**Отчет по лабораторной работе «*Применение микроконтроллеров для создания сложных систем*»**

Вариант 4

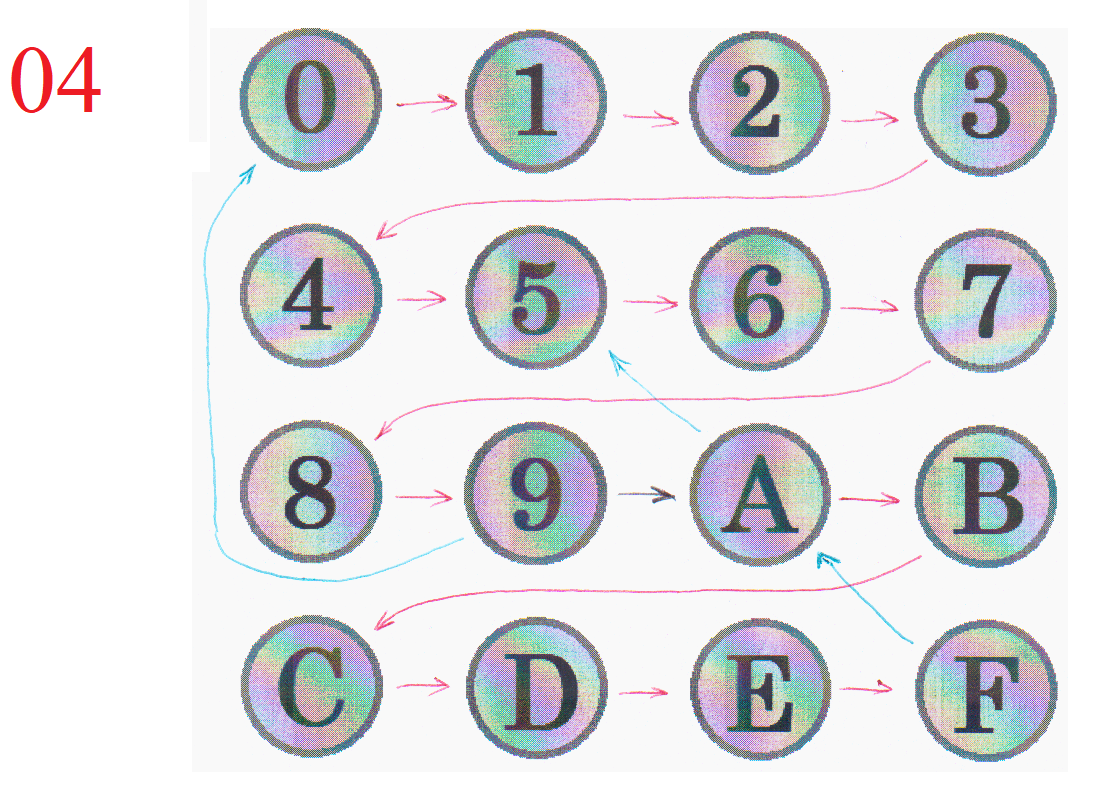
Исполнитель: Преподаватель:

Белоусов Е. А. Нестеров Ю.Г.

ИУ5-71

Москва, МГТУ – 2020

1. **Граф переходов системы:**



1. **Схема алгоритма работы системы**



1. **Программа для микроконтроллера ATmega16, реализующая разработанный алгоритм**

.include "m16def.inc" ; подключение библиотеки для работы с ATmega16

.list ; включение листинга

.def temp=r16 ; определение главного рабочего регистра

.def kold=r17

.def k\_\_\_=r18

.def s\_\_\_=r19

;--------------------------------------------

.cseg ; выбор сегмента программного кода

.org 0 ; установка текущего адреса на ноль

;--------------------------------------------

ldi temp,0x80 ; выключение компаратора

out acsr,temp

;--------------------------------------------

ldi temp,0x00 ; 0 --> temp

out ddrd,temp ; Назначаем порт rd на ввод (00000000 --> ddrd)

ldi temp,0xFF ; 0xff --> temp

out ddrb,temp ; Назначаем порт rb на вывод (11111111 --> ddrb)

out portd,temp ; Подключаем подтягивающие резисторы (11111111 --> portd)

;---------------------------------------------

ldi kold, 0x00 ; 0--->kold

ldi s\_\_\_, 0x00 ; 0--->s\_\_\_

out portb, s\_\_\_ ; Вывели s\_\_\_(=0)

;---------------------------------------------

read: ;

in k\_\_\_, pind ; Считали содержимое порта pd (--->k\_\_\_)

cp k\_\_\_, kold ; Сравнили k\_\_\_ и kold

breq read ; Если k\_\_\_=kold, read

tst kold ; Проверили kold

brne remem ; Если kold!=0, remen

tst k\_\_\_ ; Проверили k\_\_\_

breq remem ; Если k\_\_\_=0, remem

jmp lbl1 ;

;---------------------------------------------

remem: ;

mov kold, k\_\_\_ ; Записали k\_\_\_ в kold

jmp read ; Вернулись в read

;---------------------------------------------

lbl1: ;

cpi k\_\_\_, 0x01 ; Сравнили k\_\_\_ и 1

brne lbl2 ; Если k\_\_\_!=1, lbl2

cpi s\_\_\_, 0x09 ; Сравнили s\_\_\_ и 9

breq print ; Если s\_\_\_=9, Вывод

cpi s\_\_\_, 0x0F ; Сравнили s\_\_\_ и F

breq print ; Если s\_\_\_=F, то Вывод ;

inc s\_\_\_ ; s\_\_\_+1--->s\_\_\_

jmp print ; Перешли на Вывод

;---------------------------------------------

lbl2: ;

cpi k\_\_\_, 0x02 ; Сравнили k\_\_\_ и 2

brne lbl3 ; Если k\_\_\_!=2, lbl3

cpi s\_\_\_, 0x09 ; Сравнили s\_\_\_ и 9

brne print ; Если s\_\_\_ != 9, то Вывод

inc s\_\_\_ ; s\_\_\_+1--->s\_\_\_

jmp print ; Перешли на Вывод

;---------------------------------------------

lbl3:

cpi k\_\_\_, 0x4 ; Cравнили k\_\_\_ и 4

brne remem ; Если k\_\_\_ != 4, то перешли на remem

cpi s\_\_\_, 0x0F ; Сравнили s\_\_\_ и F

breq lbl3\_1 ; Если s\_\_\_ = F, то lbl3\_1

cpi s\_\_\_, 0x0A ; Сравнили s\_\_\_ и A

breq lbl3\_2 ; Если s\_\_\_ = A, то lbl3\_2

cpi s\_\_\_, 0x09 ; Сравнили s\_\_\_ и 9

breq lbl3\_3 ; Если s\_\_\_ =9 , то lbl3\_3

jmp print ; Перешли на Вывод

lbl3\_1:

ldi s\_\_\_, 0x0A ; s\_\_\_:= A

jmp print ; Перешли на Вывод

lbl3\_2:

ldi s\_\_\_, 0x05 ; s\_\_\_:= 5

jmp print ; Перешли на Вывод

lbl3\_3:

clr s\_\_\_ ; s\_\_\_:= 0

jmp print ; Перешли на Вывод

;---------------------------------------------

print: ;

out portb, s\_\_\_ ; Вывели s\_\_\_ в порт pb

jmp remem

1. **Модель системы в Proteus**

