Bitácora Módulo 3: Dispositivo Colector.

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

Abril-2018

Módulo 3: Dispositivo Colector. Implementación del HW necesario para la construcción de un sistema de sensores, colección de información de varios sensores y despliegue de información sobre pantalla LCD.

Se inició una toma de decisiones por parte del equipo de trabajo del laboratorio para definir la cantidad total de sensores que utilizaremos en este módulo como el laboratorio dispone de una total de 10 a 12 sensores utilizaremos esta cantidad para realizar nuestro proyecto e implementar un nuevo módulo el cual definimos como COLECTO.VHD.

Antes de iniciar el desarrollo de este módulo colector se resaltan modificaciones importantes al módulo anterior (Módulo 2: Despliegue sobre pantalla LCD) ya que este módulo colector depende del módulo mencionado.

la siguiente modificación fue la asignación de pines de la FPGA nexys 2 que incluye el módulo 2 y los sensores sin una lógica de salida estos sensores siempre estarán en un NIVEL ALTO esto quiere decir que el sensor mientras no esté interrumpido siempre tendrá el valor lógico de 1 y si se interrumpe el sensor su valor lógico será 0, esta es su siguiente asignación:

NET "CLK" LOC = B8;

NET "RS" LOC = T17;

NET "CORD" LOC = B18;

NET "CORI" LOC = D18;

NET "RW" LOC = R15;

NET "ENA" LOC = R18;

NET "DATA\_LCD[0]" LOC = M13;

NET "DATA\_LCD[1]" LOC = U18;

NET "DATA\_LCD[2]" LOC = T18;

NET "DATA\_LCD[3]" LOC = R16;

NET "DATA\_LCD[4]" LOC = P17;

NET "DATA\_LCD[5]" LOC = L17;

NET "DATA\_LCD[6]" LOC = K12;

NET "DATA\_LCD[7]" LOC = L15;

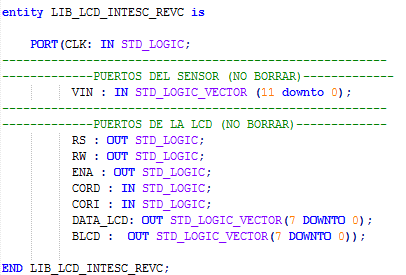
NET "VIN[0]" LOC = K13;

Se decidió por esta asignación ya que se puede conectar los pines de la LCD 16X2 de una forma más exacta y ordenada lo que nos va ayudar a optimizar el tiempo en las siguientes prácticas, este módulo puede utilizar los 12 sensores, pero solo se asignamos un pin de entrada del sensor a la FPGA para realizar la sustentación del módulo 3

Ya teniendo claro las modificaciones de los anteriores módulos procedemos

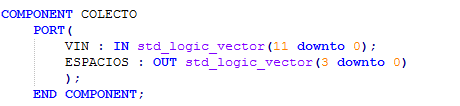
A diseñar el modulo colector:

Lo primero fue implementar las nuevas entradas de los sensores para esto utilizamos puerto VIN de tipo entrada lógica, siendo un vector de 12 posiciones (bits) la cantidad total de sensores que utilizaremos.

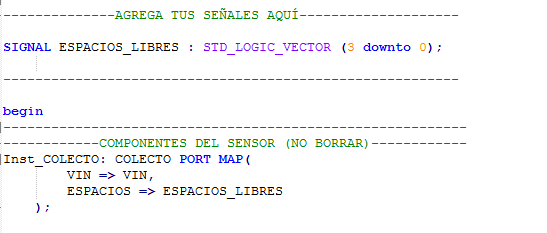


La instanciación del módulo del sensor y del módulo del LCD 16X2 en un solo proyecto.

instanciamos en el módulo LIB\_LCD\_INTESEC\_REV (Modulo Principal o Top Module) el modulo colector (COLECTO.VHD) copiando su componente y su instanciación, este módulo tiene una entrada VIN de tipo lógico y es un vector de 12 posiciones (Bits) y una salida ESPACIOS de tipo lógico también es un vector de 4 posiciones (Bits).

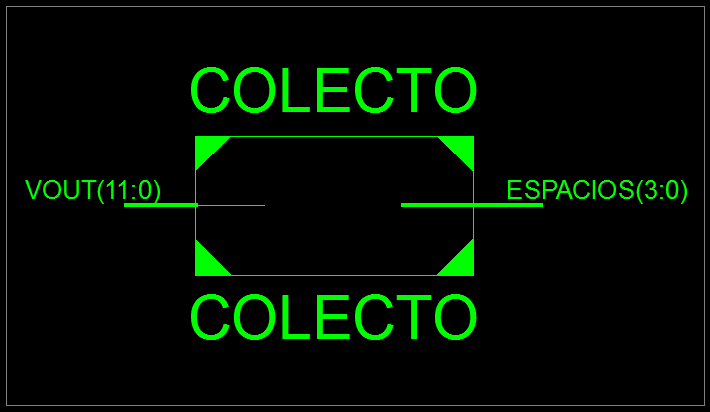


En el módulo LIB\_LCD\_INTESEC\_REV (Modulo Principal o Top Module) se crea una señal de tipo lógica ESPACIOS\_LIBRES siendo un vector de 4 posiciones (Bits) esta señal se le asigna a la salida de modulo COLECTO.VHD con esta señal desplegaremos la cantidad total de espacios libres en la LCD.



La entrada VIN del módulo LIB\_LCD\_INTESEC\_REV (Modulo Principal o Top Module) se le asigna a la entrada del módulo colector que tiene el mismo nombre VIN como las dos entradas son de tipo lógico y son un vector de 12 posiciones aseguramos la asignación de bit a bit.

