

Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Кафедра электронных вычислительных средств

П.Н. Бибило

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ САПР

Лабораторный практикум

для студентов специальности 40 02 02
«Электронные вычислительные средства»
дневной формы обучения

Минск 2003

УДК 681.32.02 (075.8)
ББК 32.97 я 73
Б 59

Бибилло П.Н.

Б 59 Теоретические основы САПР: Лаб. практикум для студентов спец. 40 02 02
«Электронные вычислительные средства» дневной формы обучения /
П.Н. Бибилло. – Мн. : БГУИР, 2003. – 31 с.: ил.

ISBN 985-444-512-7.

В данном лабораторном практикуме приведена лабораторная работа, посвященная изучению и практическому применению языка описания логических схем в системе автоматизированного проектирования заказных цифровых СБИС. Работа выполняется на ПЭВМ.

УДК 681.32.02 (075.8)
ББК 32.97 я 73

ISBN 985-444-512-7

© Бибилло П.Н., 2003
© БГУИР, 2003

Лабораторная работа

ПРОЕКТИРОВАНИЕ В САПР “CUSTOM LOGIC”

Цель работы – изучить язык SF иерархического функционально-структурного описания логических схем, получить практические навыки описания логических схем, научиться работе в системе “Custom Logic” автоматизированного проектирования (САПР) заказных цифровых сверхбольших интегральных схем (СБИС).

1. ЯЗЫК SF ИЕРАРХИЧЕСКОГО ФУНКЦИОНАЛЬНО-СТРУКТУРНОГО ОПИСАНИЯ ЛОГИЧЕСКИХ СХЕМ

Функционально-структурное описание логической схемы в языке SF состоит из последовательности функционально-структурных описаний подсхем, из которых состоит схема. Описание каждой подсхемы открывается общей декларативной частью; имеет тело описания; заканчивается строкой END_<Имя описания>.

Первым представляется описание верхнего уровня иерархии. Описания подсхем нижних уровней иерархии следуют далее в произвольном порядке.

В описании схемы используются присвоенные имена (далее – просто имена) и служебные (ключевые) слова. К **именам** относятся имена переменных (входных, выходных, внутренних), имена элементов, имена типов элементов. Длина имени не должна превышать 32 символа. Имена состоят из символов, выбираемых из множества {A, ..., Z, a, ..., z, 0, ..., 9, _} (строчные и прописные символы латинского алфавита, десятичные цифры и знак подчеркивания). **Ключевые слова** в тексте описания представляются заглавными буквами латинского алфавита. Каждое ключевое слово занимает только одну строку описания и может быть не привязано к ее началу. Ключевое слово не должно содержать разделители. **Разделителем** является пробел (K пробелов, K>0) либо символ "Новая строка". Символ “табуляция” не является разделителем.

Общая **декларативная часть** описания схемы включает:

TITLE <Имя описания>
FORMAT <Тип описания>
AUTHOR <Автор>
DATE <Дата>
PROJECT <Имя проекта>

Тело описания включает в себя разделы:

- А. Декларация переменных.
- Б. Декларация элементов.
- В. Описание связей.

Д. Описание функций.

Пример задания даты: 3-05-94 /* 5 марта 1994 г. */.

Обязательными являются разделы А и один из следующих: В или Д. Если обязателен раздел В, то обязателен еще и Б. Остальные разделы являются необязательными. Описание обязательного раздела должно быть полным.

А. Декларация переменных

Ключевым словом, открывающим декларацию переменных, является DCL_PIN. Оканчивается раздел декларации переменных словом END_PIN. Переменные, используемые в описании схемы, являются булевыми и соответствуют некоторым полюсам (проводникам, узлам) описываемых элементов. Переменные делятся на внешние и внутренние. EXT – ключевое слово для внешних переменных. INTER – ключевое слово для **внутренних** переменных. Внешние переменные делятся на входные, выходные и переменные типа вход/выход. INP – ключевое слово для **входных** переменных. OUT – ключевое слово для **выходных** переменных. За ключевым словом INP (OUT) с новой строки следует перечень имен переменных.

Переменные могут быть только скалярными (битовыми). В разделе INTER могут быть декларированы внутренние переменные, являющиеся выходными переменными элементов схемы.

Если переменная вводится для обозначения монтажного "ИЛИ", то в ее описании через равенство перечисляются все соединяемые через "ИЛИ" полюсы.

Б. Декларация элементов

Ключевым словом, открывающим декларацию элементов, является DCL_EL. Оканчивается раздел декларации элементов ключевым словом END_EL.

Входящие в схему элементы имеют схемное имя (именем может быть номер элемента при сквозной нумерации) и тип.

В описании схемы должны быть декларированы все элементы, входящие в схему. Декларация типов элементов начинается со строки

TYPE <имя типа>

Начиная с новой строки, через разделитель указываются схемные имена тех элементов, которые имеют данный тип. Имя типа элемента указывается в первой строке (TITLE) общей декларативной части описания элемента. Далее следуют разделы описания входных и выходных полюсов элементов данного типа. Они являются обязательными.

Раздел описания входных полюсов состоит из строки с ключевым словом INP и последовательности строк с перечислением имен (через разделитель) входных полюсов.

Раздел описания выходных полюсов состоит из строки с ключевым словом OUT и последовательности строк с перечислением имен (через разделитель) выходных полюсов.

В. Описание связей

Ключевым словом, открывающим раздел описания связей, является CONNECT. Оканчивается раздел декларации связей элементов схемы ключевым словом END_CONNECT.

Для каждого элемента схемы показывается, с какими полюсами и каких элементов связываются его входные полюсы. Для этого в отдельной строке задается имя элемента, а в следующей – перечисляются связи в форме равенств, левыми частями которых являются имена входных полюсов элемента, а правыми частями – имена выходных полюсов элементов схемы, связанных с входными полюсами данного элемента. Входные и выходные полюсы схемы в целом имеют отдельные имена. Имя полюса элемента отделяется от имени элемента точкой. Например, COUNT.y – полюс у элемента COUNT.

Для множества **выходных полюсов** образуется "псевдоэлемент" с именем OUT, имеющий только входные полюсы и не имеющий выходных полюсов. Это связано с тем, что связи элементов схемы задаются "по входам элементов".

Д. Описание функций

Ключевым словом, открывающим раздел описания функций, является FUNCTION. Оканчивается раздел ключевым словом END_FUNCTION.

Тело функционального описания снабжается ключевым словом – типом функционального описания. Допустимые типы функциональных описаний – системы ДНФ (**дизъюнктивных нормальных форм**) полностью определенных булевых функций, заданных в **матричной форме** (формат SDF) и в форме логических уравнений (формат LOG). В формате LOG описываются алгебраические **скобочные формы** систем булевых функций в базисе И, ИЛИ, НЕ. Примеры задания функциональных описаний в форматах SDF и LOG будут даны далее.

Приведем описание схемы двухразрядного сумматора, представленной на рис. 1:

```
TITLE ADDER_2          /* Имя описания - ADDER_2 */
FORMAT SF              /* Язык (формат) описания - SF */
AUTHOR Bibilo         /* Автор - Bibilo */
DATE 4-11-2000        /* Дата создания - 4 ноября 2000 */
PROJECT VLSI          /* Имя проекта - VLSI */
```

```

DCL_PIN          /* Ключевое слово начала декларации переменных */
EXT              /* Ключевое слово декларации внешних переменных */
INP              /* Ключевое слово декларации входных переменных */
a1 a2 b1 b2     /* Список имен */
OUT             /* Ключевое слово декларации выходных переменных */
s1 s2 c2        /* Список имен */
INTER           /* Ключевое слово декларации внутренних переменных */
c1=m1.z1        /* Список имен */
END_PIN         /* Ключевое слово окончания декларации переменных */
DCL_EL          /* Ключевое слово начала декларации элементов */
TYPE ADD1      /* Начало декларации типа - ADD1 */
    m1          /* Имена элементов, имеющих тип ADD1 */
INP             /* Начало декларации входных переменных элементов типа ADD1 */
x1 y1          /* Список входных переменных элементов типа ADD1 */
OUT            /* Начало декларации выходных переменных элементов типа ADD1 */
g1 z1          /* Список выходных переменных элементов типа ADD1 */
TYPE ADD2      /* Начало декларации типа - ADD2 */
    m2          /* Имена элементов, имеющих тип ADD2 */
INP             /* Начало декларации входных переменных элементов типа ADD2 */
x2 y2 w        /* Список входных переменных элементов типа ADD2 */
OUT            /* Начало декларации выходных переменных элементов типа ADD2 */
g2 z2          /* Список выходных переменных элементов типа ADD2 */
END_EL         /* Ключевое слово окончания декларации элементов */
CONNECT        /* Ключевое слово начала описания связей элементов */
m1             /* m1 – имя элемента */
    x1=b1 y1 = b2          /* Связи входов x1 и y1 элемента m1 */
m2             /* m2 – имя элемента */
    x2=a1 y2=a2 w=m1.z1   /* m1.z1 – полюс z1 элемента m1 */
OUT            /* Связи выходных полюсов схемы ADDER2 в “целом” */
    s1 =m1.g1 s2 = m2.g2 c2 = m2.z2
END_CONNECT    /* Ключевое слово окончания описания связей элементов */
END_ADDER_2    /* Ключевое слово окончания описания схемы ADDER_2 */

```

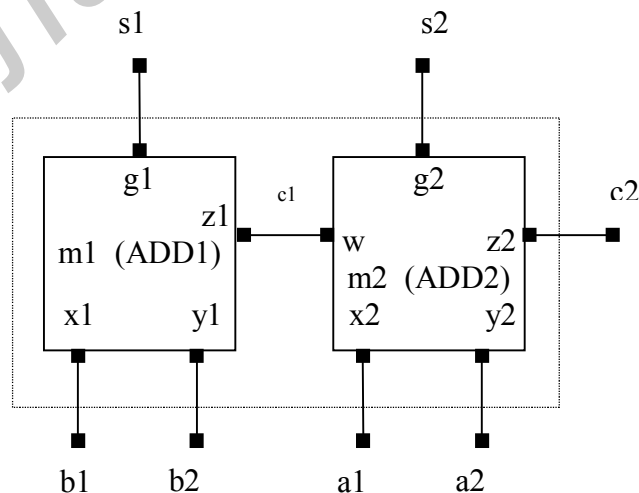


Рис. 1. Двухразрядный сумматор ADDER_2

Функциональное описание SDF (система ДНФ булевых функций в матричной форме)

Ключевым словом, открывающим тело описания SDF, является слово SDF. Оканчивается функциональное описание типа SDF ключевым словом END_SDF.

