ОСНОВНЫЕ ПОДХОДЫ К ДЕКОМПОЗИЦИИ БУЛЕВЫХ ФУНКЦИЙ ПРИ РЕАЛИЗАЦИИ НА FPGA

Баркалов А.А., Красичков А.А.
ДонНТУ, каф. ЭВМ
krasich@cs.dgtu.donetsk.ua

Abstract

Barkalov A.A., Krasichkov A.A. The basic approaches to decomposition of Boolean functions at realization on FPGA. *The methods of decomposition of Boolean functions is proposed. Methods are based on the Shannon's transformation of Boolean functions and subsequent subfunctions grouping. The methods are illustrated by examples of Boolean function realization in FPGA bases with various number of CLB inputs.*

1. Введение

В настоящее время средства вычислительной техники (ВТ) получили столь разнообразное применение, что требования к их разработке постоянно растут. Современная вычислительная система должна обладать высоким быстродействием, малыми габаритами и, что самое главное, она должна быть спроектирована за приемлемое время. Ввиду таких требований, со временем появился принципиально новый базис для реализации цифровых схем ВТ – программируемые логические интегральные схемы (ПЛИС). Микросхемы ПЛИС по своей архитектуре делятся на две разновидности: CPLD и FPGA [1]. Синтез цифровых устройств сводится к реализации системы булевых функций (БФ) в базисе микросхемы конкретной ПЛИС. Архитектура CPLD функционально соответствует широко распространенным микросхемам ПЛМ и ПМЛ, поэтому методы синтеза в этом базисе известны и особых сложностей не представляют. Микросхемы FPGA подразумевают принципиально новые методы реализации систем булевых функций, ввиду особенностей своей архитектуры [2]. Алгоритмы функционального синтеза на FPGA являются ноу-хау фирм-производителей конкретных микросхем и встроены в различные САПР иностранного происхождения, которые не удовлетворяют требованиям отечественных разработчиков средств ВТ. Поэтому актуальной является проблема реализации систем булевых функций на ПЛИС с архитектурой FPGA. В настоящей работе предлагаются методы декомпозиции БФ для реализации их в базисе FPGA.

2. Основные положения

Функциональный базис FPGA представляет собой конечное множество одинаковых комбинационных логических блоков (КЛБ) с R входами, которые могут реализовать любую булеву функцию от R переменных, либо две независимые функции от (R-1) переменных. В зависимости от конкретной модели микросхемы FPGA $R = \overline{2,5}$. Число переменных в БФ проектируемых систем обычно значительно больше и колеблется в пределах от 6 до 20. Таким образом, для реализации системы булевых функций от N переменных необходимо каждую функцию представить в виде совокупности подфункций от M переменных, причем, M ≤ R. То есть, возникает задача функциональной декомпозиции в базис с меньшим числом входов. Для решения этой задачи могут быть применимы следующие известные методы:

Минимизация при помощи карт Карно или законов алгебры логики. Этот метод имеет место, если функция задана в виде СДНФ или СКНФ и далеко не всегда приводит к искомой реализации функции на КЛБ.

Преобразование по законам Де Моргана. Этот метод всегда приводит к реализации функции на КЛБ с любым R. Однако число КЛБ значительно превышает минимально возможное для заданной функции, так как каждый блок реализует отдельный терм ДНФ или КНФ, а остальные используются для каскадирования групп термов.

Разложение Шеннона. Этот метод также приводит к реализации любой функции на КЛБ, но число подфункций, из которых состоит реализуемая функция, исчисляется степенями двойки и зависит от числа переменных разложения. Для реализации функции на КЛБ это самый нерациональный метод с точки зрения как аппаратурных затрат, так и с точки зрения быстродействия, так как схема имеет древовидную структуру.

Для реализации БФ на ПЛИС FPGA авторами предлагаются следующие методы.

