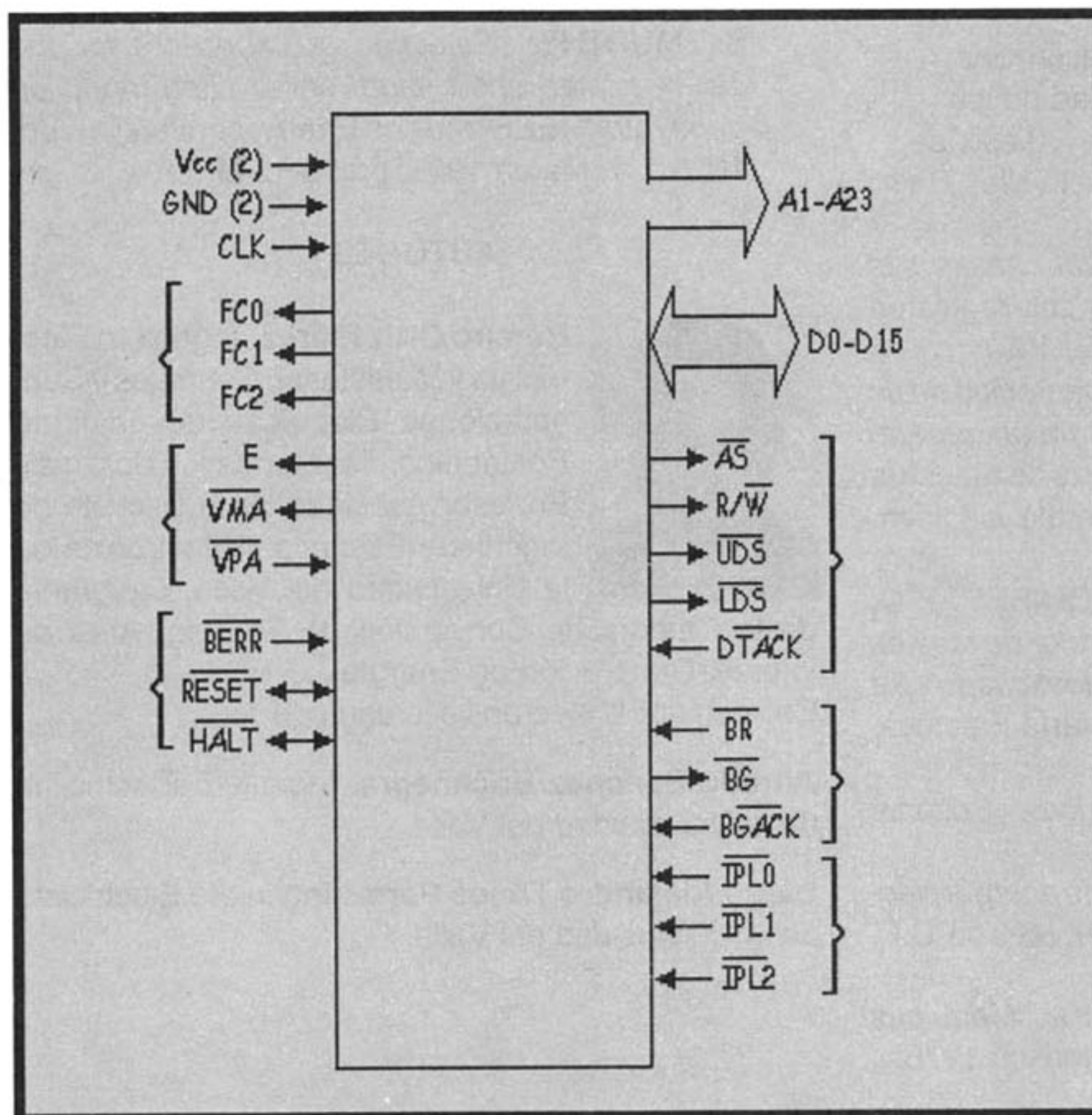


Tarjeta universal de desarrollo para microprocesadores

Diseño e implementación con FPGA



RESUMEN

Este artículo describe la estrategia de diseño e implementación para un sistema universal de desarrollo con microprocesador.

La tarjeta digital se compone de un conjunto de puertos sencillos, memoria RAM, memoria ROM y una base universal para cualquier microprocesador del tipo DIP.

Todos los cambios hardware que deberían realizarse al utilizar un determinado microprocesador, pueden ser reemplazados por modificaciones en el software de configuración de un dispositivo lógico programable (FPGA), que se convertirá en el elemento de enlace entre el microprocesador y el sistema de puertos y memoria.

PALABRAS CLAVES:

FPGA, microprocesadores, diseño con FPGA's, sistema mínimo.

ABSTRACT

This article describes the design strategy and built for a development universal system with microprocessor.

The digital board is compounded of several simple ports, RAM memory, ROM memory and a universal base for any DIP microprocessor. All hardware changes necessities to use a specific microprocessor are substituted for software chan-

☐ **Arturo Gallego Giraldo**
Ingeniero Electrónico

☐ **Rubén Darío Nieto Londoño**
Ingeniero Electricista
Profesor Auxiliar
Universidad del Valle.

☐ **Angel García Baños, Ph.D.**
Ingeniero de
Telecomunicaciones
Profesor Titular
Universidad del Valle

ges using a programmable logic device (FPGA). This element is the connection between microprocessor and memory and ports system.

KEYWORDS:

FPGA, microprocessors, design with FPGA's, DIP.

1. INTRODUCCIÓN

Considérese una unidad de desarrollo típica con tres elementos principales: microprocesador, puertos y memoria. Para que esta unidad funcione, el diseñador debe acompañar al microprocesador de todo un sistema de circuitos integrados adicionales que le permitan suministrar las señales de control y de datos indispensables para que los puertos y la memoria desempeñen su papel.

Por ejemplo, aparecerán varios chips que se encarguen de la decodificación de las líneas de dirección del microprocesador, para que él pueda acceder el puerto o modulo de memoria que le especifica el software. De la misma forma se pueden encontrar controladores de teclado, el generador de reloj, circuitos para el refresco de una posible RAM dinámica, decodificadores a display, controladores de puertos síncronos o asíncronos, etc. Generalmente estos controladores son proporcionados por el mismo fabricante del microprocesador y en algunas ocasiones no son compatibles con los de otro fabricante.

Una FPGA (Field Programmable Gate Array), posee un conjunto de puertos, que son pines configurables como entrada, salida o triestado y gran cantidad de bloques funcionales internos que toman las señales de los puertos y ejecutan la función digital ordenada por el programador en el proceso de configuración. Todo lo anterior se aprecia en la figura 1.

La estrategia de diseño de la tarjeta universal consiste en reemplazar todos aquellos circuitos integrados digitales, ajenos a los puertos y la memoria, por una FPGA (Field Programmable Gate Array) de tal forma que su programa de configuración sea el que determine cual es el microprocesador que manejará la unidad de desarrollo. Así pues, hemos convertido las incompatibilidades hardware, en un problema de software. Se decidió utilizar la FPGA referenciada como XC5206, producida por la compañía XILINX y que tiene 133 puertos de entrada/salida.

2. CONJUNTO DE PUERTOS

La TARJETA UNIVERSAL se compone de varios puertos para las aplicaciones de los usuarios, los cuales son:

- Un parlante para la indicación audible de un evento en especial; básicamente es un dispositivo cono-

