ДЕКОМПОЗИЦИЯ ДИАГРАММ ДВОИЧНОГО ВЫБОРА, ЗАДАЮЩИХ СИСТЕМЫ НЕ ПОЛНОСТЬЮ ОПРЕДЕЛЕННЫХ БУЛЕВЫХ ФУНКЦИЙ

Бибило П. Н.
Объединённый институт проблем информатики Национальной академии наук Беларуси
Минск, Республика Беларусь
E-mail: bibilo@newman.bas-net.by

Предлагается метод декомпозиции системы не полностью определенных булевых функций, представленных в виде диаграммы двоичного выбора. Минимизация числа промежуточных функций при такой декомпозиции ориентирована на увеличение быстродействия логических схем.

Введение

Диаграммы двоичного выбора (Binary Decision Diagram – BDD), называемые также диаграммами двоичных решений явились конкретной формой задания граф-схем алгоритмов выбора решений применительно к булевым функциям. Представление булевых функций и систем в виде BDD – многоуровневых представлениях на базе разложения Шеннона во многих случаях является гораздо более компактным, чем представления тех же функций в матричных формах, под которыми понимаются ДНФ (дизъюнктивные нормальные формы) и таблицы истинности. Кроме того, использование специальных структур данных при программной реализации алгоритмов получения и обработки BDD позволяет получать эффективные программы выполнения различных операций над BDD. Аппарат BDD широко используется при решении различных задач, возникающих при проектировании цифровых систем.

Декомпозиция булевых функций по входным переменным изучалась давно и использовалась при синтезе комбинационных логических схем [1]. Для синтеза структур FPGA (Field-Programmable Gate Array – программируемая пользователем вентильная матрица) декомпозиция стала одним из приемов технологического отображения в сеть настраиваемых элементов LUT (Look-Up Table – таблица, реализующая логическую функцию). При синтезе схем из библиотечных элементов декомпозиция может использоваться либо с целью разбиения функционального описания на меньшие по размерности блоки, для которых возможно использовать более эффективные (и более трудоемкие) оптимизационные процедуры, либо с целью увеличения быстродействия схем.

В данной работе рассматривается задача декомпозиции системы не полностью определенных (частичных) функций $\boldsymbol{f}(\boldsymbol{x}) = (f^1(\boldsymbol{x}), \ldots, f^m(\boldsymbol{x}))$, заданных диаграммой двоичного выбора. Основное отличие метода декомпозиции BDD, задающей систему частичных функций, от декомпозиции BDD, задающей систему полностью определенных функций, – это сравнение коэффициентов разложения Шеннона на совместимость (возможность доопределения их до одной функции), при этом коэффициенты задаются подграфами BDD. Исходная для декомпозиции BDD имеет m корневых вершин, листовые вершины BDD помечены 0, 1, либо неопределенным значением «$-$».

I. Постановка задачи

Для частичной (не полностью определенной) векторной булевой функции $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z})$, представленной в виде BDD, и заданного разбиения Y/Z множества переменных X построить функциональное разложение (провести декомпозицию векторной функции) вида

$$\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}) = \boldsymbol{g}(\boldsymbol{h}(\boldsymbol{y}), \boldsymbol{z}), \qquad (1)$$

где $\boldsymbol{h}(\boldsymbol{y}) = (h_1(\boldsymbol{y}), \ldots, h_p(\boldsymbol{y}))$. При этом требуется минимизировать число p компонент векторной функции $\boldsymbol{h}(\boldsymbol{y})$ и представить ее в виде системы ДНФ, а векторную функцию $\boldsymbol{g}(\boldsymbol{h}(\boldsymbol{y}), \boldsymbol{z})$ представить в виде BDD.

II. Метод декомпозиции

Решение задачи осуществляется в шесть этапов.

Этап 1. Построение кратчайшего разложения Шеннона векторной функции $\boldsymbol{f}(\boldsymbol{x})$ по кратчайшим разложениям Шеннона компонентных функций.

Построим классическое разложение Шеннона

$$\boldsymbol{f}(\boldsymbol{y}, \boldsymbol{z}) = \overline{y}_1\overline{y}_2 \ldots \overline{y}_r \boldsymbol{f}_{\boldsymbol{y}_0^\star}(\boldsymbol{z}) \vee \overline{y}_1\overline{y}_2 \ldots y_r \boldsymbol{f}_{\boldsymbol{y}_1^\star}(\boldsymbol{z}) \vee \cdots \vee y_1 y_2 \ldots y_r \boldsymbol{f}_{\boldsymbol{y}_{2^r-1}^\star}(\boldsymbol{z}) \qquad (2)$$

векторной функции $\boldsymbol{f}(\boldsymbol{x})$ по подмножеству аргументов Y.

Сгруппировав в один класс (множество) дизъюнктивных членов разложения (2) с одинаковыми коэффициентами $\boldsymbol{f}(\boldsymbol{z})$, получим разложение вида

$$\boldsymbol{f}(\boldsymbol{y},\boldsymbol{z}) = D^1\boldsymbol{f}_1(\boldsymbol{z}) \vee D^2\boldsymbol{f}_2(\boldsymbol{z}) \vee \dots \vee D^k\boldsymbol{f}_k(\boldsymbol{z}), \quad (3)$$

называемое *кратчайшим* (по числу дизъюнктивных членов) разложением Шеннона векторной функции. ДНФ Q_i могут быть минимизированы в классе ДНФ.

