

ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ
для проведения практических занятий по дисциплине
«Периферийные устройства и интерфейсы»
Б1.В.ДВ.11.02

Кафедра **«Информатика и вычислительная техника»**

Направление подготовки **09.03.01. Информатика и вычислительная техника**

Профиль **Программное обеспечение и интеллектуальные системы**

Формы обучения **очная, заочная**

Разработала:
Доцент кафедры ИВТ Швидченко С.А.

Содержание

ПРАКТИЧЕСКАЯ РАБОТА № 1. ПЕРЕВОД ЧИСЕЛ. ЛОГИЧЕСКИЕ ЭЛЕМЕНТЫ. ЭЛЕМЕНТЫ И УЗЛЫ ЭВМ	3
1.1. КРАТКАЯ ТЕОРИЯ.....	3
1.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	5
1.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	6
ПРАКТИЧЕСКАЯ РАБОТА № 2. СПОСОБЫ АДРЕСАЦИИ. МИКРООПЕРАЦИИ И МИКРОПРОГРАММЫ	8
2.1. КРАТКАЯ ТЕОРИЯ.....	8
2.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	9
2.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	10
ПРАКТИЧЕСКАЯ РАБОТА № 3. ОРГАНИЗАЦИЯ ПАМЯТИ ЭВМ	12
3.1. КРАТКАЯ ТЕОРИЯ.....	12
3.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	144
3.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	155
ПРАКТИЧЕСКАЯ РАБОТА № 4. ОРГАНИЗАЦИЯ ВНЕШНЕЙ ПАМЯТИ ЭВМ. АЛГОРИТМЫ УМНОЖЕНИЯ ЧИСЕЛ. АЛГОРИТМЫ УМНОЖЕНИЯ ЧИСЕЛ.....	166
4.1. КРАТКАЯ ТЕОРИЯ.....	166
4.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	17
4.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	18
ПРАКТИЧЕСКАЯ РАБОТА № 5. РЕАЛИЗАЦИЯ КОМАНД ПЕРЕСЫЛКИ И ОПЕРАЦИЙ СО СТЕКОМ. ВЫПОЛНЕНИЕ АРИФМЕТИЧЕСКИХ И ЛОГИЧЕСКИХ КОМАНД	20
5.1. КРАТКАЯ ТЕОРИЯ.....	20
5.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	21
5.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	23
ПРАКТИЧЕСКАЯ РАБОТА № 6. ПРОЦЕССОРЫ, КОНТРОЛЛЕРЫ, ВНЕШНИЕ УСТРОЙСТВА	25
6.1. КРАТКАЯ ТЕОРИЯ.....	25
6.2. МЕТОДИКА ВЫПОЛНЕНИЯ.....	26
6.3. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	27
ЛИТЕРАТУРА	288

ПРАКТИЧЕСКАЯ РАБОТА № 1.

Перевод чисел. Логические элементы. Элементы и узлы ЭВМ

1.1. КРАТКАЯ ТЕОРИЯ

Триггер представляет собой электронное устройство с двумя состояниями устойчивого равновесия, что и делает его удобным для хранения информации в двоичной системе счисления. Бистабильный триггер имеет два выхода, причём информация на одном из выходов \bar{Q} представляет собой инверсию информации на другом выходе Q ($\bar{}$ - знак инверсии логической переменной). Запись информации в триггер обычно осуществляют через схему управления.

Триггеры классифицируют по ряду признаков. По функциональным возможностям выделяют:

- а) триггер с раздельной установкой 0 и 1 (RS-триггер);
- б) триггер с приёмом информации по одному входу (D-триггер), другое название – триггер задержки;
- в) триггер со счётным входом (T-триггер);
- г) универсальный триггер (JK-триггер).

Таблица 1

Такт n			Такт n+1	Режим
Q ⁿ	R	S	Q ⁿ⁺¹	
Q ⁿ	1	0	0	Reset-уст. 0
Q ⁿ	0	1	1	Set-уст. 1
Q ⁿ	0	0	Q ⁿ	хранение
Q ⁿ	1	1	н/о	запрет

RS		00	01	11	10
Q ⁿ	0	0	1	н/о	0
	1	1	1	н/о	0

а)

б)

Рис.1. Таблица истинности (а) и карта Карно (б) для триггера как элемента памяти

синхронизируемые фронтом (реагируют на информационные сигналы в момент изменения сигнала на С-входе от 0 к 1 – положительный фронт или от 1 к 0 – отрицательный фронт).

Основной любого триггера является элемент памяти. Чаще всего в статических триггерах эту роль выполняет бистабильная ячейка памяти (простейший асинхронный RS-триггер), которая функционирует в соответствии с таблицей истинности, приведённой на рис. 1,а. Исходное состояние триггера характеризуется сигналом на прямом выходе Q^n , а новое состояние Q^{n+1} определяется комбинацией входных сигналов (н/о- не определённое состояние). Например, для установки триггера в состояние 0 необходимо на его входы подать такую комбинацию сигналов, при которой на прямом выходе сигнал будет иметь уровень логического 0, т. е. $Q^{n+1}=0$, а $\bar{Q}^{n+1}=1$.

Из таблицы истинности и карты Карно следует, что поведение такого элемента памяти описывается следующим характеристическим уравнением:

$$Q^{n+1} = S \vee \bar{R} * Q^n \dots (1)$$

Если воспользоваться правилом, что двойное отрицание не меняет значение логической функции, то выражение (1) можно записать в следующем виде: $Q^{n+1} = \bar{\bar{S}} \vee \bar{R} * Q^n \dots (2)$ и в соответствии с формулами де-Моргана $Q^{n+1} = \bar{(\bar{S} * \bar{R} * \bar{Q}^n)} \dots (3)$. Здесь знаки $\bar{\bar{()}}$ или $\bar{()}$ означают двойную или одинарную инверсию выражения в скобках.

Выражения (2) и (3) определяют два варианта построения элемента памяти - RS-триггера, функционирующего в соответствии с таблицей 1. Первый строится на логических элементах (ЛЭ) ИЛИ-НЕ, а второй – строится на элементах И-НЕ. На рис.2 приведены функциональные

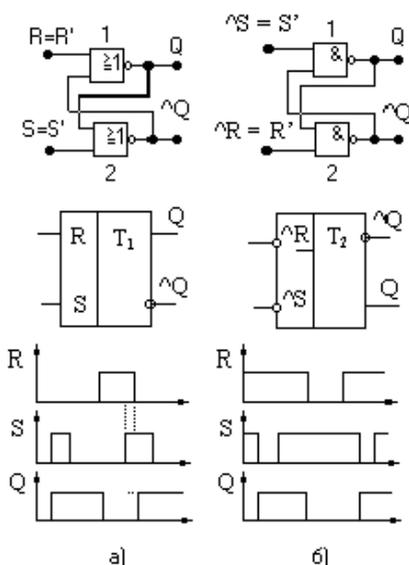


Рис.2. Функциональные схемы, условные обозначения и временные диаграммы RS-триггера на ИЛИ-НЕ (а) и И-НЕ (б)

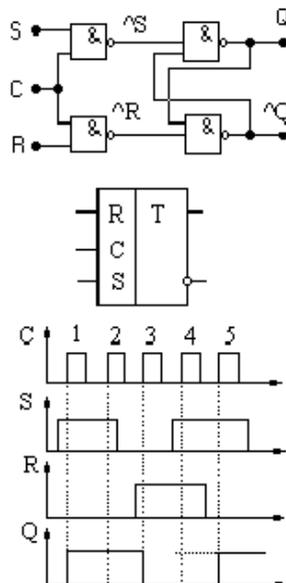


Рис.3. Функциональная схема тактируемого RS-триггера, условное обозначение и временные диаграммы

схемы, условные обозначения и временные диаграммы **асинхронных RS-триггеров**, построенных на ИЛИ-НЕ (рис. 2,а) и на И-НЕ (рис. 2,б).

Для RS-триггера на ЛЭ ИЛИ-НЕ активным является уровень логической 1. По временной диаграмме (рис. 2,а) видно, что единичный сигнал на S-входе устанавливает триггер в состояние "1" (на его выходе Q=1). По окончании действия сигнала S=1 на входах триггера сигналы S=R=0 (режим хранения). Сигнал R=1 устанавливает триггер в состояние "0" (Q=0). Пунктиром отмечен интервал времени одновременного появления сигнала на R- и S-входах логической 1, что является запрещённым состоянием для триггера этого типа. Состояние выходного сигнала Q в этот интервал времени не определено (пунктирная линия). После сброса до 0 сигнала на R-входе Q=1 за счёт действия логической 1 на S-входе. Прекращение действия сигнала на S-входе и состояние S=R=0 устанавливает режим хранения для Q=1. Для RS-триггера на ЛЭ И-НЕ активным является уровень логического 0. По временной диаграмме (рис. 2,б) видно, что состояние S=R=1 обеспечивает режим хранения, поддерживая Q=0. Логический 0 на S-входе устанавливает Q=1, сохраняющийся и при возвращении S-входа к уровню 1. Отрицательный импульс на R-входе устанавливает Q=0. Затем в течение короткого интервала времени R=S=1 (режим хранения Q=0), и установка 1 (при S=0, Q=1), что и сохраняется в дальнейшем при R=S=1.

Большее применение находит **тактируемый RS-триггер**, переключение которого происходит при наличии импульса синхронизации C. Поэтому его характеристическое уравнение: $Q^{n+1} = C*(S \vee R*Q^n) \vee \bar{C}*Q^n \dots (4)$. Для построения функциональной схемы устройства, реализующего данное уравнение, применим по аналогии с (2) и (3) двойное инвертирование уравнения (4) и формулы де-Моргана. Тогда имеем $Q^{n+1} = \bar{\bar{C}}*(\bar{C}*S \vee C*\bar{R}*Q^n \vee \bar{C}*Q^n) = \bar{\bar{C}}*(\bar{C}*S \vee Q^n*(\bar{C}*\bar{R} \vee \bar{C})) = \bar{\bar{C}}*(\bar{C}*S \vee Q^n*(\bar{R} \vee \bar{C})) = \bar{\bar{C}}*(\bar{C}*S) \wedge (Q^n \wedge (\bar{R} \vee \bar{C})) \dots (5)$. Это уравнение определяет функциональную схему RS-триггера, синхронизируемого импульсом C (рис.3). Здесь же приведены его условное обозначение и временные диаграммы работы.

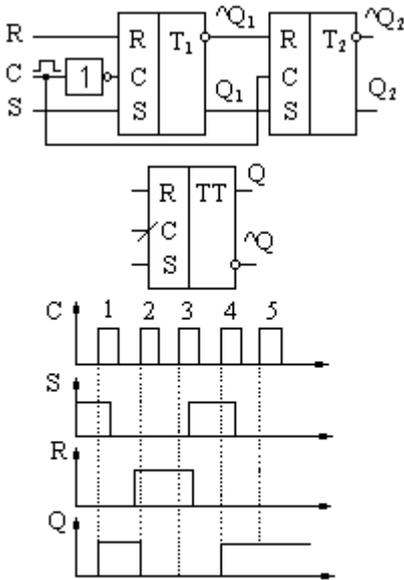


Рис.4. Функциональная схема, условное обозначение и временные диаграммы синхронизируемого передним фронтом RS-триггера

\bar{Q}_1 постоянны. Второй триггер T_2 сохраняет свое состояние.

Двухступенчатое запоминание информации в этом триггере отображается двумя буквами T в условном его обозначении. Реакция схемы на передний фронт идентифицируется отрезком под углом 45° на линии ввода сигнала в C-вход.

По временным диаграммам рис. 3 видно, что появление высокого уровня Q выходного сигнала триггера происходит при наличии первого импульса на C-входе (S=1). Это состояние сохраняется и после окончания действия импульса на S-входе за счёт второго тактового импульса C-входа. Установка триггера в «0» (R=1) происходит только при появлении третьего синхрои импульса, и состояние Q=0 поддерживается в течение времени действия третьего синхрои импульса и паузы между третьим и четвёртым синхрои импульсами. Одновременное действие S=R=1 при четвёртом синхрои импульсе и последующей паузе является запрещённым состоянием, поэтому выходной сигнал не определён (пунктирная линия) и установление Q=1 за счёт S=1 происходит только при пятом синхрои импульсе. Далее это состояние Q=1 сохраняется (режим хранения S=R=0).

Недостатком синхронизируемого импульсом RS-триггера является возможность изменения его состояния в течение всего времени действия синхрои импульса. Это обстоятельство устранено в RS-триггере, синхронизируемом фронтом импульса C.

На рис.4 приведены функциональная схема RS-триггера, синхронизируемого передним фронтом импульса, его условное обозначение и временные диаграммы работы. Синхрои импульс подаётся на C-вход триггера T_1 через инвертор. Поэтому при C=0 T_1 реагирует на изменения сигналов на входах R и S, нужным образом устанавливая Q_1 и \bar{Q}_1 . В момент изменения синхрои импульса C от 0 к 1 T_1 перестает реагировать на изменения R и S, а его последнее состояние Q_1 и \bar{Q}_1 запоминается на T_2 , так как синхрои импульс на его C-входе равен 1. Далее при C=1 T_1 не реагирует на изменения R и S, Q_1 и

Временные диаграммы работы триггера на рис. 4 показывают переход TT в единичное состояние за счёт S=1 по переднему фронту первого синхрои импульса и возвращение Q=0 по переднему фронту второго синхрои импульса ввиду R=1. Четвёртый импульс вновь устанавливает Q=1 за счёт S=1, и далее при S=R=0 реализуется режим хранения состояния Q=1. Такое схемотехническое построение триггера существенно повышает его устойчивость к воздействию различных помех.

Аналогично рис. 4 может быть построена схема RS-триггера, синхронизируемого задним фронтом синхрои импульса. Для этого инвертор должен быть включён в цепь на C-вход второго триггера. На условном обозначении управляемого задним фронтом RS-триггера отрезок на линии C-входа размещается под углом в 135° (рис. 5).

D-триггер имеет один информационный вход (D-вход) и вход для синхронизирующего импульса. Его функционирование реализуется в соответствии с таблицей истинности (табл. 2), из которой можно записать

Таблица 2

Таблица истинности D-триггера

C	Q^n	D	Q^{n+1}	Режимы
U	Q^n	x	Q^n	хранение
1	x	1	1	set- уст.1
1	x	0	0	reset- уст.0

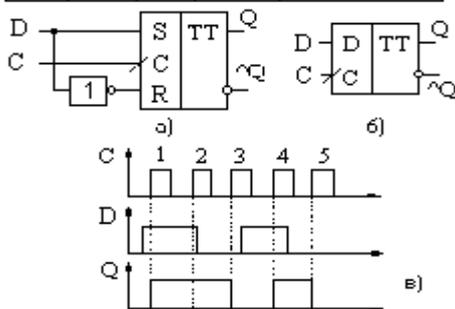


Рис.6. Функциональная схема (а), условное обозначение (б) и временная диаграмма работы (в) D-триггера

характеристическое уравнение для D-триггера : $Q^{n+1} = C * D \vee \wedge C * Q^n \dots(6)$. D-триггер строится на базе RS-триггера. Если у RS-триггера, работающего согласно уравнению $Q^{n+1} = C * (S \vee \wedge R * Q^n) \vee \wedge C * Q^n$ на вход S подать значение D (S=D), а на вход R подать значение $\wedge D$ (R= $\wedge D$), то получим характеристическое уравнение D-триггера: $Q^{n+1} = C * (D \vee \wedge D * Q^n) \vee \wedge C * Q^n = C * D \vee \wedge C * Q^n$ (рис. 6,а). Здесь же на рис. 6 показано условное обозначение D-триггера (б) и временные диаграммы его работы (в). Синхронизация D-триггера производится по переднему (для рис. 6) или заднему фронту тактирующего импульса.

