

TRABAJO PRÁCTICO 6

[66.17/86.41] Sistemas Digitales

Aritmética de punto flotante

Autor:

Franco Mariotti 102223

2do Cuatrimestre 2020

1 Objetivo

El objetivo del presente trabajo practico es el desarrollo de una unidad de multiplicación de punto flotante. Este desarrollo consta de una etapa de diseño de la arquitectura y descripción de la misma en VHDL, una etapa de simulación utilizando las herramientas GtkWave y GHDL y finalmente una etapa de síntesis e implementación utilizando la herramienta Vivado.

2 Desarrollo

2.1 Overflow

El lógica de overflow del multiplicador mira el cambio de signo cuando los exponentes son positivos, es decir si al sumar dos exponentes positivos terminamos con uno negativo, entonces el bit de overflow se pone en uno.

2.2 Underflow

La lógica de underflow del multiplicador mira el cambio de signo cuando los exponentes son negativos, es decir si al sumar dos exponentes negativos terminamos con uno positivo, entonces el bit de underflow se pone en uno. También chequea que si el exponente resultante es negativo, que no sea menor al mínimo exponente permitido.

2.3 Zero

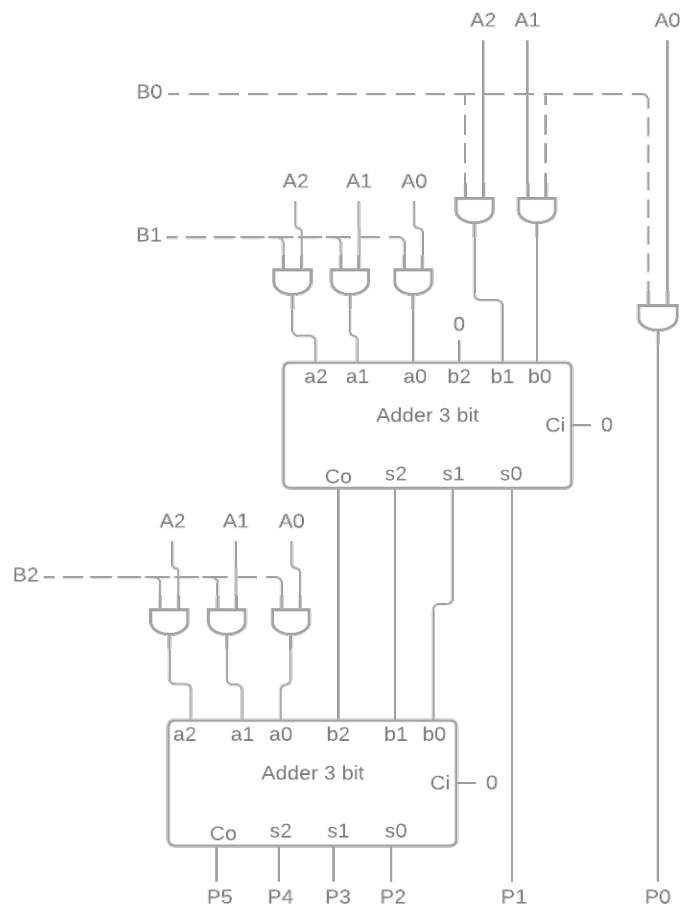
El multiplicador cuenta con un flag de Zero, cuando algunas de las entradas A o B es cero el flag se pone en 1 y la salida del multiplicador pasa a ser cero.

