Máster Universitario en Ingeniería de Sistemas Electrónicos

**Imagen que contiene Texto

Descripción generada automáticamente**

**Práctica E5: Escritura y lectura de registros de configuración**

Jose Luis, Rocabado Rocha

Gianmarco Leopoldo, Sangoi Da Roza

05/30/2022

# **ÍNDICE**

[**ÍNDICE** 2](#_Toc104820539)

[**ÍNDICE DE FIGURAS** 3](#_Toc104820540)

[Descripción del módulo 4](#_Toc104820541)

[Interfaz 9](#_Toc104820542)

[Recursos estimados 11](#_Toc104820543)

[Frecuencia máxima 11](#_Toc104820544)

[Camino critico 12](#_Toc104820545)

[Verificación 13](#_Toc104820546)

[Resolución de problemas encontrados 16](#_Toc104820547)

# **ÍNDICE DE FIGURAS**

[Ilustración 1. Modulo top CONF\_CONTROL 4](#_Toc104821215)

[Ilustración 2. Recepción y envió de datos mediante la interfaz RS232. 4](#_Toc104821216)

[Ilustración 3. Módulo REGS\_CONF (3 registros) 5](#_Toc104821217)

[Ilustración 4. Máquina de estados MAIN\_CONTROL. 6](#_Toc104821218)

[Ilustración 5. Maquina de estados WR\_CONTROL. 6](#_Toc104821219)

[Ilustración 6. Máquina de estados RD\_CONTROL. 7](#_Toc104821220)

[Ilustración 7. Bytes de escritura y lectura 7](#_Toc104821221)

[Ilustración 8. Orden de transmisión de los 11 bytes. 8](#_Toc104821222)

[Ilustración 9. Tabla de recursos del módulo top CONF\_CONTROL 11](#_Toc104821223)

[Ilustración 10. Frecuencia máxima del módulo top\_DE2115 12](#_Toc104821224)

[Ilustración 11. Camino crítico del módulo top\_DE2115 12](#_Toc104821225)

[Ilustración 12. Testbench máquina de estados MAIN\_CONTROL 13](#_Toc104821226)

[Ilustración 13. Testbench máquina de estados RD\_CONTROL 14](#_Toc104821227)

[Ilustración 14. Testbench máquina de estados WR\_CONTROL 14](#_Toc104821228)

[Ilustración 15. Testbench del módulo CONF\_CONTROL 15](#_Toc104821229)

[Ilustración 16. Diagrama del modulo top\_DE2115 15](#_Toc104821230)

[Ilustración 17. Simulación del módulo top\_DE2115 con la herramienta SignalTAP Logic Analyzer 16](#_Toc104821231)

[Ilustración 18. Estructura de shift register dada en la asignatura. 17](#_Toc104821232)

# Descripción del módulo

En esta práctica se realizará el módulo que nos permitirá comunicar la computadora con el dispositivo FPGA. Este módulo permitirá que podamos escribir y leer los registros de configuración del modulador AM/FM a través de un puerto serie RS232.

El módulo elaborado consta de 4 bloques principales:

Diagrama

Descripción generada automáticamente

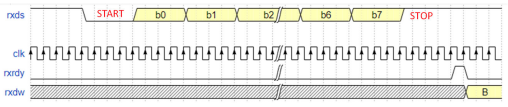
Ilustración 1. Modulo top CONF\_CONTROL

* **PLL**

Este bloque es el encargado de proporcionar la frecuencia de reloj de 230400KHz a partir del reloj de 50MHz de la placa DE2-115 (Ya dado por el docente de la asignatura). Su instanciación realmente será afuera del módulo elaborado: CONF\_CONTROL, esto se debe a que se usará un módulo TOP proporcionado por el docente para poder instanciar nuestro modulo junto al PLL, los switches y LEDs de la placa.

