

eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Visualización de caracteres por medio de un LCD LT24

Diseño Avanzado de Sistemas Digitales

2 de marzo de 2023

Unai Fernandez

Índice

1. Introducción	2
1.1. Funcionamiento del sistema	2
2. Diseño del sistema	3
2.1. Diseño de los submódulos	3
2.1.1. LCD_ctrl	3
2.1.2. LCD_uart	6
2.1.3. LCD_drawing	10
2.1.4. Main	13
3. Pruebas y Resultados obtenidos	16

1. Introducción

En este proyecto se ha diseñado un sistema, que mostrará en una pantalla LCD LT24 los caracteres que se le envíen a este mismo por el puerto serie.

El usuario necesitará para probar el proyecto, una placa de desarrollo *DE10 Standard* y un PC con el que se pueda comunicar.

Para la compilación del proyecto y la programación de la placa es necesario el programa **QUARTUS II**. Para la comunicación serial sería recomendable tener un programa como **PuTTY** (para Windows) o **Minicom** (para Linux).

1.1. Funcionamiento del sistema

Tras haber programado la placa adecuadamente, el usuario iniciará una comunicación con la LT24 por medio del puerto serie, con la aplicación **PuTTY**. Después de esto, la tecla pulsada en el teclado del PC se visualizará en la pantalla del LCD LT24.

A parte de ver las letras en pantalla y poder escribir palabras, el usuario también podrá cambiar de color las letras y borrar la pantalla entera aplicando en el fondo el color deseado pulsando el **KEY1**. La selección del color, tanto para el fondo, como para las letras se realiza por medio de los tres primeros switches de la placa.

2. Diseño del sistema

Para facilitar la realización de las diferentes tareas, el sistema está dividido en varios módulos:

- Módulo controlador del LCD (LCD_ctrl): Este módulo se encargará de mandar las señales de control adecuadas al LT24 para poder colocar el cursor y dibujar píxeles.
- Módulo de comunicación (LCD_uart): Este módulo se encarga de comunicarse con el host (PC) y obtener el código del carácter recibido.
- Módulo de dibujo (LCD_drawing): Este módulo se encarga de borrar la pantalla del LCD poniendo el fondo de un mismo color, y de dibujar cuadrados de 5x5 píxeles, que serán los que formarán las letras.
- Módulo general (Main): El main se encarga de gestionar los módulos anteriores, las entradas (Botones y switches) y acceder a la memoria ROM que contiene los datos necesarios para formar las letras

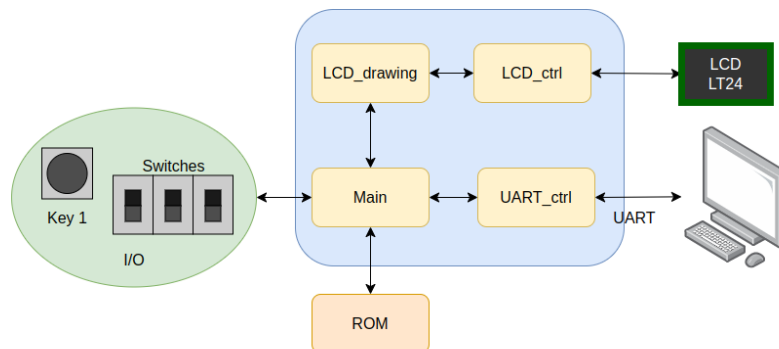


Figura 1: Esquema general del proyecto

2.1. Diseño de los submódulos

En este apartado se explicarán cada uno de los submódulos con más detalle, mostrando la unidad de proceso y la de control que hacen que cada uno de ellos funcione de forma adecuada.

2.1.1. LCD_ctrl

El módulo controlador del LCD cumple con dos simples tareas. Una, colocar el cursor donde se le indique y dibujar un número n de píxeles en una posición en concreto. En la imagen [2] a continuación se puede observar el esquema general de las entradas y salidas del módulo.

