Bitácora Módulo 2: Despliegue sobre pantalla LCD.

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

Febrero-2018

Módulo 2: Despliegue sobre pantalla LCD. Implementar un módulo que despliegue la información de cualquier tipo sobre pantalla LCD.

Se inició una investigación por parte del equipo de trabajo del laboratorio de electrónica digital para comprender el funcionamiento de un LCD16x2 y utilizarlo para realizar el Módulo 2, como hacemos uso del protocolo HD44780U (LCD-II) para el despliegue de información en la LCD se usó una librería desarrollada por INTESC que tomaba como base este protocolo.

INTESC es una empresa mexicana dedicada al diseño de sistemas embebidos basados en FPGAs y SoCs, asimismo se dedica al desarrollo y producción de nueva tecnología en áreas de electrónica y sistemas embebidos.

Lo siguiente es explicar las funciones de que conforman la librería de nuestro código en VHDL para la LCD16X2.

por ultimo demostrar el funcionamiento del código en VHDL para la pantalla LCD16X2.

La LCD 16x2 es un componente que se utiliza para la representación de caracteres en muchos dispositivos electrónicos. El principal componente de este diseño será formado por un procesador, un paquete de comandos donde tenemos nuestras funciones de la librería y caracteres especiales, los cuales harán el trabajo de facilitarnos el uso rápido de la LCD16X2.

* Procesador: Este convierte el vector que recibe del paquete comandos en lenguaje que la LCD pueda entender
* Caracteres especiales: Es donde podremos dibujar caracteres personalizados

Las funciones comandos que se utilizan son los siguientes:

* LCD\_INI (): Inicializa la LCD.

NOTA: Dentro de los paréntesis poner un vector de 2 bits para encender o apagar el cursor y activar o desactivar el parpadeo.

"1x" -- Cursor ON

"0x" -- Cursor OFF

"x1" -- Parpadeo ON

"x0" -- Parpadeo OFF

Por ejemplo: LCD\_INI ("10") -- Inicializar LCD con cursor encendido y sin parpadeo

* CHAR (): Manda una letra mayúscula o minúscula

IMPORTANTE:

1) Debido a que VHDL no es sensible a mayúsculas y minúsculas, si se quiere escribir una letra mayúscula se debe escribir una "M" antes de la letra.

2) Si se quiere escribir la letra "S" mayúscula, se declara "MS"

Por ejemplo:

CHAR(A) -- Escribe en la LCD la letra "a"

CHAR (MA) -- Escribe en la LCD la letra "A"

CHAR(S) -- Escribe en la LCD la letra "s"

CHAR (MS) -- Escribe en la LCD la letra "S"

* POS (): Escribir en la posición que se indique.

NOTA: Dentro de los paréntesis se debe poner la posición de la LCD a la que se quiere ir, empezando por el renglón seguido de la posición vertical, ambos números separados por una coma.

Por ejemplo: POS (1,2) -- Manda cursor al renglón 1, posición 2

POS (2,4) -- Manda cursor al renglón 2, posición 4

* CHAR\_ASCII (): Escribe un carácter a partir de su código en ASCII

NOTA: Dentro de los paréntesis escribir x"(número hex.)"

Por ejemplo: CHAR\_ASCII(x"40") -- Escribe en la LCD el carácter "@"

* CODIGO\_FIN (): Finaliza el código.

NOTA: Dentro de los paréntesis poner cualquier número: 1, 2, 3,4..., 8,9.

* BUCLE\_INI (): Indica el inicio de un bucle.

NOTA: Dentro de los paréntesis poner cualquier número: 1, 2, 3,4..., 8,9.

* BUCLE\_FIN (): Indica el final del bucle.

NOTA: Dentro de los paréntesis poner cualquier número: 1, 2, 3,4..., 8,9.

* INT\_NUM (): Escribe en la LCD un número entero.

NOTA: Dentro de los paréntesis poner sólo un número que vaya del 0 al 9,

si se quiere escribir otro número entero se tiene que volver a llamar la función

* CREAR\_CHAR (): Función que crea el carácter diseñado previamente en "CARACTERES\_ESPECIALES.vhd"

NOTA: Dentro de los paréntesis poner el nombre del carácter dibujado (CHAR1, CHAR2, CHAR3, …, CHAR8)

* CHAR\_CREADO (): Escribe en la LCD el carácter creado por medio de la función "CREAR\_CHAR ()"

NOTA: Dentro de los paréntesis poner el nombre del carácter creado.

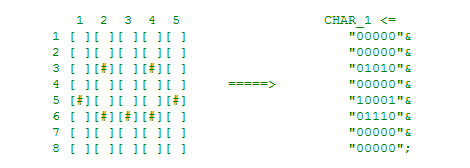
Por ejemplo:

Dentro de "CARACTERES\_ESPECIALES.vhd" se dibujan los caracteres personalizados utilizando los vectores

"CHAR\_1", "CHAR\_2","CHAR\_3”, …,"CHAR\_7","CHAR\_8"

1 = [#] - Se activa el pixel de la matriz.

