Cómo implementar un multiplexor de frecuencias para Xilinx

Creador: David Rubio G.

Entrada: https://soceame.wordpress.com/2024/12/18/como-implementar-un-multiplexor-de-frecuencias-para-xilinx/

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

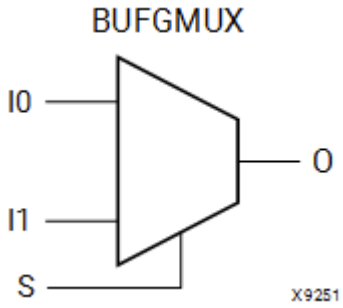
GitHub: https://github.com/DRubioG

Fecha última modificación: 23/02/2025

Una situación que puede ocurrir en una FPGA es tener dos relojes puros y querer multiplexarlos para poder conseguir que la misma lógica sea atacada por diferentes relojes.

Bien, pues debido a que se tratan de relojes puros (o sea, que no se han generado mediante un divisor de frecuencia por lógica programable) se tienen que utilizar las directivas de Xilinx propias para los relojes. Si no se utilizan estas directivas, lo único que vas a conseguir es generar una señal de habilitación, pensando que tienes un reloj a la salida.

Entonces, la directiva que da Xilinx para multiplexar un reloj es la **BUFGMUX**.



Esta directiva tiene la siguiente table de la verdad. En la que el valor 0 selecciona la entrada 0 y el valor 1 selecciona la entrada 1.

Logic Table

Inputs			Outputs
I0	I1	S	O
I0	X	0	I0
X	I1	1	I1
X	X	↑	0
X	X	↓	0

Para implementarlo, esta macro también necesita un campo CLK_SEL_TYPE, que solo tiene dos opciones **«SYNC»/»ASYNC»**, estas opciones indican para cuando se tiene que realizar el cambio en el multiplexor, si síncrono con el reloj o de forma asíncrona.

Para implementarlo Xilinx da esta estructura.

```
Library UNISIM;
use UNISIM.vcomponents.all;
```

```
-- BUFGMUX: General Clock Mux Buffer
--          Versal Premium series
-- Xilinx HDL Language Template, version 2024.2

BUFGMUX_inst : BUFGMUX
generic map (
   CLK_SEL_TYPE => "SYNC"  -- ASYNC, SYNC
)
port map (
   O => O,   -- 1-bit output: Clock output
   I0 => I0, -- 1-bit input: Clock input (S=0)
   I1 => I1, -- 1-bit input: Clock input (S=1)
   S => S    -- 1-bit input: Clock select
);

-- End of BUFGMUX_inst instantiation
```

Ejemplo

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VComponents.all;

entity mux_clk is
    Port (
        clk1 : in std_logic;
        clk2 : in std_logic;
        selection : in std_logic;
        clk_out : out std_logic
    );
end mux_clk;

architecture Behavioral of mux_clk is

signal clk1_buf, clk2_buf : std_logic;

begin

BUFG_clk1_inst : BUFG
port map (
   O => clk1_buf,
   I => clk1
);


BUFG_clk2_inst : BUFG
port map (
   O => clk2_buf,
   I => clk2
);

BUFGMUX_inst : BUFGMUX
    generic map (
       CLK_SEL_TYPE => "SYNC"
    )
    port map (
```

```
      O => clk_out,
      I0 => clk1_buf,
      I1 => clk2_buf,
      S => selection
   );

end Behavioral;
```

Nota: recordad siempre bufferear los relojes que vienen del exterior de la FPGA, en este caso con un **BUFG**.

Referencia

https://docs.amd.com/r/en-US/ug1485-versal-architecture-premium-series-libraries/BUFGMUX