

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Facultad de Ingeniería en Electricidad y Computación



LABORATORIO DE SISTEMAS DIGITALES II

Paralelo del laboratorio: 101

Avance Final de Proyecto

**Tema de Proyecto: Ordenamiento de Datos por Método de
Burbuja de Fertilizantes para el Campo Agricultor**

Presentado por:

Ricardo Andrés Romero Baldeon

Gabriel Alfredo Sancho Vera

Profesor:

Dr. Víctor Manuel Asanza Armijos

Ayudante:

Carlos Villamar

GUAYAQUIL - ECUADOR

II TÉRMINO 2022

Contenido

CAPÍTULO 1	3
1. Introducción	3
1.1. Introducción	3
CAPÍTULO 2	5
2. Objetivos	5
2.1. Objetivo General.....	5
2.2. Objetivos específicos	5
CAPÍTULO 3	6
3. Descripción	6
3.1. Descripción del problema	6
3.2. Descripción de la propuesta escogida.....	6
3.3. Diagrama de Bloques.....	7
3.4. Diagrama ASM	8
3.5. Asignación de Pines	9
3.6. Diagrama Esquemático en Altium	11
CAPÍTULO 4	14
4. Aplicaciones a futuro	14
CAPÍTULO 5	14
5. Conclusiones y recomendaciones	14
5.1. Conclusiones.....	14
5.2. Recomendaciones.....	15

CAPÍTULO 1

1. Introducción

1.1. Introducción

El ordenamiento de datos en simplicidad es colocar información de una manera específica basándose en un criterio de ordenamiento con el fin de facilitar la búsqueda en los registros, el ordenamiento de datos no solamente ayuda a mostrar información de una manera específica debido a esto permite ahorrar recursos como lo es la cantidad de tiempo requerida para ordenarlos manualmente, permitiendo analizar grandes cantidades de datos para su uso.

Como parte de la solución para la selección de fertilizantes se implementaron circuitos sincrónicos que trabajaron con lógica positiva, donde se leerán los datos recolectados de varios fertilizantes utilizados en los cultivos más comunes que existen en el Ecuador. La base de datos constara de 317 datos de los cuales solo se tomara una muestra de los primeros 50 datos para ordenarlos de manera descendente para mostrar la eficiencia de cada fertilizante al omento de utilizarlo.

```
procedimiento DeLaBurbuja ( $a_0, a_1, a_2, \dots, a_{(n-1)}$ )  
  para  $i \leftarrow 1$  hasta  $n$  hacer  
    para  $j \leftarrow 0$  hasta  $n - 2$  hacer  
      si  $a_{(j)} > a_{(j+1)}$  entonces  
         $aux \leftarrow a_{(j)}$   
         $a_{(j)} \leftarrow a_{(j+1)}$   
         $a_{(j+1)} \leftarrow aux$   
      fin si  
    fin para  
  fin para  
fin procedimiento
```

Ilustración 1. Pseudocódigo del Tipo de Ordenamiento a Utilizar.

