

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

КАФЕДРА ВТиЭ

Лабораторный практикум по курсу «Схемотехника ЭВМ»

Лабораторная работа №3

Барнаул 2016

Лабораторная работа № 3

Тема: Проектирование цифровых автоматов.

Цель работы:

Получение навыков синтеза и моделирования цифровых автоматов в САПР Altera MAX+PLUS II.

Задачи:

Задать временными диаграммами работы, цифровых автоматов с заданными параметрами в САПР Altera Max+plus II в режиме редактора временных диаграмм (WaveForm Editor). Произвести исследование временных параметров полученной модели (Simulator).

Теоретические сведения:

Разработка описания проекта в редакторе временных диаграмм является альтернативой его созданию в графическом редакторе. Здесь можно графическим способом задавать комбинации входных логических уровней и требуемых выходов. Созданный таким образом файл (.wdf) может содержать как логические входы, так и входы цифрового автомата, а также выходы комбинаторной логики, счётчиков и цифровых автоматов. Способ разработки дизайна в редакторе временных диаграмм лучше подходит для цепей с чётко определёнными последовательными входами и выходами, то есть для цифровых автоматов, счётчиков и регистров.

Для формирования некоторой выходной последовательности сигналов Y , соответствующей некоторой входной последовательности сигналов X , часто используют последовательную схему, называемую конечным или цифровым автоматом, или машиной состояний (state machine). Конечный автомат – это устройство с памятью, выходные сигналы которого зависят от предыстории поступления входных сигналов. При этом множество входных сигналов $X=\{x_1...x_L\}$ автомата определяется множеством логических условий, которые необходимо обработать, а множество выходных сигналов $Y=\{y_1...y_L\}$ автомата определяется множеством сигналов, являющихся результатом его работы.

В данном случае мы будем работать с детерминированным, синхронным видом конечного автомата (СКА). Т.е. с цифровым автоматом, при определении которого было заранее указано *начальное состояние автомата*, и память которого реализована на *триггерах*. СКА переходит из состояния в состояние только по активному перепаду (фронту или спаду) тактового сигнала синхронизирующего триггеры его памяти.

В зависимости от способа формирования выходных сигналов выделяют два класса конечных автоматов:

- Автомат Мура – автомат, выходные сигналы которого зависят только от текущего состояния автомата;
- Автомат Мили – автомат, выходные сигналы которого зависят от текущего состояния автомата и от текущих входных сигналов.

Состояние автомата определяется состоянием его памяти. Т.к. память автомата строится на триггерах, соответственно состояние памяти характеризуется двоичным набором, отображающим состояние триггеров. Данный двоичный набор называется кодом

состояния автомата. Число триггеров n , используемых для реализации памяти автомата, определяется числом состояний автомата N и способом их кодирования:

- Двоичное кодирование. При этом $n = \lceil \log_2 N \rceil$.
- Унитарное кодирование (одно состояние – один триггер). При этом $n = N$.

Выбор более эффективного способа кодирования зависит от типа ПЛИС. В данной работе мы не будем акцентировать на этом внимание и оставим способ кодирования по умолчанию (двоичное кодирование).

В данной работе, мы будем проектировать конечный автомат (машину состояний) Мура, имеющий 4 состояния:

1. Входными данными нашего автомата будет номер состояния, в которое мы хотим перейти из текущего состояния. При этом состояния кодируются следующим образом: **State_A** = 0, **State_B** = 1, **State_C** = 2, **State_D** = 3. Т.к. автомат имеет 4 состояния, нам понадобится 2 входа ($\log_2(4)=2$).
2. Выходными данными нашего автомата будет номер его состояния. Т.е. когда автомат находится в состоянии **State_A** на его выходе будет 0, в состоянии **State_B** – 1 и т.д. Т.к. автомат имеет 4 состояния, нам понадобится 2 выхода ($\log_2(4)=2$).
3. Алгоритм работы автомата задан таблицей переходов и состояний.

