



CENTRO DE INGENIERÍA Y DESARROLLO INDUSTRIAL

Nombre del proyecto:

**Desarrollo de un controlador de memorias NAND Flash basado en un
microcontrolador para datos de 8 bits.**

Presenta:

Ing. Jorge Alberto Tirzo García.

Para obtener el grado de:
Tecnólogo en Mecatrónica

007746

Nombre del tutor:

Dr. Jorge Alberto Soto Cajiga.

Nombre del asesor:

M. C. y T. Noé Amir Rodríguez Olivares.

Santiago de Querétaro, Querétaro, Noviembre de 2015

RESUMEN

En este trabajo se presenta el desarrollo de un controlador NAND Flash, para la memoria de la marca Hynix con capacidad de 4 Gbits, utilizado para el almacenamiento de información, como unidad de procesamiento se hace uso de un microcontrolador de la marca Microchip modelo PIC18LF4550. El desarrollo de un controlador para memorias NAND Flash es necesario ya que durante las aplicaciones en donde se requiere almacenar información, el dispositivo de control maestro se enfocará en otras tareas esenciales y la tarea de almacenamiento de datos quedara bajo la dirección del controlador NAND Flash. La elección de esta memoria fue realizada porque las memorias NAND Flash son escritas de forma secuencial y cada página puede ser escrita un número finito de veces antes de tener que ser borrada por bloques, lo que requiere que se aproveche al máximo el espacio disponible por página para el almacenamiento. La velocidad con la que se almacenan los datos son de mucha importancia para los procesos en donde se requiera una velocidad de almacenamiento rápido, la velocidad con la que lee el controlador es de 437 KB/s y la velocidad con la que los datos son adquirido y escritos es de 0.97 KB/s.

AGRADECIMIENTOS

Agradezco a Dios y... a los que olvidé mencionar.

CONTENIDO

RESUMEN.....	I
AGRADECIMIENTOS.....	II
LISTA DE FIGURAS	IV
LISTA DE TABLAS.....	VI
1 PLANTEAMIENTO DEL PROBLEMA	1
1.1 Antecedentes	1
1.2 Planteamiento del problema	2
1.3 Justificación.....	3
1.4 Objetivo general	3
1.5 Objetivos específicos.....	3
1.6 Cronograma de actividades	4
1.7 Alcances	4
2 MARCO TEÓRICO	5
2.1 Unidades de almacenamiento.....	5
2.2 Unidades de almacenamiento de estado sólido	6
2.3 Memorias Flash	6
2.4 Memoria NAND Flash	8
2.5 Controladores flash	9
2.6 Características de controlador NAND Flash	10
2.7 Arquitecturas de los controladores actuales	12
2.8 Microcontroladores	14
2.9 LabVIEW™	15
2.10 Osciloscopio	17
3 DESARROLLO DE PROYECTO.....	19
3.1 Arquitectura de microcontrolador	19
3.2 Memoria NAND Flash	20
3.3 Funciones	21
3.4 Tareas para la memoria NAND Flash	27
3.5 Tarea de lectura de ID	30

3.6 Puntero de Dirección	31
4 EXPERIMENTACIÓN Y RESULTADOS.....	33
4.1 Conexión de memorias.....	33
4.2 Conexión de memoria y el microcontrolador.....	33
4.3 Validación de Tareas.....	34
4.4 Pruebas de la tarea lectura.....	37
4.5 Pruebas de la tarea escritura.....	38
4.6 Prueba de borrado de memoria.....	40
4.7 Interfaz gráfica	42
5 CONCLUSIONES Y RECOMENDACIONES	43
6 BIBLIOGRAFÍA	44
7 ANEXOS	46

LISTA DE FIGURAS

Figura 2.1 Estructura interna de memoria NOR Flash [7].	8
Figura 2.2 Estructura interna de memoria NAND Flash [7].	9
Figura 2.3 Diagrama lógico de una memoria NAND Flash [8].	10
Figura 2.4 Ventana principal de LabVIEW™.....	17
Figura 2.5 Osciloscopio Analog Discovery [19]......	18
Figura 3.1 Ejemplo de arquitectura de una memoria NAND Flash de 4Gbits [22]	20
Figura 3.2 Función comando [8].	22
Figura 3.3 Función dirección [8]......	23
Figura 3.4 Función escritura de datos [8]......	24
Figura 3.5 Función lectura de datos [8]......	25
Figura 3.6 Función de lectura de estado [8].	26
Figura 3.7 Lectura de Página [8].	28
Figura 3.8 Programación de Página [8]......	29
Figura 3.9 Borrado de un bloque [8].	30
Figura 3.10 Lectura de ID [8]......	30
Figura 3.11 Diagrama de flujo de Puntero de Dirección.....	31

Figura 4.1 Conexión de memorias Hynix.....	33
Figura 4.2 Diagrama de conexión del microcontrolador y la memoria.....	34
Figura 4.3 Tarea lectura de página emitida por el microcontrolador.	35
Figura 4.4 Tarea escritura de página emitida por el microcontrolador.	35
Figura 4.5 Tarea Borrado de bloque emitida por el microcontrolador.....	36
Figura 4.6 Lectura de ID recibida por el microcontrolador.....	36
Figura 4.7 Lectura de una señal senoidal de una página.	37
Figura 4.8 Representación gráfica de tiempos en que se ejecuta la tarea lectura de página.	37
Figura 4.9 Escritura de un mismo dato en una página de la memoria.....	38
Figura 4.10 Representación gráfica de tiempos en que se ejecuta la tarea escritura de página.	39
Figura 4.11 Escritura de una señal senoidal en una página.	39
Figura 4.12 Representación gráfica de tiempos en que se ejecuta la tarea escritura de página con adquisición de datos por el microcontrolador.....	40
Figura 4.13 Borrado de toda la memoria.....	41
Figura 4.14 Representación gráfica de tiempos en que se ejecuta el borrado de toda la memoria.....	41
Figura 4.15 Interfaz gráfica en LabView™.....	42
Figura 7.1 Diagrama de conexión de la tarjeta de memoria Hynix y el microcontrolador PIC18LF4550	46
Figura 7.2 Simulación de Lectura de ID de memoria Hynix en el software ISIS 7 Professional.	47
Figura 7.3 Simulación de Borrado de página de memoria Hynix en el software ISIS 7 Professional.	47
Figura 7.4 Simulación de Escritura de página de memoria Hynix en el software ISIS 7 Professional.	48
Figura 7.5 Simulación de Lectura de página de memoria Hynix en el software ISIS 7 Professional.	48

LISTA DE TABLAS

Tabla 1.1 Cronograma de actividades	4
Tabla 2.1 Ventajas de memorias NAND Flash y NOR Flash [5].....	7
Tabla 2.2 Función de señales de memoria NANA Flash [7].....	11
Tabla 3.1 Comparación de un microcontrolador de la marca Microchip y Texas Instruments [20], [21].....	19
Tabla 3.2 Parámetros de tiempos Máximos y Mínimos para el envío de la función comando [8].	21
Tabla 3.3 Parámetros de tiempos Máximos y Mínimos para el envío de comandos [8].....	22
Tabla 3.4 Parámetros de tiempos Máximos y Mínimos para el envío de comandos [8].....	25
Tabla 3.5 Definición del Registro de Estado [8].	27
Tabla 3.6 Mapa de ciclos de dirección	32

PLANTEAMIENTO DEL PROBLEMA

1 PLANTEAMIENTO DEL PROBLEMA

1.1 Antecedentes

Se entiende por unidad no volátil, a la unidad de almacenamiento que permite almacenar de forma temporal los datos y/o programas para ser utilizados en el momento adecuado y poder ser modificados, vueltos a guardar y recuperados cuando se desee [1]. Son por lo tanto dispositivos que guardan permanentemente la información en ausencia de alimentación, siendo muchos de ellos capaces de ser transportables, es decir, poder llevar la información a otro equipo o guardarla como copia de seguridad. Las unidades de almacenamiento son fundamentales en cualquier equipo que utilice procesadores.

Existen varios tipos de dispositivos que almacenan información como los son los ópticos, magnéticos y los dispositivos flash, estos últimos son los más utilizados actualmente, debido a que los dispositivos anteriores son más susceptibles a daños como lo son golpes, ralladuras, desgaste, etc. Un aspecto importante por el que los dispositivos flash han ganado terreno en el área de almacenamiento de datos es debido a la rapidez con la que ejecutan la lectura, escritura y borrado, además, cuentan con un tamaño mucho menor que el de los primeros dos dispositivos mencionados anteriormente, esto los convierte en dispositivos fáciles de transportar. En muchas aplicaciones de sistemas electrónicos, y en general en cualquier sistema de instrumentación, es necesario almacenar grandes cantidades de datos en donde las memorias seriales EEPROM suelen ser insuficientes, por lo tanto, el uso de las memorias tipo flash brinda una gran ventaja, otorgando gran capacidad de almacenamiento y una gran disponibilidad en el mercado a muy bajo costo. Los dispositivos flash se encuentran en diferentes presentaciones como los son: Compact flash, Memory stick, Smart drive, Pendrive [2].

Los dispositivos flash están diseñados para ser implementados en equipos que cuentan con terminales de conexión previamente definidas por los fabricantes, por lo cual si se requiere realizar alguna aplicación en donde se requiere almacenar información, es necesario implementar elementos físicos y de software para poder realizar las conexiones necesarias,

PLANTEAMIENTO DEL PROBLEMA

así como también realizar previamente los protocolos de comunicación para el correcto funcionamiento.

Debido a lo anterior, realizar alguna aplicación donde se desee almacenar datos, es necesario respetar las especificaciones del fabricante y el desarrollar alguna aplicación en donde se requiera una alta rapidez de almacenamiento de información, la gestión de datos se verá restringida por la velocidad de trabajo de la memoria.

Actualmente en CIDESI (Centro de Ingeniería y Desarrollo Industrial), se ha desarrollado una arquitectura para el almacenamiento redundante del tipo RAID 6, utilizado para el almacenamiento de información obtenida de 80 señales de ultrasonido, el trabajo desarrollado utiliza dos marcas de memoria de la marca Hynix modelo HY27UF084G2B y Micron modelo MT29F64G08CBAAA, la arquitectura del controlador está basada en una tarjeta FPGA de la familia Virtex-4 de Xilinx, se probó para 8 memorias de 8 GB, 6 de estas para información (equivalente a 48 GB) y 2 para redundancia de tipo suma y producto (16 GB) [3].

1.2 Planteamiento del problema

En aplicaciones donde se requiere el almacenamiento masivo de información y la unidad de procesamiento es un microcontrolador, comúnmente se utilizan memorias SD, memorias EEPROM, entre otras, sin embargo, este tipo de unidades de memoria no satisfacen los requerimientos de velocidad de escritura para procesos donde la generación de la información es a alta velocidad. Las memorias NAND Flash se presentan como una solución a este problema, debido a que permiten tiempos de escritura cortos y son útiles para almacenamiento secuencial, es por eso que las memorias NAND Flash son la unidad de almacenamiento preferida para el desarrollo de los discos duros de estado sólido, sin embargo, para poder utilizar una memoria se requiere tener un control exacto de los tiempos de las distintas señales que sirven como interfaz, y conocer las rutinas necesarias para las distintas tareas en las memorias, a pesar de desarrollarse los controladores de memorias NAND Flash comúnmente en Arreglos de Compuertas Programables en Campo

PLANTEAMIENTO DEL PROBLEMA

(FPGA por sus siglas en inglés), en la etapa de diseño de los controladores, se realizan múltiples modificaciones a los programas, situación compleja y tardada para realizar en el FPGA.

1.3 Justificación

Desarrollar un controlador de memorias NAND Flash es totalmente pertinente, debido a que permitirá generar un nuevo sistema de almacenamiento masivo que genere menos cuellos de botella en el proceso de almacenamiento de la información. Además, la estructura de los datos en las memorias NAND Flash es de 8 bits, y si se utiliza un microcontrolador de 8 bits es posible alcanzar una compatibilidad directa en el tratamiento de los datos.

A pesar de que los controladores NAND Flash son comúnmente desarrollados en FPGAs, el utilizar un microcontrolador como controlador de las memorias permitirá comprender mejor el comportamiento de las memorias. Clarificará el diseño que se pueda realizar con un FPGA, y permitirá realizar pruebas más rápido a controladores basados en FPGA.

1.4 Objetivo general

El objetivo general de este trabajo es: Desarrollo de un controlador de memorias NAND Flash basado en un microcontrolador para datos de 8 bits.

1.5 Objetivos específicos

- Estudio de las rutinas necesarias para comunicación con las memorias.
- Selección del microcontrolador y las memorias útiles para la aplicación.
- Desarrollo de las rutinas para la realización de las distintas tareas en las memorias NAND Flash.
- Propuesta de arquitectura electrónica del controlador.

PLANTEAMIENTO DEL PROBLEMA

- Pruebas del controlador para las distintas tareas.
- Pruebas finales y validación.

1.6 Cronograma de actividades

Tabla 1.1 Cronograma de actividades

CRONOGRAMA DE ACTIVIDADES																			
ACTIVIDAD	MES																		
	MAYO				JUNIO				JULIO				AGOSTO				SEPTIEMBRE		
SELECCIÓN DE MICROCONTROLADOS	X	X	X	X	X	X													
PROPUESTA DE ARQUITECTURA				X	X	X													
PROGRAMACION DE COMANDOS					X	X	X	X											
ARMADO Y PROGRAMACION DE TAREAS									X	X	X	X	X	X					
PRUEBAS EN PROTOBOARD													X	X	X	X			
PRUEBAS DE RENDIMIENTO																	X	X	
PRESENTACION Y RESULTADOS																		X	X

1.7 Alcances

-Desarrollar un controlador para una memoria NAND Flash, utilizando un microcontrolador como elemento de control.

Realizar una interfaz entre una computadora y el microcontrolador para validar el almacenamiento de datos en las memorias NAND Flash.

2 MARCO TEÓRICO

En este capítulo se muestran los fundamentos necesarios para llevar a cabo la implementación de un controlador NAND Flash. Para ello se describe brevemente la estructura de este capítulo, que consiste primeramente en mostrar las unidades de almacenamiento de datos, seguido de los controladores actuales para las memorias Flash y las características de los microcontroladores.

2.1 Unidades de almacenamiento

Los datos con los que trabajan los equipos electrónicos tales como ordenadores, celulares, dispositivos de instrumentación, etc. se almacenan de forma temporal en la memoria RAM. Pero esta memoria necesita de corriente eléctrica para mantener la información y todos los datos alojados en ella se borran cuando se apagan. Para mantener los datos de forma definitiva, estos equipos incorporan dispositivos magnéticos, ópticos y flash que son empleados comúnmente para el almacenamiento de datos. Los dispositivos magnéticos consisten en aplicaciones de campos magnéticos a ciertos materiales cuyas partículas reaccionan a esa influencia, generalmente orientándose en unas determinadas posiciones que conservan tras dejar de aplicarse el campo magnético [4].

