

## Lampiran 1

### Listing program

```
-----  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;  
  
---- Uncomment the following library declaration if instantiating  
---- any Xilinx primitives in this code.  
--library UNISIM;  
--use UNISIM.VComponents.all;  
  
entity vga is  
    Port ( clk50_in,reset : in std_logic;  
          keypad : in std_logic_vector (26 downto 0);  
          Port_Red_out : out std_logic;  
          Port_Green_out : out std_logic;  
          port_Blue_out : out std_logic;  
          hs_out : out std_logic;  
          vs_out : out std_logic);  
  
end vga;  
  
architecture Behavioral of vga is  
    signal Clk25,f,Red_out,Green_out,Blue_out : std_logic;  
    signal code_ascii,ascii1,ascii2,ascii3,ascii4,ascii5,ascii6,ascii7,  
    ascii8,ascii9,ascii10,ascii11,ascii12 : std_logic_vector (7 downto 0);  
    signal Horizontal_Counter : std_logic_vector (9 downto 0);  
    signal Vertical_Counter : std_logic_vector (9 downto 0);  
    signal digit : std_logic_vector (3 downto 0);  
  
begin  
    Port_Red_out <= Red_out;  
    Port_Green_out <= Green_out;  
    port_Blue_out <= Blue_out;  
  
    --Generate 25Mhz Clock  
    process (clk50_in)  
    begin  
        if clk50_in'event and clk50_in='1' then  
            if (Clk25 = '0')then  
                Clk25 <= '1' after 2 ns;  
            else  
                Clk25 <= '0' after 2 ns;  
            end if;  
        end if;  
    end process;  
end process;
```

```

process (reset,keypad,code_ascii,Clk25)
begin
    if reset='1' then
        code_ascii <="00000000";
    elsif Clk25'event and Clk25 = '1' then
        case keypad is
            when "00000000000000000000000001" => code_ascii <=
"01000001";f<='1'; ----A--65
            when "00000000000000000000000010" => code_ascii <=
"01000010";f<='1'; ----B--66
            when "000000000000000000000000100" => code_ascii <=
"01000011";f<='1'; ----C--67
            when "0000000000000000000000001000" => code_ascii <=
"01000100";f<='1'; ----D--68
            when "00000000000000000000000010000" => code_ascii <=
"01000101";f<='1'; ----E--69
            when "000000000000000000000000100000" => code_ascii <=
"01000110";f<='1'; ----F--70
            when "0000000000000000000000001000000" => code_ascii <=
"01000111";f<='1'; ----G--71
            when "00000000000000000000000010000000" => code_ascii <=
"01001000";f<='1'; ----H--72
            when "000000000000000000000000100000000" => code_ascii <=
"01001001";f<='1'; ----I--73
            when "0000000000000000000000001000000000" => code_ascii <=
"01001010";f<='1'; ----J--74
            when "00000000000000000000000010000000000" => code_ascii <=
"01001011";f<='1'; ----K--75
            when "000000000000000000000000100000000000" => code_ascii <=
"01001100";f<='1'; ----L--76
            when "0000000000000000000000001000000000000" => code_ascii <=
"01001101";f<='1'; ----M--77
            when "00000000000000000000000010000000000000" => code_ascii <=
"01001110";f<='1'; ----N--78
            when "000000000000000000000000100000000000000" => code_ascii <=
"01001111";f<='1'; ----O--79
            when "0000000000000000000000001000000000000000" => code_ascii <=
"01010000";f<='1'; ----P--80
            when "0000000000000000000000001000000000000000" => code_ascii <=
"01010001";f<='1'; ----Q--81
            when "00000000000000000000000010000000000000000" => code_ascii <=
"01010010";f<='1'; ----R--82
            when "000000000000000000000000100000000000000000" => code_ascii <=
"01010011";f<='1'; ----S--83
            when "0000000000000000000000001000000000000000000" => code_ascii <=
"01010100";f<='1'; ----T--84
            when "00000000000000000000000010000000000000000000" => code_ascii <=
"01010101";f<='1'; ----U--85
            when "000000000000000000000000100000000000000000000" => code_ascii <=
"01010110";f<='1'; ----V--86
            when "0000000000000000000000001000000000000000000000" => code_ascii <=
"01010111";f<='1'; ----W--87
            when "00000000000000000000000010000000000000000000000" => code_ascii <=
"01011000";f<='1'; ----X--88

```

```

        when "001000000000000000000000" => code_ascii <=
"01011001";f<='1'; ----Y--89
        when "010000000000000000000000" => code_ascii <=
"01011010";f<='1'; ----Z--90
        when "100000000000000000000000" => code_ascii <=
"11111111";f<='1'; --SPACE--
        when others => code_ascii <= code_ascii; f<='0';
    end case;
end if;
end process;

```

```

process (f,digit,reset)
begin

```

```

    if reset = '1' then
        digit<="0000";
    elsif f='1' and f'event then
        if digit<11 then
            digit <= digit + 1;
        elsif digit=11 then
            digit <= digit + 0;
        else
            digit <=(others=>'0');
        end if;
    end if;
end process;

```

```

process(reset,ascii1,ascii2,ascii3,ascii4,ascii5,ascii6,Clk25)
begin

```

```

    if reset='1' then
        ascii1<="00000000";ascii2<="00000000";ascii3<="00000000";ascii4<="00000000";ascii5<
="00000000";ascii6<="00000000";ascii7<="00000000";ascii8<="00000000";ascii9<="00000000";as
cii10<="00000000";

```

```

    elsif Clk25='1' and Clk25'event then
        case digit is
            when "0001"=>ascii1<=code_ascii; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0010"=>ascii1<=ascii1; ascii2<=code_ascii; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0011"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=code_ascii; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0100"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=code_ascii;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0101"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=code_ascii; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0110"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=code_ascii; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "0111"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=code_ascii; ascii8<=ascii8; ascii9<=ascii9; ascii10<=ascii10;
            when "1000"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=code_ascii; ascii9<=ascii9; ascii10<=ascii10;
            when "1001"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=code_ascii; ascii10<=ascii10;

```

```

        when "1010"=>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
        ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9; ascii10<=code_ascii;
        when others =>ascii1<=ascii1; ascii2<=ascii2; ascii3<=ascii3; ascii4<=ascii4;
        ascii5<=ascii5; ascii6<=ascii6; ascii7<=ascii7; ascii8<=ascii8; ascii9<=ascii9;
        ascii10<=ascii10;
    end case;
end if;
end process;
process (Clk25)

```

```

TYPE huruf is ARRAY(0 to 11, 0 to 7) OF std_logic;
CONSTANT char_A : huruf :=
    (( '0','0','0','1','0','0','0','0'),
      ('0','0','1','1','1','0','0','0'),
      ('0','1','1','0','1','1','0','0'),
      ('1','1','0','0','0','1','1','0'),
      ('1','1','0','0','0','1','1','0'),
      ('1','1','1','1','1','1','1','0'),
      ('1','1','0','0','0','1','1','0'),
      ('1','1','0','0','0','1','1','0'),
      ('1','1','0','0','0','1','1','0'),
      ('1','1','0','0','0','1','1','0'),
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_B : huruf :=
    (( '1','1','1','1','1','1','0','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','1','1','1','0','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('1','1','1','1','1','1','0','0'),
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_C : huruf :=
    (( '0','0','1','1','1','1','0','0'),
      ('0','1','1','0','0','1','1','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('1','1','0','0','0','0','0','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','0','1','1','1','1','0','0'),
      ('0','0','0','0','0','0','0','0'),
      ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_D : huruf :=
    (( '1','1','1','1','1','1','0','0'),
      ('0','1','1','0','1','1','0','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'),
      ('0','1','1','0','0','1','1','0'));

```

```
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','1','1','0','0'),  
( '1','1','1','1','1','0','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));
```

```
CONSTANT char_E : huruf :=  
(( '1','1','1','1','1','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','0','1','0'),  
( '0','1','1','0','1','0','0','0'),  
( '0','1','1','1','1','0','0','0'),  
( '0','1','1','0','1','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '1','1','1','1','1','1','1','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));
```

```
CONSTANT char_F : huruf :=  
(( '1','1','1','1','1','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','0','1','0'),  
( '0','1','1','0','1','0','0','0'),  
( '0','1','1','1','1','0','0','0'),  
( '0','1','1','0','1','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '1','1','1','1','1','0','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));
```

```
CONSTANT char_G : huruf :=  
(( '0','0','1','1','1','1','0','0'),  
( '0','1','1','0','0','1','1','0'),  
( '1','1','0','0','0','0','1','0'),  
( '1','1','0','0','0','0','0','0'),  
( '1','1','0','0','0','0','0','0'),  
( '1','1','0','1','1','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','0','1','1','1','0','1','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));
```

```
CONSTANT char_H : huruf :=  
(( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','1','1','1','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),
```

( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_I : huruf : = (( '0','0','1','1','1','1','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','0','1','1','0','0','0'),  
( '0','0','1','1','1','1','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_J : huruf : = (( '0','0','0','1','1','1','0'),  
( '0','0','0','0','1','1','0','0'),  
( '0','0','0','0','1','1','0','0'),  
( '0','0','0','0','1','1','0','0'),  
( '0','0','0','0','1','1','0','0'),  
( '0','0','0','0','1','1','0','0'),  
( '1','1','0','0','1','1','0','0'),  
( '1','1','0','0','1','1','0','0'),  
( '1','1','0','0','1','1','0','0'),  
( '0','1','1','1','1','0','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_K : huruf : = (( '1','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','1','1','0','0'),  
( '0','1','1','1','1','0','0','0'),  
( '0','1','1','1','1','0','0','0'),  
( '0','1','1','0','1','1','0','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '1','1','1','0','0','1','1','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_L : huruf : = (( '1','1','1','1','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '1','1','1','1','1','1','1','0'));

('0','0','0','0','0','0','0','0'),  
('0','0','0','0','0','0','0','0'));

CONSTANT char\_M : huruf := (( '1','1','0','0','0','0','1','1'),  
( '1','1','1','0','0','1','1','1'),  
( '1','1','1','1','1','1','1','1'),  
( '1','1','1','1','1','1','1','1'),  
( '1','1','0','1','1','0','1','1'),  
( '1','1','0','0','0','0','1','1'),  
( '1','1','0','0','0','0','1','1'),  
( '1','1','0','0','0','0','1','1'),  
( '1','1','0','0','0','0','1','1'),  
( '1','1','0','0','0','0','1','1'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_N : huruf := (( '1','1','0','0','0','1','1','0'),  
( '1','1','1','0','0','1','1','0'),  
( '1','1','1','1','0','1','1','0'),  
( '1','1','1','1','1','1','1','0'),  
( '1','1','0','1','1','1','1','0'),  
( '1','1','0','0','1','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_O : huruf := (( '0','1','1','1','1','1','0','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '1','1','0','0','0','1','1','0'),  
( '0','1','1','1','1','1','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

CONSTANT char\_P : huruf := (( '1','1','1','1','1','1','0','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','0','0','1','1','0'),  
( '0','1','1','1','1','1','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '0','1','1','0','0','0','0','0'),  
( '1','1','1','1','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'),  
( '0','0','0','0','0','0','0','0'));

```

CONSTANT char_Q : huruf :=
  (('0','1','1','1','1','1','0','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','1','0','1','1','0'),
   ('1','1','0','1','1','1','1','0'),
   ('0','1','1','1','1','1','0','0'),
   ('0','0','0','0','1','1','0','0'),
   ('0','0','0','0','1','1','1','0'));

```

```

CONSTANT char_R : huruf :=
  (('1','1','1','1','1','1','0','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','1','1','1','0','0'),
   ('0','1','1','0','1','1','0','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','0','0','1','1','0'),
   ('0','1','1','0','0','1','1','0'),
   ('1','1','1','0','0','1','1','0'),
   ('0','0','0','0','0','0','0','0'),
   ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_S : huruf :=
  (('0','1','1','1','1','1','0','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('0','1','1','0','0','0','0','0'),
   ('0','0','1','1','1','0','0','0'),
   ('0','0','0','0','1','1','0','0'),
   ('0','0','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'),
   ('0','1','1','1','1','1','0','0'),
   ('0','0','0','0','0','0','0','0'),
   ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_T : huruf :=
  (('1','1','1','1','1','1','1','1'),
   ('1','1','0','1','1','0','1','1'),
   ('1','0','0','1','1','0','0','1'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','0','1','1','0','0','0'),
   ('0','0','1','1','1','1','0','0'),
   ('0','0','0','0','0','0','0','0'),
   ('0','0','0','0','0','0','0','0'));

```

```

CONSTANT char_U : huruf :=
  (('1','1','0','0','0','1','1','0'),
   ('1','1','0','0','0','1','1','0'));

```



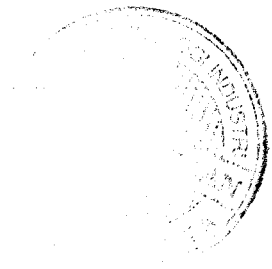
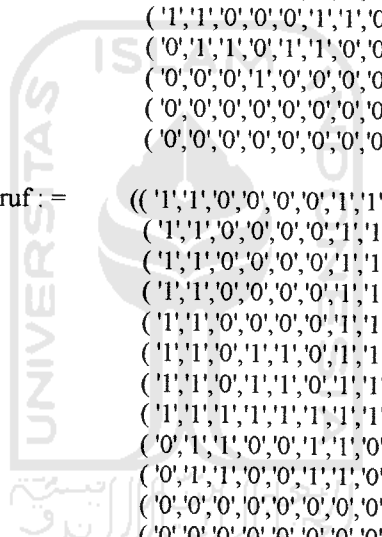
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('0','1','1','1','1','1','0','0'),  
('0','0','0','0','0','0','0','0'),  
('0','0','0','0','0','0','0','0'));

CONSTANT char\_V : huruf : = (('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('1','1','0','0','0','1','1','0'),  
('0','1','1','0','1','1','0','0'),  
('0','0','0','1','0','0','0','0'),  
('0','0','0','0','0','0','0','0'),  
('0','0','0','0','0','0','0','0'));

CONSTANT char\_W : huruf : = (('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','1','1','0','1','1'),  
('1','1','0','1','1','0','1','1'),  
('1','1','1','1','1','1','1','1'),  
('0','1','1','0','0','1','1','0'),  
('0','1','1','0','0','1','1','0'),  
('0','0','0','0','0','0','0','0'),  
('0','0','0','0','0','0','0','0'));

CONSTANT char\_X : huruf : = (('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('0','1','1','0','0','1','1','0'),  
('0','0','1','1','1','1','0','0'),  
('0','0','0','1','1','0','0','0'),  
('0','0','0','1','1','0','0','0'),  
('0','0','1','1','1','1','0','0'),  
('0','1','1','0','0','1','1','0'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('0','0','0','0','0','0','0','0'),  
('0','0','0','0','0','0','0','0'));

CONSTANT char\_Y : huruf : = (('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('1','1','0','0','0','0','1','1'),  
('0','1','1','1','1','1','0','0'),  
('0','0','1','1','1','1','0','0'));



```

( '0','0','0','1','1','0','0','0'),
( '0','0','0','1','1','0','0','0'),
( '0','0','0','1','1','0','0','0'),
( '0','0','0','1','1','0','0','0'),
( '0','0','1','1','1','1','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'));

CONSTANT char_Z : huruf : =
(( '1','1','1','1','1','1','1','1'),
( '1','1','0','0','0','0','1','1'),
( '1','0','0','0','0','1','1','0'),
( '0','0','0','0','1','1','0','0'),
( '0','0','0','1','1','0','0','0'),
( '0','0','1','1','0','0','0','0'),
( '0','1','1','0','0','0','0','0'),
( '1','1','1','0','0','0','0','1'),
( '1','1','0','0','0','0','1','1'),
( '1','1','1','1','1','1','1','1'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'));

CONSTANT char_Reset : huruf : =
(( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'),
( '0','0','0','0','0','0','0','0'));

variable Line:integer:=0;
variable Pixel:integer:=0;

begin

if Clk25'event and Clk25 = '1' then
  if (Horizontal_Counter >= "0010010000" ) -- 144
  and (Horizontal_Counter < "1100010000" ) -- 784
  and (Vertical_Counter >= "0000100001" ) -- 33
  and (Vertical_Counter < "1000000001" ) -- 513
  then
    Red_out <= '0'; ---- Buat text putih
    Green_out <= '0'; ---- Buat text putih
    Blue_out <= '0'; ---- Buat text putih
  if (Horizontal_Counter >= "0110101000" )--424
  and (Horizontal_Counter <= "0110101111")-- 431
  and (Vertical_Counter >= "0011101010" ) --234
  and (Vertical_Counter <= "0011110101" ) then -- 245
    if (Pixel <= 7) and (line <=11) then
      if (ascii1 <= "00000000") then
        Red_out <= char_Reset(Line, Pixel);

