ЛИТЕРАТУРА

1. ECC Turbo Product Code Technology. — URL: http://www.eccincorp.com/ /technology.htm

2. Berrou C., Glavieux A., Thitimajshima P. Near Shannon limit error-correcting coding and decoding: Turbo-codes // in ICC'93,- Switzerland, May 93. –P. 1064–1070.

3. Hagenauer P., Offer E., Papke L. Iterative decoding of binary block and convolutional codes // IEEE Trans. Inform. Theory. Vol. 42. No. 2, Mar. 1996.– P. 429–445.

4. Benedetto S., Montorsi G. Unveiling Turbo Codes: Some Results on Parallel Concatenated Coding Schemes // IEEE Trans. Inform. Theory. Vol. 42. No. 2, March 1996. – P. 409–428.

5. Benedetto S., Montorsi G. Design of parallel concatenated convolutional codes // IEEE Trans. Commun. Vol. 44. No. 5, May 1996.– P. 591–600.

6. Галлагер Р.Г. Коды с малой плотностью проверок на четность // Теория кодирования.–М.:Мир, 1964. С. 139–165.

7. Hunt A., Crozier S., Falconer D. Hyper-Codes: High-Performance Low-Complexity Error-Correcting Codes // 19-th Biennial Symposium on Communications, Kingston, Ontario, Canada, May 31–June 3, 1998. – P. 263–267.

УДК 004.052

А.И. Галабурда, аспирант

РАЗРАБОТКА АЛГОРИТМА ВЫБОРА ПОРОЖДАЮЩЕЙ МАТРИЦЫ КОДА ХЕММИНГА С ЦЕЛЬЮ МИНИМИЗАЦИИ АППАРАТНЫХ ЗАТРАТ ДЛЯ РЕАЛИЗАЦИИ КОДЕРА

In this article new method of selection of generating Hamming code matrix is described. Some improvements of algorithm are discussed. Given sample implementation code for key aspects of algorithm. As the result of implementation the matrix for Hamming code (12, 8) been chosen.

Технологические методы повышения надежности аппаратного обеспечения во многих случаях оказываются неэффективными, так как либо требуют больших капитальных затрат и связаны с продолжительными предварительными исследованиями, либо вообще ограничены существующим уровнем научно- технического прогресса. Одним из путей решения задачи повышения отказоустойчивости в настоящее время является использование специальных процедур, основанных на применении помехоустойчивых (корректирующих) кодов. Широкое распространение в этой области получил код Хемминга.

Любой код характеризуется следующими параметрами: основание кода q – число различных элементарных символов, выбранных для построения кода; значность n – число элементов, образующих кодовую комбинацию; мощность M – число различных кодовых комбинаций. Максимальное число кодовых комбинаций (M_{max}) при заданных n и q равно q^n. Если число кодовых слов меньше максимальной мощности, то код называется избыточным. Например, код 000, 001, 101, 110 – избыточный, так как M = 4, а M_{max} = 8. Именно избыточное кодирование позволяет не только обнаруживать ошибки, но и исправлять их.

Линейный (n, k)-код представляет собой подпространство размерностью k линейного n-мерного пространства над полем, определяемым основанием кода. Код можно задавать перечислением базисных векторов подпространства [1]. Совокупность базисных векторов

будем далее записывать в виде матрицы G размерностью k x n с единичной подматрицей в первых k строках и k столбцах.

$$G = [\, I_k \mid -A^T],$$

где A – некоторая фиксированная ((n–k) x k) – матрица из 0 и 1, а I – единичная матрица размерности k.

Матрицу G называют порождающей матрицей линейного корректирующего кода (далее – матрица) в приведено-ступенчатой форме. Однако чаще имеют дело с матрицей вида $H = [A \mid I_{n-k}]$, называемой проверочной. Матрица задает систему булевых функций, где элементарной операцией является суммирование по модулю два. Далее, исходя из удобства использования, будут рассматриваться проверочные матрицы, что не противоречит цели исследования, т.к. проверочная матрица может быть легко преобразована в порождающую. Известно, что функция алгебры логики может быть подвергнута минимизации, процедуре нахождения наиболее простого ее представления в виде суперпозиции функций, составляющих функционально полную систему, при одновременной минимизации ее технической реализации по некоторым критериям. К настоящему времени получили распространение следующие методы минимизации:

4. Расчетный метод (метод непосредственных преобразований).
5. Расчетно-табличный метод (метод Квайна–МакКласки).
6. Метод Петрика (развитие метода Квайна–МакКласки).
7. Табличный метод (карты Карно).
8. Метод гиперкубов.
9. Метод факторизации.
10. Метод функциональной декомпозиции и др.

