3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ
по выполнению домашней контрольной работы
Общие указания

Каждое контрольное задание содержит пять задач. Перед решением каждой задачи изучите рекомендуемый материал и методические указания по соответствующим темам программы.

При оформлении работы необходимо придерживаться следующих требований:

1. Работа выполняется в ученической тетради. Количество страниц должно быть достаточным для размещения всех заданий с решением, для рецензии преподавателя и последующей работы над ошибками.
2. Вначале следует переписать условие задачи с данными своего варианта.
3. Ответы надо приводить сразу же после каждого пункта условия задачи.
4. Условные графические обозначения элементов и узлов приводить в соответствии с требованиями ГОСТ 2.743-91. Краткие сведения из этого стандарта приведены в приложении В.
5. Страницы, рисунки, таблицы должны быть пронумерованы. Рисунки и таблицы должны быть подписаны (ГОСТ 2. 105-95).
6. Ответы должны быть конкретными, краткими, четкими.
7. Работу над ошибками выполняйте не в тексте контрольной работы, а после рецензии преподавателя.
8. Если работа не зачтена, то ее следует выполнить вновь и отправить на проверку вместе с первой.

Без представленной зачтенной контрольной работы и при отсутствии зачтенной курсовой работы студенты к сдаче экзамена не допускаются.

Задача №1

Таблица 8

№ варианта	Логическая функция	Набор аргументов для проверки		
		x_1	x_2	x_3
1	$x_1x_2 \vee \overline{x_1}x_3 \vee x_1\overline{x_3}$	1	1	0
2	$x_1x_2 \vee x_1x_3 \vee x_2x_3 \vee \overline{x_1}x_2\overline{x_3}$	1	1	1
3	$x_1\overline{x_3} \vee x_1\overline{x_2} \vee \overline{x_2x_3} \vee \overline{x_1}x_2x_3$	0	0	1

№ варианта	Логическая функция	Набор аргументов для проверки		
4	$\overline{x_2} \vee \overline{x_1} x_3 \vee x_1 \overline{x_3}$	0	1	1
5	$x_3 \vee \overline{x_1 x_2} \vee x_1 x_2$	0	0	0
6	$\overline{x_1} \vee x_2 x_3 \vee \overline{x_2 x_3}$	1	0	1
7	$x_1 x_2 \vee x_1 \overline{x_3} \vee x_2 \overline{x_3} \vee \overline{x_1 x_2} x_3$	1	0	0
8	$x_1 \vee \overline{x_3} x_2 \vee x_3 \overline{x_2}$	0	1	1
9	$x_2 \vee \overline{x_1} x_3 \vee x_1 \overline{x_3}$	0	0	1
10	$x_1 x_3 \vee x_1 \overline{x_2} \vee \overline{x_2} x_3 \vee \overline{x_1} x_2 \overline{x_3}$	0	1	0

Задана логическая функция f(x1,x2,x3), таблица 8.

1. Построить схему в базисе И, ИЛИ, НЕ.

2. Построить эту схему в базисе И, ИЛИ, НЕ на микросхемах серии К155 (КР1533).

3. Выполнить преобразование заданной логической функции так, чтобы она была представлена через операцию И-НЕ.

4. Построить логическую схему в базисе И-НЕ на микросхемах серии КР1533 (К155).

5. На всех построенных схемах указать логические сигналы на входах и выходах каждого элемента для кодовой комбинации, заданной таблицей 9.

6. Определить количество микросхем, используемых для построения схем в п.2 и п.4. Сделать вывод о том, какой способ реализации более экономичен.

Методические указания по выполнению задачи №1

В ходе решения этой задачи предполагается, что студент, изучив предварительно соответствующий теоретический материал ([1], [2], стр 34...38, 51...59, 63...71), приобретает навыки построения логических схем в базисах И, ИЛИ, НЕ и И-НЕ, а также в выполнении преобразований логических выражений с использование законов алгебры логики (правило де Моргана).

Порядок решения:

- выписать из табл. 8 функцию, соответствующую вашему варианту;
- в результате анализа определить какие логические элементы и в каком количестве потребуются для построения схемы.

При построении логической схемы следует учитывать приоритет выполнения операций:

- операция НЕ (инверсия);
- операция И (конъюнкция);
- операция ИЛИ (дизъюнкция).

При построении логических схем с учетом конкретной серии необходимо иметь ввиду, что в их состав входят микросхемы с определенным числом входов. Элементы И и ИЛИ обычно на два входа (2И, 2ИЛИ), элементы И-НЕ на 2, 3, 4, 8 входов (соответственно: 2И-НЕ, 3И-НЕ, 4И-НЕ, 8И-НЕ), элементы ИЛИ-НЕ – на два входа (2ИЛИ-НЕ).

Пример. Задана логическая функция (10):

$$f = \overline{x_1}x_3 \vee x_1\overline{x_2} \vee \overline{x_2}x_3 \vee x_1x_2\overline{x_3} \qquad (10)$$

Для построения схемы по приведенному выражению потребуется :

- три инвертора (все три аргумента входят в запись в инверсном и прямом значении);
- три элемента 2И (для реализации выражений $\overline{x_1}x_3$, $x_1\overline{x_2}$, $\overline{x_2}x_3$;
- один элемент 3И (для реализации выражения $x_1x_2\overline{x_3}$;
- один элемент 4ИЛИ для объединения предварительных результатов преобразования на одну общую шину.

