Bitácora de Avances Proyecto 1

Prueba del Procesador Conectado a Puertos y Memoria RAM

Asignatura:

Microprocesadores

Presentado por:

Nombre: Cód.:

Darmael Alfredo Vargas Figueredo…………………………………1.118.560.349

Duvier de Jesús Bohórquez Palacio…………………………………1.094.948.815

Braulio Loaiza Castaño……………………………………………………1.096.646.602

Profesor:

Gerardo Andrés López Orozco

Programa de Ingeniería Electrónica

Facultad de Ingeniería

Universidad del Quindío

Armenia, Quindío

Marzo de 2016

Prueba del Procesador Conectado a Puertos y Memoria RAM

En este documento se darán las descripciones correspondientes sobre los procesos realizados hasta la fecha con respecto a la implementación de los puertos diseñados, en las pasadas sesiones, añadiendo un procesador junto con un componente complementario denominado latch y la inclusión de una memoria RAM, los cuales tendrán su correspondiente análisis, en términos del funcionamiento, modos de conexión y razón por el cual son adecuados de tal manera, entre otros detalles que se irán mostrando posteriormente, a mayor profundidad. Para comprobar el funcionamiento de estos dispositivos, se realizan pruebas a partir de la ejecución de algoritmos desarrollado en la plataforma de codeblocks, la cual permite el desarrollo de archivos que son compatibles con el formato que reconoce el procesador a simular.

Antes de proceder a realizar la descripción de cada simulación y/o implementación, es necesario comprender la función principal, tanto del procesador complementado con el latch, como la que cumple la memoria RAM. En primer lugar, observando el procesador atmega8515, se debe destacar el hecho de que este se encarga de garantizar la correcta ejecución de cada sección existente dentro del controlador.

Un ejemplo que se puede mencionar, para tener un entendimiento mas claro acerca de dicho concepto, es tener en cuenta las pruebas realizadas sobre el circuito que controla cada pin del puerto, puesto que en estas era necesario conectar varios componentes “logicstate” para enviar determinados valores, comprendidos entre unos y ceros, según el tipo de configuración a realizar sobre el pin, ya sea que se desee adecuarlo como un pin de entrada o de salida, o se requiera la lectura y/o escritura de un dato por medio del mismo. En efecto todas estas órdenes, traducidas al envío de unos y ceros, son las que le corresponde administrar al procesador y esta es su función principal pues, dichas acciones, son ejecutadas por este componente al realizar el reconocimiento y procesamiento de un algoritmo que contiene tales instrucciones.

Ahora, para describir el funcionamiento del componente denominado latch, es necesario entender el hecho de que cada sección o elemento del microcontrolador debe tener asignada una dirección específica con la cual el procesador puede reconocer el mismo y, de igual manera, esto le permite entender a quien debe enviar las órdenes y/o configuraciones determinadas por este, al procesar el algoritmo contenido allí. En ese sentido, es necesario hacer uso de este circuito integrado, el latch, para complementar el funcionamiento del procesador, puesto que este dispositivo contiene 16 salidas, que son referenciadas desde la 0 hasta la 15 pero las primeras 8, correspondientes a PA0/AD0 hasta PA7/DA7, son salidas bidireccionales que se utilizan para enviar y recibir datos, mientras que las siguientes 8 se utilizan para controlar las direcciones dispuestas desde la línea de dirección 8 hasta la 15. Por tal razón es necesario hacer uso del latch para controlar las direcciones desde la línea 0 a la 7, las cuales dependerán de los datos enviados por el procesador.

Por otra parte se hace uso de una memoria RAM, la cual se encarga de almacenar datos recibidos del procesador que, posteriormente, podría requerir y esto es útil en términos velocidad y/o eficiencia del mismo. Dicho componente tiene entradas que determinan su configuración puesto que este puede funcionar en modo de lectura o de escritura, a partir de los requerimientos del procesador y, por tal razón, dicho elemento controla el funcionamiento que requiera.

