

Языки описания схем

(mk.cs.msu.ru → Лекционные курсы → Языки описания схем)

Блок 14

Понятия операционного и управляющего автоматов

лектор:

Подымов Владислав Васильевич

e-mail:

valdus@yandex.ru

Осень 2018

Что такое автомат

Прежде всего “автомат” — это математический объект, точное определение которого имеет много вариаций с общими чертами и принципами работы:

- ▶ он содержит множество **состояний** (*как правило, конечное*)
- ▶ он (*как правило*) работает в дискретном времени:
1, 2, 3, ...
- ▶ в каждый момент времени он находится в одном из состояний (**текущем состоянии**)
- ▶ в каждый момент времени он может принимать входные сигналы от окружения и отправлять выходные сигналы в окружение (*а может не принимать и не отправлять*)

Что такое автомат

Прежде всего “автомат” — это математический объект, точное определение которого имеет много вариаций с общими чертами и принципами работы:

- ▶ состояние, в котором он будет находиться в следующий момент времени, зависит от текущего состояния и от принятого входного сигнала (*если он есть*; иногда однозначно, а иногда и неоднозначно)
- ▶ он может иметь (*а может и не иметь*) начальное состояние, то есть однозначно или неоднозначно определённое состояние, в котором он находится в момент времени 1
- ▶ ...

Что такое автомат

Основная вариация понятия “автомата”, используемая в схемотехнике, — это **конечный детерминированный автомат-преобразователь Мура**, который далее называется просто “автомат”

Автомат над входным алфавитом A и выходным алфавитом B — это система (Q, T, O) , где

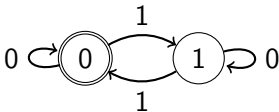
- ▶ Q — конечное множество **состояний**
- ▶ $T : Q \times A \rightarrow Q$ — **функция переходов**
- ▶ $O : Q \rightarrow B$ — функция выхода

Инициальный автомат — это система (Q, q_0, T, O) , где (Q, T, O) — автомат, и q_0 — **начальное состояние** ($q_0 \in Q$)

Что такое автомат

Пример: автоматная реализация счётчика чётности

$$\begin{aligned}A = B &= \{0, 1\} & Q &= \{even, odd\} & q_0 &= even \\T(even, 0) &= T(odd, 1) & &= even \\T(odd, 0) &= T(even, 1) & &= odd \\O(even) &= 0 & O(odd) &= 1\end{aligned}$$



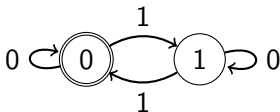
Цифровые схемы на одном из уровней абстракции очень похожи на такие автоматы

Автоматы и схемы

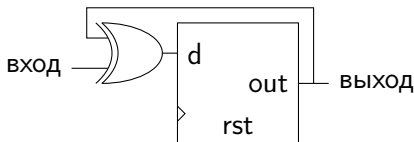
- ▶ Последовательная часть схемы — это *набор регистров* с общим тактовым сигналом
- ▶ Каждый регистр **находится в некотором состоянии**, то есть хранит в себе некоторое значение
- ▶ По фронту тактового сигнала все регистры перезаписывают хранящиеся значения — **изменяют свои состояния**
- ▶ Фронты тактового сигнала наступают **дискретно**, то есть могут служить в роли **дискретного времени**
- ▶ Набор состояний регистров может расцениваться как **состояние автомата**
- ▶ Оставшиеся части последовательной схемы — это входы, выходы и комбинационная подсхема
- ▶ Набор логических значений на входах во время переднего фронта тактового сигнала может расцениваться как **входной символ**, а непрерывно передаваемых на выход — как **выходной символ**
- ▶ Комбинационная подсхема может расцениваться как **описание функций переходов и выхода**

Автоматы и схемы

Возвращаемся к примеру



С учётом обозначенного соответствия этот автомат может расцениваться как такая **последовательная схема**:



Передний фронт сигнала сброса приводит эту схему (автомат) в начальное состояние

Чтение входного значения и изменение текущего состояния происходит по переднему фронту тактового сигнала

Автоматы и схемы

Итог рассуждений: цифровая схема — это совокупность параллельно работающих автоматов

Иногда такое представление схемы оказывается полезным, а иногда — не очень

Например, для реализации счётчика чётности на языке Verilog не нужно знать о существовании автоматов

Однако при разработке “реальных” схем привлечения автоматной терминологии, как правило, не избежать

Операционный и управляющий автоматы

При разработке большой схемы (*как и при написании большой программы*), чтобы не запутаться, повысить скорость разработки и снизить число ошибок, следует писать схему не “наобум”, а придерживаясь принципов структурированной разработки схем

Один из таких принципов уже упоминался — это принцип модульности: чтобы разработать большую и сложную схему, следует отдельно разработать её маленькие простые части, а затем правильно их соединить

Операционный и управляющий автоматы

При разработке сложных схем активно используется и другое правило — **принцип разделения данных и управления**

Входы и выходы схемных блоков, а также соединённые с ними провода, как правило, можно разделить на

- ▶ **информационные**: по ним пересылаются данные, обработка которых (сохранение, преобразование и выдача) происходит в схеме
- ▶ **управляющие**: по ним пересылаются сигналы, руководящие действиями по обработке данных (“сохрани”, “прибавь”, “сбрось”, ...)

Операционный автомат — это подсхема, состоящая из информационных проводов и блоков обработки данных, соединённых с ними

Управляющий автомат — это подсхема, организующая появление нужных логических значений в управляющих проводах

Операционный и управляющий автоматы

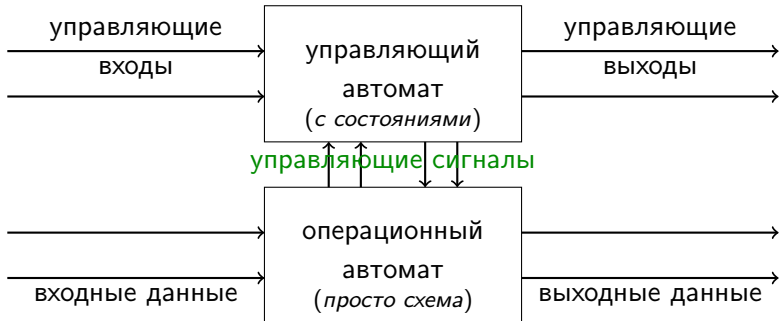
Операционный автомат, как правило, при разработке выглядит совсем не как “автомат” в математическом смысле, хотя в конечном итоге и является автоматом

Например, при разработке операционного автомата процессора не понадобится рассуждать в терминах *состояний* и *переходов*

При разработке **управляющего автомата** чаще всего наиболее естественной и наглядной оказывается именно автоматная терминология: состояния, функция переходов, функция выхода

Заключение

Типичное устройство сложной схемы, реализующей алгоритмически нетривиальный функционал, выглядит так:



Disclaimer:

Это упрощённая картинка. Естественно, на практике всё более сложно и запутанно. Но всё же при разработке нетривиальных схемных модулей такая картинка возникает достаточно часто, чтобы признать её "типичной"