

Министерство образования и науки Российской Федерации  
**Федеральное государственное бюджетное образовательное  
учреждение  
высшего образования**

**ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ  
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)**

*Кафедра промышленной электроники (ПрЭ)*

УТВЕРЖДАЮ  
Зав. кафедрой ПрЭ  
С.Г. Михальченко

## **ЛАБОРАТОРНЫЙ ПРАКТИКУМ ПО ЦИФРОВОЙ И МИКРОПРОЦЕССОРНОЙ ТЕХНИКЕ**

Руководство к выполнению лабораторных работ  
для студентов направления подготовки 11.03.04  
"Электроника и нанoeлектроника"  
заочного и вечернего факультета

Разработчик:  
доцент каф. ПрЭ  
А.И. Воронин

# Лабораторная работа №1

## Синтез комбинационных устройств на логических элементах

### 1. ЦЕЛЬ РАБОТЫ

Целью лабораторной работы является приобретение навыков логического проектирования комбинационных цифровых устройств и практическая реализация таких устройств на логических элементах с помощью программной среды моделирования ASIMES, экспериментальная проверка правильности их функционирования. Преподаватель назначает вариант задания (от 1 до 12), выполняемого каждой рабочей подгруппой.

### 2. БАЗОВЫЕ СХЕМЫ ЭЛЕМЕНТА ТТЛ

Простейший логический элемент (рис.1) строится на базе многоэмиттерного транзистора  $V_{Tm}$ , выполняющего функцию И для сигналов, подаваемых на его эмиттеры, и транзисторного ключа ( $V_{T1}$ ), выполняющего функцию НЕ. Если на всех входах высокие уровни напряжения ( $A=1$  и  $B=1$ ), закрыты эмиттерные переходы  $V_{Tm}$ , открывается переход база-коллектор этого транзистора и ток  $I$  открывает и насыщает ключевой транзистор  $V_{T1}$ , формируя на выходе низкий уровень напряжения ( $F=0$ ). Если хотя бы на одном из входов низкий уровень напряжения, ток  $I$  переключается в выходную цепь источника сигнала, закрывая  $V_{T1}$ . При этом  $F=1$ . Таким образом, схема реализует таблицу истинности логического элемента 2И-НЕ.

Для повышения быстродействия выходной ключ выполняют по схеме сложного инвертора (рис.2). При  $A=B=1$  открыты транзисторы  $V_{T2}, V_{T1}$  (транзистор  $V_{T3}$  закрыт) и емкость нагрузки быстро разряжается через сопротивление открытого ключа  $V_{T1}$ . Для любой другой комбинации входных сигналов емкость нагрузки имеет возможность быстро зарядиться до высокого уровня напряжения через низкое выходное сопротивление эмиттерного повторителя  $V_{T3}$  ( $V_{T2}$  и  $V_{T1}$  закрыты).

Элементы ТТЛШ отличаются от ТТЛ тем, что коллекторные переходы транзисторов зашунтированы диодами Шоттки. Транзисторы элементов ТТЛШ не входят в насыщение, что существенно уменьшает задержку выключения транзисторов. К тому же они меньших размеров, что уменьшает емкости р-п-переходов и потребляемую мощность.

