```
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

package dice_states is
        TYPE board_row_state is (initializing, board_row_1, board_row_2,
                 board_row_3, board_row_4, board_row_5, board_row_6,
                 board_row_7, board_row_8);
        TYPE throw_dice_state is ( move_piece1, reset_both , status_piece1,
                 bonus_piece1, penalty_piece1, stop_piece1,end_piece1,
                 move_piece2, status_piece2, bonus_piece2, penalty_piece2,
                 stop_piece2, end_piece2, who_wins);

        CONSTANT left_edge_x1   : STD_LOGIC_VECTOR(9 downto 0) :="0011000000";
                --192
        CONSTANT left_edge_x2   : STD_LOGIC_VECTOR(9 downto 0) :="0011010000";
                -- 192 +16= 208
        CONSTANT right_edge_x1  : STD_LOGIC_VECTOR(9 downto 0) :="0110100000";
        CONSTANT right_edge_x2  : STD_LOGIC_VECTOR(9 downto 0) :="0110110000";
        CONSTANT  Dx            : STD_LOGIC_VECTOR(9 downto 0)  :="0000100000";
        CONSTANT  Dy            : STD_LOGIC_VECTOR(9 downto 0)  :="0000100000";
        CONSTANT  Delta         : STD_LOGIC_VECTOR(4 downto 0)  :="10000";
        CONSTANT  startx1       : STD_LOGIC_VECTOR(9 downto 0)  :="0011000000";
        CONSTANT  startx2       : STD_LOGIC_VECTOR(9 downto 0)  :="0011010000";
        CONSTANT  starty1       : STD_LOGIC_VECTOR(9 downto 0)  :="0010100000";
        CONSTANT  starty2       : STD_LOGIC_VECTOR(9 downto 0)  :="0010110000";
        CONSTANT  panel_left_x  : STD_LOGIC_VECTOR(9 downto 0)  := left_edge_x1;
        CONSTANT  panel_right_x : STD_LOGIC_VECTOR(9 downto 0)  :="0111000000";
        CONSTANT  panel_top_y   : STD_LOGIC_VECTOR(9 downto 0)  := starty1;
        CONSTANT  panel_bottom_y: STD_LOGIC_VECTOR(9 downto 0)  :="0110100000";
        CONSTANT  raster_width  : STD_LOGIC_VECTOR(9 downto 0)  :="0000011111";
        CONSTANT  first_row_y1  : STD_LOGIC_VECTOR(5 downto 0)  :="001010";
                -- 160
        CONSTANT  first_row_y2  : STD_LOGIC_VECTOR(5 downto 0)  :="001011";
                -- 160+16=176

        CONSTANT  second_row_y1 : STD_LOGIC_VECTOR(5 downto 0)  :="001100";
                -- 160 +32 =192
        CONSTANT  second_row_y2 : STD_LOGIC_VECTOR(5 downto 0)  :="001101";
                -- 160 +32 +16=208

        CONSTANT  third_row_y1  : STD_LOGIC_VECTOR(5 downto 0)  :="001110";
                -- 160 +32 *2 = 224
        CONSTANT  third_row_y2  : STD_LOGIC_VECTOR(5 downto 0)  :="001111";
                -- 160 +32*2 +16 =240

        CONSTANT  fourth_row_y1 : STD_LOGIC_VECTOR(5 downto 0)  :="010000";
                -- 160 + 32*3 = 256
        CONSTANT  fourth_row_y2 : STD_LOGIC_VECTOR(5 downto 0)  :="010001";
                -- 160 + 32*3 +16 = 272

        CONSTANT  fifth_row_y1  : STD_LOGIC_VECTOR(5 downto 0)  :="010010";
                -- 160 + 32*4 = 288
        CONSTANT  fifth_row_y2  : STD_LOGIC_VECTOR(5 downto 0)  :="010011";
                -- 160 + 32*4 +16 = 304

        CONSTANT  sixth_row_y1  : STD_LOGIC_VECTOR(5 downto 0)  :="010100";
                 -- 160 +32*5 = 320
        CONSTANT  sixth_row_y2  : STD_LOGIC_VECTOR(5 downto 0)  :="010101";
                -- 160 +32*5 +16 = 336;

        CONSTANT  seventh_row_y1: STD_LOGIC_VECTOR(5 downto 0)  :="010110";
                -- 160 +32*6 = 352
```

```
CONSTANT  seventh_row_y2: STD_LOGIC_VECTOR(5 downto 0)  :="010111";
        -- 160 +32*6 +16 = 368

CONSTANT  eighth_row_y1 : STD_LOGIC_VECTOR(5 downto 0)  :="011000";
        -- 160 +32*7 = 384
CONSTANT  eighth_row_y2 : STD_LOGIC_VECTOR(5 downto 0)  :="011001";
        -- 160 +32*7 +16 = 400

CONSTANT  second_column_x1: STD_LOGIC_VECTOR(5 downto 0):="001110";
        -- 192+32 =224
CONSTANT  second_column_x2 : STD_LOGIC_VECTOR(5 downto 0) :="001111";
        -- 192+32+16=240


CONSTANT  third_column_x1: STD_LOGIC_VECTOR(9 downto 0) :="0100000000";
        -- 192+32*2 =256
CONSTANT  third_column_x1: STD_LOGIC_VECTOR(5 downto 0) :="010000";
        -- 192+32*2 =256
CONSTANT  third_column_x2: STD_LOGIC_VECTOR(9 downto 0) :="0100010000";
        -- 192+32*2+16 =272