Un dispositivo óptico consiste en una superficie circular de policarbonato donde la información se guarda haciendo unos surcos en la superficie del disco. El acceso a los datos se realiza cuando un material especial del disco, que suele ser de aluminio, es iluminado con un haz de láser [4].

El dispositivo flash es el que permiten la lectura o escritura de múltiples posiciones de memoria en una misma operación mediante impulsos eléctricos. Este tipo de memorias funcionan a velocidades muy superiores cuando los sistemas emplean lectura y escritura al mismo tiempo [4].

2.2 Unidades de almacenamiento de estado sólido

Actualmente los dispositivos Flash han abarcado una gran parte en los sistemas de almacenamiento utilizando principalmente la tecnología NAND Flash. Una de las principales aplicaciones son las unidades de estado sólido.

Una unidad de estado sólido (SSD) es un dispositivo de almacenamiento de datos que utiliza la memoria de estado sólido para almacenar datos con la intención de brindar acceso de la misma manera que una unidad de disco duro (HDD) tradicional. A partir de 2007, la mayoría de las SSD utilizan memoria NAND Flash no volátil para conservar los datos y no contienen piezas móviles. En comparación con las HDD, las SSD son menos susceptibles a los impactos físicos, son silenciosas, tienen menor tiempo de acceso, de latencia y brindan un rendimiento mucho más alto. Las SSD usan la misma interfaz y el mismo factor de forma que las unidades de disco duro tradicionales, lo cual hace que sean más fáciles de reemplazar en la mayoría de las plataformas informáticas [5].

2.3 Memorias Flash

2.3.1 Tecnologías de memoria NAND Flash y NOR Flash

La memoria flash es un tipo de dispositivo no volátil. Es una evolución de las memorias EEPROM, pero con una gran mejora tanto en el aspecto económico como en el físico, ya que son más pequeñas y por lo tanto fáciles de transportar. Además la velocidad es mucho mayor con respecto a las otras memorias existentes debido a su capacidad de borrar y escribir en una misma operación.

Para el uso de las memorias flash, es necesario borrar bloques de datos antes de poder escribir en ellos, lo que hace que el rendimiento de escritura sea menor que el de lectura. La

MARCO TEORICO

memoria flash sólo admite un número finito de escrituras y el número varía según la tecnología utilizada.

La memoria flash está disponible como NAND o NOR. La memoria NAND flash es más duradera, menos cara, sus celdas son más densas y las operaciones de escritura/borrado son más rápidas en comparación con la memoria NOR flash. La memoria NOR flash está diseñada para almacenar el código binario de programas y tiene un alto rendimiento en operaciones de lectura [6].

Cada una de las tecnologías NOR y NAND tiene ventajas que la hacen ideal para distintos tipos de aplicaciones, como se resumen en la tabla 2.1 [5].

Tabla 2.1 Ventajas de memorias NAND Flash y NOR Flash [5].

	Memorias NOR Flash	Memorias NAND Flash
Acceso de alta velocidad	Si	Si
Acceso a los datos en modo paginado	No	Si
Acceso aleatorio de nivel de bytes	Si	No
Usos comunes	Teléfonos celulares* Almacenamiento de BIOS para PCs Memoria de dispositivos de red	PDA's Cámaras digitales Teléfonos celulares** Reproductores MP3 Unidades de disco duro (estado sólido) Cajas Set-Top Almacenamiento industrial

2.3.2 Memoria NOR Flash

La memoria NOR Flash, así llamada por la tecnología de asignación de datos específicos (en inglés "Not OR"), es una tecnología de memoria Flash de alta velocidad. La memoria

* Para sistemas operativos

** Para almacenamiento de datos

MARCO TEORICO

NOR Flash proporciona capacidades de acceso aleatorio de alta velocidad, pudiendo leer y escribir datos en lugares específicos de la memoria sin tener que acceder a la memoria en modo secuencial. A diferencia de la memoria NAND Flash, la memoria NOR Flash permite la recuperación de datos desde un solo byte. La memoria NOR Flash es excelente en aplicaciones donde los datos se recuperan o se escriben de manera aleatoria. NOR se encuentra más frecuentemente integrada en teléfonos celulares (para almacenar el sistema operativo del teléfono) y Asistentes Digitales Personales; también se utiliza en computadoras para almacenar el programa BIOS que se ejecuta para proporcionar la función de arranque. [5]

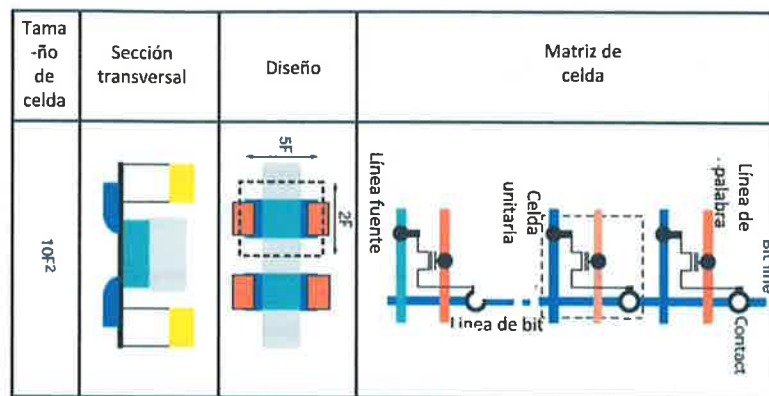


Figura 2.1 Estructura interna de memoria NOR Flash [7].

2.4 Memoria NAND Flash

La memoria NAND Flash fue inventada después de la memoria NOR Flash y tomó su nombre de la tecnología de asignación específica utilizada para datos (en inglés "Not AND"). La memoria NAND Flash lee y escribe a alta velocidad, en modo secuencial, manejando datos en bloques de tamaño pequeño ("páginas"). La memoria NAND Flash puede recuperar o escribir datos como páginas únicas, pero no puede recuperar bytes individuales como la memoria NOR Flash [5].

La memoria NAND Flash se encuentra comúnmente en unidades de disco duro de estado sólido, dispositivos Flash de medios digitales de audio y video, decodificadores de

MARCO TEORICO

televisión, cámaras digitales, teléfonos celulares (para almacenamiento de datos), y otros dispositivos donde los datos se escriben o leen, generalmente de manera secuencial. Por ejemplo, la mayoría de las cámaras digitales usan película digital basada en memoria NAND Flash, ya que las imágenes por lo general se toman y almacenan de manera secuencial.

La memoria NAND Flash también es más eficiente para leer las imágenes, pues transfiere páginas completas de datos muy rápidamente. Como medio secuencial de almacenamiento, la memoria NAND Flash es ideal para el almacenamiento de datos [5].

La memoria NAND Flash es más económica que la memoria NOR Flash y puede guardar mayor capacidad de datos almacenados en el mismo tamaño de bloque. La memoria Flash que almacena un solo bit por celda (por ej., el valor de "0" o "1" por celda) se denomina memoria Flash de celda de nivel único (SLC) [5].

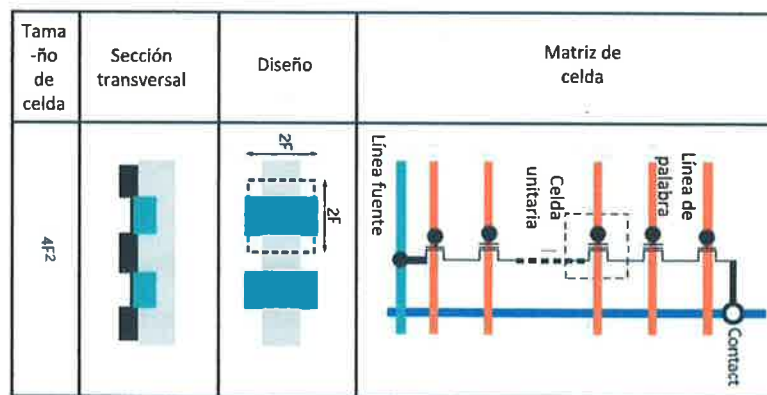


Figura 2.2 Estructura interna de memoria NAND Flash [7].

2.5 Controladores flash

Para el desarrollo de este proyecto se tomará en cuenta el uso de la memoria NAND Flash, por ser esta la mejor opción al contar con una alta capacidad para almacenar información y por el modo secuencial con el que almacena los datos. Por lo cual es indispensable la implementación de un controlador que realice la comunicación con la memoria.

MARCO TEORICO

El uso de la memoria NAND Flash se ve restringido debido a sus características intrínsecas tales como: la escritura consecutiva por página, el borrado de bloque y el número finito de escritura de la misma página antes de ser borrada nuevamente.

Por lo anterior, es debido seleccionar un controlador que tenga la capacidad de ejecutar las operaciones en los tiempos necesarios establecidos por el fabricante de la memorias NAND Flash.

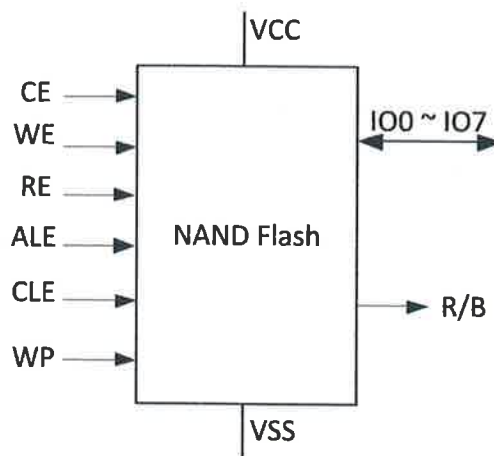


Figura 2.3 Diagrama lógico de una memoria NAND Flash [8].

Si el controlador puede soportar velocidades altas de transferencia de datos, el uso de controladores Flash optimizados, puede dar como resultado ahorros de tiempo importantes al leer o escribir datos en una memoria Flash [5].

2.6 Características de controlador NAND Flash

Un controlador para memorias NAND Flash es necesario para establecer el tiempo en el que se ejecutan los comandos, la velocidad de acceso de datos y para mejorar el tiempo de vida útil en las memorias. Una memoria Flash comúnmente se comprende de 8 tipos de señales como se muestra en la figura 2.3. Donde las señales cumplen las siguientes funciones mostradas a continuación en la tabla 2.2.

MARCO TEORICO

Tabla 2.2 Función de señales de memoria NANA Flash [7].

Señal	Función
IO0~IO7	Datos de Entrada/Salida
CLE	Latch de activación de comando
ALE	Latch de activación de dirección
CE	Habilitar memoria
RE	Habilitar lectura
WE	Habilitar escritura
WP	Protección de escritura
RB	Listo/Ocupado

Existen 6 funciones básicas para realizar el control de acceso de datos en las memorias, estas funciones son las siguientes: entrada de comandos, entrada de direcciones, entrada de datos, salida de datos, protección de escritura y modo de espera [8], estas funciones se controlan mediante la activación de las diferentes señales de entrada de la memoria

Los dispositivos NAND flash cuentan con seis señales de control (CE, ALE, CLE, RE, WE, WP), 8 o 16 señales de datos y una señal de respuesta de la memoria (RB). Los comandos son desplazados en el dispositivo a través de los pines de datos cuando el pin CLE está activo. Si el comando contiene la dirección, la información de dirección cambia al dispositivo a través de la señal de datos cuando ALE está activo. RE y WE se utilizan para desplazar los datos hacia o desde el dispositivo NAND flash. Para el comando que requiere mucho tiempo para procesar, tales como borrar, o programa, la señal de R/B indica cuando el dispositivo está ocupado [9]. WP es activado cuando se necesita hacer alguna escritura o borrado.

Antes de que un bloque pueda ser utilizado para los nuevos datos, este debe ser borrado. Esta operación establece todos los bits en el bloque a "1", y la escritura los modifica poniendo los bits a "0". Una vez que un bit es puesto a "0", solo se puede establecer el bloque a "1" por medio del borrado. El área de datos de una página puede ser escrita hasta cuatro veces entre los borrados. Las páginas de un bloque deben ser escritas de forma secuencial, de bajas direcciones a altas direcciones. Una vez que una página se ha escrito,

las páginas anteriores en el bloque ya no pueden ser escritas hasta después de la próxima escritura o borrado del bloque completo [10].

2.7 Arquitecturas de los controladores actuales

Para el desarrollo de este proyecto se realizó una búsqueda de los controladores actuales de las memorias NAND Flash, con el objetivo de considerar características que se deben tomar en cuenta para el desarrollo de un controlador, también esto servirá para conocer los avances que se tienen a cerca de este tema, así como también tener una base para poder iniciar el diseño del mismo.

NAND-Flash EVAL de SEGGER es una tarjeta de evaluación que puede ser programada mediante vía USB o vía JTAG / SWD. La tarjeta NAND-Flash EVAL está equipada con una toma de TSOP de 48 pines para la memoria flash NAND y cuenta con un microcontrolador basado en ARM Atmel® SAM3U (MCU). El MCU está preprogramado con una aplicación que automáticamente formatea una memoria flash NAND sin formato y se ejecuta una prueba de rendimiento. El rendimiento de Escritura es de 4,2 Mb/s, el rendimiento de lectura es de 6.4 Mb/s, Otra característica importante es que la tarjeta "NAND-Flash EVAL" viene precargado con una aplicación USB-MSD. Cuando se conecta a un PC a través de USB, la tarjeta se muestra como un dispositivo de almacenamiento extraíble en el sistema operativo host [11].

El controlador NAND Flash ONFI 3.2 IP Core, es fácil de usar, cuenta con un diseño sintetizable que se integra fácilmente en cualquier SoC o FPGA. Diseñado para soportar tanto memorias flash SLC y MLC, es flexible en el uso y fácil en la aplicación. El controlador funciona con cualquier dispositivo adecuado de memoria de hasta 128 Gb de los proveedores líderes de memoria, como Micron, Samsung, Toshiba, Hynix, y otros. El controlador ofrece Código Hamming (1 bit de corrección de errores y detección de errores de 2 bits). Utiliza un diferencial en la señal en el reloj y en los datos y en cualquier frecuencia de reloj de hasta 200 MHz [12].

MARCO TEORICO

Controlador Xilinx® Spartan™-3 FPGA, presenta una interfaz para memoria NAND Flash que puede ser combinada con otras aplicaciones desarrollada por Xilinx® para diseños en el FPGA de Xilinx de la línea Spartan, dónde se requiera añadir una memoria NAND Flash al diseño de un sistema, puede ser implementada para distintas familias o marcas de memorias NAND Flash como son AMD, Intel StrataFlash, Micron SyncFlash, Samsung Flash, Toshiba Flash y AMD UltraNAND. [13].