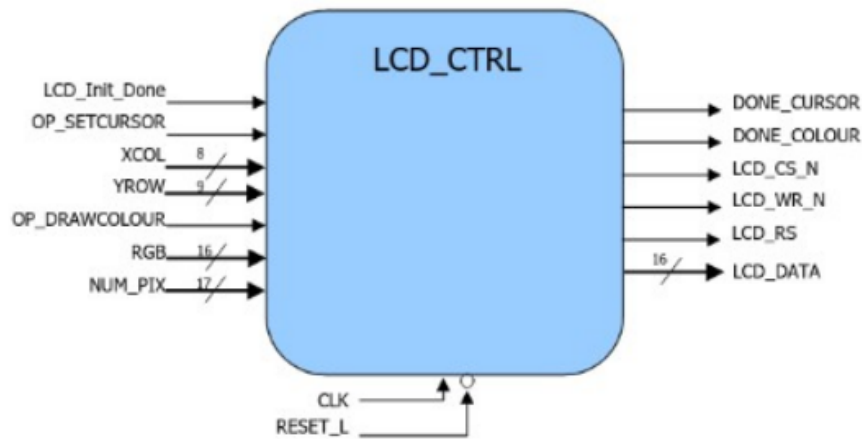


Figura 2: Esquema general del modulo LCD_ctrl

En la imagen [3] se puede ver la unidad de proceso del módulo LCD_ctrl, compuesto por tres registros, para guardar las coordenadas y el color, dos contadores (Uno para los pixeles y otro para ir eligiendo el dato o comando correspondiente), un multiplexor donde cada entrada son los datos y comandos que se le mandan al LCD, y finalmente un biestable JK.

En la figura [4] podemos ver la unidad de control del módulo de control del LCD. El algoritmo ASM consta de 12 estados diferentes.

