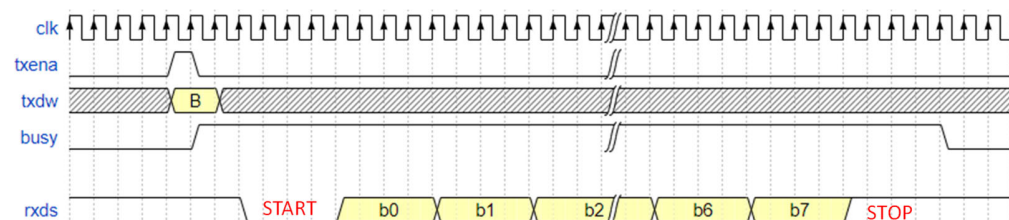
<b>txena</b>	In	bit	entrada de validación de transmisión; cuando se activa (a nivel alto) el byte presente en <b>txdw</b> se transmite en serie a través de <b>txsd</b>
<b>txbusy</b>	out	bit	salida de estado, activa a nivel alto, que indica que el bloque RS232COM está ocupado transmitiendo un byte (y no se puede realizar otra transmisión)
<b>rxrdy</b>	out	bit	salida que indica que se ha recibido un dato por el puerto serie y está disponible en <b>rxdw</b>

En las siguientes figuras se ilustran los cronogramas que muestran el funcionamiento del interfaz en el caso de recepción y transmisión serie del byte B=[b7 b6 b5 b4 b3 b2 b1 b0]:

- Ejemplo recepción de dato serie: el PC envía el byte B=[b7 b6 b5 b4 b3 b2 b1 b0] al dispositivo FPGA



- Ejemplo envío de dato serie: El dispositivo FPGA envía el byte B=[b7 b6 b5 b4 b3 b2 b1 b0] al PC. Mientras la señal busy está activa (a 1), no se puede enviar otro dato.

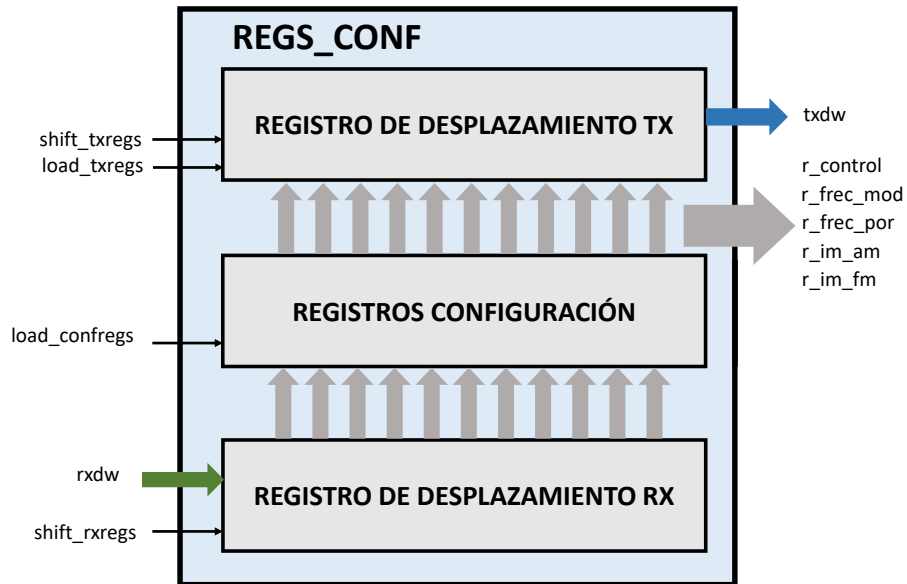


## Módulo REGS\_CONF

El módulo REGS\_CONF, cuyo esquema se muestra en la siguiente figura, contiene los registros de desplazamientos necesarios para la recepción y transmisión serie, byte a byte, de los bytes de configuración y los registros de configuración:

- Registro de desplazamiento RX. Es un registro de desplazamiento de 11 registros de tamaño byte, con carga serie y salida paralela. Se encarga de ir almacenando y desplazando los 11 bytes cuando lo indique la señal de desplazamiento **shift\_rxregs**, manteniendo el orden en el que son recibidos.
- Registros de configuración. Lo forman 11 registros de tamaño byte que se cargarán a la vez, cuando lo indique la señal de carga **load\_confregs**.
- Registro de desplazamiento de TX. Es un registro de desplazamiento de 11 registros de tamaño byte, con carga paralela y salida serie. Se encarga de capturar los valores de los registros de

configuración cuando lo indique la señal **load\_txregs** y de ir desplazándolos cuando lo indique la señal de desplazamiento **shift\_txregs** para transmitirlos al PC.



