

# О ПРОБЛЕМАХ СИНТЕЗА ИЗОМОРФНЫХ АВТОМАТОВ

В.В. Стрижов, Вычислительный центр РАН

## **Аннотация**

При синтезе автомата требуется определить такую его структуру, которая была бы оптимальна в пространстве трех параметров: времени выполнения алгоритма, ресурсов (требуемого количества логических вентилях) и точности вычислений. Рассматриваются изоморфные конечные автоматы и их применение в реконфигурируемых компьютерах.

## **Ключевые слова**

*Изоморфные автоматы, реконфигурируемые компьютеры, синтез конечных автоматов*

## **Введение**

Предлагаемая работа заключается в исследовании изоморфных описаний конечных автоматов, которые выполняют заданный класс алгоритмов. Ее целью является описание системы, которая выбирает оптимальный алгоритм из существующего набора алгоритмов для решения поставленной задачи, строит структурную схему автомата, выполняющего алгоритм, и, при необходимости, реструктурирует автомат во время выполнения алгоритма в зависимости от ресурсов и промежуточных результатов работы алгоритма. Такая система применяется в реконфигурируемых компьютерах – интегральных схемах, которые работают в составе компьютеров общего назначения.

Реконфигурируемые вычисления (reconfigurable computing) – способ выполнения вычислений с помощью аппаратного обеспечения, которое может быть конфигурировано для оптимального выполнения алгоритма и может быть реконфигурировано в течение работы алгоритма. Для этой цели используются программируемые логические интегральные схемы (ПЛИС) с достаточно большим набором вентилях. Выпускаются интегральные схемы емкостью до 200.000 системных вентилях и выше.

Задачи, которые решаются при создании реконфигурируемых систем, состоят из определения взаимозаменяемости (изоморфности) автоматов,

выполняющих алгоритмы на аппаратном уровне ИС реконфигурируемого компьютера; синтеза набора изоморфных конечных автоматов заданного класса с учетом ограничений, предъявляемых аппаратными ресурсами; поддержки существующей системы реконфигурирования ИС [1]; поддержки существующих трансляторов [2], выполняющих синтез конечных автоматов применительно к заданной ИС.

Приведем пример управления выполнением алгоритмов на аппаратном уровне. Имеются два базовых способа вычислений с помощью логических схем, построенных на булевой логике: комбинационный и рекурсивный. Операция умножения (или алгоритм, выполняющий эту операцию), выполняется первым способом за один такт автоматного времени. Для выполнения умножения таким способом используется комбинационная схема, состоящая из логических вентилях и содержащая множество всех результатов операции и получающая в качестве входного слова любую из возможных пар операндов. При получении на входе двух сомножителей, произведение появляется на выходе памяти за один такт времени.

Рекурсивный способ умножения использует сумматор и сдвиговый регистр. Для получения результата операции умножения автомат должен просуммировать и сдвинуть результат столько раз, сколько символов в алфавите выходного слова. На выполнение операции умножения требуется значительное количество тактов времени при небольшом количестве требуемых вентилях. При выполнении операции умножения первым способом требуется большое количество ресурсов – логических вентилях, но результат получается за минимальное время. Второй способ требует значительно меньше ресурсов, но результат получается за большее время.

В паре рекурсивные–комбинационные вычисления возможны промежуточные варианты построения автоматов: например, операцию умножения также можно выполнить с помощью нескольких сумматоров [3]. Множимое параллельно подключается к четырем суммирующим входам всех

арифметических блоков, а множитель подается поразрядно на управляющие входы этих блоков.

Рекурсивному способу вычисления в общем случае можно поставить в соответствие выполнение алгоритмов с помощью машины Тьюринга, а комбинационному способу вычисления можно поставить в соответствие выполнение алгоритмов с помощью просмотровой таблицы.

Для выполнения некоторой заданной операции предполагается синтезировать множество автоматов, каждый из которых принимает слова, соответствующие элементам прообраза операции и выдает слова, соответствующие элементам образа операции. Это множество автоматов задается технологией, в которой выполнена программируемая логическая интегральная схема. Например, синтезированная для некоторой операции автоматная схема может отличаться большой сложностью (количеством требуемых вентилях), но минимальным временем выполнения. Рекурсивная схема, например машина Тьюринга, потребует значительно меньшее количество вентилях, но время выполнения будет значительно выше. Требуется найти оптимальную конфигурацию с учетом того, что интегральная схема-носитель будет перепрограммироваться для выполнения операции непосредственно перед выполнением.