1.2. Antecedentes

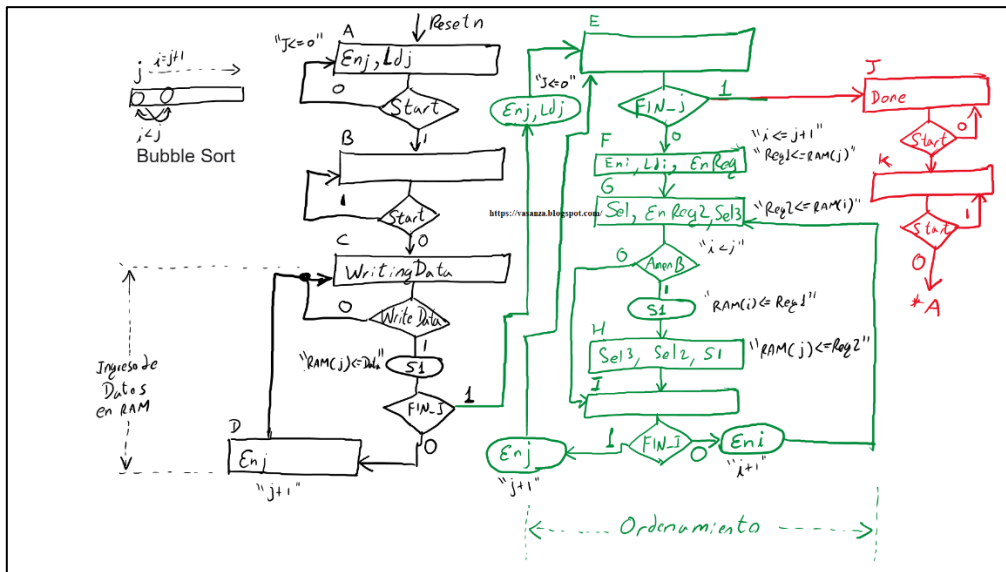


Ilustración 2. Ejemplo de ordenamiento de datos en una memoria RAM

Para la propuesta de proyecto se utilizaron las bases de un problema de ordenamiento de RAM que se encontraba en el repositorio de GITHUB, el cual se utilizó como guía para desarrollar el proyecto.

CAPÍTULO 2

2. Objetivos

2.1. Objetivo General

Ordenar datos numéricos que se encuentran almacenados en una memoria RAM utilizando el método burbuja implementando un circuito digital en un software de simulación para aplicarlo en la vida real utilizando una tarjeta programable y el circuito esquemático implementado en una protoboard.

2.2. Objetivos específicos

- Diseñar un sistema digital que permita ordenar los elementos de la RAM de mayor a menor en el menor tiempo posible.
- Mostrar los datos ordenados de la base de datos por medio de la implementación de 2 displays de 7 segmentos que representaran la eficiencia.
- Optimizar lo más posible dentro de la implementación del programa en general.

CAPÍTULO 3

3. Descripción

3.1. Descripción del problema

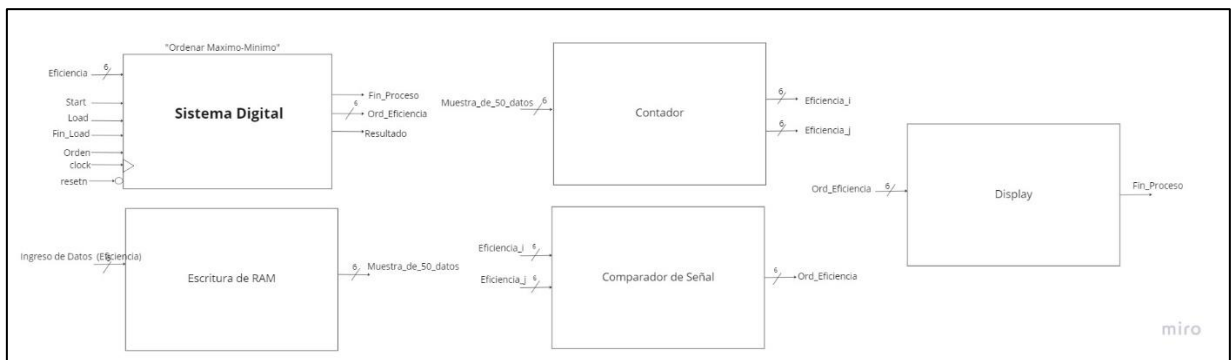
Para el proyecto propuesto se necesita realizar un ordenamiento de datos que está cargado en una memoria RAM donde el ordenamiento se ejecutara de mayor a menor dependiendo de los datos que se encuentren cargados en la RAM, cuyos datos están en un rango de 99 hasta 65 que representado en cantidad de bits seria de 6.

Se trabajó con una base de datos que contenía distintas características de los fertilizantes, en la cual solo se escogió la eficiencia. El ordenamiento de mayor a menor se realizará en base a esta variable.

3.2. Descripción de la propuesta escogida

Se procedió a utilizar la RAM que se encuentra en el GITHUB, pero se tuvo problemas en el momento de la ejecución ya que no lograba ordenar para solucionar esto se utilizó la RAM proporcionada por en el IP CATALOG, pero existía un retraso en el ciclo del reloj, logramos volver a la RAM del GITHUB y para solucionar el problema del ordenamiento agregamos una señal de CLOCK de 50MHz para que coincida con las señales y poder realizar el proceso de ordenamiento.

