Лабораторная работа №1

Комбинационные логические схемы

ЧАСТЬ 1
ВВЕДЕНИЕ В СИСТЕМУ МОДЕЛИРОВАНИЯ DEEDS И ИЗУЧЕНИЕ ФУНКЦИЙ ЛОГИЧЕСКИХ ВЕНТИЛЕЙ

ВВЕДЕНИЕ

Для ведущих профильных мировых технических университетов (MIT,Sun Jose,МИЭТ и др.) характерно применение в курсе Схемотехники языков описания аппаратуры(Hardware Description Language –**HDL**), Систем Автоматизации Проектирования (**САПР**) и сочетание математического (моделирование) и физического экспериментов с использованием отладочных плат с Программируемыми Логическими Интегральными Схемами (**ПЛИС**). При этом используется вариант построения курса, соответствующий современному способу проектирования цифровой аппаратуры, который начинается с описания проектируемого объекта на HDL, верификации проекта моделированием, синтеза и т.д.

Эти этапы представлены в табл.1. Слева этапы синтеза проекта, справа этапы его верификации.

Таблица 1

Пример последовательности этапов проектирования и верификации проектов цифровой аппаратуры с применением САПР фирмы XILINX

Этапы проектирования устройств	Этапы верификации проекта
Создание проекта, ввод исходного функционального описания проекта на HDL (Design entry)	Функциональная верификация (Behavioral simulation)
Логический Синтез (Synthesis) – автоматизированное создание схемы проекта	Логическое моделирование (Simulate Post-Translate Model)
Конструкторско-технологическое проектирование (Implementation) – размещение элементов схемы, трассировка связей	Временное моделирование (Post place & routing simulation)
Программирование (конфигурирование) микросхемы ПЛИС	Верификация схемы на отладочной плате

В ВУЗах, где «Схемотехника» является менее приоритетным предметом, используется более простой в обучении традиционный подход: графическое описание схем и временных диаграмм их работы с помощью графического редактора и проверка проектов схем моделированием на персональных компьютерах (ПК). Представленный цикл лабораторных работ базируется на этом подходе. Однако применение рассматриваемой ниже учебной системы моделирования **DEEDS**[1], в которой имеются средства стыковки с промышленной САПР ПЛИС позволяет желающим более глубоко изучать этот курс в соответствие с программами ведущих университетов.

Учебная система моделирования Deeds – это система для обучения проектированию цифровых систем [1,2]. Разработка университета г.Генуя (Италия) свободно скачивается (примерно 40 мегабайт) с сайта [1] и легко устанавливается на домашнем компьютере пользователя с операционной системой от Microsoft.

DEEDS состоит из трех подсистем.

1.***Подсистема моделирования цифровых схем*** **–d–DcS**
2.***Подсистема моделирования конечных автоматов*** **–d–FsM**
3.***Подсистема моделирования микроэвм*** **–d–McE**

В данном цикле работ при изучении комбинационных схем используется подсистема моделирования цифровых схем **d–DcS.**
Краткую справку о работе с d–DcS в найдете в приложении в конце описания работы №1.

Понятие цифрового сигнала и его моделей

Реальные сигналы в электронных устройствах имеют сложную форму и обладают множеством характеристик: уровнем напряжения, силой тока, мощностью и т.д. При их изучении используются упрощенные представления **– *модели.***
На рис. 1.1 слева представлены временная диаграмма сигнала, а справа его цифровой модели **– *цифрового сигнала*** (по вертикали– напряжение U, по горизонтали– время t).

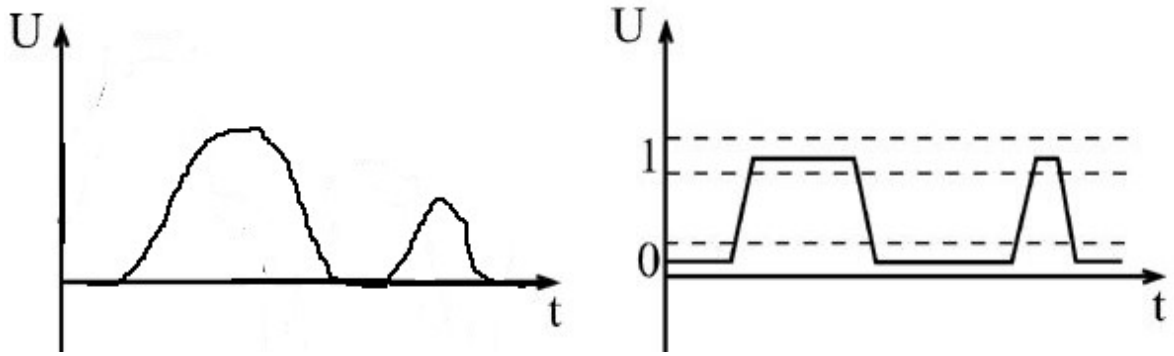


Рис.1.1. Реальный сигнал и его цифровая модель

В примере на рис.1.1 цифровой сигнал принимает два возможных значения– 0 и 1.

Задержкой распространения сигнала в логическом элементе называют время между перепадом сигнала на входе и вызванным им перепадом сигнала на выходе.

На рис.1.2 представлена задержка выходного сигнала Y(***T01- задержка переключения из 0 в 1 и T10- задержка переключения из 0 в 1***) относительно входного сигнала X в логическом элементе **НЕ** (вентиле НЕ). Заштрихована область переходного процесса - ***неопределенное X*** значение сигнала.

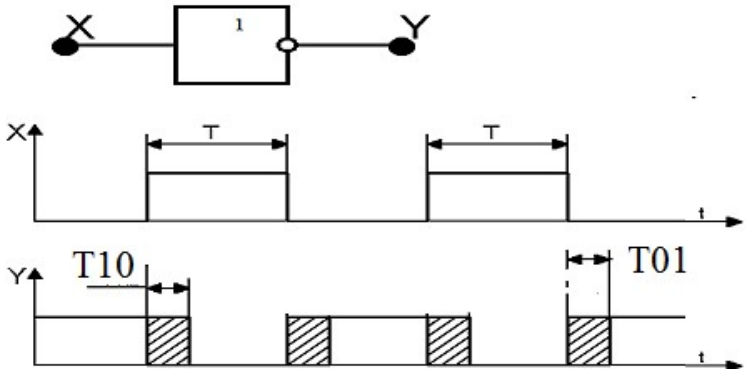


Рис.1.2.Пример временной диаграммы элемента НЕ Y=~X ,
иллюстрирующей эффект задержек сигнала

Для отображения задержки распространения сигнала используются следующие модели задержек:

нулевая задержка(задержка не учитывается- элемент как бы не обладает задержкой,это свойство Булевской мдели)),

единичная задержка (все элементы обладают задержкой равной 1);

средняя задержка (Тсреднее = (T01+T10)/2);

дифференциальная: разные задержки перепадов из 0 в 1 и наоборот (T01≠T10);

случайная минимаксная, учитывающая разброс задержек (min–max) в разных экземплярах элементов.

Кроме того, реальные логические элементы обладают ***инерционными свойствами*** и соответственно модели их задержек могут быть: ***инерционными или транспортными***.

Логические элементы с ***инерционной задержкой*** не реагируют на входные сигналы, длительность которых меньше некоторого порога, например равного времени задержки этого элемента.

А логические элементы с ***транспортной задержкой*** реагирую на входные сигналы любой длительности.

Алфавит моделирования

Алфавит моделирования определяет множество различимых значений сигнала. Соответственно этой значности различают разные ***алфавиты моделирования.***

Булевский. Традиционным и простым алфавитом является Булевская (двоичная) модель сигнала. В ней все сигналы могут принимать два возможных значения: логическая «1» (TRUE/ИСТИНА) и логический «0» (FALSE/ЛОЖЬ). Однако этот алфавит не позволяет описывать сигналы, например принимающие неопределенные значения или значение высокого импеданса. Поэтому в развитых цифровых системах моделирования и языках описания аппаратуры (HDL) применяют ***многозначный алфавит*** представления значений сигналов: четырехзначный (язык Verilog [5,6,9]) или девятизначный (язык Vhdl [5,6,9]).

Четырехзначный алфавит характеризуется значениями: **0,1, X** – неопределенное значение, **Z** – высокий импеданс (высокое сопротивление).