Lattice Semiconductor Corporation, presenta una arquitectura de controlador basado en máquinas de estado, en este trabajo se presentan los diagramas de flujo de los distintos comandos aplicados a las memorias NAND Flash, además, considera módulos ECC (Error Correction Code por sus siglas en inglés) en la arquitectura. Sin embargo, está dirigido a una memoria Samsung K9F1G08R0A, de 128 Mega bytes de capacidad de almacenamiento, además, está diseñado para ser utilizado con una sola unidad NAND Flash, y para ser implementado en un FPGA de la familia Lattice® [14].

El controlador NAND Flash de EP501 de Eureka Technology, es compatible con el código de corrección de errores (ECC) que lleva a cabo la corrección de errores singlebit y detección de errores de doble bit. El usuario puede elegir entre 2 tipos de protección ECC: ECC por cada palabra de 32 bits o ECC para todas las páginas de datos. Cuenta con 8 bancos de memoria para contener 5 memorias por cada banco, dando como resultado una capacidad de 32GBytes. Está diseñado para ser implementado en FPGA y ASIC [15].

Tomando en cuenta la bibliografía existente, se determina que se puede realizar un controlador NAND Flash, considerando que el control a desarrollar será solo para un tipo de memoria, ya que si se requiere tener un control de varias marcas y modelos de memoria se debe de realizar un programa más extenso, esto conlleva un bajo desempeño por los tiempos de operaciones para cada memoria.

2.8 Microcontroladores

Los microcontroladores se utilizan en circuitos electrónicos comerciales desde hace años de una forma masiva, debido a que permiten reducir el tamaño y el precio de los equipos. Un ejemplo de estos son los teléfonos móviles, las cámaras de video, la televisión digital, la transmisión por satélite y los hornos de microondas. Diversos fabricantes ofrecen amplias gamas de microcontroladores para todas las necesidades. Pero, sin duda, hoy en día los microcontroladores mas aceptados para diseños son los microcontroladores PIC fabricados por Microchip Technology Inc, que recientemente se anunciaba como el mayor fabricante del mundo de microcontroladores de 8 bits [16].

Los modelos de PIC constituyen gamas distintas, en función del tamaño de instrucción que emplean. Actualmente, Microchip comercializa sus microcontroladores clasificados en cuatro gamas:

La gama baja la componen la serie PIC10 y una parte de las series PIC12 y PIC16. Utilizan palabras de instrucción de 12 bits, su tamaño es reducido, así como sus características, y su coste es muy bajo [17].

La gama media está compuesta por casi toda la serie PIC16 y una porción de los PIC12. Utilizan un ancho de palabra de instrucción de 14 bits. Ésta es la gama más popular por su buena relación calidad/precio. Además, programarlos en lenguaje ensamblador resulta bastante sencillo, dentro de la complejidad del lenguaje, y es por ello que son una buena opción de cara al aprendizaje [17].

La gama alta o de alto rendimiento la forma la serie de microcontroladores PIC18. Emplean palabras de instrucción de 16 bits y están basados en los PIC de gama media pero con mejoras sustanciales: más puertos de E/S, más conversores A/D o interfaces USB [17].

MARCO TEORICO

La gama de 24 bits la componen de las series dsPIC30 y PIC24. Utilizan 24 bits como palabra de instrucción, usan palabras de memoria de datos de 16 bits (y no 8 bits), y son los que ofrecen más memoria y mayor rendimiento [17].

2.9 LabVIEW™

LabVIEW™ (Laboratory Virtual Instrument Engineering Workbench) es un lenguaje de programación gráfico para el diseño de sistemas de adquisición de datos, instrumentación y control. Labview™ permite diseñar interfaces de usuario mediante una consola interactiva basado en software. Labview™ es a la vez compatible con herramientas de desarrollo similares y puede trabajar con programas de otra área de aplicación, como por ejemplo Matlab. Tiene la ventaja de que permite una fácil integración con hardware, específicamente con tarjetas de medición, adquisición y procesamiento de datos (incluyendo adquisición de imágenes). Los Programas en LabVIEW™ son llamados instrumentos virtuales (VIs) [18].

Un instrumento virtual es un módulo de software que simula el panel frontal de instrumento común y, apoyándose en elementos de hardware accesibles por el PC (tarjetas de adquisición, tarjetas DSP, instrumentos accesibles vía GPIB, VXI, RS-232), realiza una serie de medidas como si se tratase de un instrumento real [18].

LabVIEW™ constituye un revolucionario sistema de programación gráfica para aplicaciones que involucren adquisición, control, análisis y presentación de datos. Las ventajas que proporciona el empleo de LabVIEW™ se resumen en las siguientes:

- Se reduce el tiempo de desarrollo de las aplicaciones al menos de 4 a 10 veces, ya que es muy intuitivo y fácil de aprender.
- Dota de gran flexibilidad al sistema, permitiendo cambios y actualizaciones tanto del hardware como del software.
- Da la posibilidad a los usuarios de crear soluciones completas y complejas.

MARCO TEORICO

- Con un único sistema de desarrollo se integran las funciones de adquisición, análisis y presentación de datos.
- El sistema está dotado de un compilador gráfico para lograr la máxima velocidad de ejecución posible.
- Tiene la posibilidad de incorporar aplicaciones escritas en otros lenguajes.

LabVIEW™ es un entorno de programación destinado al desarrollo de aplicaciones, similar a los sistemas de desarrollo comerciales que utilizan el lenguaje C o BASIC. Sin embargo, LabVIEW™ se diferencia de dichos programas en un importante aspecto: los citados lenguajes de programación se basan en líneas de texto para crear el código fuente del programa, mientras que LabVIEW™ emplea la programación gráfica o lenguaje G para crear programas basados en diagramas de bloques [18].

Para el empleo de LabVIEW™ no se requiere gran experiencia en programación, ya que se emplean iconos, términos e ideas familiares a científicos e ingenieros, y se apoya sobre símbolos gráficos en lugar de lenguaje escrito para construir las aplicaciones. Por ello resulta mucho más intuitivo que el resto de lenguajes de programación convencionales.

LabVIEW™ posee extensas librerías de funciones y subrutinas. Además de las funciones básicas de todo lenguaje de programación, LabVIEW™ incluye librerías específicas para la adquisición de datos, control de instrumentación VXI, GPIB y comunicación serie, análisis presentación y guardado de datos [18].

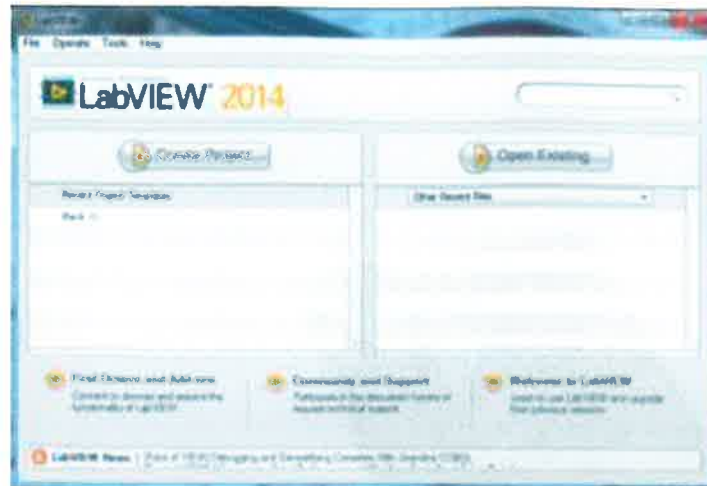


Figura 2.4 Ventana principal de LabVIEW™.

2.10 Osciloscopio

Desarrollar un controlador, en donde se requiere tener una visualización completa de las diferentes señales que se emplean para el control de la memoria, requiere tener una herramienta que permita observar el tiempo en que se ejecutan cada una de las señales, para realizar esta visualización se hace uso de un osciloscopio el cual pueda muestrear cada una de las diferentes señales generadas.

El osciloscopio Analog Discovery que ofrece Analog Devices y Digilen, realiza diferentes tareas que facilitan la adquisición de señales que se requiere para el desarrollo del controlador NAND Flash. La visualización se realiza mediante el programa WaveForms™ en donde el software muestra una interfaz gráfica fácil de manejar. El osciloscopio está integrado por [19]:

- 2 Canales de osciloscopio
- 2 Canales Generadores de forma de onda
- 16 Canales Analizadores lógicos
- 16 Canales Generador de Patrón Digital
- Fuente de alimentación de $\pm 5VDC$
- Analizador de espectro

MARCO TEORICO

- Analizador de Network
- Voltímetro
- E/S digitales



Figura 2.5 Osciloscopio Analog Discovery [19].

3 DESARROLLO DE PROYECTO

En este capítulo se lleva a cabo el desarrollo del controlador NAND Flash, mostrando la descripción del microcontrolador que se emplea para el control de la memoria, se presenta también los detalles de la memoria para realizar los tiempos de ejecución para cada comando que son necesarios para su correcto funcionamiento.

3.1 Arquitectura de microcontrolador

Para el desarrollo de este proyecto se utilizó un microcontrolador, este dispositivo fue seleccionado tras haber realizado comparaciones de velocidad de procesamiento, arquitectura de programación, entre otros, se muestra a continuación una tabla de comparación de los dispositivos de la familia de los PIC18LF4550 de Microchip y de la familia de los MSP430F261X de Texas Instruments.

Tabla 3.1 Comparación de un microcontrolador de la marca Microchip y Texas Instruments [20], [21].

Características	PIC18LF4550	MSP430F261x
Voltaje de alimentación	2.0 - 5.5 V	1.8 - 3.6 V
Numero de pines	40	64
Encapsulado	DIP	SMD
frecuencia de trabajo	Hasta 48 MHz	Hasta 16 MHz
Consumo en modo espera	0.1uA	0.5uA
Consumo en modo activo	5.8uA	365uA
Memoria RAM	2Kb	4Kb
Módulo UART	Si	Si

Para el desarrollo de este proyecto se ha elegido el microcontrolador de Microchip, debido a la frecuencia de trabajo, tipo de encapsulado y a su fácil programación.

3.2 Memoria NAND Flash

Para realizar el almacenamiento se ha optado por el uso de la memoria HY27UF084G2B de la marca Hynix, la memoria se comprende de las siguientes características.

La memoria tiene una capacidad de 512Mx8bits (donde M se refiere al prefijo Mega) y cuenta con 16Mx8bits de repuesto. La alimentación de la memoria es de 3.3 Vcc., cuenta con 8 I/O. La memoria se divide en bloques que se pueden borrar de forma independiente por lo que es posible preservar datos válidos, mientras que los datos indeseables se borran. Este dispositivo contiene 4096 bloques, compuesto de 64 páginas [8].

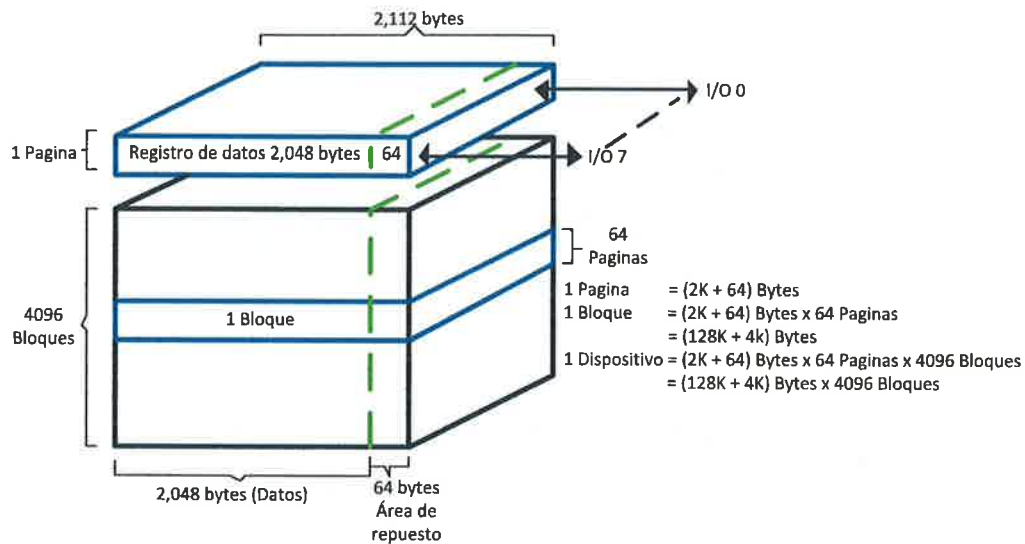


Figura 3.1 Ejemplo de arquitectura de una memoria NAND Flash de 4Gbits [22]

Para realizar la ejecución de las operaciones de la memoria NAND Flash es muy importante utilizar un controlador que genere de forma precisa los tiempos de las señales. Las funciones básicas para obtener el control de la memoria son, entrada de comando, entrada de dirección, escritura de datos, lectura de datos, protección de escritura y modo de espera.

3.3 Funciones

Las operaciones que son empleadas para el control de la memoria utilizan tiempos en los que son necesarios respetar la duración de ejecución para cada señal. En la tabla 3.2 se presenta los parámetros máximos y mínimos para realizar la función comando.

Tabla 3.2 Parámetros de tiempos Máximos y Mínimos para el envío de la función comando [8].

Parámetro	Min	Max	Unidad
tCLS	15	-	ns
tCLH	10	-	ns
tCS	25	-	ns
tCH	10	-	ns
tWP	17	-	ns
tALS	15	-	ns
tALH	10	-	ns
tDS	15	-	ns
tDH	5	-	ns

3.3.1 Función Comando

En la figura 3.2 se muestra la señal de la función comando que se utiliza para dar orden al dispositivo de memoria. El función comando es aceptado cuando CE está en bajo, CLE en alto, ALE en bajo, RE en alto y WE en flanco de subida. Los comandos que inicien una operación de modificación (escritura o borrado) deben de poner WP en flanco de subida [8].

