



GRABADOR DE MEMORIA AT28C256 POR MEDIO DEL PUERTO PARALELO

Nombre: GDEA

Afiliación: FCE

Dirección: Av. Sn. Claudio y 18 sur C.U.

E-mail: gdea@kim.ece.buap.mx

RESUMEN

Debido a la versatilidad que posee el puerto paralelo para la comunicación con una computadora hacia y desde el mundo exterior, es interesante desarrollar aplicaciones en las que se conjuguen diversos sistemas como se realiza en éste trabajo en donde se utiliza uno de los modos de operación del puerto paralelo que es el modo EPP (Puerto Paralelo Extendido) para grabar y leer información en una memoria AT28C256, de tal manera que éste sistema se vuelve dinámico pues debido a las características de la memoria, en cualquier momento se puede guardar y leer información utilizando un software en el que se especifica las direcciones y el dato que se guarda de tal manera que actúa como un compilador al generar el archivo con extensión HEX el cual se graba automáticamente en la memoria.

1. INTRODUCCION

El ser humano siempre ha albergado la idea de controlar todo su entorno, de controlar en cierto modo su destino, esta teoría se aplica incluso en niveles básicos, donde surge la necesidad de conocer y controlar cosas que ya han sido descubiertas y ampliamente utilizadas con la idea de emplear este recurso en un trabajo particular.

El presente trabajo no es la excepción, pues utiliza el hardware del Puerto Paralelo de una computadora para realizar un dispositivo que permita grabar información en una memoria, específicamente en la AT28C256.

2. EL PUERTO PARALELO

Es un hardware que nos permite realizar operaciones de lectura-escritura entre el mundo exterior y una PC, como es bien conocido, el puerto paralelo de una computadora posee un conector DB25 hembra, y posee diversos protocolos de comunicación, que son:

- SPP Standard Parallel Port
- EPP Enhanced Parallel Port
- ECP Extended Capabilities port

El protocolo que se ajusta mejor a nuestras necesidades es el EPP, pues, como se verá continuación, posee un bus de comunicaciones bidireccional, lo cual nos permitirá para

este caso específico leer y escribir información de nuestro dispositivo de memoria.

3. MODO EPP

El protocolo EPP realiza cuatro ciclos de transferencia:

1. Ciclo de escritura de datos
2. Ciclo de lectura de datos
3. Ciclo de escritura de direcciones
4. Ciclo de lectura de direcciones

Los ciclos de datos están pensados para transferir datos tanto al ordenador como al periférico. Los ciclos de direcciones son usados para transferir direcciones, canales, comandos e información de control. Las señales de salida del EPP poseen propiedades similares a las señales del SPP[1] y adicionalmente se provee protocolos para cada ciclo de transmisión de datos.

El EPP posee un nuevo grupo de registros. Sin embargo 3 de ellos se comportan de la misma forma que en el SPP. A continuación se muestra una tabla con los nuevos y los ya existentes registros del EPP.

Definición de Registros EPP

Registro	Offset	Mode	R/W	Descripción
SPP Data Port	+0	SPP/EPP	W	Puerto de datos standard SPP. Sin autostrobing.
SPP Status Port	+1	SPP/EPP	R	Lectura de las líneas Status de la interfase
SPP Control Port	+2	SPP/EPP	W	Maneja las líneas del registro de Control
EPP Address Port	+3	EPP	R/W	Genera un ciclo de lectura/escritura de direcciones concatenado
EPP Data Port	+4	EPP	R/W	Genera un ciclo de lectura / escritura de datos concatenado
Not Defined	+5 to +7	EPP	N/A	Usos diversos de acuerdo a la implementación. Puede usarse para transferencias de palabras 16 o 32 bits

Como se puede observar, las primeras tres direcciones son las mismas que en el SPP y se comportan del mismo modo. Es decir, si se usa el EPP, se puede enviar



información al registro Base + 0 en la forma como se realiza en el SPP.

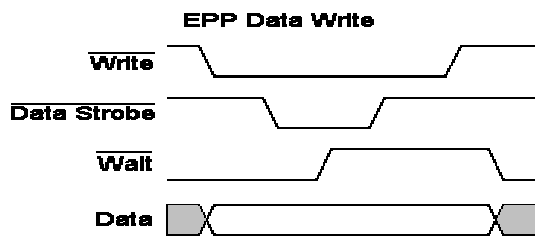
De este modo si desea comunicarse con un dispositivo externo, lo que tiene que hacer es colocar la información en el registro correspondiente (base + 4 o +3) y el EPP generará el handshaking requerido. El registro de Estado, tiene una pequeña modificación. El Bit 0, que era reservado en el SPP, ahora se llama el bit Time-Out. Este bit se cambiará cuando ocurra un tiempo excedido en el EPP. Esto pasa cuando la línea nWait no se descierta en aproximadamente 10 microsegundos desde que las líneas de escritura o lectura (IOW IOR del bus ISA) han sido acertadas.

