

УДК 681.32

## Синтез автомата-распознавателя кода

**С. Ф. Тюрин**

Пермский национальный исследовательский политехнический университет

Россия, 614990, г. Пермь, ул. Комсомольский пр., 29

tyurinsergfe@mail.ru; +7-952-32-02-510

Пермский государственный национальный исследовательский университет

Россия, 614990, г. Пермь, ул. Букирева, 15

Описывается методика синтеза и моделирования синхронного автомата-распознавателя заданной последовательности с повторяющимися символами в рамках курсовой работы на основе схем-шаблонов как продолжения лабораторного занятия. Показывается, что в шаблоне блок формирования кода клавиши может быть реализован на базе кодера, а логика переходов и выходов – на основе дешифраторов. Выполняется моделирование в системе схемотехнического моделирования NI Multisim 10 фирмы National Instruments Electronics Workbench Group. Делается вывод о целесообразности использования предложенного подхода для простой последовательности в лабораторных работах и усложненных версий синхронного автомата во время курсовой работы студентов.

**Ключевые слова:** синтез автомата-распознавателя; схема-шаблон; лабораторное занятие; курсовая работа.

DOI: 10.17072/1993-0550-2017-4-92-95

### Введение

В современных условиях высшего образования при непреклонной реализации учебных программ бакалавриата в России преподаватели часто сталкиваются с катастрофическим дефицитом времени. Это приводит как к невозможности в полной мере реализовать полноценные эксперименты на лабораторных занятиях, так и к угасанию интереса обучаемых, на которых буквально сваливаются горы информации и валы различных заданий, отчетов и пр.

В этой связи предлагается своего рода масштабирование учебных заданий с использованием шаблонов, которые дают возможность значительного сокращения времени на проведение эксперимента, так как требуется выполнить только небольшие изменения по варианту. В случае, если изменения не были сделаны, но студент может продемонстрировать хотя бы работу шаблона, это уже свидетельствует о каком-то минимальном уровне знаний! Данная методология также упрощает преподавателю проверку знаний, умений и владений студента.

Вместе с тем небольшое увеличение размерности задачи может привести к изменению шаблона, и подобное задание может быть использовано в качестве курсовой работы. Уменьшение размерности задания вполне подойдет для использования на зачете (экзамене).

Рассмотрим особенности предлагаемого подхода на примере синтеза синхронного логического автомата-распознавателя кода (кодового замка – распознаватель PIN кода) [1–3].

### 1. Блоки шаблона автомата-распознавателя кода

Рассмотрим пример синтеза синхронного автомата, распознающего последовательность на входах *ав*: "0022" (PIN код).

На правильную последовательность срабатывает индикатор *z1*, что может быть интерпретировано как "замок открыть". Для всех неверных наборов срабатывает *z2*, что значит "подать сигнал тревоги" или "замок заблокировать". Мы не будем подсчитывать количество формирования неверных последовательностей. Не будем учитывать внешний сигнал общего сброса – его будем осуществлять по штатным входам сброса триггеров.

В отличие от входных сигналов, реализуемых статическими ключами с двумя положениями "0" и "1", необходимы динамические ключи, поэтому необходим блок формирования кода ключа (клавиши) и формирователь сигнала активирования одного из ключей (клавиш).

Проведем блочный синтез шаблона автомата-распознавателя кода (рис. 1).

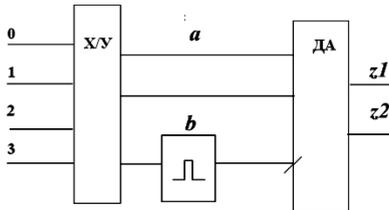


Рис. 1. Блочный синтез шаблона автомата-распознавателя кода

При нажатии кнопки блок формирования кода ключа – кодопреобразователь  $x/y$  формирует код  $ab$  и активирует одновибратор, синхронизирующий наш синхронный дискретный автомат ДА.

