



МИРЭА – Российский технологический университет  
Кафедра вычислительной техники

---

## Теория автоматов

# Практическая работа №2:

Проектирование синхронных цифровых автоматов

Автоматы распознавания языков. Асинхронный язык

Старший преподаватель:  
Боронников Антон Сергеевич  
[antboronnikov@mail.ru](mailto:antboronnikov@mail.ru)

Москва 2023



# Асинхронный язык

В асинхронный язык входят слова, в которых повторение любого из символов может быть произвольным. Точнее, если слово  $\tilde{a}$  принадлежит асинхронному языку, то в языке также содержатся все слова, полученные из  $\tilde{a}$  повторениями любых букв из  $\tilde{a}$  либо вычеркиванием из  $\tilde{a}$  некоторых повторений отдельных букв.

Функция переходов автомата, определяющего асинхронный язык, имеет следующую особенность: автомат может перейти в другое состояние только при изменении входного символа.

Назовем **ядром** асинхронного языка язык, в котором нет повторений символов. В частности, если ядро – дефинитный язык, то при синтезе автомата, распознающего асинхронный язык с таким ядром, можно воспользоваться приемом из пункта про дефинитный язык – сопоставить состояниям автомата префиксы распознаваемых слов ядра.



# Изоморфные и эквивалентные автоматы

Автоматы **изоморфны** если их описание одинаково с точностью до переобозначений.

Два инициальных автомата будем называть **эквивалентными** автоматами, если любую одну и ту же входную последовательность они перерабатывают в одну и ту же выходную последовательность. Неинициальные автоматы будем называть эквивалентными, если для любого состояния, взятого в качестве начального одного из автоматов, найдется в другом автомате состояние, назначив которое начальным получим эквивалентность переработки информации входной в выходную.



# Минимальные автоматы

Автомат, эквивалентный заданному и имеющий наименьшее возможное число состояний, называется **минимальным**. Если автоматы всюду определены и эквивалентны, то они имеют изоморфные минимальные автоматы. Для частично-определенных автоматов это положение может не выполняться.

Минимизация автомата возможна, если в автомате есть состояния, которые могут быть объединены в одно состояние. Такие состояния называют **эквивалентными состояниями**. Иначе говоря, автомат минимальный, если у него нет эквивалентных состояний.



# Достаточное условие эквивалентности

Достаточным условием эквивалентности состояний является совпадение соответствующих им строк в автоматной таблице. Такие состояния называют **явно эквивалентными**. Если строки одинаковы за исключением переходов в себя при одних и тех же входах, то такие состояния так же явно эквивалентны. Поэтому в автоматных таблицах следует переход в себя указывать не именем состояния, а каким-либо одинаковым символом, например таким: \$. После такой замены строки могут стать одинаковыми.



# Пример

Задание: Спроектировать автомат с двухразрядным входом ( $in1, in2$ ) и одноразрядным выходом ( $Z$ ), который индицирует следующее событие: после нулей на обеих линиях по каждой из линий  $in1$  и  $in2$  прошло ровно по одному стробу любой длительности.

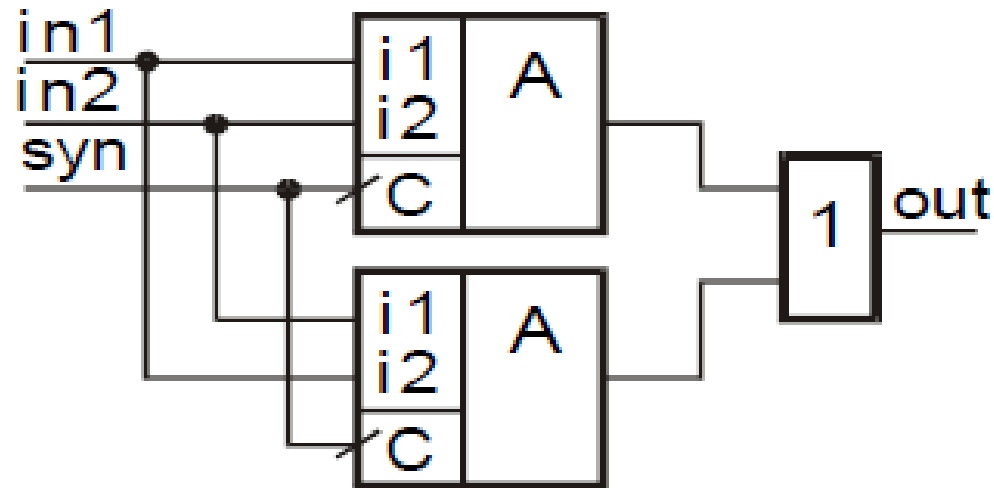
Строб – импульс («1» может продержаться всего один такт, может продержаться неопределенное время):