DESARROLLO DE PROYECTO

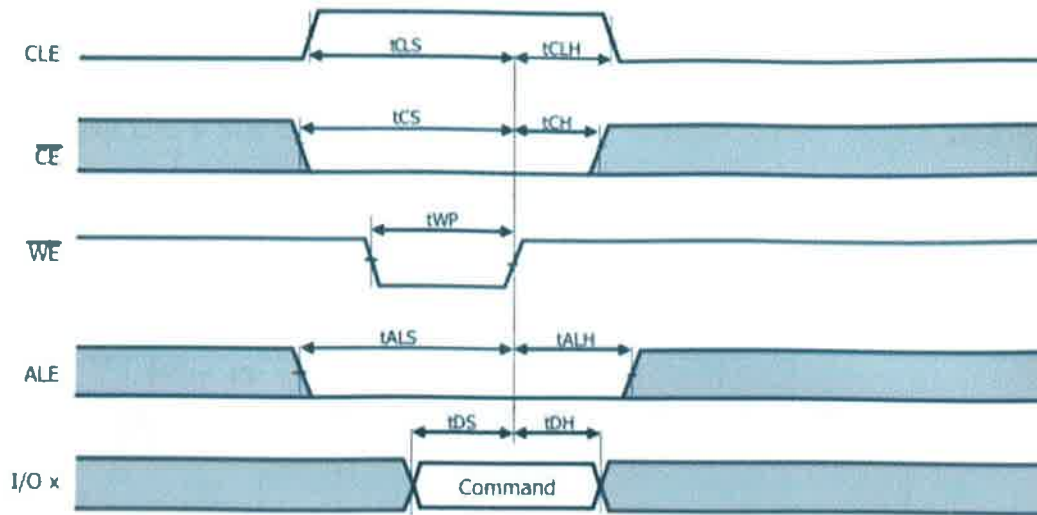


Figura 3.2 Función comando [8].

3.3.2 Función Dirección

Realizar algún cambio del almacenamiento de datos como escribir o borrar, requiere un direccionamiento, para realizar cualquiera de las acciones antes mencionada, al igual que en la función comando, requiere de tiempos para el correcto funcionamiento. La operación lectura requiere de un direccionamiento similar al de escritura utilizando 5 bytes para el direccionamiento, el borrado de la memoria se comprende de 3 bytes para su direccionamiento, ya que este se direcciona por bloques y no por páginas.

Tabla 3.3 Parámetros de tiempos Máximos y Mínimos para el envío de comandos [8].

Parametro	Min	Max	Unidad
tCLS	15	-	ns
tCS	25	-	ns
tWC	35	-	ns
tWP	17	-	ns
tALS	15	-	ns
tDS	15	-	ns
tWH	15	-	ns
tDH	5	-	ns

DESARROLLO DE PROYECTO

La función Dirección permite la inserción de la dirección de memoria utilizando cinco ciclos. La función dirección es aceptada cuando CE está en bajo, ALE está en alto, CLE en bajo, RE en alto y WE en alto. Además que si se requiere realizar alguna modificación (escritura o borrado) WP debe de estar en alto [8].

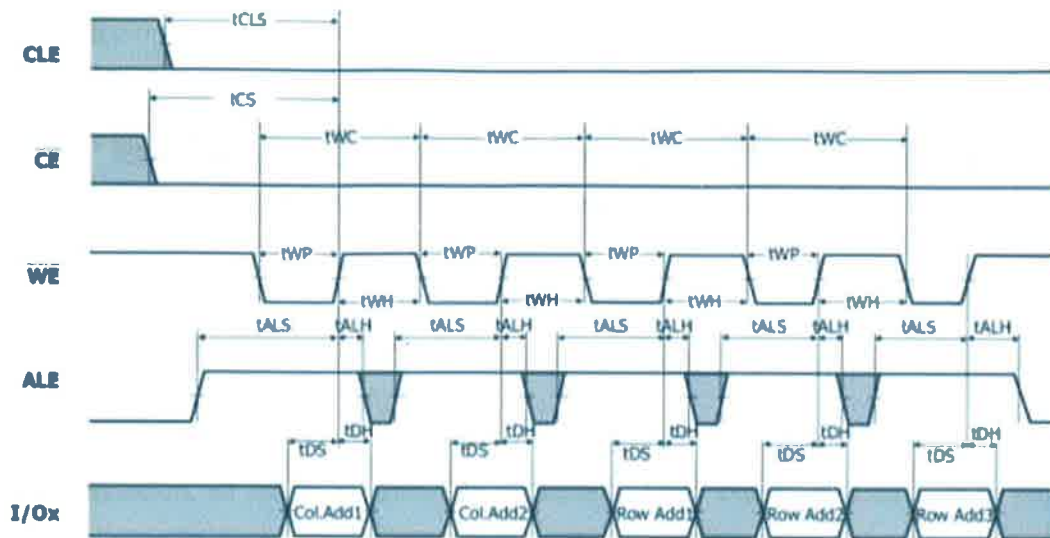


Figura 3.3 Función dirección [8].

3.3.3 Función escritura de datos

La función escritura de datos, escribe los datos suministrados a una página que han sido borrados por adelantado. En su fase inicial, los datos que se escriben en la página de destino se transfieren a través del bus de flash a un búfer de página interna en el chip flash, y luego se envía un comando de programa, junto con la dirección de la página de destino. A continuación, el chip se convierte ocupado durante el periodo de la latencia [23].

La función escritura de datos permite la entrada de datos al dispositivo para ser programado. La entrada de datos es en serie y es cronometrado mediante la habilitación de WE. La función entrada de datos es aceptada cuando CE, ALE, CLE están en estado bajo, RE, WP y WE en flanco ascendente [8].

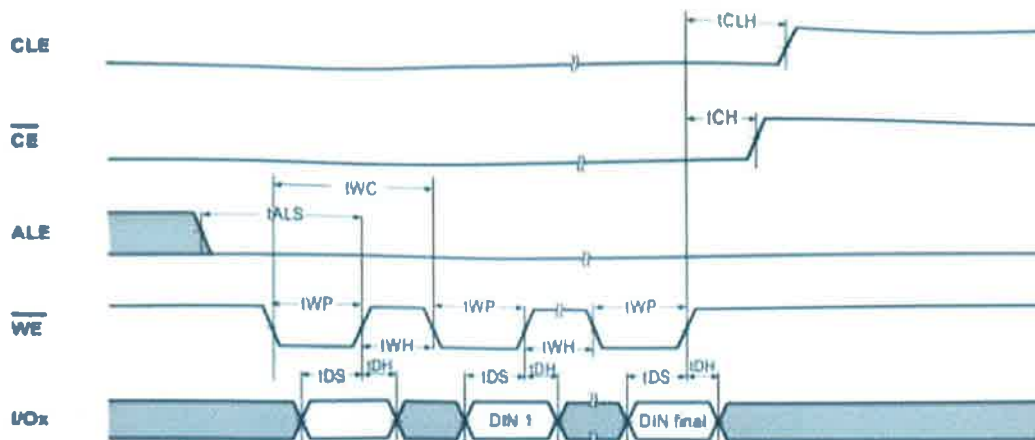


Figura 3.4 Función escritura de datos [8].

3.3.4 Función lectura de datos

La operación lectura de datos lee los datos de una página de la memoria flash. Durante la fase de iniciación, se envía un comando de lectura, junto con la dirección de la página. Cuando la latencia de lectura ha terminado y el chip se convierte de nuevo en listo. La memoria flash está sujeta a errores de bits, en el que uno o más bits en una página se invierten entre la programación y la lectura de una página. Estos pueden ser contrarrestados hasta cierto punto por la lógica de corrección de errores en el hardware o software. Los fabricantes de memoria flash NAND también permiten que los chips cuenten con un número limitado de bloques defectuosos con el fin de mejorar el rendimiento [23].

La función lectura de datos debe de cumplir con los tiempos especificado en la tabla siguiente

DESARROLLO DE PROYECTO

Tabla 3.4 Parámetros de tiempos Máximos y Mínimos para el envío de comandos [8].

Parametro	Min	Max	Unidad
tRC	35	-	ns
tREA	-	25	ns
tREH	15	-	ns
tRR	20	-	ns
tCHZ	-	50	ns
tCOH	15	-	ns
tRHOH	15	-	ns

La función de lectura de datos permite leer los datos de la matriz de memoria y comprueba el contenido del registro de estado, el contenido del registro de la EDC y los datos de identificación. Los datos son desplazados fuera en serie alternando el flanco del pin RE, CE debe estar en bajo, WE en alto, ALE y CLE en bajo [8].

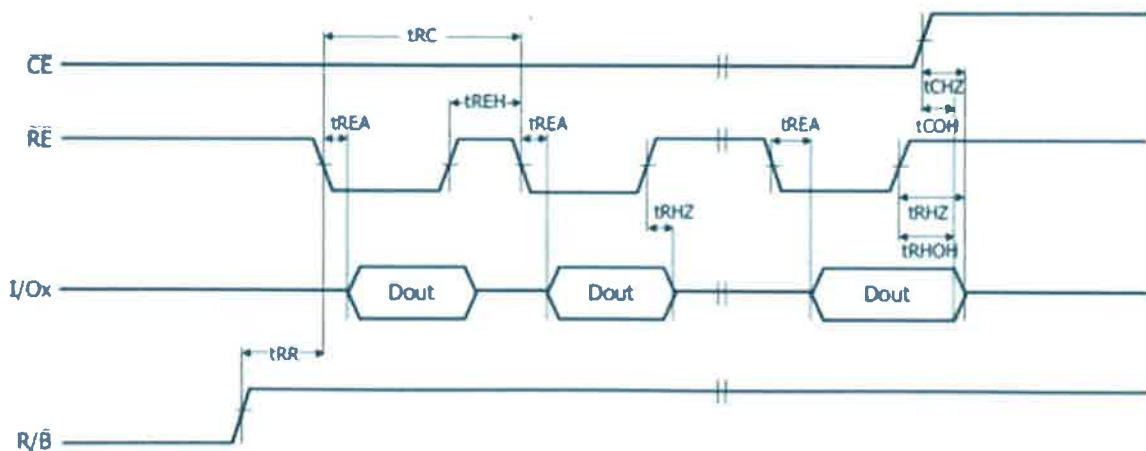


Figura 3.5 Función lectura de datos [8].

3.3.5 Función protección de escritura

La protección contra escritura se efectúa cuando el pin WP está en estado bajo. En esta condición las operaciones de modificación no se efectúan y el contenido de la memoria no

DESARROLLO DE PROYECTO

se altera, el pin WP no está enlazado con WE para garantizar la protección incluso durante el encendido [8].

3.3.6 Función modo espera

El modo en espera se realiza cuando el dispositivo está deshabilitado, las salidas están desactivadas y el consumo de energía se reduce [8]. Y se consigue al establecer un 0 lógico en los pines CLE, ALE, WP y se establecen a 1 los pines CE, WE, RE.

3.3.7 Función de lectura de estado

Para las funciones que se realizan en la memoria, se pueden verificar el estado en el que se encuentra la memoria, esto se puede hacer mediante la función estado. El dispositivo contiene un registro de estado que se puede leer para saber si las operaciones de programación o borrado se han completado correctamente. Después de escribir el comando 70h en el registro del sistema, un ciclo de lectura emite el contenido del registro de estado de los pines de E/S en el flanco descendente de CE o RE [8].

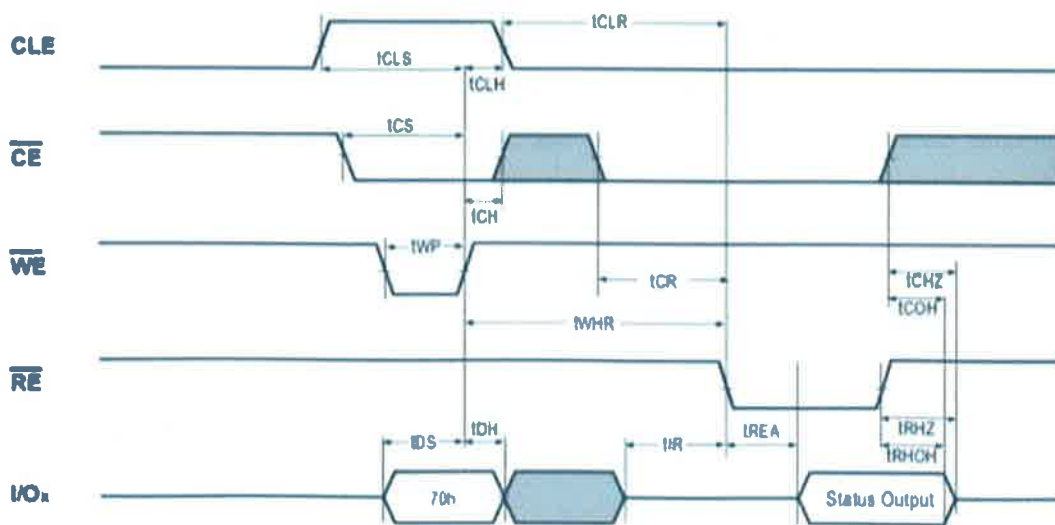


Figura 3.6 Función de lectura de estado [8].

DESARROLLO DE PROYECTO

Tabla 3.5 Definición del Registro de Estado [8].

Bit	Programar pagina	Borrar bloque	Leer	Leer cache	Código
0	Paso/Fallo	Paso/Fallo	NA	NA	Paso=0, Fallo=1
1	NA	NA	NA	NA	-
2	NA	NA	NA	NA	-
3	NA	NA	NA	NA	-
4	NA	NA	NA	NA	-
5	Listo/Ocupado	Listo/Ocupado	Listo/Ocupado	P/E/R Control de Bit	Activo=0/Inactivo=1
6	Listo/Ocupado	Listo/Ocupado	Listo/Ocupado	Listo/Ocupado	Listo/Ocupado
7	Protección de escritura	Protección de escritura	Protección de escritura	NA	Protegido=0 No protegido=1

3.4 Tareas para la memoria NAND Flash

Uniendo las funciones comando, dirección, escritura, lectura y estado se puede realizar las distintas tareas permitidas para la gestión de datos de la memoria, como son, la lectura de datos, escritura de datos, borrado de bloques, la lectura de ID de la memoria

3.4.1 Lectura de página

La operación lectura de página se inicia enviando el comando 00h, seguido de la función dirección, al terminar el ciclo de dirección se envía el comando 30h, la tarea lectura de página termina al enviar la función lectura. La función de dirección debe de contener la ubicación de la página y bloque que se va a leer [8].

DESARROLLO DE PROYECTO

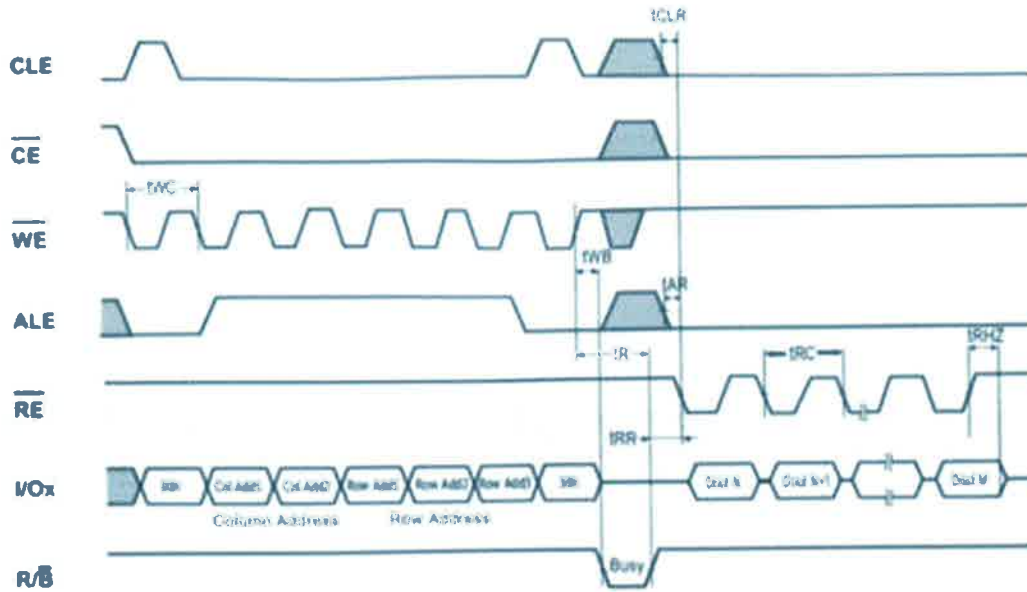


Figura 3.7 Lectura de Página [8].

