

Memoria Práctica E5

Escritura y lectura de registros de configuración

David Martínez Esteso

Néstor García García

Tabla de contenido

Sección 1: Descripción del módulo3

Sección 2: Interfaz.8

Sección 3: Recursos hardware11

Sección 4: Frecuencia de operación12

Sección 5: Verificación.....13

Sección 6: Resolución de problemas encontrados.16

Sección 1: Descripción del módulo

El propósito de la siguiente práctica es el desarrollo de un bloque que establezca la comunicación entre el PC y el dispositivo FPGA haciendo uso de el puerto serie RS232 del cual se nos ha proporcionado el módulo que lo implementa, y de los registros de configuración para el modulador FM/AM. La frecuencia de reloj necesaria para su correcto funcionamiento son 25 MHz.

La estructura del módulo a implementar, denominado CONF_CONTROL es la siguiente:

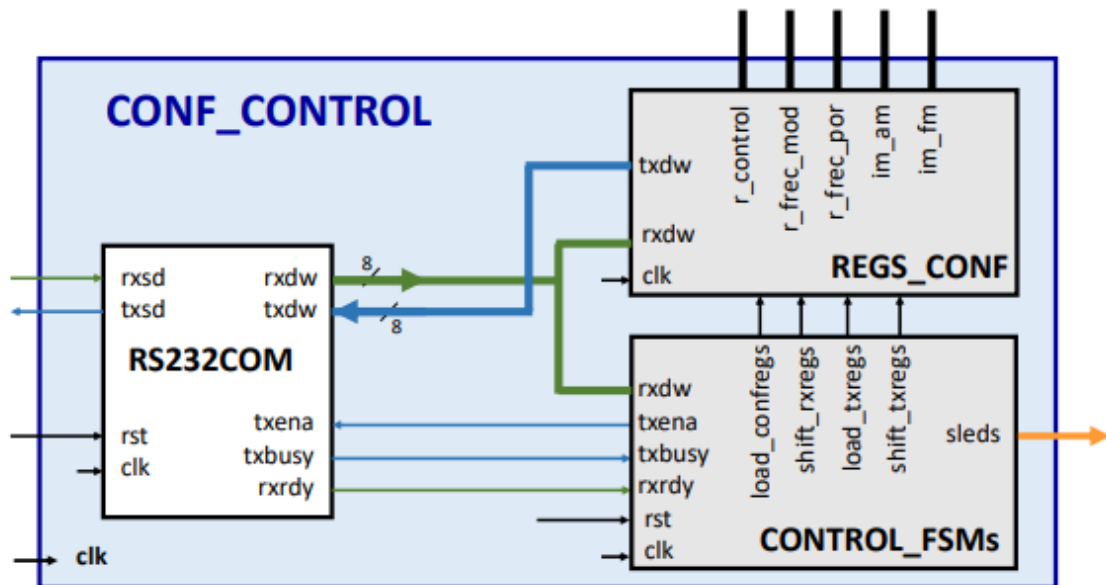


Ilustración 1: estructura módulo CONF_CONTROL

En él se establecen las conexiones entre los siguientes submódulos:

- **Módulo RS232:** se ocupa de establecer la comunicación entre el PC y el dispositivo FPGA. Configurado para transmitir/recibir bytes a 57600 baudios.

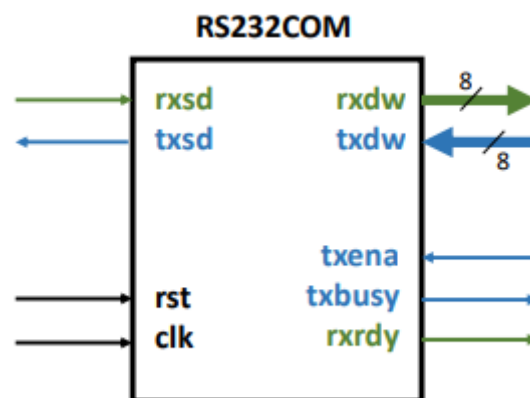


Ilustración 2: estructura módulo RS232COM

- **Módulo REGS_CONF:** contiene los registros de desplazamiento necesarios para la recepción y transmisión serie byte a byte de los bytes de configuración y los registros de configuración.

