## 5UA6J8em

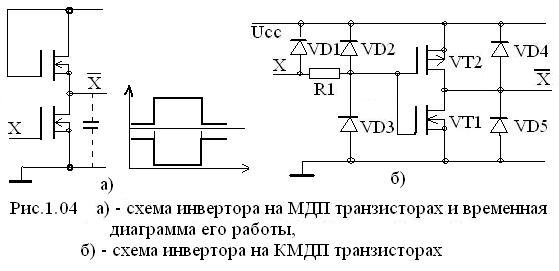
## I. Общие

1. Устройство, принцип функционирования и особенности элементов, построенных по технологии ТТЛ (рис.).
2. Устройство, принцип функционирования и особенности элементов, построенных по технологии КМОП (рис.).

МОП

КМОП

Схема базового элемента НЕ на МДП-транзисторах



VT2 используется как нагруз сопр Rн, что улучшает его динамические (временные) хар-ки элемента и уменьшает площадь на чипе.

Недостаток элемента – схема во включенном состоянии потребляет ток, ктр в статике не нужен.

Схема базового элемента НЕ на КМОП.

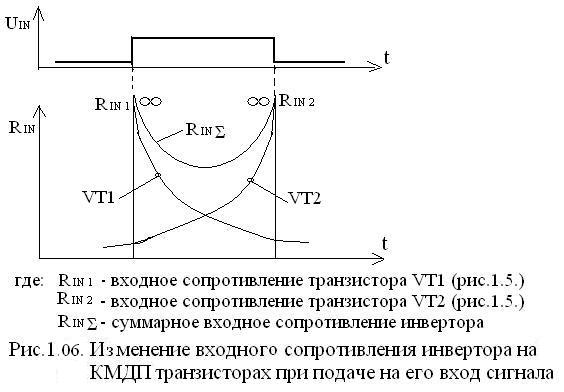
При подаче Х1 VT1 открывается, а VT2 – закрывается, и на выходе устанавливается низкий уровень . VT1 служит в качестве нагрузки.



Схема может быть разбита на 3 части:

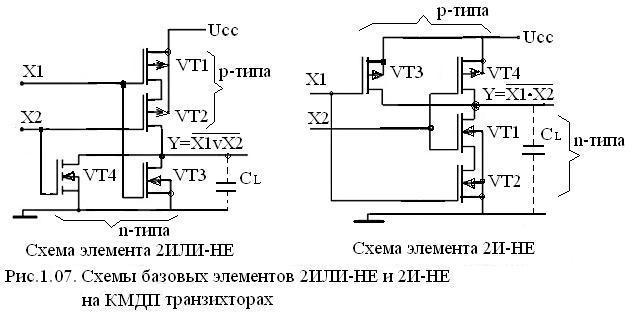
1. Диодно-резисторный(VD1, VD2 и R1) ограничитель напряжения.
2. Инвертор (VT1 и VT2).
3. Выходная диодная цепь (VD4 и VD5).

Потребление в статике практически равно нулю.



Статическое напряжение при хранении или монтаже, может пробить вход инвертора. Для предотвращения этого пробоя VD1, VD2 и VD3. Выходная цепь (VD4, VD5) накладывает ограничение на использование элемента, которое требует выполнения условия (иначе транзисторы могут сгореть) |Uвх-Uвых|<Uсс. Следовательно такие схемы требуют сначала подать на схему Ucc, а потом сигналы Uвх.

**Схемы базовых элементов 2ИЛИ-НЕ и 2И-НЕ.**



Работа элемента 2ИЛИ-НЕ. Штрих Шеффера

При отсутствии сигналов Х1 и Х2 VT3 и VT4 закрыты, а VT1 и VT2 открыты, на выходе У высокий уровень. При поступлении Х1(открывается VT3) или Х2(открывается VT4) , или одновременно Х1 и Х2 (VT3 и VT4 открываются) на выходе У устанавливается низкий уровень.

Работа элемента 2И-НЕ. Стрелка Пирса

При Х1 или Х2 на выходе У высокий уровень, так как VT1 или VT2 закрыты, а VT3 или VT4 открыты. При подаче Х1 и Х2 одновременно VT1 и VT2 одновременно открываются, и VT3 и VT4

**Выводы:**

1). Как и у ТТЛ элементы на КМДП не допускают парал-ого включения во избежание КЗ при выполнении разными элементами различных функций.

2). Если для реализации заданной функции транзисторы с каналом n-типа включаются последовательно, то парные им транзисторы p-типа включаются параллельно и наоборот.

3). Схемы на КМДП транзисторах в статике ток практически не потребляют.

4). Высокую помехоустойчивость КМДП (МДП) элементов.

5). КМДП (МДП) транзисторы более экономичны по сравнению с ТТЛ, но уступают им по быстродействию.

6). Наблюдается закономерность. Чем больше потребление элемента, тем более он быстродействующий.

Добавить уровни в вольтах и в виде временной диаграммы

1. Способы адресации, реализованные в системах команд современных микропроцессоров и микроконтроллеров.

Большая часть всех команд из системы команд предназначена для обработки данных. К таким командам относятся команды сложения, умножения, пересылки данных и т.д. Данные, обрабатываемые командами, называются **операндами**. Они могут храниться в различных сегментах ОЗУ, стеке, регистрах процессора и т.д.

Различают *исполнительный адрес операнда и адресный код команды.*

*A*исп - двоичный код номера ячейки памяти, служащей источником или приемником операнда. Этот код подается на адресные входы ЗУ и по нему происходит фактическое обращение к указанной ячейке. Если операнд хранится не в основной памяти, а в регистре процессора, его Аисп будет номер регистра.

*A*к — двоичный код в адресном поле команды, из которого необходимо сформировать Аисп.

*Способ адресации* — это способ формирования Аисп операнда по Ак.

Способы адресации классифицируют:

• по наличию адресной информации в команде (явная и неявная адресация).

При **явной адресации** операнда в команде есть поле адреса этого операнда.

При **неявной адресации** адресное поле в команде отсутствует, а адрес операнда подразумевается кодом операции.

• по кратности обращения в оперативную память. Различают:

• непосредственную адресацию (НА)

• прямую адресацию (ПА)

• косвенную адресацию (КА)

1) При НА в адресном поле команды вместо адреса содержится непосредственно сам операнд. . Для примера, в МК семейства mcs51 при таком СА не требуется обращения к регистрам или ПД, операнд поступает из ПП: add A,#77; добавить к аккумулятору число (константу) 77. # - обозначает константу.



Такой СА используется при выполнении арифметических операций, операций сравнения, а также для загрузки констант в регистры. При записи в регистр, имеющий разрядность, превышающую длину непосредственного операнда, операнд размещается в младшей части регистра, а оставшиеся свободными позиции заполняются значением знакового бита операнда.

Недостаток этого СА - размер непосредственного операнда ограничен длиной адресного поля команды.

2) При ПА адресный код прямо указывает номер ячейки памяти, к которой производится обращение, то есть Ак и Аисп совпадают.