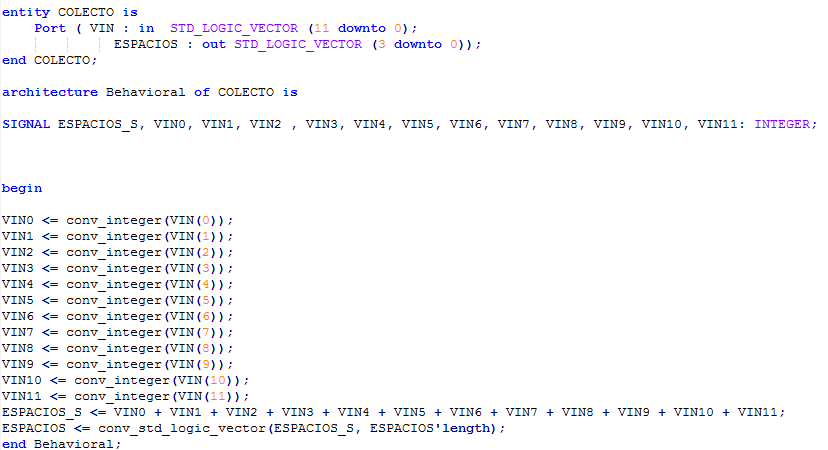
RTL Schematic COLECTO.VHD



Lo que sucede dentro dela caja negra básicamente consiste en que con la entrada VIN que contienen los valores de los sensores dependiendo de su estado generamos unas señales de tipo entero ESPACIOS\_S, VIN0, VIN2, …, VIN11

En cada señal de VIN# asignamos la conversión de las entradas de tipo lógico a entero luego de hacer esto le asignamos a la señal ESPACIOS\_S la suma de las señales VIN# por ultimo le asignamos a la salida ESPACIOS el valor de la señal ESPACIOS\_S de tipo entero, para esto tuvimos que hacer una investigación de como pasar de un entero a un vector de 4 posiciones(bits) de tipo lógico para esto se utiliza el comando:

conv\_std\_logic\_vector (signal, length);



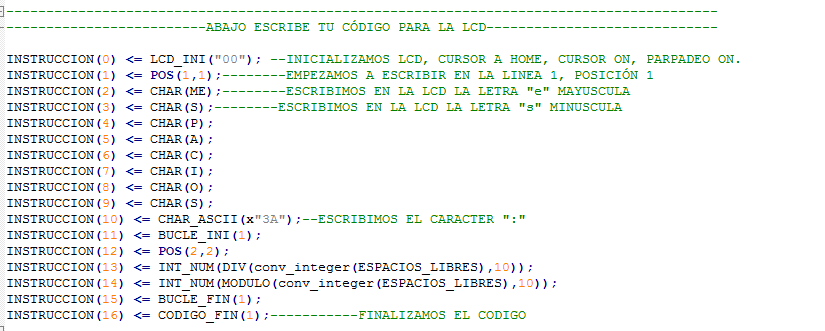
Que nos garantiza pasarle las señales directas de entrada de cada sensor a este módulo COLECTO.VHD que está siempre van a estar en un NIVEL ALTO, por lo tanto, la suma siempre va a estar en 12 (cantidad total de espacios disponibles) hasta que cambien los estados de un sensor (si detecta o no detecta) lo que hará que un sensor no sume con 1 si no con 0 reduciendo así los espacios disponibles.

Ya teniendo nuestro modulo colector pasamos al módulo LIB\_LCD\_INTESEC\_REV (Modulo Principal o Top Module) en este efectuaremos el despliegue de información en la LCD, para que la LCD este cambiando la información de la cantidad de espacios disponibles es necesario meter la INSTRUCCIÓN (12, 13 y 14) dentro de un bucle para eso utilizamos las funciones BUCLE\_INI () Y BUCLE\_FIN () Proporcionadas por la librearía que manejamos INTESEC con esto garantizamos que despliegue de información será continuo si se modifica el estado de los sensores.

Como ESPACIOS\_LIBRES es una señal lógica siendo un vector de 4 posiciones (bits) tenemos que convertir el valor binario que tenga esta señal a un entero para eso hacemos uso del siguiente comando:

conv\_integer (signal);

después lo pasamos a la LCD como un entero desplegando en pantalla la cantidad de espacios libres.



El siguiente mensaje a desplegar en la LCD seria:

Espacios:

# $

# ---> entero que indica las decenas

$ ---> entero que indica las unidades

Para hacer las divisiones y la operación modulo usamos una librería que nos permite hacer estas operaciones, esta Liberia también es diseña por la compañía

Intescmx.

Se instancia de la siguiente forma:

* USE WORK.OP\_DIVISION.ALL

Se utiliza como se ve en la siguiente imagen:



Recibe 2 enteros separados por coma (“,”) dependiendo si es división DIV (A, B)

Y si es módulo MOUDLO (A, B) siendo A y B 2 enteros

Como la biblioteca de la LCD 16X2 solo nos deja mostrar enteros desde el 0 – 9

Se utilizan estas operaciones para desplegar la cantidad de espacios separando decenas y unidades por eso se utilizó esta biblioteca ya que algunas FPGA´s

No permiten la división (“/”) con esta biblioteca resolvemos este inconveniente

Para este módulo se encontraron las siguientes dificultades y soluciones:

* El desconocimiento de conversión de diferentes tipos de variables en VHDL

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.nandland.com/vhdl/tips/tip-convert-numeric-std-logic-vector-to-integer.html>

<https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>

<https://www.youtube.com/watch?v=A7g4IkbV8PM&t=230s>

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

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