**Министерство образования и науки РФ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**Рязанский государственный радиотехнический университет**

**Кафедра радиотехнических систем**

Пояснительная записка к курсовой работе

на тему

**«Генератор тестовых видеосигналов»**

Выполнил: магистрант группы 511М

Проверил:

Холопов И.С.

**Министерство образования и науки РФ**

Федеральное государственное бюджетное образовательное учреждение высшего образования

**Рязанский государственный радиотехнический университет**

**Кафедра радиотехнических систем**

**ЗАДАНИЕ**

**на курсовую работу**

Студенту группа *511 М*

1. Тема проекта: *«Генератор тестовых видеосигналов»*

2. Срок сдачи законченного проекта: *10 июня 2016 г.*

3. Исходные данные к проекту:*генератор реализовать на ПЛИС EP2C20F484C7 фирмы «Altera» (семейство «Cyclone-II»).*

***Входные сигналы:***

*1)  сигнал задания режима работы генератора, принимаемый по протоколу RS-232:*

*- скорость передачи информации – 57 600 бит/с, частота посылок – 400 Гц;*

*- формат кода: Старт-байт – Информационные байты – Стоп-байт;*

*- количество информационных байт – 2, код режима работы передается в 1-м байте в 7..5 битах (старший бит кода – слева).*

*- количество стоповых бит в байте – 1; бит четности – отсутствует.*

*2) тактовые импульсы – 27 МГц.*

***Выходные сигналы:***

*аналоговые сигналы HSYNC, VSYNC, R, G и B по стандарту VGA для монитора с параметрами:*

*- частота кадров – 75 Гц; размер кадра – 800х600 пикселей;*

*- режимы работы генератора:*

*1) вертикальные цветные полосы;*

*2) сетчатое поле: линий по горизонтали – 8, линий по вертикали – 8 ,*

*ширина линии – 3 пикс. ;*

*3) шахматное поле: размер клетки – 100 пикс., цвет темной клетки – черный;*

*4) зеленое поле, 75 % яркости;*

*5) серая шкала, число градаций серого – 16;*

*6) горизонтальная штриховая мира с параметрами: штрихи 1, 3, 5 и 7 пикс.,*

*длина штриха – 5 % ширины экрана, расположение – по центру экрана.*

***Индикация:***

*отображение на семисегментном индикаторе номера режима работы генератора.*

*Питание – от источника постоянного тока 5 В.*

4. Содержание пояснительной записки:

*1) Введение.*

*2) Анализ технического задания.*

*3) Составление структурной схемы устройства и описание ее работы.*

*4) Разработка проекта на языке VHDL. Анализ быстродействия, затрат ресурсов ПЛИС, условий распространения сигнала.*

*5) Имитационное моделирование работы устройства средствами САПР Quartus II.*

*6) Составление электрической принципиальной схемы DC-DC преобразователя.*

*7) Заключение.*

*8) Библиографический список.*

5. Приложение: *перечень элементов СхЭ.*

6. Перечень графического материала: *структурная* *схема устройства – 1 л формата A1…A3.*

Дата выдачи задания: « 01 » марта 2016 г. Руководитель проекта \_\_\_\_\_\_\_\_\_\_

Задание принял к исполнению «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2016 г. Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание

|  |  |
| --- | --- |
| Техническое задание | 2 |
| Введение | 4 |
| Анализ технического задания | 5 |
| Составление структурной схемы и описание ее работы | 7 |
| Разработка проекта на языке VHDL. Анализ быстродействия, затрат ресурсов ПЛИС, условий распространения сигнала. | 10 |
| Имитационное моделирование работы устройства средствами САПР Quartus II | 33 |
| Составление электрической принципиальной схемы DC-DC преобразователя | 36 |
| Заключение | 37 |
| Библиографический список | 38 |
| Приложение | 39 |

Введение

Программируемые логические интегральные схемы (ПЛИС) представляют собой одно из самых быстро развивающихся направлений современной цифровой микроэлектроники. За последнее десятилетие наблюдается бурный рост рынка этих цифровых устройств и существенное улучшение их характеристик. В этой области на ближайшее время представляются самые оптимистичные прогнозы.

Привлекательность данной технологии заключается в предоставляемой конечному пользователю возможности быстрого создания цифровых устройств с произвольной внутренней структурой. Цикл разработки устройств на ПЛИС занимает меньшее время и значительно дешевле по сравнению со специализированными цифровыми микросхемами. Это связано с тем, что изменение принципиальной электрической схемы выполняется путем перепрограммирования одного и того же экземпляра микросхемы.

В настоящее время ведущими мировыми производителями ПЛИС являются фирмы Xilinx и Altera. Каждая из них выпускает целый спектр продукции, включая ПЛИС с различной архитектурой, ПЗУ для хранения конфигурации, системы автоматизированного проектирования (САПР), средства для программирования и отладки. При этом САПР минимальной конфигурации распространяется бесплатно, а ее возможности достаточны для разработки цифровых устройств начального и среднего уровня.

В данной курсовой работе разрабатывается генератор тестовых видеосигналов на ПЛИС фирмы Altera семейства "Cyclone-II", который формирует аналоговые сигналы вертикальной и горизонтальной синхронизации, а также сигналы красного, зеленого и синего цветов по стандарту VGA для 6 режимов работы монитора. Данное цифровое устройство предназначено для тестирования качества цветопередачи, разрешающей способности, а также правильности хода лучей горизонтальной и вертикальной развертки монитора.

Анализ технического задания

Согласно техническому заданию (ТЗ), генератор тестовых видеосигналов должен иметь 6 режимов работы, которые будут отличаться между собой аналоговыми сигналами R, G, B. При этом, выходные сигналы вертикальной и горизонтальной синхронизации для всех режимов работы неизменны. Таким образом, целесообразно будет сформировать различные модули генерирования сигналов цвета и синхронизации. Воспользовавшись [6], для заданных разрешения и частоты кадров монитора, найдем необходимое значение тактовой частоты устройства, а также длительности уровней в линиях синхронизации, выраженные в периодах тактового сигнала и сигнала горизонтальной синхронизации.

Сигнал задания режима работы генератора, должен быть принят по протоколу RS-232, в нашем случае с ПЭВМ. При этом формат кода, содержит 4 байта, из которых первый и последний являются словами байтовой синхронизации, по которым цифровое устройство определяет, что посылка адресована именно ему. Непосредственно код режима работы передается в 7,6 и 5 бите первого информационного байта, то есть во втором бите посылки, остальные могут быть зарезервированы для других цифровых устройств, работающих одновременно со спроектированным генератором по протоколу RS-232. Таким образом, необходимо сформировать модуль приема и декодирования сигнала по линии RS-232, который при этом выделяет код режима работы из информационной посылки только при условии совпадения старт-байта и стоп-байта указанным в техническом задании.