3.3. Diagrama de Bloques



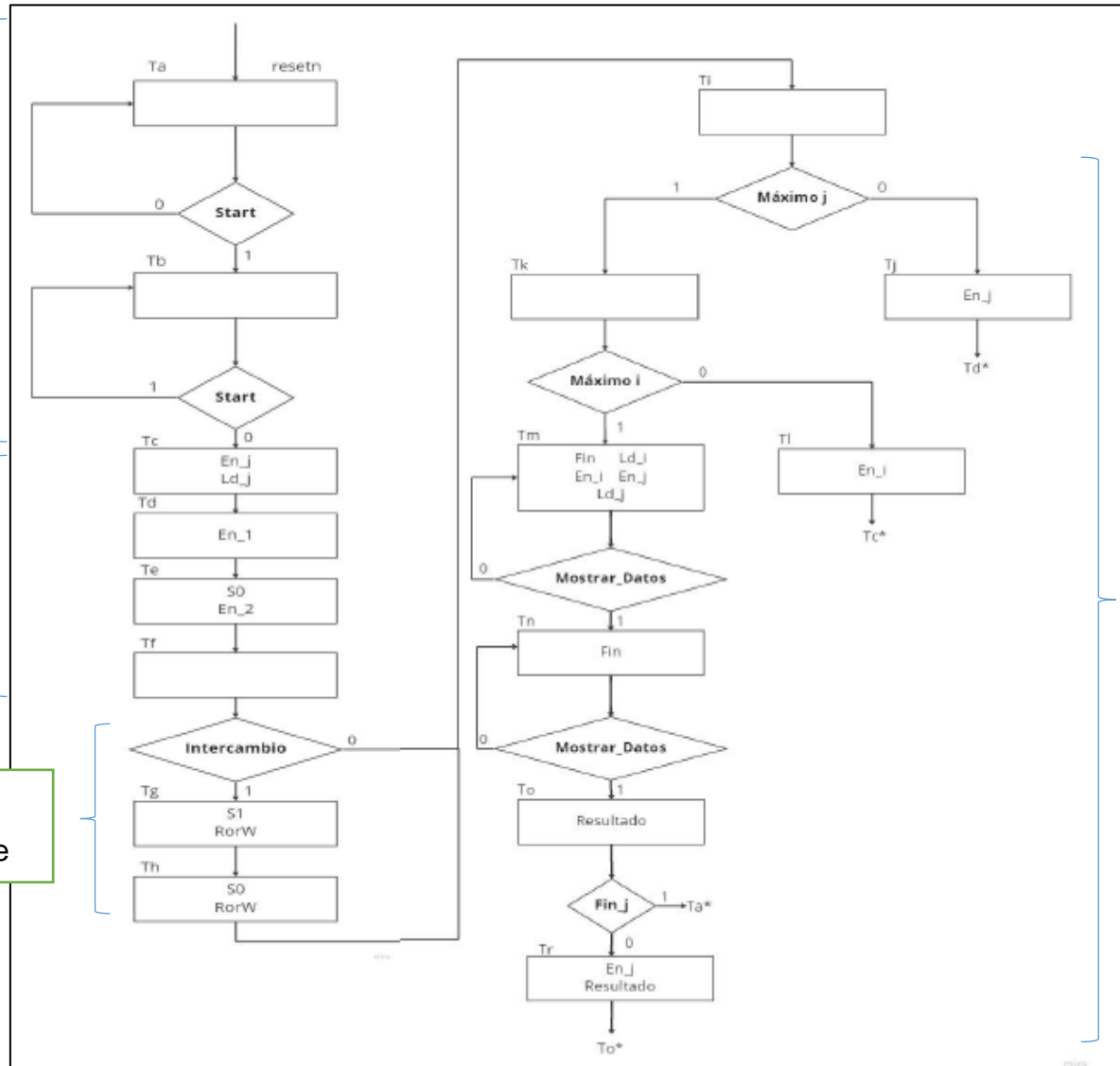
Contador: Este bloque realizará el recorrido de los datos registrados para empezar el ordenamiento.

Comparador de Señal: Compara el dato de eficiencia en la posición actual con el dato siguiente, su salida determinará cuando se produce el ordenamiento.

Memoria RAM: Se utilizará una memoria RAM, donde contendrá los datos de eficiencia de los fertilizantes para posteriormente con el contador realizar el recorrido para ordenar los datos.

Display: Representa los displays donde se mostrarán los datos ordenados.

3.4. Diagrama ASM



1era Parte:

Se presiona el botón Start para dar inicio a la secuencia en caso de presionarlo otra vez existe un anti rebote.

2da Parte:

Proceso de guardar en los registros de sostenimiento el dato de la posición actual y la posición siguiente.

