```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;

ENTITY vga_sync IS
        PORT (
                pixel_clk : IN STD_LOGIC;
                red_in    : IN STD_LOGIC;
                green_in  : IN STD_LOGIC;
                blue_in   : IN STD_LOGIC;
                red_out   : OUT STD_LOGIC;
                green_out : OUT STD_LOGIC;
                blue_out  : OUT STD_LOGIC;
                hsync     : OUT STD_LOGIC;
                vsync     : OUT STD_LOGIC;
                pixel_row : OUT STD_LOGIC_VECTOR (10 DOWNTO 0);
                pixel_col : OUT STD_LOGIC_VECTOR (10 DOWNTO 0)
        );
END vga_sync;

ARCHITECTURE Behavioral OF vga_sync IS
        SIGNAL h_cnt, v_cnt : STD_LOGIC_VECTOR (10 DOWNTO 0);

   CONSTANT H      : INTEGER := 800;
   CONSTANT V      : INTEGER := 600;
   CONSTANT H_FP   : INTEGER := 40;
   CONSTANT H_BP   : INTEGER := 88;
   CONSTANT H_SYNC : INTEGER := 128;
   CONSTANT V_FP   : INTEGER := 1;
   CONSTANT V_BP   : INTEGER := 23;
   CONSTANT V_SYNC : INTEGER := 4;

   CONSTANT FREQ   : INTEGER := 60;

BEGIN
        sync_pr : PROCESS
                VARIABLE video_on : STD_LOGIC;
        BEGIN
                WAIT UNTIL rising_edge(pixel_clk);
                -- Generate Horizontal Timing Signals for Video Signal
      -- total horizontal line width = H + H_FP + H_SYNC + H_BP
      -- Reset h_cnt when at end of line
                IF (h_cnt >= H + H_FP + H_SYNC + H_BP - 1) THEN
                        h_cnt <= (others => '0');
                ELSE
                        h_cnt <= h_cnt + 1;
                END IF;
      -- Pull down hsync after front porch
                IF (h_cnt >= H + H_FP) AND (h_cnt <= H + H_FP + H_SYNC) THEN
                        hsync <= '0';
                ELSE
                        hsync <= '1';
                END IF;

                IF (v_cnt >= V + V_FP + V_SYNC + V_BP - 1) AND (h_cnt = H + FREQ - 1)
THEN
                        v_cnt <= (others => '0');
                ELSIF (h_cnt = H + FREQ - 1) THEN
                        v_cnt <= v_cnt + 1;
                END IF;
                IF (v_cnt >= V + V_FP) AND (v_cnt <= V + V_FP + V_SYNC) THEN
                        vsync <= '0';
                ELSE
                        vsync <= '1';
                END IF;
                -- Generate Video Signals and Pixel Address
                IF (h_cnt < H) AND (v_cnt < V) THEN
                        video_on := '1';
                ELSE
                        video_on := '0';
                END IF;
                pixel_col <= h_cnt;
                pixel_row <= v_cnt;
                -- Register video to clock edge and suppress video during blanking and sync
periods
                red_out   <= red_in AND video_on;
                green_out <= green_in AND video_on;
                blue_out  <= blue_in AND video_on;
        END PROCESS;
END Behavioral;
```