

Axi4-Stream De-Mux

Implementación

Objetivo

- Diseñar un módulo demultiplexor del bus Axi4-Stream.

HDL

- VHDL

Descripción de la arquitectura

- *Gen Data & Ctrl*
 - Genera muestras de 16 bits
 - Dos muestras se concatenan para conformar el bus de salida TDATA de 32 bits.
 - Genera las señales de control para el funcionamiento del módulo Axi4-Stream Data FIFO
- *Axi4-Stream DF0 (32 bits)*
 - IP: Axi4-Stream Data FIFO
 - Señales de configuración
 - FIFO Depth: 16
 - TDATA: 32 bits
 - Analizar si las señales de control restantes son necesarias para la comunicación.
- *De-Mux*
 - Separa el bus TDATA de 32 bits a 2x16 bits
- *Axi4-Stream DF1 & DF2 (16 bits)*
 - Señales de configuración
 - FIFO Depth: 16
 - TDATA: 16 bits
 - Analizar si las señales de control restantes son necesarias para la comunicación.
- *Checker0*
 - Compara los 16 LSB ([15:0]) transmitidos y recibidos
- *Checker1*
 - Compara los 16 MSB ([31:16]) transmitidos y recibidos

Funcionamiento

- El módulo Gen Data & Ctrl genera datos que son transmitidos por el bus TDATA y las señales de control del módulo Axi4-Stream
- Los datos se generan con dos contadores de 16 bits no signados. El paso del contador0 es 1 y el paso del contador1 es -1.
- Una vez iniciada la transmisión, el módulo De-Mux separa el bus TDATA en dos buses de 16 bits
- Finalmente, los módulos checkers comparan las muestras transmitidas con las recibidas.

Entregable

- Se debe entregar el módulo Axi4-Stream De-Mux como un IP.
- Un módulo Top que instancie todo el diseño.
- Justificar su funcionamiento por medio de simulaciones (Run Behavioral Simulation).
- Generar los reportes de timing para una frecuencia de referencia de 100 MHz.
- Entregar los archivos VHDL necesarios para verificar el funcionamiento y los reportes de timing.

