

Sistemas Electrónicos Digitales

Tema #4 Diseño mediante Lenguajes de Descripción Hardware

Parte 4.5

Índice

- 5.1 Ventajas de los HDL.
- 5.2 Metodología de Diseño.
- 5.3 VHDL. Sintaxis de VHDL.
- 5.4 Codificación de circuitos lógicos en VHDL.
- **5.5 Módulos IP.**
- 5.6 Sistemas en un Chip (SoC).
- 5.7 Codiseño SW-HW.
- 5.8 SystemC

- 4.1 Ventajas de los HDL.
- 4.2 Metodología de Diseño.
- 4.3 VHDL. Sintaxis de VHDL.
- 4.4 Codificación de circuitos lógicos en VHDL.
- **4.5 Módulos IP.**
- 4.6 Codiseño SW-HW.
- 4.7 SystemC

- IP: *Intellectual Property*. Múltiples acepciones.
- En las industrias de herramientas EDA y de semiconductores:
 - Unidad de diseño que ha sido pre-verificada, pre-empaquetada y puede incluirse en diseños propios bajo licencia del propietario.
- Pueden ser también unidades de verificación:
 - P.ej.: para verificar que nuestro diseño cumple cierto estándar.
- Otras denominaciones: *Semiconductor IP (SIP)*, *Design IP (DIP)*, *IP blocks*, *IP cores* o, simplemente, IP.

Módulos IP (II)

- Actualmente existen IP para casi cualquier subsistema:
 - CPU (Control Processing Unit).
 - Interfaces E/S: UART, I2C, SPI, CAN, Ethernet, USB, PCI, PCIe, FireWire, RocketIO, Thunderbolt...
 - FPU (Floating Point Unit): co-procesadores coma flotante.
 - GPU (Graphic Processing Unit): aceleradores de gráficos.
 - DSP (Digital Signal Processing): FFT, DCT, MAC, filtros digitales, reguladores digitales...
 - MMU (Memory Management Unit): DRAM, NAND, SD...
 - Encriptación.
 - CODEC: codificadores/decodificadores audio/vídeo.
 - Gestión reloj: síntesis de frecuencias, regeneración del reloj...
 - Señal mixta: convertidores A/D y D/A.

¿Por qué se usan? (I)

- El éxito en el mercado actual depende de ofrecer sistemas más complejos en menos tiempo y a menor coste:
 - Un chip es útil sólo si puede desarrollarse en un tiempo razonable.
 - Actualmente, ni las grandes empresas pueden diseñar un chip completo desde cero con un coste y en un plazo que lo hagan viable comercialmente.
 - La tecnología quedaría obsoleta antes de terminar el diseño.
 - El uso de IP aumenta la productividad del equipo de diseño.

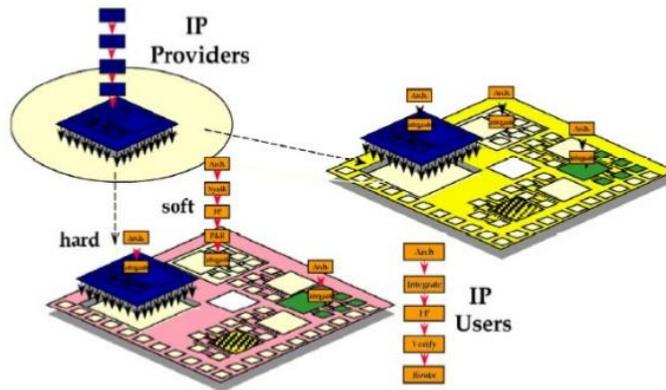
¿Por qué se usan? (II)

- Permiten que una empresa realice diseños aunque no tenga o no pueda permitirse expertos en todas las áreas necesarias.
- Permiten usar tecnologías que están protegidas por patentes.
- Permiten que la empresa se concentre en el desarrollo de lo que la diferencia de sus competidoras y en lo que es experta.
- Permiten amortizar la inversión en su desarrollo a través de la reutilización.