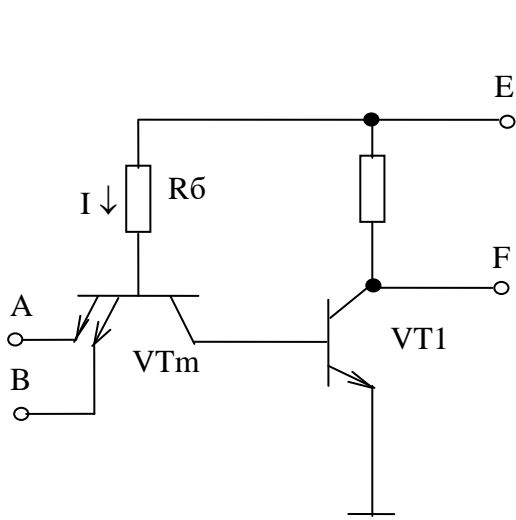


Рисунок 1

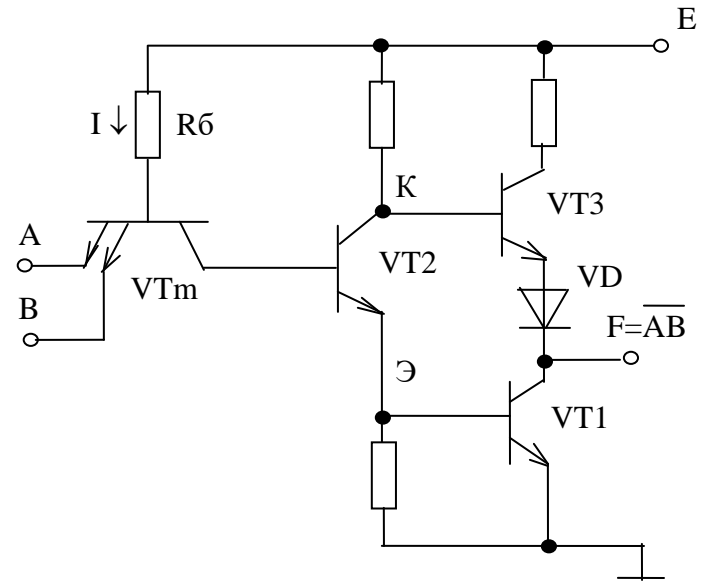


Рисунок 2 - Базовая схема элемента ТТЛ

В элементе с открытым коллекторным выходом (рис.3) VT3 и VD отсутствуют. Вместо них подключают элементы автоматики (обмотки реле) или индикации (например, светодиод). Такие элементы допускают объединение выходов. Пример применения логического элемента с открытым коллекторным выходом (микросхема К155ЛА8) показан на рис.4.

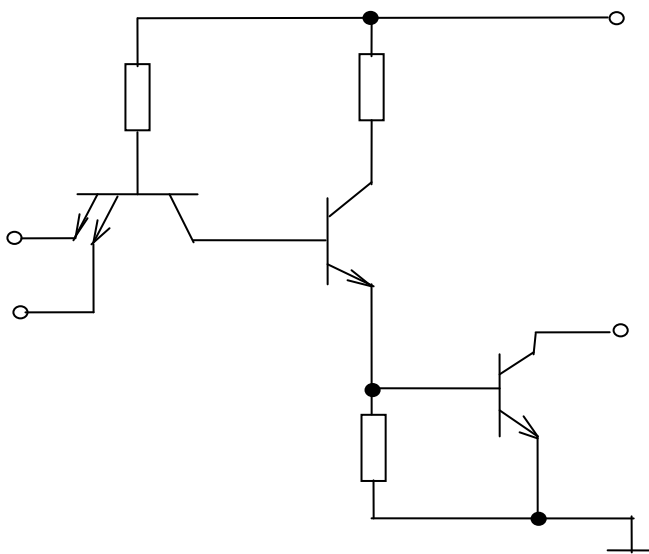


Рисунок 3

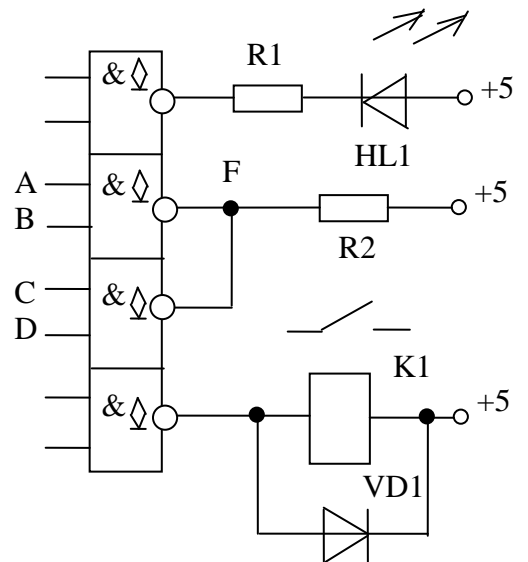


Рисунок 4

На выходе F реализуется логическая функция  $F = \overline{AB + CD}$ , т.е. логический ноль наблюдается при совпадении логических единиц на входах