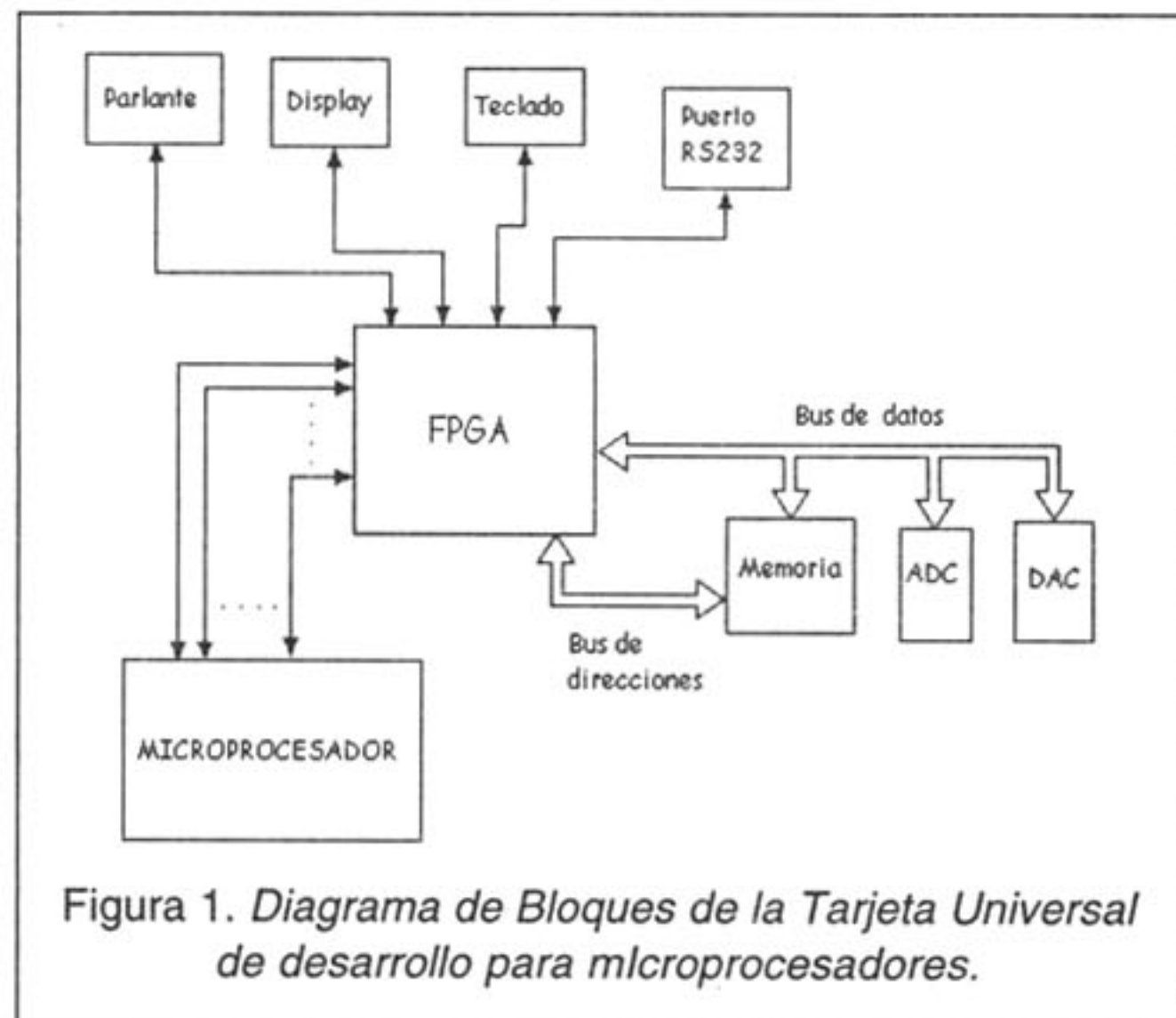


Figura 1. Diagrama de Bloques de la Tarjeta Universal de desarrollo para microprocesadores.

cido como *mini-buzzer* y que produce un sonido de 800 Hz.

- El teclado es un puerto de entrada de 10 dígitos decimales implementado con pulsadores y resistencias de pull-up. Todo problema de rebote que pueda surgir en los contactos de los pulsadores debe ser resuelto internamente en la FPGA.
- Dos displays de siete segmentos multiplexados, en configuración cátodo común.
- Para el muestreo de señales análogas la tarjeta incluye un conversor análogo a digital y que corresponde al ADC0808. Con él podremos digitalizar señales de hasta 10 KHz a un formato de 8 bits.
- El puerto complementario para el tratamiento de señales análogas es el conversor digital a análogo. El chip de base es el conversor DAC0808, que recibe como entrada un código digital de 8 bits y lo convierte en una señal análoga en forma de corriente, cuyos niveles toques son prefijados a través de los voltajes de referencia. El tiempo de conversión es el suficiente para producir hasta 8000 conversiones por segundo, pero lamentablemente el DAC0808 no posee un circuito de captura (o retención) del código digital de entrada, lo cual implica que las entradas no pueden ir directamente al bus de datos. La solución es simple y consiste de agregar un registro cuya señal de captura se active cuando en el bus hay un dato válido a convertir.
- Por último la unidad de desarrollo trae incorporado un puerto RS232 para intercambiar información con otros dispositivos.

La norma RS232 completa especifica muchas señales, pero son las características de velocidad, seguridad y complejidad de los sistemas a comunicar, las que finalmente definen cuales de esas señales serán suficientes. La Tarjeta Universal incluye cuatro señales en el puerto RS232 que son: TX, RX, CTS y RTS. Para informar a un receptor que un dato está dis-

ponible, el transmisor activa la señal RTS (Request To Send). Esta señal interrumpe al receptor o el receptor la detecta por proceso de "polling". Una vez detectada la señal, el receptor finaliza su actividad actual y responde activando CTS (Clear To Send) indicando que se está listo para aceptar caracteres. El transmisor no transmite ningún dato hasta que su entrada CTS sea activada. TX y RX serían los canales por los cuales viajan efectivamente los caracteres. Dependiendo del caso particular, un usuario podría prescindir de las señales CTS y RTS, aplicando la forma típica de comunicación serial asíncrona en la que sólo intervienen las señales TX y RX y haciendo control de flujo por software (XON/XOFF).

La norma RS232 también especifica la interface mecánica y las características eléctricas de las señales. Un conector estándar DB9 será suficiente y el integrado MAX232 se encargará de realizar el acople entre los niveles de voltaje TTL y los niveles de voltaje RS232.

3. SISTEMA DE MEMORIA

Observando sólo los microprocesadores del tipo DIP se puede concluir que la mayoría corresponden a arquitecturas para bus de datos de 8 y 16 bits. Por lo anterior lo más adecuado es utilizar dos bloques de memoria de 8 bits con habilitaciones independientes. El sistema de memoria final es el que se muestra en la figura 2. La FPGA es la encargada de proveer todas las señales de control a través de sus pines.

La señal RAMSEL se encarga de habilitar o deshabilitar los bloques de memoria RAM. Análogamente la señal ROMSEL ejecuta la misma función con los bloques de memoria ROM. La línea RD posibilita la operación de lectura en el bloque habilitado y se compone de dos señales que actúan independientemente, dependiendo del tamaño del bus de datos que define el micro-

procesador. WE actúa de igual forma que RD, pero ahora referido a la operación de escritura sobre memoria RAM.

Como memorias se utilizó la 27C256, que es una memoria ROM de 32K x 8, con tiempo de acceso ubicado en el rango de los 100 a los 250 ns; y la 62256, que es una memoria RAM del mismo tamaño, pero con 120 ns de máximo tiempo de acceso.

El tamaño máximo de memoria RAM y ROM es de 32K en una arquitectura de bus de 16 bits, y de 64K cuando se utilice un bus de 8 bits. La FPGA debe estar conectada al bus de direcciones con el objetivo de realizar internamente todo el proceso de decodificación y activar la señal de habilitación del chip de memoria respectivo.

4. PROGRAMACIÓN DE LA FPGA

FPGA es la abreviatura de "Field Programmable Gate Array", que traducida literalmente es Arreglo de Puertas Lógicas Programables por Campo.

Externamente una FPGA puede verse como un circuito integrado que agrupa todo un conjunto de puertos de entrada y salida, unidos a una estructura interna compuesta de gran cantidad de bloques funcionales programables por el usuario, con los cuales se pueden implementar sistemas digitales complejos.

La familia XC5200 de XILINX integra una variedad de FPGA's con robustas características para el diseño digital de alta densidad. Cada FPGA se compone de Bloques de Entrada/Salida programables (IOB), Bloques Lógicos configurables (CLB) e interconexiones programables, figura 3.

Los IOB's realizan la interfase entre el pin físico externo de la FPGA con su lógica interna; los CLB's constituyen los elementos funcionales con los cuales el usuario construye el sistema digital a implementar; y las Interconexiones Programables se encargan de enlazar todos los bloques funcionales dentro de la FPGA.

La compañía XILINX ha producido una serie de poderosas herramientas software para el diseño con FPGA's, denominada *XACT step*, especialmente optimizadas para aprovechar las igualmente poderosas características de los dispositivos de la familia XC5200. La figura 4 muestra el diagrama de flujo en el que se relacionan los pasos necesarios para la programación de la FPGA. La metodología básica para el diseño con FPGA's consiste de tres pasos interrelacionados: *entrada, implementación y verificación*.

