Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2343 — Arquitectura de Computadores 2019-2

Tutorial – Componentes e Instancias en VHDL

Tutorial para hacer uso de componentes e instancias dentro de una entidad en VHDL

Introducción:

Para describir un circuito en un lenguaje como VHDL, solo se necesita describir las entradas y salidas en una entidad, y dentro de dicha entidad se describe las señales y comportamientos necesarios para para dar algún resultado esperado.

Sin embargo, cuando los circuitos comienzan a ser muy grandes, resulta más óptimo describirlos con distintas entidades que se comunican entre sí. Esto permite que la descripción sea más clara y práctica al momento de encontrar errores.

Para lograr esta comunicación entre entidades en VHDL, existen los **componentes**, los que permiten dar referencia a una entidad dentro de otra y las **instancias** que permiten usar dicha referencia.

En este tutorial se enseñará las nociones básicas de como utilizar componentes e instancias de VHDL en Vivado. Para seguir este, es recomendado haber leído los seis tutoriales pasados disponibles en el *Syllabus* del curso.

1 Estructura de un archivo VHDL

1.1 Estructura vista hasta ahora:

Por el momento en todos los tutoriales hemos visto la siguiente estructura dentro de un archivo en VHDL.

Librería

```
library IEEE; -- includes
use IEEE.STD LOGIC 1164.ALL;
```

Entidad

```
entity Tutorial is
    Port ( sw : in STD_LOGIC_VECTOR (7 downto 0);
           led : out STD_LOGIC_VECTOR (7 downto 0));
end Tutorial;
```

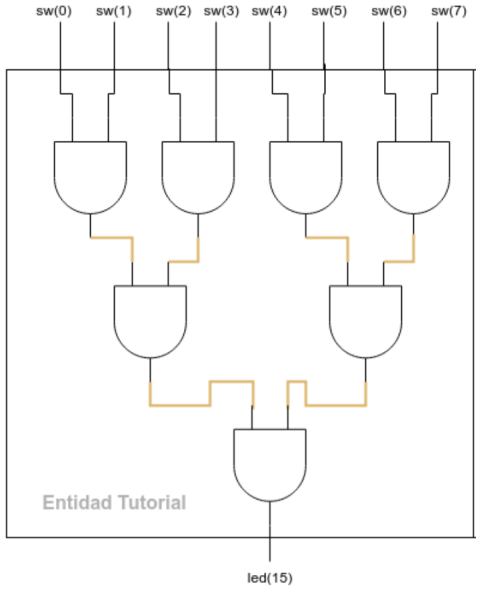
Arquitectura

```
architecture Behavioral of Tutorial is
-- Aquí empieza las declaraciones ##################################################
signal signal_and1 : STD_LOGIC;
signal signal_and2 : STD_LOGIC;
signal signal_and3 : STD_LOGIC;
signal signal_and4 : STD_LOGIC;
signal signal_and5 : STD_LOGIC;
signal signal_and6 : STD_LOGIC;
-- Aquí terminan las declaraciones ##################################################
begin -- Este begin da comienzo a los comportamientos de la entidad


signal_and1 <= sw(0) and sw(1);
signal_and2 <= sw(2) and sw(3);
signal_and3 <= sw(4) and sw(5);
signal_and4 <= sw(6) and sw(7);

signal_and5 <= signal_and1 and signal_and2;
signal_and6 <= signal_and3 and signal_and4;

led(15) <= signal_and5 and signal_and6;


end Behavioral; -- Este end da termino a los comportamientos de la entidad
```

Declaraciones

Comportamientos

1.2 Diagrama ejemplo anterior



NOTA: En amarillo se destacan las señales `signal_and`.

1.3 Descripción de los elementos estructurales

En dicha estructura tenemos:

- **Librería:** Junto con el uso de clausula, permite ocupar los tipos de datos de lógica estándar para trabajar.
- **Entidad:** Es el circuito principal a ser descrito. En dicha sección se menciona las entradas y salidas que este tiene.
- **Arquitectura:** Es el segmento donde se da inicio a toda la descripción del circuito de la entidad, donde se definen todas sus **declaraciones** y **comportamientos**.
- **Declaraciones:** Parte de la sección de la arquitectura, básicamente describe los elementos que serán ocupados dentro de la entidad distintos a sus entradas y salidas. Hasta el momento solo hemos declarado señales (pronto veremos registros).
- **Comportamientos:** Parte de la sección de la arquitectura que describe su comportamiento, es decir las conexiones mediante compuertas entre las entradas, salidas y señales.

Notar que en todas las estructuras vistas hasta el momento, el comportamiento dado se declara dentro de una sola entidad.

Si quisiéramos ocupar otras entidades que describan otros circuitos dentro de nuestra entidad principal, como una caja negra, debemos llamarla como un **componente**.

2 Componentes

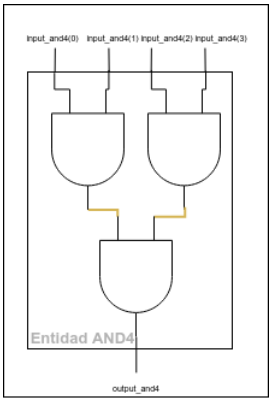
2.1 Definición y ejemplo

Los componentes son las **declaraciones** de entidades externas a la entidad principal.

