Математические структуры и моделирование
2004, вып. 13, с. 132–134

УДК 681.3

АВТОМАТЫ КАУФФМАНА КАК ЦИКЛИЧЕСКИЕ ГЕНЕРАТОРЫ ЗАРАНЕЕ ЗАДАННЫХ КОМАНД

О.Т. Данилова, Р.Т. Файзуллин

In this article an application of NK Kauffman like automata for cyclic and exact generation for preliminary given some code words is presented.

Рассмотрим сеть состоящую из N булевых логических элементов («И», «ИЛИ», сложный комбинационный автомат), где каждый элемент i имеет K_i входов и только один выход, рис.1. Пусть на выходах этих элементов x_i в некоторый момент времени с помощью внешней шины и генерируется сигнал, в виде набора нулей и единиц x_i^0.

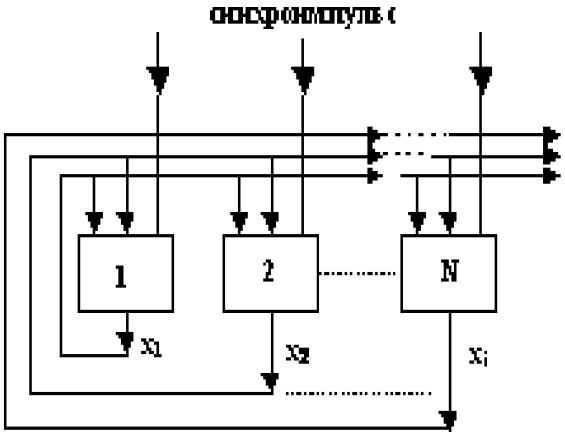

Рис. 1.

Предположим, что автомат функционирует в дискретном времени так, что выходные сигналы подаются на входы элементов (возможно инвертированные).

Частным случаем такого автомата являются NK-автоматы Кауффмана, где $K_i = K$ [1]. Очевидно, что совокупность выходных сигналов $X(t) = (x_1^t, ...x_N^t)$ сходится к некоторому аттрактору - предельному циклу. Длина аттракторов M и типичная длина аттрактора L, это важная характеристика автоматов, зависящая только от степени связности K. Будем далее называть автоматами

E-mail: rtf@univer.omsk.su
Омский государственный университет

Кауффмана и общий случай схемы, когда число входов в логические элементы переменно. Отметим, что автоматы Кауффмана использовались для моделирования генетической регуляторной системы живых клеток, как «программы» [2]. Было выяснено, что при $K = 2$ поведение подобных автоматов устойчиво к малым изменениям входных данных и это свойство прозволило моделировать процессы мутации живых организмов. Однако увеличение числа связей приводит к каскадному изменению последовательных состояний автомата и такое поведение автоматов нельзя было соотнести с моделью мутаций.

Возможно, что более общая ситуация может найти применение в другой предметной области, например, если рассматривать автоматы Кауффмана, как генераторы, распределенной по времени «памяти». В этом случае некоррелированность последовательных значений векторов выхода будет уже положительным фактором, т,к. генерируемые по тактам работы автомата «команды» векторы состояний X, обычно слабо коррелированы.

Основная задача в реализации такого подхода организации памяти заключается в том, чтобы по заданному набору команд подобрать такой список связей между логическими элементами, чтобы на каждом такте работа автомата генерировался заранее заданный код команды и, самое главное, чтобы набор команд генерировался циклически.

Учитывая, что команды не коррелированы, и их не так уж и много, а длина аттрактора (цикла) при увеличении числа связей K сильно растет, то казалось бы одновременное выполнение этих требований невозможно. Более того, очевидное построение систем булевых уравнений, связывающих значения последовательных значений векторов выхода на различных тактах, приводит к экспоненциальному росту числа уравнений, и делает совершенно невозможным построение на такой основе эффективного алгоритма проектирования необходимых связей.

Можно предложить более простой подход, при котором значения K_i минимальны и равны в большей своей части единице.