Поскольку КЛБ может реализовать совершенно любую БФ R переменных, то необходимо лишь обеспечить представление заданной функции в виде минимального числа подфункций с числом переменных M≤R. Функция выражается в виде подфункций таким образом, чтобы минимум переменных являлись аргументами различных подфункций. В идеальном случае аргументы в подфункциях не пересекаются, тогда реализация такой функции в базисе КЛБ наиболее рациональна.

Существует множество вариантов соединения КЛБ для реализации функции, которые могут состоять из фрагментов структур двух типов, представленных на рис.1 и рис.2.

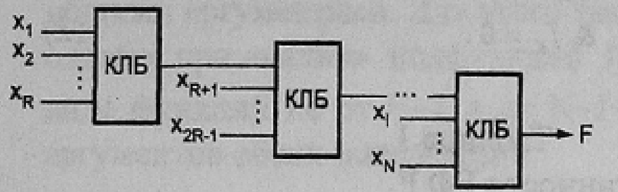

Рисунок 1 – Последовательная структура КЛБ

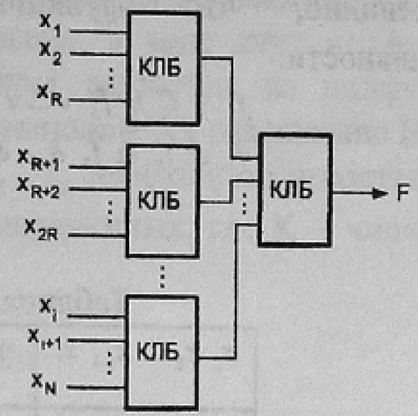

Рисунок 2. – Параллельная структура КЛБ

Быстродействие параллельной структуры выше, однако, как показали исследования авторов, большинство БФ реализуются в виде последовательной структуры. Другими словами, не каждая функция может быть реализована в виде заданной структуры КЛБ.

3. Метод декомпозиции для R=2

Метод реализации функций на КЛБ с R=2 отличается от методов, разработанных авторами для $R=\overline{3,5}$ и состоит в следующем.

Пусть функция F является БФ от N переменных: $F=F(x_1,x_2,\ldots,x_N)$ и ее необходимо реализовать на КЛБ с R=2. Поскольку заранее не известно, какие аргументы функции могут быть вынесены в подфункции, то возникает необходимость проверки такой возможности для всех возможных групп аргументов. Наиболее рационально сразу попытаться разбить функцию F на две подфункции с независимыми аргументами. При этом получится реализация функции в виде параллельной структуры. Такую возможность предлагается проверить при помощи разложения Шеннона и последующего анализа полученных подфункций. Для проверки отделяемости от функции группы I аргументов $x_1, x_2,\ldots, x_i$ запишем:

$$F = f_1(x_1,x_2,...,x_i)\,\&\,f_1^{'}(x_{i+1},x_{i+2},...,x_N) \vee f_2(x_1,x_2,...,x_i)\,\&\,f_2^{'}(x_{i+1},x_{i+2},...,x_N) \vee$$
$$\vee ... \vee f_K(x_1,x_2,...,x_i)\,\&\,f_K^{'}(x_{i+1},x_{i+2},...,x_N),$$

где $f_1^{'}, f_2^{'},...,f_K^{'}$ - функции, полученные в результате подстановки в БФ F значений аргументов $x_1,x_2,...,x_i$, $K=2^i$, а функции $f_1, f_2,...,f_K$ представлены таблицей истинности (Табл.1).

