|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего профессионального образования"Московский технологический университет" **МИРЭА** | |
| Институт информационных технологий (ИТ) | |
| Кафедра вычислительной техники (ВТ) | |

|  |  |  |
| --- | --- | --- |
| **ОТЧЕТ**  **О выполнении лабораторной работы №1** | | |
| **по дисциплине** | | |
| **«**Моделирование**»**  *Вариант 14* | | |
|  | | |
| Выполнил студент группы ИВБО-03-14 | *Кривошея М.С.* | |
| Преподаватель | *Казанцева Л.В.* | |

**Содержание:**

Введение. 3

1. 1 МЕТОД. (ДНФ)

1.1. Карты Карно и формулы, построенные по ним. 4  
1.2. Схема ДНФ из Schematic editor. 5

1. 2 МЕТОД. (КНФ)

2.1.Карты Карно и формулы, построенные по ним. 6

2.2.Содержимое файла KNF.v 6

3. 3 МЕТОД. (Поведенческое описание)

3.1. Содержимое файла BEH.v 7

4. Verilog модель модуля верхнего уровня иерархии.

4.1. Topmodule, содержимое файла TOP.v 8

5. Результаты логической верификации.  
 5.1. Тестовый модуль, содержимое файла Test\_top.v 9

5.2. Временная диаграмма из симулятора ISim. 10  
  
 Заключение. 10

# Введение:

**Техническое задание.**

Спроектировать синтезируемые модели комбинационной схемы 4х4, описанной таблицей истинности согласно варианту задания, тремя различными способами:

2.1. На вентильном уровне, методом карт Карно в виде МДНФ, в схемотехническом редакторе Schematic editor САПР Xilinx ISE Design Suite.

2.2. На вентильном уровне, методом карт Карно в виде МКНФ, на языке описания аппаратуры Verilog.

2.3. На поведенческом уровне, на языке описания аппаратуры Verilog.

Реализовать на языке Verilog тестовое окружение и провести верификацию спроектированных моделей при помощи симулятора iSim из состава САПР Xilinx ISE Design Suite.

Провести апробацию моделей при помощи отладочной платы Digilent Nexys 4 на ПЛИС Xilinx Artix 7 XC7A100T-1CSG324. Комбинации на входах комбинационных схем должны задаваться при помощи движковых переключателей отладочной платы, комбинации на выходах комбинационных схем должны отображаться светодиодами отладочной платы.

**Таблица варианта.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Вар\Вх.** | F | E | D | C | B | A | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| **14** | E | F | 0 | 9 | A | 4 | 1 | 8 | C | 5 | 2 | 7 | D | 6 | 3 | 7 |

Табл.1 Вариант задания.

Строим таблицу истинность по Табл. 1.

**Таблица истинности:**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **HEX** |  |  |  |  | **HEX** |  |  |  |  |
|  | X3 | X2 | X1 | X0 |  | Y3 | Y2 | Y1 | Y0 |
| 0 1 2 3 | 0 0 0 0 | 0 0 0 0 | 0 0 1 1 | 0 1 0 1 | E  F  0  9 | 1  1  1  0 | 1  0  0  1 | 1  0  0  0 | 0  1  1  0 |
| 4 5 6 7 | 0 0 0 0 | 1 1 1 1 | 0 0 1 1 | 0 1 0 1 | A  4  1  8 | 1  1  0  0 | 1  1  0  0 | 0  0  1  1 | 1  0  1  0 |
| 8 9 A B | 1 1 1 1 | 0 0 0 0 | 0 0 1 1 | 0 1 0 1 | C  5  2  7 | 1  1  0  0 | 0  0  0  0 | 1  1  1  0 | 1  1  1  1 |
| C D E F | 1 1 1 1 | 1 1 1 1 | 0 0 1 1 | 0 1 0 1 | D  6  3  7 | 0  0  1  1 | 0  0  0  0 | 1  1  1  1 | 0  0  1  0 |

Табл.2 Таблица истинности.

**1. Карты Карно в виде МДНФ, схем. редактор. (1 Метод)**

По таблице истинности (Табл. 2) строим карты Карно с последующей минимизацией формул МДНФ.