Entrada es el proceso de diseño del sistema digital que se desea implementar usando esquemas o símbolos; éstos pueden ser del tipo gráfico, en forma de texto, usando ecuaciones booleanas, máquinas de estado, etcétera. Todo dependiendo del ambiente del editor en que se este trabajando. *XACT step* puede soportar

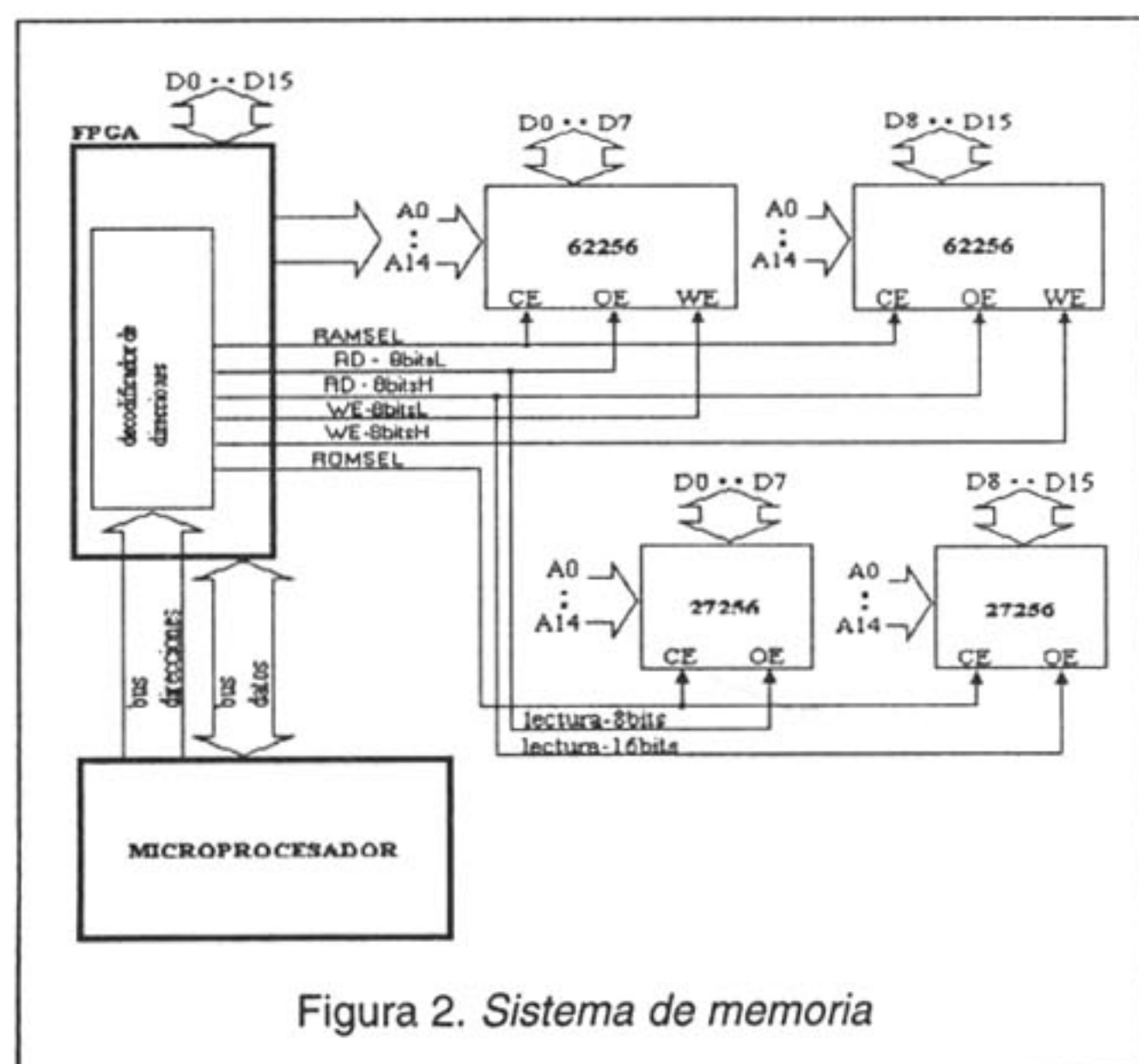
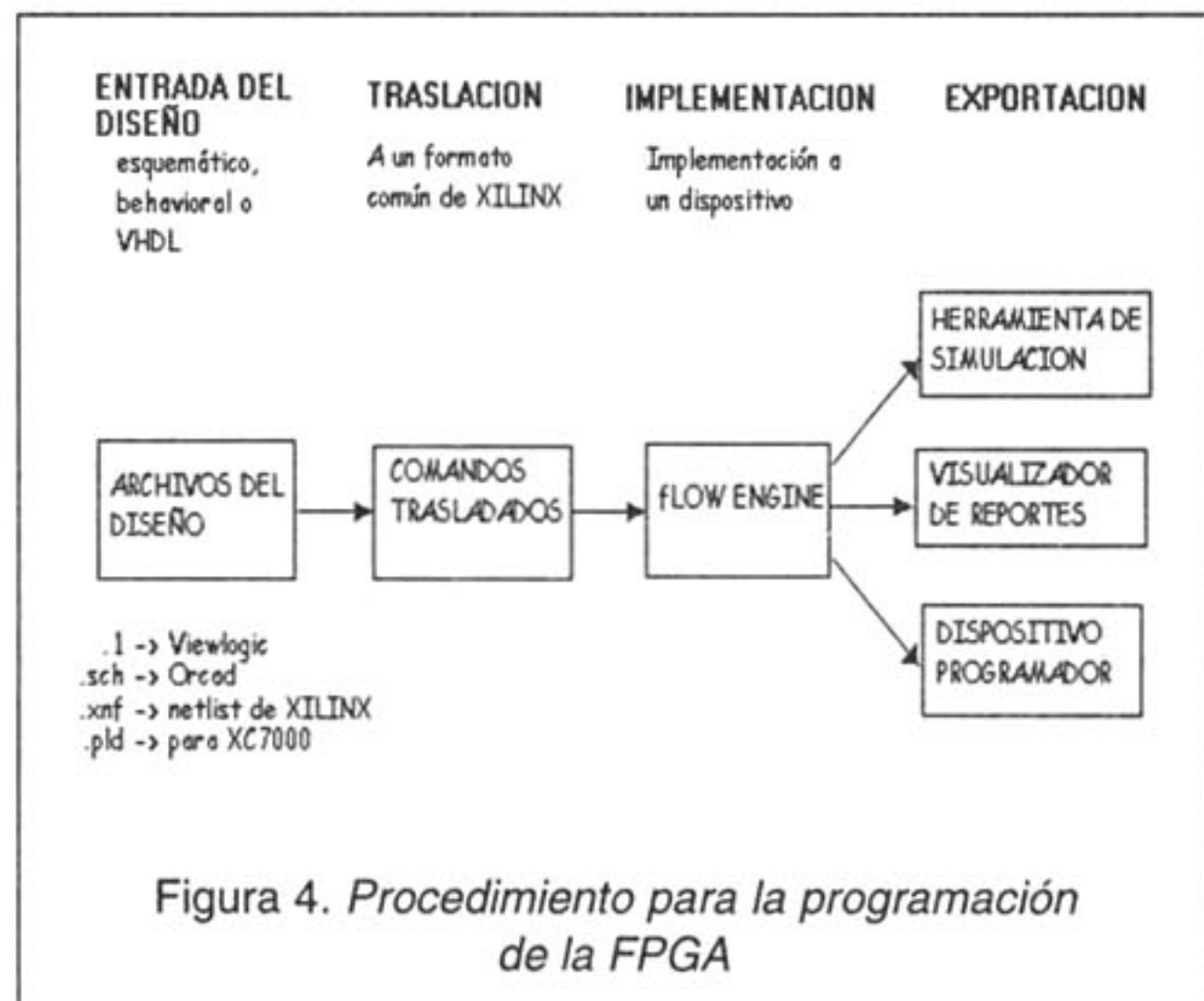
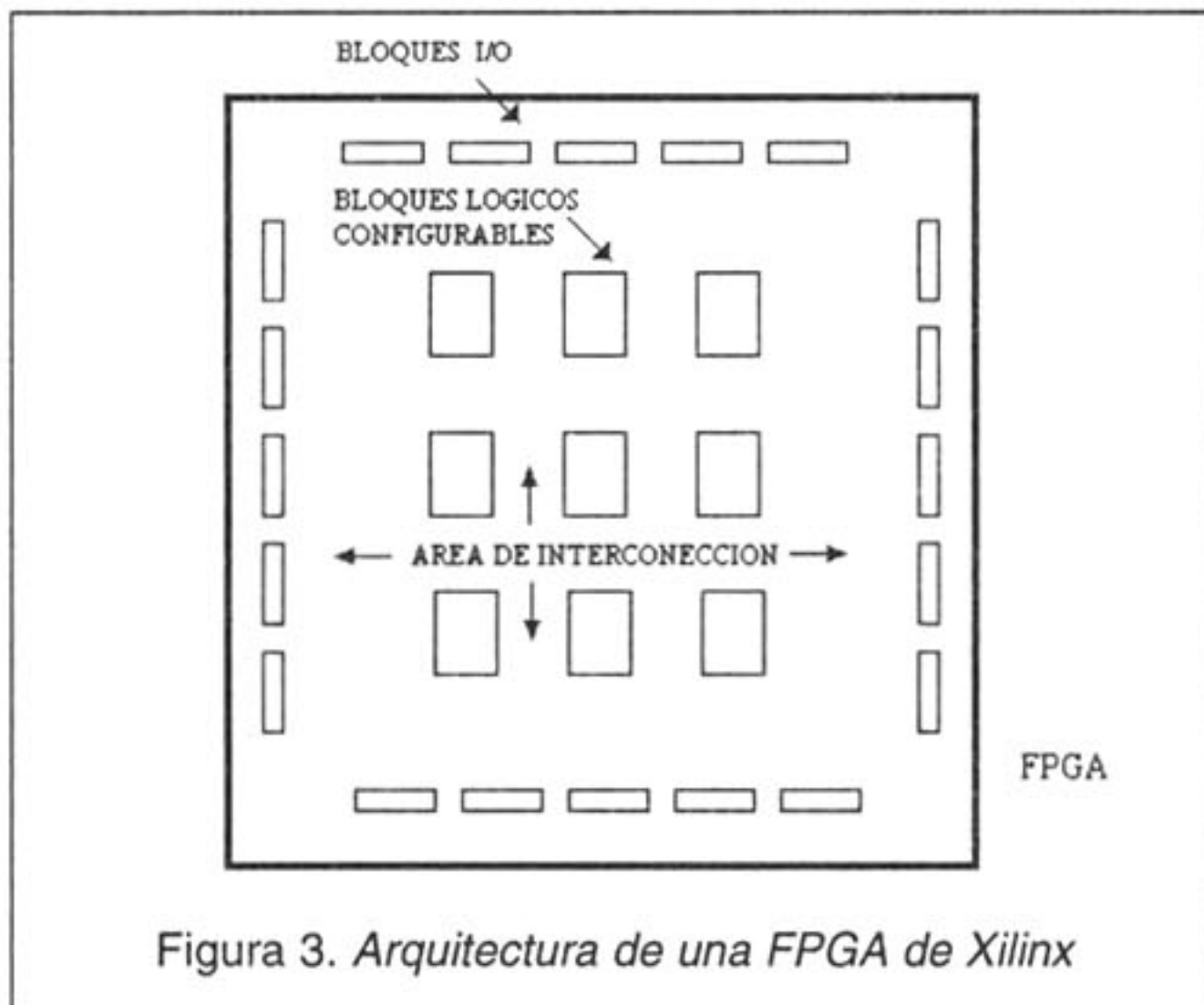


