

является совершенным двоичным кодом в $\{0, 1\}^{\mathbb{N}}$. Отметим, что при построении кода C была применена аксиома выбора.

Лемма 3. В кубе $\{0, 1\}^{\mathbb{N}}$ существуют линейные совершенные двоичные коды.

Посмотрим, как устроены изометрии куба $\{0, 1\}^{\mathbb{N}}$. Так как разные слои этого куба находятся на бесконечном расстоянии Хемминга друг от друга, то изометрия допускает, во-первых, произвольную перестановку (континуального) множества всех слоев. Далее, в каждом слое $\mathcal{L}_\alpha$ допускается (независимо от других слоёв) перестановка координат π_α и перенос на вектор $a_\alpha \in \{0, 1\}_0^{\mathbb{N}}$. Изометрия может не быть аффинным преобразованием всего куба. На этом основании вводим два различных определения.

Определение 3. Два совершенных кода $C_1, C_2 \subset \{0, 1\}^{\mathbb{N}}$ называются *изометричными* (соответственно, *эквивалентными*), если существует изометрия A пространства $\{0, 1\}^{\mathbb{N}}$ (соответственно, изометрия, являющаяся аффинным преобразованием пространства $\{0, 1\}^{\mathbb{N}}$), такая, что $A(C_1) = C_2$.

Лемма 4. Все линейные совершенные коды в $\{0, 1\}^{\mathbb{N}}$ эквивалентны между собой.

Мощность континуума принято обозначать символом $\mathfrak{c}$. Мощность всех подмножеств континуального множества будем обозначать символом $2^{\mathfrak{c}}$. Эту мощность называют также *гиперконтинуумом*.

Пример 1. В пространстве $\{0, 1\}^{\mathbb{N}}$ строится гиперконтинуальное семейство линейных совершенных двоичных кодов $\mathcal{H} = \{H_\gamma\}_{\gamma \in \Gamma}$, такое, что для любых $H_{\gamma_1}, H_{\gamma_2} \in \mathcal{H}$ ($\gamma_1 \neq \gamma_2$) не существует ни одной перестановки $\pi: \mathbb{N} \rightarrow \mathbb{N}$, такой, что $H_{\gamma_2} = \tilde{\pi}(H_{\gamma_1})$.

Теорема 2. Мощность множества всех классов эквивалентности совершенных двоичных кодов в пространстве $\{0, 1\}^{\mathbb{N}}$ равна гиперконтинууму $2^{\mathfrak{c}}$.

ЛИТЕРАТУРА

1. *Потапов В. Н.* Бесконечномерные квазигруппы конечных порядков // Матем. заметки. 2013. Т. 93. Вып. 3. С. 457–460.
2. *Августиневич С. В., Соловьева Ф. И.* Построение совершенных двоичных кодов последовательными сдвигами $\tilde{\alpha}$ -компонент // Проблемы передачи информации. 1997. Т. 33. Вып. 3. С. 15–21.
3. *Романов А. М.* О построении совершенных нелинейных двоичных кодов инверсией символов // Дискрет. анализ и исслед. операций. Сер. 1. 1997. Т. 4. № 1. С. 46–52.
4. *Phelps K. T. and LeVan M. J.* Kernels of nonlinear Hamming codes // Designs, Codes and Cryptogr. 1995. V. 6. No. 3. P. 247–257.
5. *Solov'eva F. I.* Switchings and perfect codes // Numbers, Information and Complexity. Dordrecht: Kluwer Acad. Publ., 2000. P. 311–324.
6. *Васильев Ю. Л.* О негрупповых плотно упакованных кодах // Проблемы кибернетики. М.: Физматгиз, 1962. Вып. 8. С. 75–78.

УДК 512.6

DOI 10.17223/2226308X/8/46

ЭНЕРГОСБЕРЕГАЮЩЕЕ ПРОТИВОГОНОЧНОЕ КОДИРОВАНИЕ СОСТОЯНИЙ АСИНХРОННОГО АВТОМАТА

Ю. В. Поттосин

Рассматривается задача противогоночного кодирования состояний асинхронного автомата, где наряду с минимизацией длины кода состояния минимизируется

интенсивность переключений элементов памяти. Используется подход, предполагающий рассмотрение пар переходов между состояниями и установление для них условий отсутствия опасных состязаний, представляемых в виде троичной матрицы условий. Решение рассматриваемой задачи сводится к решению задачи о взвешенном минимальном покрытии строк матрицы условий множествами строк, для каждого из которых существует вектор, имплицитующий все строки из этого множества.