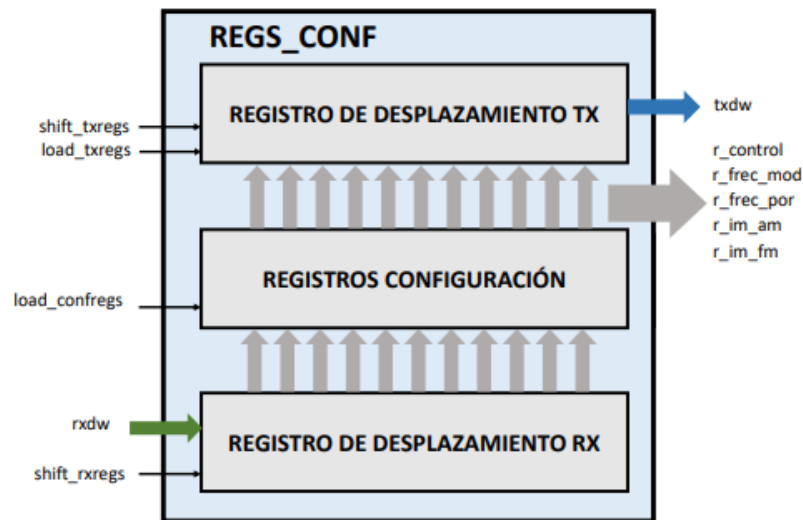


Ilustración 3: estructura módulo REGS_CONF

- **Módulo CONTROL_FMS:** se trata de una máquina de estados general y dos máquinas de estados internas, una para la lectura y otra para la escritura de los datos, las cuales se encargan de la gestión de la comunicación detectando el byte de instrucción enviado por el PC y ordenando las instrucciones pertinentes a los módulos RS232 y REGS_CONF.

A continuación se muestran las estructuras de las máquinas de estados desarrolladas:

Diagrama de estados de la máquina de estados MAIN_CONTROL:

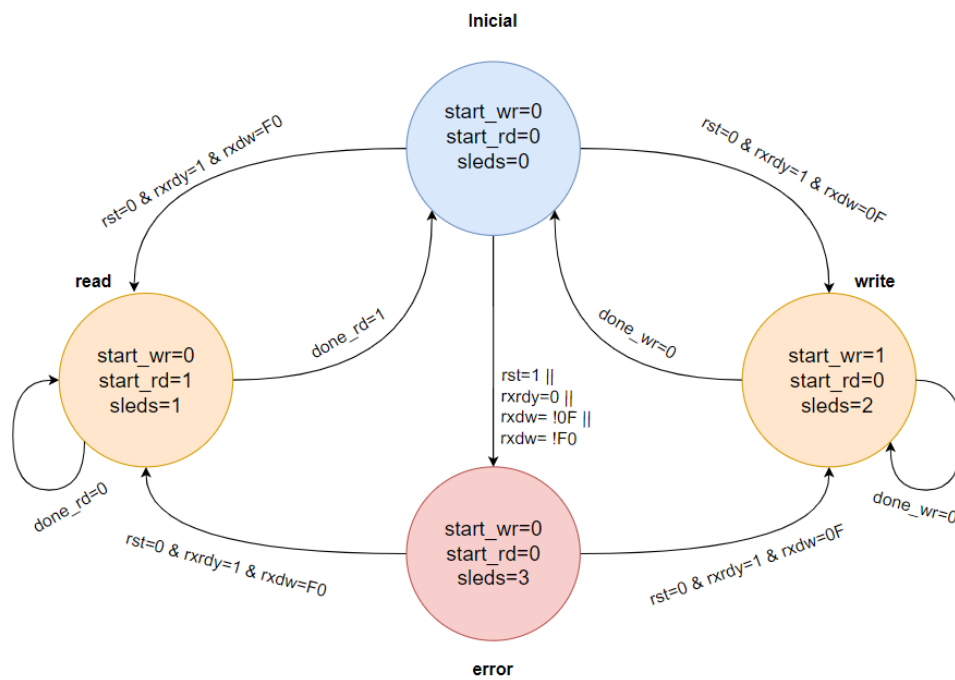


Ilustración 4: diagrama de estados MAIN_CONTROL

Diagrama de estados de la máquina de estados RD_CONTROL:

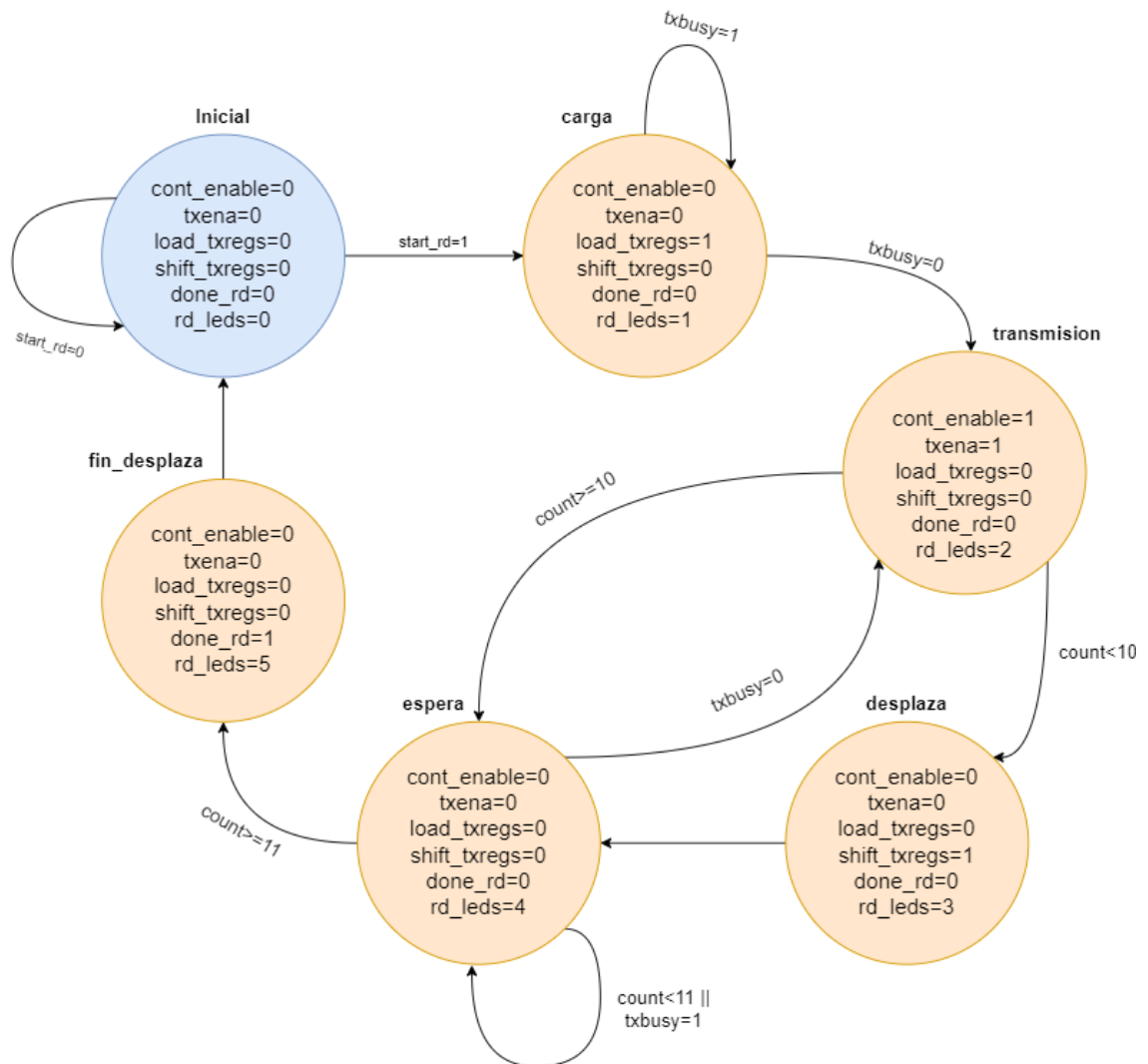


Ilustración 5: diagrama de estados RD_CONTROL

Diagrama de estados de la máquina de estados WR_CONTROL:

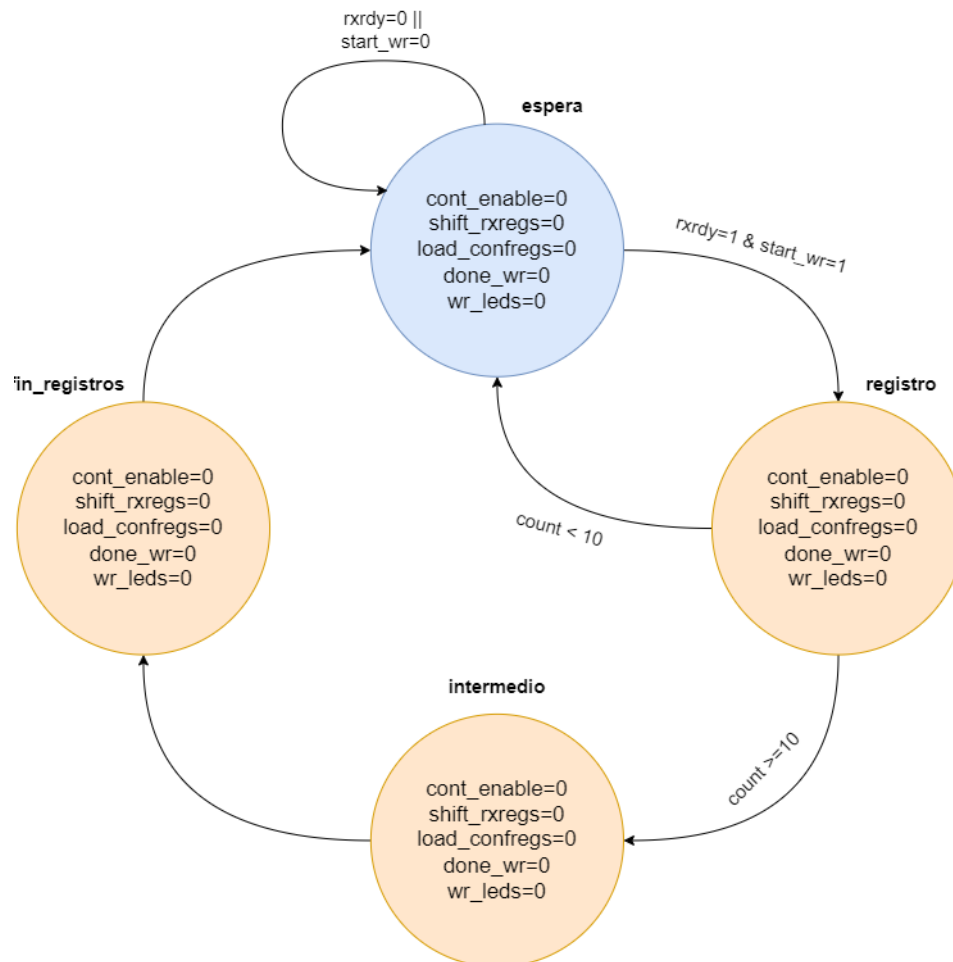


Ilustración 6: diagrama de estados WR_CONTROL

Sección 2: Interfaz.

En la siguiente sección se define el interfaz, sus formatos y sus parámetros de los distintos módulos del proyecto Verilog:

Módulo top_DE2115: INTERFAZ			
Nombre	Tipo	Formato	Descripción
UART_RXD	in	bit	Señal de entrada en serie rx
CLCK_50	in	bit	Reloj principal de 50 MHz
KEY	in	U[1,0]	Configuración botones
SW	in	U[4,0]	Configuración switches
UART_TXD	out	bit	Señal de salida en serie tx
LEDG	out	U[9,0]	Configuración leds verdes
LEDR	out	U[16,0]	Configuración leds rojos

Módulo CONF_CONTROL: INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Reset síncrono, activo a nivel alto
rxsd	in	bit	Dato (byte) recibido
txsd	out	bit	Dato (byte) a transmitir
sleds	out	U[9,0]	Valor leds de estados
r_control	out	U[8,0]	Registro de control
r_frec_mod	out	U[24,0]	Paso del DDS para generar la frecuencia de las señales de test
r_frec_por	out	U[24,0]	Paso del DDS para generar la frecuencia portadora
r_im_am	out	U[16,0]	Índice de modulación de AM
r_im_fm	out	U[16,0]	Índice de modulación de FM
view_rxdw	out	U[8,0]	Señal salida valores leds rojos izquierda

Módulo REGS_CONF: INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
load_confregs	in	Bit	Señal de activación (a nivel alto) de la carga del registro de configuración
rxdw	in	U[8,0]	Dato (byte) recibido
shift_rxregs	in	bit	Señal de activación (a nivel alto) del desplazamiento del registro RX
load_txregs	in	bit	Señal de activación (a nivel alto) de la carga del registro de configuración
shift_txregs	in	bit	Señal de activación (a nivel alto) del desplazamiento del registro TX
txdw	out	U[8,0]	Dato (byte) a transmitir
r_control	out	U[8,0]	Registro de control
r_freq_mod	out	U[24,0]	Paso del DDS para generar la frecuencia de las señales de test
r_freq_por	out	U[24,0]	Paso del DDS para generar la frecuencia portadora
r_im_am	out	U[16,0]	Índice de modulación de AM
r_im_fm	out	U[16,0]	Índice de modulación de FM

Módulo CONTROL: INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	Bit	Entrada de reset activa a nivel alto
rxdw	in	U[8,0]	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
shift_rxregs	out	bit	Señal de activación (a nivel alto) del desplazamiento del registro RX
load_conregs	out	bit	Señal de activación (a nivel alto) de la carga del registro de configuración
load_txregs	out	bit	Señal de activación (a nivel alto) de la carga del registro de desplazamiento de TX

shift_txregs	out	bit	Señal de activación (a nivel alto) del desplazamiento del registro TX
sleds	out	U[9,0]	Salidas (9 bits) de monitorización de los estados de las máquinas de estado

