Tarea 2: Diseño y verificación de un divisor secuencial basado en sumas y desplazamientos

Bernardo Enguix Chordá, Marcos Ibáñez Fandos, Salvador Mari Selfa, Arnau Mora Gras, Julia Navarro Vicent, Carlos Villena Jiménez

# **INTRODUCCIÓN**

L a tarea consiste en diseñar y verificar un divisor binario. Con el objetivo de:

Diseño Etapa RTL:

1. Diseño Componente Data-path
2. Diseño Componente Control-path
3. Descripción del sistema y verificación funcional

Diseño Verificación:

1. Simulación. Compilación del sistema y simulación lógica
2. Verificación lógica BÁSICA del diseño realizado
3. Verificación intermedia y avanzada

# **ETAPA RTL. DESCRIPCIÓN DEL SISTEMA COMPLETO Y VERIFICACIÓN FUNCIONAL.**

#### 1.1 Descripción del nivel de jerarquía top.

Nuestro diseño de nivel jerárquico top será el Divisor\_Algoritmico:

El tamaño de Números de entrada sería 32 bits.

Diagrama, Texto

Descripción generada automáticamente

Fig.1 Bloque Divisor

Puertos de entrada y salida:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Fig.2 Puertos

Diagrama

Descripción generada automáticamenteEl divisor de complementos que vamos a implementar estará basado en el código ASM dado en la tarea. Vamos a hacer una implementación directa del ASM mediante un único fichero systemVerilog, que implementará de forma única tanto el Control-Path como el Data-Path.

Este ASM tendrá 4 estados.

* **Estado 1** - Standby.
* **Estado 2** - Actualización de los valores.
* **Estado 3** - Operación de suma.
* **Estado 4** - Fin.

En ente bloque inicial podemos ver las señales externas de entrada y salida

correspondientes al acceso al divisor descritas en la Fig.2.

Texto

Descripción generada automáticamenteTambién tenemos los parámetros internos del Divisor.

Tenemos un contenedor del estado, **state**, en el que tendremos 4 constantes ya que hay 4 bloques; **S0**, **S1**, **S2**, **S3**.

Utilizaremos dos parámetros para el de valor de 2s complement; **mem** de 32 bits y **c2s**.

Para el cociente del contador tendremos las variables de 32 bits; **q** y **posDen**. También tenemos las variables **signNum** y **signDen** que indicaran el signo del numerador y denominador.

Al principio de este fichero systemVerilog vamos a implementar un FMS, que consiste en una celda altamente automatizada de Tecnologías de Grupos, que consiste de un grupo de estaciones de trabajo de procesos, interconectadas por un sistema automático de carga, almacenamiento y descarga de materiales.

#### CONTROL-PATH

El control path al final es quien organiza, administra y controla el estado.

La señal se sensibilidad del control path será la propia entrada start que se va a alimentar en su etapa inicial.

#### DATA-PATH

El data path al final es quien ejecuta todos los cambios en todas las variables existentes en el diseño.

#### Simulación del modelo RTL total generado.

#### Verificación del modelo RTL generado.

# **ETAPA LÓGICA. COMPILACIÓN DEL DIVISOR Y SIMULACIÓN LÓGICA.**

#### 2.1 Introducción

#### 2.2 Compilación del top (multipli) y verificación post-rutado.

#### 2.3 Obtención de prestaciones de vuestro diseño.

# **DISEÑO DEL DIVISOR SEGMENTADO**

#### 3.1 Realización del código RTL

#### 3.2 Verificación funcional.

#### 3.3 Compilación del diseño

# **VERIFICACIÓN EN LAS PLACAS DE PRUEBA. CONFIGURACIÓN DE LA FPGA (opcional).**