```

```

Green_out <= char_Reset(Line, Pixel);
Blue_out <= char_Reset(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000001")then
Red_out <= char_A(Line, Pixel);
Green_out <= char_A(Line, Pixel);
Blue_out <= char_A(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000010") then
Red_out <= char_B(Line, Pixel);
Green_out <= char_B(Line, Pixel);
Blue_out <= char_B(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000011") then
Red_out <= char_C(Line, Pixel);
Green_out <= char_C(Line, Pixel);
Blue_out <= char_C(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000100") then
Red_out <= char_D(Line, Pixel);
Green_out <= char_D(Line, Pixel);
Blue_out <= char_D(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000101") then
Red_out <= char_E(Line, Pixel);
Green_out <= char_E(Line, Pixel);
Blue_out <= char_E(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000110") then
Red_out <= char_F(Line, Pixel);
Green_out <= char_F(Line, Pixel);
Blue_out <= char_F(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01000111") then
Red_out <= char_G(Line, Pixel);
Green_out <= char_G(Line, Pixel);
Blue_out <= char_G(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01001000") then
Red_out <= char_H(Line, Pixel);
Green_out <= char_H(Line, Pixel);
Blue_out <= char_H(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01001001") then
Red_out <= char_I(Line, Pixel);
Green_out <= char_I(Line, Pixel);
Blue_out <= char_I(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01001010") then
Red_out <= char_J(Line, Pixel);
Green_out <= char_J(Line, Pixel);
Blue_out <= char_J(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii1 <= "01001011") then
Red_out <= char_K(Line, Pixel);

```

```

        Green_out <= char_K(Line, Pixel);
        Blue_out <= char_K(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01001100") then
        Red_out <= char_L(Line, Pixel);
        Green_out <= char_L(Line, Pixel);
        Blue_out <= char_L(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01001101") then
        Red_out <= char_M(Line, Pixel);
        Green_out <= char_M(Line, Pixel);
        Blue_out <= char_M(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01001110") then
        Red_out <= char_N(Line, Pixel);
        Green_out <= char_N(Line, Pixel);
        Blue_out <= char_N(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01001111") then
        Red_out <= char_O(Line, Pixel);
        Green_out <= char_O(Line, Pixel);
        Blue_out <= char_O(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010000") then
        Red_out <= char_P(Line, Pixel);
        Green_out <= char_P(Line, Pixel);
        Blue_out <= char_P(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010001") then
        Red_out <= char_Q(Line, Pixel);
        Green_out <= char_Q(Line, Pixel);
        Blue_out <= char_Q(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010010") then
        Red_out <= char_R(Line, Pixel);
        Green_out <= char_R(Line, Pixel);
        Blue_out <= char_R(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010011") then
        Red_out <= char_S(Line, Pixel);
        Green_out <= char_S(Line, Pixel);
        Blue_out <= char_S(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010100") then
        Red_out <= char_T(Line, Pixel);
        Green_out <= char_T(Line, Pixel);
        Blue_out <= char_T(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010101") then
        Red_out <= char_U(Line, Pixel);
        Green_out <= char_U(Line, Pixel);
        Blue_out <= char_U(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010110") then
        Red_out <= char_V(Line, Pixel);

```

```

        Green_out <= char_V(Line, Pixel);
        Blue_out <= char_V(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01010111") then
        Red_out <= char_W(Line, Pixel);
        Green_out <= char_W(Line, Pixel);
        Blue_out <= char_W(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01011000") then
        Red_out <= char_X(Line, Pixel);
        Green_out <= char_X(Line, Pixel);
        Blue_out <= char_X(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01011001") then
        Red_out <= char_Y(Line, Pixel);
        Green_out <= char_Y(Line, Pixel);
        Blue_out <= char_Y(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "01011010") then
        Red_out <= char_Z(Line, Pixel);
        Green_out <= char_Z(Line, Pixel);
        Blue_out <= char_Z(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii1 <= "11111111") then
        Red_out <= char_Reset(Line, Pixel);
        Green_out <= char_Reset(Line, Pixel);
        Blue_out <= char_Reset(Line, Pixel);
        Pixel:= Pixel+1;
    end if;
end if;
----digit 2
elsif (Horizontal_Counter >= "0110110000")--432
and (Horizontal_Counter <= "0110110111")-- 439
and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011101011") then -- 245
    if (Pixel <= 15) and (line <=11) then
        if (ascii2 <= "00000000") then
            Red_out <= char_Reset(Line, Pixel);
            Green_out <= char_Reset(Line, Pixel);
            Blue_out <= char_Reset(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii2 <= "01000001")then
            Red_out <= char_A(Line, Pixel);
            Green_out <= char_A(Line, Pixel);
            Blue_out <= char_A(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii2 <= "01000010") then
            Red_out <= char_B(Line, Pixel);
            Green_out <= char_B(Line, Pixel);
            Blue_out <= char_B(Line, Pixel);
            Pixel:= Pixel+1;
        elsif
            (ascii2 <= "01000011") then
            Red_out <= char_C(Line, Pixel);
            Green_out <= char_C(Line, Pixel);
            Blue_out <= char_C(Line, Pixel);
        end if;
    end if;
end if;

```

```

Pixel:= Pixel+1;
elsif (ascii2 <= "01000100") then
  Red_out <= char_D(Line, Pixel);
  Green_out <= char_D(Line, Pixel);
  Blue_out <= char_D(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01000101") then
  Red_out <= char_E(Line, Pixel);
  Green_out <= char_E(Line, Pixel);
  Blue_out <= char_E(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01000110") then
  Red_out <= char_F(Line, Pixel);
  Green_out <= char_F(Line, Pixel);
  Blue_out <= char_F(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01000111") then
  Red_out <= char_G(Line, Pixel);
  Green_out <= char_G(Line, Pixel);
  Blue_out <= char_G(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001000") then
  Red_out <= char_H(Line, Pixel);
  Green_out <= char_H(Line, Pixel);
  Blue_out <= char_H(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001001") then
  Red_out <= char_I(Line, Pixel);
  Green_out <= char_I(Line, Pixel);
  Blue_out <= char_I(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001010") then
  Red_out <= char_J(Line, Pixel);
  Green_out <= char_J(Line, Pixel);
  Blue_out <= char_J(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001011") then
  Red_out <= char_K(Line, Pixel);
  Green_out <= char_K(Line, Pixel);
  Blue_out <= char_K(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001100") then
  Red_out <= char_L(Line, Pixel);
  Green_out <= char_L(Line, Pixel);
  Blue_out <= char_L(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii2 <= "01001101") then
  Red_out <= char_M(Line, Pixel);
  Green_out <= char_M(Line, Pixel);
  Blue_out <= char_M(Line, Pixel);
  Pixel:= Pixel+1;
elsif(ascii2 <= "01001110") then
  Red_out <= char_N(Line, Pixel);
  Green_out <= char_N(Line, Pixel);
  Blue_out <= char_N(Line, Pixel);

```

```

Pixel:= Pixel+1;
elsif(ascii2 <= "01001111") then
    Red_out <= char_O(Line, Pixel);
    Green_out <= char_O(Line, Pixel);
    Blue_out <= char_O(Line, Pixel);
    Pixel:= Pixel+1;
elsif(ascii2 <= "01010000") then
    Red_out <= char_P(Line, Pixel);
    Green_out <= char_P(Line, Pixel);
    Blue_out <= char_P(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010001") then
    Red_out <= char_Q(Line, Pixel);
    Green_out <= char_Q(Line, Pixel);
    Blue_out <= char_Q(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010010") then
    Red_out <= char_R(Line, Pixel);
    Green_out <= char_R(Line, Pixel);
    Blue_out <= char_R(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010011") then
    Red_out <= char_S(Line, Pixel);
    Green_out <= char_S(Line, Pixel);
    Blue_out <= char_S(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010100") then
    Red_out <= char_T(Line, Pixel);
    Green_out <= char_T(Line, Pixel);
    Blue_out <= char_T(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010101") then
    Red_out <= char_U(Line, Pixel);
    Green_out <= char_U(Line, Pixel);
    Blue_out <= char_U(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010110") then
    Red_out <= char_V(Line, Pixel);
    Green_out <= char_V(Line, Pixel);
    Blue_out <= char_V(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01010111") then
    Red_out <= char_W(Line, Pixel);
    Green_out <= char_W(Line, Pixel);
    Blue_out <= char_W(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01011000") then
    Red_out <= char_X(Line, Pixel);
    Green_out <= char_X(Line, Pixel);
    Blue_out <= char_X(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii2 <= "01011001") then
    Red_out <= char_Y(Line, Pixel);
    Green_out <= char_Y(Line, Pixel);
    Blue_out <= char_Y(Line, Pixel);

```

```

        Pixel:= Pixel+1;
    elsif (ascii2 <= "01011010") then
        Red_out <= char_Z(Line, Pixel);
        Green_out <= char_Z(Line, Pixel);
        Blue_out <= char_Z(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii2 <= "11111111") then
        Red_out <= char_Reset(Line, Pixel);
        Green_out <= char_Reset(Line, Pixel);
        Blue_out <= char_Reset(Line, Pixel);
        Pixel:= Pixel+1;
    end if;
end if;
----digit3
elsif (Horizontal_Counter >= "0110111000" )--440
and (Horizontal_Counter <= "0110111111")-- 447
and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011110101") then -- 245
    if (Pixel <= 23) and (line <=11) then
        if (ascii3 <= "00000000") then
            Red_out <= char_Reset(Line, Pixel);
            Green_out <= char_Reset(Line, Pixel);
            Blue_out <= char_Reset(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii3 <= "01000001")then
            Red_out <= char_A(Line, Pixel);
            Green_out <= char_A(Line, Pixel);
            Blue_out <= char_A(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii3 <= "01000010") then
            Red_out <= char_B(Line, Pixel);
            Green_out <= char_B(Line, Pixel);
            Blue_out <= char_B(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii3 <= "01000011") then
            Red_out <= char_C(Line, Pixel);
            Green_out <= char_C(Line, Pixel);
            Blue_out <= char_C(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii3 <= "01000100") then
            Red_out <= char_D(Line, Pixel);
            Green_out <= char_D(Line, Pixel);
            Blue_out <= char_D(Line, Pixel);
            Pixel:= Pixel+1;
        elsif(ascii3 <= "01000101") then
            Red_out <= char_E(Line, Pixel);
            Green_out <= char_E(Line, Pixel);
            Blue_out <= char_E(Line, Pixel);
            Pixel:= Pixel+1;
        elsif(ascii3 <= "01000110") then
            Red_out <= char_F(Line, Pixel);
            Green_out <= char_F(Line, Pixel);
            Blue_out <= char_F(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii3 <= "01000111") then

```



```

Red_out <= char_G(Line, Pixel);
Green_out <= char_G(Line, Pixel);
Blue_out <= char_G(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001000") then
Red_out <= char_H(Line, Pixel);
Green_out <= char_H(Line, Pixel);
Blue_out <= char_H(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001001") then
Red_out <= char_I(Line, Pixel);
Green_out <= char_I(Line, Pixel);
Blue_out <= char_I(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001010") then
Red_out <= char_J(Line, Pixel);
Green_out <= char_J(Line, Pixel);
Blue_out <= char_J(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001011") then
Red_out <= char_K(Line, Pixel);
Green_out <= char_K(Line, Pixel);
Blue_out <= char_K(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001100") then
Red_out <= char_L(Line, Pixel);
Green_out <= char_L(Line, Pixel);
Blue_out <= char_L(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001101") then
Red_out <= char_M(Line, Pixel);
Green_out <= char_M(Line, Pixel);
Blue_out <= char_M(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001110") then
Red_out <= char_N(Line, Pixel);
Green_out <= char_N(Line, Pixel);
Blue_out <= char_N(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01001111") then
Red_out <= char_O(Line, Pixel);
Green_out <= char_O(Line, Pixel);
Blue_out <= char_O(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01010000") then
Red_out <= char_P(Line, Pixel);
Green_out <= char_P(Line, Pixel);
Blue_out <= char_P(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii3 <= "01010001") then
Red_out <= char_Q(Line, Pixel);
Green_out <= char_Q(Line, Pixel);
Blue_out <= char_Q(Line, Pixel);
Pixel:= Pixel+1;

```

```

elseif (ascii3 <= "01010010") then
    Red_out <= char_R(Line, Pixel);
    Green_out <= char_R(Line, Pixel);
    Blue_out <= char_R(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01010011") then
    Red_out <= char_S(Line, Pixel);
    Green_out <= char_S(Line, Pixel);
    Blue_out <= char_S(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01010100") then
    Red_out <= char_T(Line, Pixel);
    Green_out <= char_T(Line, Pixel);
    Blue_out <= char_T(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01010101") then
    Red_out <= char_U(Line, Pixel);
    Green_out <= char_U(Line, Pixel);
    Blue_out <= char_U(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01010110") then
    Red_out <= char_V(Line, Pixel);
    Green_out <= char_V(Line, Pixel);
    Blue_out <= char_V(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01010111") then
    Red_out <= char_W(Line, Pixel);
    Green_out <= char_W(Line, Pixel);
    Blue_out <= char_W(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01011000") then
    Red_out <= char_X(Line, Pixel);
    Green_out <= char_X(Line, Pixel);
    Blue_out <= char_X(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01011001") then
    Red_out <= char_Y(Line, Pixel);
    Green_out <= char_Y(Line, Pixel);
    Blue_out <= char_Y(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "01011010") then
    Red_out <= char_Z(Line, Pixel);
    Green_out <= char_Z(Line, Pixel);
    Blue_out <= char_Z(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii3 <= "11111111") then
    Red_out <= char_Reset(Line, Pixel);
    Green_out <= char_Reset(Line, Pixel);
    Blue_out <= char_Reset(Line, Pixel);
    Pixel:= Pixel+1;
end if;
end if;
----digit 4
elseif (Horizontal_Counter >= "0111000000")--448
and (Horizontal_Counter <= "0111000111")--

```

```

and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011110101") then -- 245
  if (Pixel <= 31) and (line <=11) then
    if (ascii4 <= "00000000") then
      Red_out <= char_Reset(Line, Pixel);
      Green_out <= char_Reset(Line, Pixel);
      Blue_out <= char_Reset(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000001")then
      Red_out <= char_A(Line, Pixel);
      Green_out <= char_A(Line, Pixel);
      Blue_out <= char_A(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000010") then
      Red_out <= char_B(Line, Pixel);
      Green_out <= char_B(Line, Pixel);
      Blue_out <= char_B(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000011") then
      Red_out <= char_C(Line, Pixel);
      Green_out <= char_C(Line, Pixel);
      Blue_out <= char_C(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000100") then
      Red_out <= char_D(Line, Pixel);
      Green_out <= char_D(Line, Pixel);
      Blue_out <= char_D(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000101") then
      Red_out <= char_E(Line, Pixel);
      Green_out <= char_E(Line, Pixel);
      Blue_out <= char_E(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000110") then
      Red_out <= char_F(Line, Pixel);
      Green_out <= char_F(Line, Pixel);
      Blue_out <= char_F(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01000111") then
      Red_out <= char_G(Line, Pixel);
      Green_out <= char_G(Line, Pixel);
      Blue_out <= char_G(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01001000") then
      Red_out <= char_H(Line, Pixel);
      Green_out <= char_H(Line, Pixel);
      Blue_out <= char_H(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01001001") then
      Red_out <= char_I(Line, Pixel);
      Green_out <= char_I(Line, Pixel);
      Blue_out <= char_I(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii4 <= "01001010") then
      Red_out <= char_J(Line, Pixel);

```

```

        Green_out <= char_J(Line, Pixel);
        Blue_out <= char_J(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01001011") then
        Red_out <= char_K(Line, Pixel);
        Green_out <= char_K(Line, Pixel);
        Blue_out <= char_K(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01001100") then
        Red_out <= char_L(Line, Pixel);
        Green_out <= char_L(Line, Pixel);
        Blue_out <= char_L(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01001101") then
        Red_out <= char_M(Line, Pixel);
        Green_out <= char_M(Line, Pixel);
        Blue_out <= char_M(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01001110") then
        Red_out <= char_N(Line, Pixel);
        Green_out <= char_N(Line, Pixel);
        Blue_out <= char_N(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01001111") then
        Red_out <= char_O(Line, Pixel);
        Green_out <= char_O(Line, Pixel);
        Blue_out <= char_O(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010000") then
        Red_out <= char_P(Line, Pixel);
        Green_out <= char_P(Line, Pixel);
        Blue_out <= char_P(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010001") then
        Red_out <= char_Q(Line, Pixel);
        Green_out <= char_Q(Line, Pixel);
        Blue_out <= char_Q(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010010") then
        Red_out <= char_R(Line, Pixel);
        Green_out <= char_R(Line, Pixel);
        Blue_out <= char_R(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010011") then
        Red_out <= char_S(Line, Pixel);
        Green_out <= char_S(Line, Pixel);
        Blue_out <= char_S(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010100") then
        Red_out <= char_T(Line, Pixel);
        Green_out <= char_T(Line, Pixel);
        Blue_out <= char_T(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010101") then
        Red_out <= char_U(Line, Pixel);

```

```

        Green_out <= char_U(Line, Pixel);
        Blue_out <= char_Ū(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010110") then
        Red_out <= char_V(Line, Pixel);
        Green_out <= char_V(Line, Pixel);
        Blue_out <= char_V(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01010111") then
        Red_out <= char_W(Line, Pixel);
        Green_out <= char_W(Line, Pixel);
        Blue_out <= char_W̄(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01011000") then
        Red_out <= char_X(Line, Pixel);
        Green_out <= char_X(Line, Pixel);
        Blue_out <= char_X(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01011001") then
        Red_out <= char_Y(Line, Pixel);
        Green_out <= char_Y(Line, Pixel);
        Blue_out <= char_Ȳ(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "01011010") then
        Red_out <= char_Z(Line, Pixel);
        Green_out <= char_Z(Line, Pixel);
        Blue_out <= char_Z̄(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii4 <= "11111111") then
        Red_out <= char_Reset(Line, Pixel);
        Green_out <= char_Reset(Line, Pixel);
        Blue_out <= char_Reset(Line, Pixel);
        Pixel:= Pixel+1;
    end if;
end if;
----digit 5
elsif (Horizontal_Counter >= "0111001000")--456
and (Horizontal_Counter <= "0111001111")-- 463
and (Vertical_Counter >= "0011101010")--234
and (Vertical_Counter <= "0011110101") then -- 245
    if (Pixel <= 39) and (line <=11) then
        if (ascii5 <= "00000000") then
            Red_out <= char_Reset(Line, Pixel);
            Green_out <= char_Reset(Line, Pixel);
            Blue_out <= char_Reset(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii5 <= "01000001")then
            Red_out <= char_A(Line, Pixel);
            Green_out <= char_A(Line, Pixel);
            Blue_out <= char_A(Line, Pixel);
            Pixel:= Pixel+1;
        elsif
            (ascii5 <= "01000010") then
            Red_out <= char_B(Line, Pixel);
            Green_out <= char_B(Line, Pixel);
            Blue_out <= char_B(Line, Pixel);

```

```

Pixel:= Pixel+1;
elsif (ascii5 <= "01000011") then
  Red_out <= char_C(Line, Pixel);
  Green_out <= char_C(Line, Pixel);
  Blue_out <= char_C(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01000100") then
  Red_out <= char_D(Line, Pixel);
  Green_out <= char_D(Line, Pixel);
  Blue_out <= char_D(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01000101") then
  Red_out <= char_E(Line, Pixel);
  Green_out <= char_E(Line, Pixel);
  Blue_out <= char_E(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01000110") then
  Red_out <= char_F(Line, Pixel);
  Green_out <= char_F(Line, Pixel);
  Blue_out <= char_F(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01000111") then
  Red_out <= char_G(Line, Pixel);
  Green_out <= char_G(Line, Pixel);
  Blue_out <= char_G(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001000") then
  Red_out <= char_H(Line, Pixel);
  Green_out <= char_H(Line, Pixel);
  Blue_out <= char_H(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001001") then
  Red_out <= char_I(Line, Pixel);
  Green_out <= char_I(Line, Pixel);
  Blue_out <= char_I(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001010") then
  Red_out <= char_J(Line, Pixel);
  Green_out <= char_J(Line, Pixel);
  Blue_out <= char_J(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001011") then
  Red_out <= char_K(Line, Pixel);
  Green_out <= char_K(Line, Pixel);
  Blue_out <= char_K(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001100") then
  Red_out <= char_L(Line, Pixel);
  Green_out <= char_L(Line, Pixel);
  Blue_out <= char_L(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001101") then
  Red_out <= char_M(Line, Pixel);
  Green_out <= char_M(Line, Pixel);
  Blue_out <= char_M(Line, Pixel);

```

```

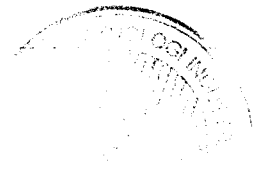
Pixel:= Pixel+1;
elsif (ascii5 <= "01001110") then
  Red_out <= char_N(Line, Pixel);
  Green_out <= char_N(Line, Pixel);
  Blue_out <= char_N(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01001111") then
  Red_out <= char_O(Line, Pixel);
  Green_out <= char_O(Line, Pixel);
  Blue_out <= char_O(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010000") then
  Red_out <= char_P(Line, Pixel);
  Green_out <= char_P(Line, Pixel);
  Blue_out <= char_P(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010001") then
  Red_out <= char_Q(Line, Pixel);
  Green_out <= char_Q(Line, Pixel);
  Blue_out <= char_Q(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010010") then
  Red_out <= char_R(Line, Pixel);
  Green_out <= char_R(Line, Pixel);
  Blue_out <= char_R(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010011") then
  Red_out <= char_S(Line, Pixel);
  Green_out <= char_S(Line, Pixel);
  Blue_out <= char_S(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010100") then
  Red_out <= char_T(Line, Pixel);
  Green_out <= char_T(Line, Pixel);
  Blue_out <= char_T(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010101") then
  Red_out <= char_U(Line, Pixel);
  Green_out <= char_U(Line, Pixel);
  Blue_out <= char_U(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010110") then
  Red_out <= char_V(Line, Pixel);
  Green_out <= char_V(Line, Pixel);
  Blue_out <= char_V(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01010111") then
  Red_out <= char_W(Line, Pixel);
  Green_out <= char_W(Line, Pixel);
  Blue_out <= char_W(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii5 <= "01011000") then
  Red_out <= char_X(Line, Pixel);
  Green_out <= char_X(Line, Pixel);
  Blue_out <= char_X(Line, Pixel);

```

```

Pixel:= Pixel+1;
elsif (ascii5 <= "01011001") then
    Red_out <= char_Y(Line, Pixel);
    Green_out <= char_Y(Line, Pixel);
    Blue_out <= char_Y(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii5 <= "01011010") then
    Red_out <= char_Z(Line, Pixel);
    Green_out <= char_Z(Line, Pixel);
    Blue_out <= char_Z(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii5 <= "11111111") then
    Red_out <= char_Reset(Line, Pixel);
    Green_out <= char_Reset(Line, Pixel);
    Blue_out <= char_Reset(Line, Pixel);
    Pixel:= Pixel+1;
end if;
end if;
----digit 6
elsif (Horizontal_Counter >= "0111010000") --464
and (Horizontal_Counter <= "0111010111")-- 471
and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011110101") then -- 245
    if (Pixel <= 47) and (line <=11) then
        if (ascii6 <= "00000000") then
            Red_out <= char_Reset(Line, Pixel);
            Green_out <= char_Reset(Line, Pixel);
            Blue_out <= char_Reset(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii6 <= "01000001")then
            Red_out <= char_A(Line, Pixel);
            Green_out <= char_A(Line, Pixel);
            Blue_out <= char_A(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii6 <= "01000010") then
            Red_out <= char_B(Line, Pixel);
            Green_out <= char_B(Line, Pixel);
            Blue_out <= char_B(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii6 <= "01000011") then
            Red_out <= char_C(Line, Pixel);
            Green_out <= char_C(Line, Pixel);
            Blue_out <= char_C(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii6 <= "01000100") then
            Red_out <= char_D(Line, Pixel);
            Green_out <= char_D(Line, Pixel);
            Blue_out <= char_D(Line, Pixel);
            Pixel:= Pixel+1;
        elsif(ascii6 <= "01000101") then
            Red_out <= char_E(Line, Pixel);
            Green_out <= char_E(Line, Pixel);
            Blue_out <= char_E(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii6 <= "01000110") then

```





```

Red_out <= char_F(Line, Pixel);
Green_out <= char_F(Line, Pixel);
Blue_out <= char_F(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01000111") then
Red_out <= char_G(Line, Pixel);
Green_out <= char_G(Line, Pixel);
Blue_out <= char_G(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001000") then
Red_out <= char_H(Line, Pixel);
Green_out <= char_H(Line, Pixel);
Blue_out <= char_H(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001001") then
Red_out <= char_I(Line, Pixel);
Green_out <= char_I(Line, Pixel);
Blue_out <= char_I(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001010") then
Red_out <= char_J(Line, Pixel);
Green_out <= char_J(Line, Pixel);
Blue_out <= char_J(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001011") then
Red_out <= char_K(Line, Pixel);
Green_out <= char_K(Line, Pixel);
Blue_out <= char_K(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001100") then
Red_out <= char_L(Line, Pixel);
Green_out <= char_L(Line, Pixel);
Blue_out <= char_L(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001101") then
Red_out <= char_M(Line, Pixel);
Green_out <= char_M(Line, Pixel);
Blue_out <= char_M(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001110") then
Red_out <= char_N(Line, Pixel);
Green_out <= char_N(Line, Pixel);
Blue_out <= char_N(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01001111") then
Red_out <= char_O(Line, Pixel);
Green_out <= char_O(Line, Pixel);
Blue_out <= char_O(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii6 <= "01010000") then
Red_out <= char_P(Line, Pixel);
Green_out <= char_P(Line, Pixel);
Blue_out <= char_P(Line, Pixel);
Pixel:= Pixel+1;

```

```

elsif (ascii6 <= "01010001") then
    Red_out <= char_Q(Line, Pixel);
    Green_out <= char_Q(Line, Pixel);
    Blue_out <= char_Q(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010010") then
    Red_out <= char_R(Line, Pixel);
    Green_out <= char_R(Line, Pixel);
    Blue_out <= char_R(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010011") then
    Red_out <= char_S(Line, Pixel);
    Green_out <= char_S(Line, Pixel);
    Blue_out <= char_S(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010100") then
    Red_out <= char_T(Line, Pixel);
    Green_out <= char_T(Line, Pixel);
    Blue_out <= char_T(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010101") then
    Red_out <= char_U(Line, Pixel);
    Green_out <= char_U(Line, Pixel);
    Blue_out <= char_U(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010110") then
    Red_out <= char_V(Line, Pixel);
    Green_out <= char_V(Line, Pixel);
    Blue_out <= char_V(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01010111") then
    Red_out <= char_W(Line, Pixel);
    Green_out <= char_W(Line, Pixel);
    Blue_out <= char_W(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01011000") then
    Red_out <= char_X(Line, Pixel);
    Green_out <= char_X(Line, Pixel);
    Blue_out <= char_X(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01011001") then
    Red_out <= char_Y(Line, Pixel);
    Green_out <= char_Y(Line, Pixel);
    Blue_out <= char_Y(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "01011010") then
    Red_out <= char_Z(Line, Pixel);
    Green_out <= char_Z(Line, Pixel);
    Blue_out <= char_Z(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii6 <= "11111111") then
    Red_out <= char_Reset(Line, Pixel);
    Green_out <= char_Reset(Line, Pixel);
    Blue_out <= char_Reset(Line, Pixel);

```

```

Pixel:= Pixel+1;
end if;
end if;
----- digit7
elsif (Horizontal_Counter >= "0111011000" )--472
and (Horizontal_Counter <= "0111011111")-- 479
and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011110101") then -- 245
if (Pixel <= 55) then
if (ascii7 <= "00000000") then
Red_out <= char_Reset(Line, Pixel);
Green_out <= char_Reset(Line, Pixel);
Blue_out <= char_Reset(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000001")then
Red_out <= char_A(Line, Pixel);
Green_out <= char_A(Line, Pixel);
Blue_out <= char_A(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000010") then
Red_out <= char_B(Line, Pixel);
Green_out <= char_B(Line, Pixel);
Blue_out <= char_B(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000011") then
Red_out <= char_C(Line, Pixel);
Green_out <= char_C(Line, Pixel);
Blue_out <= char_C(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000100") then
Red_out <= char_D(Line, Pixel);
Green_out <= char_D(Line, Pixel);
Blue_out <= char_D(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000101") then
Red_out <= char_E(Line, Pixel);
Green_out <= char_E(Line, Pixel);
Blue_out <= char_E(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000110") then
Red_out <= char_F(Line, Pixel);
Green_out <= char_F(Line, Pixel);
Blue_out <= char_F(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01000111") then
Red_out <= char_G(Line, Pixel);
Green_out <= char_G(Line, Pixel);
Blue_out <= char_G(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001000") then
Red_out <= char_H(Line, Pixel);
Green_out <= char_H(Line, Pixel);
Blue_out <= char_H(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001001") then

```

```

Red_out <= char_I(Line, Pixel);
Green_out <= char_I(Line, Pixel);
Blue_out <= char_I(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001010") then
Red_out <= char_J(Line, Pixel);
Green_out <= char_J(Line, Pixel);
Blue_out <= char_J(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001011") then
Red_out <= char_K(Line, Pixel);
Green_out <= char_K(Line, Pixel);
Blue_out <= char_K(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001100") then
Red_out <= char_L(Line, Pixel);
Green_out <= char_L(Line, Pixel);
Blue_out <= char_L(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001101") then
Red_out <= char_M(Line, Pixel);
Green_out <= char_M(Line, Pixel);
Blue_out <= char_M(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001110") then
Red_out <= char_N(Line, Pixel);
Green_out <= char_N(Line, Pixel);
Blue_out <= char_N(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01001111") then
Red_out <= char_O(Line, Pixel);
Green_out <= char_O(Line, Pixel);
Blue_out <= char_O(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01010000") then
Red_out <= char_P(Line, Pixel);
Green_out <= char_P(Line, Pixel);
Blue_out <= char_P(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01010001") then
Red_out <= char_Q(Line, Pixel);
Green_out <= char_Q(Line, Pixel);
Blue_out <= char_Q(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01010010") then
Red_out <= char_R(Line, Pixel);
Green_out <= char_R(Line, Pixel);
Blue_out <= char_R(Line, Pixel);
Pixel:= Pixel+1;
elsif (ascii7 <= "01010011") then
Red_out <= char_S(Line, Pixel);
Green_out <= char_S(Line, Pixel);
Blue_out <= char_S(Line, Pixel);
Pixel:= Pixel+1;

```

```

elseif (ascii7 <= "01010100") then
    Red_out <= char_T(Line, Pixel);
    Green_out <= char_T(Line, Pixel);
    Blue_out <= char_T(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01010101") then
    Red_out <= char_U(Line, Pixel);
    Green_out <= char_U(Line, Pixel);
    Blue_out <= char_U(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01010110") then
    Red_out <= char_V(Line, Pixel);
    Green_out <= char_V(Line, Pixel);
    Blue_out <= char_V(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01010111") then
    Red_out <= char_W(Line, Pixel);
    Green_out <= char_W(Line, Pixel);
    Blue_out <= char_W(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01011000") then
    Red_out <= char_X(Line, Pixel);
    Green_out <= char_X(Line, Pixel);
    Blue_out <= char_X(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01011001") then
    Red_out <= char_Y(Line, Pixel);
    Green_out <= char_Y(Line, Pixel);
    Blue_out <= char_Y(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "01011010") then
    Red_out <= char_Z(Line, Pixel);
    Green_out <= char_Z(Line, Pixel);
    Blue_out <= char_Z(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii7 <= "11111111")
    Red_out <= char_Reset(Line, Pixel);
    Green_out <= char_Reset(Line, Pixel);
    Blue_out <= char_Reset(Line, Pixel);
    Pixel:= Pixel+1;
end if;
end if;
----digit 8
elseif (Horizontal_Counter >= "0111100000")--480
and (Horizontal_Counter <= "0111100111")-- 487
and (Vertical_Counter >= "0011101010") --234
and (Vertical_Counter <= "0011110101") then -- 245
if (Pixel <= 63) then
if (ascii8 <= "00000000") then
    Red_out <= char_Reset(Line, Pixel);
    Green_out <= char_Reset(Line, Pixel);
    Blue_out <= char_Reset(Line, Pixel);
    Pixel:= Pixel+1;
elseif (ascii8 <= "01000001")
    Red_out <= char_A(Line, Pixel);

```

```

        Green_out <= char_A(Line, Pixel);
        Blue_out <= char_A(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000010") then
        Red_out <= char_B(Line, Pixel);
        Green_out <= char_B(Line, Pixel);
        Blue_out <= char_B(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000011") then
        Red_out <= char_C(Line, Pixel);
        Green_out <= char_C(Line, Pixel);
        Blue_out <= char_C(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000100") then
        Red_out <= char_D(Line, Pixel);
        Green_out <= char_D(Line, Pixel);
        Blue_out <= char_D(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000101") then
        Red_out <= char_E(Line, Pixel);
        Green_out <= char_E(Line, Pixel);
        Blue_out <= char_E(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000110") then
        Red_out <= char_F(Line, Pixel);
        Green_out <= char_F(Line, Pixel);
        Blue_out <= char_F(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01000111") then
        Red_out <= char_G(Line, Pixel);
        Green_out <= char_G(Line, Pixel);
        Blue_out <= char_G(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01001000") then
        Red_out <= char_H(Line, Pixel);
        Green_out <= char_H(Line, Pixel);
        Blue_out <= char_H(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01001001") then
        Red_out <= char_I(Line, Pixel);
        Green_out <= char_I(Line, Pixel);
        Blue_out <= char_I(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01001010") then
        Red_out <= char_J(Line, Pixel);
        Green_out <= char_J(Line, Pixel);
        Blue_out <= char_J(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01001011") then
        Red_out <= char_K(Line, Pixel);
        Green_out <= char_K(Line, Pixel);
        Blue_out <= char_K(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01001100") then
        Red_out <= char_L(Line, Pixel);

```

```

        Green_out <= char_L(Line, Pixel);
        Blue_out <= char_L(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01001101") then
        Red_out <= char_M(Line, Pixel);
        Green_out <= char_M(Line, Pixel);
        Blue_out <= char_M(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01001110") then
        Red_out <= char_N(Line, Pixel);
        Green_out <= char_N(Line, Pixel);
        Blue_out <= char_N(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01001111") then
        Red_out <= char_O(Line, Pixel);
        Green_out <= char_O(Line, Pixel);
        Blue_out <= char_O(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010000") then
        Red_out <= char_P(Line, Pixel);
        Green_out <= char_P(Line, Pixel);
        Blue_out <= char_P(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010001") then
        Red_out <= char_Q(Line, Pixel);
        Green_out <= char_Q(Line, Pixel);
        Blue_out <= char_Q(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010010") then
        Red_out <= char_R(Line, Pixel);
        Green_out <= char_R(Line, Pixel);
        Blue_out <= char_R(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010011") then
        Red_out <= char_S(Line, Pixel);
        Green_out <= char_S(Line, Pixel);
        Blue_out <= char_S(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010100") then
        Red_out <= char_T(Line, Pixel);
        Green_out <= char_T(Line, Pixel);
        Blue_out <= char_T(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010101") then
        Red_out <= char_U(Line, Pixel);
        Green_out <= char_U(Line, Pixel);
        Blue_out <= char_U(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010110") then
        Red_out <= char_V(Line, Pixel);
        Green_out <= char_V(Line, Pixel);
        Blue_out <= char_V(Line, Pixel);
        Pixel:= Pixel+1;
elseif (ascii8 <= "01010111") then
        Red_out <= char_W(Line, Pixel);

```

```

        Green_out <= char_W(Line, Pixel);
        Blue_out <= char_W(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01011000") then
        Red_out <= char_X(Line, Pixel);
        Green_out <= char_X(Line, Pixel);
        Blue_out <= char_X(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01011001") then
        Red_out <= char_Y(Line, Pixel);
        Green_out <= char_Y(Line, Pixel);
        Blue_out <= char_Y(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "01011010") then
        Red_out <= char_Z(Line, Pixel);
        Green_out <= char_Z(Line, Pixel);
        Blue_out <= char_Z(Line, Pixel);
        Pixel:= Pixel+1;
    elsif (ascii8 <= "11111111") then
        Red_out <= char_Reset(Line, Pixel);
        Green_out <= char_Reset(Line, Pixel);
        Blue_out <= char_Reset(Line, Pixel);
        Pixel:= Pixel+1;
    end if;
end if;
----digit 9
elsif (Horizontal_Counter >= "0111101000") --488
    and (Horizontal_Counter <= "0111101111") -- 495
    and (Vertical_Counter >= "0011101010") --234
    and (Vertical_Counter <= "0011110101") then -- 245
    if (Pixel <= 71) then
        if (ascii9 <= "00000000") then
            Red_out <= char_Reset(Line, Pixel);
            Green_out <= char_Reset(Line, Pixel);
            Blue_out <= char_Reset(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii9 <= "01000001") then
            Red_out <= char_A(Line, Pixel);
            Green_out <= char_A(Line, Pixel);
            Blue_out <= char_A(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii9 <= "01000010") then
            Red_out <= char_B(Line, Pixel);
            Green_out <= char_B(Line, Pixel);
            Blue_out <= char_B(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii9 <= "01000011") then
            Red_out <= char_C(Line, Pixel);
            Green_out <= char_C(Line, Pixel);
            Blue_out <= char_C(Line, Pixel);
            Pixel:= Pixel+1;
        elsif (ascii9 <= "01000100") then
            Red_out <= char_D(Line, Pixel);
            Green_out <= char_D(Line, Pixel);
            Blue_out <= char_D(Line, Pixel);

```



```

Pixel:= Pixel+1;
elsif (ascii9 <= "01000101") then
  Red_out <= char_E(Line, Pixel);
  Green_out <= char_E(Line, Pixel);
  Blue_out <= char_E(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01000110") then
  Red_out <= char_F(Line, Pixel);
  Green_out <= char_F(Line, Pixel);
  Blue_out <= char_F(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01000111") then
  Red_out <= char_G(Line, Pixel);
  Green_out <= char_G(Line, Pixel);
  Blue_out <= char_G(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001000") then
  Red_out <= char_H(Line, Pixel);
  Green_out <= char_H(Line, Pixel);
  Blue_out <= char_H(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001001") then
  Red_out <= char_I(Line, Pixel);
  Green_out <= char_I(Line, Pixel);
  Blue_out <= char_I(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001010") then
  Red_out <= char_J(Line, Pixel);
  Green_out <= char_J(Line, Pixel);
  Blue_out <= char_J(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001011") then
  Red_out <= char_K(Line, Pixel);
  Green_out <= char_K(Line, Pixel);
  Blue_out <= char_K(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001100") then
  Red_out <= char_L(Line, Pixel);
  Green_out <= char_L(Line, Pixel);
  Blue_out <= char_L(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001101") then
  Red_out <= char_M(Line, Pixel);
  Green_out <= char_M(Line, Pixel);
  Blue_out <= char_M(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001110") then
  Red_out <= char_N(Line, Pixel);
  Green_out <= char_N(Line, Pixel);
  Blue_out <= char_N(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01001111") then
  Red_out <= char_O(Line, Pixel);
  Green_out <= char_O(Line, Pixel);
  Blue_out <= char_O(Line, Pixel);