**1.1. Карты Карно и формулы МДНФ, построенные по ним.**

**1) Y0 ДНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0 X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 0 |
| 01 | 1 | 0 | 0 | 1 |
| 11 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 0 |

Табл. 1.1 Карта Карно Y0 ДНФ.

Y0МДНФ = (X2&~X0) | (~X3& ~X2& X0) | (~X3& ~X1& ~X0) | (X3& ~X2& ~X1&X0)

**3) Y2 ДНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0 X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 1 |

Табл. 1.2 Карта Карно Y2 ДНФ.

Y2МДНФ = (~X3 & ~X1) | (~X3 & X1) | (X1&~X0) | (X2& X1)

**2) Y1 ДНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0 X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 0 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 1 | 0 |

Табл.1.3 Карта Карно Y1 ДНФ.

Y1МДНФ = (~X3&~ X1) | (X3& X1&X0) | (X3& X2& X1) | (~X3& ~X2 &~X0)

**4) Y3 ДНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0 X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 |

Табл. 1.4 Карта Карно Y3 ДНФ.

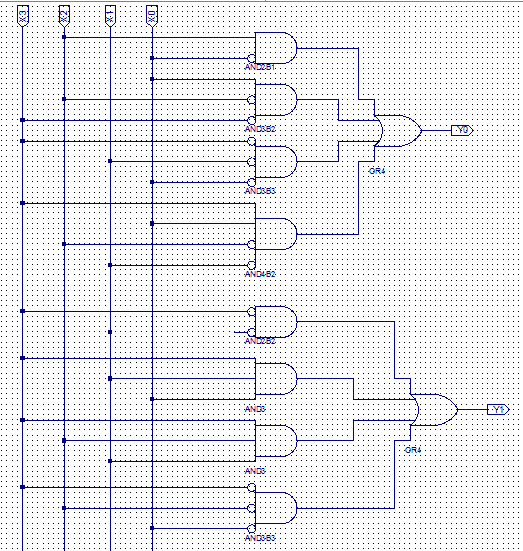
Y3МДНФ = (X1&X0) | (X3 &~ X1&~X0) | (X3& X2& ~X0)

По данным картам Карно были получены четыре формулы ДНФ для Y3, Y2, Y1 и Y0 соответственно.

Далее по ДНФ строим схему в схемотехническом редакторе Schematic editor САПР Xilinx ISE Design Suite.

**1.2 Схема ДНФ из Schematic editor.**

*(Схема из файла DNF.sch)*



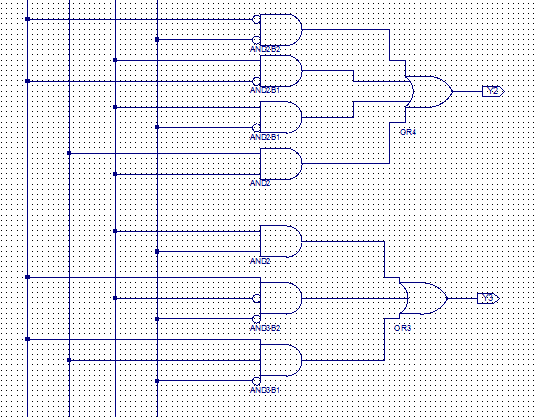


Рис. 1 Схема ДНФ из Schematic editor.

**2. Карты Карно в виде МКНФ, Verilog. (2 Метод)**

По таблице истинности (Табл. 2) строим карты Карно с последующей минимизацией формул МКНФ.

**2.1 Карты Карно и формулы МКНФ, построенные по ним.**

**1) Y0 КНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0  X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 1 | 0 |
| 01 | 1 | 0 | 0 | 1 |
| 11 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 0 |

Табл. 2.1 Карта Карно Y3 КНФ.

Y0МКНФ = (~X[2] | ~X[0]) & (X[2] | ~X[1] | X[0]) & (~X[3] | X[2] | X[0]) & (~X[3] | ~X[1] | ~X[0])

**3) Y2 КНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0  X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 0 | 1 | 1 |
| 01 | 1 | 0 | 1 | 1 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 0 | 1 |

Табл. 2.2 Карта Карно Y2 КНФ.