Переход осуществляется при нажатии очередной кнопки: кнопку нажали – срабатывает одновибратор и формирует один синхрипульс. Реализацию кодопреобразователя выполним в системе схемотехнического моделирования NI Multisim 10 фирмы National Instruments Electronics Workbench [4].

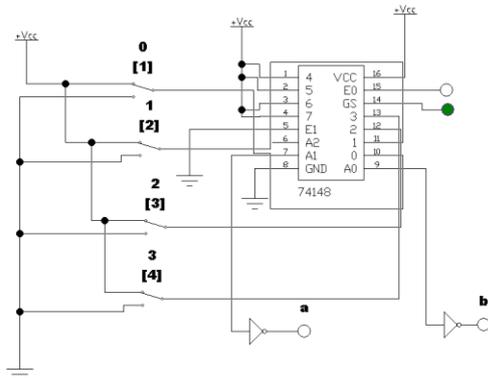


Рис. 2. Реализация кодопреобразователя (в системе схемотехнического моделирования NI Multisim 10 фирмы National Instruments Electronics Workbench Group)

Поскольку этот кодопреобразователь имеет выходы фиксации активации одного из входов  $E0$  и  $GS$ , то необходимость в одновибраторе отпадает – будем использовать  $GS$  для синхронизации триггеров памяти автомата.

## 2. Абстрактный синтез автомата-распознавателя с повторяющимися символами

Приступим к этапу абстрактного синтеза автомата. Построим граф переходов для синхронного автомата Мура (рис. 3):

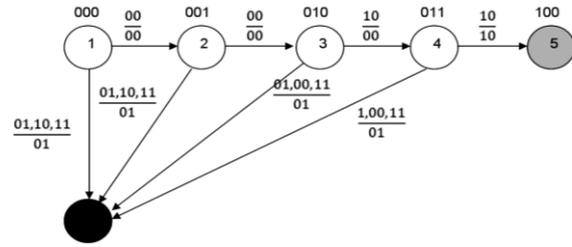


Рис. 3. Граф переходов автомата 0022

На рис. 3. стрелками обозначены переходы из одного состояния в другое. Дробные значения над стрелками: в числителе условия переходов в последующее состояние, в знаменателе – получаемое выходное значение. Цветом обозначены конечные состояния автомата: 5 – для разрешенной последовательности 0022 (6, соответственно, для запрещенных).

Используя рис. 3 строим таблицу переходов автомата (табл. 1):

Таблица 1

Переходы автомата Мура 0022

№	ab			
	00	01	10	11
1	2	6	6	6
2	3	6	6	6
3	6	6	4	6
4	6	6	5	6
5	5	5	5	5
6	6	6	6	6

В состоянии 5 формируется сигнал  $z1$  – открытие "замка", а в состоянии 6 –  $z2$  – "тревога" ("подбор кода").

Построим таблицу переходов-выходов автомата Мура (табл. 2):

Таблица 2

Переходы-выходы автомата Мура 0022

$y_3y_2y_1(t)$	$z_1z_2$	ab			
		00	01	10	11
000	00	001	101	101	101
001	00	010	101	101	101
010	00	101	101	011	101
011	00	101	101	100	101
100	10	100	100	100	100
101	01	101	101	101	101
					$y_3y_2y_1(t+1)$

В табл. 2  $y_3y_2y_1(t)$  – текущее состояние автомата,  $y_3y_2y_1(t+1)$  – следующее состояние,  $Z_1, Z_2$  – выходные значения.

### 3. Структурный синтез автомата-распознавателя с повторяющимися символами

Реализацию логических функций выполним на основе дешифраторов 74 154.

