МИНИМИЗАЦИЯ ЛОГИЧЕСКИХ ВЫРАЖЕНИЙ

Е. В. Сидоров, С. Б. Карпов
Пермский государственный национальный исследовательский университет,
614990, Пермь, Букирева, 15.

Введение. Учитывая то, что одну и ту же логическую функцию можно представить различными выражениями, перед реализацией функции в виде логической схемой весьма важным является выбор из всех возможных выражений, соответствующих данной функции, самого простого. Решить эту проблему можно за счет использования процедуры минимизации логического выражения.

Минимизацией называют преобразование заданной логической функции с целью уменьшения общего числа переменных и операций. Процесс минимизации имеет важное значение при технической реализации дискретных устройств, так как при этом уменьшается общее количество элементов, увеличивается надежность и устройства становятся боле экономичными.

Постановка задачи. Необходимо разработать алгоритм и программу, которая производила бы минимизацию произвольной логической функции, заданной в виде СДНФ. Количество переменных задается пользователем.

Наиболее подходящим методом минимизации для программной интерпретации является метод Квайна. К основным достоинствам которого можно отнести то, что всё представлено в виде чисел, с которыми легко оперировать.

В качестве языка программирования я выбрал C++ поскольку это универсальный высокоуровневый объектно-ориентированный язык программирования пригодный для задач любой сложности.

Обзор существующих решений. Можно разделить способы на те, что осуществляются с помощью программ, и те, которые человек выполняет самостоятельно.

Из множества методов минимизации наиболее часто используется минимизация методом Квайна.

В качестве исходной формы представления логического выражения используется СДНФ (1) и (2).

$$f\left(x_1,x_2,x_3,x_4\right)=\bigvee_1\left(0,1,4,5,7,8,10,12,14,15\right) \quad (1)$$

$$f\left(x_1,x_2,x_3,x_4\right)=\bar{x}_1\bar{x}_2\bar{x}_3\bar{x}_4\vee\bar{x}_1\bar{x}_2\bar{x}_3x_4\vee\bar{x}_1x_2\bar{x}_3\bar{x}_4\vee\bar{x}_1x_2\bar{x}_3x_4\vee \quad (2)$$

$$\vee\bar{x}_1x_2x_3x_4\vee x_1\bar{x}_2\bar{x}_3\bar{x}_4\vee x_1\bar{x}_2x_3\bar{x}_4\vee x_1x_2\bar{x}_3\bar{x}_4\vee x_1x_2x_3\bar{x}_4\vee x_1x_2x_3x_4$$

Метод Квайна выполняется два этапа.

Первый этап имеет своей целью получение тупиковой формы, представляющую собой дизъюнкцию (Сокр. ДНФ), в качестве слагаемых которой используются конъюнкции, каждая из которых не склеивается ни с одной другой конъюнкцией, входящей в это выражение:

$$f\left(x_1,x_2,x_3,x_4\right)=\overline{x}_1x_2x_4\vee x_2x_3x_4\vee x_1x_2x_3\vee x_1x_3\vee\overline{x}_1\overline{x}_3\vee\overline{x}_3\overline{x}_4\vee x_1\overline{x}_4 \quad (3)$$

Данный этап выполняется за счет реализации отдельных шагов. На каждом шаге на основании выражения, полученного на предыдущем шаге, выполняются все возможные операции склеивания для пар имеющихся конъюнкций. Каждый шаг понижает ранг исходных конъюнкций на единицу. Шаги повторяются до тех пор, пока это возможно.

Второй этап выполняется с помощью таблицы Квайна (рис. 1). Данный этап имеет своей целью устранения из тупиковой формы всех избыточных простых импликант, что дает в результате минимальное логическое выражение (4).

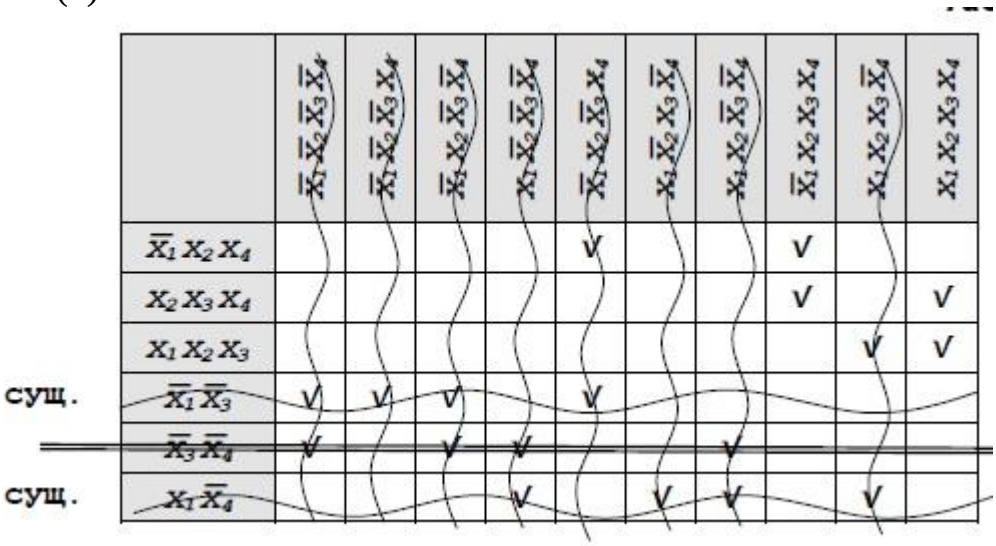

Рис. 1. *Пример таблицы Квайна*

$$f\left(x_1,x_2,x_3,x_4\right)=x_2x_3x_4\vee\overline{x}_1\overline{x}_3\vee x_1\overline{x}_4 \quad (4)$$

Программа, умеющая делать нужные нам вычисления, быстро выдаёт нам ответ, но всё это без промежуточных этапов.

