Always@ en Verilog

La instrucción process es usada siempre que quiera utilizar un **condicional** o **bucle**, cuenta con un **paréntesis donde se deben poner las entradas que se vayan a usar en el condicional o bucle**, puede tener o no nombre y tiene su propio begin y end process;

La sintaxis de process es la siguiente:

* **Declaración de process con nombre:** De esta manera puedo declarar varios process dentro de un módulo e identificar a cada uno.

nombreProcess: process (entradas que vaya a usar)

//Las entradas que vaya a usar dentro del condicional o bucle deben ir separadas entre ellas por una coma o un or en Verilog

begin

--Bucle o condicional que quiero usar

end process nombreProcess;

* **Declaración de process sin nombre:** Se usa para cuando tengo un solo process dentro del módulo.

process (dentro de su paréntesis debo poner las entradas que vaya a usar)

//Las entradas que vaya a usar dentro del condicional o bucle deben ir separadas entre ellas por una coma o un or en Verilog

begin

--Bucle o condicional que quiero usar

end process;

Operadores relacionales

Para usar **condicionales** o **bucles** se suelen usar **operadores relacionales** para crear las condiciones a evaluar:

* Igual que **==**
* Diferente que **!=**
* Mayor que **>**
* Mayor o igual que **>=**
* Menor que **<**
* Menor o igual que **<=**

Condicionales

Los **condicionales** son estructuras que analizan una **operación relacional** para ejecutar cierto código o no dependiendo del resultado de dicha operación lógica y existen 3 tipos:

* + Condicional if: Se usa cuando quiero analizar **una sola condición**.

//Módulo: Dentro de éste va todo el código escrito en Verilog

module nombreModulo (

//Las entradas y salidas: van declaradas dentro del paréntesis

        input wire  entrada1,

input wire  entrada2,

        output wire salida

);

//Fuera del paréntesis del módulo se indica lo que hará el programa con las

//entradas/salidas declaradas, aquí es donde se declaran las condicionales o bucles

process (entrada1 or entrada2)

//Las entradas que vaya a usar dentro del condicional o bucle deben ir separadas entre ellas por una coma o un or en Verilog

begin

--Bucle o condicional que quiero usar

end process;

endmodule

* + Condicional else if: Se usa cuando quiero analizar **más de una sola condición relacionada con la otra**.
  + Condicional case: Se usa cuando quiero analizar **más de un solo valor preprogramado que puede tomar una entrada, salida o variable**.
* Los **bucles** son estructuras que analizan una operación lógica para ejecutar cierto código de manera repetitiva, esto porque algunos programas tienen la necesidad de que un mismo código se ejecute varias veces, el código se ejecutará o no dependiendo del resultado de dicha operación y existen 3 tipos, divididos en dos categorías:
  + Bucle for.

Signal vs Variable

Para poder pasar este método Shift Add-3 de conversión Binario a BCD tendremos que usar un concepto nuevo llamado **signal** o **variable**, estos son usados para crear una variable que no sea ni una entrada ni una salida de la tarjeta de desarrollo y que solo pueda vivir durante la ejecución del programa, se declaran de la misma forma en la que se declara una entrada o salida en forma de vector o de 1 bit.

La diferencia entre signal y variable es la siguiente:

* **signal:** Puede usarse en cualquier punto de la arquitectura del programa y puede declararse dentro o fuera de algún process.

signal nombreSignal\_1: STD\_LOGIC\_VECTOR (2 downto 0); --signal vector

signal nombreSignal\_2: STD\_LOGIC; --signal de 1 bit

process (dentro de su paréntesis debo poner las entradas que vaya a usar)

//Dentro del paréntesis debo poner las entradas que vaya a usar dentro del condicional

//o bucle

signal nombreSignal\_3: STD\_LOGIC\_VECTOR (2 downto 0); --signal vector

signal nombreSignal\_4: STD\_LOGIC; --signal de 1 bit

//signal puede ir declarada dentro o fuera de process

begin

//Bucle o condicional que quiero usar

end process;

* **variable:** Se debe declarar siempre dentro de un process y solo puede usarse dentro del mismo.

process (dentro de su paréntesis debo poner las entradas que vaya a usar)

//Las entradas que vaya a usar dentro del condicional o bucle deben ir separadas entre ellas por una coma o un or en Verilog o por una coma solamente en VHDL

variable nombreVariable1: STD\_LOGIC\_VECTOR (2 downto 0); --variable vector

variable nombreVariable2: STD\_LOGIC; -- variable de 1 bit

//La variable debe ir declarada entre la declaración del process y su begin, además solo

//se podrá usar dentro del mismo process

begin

//Bucle o condicional que quiero usar

end process;