Diseño basado en IP (I)

- Hoy en día es el paradigma dominante de diseño
- Permite abordar el diseño de sistemas muy complejos y heterogéneos.
- Consiste en crear CC.II. mediante la integración e interconexión de módulos IP.
- La selección de los IP adecuados se convierte una de las tareas más críticas del diseño.

Diseño basado en IP (II)



Diseño basado en IP (III)

- La mayoría de los diseños actuales incorporan un procesador.
- En estos sistemas la interconexión entre componentes se realiza a través de *buses*
- La interfaz del *bus* es la frontera natural del IP.
- Se debe elegir un *bus* que:
 - Permita alcanzar la prestaciones previstas.
 - Tenga un soporte suficiente de módulos IP.

Diseño basado en IP (IV)

- Existen buses propietarios. Ej.: Altera Avalon.
- Pero, si se van a usar IP de diverso origen, conviene usar buses estándar:
 - Ej.: Especificación AMBA: *Advanced Microcontroller Bus Architecture* (<http://www.amba.com>):
 - Desarrollado por ARM, uso gratuito.
 - APB (*Advanced Peripheral Bus*): para periféricos.
 - ASB (*Advanced System Bus*): bus síncrono multimaestro.
 - AHB (*Advanced High Performance Bus*): bus de alta velocidad.
 - AXI (*Advanced eXtensible Interface*).
 - ATB (*Advanced Trace Bus*).

Tipos de IP

- Los módulos IP pueden clasificarse en dos categorías principales:
 - **Hard IP**: se suministra como ...
 - Un circuito (*layout, standard cell*) completamente prediseñado (ASIC).
 - Módulos integrados (*embedded*) (FPGA).
 - **Soft IP**: se suministra como una descripción sintetizable en un HDL (VHDL, Verilog...).

Hard IP

- **Prediseñado** ⇒
 - Poco o nada reconfigurable.
 - Conocimiento preciso (menor riesgo de diseño) de:
 - Tamaño.
 - Características eléctricas y temporales.
 - Consumo.
 - Diseñados, fabricados, probados y caracterizados (*qualified*) para un proceso de fabricación específico.

Soft IP

- **Descripción sintetizable en un HDL** ⇒
 - Más fácil de adaptar a nuestro diseño o a nuevas tecnologías.
 - Más difícil de caracterizar (mayor riesgo de diseño). Mayor incertidumbre en:
 - Tamaño.
 - Características eléctricas y temporales.
 - Consumo.
 - Suelen admitir cierto grado de configuración.

Soft vs Hard (I)

- Desventajas *Soft IP*:
 - Un *Soft IP* suele estar menos optimizado (tamaño, prestaciones) que un *Hard IP* equivalente.
 - Dado que el circuito final depende del proceso de síntesis, resto de componentes del sistema, proceso de fabricación o FPGA empleados, existe una mayor incertidumbre sobre su rendimiento final.
 - Por su naturaleza, es más fácil “piratear” un *Soft IP*. Su propietario debe incluir medidas para evitarlo, como introducir marcadores en el código.

Soft vs Hard (II)

- Ventajas *Soft IP*:
 - El ahorro en tiempo de diseño suele compensar la menor optimización.
 - Un *Soft IP* suele ser parametrizable (p.ej., para permitir distintos tamaños de *bus*), facilitando su integración en un diseño, un *Hard IP* no.
 - Un *Soft IP* puede adaptarse a una interfaz distinta de la prevista inicialmente añadiendo lógica de adaptación, denominada adaptador o *wrapper* (un *Hard IP* también, pero perdería casi todas sus ventajas).

Platform FPGA (I)

- FPGA plataforma.
- Iniciativa liderada por Xilinx basada en sus FPGA Virtex-II (2001). Incluía 1 fabricante de semiconductores (IBM) y 4 de herramientas EDA (Mentor, Synopsys, Wild River y MathWorks).
- Actualmente casi todos los fabricantes tienen productos con el mismo concepto.
- Se trata de FPGA desarrolladas especialmente para diseños de sistemas basados en IP.