А и В или С и D. Таким образом, объединение выходов позволяет путем монтажа выполнить логическую операцию ИЛИ (монтажное ИЛИ, монтажное И).

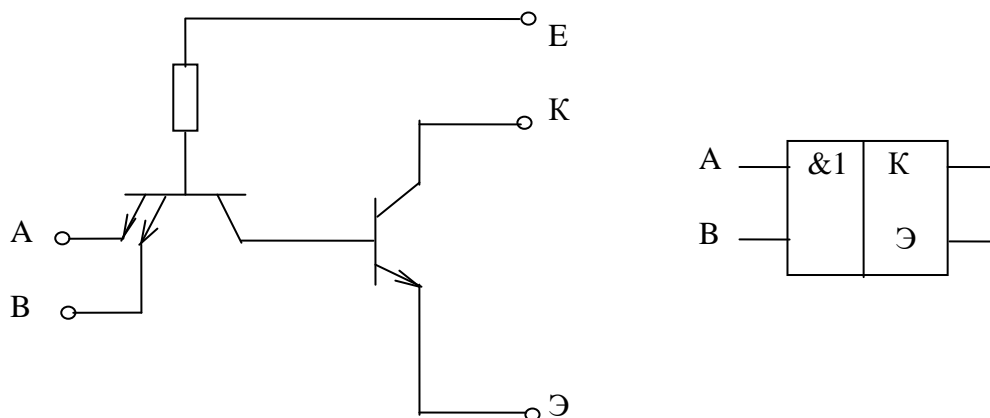


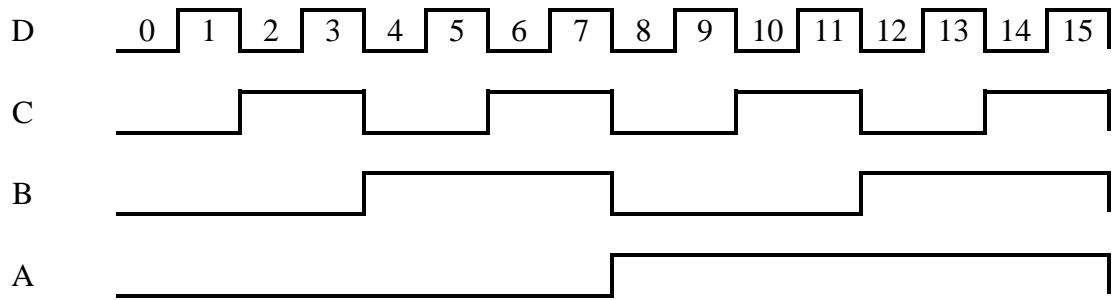
Рисунок 5 - Логический расширитель

В схеме логического расширителя (рис.5) задействованы лишь элементы  $R_b$ ,  $VT_1$  и  $VT_2$  базовой схемы. Логические расширители (например, К155ЛД1) используются совместно с другими логическими элементами, имеющими соответствующие входы К (открытый коллектор) и Э (открытый эмиттер), например, К155ЛР3 или К155ЛР1.

### 3. ВАРИАНТЫ ЗАДАНИЙ

Лабораторная работа включает проектирование генератора импульсов сложной формы по заданной временной диаграмме выходного сигнала в течение периода (рис.6). Такой генератор, включает в себя комбинационную схему, которую необходимо синтезировать и экспериментально проверить.