3era Parte:

Comparación entre los valores para realizar el intercambio o la lectura de los datos.

4ta Parte:

Comparación donde se indicará si el recorrido termino de ordenar los datos en caso de no ser así seguirá recorriendo hasta terminar, en caso de que termine se preguntara si el contador “i” ya llego a su final si llega a ser este el caso se mostraran los resultados en los displays.

3.5. Asignación de Pines

Se utilizó para la implementación la tarjeta DE-10-Nano de la familia Cyclone V.

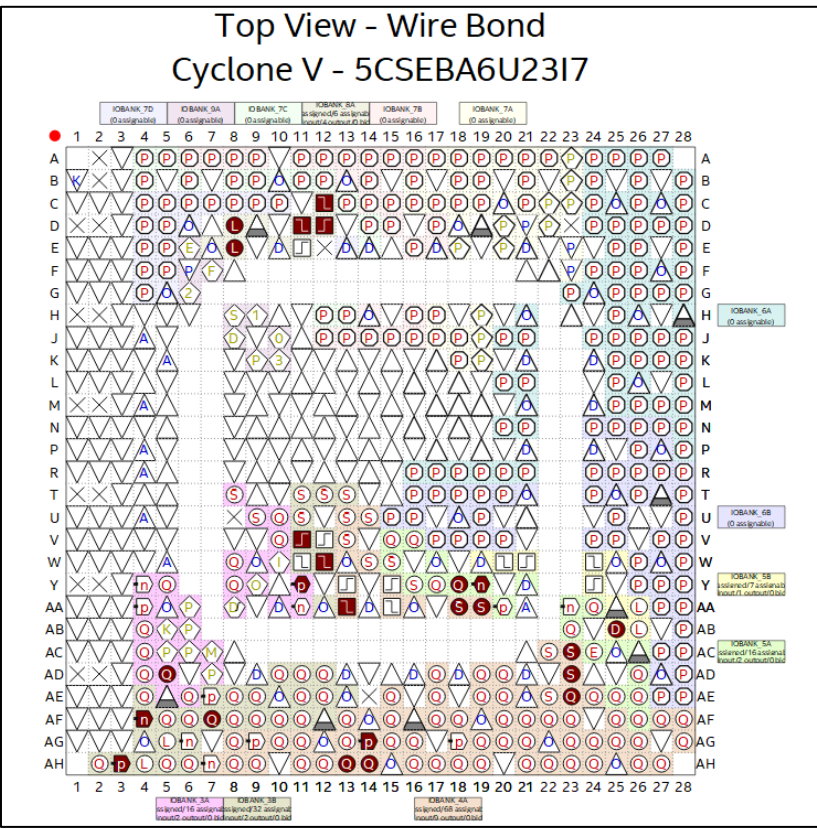


Ilustración 3. Tarjeta de Desarrollo para el Proyecto

Los pines a utilizar fueron:

Named: *	Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard
in	Clock	Input	PIN_V11	3B	B3B_No	PIN_V11	3.3-V LVTTTL
out	dec_led[6]	Output	PIN_AC23	4A	B4A_No	PIN_AC23	3.3-V LVTTTL
out	dec_led[5]	Output	PIN_Y19	5A	B5A_No	PIN_Y19	3.3-V LVTTTL
out	dec_led[4]	Output	PIN_AA19	4A	B4A_No	PIN_AA19	3.3-V LVTTTL
out	dec_led[3]	Output	PIN_AA18	4A	B4A_No	PIN_AA18	3.3-V LVTTTL
out	dec_led[2]	Output	PIN_Y18	5A	B5A_No	PIN_Y18	3.3-V LVTTTL
out	dec_led[1]	Output	PIN_AB25	5B	B5B_No	PIN_AB25	3.3-V LVTTTL
out	dec_led[0]	Output	PIN_Y11	3A	B3A_No	PIN_Y11	3.3-V LVTTTL
out	estados[4]	Output	PIN_AD5	3A	B3A_No	PIN_AD5	3.3-V LVTTTL
out	estados[3]	Output	PIN_AE23	4A	B4A_No	PIN_AE23	3.3-V LVTTTL
out	estados[2]	Output	PIN_AD23	4A	B4A_No	PIN_AD23	3.3-V LVTTTL
out	estados[1]	Output	PIN_D12	8A	B8A_No	PIN_D12	3.3-V LVTTTL
out	estados[0]	Output	PIN_C12	8A	B8A_No	PIN_C12	3.3-V LVTTTL
out	Fin	Output	PIN_AF4	3B	B3B_No	PIN_AF4	3.3-V LVTTTL
in	Mostrar	Input	PIN_AF7	3B	B3B_No	PIN_AF7	3.3-V LVTTTL
in	Resetr	Input	PIN_W12	3B	B3B_No	PIN_W12	3.3-V LVTTTL
in	Start	Input	PIN_D8	8A	B8A_No	PIN_D8	3.3-V LVTTTL