Y2МКНФ = (X[1] | ~X[0])&(~X[3] | X[1])&(~X[3] | X[2] | ~X[0])

**2) Y1 КНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0  X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 1 | 1 | 0 | 1 |
| 01 | 1 | 1 | 0 | 0 |
| 11 | 0 | 0 | 1 | 1 |
| 10 | 0 | 0 | 1 | 0 |

Табл. 2.3 Карта Карно Y1 КНФ.

Y1МКНФ = (~X[3] | X[1])&(X[3] | ~X[1] | ~X[0])&(X[3] | ~X[2] | ~X[1])&(~X[3] | X[2] | X[0]);

**4) Y3 КНФ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X1,X0  X3,X2 | 00 | 01 | 11 | 10 |
| 00 | 0 | 0 | 1 | 0 |
| 01 | 0 | 0 | 1 | 0 |
| 11 | 1 | 0 | 1 | 1 |
| 10 | 1 | 0 | 1 | 0 |

Табл. 2.4 Карта Карно Y0 КНФ.

Y3МКНФ = (X[3] | X[0])&(X[1] | ~X[0])&(X[2] | ~X[1] | X[0])

**2.2 Содержимое файла KNF.v**

`timescale 1ns / 1ps

module KNF(

input [3:0] X,

output [3:0] Y

);

assign Y[0] = (~X[2] | ~X[0]) & (X[2] | ~X[1] | X[0]) & (~X[3] | X[2] | X[0]) & (~X[3] | ~X[1] | ~X[0]);

assign Y[1] = (~X[3] | X[1])&(X[3] | ~X[1] | ~X[0])&(X[3] | ~X[2] | ~X[1])&(~X[3] | X[2] | X[0]);

assign Y[2] = (X[1] | ~X[0])&(~X[3] | X[1])&(~X[3] | X[2] | ~X[0]);

assign Y[3] = (X[3] | X[0])&(X[1] | ~X[0])&(X[2] | ~X[1] | X[0]);

endmodule

**3. Поведенческое описание. (3 Метод)**

В данном методе необходимости в минимизации нет, поэтому описываем все на языке Verilog.

**3.1 Содержимое файла BEH.v**

`timescale 1ns / 1ps

module BEH(

input [3:0] X,

output reg [3:0] Y

);

always @ (X)

begin

case(X)

4'h0: Y <= 4'h7;

4'h1: Y <= 4'h3;

4'h2: Y <= 4'h6;

4'h3: Y <= 4'hD;

4'h4: Y <= 4'h7;

4'h5: Y <= 4'h2;

4'h6: Y <= 4'h5;

4'h7: Y <= 4'hC;

4'h8: Y <= 4'h8;

4'h9: Y <= 4'h1;

4'hA: Y <= 4'h4;

4'hB: Y <= 4'hA;

4'hC: Y <= 4'h9;

4'hD: Y <= 4'h0;

4'hE: Y <= 4'hF;

4'hF: Y <= 4'hE;

endcase

end

endmodule

**4.1 Verilog модель модуля верхнего уровня иерархии.**

В проекте, особенно сложном, бывает много модулей, соединенных между собой. Прежде всего, нужно заметить, что в проекте всегда есть один модуль самого верхнего уровня (В нашем случае это - LR1\_TOP.v). Он может состоять из нескольких других модулей. Те в свою очередь могут содержать еще модули и так далее. Не обязательно, чтобы все модули были написаны на одном языке описания аппаратуры. Совсем наоборот. Все зависит от поставленной задачи и простоты ее реализации, а иногда и от личных предпочтений. Наш модуль верхнего уровня написан на языке Verilog, хотя это может быть любой другой метод описания модулей, например, схема.

**4.1 Top module, содержимое файла TOP.v**

`timescale 1ns / 1ps

module TOP(

input [11:0] SW,

output [11:0] LED

);

DNF CL1(

.X3(SW[3]),

.X2(SW[2]),

.X1(SW[1]),

.X0(SW[0]),

.Y3(LED[3]),

.Y2(LED[2]),

.Y1(LED[1]),

.Y0(LED[0]));

KNF CL2(.X(SW[7:4]), .Y(LED[7:4]));

BEH CL3(.X(SW[11:8]), .Y(LED[11:8]));

endmodule

**5. Результаты логической верификации.**

Для проверки работоспособности синтезированной модели схемы необходимо описать на языке Verilog тестовый модуль, который будет последовательно подавать на вход числа от 0 до F.