Очевидно, что функции $f_1, f_2, ..., f_K$ обладают свойствами ортогональности:

$$f_1 \vee f_2 \vee f_3 \vee ... \vee f_{K-1} \vee f_K = 1, \quad (1)$$

$$f_1 \,\&\, f_2 \,\&\, f_3 \,\&\, ... \,\&\, f_{K-1} \,\&\, f_K = 0. \quad (2)$$

Таблица 1

Таблица истинности БФ F

x_1 x_2 ... x_{i-1} x_i	f_1 f_2 f_3 ... f_{K-1} f_K
0 0 ... 0 0	1 0 0 ... 0 0
0 0 ... 0 1	0 1 0 ... 0 0
0 0 ... 1 0	0 0 1 ... 0 0
.	.
.	.
.	.
1 1 ... 1 0	0 0 0 ... 1 0
1 1 ... 1 1	0 0 0 ... 0 1

Из этого следует:

$$f_1 \vee f_2 \vee f_3 \vee ... \vee f_{K-1} = \overline{f_K},$$

....

$$f_1 \vee f_2 \vee ... \vee f_i = \overline{f_{i+1} \vee ... \vee f_{K-1} \vee f_K}.$$

Таким образом, дизъюнкция любой группы функций $f_1, f_2, ..., f_K$ равна инверсии дизъюнкции остальной части функций. Из этого можно сделать следующий вывод. Разложение $F = F_3[F_1(x_1, x_2, ..., x_i), F_2(x_{i+1}, x_{i+2}, ..., x_N)]$ имеет место, если множество функций $f_1', f_2', ..., f_K'$ можно разбить на два подмножества любой размерности таким образом, что:

1. В первом подмножестве все элементы равны, во втором – равны нулю.

2. В первом подмножестве все элементы равны, во втором – равны единице.

3. В первом подмножестве все элементы равны, во втором все элементы взаимоинверсны элементам первого множества.

4. Элементы обоих подмножеств равны.

Разбиение получается после вынесения общих сомножителей за скобки и заменой одной из групп функций $f_1, f_2, ..., f_K$ инверсией другой группы. В результате получается функция от двух аргументов-подфункций. При успешном разбиении каждая подфункция подвергается аналогичному разложению до тех пор, пока число аргументов в подфункциях не достигнет двух. Более подробно метод описан в [2].

В том случае, если на каком-либо этапе декомпозиции функция или подфункция не может быть представлена в виде двух подфункций с независимыми аргументами, выполняется разбиение на подфункции с общими аргументами. Для этого также выполняется разложение Шеннона, однако, при анализе подфункций $f_1', f_2', ..., f_K'$ необходимо представить их в виде функций не от N–I, а от N–I+K переменных, где K – число общих аргументов обеих подфункций.

4. Пример применения метода

Пусть функция F задана в виде МДНФ: $F = ab \vee ac \vee cd \vee b\overline{cd}$. Попробуем представить ее в виде: $F = f_3[f_1(ab), f_2(cd)]$. Для этого выполним разложение Шеннона по переменным a и b: $F = \overline{ab}\,\&\,(cd) \vee \overline{a}b\,\&\,(\overline{c \oplus d}) \vee a\overline{b}\,\&\,c \vee ab\,\&\,1$. То есть, $f_1' = cd$, $f_2' = \overline{c \oplus d}$, $f_3' = c$, $f_4' = 1$. Эти подфункции нельзя разбить на два подмножества, удовлетворяющих методу декомпозиции, описанному выше. Согласно методу, можно представить функцию F также в виде: $F = f_3[f_1(ab), f_2(acd)]$, то есть с общим аргументом a. Для этого составим таблицу истинности следующего вида (Табл.2).

Таблица 2

Представление функции F при разложении Шеннона по ab

acd	$f_1' = cd$	$f_2' = \overline{c \oplus d}$	$f_3' = c$	$f_4' = 1$
000	0	1	*	*
001	0	0	*	*
010	0	0	*	*
011	1	1	*	*
100	*	*	0	1
101	*	*	0	1
110	*	*	1	1
111	*	*	1	1
ab	00	01	10	11

Таким образом, из функций $f_1' - f_4'$ от двух переменных получаем частично определенные функции от трех переменных, однако, их также нельзя разбить на два подмножества для декомпозиции. Представление функции F с общим аргументом b приведет к такой же самой таблице.

