

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЯДЕРНЫЙ УНИВЕРСИТЕТ  
«МОСКОВСКИЙ ИНЖЕНЕРНО-ФИЗИЧЕСКИЙ ИНСТИТУТ»**



**Институт Интеллектуальных Кибернетических Систем  
Кафедра «Компьютерные системы и технологии»**

**Отчёт о лабораторной работе №4  
«Программная реализация конечного автомата»  
по курсу «Микропроцессорные устройства и системы»**

**Студент группы Б20-503** Коломенский В. Г. / \_\_\_\_\_ /  
**Руководитель** \_\_\_\_\_ / \_\_\_\_\_ /

**Москва 2023**

## Оглавление

1. Введение.....	1
2. Реализация программы .....	2
3. Тестирование программы.....	6
4. Список литературы и ссылки.....	8

## 1. Введение

По заданию лабораторной работы необходимо разработать автомат, который распознает заданную последовательность одинарных переключений линий порта p0.

Условие представлено на рисунке 1.

Задания для выполнения лабораторной работы №4						
Для группы Б20-503						
Студент	№ варианта	Переключения	N1	N2	N3	N4
КОЛОМЕНСКИЙ	7 503	5,4,1,0	5	$\text{Min}( N1-3 , 2)$	$(3 \cdot N1 + 2 \cdot N2) \bmod 3$	$ N2 - N1 - N3 $

*Рисунок 1. – Условие варианта 7\_503*

N1, N2, N3, N4 – определяют количество допустимых ошибок при вводе очередного переключения. При определении допустимого количества ошибок начиная со 2-го переключения учитывается количество ошибок, допущенных при вводе предшествующих переключений.

При нарушении условий по допустимым ошибкам происходит блокировка работы системы, а в порт p1 выводится код «aah».

Если распознавание переключения выполнено корректно с допустимым количеством ошибок ввода на каждом шаге, то в порт p1 выводится код 55h, а в порт p2 выводится hex номер варианта задания.

## 2.Реализация программы

Реализация программы представлена в приложении 1 ниже.

### Приложение 1.

```
1 ; P0 - input
2 ; P1 - state
3 ; P2 - variant if succes
4
5 ; P3 - number of misses at this time (to del)
6 ; P4 - maximum allowable number of misses at this time (to del)
7
8 ; R0 - previous key (P0) set
9 ; R1 - number of misses in the 1st state
10 ; R2 - number of misses in the 2nd state
11 ; R3 - number of misses in the 3rd state
12 ; R4 - number of misses in the 4th state
13 ; R5 - maximum allowable number of misses in current state (P1)
14 ; R6 - shows changed bit
15 ; R7 - register for calc (tmp register)
16
17
18 P4 EQU 0C0h ; TODEL
19 mov P0, #0ffh
20 mov P1, #01h
21 mov P2, #00h
22 mov R0, P0
23 mov R1, #00h
24 mov R2, #00h
25 mov R3, #00h
26 mov R4, #00h
27 mov R5, #05h
28 mov P3, R1 ; TODEL
29 mov P4, R5 ; TODEL
30
31
32 wait_input:
33     mov A, P0
34     xrl A, R0
35     jnz process_input
36     jmp wait_input
37
38
39 process_input:
40     mov R0, P0
41     mov R6, A
42     mov A, P1
43     cjne A, #01h, pi_1
44     jmp state_1
45 pi_1:
46     cjne A, #02h, pi_2
47     jmp state_2
48 pi_2:
```

```

49     cjne A, #03h, pi_3
50     jmp state_3
51 pi_3:
52     cjne A, #04h, pi_4
53     jmp state_4
54 pi_4:
55     jmp failed
56
57
58 state_1:
59     mov A, R6
60     jb ACC.5, s1_1
61     mov A, R5
62     clr C
63     subb A, R1
64     jnz s1_0
65     jmp failed
66 s1_0:
67     inc R1
68     mov P3, R1 ; TODEL
69     jmp wait_input
70 s1_1:
71     mov P1, #02h
72     mov A, R1
73     clr C
74     subb A, #03h
75     jc s1_2
76     mov A, R1
77     clr C
78     subb A, #03h
79     mov R7, A
80     jmp s1_3
81 s1_2:
82     mov A, #03h
83     clr C
84     subb A, R1
85     mov R7, A
86 s1_3:
87     mov A, R7
88     clr C
89     subb A, #02h
90     jc s1_4
91     mov R5, #02h
92     jmp s1_5
93 s1_4:
94     mov A, R7
95     mov R5, A
96 s1_5:
97     mov P3, R2 ; TODEL
98     mov P4, R5 ; TODEL
99     jmp wait_input
100
101
102 state_2:
103     mov A, R6
104     jb ACC.4, s2_1
105     mov A, R5
106     clr C

