SISTEMA DE ADQUISICIÓN DE DATOS IMPLEMENTADO EN UN FPGA Y UTILIZANDO LA TARJETA DE DESARROLLO NEXYS2

Sistema de Adquisición de Datos Implementado en un FPGA y Utilizando la Tarjeta de Desarrollo Nexys2

M. en C. Miguel Angel Rodríguez Fuentes

UPIITA-IPN. mrodriguezf@ipn.mx

M. en C. Alejandro Escamilla Navarro

UPIIH-IPN. aescamillan@ipn.mx

Abstract: Data Acquisition System based on FPGA and using Nexys2 development board.

Many times, Engineers need the acquisition of data for measurement and data analysis. A Data Acquisition System is developed in a Field Programmable Gate Array (FPGA) and using VHDL description language. Data is stored in a Random-Access Memory of the Nexyx2 board and then exported to a spreadsheet for further analysis.

I. Introducción

Muchos problemas de Ingeniería requieren de la medición y análisis de datos generados por un determinado sensor. Dos de los parámetros importantes en un sistema de adquisición de datos son el número de muestras requerido y el tiempo en el que se registra cada medición (tiempo de muestreo). Se presenta en el documento el uso de la tarjeta de desarrollo Nexys2 (utilizada en asignaturas del área de electrónica digital en la UPIITA) como sistema de adquisición de datos. La tarjeta Nexys2 cuenta con un FPGA (Arreglo de Compuertas Programable en Campo) Spartan3 (Xilinx Inc.) en el cual se implementará una serie de circuitos digitales expresados por medio de un lenguaje descriptivo de Hardware (VHDL) a fin de implementar el sistema de adquisición de datos donde se almacenarán datos temporalmente en una memoria RAM (Memoria de Acceso Aleatorio) de la propia tarjeta Nexys, para ser posteriormente leídos y exportados a una hoja de cálculo para su análisis posterior.

II.Desarrollo

El diagrama del sistema de adquisición de datos implementado en la tarjeta Nexys2 se muestra en la Figura 1. La tarjeta Nexys cuenta con un FPGA donde se implementan los circuitos digitales necesarios para el sistema de adquisición de datos y también cuenta con una memoria RAM donde se almacenarán temporalmente los datos. Los bloques contenidos dentro del FPGA fueron descritos en el lenguaje descriptivo de hardware VHDL y tienen de forma general las siguientes tareas:

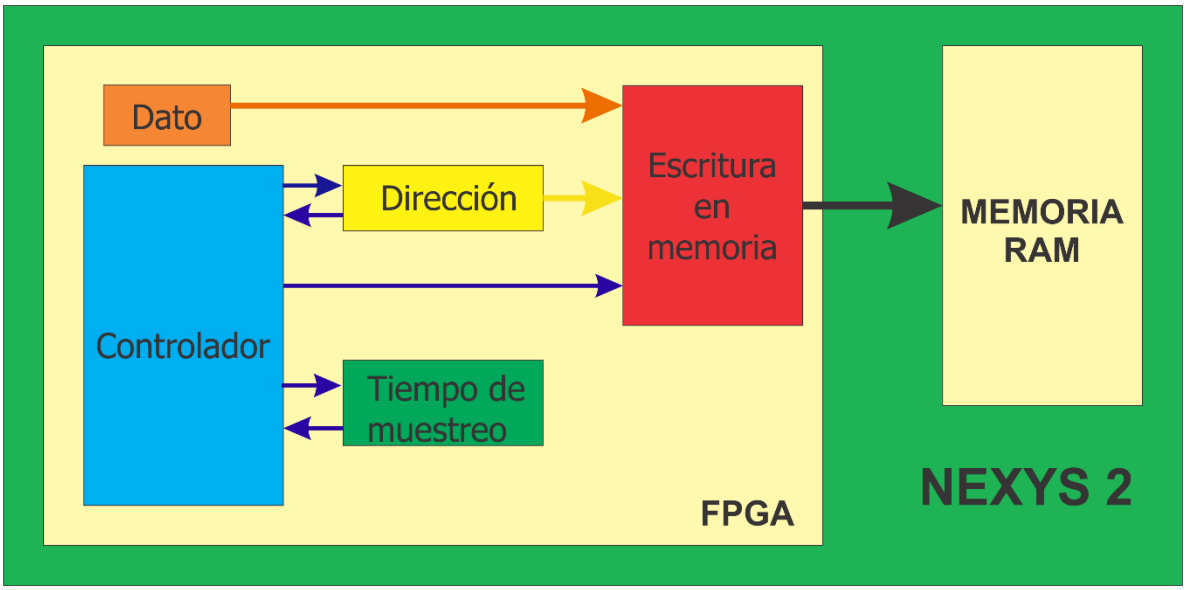


Figura 1. Diagrama a bloques del sistema de adquisición de datos.

a)Dato: Genera los datos que se escribirán en la memoria RAM, a modo de ejemplo, será simplemente un contador ascendente.

b)Escritura en memoria: Es una máquina de estados (FSM por sus siglas en ingles) la cual se encarga de escribir un dato de 16 bits en una determinada dirección de memoria.

c)Dirección: Es un contador ascendente el cual genera las direcciones de memoria en donde se guardarán los datos de forma consecutiva.

d)Tiempo de muestreo: Es un contador ascendente utilizado para controlar el tiempo de muestreo entre cada dato.

e)Controlador: Se encarga de planificar las tareas de los bloques escritura en memoria, dirección y tiempo de muestreo.

En la Figura 1 se muestra el camino de los datos (flecha de color naranja), la dirección donde se guardarán los datos (flecha de color amarillo), las señales de control (flechas de color azul) y la interconexión del FPGA con la memoria RAM (flecha de color negra).

1.Bloque Dato

Es un contador ascendente módulo 65536 activado por flanco ascendente, Figura 2. Utiliza una señal de reloj de 50MHz. Tiene una señal de habilitación de conteo de tal forma que si la señal de entrada *wr=0* el circuito cuenta y si *wr=1* el circuito deja de contar. La señal *wr* permitirá que durante la escritura de un dato en la memoria el contador deje de contar. A modo de ejemplo servirá para generar datos que se guardarán en memoria.


```
entity contador is
   Port ( clk : in  STD_LOGIC;
          wr : in std_logic;
       cuenta : out  STD_LOGIC_VECTOR (15 downto 0));
end contador;

architecture Behavioral of contador is
signal q: std_logic_vector(15 downto 0);
begin
    process(clk)
        begin
          if rising_edge(clk) then
                if wr='0' then
                    q <= q + 1;
                    else
                    q <= q;
                    end if;
              end if;
        end process;
   cuenta <= q;
end Behavioral;
```

Figura 2. Bloque generador de datos y su código VHDL.

2.Bloque de escritura en memoria

Es una máquina de estados en la cual la entrada *b* controla el proceso de escritura de un dato en la memoria, Figura 3. Si *b=0* la máquina de estados permanece en su estado *inicio* sin embargo si *b=1* indica que se desea escribir un dato en la memoria, entonces la línea de control *we* siendo activa en bajo cambia su estado de 1 a 0 para realizar el proceso de escritura del dato de la entrada *data_in* en la dirección de memoria especificada por *dir*. Considerando la señal de reloj *clk* de la tarjeta de 50MHz (periodo de 20ns) se cuenta con 4 estados de escritura (*escribe1* a *escribe4*) dando un total de 80ns para el tiempo de escritura, siendo acorde con el tiempo de acceso a memoria de la RAM conforme al fabricante. Finalmente concluye la escritura y regresa nuevamente al estado *inicio*.