Первый метод применяется при числе переменных $m <= 3$ и основан на использовании операций склеивания, поглощения и развертывания [2].

Второй и третий методы используются при $m \leq 16$ в профессиональных разработках и ориентированы на использование систем автоматизированного проектирования с применением ЭВМ [3]. Четвертый метод является самым распространенным инженерным методом минимизации функции для $m \leq 6$.

Шестой метод не имеет каких-либо существенных достижений при решении общих задач, более простых, чем метод перебора всех формул функции даже для $m = 3$. Практически он используется для уменьшения сложности минимальных дизъюнктивных и конъюнктивных нормальных форм, полученных с использованием первого или четвертого методов. Он основан на использовании скобочных форм и форм с групповыми инверсиями.

Седьмой метод основан на представлении функции, зависящей от m переменных, в виде суперпозиций функций, зависящих от меньшего числа переменных, для которых можно применить вышеперечисленные методы [2].

Однако вышеозначенные методы малоприменимы при $m > 8$, кроме того, матрица Хемминга задает систему функций, что усложняет задачу, т.к. необходимо минимизировать не одну функцию, а систему. Также желательно, чтобы результат минимизации системы, задаваемой матрицей, использовал суммирование по модулю два, что позволит использовать унифицированную элементную базу, чего вышеописанные методы гарантировать не могут.

Ввиду вышеозначенных причин возникает необходимость разработки алгоритма для решения частной задачи минимизации, направленной на сокращение затрат на реализацию

кодера кода Хемминга (подчеркнем, что основным элементом кодера является сумматор по модулю два).

Для подбора оптимальной матрицы кода Хемминга используем следующий алгоритм (рис. 1).

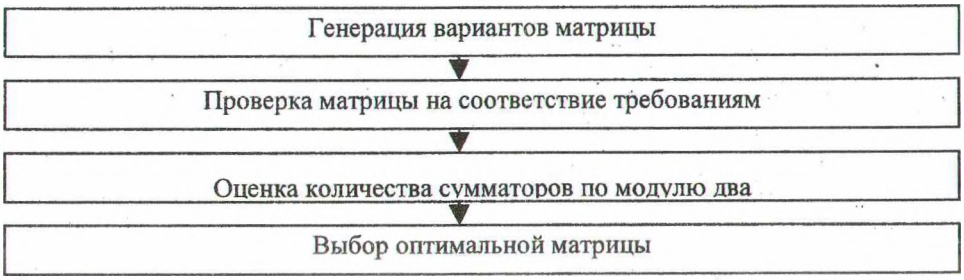

Рис. 1

Суть метода заключается в реализации следующих операций: 1) генерация всех возможных вариантов перестановок из заданного набора; 2) полученные варианты последовательно проходят проверку на соответствие заданным условиям; 3) определение веса матриц и выбор из них матрицы минимального веса. Рассмотрим эти шаги подробнее.

На первом шаге необходимо сформировать алфавит, элементы которого будут использованы при создании матриц. Т.к. мы ставим задачу уменьшения количества сумматоров кодера кода Хемминга за счет повторного использования элементов, то целесообразно использовать элементы максимального веса. Однако можно использовать и комбинированный подход, когда наряду с элементами с максимальным весом используются и элементы минимального веса. После формирования алфавита можно приступать к формированию матриц. Для этой задачи подходят рекурсивные алгоритмы. Блок-схема алгоритма формирования матриц приведена на рис. 2.

