Brayan Ramirez Benitez	**INSTITUTO POLITÉCNICO NACIONAL**	**ESCUELA SUPERIOR DE CÓMPUTO**
1er PARCIAL	**TITULO DE LA TAREA**	**10 de febrero de 2022**
3CM11	ARQUITECTURA DE COMPUTADORAS	PERIODO ESCOLAR 2022-2

3CM11 Tarea 02 Primer Parcial

Bloques principales de un archivo básico VHDL implementable

Encabezado

En el encabezado se guarda información muy variada como, por ejemplo, las definiciones de tipos de datos, de operadores aritméticos y lógicos, la descripción de diversos circuitos que resultan útiles a la hora de describir otros circuitos, etc. También puede contener paquetes (packages) los cuales, a su vez, pueden contener funciones (functions) y/o procedimientos (procedures) que se denominan genéricamente como subprogramas. Para hacer visible y por lo tanto utilizable en un diseño el contenido de un encabezado es necesario declararla previamente. Para declarar una biblioteca es necesario escribir, al menos, dos líneas de código. En la primera línea se indica su nombre, lo que hace que dicha biblioteca esté disponible en la entidad y en la arquitectura que se indican más adelante. En la siguiente o siguientes líneas se indican los paquetes de la biblioteca que se van a utilizar, haciendo que estén disponibles todas las definiciones, funciones y procedimientos que contienen.

Hay dos bibliotecas estándar que se incluyen por defecto en todos los proyectos y que por lo tanto no necesitan ser declaradas:

std: esta biblioteca contiene los paquetes standar y textio. El paquete standar contiene, entre otras cosas, las definiciones de diversos tipos de datos, de operadores lógicos, de operadores aritméticos, de operadores de desplazamiento, etc. El paquete textio contiene diversos recursos relativos al manejo de archivos de texto.

work: es la biblioteca en la que se guarda por defecto todo lo relacionado con el diseño o descripción que se está realizando.

Hay una tercera biblioteca, denominada ieee, que contiene diversas definiciones y funciones estándar utilizadas habitualmente en la descripción de circuitos. Para poder utilizarla en un diseño es necesario declararla (está disponible en todos los sintetizadores). Esta biblioteca contiene varios paquetes de los cuales, para la mayoría de los diseños, es suficiente con utilizar los paquetes ieee.std_logic_1164.all e ieee.numeric_std.all.

El paquete ieee.std_logic_1164.all es un paquete básico del ieee que, entre otras cosas, contiene la definición de los tipos de datos std_ulogic_vector, std_logic_vector, std_ulogic y std_logic11, la definición de los operadores lógicos not, and, nand, or, nor, xor y xnor, funciones de conversión entre los tipos de datos anteriores, funciones de detección de flancos de subida y de bajada, operadores de relación: =, /=, <=, >, >= y operadores de desplazamiento de los bits guardados en un vector. El paquete ieee.numeric_std.all contiene las definiciones necesarias para realizar operaciones aritméticas con datos de tipo signed y unsigned.

Entidad

La entidad en VHDL es la abstracción de un circuito, ya sea desde un complejo sistema electrónico o una simple puerta lógica. La entidad únicamente describe la forma externa del circuito, en ella se enumeran las entradas y las salidas del diseño. Una entidad es análoga a un símbolo esquemático en los diagramas electrónicos, el cual describe las conexiones del dispositivo hacia el resto del diseño.

En la entidad se define externamente al circuito, el nombre y número de puertos, tipos de datos de entrada y salida, además contiene toda la información necesaria para conectar el circuito a otros circuitos.

Los puertos pueden ser de entrada in, salida out, entrada-salida inout o buffer. Los puertos de entrada solo se pueden leer y no se pueden modificar. Los puertos de salida solo se pueden escribir, pero nunca tomar decisiones de su valor. En la entity se pueden definir unos valores genéricos (generic) que se utilizaran para declarar propiedades y constantes del circuito, independientemente de cuál sea la arquitectura. Generic se utiliza para definir retardos de señales y ciclos de reloj, para introducir una constante que será utilizada posteriormente en la arquitectura.

Ejemplo

```
Entity F is
        generic (N: natural := 8) ;
        port (A, B,: in bit_vector(N-1 downto 0);
                Y: out bit);
end F;
```

Arquitectura

Los pares de entidades y arquitecturas se utilizan para representar la descripción completa de un diseño. Una arquitectura describe el funcionamiento de la entidad a la que hace referencia, es decir, dentro de architecture tendremos que describir el funcionamiento de la entidad a la que está asociada utilizando las sentencias y expresiones propias de VHDL. La arquitectura define internamente el circuito, las señales internas, funciones, procedimientos, constantes, además, la descripción de la arquitectura puede ser estructural o por comportamiento.

El código VHDL se escribe dentro de architecture. Cada architecture va asociada a una entity y se indica en la primera sentencia. Antes de begin se definen todas las variables (señales) internas que vas a necesitar para describir el comportamiento de nuestro circuito, se definen los tipos particulares que necesitamos utilizar y los componentes, otros circuitos ya definidos y compilados de los cuales conocemos su interfaz en VHDL (su entity). Desde begin hasta end escribiremos todas las sentencias propias de VHDL, pero no todas pueden utilizarse en cualquier parte del código. Así pues, aquellas sentencias de VHDL que tengan definido un valor para cualquier valor de la entrada (y que nosotros denominamos sentencias concurrentes) podrán ir en cualquier parte del código, pero fuera de la estructura process. Aunque no es el fin de este manual, puede afirmarse que todas las sentencias concurrentes se traducirán en subcircuitos combinacionales. También fuera de la estructura process, se instanciarán los componentes, subcircuitos ya definidos,

utilizados por el circuito actual, indicando cuáles son sus entradas y sus salidas de entre las señales del circuito del que forman parte. El process es una estructura particular de VHDL (que se describe con mucho más detalle más adelante) que se reserva principalmente para contener sentencias que no tengan obligatoriamente que tener definido su valor para todas las entradas (el ejemplo más común es una estructura if-else incompleta). Esto obliga a que la estructura process almacene los valores de sus señales y pueda dar lugar (no siempre) a subcircuitos secuenciales.

Referencias en formato APA

Entrena, L. (2010). EL LENGUAJE VHDL CONCEPTOS BÁSICOS. PDF. Recuperado 10 de febrero de 2022, de http://ocw.uc3m.es/tecnologia-electronica/circuitos-integrados-y-microelectronica/teoria_vhdl/vhdl-2-conceptos-basicos-1

Lenguaje VHDL - Lógica Programable. (2014). Logica programable. Recuperado 10 de febrero de 2022, de https://sites.google.com/site/logicaprogramable/vhdl/lenguaje-vhdl

Mora Campos, A. (2018). Apuntes VHDL. PDF. Recuperado 10 de febrero de 2022, de http://www.itq.edu.mx/carreras/IngElectronica/archivos_contenido/Apuntes%20de%20materias/Apuntes_VHDL_2016.pdf

Sanchez, E. (2016). Introduccion a la programacion en VHDL. PDF. Recuperado 10 de febrero de 2022, de https://eprints.ucm.es/id/eprint/26200/1/intro_VHDL.pdf