VGA Project Notes

<https://forum.digikey.com/t/vga-controller-vhdl/12794>

<https://github.com/turkmavisi/FPGA-Guitar-Hero/blob/master/Verilog%20HDL%20Source%20Code.v>

<https://stackoverflow.com/questions/26388004/vga-controller-with-vhdl>

<https://www.youtube.com/watch?v=eJMYVLPX0no>

### **Pixel Clock**

This VGA controller requires the user to provide the pixel clock. This can be brought into the FPGA on a dedicated clock pin or can be derived inside the FPGA using a PLL. In the example project for the DE2-115 development board, the available 50MHz clock is input into one of the Cyclone IV FPGA’s PLLs to produce a 193.16MHz pixel clock, as required by the 1920x1200, 60Hz VGA mode.

IMAGE GENERATOR

22

23 LIBRARY ieee;

24 USE ieee.std\_logic\_1164.all;

25

26 ENTITY hw\_image\_generator IS

27 GENERIC(

28 pixels\_y : INTEGER := 478; --row that first color will persist until

29 pixels\_x : INTEGER := 600); --column that first color will persist until

30 PORT(

31 disp\_ena : IN STD\_LOGIC; --display enable ('1' = display time, '0' = blanking time)

32 row : IN INTEGER; --row pixel coordinate

33 column : IN INTEGER; --column pixel coordinate

34 red : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0'); --red magnitude output to DAC

35 green : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0'); --green magnitude output to DAC

36 blue : OUT STD\_LOGIC\_VECTOR(7 DOWNTO 0) := (OTHERS => '0')); --blue magnitude output to DAC

37 END hw\_image\_generator;

38

39 ARCHITECTURE behavior OF hw\_image\_generator IS

40 BEGIN

41 PROCESS(disp\_ena, row, column)

42 BEGIN

43

44 IF(disp\_ena = '1') THEN --display time

45 IF(row < pixels\_y AND column < pixels\_x) THEN

46 red <= (OTHERS => '0');

47 green <= (OTHERS => '0');

48 blue <= (OTHERS => '1');

49 ELSE

50 red <= (OTHERS => '1');

51 green <= (OTHERS => '1');

52 blue <= (OTHERS => '0');

53 END IF;

54 ELSE --blanking time

55 red <= (OTHERS => '0');

56 green <= (OTHERS => '0');

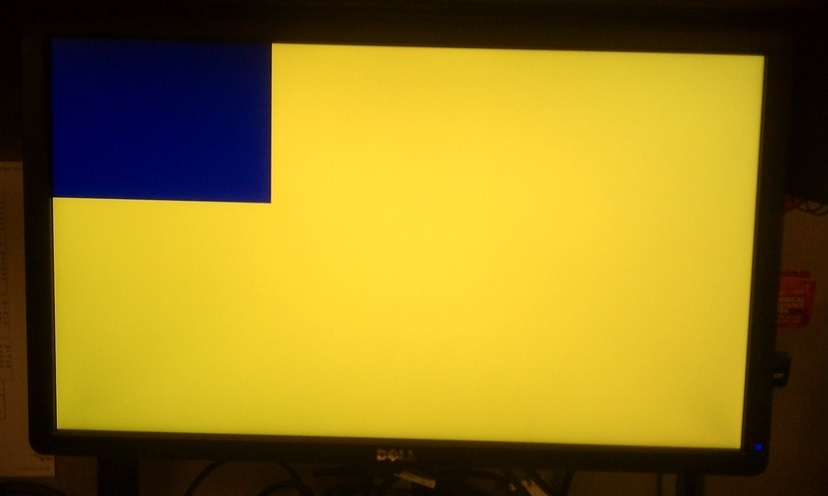
57 blue <= (OTHERS => '0');

58 END IF;

59

60 END PROCESS;

61 END behavior;



VGA VHDL

25 LIBRARY ieee;

26 USE ieee.std\_logic\_1164.all;

27

28 ENTITY vga\_controller IS

29 GENERIC(

30 h\_pulse : INTEGER := 208; --horiztonal sync pulse width in pixels

31 h\_bp : INTEGER := 336; --horiztonal back porch width in pixels

32 h\_pixels : INTEGER := 1920; --horiztonal display width in pixels

33 h\_fp : INTEGER := 128; --horiztonal front porch width in pixels

34 h\_pol : STD\_LOGIC := '0'; --horizontal sync pulse polarity (1 = positive, 0 = negative)