Схема имеет вид – рисунок 8

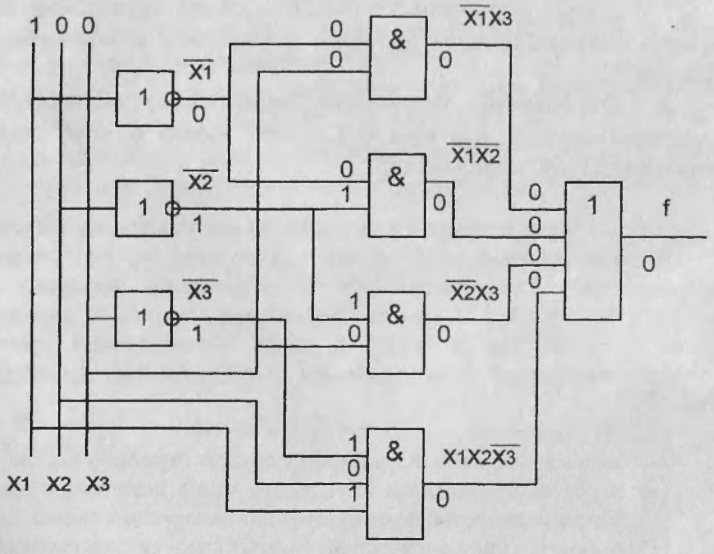


Рис.8. Схема в базисе И, ИЛИ, НЕ

Следующую схему строим также в базисе И, ИЛИ, НЕ, но уже применительно к заданной серии микросхем. Перед этим необходимо внимательно изучить состав серии и правила применения микросхем по справочнику. В серии КР1533 есть микросхемы, которые содержат четыре элемента 2И-НЕ, (КР1533ЛИ1), четыре логических элемента 2ИЛИ-НЕ (КР1533ЛЛ1), шесть логических элементов НЕ (КР1533ЛН1). С целью реализации схемы в базисе И, ИЛИ, НЕ на микросхемах серии К155 (КР(1533) сделаем преобразование исходной функции (10).

$$f = \overline{x_1}x_3 \vee \overline{x_1}x_2 \vee \overline{x_2}x_3 \vee x_1x_2\overline{x_3} = (\overline{x_1}x_3 \vee \overline{x_1}x_2) \vee (\overline{x_2}x_3 \vee (x_1x_2)\overline{x_3}) \qquad (11)$$

Из этой записи следует, что для построения схемы потребуется три инвертора, пять элементов 2И, три элемента 2ИЛИ. Схема приведена на рис. 9.

В этой схеме использованы следующие микросхемы:

- D1 - КР1533ЛН1 (из шести элементов задействовано три);
- D2, D3, - КР1533ЛИ1 (один корпус используется полностью, а во втором только один);
- D4 – КР1533ЛЛ1 (один корпус, из четырех элементов используется три).

Для построения схемы в базисе И-НЕ необходимо выполнить преобразование исходного выражения так, чтобы оно было записано через операцию Штрих Шеффера. Для этого к исходному выражению применяем закон двойного отрицания и правило де Моргана.

$$f = x_1x_3 \vee \overline{x_1}x_2 \vee \overline{x_2}x_3 \vee x_1x_2\overline{x_3} = \overline{\overline{x_1x_3 \vee \overline{x_1}x_2 \vee \overline{x_2}x_3 \vee x_1x_2\overline{x_3}}} = \overline{\overline{(x_1x_3)}\,\overline{(\overline{x_1}x_2)}\,\overline{(\overline{x_2}x_3)}\,\overline{(x_1x_2\overline{x_3})}} \qquad (12)$$

Для построения схемы допускается применение микросхем с любым числом входов, которые имеются в данной серии. Предполагается, что инверторы также должны быть выполнены на элементах базиса И-НЕ. Это легко реализуемо, т.к. для этого достаточно соединить все входы элемента И-НЕ вместе (в силу тождества конъюнкции). Схема приведена на рис. 10.

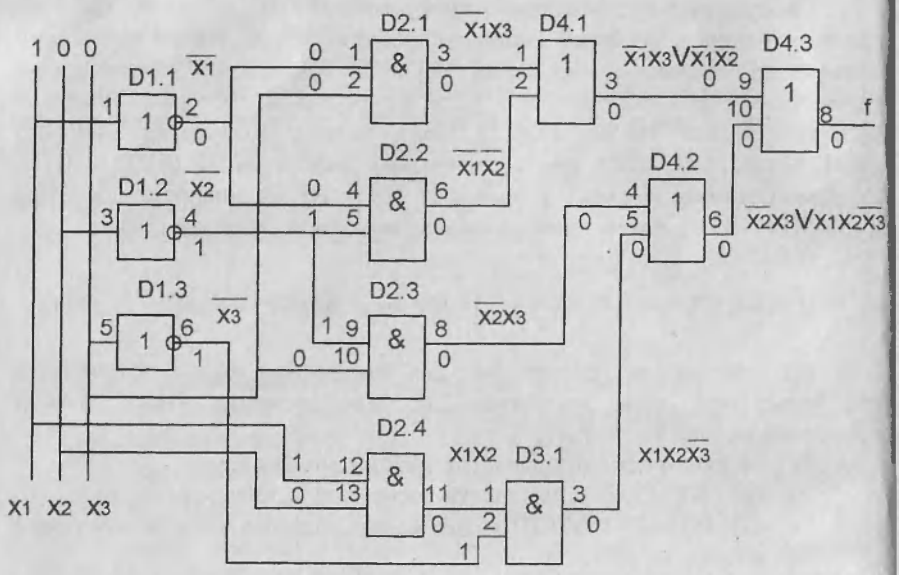


Рис. 9. Схема в базисе И, ИЛИ, НЕ на элементах серии КР1533

Схема (рис.10) построена на трех микросхемах:

- две микросхемы КР1533ЛА3 (в каждой из четырех элементов используется по три). Для реализации инверторов входы логического элемента соединены вместе и на них подается значение одного аргумента;
- один логический элемент КР1533ЛА1.(в одном корпусе два элемента). Один элемент включен по схеме 3И-НЕ. Это позволило исключить применение еще одного корпуса, например, микросхему КР1533ЛА4.

Для проверки на входы всех трех построенных схем подана одна и та же кодовая комбинация сигналов 100. Результат на выходе всех схем получился одинаковый, что необходимо контролировать при выполнении контрольной работы.

Из приведенных схем можно сделать вывод, что построение схем в базисе И-НЕ более экономично. В этом случае применяется меньше микросхем и они более эффективно используются. Аналогично строятся схемы на ИМС серии К155.