Выделенный из посылки код режима работы, должен быть использован в модуле генерации выходных сигналов цвета, чтобы каждому режиму соответствовала своя комбинация R, G, B. Кроме того, должна быть реализована индикация номера режима работы генератора, то есть код также должен поступать на семисегментный индикатор, где происходит его преобразование к виду, необходимому для отображения цифры режима работы.

Питание цифрового устройства, согласно техническому заданию должно обеспечиваться от источника постоянного тока 5 В. Так как для работы ПЛИС необходимо напряжение 3,3 В [4], необходимо использовать DC-DC стабилизатор-преобразователь напряжения и составить его схему включения.

Разработка вышеописанных модулей должны быть произведена на языке описания аппаратуры VHDL, а также необходимо произвести имитационное моделирование работы устройства, то есть показать наличие на выходе необходимых сигналов горизонтальной и вертикальной синхронизации, а также цветности. Вышеуказанные операции выполняются средствами САПР Quartus II, которая предназначена для работы с ПЛИС фирмы Altera.

Составление структурной схемы и описание ее работы

В ходе анализа технического задания были предложены следующие модули разрабатываемого цифрового устройства:

* модуль генерирования сигналов вертикальной и горизонтальной синхронизации для разрешения 800x600 и кадровой частоты 75 Гц (VGA\_Timer), который кроме вышеуказанных сигналов также формирует сигналы текущей строки и столбца, а также сигнал разрешения передачи видеоизображения;
* модуль генерирования видеоизображения (VGA\_Video\_Generator), который, используя служебные сигналы с выхода VGA\_Timer, а именно указатель строки и столбца, разрешение передачи видеоизображения, а также код режима работы, формирует необходимые для получения на мониторе изображений согласно техническому заданию для заданного режима;
* модуль UART-приемника, который осуществляет прием информационной посылки по протоколу RS-232, содержащей код режима работы генератора, при этом в нем должна быть реализована синхронизация по стартовому и стоповому байтам, указанным в техническом задании (A5 — стартовый байт, 7E — стоповый байт);
* семисегментный индикатор, входным сигналом, которого является код режима работы, состоящий из 3 бит, а выходным — последовательность из 7 бит, необходимая для отображения соответствующего числа;
* модуль фазовой автоподстройки частоты, необходимый для преобразования частоты 27 МГц кварцевого генератора на ПЛИС в тактовую частоту цифрового устройства 49,5 МГц, соответствующую по стандарту VGA размеру кадра 800x600 и кадровой частоте 75 Гц;
* делитель частоты, необходимый для работы UART-приемника, работающего со скоростью передачи информации 57 600 бит/c.

Для задания режимов работы используется формирователь и передатчик сигналов по протоколу RS-232 в составе ПЭВМ. Выходные сигналы цифрового устройства поступают на монитор стандарта VGA.

Структурная схема приведена в виде графического материала на листе формата А3. Работа схемы описывается следующим образом. На ПЭВМ пользователь, задает информационную посылку следующего вида 0хA5XYYY7E, где A5 — стартовый байт, 7E — стоповый байт, согласно ТЗ, Y — зарезервированные для других устройств, работающих по RS-232 последовательности длинной 4 бита, в нашем случае, вместо Y можно передавать последовательность вида "0000". X — тетрада задающая режим работы генератора видеосигнала, возможные комбинации приведены в таблице 1:

Таблица 1 — Последовательности, задающие режим работы генератора

|  |  |  |
| --- | --- | --- |
| Двоичная комбинация | Номер режима | Изображение на мониторе |
| 0010 | 1 | вертикальные цветные полосы |
| 0100 | 2 | сетчатое поле |
| 0110 | 3 | шахматное поле |
| 1000 | 4 | зеленое поле |
| 1010 | 5 | серая шкала |
| 1100 | 6 | горизонтальная штриховая мира |

Сформированный сигнал последовательно передается на модуль UART-приемника, где происходит сравнение первого и последнего байта посылки, и, если они соответствуют необходимым осуществляется выделение кода режима работы (сигнал mode), состоящего их первых 3 бит тетрады X, который поступает на модуль VGA\_Video\_Generator для формирования сигналов R, G, B, а также на семисегментный индикатор (Segment7). Тактовая частота кварцевого генератора ПЛИС равная 27 МГц (сигнал clock) преобразуется с помощью генератора с фазовой автоподстройкой частоты (ФАПЧ) в частоту 49,5 МГц (сигнал clk), соответствующую тактовой частоте монитора при указанных в ТЗ формате и частоте кадра. Сигнал clk является тактовым для модулей генерации синхросигналов и видеосигналов (VGA\_Timer и VGA\_Video\_Generator соответственно). В модуле VGA\_Timer формируются сигналы hsync и vsync горизонтальной и вертикальной синхронизации, сигнал разрешения передачи видеоизображения video\_on, а также служебные сигналы указатели для модуля VGA\_Video\_Generator о текущем номере строки и столбца (row и column). Логика работы VGA\_Video\_Generator следующая: если video\_on равен 1, то есть передача видео разрешена, на выходе формируются сигналы R, G, B (представляющие собой последовательность 4 бит) в соответствии с управляющим сигналом mode, если передача видео запрещена video\_on равен 0, то сигналы цветности равны нулю. Управляющий сигнал mode также подается на семисегментный индикатор Segment7, на выходе которого последовательность из 7 бит, соответствующая необходимой цифре режима работы согласно таблице 1.

Разработка проекта на языке VHDL. Анализ быстродействия, затрат ресурсов ПЛИС, условий распространения сигнала.

При использовании структурного стиля программирования проект на языке VHDL, реализующий генератор тестовых видеосигналов, содержит 5 компонентов: UART-приемник (rs232) [2], генератор с ФАПЧ (pll), семисегментный индикатор (segment7), формирователь сигналов вертикальной и горизонтальной синхронизации (VGA\_Timer) и сигналов цветности (VGA\_Video\_Generator). Программа на языке VHDL, реализующая компонент верхнего уровня приведена ниже [1]:

-- Подключение библиотек и пакета констант

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use work.my\_parameters.all;

-- Объявление объекта и описание портов

Entity VGA is

Port (clock : in std\_logic; -- тактовый сигнал 27 МГц

hsync : out std\_logic;

rx : in std\_logic;

tx : out std\_logic;

vsync : out std\_logic;

R,G,B : out std\_logic\_vector (3 downto 0);

indicator : out std\_logic\_vector(0 to 6);

indicatorP : out std\_logic\_vector(0 to 6);

modeforLED : out std\_logic\_vector (2 downto 0);

indicator1 : out std\_logic\_vector(0 to 6);

indicator2 : out std\_logic\_vector(0 to 6));

end VGA;