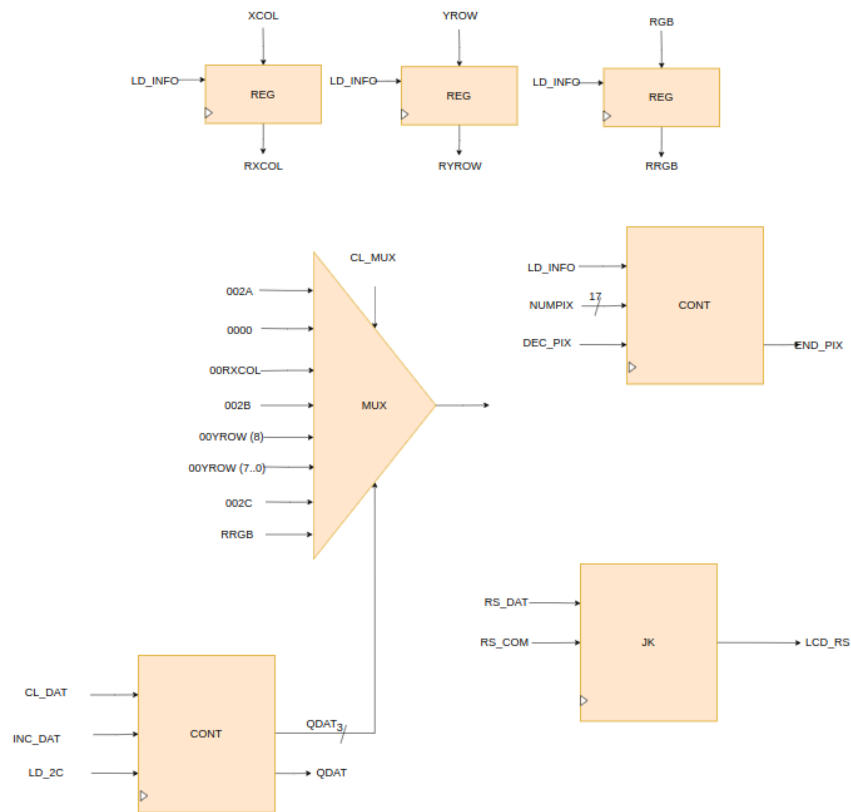


Figura 3: Unidad de proceso del módulo LCD_ctrl

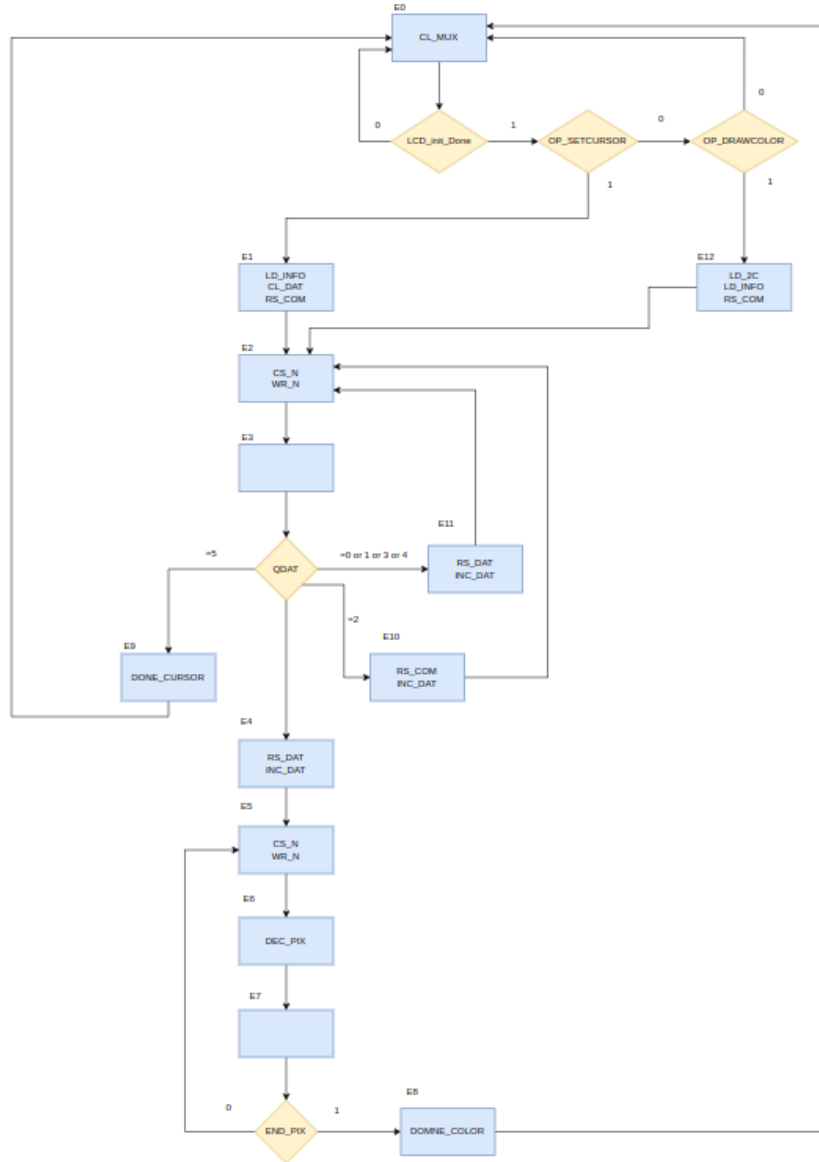


Figura 4: Unidad de proceso del módulo LCD_ctrl

2.1.2. LCD_uart

El módulo **LCD_uart**, como su nombre indica, se dedica a gestionar las señales que llegan por el puerto serie. El módulo leerá la señal, y tomará muestras periódicas que irá guardando en un registro de desplazamiento.

La señal que llega al módulo es similar a la que se puede observar en la imagen [5].

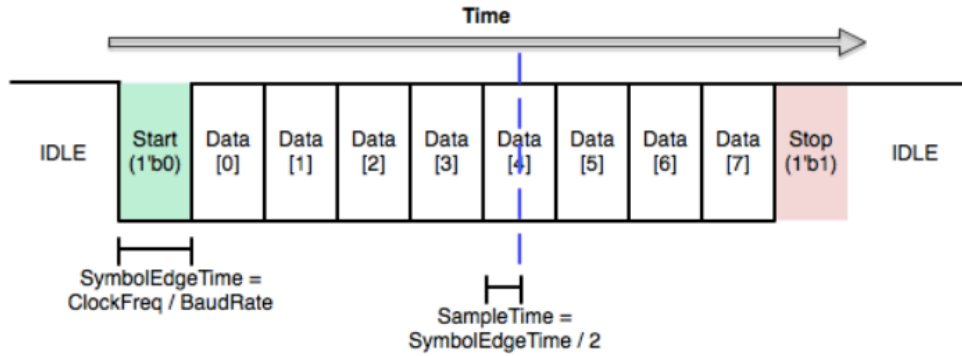


Figura 5: UART

El procedimiento para obtener los datos será el siguiente. Primero, el algoritmo esperará a que se produzca la señal de start. Esto sucede cuando la señal pasa de estar continuamente en alto a bajo. Después, se irán tomando muestras periódicamente para ir guardando los datos en el registro de desplazamiento.

Para obtener el periodo de muestreo, necesitaremos saber el tiempo que dura un bit de datos en la señal. Si el *Baud Rate* de 9600 (bits/seg) la duración de un bit será de 104167 ns.

Finalmente, lo que queda por resolver es cuantos ciclos necesitará nuestro sistema con una señal de `clk` de 50 MHz.

$$T = \frac{1}{50 * 10^6 Hz} = 0,00000002s \rightarrow 20ns \quad (1)$$

$$T_{muestreo} = \frac{104167ns}{20ns} = 5208ciclos \quad (2)$$

En la figura [6] se puede ver el esquema general del módulo. Como se puede ver es bastante simple, solo tiene una entrada y dos salidas, el dato y una señal que se activa al obtener un dato.