Девятизначный алфавит .В девятизначном алфавите различают значения: **'U'–** начальная неопределенность,**'0', '1','X'–**неопределенно,**'Z'–** высокий импеданс,**'H'–** слабая '1',**'L'–**слабый '0',**'W'–** слабая неопределенность, **'~' –** безразличное значение.

Задания к лабораторной работе №1

Задание 1. Освоение системы ДИИДС .Изучите краткое руководство по системе ДИИДС **d–DcS** (**см. Приложение** в конце этого сборника).

1.1.Постройте модель вентиля НЕ (инвертор)

1.2.проведите анимацию

1.3.получите временные диаграммы (в отчете все это должно быть)

1.4.определите задержку элемента

1.5.постройте модель пользовательского блока из двух вентилей НЕ, соединенных последовательно, выполните для нее пункты 1.3 и 1.4,

1.6.получите генерируемый ДИИДС vhdl код модели вентиля НЕ.

Задание 2. **Изучение функций логических элементов и их ДИИДС моделей** .

Булевские таблицы истинности логических функций (операций) НЕ–NOT, И–AND, НЕ_И–NAND, ИЛИ–OR, НЕ_ИЛИ–NOR, ИСКЛЮЧИТЕЛЬНОЕ_ИЛИ–XOR, ИСКЛЮЧИТЕЛЬНОЕ_НЕ_ИЛИ–XNOR представлены на рис.1.3.

Вход	Вых
A	NOT
0	1
1	0

Входы		Выходы Булевских функций					
A	B	AND	NAND	OR	NOR	XOR	XNOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1

Рис. 1.3. Булевские таблицы истинности логических функций

Таблицы истинности логических функций ***НЕ (NOT), И (AND), ИЛИ (OR) и Тристабильного буфера*** в четырехзначном алфавите представлены на рис. 1.4.

НЕ
NOT

A	NOT A
0	1
1	0
X	X
Z	X

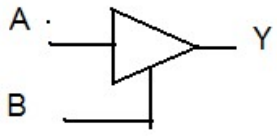
И
AND

ИЛИ
OR

БУФЕР
тристабильный

A	B	И				ИЛИ				БУФЕР тристабильный				B
		0	**1**	**X**	**Z**	**0**	**1**	**X**	**Z**	**0**	**1**	**X**	**Z**	
0		0	0	0	0	0	1	X	X	**Z**	**Z**	**Z**	**Z**	**0**
1		0	1	X	X	1	1	1	1	1	0	X	X	**1**
X		0	X	X	X	X	1	X	X	X	X	X	X	**X**
Z		0	X	X	X	X	1	X	X	X	X	X	X	**Z**

БУФЕР
тристабильный



При B=0 на выходе Y буфера Z,иначе значение входа A

Рис. 1.4. Таблица истинности функций И,ИЛИ,НЕ в четырехзначном алфавите

Постройте модель для индивидуального варианта из таблицы 1.1. Проведите анимацию и получите временные диаграммы, определите задержку элемента, постройте таблицу истинности функции логического И в 4–значном алфавите. В ДЕЕДС значение X подается на неподключенный вход элемента, значение Z можно подать с выхода тристабильного буфера.

Таблица 1.1 Индивидуальные варианты задания 2

№ варианта	функция
1	NOT

2	AND2
3	NAND2
4	OR2
5	NOR2
6	XOR2
7	XNOR2
8	EXOR2
9	Тристабильный буфер
0	NOR2

Задание 3. Изучите задержку и инерциальные свойства модели вашего элемента(см. задание 2),инерциальные - подавая сигналы длительностью меньшей, чем задержка элемента**.**

Вопросы к защите

1. В чем преимущества двоичных сигналов перед аналоговыми?
2. Из скольких строк максимально состоит булевская таблица истинности логической функции от двух аргументов?
3. Как вывести временную диаграмму на печать?
4. Какие типы логических элементов имеются в библиотеке d–DcS?
5. Чем режим анимации отличается от режима временного моделирования?
6. Какие величины задержек моделей логических элементов ДЕЕДС?
7. Как выполнить рестарт моделирования?
8. Как измерить задержки элементов ?
9. Как вывести схему на печать?
10. Что дает режим подсказки HELP в ДИИДС?
11. Как создать пользовательский блок?
12. На временной диаграмме рис.1.2 начальное значение выходного сигнала Y заштриховано- что это означает?
13. Как вывести временную диаграмму на печать?
14. В чем отличия функций or и exor?
15. Моделирование в D–DcS ведется не в двоичном, а в четырехзначном алфавите. Сопоставьте таблицы функций И в двоичном и троичном алфавите.
16. Какая модель задержек сигналов в элементах реализована в ДИИДС (инерционная или транспортная)?
17. Чем отличается модель задержек сигналов в элементах, реализованная в ДЕЕДС от задержек в реальных элементах?

18. Всегда ли цепочка из N однотипных реальных элементов имеет задержку большую, чем цепочка из N-1 таких же элементов ? А как обстоит дело в ДИИДС?

19. Какие отладочные средства можно использовать внутри пользовательского блока или в схемном проекте?

20. Назовите основные форматы файлов, которые можно создать с помощью подсистемы D–DcS? В чем их различия?

Часть 2 .
ПРОЕКТИРОВАНИЕ КОМБИНАЦИОННЫХ ЛОГИЧЕСКИХ СХЕМ С ПРИМЕНЕНИЕМ БУЛЕВОЙ АЛГЕБРЫ

Целями лабораторной работы являются знакомство с традиционным подходом к проектированию комбинационных схем на уровне вентилей , ***Схема*** представляет собой множество соединенных друг с другом логических элементов.

Логические схемы можно разделить на два основных типа: ***комбинационные и последовательностные*** [4–6]**.**

Комбинационной называют такую схему, у которой выходные сигналы в момент времени **T** однозначным образом зависят только от текущего состояния входных сигналов в этот же момент времени.

Последовательностной называют такую схему, у которой выходные сигналы в момент времени **T** зависят не только от текущего состояния входных сигналов, но и от их предыдущего состояния в предыдущий момент времени **T-1.**

Критерии оценки качества проекта схемы

Сложность схемы. Оценивается количеством оборудования, из которых состоит схема и обычно измеряется числом элементов, используемых в схемах.

Задержка *схемы*. Оценивается максимальной задержкой распространения сигнала при прохождении его от входа схемы к выходу(***критический путь***). В комбинационной схеме можно вычислить эту задержку по формуле **Tсх= Tз*L**, где **Tз** – задержка логического элемента. **L** – количество уровней комбинационной схемы или длина критического пути.

Этапы проектирования комбинационных схем на уровне логических вентилей

Проектирование комбинационных схем традиционным (неавтоматизированным) способом сводится к следующим этапам.

1.***Построение таблицы истинности***, определяющей зависимость значений выходных сигналов от входных.

2.***Составление системы булевских уравнений***, описывающих эти зависимости. Для каждого выходного сигнала строится ***Совершенная Ди-***

зьюнктивная Нормальная Форма (СДНФ) или Совершенная Конюнктивная Нормальная Форма (СКНФ).
3.***Выбор логического базиса*** (серии элементов) для построения комбинационной схемы, если он не задан по условию и
выбор критериев оценки качества проектирования, например– уменьшение общего количества используемых элементов и/или количества микросхем в проектируемой схеме или задержек распространения сигналов.
4. ***Минимизацию системы булевских уравнений(*** если она является одним из критериев проектирования. см.приложение 2.1).
5. ***Покрытие этих уравнений функциями элементов используемого логического базиса*** с учетом нагрузочных соотношений.

Обычными критериями качества проектирования являются минимизация количества элементов схемы и минимизация задержки выходного сигнала.

Логический базис

Помимо булевского логического базиса(***И,ИЛИ,НЕ***) существуют ***универсальные логические базисы****:* ***И–НЕ*** *и* ***ИЛИ–НЕ***. То есть всю логическую схему такие универсальные базисы позволяют построить только используя один такой тип элемента.

Задание 1. Постройте схему (рис.2.1), реализующую функцию 3И(AND3) на вентилях 2И(AND2) и ее тест стенд. Т.е добавьте источники входных сигналов и средства наблюдения выходных. Для нашего случая это двухуровневая схема из двух вентилей 2И, длина критического пути которой равна 2.