Память построим на синхронных триггерах D-типа. Шаблон для схемы управления памятью имеет вид (рис. 4):

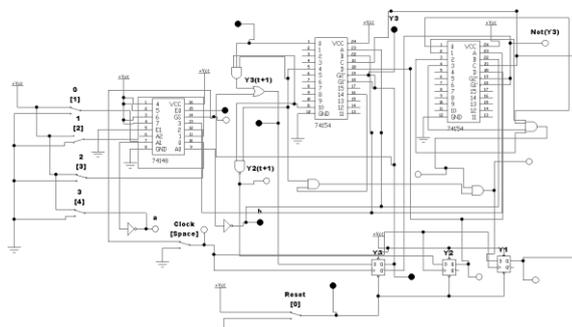


Рис. 4. Реализация логических функций  $y_3(t+1)y_2(t+1)y_1(t+1)$

Построим подсхему реализации  $z_1 z_2$  для соответствующего автомата Мура: в состоянии 4 (100) активируется  $z_1$ , в состоянии 5 (101) активируется  $z_2$ .

Исходное состояние (рис. 5):

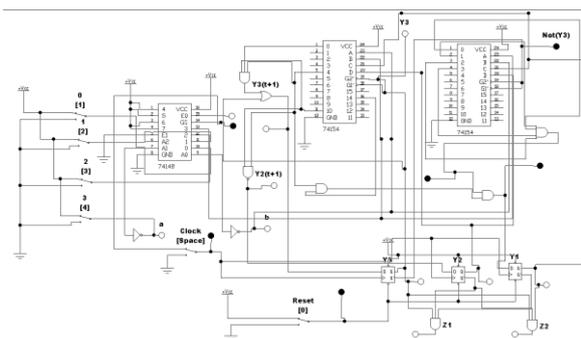


Рис. 5. Реализация логических функций  $z_1 z_2$

### 4. Методика тестирования автомата-распознавателя с повторяющимися символами

Проверяем реакцию автомата на правильную последовательность 0022 по табл. 2. Проводим общий сброс-состояние автомата

000. Нажимаем 0, автомат переходит в состояние 001 (рис. 6):

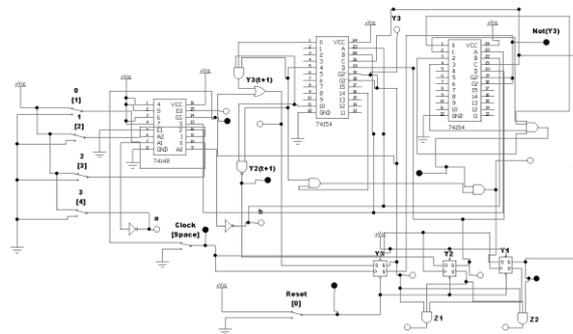


Рис. 6. Реакция автомата на нажатие клавиши "0"-переход в состояние 001

Еще раз нажимаем 0, автомат переходит в состояние 010 (рис. 7):

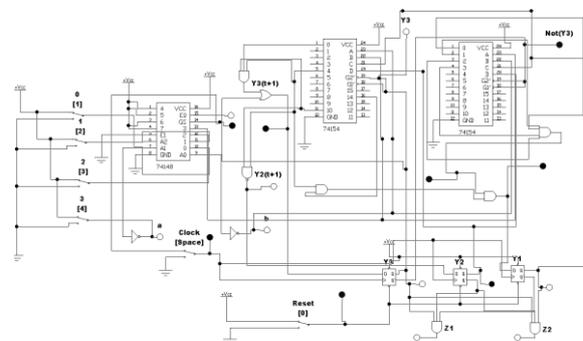


Рис. 7. Реакция автомата на повторное нажатие клавиши "0"-переход в состояние 010

Далее два раза нажимаем 2 – автомат переходит в состояние 4 (100) и формируется сигнал –  $z_1$  (рис. 8):