35 v\_pulse : INTEGER := 3; --vertical sync pulse width in rows

36 v\_bp : INTEGER := 38; --vertical back porch width in rows

37 v\_pixels : INTEGER := 1200; --vertical display width in rows

38 v\_fp : INTEGER := 1; --vertical front porch width in rows

39 v\_pol : STD\_LOGIC := '1'); --vertical sync pulse polarity (1 = positive, 0 = negative)

40 PORT(

41 pixel\_clk : IN STD\_LOGIC; --pixel clock at frequency of VGA mode being used

42 reset\_n : IN STD\_LOGIC; --active low asycnchronous reset

43 h\_sync : OUT STD\_LOGIC; --horiztonal sync pulse

44 v\_sync : OUT STD\_LOGIC; --vertical sync pulse

45 disp\_ena : OUT STD\_LOGIC; --display enable ('1' = display time, '0' = blanking time)

46 column : OUT INTEGER; --horizontal pixel coordinate

47 row : OUT INTEGER; --vertical pixel coordinate

48 n\_blank : OUT STD\_LOGIC; --direct blacking output to DAC

49 n\_sync : OUT STD\_LOGIC); --sync-on-green output to DAC

50 END vga\_controller;

51

52 ARCHITECTURE behavior OF vga\_controller IS

53 CONSTANT h\_period : INTEGER := h\_pulse + h\_bp + h\_pixels + h\_fp; --total number of pixel clocks in a row

54 CONSTANT v\_period : INTEGER := v\_pulse + v\_bp + v\_pixels + v\_fp; --total number of rows in column

55 BEGIN

56

57 n\_blank <= '1'; --no direct blanking

58 n\_sync <= '0'; --no sync on green

59

60 PROCESS(pixel\_clk, reset\_n)

61 VARIABLE h\_count : INTEGER RANGE 0 TO h\_period - 1 := 0; --horizontal counter (counts the columns)

62 VARIABLE v\_count : INTEGER RANGE 0 TO v\_period - 1 := 0; --vertical counter (counts the rows)

63 BEGIN

64

65 IF(reset\_n = '0') THEN --reset asserted

66 h\_count := 0; --reset horizontal counter

67 v\_count := 0; --reset vertical counter

68 h\_sync <= NOT h\_pol; --deassert horizontal sync

69 v\_sync <= NOT v\_pol; --deassert vertical sync

70 disp\_ena <= '0'; --disable display

71 column <= 0; --reset column pixel coordinate

72 row <= 0; --reset row pixel coordinate

73

74 ELSIF(pixel\_clk'EVENT AND pixel\_clk = '1') THEN

75

76 --counters

77 IF(h\_count < h\_period - 1) THEN --horizontal counter (pixels)

78 h\_count := h\_count + 1;

79 ELSE

80 h\_count := 0;

81 IF(v\_count < v\_period - 1) THEN --veritcal counter (rows)

82 v\_count := v\_count + 1;

83 ELSE

84 v\_count := 0;

85 END IF;

86 END IF;

87

88 --horizontal sync signal

89 IF(h\_count < h\_pixels + h\_fp OR h\_count >= h\_pixels + h\_fp + h\_pulse) THEN

90 h\_sync <= NOT h\_pol; --deassert horiztonal sync pulse

91 ELSE

92 h\_sync <= h\_pol; --assert horiztonal sync pulse

93 END IF;

94

95 --vertical sync signal

96 IF(v\_count < v\_pixels + v\_fp OR v\_count >= v\_pixels + v\_fp + v\_pulse) THEN

97 v\_sync <= NOT v\_pol; --deassert vertical sync pulse

98 ELSE

99 v\_sync <= v\_pol; --assert vertical sync pulse

100 END IF;

101

102 --set pixel coordinates

103 IF(h\_count < h\_pixels) THEN --horiztonal display time

104 column <= h\_count; --set horiztonal pixel coordinate

105 END IF;

106 IF(v\_count < v\_pixels) THEN --vertical display time

107 row <= v\_count; --set vertical pixel coordinate

108 END IF;

109

110 --set display enable output

111 IF(h\_count < h\_pixels AND v\_count < v\_pixels) THEN --display time

112 disp\_ena <= '1'; --enable display

113 ELSE --blanking time

114 disp\_ena <= '0'; --disable display

115 END IF;

116

117 END IF;

118 END PROCESS;

119

120 END behavior;

<https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf>

Page 17, 24 -> LCD High Score/Stars?