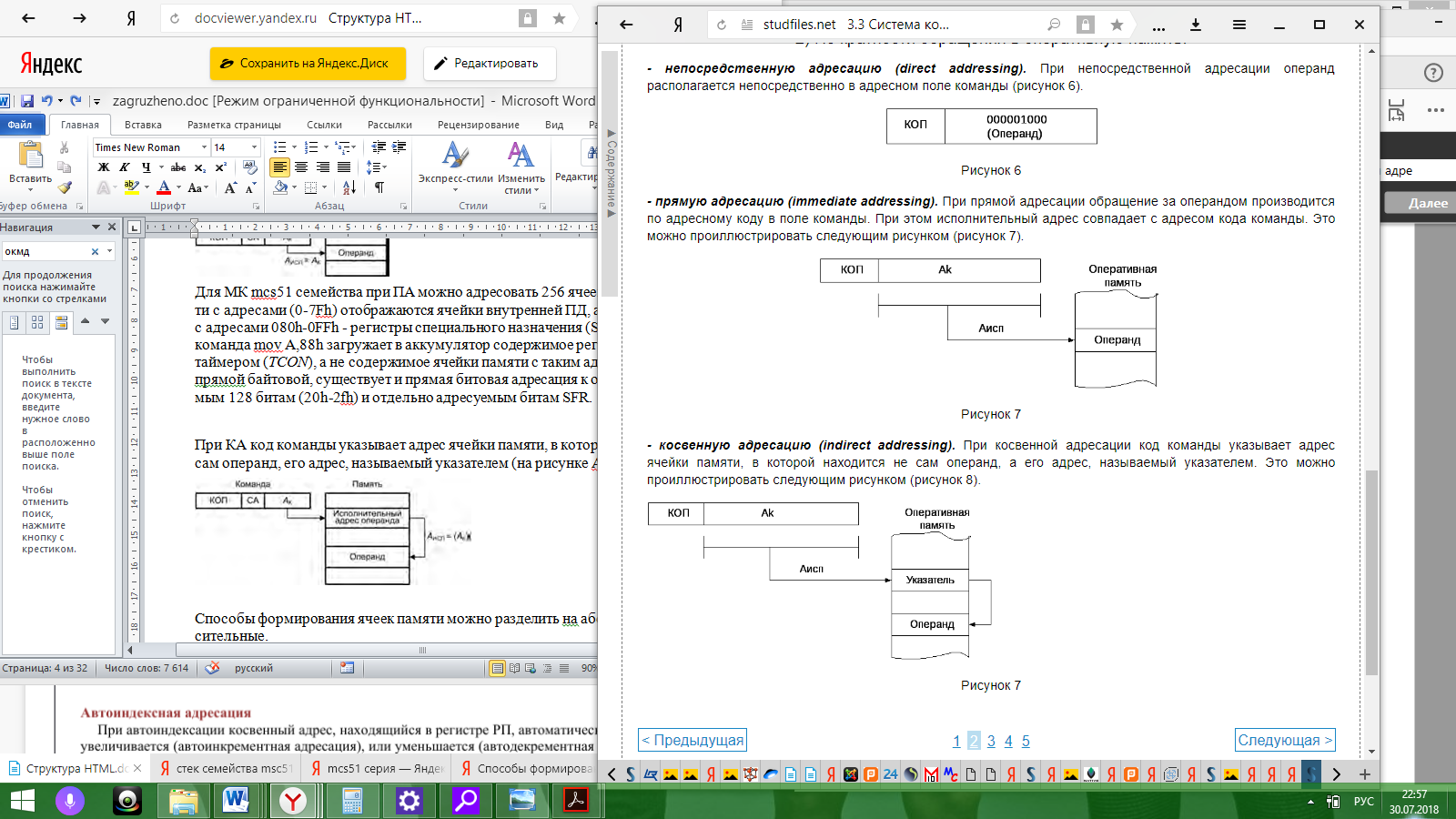
Для МК mcs51 семейства при ПА можно адресовать 256 ячеек. На ячейки памяти с адресами (0-7Fh) отображаются ячейки внутренней ПД, а на ячейки памяти с адресами 080h-0FFh - регистры специального назначения (SFR). Например: команда mov A,88h загружает в аккумулятор содержимое регистра управления таймером, а не содержимое ячейки памяти с таким адресом. Кроме прямой байтовой адресации, существует и прямая битовая адресация к отдельно адресуемым 128 битам (20h-2fh) и отдельно адресуемым битам SFR.

*Регистровая адресация* (РА). Схожа с ПА. Разница в том, что адресное поле команды указывает не на ячейку памяти, а на регистр процессора (рис). .



Двумя основными преимуществами регистровой адресации являются: короткое адресное поле в команде и исключение обращений к памяти.

3) При КА код команды указывает адрес ячейки памяти, в которой находится не сам операнд, его адрес, называемый указателем.



КА позволяет упростить обработку массивов, а также передачу параметров подпрограммам.

**В МК семейства mcs51.**

**Косвенно-регистровая адресация** осуществляется с помощью регистров *R*0 или *R*1текущего банка, т.е. значение одного из этих регистров интерпретируется как 8-разрядный адрес для обращения к первым 256 ячейкам памяти. При этом такая адресация не позволяет обращаться к регистрам по адресам 080h-0FFh (SFR). Пример: orl A,@R0 – логическое сложение аккумулятора с байтом по адресу, расположенному в регистре *R*0 (т.е. если *R*0=88h, то обращение будет к ячейке ОЗУ, а не к регистру *TCON* как при прямой адресации).

**Косвенно-регистровая адресация со смещением** образуется при помощи 16- разрядного индексного регистра *DPTR*. Этот вид адресации удобен для доступа к структурированным переменным образующих массив: mov A,@(DPTR+const).

Для доступа к таблицам (константам) хранящимся в памяти программ можно использовать команды: movс A,@A+DPTR или movc A,@A+PC, которые загружают в аккумулятор значение из ячейки с адресом образуемым содержимым аккумулятора и регистра *DPTR* или *PC* соответственно. Последняя адресует в памяти программ байт, расположенный со смещением относительно адреса текущей команды. Значение смещения содержится в аккумуляторе и вычисляется заранее.

• по способу формирования адресов ячеек памяти. Можно разделить на абсолютные и относительные.

1 Абсолютные способы формирования предполагают, что двоичный код адреса ячейки памяти м/б целиком извлечен либо из адресного поля команды, либо из какой-нибудь другой ячейки в случае косвенной адресации

2 Относительные способы формирования предполагают, что двоичный код адресной ячейки памяти образуется из нескольких составляющих:

• Б - код базы, • И - код индекса, • С - код смещения. Эти составляющие используются в различных сочетаниях. Т.е при относительной адресации применяется способ вычисления адреса путем суммирования кодов, составляющих адрес. Аи = Б + И + С или Аи = Б + С или Аи = И + С

**Индексная адресация.** (ИА)



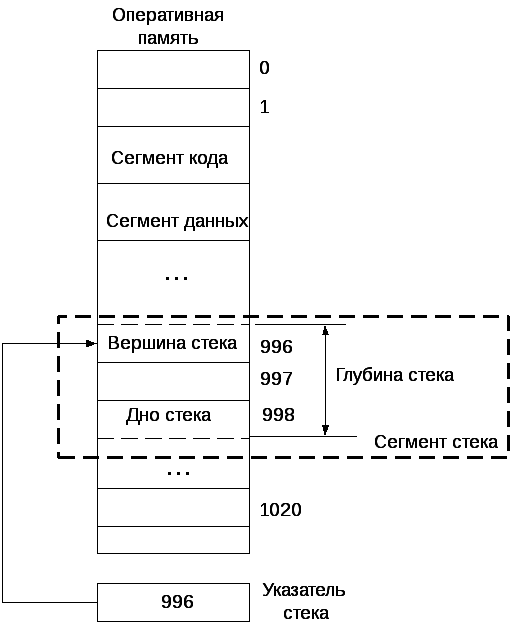
При ИА поле *А*К содержит полноразрядный адрес ячейки памяти, а регистр (указанный явно или неявно) — смещение относительно этого адреса. Используется для организации итеративных вычислений.

**Автоиндексная адресация**

При автоиндексации адрес, находящийся в регистре РП, автоматически увеличивается (автоинкрементная адресация), или уменьшается (автодекрементная адресация) на постоянную величину до или после выполнения операции.

**Стековая адресация.**

Стековая память реализуется на основе обычной памяти с использованием указателя стека и автоиндексной адресации. Логически ячейки памяти, отводимые под стек, организованы так, чтобы считывание последнего производилась первым, а первого записанного производилась последним. Такая логическая организация формируется специальным счётчиком. Этот счётчик называется указателем стека SP – это регистр микропроцессора. Ячейка памяти, в которую в данный момент может быть записан, например адрес возврата из подпрограммы, называется ***вершиной стека***. Количество ячеек памяти, предназначенных для организации стека, называется ***глубиной стека***. Последняя ячейка памяти, в которую можно производить запись называется ***дном стека***. Запись в стек производится с использованием автодекрементной адресации, а чтение - с использованием автоинкрементной адресации.