T-триггер – это триггер с одним так называемым счётным входом, изменяющий своё состояние с приходом каждого входного импульса. Функционирование триггера должно идти в соответствии с таблицей истинности табл.3, на основании которой можно записать характеристическое уравнение в виде: $Q^{n+1} = C * Q^n \vee \wedge C * Q^n \dots(8)$, а на основании последнего получить T- триггер из D-триггера, положив $D = \wedge Q^n$. Функциональная схема T-триггера показана на рис. 7.

Большими логическими возможностями обладает TV-триггер, имеющий два логических входа T и V, первый из которых является счётным, а второй – разрешающим. TV триггер при V=1 выполняет функции T-триггера; если на входе V действует запрещающий сигнал 0, состояние триггера не меняется. Характеристическое уравнение,

Таблица 3
Таблица истинности
T-триггера

T	Q^n	Q^{n+1}
0	Q^n	Q^n
1	Q^n	$\wedge Q^n$

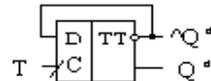
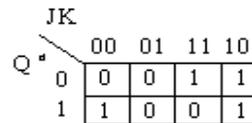


Рис.7. Функциональная схема T-триггера

Таблица 4

Q^n	J	K	Q^{n+1}	Режим
Q^n	1	0	1	set – уст.1
$\wedge Q^n$	0	1	0	reset – уст.0
Q^n	0	0	Q^n	хранение
$\wedge Q^n$	1	1	$\wedge Q^{n+1}$	счётчик



а) б)
Рис.8. Таблица истинности (а) и карта Карно (б) для JK-триггера при C=1

описывающее логические возможности TV-триггера, определяется соотношением: $Q^{n+1} = Q^n (\wedge T \vee \wedge V) \vee \wedge Q^n T * V$.

JK-триггер имеет два информационных входа: J и K, а так же вход для тактовых импульсов C. Правило работы JK-триггера определяется его таблицей истинности табл. 4 и картой Карно (рис. 8), на основании которых можно записать следующее характеристическое уравнение: $Q^{n+1} = C * (J * \wedge Q^n \vee \wedge K * Q^n) \vee C * Q^n$, на основании чего можно составить функциональную схему JK-триггера (рис. 9,а). Этот триггер строится на базе RS-триггера введением обратных связей с прямого и инверсного выходов на два входных элемента И, объединяющие сигналы обратной связи с сигналами управления J и K. На рис. 9,б приведено условное обозначение триггера этого типа, временные диаграммы (в) иллюстрируют его работу. В момент совместного действия логических 1 на входах J и K передний фронт третьего синхроимпульса вызывает срабатывание триггера в счётном режиме, изменяя Q с величины 0 на значение 1.

При отсутствии синхроимпульсов, т.е. C=0 JK-триггер работает в режиме хранения. На базе JK-триггера можно получить схемы всех остальных триггеров, что и приведено на рис. 10.

1.2. МЕТОДИКА ВЫПОЛНЕНИЯ

2.1. Включите необходимое программное обеспечение.

2.2. Соберите на монтажном столе схему асинхронного статического триггера по рис. 2,б. Убедитесь в её работоспособности,

спланируйте эксперимент и результаты запишите в виде таблицы истинности $Q^{n+1} = f(S, R, Q^n)$. Логические переменные R и S на входах задавайте с помощью переключателя кнопочного на два положения из контейнера **Control**, подключённые одним выводом к источнику +5 вольт, а другим выводом – к земле. Индикацию состояния выходов триггера обеспечьте индикаторами из контейнера **Indicators**. Составьте характеристическое уравнение RS-триггера.

2.3. При исследовании схем с триггерами перед началом эксперимента необходимо установить триггера в

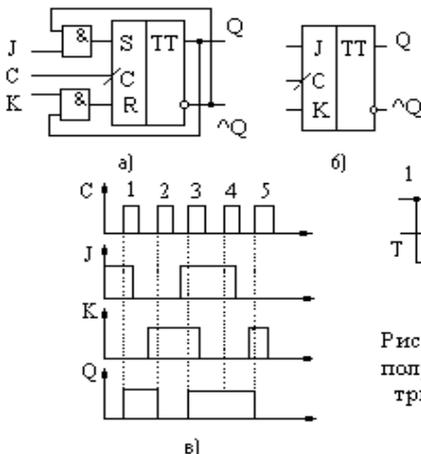


Рис.9. Функциональная схема (а), условное обозначение (б) и временные диаграммы (в) JK-триггера при C=1

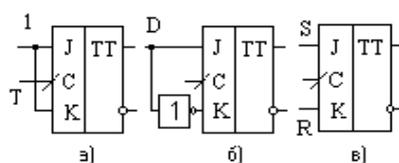


Рис.10. Использование JK-триггера для получения триггеров других типов: а) T-триггер; б) D-триггер; в) JK-триггер

определённое состояние (например, в логический 0), так как моделирующая программа не может разрешить противоречие: неизвестны значения Q^n и \bar{Q}^n , а $S=R=0$, т.е. задан режим хранения, а состояние Q^n – неизвестно. Поэтому во всех схемах этой работы первым шагом моделирования должна быть установка триггеров в определённое состояние (логический 0). Для этого в схему синхронизируемого импульсом RS-триггера на рис. 3 в логическом элементе 4 введён переключатель с задержками во времени из контейнера **Control**, у которого Time On =100 нс, а Time Off=10000 с. (рис.11). Этот переключатель в момент начала моделирования включается и подаёт логический 0 на дополнительный вход статического элемента памяти и устанавливает его в состояние логического 0, а через 100 нс отключается и по-даёт на этот вход логическую 1. С этого момента времени элемент памяти может нормально функционировать в моделирующей программе.

Соберите схему синхронизируемого импульсом RS-триггера по рис. 11,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности $Q^{n+1}=f(S,R,Q^n)$ при $C=1$. Составьте характеристическое уравнение. Убедитесь, что в начале моделирования триггер устанавливается в состояние 0. Поместите эту схему в подсхему RST_CI (рис.11,б).

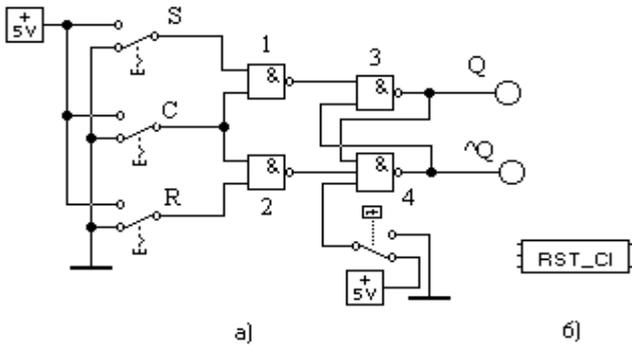


Рис.11. RS-триггер, синхронизируемый импульсом C, (а) и подсхема RST-CI на его основе (б)

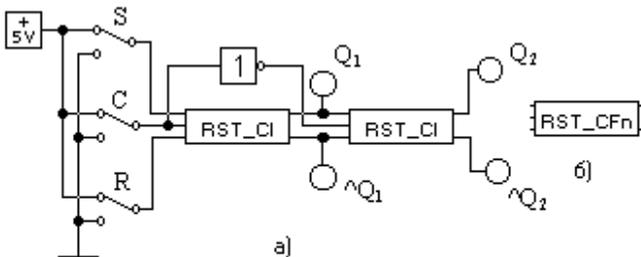


Рис.12. RS-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

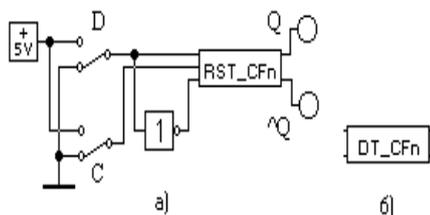


Рис.13. D-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

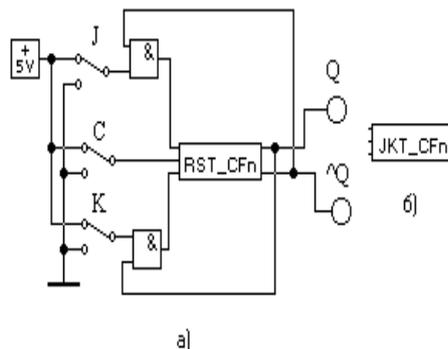


Рис.14. JK-триггер, синхронизируемый отрицательным фронтом импульса, (а) и его подсхема (б)

2.4. Используя подсхему RST_CI, соберите RS-триггер, синхронизируемый отрицательным фронтом импульса синхронизации по рис.12,а. Спланируйте эксперимент по исследованию схемы. С помощью логического анализатора из моделирующей программы продемонстрируйте преподавателю временные диаграммы зависимостей Q_1 и Q_2 от R, S и C. Поместите RS-триггер с синхронизацией от отрицательного фронта импульса в подсхему RST_CFn (рис.12,б), убрав предварительно индикацию состояний Q_1, \bar{Q}_1 .

2.5. Соберите схему D-триггера по рис. 13,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности $Q^{n+1} = f(D,C,Q^n)$. Составьте характеристическое уравнение для D-триггера. Поместите схему в подсхему DT_CFn (рис.13,б).

2.6. Соберите схему JK-триггера по рис. 14,а. Убедитесь в её работоспособности, спланируйте эксперимент и результаты запишите в виде таблицы истинности $Q^{n+1} = f(J,K,C,Q^n)$. Составьте характеристическое уравнение для JK-триггера. Поместите эту схему в подсхему JKT_CFn (рис.14,б).

Для использования созданных подсхем в новых схемах необходимо поместить их в контейнер пользователя файла **default.ca4** (через буфер обмена) и сохранить. Теперь при создании новой схемы следует выбирать существующий файл **default. ca4**, в котором будет появляться монтажный стол с контейнером, содержащим созданные вами подсхемы.

1.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как классифицируются триггеры по функциональным возможностям?
2. На базе каких логических элементов может строиться асинхронный RS-триггер? Какие уровни логической переменной являются активными для каждого из них?
3. Почему возникла необходимость введения тактируемых триггеров? Поясните по временной диаграмме рис. 3 работу такого триггера.

4. Поясните по временной диаграмме рис. 4 работу синхронизируемого передним фронтом RS–триггера.
5. В чём функциональное отличие между RS- и JK-триггерами, поясните работу последнего по временной диаграмме рис. 9.
6. Покажите схемные решения использования JK–триггера для построения триггеров всех других видов.

Контрольные вопросы ПЗ1 (ПК-4):

1. Что такое система счисления?
2. Что такое основание системы счисления?
3. Как зависит разрядность чисел от величины основания системы счисления?
4. Почему информация в компьютере представляется в двоичной системе счисления?
5. Какие системы счисления используются для представления информации в компьютере?
6. Для чего применяются двоично-десятичные коды?
7. Чем отличаются формы представления чисел с фиксированной и плавающей точками?
8. Как кодируются знаки чисел?
9. Можно ли различить форматы целых и дробных чисел с фиксированной точкой?
10. Какие элементы формата чисел с плавающей точкой вы знаете?
11. Чем отличается нормализованное число с плавающей точкой от ненормализованного числа?
12. Для чего применяются смещенные порядки?
13. Какой прием используется для расширения диапазона чисел с плавающей точкой?
14. В чем заключается способ скрытой единицы?
15. Охарактеризуйте форматы двоично-десятичных чисел.
16. Как кодируются знаки двоично-десятичных чисел?
17. Для чего используются машинные коды чисел?
18. Чем отличаются дополнительный и обратный коды отрицательных чисел?
19. Как из дополнительного кода отрицательного числа получить прямой код числа?
20. Каким образом кодируется текстовая информация?
21. Перечислите составные части компьютера в порядке уменьшения функциональной сложности.
22. Чем различаются узлы комбинационного и накапливающего типов?
23. Что такое логический элемент?
24. Что такое функционально полная система логических элементов?
25. Что такое состояние триггера?
26. Почему триггер имеет два устойчивых состояния?
27. Чем различаются асинхронный и синхронный триггеры?
28. Что такое таблица переходов триггера?
29. В каких случаях используются двухтактные триггеры?
30. Чем различаются триггеры разных типов?
31. Почему асинхронный D-триггер не используют в качестве элемента памяти?
32. Какие сигналы нужно подать на входы синхронного RS-триггера, чтобы записать в него «1»?
33. Что такое дешифратор?
34. Сколько входов и выходов имеет одноразрядный сумматор?
35. Чем различаются сумматоры с последовательными, сквозными и параллельными переносами?
36. Почему для сложения тетрад двоично-десятичных чисел нельзя использовать обычный двоичный сумматор?
37. Для чего используют мультиплексоры?
38. Из каких элементов состоит регистр?
39. Почему для сдвига данных используют двухтактные триггеры?
40. В чем состоит отличие схем суммирующего и вычитающего счетчиков?

ПРАКТИЧЕСКАЯ РАБОТА № 2.

Способы адресации. Микрооперации и микропрограммы

2.1. КРАТКАЯ ТЕОРИЯ

Совокупность из нескольких триггеров, предназначенных для хранения двоичного кода, называется регистром. Введением дополнительной логики функциональные возможности регистров расширяются, обеспечивая возможность сдвига записанной информации в ту или иную сторону.

В соответствии с назначением различают регистры хранения, регистры сдвига и универсальные регистры.

По принципу хранения информации регистры делятся на статические и динамические. Статические регистры строят на потенциальных элементах памяти (триггерах), которые могут хранить записанную информацию сколько угодно долго (при наличии напряжения питания). Динамические регистры строят на элементах памяти такого типа, как конденсатор, которые могут хранить информацию лишь в течение некоторого промежутка времени. Поэтому в динамических регистрах необходима регенерация записанной информации.

Важнейшие характеристики регистров – разрядность и быстродействие. Разрядность определяется количеством триггеров. Быстродействие характеризуется максимальной тактовой частотой, с которой может производиться запись, чтение и сдвиг информации.

Наиболее широко распространены регистры сдвига, предназначенные для преобразования информации путём её сдвига под воздействием тактовых импульсов. Такие регистры представляют собой совокупность последовательно соединённых триггеров, как правило, двухступенчатой структуры. По направлению сдвига информации различают регистры прямого сдвига (вправо, т.е. в сторону младшего разряда), обратного сдвига (влево, т.е. в сторону старшего разряда) и реверсивные, допускающие сдвиг в обоих направлениях.

В простейшем регистре триггеры соединены последовательно: выходы Q и \bar{Q} предыдущего триггера передают бит данных на входы R и S последующего. Все тактовые входы C триггеров соединены параллельно. При таком включении единица, записанная в виде напряжений низкого и высокого уровней по входам R и S первого триггера, после подачи одного тактового импульса перейдёт во второй триггер, затем во время следующего тактового импульса она попадёт в третий триггер и так последует далее, до конца регистра.

Аналогично продвигается по регистру многоразрядное слово: оно поразрядно вводится на входы R и S первого триггера. Простейший регистр имеет один вход и один выход – последовательные. Вход управления также единственный – тактовый. Если ко входу каждого триггера добавить разрешающую логику, можно получить дополнительные, так называемые параллельные входы одновременной загрузки байта в регистр. Здесь, как правило, используют дополнительные защёлки, где фиксируются данные, поступившие на входы после прихода тактового импульса. В такую схему добавляется вход разрешения записи.

