

Глава 16

Методы построения бинарных графов для автоматов без памяти

16.1. Структурирование линейных бинарных графов

В [1, 2] рассмотрены три подхода к структурированию неструктурированных граф-схем (ГС): дублирование вершин, введение состояний и введение булевых признаков.

Рассмотрим применение этих методов для структурирования линейных бинарных графов (ЛБГ).

16.1.1. Дублирование вершин

Если ЛБГ по всем S путям развернуть в дерево, то получится структурированный БГ (СБГ), сложность которого определяется числом и длиной путей в ЛБГ.

При этом число операторных вершин (ОВ) в СБГ определяется соотношением

$$B_{\text{оп}} = S_0 + S_1 = S,$$

а число условных вершин (УВ) — соотношением

$$B_y = S - 1.$$

Таким образом, общее число вершин в СБГ

$$B = 2S - 1.$$

В предыдущей главе приведено следующее соотношение

$$h + 1 \leq S \leq F_{h+2},$$

На рис. 16.3 приведен ЛБГ для заданной булевой формулы, а на рис. 16.4 — соответствующий СБГ, в котором число вершин определяется нижней оценкой.

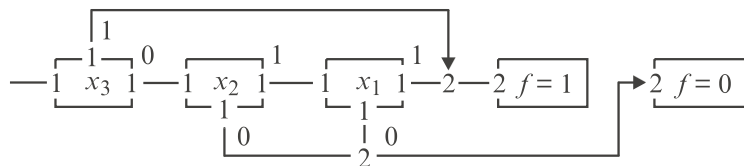


Рис. 16.3

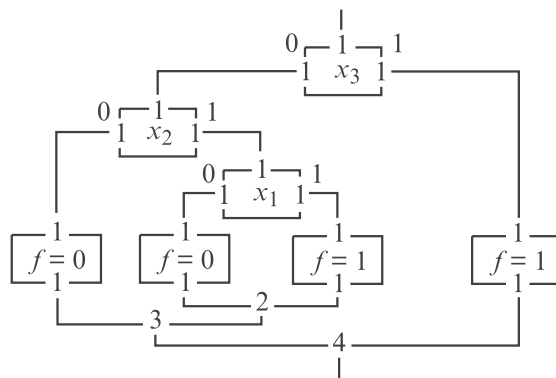


Рис. 16.4

Если после построения СБГ выполнить минимизацию числа вершин за счет подъема вверх ОБ, то число вершин в таких СБГ может быть оценено следующим образом:

$$B_{\text{оп}} = \min(S_0, S_1) + 1; \quad B_y = S - 1; \quad B_m = S + \min(S_0, S_1).$$

Так как $1 \leq \min(S_0, S_1) \leq F_{h+1}$, то $h + 2 \leq B_m \leq F_{h+3}$.

Как и в случае СБГ без минимизации числа вершин, эта верхняя оценка является завышенной.

Изложенный метод [4] не является эффективным, так как приведенные верхние оценки являются экспоненциальными от числа букв.

16.1.2. Введение состояний

При применении этого метода каждой точке объединения дуг в ЛБГ сопоставляется номер. Если при этом построить граф-схему, содержащую дешифратор введенных номеров, к каждому вы-

ходу которого подключить соответствующий фрагмент ЛБГ, в котором ОВ помечены номерами и (или) значениями выходной переменной, а после операторных вершин, помеченных значениями выходной переменной, осуществить сброс номера в других ОВ и это проверить, то, введя обратные связи, получится структурированная ГС (СГС), которая реализует заданную ЛБГ.

Эта СГС является арифметической [5] и эффективно реализуется, например, оператором switch языка СИ. Такая СГС является автоматной, а введенные номера соответствуют состояниям эквивалентного автомата.

Пример 16.3. Построить СГС для формулы $f = (x_1 \vee x_2)x_3 \vee x_4$.

На рис. 16.5 приведен ЛБГ для заданной БФ, содержащий три точки обьединения дуг, а на рис. 16.6 — соответствующая СГС.

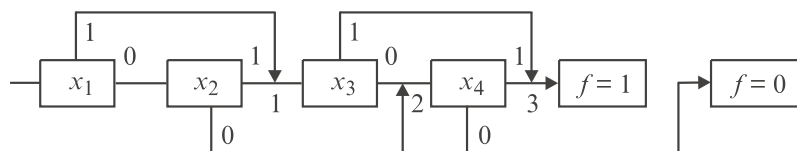


Рис. 16.5

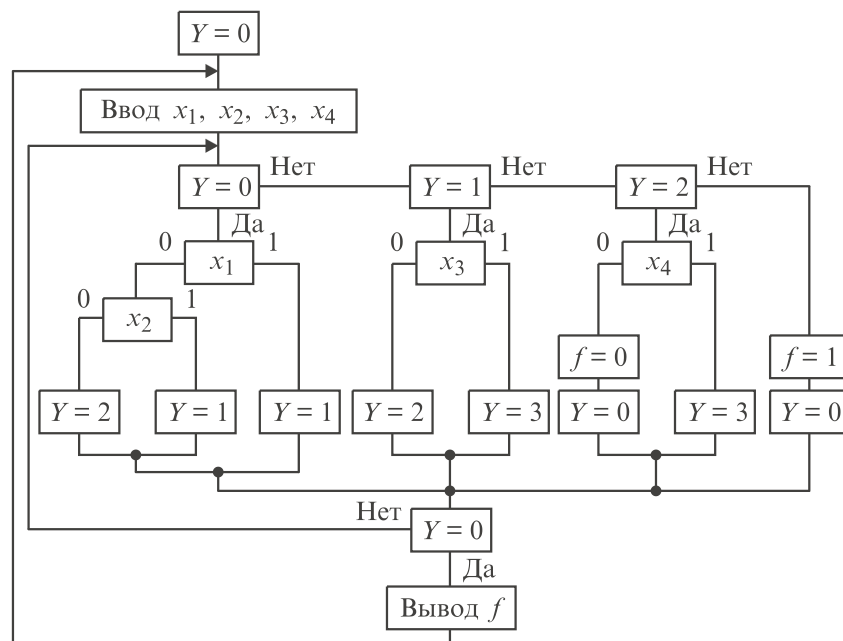


Рис. 16.6

Оценим число вершин в СГС этого типа: число бинарных УВ равно h , число бинарных ОВ равно двум, число арифметических УВ не превышает h , число арифметических ОВ не превышает $2h + 1$.

Таким образом,

$$B \leq 4h + 3.$$

Эта оценка, полученная автором и В. И. Рубиновым в [6], является линейной относительно числа букв в реализуемой БФ, правда при использовании арифметических УВ и ОВ.

16.1.3. Введение булевых признаков

При применении этого метода для каждой точки объединения дуг остова ЛБГ (за исключением точки, связанной с ОВ, если такая имеется) вводится булев признак (промежуточная переменная). После этого ЛБГ разбивается на фрагменты, которые структурируются методом дублирования с введением операторных вершин, в которых осуществляется присвоение булевым признакам значения, равного единице. Фрагменты объединяются с помощью условных вершин, предназначенных для проверки значений введенных признаков. На входе СБГ размещаются операторные вершины, в которых выполняется исходное обнуление этих признаков.

Оценим число вершин в таком СБГ, полагая, что остов ЛБГ имеет m точек объединения дуг, не связанных с операторными вершинами. При этом ЛБГ разбивается на $m + 1$ фрагментов. Так как i -й фрагмент уже не содержит точек объединения, то

$$B_i = 2h_i + 1.$$

При этом число вершин во всех фрагментах определяется соотношением

$$B_\phi = \sum_{i=1}^{m+1} (2h_i + 1) = 2h + m + 1.$$

Число дополнительных условных и операторных вершин определяется соотношением: $B_y = B_{оп} = m$. Таким образом, суммарное число вершин

$$B = 2h + 3m + 1.$$

Так как $0 \leq m \leq h - 2$, то

$$2h + 1 \leq B \leq 5(h - 1).$$

При этом отметим, что уже при $h > 4$ выполняется неравенство

$$5(h - 1) < 2F_{h+2} - 1.$$

Если ЛБГ минимизируется по числу путей, то из [6] следует, что

$$m = \begin{cases} (h-2)/2 & \text{при } h=2k; \\ (h-2)/3 & \text{при } h=2k+1. \end{cases}$$

Следовательно, $m = \lceil (h-2)/2 \rceil$. Таким образом,

$$B_{\text{макс}} = 2h + 3 \lceil (h-2)/2 \rceil + 1; \\ B_{\text{макс}} = \begin{cases} 3.5h - 2 & \text{при } h=2k; \\ 3.5(h-1) \leq [3.5h - 2] & \text{при } h=2k+1. \end{cases}$$

Поэтому

$$2h + 1 \leq B_{\text{макс}} \leq [3.5h - 2].$$

Рассмотренный метод структурирования [6] по оценкам сложности более эффективен, чем методы, рассмотренные выше. Он обеспечивает получение линейной верхней оценки числа вершин, которые имеют только двоичные пометки.

Пример 16.4. Структурировать ЛБГ (рис. 16.1) с помощью рассмотренного метода.

В этом случае $h = 3$, $m = 1$, а ЛБГ построен без минимизации числа путей. Поэтому $7 \leq B \leq 10$. На рис. 16.7 приведен СБГ с десятью вершинами.

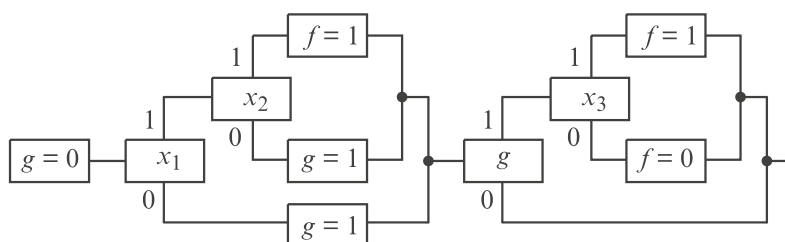


Рис. 16.7

16.1.4. Использование выходной переменной в качестве булева признака

1. Реализация неповторных пороговых формул. Исключить в методе введения булевых признаков проверку промежуточных переменных, соответствующих точкам остова, заменив ее проверкой значений выходной переменной, удастся в тех случаях, когда при каждом разложении Шеннона по одной переменной заданной БФ и всех ее остаточных по крайней мере одна из остаточных равна кон-

станте. Этот класс формул образуют бесповторные пороговые формулы (БПФ), которые обладают указанным свойством при их разложении в порядке невозрастания весов переменных.

Построение СБГ в этом случае выполняется следующим образом. Для заданной бесповторной пороговой формулы строится линейный бинарный граф с максимальным числом путей. Этот ЛБГ читается от выходов к входу и по нему формируются h фрагментов, первый из которых является блоком «полный выбор», а остальные — блоками «неполный выбор». При этом в каждом блоке используется только одна из условных вершин ЛБГ. Эти блоки формируются следующим образом.

Если в ЛБГ дуга, исходящая из условной вершины и не входящая в остов, соединена с нулевой (единичной) операторной вершиной или точкой остова, то соответствующая дуга условной вершины, расположенной в блоке, соединяется с операторной вершиной $f=0$ ($f=1$).

Блоки СБГ соединяются между собой последовательно до тех пор, пока в остове ЛБГ не встретится точка, которая заменяется условной вершиной, проверяющей значение выходной переменной и одна из дуг которой помечается символом, соответствующим символу остова, связанного с точкой, а другая соединяется с выходом.

После построения СБГ за счет выноса «вверх» ОВ (рис. 16.8) по нему в случае, изображенном на рис. 16.8, *а*, может быть построен линейризованный СБГ (ЛСБГ), содержащий тоже число вершин.

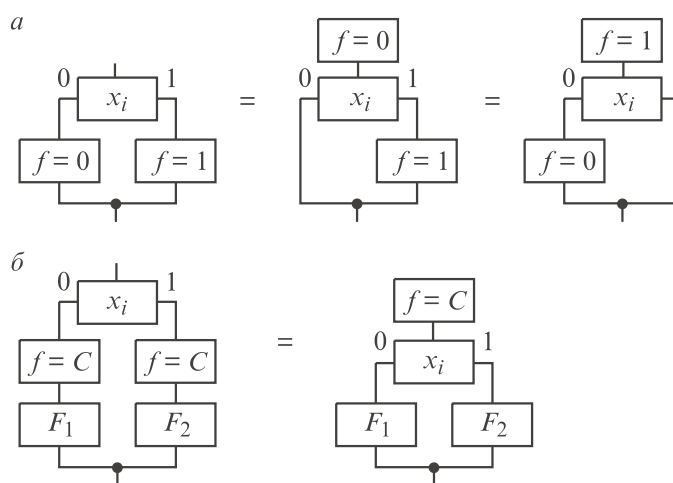


Рис. 16.8

Построенный ЛСБГ обладает длинными переходами, охватывающими несколько последовательно соединенных блоков.

Оценим число вершин в таких графах. В них число УВ с пометками входными переменными равно h , число ОВ равно $h + 1$, а число УВ с пометками выходной переменной — m . Таким образом, общее число вершин

$$B = 2h + m + 1.$$

Так как $0 \leq m \leq h - 2$, то

$$2h + 1 \leq B \leq 3h - 1.$$

Следовательно, предлагаемый метод позволяет для рассмотренного класса формул снизить верхнюю оценку сложности.

Пример 16.5. Построить ЛСБГ для неповторной пороговой формулы, рассмотренной в примере 16.1.

Для этой формулы $h = 3$. Поэтому $7 \leq B \leq 8$.

Построим по заданной формуле ЛБГ с максимальным числом путей (рис. 16.1). Структурированный бинарный граф, построенный по этому ЛБГ, приведен на рис. 16.9, а линейризованный СБГ — на рис. 16.10. Эти графы содержат число вершин, определяемое верхней оценкой.

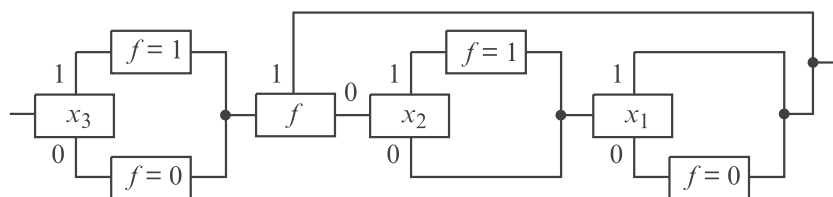


Рис. 16.9

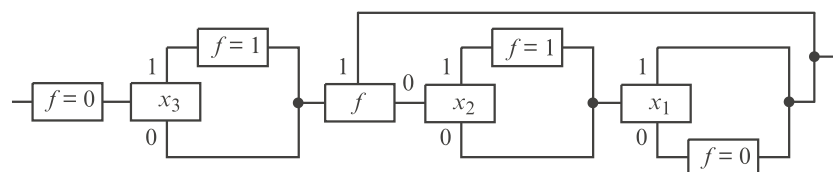


Рис. 16.10

2. Реализация каскадных булевых функций. Рассмотрим класс функций, реализуемых каскадами Майтра (разд. 14.2). В [8] эти функции были названы каскадными. Каждой каскадной функции соответствует каскадная формула, реализуемая ненастраиваемым каскадом Майтра. Функции этого класса могут быть также реализованы СБГ, в которых выходная переменная используется в качестве булева признака, если в построенном настраиваемом каскаде

Майтра каждый из $h - 1$ мультиплексоров «2 в 1» заменить моделирующим его фрагментом СБГ, в котором применяется управляющая конструкция «полный выбор». При таком методе построения каждая промежуточная переменная в каскаде может быть заменена в СБГ выходной переменной, используемой в качестве булева признака.

По сравнению с подходом, изложенным выше, данный подход расширяет класс функций, реализуемых СБГ с указанным свойством, так как БПФ являются лишь подклассом каскадных функций.

Пример 16.6. Построить СБГ для БПФ, рассмотренной в примере 16.1.

Настраиваемый каскад Майтра, реализующий эту формулу, приведен на рис. 16.11, а построенный по нему СБГ — на рис. 16.12.

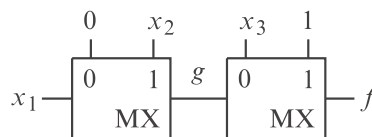


Рис. 16.11

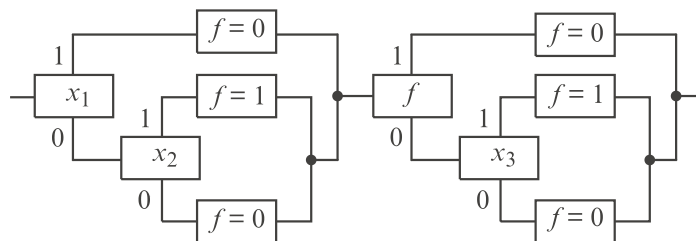


Рис. 16.12

Построенный СБГ содержит девять вершин против восьми у ЛСБГ (рис. 16.10).

Пример 16.7. Построить СБГ для булевой функции, реализуемой формулой

$$f = !x_1 x_3 \vee x_1 !x_3 \vee x_2 x_3.$$

Эта формула не является каскадной. Не является также каскадной и формула

$$f = (x_1 \oplus x_3) \vee x_2 x_3.$$

Однако каскадной является формула $f = (!x_1 \vee !x_2)x_3 \oplus x_1$. На рис. 16.13 представлен настраиваемый каскад Майтра, реализующий

щий эту формулу, а на рис. 16.14 — СБГ, соответствующий этому каскаду.

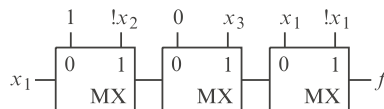


Рис. 16.13

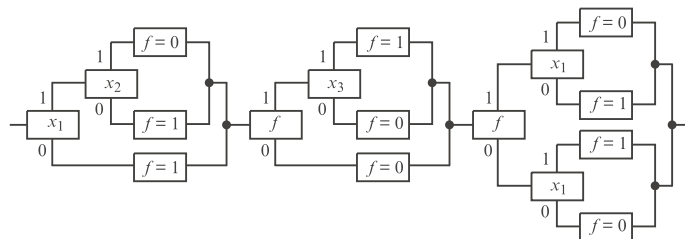


Рис. 16.14

16.2. Метод зависимых фрагментов

16.2.1. Реализация неповторных пороговых формул

Покажем, что для БПФ можно построить линейризованный СБГ, который не будет содержать булевых признаков и будет иметь только короткие переходы [7]. При этом возможны два подхода: структурирование ЛБГ и построение ЛСБГ непосредственно по формуле.

Первый подход состоит в построении линейного бинарного графа с минимальным числом путей и применении метода, изложенного в первом пункте предыдущего раздела.

Так как в таких ЛБГ для БПФ $t = 0$, то для ЛСБГ в этом случае $V = 2h + 1$.

Пример 16.8. Построить ЛСБГ для БПФ $f = x_1 x_2 \vee x_3$.

Линейный бинарный граф с минимальным числом путей приведен на рис. 16.3, структурированный БГ — на рис. 16.15, а линейри-

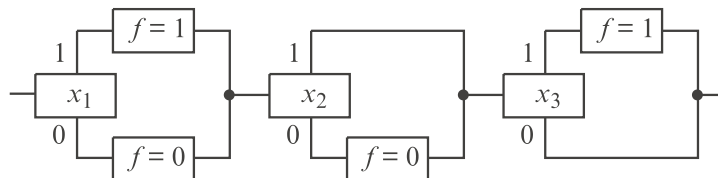


Рис. 16.15

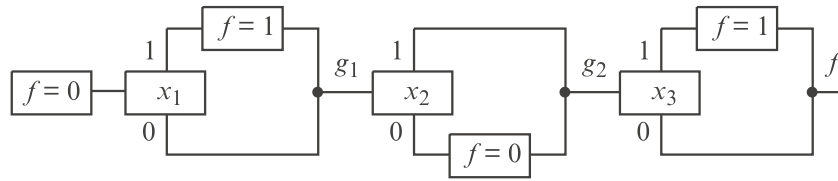


Рис. 16.16

зованный СПГ с короткими переходами, содержащий всего семь вершин, — на рис. 16.16.

Второй подход состоит в следующем. Запишем заданную БПФ в порядке неубывания весов переменных. После этого заменим подформулы в ней фрагментами ЛСБГ, используя следующие правила:

1. $f = 0 \& !f_2 \vee f_1 f_2 = f_1 f_2$ (рис. 16.17);
2. $f = f_1 !f_2 \vee 1 \& f_2 = f_1 \vee f_2$ (рис. 16.18);
3. $f = f_1 !f_2 \vee 0 \& f_2 = f_1 !f_2$ (рис. 16.19);
4. $f = 1 \& !f_2 \vee f_1 f_2 = !f_1 \vee f_2$ (рис. 16.20).