Рассмотрим общий случай, когда $N = 2^I$ и генерируется $I + 1$ команда, причем $I + 1$ команда совпадает с первой:

$$x_1 = x_1^1 x_1^2 .. x_1^I x_1^1 \tag{1}$$

$$...$$

$$x_N = x_N^1 x_N^2 .. x_N^I x_N^1$$

Если все строки различны, то для любой строки

$$x_J = x_J^1 ... x_J^I x_J^1$$

найдется строка

$$x_M = x_M^1 ..... x_M^I x_M^1$$

где будут выполняться равенства:

$$x_J^2 = x_M^1, x_J^3 = x_M^2 ... \tag{2.1}$$

$$x_J^2 = \overline{x}_M^1, x_J^3 = \overline{x}_M^2 ... \tag{2.2}$$

Строку M будем называть прародителем строки J.

Если в списке встречаются одинаковые строки то очевидно, что для некоторых из строк в (1) не обязательно должен существовать прародитель вида (2.1) или (2.2), но можно дополнить список новыми служебными строками, которые будут прародителями и в свою очередь будут иметь прародителей среди расширенного списка.

Данный пример очевидным образом показывает, что заданных I команд, всегда можно подобрать связи между 2^{I+1} элементами «И», так чтобы заданные команды генерировались циклически.

Заметим, что связи устанавливаются неоднозначно и можно выбирать наиболее экономичный вариант с конструкторской точки зрения.

Рассмотрим пример, когда $N = 16$, $I = 4$

$x_1 = 10100$ $x_2 = 11101$ $x_3 = 01010$ $x_4 = 01111$

$x_5 = 00110$ $x_6 = 11101$ $x_7 = 11011$ $x_8 = 10111$

$x_9 = 11101$ $x_{10} = 01010$ $x_{11} = 10111$ $x_{12} = 10111$

$x_{13} = 01110$ $x_{14} = 11001$ $x_{15} = 10111$ $x_{16} = 11101$

Генерацию можно реализовать в виде:

$x_1 = x_2x_4x_{14}$ $x_2 = x_7$ $x_3 = x_1$ $x_4 = x_{11}$

$x_5 = x_{16}x_{13}$ $x_6 = x_7$ $x_7 = x_{11}$ $x_8 = x_4$

$x_9 = x_7$ $x_{10} = x_1$ $x_{11} = x_4$ $x_{12} = x_4$

$x_{13} = x_{17}$ $x_{14} = x_7x_8$ $x_{15} = x_4$ $x_{16} = x_7$

где введен новый добавочный элемент $x_{17} = 11100, x_{17} = x_{14}$.

Заметим, что введения добавочного логического элемента казалось бы можно избежать, но за это придется платить нарушением единообразия линейки элементов. Так, в рассмотренном примере можно записать: $x_{13} = \overline{x}_4 + x_{13}x_{16}$. Это означает, что в качестве логического элемента с выходом x_{13} мы выбираем сложный комбинационный элемент, состоящий из комбинации элементов «ИЛИ» и «И», причем всегда необходим еще один дополнительный элемент «И», на котором происходит конъюнкция с синхроимпульсом. Таким образом экономия на числе элементов не происходит.

Предложенная модель может быть использована, например, как ПЗУ с параллельной выборкой микрокоманд [3, с.230-231]. В этом случае можно воспользоваться тем, что длина слова, т.е. (число элементов «И») достаточно произвольно. В качестве вспомогательных элементов в этом случае могут служить строки разрядов, генерируемые соседними микрокомандами.

Литература

1. Kauffman S.A., Smith R.G. *Adaptive automata based onDarwinian selection* // Phisica D. 1986. V.22, N.1-3. P.68-82.
2. Kauffman S.A., *Origins of order: self-organization and selection in evolution.* N.Y.: Oxford Univ.Press, 1993.
3. Майоров С.А., Новиков Г.И. *Принципы организации цифровых машин.* Ленинград.: Машиностроение. Ленинградское отделение, 1974.