



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА - Российский технологический университет»

РТУ МИРЭА

Институт Информационных Технологий
Кафедра Вычислительной Техники (ВТ)

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2

«Проектирование комбинационных логических схем в САПР Quartus II 9.0»

по дисциплине

«Проектирование и разработка систем на базе программируемых логических интегральных
схем»

Выполнил студент группы
ИВБО-05-22

Нгуен Д.Х.

Принял ассистент кафедры ВТ

Кряхтунов Г. М.

Практическая работа выполнена

«__»_____2025 г.

«Зачтено»

«__»_____2025 г.

Москва 2025

АННОТАЦИЯ

Данная работа включает в себя 32 рисунка, 6 листингов. Количество страниц в работе — 27.

СОДЕРЖАНИЕ

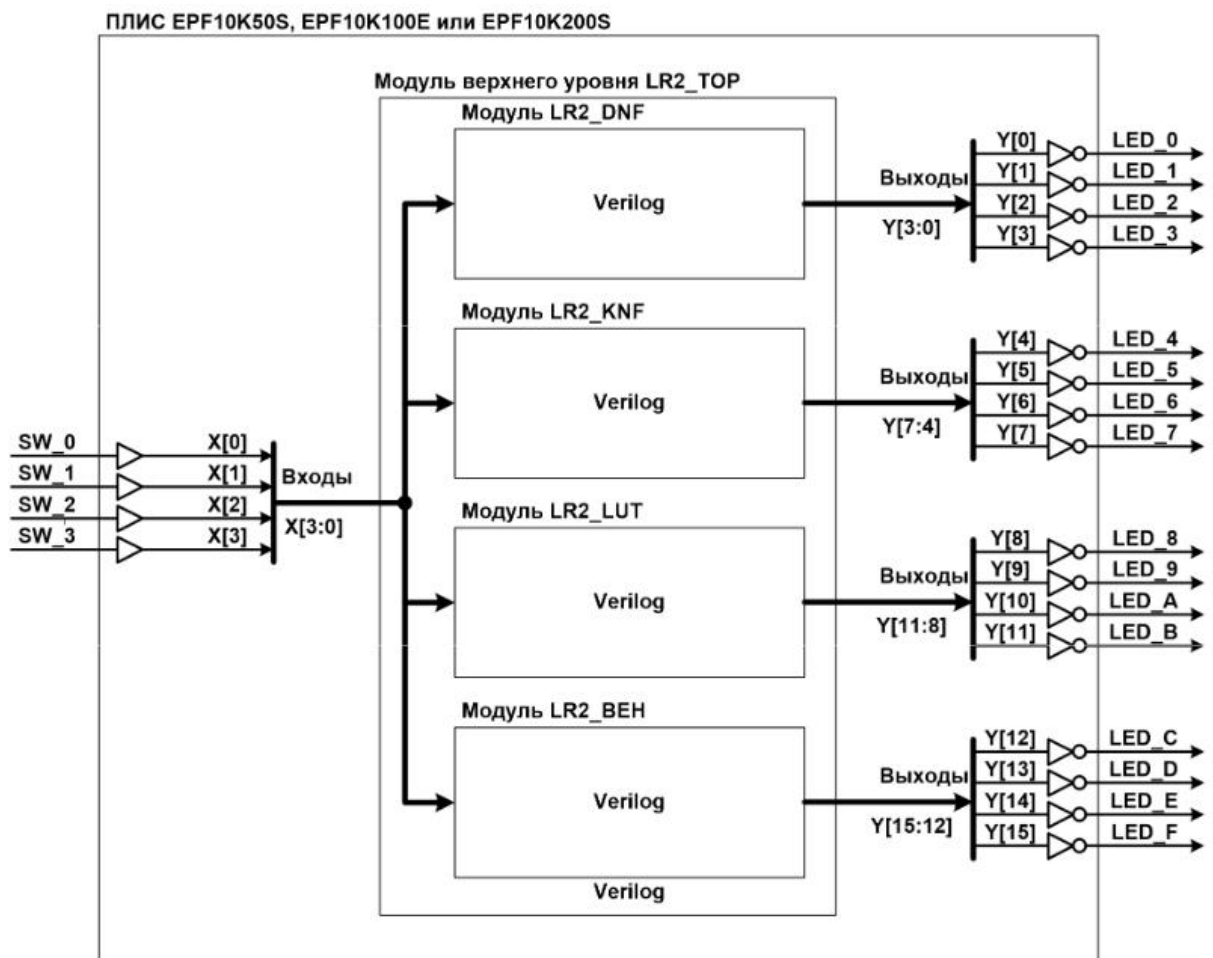
1 ОСНОВНОЙ РАЗДЕЛ	6
1.1 Таблица истинности МДНФ/МКНФ	6
1.2 Модуль верхнего уровня	11
1.3 Модуль МДНФ	11
1.4 Модуль МКНФ	11
1.5 Модуль LUT	12
1.6 Модуль ВЕН	12
1.7 Модуль верхнего уровня	12
1.8 Тестирование на плате	12
1.9 Количество занимаемых ресурсов ПЛИС	13
1.10 Синтезированная схема RTL-уровня	13
1.11 Синтезированная схема (нетлиста) технологических примитивов ПЛИС .	17
ПРИЛОЖЕНИЯ	26
Приложение А	27

ВВЕДЕНИЕ

В лабораторной работе требуется спроектировать комбинационную логическую схему, реализованную в объёме одной из следующих ПЛИС: EPF10K50S, EPF10K100E или EPF10K200S. Комбинационная схема реализуется в объёме ПЛИС четырьмя различными способами:

1. модель на языке Verilog на основе минимальной ДНФ;
2. модель на языке Verilog на основе минимальной КНФ;
3. модель на языке Verilog на основе элементов LUT;
4. поведенческая модель на языке Verilog.

Реализуемое в ПЛИС устройство имеет структурную схему,



представленную ниже (Рисунок 1.1).

Рисунок 1.1 – Структурная схема проектируемого устройства

Персональный вариант: A8D9E1D702288000.

1 ОСНОВНОЙ РАЗДЕЛ

1.1 Таблица истинности МДНФ/МКНФ

Построим таблицу истинности в двоичной системе счисления (Таблица 1.1).

Таблица 1.1 – Таблица истинности в двоичной системе счисления

Входы					Выходы					INIT			
Hex	X3	X2	X1	X0	Hex	Y3	Y2	Y1	Y0	3	2	1	0
0	0	0	0	0	D	1	0	1	1	5	C	2	9
1	0	0	0	1	3	1	0	1	1				
2	0	0	1	0	2	0	1	1	1				
3	0	0	1	1	C	1	1	1	1				
4	0	1	0	0	6	1	1	1	0	6	D	D	9
5	0	1	0	1	9	1	1	1	1				
6	0	1	1	0	4	1	1	0	0				
7	0	1	1	1	4	1	0	0	1				
8	1	0	0	0	7	1	0	1	0	E	B	1	6
9	1	0	0	1	2	0	1	1	1				
A	1	0	1	0	D	1	1	0	1				
B	1	0	1	1	E	1	0	0	0				
C	1	1	0	0	8	0	0	0	0	C	5	2	3
D	1	1	0	1	6	0	0	0	1				
E	1	1	1	0	F	1	0	0	0				
F	1	1	1	1	3	0	1	0	0				

По таблице истинности в отчёте построим карты Карно и выведем формулы для МДНФ (Рисунки 1.1, 1.2, 1.3, 1.4) (Формулы 1.1, 1.2, 1.3, 1.4) и МКНФ (Рисунки 1.5, 1.6, 1.7, 1.8) (Формулы 1.5, 1.6, 1.7, 1.8) для каждого из четырёх выходов $Y[3:0]$.

