Práctica 3

Objetivos

- Uso de módulos parametrizables de Altera (LPM) para el diseño de circuitos digitales más complejos.
- Familiarización con el uso de "In-System Memory Content Editor"
- Realización hardware de un algoritmo de procesamiento de señales sencillo.
- Familiarizarse con el uso de las herramientas de análisis de tiempos
- Experimentar técnicas que permiten aumentar la frecuencia máxima de funcionamiento de un circuito.

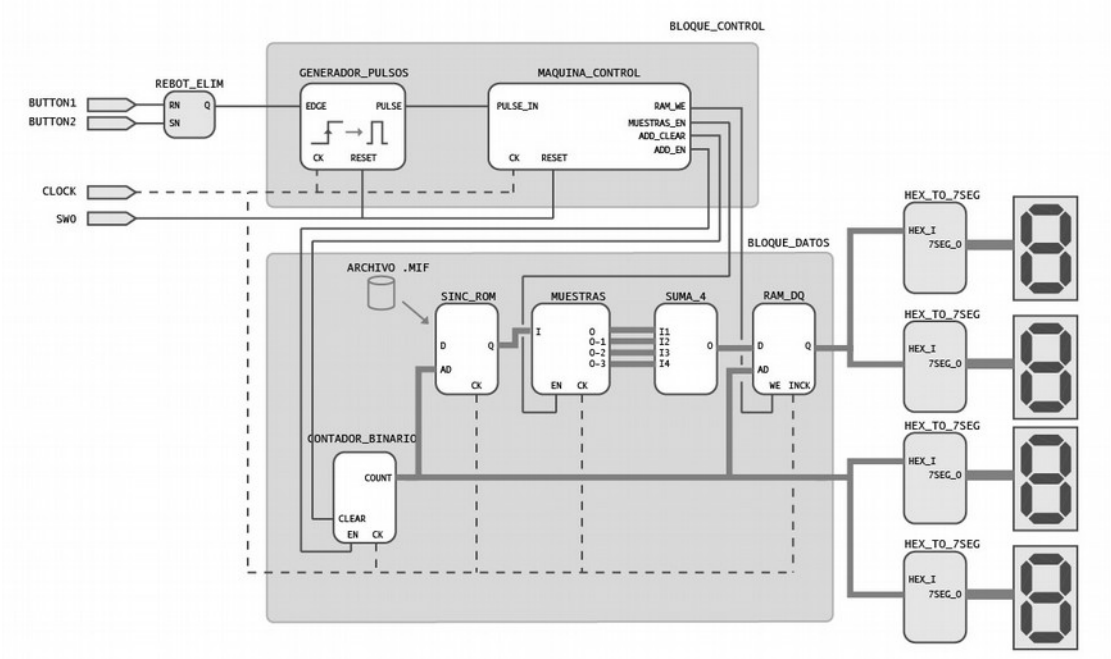
Parte 1

Descripción general del problema

El circuito a diseñar consiste en un filtro pasabajos que procesa una señal muestreada de acuerdo al siguiente algoritmo:

$$Yn = 1/2\ Xn + 1/4\ Xn\text{-}1 + 1/8\ Xn\text{-}2 + 1/8\ Xn\text{-}3$$

El flujo de muestras, de 8 bits de ancho, se simulará leyéndolas desde un bloque de memoria ROM (SINC_ROM) y los resultados deben escribirse en otro bloque de memoria RAM (RAM_DQ). Ambas memorias se recorrerán en forma cíclica.
Los bloques de memoria se construirán sobre los bloques de memoria M9K presentes en el chip EP3C16 de la placa DE0. Esto es realizado automáticamente por el Quartus si se definen las memorias como instanciaciones del bloque parametrizable altsyncram que se indica más adelante. Las muestras de entrada estarán definidas en un archivo .MIF y se cargarán en la memoria ROM en el momento de la configuración del chip.



El circuito se estructurará en 2 grandes partes: Bloque de Datos y de Control.