Можно предусмотреть также логическую схему параллельного отображения на выходе состояния каждого триггера. Тогда после заполнения регистра от последовательного или параллельного входов по команде разрешения выхода накопленное цифровое слово можно отобразить поразрядно сразу на всех параллельных выходах. Для удобства поочерёдной выдачи данных от таких регистров – буферных накопителей в шину данных обрабатывающего устройства – процессора – параллельные выходы регистров снабжаются выходными буферными усилителями, имеющими третье, разомкнутое Z -состояние. По многопроводной шине данных процессор получает цифровое слово – байт от выходов того регистра, которому дана команда разрешения выдачи.

Наиболее широко распространены регистры сдвига на D -триггерах (рис. 1). Такие регистры имеют один информационный вход, вход для импульсов синхронизации (импульсов сдвига) и установочный вход R . Выходы в регистре могут быть с каждого разряда для считывания информации в параллельном коде. Также имеется один выход с последнего (относительно входа) разряда для считывания информации последовательно во времени, т.е. последовательным кодом. Вход регистра для импульсов сдвига получается объединением C -входов всех триггеров, а установочный вход – R -входов.

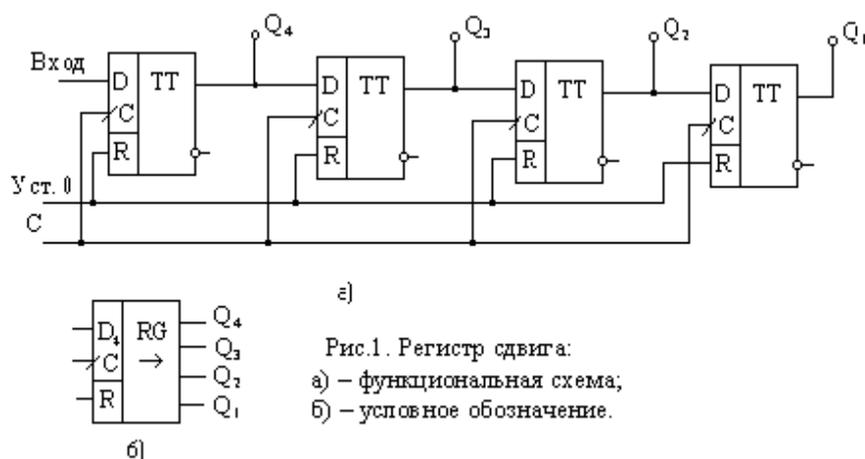


Рис.1. Регистр сдвига:
а) – функциональная схема;
б) – условное обозначение.

Перед записью информации регистр устанавливается в нулевое состояние подачей положительного импульса по шине «Уст. 0». Записываемая информация должна быть представлена последовательным кодом. Запись осуществляется поразрядно со стороны старшего (рис. 1) или младшего разряда (направление сдвига указывается

стрелкой на условном обозначении регистра) путём продвижения кодовой комбинации с каждым тактовым импульсом от разряда к разряду. Следовательно, для записи N -разрядного слова необходимы N -импульсов сдвига.

Считывание информации последовательным кодом осуществляется, как и запись, поразрядным сдвигом записанной кодовой комбинации к выходу с каждым тактовым импульсом. Следовательно, для считывания N -разрядного слова необходимы N импульсов сдвига. Считывание информации параллельным кодом происходит в паузе между последним импульсом сдвига одного цикла записи и первым импульсом сдвига другого цикла записи, т.е. в интервале времени, когда на C -входах триггеров нулевой уровень, и они находятся в режиме хранения.

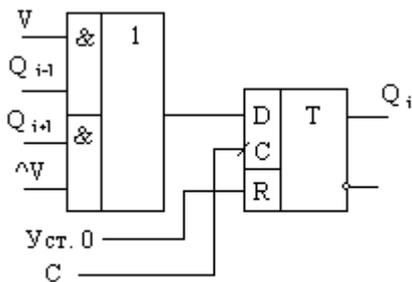


Рис.2. Разряд реверсивного регистра

Таким образом, с помощью регистра сдвига можно осуществлять преобразование информации из последовательной формы представления в параллельную. Очевидно, если предусмотрена запись информации параллельным кодом, то можно преобразовывать информацию из параллельной формы представления в последовательную.

Регистры сдвига могут быть построены и на триггерах одноступенчатой структуры. В этом случае в каждом разряде регистра нужно использовать два триггера, которые управляются двумя сдвинутыми во времени тактовыми импульсами. Наличие двух триггеров в одном разряде позволяет поразрядно продвигать информацию в регистре от входа к выходу. Если бы в регистре были применены одноступенчатые триггеры по одному на разряд, то правило работы регистра было бы нарушено: при первом же импульсе сдвига

информация, записавшись в первый разряд, перешла бы во второй, затем в третий и т.д.

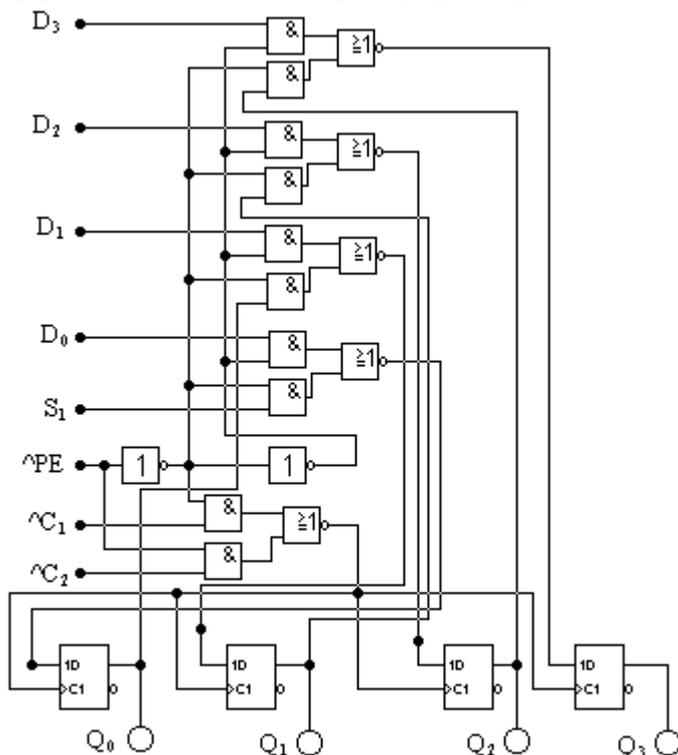


Рис.3. Схема четырёхразрядного сдвигового регистра K155IP1

триггеров. Регистр имеет два тактового входа $\wedge C_1$ и $\wedge C_2$. От любого из пяти входов данных код поступит на выходы синхронно с отрицательным перепадом, поданным на выбранный тактовый вход.

Вход разрешения параллельной загрузки $\wedge PE$ служит для выбора режима работы регистра. Если на вход $\wedge PE$ даётся напряжение высокого уровня, разрешается работа тактовому входу $\wedge C_2$. В момент прихода на этот вход отрицательного перепада тактового импульса в регистр загружаются данные от параллельных входов D_0 - D_3 .

Если на вход $\wedge PE$ подано напряжение низкого уровня, разрешается работа тактовому входу $\wedge C_1$. Отрицательные фронты последовательности тактовых импульсов сдвигают данные от последовательного входа S_1 на выход Q_0 , затем на Q_1 , Q_2 и Q_3 , т.е. вправо. Сдвиг данных по регистру влево получится, если соединить выход Q_3 и вход D_2 , выход Q_2 и вход D_1 , Q_1 и D_0 по схеме рис. 2. Регистр можно перевести в параллельный режим, подав на вход $\wedge PE$ напряжение высокого уровня. Напряжение на входе $\wedge PE$ можно менять только, если на обоих тактовых входах уровни низкие. Однако если на входе $\wedge C_1$ напряжение низкого уровня, перемена сигнала на входе $\wedge PE$ от низкого уровня к высокому не меняет состояния выходов.

Реверсивные регистры сдвига объединяют в себе свойства регистров прямого и обратного сдвига. Строятся они по тем же схемотехническим принципам, что и рассмотренные регистры, но с использованием дополнительных логических элементов в межразрядных связях. Указанная особенность реверсивного регистра показана на примере i -разряда (рис. 2), состоящего из D -триггера и логической схемы, на входы которой поданы: Q_{i-1} -сигнал с выхода младшего разряда, Q_{i+1} -сигнал с выхода старшего разряда, V -сигнал управления направлением сдвига: $V=1$ –вправо, $V=0$ –влево.

Существуют многорежимные регистры. Их входные и выходные линии данных объединены и образуют так называемый порт данных. Это означает, что от шины данных процессора приходит один провод (а не два), который по команде служит или входным или выходным. Число сигнальных входов и выходов микросхемы за счёт портовой организации можно уменьшить в два раза.

Подробное рассмотрение регистров проведём на примере микросхемы K155IP1 – четырёхразрядный сдвиговый регистр, принципиальная схема которого приведена на рис.3. Он имеет последовательный вход данных S_1 , четыре параллельных входа D_0 – D_3 , а также четыре выхода Q_0 – Q_3 от каждого из

2.2. МЕТОДИКА ВЫПОЛНЕНИЯ

2.1. Включите необходимое программное обеспечение.

2.2. Соберите схему простейшего сдвигающего регистра на D-триггерах по рис. 4, воспользовавшись триггерами из контейнера **Se-sequential**.

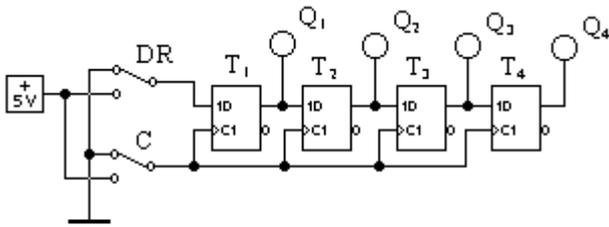


Рис.4. Сдвигающий регистр на D-триггерах

Спланируйте эксперимент и постройте временные диаграммы $Q_i = f(D, R, C)$.

2.3. Соберите схему для исследования универсального регистра по рис. 5. Для этого используется интегральная микросхема 74198 – восьмибитовый регистр сдвига из контейнера **Sequential** в шаблоне SHIFT REGS.

На этом рисунке:

D_a, \dots, D_h - входы для параллельной записи в регистр байта

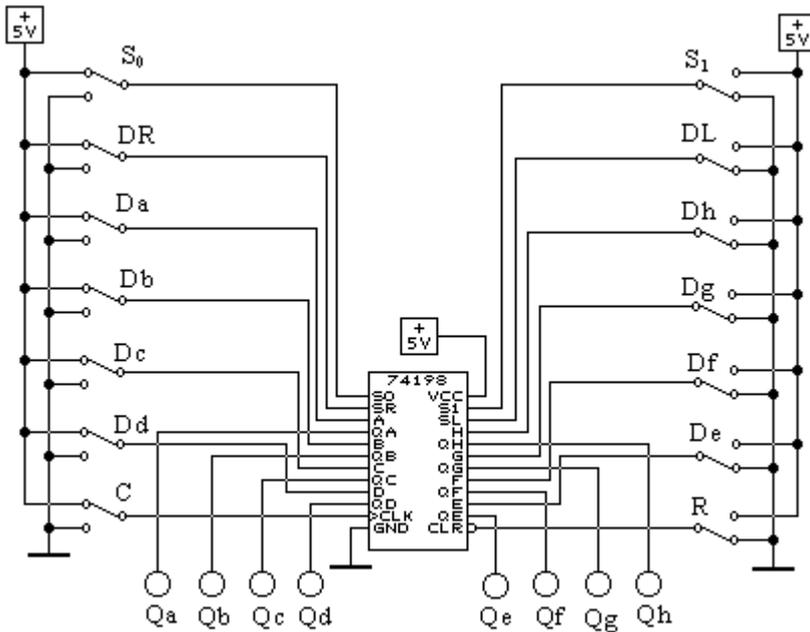


Рис.5. Схема для исследования универсального регистра

$a b c d e f g h$;

Q_a, \dots, Q_h - выходы регистра;

C - вход синхронизации;

R - вход начальной установки всех разрядов в нуль;

DR, DL - входы для последовательного ввода информации слева и справа;

S_1, S_0 - входы, задающие режим работы регистра: параллельная запись,

хранение, сдвиги влево и вправо. Для данной схемы спланировать эксперимент и составить таблицу,

описывающую поведение регистра при различных режимах (S_1 и S_0), различных значениях на входах регистра (D_a, \dots, D_h ,

DR и DL). Определить в каких режимах на выполнение операции влияет C, какое действие оказывает R. Выясните необходимые состояния управляющих сигналов для реализации различных режимов работы регистра – запись,

хранение, сдвиг влево, сдвиг вправо.

Для режимов сдвига вправо и влево построить временные диаграммы.

2.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое регистр, каково его назначение?
2. Что определяет разрядность записываемого в регистр числа? Что такое быстродействие регистра?
3. Как реализуется преобразование параллельного кода записанного числа в последовательный код?
4. Как реализуется операция сдвига числа в регистре в ту или иную сторону?
5. Поясните по принципиальной схеме рис. 3 работу регистра K155ИР1.

Контрольные вопросы ПЗ2 (ПК-4):

1. Назовите основные компоненты, из которых состоит настольная или портативная компьютерная система.
2. Что такое системная шина и какова ее роль в компьютерной системе?
3. Что такое контроллер?
4. Назовите и кратко охарактеризуйте основные типы портов в компьютерной системе.
5. Какие области памяти используются в операциях ввода-вывода?
6. Каким образом контроллер устройства информирует процессор об окончании операции ввода-вывода?
7. Как организована обработка прерываний?
8. Что такое вектор прерываний?
9. Как система обрабатывает ситуацию, когда при обработке прерывания возникает другое прерывание?
10. Что такое программируемое прерывание (ловушка) и когда оно генерируется?
11. Каково назначение прерывания по таймеру?
12. Что такое опрос устройств и какова его роль в системе?

13. Чем синхронный ввод-вывод отличается от асинхронного?
14. Что такое таблица состояния устройств и для чего она используется операционной системой?
15. В чем отличие основной памяти от внешней памяти?
16. Опишите кратко устройство жесткого диска.
17. Что такое ассоциативная память (кэш) и как она позволяет оптимизировать обращение к более медленным видам памяти?
18. Назовите основные виды внешней памяти.
19. Какова цель введения привилегированного режима и какие действия в нем разрешены, по сравнению с обычным режимом исполнения программ?
20. Каким образом система определяет текущий режим исполнения?
21. Что такое системный вызов и как они используются для организации ввода-вывода?
22. Что такое регистр базы и регистр границы и как они используются для защиты памяти?
23. Что такое таймер и как прерывания по таймеру используются операционной системой?
24. Как организована защита памяти в системах с теговой архитектурой?
25. Что такое дескриптор и для чего он предназначен?

ПРАКТИЧЕСКАЯ РАБОТА № 3.

Организация памяти ЭВМ

3.1. КРАТКАЯ ТЕОРИЯ

Цифровым счётчиком импульсов называют последовательный цифровой узел, который осуществляет счёт поступающих на его вход импульсов. Результат счёта формируется счётчиком в заданном коде и может храниться необходимое время.