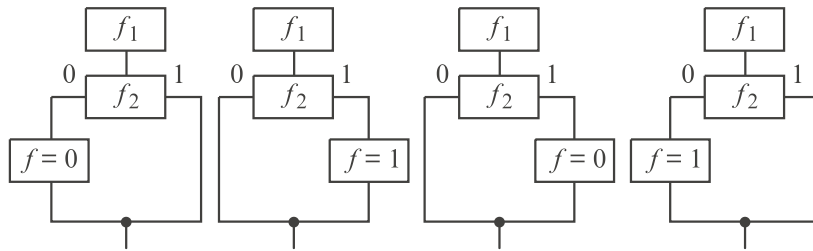


Рис. 16.17

Рис. 16.18

Рис. 16.19

Рис. 16.20

При этом отметим, что в отличие от метода независимых фрагментов (разд. 16.3), в котором правила определяют соответствующие операторы, приведенные соотношения соответствуют правилам алгебры логики.

Из этих правил следует, что решение для первой буквы БПФ неоднозначно: если она безынверсна, то могут применяться правила 1 и 2, если инверсна — правила 3 и 4.

Этот метод строит ЛСБГ с короткими переходами, содержащий $2h + 1$ вершин, в котором булевы признаки отсутствуют, а все блоки соединены последовательно. Он назван методом зависимых фрагментов (МЗФ), так как в построенном с его помощью ЛСБГ блоки, содержащие условные вершины, нельзя менять местами.

Пример 16.9. Построить ЛСБГ для БПФ $f = x_1 x_2 \vee x_3$.

Рассмотрим подформулу $g_1 = x_1$. Полагая $f_1 = (f = 0)$, $f_2 = x_1$, воспользуемся вторым правилом. В результате строится первый фрагмент ЛСБГ (рис. 16.21).

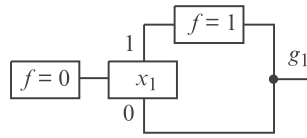


Рис. 16.21

Рассмотрим подформулу $g_2 = g_1 x_2$. Полагая $f_1 = g_1, f_2 = x_2$, воспользуемся первым правилом. В результате строится второй фрагмент ЛСБГ (рис. 16.22).

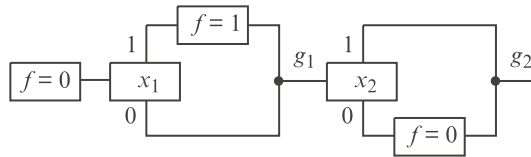


Рис. 16.22

Рассмотрим подформулу $f = g_2 \vee x_3$. Полагая $f_1 = g_2, f_2 = x_3$, воспользуемся вторым правилом. В результате строится искомый ЛСБГ (рис. 16.16).

16.2.2. Минимизация числа вершин

Расширим число типов фрагментов, которые можно применять при построении СБГ. Первый тип образуют фрагменты, реализующие конъюнкции, а второй — фрагменты, реализующие дизъюнкции. В этих фрагментах, в отличие от рассмотренных выше, блоки, содержащие условные вершины, являются вложенными.

На рис. 16.23 приведен ЛСБГ, реализующий конъюнкцию $f = x_1 x_2 x_3$, а на рис. 16.24 — дизъюнкцию $f = x_1 \vee x_2 \vee x_3$.

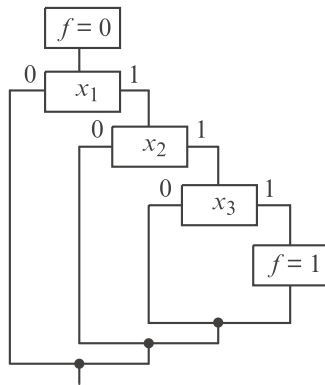


Рис. 16.23

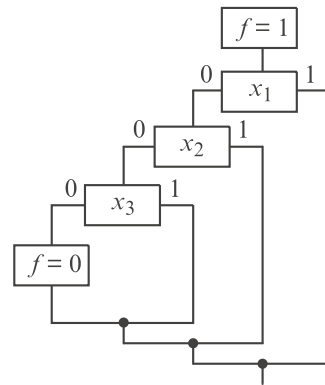


Рис. 16.24

Будем считать, что эти ЛСБГ с $h + 2$ вершинами также содержат только короткие переходы, так как в них отсутствуют «охватываемые» блоки, соединенные последовательно. При этом отметим, что для указанных формул метод, рассмотренный в предыдущем разделе, строит ЛСБГ с более короткими переходами, которые, однако, содержат $2h + 1$ вершин.

Для упрощения построения ЛСБГ может применяться соотношение

$$f = !f^n,$$

где f^n — формула, инверсная заданной.

При этом построение ЛСБГ для формулы f выполняется по ЛСБГ для формулы f^n за счет изменения пометок всех операторных вершин на противоположные: $f = 0$ на $f = 1$ и $f = 1$ на $f = 0$.

Пример 16.10. Построить ЛСБГ для БПФ $f = !x_3 (!x_2 \vee !x_1)$.

Так как $f = !(x_3 \vee x_2 x_1)$, то преобразуем указанным образом ЛСБГ на рис. 16.16. Искомая ЛСБГ приведена на рис. 16.25.

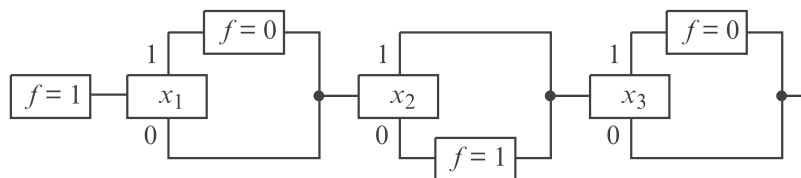


Рис. 16.25

Использование предложенных реализаций дизъюнкций и конъюнкций, а также последнего соотношения позволяет при $h > 1$ уменьшить оценку числа вершин в ЛСБГ, реализующих БПФ:

$$h + 2 \leq B \leq 2h.$$

При этом нижняя оценка достигается для дизъюнкций и конъюнкций, а верхняя оценка учитывает тот факт, что любая БПФ при ее записи в порядке неубывания весов переменных в однотипной с ней положительно монотонной формуле начинается либо с дизъюнкции, либо с конъюнкции, состоящих по крайней мере из двух букв и требующих для своей реализации четырех вершин, а не пяти, как в случае применения метода зависимых фрагментов.

Из изложенного следует, что рассмотренный подход может быть назван методом зависимых фрагментов с оптимизацией (МЗФО).

Пример 16.11. Реализовать БПФ $f = x_1 x_2 (x_3 \vee x_4 \vee x_5)$.

Метод зависимых фрагментов строит ЛСБГ с короткими переходами, содержащий 11 вершин (рис. 16.26).

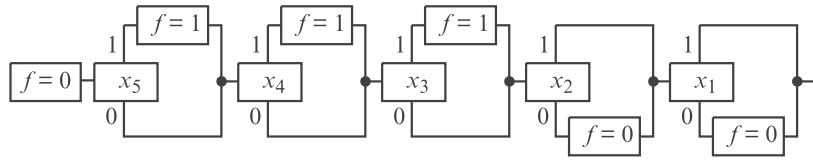


Рис. 16.26

Используем МЗФО. Для этого запишем заданную неповторную пороговую формулу в виде:

$$f = x_1 x_2 (!x_3 !x_4 x_5) = g_1 !g_2.$$

Для реализации подформулы g_1 применим второе соотношение для МЗФ, а для формулы в целом — третье (рис. 16.27).

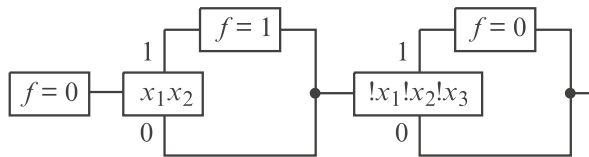


Рис. 16.27

Реализуя конъюнкции, как изложено выше, получим искомый ЛСБГ с короткими переходами, содержащий восемь вершин (рис. 16.28).

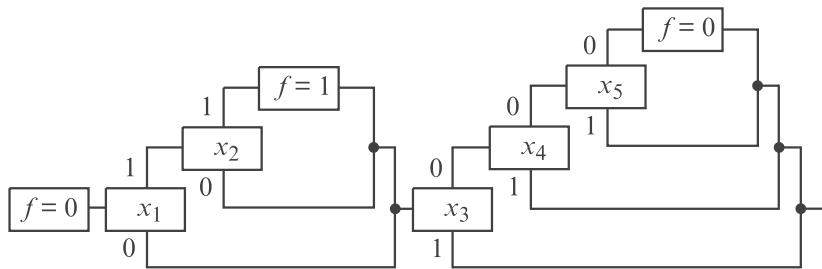


Рис. 16.28

Можно показать, что с помощью МЗФО могут быть реализованы непороговые неповторные БФ (ББФ), если они имеют неповторный образ, т.е. сводятся к БПФ либо после замены одиночными буквами отдельных дизъюнкций или конъюнкций в заданной ББФ или ее инверсии, либо после, по крайней мере, частичного раскрытия скобок в ББФ или ее инверсии. В первом случае строится бесповторный ЛСБГ, а во втором — повторный.

Пример 16.12. Реализовать непороговую ББФ

$$f = (x_1 x_2 \vee x_3 x_4) (x_5 \vee x_6).$$

Построим для этой ББФ пороговый образ $f = (z_1 \vee z_2) z_3$ и реализуем его ЛСБГ (рис. 16.29). Реализуя подформулы z_1, z_2, z_3 , построим искомый бесповторный ЛСБГ (рис. 16.30).

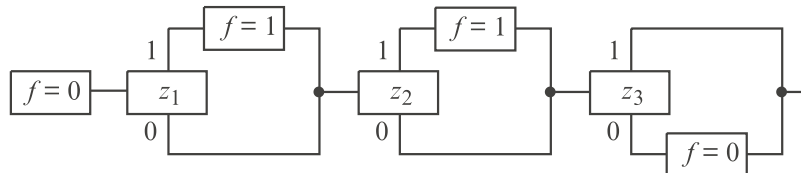


Рис. 16.29

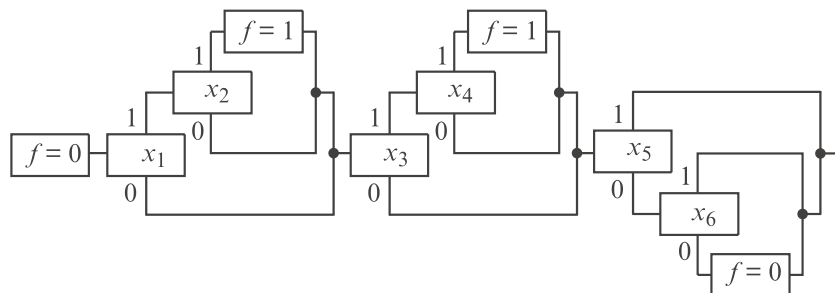


Рис. 16.30

Пример 16.13. Реализовать непороговую ББФ

$$f = (x_1 x_2 \vee x_3) (x_4 x_5 \vee x_6).$$

В данном случае пороговый образ имеет инверсия этой формулы при ее записи в виде: $!f = !((x_1 x_2) !x_3 \vee !x_4 !x_6 \vee !x_5 !x_6)$. Таким образом, в данном случае имеет место повторная реализация ББФ (рис. 16.31).

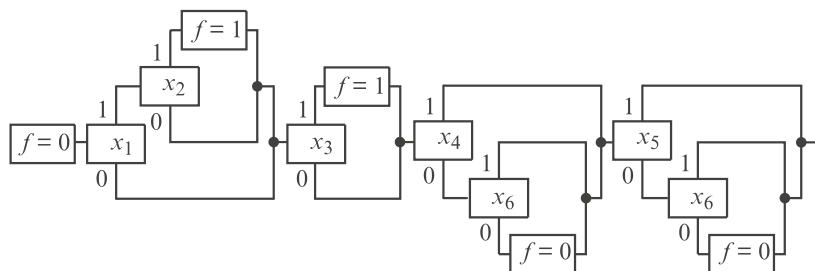


Рис. 16.31

Исследования, выполненные автором, показали, что: при $h \leq 6$ произвольная ББФ при применении указанных методов в отдельности реализуется бесповторным ЛСБГ;

при $h = 7$ существуют два из 180 представителей PN -типов ББФ, которые этими методами, используемыми в отдельности, не реализуются бесповторными ЛСБГ. Эти представители типов имеют вид:

$$f = (x_1 x_2 \vee x_3 x_4) (x_5 x_6 \vee x_7) \text{ и}$$

$$f = (x_1 \vee x_2) (x_3 \vee x_4) \vee (x_5 \vee x_6) x_7.$$

16.2.3. Смешанный метод

Для уменьшения сложности реализации некоторых формул, и в частности для обеспечения бесповторности реализации при $h = 7$ в классе ЛСБГ с длинными переходами, используем совместно МЗФО и метод независимых фрагментов (МНФ), который будет рассмотрен в разд. 16.3.

Пример 16.14. Реализовать ББФ

$$f = (x_1 x_2 \vee x_3 x_4) (x_5 x_6 \vee x_7).$$

Представим эту формулу в виде:

$$f = (x_1 x_2 \vee x_3 x_4) ! (x_7 ! (x_5 \vee ! x_6)) = (z_1 \vee z_2) z_3.$$

Реализуя пороговый образ с помощью МЗФО, а подформулы z_1, z_2, z_3 с помощью МНФ, построим бесповторный ЛСБГ (рис. 16.32).

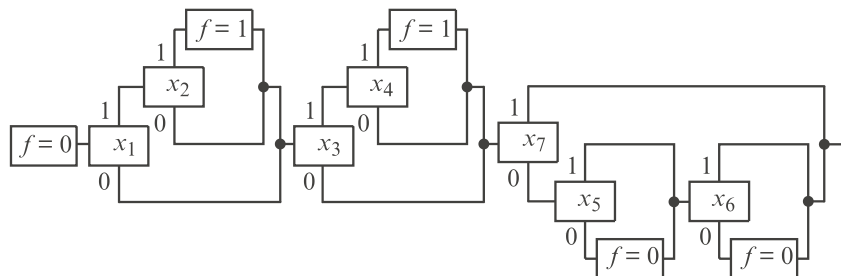


Рис. 16.32

Можно показать, что ББФ $f = (x_1 \vee x_2) (x_3 \vee x_4) \vee (x_5 \vee x_6) x_7$ с помощью смешанного метода также может быть реализована бесповторным ЛСБГ.

Таким образом, из изложенного следует, что при $h \leq 7$ произвольная ББФ реализуется бесповторным ЛСБГ. При этом можно показать, что:

при $h \leq 5$ произвольная ББФ реализуется бесповторным ЛСБГ с короткими переходами, число вершин в которых не превышает $2h$;

при $h \leq 7$ произвольная ББФ реализуется неповторным ЛСБГ с длинными переходами, число вершин в которых не превышает $2h - 1$.

В заключение раздела отметим, что на основе правил, рассмотренных в разд. 16.3, каждый из рассмотренных ЛСБГ может быть описан оператором структуры. Эти операторы будут применены в разд. 16.5.5 при реализации систем формул.

16.2.4. Использование разложения Шеннона

Применение этого разложения позволяет упростить ЛСБГ для некоторых формул. Это разложение целесообразно использовать для ББФ вида $f = x_i \vee F$ или $f = x_i F$, если начальная операторная вершина при реализации фрагмента F получена в виде $f = 1$ или $f = 0$ соответственно. При этом к заданной ББФ применяется разложение Шеннона по переменной x_i ; строится ЛСБГ для фрагмента F ; осуществляется подъем вверх одинаковых операторных вершин, связанных с условной вершиной, реализующей указанное разложение.

Пример 16.15. Реализовать БПФ

$$f = ((x_1 \vee x_2 \vee x_3)x_4x_5 \vee x_6)x_7.$$

Выполним разложение Шеннона по переменной x_7 и преобразуем неконстантную остаточную к виду, удобному для применения МЗФО:

$$f = 0 \& !x_7 \vee (x_4x_5!(x_1!x_2!x_3) \vee x_6)x_7.$$

Реализуем неконстантную остаточную с начальной вершиной $f = 0$. Для формулы в целом эту ОВ можно вынести вверх (рис. 16.33).

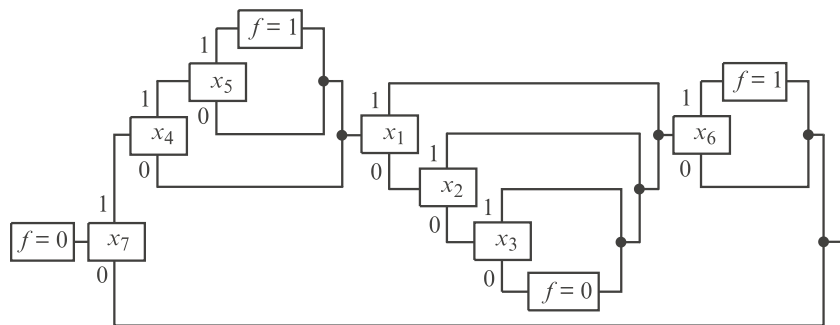


Рис. 16.33

Этот ЛСБГ содержит 11 вершин, в то время как МЗФО и МНФ в отдельности строят в этом случае граф-схемы с 12 вершинами.

16.2.5. Использование булевых признаков

Снижение трудоемкости построения ЛСБГ при $h \geq 8$ достигается за счет применения булевых признаков. При этом предлагается два подхода: использование «пороговых элементов» и использование «многофункциональных логических модулей».

При применении первого из них заданная ББФ представляется суперпозицией бесповторных пороговых формул, каждая из которых реализуется с помощью МЗФО. Построенные фрагменты соединяются последовательно. Можно показать, что в этом случае число вершин в ЛСБГ с короткими переходами определяется соотношением

$$h + 2 \leq B \leq 2(h + [h/2] - 1).$$

Пример 16.16. Реализовать ББФ $f = (x_1 \vee x_2)(x_3 \vee x_4)$. Представим эту формулу суперпозицией вида:

$$f_1 = x_1 \vee x_2 = !(x_1!x_2), \quad f = f_1(x_3 \vee x_4) = !(x_3!x_4)f_1.$$

Реализуем эту суперпозицию с помощью предлагаемого подхода (рис. 16.34).

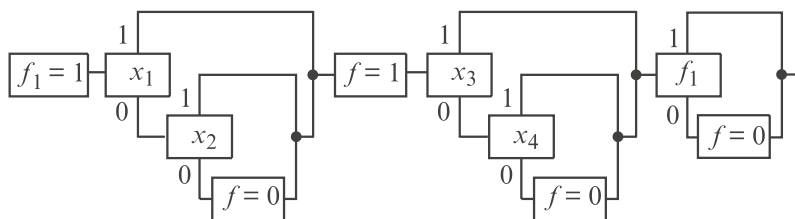


Рис. 16.34

При использовании второго подхода, применяя формульный метод построения схем (гл. 2), представим заданную ББФ в виде суперпозиции бесповторных булевых формул, каждая из которых содержит не более q букв. Реализуем каждую из этих формул с помощью ЛСБГ. Соединяя эти ЛСБГ последовательно, получим искомый ЛСБГ. Можно показать, что этот метод строит:

ЛСБГ с короткими переходами, число вершин в которых удовлетворяет соотношению

$$h + 2 \leq B \leq]2.8(h - 1)[;$$

ЛСБГ с длинными переходами, число вершин в которых удовлетворяет соотношению

$$h + 2 \leq B \leq]2.28(h - 1)[.$$

Пример 16.17. Реализовать ББФ

$$f = (x_1 \vee x_2) (x_3 \vee x_4) \vee (x_5 \vee x_6) (x_7 \vee x_8).$$

При $q = 5$ представим заданную ББФ в виде суперпозиции формул:

$$f_1 = (x_1 \vee x_2) (x_3 \vee x_4) = \overline{!x_1!x_2} \overline{!x_3!x_4},$$

$$f = f_1 \vee (x_5 \vee x_6) (x_7 \vee x_8) = f_1 \vee \overline{!x_5!x_6} \overline{!x_7!x_8}.$$

При этом заданная ББФ может быть реализована ЛСБГ с короткими переходами, представленным на рис. 16.35.

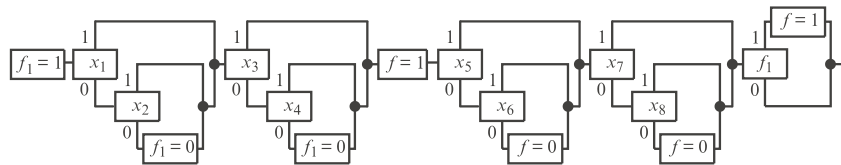


Рис. 16.35

16.2.6. Использование выходной переменной в качестве булева признака

Этот метод строит неповторные ЛСБГ, используя приведенные ниже соотношения.