2.4 Infinity

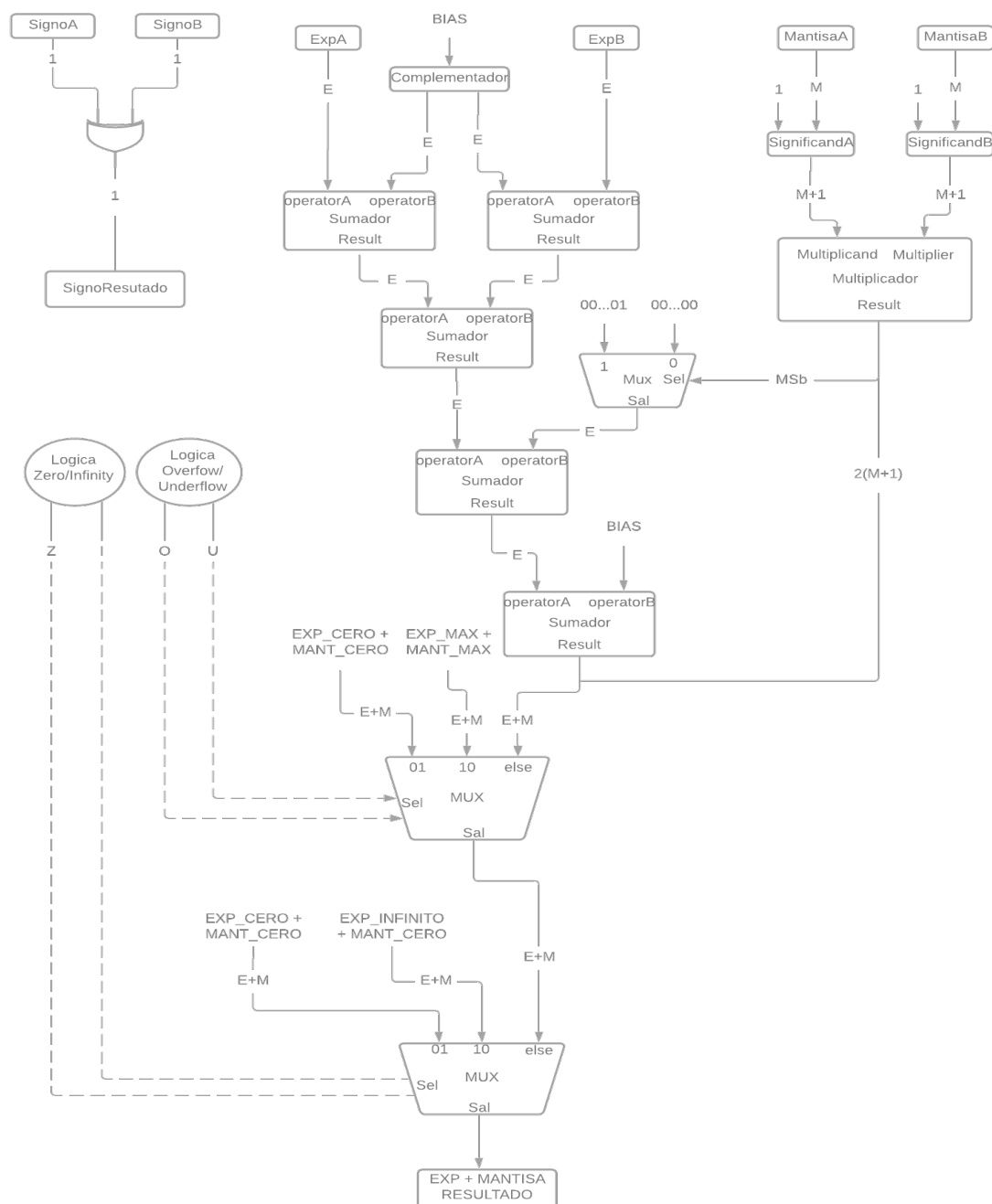
El multiplicador cuenta con un flag de Infinity, cuando algunas de las entradas A o B es infinito el flag se pone en 1 y la salida del multiplicador pasa a ser infinito.

2.5 Multiplicador N bits

El multiplicador de N bits se utiliza para la multiplicación entera de los significands. El mismo se implemento como un conjunto de FAs(Full Adders) puestos en cascada. A continuación se muestra un ejemplo de como quedaría la arquitectura cuando los números a multiplicar son de 3 bits.

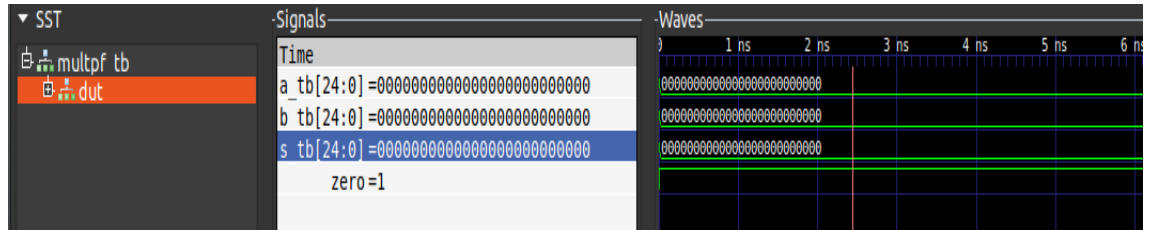


2.6 Multiplicador Punto Flotante de N bits

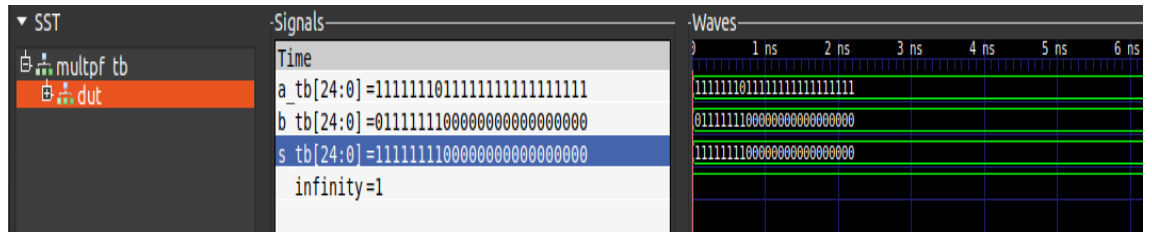


3 Simulaciones

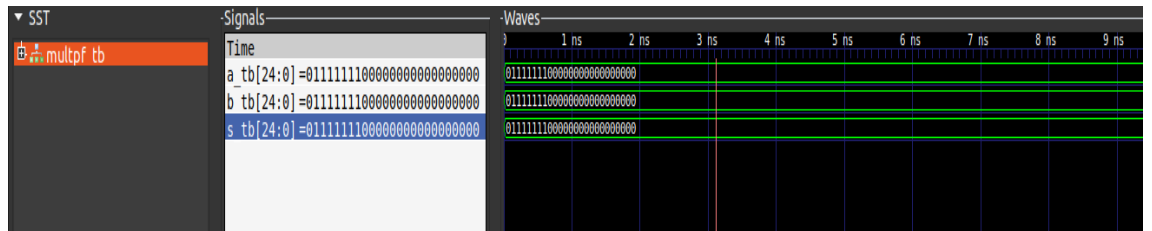
3.1 Caso: producto de operadores nulos



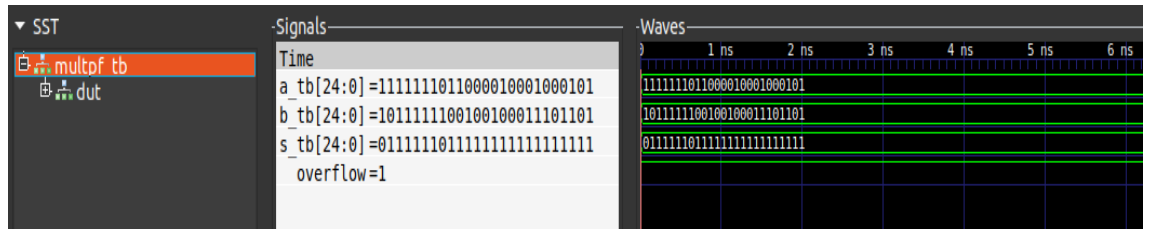
3.2 Caso: producto de operador por infinito