* **RS232COM**

Este bloque descrito en HDL es el encargado de realizar la capa física para la comunicación RS232 (Ya dado por el docente de la asignatura). El funcionamiento de la interfaz se da en las siguientes ilustraciones de recepción y envió respectivamente:



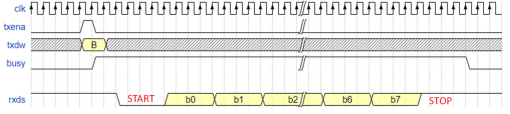


Ilustración 2. Recepción y envió de datos mediante la interfaz RS232.

* **REGS\_CONF**

Este bloque es el que contiene los registros para la configuración del modulador AM/FM. Esta diseñado en base a 3 registros de 11 bytes, desplazamiento de lectura, desplazamiento de escritura y el registro de configuración. En el siguiente diagrama se podrá observar su estructura:

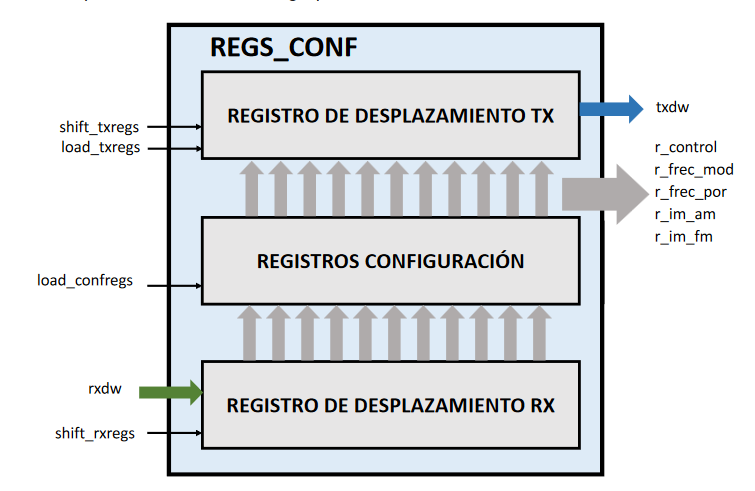


Ilustración 3. Módulo REGS\_CONF (3 registros)

* **CONTROL\_FSMs**

Este bloque se basa en 3 maquinas de estados que controlan el envío y recepción de los bytes del REGS\_CONF:

* Máquina de estado MAIN\_CONTROL (principal):

Imagen que contiene Tabla

Descripción generada automáticamente

Imagen que contiene diferente, tabla, diversos, pequeño

Descripción generada automáticamente

Ilustración 4. Máquina de estados MAIN\_CONTROL.

* Máquina de estado RD\_CONTROL (escritura):

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Un grupo de personas con instrumentos musicales y micrófonos en un escenario

Descripción generada automáticamente con confianza baja

Ilustración 5. Máquina de estados WR\_CONTROL.

* Máquina de estado WR\_CONTROL (lectura):

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

Diagrama

Descripción generada automáticamente

Ilustración 6. Máquina de estados RD\_CONTROL.

El funcionamiento conjunto de las máquinas de estado es de la siguiente manera:

Primero el modulo se encarga de detectar el primer byte transmitido desde la PC (MATLAB) a nuestra FPGA, este nos indica si el procedimiento es el de lectura o escritura del REGS\_CONF:

Diagrama, Forma, Rectángulo

Descripción generada automáticamenteDiagrama, Forma, Rectángulo

Descripción generada automáticamente

Ilustración 7. Bytes de escritura y lectura

Una vez que se interpreta este comando se procede a realizar el desplazamiento de bytes de los registros de lectura y escritura que posee el bloque REGS\_CONF, lo hace en base al siguiente orden:

Diagrama

Descripción generada automáticamente

Ilustración 8. Orden de transmisión de los 11 bytes.

Si el procedimiento es de escritura primero realiza el desplazamiento de los 11 bytes al registro de escritura y luego los carga en el registro de configuración para poder configurar nuestro modulador AM/FM. Si el procedimiento es de lectura, primero se carga los bytes en el registro de lectura desde el registro de configuración y luego se desplazan 1 a 1 para poder ser leídos.