Los tres registros, Base +5, Base + 6 y Base +7 pueden ser usados para operaciones de lectura escritura de 16 o 32 bits.

4. EL HANDSHAKE EN EL EPP

Para desempeñar un correcto intercambio de información con el EPP, se debe seguir el protocolo de handshake establecido para el EPP. Debido a que el EPP realiza todo el trabajo, el handshake tiene que ser implementado por hardware y no por software como en el caso del SPP. Para inicializar un ciclo EPP, el software solo necesita realizar una operación de lectura / escritura en el registro respectivo.

De este modo un ciclo típico de escritura de datos se realiza de la siguiente manera:



1. El programa ejecuta un ciclo de escritura E/S a la dirección base + 4 (Puerto de datos EPP).
2. La línea nWRITE indica la salida de datos hacia el puerto paralelo.
3. Los datos se ponen en las líneas 0-7.
4. La línea nDataStrobe se acierta si nWait esta en bajo (iniciar ciclo).
5. El EPP espera que el receptor ponga en alto la línea nWait (terminar ciclo).
6. El ciclo ISA E/S finaliza.
7. La línea nDataStrobe es desacertada.
8. El ciclo de escritura del EPP termina.

Una de las características mas importantes es que la transferencia de datos ocurre en un ciclo ISA E/S. Lo que

implica que la transmisión de datos un sistema puede mejorar las tasas de transferencias desde 500 K hasta 2Mbytes por segundo.

El ciclo de escritura de direcciones se realiza de la misma forma solo que en lugar de activar la señal DataStrobe activará la señal AddStrobe que como el mismo nombre de la señal indica que se está realizando un ciclo de escritura de direcciones[1].

El ciclo de lectura se realiza de diferente manera[1], Cabe mencionar que el flanco de subida de la línea DataStrobe determina el momento en el que se toma la información del bus de datos; el tiempo que permanece activa la señal DataStrobe(en bajo) es determinado por el hardware, es decir, que mientras la línea DataStrobe esté activa en bajo, debe estar el dato válido en el bus de datos, si no se tiene a tiempo la información ocurrirá un error al capturar datos que no son los que necesitamos, además en el modo de lectura, es importante que el dispositivo que envía la información active la señal Wait mientras la señal DataStrobe esté activa baja pues si no se realiza este requisito, el protocolo indicará un error, pues el Puerto Paralelo no puede estar esperando mas de 15 microsegundos para tomar los datos válidos. El ciclo de lectura de datos se realiza de la misma forma con la única diferencia que se activa la línea DataStrobe.

5. CONSIDERACIONES DE PROGRAMACIÓN EN EL EPP

Antes de que se pueda inicializar cualquier ciclo del EPP escribiendo o leyendo de los registros correspondientes, el puerto debe ser configurado correctamente. Así pues nuestro primer paso es inicializar el modo EPP usando los registros del SPP, basta con escribir 00000100 en el registro de control, y con esto el puerto se encuentra inicializado.

En algunos puertos, configurados en modo del lectura, el ciclo de escritura no se puede llevar a cabo, por lo que es necesario reiniciar el modo EPP configurando el bit 5 del registro de control.

Cuando el bit Time-Out del EPP se cambia, puede ocasionar que el EPP no funcione correctamente, por lo que basta con mantener monitoreado éste bit para corregir problemas.

6. ESCRITURA / LECTURA EN LA MEMORIA AT28C256

De acuerdo a las características propias de la memoria AT28C256[2] se puede realizar un primer acercamiento entre el puerto paralelo y la memoria de la cual se obtiene la siguiente información:

- 1) El puerto paralelo posee un bus de datos de 8 bits [1].



2) El proceso de lectura o de escritura en la memoria implica que se mande la dirección o los 15 bits en donde se va a guardar el dato, y posteriormente, enviar los 8 bits que contienen el dato.

Ésta primera relación nos dice que como solamente contamos con un bus de datos del puerto paralelo de 8 bits, se necesitará latchedear en primera instancia, 8 bits de la dirección, posteriormente los restantes 7 bits, para completar los 15 que se necesitan para la dirección, y por último mandar los 8 bits que contienen la información.

La memoria realiza los ciclos de lectura y escritura de la siguiente manera:

Ciclo de lectura:

La memoria AT28c256, es accesada como una memoria Ram estática, cuando nCE y nOE son activas bajas, y nWE es activa alta, la información es guardada en la locación de memoria determinada por los pines de dirección acertados(activos alto), las salidas del CI son puestas en estado de alta impedancia cuando tanto nCE o nOE estén en alto.

