Uso de ‘ – ‘ (don’t care) en VHDL

Nota Técnica 04

Cristian Sisterna

Introducción

El valor ' - ', comúnmente llamado don't care o no importa, normalmente es usado en funciones booleanas para representar indistintamente un '1' o un '0'. Se usa principalmente para minimizar una función booleana al poder usar el valor lógico ('1' o '0') que resulta en una menor implementación del hardware de la función. Sin embargo en VHDL el tipo *std_logic* define a ' - ' como un valor en sí. Es decir no puede ser indistintamente '1' o '0' porque es ' - '.

Veremos a continuación como se describe en VHDL una función en la que ' - ' puede tomar valor '1' o '0'.

Nota: en alguna bibliografía se usa 'X' en lugar de ' - ' para representar el don't care. En VHDL 'X" es otro valor definido en el tipo *std_logic*, por lo que NO se debe usar para representar don't care. 'X' en VHDL es un valor 'don't know', o 'no sé'.

' - ' como entrada en funciones Booleanas

Normalmente las funciones booleanas son representadas por una tabla de verdad que describe los distintos valores que se obtendrán en la salida de la función como respuesta a distintos valores de entrada.

Veamos un ejemplo, la función de un decodificador con prioridad tiene la siguiente tabla de verdad:

Entrada			**Salida**	
I2	**I1**	**I0**	**D1**	**D0**
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0
0	1	0	0	1
0	1	1	0	1
0	0	1	0	0
0	0	0	0	0

Fácilmente se puede observar que la salida de la función es '10' cuando I2 es '1', sin importar en lo más mínimo el valor de I1 e I0. Así, la tabla de verdad se puede re-escribir de manera reducida teniendo en cuenta lo recién explicado, quedando expresada de la siguiente manera:

Entrada			Salida	
I2	**I1**	**I0**	**D1**	**D0**
1	-	-	1	0
0	1	-	0	1
0	0	-	0	0

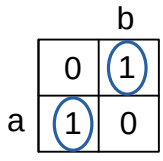
' - ' como salida en funciones Booleanas

En el caso de usar ' - ' como el valor que toma la función ante cierta condición de entrada, significa que realmente el valor de la salida 'no importa' que sea '0' o '1'. Aunque esto suene raro, como no va importar el valor de salida?!?!?!, en realidad se usa para los casos en que esa condición/valor particular de las entradas no es esperado, ni debería darse bajo ninguna circunstancia.

Veamos en detalle la siguiente tabla de verdad:

Entrada		Salida
a	**b**	**f**
0	0	0
0	1	1
1	0	1
1	1	-

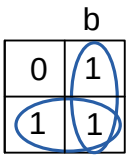
Para este caso cuando la entrada es '11' la salida puede ser '1' o '0'. Sí, por ejemplo cuando se implementa la función, y se le asigna valor '0' para la combinación de entradas '11', el mapa de Karnough de la función es el siguiente:



$f = a'b+ab'$

Y la función resultante de la lectura del mapa es, f = a'b + ab'

Cuando se le asigna un valor '1', el respectivo mapa de Karnough es:



f = a+b

y la función resultante: f = a+b .

Esta última función requiere menos hardware, por ello el uso de ' - ' puede ser importante en casos en que el número de componentes lógicos (compuertas, LUTs, etc) sea calve en el diseño que se quiere implementar.

Uso de ' - ' en VHDL

El valor ' - ' está definido en el tipo *std_logic*. SIN EMBARGO, VHDL considera ' - ' como un valor lógico en sí, totalmente diferente de lo explicado anteriormente. Es decir ' - ' no puede tomar valor '0' o '1' porque en sí tiene un valor, que es ' - '.

Así por ejemplo el siguiente código evalúa **siempre** como falso,

```
1       if data = "1- - 1" then ...;
```

La razón porque esta comparación es siempre falso es porque nunca en hardware existirá el valor "1 - - 1" para *data*.

Veamos entonces como hacer para que ' - ' pueda, de algún modo, ser usado como es usado en el álgebra de Boole, pudiendo tomar valor '1' o '0'.

Uso de ' - ' como entrada en código VHDL

Usando los conocimientos de VHDL (q se supone q tienes :) . . .) la tabla de verdad del decodificador con prioridad se podría describir como:

```
y <= "10" when I = "1--" else--siendo I la concatenación de I2 I1 I0
     "01" when I = "01-" else
     "00" when I = "001" else
     "00";
```

Escrito como está, este código NO describe un decodificador con prioridad, porque?... Bien, cuando este código es implementado la condición "1--" o la condición "01-" NUNCA es verdadera. Porque?... como ya se sabe físicamente en un circuito digital solo se tienen dos valores lógicos o '1' o '0'. Así, si por ejemplo la señal de entrada tienen un valor "111", ninguna expresión de la instrucción *when* será verdadera y el valor asignado a Y será el valor que se le asigna cuando ninguna de las anteriores es verdadera, es decir en este caso se le asigna "00'.

Una solución a este problema sería describir el decodificador con prioridad de la siguiente manera:

```
y <= "10" when I(2) = '1' else
     "01" when I(2 downto 1) = "01" else
     "00" when I(2 downto 0) = "001" else
     "00";
```

Cuya versión usando instrucción *if* en un proceso es la siguiente:

```
addr_dec_proc: process(I)
begin
  if ((I(2) = '1')then
     y <= "10";
  elsif (I(2 downto 1) = "01") then
     y <= "01";
  elsif (I(2 downto 0) = "001") then
     y <= "00";
  else
      y <= "00";
  endif;
end process addr_dec_proc;
```

Estas dos últimas versiones describen un poco mejor lo que se desea implementar. Sin embargo hay una manera de describir lo mismo pero usando el valor '-' de la forma que comúnmente se usa en el algebra de Boole.

En el paquete *numeric_std* hay una función llamada *std_match(),* que básicamente realiza una interpretación de ' -' de acuerdo a lo q el diseñador espera, es decir trata el valor ' - ' del tipo *std_logic* como valor '1' o '0'. La función compara dos vectores de tipo *std_logic_vector* e interpreta ' - ' como un valor 'no importa', y da como resultado un boolean (F/T).

Usando la función *std_match()* se puede re-escribir el primer ejemplo como:

```
1       if std_match(data, "1 - - 1") then . . . .
```

Lo que hace la función *std_match()* es comparar el vector *data* con el vector "1 - - 1", evaluando como verdadero si el primer y el último elemento de data son '1', sin interesar los otros dos valores.

Nota: std_match también puede usarse con tipos *unsigned, signed*, *std_ulogic, std_ulogic_vector, y std_logic.*

Del mismo modo el código del decodificador con prioridad anterior puede re-escribirse así:

```
1 use ieee.numeric_std.all;
2       ....
3
4 y <= "10" when std_match(I, "1--") else
5      "01" when std_match(I, "01-") else
6      "00" when std_match(I, "001") else
7      "00";
```

Uso de ' - ' como salida en código VHDL

Es bastante común el uso de ' - ' en una salida. Normalmente se lo utiliza como el valor que se la asigna a la salida cuando todos los valores de entradas han sido ya asignados y queda solo *'when others' , 'else'* ,etc. Por ejemplo la tabla de verdad descrita anteriormente se puede describir en VHDL de la siguiente manera:

```
1 tmp <= a & b;
2 with tmp select
3     y <= '0' when "00",
4          '1' when "01",
5          '1' when "10",
6          '-' when others;
```

De este modo el sintetizador puede utilizar el '-' como '1' o '0' para optimizar el hardware a implementar.