Затем в отдельной строке идут параметры:

- число входных переменных;
- число выходных переменных;
- число элементарных конъюнкций,

на которых заданы функции системы. После этого следует пара рядом расположенных матриц: **троичной матрицы**, строки которой задают элементарные конъюнкции, и **булевой матрицы**, единицы которой указывают на вхождения элементарных конъюнкций в ДНФ функций системы. Разделителей между элементами строк одной матрицы нет, разделители между матрицами есть.

Предполагается, что порядок следования переменных в списке INT совпадает с порядком следования (слева направо) входных переменных троичной матрицы, а порядок следования переменных в списке OUT совпадает с порядком следования выходных переменных булевой матрицы.

Функциональное описание полного одноразрядного сумматора **ADD2** в формате SDF имеет вид

```
TITLE ADD2
FORMAT SF
AUTHOR BIBILO
DATE 2-21-2003
PROJECT VLSI
DCL_PIN
EXT
INP
x2 y2 w
OUT
z2 g2
INTER
END_PIN
FUNCTION
SDF
3 2 7
11- 10
-11 10
1-1 10
111 01
010 01
100 01
001 01
END_SDF
END_FUNCTION
END_ADD2
```

Функциональное описание LOG (система ДНФ булевых функций в форме логических уравнений)

Ключевым словом, открывающим тело описания LOG, является слово LOG. Оканчивается функциональное описание типа LOG ключевым словом END_LOG. Затем в отдельной строке указываются параметры:

- число входных переменных;
- число выходных переменных;
- число внутренних переменных.

Каждое уравнение начинается с новой строки и заканчивается символом ";". Допускаются скобочные выражения.

В уравнениях используются следующие булевы операции:

"*" – конъюнкция;

"+" – дизъюнкция;

"^" – инверсия.

Функциональное описание (эквивалентное приведенному выше) полного одноразрядного сумматора **ADD2** на языке логических уравнений (в формате LOG) имеет вид

```
TITLE ADD2
FORMAT SF
AUTHOR BIBILO
DATE 2-21-2003
PROJECT VLSI
DCL_PIN
EXT
INP
x2 y2 w
OUT
z2 g2
INTER
END_PIN
FUNCTION
LOG
3 2 0
g2= ^x2*^y2*w + x2*^y2*^w + ^x2*y2*^w + x2*y2*w;
z2= x2*w + y2*w + x2*y2;
END_LOG
END_FUNCTION
END_ADD2
```

Логическая схема (рис. 2) в базисе И, ИЛИ, НЕ, реализующая функции одноразрядного полусумматора **ADD1**, описывается на языке SF следующим образом:


```

TITLE ADD1
FORMAT SF
AUTHOR BIBILO
DATE 2-21-2003
PROJECT VLSI
DCL_PIN
EXT
INP
x1 y1
OUT
g1 z1
INTER

END_PIN
DCL_EL
TYPE A2
e2 e1 e_z1
INP
A B
OUT
Y
TYPE O2
e_g1
INP
A B

OUT
Y
TYPE N
e4 e3
INP
A
OUT
Y
END_EL
CONNECT
e4
A=y1
e3
A=x1
e2
A=x1 B=e4.Y
e1
A=e3.Y B=y1
e_z1
A=x1 B=y1
e_g1
A=e2.Y B=e1.Y
OUT
z1=e_z1.Y g1=e_g1.Y
END_CONNECT
END_ADD1

```

Приведенное выше описание логической схемы ADD1 является **структурным** – схема (рис. 2) описывается в виде соединения логических элементов.

В иерархическом описании схемы **листовые** описания (описания логических элементов) всегда должны быть **функциональными**, т.е. представленными в формате SDF либо LOG.

Функциональное описание двухвходового **конъюнктора** – тип элемента A2:

```

TITLE A2
FORMAT SF
AUTHOR INTEGRAL
DATE 10-19-98
PROJECT
DCL_PIN
EXT
INP
A B
OUT
Y
INTER
END_PIN

```

```

FUNCTION
LOG
2 1 0
Y=A*B;
END_LOG
END_FUNCTION
END_A2

```

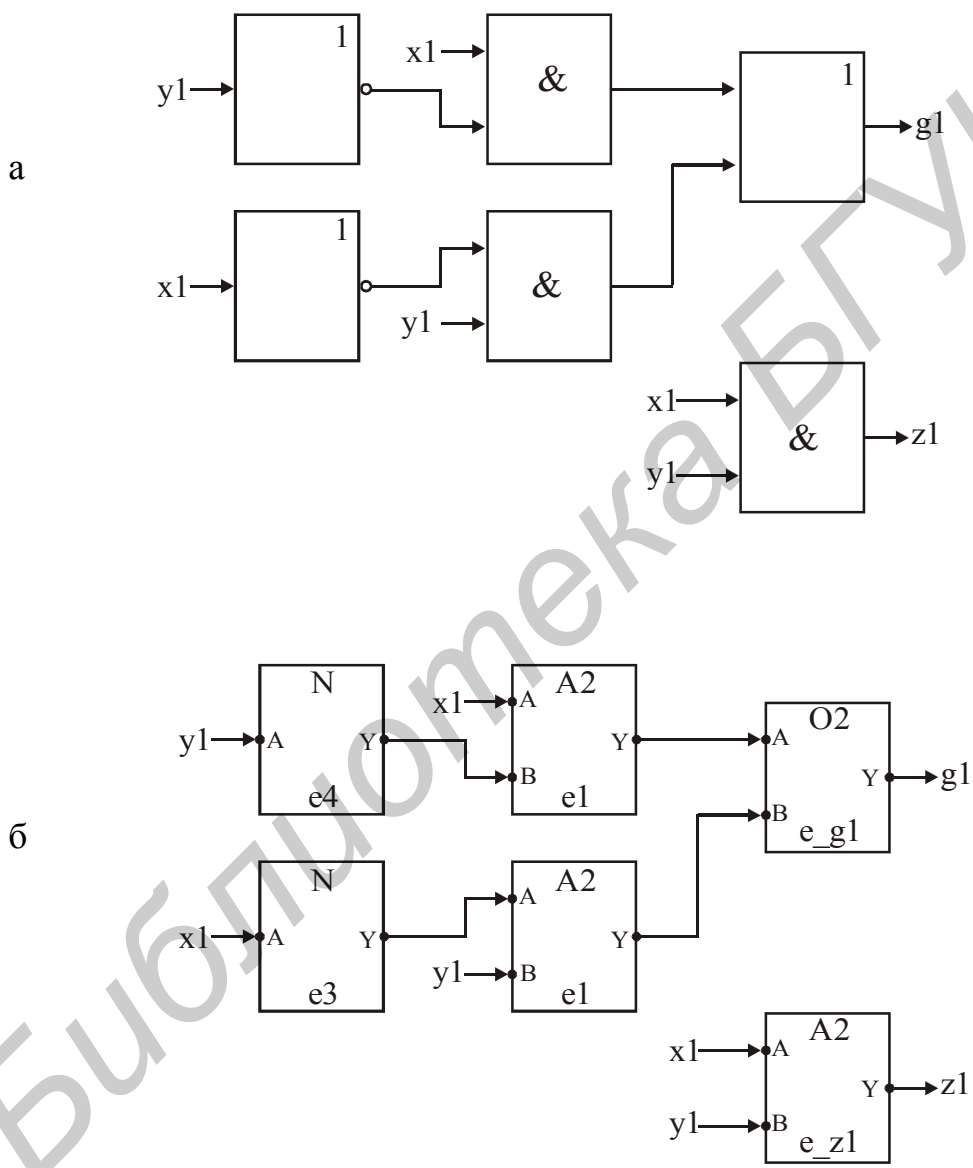


Рис. 2. Логическая схема ADD1 (а),
описание схемы с указанием типов, имен и полюсов элементов (б)

Функциональное описание двухвходового дизъюнктора – тип элемента
O2:

```
TITLE      O2
FORMAT    SF
AUTHOR    INTEGRAL
DATE      10-19-98
PROJECT
DCL_PIN
EXT
INP
A B
OUT
Y
INTER
END_PIN
FUNCTION
LOG
2 1 0
Y=A+B;
END_LOG
END_FUNCTION
END_O2
```

Функциональное описание одновходового инвертора – тип элемента N:

```
TITLE      N
FORMAT    SF
AUTHOR    INTEGRAL
DATE      10-19-98
PROJECT
DCL_PIN
EXT
INP
A
OUT
Y
INTER
END_PIN
FUNCTION
LOG
1 1 0
Y=^A;
END_LOG
END_FUNCTION
END_N
```

ВНИМАНИЕ !

Описания на языке SF представляют собой текстовые файлы с расширением .SF, например, описание схемы ADDER_2 должно содержаться в текстовом файле ADDER_2.SF.

2. РЕГУЛЯРНЫЕ МАТРИЧНЫЕ СТРУКТУРЫ ЗАКАЗНЫХ СБИС И ЯЗЫКИ ИХ ОПИСАНИЯ

Программируемая логическая матрица (ПЛМ)

ПЛМ предназначена для реализации системы ДНФ булевых функций $F(\underline{x}) = (f^1(\underline{x}), \dots, f^m(\underline{x}))$, $\underline{x} = (x_1, \dots, x_n)$ и состоит из двух последовательно соединенных подсхем, которые называются матрицами **И**, **ИЛИ**. На входы ПЛМ подаются сигналы, соответствующие входным переменным x_i , $i = 1, \dots, n$. На входы матрицы **И** подаются литералы входных переменных x_i . Литералом переменной x_i называется переменная x_i либо ее инверсия $\overline{x_i}$. На матричной подсхеме первого уровня (матрице **И**) реализуется k элементарных конъюнкций системы ДНФ, матричная подсхема второго уровня (матрица **ИЛИ**) служит для реализации дизъюнкций элементарных конъюнкций.

