Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления» (ИУ)

КАФЕДРА «Информационная безопасность» (ИУ8)

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

АППАРАТНЫЕ СРЕДСТВА ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

«Исследование битовых команд микроконтроллера i8051»

Преподаватель:

Рафиков А.Г. _____
(подпись, дата)

Студент:

Девяткин Е.Д., группа ИУ8-74 (4 курс) _____
(подпись, дата)

Москва, 2025

Содержание

Цель работы

Собрать в Протеусе модель на МК i8051. Реализовать ФАЛ следующими способами:

– С помощью битовых команд (см. семинар);

– С помощью проверки булевых переменных (с помощью логических переходов);

– С помощью байтовых команд;

Теоретическая часть

На рисунке 1 микроконтроллер 8051:

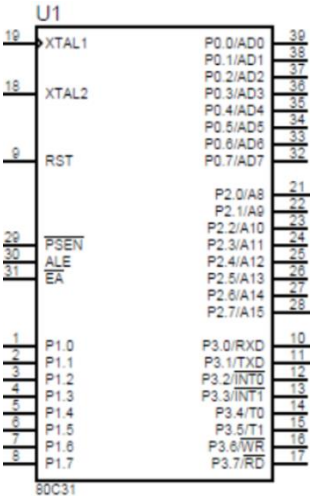


Рис. 1 – Схема микроконтроллера i8051 в Proteus.

Входные и выходные сигналы микроконтроллера i8051 имеют следующие назначения:

– XTAL и XTAL2 – входы подключения кварцевого резонатора для работы генератора тактовой частоты микроконтроллера;

– PSEN – сигнал, используемый при обращении к внешней памяти программ;

– ALE – выходной сигнал разрешения фиксации адреса при обращении к внешней памяти программ/данных;

– EA – сигнал, блокирующий работу с внутренней памятью;

– RST – сигнал общего сброса;

– P0 – P3 – выводы портов ввода/вывода микроконтроллера;

– Vss и Vcc – выводы подачи напряжения питания.

Микроконтроллеры семейства 8051 являются микропроцессорными устройствами с архитектурой CISC со стандартным набором команд, характерных для данной архитектуры. Система команд 8051-совместимых устройств включает 111 основных команд размером от одного до трех байт, но большая часть этих команд – одно- или двухбайтовая. Почти все команды выполняются за один или два машинных цикла, что по времени приблизительно равно 1–2 мкс при тактовой частоте 12 МГц, за исключением команд умножения и деления, которые требуют для выполнения четыре машинных цикла.

Команды микроконтроллеров i8051 используют прямую, непосредственную, косвенную и неявную адресацию данных (рисунок 2). При этом в качестве операндов команд могут выступать отдельные биты, четырехбитовые комбинации (тетрады), байты и слова из двух байт.

CLR	C	clear bit to zero	1	12
CLR	bit		2	12
SETB	C	set bit to one	1	12
SETB	bit		2	12
CPL	C	complement bit	1	12
CPL	bit		2	12
ANL	C,bit	AND bit with C	2	24
ANL	C,/bit	...NOTbit with C	2	24
ORL	C,bit	OR bit with C	2	24
ORL	C,/bit	...NOTbit with C	2	24
MOV	C,bit	move bit to bit	2	12
MOV	bit,C		2	24
JC	rel	jump if C set	2	24
JNC	rel	jmp if C not set	2	24
JB	bit,rel	jump if bit set	3	24
JNB	bit,rel	jmp if bit not set	3	24
JBC	bit, rel	jmp&clear if set	3	24

Рис. 2 – Таблица команд.

В данной лабораторной работе используются следующие команды:

– пересылки данных;
– логических операций;
– операций над битами.

Команда MOV выполняет пересылку данных из второго операнда в первый. Эта команда не работает с данными, находящимися во внешней памяти данных или в памяти программ. Для работы с данными, находящимися во внешней памяти данных, предназначены команды MOVX, а для работы с константами, записанными в память программ, – команда MOVC. Первая из них обеспечивает чтение/запись байтов из внешней памяти данных, вторая – чтение байтов из памяти программ.

Команды логических операций манипулируют байтами и позволяют выполнить следующие операции:

– логическое И (ANL);
– логическое ИЛИ (ORL);
– исключающее ИЛИ (XRL);
– инверсию (CPL);
– очистку байта (CLR);
– обычные и циклические сдвиги (RL, RLC, RR, RRC).

