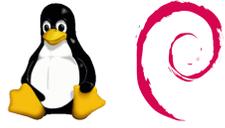




Microcontrolador compatible con AVR, interfaz de depuración y bus WISHBONE



Salvador E. Tropea, David M. Caruso

Instituto Nacional de Tecnología Industrial,
Electrónica e Informática, DESoL, Argentina



1. INTRODUCCIÓN

En este trabajo presentamos un microcontrolador compatible con la línea AVR de Atmel. Esta implementación puede ser configurada para ser compatible con los AVR de segunda (ej. ATtiny22), tercera (ej. ATmega103) y cuarta (ej. ATmega8) generación. Este desarrollo incluye los siguientes periféricos compatibles: controlador de interrupciones, puertos de entrada y salida, temporizadores y contadores, UART y watchdog. Para adaptarlo a distintas necesidades se incluyó una interfaz de expansión que utiliza el estándar de interconexión WISHBONE. A los fines de facilitar el desarrollo de aplicaciones sobre esta plataforma se lo dotó de una unidad de depuración y se adaptó el software necesario para poder realizar depuración de alto nivel con una interfaz de usuario simple e intuitiva. El diseño fue verificado usando simuladores y FPGAs de Xilinx (Spartan II y 3A).

2. OBJETIVOS

- Obtener un microcontrolador programable en C.
- Utilizar una arquitectura actualmente en uso para desarrollos embebidos en nuestro laboratorio.
- Poder depurar los desarrollos *in-circuit* y a nivel de fuentes.

3. MICROCONTROLADOR

3.1 Arquitectura

- RISC de 8 bits.
- Espacios de memoria de programa y datos separados.
- Instrucciones de 16 bits (algunas de 32 bits).
- 32 registros de 8 bits (6 pueden formar pares de 16 bits).
- 64 direcciones de I/O.
- Stack pointer (útil para compilar código C).
- 1 ciclo de reloj por instrucción (la mayoría, algunas hasta 4, ej. CALL).

3.2 Implementación

- Utilizando las herramientas recomendadas por el proyecto FPGALibre.
- VHDL93 estándar, independiente del tipo de FPGA
- Basado en el proyecto AVR Core de OpeCores.org.
- Soporte para las 2da, 3ra y 4ta generaciones de AVR.

3.3 Periféricos

- Controlador de interrupciones
- Puertos de entrada y salida
- Temporizador y contadores
- USART
- Watchdog

3.4 Bus de Expansión

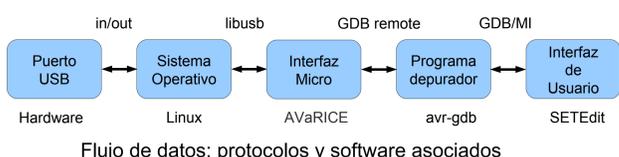
- Bajo la especificación abierta WISHBONE:
 - Para casos simples (un maestro y uno o más esclavos) se reduce a poca o ninguna lógica adicional.
 - Pensado para casos más complejos (más de un maestro, reintento, notificación de error, etc.).
 - No posee *royalties* y puede ser usado sin costo alguno. La especificación completa se encuentra disponible en internet.
- Usando los registros originalmente dedicados a la EEPROM (1E/1F).
- Hasta 256 registros de I/O.

3.5 Configuraciones Equivalentes

- **ATtiny22:** 2da generación, 1 puerto de I/O de 5 bits, 1 timer de 8 bits, WISHBONE, stack pointer de 8 bits, watchdog, una fuente de interrupción externa, 128 bytes de memoria RAM, 1024 words de memoria de programa.
- **ATmega103:** 3ra generación, 6 puertos de I/O, UART, WISHBONE, 2 timers de 8 bits, stack pointer de 16 bits, watchdog, 8 fuentes de interrupción externa, 4096 bytes de memoria RAM, hasta 65536 words de memoria de programa.
- **ATmega8:** 4ta generación, 3 puertos de I/O, UART, WISHBONE, 2 timers de 8 bits, stack pointer de 16 bits, watchdog, 2 fuentes de interrupción externa, 1024 bytes de memoria RAM, hasta 65536 words de memoria de programa.

4. HERRAMIENTAS DE DESARROLLO

- Se utilizaron herramientas de Software Libre gracias al alto grado de compatibilidad del core desarrollado.
- **Ensamblador:** avra (GPL) compatible con el ensamblador de Atmel.
 - **Compilador C/C++:** gcc-avr (GPL) excelente optimización, soporte de 2da a 5ta generación.
 - **Biblioteca Estándar de C:** avr-libc (modified Berkeley) muy compacta y completa
 - **Depurador:** avr-gdb (GPL) con la misma potencia que para depurar programas de PC, apto para depuración remota.
 - **Simulador:** simulavr (GPL) puede usarse desde avr-gdb.