На рис. 16.36 приведена структура, реализующая соотношение

$$f = f_1 \overline{!f_1} \vee f_2 f_1 = f_1 f_2.$$

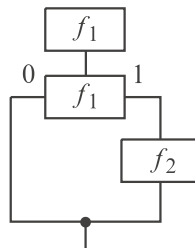


Рис. 16.36

Эта структура упрощается при замене в условной вершине пометки f_1 на f (рис. 16.37).

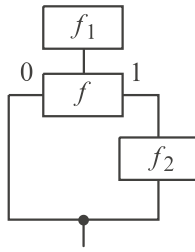


Рис. 16.37

На рис. 16.38 приведена структура, реализующая соотношение:

$$f = f_2!f_1 \vee f_1f_1 = f_1 \vee f_2.$$

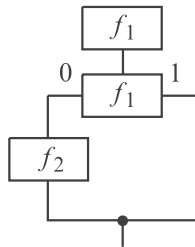


Рис. 16.38

Эта структура упрощается при замене в условной вершине пометки f_1 на f (рис. 16.39).

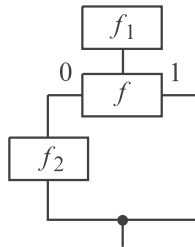


Рис. 16.39

Эти структуры позволяют за счет применения условной вершины с пометкой f присоединять подформулу f_2 к вычисленной подформуле f_1 .

Можно показать, что изложенный метод строит ЛСБГ с длинными переходами, число вершин в которых определяется соотношением

$$B \leq]2.25h - 1[.$$

Пример 16.18. Реализовать ББФ, приведенную в предыдущем примере.

Выделим в заданной ББФ две подформулы:

$$f_1 = (x_1 \vee x_2) (x_3 \vee x_4) = (!x_1!x_2)!(x_3!x_4),$$

$$f_2 = (x_5 \vee x_6) (x_7 \vee x_8) = (!x_5!x_6)!(x_7!x_8).$$

Так как в данном случае $f = f_1 \vee f_2$, то используем четвертую структуру (рис. 16.39) и заменим в ней операторные вершины с помечками f_1 и f_2 их реализациями (рис. 16.40).

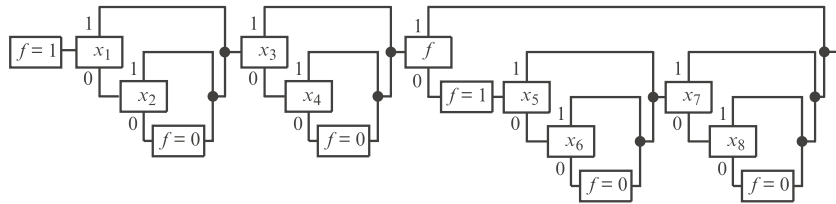


Рис. 16.40

Этот ЛСБГ содержит меньшее число вершин по сравнению с графом, приведенным на рис. 16.35.

16.3. Метод независимых фрагментов

Этот метод позволяет по заданной БФ в базе $\{\&, \vee, !\}$ определять оператор структуры (ОС), по которому изоморфно строится структурированный бинарный граф (СБГ).

Этот метод в некотором смысле эквивалентен методу построения параллельно-последовательной контактной схемы по БФ в указанном базисе, так как при минимизации ОС по числу букв минимизируется также и число вершин в СБГ.

При этом если оператор структуры содержит в правой части h_2 букв, то и ЛСБГ содержит h_2 вершин.

Если по ОС может быть построен не только ЛСБГ, но и СБГ, то последний содержит число вершин меньшее чем h_2 .

Метод состоит в следующем.

1. Заданная БФ из h букв представляется ДНФ из h_1 букв, состоящей из t конъюнкций ($h_1 \geq h$).

2. В ДНФ конъюнкция k_i ($i = 1, \dots, t$) помечается символом f .

3. Составляется ОС вида $F = !f | k_1 f | \dots | k_t f$.

4. Оператор структуры изоморфно реализуется с помощью ЛСБГ с короткими переходами, число вершин в котором определяется соотношением

$$h_2 = h_1 + t + 1.$$

Так как в [9] показано, что

$$t \leq 3^{h/3},$$

то

$$h_2 \leq h_1 + 3^{h/3} + 1.$$

Для булевых формул с максимальным числом термов

$$h_1 \leq \lceil h/3 \rceil \cdot 3^{h/3}.$$

Таким образом, для этого класса формул

$$h_2 \leq (\lceil h/3 \rceil + 1) \cdot 3^{h/3} + 1.$$

5. И наконец, если оператор структуры проминимизировать за счет выноса влево за скобки повторяющихся символов входных переменных и (или) их инверсий, то минимизированный ОС (МОС) из h_3 букв ($h_3 \leq h_2$) может быть изоморфно реализован ЛСБГ с длинными переходами, состоящим из h_3 вершин, либо СБГ, число вершин в котором меньше чем h_3 .

При этом отметим, что в МОС будем использовать только круглые скобки, которые могут быть произвольной глубины.

Переход от оператора структуры или минимизированного ОС к ЛСБГ осуществляется с помощью следующих замен фрагментов операторов на управляющие конструкции (УК): фрагмент $F_1 | F_2$ — на УК на рис. 16.41; фрагмент $x_i F$ — на УК на рис. 16.42; фрагмент $!x_i F$ — на УК на рис. 16.43.

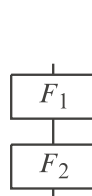


Рис. 16.41

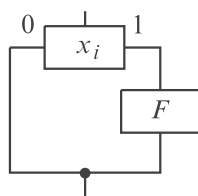


Рис. 16.42

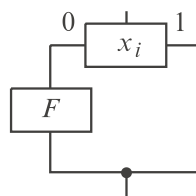


Рис. 16.43

При этом отметим, что если ОС (или МОС) содержит фрагмент $!x_i F_1 | x_i F_2$, то он реализуется структурой, представленной на рис. 16.44.

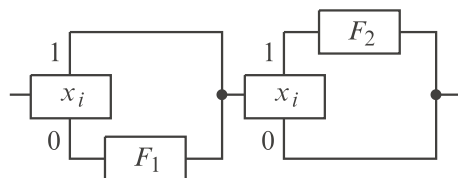


Рис. 16.44

Если в операторе фрагмент $!x_i F_1 | x_i F_2$ заменить на фрагмент $[!x_i F_1 \vee x_i F_2]$, то получим оператор с квадратными скобками. Составим фрагменту с квадратными скобками структуру, представленную на рис. 16.45.

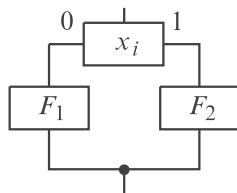


Рис. 16.45

Пример 16.19. Реализовать булеву формулу $f = x_1(x_2 \vee x_3)$. Оператор структуры в этом случае имеет вид: $F = !f | x_1 x_2 f | x_1 x_3 f$. Этот оператор реализуется ЛСБГ с короткими переходами, содержащим семь вершин (рис. 16.46).

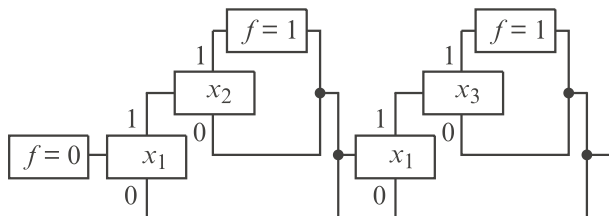


Рис. 16.46

Вынося переменную x_1 влево за скобки, построим МОС: $F = !f | x_1(x_2 f | x_3 f)$, который реализуется ЛСБГ с длинным переходом, содержащим шесть вершин (рис. 16.47).

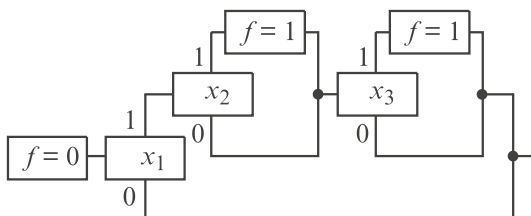


Рис. 16.47

Пример 16.20. Реализовать булеву формулу $f = !x_1 !x_2 !x_3 \vee x_1 x_2 x_3$.

Оператор структуры в этом случае имеет вид $F = !f | !x_1 !x_2 !x_3 f | x_1 x_2 x_3 f$ и реализуется ЛСБГ с короткими переходами, содержащим девять вершин.

Если построить оператор структуры с квадратными скобками

$$F = !f | [!x_1!x_2!x_3f \vee x_1x_2x_3f],$$

то он может быть реализован СБГ с восемью вершинами (рис. 16.48).

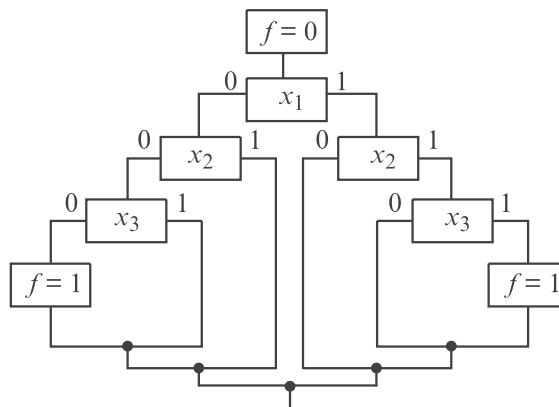


Рис. 16.48

Класс СБГ, который строится рассматриваемым методом, может быть расширен, если этот метод применять не только к ДНФ заданной функции, но и к ДНФ ее инверсии, содержащей t_1 конъюнкций. При этом оператор структуры ДНФ инверсии записывается в виде:

$$F_{\text{и}} = f | k_1^{\text{и}}!f | \dots | k_{t_1}^{\text{и}}!f.$$

Пример 16.21. Реализовать булеву формулу $f = x_1x_2x_3$.

В этом случае оператор структуры имеет вид $F = !f | x_1x_2x_3f$ и реализуется ЛСБГ с короткими переходами, представленным на рис. 16.23. Тело этого графа состоит из вложенных управляющих конструкций «неполный выбор».

Если построить оператор структуры ДНФ инверсии ($F_{\text{и}} = f | !x_1!f | !x_2!f | !x_3!f$), то этот оператор реализуется ЛСБГ с еще более короткими переходами, так как в нем все блоки соединены только последовательно (рис. 16.49).

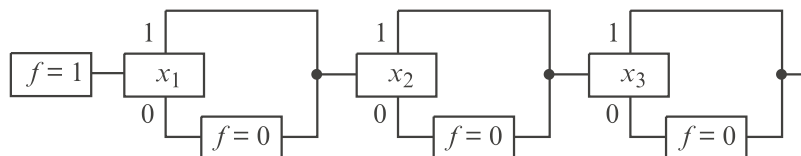


Рис. 16.49

В заключение отметим, что выше было показано как по оператору структуры, в частности, построить ЛСБГ. Естественно, может быть выполнен и обратный переход: по ЛСБГ, построенному любым методом, всегда может быть построен оператор структуры.

Анализ реализаций БПФ и их инверсий, построенных с помощью метода независимых фрагментов, показал, что для каждой пары таких формул суммарное число вершин в ЛСБГ, их реализующих, удовлетворяет соотношению

$$B = 3(h + 1).$$

Изложенный метод назван методом независимых фрагментов, так как правильность построения с его помощью бинарных графов не зависит от порядка расположения последовательно соединенных блоков, содержащих условные вершины.

Идея метода и его название были предложены в [10], а используемый математический аппарат и возможность его применения к ДНФ инверсии заданной функции — в [7, 11].

16.4. Методы построения бинарных графов для одной булевой функции

16.4.1. Распределительный метод

Предложим метод, который позволяет строить бинарный граф (БГ), в котором уровни условных вершин, рассматриваемые от корня графа, соответствуют входным переменным в таблице истинности (ТИ), читаемым в ней справа налево.

Метод состоит в распределении в условной вершине x_i двух частей столбца значений или его остатка. При этом часть, соответствующую $x_i=0$, образуют значения функции или ее остатка с четными номерами, а часть, соответствующую $x_i=1$, образуют значения функции или ее остатка с нечетными номерами.

Метод строит бинарные графы, в которых на каждом пути от входа к выходу каждая входная переменная встречается в условной вершине не более одного раза, а каждый путь завершается в операторной вершине.

Пример 16.22. Реализовать булеву функцию $f(x_3, x_2, x_1) = |0011\ 1101|^T$.

Построение искомого бинарного графа с семью вершинами выполнено на рис. 16.50.

Изменим порядок входных переменных в таблице истинности на противоположный.

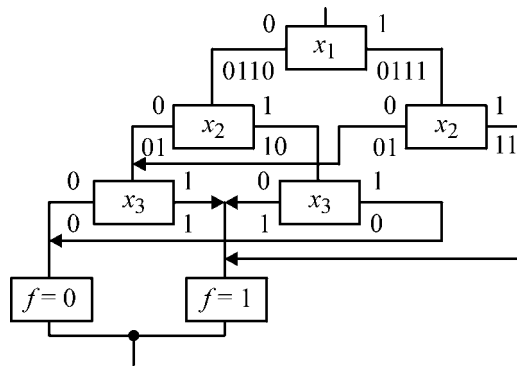


Рис. 16.50

Пример 16.23. Реализовать булеву функцию $f(x_1, x_2, x_3) = |0110\ 0111|^T$.

Построение искомого бинарного графа с шестью вершинами выполнено на рис. 16.51.

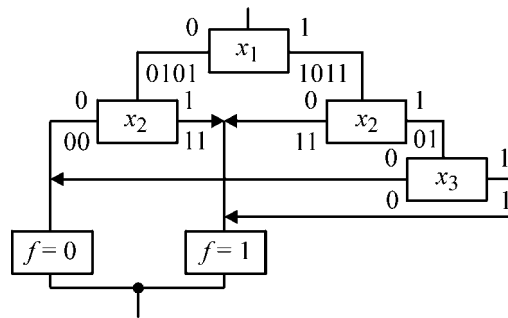


Рис. 16.51

Таким образом, перестановка входных переменных в таблице истинности позволила упростить бинарный граф.

16.4.2. Канонический метод Блоха

Рассмотрим метод [12], который позволяет строить БГ, в котором уровни УВ, рассматриваемые от корня графа, соответствуют входным переменным в ТИ, читаемым в ней слева направо.

Этот метод состоит из четырех этапов:

— построение канонической структуры (КС) по таблице истинности;

— кодирование кустов КС цифрами. При этом кусты первого уровня кодируются следующим образом: $|00|^T \rightarrow 0$; $|01|^T \rightarrow 2$; $|10|^T \rightarrow 3$; $|11|^T \rightarrow 1$. После этого если корни куста помечены одина-

ковыми цифрами, то и его вершина помечается той же цифрой. В противном случае для пометки вершины применяется следующая по значению цифра, которая при кодировании предыдущих кустов еще не была использована;

- объединение кустов, помеченных одинаковыми цифрами;
- построение по минимизированной КС изоморфного БГ.

При этом отметим, что если максимальный номер в КС равен N , то число вершин в построенном по ней бинарном графе определяется соотношением

$$B = N + 1.$$

Пример 16.24. Реализовать булеву функцию $f(x_1, x_2, x_3) = |0110\ 0111|^T$.

На рис. 16.52 приведена КС с $N = 6$, а на рис. 16.50 — искомый бинарный граф, содержащий семь вершин.

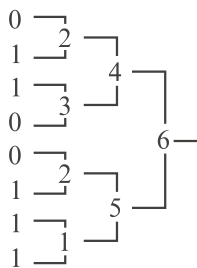


Рис. 16.52

Пример 16.25. Реализовать булеву функцию $f(x_3, x_2, x_1) = |0011\ 1101|^T$.

На рис. 16.53 приведена КС с $N = 5$, а на рис. 16.51 — искомый бинарный граф, содержащий шесть вершин.

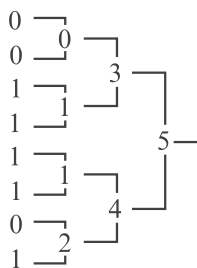


Рис. 16.53

Из изложенного следует, что применение канонического и распределительного методов к одной и той же таблице истинности позволяет без изменения порядка переменных в ней строить два бинарных графа и выбирать простейший из них.

Канонический метод позволяет строить также структурированные БГ, число вершин в которых определяется соотношением

$$2n + 1 \leq B \leq 3 \cdot 2^{n-1}.$$

После построения СБГ он может быть упрощен за счет подъема вверх тех ОВ, которых в построенном графе больше.

Пример 16.26. Реализовать БФУ

$$f(x_1, x_2, x_3) = |0111\ 1111|^T = x_1 \vee x_2 \vee x_3.$$

Построение СБГ по КС выполнено на рис. 16.54. После выноса ОВ с пометкой $f=1$ вверх получается СБГ с минимальным числом вершин, равным пяти (рис. 16.24).

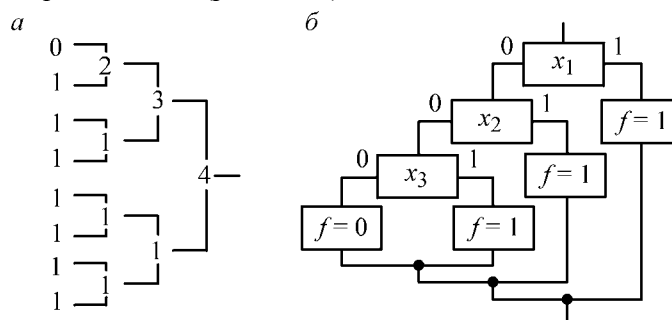


Рис. 16.54

В заключение раздела отметим, что если БФУ $f(x_1, x_2, x_3) = |0000\ 0111|^T$ реализуется рассматриваемым методом бесповторным БГ, состоящим из пяти вершин, который изоморфен планарному ЛБГ, построенному по формуле $f = x_1(x_2 \vee x_3)$, то эта БФУ, записанная как $f(x_3, x_1, x_2) = |0001\ 0011|^T$, реализуется этим методом непланарным повторным БГ, состоящим из шести вершин.

16.4.3. Реализация булевых функций линейными бинарными графами

В разд. 15.1 показано, что произвольная ББФ в базисе $\{\&, \vee, !\}$ может быть реализована как бесповторным планарным ЛБГ, так и бесповторной параллельно-последовательной контактной схемой.

Известно [13] также, что некоторые повторные БФ могут быть реализованы бесповторными контактными схемами, которые являются мостиковыми.

Покажем, что существуют повторные БФ, реализующиеся бесповторными ЛБГ, которые не являются планарными при рассмотрении их в качестве блоков с одним входом и одним выходом [3].

Пример 16.27. Реализовать повторную булеву формулу $f = !x_1x_2 \vee (x_1 \vee x_3)x_4$ бесповторным ЛБГ.
Эта БФ соответствует булевой функции

$$f(x_1, x_2, x_3, x_4) = |0001\ 1111\ 0101\ 0101|^T.$$

Построим по этой БФУ каноническую схему (рис. 16.55) и преобразуем ее в бинарный граф (рис. 16.56).

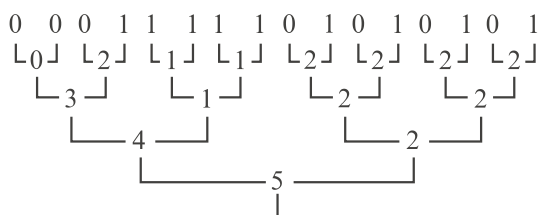


Рис. 16.55

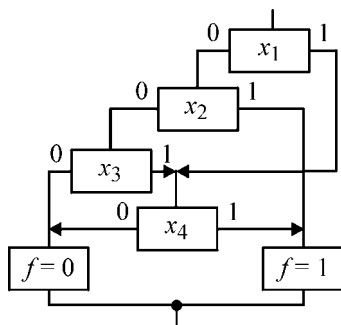


Рис. 16.56

Из рис. 16.56 следует, что заданная повторная БФ реализуется непланарным бесповторным бинарным графом. Линеаризуем этот граф [14] (рис. 16.57).

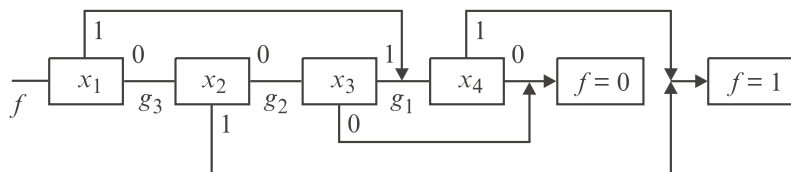


Рис. 16.57

Полученный граф является линейным и планарным, но после объединения выходов операторных вершин становится непланарным. В этом графе безынверсной переменной x_2 соответствует ребро, помеченное единицей и направленное вниз, а ОВ с пометкой $f=0$ расположена перед ОВ с пометкой $f=1$. Это отличает построенный граф от планарного повторного ЛБГ, построенного формульным методом (рис. 16.58).

