ALU AUTOMATIZADA

Integrantes:

Kennedy Villa Carolina

Murrieta Villegas Alfonso

Reza Chavarria Sergio Gabriel

Valdespino Mendieta Joaquin

Objetivos

- El alumno realizará la conexión de componentes, estos revisados durante el curso, para la creación de una Unidad Lógica Aritmética que muestre las operaciones y resultados, de manera automática, por medio del puerto VGA.

Introducción

Componentes utilizados

- UNIDAD LÓGICA ARITMÉTICA (ALU)
- MEMORIA ROM
- CONTADORES
- PUERTO VGA

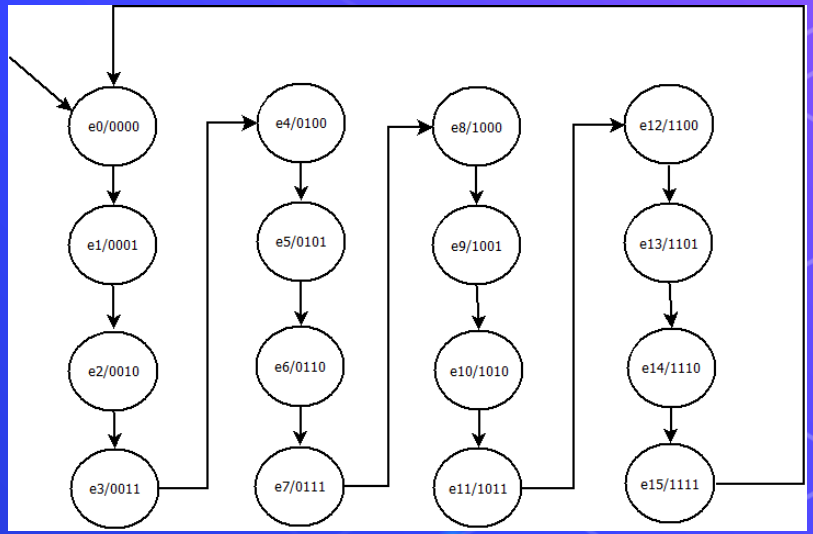
Desarrollo

- Se necesitaron tomar en cuenta las operaciones que realiza la ALU.
- Se utiliza las entradas sel y cin para modificar las operaciones, posteriormente, en el VGA

Cin	Sel(2)	Sel(1)	Sel(0)	MUX B	Unidad	Operación	Función
0	0	0	0	$'\mathbf{0}'$	UA	$suma(A + '0')$	$F = A$
0	0	0	1	$\bar{\boldsymbol{B}}$	UA	$suma(A + \bar{B})$	$F = A + \bar{B}$
0	0	1	0	$\boldsymbol{B}$	UA	$suma(A + B)$	$F = A + B$
0	0	1	1	$'\mathbf{1}'$	UA	$suma(A +' 1')$	$F = A - 1 = A - -$
0	1	0	0	$'\mathbf{0}'$	UL	AND	$F = A\ AND\ B$
0	1	0	1	$\bar{\boldsymbol{B}}$	UL	OR	$F = A\ OR\ B$
0	1	1	0	$\boldsymbol{B}$	UL	XOR	$F = A\ XOR\ B$
0	1	1	1	$'\mathbf{1}'$	UL	NOT	$F = NOT\ A$
1	0	0	0	$'\mathbf{0}'$	UA	$resta(A - 0')$	$F = A + 1$
1	0	0	1	$\bar{\boldsymbol{B}}$	UA	$resta(A - \bar{B})$	$F = A + \bar{B} + 1 = A - B$
1	0	1	0	$\boldsymbol{B}$	UA	$resta(A - B)$	$F = A + B + 1$
1	0	1	1	$'\mathbf{1}'$	UA	$resta(A -' 1')$	$F = A$
1	1	0	0	$'\mathbf{0}'$	UL	AND	$F = A\ AND\ B$
1	1	0	1	$\bar{\boldsymbol{B}}$	UL	OR	$F = A\ OR\ B$
1	1	1	0	$\boldsymbol{B}$	UL	XOR	$F = A\ XOR$
1	1	1	1	$'\mathbf{1}'$	UL	NOT	$F = NOT\ A$

Desarrollo

- Realización de un contador de 16 estados para el proceso automático. El contador se conectará a la ALU



Desarrollo

- Se utilizó las memorias ROM la cuales guardan:
  - Estructura de display de 7 segmentos
  - Representación en código Binario

Representación

- Cin: Cambio entre suma o resta
- SEL (2): Cambio entre operación aritmética y lógica
- SEL (0 to 1): Cambio de operaciones

Caso UA

- 00: Operación con solo ceros.
- 01: B negada
- 10: B
- 11: Operación con solo unos.

Caso UL

- 00: AND (X)
- 01: OR (+)
- 10: XOR (+ con una línea)
- 11: NOT (Línea arriba del a, no se mostrará b)

Código (ALU)

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

Search altera.com

```
library ieee;
use ieee.std_logic_1164.all;

entity mux4x1 is port(
    b: in std_logic_vector(2 downto 0);
    s: in std_logic_vector(1 downto 0);
    sal: out std_logic_vector(2 downto 0)
);
end mux4x1;

architecture arqmux of mux4x1 is
begin
    with s select

    sal   <= (others => '0') when "00",
             not b           when "01",
             b               when "10",
             (others => '1') when "11",
             (others => '0') when others;


end arqmux;
```

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

Search altera.com

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity UA is port(
    a,b: in std_logic_vector(2 downto 0);
    s0: in std_logic_vector (1 downto 0);
    cin: in std_logic;
    salsum: out std_logic_vector(2 downto 0);
    cout: out std_logic
);
end entity;

architecture arq_UA of UA is
signal sal: std_logic_vector(2 downto 0);
begin
    u1: entity work.mux4x1(arqmux) port map(b,s0,sal);
    u2: entity work.sum(arqsum) port map (a, sal, cin,salsum, cout);

end arq_UA;
```

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

Search altera.com

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity sum is port(
    a,b: in std_logic_vector(2 downto 0);
    cin: in std_logic;
    salsum: out std_logic_vector(2 downto 0);
    cout: out std_logic
);
end sum;

architecture arqsum of sum is
signal mid: std_logic_vector(3 downto 0);

begin
    mid<=('0'&a)+('0'&b)+cin;
    cout<=mid(3);
    salsum<=mid(2 downto 0);

end architecture arqsum;
```

