

УДК 004.312.02

Н.Г. КОРОБКОВ<sup>1</sup>, Е.Н. КОРОБКОВА<sup>2</sup><sup>1</sup>Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Украина<sup>2</sup>Белгородский государственный технологический университет, Россия

## СИНТЕЗ УНИВЕРСАЛЬНЫХ ЛОГИЧЕСКИХ МОДУЛЕЙ С ПАМЯТЬЮ

Предложен метод синтеза универсальных логических модулей с памятью, основанный на представлении функций в форме обобщённых логических функций с зависимыми параметрами.

**логический модуль, синтез, кодирование, оптимизация, обобщённая логическая функция, зависимые параметры**

### Введение

**Общая характеристика проблемы.** Большинство существующих методов обеспечения отказоустойчивых цифровых устройств основано на структурном резервировании каналов. Одним из недостатков такого способа является увеличение аппаратных затрат, особенно если, исходя из требований минимальной задержки, реализация логических функций осуществляется двухъярусными комбинационными схемами, ориентированными на использование ПЛИС, что при сложном характере функций экономически не оправдано. Снизить аппаратные затраты можно за счет использования скобочных форм представления функций, которым соответствуют многоярусные комбинационные схемы. Однако такой способ реализации логических функций приводит к значительному увеличению задержки, и, следовательно, снижение производительности системы в целом [1].

**Анализ исследований, посвящённых решению проблемы.** Увеличение производительности многоярусных схем можно осуществить за счёт конвейерной передачи информации от яруса к ярусу с промежуточным ее запоминанием [2]. Для этой цели между ярусами устанавливаются регистры памяти. В быстродействующих конвейерных устройствах используют регистры, выполненные на триггерах Эрла [3]. Несмотря на сложность этих структур, они

находят применение, поскольку их быстродействие максимально возможное.

Функцию выхода триггера Эрла, свободную от состязаний можно представить следующим образом:

$$Q^{n+1} = DL \vee Q^n \bar{L} \vee Q^n D.$$

Из приведенного соотношения видно, что если результат преобразования данных некоторая функция других переменных, то представляя ее в ДНФ и подставляя в уравнение триггера, порядок этого уравнения, а, следовательно, и задержки в преобразовании и запоминания информации не изменится.

Синтез триггеров Эрла с фиксированным характером функции  $D = F(D_{n-1}D_{n-2} \dots D_0)$  не вызывает проблем [3]. Задача несколько усложняется при построении резервированных систем с перестраиваемым характером функции от информационных переменных. Настройка на выполнение той или другой функции осуществляется значениями настроечных переменных. Следовательно, по сути ставится задача синтеза универсального логического модуля (УЛМ) и триггера Эрла как единого целого.

Такую логическую структуру будем трактовать как УЛМ с памятью.

При этом усложнение задачи состоит не только и возможно даже не столько от заданного списка реализуемых функций. Основная сложность состоит в выборе оптимального варианта настройки модуля на выполнение заданных функций, поскольку от него

зависит сложность схемной реализации функции  $D$  как функции от информационных и настроечных переменных.

**Цель работы** – совершенствование метода синтеза УЛМ с памятью.

### Метод решения

Для нахождения оптимального варианта настройки, обеспечивающего получение минимально возможного представления функции  $D$  в ДНФ как функции от информационных и настроечных переменных, предлагается рассматривать эту функцию как ОЛФ с зависимыми параметрами, трактуя настроечные переменные как первичные, а информационные переменные как вторичные [4].

При этом размещение реализуемых функций в точках области определения должно быть выполнено таким образом, чтобы все множество их можно было бы покрыть минимально возможным числом импикант минимального ранга.

Если число частичных функций равно  $N$ , то число настроечных переменных  $r$  можно найти как  $r = \log_2 N$ , округленное до большего целого.

В этом случае число точек области определения будет равно  $2^r$ . Если это число будет больше заданного числа реализуемых функций, то при размещении заданных функций в точках области определения некоторые точки окажутся не востребованными. Эти точки трактуются как избыточные, используя их в дальнейшем для доопределения.

Поскольку число точек области определения равно  $2^r$ , то число всех возможных вариантов размещения заданных функций будет равно числу перестановок, т.е.  $2^r!$ . Даже при относительно небольшой длине списка реализуемых функций число вариантов размещения довольно велико и на первый взгляд задача выбора оптимального варианта кажется довольно сложной.