3.4.2 Programación de página

El dispositivo es programado por páginas, el direccionamiento debe de hacerse por orden secuencial en un bloque. Un ciclo de programa de página consiste en un período de carga de datos en serie en el que hasta 2112 bytes (dispositivo X8) de datos puede ser cargado en el registro de datos, seguido por un período de programación no volátil, donde los datos cargados se programa en la celda apropiada [8].

El periodo de escritura de datos comienza cuando el comando 80h es enviado seguido de la función dirección, para que los datos sean escritos se envía la función escritura de datos, la escritura de datos inicia el proceso cuando se confirma al enviar el comando 10h. El controlador del sistema puede detectar la finalización de un ciclo de programa mediante la salida de R/B [8].

DESARROLLO DE PROYECTO

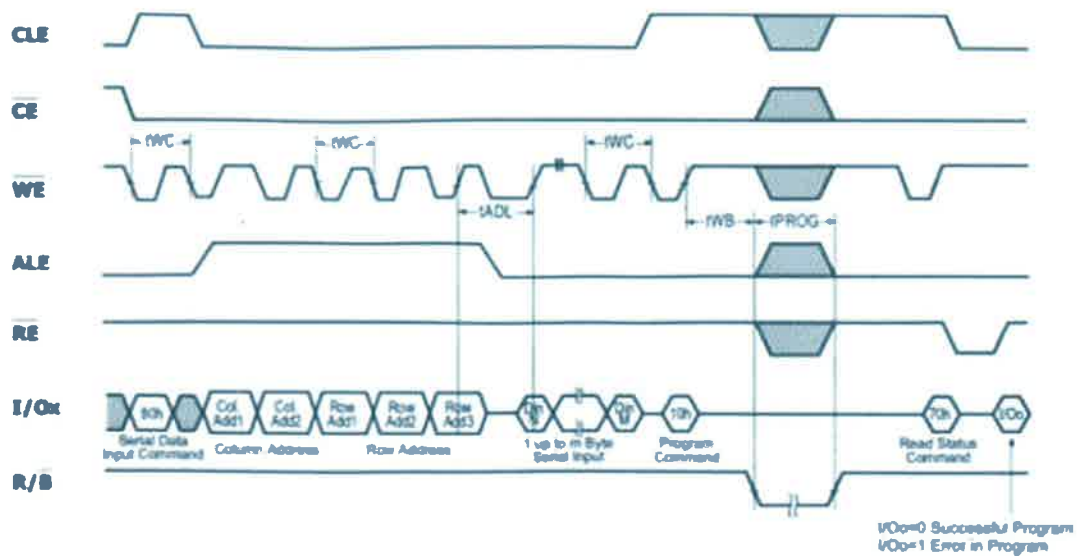


Figura 3.8 Programación de Página [8].

3.4.3 Borrado de bloque

La tarea de borrado se realiza sobre un bloque. La dirección del bloque se lleva a cabo en tres ciclos después de la función comando 60h. El comando borrado confirma con D0h la carga de dirección de bloque iniciando después el proceso de borrado interna. Esta secuencia de dos pasos de configuración seguida de comando de ejecución garantiza que el contenido de la memoria no se borra accidentalmente debido a las condiciones de ruido externo. El controlador del sistema puede detectar la terminación de un borrado mediante la supervisión de la salida R/B y puede ser confirmado su éxito de borrado mediante la función lectura de estado [8].

DESARROLLO DE PROYECTO

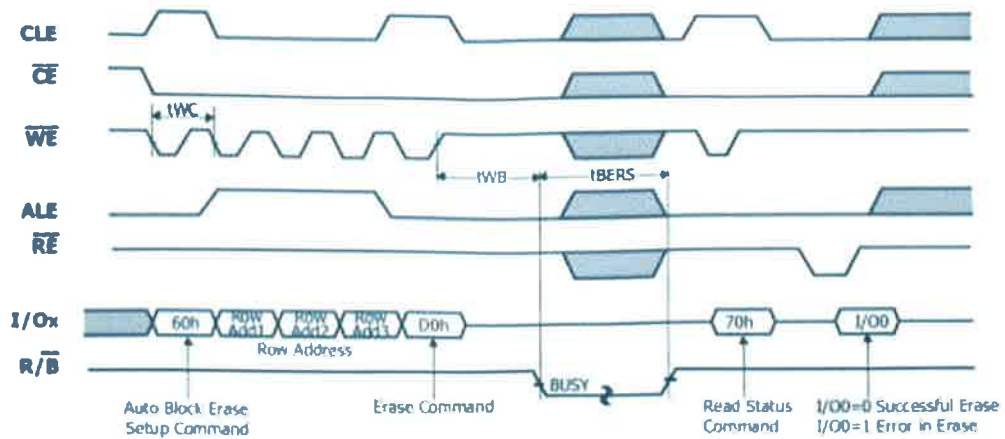


Figura 3.9 Borrado de un bloque [8].

3.5 Tarea de lectura de ID

La lectura de ID, es el modo de identificación del producto, esta se obtiene enviando el comando 90h, seguido de la dirección 00h. Cuatro ciclos de lectura de datos son necesarios para obtener el ID de la memoria, en donde el primer ciclo contiene el código de fabricante, el segundo ciclo la identificación del dispositivo, el tercer y cuarto ciclo el ID [8].

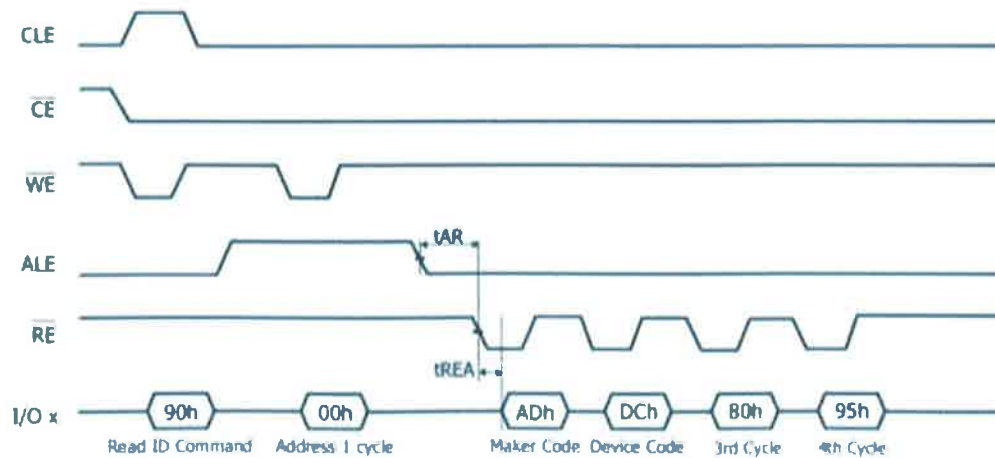


Figura 3.10 Lectura de ID [8].

3.6 Puntero de Dirección

Realizar la lectura, escritura o borrado de la memoria, requiere que sean direccionadas mediante la función dirección. La función dirección lleva a la función comando al área correspondiente a la cual se debe de ejecutar. A continuación se muestra un diagrama de flujo que realiza la acción del puntero que estará inmerso en la función dirección.

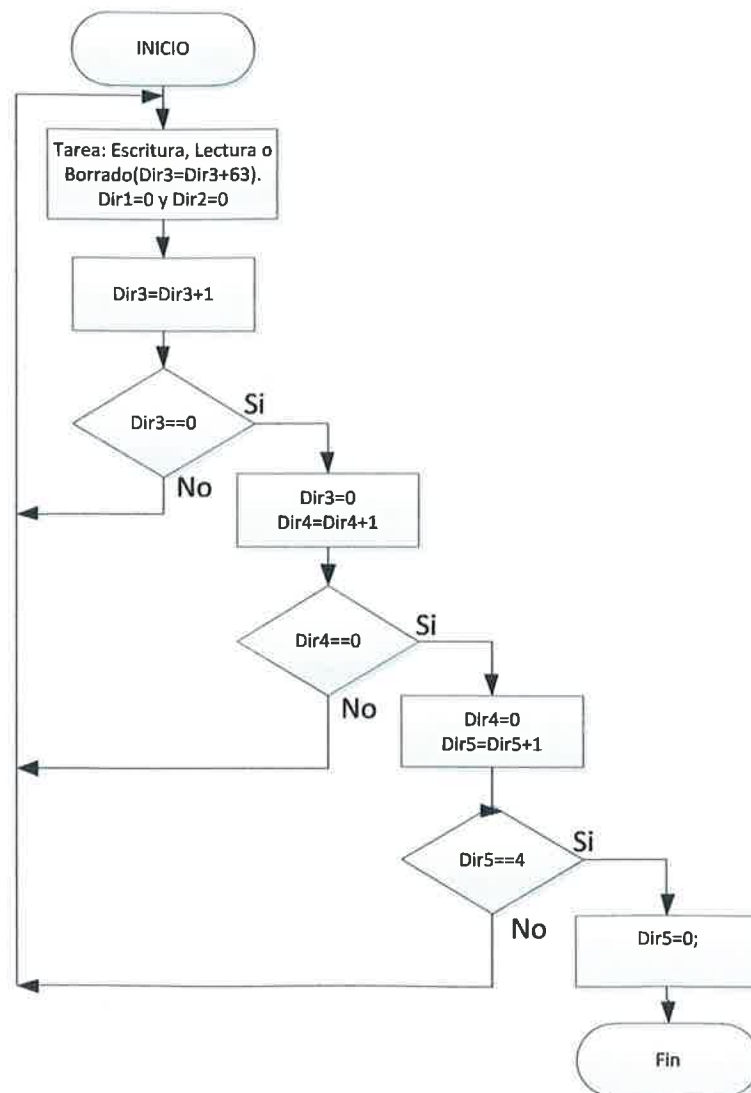


Figura 3.11 Diagrama de flujo de Puntero de Dirección.

DESARROLLO DE PROYECTO

La tabla 3.6 muestra la posición correspondiente para direccionar los comandos para que las tareas antes mencionadas sean ejecutadas.

Tabla 3.6 Mapa de ciclos de dirección

	I/O 0	I/O 1	I/O 2	I/O 3	I/O 4	I/O 5	I/O 6	I/O 7
Dir1	A0	A1	A2	A3	A4	A5	A6	A7
Dir2	A8	A9	A10	A11	L(1)	L(1)	L(1)	L(1)
Dir3	A12	A13	A14	A15	A16	A17	A18	A19
Dir4	A20	A21	A22	A23	A24	A25	A26	A27
Dir5	A28	A29	L(1)	L(1)	L(1)	L(1)	L(1)	L(1)

En donde A0 hasta A11 se encuentran en Dir1 y Dir2 utilizados para direccionar a cada byte, teniendo un direccionamiento de 2048 bytes en cada página. En el diagrama de flujo Dir1 y Dir2 se inicializan con 0 debido a que los comandos empleados para la lectura y escritura hacen que la gestión de datos sea secuencial.

A12 hasta A17 se utilizan para direccionar las 64 páginas de cada bloque y en el diagrama de flujo corresponde a Dir3. Para direccionar los 4096 bloques se utilizan A18 hasta A29 y en el diagrama de flujo corresponde a Dir4 y Dir5.

Cada tarea realizada a excepción de la lectura de ID necesita direccionar cada página y cada bloque, esta dirección se escribe en la función dirección. La tarea lectura y escritura de página debe realizar su direccionamiento secuencialmente. A comparación de las tareas anteriores, la tarea de borrado se realiza por bloques, cada bloque se compone de 64 páginas, es por eso que la tarea borrado se inicializa con un valor de 63, dando como resultado una ejecución en menos tiempo a comparación de las dos tareas mencionadas anteriormente.

4 EXPERIMENTACIÓN Y RESULTADOS

En este capítulo se muestra la validación para el controlador de memoria NAND Flash, en donde se muestran las conexiones del microcontrolador a la memoria, comparaciones de graficas en las tareas, tiempos de ejecución en las tareas y pruebas realizadas.

4.1 Conexión de memorias

Se realizaron las pruebas utilizando una tarjeta que cuenta con un arreglo físico de las memorias Hynix HY27UF084G2B, esta memoria fue utilizada anteriormente para el diseño de arquitectura para almacenamiento redundante de información [3]. La tarjeta cuenta con dos marcas de memoria que son Micron y Hynix, las pruebas se realizaron con la segunda marca de memoria debido a la mala sincronización entre la primera memoria y el microcontrolador.

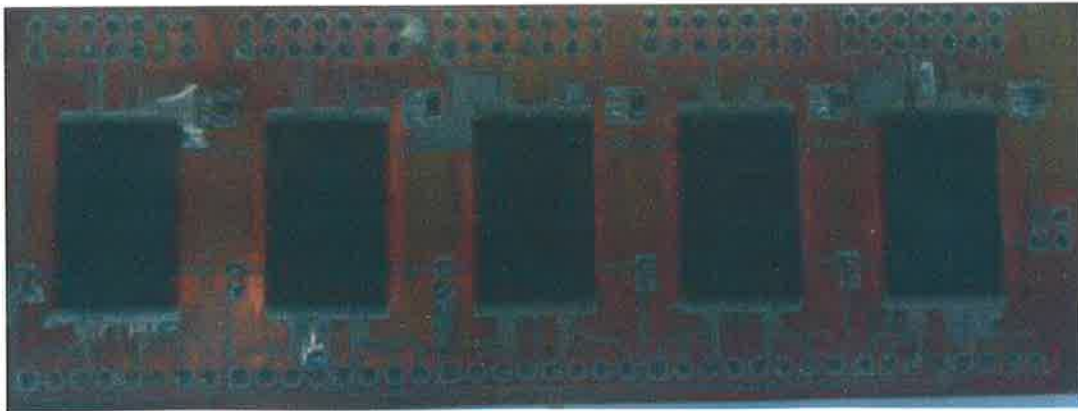


Figura 4.1 Conexión de memorias Hynix.