При СБИС-реализации (на структурном уровне описания) площадь информационных матриц ПЛМ(n, k, m), реализующей систему ДНФ m булевых функций, зависящих от n аргументов и заданных на k общих элементарных конъюнкциях, на уровне логического проектирования вычисляется по формуле

$$S_{плм} = (2n + m)k, \text{ бит.} \quad (1)$$

Пример ПЛМ изображен на рис. 3, ее площадь составляет 27 бит ($n=3, m=3, k=3$).

ПЛМ, изображенная на рис. 3, реализует следующую систему ДНФ булевых функций.

$$\begin{cases} f^1 = \overline{x} \overline{y} \overline{z} \vee xy; \\ f^2 = xy \vee xz; \\ f^3 = \overline{x} \overline{y} \overline{z} \vee xz. \end{cases} \quad (2)$$

Язык структурного описания макроэлемента ПЛМ (язык PLA)

Ключевым словом, открывающим тело описания PLA, является слово INP. Оканчивается функциональное описание типа PLA ключевым словом END_< имя описания>.

После ключевого слова INP через разделители идут имена переменных и их инверсий (литералов), подаваемых на входные шины ПЛМ. Имя инверсного литерала помечается знаком инверсии (^) и идет (через разделитель) сразу же после имени прямого литерала. После ключевого слова OUT перечисляются имена функций (выходных переменных). Затем в отдельной строке указывается

ключевое слово INVERT. В следующей строке располагается булев вектор, задающий признак формы (прямой f , инверсной \bar{f}) реализации функции.

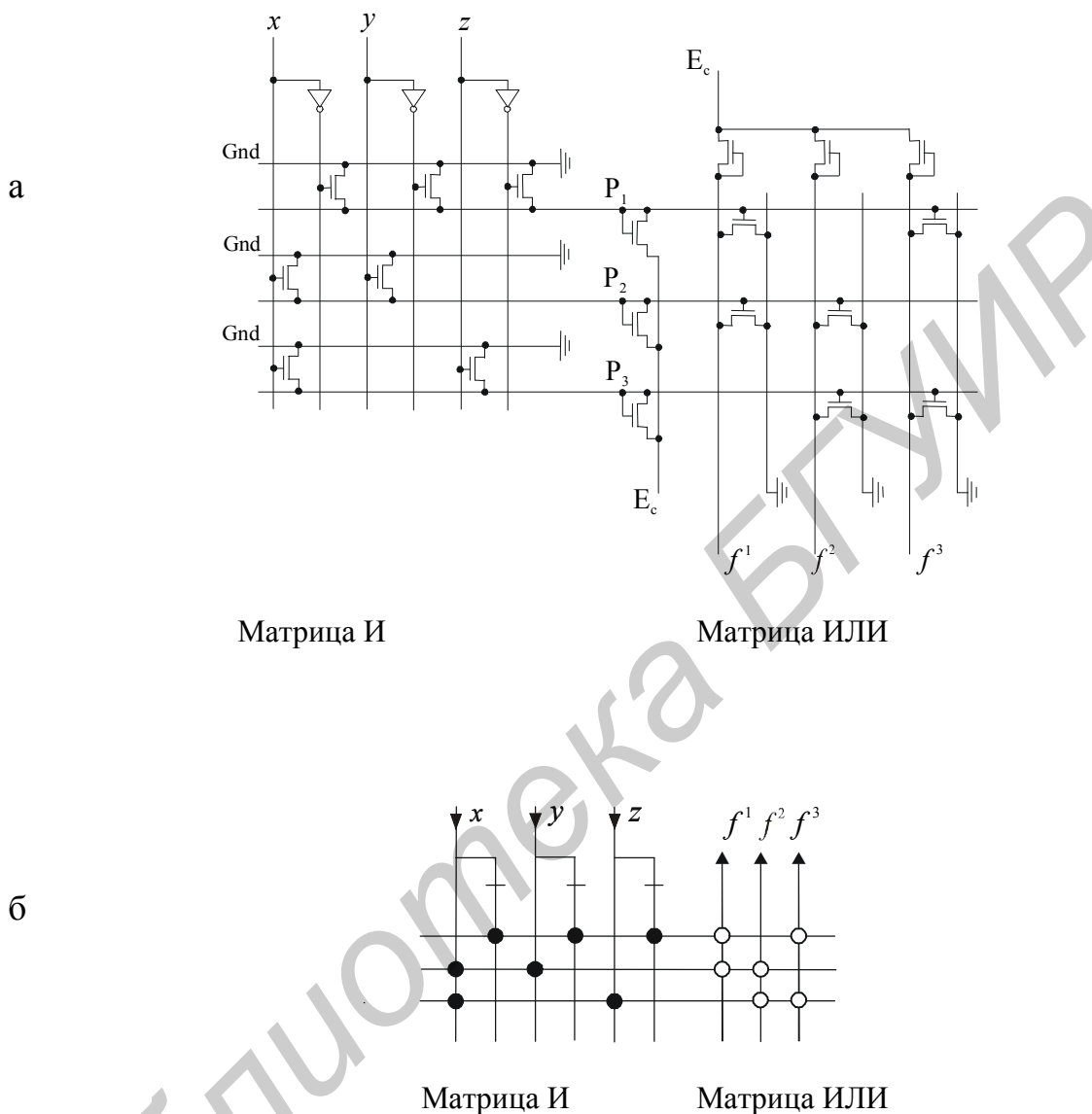


Рис. 3. Программируемая логическая матрица (а) и ее условное изображение (б)

Инверсная форма соответствующей функции отмечается единицей в булевом векторе. Разряды булева вектора идут подряд (не через разделитель).

Затем в отдельной строке идут параметры:

- число входных переменных;
- число выходных переменных;
- число промежуточных шин.

Перед строкой и после строки параметров находится разделитель в виде пяти символов "#".

Затем следует пара рядом расположенных структурных матриц: **троичной матрицы**, строки которой задают элементарные конъюнкции, реализованные на промежуточных шинах ПЛИС, и **булевой матрицы**, единицы которой указывают на вхождения элементарных конъюнкций в ДНФ функций системы, реализованных на ПЛИС. Разделителей между элементами строк одной матрицы нет, разделители между матрицами есть. Рассмотрим систему булевых функций:

$$\begin{aligned}
 f^1 = y^1 &= x_1 x_3 \vee x_1 x_2 x_4 \vee x_2 x_3 x_4; \\
 f^2 = y^2 &= \bar{x}_1 x_3 \bar{x}_4 \vee x_1 \bar{x}_3 x_4 \vee \bar{x}_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_5 \vee \\
 &\vee x_1 x_2 x_3 x_4 x_5 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 x_5; \\
 f^3 = y^3 &= x_2 \bar{x}_4; \\
 f^4 = y^4 &= x_2 \bar{x}_4 \vee \bar{x}_1 x_3 \bar{x}_4; \\
 f^5 = y^5 &= x_1 x_3 \vee \bar{x}_1 x_3 \bar{x}_4.
 \end{aligned}
 \tag{3}$$

Описание ПЛИС, реализующей систему функций (3), на языке PLA имеет вид

```

TITLE example1
FORMAT PLA
AUTHOR Bibilo
DATE 4-11-2000
PROJECT VLSI
INP
  X1 ^X1 X2 ^X2 X3 ^X3 X4 ^X4 X5 ^X5
OUT
  Y1 Y2 Y3 Y4 Y5
INVERT
00000
#####
5 5 10 /* Число входных, выходных переменных, промежуточных шин */
#####
1-1-- 10001
-1-0- 00110
0-10- 01011
1-00- 01000
001-- 01000
11-1- 10000
-111- 10000
100-0 01000
11111 01000
01011 01000
END_example1

```

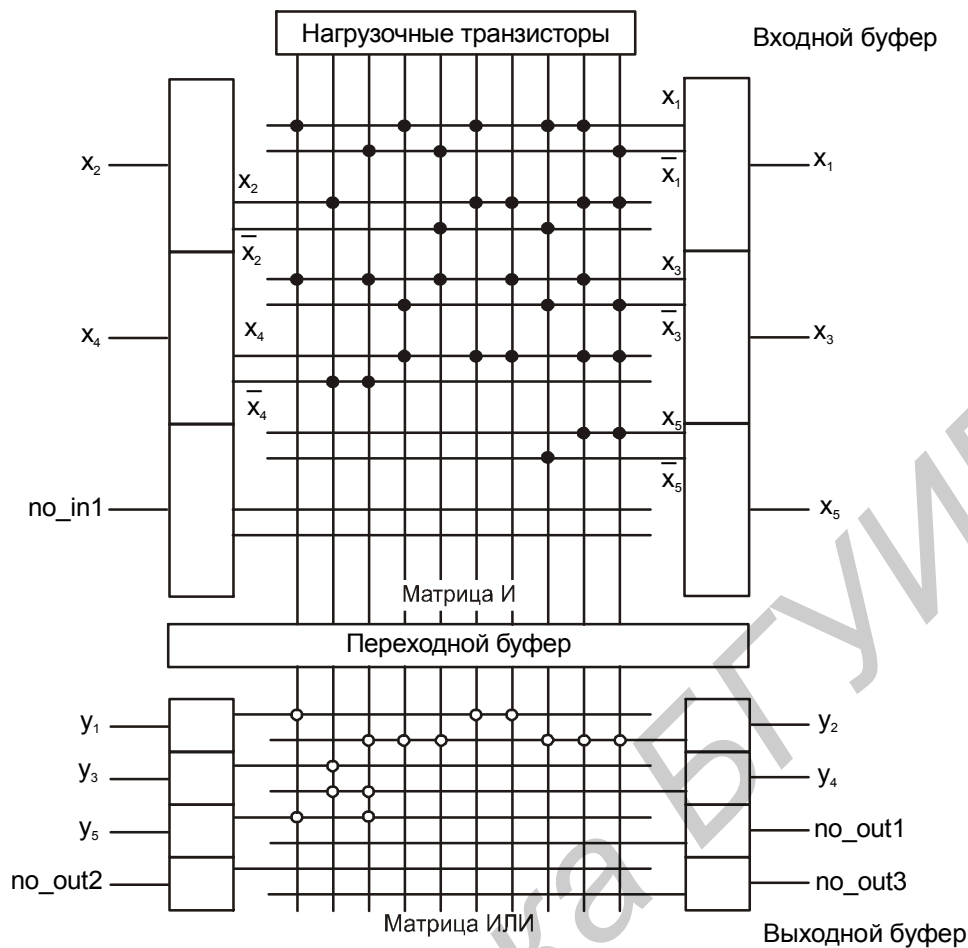


Рис. 4. Параметризованная ПЛМ

При реализации на кристалле заказной СБИС ПЛМ должна иметь законченную (параметризованную) структуру.