Теперь попробуем представить функцию в виде: $F = f_3[f_1(bc), f_2(ad)]$. Для этого выполним разложение Шеннона по переменным b и c: $F = \overline{bc}\,\&\,0 \vee \overline{b}c\,\&\,(a \vee d) \vee b\overline{c}\,\&\,(a \vee \overline{d}) \vee bc\,\&\,(a \vee d)$. То есть, $f_1' = 0$, $f_2' = (a \vee d)$, $f_3' = (a \vee \overline{d})$, $f_4' = (a \vee d)$. Эти подфункции тоже нельзя разбить на два подмножества, удовлетворяющих методу. Представим функцию F в виде:

$F = f_3[f_1(bc), f_2(acd)]$ (Табл.3). Видно, что разбиение имеет место. Это следует из того, что $f_1' = 0$, а подфункции f_2', f_3' и f_4' можно склеить в одну, используя звездочки: $f_2' = f_3' = f_4' = a \vee \overline{(c \oplus d)}$, причем дальнейшая декомпозиция для функции F не требуется, так как полученная подфункция уже поделена по аргументам a и c, d.

Реализация функции представлена на рис.3.

Таблица 3

Представление функции F при разложении Шеннона по bc

acd	$f_1' = 0$	$f_2' = (a \vee d)$	$f_3' = (a \vee \overline{d})$	$f_4' = (a \vee d)$
000	0	*	1	*
001	*	*	0	*
010	0	0	*	0
011	*	1	*	1
100	0	*	1	*
101	*	*	1	*
110	0	1	*	1
111	*	1	*	1
bc	00	01	10	11

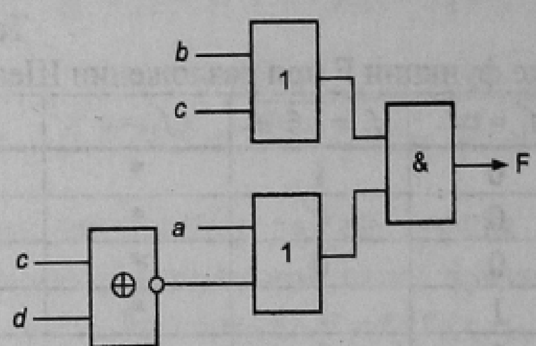

Рисунок 3 – Реализация функции F на КЛБ

5. Метод декомпозиции для $R = \overline{3,5}$

Рассмотрим метод реализации функции на КЛБ с $R = \overline{3,5}$ в случае, когда функция F является БФ от 10 переменных: $F=F(x_1,x_2,\ldots,x_{10})$ и R=4. При реализации в виде последовательной структуры применяется следующий метод.

Для начала, необходимо попытаться разбить функцию на две подфункции – от трех и семи аргументов соответственно. Тогда она будет выглядеть следующим образом (Рис.4).