Счётчики строят на Т-триггерах и TV-триггерах с применением при необходимости логических элементов в цепях межразрядных связей. Количество триггеров N должно быть таким, чтобы множество внутренних состояний счётчика 2^N было не меньше максимального числа импульсов, которое должно быть посчитано. С приходом очередного счётного импульса изменяется состояние счётчика, которое в заданном коде отображает результат счёта. Если количество счётных импульсов не ограничивать, то счётчик будет работать в режиме деления их числа на коэффициент (модуль) счёта $K_{сч}$, равный 2^N . Через каждые 2^N импульсов он будет возвращаться в начальное состояние и снова считать импульсы. Если необходимый коэффициент счёта не равен 2^N , применяют различные способы сокращения числа внутренних состояний счётчика.

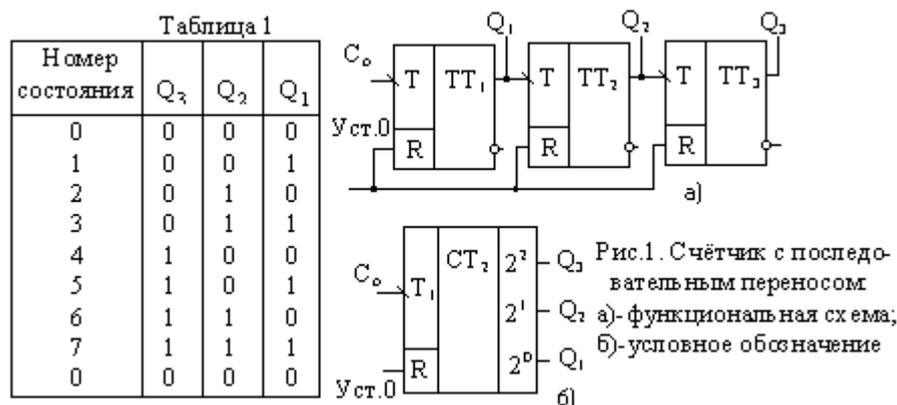


Рис.1. Счётчик с последовательным переносом

а) - функциональная схема;
б) - условное обозначение

Счётчики можно классифицировать по ряду признаков. По направлению счёта их делят на суммирующие (с прямым счётом), вычитающие (с обратным счётом) и реверсивные. По способу организации переноса различают счётчики с последовательным, параллельным и комбинированным (параллельно-последовательным) переносом.

Рассмотрим пример реализации трёхразрядного суммирующего счётчика в коде 4-2-1 с последовательным переносом. Порядок смены состояний счётчика

задан табл.1. Как следует из таблицы, с приходом очередного счётного импульса к содержимому счётчика прибавляется единица. При этом увеличивается на единицу номер состояния, являющийся десятичным эквивалентом соответствующего данному состоянию двоичного числа.

Изменение состояния каждого последующего разряда происходит при изменении состояния предыдущего разряда от 1 к 0. Это означает, что всякий раз, когда данный триггер в счётчике переходит из состояния 1 в состояние 0, на его выходе должен формироваться сигнал переноса, вызывающий срабатывание следующего триггера. Если же данный триггер переходит из 0 в 1, то сигнала переноса на его выходе не должно быть. Из табл. 1 также следует, что триггер первого, самого младшего разряда, должен менять своё состояние каждый раз с приходом очередного счётного импульса, а триггер каждого последующего разряда – вдвое реже триггера предыдущего разряда.

Описанный порядок смены состояний счётчика и характер процесса их установления могут быть реализованы, если счётчик будет построен на последовательно соединённых Т-триггерах. Каждый последующий разряд при этом будет переключаться сигналом переноса, формируемым на выходе предыдущего разряда. Счётные импульсы должны быть поданы на вход триггера самого младшего разряда. Счётчики, построенные таким образом, получили название счётчиков с последовательным переносом. Пример трёхразрядного счётчика на Т-триггерах двухступенчатой структуры приведён на рис. 1. Для установки исходного состояния служит шина «Уст.0», которой объединены R-входы всех триггеров. Нулевое состояние триггеров устанавливается подаваемым по этой шине положительным импульсом напряжения между уровнями 0 и 1. На левом поле условного графического обозначения счётчика (рис. 1,б) показано, что его входом является T_1 -вход первого разряда, а на правом поле указан «вес» каждого разряда.

Вычитающий счётчик с последовательным переносом имеет обратный порядок смены состояний: с приходом очередного счётного импульса содержащееся в счётчике число уменьшается на единицу (табл. 2). Другая особенность вычитающего счётчика – триггер каждого последующего разряда переключается в противоположное состояние при изменении уровня на выходе триггера предыдущего разряда от 0 к 1, т.е. при сигнале займа, обратном сигналу переноса в суммирующем счётчике. Строится вычитающий счётчик так же, как суммирующий, но с тем отличием, что со входом каждого последующего триггера в отличие от рис.1 соединяется инверсный выход предыдущего триггера.

Из работы трёхразрядного счётчика следует, что в наихудшем случае новое его состояние устанавливается с задержкой, равной утроенной задержке переключения одного триггера, что вызвано последовательным во времени распространением сигнала переноса через все разряды счётчика. Таким образом, в счётчике с последовательным переносом неэффективно используется быстродействие триггеров, особенно при большом числе разрядов. В этом состоит существенный недостаток счётчиков с последовательным переносом, из-за которого область их применения ограничивается цифровыми устройствами с небольшим числом разрядов и невысоким быстродействием.

Таблица 2

№ состояния	Q ₃	Q ₂	Q ₁
7	1	1	1
6	1	1	0
5	1	0	1
4	1	0	0
3	0	1	1
2	0	1	0
1	0	0	1
0	0	0	0
7	1	1	1

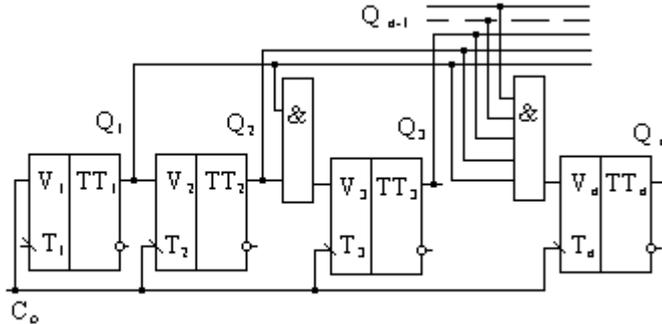


Рис.2. Счётчик с параллельным переносом

Один из широко применяемых способов ускорения переноса в счётчике основан на введении логических элементов, с помощью которых достигается возможность одновременного (параллельного) формирования сигнала переноса для всех разрядов. Для реализации этого способа применяют TV-триггеры. На Т-входы всех триггеров одновременно подаются счётные импульсы, а на V-вход каждого триггера поступает сигнал переноса, формируемый логической схемой в виде уровня 1. Триггеры, на V-входе которых имеется сигнал переноса, одновременно переключаются с приходом очередного счётного импульса, и, таким образом, устанавливается новое состояние счётчика. Для определения вида цепи переноса обратимся к уже рассмотренной табл.1. Из неё следует, что первый разряд, как и в счётчике с последовательным переносом, должен быть построен на Т-триггере. Если применяется TV – триггер, то его V-вход следует соединить с Т-входом.

Второй триггер перебрасывается в противоположное состояние счётным импульсом при наличии 1 на выходе первого триггера, а третий триггер перебрасывается при наличии 1 на выходах двух предыдущих триггеров. Обобщая эту закономерность на случай N-разрядного счётчика получим, что каждый последующий триггер должен переключаться под воздействием счётного импульса при наличии 1 на выходах всех предыдущих триггеров. Следовательно, для формирования сигнала переноса в каждый разряд счётчика необходимо включить элемент И и соединить его входы с прямыми выходами всех предыдущих разрядов, а выход – с V-входом триггера данного разряда. Пример суммирующего счётчика с параллельным переносом на TV – триггерах приведён на рис.2. Быстродействие этого счётчика выше, чем счётчика с последовательным переносом, поскольку оно равно быстродействию переноса одного разряда. Недостаток – необходимость включения в схему элемента И с нарастающим от разряда к разряду числом входов. Это нарушает регулярность структуры и ограничивает возможность наращивания его схемы. Частично этот недостаток можно устранить при использовании триггеров с входной логикой.

Многие серии микросхем содержат JK-триггеры с входной логикой. Для преобразования JK-триггера в TV-триггер необходимо объединить входы J и K в один, это и будет V-вход. У триггера с тремя конъюнктивно связанными J-входами и тремя конъюнктивно связанными K-входами могут быть образованы, следовательно, три конъюнктивно связанные V-входа. При реализации счётчика на таких триггерах исключаются дополнительные логические элементы в цепях переноса. Однако ограничение в числе разрядов остаётся. На таких триггерах можно построить лишь четырёхразрядный счётчик (рис. 3).

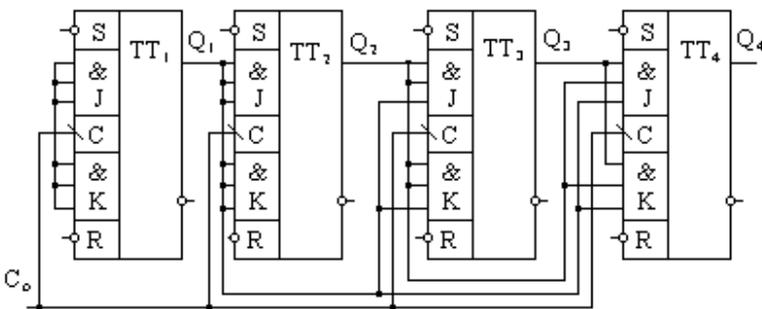


Рис.3. Счётчик на JK-триггерах с входной логикой

Вычитающий счётчик с параллельным переносом строится так же, как и суммирующий, но сигналы переноса снимаются с инверсных относительно используемых в суммирующем счётчике выходов триггеров.

Реверсивный счётчик, объединяющий возможности суммирующего и вычитающего, строится таким образом, чтобы обеспечивалось управление направлением счёта с помощью сигналов разрешения на реализацию операций сложения C_с и вычитания C_в. Поэтому его схема содержит дополнительную комбинационную часть, выполняющую указанную функцию.

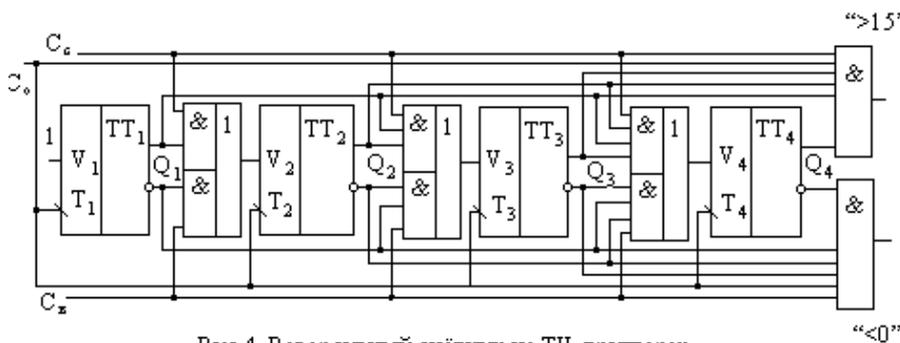


Рис.4. Реверсивный счётчик на TV-триггерах

Нередко счётчики с параллельным переносом, выпускаемые в виде микросхем, имеют помимо основных выходов – дополнительные, как это показано, например, на рис. 4. На одном из выходов, обозначенном «>15», сигнал 1 появляется при заполнении счётчика единицами, т.е. когда он перешёл в состояние с номером 15. Следовательно, на этом выходе формируется сигнал переноса в следующий счётчик. На другом выходе, обозначенном «<0», сигнал появляется при заполнении счётчика нулями и является сигналом займа в следующий счётчик в режиме вычитания.

Реверсивный счётчик можно построить и на Т-триггерах (рис. 5). Как и в рассмотренном ранее суммирующем счётчике, счётные импульсы поступают на Т-вход триггера через логические элементы только в том случае, если на логических элементах имеются сигналы разрешения с выходов предыдущих разрядов.

В счётчике на рис. 5,а для счётных импульсов предусмотрены два входа. Если счётчик должен работать в режиме прямого счёта, импульсы следует подавать на вход «+1», в режиме обратного счёта – на вход «-1». При использовании такого счётчика в качестве реверсивного с одним источником импульсов необходимо предусмотреть внешнее устройство коммутации счётных импульсов на суммирующий «+1» либо на вычитающий «-1» входы.

Вариант такой коммутирующей приставки к счётчику приведён на рис. 5,б. При подаче положительных импульсов на S-вход RS-триггера на его прямом выходе установится единичный уровень, который откроет элемент 1 для счётных импульсов C₀. Счётчик будет работать в режиме сложения. Если подать положительный импульс на R-вход триггера, откроется для счётных импульсов элемент 2, и счётчик будет работать в режиме вычитания.

3.2. МЕТОДИКА ВЫПОЛНЕНИЯ

- 2.1. Включите необходимое программное обеспечение.
- 2.2. Соберите схему простейшего

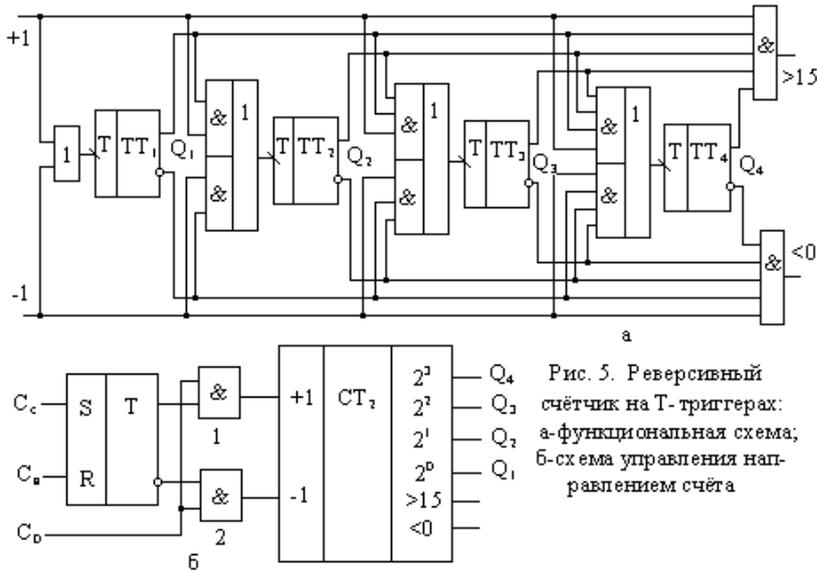


Рис. 5. Реверсивный счётчик на Т-триггерах: а-функциональная схема; б-схема управления направлением счёта

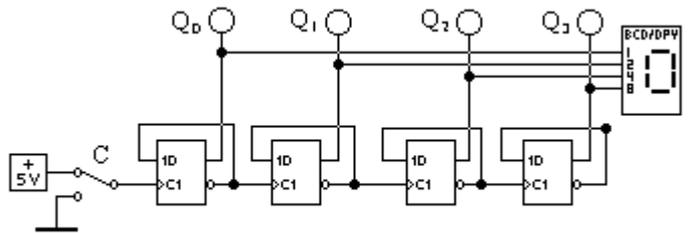


Рис.6. Суммирующий счётчик с последовательным переносом

суммирующего счётчика с последовательным переносом по рис. 6. Для этого следует воспользоваться синхронизируемыми положительным фронтом D-триггерами из контейнера **Sequential**. Для индикации состояния счётчика в виде десятичного числа используется семисегментный индикатор из контейнера **Indicators**. Спланируйте эксперимент и постройте временные диаграммы $Q_i = f(C)$ для этой схемы.