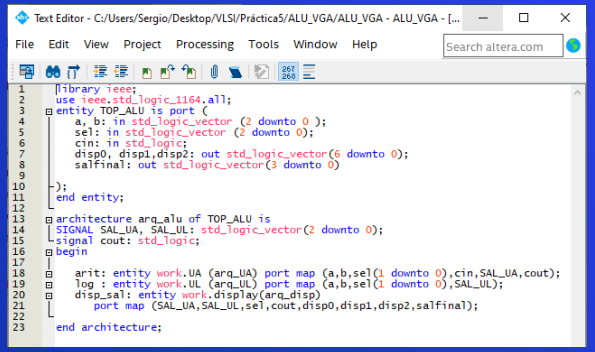
Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

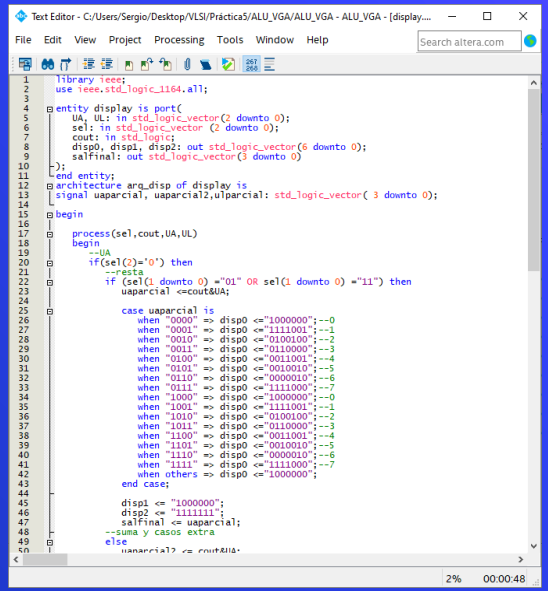
Search altera.com

```
library ieee;
use ieee.std_logic_1164.all;
entity UL is port(
    a,b: in std_logic_vector (2 downto 0);
    sel: in std_logic_vector (1 downto 0);
    sallog: out std_logic_vector(2 downto 0)
);
end entity;

architecture arq_UL of UL is
signal cand,cor,cxor,cnot: std_logic_vector (2 downto 0);
begin
    cand<=a and b;
    cor <=a or b;
    cxor<=a xor b;
    cnot<=not a;

    with sel select
    sallog <=
             cand when "00",
             cor  when "01",
             cxor when "10",
             cnot when "11";

end arq_UL;
```

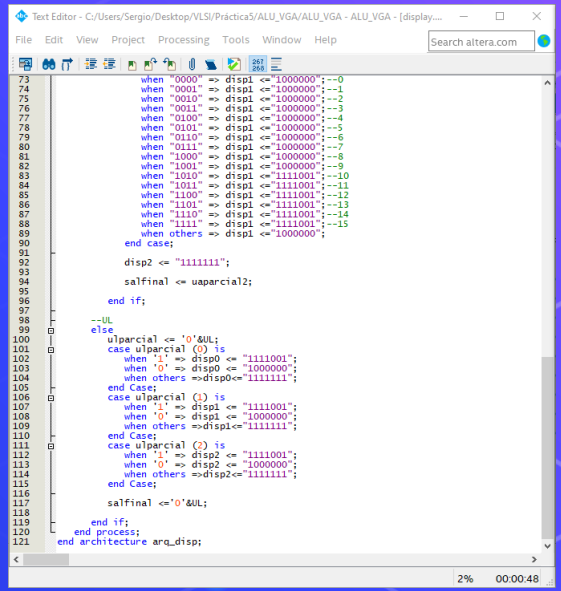
Código (ALU)



```
library ieee;
use ieee.std_logic_1164.all;
entity TOP_ALU is port (
    a, b: in std_logic_vector (2 downto 0 );
    sel: in std_logic_vector (2 downto 0);
    cin: in std_logic;
    disp0, disp1,disp2: out std_logic_vector(6 downto 0);
    salfinal: out std_logic_vector(3 downto 0)

);
end entity;

architecture arq_alu of TOP_ALU is
SIGNAL SAL_UA, SAL_UL: std_logic_vector(2 downto 0);
signal cout: std_logic;
begin

    arit: entity work.UA (arq_UA) port map (a,b,sel(1 downto 0),cin,SAL_UA,cout);
    log : entity work.UL (arq_UL) port map (a,b,sel(1 downto 0),SAL_UL);
    disp_sal: entity work.display(arq_disp)
        port map (SAL_UA,SAL_UL,sel,cout,disp0,disp1,disp2,salfinal);

end architecture;
```



```
library ieee;
use ieee.std_logic_1164.all;

entity display is port(
    UA, UL: in std_logic_vector(2 downto 0);
    sel: in std_logic_vector (2 downto 0);
    cout: in std_logic;
    disp0, disp1, disp2: out std_logic_vector(6 downto 0);
    salfinal: out std_logic_vector(3 downto 0)
);
end entity;
architecture arq_disp of display is
signal uaparcial, uaparcial2,ulparcial: std_logic_vector( 3 downto 0);

begin

    process(sel,cout,UA,UL)
    begin
        --UA
        if(sel(2)='0') then
            --resta
            if (sel(1 downto 0) ="01" OR sel(1 downto 0) ="11") then
                uaparcial <=cout&UA;

                case uaparcial is
                    when "0000" => disp0 <="1000000";--0
                    when "0001" => disp0 <="1111001";--1
                    when "0010" => disp0 <="0100100";--2
                    when "0011" => disp0 <="0110000";--3
                    when "0100" => disp0 <="0011001";--4
                    when "0101" => disp0 <="0010010";--5
                    when "0110" => disp0 <="0000010";--6
                    when "0111" => disp0 <="1111000";--7
                    when "1000" => disp0 <="1000000";--0
                    when "1001" => disp0 <="1111001";--1
                    when "1010" => disp0 <="0100100";--2
                    when "1011" => disp0 <="0110000";--3
                    when "1100" => disp0 <="0011001";--4
                    when "1101" => disp0 <="0010010";--5
                    when "1110" => disp0 <="0000010";--6
                    when "1111" => disp0 <="1111000";--7
                    when others => disp0 <="1000000";
                end case;

                disp1 <= "1000000";
                disp2 <= "1111111";
                salfinal <= uaparcial;
            --suma y casos extra
            else
                uaparcial2 <= cout&UA;
```



```
                when "0000" => disp1 <="1000000";--0
                when "0001" => disp1 <="1000000";--1
                when "0010" => disp1 <="1000000";--2
                when "0011" => disp1 <="1000000";--3
                when "0100" => disp1 <="1000000";--4
                when "0101" => disp1 <="1000000";--5
                when "0110" => disp1 <="1000000";--6
                when "0111" => disp1 <="1000000";--7
                when "1000" => disp1 <="1000000";--8
                when "1001" => disp1 <="1000000";--9
                when "1010" => disp1 <="1111001";--10
                when "1011" => disp1 <="1111001";--11
                when "1100" => disp1 <="1111001";--12
                when "1101" => disp1 <="1111001";--13
                when "1110" => disp1 <="1111001";--14
                when "1111" => disp1 <="1111001";--15
                when others => disp1 <="1000000";
            end case;

            disp2 <= "1111111";

            salfinal <= uaparcial2;

        end if;

    --UL
    else
        ulparcial <= '0'&UL;
        case ulparcial (0) is
            when '1' => disp0 <= "1111001";
            when '0' => disp0 <= "1000000";
            when others =>disp0<="1111111";
        end Case;
        case ulparcial (1) is
            when '1' => disp1 <= "1111001";
            when '0' => disp1 <= "1000000";
            when others =>disp1<="1111111";
        end Case;
        case ulparcial (2) is
            when '1' => disp2 <= "1111001";
            when '0' => disp2 <= "1000000";
            when others =>disp2<="1111111";
        end Case;

        salfinal <='0'&UL;

    end if;
  end process;
end architecture arq_disp;
```