Como ejemplo, usaremos una nueva entidad, mediante una *Source* llamada AND4 (para saber como crear una *Source* favor revisar tutoriales pasados). Esta entidad se muestra a continuación:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity AND4 is
    Port ( input_and4 : in STD_LOGIC_VECTOR (3 downto 0);
           output_and4 : out STD_LOGIC);
end AND4;

architecture Behavioral of AND4 is

signal signal_and1: STD_LOGIC;
signal signal_and2: STD_LOGIC;

begin

signal_and1 <= input_and4(0) and input_and4(1);
signal_and2 <= input_and4(2) and input_and4(3);

output_and4 <= signal_and1 and signal_and2;

end Behavioral;
```

2.2 Diagrama ejemplo anterior



2.3 Declarando un componente dentro de una entidad

Ahora para ocupar la entidad recién creada, debemos llamarla como un **componente** en el sector de las declaraciones.

La forma de llamada es similar a la de una entidad, solo que en vez de escribir `entity`, se escribe `compoment`, y en vez de terminar con el nombre de la entidad, termina con la palabra `compoment`. Lo anterior, se se muestra en el siguiente código:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Tutorial is
    Port ( sw : in STD_LOGIC_VECTOR (7 downto 0);
           led : out STD_LOGIC_VECTOR (15 downto 0));
end Tutorial;

architecture Behavioral of Tutorial is

component AND4 is
    Port ( input_and4 : in STD_LOGIC_VECTOR (3 downto 0);
           output_and4 : out STD_LOGIC);
end component;

signal signal_and1 : STD_LOGIC;
signal signal_and2 : STD_LOGIC;


begin

end Behavioral;
```

2.4 Consideraciones a tomar en cuenta:

En el ejemplo anterior hay varias consideraciones a tener en cuenta, dentro de las cuales están:

- Las declaraciones no tienen un orden en particular. Esto quiere decir que podemos declarar primero las señales y luego componentes, viceversa o intercalado. Lo importante es que declararé en la **sección de declaraciones** dentro de la **sección de arquitectura** (revisar la estructuración del primer punto para más información).
- Aun considerando el punto anterior es **recomendable** que se mantenga un orden consistente para mejor legibilidad de la descripción de su circuito.

3 Instancias

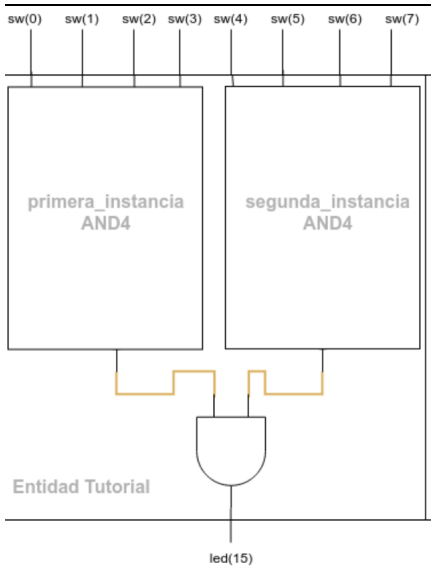
3.1 Definición y ejemplo

Las instancias de un componente son referencias de bajo nivel del componente en el diseño de la entidad. En esencia crean copias únicas (o instancias) de estos componentes. Las instancias de componentes no tienen un orden para ser referido. No obstante, para instanciar cualquier componente, este debe estar escrito en la **sección de arquitectura** (para mayor información de las secciones revisar el primer punto del tutorial).

Respecto al sintaxis, se debe escribir con un nombre característico seguido de un `statment Port map`, luego, se describe la conexión directa de las señales respectivas con las entradas de la instancia. Para este ejemplo, se ocuparán dos instancias del componente `AND4` conectadas con las entradas, señales y salida de la entidad Tutorial. Lo anterior, se se muestra en el siguiente código :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity Tutorial is
    Port ( sw : in STD_LOGIC_VECTOR (7 downto 0);
           led : out STD_LOGIC_VECTOR (15 downto 0));
end Tutorial;

architecture Behavioral of Tutorial is

component AND4 is
    Port ( input_and4 : in STD_LOGIC_VECTOR (3 downto 0);
           output_and4 : out STD_LOGIC);
end component;

signal signal_and1 : STD_LOGIC;
signal signal_and2 : STD_LOGIC;

begin

led(15) <= signal_and1 and signal_and2; -- señales con salida de ambas instancias

primera_instanciaAND4: AND4 port map(
    input_and4       => sw(3 downto 0), -- Input de la entidad es conectado input de la instancia
    output_and4       => signal_and1); -- Aquí el output es pasado a la señal signal_and1

segunda_instanciaAND4: AND4 port map(
    input_and4       => sw(7 downto 4), -- Input de la entidad es conectado input de la instancia
    output_and4       => signal_and2); -- Aquí el output es pasado a la señal signal_and2

end Behavioral;
```

3.2 Diagrama ejemplo anterior



3.3 Consideraciones a tomar en cuenta:

En el ejemplo anterior hay varias consideraciones a tener en cuenta, dentro de las cuales están:

- La entidad principal a ser compilada en la placa debe estar con mayor prioridad. En este caso Tutorial, debería estar como *Set as Top* dentro de la carpeta *Desing Source* (revisar tutoriales pasados para saber como hacerlo).
- Al igual que en las declaraciones, no hay un orden especifico para describir las conexiones entre señales e instancias, mientras esté en la sección de comportamiento.
- Aun considerando el punto anterior es **recomendable** que mantenga un orden consistente para mejor legibilidad de la descripción de su circuito.

Bibliografía:

Para más información revisar los siguientes enlaces

- `https://www.ics.uci.edu/~jmoorkan/vhdlref/Synario%20VHDL%20Manual.pdf`
- `http://valhalla.altium.com/Learning-Guides/TR0114%20VHDL%20Language%20Reference.pdf`
- `http://web.engr.oregonstate.edu/~traylor/ece474/vhdl_lectures/component_instantiaton.pdf`