2.3. Соберите схему простейшего вычитающего счётчика (рис. 7), отличие которого от схемы на рис. 6 в том, что входы синхронизации C1 у 2-го, 3-го и 4-го триггеров соединены не с инверсными, а с прямыми выходами 1-го, 2-го и 3-го триггеров. Спланируйте эксперимент и постройте временные диаграммы $Q_i = f(C)$ для этой схемы.

2.3. На рис. 8 приведена схема для исследования универсального двоично-десятичного счётчика. Соберите эту схему, используя счётчик **74192 (4-Bit Up/ Down Counter)** из контейнера **Sequential** в шаблоне **CO-UNTERS**. Для индикации состояния счётчика в виде десятичного числа используется семисегментный индикатор из контейнера **Indicators**.

Спланируйте эксперимент и постройте временные диаграммы для различных режимов работы этого счётчика: установка в 0- (R), параллельная запись -(Load), +1 и -1. Постройте графы переходов для счётчика, установленного в начальные состояния 10-(A), 11-(B), 12-(C), 13-(D), 14-(E), 15-(F) при сигналах +1 и -1.

На рис. 8 использованы обозначения:

- DCBA – входы для параллельной записи в счётчик двоичного числа;
- Load – (инверсный сигнал) – синхронизация параллельной записи числа в счётчик;
- R – установка счётчика в состояние 0000;

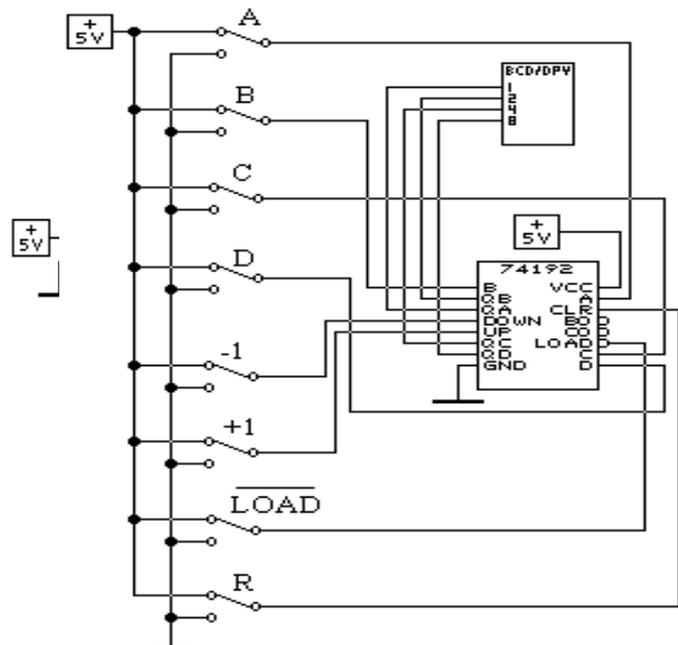


Рис.8. Исследование универсального двоично-десятичного счётчика

+1 – прибавление к содержимому счётчика единицы (положительный фронт);

-1 – вычитание из содержимого счётчика единицы (положительный фронт).

2.4. Получите у преподавателя задание на проектирование счётчика с параллельным переносом на D-триггерах и нетрадиционным порядком счёта:

1 вариант: счёт – 1,3,7,4,0; 2 вариант: счёт – 0,3,6,4,0;

3 вариант: счёт – 1,3,7,4,8; 4 вариант: счёт – 7,5,3,1,0;

5 вариант: счёт – 0,3,1,4,2; 6 вариант: счёт – 5,7,3,2,0;

7 вариант: счёт – 3,2,1,5,7; 8 вариант: счёт – 2,3,4,1,5.

Соберите схему спроектированного счётчика и исследуйте её на соответствие вашему варианту задания. Постройте граф переходов.

3.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните принципы построения суммирующего и вычитающего счётчиков по табл. 1 и 2.
2. Как реализуется параллельное формирование сигнала переноса во всех разрядах счётчика?
3. Поясните построение и работу реверсивного счётчика.
4. Как функционируют выходы “>15” и “<0” реверсивного счётчика, каково их практическое применение?

Контрольные вопросы ПЗЗ (ПК-4):

1. Какие операции определяет понятие «обращение к ЗУ»?
2. Какие единицы измерения используются для указания ёмкости запоминающих устройств?
3. В чем отличие между временем доступа и периодом обращения к запоминающему устройству?
4. Чем вызвана необходимость построения системы памяти по иерархическому принципу?
5. Что включает в себя понятие «локальность по обращению»?
6. Благодаря чему среднее время доступа в иерархической системе памяти определяется быстродействующими видами ЗУ?
7. Что в иерархической системе памяти определяют термины «промах» и «попадание»?
8. На какие вопросы необходимо ответить, чтобы охарактеризовать определенный уровень иерархической памяти?
9. Какие виды запоминающих устройств может содержать основная память?
10. Охарактеризуйте возможные варианты построения блочной памяти.
11. Какие возможности по сокращению времени доступа к информации предоставляет блочная организация памяти?
12. Чем обусловлена эффективность расслоения памяти?
13. Какая топология запоминающих элементов лежит в основе организации полупроводниковых ЗУ?
14. Какое минимальное количество линий должен содержать столбец ИМС памяти?
15. Поясните назначение управляющих сигналов в микросхеме памяти.
16. Чем отличаются страничный, быстрый страничный и пакетный режимы доступа к памяти?
17. Чем обусловлена необходимость регенерации содержимого динамических ОЗУ?
18. Охарактеризуйте основные сферы применения статических и динамических ОЗУ.
19. Какое влияние на асинхронный режим работы памяти оказывает синхронный характер работы контроллера памяти?
21. Какой вид ПЗУ обладает наиболее высокой скоростью перепрограммирования?
22. Какими методами обеспечивается энергонезависимость ОЗУ?
23. В чем состоит различие между режимами стандартной и запаздывающей записи в статических ОЗУ?
24. В чем проявляется специфика ОЗУ, предназначенных для видеосистем??

ПРАКТИЧЕСКАЯ РАБОТА № 4.

Организация внешней памяти ЭВМ. Алгоритмы умножения чисел. Алгоритмы умножения чисел

4.1. КРАТКАЯ ТЕОРИЯ

Сумматор по модулю 2 – цифровой узел с m входами и одним выходом, работающий в соответствии со следующим правилом: сигнал 1 появляется на его выходе всякий раз, когда в наборе входных сигналов содержится нечётное число 1. Поэтому этот узел ещё называют схемой проверки на чётность. В частном случае при числе входов равном 2,

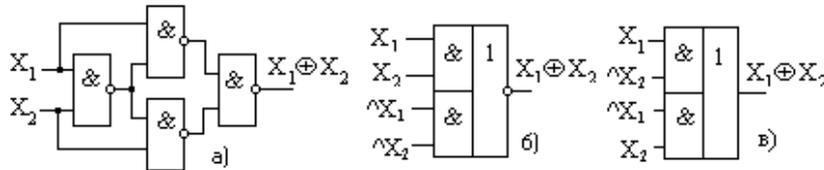


Рис.1. Сумматор по модулю 2: а) - функциональная схема на элементах И-НЕ; б) - на элементах И-ИЛИ-НЕ; в) - на элементах И-ИЛИ

сумматор по модулю 2 выполняет функцию логического элемента «Исключающее ИЛИ»: на выходе 1 будет только при 1 на одном из входов. Функциональные схемы двухвходовых сумматоров по модулю 2 приведены на рис. 1. Реализуемая здесь логическая функция $\Sigma = A \oplus B = A \oplus B$, где символ \wedge есть признак инверсии. При суммировании двух одноразрядных чисел имеем: $0+0=0$, $1+0=1$, $0+1=1$ и $1+1=10$. В последнем случае число 10 (в десятичной записи это 2) оказалось двоичным двухразрядным. Появившаяся в старшем разряде суммы единица называется единицей переноса.

К схеме «Исключающее ИЛИ» несложно добавить выход переноса, т. е. генератор старшего разряда. Для этого оба суммируемых одноразрядных числа следует подать на схему И, выход которой даст необходимый старший разряд переноса $1 \cdot 1 = 1$. На рис. 2,а показана реализация схемы суммирования двух одноразрядных чисел, состоящая из элементов «Исключающее ИЛИ» и «И». Схема имеет два выходных сигнала: суммы Σ и переноса C . Такая схема называется полусумматором. Таблица состояний полусумматора показана на рис. 2,б.

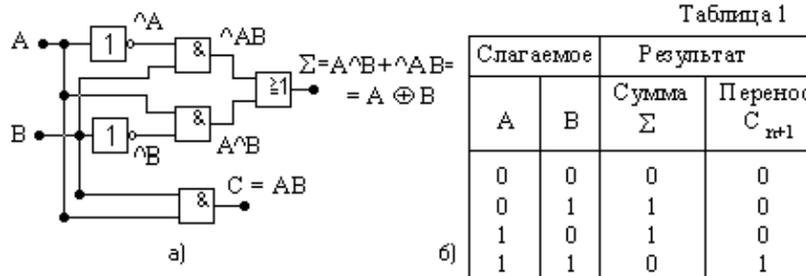


Рис.2. Полусумматор (а) и таблица его состояний (б)

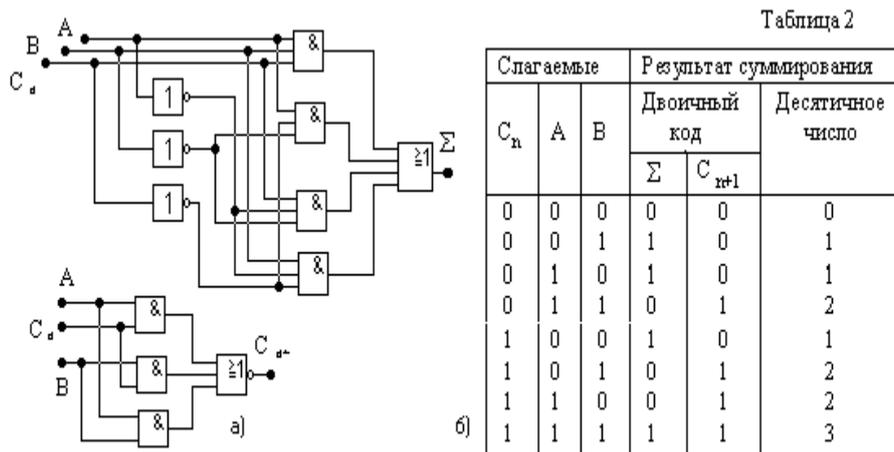


Рис.3. Полный сумматор (а) и таблица его состояний (б)

Полный сумматор должен иметь вход для сигнала переноса C_n от со-седнего младшего разряда . Схема полного сумматора двух одноразрядных слов показана на рис. 3,а, а таблица истинности – на рис. 3,б. В последнем

столбце таблицы результаты суммирования даны в десятичной форме. В присутствии входной единицы переноса C_n сумма чисел A и B увеличивается на единицу.

Полные сумматоры многоразрядных чисел составляются из одноразрядных и могут складывать многоразрядные числа с двумя способами переноса: параллельным или последовательным. При параллельном переносе реализуется одновременное поразрядное сложение слов, при последовательном переносе - формирование сигналов переноса для старших разрядов происходит после выполнения операции сложения в младших разрядах. Поэтому время выполнения операции сложения N -разрядных чисел равно $N \cdot T_3$, где T_3 - задержка на одном одноразрядном сумматоре.

4.2. МЕТОДИКА ВЫПОЛНЕНИЯ

2.1. Включите необходимое программное обеспечение.

2.2. Соберите схему одноразрядного сумматора по рис. 4,а, экспериментально получите таблицы истинности для суммы S_i и переноса P_i . Создайте из схемы рис. 4,а подсхему sm одноразрядного сумматора по рис. 4,б.

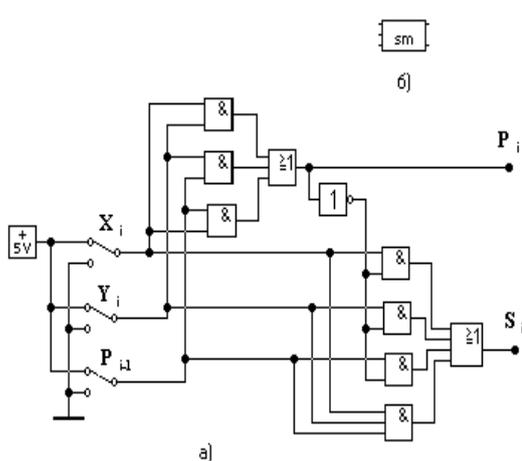


Рис.4. Одноразрядный сумматор: а) - функциональная схема; б) - подсхема

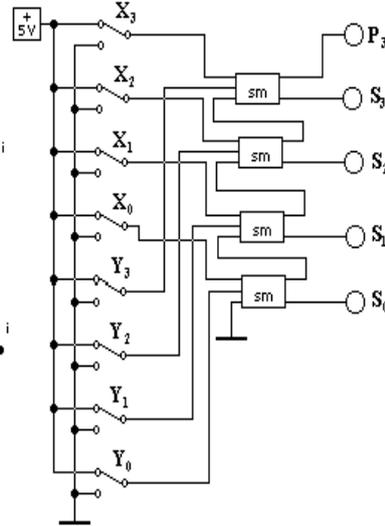


Рис.5. Четырёхразрядный сумматор

2.3. Соберите схему четырёхразрядного сумматора по рис.5 на базе подсхемы sm и исследуйте его работу при различных значениях двоичных чисел X и Y . Результаты оформить в виде табл. 3.

2.4. Соберите схему для исследования работы арифметико-логического устройства АЛУ по рис. 6. Для этого следует выбрать из контейнера **Combinational** (шаблон **ARITHMETIC**) микросхему **74181 ALU** -арифметико-логическое устройство. Это АЛУ выполняет поразрядные логические операции (при $M=1$) над четырёхразрядными словами A и B , а так же арифметические операции (при $M=0$). На результат логических операций не оказывает влияния значение P_0 -перенос от младших разрядов. Код выполняемой операции задаётся значениями $S_3S_2S_1S_0$.

Обратите внимание на то, что входы АЛУ для операндов A и B - инверсные. Поэтому на переключателях A_i , B_i значению логической единицы в АЛУ соответствуют верхние положения ключей. Выходы $F_3F_2F_1F_0$ - так же инверсные. Для удобства исследования АЛУ выходы подключены к индикаторам через инверторы.