Значения булевой функции  $F$ , реализуемой комбинационной схемой на 16 наборах входных сигналов А, В, С, D (выходы двоичного счетчика), соответствуют уровням выходного напряжения генератора на 16 тактах периода генерируемых импульсов. Таким образом, для каждого варианта задана таблица истинности логической функции четырех переменных  $F$ .



### Варианты заданий

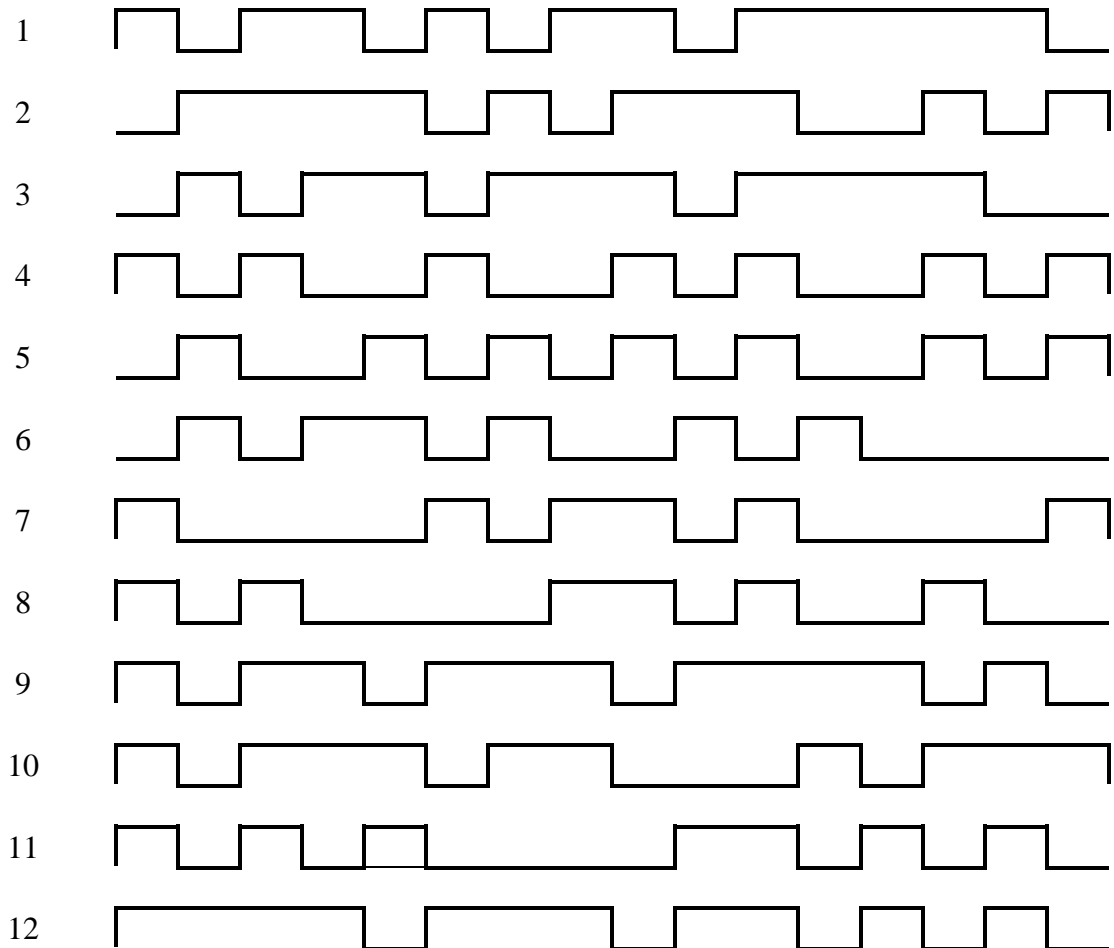


Рисунок 6 - Варианты заданий

При проектировании комбинационного цифрового устройства на логических элементах для минимизации булевой функции удобно воспользоваться картой Карно. Пример карты Карно приведен на рис.7, в каждую клетку карты Карно записывается значение булевой функции на соответствующем тактовом интервалом работы комбинационного устройства, рис. 8.