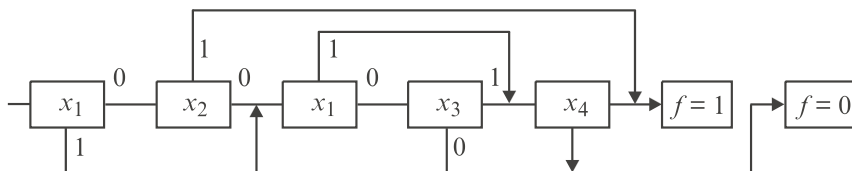


Рис. 16.58

Выполним верификацию построенного непланарного ЛБГ (рис. 16.57):

$$\begin{aligned}
 g_1 &= 0 \& !x_4 \vee 1 \& x_4 = x_4; \\
 g_2 &= 0 \& !x_3 \vee g_1 x_4 = x_3 x_4; \\
 g_3 &= g_2 !x_2 \vee 1 \& x_2 = x_2 \vee x_3 x_4; \\
 f &= g_3 !x_1 \vee g_1 x_1 = !x_1 x_2 \vee (x_1 \vee x_3) x_4.
 \end{aligned}$$

Отметим, что формула g_3 не является ни подформулой, ни фрагментом заданной булевой формулы, что отличает рассмотренный метод от формульного.

Известно [15], что ББФ в базисе $\{\&, \vee, !\}$, существенно зависящие от всех своих переменных, содержат нечетное число единиц в столбцах значений таблиц истинности.

Из приведенного примера следует, что нечетность числа единиц является необходимым, но недостаточным условием реализуемости БФУ формулой, бесповторной в базисе $\{\&, \vee, !\}$.

Автор предполагает также, что нечетность числа единиц является необходимым условием для реализации БФУ с помощью ЛБГ. Покажем, что это условие не является достаточным и в этом случае.

Пример 16.28. Реализовать булеву функцию

$$\begin{aligned}
 f(x_1, x_2, x_3) &= |0001 \ 0110|^T = \\
 &= !x_1 x_2 x_3 \vee x_1 (x_2 \oplus x_3) = x_1 (x_2 \vee x_3) \oplus x_2 x_3,
 \end{aligned}$$

которая существенно зависит от всех переменных и содержит нечетное число единиц.

Построение бинарного графа выполнено на рис. 16.59.

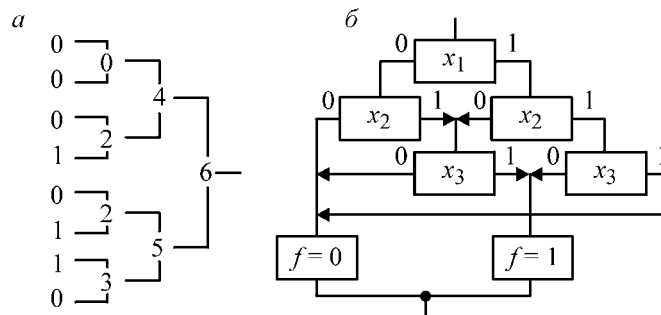


Рис. 16.59

Сформулируем необходимое и достаточное условие реализуемости булевой функции n переменных с помощью неповторного ЛБГ [3]. Это условие состоит из трех составляющих:

- БФУ существенно зависит от всех переменных;
- число единиц в столбце значений ТИ нечетно;
- максимальный номер в канонической структуре равен n , что обеспечивается тем, что каждый ее столбец содержит не более одного нового номера.

Это условие выполняется для булевой функции из примера 16.27 и не выполняется для функции из примера 16.28.

16.4.4. Оценки числа условных вершин в бинарных графах

Известно [12, 16, 17], что для булевой функции n переменных может быть построен бинарный граф, число условных вершин в котором при $n \rightarrow \infty$ определяется соотношением

$$V_y \sim 2^n / n$$

Однако метод, который строит такие бинарные графы, весьма сложен и неэффективен при малых значениях n .

Канонический, а как следует из изложенного выше, и распределительный методы обладают тем свойством, что они строят эффективные бинарные графы при малых значениях n , а при $n \rightarrow \infty$ для них справедливо соотношение

$$V_y \sim 2^{n+1} / n$$

Докажем эту оценку несколько иначе, чем это сделано в [12], с целью получения расчетных соотношений не только для случая $n \rightarrow \infty$ [6].

Будем рассматривать правильные канонические структуры [12], в которых все вершины одного яруса помечены одной переменной.

Наиболее сложно реализуемой БФУ является та, в которой каждый столбец КС имеет максимально возможное число новых номеров. Каноническая структура для БФУ n переменных состоит из n столбцов. Разобьем эту схему на две части. При этом первая будет содержать m столбцов, а вторая — $(n - m)$ столбцов, где $m = 0, 1, \dots, n$. Для столбца значений таблицы истинности $m = 0$.

В первой половине КС номера повторяются, а во второй — для самых сложных БФУ все номера различны. Максимальное число номеров в столбце с номером m в КС определяется соотношением

$$N_m^1 = 2^{2^m}.$$

Максимальное число новых номеров в этом столбце:

$$N_m = N_m^1 - N_{m-1}^1 = 2^{2^m} - 2^{2^{m-1}}.$$

Максимальное число различных номеров в первых m столбцах канонической структуры определяется соотношением

$$N_{r,1} = \sum_{i=1}^m (2^{2^i} - 2^{2^{i-1}}) = 2^{2^m} - 2.$$

Рассмотрим столбец канонической структуры с номером $n - m$. Максимальное число различных номеров в нем:

$$N_{n-m} = 2^{n-m-1}.$$

Максимальное число различных номеров в столбцах с номерами от $n - m$ до n определяется соотношением

$$N_{r,2} = 2^{n-m} - 1.$$

Максимальное число различных номеров в КС:

$$N = N_{r,1} + N_{r,2} = 2^{2^m} + 2^{n-m} - 3.$$

Максимальное число различных номеров в канонической структуре равно максимальному числу условных вершин в БГ:

$$B_{y,m} = N.$$

Определим величину m , при которой значение $B_{y,m}$ минимально. Предположим, что $m_1 = \log n$, тогда

$$B_{y,1} = 2^n + \frac{2^n}{n} - 3.$$

Эта величина превышает число вершин в полном дереве, для которого

$$B_y = 2^n - 1.$$

Уменьшим величину m . Пусть

$$m_2 = \lceil \log(n - \lfloor \log n \rfloor) \rceil.$$

При этом

$$B_{y_2} \approx \frac{2^n}{n} + \frac{2^n}{n - \log n} - 3 = \frac{2^n}{n} \left(1 + \frac{n}{n - \log n} \right) - 3.$$

При $n \rightarrow \infty$

$$B_{y_2} \sim \frac{2^{n+1}}{n}.$$

Отметим, что $B_{y_1} > B_{y_2}$. Для того чтобы m_2 обеспечивало минимум, уменьшим m по сравнению с m_2 . Пусть

$$m_3 = \log(n - 2 \log n).$$

При этом

$$B_{y_3} = \frac{2^n}{n^2} + \frac{2^n}{n - 2 \log n} - 3.$$

Уже при малых значениях n справедливо, что $B_{y_3} > B_{y_2}$. Таким образом, минимум числа условных вершин для самых сложных БФУ достигается при

$$m = \lceil \log(n - \lfloor \log n \rfloor) \rceil; \quad 2^m + m \leq n \leq 2^{m+1} + m.$$

Построим на основе последнего соотношения табл. 16.1.

Т а б л и ц а 16.1

m	0	1	2	3	4	5	6	7
$2^m + m$	1	3	6	11	20	37	70	135
$2^{m+1} + m$	2	5	10	19	36	69	134	263

Таким образом, при $n = 1, 2$ $B_{ym} = 2^n - 1$; при $n = 3 \div 5$ $B_{ym} = 2^{n-1} + 1$; при $n = 6 \div 10$ $B_{ym} = 2^{n-2} + 13$; при $n = 11 \div 19$ $B_{ym} = 2^{n-3} + 253$; при $n = 20 \div 36$ $B_{ym} = 2^{n-4} + 65\,533$.

Используя эти соотношения, построим табл. 16.2.

Т а б л и ц а 16.2

n	1	2	3	4	5	6	7	8	9	10
B_{ym}	1	3	5	9	17	29	45	77	141	269

Эти значения получены также в [12]. В этой работе показано, что данные оценки достижимы и не могут быть уменьшены за счет перестановки входных переменных. Они достигаются для следующих типов функций, заданных конститuentами единицы: при $n = 1$ $f = \{1\}$; при $n = 2$ $f = \{1, 2\}$; при $n = 3$ $f = \{1, 2, 4, 7\}$; при $n = 4$ $f = \{0, 1, 2, 9, 10, 11, 15\}$; $f = \{0, 1, 2, 5, 11, 12, 14, 15\}$.

Как отмечено в разд. 15.10, максимальное число путей в бинарных графах $S = 2^n$ и достигается для функций типа $f = x_1 \oplus x_2 \oplus \dots \oplus x_n$, а число условных вершин определяется соотношением $B_y = 2n - 1$.

При $n = 4$ для линейной функции $B_y = 7$, в то время как из табл. 16.2 следует, что при $n = 4$ $B_{ym} = 9$.

Таким образом, самые сложные по числу вершин бинарные графы не являются таковыми по числу путей, и наоборот.

Из изложенного следует, что

$$n \leq B_y(n) \leq B_{ym}; n + 1 \leq S(n) \leq 2^n.$$

16.4.5. Сокращение перебора при поиске минимальной реализации заданной булевой функции

Построение для заданной БФУ бинарного графа с минимальным числом вершин и (или) минимальным числом путей связано с перебором по порядку расположения входных переменных в ТИ.

1. Предложим подход, который при построении БГ с минимальным числом путей позволяет либо устранить или уменьшить перебор, либо оценить, насколько найденное без перебора решение по числу путей отличается от нижней оценки их числа, которая, возможно, недостижима [6].

Для заданной БФУ в качестве нижней оценки числа путей предлагается выбрать число максимальных ортогональных областей, которые можно выделить в карте Карно, построенной для этой функции. Это объясняется тем, что, во-первых, это число не зависит от порядка входных переменных в таблице истинности, а, во-вторых, пути в бинарном графе соответствуют ортогональным областям в карте Карно, которые не обязательно являются максимальными. При этом число максимальных единичных (нулевых) ортогональных областей является нижней оценкой числа единичных (нулевых) путей в бинарном графе.

В качестве верхней оценки числа путей для заданной БФУ может быть выбрано минимальное число путей в бинарных графах, построенных при заданном порядке переменных (без перебора) каноническим и распределительным методами.

При этом если в построенном БГ число путей совпадает с нижней оценкой, то она является минимальной по числу путей.

Если число путей отличается от нижней оценки, то становится ясно, на какую величину может быть возможно сократить число путей в результате перебора. Если при переборе достигается нижняя оценка, то перебор может быть завершен.

Пример 16.29. Использовать предложенный подход для БФУ

$$f(x_1, x_2, x_3) = |0101\ 0111|^T.$$

Так как в этом случае $n = 3$, то $4 \leq S(3) \leq 8$. Построим карту Карно для этой функции и выделим в ней максимальные ортогональные области (табл. 16.3).

Т а б л и ц а 16.3

x_i	x_2x_3			
	00	01	11	10
0	0	1	1	0
1	0	1	1	1

Из рассмотрения этой карты следует, что она содержит четыре максимальные ортогональные области, и поэтому в данном случае приведенная выше нижняя оценка не может быть повышена.

Канонический метод строит в данном случае бинарный граф с пятью путями, которые соответствуют ортогональным областям, выделенным в табл. 16.4.

Т а б л и ц а 16.4

x_1	x_2x_3			
	00	01	11	10
0	0	1	1	0
1	0	1	1	1

Распределительный метод в данном случае строит бинарный граф с четырьмя путями. Таким образом, минимальное решение найдено в данном случае без перебора.

Пример 16.30. Использовать предложенный подход для БФУ

$$f(x_1, x_2, x_3) = |0110 \ 1001|^T.$$

Так как в этом случае $n = 3$, то $4 \leq S(3) \leq 8$. Карта Карно для этой БФУ содержит восемь максимальных ортогональных областей, и поэтому $S = 8$. Бинарный граф с этим числом путей строит как канонический, так и распределительный методы.

Пример 16.31. Использовать предложенный подход для БФУ

$$f(x_1, x_2, x_3, x_4) = |0001 \ 1011 \ 1000 \ 0111|^T.$$

В этом случае $5 \leq S(4) \leq 16$. Карта Карно для этой булевой функции содержит 10 максимальных ортогональных областей. Поэтому $10 \leq S(f) \leq 16$.

Канонический и распределительный методы строят в данном случае БГ с 12 путями. Поэтому $10 \leq S(f) \leq 12$. Таким образом, если перебор проводить, то нельзя построить бинарный граф, который будет содержать на три пути меньше, чем бинарные графы, построенные без перебора.

2. Сокращение перебора при поиске БГ, с минимальным числом вершин может обеспечить построение по ТИ булевой формулы в базисе $\{\&, \vee, !\}$ с минимизированным числом букв.

Если эта БФ содержит h букв, то, как было показано в разд. 15.1, формульный метод строит ЛБГ с h условными вершинами. Если при этом $h < B_{ym}$, то это позволяет снизить верхнюю оценку сложности реализации заданной БФУ. Таким образом, $n \leq B_y(f) \leq h$. Если $h = n$, то формульный метод строит ЛБГ, минимальный по числу вершин, а точный метод, изложенный в разд. 15.11, позволяет построить без перебора ЛБГ, который минимален также и по числу путей.

16.4.6. Модификация канонического метода

Эта модификация состоит в том, что предлагается [7] после построения канонической структуры с целью сокращения числа номеров (разнотипных кустов) вынести на нее числа, уменьшая значения, указанные в столбце значений ТИ, и отмечая точками неизменяющиеся значения. Это, в свою очередь, позволяет сократить число вершин в граф-схеме (ГС) по сравнению с их числом в бинарном графе, построенном каноническим методом. При этом ГС в от-

личие БГ содержит операторные вершины, помеченные арифметическими операциями.

Если в канонической структуре вынесенной оказывается константа C и если она размещается в корне дерева или является нулем, то в ГС ей соответствует ОВ с пометкой $f = C$. Если вынесенная константа размещается не в корне КС и не является нулем, то в ГС ей соответствует ОВ с пометкой $f = f + C$. При этом отметим, что константа C может быть и отрицательным числом.

Двигаясь по построенной ГС сверху вниз, пометки некоторых ОВ могут быть упрощены. Метод строит граф-схемы, в которых на каждом пути от входа к выходу каждая входная переменная, помечающая условные вершины, встречается не более одного раза, а операторные вершины (в том числе и однотипные) могут встречаться на одном пути многократно.

Пример 16.32. Построить СГС для булевой функции

$$f(x_1, x_2, x_3) = |0000 \ 0111|^T = x_1(x_2 \vee x_3).$$

Используя формульный метод, для заданной БФ может быть построен ЛБГ с пятью вершинами. Применяя канонический метод, можно построить СБГ с семью вершинами. На рис. 16.60 показано построение с помощью предлагаемой модификации канонического метода линеаризованной структурированной граф-схемы (ЛСГС) с шестью вершинами.

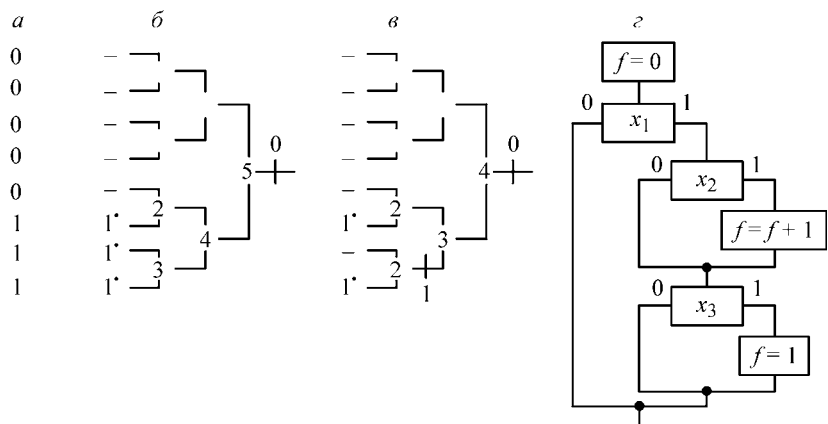


Рис. 16.60

В построенной граф-схеме пометка операторной вершины $f = f + 1$ может быть заменена на пометку $f = 1$.

Построенная ЛСГС может быть изоморфно запрограммирована с применением одноадресных условных переходов. Однако, так как

ЛСГС имеет длинный переход, она не может быть изоморфно запрограммирована на языке инструкций некоторых программируемых логических контроллеров, например [18], который позволяет изоморфно реализовывать только ЛСГС с короткими переходами.

Пример 16.33. Построить ЛСГС с короткими переходами для БФУ, рассмотренной в предыдущем примере.

На рис. 16.61 показано построение с помощью модификации канонического метода ЛСГС с короткими переходами, также содержащей шесть вершин.

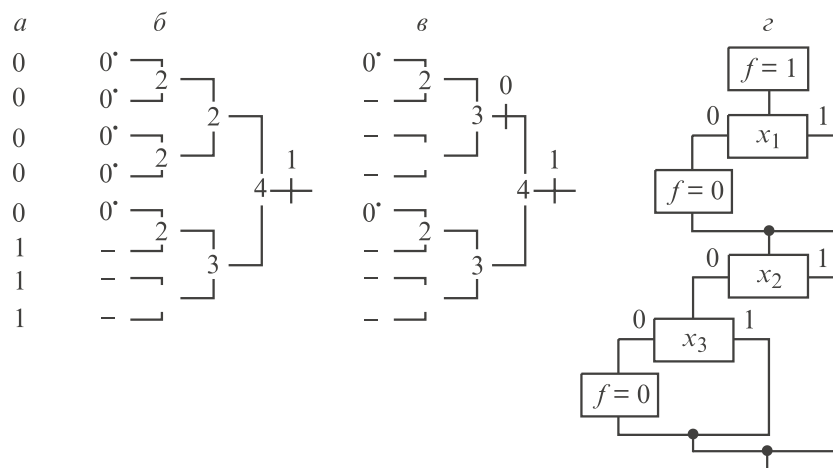


Рис. 16.61

Реализуем построенную ЛСГС программой на языке инструкций [18], записанной в два столбца:

```
EQ SO f    IF NI x2
IF NI x1   IF NI x3
EQ RO f    EQ RO f
CONT       CONT
           END
```

Из приведенного примера следует, что число команд в этой программе равно суммарному числу компонент (УВ, ОВ и точки объединения дуг) в ЛСГС с короткими переходами.

Пример 16.34. Построить ЛСГС с короткими переходами для БФУ, рассмотренной в примере 16.32, в которой изменен порядок входных переменных в таблице истинности:

$$f(x_3, x_2, x_1) = |0001\ 0101|^T = (x_3 \vee x_2)x_1.$$

На рис. 16.62 показано построение с помощью модификации канонического метода ЛСГС с короткими переходами, которая преобразована в ЛСБГ (рис. 16.63).

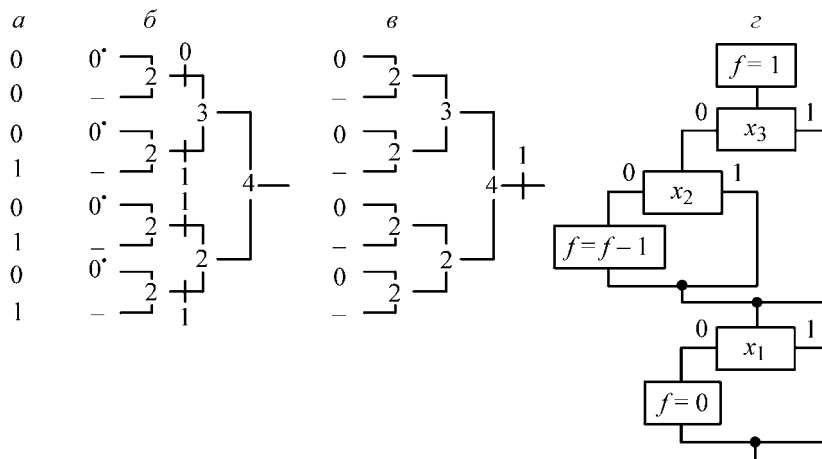


Рис. 16.62

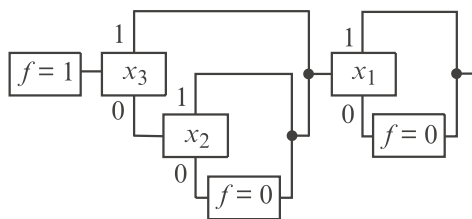


Рис. 16.63

16.5. Методы построения бинарных графов для систем булевых функций

Ветвящиеся граф-схемы разделим на два класса:
 — без арифметических операций;
 — с арифметическими операциями.

