

Organización y estructura del lenguaje vhdl

DIEGO HILDEBRANDO RAMIREZ AGUILERA

**Organización y estructura del lenguaje vhdl**

VHDL es un lenguaje de especificación definido por el IEEE (Institute of Electrical and Electronics Engineers) (ANSI/IEEE 1076-1993) utilizado para describir circuitos digitales y para la automatización de diseño electrónico . VHDL es acrónimo proveniente de la combinación de dos acrónimos: VHSIC (Very High Speed Integrated Circuit) y HDL (Hardware Description Language). Aunque puede ser usado de forma general para describir cualquier circuito digital se usa principalmente para programar PLD (Programable Logic Device - Dispositivo Lógico Programable), FPGA (Field Programmable Gate Array), ASIC y similares.

Originalmente, el lenguaje VHDL fue desarrollado por el departamento de defensa de los Estados Unidos a inicios de los años 80 basado en el lenguaje de programación ADA con el fin de simular circuitos eléctricos digitales. Posteriormente se desarrollaron herramientas de síntesis e implementación en hardware a partir de los archivos VHD.

Otros métodos para diseñar circuitos son la captura de esquemas (con herramientas CAD) y los diagramas de bloques, pero estos no son prácticos en diseños complejos. Otros lenguajes para el mismo propósito, pero con un nivel de abstracción superior son Verilog y ABEL.

**Organización**

El primer paso del diseño consiste en la construcción del diagrama en bloque del sistema. En diseños complejos como en software los programas son generalmente jerárquicos y VHDL ofrece un buen marco de trabajo para definir los módulos que integran el sistema y sus interfaces, dejando los detalles para pasos posteriores.

El segundo paso es la elaboración del código en VHDL para cada módulo, para sus interfaces y sus detalles internos. Como el VHDL es un lenguaje basado en texto, se puede utilizar cualquier editor para esta tarea, aunque el entorno de los programas de VHDL incluye su propio editor de texto. Después de que se ha escrito algún código se hace necesario compilarlo. El compilador de VHDL analiza este código y determina los errores de sintaxis y chequea la compatibilidad entre módulos. Crea toda la información necesaria para la simulación. El próximo paso es la simulación, la cual le permite establecer los estímulos a cada módulo y observar su respuesta. El VHDL da la posibilidad de crear bancos de prueba que automáticamente aplica entradas y compara las salidas con las respuestas deseadas. La simulación es un paso dentro del proceso de verificación. El propósito de la simulación es verificar que el circuito trabaja como se desea, es decir es más que comparar entradas y salidas. En proyectos complejos se hace necesario invertir un gran tiempo en generar pruebas que permitan evaluar el circuito en un amplio rango de operaciones de trabajo. Encontrar errores en este paso del diseño es mejor que al final, en donde hay que repetir entonces una gran cantidad de pasos del diseño. Hay dos dimensiones a verificar:

Su comportamiento funcional, en donde se estudia su comportamiento lógico independiente de consideraciones de tiempo, como las demoras en las compuertas.

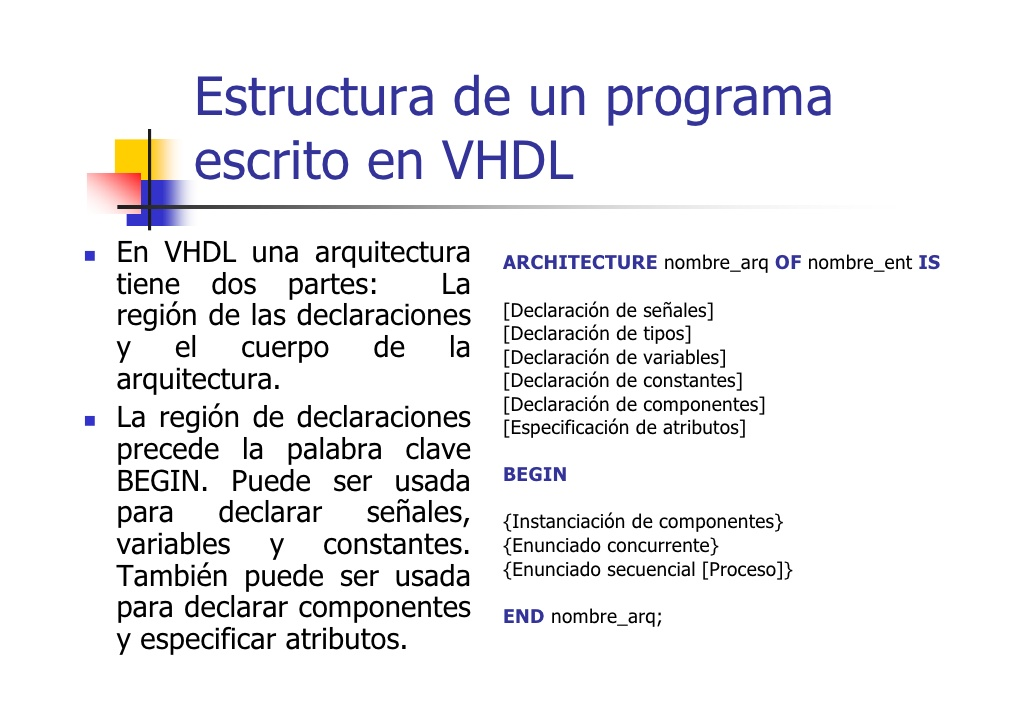
Su verificación en el tiempo, en donde se incluyen las demoras de las compuertas y otras consideraciones de tiempo, como los tiempos de establecimiento (set-up time) y los tiempos de mantenimiento (hold time).