1. Вытесняющие и невытесняющие алгоритмы планирования в многозадачных операционных системах.
2. Триггеры, классификация, область применения.

**Триггер** – логическая схема с положительной обратной связью, способной формировать 2 устойчивых состояния на выходах Q и 𝑄̅ . Изменение выходов триггера происходит **под действием входного сигнала**. Предназначен для хранения 1 бита информации ( либо лог. 1, либо лог. 0).

Схема триггера обеспечивает запись, считывание, хранение, стирание и индикацию двоичной информации.

На основе триггеров строят типовые функциональные узлы ЭВМ – регистры, счётчики, накапливающие сумматоры и микропрограммные автоматы.

Состояние триггеров определяется по выходам:Q- прямой, 𝑄̅ – инверсный выход

Классификация триггеров:

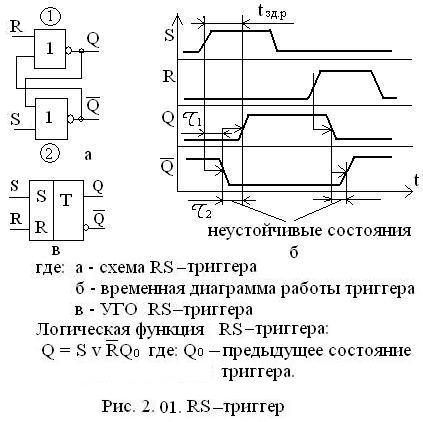
* по типу используемых входов (RS, JK, D, DV, T, TV)
* по способу записи информации (синхронные и асинхронные)
* по моменту реакции на входной сигнал (статические и динамические)
* по числу тактов синхронизации (одно-, двух-, трех- тактовые)
* по числу ступеней (одно и двух ступенчатые)

Реакция на входной сигнал может быть: на фронт или спад (прямому или инверсному динамическому сигналу) (то есть на перепад из 0 в 1 или из 1 в 0), а так же по прямому или инверсному статическому сигналу)

Основные инф-онные входы:

S – раздельный вход установки; R - ----//---- сброса; J – вход установки универсального триггера; K - -----//------ сброса; T – счетный вход; D - информационный; C – синхронизирующий вход; V – вход разрешения/блокировки передачи сигнала со входа на выход с целью длительного хранения.

**RS – триггеры**



|  |  |  |
| --- | --- | --- |
| S | R | Q(n+1) |
| 0 | 0 | Q(n) |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | Запрещено! |

**Принцип действия:**

Q=1 поступает на вход элемента 2 и вызывает на его выходе 0, который в свою очередь поступая, на вход элемента1 поддерживает на его выходе Q=1 в таком состоянии схема может находится сколь угодно долго. триггер установлен. Триггер также будет устойчив при Q=0 и Q=1.

**Основные свойства:**

1) Исходное состояние нулевое (Q=0,¬Q=1). **После поступления S=1 сигнал хранится до появления R=1 сколь угодно долго!**

2) **Оба элемента ИЛИ-НЕ переключаются поочередно последовательно!**

Во время переключения снятие входного сигнала запрещено! Триггер может вернуться в первоначальное состояние, что опасно для последующих схем.

3) **Время задержки переключения триггера –**



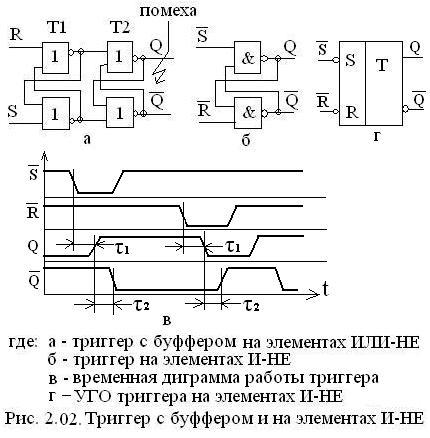
– время переключения 1 элемента; – время переключения 2 элемента



4) **При подаче на вход одновременно двух сигналов S=1 и R=1 на выходах Q и появятся 0.** Если эти сигналы убрать одновременно, оба элемента начнут переключаться в единичное состояние, стремясь оставить партнера в 0.



5) У триггера есть ОС, и **любая короткая помеха может** подействовать на другие входы и **привести к переключению триггера**. Двухступенчатый триггер исправляет это, т.к. на входах Т2 постоянно действуют сигналы с выходов Т1. **>>**

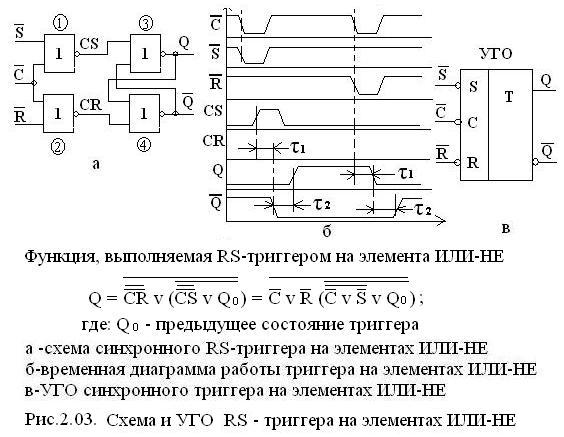


Можно получить **RS – триггерах на элементах И-НЕ**.

|  |  |  |
| --- | --- | --- |
| notS | notR | Q(n+1) |
| 0 | 0 | Запрещено! |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | Q(n) |

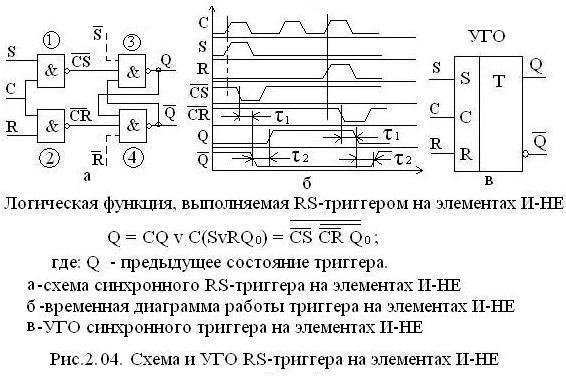
**Синхронные RS – триггеры**

**Первые два эл-та ИЛИ-НЕ** (асинхронный триггер-фиксатор) **составляют схему управления RS – триггера на эл-тах 3,4**.



|  |  |  |  |
| --- | --- | --- | --- |
| nС | nS | nR | Q(n+1) |
| 0 | 0 | 0 | Запрещено! |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | Q(n) |
| 1 | \* | \* | Хранение |

**Cинхронный триггер на И-НЕ аналогичен триггеру на элементах ИЛИ-НЕ** (схема управлен. на элементах 1 и 2 с прямыми входами RS C).



|  |  |  |  |
| --- | --- | --- | --- |
| С | S | R | Q(n+1) |
| 0 | \* | \* | Хранение |
| 1 | 0 | 0 | Q(n) |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | Запрещено! |

Задержка распространения сигнала в синхронном RS – триггерах составляет 3



**D – триггеры. DV – триггеры.**

**D – триггер** – синхронный триггер **с 1 входом D**, информация с которого **записывается по сигналу С**.



Latch – **при С=1 триггер прозрачен**, т.е. любое изменение D- сигнала, меняет состояние выхода, в соответствие с D – выходом.

|  |  |  |
| --- | --- | --- |
| C | D | Q(n+1) |
| 0 | \* | Q(n) |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Иногда на D- триггере вводятся дополнительные входы:

1. **- приоритетный сброс** (только при С=0). При **С=1 & D=1** образуется **RS – триггер на элементах 3,4**. Поэтому подача **=0 запрещена**, т.к. приводит к неопределённости выходов.