Módulo MAIN_CONTROL:
INTERFAZ

Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Entrada de reset activa a nivel alto
rxrdy	in	bit	Señal (activa a nivel alto) que indica que se ha recibido un dato por el puerto serie
rxdw	in	U[8,0]	Dato (byte) recibido
done_wr	in	bit	Señal finalización proceso escritura
done_rd	out	bit	Señal finalización proceso lectura
start_wr	out	bit	Señal inicio proceso escritura
start_rd	out	bit	Señal inicio proceso lectura
sleds		U[3,0]	Señales leds estado máquinas

Módulo WR_CONTROL: INTERFAZ

Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Entrada de reset activa a nivel alto
rxrdy	in	bit	Señal (activa a nivel alto) que indica que se ha recibido un dato por el puerto serie
shift_rxregs	out	bit	Señal de activación (a nivel alto) del desplazamiento del registro RX
done_wr	out	bit	Señal finalización proceso escritura
load_confregs	out	bit	Señal de activación (a nivel alto) de la carga del registro de configuración
start_wr	in	bit	Señal inicio proceso escritura
sleds	out	U[3,0]	Señales leds estado máquinas

Módulo RD_CONTROL: INTERFAZ			
Nombre	Tipo	Formato	Descripción
clk	in	bit	Entrada de reloj
rst	in	bit	Entrada de reset activa a nivel alto
txbusy	in	bit	Señal de estado (activa a nivel alto) que indica que el bloque RS232COM está ocupado transmitiendo un byte
txena	out	bit	Señal de validación (activa a nivel alto) de transmisión
load_txregs	out	bit	Señal de activación (a nivel alto) de la carga del registro de desplazamiento de TX
done_rd	out	bit	Señal finalización proceso lectura
shift_txregs	in	bit	Señal de activación (a nivel alto) del desplazamiento del registro TX
start_rd	out	bit	Señal inicio proceso lectura
sleds	out	U[3,0]	Señales leds estado máquinas

Sección 3: Recursos hardware

Teniendo en cuenta los numerosos ficheros utilizados en el desarrollo del proyecto se estima el uso de una gran cantidad de elementos lógicos (LEs).

Una vez compilado el proyecto en Quartus se ha ejecutado el sumario del uso de recursos total como se muestra a continuación:

Resource	Usage
Estimated Total logic elements	1,740
Total combinational functions	917
▼ Logic element usage by number of LUT inputs	
-- 4 input functions	400
-- 3 input functions	309
-- <=2 input functions	208
▼ Logic elements by mode	
-- normal mode	807
-- arithmetic mode	110
▼ Total registers	1387
-- Dedicated logic registers	1387
-- I/O registers	0
I/O pins	33
Total memory bits	753664
Embedded Multiplier 9-bit elements	0
▼ Total PLLs	1

Ilustración 7: sumario recursos proyecto Quartus

Comentando la tabla anterior el proyecto hace uso de 1387 registros y un total de 1740 elementos lógicos (LEs), como hemos supuesto se hace uso de una gran cantidad de recursos en el presente proyecto.

Sección 4: Frecuencia de operación

Para la medida de la frecuencia máxima de operación se ha diseñado el módulo wrap registrando las distintas señales utilizadas en el módulo top del proyecto, para así simular todas las entradas que se incorporan al circuito en el mismo ciclo.

La frecuencia de operación del sistema obtenida mediante la herramienta Time Quest Timing analyzer es de 96.65 MHz. Se cumple por lo tanto la especificación de diseño para $F_{max} \geq 25$ MHz.

El camino crítico se encuentra entre conexiones realizadas por el verificador signaltap