Código (Contador)

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

Search altera.com

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_arith.all;
4  entity relojlento is port (
5      clk1: in std_logic;
6      led: buffer std_logic:='0'
7  );
8  end relojlento;
9
10 architecture arqrelojlento of relojlento is
11    signal conteo: integer range 0 to 50000000;
12 begin
13    process (clk1)
14       begin
15       if (clk1' event and clk1='1') then
16          conteo <=conteo+1;
17          if (conteo=50000000) then
18             conteo<=0;
19             led<=not(led);
20          end if;
21       end if;
22    end process;
23
24 end arqrelojlento;
```

0% 00:00:00

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [contad...

File Edit View Project Processing Tools Window Help

Search altera.com

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  entity contador is port(
4      clk, reset: in std_logic;
5      sal: out std_logic_vector(3 downto 0)
6  );
7  end entity;
8  architecture arqcont of contador is
9  subtype state is std_logic_vector(3 downto 0);
10 signal present_state, next_state: state;
11    constant e0: state   :="0000";
12    constant e1: state   :="0001";
13    constant e2: state   :="0010";
14    constant e3: state   :="0011";
15    constant e4: state   :="0100";
16    constant e5: state   :="0101";
17    constant e6: state   :="0110";
18    constant e7: state   :="0111";
19    constant e8: state   :="1000";
20    constant e9: state   :="1001";
21    constant e10: state  :="1010";
22    constant e11: state  :="1011";
23    constant e12: state  :="1100";
24    constant e13: state  :="1101";
25    constant e14: state  :="1110";
26    constant e15: state  :="1111";
27 begin
28    process(clk)
29    begin
30       if rising_edge(clk) then
31          if (reset='0') then
32             present_state<=e0;
33          else
34             present_state<=next_state;
35          end if;
36       end if;
37    end process;
38 process (present_state)
39 begin
40    case present_state is
41       when e0=> next_state <=e1;
42       when e1=> next_state <=e2;
43       when e2=> next_state <=e3;
44       when e3=> next_state <=e4;
45       when e4=> next_state <=e5;
46       when e5=> next_state <=e6;
47       when e6=> next_state <=e7;
48       when e7=> next_state <=e8;
49       when e8=> next_state <=e9;
50       when e9=> next_state <=e10;
51       when e10=> next_state <=e11;
52       when e11=> next_state <=e12;
53       when e12=> next_state <=e13;
54       when e13=> next_state <=e14;
55       when e14=> next_state <=e15;
56       when e15=> next_state <=e0;
57       when others=> next_state <=e0;
58    end case;
59    sal<=present_state;
60 end process;
61 end arqcont;
```

Código (VGA)

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

File Edit View Project Processing Tools Window Help

Search altera.com

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_UNSIGNED.all;
4  entity romDatos is port
5  (
6      bus_dir: in std_logic_vector(2 downto 0);
7      cs: in std_logic;
8      bus_datos: out std_logic_vector(2 downto 0)
9  );
10 end romDatos;
11 architecture arqromDatos of romDatos is
12     constant L0: std_logic_vector(2 downto 0):="000";
13     constant L1: std_logic_vector(2 downto 0):="001";
14     constant L2: std_logic_vector(2 downto 0):="010";
15     constant L3: std_logic_vector(2 downto 0):="011";
16     constant L4: std_logic_vector(2 downto 0):="100";
17     constant L5: std_logic_vector(2 downto 0):="101";
18     constant L6: std_logic_vector(2 downto 0):="110";
19     constant L7: std_logic_vector(2 downto 0):="111";
20     type memoria is array (7 downto 0) of std_logic_vector(2 downto 0);
21     constant mem_rom:memoria:=(L7,L6,L5,L4,L3,L2,L1,L0);
22     signal dato: std_logic_vector(2 downto 0);
23 begin
24     prom: process(bus_dir)
25     begin
26         dato<=mem_rom(conv_integer(bus_dir));
27     end process prom;
28
29     pbuf: process(dato,cs)
30     begin
31         if(cs='1') then
32             bus_datos<=dato;
33         else
34             bus_datos<=(others=>'Z');
35         end if;
36     end process pbuf;
37 end arqromDatos;
```

2% 00:00:48

Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [...

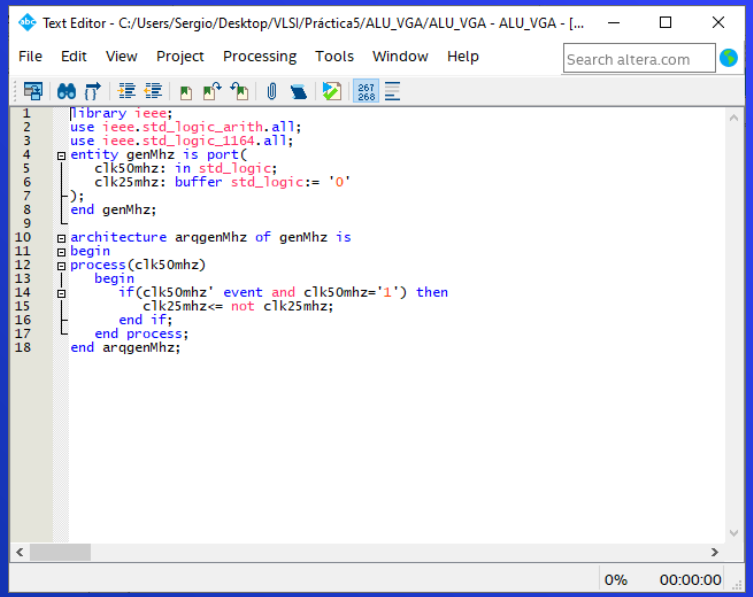
File Edit View Project Processing Tools Window Help

Search altera.com