Параметризованные ПЛМ характеризуются тем, что входные и выходные буферы ПЛМ располагаются по обеим сторонам матриц И, ИЛИ, служащих для реализации конъюнкций и дизъюнкций (функций системы ДНФ). Строки матриц И, ИЛИ состоят из параллельно соединенных транзисторов, это требует разводки линий “земли” (Gnd) между промежуточными шинами ПЛМ: на восемь промежуточных шин выделяется одна линия “земли”. На рис. 4 изображена параметризованная ПЛМ, реализующая систему (3) ДНФ булевых функций.

ВНИМАНИЕ !

Описания на языке PLA представляют собой текстовые файлы с расширением .PLA, например, описание схемы ADDER_2 должно содержаться в текстовом файле ADDER_2.PLA.

Регулярная схема с последовательным соединением транзисторов (РМОП-схема)

Параллельное подключение транзисторов в схеме ПЛМ обеспечивает достаточное быстродействие схемы. Если же требования к задержке схемы не являются слишком жесткими и удовлетворяются электрические характеристики, то для реализации логических блоков заказных цифровых СБИС можно применять регулярные схемы на базе МОП-ячеек; МОП (металл-окисел-полупроводник) – технология изготовления СБИС.

МОП-схема регулярной структуры, предназначенная для раздельной реализации булевых функций, заданных в ДНФ, называется (сокращенно) РМОП-схемой. Пример РМОП-схемы дан на рис. 5.

В РМОП-схемах используются последовательные соединения транзисторов, что обеспечивает значительное сокращение площади кристалла, так как нет необходимости подводить линию Gnd к каждому ключевому (информационному) транзистору, что требуется в случае ПЛМ. РМОП-схема реализует булевы функции вида

$$f^j = \overline{k_{i_1}^j \vee k_{i_2}^j \vee \dots \vee k_{i_p}^j},$$

где $k_{i_q}^j$, $q = 1, \dots, p$, – элементарная конъюнкция, входящая в ДНФ функции f^j .

Пусть каждая из функций системы $F(\underline{x}) = (f^1(\underline{x}), \dots, f^m(\underline{x}))$, $\underline{x} = (x_1, \dots, x_n)$ задана своей ДНФ D_{f^j} , состоящей из c_j элементарных конъюнкций. Обозначим соответствующую РМОП-схему через $\text{РМОП}(n, c_j, m)$. На этапе структурного описания площадь $\text{РМОП}(n, c_j, m)$ подсчитывается по формуле

$$S_{\text{рмоп}} = (2n + 1) \sum_{j=1}^m c_j, \text{ бит.} \quad (4)$$

В данной формуле учитывается только сложность информационной матрицы \mathbf{I} , так как при топологической реализации столбцы для "объединения" элементарных конъюнкций обычно совмещаются в один столбец (с соответствующими разрывами).

На рис. 5 дана РМОП-схема, реализующая следующую систему ДНФ булевых функций:

$$\begin{aligned} f^1 &= x_3 \vee \overline{x_1 x_2} \vee \overline{\overline{x_2 x_3}}; & (\text{шины a, b, d}) \\ f^2 &= \overline{x_1 x_2} \vee \overline{x_1 x_2}; & (\text{шины b, c}) \\ f^3 &= \overline{x_1 x_2} \vee \overline{x_2 x_3} \vee \overline{x_1 x_2 x_3}. & (\text{шины c, d, e}) \end{aligned} \quad (5)$$

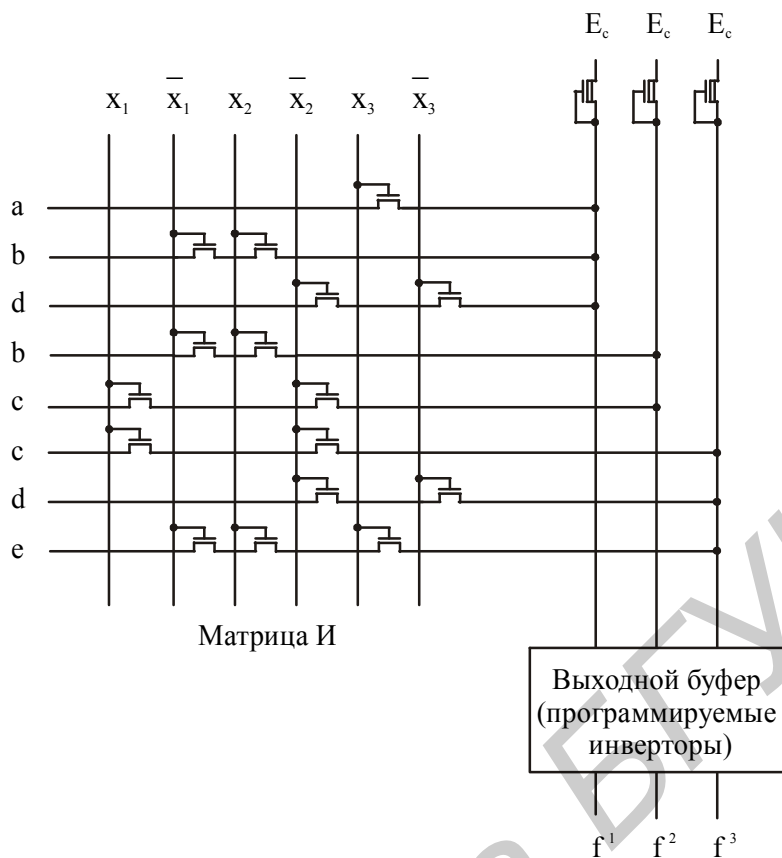


Рис. 5. PMOS-схема

Язык структурного описания PMOS-схем (язык MOS)

Тело структурного описания PMOS-схемы на языке MOS состоит из следующих разделов.

1. Перечни входных/выходных переменных.
2. Булев вектор, m компонент которого задают формы (прямую, инверсную) реализации функций.
3. Разделитель.
4. Номера шин, объединяемых для получения функций.
5. Разделитель.
6. Параметры PMOS-схемы: число входных, выходных переменных, число промежуточных шин.
7. Разделитель.
8. Структурная матрица.

```
TITLE PRIM_1
FORMAT MOS
AUTHOR BIBILO
DATE 2-21-2003
PROJECT VLSI
```

```
INP
x1 ^x1 x2 ^x2 x3 ^x3
OUT
f1 f2 f3
INVERT
000
#####
0 1 2 ;
3 ;
4 5 ;
#####
3 3 6
#####
--1
-0-
01-
-0-
-00
01-
END_PRIM_1
```

ВНИМАНИЕ !

Описания на языке MOS представляют собой текстовые файлы с расширением .MOS, например, описание схемы ADDER_2, реализованное на РМОП-схеме, должно содержаться в текстовом файле ADDER_2.MOS.

3. САПР “CUSTOM LOGIC”

САПР CL (Custom Logic) предназначена для проведения сквозного проектирования устройств управления заказных цифровых СБИС, выполненных на основе МОП-технологии. Устройства управления реализуются на базе матричных структур. Система обеспечивает возможность проектирования устройств управления на основе макроэлементов, результирующая (послойная) топология которых может быть получена автоматически.

Работа в системе CL состоит из трех этапов:

1. Настройка системы.
2. Создание проекта.
3. Проектирование (работа с проектом).

Этап 1. Настройка системы

В режиме настройки окно программы CL выглядит так, как это показано на рис.1. Под заголовком окна находится главное меню системы. Нижняя строка окна содержит информационную панель.

Вызов элемента главного меню приводит к автоматическому раскрытию соответствующего пункта. Раскрытое меню «Проект» содержит функции обслуживания проекта (рис. 6).