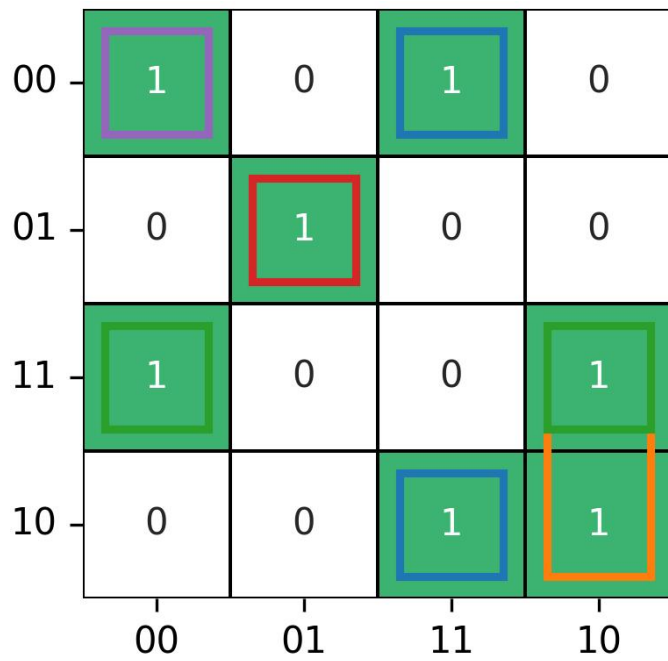


Рисунок 1.1 – Карта Карно для МДНФ Y_3

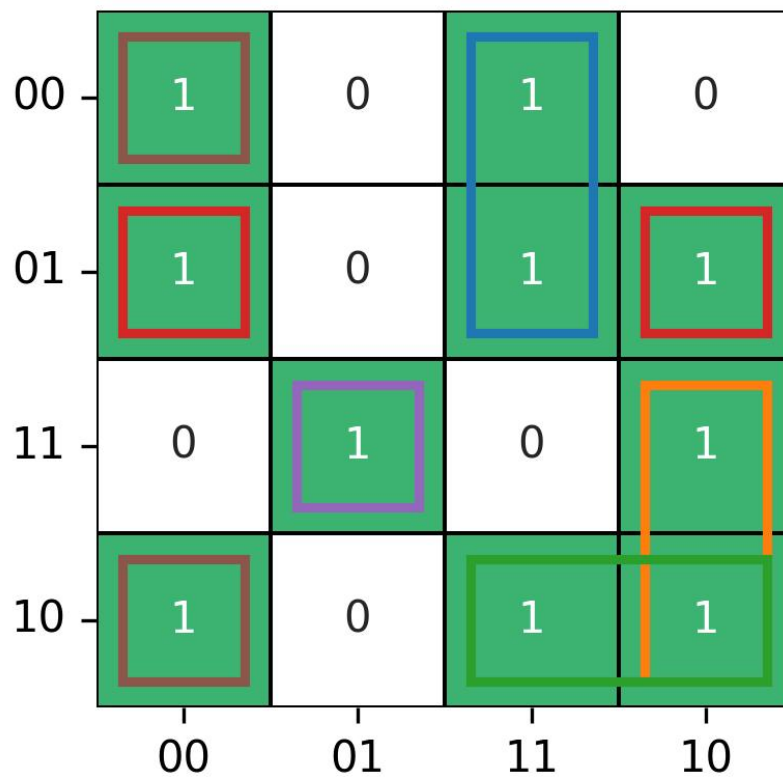


Рисунок 1.2 – Карта Карно для МДНФ Y_2

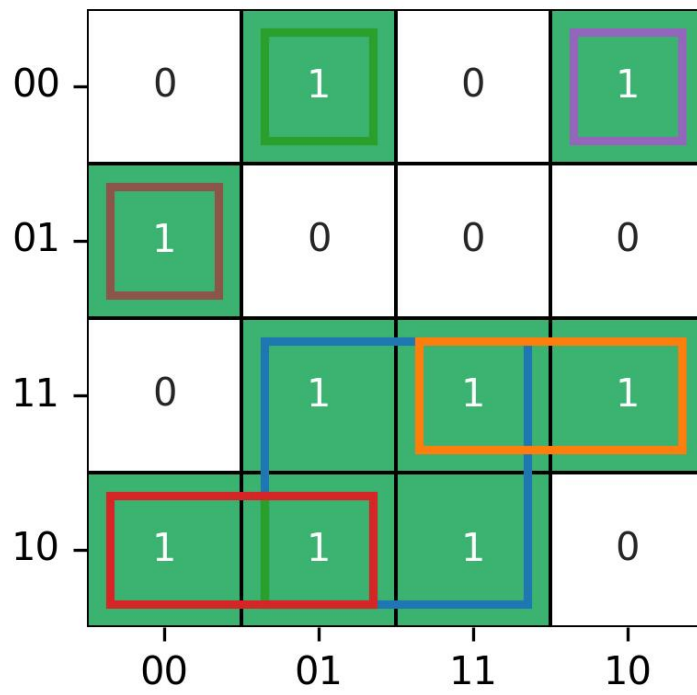


Рисунок 1.3 – Карта Карно для МДНФ Y1

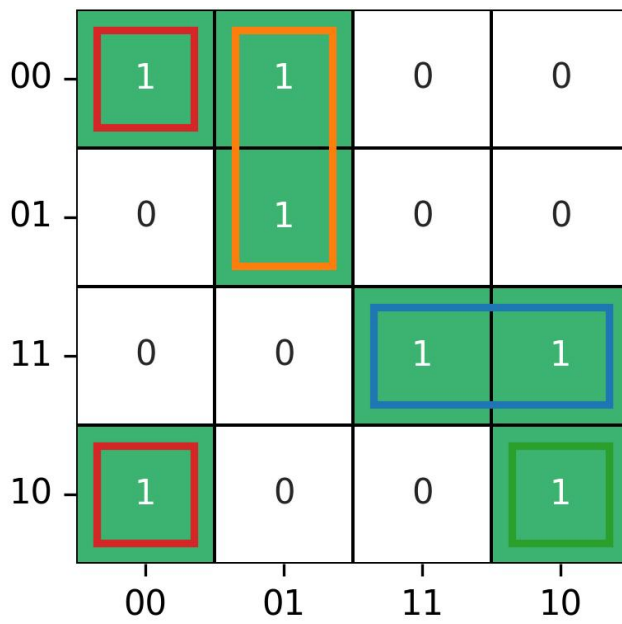


Рисунок 1.4 – Карта Карно для МДНФ Y0

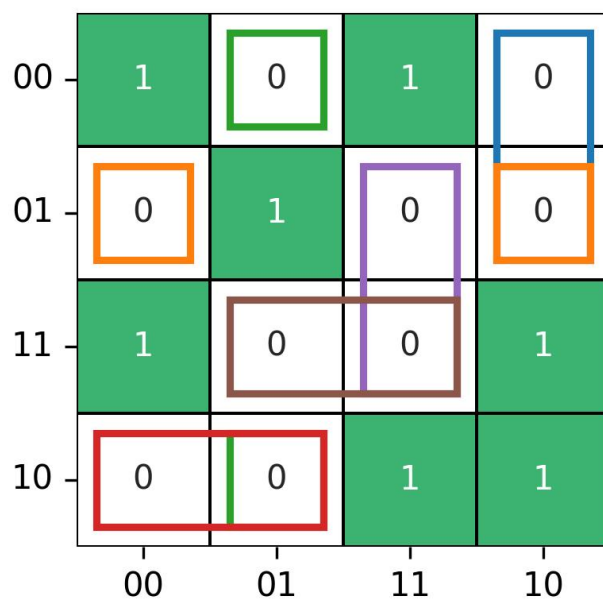


Рисунок 1.5 – Карта Карно для МКНФ Y3

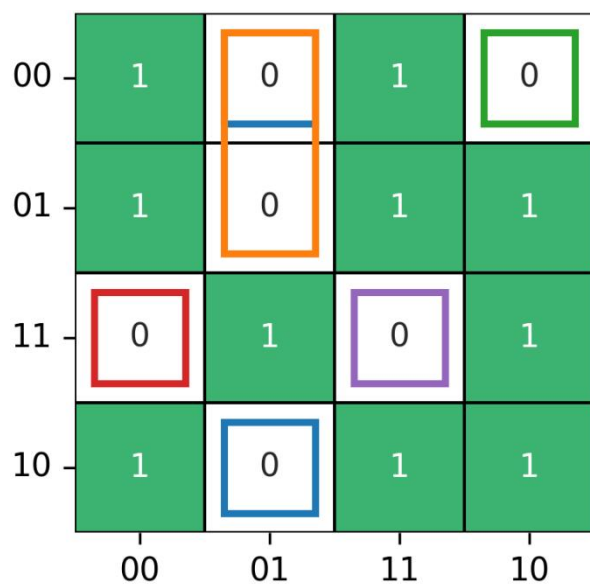


Рисунок 1.6 – Карта Карно для МКНФ Y2

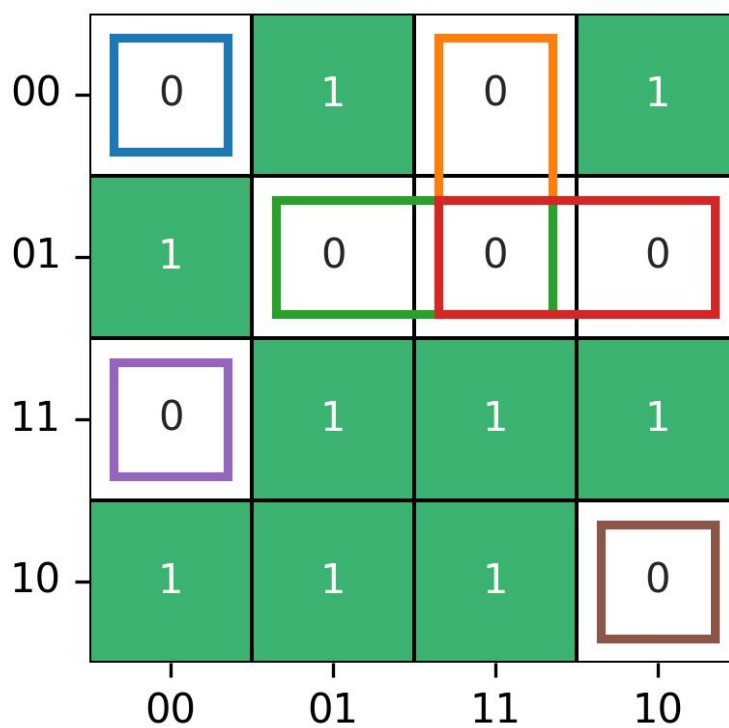


Рисунок 1.7 – Карта Карно для МКНФ Y1

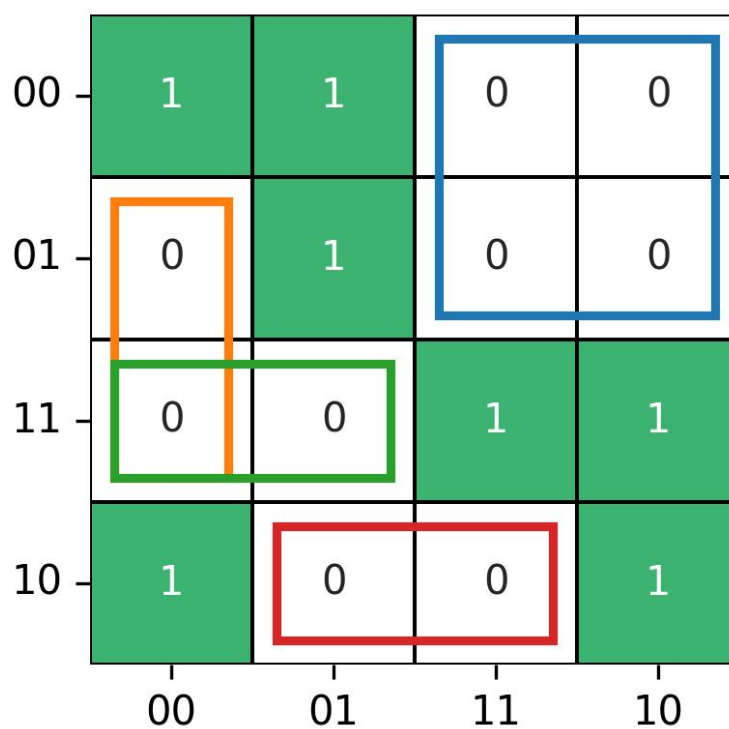


Рисунок 1.8 – Карта Карно для МКНФ Y0

$$Y_3 = (X_0 X_1 \overline{X_2}) + (X_1 X_3 \overline{X_0}) + (X_2 X_3 \overline{X_0}) + (X_0 X_2 \overline{X_1 X_3}) + (\overline{X_0 X_1 X_2 X_3}) \quad \#(1.1)$$

