

СИНТЕЗ ЛОГИЧЕСКОЙ ФУНКЦИИ ПРОЕКТИРУЕМОЙ КОМБИНАТОРНОЙ СТРУКТУРЫ СИСТЕМЫ ЗАЩИТЫ

Введение

Научно-технический прогресс в области электроники и вычислительной техники за последние годы настоятельно потребовал изменения методик комбинаторных структур систем с учетом применения новой элементной базы. При проектировании систем защиты информации, особенно создании аппаратуры построения помех необходимо разработать сложные электронные системы с использованием методов синтеза.

Основная часть

В основе традиционных методов синтеза лежит представление произвольной функции с помощью ограниченного набора операций, образующих функционально полную систему, т.е. любую логическую функцию можно представить в совершенной дизъюнктивно-нормальной или в совершенной конъюнктивно - нормальной форме. Полученные формы представления функции являются функционально полными и не всегда простыми. Для их упрощения используются логические тождества и методы минимизации. При минимизации функций с относительно небольшим числом переменных ($k \leq 6$) наиболее удобны и наглядным является графический метод, использующий карты Карно. Причем с ростом числа переменных ($k > 6$) карты Карно становятся громоздкими и неудобными для практического применения.

В течение последних лет для генерирования логических функций часто используются стандартные интегральные микросхемы, что позволяет уменьшить число физических блоков проектируемой логической структуры, а также ее стоимость. Одной из таких перспективных интегральных микросхем со средним уровнем интеграции является мультиплексор. Использование мультиплексора как генератора логической функции требует разработки соответствующих методов логического проектирования. Если количество переменных реализуемой булевой функции увеличивается, то необходимо использовать более одного уровня мультиплексирования или при минимизации исходной булевой функции применять несколько карт Карно для истинной функции и ее дополнения [1]. Это приводит к построению многоуровневых мультиплексорных деревьев, увеличивает время разработки и аппаратные затраты. Поэтому необходима эффективная методика синтеза логических функций, ориентированная на использование мультиплексоров и сводящая задачу синтеза к последовательной декомпозиции заданной функции на более простые, имеющие меньшее количество переменных, с последующей их минимизацией.

Рассмотрим выражение для выходного сигнала мультиплексора:

$$F = \sum_{i=1}^{2^k} I_i \sigma_i \quad (1)$$

В данном выражении I_i - значение функции на i -м адресном входе мультиплексора, соответствующее i -му набору переменных на его управляющих входах; σ_i - значение i -го набора переменных на управляющих входах, $\sigma_i = f_i(x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, x_k, \bar{x}_k)$, где k - количество переменных в наборе, соответствующих количеству входов управления мультиплексора; 2^k - количество наборов переменных, соответствующих количеству адресных входов мультиплексоров.

Из выражения (1) видно, что у мультиплексора имеется для каждого набора переменных на его входах управления соответствующий адресный вход, т.е. применения мультиплексора для решения задачи построения логической функции N переменных сводится к задаче построения 2^k функций ($N-k$) переменных, подаваемых на адресные входы мультиплексора. Например, для четырехвходового мультиплексора выражение для выходного сигнала имеет вид:

$$F = I_1 \bar{x}_2 \bar{x}_1 + I_2 \bar{x}_2 x_1 + I_3 x_2 \bar{x}_1 + I_4 x_2 x_1$$

Применение четырехходового мультиплексора для решения задачи построения логической функции N переменных сводится к задаче построения четырех остаточных функций $(N-2)$ переменных, подключенных на его адресные входы. Аналогичным образом восьмивходовый мультиплексор позволяет исключить три переменные, шестнадцативходовый – четыре переменные и т.д.

Следовательно, исключение переменных, связанных с адресными входами, обеспечивает минимальные логические функции.

Для генерирования остаточных логических функций, связанных с адресными входами мультиплексора, выбираются такие схемы, которые обеспечивают получение тривиальных или неоднократно встречающихся остаточных функций.

Применение мультиплексоров для проектирования комбинационных схем оправдано при реализации сложных функций большого количества переменных. Для реализации простых функций небольшого количества переменных (до двух) оптимальнее использовать вентильные схемы. Однако, когда необходима реализация сложных функций преобразователей кодов или реализация функции возбуждения управляющей памяти блоков управления с жесткой логикой или шифрация заданной функции, применение мультиплексоров бесспорно оправдано. И здесь весьма кстати сочетание методов синтеза и минимизации комбинационных схем на базе мультиплексоров с применением метода декомпозиции исходных функций.

Пусть имеется функция N переменных. Эту логическую функцию большого количества переменных представим в виде композиции функций меньшего количества переменных в виде:

$$F(x_1, x_2, x_3, \dots, x_N) = x_1 F_0(0, x_2, x_3, \dots, x_N) + x_1 F_1(1, x_2, x_3, \dots, x_N), \quad (2)$$

где x_1 - выделяемая переменная; $F_0(0, x_2, x_3, \dots, x_N)$ - функция, получаемая из исходной функции F подстановкой значения $x_1=0$; $F_1(1, x_2, x_3, \dots, x_N)$ - функция, получаемая из исходной функции F подстановкой значения $x_1=1$

Полученные функции F_0 и F_1 подвергаются дальнейшей декомпозиции. Процесс выделения более простых функций производится до выделения такой функции, которая наиболее оптимально минимизирована с помощью мультиплексоров. Такими оптимальными функциями могут быть в большинстве случаев функции четырех, пяти, или шести переменных. Дальнейшее выделение функций меньшего количества переменных значительно увеличивает количество необходимых вентильных элементов, применяемых в схеме наряду с мультиплексорами, а выделение функции с количеством переменных больше шести вынуждает прибегать к наращиванию каналов в мультиплексорах, т.е. создавать сложные мультиплексорные структуры, что ведет к увеличению количества корпусов интегральных схем для заданной функции.

Таким образом, оптимальный вариант можно найти только в каждом конкретном случае, когда известна заданная логическая функция и элементная база для ее реализации. Тогда, имея таблицу истинности разработчик имеет возможность в конкретном варианте схемы оценить на каждом этапе декомпозиции целесообразно остановиться, какой разрядности мультиплексор применить для реализации функции с минимальными аппаратными затратами.

Исходя из этого, для функции N переменных выделяется такое количество переменных, чтобы в остаточной логической функции их осталось две. Объясняется это тем, что для генерации функции двух переменных $F(x_0, x_1)$ необходимо семь инверторов и пять двухходовых вентилях И-НЕ. В данном случае генерируется 16 функций двух переменных, шесть из которых являются тривиальными:

$$x_0, \overline{x_0}, x_1, \overline{x_1}, 1, 0 \text{ и функции } \overline{x_0}, \overline{x_0}x_1, \overline{x_0}x_1, x_0, \overline{x_0}x_1, x_0 + x_1, \overline{x_0} + x_1, x_0 + x_1, x_0 + \overline{x_1}, x_0 + x_1, x_0x_1 + \overline{x_0}x_1, x_0x_1 + \overline{x_0}x_1.$$

Поскольку для N переменных можно образовать $l_N = 2^{2^N}$ различных функций, то при увеличении количества переменных чрезвычайно быстро растет количество функций, необходимых для их генерации. Поэтому создавать генератор остаточной функции трех, а тем более четырех переменных на базе логических вентилях в аппаратном плане

достаточно громоздко и неэкономично по сравнению с генерацией функций двух переменных.

На основании вышеизложенного предлагается методика синтеза логической функции проектируемой комбинаторной структуры.

1. Строится таблица истинности синтезируемой логической функции на базе технического задания на разработку. Такая таблица истинности содержит все возможные наборы значений логических переменных $x_0, x_1, x_2, \dots, x_{N-1}$ и значения функций $F_0, F_1, F_2, \dots, F_{N-1}$, соответственно каждому из наборов.

2. Осуществляется переход от таблицы истинности к алгебраическому представлению логических функций в совершенной дизъюнктивно-нормальной форме.

3. В случае, если полученная функция является сложной функцией многих переменных ($k > 6$), производится декомпозиция логической функции, т.е. логическую функцию большого количества переменных представляют в виде композиции функций меньшего количества переменных согласно выражению (2).

4. Полученные композиции рассматриваются как отдельные функции, которые строятся и минимизируются на базе мультиплексоров. Для каждой функции меньшего количества переменных производится обратный переход от ее алгебраического представления к табличному (в виде управляющей таблицы). Указанная таблица заполняется следующим образом:

- в первом столбце представляется порядковый номер каждого набора переменных функций;

- во втором столбце представляются значения всех наборов переменных (все минтермы функции N переменных, количество которых равно 2^N);

- в третьем столбце представляются значения функции для каждого минтерма.

5. Для справок составляется карта Карно, в каждой клетке которой отмечаются номера минтермов данной функции; а также незаполненная карта Карно.

6. Исходя из полученной управляющей таблицы, незаполненная карта Карно заполняется путем записи единиц в клетки с порядковыми номерами минтермов, соответствующее значение функции для которых равно единице.

7. Полученная карта Карно анализируется и для функции N переменных выбираются две переменные остаточной функции, подаваемые на адресный вход мультиплексора. Чтобы остаточная функция была тривиальной, при анализе полученной карты рассматриваются все возможные варианты выделения функций двух переменных и определяется оптимальный вариант.

Для функции четырех переменных процесс определения оптимального варианта выделения двух переменных может быть автоматизирован.

8. Каждый участок карты Карно остаточной функции рассматривается как отдельная функция двух переменных и минимизируется. Полученное значение минимизированной функции с данного участка является входной функцией (настроечной) соответствующего адресного входа мультиплексора.

9. После реализации отдельных функций исходной декомпозированной функции составляется сама исходная функция.

10. Если исходная функция не требует своей декомпозиции и может быть сразу построена и минимизирована на базе мультиплексора то необходимо выполнить пункты 4-8 данной методики.

Выводы

Предлагаемая выше методика синтеза логических функций проектируемой комбинаторной структуры ориентирована на использование мультиплексоров и сводит задачу к последовательной декомпозиции заданной функции на более простые, имеющие меньшее количество переменных, с последующей их минимизацией.

Список литературы

1. Голдсуорт Б. – Проектирование цифровых логических устройств/Голдсуорт Б. –М.:

Машиностроение, 1985. – 297 с.

2. Молчанов А.А. – Моделирование и проектирование сложных систем / Молчанов А.А. – К.: Вища школа, 1998. – 358 с.

3. Закревский А.Д. Логический синтез каскадных схем. М.: Наука 1999. – 215 с.

4. Браммер Ю.А. Цифровые устройства: Учеб. пособие для вузов. –М.:Высш. шк., 2006. –229с.

Рецензент: д.т.н., проф. Хорошко В.О.

Надійшла 22.03.2010 р.