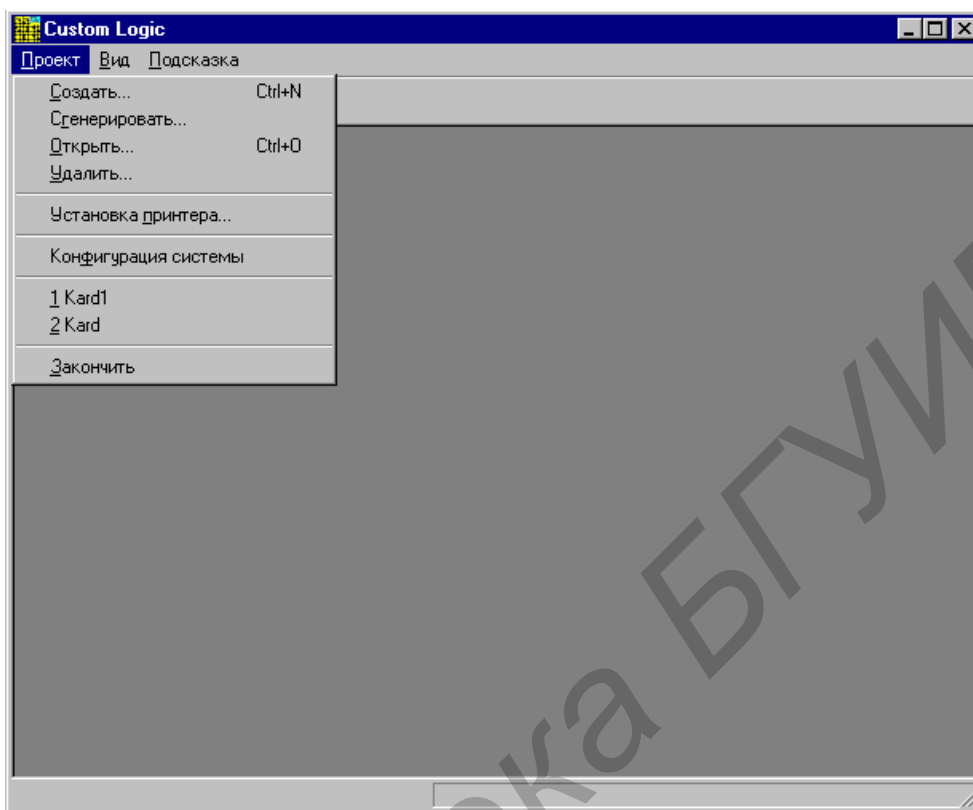


Рис. 6. Основное окно системы CL в режиме настройки

Функция "Создать..." приводит к активизации диалоговой панели для создания нового проекта, которая приведена на рис. 7.

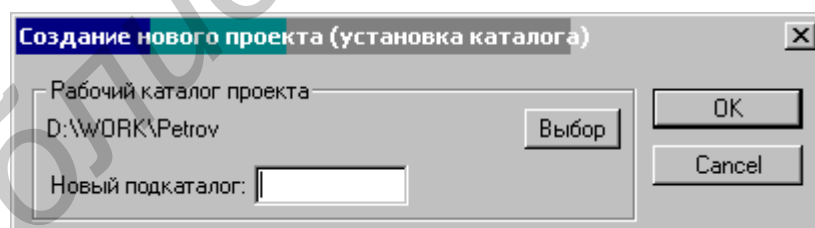


Рис. 7. Диалоговая панель создания нового проекта

Функция "Сгенерировать..." приводит к активизации диалоговой панели для задания параметров для генерируемого проекта – псевдослучайной системы булевых функций, параметры которой должен указать пользователь.

Функция "Открыть..." приводит к активизации стандартной диалоговой панели для открытия существующего файла проекта с расширением .cl. Эта диалоговая панель позволяет выбрать проект для последующей обработки.

Функция "Конфигурация системы..." приводит к активизации диалоговой панели для установки файлов системных параметров, каталогов библиотек. Эта диалоговая панель рассмотрена на рис. 8.

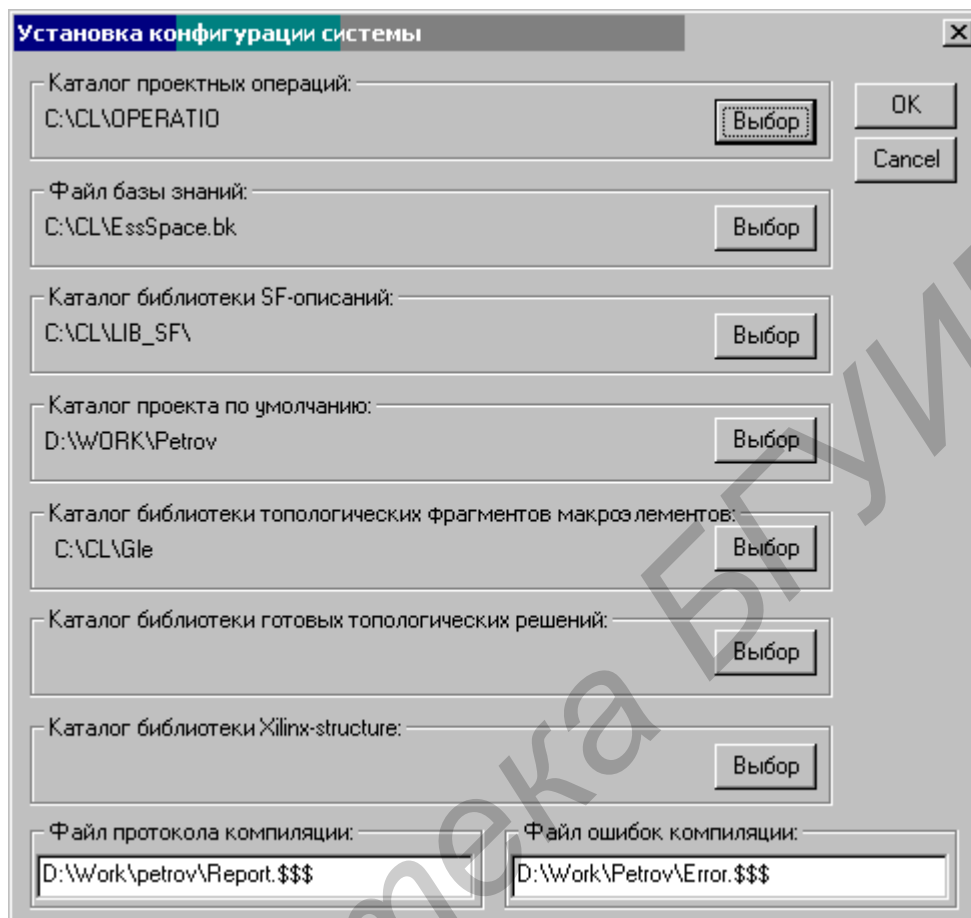


Рис. 8. Диалоговая панель «Конфигурация системы» для настройки системы CL

ВНИМАНИЕ !

Для настройки системы пользователь должен указать директорий — папку, в которой располагается "Каталог проекта по умолчанию", а также папку, в которую должны записываться "Файл протокола компиляции" и "Файл ошибок компиляции".

На рис. 8 в качестве каталога проекта выбрана папка **D:\Work\Petrov**.

Этап 2. Создание проекта

Рассмотрим действия, которые необходимо выполнить для создания проекта. Будем создавать проект из SF-описаний схемы ADDER_2. В целом, SF-описания данной схемы состоят из следующих текстовых файлов:

- ADDER_2.sf (головное описание)
- ADD1.sf
 - A2.sf
 - O2.sf
 - N.sf
- ADD2.sf

Пусть эти файлы располагаются в папке D:\WORK\Petrov.

Выберем в меню «Проект» раздел «Создать» (см. рис. 7).

По нажатии кнопки "OK" на экран выводится диалоговая панель с названием "Редактирование SF-описания проекта". На этой панели необходимо выбрать кнопку «Добавить» (рис. 9), по нажатии которой открывается новая диалоговая панель с названием "Добавление в проект SF-описаний" (рис.10).

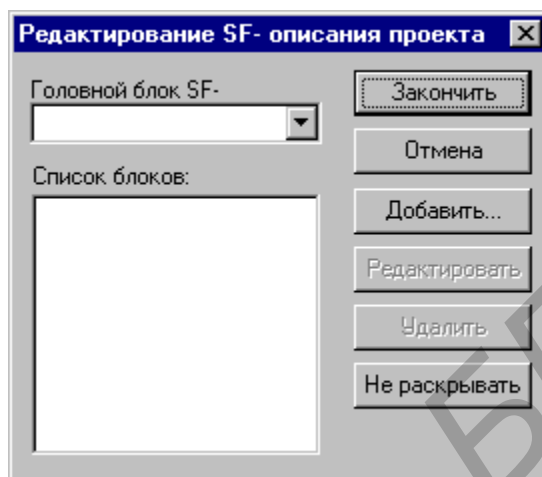


Рис. 9. Диалоговая панель формирования проекта

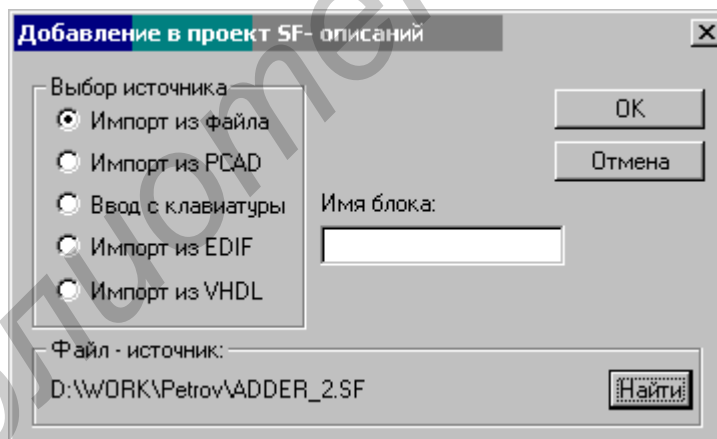


Рис. 10. Диалоговая панель добавления блоков в проект

Найдем в папке **D:\WORK\Petrov** файл с именем **ADDER_2.sf** и включим, нажав кнопку “OK”, SF-описание **ADDER_2.sf** (см. рис.10).

Аналогичным образом (“Добавить”, “Найти”, “OK”) включим в проект описания **ADD1.sf**, **A2.sf**, **O2.sf**, **N.sf**, **ADD2.sf**.

Имена включаемых блоков размещаются в списке (рис. 11), и из них выбирается имя **ADDER_2 головного** блока проекта.