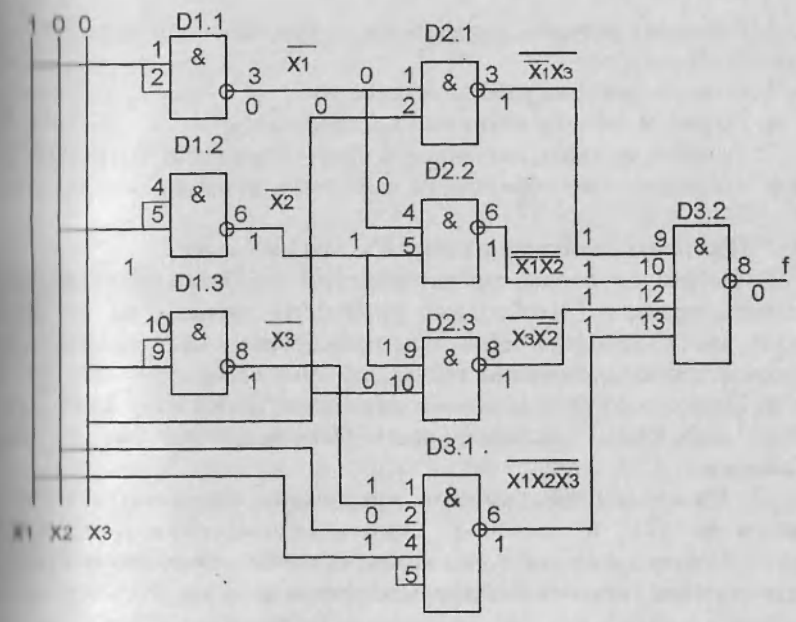


Рис. 10. Схема в базисе И-НЕ на микросхемах серии КР1533

Задача №2

Таблица 9

№ варианта	Тип КЛУ	Структура КЛУ	Базис для реализации	Тип микросхемы
1	Шифратор	10X4	И-НЕ	К155ИВ1
2	Мультиплексор	8→1	ИЛИ-НЕ	К555КП15
3	Дешифратор	4X10	ИЛИ-НЕ	К155ИД10
4	Демультиплексор	1→8	И-НЕ	К155ИД3
5	Шифратор	16X4	И-НЕ	К555ИВ2
6	Мультиплексор	1→8 8→1	И-НЕ	К555КП7
7	Дешифратор	4X10	И, ИЛИ, НЕ	К555ИД7
8	Демультиплексор	8→1	ИЛИ-НЕ	К555ИД5
9	Шифратор	10X4	ИЛИ-НЕ	К555ИВ3
10	Дешифратор	3X8	И-НЕ	К155ИД4

1. Дать определение КЛУ, заданного в табл. 9.

2.Привести условное графическое обозначение устройства с указанной структурой.

3 .Описать принцип работы устройства.

4. Привести таблицу истинности устройства.

5. Записать функции для выходов через операции И, ИЛИ, НЕ.

6. Выполнить преобразование исходных функций под заданный базис.

7. Построить логическую схему в заданном базисе.

8. Подать на входы любую кодовую комбинацию сигналов и выполнить проверку. Необходимо проставить сигналы на входах и выходах всех элементов. Для мультиплексора и демультиплексора подавать кодовую комбинацию только адресных сигналов.

9. Привести УГО получившегося устройства. На УГО на входах и выходах проставить сигналы, соответствующие той же кодовой комбинации.

10. Из справочника выбрать микросхему, заданную в табл. 9. Привести ее УГО и описание. Указать назначение всех входов и выходов. Подать на входы ту же кодовую комбинацию, что и в п. 8 и указать значение сигналов на информационных выходах.

Методические указания по выполнению задачи №2

В задаче рассматриваются типовые комбинационные цифровые устройства. Прежде чем приступить к решению этой задачи изучите соответствующий теоретический материал в [1], [2] ст110-117, 123-126. Обратите внимание на теоретический материал данного пособия стр. 19 и 20.

Рассмотрим синтез демультиплексора со структурой 1→ 4.

Исходя из определения и задания на рисунке 11, приведено УГО демультиплексора, а табл. 10 является его таблицей истинности.

Таблица 10

Адрес		Выходы			
A_1	A_0	Q_0	Q_1	Q_2	Q_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

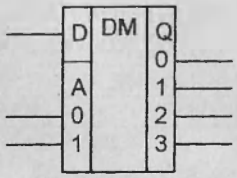


Рис. 11 – УГО демультиплексора

По данным таблицы 10 запишем функции для выходов демультиплексора.

$$Q_0 = D\overline{A_1}\,\overline{A_0} \quad (13)$$

$$Q_1 = D\overline{A_1}A_0 \quad (14)$$

$$Q_2 = DA_1\overline{A_0} \quad (15)$$

$$Q_3 = DA_1A_0 \quad (16)$$

По формулам (13) – (16) строим схему в базисе И, ИЛИ, НЕ (рис. 12).

Если требуется построить схему в базисе И-НЕ, то можно над левой и правой частью функций (13)-(16) поставить знак отрицания и в этом случае получится устройство с инверсными выходами. Можно поставить над правой частью двойное отрицание. В этом случае демультиплексор останется с прямыми выходами, но в схему будут включены дополнительные элементы И-НЕ, выполняющие функции инверторов.

При построении схемы в базисе ИЛИ-НЕ исходные функции (13)-(16) также необходимо предварительно преобразовать. В этом случае к правой части следует применить закон двойного отрицания и правило де Моргана.

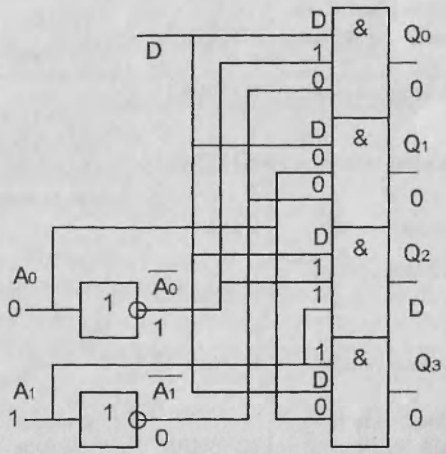


Рис. 12. Схема демультиплексора в базисе И, ИЛИ, НЕ

Для определения сигналов на выходе подставим в формулы (13) - (16) значения переменных.

$$Q_0 = D\overline{A_1 A_0} = D \cdot 0 \cdot 1 = 0$$
$$Q_0 = D\overline{A_1} A_0 = D \cdot 0 \cdot 0 = 0$$
$$Q_0 = DA_1 \overline{A_0} = D \cdot 1 \cdot 1 = D$$
$$Q_0 = DA_1 A_0 = D \cdot 1 \cdot 0 = 0$$

Таким образом информация с входа D передается на выход Q_2. Те же значения подставлены в схему рис. 12.

