

Cómo definir constantes temporales en VHDL

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/02/24/como-definir-constantes-temporales-en-vhdl/>

Blog: <https://soceame.wordpress.com/>

GitHub: <https://github.com/DRubioG>

Fecha última modificación: 24/02/2025

Uno de los inconvenientes que tiene VHDL es que no es un lenguaje fácil para definir variables temporales. Por ello se suele recurrir a el uso de la calculadora en modo programador para obtener los valores binarios y luego estos valores se copian a FW, y el problema es que tienes que definir de dónde proceden estos valores, y además en caso cualquier modificación de este valor tienes que volver a recalcular el valor.

Pues en esta entrada te voy a explicar cómo evitar tener que hacer estos cálculos declarando constantes.

Como constantes temporales hay que definir dos tipos, las constantes de comparación y las constantes de simulación.

Constantes de comparación

Estas constantes solo tienen el objetivo de utilizarse en comparaciones, por lo que son sintetizables debido a que el sintetizador calcula el valor antes de hacer el resto de la síntesis.

Por ejemplo, imaginemos que tenemos un reloj de 100 MHz y queremos calcular la constante que nos da el contador para unos baudios de 9600.

```
constant FREQ : integer := 100_000_000;  
constant BAUD : integer := 9600;  
constant FREQ_DIV : integer := FREQ / BAUD;
```

Entonces, ya no es necesario hacer el cálculo y conseguir la constante de salida. Por lo que podemos modificar parámetros como la frecuencia o los baudios sin la necesidad de hacer de nuevo el cálculo.

Constantes de simulación

Estas constantes son solo para simulación, por lo que no son sintetizables, y además, están definidas como tipo *time*. El cálculo se hace como en las constantes de simulación solo que con la escala de tiempos que se utiliza.

Por ejemplo en el caso anterior, se tiene que multiplicar por 1ns el valor obtenido del cálculo anterior. Esta constante sí que puede ser utilizada por un *wait for*.

```
constant FREQ : integer := 100_000_000;  
constant BAUD : integer := 9600;  
constant FREQ_DIV : integer := FREQ / BAUD;  
constant FREQ_DIV_ns : time := FREQ_DIV * 1 ns;
```

```
...  
process begin  
    a <= '0';  
    wait for FREQ_DIV_ns;  
    a <= '1';  
    wait;  
end process;  
...
```

<https://soceame.wordpress.com/>