# Interfaz

Texto, Tabla

Descripción generada automáticamenteTexto

Descripción generada automáticamente

Tabla

Descripción generada automáticamente

Tabla

Descripción generada automáticamenteTexto, Tabla

Descripción generada automáticamente con confianza media

# Recursos estimados

Los recursos de nuestro módulo top\_DE2115 fueron los siguientes:

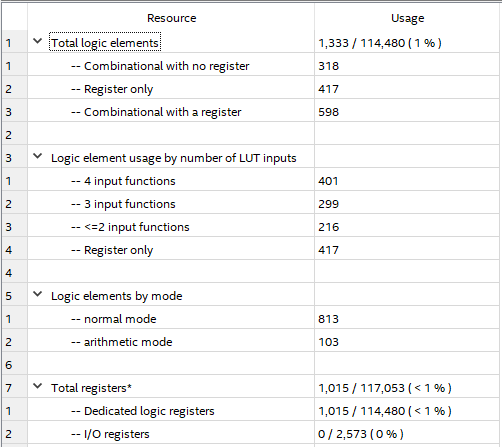


Ilustración 9. Tabla de recursos del módulo top top\_DE2115

Registros aproximados

* Registros por el módulo REGS\_CONF: 8 x 11 x 3 = 264 aprox.
* Registros en del módulo RS232COM: 4+3+2+2+8+2+9+6+4+4 = 44 aprox
* Registros del módulo de control: 9+8 = 17 aprox

# Frecuencia máxima

La placa utilizada en el laboratorio de prácticas consiste en una Cyclone IV DE-115 cuya frecuencia máxima es de 250MHz.

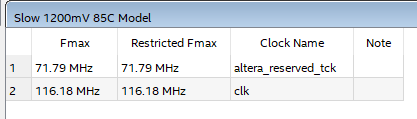


Ilustración 10. Frecuencia máxima del módulo top\_DE2115

Mediante la herramienta *TimeQuest Timing Analyzer*, podremos calcular la frecuencia máxima a la que trabajará el modulo diseñado instanciado con el top\_DE2115. En este caso la frecuencia máxima de trabajo de ~116.18MHz tal y como se observa en la ilustración.

# Camino critico

El camino critico es aquel en que la señal tarda más desde su ingreso hasta su destino. Mediante el *Technology Map Viewer* podremos encontrar en que parte del circuito se encuentra.

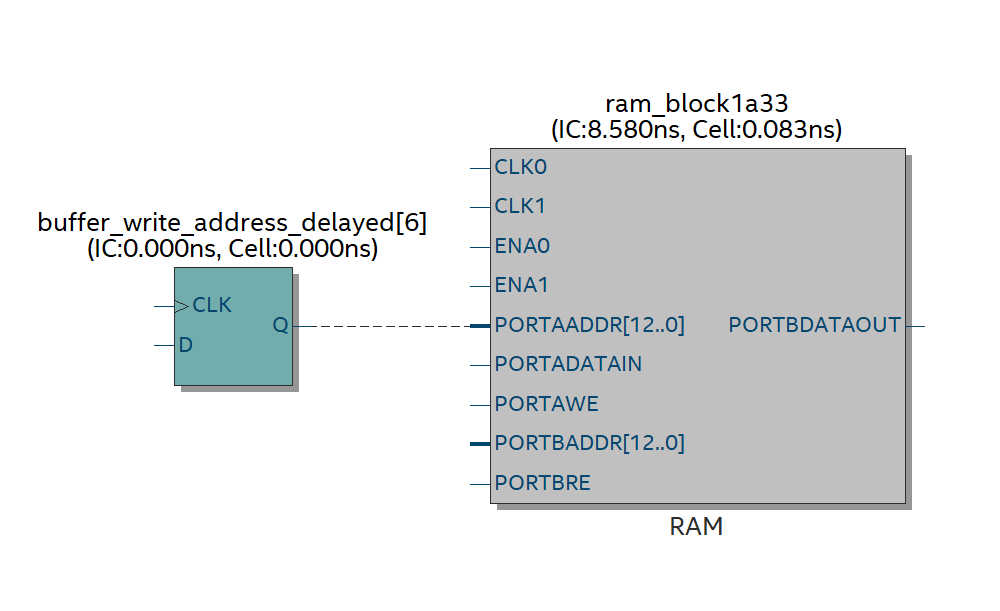


Ilustración 11. Camino crítico del módulo top\_DE2115

# Verificación

Para la verificación de esta entrega se realizaron 3 tipos de verificaciones distintas:

1. Verificación Testbenchs:

El primer método de verificación utilizado fue el de simulación mediante testbenchs con la herramienta *MODELSIM.* Se procedió a realizar un testbench para cada máquina de estado diseñada y luego de comprobar su correcto funcionamiento se usó el testbech dado por el docente (CONF\_CONTROL\_TB.v) para poder verificar el módulo de REGS\_CONTROL y el módulo de control CONTROL\_FSMs.

* MAIN\_CONTROL:

Esta es la máquina de estados que se encarga de leer el byte que contiene el comando de escritura o lectura dirigirse al estado de ese procedimiento, luego de acabar el procedimiento la maquina debe de volver al estado inicial.

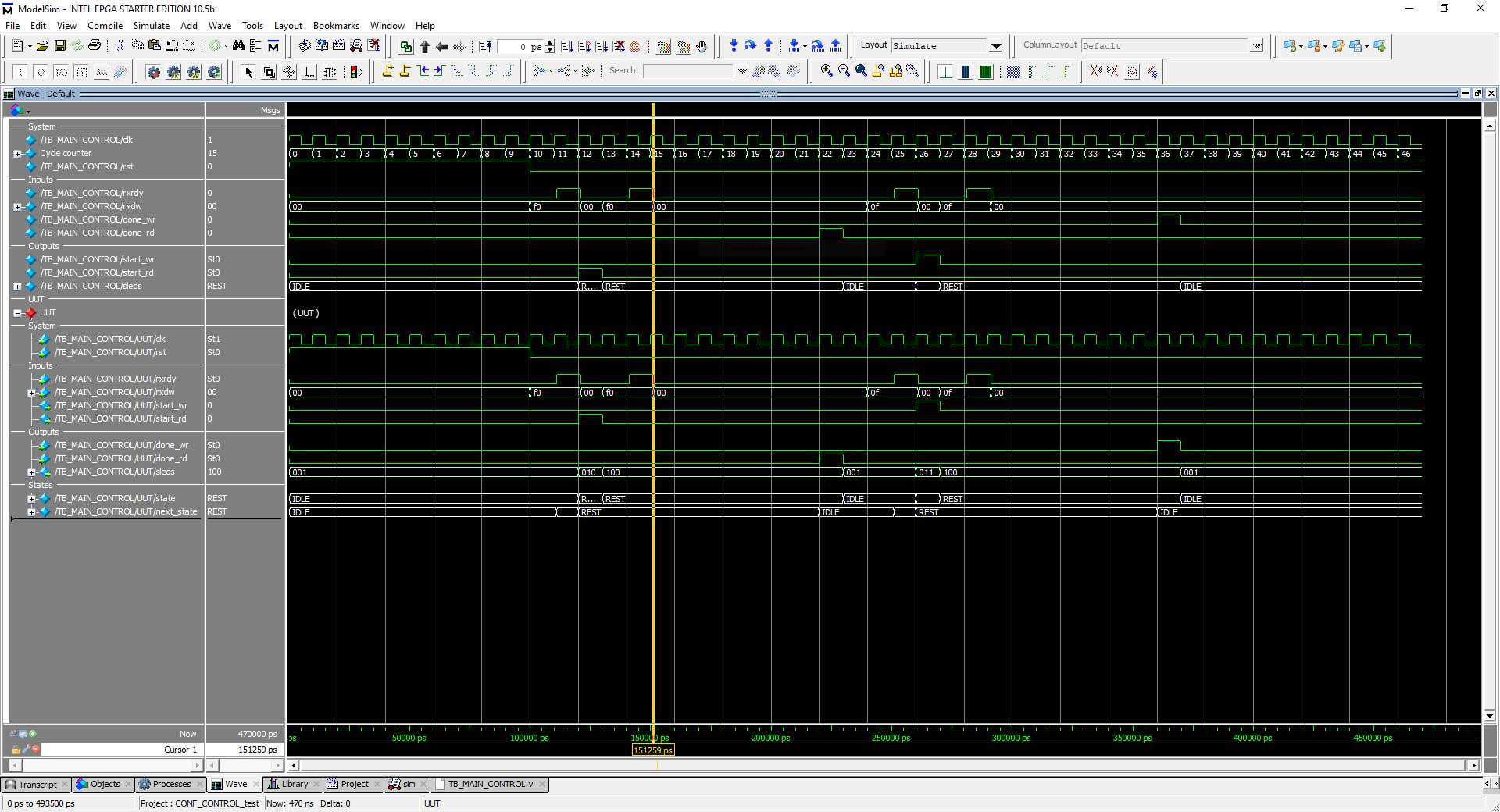


Ilustración 12. Testbench máquina de estados MAIN\_CONTROL

* RD\_CONTROL:

Esta es la máquina de estados que se encarga de controlar la transmisión de los 11 bytes del registro de desplazamiento tx de REGS\_CONF cuando la PC los solicite.

