

Студент	№ варианта	Последовательность переключений	N1		N2	N3	N4
Серeda	2404	2,2,7,5	4		$\max(\text{mod}_3(N1+1), 2)$	$\min(3 \cdot N1, 2 \cdot N2, 1)$	$ \max(N2, N3) - 2 \cdot N1 $

Цель работы

Цель работы - разработать автомат, который распознает заданную последовательность одинарных переключений линий порта P0 (каждый раз меняется состояние только одной линии - входной одинарный разностный сигнал). То есть входным сигналом для автомата является не само значение, а изменение значения на заданной линии порта.

Данный способ задания входных сигналов ориентирован на реализацию автомата с использованием виртуального стенда. состояния виртуальных тумблеров, подключенных к линиям порта P0 в этом стенде меняются одним «кликом» мыши. ясно, что одним кликом можно изменить значение только одной линии порта.

Образец задания

№ варианта	последовательность переключений	N1	N2	N3	N4
xxxx	2, 7, 1, 5	2	$N1 \bmod 2$	$N1 + N2$	$\max(N3, 3)$

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.

Пример: при вводе первого переключения допущена одна ошибка (что допустимо). Тогда $N2 = N1 \bmod 2 = 1$. Если при вводе второго переключения ошибок не допущено, то $N3 = 1 + 0 = 1$.

Вывод контрольной информации

- После запуска программы в порт P1 выводится номер очередного вводимого переключения (1,2,3,4).
- 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.

Примечание: предварительно все линии портов P1 и P2 устанавливаются в нулевое состояние. Начальное состояние линий порта P0 произвольное.

Циклический буфер

- Реализация программного автомата предполагает выполнение циклического опроса порта p0.
- Как фиксировать изменение состояния линии порта «p0» при его циклическом опросе?
- Идея – использовать циклический буфер длиной «2» для хранения предыдущего и текущего значений порта.
- Линия, изменившая состояние определяется путем выполнения логической операции исключающее «или» над содержимым двух ячеек буфера.

Тестовая программа циклического буфера

Программа реализует циклический буфер с двумя ячейками внутренней памяти данных с адресами 24h и 25h. Для имитации циклической записи в буфер используется инкрементируемый счетчик на основе регистра «R7». Результат заполнения буфера можно наблюдать в окне «память 1». Тестирование программы удобно выполнять при помощи установки контрольной точки на строку «15». Нажатием «F5» выполняется одна итерация цикла.

The screenshot displays the uVision IDE interface for a project named "CIRC_BUF.uvproj". The main window shows the assembly code for the "CIRC_BUF.a51" file. The code includes a loop that increments a counter (R7) and jumps to a label "MET1" until it reaches 0x07. The "Registers" window on the left shows the current state of the registers, with R7 set to 0x04. The "Memory 1" window on the right shows a memory dump starting at address 0x00, with the value at 0x24 being 02 03.

Registers

Register	Value
r0	0x24
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x04
a	0x24
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC	0x0000E
states	34
sec	0.00001360
psw	0x00

Disassembly

```
15: AJMP MET1;
C:0x0000E 0107 AJMP MET1 (C:0007)
C:0x0010 00 NOP
C:0x0011 00 NOP
C:0x0012 00 NOP
C:0x0013 00 NOP
C:0x0014 00 NOP
C:0x0015 00 NOP
```

CIRC_BUF.a51

```
5
6 MOV R7, #00h; IMITATOR PORTA
7 MOV RO, #24h; ADRES BUFERA 0
8 MOV 25h, #00h; ADRES BUFERA 1
9 MET1:
10 MOV @RO, 07h; PRIEM DANNIH V BUFER 0
11 MOV A, RO;
12 XRL A, #01h;
13 MOV RO, A; IZMENENIE ADRESA BUFERA
14 INC R7; INCREMENT IMITATORA
15 AJMP MET1;
16 END
```

Command

```
Running with Code Size Limit: 2K
Load "G:\МЕТОДИЧЕСКАЯ РАБОТА_УЧЕБНЫЕ ПЛАНЫ\ПРАКТИКУМ_М
BS \CIRC_BUF\15
```

Memory 1

Address: 0x00

I:0x00:	24 00 00 00 00 00 00 04 00
I:0x09:	00 00 00 00 00 00 00 00 00
I:0x12:	00 00 00 00 00 00 00 00 00
I:0x1B:	00 00 00 00 00 00 00 00 00
I:0x24:	02 03 00 00 00 00 00 00 00
I:0x2D:	00 00 00 00 00 00 00 00 00
I:0x36:	00 00 00 00 00 00 00 00 00
I:0x3F:	00 00 00 00 00 00 00 00 00

ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList

Циклический буфер с вводом данных через порт «P0»

Ячейки буфера – с адресами 24h и 25h. При изменении данных, принимаемых через порт «P0» с 75 на 74 в аккумулятор после выполнения 16 строки загружается значение 01. Фиксируется изменение в «0» разряде данных, принимаемых через порт «P0».

Registers

Register	Value
r0	0x25
r1	0x74
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
a	0x01
b	0x00
sp	0x07
sp_max	0x07
dptr	0x0000
PC	0x0014
states	122
sec	0.00004880
psw	0x01

Disassembly

```

17:      NOP
C:0x0014 00      NOP
18:      AJMP MET1;
C:0x0015 0108    AJMP MET1(C:0008)
C:0x0017 00      NOP
C:0x0018 00      NOP
C:0x0019 00      NOP

```

Source: CIRC_PORT.a51

```

3      ; ПРОГРАММА ЦИКЛИЧЕСКИ ЧИТАЕТ
4      ; И ЗАПИСЫВАЕТ В БУФЕР
5      MOV PO,#0FFh
6
7      MOV R0, #25h; АДРЕС БУФЕРА 0
8      MOV 24h,#00h; АДРЕС БУФЕРА 1
9  MET1: MOV A, PO; ВВОД ПОРТО
10     MOV @R0,A; ПРИЕМ ДАННЫХ В БУФЕР
11     MOV A, R0;
12     XRL A, #01h;
13     MOV R0, A; ИЗМЕНЕНИЕ АДРЕСА БУФЕРА
14     MOV R1, 24h
15     MOV A, 25h
16     XRL A, R1
17     NOP
18     AJMP MET1;
19     END

```

Parallel Port 0

Port 0: 0xFF 7 Bits 0

Pins: 0x74

Command

Running with Code Size Limit: 2K

ASM ASSIGN BreakDisable BreakEnable BreakKill

Memory 1

Address: 0:00

```

I:0x18: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x24: 74 75 00 00 00 00 00 00 00 00 00 00 00 00 00 00
I:0x30: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

```