		A				
		1	0	0	0	
		1	1	0	1	D
		1	1	0	0	
C		1	1	1	1	
		B				

Рисунок 7 - Карта Карно

0	4	12	8
1	5	13	9
3	7	15	11
2	6	14	10

Рисунок 8

Правила записи минимизированного выражения для логической функции по карте Карно:

- 1) выделяются блоки, заполненные единицами;
- 2) блок должен быть прямоугольным и содержать 1, 2, 4, 8 клеток;
- 3) блоки должны быть возможно большими, а их количество наименьшим;
- 4) левая и правая, а также верхняя и нижняя строки карты считаются соседними;
- 5) одна и та же клетка может входить в несколько блоков;
- 6) функция может доопределяться произвольно (на тех наборах, где стоят X), чтобы получить наиболее крупные блоки;
- 7) функция записывается в виде логических произведений (ЛП), описывающих выделенные блоки;
- 8) переменная не включается в ЛП, если блок областью ее прямых значений делится пополам;
- 9) переменная включается в ЛП с инверсией, если рассматриваемый блок лежит в области ее инверсных значений;
- 10) при группировке в блоки клеток, заполненных нулями, по тем же правилам получаем инверсное значение логической функции.

## 4. ПРОГРАММА РАБОТЫ

4.1 Провести синтез комбинационного цифрового устройства для заданного варианта на логических элементах. Провести моделирование полученной схемы в программной среде ASIMES. Снять временные диаграммы на выходе устройства. Сравнить с заданной временной диаграммой.

4.2 Подготовить отчет, включив в него результаты проектирования, временные диаграммы, ответы на контрольные вопросы.

## 5. КОНТРОЛЬНЫЕ ВОПРОСЫ

- 5.1. Какие способы применяются для минимизации логических функций?
- 5.2. Запишите выражение для логической функции, связывающей выходной и входные сигналы мультиплексора.
- 5.3 Запишите минимизированное выражение для булевой функции по карте Карно, представленной на рис.7, организовав блоки по единицам.
- 5.4 Чем отличаются комбинационные цифровые устройства от последовательностных? Приведите примеры микросхем цифровых устройств комбинационного типа.
- 5.3. Сравните элементы ТТЛ, ТТЛШ и КМОП по быстродействию и экономичности.

## Лабораторная работа №2

### Программная модель и система команд МК51

#### 1 Цель работы

Целью лабораторной работы является приобретение навыков программирования на языке Ассемблер и использования интегрированной среды разработки MSU 8051 IDE (<http://www.moravia-microsystems.com/mcu-8051-ide>, далее среда разработчика) при отладке прикладных программ для однокристальных микроконтроллеров семейства MCS-51.

#### 2 Запись программы на языке ассемблера и ее трансляция

Язык ассемблера допускает представление всех элементов программы в символической (буквенно-цифровой) форме, отражающей их содержательный смысл. В качестве алфавита допустимых символов принят код ASCII (американский стандартный код для обмена информацией). Каждая строка ассемблера соответствует одной команде или псевдокоманде (директиве) и может содержать поля метки, мнемоники команды, операнда и комментария. При наличии в программе синтаксических ошибок ассемблер в процессе трансляции выдает сообщения об ошибках.

Метка ассоциируется с 16-битовым адресом той ячейки памяти, где будет размещен первый байт отмеченной команды. Использование меток освобождает программиста от необходимости оперировать абсолютными адресами памяти при записи команд передачи управления. Метка должна начинаться с буквы и заканчиваться двоеточием. Не допускается использовать в качестве меток мнемокоды команд, обозначения операндов и директив ассемблера. Символическое обозначение может появиться в поле метки только один раз.