Figura 2. Sistema de memoria



archivos de varios ambientes de editores, entre los cuales se cuenta: la Serie Inicial de Xilinx, Sistemas Viewlogic, Mentor Graphics V8, ORCAD, Synopsys, Xilinx-ABEL y X-BLOX.

Para la parte de *implementación* se genera el programa que debe contener la FPGA seleccionada para reemplazar el sistema digital especificado en la parte de diseño.

En la *verificación* el usuario confronta los resultados entregados por la implementación, con los generados en la parte de diseño. Se pueden utilizar dos técnicas: *simulación*, en la cual se puede observar la respuesta del sistema ante situaciones especiales; o *depuración*, utilizando un circuito de prueba, con lo cual el proceso de verificación se ejecuta en tiempo real.

La secuencia ordenada de pasos para programar nuestra FPGA se resume así:

- Teniendo como entrada el diseño en Orcad, se traslada el archivo de diseño al formato interno de la XILINX.
- Implementa el diseño para el dispositivo específico XC5206 de XILINX.
- Genera reportes que muestra el estado del diseño.
- Programa la FPGA.

5. EJEMPLO DE APLICACIÓN

Antes de programar la FPGA es necesario una preparación teórica básica sobre el microprocesador que se va a trabajar, sobre todo en cuanto a los requerimientos hardware. Como por ejemplo, las condiciones para el reset, las características de la señal de reloj, los ciclos de lectura y escritura sobre el bus, etcétera.

Después de cubrir esta parte, se deben editar los circuitos en Orcad que se desean probar sobre la tarjeta.

Para ilustrar el funcionamiento de la unidad de desa-

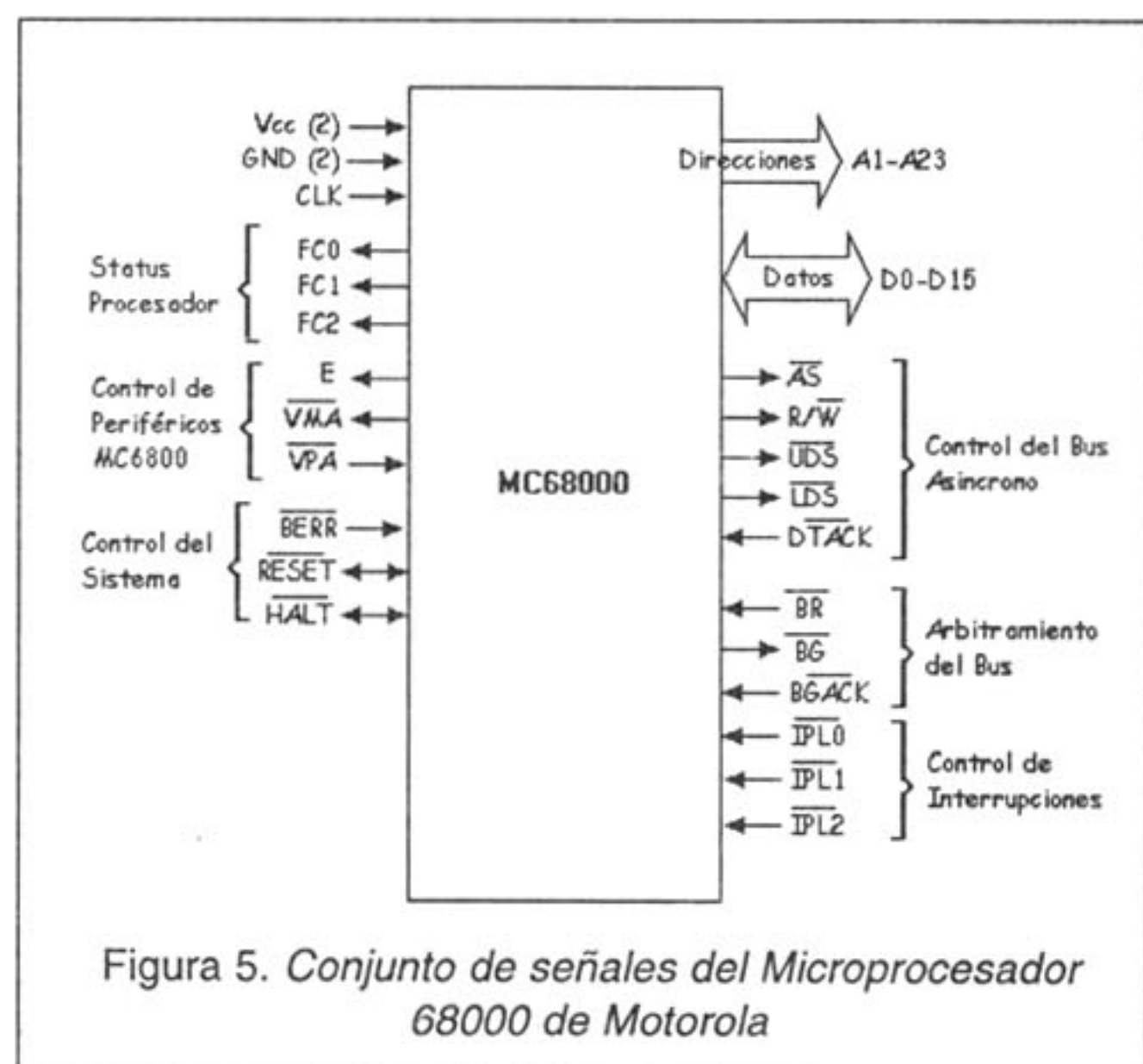
rollo universal se utilizará al microprocesador fabricado por Motorola con referencia MC68000 (ver figura 5).