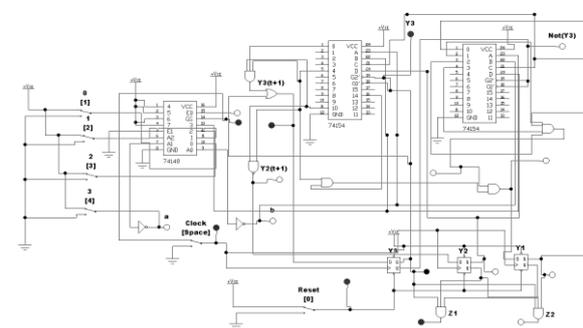


Рис. 8. Правильная последовательность: переход в состояние 100

При нарушении последовательности, например, 0023 – автомат переходит в состояние 5(101) и формируется сигнал  $z_2$  (рис. 9):

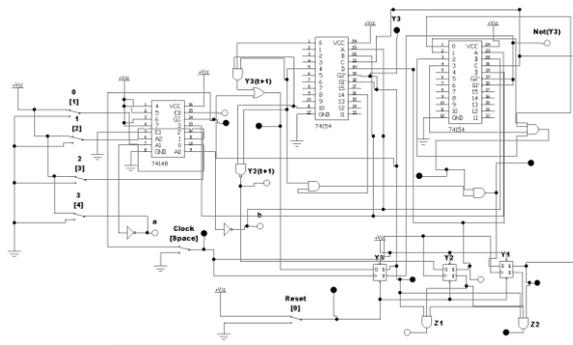


Рис. 9. Реакция автомата на последовательность 0023-переход в состояние 101, формирование сигнала z2

### Выводы

Таким образом, синтез автомата-распознавателя с повторяющимися символами может быть осуществлен без минимизации логических функций в системе Electronics Workbench по заданному шаблону (он может быть в библиотеке схем) на базе кодера и дешифраторов. Кодер 74148 позволяет формировать код одной из восьми нажатых клавиш и сигнал активации клавиш, который может быть использован для синхронизации триггеров. В рамках 4-часовой лабораторной работы целесообразна реализация автомата, реагирующего на последовательности нажатий 4 клавиш. Более сложный автомат с восемью кла-

вишами может быть вынесен на курсовую работу студентов.

Таким образом, реализация синхронного автомата-распознавателя позволяет сформировать существенно большее число вариантов, чем для автомата с неповторяющимися символами.

Предлагаемая методика может обеспечить масштабирование заданий и времени их выполнения (контроля), что в условиях дефицита времени вносит определенный вклад в дело повышения качества подготовки отечественных специалистов в области электронной компонентной базы.

### Список литературы

1. Тюрин С.Ф., Аляев Ю.А. Дискретная математика: практическая дискретная математика и математическая логика. М.: Финансы и статистика, 2010. 394 с.
2. Тюрин С.Ф., Громов О.А., Греков А.В. Реализация цифровых автоматов в системе Quartus фирмы Altera: лабораторный практикум. Пермь: Изд-во ПНИПУ, 2011. 133 с.
3. Угрюмов Е.П. Цифровая схемотехника: учеб. пособие. 2-е изд., перераб. и доп. СПб.: БХВ-Петербург, 2007. 782 с.
4. Сайт разработчика National Instruments. URL: // <http://www.ni.com/multisim/> (дата обращения: 22.12.2016).

## Design of the finite state machine-recognizer

S. F. Tyurin

Perm National Research Polytechnic University; 29, Komsomolsky pr., Perm, 614990, Russia

Perm State University; 15, Bukireva st., Perm, 614990, Russia

tyurinsergfeo@yandex.ru; +7 952-320-02-510

The paper describes the procedure for the synthesis and modeling of a synchronous finite state machine-recognizer of a specified sequence with recurring symbols as part of course work ("Circuitry" discipline) based on templates and being a continuation of laboratory classes. It is shown that in such a pattern, the key code forming unit can be implemented based on an encoder, and the logic of transition and output – on decoders. The simulation is performed in NI Multisim 10 circuit simulation system by National Instruments Electronics Workbench Group. The conclusion is drawn about the applicability of the proposed approach for a simple sequence within laboratory classes and sophisticated versions of a synchronous machine-recognizer in students' course work.

**Keywords:** *finite state machine-recognizer; templates; laboratory classes; course work.*