Мнемокод команды может включать до четырех символов и вместе с обозначениями операндов образует группу ключевых слов ассемблера.

Поле операнда содержит числовые и символьные непосредственные данные, обозначения регистров и регистровых пар МК, адреса памяти. Возможно использование выражений, содержащих простейшие арифметические и логические операции (см. Таб 2.1). Операнд в виде строки символов, заключенной в апострофы, транслируется в последовательность кодов ASCII этих символов.

Поле комментария начинается с точки с запятой и полностью игнорируется ассемблером. В поле комментария фиксируется обычно функция, которую выполняет группа команд в конкретной прикладной программе.

Кроме команд программа может содержать директивы ассемблера (или псевдокоманды). Директивы выполняются компилятором и дают программисту дополнительный сервис при компоновке прикладных



программ. Список наиболее употребляемых директив приведен в Табл. 2.2. Полный список директив приведен в среде разработчика **Главное меню/Help/Handbook**.

Таблица 2.1

Допустимые выражения Ассемблера

Оператор	Выражение	Пример
<b>Унарные операторы</b>		
NOT	Отрицание (инверсия)	NOT 0A5H
LOW	Выделение младшего байта	LOW 0A5F5H
HIGH	Выделение старшего байта	HIGH 0A5F5H
<b>Бинарные операторы</b>		
+	Сложение	11+12
-	Вычитание	12-11
*	Умножение	11*12
/	Деление	15/3
AND	Логическое И	12 AND 11
OR	Логическое ИЛИ	12 OR 11
XOR	Исключающее ИЛИ	12 XOR 11
SHL	Логический сдвиг влево	12 SHL 2
SHR	Логический сдвиг Вправо	12 SHR 2

Таблица 2.2

Директивы Ассемблера

Директива	Наименование	Синтаксис	Пример
ORG	Начальный адрес массива	ORG <адрес>	ORG 2BH
END	Прекращение трансляции	END	END
DB	Определить байт	DB <данные>, <данные> ....	DB 23,'A', 23H, (22+15)
DW	Определить слово (два байта)	DW <данные>, <данные> ....	DW 2215, 0A22H
EQU	Присвоение, эквивалентность	<символ> EQU выражение>	abc EQU R0 xyz EQU 4EH+12
SET	Определение переменной или регистра	<имя> SET <значение или регистр>	abc SET R1 xyz SET 2AH*2
BIT	Определение прямо адресуемого бита	<имя> BIT <значение>	abc BIT P4.2

В таблице 2.3 приведены правила записи констант в допустимых системах счисления. Строковые константы в памяти микроконтроллера хранятся в виде байта ASCII. Таблицу можно найти в среде разработчика **Главное меню/Utilites/ASCII Charts**. Например, при записи abc SET 'C' константе abc будет присвоено значение 43H.

Таблица 2.3

Система счисления	Допустимые символы	Пример записи
Двоичная	0,1	00110011b
Восьмеричная	0,1,2,3,4,5,6,7	321q
Десятичная	0,1,2,3,4,5,6,7,8,9	123
Шестнадцатеричная	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F	02Fh

### 3 Программа работы

1. Создать проект, имя проекта должно быть уникальным, набрано латинским шрифтом. Рекомендуется все программы по циклу лабораторных работ записывать в одном проекте. Определив номер варианта (от N=1 до N=12), создать программу TEST.ASM:

; Программа тестирования ассемблера

```
VAR EQU N
X EQU (512/2+(VAR))
Y SET LOW(X)
Z SET HIGH(X)
ABC SET 'A'
    ORG 0H
    MOV R0, #Y
    MOV R1, #Z
    CALL CHAR
    MOV R2, A
    JMP $
CHAR: MOV A, #(ABC+1)
    ADD A, #VAR
    RET
    END
```

Написать комментарий к каждой строке программы. До выполнения программы зафиксировать значение указателя стека. Объяснить содержимое R0, R1, R2 после выполнения программы. С какого адреса располагается стек? Что находится в вершине стека? Как изменяется содержимое указателя стека?

