Bitácora Módulo 4: Protocolo.

Presentado por:

Cristian David Patiño Londoño  
1088334538

Jairo Alejandro Castrillón  
1112782874

Profesor:   
Ramiro Andrés Barrios Valencia

Ingeniería en Sistemas y Computación

Universidad Tecnológica de Pereira

Pereira-Risaralda

Mayo-2018

Módulo 4:

Protocolo de modularidad: Diseñar un protocolo que permita la interconexión entre módulos colectores para administrar la información de movilidad global del parqueadero y la comunicación de la información al dispositivo central.

Para la solución de este módulo se empezó una investigación por parte del equipo de trabajo del laboratorio para definir que protocolo de comunicación es más efectivo, la tarjeta de desarrollo a utilizar fue la nexys 2 la cual nos permite utilizar

Se descartó protocolo de Ethernet pues la FPGA carece de los puertos que soporta este mismo, entonces se decidió hacer uso del protocolo RS232 (UART) para la conexión entre las dos FPGA´s utilizando un puerto de transmisión y uno de recepción.

Antes de iniciar el desarrollo de este protocolo se resaltan las modificaciones importantes al módulo.

La siguiente modificación fue la asignación de pines de la FPGA nexys 2 que incluye el módulo. Asignamos las entradas de los sensores, los interruptores (Switches) de FPGA pues estos cumplen la misma función que un sensor enviarnos un 1 o 0 lógico, otro cambio importante fue que dejamos de utilizar la librería de la LCD 16x2 de 8 pines y pasamos a utilizar la LCD 16x2 de 4 pines, lo que nos ahorra utilizar todos los pines de la LCD. Seguimos utilizando el modulo colector que nos hace la suma de espacios disponibles que es controlado por los switches de la FPGA

NET "CLK" LOC = B8;

NET "VIN[0]" LOC = G18;

NET "VIN[1]" LOC = H18;

NET "VIN[2]" LOC = K18;

NET "VIN[3]" LOC = K17;

NET "VIN[4]" LOC = L14;

NET "VIN[5]" LOC = L13;

NET "VIN[6]" LOC = N17;

NET "VIN[7]" LOC = R17;

NET "RS" LOC = L17;

NET "RW" LOC = K12;

NET "ENA" LOC = L15;

NET "DATA\_LCD[0]" LOC = M16;

NET "DATA\_LCD[1]" LOC = M14;

NET "DATA\_LCD[2]" LOC = L16;

NET "DATA\_LCD[3]" LOC = K13;

Se decidió hacer esta asignación ya que se puede conectar los pines de la LCD 16X2 de 4 pines una forma más exacta y ordenada lo que nos va ayudar a optimizar el tiempo en las siguientes prácticas, este módulo puede utilizar los 8 switches y el resultado se ve reflejado en el módulo colector.

Ya teniendo claro las modificaciones de los anteriores módulos procedemos

A diseñar el protocolo RS232.

Se optó por la decisión de que el modulo central recibiera la información de otros (2 módulos colectores), cada módulo colector recibe a su vez 12 sensores.

El protocolo ya ha sido implementado por la empresa INTESC en una librería, para implementarla solo tuvimos que estudiar que funciones nos ayudaban a completar satisfactoriamente la transmisión y recepción entre las tarjetas de desarrollo.

Esta librería funciona de la siguiente manera:

Descripción: Las características de este módulo son:

\* 1 bit de inicio

\* 8 Bits de transmisión/recepción

\* 1 bit de Paro

\* Sin paridad

La recepción es asíncrona y podría ocurrir en cualquier momento.

Para saber cuándo se ha recibido un BYTE se usa RX\_IN. Este puerto se pone a '1' durante un ciclo de reloj cuando se ha recibido un byte.

Es responsabilidad del diseñador monitorearlo constantemente. El BYTE recibido estará disponible en DOUT. Por ejemplo, si se recibe un 55 hexadecimal, entonces RX, RX\_IN y DOUT se comportan de la siguiente manera

RX

¯¯¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_

RX\_IN

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/¯¯\\_\_\_\_

DOUT

--------------------------DESCONOCIDO------------><55------

La transmisión es controlada por 3 puertos: TX\_IN, TX\_FIN y TX.

Cuando se quiere comenzar a enviar, se debe poner TX\_INI a '1' y se debe cargar la información en DATAIN. Cuando se han terminado de enviar los 10 bits (1 bit de inicio, 8 bits de información y 1 bit de paro) entonces se pone TX\_FIN a '1' y no se permite otra transmisión hasta que TX\_INI sea '0'. EL diseñador debe poner TX\_INI y DATAIN. TX\_FIN se pone a '1' o a '0' automáticamente.

Por ejemplo, si se desea enviar el 55 hexadecimal y preparar para un nuevo envío, entonces la secuencia es la siguiente

TX

¯¯¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_/¯¯¯¯\\_\_\_\_

TX\_INI

\_\_\_\_\_/¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯¯

TX\_FIN\_

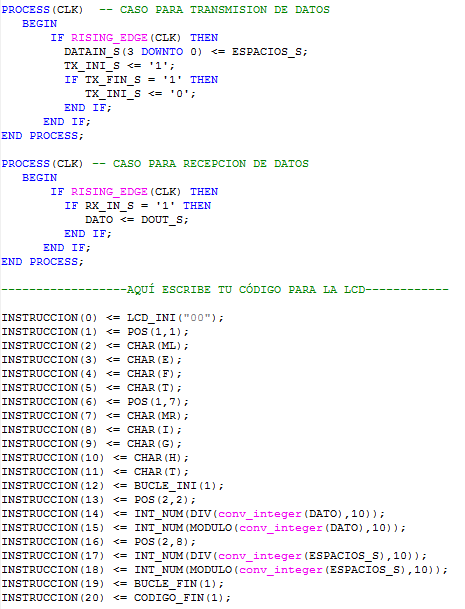
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/¯¯¯\\_\_\_\_

DATAIN

--------------------------55--------------------------------- ><----------

No es necesario que DATAIN tenga la misma información durante toda la transmisión. Un registro interno copia la información cuando se inicia la transmisión dando la posibilidad de cambiar la información de DATA\_IN antes de que termine la transmisión sin problemas de colisión.

El código que se diseñó en el módulo de alto nivel es el siguiente:



El siguiente mensaje a desplegar en la LCD seria:

Left Rigth

#$ #$

# ---> entero que indica las decenas

$ ---> entero que indica las unidades

Para este módulo se encontraron las siguientes dificultades y soluciones:

* El desconocimiento de la implementación del protocolo RS232, con respecto a esta situacion se inició una investigación obteniendo la información buscada.
* La asignación de pines en la FPGA digitalmente se asignó de manera más eficiente.
* Las dificultades a la hora de conectar los puertos a la tarjeta desarrolladora físicamente para esto se realizó una nueva asignación de pines.
* Los limitantes del hardware proporcionado para realizar esta práctica.

Bibliografía

<https://www.youtube.com/watch?v=vR9WBsYrXtA>

<http://www.intesc.mx/librerias/>

<https://reference.digilentinc.com/_media/reference/programmable-logic/nexys-2/nexys2_rm.pdf>