

Pre-Informe 3

Hardware Description Languages

Ignacio Cea Ruiz
201773009-9

Pregunta 1: Investigación de Conceptos

¿Qué es un lenguaje de descripción de hardware? ¿Qué lo diferencia de un lenguaje de programación?

Un lenguaje de descripción de hardware o HDL, por sus siglas en inglés, es una forma de modelar un circuito electrónico y, sobre todo, circuitos lógicos. Este lenguaje consta de asignaciones, módulos, operaciones lógicas, tal como un lenguaje de programación cualquiera. Sin embargo, la principal diferencia de los HDL con los lenguajes de programación es que los primeros describen explícitamente conexiones en un circuito físico, no operaciones a realizar dentro de una CPU como lo hace un lenguaje de programación. De esta forma, la gran utilidad de los HDL radica en que permiten crear y simular circuitos que luego pueden construirse físicamente sobre circuitos integrados para así utilizarlos en dispositivos electrónicos.

Explique la funcionalidad de los tipos de variable *wire*, *reg* y *logic*, y cuáles son sus diferencias.

Las variables del tipo *logic* equivalen a variables booleanas, es decir, 0 o 1. También pueden definirse como un *bus* de varias variables, de esta forma podemos asignar números a variables *logic* en su forma binaria.

Por otro lado, *reg* y *wire* son tipos de variables en desuso mayormente usadas en el predecesor de SystemVerilog, Verilog. *reg* puede ser el output de un registro tal como un flip-flop o latch, pero también puede usarse equivalente a una variable *logic*. Sin embargo, si se usa como output de un registro, debe declararse explícitamente como *reg*. Por otro lado, *wire* (literalmente *cable*, en el idioma inglés) es el tipo de variable que toman todas las variables a las que no se les asigna un tipo explícitamente.

¿Cuál es la diferencia entre los operadores de asignación *assign*, \leq y $=$, y en qué secciones del módulo se utilizan?

El operador *assign* se usa para describir lógica combinacional, en conjunto con el operador $=$, de la forma *assign var = statement*. Por lo tanto puede utilizarse en cualquier parte del módulo, excepto cuando se quiera describir lógica secuencial; para ello debe usarse el operador \leq . Este último se utiliza dentro de una sección *always*, la cual tiene una *sensitivity list*. De esta forma, si dentro de la sección *always* se tiene $q \leq d$, quiere decir que q obtendrá el valor de d cuando se cumpla una condición en la *sensitivity list* (i.e., el canto de subida o bajada del reloj).

Dada la siguiente expresión:

```
reg [15:0] g = 16'hA6B2;
```

Explique su estructura, significado de cada una de sus partes y cuál es el equivalente binario del valor definido.

La expresión muestra la asignación de un número a una variable de 16 bits ($g[15] \dots g[0]$) cuyo nombre es g y es de tipo **reg**. Que sea de tipo **reg** implica que es equivalente a una variable de tipo *logic* pero que también puede ser usada en un bloque de lógica secuencial. El valor $16'hA6B2$ se descompone en el 16 a la izquierda del apóstrofe, que indica un número de 16 bits, la letra **h** a la derecha, que indica que se trata de un número en base hexadecimal y, luego, el valor del número dadas las dos indicaciones anteriores: $A6B2_{16} = 10 \times 16^3 + 6 \times 16^2 + 11 \times 16^1 + 2 \times 16^0 = 42674_{10} = 1010011010110010_2$