out uni_led[6]	Output	PIN_AA13	4A	B4A_N0	PIN_AA13	3.3-V LVTTTL
out uni_led[5]	Output	PIN_E8	8A	B8A_N0	PIN_E8	3.3-V LVTTTL
out uni_led[4]	Output	PIN_D11	8A	B8A_N0	PIN_D11	3.3-V LVTTTL
out uni_led[3]	Output	PIN_AH13	4A	B4A_N0	PIN_AH13	3.3-V LVTTTL
out uni_led[2]	Output	PIN_AH14	4A	B4A_N0	PIN_AH14	3.3-V LVTTTL
out uni_led[1]	Output	PIN_AH3	3B	B3B_N0	PIN_AH3	3.3-V LVTTTL
out uni_led[0]	Output	PIN_AG14	4A	B4A_N0	PIN_AG14	3.3-V LVTTTL

Ilustración 4.Lista de Asignación de Pines

Se utilizó la hilera GPIO 0 (JP1)

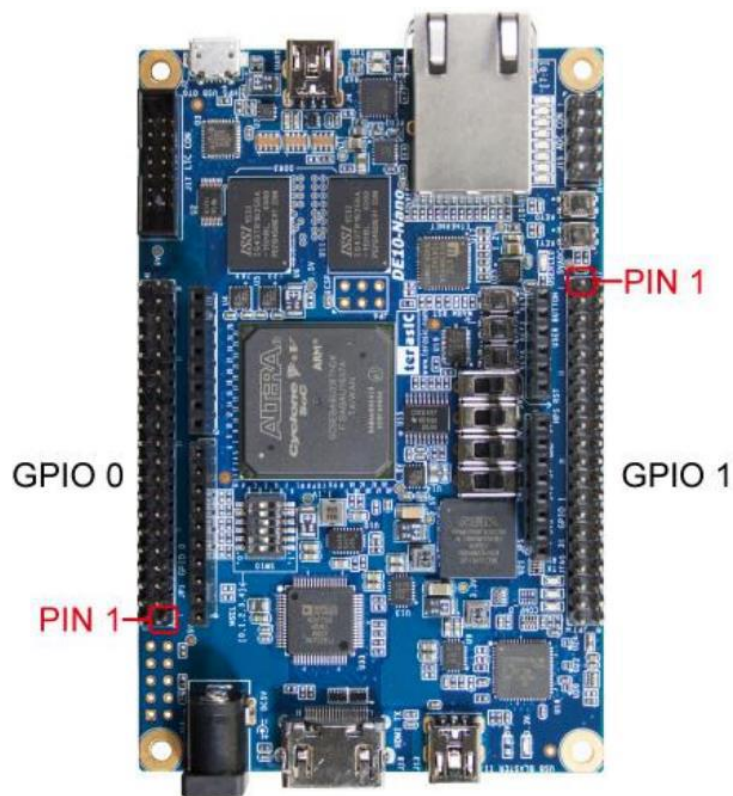






















Ilustración 5.FPGA DE10-Nano

GPIO 0 (JP1)					
PIN_V12	GPIO_0[0]	1		2	GPIO_0[1] PIN_E8
PIN_W12	GPIO_0[2]	3		4	GPIO_0[3] PIN_D11
PIN_D8	GPIO_0[4]	5		6	GPIO_0[5] PIN_AH13
PIN_AF7	GPIO_0[6]	7		8	GPIO_0[7] PIN_AH14
PIN_AF4	GPIO_0[8]	9		10	GPIO_0[9] PIN_AH3
	5V	11		12	GND
PIN_AD5	GPIO_0[10]	13		14	GPIO_0[11] PIN_AG14
PIN_AE23	GPIO_0[12]	15		16	GPIO_0[13] PIN_AE6
PIN_AD23	GPIO_0[14]	17		18	GPIO_0[15] PIN_AE24
PIN_D12	GPIO_0[16]	19		20	GPIO_0[17] PIN_AD20
PIN_C12	GPIO_0[18]	21		22	GPIO_0[19] PIN_AD17
PIN_AC23	GPIO_0[20]	23		24	GPIO_0[21] PIN_AC22
PIN_Y19	GPIO_0[22]	25		26	GPIO_0[23] PIN_AB23
PIN_AA19	GPIO_0[24]	27		28	GPIO_0[25] PIN_W11
	3.3V	29		30	GND
PIN_AA18	GPIO_0[26]	31		32	GPIO_0[27] PIN_W14
PIN_Y18	GPIO_0[28]	33		34	GPIO_0[29] PIN_Y17
PIN_AB25	GPIO_0[30]	35		36	GPIO_0[31] PIN_AB26
PIN_Y11	GPIO_0[32]	37		38	GPIO_0[33] PIN_AA26
PIN_AA13	GPIO_0[34]	39		40	GPIO_0[35] PIN_AA11

