



**Universidad Tecnológica de Santiago**

**Facultad de arquitectura e ingeniería**

**Materia:**

**Taller de microprocesadores II**

**Trabajo:**

**Diseño de bancos de memorias para uC 8051**

**Profesor:**

**Ing. Raúl Toribio**

**Estudiante:**

**Ramses Garcia Dilone**

**Matricula:**

**1-10-0299**

**Fecha:**

**Martes, 27 de octubre de 2015**

## **DESCRIPCION PROYECTO**

Configuración de solo direccionamiento externo del uC 8051, utilizando un banco de memoria 12Kx8 con tres memorias ROM 4Kx8 y una memoria RAM 2Kx8. El circuito debe almacenar el código de programa en la memoria ROM 0. Se agregó un led en parpadeo en el P1 de uC para comprobar el funcionamiento del condigo.

Diseñar un banco de memoria ROM de tamaño total de 12K x 8 el microcontrolador debe ser conectado en configuración de solo memoria externa, para este diseño solo hay disponible memorias del tipo M2732 (4K x 8) una de ellas debe contener el código del programa creado, otra de ellas deberá contener las primeras 10 letras del alfabeto en carácter ASCII. Para el banco de memoria RAM externa solo se disponen de dos memorias del tipo HM6116 (2K x 8), el banco de memoria RAM y ROM debe ser dispuesto para que inicien en las direcciones 0000H. Para el programa desarrollado se debe realizar la operación de lectura de la tabla que dispone del abecedario y ser mostrado su código ASCII sobre el P1/P3 y desplegado sobre 7 segmentos.

## **OBJETIVO DEL PROYECTO**

Poner a prueba la habilidad de diseño de bancos de memoria para trabajar con el 8051, configurar de manera correcta el 8051, de forma que pueda ser capaz de direccionar el código de modo externo y quemar el archivo .hex en la memoria para que proteus pueda simular en circuito habilitando la opción de fetching en el micro. Crear una tabla con los códigos del alfabeto en ASCII para que puedan ser leídos desde la memoria ROM externa.

## **RECURSOS A UTILIZAR**

Para este circuito se disponen solo de memorias EEPROM 4Kx8 para en banco de memoria ROM y memorias SRAM 2Kx8 para la memoria RAM, se utiliza un latch 74HC373 para el arreglo del puerto P0, leds y resistencias además del cristal de 12Mhz para el reloj del uC con sus capacitores.

1-uC 80C51

3-Memorias eeprom 2732

2-Memoria sram 6116

1-Latch 74hc737

1-Display de 7 segmentos

-Dispositivos pasivos, resistencias, capacitores, etc..

## TABLA DE DIRECCIONAMIENTO

Esta tabla muestra el direccionamiento de todas la memorias y como están ubicadas en el banco de memoria.

A15	A14	A13	A12	A11	A10	A09	A08	A07	A06	A05	A04	A03	A02	A01	A0	HEXA	MEMORIA
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0X0000	MEORIA RAM 0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0X07FF	MEORIA RAM 0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0x0800	MEORIA RAM 1
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0X0FFF	MEORIA RAM 1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0X0000	MEMORIA ROM 0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0X0FFF	MEMORIA ROM 0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0X1000	MEMORIA ROM 1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0X1FFF	MEMORIA ROM 1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0X2000	MEMORIA ROM 2
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	0X2FFF	MEMORIA ROM 2

RAM	ROM	Tamaño	
	0x2FFF 0x2000	4K ROM 2	
	0x1FFF 0x1000	4K ROM 1	
0x0FFF 0x0800	0x0FFF 0x0000	2K RAM 1	2K ROM 0
0x07FF 0x0000		2K RAM 0	2K ROM 0

