



Калужский филиал  
МГТУ им. Н.Э. Баумана  
Кафедра систем  
автоматического управления

# Логическое проектирование комбинационных устройств на основе ПЛИС фирмы Xilinx



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУ-КФ «Информатика и управление»

КАФЕДРА ИУЗ-КФ «Системы автоматического управления и  
электротехника»

## РАСЧЕТНО - ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому(ой) проекту (работе) на тему:

Логическое проектирование комбинационных устройств на основе ПЛИС  
фирмы Xilinx

по дисциплине Конструирование приборов

Студент гр.УТС.Б-61 \_\_\_\_\_ (Папкина М.П.)  
(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_ (Мельников Д.В.)  
(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_  
30-50 (дата)

Оценка проекта \_\_\_\_\_ баллов \_\_\_\_\_  
(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_ (\_\_\_\_\_)  
(подпись) (Ф.И.О.)

\_\_\_\_\_ (\_\_\_\_\_)  
(подпись) (Ф.И.О.)

\_\_\_\_\_ (\_\_\_\_\_)  
(подпись) (Ф.И.О.)

Калуга, 2020

УТВЕРЖДАЮ

Заведующий кафедрой ИУЗ-КФ

Мельников Д.В. ( Мельников Д.В. )

«15» февраля 2020.

## **ЗАДАНИЕ** **на выполнение курсового проекта**

по дисциплине **Конструирование приборов**

Студент Папкина М.П. УТС.Б-61

(фамилия, инициалы, индекс группы)

Руководитель Мельников Д.В.

(фамилия, инициалы)

График выполнения проекта: 25% к 4 нед., 50% к 7 нед., 75% к 10 нед., 100% к 14 нед.

### **1. Тема курсового проекта**

**Логическое проектирование комбинационных устройств на основе ПЛИС  
фирмы Xilinx**

### **2. Техническое задание**

Синтезировать логическую схему сумматора по модулю 10. Весовые коэффициенты  
кода: 6-4-2-1. Кодирование цифр 0000-0, 0001-1, 0010-2, 0011-3, 0100-4, 0101-5, 0110-  
6, 0111-7, 1010-8, 1011-9

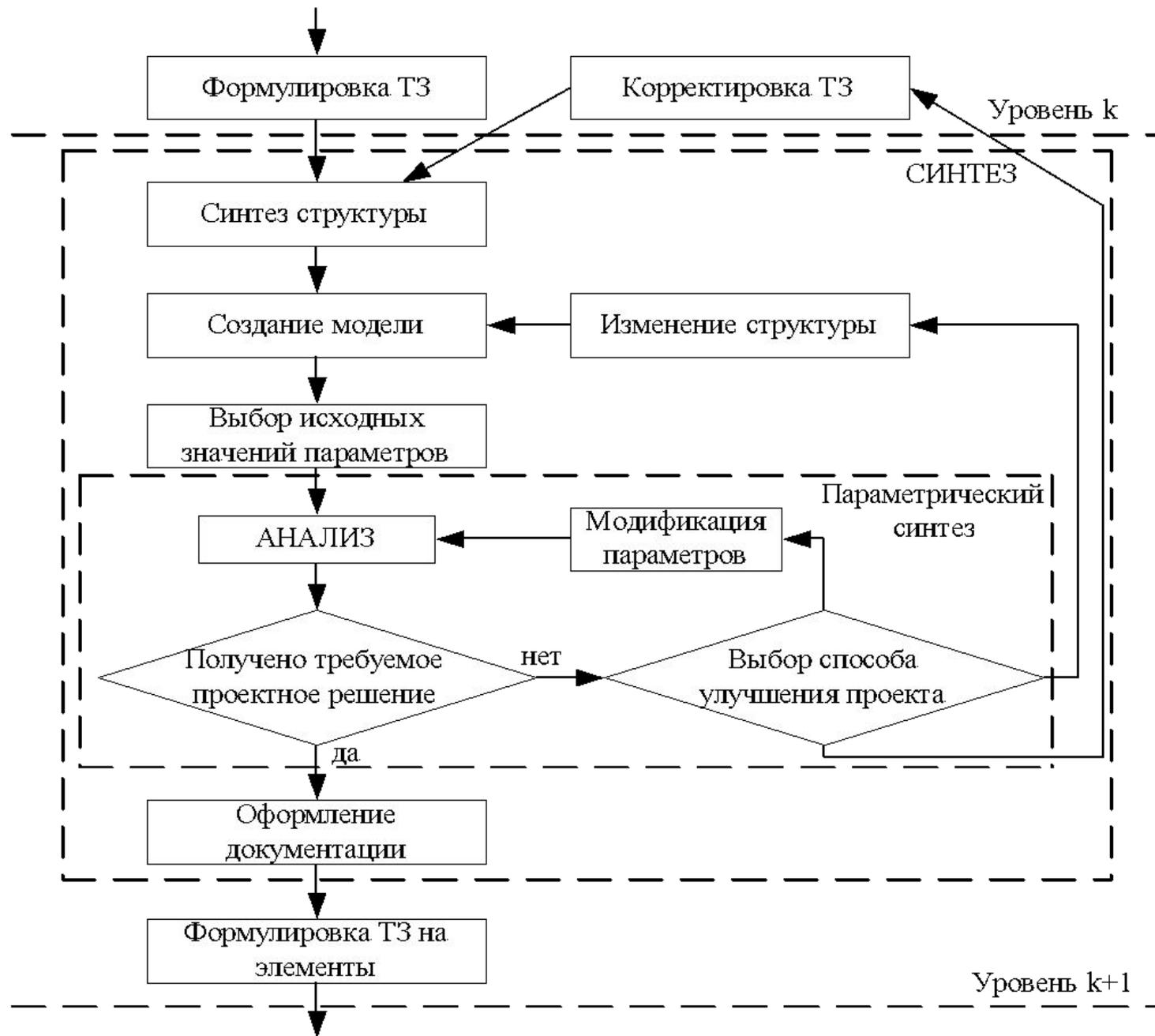
Товарищу НАННИКОВУ Б.И.

Тактико-техническое задание  
на Атомную Бомбу.

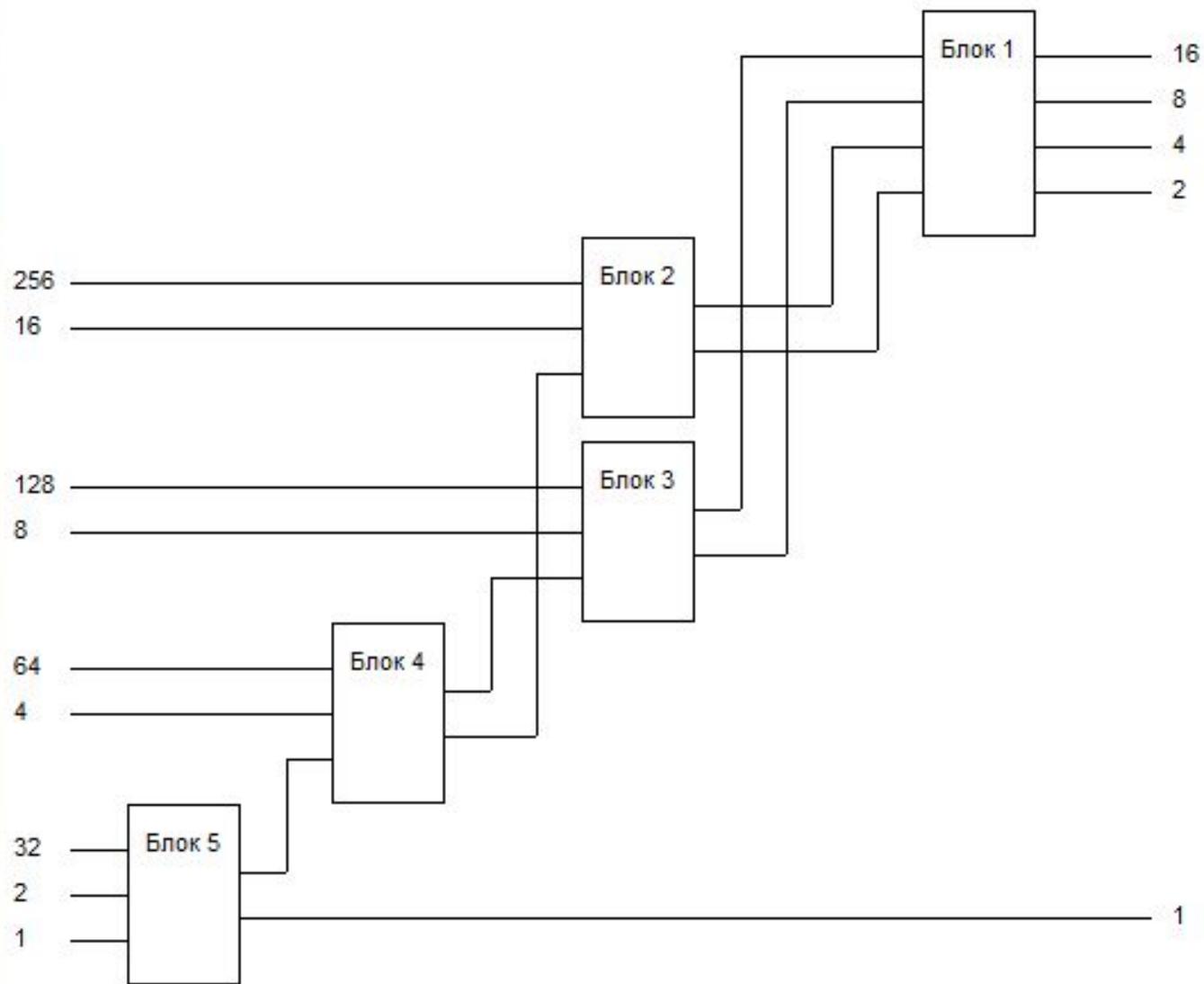
1. Атомная БМБА разрабатывается в двух вариантах.  
В варианте I рабочим веществом является плутоний  
В варианте II — уран 235
2. В варианте I переход через критическое состояние  
осуществляется посредством взрыва специально сконструированного заряда составленного из блоков обычного взрывчатого вещества, образующих полую сферу с плутонием в варианте II переход осуществляется посредством сближения двух тел из урана внутри сферы из стальной пушки
3. В первом варианте в центре плутониевого заряда помещается нейтронный взрыватель, обеспечивающий возникновение варианта основного заряда плутония при максимальной плотности
4. Детонация сферы из взрывчатого вещества осуществляется посредством одновременного (с точностью до 0,3 микросекунд) срабатывания группы распределенных по поверхности сфер не управляемых электронно-детонаторов, автоматически высвобождающих взрывчатое

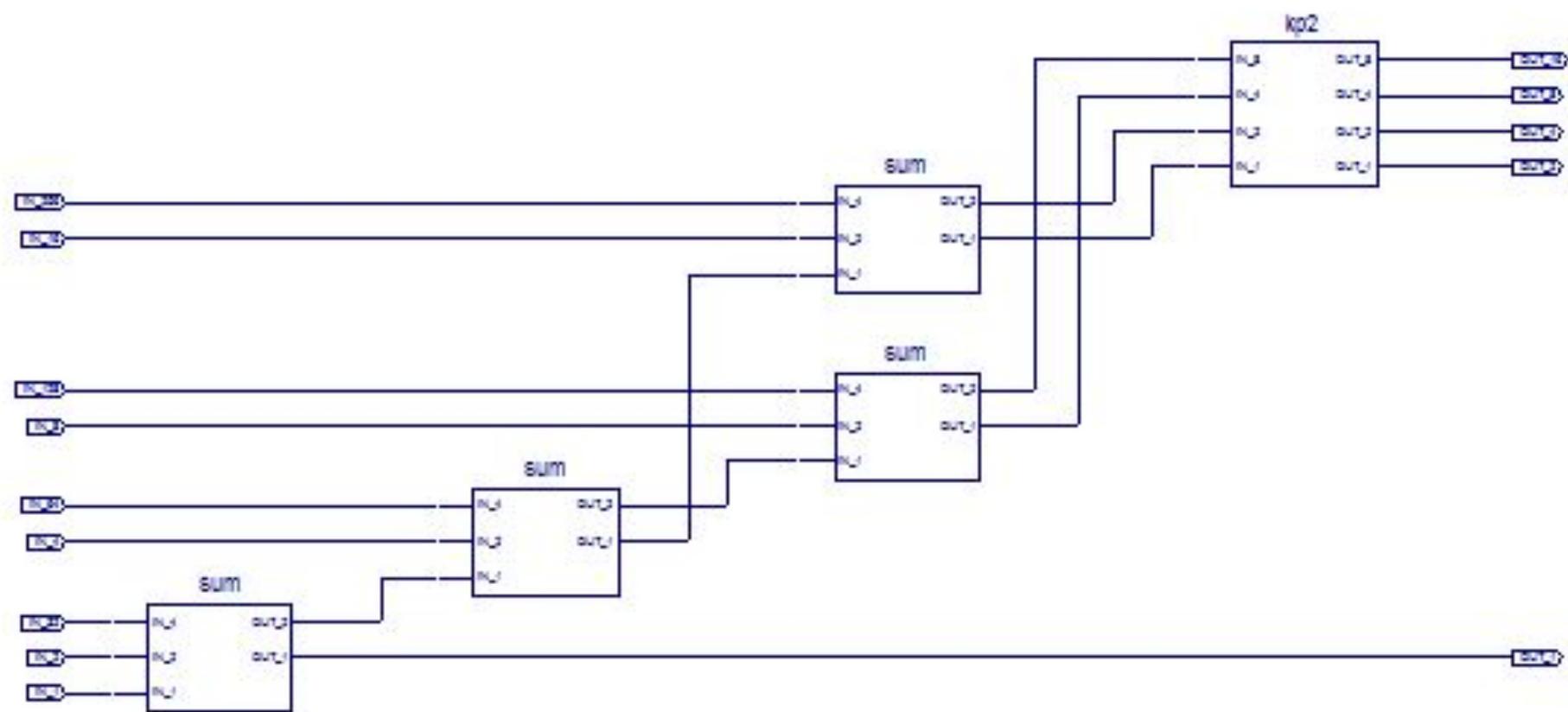
Тактико-техническое задание на атомную бомбу,  
подготовленное Ю. Б. Харитоном

**Тактико-техническое задание  
на атомную бомбу.**

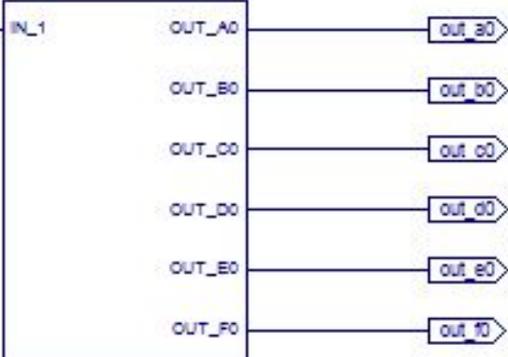


# 1 вариант схемы

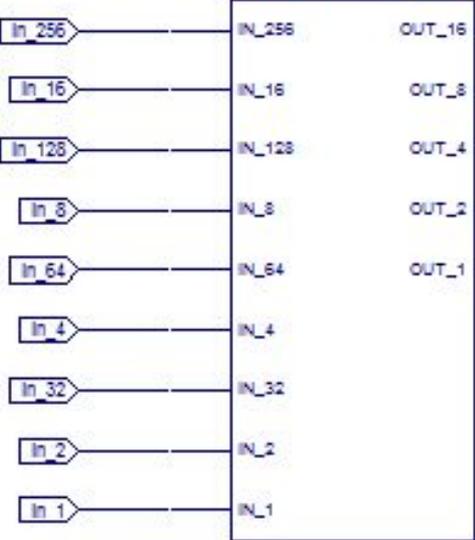




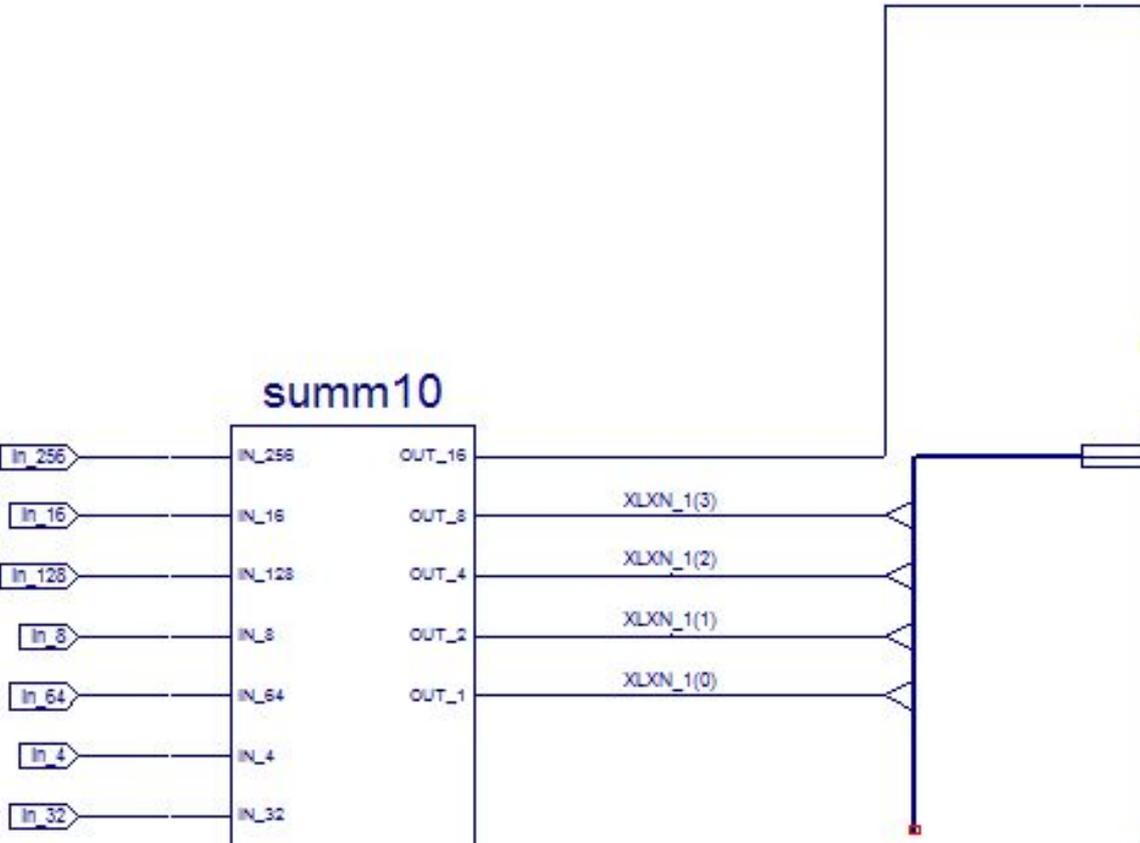
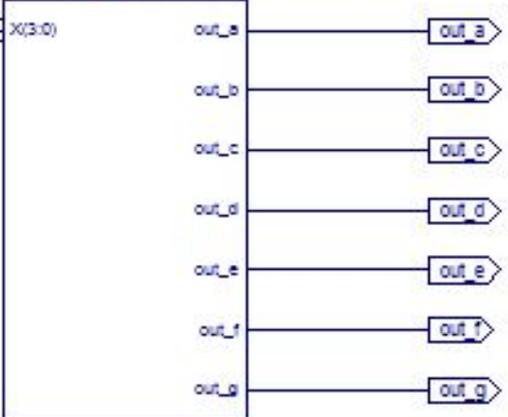
dc6