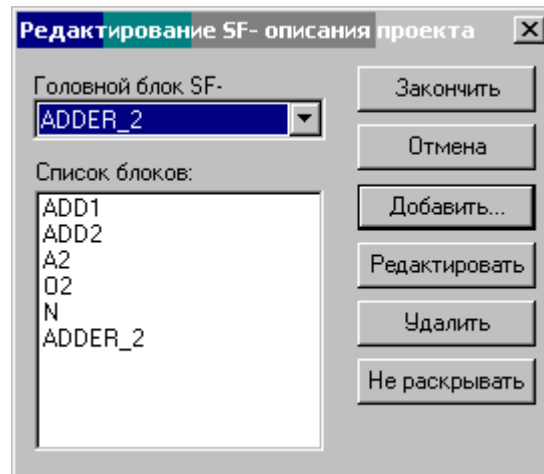


Рис. 11. Итоговое состояние панели формирования проекта

После указания головного блока созданный проект сохраняется в библиотеке проекта (рис. 12). После сохранения проекта можно выполнять этап 3 (проектирование).



Рис. 12. Сохранение созданного проекта

ВНИМАНИЕ !

Проект сохраняется в файле с расширением .cl.

Этап 3. Проектирование

Основное рабочее окно системы CL после сохранения проекта представлено на рис. 13. Всего в окне видно пять панелей, расположенных в два столбца, – три панели в левом столбце и две панели в правом. В верхней панели левого столбца показывается **иерархия** подсхем проекта. Данная панель является **активной** в том смысле, что переключение активного элемента в ней порождает изменение информации и во всех других панелях. Непосредственно под ней располагается вспомогательная панель, отражающая полученную при компиляции суммарную площадь макроэлементов, подчиненных узлу схемы проекта (см. рис. 13). В нижней панели левого столбца отражаются некоторые атрибуты активного блока проекта. На панелях правого столбца представляется информация о проектируемой схеме: на верхней панели – описательная часть, на нижней – функциональное описание или информация о связях активного блока (подсхемы). Панель описательной части схемы также является активной – изменение отражаемого на ней имени подсхемы приводит к изменению информации на других панелях окна. Проверка корректности **синтаксического описания** активного блока проекта осуществляется в меню “Выполнить” → “Проверка синтаксиса активного блока”

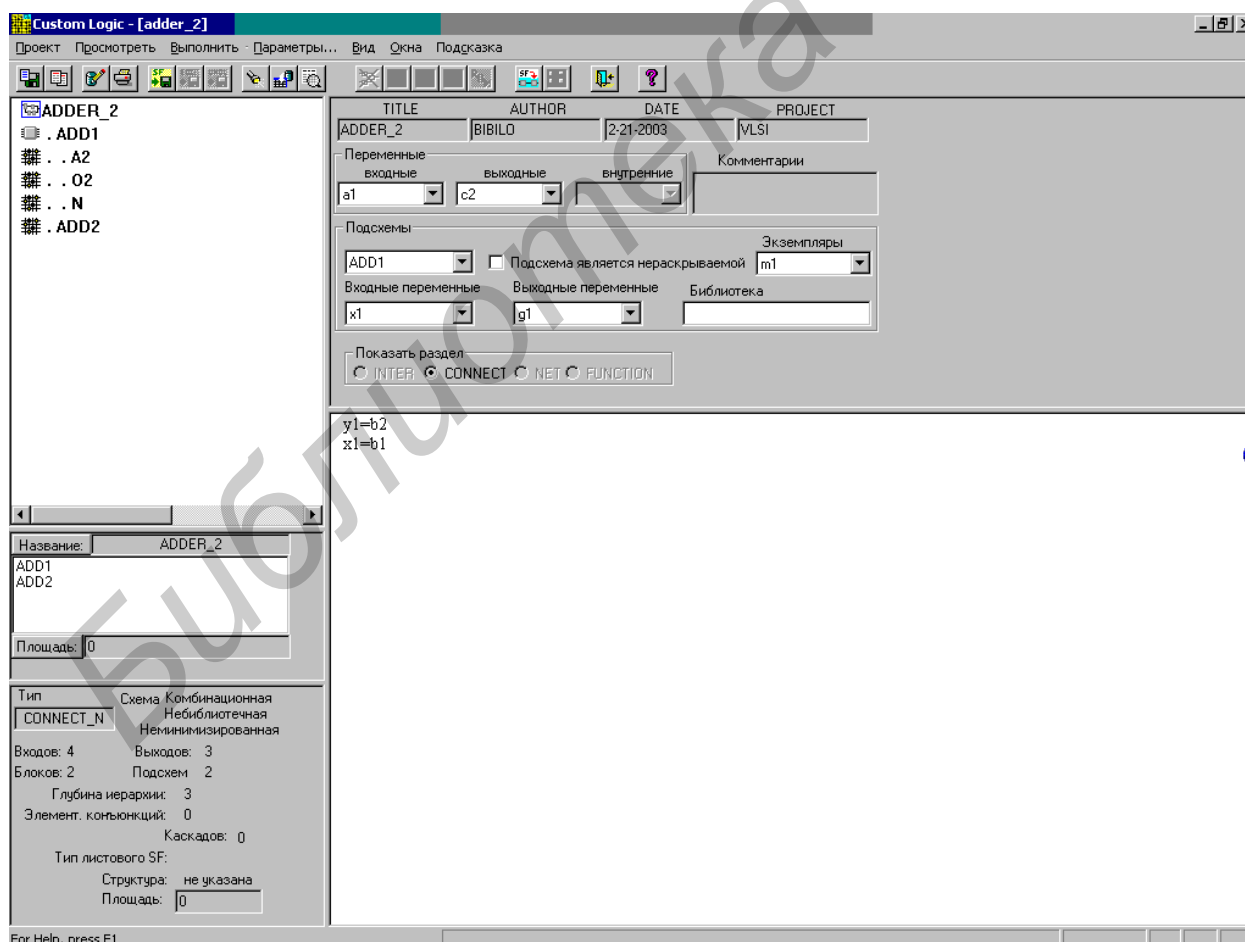


Рис. 13. Начальное состояние проекта (число уровней иерархии описания равно 3)

Пусть целью проектирования будет реализация логической схемы ADDER_2 на ПЛИМ.

Для получения формата SDF необходимо сначала выполнить проектную операцию “Устранение иерархии” (рис. 14).

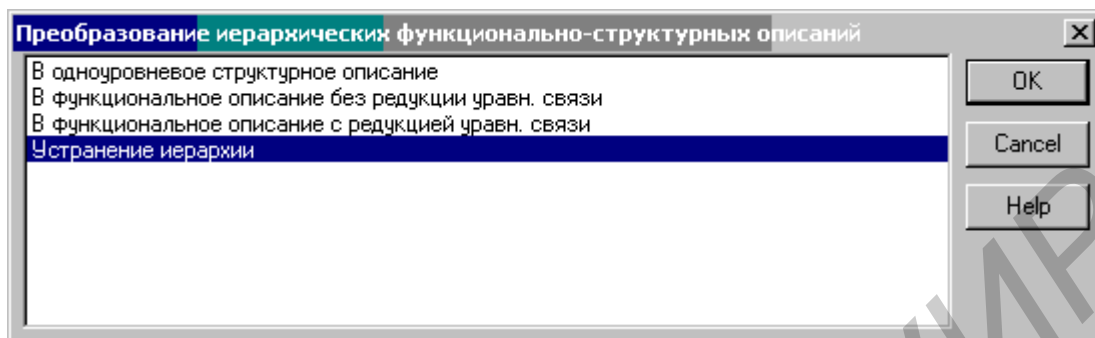


Рис. 14. Выбор проектной операции “Устранение иерархии”

При этом экран будет иметь вид, представленный на рис. 15. Иерархия описания будет равна единице, схема ADDER_2 будет описана на функциональном уровне в формате LOG.

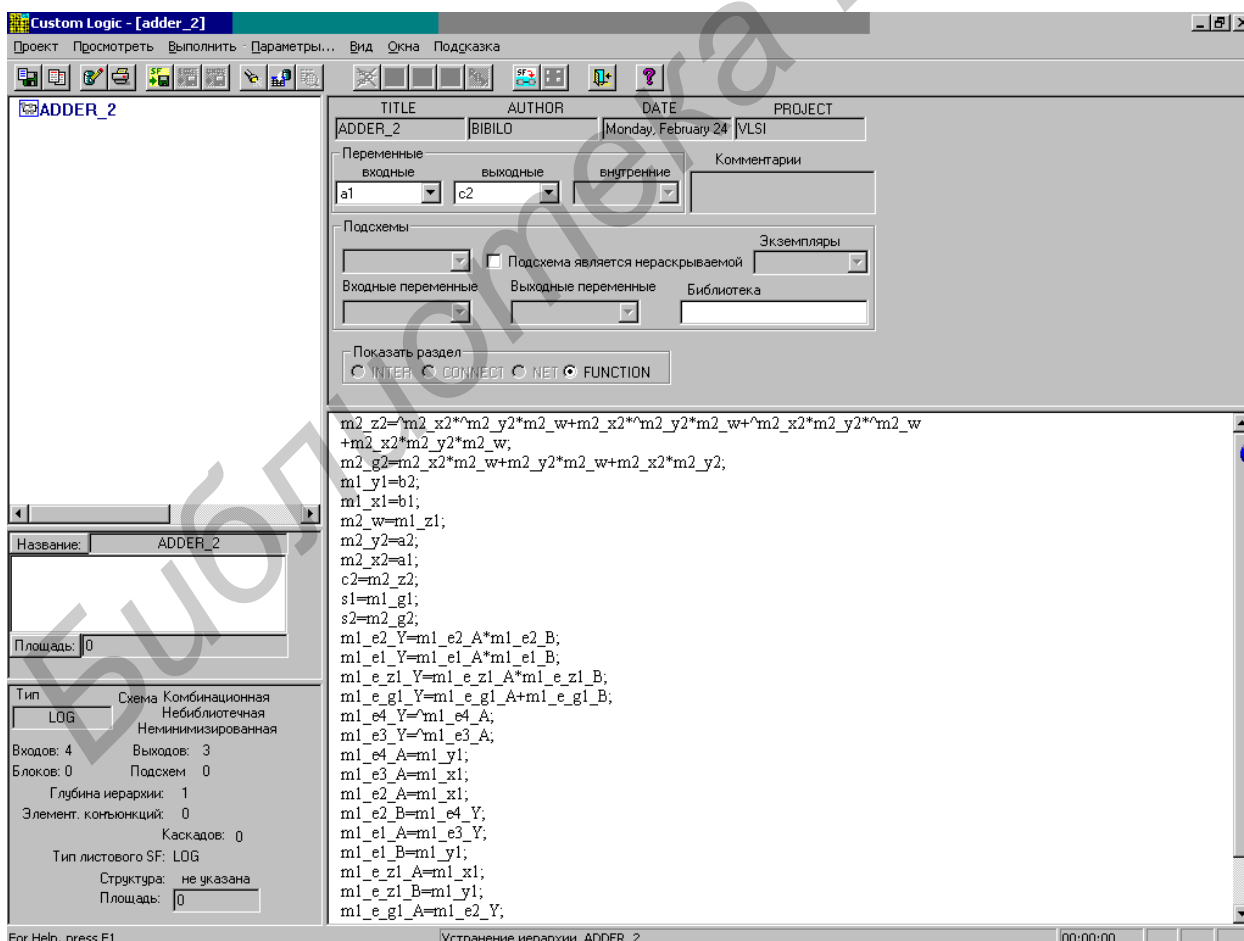


Рис. 15. Состояние проекта после операции “Устранение иерархии”

Затем требуется преобразовать формат LOG в формат SDF с исключением внутренних переменных. Для этого выполняется проектная операция “LOG → SDF с исключением внутренних переменных” (рис. 16).

