

Cómo utilizar el analizador de máquinas de estados de Quartus

Creador: David Rubio G.

Entrada: <https://soceame.wordpress.com/2025/01/15/como-utilizar-el-analizador-de-maquinas-de-estados-de-quartus/>

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

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

Fecha última modificación: 24/02/2025

Quartus tiene internamente un analizador de máquinas de estados, que te muestra la máquina de estados sintetizada de nuestro proyecto.

Primero tenemos que tener nuestro código.

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY SM3 IS
    PORT (
        clock : IN STD_LOGIC;
        reset : IN STD_LOGIC := '0';
        a : IN STD_LOGIC := '0';
        b : OUT STD_LOGIC
    );
END SM3;

ARCHITECTURE BEHAVIOR OF SM3 IS
    TYPE type_fstate IS (S0,S1,S2);
    SIGNAL fstate : type_fstate;
    SIGNAL reg_fstate : type_fstate;
BEGIN
    PROCESS (clock,reg_fstate)
    BEGIN
        IF (clock='1' AND clock'event) THEN
            fstate <= reg_fstate;
        END IF;
    END PROCESS;

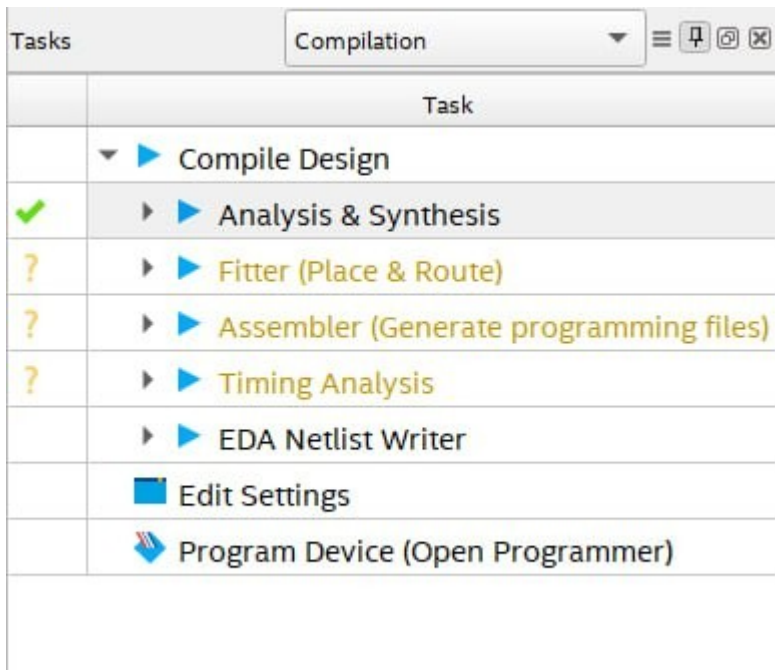
    PROCESS (fstate,reset,a)
    BEGIN
        IF (reset='1') THEN
            reg_fstate <= S0;
            b <= '0';
        ELSE
            b <= '0';
            CASE fstate IS
                WHEN S0 =>
                    IF ((a = '1')) THEN
                        reg_fstate <= S1;
                        -- Inserting 'else' block to prevent latch inference
                    ELSE
                        reg_fstate <= S0;
                    END IF;

                    b <= '0';
                WHEN S1 =>
                    IF (NOT((a = '1')))) THEN
                        reg_fstate <= S2;
                        -- Inserting 'else' block to prevent latch inference
                    ELSE
                        reg_fstate <= S1;
                    END IF;

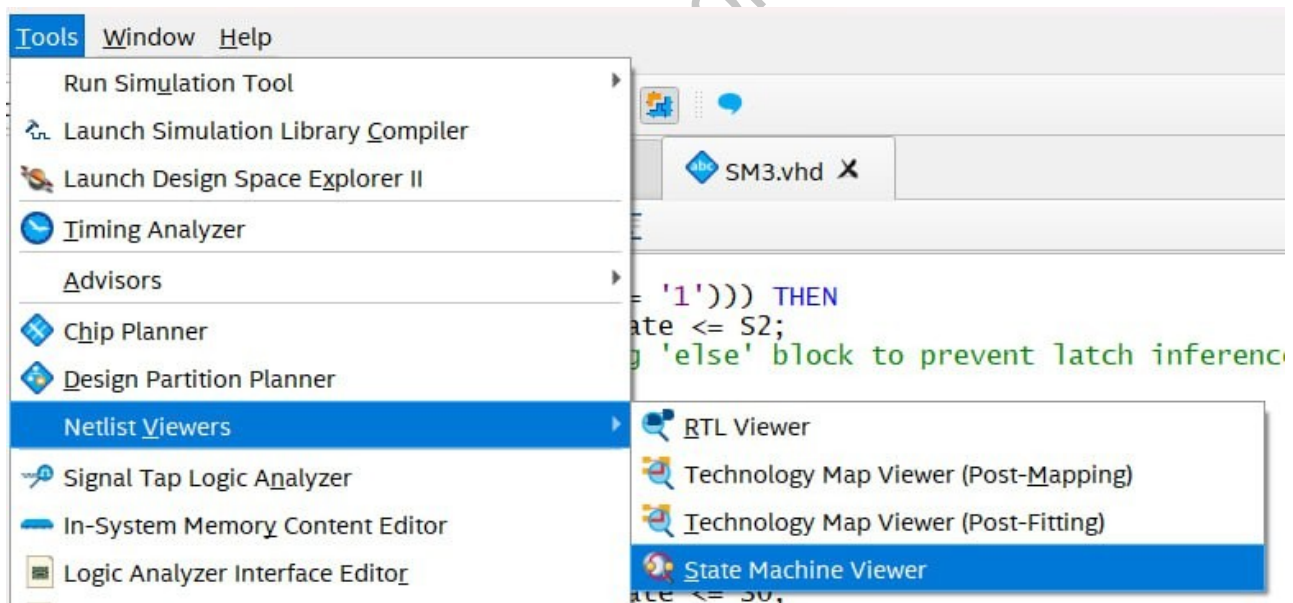
                    b <= '0';
                WHEN S2 =>
                    IF ((a = '1')) THEN
                        reg_fstate <= S0;
                        -- Inserting 'else' block to prevent latch inference
                    ELSE
                        reg_fstate <= S2;
                    END IF;