Задача №3

Таблица 11

№ варианта	Тип устройства	Тип микросхемы	Кодовая комбинация и действие	Начальное состояние	Число входных импульсов
1	Счетчик суммирующий	К155ИЕ6	—	0101	48
2	Регистр параллельный	К555ИР8	10011011 запись	—	—
3	Счетчик вычитающий	К155ИЕ7	—	1101	53
4	Регистр последовательно - параллельный со сдвигом влево	К555ИР9	11010011 вывод	—	—
5	Счетчик суммирующий	К555ИЕ13	—	1010	44
6	Регистр параллельный	К155ИР13	100101110 последовательный ввод со сдвигом вправо	—	—
7	Счетчик вычитающий	К555ИЕ14	—	0110	36
8	Регистр последовательно – параллельный со сдвигом вправо	К555ИР11	1011 сдвиг на четыре разряда влево	—	—
9	Счетчик суммирующий	К555ИЕ10	—	1100	57
10	Регистр последовательно – параллельный со сдвигом вправо	К155ИР1	1101 сдвиг на четыре разряда вправо	—	—

Методические указания по выполнению задачи №3

Порядок решения для вариантов 1, 3, 5, 7, 9:

1. Для заданного типа устройства (таблица 11) привести четырехразрядную логическую схему на счетных триггерах.
2. Построить временную диаграмму для полного цикла счета.
3. Выбрать из справочника заданную микросхему и привести ее [illegible]
4. Объяснить назначение этого счетчика, определить его разрядность (n) и коэффициент счета $Ксч$.
5. Указать назначение всех выводов.
6. Указать на УГО значение всех входных сигналов (в том числе и управляющих), которые надо подать, чтобы зафиксировать поданное на [illegible] двоичное число.
7. Привести расчет состояния счетчика после поступления на вход [illegible] числа импульсов при указанном режиме работы.

Порядок решения для вариантов 2, 4, 6, 8, 10:

1. Привести логическую схему четырехразрядного регистра [illegible] типа (таблица 11) на D-триггерах.
2. Обозначить на схеме буквенными символами все входы и [illegible], укажите их назначение.
3. Выбрать из справочника заданную микросхему. Привести ее [illegible] графическое обозначение.
4. Привести описание этого регистра (выполняемые функции и [illegible]ности, разрядность, порядок управления работой), указать [illegible]начение выводов.
5. На рисунке в п.3 указать сигналы на информационных входах [illegible] кодовой комбинации, заданной таблицей 11. На управляющих [illegible]ах проставить сигналы обеспечивающие заданный режим работы.
6. Привести временную диаграмму, поясняющую заданный режим работы для микросхемы.

В этой задаче предлагается ответить на вопросы по структуре последовательностных цифровых устройств. Решению этой задачи предшествует изучение тем: «Триггеры», «Счетчики», «Регистры». Простейшие последовательностные устройства - триггеры являются основой для построения схем регистров и счетчиков. Поэтому с

методической точки зрения приступать к ответу на вопросы задачи №4 без внимательного, детального изучения триггеров нет смысла.

Подробный теоретический материал по предлагаемым темам Вы найдете в [1], [2] стр. 98-109, 126-143, Методические указания стр. 23-26. Рекомендуется использовать также и другие источники: [4], [7], [8], [11].

Условное графическое обозначение заданной микросхемы и ее описание можно найти в [5], [6], [8], [13], [15], [17.]

Перед ответом на вопросы обратите внимание на *Ксч* счетчика и разрядность регистра. Предлагаемые к изучению микросхемы счетчиков имеют *Ксч* либо10 (двоично-десятичные), либо 16 (двоичные). Микросхемы регистров многофункциональные. От правильного определения выполняемых функций, соответствия выводов весовым разрядам числа зависит точность ответов на последующие вопросы. Пример построения временной диаграммы для счетчика показан на рисунке 7 , а расчет состояния при поступлении заданного числа импульсов поясняется формулой 9.

Приведем пример.

На вход десятичного счетчика, начальное состояние которого M = 6, поступило 27 импульсов. Какое состояние будет зафиксировано в счетчике после окончания счета?

Определим общее количество импульсов, поступивших на вход счетчика:

6+27=33.

Чтобы зафиксировать число 33, счетчик просчитает три полных цикла (*i*) и в следующем цикле зафиксирует остаток:

$$33\text{-}3*10=3_{(10)}=0011_{(2)}.$$

Таким образом в счетчике будет записано число $0011_{(2)}$, т.е $Q_4=0$, $Q_3=0$, $Q_2=1$ и $Q_1=1$.

Все действия в регистрах зависят от значения управляющих сигналов и в происходят в момент поступления на вход разрешающего сигнала (тактирования, синхросигнала). Обычно переключения происходят в момент действия либо фронта импульса, либо среза. Такие регистры построены на триггерах с динамическим входом. Со статическим управлением могут быть только параллельные регистры.

Понятие «весовой коэффициент» к разрядам регистра в отличие от счетчика неприменимо, поскольку весовая зависимость между отдельными разрядами целиком определяется записанной в регистр информацией. Однако разработчики микросхем для однозначного понимания действий, связанных с определением таких понятий, как направление сдвига (вправо или влево) и др. такую зависимость определяют. Поэтому при выборе микросхемы следует определить

ший выход соответствует старшему разряду, а какой младшему. Это не ...ется определяется старшинством индексов при буквенных символах ...ходов (К155ИР13).

Пример построения временной диаграммы для регистра сдвига ...подробно показан в [1], [2] стр. 129.

Задача №4

1. Выбрать из таблицы 12 микросхему запоминающего устройства для ... варианта.

2. Привести условное графическое обозначение микросхемы.

3. Привести краткое описание микросхемы: тип ЗУ, особенности ... чтения, технология изготовления, совместимость с другими ИМС и ... других серий, определить организацию микросхемы и ее емкость. ...сты необходимо обосновать.