$$Y_2 = (X_0 X_1 \overline{X_3}) + (X_1 X_3 \overline{X_0}) + (X_1 X_3 \overline{X_2}) + (X_2 \overline{X_0 X_3}) + (X_0 X_2 X_3 \overline{X_1}) + (\overline{X_0 X_1 X_2}) \quad \#(1.2)$$

$$Y_1 = (X_0X_3) + (X_1X_2X_3) + (X_0\overline{X_1X_2}) + (X_3\overline{X_1X_2}) + \#(1.3)$$

$$(X_1\overline{X_0X_2X_3}) + (X_2\overline{X_0X_1X_3})$$

$$Y_0 = (X_1X_2X_3) + (X_0\overline{X_1X_3}) + (X_3\overline{X_0X_2}) + (\overline{X_0X_1X_2})\#(1.5)$$

$$Y_0 = (X_3 + \overline{X_1})(X_0 + X_1 + \overline{X_2})(X_1 + \overline{X_2} + \overline{X_3})(X_2 + \overline{X_0} + \overline{X_3}) \#(1.6)$$

$$Y_1 = (X_0 + X_1 + X_2 + X_3)(X_3 + \overline{X_0} + \overline{X_1})(X_3 + \overline{X_0} + \overline{X_2}) \#(1.7)$$

$$(X_3 + \overline{X_1} + \overline{X_2})(X_0 + X_1 + \overline{X_2} + \overline{X_3})(X_0 + X_2 + \overline{X_1} + \overline{X_3})$$

$$Y_2 = (X_1 + X_2 + \overline{X_0})(X_1 + X_3 + \overline{X_0})(X_0 + X_2 + X_3 + \overline{X_1}) \#(1.8)$$

$$(X_0 + X_1 + \overline{X_2} + \overline{X_3})(\overline{X_0} + \overline{X_1} + \overline{X_2} + \overline{X_3})$$

$$Y_3 = (X_0 + X_3 + \overline{X_1})(X_0 + X_3 + \overline{X_2})(X_1 + X_2 + \overline{X_0}) \#(1.9)$$

$$(X_1 + X_2 + \overline{X_3})(\overline{X_0} + \overline{X_1} + \overline{X_2})(\overline{X_0} + \overline{X_2} + \overline{X_3})$$

1.2 Модуль верхнего уровня

Создадим модуль верхнего уровня иерархии проекта LR2_TOP на языке Verilog (Приложение А).

1.3 Модуль МДНФ

Создадим модуль на языке Verilog LR2_DNF и опишем в нём схему согласно формулам МДНФ с помощью операторов assign (Приложение А).

1.4 Модуль МКНФ

Создадим модуль на языке Verilog LR2_KNF и опишем в нём схему согласно формулам МКНФ с помощью операторов assign (Приложение А).

1.5 Модуль LUT

Создадим модуль на языке Verilog LR2_LUT и опишем в нём схему с помощью модулей KRG_LUT, заполнить параметры INIT модулей KRG_LUT согласно таблице истинности (Приложение А).

1.6 Модуль ВЕН

Создадим модуль на языке Verilog модуль LR2_ВЕН и описать в нём на поведенческом уровне абстракции комбинационную схему согласно таблице истинности (Приложение А).

1.7 Модуль верхнего уровня

Опишем схему, объединяющую модули LR2_DNF, LR2_KNF, LR2_LUT и LR2_ВЕН в объёме модуля верхнего уровня иерархии проекта LR2_TOP (Приложение А).

1.8 Тестирование на плате

Протестируем построенную комбинационную схему на плате (Рисунок 1.10).

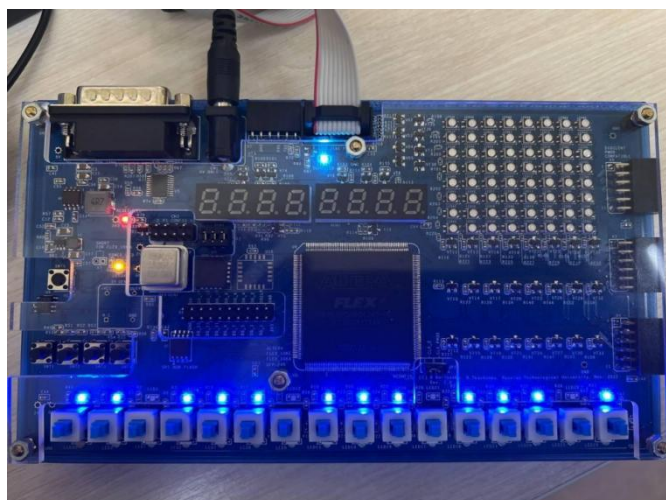


Рисунок 1.10 – Тестирование на плате

1.9 Количество занимаемых ресурсов ПЛИС

В интерактивном отчёте откроем вкладку «Fitter => Resource Section => Resources Usage Summary» (Рисунок 1.11).

Fitter Resource Usage Summary			
	Resource	Usage	
1	Total logic elements	16 / 2,880 (< 1 %)	
2	Registers	0 / 2,880 (0 %)	
3	Logic elements in carry chains	0	
4	User inserted logic elements	0	
5	<input type="checkbox"/> I/O pins	20 / 189 (11 %)	
6	-- Clock pins	0 / 2 (0 %)	
7	-- Dedicated input pins	0 / 4 (0 %)	
8	Global signals	0	
9	EABs	0 / 10 (0 %)	
10	Total memory bits	0 / 40,960 (0 %)	
11	Total RAM block bits	0 / 40,960 (0 %)	
12	Maximum fan-out node	SW_2	
13	Maximum fan-out	16	
14	Highest non-global fan-out signal	SW_2	
15	Highest non-global fan-out	16	
16	Total fan-out	80	
17	Average fan-out	2.22	

Рисунок 1.11 – Количество занимаемых ресурсов ПЛИС

1.10 Синтезированная схема RTL-уровня

Добавим в отчёт по лабораторной работе примеры синтезированной схемы RTL-уровня (Рисунки 1.12, 1.13, 1.14, 1.15, 1.16).

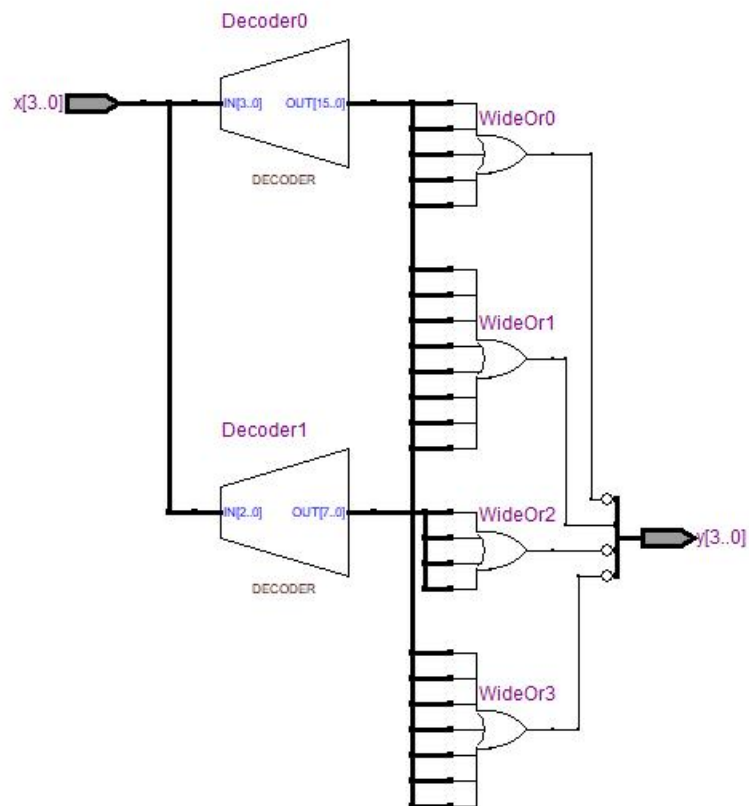
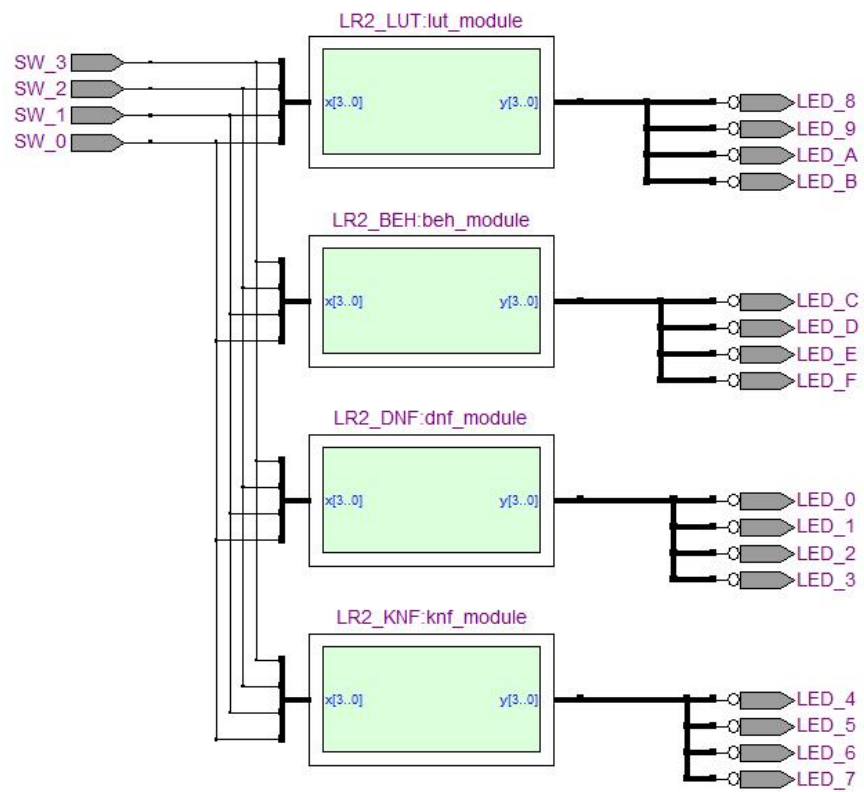