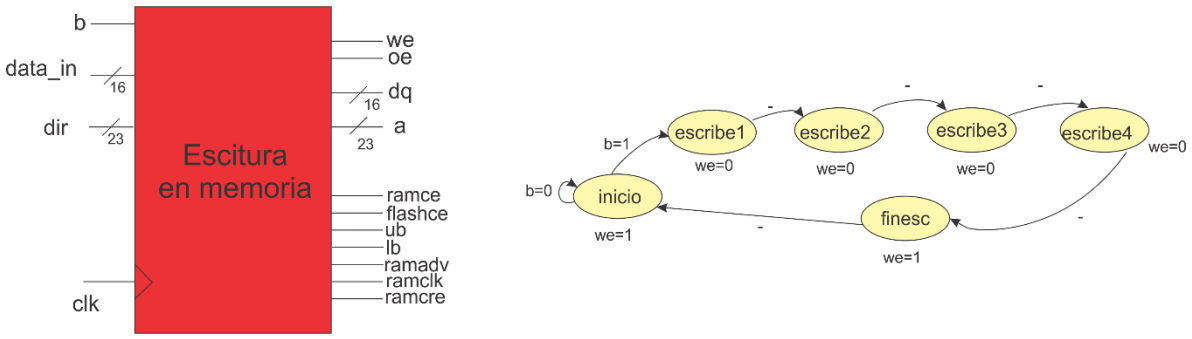


Figura 3. Bloque de escritura en memoria RAM y su máquina de estados.

3.Bloque de dirección y bloque de tiempo de muestreo

Tanto el bloque de la dirección de memoria como el bloque del tiempo de muestreo son simplemente contadores ascendentes (Figura 4) con señales de *reset* y habilitación de conteo *hab*, por lo tanto si *reset=1* el ambos contadores se van a cero, de lo contrario cada que llegue un flanco ascendente de la señal de reloj *clk* y la línea *hab=1* entonces los contadores se incrementarán en uno. Las líneas de *reset* y *hab* serán reguladas por el bloque controlador.

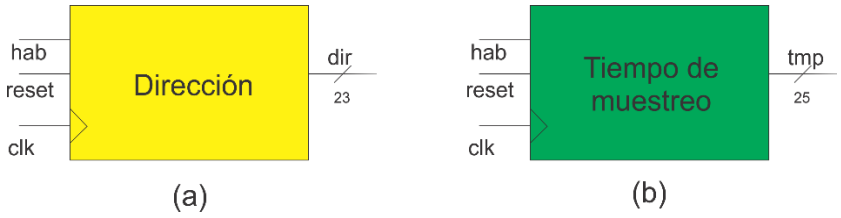


Figura 4. Bloque de dirección y bloque de tiempo de muestreo.

4.Bloque controlador

El bloque controlador es una máquina de estados cuya función es regular las tareas de todos los bloques del sistema de adquisición. Por ejemplo, controla la habilitación o deshabilitación del bloque dato, controla las líneas de habilitación y reset del bloque dirección y del bloque tiempo de muestreo, y le indica al bloque escritura en memoria cuando deberá escribir un dato en la memoria RAM (Figura 5).