Так как по BDD легко строятся [2] кратчайшие разложения Шеннона компонентных функций, то возникает задача построения кратчайшего разложения Шеннона (3) векторной функции по кратчайшим разложениям Шеннона компонентных функций. Данная задача сводится к задаче нахождения минимального дизъюнктивного базиса для системы S ДНФ. Систему S образуют ДНФ $Q_i^j(\boldsymbol{y})$, входящие в кратчайшие разложения Шеннона компонентных функций $f^j(\boldsymbol{x})$, $j = 1,\dots,m$. *Минимальным дизъюнктивным базисом* системы S ДНФ называется минимальная по мощности система попарно ортогональных ДНФ $D = \{D^1(\boldsymbol{y}),\dots,D^k(\boldsymbol{y})\}$ такая, что каждая ДНФ $Q_i^j(\boldsymbol{y}) \in S$ равна дизъюнкции некоторого подмножества ДНФ системы D. Нахождение минимального дизъюнктивного базиса D сводится к выполнению операций перемножения ДНФ системы S.

Этап 2. Определение минимального числа промежуточных функций h и их построение.

Выполнение этого этапа сводится к построению графа $G_{\boldsymbol{f}}$ отношения несовместимости ДНФ D, раскраске вершин графа $G_{\boldsymbol{f}}$ в минимальное число цветов и кодированию подмножеств одноцветных вершин графа $G_{\boldsymbol{f}}$ попарно ортогональными булевыми либо троичными кодами [1]. Две ДНФ D^i, D^j являются несовместимыми тогда и только тогда, когда их множителями в разложении (12) являются несовместимые векторные частичные функции $\boldsymbol{f}_i(\boldsymbol{z})$, $\boldsymbol{f}_j(\boldsymbol{z})$. Две частичные векторные функции $\boldsymbol{f}_i(\boldsymbol{z})$, $\boldsymbol{f}_j(\boldsymbol{z})$ назовем *несовместимыми*, тогда и только тогда, когда несовместимой является хотя бы одна пара соответствующих компонент данных векторных функций. В противном случае векторные функции назовем *совместимыми*. Легко видеть, что несовместимые векторные функции не могут быть доопределены до одной и той же векторной функции, а совместимые – могут. Если функции $\boldsymbol{f}_i(\boldsymbol{z})$, $\boldsymbol{f}_j(\boldsymbol{z})$ являются несовместимыми, то в графе $G_{\boldsymbol{f}}$ между вершинами D^i, D^j имеется ребро, если же данные функции являются совместимыми, то вершины D^i, D^j являются несмежными (не соединяются ребром). В свою очередь частичные функции $\boldsymbol{f}_a^j(\boldsymbol{z})$, $\boldsymbol{f}_b^j(\boldsymbol{z})$ называются *несовместимыми* тогда и только тогда, когда найдется хотя бы один набор $\boldsymbol{z}^\star$ значений аргументов, для которого обе функции $\boldsymbol{f}_a^j(\boldsymbol{z})$, $\boldsymbol{f}_b^j(\boldsymbol{z})$ определены и не равны [1]. В противном случае эти функции называются *совместимыми*. Проверка отношения несовместимости частичных функций может быть осуществлена с помощью алгоритмов, представленных в [2]. Можно применять также алгоритмы выполнения операции $\oplus$ (исключающее ИЛИ) над BDD [3, с. 259].

Векторная функция $\boldsymbol{h}(\boldsymbol{y}) = (h_1(\boldsymbol{y}),\dots,h_p(\boldsymbol{y}))$ в (1) строится кодированием одноцветных (одинаково раскрашенных) ДНФ $D^1(\boldsymbol{y}),\dots,D^k(\boldsymbol{y})$ попарно ортогональными булевыми либо троичными векторами.

Этап 3. Построение выходной векторной функции $\boldsymbol{g}$ в виде системы ДНФ. Построение ДНФ выходной векторной функции осуществляется заменой ДНФ D^i кодами.

Этап 4. Построение BDD представления частичной векторной функции $\boldsymbol{g}$ последовательным разложением Шеннона по промежуточным переменным h.

Этап 5. Минимизация в классе ДНФ системы частичных функций $\boldsymbol{h}(\boldsymbol{y}) = (h_1(\boldsymbol{y}),\dots,h_p(\boldsymbol{y}))$ и получение ДНФ полностью определенных функций, реализующих минимизированные функции.

Этап 6. Минимизация сложности BDD, представляющих частичные функции g^i, и получение BDD полностью определенных функций, реализующих g^j.

Заключение

Декомпозиция BDD может служить средством предварительной технологически независимой оптимизации для увеличения быстродействия синтезируемой логической схемы из библиотечных элементов либо использоваться при синтезе логических схем FPGA на этапе технологического отображения. При такой декомпозиции нужно минимизировать число промежуточных переменных. Декомпозиция позволяет сократить число уровней BDD для функций выходного блока, если число полученных промежуточных переменных меньше числа перекодируемых переменных.

1. Бибило, П. Н. Декомпозиция булевых функций на основе решения логических уравнений / П. Н. Бибило. – Минск: Беларус. навука, 2009. – 211 с.
2. Бибило, П. Н. Минимизация диаграмм двоичного выбора для систем не полностью определенных булевых функций / П. Н. Бибило. // Известия РАН. Теория и системы управления. – 2013. – № 6. – С. 54–73.
3. Кнут, Д. Э. Искусство программирования. Том 4, А. Комбинаторные алгоритмы. Часть 1. / Д. Э. Кнут. – Пер. с англ.: ООО «И. Д. Вильямс», 2013. – 960 с.