

РЕАЛИЗАЦИЯ СТРУКТУРНО ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА ВЫВОДА НА ДИСПЛЕЙ МАКЕТА M1AFS ЧАСТЬ 1

Зоев И.В., Прокопюк С.Ю.

А.Н. Мальчуков

Томский политехнический университет
zoev.ivan@ya.ru

Введение

На основе структурно-функциональной схемы были реализованы блок управления - main, инициализации - init, стирания - erase и прокрутки дисплея - scroll. Взаимодействие блоков происходит по принципу master-slave.

Main является первым уровнем иерархии [1] в функциональной схеме и управляет блоками init, string и memory controller, которые занимают второй уровень. Scroll может работать параллельно с блоками второй иерархии. Третий уровень занимают блоки erase и symbol. Буфером данных является очередь FIFO.

Блок Main

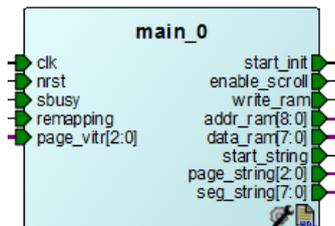


Рис.1. Внешний вид блока Main

Входы:

clk – входной синхросигнал. 50 МГц
nrst – асинхронный сброс. Активный уровень сигнала – низкий.
sbudy – вход занятости подчиненного устройства
remapping – сигнал обновления информации на DDRAM дисплея
page_virt[2:0] – номер измененной страницы в SRAM.

Выходы:

Start_init – сигнал начала работы блока init.
Enable_scroll – сигнал начала работы блока scroll
Write_RAM – сигнал записи данных в SRAM.
Активный уровень сигнала – высокий.
Addr_RAM[8:0]- выход адреса памяти SRAM.
Data_RAM[7:0] – данные записи в SRAM
Start_string - сигнал начала работы блока string
Page_string[2:0] – номер страницы вывода строки в DDRAM
Seg_string[7:0] – номер сегмента вывода строки

Принцип работы:

В начале работы устройства, main посылает управляющий сигнал start_init для начала работы init. Вход sbusy, сообщает main как о начале работы подчиненных блоков, так и о его завершении. Пока работает блок init, main ожидает завершения его работы. После завершения работы блока init, main посылает сигнал enable_scroll - разрешение работы блока

scroll. Затем, main формирует начальное сообщение для дисплея. Коды символов ASCII посылаются в память SRAM через выходной регистр data_RAM. Отправка data_RAM на блок memory controller вместе осуществляется вместе с управляющим сигналом записи write_RAM и адресом расположения символа в памяти addr_RAM. После завершения записи строки, из memory controller на main приходит сигнал, об изменении информации в SRAM, что говорит о необходимости запуска блока string. Такая структура взаимодействия позволяет динамично изменять информацию, выводимую на дисплей, а также обеспечить доступ к выводу информации на дисплей внешним блокам. После прихода сигнала remapping, main считывает с входа page_virt, какую страницу в памяти DDRAM необходимо обновить и устанавливает его на выход page_string. На выход seg_string устанавливается значение 04h, т.е. начальный сегмент вывода строки. Затем идет посылка сигнала start_string.

Блок Init

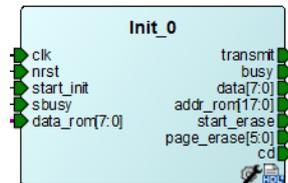


Рис.2. Внешний вид блока Init

Входы:

clk – входной синхросигнал. 50 МГц
nrst – асинхронный сброс. Активный уровень сигнала – низкий.
start_init – сигнал начала работы блока
sbudy – вход занятости подчиненных устройств
data_ROM[7:0] – входные данные из Flash памяти

Выходы:

transmit – сигнал записи в FIFO
busy – сигнал занятости блока.
Data[7:0] – байт, записываемый в очередь FIFO
cd – выход определяющий выходной байт (данные, команда)
Addr_ROM – выход адреса Flash памяти
Start_erase – сигнал начала работы блока erase
Erase_page[5:0] – битовая маска отчистки страниц

Принцип работы:

Для работы OLED дисплея макета M1AFS необходимо отправить последовательность команд в контроллер дисплея. Для хранения информации используется Flash память чипа FUSION[2]. При приходе сигнала start_init, блок устанавливает, сигнал занятости busy. Затем

считывается последовательность длиной 28 байт. Начиная с 0-го по 27-ой адреса памяти устанавливаются на выход addr_ROM. Считанная информация из памяти записывается в выходной регистр data и затем посылается в FIFO очередь с сигналом начала записи transmit и высоким уровнем сигнала cd. После завершения отправки последовательности, подается сигнал Start_erase вместе с битовой маской Erase_page со значением 111111b т.е. произойдет отчистка всех страниц, находящихся в DDRAM дисплея. После окончания работы блока erase, init заканчивает работу и сбрасывает сигнал busy.

Блок Erase

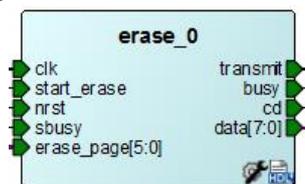


Рис.3. Внешний вид блока Erase

Входы:

clk – входной синхросигнал. 50 МГц
start_erase – вход начала работы блока.
nrst – Асинхронный сброс. Активный уровень сигнала – низкий.
sbusy – вход занятости подчиненных устройств.
erase_page[5:0] – битовая маска отчистки страниц

Выходы:

transmit – сигнал записи в FIFO очередь.
busy – сигнал, означающий занятость блока.
cd – выход определяющий выходной байт (данные, команда)
data[7:0] – байт записываемый в очередь FIFO

Принцип работы:

После принятия команды начала start_erase, блок устанавливает сигнал busy. В зависимости от входной битовой маски (erase_page) поочередно начинает стирать страницы. Для начала в очередь FIFO посылается команда выбора страницы для стирания – 00h-05h и сигнал cd 1. Для записи в FIFO используется сигнал transmit. Затем в очередь 104 раза [3] посылаются данные (data) отчистки 00h и cd 0. Если в Битовой маске установлена еще страница для стирания, то повторяются описанные выше действия. По завершению отчистки блок сбрасывает сигнал busy.

Блок Scroll

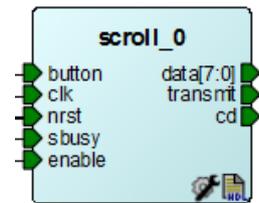


Рис.4. Внешний вид блока Scroll

Входы:

button – вход для кнопки
clk - входной синхросигнал. 50 МГц
nrst - Асинхронный сброс. Активный уровень сигнала – низкий.
sbusy – вход занятости подчиненных устройств.
enable – разрешающий вход работы блока.

Выходы:

transmit – сигнал записи в FIFO очередь.
cd – выход определяющий входной байт (данные, команда)

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

Принцип работы:

Блок начинает работу после того как main устанавливает сигнал enable_scroll. По нажатию кнопки button, scroll посылает в FIFO команду сдвига видимой области экрана на следующий участок DDRAM экрана, записанную в data и cd 1. Сдвиг происходит на одну страницу вниз при каждом нажатии. При достижении нижней границы DDRAM, передается команда возврата к началу DDRAM.

Заключение

В работе представлена реализация блоков, а также описан принцип работы и взаимодействие блоков main, init, erase и scroll. Совместная работа данных блоков позволяет провести инициализацию дисплея, отчистку дисплея и управлять прокруткой дисплея.

Список литературы

1. О применении блочно-ориентированного подхода к разработке устройств на ПЛИС. // Вестник науки Сибири // URL: <http://sjs.tpu.ru/journal/article/view/78/132/> // (дата обращения: 16.10.2015)
2. Embedded Flash memory blocks // Slideshare // <http://www.slideshare.net/Flashdomain/fusion-embedded-flash-memory-blocks/> // (дата обращения: 20.10.2015)
3. Datasheet OLED SSD0300// CEC-MC // <http://www.cec-mc.ru/data/files/File/PDF/SSD0300-Revision%201.4.pdf> // (дата обращения: 20.10.2015)