По клавише «F1» можно раскрыть таблицу истинности для АЛУ. В этой табл. 4 все выражения записаны для прямых значений операндов A , B и результата F . В столбце «**LOGIC FUNCTION**» символом « $+$ » обозначена поразрядная логическая операция «ИЛИ», символом « \oplus » обозначена поразрядная логическая операция «ИСКЛЮЧАЮЩЕЕ ИЛИ». В столбцах «**ARITHMETIC OPERATIONS**» словами «**PLUS**» и «**MINUS**»

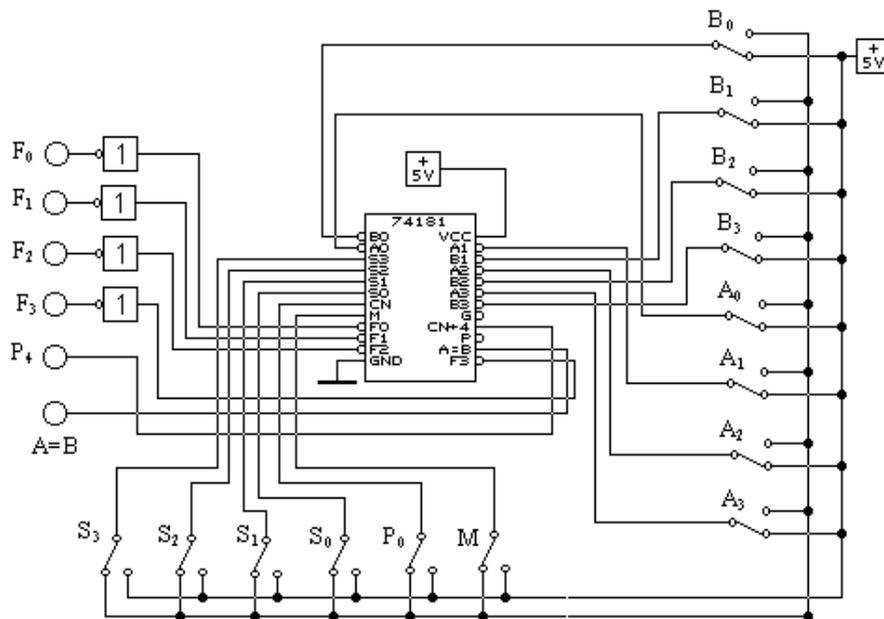


Рис.6. Схема для исследования арифметико-логического устройства

обозначены соответствующие арифметические операции. Символами «Н», «L» в табл. 4 обозначены логические «1» и «0».

2.5. Задайте $M=1$. Из имеющихся 16 логических операций проверьте правильность реализации любых 4 для двух произвольных комбинаций операндов А и В в каждой из них. Из восьми наборов операндов А и В в двух случаях эти операнды должны совпадать для проверки истинности сигнала $A=B$ на выходе АЛУ. Составьте табл. 5 результатов проверки, в которой укажите теоретические значения выходных величин для заданной логической функции и их экспериментальные значения. Сделайте вывод.

2.6. Задайте $M=0$ при отсутствии переноса ($P_0=0$). Затем повторите действия п. 2.5. для проверки реализации арифметических операций, сведя результаты в табл. 6. Сделайте вывод.

2.7. Задайте $M=1$ при наличии переноса ($P_0=1$). Затем повторите действия п. 2.5. для проверки реализации арифметических операций, сведя результаты в табл. 7. Сделайте вывод.

Таблица 4

Selection S3 S2 S1 S0	M = H LOGIC FUNCTIONS	M = L Arithmetic Operations	
		Cn = L (no carry)	Cn = H (with carry)
0 0 0 0	$F = \text{^}A$	F = A MINUS 1	F = A
0 0 0 1	$F = \text{^}(AB)$	F = AB MINUS 1	F = AB
0 0 1 0	$F = \text{^}A + B$	F = A^B MINUS 1	F = A^B
0 0 1 1	F = 1	F = MINUS 1 (2's comp)	F = Zero
0 1 0 0	$F = \text{^}(A+B)$	F = A PLUS (A+^B)	F = A PLUS (A+^B) PLUS 1
0 1 0 1	$F = \text{^}B$	F = AB PLUS (A+^B)	F = AB PLUS (A+^B) PLUS 1
0 1 1 0	$F = \text{^}(A''+''B)$	F = A MINUS B MINUS 1	F = A MINUS B
0 1 1 1	$F = A+\text{^}B$	F = A + ^B	F = (A+^B) PLUS 1
1 0 0 0	$F = \text{^}AB$	F = A PLUS (A+B)	F = A PLUS (A+B) PLUS 1
1 0 0 1	$F = A''+''B$	F = A PLUS B	F = A PLUS B PLUS 1
1 0 1 0	F = B	F = A^B PLUS (A+B)	F = AB PLUS (A+B) PLUS 1
1 0 1 1	F = A + B	F = (A+B)	F = (A+B) PLUS 1
1 1 0 0	F = 0	F = A PLUS A	F = A PLUS A PLUS 1
1 1 0 1	F = A^B	F = AB PLUS A	F = AB PLUS A PLUS 1
1 1 1 0	F = AB	F = A^B PLUS A	F = A^B PLUS A PLUS 1
1 1 1 1	F = A	F = A	F = A PLUS 1

4.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Поясните работу схем сумматоров по модулю 2 на рис. 1.
2. Опишите работу схемы полусумматора и таблицы его состояний по рис. 2.
3. Поясните работу схемы полного сумматора и таблицу его истинности по рис. 3.
4. Опишите работу функциональных схем по рис. 4 и рис. 5.

Контрольные вопросы ПЗ4 (ПК-4):

1. Чем объясняется тенденция размещения стека в области старших адресов основной памяти?
2. Какая информация хранится в указателе стека?
3. Поясните назначение маски в ассоциативном запоминающем ЗУ.
4. Как реализуется запись новой информации в ассоциативное ЗУ?
5. Какие виды поиска можно осуществлять в ассоциативном ЗУ?
6. Поясните назначение и логику работы кэш-памяти.
7. Какие проблемы порождает включение в иерархию ЗУ кэш-памяти?
8. Чем обусловлено разнообразие методов отображения основной памяти на кэш-память?
9. Какому требованию должен отвечать «идеальный» алгоритм замещения содержимого кэш-памяти?
10. Какими методами обеспечивается согласованность содержимого основной и кэш-памяти?
11. Чем обусловлено введение дополнительных уровней кэш-памяти?
12. Какие факторы влияют на выбор емкости кэш-памяти и размера блока?
13. Как соотносятся характеристики обычной и дисковой кэш-памяти?
14. Какими средствами обеспечивается виртуализация памяти?
15. Существует ли ограничение на размер виртуального пространства?
16. Что определяет объем страничной таблицы?
17. Какими приемами достигают сокращения объема страничных таблиц?
18. Какие алгоритмы замещения используются при загрузке в основную память новой виртуальной страницы?
19. Поясните назначение буфера быстрого преобразования адреса (TLB).

20. Чем мотивируется разбиение виртуальных секторов на страницы?
21. Какая часть виртуального адреса остается неизменной при его преобразовании в физический адрес?
22. Чем обусловлена необходимость защиты памяти?
23. Охарактеризуйте состав операционных устройств, входящих в АЛУ. Из каких соображений и каким образом он может изменяться?
24. Поясните понятие «операционные устройства с жесткой структурой». В чем заключается жесткость их структуры? Каковы их достоинства и недостатки?
25. Чем обусловлено название операционных устройств с магистральной структурой? Сравните магистральные структуры с жесткими структурами, выделяя достоинства, недостатки и область применения.
26. Дайте развернутую характеристику классификации операционных устройств с магистральной структурой. Поясните достоинства и недостатки «минимального» и «максимального» вариантов.
27. Поясните функциональный состав параллельного операционного блока магистрального ОПУ. Каким образом можно минимизировать количество внешних связей этого блока? Ответ сопроводите конкретным примером.
28. Чем обусловлена Специфика целочисленного сложения и вычитания? Какую роль играет в них дополнительный код? К чему бы привел отказ от дополнительного кода? Ответ поясните на примерах. Как выявляется переполнение в этих операциях?
29. Сформулируйте достоинства, недостатки и область применения четырех вариантов целочисленного «традиционного» умножения. Как учитываются знаки сомножителей?
30. Охарактеризуйте суть двух групп логических методов ускорения умножения.
31. Попарно сравните алгоритм Бута, модифицированный алгоритм Бута, алгоритм Лемана.
32. Разработайте алгоритм умножения с обработкой за шаг трех разрядов множителя.
33. Поясните суть аппаратных методов ускорения умножения, выделив три возможных подхода.
34. В чем заключается основная идея древовидных умножителей? Каковы особенности их организации?
35. С какой целью и каким образом выполняется конвейеризация матричных и древовидных умножителей? Приведите (и поясните) конкретные примеры.
36. На конкретном примере объясните, как можно нарастить разрядность аппаратного умножителя.
37. Сравните организацию целочисленного деления с восстановлением остатка и без восстановления остатка. Как учитываются при делении знаки операндов?
38. Обоснуйте возможность совмещения структур умножителя и делителя. Опишите объединенную структуру.
39. Сформулируйте четыре пути ускорения целочисленного деления. Сравните между собой их возможную реализацию.
40. Какие из операций с плавающей запятой считаются наиболее сложными? Ответ обоснуйте на конкретных примерах.
41. В чем состоит специфика умножения с плавающей запятой? Поясните эту специфику на примере.

ПРАКТИЧЕСКАЯ РАБОТА № 5.

Реализация команд пересылки и операций со стекком. Выполнение арифметических и логических команд

5.1. КРАТКАЯ ТЕОРИЯ

В цифровых устройствах часто возникает задача передачи цифровой информации от различных источников к одному приёмнику. Для этого на входе канала устанавливается устройство, называемое мультиплексором

(переключателем MS), которое согласно коду адреса подключает к выходу один из источников информации. Например, из четырёх источников D0, D1, D2 и D3, которые подключены к информационным входам мультиплексора, необходимо выбрать один. Для этого должен быть указан номер информационного входа. Обычно он задаётся двоичным кодом на управляющих входах мультиплексора. Для MS с двумя информационными входами достаточно одного управляющего входа X0, для MS с четырьмя информационными входами – достаточно двух - X1 и X0 и т.д. Булева функция, описывающая MS с четырьмя информационными входами, имеет вид: $Y = D0 \wedge X1 \wedge X0 \vee D1 \wedge X1 \wedge X0 \vee D2 \wedge X1 \wedge X0 \vee D3 \wedge X1 \wedge X0, \dots$ (1), где Y – значение логического сигнала на выходе MS;

D0, D1, D2 и D3 – значения логических сигналов на информационных входах MS;

X1, X0 – управляющие переменные, задающие номер информационного входа MS.

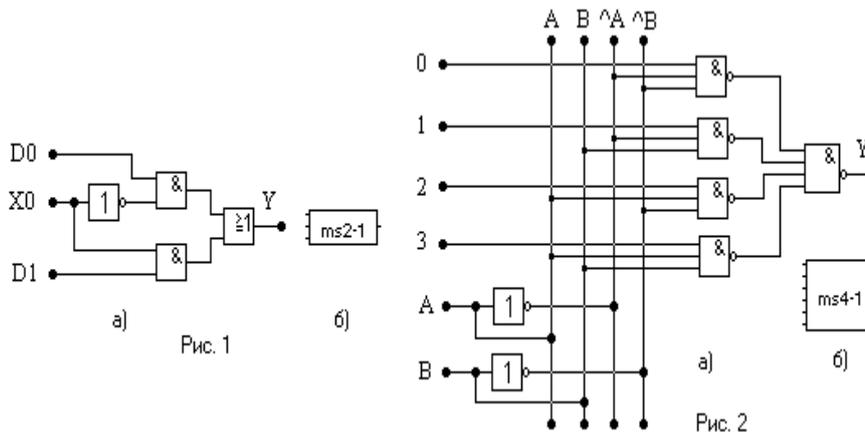
Аналогичный вид имеют выражения для MS с 2-мя, 8-ю и т.д. информационными входами. Обычно мультиплексоры обозначаются так: MS 2-1 – мультиплексор с двумя информационными входами на один выход; MS 8-1 – мультиплексор с восемью информационными входами на один выход и т.п. Мультиплексоры могут быть собраны из простейших логических элементов И, ИЛИ, НЕ или могут использоваться готовые MS в виде ИМС.

Схема MS 2-1, собранная из простейших логических элементов, приведена на рис. 1,а. Здесь же изображён вид субсхемы ms2-1 (рис. 1,б).

Схема MS4-1, собранная на базе простейших логических элементов, приведена на рис. 2,а. Здесь же показана субсхема ms4-1 (рис. 2,б). Информационные входы обозначены как 0, 1, 2 и 3, а управляющие – А, В.

Другая сфера применения мультиплексоров – построение логических функций нескольких переменных в виде дизъюнктивной нормальной формы. Это применение основано на разложении булевой функции по 1-ой, 2-ум и т.д. переменным по методу Шеннона. В качестве примера рассмотрим применение разложения Шеннона для реализации одной и той же булевой функции $Y = U(0, 1, 2, 3, 4, 8, 12, 13, 15)$ от четырёх переменных А, В, С, D с помощью мультиплексоров различного типа. Числа в скобках для функции U указывают те строки таблицы истинности, в которых Y принимает значения, равные 1. В СДНФ (совершенной дизъюнктивной нормальной форме) эта булева функция имеет вид:

$$Y = \wedge A \wedge \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge A \wedge \wedge B \wedge \wedge C \wedge D \vee \wedge A \wedge \wedge B \wedge C \wedge \wedge D \vee \wedge A \wedge \wedge B \wedge C \wedge D \vee \wedge A \wedge \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge A \wedge \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge A \wedge \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge A \wedge \wedge B \wedge \wedge C \wedge \wedge D, \dots (2)$$



где знак «*» означает операцию логического умножения.

Для построения комбинационной схемы, реализующей функцию (2) на базе мультиплексоров MS 2-1, разложим функцию по одной переменной, начиная, например, с переменной старшего разряда - А. Сгруппируем в выражении (2) члены, содержащие переменные А и $\wedge A$. Получим:

$$Y = \wedge A * (\wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge D \vee \wedge B \wedge C \wedge \wedge D \vee \wedge B \wedge C \wedge D \vee B \wedge \wedge C \wedge \wedge D) \vee A * (\wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge D \vee \wedge B \wedge C \wedge \wedge D) = \wedge A * f_1 \vee A * f_2, \dots (3)$$

$$\text{где функции } f_1 = \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge D \vee \wedge B \wedge C \wedge \wedge D \vee \wedge B \wedge C \wedge D; \\ f_2 = \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge \wedge D \vee \wedge B \wedge \wedge C \wedge D \vee \wedge B \wedge C \wedge \wedge D.$$

Аналогично разложим функции f_1 и f_2 по переменным С, $\wedge C$ и D, $\wedge D$ соответственно:

$$f_1 = \wedge C * (\wedge B \wedge \wedge D \vee \wedge B \wedge D \vee B \wedge \wedge D) \vee C * (\wedge B \wedge \wedge D \vee \wedge B \wedge D) = \wedge C * (\wedge B * (\wedge D \vee D) \vee B * \wedge D) \vee C * (\wedge B * (\wedge D \vee D) \vee B * \wedge D) \vee C * \wedge B; \dots (4)$$

$$f_2 = \wedge D * (\wedge B \wedge \wedge C \vee B \wedge \wedge C) \vee D * (B \wedge \wedge C \vee B \wedge C) = \wedge D * \wedge C \vee D * B. \dots (5)$$

Выражения (3), (4) и (5) позволяют построить принципиальную схему комбинационного устройства на MS2-1 для функции (2) - рис. 3. Схема содержит четыре мультиплексора MS2-1. Первый реализует зависимость (3), второй и четвёртый реализуют функцию f_1 , а третий мультиплексор – функцию f_2 .