Таким образом, мы пришли к оптимизационной задаче с двумя критериями оптимальности: требуемые ресурсы и время выполнения результата. Эти два критерия принимают значения на соответствующих множествах и их отношение (некоторое множество пар на декартовом произведении элементов этих двух критериев) отражает характеристики возможных автоматов. Характеристики оптимальных автоматов принимают значения на Парето-оптимальном фронте этого множества. Выбор автомата с характеристиками на Парето-оптимальном фронте позволяет оптимизировать выполнение алгоритмов на аппаратном уровне.

## Структурный синтез автоматов

Абстрактный автомат  $S=(Z, W, A, l, d)$  является моделью цифрового устройства которая характеризуется алфавитом входа  $Z=\{z_1, \dots, z_m\}$ , алфавитом выхода  $W=\{w_1, \dots, w_n\}$ , алфавитом состояний  $A=\{a_1, \dots, a_p\}$ , функцией выхода  $l: A \times Z \rightarrow W$  и функцией перехода  $d: A \times Z \rightarrow A$ .

Два его состояния  $a_i$  и  $a_j$  называются эквивалентными, и обозначаются как  $a_i \sim a_j$ , если  $l(a_i, z_k) = l(a_j, z_k)$  для любого слова, то есть реакция автомата в этих состояниях на любые слова одинакова. Определение: Два автомата  $S=(Z, W, A, l, d)$  и  $S'=(Z', W', A', l', d')$  называются эквивалентными, то есть  $S \sim S'$ , если для любого  $a_i \in A$  существует  $a_j \in A'$  такое, что  $a_i \sim a_j$ , и если для любого  $a_j \in A'$  существует  $a_i \in A$  такое, что  $a_j \sim a_i$ .

Два автомата называется изоморфными если они, вне зависимости от их внутренней структуры, выполняют одно и то же отображение:  $\phi: Z \rightarrow W$ , допускающее операцию выравнивания длин слов.

Как указано в [4], основной задачей теории структурного синтеза автоматов является нахождение общих приемов построения структурных схем автоматов, принадлежащих к заранее заданному типу автоматов.

Целью структурного синтеза является построение схемы, реализующей автомат из логических элементов заданного типа. В структурном автомате учитывается структура входных и выходных сигналов автомата, а также его внутреннее устройство на уровне структурных схем. На этапе структурного синтеза предварительно выбираются элементарные автоматы, из которых затем путем их композиции строится структурная схема автомата Мили [5] или Мура [6].

Поведение автомата Мили описывается с помощью уравнений 
$$\begin{cases} a_{t+1} = d(z_t, a_t), \\ w_t = l(z_t, a_t). \end{cases}$$

Здесь функция переходов определяет следующее внутреннее состояние автомата (состояние перехода)  $a_{t+1}$ , а функция выходов – формируемый выходной набор  $w_t$ . Характерным для автомата Мили является то, что выходной набор  $w_t$  зависит как от входного набора  $z_t$ , так и от внутреннего состояния  $a_t$ . В

автомате Мура функция выходов зависит только от состояния  $a_t$  и не зависит от входного набора, поэтому поведение автомата Мура описывается уравнениями: 
$$\begin{cases} a_{t+1} = d(z_t, a_t), \\ w_t = l(a_t). \end{cases}$$

Если решение задачи структурного синтеза существует, говорят, что заданная система автоматов структурно полна. Теоретическим обоснованием канонического метода структурного синтеза автоматов является теорема о структурной полноте: всякая система элементарных автоматов, которая содержит автомат Мура с нетривиальной памятью, обладающий полной системой переходов и полной системой выходов, и какую-либо функционально полную систему логических элементов, является структурно полной.

Существует общий конструктивный прием – канонический метод структурного синтеза, позволяющий в рассматриваемом случае свести задачу структурного синтеза произвольных автоматов к задаче синтеза комбинационных схем.

Результатом канонического метода структурного синтеза является система логических уравнений, выражающая зависимость выходных сигналов автомата и сигналов, подаваемых на входы запоминающих элементов, от сигналов, приходящих на вход автомата в целом, и сигналов, снимаемых с выхода элементов памяти.