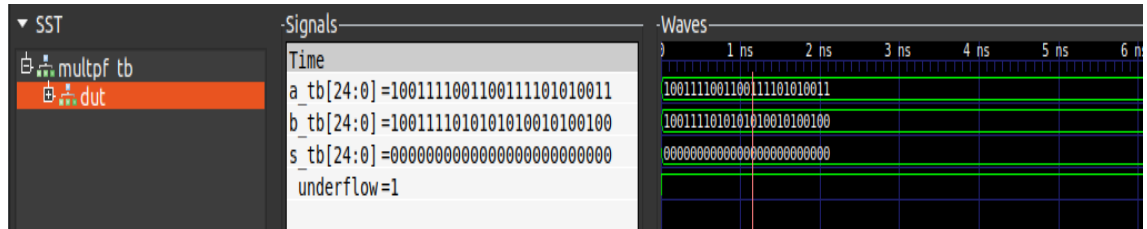
3.3 Caso: infinito por infinito



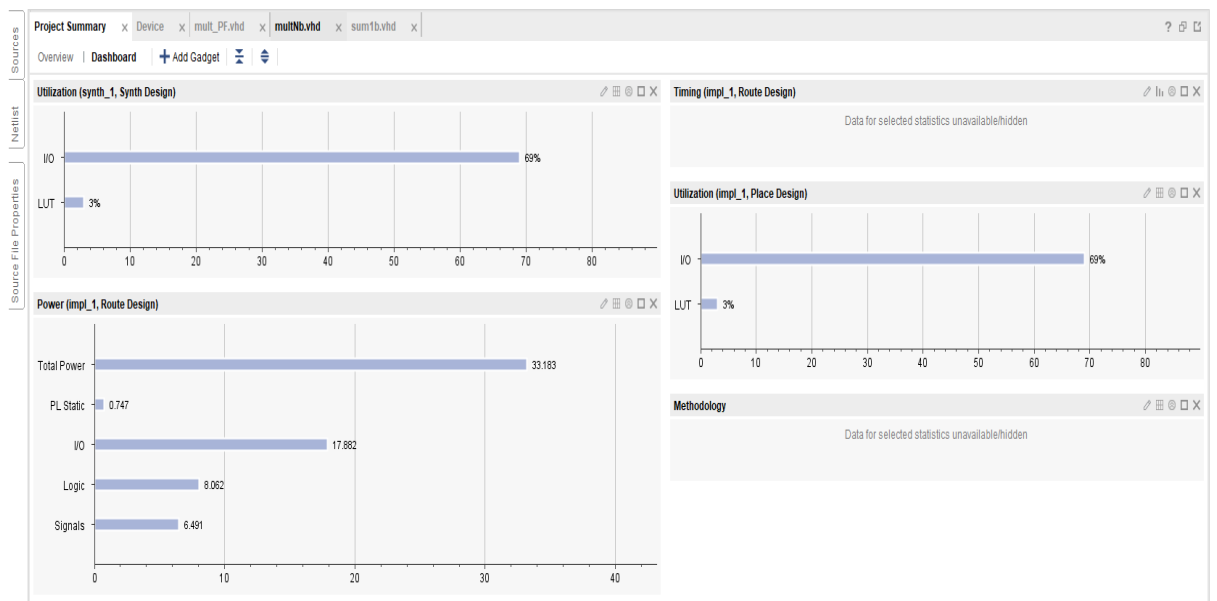
3.4 Caso: Overflow



3.5 Caso: Underflow



4 Síntesis



5 Código

Repositorio de Github:

<https://github.com/FrancoMariotti/SistemasDigitales/tree/main/tps/tp6-final>