Литература:

1. Altera MAX+PLUS® II ver. 10.2 Help
2. Р.И.Грушвицкий, А.Х.Мурсаев, Е.П.Угрюмов Проектирование систем на микросхемах с программируемой структурой. – 2-е изд., перераб. и доп. - СПб.: БХВ-Петербург, 2006. – 736с.
3. В.Б.Стещенко ПЛИС фирмы ALTERA: проектирование устройств обработки сигналов.- М.: ДОДЭКА, 2000. - 128с.

Выполнение работы:

Выполнение лабораторной работы можно разделить на следующие этапы:

1. На основании кода переходов (Приложение 1, свой вариант), строится таблица переходов и состояний.
2. По таблице переходов и состояний строится граф переходов цифрового автомата.
3. По графу переходов определяется маршрут обхода графа, включающий все его ребра. В качестве начала маршрута (начальное состояние автомата) берется вершина **A**.
4. С помощью таблицы переходов и состояний строится последовательность входных воздействий (включая воздействия не вызывающие изменение состояний) соответствующая полученному маршруту.
5. С использованием редактора временных диаграмм пакета MAX+PLUS (WaveForm Editor) создается сигнальный файл проекта (**.wdf**), который содержит временные диаграммы, описывающие логику работы цифрового автомата. Для этого:
 - нажать кнопку открытия нового файла на панели инструментов, в открывшемся диалоговом окне «New» отметить пункт «Waveform Editor file», в соседнем окне выбрать расширение (**.wdf**) и нажать «ОК», после чего откроется окно редактора;
 - задать временные диаграммы для входов Rst, Clk и входов IN[1..0], на которые подается последовательность управляющих воздействий, полученная в п.4, а также соответствующие временные диаграммы внутреннего состояния самого автомата (использовать пункт «Insert

Mode», в «I/O Type» указать «Buried Node», в «Node Type» указать «Machine» + «Secondary Inputs» «Clock: Clk, Reset: Rst») и временные диаграммы выходов (в нашем случае они совпадают с состояниями самого автомата);

- создать символ цифрового автомата через подменю «Project», в котором следует выбрать пункт «Create Default Symbol». Созданный символ будет помещён в каталог проекта. Использование созданных символов, так же как и элементов других библиотек, производится через диалоговое окно «Enter Symbol».
6. С использованием графического редактора пакета MAX+PLUS (Graphic Editor) строится схема, состоящая из созданного символа цифрового автомата с присоединенными к нему элементами входов и выходов. **ВНИМАНИЕ: Имя проекта ни в коем случае не должно совпадать с именем символа цифрового автомата!**
 7. Выполняется компилирование схемы (Compiler).
 8. С помощью редактора WaveForm Editor задаются внешние воздействия, необходимые для проверки работоспособности схемы. В данном случае это может быть произвольная последовательность из 6 или более входных воздействий.
 9. Запускается симулятор работы схемы (Simulator).
 10. Измеряются временные задержки, возникающие при работе схемы. Измерение задержек можно проводить как вручную в WaveForm Editor, так и с помощью Timing Analyzer.

Требования к защите работы:

1. Демонстрация схемы цифрового автомата и результатов моделирования на компьютере.
2. Отчет по лабораторной работе (оформленный в соответствии с Приложением 2).
3. Правильные ответы на вопросы преподавателя по теме работы.

Пример выполнения лабораторной работы:

Задание:

Вариант №1. Необходимо синтезировать цифровой автомат по заданной таблице переходов.

Выполнение работы:

На основании кода переходов для заданного варианта, строим таблицу переходов и состояний.

0x7D44 = 0111'1101'0100'0100

Первая тетрада кода соответствует строке таблицы переходов для состояния **State_A**, вторая тетрада соответствует строке таблицы переходов для состояния **State_B** и т.д.

Состояние	State_A	State_B	State_C	State_D
State_A	0	1	1	1
State_B	1	1	0	1
State_C	0	1	0	0
State_D	0	1	0	0

Для управления автоматом необходим 2-разрядный вход $IN[1..0]$.