-- Объявление архитектуры

architecture behavior of VGA is

Component rs232 is -- Приемник UART

Generic (N: natural := 4;

R: natural := 57600);

Port (clock :in std\_logic;

rx :in std\_logic;

tx :out std\_logic;

modeforLED :out std\_logic\_vector (2 downto 0);

mode :out std\_logic\_vector (2 downto 0)

);

end component;

Component segment7 is -- Семисегментный индикатор

port

(

mode :in std\_logic\_vector(2 downto 0);

indicator :out std\_logic\_vector(0 to 6)

);

end component;

Component pll is -- Генератор с ФАПЧ

PORT

(

inclk0 : IN STD\_LOGIC;

c0 : OUT STD\_LOGIC

);

end component;

Component VGA\_Timer is -- Формирователь синхросигналов

Port

(clk : in std\_logic;

hsync : out std\_logic;

vsync : out std\_logic;

row : out natural range 0 to 2000;

column : out natural range 0 to 2000;

video\_on : out std\_logic);

end component;

Component VGA\_Video\_Generator is -- Формирователь видеосигнала

Port (clk : in std\_logic;

row : in natural range 0 to 2000;

column : in natural range 0 to 2000;

video\_on : in std\_logic;

R,G,B : out std\_logic\_vector (3 downto 0);

mode : in std\_logic\_vector (2 downto 0));

end component;

signal clk: std\_logic; --Внутренний тактовый сигнал

signal row: natural range 0 to 2000;

signal column: natural range 0 to 2000; -- Указатели текущей строки и столбца

signal video\_on: std\_logic; -- Сигнал разрешения передачи видео

signal mode: std\_logic\_vector(2 downto 0); -- Сигнал режима работы

begin

indicator1 <= "1111111";

indicator2 <= "1111111";

indicatorP <= "0011000";

-- Подключение компонентов

pll\_1: pll port map (clock,clk);

rxrs232: rs232 generic map (N => 4, R => 57600) port map (clock, rx, tx, modeforLED, mode);

ind: segment7 port map (mode, indicator);

Timer\_2: VGA\_Timer port map

(clk => clk,

hsync => hsync,

vsync => vsync,

row => row,

column => column,

video\_on=> video\_on);

Generator\_3: VGA\_Video\_Generator port map

(clk => clk,

row => row,

column => column,

video\_on => video\_on,

R =>R,

G =>G,

B =>B,

mode => mode);

end behavior;

UART-приемник реализуется в виде цифрового конечного автомата (ЦКА) с 4 состояниями: sleep (ожидание), start (начало приема), drx (прием посылки) и stop (окончание приема посылки). Архитектура такого ЦКА описывается 7 процессами, а программа на языке VHDL приведена ниже:

-- Пакет констант

package my\_parameters232 is

type RS232\_states is (sleep, start, drx, stop); -- Состояние ЦКА

end my\_parameters232;

-- Подключение библиотек и пакета констант

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use work.my\_parameters232.all;

entity rs232 is

Generic (N: natural := 4; -- Число принимаемых байт данных

R: natural := 57600); -- Скорость передачи информации

Port (clock :in std\_logic;

rx :in std\_logic;

tx :out std\_logic;

mode :out std\_logic\_vector (2 downto 0);

modeforLED :out std\_logic\_vector (2 downto 0)

);

end rs232;

architecture behavior of rs232 is

Component divider is

generic (N: natural := 12);

port

(clock : in std\_logic;

go : in std\_logic;

clk : out std\_logic);

end component;

type RX\_DATA is array (0 to 3) of std\_logic\_vector (0 to 7);--Массив принятых байт

signal n\_bit: natural range 0 to 7; -- Для счетчика бит

signal clk: std\_logic; -- Тактовый сигнал, сигнал счета бит

signal n\_byte: natural range 0 to N; -- Для счетчика байт

signal cnt: std\_logic; -- Для сброса счетчика бит

signal recieve: std\_logic; -- Для подтверждения прихода старт-бита

signal go: std\_logic; -- Для определения момента приема старт-бита

signal state,next\_state:RS232\_states; --Текущее и следующее состояние ЦКА

signal period: std\_logic; -- Для сброса счетчика байт

signal sent: std\_logic; -- Сигнал счета байт

Begin

m1: divider generic map (N =>27000000/R) port map (clock,go,clk);

tx <='1'; -- Для обеспечения штатного режима работы приемопередатчика ПЭВМ

process(rx, state) -- Процесс определения момента приема старт-бита

begin

if(state /=sleep and state/=stop) then go <='0';

elsif rx'event and rx='0' then

go <='1';

end if;

end process;

process(clk) -- Процесс поиска первого байта

variable cycle: natural;

begin

if (clk'event and clk ='1') then

if (rx ='1') then

if cycle > 10

then period <='0';

cycle:=0;

else period<='1';

cycle:= cycle+1;

end if;

else cycle:=0;

end if;

end if;

end process;

process (clk, go) -- Процесс смены состояний ЦКА

begin

if go ='1' then state <= start;

elsif (clk'event and clk='0') then

state<= next\_state;

end if;

end process;

-- Процесс, описывающий формирование выходных сигналов

process (state,clk,n\_bit,n\_byte)

variable buf: RX\_data;

variable RTF: std\_logic\_vector(2 downto 0);

begin

if (clk'event and clk='1') then

case state is

when sleep => cnt <='0';

sent <='0';

when start => cnt <='0';

sent <='0';

recieve <= rx;

when drx => cnt <='1';

sent <='0';

buf(n\_byte)(n\_bit) := rx;

if (n\_bit = 7) then cnt <='0'; end if;

when stop => cnt <='0';

sent <='1';

if (n\_byte = N-1) then if (buf(0)=x"A5" and buf(N-1)=x"7E") then

RTF:= buf(1)(7)&buf(1)(6)&buf(1)(5);

else RTF :="000";

end if;

modeforLED <= RTF;

mode <= RTF;

end if;

when others => null;

end case;

end if;

end process;

process (state, recieve, n\_bit) -- Процесс нахождения след. состояния

begin

case state is

when sleep => next\_state<= sleep;

when start => if recieve='0' then next\_state <= drx;

else next\_state <= sleep;

end if;

when drx => if n\_bit = 7

then next\_state <= stop;

else next\_state <= drx;

end if;

when stop => next\_state<=sleep;

when others => null;

end case;

end process;

process (clk, cnt) -- Счетчик бит

begin

if clk'event and clk='0'

then

if cnt='0'

then n\_bit <= 0;

else n\_bit<=n\_bit+1;

end if;

end if;

end process;

process (sent, period) -- Счетчик байт

begin

if period ='0'

then n\_byte <=0;

elsif sent'event and sent ='1'

then n\_byte <= n\_byte+1;

end if;

end process;

end behavior;