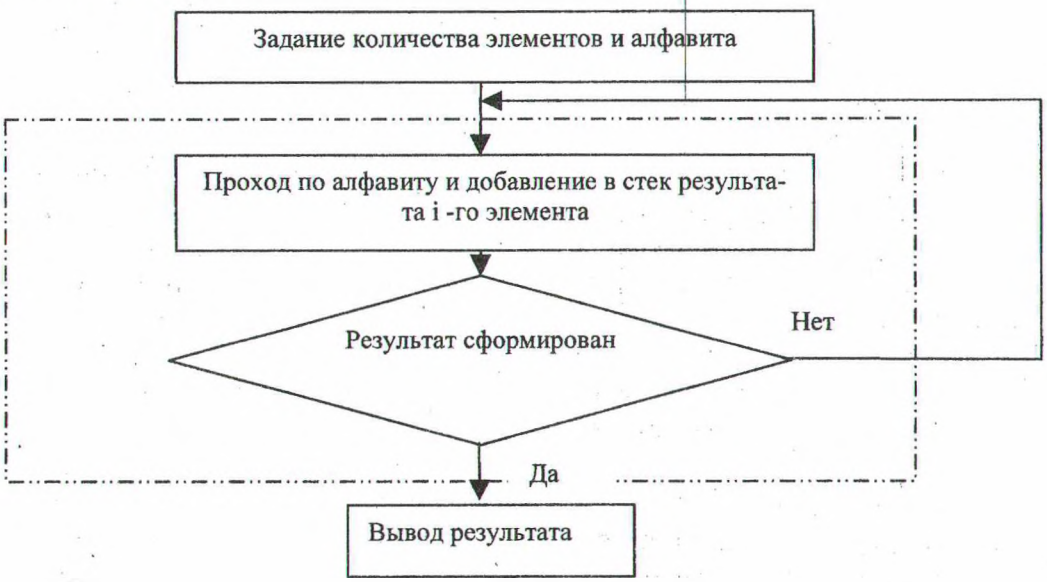

Рис. 2

Пунктирной линией на рисунке обозначено элементарное преобразование, повторяемое многократно. Исходными данными алгоритма являются: алфавит символов (множест-

во беззнаковых чисел) и количество итераций, а результатом является набор структур данных, описывающих матрицу.

Оценим зависимость потребности в вычислительных ресурсах от размера матрицы кода. Количество вычислительных операций прямо пропорционально количеству возможных выборок. Обозначим количество элементов исходного множества, из элементов которого происходит формирование выборок, как n, а объем формируемых выборок r. Тогда число различных выборок из исходного множества, элементы которых могут повторяться [4],

$$A_n^r = n^r. \quad (1)$$

Т.к. в нашем случае n = r, то формула трансформируется в

$$A_n^n = n^n. \quad (2)$$

Однако в данном случае необходимо проверять матрицы на соответствие требованиям, предъявляемым к матрицам кода Хемминга.

Т.к. известно, что столбцы матрицы не могут повторяться, то формула примет вид

$$A_n = n! \quad (3)$$

Алгоритм может быть модифицирован с целью уменьшения количества вычислений, например исключением символа из алфавита, который вставлен в стек результата на текущей итерации, и уже модифицированный таким образом алфавит использовать для формирования результата на последующих итерациях.

После формирования вариантов матрицы необходимо определить число сумматоров по модулю два для реализации кодера. За счет повторного использования оборудования число сумматоров может быть уменьшено. Рассмотрим, например, матрицу для кода (n = 7, k = 4):

$$H = \begin{vmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{vmatrix}$$

Для получения первого контрольного символа необходимо просуммировать по модулю 2 первый и второй биты исходной информации, а затем полученный результат сложить с четвертым битом, т.е. $x_1 + x_2 + x_4 = x_1$. В то же время для получения второго контрольного символа необходимо просуммировать первый, второй и третий биты, т.е. $x_1 + x_2 + x_3 = x_2$. Возможная реализация кодера кода Хемминга представлена на рис. 3а. Прямоугольниками обозначены двухвходовые сумматоры по модулю 2. Как видно из соотношений, для вычисления первого и второго (проверочных) символов можно использовать общий блок суммирования первого и второго битов исходной информации. Сумматоры по модулю 2, используемые для сложения первого и второго битов, выделены на рис.3а серым цветом. Модифицированная схема приведена на рис. 3b, на ней исключен избыточный сумматор.

Вычисление необходимого числа сумматоров для кодера проводится поэтапно. На первом этапе сумматоры разбиваем на пары. А на втором – последовательно перебираются полученные пары. Для определения того, использовался ли данный сумматор где-либо в матрице, будем использовать вектор. Если по адресу, который представляется значением пары, записан ноль, то такого сумматора нет, иначе в ячейке записано число сумматоров.