```
1  library ieee;
2  use ieee.std_logic_1164.all;
3  use ieee.std_logic_UNSIGNED.all;
4  entity rom is port
5  (
6      bus_dir: in std_logic_vector(2 downto 0);
7      cs: in std_logic;
8      bus_datos: out std_logic_vector(6 downto 0)
9  );
10 end rom;
11 architecture arqrom of rom is
12     constant L0: std_logic_vector(6 downto 0):="1000000";
13     constant L1: std_logic_vector(6 downto 0):="1111001";
14     constant L2: std_logic_vector(6 downto 0):="0100100";
15     constant L3: std_logic_vector(6 downto 0):="0110000";
16     constant L4: std_logic_vector(6 downto 0):="0011001";
17     constant L5: std_logic_vector(6 downto 0):="0010010";
18     constant L6: std_logic_vector(6 downto 0):="0000010";
19     constant L7: std_logic_vector(6 downto 0):="1111000";
20     type memoria is array (7 downto 0) of std_logic_vector(6 downto 0);
21     constant mem_rom:memoria:=(L7,L6,L5,L4,L3,L2,L1,L0);
22     signal dato: std_logic_vector(6 downto 0);
23 begin
24     prom: process(bus_dir)
25     begin
26         dato<=mem_rom(conv_integer(bus_dir));
27     end process prom;
28
29     pbuf: process(dato,cs)
30     begin
31         if(cs='1') then
32             bus_datos<=dato;
33         else
34             bus_datos<=(others=>'Z');
35         end if;
36     end process pbuf;
37 end arqrom;
```

2% 00:00:48

Código (VGA)

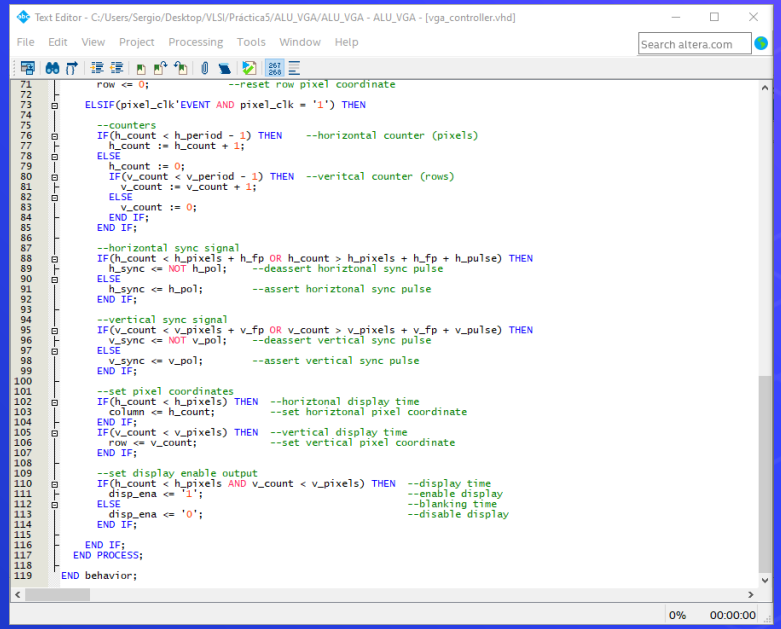



```
library ieee;
use ieee.std_logic_arith.all;
use ieee.std_logic_1164.all;
entity genMhz is port(
    clk50mhz: in std_logic;
    clk25mhz: buffer std_logic:= '0'
);
end genMhz;

