УДК 681.324

А.А. БАРКАЛОВ, А.В. МАТВИЕНКО, А.А. КРАСИЧКОВ

ОСОБЕННОСТИ РЕАЛИЗАЦИИ СИСТЕМ БУЛЕВЫХ ФУНКЦИЙ НА ПЛИС

Предлагается метод синтеза системы булевых функций на программируемых больших интегральных схемах с архитектурой FPGA. Метод ориентирован на число входов в комбинационных логических блоках. Рассмотрены модификации метода для двух и более входов комбинационных логических блоков. Приведены примеры применения предложенных методов.

В настоящее время требования к характеристикам средств вычислительной техники (ВТ) постоянно растут. Современная вычислительная система должна обладать высоким быстродействием, малыми габаритами и приемлемым временем проектирования. Для удовлетворения этих требований при проектировании цифровых схем ВТ используются программируемые логические интегральные схемы (ПЛИС). Наиболее широко применяются две разновидности ПЛИС: CPLD и FPGA [1]. Синтез цифровых устройств сводится к реализации системы булевых функций (БФ) в базисе конкретной ПЛИС. Архитектура CPLD функционально соответствует широко распространенным микросхемам программируемых логических матриц (ПЛМ), поэтому методы синтеза в этом базисе известны и особых сложностей не представляют [2]. Микросхемы FPGA подразумевают принципиально новые методы реализации систем булевых функций ввиду особенностей своей архитектуры. Алгоритмы функционального синтеза на FPGA являются ноу-хау фирм-производителей конкретных микросхем и встроены в различные САПР иностранного происхождения, которые не всегда отвечают 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.

Функциональный базис FPGA представляет собой конечное множество комбинационных логических блоков (КЛБ), которые могут реализовать любую булеву функцию от R переменных, либо две независимые функции от (R-1) переменных, где R – количество входов КЛБ. В производимых в настоящее время серийно микросхемах FPGA R = 2…5 [1]. Число переменных в БФ проектируемых систем обычно значительно больше и колеблется в широких пределах. Поэтому для реализации системы булевых функций от N переменных необходимо каждую функцию представить в виде совокупности подфункций от M переменных, где M≤ R. То есть возникает задача функциональной декомпозиции в базис с меньшим числом входов, для решения которой можно использовать следующие известные методы:

1. Минимизация с помощью карт Карно или законов алгебры логики. Этот метод применим, если функция задана в виде совершенной дизъюнктивной или конъюнктивной нормальной формы (СДНФ или СКНФ соответственно) и не всегда приводит к реализации функции на КЛБ, так как не ориентирован на него.

2. Использование законов Де-Моргана. Данный метод всегда приводит к реализации функции на КЛБ с любым R, однако число КЛБ значительно превышает минимально возможное для реализации заданной функции, так как каждый блок реализует отдельный терм СДНФ или СКНФ, а остальные блоки используются для объединения групп термов.

3. Разложение Шеннона. Этот метод также приводит к реализации любой функции на КЛБ, но число подфункций, из которых состоит реализуемая функция, исчисляется целыми 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 быстродействия, так как схема имеет древовидную структуру.

Поскольку КЛБ может реализовать все БФ R переменных, достаточно обеспечить представление заданной функции в виде минимального числа подфункций с числом переменных M ≤ R. Функция выражается в виде подфункций таким образом, чтобы пересечение по аргументам в различных подфункциях было минимальным. 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 последовательного и параллельного типов, представленных на рис. 1.

Быстродействие при параллельном соединении КЛБ выше, однако, как показали исследования авторов, большинство БФ реализуются в виде последовательного соединения.

Для реализации БФ на ПЛИС FPGA авторы предлагают следующие методы, отличающиеся параметром R.

1. Метод реализации функций на КЛБ с R = 2.

