PROYECTO FINAL DE DISEÑO DE SISTEMAS DIGITALES

TECNOLÓGICO
DE MONTERREY ®

ENERO-MAYO 2008

Profesora:
Ing. Norma Frida Roffe

Intructores:
Marco Quezada y Jorge Villaseñor

Alumnos:
Antonio Reyes Lúa (796600)
Daniel Arturo Nuñez González (1052478)
Alejandro Domínguez Medina(509550)

INDICE

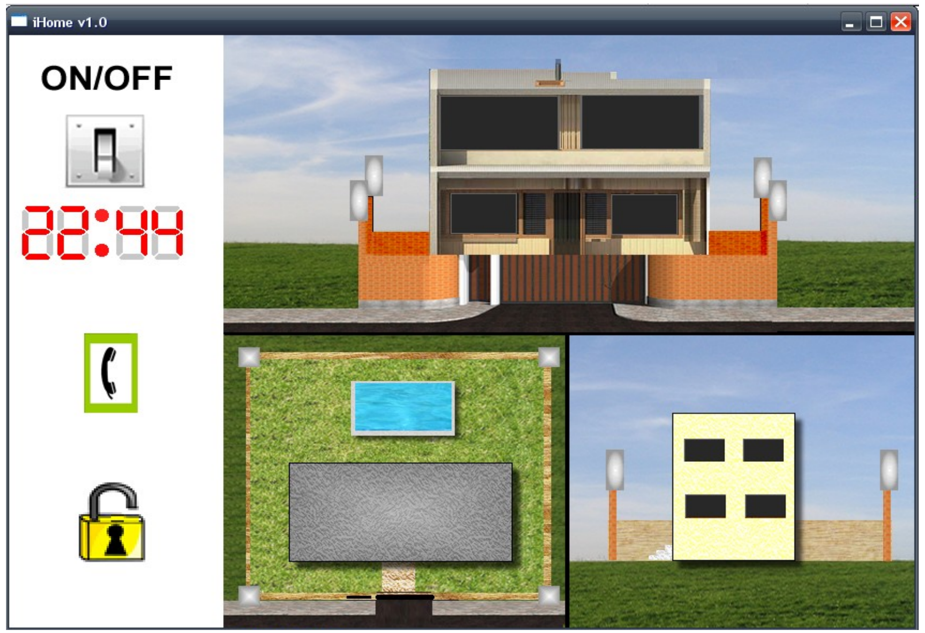
Descripción

Este proyecto fue diseñado para simular la automatización de diversos dispositivos de una casa como lo son las luces de los cuartos o de los faros, el sistema de alarmas, las puertas, el portón y el jacuzzi de la alberca. Para ello se implementó una interfáz gráfica en C que controlara desde la PC a los diversos dispositivos. A través de ella es posible encender y apagar cualquier elemento de la casa con un click o programar horarios para estas funciones.

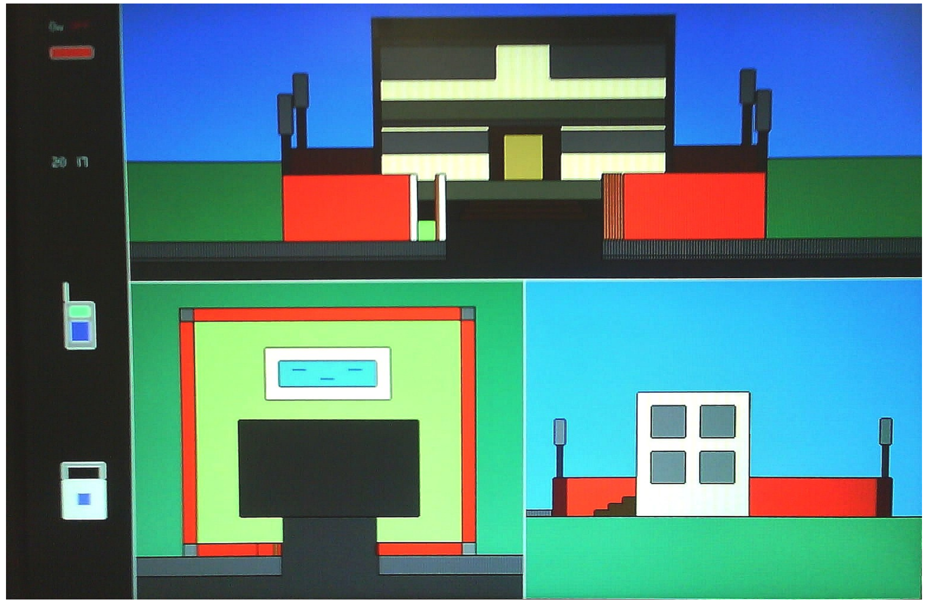
- Las luces de los cuartos y los faros tienen 4 intensidades: nula, baja, media, alta.
- El jacuzzi tiene 2 modos: encendido y apagado.
- Las puertas y portón tienen 2 modos: abierto y cerrado.
- La alarma tiene 3 modos: desactivada, activada, emergencia en curso.

El VGA pretende simular lo que sería la “casa de verdad” que es controlada. De esta manera, todas las instrucciones generadas en la PC a través de la interfáz gráfica se ven reflejadas en el VGA. De igual forma, al ser el VGA la “casa de verdad”, todos los cambios realizados en ella se ven igualmente reflejados en la PC pues la interfaz es un “sistema de monitoreo en tiempo real”. La forma de realizar estos cambios es a través del teclado:

- Luces cuartos:
    - Interruptor LUCES en '1'
    - Número del 1 al 6(dependiendo de cuál se quiera encender) ó
    - Número del 7 al 9(todo el Piso1, Piso2 o TODAS)
    - ENTER
- Luces faros
    - Interruptor FAROS en '1'
    - Número del 1 al 4(dependiendo de cuál se quiera encender) ó
    - Número del 7 al 9(todas las del FRENTE, del FONDO o TODAS)
    - ENTER
- Puertas: 'F' y ENTER
- Portón: 'G' y ENTER
- Alarma: 'A' y ENTER
- Sensor de Robo: 'R' y ENTER
- Jacuzzi: 'L' y ENTER