Если же самостоятельно сидеть и минимизировать выражение на листе бумаги, то можно легко ошибиться. Необходим способ, совмещающий в себе первые два.

Описание практической части. Сначала пользователь вводит количество аргументов в функции и вектор значений, на которых наше выражение равно единице. Далее программа в автоматическом режиме преобразует каждое значение в двоичный вид и заносит все значения в последовательность минтермов. Составляется совершенная дизъюнктивная нормальная форма (СДНФ). Далее мы производим всевозможные склеивания, пока это возможно. Готово, получена сокращенная дизъюнктивная нормальная форма (Сокр. ДНФ). По полученным данным строится таблица Квайна, заполняется. Составляется тупиковая ветвь, из которой методом перебора откидываются ненужные элементы.

Программа реализована с учётом принципов объектно-ориентированного программирования. Результат работы программы для минимизации указанной выше логической функции приведён на рис. 2.

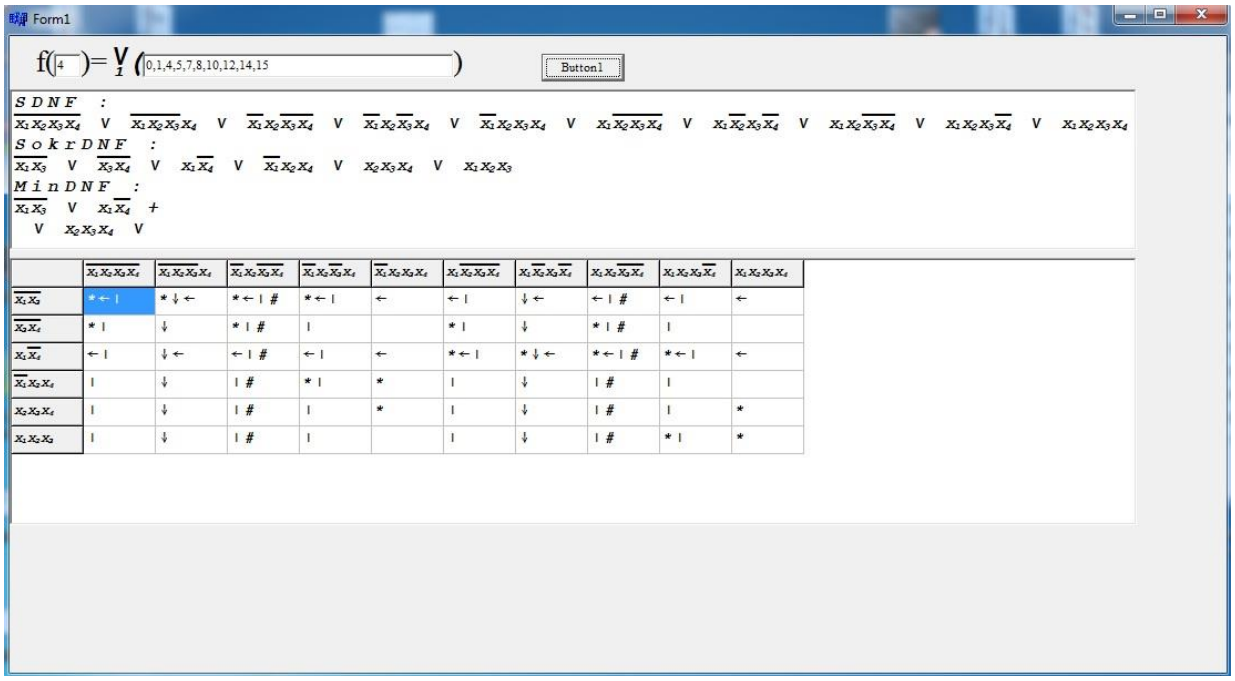

Рис. 2. *Пример работы реализованной программы*

Заключение. Видно, в программе можно задавать количество переменных, задавать значения, на которых функция равна единице. Плюс ко всему, вставлен собственный шрифт Logic Formul, благодаря которому вывод информации стал приятнее на вид и легче осуществился в коде. Преимуществом моей программы является поэтапный вывод информации. Выводятся СДНФ, сокращенная форма, минимальная форма. Рисуется таблица Квайна с обозначениями, из которых выписываются сразу существенные импликанты, затем составляется формальное выражение Петрика и методом перебора отбрасываются лишние элементы. Пользователь сможет сверять свои расчеты на каждом этапе. Это довольно удобно, т.к. расчеты на бумаге утомительны и однообразны. Можно легко запутаться или что-нибудь потерять.

Список литературы

1. *Карпов С. Б.* Учебно-методическое пособие по спецкурсу «Цифровая схемотехника: комбинационные схемы». 2013. С. 25–31.