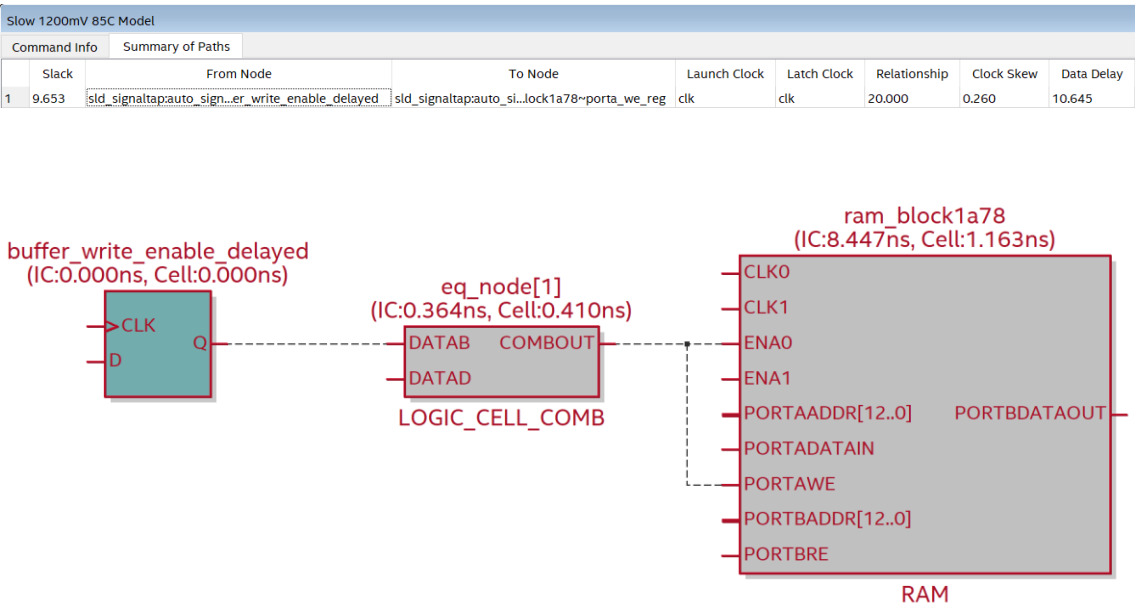


Ilustración 8: Visión RTL del camino crítico del proyecto

Sección 5: Verificación

Bloque REGS CONF.v:

Teniendo en cuenta el comportamiento de este módulo de registros de desplazamiento y las asignaciones de los diferentes bytes de configuración se ha desarrollado el fichero de testbench REGS_CONF_TB, en el cual las formas de onda de las señales se muestran de la siguiente manera:

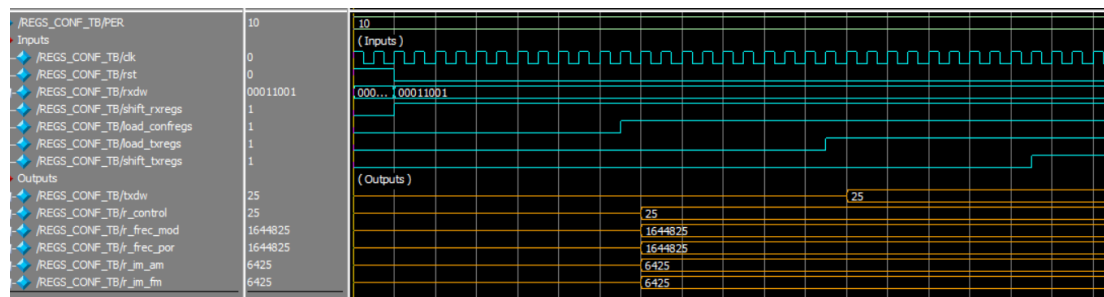


Ilustración 9: Formas de onda señales bloque REGS_CONF

Se puede observar que en las señales de output se asignan los valores correspondientes a los bytes de configuración.

Bloque RD CONTROL:

Una vez desarrollada la máquina de estados del proceso de lectura como se ha mostrado en la primera sección de esta memoria se ha implementado esta estructura en código Verilog e implementado el testbench de verificación se arrojan las siguientes señales:

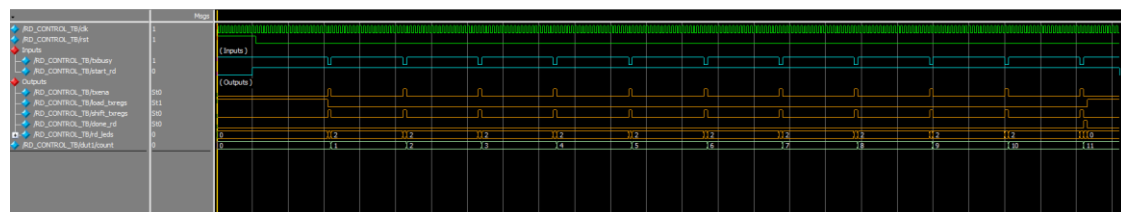


Ilustración 10: Formas de onda señales bloque RD_CONTROL

Puede observarse el correcto comportamiento de la señales txbusy y txena y en consecuencia el correcto funcionamiento de la señal shift_txregs realizando el desplazamiento de los datos por el puerto RX, mostrándose también los cambios de la señal de estado, los valores del contador interno así como de la activación de la señal done rd al final del proceso de lectura.

