

РЕАЛИЗАЦИЯ СТРУКТУРНО ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА ВЫВОДА НА ДИСПЛЕЙ МАКЕТА M1AFS ЧАСТЬ 2

Зоев И.В., Прокопюк С.Ю.

Мальчуков А.Н.

Томский политехнический университет

zoev.ivan@ya.ru

Введение

На основе структурно-функциональной схемы были реализованы блоки вывода строк (string и SRAM), знакогенератор – symbol. Взаимодействие блоков происходит по принципу master-slave. Блоки string и memory controller управляются блоком main.

В реализации блока вывода строк учувствуют так же блок памяти [1] (SRAM) и управления памятью (memory_controller). В памяти, при помощи memory_controller формируются строки, которые будут выводиться на экран. String из памяти SRAM считывает строку и производит управление ее вывода по средствам symbol.

Блок String

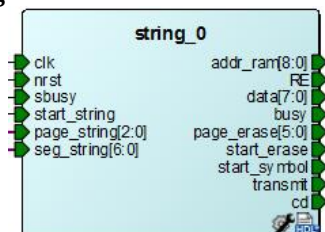


Рис.1. Внешний вид блока String

Входы:

clk – входной синхросигнал. 50 МГц

nrst – асинхронный сброс. Активный уровень сигнала – низкий.

sbudy – вход, занятости подчиненных устройства

Start_string - сигнал начала работы блока string

Page_string[2:0] – номер страницы вывода строки в DDRAM

Seg_string[7:0] – номер страницы вывода строки в DDRAM

Выходы:

Addr_RAM[8:0] - выход адреса памяти SRAM.

RE – сигнал чтения данных из SRAM. Активный уровень сигнала – высокий.

Data[7:0] – байт, записываемый в очередь FIFO

busy – сигнал занятости блока.

erase_page[5:0] – битовая маска отчистки страниц

start_erase – вход начала работы блока erase.

Start_symbol – сигнал начала работы блока symbol.

transmit – сигнал записи в FIFO очередь

cd – выход определяющий входной байт (данные, команда)

Принцип работы:

Блок string реализует посимвольный вывод информации из памяти SRAM. Когда приходит сигнал start_string блок устанавливает сигнал busy и начинает работу. На выход data устанавливается значение страницы B0-B5, на которую будет происходить вывод информации вместе с высоким уровнем сигнала cd. Затем идет отправка в FIFO по сигналу transmit. Затем в data записывается номер

сегмента, с которого будет происходить вывод информации и посылается очередь. Эти действия инициализируют строку для вывода на нее информации. Затем, на выход page_erase устанавливается маска строки, на которую мы будем записывать данные и посылается сигнал start_erase. После отчистки блок string производит обращение к памяти SRAM с установкой высокого уровня сигнала на RE. Начальный адрес ячейки, с которой начинается строка в памяти, устанавливается на выход addr_ram исходя из соотношения page_string*20. Затем посылается сигнал управления start_symbol. После окончания работы symbol, адрес на addr_ram увеличивается на 1, и так 19 раз, для вывода полной строки на дисплей. После каждого обращения к памяти запускается блок symbol. По завершении вывода строки string сбрасывает busy.

Блок Symbol

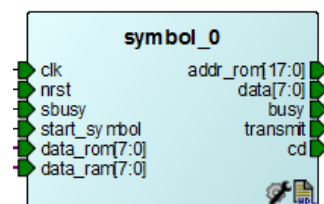


Рис.2. Внешний вид блока Symbol

Входы:

clk – входной синхросигнал. 50 МГц

nrst – асинхронный сброс. Активный уровень сигнала – низкий.

sbudy – вход занятости подчиненного устройства

Start_symbol – сигнал начала работы блока symbol.