Рисунок 1.12 – RTL-схема верхнего модуля

Рисунок 1.13 – RTL-схема модуля BEH

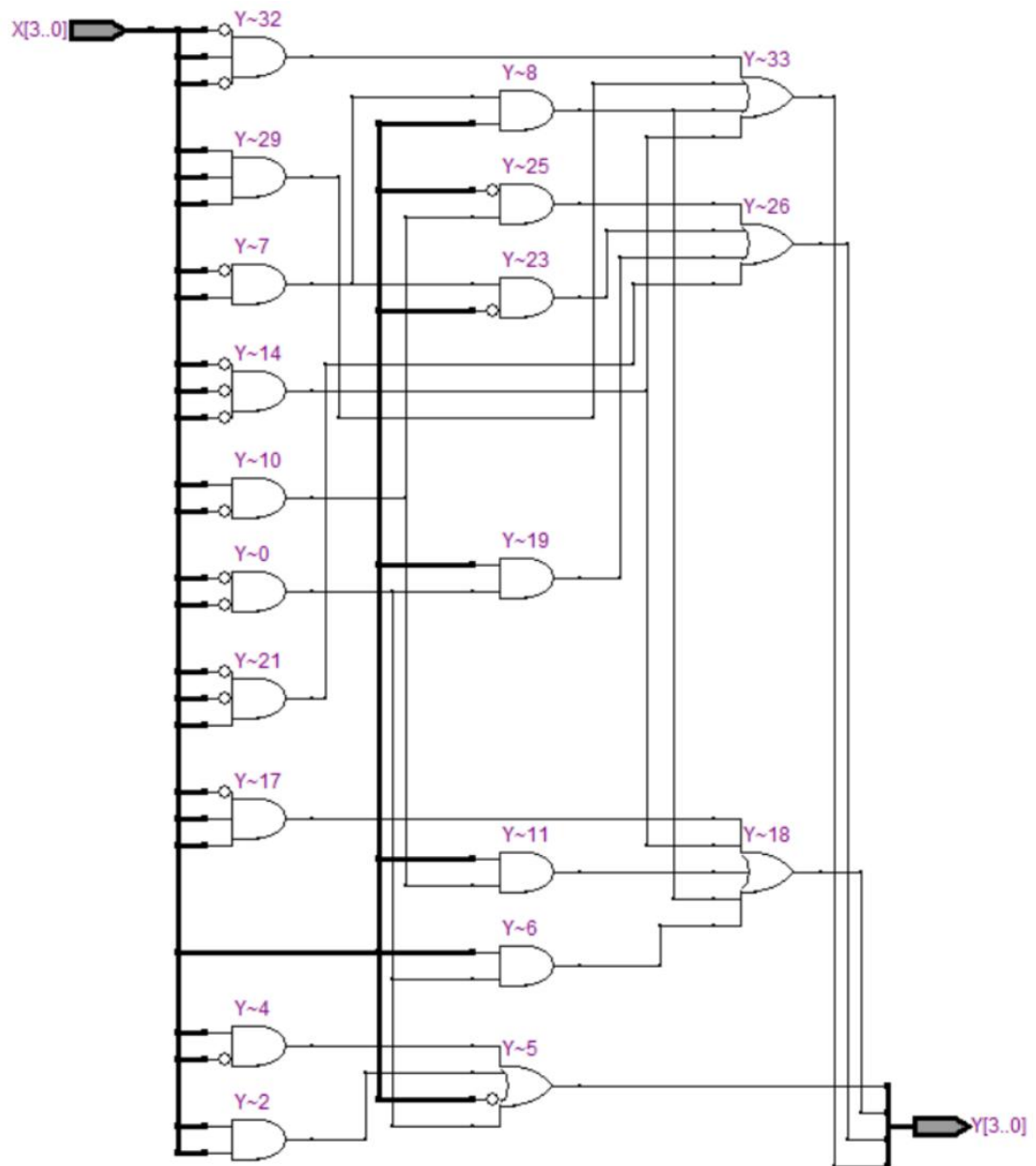


Рисунок 1.14 – RTL-схема модуля DNF

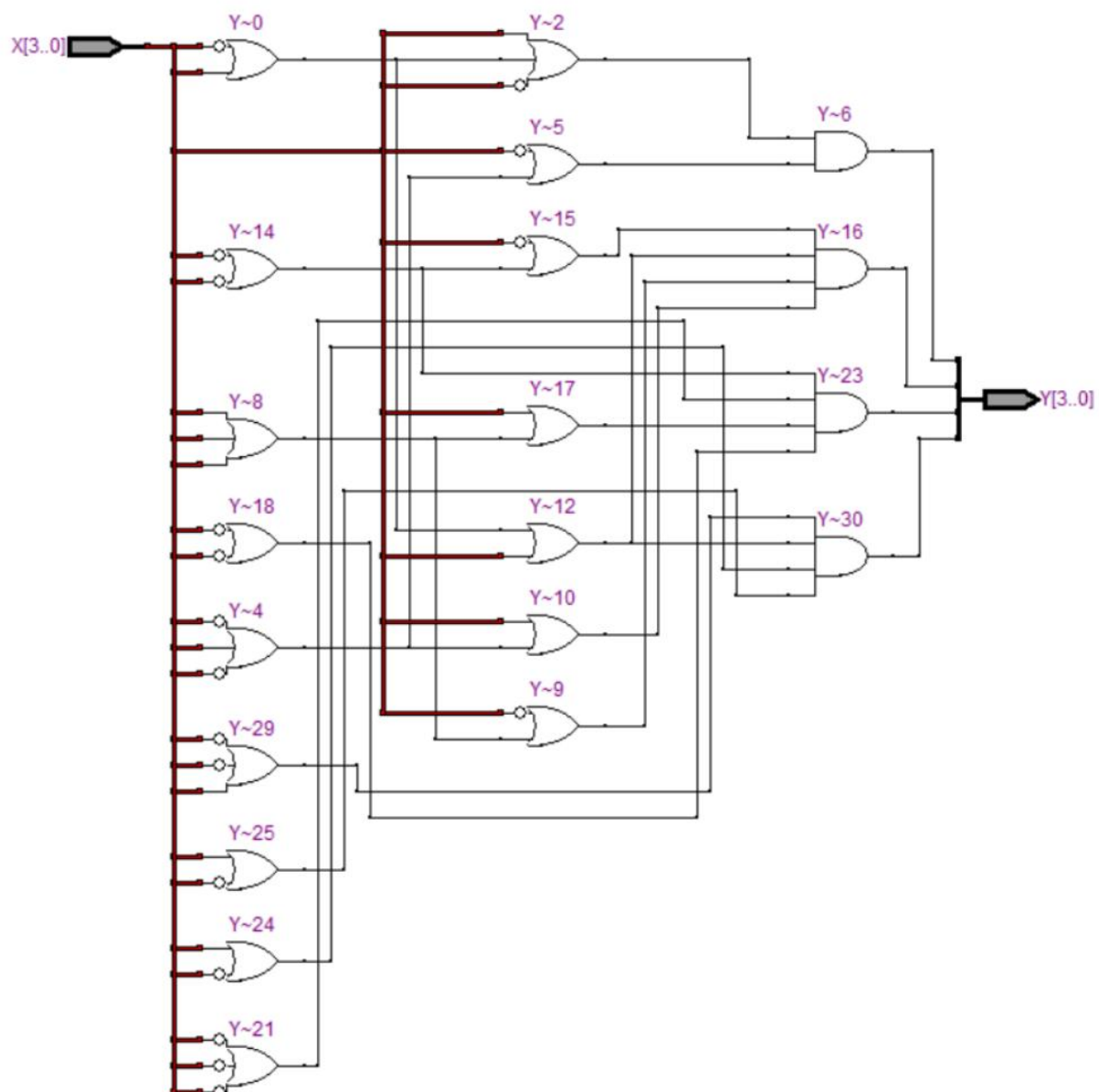


Рисунок 1.15 – RTL-схема модуля KNF

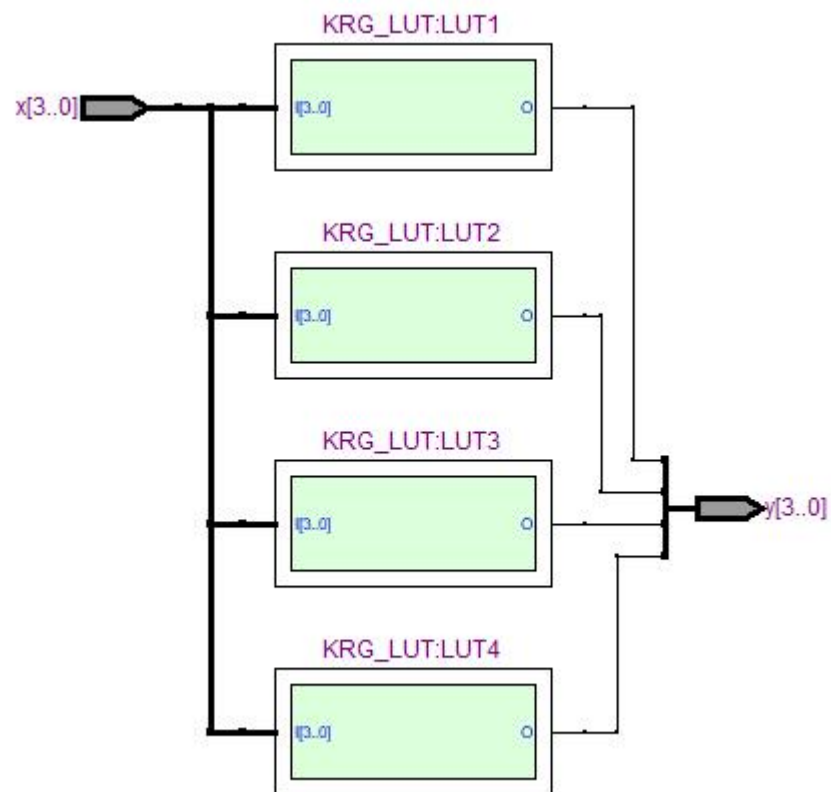


Рисунок 1.16 – RTL-схема модуля LUT

1.11 Синтезированная схема (нетлиста) технологических примитивов ПЛИС

Для просмотра синтезированной схемы (нетлиста) уровня технологических примитивов выбранного типа ПЛИС выполнить пункт «Technology Map Viewer» (Рисунки 1.17, 1.18, 1.19, 1.20).

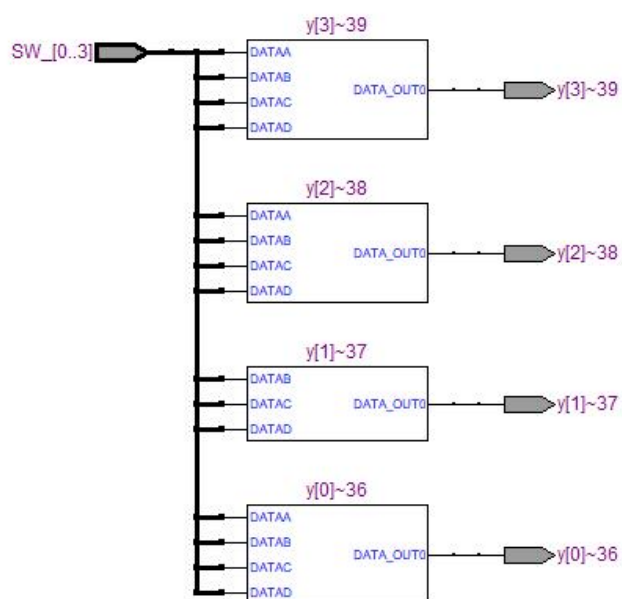
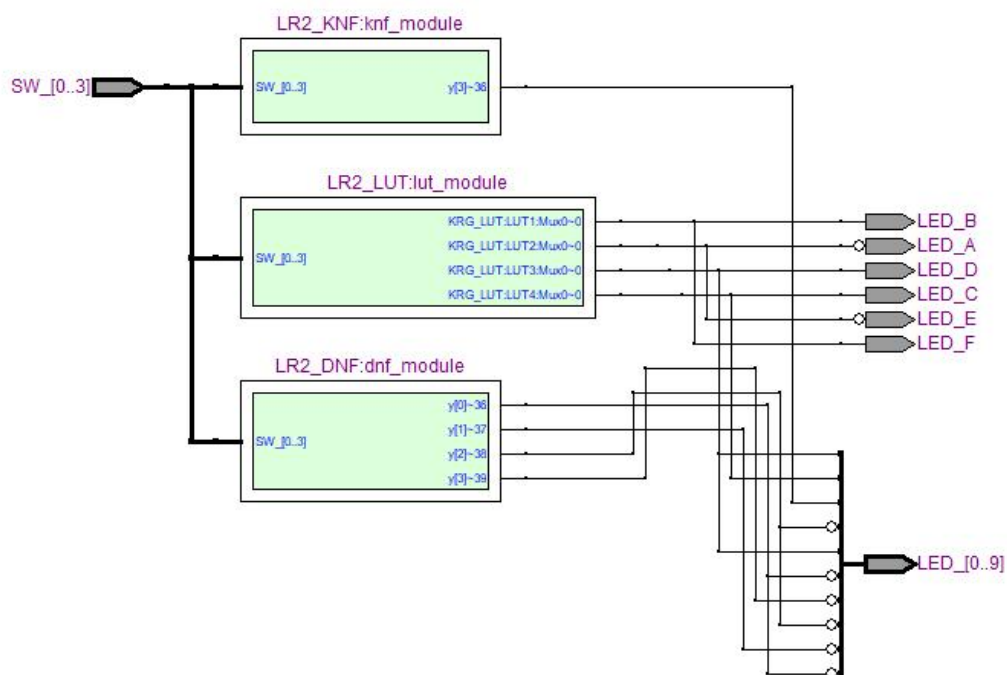