Ciclo de escritura:

Un pulso bajo en nWE o nCE con nCE o nWE en bajo(respectivamente) y nOE en alto, inicializa el ciclo de escritura. La dirección es latchedea en el flanco de bajada de nCE o nWE, o lo que ocurra al final. La información es latchedea en el primer flanco de subida de nCE o nWE.

Notar que se puede controlar la operación de lectura escritura con una sola línea[2], es decir, basta con controlar el estado lógico de una señal para asegurar la lectura o escritura en nuestro dispositivo de memoria.

Como se vio anteriormente, el modo EPP maneja dos protocolos, uno para escritura de datos y otro para escritura de direcciones, por conveniencia el protocolo de escritura de datos se utilizará para escribir los 8 bits de información válida en la memoria, pues podemos controlar el ciclo de escritura / lectura en la memoria con una sola señal; por lo que el ciclo de escritura de datos del EPP activa una señal que es nDataStrobe, así que el ciclo de escritura de la memoria, se puede controlar con la señal de nDataStrobe del EPP, como las dos son activas en bajo, no es necesario agregar lógica adicional; de este modo, hasta ahora aseguramos el proceso de escritura de los 8 bits del dato. Pero como es señalado en el protocolo del EPP, necesitamos una señal de retroalimentación o Handshake en la línea nWait, por lo que ese puede emplear la misma señal nDataStrobe para mandar la retroalimentación o Handshake al EPP, para hacer esto posible, debido a los niveles lógicos invertidos, lo que se procede es ha realizar la interfaz con un inversor lógico, cabe señalar que en el protocolo de escritura del EPP, se maneja un tiempo de retraso para la llegada de la señal, y

este tiempo de retraso mínimo es agregado por el inversor lógico.

Para cubrir las necesidades de la memoria en las líneas nOE y nWE utilizaremos la línea nWrite del EPP conectada directamente a la terminal nWE pues no existe problema con el acoplamiento de los niveles lógicos, y para conectar la línea nWrite a nOE, se utilizará de interfaz un inversor lógico, pues para realizar un proceso de escritura o lectura en la memoria, se requiere que nOE este en alto, nWE en bajo y nWE esté en alto , nOE en bajo respectivamente.

Con esto aseguramos el cumplimiento en ambos protocolos, tanto del EPP como de la memoria para el proceso de lectura/escritura.

Ahora, procedemos a implementar el proceso de latchedeo de los 15 bits de dirección para la memoria, y se realiza utilizando el protocolo de escritura que , como se vio anteriormente, el proceso de escritura de direcciones en el EPP genera la activación de una señal nAddressStrobe que podemos utilizar como señal de mando para guardar la información en un latch; por comodidad se utilizará el latch 74HC373, el cual debido a sus características[3] nos permite realizar el proceso de latchedeo de la siguiente manera:

El protocolo de escritura de direcciones del EPP genera una señal nAddressStrobe activa baja, esta señal nos sirve para guardar los 7 bits más significativos de la dirección de la memoria en el match, solamente haciendo un cambio en el nivel lógicos de la señal nAdressStrobe para conectarla directamente a la línea Latch Enable del 74HC373, cabe mencionar que esta misma señal se puede utilizar para cubrir las necesidades de handshake (nWait)del mismo modo como se cubrieron en el proceso de escritura de datos en la memoria; por lo tanto manteniendo en nivel de la línea de entrada Output Control del latch en bajo, el comportamiento del latch será de la siguiente manera, al recibir el pulso alto en la entrada Latch Enable , los datos que estén en las líneas 1D-7D, se guardarán, una vez que el nivel de la línea Latch Enable regresa a bajo, la información que se guardó previamente se tendrá en las salidas 1Q-7Q, las cuales se conectan a los 7 bits mas significativos de la dirección de la memoria.

Se observa que no se utilizó el último bit del latch y el 8º bit del bus del puerto paralelo, puesto que no es necesario, pues se necesitan solo 7 bits para la parte alta de la dirección de la memoria.