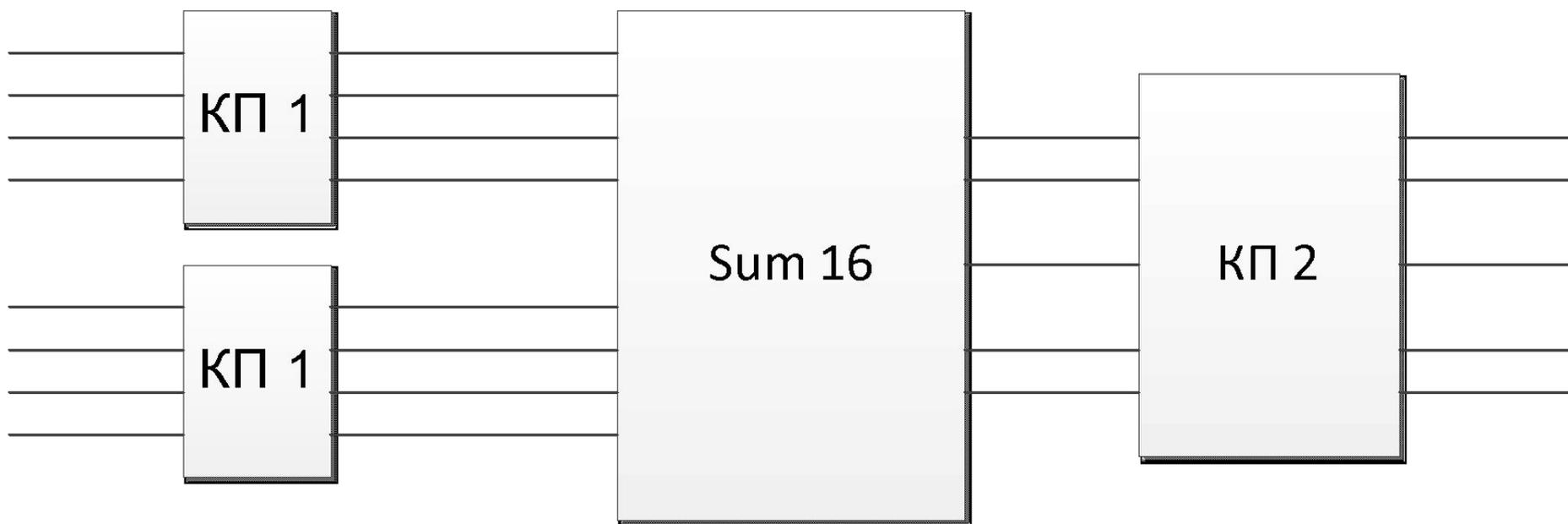
summ10



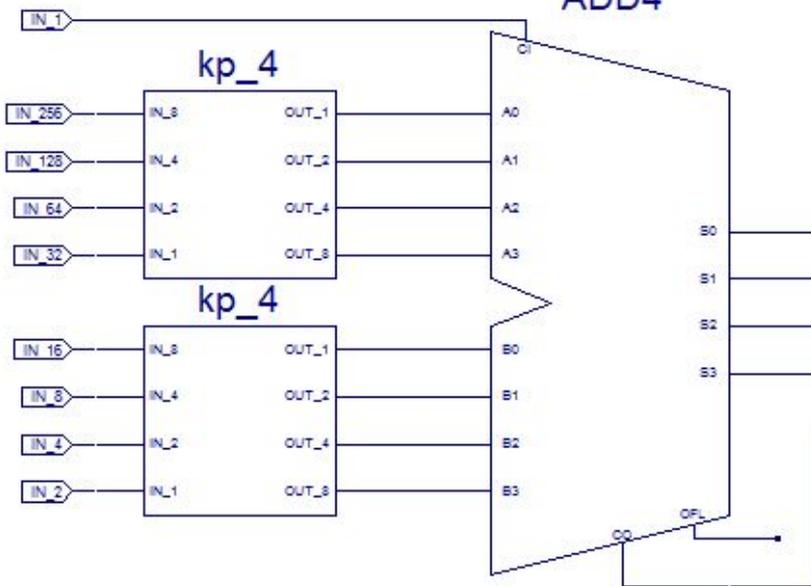
dc7



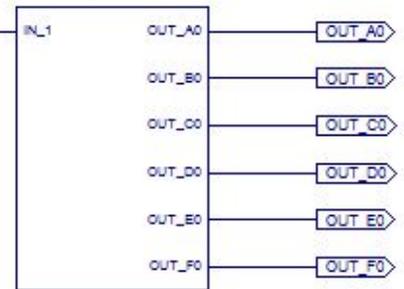
# 2 вариант схемы



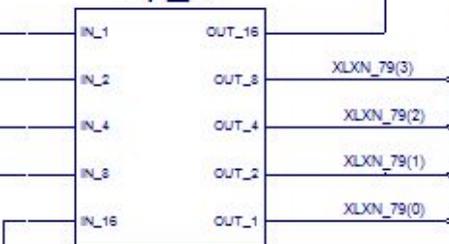
# ADD4



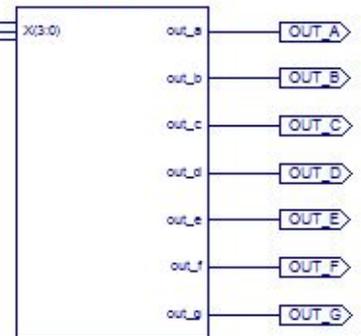
# dc6



# kp\_5



# dc7



# 3 вариант схемы – описание на языке VHDL

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ADD is
    Port(A: in std_logic_vector(3 downto 0);
         B: in std_logic_vector(3 downto 0);
         S: out std_logic_vector(3 downto 0);
         CI: in std_logic;
         CO: out std_logic);
end ADD;

architecture Behavioral of ADD is
    signal sum: std_logic_vector(4 downto 0);

begin
    sum <= ('0'&A) + ('0'&B) + CI;
    S <= sum(3 downto 0) when sum < "01010" else (sum - "01010");
    CO <= '0' when sum < "01010" else '1';

end Behavioral;
```

## 2.5. Описание сумматора на языке VHDL

Здесь приведен программный код, позволяющий описать работу двоичного сумматора по модулю 12:

```
entity blok_1 is
port(a,b:in integer range 0 to 15;
      pi:in integer range 0 to 1;
      c:out integer range 0 to 15;
      pil:out integer range 0 to 1);
end blok_1;

architecture functional of blok_1 is
signal aa,bb,dd:integer range 0 to 15;
signal cc:integer range 0 to 31;
begin
p0:process(a,b,pi)
begin
--замена на числа в десятичной системе
case a is
  when 8 =>
    aa<=0;
  when 9=>
    aa<=0;
  when 10=>
    aa<=0;
  when 11 =>
    aa<=8;
  when 12 =>
    aa<=9;
  when 13=>
    aa<=10;
  when 14 =>
    aa<=11;
  when 15 =>
    aa<=0;
  when others =>
    aa<=a;
end case;

case b is

  when 8 =>
    bb<=0;
  when 9=>
    bb<=0;
  when 10=>
    bb<=0;
  when 11 =>
    bb<=8;
  when 12 =>
    bb<=9;
  when 13=>
    bb<=10;
  when 14 =>
    bb<=11;
  when 15 =>
    bb<=0;

    when others =>
      bb<=b;
end case;

cc<=aa+bb+pi;

--переход к нашей системе счисления
pil<=0;

if (cc>11) then
cc<=cc-11;
pil<=1;
end if;

case cc is
  when 8=>
    dd<=11;
  when 9=>
    dd<=12;
  when 10=>
    dd<=13;
  when 11=>
    dd<=14;
  when 12=>
    dd<=0;
  when 13=>
    dd<=0;
  when 14=>
    dd<=0;
  when 15=>
    dd<=0;
  when others =>
    dd<=cc;
end case;