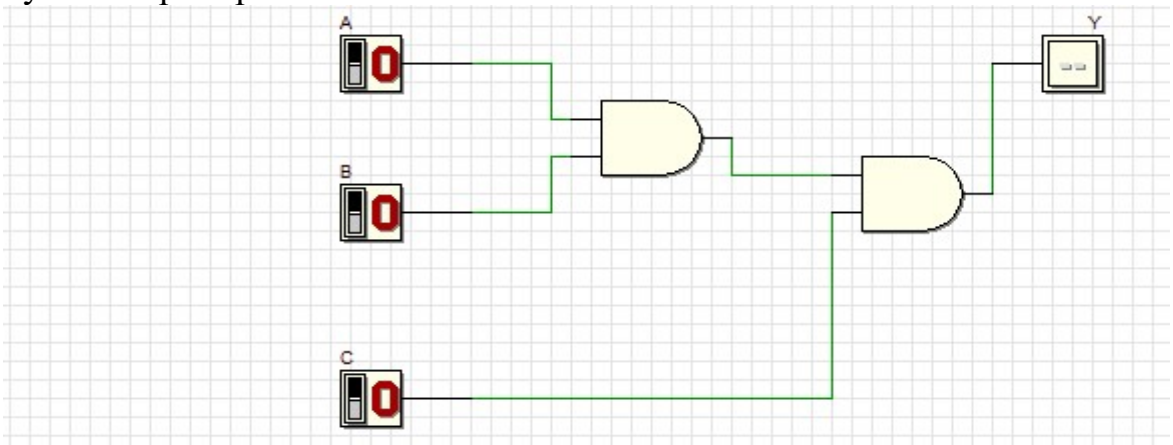


Рис.2.1. Схема, реализующая функцию 3И на вентилях 2И

1.1.Проведите на ней модельный эксперимент, доказывающий правильность работы этой схемы.
12. Получите временную диаграмму,

1.3.VHDL описание схемы и прокомментируйте ее код. При наличии САПР ПЛИС и отладочных плат повторите анимационный эксперимент на плате.

Задание 2. Постройте схему, реализующую функцию ***Y=F(A,B,C,D),*** соответствующую варианту задания и проверьте ее моделированием (табл. 2.4, в двоичном алфавите, для всех возможных комбинаций сигналов на входах). Определите задержку и количество оборудования. Получите VHDL код.

Таблица 2.1. Варианты функций для задания 5

№ варианта	функция
1	Схема определения того, что трехразрядный двоичный код симметричен относительно среднего бита
2	Определение, является ли трехразрядный двоичный код составным числом (не является простым числом)
3	Определение, четное или нечетное число единиц в трехразрядном коде
4	Мажоритарная функция 2 из 3 для трехразрядного кода
5	Схема определения того, что трехразрядный двоичный код делится на 3 без остатка
6	Определение, есть ли комбинация 11 в трехразрядном коде
7	Определение, есть ли комбинация 00 в трехразрядном коде
8	Перевод трехразрядного кода Грея в обычный двоичный код числа
9	Перевод трехразрядного кода двоичного кода числа в код Грея
0	Схема определения наличия последовательности 01 в четырехразрядном двоичном коде

Задание 3. Проектирование устройств с применением Программируемых Логических Матриц (ПЛМ).

Программируемые логические интегральные схемы (ПЛИС) - это одно из перспективных направлений развития элементной базы. Прототипами современных ПЛИС являлись микросхемы ***ПЛМ – программируемые логические матрицы (PLA****–Programmable Logic Array*) и ПМЛ – программируемые матрицы логики (***PAL****–Programmable Array Logic*). ***ПЛМ*** – комбинационная схема, которая имеет N входов и M выходов. Пример- входы IN и выходы OUT на рис 2.3. Она состоит из двух блоков– ***матрицы И и матрицы ИЛИ*** (рис.2.3), а так же входных и выходных буферов. ***Матрица И*** представляет собой совокупность из **К** штук **2*N**– входовых схем **И,** образующих строки матрицы (термы). На стро-

ки из входных буферов поступают прямые и инверсные значения внешних входов ПЛМ. Выходы каждой из этих строк можно соединить с входами **К**– входовых схем **ИЛИ**, образующих столбцы **матрицы ИЛИ**, их **М** штук. Выходы этих схем ИЛИ и являются выходами ПЛМ.

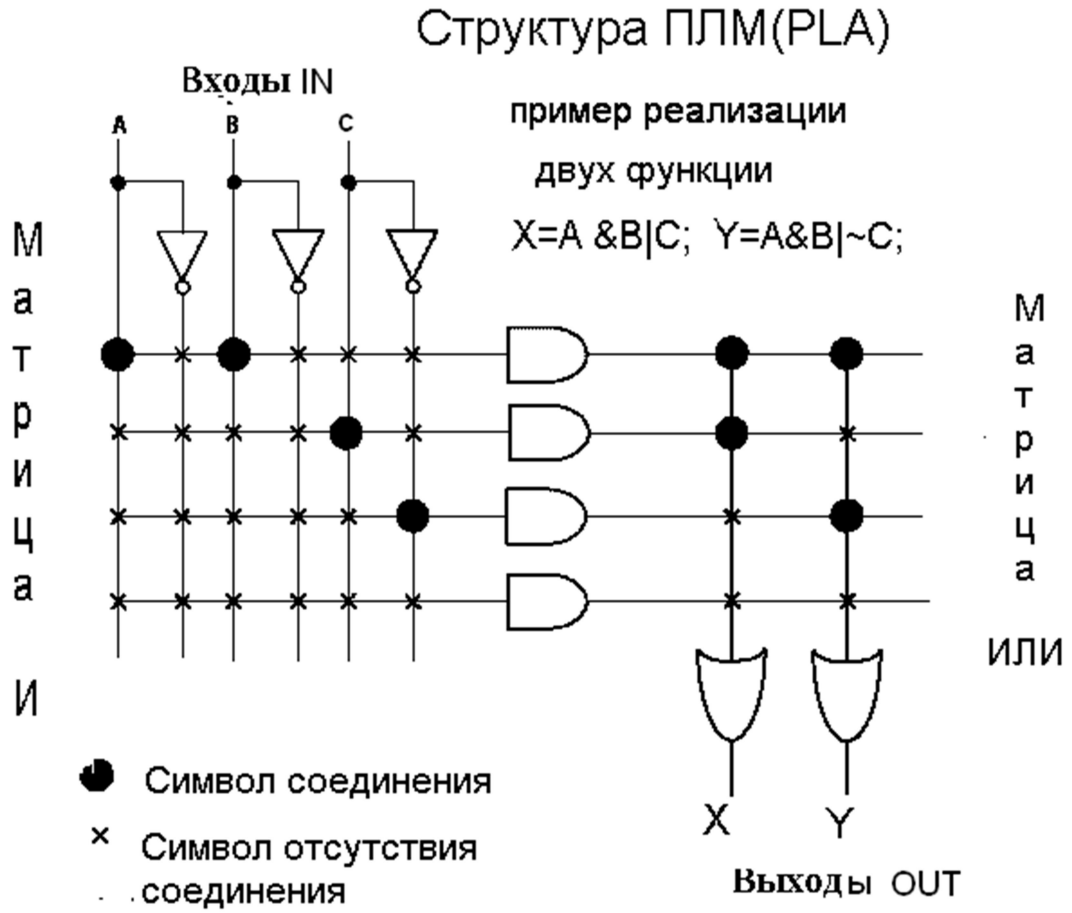


Рис.2.3. Структура ПЛМ и пример реализации функций
X=(A & B) |C и Y=(A & B) | ~ C

.

При программировании ПЛМ каждый вход каждой схемы **И** может быть соединен с прямым или инверсным (рис.2.3) значением соответствующего входного сигнала или быть разомкнут, что соответствует значению логической 1 на этом входе. Итак, **матрица И** имеет **2*N столбцов** и **К строк**, а **матрица ИЛИ** – **К строк** и **М столбцов**. Нетрудно провести аналогию с реализацией СДНФ произвольной логической функции на элементах И, имеющими N входов, элементах ИЛИ, имеющими К входов и N элементов НЕ.На рис.2.3 приведен пример реализации двух логических функций(& | ~- обозначения логических операций И, ИЛИ,НЕ)
X=(A & B) |C ; Y=(A & B) | ~ C;

В общем случае на ПЛМ с N–входами и М– выходами можно реализовать М разных N – входовых логических функций, число термов в которых ограничивается К.