Platform FPGA (II)

- Contienen:
 - Módulos *Hard IP* integrados para componentes de uso común, complejos y/o de altas prestaciones:
 - Procesadores (PowerPC, ARM).
 - Núcleos DSP.
 - E/S de alta velocidad.
 - Interfaces estándar (PCI, PCIe...).
 - Convertidores A/D.
 - Memoria caché.
 - Lógica programable para adaptar el diseño a nuestra aplicación específica mediante módulos *Soft IP*.

Platform FPGA (III)

- El concepto no se limita al C.I. e incluye:
 - Acuerdos con suministradores de módulos IP para disponer de una oferta amplia.
 - Herramientas de ayuda al diseño HW.
 - SDK's (*Software Development Kits*).
 - SS.OO. (RTOS, Linux, WinCE...).
 - *Drivers* para los distintos SS.OO. y módulos IP.
- Ejemplos:
 - Xilinx: EDK (*Embedded Development Kit*).
 - Altera: Nios II EDS (*Embedded Development Suite*).

Selección de los IP (I)

- La selección se basa en:
 - Función que realiza el módulo.
 - Interfaz.
 - Procesos/dispositivos compatibles.
 - Recursos consumidos (área, macroceldas, biestables, patillas...).
 - Prestaciones (*performance*): frecuencia de reloj, capacidad de procesamiento (*throughput*)...
 - Consumo de energía.

Selección de los IP (II)

- También es importante considerar el proveedor:
 - Documentación exhaustiva.
 - Plataforma de desarrollo: oferta de una familia de módulos y buses compatibles, herramientas de diseño, herramientas de desarrollo SW, drivers, SS.OO.
 - Soporte: formación, asesoría, resolución de errores.
 - Desarrollo regular de nuevas versiones.
 - Metodologías/Sistemas de Calidad que garanticen la calidad de los módulos en el tiempo.
 - Disponibilidad a largo plazo.

Modelo de Negocio (I)

- El uso de un módulo IP normalmente lleva aparejado el pago de:
 - Una licencia (*license fee*): pago único o periódico.
 - y/o
 - Una regalía (*royalty*) por cada dispositivo que incluya el IP: % sobre precio individual.
- Muchos fabricantes de semiconductores incluyen en sus herramientas IP de uso gratuito en sus propios dispositivos (Xilinx: MicroBlaze, Altera: NIOS II,...).

Modelo de Negocio (II)

- A parte del pago, el proveedor puede exigir ciertos términos de uso (*terms of usage*): acuerdos de confidencialidad, pertenecer a un consorcio...
- En el caso IP complejos, novedosos o muy específicos puede ser necesaria una negociación previa.
- Esto puede requerir un tiempo que habrá que sumar al tiempo total de desarrollo.

Modelo de negocio (III)

- Algunas empresas llegan a basar su negocio en licenciar sus módulos IP y en servicios de asesoría, siendo terceros quienes fabrican y usan sus dispositivos.
- Ejemplo: ARM Holdings
 - Familia de procesadores ARM.
 - Familia de GPU Mali.
- También existen desarrolladores individuales de IP.

- Finalmente, también existen iniciativas de código abierto para HW:
 - OpenCores: <http://www.opencores.org>
 - Free Model Foundry: <http://www.freemodelfoundry.com>
 - LEON: <http://www.gaisler.com/index.php/products/processors>
 - SpaceWire:
 - <http://spacewire.esa.int/content/Home/HomeIntro.php>
 - [http://www.esa.int/Our_Activities/Space_Engineering_Technology/Microelectronics/SpaceWire-RMAP - HDL](http://www.esa.int/Our_Activities/Space_Engineering_Technology/Microelectronics/SpaceWire-RMAP_-_HDL)
 - <https://en.wikipedia.org/wiki/SpaceWire>