

IP core Puente USB a WISHBONE



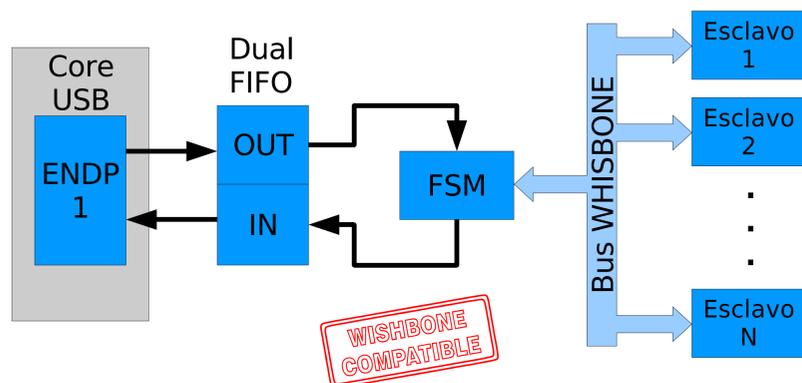
Rodrigo A. Melo, Salvador E. Tropea
(rmelo@inti.gov.ar, salvador@inti.gov.ar)

Instituto Nacional de Tecnología Industrial
Electrónica e Informática



Introducción

Se desarrollo un *core* puente entre USB y WISHBONE. Es un maestro WISHBONE controlado mediante USB. Permite validar en *hardware* descripciones a través del bus WISHBONE o realizar periféricos abstrayéndonos de las dificultades USB.



Motivación

- Puertos serie y paralelo desplazados por USB.
- Desarrollo de sistemas embebidos que precisan comunicación con la PC para validación en *hardware*.
- Utilización de bus WISHBONE para interconexión entre *cores*.

Comandos implementados

Se implementaron comandos de lectura, escritura y control sobre WISHBONE. Se des/habilitan por grupos mediante *generics*.

Grupo	Descripción
básicos	funcionalidad mínima del <i>core</i>
autoincremento	des/habilitan incremento automático de dirección WISHBONE
demora	permite implementar demoras
directos	utilizan operandos recibidos como dirección o valor
repetición	repite operaciones
repetición con demoras	repite operaciones con demoras

Resultados

- Escrito en VHDL93 estándar, independiente de la tecnología.
- Se utilizaron herramientas recomendadas por proyecto FPGALibre.
- **Validado en *hardware*** utilizando FPGAs *Spartan II* y *Spartan 3* de Xilinx y *software* ISE WebPack 9.2.03i J.39.
- El *host* es una PC corriendo Debian GNU/Linux.

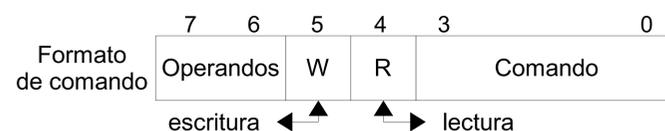
Se comparó el área ocupada para la mínima y máxima cantidad de comandos habilitados corriendo a FS.

Comandos	LUTs	FFs	Slices
básicos	677	385	476
todos	791	429	537



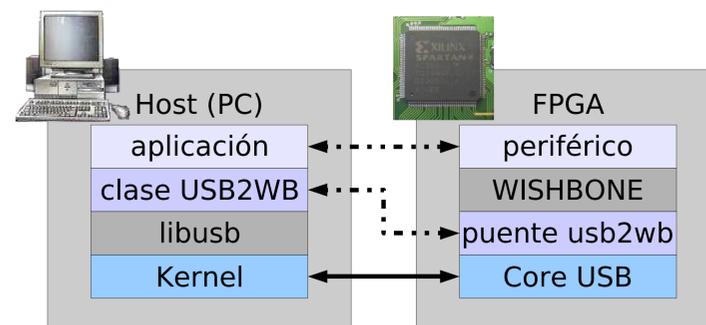
Arquitectura

- **Core USB:** resuelve la comunicación USB (*Full* o *High Speed*) brindando como interfaz el *endpoint 1*.
- **Dual FIFO:** dos *buffers* asociados a las direcciones *out* e *in* del *endpoint 1*. La FIFO *out* se inicia libre y acepta recibir comandos del *host*. Una vez llena y confirmados sus datos, pueden ser consumidos por el puente el cual los procesa y coloca los resultados en la FIFO *in*. Terminado el proceso de datos, *endpoint in* está disponible. El *host* retira los datos y FIFO *out* vuelve a estar disponible.
- **Máquina de estados:** se compone de 2 unidades.
 1. **Unidad de procesamiento:** encargada de dialogar con la *dual FIFO* e interpretar los comandos de 8 bits.
 2. **Unidad de ejecución:** lee los operandos y ejecuta los comandos sobre el bus WISHBONE.



Uso desde la PC

- **Clase USB2WB:** abstrae la funcionalidad en una **clase de C++**.
- **API USB2WB:** provee funciones para encolar, enviar y recibir datos. Estas implementan reintento en caso de *time-out* y utilizan *signals* para errores que no pueden solucionarse.



Conclusiones

- Implementación compacta que permite el uso de FPGAs pequeñas.
- Las herramientas propuestas por proyecto FPGALibre demostraron ser adecuadas para un proyecto de estas características.
- Se puede validar en *hardware* de manera rápida y sencilla *cores* WISHBONE utilizando una PC y *software* de alto nivel (C++).
- Alternativa rápida para desarrollo de dispositivos USB.
- La clase USB2WB resume la realización de un *software* de aplicación al uso de simples funciones de envío y recepción.