В библиотеке **SXEM_LAB_DEEDS в каталоге PLM** имеются пользовательские блоки- модели ПЛМ . Необходимо используя их создать собственные пользовательские блоки ДИИДС путем замыкания пересечений столбцов и строк матриц и таким образом реализовать на этой элементной базе и проверить моделированием ваш проект , выполненный ранее **в задании 2** на обычной элементной базе ДИИДС.

Контрольные вопросы

1. Что такое комбинационная схема? Назовите основные ее особенности? Объясните отличие от последовательностных схем?
2.Назовите и кратко опишите основные этапы проектирования комбинационной логической схемы.
3.Объясните, когда удобнее применять СДНФ, а когда СКНФ? Какая из этих форм представления будет эффективнее для элемента ИСКЛЮЧАЮЩЕЕ 3-ИЛИ(XOR3)?
4.Какие способы минимизации вам известны?
5.Назовите критерии оценки качества комбинационной схемы.
6.Сколько элементов 2ИЛИ понадобится для реализации функции 3ИЛИ?
7. Какие задержки имеют логические элементы в D–DcS?
8. Постройте схему , реализующую функцию 2И на элементах 2И–Не(NAND2)?
9. Что такое коэффициент разветвления элемента по выходу?
10. Какие задержки имеет выходной сигнал в схеме задания 2?
11.Четыре элементов 2И–НЕ понадобилось для реализации функции 3И – можно ли уменьшить их число?
12. Какая из двух форм представления данной в задании 3 функции использует меньше логических операций?
13. Какая форма представления в задании3 функции эффективнее в случае, когда число единиц на выходе таблицы истинности функции намного меньше числа нулей?
14.Какие логические базисы помимо базиса И,ИЛИ,НЕ можно использовать для представления логической функции?
15. Какие задержки имеет выходной сигнал в вашем варианте задания 4?
16. С формулируйте теорему Де Моргана и проиллюстрируйте ее применение на примере.
17.Какие преимущества и недостатки минимизированной схемы(оборудование, задержки)?

18. Исследуйте возможность использования другого логического базиса для реализации минимизированной схемы задания 4?

19.Запишите формулу Де Моргана.

20. Опишите на VHDL ваш проект задания 2, используя структурное описание архитектуры.

21. Назовите известные вам операторы языка VHDL.

22. Назовите особенности применения VHDL оператора назначения сигнала «<=».

23. Что такое ПЛИС и ПЛМ?

24.Что такое интегральная микросхема

25. Какую логическую функцию реализует микросхема КР1533ЛА3 ?

26.Какие критерии проектирования устройств с применением интегральных микросхем?

27. Что такое модель константных неисправностей?

Приложение 1.1 **КРАТКИЙ СПРАВОЧНИК**

по подсистеме моделирования цифровых схем– D-DcS

Учебная система цифрового моделирования DEEDS *разработана в Генуэзском Университете под руководством Д.Донзеллини и свободно скачивается с официального сайта [1].*

Система DEEDS покрывает следующие области изучения методов проектирования и моделирования цифровых систем:

Комбинационные схемы **(**от вентилей И,ИЛИ,НЕ до дешифраторов, сумматоров, АЛУ)

Последовательностные схемы **(**от триггеров и защелок до регистров, счетчиков и памятей)

Моделирование Конечных Автоматов(FSM)

Программирование на языке Ассемблера (Интел 8080)

Моделирование микропроцессорных систем на базе микроэвм

Изучение основ языка описания аппаратуры VHDL

Применение отладочных плат с ПЛИС типа FPGA

DEEDS включает три подсистемы*::*

Deeds–DcS *(Digital Circuit Simulator– система моделирования цифровых схем),*

Deeds–Fsm *(Finite State Machine Simulator– система моделирования автоматов),*

Deeds–Mce (Micro Computer Simulator–подсистема моделирования микроэвм)

На рис. П.1представлен экран главного меню *Deeds–DcS*– окно проекта схемы с подменю команд.

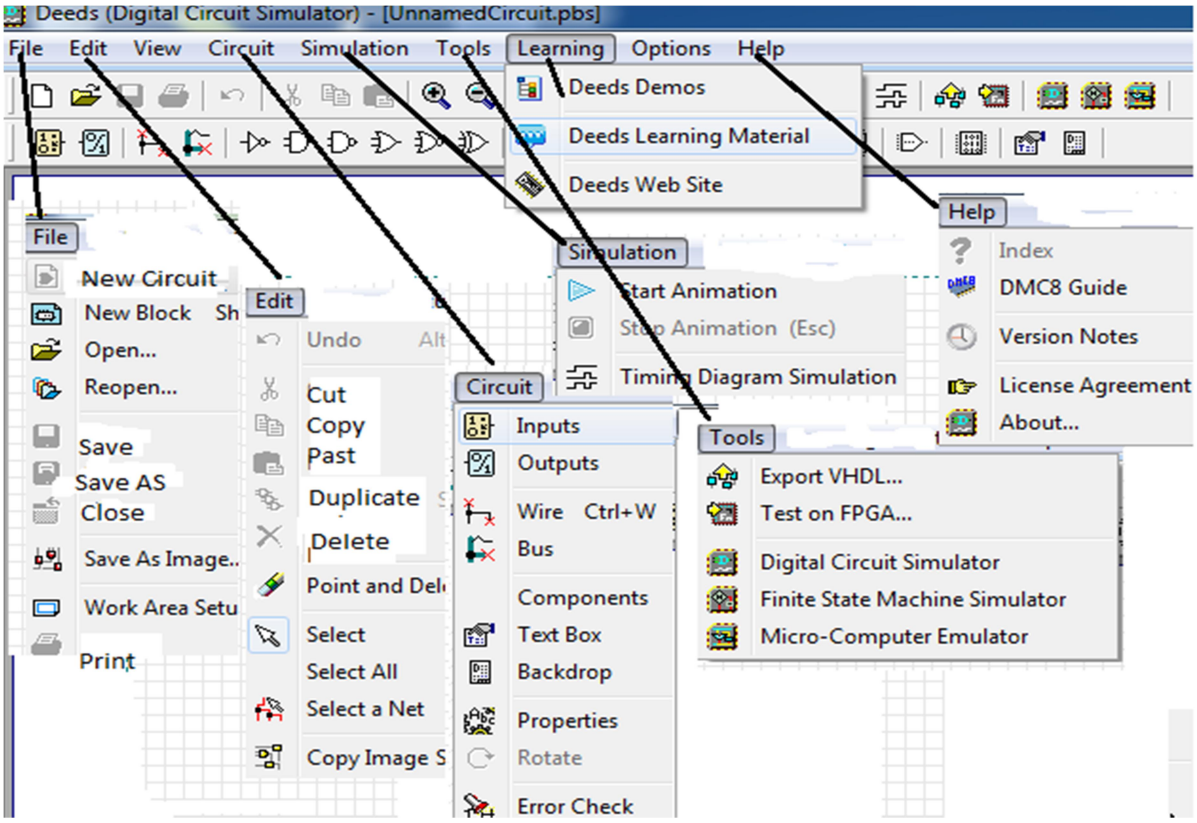


Рис. П.1. Главный экран подменю команд

В последней версии ДИИДС расширен список HELP документов- инструкций по работе с отдельными обьектами - не только с системой моделирования микроэвм DMC8 , но и с блоками памяти и тп. В фукции Circuit появилась опция введения неисправности в схему .

На рис. П.2 представлен экран главного окна *Deeds–DcS* с пояснениями функций пиктограмм отдельных типов встроенных блоков, используемых как компоненты схем проектов.