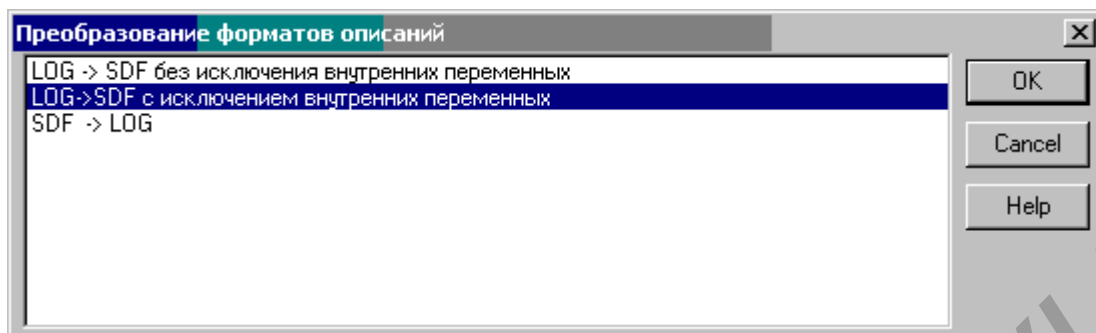


Рис. 16. Выбор проектной операции “LOG → SDF с исключением внутренних переменных”

Для **минимизации площади** ПЛМ выполняется проектная операция совместной минимизации системы булевых функций (рис. 17).

При минимизации площади ПЛМ минимизируется число промежуточных шин ПЛМ, т.е. число элементарных конъюнкций, на которых заданы ДНФ функций системы.

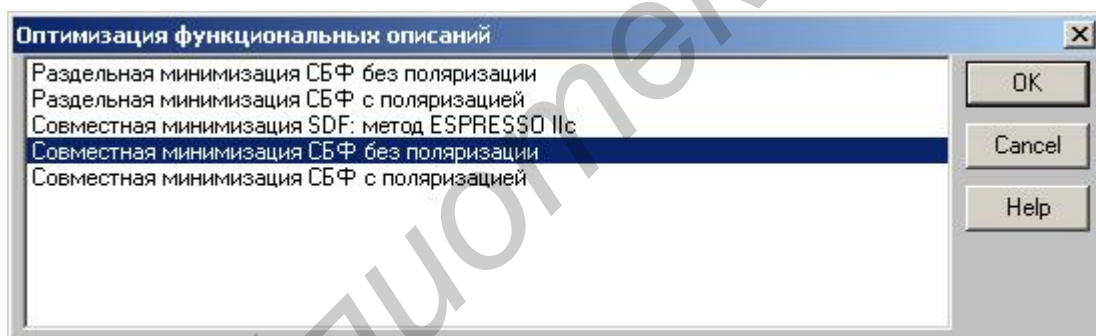


Рис. 17. Выбор проектной операции “Совместная минимизация СБФ без поляризации”

Итак, получена минимизированная система булевых функций, которую можно реализовать на ПЛМ. Заказать целевую структуру ПЛМ можно, нажав кнопку “PLA” желтого цвета. При этом значок схемы в иерархии описания (верхнее левое окно) также окрасится в желтый цвет. Затем выполняется проектная операция “**Компиляция**” – имеется в виду компиляция **послойной топологии – геометрического описания слоев** полупроводникового кристалла СБИС. Геометрическое описание слоев кристалла является пригодным для производства СБИС.

Компиляция послойной топологии проводится в три шага:

- на первом шаге строится структурное описание – файл ADDER_2.pla;
- на втором шаге строится описание символьной топологии – файл ADDER_2.tpl;
- на третьем шаге строится описание послойной топологии – файл ADDER_2.sou.

Структурное описание схемы ADDER_2 в языке PLA (файл ADDER_2.pla) имеет вид:

```
TITLE ADDER_2
FORMAT PLA
AUTHOR BIBILO
DATE Wednesday, March 19, 2003
PROJECT VLSI
INP
a1 ^a1 a2 ^a2 b1 ^b1 b2 ^b2
OUT
c2 s2 s1
INVERT
000
#####
4 3 11
#####
1111 100
010- 100
01-0 100
100- 100
10-0 100
0011 100
1-11 010
-111 010
11-- 010
--10 001
--01 001
END_ADDER_2
```

Описание символьной топологии ПЛМ, реализующей двухразрядный сумматор ADDER_2 (файл ADDER_2.tpl) имеет вид:

```
TITLE ADDER_2
FORMAT TPL
AUTHOR BIBILO
DATE 03/19/03
PROJECT VLSI
INP
a1 ^a1 a2 ^a2 b1 ^b1 b2 ^b2
OUT
c2 s2 s1
no_out3
LEFT
3 a2 ^a2
```

```

5  b2 ^b2
8  c2
11 s1
RIGHT
2  a1 ^a1
4  b1 ^b1
7  s2
10 no_out3
#####
BLV      T1          T2          T1          T2 BPV
BL_      1 0 0 1  GM1  1 0 1 - 1 - - -  GM1  - - - - BP
BL_      1 1 1 0  GM1  0 0 - 1 1 - - -  GM1  - - - - BP^
BL_      1 0 - 0  GM1  - 1 1 1 - 1 0 -  GM1  - - - - BP
BL_      1 - 0 -  GM1  0 1 1 1 - 0 1 -  GM1  - - - - BP^
OPL      P1          P2          P1          P2  OPP2
VL_      11 11          11 00 00 00          00 00  VP
VL_      00 00          00 11 10 00          00 00  VP^
VL_      00 00          00 00 01 10          00 00  VP
VL_      00 00          00 00 00 00          00 00  VP^
GM2L GM2 GM2          GM2 GM2 GM2 GM2          GM2 GM2 GM2P
END_ADDER_2

```

Ячейки (0, 1, GM1, BLV, T1 и т.д.) описывают **фрагменты** послойной топологии матричной структуры.

Корректное завершение компиляции сопровождается выдачей отчета о компиляции (рис. 18).

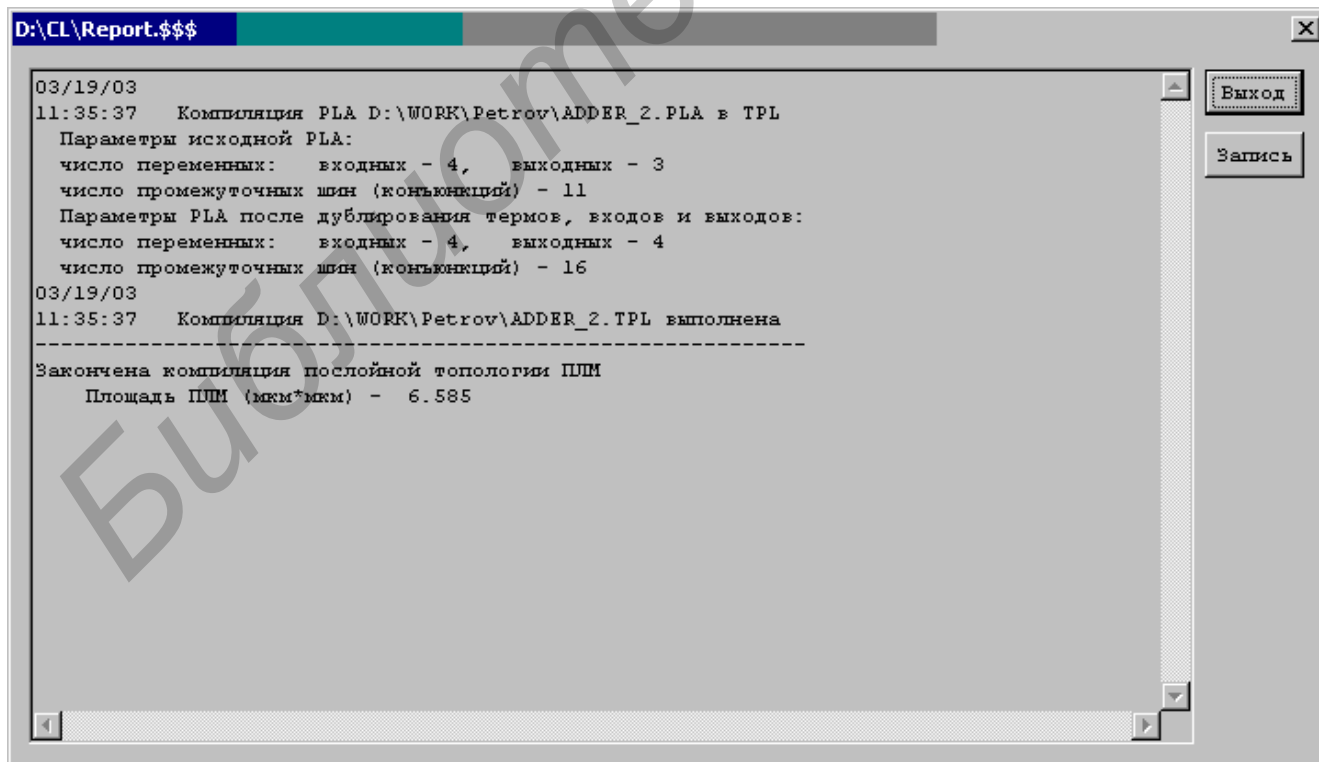


Рис. 18. Протокол компиляции в базовую структуру ПЛМ

Просмотр послойной топологии (файла ADDER_2.sou) осуществляется в меню

“Просмотреть” → “Символьная топология” → “SOU”

Для работы в вызванном редакторе послойной топологии GLE (Graphic Layout Editor) требуется нажать клавишу “Num Lock”. Управление масштабом изображения осуществляется клавишами “PgUp” (увеличение изображения) , “PgDn” (уменьшение изображения). Перемещение изображения производится с помощью клавиатуры (четырьмя стрелками).