И это действительно так, если рассматривать

реализуемые функции независимо одна от другой и пытаться разместить каждую из них в наиболее оптимальную точку области определения. При таком подходе трудно зрительно оценить все возможные варианты при ручном способе, а при программном перебрать их и выбрать один или несколько оптимальных.

Предлагается алгоритм нахождения оптимального варианта размещения, основанный на представлении множества функции заданного списка в СДНФ в виде логической суммы минтермов, образуемых литералами информационных переменных ( $m_i = \bar{D}_1 \bar{D}_0$ ). Заданные функции представляем в общем виде, с последующим разбиением множества их на несколько подмножеств, удовлетворяющих некоторым условиям толерантности, обеспечивающих покрытие каждого из выделенных подмножеств произведением минимально возможного ранга, образуемым литералами настроечных переменных. На первом шаге алгоритма литералы настроечных переменных не конкретизируются.

Анализ предлагаемого алгоритма проведем на примере проектирования УЛМ с памятью, программируемого на выполнение всех функций от двух информационных переменных  $D_1 D_0$ . Такой модуль может найти применение не только в конвейерных системах, но и при построении логических блоков АЛУ с памятью.

Число этих функций, как известно, равно шестнадцати. Запишем эти функции, представляя каждую из них в явной форме и в виде логической суммы минтермов, образуемых литералами информационных переменных.

$$f_1 = 0 = 0;$$

$$f_2 = 1 = m_0 \vee m_1 \vee m_2 \vee m_3;$$

$$f_3 = D_0 = m_1 \vee m_3;$$

$$f_4 = \bar{D}_0 = m_0 \vee m_2;$$

$$f_5 = D_1 = m_2 \vee m_3;$$



$P_2$  и литерала  $\tilde{a}_l$  на логическую сумму функций, входящих в подмножество  $P_3$ .

В каждое из перечисленных подмножеств входит функция, равная одному из минтермов, и остальные семь функций, представленных логическими суммами, содержащими этот минтерм.

Логическую сумму всех функций в каждом из подмножеств можно представить равной соответствующему общему минтерму.

Логическая сумма функций, входящих в подмножество  $P_0$ , равна  $m_0, P_1 - m_1, P_2 - m_2, P_3 - m_3$ .

Следовательно, в общем виде минимальная ДНФ функции  $D$  может быть представлена следующим образом:

$$D = m_0\tilde{a}_i \vee m_1\tilde{a}_k \vee m_2\tilde{a}_j \vee m_3\tilde{a}_l.$$

Каждый литерал настроечной переменной, входящий в то или другое произведение, может быть как прямое, а также инверсное значение любой из этих переменных. Отсюда следует, что если рассматривать все возможные комбинации литералов, начиная от четырех инверсных значений и заканчивая четырьмя прямыми, то число комбинаций будет равно числу  $2^4$ .

В свою очередь, в каждой комбинации можно выполнить все возможные перестановки из четырех литералов, число которых равно  $4!$ . Следовательно, общее число вариантов минимальной ДНФ функции  $D$  будет равно  $2^4 \cdot 4! = 384$ . Все эти варианты по сложности равноценны. Отличие их будет только лишь в кодах настройки на выполнение той или другой функции заданного списка.

Рассмотрим вариант, который минимален по числу дополнительных инверторов

$$\tilde{a}_i = a_0, \tilde{a}_k = a_1, \tilde{a}_j = a_2, \tilde{a}_l = a_3.$$

Подставляя литералы настроечных переменных в приведенное выше выражение, получаем минимальную ДНФ функции  $D$  для первого варианта:

$$D = m_0a_0 \vee m_1a_1 \vee m_2a_2 \vee m_3a_3.$$

Раскрывая значения минтермов и подставляя

значение функции в общее уравнение триггера Эрла, получаем уравнение проектируемого УЛМ:

$$Q^{n+1} = \bar{D}_1\bar{D}_0La_0 \vee \bar{D}_1D_0La_1 \vee D_1\bar{D}_0La_2 \vee D_1D_0La_3 \vee Q^n\bar{L} \vee \bar{D}_1\bar{D}_0Q^n a_0 \vee \bar{D}_1D_0Q^n a_1 \vee D_1\bar{D}_0Q^n a_2 \vee D_1D_0Q^n a_3$$

Схема модуля, построенная в соответствии с полученным выражением (с целью сокращения объема статьи она не приведена), при разрыве цепи обратной связи с выхода  $Q$  имеет второй порядок, которому соответствует максимально возможное быстроедействие при реализации и запоминании любой функции от двух переменных.

