



Materia: Microprocesadores y microcontroladores

Maestro: García López Jesús Adán

Alumno: Christian Alfredo Montero Martínez

Practica 4: Sección de Memoria (Prueba de memoria RAM)

Fecha: 10/03/2017

## Práctica No. 4

### Sección de Memoria (Prueba de memoria RAM)

Objetivo: El alumno diseñará e implementará un decodificador para la memoria de la práctica 3 y hará uso de los procedimientos de la práctica 3 para probar la memoria.

#### Material:

- Computadora Personal (PC)
- Programa Edito de texto (ASCII), TASM y TLINK
- Tarjeta T-Juino (con intérprete 80x86)
- Compuertas lógicas (según diseño).
- Memoria RAM y Latch para T-Juino.
- Protoboard.

#### Equipo:

- Computadora Personal
- Programa emulador de terminal

#### Teoría:

##### Decodificadores a Memoria

La función básica de un decodificador es detectar la presencia de una determinada combinación de bits (código) en sus entradas y señalar la presencia de este código mediante un cierto nivel de salida. En su forma general, un decodificador posee  $n$  líneas de entrada para gestionar  $n$  bits. Para cualquier código dado en las entradas usualmente solo se activa una de las  $N$  posibles salidas, a menos de que el diseñador requiera que se repitan algunas salidas.

La decodificación de memoria es importante debido a que, entre otras características nos ofrece la posibilidad de conectar más de un dispositivo al microprocesador. Estos dispositivos pueden ser memorias ROM, RAM, Buffers, latches, etc. Asimismo nos permite asegurar un bloque de memoria a un determinado dispositivo. Es decir, la decodificación de memoria consiste en asegurar una determinada localidad/localidades de memoria para cada dispositivo externo al procesador.

Esto se logra mediante la combinación de líneas de control, selección y dirección, para generar una única señal de habilitación para el dispositivo deseado. Puesto que la mayoría de los dispositivos usan lógica negativa para habilitarlos, una técnica básica de decodificación es usando compuertas NAND. LA combinación de entradas a estas compuertas generan una salida en bajo (0 lógico) siempre que todas sus entradas estén en alto (1 lógico). Esta señal en bajo habilita un dispositivo en la dirección formada por la combinación de las señales de entrada.

Además los decodificadores se pueden utilizar para evitar que existan direcciones espejo; condición que se presenta cuando el dispositivo conectado al microprocesador tiene una cantidad menor de líneas de direcciones que el uP. Por ejemplo si se desea conectar una memoria con 2 líneas de direcciones (4 localidades) a un microprocesador con 4 líneas de direcciones (direcciona 16 localidades) dado que la memoria no puede ver 2 líneas de direcciones más significativas se accedería a la misma localidad en 4 ocasiones distintas, como se muestra en el siguiente gráfico:

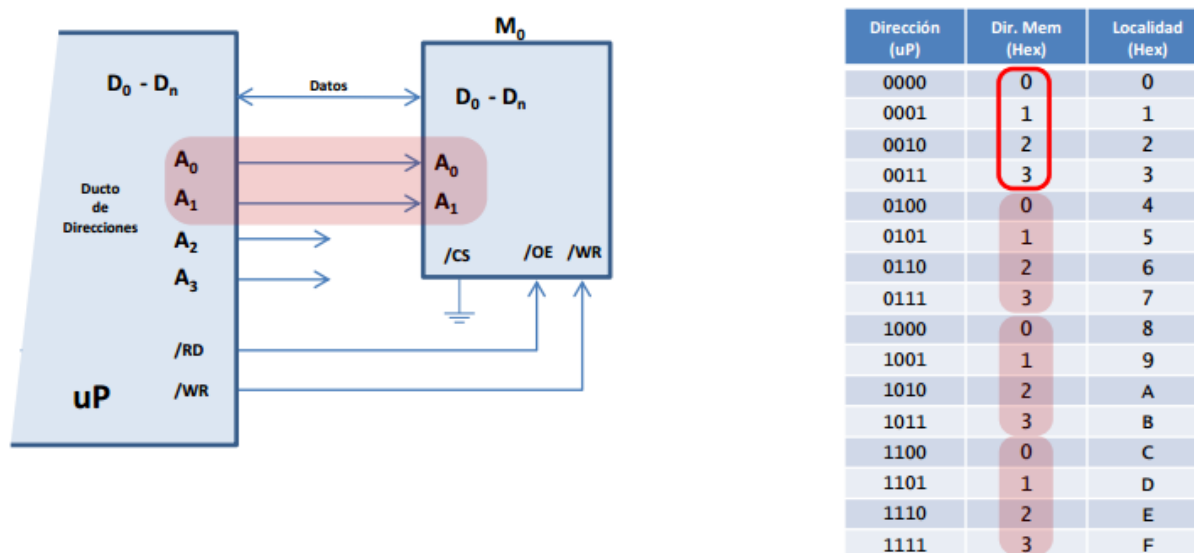


Figura 1: Visualización de las direcciones espejo.

Para lograr evitar esta condición sería necesario forzar a la memoria a “visualizar” las otras líneas de direccionamiento. Justo aquí es donde es necesario la implementación de un decodificador. Para diseñar el decodificador es necesario hacer un análisis para determinar bajo qué condiciones se desea activar la memoria, usualmente se utilizan mapas de Karnaugh para determinar la ecuación bajo la cual se diseñará la circuitería de nuestro decodificador. Una vez realizado el análisis las líneas adicionales entran al decodificador y se asigna una salida a cada combinación de entradas, por ejemplo si solo queremos activar la memoria al direccionar a las primeras 4 localidades, el siguiente gráfico muestra el procedimiento:

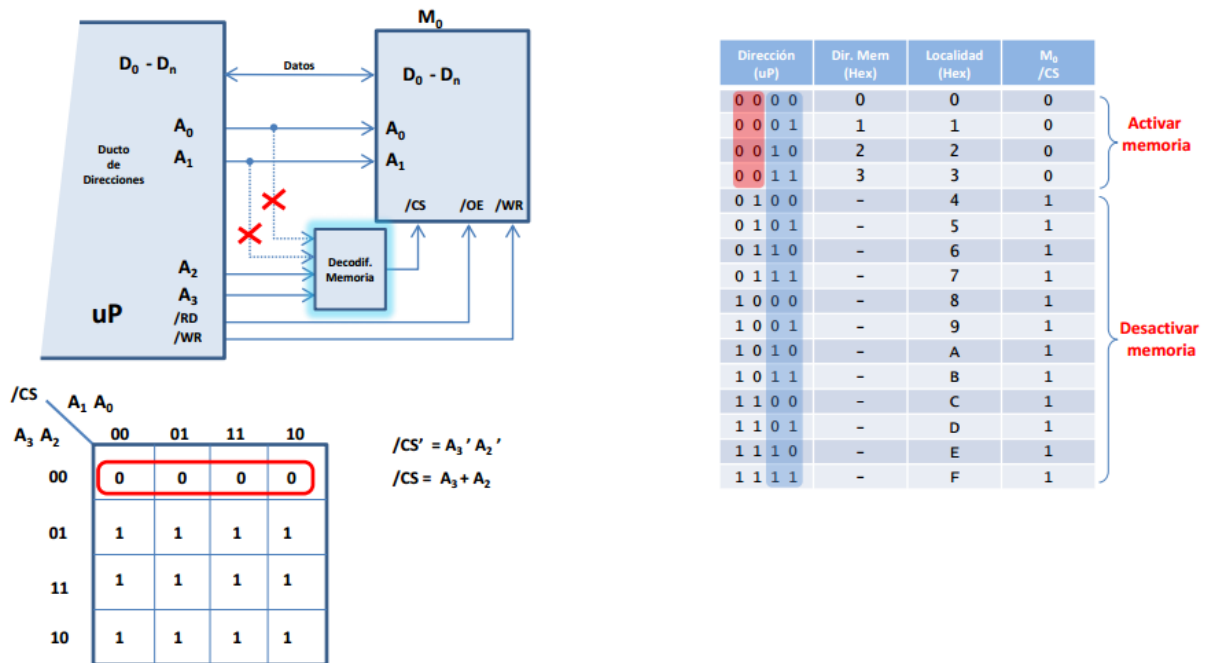


Figura 2: Decodificador para evitar direcciones espejo.