1. => **DV – триггер** – работает как D – триггер только при **D=1 & V=1**.



**Динамическая характеристика D – триггера (рис 2.05)**

Временная диаграмма построена с учетом состояние неопределённости. Переходной процесс в триггере характеризуется наличием задержки распространения сигнала по двум трактам:

tзад, СQ от входа сигнала С до выхода Q при входном сигнале D=const

tзад, DQ от входа D до выхода Q при входном сигнале С=1=const

**Выводы:**

1) Для фиксации состояния D – входа на выход Q без сбоев переходных процессов в схеме вызванный срезом С сигнала не должен накладываться на переходный процесс, вызванный переключением D – входа, т.е. tпд и tвд (подготовки, выдержки).

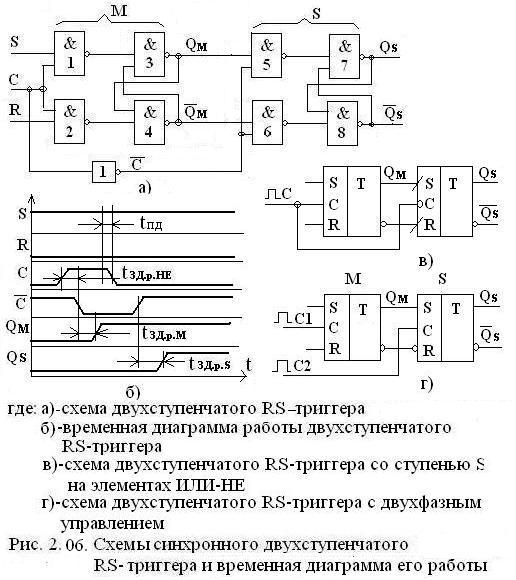
Обычно tпд=(1), а tвд=(0 , где – длительность фронтов при переключении сигнала, определяет максимально допустимую частоту переключения триггера.



2) Существует минимальная длительность С сигнала, которая обеспечивает отсутствие сбоев из-за наложения переходных процессов, вызванных передним и задним фронтами сигнала.

**Двухступенчатые синхронные RS – триггеры**

**Строятся последовательным соединением двух синхронных RS-триггеров**, называемых   
**M-ступень** (Master – хозяин)   
и **S-ступень** (Slave – раб) , **управляемых** **общим С**, обеспечивают ***совмещение записи новой информации и считывание ранее записанной***.



Во время действия С сигнала 1-ая ступень m-может принимать информацию по входу С, а со 2-ой ступени в это время можно считывать информацию ранее записанную в нее, с помощью этого же С – сигнала.

**Запись в первый триггер** происходит при **С=1** (второй триггер в это время в режиме хранения). При смене значения С на **С=0** происходит **запись значений из первого триггера во второй**. Т.о., запись происходит по спаду сигнала синхронизации С.

В приведенной схеме RS–триггера есть 2 пути распределения С–сигнала. Один через элементы 1-2-3-4, а второй через инвертор и далее в элементах 5 и 6 эти сигналы встречаются возможны гонки.



1. **Построение противогоночной схемы** Ступень m выполнена на элементах И-НЕ, а s ступень на ИЛИ-НЕ . Следующие ступени будут срабатывать на противоположных уровнях сигнала С, тогда влияние задержки С-сигнала 0.

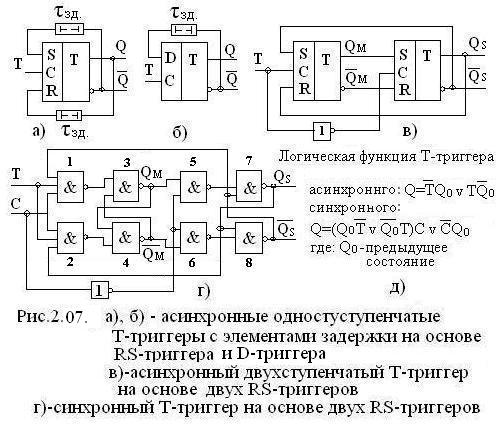


1. **Двухтактная синхронизация.** Управление разнесенными во времени С1 и С2.

**Двухступенчатый Т – триггер (**счётный**)**

**Т – триггер** – это триггер с одним входом Т состояние которого меняется после прихода, каждого счётного импульса на Т вход.

|  |  |
| --- | --- |
| T | Q(n+1) |
| 0 | Q(n) |
| 1 | notQ(n) |



|  |  |  |
| --- | --- | --- |
| T | C | Q(n+1) |
| 0 | \* | Q(n) |
| 1 | 0 | Q(n) |
| 1 | 1 | notQ(n) |

Проблему надёжного функционирования счетного одноступенчатого триггера решает схема двуступенчатого RS – триггера. Рис 2. В этом синхронном Т – триггере на входе элемента 1 окажутся три 1–цы , информация запоминается в первую ступень, а после их окончания переписывается во вторую ступень s.

Например: допустим, триггер сброшен (Qs=0, =1). При поступление сигналов С и Т на входе элемента 1 окажется три единицы и информация Т запишется в первую ступень, а при их окончании запишется во вторую.



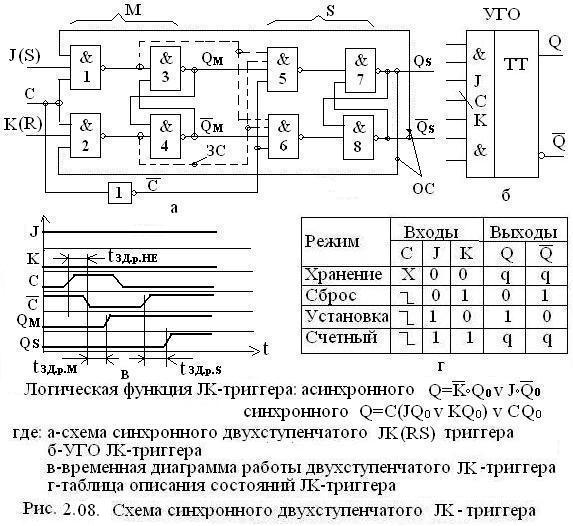
Совершенно аналогично при обратном состоянии триггера.

Триггер осуществляет сложение сигналов по mod2 . У такого Т – триггера могут быть дополнительные входы, сигналы С, Т, V, где сигнал V – сигнал блокировки ( подали С и Т, а V – нет триггер не работает.



**JK – триггер**

**JK – триггер** строится на основе двухступенчатого RS – триггера



**Является универсальным триггером**, поскольку на его основе могут быть построены другие ранее рассмотренные (Т, DV, D – триггеры).

В таблице: «Z» – означает, что срабатывание происходит по срезу С сигнала ( по заднему фронту), «q» – состояние выхода триггера перед поступление С сигнала, «х» - сигнал не действует на триггере, состояние « безразличия»

Работа JK – триггера( без учета пунктирной линии) (ЗС-запрещающие связи)

Очевидно, что на входе одного из элементов 3И-НЕ, поступает с выхода Qs или Q либо 0, либо 1

Допустим исходное состояние JK – триггера Qs=0, s=1 и Qm=0, m=1 на вход элемента 1по обратной связи поступает 1, а на вход элемента 2 поступает 0. При подачи С=1 и сигнала J=1 первая ступень по фронту С перебросится, т.е на выходах окажутся Qm=1, m=0, а по срезу С сигнала (через инвертор) перебросится и ступень S, т.е Qs=1, s=0.



Все аналогично происходит при противоположном состояние триггера, т.е при С и К=1 на выходе Qs=0, s=1, т.е триггер сбросится.



**Основные свойства JK – триггера.**

**1 свойство:** не при С=0, не при С=1 триггер для входных сигналов не прозрачен, хотя каждая ступень в отдельности при С=1 прозрачная.

