## Лабораторная работа №4

## Разработка конечного автомата для управления периферийным устройством.

Цель работы: разработать интерфейсный модуль RS-232 с использованием методики построения синхронного конечного автомата.

1. Создать новый проект в САПР ISE. Выбрать для проекта ПЛИС XC3S500-FT256. Выбрать в качестве верхнего уровня файл типа Schematic. Добавить к проекту файл top.sch.

2. Разработать управляющий автомат для передачи данных по протоколу RS-232.

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

2.2. В окне мастера при создании модуля ввести сигналы:

- clk (направление передачи in)
- din (шина, направление передачи in, разрядность 7:0)
- transmit (направление передачи in)
- tx (направление передачи out)

2.3. Выполнить поведенческое описание модуля uart на языке VHDL.

2.3.1. Выбрать параметры передачи RS-232: скорость 9600 бит/с, 8 бит данных, нет бита четности, 2 стоп-бита.

2.3.2. Используя в качестве источника тактового сигнала генератор с частотой 50 МГц, рассчитать длительность число периодов тактового сигнала, необходимое для передачи одного бита.

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 увеличения состояния на единицу с каждым тактом. При достижении номера состояния, соответствующего переходу к передаче следующего бита, этот бит записывается в выход tx. Заполнить таблицу, приняв за N число периодов тактового сигнала, необходимое для передачи одного бита.

Таблица

(	Соответствие вы	полняемых дейс	твий номеру	состояния	автомата

Номер состояния	Десятичное	значение	номера	Передаваемый сигнал
	состояния			
0				1 (отсутствие передачи)
1				0 (старт-бит)
Ν				Di(0)

2N	Di(1)
3N	Di(2)
4N	Di(3)
5N	Di(4)
6N	Di(5)
7N	Di(6)
8N	Di(7)
9N	1 (стоп-бит)
10N	1 (стоп-бит)
11N	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 М – 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. При достижении номеров, соответствующих числам, полученным в таблице, в строке операторов записать присваивание переменной st нового значения, и запись в выход tx соответствующего бита данных.

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 uart.

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

2.4.2. В окне мастера при создании модуля ввести сигналы:

- din (шина, направление передачи – out, разрядность 7:0)

- transmit (направление передачи - out)

2.4.3. В разделе architecture выполнить асинхронное назначение сигналов: transmit <= '1'; din <= "01010101";

2.4.4. Сохранить файл и создать графическое представление компонента Design Utilites -> 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 одноименные сигналы компонентов uart и uart\_ctrl. На тактовый вход компонента uart подать внешний сигнал, выход tx соединить с выходным маркером вводавывода.

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

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

2.8. В редакторе проектных ограничений ввести обозначения выводов, соответствующих тактовому входу clk и выходу tx созданного проекта. Номера выводов определить из документации на используемую плату.

2.9. Запустить процесс трансляции проекта.

2.10. Соединить разъем UART (RS-232) платы с портом COM компьютера с помощью нуль-модемного кабеля. 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 убедиться в приеме потока символов с кодом  $55_{10}$  (01010101<sub>2</sub>).