

Лабораторная работа №5

Разработка конечного автомата для обмена данными с АЦП и ЦАП с интерфейсом SPI.

Цель работы: разработать интерфейсный модуль для обмена данными с АЦП и ЦАП с использованием методики построения синхронного конечного автомата.

1. Создать новый проект в САПР ISE. Выбрать для проекта ПЛИС XC3S500E-FT256. Выбрать в качестве верхнего уровня файл типа Schematic. Добавить к проекту файл top.sch.

2. Разработать управляющий автомат для формирования временных диаграмм для обмена с устройством, имеющим интерфейс SPI.

2.1. Создать новый модуль с именем spi_adc и типом VHDL Module (Add New Source).

2.2. В окне мастера при создании модуля ввести сигналы:

- clk (направление передачи - in)
- d (шина, направление передачи – out, разрядность 11:0)
- transmit (направление передачи - in)
- spi_clk (направление передачи – out)
- spi_mosi (направление передачи – out)
- spi_cs (направление передачи – out)
- spi_miso (направление передачи – in)

2.3. Выполнить поведенческое описание модуля spi_adc на языке VHDL.

2.3.1. Выбрать параметры обмена по интерфейсу SPI. Для этого ознакомиться с документацией на АЦП и определить:

- минимальную длительность периода тактового сигнала spi_clk;
- минимальное время удержания данных на входе АЦП clk_mosi;
- минимальное время задержки между формированием фронта тактового сигнала и появлением данных на выходе АЦП.

2.3.2. Используя в качестве источника тактового сигнала генератор с частотой 50 МГц (и периодом 20 нс), нарисовать диаграмму обмена с АЦП, приняв за единицу времени один период тактового сигнала ПЛИС (т.е. 20 нс). Определить требуемые задержки при формировании сигналов SPI и чтении данных с АЦП в единицах измерения «20 нс».

2.3.3. Принять в качестве исходного состояния конечного автомата состояние «0». При наличии на входе transmit логической единицы автомат переходит к состоянию «1» и инициирует последовательность чтения данных из АЦП. При достижении состояний с

номером, соответствующим возможности чтения очередного бита данных из АЦП, он заносится в один из разрядов сигнала d. Заполнить таблицу.

Таблица

Соответствие выполняемых действий номеру состояния автомата

Десятичное значение номера состояния	Изменяемые сигналы
0	
M	Возврат к состоянию 0

2.3.4. В раздел architecture перед ключевым словом begin добавить описание сигнала st, хранящего номер состояния автомата. Тип сигнала выбрать как integer range 0 to <M>, где M – максимальный номер состояния автомата.

2.3.5. В раздел architecture описания автомата на VHDL ввести оператор process, описывающий работу модуля по фронту тактового сигнала clk. В теле оператора process реализовать конечный автомат с помощью оператора case

case st is

when 0 => <операторы>

when 1 => <операторы>

...

when others => st <= st + 1;

end case;

В прочих (others) состояниях использовать инкремент сигнала st, соответствующий последовательной смене состояния.

2.3.6. Сохранить файл и создать графическое представление компонента Design Utilites -> Create Schematic Symbol в панели Processes.

2.3.7. Перейти в графический файл верхнего уровня, выбрать палитру компонентов и разместить полученный компонент на листе проекта.

2.4. Выполнить поведенческое описание вспомогательного модуля на языке VHDL. Назначением вспомогательного модуля является обеспечение сигналов, требуемых для проверки работоспособности модуля spi_adc.

2.4.1. Создать новый модуль с именем adc_ctrl и типом VHDL Module (Add New Source).

2.4.2. В окне мастера при создании модуля ввести сигналы:

- transmit (направление передачи - out)

2.4.3. В разделе architecture выполнить асинхронное назначение сигналов: transmit <= '1';

2.4.4. Сохранить файл и создать графическое представление компонента Design Utilities -> Create Schematic Symbol в панели Processes.

2.4.5. Перейти в графический файл верхнего уровня, выбрать палитру компонентов и разместить полученный компонент на листе проекта.

2.5. Соединить сигналы transmit компонентов spi_adc и adc_ctrl. Остальные выводы компонентов подключить к маркерам ввода-вывода соответствующего направления.

2.6. Разместить на листе проекта компонент uart. Подключить к линиям данных этого компонента линии d(11:4) модуля spi_adc. Подключить к линии transmit линию transmit модуля adc_ctrl, обеспечив тем самым непрерывную передачу в СОМ-порт компьютера старших 8 разрядов считанного из АЦП числа.

2.7. Создать файл проектных ограничений (pins, тип Implementation Constraints File).

2.8. Запустить редактор проектных ограничений двойным щелчком на созданном файле pins.

2.9. В редакторе проектных ограничений ввести обозначения выводов, соответствующих тактовому входу clk, сигналам spi_clk, spi_miso, spi_mosi, spi_cs, tx созданного проекта. Номера выводов определить из документации на используемую плату.

2.10. Запустить процесс трансляции проекта.

2.10. Выполнить подключение платы к программатору и загрузить конфигурацию ПЛИС.

2.11. Запустить программу для обмена данными с последовательным портом (напр., HyperTerminal). В настройках порта указать номер, к которому физически подключен нуль-модемный кабель, скорость обмена 9600, 8 бит данных, нет контроля четности, 2 стоп-бита. Разрешить прием данных и сохранить принимаемый поток символов в файл в двоичном виде.

2.12. Проанализировать содержимое принятых файлов при различных уровнях напряжения, поданного на вход АЦП. В качестве источников напряжения можно использовать уровень нулевого потенциала, опорное напряжение, а также напряжения питания ПЛИС. Во избежание повреждения АЦП рекомендуется использовать делители напряжения, подключаемые к выводам питания платы и обеспечивающие фиксированный набор входных напряжений.

3. Выполнить отчет о проделанной работе. Отчет должен содержать:

- описание основных действий конечного автомата и разъяснение порядка переходов между состояниями;

- таблицы значений, считанных из АЦП при различных уровнях входного напряжения;
- сведения о ресурсах ПЛИС, необходимых для реализации проекта;
- сведения о максимальной тактовой частоте проекта.