Su interfaz es el siguiente:

Módulo REGS_CONF			
Nombre	Tipo	Formato	Descripción
clk	in	Bit	Entrada de reloj
rxdw	in	8 bits	Dato (byte) recibido
txdw	out	8 bits	Dato (byte) a transmitir
shift_rxregs	in	bit	Señal de activación (a nivel alto) del desplazamiento del registro de desplazamiento de RX
load_confgregs	in	bit	Señal de activación (a nivel alto) de la carga del registro de configuración
load_txregs	in	bit	Señal de activación (a nivel alto) de la carga del registro de desplazamiento de TX
shift_txregs	in	bit	Señal de activación (a nivel alto) del desplazamiento del registro de desplazamiento de TX
r_control	out	Byte	Registro de control
r_freq_mod	out	U[24,24]	Paso del DDS para generar la frecuencia moduladora
r_freq_por	out	U[24,24]	Paso del DDS para generar la frecuencia portadora
r_im_am	out	U[16,15]	Índice de modulación de AM
r_im_fm	out	U[16,16]	Índice de modulación de FM

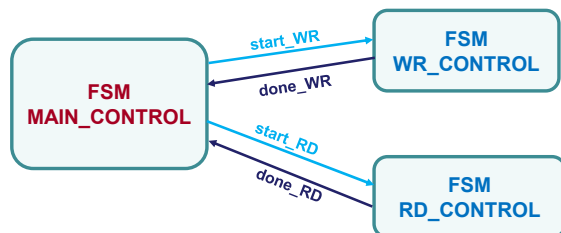
## Módulo CONTROL

El módulo CONTROL es el que gestiona la comunicación. Tendrá que detectar el byte de instrucción enviado por el PC y proceder a dar las órdenes a los módulos RS232 y REGS\_CONF para ejecutar la instrucción detectada. Su comportamiento es el siguiente:

- Si se recibe una instrucción de escritura deberá esperar a que se reciban los 11 bytes e ir almacenándolos en el registro de desplazamiento RX. Una vez han llegado todos los bytes debe cargarlos en los registros de configuración y volver a un estado que le permita decodificar otra instrucción.
- Si se recibe una instrucción de lectura deberá almacenar los contenidos de los registros de configuración en el registro de desplazamiento de TX e ir transmitiendo los 11 bytes teniendo en cuenta que el bloque RS232COM no esté en estado ocupado. Una vez se hayan transmitido todos los bytes debe volver a un estado que le permita decodificar otra instrucción.
- Si no se recibe ni la instrucción de escritura ni de lectura deberá ir a un estado de ERROR\_DE\_INSTRUCCIÓN, del que podrá salir si se le envía una instrucción correcta.

Para simplificar el diseño del módulo CONTROL éste se ha dividido en módulos que interactúan entre sí, tal y como se muestra en la figura:

- MAIN\_CONTROL es el módulo encargado de decodificar la instrucción y de iniciar las tareas de lectura o escritura o generar el estado de error de instrucción.
- WR\_CONTROL genera las señales de control necesarias para completar el proceso de escritura de los bytes recibidos en los registros de configuración
- RD\_CONTROL genera las señales de control necesarias para completar el proceso de lectura de los registros de configuración y su transmisión por el puerto serie



El interfaz del módulo CONTROL es el siguiente:

Módulo CONTROL			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Entrada de reset activa a nivel alto
rxdw	in	8 bits	Dato (byte) recibido
txena	out	bit	Señal de validación (activa a nivel alto) de transmisión
txbusy	in	bit	Señal de estado (activa a nivel alto) que indica que el bloque RS232COM está ocupado transmitiendo un byte
rxrdy	in	bit	Señal (activa a nivel alto) que indica que se ha recibido un dato por el puerto serie