Для правильной работы схем, очевидно, нельзя разрешать, чтобы сигналы на входе запоминающих элементов непосредственно участвовали в образовании выходных сигналов, которые по цепям обратной связи подавались бы в тот же самый момент времени на эти входы. В связи с этим запоминающими элементами должны быть не автоматы Мили, а автоматы Мура.

Таким образом, структурно полная система элементарных автоматов должна содержать хотя бы один автомат Мура. В то же время для синтеза любых автоматов с минимальным числом элементов памяти необходимо в качестве таких элементов выбирать автоматы Мура, имеющие полную систему переходов и полную систему выходов – так называемые полные автоматы.

При разработке различных цифровых устройств и систем очень важно применять эффективные методы синтеза конечных автоматов. Как показано в [7], реализованные в известных пакетах автоматизированного проектирования, методы синтеза основаны на традиционных подходах, которые не позволяют эффективно использовать архитектурные возможности новой элементной базы. До сих пор не исследованы многие модели конечных автоматов, допускающих реализацию на современных ПЛИС, не определена сложность реализации на ПЛИС различных моделей конечных автоматов.

В работе [8] рассмотрены формальные методы синтеза конечных автоматов, которые позволяют наиболее полно и эффективно использовать архитектурные возможности ПЛИС. Отличительной чертой всех методов синтеза является простота, что делает возможным их применение даже при ручном проектировании. Одним из способов синхронизации работы изоморфных автоматов является использование регистров на входах и выходах автомата для буферизации входных сигналов и согласование выходных сигналов с сигналами синхронизации.

Для этого вводятся различные классы конечных автоматов, на основании которых определяются структурные модели конечных автоматов, реализуемых на ПЛИС. При построении конечных автоматов функции переходов и выходов реализуются с помощью комбинационных схем, а память автомата реализуется в виде регистра, где в каждый момент автоматного времени  $t$  хранится код внутреннего состояния  $a_t$ . Совмещать в одной структуре несколько моделей различных классов конечных автоматов можно только в случае совпадения временных параметров выходных сигналов каждой модели. В противном случае, может сложиться ситуация, когда значения выходных сигналов, формируемых в одном и том же состоянии конечного автомата, на выходе схемы будут появляться в различных тактах автоматного времени.

## **Заключение**

Каждому из множества автоматов, соответствующему оператору, отображающего фиксированный элемент из конечного множества-прообраза в

множество-образ, сопоставляется тройка параметров (время выполнения операции, сложность автомата, точность выполнения операции). Далее решается задача оптимизации автомата, причем аргументами функции ценности являются взвешенные элементы указанной тройки. Задача оптимизации решается на некотором заданном множестве автоматов.

Рассматриваемый метод синтеза конечных автоматов предназначен для их реализации на ПЛИС. В рамках одной структурной модели, названной изоморфной, объединяется несколько различных моделей конечных автоматов. Это позволяет наиболее полно использовать архитектурные возможности ПЛИС и положительные качества каждой модели для реализации конечных автоматов.

### **Список литературы**

- [1] Two Flows for Partial Reconfiguration: Module Based or Difference Based, XAPP290 (v1.1), Xilinx, Inc., San Jose, CA, November 25, 2003.
- [2] X-BLOX Design Tool User Guide, Xilinx, Inc., San Jose, CA, 1992.
- [3] Титце, У., Шенк, К., Полупроводниковая схемотехника, М.: Мир, 1982, с. 340–341.
- [4] Глушков, В. М., Синтез цифровых автоматов, М.: Физматгиз, 1962, с. 134.
- [5] Mealy, G. H., A method for synthesizing sequential circuits, Bell System Techn. J., v. 34, 1955, pp. 1045–1079.
- [6] Moore, E. F., Gedanken-experiments on sequential machines, In C. Shannon and J. McCarthy editors. Automata Studies Princeton University Press, 1956, pp. 129–153.
- [7] Соловьев, В., Структурные модели конечных автоматов при их реализации на ПЛИС, Chip News, N. 9, 2002, с. 4–14.
- [8] Соловьев, В., Климович, А., Синтез на ПЛИС совмещенных модулей конечных автоматов, Chip News, N. 3, 2003, с. 10–12.