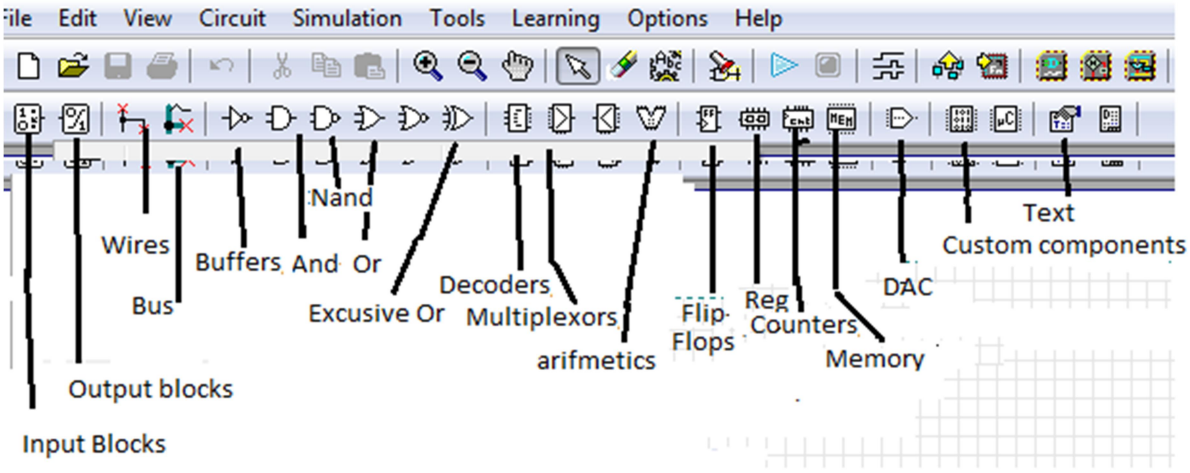
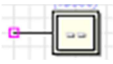


Рис. П.2. Панель быстрого доступа

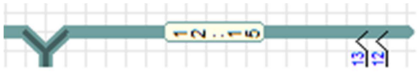
Входные сигналы. Подача входных сигналов производится с помощью: одиночного переключателя , группы переключателей, кнопки, генератора тактового сигнала, имеющего диапазон изменения частоты, источников константного значения сигнала(0,1),цифрового шестнадцатеричного источника. Имеется возможность использования индивидуальных соединений и шинных(bus).

Выходные и внутренние сигналы. Информацию о состоянии выходных, а также внутренних сигналов можно отображать на:

светодиодном индикаторе  , шестнадцатеричном табло, контрольном светодиоде **Test LED**  , контрольной точке **TEST POINT**  Контрольный светодиод применяется в режиме анимации, а контрольная точка – в режиме временного моделирования. Имеется возможность использования индивидуальных соединений и шинных(bus).

.

Межкомпонентные соединения: –провода(**wires**) , –шины (**bus**) , шинные разветвители(**bus splitter**)  и отводы(**bus tap**)  .Пример фрагмента схемы с шиной, разветвителем и отводами ниже

 .

Обозначения триггерных и регистровых блоков.

Триггеры разделяют на ***асинхронные*** (не тактируемые) и ***синхронные*** (тактируемые). Тактируемые триггеры подразделяются на управляемые ***по фронту*** и ***по уровню***. Тактирующие входы(синхросигналы) в свою очередь подразделяются на ***прямые(PET– активен фронт 0–1синхросигнала)*** и ***инверсные(NET активен фронт 1–0 синхросигнала)***.Примеры D–триггеров: 

.

Регистры с параллельным входом–Parallel Input и выходом–Output (**P.I.P.O**). Регистры с параллельным входом и последовательным–Serial выходом (**P.I.S.O**).

Сигналы сброса и установки во всех последовательностных блоках асинхронные.

Создание проекта схемы

Для этого необходимо:

1) Запустить **D–DcS**. 2) Перейти к рисованию новой схемы–последовательность **File/New Circuit**. 3) Переместить в окно проекта (рис. П.3) схемы входные, выходные и логические блоки. 4) Соединить их проводами и шинами– wires and busses.5) Проверить корректность связей– **Circuit/Error Check** или нажать  .6) Сохранить проект(файл

.pbs)– **File/Save AS**.7)Перейти к моделированию проекта с целью его верификации. Ниже пример проекта модели схемы с элементом НЕ.



Рис. П.3. Окно проекта НЕ

Создание проекта пользовательского блока

Для этого необходимо:

1) Запустить **D–DcS**. 2) Перейти к рисованию схемы блока – последовательность **File/New Block**. 3) Переместить в окно проекта (рис. П.4) блока входные, выходные и логические блоки. 4) Соединить их проводами и шинами– wires and busses. 5) Проверить корректность связей– **Circuit/Error Check** или нажать (если на консоли появится сообщение, отличное от **Circuit OK**,место ошибки на схеме выделится красным цветом).6) Сохранить проект(файл .cbe)– **File/Save AS**. Ниже проект модели пользовательского блока НЕ с поясняющим входы–выходы текстом.

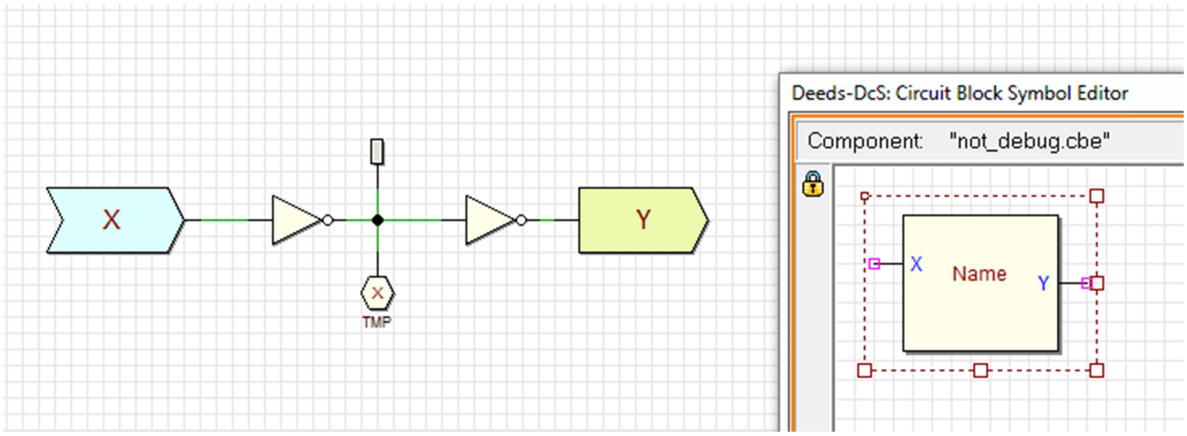


Рис. П.4. Окно проекта пользовательского блока, которому дано имя **not_debug** с добавленными средствами отладки- тестовым светодиодом-**Test Led** и контрольной точкой-**TEST_point** названной **tmp**.Справа-форма редактирования изображения этого пользовательского блока (View/Show Symbol editor)в результате которого выходной сигнал Y перенесен направо.

Создание пояснений в проекте–

Для этого можно использовать: их непосредственное описание в блоке текста (Text Box) либо сопоставлением функциональной схемы со структурной схемой (Backdrop).

Для проверки проекта анимацией необходимо выполнить:
1) **Simulation/Animation** или нажать кнопку пуск . 2)Манипулируя с значениями входных сигналов наблюдать индикаторы выходных.
3)Остановить анимацию нажав кнопку **Stop** .

Для моделирования и получения временных диаграмм сигналов надо выполнить:1) **Simulation/Tim Diagr.** или нажать кнопку –появится окно временных диаграмм (рис. П.3). 2) Нарисовать диаграммы входных сигналов. 3) Нажать кнопку пуск несколько раз и наблюдать выходные сигналы . 4) Остановить моделирование, нажав кнопку **Stop**. 5) Сохранить, если надо временные диаграммы .
Функции пиктограмм на экране временных диаграмм поясняются при подводе курсора мышки. Например для измерения задержек сигналов используются появляющиеся на диаграмме курсоры, вызываемые пиктограммой . На рис. П.5 окно временной диаграммы моделирования вентиля НЕ с курсорами отметки задержки.