Ветвящиеся ГС могут быть классифицированы и по другому признаку — по расположению операторных вершин в них. Если эти вершины располагаются только после условных вершин, то такие ГС называются простыми, в противном случае — обобщенными [16].

В настоящем разделе рассматриваются ГС без арифметических операций, а в разд. 16.6 — с таковыми.

16.5.1. Метод построения простых бинарных графов

Изложим еще одну модификацию канонического метода, предназначенную для построения бинарных графов для систем БФУ (СБФУ). Эта модификация состоит в следующем:

- по таблице истинности СБФУ строится таблица истинности одной арифметической функции (АФ), заменяя каждый набор значений N заданных функций их десятичным эквивалентом;
- арифметическая функция реализуется каноническим методом;
- в построенной ГС каждая ОВ заменяется N операторными вершинами с двоичными операторами присваивания;
- в построенном БГ выполняется минимизация числа ОВ.

Пример 16.35. Реализовать параллельный одноразрядный двоичный сумматор, заданный табл. 16.7.

Т а б л и ц а 16.7

x_1	x_2	x_3	f_1	f_2	F
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	2
1	1	0	1	0	2
1	1	1	1	1	3

Реализуем АФ каноническим методом (рис. 16.64).

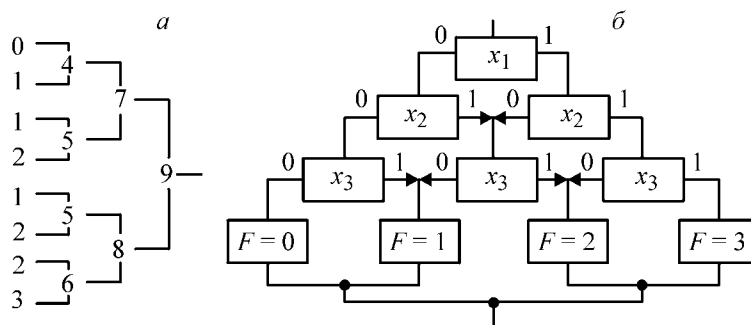


Рис. 16.64

Заменяя в этой ГС операторные вершины, получим искомый бинарный граф с 14 вершинами (рис. 16.65). Сокращение числа вершин до 12 достигается за счет объединения и изменения порядка расположения одинаковых операторных вершин (рис. 16.66). Изложенный метод впервые описан [19].

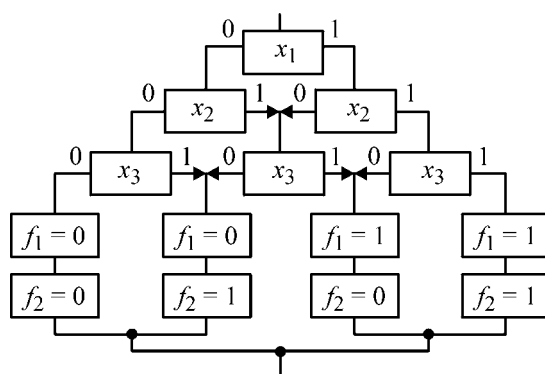


Рис. 16.65

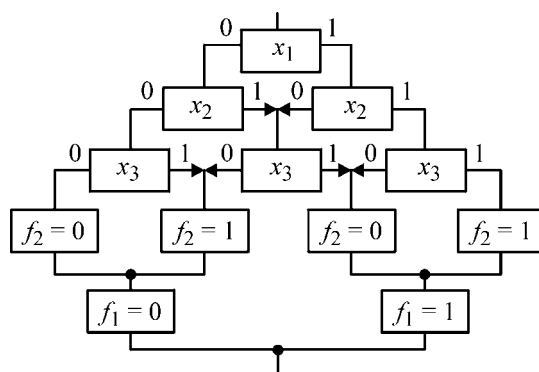


Рис. 16.66

16.5.2. Метод последовательного соединения простых бинарных графов

Этот метод состоит в построении простых бинарных графов для каждой БФУ системы и в последовательном их соединении. Получающийся в результате БГ является обобщенным.

Пример 16.36. Реализовать параллельный одноразрядный двоичный сумматор предлагаемым методом.

Обобщенный бинарный граф, построенный этим методом и содержащий 13 вершин, приведен на рис. 16.67.

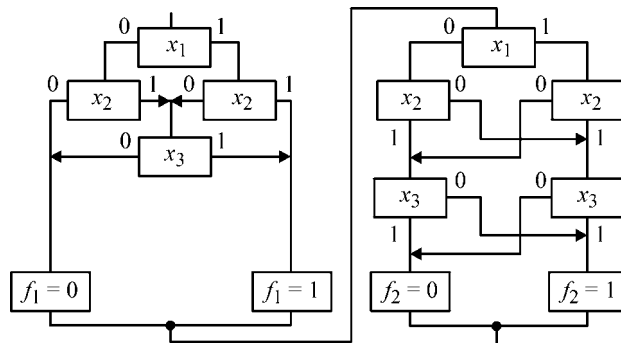


Рис. 16.67

Этот метод впервые описан в [20].

16.5.3. Метод построения обобщенных бинарных графов

Метод состоит в следующем. Построим каноническую структуру и вынесем на нее для отдельных функций рядом расположенные одинаковые значения, входящие в кусты любого уровня, с целью повышения однотипности кустов. По такой структуре изоморфно строится бинарный граф, который после этого минимизируется за счет объединения соответствующих ОВ нижнего уровня. Дальнейшее сокращение числа вершин может обеспечиваться за счет подъема вверх некоторых операторных вершин.

Пример 16.37. Реализовать параллельный одноразрядный двоичный сумматор предлагаемым методом.

Каноническая структура с вынесенными на нее операторными вершинами приведена на рис. 16.68.

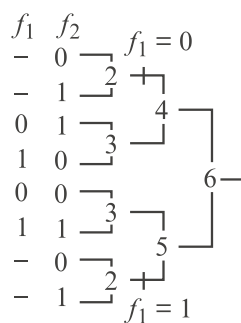


Рис. 16.68

Обобщенный БГ, построенный по этой структуре и содержащий 11 вершин, приведен на рис. 16.69. Дальнейшее сокращение числа вершин в БГ достигается за счет подъема вверх ОВ с пометкой $f_1 = 0$ (рис. 16.70). Этот граф содержит 10 вершин.

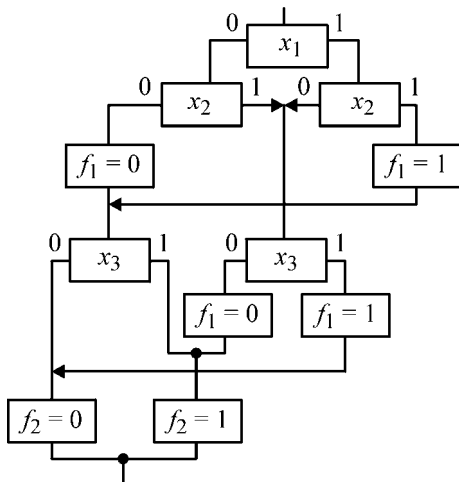


Рис. 16.69

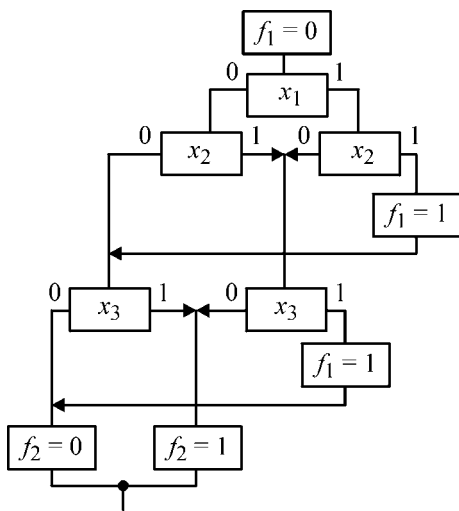


Рис. 16.70

Изложенный метод впервые описан в [21].

16.5.4. Метод независимых фрагментов

Этот метод применительно к реализации одной булевой формулы изложен в разд. 16.3. Расширим этот метод применительно к системам булевых формул (СБФ) [10, 11]:

— каждая из N булевых формул системы записывается в дизъюнктивной нормальной форме (ДНФ);

— каждая конъюнкция, входящая в i -ю БФ, помечается символом выходной переменной f_i , где $i = 1, \dots, N$. Если конъюнкция входит в m формул, то она помечается m указанными символами;

— все символы f_i и все помеченные конъюнкции объединяются в оператор структуры (ОС) с помощью символа $|$;

— ОС минимизируется по числу букв за счет выноса влево за скобки произвольной глубины одинаковых входных переменных x_j , входящих в разные конъюнкции, если такая возможность имеется;

— по минимизированному ОС (МОС) изоморфно строится ЛСБГ или СБГ, используя правила, изложенные в разд. 16.3.

Так же как и для одной БФ, в данном случае допускается, чтобы для некоторых БФ системы, например f_i и f_k , в ОС наряду со всеми конъюнкциями ДНФ остальных формул системы объединялись также все конъюнкции ДНФ инверсий формул f_i и f_k , помеченных символами $!f_i$ и $!f_k$ соответственно. В качестве символов начальной установки для этих БФ применяются символы f_i и f_k .

Если ОС содержит h_1 букв, то по нему может быть построен ЛСБГ с короткими переходами, содержащий h_1 вершин.

Если МОС содержит h_1 букв, то по нему может быть построен ЛСБГ с длинными переходами, содержащий h_1 вершин.

Если оператор содержит квадратные скобки и h_1 букв, то по нему может быть построен СБГ, содержащий менее чем h_1 вершин.

Пример 16.38. Реализовать СБФ $f_1 = x_1 \vee x_2 \vee x_3$, $f_2 = (!x_1!x_2 \vee \vee x_3)x_4$.

Построим оператор структуры:

$$F = !f_1!f_2|x_1f_1|x_2f_1|x_3f_1|!x_1!x_2x_4f_2|x_3x_4f_2.$$

Этот оператор состоит из 15 букв и может быть реализован ЛСБГ с короткими переходами, содержащим 15 вершин. В данном случае существуют два МОС, содержащих по 14 букв:

$$F = !f_1!f_2|x_1f_1|x_2f_1|x_3(f_1|x_4f_2)|!x_1!x_2x_4f_2;$$

$$F = !f_1!f_2|x_1f_1|x_2f_1|x_3f_1|x_4(!x_1!x_2x_4f_2|x_3f_2).$$

Каждый из этих операторов реализуется ЛСБГ с длинными переходами, содержащим 14 вершин. Если в первый МОС ввести квадратные скобки:

$$F = !f_1!f_2|[!x_1!x_2x_4f_2 \vee x_1f_1]|x_2f_1|x_3(f_1|x_4f_2),$$

то по нему может быть построен СБГ с 13 вершинами (рис. 16.71).

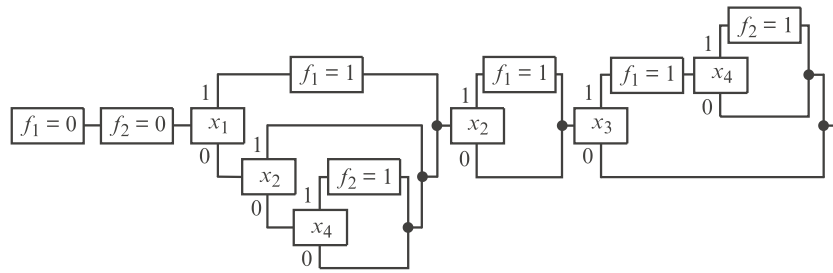


Рис. 16.71

Если первую формулу заданной системы заменить на инверсию, то

$$F = f_1 | !f_2 | !x_1 !x_2 !x_3 !f_1 | !x_1 !x_2 x_4 f_2 | x_3 x_4 f_2.$$

Этот оператор структуры, содержащий 13 букв, реализуется ЛСБГ с короткими переходами, состоящим из 13 вершин (рис. 16.72).

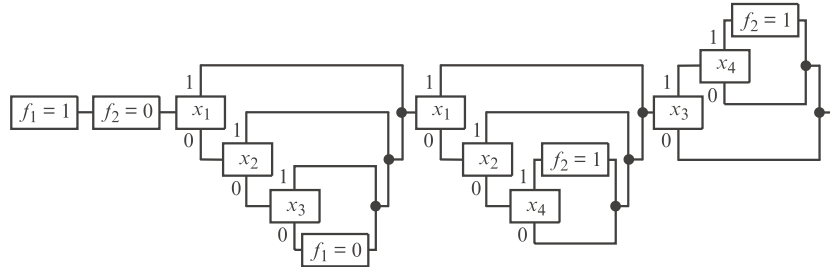


Рис. 16.72

Минимизируя полученный оператор по числу букв, получим МОС, содержащий 11 букв:

$$F = f_1 | !f_2 | !x_1 !x_2 (!x_3 !f_1 | x_4 f_2) | x_3 x_4 f_2.$$

Этот оператор реализуется ЛСБГ с длинными переходами, содержащим 11 вершин (рис. 16.73).

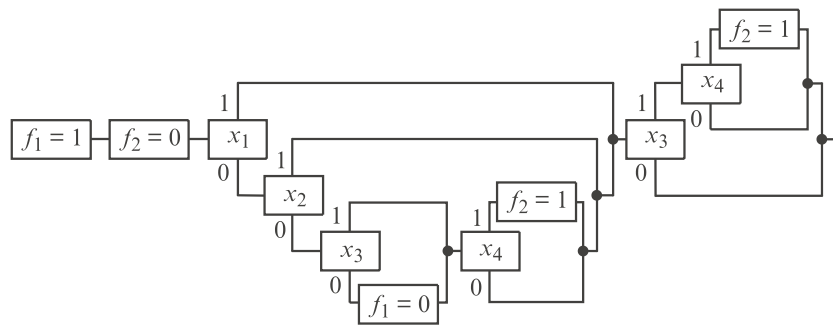


Рис. 16.73

16.5.5. Метод зависимых фрагментов

Для каждой булевой формулы, входящей в систему, с помощью метода зависимых фрагментов, изложенного в разд. 16.2, строится ЛСБГ. Для каждого ЛСБГ строится изоморфный оператор структуры (ОС) по правилам, используемым в методе независимых фрагментов (разд. 16.3).

Объединение построенных операторов осуществляется с учетом следующих особенностей:

— блоки с одним входом и одним выходом, соответствующие одной выходной переменной, не влияют на вычисление значений других выходных переменных;

— в объединенном ОС должен быть сохранен порядок относительного расположения фрагментов в каждом операторе;

— объединенный ОС может быть минимизирован за счет вынесения одинаковых (если они имеются) входных переменных влево за скобки.

Пример 16.39. Определить число вершин в ЛСБГ, реализующем СБФ:

$$f_1 = (x_1 \vee x_2 \vee x_3)x_4x_5; \quad f_2 = x_4x_5x_6 \vee x_7x_8.$$

Используя метод зависимых фрагментов, для каждой заданной БФ построим ЛСБГ и запишем по нему оператор структуры:

$$F_1 = !f_1|x_4x_5f_1|!x_1!x_2!x_3!f_1; \quad F_2 = !f_2|x_4x_5x_6f_2|x_7x_8f_2.$$

Построим объединенный ОС и проминимизируем его:

$$\begin{aligned} F &= !f_1|x_4x_5f_1|!x_1!x_2!x_3!f_1|!f_2|x_4x_5x_6f_2|x_7x_8f_2 = \\ &= !f_1|!f_2|x_4x_5(f_1|x_6f_2)|!x_1!x_2!x_3!f_1|x_7x_8f_2. \end{aligned}$$

Этот оператор содержит 14 букв и изоморфно реализуется ЛСБГ, состоящим из 14 вершин.

Отметим, что метод независимых фрагментов строит в данном случае оператор структуры

$$F = !f_1|!f_2|x_4x_5(x_1f_1|x_2f_1|x_3f_1|x_6f_2)|x_7x_8f_2$$

из 15 букв, который реализуется ЛСБГ, содержащим 15 вершин.

В заключение разд. 16.5 покажем, что существуют системы, состоящие только из двух БФУ, зависящих от одних и тех же переменных, совместная реализация которых с помощью модификации канонического метода вне зависимости от порядка переменных в ТИ приводит к построению бинарных графов с экспоненциальной (относительно числа переменных) оценкой числа вершин в них.

При этом отметим, что при применении этого метода сложность бинарного графа, реализующего две БФУ, не может быть меньше

сложности бинарного графа с наибольшим числом вершин, реализующего одну из этих функций, так как в противном случае эта функция могла бы быть реализована за счет исключения из бинарного графа, построенного для системы, операторных вершин, соответствующих второй функции.

Системы указанного типа формируются, если при фиксированном n в качестве первой выбрать функцию из первого класса функций, приведенных в конце разд. 15, а в качестве второй — функцию из второго класса приведенных там же функций [35]. Для этой системы функций изменение порядка переменных в их таблице истинности в общем случае не позволяет упростить совместную реализацию.

При этом отметим, что метод независимых фрагментов строит для таких систем структурированные бинарные графы с линейной оценкой сложности.

В общем случае вопрос о выборе метода эффективной реализации систем БФУ остается открытым.

16.6. Методы построения арифметических граф-схем для систем булевых функций

16.6.1. Реализация одной булевой функции

Известно [22, 23], что произвольная булева функция n переменных может быть реализована арифметическим полиномом (АП), содержащим не более 2^n ненулевых коэффициентов, который, в свою очередь, всегда может быть реализован линеаризованной СГС (ЛСГС) с короткими переходами, в которой в операторных вершинах выполняются арифметические операции вида: $f = f + a$; $f = f - a$, где a — ненулевой коэффициент АП.

Пример 16.40. Реализовать булеву формулу $f = x_1(x_2 \vee x_3)$.

Используя результаты [23], построим АП для заданной булевой формулы:

$$\begin{aligned} f &= x_1(x_2 \vee x_3) = x_1(x_2 + (1 - x_2)x_3) = \\ &= x_1x_2 + x_1x_3 - x_1x_2x_3. \end{aligned}$$

Этот АП реализуется ЛСГС, представленной на рис. 16.74.

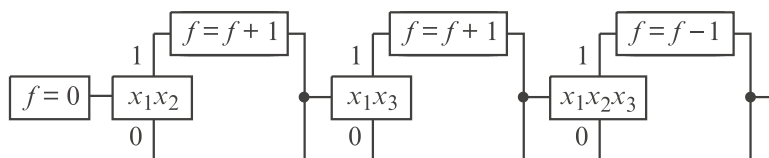


Рис. 16.74

Если АП проминимизировать, то по полученному арифметическому выражению

$$f = x_1(x_2 + x_3 - x_2x_3)$$

может быть построена упрощенная ЛСГС (рис. 16.75).

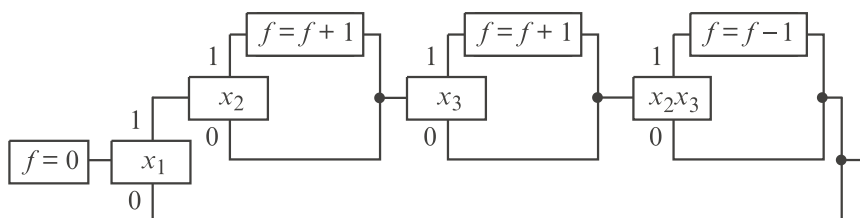


Рис. 16.75

Рассмотренный метод является своеобразным методом независимых фрагментов, так как в построенной с его помощью ЛСГС последовательно соединенные блоки, содержащие условные вершины, могут переставляться.

16.6.2. Метод вынесения операторных вершин

Заменяем заданную систему булевых функций n переменных одной арифметической функцией $F(x_1, \dots, x_n)$.

Применим к этой функции модификацию канонического метода, рассмотренного в разд. 16.4.6 [7]. Для получения по вычисленному значению кортежа битов в конце ГС должна размещаться операторная вершина, выполняющая операцию «десятичное число — двоичное число», которая обозначается символом «bin F». В построенных ниже граф-схемах эта вершина умалчивается.

Пример 16.41. Реализовать арифметическую функцию (АФ)

$$F(x_1, x_2, x_3) = |2 \ 3 \ 2 \ 4 \ 1 \ 3 \ 2 \ 3|^T.$$

Вынесем на каноническую структуру (КС) числа, на которые могут быть уменьшены пары значений АФ, объединенные кустами нижнего уровня, а затем вынесем число 2 в корень структуры, исключив и заменив соответствующие пометки в первоначально построенной канонической структуре (рис. 16.76).

Построим по полученной канонической структуре изоморфную ей граф-схему (рис. 16.77) и упростим в последней пометки операторных вершин, просматривая ее сверху вниз (рис. 16.78).