Команды операций над битами микроконтроллера 8051 включают 12 команд, позволяющих выполнять операции над отдельными битами: сброс, установку, инверсию, а также «логическое И» (&) и «логическое ИЛИ» (|). В качестве операндов могут выступать 128 бит из внутренней памяти данных микроконтроллера, а также регистры специальных функций, допускающие адресацию отдельных битов.

Последняя группа команд – это группа команд передачи управления микроконтроллера. В группе представлены команды безусловного и условного переходов, команды вызова подпрограмм и команды возврата из подпрограмм.

Выполнение работы

Задание 1

Для вывода значения Q на светодиод потребуется установить токоограничивающий резистор. Вычислим его сопротивление при схеме включения диода, изображенной на рисунке 3.

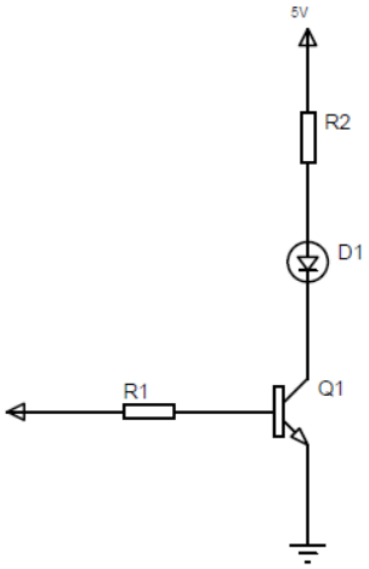


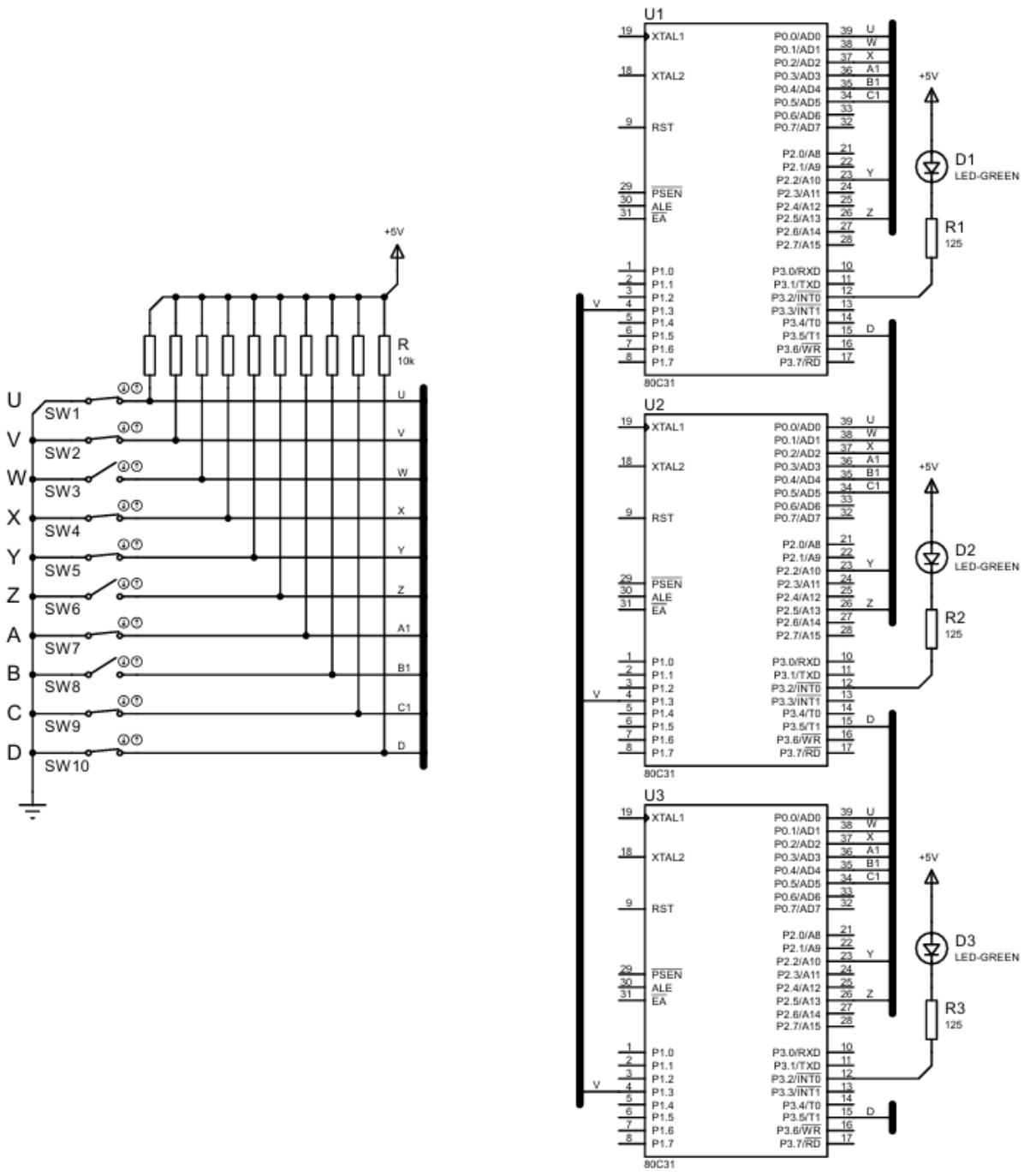
Рис. 3 – Схема включения диода.