На рис. 19 дано изображение послойной топологии макроэлемента параметризованной ПЛИМ, реализующей схему ADDER_2. В центральной части макроэлемента расположены информационные матрицы И, ИЛИ.

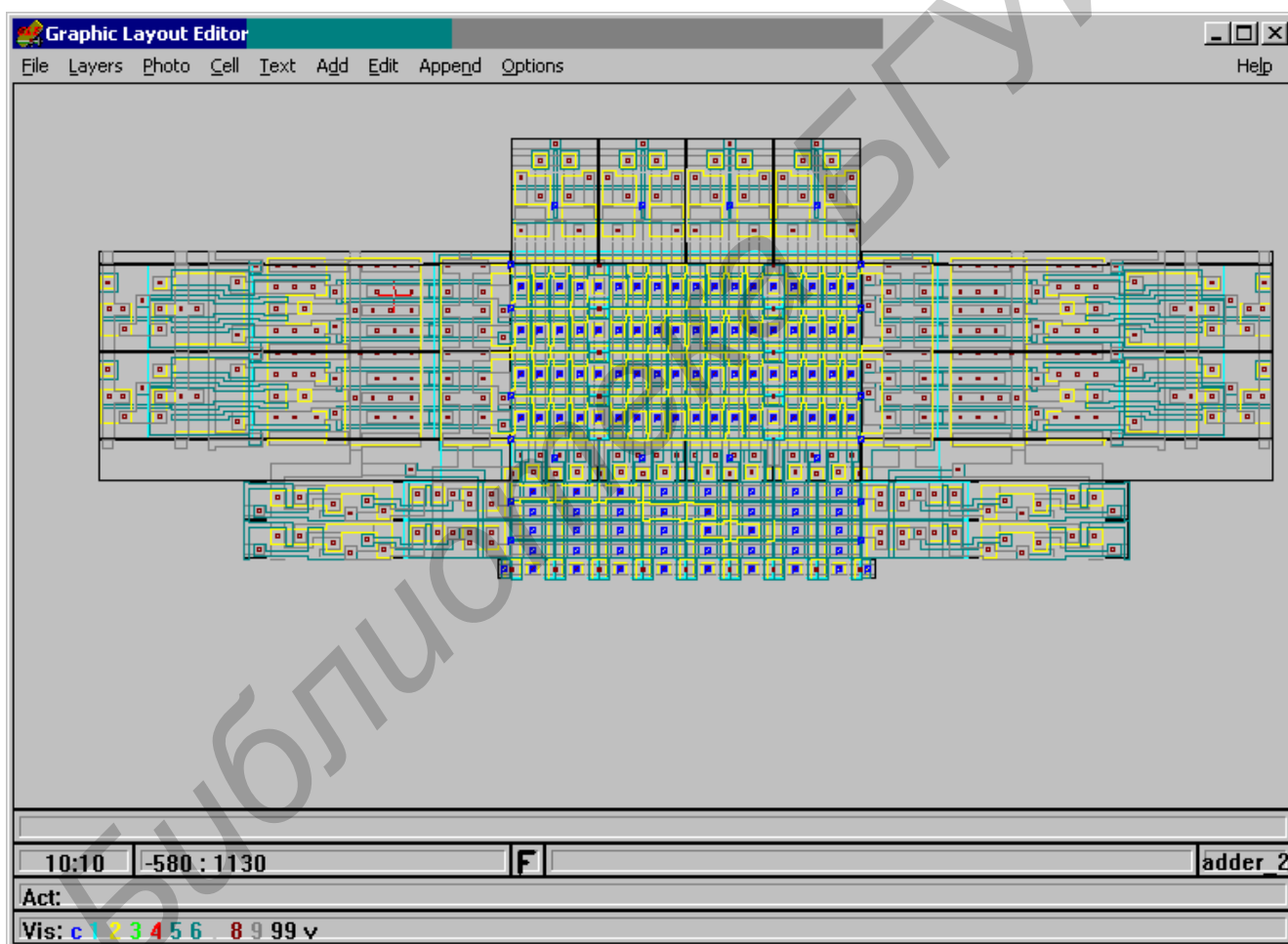


Рис. 19. Изображение в графическом редакторе GLE послойной топологии ПЛИМ, реализующей схему ADDER_2

ВНИМАНИЕ !

Получение послойной топологии базовой структуры возможно только для проекта, представленного в формате SDF.

Маршрут проектирования

1. Проверка синтаксиса SF-описаний проекта.
2. “Устранение иерархии”.
3. “LOG → SDF с исключением внутренних переменных”.
4. “Совместная минимизация СБФ без поляризации”.
5. Определение базовой структуры (PLA, MOS).
6. “Компиляция”.
7. Просмотр результатов.

4. ПОРЯДОК ВЫПОЛНЕНИЯ ЛАБОРАТОРНОЙ РАБОТЫ

1. Получить у преподавателя описание логической **схемы 1** на языке SF.
2. Нарисовать схему 1, обозначая имена входных, выходных полюсов элементов и схемы в целом, а также имен и типов элементов, входящих в схему 1.
3. Составить SF-описания логических элементов, входящих в схему 1.
4. Провести проектирование схемы 1 на базовой структуре типа ПЛМ (PLA) и получить систему булевых функций, реализуемых схемой 1.
5. Получить у преподавателя графическое изображение логической **схемы 2**.
6. Составить SF-описание схемы 2 и SF-описания логических элементов, входящих в схему 2.
7. Провести проектирование схемы 2 на базовой структуре типа ПЛМ (PLA) и получить систему булевых функций, реализуемых схемой 2.

5. СОДЕРЖАНИЕ ОТЧЕТА

1. Цель работы
2. Текстовое описание схемы 1.
3. Графическое изображение схемы 1.
4. Система булевых функций, реализуемая схемой 1.
5. Текстовое описание схемы 2.
6. Графическое изображение схемы 2.
7. Система булевых функций, реализуемая схемой 2.

6. КОНТРОЛЬНЫЕ ВОПРОСЫ И УПРАЖНЕНИЯ

1. Расшифруйте аббревиатуры САПР, СБИС, ДНФ, ПЛМ, РМОП-схема.
2. Можно ли в языке SF образовать проект только из структурных описаний?
3. Какие описания в проекте, имеющем несколько уровней иерархии описания, обязательно должны быть функциональными?

4. Может ли проект быть образован только одним функциональным описанием?

5. Перечислите ключевые слова языка SF.

6. Опишите в формате LOG функции логических элементов

инвертор;

n -входовый дизъюнктор;

n -входовый конъюнктор ($n = 1, \dots, 5$).

7. Запишите в форматах LOG и SDF систему функций

$$\begin{cases} f^1 = \bar{x}\bar{y}\bar{z} \vee xy; \\ f^2 = xy \vee xz; \\ f^3 = \bar{x}\bar{y}\bar{z} \vee xz. \end{cases}$$

8. Продемонстрируйте на ЭВМ создание проекта в системе Custom Logic.

9. Продемонстрируйте на ЭВМ основной маршрут проектирования базовых структур ПЛМ и РМОП-схем.

10. Перечислите языки описания схем и структур в системе Custom Logic.

11. Какие из этих языков являются внешними языками, а какие внутренними языками САПР ?

12. Что является результатом проектирования в системе Custom Logic ?

13. Перечислите основные проектные операции в маршруте проектирования послонной топологии ПЛМ.

14. Для чего производится совместная минимизация функций? Что при этом минимизируется?

15. С помощью какой проектной операции минимизируется площадь ПЛМ?

16. Запишите формулу, по которой высчитывается площадь ПЛМ.

17. Запишите формулу, по которой высчитывается площадь РМОП-схемы.

18. Какой параметр ПЛМ минимизируется при минимизации площади ПЛМ?

ЛИТЕРАТУРА

Бибило П.Н. Кремниевая компиляция заказных СБИС. – Мн.: Ин-т техн. кибернетики АНБ, 1996. – 268 с.

Учебное издание

Бибило Петр Николаевич

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ САПР

Лабораторный практикум

для студентов специальности 40 02 02
“Электронные вычислительные средства”
дневной формы обучения

Редактор Т.А. Лейко
Корректор Е.Н. Батурчик
Компьютерная верстка Т.В. Шестакова

Подписано в печать 12.05.2003.
Печать ризографическая.
Уч.-изд. л. 1,4.

Формат 60x84 1/16.
Гарнитура «Таймс».
Тираж 200 экз.

Бумага офсетная.
Усл. печ. л. 1,98.
Заказ 142.

Издатель и полиграфическое исполнение:
Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Лицензия ЛП № 156 от 30.12.2002.
Лицензия ЛТТ № 509 от 03.08.2001.
220013, Минск, П. Бровка, 6.