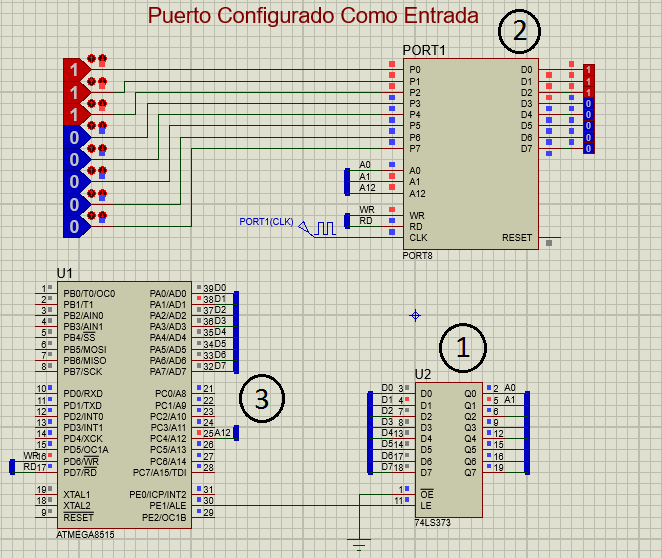


Figura 1: Circuito montado para probar el puerto realzado en la práctica anterior.

1. Integrado latch (74LS373): Este integrado controla las direcciones de A0 y A1 a partir de los datos que le transmite el integrado número 3. Para este caso tal componente maneja estas direcciones para comunicarse con el puerto utilizado
2. Puerto: Este es el puerto de lectura y salida, el cual se ha venido desarrollando desde el inicio del proyecto, en esta prueba el puerto solo esta configura en modo de lectura puesto que el código implementado en el integrado 3 funciona solo como lectura.
3. Microprocesador (ATMEGA8515): En este caso este es el que controla todas las funciones del puerto el cual lo configura ya sea de lectura o escritura, a través de los datos transmitidos al latch (integrado 1) y la dirección A12.

El código usado para la prueba con un solo puerto fue el siguiente:

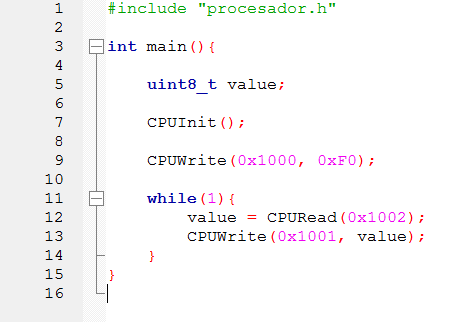


Figura 2: Código con el cual se probó el circuito de la figura 1.

Este código es muy sencillo puesto que consiste, básicamente, en la configuración constante e iterativa del puerto, cuya dirección es 0x1000. Inicialmente se crea una variable de tipo unit8\_t value, como se indica en la figura 2, la cual será utilizada posteriormente en el ciclo infinito while. Por otra parte se hace uso de funciones que permiten inicializar el procesador, la cual depende de CPUInit(); y establecer la configuración correspondiente en la dirección requerida, por medio de CPUWite(0x1000, 0xF0); En el ciclo while se configura el puerto como lectura, haciendo uso de la sentencia que se observa en la línea 12 para, posteriormente, realizar una escritura del dato adquirido allí, a partir de la función CPUWrite, como se observa en la línea 13 de la figura 2.