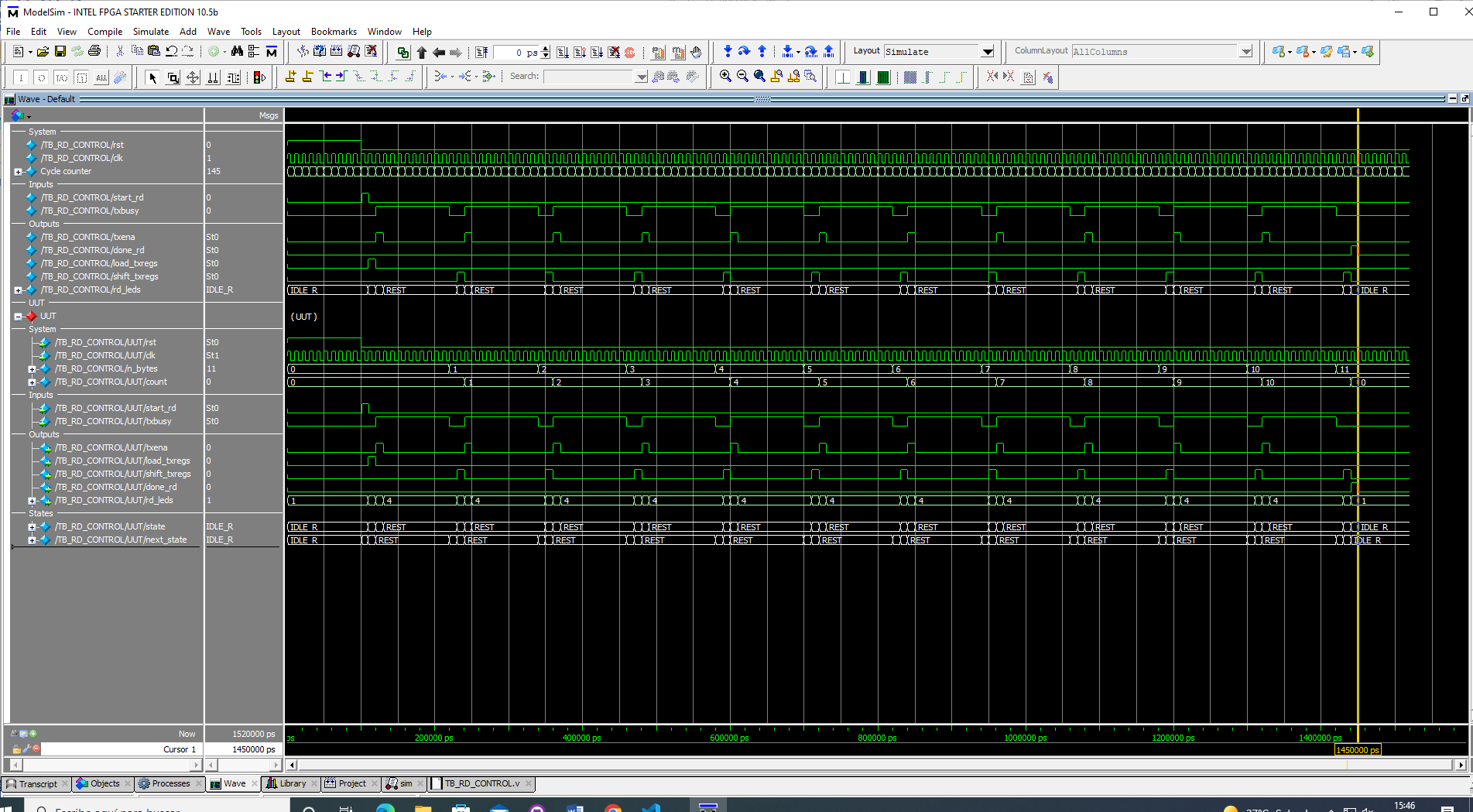


Ilustración 13. Testbench máquina de estados RD\_CONTROL

* WR\_CONTROL:

Esta es la máquina de estados que se encarga de escribir los 11 bytes que provienen de la PC en el módulo CONF\_REGS.