4. Определить и указать значения логических сигналов на УГО, ...емых на входы микросхемы для обеспечения заданного режима работы.

5. Записать адрес ячейки памяти, к которой происходит обращение, в ...ичной и шестнадцатеричной системах счисления.

Таблица 12

№ варианта	Тип микросхемы	Режим работы	Адрес ячейки памяти
1	К176РУ2	запись 1	10011011
2	К537РУ2А	запись 0	011100101100
3	К541РУ1	чтение	100111011010
4	К556РТ5	чтение	011110010
5	К565РУ3А	запись 0	1000111
6	К573РФ5	чтение	00111110010
7	К1601РР1	запись 1	1010101110
8	КР1610РЕ1	чтение	10000111110
9	К573РР2	чтение	11111001101
10	К1601РР3	запись 0	11101011001

Запоминающее устройство (ЗУ) – это совокупность аппаратных ...дств, обеспечивающих существование информации во времени. В ...тав вычислительных систем входит одновременно несколько типов ...поминающих устройств, а чаще всего все: сверхоперативные, ...перативные, постоянные и внешние. Все они отличаются принципом ...йствия, характеристиками, техническими решениями.

Основными действиями в памяти являются операции записи, ...нения и чтения. Время выполнения операции обращения к памяти

определяет быстродействие данного типа ЗУ. Количество информации которое может одновременно храниться в ЗУ, называется емкостью. Показатель, определяющий количество слов и их разрядность, которые могут храниться в ЗУ, называется организацией.

Ёмкость запоминающих устройств принято измерять в битах. Б – это один двоичный разряд. Более крупная единица – байт, э машинное слово длиной в восемь бит. Далее используются следующ единицы:

$$1\text{Кбайт} = 1024 \text{ байта} = 8*2^{10} \text{ бит};$$
$$1 \text{ Мбайт} = 2^{20} \text{ байт} = 8*2^{20} \text{ бит};$$
$$1\text{Гбайт} - 2^{30} \text{ байт} = 8*2^{30} \text{ бит}.$$

ОЗУ предназначены для хранения программ и данных непосредственно используемых процессором в ходе выполнени операций. Содержимое ОЗУ в ходе решения задачи изменяется. Дл повышения быстродействия ОЗУ в состав вычислительного устройств вводится сверхоперативная память (СОЗУ). Она предназначена дл хранения небольшого числа слов (несколько десятков) и конструктивн входит в состав процессора.

Постоянное ЗУ (ПЗУ) содержит информацию, которая не должн изменяться в ходе выполнения процессором задачи. Это обычно стандартные программы, таблицы данных, постоянные коэффициенты Существует разновидность постоянных ЗУ, допускающа неоднократное перепрограммирование (РПЗУ). Как и ОЗУ, постоянны ЗУ реализуются на основе микроэлектронной элементной базе и выпускаются в виде микросхем.

Внешние ЗУ предназначены для хранения больших массиво информации в течение длительного промежутка времени. Они обычно строятся на основе магнитных свойств вещества.

Основной частью запоминающего устройства является масси элементов памяти, объединенных в матрицу накопителя. Элемент памяти (ЭП) может хранить один бит информации (0 или 1). Каждый ЭП имеет свой адрес. Для обращения к ЭП необходимо его выбрать с помощью кода адреса, который подается на входы. ЭП объединяются в группу, называемую ячейкой. Все элементы ячейки выбираются одним адресом. В качестве оперативной памяти обычно используются полупроводниковые ЗУ, которые по способу хранения информации делятся на статические и динамические. В статических ОЗУ в качестве ЭП применяются статические триггеры на биполярных или МДП транзисторах. При наличии напряжения питания триггер сохраняет свое состояние неограниченное время.

В динамических ОЗУ элементы памяти выполнены на основе конденсаторов, сформированных внутри полупроводникового кристалла. Такие элементы памяти не могут долгое время сохранять свое состояние и нуждаются в периодическом восстановлении (регенерации). Динамические ОЗУ отличаются от статических большей информационной емкостью, что обусловлено меньшим числом компонентов в одном элементе памяти и, следовательно, более плотным их размещением в кристалле полупроводника. Однако они сложнее в применении, т.к. нуждаются в организации принудительной регенерации, в дополнительном оборудовании и в усложнении сигналов управления.

Методические указания по выполнению задачи №4

Перед решением этой задачи необходимо изучить материал стр. 113-181 [1], [2], стр. 8-45 [10].

1. УГО микросхемы и назначение выводов можно найти в Приложении В. Более подробное описание микросхем ЗУ приведено в [1], [2].

2. Тип микросхемы ЗУ можно определить по маркировке (Приложение Б) и по условному графическому обозначению (Приложение А). Организация и емкость микросхемы памяти определяется по числу адресных и информационных входов/выходов. По формуле (17) можно выполнить расчет емкости ЗУ.

$$E = n*2^m \qquad (17)$$

где m – число адресных входов;
n – разрядность хранимых слов;
E – емкость.

Пример.

Задана микросхема К1500РУ073. Режим работы - запись числа $1011_{(2)}$ в ячейку памяти с адресом $1001101_{(2)}$.

1. Условное графическое обозначение микросхемы приведено на рис. 13.

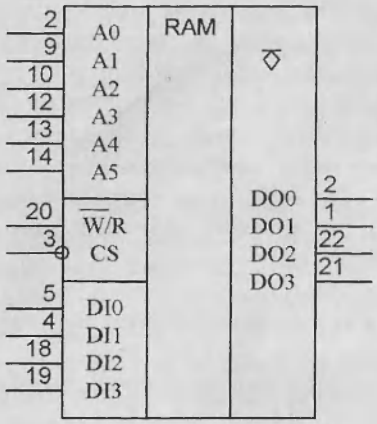


Рис.13. УГО микросхемы К1500РУ073

Эта микросхема является оперативным запоминающим устройством, на что указывает обозначение на УГО – RAM и третий элемент маркировки – РУ. Может работать в трех режимах: запись, хранение и чтение информации. При высоком уровне на входе CS микросхема переводится в режим хранения. На выходах устанавливаются низкие уровни. При CS = 0 и WR/RD = 0 (на УГО этот вывод обозначен $\overline{W}/R$) возможна запись числа, установленного на входах DI0 – DI3. При CS = 0 и при WR/RD = 1 микросхема переводится в режим чтения. На выходах DO0 – DO3 устанавливается код числа записанного в ячейку памяти, адрес которого установлен на входах A0 – A5. У микросхемы выходы выполнены по схеме с открытым эмиттером, на что указывает символ - $\overline{\diamond}$. Микросхема выполнена по технологии ЭСЛ По электрическим параметрам микросхема совместима с ЭСЛ – схемами. Микросхема имеет организацию 64×4, т.е. 64 слова (шесть адресных входов) по 4 бита (шина данных вход и выход – четырехразрядная).