**5.1** **Тестовый модуль, содержимое файла Test\_top.v**

`timescale 1ns / 1ps

module Test\_top;

reg [11:0] SW;

wire [11:0] LED;

TOP uut (

.SW(SW),

.LED(LED)

);

initial begin

SW = 12'h000; #100;

SW = 12'h111; #100;

SW = 12'h222; #100;

SW = 12'h333; #100;

SW = 12'h444; #100;

SW = 12'h555; #100;

SW = 12'h666; #100;

SW = 12'h777; #100;

SW = 12'h888; #100;

SW = 12'h999; #100;

SW = 12'hAAA; #100;

SW = 12'hBBB; #100;

SW = 12'hCCC; #100;

SW = 12'hDDD; #100;

SW = 12'hEEE; #100;

SW = 12'hFFF; #100;

end

endmodule

Теперь можно запустить симуляцию данного модуля и проверить его работоспособность в симуляторе **ISim**.

**5.2 Временная диаграмма из симулятора ISim.**

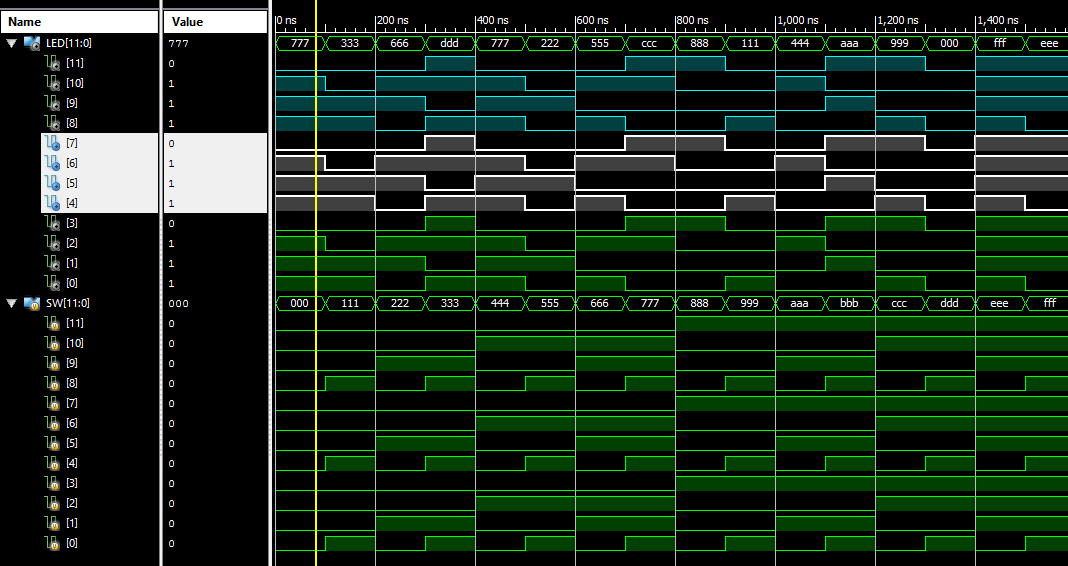


Рис. 2. Временная диаграмма ISim

**Заключение.**

В данной работе тремя различными способами были синтезированы модели комбинационной схемы 4х4:  
1. На вентильном уровне, методом карт Карно в виде МДНФ, в схемотехническом редакторе Schematic editor САПР Xilinx ISE Design Suite.  
2. На вентильном уровне, методом карт Карно в виде МКНФ, на языке описания аппаратуры Verilog.  
3. На поведенческом уровне, на языке описания аппаратуры Verilog.

На языке Verilog было реализовано тестовое окружение, а также проведена верификация спроектированных моделей при помощи симулятора iSim, который входит в состав САПР Xilinx ISE Design Suite.

В ходе симуляции ошибок выявлено не было, комбинационные схемы выполняют свои задачи исправно.

Проделав работу, самым удобным является метод поведенческого описания(простота и скорость реализации). Схемотехнический метод является самым трудоемким, но дает наглядно представление.