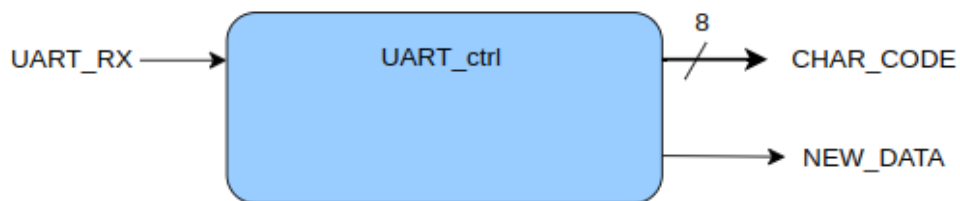


Figura 6: Esquema general del modulo LCD_uart

La imagen [7] muestra la unidad de proceso del módulo. Esta unidad de proceso la componen un registro de desplazamiento para ir almacenando los datos, dos contadores, uno para contar el tiempo y el otro para contar el número de bits, y un multiplexor.

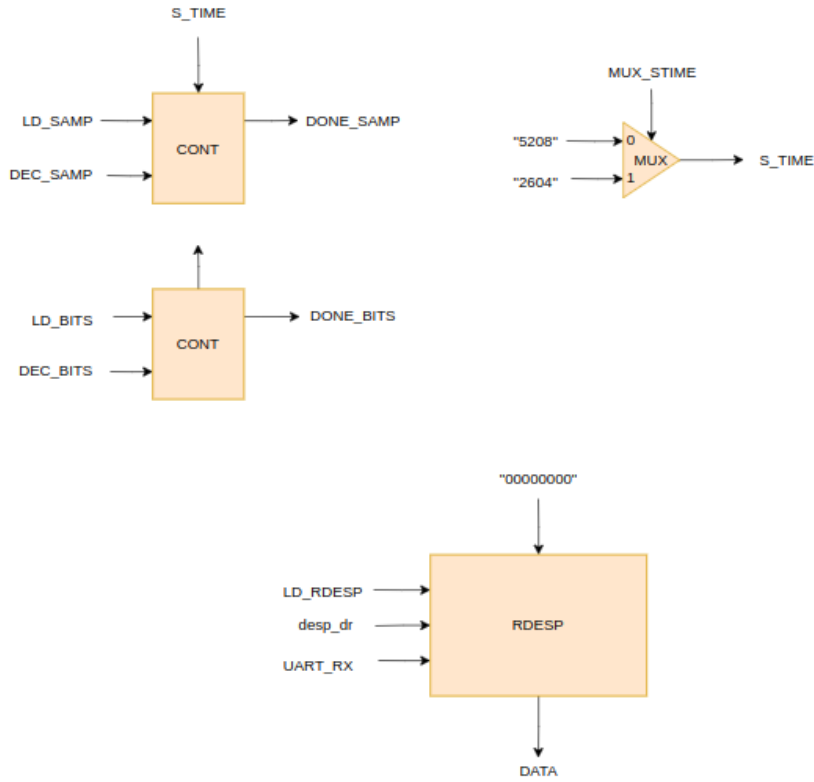


Figura 7: Unidad de proceso del módulo LCD_uart

En la figura [8] se puede observar la unidad de control del módulo LCD_uart, compuesta por 7 estados.