**2 свойство:** при J=K=0 режим хранения, С-сигнал в это время действовать не может.

**3 свойство:** при J=1, а K=0, C – сигнал открывает только элемент 1, при условии, что на выходах триггера Qs=0, a s=1.Если до этого было Qs=1, а s=1 , то триггер остается без изменений, т.е в этом состояние J – вход выполняет функцию S – входа, RS – триггера. В силу симметричности схемы аналогичный выход будет и при исходном состояние J=0, K=1. В этом случае K – вход выполняет функцию R – входа RS – триггера.



**4 свойство:** при J=K=1, JK – триггер существенно отличается от RS – триггера для которого такая комбинация была запрещена, в этом случае JK – триггер при любом предыдущем состояние, по срезу С – сигнала меняет состояние выходов на противоположные, такой режим работы JK – триггера называется счётным или режим кувыркания, как в Т – триггере.

Некоторые JK – триггеры могут иметь двойные или даже тройные конъюнктивные входы, что на его свойства не влияет. В этом случае на J и K должны подаваться конъюнкции сигналов на входе с другими сигналами.

На УГО JK – триггера (рис.2.08б) две буквы ТТ обозначают двухступенчатый триггер.

**Некоторые особенности двухступенчатых триггеров:**

1. **У JK – триггера есть параллельные пути распределения сигнала**, а именно через m – ступень и через инвертор, после чего они встречаются в элементе 5 или 6, есть **возможность возникновения гонок.**
2. В JK – триггере, кроме построения S – ступени в дуальном варианте или разнесенных во времени сигналов С1 и С2, иногда применяют **способ построения триггера с обратными запрещенными связями**.
3. В схемах с запрещающими связями возможно **явление проскока фронта**.

**Суть его в следующем:**

Пусть JK – триггер в режиме хранения (т.е J=K=0). Тогда при С=1 на одном из входов J или K, конъюнктур которого 1 или 2 открыт по цепи обратной связи S – ступени поступают короткая одиночная помеха, в том числе и по цепи запрещающей связи на S – ступени, на его выходе вызовет переключение m – ступени, которая сразу же пройдет в S – ступень, переключает её по своему фронту не дожидаясь С – сигнала, если при С=1, на J или K поступает 0 помеха, произойдет аналогичное переключение S – ступени, но уже по отрицательному фронту не дожидаясь переключения С – сигнала. В обоих случаях произойдет сбой в работе и триггер зафиксирует нужный сигнал.

1. В синхронном триггере без запрещающих связей возможно **явление захвата 1**.

**Суть:** Если задержка в инверторе превышает задержку в m – ступени, то при поступление С – сигнала новое состояние m – ступени успеет проскочить в S – ступень, прежде чем инвертор закрывает элементы 5 и 6 и след-но выход триггер Qs – изменится не по фронту С – сигнала, а по его срезу, что может вызвать сбой в узле, который принимает сигнал из этого триггера.

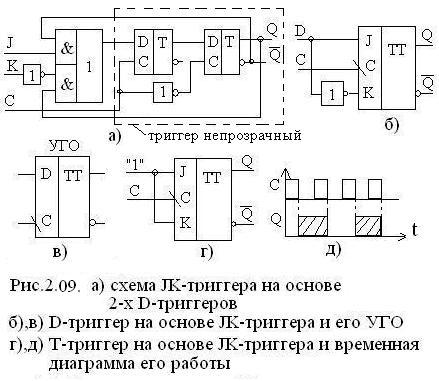
Метод борьбы достигается за счёт регулирования порогового срабатывания. В рассмотренных триггерах так же возможны гонки по выходу.

**Из рассмотренных случаев сбоя в работе двухступенчатых триггеров можно сделать вывод:**

1. Надо обеспечить окончание всех переходных процессов в логических схемах, формирующих сигналы для входов J и K до начала действия С – сигнала, т.е. обеспечить промежуток времени tвд( время выдержки) и tпд (время подготовки)
2. В течение действия С – сигнала уровни на входе J или K не должны изменится.

**Применение JK – триггера для построения Т и D триггеров.**

На рис. 2.09(а) приведена схема **JK – триггера на основе D – триггера**.



Важным свойство –**отсутствие явления захвата единицы**. Этот триггер не проницаем для помех, поэтому составляющие входов J и K можно менять, как при С=0, как и при С=1 на сигнала и tвыд .после среза.

Часто такие триггеры называют триггерами переключаемые перепадом. Их применение выгодно в цепях с помехами. В остальное время триггер помех не боится.

В отличие от ранее рассмотренного D-триггера этот триггер будет не прозрачен по D – входу.

Явление проскока фронта и захвата 1 отсутствует, поскольку триггер переключается перепадом сигнала, однако если такой триггер построить на основе JK – триггера с запрещ. связям, то у него при С=1 будет проскок фронта по D – входу

В этом случае на входы J и К подача логич 1. Такой триггер используется только с счётчиком и будет по сути Т-триггером.

1. Язык HTML.

**HTML** – язык разметки гипертекста. Данный язык является основным для создания сайтов. Предназначен для написания гипертекстовых документов, публикуемых в WWW.

Важным в HTML является то, что за его основу принят обычный текстовый файл, что решило проблему переносимости между различными операционными системами и аппаратными платформами.

HTML-страница и HTML-документ это разные понятия.

**HTML-документ** – это текстовый файл содержащий данные в формате HTML.

**HTML-страница** – это некая единица текстовой и графической информации отображаемая в браузере.

Часто страница полученные простым считыванием HTML-документа назыв. статическими, а страницы полученные в результате работы скрипта (программы) – динамическими.

HTML является описательным языком разметки. В нем используются указатели разметки (**теги или дескрипторы**).

Теговая модель описывает документ как совокупность контейнеров, каждый из которых начинается и заканчивается тегом. Тег состоит из имени, после которого может идти не обязательный список атрибутов. Тест тега заключается в такие скобки <>. В конечном теге после имени атрибута ставится /. Конечные теги никогда не содержат атрибутов.

**Атрибуты тега** следуют сразу после имени, отделяясь от него пробелом, знакам табуляции или переходом на новую строку. Порядок следования атрибутов не важен. Регистр символов в именах тегов и атрибутов не учитывается, чего нельзя сказать о значении атрибутов.

**Вложенные теги** необходимо закрывать начиная с последнего. Некоторые элементы разметки не имеют конечного компонента, т.к. являются автономными объектами(BR, HR и т.д.). Некоторые браузеры воспринимают конец тега, началом предыдущего (если не стоит конечный тег).

HTML-документ: 1) ограничивается тегами <HTML></HTML> - браузер должен знать, где страница начинается и заканчивается. В заголовке находится название страницы (теги <TITLE> </TITLE>). 2) состоит из заголовка <HEAD></HEAD> содержащий различную информацию о документе. 3) тело документа ( то что отображается в окне браузера) <BODY></BODY>

Последовательность символов, составляющая текст, может состоять из пробелов, табуляций, символов перехода на новую строку, символов возврата каретки, букв, знаков препинания, цифр, и специальных символов (например #, +, $, @), **за исключением следующих четырех символов, имеющих в HTML специальный смысл: < (меньше), > (больше), & (амперсанд) и " (двойная кавычка)**. Если необходимо включить его в текст, то следует закодировать его особой последовательностью символов.