Управляющее воздействие можно определить следующим образом: для перехода из текущего состояния в состояние **State_A** (если этот переход задан графом) необходимо подать на вход автомата входное воздействие $0x0$, для перехода из текущего состояния в состояние **State_B** необходимо подать на вход автомата входное воздействие $0x1$ и т.д. Если же переход невозможен, то данное входное воздействие оставляет автомат к прежнем состоянии. Для определения состояния автомата необходим 2-разрядный выход $OUT[1..0]$. Информацию о состоянии цифрового автомата можно определить по значению установленном на данном выходе:

State_A: $OUT[1..0] = 0$;

State_B: $OUT[1..0] = 1$;

State_C: $OUT[1..0] = 2$;

State_D: $OUT[1..0] = 3$.

По таблице переходов строим граф переходов заданного цифрового автомата.

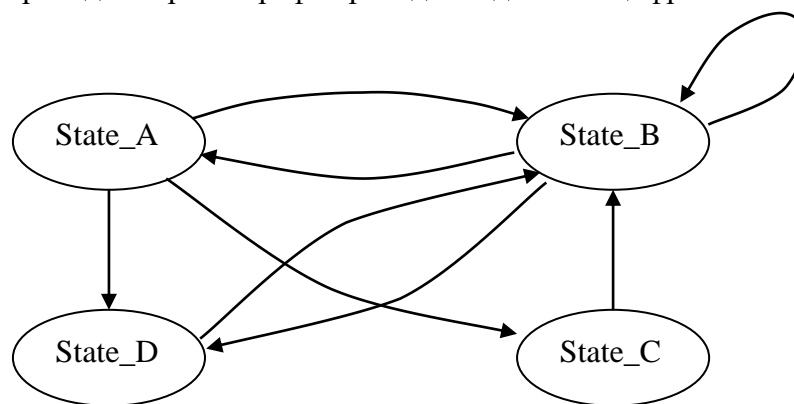


Рис.3.1 Граф цифрового автомата.

По графу переходов определяем маршрут обхода графа, исходя из того, что данный маршрут должен включать все ребра графа (при необходимости одну и ту же вершину можно проходить несколько раз, главное, чтобы маршрут включал все ребра графа).

Т.к. работа нашего цифрового автомата начинается из состояния **State_A**, то для заданного графа маршрут будет следующим:

A->B->B->A->C->B->D->B->A->D

По данному маршруту, используя таблицу переходов и состояний, строим последовательность входных воздействий приводящих к прохождению всего маршрута. Причем последовательность должна включать в себя еще и воздействия не вызывающие изменение состояний. Это делается потому, что компилятор считает, что результат работы цифрового автомата для неуказанных входных воздействий нас не интересует и может при подаче неучтенных воздействий вывести на выход все что угодно. Чтобы этого не произошло необходимо для каждого состояния автомата указать реакцию на все возможные входные воздействия. Таким образом, граф переходов заданного цифрового автомата приобретает следующий вид (цифрами указаны входные воздействия, приводящие к переходу в то или иное состояние):

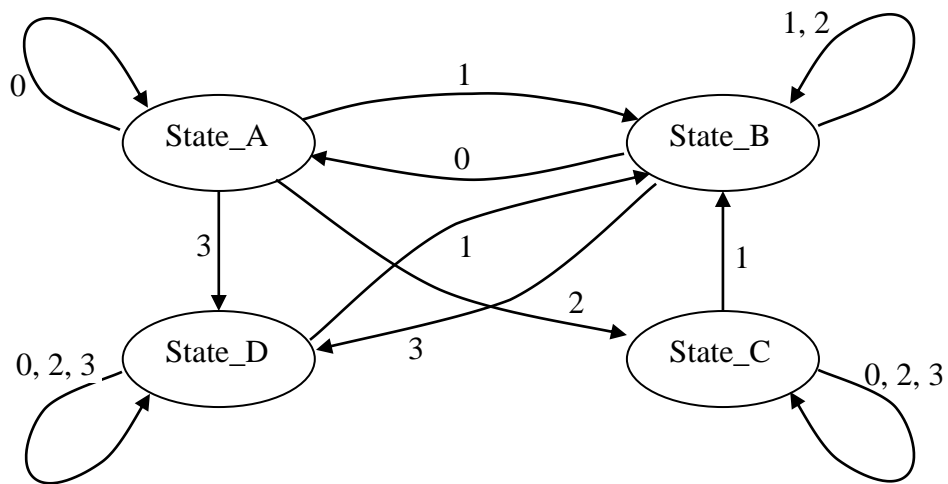


Рис.3.2 Граф цифрового автомата.