                    b <= '1';
                WHEN OTHERS =>
                    b <= 'X';
                    report "Reach undefined state";
            END CASE;
        END IF;
    END PROCESS;
END BEHAVIOR;
```

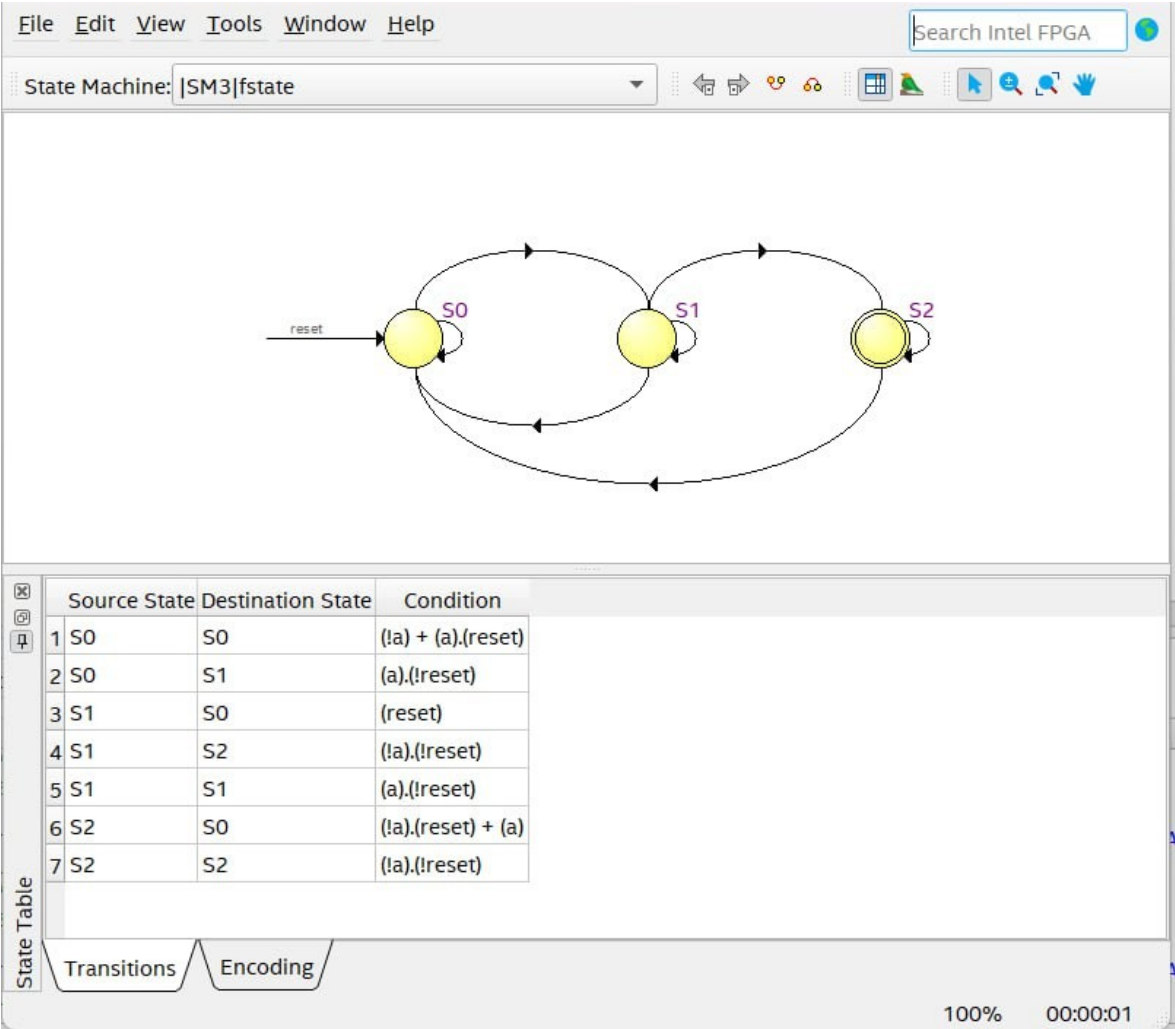
Después sintetizamos el código.



Una vez sintetizado, nos vamos a *Tools* en *Netlist Viewers* y buscamos *State Machine Viewer*.



Esto nos abre el una imagen con la máquina de estados sintetizada. Abajo figura la información de cambio de estados de la máquina.



También figura la codificación de la máquina de estados.

	Name	S2	S1	S0
1	S0	0	0	0
2	S1	0	1	1
3	S2	1	0	1

Transitions

Encoding