# Решение. Композиция автоматов

Решение будем искать в виде симметричной композиции двух одинаковых автоматов:



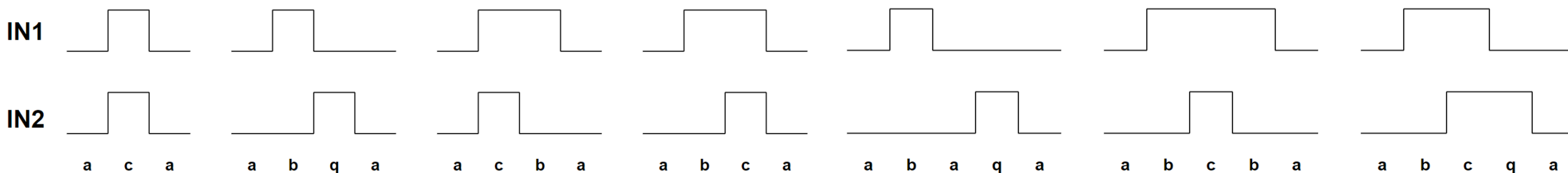


# Решение. Получение распознаваемых слов

Обозначим возможные символы (сочетания значений сигналов) на входах автомата следующим образом:

$$\frac{\text{in1}}{\text{in2}} \quad \frac{0}{0} = a \quad \frac{1}{0} = b \quad \frac{0}{1} = q \quad \frac{1}{1} = c$$

Если удалить все повторения символов, то получим для одного из автоматов дефинитный язык со следующими минимальными последовательностями, приводящими к событию, которое должен распознать автомат:



Другой автомат распознает последовательности, полученные из указанных заменой b на q, q на b.)





# Решение. Автоматная таблица

Автомат Мили задается следующей автоматной таблицей:

№	Q \ A	00	01	10	11	эквивалентности
		a	q	b	c	
0	$\lambda$	a	$\lambda$	$\lambda$	$\lambda$	
1	a	\$	$\lambda$	ab	ac	
2	ab	aba	abq	\$	abc	
3	ac	a/1	$\lambda$	acb	\$	
4	aba	\$	abaq	ab	ac	
5	abq	a/1	\$	$\lambda$	$\lambda$	$5 \equiv 8 \equiv 10$
6	abc	a/1	abcq	abcb	\$	
7	acb	a/1	$\lambda$	\$	$\lambda$	$7 \equiv 9$
8	abcq	a/1	\$	$\lambda$	$\lambda$	$5 \equiv 8 \equiv 10$
9	abcb	a/1	$\lambda$	\$	$\lambda$	$7 \equiv 9$
10	abaq	a/1	\$	$\lambda$	$\lambda$	$5 \equiv 8 \equiv 10$

Обозначения:

Q – текущее состояние

A – входы

\$ – переход в то же состояние

Слова, которые должен  
распознавать автомат:

aca  
abqa  
acba  
abca  
abaqa  
abcba  
abcqa



# Решение. Объединение эквивалентных состояний

Объединив эквивалентные состояния, получим таблицу автомата Мили:

№	Q \ A	00	01	10	11
		a	q	b	c
S0	$\lambda$	a	$\lambda$	$\lambda$	$\lambda$
S1	a	\$	$\lambda$	ab	ac
S2	ab	aba	abq	\$	abc
S3	ac	<b>a/1</b>	$\lambda$	acb	\$
S4	aba	\$	abaq	ab	ac
S5	abq	<b>a/1</b>	\$	$\lambda$	$\lambda$
S6	abc	<b>a/1</b>	abcq	abcb	\$
S7	acb	<b>a/1</b>	$\lambda$	\$	$\lambda$
S5	abcq	<b>a/1</b>	\$	$\lambda$	$\lambda$
S7	abcb	<b>a/1</b>	$\lambda$	\$	$\lambda$
S5	abaq	<b>a/1</b>	\$	$\lambda$	$\lambda$