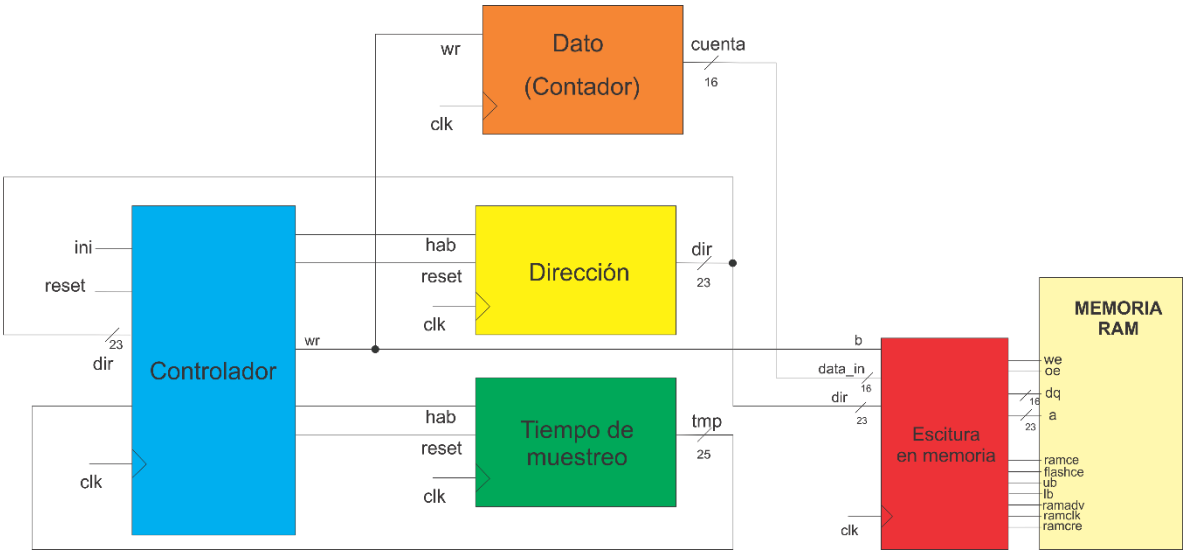


Figura 5. Interacción del Controlador con todos los bloques.

La máquina de estados general del controlador se muestra en la Figura 6. Como se puede observar se ha relacionado los colores de los bloques del sistema desarrollado con los colores

de los estados de la máquina a fin de relacionarlos. En el estado *inicio* se limpia (*reset*) el contador de direcciones y el contador del tiempo de muestreo.

En el estado de *espera* el controlador aguarda a que el usuario oprima un botón *(ini*) para inicial el proceso de adquisición de datos, cuando inicia se le indica al bloque de escritura que guarde un dato en la memoria (estado *escribe*), los estados siguientes (*espera2* a *espera7*) son para dar tiempo a que la memoria guarde el dato, posteriormente se habilita el contador del tiempo de muestreo en el estado *tmuestreo*, el contador empieza su conteo el cual verifica con un valor que es el tiempo de muestreo definido por el usuario, mientras no se llegue a ese tiempo, entonces la máquina de estados permanece en *tmuestreo* de lo contrario brinca a *incdir*. En *incdir* se habilita el contador de direcciones a fin de incrementar en uno la dirección de memoria y evitar que se vaya a sobreescribir un dato, en éste estado también se verifica si el conteo de *dir* ha llegado al número de muestras definidas por el usuario, si no es así, brinca a *escribe* para realizar la escritura del siguiente dato y así continuar sucesivamente el proceso, pero si se ha llegado al numero de muestras deseado, entonces se brinca a un estado de *stop*, donde el proceso de adquisición de datos termina.

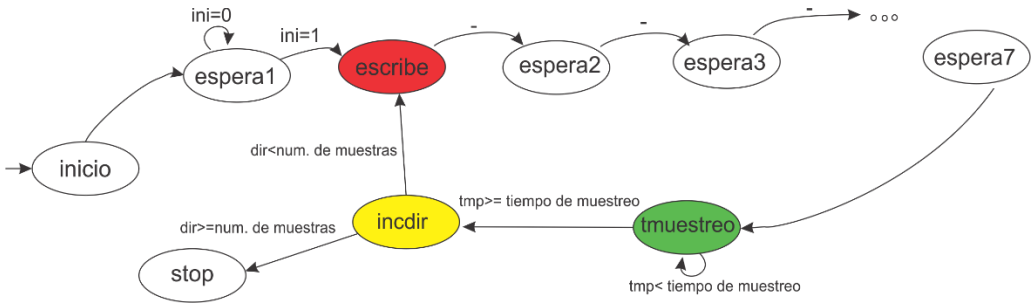


Figura 6. Diagrama de estados general del controlador.

Lectura y exportación de los datos

Una vez programado el FPGA y habiendo realizado el proceso de adquisición de datos, utilizando el programa ADEPT propio de la tarjeta Nexys2 en la ventana de *Memory* (Figura 7) se puede leer la memoria RAM y guardar los datos en un archivo binario.