end process;
c<=dd;
end functional;
```

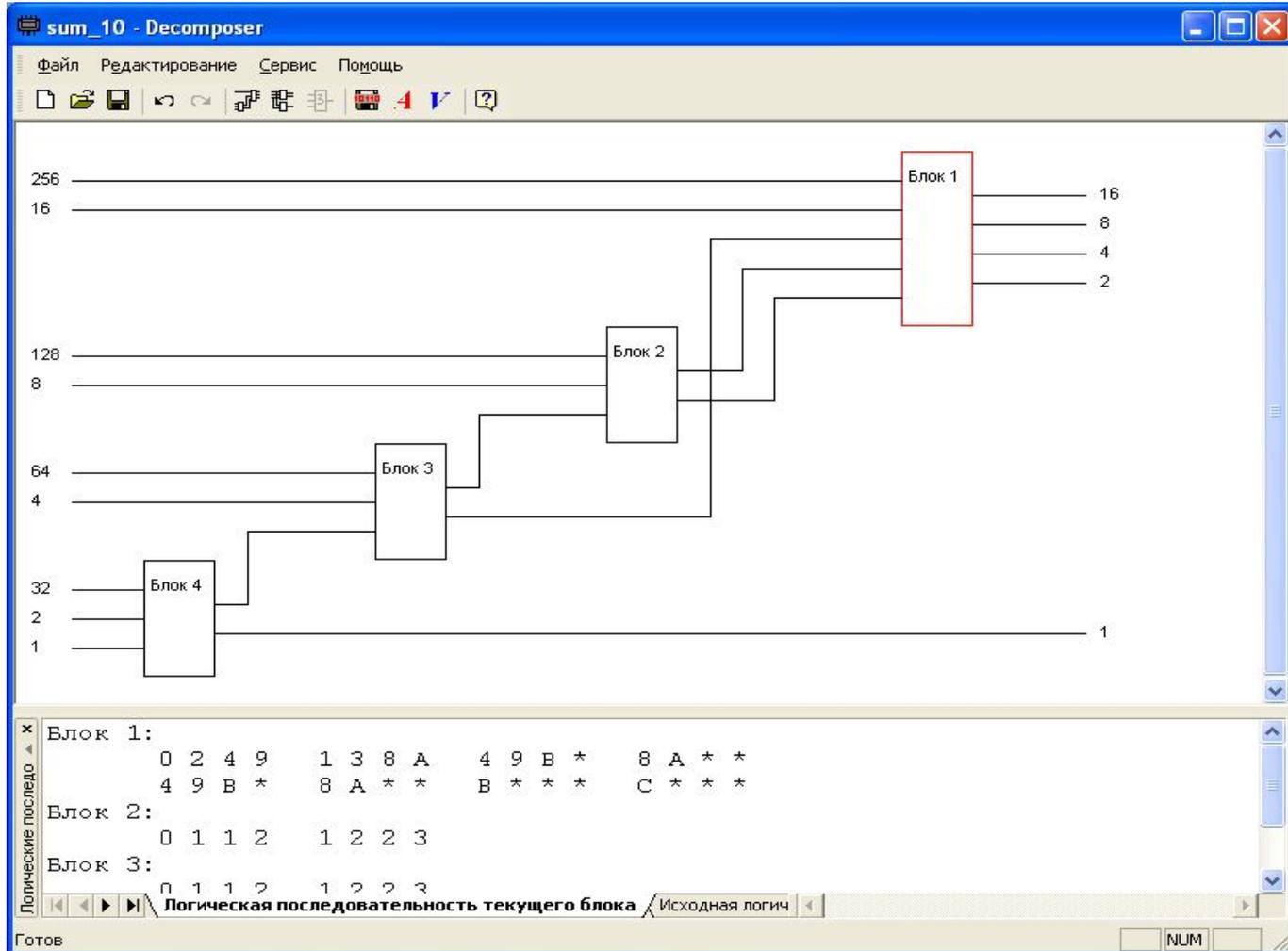
# Сравнение затрат ресурсов ПЛИС

Вариант реализации	Макроячейки ( <i>macrocells</i> )	Конъюнкт. термы ( <i>product terms</i> )	Входы функц. блоков ( <i>func. block inputs</i> )
Схема на основе методов декомпозиции	32 ( из 288) (11%)	337 (из 1440) (23%)	188 (из 864) (22%)
Схема на основе стандартного сумматора	30 ( из 288) (10%)	322 (из 1440) (22%)	163 (из 864) (19%)
Описание на языке VHDL	44 ( из 288) (15%)	430 (из 1440) (30%)	245 (из 864) (28%)

Можно получить несколько  
подобных таблиц:  
для разных семейств ПЛИС:

- CPLD xc95288xl;
- CPLD CoolRunner;
- FPGA Spartan3;
- для различных настроек  
программы трассировки

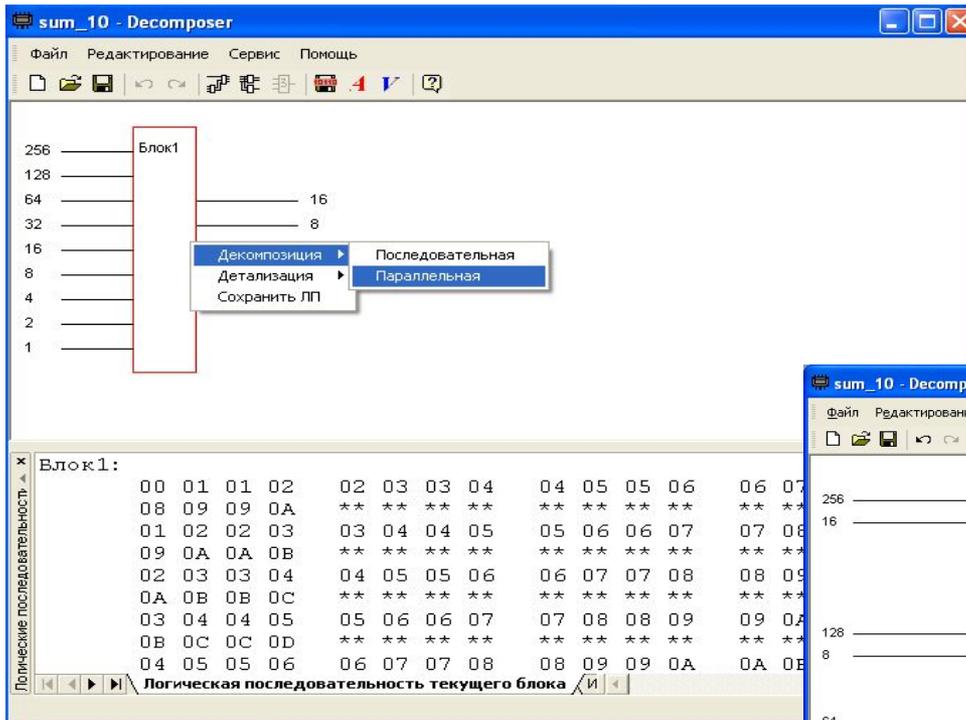
# САПР *Decomposer*



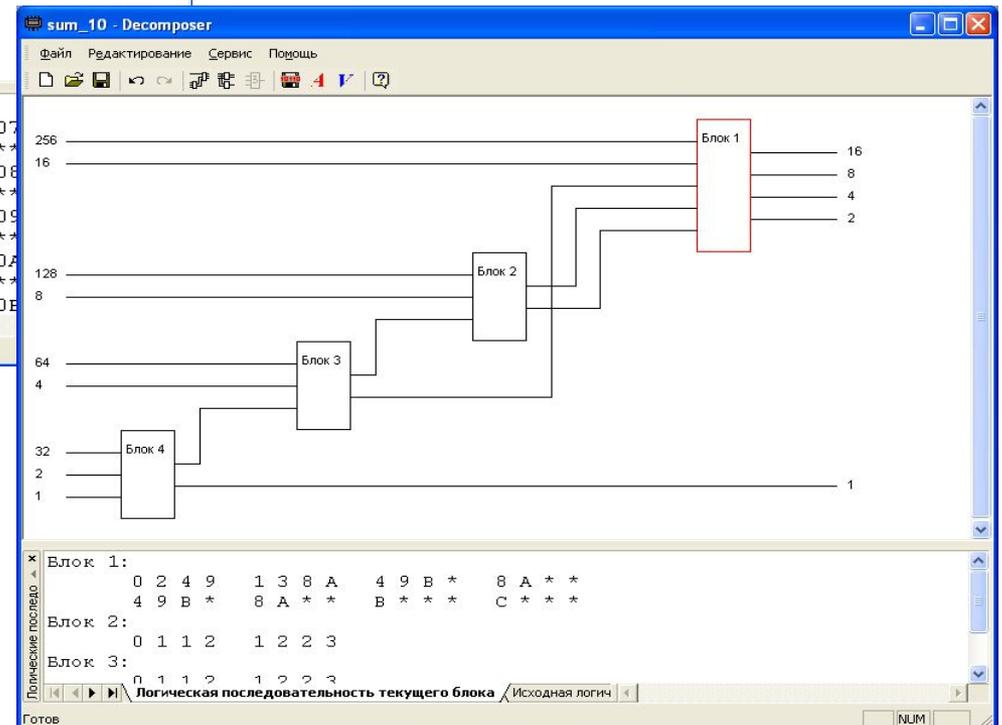
# Возможности САПР *Decomposer*

- Задание проектируемой схемы в виде числовой (логической) последовательности
- Проведение параллельной и последовательной декомпозиции
- Детализация схемы до уровня двухвходовых блоков
- Анализ полученной схемы на соответствие исходному описанию
- Трансляция схемы в описание на языке *VHDL*

# Главное окно приложения



Результат декомпозиции



Исходное описание

## 2. ПРИМЕР СИНТЗА КОМБИНАЦИОННОГО УСТРОЙСТВА

### 2.1. Абстрактный синтез проектируемого устройства

В качестве примера рассмотрим сумматор, который изображен на рис. 13.

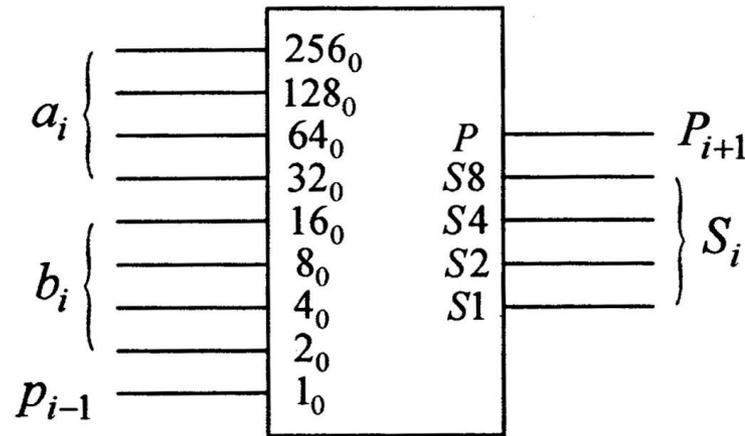


Рис. 13. Структурная схема синтезируемого устройства

Схема данного сумматора имеет 9 входов и 5 выходов.  $a_i$  и  $b_i$  – шестнадцатеричные цифры – 0, 1, 2 ..., A, B, C, D, E, F.  $p_{i-1}$  и  $P_{i+1}$  входной и выходной сигналы переноса, принимают значения 1 или 0 (есть перенос или нет). Сложность исходной схемы  $S_0 = 5 \cdot 2^9 = 2560$  бит.

*Исходные данные:*

Весовые коэффициенты: 5-4-2-1

Таблица 2

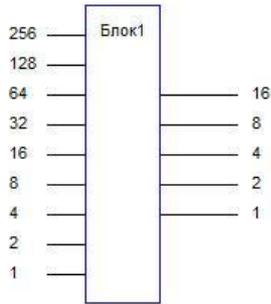
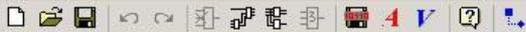
Цифры	Код
0	0000 (0)
1	0001 (1)
2	0010 (2)
3	0011 (3)
4	0100 (4)
5	0101 (5)
6	0110 (6)
7	0111 (7)
8	1011 (B)
9	1100 (C)





Логические последовательности

00	01	01	02	02	03	03	04	04	05	05	06	06	07	07	0E
**	**	**	**	**	**	**	**	**	**	**	**	0E	0F	0F	10
01	02	02	03	03	04	04	05	05	06	06	07	07	0E	0E	0F
**	**	**	**	**	**	**	**	**	**	**	**	0F	10	10	11
02	03	03	04	04	05	05	06	06	07	07	0E	0E	0F	0F	10
**	**	**	**	**	**	**	**	**	**	**	**	10	11	11	12
03	04	04	05	05	06	06	07	07	0E	0E	0F	0F	10	10	11
**	**	**	**	**	**	**	**	**	**	**	**	11	12	12	13
04	05	05	06	06	07	07	0E	0E	0F	0F	10	10	11	11	12
**	**	**	**	**	**	**	**	**	**	**	**	12	13	13	14
05	06	06	07	07	0E	0E	0F	0F	10	10	11	11	12	12	13
**	**	**	**	**	**	**	**	**	**	**	**	13	14	14	15
06	07	07	0E	0E	0F	0F	10	10	11	11	12	12	13	13	14
**	**	**	**	**	**	**	**	**	**	**	**	14	15	15	16
07	0E	0E	0F	0F	10	10	11	11	12	12	13	13	14	14	15
**	**	**	**	**	**	**	**	**	**	**	**	15	16	16	17
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
**	**	**	**	**	**	**	**	**	**	**	**	**	**	**	**
0E	0F	0F	10	10	11	11	12	12	13	13	14	14	15	15	16
**	**	**	**	**	**	**	**	**	**	**	**	16	17	17	1E
0F	10	10	11	11	12	12	13	13	14	14	15	15	16	16	17
**	**	**	**	**	**	**	**	**	**	**	**	17	1E	1E	1F



### Сохранить как

Папка: Рабочий стол

- Недавние места
- Рабочий стол
- Библиотеки
- Компьютер
- Сеть

- Библиотеки
- Домашняя группа
- КВН
- Компьютер
- Сеть
- ChipProg\_LPT
- Tor Browser
- ВД\_ЛСЦ
- Материалы по КР ОЦСх\_2017
- Материалы по VHDL
- Материалы по ПрМК\_1\_2017
- Материалы по ТСАиУ\_2017
- Тест ДПТ\_ТГ\_ГН
- 2016\_Методички\_В разработке -
- 2016\_НОВЫЕ ПРГРАММЫ - Ярлы
- 2016\_ПОЛЕВЫЕ КЛЮЧИ - Ярлык
- ARHIV (H)
- CD-дисковод
- Downloads
- Sum\_11\_8421.lof
- Библиотеки - Ярлык
- Диск (C)
- Диск (D)
- Диск (E)
- Диск (G)
- Компьютер

Имя файла: \*.lof

Тип файла: **Файлы логической последовательности (\*.lof)**

Сохранить Отмена

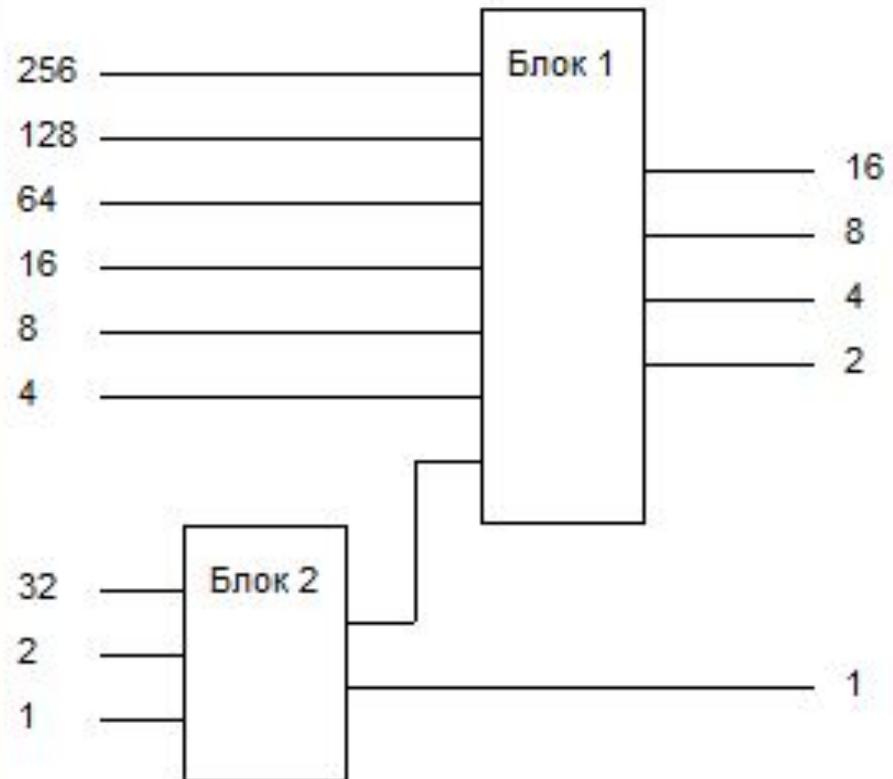
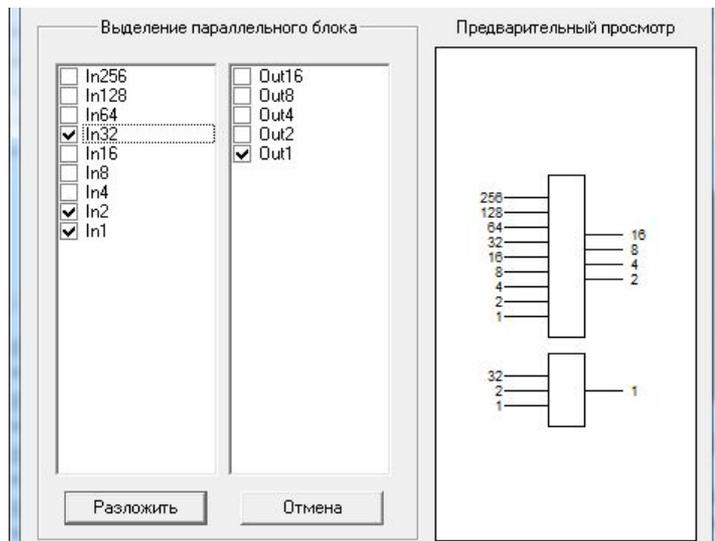
```

Блок1:
00 01 01 02    02 03 03 04    04 05 05 06    06 07 07 08
08 09 09 0A    0A 10 ** **    ** ** ** **
01 02 02 03    03 04 04 05    05 06 06 07    07 08 08 09
09 0A 0A 10    10 11 ** **    ** ** ** *
02 03 03 04    04 05 05 06    06 07 07 08    08 09 09 0A
0A 10 10 11    11 12 ** **    ** ** ** *
03 04 04 05    05 06 06 07    07 08 08 09    09 0A 0A 10
10 11 11 12    12 13 ** **    ** ** ** *
04 05 05 06    06 07 07 08    08 09 09 0A    0A 10 10 11
11 12 12 13    13 14 ** **    ** ** ** *
    
