|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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) |

**Содержание**

[Цель работы 3](#_Toc208567293)

[Теоретическая часть 3](#_Toc208567294)

[Выполнение работы 6](#_Toc208567295)

[Задание 1 6](#_Toc208567296)

[Задание 2 – Битовые логические инструкции. 8](#_Toc208567297)

[Задание 3 – Тест битов. 9](#_Toc208567298)

[Задание 4 – Байтовые логические инструкции. 11](#_Toc208567299)

# 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:

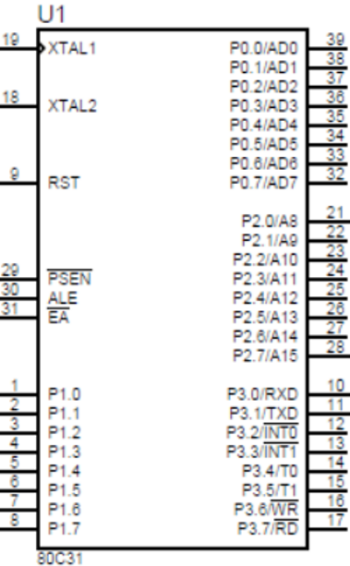


Рис. 1 – Схема микроконтроллера i8051 в Proteus.

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.

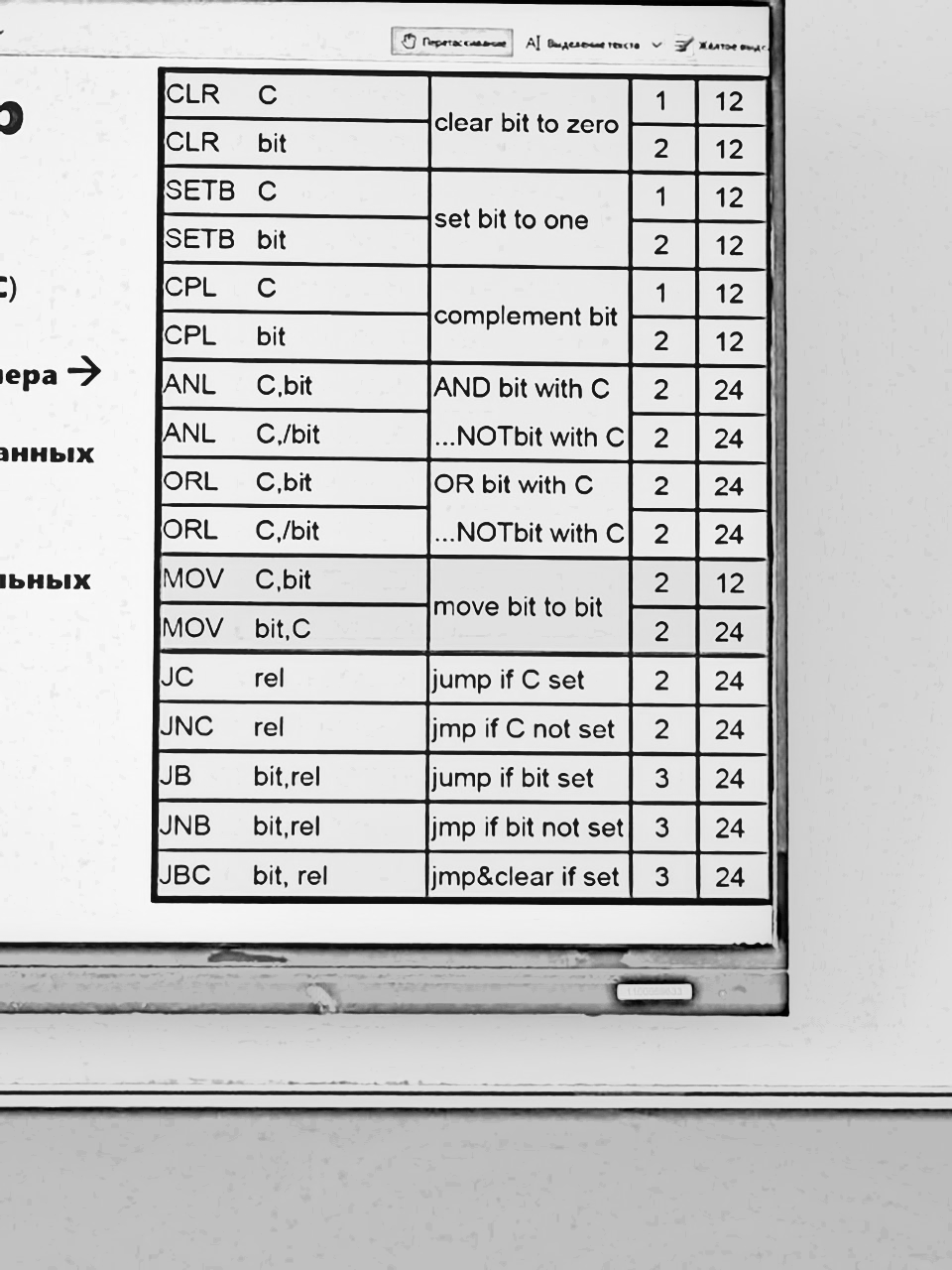


Рис. 2 – Таблица команд.

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.

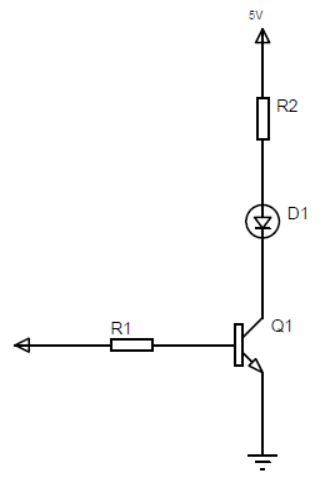


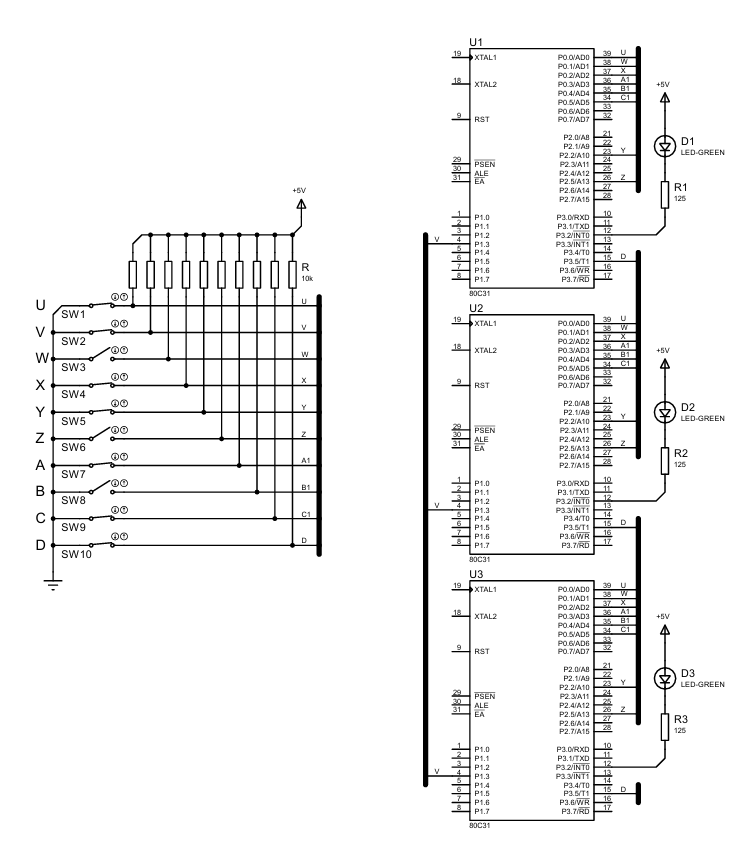
Рис. 3 – Схема включения диода.

Напряжение питания , напряжение на транзисторе , напряжение на диоде , силу тока примем равной . Значение сопротивления равно .

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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.

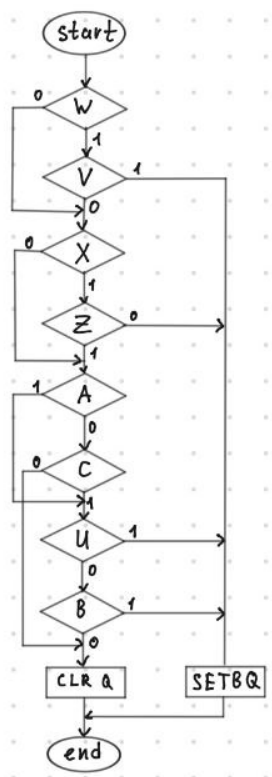


Рис. 4 – Блок-схема.

; $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