

Лабораторная работа №4 “Устройства ввода-вывода”

О найденных ошибках и опечатках просьба сообщать на riropov@gmail.com

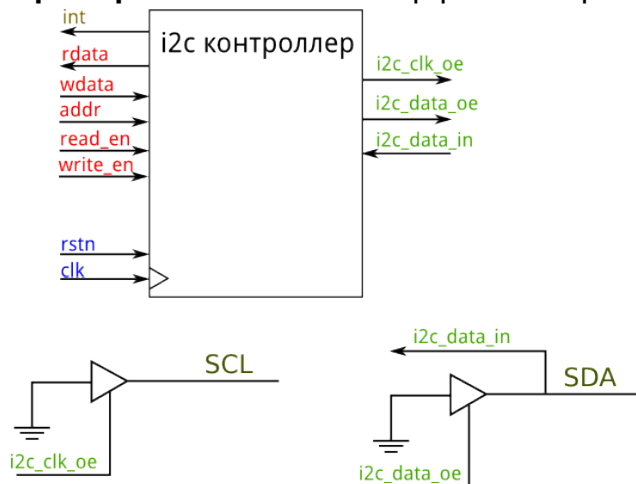
Цель работы:

1. Изучение принципов взаимодействия процессора и периферийных устройств.
2. Разработка контроллера ввода-вывода для заданного интерфейса.

Порядок выполнения работы:

- 1) **Ознакомьтесь с документацией на прототипную плату Spartan-3AN Starter Kit**
http://www.xilinx.com/support/documentation/boards_and_kits/ug334.pdf
- 2) **Ознакомьтесь с вариантом задания, прочитайте документацию по заданному интерфейсу.** Рекомендуемая литература:
 - a) Pong P. Chu. FPGA Prototyping By Verilog Examples: Xilinx Spartan-3 Version
 - b) <http://www.computer-engineering.org/ps2protocol/> - PS/2
- 3) **Разработайте сигнальный интерфейс для заданного контроллера.** Контроллер должен иметь возможность взаимодействия с процессорным ядром и внешним устройством. Процессор конфигурирует контроллер, а затем осуществляет через него обмен данными с внешним устройством.

Пример: Сигнальный интерфейс контроллера шины i2c



Имя порта	Направление	Ширина	Описание
clk	вход	1	Вход сигнала синхронизации (синхронизация по переднему фронту)
rstn	вход	1	Синхронный сброс по уровню лог '0'

rdata	выход	32	Интерфейс с процессорным ядром: rdata — прочитанные данные wdata — данные для записи addr — адрес чтения/записи read_en — разрешение чтения write_en — разрешение записи
wdata	вход	32	
addr	вход	10	
read_en	вход	1	
write_en	вход	1	
int	выход	1	состояние, когда оба сигнала read_en и write_en установлены в лог '1' не разрешено Сигнал прерывания
i2c_clk_oe	выход	1	Интерфейс I2C: сигналы i2c_clk_oe и i2c_data_oe управляют выходами с открытым стоком SCL и SDA: лог «1» - подключает выход к «земле» лог «0» - переводит выход в состояние с высоким импедансом i2c_data_in - входные данные с шины SDA
i2c_data_oe	выход	1	
i2c_data_in	вход	1	

4) Разработайте регистровый интерфейс для заданного контроллера.

Пример: Регистровый интерфейс контроллера i2c

Адреса регистров выровнены на границу слова. Таким образом, чтение и запись любого регистра, вне зависимости от его ширины(разрядности) возможно с помощью команд LW и SW.

Контроллер содержит внутри себя два FIFO буфера на 1024 байта для принимаемых и отправляемых данных.

FIFO буфер отправляемых данных (выходной буфер) позволяет процессору загружать данные для отправки на внешнее устройство, не дожидаясь завершения передачи каждого отдельного байта.