Después de la verificación se está listo para entrar en la fase final del diseño. La naturaleza y herramientas en esta fase dependen de la tecnología, pero hay tres pasos básicos. El primero es la síntesis, que convierte la descripción en VHDL en un conjunto de componentes que pueden ser realizados en la tecnología seleccionada. Por ejemplo, con PLD se generan las ecuaciones en suma de productos. En ASIC genera una lista de compuertas y un netlist que especifica cómo estas compuertas son interconectadas. El diseñador puede ayudar a la herramienta de síntesis especificando requerimientos a la tecnología empleada, como el máximo número de niveles lógicos o la capacidad de salida que se requiere. En el siguiente paso de ajuste (fiting) los componentes se ajustan a la capacidad del dispositivo que se utiliza. Para PLD esto significa que acopla las ecuaciones obtenidas con los elementos AND – OR que dispone el circuito. Para el caso de ASIC se dibujarían las compuertas y se definiría como conectarlas. En el último paso se realiza la verificación temporal, ya que a esta altura es que se pueden calcular los elementos parásitos, como las capacidades de las conexiones. Como en cualquier otro proceso creativo, puede ser que ocasionalmente se avance dos pasos hacia delante y uno hacia atrás (o peor).

**Estructura de programa**

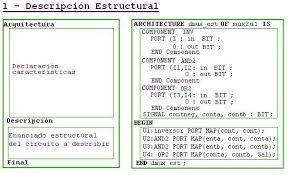
VHDL fue diseñado con base a los principios de la programación estructurada. La idea es definir la interfaz de un módulo de hardware mientras deja invisible sus detalles internos. La entidad (ENTITY) en VHDL es simplemente la declaración de las entradas y salidas de un módulo mientras que la arquitectura (ARCHITECTURE) es la descripción detallada de la estructura interna del módulo o de su comportamiento. En la siguiente figura se ilustra el concepto anterior. Muchos diseñadores conciben la Entity como una funda de la arquitectura dejando invisible los detalles de lo que hay dentro (architecture).

Esto forma la base de un sistema de diseño jerárquico, la arquitectura de la entidad de más nivel (top level) puede usar otras entidades, dejando invisible los detalles de la arquitectura de la identidad de menos nivel. En la figura las entidades B, E y F no utilizan otras entidades. Mientras que la entidad A utiliza todas las demás. A la pareja entidad-arquitectura se la llama modelo. En un fichero texto VHDL la entidad y la arquitectura se escriben separadas, por ejemplo, a continuación se muestra un programa muy simple en VHDL de una compuerta de 2 entradas. Como otros programas, VHDL ignora los espacios y saltos de líneas. Los comentarios se escriben con 2 guiones (--) y terminan al final de la línea. En la figura siguiente se muestra la estructura de un modelo en VHDL. SINTAXIS PARA LA DECLARACIÓN DE LA ENTIDAD VHDL define muchos caracteres especiales llamados “palabras reservadas”. Aunque las palabras reservadas no son sensibles a las mayúsculas o minúsculas, en el ejemplo que sigue las utilizaremos en mayúsculas y negrita para identificarlas.



Además de darle nombre a la entidad el propósito de la declaración es definir sus señales (o ports) de interfaz externa en su declaración de ports. Además de las palabras reservadas o claves **ENTITY**, **IS**, **PORT** and **END**, una ENTITY tiene los siguientes elementos.

* Nombre entidad; es un identificador seleccionado por el usuario para seleccionar la entidad.
* Nombre de señal; es una lista de uno o más identificadores separados por una coma y seleccionados por el usuario para identificar las señales externas de la interfaz.
* MODO es una de las 4 siguientes palabras reservadas para indicar la dirección de la señal



**Bibliografía**

[**https://editorialpatria.com.mx/pdffiles/9786074386219.pdf**](https://editorialpatria.com.mx/pdffiles/9786074386219.pdf)

[**https://eprints.ucm.es/26200/1/intro\_VHDL.pdf**](https://eprints.ucm.es/26200/1/intro_VHDL.pdf)