Назначение выводов:

A5 – A0 адресные входы;

DI3 - DI0 – входы данных;

DO3 – DO0 – выходы данных;

$\overline{WR}/RD$ – запись/чтение. При значении сигнала "1" – чтение;

CS – разрешение обращения к микросхеме. При CS=0 обращение разрешено.

2. Значение сигналов на входах и выходах указано на рисунке 14. Производится запись числа $11_{(10)}$ в ячейку памяти с адресом $37_{(10)}$.

3. Если на адресных входах установлен код $100101_{(2)}$, то выбрана будет ячейка памяти $37_{(10)}$. Расчет:

$$100101_{(2)} = 1\cdot 2^5 + 0\cdot 2^4 + 0\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 32 + 4 + 1 + = 37_{(10)}$$

$37_{(10)} = 25_{(16)}$

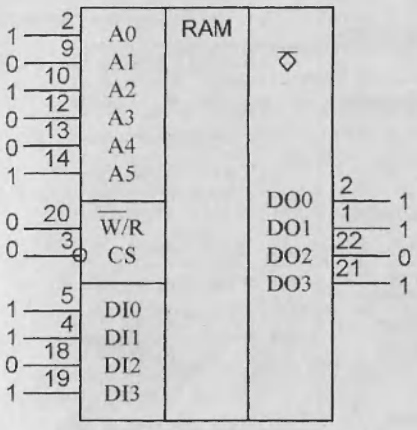


Рис. 14. Совместное действие сигналов обращения к памяти

Задача №5

1. Привести структурную схему микропроцессора К580ВМ80.
2. Привести описание узлов микропроцессора, заданных в таблице 13.
3. Составить программу для разветвляющегося вычислительного процесса по выражению, заданному в таблице 13.

Методические указания по выполнению задачи №5

В задаче предложены вопросы по архитектуре микропроцессора и системе команд. За основу взят микропроцессор серии К580. Перед решением задачи необходимо изучить материал [1], [2] стр. 233-269. Обратите внимание на то, какие узлы являются программно доступными, какие способы адресации используются и в чем состоит отличие, что представляет собой формат команды и как записываются код операции и данные.

1. Структурная схема микропроцессора К580 ВМ80 приведена на стр. 236 [1], [2], стр. 236.
2. Описание узлов также можно найти в [1], [2] на стр. 235-238
3. Рассмотрим пример решения задачи на программирование (формула 18). Основные команды микропроцессора приведены в Приложении Г.

Таблица 13

№ варианта	Узлы микропроцессора	Формула
1	Регистр данных	$y=\begin{cases}4-(2x\wedge 54) & если \quad x<52\\ 3x-k & если \quad x=52\\ 6+x & если \quad x>52\end{cases}$
2	Счетчик команд	$y=\begin{cases}\overline{3x} & если \quad x<35\\ (k-x) & если \quad x=35\\ x+5 & если \quad x>35\end{cases}$
3	Арифметико-логическое устройство	$y=\begin{cases}x+87 & если \quad x<23\\ 2x-k & если \quad x=23\\ 25\vee x & если \quad x>23\end{cases}$
4	Регистр признаков	$y=\begin{cases}5\vee k & если \quad x<60\\ 4x+42 & если \quad x=60\\ k/2\oplus x & если \quad x>60\end{cases}$
5	Схема управления	$y=\begin{cases}6x-k & если \quad x<12\\ 8+x & если \quad x=12\\ (25\vee x)+k-7 & если \quad x>12\end{cases}$
6	Регистр данных	$y=\begin{cases}5x & если \quad x<35\\ 24-k & если \quad x=35\\ x/4 & если \quad x>35\end{cases}$
7	Счетчик команд	$y=\begin{cases}3+x & если \quad x<54\\ 2x-51 & если \quad x=54\\ k-x & если \quad x>54\end{cases}$
8	Арифметико-логическое устройство	$y=\begin{cases}5k-x & если \quad x<17\\ (x\vee k)-37 & если \quad x=17\\ 2x+3 & если \quad x>17\end{cases}$
9	Регистр признаков	$y=\begin{cases}x+k & если \quad x<10\\ 5x-k & если \quad x=10\\ x/2+25 & если \quad x>10\end{cases}$
10	Схема управления	$y=\begin{cases}x+87 & если \quad x<23\\ 2x-k & если \quad x=23\\ 25\vee x & если \quad x>23\end{cases}$

Перед тем как приступить к работе над этой задачей необходимо по [1], [2] рассмотреть примеры программирования последовательных алгоритмов, программирование разветвлений и программирование циклических вычислительных процессов (стр. 257 – 265).

Решение задачи начинается с распределения памяти. Область памяти, доступная программисту, определяется техническими условиями устройства, на котором пишется программа. В данной задаче предлагается программу разместить в памяти, начиная с адреса.0800H (символ «H» после числа указывает на то, что число записано в шестнадцатеричной системе счисления), а исходные данные и конечный результат в ячейках памяти, начиная с адреса 0850H.

Число "x" – переменная, от которой зависит выбор выражения для вычислений. Число "k" – величина, значение которой должно быть определено на момент начала работы программы, а перед составлением программы надо знать адрес этого числа в памяти. Предлагается разместить исходные данные и результат следующим образом (в контрольной работе используйте эти же ячейки):

результат поместить в ячейку – 0850H;
число "x" – в ячейку – 0851H;
число "k" – в ячейку 0852H.

$$y=\begin{cases} 41-x/2 & \text{если} \quad x<81 \\ 3x \vee k & \text{если} \quad x=81 \\ \overline{k+x} & \text{если} \quad x>81 \end{cases} \qquad (18)$$

Далее составляем алгоритм вычислений (рисунок 15).