Data_ROM[7:0] – данные из Flash памяти

Data_RAM[7:0] – данные из SRAM памяти

Выходы:

Addr_ROM - адресация к Flash памяти

Data[7:0] – байт, записываемый в очередь FIFO busy – сигнал означающий занятость блока.

transmit – сигнал записи в FIFO очередь

cd – выход определяющий входной байт (данные, команда)

Принцип работы:

После принятия сигнала управления start_symbol, блок symbol устанавливает сигнал busy. С входа data_ram, считывается ASCII код выводимого символа. На выход addr_rom устанавливается начальный адрес матрицы отображения символа на дисплее, из соотношения data_ram*5. Затем блок считывает данные из flash памяти с входа data_rom и устанавливает их на выход data. Адрес увеличивается на 1, для считывания следующего элемента отображения

символа. Затем, идет установка низкого уровня на cd и сигнал transmit для записи данных в FIFO. По завершении записи в очередь процедура обращения к памяти и запись происходит еще 4 раза, а потом завершает работу и сбрасывает busy .

Блок Memory Controller

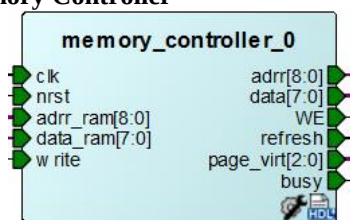


Рис.3. Внешний вид блока Memory Controller

Входы:

clk – входной синхросигнал. 50 МГц
 nrst – асинхронный сброс. Активный уровень сигнала – низкий.
 $\text{Data_RAM}[7:0]$ – данные для записи в SRAM
 $\text{Addr_RAM}[8:0]$ – адресация к памяти SRAM.
 Write – сигнал записи данных в SRAM. Активный уровень сигнала – высокий.

Выходы:

$\text{Addr_RAM}[8:0]$ – адресация к памяти SRAM.
 $\text{Data}[7:0]$ – данные считанные из SRAM
 WE – сигнал записи данных в SRAM. Активный уровень сигнала – высокий.
 refresh – сигнал для обновления информации на DDRAM дисплея
 busy – сигнал занятости блока.
 $\text{page_virt}[2:0]$ – номер страницы обновленной в SRAM.

Принцип работы:

Блок по сигналу write устанавливает сигнал занятости busy и начинает последовательно принимать сообщение на входные значения addr_ram и data_ram , которые переписываются в выходные регистры addr и data . Запись непосредственно в SRAM производится по сигналу WE . После сброса сигнала write , блок устанавливает значение виртуальной страницы измененной в памяти на выход page_virt и формирует сигнал refresh . После этого блок заканчивает работу и сбрасывает сигнал busy .

Блок SRAM

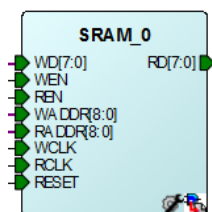


Рис.4. Внешний вид блока SRAM

Входы:

$\text{WD}[7:0]$ – вход данных для записи в SRAM
 WEN – разрешающий вход для записи данных. Активный уровень - высокий
 REN – разрешающий вход для чтения данных. Активный уровень - высокий
 $\text{WADDR}[8:0]$ – адрес ячейки памяти SRAM для записи
 $\text{RADDR}[8:0]$ – адрес ячейки памяти SRAM для чтения
 WCLK – вход синхронизации для записи данных.
 RCLK – вход синхронизации для чтения данных.
 RESET – Асинхронный сброс. Активный уровень - низкий

Выходы:

$\text{RD}[7:0]$ - выход данных для чтения из SRAM

Принцип работы:

Блок взят из библиотеки Libero SOC. Чтение и запись происходит при помощи управляющих сигналов RE и WE соответственно. Для обращения к ячейки памяти используются входы WADDR и RADDR . На вход WD подаются записываемые данные, а чтение происходит с выхода RD . Поскольку данных блок работает с двухпортовой памятью то запись и чтение можно производить одновременно. Цикл чтения и записи происходит за один такт [1].

Заключение

В работе представлена реализация блоков вывода строк на экран OLED дисплея, а также описан принцип работы и взаимодействие блоков string , symbol , memory controller , SRAM . При взаимодействии блоков main , memory controller и SRAM можно записать строку в память. Взаимодействие блоков main , string , symbol и SRAM позволяет считать и преобразовать строку для вывода ее на дисплей.

Список литературы

1. Configuring SRAM FPGAs Using Microsemi Fusion App Note // Microsemi // <http://www.microsemi.com/products/fpga-soc/fpga/fusion#documents> // (дата обращения: 20.10.2015)