

РЕАЛИЗАЦИЯ СТРУКТУРНО ФУНКЦИОНАЛЬНОЙ СХЕМЫ УСТРОЙСТВА ВЫВОДА НА ДИСПЛЕЙ МАКЕТА M1AFS ЧАСТЬ 3

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 очередью FIFO (cq_in, cq_out и FIFO) и блок интерфейсного взаимодействия (APB_to_I2C).

Реализация блока управления очередью позволяет разделить управление master - slave. Таким образом cq_in является самым нижним уровнем иерархии блока управления (main). Однако, блок cq_out является блоком первого уровня после блока FIFO, который управляет блоком APB_to_I2C. Cq_in и cq_out управляют по средствам FIFO[1] чтением и записью в очередь. APB_to_I2C управляет блоком CoreI2C.

Блок Cq_in

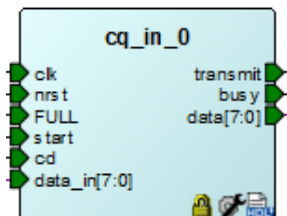


Рис.1. Внешний вид блока Cq_in

Входы:

clk – входной синхросигнал. 50 МГц

nrst – Асинхронный сброс. Активный уровень сигнала – низкий.

FULL – вход обозначения полного заполнения очереди FIFO

start – сигнал начала работы блока

cd – вход определяющий входной байт (данные, команда)

data_in[7:0] – байт записываемый в FIFO

Выходы:

transmit – сигнал записи в FIFO очередь.

busy – сигнал занятости блока.

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

Принцип работы:

При принятии сигнала start, блок устанавливает сигнал busy и считывает с входа data_in, байт, который должен быть отправлен на контроллер дисплея. В выходной регистр data записывается значение 80h, если cd 1 или C0h, если cd 0. По сигналу transmit значения выходного регистра data записывается в очередь, при условии, что входной сигнал FULL имеет низкий уровень, т.е. в очереди есть свободные места для записи. Затем в выходной регистр записывается значение data_in и посылается сигнал FIFO transmit. Если очередь заполнена, блок ожидает, пока из нее не будет считан хотя бы один байт, а потом записывает данные в очередь.

Блок FIFO

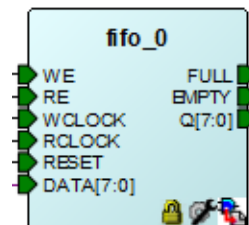


Рис.2. Внешний вид блока FIFO

Входы:

WE – Разрешающий вход для записи в FIFO. Активный уровень сигнала – высокий.

RE – Разрешающий вход для чтения из FIFO. Активный уровень сигнала – высокий.

WCLOCK – синхросигнал записи FIFO. 50 МГц

RCLOCK – синхросигнал чтения FIFO. 50 МГц

RESET – Асинхронный сброс. Активный уровень сигнала – низкий.

DATA[7:0] – записываемый байт в FIFO.

Выходы:

FULL – выход, для обозначения полного заполнения очереди FIFO

EMPTY – выход, для обозначения окончания очереди FIFO

Q[7:0] – выход считывания для байта из FIFO.

Принцип работы:

Блок взят из библиотеки Libero SOC. Очередь FIFO основана на памяти SRAM поэтому взаимодействие с блок схоже с блоком SRAM. Цикл записи происходит за один такт, а чтения за два. Данная FIFO очередь сконфигурирована таким образом, чтобы вмещать в себя 512 слов по 8 бит в каждом. Сигналы FULL и EMPTY говорят о заполнении и окончании очереди соответственно[1].

Блок Cq_out

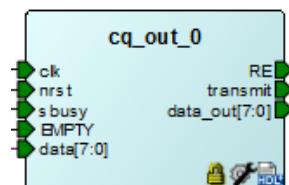


Рис.3. Внешний вид блока Cq_out

Входы:

clk – входной синхросигнал. 50 МГц

nrst – Асинхронный сброс. Активный уровень сигнала – низкий.

sbusy – вход, для обозначения начала работы подчиненных устройств.

