

ОПЕРАЦИОННОЕ ФОРМИРОВАНИЕ КОДОВ СОСТОЯНИЙ В МИКРОПРОГРАММНЫХ АВТОМАТАХ

Ключевые слова: микропрограммный автомат, граф-схема алгоритма, состояние, операция перехода, оптимизация аппаратных затрат.

ВВЕДЕНИЕ

Устройство управления (УУ) является одним из центральных узлов современных цифровых вычислительных систем, во многом определяющее основные характеристики системы [1–3]. Из традиционных способов построения УУ выделим их представление в виде конечных автоматов Мили или Мура с последующей реализацией в виде управляющего автомата (УА) с «жесткой» логикой (также называемого микропрограммным автоматом (МПА)) в соответствии с принципом микропрограммного управления [2].

Задача проектирования УУ постоянно возникает при разработке цифровой системы, поскольку для различных систем именно УУ необходимо каждый раз проектировать заново. Поскольку УУ определяется алгоритмом функционирования, который может быть реализован различными конечными автоматами, задача синтеза УУ сводится к выбору структуры, обеспечивающей выполнение критериев оптимизации на используемой элементной базе.

Сложность задач, решаемых вычислительными системами, постоянно увеличивается, отсюда актуальность задачи разработки новых эффективных структур УУ, в которых оптимизированы стоимостные характеристики схемы устройства. Цель данной работы — разработка новой структуры МПА, использующей особый принцип реализации микропрограммных переходов и позволяющей снизить аппаратные затраты в схеме устройства.

КАНОНИЧЕСКИЙ МЕТОД СИНТЕЗА МПА

Данный метод основан на теореме о структурной полноте, доказанной В.М. Глушковым [1]. Результатом применения этого метода является система уравнений, задающая зависимость выходных сигналов и сигналов, подаваемых на входы запоминающих элементов, от сигналов, поступающих на входы автомата извне, и сигналов, снимаемых с выходов запоминающих элементов. Эти уравнения называются каноническими.

Следует отметить, что традиционная форма представления канонических уравнений структурного автомата — булевы функции (БФ), объединяющие булевы переменные с помощью логических операций. При этом в абстрактной постановке задачи синтеза автомата используется два критерия:

- 1) минимальное число состояний в автомате;
- 2) нахождение минимальной дизъюнктивной формы (ДНФ) функций переходов и функций выходных сигналов.

Основное преимущество БФ заключается в том, что методы их синтеза и минимизации, ориентированные на различные элементные базисы, достаточно изучены [4]. Современные элементные базисы программируемых логических интегральных схем (ПЛИС) типа FPGA, CPLD поддерживают синтез булевых выражений, а многие САПР ПЛИС имеют встроенные средства минимизации логических схем [5, 6].

© А.А. Баркалов, Р.М. Бабаков, 2011

В то же время при использовании БФ в качестве математической основы реализации уравнений для функций переходов МПА следует учитывать, что сложность систем булевых уравнений возрастает пропорционально увеличению числа переходов заданного алгоритма функционирования автомата. Увеличение числа переходов приводит как к увеличению количества булевых термов в функциях переходов, так и к увеличению числа литералов в термах в силу увеличения разрядности кода состояния. Минимизация ДНФ булевых функций может эффективно выполняться только при полном переборе возможных решений, что является трудоемкой и не всегда реальной задачей [7]. Различные эвристические методы, позволяющие сократить количество переборов, не гарантируют получения минимальных ДНФ [8].

Также следует помнить, что при ограниченном числе входов логических элементов используемого элементного базиса возможно увеличение количества уровней преобразования логических сигналов, что снижает быстродействие схемы устройства. Данная особенность справедлива как для элементного базиса дискретных элементов, так и для современных БИС.

Проведенный авторами обширный анализ литературы (больше тысячи наименований) показал, что в настоящее время для МПА не существует методов организации схемы формирования функций переходов, позволяющих ограничить рост аппаратных затрат в комбинационной схеме автомата при увеличении числа микропрограммных переходов. Поскольку коды состояний в граф-схеме алгоритма управления (ГСА) уникальны, увеличение аппаратных затрат с ростом сложности ГСА — естественный процесс и частично может компенсироваться применением различных методов оптимизации схем МПА. При этом для некоторых элементных базисов зависимости, отражающие рост аппаратных затрат от сложности ГСА, можно получить аналитически [2, 9, 10].

В настоящей работе предлагается новый подход к структурной организации МПА, позволяющий отказаться от представления функций переходов в виде булевых выражений и улучшить аппаратные характеристики схемы автомата. Идея подхода заключается в том, что коды состояний рассматриваются и обрабатываются как арифметико-логические величины.