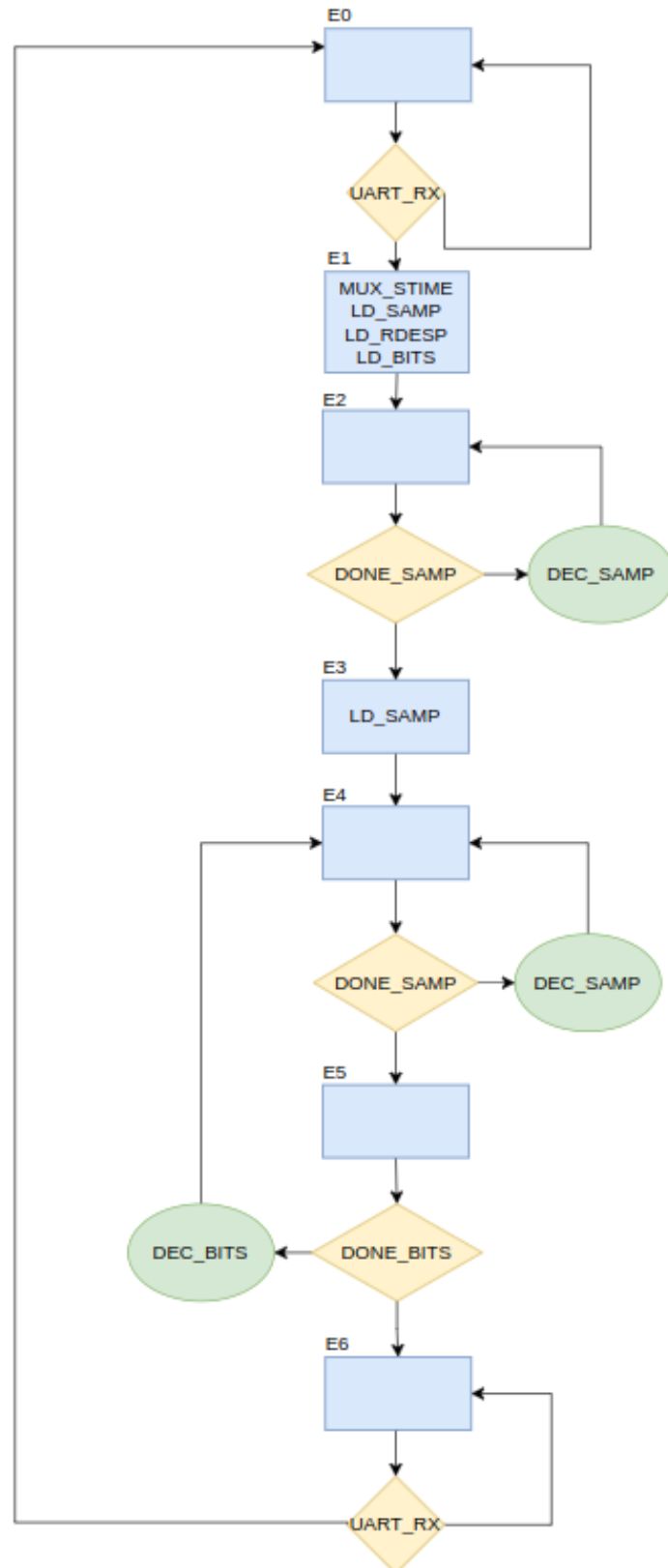


Figura 8: Unidad de control del módulo LCD_uart

2.1.3. LCD_drawing

El módulo LCD_drawing es el que se va a encargar de dibujar los cuadrados, que compondrán las letras, en la posición que se le indique.

El procedimiento para dibujar los cuadrados, es el siguiente. Primero, se ubica el cursor en las coordenadas deseadas, y después de va formando un cuadrado pintando n filas de n píxeles. El módulo LCD_drawing se comunica con el módulo LCD_ctrl, para pasarle las coordenadas y el número de pixeles a pintar. Por otra parte el LCD_ctrl también le avisa al LCD_drawing cuando ha terminado de ejecutar la operación mediante las señales DONE_DRAW y DONE_CURSOR.

La imagen [9] muestra el esquema general del módulo LCD_drawing, en el que aparecen todas las entradas y salidas.

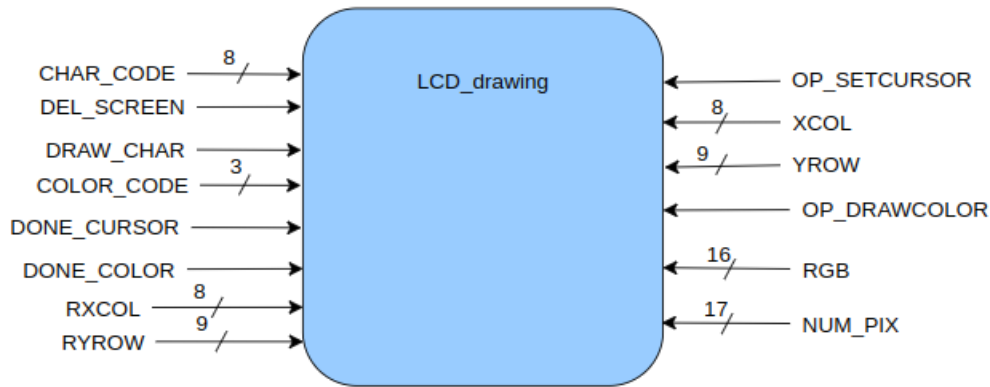


Figura 9: Esquema general del modulo LCD_drawing

La figura [10] muestra la unidad de proceso del módulo LCD_drawing. En ella se pueden ver un biestable JK, un registro, un contador para las filas, un comparador y tres multiplexores.