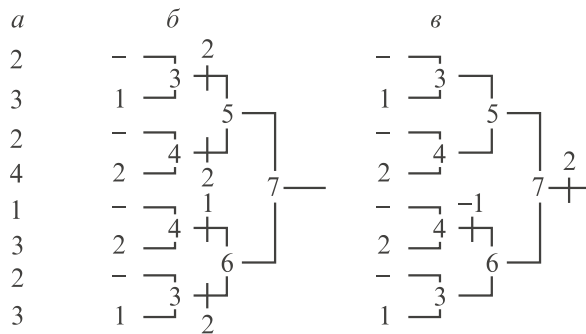


Рис. 16.76

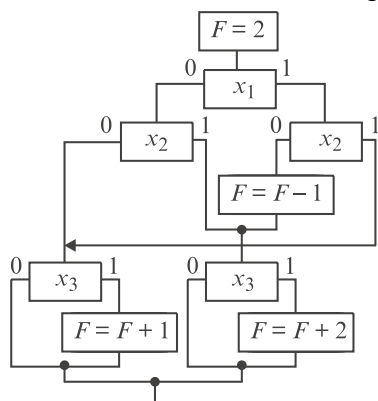


Рис. 16.77

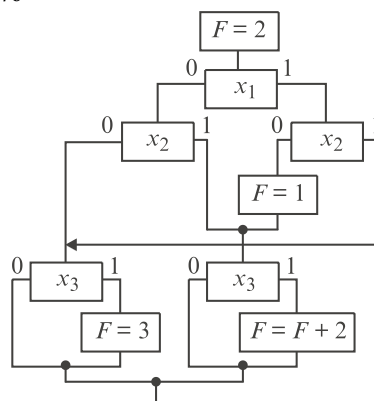


Рис. 16.78

Пример 16.42. Реализовать арифметическую функцию

$$F(x_1, x_2, x_3) = |0\ 2\ 0\ 2\ 2\ 4\ 1\ 3|^T.$$

Уменьшим все значения АФ (кроме нулевых) на число два, вынесем его в корень канонической структуры, синтезированной по полученной функции, и построим по ней СГС (рис. 16.79).

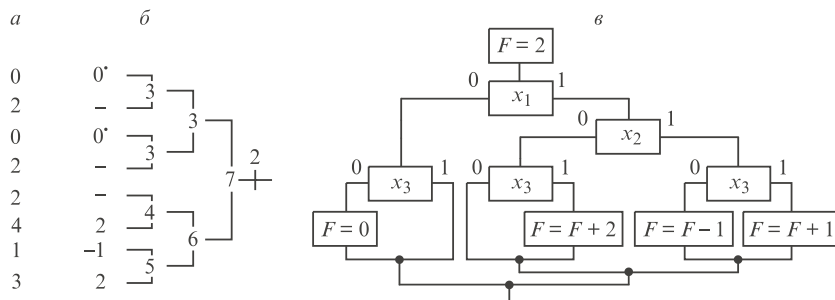


Рис. 16.79

В этой ГС могут быть выполнены следующие замены пометок ОВ: $F + 2$ на $F = 4$, $F - 1$ на $F = 1$, $F + 1$ на $F = 3$.

Покажем, что при другом выносе чисел на КС изоморфная ей ГС по сравнению с предыдущей упрощается (рис. 16.80).

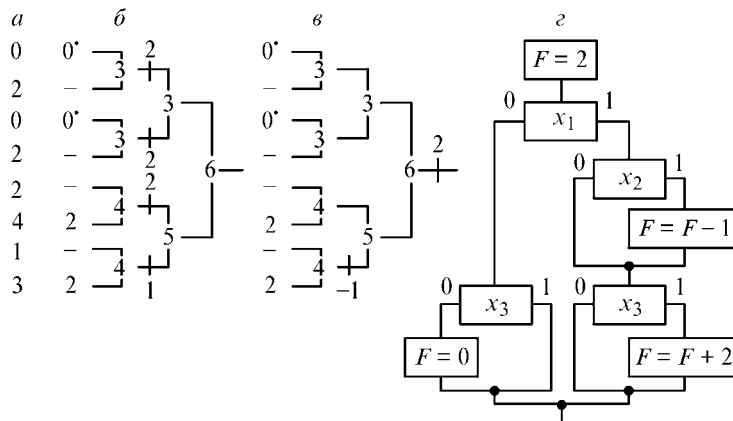


Рис. 16.80

В этой ГС может быть выполнена следующая замена пометки операторной вершины: $F - 1$ на $F = 1$.

Пример 16.43. Реализовать параллельный двоичный одноразрядный сумматор

$$F(x_1, x_2, x_3) = |0 \ 1 \ 1 \ 2 \ 1 \ 2 \ 2 \ 3|^T.$$

На рис. 16.81 показан процесс оптимизации КС, а на рис. 16.82 приведена ЛСГС, соответствующая оптимизированной структуре.

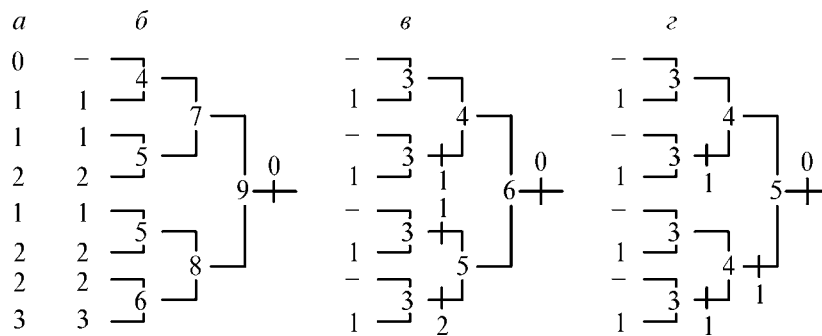


Рис. 16.81

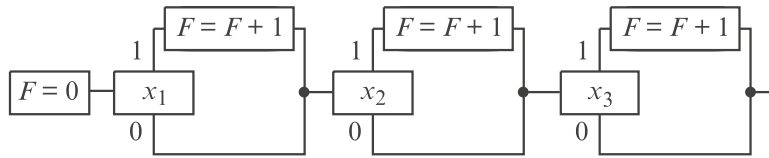


Рис. 16.82

В построенной ГС в крайней левой операторной вершине пометка $F = F + 1$ может быть заменена на пометку $F = 1$.

Сформулируем утверждение: если ЛСГС состоит только из последовательного соединения операторной вершины с пометкой « $F = C_0$ » и n блоков «неполный выбор», операторные вершины в которых имеют пометки « $F = F + C_1$ », ..., « $F = F + C_n$ » соответственно, то СБФ, соответствующая этой граф-схеме, может быть реализована линейным арифметическим полиномом (ЛАП) вида: $F = C_0 + C_1 x_1 + \dots + C_n x_n$.

Для рассмотренного примера $F = x_1 + x_2 + x_3$.

Пример 16.44. Реализовать арифметическую функцию

$$F(x_1, x_2, x_3) = |0 \ 2 \ 4 \ 6 \ 3 \ 5 \ 7 \ 9|^T.$$

Процесс построения канонической структуры изображен на рис. 16.83, а на рис. 16.84 приведена ЛСГС, построенная по этой структуре.

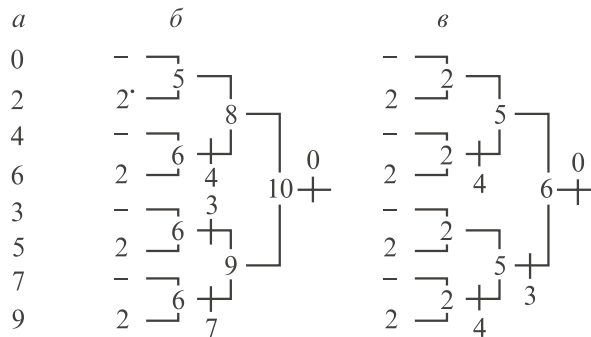


Рис. 16.83

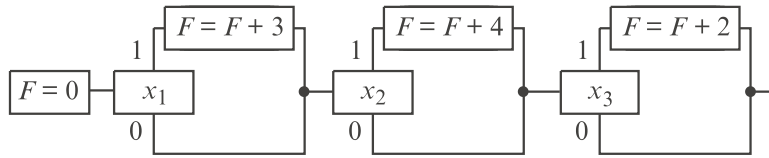


Рис. 16.84

В построенной ГС пометка операторной вершины « $F = F + 3$ » может быть заменена на пометку « $F = 3$ ». Из изложенного следует, что $F = 3x_1 + 4x_2 + 2x_3$.

Таким образом, можно утверждать, что изложена новая трактовка условия линейности арифметического полинома, предложенного в [23].

Если построенная СГС имеет структуру, отличную от описанной выше, то система БФУ реализуется нелинейным АП.

Пример 16.45. Реализовать арифметическую функцию

$$F(x_1, x_2, x_3) = |2 \ 2 \ 2 \ 2 \ 1 \ 2 \ 2 \ 3|^T.$$

Процесс построения канонической структуры изображен на рис. 16.85, а на рис. 16.86 приведена ЛСГС, построенная по этой КС.

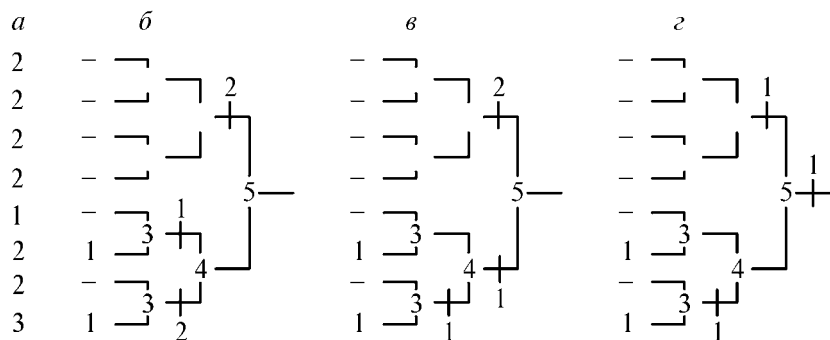


Рис. 16.85

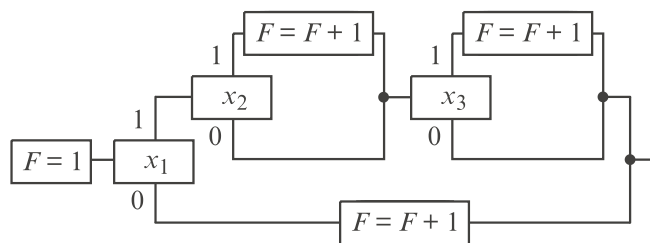


Рис. 16.86

В построенной ГС в двух левых операторных вершинах пометка « $F = F + 1$ » может быть заменена на пометку « $F = 1$ ». Построим арифметическое выражение по этой СГС:

$$F = 1 + 1(1 - x_1) + x_1(1x_2 + 1x_3) = 2 + x_1(1x_2 + 1x_3 - 1).$$

Построим СГС по этому выражению (рис. 16.87).

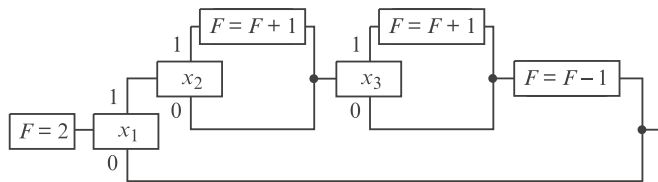


Рис. 16.87

В построенной ГС в левой операторной вершине пометка « $F = F + 1$ » может быть заменена на пометку « $F = 1$ ». Преобразуем полученное выражение в АП:

$$F = 2 - x_1 + x_1 x_2 + x_1 x_3.$$

Этому полиному соответствует ЛСГС, представленная на рис. 16.88.

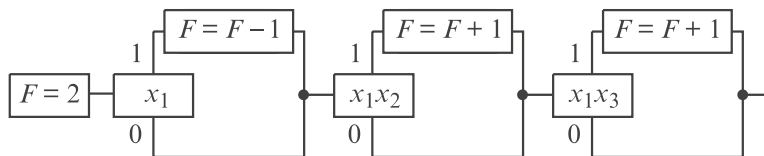


Рис. 16.88

В построенной ГС пометка операторной вершины « $F = F - 1$ » может быть заменена на пометку « $F = 1$ ».

В заключение раздела отметим, что предложенный подход [7] может рассматриваться в качестве графического метода построения арифметических полиномов, что отличает его от подходов, описанных в гл. 20 и [22, 23].

16.6.3. Метод взвешивания функций

Каждой из N функций системы может быть присвоен вес от 2^{N-1} до 2^0 .

После этого система БФУ может быть реализована ЛСГС с короткими переходами, представленной на рис. 16.89.

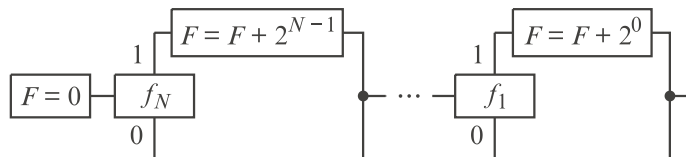


Рис. 16.89

Для одноразрядного сумматора $N=2$, $f_2 = x_1 \# x_2 \# x_3$, $f_1 = x_1 \oplus x_2 \oplus x_3$.

16.6.4. Стандартная реализация строк таблицы истинности

Предположим, что F_i — значение арифметической функции F на входном наборе i . При этом, например, для $n = 2$ функция F может быть реализована стандартными схемами на рис. 16.90—16.92.

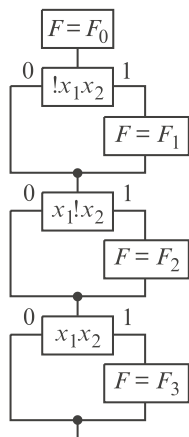


Рис. 16.90

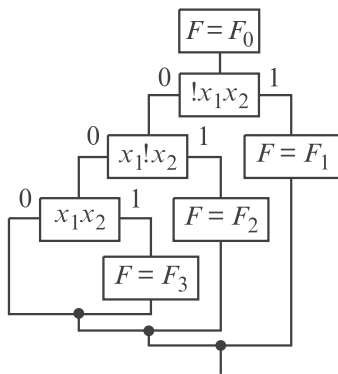


Рис. 16.91

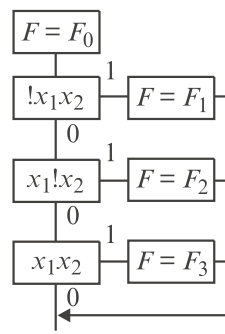


Рис. 16.92

Можно показать, что стандартной также является СГС на рис. 16.93, которая проще СГС, построенной по арифметическому полиному (рис. 16.94).

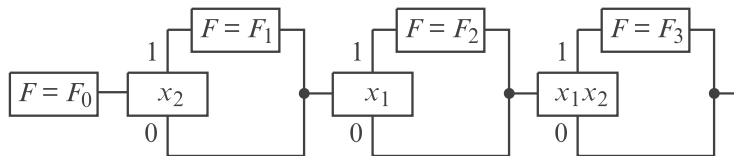


Рис. 16.93

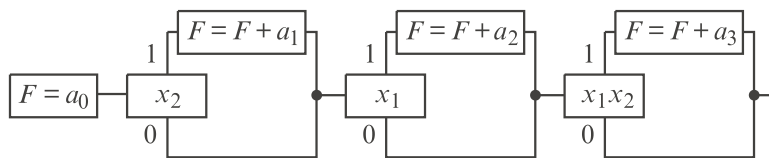


Рис. 16.94

16.6.5. Реализация типов значений арифметических функций

Предлагаемый метод строит ЛСГС (с короткими переходами), которые состоят из операторных вершин с пометкой « $F = F_j$ » и $R - 1$ блоков «неполный выбор», в каждом из которых в УВ записана булева формула g_i , осуществляющая выбор значения F_i , а ОВ в этом блоке имеет пометку « $F = F_i$ ». Здесь R — число типов значений АФ. Так как булевы формулы в ОВ ортогональны, то блоки в построенной таким образом ГС могут переставляться.

Пример 16.46. Реализовать параллельный двоичный одноразрядный сумматор $F(x_1, x_2, x_3) = |0\ 1\ 1\ 2\ 1\ 2\ 2\ 3|^T$.

Формула, выбирающая значение $F = 1$, имеет вид:

$$g_1(x_1, x_2, x_3) = |0110\ 1000|^T = !x_1(!x_2 \vee !x_3) \oplus !x_2!x_3.$$

Формула, выбирающая значение $F = 2$, имеет вид:

$$g_2(x_1, x_2, x_3) = |0001\ 0110|^T = x_1(x_2 \vee x_3) \oplus x_2x_3.$$

Формула, выбирающая значение $F = 3$, имеет вид:

$$g_3(x_1, x_2, x_3) = |0000\ 0001|^T = x_1x_2x_3.$$

На рис. 16.95 приведена ЛСГС, построенная предложенным методом.

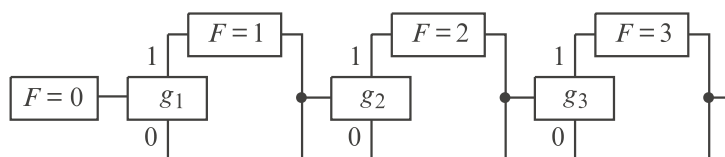


Рис. 16.95

Приведенная реализация неэффективна, так как суммарная сложность булевых формул g_1, g_2, g_3 выше суммарной сложности булевых формул f_2, f_1 , непосредственно описывающих сумматор.

Приведем пример реализации СБФ, для которой этот метод весьма эффективен.

Пример 16.47. Реализовать арифметическую функцию

$$F(x_1, x_2, x_3) = |0\ 7\ 12\ 0\ 21\ 0\ 0\ 0|^T.$$

В этом случае для $F = 7$ $g_1 = !x_1!x_2x_3$, для $F = 12$ $g_2 = !x_1x_2!x_3$, для $F = 21$ $g_3 = x_1!x_2!x_3$, что приводит к эффективной реализации (рис. 16.96), так как арифметическая функция в данном случае задает весьма сложную СБФ:

$$f_5 = x_1!x_2!x_3, f_4 = !x_1x_2!x_3, f_3 = !x_1(x_2 \oplus x_3) \vee x_1!x_2!x_3, \\ f_2 = !x_1!x_2x_3, f_1 = (x_1 \oplus x_3)!x_2.$$

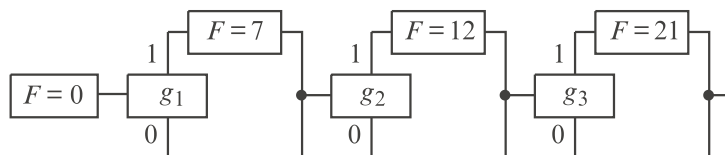


Рис. 16.96

16.6.6. Упрощение булевых формул, осуществляющих выбор типов значений арифметических функций

Ниже излагается метод построения СГС с $R - 1$ условными и R операторными вершинами, в которых сложность булевых формул, помечающих условные вершины, не превышает сложности соответствующих булевых формул, которые строятся методом, изложенным в предыдущем разделе.

Метод основан на том, что после формирования каждой булевой формулы, осуществляющей выбор однотипных значений арифметической функции, эти значения исключаются из таблицы истинности, повышая тем самым степень недоопределенности заданной функции. Это позволяет при формировании следующих булевых формул так доопределять столбец значений таблицы истинности, чтобы булевы формулы, осуществляющие выбор однотипных значений, получались максимально простыми. При использовании этого метода каждый тип значений арифметической функции вычисляется однократно.

Этот метод позволяет строить СГС как от входа к выходу, так и в обратном порядке. Получающиеся при этом СГС имеют разнотипные структуры.

Пример 16.48. Реализовать параллельный двоичный одноразрядный сумматор, строя СГС от входа к выходу.

Выберем значение $F = 3$ с помощью БФ $g_1 = x_1x_2x_3$. При этом $F = |0\ 1\ 1\ 2\ 1\ 2\ 2\ 2|{}^T$. Доопределим эту функцию следующим образом: $F = |0\ 1\ 1\ 2\ 1\ 2\ 2\ 2|{}^T$ и выберем значение $F = 2$ с помо-

щью БФ $g_2 = x_1 \# x_2 \# x_3$. После этого АФ приобретает вид $F = |0 \ 1 \ 1 \ - \ 1 \ - \ - \ -|^T$. Доопределим АФ следующим образом: $F = |0 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1|^T$. Выбирая $F = 1$ с помощью формулы $g_3 = x_1 \vee x_2 \vee x_3$, завершаем построение СГС (рис. 16.97).

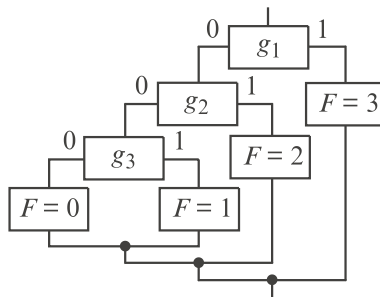


Рис. 16.97

Таким образом, две булевы формулы — f_2 и f_1 , описывающие одноразрядный сумматор, реализуются с применением трех булевых формул, причем $f_2 = g_2$.