Bloque de Datos

En el bloque de datos se encuentran las memorias mencionadas anteriormente:

- SINC_ROM: basada en una altsyncram (en modo ROM) con la asignación de parámetros correspondientes para generar una ROM síncrona.
- RAM_DQ: basada en una altsyncram (en modo RAM) con la asignación de parámetros correspondientes para generar una RAM con salida asíncrona y escritura síncrona habilitada por la señal WE

También se encuentra un contador para generar las direcciones y recorrer las memorias en forma cíclica. El contador debe incrementar su salida si en el flanco de reloj EN=1 y debe llevar sus salidas a 0 en forma sincrónica con CLEAR=1.

El bloque **muestras** almacena las muestras anteriores que intervienen en el cálculo de la salida y el bloque **suma4** las suma, previa división por el valor correspondiente, para obtener el resultado que debe almacenarse en la memoria de salida. La salida O es igual a la entrada I en todo momento.

Bloque de Control

En el bloque de control se encuentra la máquina de control. El funcionamiento de ésta deberá ser el siguiente:

- En estado de reset (RESET=1) debe deshabilitar todas las señales de ENABLE del bloque de datos y resetear el contador.
- Una vez que se desactiva RESET, debe manejar las señales de ENABLE del bloque de datos para recorrer en forma cíclica las memorias y almacenar los resultados del algoritmo.
- Se deberá recorrer las memorias en forma cíclica hasta que se reciba un pulso en la entrada PULSE_IN. En ese instante se deberá dejar de ejecutar el algoritmo y se deberá llevar el contador de direcciones a cero.
- A partir de ese momento, cada vez que se reciba un pulso se incrementará en 1 el valor del contador. Esto será utilizado para observar los resultados del cálculo almacenados en la memoria RAM.

El bloque generador pulsos se encarga de generar un pulso de un período de reloj cada vez que llega un flanco en su entrada EDGE.

Se pide:

a) Diseñar y simular el bloque generador de pulsos.
b) Diseñar y simular el bloque de datos.
c) Diseñar el bloque de control como una máquina de estados que genere las señales de habilitación y borrado indicadas en el diagrama.
d) Probar el sistema diseñado en la placa. Para esto se deberá cargar en la ROM que contiene las muestras de entrada una señal escalón. Esto puede hacerse suponiendo que la señal comienza en 0 y luego salta a un determinado valor y se mantiene, para este caso las primeras direcciones de memoria contendrían 0 como dato y el resto de la memoria el valor de la señal. Utilizar el reloj de 50MHz de la placa como reloj del sistema.

Parte 2 (para hacer en clase)

Descripción general

En esta parte de la práctica, se modificará el algoritmo utilizado en el circuito anterior.

Para el nuevo sistema se determinará la frecuencia máxima de reloj que asegure un correcto funcionamiento.

Luego se probará en hardware el sistema para frecuencias iguales y superiores a la frecuencia máxima de reloj hallada.

Finalmente se analizarán técnicas que permitan aumentar la frecuencia máxima de funcionamiento.

Modificación del algoritmo

Modificar el circuito realizado en la parte uno, para implementar el siguiente algoritmo:

$$Yn = 1/3\ Xn + 1/5\ Xn\text{-}1 + 1/7\ Xn\text{-}2 + 1/7\ Xn\text{-}3$$

Utilizar el módulo lpm_divide de Altera para implementar las divisiones.

Análisis de tiempos y pruebas a frecuencia máxima

Utilizar las herramientas de análisis de tiempos para determinar la frecuencia máxima de funcionamiento del nuevo sistema.

Para la prueba en hardware a diferentes frecuencias de reloj, se deberá utilizar el módulo altpll de Altera.

El módulo altpll permite generar a partir del reloj de la placa, una señal de reloj de frecuencia menor o mayor.

Aumento de frecuencia máxima de funcionamiento

Se deberán proponer modificaciones a la implementación del sistema de forma de aumentar la frecuencia máxima de funcionamiento.