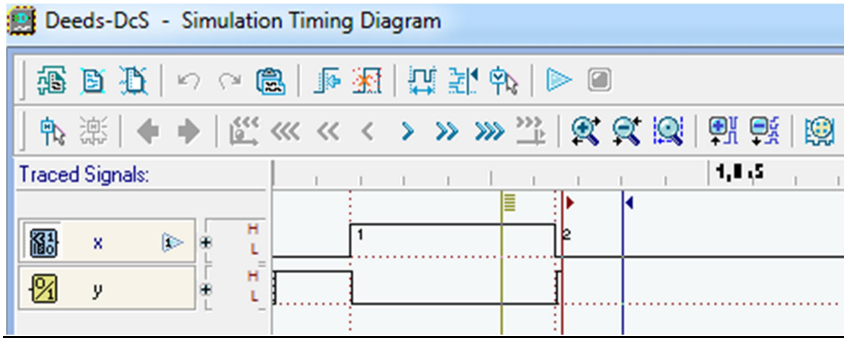


Рис. П.5. Пример окна с временной диаграммой вентиля НЕ

Отладка моделей схем.
Для отслеживания **внутренних сигналов** схемы в **режиме анимации** их надо подсоединять к контрольным светодиодам (**Test LED**) , а при **временном моделировании** к индикаторам контрольных точек(**TEST POINTS**).
На рис. П.6 представлен пример анимации проекта **not_debug_tb**, использующего пользовательский блок **not_debug** с выводом его внутренней схемы на экран(щелчком на его изображении). На внутренней схеме пользовательского блока виден светодиод, отражающий значение 1.

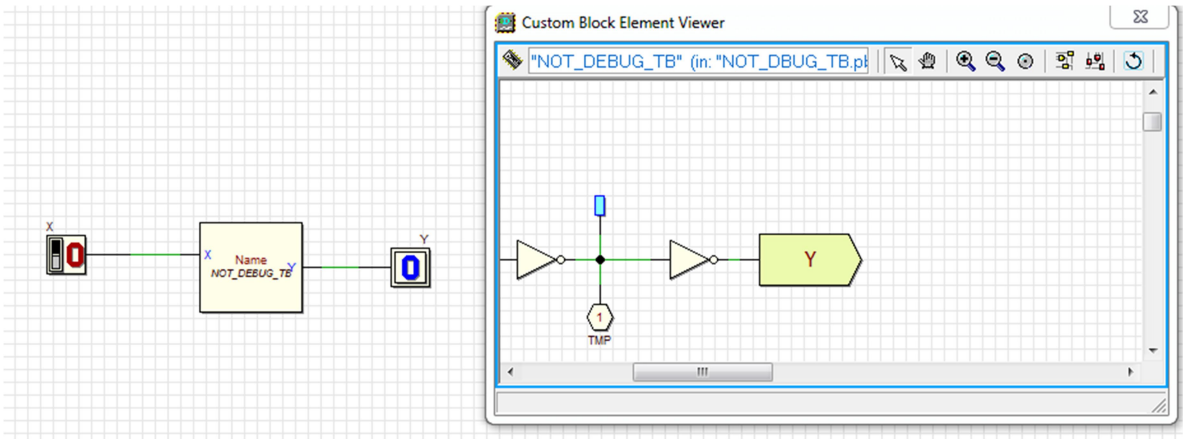


Рис. П.6. Пример анимации с индикацией внутреннего для пользовательского блока тестового светодиода

На рис. П.7 представлен пример подготовки к получению временной диаграммы проекта **not_debug_tb** , использующего пользовательский блок **not_debug** с индикацией на временной диаграмме внутреннего сигнала **TMP** пользовательского блока. Для этого пришлось выбрать в списке сигналов ***not_debug***,в появившемся меню ***Track Handling***, стрелкой налево переместить внутренний сигнал TMP из списка ***Hidden Tracks*** в список наблюдаемых ***Tracks*** .

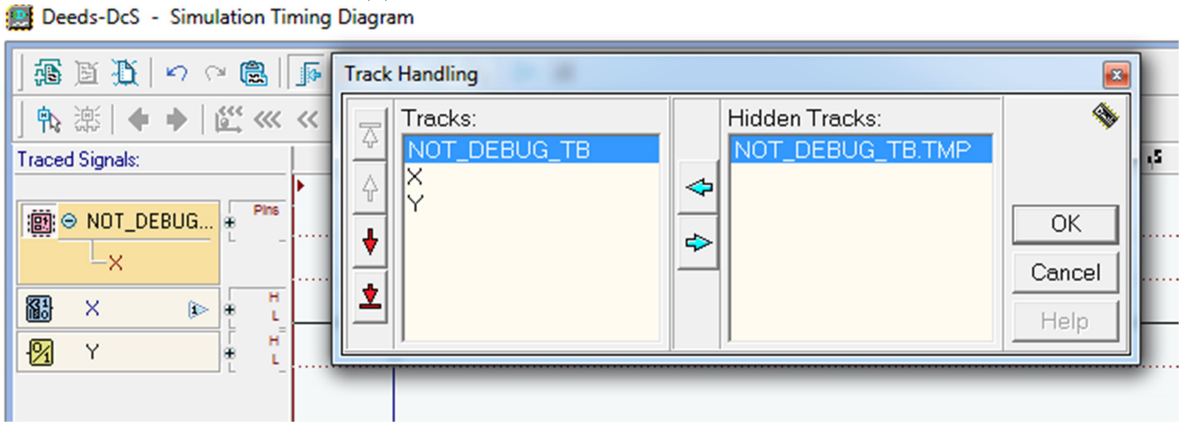


Рис. П.7. Подготовка к перемещению внутреннего сигнала TMP из списка скрытых в список наблюдаемых на временной диаграмме.

На рис .П.8 представлен момент, когда внутренний сигнал TMP уже перенесен в список наблюдаемых на временной диаграмме.

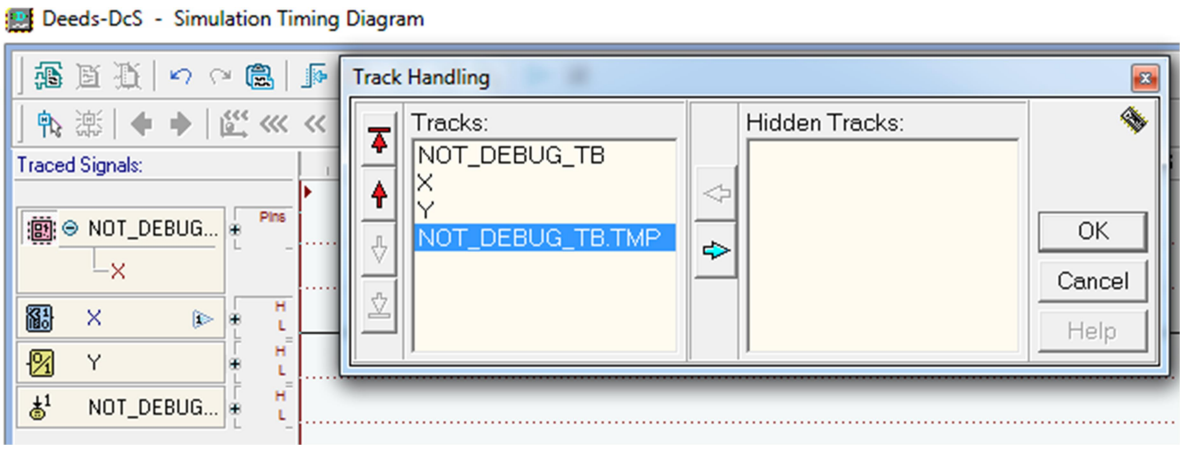


Рис .П.9. Внутренний сигнал TMP перенесен в список наблюдаемых на временной диаграмме

На рис .П.10 показана временная диаграмма с выведенным внутренним сигналом TMP

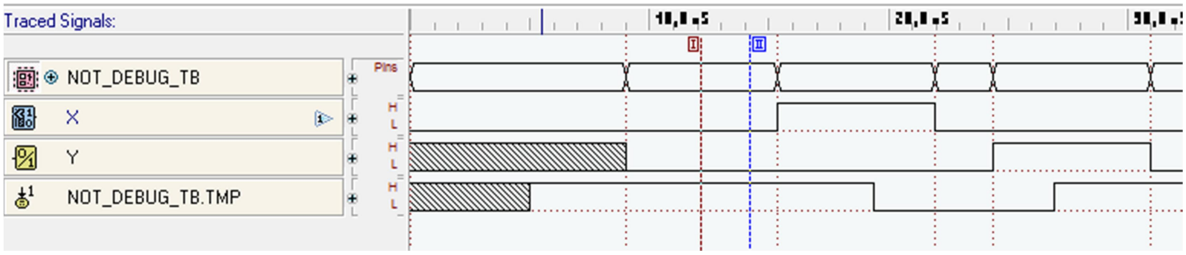


Рис .П.10. Внутренний сигнал TMP виден на временной диаграмме

На рис. П.11 представлен пример схемы с общей шиной, в котором при анимации красным цветом ДИИДС показал, что возникла ошибка на шине из за одновременного включения двух буферов.