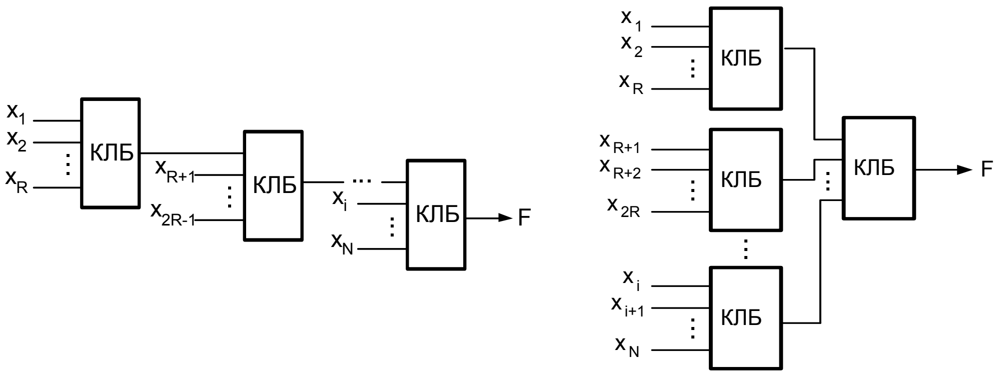

РИС. 1. Последовательное и параллельное соединения КЛБ

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 неизвестно, 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. Это приводит к реализации функции в виде параллельного соединения КЛБ. Такую возможность предлагается проверить с помощью разложения Шеннона и последующего анализа полученных подфункций. Для проверки возможности отделить от функции группы i аргументов $x_1, x_2, \ldots$, ..., x_i запишем:

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

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

ТАБЛИЦА 1

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

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

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

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

Из формул (1) и (2) следует

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

……..

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

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

- в первом подмножестве все элементы равны, во втором – равны нулю;
- в первом подмножестве все элементы равны, во втором – равны единице;
- в первом подмножестве все элементы равны, элементы второго есть инверсия элементов первого множества;
- элементы обоих подмножеств равны.

Разбиение получается после вынесения общих сомножителей за скобки и замены 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 [3].

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 осуществляется разложение Шеннона, причем подфункции $f_1', f_2', ..., f_K'$ представляются в виде функций не от N–I, а от N–I+K переменных, где K – число общих аргументов обеих подфункций.

Рассмотрим пример применения данного метода.

Пусть функция F задана в виде $F = ab \vee ac \vee cd \vee b\overline{c}\overline{d}$. Представим ее в виде $F = f_3[f_1(ab), f_2(cd)]$. Для этого выполним разложение Шеннона по переменным a и b:

$$F = \overline{a}\overline{b} \,\&\, (cd) \vee \overline{a}b \,\&\, (\overline{c \oplus d}) \vee a\overline{b} \,\&\, c \vee ab \,\&\, 1.$$

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 вышеописанному методу декомпозиции. Согласно методу функцию F можно представить также в виде $F = f_3[f_1(ab), f_2(acd)]$, т. е. с общим аргументом *a*. Для этого составим таблицу истинности следующего вида (табл. 2).

ТАБЛИЦА 2

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 таблице.

Далее попробуем представить функцию в виде $F = f_3[f_1(bc), f_2(ad)]$. Для этого выполним разложение Шеннона по переменным *b* и *c*:

$$F = \overline{b}\overline{c} \,\&\, 0 \vee \overline{b}c \,\&\, (a \vee d) \vee b\overline{c} \,\&\, (a \vee \overline{d}) \vee bc \,\&\, (a \vee d).$$

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).

Из табл. 3 видно, что такое разбиение существует. Это следует из того, что $f_1^{'} = 0$, а подфункции $f_2^{'}, f_3^{'}$ и $f_4^{'}$ можно склеить в одну, используя неопределенности $f_2^{'} = f_3^{'} = f_4^{'} = a \vee (\overline{c \oplus d})$, 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. 2.

2. Метод реализации функций на КЛБ с R>2.

Рассмотрим метод реализации функции на КЛБ с R=3….5 для случая, когда функция F является БФ от 10 переменных: F = F(x_1, x_2, …, x_{10}) и R = 4. При реализации в виде последовательного соединения КЛБ применяется следующий метод.

ТАБЛИЦА 3

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

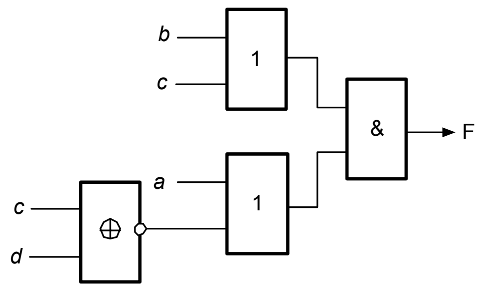

РИС. 2. Реализация функции F на КЛБ

Вначале проверяется возможность разбиения функции на две подфункции – от трех и семи аргументов, что соответствует ее реализации схемой, показанной на рис. 3.