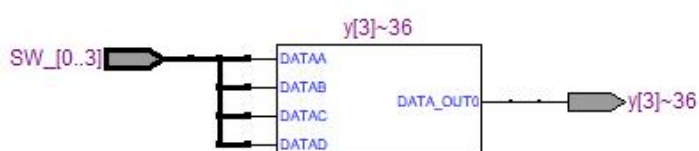


Рисунок 1.20 – Синтезированная схема технологических примитивов ПЛИС

Рисунок 1.21 – Синтезированная схема технологических примитивов ПЛИС

Рисунок 1.22 – Синтезированная схема технологических примитивов ПЛИС



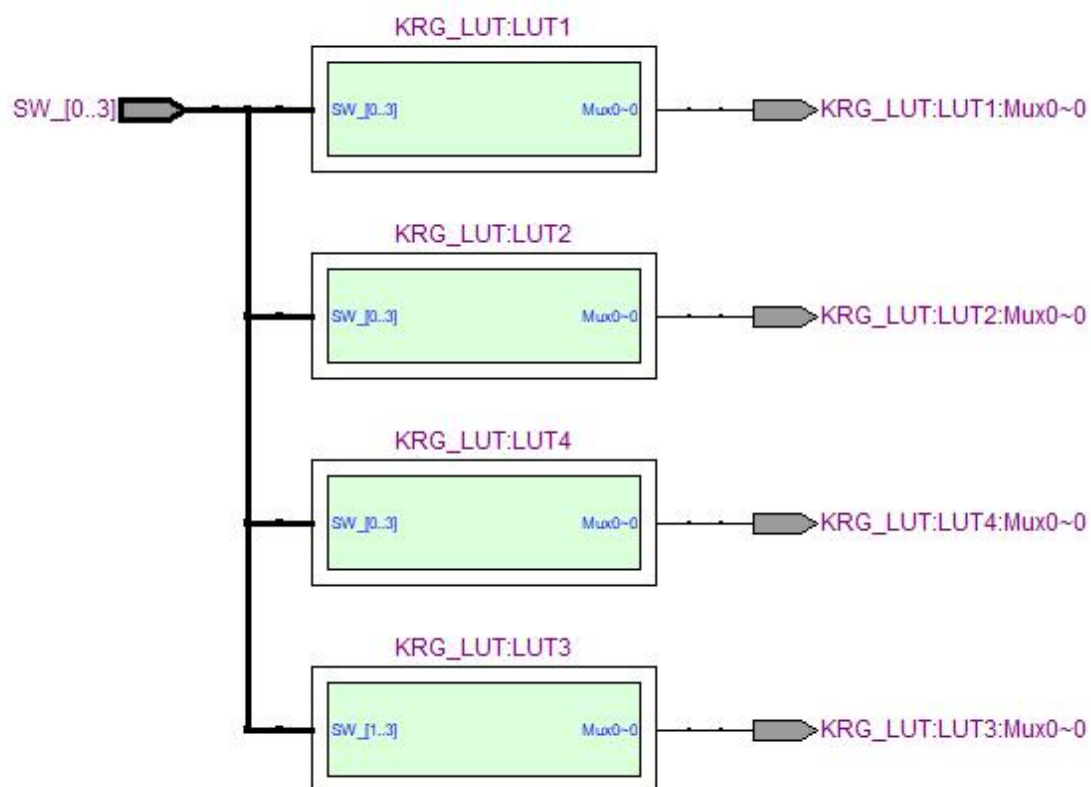


Рисунок 1.23 – Синтезированная схема технологических примитивов ПЛИС

Далее добавить конструкцию «`/*synthesis keep*/`» после объявления выходов «Y». Аналогичным способом проверим и добавим в отчёт по лабораторной работе схемы RTL-моделей всех модулей проекта независимо от уровня иерархии (Рисунки 1.24, 1.25, 1.26, 1.27, 1.28).