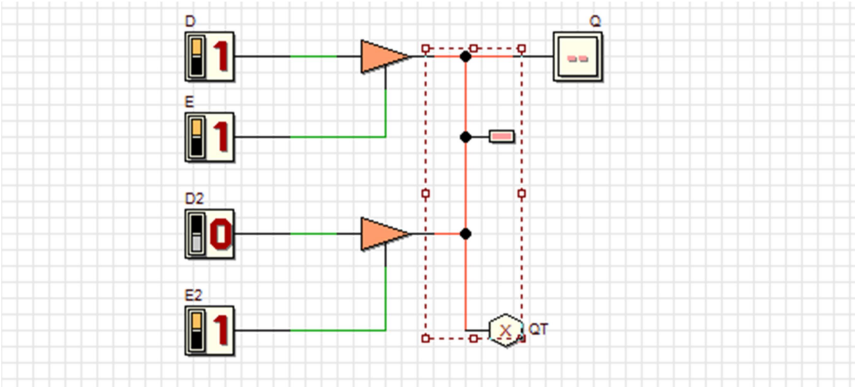


Рис. П.11. Пример схемы с общей шиной на тристабильных буферах с индикацией ошибки при анимации

Внесение исправлений в схему

Для этого **(**см. рис. П.12**)**:щелкните левой кнопкой мыши на интересующем вас обьекте– он покраснеет, щелкните правой– появится окно меню, выберите подменю и в нем нужное действие, например убрать(**Delete**).

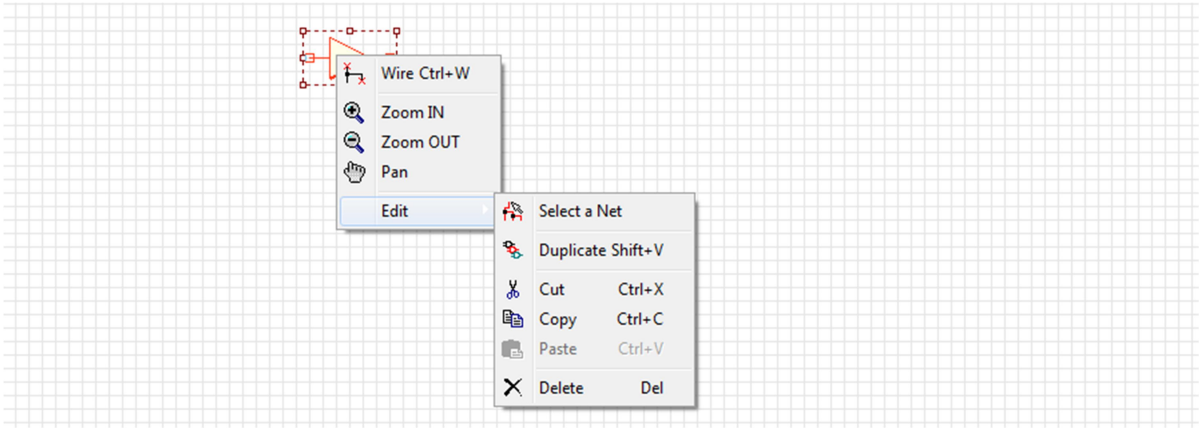


Рис.П.12. Подменю редактирования схемы

Основные параметры моделей сигналов и задержек в ДИИДС

Алфавит моделирования четырехзначный : 0,1,X,Z.
Начальное значение сигналов X.
Выходы невыбранных микросхем памяти=0.
Модель задержки элементов - транспортная.
Для всех элементов одного и того же типа задано среднее время задержки (T01=T10).
В ДИИДС элементы не имеют инерции(транспортная модель задержки).

Внесение неисправностей(Faults)

Моделируются неисправности типа тождественный 0 , тождественная 1 на выходах логических вентилей. Для этого выбрать **Circuit/Fault/set mode on** .Появится список доступных для моделирования неисправностей вентилей и буква F.Выбрать вентиль ,установить в нужное место схемы, появится окно меню с выбором типа неисправности, указать нужную.Исправить схему можно нажав на F.

Получение VHDL– кода модели проекта
Tools/Export VHDL /*войдите в каталог Components*
Ниже VHDL код описания компонента NOT_gate (НЕ)

```
library ieee; use ieee.std_logic_1164.all;
ENTITY NOT_gate IS   PORT( I: IN std_logic;
                           O: OUT std_logic );
```

```
END NOT_gate;
ARCHITECTURE behavioral OF NOT_gate IS    BEGIN O <=
(not I);
END behavioral;
```

Использование отладочных плат с ПЛИС типа FPGA для физического эксперимента с проектом .

На компьютере должен быть установлен САПР QUARTUS II (www.altera.com).

Подключить плату к ПК кабелем(если питание отладочной платы подается не через него– подключить блок питания). На плате загорится индикатор наличия питания.

Выбрать в меню ДЕЕДС режим **Tools->Test on FPGA** (на экране появится меню выбора типа платы).

Выбрать из списка FPGA Board/Brand используемую плату..

Всем входным/выходным сигналам проверяемой схемы сопоставить переключатели–кнопки/ светодиодные индикаторы на плате. Если в проекте используется тактовый сигнал– выбрать вариант пониженной частоты от тактового генератора платы или подачи сигнала от нажимной кнопки.

Выбрать **Generate Project** (создается каталог проекта).

Выбрать **Start Compilation** (начинается синтез схемы в логическом базисе ПЛИС. Предупреждения(Warnings) можно игнорировать.

Выбрать **Programmer**(подготовка файла программирования ПЛИС).

Нажав **Hardware Setup** (подготовка платы к программированию).

Выбрав **Start** начать процесс заливки проекта на плату.

Изменяя значения входных сигналов переключателями проверить индикаторы выходных сигналов на соответствие ожидаемым значениям **.**

Приложение 1.2. Элементы языка VHDL

Язык описания цифровой аппаратуры VHDL позволяет описывать любое цифровое устройство тремя основными способами, используемыми и для синтеза, и для моделирования:

Поведенческое, позволяющее описать алгоритм работы устройства, а не его структуру;

Структурное, позволяющее описать устройство в виде взаимосвязанных компонентов (components) более низкого уровня, то есть описание по иерархическому принципу;

В подсистеме моделирования цифровых схем –d-DcS при генерации ею VHDL описаний схемы создается два файла:

«***Components.vhd***» и «***Имя_Проекта.vhd***».

Первый файл содержит VHDL-описания всех используемых в схеме типов логических элементов(компонентов), описанных в поведенческом стиле.

Второй файл содержит описание связей компонентов. Связывание экземпляров компонентов выполняется с помощью общих сигналов(цепей) и конструкций: «Имя_экземпляра компонента_в_схеме» : «имя_компонента» PORT MAP («список сигналов соединения»);.

Следует отметить, что поведенческое описание может выполняться различными способами, поскольку все операторы делятся на два типа:

параллельные (оператор назначения сигнала «<=», оператор процесса «process», оператор блока block, оператор условного назначения сигналу «when» и другие);

последовательные (оператор назначения сигнала «<=»; оператор присваивания переменной «:=»; конструкции if, case, loop и другие), выполняются внутри процессов, процедур и функций.

Необходимо учитывать разницу между операторами «<=» и «:=»: оператор «:=» выполняет присваивание значения мгновенно, а оператор «<=» назначает новое значение сигнала с некоторой задержкой.

Пример поведенческого описания вентиля НЕ NOT_1

```
library IEEE; use IEEE.STD_LOGIC_1164.all;
  entity NOT_1 is
     port ( X:in std_logic; Y:out std_logic);
  end;
architecture BEH of  NOT_1  is
begin
```

```
  Y <= NOT  X;
end;
```

ПРИЛОЖЕНИЕ 2.1.Законы Булевой Алгебры

Традиционным логическим базисом является Булевский (операции И, ИЛИ, НЕ). Свойства этих операций представлены в табл.2.1.