Напряжение питания $U = 5$ В, напряжение на транзисторе $U_т = 0.2$ В, напряжение на диоде $U_д = 2.3$ В, силу тока примем равной $I = 0.02$ А. Значение сопротивления равно $R = \frac{U - U_д - U_т}{I} = \frac{(5 - 0.2 - 2.3)\text{ В}}{0.02\text{ А}} = 125$ Ом.

Q	U	V	W	X	Y	Z	A	B	C	D
P3.2	20H.3	P1.3	22H.0	28H.2	P2.2	P2.5	28H.5	21H.4	25H.0	P3.5

Q=(WV+X*/Z)+(A+C)(U+B)

СХЕМА



Диод загорается, когда на входе 0. Чтобы диод загорался в программах считается /Q. Поэтому когда Q=1, диод будет гореть.

Задание 2 – Битовые логические инструкции.

Суть метода: используя логические выражения, вычисляется функция по действиям. В этом методе используется бит переноса C.

```
; $NOMOD51
; $INCLUDE (8051.MCU)

Q BIT P3.2
Uq BIT 20H.3
Vq BIT P1.3
Wq BIT 22H.0
Xq BIT 28H.2
Yq BIT P2.2
Zq BIT P2.5
Aq BIT 28H.5
Bq BIT 21H.4
Cq BIT 25H.0
Dq BIT P3.5

U_ BIT P0.0
W_ BIT P0.1
X_ BIT P0.2
A_ BIT P0.3
B_ BIT P0.4
C_ BIT P0.5

F1 BIT 20H.0

org 0h

MAIN:
   MOV C, Vq
   ANL C, W_
   MOV F0, C

   MOV C, X_
   ANL C, /Zq
   ORL C, F0
   MOV F0, C

   MOV C, A_
   ORL C, C_
   MOV F1, C

   MOV C, U_
   ORL C, B_
   ANL C, F1

   ORL C, F0
   MOV Q, C
   CPL Q
END
```

Задание 3 – Тест битов.

Суть метода: строится блок-схема по данной функции, которая показывает переходы исходя из равенства переменной нулю или единице. По нарисованной блок-схеме пишется код с условными переходами. JNB – переход при условии, что переменная равна 0. JB – переход при условии, что переменная равна 1.