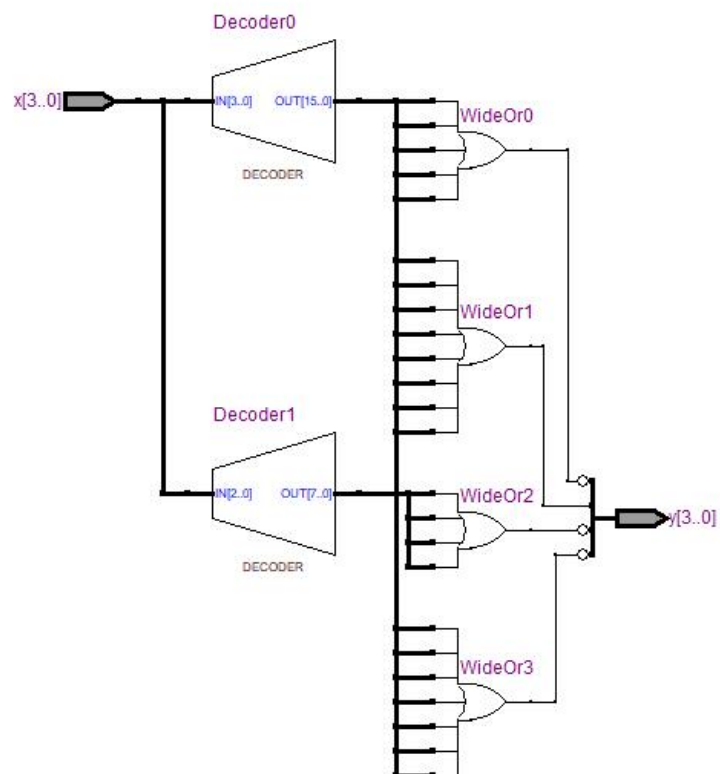
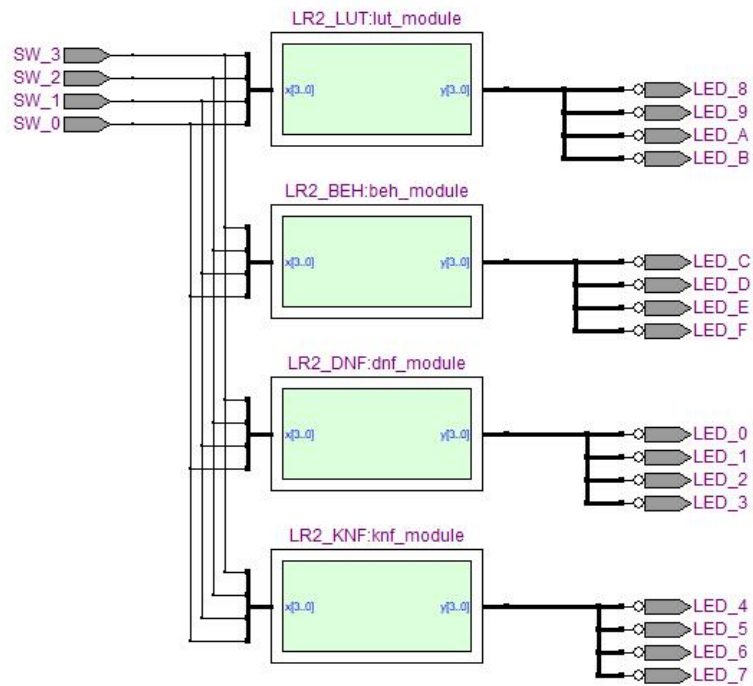