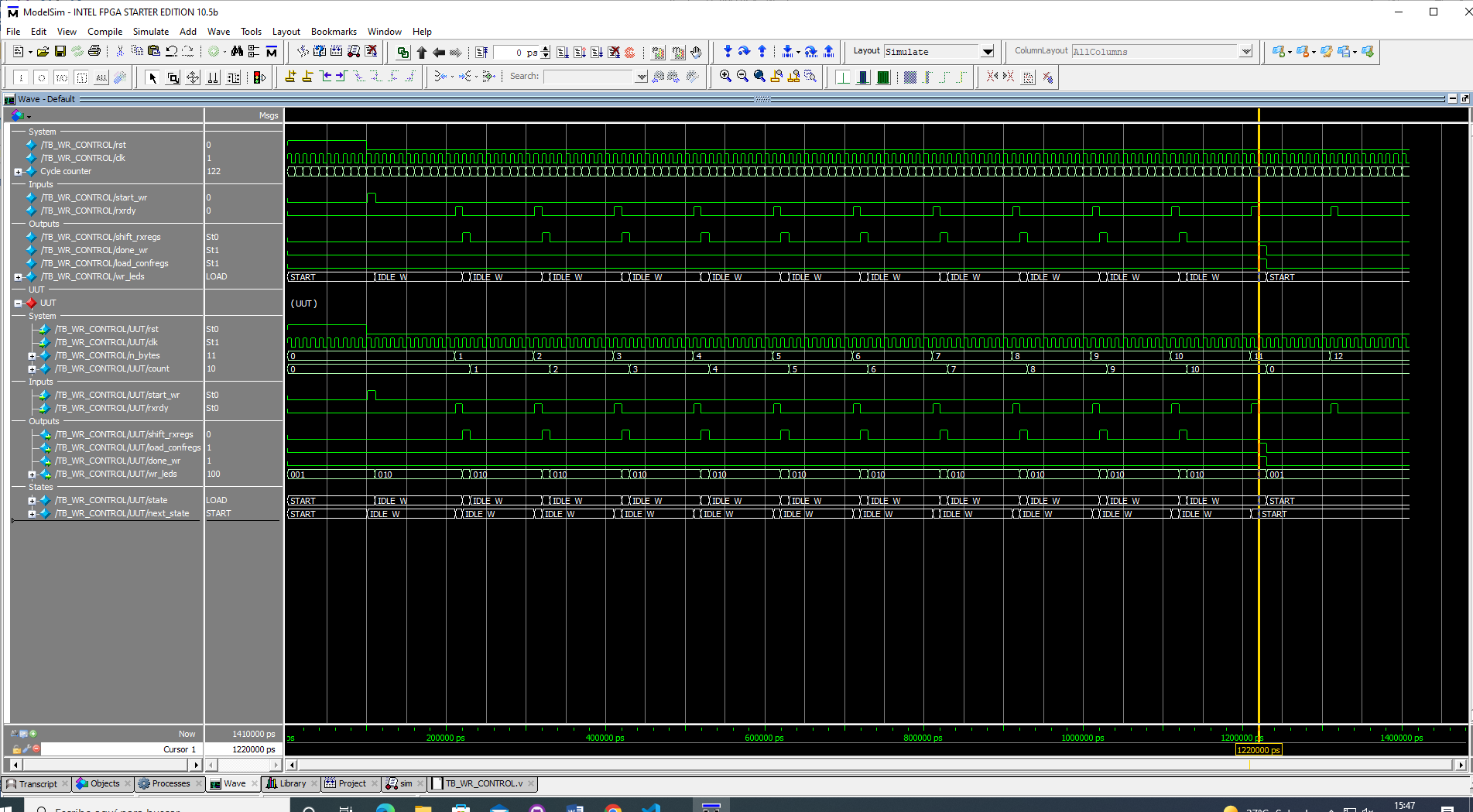


Ilustración 14. Testbench máquina de estados WR\_CONTROL

* CONF\_CONTROL:

Finalmente el módulo top CONF\_CONTROL fue simulado con el testbench que nos proporcionó el docente. Este le inyecta a nuestro modulo el “comando” de escritura junto con los 11 bytes de configuración a transmitir. Luego le pasa el comando de lectura y revisa que los bytes han sido desplazados correctamente desde el registro de desplazamiento tx.

Diagrama, Esquemático

Descripción generada automáticamente

Ilustración 15. Testbench del módulo CONF\_CONTROL

1. Verificación física:

Al diseñar nuestra maquina de estados se le asigno a cada estado un valor de 4 bits que representa 4 LEDs de la placa. Esto se hizo para que en conjunto con el modulo top proporcionado por el docente (top\_DE2115) y el archivo de asignación de pines se pudiera verificar el funcionamiento del módulo físicamente.

Diagrama, Esquemático

Descripción generada automáticamente

Ilustración 16. Diagrama del módulo top\_DE2115

Como se puede observar los verdes LEDs (del 0 al 8) nos dicen el estado, 4 switches seleccionan el byte a mostrar por 8 LEDs rojos (del 0 al 7) y los LEDs rojos de 8 al 15 nos indican el byte recibido por el puerto serie.

1. Verificación con SignalTap:

Esta verificación se hace mediante la herramienta *SignalTap Logic Analyzer* que proporciona *Quartus*, con ella podemos observar la transmisión y recepción de datos que realiza nuestro modulo simplemente cargando el diseño en la placa y corriendo el fichero de Matlab con los bytes de configuració.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 17. Simulación del módulo top\_DE2115 con la herramienta SignalTap Logic Analyzer

El reloj utilizado para la configuración de esta herramienta fue el que arroja la salida del bloque PLL (25KHz aprox.). Para poder visualizar la recepción y transmisión se utilizaron las señales rxrdy y txena respectivamente como señales de disparo cuando están a “1”.

Como se puede observar en la ilustración 11, al cargar el fichero en Matlab con los 11 bytes de prueba se comprueba que nuestro diseño recibe y transmite los 11 bytes.

# Resolución de problemas encontrados

En cuanto a los problemas encontrados el principal fue que al realizar la simulación en *Modelsim* de nuestro modulo Top CONF\_CONTROL se vio como había un problema, esto era debido a que el código en verilog para el módulo REGS\_CONF no se había hecho de una manera óptima (se colocaban 3 registros de 87 bits) lo cual no solamente nos arrojaba un error, sino que también dificultaba su testeo, esto fue arreglado cambiando el código por la estructura dada en clase:

Diagrama

Descripción generada automáticamente

Ilustración 18. Estructura de shift register dada en la asignatura.