Итоговая последовательность входных воздействий, в таком случае, будет такова (в скобках указано состояние автомата, к которому приводит воздействие):

0(A)->1(B)->1(B)->2(B)->0(A)->2(C)->2(C)->3(C)->0(C)->1(B)->3(D)->3(D)->2(D)->0(D)->->1(B)->0(A)->3(D);

Строим диаграмму работы автомата в редакторе Waveform Editor(Altera Max+II):

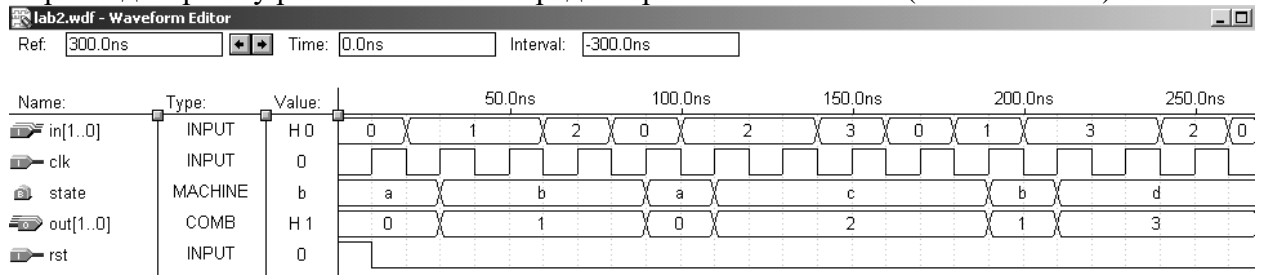


Рис.3.3. Диаграмма работы цифрового автомата (Waveform Editor).

Проверяем работу автомата на произвольной последовательности входных воздействий (не менее 6): 1 – 3 – 2 – 0 – 2 – 1

В редакторе Waveform Editor строим диаграмму входных воздействий:

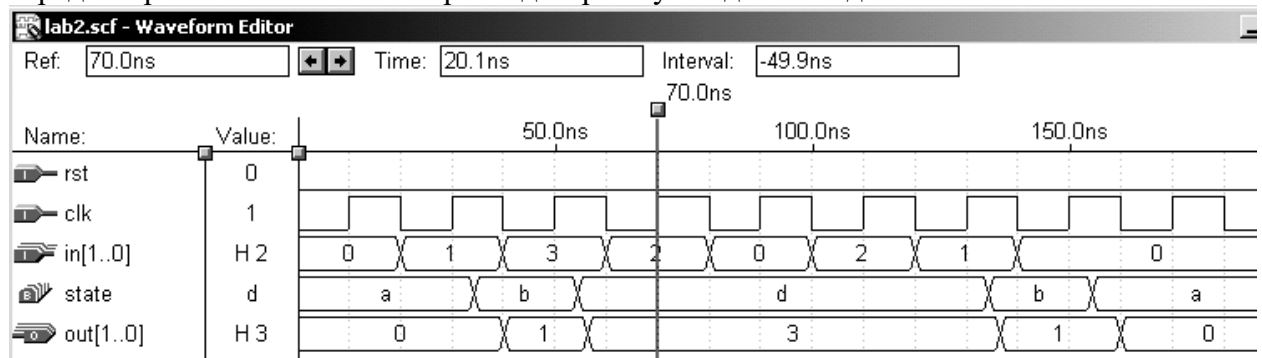


Рис.3.4. Отклик созданный автоматом на входное воздействие (Waveform Editor).

Отклик демонстрирует правильную реакцию созданной по диаграмме автомата.