Рисунок 1.24 – RTL-схема модуля LUT

Рисунок 1.25 – RTL-схема модуля BEH

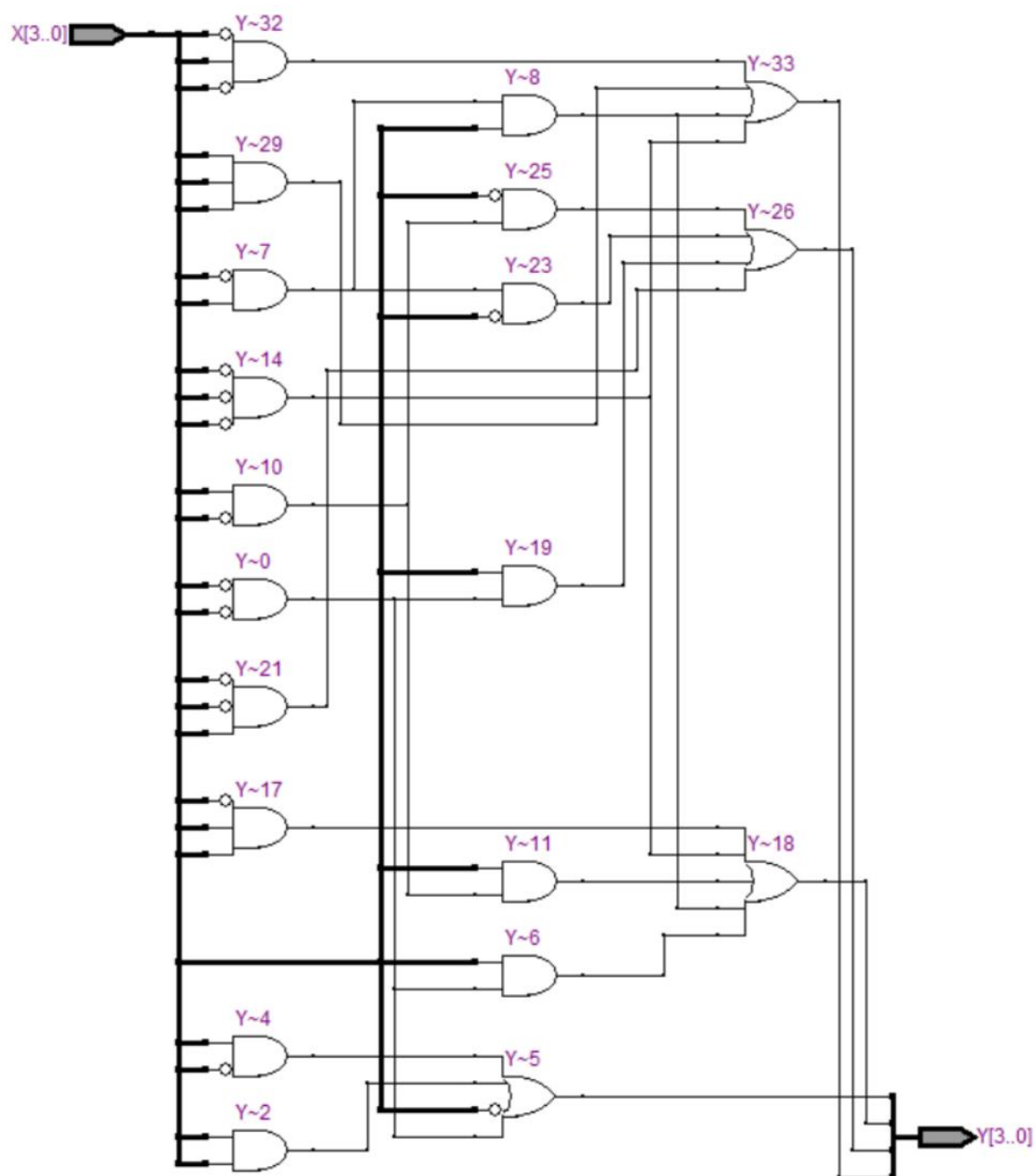


Рисунок 1.26 – RTL-схема модуля DNF

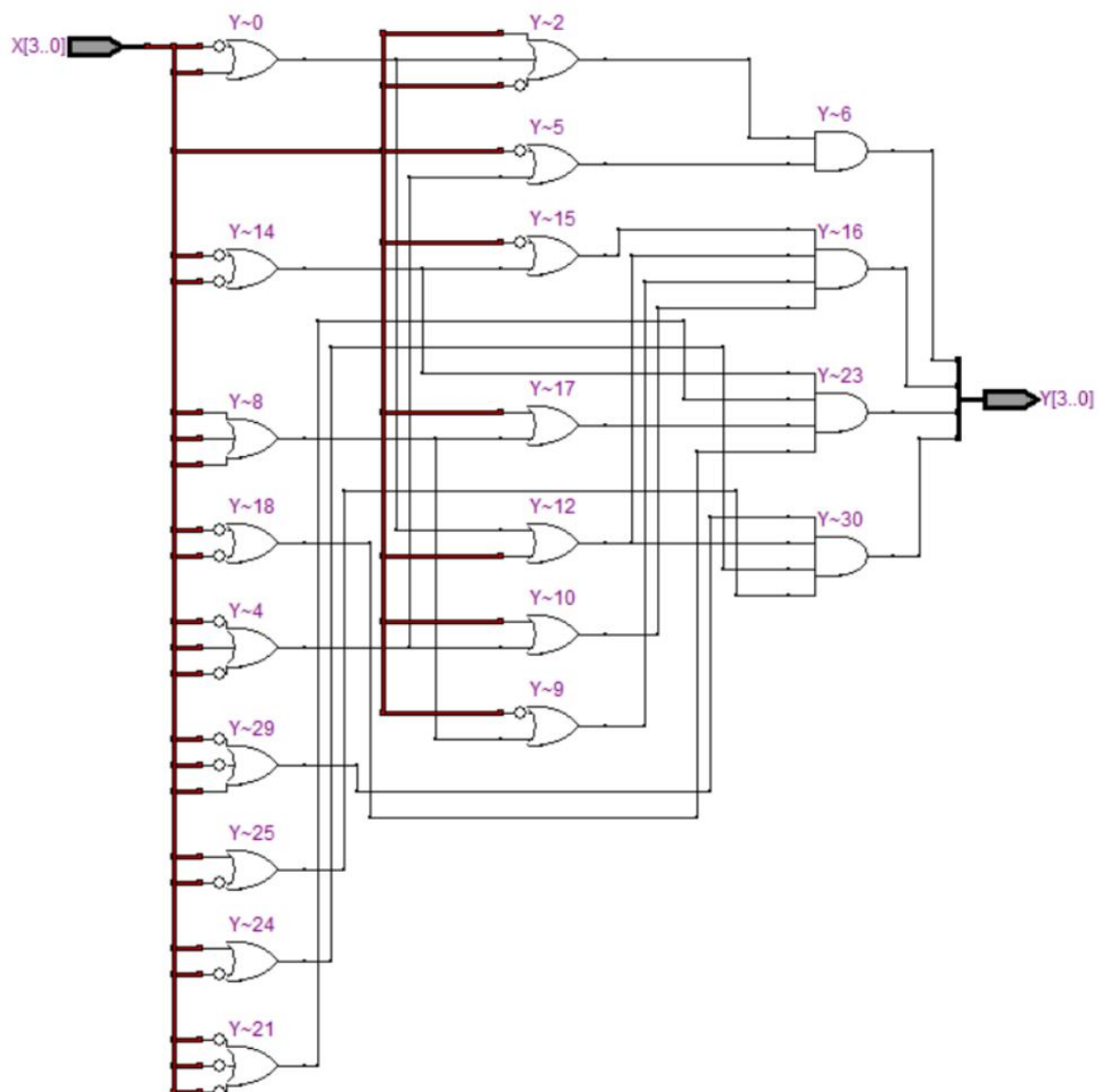


Рисунок 1.27 – RTL-схема модуля KNF

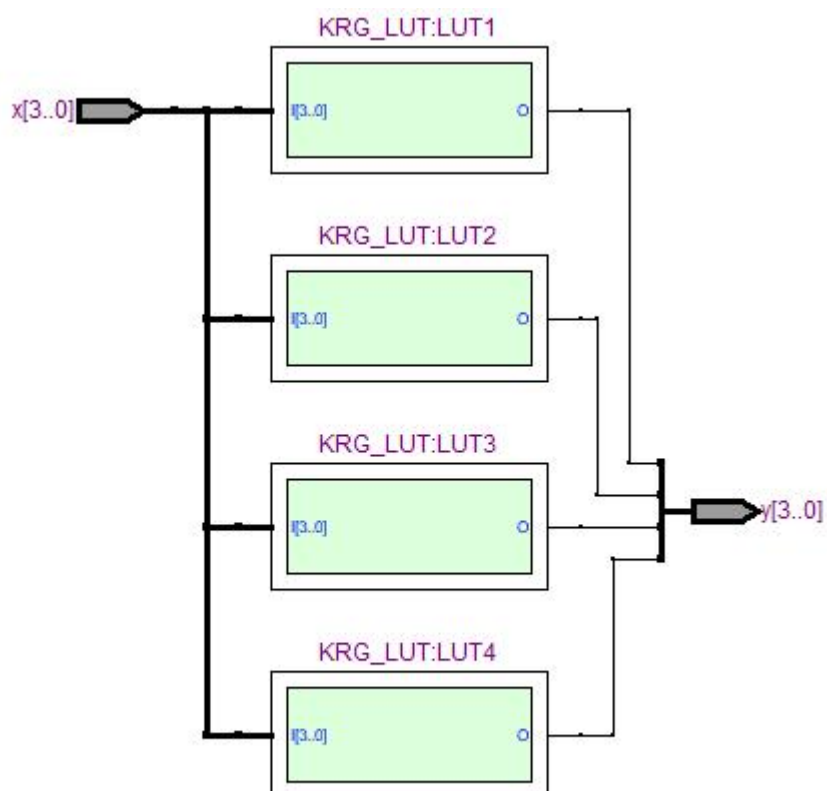
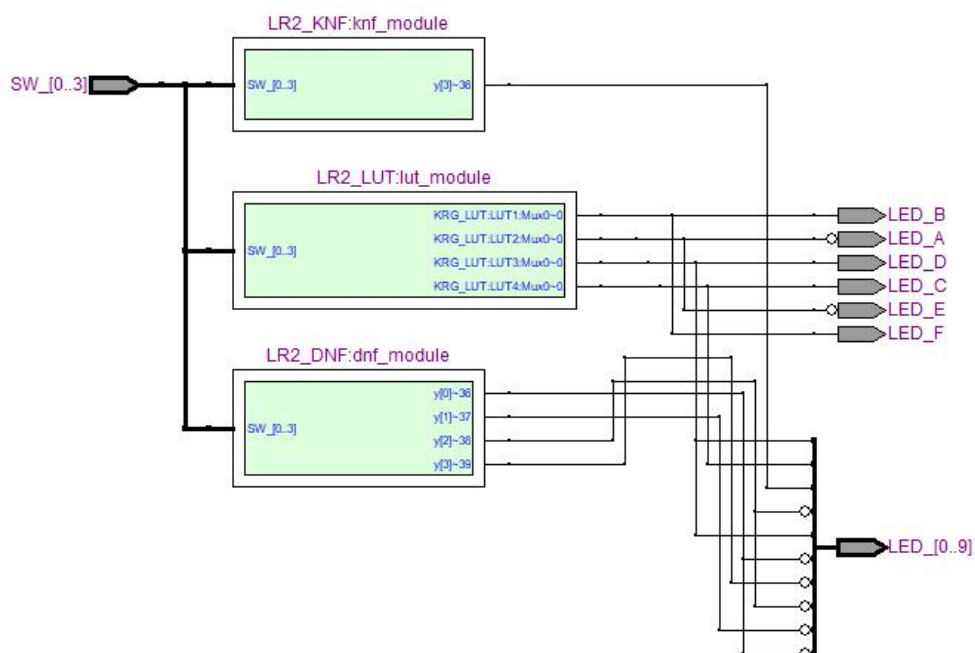


Рисунок 1.28 – RTL-схема модуля LUT

Аналогично добавить в отчёт по лабораторной работе схемы всех уровней



иерархии технологического нетлиста (Рисунок 1.29, 1.30, 1.31, 1.32).

Рисунок 1.29 – Синтезированная схема технологических примитивов ПЛИС

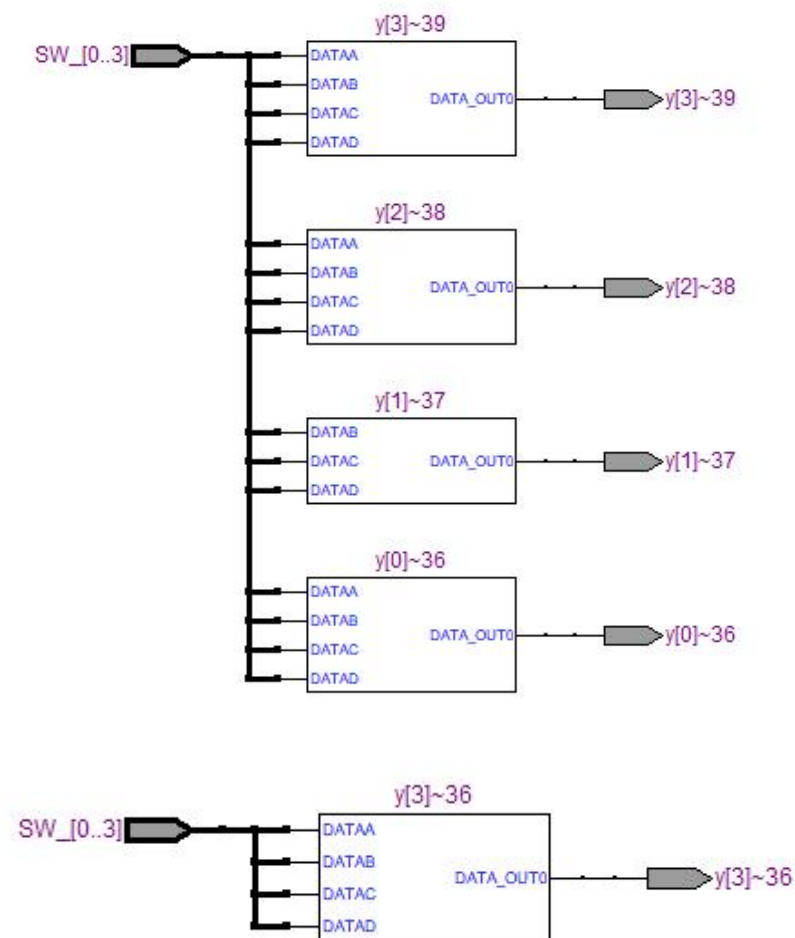
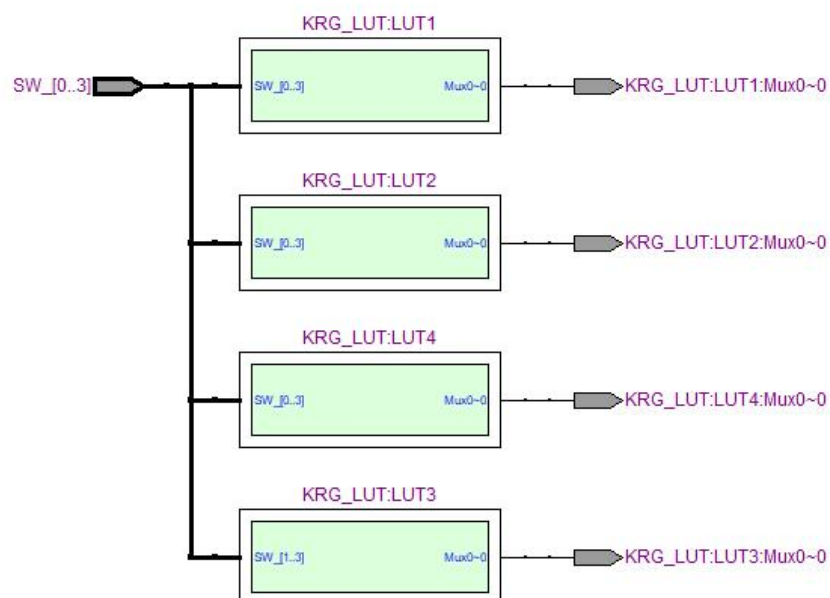


Рисунок 1.30 – Синтезированная схема технологических примитивов ПЛИС

Рисунок 1.31 – Синтезированная схема технологических примитивов ПЛИС

Рисунок 1.32 – Синтезированная схема технологических примитивов ПЛИС



ЗАКЛЮЧЕНИЕ

В рамках данной лабораторной работы спроектировали комбинационную логическую схему, реализованную в объёме одной из следующих ПЛИС: EPF10K50S, EPF10K100E или EPF10K200S. Комбинационная схема реализуется в объёме ПЛИС четырьмя различными способами:

1. модель на языке Verilog на основе минимальной ДНФ;
2. модель на языке Verilog на основе минимальной КНФ;
3. модель на языке Verilog на основе элементов LUT;
4. поведенческая модель на языке Verilog.