```

```

107     subb A, R2
108     jz failed
109     inc R2
110     mov P3, R2 ; TODEL
111     jmp wait_input
112 s2_1:
113     mov P1, #03h
114     mov A, R1
115     mov B, #03h
116     mul AB
117     mov R7, A
118     mov A, R2
119     mov B, #02h
120     mul AB
121     add A, R7
122     mov B, #03h
123     div AB
124     mov R5, B
125     mov P3, R3 ; TODEL
126     mov P4, R5 ; TODEL
127     jmp wait_input
128
129
130 state_3:
131     mov A, R6
132     jb ACC.1, s3_1
133     mov A, R5
134     clr C
135     subb A, R3
136     jz failed
137     inc R3
138     mov P3, R3 ; TODEL
139     jmp wait_input
140 s3_1:
141     mov P1, #04h
142     mov A, R1
143     add A, R3
144     mov R7, A
145     clr C
146     subb A, R2
147     jc s3_2
148     mov A, R7
149     clr C
150     subb A, R2
151     mov R5, A
152     jmp s3_3
153 s3_2:
154     mov A, R2
155     clr C
156     subb A, R7
157     mov R5, A
158 s3_3:
159     mov P3, R4 ; TODEL
160     mov P4, R5 ; TODEL
161     jmp wait_input
162
163
164 state_4:

```

```

165     mov A, R6
166     jb ACC.0, passed
167     mov A, R5
168     clr C
169     subb A, R4
170     jz failed
171     inc R4
172     mov P3, R4 ; TODEL
173     jmp wait_input
174
175
176 failed:
177     mov P1, #0AAh
178     jmp $
179
180
181 passed:
182     mov P1, #55h
183     mov P2, #07h
184     jmp $
185
186 end

```

### 3. Тестирование программы

Разработанная программа тестировалась на примерах, представленных на рисунке 2. В этих примерах проверяются все «интересные» случаи: блокировка автомата на каждом из возможных состояний, ввод правильной комбинации и т.п. Также проверяется корректность вычисления  $N2-N4$ . К примеру, для операции «модуль числа» проверяется 3 случая: значение под модулем больше нуля, равно нулю, меньше нуля. Подобные проверки делаются и для операции взятия минимального значения.

Тестирование лаб. 4				
$N^{\circ}$	$N1$	$N2 = \text{Min}( N1-3 , 2)$	$N3 = (3N1 + 2N2) \bmod 3$	$N4 =  N2 - N1 - N3 $
1	$5 - 6 = \underline{-1}$	×	×	×
2	$5 - 0 = 5$	$\begin{matrix} < > \\ 2 - 1 = 1 \end{matrix}$	$\textcircled{2} - 1 = 1$	$\textcircled{5} - 1 = \underline{-1}$
3	$5 - 3 = 2$	$\begin{matrix} = < \\ 0 - 0 = 0 \end{matrix}$	$\textcircled{0} - 1 = \underline{-1}$	×
4	$5 - 5 = 0$	$\begin{matrix} > = \\ 2 - 2 = 0 \end{matrix}$	$\textcircled{1} - 0 = 1$	$\begin{matrix} < \\ 3 - 0 = \underline{3} \end{matrix}$
5	$5 - 2 = 3$	$1 - 2 = \underline{-1}$	×	×
6	$5 - 0 = 5$	$2 - 2 = 0$	$1 - 0 = 1$	$\begin{matrix} > \\ 2 - 0 = \underline{2} \end{matrix}$

Рисунок 2. – Тестирование программы

Ввиду невозможности адекватно представить результаты тестирования в виде изображений будет показаны только результат ввода правильной последовательности переключений и результат ввода неправильной последовательности с превышением допустимого количества ошибок (см. рисунок 3.1 и 3.2).