Interfáz Gráfica en C



Simulación en VGA

Dispositivos utilizados

- ✔ *Spartan3*

- ✔ *Cable USB J-TAG*

- ✔ *Nexys VGA Module*

- ✔ *Cable serial*

- ✔ *Monitor VGA*

- ✔ *Teclado*

Código

CASA

```
----------------------------------------------------------------------------------
-- Company:      Diseño de Sistemas Digitales
-- Engineers:    Antonio Reyes Lúa
--               Daniel Arturo Nuñez
--                Alejandro Dominguez
--
-- Create Date:  19:22:32 05/03/2008
-- Design Name:  Simulación de Casa Automatizada
-- Module Name:  casaPF - Behavioral
-- Project Name: Proyecto Final de Diseño de Sistemas Digitales
--
-- Description:     Este código simula la automatización de una casa a través de un monitor de VGA.
-- Dependencies: Este código hace uso del Display del Spartan, de un teclado externo, de un adaptador
--                para SuperVGA, de un cable serial y de una interfaz gráfica auxiliar escrita en C.

----------------------------------------------------------------------------------
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity casaPF is
Port (  CLK             : in  STD_LOGIC;
        --interruptores
        iModo           : in  STD_LOGIC;
        iLuces          : in  STD_LOGIC;
        iFaros          : in  STD_LOGIC;
        --Teclado
        kb_clk          : in      STD_LOGIC;
        kb_data         : in      STD_LOGIC;
        --Serial
        ioRx            : in  STD_LOGIC;
        ioTx            : out STD_LOGIC;
        --Display
        salBCD          : out STD_LOGIC_VECTOR (7 downto 0);
        ENE             : out STD_LOGIC_VECTOR (3 downto 0);
        --LEDs
        ledprueba: out std_logic;
        ledprueba2: out std_logic;
        ledprueba3: out std_logic;
        --VGA
        HS              : out std_logic;
        VS              : out std_logic;
```

```
          Green           : out STD_LOGIC_VECTOR (3 downto 0);
          Blue            : out STD_LOGIC_VECTOR (3 downto 0);
          Red             : out STD_LOGIC_VECTOR (3 downto 0));
end casaPF;

architecture Behavioral of casaPF is

--**********COMPONENTES**************************************************
component CLKs is
          Port (  Clk_in : in  STD_LOGIC;
         CLK_out: out STD_LOGIC);
end component;

component contador is
    Port ( inc     : in std_logic;
         reset    : in std_logic;
          run     : in std_logic;
         s        : inout std_logic_vector(9 downto 0));
end component;

component contadorHr is
    Port ( inc     : in std_logic;
         reset    : in std_logic;
          run     : in std_logic;
          set     : in std_logic;
          newTime: in std_logic_vector(3 downto 0);
         s        : inout std_logic_vector(3 downto 0));
end component;

component testram is
Port (
          address: in std_logic_vector( 6 downto 0 );
          data    : out std_logic_vector( 3 downto 0 ));
end component;

component keyboard is
          port(    keyboard_clk   : IN     STD_LOGIC;
                   keyboard_data  : IN     STD_LOGIC;
                   clock_25Mhz    : IN     STD_LOGIC;
                   reset          : IN     STD_LOGIC;
                   read_kb        : IN     STD_LOGIC;
                   scan_code      : OUT    STD_LOGIC_VECTOR(7 DOWNTO 0);
                   scan_ready     : OUT    STD_LOGIC);
end component;

component Display is
```

```
Port (
        -- Input
        Clk_in   : in  STD_LOGIC;                    -- Clock
        -- Data
        D0: in  STD_LOGIC_VECTOR (3 downto 0);
        D1: in  STD_LOGIC_VECTOR (3 downto 0);
        D2: in  STD_LOGIC_VECTOR (3 downto 0);
        D3: in  STD_LOGIC_VECTOR (3 downto 0);

        -- Output
        D7: out  STD_LOGIC_VECTOR (7 downto 0);      -- Data
        En: out  STD_LOGIC_VECTOR (3 downto 0)       -- Enable
                );
end component;

component uart_tx is
   Port (   data_in               : in std_logic_vector(7 downto 0);
            write_buffer          : in std_logic;
            reset_buffer          : in std_logic;
            en_16_x_baud          : in std_logic;
            serial_out            : out std_logic;
            buffer_full           : out std_logic;
            buffer_half_full      : out std_logic;
            clk                   : in std_logic);
end component;

component uart_rx is
   Port (   serial_in             : in std_logic;
            data_out              : out std_logic_vector(7 downto 0);
            read_buffer           : in std_logic;
            reset_buffer          : in std_logic;
            en_16_x_baud          : in std_logic;
            buffer_data_present   : out std_logic;
            buffer_full           : out std_logic;
            buffer_half_full      : out std_logic;
            clk                   : in std_logic);
end component;

--***********************************************************************

--************SEÑALES**************************************************

--Serial=================================================
--//señales necesarias para los módulos UART del paquete de PICOBLAZE para trans. serial bajado de la página de Xilinx
signal baud_count: integer range 0 to 650:=0;
signal en_16_x_baud: std_logic;
signal datoTx                  : std_logic_vector(7 downto 0);
signal wbufferTx               : std_logic;
signal resetbufferTx           : std_logic;
signal bufferFTx               : std_logic;
signal bufferHFTx              : std_logic;

signal datoRx                  : std_logic_vector(7 downto 0);
signal rbufferRx               : std_logic;
signal resetbufferRx           : std_logic;
signal bufferReadyRx           : std_logic;
signal bufferFRx               : std_logic;
signal bufferHFRx              : std_logic;
--

signal sendAlgo:std_logic; --señal para indicar que se ha cambiado un valor y es necesario transmitirlo serialmente al programa en C

--//paquetes Tx y Rx utilizados (55:id:C1:C2:C3:C4:AA)
signal dato55Tx        :std_logic_vector(7 downto 0):="01010101";
signal datoIdTx        :std_logic_vector(7 downto 0):="00000010";
signal datoC1Tx        :std_logic_vector(7 downto 0); --quién: Luces, Faros, Seguridad, Puertas...
signal datoC2Tx        :std_logic_vector(7 downto 0); --cuál : ej(la de arriba a la izquierda, la de atras al fondo...)
signal datoC3Tx        :std_logic_vector(7 downto 0); --cómo : intensidad nula, intensidad 1, intensidad 2, intensidad 3
signal datoC4Tx        :std_logic_vector(7 downto 0):="00000000";
signal datoAATx        :std_logic_vector(7 downto 0):="10101010";

signal dato55Rx        :std_logic_vector(7 downto 0);
signal datoIdRx        :std_logic_vector(7 downto 0);
signal datoC1Rx        :std_logic_vector(7 downto 0);--quién
signal datoC2Rx        :std_logic_vector(7 downto 0);--cuál
signal datoC3Rx        :std_logic_vector(7 downto 0);--cómo
signal datoC4Rx        :std_logic_vector(7 downto 0);
signal datoAARx        :std_logic_vector(7 downto 0);

--//señales de los Estados para la transmisión y recepción de datos
signal stateTx,nstateTx: integer range 0 to 15:=0;
signal stateRx,nstateRx: integer range 0 to 15:=0;

signal setNewValueRx:std_logic:='0';
signal horaHRx, horaLRx, minHRx, minLRx:std_logic_vector(3 downto 0):="0000"; --valores de la hora que se recibe serialmente
signal setTimeRx:std_logic; --funciona como LOAD para cargar la hora recibida por serial al registro contador de la hora

--CLKs=================================================

signal clk25 : std_logic; --25MHZ
signal CLK1s : std_logic; --1HZ
signal cont1seg: STD_LOGIC_VECTOR(25 downto 0);

signal CLKmedioS : std_logic;--2HZ --
signal contMedioSeg: STD_LOGIC_VECTOR(25 downto 0);

signal CLK1min: std_logic;
signal cont1min: STD_LOGIC_VECTOR(25 downto 0);

signal msDelay, sDelay: STD_LOGIC_VECTOR(25 downto 0);
signal clk_lento: std_logic_vector(20 downto 0);

signal clkMov,clkMov2,clkMov3: std_logic;
signal contMov3: STD_LOGIC_VECTOR(25 downto 0);

signal CLKcuadro : std_logic;
signal contCuadro: STD_LOGIC_VECTOR(25 downto 0);

--TECLADO================================================

signal kbcode   : std_logic_vector(7 downto 0);
signal kbready  : std_logic;
signal resetKB  : std_logic;
signal readKB   : std_logic  :='1';

signal teclaKB: std_logic_vector(3 downto 0):="1010"; --comienza en 1010 pues es el valor que tiene el módulo del Display  para cuando no se dibuja nada sobre él
signal enterPUSHED, enterOFF:std_logic; --banderas que indican cuando está siendo presionado el ENTER y cuando es levantada


--Display================================================

signal disp1,disp2,disp3,disp4: std_logic_vector(3 downto 0):="1010";

--Señales de RAM--========================================

--**relojito

signal hrH: std_logic_vector( 3 downto 0 ); --decenas hora
signal hrH_address: std_logic_vector( 6 downto 0 );
signal hrL: std_logic_vector( 3 downto 0 ); --unidades hora
signal hrL_address: std_logic_vector( 6 downto 0 );
signal minH: std_logic_vector( 3 downto 0 ); --decenas min
signal minH_address: std_logic_vector( 6 downto 0 );
signal minL: std_logic_vector( 3 downto 0 ); --unidades min
signal minL_address: std_logic_vector( 6 downto 0 );
signal dosPuntos: std_logic_vector( 3 downto 0 );
signal dosPuntos_address: std_logic_vector( 6 downto 0 );
signal onHoraLB, onHrH, onHrL, onMinH, onMinL, on2p:std_logic; --banderas que indican si el pixel que se va a pintar es alguno de los dígitos de la hora

--//registros contadores que llevan la cuenta de la hora
signal contHrH: std_logic_vector(3 downto 0);
signal contHrL: std_logic_vector(3 downto 0);
signal contMinH: std_logic_vector(3 downto 0);
signal contMinL: std_logic_vector(3 downto 0);

--//señales de reset de los contadores
signal resetMinL,resetMinH,resetHrL,resetHrH:std_logic;


--**interruptor ON/OFF

signal onO: std_logic_vector( 3 downto 0 );
signal onO_address: std_logic_vector( 6 downto 0 );
signal onN: std_logic_vector( 3 downto 0 );
signal onN_address: std_logic_vector( 6 downto 0 );
signal offO: std_logic_vector( 3 downto 0 );
signal offO_address: std_logic_vector( 6 downto 0 );
signal offF1: std_logic_vector( 3 downto 0 );
signal offF1_address: std_logic_vector( 6 downto 0 );
signal offF2: std_logic_vector( 3 downto 0 );
signal offF2_address: std_logic_vector( 6 downto 0 );
signal onON_o,onON_n,onOFF_o,onOFF_f1,onOFF_f2, onLetreroON, onLetreroOFF:std_logic; --banderas que indican si el pixel que se va a pintar es alguno de las letras de "ON/OFF"




--Señal de VGA===========================================================

--//señal principal
signal Hcont,Vcont: std_logic_vector(9 downto 0);
signal resetH, resetV, HSaux,VSaux: std_logic;

--//señales de control de intensidades y estados de encendido
signal edoAlarma,edoAlberca,edoRobo,edoPorton,edoPuerta,edoSistema:std_logic:='0'; --empiezan en modo Apagado

signal intFaroFL:std_logic_vector(1 downto 0):="00"; --faro FrontLeft
signal intFaroFR:std_logic_vector(1 downto 0):="00"; --faro FrontRight
signal intFaroBL:std_logic_vector(1 downto 0):="00"; --faro BackLeft
signal intFaroBR:std_logic_vector(1 downto 0):="00"; --faro BackRight

signal intventanaULF:std_logic_vector(1 downto 0):="00"; --ventana UpLeftFront
signal intventanaDLF:std_logic_vector(1 downto 0):="00"; --ventana DownLeftFront
signal intventanaURF:std_logic_vector(1 downto 0):="00"; --ventana UpRightFront
signal intventanaDRF:std_logic_vector(1 downto 0):="00"; --ventana DownRightFront
signal intventanaURB:std_logic_vector(1 downto 0):="00"; --ventana UpLeftBack
signal intventanaDRB:std_logic_vector(1 downto 0):="00"; --ventana DownRightBack

signal banderaTodasLucesIgual:std_logic;
signal banderaTodosFarosIgual:std_logic;
signal banderaTodasLucesP1Igual:std_logic;
signal banderaTodosFarosFIgual:std_logic;
signal banderaTodasLucesP2Igual:std_logic;
signal banderaTodosFarosBIgual:std_logic;

--//colores recuadros
signal RGBc1: std_logic_vector(11 downto 0); --señal de RGB del recaudro1
signal RGBc2: std_logic_vector(11 downto 0); --señal de RGB del recaudro2
signal RGBc3: std_logic_vector(11 downto 0); --señal de RGB del recaudro3
signal RGBlb: std_logic_vector(11 downto 0); --señal de RGB de la barra lateral

signal RGBhrH: std_logic_vector(11 downto 0);    --señal de RGB de las decenas de horas
signal RGBhrL: std_logic_vector(11 downto 0);    --señal de RGB de las unidades de horas
signal RGBminH: std_logic_vector(11 downto 0); --señal de RGB de las decenas de minutos
signal RGBminL: std_logic_vector(11 downto 0);  --señal de RGB de las unidades de minutos
signal RGB2p: std_logic_vector(11 downto 0);     --señal de RGB de los 2 puntos del reloj

signal RGBletreroONverde: std_logic_vector(11 downto 0):="000011110000";
signal RGBonO, RGBonN:std_logic_vector(11 downto 0); --señal de RGB de "ON"
signal RGBletreroOFFrojo: std_logic_vector(11 downto 0):="111100000000";
signal RGBoffO, RGBoffF1, RGBoffF2:std_logic_vector(11 downto 0); --señal de RGB de "OFF"

--RGB auxiliares de C1
signal RGBcalleC1: std_logic_vector(11 downto 0):="110011001100";
signal RGBbanquetaC1: std_logic_vector(11 downto 0):="110011001100";
signal RGBbordeBanquetaC1: std_logic_vector(11 downto 0):="111011101110";
signal RGBpastoFueraC1: std_logic_vector(11 downto 0):="000000010010";
signal RGBfaroFLc1: std_logic_vector(11 downto 0);
signal RGBfaroFRc1: std_logic_vector(11 downto 0);
signal RGBfaroBLc1: std_logic_vector(11 downto 0);
signal RGBfaroBRc1: std_logic_vector(11 downto 0);
signal RGBanyPoste: std_logic_vector(11 downto 0):="110011001100";
signal RGBporton:std_logic_vector(11 downto 0):="110000000000";
signal RGBparedPuertaFrenteC1:std_logic_vector(11 downto 0):="111111111111";
signal RGBmuroFC1:std_logic_vector(11 downto 0):="111100000000";
signal RGBmuroBC1:std_logic_vector(11 downto 0):="111100000000";
signal RGBmurosCurvosC1:std_logic_vector(11 downto 0):="111000000000";
signal RGBventanaULFc1:std_logic_vector(11 downto 0);
signal RGBventanaURFc1:std_logic_vector(11 downto 0);
signal RGBventanaDLFc1:std_logic_vector(11 downto 0);
signal RGBventanaDRFc1:std_logic_vector(11 downto 0);
signal RGBmurosLateralesC1:std_logic_vector(11 downto 0):="111011001000";
signal RGBmurosLateralesTechoC1:std_logic_vector(11 downto 0):="110010001000";
signal RGBtechitosC1:std_logic_vector(11 downto 0):="111011101110";
signal RGBbalconesC1:std_logic_vector(11 downto 0):="111011101110";
signal RGBazoteaC1:std_logic_vector(11 downto 0):="110011001100";
signal RGBpuertaCasaC1:std_logic_vector(11 downto 0):="110111011000";
signal RGBmarcoVentanasYpuertaC1:std_logic_vector(11 downto 0):="110011000000";
signal RGBfachadaCasa:std_logic_vector(11 downto 0):="111111110011";
signal RGBescalonesC1:std_logic_vector(11 downto 0):="111011001000";
signal RGBfachadaEscalonesC1:std_logic_vector(11 downto 0):="110011001100";
signal RGBpastoDentroC1:std_logic_vector(11 downto 0):="000101110110";
signal RGBcieloC1:std_logic_vector(11 downto 0):="000000010111";

--RGB auxiliares de C2
signal RGBcalleC2:std_logic_vector(11 downto 0):="110011001100";
signal RGBbanquetaC2:std_logic_vector(11 downto 0):="100010001000";
signal RGBazoteaC2:std_logic_vector(11 downto 0):="110011001100";
signal RGBalbercaC2:std_logic_vector(11 downto 0):="000000110011";
signal RGBmarcoAlbercaC2:std_logic_vector(11 downto 0):="111111110011";
signal RGBpastoFueraC2:std_logic_vector(11 downto 0):="000000011010";
signal RGBpastoDentroC2:std_logic_vector(11 downto 0):="000101110110";
signal RGBfaroFLc2: std_logic_vector(11 downto 0);
signal RGBfaroFRc2: std_logic_vector(11 downto 0);
signal RGBfaroBLc2: std_logic_vector(11 downto 0);
signal RGBfaroBRc2: std_logic_vector(11 downto 0);
signal RGBolasON: std_logic_vector(11 downto 0);
signal RGBolasOFF: std_logic_vector(11 downto 0);
signal RGBolasONaux: std_logic_vector(11 downto 0);
signal RGBolasOFFaux: std_logic_vector(11 downto 0);

--RGB auxiliares de C3
signal RGBventanaDRBc3:std_logic_vector(11 downto 0);
signal RGBventanaDRFc3:std_logic_vector(11 downto 0);
signal RGBventanaURBc3:std_logic_vector(11 downto 0);
signal RGBventanaURFc3:std_logic_vector(11 downto 0);
signal RGBpastoFueraC3:std_logic_vector(11 downto 0):="000000011010";
signal RGBbanquetaC3:std_logic_vector(11 downto 0):="100010001000";
signal RGBmurosVc3:std_logic_vector(11 downto 0):="111100000000";
signal RGBfaroFRc3: std_logic_vector(11 downto 0);
signal RGBfaroBRc3: std_logic_vector(11 downto 0);
signal RGBescalonesC3: std_logic_vector(11 downto 0):="111011001000";
signal RGBmuroHc3: std_logic_vector(11 downto 0);
signal RGBcieloC3: std_logic_vector(11 downto 0):="000000010111";
```

```
--RGB auxiliares de LB
signal RGBinterruptor: std_logic_vector(11 downto 0);
signal RGBcelular,RGBcelularAUX: std_logic_vector(11 downto 0);
signal RGBcelTeclado:std_logic_vector(11 downto 0);
signal RGBcelPantalla:std_logic_vector(11 downto 0);
signal RGBlockCentro:std_logic_vector(11 downto 0);
signal RGBlockCandado:std_logic_vector(11 downto 0);
signal RGBlockArco:std_logic_vector(11 downto 0);


--//banderas sobre donde se está==========================================
signal onScreen: std_logic;
signal onLeftBar: std_logic;
signal onCap1, onCap2, onCap3: std_logic;
signal onAnyCap: std_logic;

signal onLeftBarFrame: std_logic;
signal onMiddleCapsHFrame: std_logic;
signal onMiddleCapsVFrame: std_logic;
signal onCapsFrame: std_logic;

signal onFrameC1, onFrameC2, onFrameC3: std_logic;

----***banderas de leftBar:
signal interruptorLB:std_logic;
signal onFrameInterruptor:std_logic;

signal celAntenaLB:std_logic;
signal celPantallaLB:std_logic;
signal celTecladoLB:std_logic;
signal celCarcazaLB:std_logic;
signal onFrameCel:std_logic;

signal lockCentroLB:std_logic;
signal lockCandadoLB:std_logic;
signal lockRVarcoLB:std_logic;
signal lockHarcoLB:std_logic;
signal lockLV1arcoLB:std_logic;
signal lockLV2arcoLB:std_logic;
signal onFrameLock:std_logic;

----***banderas de c1:
signal cieloC1 :std_logic;
signal pastoFueraC1:std_logic;
signal banquetaC1, bordeBanquetaC1:std_logic;
signal calleC1:std_logic;
signal faroFRc1,faroBRc1,faroFLc1,faroBLc1,anyFaroC1:std_logic;
signal posteFRc1,posteBRc1,posteFLc1,posteBLc1,anyPosteC1:std_logic;
signal muroFc1, muroBc1, muroFcurvoC1:std_logic;
signal muroRcasaC1, muroRtechoCasaC1, muroLcasaC1,muroLtechoCasaC1:std_logic;
signal fachadaCasaC1:std_logic;
signal azoteaC1:std_logic;
signal techoUC1, techoDC1:std_logic;
signal balconUC1,balconDC1:std_logic;
signal paredPuertaFrenteC1,puertaFrenteC1,puertaCasaC1,portonC1,bordePuertaFrenteC1:std_logic;
signal ventanaURFc1,ventanaULFc1,ventanaDRFc1,ventanaDLFc1,marcoVentanasYpuertaC1,anyVentanaC1:std_logic;
signal escalon1C1,escalon2C1,escalon3C1,anyEscalonC1:std_logic;
signal fachadaCasaEscalonesC1: std_logic;
signal pastoDentroC1:std_logic;
signal anyElementoC1:std_logic;


----***banderas de c2:
signal calleC2,banquetaC2:std_logic;
signal puertaFrenteC2, portonC2:std_logic;
signal murosVc2, murosHc2:std_logic;
signal pastoFueraC2, pastoDentroC2:std_logic;
signal azoteaC2:std_logic;
signal caminitoC2:std_logic;
signal albercaC2, marcoAlbercaC2:std_logic;
signal faroFDc2, faroFIc2, faroBDc2, faroBIc2:std_logic;
signal ola1,ola2,ola3,ola4,ola5,ola6,anyOla:std_logic;

----***banderas de c3:
signal cieloC3, pastoFueraC3, banquetaC3:std_logic;
signal murosVc3, muroHc3, muroRcasaC3:std_logic;
signal faroFRc3, faroBRc3:std_logic;
signal posteFRc3, posteBRc3:std_logic;
signal ventanaURFc3, ventanaURBc3, ventanaDRFc3, ventanaDRBc3,anyVentanaC3:std_logic;
signal escalon1c3, escalon2c3, escalon3c3,anyEscalonC3:std_logic;


--// "constantes" de limites===========================================================
signal limCapsD: integer:=80;
signal limCapsMv:integer:=360;
signal limCapsMh:integer:=240;


----***limites de LeftBar***********************************************

--lim del interruptor============
signal limLinterruptorLB:integer:=30;
signal limRinterruptorLB:integer:=60;
signal limSUPinterruptorLB:integer:=65;
signal limINFinterruptorLB:integer:=73;

signal limSupLetreroONOFF: integer:=45;
signal limInfLetreroONOFF: integer:=53;

signal limLON_o: integer:=30 ;
signal limRON_o: integer:=34 ;
signal limLON_n: integer:=35 ;
signal limRON_n: integer:=39 ;

signal limLOFF_o: integer:=45 ;
signal limROFF_o: integer:=49 ;
signal limLOFF_f1: integer:=50 ;
signal limROFF_f1: integer:=54 ;
signal limLOFF_f2: integer:=55 ;
signal limROFF_f2: integer:=59 ;


--limdel celular==================

signal limRcelAntenaLB:integer:=37;
signal limSUPcelAntenaLB:integer:=240;

signal limSUPcelPantallaLB:integer:=258;
signal limINFcelPantallaLB:integer:=266;

signal limSUPcelTecladoLB:integer:=269;
signal limINFcelTecladoLB:integer:=285;

signal limSUPcelCarcazaLB:integer:=255;
signal limINFcelCarcazaLB:integer:=290;
signal limLcelCarcazaLB:integer:=35;
signal limRcelCarcazaLB:integer:=55;

--lims del CANDADO======
signal limLlockCandadoLB:integer:=30;
signal limRlockCandadoLB:integer:=60;
signal limSUPlockCandadoLB:integer:=390;
signal limINFlockCandadoLB:integer:=420;

signal limRlockRVarcoLB:integer:=33;
signal limSUPlockRVarcoLB:integer:=380;
signal limSUPlockHarcoLB:integer:=377;

signal limLlockLVarcoLB:integer:=57;
signal limINFlockLVarcoLB:integer:=385;

--lims de la HORA============
signal limSupHora: integer:=146;
signal limInfHora: integer:=154;

signal limLhrH: integer:=30 ;
signal limRhrH: integer:=34 ;
signal limLhrL: integer:=35 ;
signal limRhrL: integer:=39 ;

signal limLdosPuntos: integer:=40 ;
signal limRdosPuntos: integer:=44 ;

signal limLminH: integer:=45 ;
signal limRminH: integer:=49 ;
signal limLminL: integer:=50 ;
signal limRminL: integer:=54 ;


----***limites de c1****************************************************
signal limHorizonteC1   :integer:=150;
signal limSupBanquetaC1 :integer:=210;
signal limInfBanquetaC1 :integer:=225;
signal limLentreBanquetasC1  :integer:=305;
signal limRentreBanquetasC1  :integer:=415;

signal limLMuroFrente1C1 :integer:=190;
signal limRMuroFrente1C1 :integer:=305;
signal limLMuroFrente2C1 :integer:=430;
signal limRMuroFrente2C1 :integer:=530;
signal limSupMuroFrenteC1:integer:=160;

signal limLMuroAtras1C1 :integer:=201;
signal limRMuroAtras1C1 :integer:=255;
signal limLMuroAtras2C1 :integer:=465;
signal limRMuroAtras2C1 :integer:=519;
signal limSupMuroAtrasC1:integer:=140;

signal limLPuertaFrenteC1  :integer:=285;
signal limRPuertaFrenteC1  :integer:=300;
signal limLPortonC1          :integer:=305;
signal limRPortonC1          :integer:=430;
signal xPuertaFrenteC1      :std_logic_vector(8 downto 0);
signal xPortonC1            :std_logic_vector(8 downto 0);

signal limLPuertaFrenteC1aux:std_logic_vector(8 downto 0);
signal limLPortonC1aux:std_logic_vector(8 downto 0);


signal limRPosteFLc1    :integer:=196;
signal limSupPostesFc1  :integer:=130;
signal limLPosteFRc1    :integer:=524;

signal limLPosteBLc1    :integer:=201;
signal limRPosteBLc1    :integer:=207;
signal limSupPostesBc1  :integer:=110;
signal limLPosteBRc1    :integer:=513;
signal limRPosteBRc1    :integer:=519;

signal limSupFarosFc1   :integer:=100;
signal limSupFarosBc1   :integer:=85;

signal limIntMuroLcasaC1 :integer:=260;
signal limIntMuroRcasaC1 :integer:=460;

signal limSupAzoteaC1   :integer:=45;
signal limInfAzoteaC1   :integer:=60;
signal limInfTechoUC1   :integer:=65;
signal limInfFachadaP2C1:integer:=105;
signal limSupTechoDC1   :integer:=120;
signal limInfTechoDC1   :integer:=125;
signal limInfFachadaP1C1:integer:=165;
signal limInfCasaC1     :integer:=195;

signal limSupEscalon1C1 :integer:=180;
signal limSupEscalon2C1 :integer:=185;
signal limSupEscalon3C1 :integer:=190;
signal limLEscalon1C1  :integer:=321;
signal limREscalon1C1  :integer:=399;
signal limLEscalon2C1  :integer:=318;
signal limREscalon2C1  :integer:=402;
signal limLEscalon3C1  :integer:=315;
signal limREscalon3C1  :integer:=405;

signal limInfVentanasP2C1 :integer:=90;
signal limLenMedioVentanasP2C1 :integer:=341;
signal limRenMedioVentanasP2C1 :integer:=379;
signal limSupVentanasP1C1  :integer:=128;
signal limInfVentanasP1C1   :integer:=145;
signal limLintVentanasP1C1 :integer:=336;
signal limRintVentanasP1C1 :integer:=386;

signal limLPuertaCasaC1:integer:=346;
signal limRPuertaCasaC1:integer:=374;
signal limSupPuertaCasaC1:integer:=131;


------***limites de c2*******************************************
signal limSupBanquetaC2    :integer:=450;
signal limInfBanquetaC2     :integer:=465;
signal limLIntBanquetaC2   :integer:=185;
signal limRIntBanquetaC2   :integer:=255;

signal limSupMuroFrenteC2      :integer:=440;
signal limSupMuroAtrasC2       :integer:=260;
signal limInfMuroAtrasC2        :integer:=270;

signal limExtMuroIzqC2          :integer:=115;
signal limExtMuroDerC2          :integer:=325;
signal limIntMuroDerC2          :integer:=315;
signal limIntMuroIzqC2           :integer:=125;

signal limLPuertaFrenteC2      :integer:=168;
signal limRPuertaFrenteC2      :integer:=182;
signal limSupPuertaPortonC2    :integer:=440;
signal xPuertaFrenteC2          :std_logic_vector(8 downto 0);
signal xPortonC2                      :std_logic_vector(8 downto 0);

signal limLPuertaFrenteC2aux:std_logic_vector(8 downto 0);
signal limLPortonC2aux:std_logic_vector(8 downto 0);


signal limSupAlbercaC2 :integer:=300;
signal limInfAlbercaC2  :integer:=320;
signal limLAlbercaC2    :integer:=185;
signal limRAlbercaC2    :integer:=255;

signal limLOla1:integer:=195;
signal limROla1:integer:=205;
signal limLOla2:integer:=215;
signal limROla2:integer:=225;
signal limLOla3:integer:=235;
signal limROla3:integer:=245;
signal limOlasSup:integer:=307;
signal limOlasInf:integer:=314;

signal limSupMAlbercaC2:integer:=290;
signal limInfMAlbercaC2:integer:=330;
signal limLMAlbercaC2  :integer:=175;
signal limRMAlbercaC2  :integer:=265;

signal limSupCasaC2     :integer:=345;
signal limInfCasaC2     :integer:=420;
signal limLCasaC2       :integer:=155;
signal limRCasaC2       :integer:=285;


----***limites de c3*******************************************
signal limSupCasaC3     :integer:=325;
signal limInfCasaC3     :integer:=420;
signal limLCasaC3       :integer:=440;
signal limRCasaC3       :integer:=520;
signal limSupBanquetaC3 :integer := 415;

signal limSupVentanasP1C3 :integer:=370;
signal limInfVentanasP1C3  :integer:=395;
signal limSupVentanasP2C3 :integer:=335;
signal limInfVentanasP2C3  :integer:=360;
signal limLVentanasLC3       :integer:=450;
signal limRVentanasLC3       :integer:=475;
signal limLVentanasRC3       :integer:=485;
signal limRVentanasRC3       :integer:=510;

signal limSupEscalon1C3 :integer:=405;
signal limSupEscalon2C3 :integer:=410;
signal limSupEscalon3C3 :integer:=415;
signal limLEscalones1C3  :integer:=430;
signal limLEscalones2C3  :integer:=420;
signal limLEscalones3C3  :integer:=410;

signal limLMuroFrenteC3 :integer:=380;
signal limRMuroFrenteC3 :integer:=390;
signal limLMuroAtrasC3 :integer:=610;
signal limRMuroAtrasC3 :integer:=620;
signal limSupMurosC3:integer:=390;

signal limSupFarosC3    :integer:=345;
signal limInfFarosC3    :integer:=365;
signal limLFaroFRC3     :integer:=381;
signal limRFaroFRC3     :integer:=389;
signal limLFaroBRC3     :integer:=611;
signal limRFaroBRC3     :integer:=619;

signal limLPosteFRC3    :integer:=383;
signal limRPosteFRC3    :integer:=387;
signal limLPosteBRC3    :integer:=613;
signal limRPosteBRC3    :integer:=617;

--=====================================================================

begin

--RELOJES-=============================================================
clk25MHz: Clks port map(CLK,clk25); --reloj de 25MHz