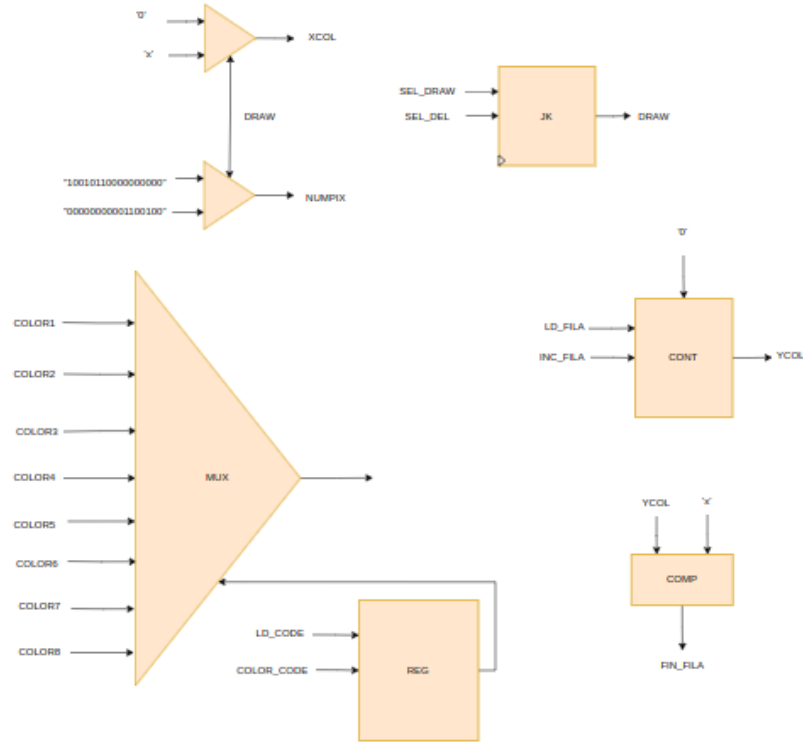


Figura 10: Unidad de proceso del módulo LCD_drawing

En la imagen [11] se ve la unidad de control del módulo LCD_drawing, compuesta por 6 estados.

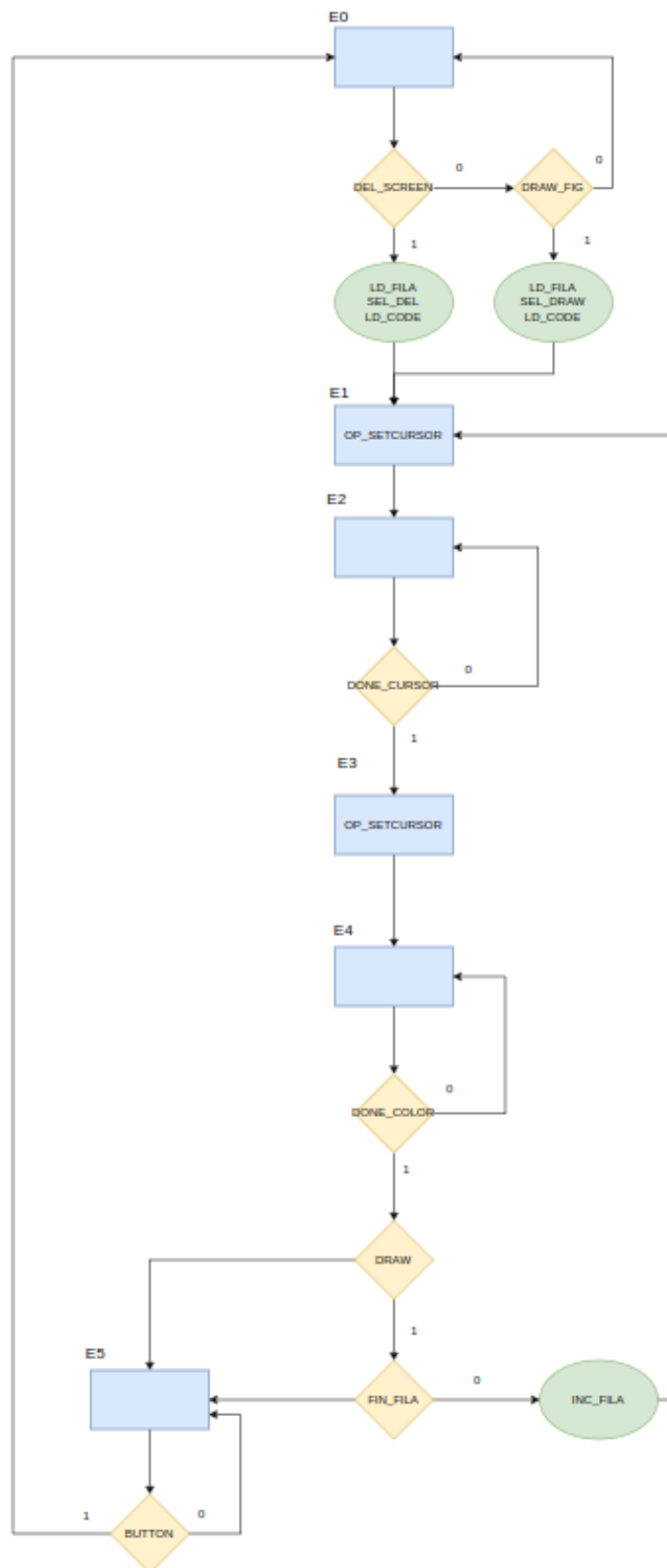


Figura 11: Unidad de control del módulo LCD_drawing

2.1.4. Main

El main o gestor principal se va a encargar de recibir el código proveniente del módulo LCD_uart y dependiendo de su valor obtener la posición de los pixeles que hay que pintar para formar las letras.