Выполнили этапы маршрута проектирования для синтеза, трансляции, реализации в кристалле ПЛИС и генерации файла конфигурации.

Протестировали спроектированную комбинационную логическую схему на плате.

ПРИЛОЖЕНИЯ

Приложение А — Исходный код проекта на языке Verilog

Приложение А

Исходный код проекта на языке Verilog

Листинг 1.1 — Модуль верхнего уровня иерархии на языке Verilog

```
module LR2_TOP(
    input SW_0, SW_1, SW_2, SW_3,

    output LED_0, LED_1, LED_2, LED_3,
    output LED_4, LED_5, LED_6, LED_7,
    output LED_8, LED_9, LED_A, LED_B,
    output LED_C, LED_D, LED_E, LED_F
);

    wire [3:0] y_BEH, y_DNF, y_KNF, y_LUT;
    wire [3:0] x;

    assign x = {SW_3, SW_2, SW_1, SW_0};

    LR2_KNF knf_module (
        .x(x),
        .y(y_KNF)
    );

    LR2_DNF dnf_module (
        .x(x),
        .y(y_DNF)
    );

    LR2_BEH beh_module (
        .x(x),
        .y(y_BEH)
    );

    LR2_LUT lut_module (
        .x(x),
        .y(y_LUT)
    );

    assign {LED_3, LED_2, LED_1, LED_0} = ~y_DNF;
    assign {LED_7, LED_6, LED_5, LED_4} = ~y_KNF;
    assign {LED_B, LED_A, LED_9, LED_8} = ~y_LUT;
    assign {LED_F, LED_E, LED_D, LED_C} = ~y_BEH;

endmodule
```

Листинг 1.2 — Модуль МДНФ с помощью операторов assign на языке Verilog

```
`timescale 1ns / 1ps

module LR2_DNF (
    input [3:0] x,
    output [3:0] y
);

    assign y[0] = x[1] & x[2] & x[3] | x[0] & ~x[1] & ~x[3] | x[3] & ~x[0] &
~x[2] | ~x[0] & ~x[1] & ~x[2];
    assign y[1] = x[0] & x[3] | x[1] & x[2] & x[3] | x[0] & ~x[1] & ~x[2] | x[3]
& ~x[1] & ~x[2] | x[1] & ~x[0] & ~x[2] & ~x[3] | x[2] & ~x[0] & ~x[1] & ~x[3];
    assign y[2] = x[0] & x[1] & ~x[3] | x[1] & x[3] & ~x[0] | x[1] & x[3] & ~x[2]
| x[2] & ~x[0] & ~x[3] | x[0] & x[2] & x[3] & ~x[1] | ~x[0] & ~x[1] & ~x[2];

endmodule
```

Продолжение Листинга 1.2

```
assign y[3] = x[0] & x[1] & ~x[2] | x[1] & x[3] & ~x[0] | x[2] & x[3] & ~x[0] |  
x[0] & x[2] & ~x[1] & ~x[3] | ~x[0] & ~x[1] & ~x[2] & ~x[3];  
  
endmodule
```

Листинг 1.3 — Модуль МКНФ с помощью операторов assign на языке Verilog

```
`timescale 1ns / 1ps  
  
module LR2_KNF(  
    input [3:0] x,  
    output [3:0] y  
);  
  
    assign y[0] = (x[3] | ~x[1]) & (x[0] | x[1] | ~x[2]) & (x[1] | ~x[2] | ~x[3])  
& (x[2] | ~x[0] | ~x[3]);  
    assign y[1] = (x[0] | x[1] | x[2] | x[3]) & (x[3] | ~x[0] | ~x[1]) & (x[3] |  
~x[0] | ~x[2]) & (x[3] | ~x[1] | ~x[2]) & (x[0] | x[1] | ~x[2] | ~x[3]) & (x[0]  
| x[2] | ~x[1] | ~x[3]);  
    assign y[2] = (x[1] | x[2] | ~x[0]) & (x[1] | x[3] | ~x[0]) & (x[0] | x[2] |  
x[3] | ~x[1]) & (x[0] | x[1] | ~x[2] | ~x[3]) & (~x[0] | ~x[1] | ~x[2] | ~x[3]);  
    assign y[3] = (x[0] | x[3] | ~x[1]) & (x[0] | x[3] | ~x[2]) & (x[1] | x[2] |  
~x[0]) & (x[1] | x[2] | ~x[3]) & (~x[0] | ~x[1] | ~x[2]) & (~x[0] | ~x[2] |  
~x[3]);  
  
endmodule
```

Листинг 1.4 — Модуль KRG LUT на языке Verilog

```
module KRG_LUT #(  
    parameter [15:0] INIT = 16'h0000  
)  
(  
    input [3:0] I,  
    output O  
);  
  
wire [15:0] RG_MEM;  
assign RG_MEM = INIT;  
assign O = RG_MEM[I];  
endmodule
```

Листинг 1.5 — Модуль LR2_BEH на языке Verilog

```
`timescale 1ns / 1ps  
  
module LR2_BEH(  
    input [3:0] x,  
    output reg [3:0] y  
);  
  
initial y = 4'h0;  
  
always@(x)  
    case(x)  
        4'h0: y = 4'hD;  
        4'h1: y = 4'h3;  
        4'h2: y = 4'h2;  
        4'h3: y = 4'hC;  
        4'h4: y = 4'h6;  
        4'h5: y = 4'h9;  
        4'h6: y = 4'h4;  
        4'h7: y = 4'h4;  
        4'h8: y = 4'h7;
```

```
4'h9: y = 4'h2;
```

Продолжение Листина 1.5

```
4'ha: y = 4'hD;  
4'hb: y = 4'hE;  
4'hc: y = 4'h8;  
4'hd: y = 4'h6;  
4'he: y = 4'hF;  
4'hf: y = 4'h3;  
endcase  
endmodule
```

Листинг 1.6 — Модуль верхнего уровня на языке Verilog

```
module LR2_TOP(  
    input SW_0, SW_1, SW_2, SW_3,  
  
    output LED_0, LED_1, LED_2, LED_3,  
    output LED_4, LED_5, LED_6, LED_7,  
    output LED_8, LED_9, LED_A, LED_B,  
    output LED_C, LED_D, LED_E, LED_F  
);  
  
    wire [3:0] y_BEH, y_DNF, y_KNF, y_LUT;  
    wire [3:0] x;  
  
    assign x = {SW_3, SW_2, SW_1, SW_0};  
  
    LR2_KNF knf_module (  
        .x(x),  
        .y(y_KNF)  
    );  
  
    LR2_DNF dnf_module (  
        .x(x),  
        .y(y_DNF)  
    );  
  
    LR2_BEH beh_module (  
        .x(x),  
        .y(y_BEH)  
    );  
  
    LR2_LUT lut_module (  
        .x(x),  
        .y(y_LUT)  
    );  
  
    assign {LED_3, LED_2, LED_1, LED_0} = ~y_DNF;  
    assign {LED_7, LED_6, LED_5, LED_4} = ~y_KNF;  
    assign {LED_B, LED_A, LED_9, LED_8} = ~y_LUT;  
    assign {LED_F, LED_E, LED_D, LED_C} = ~y_BEH;  
  
endmodule
```

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Методические указания по ПР № 1 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=405132> (Дата обращения: 23.09.2022).
2. Методические указания по ПР № 2 — URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=409130> (Дата обращения: 23.09.2022).
3. Смирнов С.С. Информатика [Электронный ресурс]: Методические указания по выполнению практических и лабораторных работ / С.С. Смирнов — М., МИРЭА — Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).
4. Тарасов И.Е. ПЛИС Xilinx. Языки описания аппаратуры VHDL и Verilog, САПР, приемы проектирования. — М.: Горячая линия — Телеком, 2021. — 538 с.: ил.
5. Антик М.И. Дискретная математика [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА — Российский технологический университет, 2018 — 1 электрон. опт. диск (CD-ROM).
6. Антик М.И. Математическая логика и программирование в логике [Электронный ресурс]: Учебное пособие / Антик М.И., Бражникова Е.В.— М.: МИРЭА – Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).
7. Жемчужникова Т.Н. Конспект лекций по дисциплине «Архитектура вычислительных машин и систем» — URL: https://drive.google.com/file/d/12OAi2_axJ6mRr4hCbXs-mYs8Kfp4YEFj/view?usp=sharing (Дата обращения: 23.09.2022).
8. Антик М.И. Теория автоматов в проектировании цифровых схем [Электронный ресурс]: Учебное пособие / Антик М.И., Казанцева Л.В. — М.: МИРЭА – Российский технологический университет, 2020. — 1 электрон. опт. диск (CD-ROM).