EMPTY – вход обозначения окончания очереди FIFO

start – сигнал начала работы блока

data[7:0] – считанный байт из FIFO

Выходы:

RE – управляющий сигнал, разрешения чтения из FIFO.

transmit – сигнал работы APB_to_I2C блока.

data_out[7:0] – байт считанный из очереди FIFO

Принцип работы:

Блок просматривает занятость *apb_i2c* блока при помощи сигнала *sbusy*. Если *apb_to_i2c* свободен и в очереди имеются элементы для передачи (сигнал *EMPTY* не установлен), тогда происходит считывание байта из FIFO. Для этого устанавливается сигнал *RE*. Через два такта в выходной регистр *data_out* записывается считанное значение из очереди (*data*). Отправка данных *data_out* на блок *APB_to_I2C* происходит по сигналу *transmit*.

Блок APB_to_I2C

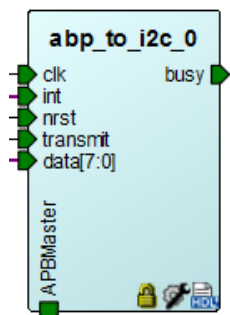


Рис.4. Внешний вид блока APB_to_I2C

Входы:

clk – входной синхросигнал. 50 МГц

int – вход для принятия сигнала прерывания.

nrst – Асинхронный сброс. Активный уровень сигнала – низкий.

transmit – сигнал начала работы блока.

data[7:0] – байт, считанный из FIFO

PDATA[7:0] – вход для считывания из APB slave данных. Находится в APBMaster.

Выходы:

busy – сигнал занятости блока

RADDR[8:0] – адрес регистров подчиненного APB устройства. Находится в APBMaster.

PSEL – сигнал выбора подчиненного APB устройства. Находится в APBMaster.

PENA – сигнал стробирования подчиненного устройства. Находится в APBMaster.

PWRITE[7:0] – выход записи в подчиненное APB устройство. Находится в APBMaster.

Принцип работы:

Когда на *APB_to_I2C* приходит сигнал начала работы *transmit*, блок устанавливает сигнал *busy* и проверяет, открыта ли передача на интерфейсе I2C. Если передача закрыта, блок *APB_to_I2C* по устанавливает на выход *PWDATA* значение 63h, которое включает в себя бит начала передачи. На *PADDR* устанавливается значение 00h, указывающее адрес внутреннего регистра *CORE_I2C*. Запись в регистр происходит после оплавки и последующего сброса сигнала *PENA*. Затем блок переходит в режим ожидания прерывания. После прихода прерывания, передача на I2C открыта и необходимо инициализировать принимающее устройство. Блок устанавливает на *PADDR* 04h, а на *PWDATA* значение 78h. Затем блок записывает значение 43h на выход *PWDATA*, которое сбрасывает бит прерывания. Адрес внутреннего регистра *PADDR* - 00h. После передачи *PWDATA* происходит передача данных по I2C. Блок *APB_to_I2C* переходит в состояние ожидания окончания передачи. После ожидания, блок записывает в *PADDR* - 04h, а в *PWDATA* передаваемый байт. Затем записываются значения *PADDR* 00h, а в *RWDATA* 43h. *APB_to_I2C* опять переходит в режим ожидания окончания передачи. После окончания передачи блок сбрасывает сигнал *busy*. Блок ожидает прихода сигнала *transmit* и следующего байта из очереди FIFO. Если передача открыта тогда блок передает данные, как описано выше, но без открытия передачи и посылки адреса подчиненного устройства

Если же с FIFO определенное время не приходит сигнал *transmit*, тогда на *PADDR* подается значение 00h, а на *PWDATA* - 53h, что означает закрытие передачи. Если же из FIFO приходит еще один байт, а передачу *APB_to_I2C* закрыл то повторяются действия по открытию передачи.

Заключение

В работе представлена реализация блоков очереди FIFO *cq_in*, *fifo*, *cq_out* и блока интерфейсного взаимодействия *APB_to_I2C*. При взаимодействии блоков *cq_in*, *fifo*, *cq_out* происходит чтение и запись в очередь. Взаимодействуя с очередью блок *APB_to_I2C* пересылает данные на IP Core – *CoreI2C*.

Список литературы

4. CoreFIFO Handbook // Actel // http://www.actel.com/ipdocs/CoreFIFO_HB.pdf // (дата обращения: 20.10.2015)