

Prototipado de micro-controlador DSP

Arquitectura y diseño

1. Diagrama de Arquitectura

El siguiente diagrama muestra el diagrama general del sistema a desarrollar. El área gris contiene la lógica implementada en el FPGA. Por la cantidad de memoria requerida, se determina el uso de memoria externa como parte de una tarjeta de desarrollo. Esta tarjeta se define con el área de color azul. Las interfaces SPI e I2C se utilizan para comunicarse con periféricos en circuitos externos.

Para prototipar los ADC y DAC, se define el uso de una tarjeta externa, con comunicación I2C, que se comunica con la tarjeta de desarrollo y la FPGA.

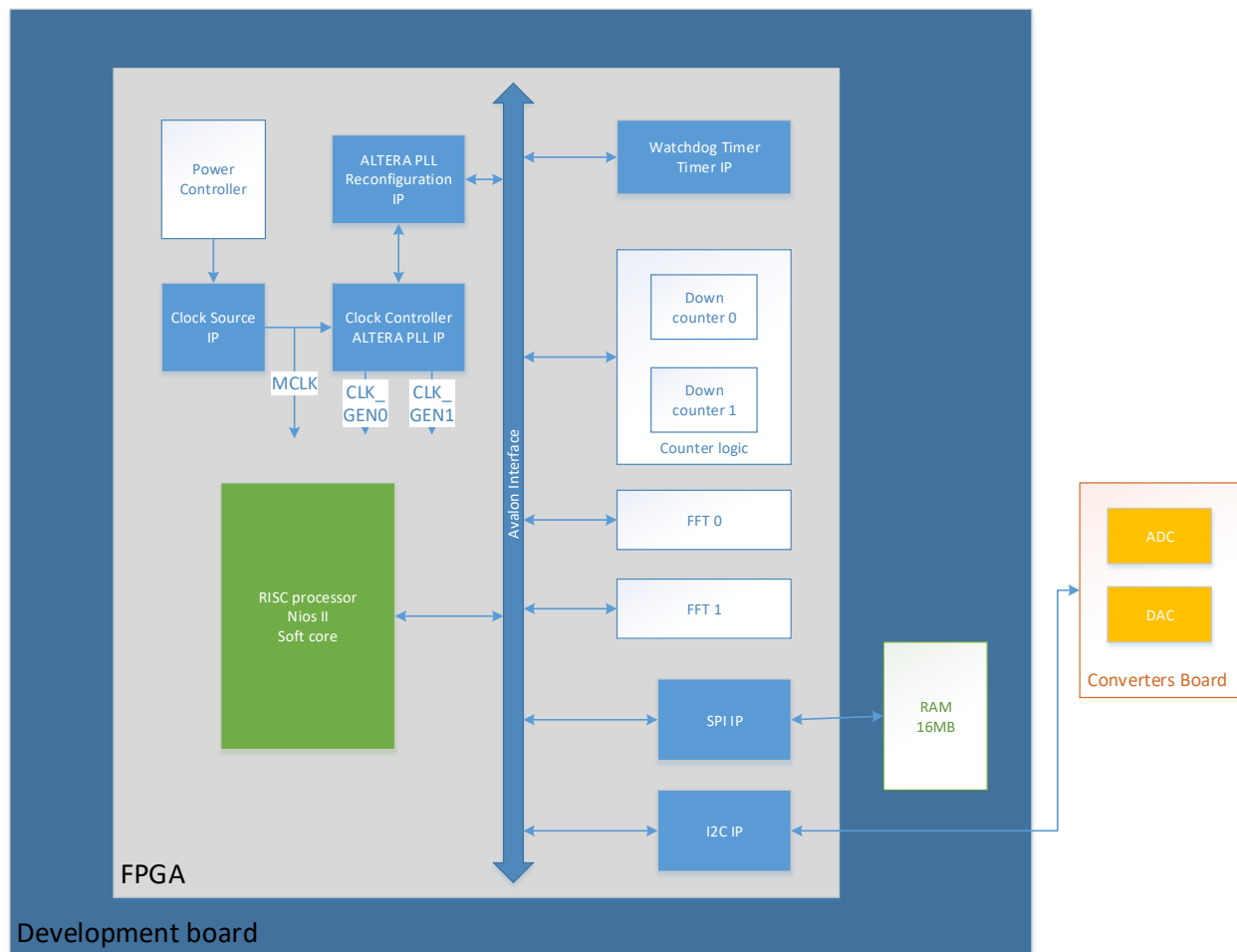


Figura 1. Diagrama de arquitectura del sistema

2. Diseño de Sistema

Se detalla a continuación el diseño de cada uno de los bloques del sistema y la descripción de su implementación. Todos los IP utilizados para el diseño son parte de las librerías gratuitas del paquete de Quartus II Lite 16.0. No se requirió utilizar cores de IP externos o de terceros.

1. Procesador RISC

El procesador RISC para el ASIC se reutilizará de un IP de la empresa. Para simular este procesador en el prototipo de FPGA, se utiliza un Soft core del procesador Nios II que también es RISC. Se debe tomar en cuenta que ya que el IP Core no es el mismo que el que sería utilizado en el ASIC, el consumo de recursos puede variar para este bloque. El uso del procesador Nios II permite tener una aproximación de los recursos a utilizar.