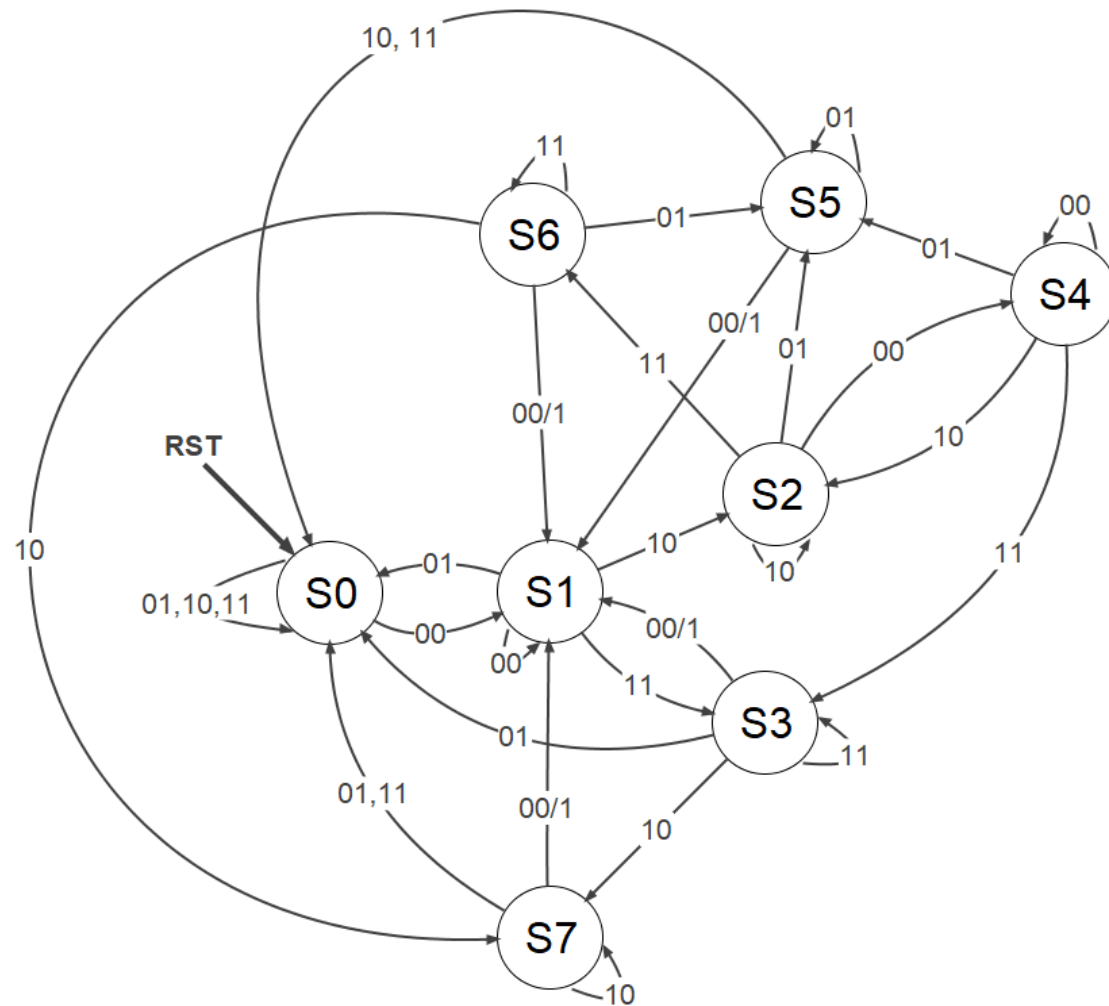
Q \ A	00	01	10	11
	a	q	b	c
S0	S1	S0	S0	S0
S1	S1	S0	S2	S3
S2	S4	S5	S2	S6
S3	<b>S1/1</b>	S0	S7	S3
S4	S4	S5	S2	S3
S5	<b>S1/1</b>	S5	S0	S0
S6	<b>S1/1</b>	S5	S7	S6
S7	<b>S1/1</b>	S0	S7	S0



# Решение. Граф переходов

На базе автоматной таблицы построим граф переходов состояний:

Q \ A				
	00	01	10	11
a	S1	S0	S0	S0
q	S1	S0	S2	S3
b	S4	S5	S2	S6
c	S1/1	S0	S7	S3
S0	S1	S0	S0	S0
S1	S1	S0	S2	S3
S2	S4	S5	S2	S6
S3	S1/1	S0	S7	S3
S4	S4	S5	S2	S3
S5	S1/1	S5	S0	S0
S6	S1/1	S5	S7	S6
S7	S1/1	S0	S7	S0





