|  |  |  |
| --- | --- | --- |
| CASTRO CRUCES JORGE E. | **INSTITUTO POLITÉCNICO NACIONAL** | **ESCUELA SUPERIOR DE CÓMPUTO** |
| PRIMER PARCIAL | **TAREA 1** | **03/03/2021** |
| 3CM12 | ARQUITECTURA DE COMPUTADORAS |  |

**TAREA 1**

**ARQUITECTURA:**

Una arquitectura se define como la estructura que describe el funcionamiento de una entidad, de tal forma que permita el desarrollo de los procedimientos que se llevarán a cabo con el fin de que la entidad cumpla las condiciones de funcionamiento deseadas.

La gran ventaja que presenta VHDL para definir una arquitectura radica en la manera en que pueden describirse los diseños; es decir, mediante el algoritmo de programación empleado se puede describir desde el nivel de compuertas hasta sistemas complejos.

De manera general, los estilos de programación utilizados en el diseño de arquitecturas se clasifican como:

• Estilo funcional

• Estilo por flujo de datos

• Estilo estructural

El nombre asignado a estos estilos no es importante, ya que es tarea del diseñador escribir el comportamiento de un circuito utilizando uno u otro estilo que a su juicio le sea el más acertado.

**ESTILOS DE PROGRAMACIÓN EN VHDL**

**Descripción funcional o algorítmica:**

Como se puede observar, la descripción funcional se basa principalmente en el uso de procesos y de declaraciones secuenciales, las cuales permiten modelar la función con rapidez.

Algunos atributos son:

* Debe tener al menos 1 proceso declarado.
* Es útil a la hora de programar circuitos secuenciales.

**Descripción por flujo de datos:**

La descripción por flujo datos indica la forma en que los datos se pueden transferir de una señal a otra sin necesidad de declaraciones secuenciales (if-then-else). Este tipo de descripciones permite definir el flujo que tomarán los datos entre módulos encargados de realizar operaciones. En este tipo de descripción se pueden utilizar dos formatos: mediante instrucciones whenelse (cuando-si no) o por medio de ecuaciones booleanas.

Algunos atributos son:

* Las señales pasan de las entradas a las salidas vía una función o de forma directa.
* Es útil a la hora de programar circuitos combinatorios o combinacionales.

**Descripción estructural o jerárquico:**

Como su nombre indica, una descripción estructural basa su comportamiento en modelos lógicos establecidos (compuertas, sumadores, contadores, etc.). El usuario puede diseñar estas estructuras y guardarlas para su uso posterior o tomarlas de los paquetes contenidos en las librerías de diseño del software que se esté utilizando. Para iniciar la programación de una entidad de manera estructural, es necesario la descomposición lógica del diseño en pequeños submódulos (jerarquizar), los cuales permiten analizar de manera práctica el circuito, ya que la función de entrada/salida es conocida.

Algunos atributos son:

* Es útil a la hora de programar mediante arquitectura por bloques o de forma jerárquica.

**BLOQUES BÁSICOS DE UN ARCHIVO VHDL (implementable)**

**Entidad:**

El diseño de la entidad puede ser descrita en términos de una jerarquía de bloques, cada uno de los cuales representa una parte del diseño. El bloque de nivel superior en la jerarquía es la entidad del diseño en sí; ya que un bloque puede ser parte de una biblioteca o librería dicho bloque puede ser utilizado en otros diseños. Los bloques anidados en una jerarquía son bloques internos definidos en la declaración del bloque que lo contiene.

Para la declaración de una entidad hay que definir un nombre y también el nombre de los puertos de I/O que utilizara el diseño, a continuación mostramos un esquema básico para la declaración de entidades. Las entidades que usaremos inicialmente serán entidades en donde solo se declararán puertos solamente.:

entity Nombre\_Entidad is

end Nombre\_Entidad;

Dentro de las declaraciones de las entidades, hay que realizar la declaración de los puertos utilizados en el diseño del dispositivo. Los puertos se declaran utilizando el comando *port* en la cual se debe definir los nombres, modo y tipos de las señales utilizadas como I/O.

El formato para la declaración de puertos es:

Port(

Nombre\_puerto1 : [Modo del puerto] [Tipo de dato del puerto];

Nombre\_puerto2 : [Modo del puerto] [Tipo de dato del puerto];

Nombre\_puerto3 : [Modo del puerto] [Tipo de dato del puerto]);

Las declaraciones de los puertos se realizan dentro de la instrucción port la cual debe terminar en un carácter de punto y coma; además de que cada declaración de un puerto debe terminar también en un carácter de punto y coma, con excepción de la última declaración en donde no es necesario. La declaración de los puertos consta de un nombre del puerto seguido de un carácter de dos puntos (:), luego se debe especificar el modo del puerto y el tipo de dato. A continuación se listan los modo y tipos de datos más utilizados para la declaración de puertos.

**Arquitectura:**

En la arquitectura es en donde nos vamos a preocupar por el código y las instrucciones necesarias para que el dispositivo realice el trabajo que le permita cumplir con la finalidad del diseño.

El formato para la declaración de una arquitectura contempla el nombre de la entidad a la cual la arquitectura está ligada, un nombre con el que se va a conocer la arquitectura. A continuación mostraremos un formato estándar para la declaración de una arquitectura, veamos:

architecture Nombre\_Arq of Nombre\_Entidad is

begin

end Nombre\_Arq;

Entre el encabezado principal de la declaración de la arquitectura y la instrucción *begin* se encuentra el espacio destinado para la declaración de señales utilizadas en el código que conforma los comando, instrucciones y estructuras para que el dispositivo pueda tomar las señales que proviene de los puertos de entrada ejecutar el código y producir las salidas.

El conjunto de instrucciones y estructuras se deben colocar entre la instrucción *begin* y el cierre de la declaración de la arquitectura. El estilo de programación de la arquitectura dependerá finalmente del estilo de programación del diseñador, en este trabajo se tratarán los estilos de programación: Combinacional, comportamental, diagrama de estados y captura esquemática.

**Referencias:**

Maxinez D. G. & Alcalá J.. (2002). 2. VHDL: Su organización y arquitectura. *En VHDL El arte de programar sistemas digitales*(pp.46-55). México: CONTINENTAL.

S/I. (S/I). Lenguaje VHDL. Marzo 01,2021, de LógicaProgramable Sitio web: <https://sites.google.com/site/logicaprogramable/vhdl/lenguaje-vhdl>