sDelay  <= "01011111010111100000111111" when (iModo='0') else "00000000010011000100101101"; --modo normal/modo flash
msDelay <= "00101111101011110000011111" when (iModo='0') else "00000000001001100010010110"; --modo normal/modo flash


process(CLK)
	begin
		if CLK'event and CLK='1' then
			if cont1seg = sDelay then
				CLK1s <= not CLK1s;
				cont1seg <=(others =>'0');
			else
				cont1seg <= cont1seg+1;
			end if;
		end if;
end process;

process(CLK)
	begin
		if CLK'event and CLK='1' then
			if contMedioSeg = msDelay then
				CLKmedioS <= not CLKmedioS;
				contMedioSeg <=(others =>'0');
			else
				contMedioSeg <= contMedioSeg+1;
			end if;
		end if;
end process;

process(CLK)
```

```
    begin
        if CLK'event and CLK='1' then
            if contMov3 = "00000000001001100010010110" then
                clkMov3 <= not clkMov3;
                contMov3 <=(others =>'0');
            else
                contMov3 <= contMov3+1;
            end if;
        end if;
end process;

process(CLK1s)
    begin
        if CLK1s'event and CLK1s='1' then
            if cont1min = 29 then
                CLK1min <= not CLK1min;
                cont1min <=(others =>'0');
            else
                cont1min <= cont1min+1;
            end if;
        end if;
end process;

process(CLK)
    begin
        if CLK'event and CLK='1' then
            if contCuadro = 2 then
                CLKcuadro <= not CLKcuadro;
                contCuadro <=(others =>'0');
            else
                contCuadro <= contCuadro+1;
            end if;
        end if;
end process;

process(CLK)
begin
    if(CLK = '1' and CLK'event) then
        clk_lento <= clk_lento + 1;
    end if;
end process;
--//relojes más lentos para el movimiento de las puertas y portones
clkMov <= clk_lento(20);
clkMov2<= clk_lento(19);
--=====================================================================================
--SERIAL-=========================================================================

baud_Timer: process(CLK) --reloj para determinar los bits por segundo. En este caso se determinó utilizar 4800bps por lo que el contador
                         -- necesita llegar hasta 650. Para meyor información referirse al manual de usuario del PICOBLAZE
begin
    if CLK'event and CLK='1' then
        if baud_count=650 then
            baud_count<=0;
            en_16_x_baud <='1';
        else
            baud_count<=baud_count + 1;
            en_16_x_baud <='0';
        end if;
    end if;
end process baud_timer;


transmit: uart_tx port map(datoTx,wbufferTx,resetbufferTx,en_16_x_baud,ioTx,bufferFTx,bufferHFTx,CLK);
receive : uart_rx port map(ioRx,datoRx,rbufferRx,resetbufferRx,en_16_x_baud,bufferReadyRx,bufferFRx,bufferHFRx,CLK);



--//máquina de Edos para la recepción de datos
RECIBE:process(stateRx)
begin
    case stateRx is
        when 0 => if bufferReadyRx='1' then
                        nstateRx<=1;
                        rbufferRx<='1';
                        resetbufferRx<='0';
                        dato55Rx<="00000000";
                        setTimeRx<='0';
                        datoAARx<="00000000";
                        setNewValueRx<='0';
                    else
                        nstateRx<=0;
                        resetbufferRx<='0';
                        rbufferRx<='0';
                        dato55Rx<="00000000";
                        datoAARx<="00000000";
                        setNewValueRx<='0';
                        setTimeRx<='0';
                    end if;

        when 1 => rbufferRx<='0';                                   --recibe 1er octeto
```

```
                       dato55Rx<= datoRx;
                       if datoRx = "01010101" then
                             ledprueba<='1';
                       else
                             ledprueba<='0';
                       end if;
                       nstateRx<=2;


when 2 => if bufferReadyRx='1' then
                             nstateRx<=3;
                             rbufferRx<='1';
                      else
                             nstateRx<=2;
                      end if;
when 3 => rbufferRx<='0';                                 --recibe 2do octeto
                       datoIdRx<= datoRx;
                       if datoRx = "00000001" then
                             ledprueba2<='1';
                       else
                             ledprueba2<='0';
                       end if;
                       nstateRx<=4;


when 4 => if bufferReadyRx='1' then
                             nstateRx<=5;
                             rbufferRx<='1';
                      else
                             nstateRx<=4;
                      end if;
when 5 => rbufferRx<='0';                                 --recibe 3er octeto
                       datoC1Rx<= datoRx;
                       nstateRx<=6;


when 6 => if bufferReadyRx='1' then
                             nstateRx<=7;
                             rbufferRx<='1';
                      else
                             nstateRx<=6;
                      end if;

when 7 => rbufferRx<='0';                                 --recibe 4to octeto
                       datoC2Rx<= datoRx;
                       nstateRx<=8;
```

```
            when 8 => if bufferReadyRx='1' then
                                     nstateRx<=9;
                                     rbufferRx<='1';
                               else
                                     nstateRx<=8;
                               end if;
            when 9 => rbufferRx<='0';                               --recibe 5to octeto
                                datoC3Rx<= datoRx;
                                nstateRx<=10;

            when 10 => if bufferReadyRx='1' then
                                     nstateRx<=11;
                                     rbufferRx<='1';
                               else
                                     nstateRx<=10;
                               end if;
            when 11 => rbufferRx<='0';                              --recibe 6to octeto
                                 datoC4Rx<= datoRx;
                                 nstateRx<=12;

            when 12 => if bufferReadyRx='1' then
                                     nstateRx<=13;
                                     rbufferRx<='1';
                               else
                                     nstateRx<=12;
                               end if;
            when 13 => rbufferRx<='0';                              --recibe 7mo octeto
                           datoAARx<= datoRx;
                                if datoRx = "10101010" then
                                     ledprueba3<='1';
                                     setNewValueRx<='1';
                                else
                                     ledprueba3<='0';
                                end if;
                                if (datoC1Rx="0000110")then
                                           setTimeRx<='1';
                                else
                                           setTimeRx<='0';
                                end if;
                                nstateRx<=0;
            when others => null;
      end case;
end process;
```

```
--//máquina de Edos para la transmisión de datos
TRANSMITE:process(sendAlgo)
begin
       case stateTx is
              when 0=> if sendAlgo='1' then
                             nstateTx<=15;
                       else  nstateTx<=0;
                     end if;
              when 15=> if sendAlgo='0' then
                             nstateTx<=1;
                         else nstateTx<=15;
                         end if;

              when 1=> datoTx<= dato55Tx;                        --trasmite 1er octeto
                       wbufferTx<='1';
                       nstateTx<=2;
              when 2=> wbufferTx<='0';nstateTx<=3;
              when 3=> datoTx<= datoIdTx;                        --trasmite 2do octeto
                        wbufferTx<='1';
                        nstateTx<=4;
              when 4=> wbufferTx<='0';nstateTx<=5;
              when 5=> datoTx<= datoC1Tx;                        --trasmite 3er octeto
                        wbufferTx<='1';
                        nstateTx<=6;
              when 6=> wbufferTx<='0';nstateTx<=7;
              when 7=> datoTx<= datoC2Tx;                        --trasmite 4to octeto
                        wbufferTx<='1';
                        nstateTx<=8;
              when 8=> wbufferTx<='0';nstateTx<=9;
              when 9=> datoTx<= datoC3Tx;                        --trasmite 5to octeto
                        wbufferTx<='1';
                        nstateTx<=10;
              when 10=> wbufferTx<='0';nstateTx<=11;
              when 11=> datoTx<= datoC4Tx;                       --trasmite 6to octeto
                         wbufferTx<='1';
                         nstateTx<=12;
              when 12=> wbufferTx<='0';nstateTx<=13;
              when 13=> datoTx<= datoAATx;                       --trasmite 7mo octeto
                         wbufferTx<='1';
                        nstateTx<=14;
              when 14=> wbufferTx<='0';nstateTx<=0;
              when others => null;
       end case;
end process;

NextState: process(CLK, nstateTx)
```

```
begin
          if(CLK = '1' and CLK'event)then
                              stateTx <= nstateTx;
                              stateRx <= nstateRx;
          end if;
end process;


--===================================================================================


--DISPLAY del SPARTAN=================================================================

disp: Display port map(clk25,disp1,disp2,disp3,disp4,salBCD,ENE);

disp4 <= "1011" when (iLuces='1' and iFaros='0' and teclaKB<10) else
        "1111" when (iLuces='0' and iFaros='1' and teclaKB<10) else
                       "1010"; --nada

disp3<= teclaKB when (teclaKB<10)else "1010";

disp2<= "1010"; --nada

disp1<=           teclaKB when (teclaKB>10) else "1010";


--Lee del TECLADO=====================================================================


teclado: keyboard port map(kb_clk,kb_data,clk25,resetKB,readKB,kbcode,kbready);

--proceso para establecer el read y reset del teclado
process (clk25,kbready)
begin
  if clk25'EVENT and clk25 = '1' then
                    if (kbready = '1')then --si ya se leyó algo es necesario darle reset y prepararse para una nueva lectura
                              resetKB <= '1';
                              readKB  <= '1';
                    else
                              resetKB <= '0';
                              readKB  <= '0';
                    end if;
  end if;
end process;
```

```
process(clk25, kbready,kbcode)
begin
    if clk25'EVENT and clk25 = '1' then
        if (kbready = '1')then --si hay un dato listo del teclado
                case kbcode is
                    when "00010110" => teclaKB<="0001"; --0x16 1
                    when "00011110" => teclaKB<="0010"; --0x1E 2
                    when "00100110" => teclaKB<="0011"; --0x26 3
                    when "00100101" => teclaKB<="0100"; --0x25 4
                    when "00101110" => teclaKB<="0101"; --0x2E 5
                    when "00110110" => teclaKB<="0110"; --0x36 6
                    when "00111101" => teclaKB<="0111"; --0x3D 7
                    when "00111110" => teclaKB<="1000"; --0x3E 8
                    when "01000110" => teclaKB<="1001"; --0x46 9
                    when "00101001" => teclaKB<="1010"; --0x29 BORRAR

                    when "01001011" => teclaKB<="1011"; --0x4B     L alberca
                    when "00101101" => teclaKB<="1100"; --0x2D     R robo
                    when "00011100" => teclaKB<="1101"; --0x1C     A alarma
                    when "00110100" => teclaKB<="1110"; --0x34     G porton
                    when "00101011" => teclaKB<="1111"; --0x2B     F puertaFrente

                    when others      => teclaKB<=teclaKB;
                end case;
        end if;
 end if;
end process;

process(clk25, kbready,kbcode)
begin
    if clk25'EVENT and clk25 = '1' then
        if (kbready = '1')then
                case kbcode is
                    when "11110000" => if enterPUSHED = '1' then --0xF0 soltar tecla
                                                enterOFF <='1';
                                              end if;
                    when "01011010" => --ENTER
                                            if enterOFF = '1' then
                                                enterPUSHED <='0';
                                                enterOFF <= '0';
                                                sendAlgo<='0';
                                            else
                                                enterPUSHED <='1';
                                                --// enciende la bandera SENDalgo solamente si en realidad se va a mandar algo válido
                                                if ((iLuces='1' and iFaros='0') and teclaKB<10) then
                                                    sendAlgo<='1';
```

```
                                elsif ((iLuces='0' and iFaros='1')and (teclaKB<5 or (teclaKB>=7 and teclaKB<10))) then
                                    sendAlgo<='1';
                                elsif (teclaKB>10 and teclaKB<=15) then
                                    sendAlgo<='1';
                                else
                                    sendAlgo<='0';
                                end if;
                            end if;
                    when others      => null;
                end case;
        end if;
 end if;
end process;


--proceso donde se establecen los valores de los edos de las cosas y las intensidades de faros y luces ya sea
--por recibirlos serialmente o a través del teclado.

process(setNewValueRx,enterPUSHED,teclaKB, iLuces, iFaros, dato55Rx,datoC1Rx,datoC2Rx,datoC3Rx,datoAARx)
begin
    if (setNewValueRx='1') then --se recibió algo por serial
            if (datoC1Rx<="00000000") then          --sistema
                    if    (datoC2Rx<="00000000") then
                            if    (datoC3Rx<="00000000") then
                                    edoSistema<='0';
                            elsif (datoC3Rx<="00000001") then
                                    edoSistema<='1';
                            end if;
                    end if;

            elsif    (datoC1Rx<="00000001") then
                    if    (datoC2Rx<="00000001") then --cuarto arriba frente izquierda
                            if    (datoC3Rx<="00000000") then
                                    intventanaULF <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaULF <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaULF <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaULF <= "11";
                            end if;
                    elsif (datoC2Rx<="00000010") then --cuarto arriba frente derecha
                            if    (datoC3Rx<="00000000") then
                                    intventanaURF <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaURF <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaURF <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaURF <= "11";
                            end if;
                    elsif (datoC2Rx<="00000100") then --cuarto arriba atras derecha
                            if    (datoC3Rx<="00000000") then
                                    intventanaURB <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaURB <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaURB <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaURB <= "11";
                            end if;
                    elsif (datoC2Rx<="00000101") then --cuarto abajo frente izquierda
                            if    (datoC3Rx<="00000000") then
                                    intventanaDLF <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaDLF <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaDLF <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaDLF <= "11";
                            end if;
                    elsif (datoC2Rx<="00000110") then --cuarto abajo frente derecha
                            if    (datoC3Rx<="00000000") then
                                    intventanaDRF <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaDRF <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaDRF <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaDRF <= "11";
                            end if;
                    elsif (datoC2Rx<="00001000") then --cuarto abajo atras derecha
                            if    (datoC3Rx<="00000000") then
                                    intventanaDRB <= "00";
                            elsif (datoC3Rx<="00000001") then
                                    intventanaDRB <= "01";
                            elsif (datoC3Rx<="00000010") then
                                    intventanaDRB <= "10";
                            elsif (datoC3Rx<="00000011") then
                                    intventanaDRB <= "11";
                            end if;
                    elsif (datoC2Rx<="00001001") then --piso arriba
```

```
                if    (datoC3Rx<="00000000") then
                                intventanaULF <= "00";
                                intventanaURF <= "00";
                                intventanaURB <= "00";
                elsif (datoC3Rx<="00000001") then
                                intventanaULF <= "01";
                                intventanaURF <= "01";
                                intventanaURB <= "01";
                elsif (datoC3Rx<="00000010") then
                                intventanaULF <= "10";
                                intventanaURF <= "10";
                                intventanaURB <= "10";
                elsif (datoC3Rx<="00000011") then
                                intventanaULF <= "11";
                                intventanaURF <= "11";
                                intventanaURB <= "11";
                end if;
elsif (datoC2Rx<="00001010") then --piso abajo
                if    (datoC3Rx<="00000000") then
                                intventanaDLF <= "00";
                                intventanaDRF <= "00";
                                intventanaDRB <= "00";
                elsif (datoC3Rx<="00000001") then
                                intventanaDLF <= "01";
                                intventanaDRF <= "01";
                                intventanaDRB <= "01";
                elsif (datoC3Rx<="00000010") then
                                intventanaDLF <= "10";
                                intventanaDRF <= "10";
                                intventanaDRB <= "10";
                elsif (datoC3Rx<="00000011") then
                                intventanaDLF <= "11";
                                intventanaDRF <= "11";
                                intventanaDRB <= "11";
                end if;
elsif (datoC2Rx<="00001011") then --todas
                if    (datoC3Rx<="00000000") then
                                intventanaULF <= "00";
                                intventanaURF <= "00";
                                intventanaURB <= "00";
                                intventanaDLF <= "00";
                                intventanaDRF <= "00";
                                intventanaDRB <= "00";
                elsif (datoC3Rx<="00000001") then
                                intventanaULF <= "01";
                                intventanaURF <= "01";
```

```
            intventanaURB <= "01";
            intventanaDLF <= "01";
            intventanaDRF <= "01";
            intventanaDRB <= "01";
        elsif (datoC3Rx<="00000010") then
            intventanaULF <= "10";
            intventanaURF <= "10";
            intventanaURB <= "10";
            intventanaDLF <= "10";
            intventanaDRF <= "10";
            intventanaDRB <= "10";
        elsif (datoC3Rx<="00000011") then
            intventanaULF <= "11";
            intventanaURF <= "11";
            intventanaURB <= "11";
            intventanaDLF <= "11";
            intventanaDRF <= "11";
            intventanaDRB <= "11";
        end if;
    end if;
elsif (datoC1Rx<="00000010") then
    if    (datoC2Rx<="00000001") then --poste frente izquierda
        if    (datoC3Rx<="00000000") then
            intFaroFL <= "00";
        elsif (datoC3Rx<="00000001") then
            intFaroFL <= "01";
        elsif (datoC3Rx<="00000010") then
            intFaroFL <= "10";
        elsif (datoC3Rx<="00000011") then
            intFaroFL <= "11";
        end if;
    elsif (datoC2Rx<="00000010") then --poste frente derecha
        if    (datoC3Rx<="00000000") then
            intFaroFR <= "00";
        elsif (datoC3Rx<="00000001") then
            intFaroFR <= "01";
        elsif (datoC3Rx<="00000010") then
            intFaroFR <= "10";
        elsif (datoC3Rx<="00000011") then
            intFaroFR <= "11";
        end if;
    elsif (datoC2Rx<="00000011") then --poste atras izquierda
        if    (datoC3Rx<="00000000") then
            intFaroBL <= "00";
        elsif (datoC3Rx<="00000001") then
            intFaroBL <= "01";
        elsif (datoC3Rx<="00000010") then
                intFaroBL <= "10";
        elsif (datoC3Rx<="00000011") then
                intFaroBL <= "11";
        end if;
elsif (datoC2Rx<="00000100") then --poste atras derecha
        if    (datoC3Rx<="00000000") then
                intFaroBR <= "00";
        elsif (datoC3Rx<="00000001") then
                intFaroBR <= "01";
        elsif (datoC3Rx<="00000010") then
                intFaroBR <= "10";
        elsif (datoC3Rx<="00000011") then
                intFaroBR <= "11";
        end if;
elsif (datoC2Rx<="00001010") then --postes frente
        if    (datoC3Rx<="00000000") then
                intFaroFL <= "00";
                intFaroFR <= "00";
        elsif (datoC3Rx<="00000001") then
                intFaroFL <= "01";
                intFaroFR <= "01";
        elsif (datoC3Rx<="00000010") then
                intFaroFL <= "10";
                intFaroFR <= "10";
        elsif (datoC3Rx<="00000011") then
                intFaroFL <= "11";
                intFaroFR <= "11";
        end if;
elsif (datoC2Rx<="00001001") then --postes atras
        if    (datoC3Rx<="00000000") then
                intFaroBL <= "00";
                intFaroBR <= "00";
        elsif (datoC3Rx<="00000001") then
                intFaroBL <= "01";
                intFaroBR <= "01";
        elsif (datoC3Rx<="00000010") then
                intFaroBL <= "10";
                intFaroBR <= "10";
        elsif (datoC3Rx<="00000011") then
                intFaroBL <= "11";
                intFaroBR <= "11";
        end if;
elsif (datoC2Rx<="00001011") then --todos postes
        if    (datoC3Rx<="00000000") then
                intFaroFL <= "00";
```

```
                        intFaroFR <= "00";
                        intFaroBL <= "00";
                        intFaroBR <= "00";
                elsif (datoC3Rx<="00000001") then
                        intFaroFL <= "01";
                        intFaroFR <= "01";
                        intFaroBL <= "01";
                        intFaroBR <= "01";
                elsif (datoC3Rx<="00000010") then
                        intFaroFL <= "10";
                        intFaroFR <= "10";
                        intFaroBL <= "10";
                        intFaroBR <= "10";
                elsif (datoC3Rx<="00000011") then
                        intFaroFL <= "11";
                        intFaroFR <= "11";
                        intFaroBL <= "11";
                        intFaroBR <= "11";
                end if;
        end if;
elsif (datoC1Rx<="00000011") then
        if    (datoC2Rx<="00000001") then --candado
                if    (datoC3Rx<="00000000") then
                        edoAlarma<='0';
                elsif (datoC3Rx<="00000001") then
                        edoAlarma<='1';
                elsif (datoC3Rx<="00000010") then
                        edoRobo <= '1';
                elsif (datoC3Rx<="00000011") then
                        edoRobo <= '0';
                end if;
        end if;
elsif (datoC1Rx<="00000100") then
        if    (datoC2Rx<="00000001") then --alberca
                if    (datoC3Rx<="00000000") then
                        edoAlberca<='0';
                elsif (datoC3Rx<="00000001") then
                        edoAlberca<='1';
                end if;
        end if;
elsif (datoC1Rx<="00000101") then
        if    (datoC2Rx<="00000001") then --porton
                if    (datoC3Rx<="00000000") then
                        edoPorton<='0';
                elsif (datoC3Rx<="00000001") then
                        edoPorton<='1';
                end if;
        elsif (datoC2Rx<="00000010") then --puerta
                if    (datoC3Rx<="00000000") then
                        edoPuerta<='0';
                elsif (datoC3Rx<="00000001") then
                        edoPuerta<='1';
                end if;
        end if;

elsif (datoC1Rx<="00000110") then
        if (datoC2Rx<"00001010")then --0a9
                horaHRx<= "0000";
                horaLRx<= datoC2Rx(3 downto 0);
        elsif (datoC2Rx="00001010")then --10
                horaHRx<= "0001";
                horaLRx<= "0000";
        elsif (datoC2Rx="00001011")then--11
                horaHRx<= "0001";
                horaLRx<= "0001";
        elsif (datoC2Rx="00001100")then--12
                horaHRx<= "0001";
                horaLRx<= "0010";
        elsif (datoC2Rx="00001101")then--13
                horaHRx<= "0001";
                horaLRx<= "0011";
        elsif (datoC2Rx="00001110")then--14
                horaHRx<= "0001";
                horaLRx<= "0100";
        elsif (datoC2Rx="00001111")then--15
                horaHRx<= "0001";
                horaLRx<= "0101";
        elsif (datoC2Rx="00010000")then--16
                horaHRx<= "0001";
                horaLRx<= "0110";
        elsif (datoC2Rx="00010001")then--17
                horaHRx<= "0001";
                horaLRx<= "0111";
        elsif (datoC2Rx="00010010")then--18
                horaHRx<= "0001";
                horaLRx<= "1000";
        elsif (datoC2Rx="00010011")then--19
                horaHRx<= "0001";
                horaLRx<= "1001";
        elsif (datoC2Rx="00010100")then--20
                horaHRx<= "0010";
                horaLRx<= "0000";
                elsif (datoC2Rx="00010101")then--21
                        horaHRx<= "0010";
                        horaLRx<= "0001";
                elsif (datoC2Rx="00010110")then--22
                        horaHRx<= "0010";
                        horaLRx<= "0010";
                elsif (datoC2Rx="00010111")then--23
                        horaHRx<= "0010";
                        horaLRx<= "0011";

                end if;
                minHRx <= datoC3Rx(3 downto 0);
                minLRx <= datoC4Rx(3 downto 0);
        end if;

elsif enterPUSHED='1' and enterPUSHED'event then
            case teclaKB is
                        when "0001"=> --1
                                    if(iLuces='1' and iFaros='0')then
                                                intventanaULF<= intventanaULF +'1';
                                                banderaTodasLucesIgual<='0';