# Решение. Кодирование состояний

	00	01	10	11	
Q \ A	a	q	b	c	код
S0	S1	S0	S0	S0	000
S1	S1	S0	S2	S3	001
S2	S4	S5	S2	S6	010
S3	S1/1	S0	S7	S3	011
S4	S4	S5	S2	S3	100
S5	S1/1	S5	S0	S0	101
S6	S1/1	S5	S7	S6	110
S7	S1/1	S0	S7	S0	111

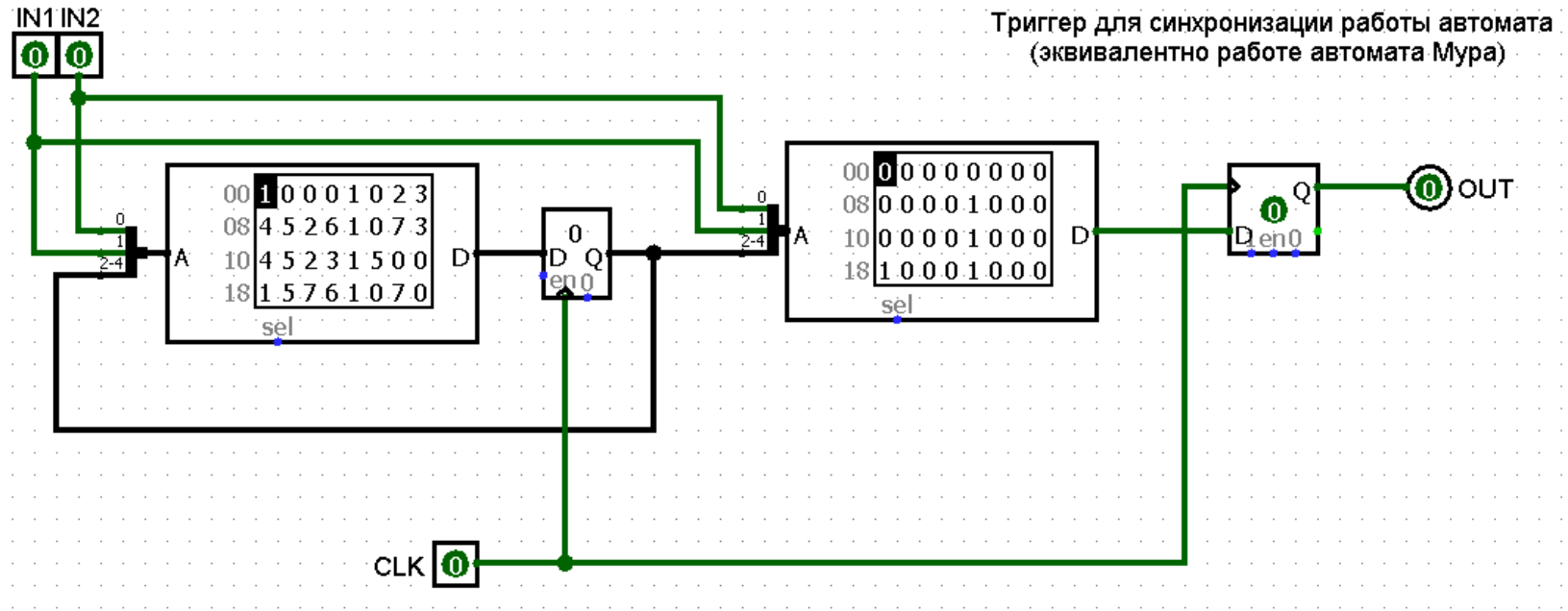
Кодировка состояний – произвольная. Чтобы закодировать 8 состояний необходим минимум трехразрядный код.

Но если использовать в качестве комбинационной логики формирования следующего состояния ПЗУ, то лучше будет закодировать состояния начиная с 000 и по увеличению на 1, для удобства «прошивки».



# Решение. Схема локального автомата

	00	01	10	11	
Q \ A	a	q	b	c	код
S0	S1	S0	S0	S0	000
S1	S1	S0	S2	S3	001
S2	S4	S5	S2	S6	010
S3	S1/1	S0	S7	S3	011
S4	S4	S5	S2	S3	100
S5	S1/1	S5	S0	S0	101
S6	S1/1	S5	S7	S6	110
S7	S1/1	S0	S7	S0	111





# Решение. Схема полного автомата

Индикация события: после нулей на обеих линиях по каждой из линий IN1 и IN2 прошло ровно по одному стробу любой длительности