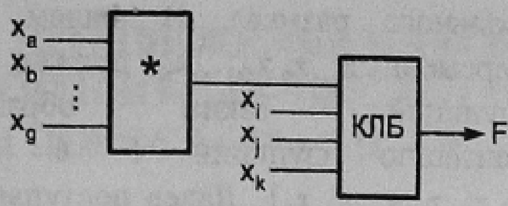

Рисунок 4 – Начальная стадия декомпозиции функции F

Блок, обозначенный звездочкой, представляет собой пока еще не определенное множество КЛБ. Для проверки такой возможности выполняется разложение Шеннона по переменным x_a - x_g. Чтобы данное разбиение имело место, необходимо и достаточно, чтобы выполнялось следующее условие: полученные подфункции разбиваются на два подмножества произвольного размера, в каждом из которых подфункции равны. В нашем случае, среди 128 подфункций от переменных x_i, x_j, x_k должно быть два класса одинаковых подфункций. Таким образом, получаем промежуточную реализацию функции F в следующем виде: $F = f_3[f_1(x_a, x_b, ..., x_g), x_i, x_j, x_k]$. Далее поступаем таким же образом с подфункцией f_1, а затем и с ее подфункциями до тех пор, пока на каком-то шаге число аргументов не станет меньше либо равным четырем. В случае если на каком-либо этапе декомпозиция не буде иметь место для всех возможных комбинаций аргументов, необходимо попытаться выполнить декомпозицию с общими аргументами, как было описано выше. Число таких комбинаций значительно выше, чем при декомпозиции без общих аргументов. Если же и такое разложение не будет иметь места, то возможно, данная функция может быть реализована в виде параллельной структуры, для реализации которой справедлива следующая методика.

Сначала, необходимо попытаться разбить функцию на две подфункции – от трех и семи аргументов таким образом, чтобы она выглядела следующим образом (Рис.5).

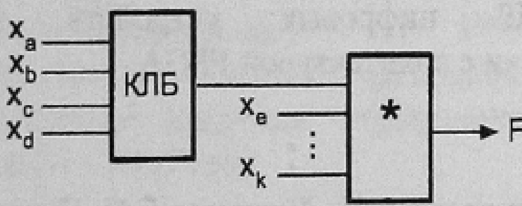

Рисунок 5 – Первое разбиение функции F

Блок, обозначенный звездочкой, представляет собой не определенное еще множество КЛБ. Для проверки такой возможности выполняется разложение Шеннона по переменным x_a - x_d. Чтобы данное разбиение имело место, как и в прошлый раз, необходимо и достаточно, чтобы полученные подфункции разбивались на два однородных подмножества произвольного размера. В нашем случае, среди 16 подфункций от 6 переменных $x_e, x_f, \ldots, x_k$ должно быть два класса одинаковых подфункций. Таким образом, получаем промежуточную реализацию функции F в следующем виде: $F = f_3[f_1(x_a, x_b, x_c, x_d), x_e, x_f, x_g, x_h, x_j, x_k]$. Далее поступаем таким образом. Функция, реализуемая КЛБ по переменным x_a - x_d представляет собой не что иное, как булеву переменную – седьмой аргумент функции-блока со звездочкой. Далее применяется описанная методика декомпозиции для подфункции от семи переменных до тех пор, пока все аргументы не будут собраны в группы по 4. В случае если на каком-либо этапе декомпозиция не буде иметь место для всех возможных комбинаций аргументов, необходимо попытаться выполнить декомпозицию с общими аргументами, как было уже описано. Если же и такое разложение не будет иметь места, то данная функция может быть реализована в виде последовательно-параллельной структуры со значительным числом общих аргументов в разных КЛБ. Для этого заданная функция разбивается на группы термов СДНФ или СКНФ, каждая из которых реализуется на КЛБ описанными методами. Число таких разбиений настолько велико, что на каком-нибудь шаге будет получена реализация заданной функции на КЛБ. Недостатком, влияющим на скорость алгоритма, является перебор всех вариантов. Существуют методы, позволяющие реализовать такие функции на КЛБ более рационально и быстро, однако они выходят за рамки данной статьи и рассматриваться не будут.

6. Заключение

Исследования авторов показали, что применение описанных методов приводит к более выгодной реализации БФ на КЛБ, чем в некоторых САПР западного производства. Таких, например, как Synplicity 7.0. Критерием оценки служило число используемых для реализации КЛБ. Предлагаемые методы могут быть использованы при создании отечественных САПР цифровых устройств на микросхемах программируемой логики с архитектурой FPGA.

Литература

1. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608с.
2. Баркалов А.А., Красичков А.А. Методы декомпозиции булевых функций. // Наукові праці ДонНТУ. Серія: Інформатика, кібернетика та обчислювальна техніка, випуск 39: – Донецьк: ДонНТУ, 2002. – сс. 116–121.

Дата надходження до редколегії: 25.12.2003 р.