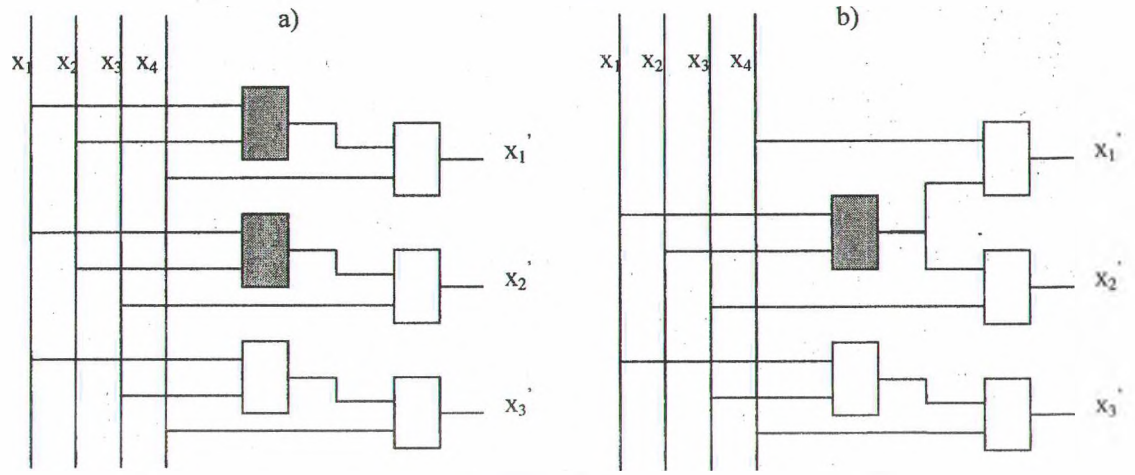

Рис. 3

В начале вычислений вектор инициализирован нулями. Таким образом, на втором этапе при переборе пар мы увеличиваем на единицу значение по адресу, равное значению пары. Параллельно мы подсчитываем максимальное количество сумматоров, необходимое для расчета, оно на единицу меньше веса строки матрицы Хемминга(без единичной матрицы). Для определения числа сумматоров мы вычитаем из числа, представляющего максимальное количество сумматоров, значения ячеек, за вычетом единицы. Это число и будет равно количеству сумматоров. Оптимальная матрица выбирается по минимуму числа сумматоров.

ЗАКЛЮЧЕНИЕ

Разработан алгоритм выбора порождающей матрицы кода Хемминга, позволяющий уменьшить количество элементов для реализации кодера. Необходимость разработки алгоритма обусловлена невозможностью применения известных методов минимизации булевых функций для решения данной задачи ввиду ее сложности.

С помощью программной реализации описанного выше алгоритма было проведено, для примера, исследование по выбору оптимальной порождающей матрицы линейного корректирующего кода Хемминга (n = 12, k = 8).

В результате работы программы получена следующая матрица:

$$H = \begin{array}{|cccccccc|cccc|} \hline 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Если реализовать кодер на основе этой матрицы без исключения повторяющихся элементов, то потребуется 18 сумматоров по модулю 2 (число сумматоров на кодирование строки равно числу единиц за вычетом двойки). Однако, исключив повторяющиеся вычисления (например, x_1+x_2 в первой и второй строках, x_3+x_4 в первой, второй и четвертой строках матрицы и т.д.), для реализации кодера потребуется всего 13 сумматоров по модулю 2. Таким образом, количество элементов может быть сокращено на треть.

ЛИТЕРАТУРА

1. Урбанович П.П., Алексеев В.Ф., Верниковский Е.А. Избыточность в полупроводниковых интегральных микросхемах памяти. – Мн.: Наука и техника, 1995. –262 с.

2. http://www5.newmail.ru/Lections/min_fal/vorob04.htm

3. Колдуэлл С. Логический синтез релейных устройств / Пер. с англ. –М.: Изд-во иностранной литературы, 1962. –740 с.

4. Цыпкин А.Г., Пинский А.И. Справочник по методам решения задач по математике. – М.: Наука, 1989. –567с.

СОДЕРЖАНИЕ