<b>shift_rxregs</b>	out	bit	Señal de activación (a nivel alto) del desplazamiento del registro de desplazamiento de RX
<b>load_conregs</b>	out	bit	Señal de activación (a nivel alto) de la carga del registro de configuración
<b>load_txregs</b>	out	bit	Señal de activación (a nivel alto) de la carga del registro de desplazamiento de TX
<b>shift_txregs</b>	out	bit	Señal de activación (a nivel alto) del desplazamiento del registro de desplazamiento de TX
<b>sleds</b>	out	9 bits	Salidas (9 bits) de monitorización de los estados de las máquinas de estado

**Ficheros** necesarios para realizar las prácticas:

- E5.zip: Proyecto de Quartus para verificación con la tarjeta DE2\_115
- top\_DE2115.v: Modelo Verilog completo
- CONF\_CONTROL.v: Modelo Verilog completo
- REGS\_CONF.v: Interfaz Verilog del módulo REGS\_CONF
- \*REGS\_CONF\_TB.v: Banco de pruebas del módulo REGS\_CONF
- CONTROL.v: Modelo Verilog del módulo CONTROL
- \*CONTROL\_TB.v: Banco de pruebas del módulo CONTROL
- MAIN\_CONTROL.v: Interfaz Verilog del módulo MAIN\_CONTROL
- \*MAIN\_CONTROL\_TB.v: Banco de pruebas del módulo MAIN\_CONTROL
- WR\_CONTROL.v: Interfaz Verilog del módulo WR\_CONTROL
- \*WR\_CONTROL\_TB.v: Banco de pruebas del módulo WR\_CONTROL
- RD\_CONTROL.v: Interfaz Verilog del módulo RD\_CONTROL
- \*RD\_CONTROL\_TB.v: Banco de pruebas del módulo RD\_CONTROL
- CONF\_CONROL\_VERIFICA.v, CONF\_CONTROL\_TB.sv y CONF\_CONTROL\_TB.do: Ficheros para verificar los módulos REGS\_CONF y CONTROL
- use\_serial\_port.m: script de Matlab con un ejemplo de configuración, escritura y lectura del puerto serie

NOTA: Los ficheros marcados con un “\*” no están disponibles en Poliformat y los tendrá que generar el alumno.

### Tareas a realizar

Teniendo en cuenta las especificaciones y descripción del sistema realizada anteriormente:

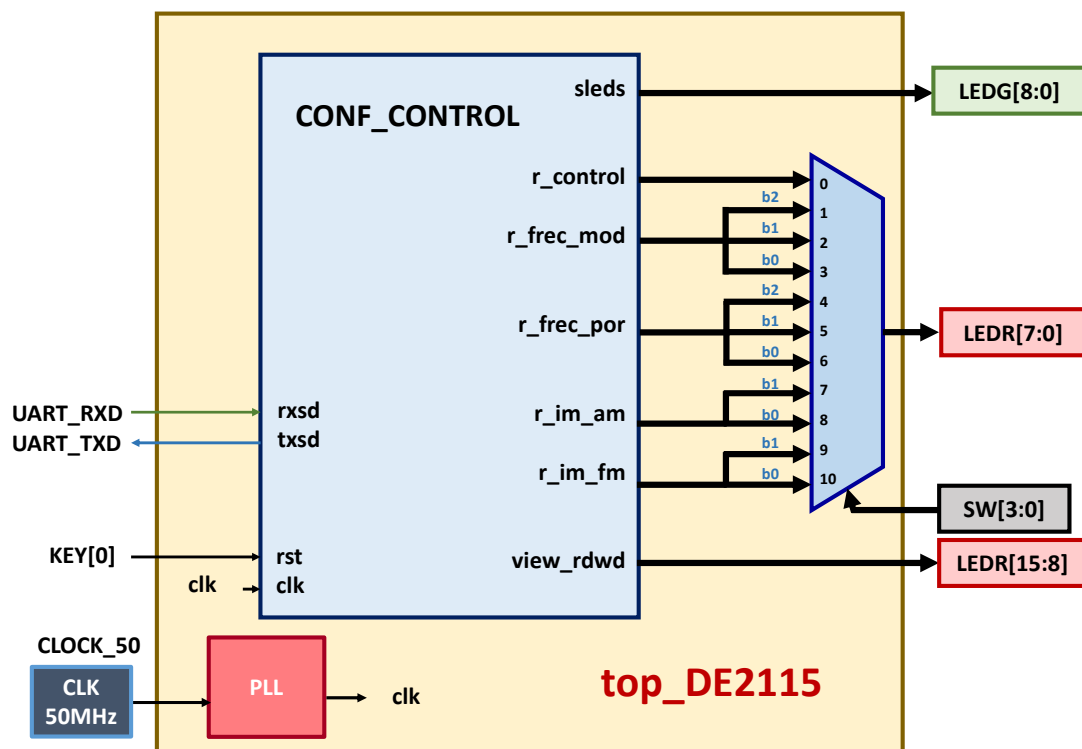
- 1) Modele en Verilog el módulo CONF\_REGS utilizando el interfaz definido en REGS\_CONF.v y realice un banco de pruebas para validar su comportamiento.
- 2) Abra el fichero CONTROL.v y dibuje el esquema de conexiones entre los módulos MAIN\_CONTROL, WR\_CONTROL y RD\_CONTROL. Tenga en cuenta las conexiones entre dichos bloques para realizar la siguiente tarea.