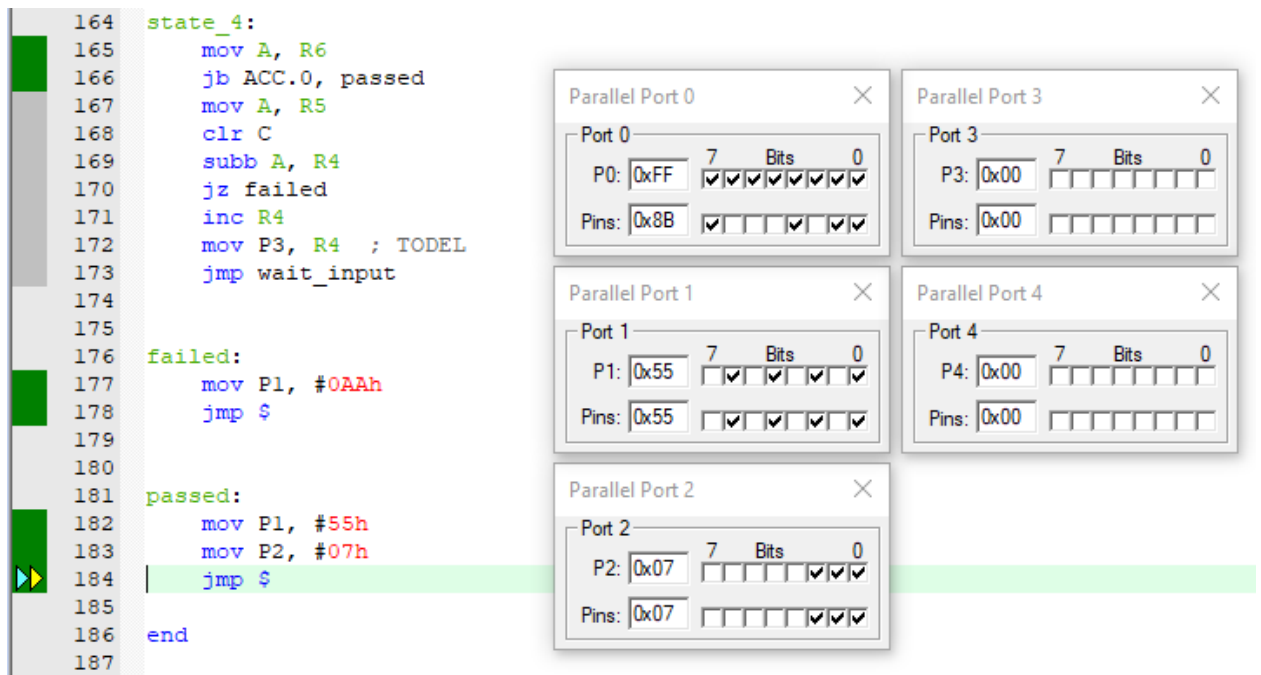


Рисунок 3.1 – Успех

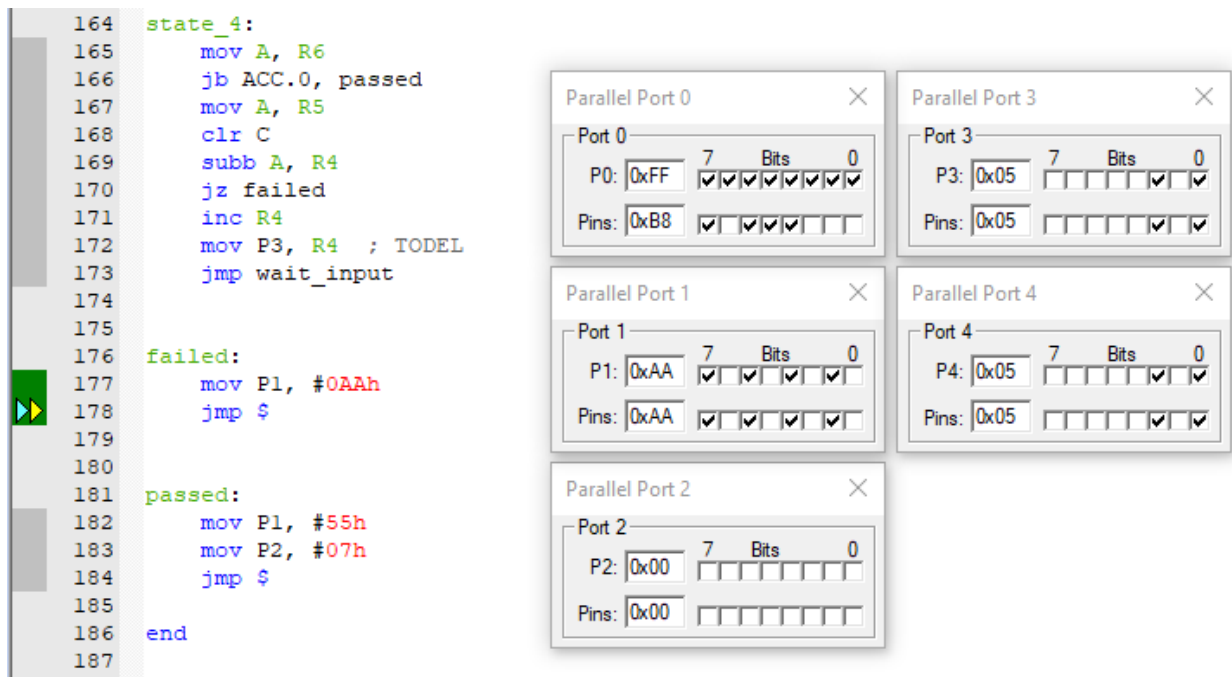


Рисунок 3.2 – Блокировка автомата

#### 4. Список литературы и ссылки

1. Стрелец А. И., Иванников В. С., Ёхин М.Н. Методические указания для выполнения лабораторной работы “Битовый процессор” по курсу “Микропроцессорные устройства и системы” с использованием виртуального стенда. Москва 2018.
2. Е. В. Моисейкин. Микроконтроллеры семейства MCS-51 Теория и практика. Учебно-методическое пособие. Екатеринбург Издательство Уральского университета 2017.
3. Исходный код программы [Электронный ресурс] // <https://github.com/Hypex146/MDnS-Lab-4>.