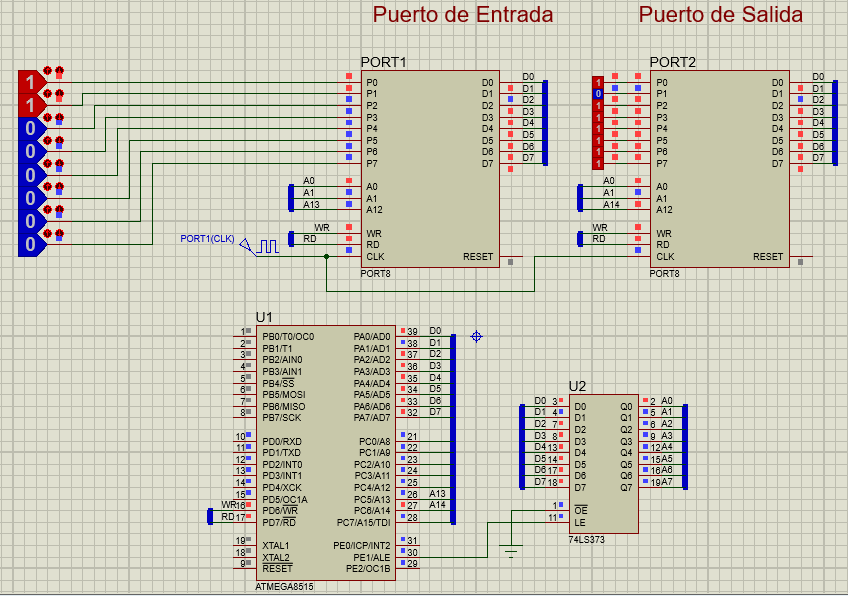
****

Figura 3: Circuito realizado para probar dos puerto, uno para lectura y otro para escritura.

Este circuito es igual al circuito de la figura 1, pero este ya tiene implementado otro puerto con el fin de tener un puerto para la entrada y uno para la salida. El microcontrolador, el latch y el puerto de entrada son los mismos, pero como se puede notar en el latch, este ya maneja ocho direcciones a diferencia del anterior que solo manejaba 2; esto se debe a que en el circuito hay un puerto de mas como salida, el cual requiere de las otras direcciones para poder trabajar.

En vista de que ya hay dos puertos, el código de instrucciones cargado en el microprocesador, también cambia para poder involucrar el puerto de salida. Este código es:

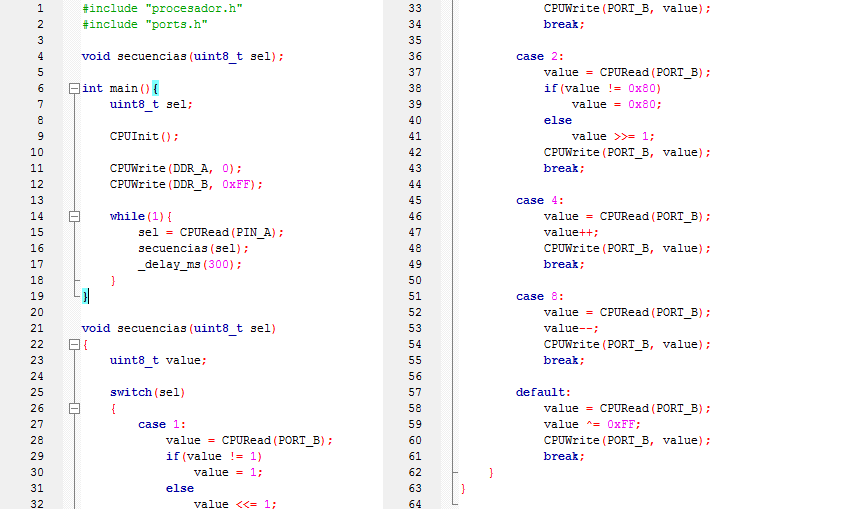


Figura 4: Código utilizado para probar el circuito con los dos puertos.

Para el segundo código se crea una variable uint8\_t llamada sel, luego se inicializa el microprocesador con la función CPUInit() y se procede a establecer los puertos de lectura o de escritura usando la función CPUWrite(Puerto, Rol), en este caso DDR\_A va a ser de lectura y DDR\_B de escritura.

Siguiendo con la estructura del programa se entra al ciclo while(1) donde la variable “sel” adquiere el valor leído por el PIN\_A usando la función CPURead() , luego se hace un llamado a la función ,secuencias declarada al principio del código y a esta se le pasa el valor entero de “sel”.

La función cuyo nombre es secuencias toma el valor de sel y define un valor uint8\_t llamado value para entrar a un switch case en donde:

-Caso 1: value adopta el valor adquirido por PORT\_B y entra en un bloque de decisión simple donde se mira si value es diferente de 1 el valor de value va a ser igual a 1 , en caso contrario value hará una asignación con desplazamiento a la izquierda (operador a nivel de bit que desplaza hacia la izquierda el valor de una variable el número de bits especificado en el valor de una expresión y asigna el resultado a la variable.), luego toma ese valor y lo escribe en el PORT\_B.