```

```

Pixel:= Pixel+1;
elsif (ascii9 <= "01010000") then
  Red_out <= char_P(Line, Pixel);
  Green_out <= char_P(Line, Pixel);
  Blue_out <= char_P(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010001") then
  Red_out <= char_Q(Line, Pixel);
  Green_out <= char_Q(Line, Pixel);
  Blue_out <= char_Q(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010010") then
  Red_out <= char_R(Line, Pixel);
  Green_out <= char_R(Line, Pixel);
  Blue_out <= char_R(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010011") then
  Red_out <= char_S(Line, Pixel);
  Green_out <= char_S(Line, Pixel);
  Blue_out <= char_S(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010100") then
  Red_out <= char_T(Line, Pixel);
  Green_out <= char_T(Line, Pixel);
  Blue_out <= char_T(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010101") then
  Red_out <= char_U(Line, Pixel);
  Green_out <= char_U(Line, Pixel);
  Blue_out <= char_U(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010110") then
  Red_out <= char_V(Line, Pixel);
  Green_out <= char_V(Line, Pixel);
  Blue_out <= char_V(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01010111") then
  Red_out <= char_W(Line, Pixel);
  Green_out <= char_W(Line, Pixel);
  Blue_out <= char_W(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01011000") then
  Red_out <= char_X(Line, Pixel);
  Green_out <= char_X(Line, Pixel);
  Blue_out <= char_X(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01011001") then
  Red_out <= char_Y(Line, Pixel);
  Green_out <= char_Y(Line, Pixel);
  Blue_out <= char_Y(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii9 <= "01011010") then
  Red_out <= char_Z(Line, Pixel);
  Green_out <= char_Z(Line, Pixel);
  Blue_out <= char_Z(Line, Pixel);

```

```

Pixel:= Pixel+1;
elsif (ascii9 <= "11111111") then
  Red_out <= char_Reset(Line, Pixel);
  Green_out <= char_Reset(Line, Pixel);
  Blue_out <= char_Reset(Line, Pixel);
  Pixel:= Pixel+1;
end if;
end if;

```

----digit 10

```

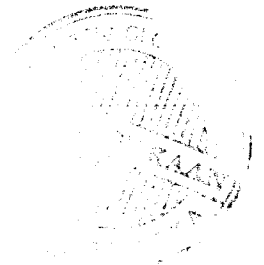
elsif (Horizontal_Counter >= "0111110000")--496
  and (Horizontal_Counter <= "0111110111")-- 503
  and (Vertical_Counter >= "0011101010") --234
  and (Vertical_Counter <= "0011101011") then -- 245
  if (Pixel <= 82) then
    if (ascii10 <= "00000000") then
      Red_out <= char_Reset(Line, Pixel);
      Green_out <= char_Reset(Line, Pixel);
      Blue_out <= char_Reset(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000001")then
      Red_out <= char_A(Line, Pixel);
      Green_out <= char_A(Line, Pixel);
      Blue_out <= char_A(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000010") then
      Red_out <= char_B(Line, Pixel);
      Green_out <= char_B(Line, Pixel);
      Blue_out <= char_B(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000011") then
      Red_out <= char_C(Line, Pixel);
      Green_out <= char_C(Line, Pixel);
      Blue_out <= char_C(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000100") then
      Red_out <= char_D(Line, Pixel);
      Green_out <= char_D(Line, Pixel);
      Blue_out <= char_D(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000101") then
      Red_out <= char_E(Line, Pixel);
      Green_out <= char_E(Line, Pixel);
      Blue_out <= char_E(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000110") then
      Red_out <= char_F(Line, Pixel);
      Green_out <= char_F(Line, Pixel);
      Blue_out <= char_F(Line, Pixel);
      Pixel:= Pixel+1;
    elsif (ascii10 <= "01000111") then
      Red_out <= char_G(Line, Pixel);
      Green_out <= char_G(Line, Pixel);
      Blue_out <= char_G(Line, Pixel);
      Pixel:= Pixel+1;
    end if;
  end if;
end if;

```

```

elsif (ascii10 <= "01001000") then
    Red_out <= char_H(Line, Pixel);
    Green_out <= char_H(Line, Pixel);
    Blue_out <= char_H(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001001") then
    Red_out <= char_I(Line, Pixel);
    Green_out <= char_I(Line, Pixel);
    Blue_out <= char_I(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001010") then
    Red_out <= char_J(Line, Pixel);
    Green_out <= char_J(Line, Pixel);
    Blue_out <= char_J(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001011") then
    Red_out <= char_K(Line, Pixel);
    Green_out <= char_K(Line, Pixel);
    Blue_out <= char_K(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001100") then
    Red_out <= char_L(Line, Pixel);
    Green_out <= char_L(Line, Pixel);
    Blue_out <= char_L(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001101") then
    Red_out <= char_M(Line, Pixel);
    Green_out <= char_M(Line, Pixel);
    Blue_out <= char_M(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001110") then
    Red_out <= char_N(Line, Pixel);
    Green_out <= char_N(Line, Pixel);
    Blue_out <= char_N(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01001111") then
    Red_out <= char_O(Line, Pixel);
    Green_out <= char_O(Line, Pixel);
    Blue_out <= char_O(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01010000") then
    Red_out <= char_P(Line, Pixel);
    Green_out <= char_P(Line, Pixel);
    Blue_out <= char_P(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01010001") then
    Red_out <= char_Q(Line, Pixel);
    Green_out <= char_Q(Line, Pixel);
    Blue_out <= char_Q(Line, Pixel);
    Pixel:= Pixel+1;
elsif (ascii10 <= "01010010") then
    Red_out <= char_R(Line, Pixel);
    Green_out <= char_R(Line, Pixel);
    Blue_out <= char_R(Line, Pixel);

```



```

Pixel:= Pixel+1;
elsif (ascii10 <= "01010011") then
  Red_out <= char_S(Line, Pixel);
  Green_out <= char_S(Line, Pixel);
  Blue_out <= char_S(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01010100") then
  Red_out <= char_T(Line, Pixel);
  Green_out <= char_T(Line, Pixel);
  Blue_out <= char_T(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01010101") then
  Red_out <= char_U(Line, Pixel);
  Green_out <= char_U(Line, Pixel);
  Blue_out <= char_U(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01010110") then
  Red_out <= char_V(Line, Pixel);
  Green_out <= char_V(Line, Pixel);
  Blue_out <= char_V(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01010111") then
  Red_out <= char_W(Line, Pixel);
  Green_out <= char_W(Line, Pixel);
  Blue_out <= char_W(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01011000") then
  Red_out <= char_X(Line, Pixel);
  Green_out <= char_X(Line, Pixel);
  Blue_out <= char_X(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01011001") then
  Red_out <= char_Y(Line, Pixel);
  Green_out <= char_Y(Line, Pixel);
  Blue_out <= char_Y(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "01011010") then
  Red_out <= char_Z(Line, Pixel);
  Green_out <= char_Z(Line, Pixel);
  Blue_out <= char_Z(Line, Pixel);
  Pixel:= Pixel+1;
elsif (ascii10 <= "00100000") then
  Red_out <= char_Reset(Line, Pixel);
  Green_out <= char_Reset(Line, Pixel);
  Blue_out <= char_Reset(Line, Pixel);
  Pixel:= Pixel+1;
end if;
end if;

```

```

elsif(Pixel >= 83) then -- Semua yang ga aktif hitam
  Red_out <= '0';
  Green_out <= '0';

```

```

        Blue_out <= '0';

    end if;
end if;

if (Horizontal_Counter > "0000000000" )
    and (Horizontal_Counter < "0001100001" ) -- 96+1
    then
        hs_out <= '0';
    else
        hs_out <= '1';
    end if;

if (Vertical_Counter > "0000000000" )
    and (Vertical_Counter < "0000000011" ) -- 2+1
    then
        vs_out <= '0';
    else
        vs_out <= '1';
    end if;

Horizontal_Counter <= Horizontal_Counter+"0000000001";
if (Horizontal_Counter="1100100000") then ---800
    Vertical_Counter <= Vertical_Counter+"0000000001";
    Horizontal_Counter <= "0000000000";
    Pixel:= 0;
if (Vertical_Counter >= "0011101010") ----Line pertama text
and (Vertical_Counter <= "0100000100") then ----Line paling bawah dari text 260
    if (Line <= 11) then
        Line:= Line+1;
    elsif (Line >= 12) then
        Line:= 0;
    end if;
end if;
end if;
end if;
if (Vertical_Counter="1000001001") then ----521
    Vertical_Counter <= "0000000000";
    Line:= 0;

    end if;
end if;
end process;
end Behavioral;

```

## LAMPIRAN 2

Gambar RTL Schematic