Для работы UART-приемника для приема информации со скоростью 57 600 бит/c необходим делитель на выходе которого будет частота 57 600 Гц, для этого частоту кварцевого генератора необходимо разделить на 469, настроечная константа делителя равна 27000000/57600. Делитель представляет собой счетчик до коэффициента деления N, причем первую половину периода на выходе формируется низкий логический уровень, а вторую — высокий. Также в делителе предусмотрен асинхронный сброс по вспомогательному сигналу go. Текст программы приведен ниже:

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

Entity divider is

generic (N: natural := 12);

port

(clock : in std\_logic;

go : in std\_logic;

clk : out std\_logic); --выходной сигнал

end divider;

architecture Behavior\_divider of divider is

begin

process(clock, go)

variable cnt1: natural; --счетная переменная

begin

if go ='1' then cnt1:=0; clk <= '0';

elsif clock'event and clock = '1' then

if cnt1 < N/2 then

clk <= '0';

else

clk <= '1';

end if;

if cnt1 = N-1 then cnt1 := 0;

else cnt1 := cnt1+1;

end if;

end if;

end process;

end Behavior\_divider;

Для работы семисегментного индикатора необходимо сформировать на его входе управляющую последовательность из 7 бит, при этом низкий логический уровень, приведет к включению сегмента [1]. Для удобства работы на первые два индикатора подаются управляющие последовательности "1111111", полностью выключающие их, на третий индикатор — последовательность "0011000", которая соответствует букве "Р — Режим". Четвертый индикатор преобразует 3 бита режима работы в последовательности для отображения цифр от 1 до 6. Текст программы приведен ниже:

library ieee;

use ieee.std\_logic\_1164.all;

entity segment7 is

port

(

mode: in std\_logic\_vector(2 downto 0);

indicator: out std\_logic\_vector(0 to 6)

);

end segment7;

architecture Behavior of segment7 is

begin

process(mode)

begin

case mode is

when "001" =>indicator<="1001111";

when "010" =>indicator<="0010010";

when "011" =>indicator<="0000110";

when "100" =>indicator<="1001100";

when "101" =>indicator<="0100100";

when "110" =>indicator<="0100000";

when others => null;

end case;

end process;

end Behavior;

Модуль формирования синхросигналов содержит 2 процесса, а именно, подсчет количества периодов тактового сигнала и вывод сигнала горизонтальной синхронизации, а также подсчет периодов сигнала горизонтальной синхронизации и вывод сигнала вертикальной синхронизации [3]. Для нахождения констант, определяющих границы интервалов для сигналов синхронизации, использованы материалы из [6]. Искомые значения интервалов приведены в таблице 2.

Таблица 2 — Границы интервалов сигналов синхронизации

|  |  |  |  |
| --- | --- | --- | --- |
| Горизонтальная синхронизация | | Вертикальная синхронизация | |
| Тип интервала | Периоды |  | Периоды |
| Видимая область | 800 | Видимая область | 600 |
| Передний защитный интервал | 16 | Передний защитный интервал | 1 |
| Импульс синхронизации | 80 | Импульс синхронизации | 3 |
| Задний защитный интервал | 160 | Задний защитный интервал | 21 |
| Весь синхросигнал | 1056 | Весь синхросигнал | 625 |

При этом, под периодами в таблице понимается: для горизонтальной синхронизации, число периодов тактового сигнала, а для вертикальной синхронизации — число периодов сигнала горизонтальной синхронизации .

Кроме этого формируется сигнал разрешения передачи видеосигнала, путем применения операции "И" над сигналами разрешения передачи видеосигнала по горизонтали и по вертикали. Текст программы приведен ниже:

-- Пакет констант

package my\_parameters is

type VGA\_params is

record

Video: INTEGER range 0 to 2000;

FP: INTEGER range 0 to 2000;

Sync: INTEGER range 0 to 2000;

BP: INTEGER range 0 to 2000;

end record;

Constant H\_TIMING: VGA\_params := (800, 800+16, 800+16+80, 800+16+80+160);

Constant V\_TIMING: VGA\_params := (600, 600+1, 600+1+3, 600+1+3+21);

--Constant H\_TIMING: VGA\_params := (12, 14, 16 ,18); настройки для имита-

--Constant V\_TIMING: VGA\_params := (6, 8, 10, 12); ционного моделирования

end my\_parameters;

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

use work.my\_parameters.all;

Entity VGA\_Timer is

Port

(clk : in std\_logic;

hsync : out std\_logic;

vsync : out std\_logic;

row : out natural range 0 to 2000;

column : out natural range 0 to 2000;

video\_on : out std\_logic);

end VGA\_Timer;

architecture behavior of VGA\_Timer is

signal videov,videoh : std\_logic;

Constant H: VGA\_params := H\_TIMING;

Constant V: VGA\_params := V\_TIMING;

signal vcount : natural range 0 to 2000;

begin

video\_on <= videoh and videov;

process (clk) -- Для сигналов горизонтальной синхронизации

variable hcount:natural range 0 to 2000 :=0;

begin

if clk'event and clk='1' then

if (hcount>=0 and hcount<H.Video) then

videoh <= '1';

column <= hcount;

hsync <='1';

hcount:=hcount+1;

elsif hcount>=H.Video and hcount<H.FP then

videoh <= '0';

column <= 0;

hsync <='1';

hcount:=hcount+1;

elsif hcount>=H.FP and hcount<H.Sync then

videoh <= '0';

column <= 0;

hsync <='0';

hcount:=hcount+1;

elsif hcount>=H.Sync and hcount<H.BP then

videoh <= '0';

column <= 0;

hsync <='1';

hcount:=hcount+1;

end if;

if hcount = H.BP then hcount:=0;

if vcount=V.BP-1 then vcount <= 0;

else

vcount <= vcount+1;

end if;

end if;

end if;

end process;

process (vcount,clk) -- Для сигналов вертикальной синхронизации

begin

if clk'event and clk='1' then

if vcount>=0 and vcount<(V.Video) then

videov <= '1';

row <= vcount;

vsync <='1';

elsif vcount>=V.Video and vcount<V.FP then

videov <= '0';

row <= 0;

vsync <='1';

elsif vcount>=V.FP and vcount<V.Sync then

videov <= '0';

row <= 0;

vsync <='0';

elsif vcount>=V.Sync and vcount<V.BP then

videov <= '0';

row <= 0;

vsync <='1';

end if;

end if;

end process;

end behavior;