Простейший документ HTML

<HTML>

<HEAD>

<TITLE>Заголовок</TITLE>

</HEAD>

<BODY><!--Здесь находится текст и различные картинки страницы-->

</BODY>

</HTML>

**Комментарий:** <!-- Этот текст не будет выведен в окно браузера -->

**Форматирование текста:**

Элемент **абзаца** **<P align="left/center/right">Текст</P>**

Атрибут позиционирования **align**: лево, центр, право

Элемент заголовка – 6 уровней: **<H1>...</H1>** - самый большой размер заголовка

**<H6>...</H6>** - самый малый размер заголовка

**Принудительный переход на новую строку** – только начальный тег.<BR>

**Горизонтальная линия** – <HR>

**Выделение текста полужирным шрифтом – <B>...</B>**

**Выделение текста *курсивом*** – <I>...</I>

**~~Зачеркнутое~~ начертание текста** – **<S>...</S>**

**Подчеркнутое начертание текста** – **<U>...</U>**

**<FONT>...</FONT>**  Данный контейнер определяет тип, размер и цвет шрифта.

***Теги для формирования списков:***

**<OL> и </OL>** – начало и конец **нумерованного списка**

**<UL> и </UL> –**начало и конец **маркированного списка**.

**<LI> –** **Элемент списка**

**<a href="URL"></a>** – **Гиперссылка** на другие документы/ часть текущего документа.

**<table></table>** – **Создает** таблицу.

**<tr></tr>** – Определяет **строку в таблице**.

**<td></td>** – Определяет отдельную **ячейку в таблице**.

**<th></th>** – **заголовок таблицы** (ячейка с отцентрованным жирным текстом)

**<img src="name">**- Добавляет изображение в HTML документ

## II. ВМСС

1. Состав и принцип функционирования вычислительного узла кластера (рис.).
2. Системы ОКМД: определение, пример, особенности функционирования (рис.).
3. Проблема синхронизации, средства синхронизации в многозадачных операционных системах.
4. Принцип обработки сообщений в ОС Windows (рис.).
5. Эталонная семиуровневая модель взаимодействия открытых систем. Примеры протоколов Internet, реализующих функции каждого уровня.
6. Микроконтроллеры семейства MCS51: принципы организации, особенности функционирования.

## III. ИИТ

1. Четыре группы АЦП

Широкое распространение получили четыре группы АЦП:

* АЦП поразрядного уравновешивания (АЦП последовательных приближений; кодоимпульсные АЦП). Уравновешивание происходит последовательно во времени, разряд за разрядом. Время преобразования у них – единицы и доли микросекунды. Они могут иметь до 18 двоичных разрядов. Есть два варианта таких АЦП:

– с резистивными цепочками; – на конденсаторах.

* Параллельные АЦП. У них время преобразования – наносекунды, но малое число разрядов (не более 10).
* Последовательно-параллельные АЦП (двухшаговые; конвейерные); это компромисс, благодаря которому удаётся получить до 14 разрядов при несущественном увеличении времени преобразования по сравнению с параллельными АЦП.
* «Сигма-дельта» АЦП. Они имеют рекордное число разрядов – до 24, но сравнительно большое время преобразования.

1. Погрешность квантования ЦИП

Обозначения:

X/q

0

1 2 3 4

-4 -3 -2 -1

4

3

2

1

-2

-3

-4

N

X/q

0,5

-0,5

Δкв/q

Х – измеряемая величина; q – квант;

N – число на ЦОУ без учёта запятой;

Δкв – абсолютная погрешность квантования; Δкв = Nq – X.

Ступенчатая линия на верхнем рисунке – это начальный участок идеальной зависимости «вход – выход», т.е. без учёта инструментальных погрешностей ЦИП. Пилообразная линия на нижнем рисунке – это зависимость погрешности квантования от значения измеряемой величины. Измеряемая величина и погрешность квантования отложены по осям в безразмерном виде (X/q и Δкв/q).

Из нижнего рисунка видно, что Δкв не выходит за пределы ± 0,5q; | Δкв,max | = 0,5q. Исключить или уменьшить Δкв (не считая уменьшения q) невозможно принципиально. Это говорилось об абсолютной погрешности квантования. В свою очередь

Относительная: δкв =  = ; | δкв,max | = 0,5 = = , %;

Приведённая: γкв =  = ; | γкв,max| = = , %

Здесь Хном и Nном – номинальные значения Х и N. Легко заметить, что

|δкв,max| = |γкв,max| только при Х = Хном, а при всех остальных значениях Х < Хном имеем |δкв,max| > |γкв,max|, причём |δкв,max| → ∞ при Х = 0.

**Пример.** ЦВ с конечными значениями измеряемого напряжения на данном диапазоне ± 1,19999 В измерено напряжение U = – 0,53186 В. В данном случае имеем следующее:

Uном = 1 В; q = 10 мкВ; Nном = 105; | Δкв,max| = 5 мкВ; | γкв,max| = 50/105 = 0,0005 %; | δкв,max | = 50/53186 ≈ 0,00094 %.

Таким образом, оцениваются *максимальные* значения погрешности квантования в разных формах. Если говорить обо всех остальных значениях погрешности, то вся совокупность возможных значений этой погрешности – это СВ по множеству возможных значений Х. Значения Х в пределах каждого кванта естественно принять равновероятными. Тогда эта СВ имеет равномерный закон распределения. Для абсолютной погрешности он имеет вид:

Δкв

f(Δкв)

0,5q

– 0,5q

0

Закон распределения абсолютной погрешности квантования

Здесь f(Δкв) – плотность вероятности, плотность распределения Δкв.

Из симметрии рисунка очевидно, что математическое ожидание этой случайной величины М(Δкв) = 0.

Определим дисперсию:

.

Среднеквадратическое отклонение:

.

Получается, что все характеристики погрешности квантования известны, если известны q и Nном. Эта погрешность не требует экспериментальной проверки. В этом отношении это исключительный случай в метрологии. Поэтому нормируется просто значение q. Заметим только, что нормируется номинальное значение qном. Отличие реальных квантов от номинального значения связано с присутствием инструментальных погрешностей.

1. Аддитивная и мультипликативная погрешности ЦИП

Рассмотрим аддитивную Δа и мультипликативную δм составляющие. На рис. в произвольной форме показана зависимость общей погрешности Δ от значения измеряемой величины Х. На оси абсцисс выбраны две точки: Х1 и Х2. Перпендикуляры, восстановленные из этих точек пересекают кривую линию в точках А и В. Через них проведена наклонная прямая.

А

В

Х

Δ

Х1

Х2

0

Δм

Δа

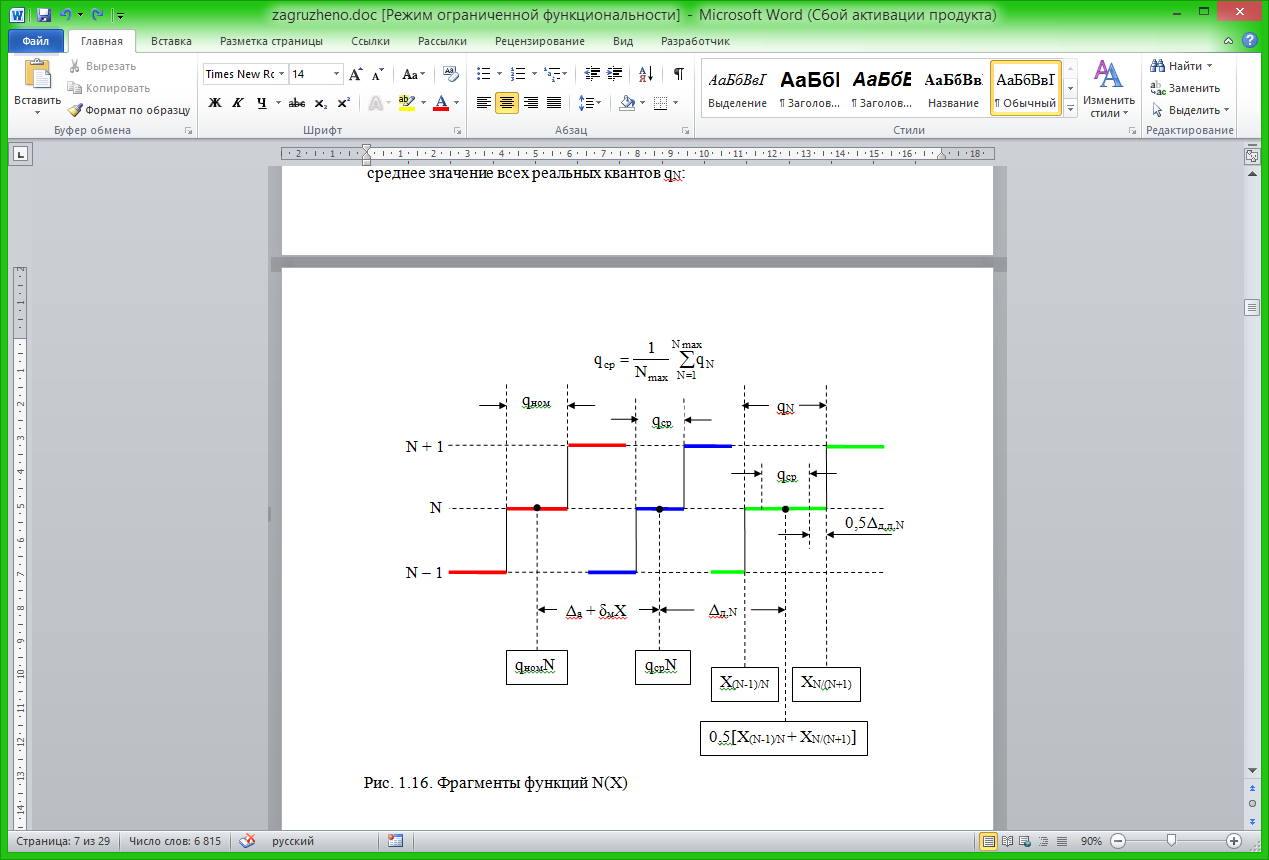
Абсолютная аддитивная погрешность Δа – это значение Δ в выбранной точке Х1. Для униполярных ЦИП, т.е. при Х ≥ 0, выбирают Х1 = 0, для биполяных, у которых обычно диапазон симметричен относительно нуля, т.е. Хmin = – Хmax, выбирают либо Х1 = 0, либо Х1 = Хmin.