FIFO буфер принимаемых данных (входной буфер) позволяет контроллеру хранить принятые данные до момента, когда они будут прочитаны процессором.

Также контроллер имеет буфер команд на 1024 команды (команды чтения или записи).

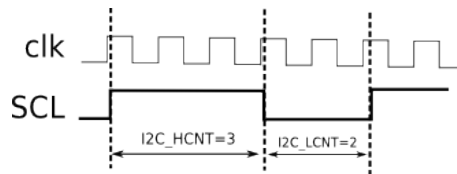
Имя регистра	Смещение (Относительно базового адреса)	Ширина	Доступ (чтение/запись)	Описание, назначение битов в регистре
I2C_CTRL	0x00	1	чтение/запись	[1] — Запись «1» в данный бит переводит контроллер в режим

				ожидания (сна), все внутренние регистры сбрасываются, транзакции на шине прерываются. Запись «0» возвращает контроллер в нормальный режим работы. [0] — Режим адресации: <ul style="list-style-type: none"> • 1 : 10-битные адреса • 0 : 7-битные адреса
I2C_TAR_ADDR	0x04		чтение/запись	[9:0] — Адрес устройства на шине i2c, передается в начале транзакции на шине
I2C_INT_STAT	0x08	9	только чтение	Регистр статуса (прерываний) [8] — ошибка на шине [7] — FIFO команд пустое [6] — Переполнение буфера команд [5] — Попытка чтения из пустого входного FIFO [4] — Выходное FIFO достигло уровня порога I2C_TX_THLD [3] — Входное FIFO достигло уровня порога I2C_RX_THLD [2] — Выходное FIFO переполнилось [1] — Входное FIFO переполнилось [0] — Идет транзакция на шине
I2C_INT_MASK	0x0c	9	чтение/запись	Маска прерываний: 0 — разрешает прерывание по соответствующему признаку 1 — запрещает прерывание по соответствующему признаку Описание битов см. I2C_INT_STAT
I2C_INT_CLR	0x10	9	только запись	Запись «1» сбрасывает соответствующее прерывание. Описание битов, см. I2C_INT_STAT
I2C_RFIFO_SIZE	0x14	10	только чтение	Число элементов во входном буфере

I2C_WFIFO_SIZE	0x18	10	только чтение	Число элементов в выходном буфере
I2C_CFIFO_SIZE	0x1c	10	только чтение	Число элементов в буфере команд
I2C_TX_CMD	0x20	9	только запись	<p>Запись в I2C_TX_CMD отправляет запрос на чтение/запись в очередь команд.</p> <p>[8] — Команда:</p> <ul style="list-style-type: none"> • 1 — Запись • 0 — Чтение <p>[7:0] — Данные на запись. Данные биты игнорируются в случае если отправляемая команда — чтение.</p>
I2C_RX_BUF	0x24	8	только чтение	Чтение из I2C_RX_BUF возвращает первый элемент из буфера входных данных. Если буфер пустой то будет прочитан «0» и установлен признак ошибки в I2C_INT_STAT
I2C_HCNT	0x28	16	чтение/запись	[15:0] — Длительность лог «1» на шине SCLK в тактах (см. рисунок ниже)
I2C_LCNT	0x2c	16	чтение/запись	[15:0] — Длительность лог «0» на шине SCLK в тактах (см. рисунок ниже)
I2C_TX_THLD	0x30	10	чтение/запись	<p>Порог очищения буфера отправляемых данных.</p> <p>Пример: Если I2C_TX_THLD = 3, то после того как в FIFO буфере останется только 3 байта, будет установлен сигнал прерывания.</p>
I2C_RX_THLD	0x34	10	чтение/запись	<p>Порог заполнения буфера принимаемых данных до возникновения прерывания.</p> <p>Пример: Если I2C_RX_THLD = 3, то после приема 3-х байт с внешнего устройства и отправки</p>

				их в FIFO буфер будет установлен сигнал прерывания.
--	--	--	--	---

Пояснение к регистрам I2C_HCNT и I2C_LCNT



5) Разработать модель внешнего устройства для симуляции.