При построении комбинационного устройства для реализации функции (2) на базе MS4-1 в исходной функции сгруппируем члены, вынося за знак скобок произведение двух переменных, (например А и В), следующим образом:

$$Y = \wedge A \wedge \wedge B * (\wedge C \wedge \wedge D \vee \wedge C \wedge D \vee C \wedge \wedge D \vee C \wedge D) \vee \wedge A \wedge B * (\wedge C \wedge \wedge D) \vee A \wedge \wedge B * (\wedge C \wedge \wedge D) \vee A \wedge B * (\wedge C \wedge \wedge D \vee \wedge C \wedge D \vee C \wedge \wedge D) = \wedge A \wedge \wedge B * I \vee \wedge A \wedge B * (\wedge C \wedge \wedge D) \vee A \wedge \wedge B * (\wedge C \wedge \wedge D) \vee A \wedge B * (\wedge C \wedge \wedge D \vee \wedge C \wedge D \vee C \wedge \wedge D). \dots (6)$$

Полученное выражение позволяет составить принципиальную схему комбинационного устройства для этого случая - рис. 4. На информационные входы MS1 (в соответствии с их адресом, определяемым переменными А и В)

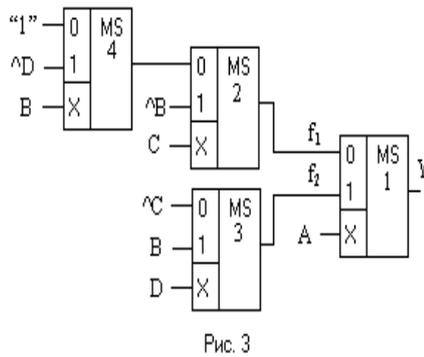


Рис. 3

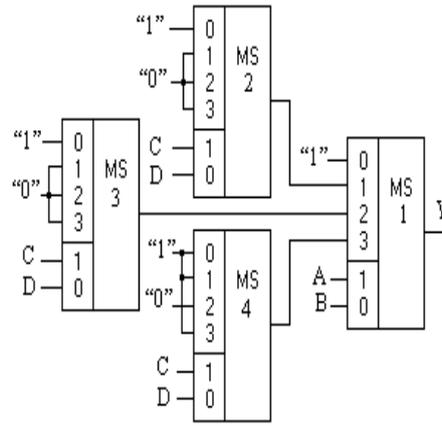


Рис. 4

подаются соответствующие функции (в скобках). Адрес $\wedge A \wedge B$ определяет нулевой информационный вход, поэтому сюда подводится логическая 1 (в соответствии с выражением для первого термина функции (6)). Адрес $\wedge A \wedge B$ определяет первый информационный вход, поэтому к нему подключён выход MS2, реализующего зависимость $(\wedge C \wedge \wedge D)$. Адрес $A \wedge \wedge B$ определяет второй информационный вход MS1, к которому подведён выход MS3, реализующего зависимость $(\wedge C \wedge D)$. Адрес $A \wedge B$ определяет третий информационный вход MS1, на который подаётся сигнал с выхода MS4, реализующего зависимость $(\wedge C \wedge \wedge D \vee \wedge C \wedge D \vee C \wedge D)$.

Рассмотрим построение комбинационной схемы на базе мультиплекса MS8-1. Для этого функцию (2) необходимо разложить по трём переменным, например, по А, В, С, сочетания прямых или инверсных значений которых будут определять в мультиплексе MS8-1 адреса определённых информационных входов. Сгруппируем в выражении (2) члены с одними и теми же адресами информационных входов, вынося их за скобки:

$$Y = \wedge A \wedge B \wedge C (\wedge D \vee D) \vee \wedge A \wedge B \wedge C (\wedge D \vee D) \vee \wedge A \wedge B \wedge C (\wedge D) \vee A \wedge B \wedge C (D) \vee A \wedge B \wedge C (\wedge D \vee D) \vee A \wedge B \wedge C (D) \quad (7)$$

Здесь адрес $\wedge A \wedge B \wedge C$ определяет информационный вход MS8-1 за номером 0; адрес $\wedge A \wedge B \wedge C$ определяет информационный вход MS8-1 за номером 1 и т.д.. Следует отметить, что в выражении (7) отсутствуют комбинации $\wedge A \wedge B \wedge C$ и $A \wedge B \wedge C$, определяющие третий и пятый информационные входы MS8-1. Выражения в круглых скобках (7) определяют подаваемый на данный вход сигнал. То есть сигнал на нулевой информационный вход $(\wedge D \vee D)=1$ обеспечивается подачей на него логической 1. Та же самая ситуация с первым информационным входом. На второй информационный вход с адресом $\wedge A \wedge B \wedge C$ в соответствии с (7) необходимо подать сигнал $(\wedge D)$. И так далее до входа с номером 7. На те адреса, которые отсутствуют в (7) – третий и пятый, необходимо подать логические 0. С учётом всех этих замечаний выражение (7) можно записать так:

$$Y = \wedge A \wedge B \wedge C (1) \vee \wedge A \wedge B \wedge C (1) \vee \wedge A \wedge B \wedge C (\wedge D) \vee \wedge A \wedge B \wedge C (0) \vee A \wedge B \wedge C (D) \vee A \wedge B \wedge C (0) \vee A \wedge B \wedge C (1) \vee A \wedge B \wedge C (D). \dots (8)$$

Таблица 1

Адрес инф. входа		Подаваемое значение
Код	Номер	
$\wedge A \wedge B \wedge C$	0	$(\wedge D \vee D)=1$
$\wedge A \wedge B \wedge C$	1	$(\wedge D \vee D)=1$
$\wedge A \wedge B \wedge C$	2	$(\wedge D)$
$\wedge A \wedge B \wedge C$	3	(0)
$A \wedge B \wedge C$	4	(D)
$A \wedge B \wedge C$	5	(0)
$A \wedge B \wedge C$	6	$(\wedge D \vee D)=1$
$A \wedge B \wedge C$	7	(D)

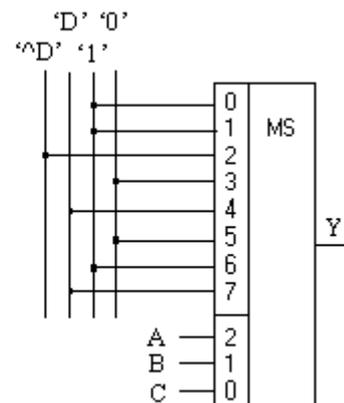


Рис. 5

Порядок программирования MS8-1 для реализации функции (2) приведён в табл.1. На рис. 5 показана принципиальная схема реализации функции (2) на мультиплексе MS8-1. Выбор адреса того или иного канала обеспечивается переменными А, В, С, подаваемыми на адресные входы 2, 1, 0. Соединение информационных входов с шинами “ $\wedge D$ ”, “D”, “1”, “0” выполнено по табл. 1.

В данной лабораторной работе для реализации функции на MS8-1 используются два MS4-1, которые выполнены в виде одной микросхемы **74153 (2 4 -to -1 Data Selectors/Mux)** из контейнера **Combinational (шаблон MUX)**. Обозначения на корпусе для первого MS4-1: 1C0,1C1,1C2,1C3 – информационные входы; 1Y – выход; A и B – входы сигналов управления. Для второго MS4-1: 2C0,2C1,2C2,2C3 – информационные входы; 2Y – выход; A и B – входы сигналов управления. Для объединения двух MS4-1 в один используются дополнительные входы 1G и 2G – разрешение работы 1-го и 2-го MS4-1. Для объединения 2-х MS4-1 в один MS8-1 используются дополнительные элементы: инвертор, через который старший разряд A подаётся на вход 2G – разрешения второго MS4-1, и элемент ИЛИ, объединяющий выходы мультиплексоров. Пример подсоединения микросхемы 74153 в качестве

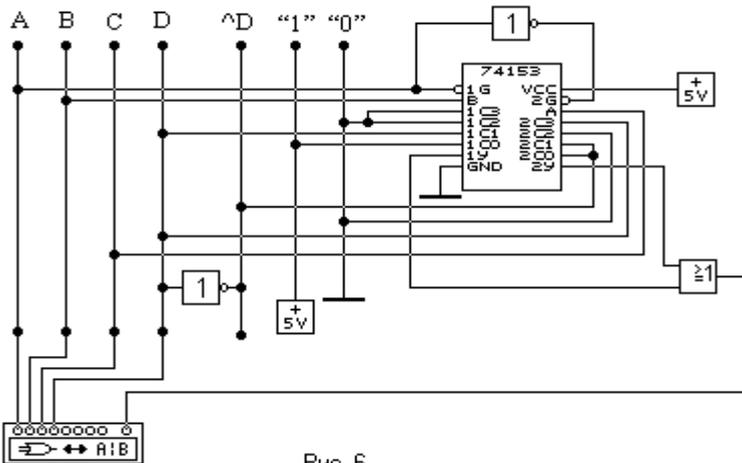


Рис. 6

мультиплексора MS8-1 для исследования с помощью логического преобразователя реализации некоторой булевой функции $F(A,B,C,D)$ приведён на рис. 6.

Для исследования одновыходных логических схем будем использовать логический преобразователь – устройство, обеспечивающее взаимные преобразования различных способов представления логических схем. С его помощью можно:

- по таблице истинности получить логическое выражение;
- по логическому выражению получить логическую схему или таблицу истинности;
- по логической схеме построить таблицу истинности.

Прибор предусматривает возможность использования до восьми входных логических переменных от **A** до **H**, подключаемых к входам исследуемой схемы, и вход **OUT**, к которому должен быть подключён выход исследуемой комбинационной схемы. Для активации переменной нужно щёлкнуть расположенный над ней кружок, после чего он станет тёмным. Если таблицу истинности задаёт пользователь, то после выбора входных переменных нужно активизировать переменную **OUT** и для каждого набора значений входных переменных задать значение выходной логической переменной. Подробная инструкция работы с этим прибором изложена в Программе Electronic Workbench для анализа электронных схем: Учебно-метод. пособие. В.М. Чухонцев. Самар. гос. тех. ун-т. Самара, 1999.- 92с.

5.2. МЕТОДИКА ВЫПОЛНЕНИЯ

1. Включите необходимое программное обеспечение.
2. Используя логические элементы НЕ, И и ИЛИ из контейнера **Gates**, соберите схемы мультиплексоров **MS2-1**, **MS4-1** согласно рис. 1,а и рис. 2,а. При этом собранные Вами схемы необходимо сохранить в виде субсхем. Для этого выделите собранную Вами схему с помощью мыши и нажмите клавиши **Ctrl** и **B**; введите имена Ваших субсхем **ms2-1** и **ms4-1** соответственно и нажмите кнопку **Copy from Circuit**. В результате схемы мультиплексоров в виде отдельных субсхем будут находиться в контейнере **Custom**.

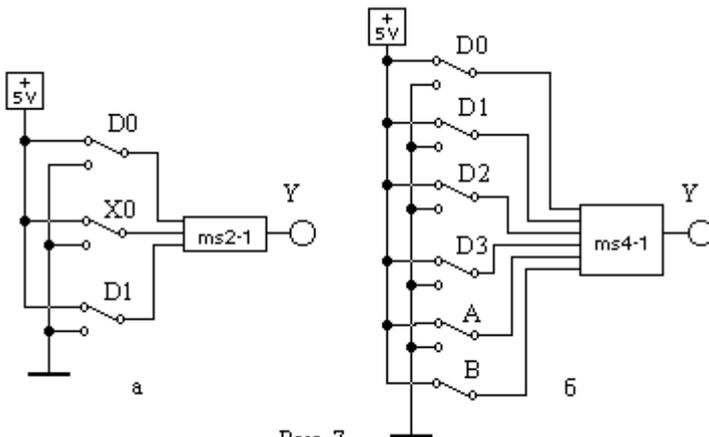


Рис. 7

3. Соберите схемы по рис. 7 для проверки работы субсхемы ms2-1 (рис. 7,а) и субсхемы ms4-1 (рис. 7,б). Задавая управляющими сигналами номер того или иного информационного входа убедитесь в появлении на выходе субсхемы сигнала именно нужного параметра D_i . Результаты эксперимента сведите в табл.

4. Используя разложение Шеннона по 1-й, 2-м и 3-м переменным, постройте схемы, реализующие заданную преподавателем функцию из приведённых ниже вариантов, на MS2-1, MS4-1 и MS8-1. Проверьте правильность выполнения задания с помощью логического преобразователя, таблицы истинности на экране которого для всех трёх вариантов схемной реализации должны совпадать с заданным вариантом.

- | | |
|-----------------------------------|------------------------------------|
| 1. $Y = U(0, 5, 15, 10, 8, 9)$ | 6. $Y = U(0, 15, 1, 10, 8, 11, 2)$ |
| 2. $Y = U(0, 1, 5, 4, 8, 14, 6)$ | 7. $Y = U(6, 5, 7, 14, 15, 10, 8)$ |
| 3. $Y = U(4, 5, 7, 15, 11, 9)$ | 8. $Y = U(7, 0, 5, 15, 14, 13)$ |
| 4. $Y = U(2, 3, 4, 5, 13, 9, 11)$ | 9. $Y = U(0, 2, 3, 5, 7, 12, 14)$ |
| 5. $Y = U(1, 2, 5, 7, 12, 13)$ | 10. $Y = U(10, 15, 3, 9, 6, 4, 0)$ |

Числа в скобках задаваемых функций определяют те строки таблицы истинности, в которых последняя принимает значение логической единицы.

Номер варианта соответствует номеру бригады на лабораторных занятиях.

5. Определите функцию $F(A,B,C,D)$, реализуемую схемой на рис. 6, с помощью разложения Шеннона.

5.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое мультиплексор и каковы области его применения?
2. Поясните принцип работы мультиплексора на примере MS4-1.
3. Объясните механизм разложения Шеннона булевой функции при разработке электрической схемы её реализации.
4. Какие возможности обеспечивает логический преобразователь при исследовании цифровых схем?

Контрольные вопросы ПЗ5 (ПК-4):

1. Охарактеризуйте основные функции устройства управления.
2. Какие этапы включаются в машинный цикл команды типа «Сложение»?
3. Какие этапы входят в машинный цикл команды типа «Запись»? Обоснуйте отсутствие одного из этапов.
4. Какие этапы входят в машинный цикл команды типа «Условный переход»? Обоснуйте отсутствие двух этапов.
5. Дайте характеристику входной и выходной информации модели УУ.
6. На какие две части делится структура УУ? Что входит в состав каждой части? Какое назначение имеют элементы частей УУ?
7. Перечислите достоинства и недостатки МПА с жесткой логикой.
8. Дайте характеристику элементов структуры МПА с программируемой логикой.
9. Объясните принцип управления на основе МПА с программируемой логикой.
10. Какие способы кодирования микрокоманд вы знаете? Перечислите их достоинства и недостатки.
11. Поясните подходы к адресации микрокоманд, охарактеризуйте их сильные и слабые стороны.
12. Какие способы организации памяти микрокоманд вы знаете, чем они обусловлены?
13. В каких случаях в МПА следует применять ПЛМ, а не ПЗУ? Ответ аргументируйте.
14. Сформулируйте числовые параметры МПА с программируемой логикой. От чего они зависят?
15. Поясните методику минимизации количества слов памяти микрокоманд.
16. Охарактеризуйте методику минимизации разрядности микрокоманд.
17. Какие подходы к повышению быстродействия МПА с программируемой логикой
18. Поясните достоинства и недостатки трех вариантов подключения системы ввода/вывода к процессору ВМ.
19. Сформулируйте достоинства, недостатки и область применения двух способов организации адресного пространства ввода/вывода
20. В чем состоит локализация данных, выполняемая модулем ввода/вывода?
21. Опишите содержание процедуры «рукопожатия» при выполнении операции ввода.
22. Конкретизируйте последовательность действий процессора при обмене информацией с жестким диском.
23. Сравните известные вам методы управления вводом/выводом по трем параметрам: достоинствам, недостаткам, области применения.
24. Поясните классификацию методов ввода/вывода по прерыванию.
25. Охарактеризуйте известные вам режимы прямого доступа к памяти, сформулируйте их достоинства и недостатки.
26. Сравните ввод/вывод по прерыванию с вводом/выводом при ПДП. Для какого режима ПДП эти методы наиболее близки и почему?
27. Проведите сравнительный анализ контроллера ПДП и канала ввода/вывода. В чем их сходство? Чем они отличаются друг от друга?
28. Опишите процесс взаимодействия ЦП и КВВ. Какая при этом используется управляющая информация?
29. Опишите задачи посредника между КВВ и ВУ.
30. Как формируется код символа в клавиатуре? Как он передается память компьютера?
31. Каково назначение мыши? На чем основана работа оптической мыши? Поясните принцип

ее работы.