3.6. Diagrama Esquemático en Altium

Ilustración 7. Esquemático Parte 1

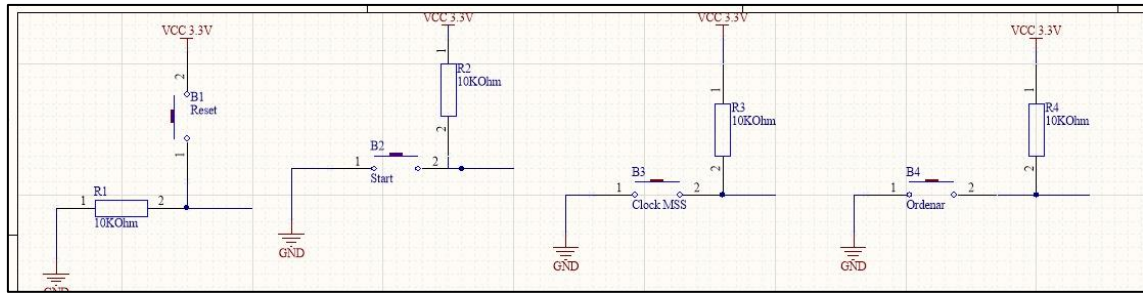


Ilustración 8. Esquemático Parte 2

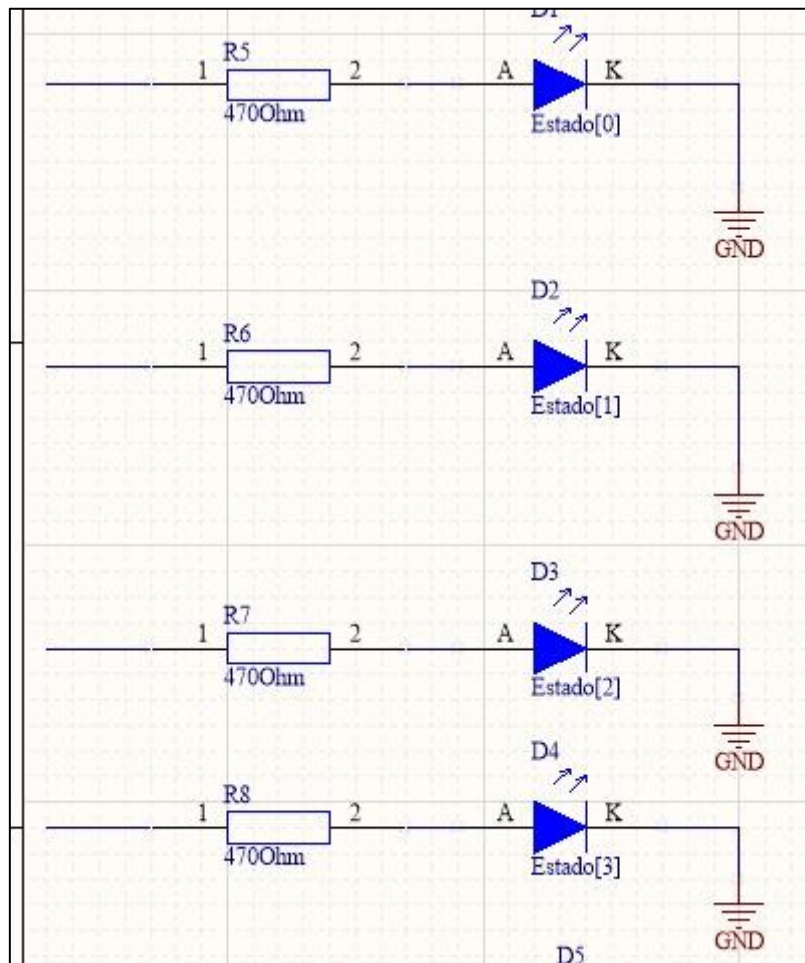


Ilustración 9. Esquemático Parte 3

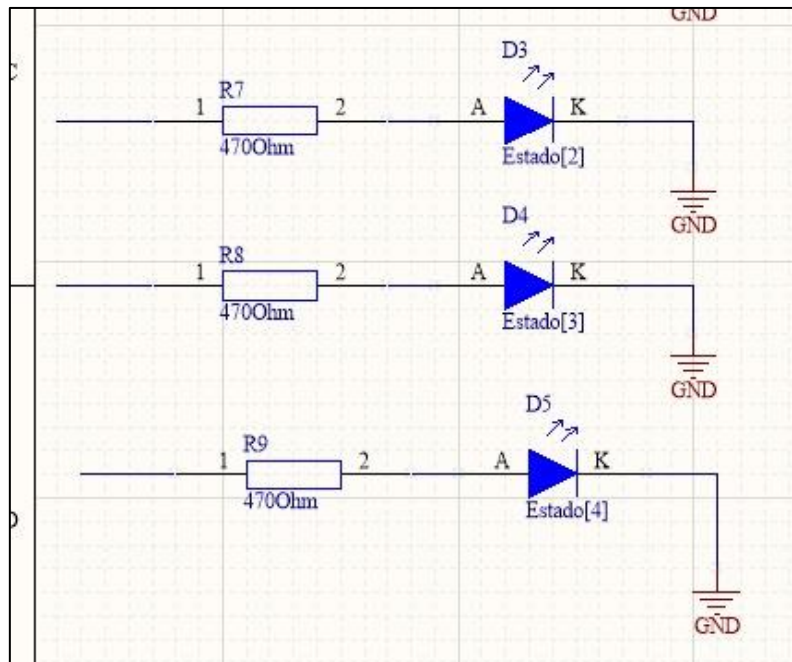


Ilustración 10. Esquemático Parte 4

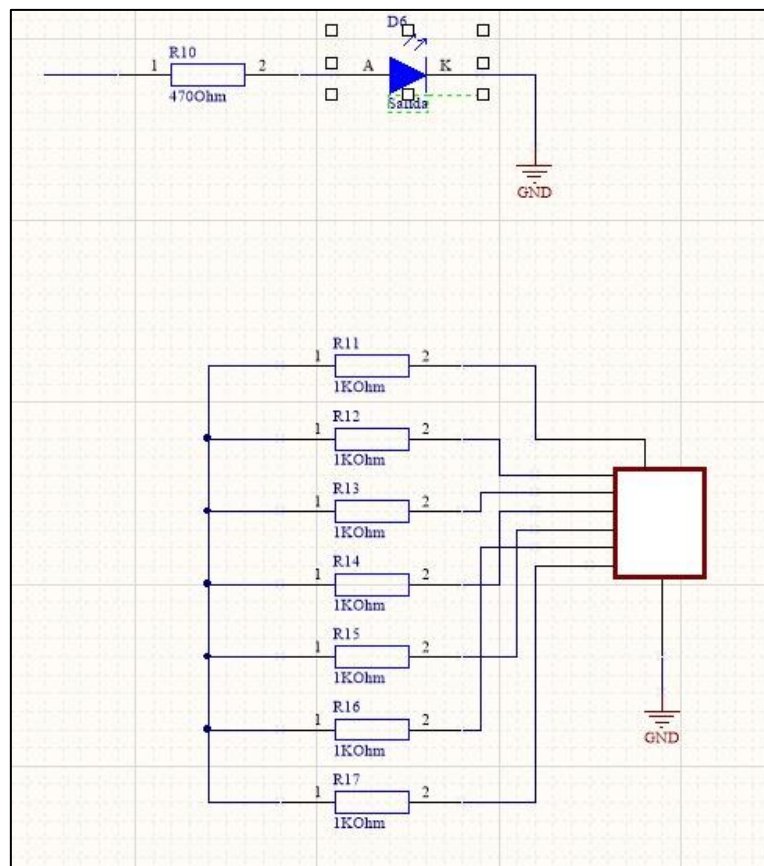


Ilustración 11. Esquemático Parte 5

CAPÍTULO 4

4. Aplicaciones a futuro

- Una mejora que se puede realizar es implementar la posibilidad de que se pueda ingresar los propios datos por el usuario y al no importar cuando se ingrese ya que debido a que la RAM realiza el proceso de ordenamiento se mostraran los números ordenados con los nuevos datos ingresados.