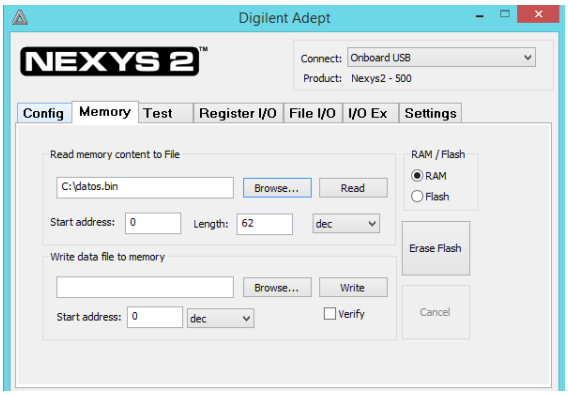
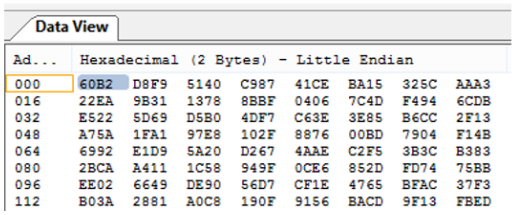
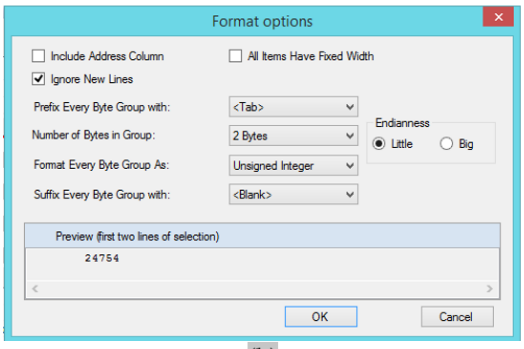


Figura 7. Uso del programa ADEPT para guardar datos en un archivo binario.

A modo de ejemplo se guardan los datos en un archivo llamados *datos.bin*, y se leen 62 datos de memoria. Para poder ver los datos, se utilizan programas visualizadores de datos binarios, por ejemplo el *Binary Viewer*, donde es posible ver los datos (Figura 8a) y exportarlos con un determinado formato (Figura 8b) a fin de poderlos trabajar posteriormente en una hoja de cálculo (Figura 9).



(a)



(b)

Figura 8 (a) Visualización de los datos en Binary Viewer. (b)Formato para exportar datos.

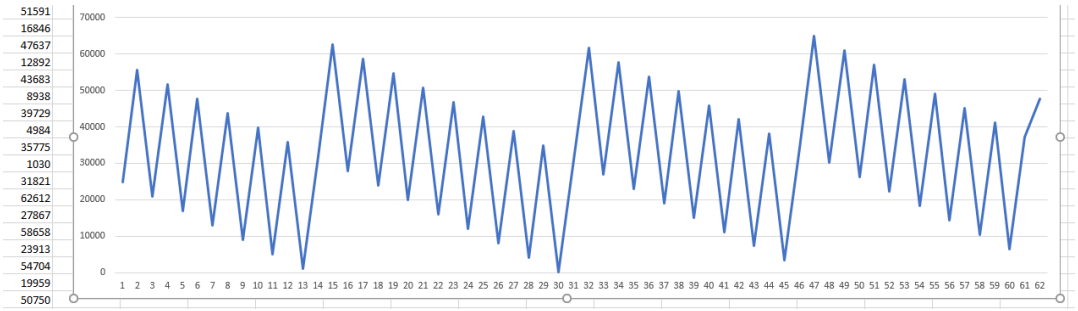


Figura 9. Datos adquiridos y graficados en hoja de cálculo.

III.Conclusiones

Es posible el uso de la tarjeta Nexys2 como un sistema de adquisición de datos (DAQ) a partir de describir bloques digitales simples en lenguaje descriptivo VHDL. Representa una alternativa a los DAQ comerciales con una arquitectura rígida, al poderse modificar conforme a las necesidades del usuario en el caso múltiples sensores, número de muestras y tiempos de muestreo. Las limitantes del sistema se dan principalmente por la capacidad de la memoria RAM (8 Mbytes x 16 bits) y el tiempo de escritura de memoria que es aproximadamente de 80ns.

IV.Referencias

Digilent Inc (2014). Nexys 2 Reference manual, http://www.digileninc.com

Haskel, R. E. & Hanna D. M. (2009) Learning by Example Using VHDL. Advanced Digital Design. Rochester, MI LBE Books.

Vahid, F. (2010) Digital Design with RTL Design. (2a Ed). John Wiley & Sons Inc.