                                                banderaTodasLucesP2Igual<='0';

                                                ---=================
                                                datoC1Tx<="0000"&"0001";
                                                datoC2Tx<="0000"&"0001";
                                                if intventanaULF ="11" then
                                                    datoC3Tx<="00000000";
                                                else
                                                    datoC3Tx<="000000"&intventanaULF+1;
                                                end if;
                                                ---=================


                                    elsif (iLuces='0' and iFaros='1') then
                                                intFaroFL<= intFaroFL+'1';
                                                banderaTodosFarosIgual<='0';
                                                banderaTodosFarosFIgual<='0';

                                                ---=================
                                                datoC1Tx<="0000"&"0010";
                                                datoC2Tx<="0000"&"0001";
                                                if intFaroFL ="11" then
                                                    datoC3Tx<="00000000";
                                                else
                                                    datoC3Tx<="000000"&intFaroFL+1;
                                                end if;

                                                ---=================
                                    end if;
                        when "0010"=> --2
                                    if(iLuces='1' and iFaros='0')then
                                                intventanaURF<= intventanaURF +'1';
                                                banderaTodasLucesIgual<='0';
                                                banderaTodasLucesP2Igual<='0';
                                                ---=================
                                                datoC1Tx<="0000"&"0001";
                                                datoC2Tx<="0000"&"0010";
                                                if intventanaURF ="11" then
                                                    datoC3Tx<="00000000";
                                                else
                                                    datoC3Tx<="000000"&intventanaURF+1;
                                                end if;
                                                ---=================

                                    elsif (iLuces='0' and iFaros='1') then
                                                intFaroFR<= intFaroFR+'1';
                                                banderaTodosFarosIgual<='0';
                                                banderaTodosFarosFIgual<='0';

                                                ---=================
                                                datoC1Tx<="0000"&"0010";
                                                datoC2Tx<="0000"&"0010";
                                                if intFaroFR ="11" then
                                                    datoC3Tx<="00000000";
                                                else
                                                    datoC3Tx<="000000"&intFaroFR+1;
                                                end if;

                                                ---=================

                                    end if;
                        when "0011"=> --3
                                    if(iLuces='1' and iFaros='0')then
                                                intventanaDLF<= intventanaDLF +'1';
                                                banderaTodasLucesIgual<='0';
                                                banderaTodasLucesP1Igual<='0';

                                                ---=================
                                                datoC1Tx<="0000"&"0001";
                                                datoC2Tx<="0000"&"0101";
                                                if intventanaDLF ="11" then
```

```
                    datoC3Tx<="00000000";
                else
                    datoC3Tx<="000000"&intventanaDLF+1;
                end if;
                ---=================


        elsif (iLuces='0' and iFaros='1') then
                intFaroBL<= intFaroBL+'1';
                banderaTodosFarosIgual<='0';
                banderaTodosFarosBIgual<='0';

                ---=================
                datoC1Tx<="0000"&"0010";
                datoC2Tx<="0000"&"0011";
                if intFaroBL ="11" then
                    datoC3Tx<="00000000";
                else
                    datoC3Tx<="000000"&intFaroBL+1;
                end if;
                ---=================

        end if;
when "0100"=> --4
        if(iLuces='1' and iFaros='0')then
                intventanaDRF<= intventanaDRF +'1';
                banderaTodasLucesIgual<='0';
                banderaTodasLucesP1Igual<='0';

                ---=================
                datoC1Tx<="0000"&"0001";
                datoC2Tx<="0000"&"0110";
                if intventanaDRF ="11" then
                    datoC3Tx<="00000000";
                else
                    datoC3Tx<="000000"&intventanaDRF+1;
                end if;
                ---=================

        elsif (iLuces='0' and iFaros='1') then
                intFaroBR<= intFaroBR+'1';
                banderaTodosFarosIgual<='0';
                banderaTodosFarosBIgual<='0';

                ---=================
                datoC1Tx<="0000"&"0010";
```

```
                        datoC2Tx<="0000"&"0100";
                        if intFaroBR ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intFaroBR+1;
                        end if;
                        ---=================

            end if;
when "0101"=> --5
            if(iLuces='1' and iFaros='0')then
                        intventanaURB<= intventanaURB +'1';
                        banderaTodasLucesIgual<='0';
                        banderaTodasLucesP2Igual<='0';

                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"0100";
                        if intventanaURB ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intventanaURB+1;
                        end if;
                        ---=================

            end if;
when "0110"=> --6
            if(iLuces='1' and iFaros='0')then
                        intventanaDRB<= intventanaDRB +'1';
                        banderaTodasLucesP1Igual<='0';
                        banderaTodasLucesIgual<='0';

                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"1000";
                        if intventanaDRB ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intventanaDRB+1;
                        end if;
                        ---=================

            end if;
when "0111"=> --7 --todas primer piso o frente en intensidad1 primero y luego aumenta
            if(iLuces='1' and iFaros='0')then
                    if (banderaTodasLucesP1Igual='0') then
            intventanaDLF<="01";
            intventanaDRF<="01";
            intventanaDRB<="01";
            banderaTodasLucesP1Igual<='1';
            banderaTodasLucesIgual<='0';

            ---=================
            datoC1Tx<="0000"&"0001";
            datoC2Tx<="0000"&"1010";
            datoC3Tx<="00000001";
            ---=================

        else
            intventanaDLF<= intventanaDLF+'1';
            intventanaDRF<= intventanaDRF+'1';
            intventanaDRB<= intventanaDRB+'1';
            banderaTodasLucesIgual<='0';

            ---=================
            datoC1Tx<="0000"&"0001";
            datoC2Tx<="0000"&"1010";
            if intventanaDRB ="11" then
                datoC3Tx<="00000000";
            else
                datoC3Tx<="000000"&intventanaDRB+1;
            end if;
            ---=================

        end if;
elsif (iLuces='0' and iFaros='1') then
        if (banderaTodosFarosFIgual='0') then
            intFaroFL<= "01";
            intFaroFR<= "01";
            banderaTodosFarosFIgual<='1';
            banderaTodosFarosIgual<='0';

            ---=================
            datoC1Tx<="0000"&"0010";
            datoC2Tx<="0000"&"1010";
            datoC3Tx<="00000001";
            ---=================
        else
            intFaroFL<= intFaroFL+'1';
            intFaroFR<= intFaroFR+'1';
            banderaTodosFarosIgual<='0';
```

```
                        ---=================
                        datoC1Tx<="0000"&"0010";
                        datoC2Tx<="0000"&"1010";
                        if intFaroFL ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intFaroFL+1;
                        end if;
                        ---=================


                    end if;
            end if;
when "1000"=> --8 --todas segundo piso o atras en intensidad1 primero y luego aumenta
            if(iLuces='1' and iFaros='0')then
                    if (banderaTodasLucesP2Igual='0') then
                        intventanaULF<="01";
                        intventanaURF<="01";
                        intventanaURB<="01";
                        banderaTodasLucesP2Igual<='1';
                        banderaTodasLucesIgual<='0';

                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"1001";
                        datoC3Tx<="00000001";
                        ---=================

                    else
                        intventanaULF<= intventanaULF+'1';
                        intventanaURF<= intventanaURF+'1';
                        intventanaURB<= intventanaURB+'1';
                        banderaTodasLucesIgual<='0';

                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"1001";
                        if intventanaURB ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intventanaURB+1;
                        end if;
                        ---=================

                    end if;
            elsif (iLuces='0' and iFaros='1') then
```

```
                    if (banderaTodosFarosBIgual='0') then
                        intFaroBL<= "01";
                        intFaroBR<= "01";
                        banderaTodosFarosBIgual<='1';
                        banderaTodosFarosIgual<='0';

                        ---=================
                        datoC1Tx<="0000"&"0010";
                        datoC2Tx<="0000"&"1001";
                        datoC3Tx<="00000001";
                        ---=================

                    else
                        intFaroBL<= intFaroBL+'1';
                        intFaroBR<= intFaroBR+'1';
                        banderaTodosFarosIgual<='0';
                        ---=================
                        datoC1Tx<="0000"&"0010";
                        datoC2Tx<="0000"&"1001";
                        if intFaroBL ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intFaroBL+1;
                        end if;
                        ---=================

                    end if;
            end if;
when "1001"=> --9 --todas en intensidad1 primero y luego aumenta
            if(iLuces='1' and iFaros='0')then
                    if (banderaTodasLucesIgual='0') then
                        intventanaULF<="01";
                        intventanaURF<="01";
                        intventanaDLF<="01";
                        intventanaDRF<="01";
                        intventanaURB<="01";
                        intventanaDRB<="01";
                        banderaTodasLucesIgual<='1';
                        banderaTodasLucesP1Igual<='1';
                        banderaTodasLucesP2Igual<='1';
                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"1011";
                        datoC3Tx<="00000001";
                        ---=================
                    else
                        intventanaULF<= intventanaULF+'1';
                        intventanaURF<= intventanaURF+'1';
                        intventanaDLF<= intventanaDLF+'1';
                        intventanaDRF<= intventanaDRF+'1';
                        intventanaURB<= intventanaURB+'1';
                        intventanaDRB<= intventanaDRB+'1';

                        ---=================
                        datoC1Tx<="0000"&"0001";
                        datoC2Tx<="0000"&"1011";
                        if intventanaURB ="11" then
                            datoC3Tx<="00000000";
                        else
                            datoC3Tx<="000000"&intventanaURB+1;
                        end if;
                        ---=================

                    end if;
            elsif (iLuces='0' and iFaros='1') then
                    if (banderaTodosFarosIgual='0') then
                        intFaroFL<= "01";
                        intFaroFR<= "01";
                        intFaroBL<= "01";
                        intFaroBR<= "01";
                        banderaTodosFarosIgual<='1';
                        banderaTodosFarosFIgual<='1';
                        banderaTodosFarosBIgual<='1';
                        ---=================
                        datoC1Tx<="0000"&"0010";
                        datoC2Tx<="0000"&"1011";
                        datoC3Tx<="00000001";
                        ---=================

                    else
                        intFaroFL<= intFaroFL+'1';
                        intFaroFR<= intFaroFR+'1';
                        intFaroBL<= intFaroBL+'1';
                        intFaroBR<= intFaroBR+'1';

                        ---=================
                        datoC1Tx<="0000"&"0010";
                        datoC2Tx<="0000"&"1011";
                        if intFaroBL ="11" then
                            datoC3Tx<="00000000";
                        else
```

```
                            datoC3Tx<="000000"&intFaroBL+1;
                        end if;
                        ---=================

                    end if;
             end if;


when "1011"=> --11 Alberca
                    edoAlberca<= not edoAlberca;

                    ---=================
                    datoC1Tx<="0000"&"0100";
                    datoC2Tx<="0000"&"0001";
                    if edoAlberca ='0' then
                        datoC3Tx<="00000001";
                    else
                        datoC3Tx<="00000000";
                    end if;
                    ---=================

when "1100"=> --12 Robo
                    edoRobo<= not edoRobo;

                    ---=================
                    datoC1Tx<="0000"&"0011";
                    datoC2Tx<="0000"&"0001";
                    if edoRobo ='0' then
                        datoC3Tx<="00000010";
                    else
                        datoC3Tx<="00000011";
                    end if;
                    ---=================



when "1101"=> --13 Alarma
                    edoAlarma<= not edoAlarma;

                    ---=================
                    datoC1Tx<="0000"&"0011";
                    datoC2Tx<="0000"&"0001";
                    if edoAlarma ='0' then
                        datoC3Tx<="00000001";
                    else
                        datoC3Tx<="00000000";
```

```
                                    end if;
                                    ---=================


                        when "1110"=> --14 Porton
                                    edoPorton<= not edoPorton;

                                    ---=================
                                    datoC1Tx<="0000"&"0101";
                                    datoC2Tx<="0000"&"0001";
                                    if edoPorton ='0' then
                                            datoC3Tx<="00000001";
                                    else
                                            datoC3Tx<="00000000";
                                    end if;
                                    ---=================


                        when "1111"=> --15 PuertaFrente
                                    edoPuerta<= not edoPuerta;

                                    ---=================
                                    datoC1Tx<="0000"&"0101";
                                    datoC2Tx<="0000"&"0010";
                                    if edoPuerta ='0' then
                                            datoC3Tx<="00000001";
                                    else
                                            datoC3Tx<="00000000";
                                    end if;
                                    ---=================



                        when others     => null;
                end case;
 end if;
end process;


--=================================================================================================================================
--GENERA SEÑAL de VGA-*==========================================================================

contHcont: contador port map(clk25,resetH,'1',Hcont);
contVcont: contador port map(clk25,resetV,resetH,Vcont);

resetH <= '1' when Hcont = 799 else '0'; --cuando = 799
resetV <= '1' when Vcont = 519 else '0'; --cuando = 519

HSaux <= '1' when ((Hcont < 656) or (Hcont > 752)) else '0'; --<656 o >752
VSaux <= '1' when ((Vcont < 490) or (Vcont > 492)) else '0'; --<490 o >492

HS <= HSaux;
VS <= VSaux;

--==============================================================================================




--Checa BANDERAS de UBICACIóN-=====================================================================

--**generales*****************************************************************************
onScreen  <= '1' when (HSaux = '1' and VSaux='1' ) else '0';
onLeftBar <= '1' when (Hcont < limCapsD and onScreen = '1') else '0';
onCap1    <= '1' when (Hcont > limCapsD and Vcont < limCapsMh and onScreen = '1') else '0';
onCap2    <= '1' when (Hcont > limCapsD and Vcont > limCapsMh and Hcont < limCapsMv and onScreen = '1') else '0';
onCap3    <= '1' when (Hcont > limCapsD and Vcont > limCapsMh and Hcont > limCapsMv and onScreen = '1') else '0';
onAnyCap  <= onCap1 or onCap2 or onCap3;

onLeftBarFrame          <= '1' when (Hcont = limCapsD) else '0';
onMiddleCapsHFrame <= '1' when (Hcont > limCapsD  and Vcont = limCapsMh and onScreen = '1') else '0';
onMiddleCapsVFrame <= '1' when (Vcont > limCapsMh and Hcont = limCapsMv and onScreen = '1') else '0';
onCapsFrame            <= onLeftBarFrame or onMiddleCapsHFrame or onMiddleCapsVFrame;
----------------------------------------------------------------------------------------


------**left Bar**************************************************************************

interruptorLB<= '1' when(Hcont>limLinterruptorLB and Hcont<limRinterruptorLB and Vcont>limSUPinterruptorLB and Vcont<limINFinterruptorLB) else'0';
onFrameInterruptor <= '1' when (Hcont>limLinterruptorLB and Hcont<limRinterruptorLB and (Vcont=limSUPinterruptorLB or Vcont=limINFinterruptorLB)) else
                                '1' when ((Hcont=limLinterruptorLB or Hcont=limRinterruptorLB) and Vcont>limSUPinterruptorLB and Vcont<limINFinterruptorLB) else '0';

onLetreroON<= onON_o or onON_n ;
```

```
onLetreroOFF<= onOFF_o or onOFF_f1 or onOFF_f2 ;

onON_o   <= '1' when (Hcont>=limLON_o and Hcont<limRON_o and Vcont>=limSupLetreroONOFF and Vcont<=limInfLetreroONOFF) else '0';
onON_n   <= '1' when (Hcont>=limLON_n and Hcont< limRON_n and Vcont>=limSupLetreroONOFF and Vcont<=limInfLetreroONOFF) else '0';
onOFF_o  <= '1' when (Hcont>=limLOFF_o and Hcont<limROFF_o and Vcont>=limSupLetreroONOFF and Vcont<=limInfLetreroONOFF) else '0';
onOFF_f1 <= '1' when (Hcont>=limLOFF_f1 and Hcont<limROFF_f1 and Vcont>=limSupLetreroONOFF and Vcont<=limInfLetreroONOFF) else '0';
onOFF_f2 <= '1' when (Hcont>=limLOFF_f2 and Hcont<limROFF_f2 and Vcont>=limSupLetreroONOFF and Vcont<=limInfLetreroONOFF) else '0';


onHoraLB<= onHrH or onHrL or onMinH or onMinL or on2p;

onHrH <= '1' when (Hcont>=limLhrH and Hcont<limRhrH and Vcont>=limSupHora and Vcont<=limInfHora) else '0';
onHrL <= '1' when (Hcont>=limLhrL and Hcont<limRhrL and Vcont>=limSupHora and Vcont<=limInfHora) else '0';
onMinH<= '1' when (Hcont>=limLminH and Hcont<limRminH and Vcont>=limSupHora and Vcont<=limInfHora) else '0';
onMinL<= '1' when (Hcont>=limLminL and Hcont<limRminL and Vcont>=limSupHora and Vcont<=limInfHora) else '0';
on2p  <= '1' when (Hcont>=limLdosPuntos and Hcont<limRdosPuntos and Vcont>=limSupHora and Vcont<=limInfHora) else '0';


celAntenaLB <= '1' when (Hcont>limLcelCarcazaLB and Hcont<limRcelAntenaLB and Vcont>limSUPcelAntenaLB and Vcont<limSUPcelCarcazaLB) else'0';
celPantallaLB <= '1' when (Hcont>limLcelCarcazaLB+3 and Hcont<limRcelCarcazaLB-3 and Vcont>limSUPcelPantallaLB and Vcont<limINFcelPantallaLB) else '0';
celTecladoLB <= '1' when (Hcont>limLcelCarcazaLB+3 and Hcont<limRcelCarcazaLB-3 and Vcont>limSUPcelTecladoLB and Vcont<limINFcelTecladoLB) else '0';
celCarcazaLB <= '1' when (celTecladoLB='0' and celPantallaLB='0' and Hcont>limLcelCarcazaLB and Hcont<limRcelCarcazaLB and Vcont>limSUPcelCarcazaLB and Vcont<limINFcelCarcazaLB and onFrameCel='0') else '0';
onFrameCel <= '1' when (Hcont>limLcelCarcazaLB and Hcont<limRcelAntenaLB and Vcont=limSUPcelAntenaLB) else
                   '1' when ((Hcont=limLcelCarcazaLB or Hcont=limRcelAntenaLB) and Vcont>limSUPcelAntenaLB and Vcont<limSUPcelCarcazaLB) else
                   '1' when (Hcont>limLcelCarcazaLB+3 and Hcont<limRcelCarcazaLB-3 and (Vcont=limSUPcelPantallaLB or Vcont=limINFcelPantallaLB)) else
                   '1' when ((Hcont=limLcelCarcazaLB+3 or Hcont=limRcelCarcazaLB-3) and Vcont>limSUPcelPantallaLB and Vcont<limINFcelPantallaLB) else
                   '1' when (Hcont>limLcelCarcazaLB+3 and Hcont<limRcelCarcazaLB-3 and (Vcont=limSUPcelTecladoLB or Vcont=limINFcelTecladoLB)) else
                   '1' when ((Hcont=limLcelCarcazaLB+3 or Hcont=limRcelCarcazaLB-3) and Vcont>limSUPcelTecladoLB and Vcont<limINFcelTecladoLB) else
                   '1' when (Hcont>limLcelCarcazaLB and Hcont<limRcelCarcazaLB and (Vcont=limSUPcelCarcazaLB or Vcont=limINFcelCarcazaLB)) else
                   '1' when ((Hcont=limLcelCarcazaLB or Hcont=limRcelCarcazaLB) and Vcont>limSUPcelCarcazaLB and Vcont<limINFcelCarcazaLB) else
                   '0';

lockCentroLB  <= '1' when (Hcont>limLlockCandadoLB+10 and Hcont<limRlockCandadoLB-10 and Vcont>limSUPlockCandadoLB+10 and Vcont<limINFlockCandadoLB-10) else'0';
lockCandadoLB <= '1' when (onFrameLock='0' and lockCentroLB='0' and Hcont>limLlockCandadoLB and Hcont<limRlockCandadoLB and Vcont>limSUPlockCandadoLB and Vcont<limINFlockCandadoLB) else'0';
lockRVarcoLB  <= '1' when (Hcont>limLlockCandadoLB and Hcont< limRlockRVarcoLB and Vcont>=limSUPlockRVarcoLB and Vcont<limSUPlockCandadoLB) else'0';
lockHarcoLB   <= '1' when (Hcont>limLlockCandadoLB and Hcont<limRlockCandadoLB and Vcont>limSUPlockHarcoLB and Vcont<limSUPlockRVarcoLB) else'0';
lockLV1arcoLB <= '1' when (onFrameLock='0' and Hcont>limLlockLVarcoLB and Hcont<limRlockCandadoLB and Vcont>=limSUPlockRVarcoLB and Vcont<= limINFlockLVarcoLB) else'0';
lockLV2arcoLB <= '1' when (edoAlarma='1' and Hcont>limLlockLVarcoLB and Hcont<limRlockCandadoLB and Vcont>limINFlockLVarcoLB and Vcont< limSUPlockCandadoLB) else'0';
onFrameLock   <= '1' when (Hcont>limLlockCandadoLB+10 and Hcont<limRlockCandadoLB-10 and (Vcont=limSUPlockCandadoLB+10 or Vcont=limINFlockCandadoLB-10)) else
                      '1' when ((Hcont=limLlockCandadoLB+10 or Hcont=limRlockCandadoLB-10) and Vcont>limSUPlockCandadoLB+10 and Vcont<limINFlockCandadoLB-10) else
                      '1' when (Hcont>limLlockCandadoLB and Hcont<limRlockCandadoLB and (Vcont=limSUPlockCandadoLB or Vcont=limINFlockCandadoLB)) else
                      '1' when ((Hcont=limLlockCandadoLB or Hcont=limRlockCandadoLB) and Vcont>limSUPlockCandadoLB and Vcont<limINFlockCandadoLB) else
                      '1' when (Hcont=limRlockRVarcoLB and Vcont>limSUPlockHarcoLB and Vcont<limSUPlockCandadoLB) else
                      '1' when (Hcont=limLlockCandadoLB and  Vcont>limSUPlockRVarcoLB and Vcont<limSUPlockCandadoLB) else

                    '1' when (Hcont>limLlockCandadoLB and Hcont<limRlockCandadoLB and Vcont=limSUPlockHarcoLB) else
                    '1' when (Hcont>limRlockRVarcoLB  and Hcont<limLlockLVarcoLB  and Vcont=limSUPlockRVarcoLB) else

                    '1' when (Vcont>limSUPlockHarcoLB and Vcont<=limINFlockLVarcoLB and Hcont=limLlockLVarcoLB) else
                    '1' when (Vcont>limSUPlockRVarcoLB and Vcont<=limINFlockLVarcoLB and Hcont=limRlockCandadoLB) else

                    '1' when (edoAlarma='1' and Vcont>limINFlockLVarcoLB and Vcont< limSUPlockCandadoLB and Hcont=limRlockCandadoLB) else
                    '1' when (edoAlarma='1' and Vcont>limINFlockLVarcoLB and Vcont< limSUPlockCandadoLB and Hcont=limLlockLVarcoLB) else
                    '1' when (edoAlarma='0' and Hcont>limLlockLVarcoLB and Hcont<limRlockCandadoLB and Vcont=limINFlockLVarcoLB+1) else

                    '1' when ((Hcont=limLlockCandadoLB or Hcont=limRlockCandadoLB) and Vcont>limSUPlockHarcoLB and Vcont<=limSUPlockRVarcoLB) else
                    '0';

-----------------------------------------------------------------------------------------



------**c1******************************************************************************
cieloC1         <= '1' when (onCap1='1' and anyElementoC1='0' and onFrameC1='0') else '0';
pastoFueraC1 <= '1' when (onCap1='1' and Vcont > limHorizonteC1 and Vcont < limSupBanquetaC1 and Hcont <limLMuroFrente1C1 ) else
                     '1' when (onCap1='1' and Vcont > limHorizonteC1 and Vcont < limSupBanquetaC1 and Hcont >limRMuroFrente2C1 ) else '0';
banquetaC1   <= '1' when (onCap1='1' and Vcont>limSupBanquetaC1 and Vcont<limInfBanquetaC1-3 and Hcont <limLentreBanquetasC1) else
                     '1' when (onCap1='1' and Vcont>limSupBanquetaC1 and Vcont<limInfBanquetaC1-3 and Hcont >limRentreBanquetasC1) else '0';
bordeBanquetaC1 <= '1' when (onCap1='1' and Vcont>limInfBanquetaC1-3 and Vcont<limInfBanquetaC1 and Hcont <limLentreBanquetasC1) else
                     '1' when (onCap1='1' and Vcont>limInfBanquetaC1-3 and Vcont<limInfBanquetaC1 and Hcont >limRentreBanquetasC1) else '0';
calleC1          <= '1' when (onCap1='1' and onFrameC1='0' and Vcont>limInfCasaC1 and Vcont <= limSupBanquetaC1 and Hcont >limLentreBanquetasC1 and Hcont <limRentreBanquetasC1 and portonC1='0') else
                      '1' when (onCap1='1' and onFrameC1='0' and Vcont>limSupBanquetaC1 and Vcont <= limInfBanquetaC1 and Hcont >limLentreBanquetasC1 and Hcont <limRentreBanquetasC1) else
                      '1' when (onCap1='1' and Vcont>limInfBanquetaC1) else '0';

faroFLc1 <= '1' when (onCap1='1' and Hcont>limLMuroFrente1C1-2 and Hcont<limRPosteFLc1+2 and Vcont>limSupFarosFc1 and Vcont<limSupPostesFc1) else '0';
faroBLc1 <= '1' when (onCap1='1' and Hcont>limLPosteBLc1-2 and Hcont<limRPosteBLc1+2 and Vcont>limSupFarosBc1 and Vcont<limSupPostesBc1) else '0';
faroFRc1 <= '1' when (onCap1='1' and Hcont>limLPosteFRc1-2 and Hcont<limRMuroFrente2C1+2 and Vcont>limSupFarosFc1 and Vcont<limSupPostesFc1) else '0';