- Una aplicación al futuro podría ser que no solamente se reciba los datos de eficiencia sino también de cantidad o mostrar la calidad mediante letras siendo: "A" de alta calidad, "B" calidad aceptable, "C" calidad media, "D" baja calidad.
- Adicionando también que se pueda realizar un conteo específico de cuantas son de alta calidad, de calidad aceptable y así sucesivamente para saber si la base de datos cuenta con información que será útil ya que para el sector agricultor siempre se buscara una mejor calidad.

CAPÍTULO 5

5. Conclusiones y recomendaciones

5.1. Conclusiones

- Se diseñó un sistema digital para obtener la solución deseada a la problemática planteada implementando el ordenamiento de burbuja, para que el usuario pueda ordenar los datos de eficiencia de los fertilizantes de mayor a menor.
- Se logró ordenar los datos de mayor a menor y presentarlos mediante displays de 7 segmentos en el menor tiempo posible.

5.2. Recomendaciones

- En el momento de la implementación tener una sección para cada elemento a utilizar, una sección para las botoneras, una sección de leds y otras de estados que no se choquen entre si ya que al momento de conectar el bus de datos pueden unirse los cables y realizar el proceso puede ser complicado.
- Asignar un switch de la DE10-Nano como clock para que este en subida y no tener que ingresar el clock manual mente mediante botoneras

Anexos

Links al Repositorio de Github con todos los códigos empleados y vídeo de sustentación:

<https://github.com/RicRomBald/PROYECTO-FINAL-SD2>

<https://youtu.be/hCLq9N2mrcl>

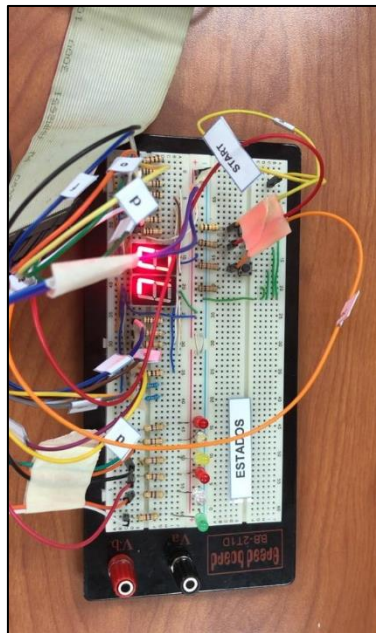


Ilustración 12. Protoboard con Circuito Implementado

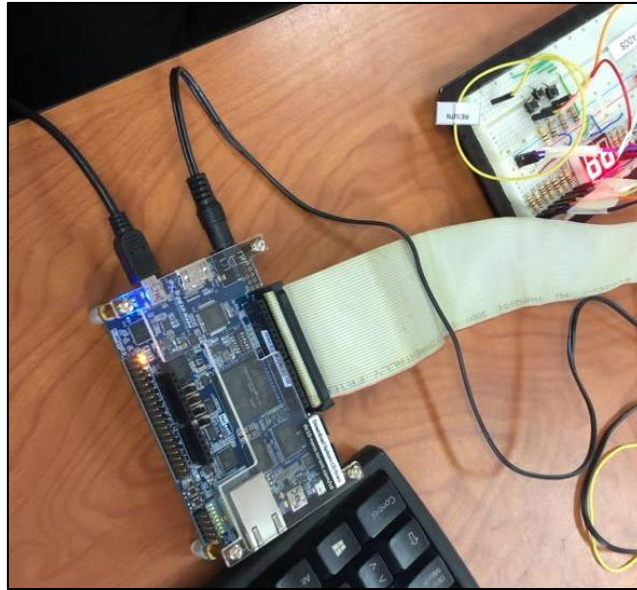


Ilustración 13. Circuito Implementado con FPGA