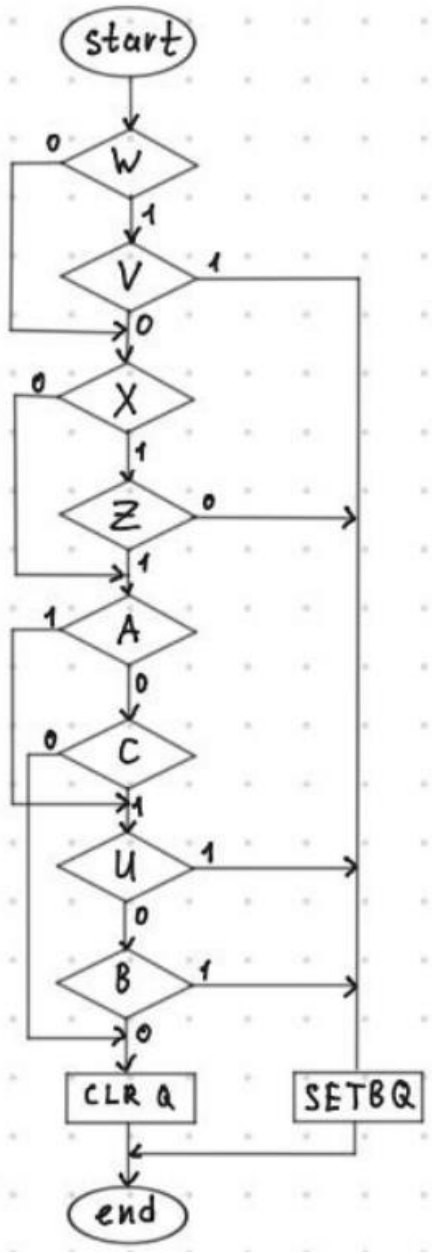


Рис. 4 – Блок-схема.

```
; $NOMOD51
; $INCLUDE (8051.MCU)

Q BIT P3.2
Uq BIT 20H.3
Vq BIT P1.3
Wq BIT 22H.0
Xq BIT 28H.2
Yq BIT P2.2
Zq BIT P2.5
Aq BIT 28H.5
Bq BIT 21H.4
Cq BIT 25H.0
Dq BIT P3.5

U_ BIT P0.0
W_ BIT P0.1
X_ BIT P0.2
A_ BIT P0.3
B_ BIT P0.4
C_ BIT P0.5

org 0h

MAIN:
   JNB W_, TEST_X ;W=0
   JB W_, TEST_V  ;W=1
   TEST_V: JNB Vq, TEST_X
      JB Vq, SET_Q
   TEST_X: JNB X_, TEST_A
      JB X_, TEST_Z
   TEST_Z: JNB Zq, SET_Q
      JB Zq, TEST_A
   TEST_A: JNB A_, TEST_C
      JB A_, TEST_U
   TEST_C: JNB C_, CLR_Q
      JB C_, TEST_U
   TEST_U: JNB U_, TEST_B
      JB U_, SET_Q
   TEST_B: JNB B_, CLR_Q
      JB B_, SET_Q
   SET_Q: ; Установить Q в 1 (но делаем обратное, т.к. при 0 включается диод)
      CLR Q
      JMP MAIN
   CLR_Q:
      SETB Q
      JMP MAIN
END
```

Задание 4 – Байтовые логические инструкции.

Суть метода: используется байтовые команды микроконтроллера. Для написания кода также следует ориентироваться по блок-схеме. Чтобы проверить состояние нужного бита, байт загружается в аккумулятор A, и с помощью операции ANL A, #mask (логическое И с маской) выделяется нужный бит. Затем проверяется состояние аккумулятора (флаг Z в PSW) командами условного перехода JZ/JNZ. По сути, это аналог второго метода.

```
; $NOMOD51
; $INCLUDE (8051.MCU)

Q BIT P3.2
Uq BIT 20H.3
Vq BIT P1.3
Wq BIT 22H.0
Xq BIT 28H.2
Yq BIT P2.2
Zq BIT P2.5
Aq BIT 28H.5
Bq BIT 21H.4
Cq BIT 25H.0
Dq BIT P3.5

U_ BIT P0.0
W_ BIT P0.1
X_ BIT P0.2
A_ BIT P0.3
B_ BIT P0.4
C_ BIT P0.5

org 0h

MAIN:
   MOV A, P0 ;для W
   ANL A, #00000010B
   JZ TEST_X
   JNZ TEST_V

   TEST_V: MOV A, P1
   ANL A, #00001000B
   JZ TEST_X
   JNZ SET_Q

   TEST_X: MOV A, P0
   ANL A, #00000100B
   JZ TEST_A
   JNZ TEST_Z

   TEST_Z: MOV A, P2
   ANL A, #00100000B
   JZ SET_Q
   JNZ TEST_A
```

```
   TEST_A: MOV A, P0
   ANL A, #00001000B
   JZ TEST_C
   JNZ TEST_U

   TEST_C: MOV A, P0
   ANL A, #00100000B
   JZ CLR_Q
   JNZ TEST_U

   TEST_U: MOV A, P0
   ANL A, #00000001B
   JZ TEST_B
   JNZ SET_Q

   TEST_B: MOV A, P0
   ANL A, #00010000B
   JZ CLR_Q
   JNZ SET_Q

   SET_Q: ; Установить Q в 1 (но делаем обратное, т.к. при 0 включается диод)
      CLR Q
      JMP MAIN
   CLR_Q:
      SETB Q
      JMP MAIN
END
```