5.1 Características Hardware del Microprocesador 68000 de Motorola

En la figura 5 se puede observar el conjunto de señales que involucra el microprocesador, este posee un bus de datos externo de 16 bits y utiliza 24 líneas como bus de direcciones, lo que implica una capacidad de direccionamiento de 16 Mb.

AS (Address Strobe) es una línea de salida que se activa cuando el bus de direcciones contiene una dirección válida. El bit A0 del bus de direcciones es interno al microprocesador y al asumirlo como un 0 se tendrá una dirección par.

El 68000 siempre lee una dirección par cuando captura instrucciones, así que A0 es innecesario para ejecución de programas. Sin embargo cuando se necesitan acceder datos en bytes, se utilizan las señales UDS



(upper data strobe) y LDS (lower data strobe). Si UDS=0, la lectura o escritura se hace sobre D8-D15 en el word direccionado; si LDS=0, se accede a D0-D7. Si ambas señales se activan simultáneamente durante el mismo ciclo de bus, la operación de escritura o lectura se aplica a todo el word.

La entrada DTACK (Data Acknowledge) debe ser activada por un dispositivo externo que participe en una operación de transferencia de datos. El 68000 esperará indefinidamente durante cada ciclo de bus el DTACK del dispositivo que fue direccionado, apenas llegue terminará el ciclo normal. Debido a esta forma de control asíncrona del bus, se debe acompañar al microprocesador de una lógica para control de fallos. FC0, FC1 y FC2 (Function Code) indican mediante un código de salida el tipo de actividad del bus en progreso. BERR (Bus Error) informa al microprocesador por parte de un circuito externo que ha ocurrido un error en el ciclo de bus. BR (Bus Request), BG (Bus Grant) y BGACK (Bus Grant Acknowledge) son señales que intervienen cuando otro dispositivo ajeno al microprocesador pretende tomar el control del bus, por ejemplo, en caso de tener un controlador de DMA. IPL0, IPL1 e IPL2 (Interrupt Priority Level) permiten la implementación de interrupciones externas con nivel de prioridad. RESET y HALT proveen la forma de inicializar (reset) o detener al microprocesador en un momento dado.

5.2 Diseñando circuitos para la FPGA

5.2.1 Reset del microprocesador. Es imprescindible el diseño de un circuito de Reset externo que permita inicializar al microprocesador. Según el fabricante, el estado de Reset se consigue colocando las entradas RESET y HALT simultáneamente en estado bajo por un tiempo mínimo de 100 milisegundos. Esto se consigue en un circuito como el mostrado en la figura 6.

Para lograr esta temporización se pueden utilizar una serie de flip-flops tipo D, como se observa en la figura

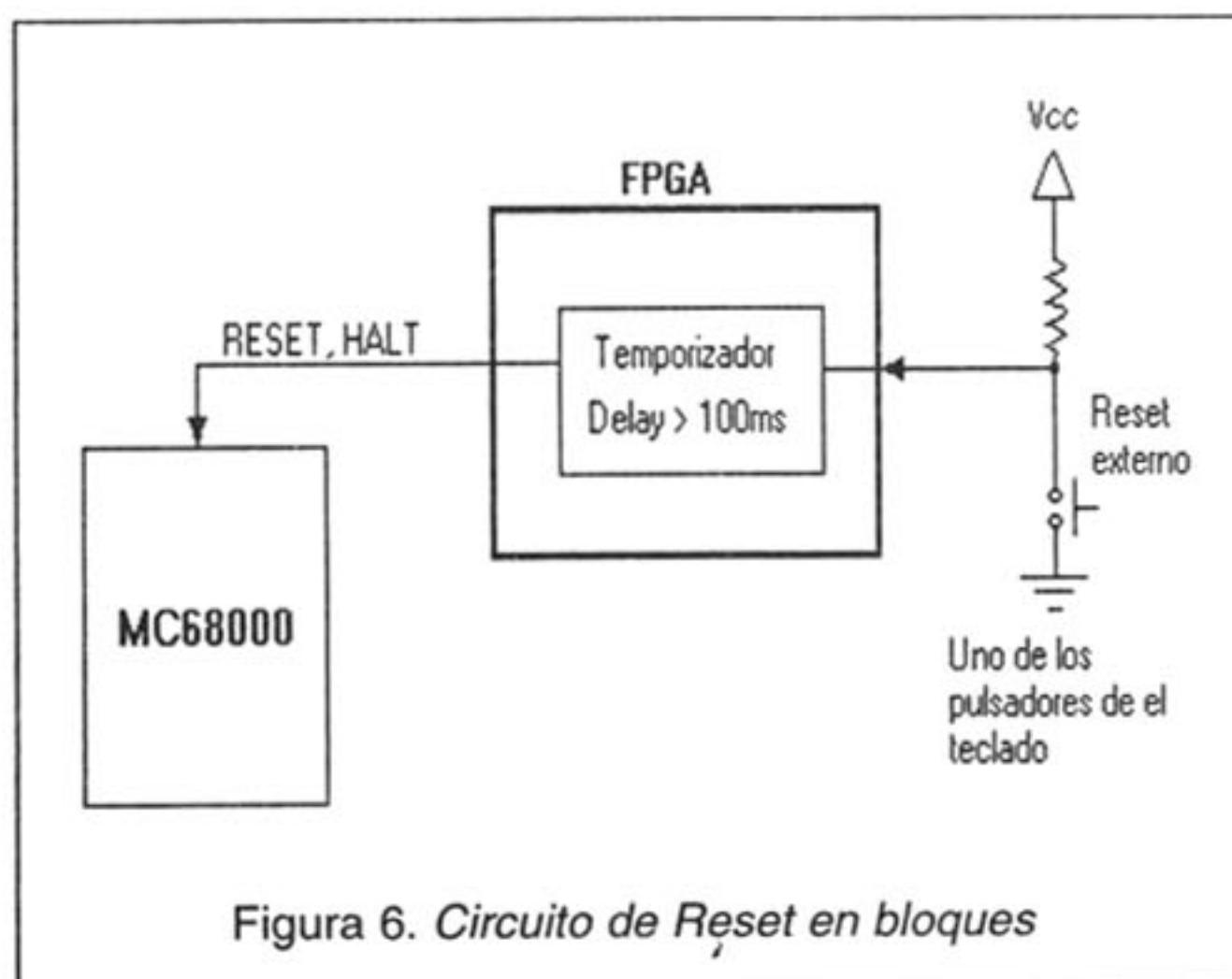


Figura 6. Circuito de Reset en bloques

7. Un nivel lógico 1 rotará por todos los flip-flops, pero apenas se envíe un pulso de reset, Q3 permanecerá en 0 por un espacio de tiempo de 3 períodos de reloj. En cuanto a la fuente de reloj podemos aprovechar que la FPGA XC5206 trae un divisor de frecuencia interno programable. Durante la configuración el usuario escoge si requiere división por 2, 4, 8, 16, 32, 64, 128, 256, 1024, 4096, o 65536 y si la fuente de reloj será el oscilador interno de 16 MHz o una fuente de reloj externa.

El diseño usa el reloj interno de 16 MHz y lo divide inicialmente por 65536, con lo cual se disminuye la frecuencia a 244.14 Hz. Luego, a partir de cuatro flip-flops T en cascada, se divide nuevamente la frecuencia por 16 y finalmente se tiene una señal oscilatoria de 15.25 Hz, que es adecuada para aplicarla a la serie de flip-flops D. Q3 es una buena señal de Reset, pues 3 períodos demorarán aproximadamente 200 milisegundos.