В первом блоке выполняется чтение числа "x" в аккумулятор. Это число потребуется в дальнейших операциях и чтобы не повторять операцию чтения, сохраним число в регистре B (второй блок). Для этих действий можно использовать команды *LDA addr* и *MOV R1,R2*. Обозначение *addr* означает шестнадцатеричный адрес ячейки памяти, например, *LDA 023E* чтение содержимого ячейки памяти 023EH в аккумулятор.

Командой третьего блока проводим сравнение числа "x" с константой 81. Для этой цели надо использовать команду сравнения *CPI data* (сравнение с константой). По этой команде от содержимого аккумулятора вычитается второй байт команды т.е. выполняется действие: x-81. Результат не сохраняется, а признаки устанавливаются в зависимости от результата вычитания. Признак Tc (C) в этом случае является признаком займа. Он устанавливается в "1", если содержимое аккумулятора меньше константы ((A) < data) и устанавливается в "0", если содержимое аккумулятора больше константы ((A) > data). Следующая команда (блок 4) - команда условной передачи управления. Это может быть команда *JNC addr* или *JC addr*. При выполнении этих команд процессор проверяет признак Tc. Если значение

признака истинно (для команды *JNC* истинным будет Тс = 0, а для команды *JC* - Тс = 1), то следует переход к команде адрес которой *addr*. Если условие ложно, перехода нет, и процессор выполняет команду следующую за командой передачи управления. Для дальнейшего описания выберем вариант, когда Тс = 1. В этом случае число "х" меньше числа 81, выполняется вычисление по формуле:
y = 41-x 2.

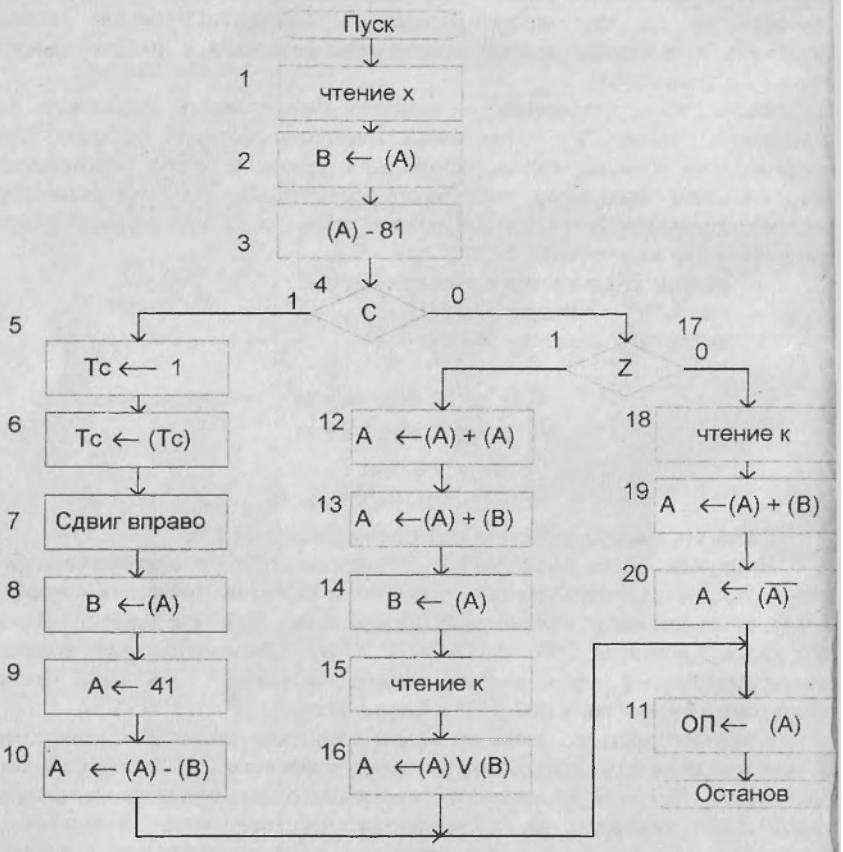


Рис. 15 . Алгоритм вычислений

Операцию деления на два (блок 7) можно реализовать *командой сдвига вправо через триггер переноса RAR.* По этой команде содержимое

аккумулятора циклически сдвигается на один разряд вправо. При этом младший бит перемещается в триггер переноса Тс, а на место старшего бита помещается прежнее значение триггера переноса. Это действие будет эквивалентно арифметической операции деления в том случае, если в триггере переноса будет записан "0". В этом случае значение Тс неизвестно и необходимо принять меры к установке этого триггера. Команды блоков 5 и 6 и предназначены для выполнения этой процедуры. Такие же действия необходимо предпринимать каждый раз перед командой сдвига, если она рассматривается как команда деления на два. С помощью последовательности этих команд можно выполнить деление на два (один сдвиг вправо), деление на четыре (два сдвига вправо), деление на восемь (три сдвига вправо) и т.д. В рассматриваемом примере триггер Тс уже установлен в единицу (признак С = 1) и блок 5можно опустить.

Следует иметь ввиду, что младший разряд при таких действиях теряется и ответ будет без дробной части. При делении нечетного числа (младший разряд равен единице) в ответе будет только целая часть. Например, если делить число $13_{(10)}= 1101_{(2)}$, то ответ будет $6_{(10)} = 0110_{(2)}$.

Если надо выполнить деление на любое другое число не кратное степени двух (2, 4, 8, 16, 32 и т.д.), то надо использовать специальные программы, например [3], стр. 117-120.

Чтобы из числа 41 вычесть полученный выше результат (блок 10), предварительно сохранить его (результат) в каком-либо регистре (например, B), а в аккумулятор записать 41 (блок 9). На этом вычисления по заданному выражению заканчиваются и далее командой блока 11 (*STA addr*) производится загрузка результата в память. По условию задачи это ячейка № 0850H.

