Instituto Tecnológico De Estudios Superiores de Monterrey

Laboratorio de Sistemas Digitales Avanzados

Teclado PS/2

Héctor Javier Pequeño A01246364

Prof. Emilio Caballero

Links Github:

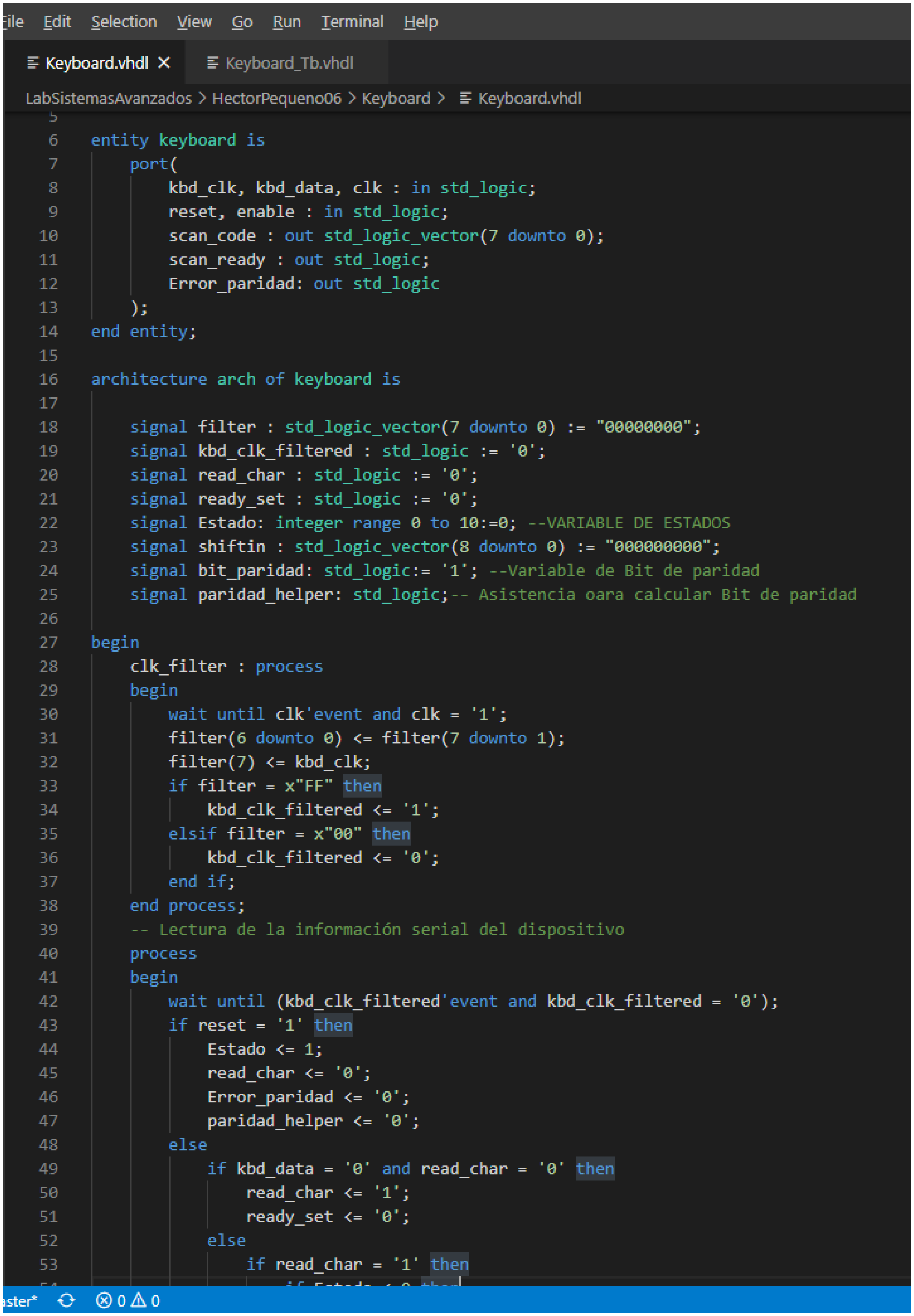
Código Fuente PS/2:

https://github.com/HectorPequeno06/HectorPequeno06/blob/master/Keyboard/Keyboard.vhdl

Código Test Bench PS/2:

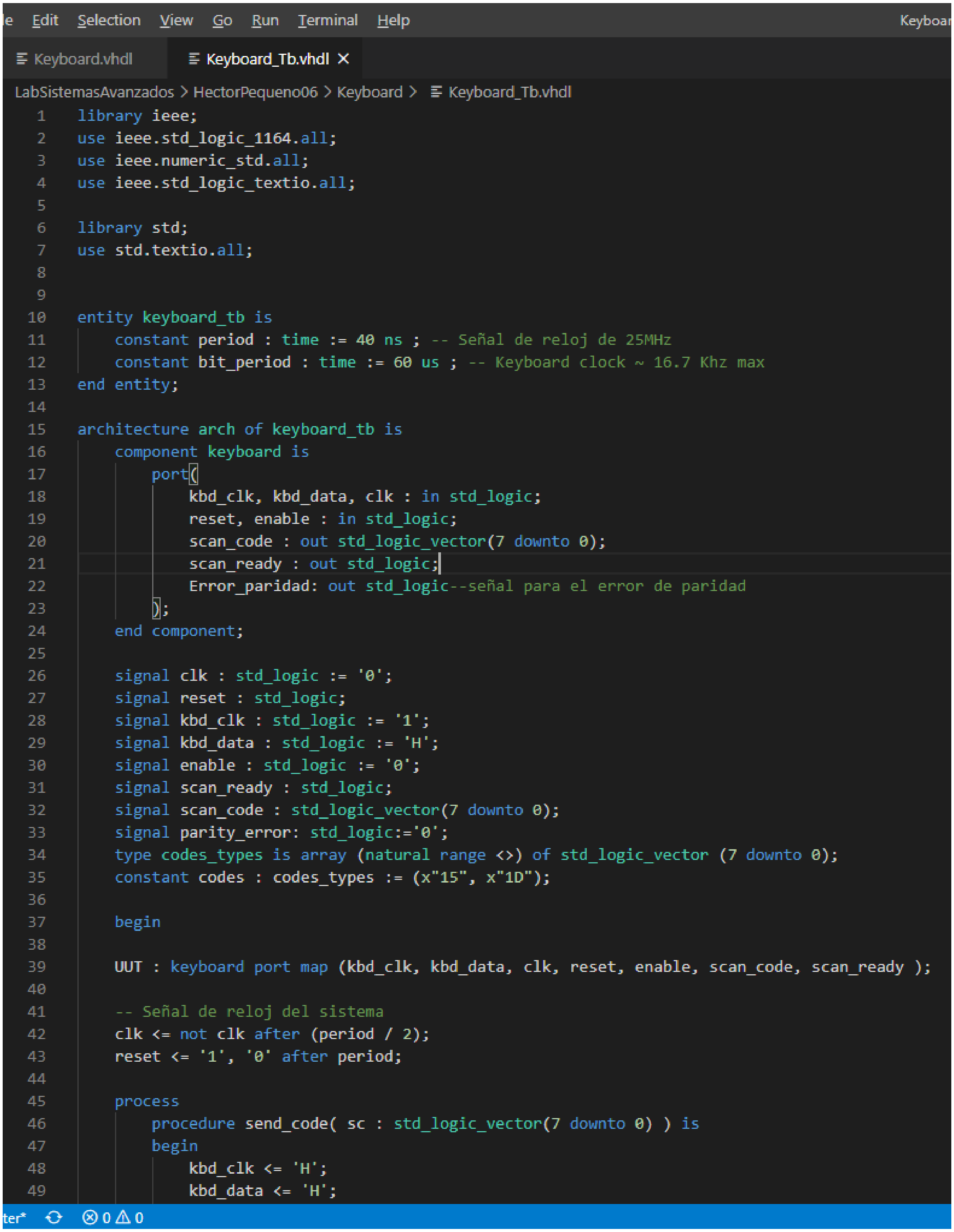
https://github.com/HectorPequeno06/HectorPequeno06/blob/master/Keyboard/Keyboard_Tb.vhdl

Teclado PS/2 Código Fuente:


File Edit Selection View Go Run Terminal Help
Keyboard.vhdl
Keyboard_Tb.vhdl
LabSistemasAvanzados > HectorPequeno06 > Keyboard > Keyboard.vhdl
aster*
0 0

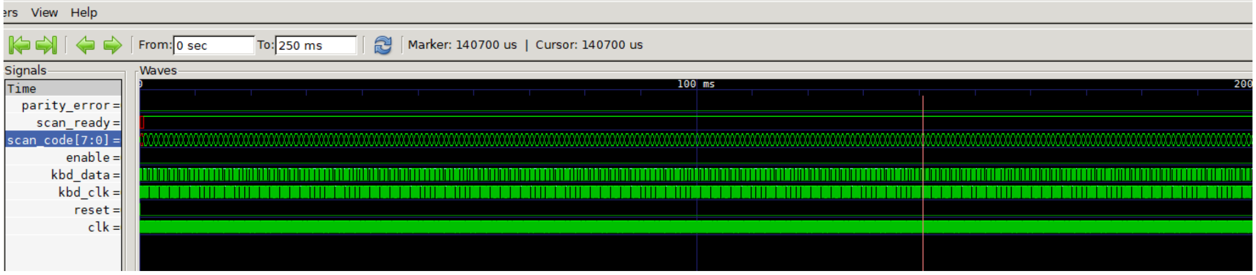
```vhdl
entity keyboard is
    port(
        kbd_clk, kbd_data, clk : in std_logic;
        reset, enable : in std_logic;
        scan_code : out std_logic_vector(7 downto 0);
        scan_ready : out std_logic;
        Error_paridad: out std_logic
    );
end entity;

architecture arch of keyboard is

    signal filter : std_logic_vector(7 downto 0) := "00000000";
    signal kbd_clk_filtered : std_logic := '0';
    signal read_char : std_logic := '0';
    signal ready_set : std_logic := '0';
    signal Estado: integer range 0 to 10:=0; --VARIABLE DE ESTADOS
    signal shiftin : std_logic_vector(8 downto 0) := "000000000";
    signal bit_paridad: std_logic:= '1'; --Variable de Bit de paridad
    signal paridad_helper: std_logic;-- Asistencia oara calcular Bit de paridad

begin
    clk_filter : process
    begin
        wait until clk'event and clk = '1';
        filter(6 downto 0) <= filter(7 downto 1);
        filter(7) <= kbd_clk;
        if filter = x"FF" then
            kbd_clk_filtered <= '1';
        elsif filter = x"00" then
            kbd_clk_filtered <= '0';
        end if;
    end process;
    -- Lectura de la información serial del dispositivo
    process
    begin
        wait until (kbd_clk_filtered'event and kbd_clk_filtered = '0');
        if reset = '1' then
            Estado <= 1;
            read_char <= '0';
            Error_paridad <= '0';
            paridad_helper <= '0';
        else
            if kbd_data = '0' and read_char = '0' then
                read_char <= '1';
                ready_set <= '0';
            else
                if read_char = '1' then
                    if Estado < 0 then
```

Teclado PS/2 código Test Bench:



```vhdl
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee.std_logic_textio.all;

library std;
use std.textio.all;


entity keyboard_tb is
    constant period : time := 40 ns ; -- Señal de reloj de 25MHz
    constant bit_period : time := 60 us ; -- Keyboard clock ~ 16.7 Khz max
end entity;

architecture arch of keyboard_tb is
    component keyboard is
        port(
            kbd_clk, kbd_data, clk : in std_logic;
            reset, enable : in std_logic;
            scan_code : out std_logic_vector(7 downto 0);
            scan_ready : out std_logic;
            Error_paridad: out std_logic--señal para el error de paridad
        );
    end component;

    signal clk : std_logic := '0';
    signal reset : std_logic;
    signal kbd_clk : std_logic := '1';
    signal kbd_data : std_logic := 'H';
    signal enable : std_logic := '0';
    signal scan_ready : std_logic;
    signal scan_code : std_logic_vector(7 downto 0);
    signal parity_error: std_logic:='0';
    type codes_types is array (natural range <>) of std_logic_vector (7 downto 0);
    constant codes : codes_types := (x"15", x"1D");

    begin

    UUT : keyboard port map (kbd_clk, kbd_data, clk, reset, enable, scan_code, scan_ready );

    -- Señal de reloj del sistema
    clk <= not clk after (period / 2);
    reset <= '1', '0' after period;

    process
        procedure send_code( sc : std_logic_vector(7 downto 0) ) is
        begin
            kbd_clk <= 'H';
            kbd_data <= 'H';
```

Simulación PS/2


ers View Help
From: 0 sec
To: 250 ms
Marker: 140700 us | Cursor: 140700 us
Signals
Waves
Time
parity_error =
scan_ready =
scan_code[7:0] =
enable =
kbd_data =
kbd_clk =
reset =
clk =
0
100 ms
200