Пример 16.49. Реализовать параллельный двоичный одноразрядный сумматор, строя СГС от выхода к входу.

Выберем значение $F = 3$ с помощью БФ $g_{31} = x_1 x_2 x_3$. При этом $F = |0 \ 1 \ 1 \ 2 \ 1 \ 2 \ 2 \ -|^T$. Доопределим эту функцию следующим образом: $F = |0 \ 1 \ 1 \ 2 \ 1 \ 2 \ 2 \ 2|^T$. Выберем значение $F = 1$ с помощью формулы $g_{21} = x_1 \oplus x_2 \oplus x_3$. После этого АФ приобретает вид $F = |0 \ - \ - \ 2 \ - \ 2 \ 2 \ -|^T$. Доопределим АФ следующим образом: $F = |0 \ 2 \ 2 \ 2 \ 0 \ 2 \ 2 \ 2|^T$. Выбирая $F = 2$ с помощью $g_{11} = x_2 \vee x_3$, завершаем построение СГС (рис. 16.98).

Таким образом, две булевы формулы — f_2 и f_1 , описывающие одноразрядный сумматор, реализуются с использованием трех булевых формул, причем $f_1 = g_{21}$.

Изложенный метод впервые описан в [7].

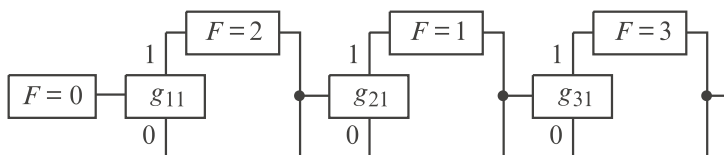


Рис. 16.98

16.6.7. Реализация с многократным вычислением однотипных значений арифметических функций

Если при применении метода, рассмотренного в предыдущем разделе, снять ограничение на однократность вычисления каждого типа значений АФ, то за счет увеличения числа вершин в СГС, количество которых априори становится неизвестным, появляется возможность эффективной реализации некоторых функций.

Пример 16.50. Реализовать арифметическую функцию

$$F(x_1, x_2, x_3) = |2 \ 2 \ 1 \ 1 \ 2 \ 0 \ 0 \ 0|^T.$$

Реализуя эту СГС от входа к выходу, применим указанную выше модификацию метода, изложенного в предыдущем разделе. Построенная СГС приведена на рис. 16.99. При этом отметим, что АФ в данном случае соответствует СБФ: $f_2 = !x_1!x_2 \vee x_1!x_2!x_3$, $f_1 = !x_1 x_2$.

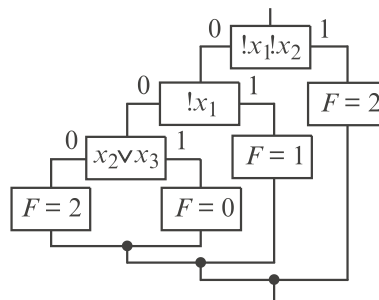


Рис. 16.99

16.7. Методы построения векторных граф-схем

Рассмотренные граф-схемы (ГС) обладали тем свойством, что в них в каждой условной вершине проверялись значения только одной входной переменной или одной булевой формулы.

Рассмотрим другой класс ГС, в которых в каждой условной вершине выполняется сравнение входного вектора с соответствующей маской (константой). Назовем такие ГС векторными (ВГС).

Некоторые методы векторных вычислений изложены в гл.18.

Опишем метод построения ВГС по таблице истинности [6]. При этом для компактности указанное выше сравнение будем проводить в десятичной форме.

Рассмотрим класс положительных монотонных неповторных пороговых формул (БПФ), каждая из которых может быть записана таким образом, что ее столбец значений будет иметь «точку переги-

ба», так как в нем сначала расположены только нули, а затем только единицы. Например, БПФ

$$f = x_1 \vee x_2 (x_3 \vee (x_4 (x_5 \vee x_6 (x_7 \vee x_8))))$$

на первых 85 входных наборах (с десятичными эквивалентами 0, ..., 84) принимает нулевые значения, а на остальных 171 входных наборах (с десятичными эквивалентами 85, ..., 255) — единичные.

Это позволяет при порядке переменных x_1, \dots, x_8 реализовать эту БПФ векторной ГС, содержащей только одну условную вершину:

$$\text{if } (X \geq 85) \text{ f} = 1 \text{ else f} = 0,$$

где X — десятичный эквивалент значений входного набора (вектора). При этом $X = 128x_1 + 64x_2 + 32x_3 + 16x_4 + 8x_5 + 4x_6 + 2x_7 + 1x_8$.

Такой ГС соответствует пороговая реализация вида:

$$f = \text{sign} (128x_1 + 64x_2 + 32x_3 + 16x_4 + 8x_5 + 4x_6 + 2x_7 + 1x_8 - 85).$$

Используя метод, изложенный в разд. 15.5, можно построить минимальную пороговую реализацию:

$$f = \text{sign} (21x_1 + 13x_2 + 8x_3 + 5x_4 + 3x_5 + 2x_6 + x_7 + x_8 - 21).$$

Несмотря на то что эта реализация имеет существенно меньшие значения весов и порога, она не может быть столь эффективно программно реализована, как это предложено выше.

Если порядок переменных во входном наборе не совпадает с порядком переменных, при котором БПФ имеет одну точку перегиба, то сложность ВГС увеличивается.

Пример 16.51. Реализовать БПФ $f = x_1 x_2 \vee x_3$.

При порядке переменных x_1, x_2, x_3 эта БПФ реализуется по диапазонам значений наборов входных переменных, так как

$$f(x_1, x_2, x_3) = |0101 \ 0111|^T.$$

При этом строится ВГС, представленная на рис. 16.100.

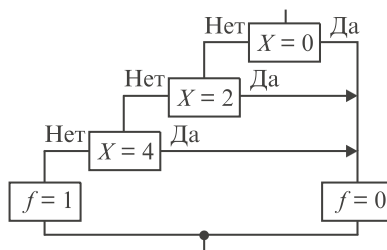


Рис. 16.100

При порядке переменных x_3, x_2, x_1 эта функция реализуется существенно проще:

$$\text{if } (X \geq 3) \text{ f} = 1 \text{ else f} = 0,$$

так как

$$f(x_3, x_2, x_1) = |0001 \ 1111|^T.$$

По диапазонам входных переменных может быть реализована и произвольная булева функция n переменных.

Пример 16.52. Реализовать булеву функцию

$$f(x_1, x_2, x_3, x_4) = |0001 \ 0001 \ 0001 \ 1111|^T = \\ = x_1 x_2 \vee x_3 x_4.$$

При порядке переменных x_1, x_2, x_3, x_4 эта БФУ реализуется ВГС, приведенной на рис. 16.101.

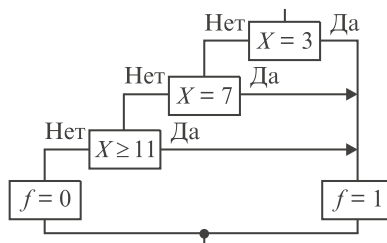


Рис. 16.101

Изложенный подход может быть использован и для реализации арифметических функций. Так, например, на рис. 16.102 приведена ВГС, реализующая параллельный двоичный одноразрядный сумматор

$$F(x_1, x_2, x_3) = |0 \ 1 \ 1 \ 2 \ 1 \ 2 \ 2 \ 3|^T$$

при порядке входных переменных x_1, x_2, x_3 .

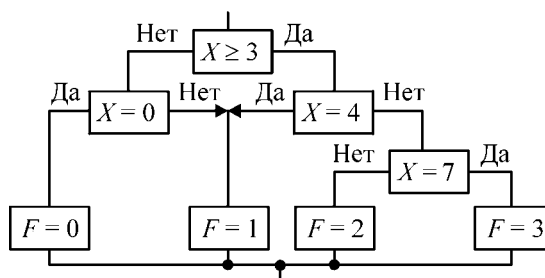


Рис. 16.102

Кроме работ, на которые в настоящей и предыдущей главах имеются ссылки, существует также и много других работ, посвященных бинарным графам, например [24—48].

16.8. Построение комбинационных схем из демультиплекторов

В последнее время бинарные графы все чаще применяются как при программной, так и при аппаратной реализации (в том числе и при проектировании сверхбольших интегральных схем на основе программируемой логики) в качестве средства представления отдельных БФУ (БФ) и систем БФУ (БФ) [30—43].

В разд. 5.1 изложены методы реализации бинарных графов (БГ), построенных для отдельных БФУ (БФ), изоморфными схемами из мультимплекторов «2 в 1». Эти методы основаны на рассмотрении каждого бинарного графа от его выходов к входу и поэтому не позволяют реализовать системы БФУ (БФ) совместно.

Ниже предлагается метод совместной схемной реализации указанных систем, основанный на рассмотрении бинарного графа, построенного для заданной системы с помощью методов, изложенных в разд. 16.5, от входа к выходам.

Метод базируется на применении демультиплекторов (DX) «1 в 2» (рис. 16.103), каждый из которых имеет два входа — b и x_i и два выхода — u_1 и u_0 , причем $u_1 = bx_i$, $u_0 = b!x_i$ [32, 49].

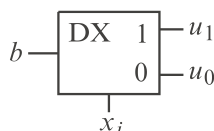


Рис. 16.103

Рассмотрим первоначально метод реализации N заданных функций системы и их инверсий по бинарному графу, который не содержит операторных вершин начальной установки. Этот метод состоит в следующем:

- каждая условная вершина бинарного графа заменяется демультиплектором «1 в 2» таким образом, что ее входу соответствует вход демультиплектора, а ее нулевому (единичному) выходу — нулевой (единичный) выход демультиплектора;
- каждая точка объединения дуг заменяется двухвходовым элементом ИЛИ;
- на b -вход первого демультиплектора подается константа «1», а на нижний вход i -го демультиплектора — входная переменная, указанная в i -й условной вершине графа;

— все одноименные операторные вершины $f_j = 1$ заменяются линиями, которые подключаются к элементу ИЛИ, на выходе которого реализуется соответствующая функция f_j ;

— все одноименные операторные вершины $f_j = 0$ заменяются линиями, которые подключаются к элементу ИЛИ, на выходе которого реализуется соответствующая функция $!f_j$.

В результате строится схема с $2N + 1$ выходами, на N из которых реализуются заданные функции, на N других выходах — их инверсии, а на последнем выходе — функция $f_k = 1$.

Число элементов в построенной таким образом схеме, изоморфной бинарному графу с точностью до реализации операторных вершин в нем, определяется соотношением:

$$n_3 = B_y + n_T + 2N,$$

где B_y — число условных вершин, n_T — число точек объединения дуг.

При реализации одной функции и ее инверсии справедливо соотношение

$$n_3 = B_y + n_T.$$

Построенная указанным образом схема может быть упрощена за счет исключения элемента ИЛИ, реализующего функцию f_k .

На рис. 16.104 приведена схема из шести элементов ($B_y = 3$, $n_T = 3$), реализующая формулы $f = x_1 x_2 \vee x_3$ и $!f = (!x_1 \vee !x_2)!x_3$, которая изоморфна ЛБГ, представленному на рис. 15.1.

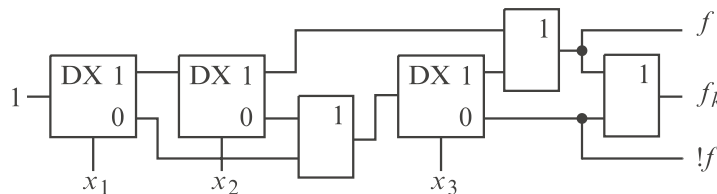


Рис. 16.104

После исключения последнего элемента ИЛИ полученная схема будет состоять из пяти элементов.

Так как в ЛБГ с одним входом и одним выходом при реализации формулы в базе $\{\&, \vee, !\}$ из h букв $B_y = h$, $n_T = h$, то число элементов в рассматриваемой схеме, одновременно реализующей формулу и ее инверсию, удовлетворяет соотношению $n_3 = 2h$. При этом число элементов в схеме не зависит от порядка записи формулы.

На рис. 16.105 приведена схема, также содержащая шесть элементов, реализующая формулу $f = x_3 \vee x_2 x_1$ и ее инверсию.

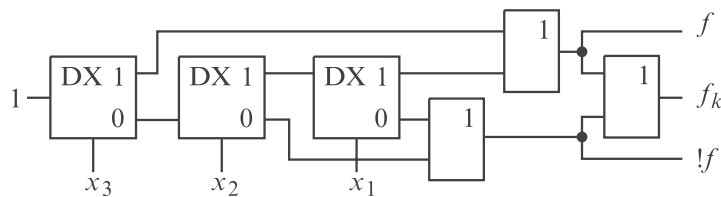


Рис. 16.105

После исключения последнего элемента ИЛИ полученная схема будет состоять из пяти элементов.

Приведенные схемы построены по ЛБГ. На рис. 16.106 приведены две схемы, построенные по бинарному графу (рис. 16.67), первая из которых реализует мажоритарную функцию трех переменных и ее инверсию, а вторая — линейную функцию трех переменных и ее инверсию.

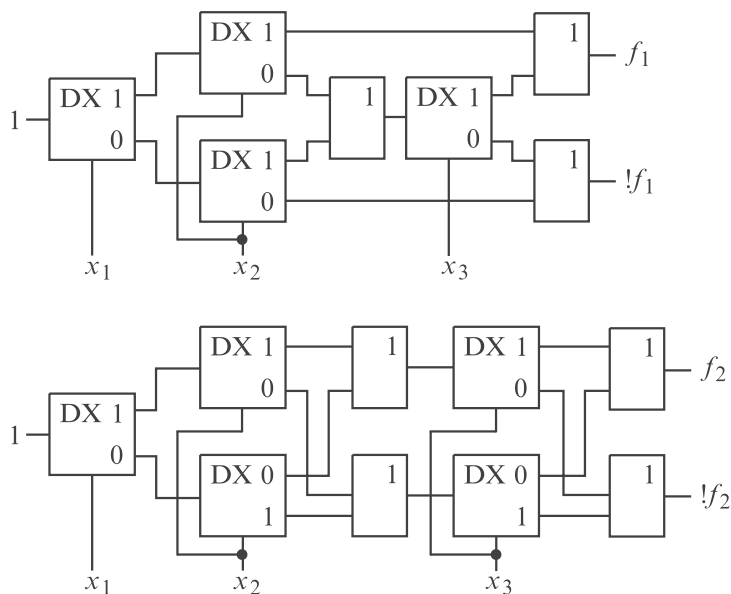


Рис. 16.106

Эти схемы реализованы независимо, несмотря на то что в БГ соответствующие им блоки соединены последовательно. Это объясняется тем, что если в БГ блоки с одним входом и одним выходом соединены последовательно и не «охватываются» другими блоками, то в схеме, изоморфной этому БГ, на выходе f_k каждой подсхемы, соответствующей блоку, формируется тождественная единица, что позволяет исключить элементы ИЛИ, связывающие эти блоки, подав единицы на входы каждой из образовавшихся при этом схем.

Эффективность предлагаемого метода повышается в том случае, когда схема, реализующая систему функций и их инверсии, строится по совместно минимизированному бинарному графу. На рис. 16.107 приведена схема из 11 элементов, построенная по бинарному графу (рис. 16.69), в котором исключен элемент ИЛИ, реализующий функцию f_k .

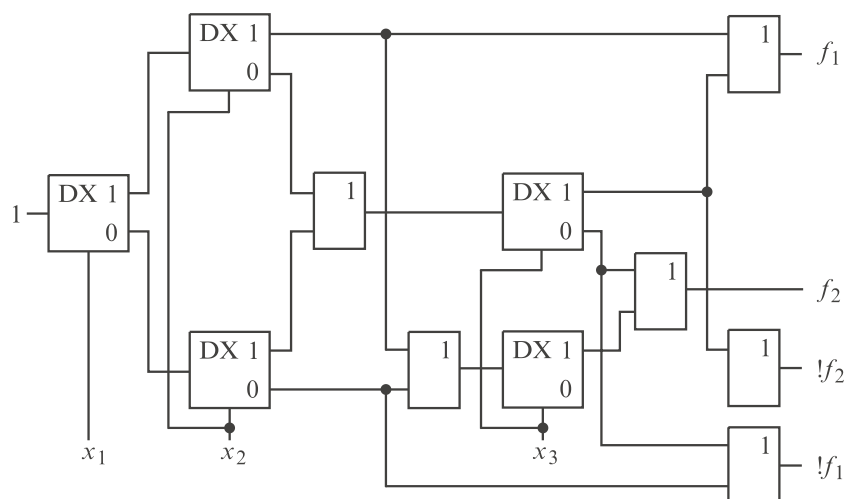


Рис. 16.107

Для реализации функций системы (без их инверсий) построенная схема должна быть упрощена за счет исключения и упрощения элементов, связанных с реализацией инверсных функций и функции f_k .

Для непосредственной реализации функций системы изложенный метод должен быть модифицирован. Эта модификация применима к бинарным графам, которые либо не содержат операторных вершин начальной установки, либо эти вершины должны быть типа $f_j = 0$.

Предлагаемая модификация связана с изменением пятого этапа изложенного выше метода, который в данном случае формулируется следующим образом: все операторные вершины типа $f_j = 0$ заменяются линиями.

Число элементов в схеме в этом случае определяется соотношением:

$$n_{э1} \leq B_y + n_{т1} + N,$$

где $n_{т1}$ — число точек объединения дуг, связанных с вычислением функций f .

Построенная указанным образом схема может быть упрощена за счет исключения элементов ИЛИ, реализующих функцию f_k , а также каждого демультиплексора совместно с элементом ИЛИ, подсоединенным к его выходам. В последнем случае функция, формируемая на выходе элемента ИЛИ, переносится в точку подключения входа исключаемого демультиплексора. Дальнейшее упрощение схемы достигается за счет замены некоторых из оставшихся демультиплексоров двухвходовыми элементами И, ЗАПРЕТ, перенными и (или) их инверсиями.

Так как в линейном бинарном графе $B_y = h$, $0 \leq n_{\tau 1} \leq h - 1$, то справедливо соотношение

$$h \leq n_{\tau 1} \leq 2h - 1.$$

При этом нижняя оценка достигается для функции И, а верхняя — для функции ИЛИ.

Сложность схемы, построенной по ЛБГ, в этом случае зависит от порядка ее записи, так как при этом может измениться значение переменной $n_{\tau 1}$.

На рис. 16.108 приведена схема, реализующая формулу $f = x_1 x_2 \vee x_3$, а на рис. 16.109 — формулу $f = x_3 \vee x_2 x_1$.

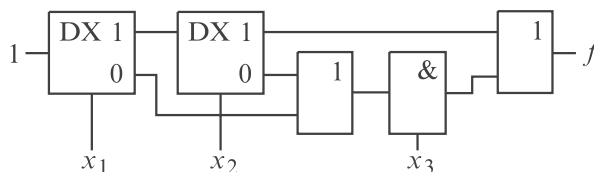


Рис. 16.108

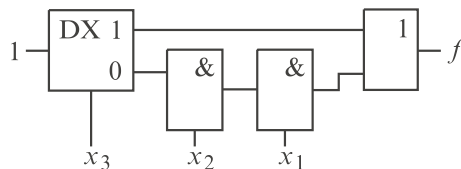


Рис. 16.109

Из приведенных примеров следует, что при реализации булевых формул, неповторных в базисе $\{\&, \vee, !\}$, предлагаемый метод менее эффективен по сравнению с формульным методом построения схем из двухвходовых элементов И и ИЛИ, для которого $n_{\tau 1}^{\phi} = h - 1$.

Предлагаемый метод при реализации повторных формул в указанном базисе обычно более эффективен, чем указанный формульный метод.

Так, например, схема, реализующая предлагаемым методом линейную функцию n переменных, состоит из $2(n - 1)$ демультиплек-

соров, $2n - 3$ двухвходовых элементов ИЛИ и одного двухвходового элемента И ($n_{э1} = 4(n - 1)$). На рис. 16.110 приведена схема восьми двухвходовых элементов, реализующая линейную функцию трех переменных, которая может быть также построена в результате упрощения второй схемы, представленной на рис. 16.106.

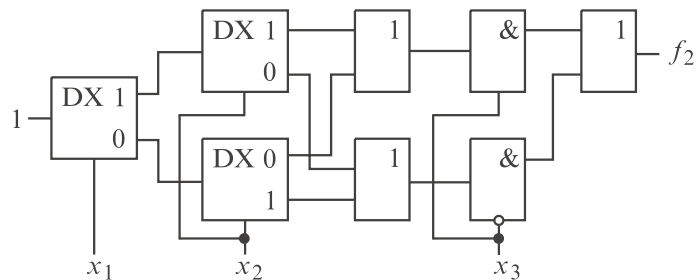


Рис. 16.110

При этом отметим, что для функций этого класса при их записи в базисе $\{\&, \vee, !\}$ справедливо соотношение

$$h = 3 \cdot 2^{n-1} - 2.$$

Эта схема может быть упрощена, если три последних элемента в ней заменить мультиплексором «2 в 1» (рис. 16.111).