architecture arqgenMhz of genMhz is
begin
process(clk50mhz)
    begin
        if(clk50mhz' event and clk50mhz='1') then
            clk25mhz<= not clk25mhz;
        end if;
    end process;
end arqgenMhz;
```

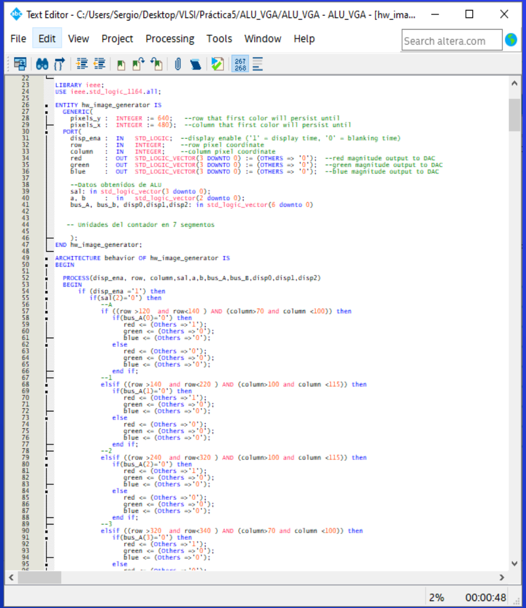
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY vga_controller IS
  GENERIC(
    h_pulse  :  INTEGER   := 96;   --horiztonal sync pulse width in pixels
    h_bp     :  INTEGER   := 48;   --horiztonal back porch width in pixels
    h_pixels :  INTEGER   := 640;  --horiztonal display width in pixels
    h_fp     :  INTEGER   := 16;   --horiztonal front porch width in pixels
    h_pol    :  STD_LOGIC := '0';   --horizontal sync pulse polarity (1 = positive, 0
    v_pulse  :  INTEGER   := 2;    --vertical sync pulse width in rows
    v_bp     :  INTEGER   := 33;    --vertical back porch width in rows
    v_pixels :  INTEGER   := 480;  --vertical display width in rows
    v_fp     :  INTEGER   := 10;    --vertical front porch width in rows
    v_pol    :  STD_LOGIC := '0');  --vertical sync pulse polarity (1 = positive, 0 =

  PORT(
    pixel_clk :  IN   STD_LOGIC;  --pixel clock at frequency of VGA mode being used
    reset_n   :  IN   STD_LOGIC;  --active low asycnchronous reset
    h_sync    :  OUT  STD_LOGIC;  --horiztonal sync pulse
    v_sync    :  OUT  STD_LOGIC;  --vertical sync pulse
    disp_ena  :  OUT  STD_LOGIC;  --display enable ('1' = display time, '0' = blanking
    column    :  OUT  INTEGER;    --horizontal pixel coordinate
    row       :  OUT  INTEGER;--;    --vertical pixel coordinate
    n_blank   :  OUT  STD_LOGIC;  --direct blacking output to DAC
    n_sync    :  OUT  STD_LOGIC); --sync-on-green output to DAC
END vga_controller;
```

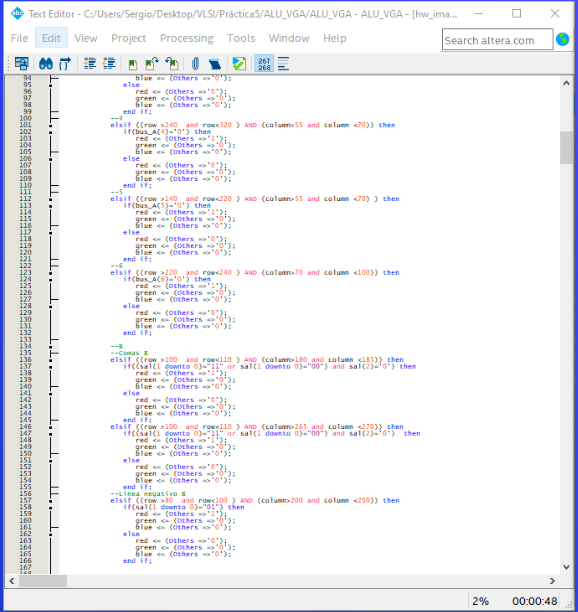
```
ARCHITECTURE behavior OF vga_controller IS
  CONSTANT  h_period  :  INTEGER := h_pulse + h_bp + h_pixels + h_fp;  --total number of pixel clocks in a row
  CONSTANT  v_period  :  INTEGER := v_pulse + v_bp + v_pixels + v_fp;  --total number of rows in column
BEGIN

  --n_blank <= '1';  --no direct blanking
  --n_sync <= '0';   --no sync on green

  PROCESS(pixel_clk, reset_n)
    VARIABLE h_count  :  INTEGER RANGE 0 TO h_period - 1 := 0;  --horizontal counter (counts the columns)
    VARIABLE v_count  :  INTEGER RANGE 0 TO v_period - 1 := 0;  --vertical counter (counts the rows)
  BEGIN

    IF(reset_n = '0') THEN  --reset asserted
      h_count := 0;         --reset horizontal counter
      v_count := 0;         --reset vertical counter
      h_sync <= h_pol;  --deassert horizontal sync
      v_sync <= v_pol;  --deassert vertical sync
      disp_ena <= '0';        --disable display
      column <= 0;            --reset column pixel coordinate
      row <= 0;               --reset row pixel coordinate

    ELSIF(pixel_clk'EVENT AND pixel_clk = '1') THEN

      --counters
      IF(h_count < h_period - 1) THEN    --horizontal counter (pixels)
        h_count := h_count + 1;
      ELSE
        h_count := 0;
        IF(v_count < v_period - 1) THEN  --veritcal counter (rows)
          v_count := v_count + 1;
        ELSE
          v_count := 0;
        END IF;
      END IF;

      --horizontal sync signal
      IF(h_count < h_pixels + h_fp OR h_count > h_pixels + h_fp + h_pulse) THEN
        h_sync <= NOT h_pol;    --deassert horiztonal sync pulse
      ELSE
        h_sync <= h_pol;        --assert horiztonal sync pulse
      END IF;

      --vertical sync signal
      IF(v_count < v_pixels + v_fp OR v_count > v_pixels + v_fp + v_pulse) THEN
        v_sync <= NOT v_pol;    --deassert vertical sync pulse
      ELSE
        v_sync <= v_pol;        --assert vertical sync pulse
      END IF;
```

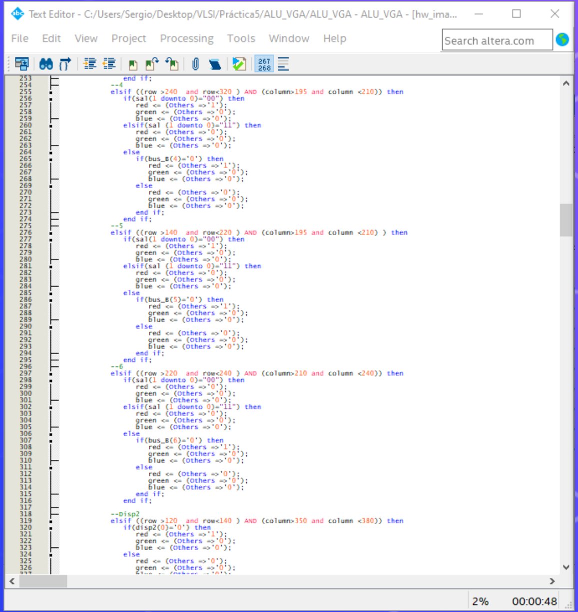
Código (VGA)


Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [vga_controller.vhd]
File Edit View Project Processing Tools Window Help
Search altera.com
row <= 0; --reset row pixel coordinate
ELSIF(pixel_clk'EVENT AND pixel_clk = '1') THEN
--counters
IF(h_count < h_period - 1) THEN --horizontal counter (pixels)
h_count := h_count + 1;
ELSE
h_count := 0;
IF(v_count < v_period - 1) THEN --veritcal counter (rows)
v_count := v_count + 1;
ELSE
v_count := 0;
END IF;
END IF;
--horizontal sync signal
IF(h_count < h_pixels + h_fp OR h_count > h_pixels + h_fp + h_pulse) THEN
h_sync <= NOT h_pol; --deassert horiztonal sync pulse
ELSE
h_sync <= h_pol; --assert horiztonal sync pulse
END IF;
--vertical sync signal
IF(v_count < v_pixels + v_fp OR v_count > v_pixels + v_fp + v_pulse) THEN
v_sync <= NOT v_pol; --deassert vertical sync pulse
ELSE
v_sync <= v_pol; --assert vertical sync pulse
END IF;
--set pixel coordinates
IF(h_count < h_pixels) THEN --horiztonal display time
column <= h_count; --set horiztonal pixel coordinate
END IF;
IF(v_count < v_pixels) THEN --vertical display time
row <= v_count; --set vertical pixel coordinate
END IF;
--set display enable output
IF(h_count < h_pixels AND v_count < v_pixels) THEN --display time
disp_ena <= '1'; --enable display
ELSE --blanking time
disp_ena <= '0'; --disable display
END IF;
END IF;
END PROCESS;
END behavior;
0% 00:00:00

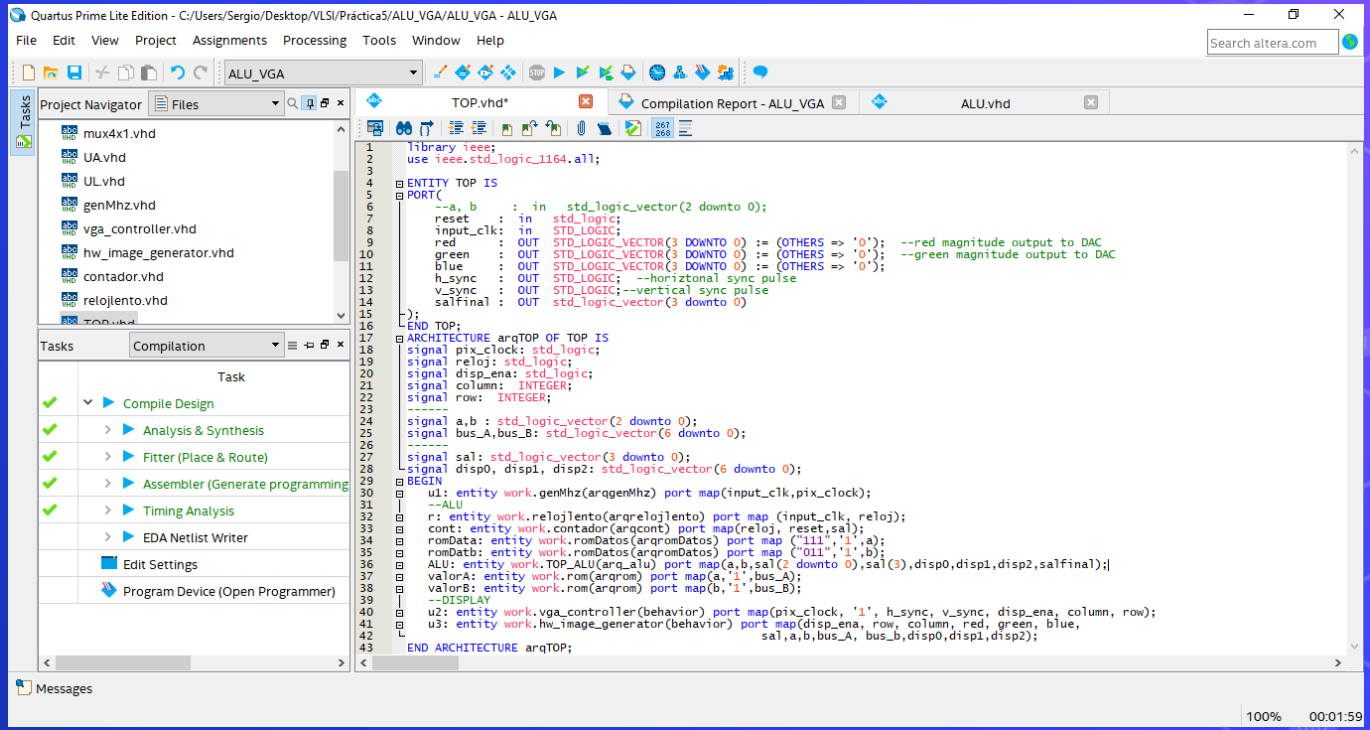
Código (VGA)


Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [hw_ima...
File Edit View Project Processing Tools Window Help
Search altera.com
2% 00:00:48


Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [hw_ima...
File Edit View Project Processing Tools Window Help
Search altera.com
2% 00:00:48


Text Editor - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA - [hw_ima...
File Edit View Project Processing Tools Window Help
Search altera.com
2% 00:00:48

Código (TOP)