Ключевые слова: *асинхронный автомат, противогоночное кодирование состояний, энергосберегающее кодирование состояний.*

Моделью поведения логической схемы с памятью является конечный автомат, представляющий собой пятерку (A, B, Q, Ψ, Φ) , где A , B и Q — соответственно множества входных символов, выходных символов и состояний автомата, а Ψ и Φ — функции $\Psi : A \times Q \rightarrow Q$ и $\Phi : A \times Q \rightarrow B$, называемые соответственно функцией переходов и функцией выходов. Для $q_i, q_j \in Q$ и $a \in A$ состояние $q_j = \Psi(a, q_i)$ является тем состоянием, в которое автомат переходит из состояния q_i под воздействием входного символа a . Рассматриваемая задача позволяет игнорировать функцию Φ , поэтому в дальнейшем она не будет здесь упоминаться.

Задача кодирования состояний автомата заключается в присвоении каждому состоянию определённого булева вектора $(z_1, z_2, \dots, z_k)$, называемого кодом состояния, который соответствует набору состояний двоичных элементов памяти (триггеров) в логической схеме, где каждый переход из состояния в состояние представляется переключением одного или нескольких триггеров. Естественно, что это переключение происходит не одновременно. В реальных асинхронных схемах это явление называется *состязаниями*, или *гонками* элементов памяти. Принято называть состязания *неопасными*, если все промежуточные состояния, в которых автомат может оказаться при переходе из одного состояния в другое под воздействием некоторого входного сигнала a , являются неустойчивыми для сигнала a , т. е. при любом порядке переключений элементов памяти автомат из некоторого состояния q_i под воздействием входного сигнала a переходит всегда в состояние $q_j = \Psi(a, q_i)$. Если же при этом автомат может оказаться в некотором устойчивом состоянии q_k , отличном от q_j , то состязания называются *опасными*.

Кодирование состояний, обеспечивающее отсутствие опасных состязаний (гонок), называется *противогоночным*. Естественно, здесь возникает задача минимизации длины кода состояния, приводящая к наименьшему числу элементов памяти в реальной схеме.

Другим критерием оптимизации схемы является величина потребляемой энергии. Это обусловлено, с одной стороны, стремлением увеличить время действия источника энергии в портативных приборах и, с другой стороны, стремлением снизить остроту проблемы отвода тепла при проектировании сверхбольших интегральных схем. Как отмечено в [1], потребляемая мощность схемы, построенной на основе КМОП-технологии, пропорциональна частоте изменения сигналов. Это даёт возможность частично решать данную проблему на уровне логического проектирования. В частности, снижения энергопотребления можно добиваться при кодировании состояний автомата. Кодировать состояния при этом надо таким образом, чтобы при переходе автомата из одного состояния в другое меняли свое состояние как можно меньше элементов памяти. Проблеме энергосберегающего кодирования состояний синхронного автомата посвящено довольно много работ, одной из которых является, например, работа [2], где процесс кодирования состояний автомата представляется как размещение состояний

в булевом пространстве внутренних переменных. Проблема энергопотребления логических схем рассматривается также в [3], где решается задача определения режима максимального потребления энергии в схеме, реализующей конечный автомат. В данной работе рассматривается возможность учёта энергосбережения при противогоночном кодировании состояний асинхронного автомата. Автору не известны публикации, посвящённые данной проблеме.

Условие отсутствия опасных состязаний для пары переходов $q_i \rightarrow q_j$, $q_k \rightarrow q_l$ ($q_j \neq q_l$) при одном и том же входном сигнале a можно выразить троичным вектором [4], в котором компоненты соответствуют состояниям автомата, компоненты i и j имеют одно значение, 0 или 1, а компоненты k и l — противоположное ему значение. Остальным компонентам приписывается значение «—». В схеме, реализующей заданный автомат, это условие выполняется триггером, который в процессе одного перехода рассматриваемой пары хранит состояние 0, а в процессе другого — 1.

На множестве векторов, представляющих условия отсутствия опасных состязаний, имеется отношение импликации: троичный вектор $\mathbf{a}$ имплицирует троичный вектор $\mathbf{b}$, если $\mathbf{b}$ получается из $\mathbf{a}$ заменой некоторых нулей или единиц значением «—» и, возможно, инвертированием полученного результата. Например, вектор $(10 - - 101)$ имплицирует векторы $(10 - - - 01)$ и $(01 - - - 1-)$. Смысл этого отношения в том, что условие, представленное вектором $\mathbf{b}$, автоматически выполняется при соблюдении условия, представленного вектором $\mathbf{a}$.