5.2.2 Sistema mínimo. Ahora se implementará un sistema mínimo con el microprocesador MC68000 de Motorola. Este sistema permite visualizar, en un osciloscopio, las señales del microprocesador que intervienen en un ciclo de bus normal. La idea es poner a correr al microprocesador libremente, es decir, que capture (fetch) una instrucción de tamaño word, la ejecute, capture otra instrucción igual en la próxima dirección par, y así hasta recorrer todo el rango de 16 Mb. Un osciloscopio puede ser colocado para ver las líneas de control y las líneas de dirección y datos actuando en el proceso normal de lectura de bus del microprocesador. La FPGA está programada para enviar siempre 0 a través del bus de datos, figura 8. La lógica de reset es exactamente igual a la descrita en los párrafos anteriores.

Una vez el 68000 detecta el pulso de reset se enciende el led conectado a la línea HALT y hace cuatro lecturas sobre el bus para actualizar los valores de los vectores SSP (Supervisor Stack Pointer) y PC (Program Counter), que son registros internos del microprocesador. Por tanto SSP=0000 y PC=0000. Inmediatamente el microprocesador buscará la primera instrucción en la dirección que le indica el PC. La lectura dará como resultado un Op-code de 0000, que corresponde a una instrucción real del 68000, cuyo mnemónico es *ORI.B #0,D0* y que realiza la operación lógica OR sobre el registro interno D0. El bus de direcciones señalará a la siguiente dirección par y repetirá el proceso en forma iterativa.

El tiempo de ejecución de este "programa" es fácil de calcular. Si cada instrucción toma 8 ciclos de reloj, con un reloj de 16 MHz, el tiempo de ejecución de cada instrucción es de aproximadamente 500 nanosegundos. Un recorrido completo a través de los 16 Mb demorará cerca de 2 segundos. Si conectamos el led

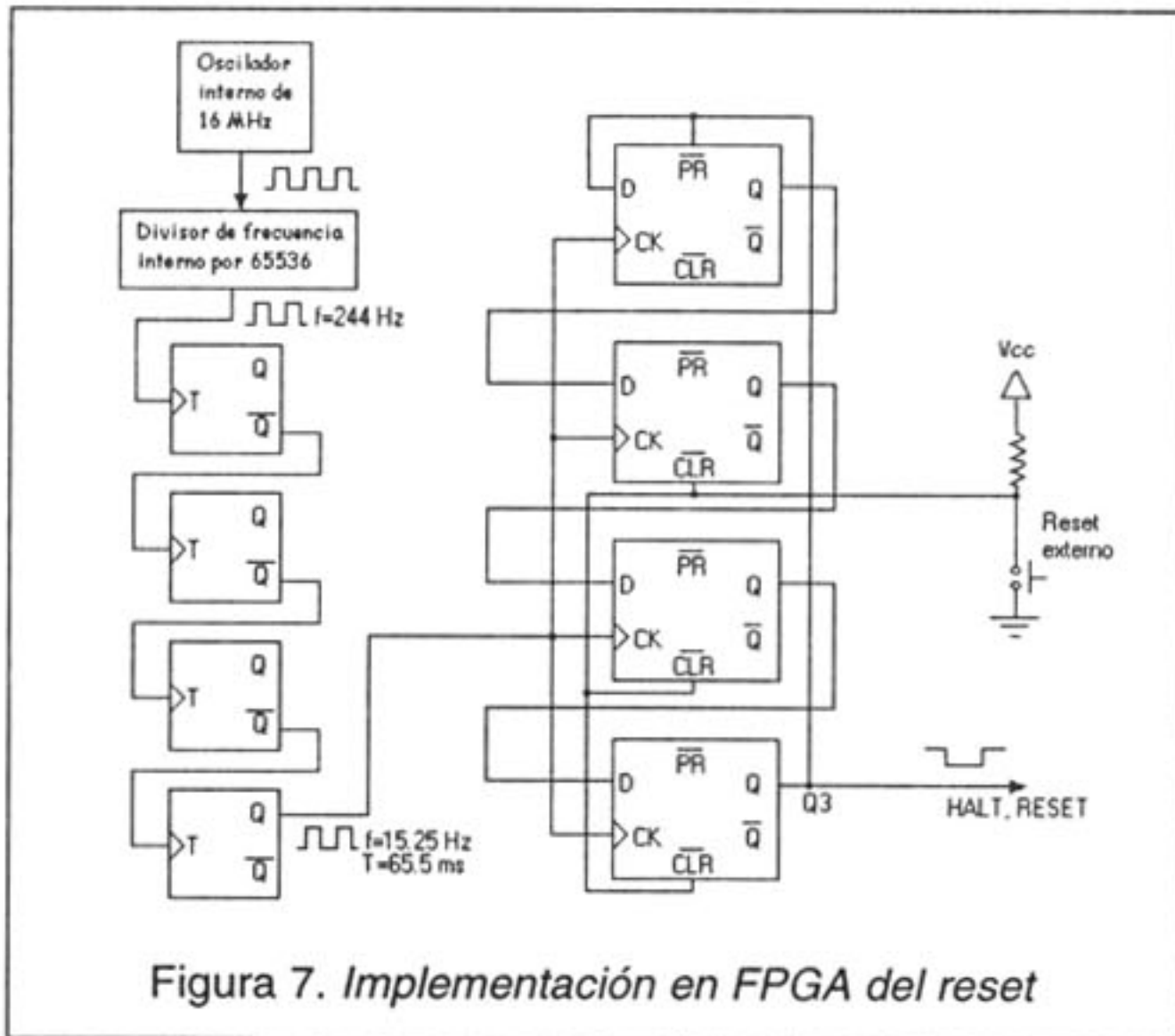


Figura 7. Implementación en FPGA del reset

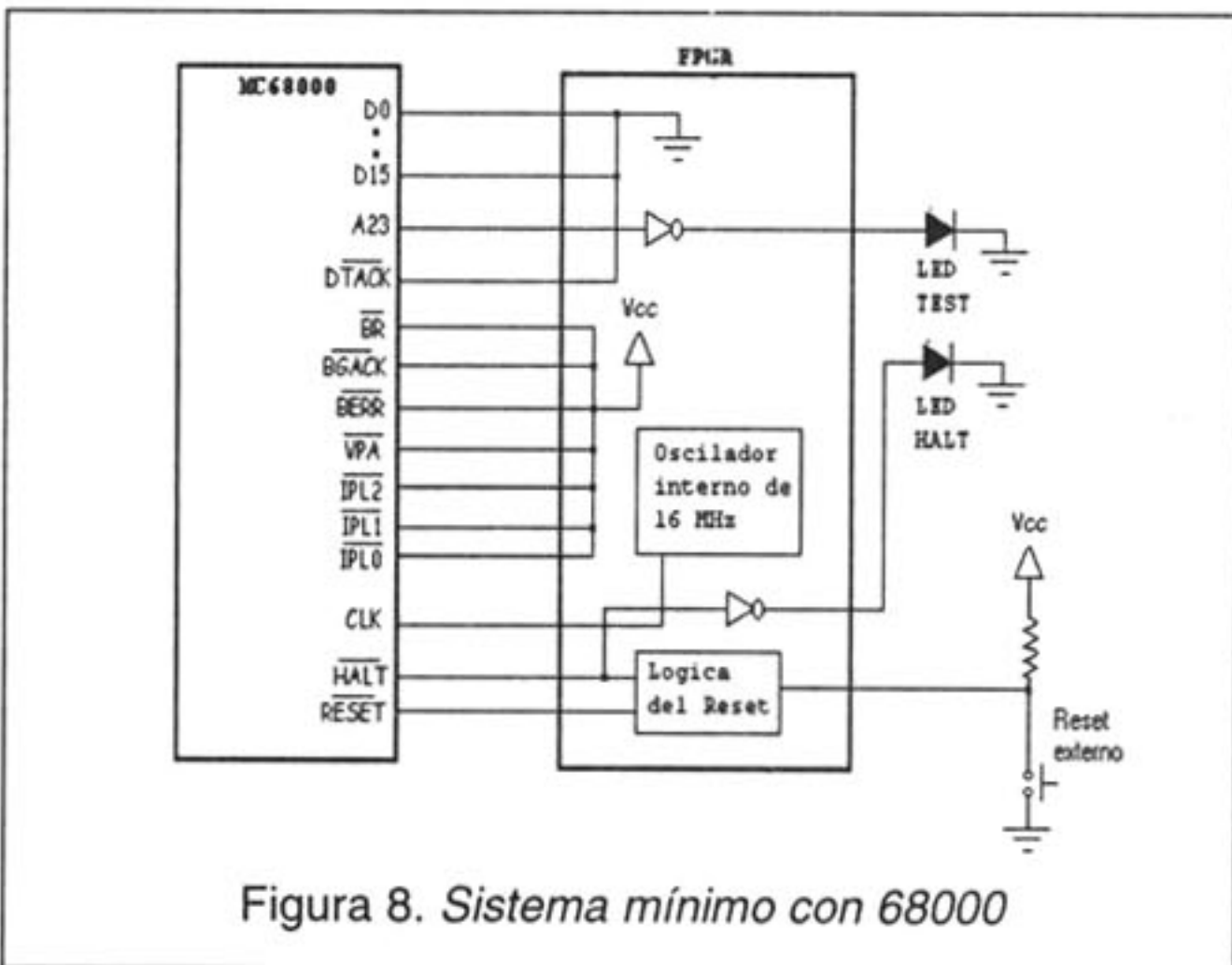


Figura 8. Sistema mínimo con 68000

TEST en la línea de dirección A23, encenderá por espacio de 1 segundo, cada 2 segundos.