Аналогичным образом могут быть построены УЛМ с памятью, обеспечивающие реализацию функций от большего числа переменных.

Для нахождения кодов настройки на реализацию функций заданного списка строим карту, размечая её значениями литералов и определяя номера клеток, образуемых значениями настроечных переменных, покрывающих эти клетки (рис. 2).

На основании полученной карты можно построить таблицу настройки, представляя коды настройки, образуемые переменными  $a_3a_2a_1a_0$ , представленные в двоичной системе, а реализуемые функции на каждом наборе этих переменных представляем в СДНФ и явной форме (табл. 1).

	_____				_____				$a_1$
	_____				_____				$a_0$
	0	1	3	2	4	5	7	6	
	-	-	m <sub>0</sub>	-	m <sub>0</sub>	m <sub>1</sub>	-	m <sub>1</sub>	
	-	-	-	-	-	-	-	-	
	-	-	m <sub>0</sub>	-	m <sub>0</sub>	m <sub>1</sub>	-	m <sub>1</sub>	
m <sub>2</sub>	-	-	m <sub>2</sub>	-	m <sub>2</sub>	-	m <sub>2</sub>	-	
	-	-	m <sub>0</sub>	-	m <sub>0</sub>	m <sub>1</sub>	-	m <sub>1</sub>	
m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	m <sub>2</sub>	
	8	9	11	10					
	-	m <sub>2</sub>	-	m <sub>2</sub>	-	m <sub>2</sub>	-	m <sub>2</sub>	
	-	m <sub>2</sub>	-	m <sub>2</sub>	-	m <sub>2</sub>	-	m <sub>2</sub>	
a <sub>3</sub>	a <sub>2</sub>	a							

Рис. 2. Карта настройки модуля

Таблица 1  
Таблица настройки первого варианта кодирования

Код настройки				Выполняемая функция	
$a_3$	$a_2$	$a_1$	$a_0$	в СДНФ	в явном виде
0	0	0	1	$m_0$	$\overline{D_0} \vee D_1$
0	0	1	0	$m_1$	$\overline{D_1} D_0$
0	0	1	1	$m_0 \vee m_1$	$\overline{D_1}$
0	1	0	0	$m_2$	$D_1 \overline{D_0}$
0	1	0	1	$m_0 \vee m_2$	$\overline{D_0}$
0	1	1	0	$m_1 \vee m_2$	$D_0 \oplus D_1$
0	1	1	1	$m_0 \vee m_1 \vee m_2$	$\overline{D_0} \overline{D_1}$
1	0	0	0	$m_3$	$D_0 D_1$
1	0	0	1	$m_0 \vee m_3$	$\overline{D_0} \oplus D_1$
1	0	1	0	$m_1 \vee m_3$	$D_0$
1	0	1	1	$m_0 \vee m_1 \vee m_3$	$\overline{D_1} \vee D_0$
1	1	0	0	$m_2 \vee m_3$	$D_1$
1	1	0	1	$m_0 \vee m_2 \vee m_3$	$\overline{D_0} \vee D_1$
1	1	1	0	$m_1 \vee m_2 \vee m_3$	$D_0 \vee D_1$
1	1	1	1	$m_0 \vee m_1 \vee m_2 \vee m_3$	1

## Выводы

**Научная новизна.** Впервые предложен метод синтеза универсального логического модуля с памятью, основанный на представлении функций в обобщённой форме.

**Практическая ценность.** Реализация данного подхода даёт возможность получения оптимальных программируемых логических структур с памятью, практически исключив процедуру перебора.

**Одним из направлений дальнейших исследований** является описание предложенного способа синтеза в терминах структур данных и функций на языках высокого уровня.

## Литература

1. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических инте-

гральных схем. – М.: Горячая линия – Телеком, 2001. – 636 с.

2. Коуги П.М. Архитектура конвейерных ЭВМ. – М.: Радио и связь, 1985. – 360 с.

3. Пухальский Г.И., Новосельцева Т.Я. Цифровые устройства. – СПб.: Политехника, 1996. – 885 с.

4. Коробкова Е.Н. Графоаналитический метод минимизации полностью определенных логических функций // Системи обробки інформації. – Х.: ХВУ, 2002. – Вип. 6 (22). – С. 288-298.

Поступила в редакцию 13.03.2007

**Рецензент:** д-р техн. наук, проф. О.Е. Федорович, Национальный аэрокосмический университет им. Н.Е. Жуковского “ХАИ”, Харьков.