

Divisor de Frecuencia Parametrizable

Andrés Amorocho, Sebastián Gómez

Universidad Nacional de Colombia - Sede Bogotá

afamorochom@unal.edu.co – Cód. 285896

jsgomezj@unal.edu.co – Cód. 285908

Resumen— El siguiente informe, muestra la descripción del proceso de diseño e implementación del código en Verilog, para un divisor de frecuencia parametrizable y versátil que permite obtener una gama lo suficientemente amplia en frecuencias de salida con una señal de entrada fija; adicionalmente se muestran algunos de los resultados de las simulaciones.

Palabras clave: Divisor de frecuencia parametrizable, frecuencia de entrada y salida

I. INTRODUCCIÓN

Los sistemas digitales y su implementación requieren, en procesos de sincronismo, adaptación de velocidad de trabajo y transmisión de datos, entre otras aplicaciones más simples, hacer uso de los divisores de frecuencia que los hagan compatibles; adicionalmente se desea crear familiaridad con los lenguajes de programación que son soportados y que pueden implementarse en aplicaciones generales en la FPGA, además su compatibilidad con la articulación con otros módulos nos permitirá hacer reciclaje computacional. En este sentido realizar un divisor de frecuencia parametrizable nos aporta versatilidad en diferentes formas de trabajo, por esta razón a continuación se presentan las generalidades de los diseños preliminares y finales de un divisor de frecuencia parametrizable, para lograr este objetivo hacemos uso de herramientas de creación de módulos en verilog que son ofrecidos por xilinx.

II. GENERALIDADES DEL DISEÑO

En primer lugar se presenta el diseño del primer divisor de frecuencias cuyos cálculos están basados en esta misma, y aunque su diseño no es muy eficiente por la forma en que deben definirse las variables, se introduce a grandes rasgos con el fin de mostrar el proceso de retroalimentación y corrección de errores.

Este primer diseño se realiza con la premisa de poder obtener frecuencias que sean múltiplos enteros de la frecuencia de entrada y si no es este el caso que se asigne el valor en la variable de conteo aproximada al mayor entero, en este sentido la mayor frecuencia que se puede obtener con este diseño será la mitad de la frecuencia de entrada ya que un contador de un solo bit dividirá la frecuencia en dos.

Este divisor contaba con una salida `clk_out` y con una frecuencia de entrada `clk`, el parámetro divisor estaba determinado como la frecuencia de entrada sobre dos veces la frecuencia de salida, el contador del divisor tendría la función de generar la señal de salida cada cierto número de ciclos, los cuales estarían determinados por la razón ya descrita, sin embargo como la señal de reloj se mantiene la misma cantidad de ciclos en alto que en bajo, se agregó a esta división el dos al divisor para que cuente los flancos de subida de la señal y así genere la señal en el tiempo indicado, además como el contador iniciaría desde cero la cuenta se realizaría

Hasta divisor menor 1 este número se calcula como sigue, también se muestra las señales de entrada y salida para una frecuencia típica (ver figura 1).

$$\text{Bitcontador} = \frac{\ln(\text{divisor} - 1)}{\ln(2)}$$



Figura 1- divisor de 50(rojo) a 25(azul) MHz

El segundo y diseño final está basado en los periodos de cada señal, esto es más factible si consideramos que el conteo se realizara siempre para la división entre el periodo de la señal de entrada con respecto a la salida, de manera y gracias a que la señal de salida siempre presentara un periodo mayor que el de la frecuencia de entrada el resultado siempre será mayor que 1.

El diseño de este módulo, al igual que el anterior se inicia con la declaración de las variables de entrada y salida, clk y clk_out respectivamente, otra de las partes comunes es la creación de un contador, en este caso usamos un registro, que es como un vector de n bits, que sirve para almacenar el valor que cuenta los flancos de subida o bajada de la señal de entrada y que por lo tanto será el que determine el momento en que deba ser cambiado el estado de la señal de salida, este registro puede crearse del tamaño en el que se desee, lo único que debe tenerse en cuenta es que debe contar con los bits suficientes para almacenar el valor del contador, sin embargo como una premisa de diseño o para quien lo use posteriormente puede crearse de los bits que el usuario considere necesarios para los usos que se le pretenda dar al módulo.

Seguidamente se crean dos parámetros, usando el comando *parameter*, a los que se les asigna los valores de entrada y salida de los periodos, y se los nombra como Ts y Te, los cuales representan el periodo de la señal de reloj de salida y entrada respectivamente; uno de los beneficios que sobreviene a estas alturas es que si tomamos los dos periodos en las mismas unidades los números pueden escribirse adimensionales ya que en procesos posteriores sus unidades serían canceladas, a continuación se crea el parámetro n que es la división del periodo de salida con 2 veces la entrada menos 1 la entrada y sirve para cambiar de estado la señal de salida.

El primer ciclo *always* que se crea y se comienza el conteo, con el condicional se prueba si el contador es igual a n, si es así el contador se pone en cero o 1 e indica que hay un cambio de estado en la variable de salida, sino se suma uno al contador y se continúa la iteración.

El diseño de este módulo es flexible, sin embargo no permite obtener cualquier valor en la señal de salida, ya que si el parámetro n es racional no podrá ser correctamente asignado y el módulo fallará, de manera que se deja como premisa la aproximación al mayor entero en caso de que esto suceda.

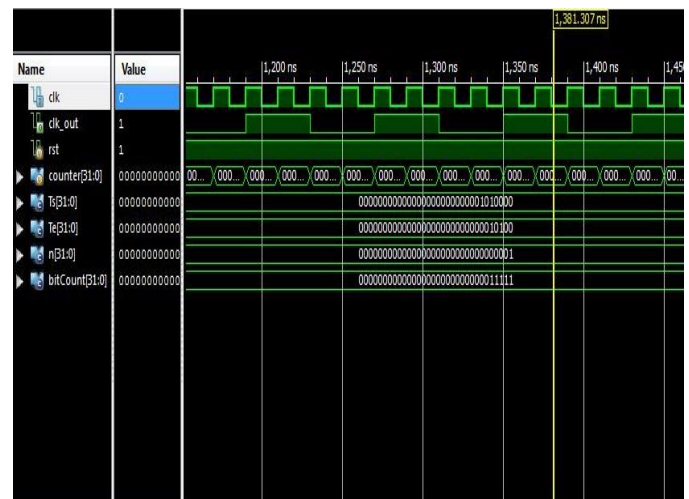


Figura 2. Divisor de un cuarto de la frecuencia de entrada.