

# ПРОЕКТИРОВАНИЕ КОНЕЧНЫХ АВТОМАТОВ

## *ВВЕДЕНИЕ*

Конструкции конечных автоматов широко используются в последовательных управляющих логических системах, которые формируют ядро многих цифровых систем. Конечные автоматы применяются во множестве устройств широкого диапазона действия и сложности, например в: устройствах управления низкого уровня "микропроцессор-VLSI-периферийные интерфейсы", арбитражах шин и устройствах генерации времени в традиционных микропроцессорах, в секционных микропроцессорах, системах кодирования и декодирования данных, в протоколах передачи данных – это лишь несколько примеров систем, где требуется применение конечных автоматов.

Как правило, система логического управления – это последнее, что должно разрабатываться в цикле проектирования устройства, так как конструкция управляющего автомата зависит от особенностей построения устройства. Применение программируемой логики – оптимальное решение для проектирования систем логического управления, потому что это позволяет производить модификации в проекте, не нарушая физической структуры компьютерного устройства. Применение элементов программируемой логики позволяет быстро исправлять и модифицировать проекты, практически не влияя при этом на время их выхода на рынок.

В большинстве проектов систем логического управления удобно и целесообразно использовать для их реализации микросхемы ПЛМ типа PAL. Сейчас есть возможность реализовывать высокоскоростные и сложные проекты устройств управления на микросхемах программируемой логики. При проектировании систем логического управления возможно применение широкого диапазона различных функциональных

решений. В этой статье мы рассмотрим функции, которые выполняют системы логического управления, реализацию этих систем на различных устройствах и выбор конструкции системы.

## ***ЧТО ТАКОЕ КОНЕЧНЫЙ АВТОМАТ?***

Конечный автомат – это цифровое устройство, работа которого заключается в прохождении через последовательность предопределенных состояний определенным образом. Состояния – это набор регистрируемых в разных частях схемы сигналов. Традиционная структура системы управления состоит из PAL-элементов, реализующих комбинаторную логику, выходных регистров и регистров состояний. В такой структуре состояния последовательно сохраняются в регистре состояний и выходном регистре.

Общая структура системы управления изображена на рис. 1. Кроме входов и выходов в структуре системы управления выделяют два основных структурных элемента – схему комбинаторной логики и схему памяти (регистр). Подобная структура является базовой и простейшей из структур систем управления. Элементы памяти в системе используются для хранения состояний автомата. Блок комбинаторной логики может рассматриваться в виде композиции двух функционально различных частей: блока формирования следующего состояния и блока формирования выходных сигналов (рис. 2). Блок формирования следующего состояния отвечает за формирование сигналов, необходимых для перехода автомата в следующее состояние, а блок формирования выходных сигналов отвечает за формирование управляющих слов на выходе управляющего автомата. Хотя эти два блока и выполняют совершенно разные функции, они обычно логически объединяются в единый блок комбинаторной логики, как на рис. 1.

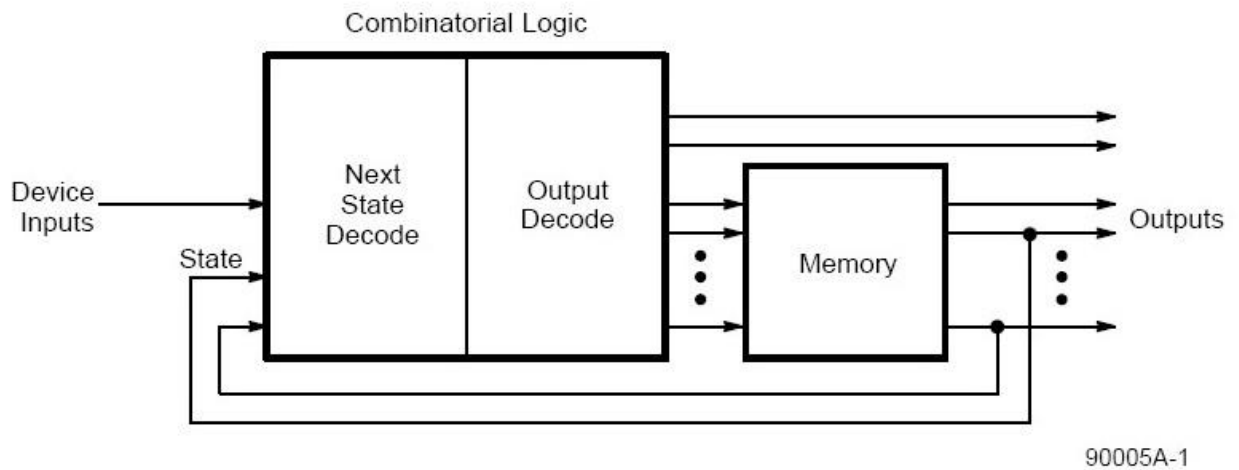


Рисунок 1. Структурная схема тривиального конечного автомата

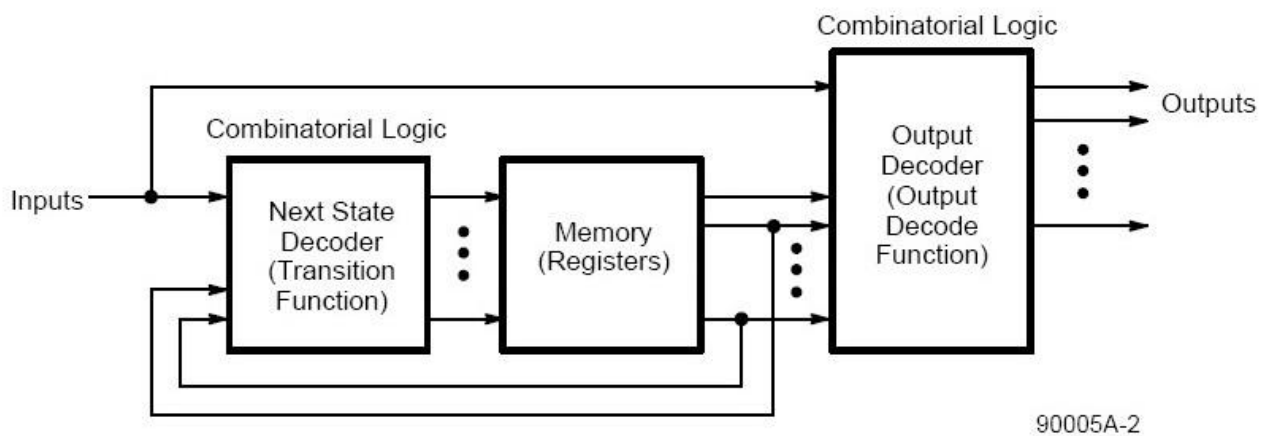


Рисунок 2. Структура конечного автомата со схемой формирования следующего состояния автомата и схемой формирования выходных слов

Управляющие автоматы выполняют две следующие основные операции:

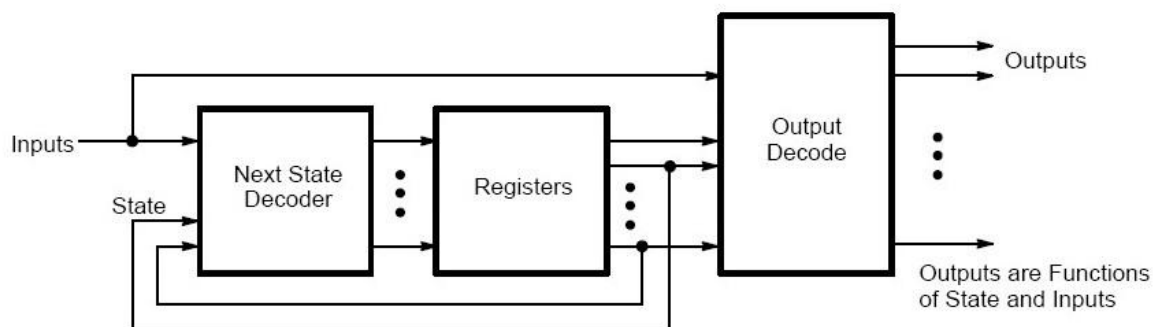
1. Переход в следующее состояние, который осуществляется с помощью схемы формирования функций возбуждения памяти (Next State Decoder (Transition Function)), которая вырабатывает свои выходные сигналы с учетом входных условных сигналов и предыдущего состояния автомата.
2. Формирование выходных управляющих сигналов, которые формируются на выходе схемы формирования выходных сигналов (Output Decoder (Output Decode Function)) с учетом текущего состояния, в котором находится автомат, и входных условных сигналов.

Использование входных сигналов для формирования следующего состояния автомата называют переходом. Кроме переходов в следующее состояние, в управляющем автомате возможна организация повторения последовательностей состояний (циклы) и организация подпрограмм. Переходы из одного состояния в другое называются "управляющей последовательностью", а логические функции, которые требуется решить для перехода в следующее состояние, называются "функциями перехода" (Transition Function) (рис. 2).

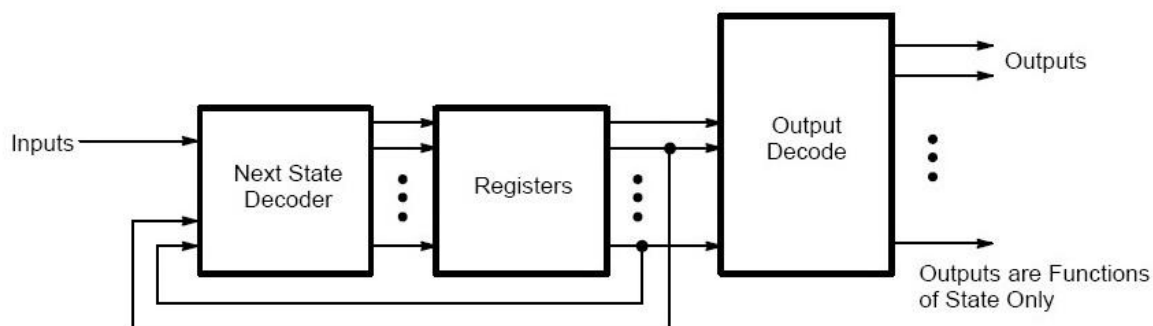
Способ использования входных условных сигналов для формирования выходных управляющих воздействий определяет тип управляющего автомата. Есть два широко известных типа управляющих конечных автоматов: автомат Мили и автомат Мура (рис. 3). В автомате Мура выходные сигналы зависят только от текущего состояния, в котором находится автомат. В более распространенных автоматах Мили выходные сигналы зависят как от текущего состояния автомата, так и от входных сигналов условий. Логические функции, которые определяют значения выходных сигналов, называются "выходными функциями". Для обоих типов конечных автоматов управляющие последовательности зависят от предыдущего и следующего состояния автомата и от входных сигналов.

Большинство управляющих автоматов, широко применяемых на практике, действуют по синхронному принципу по кругу, они совершают переход в следующее состояние по активному состоянию синхросигнала. Генератор синхросигналов связан со всеми состояниями и с триггерами-защелками регистров памяти, эти триггеры запоминают входные сигналы по переднему фронту синхросигнала. Возможно построение и асинхронных управляющих автоматов, работа которых основана на задержке распространения сигнала в логических элементах схемы комбинаторной логики, эта задержка используется для организации памяти автомата. Такие автоматы сильно подвержены сбоям, их трудно проектировать и поэтому они не нашли широкого применения на практике. В

этой статье мы сосредоточим внимание только на последовательных управляющих автоматах.



a. Mealy State Machines



b. Moore State Machines

90005A-3

Рисунок 3. Два стандартных типа управляющих автоматов

## ***ПРИМЕНЕНИЕ УПРАВЛЯЮЩИХ АВТОМАТОВ***

Управляющие автоматы применяются в основном для контроля функционирования цифровых систем. Осуществление выборки команд в таких системах и применение для этого управляющих автоматов описано ниже.

Применение управляющих автоматов в цифровых сигнальных микроконтроллерах (DSP) позволяет достичь максимальную скорость и широкие функциональные возможности функционирования этих микроконтроллеров без применения сложных микропроцессоров. В простых алгоритмах, например в алгоритме быстрого преобразования Фурье (FFT),

управляющие автоматы могут управлять умножением и добавлением в микроконтроллер наборов векторов. Для выполнения сложных операций в DSP лучше применять программируемые микроконтроллеры. Но с другой стороны, программируемые DSP уступают по скорости функционирования специализированным системам.

Рассмотрим функционирование видео контроллера. В этих системах требуется формировать адреса для выборки и вычисления переменных различной длины в разнообразных последовательностях. Эти последовательности вычислений контролируются и осуществляются с помощью переходов управляющего конечного автомата. Применение автомата позволяет заметно сократить аппаратные затраты в системе. Счет может быть установлен или возобновлен управляющим автоматом, за счет чего ресурсы микропроцессора освобождаются для выполнения других операций.

В сфере периферийного контроля применение управляющих автоматов может быть очень эффективным. Рассмотрим случай RLL кодирования. Задачи кодирования и декодирования могут быть возложены на управляющий конечный автомат, который будет читать входной поток данных и формировать выходной поток данных.

Применение управляющих автоматов также возможно в таких областях, как промышленный контроль и робототехника. Решение таких задач, как расположение механического манипулятора и вычисление тригонометрических функций, обычно не требует применения мощных микропроцессоров со стеками и указателями. Скорее требуется устройство, способное хранить ограниченное число состояний и осуществляющее простые условные переходы.

Кодирование и декодирование данных – это те проблемы, которые требуется решать при шифровании и расшифровании информации в сфере технического обеспечения средств массовой информации, в этой сфере требуется применять системы, надежно защищенные от взлома. Для реализации этих задач идеально подходят управляющие конечные

автоматы, так как их элементы памяти программируются во время разработки автомата и потом невозможно будет получить доступ к этим элементам извне.

## ***ВЫПОЛНЯЕМЫЕ ФУНКЦИИ***

Все функции, которые выполняет конечный управляющий автомат, могут быть выделены в следующие основные категории:

- арбитраж;
- контроль событий;
- многократное тестирование условий;
- выбор времени задержек;
- контроль генерации управляющих сигналов.

Ниже мы рассмотрим пример проекта и покажем на нем, как можно использовать перечисленные функции при проектировании управляющего автомата.

## ***ТЕОРИЯ УПРАВЛЯЮЩИХ АВТОМАТОВ***

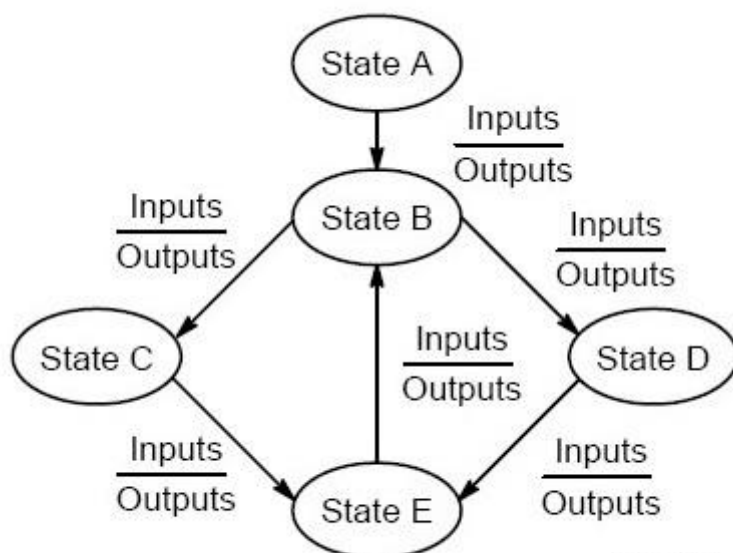
Давайте бегло рассмотрим теоретические основы последовательных логических схем, конечных управляющих автоматов (КУА) или просто управляющих автоматов.

Те части цифровых систем, выходные сигналы которых зависят от прошлых и текущих входных сигналов, могут быть представлены и промоделированы в виде конечных управляющих автоматов. "История" состояний машины влияет на ее текущее внутреннее состояние. Когда на какой-либо из входов КУА поступает активное значение сигнала, это заставляет осуществить переход автомата в новое состояние и генерировать новые выходные сигналы. В свою очередь это новое состояние автомата также зависит от значений входных сигналов и от текущего со-

стояния управляющего автомата. Структура КУА изображена на рис. 2. Внутреннее состояние управляющего автомата хранится в блоке памяти (регистре). Как было сказано ранее, в автомате присутствуют две комбинационные логические схемы: схема формирования функций перехода и схема формирования выходных сигналов управляющего автомата.

### **ПОСТРОЕНИЕ ДИАГРАММ СОСТОЯНИЙ**

Поведение конечных управляющих автоматов может быть представлено в графическом виде, как показано на рис. 4. Представление функционирования автомата в таком виде называют диаграммой состояний или диаграммой переходов. Каждый кружок представляет собой отдельное состояние, а стрелки – это переходы между состояниями. Условия, которые вызывают тот или иной переход, показываются возле каждой стрелки перехода.



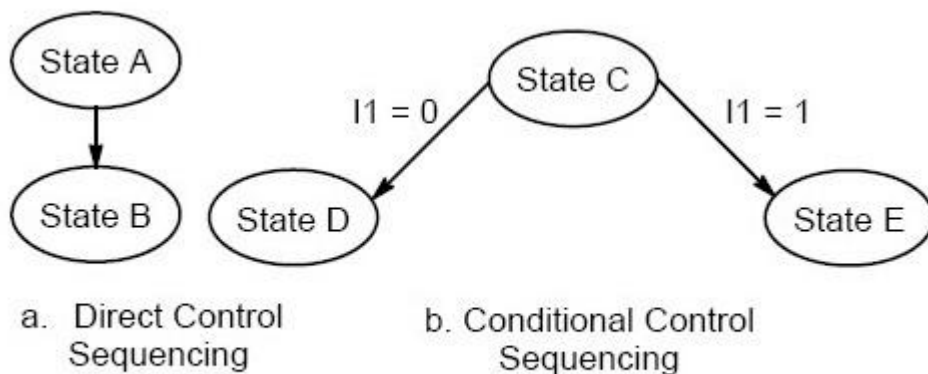
90005A-4

Рисунок 4. Графическое представление функционирования управляющего автомата

Упорядоченные переходы между состояниями автомата представлены на рис. 5. Прямой переход изображается как переход из состояния



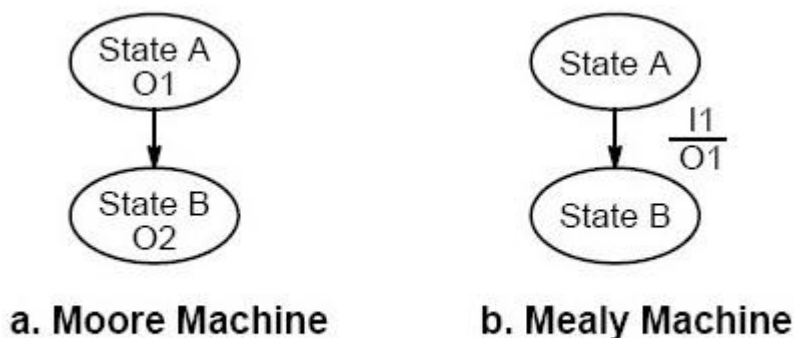
A в состояние B. Условный переход из состояния C осуществляется в зависимости от входного сигнала I1 либо в состояние D, либо в состояние E.



90005A-5

Рисунок 5. Виды переходов между состояниями

Переход между состояниями в автомате Мура показан на рис. 6. В автомате Мили переход между состояниями осуществляется так, как показано на рис. 4. Более подробное описание процесса перехода между состояниями в автоматах Мили и Мура будет дано позже.



90005A-6

Рисунок 6. Организация переходов в автоматах Мура и Мили

В приведенных примерах может возникнуть путаница в процессе определения входных и выходных сигналов, инициирующих переход между состояниями и обозначенных около соответствующих стрелок переходов. Входные и выходные сигналы пишутся через дробь для того,

чтобы их не перепутать (рис. 4 и 6). Иногда также используются более детальные надписи, которые описывают входные и выходные сигналы переходов.

Диаграммы состояний можно сделать более компактными, ставя на переходах не входные переменные, которые вызывают этот переход, как на рис. 4, а булевы выражения функций, которые вызывают этот переход. Например, в диаграмме на рис. 7 входные сигналы, инициирующие тот или иной переход, обозначены как Start, X1 и X2. В переходе между состояниями 1 и 2 сигналы X1 и X2 не используются (то есть они не влияют на переход), и потому они не обозначены возле стрелки соответствующего перехода на диаграмме. Этот прием сокращает запись и делает диаграмму более читаемой.

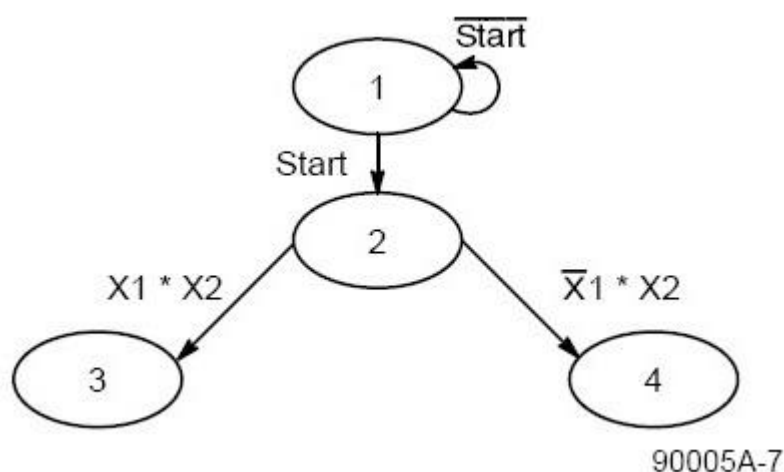


Рисунок 7. Диаграмма состояний с обозначением функций переходов в виде булевых функций

При применении этого метода обозначения переходов могут возникнуть неопределенные ситуации. Переходы между состояниями диаграммы на рис. 8 показывают то, что может случиться. Есть три входные комбинации  $(I0, I1, I2, I3) = \{1011\}, \{1101\}$  и  $\{1111\}$ , которые делают обе функции  $(I0 * I2 + I3)$  и  $(I0 * I1 + I0 * I2)$  истинными. Переход же одновременно в два состояния невозможен, эта ситуация – ошибка спецификации диаграммы. В этом случае функции переходов должны быть из-

менены или надо гарантировать, что три вышеописанные комбинации никогда не возникнут. В этом примере проблему можно решить, изменив  $(I_0 * I_1 + I_0 * I_2)$  на  $(I_0 * I_1 + I_0 * I_2) * I_3$ .

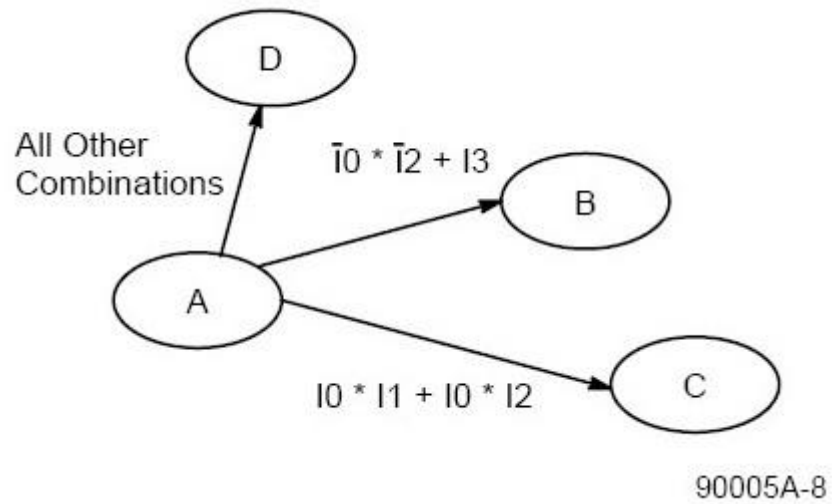


Рисунок 8. Диаграмма состояний с примером конфликтных переходов