В точке Х2 общая абсолютная погрешность равна сумме аддитивной и мультипликативной абсолютных погрешностей: Δ(Х2) = Δа + Δм, => Δм = Δ(Х2) – Δа. Абсолютная мультипликативная погрешность линейно зависит от Х, а относительная в этом отношении остаётся постоянной: .

Обычно выбирают Х2 = Хmax или Х2 = Хном.

Значения абсолютной аддитивной погрешности Δа и приведённой аддитивной погрешности γа, не зависят от значения Х, но значение относительной аддитивной погрешности δа обратно пропорционально Х. Значение относительной мультипликативной погрешности δм, наоборот, не зависит от значения Х, а значения Δм и γм прямо пропорциональны Х.

На рис. показаны фрагменты функции N(X): красным – без учёта инструментальных погрешностей, синим – с учётом аддитивной и мультипликативной составляющих, зелёным – с учётом всех инструментальных погрешностей.



В каждом из этих фрагментов отмечены середины горизонтальных участков ступеней, соответствующих числу N. В красном фрагменте эта середина соответствует значению Х = qномN, в синем – значению Х = qсрN, где qср – среднее значение всех реальных квантов qN:  в униполярном варианте, а в биполярном .

В зелёном фрагменте середина соответствует Х = 0,5[X(N-1)/N + XN/(N+1)], где X(N-1)/N и XN/(N+1) значения Х, при которых соответственно число (N – 1) меняется на N и N – на (N + 1).

Расстояние по горизонтали между серединами красного и синего фрагментов – это сумма аддитивной Δа и мультипликативной δмХ составляющих, а между серединами синего и зелёного – это погрешность линейности Δл,N.

Дифференциальная погрешность линейности – это разность кванта qN в данной точке N и среднего кванта qср. Т.к. средний и номинальный кванты близки – разность qN и qном: Δд,л,N = qN – qср ≈ qN – qном.

Из рис. => что самое большое отличие измеренного значения qномN от Х, т.е. предельное значение общей погрешности

Δп = qномN – Х = Δа + δмХ + Δл,N + 0,5 qN

или, учитывая, что qN = qср + Δд,л,N ≈ qном + Δд,л,N и что |Δкв,max| = 0,5 qном,

получим Δп = qномN – Х = Δа + δмХ + Δл,N + 0,5 Δд,л,N + |Δкв,max|.

Всё сказанное в принципе можно отнести не только к ЦИП, но и к АЦП, но есть существенное различие:

* в ЦИП нормируются предельные значения Δп основной и дополнительных погрешностей без разделения на составляющие;
* в АЦП нормируются предельные значения Δа; δм; Δл,N; Δд,л,N.

Причина этого различия в том, что АЦП работает в соединении с компьютером, в котором легко учесть постоянные величины Δа и δм.

Вообще же и для ЦИП и для АЦП аддитив и мультипликатив в некотором смысле «лучше» погрешностей линейности, потому что против них есть давно известное и сравнительно простое средство: выполнение операций «установка нуля» и «калибровка» (рис. 1.17).

Точечная линия на левой картинке – идеальная характеристика «вход – выход»: она проходит через начало координат и имеет номинальный наклон (картинки показаны без учёта квантования и нелинейности).

Х

N

Х

N

Х

N

Xк

Nк

0

0

0

исходная

картина

после «установки нуля»

после

«калибровки»

идеал

Рис. 1.17. Влияние операций «установка нуля» и «калибровка» на функцию N(X)

Пунктирная линия учитывает аддитив, штрихпунктирная – ещё и мультипликатив.

На средней картинке аддитив исключён установкой нуля, но мультипликатив остался. На правой картинке мультипликатив исключён калибровкой.

Последовательность выполнения операций обязательно должна быть такой, как она показана на рисунке: сначала установка нуля, потом калибровка. В противном случае потребуются многократные итерации.

Аддитив и мультипликатив, исключённые рассмотренными операциями, частично возникают вновь, потому что проходит время, изменяются внешние условия (температура и др.), но всё-таки эффект получается очень большой по сравнению с тем, как если бы этих операций вообще не выполнять.

Описанные процедуры хороши, когда предельная общая погрешность Δп составляет несколько квантов. Так обычно и бывает. Но если Δп близка к Δкв,max = ± 0,5qном, то действовать надо более тонко (рис. 1.18).

N

Nизв

Nизв-1

Nизв

Nизв – 1

1

Х

q/2

•

•

•

Хизв

Рис. 1.18. Операции «установка нуля» и «калибровка», когда Δп близка к Δкв,max

При «установке нуля» на вход подают не Х = 0, а Х = q/2 и регулировкой, называемой «установка нуля», добиваются попадания в точку, где показания N = 0 и N = 1 попеременно сменяют друг друга. После этого при калибровке на вход подаётся величина с известным значением Хизв и регулировкой, называемой «калибровка», добиваются попадания в точку, где попеременно сменяют друг друга показания (Nизв – 1) и Nизв, где Nизв – показание, соответствующее Хизв.

Из-за случайных составляющих аддитива и мультипликатива переходы между N = 0 и N = 1 и между (Nизв – 1) и Nизв могут занимать некоторые зоны (рис. 1.19):

Рис. 1.19. Влияние случайных составляющих аддитива и мультипликатива

Тогда надо стараться попасть в середины этих зон.

Из вышесказанного можно сделать следующие выводы о причинах, по которым нельзя полностью устранить аддитив и мультипликатив:

– Хк хоть и известно, но с некоторой погрешностью;

– аддитив и мультипликатив изменяются со временем;

– условия, при которых выполняются операции «установка нуля» и «калибровка» (температура и др.) отличаются от условий, при которых происходит измерение;

– в ЦИП с несколькими диапазонами калибровка производится на одном из них, а измерение – на любом

В приборах высокого класса операции «установка нуля» и «калибровка» выполняются автоматически. В этом случае для калибровки в приборе предусмотрен источник Хк.

В ЦИП со встроенными микропроцессорами вместо выполнения операций «установка нуля» и «калибровка» могут автоматически вноситься поправки в результаты измерений (рис. 1.20).