2. Проверить работу программы IND.asm. Программа считает количество нажатий кнопки в двоично-десятичном коде подключенной к P3.2 и выводит на двухразрядный семисегментный индикатор число нажатий кнопки.

```
org 0
mov dptr,#0050h
mov a, #0h
m1: jb p3.2, $
jnb p3.2, $
inc a
da a
call simvol
```

```

jmp m1
simvol: push acc
mov b, a
anl a, #0fh
movc a, @a+dptr
mov p1, a
mov a, b
swap a
anl a, #0fh
movc a, @a+dptr
mov p2, a
pop acc
ret
org 50h
db 11000000b ; 0
db 11111001b ; 1
db 10100100b ; 2
db 10110000b ; 3
db 10011001b ; 4
db 10010010b ; 5
db 10000010b ; 6
db 11111000b ; 7
db 10000000b ; 8
db 10010000b ; 9
end

```

Семисегментные индикаторы подключить к портам P1 (единицы), P2 (десятки). Зайдите в настройки индикатора, установите схему включения индикатора с общим анодом (Common electrode/Common anode), установите флажок "Поверх всех окон" (Window always on top).

Написать комментарий к программе. Каким образом в программе определяется код выводимого символа на семисегментный индикатор?

Заблокировать команду da a, повторить эксперимент, объяснить результат.

3. Разработать и отладить на эмуляторе индивидуальное задание в соответствии со своим вариантом:

3.1 Установка 0 на линии P1.0 циклически изменяет состояние семисегментного индикатора от цифры 0 до цифры 5, фиксация на P1.0 уровня логической единицы останавливает счет.

3.2 Первое нажатие кнопки считывает двоично-десятичный код (числа 0...99) с параллельного порта, второе нажатие – выводит двоичный код на двухразрядный семисегментный индикатор.

3.3 Первое нажатие кнопки считывает двоичный код (числа 0...99) с параллельного порта, второе нажатие – выводит двоично-десятичный код на

двухразрядный семисегментный индикатор.

3.4 На одноразрядном семисегментном индикаторе проимитировать работу двоично-десятичного счетчика с коэффициентом пересчета 7.

3.5 На одноразрядном семисегментном индикаторе проимитировать работу двоичного счетчика с коэффициентом пересчета 12.

3.6 Нажатие кнопки циклически выводит на семисегментный индикатор следующие символы: A, b, C, d, E, F.

3.7 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести сумму тетрад, если результат больше числа 9, на индикатор вывести "0".

3.8 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести разность тетрад, если результат отрицательный, на индикатор вывести символ "-".

3.9 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести произведение тетрад, если результат больше числа 9, на индикатор вывести символ "П".

3.10 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести целую часть от деления тетрад, при делении на 0, на индикатор вывести символ "-".

3.11 Считать с параллельного порта байт данных (двоично-десятичный код), на одноразрядный семисегментный индикатор вывести остаток от деления тетрад, при отсутствии остатка, на индикатор вывести символ "0".

3.12 На одноразрядном семисегментном индикаторе проимитировать работу вычитающего двоично-десятичного счетчика с коэффициентом пересчета 7.

#### **4 Контрольные вопросы**

1. Чему равно содержимое регистров SFR MK51 после системного сброса?
2. Чем отличаются друг от друга команды MOV R5,7 и MOV 5,#7?
3. Как выполняется команда MUL AB?
4. С помощью каких команд можно прочитать в регистр В информацию с датчиков, подключенных к линиям порта P1?

#### **5 Содержание отчета**

Отчет по лабораторной работе должен содержать: пункты Программы работы, исходные тексты программ, ответы на вопросы.