faroBRc1 <= '1' when (onCap1='1' and Hcont>limLPosteBRc1-2 and Hcont<limRPosteBRc1+2 and Vcont>limSupFarosBc1 and Vcont<limSupPostesBc1) else '0';
anyFaroC1<= faroFLc1 or faroBLc1 or faroFRc1 or faroBRc1;


posteFLc1 <= '1' when (onCap1='1' and Vcont>limSupPostesFc1 and Vcont<limSupMuroFrenteC1 and Hcont>limLMuroFrente1C1 and Hcont<limRPosteFLc1) else '0';
posteBLc1 <= '1' when (onCap1='1' and Vcont>limSupPostesBc1 and Vcont<limSupMuroAtrasC1 and Hcont>limLPosteBLc1 and Hcont<limRPosteBLc1) else '0';
posteFRc1 <= '1' when (onCap1='1' and Vcont>limSupPostesFc1 and Vcont<limSupMuroFrenteC1 and Hcont>limLPosteFRc1 and Hcont<limRMuroFrente2C1) else '0';
posteBRc1 <= '1' when (onCap1='1' and Vcont>limSupPostesBc1 and Vcont<limSupMuroAtrasC1 and Hcont>limLPosteBRc1 and Hcont<limRPosteBRc1) else '0';
anyPosteC1<= posteFLc1 or posteBLc1 or posteFRc1 or posteBRc1;

muroFc1 <= '1' when (onCap1='1' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limLMuroFrente1C1  and Hcont<limRMuroFrente1C1-45 ) else
               '1' when (onCap1='1' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limLMuroFrente2C1+30  and Hcont<limRMuroFrente2C1 ) else '0';

muroFcurvoC1 <= '1' when (onCap1='1' and onFrameC1='0' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>=limRMuroFrente1C1-45  and Hcont<limRMuroFrente1C1-25 ) else
                    '1' when (onCap1='1' and onFrameC1='0' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limLMuroFrente2C1  and Hcont<=limLMuroFrente2C1+30 ) else '0';
```

```
muroBc1 <= '1' when (onCap1='1' and Vcont> limSupMuroAtrasC1 and Vcont<limSupMuroFrenteC1 and Hcont>limRPosteFLc1  and Hcont<limRMuroAtras1C1 ) else
            '1' when (onCap1='1' and Vcont> limSupMuroAtrasC1 and Vcont<limSupMuroFrenteC1 and Hcont>limLMuroAtras2C1  and Hcont<limLPosteFRc1 ) else '0';

paredPuertaFrenteC1 <=  '1' when (onCap1='1' and onFrameC1='0' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limRMuroFrente1C1-25 and Hcont<limRMuroFrente1C1-20) else
                        '1' when (onCap1='1' and onFrameC1='0' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limRMuroFrente1C1-5 and Hcont<limRMuroFrente1C1) else '0';


--//abre y cierra porton
process(clkMov2)
begin
        if clkMov2='1' and clkMov2'event then
                        if(edoPorton='0' and limLPortonC1Aux>limLPortonC1 and xPortonC1>"000000000")then
                                xPortonC1<= xPortonC1-'1';
                        elsif (edoPorton='1' and limLPortonC1aux<limRentreBanquetasC1)then
                                xPortonC1<= xPortonC1+'1';
                        end if;
        end if;
end process;



--//abre y cierra puerta
process(clkMov2)
begin
        if clkMov2='1' and clkMov2'event then
                if (edoPuerta='0' and limLPuertaFrenteC1aux>limLPuertaFrenteC1 and xPuertaFrenteC1>"00000000") then
                        xPuertaFrenteC1<= xPuertaFrenteC1-'1';
                elsif (edoPuerta='1' and limLPuertaFrenteC1aux<limRPuertaFrenteC1-2) then
                  xPuertaFrenteC1<= xPuertaFrenteC1+'1';
                end if;
        end if;

end process;


limLPuertaFrenteC1aux<= limLPuertaFrenteC1 + xPuertaFrenteC1;
limLPortonC1aux<= limLPortonC1 + xPortonC1;

puertaFrenteC1 <= '1' when (onCap1='1' and Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and Hcont>limLPuertaFrenteC1aux and Hcont<limRPuertaFrenteC1) else '0';
portonC1 <= '1' when (onCap1='1' and Hcont>limLPortonC1aux and Hcont<limRPortonC1 and Vcont>limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 ) else '0';


puertaCasaC1 <= '1' when (onCap1='1' and Vcont>limSupPuertaCasaC1 and Vcont<limInfFachadaP1C1 and Hcont>limLPuertaCasaC1 and Hcont<limRPuertaCasaC1 and onFrameC1='0' and portonC1='0') else '0';
ventanaULFc1 <= '1' when (onCap1='1' and Vcont>limInfTechoUC1 and Vcont<limInfVentanasP2C1 and Hcont>limIntMuroLcasaC1 and Hcont<limLenMedioVentanasP2C1) else '0';
ventanaURFc1 <= '1' when (onCap1='1' and Vcont>limInfTechoUC1 and Vcont<limInfVentanasP2C1 and Hcont>limRenMedioVentanasP2C1 and Hcont<limIntMuroRcasaC1 ) else '0';
```

```
ventanaDLFc1 <= '1' when (onCap1='1' and Vcont>limSupVentanasP1C1 and Vcont<limInfVentanasP1C1 and Hcont>limIntMuroLcasaC1 and Hcont<limLintVentanasP1C1 ) else '0';
ventanaDRFc1 <= '1' when (onCap1='1' and Vcont>limSupVentanasP1C1 and Vcont<limInfVentanasP1C1 and Hcont>limRintVentanasP1C1 and Hcont<limIntMuroRcasaC1 ) else '0';
anyVentanaC1 <= ventanaULFc1 or ventanaURFc1 or ventanaDLFc1 or ventanaDRFc1;
marcoVentanasYpuertaC1 <= '1' when (onCap1='1' and onFrameC1='0' and puertaCasaC1='0'  and portonC1='0' and Vcont>limInfTechoDC1 and Vcont<limInfFachadaP1C1 and Hcont>limLintVentanasP1C1 and
                                    Hcont<limRintVentanasP1C1) else '0';

muroLtechoCasaC1  <= '1' when (onCap1='1' and Vcont>limSupAzoteaC1 and Vcont<limInfAzoteaC1 and Hcont>limRMuroAtras1C1 and Hcont< limIntMuroLcasaC1) else '0';
muroLcasaC1       <= '1' when (onCap1='1' and Vcont>limInfAzoteaC1 and Vcont<limSupMuroFrenteC1 and Hcont>limRMuroAtras1C1 and Hcont< limIntMuroLcasaC1) else '0';
muroRtechoCasaC1  <= '1' when (onCap1='1' and Vcont>limSupAzoteaC1 and Vcont<limInfAzoteaC1 and Hcont>limIntMuroRcasaC1 and Hcont< limLMuroAtras2C1) else '0';
muroRcasaC1       <= '1' when (onCap1='1' and Vcont>limInfAzoteaC1 and Vcont<limSupMuroFrenteC1 and Hcont>limIntMuroRcasaC1 and Hcont< limLMuroAtras2C1) else '0';

fachadaCasaC1 <= '1' when (onCap1='1' and Vcont> limInfTechoUC1 and Vcont<limInfFachadaP1C1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1
                           and anyVentanaC1='0' and puertaCasaC1='0' and marcoVentanasYpuertaC1='0' and balconUC1='0' and techoUC1='0' and techoDC1='0'
                           and onFrameC1='0' and portonC1='0' and muroFcurvoC1='0' and puertaFrenteC1='0' and paredPuertaFrenteC1='0') else '0';

azoteaC1 <= '1' when (onCap1='1' and Vcont>limSupAzoteaC1 and Vcont<limInfAzoteaC1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1) else '0';
techoUC1 <= '1' when (onCap1='1' and Vcont>limInfAzoteaC1 and Vcont<limInfTechoUC1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1) else '0';
techoDC1 <= '1' when (onCap1='1' and Vcont>limSupTechoDC1 and Vcont<limInfTechoDC1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1) else '0';
balconUC1 <= '1' when (onCap1='1' and Vcont>limInfFachadaP2C1 and Vcont<limSupTechoDC1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1) else '0';
balconDC1 <= '1' when (onCap1='1' and Vcont>limInfFachadaP1C1 and Vcont<limSupEscalon1C1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1
                           and onFrameC1='0' and portonC1='0' and muroFcurvoC1='0' and puertaFrenteC1='0' and paredPuertaFrenteC1='0') else '0';

escalon1C1<= '1' when (onCap1='1' and onFrameC1='0' and portonC1='0' and Vcont>limSupEscalon1C1 and Vcont<limSupEscalon2C1 and Hcont>limLEscalon1C1 and Hcont< limREscalon1C1) else '0';
escalon2C1<= '1' when (onCap1='1' and onFrameC1='0' and portonC1='0' and Vcont>limSupEscalon2C1 and Vcont<limSupEscalon3C1 and Hcont>limLEscalon2C1 and Hcont< limREscalon2C1) else '0';
escalon3C1<= '1' when (onCap1='1' and onFrameC1='0' and portonC1='0' and Vcont>limSupEscalon3C1 and Vcont<limInfCasaC1     and Hcont>limLEscalon3C1 and Hcont< limREscalon3C1) else '0';
anyEscalonC1 <= escalon1C1 or escalon2C1 or escalon3C1;

fachadaCasaEscalonesC1<= '1' when (onCap1='1' and onFrameC1='0' and Vcont>limSupEscalon1C1 and Vcont<limInfCasaC1 and Hcont>limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1
                                    and portonC1='0' and muroFcurvoC1='0' and puertaFrenteC1='0' and paredPuertaFrenteC1='0' and anyEscalonC1='0') else '0';

pastoDentroC1<= '1' when (onCap1='1' and puertaFrenteC1='0' and onFrameC1='0' and Vcont>limInfCasaC1 and Vcont<limSupBanquetaC1 and Hcont>limRMuroFrente1C1-20 and Hcont<limRMuroFrente1C1-5)else'0';

anyElementoC1<= '1' when (anyVentanaC1='1' or anyPosteC1='1' or anyFaroC1='1' or muroBc1='1') else
                '1' when (onCap1='1' and Vcont>limSupAzoteaC1-1 and Vcont<limHorizonteC1 and Hcont>limRMuroAtras1C1-1 and Hcont< limLMuroAtras2C1+1 ) else
                '1' when (onCap1='1' and Vcont> limHorizonteC1-1) else '0';

onFrameC1<= '1' when (Vcont> limSupPostesFc1 and Vcont<limHorizonteC1 and (Hcont=limLMuroFrente1C1 or Hcont=limRMuroFrente2C1 or Hcont=limRPosteFLc1 or Hcont=limLPosteFRc1)) else
            '1' when (Vcont> limSupPostesBc1 and Vcont<limSupMuroAtrasC1 and (Hcont=limLPosteBLc1 or Hcont=limRPosteBLc1 or Hcont=limLPosteBRc1 or Hcont=limRPosteBRc1)) else
            '1' when (Vcont= limSupMuroAtrasC1 and anyElementoC1='0' and Hcont>limRPosteFLc1 and Hcont<limLPosteFRc1) else
            '1' when (Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and (Hcont=limRMuroFrente1C1-25 or Hcont=limRMuroFrente1C1-20 or Hcont=limRMuroFrente1C1-5 or Hcont=limRMuroFrente1C1 or
                           Hcont=limRPortonC1)) else
            '1' when (Vcont> limSupFarosFc1 and Vcont<limSupPostesFc1 and (Hcont=limLMuroFrente1C1-2 or Hcont=limRPosteFLc1+2 or Hcont=limLPosteFRc1-2 or Hcont=limRMuroFrente2C1+2 )) else
            '1' when (Vcont> limSupFarosBc1 and Vcont<limSupPostesBc1 and (Hcont=limLPosteBLc1-2 or Hcont=limRPosteBLc1+2 or Hcont=limLPosteBRc1-2 or Hcont=limRPosteBRc1+2)) else
            '1' when (Hcont> limLMuroFrente1C1-2 and Hcont<limRPosteFLc1+2 and (Vcont=limSupFarosFc1 or Vcont=limSupPostesFc1)) else
            '1' when (Hcont> limLPosteBLc1-2 and Hcont<limRPosteBLc1+2 and (Vcont=limSupFarosBc1 or Vcont=limSupPostesBc1)) else
              '1' when (Hcont> limLPosteFRc1-2 and Hcont<limRMuroFrente2C1+2 and (Vcont=limSupFarosFc1 or Vcont=limSupPostesFc1)) else
```

```
                '1' when (Hcont> limLPosteBRc1-2 and Hcont<limRPosteBRc1+2 and (Vcont=limSupFarosBc1 or Vcont=limSupPostesBc1)) else
                '1' when (Hcont> limRMuroFrente1C1-45 and Hcont<limRMuroFrente1C1-20 and Vcont=limSupMuroFrenteC1) else
                '1' when (Hcont> limLMuroFrente2C1 and Hcont<limLMuroFrente2C1+30 and Vcont=limSupMuroFrenteC1) else
                '1' when (Vcont= limSupMuroFrenteC1 and Hcont>limRMuroFrente1C1-5 and Hcont<limRMuroFrente1C1) else
                '1' when (Hcont> limIntMuroLcasaC1 and Hcont< limIntMuroRcasaC1 and (Vcont=limInfFachadaP2C1 or Vcont=limSupTechoDC1 or Vcont=limInfTechoDC1) ) else
                '1' when (Vcont = limInfFachadaP1C1 and Hcont>limRMuroFrente1C1-20 and Hcont<limRMuroFrente1C1-5) else
                '1' when (Vcont = limInfFachadaP1C1 and Hcont>limLPortonC1 and Hcont<limRPortonC1 and portonC1='0') else
                '1' when (Vcont = limSupEscalon1C1 and Hcont>limRMuroFrente1C1-20 and Hcont<limRMuroFrente1C1-5) else
                '1' when (Vcont = limSupEscalon1C1 and Hcont>limLPortonC1 and Hcont<limRPortonC1 and portonC1='0') else
                '1' when (Vcont = limInfCasaC1 and Hcont>limRMuroFrente1C1-20 and Hcont<limRMuroFrente1C1-5) else
                '1' when (Vcont = limInfCasaC1 and Hcont>limLPortonC1 and Hcont<limRPortonC1 and portonC1='0') else
                '1' when (portonC1='0' and Vcont=limSupEscalon2C1 and Hcont>limLEscalon2C1 and Hcont< limREscalon2C1) else
                '1' when (portonC1='0' and Vcont=limSupEscalon3C1 and Hcont>limLEscalon3C1 and Hcont< limREscalon3C1) else
                '1' when (portonC1='0' and Vcont>limSupEscalon1C1 and Vcont<limSupEscalon2C1 and (Hcont=limLEscalon1C1 or Hcont= limREscalon1C1))else
                '1' when (portonC1='0' and Vcont>limSupEscalon2C1 and Vcont<limSupEscalon3C1 and (Hcont=limLEscalon2C1 or Hcont=limREscalon2C1))else
                '1' when (portonC1='0' and Vcont>limSupEscalon3C1 and Vcont<limInfCasaC1 and (Hcont=limLEscalon3C1 or Hcont=limREscalon3C1))else
                '1' when (portonC1='0' and Vcont>limSupPuertaCasaC1 and Vcont<limInfFachadaP1C1 and (Hcont=limLPuertaCasaC1 or Hcont=limRPuertaCasaC1))else
                '1' when (portonC1='0' and Vcont>limInfTechoDC1 and Vcont<limInfFachadaP1C1 and (Hcont=limLintVentanasP1C1 or Hcont=limRintVentanasP1C1))else
                '1' when ((Vcont=limSupVentanasP1C1 or Vcont=limInfVentanasP1C1) and Hcont>limRintVentanasP1C1 and Hcont<limIntMuroRcasaC1) else
                '1' when ((Vcont=limSupVentanasP1C1 or Vcont=limInfVentanasP1C1) and Hcont>limIntMuroLcasaC1 and Hcont<limLintVentanasP1C1 )else
                '1' when (Vcont=limInfVentanasP2C1 and Hcont>limRenMedioVentanasP2C1 and Hcont<limIntMuroRcasaC1)else
                '1' when (Vcont=limInfVentanasP2C1 and Hcont>limIntMuroLcasaC1 and Hcont<limLenMedioVentanasP2C1)else
                '1' when (Vcont>limInfTechoUC1 and Vcont<limInfVentanasP2C1 and (Hcont=limRenMedioVentanasP2C1 or Hcont=limLenMedioVentanasP2C1))else
                '1' when (Vcont=limSupPuertaCasaC1 and Hcont>limLPuertaCasaC1 and Hcont<limRPuertaCasaC1)else
                '1' when (Hcont>limLPortonC1aux and Hcont<limRPortonC1 and (Vcont=limSupMuroFrenteC1 or Vcont=limSupBanquetaC1))else
                '1' when ((Hcont=limLPortonC1aux or Hcont=limRPortonC1) and Vcont>limSupMuroFrenteC1 and Vcont<limSupBanquetaC1)else
                '1' when (Vcont> limSupMuroFrenteC1 and Vcont<limSupBanquetaC1 and (Hcont=limLPuertaFrenteC1aux or Hcont=limRPuertaFrenteC1))else
                '1' when ((Vcont=limSupMuroFrenteC1 or Vcont=limSupBanquetaC1) and Hcont>limLPuertaFrenteC1aux and Hcont<limRPuertaFrenteC1)else
                '0';

----------------------------------------------------------------------------------------------


----**c2*************************************************************************************

calleC2<= '1' when (onCap2='1' and Vcont>limInfCasaC2 and Hcont >limLIntBanquetaC2 and Hcont <limRIntBanquetaC2 and portonC2='0' and  onFrameC2='0') else
          '1' when (onCap2='1' and Vcont>limInfBanquetaC2 and onFrameC2='0') else '0';

banquetaC2<= '1' when (onCap2='1' and Vcont>limSupBanquetaC2 and Vcont<limInfBanquetaC2 and (Hcont <limLIntBanquetaC2 or Hcont >limRIntBanquetaC2 )) else '0';


--//abre y cierra porton

process(clkMov)
begin
        if clkMov='1' and clkMov'event then
```

```
                        if(edoPorton='0' and limLPortonC2aux>limLIntBanquetaC2 and xPortonC2>"000000000")then
                                        xPortonC2<= xPortonC2-'1';
                        elsif (edoPorton='1' and limLPortonC2aux<limRIntBanquetaC2-2)then
                                        xPortonC2<= xPortonC2+'1';
                        end if;
        end if;
end process;


--//abre y cierra puerta

process(clkMov)
begin
        if clkMov='1' and clkMov'event then
                        if(edoPuerta='0' and limLPuertaFrenteC2aux>limLPuertaFrenteC2 and xPuertaFrenteC2>"000000000")then
                                        xPuertaFrenteC2<= xPuertaFrenteC2-'1';
                        elsif (edoPuerta='1' and limLPuertaFrenteC2aux< limRPuertaFrenteC2-2)then
                                        xPuertaFrenteC2<= xPuertaFrenteC2+'1';
                        end if;
        end if;
end process;

limLPortonC2aux<= limLIntBanquetaC2 + xPortonC2;
limLPuertaFrenteC2aux<= limLPuertaFrenteC2 + xPuertaFrenteC2;

puertaFrenteC2<= '1' when (onCap2='1'  and Vcont > limSupPuertaPortonC2 and Vcont < limSupBanquetaC2 and Hcont >limLPuertaFrenteC2aux and Hcont <limRPuertaFrenteC2) else '0';
portonC2<= '1'  when (onCap2='1'  and Vcont > limSupPuertaPortonC2 and Vcont < limSupBanquetaC2 and Hcont >limLPortonC2aux and Hcont <limRIntBanquetaC2) else '0';

murosVc2<= '1' when (onCap2='1' and Hcont >limExtMuroIzqC2 and Hcont <limIntMuroIzqC2 and Vcont>limInfMuroAtrasC2 and Vcont<limSupMuroFrenteC2) else
            '1' when (onCap2='1' and Hcont >limIntMuroDerC2 and Hcont <limExtMuroDerC2 and Vcont>limInfMuroAtrasC2 and Vcont<limSupMuroFrenteC2) else
            '0';
murosHc2<= '1' when (onCap2='1' and  onFrameC2='0' and Hcont >limIntMuroIzqC2 and Hcont <limIntMuroDerC2 and Vcont>limSupMuroAtrasC2 and Vcont<limInfMuroAtrasC2) else
            '1' when (onCap2='1' and  onFrameC2='0' and Hcont >limIntMuroIzqC2 and Hcont <limIntMuroDerC2 and Vcont>limSupMuroFrenteC2 and Vcont<limSupBanquetaC2 and puertaFrenteC2 = '0' and portonC2 = '0' and
                        pastoDentroC2 = '0') else   '0';

pastoDentroC2<= '1' when (onCap2='1' and  onFrameC2='0' and azoteaC2='0' and albercaC2 ='0' and marcoAlbercaC2='0' and caminitoC2='0' and
                                Hcont >limIntMuroIzqC2 and Hcont <limIntMuroDerC2 and Vcont>limInfMuroAtrasC2 and Vcont<limSupMuroFrenteC2) else '0';
pastoFueraC2<= '1' when (onCap2='1' and Vcont<limSupMuroAtrasC2) else
                '1' when (onCap2='1' and Hcont<limExtMuroIzqC2  and Vcont>=limSupMuroAtrasC2 and Vcont<limSupBanquetaC2) else
                '1' when (onCap2='1' and Hcont>limExtMuroDerC2  and Vcont>=limSupMuroAtrasC2 and Vcont<limSupBanquetaC2) else '0';

azoteaC2<= '1' when (onCap2='1' and Hcont>limLCasaC2 and Hcont< limRCasaC2 and Vcont>limSupCasaC2 and Vcont<limInfCasac2) else '0';
caminitoC2<= '1' when (onCap2='1' and  onFrameC2='0' and Hcont>limLIntBanquetaC2 and Hcont<limRIntBanquetaC2  and Vcont>limInfCasaC2 and Vcont<limSupPuertaPortonC2) else '0';
albercaC2<= '1' when (onCap2='1' and anyOla='0' and Hcont>limLAlbercaC2 and Hcont<limRAlbercaC2 and Vcont>limSupAlbercaC2 and Vcont<limInfAlbercaC2 ) else '0';
marcoAlbercaC2<= '1' when (onCap2='1' and  onFrameC2='0' and albercaC2= '0' and Hcont>limLMAlbercaC2 and Hcont<limRMAlbercaC2 and Vcont>limSupMAlbercaC2 and Vcont<limInfMAlbercaC2 ) else '0';
```

```
faroBIc2<= '1' when (onCap2='1' and Hcont >limExtMuroIzqC2 and Hcont <limIntMuroIzqC2 and Vcont>limSupMuroAtrasC2 and Vcont<limInfMuroAtrasC2 ) else '0';
faroBDc2<= '1' when (onCap2='1' and Hcont >limIntMuroDerC2 and Hcont <limExtMuroDerC2 and Vcont>limSupMuroAtrasC2 and Vcont<limInfMuroAtrasC2 ) else '0';
faroFIc2<= '1' when (onCap2='1' and Hcont >limExtMuroIzqC2 and Hcont <limIntMuroIzqC2 and Vcont>limSupMuroFrenteC2 and Vcont<limSupBanquetaC2 ) else '0';
faroFDc2<= '1' when (onCap2='1' and Hcont >limIntMuroDerC2 and Hcont <limExtMuroDerC2 and Vcont>limSupMuroFrenteC2 and Vcont<limSupBanquetaC2) else '0';

ola1<= '1' when(onCap2='1' and Hcont>=limLOla1 and Hcont<limROla1 and Vcont=limOlasSup) else '0';
ola2<= '1' when(onCap2='1' and Hcont>=limLOla2 and Hcont<limROla2 and Vcont=limOlasInf) else '0';
ola3<= '1' when(onCap2='1' and Hcont>=limLOla3 and Hcont<limROla3 and Vcont=limOlasSup) else '0';
ola4<= '1' when(onCap2='1' and Hcont>=limLOla1 and Hcont<limROla1 and Vcont=limOlasInf) else '0';
ola5<= '1' when(onCap2='1' and Hcont>=limLOla2 and Hcont<limROla2 and Vcont=limOlasSup) else '0';
ola6<= '1' when(onCap2='1' and Hcont>=limLOla3 and Hcont<limROla3 and Vcont=limOlasInf) else '0';
anyOla <= ola1 or ola2 or ola3 or ola4 or ola5 or ola6;

onFrameC2 <=   '1' when (Vcont > limSupPuertaPortonC2 and Vcont < limSupBanquetaC2 and (Hcont=limLIntBanquetaC2 or Hcont=limRIntBanquetaC2)) else
               '1' when (Hcont>limLCasaC2 and Hcont< limRCasaC2 and (Vcont=limSupCasaC2 or Vcont=limInfCasac2)) else
               '1' when ((Hcont=limLCasaC2 or Hcont= limRCasaC2) and Vcont>limSupCasaC2 and Vcont<limInfCasac2) else
               '1' when (Hcont>limLAlbercaC2 and Hcont<limRAlbercaC2 and (Vcont=limSupAlbercaC2 or Vcont=limInfAlbercaC2)) else
               '1' when ((Hcont=limLAlbercaC2 or Hcont=limRAlbercaC2) and Vcont>limSupAlbercaC2 and Vcont<limInfAlbercaC2 ) else
               '1' when (Hcont>limLMAlbercaC2 and Hcont<limRMAlbercaC2 and (Vcont=limSupMAlbercaC2 or Vcont=limInfMAlbercaC2)) else
               '1' when ((Hcont=limLMAlbercaC2 or Hcont=limRMAlbercaC2) and Vcont>limSupMAlbercaC2 and Vcont<limInfMAlbercaC2)else
               '1' when (Vcont > limSupPuertaPortonC2 and Vcont < limSupBanquetaC2 and (Hcont =limLPuertaFrenteC2 or Hcont =limLPuertaFrenteC2aux or Hcont=limRPuertaFrenteC2))else
               '1' when ((Vcont=limSupPuertaPortonC2 or Vcont=limSupBanquetaC2) and Hcont >limLPuertaFrenteC2aux and Hcont <limRPuertaFrenteC2)else
               '1' when (Vcont > limSupPuertaPortonC2 and Vcont < limSupBanquetaC2 and (Hcont=limLPortonC2aux or Hcont=limRIntBanquetaC2))else
               '1' when ((Vcont = limSupPuertaPortonC2 or Vcont = limSupBanquetaC2) and Hcont >limLPortonC2aux and Hcont <limRIntBanquetaC2)else
               '0';

------------------------------------------------------------------------------------------



----**c3*****************************************************************************************
cieloC3<= '1' when (onCap3='1' and pastoFueraC3='0' and banquetaC3='0' and murosVc3='0' and muroHc3='0'  and muroRcasaC3='0' and anyVentanaC3='0' and anyEscalonC3='0' and posteFRc3='0'
                  and posteBRc3='0' and faroFRc3='0' and faroBRc3='0' and  onFrameC3='0') else '0';
pastoFueraC3<= '1' when (onCap3='1' and Vcont>limInfCasaC3) else '0';

banquetaC3<= '1' when (onCap3='1' and Vcont<limInfCasaC3 and Vcont>limSupBanquetaC3 and Hcont<limLMuroFrenteC3) else '0';

murosVc3<= '1' when (onCap3='1' and Vcont>limSupMurosC3 and Vcont<limInfCasaC3 and Hcont>limLMuroFrenteC3 and Hcont<limRMuroFrenteC3) else
           '1' when (onCap3='1' and Vcont>limSupMurosC3 and Vcont<limInfCasaC3 and Hcont>limLMuroAtrasC3 and Hcont<limRMuroAtrasC3) else '0';
muroHc3<=  '1' when (onCap3='1' and  onFrameC3='0' and Vcont>limSupMurosC3 and Vcont<limInfCasaC3 and Hcont>limRMuroFrenteC3 and Hcont<limLCasaC3 and anyEscalonC3='0') else
           '1' when (onCap3='1' and Vcont>limSupMurosC3 and Vcont<limInfCasaC3 and Hcont>limRCasaC3 and Hcont<limLMuroAtrasC3 ) else '0';
muroRcasaC3<= '1' when (onCap3='1' and  onFrameC3='0' and Vcont>limSupCasaC3 and Vcont<limInfCasaC3 and Hcont>limLCasaC3 and Hcont<limRCasaC3 and anyVentanaC3='0') else '0';

faroFRc3 <= '1' when (onCap3='1' and Vcont>limSupFarosC3 and Vcont<limInfFarosC3 and Hcont>limLFaroFRC3 and Hcont<limRFaroFRC3) else '0';
faroBRc3 <= '1' when (onCap3='1' and Vcont>limSupFarosC3 and Vcont<limInfFarosC3 and Hcont>limLFaroBRC3 and Hcont<limRFaroBRC3) else '0';
posteFRc3<= '1' when (onCap3='1' and Vcont>limInfFarosC3 and Vcont<limSupMurosC3 and Hcont>limLPosteFRC3 and Hcont<limRPosteFRC3) else '0';
```

```
posteBRc3<= '1' when (onCap3='1' and Vcont>limInfFarosC3 and Vcont<limSupMurosC3 and Hcont>limLPosteBRC3 and Hcont<limRPosteBRC3) else '0';

ventanaURFc3<= '1' when (onCap3='1' and Hcont>limLVentanasLC3 and Hcont<limRVentanasLC3 and Vcont<limInfVentanasP2C3 and Vcont> limSupVentanasP2C3) else '0';
ventanaURBc3<= '1' when (onCap3='1' and Hcont>limLVentanasRC3 and Hcont<limRVentanasRC3 and Vcont<limInfVentanasP2C3 and Vcont> limSupVentanasP2C3) else '0';
ventanaDRFc3<= '1' when (onCap3='1' and Hcont>limLVentanasLC3 and Hcont<limRVentanasLC3 and Vcont<limInfVentanasP1C3 and Vcont> limSupVentanasP1C3) else '0';
ventanaDRBc3<= '1' when (onCap3='1' and Hcont>limLVentanasRC3 and Hcont<limRVentanasRC3 and Vcont<limInfVentanasP1C3 and Vcont> limSupVentanasP1C3) else '0';
anyVentanaC3<= ventanaURFc3 or ventanaURBc3 or ventanaDRFc3 or ventanaDRBc3;

escalon1c3<= '1' when (onCap3='1' and  onFrameC3='0' and Hcont<=limLCasaC3 and Hcont>limLEscalones1C3 and Vcont>limSupEscalon1C3 and Vcont<limSupEscalon2C3) else '0';
escalon2c3<= '1' when (onCap3='1' and  onFrameC3='0' and Hcont<=limLCasaC3 and Hcont>limLEscalones2C3 and Vcont>=limSupEscalon2C3 and Vcont<=limSupEscalon3C3) else '0';
escalon3c3<= '1' when (onCap3='1' and  onFrameC3='0' and Hcont<=limLCasaC3 and Hcont>limLEscalones3C3 and Vcont>limSupEscalon3C3 and Vcont<limInfCasaC3) else '0';
anyEscalonC3<= escalon1C3 or escalon2C3 or escalon3C3;

onFrameC3<=   '1' when (onCap3='1' and Vcont=limInfCasaC3) else
              '1' when (Vcont=limSupMurosC3 and Hcont>limLMuroFrenteC3 and Hcont<limLCasaC3) else
              '1' when (Vcont=limSupMurosC3 and Hcont>limRCasaC3 and Hcont<limRMuroAtrasC3) else
              '1' when (Vcont>limSupCasaC3 and Vcont<limInfCasaC3 and (Hcont=limLCasaC3 or Hcont=limRCasaC3)) else
              '1' when ((Vcont=limSupCasaC3 or Vcont=limInfCasaC3) and Hcont>limLCasaC3 and Hcont<limRCasaC3) else
              '1' when (onCap3='1' and  Vcont=limSupBanquetaC3 and Hcont<limLMuroFrenteC3) else
              '1' when (Vcont>limSupMurosC3 and Vcont<limInfCasaC3 and(Hcont=limLMuroFrenteC3 or Hcont=limRMuroFrenteC3 or Hcont=limLMuroAtrasC3 or Hcont=limRMuroAtrasC3 or Hcont=limLCasaC3 or
                      Hcont=limRCasaC3 )) else
              '1' when (Vcont>limInfFarosC3 and Vcont<limSupMurosC3 and (Hcont=limLPosteFRC3 or Hcont=limRPosteFRC3 or Hcont=limLPosteBRC3 or Hcont=limRPosteBRC3)) else
              '1' when ((Vcont=limInfFarosC3 or Vcont=limSupMurosC3) and Hcont>limLPosteFRC3 and Hcont<limRPosteFRC3) else
              '1' when ((Vcont=limInfFarosC3 or Vcont=limSupMurosC3) and Hcont>limLPosteBRC3 and Hcont<limRPosteBRC3) else
              '1' when (Vcont>limSupFarosC3 and Vcont<limInfFarosC3 and (Hcont=limLFaroFRC3 or Hcont=limRFaroFRC3 or Hcont=limLFaroBRC3 or Hcont=limRFaroBRC3)) else
              '1' when ((Vcont=limSupFarosC3 or Vcont=limInfFarosC3) and Hcont>limLFaroFRC3 and Hcont<limRFaroFRC3) else
              '1' when ((Vcont=limSupFarosC3 or Vcont=limInfFarosC3) and Hcont>limLFaroBRC3 and Hcont<limRFaroBRC3) else
              '1' when (Hcont>limLVentanasLC3 and Hcont<limRVentanasLC3 and (Vcont=limInfVentanasP1C3 or Vcont=limSupVentanasP1C3)) else
              '1' when (Hcont>limLVentanasRC3 and Hcont<limRVentanasRC3 and (Vcont=limInfVentanasP1C3 or Vcont=limSupVentanasP1C3)) else
              '1' when ((Hcont=limLVentanasRC3 or Hcont=limRVentanasRC3 or Hcont=limLVentanasLC3 or Hcont=limRVentanasLC3) and Vcont<limInfVentanasP1C3 and Vcont> limSupVentanasP1C3) else
              '1' when (Hcont>limLVentanasLC3 and Hcont<limRVentanasLC3 and (Vcont=limInfVentanasP2C3 or Vcont=limSupVentanasP2C3)) else
              '1' when (Hcont>limLVentanasRC3 and Hcont<limRVentanasRC3 and (Vcont=limInfVentanasP2C3 or Vcont=limSupVentanasP2C3)) else
              '1' when ((Hcont=limLVentanasRC3 or Hcont=limRVentanasRC3 or Hcont=limLVentanasLC3 or Hcont=limRVentanasLC3) and Vcont<limInfVentanasP2C3 and Vcont> limSupVentanasP2C3) else
              '1'  when (Hcont<limLCasaC3 and Hcont>limLEscalones1C3 and Vcont=limSupEscalon1C3) else
              '1'  when (Hcont=limLEscalones1C3 and Vcont>limSupEscalon1C3 and Vcont<limSupEscalon2C3) else
              '1'  when (Hcont<=limLEscalones1C3 and Hcont>limLEscalones2C3 and Vcont=limSupEscalon2C3) else
              '1'  when (Hcont=limLEscalones2C3 and Vcont>limSupEscalon2C3 and Vcont<limSupEscalon3C3) else
              '1'  when (Hcont<=limLEscalones2C3 and Hcont>limLEscalones3C3 and Vcont=limSupEscalon3C3) else
              '1'  when (Hcont=limLEscalones3C3 and Vcont>limSupEscalon3C3 and Vcont<limInfCasaC3) else
              '0';


--========================================================================
--Uso de MEMORIA=====================================================
--Trae de MEMORIA los digitos=============================================

HrHGEN:  testram port map (hrH_address, hrH);
HrLGEN:  testram port map (hrL_address, hrL);
MinHGEN: testram port map (minH_address, minH);
MinLGEN: testram port map (minL_address, minL);
DosPuntosGEN: testram port map (dosPuntos_address, dosPuntos);

        ----calcula la address de RAM de los digitos

process(Vcont) --calcula la fila del número que se quiere llamar
begin
       if ((Vcont >=limSupHora) and (Vcont <=limInfHora)) then
              hrH_address(2 downto 0) <= Vcont - limSupHora;  --de 0 a 7
              hrL_address(2 downto 0) <= Vcont - limSupHora;  --de 0 a 7
              minH_address(2 downto 0)<= Vcont - limSupHora;  --de 0 a 7
              minL_address(2 downto 0)<= Vcont - limSupHora;  --de 0 a 7
              dosPuntos_address(2 downto 0) <= Vcont - limSupHora;  --de 0 a 7
       else
              hrH_address(2 downto 0) <= "000";
              hrL_address(2 downto 0) <= "000";
              minH_address(2 downto 0) <= "000";
              minL_address(2 downto 0) <= "000";
              dosPuntos_address(2 downto 0) <= "000";
       end if;
end process;

--calcula cuál número se quiere llamar
hrH_address(6 downto 3)  <= contHrH(3 downto 0);
hrL_address(6 downto 3)  <= contHrL(3 downto 0);
minH_address(6 downto 3) <= contMinH(3 downto 0);
minL_address(6 downto 3) <= contMinL(3 downto 0);
dosPuntos_address(6 downto 3) <= "1010" when (CLK1s='1') else "1011";


        --Calcula Hora------------------------------

process(CLK)
begin
if CLK='1' and CLK'event then
       if (contHrL(3)='1' and contHrL(0)='1' and resetMinH='1')then
                     resetHrL<='1';
       elsif (contHrL(1)='1' and contHrL(0)='1' and contHrH(1)='1' and resetMinH='1') then
                     resetHrL<='1';
       else
```

```
            resetHrL<='0';
    end if;
end if;
end process;

resetMinL <= contMinL(3) and contMinL(0); --cuando llega a 9
resetMinH <= contMinH(2) and contMinH(0) and resetMinL; --cuando llega a 5
resetHrH  <= '1' when (contHrH(1)='1' and resetHrL='1' ) else '0';--cuando está en 2 y la HrLow se reseta

contadorMinLow:  contadorHr port map(CLK1min,resetMinL ,'1',setTimeRx,minLRx,contMinL);
contadorMinHigh: contadorHr port map(CLK1min,resetMinH,resetMinL ,setTimeRx,minHRx,contMinH);
contadorHrLow:   contadorHr port map(CLK1min,resetHrL,resetMinH,setTimeRx,horaLRx,contHrL);
contadorHrHigh:  contadorHr port map(CLK1min,resetHrH,resetHrL,setTimeRx,horaHRx,contHrH);

--===========================================================================================


--***Trae de MEMORIA el letrerito ON/OFF=====================================================


onO_GEN:   testram port map (onO_address, onO);
onN_GEN:   testram port map (onN_address, onN);
offO_GEN:  testram port map (offO_address, offO);
offF1_GEN: testram port map (offF1_address, offF1);
offF2_GEN: testram port map (offF2_address, offF2);


     ----calcula la address de RAM del letrerito ON/OFF

process(Vcont)--calcula la fila de la letra que se quiere llamar
begin
    if ((Vcont >=limSupLetreroONOFF) and (Vcont <=limInfLetreroONOFF)) then
        onO_address(2 downto 0)   <= Vcont - limSupLetreroONOFF;  --de 0 a 7
        onN_address(2 downto 0)   <= Vcont - limSupLetreroONOFF;  --de 0 a 7
        offO_address(2 downto 0)  <= Vcont - limSupLetreroONOFF;  --de 0 a 7
        offF1_address(2 downto 0) <= Vcont - limSupLetreroONOFF;  --de 0 a 7
        offF2_address(2 downto 0) <= Vcont - limSupLetreroONOFF;  --de 0 a 7
    else
        onO_address(2 downto 0)  <= "000";
        onN_address(2 downto 0)  <= "000";
        offO_address(2 downto 0) <= "000";
        offF1_address(2 downto 0)<= "000";
        offF2_address(2 downto 0)<= "000";
    end if;
end process;
```

```
--calcula cuál letra se quiere llamar
onO_address(6 downto 3)  <= "0000";
onN_address(6 downto 3)  <= "1100";
offO_address(6 downto 3) <= "0000";
offF1_address(6 downto 3)<= "1101";
offF2_address(6 downto 3)<= "1101";

--==============================================================================================================================


--Calcula Colores de cosas=============================================================================================

--color del Reloj de LEFT BAR****************************************

RGBhrH  <= "111111111111" when (hrH(3)='1' and Hcont(1 downto 0)="10" and contHrH>0) else
                              "111111111111" when (hrH(2)='1' and Hcont(1 downto 0)="11" and contHrH>0) else
                              "111111111111" when (hrH(1)='1' and Hcont(1 downto 0)="00" and contHrH>0) else
                              "111111111111" when (hrH(0)='1' and Hcont(1 downto 0)="01" and contHrH>0) else
                              "000000000000"; --pinta blanco los 1's  y de negro los 0's para que se aprecie el digito

RGBhrL  <= "111111111111" when (hrL(3)='1' and Hcont(1 downto 0)="11") else
                              "111111111111" when (hrL(2)='1' and Hcont(1 downto 0)="00") else
                              "111111111111" when (hrL(1)='1' and Hcont(1 downto 0)="01") else
                              "111111111111" when (hrL(0)='1' and Hcont(1 downto 0)="10") else
                              "000000000000";--pinta blanco los 1's  y de negro los 0's para que se aprecie el digito

RGBminH <= "111111111111" when (minH(3)='1' and Hcont(1 downto 0)="01") else
                              "111111111111" when (minH(2)='1' and Hcont(1 downto 0)="10") else
                              "111111111111" when (minH(1)='1' and Hcont(1 downto 0)="11") else
                              "111111111111" when (minH(0)='1' and Hcont(1 downto 0)="00") else
                              "000000000000";--pinta blanco los 1's  y de negro los 0's para que se aprecie el digito

RGBminL <= "111111111111" when (minL(3)='1' and Hcont(1 downto 0)="10") else
                              "111111111111" when (minL(2)='1' and Hcont(1 downto 0)="11") else
                              "111111111111" when (minL(1)='1' and Hcont(1 downto 0)="00") else
                              "111111111111" when (minL(0)='1' and Hcont(1 downto 0)="01") else
                              "000000000000";--pinta blanco los 1's  y de negro los 0's para que se aprecie el digito

RGB2p  <=  "111111111111" when (dosPuntos(3)='1' and Hcont(1 downto 0)="00") else
                              "111111111111" when (dosPuntos(2)='1' and Hcont(1 downto 0)="01") else
                              "111111111111" when (dosPuntos(1)='1' and Hcont(1 downto 0)="10") else
                              "111111111111" when (dosPuntos(0)='1' and Hcont(1 downto 0)="11") else
                              "000000000000";--pinta blanco los 1's  y de negro los 0's para que se aprecie el digito
```

```
--color del letrero ON/OFF e interruptor******************************

RGBonO  <=      RGBLetreroONverde when (onO(3)='1' and Hcont(1 downto 0)="10" and edoSistema='1') else
                RGBLetreroONverde when (onO(2)='1' and Hcont(1 downto 0)="11" and edoSistema='1') else
                RGBLetreroONverde when (onO(1)='1' and Hcont(1 downto 0)="00" and edoSistema='1') else
                RGBLetreroONverde when (onO(0)='1' and Hcont(1 downto 0)="01" and edoSistema='1') else
                 "111111111111"    when (onO(3)='1' and Hcont(1 downto 0)="10" and edoSistema='0') else
                 "111111111111"    when (onO(2)='1' and Hcont(1 downto 0)="11" and edoSistema='0') else
                 "111111111111"    when (onO(1)='1' and Hcont(1 downto 0)="00" and edoSistema='0') else
                 "111111111111"    when (onO(0)='1' and Hcont(1 downto 0)="01" and edoSistema='0') else
                 "000000000000";

RGBonN  <=      RGBLetreroONverde when (onN(3)='1' and Hcont(1 downto 0)="11" and edoSistema='1') else
                RGBLetreroONverde when (onN(2)='1' and Hcont(1 downto 0)="00" and edoSistema='1') else
                RGBLetreroONverde when (onN(1)='1' and Hcont(1 downto 0)="01" and edoSistema='1') else
                RGBLetreroONverde when (onN(0)='1' and Hcont(1 downto 0)="10" and edoSistema='1') else
                "111111111111"    when (onN(3)='1' and Hcont(1 downto 0)="11" and edoSistema='0') else
                "111111111111"    when (onN(2)='1' and Hcont(1 downto 0)="00" and edoSistema='0') else
                "111111111111"    when (onN(1)='1' and Hcont(1 downto 0)="01" and edoSistema='0') else
                "111111111111"    when (onN(0)='1' and Hcont(1 downto 0)="10" and edoSistema='0') else
                 "000000000000";


RGBoffO <=      RGBLetreroOFFrojo when (offO(3)='1' and Hcont(1 downto 0)="01" and edoSistema='0') else
                RGBLetreroOFFrojo when (offO(2)='1' and Hcont(1 downto 0)="10" and edoSistema='0') else
                RGBLetreroOFFrojo when (offO(1)='1' and Hcont(1 downto 0)="11" and edoSistema='0') else
                RGBLetreroOFFrojo when (offO(0)='1' and Hcont(1 downto 0)="00" and edoSistema='0') else
                 "111111111111"    when (offO(3)='1' and Hcont(1 downto 0)="01" and edoSistema='1') else
                 "111111111111"    when (offO(2)='1' and Hcont(1 downto 0)="10" and edoSistema='1') else
                 "111111111111"    when (offO(1)='1' and Hcont(1 downto 0)="11" and edoSistema='1') else
                 "111111111111"    when (offO(0)='1' and Hcont(1 downto 0)="00" and edoSistema='1') else
                 "000000000000";

RGBoffF1 <=     RGBLetreroOFFrojo when (offF1(3)='1' and Hcont(1 downto 0)="10" and edoSistema='0') else
                RGBLetreroOFFrojo when (offF1(2)='1' and Hcont(1 downto 0)="11" and edoSistema='0') else
                RGBLetreroOFFrojo when (offF1(1)='1' and Hcont(1 downto 0)="00" and edoSistema='0') else
                RGBLetreroOFFrojo when (offF1(0)='1' and Hcont(1 downto 0)="01" and edoSistema='0') else
                "111111111111"   when (offF1(3)='1' and Hcont(1 downto 0)="10" and edoSistema='1') else
                "111111111111"  when (offF1(2)='1' and Hcont(1 downto 0)="11" and edoSistema='1') else
                "111111111111"  when (offF1(1)='1' and Hcont(1 downto 0)="00" and edoSistema='1') else
                "111111111111"   when (offF1(0)='1' and Hcont(1 downto 0)="01" and edoSistema='1') else
                "000000000000";

RGBofff2 <=     RGBLetreroOFFrojo when (offF2(3)='1' and Hcont(1 downto 0)="11" and edoSistema='0') else
                RGBLetreroOFFrojo when (offF2(2)='1' and Hcont(1 downto 0)="00" and edoSistema='0') else
                RGBLetreroOFFrojo when (offF2(1)='1' and Hcont(1 downto 0)="01" and edoSistema='0') else
```

```
               RGBLetreroOFFrojo when (offF2(0)='1' and Hcont(1 downto 0)="10" and edoSistema='0') else
                "111111111111"           when (offF2(3)='1' and Hcont(1 downto 0)="11" and edoSistema='1') else
                "111111111111"           when (offF2(2)='1' and Hcont(1 downto 0)="00" and edoSistema='1') else
                "111111111111"           when (offF2(1)='1' and Hcont(1 downto 0)="01" and edoSistema='1') else
                "111111111111"           when (offF2(0)='1' and Hcont(1 downto 0)="10" and edoSistema='1') else
                "000000000000";

RGBinterruptor <= RGBLetreroONverde when(edoSistema='1')else RGBLetreroOFFrojo;

--color del celular****************************

RGBcelularAux <= "111111111111" when (CLKmedioS='1') else "000000000111"; --alterna entre rojo y azul
RGBcelular <= RGBcelularAux when (edoAlarma='1' and edoRobo='1' and edoSistema='1') else "010101010101"; --colorEmergencia/colorNormal(gris)
RGBcelPantalla<= "000010110001";
RGBcelTeclado<=  "011001100011";


--color del candado*****************************

RGBlockCentro <= "101010100011";
RGBlockArco   <= "010101010101";
RGBlockCandado<= "000011110000" when (edoAlarma='1' and edoRobo='0' and edoSistema='1') else
           "111100000000" when (edoAlarma='1' and edoRobo='1' and edoSistema='1') else
           "111111111111";


--color de olas*********************************

RGBolasOFFaux <= RGBalbercaC2   when (clkMov3='1') else "100010000110";
RGBolasONaux  <= "100010000110" when (clkMov3='1') else RGBalbercaC2;

RGBolasOFF<= "100010000110" when (edoAlberca='0')else RGBolasOFFaux;
RGBolasON <= RGBalbercaC2 when (edoAlberca='0')else RGBolasONaux;


--color de intensidades*********************************
--==faros==

RGBfaroFLc1<= "111011101110" when (intFaroFL="00") else
                              "001100110000" when (intFaroFL="01") else
                              "011101110000" when (intFaroFL="10") else
                              "111111110000" when (intFaroFL="11") else
                              "000000000000";

RGBfaroFLc2<= "111011101110" when (intFaroFL="00") else
                              "001100110000" when (intFaroFL="01") else
```

```
                                    "011101110000" when (intFaroFL="10") else
                                    "111111110000" when (intFaroFL="11") else
                                    "000000000000";

RGBfaroFRc1<= "111011101110" when (intFaroFR="00") else
                                    "001100110000" when (intFaroFR="01") else
                                    "011101110000" when (intFaroFR="10") else
                                    "111111110000" when (intFaroFR="11") else
                                    "000000000000";

RGBfaroFRc2<= "111011101110" when (intFaroFR="00") else
                                    "001100110000" when (intFaroFR="01") else
                                    "011101110000" when (intFaroFR="10") else
                                    "111111110000" when (intFaroFR="11") else
                                    "000000000000";

RGBfaroFRc3<= "111011101110" when (intFaroFR="00") else
                                    "001100110000" when (intFaroFR="01") else
                                    "011101110000" when (intFaroFR="10") else
                                    "111111110000" when (intFaroFR="11") else
                                    "000000000000";

RGBfaroBLc1<= "111011101110" when (intFaroBL="00") else
                                    "001100110000" when (intFaroBL="01") else
                                    "011101110000" when (intFaroBL="10") else
                                    "111111110000" when (intFaroBL="11") else
                                    "000000000000";

RGBfaroBLc2<= "111011101110" when (intFaroBL="00") else
                                    "001100110000" when (intFaroBL="01") else
                                    "011101110000" when (intFaroBL="10") else
                                    "111111110000" when (intFaroBL="11") else
                                    "000000000000";

RGBfaroBRc1<= "111011101110" when (intFaroBR="00") else
                                    "001100110000" when (intFaroBR="01") else
                                    "011101110000" when (intFaroBR="10") else
                                    "111111110000" when (intFaroBR="11") else
                                    "000000000000";

RGBfaroBRc2<= "111011101110" when (intFaroBR="00") else
                                    "001100110000" when (intFaroBR="01") else
                                    "011101110000" when (intFaroBR="10") else
                                    "111111110000" when (intFaroBR="11") else
                                    "000000000000";
```

```
RGBfaroBRc3<= "111011101110" when (intFaroBR="00") else
                              "001100110000" when (intFaroBR="01") else
                              "011101110000" when (intFaroBR="10") else
                              "111111110000" when (intFaroBR="11") else
                              "000000000000";
----
--==ventanas==

RGBventanaULFc1<="111011101110" when (intventanaULF="00") else
                             "001100110000" when (intventanaULF="01") else
                              "011101110000" when (intventanaULF="10") else
                              "111111110000" when (intventanaULF="11") else
                              "000000000000";



RGBventanaURFc1<="111011101110" when (intventanaURF="00") else
                              "001100110000" when (intventanaURF="01") else
                              "011101110000" when (intventanaURF="10") else
                              "111111110000" when (intventanaURF="11") else
                              "000000000000";

RGBventanaURFc3<="111011101110" when (intventanaURF="00") else
                              "001100110000" when (intventanaURF="01") else
                              "011101110000" when (intventanaURF="10") else
                              "111111110000" when (intventanaURF="11") else
                              "000000000000";


RGBventanaDLFc1<="111011101110" when (intventanaDLF="00") else
                              "001100110000" when (intventanaDLF="01") else
                              "011101110000" when (intventanaDLF="10") else
                              "111111110000" when (intventanaDLF="11") else
                              "000000000000";


RGBventanaDRFc1<="111011101110" when (intventanaDRF="00") else
                              "001100110000" when (intventanaDRF="01") else
                              "011101110000" when (intventanaDRF="10") else
                              "111111110000" when (intventanaDRF="11") else
                              "000000000000";

RGBventanaDRFc3<="111011101110" when (intventanaDRF="00") else
                              "001100110000" when (intventanaDRF="01") else
                              "011101110000" when (intventanaDRF="10") else
                              "111111110000" when (intventanaDRF="11") else
```

```
                                        "000000000000";


RGBventanaDRBc3<="111011101110" when (intventanaDRB="00") else
                                        "001100110000" when (intventanaDRB="01") else
                                        "011101110000" when (intventanaDRB="10") else
                                        "111111110000" when (intventanaDRB="11") else
                                        "000000000000";


RGBventanaURBc3<="111011101110" when (intventanaURB="00") else
                                        "001100110000" when (intventanaURB="01") else
                                        "011101110000" when (intventanaURB="10") else
                                        "111111110000" when (intventanaURB="11") else
                                        "000000000000";

---------------

--// determina los colores a través de relojes. De esta manera el color va cambiando y al dibujarse en el
--// monitor producen diversos efectos dependiendo del reloj utilizado.

process(CLK25)
          begin
                    if clk25'event and clk25='1' then
                              RGBporton(4) <= not RGBporton(4);
                              RGBporton(9 downto 8) <= RGBporton(11 downto 8) + '1';
                              RGBazoteaC1(10)<=not RGBazoteaC1(10);
                              RGBazoteaC1(6)<=not RGBazoteaC1(6);
                              RGBazoteaC1(2)<=not RGBazoteaC1(2);
                              RGBazoteaC2(10)<=not RGBazoteaC2(10);
                              RGBazoteaC2(6)<=not RGBazoteaC2(6);
                              RGBazoteaC2(2)<=not RGBazoteaC2(2);
                    end if;
end process;


process(CLKcuadro)
          begin
                    if CLKcuadro'event and CLKcuadro='1' then
                              RGBmuroFC1(11 downto 10)<= RGBmuroFC1(11 downto 10) + '1';
                              RGBmurosCurvosC1(11 downto 10)<= RGBmurosCurvosC1(11 downto 10)+'1';
                              RGBmarcoVentanasYpuertaC1(10 downto 9)<= RGBmarcoVentanasYpuertaC1(10 downto 9)+'1';
                              RGBtechitosC1(3 downto 2)<= RGBtechitosC1(3 downto 2)+'1';
                              RGBpastoDentroC1(1 downto 0)<= RGBpastoDentroC1(1 downto 0)+'1';
                    end if;
end process;
```

```
process(CLK)
        begin
                if CLK'event and CLK='1' then
                        RGBmuroBC1(9 downto 8) <= RGBmuroBC1(9 downto 8) + '1';
                        RGBalbercaC2(5 downto 4) <= RGBalbercaC2(5 downto 4) + '1';
                        RGBfachadaCasa(3 downto 0) <= RGBfachadaCasa(3 downto 0)+'1';
                        RGBbalconesC1(10 downto 9)<= RGBbalconesC1(10 downto 9) + '1';
                        RGBbalconesC1(1 downto 0)<=RGBbalconesC1(1 downto 0)+'1';
                        RGBpuertaCasaC1(6 downto 5)<=RGBpuertaCasaC1(6 downto 5)+'1';
                        RGBmurosLateralesTechoC1(3 downto 2)<= RGBmurosLateralesTechoC1(3 downto 2)+'1';
                        RGBbordeBanquetaC1(1 downto 0)<=RGBbordeBanquetaC1(1 downto 0)+'1';
                        RGBbordeBanquetaC1(5 downto 4)<=RGBbordeBanquetaC1(5 downto 4)+'1';
                        RGBbordeBanquetaC1(9 downto 8)<=RGBbordeBanquetaC1(9 downto 8)+'1';
                        RGBpastoFueraC1(1 downto 0)<= RGBpastoFueraC1(1 downto 0)+'1';
                end if;
end process;

--=======================================================================

--GENERA Salidas auxiliares de RGB de cada CAP=================================

RGBc1<= RGBcalleC1 when (calleC1='1') else
                  RGBbordeBanquetaC1 when (banquetaC1='1') else
                  RGBbordeBanquetaC1 when (bordeBanquetaC1='1') else
                  RGBpastoFueraC1 when (pastoFueraC1='1') else
                  RGBfaroFLc1 when (faroFLc1='1')else
                  RGBfaroFRc1 when (faroFRc1='1')else
                  RGBfaroBLc1 when (faroBLc1='1')else
                  RGBfaroBRc1 when (faroBRc1='1')else
                  RGBanyPoste when (anyPosteC1='1') else
                  RGBporton when (portonC1='1' or puertaFrenteC1='1') else
                  RGBparedPuertaFrenteC1 when (paredPuertaFrenteC1='1') else
                  RGBmuroBc1 when (muroFc1='1') else
                  RGBmuroBc1 when (muroFcurvoC1='1') else
                  RGBmurosCurvosC1 when (muroBc1='1') else
                  RGBventanaULFc1 when (ventanaULFc1='1') else
                  RGBventanaURFc1 when (ventanaURFc1='1') else
                  RGBventanaDLFc1 when (ventanaDLFc1='1') else
                  RGBventanaDRFc1 when (ventanaDRFc1='1') else
                  RGBazoteaC1 when (muroRcasaC1='1' or muroLcasaC1='1') else
                  RGBbordeBanquetaC1 when (muroRtechoCasaC1='1' or muroLtechoCasaC1='1') else
                  RGBbordeBanquetaC1 when (techoDC1='1' or techoUC1='1') else
                  RGBbalconesC1 when (balconUC1='1' or balconDC1='1') else
                  RGBazoteaC1 when (azoteaC1='1') else
                  RGBpuertaCasaC1 when (puertaCasaC1='1') else
```

```
              RGBmarcoVentanasYpuertaC1 when (marcoVentanasYpuertaC1='1') else
              RGBfachadaCasa when (fachadaCasaC1='1') else
              RGBmarcoVentanasYpuertaC1 when (anyEscalonC1='1')else
              RGBazoteaC1 when (fachadaCasaEscalonesC1='1') else
              RGBpastoDentroC1 when (pastoDentroC1='1') else
              RGBcieloC1 when (cieloC1='1') else
              "000000000000" when (onFrameC1='1') else
              "000000000000";


RGBc2<= RGBcalleC2         when (calleC2='1') else
        RGBbordeBanquetaC1 when (banquetaC2='1') else
        RGBmuroBC1         when (murosVc2='1' or murosHc2='1') else
        RGBazoteaC2        when (azoteaC2='1') else
        RGBolasOFF         when (ola1='1' or ola2='1' or ola3='1')else
        RGBolasON          when (ola4='1' or ola5='1' or ola6='1')else
        RGBalbercaC2       when (albercaC2='1') else
        RGBporton          when (puertaFrenteC2='1' or portonC2='1') else
        RGBmarcoAlbercaC2  when (marcoAlbercaC2='1') else
        RGBpastoFueraC1    when (pastoFueraC2='1') else
        RGBpastoDentroC1   when (pastoDentroC2='1') else
        RGBfaroBLc2        when (faroBIc2='1') else
        RGBfaroBRc2        when (faroBDc2='1') else
        RGBfaroFLc2        when (faroFIc2='1') else
        RGBfaroFRc2        when (faroFDc2='1') else
        "000000000000"      when (onFrameC2='1') else
        "000000000000";


RGBc3<= RGBventanaDRBc3    when (ventanaDRBc3='1') else
        RGBventanaDRFc3    when (ventanaDRFc3='1') else
        RGBventanaURBc3    when (ventanaURBc3='1') else
        RGBventanaURFc3    when (ventanaURFc3='1') else
        RGBpastoFueraC1    when (pastoFueraC3='1') else
        RGBbordeBanquetaC1 when (banquetaC3='1') else
        RGBmurosCurvosC1   when (murosVc3='1') else
        RGBfaroFRc3        when (faroFRc3='1') else
        RGBfaroBRc3        when (faroBRc3='1') else
        RGBanyPoste        when (posteFRc3='1' or posteBRc3='1') else
        RGBmarcoVentanasYpuertaC1  when (anyEscalonC3='1' ) else
        RGBfachadaCasa     when (muroRcasaC3='1') else
        RGBmuroFc1         when (muroHc3='1') else
        RGBcieloC3         when (cieloC3='1') else
        "000000000000"     when (onFrameC3='1') else
        "000000000000";
RGBlb<= RGBhrH                   when (onHrH='1') else
```

```
                    RGBhrL              when (onHrL='1') else
                    RGBminH             when (onMinH='1') else
                    RGBminL             when (onMinL='1') else
                    RGB2p               when (on2p='1') else
                    RGBonO              when (onON_o='1') else
                    RGBonN              when (onON_n='1') else
                    RGBoffO             when (onOFF_o='1') else
                    RGBoffF1            when (onOFF_f1='1') else
                    RGBoffF2            when (onOFF_f2='1') else
                    RGBinterruptor      when (interruptorLB='1')else
                    RGBcelular          when (celAntenaLB='1' or celCarcazaLB='1') else
                    RGBcelPantalla      when (celPantallaLB='1')else
                    RGBcelTeclado       when (celTecladoLB='1')else
                    RGBlockCentro       when (lockCentroLB='1') else
                    RGBlockCandado      when (lockCandadoLB='1') else
                    RGBlockArco         when (lockRVarcoLB='1' or lockHarcoLB='1' or lockLV1arcoLB='1' or lockLV2arcoLB='1') else
                    "111111111111"      when (onFrameLock='1' or onFrameCel='1' or onFrameInterruptor='1') else
                    "000000000000";

--=================================================================

--GENERA Salida final de RGB==================================================

Red<= RGBc1(11 downto 8) when (onCap1 = '1' and onCapsFrame = '0') else
                   RGBc2(11 downto 8) when (onCap2 = '1' and onCapsFrame = '0') else
                   RGBc3(11 downto 8) when (onCap3 = '1' and onCapsFrame = '0') else
                   RGBlb(11 downto 8) when (onLeftBar = '1' and onCapsFrame = '0') else
                   "1111" when (onCapsFrame = '1') else
                   "0000";

Green<= RGBc1(7 downto 4) when (onCap1 = '1' and onCapsFrame = '0') else
                    RGBc2(7 downto 4) when (onCap2 = '1' and onCapsFrame = '0') else
                    RGBc3(7 downto 4) when (onCap3 = '1' and onCapsFrame = '0') else
                    RGBlb(7 downto 4) when (onLeftBar = '1' and onCapsFrame = '0') else
                    "1111" when (onCapsFrame = '1') else
                    "0000";

Blue<= RGBc1(3 downto 0) when (onCap1 = '1' and onCapsFrame = '0') else
                   RGBc2(3 downto 0) when (onCap2 = '1' and onCapsFrame = '0') else
                   RGBc3(3 downto 0) when (onCap3 = '1' and onCapsFrame = '0') else
                   RGBlb(3 downto 0) when (onLeftBar = '1' and onCapsFrame = '0') else
                   "1111" when (onCapsFrame = '1') else
                   "0000";
--=================================================================
end Behavioral;
```

CLKs

```
entity CLKs is
          Port (  Clk_in : in  STD_LOGIC;
         CLK_out: out STD_LOGIC);
end CLKs;

architecture Behavioral of CLKs is
signal cont: STD_LOGIC_VECTOR(25 downto 0);
signal CLK1: STD_LOGIC;
begin

          process(Clk_in)
          begin
                    if CLK_in'event and Clk_in='1' then
                                        CLK1 <= not CLK1;
                    end if;
          end process;

          CLK_out<= CLK1;

end Behavioral;
```

CONTADOR

```
entity contador is
    Port ( inc : in std_logic;
         reset : in std_logic;
                               run: in std_logic;
         s : inout std_logic_vector(9 downto 0));
end contador;

architecture Behavioral of contador is

begin
process(inc, reset,run)
begin
          if (inc='1') and inc'event and run='1' then
                    if reset='1' then
                              s<="0000000000";
                    else
                              s<= s+1;
                    end if;
end if;
end process;
end Behavioral;
```

CONTADORHr

```
entity contadorHr is
    Port ( inc : in std_logic;
         reset : in std_logic;
                      run: in std_logic;
                      set: in std_logic;
                      newTime: in std_logic_vector(3 downto 0);
         s : inout std_logic_vector(3 downto 0));
end contadorHr;

architecture Behavioral of contadorHr is

begin
process(inc, reset,run,set)
begin
          if (set='1')then
                    s<=newTime;
          elsif (inc='1') and inc'event and run='1' then
          if reset='1' then
                    s<="0000";
          else
                    s<= s+1;
          end if;
          end if;
end process;
end Behavioral;
```

RAM

```
entity testram is Port (
        address: in std_logic_vector(6 downto 0);
        data:    out std_logic_vector(3 downto 0)
        );
end testram;

architecture behavioral of testram is

type mem_array is array (0 to 111) of std_logic_vector(3 downto 0);
constant characters: mem_array := (
        -- 0
        "0000", "0110", "1001", "1001", "1001", "1001", "1001", "0110",
        -- 1
        "0000", "0001", "0001", "0001", "0001", "0001", "0001", "0001",
        -- 2
        "0000", "1110", "0001", "0001", "0111", "1000", "1000", "1111",
        -- 3
        "0000", "1110", "0001", "0001", "1111", "0001", "0001", "1110",
        -- 4
        "0000", "1001", "1001", "1001", "1111", "0001", "0001", "0001",
        -- 5
        "0000", "1111", "1000", "1000", "1110", "0001", "0001", "1110",
        -- 6
        "0000", "0111", "1000", "1000", "1110", "1001", "1001", "0110",
        -- 7
        "0000", "1111", "0001", "0001", "0001", "0001", "0001", "0001",
        -- 8
        "0000", "0110", "1001", "1001", "0110", "1001", "1001", "0110",
        -- 9
        "0000", "0110", "1001", "1001", "0111", "0001", "0001", "0001",
        -- :
        "0000", "0000", "0110", "0110", "0000", "0110", "0110", "0000",
        -- " "
        "0000", "0000", "0000", "0000", "0000", "0000", "0000", "0000",
        -- N
        "0000", "0000", "0000", "0000", "1001", "1101", "1011", "1001",
        -- F
        "0000", "0110", "1001", "1000", "1111", "1000", "1000", "1000"   );

begin
        process (address )
        begin
                data <= characters(conv_integer(address));
        end process;
end behavioral;
```

TECLADO

```
entity keyboard is
        port(   keyboard_clk   : IN     STD_LOGIC;
                keyboard_data  : IN     STD_LOGIC;
                clock_25Mhz    : IN     STD_LOGIC;
                reset          : IN     STD_LOGIC;
                read_kb        : IN     STD_LOGIC;
                scan_code      : OUT    STD_LOGIC_VECTOR(7 DOWNTO 0);
                scan_ready     : OUT    STD_LOGIC);
end keyboard;

architecture Behavioral of keyboard is
        signal INCNT                    : std_logic_vector(3 downto 0);
        signal SHIFTIN                  : std_logic_vector(8 downto 0);
        signal READ_CHAR                : std_logic;
        signal INFLAG, ready_set        : std_logic;
        signal keyboard_clk_filtered    : std_logic;
        signal filter                   : std_logic_vector(7 downto 0);

begin

process (read_kb, ready_set)
begin
  if read_kb = '1' then
                scan_ready <= '0';
  elsif ready_set'EVENT and ready_set = '1' then
                scan_ready <= '1';
  end if;
end process;


--Este proceso filtra la raw signal del clock que viene del teclado usando un shift register y dos ands
Clock_filter: process
begin
        wait until clock_25Mhz'EVENT and clock_25Mhz= '1';
                filter (6 downto 0) <= filter(7 downto 1) ;
                filter(7) <= keyboard_clk;
                if filter = "11111111" then
                        keyboard_clk_filtered <= '1';
                elsif  filter= "00000000" then
                        keyboard_clk_filtered <= '0';
                end if;
end process Clock_filter;
```

```
--Este proceso lee data serial que llega de la terminal
process
begin
wait until (KEYBOARD_CLK_filtered'EVENT and KEYBOARD_CLK_filtered='1');
        if RESET='1' then
      INCNT <= "0000";
      READ_CHAR <= '0';
        elsif KEYBOARD_DATA='0' and READ_CHAR='0' then
      READ_CHAR<= '1';
      ready_set<= '0';
        elsif READ_CHAR = '1' then -- Shift in los sig 8 data bits para armar el scan code
      if INCNT < "1001" then
        INCNT <= INCNT + 1;
        SHIFTIN(7 downto 0) <= SHIFTIN(8 downto 1);
        SHIFTIN(8) <= KEYBOARD_DATA;
                        ready_set <= '0';
      else -- Fin del char de scan code, así que pon las banderas y salte del loop
              scan_code <= SHIFTIN(7 downto 0);
              READ_CHAR <='0';
              ready_set <= '1';
              INCNT <= "0000";
      end if;
   end if;
end process;

end Behavioral;
```

DISPLAY (código proporcionado por JorgeVillaseñor)

```
entity Display is
Port (
        -- Input
        Clk_in  : in  STD_LOGIC;                    -- Clock
        -- Data
        D0      : in  STD_LOGIC_VECTOR (3 downto 0);
        D1      : in  STD_LOGIC_VECTOR (3 downto 0);
        D2      : in  STD_LOGIC_VECTOR (3 downto 0);
        D3      : in  STD_LOGIC_VECTOR (3 downto 0);

        -- Output
        D7      : out  STD_LOGIC_VECTOR (7 downto 0);-- Data
        En      : out  STD_LOGIC_VECTOR (3 downto 0)-- Enable
                );
end Display;

architecture Behavioral of Display is
        -- Display signal
        signal Dmix : std_logic_vector(3 downto 0);

        -- Clock signals
        signal clk_cont : std_logic_vector(28 downto 0);
        signal clk          : std_logic;

        -- SM signals
        signal pstate     :std_logic_vector(3 downto 0);
        signal nstate     :std_logic_vector(3 downto 0);

begin

        -- Clock stuff
        C1:process(Clk_in)begin
                if(clk_in = '1' and clk_in'event)then
                        clk_cont <= clk_cont+1;
                end if;
        end process;

        clk <= clk_cont(8);

        -- State Machine
        nstate <= "0001" when pstate = "1000" else
                  "0010" when pstate = "0001" else
                  "0100" when pstate = "0010" else
                  "1000" when pstate = "0100" else  "0001";
```

```
          SM:process(clk)begin
          if(clk = '1' and clk'event)then
                  pstate <= nstate;
          end if;
          end process;

          -- Data coding
          Dmix <=D0 when pstate = "0001" else
                      D1 when pstate = "0010" else
                      D2 when pstate = "0100" else
                      D3 when pstate = "1000" else
                      x"A";

          D7(6 downto 0) <=             "0000001" when Dmix = "0000" else      -- 0
                                        "1001111" when Dmix = "0001" else      -- 1
                                        "0010010" when Dmix = "0010" else      -- 2
                                        "0000110" when Dmix = "0011" else      -- 3
                                        "1001100" when Dmix = "0100" else      -- 4
                                        "0100100" when Dmix = "0101" else      -- 5
                                        "0100000" when Dmix = "0110" else      -- 6
                                        "0001111" when Dmix = "0111" else      -- 7
                                        "0000000" when Dmix = "1000" else      -- 8
                                        "0000100" when Dmix = "1001" else      -- 9
                                        "1111111" when Dmix = "1010" else      -- " "
                                        "1110001" when Dmix = "1011" else      -- L
                                        "1111010" when Dmix = "1100" else      -- r
                                        "0001000" when Dmix = "1101" else      -- A
                                        "0100000" when Dmix = "1110" else      -- G
                                        "0111000" when Dmix = "1111" else      -- F
                                        "1010101" ;
          D7(7) <=  '0' when ((pstate = "0100") and (Dmix < "1010")) else '1';
          En <= not pstate;

end Behavioral;
```

UART_TX (código obtenido del moudlo KCPSM3 de PICOBLAZE a través de la página de Xilinx)

```
-- Main Entity for UART_TX

entity uart_tx is
    Port (            data_in : in std_logic_vector(7 downto 0);
                 write_buffer : in std_logic;
                 reset_buffer : in std_logic;
                 en_16_x_baud : in std_logic;
                   serial_out : out std_logic;
                  buffer_full : out std_logic;
             buffer_half_full : out std_logic;
                          clk : in std_logic);
    end uart_tx;

-- Start of Main Architecture for UART_TX

architecture macro_level_definition of uart_tx is

-- Components used in UART_TX and defined in subsequent entities.

-- Constant (K) Compact UART Transmitter
--
component kcuart_tx
    Port (        data_in : in std_logic_vector(7 downto 0);
           send_character : in std_logic;
             en_16_x_baud : in std_logic;
               serial_out : out std_logic;
              Tx_complete : out std_logic;
                      clk : in std_logic);
    end component;
--
-- 'Bucket Brigade' FIFO
--
component bbfifo_16x8
    Port (       data_in : in std_logic_vector(7 downto 0);
                data_out : out std_logic_vector(7 downto 0);
                   reset : in std_logic;
                   write : in std_logic;
                    read : in std_logic;
                    full : out std_logic;
               half_full : out std_logic;
            data_present : out std_logic;
                     clk : in std_logic);
    end component;
```

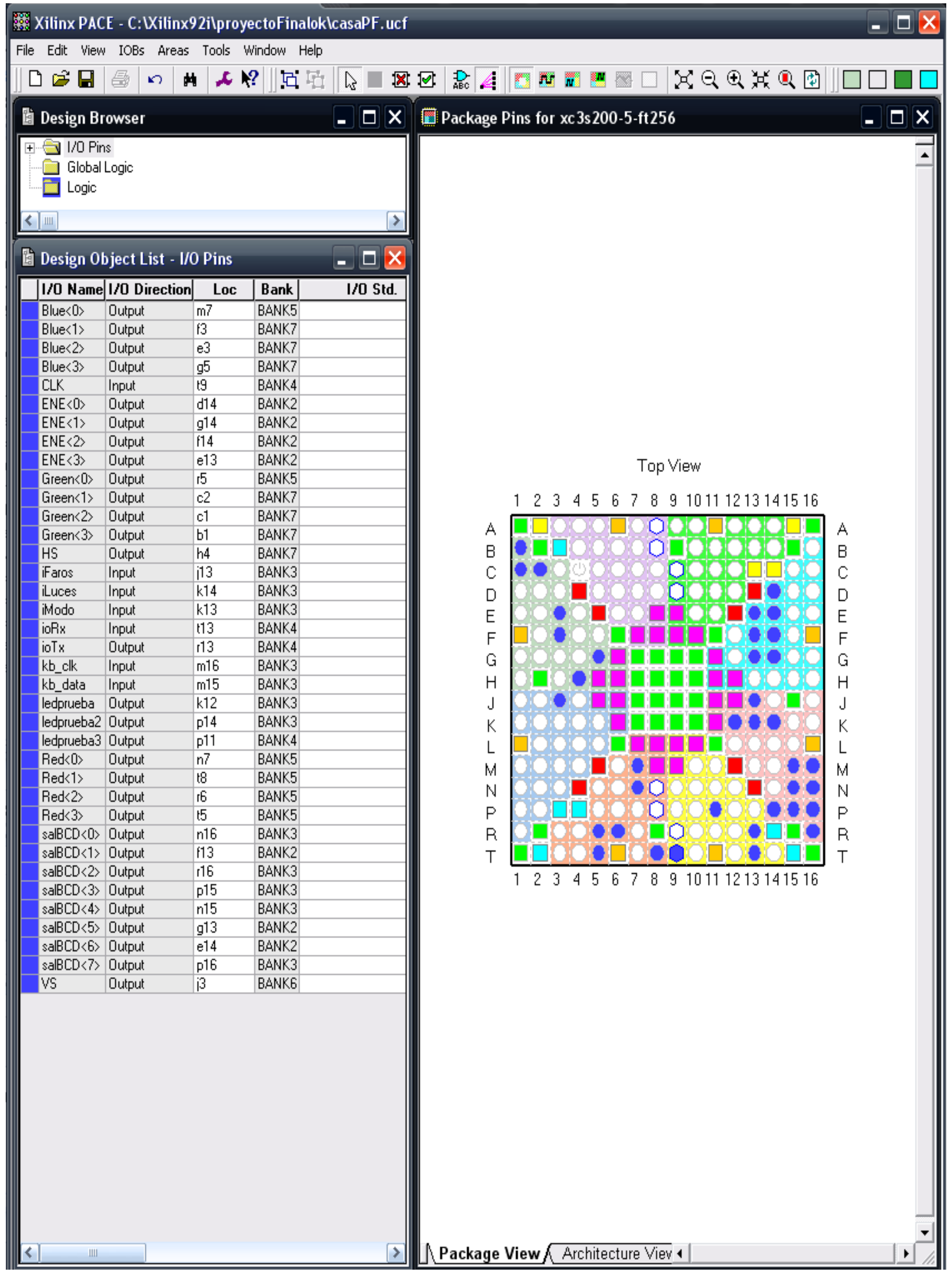
```
--------------------------------------------------------------------------------
--
-- Signals used in UART_TX

signal fifo_data_out      : std_logic_vector(7 downto 0);
signal fifo_data_present  : std_logic;
signal fifo_read          : std_logic;
--
-- Start of UART_TX circuit description
begin

  -- 8 to 1 multiplexer to convert parallel data to serial

  kcuart: kcuart_tx
  port map (        data_in => fifo_data_out,
             send_character => fifo_data_present,
               en_16_x_baud => en_16_x_baud,
                 serial_out => serial_out,
                Tx_complete => fifo_read,
                        clk => clk);


  buf: bbfifo_16x8
  port map (       data_in => data_in,
                  data_out => fifo_data_out,
                     reset => reset_buffer,
                     write => write_buffer,
                      read => fifo_read,
                      full => buffer_full,
                 half_full => buffer_half_full,
              data_present => fifo_data_present,
                       clk => clk);

end macro_level_definition;

-- END OF FILE UART_TX.VHD
```

UART_RX (código obtenido del moudlo KCPSM3 de PICOBLAZE a través de la página de Xilinx)

```
-- Main Entity for UART_RX
--
entity uart_rx is
    Port (            serial_in : in std_logic;
                       data_out : out std_logic_vector(7 downto 0);
                    read_buffer : in std_logic;
                   reset_buffer : in std_logic;
                   en_16_x_baud : in std_logic;
            buffer_data_present : out std_logic;
                    buffer_full : out std_logic;
               buffer_half_full : out std_logic;
                            clk : in std_logic);
    end uart_rx;
--

-- Start of Main Architecture for UART_RX
--
architecture macro_level_definition of uart_rx is
--
--------------------------------------------------------------------------------
-- Components used in UART_RX and defined in subsequent entities.
--------------------------------------------------------------------------------
-- Constant (K) Compact UART Receiver
component kcuart_rx
    Port (      serial_in : in std_logic;
                 data_out : out std_logic_vector(7 downto 0);
              data_strobe : out std_logic;
             en_16_x_baud : in std_logic;
                      clk : in std_logic);
    end component;
--
-- 'Bucket Brigade' FIFO
component bbfifo_16x8
    Port (       data_in : in std_logic_vector(7 downto 0);
                data_out : out std_logic_vector(7 downto 0);
                   reset : in std_logic;
                   write : in std_logic;
                    read : in std_logic;
                    full : out std_logic;
               half_full : out std_logic;
            data_present : out std_logic;
                     clk : in std_logic);
    end component;
```

```
-- Signals used in UART_RX

signal uart_data_out      : std_logic_vector(7 downto 0);
signal fifo_write         : std_logic;


--
-- Start of UART_RX circuit description
--
begin

  -- 8 to 1 multiplexer to convert parallel data to serial


  kcuart: kcuart_rx
  port map (      serial_in => serial_in,
                   data_out => uart_data_out,
                data_strobe => fifo_write,
               en_16_x_baud => en_16_x_baud,
                        clk => clk );



  buf: bbfifo_16x8
  port map (        data_in => uart_data_out,
                   data_out => data_out,
                      reset => reset_buffer,
                      write => fifo_write,
                       read => read_buffer,
                       full => buffer_full,
                  half_full => buffer_half_full,
               data_present => buffer_data_present,
                        clk => clk);

end macro_level_definition;
-- END OF FILE UART_RX.VHD
```

Asignación de Pines



I/O Name	I/O Direction	Loc	Bank	I/O Std.
Blue<0>	Output	m7	BANK5	
Blue<1>	Output	f3	BANK7	
Blue<2>	Output	e3	BANK7	
Blue<3>	Output	g5	BANK7	
CLK	Input	t9	BANK4	
ENE<0>	Output	d14	BANK2	
ENE<1>	Output	g14	BANK2	
ENE<2>	Output	f14	BANK2	
ENE<3>	Output	e13	BANK2	
Green<0>	Output	r5	BANK5	
Green<1>	Output	c2	BANK7	
Green<2>	Output	c1	BANK7	
Green<3>	Output	b1	BANK7	
HS	Output	h4	BANK7	
iFaros	Input	j13	BANK3	
iLuces	Input	k14	BANK3	
iModo	Input	k13	BANK3	
ioRx	Input	t13	BANK4	
ioTx	Output	r13	BANK4	
kb_clk	Input	m16	BANK3	
kb_data	Input	m15	BANK3	
ledprueba	Output	k12	BANK3	
ledprueba2	Output	p14	BANK3	
ledprueba3	Output	p11	BANK4	
Red<0>	Output	n7	BANK5	
Red<1>	Output	t8	BANK5	
Red<2>	Output	r6	BANK5	
Red<3>	Output	t5	BANK5	
salBCD<0>	Output	n16	BANK3	
salBCD<1>	Output	f13	BANK2	
salBCD<2>	Output	r16	BANK3	
salBCD<3>	Output	p15	BANK3	
salBCD<4>	Output	n15	BANK3	
salBCD<5>	Output	g13	BANK2	
salBCD<6>	Output	e14	BANK2	
salBCD<7>	Output	p16	BANK3	
VS	Output	j3	BANK6	

Comentarios

- Se utilizó un adaptador de Super VGA para aumentar la gama de colores posibles de 8 a 4096. El adaptador se conectó al puerto de expansión A1, sin embargo, también es posible conectarlo a cualquiera de los otros 2 puertos.

- En cuanto a la conexión serial, cabe decir que para configurar correctamente la comunicación fue necesario utilizar primero el HyperTerminal de la computadora para sincronizar adecuadamente ambos dispositivos. Esta solución resultó efectiva para evitar los problemas de comunicación que se presentaban cada vez que recién se conectaba el cable serial.