```

Логическая последовательность текущего блока / Исходная логика

# Параллельная декомпозиция

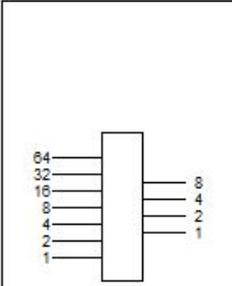
	In1	In2	In4	In8	In16	In32	In64	In128	In256
Out16	1	1	1	1	1	1	1	1	1
Out8	1	1	1	1	1	1	1	1	1
Out4	1	1	1	1	1	1	1	1	1
Out2	1	1	1	1	1	1	1	1	1
Out1	1	1	0	0	0	1	0	0	0



# Последовательная декомпозиция

Последовательная декомпозиция

Исходный блок

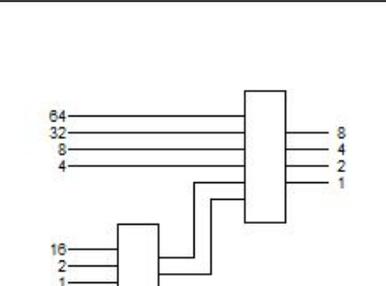


Исходная сложность: 512

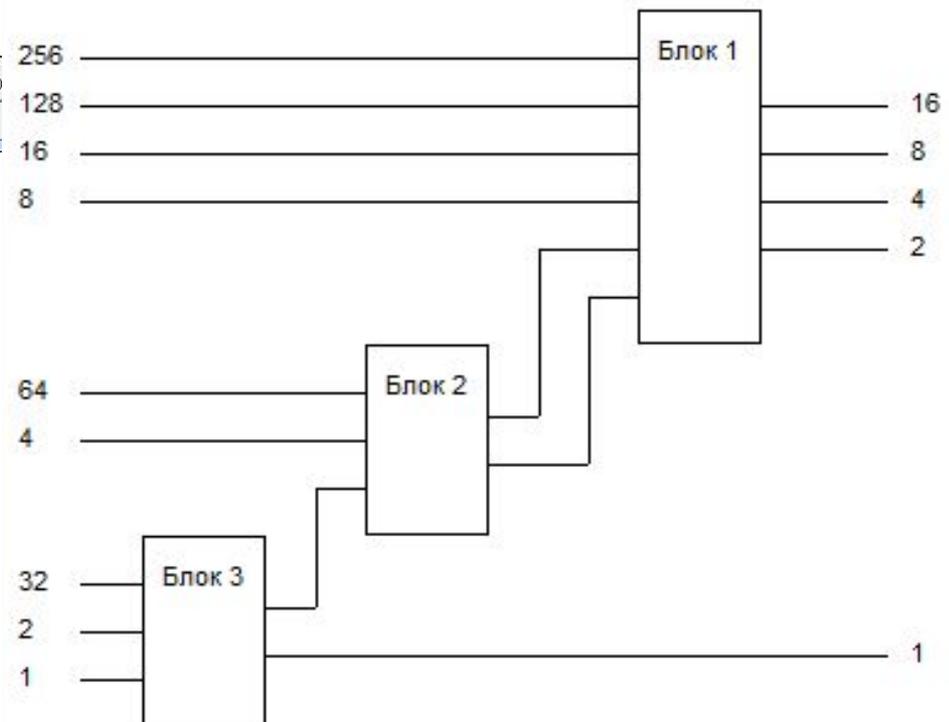
Поиск разложения с неувеличением сложности

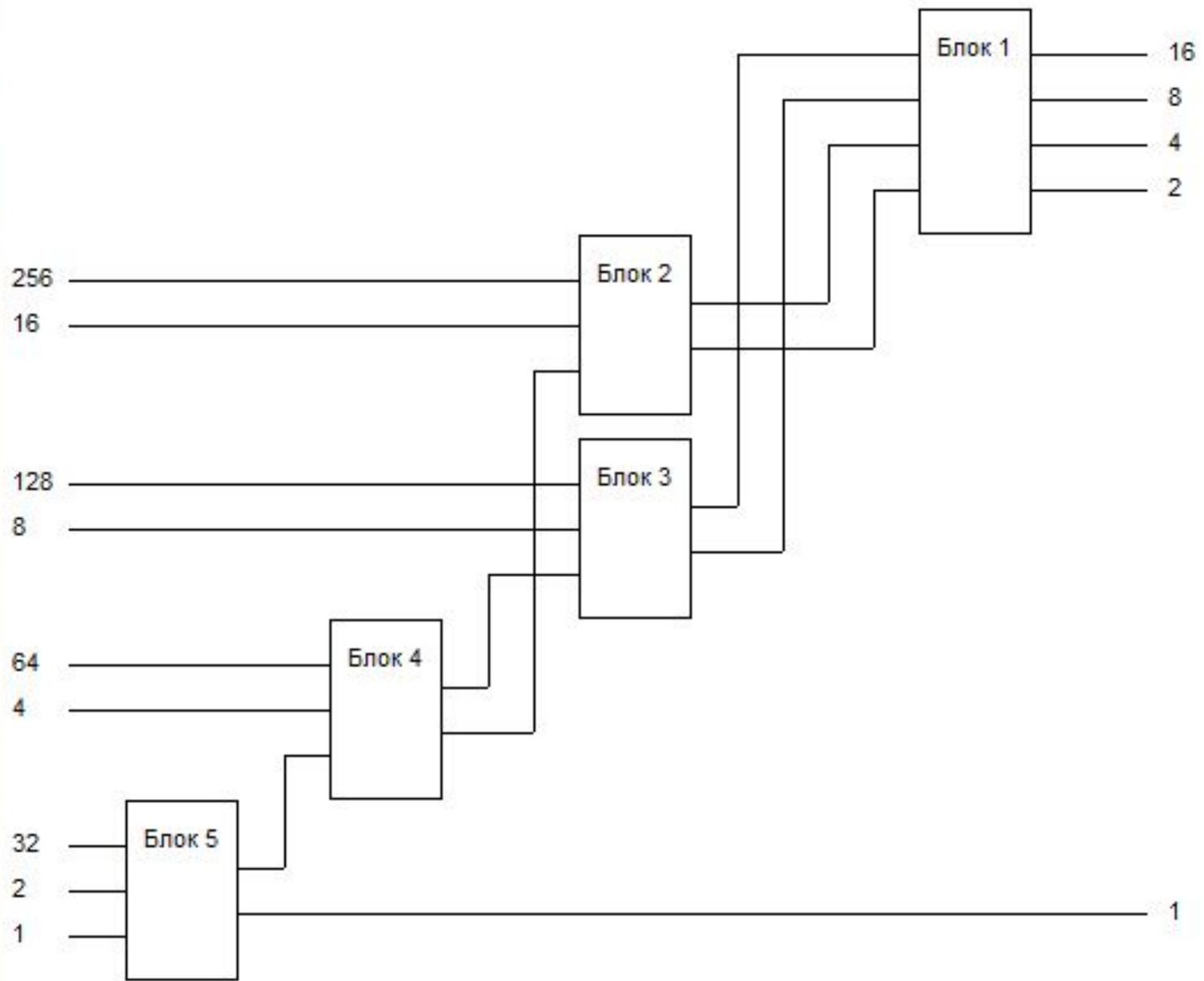
Сложность	Старший блок (веса)	Младший блок
Полученная 224	64, 32, 16, 2, 1,	8, 4,
Полученная 224	64, 32, 16, 4, 1,	8, 2,
Полученная 224	64, 32, 16, 4, 2,	8, 1,
Полученная 224	64, 32, 16, 8, 1,	4, 2,
Полученная 224	64, 32, 16, 8, 2,	4, 1,
Полученная 224	64, 32, 16, 8, 4,	2, 1,
Полученная 256	64, 16, 8, 4, 2, 1,	32,
Полученная 256	64, 32, 16, 8, 2, 1,	4,
Полученная 272	8, 2, 1,	64, 32, 16, 4,
<b>Полученная 272</b>	<b>16, 2, 1,</b>	<b>64, 32, 8, 4,</b>
Полученная 272	16, 8, 1,	64, 32, 4, 2,
Полученная 272	16, 8, 2,	64, 32, 4, 1,
Полученная 272	64, 2, 1,	32, 16, 8, 4,