ОПЕРАЦИОННОЕ ПРЕОБРАЗОВАНИЕ КОДОВ СОСТОЯНИЙ

Задача УА в составе вычислительной системы — формирование последовательности наборов микроопераций (МО), поступающих в операционный автомат (ОА) в соответствии с некоторым заданным алгоритмом функционирования. ОА под воздействием набора МО выполняет некоторую обработку данных. При этом такие операционные узлы, как сумматор, вычитатель, умножитель, сдвигатель и т.п., ассоциируются именно с операционным автоматом, поскольку в традиционных структурах УА в принципе нет данных, которые необходимо обрабатывать с помощью сложных операционных узлов. За редким исключением (использование счетчика адресов в композиционных микропрограммных устройствах управления [3]) обработка логических сигналов внутри УА проводится с помощью булевых функций или табличным способом (например, принудительная адресация микрокоманд в микропрограммном УУ, где адреса переходов заданы в таблице, задающей микропрограммную память).

При реализации заданного алгоритма управления в качестве альтернативы канонической структуре МПА предлагается управляющий автомат с операционным формированием переходов (УА с ОФП), имеющий следующие характерные особенности.

1. Код состояния, являясь уникальным (в пределах заданной ГСА) набором

бит, рассматривается и как уникальная арифметическая величина (некоторое число), преобразование которой может выполняться с помощью логических и арифметических операций.

2. Преобразование кода текущего состояния $K(a^t)$ в код следующего состояния (состояния перехода) $K(a^{t+1})$, где $t = 0, 1, \dots$ — автоматное время, выполняется с использованием операции перехода (ОП). Под ОП понимается арифметико-логическая операция над кодом текущего состояния, реализуемая отдельной комбинационной схемой и характеризуемая аппаратными затратами и быстродействием. Для реализации ОП могут использоваться стандартные функциональные узлы (в виде дискретных элементов или на базе современных ПЛИС): сумматоры, сдвигатели, умножители, мультиплексоры и др.

3. Основу структуры УА с ОФП составляет локальный операционный автомат (ЛОА). В нем преобразование кодов состояний выполняется с помощью множества ОП, число которых в общем случае меньше числа переходов в граф-схеме заданного алгоритма управления. Выбор ОП для того или иного перехода определяется кодом операции, формируемым на основании значения кода текущего состояния.

Объясним суть операционного формирования переходов на примере. Пусть в заданной ГСА имеется безусловный переход из состояния a_i с кодом $K(a_i) = 0011101_2$ в состояние a_j с кодом $K(a_j) = 1010111_2$. При каноническом методе синтеза каждому разряду $K(a_j)$, равному единице (в случае синтеза по ДНФ), должен соответствовать терм, определяемый значениями разрядов $K(a_i)$. Поскольку в $K(a_j)$ имеется пять разрядов, равных единице, данный переход добавляет в комбинационную схему пять термов (каждый состоит из семи переменных). Данные термы в силу уникальности кодов состояний не могут использоваться для формирования никакого другого перехода в данной ГСА.

Рассмотрим значения $K(a_i)$ и $K(a_j)$ как арифметические величины в десятичной системе счисления. В этом случае $K(a_i) = 29$, $K(a_j) = 87$. Заметим, что код следующего состояния можно получить таким образом: $K(a_j) = K(a_i) + 2 \cdot K(a_i)$. Поскольку в двоичной системе счисления умножение на 2 может быть реализовано операцией двоичного сдвига влево на один разряд, то

$$K(a_j) = K(a_i) + L_1(K(a_i)). \quad (1)$$

Таким образом, преобразование данного кода текущего состояния в код следующего состояния реализуется схемой сдвига и суммирования, которые обычно являются типовыми функциональными узлами операционных автоматов, т.е. библиотечными элементами современных САПР. При этом вполне возможно, что в данной ГСА существуют переходы, которые можно осуществить с помощью выражения (1) (например, переход из a_k с кодом $K(a_k) = 0001110_2 = 14$ в состояние $K(a_m) = 0101010_2 = 42$), что позволит использовать одну и ту же комбинационную схему для реализации нескольких переходов. Кроме того, всегда имеется возможность задать коды состояний таким образом, чтобы одна и та же КС использовалась для как можно большего числа переходов. Выражения, подобные (1), позволяющие реализовать микропрограммные переходы с помощью арифметико-логических операций, назовем операциями перехода.

Предлагаемая структуру МПА Мура с операционным формированием кодов состояний показана рис. 1.