4.2 Conexión de memoria y el microcontrolador

La figura 4.2 representa la conexión de la memoria y el microcontrolador, en donde se muestra los pines de comunicación a la computadora por puerto serial, la conexión de las I/O en el puerto B, las señales de control en el puerto D y en el puerto C0 se encuentra

EXPERIMENTACION Y RESULTADOS

conectado el pin RB de la memoria, también cuenta con Leds que indican cuando una tarea se realizó, el puntero está desbordado y cuando se está ocupando el puntero.

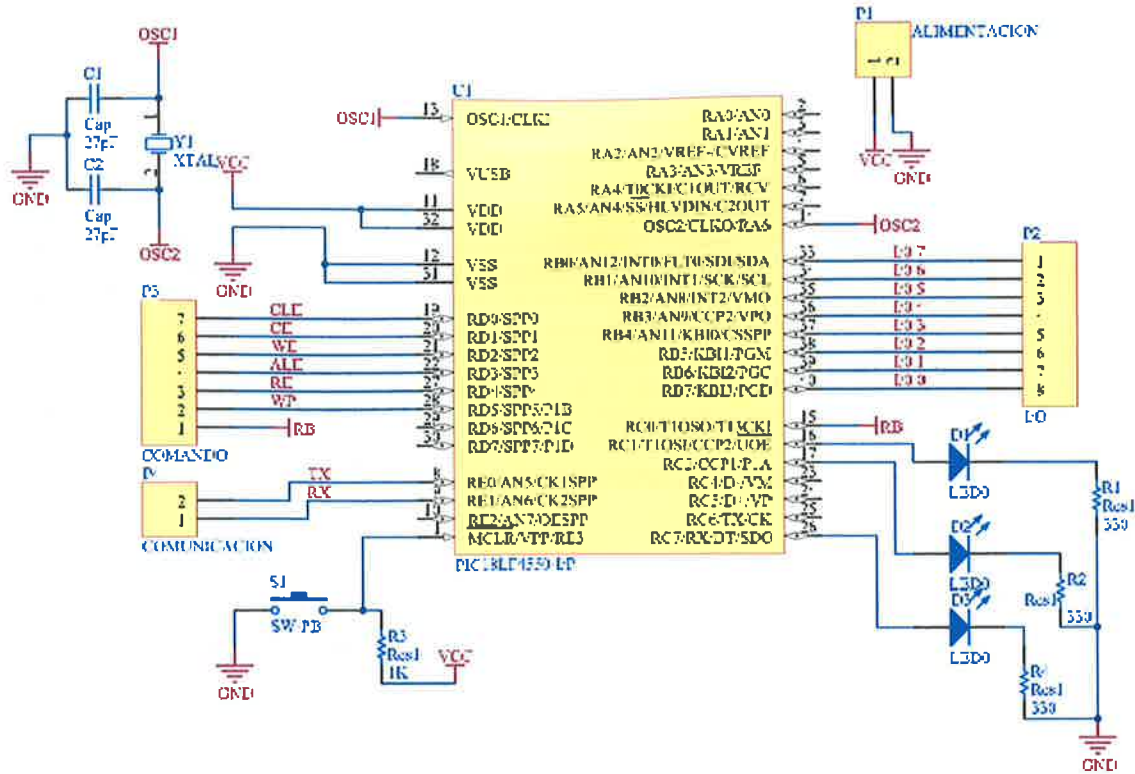


Figura 4.2 Diagrama de conexión del microcontrolador y la memoria.

4.3 Validación de Tareas

Las principales tareas que se validaron son la escritura, lectura y borrado de la memoria.

La figura 4.3 muestra el comportamiento de las señales de la tarea lectura de página que se puede comparar con la figura 3.7 del capítulo 3.

EXPERIMENTACION Y RESULTADOS

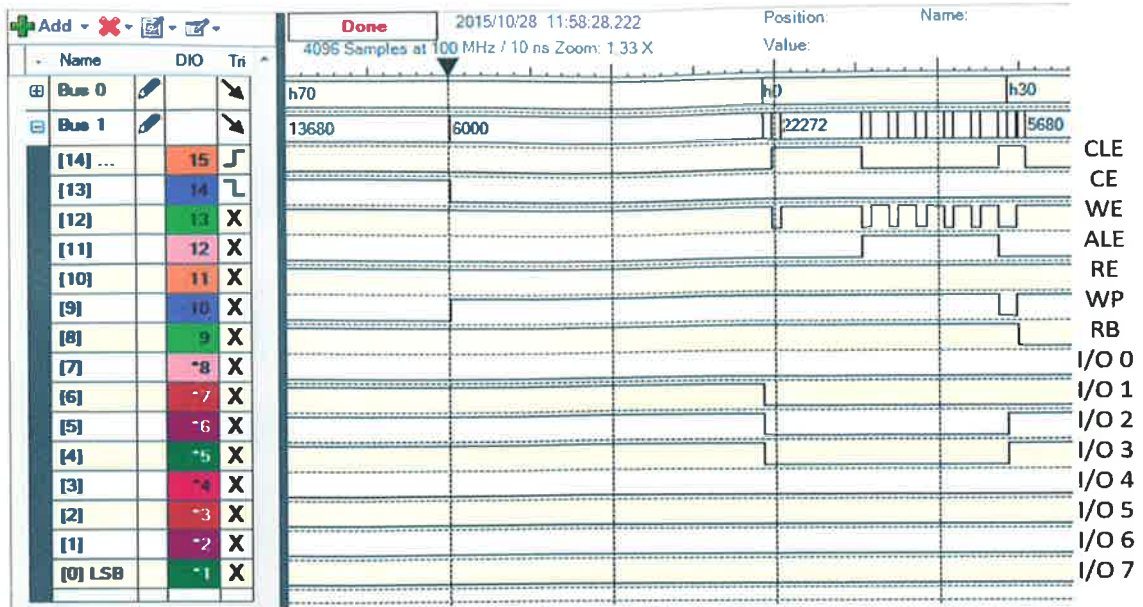


Figura 4.3 Tarea lectura de página emitida por el microcontrolador.

La figura 4.4 muestra el comportamiento de las señales que emite el controlador al escribir una página, esta tarea se puede comparar con la figura 3.8 del capítulo 3. La figura muestra la escritura de un solo valor.

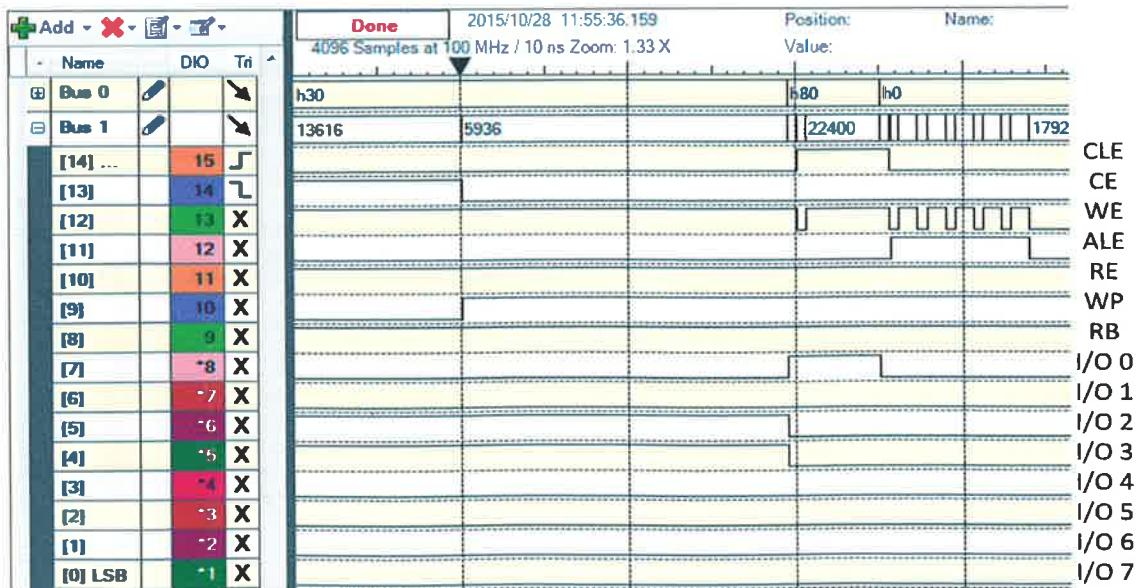


Figura 4.4 Tarea escritura de página emitida por el microcontrolador.

EXPERIMENTACION Y RESULTADOS

La figura 4.5 muestra la tarea borrado de bloque, se puede comparar con la figura 3.9 del capítulo 3, se observa que la función dirección se ejecuta en tres ciclos.

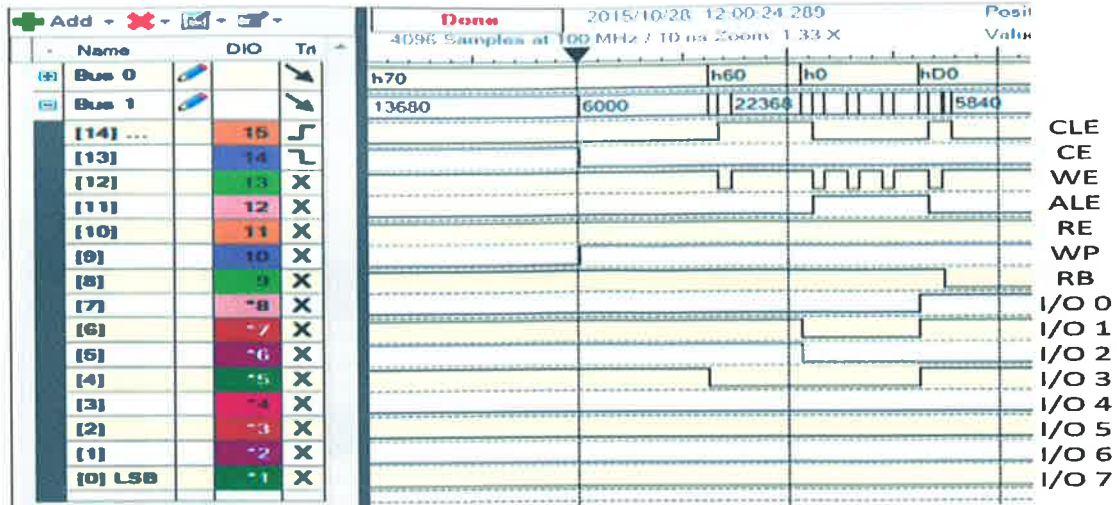


Figura 4.5 Tarea Borrado de bloque emitida por el microcontrolador.

La figura 4.6 muestra la lectura de ID de la memoria, los datos arrojados por la memorias son ADh, DCh, 10h, 99h, se puede comparar con la figura 3.10 del capítulo 3.

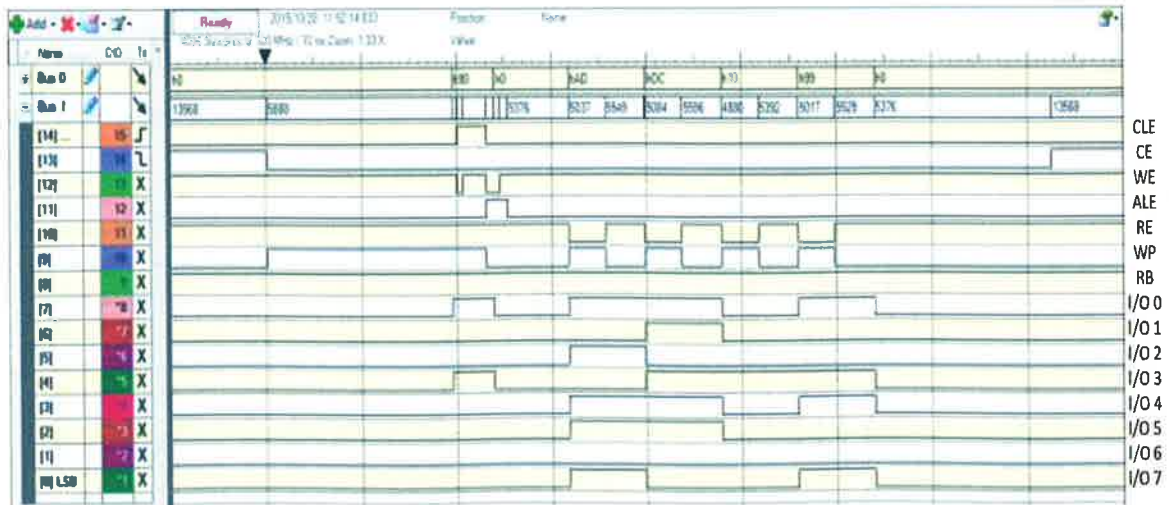


Figura 4.6 Lectura de ID recibida por el microcontrolador.

EXPERIMENTACION Y RESULTADOS

4.4 Pruebas de la tarea lectura.

Se realizaron pruebas de lectura en donde se midió el tiempo en que tarda la tarea lectura incluyendo los datos leídos. Para realizar esta prueba se monitoreo un puerto del microcontrolador, este se activa desde el momento que se llama a la tarea lectura, la figura 4.7 muestra una prueba de la lectura de página.

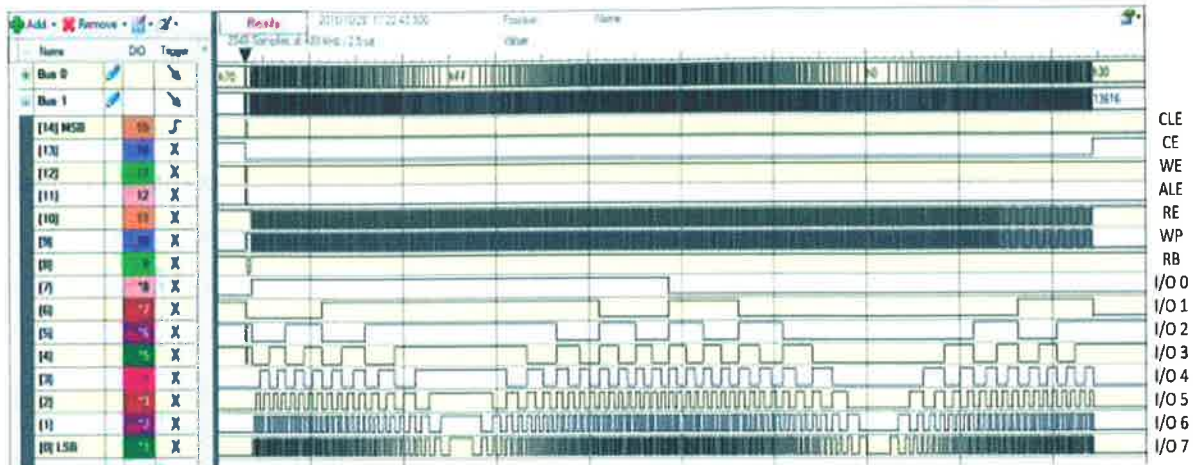


Figura 4.7 Lectura de una señal senoidal de una página.