Asimismo si se requiere conectar dos memorias y que cada una de ellas direcciona a localidades diferentes, se podrían implementar dos decodificadores de la siguiente manera:

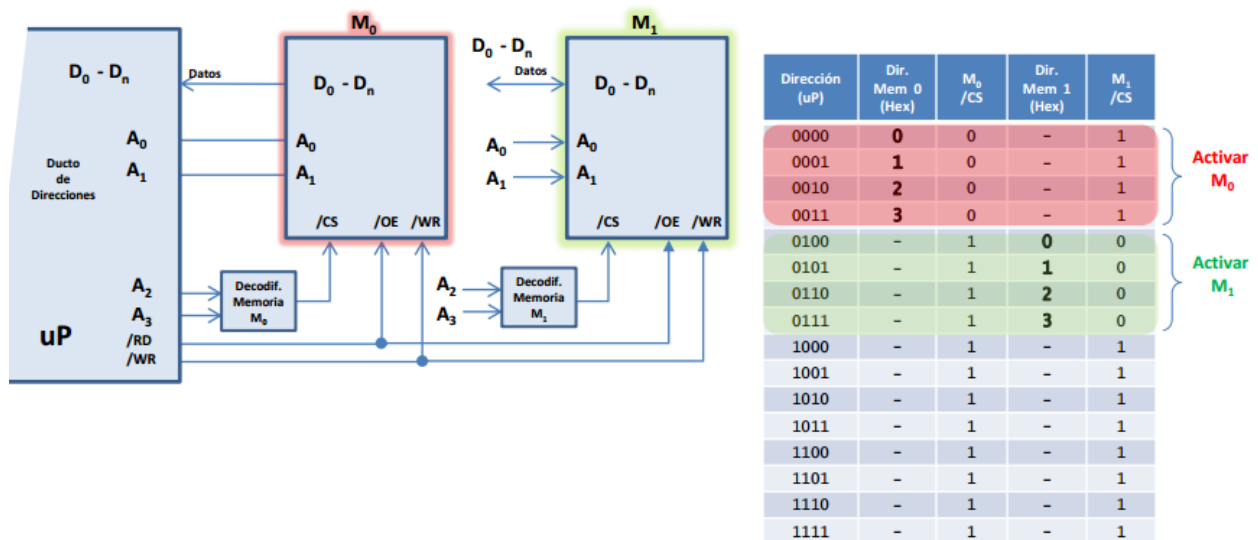


Figura 3: Implementación de 2 decodificadores para conectar 2 memorias.

## Desarrollo

1) Diseñe e implemente un decodificador de memoria para que una memoria RAM sea vista por el procesador en un determinado rango de memoria (el rango se define durante la sesión de Laboratorio).

Las direcciones que me fueron asignadas fueron las siguientes:

Dirección inicial: 3000h

Dirección espejo: B000h

Dado que se utilizó una memoria de 2K que cuenta con 11 líneas de direcciones y el procesador que cuenta con 16 líneas de direcciones, se realizó el siguiente análisis.

Dirección inicial	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Dirección inicial + 2K	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1
Dirección espejo	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
Dirección espejo + 2k	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1

Líneas al decodificador
Líneas a la memoria

Figura 4: Dirección inicial y espejo.

Como se puede observar las líneas en tomar a consideración son de A11 a A15 para diseñar el decodificador. Después de realizar el análisis mediante el uso de mapas de Karnaugh se llegó a la siguiente ecuación:

$$A'_{11}A_{12}A_{13}A'_{14}$$

Y el circuito se diseñó utilizando una compuerta AND y una NOT como se muestra en la siguiente figura:

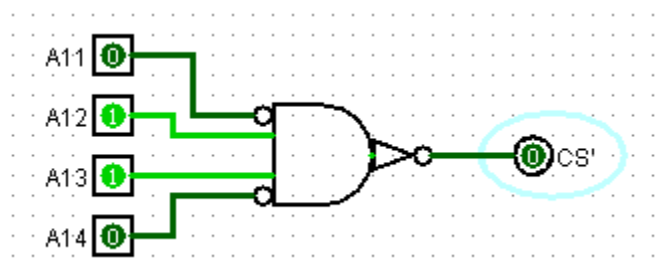


Figura 5: Circuito decodificador de direcciones.

2) Realice los cambios necesarios al programa de la práctica 3 para realizar la prueba de la nueva memoria.

a) Probar el rango correspondiente de la RAM. ¿Qué es lo que resulta de la prueba?

Corriendo el test de memoria a partir de la dirección 3000 funciona perfectamente imprimiendo la localidad a la cual está accediendo y permanece haciendo hasta haber recorrido 2K localidades, al llegar a la primera dirección fuera del rango, imprime la localidad y la letra F de falla.

b) Probar un rango no correspondiente. ¿Qué es lo que resulta de la prueba?

Al acceder a una localidad fuera del rango de direcciones para las cuales fue diseñado el decodificador, imprime la localidad a la cual accedió y la letra F de falla.

## Resultados

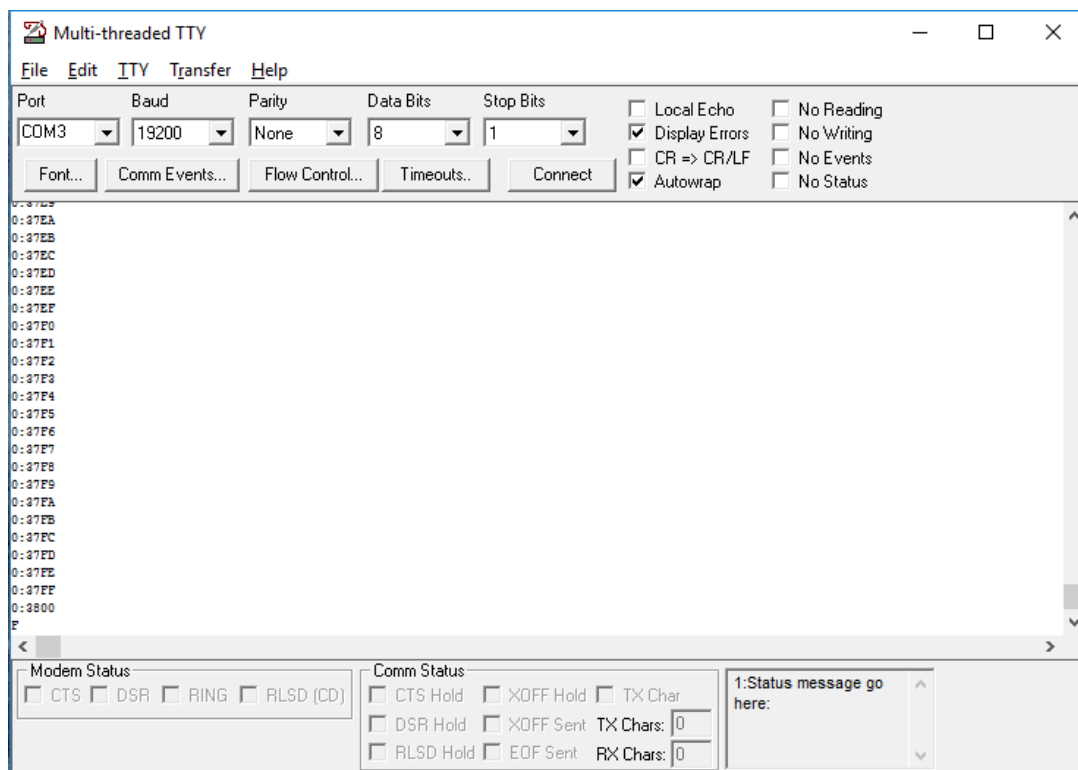


Figura 6: Corrida del programa en el rango (3000h -37FFh).

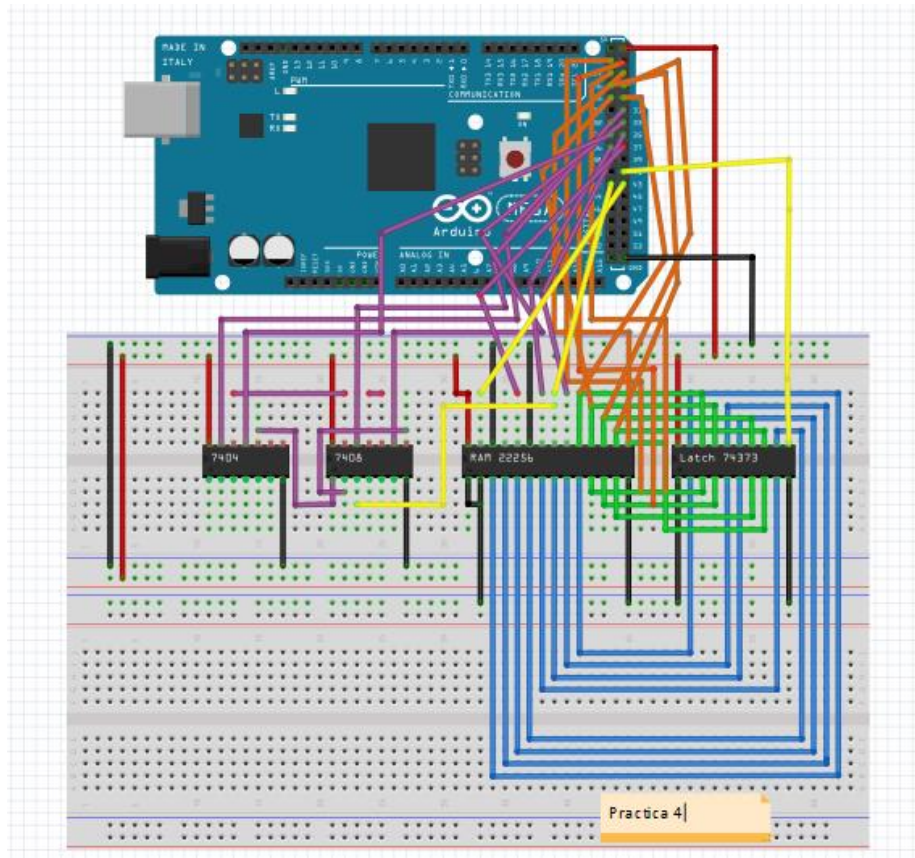


Figura 7: Esquema del circuito en Fritzing

## Conclusiones y Comentarios

La implementación de decodificadores de memoria es un tema muy importante ya que se puede dar solución a diversas situaciones que se nos pueden presentar al momento de conectar uno o más dispositivos de memoria al microprocesador, entre las más comunes encontramos evitar la existencia de direcciones espejo, asignar bloques de memoria específicos a determinados dispositivos conectados. Un caso particular, se presenta cuando queremos guardar simultáneamente en dos memorias un solo dato, y en una de ellas se almacena la parte más significativa y en otra la menos significativa. La implementación de los decodificadores se puede considerar una tarea sencilla ya que solo hace falta hacer un análisis no muy complejo y encontrar la ecuación mediante el uso de mapas de Karnaugh para diseñar el circuito.