Модуль формирования сигнала цветности содержит процесс формирования сигналов R, G, B, в зависимости от сигнала mode. При этом используется оператор множественного выбора case [3]. Программа на языке VHDL имеет следующий вид:

LIBRARY ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

use ieee.std\_logic\_arith.all;

Entity VGA\_Video\_Generator is

Port (clk : in std\_logic;

row : in natural range 0 to 2000;

column : in natural range 0 to 2000;

video\_on : in std\_logic;

mode : in std\_logic\_vector (2 downto 0);

R,G,B : out std\_logic\_vector (3 downto 0));

end VGA\_Video\_Generator;

architecture behavior of VGA\_Video\_Generator is

Constant screen\_wide: natural := 800;

Constant screen\_height: natural := 600;

begin

process (clk, video\_on, column, row, mode)

variable R1: std\_logic\_vector(3 downto 0);

variable G1: std\_logic\_vector(3 downto 0);

variable B1: std\_logic\_vector(3 downto 0);

variable R2: std\_logic\_vector(3 downto 0);

variable G2: std\_logic\_vector(3 downto 0);

variable B2: std\_logic\_vector(3 downto 0);

begin

if video\_on='0' then

R <="0000";

G <="0000";

B <="0000";

elsif clk'event and clk='1' then

case mode is

when "001" =>

case column/(screen\_wide/8) is

when 0 => R <="1111";

G <="1111";

B <="1111";

when 1 => R <="1000";

G <="1000";

B <="0000";

when 2 => R <="0000";

G <="1000";

B <="1000";

when 3 => R <="0000";

G <="1000";

B <="0000";

when 4 => R <="1000";

G <="0000";

B <="1000";

when 5 => R <="1000";

G <="0000";

B <="0000";

when 6 => R <="0000";

G <="0000";

B <="1000";

when 7 => R <="0000";

G <="0000";

B <="0000";

when others => null;

end case;

when "010" =>

case column rem (screen\_wide/8) is

when 0 to 1 =>

R1 :="1111";

G1 :="1111";

B1 :="1111";

when others =>

R1 :="0000";

G1 :="0000";

B1 :="0000";

end case;

case row rem (screen\_height/8) is

when 0 to 2 =>

R2 :="1111";

G2 :="1111";

B2 :="1111";

when others =>

R2 :="0000";

G2 :="0000";

B2 :="0000";

--when others => null;

end case;

R <= R1 or R2;

G <= G1 or G2;

B <= B1 or B2;

when "011" =>

if ((column/80) + (row/60)) rem 2 = 0 then R <="1111";

G <="1111";

B <="1111";

else

R <="0000";

G <="0000";

B <="0000";

end if;

when "100" =>

R <="0000";

G <="1100";

B <="0000";

when "101" => case column/(screen\_wide/16) is

when 0 => R <="1111";

G <="1111";

B <="1111";

when 1 => R <="1110";

G <="1110";

B <="1110";

when 2 => R <="1101";

G <="1101";

B <="1101";

when 3 => R <="1100";

G <="1100";

B <="1100";

when 4 => R <="1011";

G <="1011";

B <="1011";

when 5 => R <="1010";

G <="1010";

B <="1010";

when 6 => R <="1001";

G <="1001";

B <="1001";

when 7 => R <="1000";

G <="1000";

B <="1000";

when 8 => R <="0111";

G <="0111";

B <="0111";

when 9 => R <="0110";

G <="0110";

B <="0110";

when 10 => R <="0101";

G <="0101";

B <="0101";

when 11 => R <="0100";

G <="0100";

B <="0100";

when 12 => R <="0011";

G <="0011";

B <="0011";

when 13 => R <="0010";

G <="0010";

B <="0010";

when 14 => R <="0001";

G <="0001";

B <="0001";

when 15 => R <="0000";

G <="0000";

B <="0000";

when others => null;

end case;

when "110" =>

R <="1111";

G <="1111";

B <="1111";

if column > (screen\_wide/2 - screen\_wide/20) and column < (screen\_wide/2 + screen\_wide/20) then

case row is

when screen\_height/2 =>

R <="0000";

G <="0000";

B <="0000";

when screen\_height/2+2 to screen\_height/2+5=>

R <="0000";

G <="0000";

B <="0000";

when screen\_height/2+8 to screen\_height/2+13 =>

R <="0000";

G <="0000";

B <="0000";

when screen\_height/2+18 to screen\_height/2+25 =>

R <="0000";

G <="0000";

B <="0000";

when others => null;

end case;

end if;

when others => null;

end case;

end if;

end process;

end behavior;

Сигналы цвета формируются в цифровом виде, после чего поступают на ЦАП, который представлен матрицей R-2R. При этом чем больше двоичное число, тем выше будет амплитуда аналогового сигнала и соответственно выше яркость этого цветового компонента.

Цифровое устройство использует 48 выводов ПЛИС, возможно уменьшение числа используемых выводов до 24 при необходимости (при этом убрав подачу сигналов на первые 3 индикатора и дополнительную индикацию на 3 красных светодиодах). Высокое быстродействие устройства достигается самой архитектурой ПЛИС, процессы выполняются параллельно, принятый код режима работы поступает на другие модули без задержек. Использование сигналов указателей номера строки и столбца обеспечивает соответствие номера пикселя синхросигнала и сигналов цвета.

Имитационное моделирование работы устройства средствами САПР Quartus II

В ходе имитационного моделирования проверяется правильность формирования синхросигналов и сигналов цветности, а также проверка работы устройства путем отправления с помощью программы HTerm на ПЭВМ необходимой посылки по протоколу RS-232 и визуального наблюдения изображения на мониторе [5].

Для моделирования временно изменим константу модуля VGA\_Timer, на меньшие значения, чтобы была возможность увидеть целиком импульс вертикальной синхронизации. Результат моделирования формирователя синхросигналов приведен на рисунке 1.