Se realizaron 40 muestras de esta tarea, a continuación se presenta una gráfica que representa los tiempos en que se ejecuta.

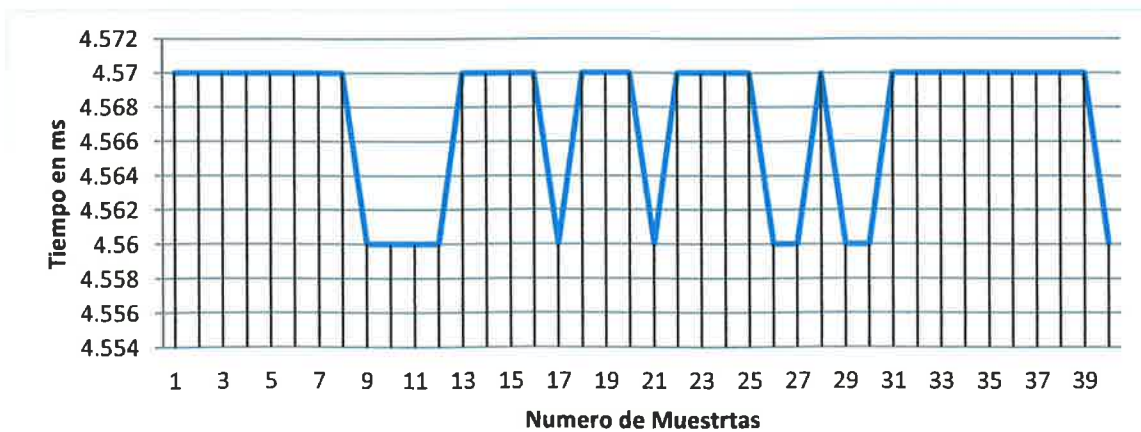


Figura 4.8 Representación gráfica de tiempos en que se ejecuta la tarea lectura de página.

EXPERIMENTACION Y RESULTADOS

El tiempo en que tarda la lectura en KB/s de cada página se obtiene utilizando el tiempo promedio de la lectura que es de $4.56725\text{ms} = 0.00456725\text{seg}$.

$$\frac{2\text{KB}}{0.00456725\text{ s}} = 437.9002682\text{ KB/s}$$

4.5 Pruebas de la tarea escritura

En esta prueba se muestra la escritura de una página que consiste en escribir un dato, es decir, el microcontrolador se ha programado para que escriba automáticamente un valor en la memoria cuando se ejecute la orden de escribir. La figura 4.9 muestra la escritura de un solo valor en la memoria.

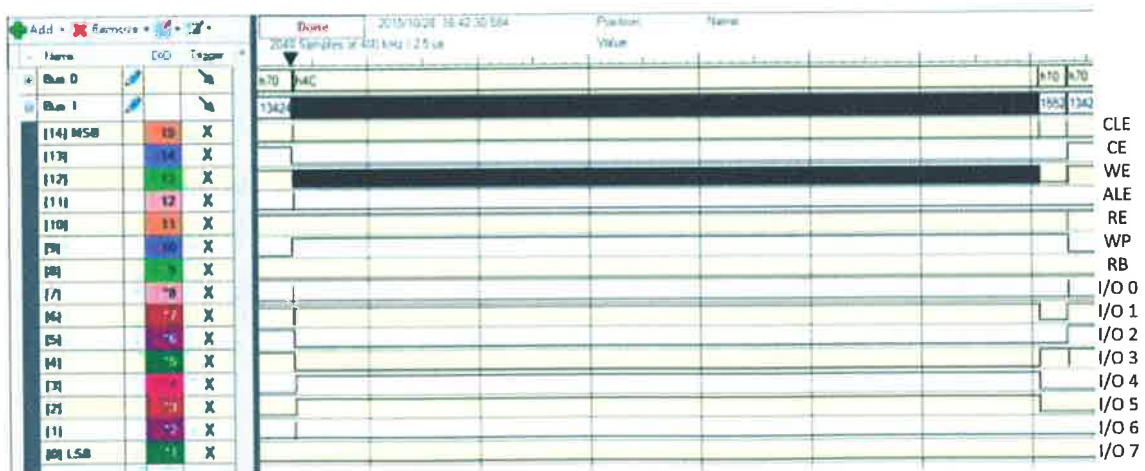


Figura 4.9 Escritura de un mismo dato en una página de la memoria.

Se realizaron pruebas en donde se midió el tiempo en que tarda la tarea escritura de página. La figura 4.10 representa los tiempos realizados en 40 pruebas.

EXPERIMENTACION Y RESULTADOS



Figura 4.10 Representación gráfica de tiempos en que se ejecuta la tarea escritura de página.

El tiempo en que tarda la escritura en KB/s de cada página se obtiene utilizando el tiempo promedio de la escritura que es de $4.04225\text{ms} = 0.00404225\text{seg}$.

$$\frac{2\text{KB}}{0.00404225\text{ s}} = 494.77395\text{ KB/s}$$

Para realizar un estudio más detallado en la escritura de la memoria se realizaron pruebas en las que se envió información al microcontrolador y este a la memoria. En la figura 4.11 se aprecia que la adquisición de datos por parte del microcontrolador requiere de un tiempo mayor que el que se usa para escribir todos los datos adquiridos en una página de la memoria.



Figura 4.11 Escritura de una señal senoidal en una página.

EXPERIMENTACION Y RESULTADOS

La grafica que se presenta en la figura 4.12 muestra 40 pruebas realizadas para saber el tiempo en que se tarda la adquisición de datos y la escritura de una página.



Figura 4.12 Representación gráfica de tiempos en que se ejecuta la tarea escritura de página con adquisición de datos por el microcontrolador.

El tiempo en que tarda la escritura tomando en cuenta la lectura de datos en KB/s de cada página se obtiene utilizando el tiempo promedio de la escritura que es de 2.043875 seg.

$$\frac{2KB}{2.043875 \text{ seg}} = 0.97853342 \text{ KB/s}$$

4.6 Prueba de borrado de memoria

Como se ha mencionado anteriormente, el borrado se realiza por bloques, la cantidad de bloques que se encuentran en la memoria conforman un total de 4096, esta operación se realiza en una sola acción y se aplica en todos los bloques, es decir que la tarea borrado se debe de realizar automáticamente desde el bloque con la dirección 0 hasta el bloque con la dirección 4095 (ver figura 4.13) y no como las tareas anteriores que se espera a que se termine de realizar la acción en cada página y esperan a ser llamadas nuevamente.

EXPERIMENTACION Y RESULTADOS

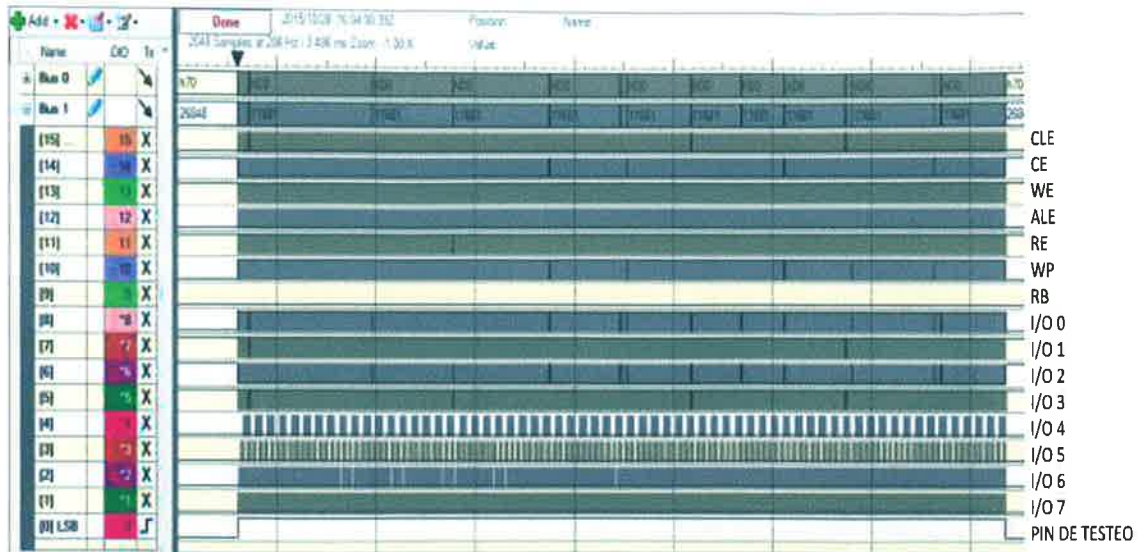


Figura 4.13 Borrado de toda la memoria.

Realizar la escritura y lectura de cada página de toda la memoria requiere de mucho más tiempo que la operación borrado de memoria. La figura 4.14 muestra una gráfica de 40 pruebas del tiempo de borrado de toda la memoria. El tiempo promedio de borrado de toda la memorias es de 5.95025 seg.



Figura 4.14 Representación gráfica de tiempos en que se ejecuta el borrado de toda la memoria.

EXPERIMENTACION Y RESULTADOS

4.7 Interfaz gráfica

Para realizar las pruebas anteriores, se utilizó el software LabVIEW™, se realizó un VI que adquiere los valores del ID de la memoria, ejecuta la lectura de 2000 datos, el borrado de toda la memoria, la escritura de una señal senoidal con 2000 muestras para ser enviadas, y el reinicio del puntero de la memoria. En la figura 4.13 se muestra la interfaz gráfica de este VI.

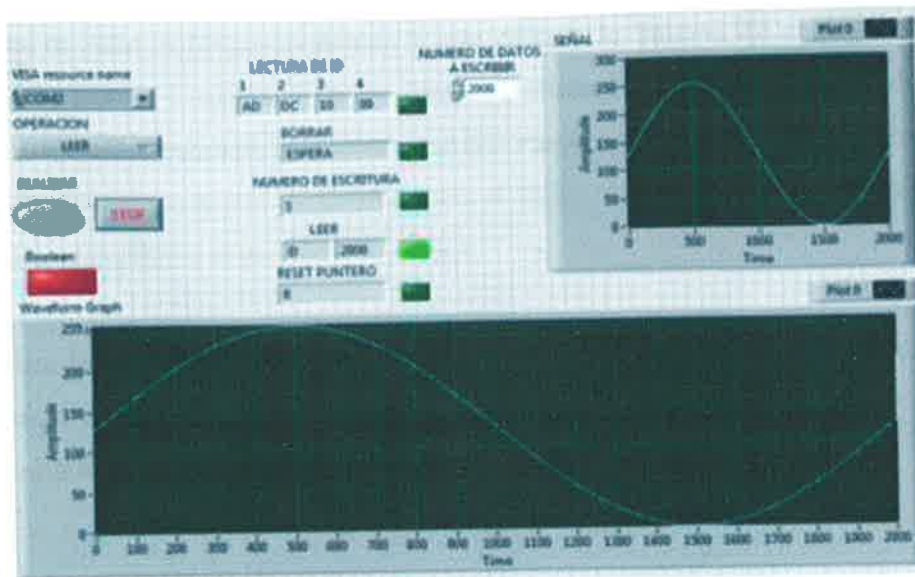


Figura 4.15 Interfaz gráfica en LabView™

CONCLUSIONES Y RECOMENDACIONES

5 CONCLUSIONES Y RECOMENDACIONES

Debido a los resultados de las pruebas realizadas se concluye que si es posible realizar un controlador para memorias NAND flash con un microcontrolador, este controlador puede reducir en gran parte el tiempo de procesos que los controladores maestros utilizan para el almacenamiento de información.

Se recomienda emplear un ECC, que es un requerimiento para un controlador de memorias, el ECC es de gran importancia debido a que cumple la función de corregir errores que surgen en la lectura o escritura de los datos. Realizando la implementación del ECC se puede aprovechar al máximo la memoria NAND Flash, esto se hace utilizando los espacios de área de repuesto que son dedicados para el ECC. Además que se podrá aplicar este código para cada página al gestionar los datos.

Para realizar la implementación del ECC se necesita un microcontrolador con una memoria mucho más grande que el empleado en este proyecto, el actual microcontrolador cuenta con una memoria RAM utilizada al 98% según el compilador CCS C compiler.