Los cuadrados que dibuja el módulo LCD_drawing, serán los pixeles que forman las letras. Cada una de las letras está formada por 4 columnas y 5 filas.

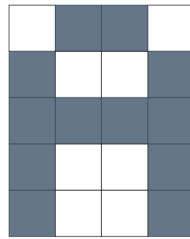


Figura 12: Ejemplo, letra A

En la memoria ROM con la que el main interactúa, almacenan los cuadrados que hay que pintar en cada una de las filas. Por cada una de las letras hay 5 vectores de 4 elementos.

Para obtener la dirección donde se almacenan los datos correspondientes a cada una de las teclas se usa el código que ha recibido el módulo LCD_uart y se hace la siguiente operación.

$$(char_code - 97) * 5 \quad (3)$$

Los códigos de las letras minúsculas empiezan por el número 97 por eso se le restan 97, y se multiplica por 5 porque cada letra tiene 5 datos en la ROM.

En la figura [13] se ve la unidad de proceso del módulo principal. Este módulo es el más complejo de todos, y como se ve en la imagen la componen múltiples sumadores y registros que se encargan de calcular y guardar las coordenadas actuales y las del siguiente carácter, dos contadores para leer los datos de la memoria ROM y una memoria ROM.

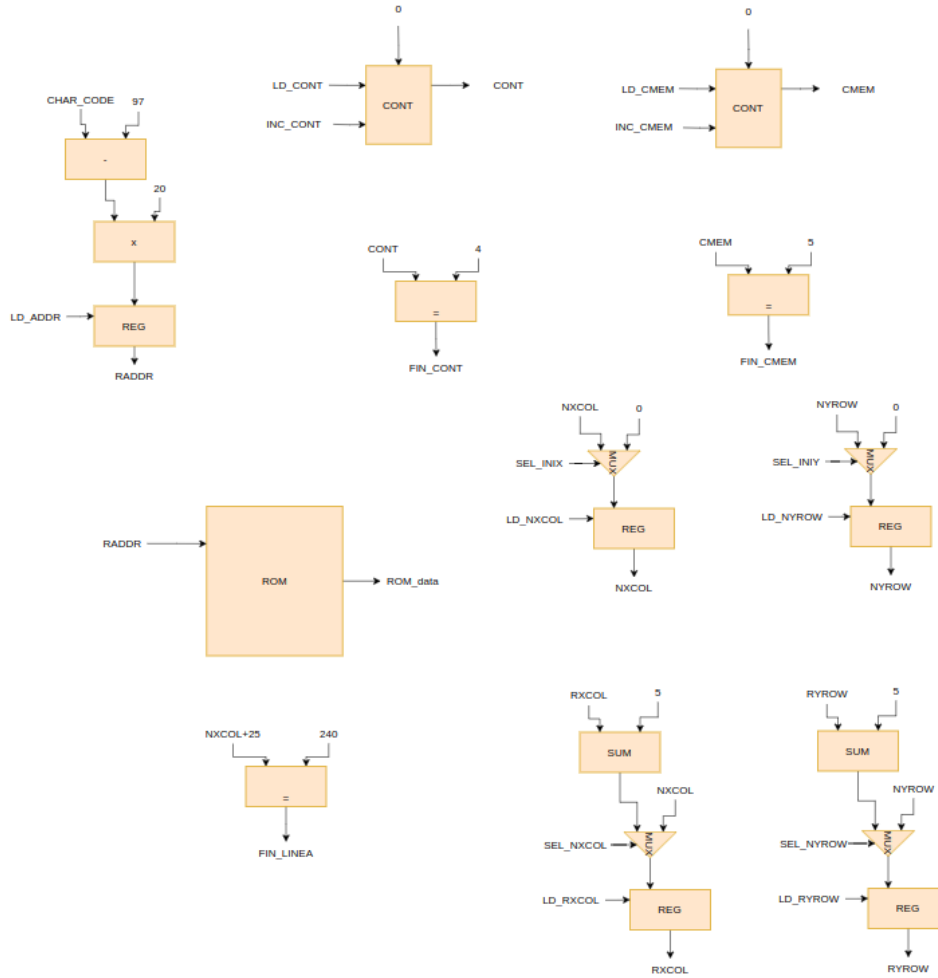


Figura 13: Unidad de proceso del main

La imagen [14] muestra la unidad de control del módulo principal, compuesta por 12 estados diferentes.