Если при проверке условия в блоке 4 признак Тс окажется равным единице, это может означать, что x ≥ 81. Чтобы определить, в какую область попадает число "x", необходимо проверить значение признак Z. Этот признак будет установлен командой сравнения (блок 3) и при выполнении команды проверки признака Тс останется неизменным. Проверку признака Z (блок 17) можно выполнить также с помощью двух команд: *JNZ addr* и *JZ addr*. В первом случае переход по указанному адресу будет иметь место, если Z=0, а во втором – если Z=1.

Продолжим рассматривать алгоритм при условии, что Z=1. В этом случае надо выполнить расчет по выражению: $y = 3x \vee k$.

Действия, соответствующие арифметическому умножению, можно организовывать так же, как и команды деления, но сдвигать надо влево. Это будет соответствовать умножению на 2, 4, 8 и т.д. Триггер переноса предварительно обнуляется. Но можно выполнить и другие операции,

результат действия которых будет эквивалентен арифметическому умножению. Это сложение чисел. В микропроцессоре предусмотрена команда сложения *ADD R*. R - это любой регистр. Если записать команду *ADD A*, то процессор выполнит двойное сложение аккумулятора и результат поместит в аккумулятор (блок 12). Таким образом в аккумуляторе будет число равное 2x. Затем к этому числу прибавляем еще раз число x (оно было записано в регистр B (блок 2)), и итоге получается значение 3x.

Прямое чтение из памяти возможно только в аккумулятор, поэтому командой блока 14 полученный результат сохраняется в регистре B, а затем выполняется чтение числа *k*. Над двумя числами *k* и 3x выполняется логическая операция ИЛИ (блок 16). Такая команда в процессоре предусмотрена: *ORA R*. По этой команде содержимое аккумулятора и адресуемого регистра поразрядно перемножается и результат загружается в аккумулятор. Затем следует переход к записи результата в ячейку № 0850H оперативной памяти.

Если при проверке признака Z (блок 17) окажется, что он равен "0", то выполняется выражение: $y = \overline{k + x}$. Число "x" находится в регистре B (блок 2). Команда блока 18 выбирает из памяти число k и затем выполняется команда арифметического сложения (блок 19). Команда блока 20 выполняет инвертирование содержимого аккумулятора. Для этого предусмотрена команда CMA. Далее следует загрузка результата в память.

Последней командой в программе должна быть команда останова – *HLT*.

Далее приведем текст программы. Для этого удобно использовать следующую таблицу (табл. 14). Первый столбик указывает абсолютное значение адреса команды. Второй – мнемоническое обозначение команды. Третий – шестнадцатеричный код этой команды. Последний столбик содержит пояснения к команде или группе команд.

Все цифровые величины в программе должны быть представлены в шестнадцатеричной системе счисления. Поэтому выполним перевод чисел:

$81_{(10)} = 51_{(16)}$;

$41_{(10)} = 29_{(16)}$.

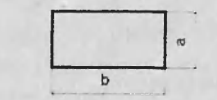
Таблица 14. Текст программы

Адрес	Мнемоника	Шестнадцатеричный код	Комментарий
0800	LDA 0851	3A5108	Чтение числа x в аккумулятор
0803	MOV B,A	47	Запись числа x в регистр B
0804	CPI 51	FE51	Сравнить число "x" с числом 81
0806	JNC 0813	D21308	Перейти, если признак C=0
0809	STC	37	Обнулить триггер переноса
080A	CMC	3F	
080B	RAR	1F	Делить на два
080C	MOV B,A	47	Сохранить промежуточный результат
080D	MVI A,29	3E29	Загрузить в аккумулятор число 41
080F	SUB B	90	Выполнить действие: 41 – x/2
0810	JMP 0825	C32508	Перейти к загрузке результата
0813	JZ 081E	CA1E08	Перейти, если x = 81
0816	LDA 0852	3A5208	Чтение числа k
0819	ADD B	80	Сложить числа x и k
081A	CMA	2F	Инвертировать сод-мое аккум-тора ($\overline{k+x}$)
081B	JMP 0825	C32508	Перейти к загрузке результата
081E	ADD A	87	Умножить число x на три и сохранить результат
081F	ADD B	80	
0820	MOV B,A	47	
0821	LDA 0852	3A5208	Чтение числа k
0824	ORA B	B0	Логическое сложение: $3x \vee k$
0825	STA 0850	325008	Загрузка результата
0828	HLT	76	Останов

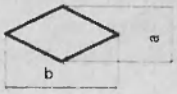
Необходимо обратить внимание на то, что размещение команд по адресам памяти зависит от их формата. Например, команда *LDA 0851* трехбайтовая и она занимает три ячейки памяти, поэтому следующая по адресу команда *MOV B,A* занимает ячейку № 0803.

Если необходимо выполнить проверочные расчеты, то следует помнить, что процессор представляет отрицательный результат в дополнительном коде, при этом знак числа в коде не отражается.

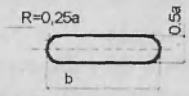
При вычерчивании схемы алгоритма (задача №5 и курсовая работа) необходимо соблюдать требования ГОСТ 19.701-90 «Схемы алгоритмов, программ, данных и систем». Этот стандарт устанавливает правила изображения алгоритмов и его элементов. При выполнении схемы алгоритма рекомендуется соблюдать следующие размеры элементов (рис. 16).



Символ Процесс



Символ Решение



Символ начала алгоритма и ок
ончания алгоритма

Рис. 16. Элементы алгоритма

Размер a выбирается из ряда 10, 15, 22 мм. Допускается увеличивать размер a на число кратное 5. Размер b равен 2a.

4. Задания для курсовой работы

4.1. Порядок выполнения курсовой работы

По структуре курсовая работа состоит из:

- введения, в котором раскрывается актуальность разрабатываемой темы, формулируются цели и задачи работы;
- основной части;
- заключения;
- списка используемой литературы;
- приложения.

Основная часть состоит из двух разделов.

В первом разделе содержатся теоретические основы разрабатываемой темы.

Вторым разделом является практическая часть, которая должна содержать следующие пункты:

- алгоритм функционирования цифрового автомата;
- определение состояний МПА;
- кодирование состояний, описание заданного триггера;
- построение графа;
- таблица функционирования МПА;