5. INTERFAZ DE DEPURACIÓN

- Elemento clave en sistemas embebidos donde no se dispone de pantalla y/o teclado, con escasos recursos de memoria.
- Basada en un trabajo anterior de nuestro laboratorio, no compatible con la original de Atmel.
 - Conectable a un bus WISHBONE, controlada desde la PC usando USB.
 - Características principales:
 - Detener/Reanudar la ejecución del microcontrolador en cualquier momento.
 - Ejecutar su programa paso a paso.
 - Detener la ejecución cuando se alcanzó una posición de memoria determinada, punto de parada o breakpoint. La cantidad de breakpoints es configurable entre 1 y 256.
 - Reinicializar el microcontrolador.
 - Acceder a todos los registros, incluyendo el contador de programa.
 - Acceder al espacio de entrada y salida.
 - Inspeccionar la pila de llamadas (calling stack).
 - Detener la ejecución cuando se accede a una posición de memoria de datos, watchpoint. Los accesos pueden seleccionarse para detenerse por lectura, escritura o ambos. El número de watchpoints es configurable entre 1 y 256.
 - Alterar la memoria de programa.
 - Detectar desbordes en la pila y detener la ejecución cuando esto sucede.
 - Software de interfaz entre el depurador y el hardware: AVaRICE (GPL), modificado para soportar nuestra interfaz.
 - Software de interfaz entre el depurador y el usuario: SETEdit (GPL), recomendado por el proyecto FPGALibre.
 - Mecanismo para "grabar" la memoria de programa incluido en el sistema de depuración.

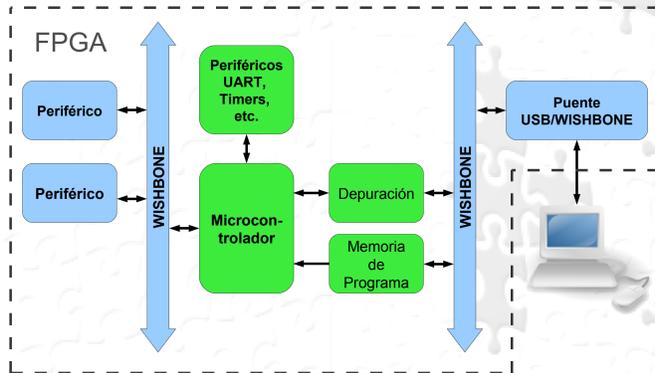
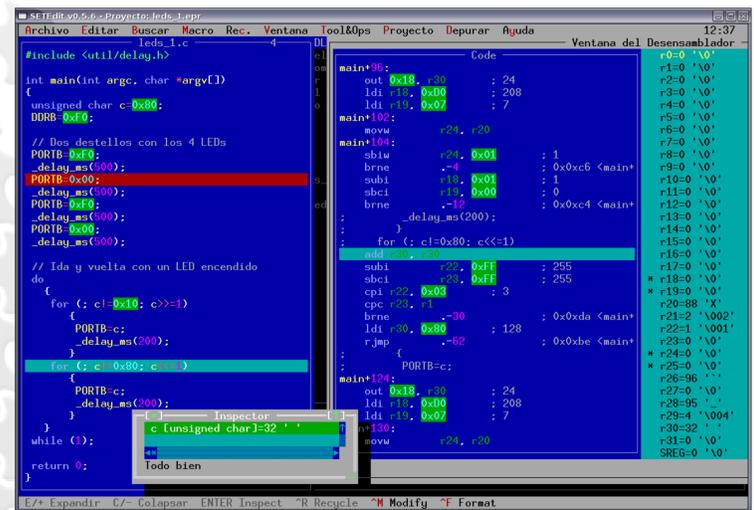


Diagrama de conexiones de los bloques de hardware



Sesión de depuración utilizando SETEdit. Código fuente, código desensamblado y variable monitorizada.

6. RESULTADOS

Verificado por simulación (GHDL) y usando FPGAs de Xilinx (Spartan II y 3A, ISE Webpack 10.1.02 K.37)

Configuraciones de prueba										
AVR	Datos	Programa	UART	I/O	FFs	LUTs	BRAMs	Debug	Spartan	Fmax [MHz]
3	4k	4k	SI	0	302	1153	3	NO	3	31
3	4k	1k	WB	0	245	1124	3	NO	3A	33
4 ¹	1k	1k	WB	0	275	1244	2	NO	3A	30
2	256	1k	WB	0	237	1053	2	NO	3A	37
4 ¹	1k	1k	WB	8	902	2477	4	USB	3A	25 ²

1 Usa un multiplicador
2 Frecuencia fijada a 24 MHz por el USB

7. CONCLUSIONES

- La elección de la arquitectura AVR permitió contar con muchas herramientas y bibliotecas. Se comprobó que programar este dispositivo es tan fácil como programar un AVR comercial.
- La unidad de depuración implementada es poderosa, capaz de realizar la mayor parte de las operaciones realizadas por depuradores usados en PCs, y es de gran ayuda a la hora de buscar errores en sistemas funcionando en tiempo real.
- El uso de FPGAs posee como ventaja el hecho de que la unidad de depuración puede removerse en la versión definitiva del diseño. En el caso en que el diseño ocupe prácticamente el total de la FPGA basta con usar una FPGA más grande durante la etapa de desarrollo.
- La selección del estándar de interconexión WISHBONE permitió el reuso de un puente de puerto USB y abre la posibilidad a la implementación de otro tipo de mecanismos de comunicación, como podrían ser RS-232 o Ethernet.
- La elección de modificar un programa como AVaRICE permitió acelerar notablemente el desarrollo y reusar interfaces de usuario ya existentes y con las cuales nuestro equipo ya se encontraba familiarizado.
- La utilización de las herramientas propuestas por el proyecto FPGALibre mostró ser adecuada para este desarrollo.