Полученная 272	64, 8, 1,	32, 16, 4, 2,
Полученная 272	64, 8, 2,	32, 16, 4, 1,

Полученные блоки

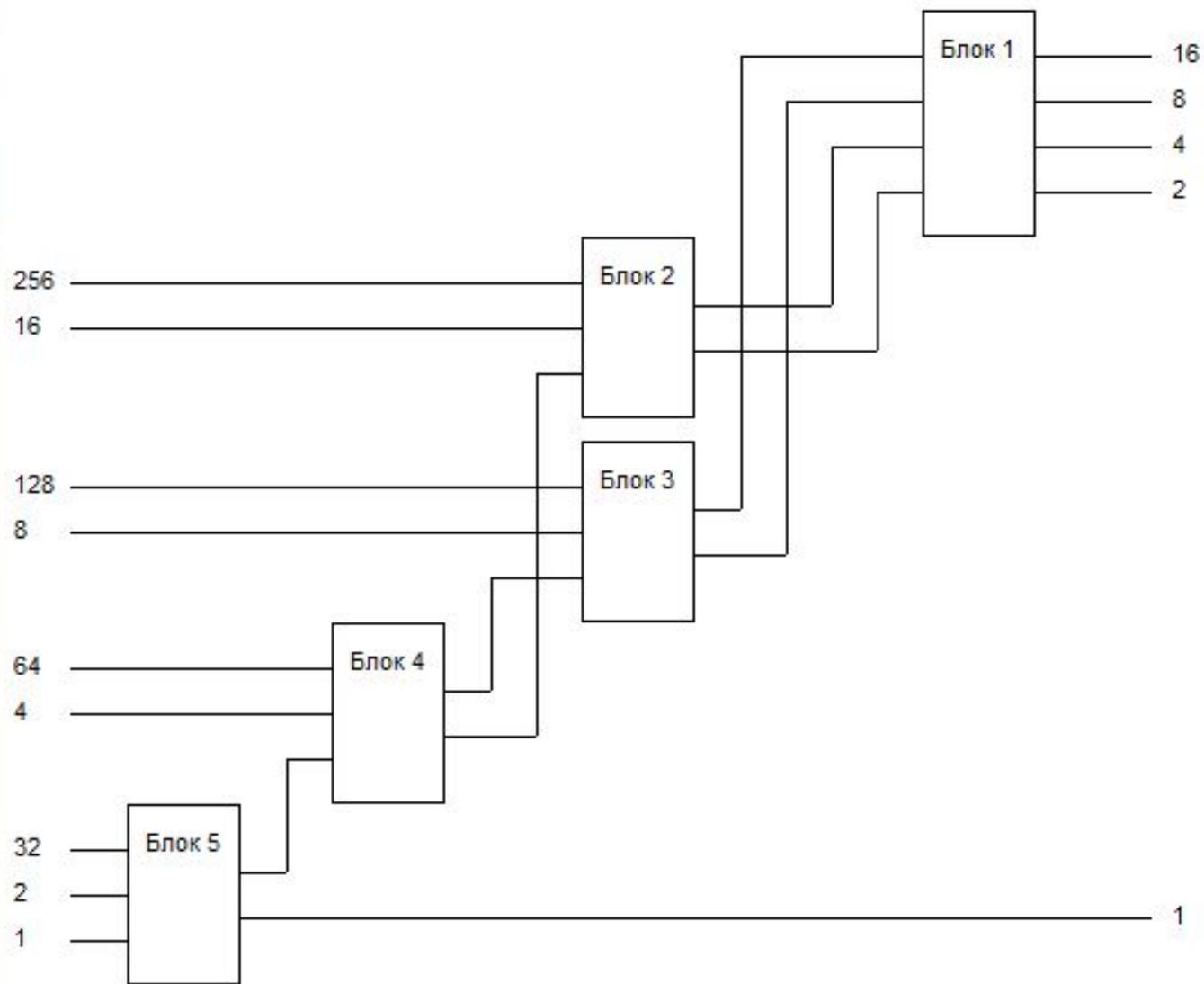


Разло





# 1 вариант схемы

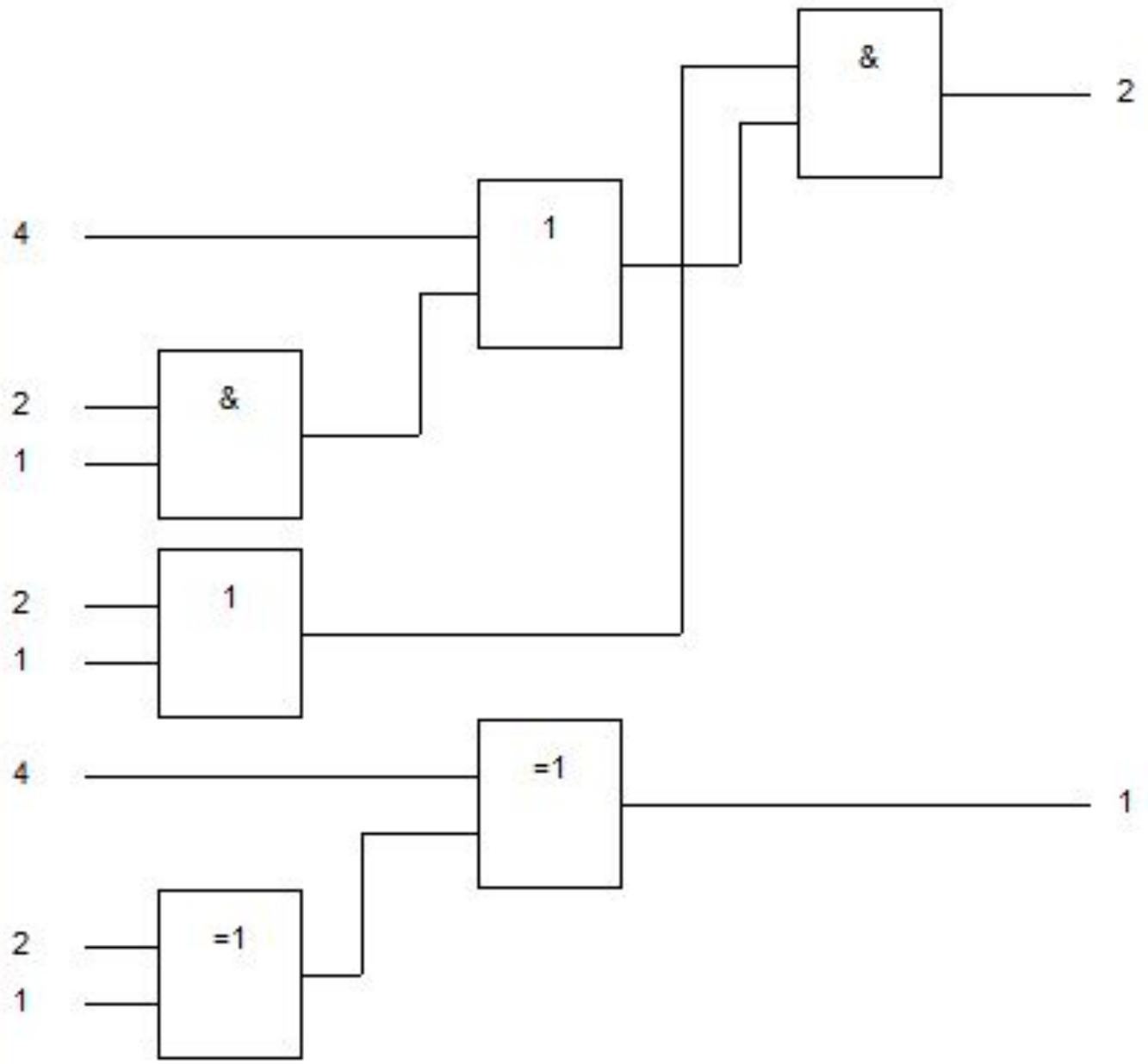


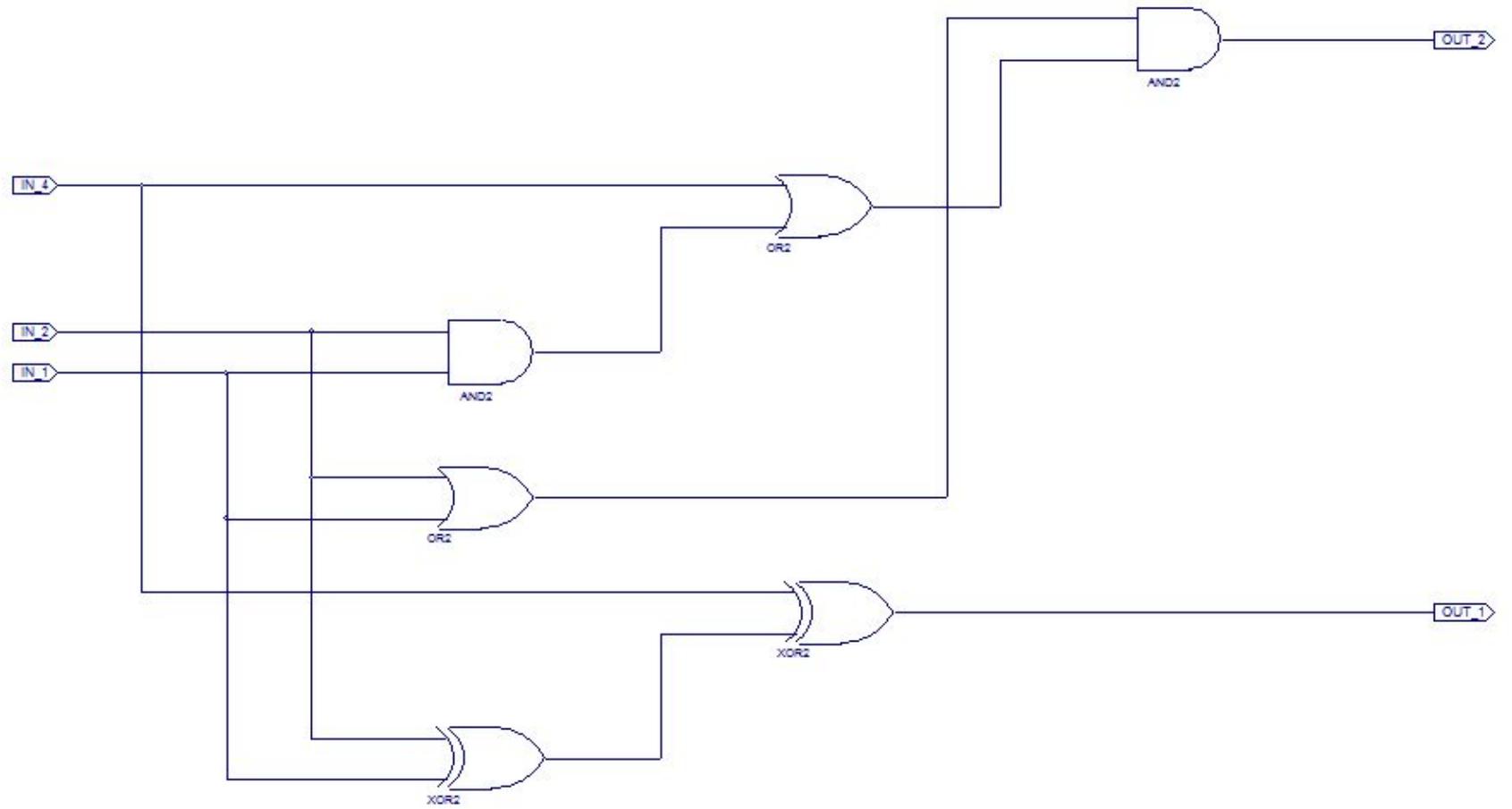
# Анализ схемы

		Исходная последовательность				Полученная последовательность					
		0	1	2	3		0	1	2	3	
OK	0	00	01	01	02		00	01	01	02	▲
OK	1	02	03	03	04		02	03	03	04	
OK	2	04	05	05	06		04	05	05	06	
OK	3	06	07	07	0E		06	07	07	0E	
OK	4	**	**	**	**		02	03	03	**	
OK	5	**	**	**	**		**	**	**	06	
OK	6	**	**	**	**		06	07	07	0E	
OK	7	0E	0F	0F	10		0E	0F	0F	10	
OK	8	01	02	02	03		01	02	02	03	
OK	9	03	04	04	05		03	04	04	05	
OK	10	05	06	06	07		05	06	06	07	
OK	11	07	0E	0E	0F		07	0E	0E	0F	
OK	12	**	**	**	**		03	**	**	**	
OK	13	**	**	**	**		**	06	06	07	▼

 OK

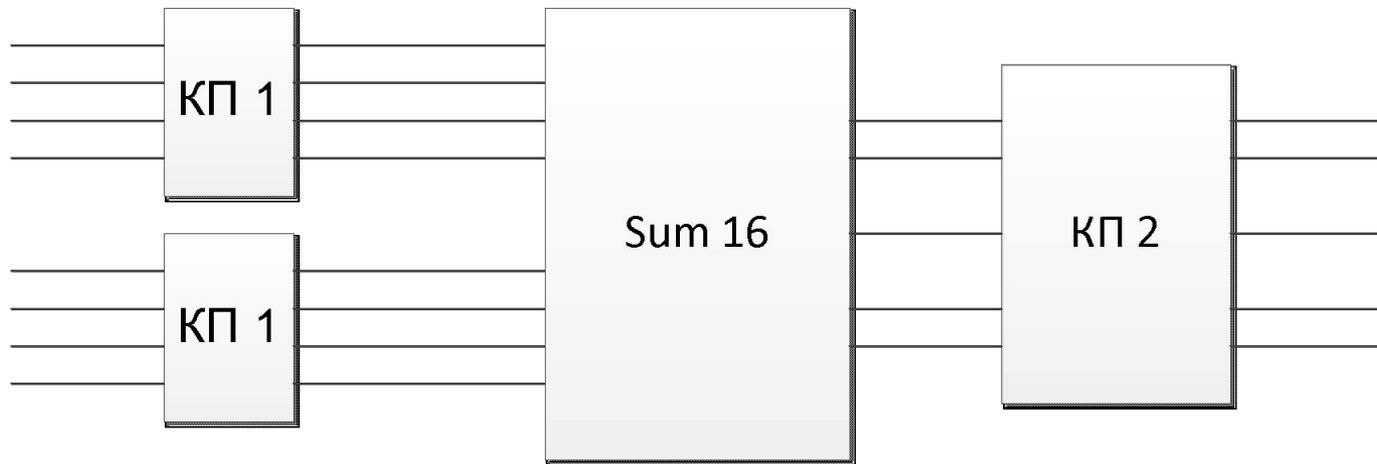
**Детализация схемы и  
покрытие заданными  
логическими элементами**





## 2 вариант схемы:

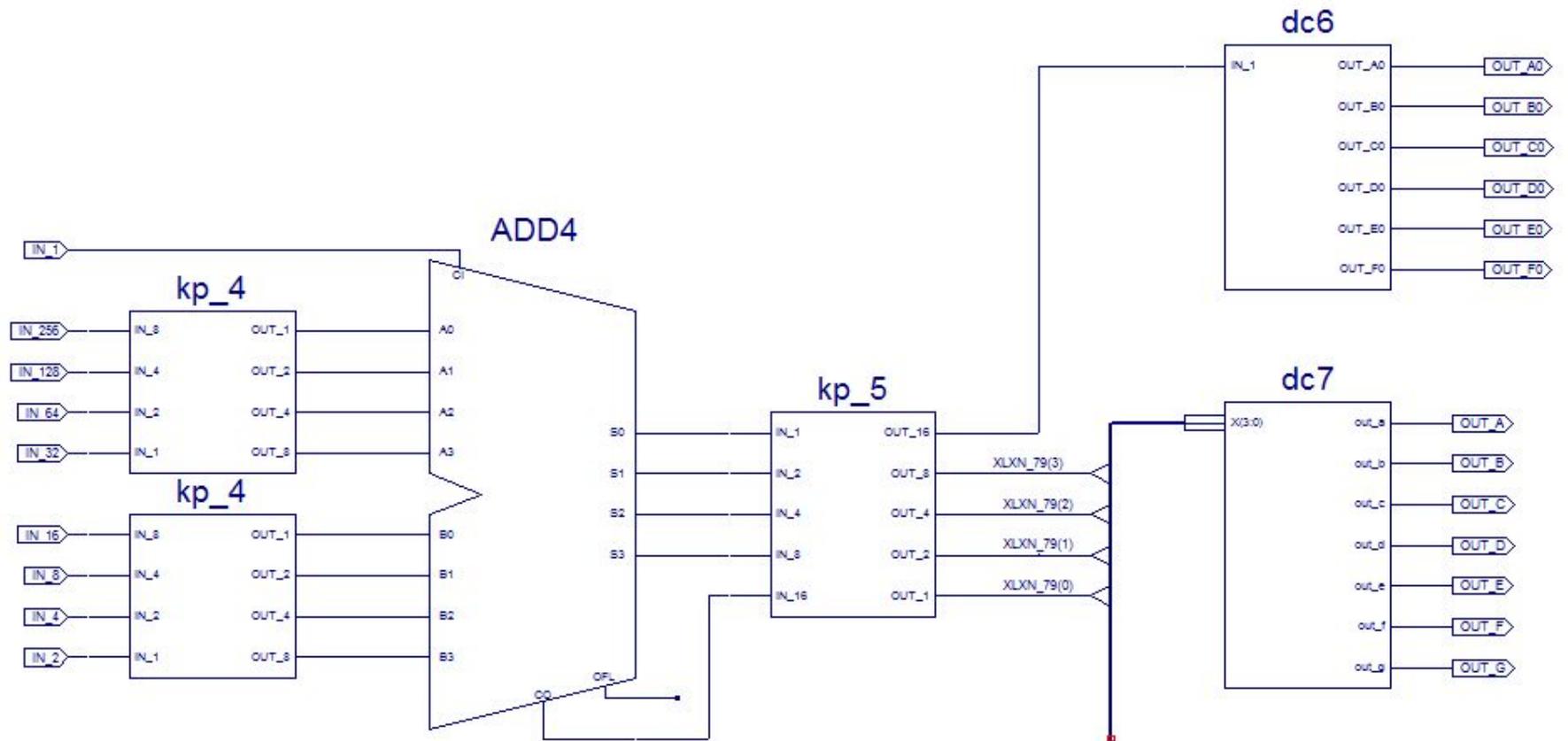
на основе стандартного четырехразрядного сумматора и входных/выходных кодопреобразователей



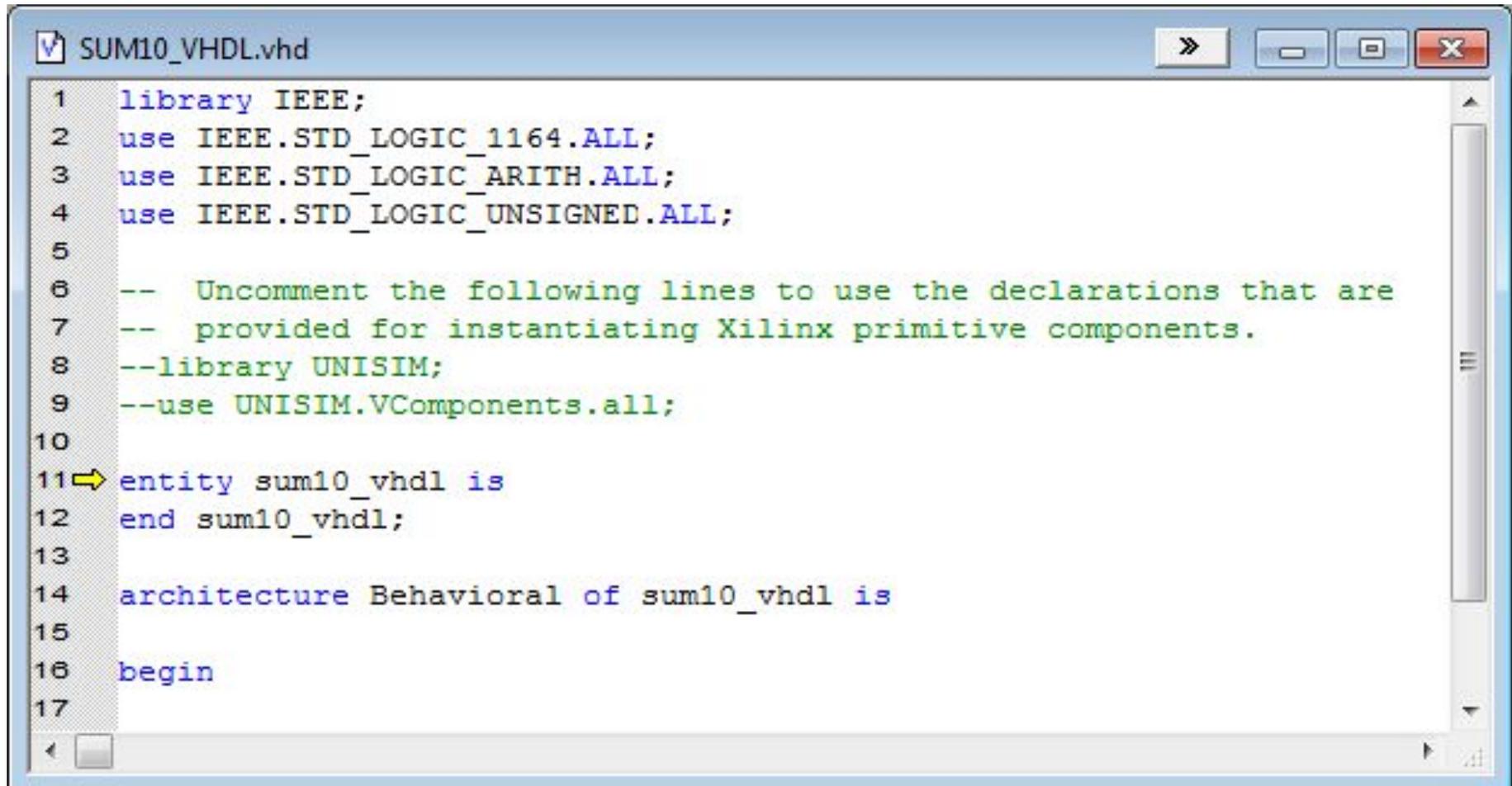
Числовая последовательность  
ВХОДНОГО  
кодопреобразователя:

0123 4567 \*\*\*8 9\*\*\*





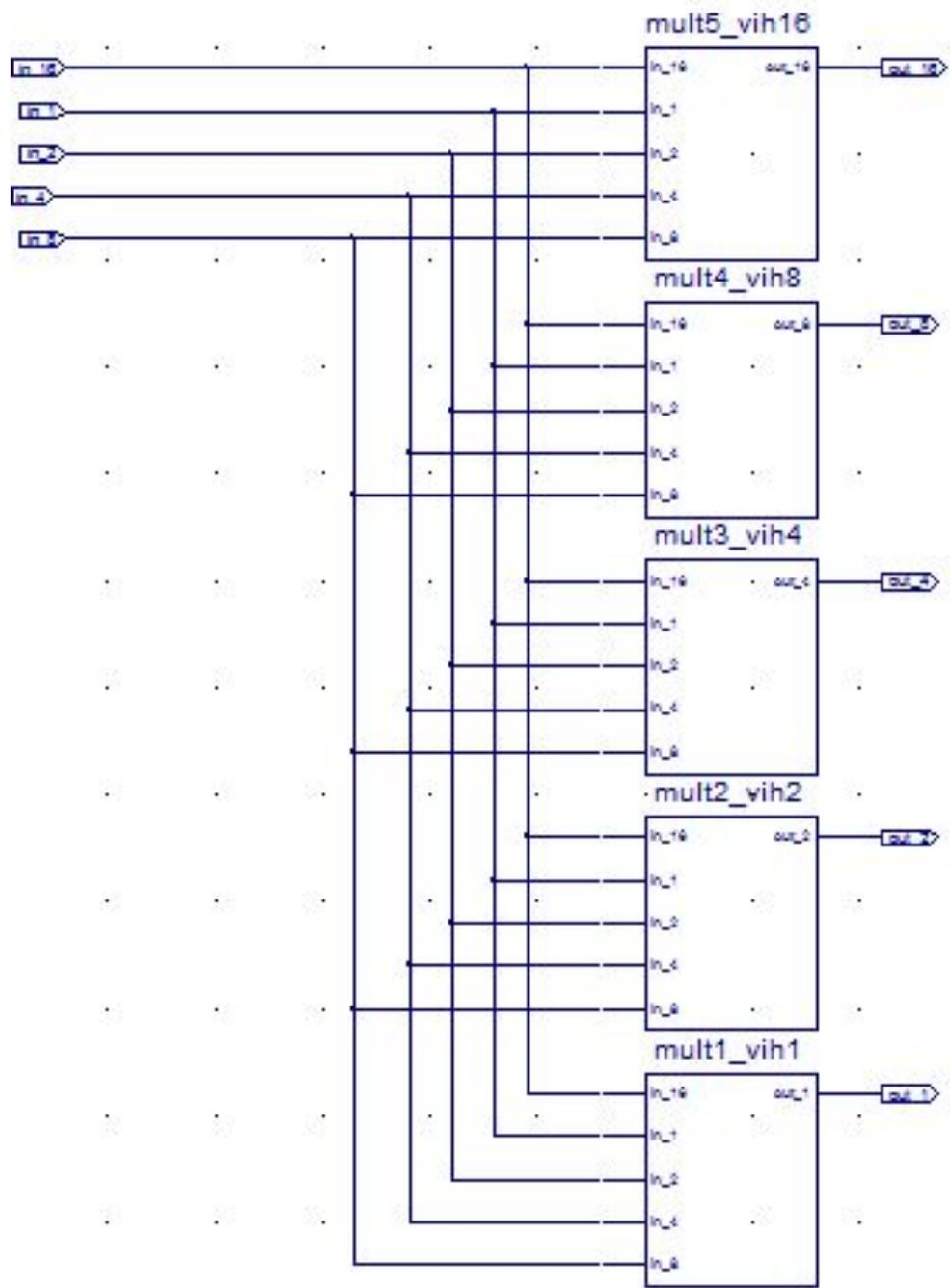
# Описание на VHDL



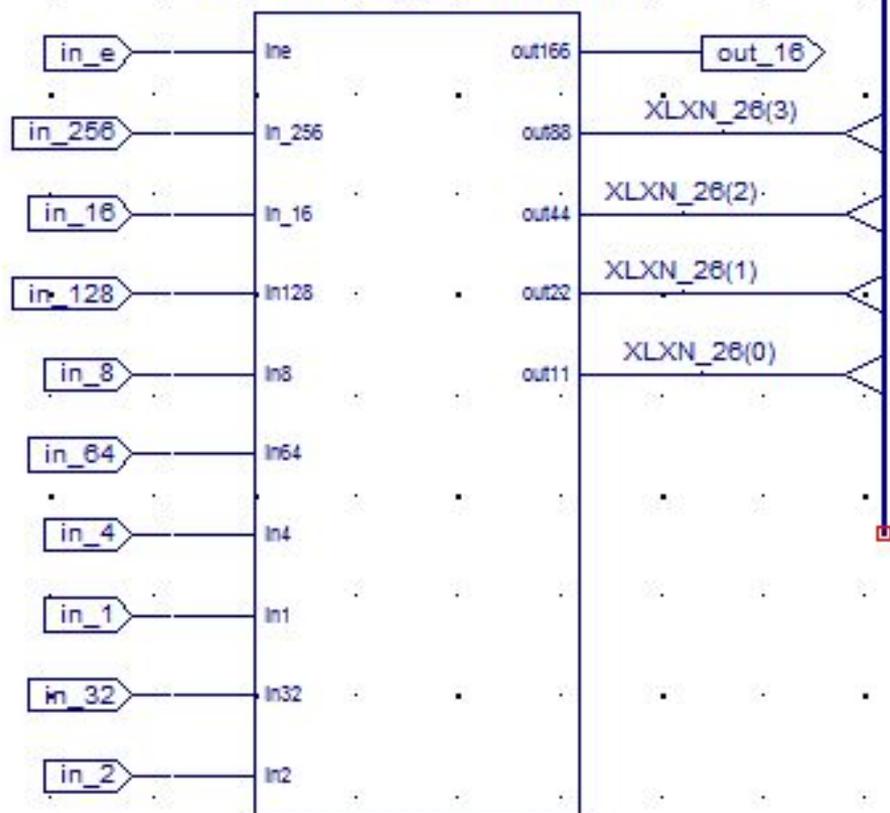
```
SUM10_VHDL.vhd
1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  -- Uncomment the following lines to use the declarations that are
7  -- provided for instantiating Xilinx primitive components.
8  --library UNISIM;
9  --use UNISIM.VComponents.all;
10
11 → entity sum10_vhdl is
12     end sum10_vhdl;
13
14     architecture Behavioral of sum10_vhdl is
15
16     begin
17
```

# Типовые ошибки в КР:

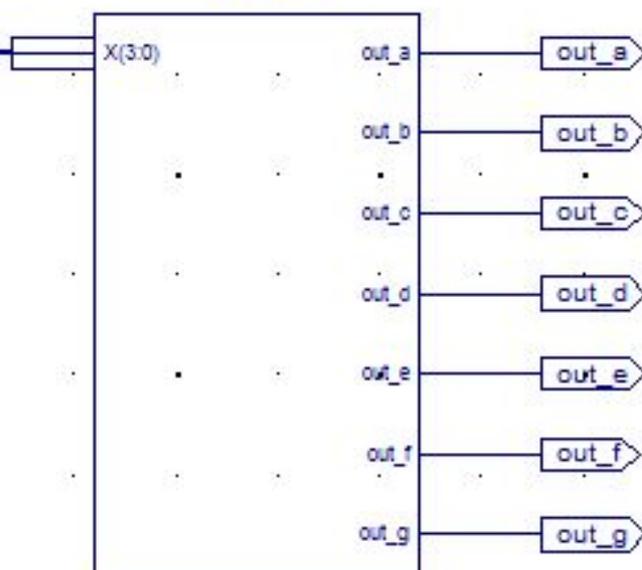
- нарушение логической структуры изложения материала в пояснительной записке КР («электронный компонент ПЛИС»);
- несоблюдение требований ГОСТ по оформлению КР;
  - «некорректные» названия блоков и схем;
- покрытие простых блоков «сложными» элементами (мультиплексорами 16/1 и т.п.);

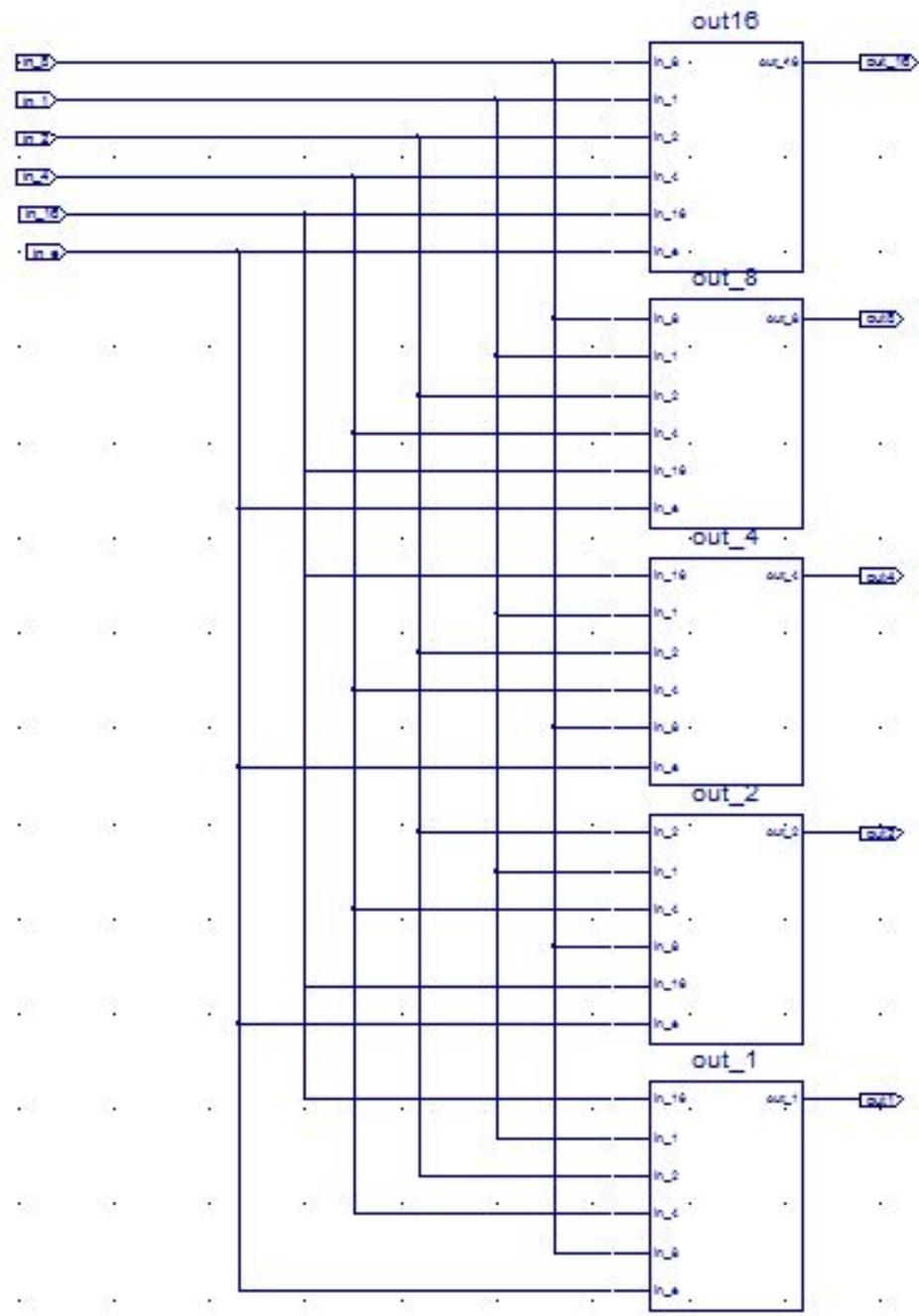


shems\_osnova

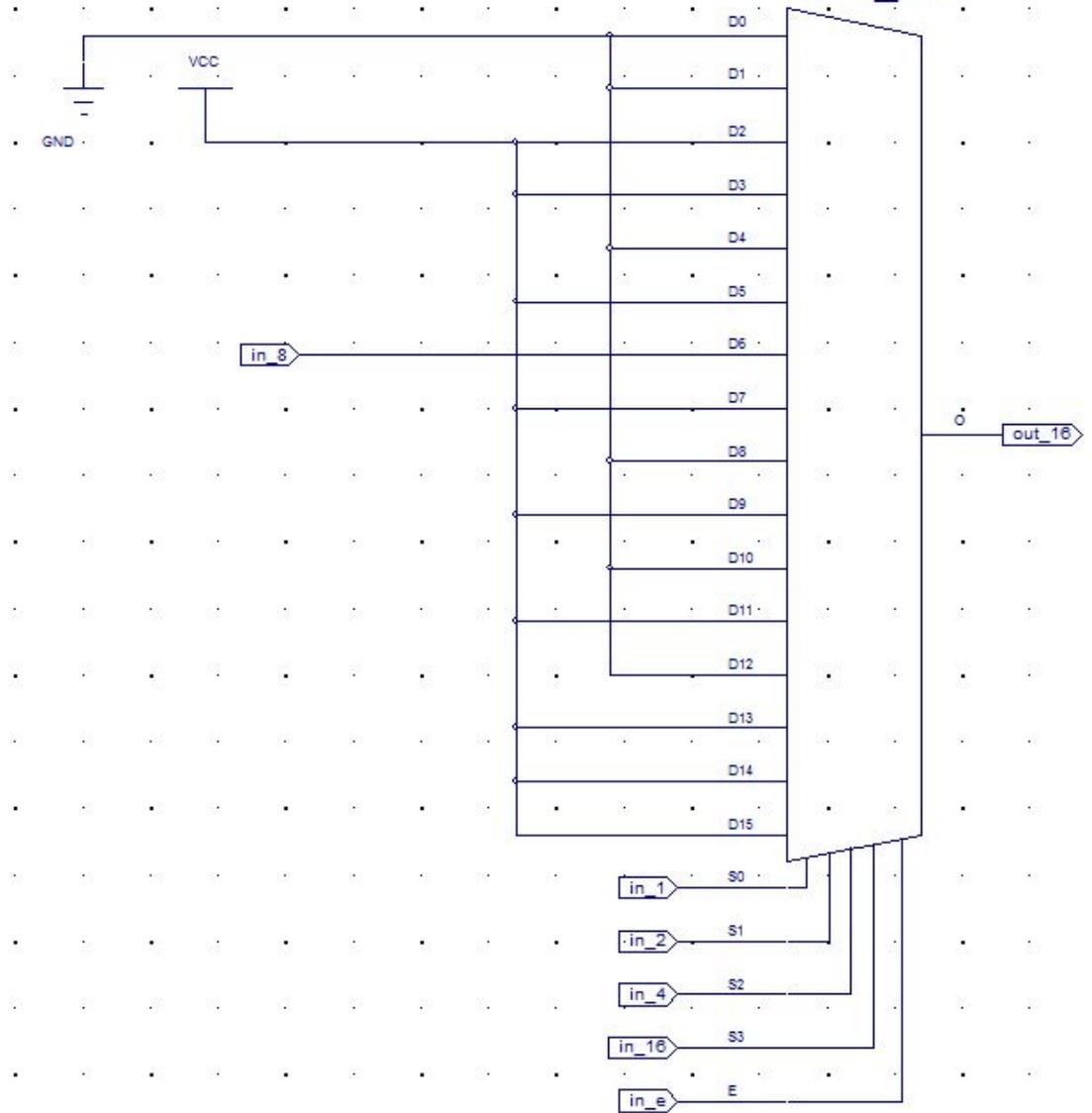


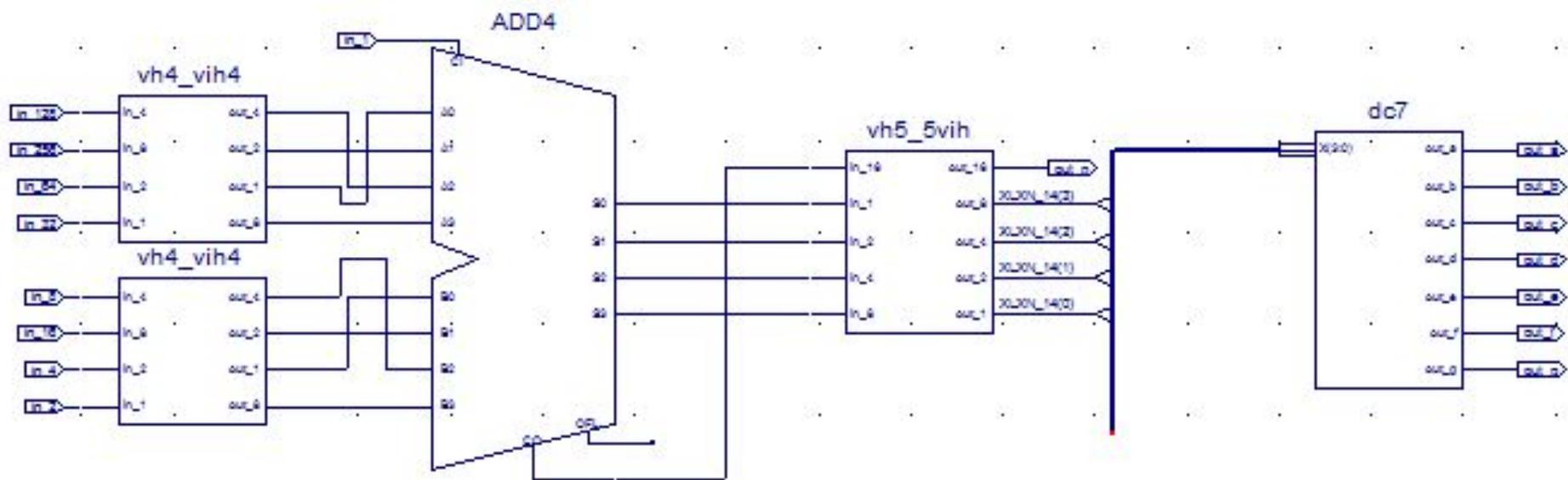
dc7

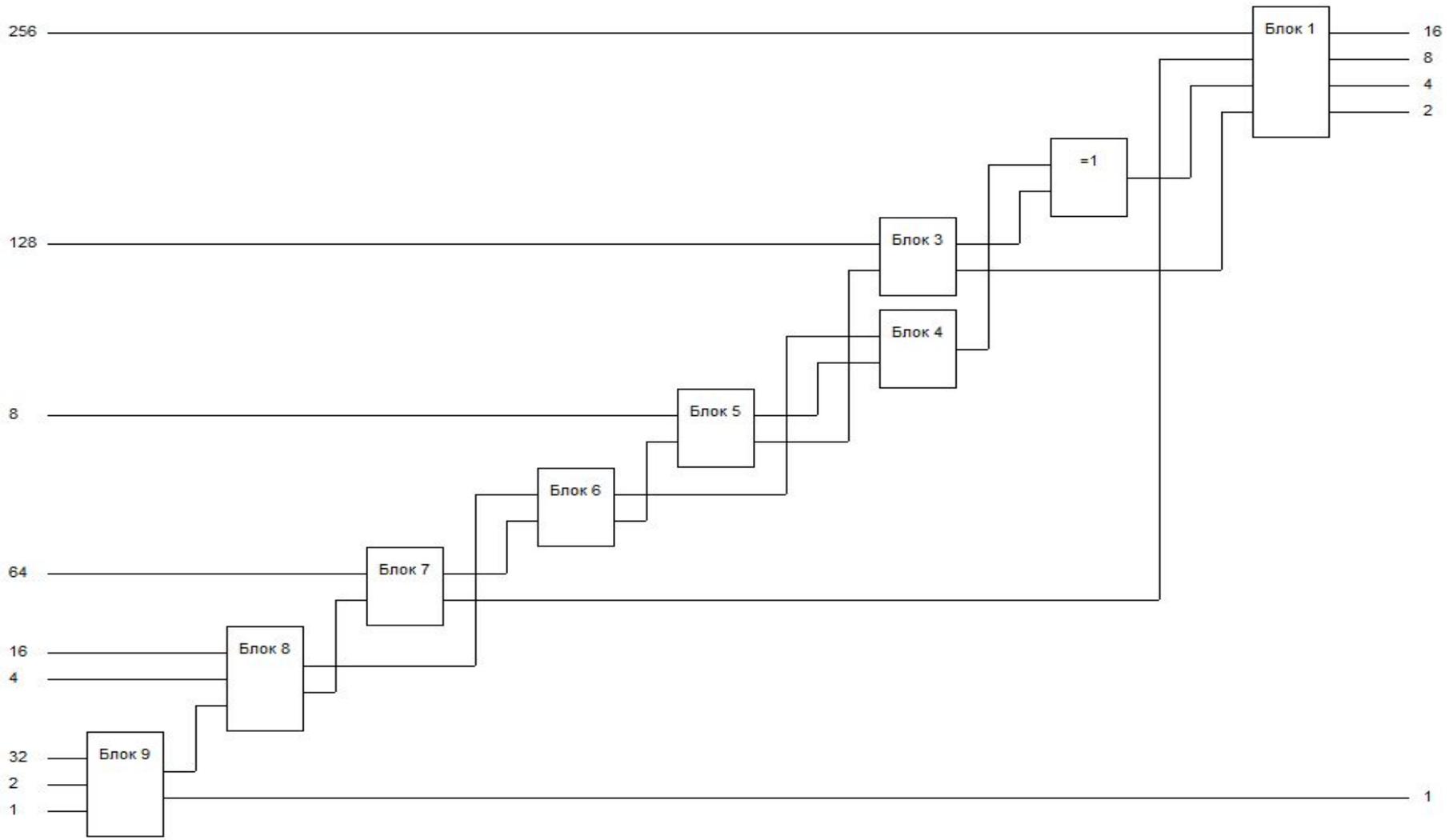


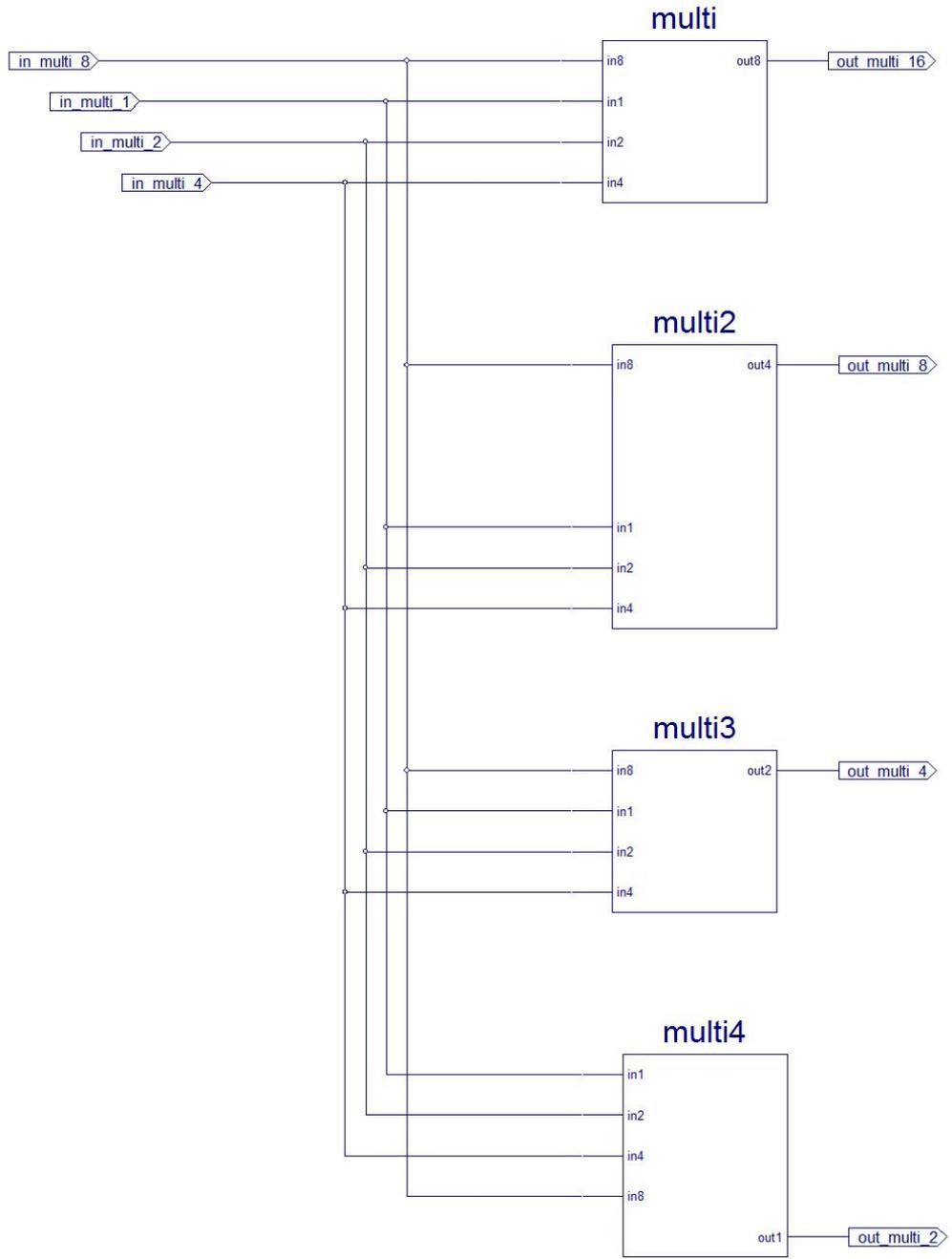


# M16\_1E

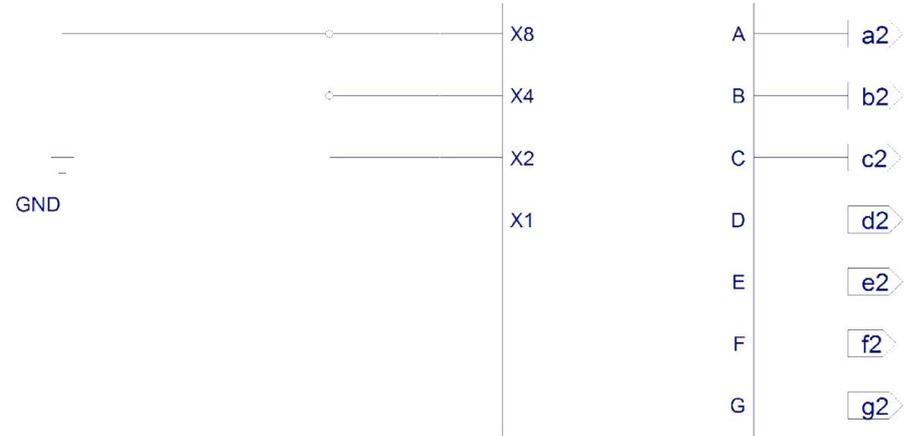