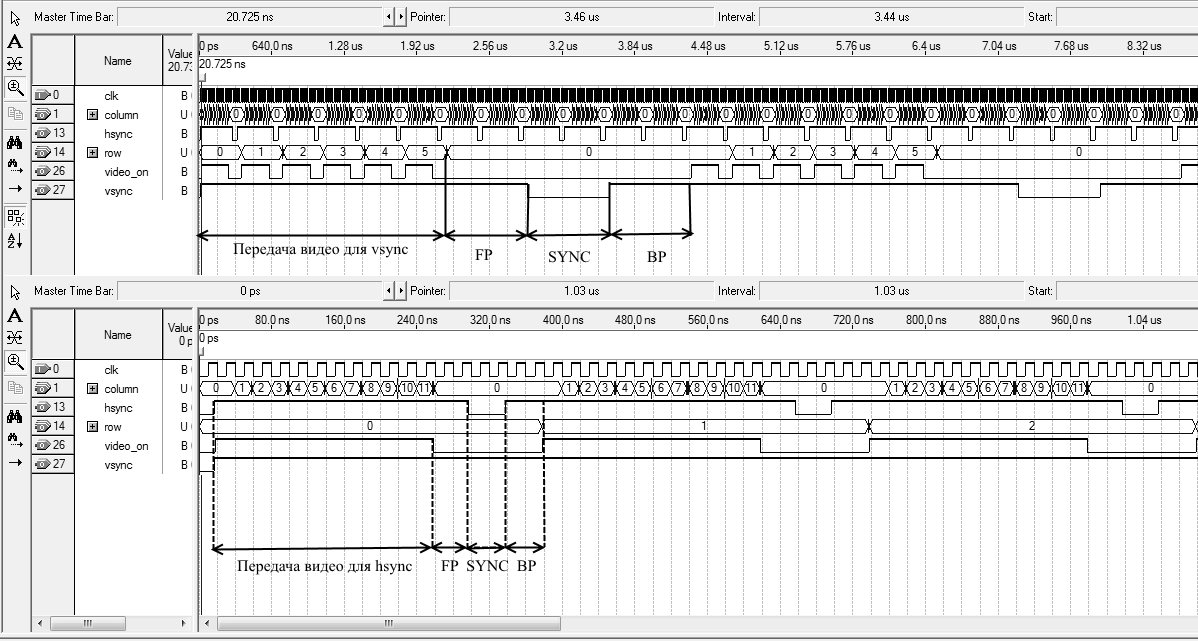


Рисунок 1 — Результат моделирования работы модуля VGA\_Timer

На рисунке видно, что синхросигналы соответствуют стандарту VGA то есть имеют интервал передачи видео, интервал синхронизации (SYNC), а также защитные интервалы (передний — FP, задний — BP). При этом, производится правильный подсчет числа столбцов по заднему фронту тактового сигнала (в примере их 12) и строк по переднему фронту сигнала горизонтальной синхронизации (в примере их 6). Также видно, что сигнал разрешения передачи видео (video\_on) есть операция "И" над сигнала разрешения передачи по горизонтали и по вертикали.

Имитационное моделирование работы модуля формирования сигналов цвета произведем для первого режима работы (вертикальные цветные полосы). Ширину полосы в данном случае уменьшим до 10 пикселей, порядок следования цветов: белый, желтый, голубой, зеленый, пурпурный, красный, синий, черный. Результат моделирования приведен на рисунке 2.

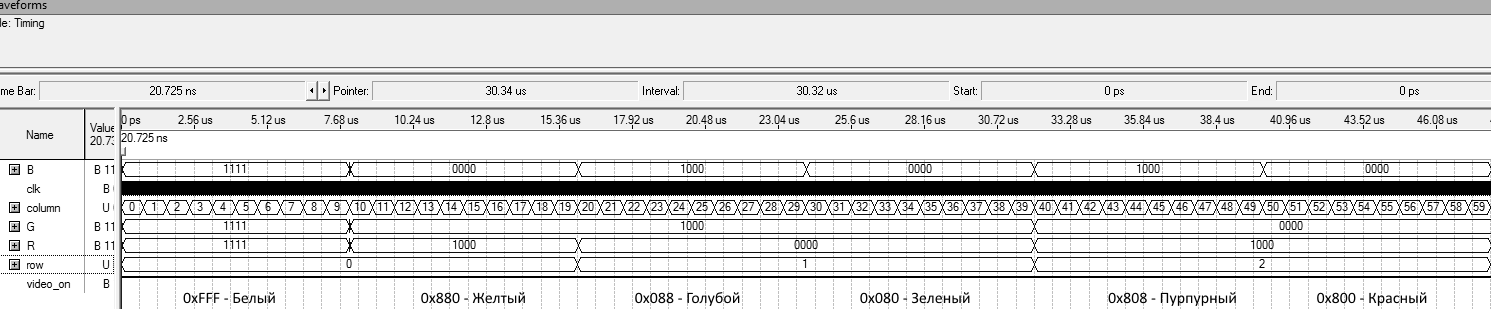


Рисунок 2 — Результат моделирования работы модуля VGA\_Video\_Generator

На рисунке видно, что каждый десятый столбец изменяются сигналы R, G, B. При этом внизу подписан эквивалент в шестнадцатеричном виде, где первый символ обозначает красный, второй — зеленый, третий — синий цвет. В данном случае яркость цветов, кроме белого и черного будет составлять 50%.

Для проверки работы всего устройства используется программа HTerm, в которой указывается номер COM-порта, скорость передачи информации, количество информационных бит в байте, число стоповых байт, при необходимости бит четности. Затем на панели Input control выбирается тип записи "HEX", то есть шестнадцатеричная запись и записывается посылка, как показано на рисунке 3. Далее открывается заданный COM-port нажатием кнопки Connect, после чего нажатием кнопки ASend осуществляется передача заданного сообщения. Проверка работоспособности произведена для двух режимов работы: режим 3 — шахматное поле, режим 6 — штриховая мира.