Caso 2:value adopta el valor adquirido por PORT\_B y entra en un bloque de decisión simple donde se mira si value es diferente a la dirección 0x80, de ser así tomará el valor de esta dirección, de caso contrario hace una asignación con desplazamiento a la derecha (operador a nivel de bit que desplaza hacia la derecha el valor de una variable el número de bits especificado en el valor de una expresión y asigna el resultado a la variable.), luego toma ese valor y lo escribe en el PORT\_B.

Caso 3(en el que sel toma el valor de 8):value adopta el valor adquirido por PORT\_B y le hace un decremento de 1, luego toma ese valor y lo escribe en el PORT\_B.

Default(por defecto, si no se cumple ninguno de los casos):value adopta el valor adquirido por PORT\_B y hace una asignación con XOR binario (operador a nivel de bit que realiza una operación OR exclusiva bit a bit en el valor de una variable y en el de una expresión, y asigna el resultado a la variable) a value con la dirección 0xFF luego escribe en el PORT\_B.

Y para finalizar el programa hay un retraso de 5 milisegundos entre ciclo y ciclo, esto para que no se sature de información el microcontrolador.

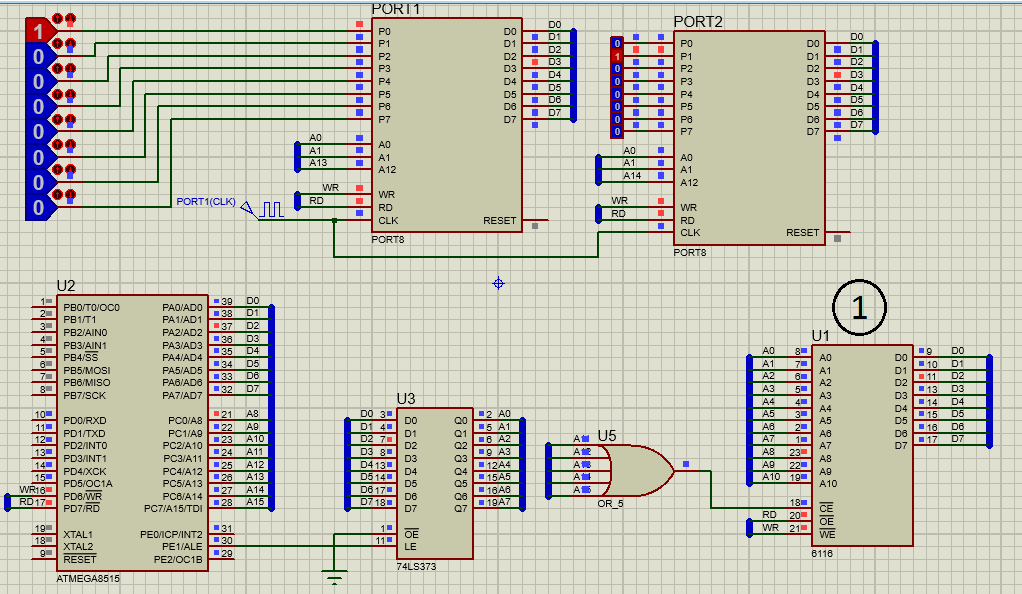


Figura 5: Circuito final para probar la memoria RAM.

Para este circuito todas las características de los anteriores son válidas, solo que a este se le sumo un integrado más para la memoria RAM, para la cual se utilizó el integrado 6116. Esta memoria RAM puede leer y escribir según lo declare el microprocesador, tal y como se menciona anteriormente. La memoria cuenta con 10 líneas de memoria de las cuales 7 son del latch y las otras tres salen del microprocesador. Es activado con un enable el cual es designado por una compuerta OR administrada por el microprocesador, ya que el circuito solo puede hacer una tarea a la vez. Esto se traduce al hecho de que, en este caso, los puertos tienen asignadas direcciones que se encuentran por fuera de estas líneas de dirección y, por lo tanto, si se trabaja con estos componentes, se encontrara activo uno de las direcciones posicionadas en la compuerta or, garantizando un uno que evitara la activación de la memoria RAM en ese instante de tiempo.