5.2.3 Ejecutando un programa. Por último se ilustra una aplicación en la que el 68000 ejecuta un programa simple en memoria baja usando sólo memoria ROM. El ciclo de lectura de bus mínimo consume 4 ciclos de reloj, figura 9. Al final de S4, el 68000 verifica si se ha producido el DTACK. Si es así, en el próximo flanco negativo capturará el contenido del bus de datos. Pero si no, el procesador insertará estados de espera (wait states) directamente después de S4 hasta que el DTACK sea detectado en un flanco de caída de reloj. Los wait states son agregados en pares (un par igual a un ciclo de reloj). El ciclo de lectura de bytes es idéntico al de lectura de words, pero ahora sólo se activaría la señal UDS o LDS respectivamente, dependiendo si la dirección es par o impar. Si dejamos a DTACK siempre en 0, la memoria debe tener un tiempo de acceso menor al tiempo que demoran 4 estados (S3 a S6); Con un reloj de 16 MHz, esto equivale a 62.5 nanosegundos.

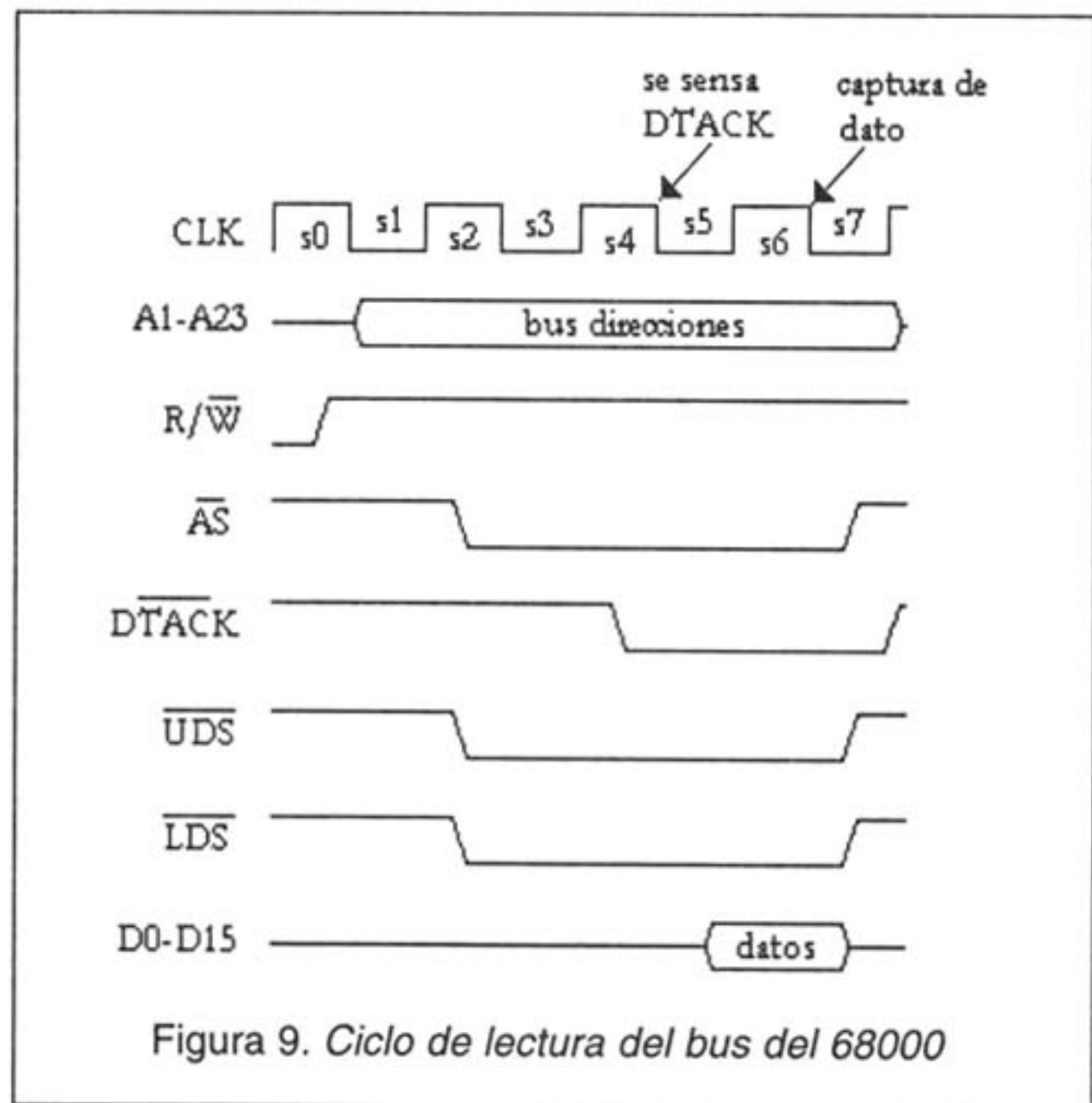


Figura 9. Ciclo de lectura del bus del 68000

Como la memoria 27C256 no cumple con estas especificaciones necesitamos una lógica adicional para generar DTACK, como el que se muestra en la figura 10. Apenas es detectado el inicio del ciclo de bus las salidas del registro de corrimiento se ponen en 0 por varios ciclos de reloj. La programación de la FPGA deberá conectar DTACK en la salida adecuada del registro de forma tal que respete el tiempo de acceso de la memoria.

El diseño para memoria EPROM puede ser tan simple como el de la figura 11. Usando memorias 27C256 (32Kx8) el mapa de memoria repetirá los mismos datos cada 64K direcciones. Al ocurrir el reset (led HALT encendido), el 68000 lee el SSP en la dirección 0, luego lee el PC en la dirección 4. El PC le ordena al microprocesador hacer fetch en la dirección 8, donde tendremos la primera instrucción del programa la cual corresponde a la operación de salto incondicional (JUMP).