Hasta este momento nos hace falta proporcionar el byte menos significativo de la dirección de la memoria, y como se han utilizado los dos protocolos de escritura/lectura del EPP se han utilizado, procedemos a crear un protocolo de comunicación, para lo que nos



pueden servir las señales del SPP, es decir, antes de inicializar el modo EPP en el puerto paralelo podemos utilizar el modo SPP para proporcionar el byte restante para la dirección de la memoria, para lo que se utiliza el siguiente algoritmo:

- 1) Se activa la señal nInit del SPP, es decir, se escribe en el bit 3 del registro de control del SPP, escribiendo en la dirección base +2 [1] un número decimal 4 que en binario represente un 1 en el bit 3; con lo que ponemos en alto la señal Latch Enable del Latch.
- 2) Escribimos la dirección en el bus de datos del SPP (dirección base), con esto le enviamos los datos al Latch. Cabe mencionar que una vez que escribimos un dato en el bus del SPP, éste se mantendrá en el bus hasta que no se escriba otro dato.
- 3) Se desactiva la señal nInit escribiendo en el registro de control base +2 un número 0 para regresar al estado inicial, con lo que guardamos la dirección en el latch.
- 4) Por último escribimos un byte de 0's en el bus del SPP para regresar al estado inicial.

Con esto hemos cubierto todas las necesidades de escritura y lectura en la memoria y el EPP, por lo que el algoritmo completo de lectura/escritura del sistema completo es el siguiente:

Lectura:

- 1) Escribimos el LSB utilizando el protocolo arriba descrito, con esto almacenamos el LSB en el Latch y automáticamente lo proporcionamos a la memoria.
- 2) Inicializamos el modo EPP escribiendo un 1 en el bit 5 del registro de control del SPP.
- 3) Escribimos los 7 bits más significativos de la dirección en el latch utilizando del proceso de escritura de direcciones del EPP.

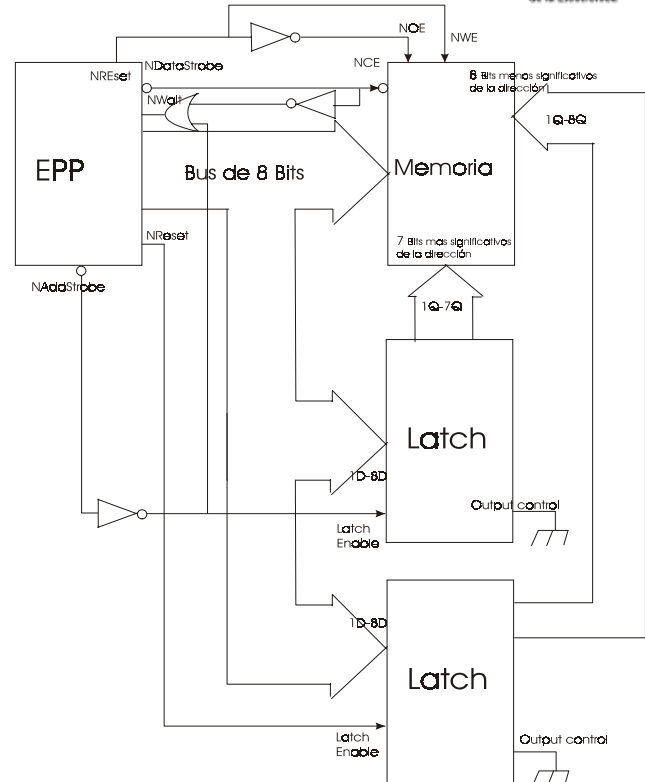
Se lee el dato válido en la memoria utilizando el proceso de lectura de datos del EPP

Escritura:

- 1) Escribimos el LSB utilizando el protocolo arriba descrito, con esto almacenamos el LSB en el Latch y automáticamente lo proporcionamos a la memoria.
- 2) Inicializamos el modo EPP escribiendo un 1 en el bit 5 del registro de control del SPP.
- 3) Escribimos los 7 bits más significativos de la dirección en el latch utilizando del proceso de escritura de direcciones del EPP.

Escribimos el dato válido en la memoria utilizando el proceso de escritura de datos del EPP.

La siguiente figura nos muestra el diagrama de construcción del sistema:



El control sobre el proceso de lectura/escritura, se realiza por medio de software programado en Microsoft Visual Basic de tal manera que puedo generar un archivo al estilo ensamblador en donde se indica la dirección y el dato para escribir, para leer se determina una dirección de inicio de lectura y los resultados de la misma se escriben en un archivo asignado por el usuario.

7. CONCLUSIONES

Este sistema nos permite guardar información simulando un sistema de almacenamiento típico de un computador, por lo que bien nos puede servir para cualquier aplicación electrónica y aquellas en las que necesite guardar información en memoria. Como trabajo futuro se propone implementar el sistema en un circuito impreso con la mejora de añadir protocolos de lectura/escritura de diferentes dispositivos de memorias.

8. REFERENCIAS

- [1] <http://www.beyondlogic.org/epp/epp.htm>
- [2] <http://www.atmel.com/atmel/acrobat/doc0006.pdf>
- [3] www.ee.washington.edu/stores/DataSheets/cd4000/74hc373.pdf

9. AGRADECIMIENTOS



Al laboratorio de Electrónica digital por brindarnos un espacio de trabajo.