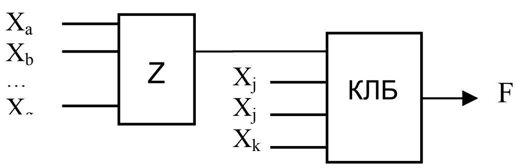

РИС. 3. Начальная стадия декомпозиции функции F

Блоком, отмеченным Z, обозначим пока еще неопределенное множество КЛБ. В дальнейшем будем называть его Z-блок. Выполним разложение Шеннона по переменным $x_a - x_g$. Необходимым и достаточным признаком существования такого разбиения является выполнение условия: полученные подфункции разбиваются на два подмножества произвольного размера, в каждом из которых подфункции равны. В данном случае среди 128 подфункций от переменных x_i, x_j, x_k должно быть два класса одинаковых подфункций. Промежуточная реализация функции $F = f_3[f_1(x_a, x_b, ..., x_g), x_i, x_j, x_k]$. Далее поступаем таким же образом с подфункцией f_1, а затем и с ее подфункциями до тех пор, пока на очередном шаге число аргументов не станет меньше либо равным четырем. Если на каком-либо этапе декомпозиция неосуществима для всех возможных комбинаций аргументов, необходимо попытаться выполнить декомпозицию с общими аргументами. Число таких комбинаций значительно больше, чем при декомпозиции без общих аргументов. Если же и такое разложение невозможно, то данная функция может быть реализована в виде параллельного соединения КЛБ, для реализации которой справедлива следующая методика.

Вначале проверяется возможность разбиения функции на две подфункции – от четырех и шести аргументов, что соответствует ее реализации схемой, показанной на рис. 4.

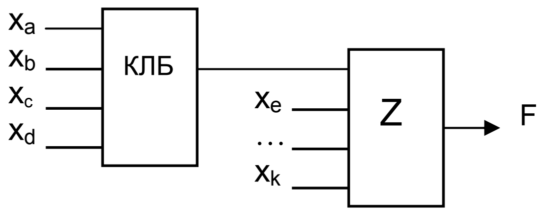

РИС. 4. Начальное разбиение функции F

Для этого выполняется разложение Шеннона по переменным x_a–x_d. Чтобы такое разбиение существовало, как и ранее, необходимо и достаточно, чтобы полученные подфункции разбивались на два однородных подмножества произвольного размера. В данном случае среди 16 подфункций от 6 переменных x_e, $x_f, \ldots, x_k$ должно быть два класса одинаковых подфункций.

Промежуточная реализация функции

$$F = f_3[f_1(x_a, x_b, x_c, x_d), x_e, x_f, x_g, x_h, x_j, x_k].$$

Функция, реализуемая КЛБ по переменным x_a - x_d, представляет собой булеву переменную для Z-блока. Далее применяется описанная методика декомпозиции для подфункции от семи переменных до тех пор, пока все аргументы не будут собраны в группы по четыре. Если на каком-либо этапе декомпозиция неосуществима для всех возможных комбинаций аргументов, необходимо попытаться выполнить декомпозицию с общими аргументами, как было описано выше. Если и такое разложение невозможно, функцию можно представить в виде последовательно-параллельного соединения КЛБ со значительным числом общих аргументов в разных КЛБ. Для этого функция разбивается на группы термов СДНФ или СКНФ, каждая из которых реализуется на КЛБ описанными методами. Недостатком такого подхода является необходимость перебора всех вариантов разбиения. Авторы разработали методы, позволяющие реализовать такие функции на КЛБ с сужением пространства поиска, однако они выходят за рамки данной статьи.

Исследования показали, что применение предложенных методов приводит к более экономичной по числу КЛБ реализации системы булевых функций на КЛБ, чем в некоторых САПР западного производства [1]. Предлагаемые методы могут быть использованы при создании отечественных САПР цифровых устройств на микросхемах программируемой логики с архитектурой FPGA.

1. *Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П.* Проектирование систем на микросхемах программируемой логики. – СПб.: БХВ-Петербург, 2002. – 608 с.
2. *Баркалов А.А., Палагин А.В.* Синтез микропрограммных устройств управления. – Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 1997. – 136 с.
3. *Баркалов А.А., Красичков А.А.* Методы декомпозиции булевых функций // Науч. тр. ДонНТУ. Сер. ИКВТ-2002. – 2002. – Вып. 39. – С. 72 – 80.

Получено 02.03.2004

.