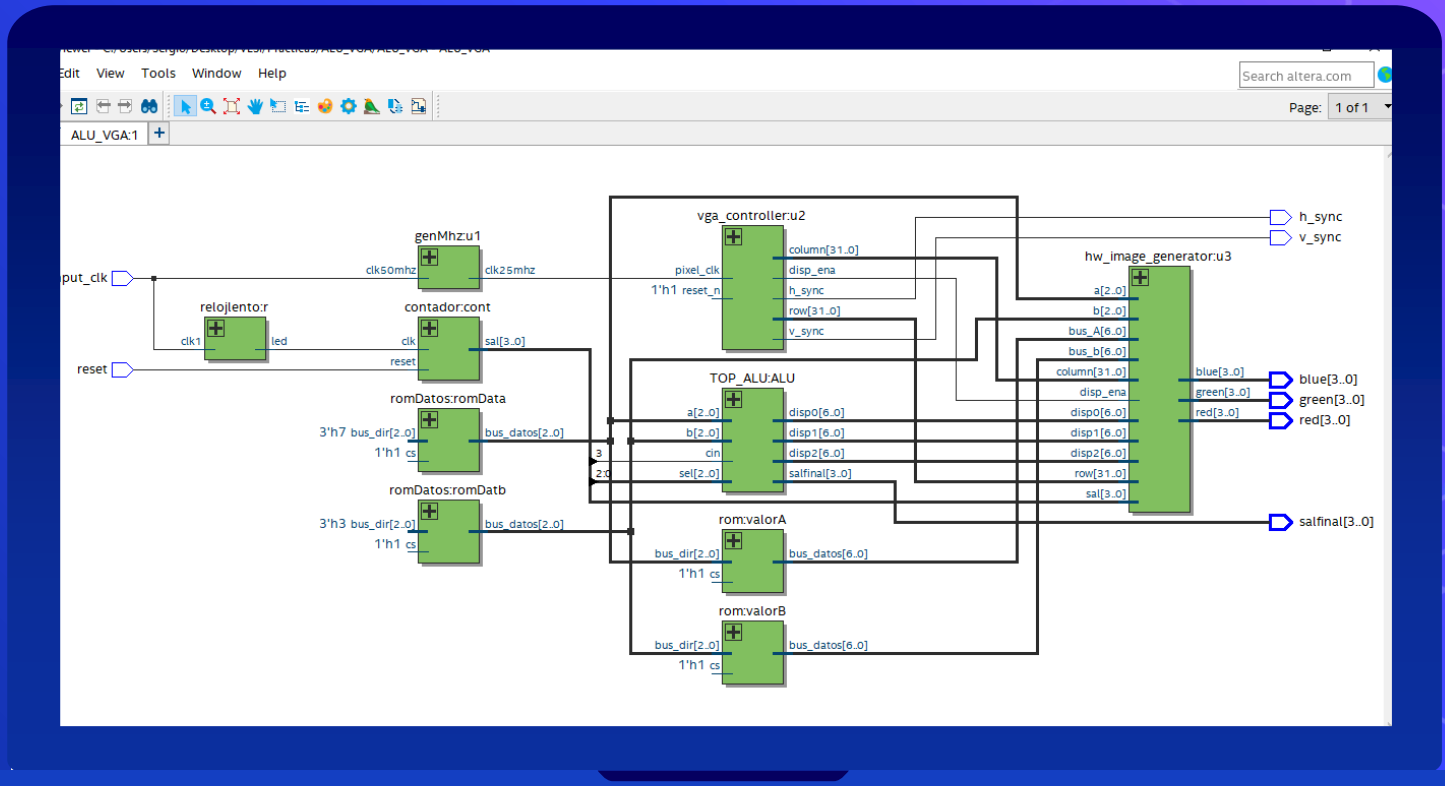

Quartus Prime Lite Edition - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA
File Edit View Project Assignments Processing Tools Window Help
Search altera.com
ALU_VGA
Project Navigator
Files
mux4x1.vhd
UA.vhd
UL.vhd
genMhz.vhd
vga_controller.vhd
hw_image_generator.vhd
contador.vhd
relojlento.vhd
TOP.vhd
Tasks
Compilation
Task
Compile Design
Analysis & Synthesis
Fitter (Place & Route)
Assembler (Generate programming
Timing Analysis
EDA Netlist Writer
Edit Settings
Program Device (Open Programmer)
TOP.vhd*
Compilation Report - ALU_VGA
ALU.vhd
library ieee;
use ieee.std_logic_1164.all;
ENTITY TOP IS
PORT(
--a, b : in std_logic_vector(2 downto 0);
reset : in STD_LOGIC;
input_clk: in STD_LOGIC;
red : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) := (OTHERS => '0'); --red magnitude output to DAC
green : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) := (OTHERS => '0'); --green magnitude output to DAC
blue : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) := (OTHERS => '0');
h_sync : OUT STD_LOGIC; --horiztonal sync pulse
v_sync : OUT STD_LOGIC;--vertical sync pulse
salfinal : OUT std_logic_vector(3 downto 0)
);
END TOP;
ARCHITECTURE arqTOP OF TOP IS
signal pix_clock: std_logic;
signal reloj: std_logic;
signal disp_ena: std_logic;
signal column: INTEGER;
signal row: INTEGER;
------
signal a,b : std_logic_vector(2 downto 0);
signal bus_A,bus_B: std_logic_vector(6 downto 0);
------
signal sal: std_logic_vector(3 downto 0);
signal disp0, disp1, disp2: std_logic_vector(6 downto 0);
BEGIN
ul: entity work.genMhz(arqgenMhz) port map(input_clk,pix_clock);
--ALU
r: entity work.relojlento(arqrelojlento) port map (input_clk, reloj);
cont: entity work.contador(arqcont) port map(reloj, reset,sal);
romData: entity work.romDatos(arqromDatos) port map ("111",'1',a);
romDatb: entity work.romDatos(arqromDatos) port map ("011",'1',b);
ALU: entity work.TOP_ALU(arq_alu) port map(a,b,sal(2 downto 0),sal(3),disp0,disp1,disp2,salfinal);
valorA: entity work.rom(arqrom) port map(a,'1',bus_A);
valorB: entity work.rom(arqrom) port map(b,'1',bus_B);
--DISPLAY
u2: entity work.vga_controller(behavior) port map(pix_clock, '1', h_sync, v_sync, disp_ena, column, row);
u3: entity work.hw_image_generator(behavior) port map(disp_ena, row, column, red, green, blue,
sal,a,b,bus_A, bus_b,disp0,disp1,disp2);
END ARCHITECTURE arqTOP;
Messages
100%
00:01:59