0 = [ ] - Se desactiva el pixel de la matriz.

CREAR\_CHAR(CHAR1) -- Crea el carácter personalizado (CHAR1)

CHAR\_CREADO(CHAR1) -- Muestra en la LCD el carácter creado (CHAR1)

* LIMPIAR\_PANTALLA (): Manda a limpiar la LCD.

NOTA: Ésta función se activa poniendo dentro de los paréntesis, un '1' y se desactiva con un '0'.

Por ejemplo:

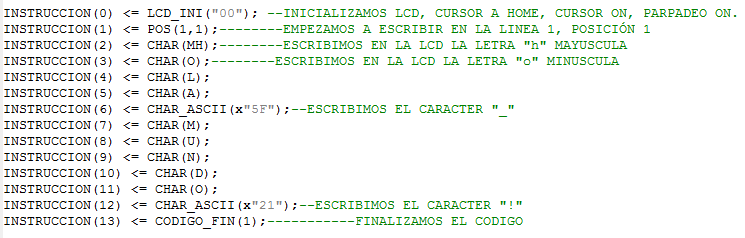
* + - LIMPIAR\_PANTALLA ('1') -- Limpiar pantalla está activado
    - LIMPIAR\_PANTALLA ('0') -- Limpiar pantalla está desactivado.

Las entradas "CORD" y "CORI" se hacen corrimientos a la derecha y a la Izquierda respectivamente.

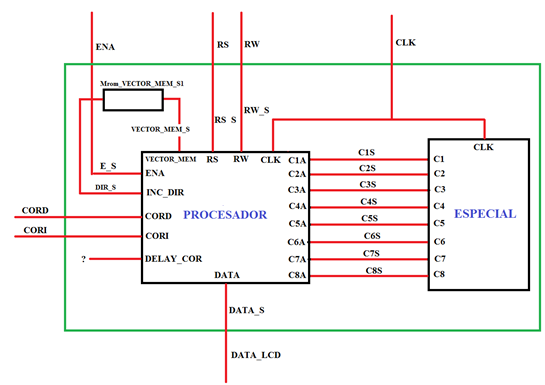
NOTA: La velocidad del corrimiento se puede cambiar modificando la variable "DELAY\_COR".

IMPORTANTE: Cada función se acompaña con

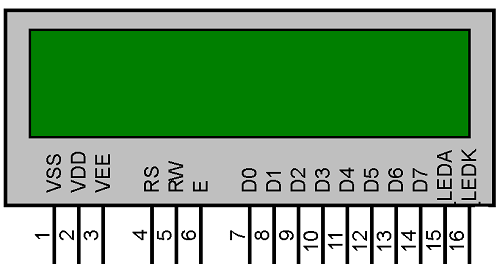
"INSTRUCCIÓN (NUM) <= FUNCIÓN ()”.



Esquemático del Funcionamiento de los módulos Procesador y Caracteres especiales



Pines de la LCD 16X2



La siguiente parte del módulo fue asignar los puertos de entradas y salidas digitalmente teniendo en cuenta que para este segundo módulo seguiremos haciendo uso de la FPGA Nexys 2.

Siguiendo la guía las referencias de esta FPGA son las siguientes:

Family: Spartan 3E, Device: XC3S500C, Package: FG320.

Con la herramienta Xilinx PlanAhead 14.7 se le asigno los pines de entrada y salidas digitalmente a nuestro diseño haciendo uso de la biblioteca proporcionada por INTESC.

NET "CLK" LOC = B8;

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

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

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

NET "DATA\_LCD[4]" LOC = M15;

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

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

NET "DATA\_LCD[1]" LOC = M13;

NET "DATA\_LCD[0]" LOC = R18;

NET "ENA" LOC = R15;

NET "RS" LOC = T17;

NET "RW" LOC = P17;

NET "CORI" LOC = D18;

NET "CORD" LOC = B18;

Después de asignar los pines exitosamente se procedió a generar un archivo programable y con la Tarjeta desarrolladora a realizar las pruebas correspondientes conectar cada uno de los puertos asignados físicamente y por ultimo impactar la FPGA para evaluar que la información si se desplegué por la LCD 16X2 lo cual garantizaba que el diseño se había realizado con éxito.

Para este módulo se encontraron las siguientes dificultades:

* El desconocimiento del protocolo HD44780U (LCD-II)
* El desconocimiento de la biblioteca de INTESC la cual maneja los 8 pines de la LCD 16X2
* La asignación de pines en la FPGA digitalmente
* Las dificultades a la hora de conectar los puertos a la tarjeta desarrolladora físicamente

Bibliografía

<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>

<http://itatienredes.blogspot.com.co/2013/10/que-es-y-como-se-estructura-una-bitacora_8.html>