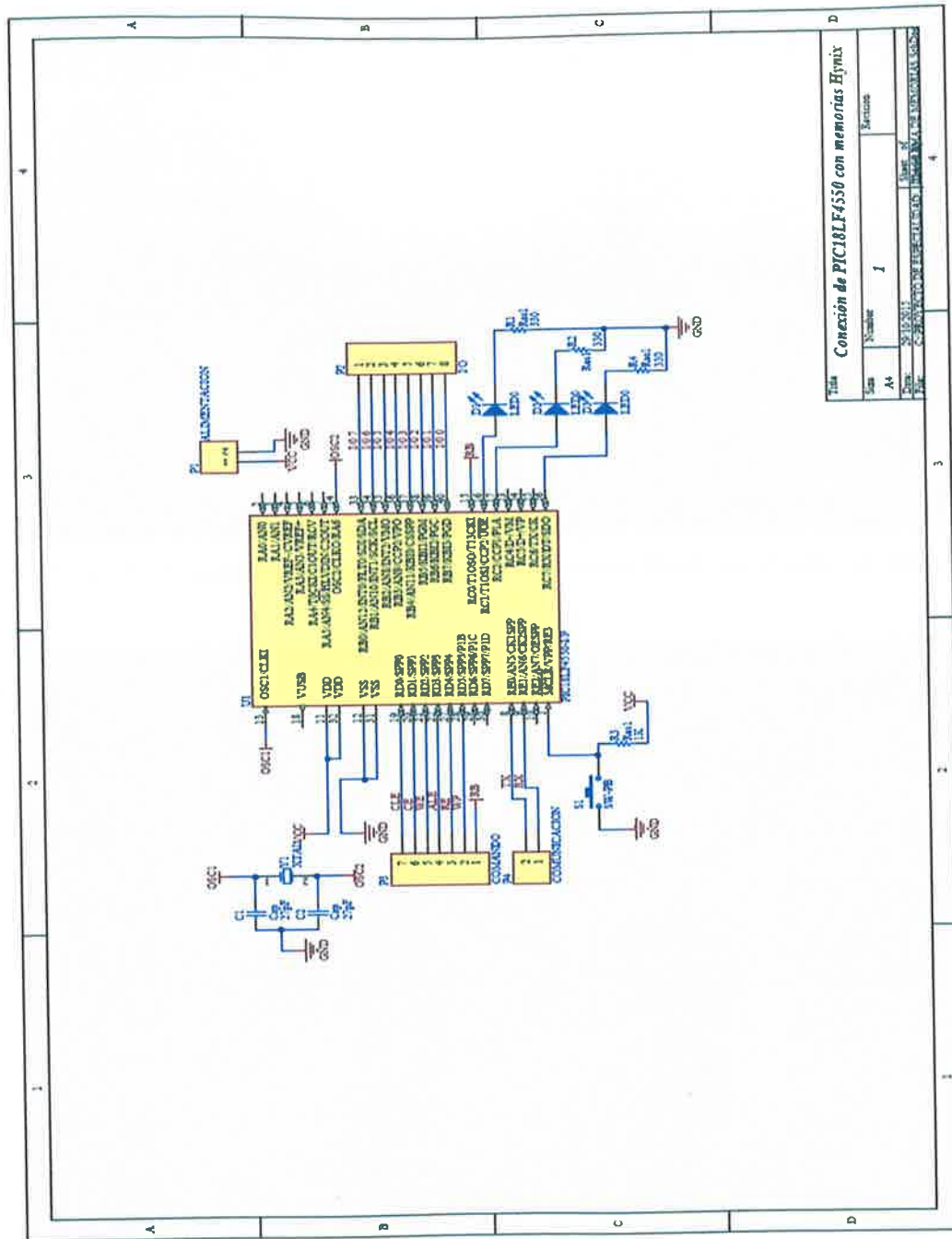
6 BIBLIOGRAFÍA

- [1] SEBASTIAN CASTAÑEDA CENTENO, SERGIO ANDRES RAMOS, and JAVIER RIZO SOLA, MEMORIAS FLASH.
- [2] TRC, Almacenamiento Flash.
- [3] Noe A. Rodriguez Olivarez, Diseño de arquitectura para almacenamiento redundante de información de señales de ultrasonido para inspección de ductos., 2014.
- [4] Miguel Rebollo Pedruelo, Dispositivos de almacenamiento.
- [5] Kingston Technology Corporation, "Guía de productos de memoria Flash," pp. 4-5, 2012.
- [6] Seagate, "Almacenamiento de estado sólido," p. 1, Marzo 2010.
- [7] Inc Micron Technology, NAND Flash 101: An Introduction to NAND Flash and How to Design It In to Your Next Product, 2006.
- [8] Hynix, 4Gb NAND FLASH HY27UF084G2M, Diciembre 2006.
- [9] Eureka Technology, NAND Flash FAQ.
- [10] Jose Soya Treasa and Pradeep.C, Design of a Multichannel NAND Flash Memory Controller for Efficient Utilization of Bandwidth in SSDs.
- [11] SEGGER Microcontroller, SEGGER introduces new "NAND-Flash EVAL" and Test Board, Aug. 29, 1011.
- [12] Arasan Chip Systems Inc. (2015) Arasan. [Online]. <http://arasan.com/products/nand-flash/onfi-3-0/>
- [13] (2001, May) Xilinx Generic Flash Memory Interface Solutions.
- [14] Lattice, NAND Flash Controller, Nov. 2010.
- [15] Eureka Technology, EP501 NAND FLASH Controller.
- [16] Enrique Palacios Municio, Fernando Remiro Dominguez, and Lopez Perez. Lucas J., Microcontrolador PIC 16F84, Desarrollo de proyectos, 2004.
- [17] Compilador de C para el microcontrolador Microchip PIC18F4550.

BIBLIOGRAFIA

- [18] JUAN CAMILO SANCHEZ, TUTORIAL LABVIEW, 2011.
- [19] Digilent. Digilent. [Online].
<https://www.digilentinc.com/Products/Detail.cfm?NavPath=2,1040,1043&Prod=ANALOG-DISCOVERY>
- [20] Microchip Technology Inc., PIC18F2455/2550/4455/4550, 2004.
- [21] Texas Instruments Incorporated, MIXED SIGNAL MICROCONTROLLER MSP430F261x, MSP430F241x, 2007.
- [22] ELNEC, NAND Flash Memories and Programming NAND Flash Memories Using Elnec Device Programmers, 2014.
- [23] Bryan Suk Joon Kim, Eyee Hyun Nam, Ozone (O3): An Out-of-Order Flash Memory Controller Architecture, Hyeonsang Eom, Mayo 2011.

7 ANEXOS



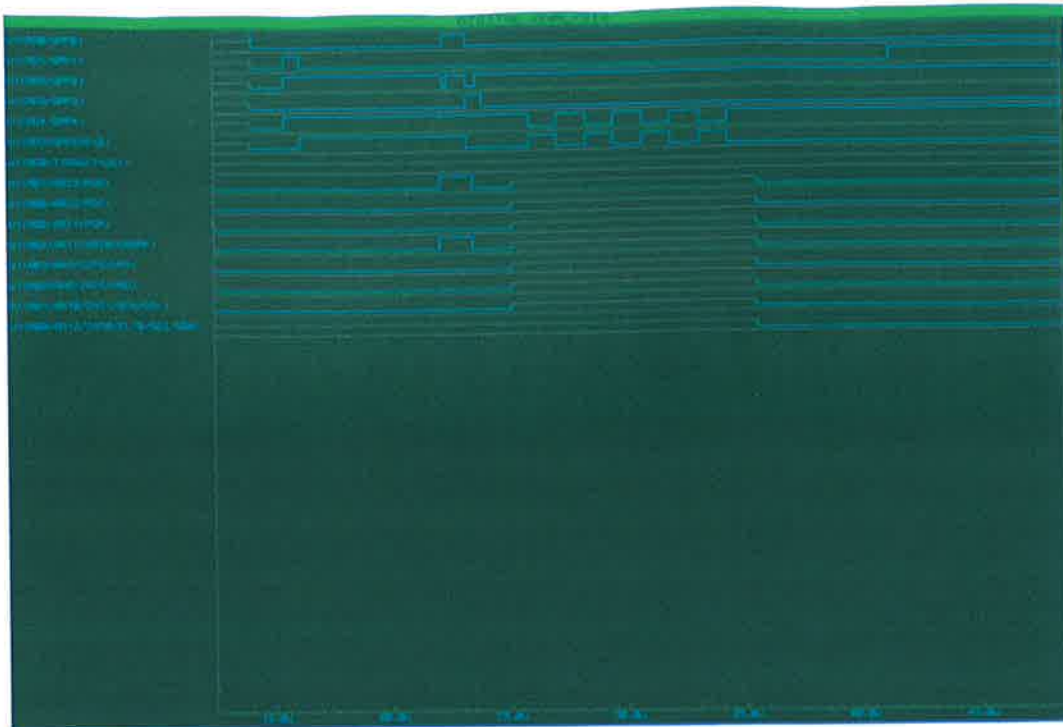


Figura 7.2 Simulación de Lectura de ID de memoria Hynix en el software ISIS 7 Professional.

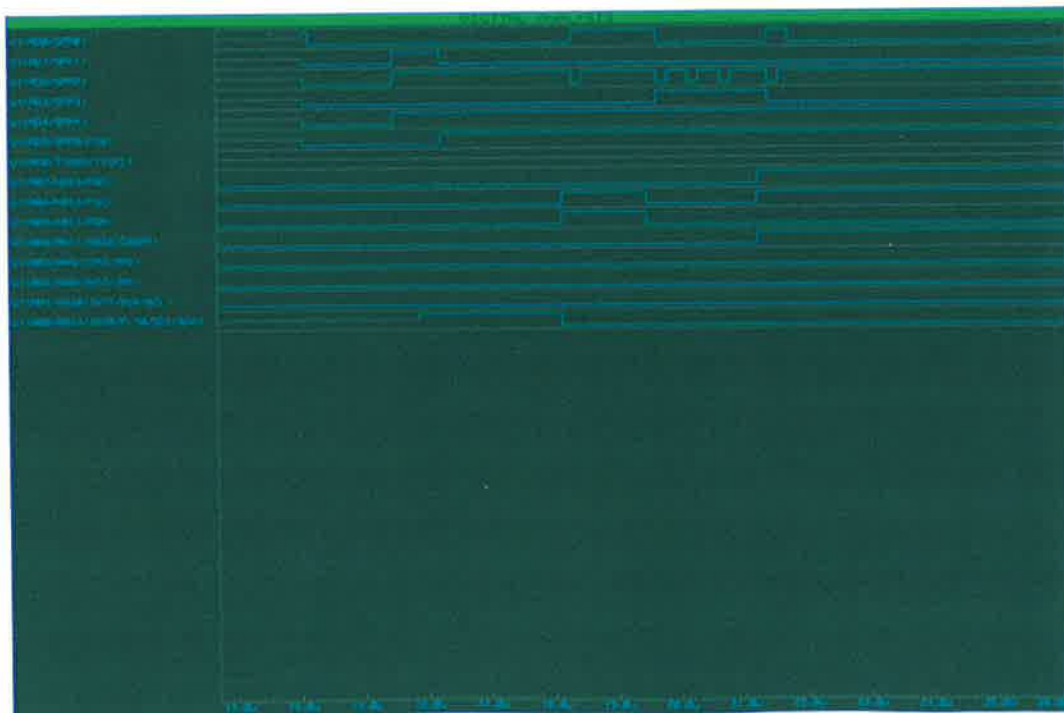


Figura 7.3 Simulación de Borrado de página de memoria Hynix en el software ISIS 7 Professional.

ANEXOS

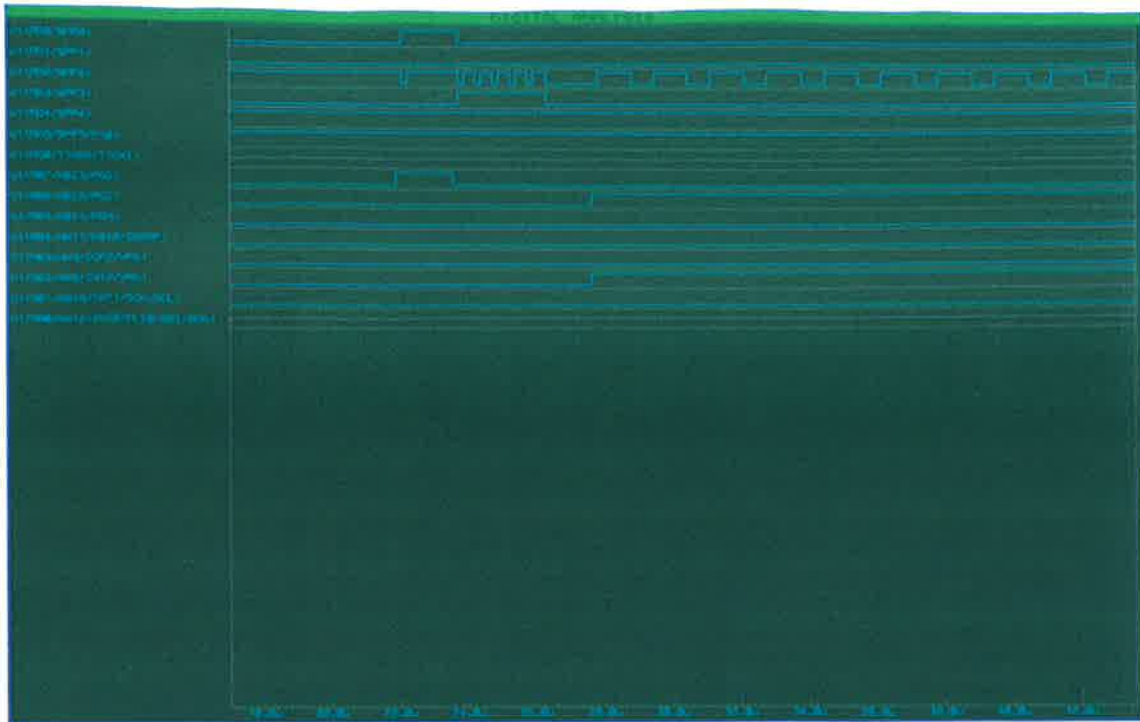


Figura 7.4 Simulación de Escritura de página de memoria Hynix en el software ISIS 7 Professional.

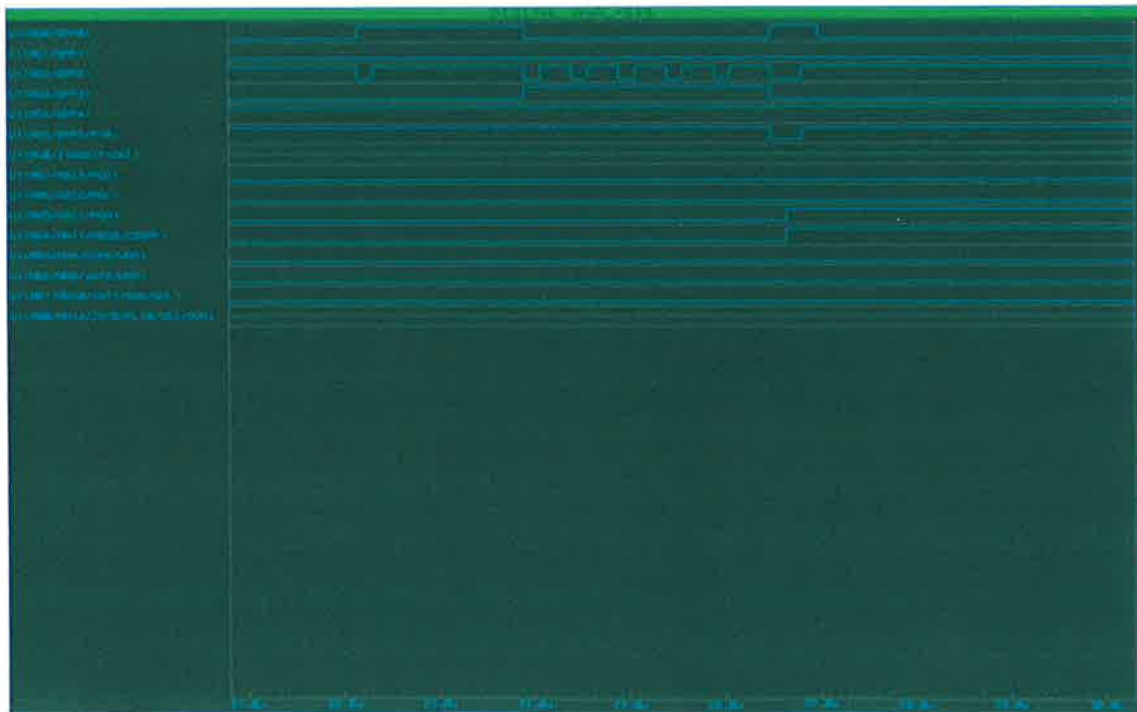


Figura 7.5 Simulación de Lectura de página de memoria Hynix en el software ISIS 7 Professional.