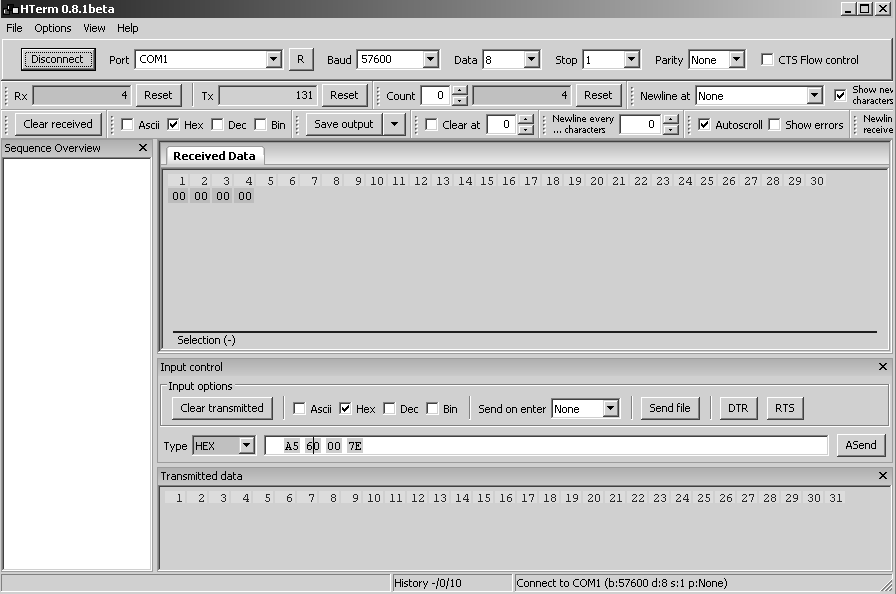


Рисунок 3 — Включение 3 режима генератора с помощью программы HTerm

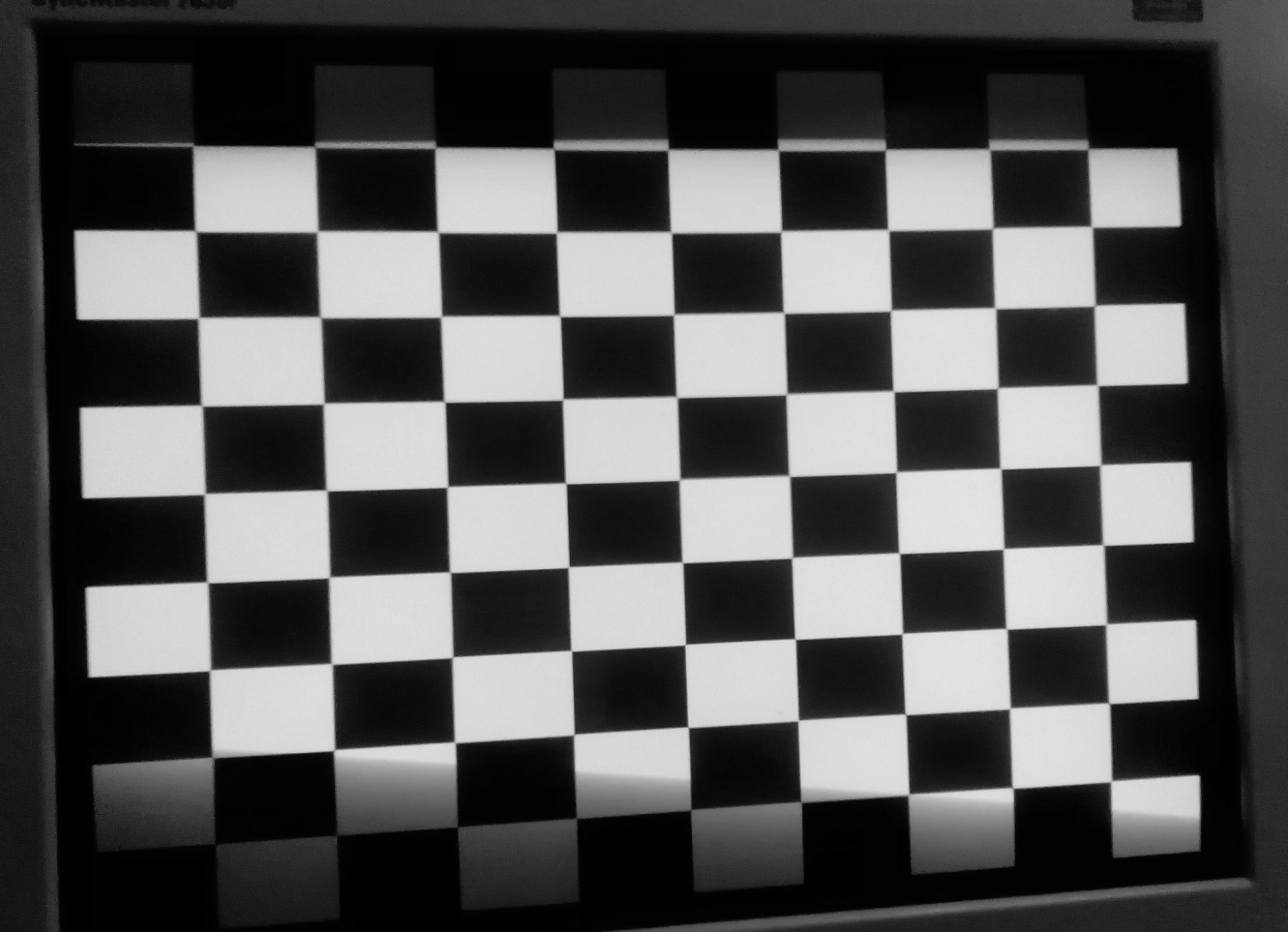


Рисунок 4 — Изображение на мониторе при включении 3 режима работы

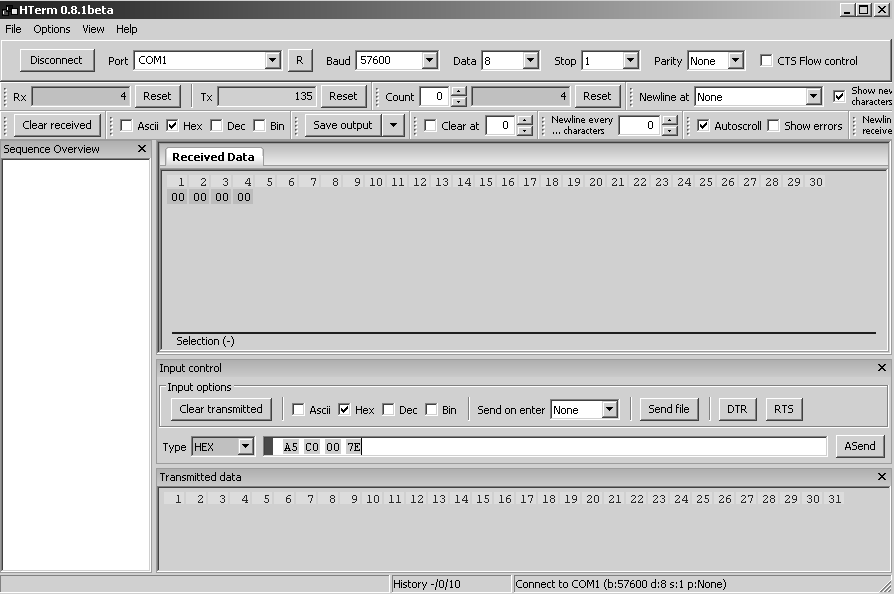


Рисунок 5 — Включение 6 режима генератора с помощью программы HTerm

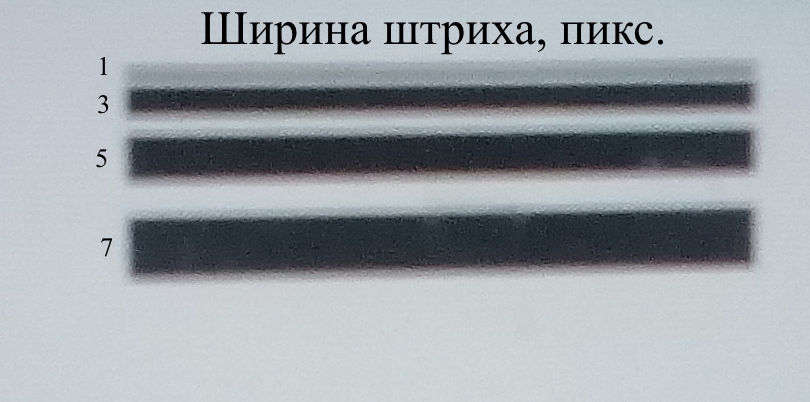


Рисунок 6 — Изображение на мониторе при включении 6 режима работы

В режиме 6 — штриховая мира интервал между штрихами, равен ширине предыдущего штриха при движении сверху вниз. Нечеткость картинки вызвана плохим качеством съемки, а не работой монитора.