Все условия отсутствия опасных состязаний в виде описанных векторов составляют троичную матрицу, в которой отсутствуют имплицируемые строки. Эта матрица называется *матрицей условий* [4]. Говорят, что троичная матрица R *имплицирует* троичную матрицу S , если для каждой строки матрицы S в матрице R найдётся имплицирующая её строка. Задача противогоночного кодирования с минимизацией длины кода состояния сводится к нахождению матрицы с минимальным числом строк, имплицирующей матрицу условий и называемой *кратчайшей имплицирующей формой* матрицы условий. Столбцы этой матрицы будут представлять искомые коды состояний, а получаемая в результате её транспонирования матрица называется *матрицей кодирования*. Строкам матрицы кодирования соответствуют состояния автомата, а столбцам — внутренние переменные, и строки этой матрицы представляют коды соответствующих состояний.

Кратчайшая имплицирующая форма матрицы условий находится следующим образом. Множество строк матрицы условий называется *совместимым*, если существует вектор, имплицирующий каждую строку этого множества. Совместимое множество называется *максимальным*, если оно не является собственным подмножеством другого совместимого множества. Надо найти кратчайшее покрытие множества строк матрицы условий максимальными совместимыми множествами. Каждому совместимому множеству соответствует вектор, имплицирующий все строки, принадлежащие этому множеству. Векторы, соответствующие элементам полученного покрытия, в качестве строк составят кратчайшую имплицирующую форму заданной матрицы условий.

При применении описанного подхода к решению задачи противогоночного кодирования состояний автомата для снижения интенсивности переключений элементов памяти можно использовать следующие соображения.

Каждому столбцу матрицы кодирования можно поставить в соответствие множество переходов. Это множество для i -го столбца составляют те переходы, которыми связаны состояния автомата, в кодах которых переменная z_i имеет различные значения, т.е. при таких переходах i -й триггер в реальной схеме, реализующей автомат,

меняет своё состояние. Следовательно, для снижения интенсивности переключений элементов памяти надо выбрать такой вариант противогоночного кодирования состояний, который соответствует наименьшему множеству переходов между состояниями.

Если удаётся вычислить вероятности переходов, то столбцу матрицы кодирования состояний ставится в соответствие вероятность события, которое заключается в том, что происходит некоторый переход из множества переходов, связанных с данным столбцом. Поскольку переходы между состояниями автомата являются несовместимыми событиями, эта вероятность равна сумме вероятностей отдельных переходов из данного множества. Для подсчёта вероятностей переходов между состояниями в [2] используется метод Чэпмена — Колмогорова, где данные вероятности получаются в результате решения системы линейных уравнений с этими вероятностями в качестве неизвестных. Однако этот метод можно применять только тогда, когда автомат является полностью определённым, а его граф поведения является сильносвязным ориентированным графом. В противном случае столбцу матрицы кодирования состояний автомата приписывается мощность связанного с ним множества переходов.

Таким образом, каждому совместимому множеству строк матрицы условий и соответственно вектору, имплицитующему все строки из этого множества, приписывается вес в виде числа переходов или суммы вероятностей переходов, связанных с этим вектором. Искомое решение получается в виде покрытия множества строк матрицы условий максимальными совместимыми множествами, обладающего минимальным весом. Весом покрытия является сумма весов принадлежащих ему элементов.

Кодирование состояний автомата можно представить как размещение состояний в пространстве внутренних переменных $z_1, z_2, \dots, z_k$ [2], т. е. по вершинам булева гиперкуба, представляющего это пространство. В [2] введён критерий качества такого размещения с точки зрения интенсивности переключений элементов памяти. Этот критерий выражается формулой $D = \sum w_{ij}(d_{ij} - 1)$, где d_{ij} — расстояние по Хэммингу между кодами состояний q_i и q_j ; w_{ij} — число переходов или вероятность перехода между состояниями q_i и q_j , и суммирование берётся по всем парам состояний, соответствующим парам вершин в гиперкубе. Очевидно, чем меньше значение D , тем лучше результат размещения, и $D = 0$, если всем парам состояний, связанным переходами, соответствуют рёбра гиперкуба. Тогда при любом переходе из состояния в состояние переключается ровно один элемент памяти. Сравнение по критерию D результатов решения примеров показало целесообразность использования предлагаемого метода.

ЛИТЕРАТУРА

1. *Мурога С.* Системное проектирование сверхбольших интегральных схем. В 2-х кн. Кн. 1. М.: Мир, 1985.
2. *Закревский А. Д.* Алгоритмы энергосберегающего кодирования состояний автомата // Информатика. 2011. № 1(29). С. 68–78.
3. *Закревский А. Д.* Нахождение режима максимального энергопотребления логической схемы // Прикладная дискретная математика. 2012. № 2. С. 100–104.
4. *Закревский А. Д.* Логический синтез каскадных схем. М.: Наука, 1981.