С помощью средств WaveForm Editor производим замер временных задержек на каждом переключении, относительно фронта сигнала CLK с указанием кода переключения (Рис.3.4).

$T(\text{State_A} \rightarrow \text{State_B}, \text{clk}) = 4.2 \text{ нс.}$

$T(\text{State_B} \rightarrow \text{State_D}, \text{clk}) = 4.6 \text{ нс.}$

Вопросы к работе:

1. Приведите пример цифрового автомата.
2. Почему переключение сигнала на выходах OUT не совпадает с переключением состояния автомата?
3. Почему переключение состояния автомата не совпадает с фронтом сигнала CLK?
4. Зачем необходимы входы CLK и RST ?
5. Какой из автоматов (Мили или Мура) синтезирован в работе?
6. Каким способом можно перейти из любого состояния в состояние State_A, не дожидаясь сигнала CLK?

Приложение 1

Варианты для лабораторной работы №3

<i>Вариант №</i>	Код переходов	<i>Вариант №</i>	Код переходов	<i>Вариант №</i>	Код переходов
1	7D44	33	2ED4	65	D1A2
2	63A5	34	7FB6	66	2F3E
3	F13A	35	57A4	67	9868
4	E37C	36	E767	68	6384
5	591A	37	2F72	69	E93E
6	135B	38	E137	70	2196
7	4AE2	39	F785	71	A378
8	E2B6	40	A37E	72	CBE2
9	A7F2	41	9EF2	73	1FF4
10	BC34	42	38DA	74	3A69
11	25E3	43	C3A5	75	3B55
12	4AF4	44	593A	76	2983
13	E1B2	45	DEBE	77	4294
14	F3A2	46	B7AD	78	D516
15	C54F	47	A169	79	E132
16	456E	48	D7B8	80	A65E
17	2D6A	49	3BB2	81	F8D8
18	F78B	50	187A	82	A7E9
19	531F	51	633A	83	79A3
20	7E85	52	989B	84	2E35
21	ADFF	53	7A3E	85	BB36
22	EA96	54	3E9F	86	9563
23	39AD	55	A3E3	87	437A
24	69A2	56	B969	88	3AEB
25	5692	57	D398	89	694E
26	3F48	58	F1E7	90	AE8F
27	18F2	59	19D6	91	CDE4
28	B869	60	2FBA	92	45F8
29	3782	61	5FAE	93	5953
30	F839	62	9B4F	94	3E24
31	D694	63	A782	95	D25F
32	E7F2	64	32E5	96	E4A5

Правила оформления отчетов к лабораторным работам

Отчет, является документом, отражающим результаты и ход выполнения лабораторной работы. Отчет должен содержать следующие пункты:

1. Титульный лист – содержащий тему и номер лабораторной работы, фамилии выполнявшего студента и проверявшего преподавателя (пример в конце приложения).
2. Цель работы – указывается цели выполняемой работы.
3. Задачи – указываются задачи, решаемые в ходе выполнения лабораторной работы, приводится и расшифровывается собственный вариант задания.
4. Выполнение работы – указываются расчеты, проведенные в ходе работы, а также результаты этих расчетов.
5. Результаты работы – прикладывается распечатка, полученных в результате работы схем, модулей, временных диаграмм и результаты замеров временных задержек, возникающих при работе схемы, на каждом переключении с указанием кода переключения.
6. Выводы – пункт содержит перечень решенных в ходе работы задач и выполненных целей, а также проблемы, возникшие в ходе работы.

Обратить внимание:

1. Отчет принимается только в бумажном виде. Т.е. отчет должен быть полностью набран на компьютере и распечатан. Отчеты в электронном виде рассматриваться не будут.
2. Листы отчета должны быть пронумерованы и скреплены между собой.

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

ФГБОУ ВО «АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

КАФЕДРА ВТиЭ

Отчет по лабораторной работе №_
по курсу «Схемотехника ЭВМ»

« Тема лабораторной работы »

Выполнил:

студент _____ группы
_____Иванов И.И.

Проверил: доцент

_____Шмидт В.В.

Барнаул 2016