Таблица 2.1
Законы Булевой алгебры

Название закона	Для операции И	Для операции ИЛИ
Двойное отрицание	$\bar{\bar{A}} = A$	
Коммутативный	$A \& B = B \& A$	$A \vee B = B \vee A$
Ассоциативный	$(A \& B) \& C = A \& (B \& C)$	$(A \vee B) \vee C = A \vee (B \vee C)$
Дистрибутивный	$A \& (B \vee C) = (A \& B) \vee (A \& C)$	$A \vee (B \& C) = (A \vee B) \& (A \vee C)$
Идемпотентности	$A \& A = A$	$A \vee A = A$
Законы с константами	$A \& 1 = A;\ A \& 0 = 0$	$A \vee 1 = 1;\ A \vee 0 = A$
Закон противоречия, исключения третьего	$A \& \bar{A} = 0$	$A \vee \bar{A} = 1$
Закон поглощения	$A \vee (A \& B) = A$ $A \& (A \vee B) = A$	
Закон де Моргана	$\overline{A \& B} = \bar{A} \vee \bar{B}$	$\overline{A \vee B} = \bar{A} \& \bar{B}$

Приложение 2.2. Пример решения задачи средствами ДЕЕДС

Проведите синтез и минимизацию схемы, реализующей простую логическую функцию F(X1,X2,X3),заданную таблицей истинности (табл.2.2), проверьте схему моделированием, определите максимальную задержку сигнала, получите vhdl код.

Таблица 2.2. Таблица истинности функции F

X1	X2	X3	F
0	0	0	0
0	0	1	1
0	1	0	0

0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Полученная СДНФ функции
F= ~X1 & ~X2 &X3 | ~X1 & X2 & X3 | X1 & ~X2 & X3
Ее схема, реализованная в элементном базисе ДИИДС представлена на рис.2.4.

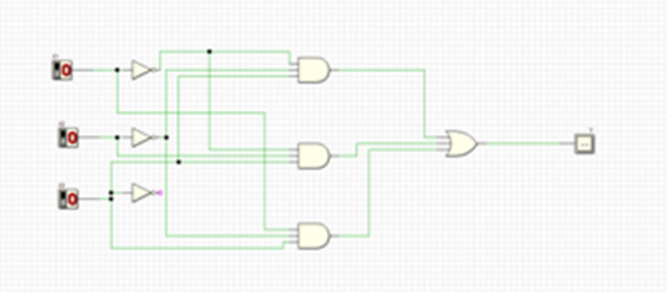


Рис.2.4. Схема, реализующая функцию F в логических блоках ДИИДС .

Приложение 2.3. Функциональные и принципиальные схемы на ИС серии КР1533, выполненные средствами ДЕЕДС

Ниже представлена ***ФУНКЦИОНАЛЬНАЯ схема***, реализующая функцию Y=(~(A &B) & ~C) микросхемой ЛА3 серии КР1533, выполненная средствами ДИИДС

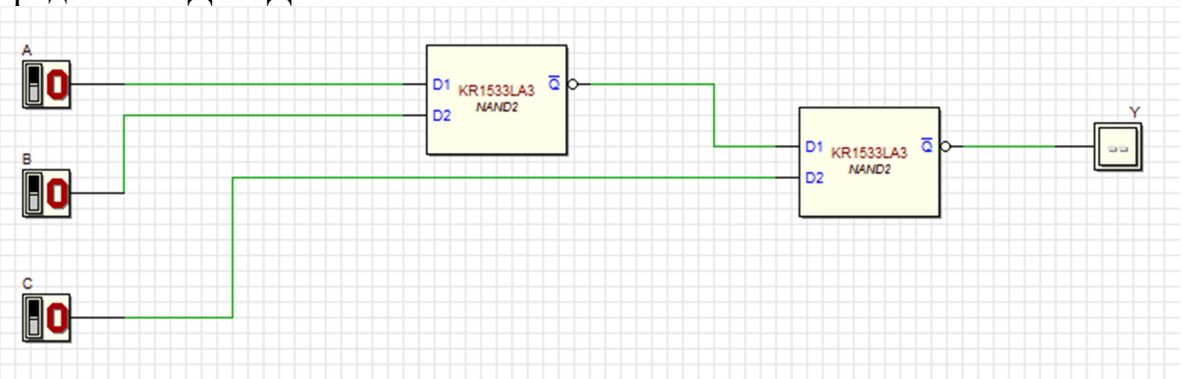


Рис.П.13. Схема, реализующая функцию F микросхемой ЛА3.

Ниже представлена ***ПРИНЦИПИАЛЬНАЯ ЭЛЕКТРИЧЕСКАЯ*** схема, реализующая функцию Y=(~(A &B) & ~C) двумя компонентами 2И-НЕ микросхемы КР1533Ла3,содержащей 4 таких компоненты, выполненная средствами ДИИДС.

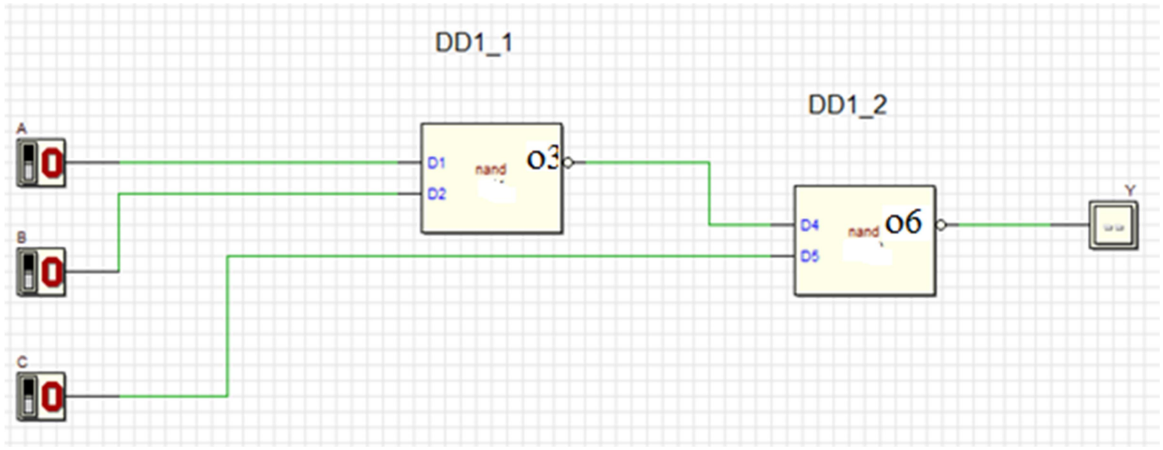


Рис. П.14. Схема, реализующая функцию F микросхемой ЛА3

На принципиальной схеме обозначены номера контактов используемых компонент (DD1_1 и DD1_2) микросхемы DD1

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

Основная

1.https://www.digitalelectronicsdeeds.com Официальный сайт разработчиков система моделирования Deeds
2.https://www.digitalelectronicsdeeds.com/demos/demopage_comb.html
На сайте представлены примеры проектов с пояснениями.
3. https://www.digitalelectronicsdeeds.com/learningmaterials/labtopics.html
Представлены описания нескольких десятков лабораторных работ по разделам схемотехники, теории автоматов и микропроцессорных систем.
4.Новиков Ю.В. Введение в цифровую схемотехнику. Учебное пособие. –М: Бином. 2007.
5.Харрис Д.М., Харрис С.Л . Цифровая схемотехника и архитектура компьютера. Электронная версия книги может быть загружена с сайта http://easyelectronics.ru/files/Book/digital-design-and-computer-architecture-russian-translation.pdf
6. Угрюмов Е.П. Цифровая схемотехника. Учебное пособие. Издание 3. СПБ.:BXB Петербург, 2010.

Дополнительная

7.Потемкин И.С. Функциональные узлы цифровой автоматики. –М.: Энергоатомиздат,1988.
8. Петровский И.И. и др. - Логические ИС Кр1533, Кр1554. Справочник. –М.:БИНОМ, 1993.
9. Поляков А.К. ЯЗЫКИ VHDL и VERILOG В ПРОЕКТИРОВАНИИ ЦИФРОВОЙ АППАРАТУРЫ НА ПЛИС.–М.: Изд.Дом МЭИ,2012.