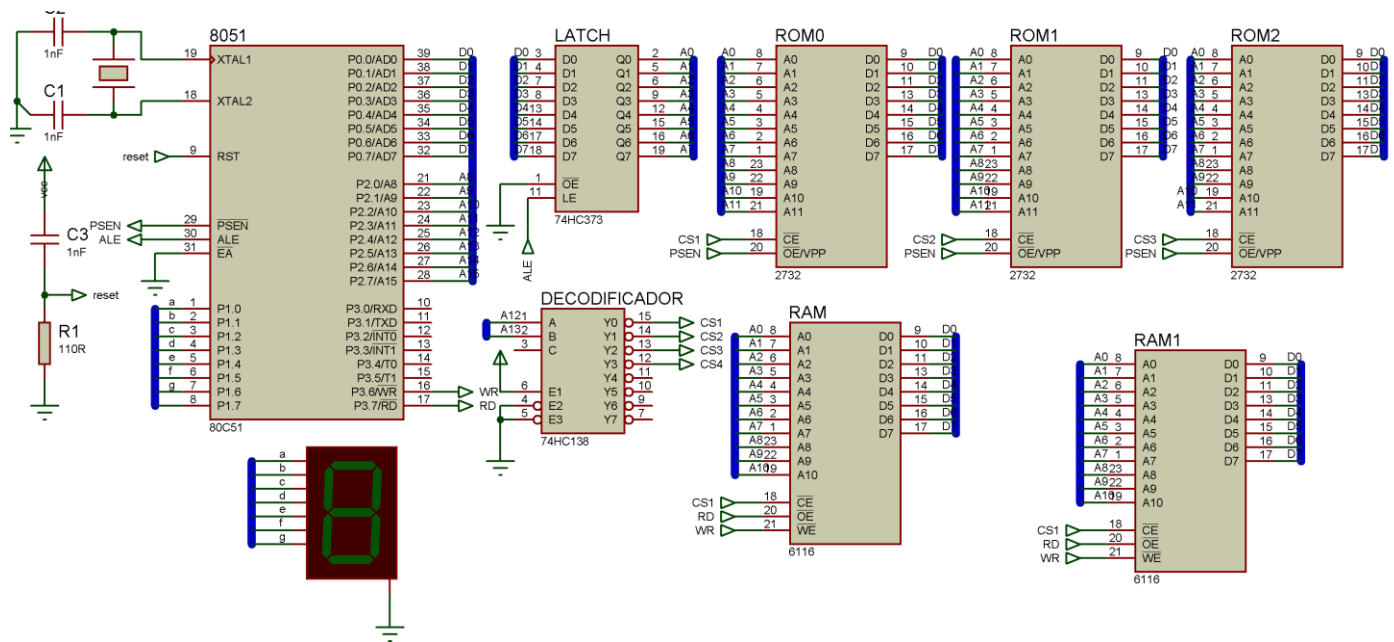
Tanto la Memoria RAM como la ROM deben empezar en la dirección 0x0000 por lo cual como se muestra en el mapa de direccionamiento la mitad del bloque de memoria pertenece a los 4K de RAM divididas en dos y el bloque completo representa los 4K de ROM 0 de la primera memoria, en la ROM 0 es donde se almacena el código del programa y la ROM 1 contiene la tabla con el código ASCII.

## DISEÑO ESQUEMATICO

Este diagrama muestra cómo debe conectarse el micro 8051 en modo de direccionamiento externo con un banco de memoria ROM 12Kx8 y RAM 2Kx8. El reto más difícil fue la conexión ya que hay que separar el bus de datos del bus de direcciones y ambos corresponden al P0, así que tomando solo las direcciones después del latch se elimina este problema, además de que la señal de oe del latch debe permanecer fija a tierra.

En la memoria ROM 0 se encuentra el código del programa escrito en lenguaje C, y en la memoria ROM 1 se encuentra una tabla que contiene el código ASCII de las primeras diez letras de abecedarios para ser mostradas en el display de 7 segmentos.

## DIAGRAMA.



## CONCLUSION

Los bancos de memoria nos proporcionan un diseño más robusto y de mayor tamaño al no depender de la capacidad ilimitada del micro, además podemos ser más puntuales con el manejo del código y las variables.