Здесь операционная часть (ОЧ) формирует код следующего состояния Φ на основании кода текущего состояния T , сигналов логических условий (ЛУ) X и кода операции (КО):

$$\Phi = \Phi(T, X, КО). \quad (2)$$

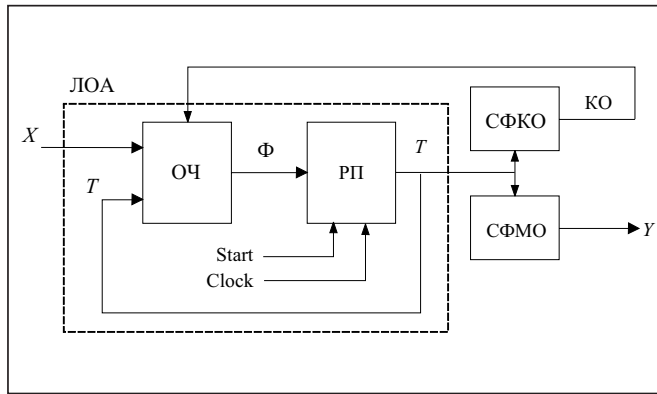


Рис. 1

Код следующего состояния поступает на входы регистра памяти (РП), который совместно с ОЧ образует локальный операционный автомат (ЛОА). На основании кода текущего состояния схема формирования микроопераций (СФМО) формирует некоторые элементы множества микроопераций Y , соответствующие текущему состоянию

янию УА и поступающие в ОА. Также на основании сигналов T схемой формирования кодов операций (СФКО) формируется КО перехода, поступающий в схему ОЧ. По сигналу Start РП инициализируется значением кода начального состояния. Переход в следующее состояние происходит по сигналу Clock.

Схема ОЧ представляет собой совокупность операционных блоков, реализующих множество ОП для заданной ГСА. При этом конечный результат (код следующего состояния автомата) формируется путем мультиплексирования выходов операционных блоков под управлением сигналов КО.

ОЧ может реализовывать как условные, так и безусловные микропрограммные переходы. Если из состояния a^t в состояние a^{t+1} осуществляется безусловный переход, то функция перехода зависит только от кода текущего состояния:

$$K(a^{t+1}) = o^t(K(a^t)). \quad (3)$$

Здесь $K(a^{t+1})$ — код состояния перехода; $K(a^t)$ — код текущего состояния; o^t — функция, реализующая ОП при переходе из состояния a^t .

Если же переход из состояния a^t условный, то в общем случае он может быть осуществлен в одно из нескольких состояний в зависимости от значений сигналов логических условий, проверяемых при данном переходе. При этом функция преобразования имеет вид

$$K(a^{t+1}) = o^t(K(a^t), X^t). \quad (4)$$

Здесь X^t — множество сигналов ЛУ, проверяемых при переходе из текущего состояния; o^t — ОП, реализующая условный переход из текущего состояния.

Таким образом, в структурном описании УА с ОФП фигурируют следующие базовые множества.

1. Множество кодов состояний $K = \{K(a_1), K(a_2), \dots, K(a_M)\}$, где M — количество состояний в автомате для заданной ГСА.

2. Множество операций перехода $O = \{o_1, \dots, o_M\}$, каждому элементу которого соответствует единственный элемент множества K , имеющий такой же индекс. Каждый элемент множества O в общем случае допускает различную реализацию. Например, переход из состояния a_i с кодом 10 в состояние a_j с кодом 20 может быть реализован как арифметической операцией «+10», так и операцией двоичного сдвига на один разряд влево. Выбор реализации каждой ОП может существенно повлиять как на аппаратные затраты, так и на быстрдействие результирующей схемы ЛОА и УА в целом.

Если двум или более операциям перехода соответствуют одинаковые комбинационные схемы, то такие операции перехода будем называть эквивалентными

и образующими класс эквивалентных операций перехода. Разобьем множество на классы эквивалентных операций перехода, причем если для операций перехода o_i найти эквивалентную операцию перехода невозможно, то o_i будет единственным элементом своего класса.

Каждому классу эквивалентных операций перехода поставим в соответствие обобщенную операцию перехода (ООП), КС которой будет такая же, как и у ОП данного класса. Таким образом получим множество ООП $Q = \{q_1, \dots, q_E\}$, где E — мощность данного множества. При $E < M$ имеет место уменьшение количества комбинационных схем, совокупность которых для элементов множества Q образует схему ОЧ на рис. 1.

На основании изложенного выше можно сделать вывод, что оптимизация аппаратных затрат в схеме управляющего автомата с ОФП может быть достигнута уменьшением числа элементов множества Q . Методика уменьшения мощности множества Q неочевидна из заданной ГСА и требует отдельного исследования.

В целом структура управляющего автомата с ОФП определяется следующей системой множеств:

$$S = \langle K, Y, Q, K(Q) \rangle, \quad (5)$$

где Y — множество МО, формируемое автоматом на основании значений кодов состояний, $K(Q)$ — множество КО переходов.