Bloque WR CONTROL:

Una vez desarrollada la máquina de estados del proceso de escritura como se ha mostrado en la primera sección de esta memoria se ha implementado esta estructura en código Verilog e implementado el testbench de verificación se arrojan las siguientes señales:

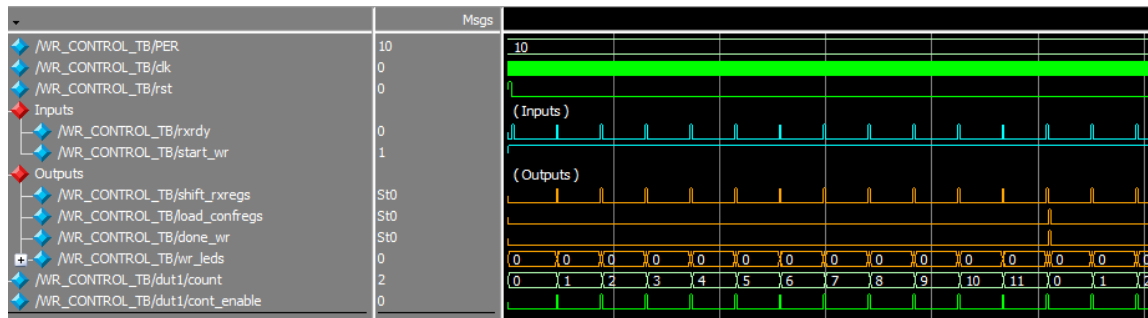


Ilustración 11: Formas de onda señales bloque WR_CONTROL

Puede observarse el correcto comportamiento de la señales rxrdy y cont_enable en consecuencia el correcto funcionamiento de la señal shift_rxregs realizando el desplazamiento de los datos por el registro de desplazamiento TX, mostrándose también los cambios de la señal de estado, los valores del contador interno así como de la activación de la señales done_wr y load_confregs al final del proceso de escritura.

Bloque MAIN CONTROL:

Una vez desarrollada la máquina de estados que gestiona la ejecución de los procesos de escritura y lectura como se ha mostrado en la primera sección de esta memoria se ha implementado esta estructura en código Verilog e implementado el testbench de verificación se arrojan las siguientes señales:

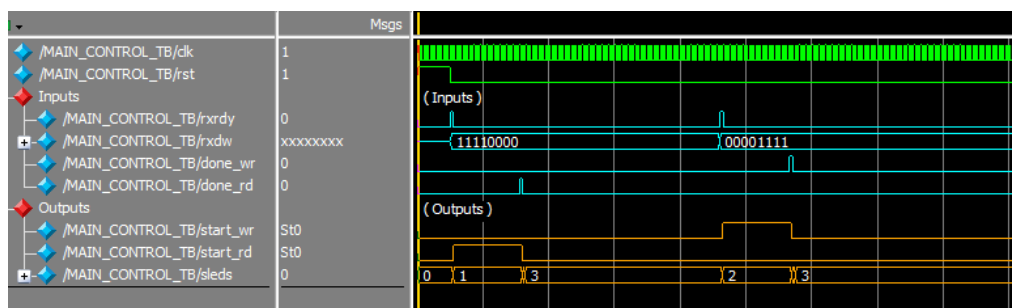


Ilustración 12: Formas de onda señales bloque MAIN CONTROL

En estas formas de onda simplemente se observa como se comportan las señales de final de proceso de escritura y lectura donde wr y rd respectivamente así como de

las señales de inicialización de estos procesos start_wr y start_rd así como de los cambios en el valor de la señal de estados viéndose claramente cuando se encuentra la máquina en el estado de error cuando son se encuentran en ninguno de los procesos mencionados.

Bloque CONF_CONTROL:

Una vez comprobado el correcto funcionamiento de los bloques desarrollados anteriormente se ha comprobado el correcto funcionamiento del proyecto completa haciendo uso del fichero de testbench proporcionado junto a el enunciado de esta terea. Una vez realizados las conexiones pertinentes se visualiza el siguiente comportamiento de las señales:

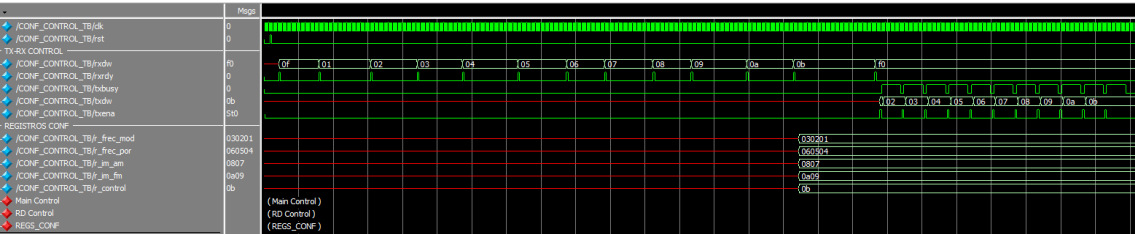


Ilustración 13: Formas de onda señales testbench CONF_CONTROL

Se comprueba el correcto comportamiento de los procesos de lectura y escritura de los datos mostrados en rxdw y txdw respectivamente, así como de el funcionamiento de las señales de ambos procesos y de la asignación de los datos correspondientes a los bytes de configuración.