Pin planner


Pin Planner - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA
File Edit View Processing Tools Window Help
Groups
Named: *
Node Name Direction
blue[3..0] Output Group
green[3..0] Output Group
red[3..0] Output Group
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Top View - WireBond
MAX 10 - 10M50DAF484C7G
Pin Legend
Symbol Pin Type
User I/O
User assigned...
Fitter assigne...
Unbonded pad
Reserved pin
Other configu...
DEV_OE
DEV_CLR
DIFF_n
DIFF_p
Named: * Edit: Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard Reserved Current Strength Slew Rate Differential Pair
blue[3] Output PIN_N2 2 B2_N0 PIN_N2 2.5 V 12mA (default) 2 (default)
blue[2] Output PIN_P4 2 B2_N0 PIN_P4 2.5 V 12mA (default) 2 (default)
blue[1] Output PIN_T1 2 B2_N0 PIN_T1 2.5 V 12mA (default) 2 (default)
blue[0] Output PIN_P1 2 B2_N0 PIN_P1 2.5 V 12mA (default) 2 (default)
green[3] Output PIN_R1 2 B2_N0 PIN_R1 2.5 V 12mA (default) 2 (default)
green[2] Output PIN_R2 2 B2_N0 PIN_R2 2.5 V 12mA (default) 2 (default)
green[1] Output PIN_T2 2 B2_N0 PIN_T2 2.5 V 12mA (default) 2 (default)
green[0] Output PIN_W1 2 B2_N0 PIN_W1 2.5 V 12mA (default) 2 (default)
h_sync Output PIN_N3 2 B2_N0 PIN_N3 2.5 V 12mA (default) 2 (default)
input_clk Input PIN_P11 3 B3_N0 PIN_P11 2.5 V 12mA (default)
red[3] Output PIN_Y1 3 B3_N0 PIN_Y1 2.5 V 12mA (default) 2 (default)
red[2] Output PIN_Y2 3 B3_N0 PIN_Y2 2.5 V 12mA (default) 2 (default)
red[1] Output PIN_V1 2 B2_N0 PIN_V1 2.5 V 12mA (default) 2 (default)
red[0] Output PIN_AA1 3 B3_N0 PIN_AA1 2.5 V 12mA (default) 2 (default)
All Pins
0% 00:00:00


Pin Planner - C:/Users/Sergio/Desktop/VLSI/Práctica5/ALU_VGA/ALU_VGA - ALU_VGA
File Edit View Processing Tools Window Help
Groups
Named: *
Node Name Direction
blue[3..0] Output Group
green[3..0] Output Group
red[3..0] Output Group
Groups Report
Tasks
Early Pin Planning
Early Pin Planning...
Run I/O Assignment Analysis
Top View - WireBond
MAX 10 - 10M50DAF484C7G
Pin Legend
Symbol Pin Type
User I/O
User assigned...
Fitter assigne...
Unbonded pad
Reserved pin
Other configu...
DEV_OE
DEV_CLR
DIFF_n
DIFF_p
Named: * Edit: Filter: Pins: all
Node Name Direction Location I/O Bank VREF Group Fitter Location I/O Standard Reserved Current Strength Slew Rate Differential Pair
h_sync Output PIN_N3 2 B2_N0 PIN_N3 2.5 V 12mA (default) 2 (default)
input_clk Input PIN_P11 3 B3_N0 PIN_P11 2.5 V 12mA (default)
red[3] Output PIN_Y1 3 B3_N0 PIN_Y1 2.5 V 12mA (default) 2 (default)
red[2] Output PIN_Y2 3 B3_N0 PIN_Y2 2.5 V 12mA (default) 2 (default)
red[1] Output PIN_V1 2 B2_N0 PIN_V1 2.5 V 12mA (default) 2 (default)
red[0] Output PIN_AA1 3 B3_N0 PIN_AA1 2.5 V 12mA (default) 2 (default)
reset Input PIN_F15 7 B7_N0 PIN_F15 2.5 V 12mA (default)
salfinal[3] Output PIN_B10 7 B7_N0 PIN_B10 2.5 V 12mA (default) 2 (default)
salfinal[2] Output PIN_A10 7 B7_N0 PIN_A10 2.5 V 12mA (default) 2 (default)
salfinal[1] Output PIN_A9 7 B7_N0 PIN_A9 2.5 V 12mA (default) 2 (default)
salfinal[0] Output PIN_A8 7 B7_N0 PIN_A8 2.5 V 12mA (default) 2 (default)
v_sync Output PIN_N1 2 B2_N0 PIN_N1 2.5 V 12mA (default) 2 (default)
<<new node>>
All Pins
0% 00:00:00

RTL


Edit View Tools Window Help
Search altera.com
Page: 1 of 1
ALU_VGA:1
genMhz:u1
clk50mhz
clk25mhz
put_clk
reset
relojlento:r
clk1
led
contador:cont
clk
reset
sal[3..0]
romDatos:romData
3'h7 bus_dir[2..0]
1'h1 cs
bus_datos[2..0]
romDatos:romDatb
3'h3 bus_dir[2..0]
1'h1 cs
bus_datos[2..0]
vga_controller:u2
pixel_clk
1'h1 reset_n
column[31..0]
disp_ena
h_sync
row[31..0]
v_sync
TOP_ALU:ALU
a[2..0]
b[2..0]
cin
sel[2..0]
disp0[6..0]
disp1[6..0]
disp2[6..0]
salfinal[3..0]
rom:valorA
bus_dir[2..0]
1'h1 cs
bus_datos[6..0]
rom:valorB
bus_dir[2..0]
1'h1 cs
bus_datos[6..0]
hw_image_generator:u3
a[2..0]
b[2..0]
bus_A[6..0]
bus_b[6..0]
column[31..0]
disp_ena
disp0[6..0]
disp1[6..0]
disp2[6..0]
row[31..0]
sal[3..0]
blue[3..0]
green[3..0]
red[3..0]
h_sync
v_sync
blue[3..0]
green[3..0]
red[3..0]
salfinal[3..0]

PRUEBA


ALU
AUTOMATIZADA
Integrantes:
Kennedy Villa Carolina
Murrieta Villegas Alfonso
Reza Chavarria Sergio Gabriel
Valdespino Mendieta Joaquin

Conclusiones

- A partir del conjunto de componentes se puede realizar una gran cantidad de proyectos complejos para un propósito en especifico.
- Se pudo desarrollar la ALU automatizada con salida VGA gracias al uso de los conceptos vistos a lo largo del curso.

Referencias

- Fonseca E. (2020) VGA 3. Encontrado el 12 de noviembre de 2020, en https://youtu.be/dDPRDm7rY7Y
- Fonseca E. (2020) ULA. Encontrado el 12 de noviembre de 2020 en https://youtu.be/8vszGhGa1RI
- Fonseca E. (2016). *RESUMEN práctica VGA.* Recuperado el 12 de noviembre de 2020: https://bloglabdsd.files.wordpress.com/2015/02/resumen-practiva-vga.pdf