В соответствии с (5) синтез схемы управляющего автомата с ОФП сводится к формированию и схемной реализации элементов перечисленных множеств. Формирование множества Y должно отвечать типу управляющего автомата (Мили или Мура), в то время как для формирования множества Q предварительно должно быть сформировано множество O . Здесь можно предложить два подхода.

1. Если значения кодов состояний в силу каких-либо причин заданные, то для каждого перехода рекомендуется выбирать такую операцию, аппаратная реализация которой обладала бы наилучшими характеристиками согласно выбранному критерию оптимизации (аппаратурные затраты или быстродействие схемы). Назовем данный подход синтезом с предварительным кодированием состояний.

2. Если значения кодов состояний могут выбираться произвольно, то их следует выбирать таким образом, чтобы аппаратные реализации элементов множества ОП обладали оптимальными характеристиками, а мощность множества ООП была бы минимально возможной. Данный подход назовем синтезом с параллельным кодированием состояний (кодирование состояний осуществляется параллельно с формированием множества операций перехода).

В обоих случаях после того, как сформировано множество O , необходимо сформировать множество Q в целях минимизации аппаратных затрат в схеме устройства.

Оптимальной будем считать такую структурную реализацию модели управляющего автомата с ОФП, для которой в соответствии с выбранными критериями оптимизации найдено оптимальное содержимое множества Q . Поиск множества неоднозначен в силу следующих причин.

- Неоднозначным является формирование множества O , так как одному и тому же переходу можно сопоставить различные арифметико-логические операции.

- Неоднозначным является кодирование состояний (в случае параллельного кодирования состояний), которое влияет на формирование множества операций перехода.

Оптимизировать влияние данных факторов на содержимое множества Q можно либо их последовательным перебором в некотором диапазоне, либо

разработкой и применением специальных методов оптимизации, снижающих по-добную неоднозначность.

ЗАКЛЮЧЕНИЕ

Предложенная в настоящей работе структура МПА с операционным формированием переходов обладает следующими преимуществами по сравнению с канонической структурой, использующей реализацию переходов на основе БФ.

Все множество переходов может быть реализовано с помощью ограниченного количества арифметико-логических операций, что может привести к снижению аппаратных затрат при реализации ГСА большой сложности по сравнению с канонической реализацией автомата.

В структуре УА с ОФП локальный ОА может рассматриваться как самостоятельный операционный автомат, и оптимизировать его можно с помощью известных методов оптимизации схем ОА.

Использование арифметико-логических операций позволяет задействовать для реализации операций переходов стандартные операционные узлы современных ПЛИС, являющиеся стандартными библиотечными элементами САПР. Это позволит повысить быстродействие схемы, одновременно сэкономя площадь кристалла ПЛИС по сравнению с реализацией функций переходов в виде булевых выражений. Кроме того, использование библиотечных элементов уменьшает время проектирования и увеличивает тестопригодность схемы [11].

СПИСОК ЛИТЕРАТУРЫ

1. Глушков В.М. Синтез цифровых автоматов. — М.: Физматгиз, 1962. — 476 с.
2. Майоров С.А., Новиков Г.И. Принципы организации цифровых машин. — Ленинград: Машиностроение, 1974. — 432 с.
3. Баркалов А.А., Палагин А.В. Синтез микропрограммных устройств управления. — Киев: ИК НАН Украины, 1997. — 136 с.
4. Поспелов Д.А. Логические методы анализа и синтеза схем. — М.: Сов. радио, 1974. — 386 с.
5. Грушвицкий Р.И., Мурсаев А.Х., Угрюмов Е.П. Проектирование систем на микросхемах программируемой логики. — СПб.: БХВ-Петербург, 2002. — 608 с.
6. Соловьев В.В. Проектирование цифровых систем на основе программируемых логических интегральных схем. — М.: Горячая линия — Телеком, 2001. — 636 с.
7. Закревский А.Д. Логический синтез каскадных схем. — М.: Наука, 1981. — 416 с.
8. Погарцев А.Г. Новые алгоритмы совместной минимизации булевых функций // Автоматика и вычислительная техника. — 1980. — № 1. — С. 34–41.
9. Баркалов А.А. Принципы оптимизации логической схемы микропрограммного автомата Мура // Кибернетика и системный анализ. — 1998. — № 1. — С. 65–72.
10. Баркалов А.А. Структуры многоуровневых схем микропрограммных автоматов на ПЛМ // Там же. — 1994. — № 4. — С. 22–29.
11. Maxfield C. The design warrior's guide to FPGAs. — Amsterdam: Elsevier Sci., 2004. — 642 p.

Поступила 17.04.2009

После доработки 28.09.2009