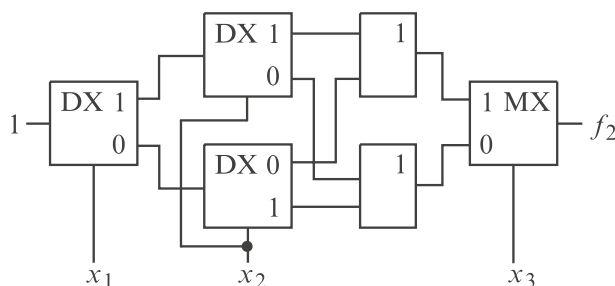


Рис. 16.111

Отметим, что линейные функции при их записи с использованием операций «сумма по модулю два» могут быть реализованы более эффективно схемами из мультиплексоров «2 в 1» с помощью формульного метода, изложенного в разд. 5.1.6. При этом $n_{э1}^{\phi} = n - 1$.

Изложенный метод позволяет реализовать мажоритарную функцию трех переменных, являющуюся симметрической, схемой из шести двухвходовых элементов (рис. 16.112), которая может быть также получена в результате упрощения первой схемы (рис. 16.106).

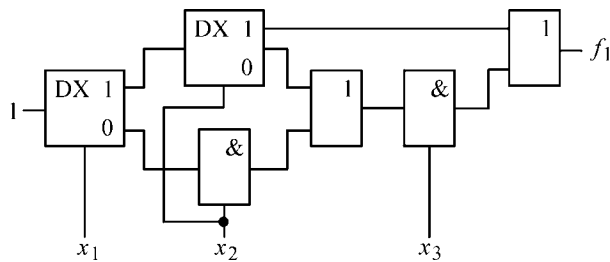


Рис. 16.112

Симметрические функции n переменных могут быть также реализованы и на основе схемы, изоморфной симметрической структуре из переключаемых контактов [50]. Эта схема состоит из $(n+1)n/2$ демультиплекторов и $n(n-1)/2$ двухвходовых элементов ИЛИ ($n_3 = n^2$). На рис. 16.113 приведена схема этого класса при $n = 3$.

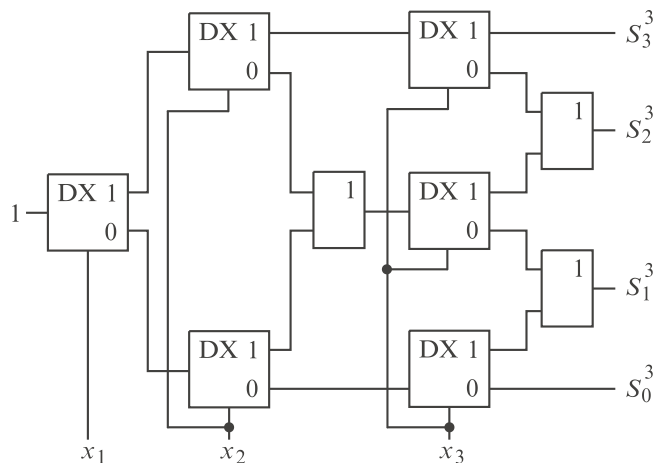


Рис. 16.113

Применение этой структуры в качестве универсальной для реализации одиночных симметрических функций неэффективно, так как для этих целей существует схема, построенная из $(n+1)n/2$ мультиплекторов «2 в 1». Такая схема для $n = 3$ приведена на рис. 9.9.

Однако на основе предложенной схемы в отличие от других методов можно достаточно просто (по трудоемкости) реализовать системы симметрических функций n переменных. На рис. 16.114 приведена схема, реализующая параллельный одноразрядный двоичный сумматор и состоящая из 11 двухвходовых логических эле-

ментов. При этом отметим, что раздельная реализация функций сумматора, выполненная с помощью предлагаемого метода, строит схему из 12 логических элементов (рис. 16.111, 16.112).

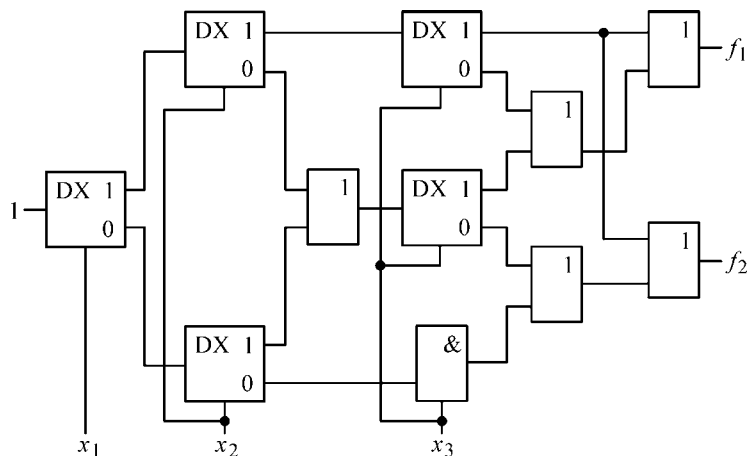


Рис. 16.114

Эта схема может быть также построена и с помощью предлагаемого метода по простому БГ (рис. 16.66).

Если этот метод применить к обобщенному БГ (рис. 16.70), то рассматриваемый сумматор реализуется схемой, содержащей девять двухвходовых элементов (рис. 16.115).

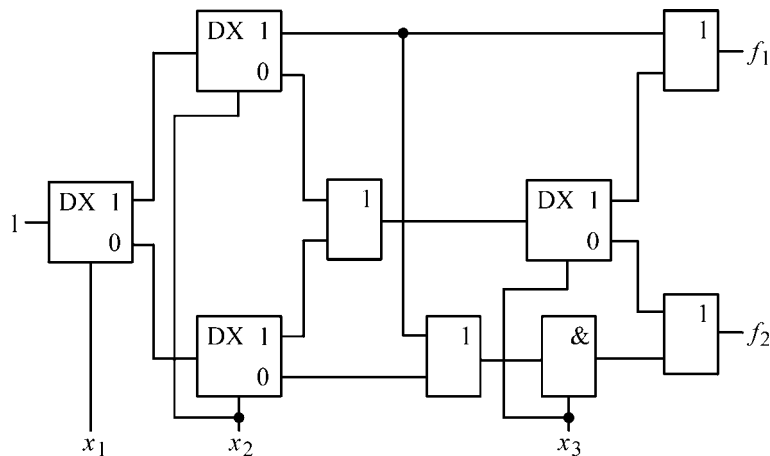


Рис. 16.115

Несмотря на то что известна схема рассматриваемого сумматора, состоящая всего лишь из пяти двухвходовых элементов

(рис. 10.19), это не уменьшает значения предлагаемого универсального метода.

Рассмотрим систему формул, отличных от рассмотренных выше, которая приведена в примере 16.38. После преобразования эта система приобретает вид:

$$f_1 = x_1 \vee x_2 \vee x_3, \quad f_2 = (!x_1 \vee x_2) \vee x_3)x_4.$$

Выполним факторизацию этих формул:

$$\Phi = x_1 \vee x_2, \quad f_1 = \Phi \vee x_3, \quad f_2 = (!\Phi \vee x_3)x_4.$$

Полученная система реализуется схемой, использующей де-мультиплексор и содержащей пять элементов (рис. 16.116).

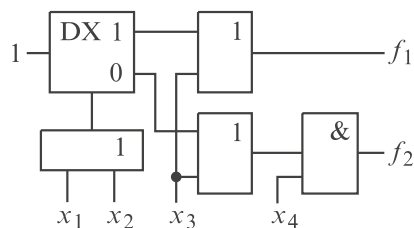


Рис. 16.116

Предлагаемый метод строит по структурированному БГ (рис. 16.71) изоморфную ему схему менее трудоемко (рис. 16.117).

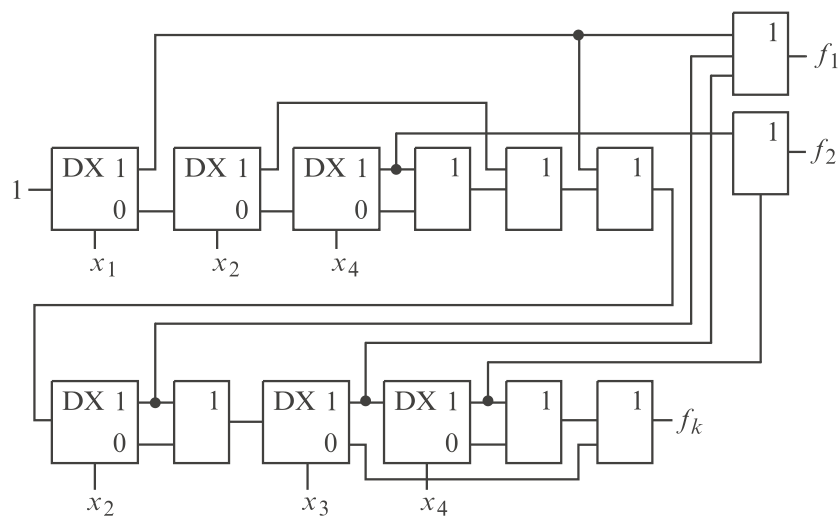


Рис. 16.117

Эта схема состоит из 14 элементов, так как в этом случае $B_y = 6$, $n_{T1} = 6$, $N = 2$. На рис. 16.118 приведена схема из шести элементов, построенная в результате упрощения предыдущей схемы.

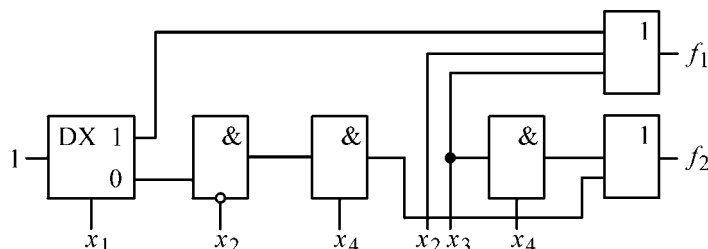


Рис. 16.118

В заключение раздела отметим, что предложенный метод дополняет изложенные в разд. 1.6 методы совместной реализации булевых функций.

Выводы

1. Рассмотрены три метода структурирования линейных бинарных графов и получены оценки числа вершин в структурированных бинарных графах (СБГ). Эти оценки являются в том числе и линейными от числа букв в реализуемой булевой формуле в базисе $\{\&, \vee, !\}$.

2. Предложен метод зависимых фрагментов, позволяющий по булевой формуле в указанном базисе строить линейризованные СБГ (ЛСБГ). Показано, что этот метод реализует неповторные пороговые формулы (БПФ) с помощью ЛСБГ с числом вершин $2h + 1$.

3. Предложен метод зависимых фрагментов с оптимизацией, который обеспечивает реализацию неповторных пороговых формул с помощью ЛСБГ с числом вершин

$$h + 2 \leq B \leq 2h.$$

4. Показано, как с помощью метода зависимых фрагментов с оптимизацией реализовать булевы формулы, имеющие пороговый образ.

5. Усовершенствован метод независимых фрагментов, позволяющий по булевой формуле и системе булевых формул строить оператор структуры, число букв в котором равно числу вершин в ЛСБГ. Изложен метод минимизации числа букв в этом операторе. Показано, что в отдельных случаях этот метод позволяет строить СБГ, число вершин в которых меньше числа букв в указанном операторе.

6. Предложен распределительный метод, позволяющий реализовывать булевы функции, заданные таблицами истинности, при порядке переменных, противоположном таковому в каноническом методе Блоха. Это позволяет по одной таблице истинности (без перестановки столбцов входных переменных в ней) строить два бинарных графа и выбирать простейший из них.

7. Показано, что существуют повторные булевы формулы, реализуемые непланарными неповторными линейными бинарными графами. Определено необходимое и достаточное условие, обеспечивающее получение такой реализации.

8. Показано, что самые сложные по числу путей бинарные графы не являются самыми сложными по числу вершин, и наоборот.

9. Показано, что число максимальных ортогональных областей в карте Карно является нижней оценкой сложности числа путей в бинарном графе.

10. Предложена модификация канонического метода Блоха для построения простых бинарных графов, реализующих системы булевых функций.

11. Предложена модификация канонического метода Блоха для построения обобщенных бинарных графов, реализующих системы булевых функций, основанная на вынесении значений функций на остов канонической структуры.

12. Предложены методы построения для систем булевых функций, арифметических граф-схем, в операторных вершинах которых применяются операторы вида $F = F * C$, где $*$ = {+, -}; C — константа.

13. Для булевых функций, и систем булевых функций, предложены методы построения векторных бинарных графов, в условных вершинах которых выполняется сравнение значений входного вектора с соответствующими масками.

14. Предложен метод построения по бинарным графам изоморфных им схем из демультиплексоров «1 в 2» и двухвходовых элементов ИЛИ, совместно реализующих булевы функции.

Л и т е р а т у р а

1. Иодан Э. Структурное проектирование и конструирование программ. М.: Мир, 1979.
2. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. М.: Мир, 1982.
3. Артохов В. Л., Кузнецов Б. П., Шальто А. А. Настраиваемые логические устройства для судовых управляющих систем. Л.: ИПК СП, 1986.
4. Кузнецов Б. П., Шальто А. А. Структурный подход к программной реализации булевых формул // Автоматика и вычисл. техника. 1985. № 5.
5. Aschcroft E., Manna Z. The translation of «goto» programs into «while» programs // Proceedings of 1971 IFIP Congress.
6. Шальто А. А. Реализация алгоритмов судовых управляющих логических систем при использовании микропроцессорной техники. Л.: ИПК СП, 1988.

7. Шальто А. А. Switch-технология. Алгоритмизация и программирование задач логического управления. СПб: Наука, 1998.
8. Артюхов В. Л., Шальто А. А. Реализация булевых формул однородными мультиплексорными и мажоритарными каскадами // Изв. РАН. Теория и системы управления. 1996. № 5.
9. Артюхов В. Л., Кузнецова О. С., Шальто А. А. Оценка функциональных возможностей программируемых логических матриц // Автоматика и вычисл. техника 1985. № 2.
10. Кузнецов Б. П. Структурирование бинарных программ // Вопросы судостроения. Сер. Судовая автоматика. 1983. Вып.29.
11. Кузнецов Б. П., Шальто А. А. Метод независимых фрагментов для построения линеаризованных структурированных граф-схем алгоритмов, реализующих системы булевых формул // Автоматика и телемеханика. 1998. № 9.
12. Блох А. Ш. Граф-схемы и их применение. Минск: Вышэйшая школа, 1975.
13. Лофгерен Л. Теория однородных переключаемых сетей // Синтез релейных структур. М.: Наука, 1965.
14. Кириллов А. П., Кузнецов Б. П., Шальто А. А. Минимизация числа безусловных переходов в бинарных программах // Судостроит. пром-сть. Сер. Автоматика и телемеханика. 1986. Вып.1.
15. Артюхов В. Л., Копейкин Г. А., Шальто А. А. Настраиваемые модули для управляющих логических устройств. Л.: Энергоиздат, 1981.
16. Кузнецов О. П. О программной реализации логических функций и автоматов // Автоматика и телемеханика. 1977. № 7, 9.
17. Кузьмин В. А. Оценки сложности реализации ФАЛ простейшими видами бинарных программ // Методы дискретного анализа и теория кодов и схем. Новосибирск, 1976. Вып.29.
18. Autolog 32. Руководство пользователя. FF-Automation OY.
19. Рубинов В. И., Шальто А. А. Метод построения граф-схем простых бинарных программ для систем булевых функций // Автоматика и вычисл. техника. 1986. № 4.
20. Артюхов В. Л., Рубинов В. И., Шальто А. А. Метод построения обобщенных бинарных программ для систем булевых функций // Вопросы судостроения. Сер. Судовая автоматика. 1982. Вып. 27.
21. Рубинов В. И., Шальто А. А. Построение граф-схем бинарных программ для систем булевых функций, заданных таблицами истинности // Автоматика и вычисл. техника. 1988. № 1.
22. Малюгин В. Д. Реализация булевых функций арифметическими полиномами // Автоматика и телемеханика. 1982. № 4.
23. Артюхов В. Л., Кондратьев В. Н., Шальто А. А. Реализация булевых функций арифметическими полиномами // Автоматика и телемеханика. 1988. № 4.
24. Lee C. Y. Representation of switching circuits by binary decision programs // Bell Syst. Tech. J. 1959. N 7.
25. Arden B. W., Caller B. A., Graham R. M. An algorithm for translating boolean expressions // J. Assoc. Comput. Mach. 1962. N 4.
26. Clare C. R. Designing logic systems using state machines. New York: McGraw-Hill, 1973.
27. Boute R. T. The binary decision machine as programmable controllers // Euromicro News Lett. 1976. N 2.
28. Payne H. J., Meisel W. S. An algorithm for constructing optimal binary decision trees // IEEE Trans. on Comput. 1977. N 9.
29. Akers S. B. Binary decision diagrams // IEEE Trans. on Comput. 1978. N 6.
30. Akers S. B. Functional testing with binary decision diagrams // Proc. 8th Ann. IEEE Conf. Fault-Tolerant Comput. 1978.
31. Prather R. E., Casstevens H. T. Realization of boolean expressions by atomic digraphs // IEEE Trans. on Comput. 1978. N 8.
32. Cerny E., Mange D., Sanchez E. Synthesis of minimal binary decision trees // IEEE Trans. on Comput. 1979. N 9.

33. *Moret B. M.* Decision trees and diagrams // *Ass. Comput. Math., Comput. Surv.* 1982. N 12.
34. *Bryant R. E.* et al. Logic minimization algorithms for VLSI synthesis. Hingham, MA : Kluwer, 1984.
35. *Bryant R. E.* Graf-based algorithms for boolean function manipulation // *IEEE Trans. on Comput.* 1986. N 8.
36. *Brace K. S., Rudell R. L., Bryant R. E.* Efficient implementation of a BDD package // *Proceeding of the 27th Design Automation Conf. Orlando, FL.* 1990.
37. *Friedman S. J., Supowit K. J.* Finding the optimal variable ordering for binary decision diagrams // *IEEE Trans. on Comput.* 1990. N 5.
38. *Ishura N., Sawada H., Yajima S.* Minimization of binary decision diagrams based on exchanges of variables // *Proceeding of the Intern. Conf. on Computer-Aided Design.* Santa Clara, CA. 1991.
39. *Fujita M., Matsunaga Y., Kakuda T.* On variable ordering of binary decision diagrams for the application of multilevel synthesis // *Proceeding of the European Conf. on Design Automation.* Amsterdam, 1991.
40. *Brayton R. K., Sentovich E. M., Singh K. J.* et al. Sequential circuit design using synthesis and optimization // *Proceeding of the Intern. Conf. on Computer Design.* Cambridge, MA. 1992.
41. *Brayton R. K. et al.* SIS : A System for Sequential Circuit Synthesis // *Technical report UCB/ERL M92/41.* Electronics Research Laboratory. Univ. of California. Berkley, 1992.
42. *Jeong S. W., Kim T. S., Somenzi F.* An efficient method for optimal BDD ordering computation // *Intern. Conf. on VLSI and CAD (ICVC'93).* Taejon. Korea, 1993.
43. *Rudell R.* Dynamic variable ordering for ordered binary decision diagrams // *Proceeding of the Intern. Conf. on Computer-Aided Design.* Santa Clara, CA. 1993.
44. *Bahar R. I., Frofm E. A., Gaona C. M.* et al. Algebraic decision diagrams and their application // *Proceeding of the Intern. Conf. on Computer-Aided Design.* Santa Clara, CA. 1993.
45. *Panda S., Somenzi F., Plessior B. F.* Symmetry detection and dynamic variable ordering of decision diagrams // *Proceeding of the Intern. Conf. on Computer-Aided Design.* San Jose, CA. 1994.
46. *Panda S., Somenzi F.* Who are the variables in your neighborhood // *Proceeding of the Intern. Conf. on Computer-Aided Design.* San Jose, CA. 1995.
47. *Brayton R. K. et al.* VIS: A System for verification and synthesis // *Technical report UCB/ERL M95/104.* Electronics Research Laboratory. Univ. of California. Berkley, 1995.
48. *Карпов Ю. Г.* Теория алгоритмов и автоматов. Курс лекций. СПб.: СПбГТУ, Нестор, 1998.
49. *Артюхов В. Л., Шальто А. А.* Судовые управляющие логические системы. Л.: ИПК СП, 1984.
50. *Шеннон К.* Работы по теории информации и кибернетике. М.: Изд-во иностр. литер., 1962.