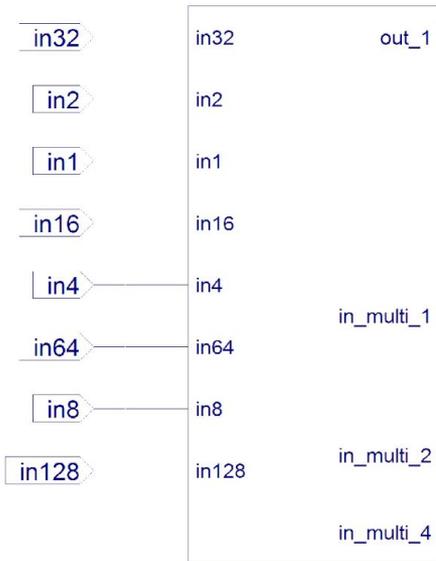




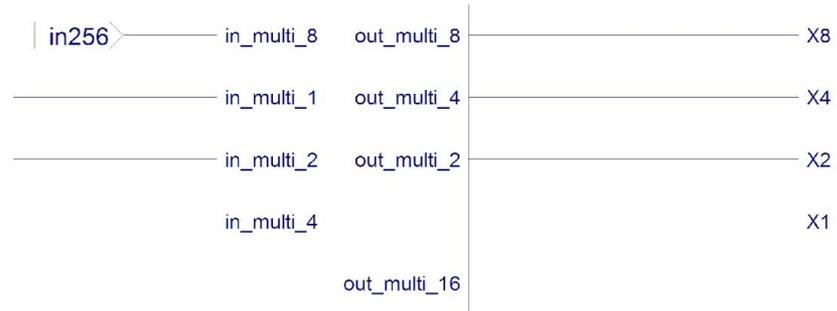
## main



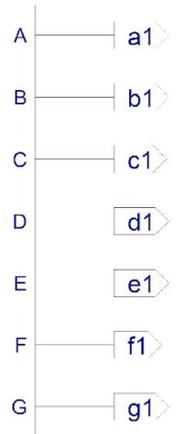
## summ

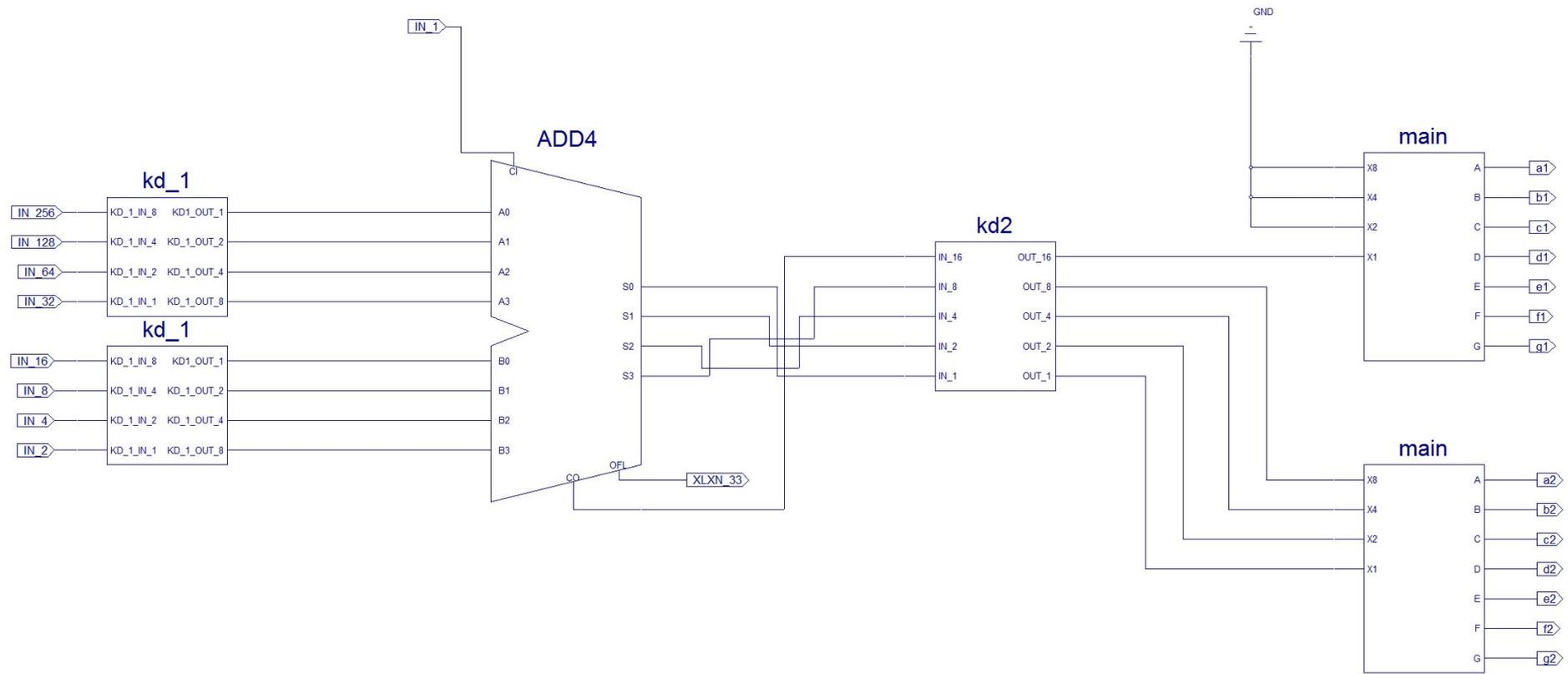


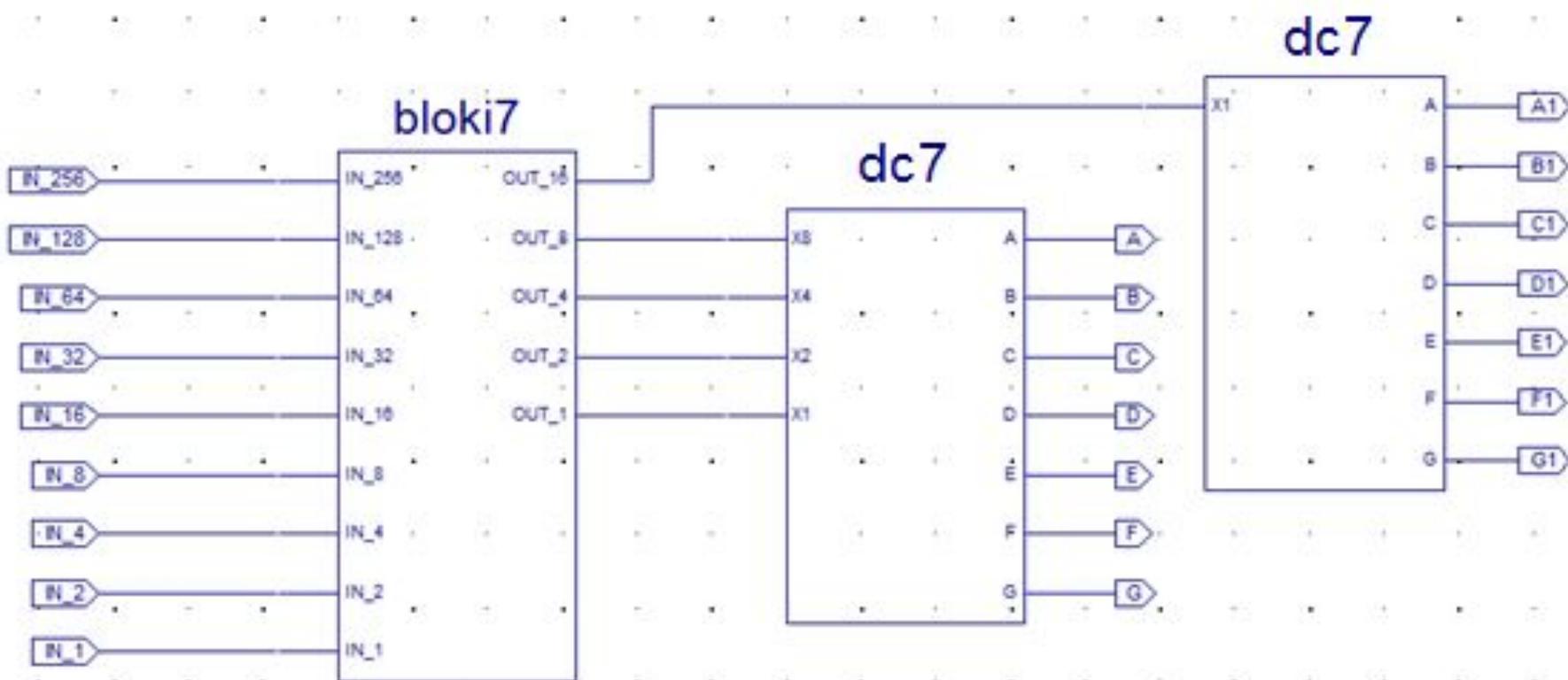
## multi\_all

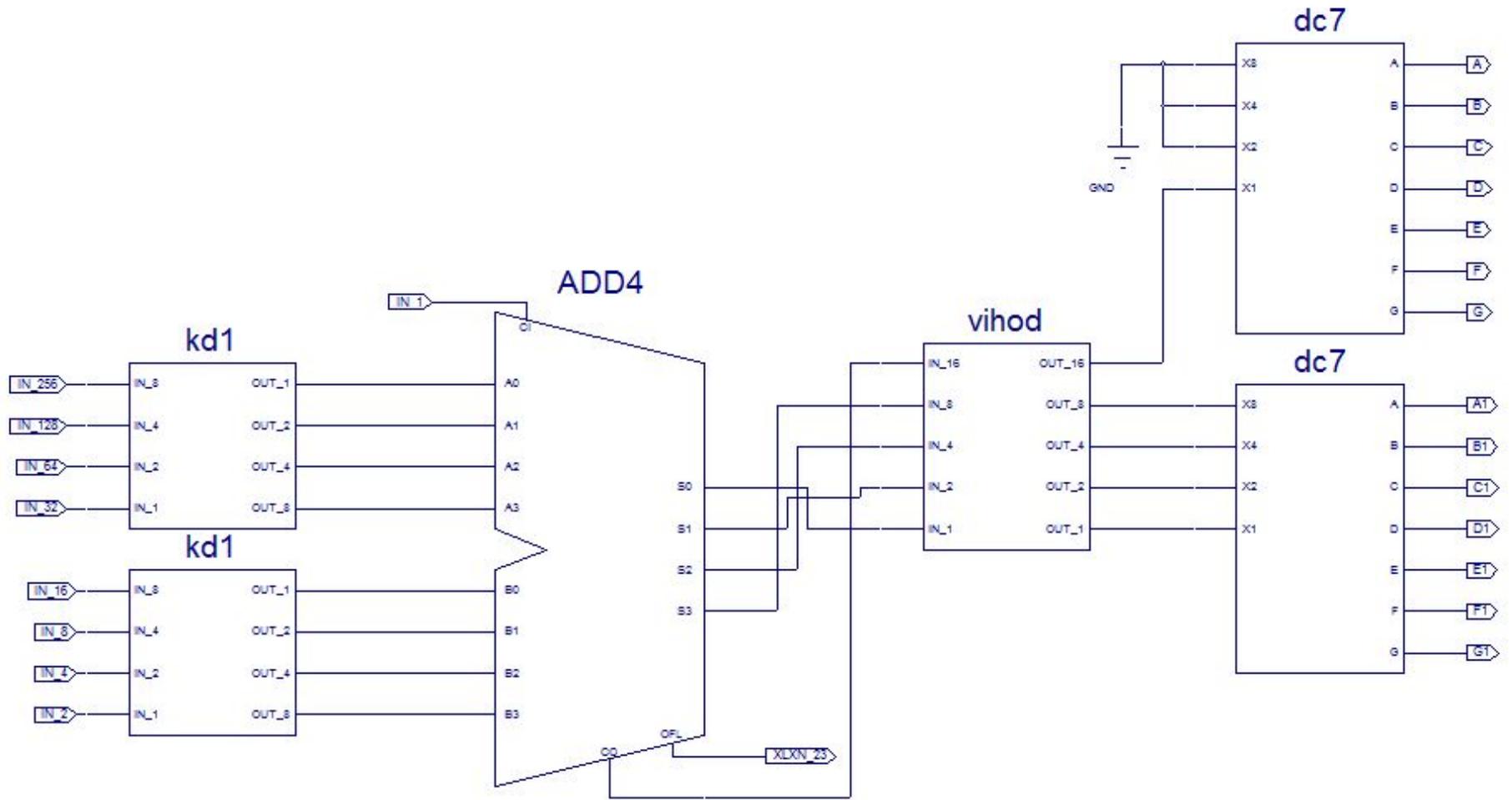


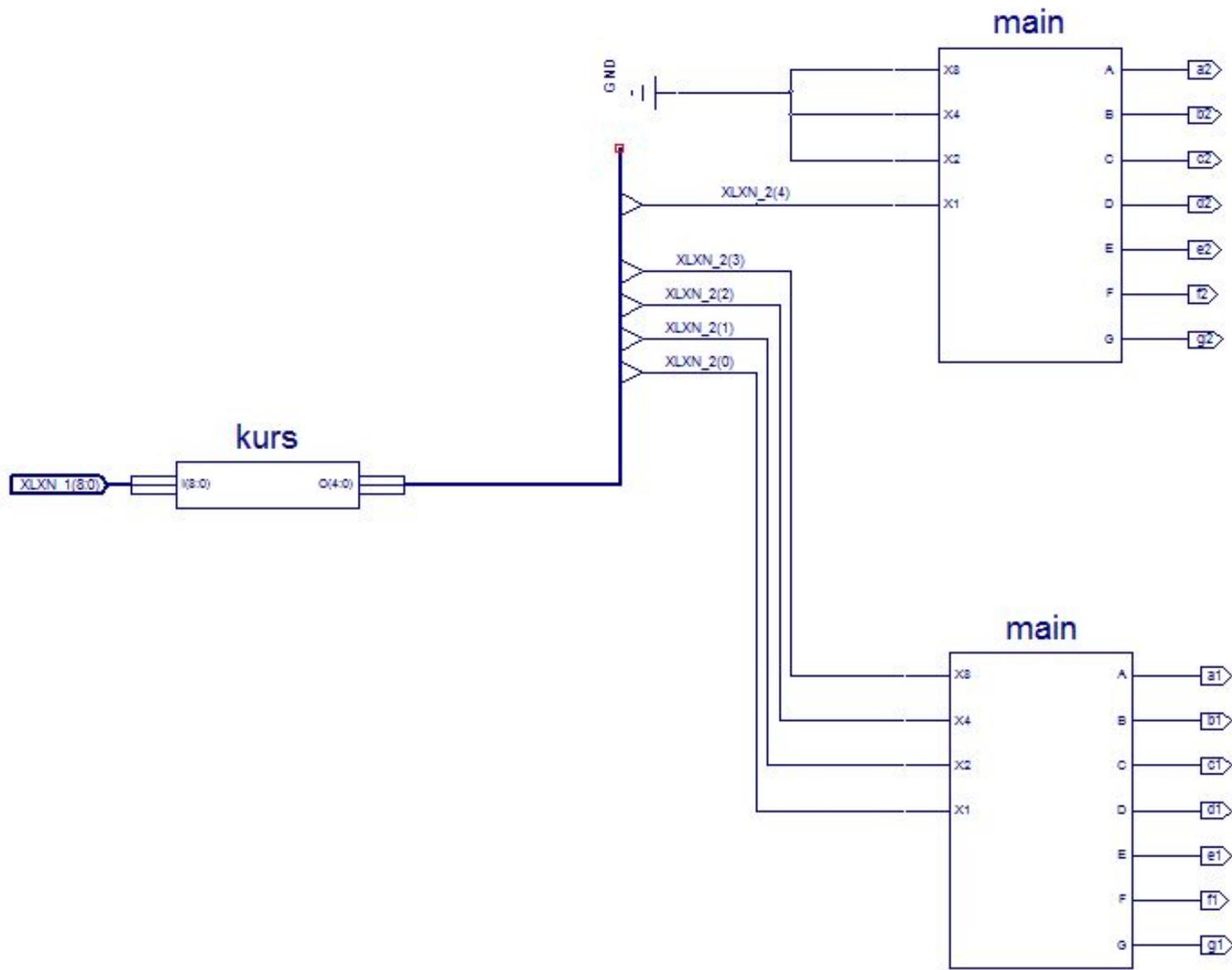
## main

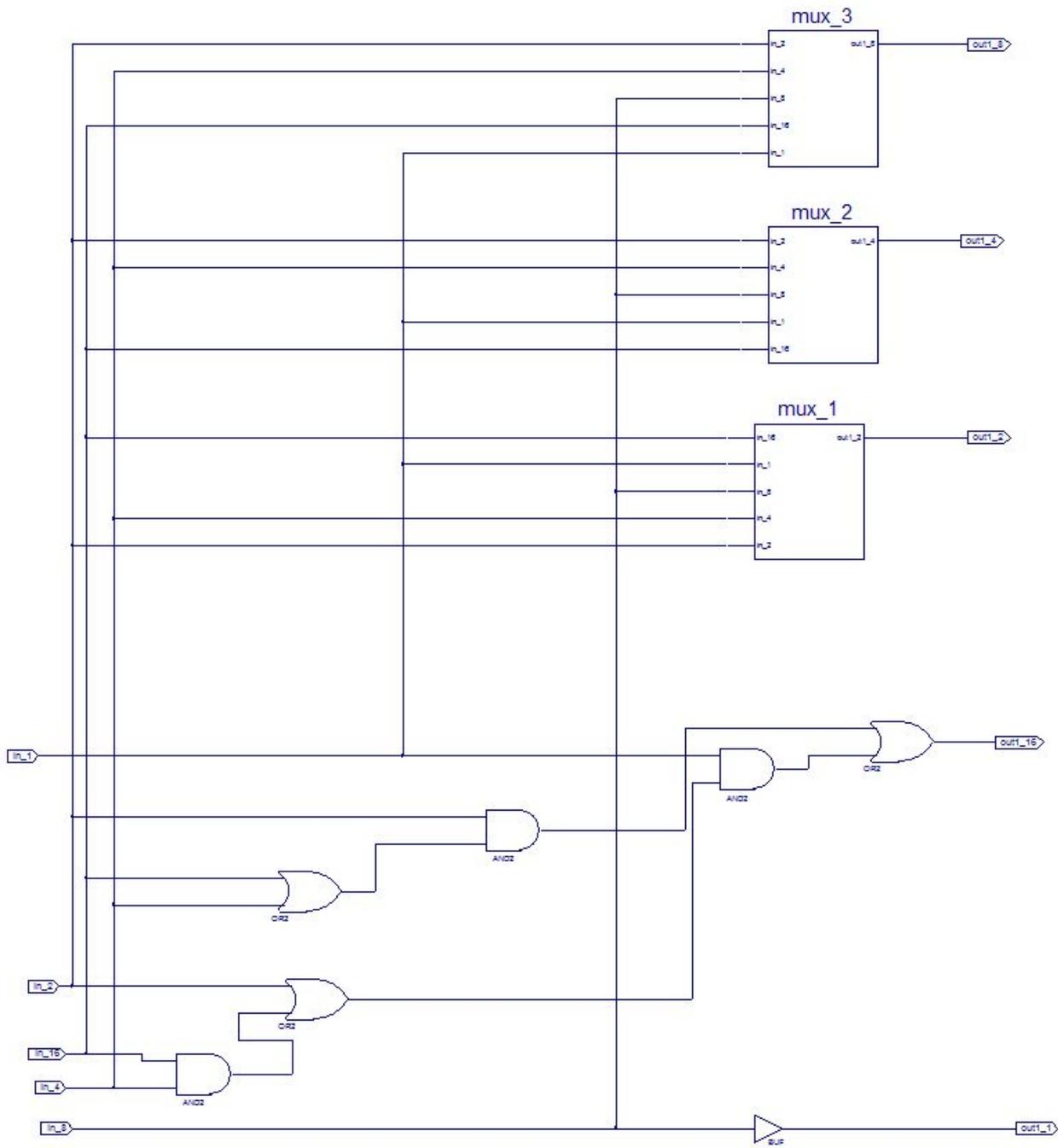


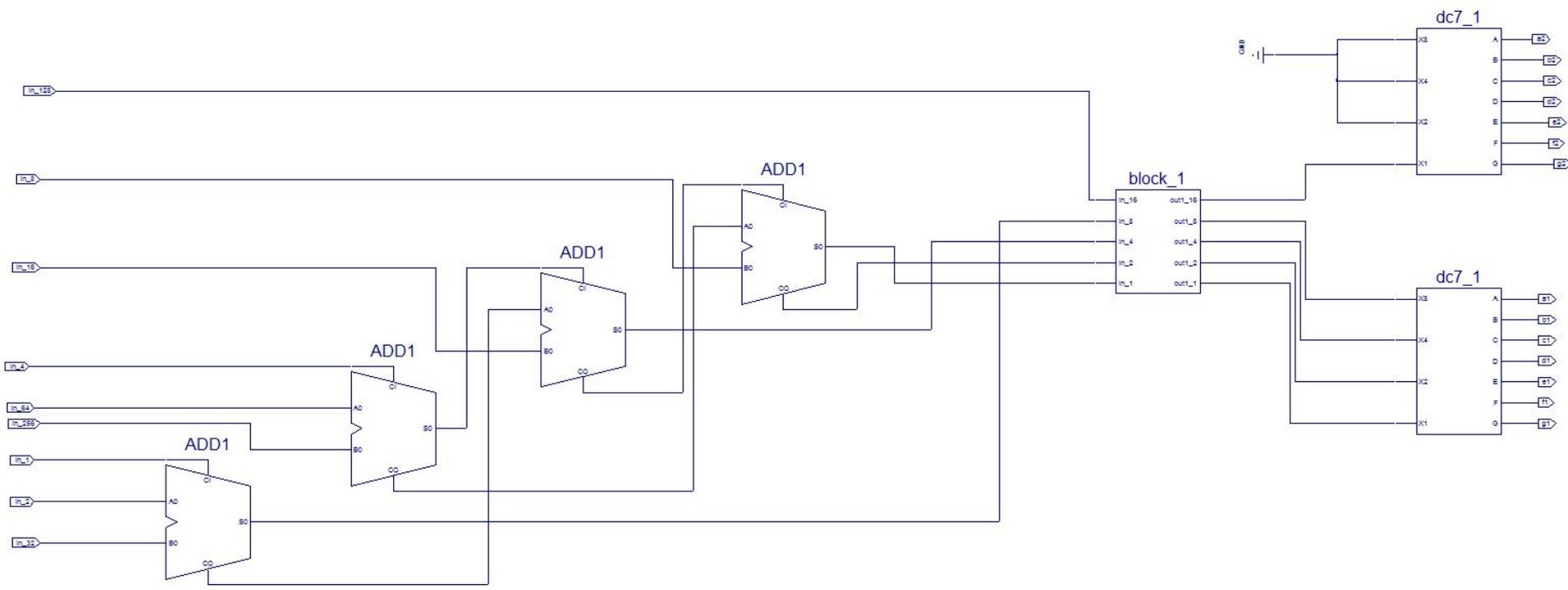




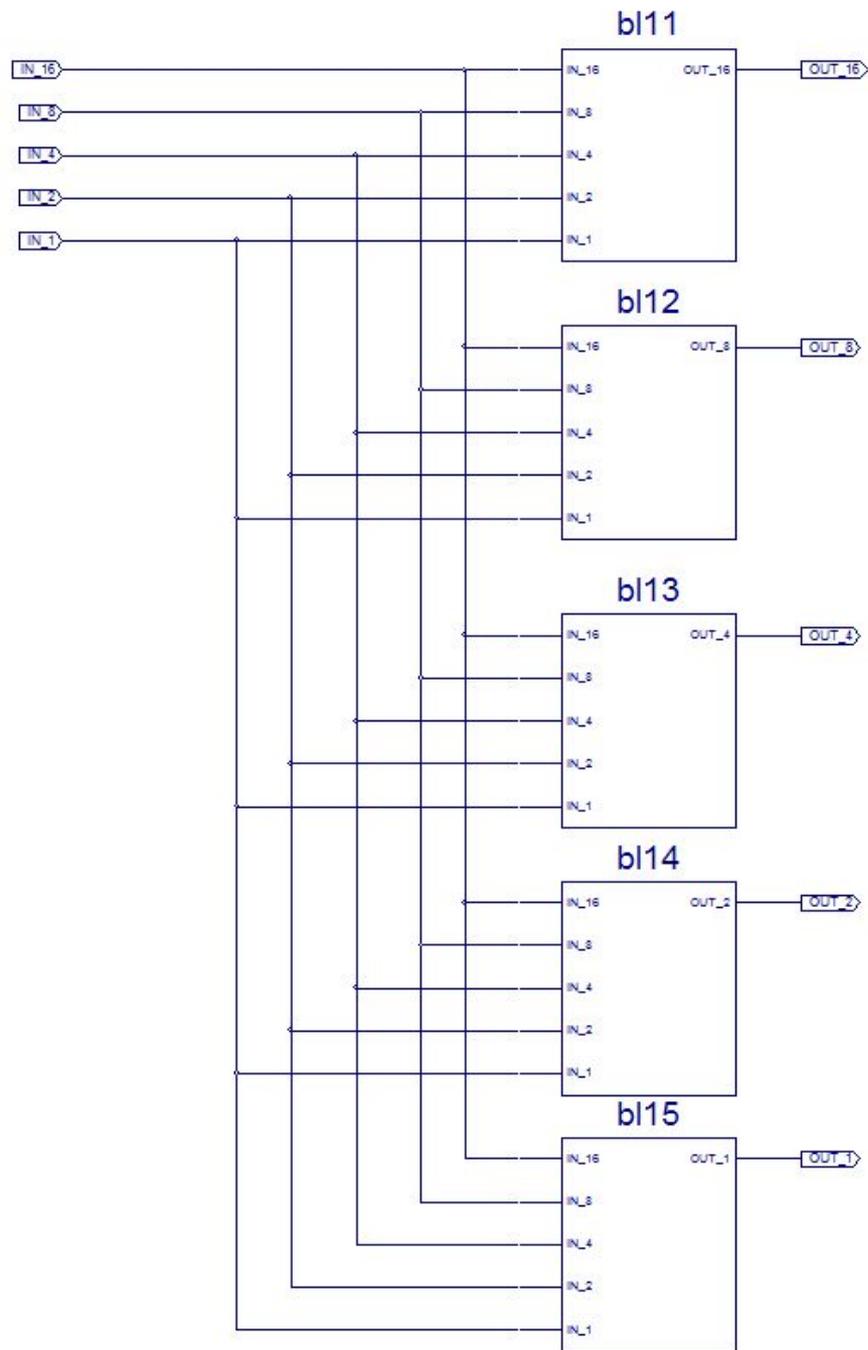


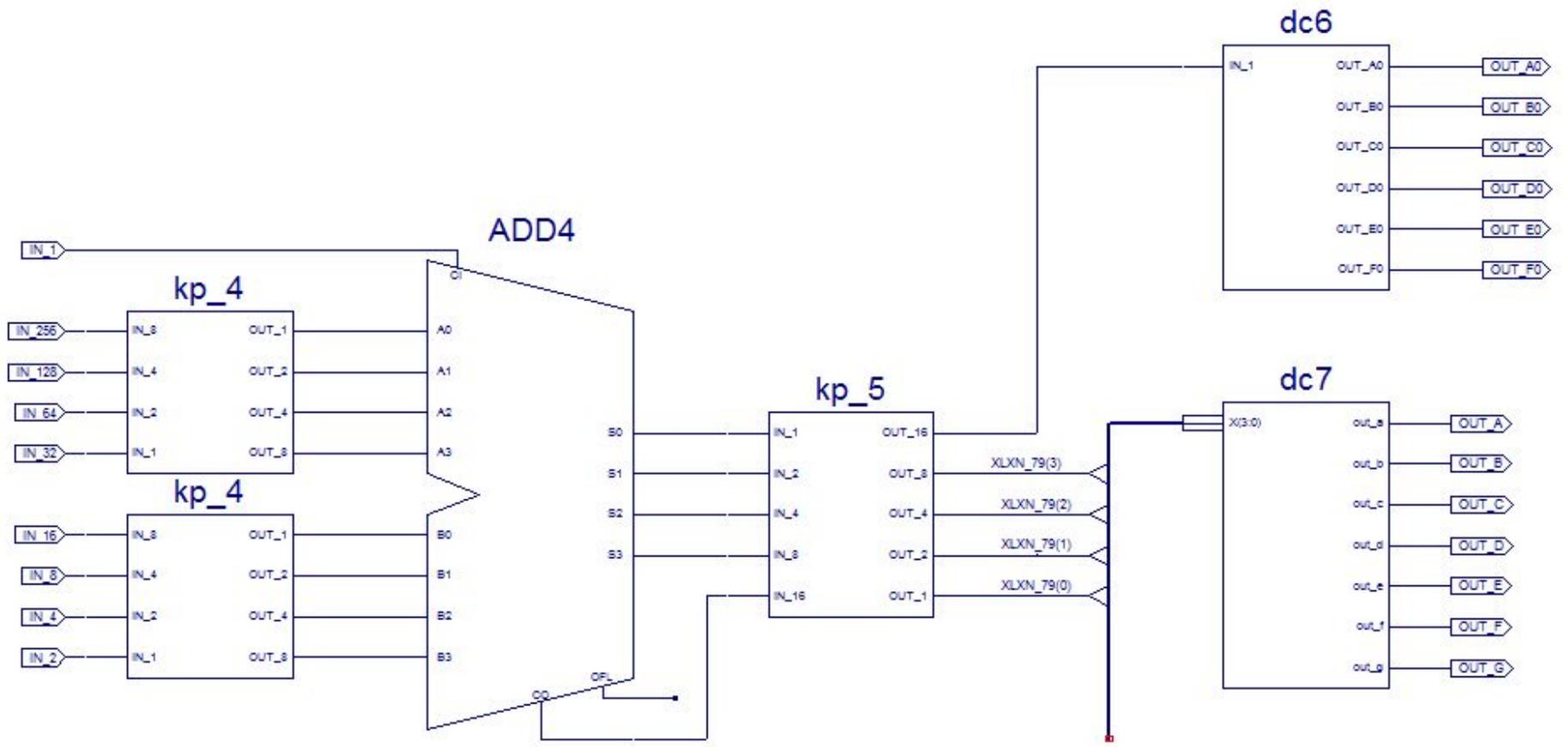






# Правильные схемы

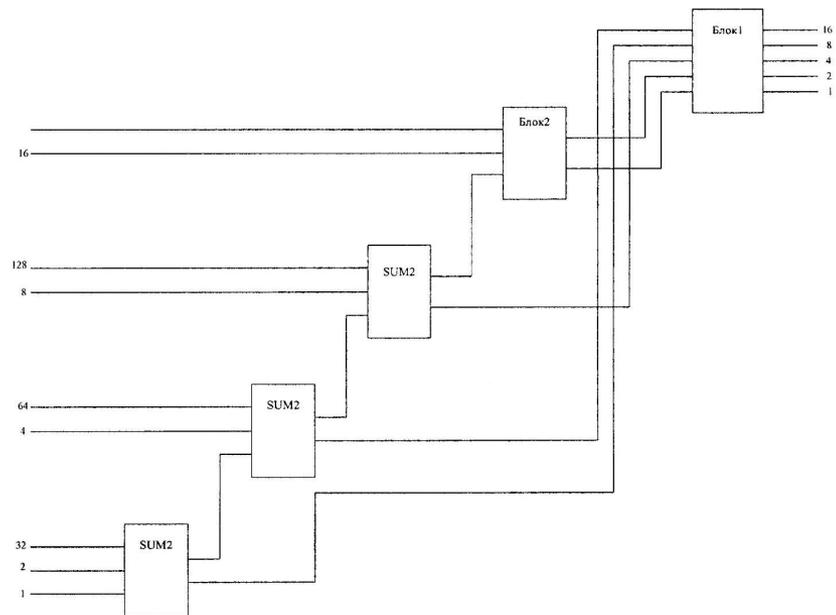




# ВАРИАНТ 7

Весовые коэффициенты: 5-4-2-1

Цифры	Код
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1011
9	1100



SUM – 0112 1223

Блок 2 – 0123 2332

Блок 1 – 000B 1B13 0412 0C17 010C 1C14 0513 10\*\*  
0210 \*\*15 0614 1116 0311 0B16 0715 1217

## **Структура расчетно-пояснительной записки курсовой работы:**

- титульный лист
- подписанное и утвержденное задание на курсовую работу;
- содержание;
- условные обозначения и сокращения;
- введение (постановка задачи);
- краткое описание методов логического синтеза, используемых при проектировании;
- построение числовой последовательности синтезируемой схемы;
- логический синтез устройства с использованием учебной САПР Decomposer;
- покрытие отдельных блоков схемы логическими элементами заданного базиса;
- описание синтезируемого устройства на языке VHDL;
- реализация синтезированных вариантов схемы на лабораторном стенде ЛСЦ-003 и оценка их работоспособности;
- сравнительный анализ затрат ресурсов ПЛИС для реализации различных вариантов схемы;
- заключение;
- список литературы;
- приложения.

## Основные этапы выполнения курсовой работы

Неделя	Объем и содержание контролируемых этапов выполнения курсовой работы	Оценка в баллах	
		мин.	макс.
04	построение числовой последовательности синтезируемой схемы, логический синтез устройства с использованием учебной САПР Decomposer	10	15
07	разработка описания синтезируемого устройства на языке VHDL	10	15
10	реализация синтезированных вариантов схемы на лабораторном стенде и оценка их работоспособности	10	15
14	сравнительный анализ затрат ресурсов ПЛИС для реализации различных вариантов схемы, оформление чертежей и расчетно-пояснительной записки	10	15
14-15	Итоговый контроль	12	30
	Оценка личностных качеств	0	10