SignalTap:

Finalmente se ha programado la tarjeta de la FPGA DE2-115 y se ha procedido ha realizar la verificación real del sistema haciendo uso de la herramienta SignalTap a través de conexión USB y una estructura de 16K muestras mostrando el comportamiento de los nodos seleccionados del sistema. Una vez configurados estos parámetros para la verificación. La configuración de los procesos de escritura y lectura se ha realizado a través del fichero de Matlab el cuál envía y recibe los datos.

El comportamiento de las señales en los nodos durante la ejecución del proceso de lectura es el siguiente:

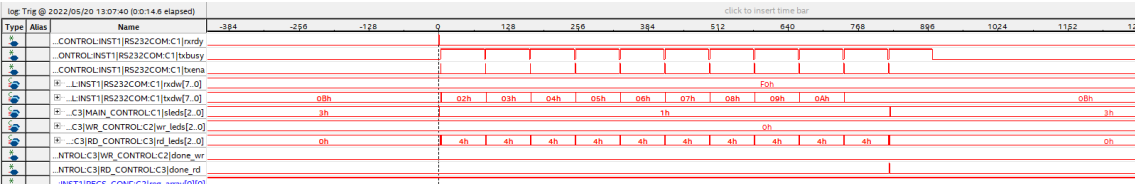


Ilustración 14: comportamiento señales SignalTap proceso de lectura

Se comprueba con toda precisión como se realiza el proceso de lectura de forma correcta habilitando el paso de los datos en los momentos que corresponden, activándose finalmente la señal done_wr.

El comportamiento de las señales en los nodos durante la ejecución del proceso de escritura es el siguiente:

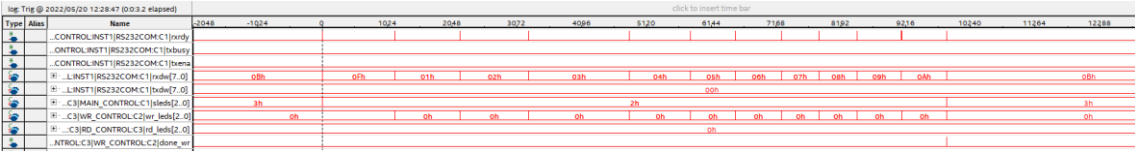


Ilustración 15: comportamiento señales SignalTap proceso de escritura

Se comprueba el correcto comportamiento de las señales en este proceso de escritura. Se aprecian los flancos de rxrdy indicando que se ha recibido un nuevo dato por el puerto serie y la activación de la señal done_wr una vez ha concluido el proceso de escritura.

Sección 6: Resolución de problemas encontrados.

A continuación se presentan los problemas encontrados en la realización de la práctica y su solución:

-En el desarrollo del módulo RD_CONTROL a la hora de verificar su funcionamiento se apreciaba que las señales txbusy y txena no se comportaban de forma correcta y no se producía un correcto envío de los datos a través del puerto serie RX. Para solucionar este inconveniente se ha añadido un estado de espera a la máquina de estados el cuál comprueba el valor de la cuenta del contador interno haciendo regresar al estado de transmisión si la señal txbusy es 0, es decir si no está ocupado el puerto serie, y se pregunte constantemente el valor de la cuenta del contador interno manteniéndose en este estado de espera si el valor es menor a 10 y pasando al estado fin_desplaza un ves se llega al final de la cuenta enviando así el dato procesado de forma correcta.

-Durante las primeras comprobaciones del sistema haciendo uso del SignalTap se apreciaban flancos incorrectos en algunas de las señales, principalmente en las señales que indican el valor de los estados de las diferentes máquinas de estados. Esto era debido a que es necesario siempre incluir los casos else en las estructuras if y los casos de default en loas estructuras case de las máquinas de estados. Una vez añadidas estas líneas se eliminaros estos flancos erróneos.

-Como último inconveniente detectamos que tan solo se realizaban correctamente los procesos de lectura y escritura en su primera ejecución, es decir, si no se activaba la señal de reset los valores de los registros de configuración no se asignaban correctamente. Esto era debido a un mal reseteo de los contadores internos de las máquinas de estado, por lo que solo funcionaban correctamente una sola vez. Una vez implementados correctamente estos resets el proyecto ya es capaz de realizar tantos procesos de escritura y lectura como precise el usuario sin problemas en las asignación de los registros de configuración.