Составление электрической принципиальной схемы DC-DC преобразователя

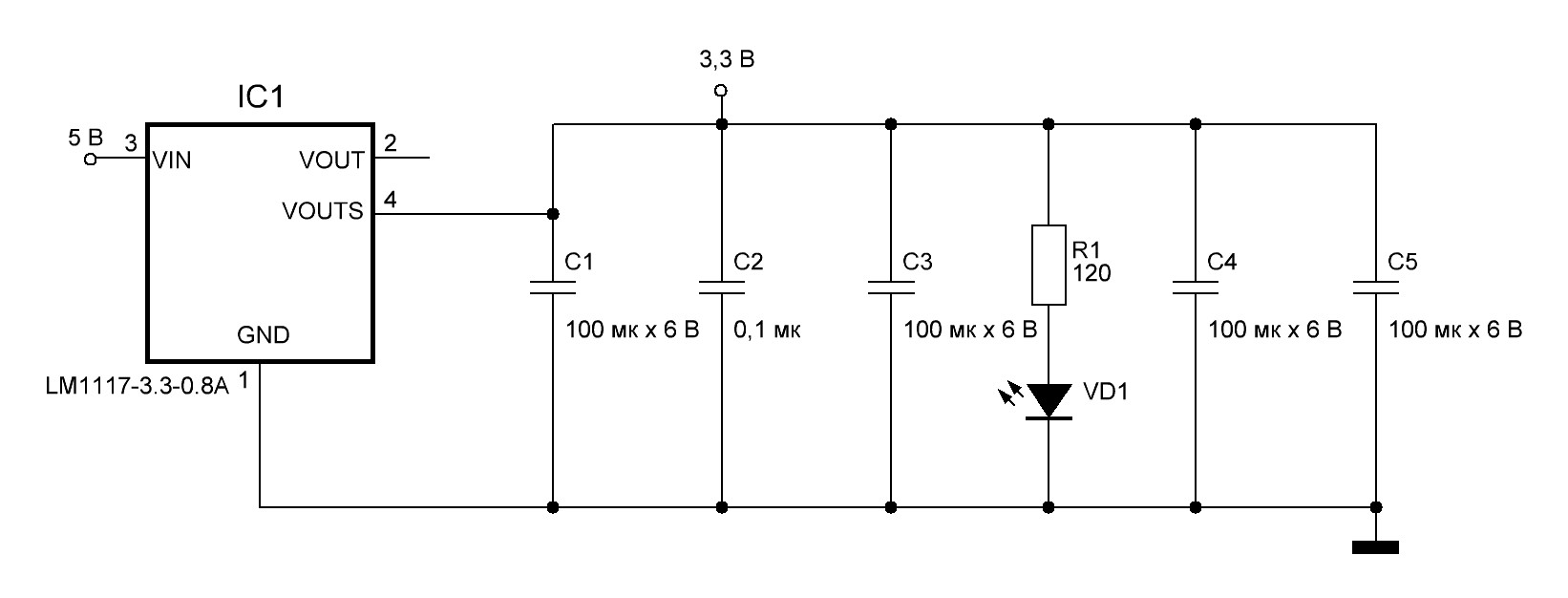
Для обеспечения необходимого режима работы ПЛИС по напряжению, необходимо преобразовать входное напряжение источника постоянного тока 5 В в напряжение 3,3 В. Для этого будем использовать микросхему LM1117-3.3-0.8A. Принципиальная схема включения стабилизатора напряжения на ИМС приведена на рисунке7. 

Рисунок 7 — Схема электрическая принципиальная

Заключение

В данной курсовой работе было спроектировано цифровое устройство, генерирующее тестовые видеосигналы, а также сигналы синхронизации, согласно стандарту VGA для размера кадра 800х600 и кадровой частоты 75 Гц. Разработаны программы на языке описания цифровых устройств VHDL реализующие определение режима работы, который задается с помощью ПЭВМ, генерации необходимых выходных сигналов. Произведено моделирование работы отдельных модулей программного проекта, а также всего устройства в целом. Составлена структурная схема экспериментального стенда, включающего в себя спроектированный генератор тестовых видеосигналов. Программные компоненты могут быть перенастроены для работы с другими исходными данными, например с другой частотой и размерами кадра, что обеспечивает универсальность использования.

Библиографический список

1. Основы проектирования цифровых устройств с использованием языка описания аппаратуры VHDL: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. И.С. Холопов. Рязань, 2014. 24 с.

2. Реализация протоколов RS-232 и 1-Wire с использованием языка описания аппаратуры VHDL: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. И.С. Холопов. Рязань, 2015. 24 с.

3. Реализация цифровых протоколов передачи информации и систем на кристалле на ПЛИС: методические указания к лабораторным работам / Рязан. гос. радиотехн. ун-т; сост. И.С. Холопов. Рязань, 2016. 48 с.

4. DE1 Development and Education Board User Manual [s.a.][s.l.]. — 52 pp.

5. Зобенко А. А., Филиппов А. С., Комолов Д. А., Мяльк Р. А.. Системы автоматизированного проектирования фирмы Altera MAX+plus II и Quartus II. Краткое описание и самоучитель. – издательство «РадиоСофт» . — 2002 г. — 360 с.

6. http://www.tinyvga.com/vga-timing

Приложение. Перечень элементов СхЭ

| Поз.  обознач. | Наименование | Кол. | Примечание |
| --- | --- | --- | --- |
|  | Конденсаторы |  |  |
| С1, С3 - С5 | 100 мкФ- 6 В SMD E24 | 4 |  |
| С2 | 0,1 мкФ SMD E24 | 1 |  |
|  |  |  |  |
|  | Микросхемы |  |  |
| IC1 | |  | | --- | | LM1117-3,3-0,8A | | 1 |  |
|  | |  | | --- | |  | |  |  |
|  | Резисторы |  |  |
| R1 | 120 Ом SMD E24 | 1 |  |
|  |  |  |  |
|  | Диоды |  |  |
| VD1 | BL-L101URC | 1 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |