

Языки описания схем

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

Блок 17

Как спроектировать управляющий автомат

лектор:

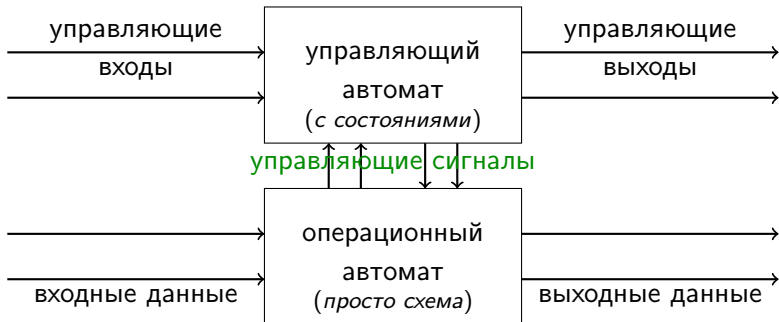
Подымов Владислав Васильевич

e-mail:

valdus@yandex.ru

Осень 2018

Вступление



Чтобы спроектировать операционный автомат, достаточно представить себе цикл жизни **данных** и реализовать представленное в виде регистров и комбинационных под схем, оставив **управление** данными нереализованным

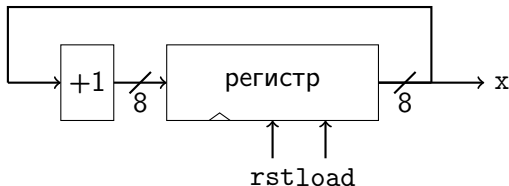
Управляющий автомат не содержит **данных**, поэтому не может быть разработан аналогичным образом

Нередко управляющий автомат разрабатывается как реализация **автомата** в математическом смысле (**блок 14**)

Сквозной пример

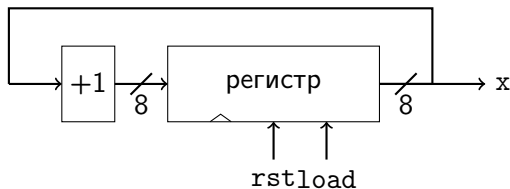
Представим себе, что кто-то захотел спроектировать счётчик, способный сбрасывать число x ширины 8 и увеличивать его на единицу при *особых обстоятельствах*

Независимо от обстоятельств основная часть операционного автомата будет выглядеть примерно так:



Особые обстоятельства в этом операционном автомате представлены сигналами `rst` и `load`, и выставление этих сигналов оставлено управляющему автомату

Простой счётчик



Уточним понятие “особых обстоятельств”

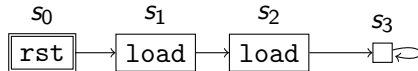
Попробуем реализовать счётчик S_1 , после асинхронного сброса (reset) управляющего автомата последовательно изменяющий число x так:

- ▶ сброс (записывается 0)
- ▶ прибавляется 1
- ▶ прибавляется 1
- ▶ до следующего асинхронного сброса управляющего автомата число не изменяется

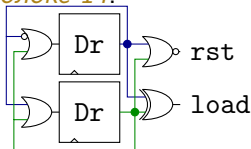
Простой счётчик

- ▶ сброс (записывается 0)
- ▶ прибавляется 1
- ▶ прибавляется 1
- ▶ до следующего асинхронного сброса управляющего автомата число не изменяется

Диаграмма Мура управляющего автомата счётчика S_1 :

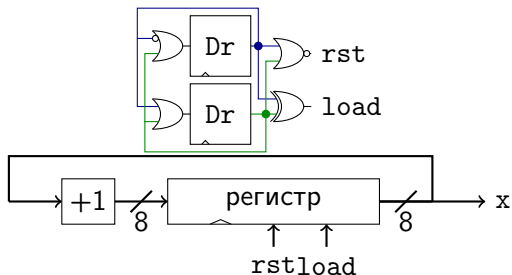


Последовательная схема, соответствующая этому автомату согласно описанию в *блоке 14*:

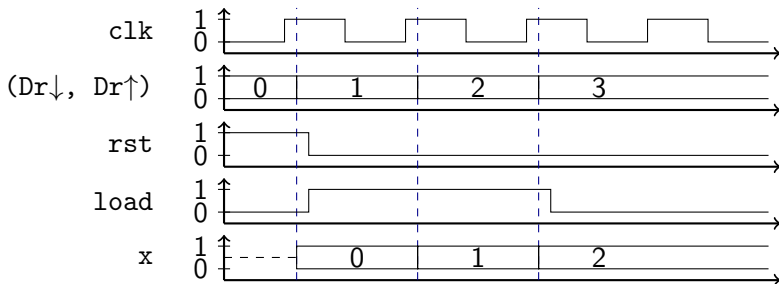


(Dr - D-триггер с асинхронным сбросом)

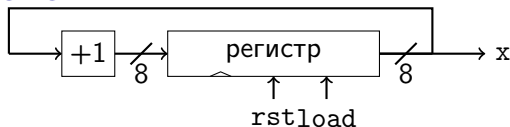
Простой счётчик



После глобального сброса схема будет работать так
(с учётом небольших задержек распространения сигналов):



Сложный счётчик



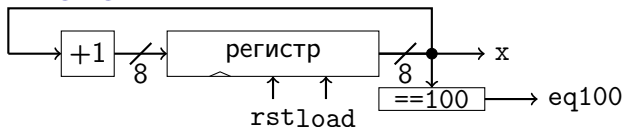
Модифицируем счётчик, заставив его прибавлять 1 не два раза, а **сто** раз, при этом не требуя увеличения выводимого числа *каждый* такт: можно увеличивать число раз в несколько тактов

Схему можно модифицировать двумя способами:

1. Не изменять операционный автомат, и заменить управляющий автомат на аналогичный, содержащий 102 состояния
2. Спроектировать **небольшой** управляющий автомат, дополненный входным сигналом, непрерывно показывающим, выводится ли сейчас число 100, и подходящим образом дополнить операционный автомат

Рассмотрим подробно второй способ

Сложный счётчик

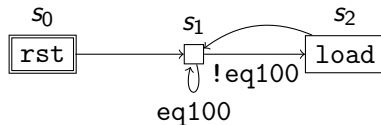


Число x относится к **данным**, и пересылать это значение напрямую в управляющий автомат *нехорошо* (управляющий автомат не работает с **данными**)

При этом в управляющий автомат не нужно пересылать точное значение x , достаточно переслать однобитовое **управляющее** значение $eq100$: $eq100 = 1 \Leftrightarrow x = 100$

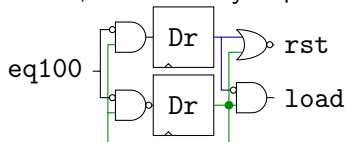
Сложный счётчик

Диаграмма Мура управляющего автомата счётчика S_2 , отсчитывающего до 100:

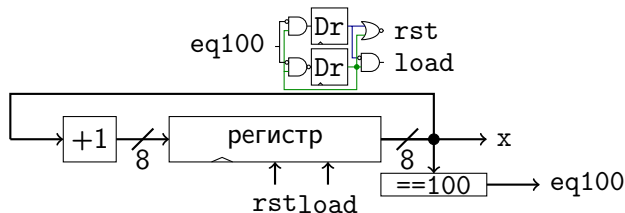


Булево выражение, помечающее переход, обозначает то, при каких значениях входных сигналов выбирается этот переход

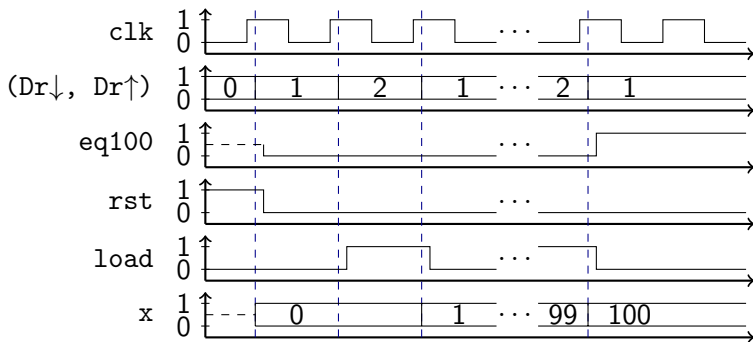
Последовательная схема, соответствующая этому автомату:



Сложный счётчик



После глобального сброса схемы будут работать так
(с учётом небольших задержек распространения сигналов):



Как разработать управляющий автомат

Чтобы разработать управляющий автомат в цифровой схеме, достаточно придумать конечное число режимов работы схемы, таких что

- ▶ в каждом режиме схема находится ровно один такт
- ▶ в каждом режиме однозначно определены значения всех входных управляющих сигналов операционного автомата
- ▶ состояние на следующем такте однозначно определяется текущим состоянием и значениями входных сигналов управляющего автомата в момент перехода

Например, счётчик S_2 имел три режима работы:

1. только что схема сбросилась
2. проверка сигнала “пора остановиться”
3. увеличение выводимого числа

Как разработать управляющий автомат

Схема может быть устроена намного более хитро, чем в рассмотренных примерах, но при этом принципы разработки управляющего автомата остаются *приблизительно* такими же

Примеры усложнений:

- ▶ в схеме содержится несколько тактовых сигналов разной частоты и фазы, и эти сигналы генерируются и используются не по назначению
 - ▶ достаточно внимательно посмотреть, как соотносятся фронты этих сигналов и как синхронизируются последовательные части схемы, и реализовать подходящие режимы работы
- ▶ операционный автомат содержит асинхронные управляющие входы
 - ▶ частота работы управляющего автомата может быть выше, чем операционного, если требуется за один “медленный” такт успеть выставить и убрать асинхронный сигнал

Как разработать управляющий автомат

Схема может быть устроена намного более хитро, чем в рассмотренных примерах, но при этом принципы разработки управляющего автомата остаются *приблизительно* такими же

Примеры усложнений:

- ▶ управляющие входы операционного автомата имеют очень нетривиальный смысл
 - ▶ вместо одного управляющего автомата можно использовать несколько параллельно работающих автоматов,
 - ▶ каждый из которых руководит своим управляющим сигналом, или
 - ▶ некоторые из которых руководят “частями” управляющего сигнала, подаваемыми на вход другому управляющему автомату
- ▶ ...