32. В чем принцип работы дисплея на базе ЭЛТ? Что такое частота регенерации изображения и чему она равна в современных дисплеях?
33. Как работает ЖК-дисплей? Что такое разрешающая способность? Какой разрешающей способностью обладают современные дисплеи?
34. Какие принтеры в настоящее время наиболее распространены? Какая разрешающая способность для них характерна?
35. Как работает струйный принтер? Какими достоинствами и недостатками он обладает?
36. Как работает лазерный принтер? Перечислите все этапы формирования изображения.

ПРАКТИЧЕСКАЯ РАБОТА № 6.

Процессоры, контроллеры, внешние устройства

6.1. КРАТКАЯ ТЕОРИЯ

При передаче данных по линиям связи или между устройствами ЭВМ часто используют преобразование параллельного кода (байта) в последовательный (биты) и обратное преобразование в параллельный код. Преобразование параллельного кода в последовательный производится с помощью сдвигающего регистра: сначала в него записывают байт (параллельная запись), а затем переводят регистр в режим сдвига и подают 8 импульсов синхронизации. Таким образом, на выходе одного из разрядов регистра (старшего или младшего в зависимости от направления сдвига) последовательно появляются значения всех разрядов записанного байта. Эта последовательность по линии связи передается в приемник, главным элементом которого является такой же сдвигающий регистр. Через последовательный вход (по импульсам синхронизации) информация последовательно вводится в регистр приемника. Через 8 импульсов синхронизации передаваемый байт в регистре приемника готов для дальнейшей обработки (обычно - записи в буферный регистр). Подробнее о регистрах можно почитать в разделе «Краткая теория» лабораторной работы № 4.

6.2. МЕТОДИКА ВЫПОЛНЕНИЯ

На рис.1 приведена схема простейшего передатчика с ручным управлением. Она собрана на 8-ми битовом универсальном регистре типа 74198 из библиотеки **default** модели **ideal** (контейнер **Sequential**). Значение передаваемого байта ABCDEFGH задается с помощью переключателей. Режим работы регистра (запись, хранение, сдвиг влево, сдвиг вправо) задается с помощью переключателей (0, 1), соединенных с управляющими входами (S0,S1). Состояние разрядов регистра отображается индикаторными лампочками. С помощью переключателей (0, 1) можно задать режим параллельной записи в регистр передатчика и записать в него байт, затем задать режим сдвига и передать этот байт (последовательно по битам) по линии связи в приемник (рис. 2). Приемник соединен с передатчиком 2-мя

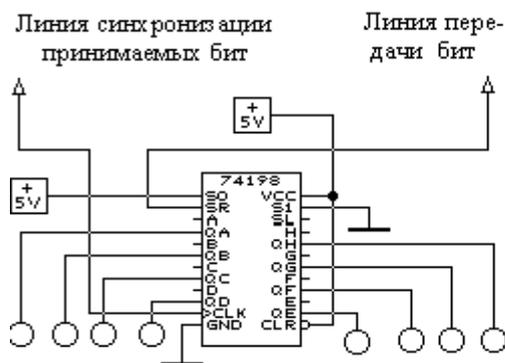
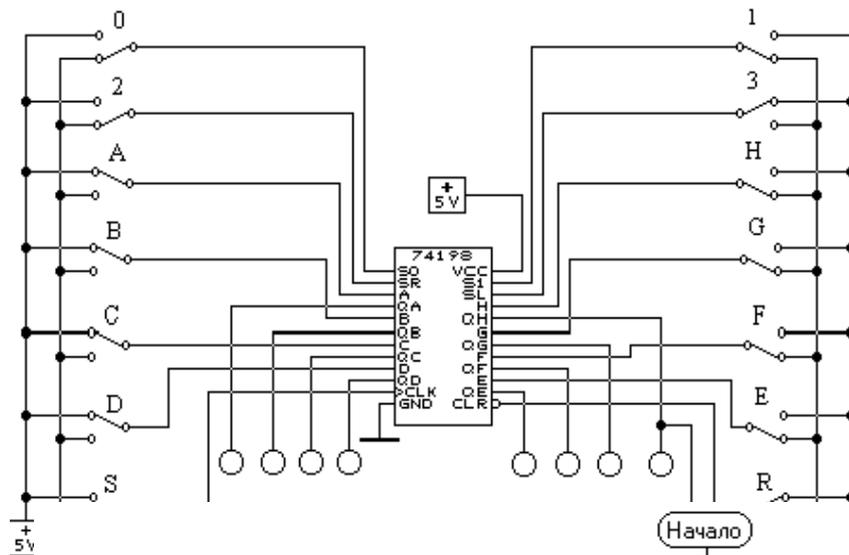


Рис. 2

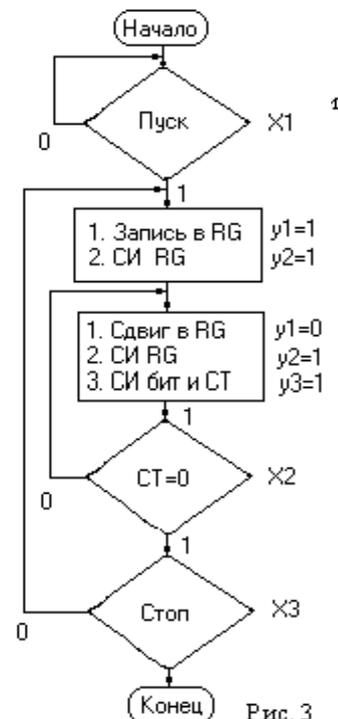


Рис. 3

линиями: по одной передаются биты, по другой - СИ (импульсы синхронизации бит).

Алгоритм работы передатчика приведен на рис. 3. В нем присутствуют условия “Пуск” и “Стоп”. По сигналу “Пуск” (кнопка) начинается работа передатчика. В начале регистр переводится в режим параллельной записи (Запись в RG), на входы параллельной загрузки подается байт, а затем - СИ на вход синхронизации регистра. Для передачи данных регистр переводится в режим сдвига, вырабатывается СИ бит (который передается в приемник и используется также для подсчета количества переданных бит) и СИ RG (син-хроимпульс сдвига). Следует обратить внимание на последовательность формирования фронтов СИ бит и СИ RG: сначала необходимо передать СИ бит в приемник, а затем сдвинуть содержимое регистра передатчика. Счетчик переданных бит изменяет свое состояние (+1) на каждом СИ бит. После передачи последнего 8-го бита содержимое счетчика (по MOD 8) становится равным нулю. Если нет команды “СТОП”, то начинается цикл передачи очередного байта.

На основании приведенного алгоритма необходимо спроектировать управляющий автомат, входами которого будут команды (кнопки) “ПУСК” и “СТОП”, а также логическое условие “СТ=0”. В целом устройство будет состоять из 2-х автоматов - операционного (ОА) и управляющего (УА). ОА будет содержать 2 основных операционных элемента: универсальный регистр и двоичный счетчик по модулю 8. Эти элементы необходимо выбрать самостоятельно из контейнера **Sequential**. Управляющий автомат следует спроектировать как автомат МУРА.

Спроектированный автомат УА следует собрать, проверить правильность его функционирования согласно исходной ГСА и оформить в виде подсхемы АWT. Для удобства отладки автомата максимально используйте

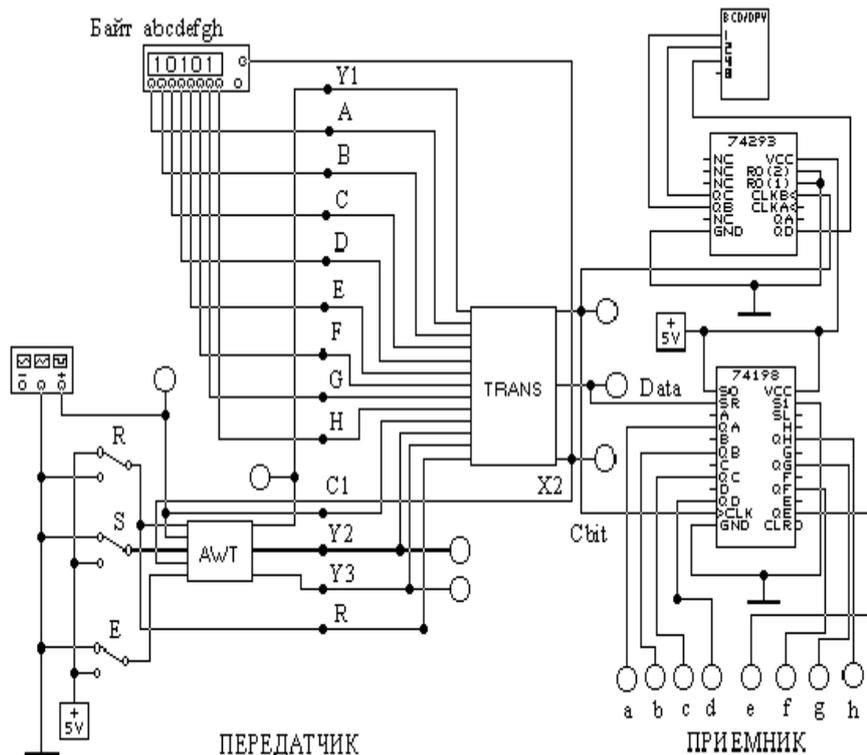


Рис. 4

элементы индикации (в том числе и цифровые индикаторы).

Перед подключением операционных элементов ОА к УА постройте временные диаграммы, из которых должна быть наглядно видна последовательность всех формируемых сигналов. В ОА для обеспечения этой последовательности можно использовать простейшие логические элементы И, ИЛИ, НЕ и т.п. При наладке ОА также используйте элементы индикации. Оформите отлаженный ОА в виде подсхемы TRANS. В качестве источника СИ для ОА и УА используйте FUNCTION GENERATOR, а источника передаваемых байт - WORD GENERATOR, который можно синхронизировать, например, логическим условием X2.

Схема приемника значительно проще схемы передатчика. На рис. 4 приведен один из вариантов общей схемы - передатчик + приемник.

В приемнике кроме сдвигающего регистра показан счетчик принятых бит и индикатор количества принятых бит. Регистр приемника всегда работает в одном режиме - сдвиг вправо. В передатчике используются кнопки:

- R - начальный сброс всех элементов передатчика;
- S - Старт; кнопка должна быть нажата в течении 2 СИ;
- E - Стоп.

Частоту генератора СИ следует выбрать 0.5 ... 1 Гц, чтобы можно было успевать наблюдать за работой устройства. При отладке генератор можно заменить на кнопку.

ЗАДАНИЕ

1. Соберите схемы (рис. 1 и рис. 2) и в ручном режиме проанализируйте работу каждой из них. Экспериментально определите сочетания управляющих команд для реализации различных режимов регистров.

2. Разработайте и соберите УА, затем ОА и устройство передачи-приёма в целом. Убедитесь в правильности работы УА и ОА.
3. Представьте принципиальные схемы Уа и ОА.
4. Нарисуйте временные диаграммы, иллюстрирующие работу устройства.

6.3. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Опишите работу схемы передатчика по рис. 1.
2. Опишите работу схемы приёмника по рис. 2.
3. Представьте и опишите работу подсхем управляющего и операционных автоматов передатчика.
4. Поясните работу передающего-приёмного устройства по рис. 3.
5. Каково назначение счётчика в цепи приёмника?

Контрольные вопросы ПЗ6 (ПК-4):

1. В чем суть идеи конвейеризации? В каких случаях в конвейер следует вводить буферные регистры? В каких случаях буферные регистры нужно заменять буферной памятью? Ответы обоснуйте.
3. В чем суть статического предсказания переходов? Сформулируйте достоинства и недостатки известных способов статического предсказания переходов.
4. В чем заключается смысл динамического предсказания переходов? Дайте развернутую характеристику достоинств и недостатков известных способов динамического предсказания переходов. Из каких соображений следует выбирать конкретную схему динамического предсказания? От чего зависит выбор?
5. Поясните идею суперконвейера. В чем заключаются достоинства и недостатки суперконвейеризации?
6. Поясните достоинства и недостатки ВМ с полным набором команд. Какие исторические причины привели к их возникновению?
7. Какие исторические причины способствовали появлению ВМ с сокращенным набором команд?
8. Перечислите основные характеристики ВМ с сокращенным набором команд.
9. Опишите возможности совместного использования в одной ВМ CISC-архитектуры и RISC-архитектуры.
10. Для чего вводится механизм регистровых окон? Поясните структуру окна. Ради какой цели окна организуются в виде циклического буфера?
11. Обоснуйте основные недостатки ВМ с сокращенным набором команд.
12. Дайте развернутую характеристику назначения и структурной организации суперскалярного процессора. Какие уровни параллелизма здесь используются?
13. На конкретных примерах поясните суть проблемы неупорядоченности команд в суперскалярных процессорах.
15. На примере конкретного программного фрагмента поясните суть метода переименования регистров. В чем состоят недостатки этого метода? Как их смягчить?
16. На примере конкретного программного фрагмента поясните смысл метода переупорядочивания команд. В чем заключается разница между централизованным и распределенным окном команд? Проведите сравнительный анализ организации этих окон.
17. Опишите назначение, организацию и порядок работы буфера восстановления последовательности. Для решения каких задач он применяется в современных процессорах?
18. Каким образом и при каких условиях гиперпоточковая обработка способствует повышению производительности процессора?
19. Сформулируйте правила объединения простых команд в командное слово сверхбольшой длины.

ЛИТЕРАТУРА

1. Агаханян Т.М., Плеханов С.П. Интегральные триггеры устройств автоматики. – М.: Машиностроение, 1978, - 368 с.
2. Шило В.Л. Популярныe цифровые микросхемы: Справочник. – Челябинск: Металлургия, Челябинское отделение, 1988, - 352 с.
3. Микроэлектронные устройства автоматики: Учеб. Пособие для вузов / А. А. Сазонов, А. Ю. Лукичѳв, В. Т. Николаев, и др./ Под ред. А. А. Сазонова. – М.: Энергоатомиздат, 1991. – 384 с.
4. Токхейм Р. Основы цифровой электроники: Пер. с англ. – М.: Мир, 1988. – 392 с.
5. Программа Electronic Workbench для анализа электронных схем: Учеб.- метод. пособие. В.М. Чухонцев. Самар. гос. тех. ун-т. Самара, 1999.- 92с.