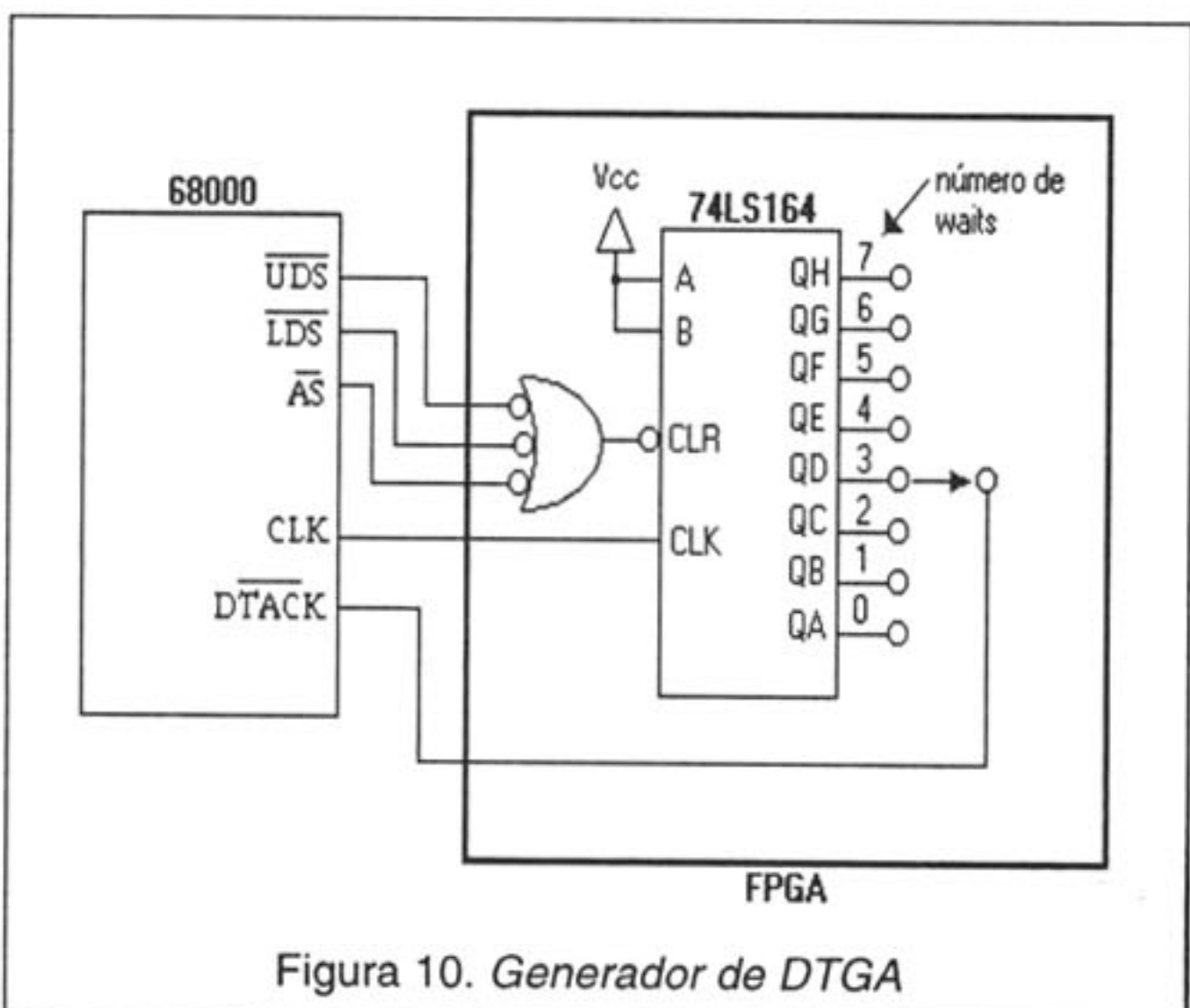
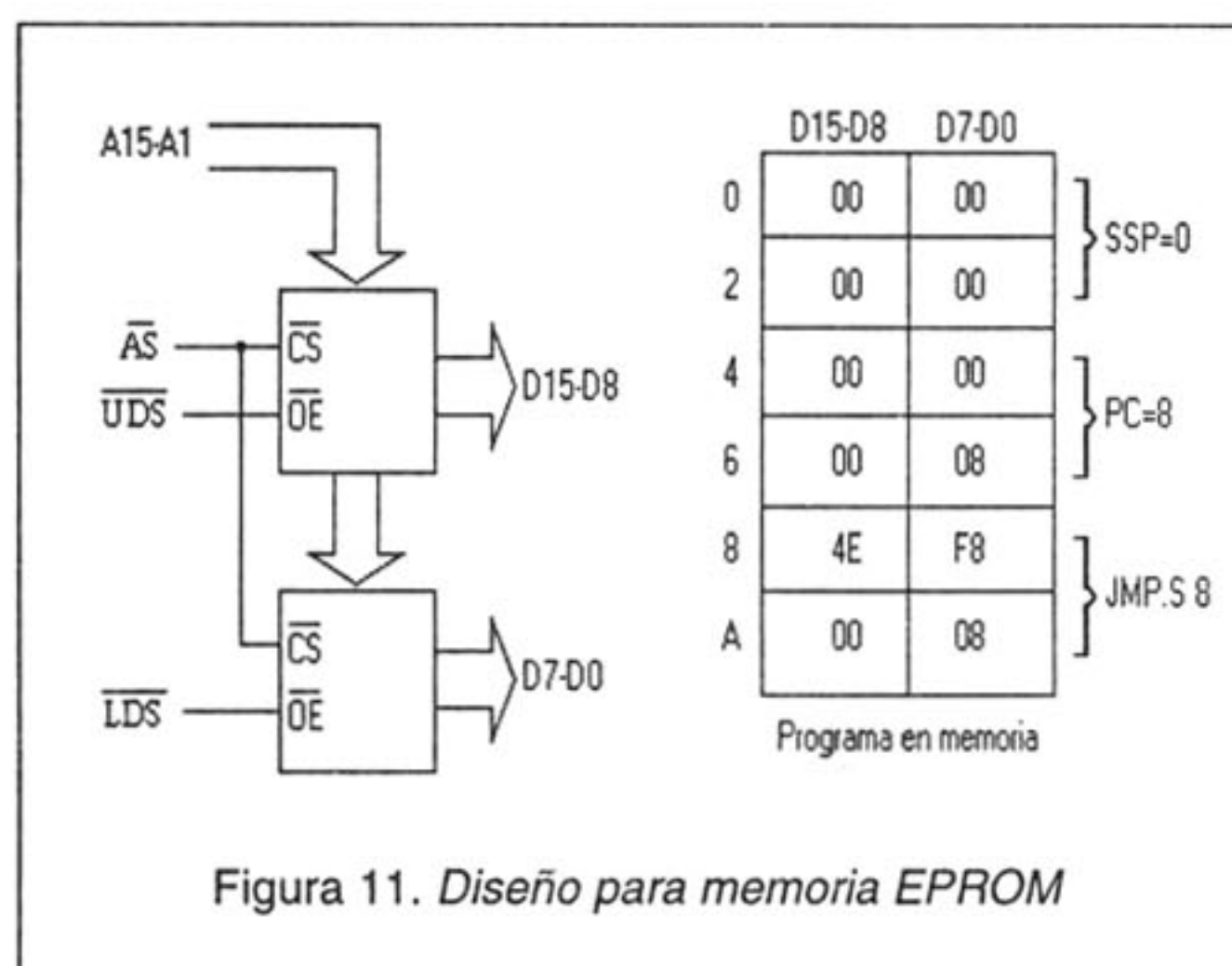


Figura 10. Generador de DTACK



6. CONCLUSIONES

6.1 La Tarjeta Universal de Desarrollo para Microprocesadores es un sistema de puertos y memoria con todos los elementos necesarios que le permitan conectarse a toda una variedad de microprocesadores del tipo DIP.

6.2 Desde el punto de vista pedagógico, la Tarjeta Universal constituye una poderosa herramienta para el aprendizaje del manejo software de un determinado microprocesador, mediante la ejecución de programas que utilicen los recursos de la unidad de desarrollo. Además proporciona todos los elementos necesarios para la implementación física de circuitos digitales, que conducen al conocimiento del manejo hardware de un microprocesador específico.

6.3 Finalmente queda comprobada la gran versatilidad de las FPGA's para solucionar numerosos problemas en el diseño de sistemas digitales.

7. BIBLIOGRAFÍA

[1.] WILCOX, A. D. 68000 Microcomputer Systems, Designing and Troubleshooting. Prentice Hall, USA, 1987.

- [2.] RAFIQUZZAMAN, M. Microprocessors, Theory and Applications. Prentice Hall, USA, 1992.
 [3.] XILINX, Inc. The Programmable Logic Data Book. USA, 1994.
 [4.] XILINX, Inc. The Programmable Logic Data Book. USA, 1996.
 [5.] XILINX, Inc. OrCAD Interface/Tutorial Guide. USA, 1995.

AUTORES

Arturo Gallego Giraldo. Ingeniero Electrónico por la Universidad del Valle. Areas de interés: Sistemas digitales y Microprocesadores. En la actualidad se encuentra laborando en la empresa privada en el sector de las telecomunicaciones.

Ce: agallego@telesat.com



Rubén Darío Nieto Londoño. Ingeniero Electricista por la Universidad del Valle. Profesor Auxiliar adscrito a la cátedra de Arquitecturas Digitales de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle. Su trabajo se centra en el área de diseño lógico con FPGA's y microcontroladores. En la actualidad trabaja en tesis de Maestría en Automática en la Universidad del Valle.

Ce: rnieto@eiee.univalle.edu.co

Ángel García Baños. Ingeniero de Telecomunicaciones por la Universidad Politécnica de Madrid y profesor titular de la Escuela de Ingeniería Eléctrica y Electrónica de la Universidad del Valle. Dirige un grupo de trabajo en FPGAs y acaba de culminar su tesis doctoral en aplicación de las FPGAs a sistemas de tiempo real.

Ce: angarcia@eiee.univalle.edu.co