3) Para cada uno de los tres módulos que forman CONTROL: a) MAIN\_CONTROL; b) WR\_CONTROL; y c) RD\_CONTROL: Dibuje el diagrama de estados que describe su comportamiento, modélelo en Verilog y compruebe su funcionamiento escribiendo un banco de pruebas. Tenga en cuenta que las salidas sled de cada módulo se han incluido para facilitar la depuración del sistema. A ellas debe asignar un código que indique en qué estado está la máquina de estados.

4) Depure el funcionamiento del módulo CONTROL y REGS\_CONF utilizando el banco de pruebas del fichero CONF\_CONTROL\_TB.sv, que instancia el módulo CONF\_CONTROL\_VERIFICA.v. Este fichero contiene los dos módulos a verificar conectados. Los resultados del test se visualizan utilizando el fichero CONF\_CONTROL\_TB.do. El banco de pruebas transmite una trama de escritura enviando los valores del 1 al 11 en cada byte transmitido y, posteriormente, inicia una trama de lectura y comprueba que se han recibido correctamente los 11 bytes. También, envía ciertos mensajes por la consola indicando si se han transmitido correctamente los bytes desde la FPGA y si los registros quedan bien configurados.

### Verificación del módulo CONF\_CONTROL

La verificación del funcionamiento del módulo CONF\_CONTROL se va a realizar utilizando la tarjeta DE2-115. El módulo CONF\_CONTROL se ha instanciado en el módulo top\_DE2115, cuyo esquema se muestra a continuación. Como puede observarse, las salidas sled, que monitorizan el estado del bloque CONTROL, están conectadas a los LEDs verdes, los bytes de los registros de configuración están conectados, a través de un multiplexor controlado por los interruptores SW[3:0], a los LEDs rojos LEDR[7:0] y el byte recibido por el puerto serie está conectado a los LEDs rojos LEDR[15:8].





5) El fichero E5.zip contiene un fichero llamado E5.qsf. Abra un nuevo proyecto con la herramienta Quartus y, posteriormente, ciérrelo. Copie en el directorio del proyecto los ficheros desarrollados en las secciones anteriores. Busque el fichero con el nombre del proyecto extensión “qsf” y sustituya su contenido por el de E5.qsf. Abra el proyecto y compile el módulo top\_DE2115 y compruebe que no aparecen errores de sintaxis y que el circuito puede funcionar a una frecuencia de al menos 25 MHz.

6) El fichero use\_serial\_port.m de Matlab contiene un ejemplo de configuración y uso del puerto serie para escribir y leer bytes. Modifique dicho fichero para realizar las operaciones de escritura y lectura de los 11 registros y utilícelo para depurar el módulo CONF\_REGS. Recuerde que siempre hay que enviar primero el byte con la instrucción a realizar.