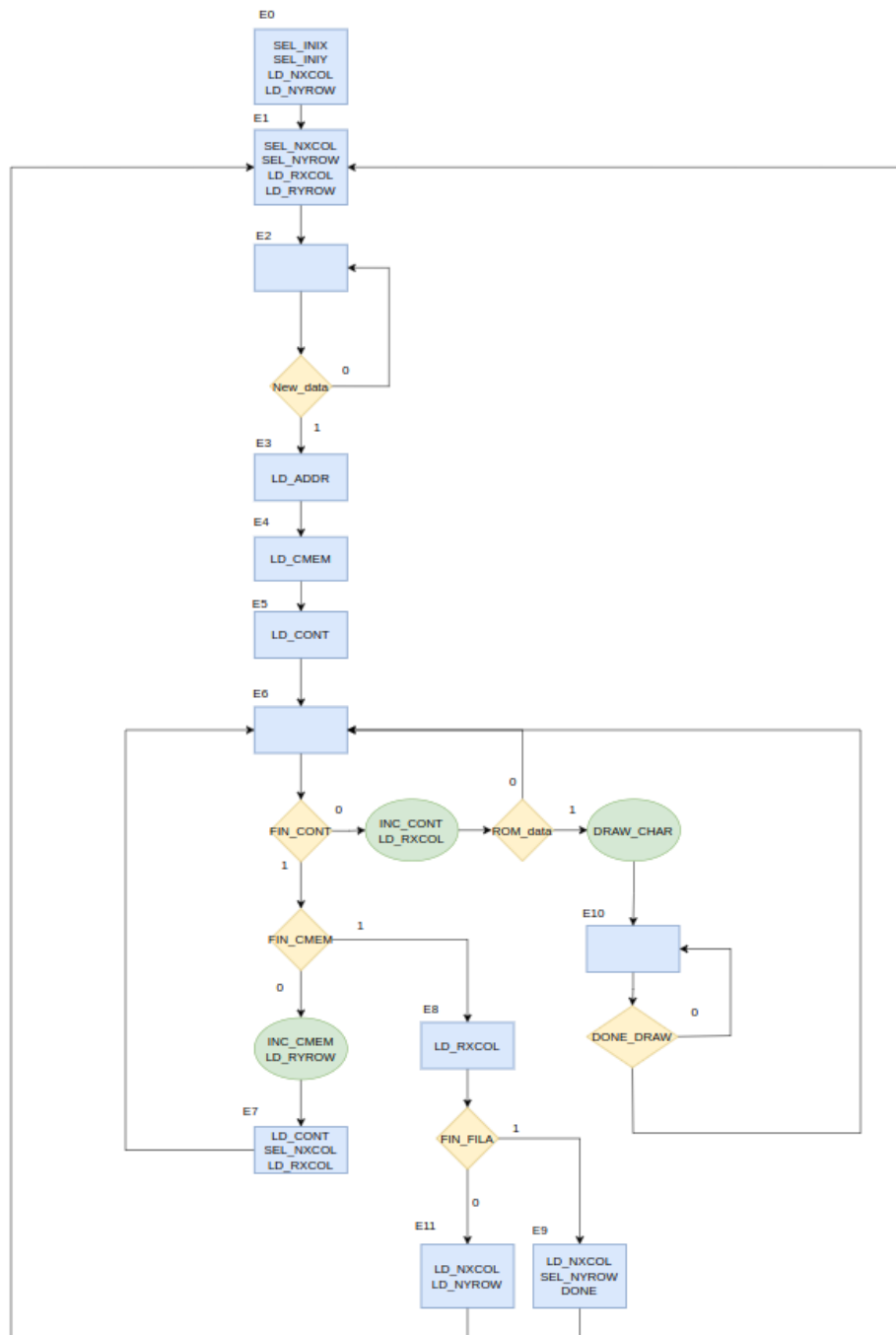


Figura 14: Unidad de control del main

3. Pruebas y Resultados obtenidos

En un sistema tan complejo de diseñar como es este, es conveniente dividirlo en diferentes módulos. Como ya se ha comentado en el apartado de *Diseño del sistema* los módulos que componen el sistema son LCD_ctrl, LCD_drawing y LCD_uart.

Cada uno de estos módulos se ha diseñado y testeado de manera individual, para ello se ha usado el programa **ModelSim**. Este programa permite ejecutar periodos de tiempo que el usuario elija, y ver la evolución de las señales seleccionadas en un cronograma.

Las pruebas correspondientes a cada módulo se pueden encontrar en la carpeta **Pruebas**.

En el caso del main o gestor principal, se ha desarrollado usando **Quartus II** y se ha comprobado el correcto funcionamiento del mismo, usando la utilidad **SignalTap** que incorpora **Quartus**.