2. Periféricos:

a. Controlador de reloj

Para implementar este controlador, se utiliza una combinación de dos IPs: el ALTERA PLL, el cual permite generar relojes a partir de un reloj de entrada, aplicando factores de multiplicación o división, tal como es requerido. El segundo es el IP de Altera PLL Reconfig, el cual se utiliza para poder reconfigurar dinámicamente el PLL, controlado mediante la interfaz de Avalon. Éste IP controla el PLL para poder modificar en tiempo de ejecución los factores de multiplicación/división de los relojes generados.

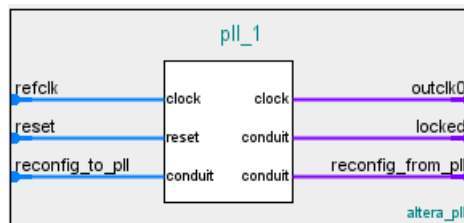


Figura 2. Bloque IP Altera PLL con opción de reconfiguración

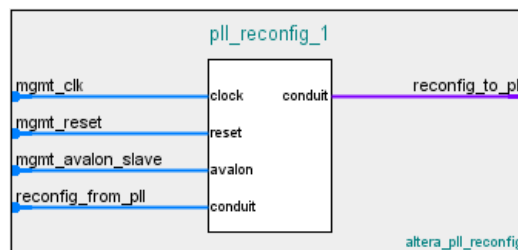


Figura 3. Bloque IP Altera PLL Reconfig interfaz Avalon

b. Contador decremental

Este bloque consta de dos contadores decrementales. Estos bloques se implementan mediante código HDL, con ayuda de la plantilla de up/down binary counter del Quartus

II. La lógica de interrupción e inicialización se agrega al código HDL y se crea un bloque personalizado que pueda ser instanciado mediante Qsys e integrado en el bus de Avalon.

c. Temporizador Watchdog

Se puede implementar mediante el IP de Interval Timer, en operación de WDT, con período de 30 segundos a 50MHz.

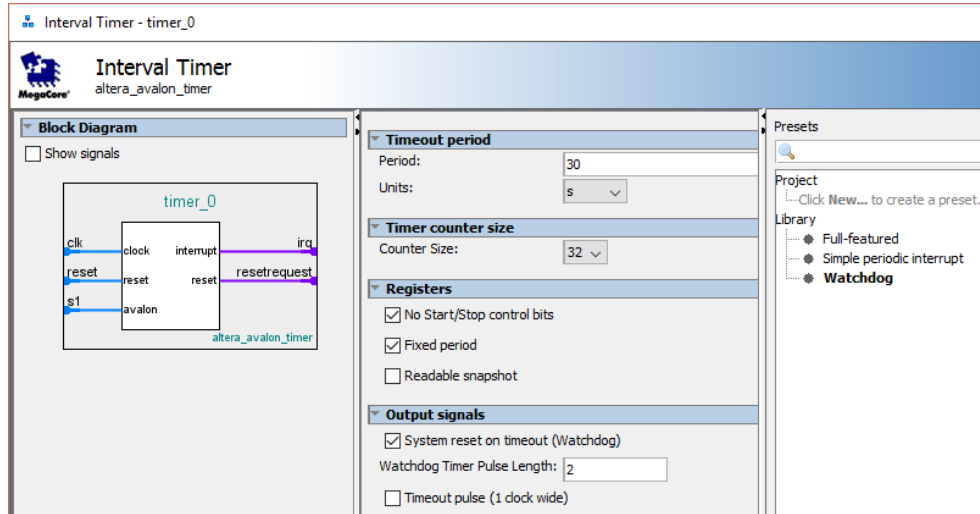


Figura 4. Configuración del bloque IP Interval Timer como función Watchdog

d. Bloque FFT

Se implementa mediante el IP de FFT de la librería. Es necesario utilizar otro IP para adaptar la interfaz Avalon a una interfaz de streaming que permita la transferencia de paquetes de datos a través del IP de FFT. Este IP se llama “Avalon Packets to Transaction Converter”.

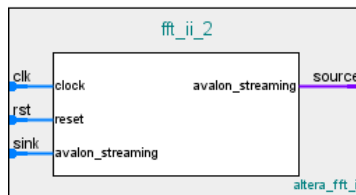


Figura 5. Bloque IP para FFT con interfaz Avalon Streaming

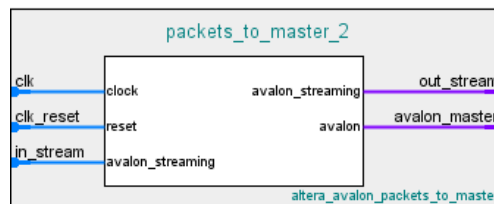


Figura 6. Bloque IP de convertidor de paquetes, para conectar interfaz Avalon Streaming con Avalon Master

e. ADC y DAC

Ya que estos componentes no forman parte de la FPGA ni de la tarjeta de desarrollo elegida, se utiliza la tarjeta PCF8591 con convertidores, con interfaz I2C. En el testbench, es posible simular el comportamiento tanto del ADC como el DAC, para comprobar la lógica de comunicación y el software que lo utiliza. La frecuencia de muestreo depende de la configuración del controlador I2C, manejado por el sistema en FPGA.

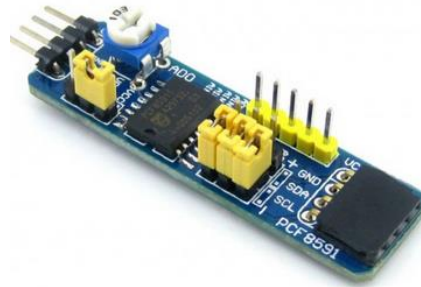


Figura 7. Tarjeta PFC8591 con convertidores AD y DA de 8-bit por I2C.

f. SPI y I2C

Se utilizan los IPs respectivos de SPI y I2C de la librería gratuita.

g. Memoria en chip

Al requerir 16MB de memoria, se decide utilizar un chip externo de memoria RAM de un solo acceso, tipo SDRAM, parte de la tarjeta de desarrollo, y con comunicación I2C o SPI. Muchas de las tarjetas de desarrollo, útiles para prototipar sistemas en FPGA, tienen memorias SDRAM incluidas desde 16MB hasta 1GB.

h. Controlador de energía

Este bloque se puede prototipar de manera parcial mediante código HDL. Los modos de operación “activo” y “P1” se pueden implementar, mientras que el modo “P2” es imposible de implementar por software, ya que requiere de un hardware que permita levantar el dispositivo cuando el procesador y todos los periféricos se encuentran apagados.

i. Bancos de I/O

No es necesario prototipar los bancos de I/O, a menos que se dedique alguna entrada o salida para probar alguna funcionalidad del Sistema.