Данная модель может быть написана с использованием всех возможностей Verilog HDL (а не только синтезируемого подмножества). В дальнейшем разработанную модель следует использовать для тестирования контроллера.

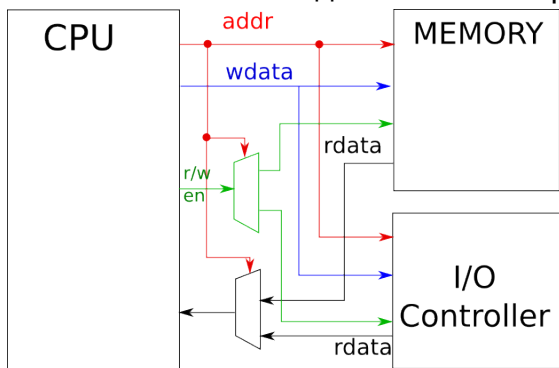
6) Разработать тестовое окружение и тесты для контроллера.

Например, тест конфигурирует контроллер и модель внешнего устройства для работы в одном из возможных режимов (частоты передачи данных, формата передаваемого пакета и др.), после чего проверяет корректность обмена данными между контроллером и моделью внешнего устройства.

7) Разработать заданный контроллер и осуществить его тестирование путем симуляции.

9) Интегрировать контроллер с процессорным ядром из ЛР2 или ЛР3.

Возможная схема подключения контроллера:



10) Протестировать интегрированную систему путем симуляции.

11) Провести тестирование интегрированной системы из процессора и контроллера внешнего устройства на плате Spartan-3AN Starter Kit

Обратите внимание на то, что Xilinx поддерживает синтез процедуры \$readmemh для инициализации встроенной на ПЛИС SRAM. Подробности в XST User guide: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/xst.pdf

Варианты задания

1.	<p>Контроллер PS/2 Мыши. Контроллер реализует интерфейс к PS/2 мыши и светодиодам LED0-7.</p> <p>Тестовая программа: Тестовая программа реализует счетчик. При движении мыши вверх по оси Y значение счетчика увеличивается, при движении вниз - уменьшается. Значение счетчика выводится на светодиоды.</p>
2.	<p>Контроллер PS/2 Клавиатуры. Контроллер реализует интерфейс к PS/2 клавиатуре и светодиодам LED0-7.</p> <p>Тестовая программа: Код нажатой на клавиатуре клавиши выводится на светодиоды.</p>
3.	<p>Приемопередатчик UART. Контроллер реализует приемопередатчик UART, интерфейс к кнопке BTN_EAST и светодиодам LED0-7.</p> <p>Тестовая программа: При нажатии BTN_EAST в последовательный канал отправляется очередной элемент последовательности Фибоначчи. Значение принятого по UART байта выводится на светодиоды.</p>
4.	<p>Контроллер VGA. Контроллер реализует интерфейс VGA с двойным видеобуфером и интерфейс к кнопкам BTN_*.</p> <p>Тестовая программа: Выводит прямоугольный курсор на экран, нажатие кнопок перемещает курсор по экрану.</p> <p>Внимание: т.к. микросхема ПЛИС Spartan 3AN относится к бюджетному сегменту, ресурсов (встроенной SRAM) для реализации видеобуфера</p>

для стандартных видеорежимов будет не достаточно. Обычно для хранения больших объемов данных используются внешние микросхемы памяти (например, DDR SDRAM в случае платы *Spartan-3AN Starter Kit*).

В данной работе интерфейс к внешней памяти реализовывать не требуется. Вместо этого, видеобuffer реализованный в контроллере должен быть небольшим, например 64x48 пикселей. Тогда при выводе буфера в режиме VGA 640x480 один пиксель видеобufferа будет отображен на 100 соответствующих пикселей на экране.