CONSTANT  third_column_x2: STD_LOGIC_VECTOR(5 downto 0) :="010001";
        -- 192+32*2+16 =272
CONSTANT  fourth_column_x1:STD_LOGIC_VECTOR(9 downto 0):="0100100000";
        -- 192+ 32*3 = 288
CONSTANT  fourth_column_x1: STD_LOGIC_VECTOR(5 downto 0):="010010";
        -- 192+ 32*3 = 288
CONSTANT  fourth_column_x2: STD_LOGIC_VECTOR(9 downto 0):="0100110000";
        -- 192+ 32*3 +16 =304
CONSTANT  fourth_column_x2: STD_LOGIC_VECTOR(5 downto 0)  :="010011";
        -- 192+ 32*3 +16 =304
CONSTANT  fifth_column_x1 : STD_LOGIC_VECTOR(9 downto 0)
        :="0101000000"; -- 192+ 32*4 = 320
CONSTANT  fifth_column_x1 : STD_LOGIC_VECTOR(5 downto 0)  :="010100";
        -- 192+ 32*4 = 320
CONSTANT  fifth_column_x2       : STD_LOGIC_VECTOR(9 downto 0)
        :="0101010000"; -- 192+ 32*4 +16 =336
CONSTANT  fifth_column_x2       : STD_LOGIC_VECTOR(5 downto 0)
        :="010101"; -- 192+ 32*4 +16 =336
CONSTANT  sixth_column_x1       : STD_LOGIC_VECTOR(9 downto 0)
        :="0101100000"; -- 192+32*5 =352
CONSTANT  sixth_column_x1       : STD_LOGIC_VECTOR(5 downto 0)
        :="010110"; -- 192+32*5 =352

CONSTANT  sixth_column_x2       : STD_LOGIC_VECTOR(9 downto 0)
        :="0101110000"; -- 192+32*5+16 =368
CONSTANT  sixth_column_x2       : STD_LOGIC_VECTOR(5 downto 0)
        :="010111"; -- 192+32*5+16 =368

CONSTANT  seventh_column_x1     : STD_LOGIC_VECTOR(9 downto 0)
        :="0110000000"; -- 192+32*6 =384
CONSTANT  seventh_column_x1     : STD_LOGIC_VECTOR(5 downto 0)
        :="011000";     -- 192+32*6 =384
CONSTANT  seventh_column_x2     : STD_LOGIC_VECTOR(9 downto 0)
        :="0101110000"; -- 192+32*6+16 =400
CONSTANT  seventh_column_x2     : STD_LOGIC_VECTOR(5 downto 0)
        :="011001";  -- 192+32*5+16 =400

CONSTANT  eighth_column_x1      : STD_LOGIC_VECTOR(5 downto 0)
        :="011010";  -- 192+32*7 =416

CONSTANT  seventh_column_x2     : STD_LOGIC_VECTOR(9 downto 0)
        :="0101110000"; -- 192+32*7+16 =432
CONSTANT  eighth_column_x2      : STD_LOGIC_VECTOR(5 downto 0)
        :="011011"; -- 192+32*7+16 =432
```

```
        CONSTANT  raster_length     : STD_LOGIC_VECTOR(9 downto 0)
                :=raster_width;
        CONSTANT  all_zeros         : STD_LOGIC_VECTOR(9 downto 0)
                :="0000000000";
-------------------------------------------------------------------------------
end package dice_states;
LIBRARY IEEE;
USE ieee.std_logic_1164.all;
USE ieee.std_logic_arith.all;
USE ieee.std_logic_unsigned.all;

ENTITY SyncGen IS
        PORT
        (
                Clock                   : IN    STD_LOGIC;
                HSync, VSync            : OUT   BOOLEAN;
                vclock                  : OUT   STD_LOGIC;
                vga_h_sync, vga_v_sync  : OUT   BOOLEAN
        );
END SyncGen;

ARCHITECTURE ASyncGen OF SyncGen IS
        SIGNAL  CounterX, CounterY: INTEGER RANGE 0 TO 1023;
        SIGNAL  HS, VS, vga_HS, vga_VS: BOOLEAN;
BEGIN
        PROCESS (Clock)
        VARIABLE
                EnableCntY, ResetCntY: BOOLEAN;
        BEGIN
                EnableCntY := (CounterX=40);
                ResetCntY := (CounterY=524);

                IF Clock'EVENT AND Clock='1' THEN
                        IF CounterX=799 THEN
                                CounterX <= 0;
                        ELSE
                                CounterX <= CounterX + 1;
                        END IF;

                        IF EnableCntY THEN
                                IF ResetCntY THEN
                                        CounterY <= 0;
                                ELSE
                                        CounterY <= CounterY + 1;
                                END IF;
                        END IF;

                        vga_HS <= (CounterX<96);
                        vga_VS <= (CounterY< 2);
                        HS <= (CounterX=128);
                        VS <= (CounterY= 32) AND HS;
                END IF;
        END PROCESS;

        vga_h_sync <= vga_HS;
        vga_v_sync <= vga_VS;
        HSync <= HS;
        VSync <= VS;
        vclock<= '1' when VS;
END ASyncGen;
```