Точечная линия на рисунке соответствует идеальной характеристике ЦИП, пунктирная учитывает аддитив, сплошная – ещё и мультипликатив. Число Nк, ид = Хк/qном, идеально соответствующее Хк, постоянно находится в памяти. Число N0, возникающее при Х = 0, и число Nк, возникающее при Х = Xк, вводятся в память перед каждым измерением. Неправильное число N1, возникающее при измерении данного Х = Х1, тоже вводится в память. Исправленное число Nиспр вычисляется по числам Nк, ид; N0; Nк; N1.

Х

N

Nк

Хк

Х1

0

N0

Nк, ид

N1

Nиспр

Рис. 1.20. Внесение поправок в результаты измерений (преобразований)

*Задача 1.3. Вывести формулу для этого вычисления.*

Существуют также и другие способы снижения аддитива и мультипликатива. Но, несмотря на все различным меры, погрешности остаются, хотя и в значительно уменьшенной доле.

1. Формы нормирования погрешностей ЦИП см лекции

Нормируются пределы допускаемых значений погрешности. Если бы присутствовала только одна аддитивная погрешность, можно было бы нормировать предельные значения абсолютной погрешности ± Δп или приведённой погрешности ± γп в виде одного числа Δп или γп. Если бы присутствовала только одна мультипликативная погрешность, можно было бы нормировать предельные значения относительной погрешности ± δп в виде одного числа δп. Но обычно присутствуют и аддитивная и мультипликативная погрешности и ещё есть погрешности линейности и квантования.

Δп

Δ

– Δп

Х

На рис. зависимость суммарной погрешности и её составляющих от значения измеряемой величины. Зелёные линии отражают предельные значения аддитива, синие – суммы аддитива и мультипликатива, красные – суммы всех составляющих. Погрешностей со значениями выше верхних штрих-пунктирной линий и ниже нижних не бывает. В принципе, если мы нормируем цифровые средства измерения одним числом |Δп|, то говорим, что погрешность может лежать выше и ниже (-\*-\*-\*-). Но в этом случае, мы характеризуем прибор хуже, чем он есть на самом деле. Это не выгодно, особенно, если прибор дорогой.

Поэтому для цифровых средств измерения применяют более сложный способ нормирования – не в виде числа, а в виде функции, обычно самой простой, выражающей линейную зависимость предельно допускаемых значений погрешности от значения измеряемой величины (рис.): Δп = ± (а + b |Х|)

bX

a

– a

X

Δп

Рис. Нормирование погрешности ЦИП (Зав-ть ПДП от значения измеряемой величины)

Чаще нормируют предельно допустимые значения не абсолютной, а относительной погрешности в форме: δп = ±  %. где Хк – конечное значение Х, а с/d (условн. обозначение) - класс точности прибора. Числа «с» и «d» должны быть выбраны из ряда (1; 1,5; 2; 2,5; 4; 5; 6)10k, где k = 1; 0; – 1; – 2; …, причём c > d.

Легко показать, что c и d связаны с а и b так:

с = 100; d = 100

*Задача 1.4. Нарисуйте график функции δп(Х),* *соответствующей (1.16) и покажите на нём «с» и «d».*

Иностранные фирмы нормируют предельно допустимые значения погрешности иначе, например, так: ± (0,015 % of R + 0.01 % of FS),

где R – начальная буква слова reading – читаемое, т.е. данное прочтённое на ЦОУ значение; FS – начальные буквы слов full scale – полная шкала, т.е. предельное значение диапазона измерения.

Нормирование у нас и за границей различаются не только формой записи предельно допускаемых значений погрешности, что не существенно, но и условиями, для которых нормируются эти предельные значения. У нас, как известно, нормируется основная погрешность, т.е. погрешность в нормальных условиях (температура и др.), и дополнительные погрешности, возникающие при выходе влияющих факторов за пределы, соответствующие нормальным условиям.

За границей не пользуются понятиями основной и дополнительных погрешностей. Там нормируют, например, так:

|Δ| < |Δп,1| за 1 год при температуре (23 ± 5) 0С;

|Δ| < |Δп,2| за 3 месяца при температуре (23 ± 5) 0С;

|Δ| < |Δп,3| за 24 часа при температуре (23 ± 1) 0С.

Ясно, что |Δп,1| > |Δп,2| > |Δп,3|.

Кроме этой интегральной оценки могут быть даны отдельно сведения о временнóй о температурной нестабильности, например:

– временная нестабильность при температуре θ = const ± 1 0C за 3 месяца (долговременная) и за 24 часа (кратковременная);

– температурный коэффициент в %/0С в рабочем диапазоне от 0 до 50 0С.

1. Быстродействие ЦИП

Для ЦИП быстродействие характеризуется длительностью одного преобразования измеряемой величины Х в код числа N или частотой повторения этих преобразований. Например, для ЦВ 20 мс – длительность преобразования измеряемого напряжения U в код числа N или 50 преобразований в секунду – частота повторения преобразований. Такие данные требуются при подключении к ЦИП внешних устройств, например, компьютера. Человек не может с такой скоростью воспринимать информацию. Для человека есть ручка на передней панели прибора для плавной регулировки выдержки цифрового показания на ЦОУ (цифровое отсчетное устройство), например, от 0,5 до 5 с.

Для цифрового частотомера (ЦЧ) быстродействие характеризуется временем счёта, которое по выбору можно устанавливать 10 с; 1 с; 0,1 с… Это и есть время преобразования измеряемой частоты f в код. При времени

1 с и менее такой же ручкой можно установить удобную выдержку показания.

Заметим, что названные характеристики хоть и характеризуют быстродействие ЦИП, но не являются динамическими характеристиками: предполагается, что измеряемая величина в течении одного преобразования не изменяется во времени.

Для ЦАП и АЦП это не годится, там нужны именно динамические характеристик, т.е. реакция на изменение входной величины. Такие характеристики более сложны и мы их будем изучать после того, как узнаем о самих ЦАП и АЦП во 2й части.

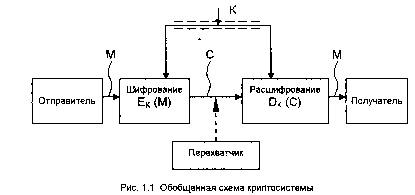
## IV. ВТ

1. Арифметические основы ЭВМ. Системы счисления. Основы алгебры логики. Функции алгебры логики (ФАЛ) и способы их задания.
2. Два направления в разработке процессоров с архитектурой RISC и CISC и особенности построения ВС на их основе. Современная элементная база вычислительных систем (ВС). Программируемые логические интегральные схемы (ПЛИС) с различной архитектурой. Проектирование и создание цифровых устройств на их основе.
3. Графические системы. Понятие стандарта на графические системы. Основные концепции. Поддержка работы с графическими устройствами в языках высокого уровня, форматы представления графических данных в информационных системах.
4. Объектно-ориентированные технологии. Технология разработки программного обеспечения. СОМ-технология. СУБД и проектирование баз данных. Распределенные базы данных. Объектно-ориентированные системы управления базами данных.
5. Имитационные модели. Сети Петри. Структурный синтез систем. Формальные и неформальные процедуры синтеза структур. VHDL — язык описания и моделирования дискретных схем.
6. Современная классификация САПР по различным критериям. Области применения САПР: САПР для подготовки чертёжно-конструкторской документации. САПР механических изделий. САПР электронных изделий. Развитие технологий САПР - CAD/CAM/CAE/PLM.

## V. ЭФИС

1. Структурная схема симметричной криптосистемы. Примеры шифров и их характеристики.

*Отправитель* генерирует ОТ исходного сообщения М, ктр должно быть передано *получателю* по незащищенному каналу. За каналом следит *перехватчик* с целью перехватить и раскрыть передаваемое сообщение. Чтобы перехватчик не смог узнать содержание сообщения М, отправитель шифрует его с помощью **обратимого преобразования Ек,** выбираемого из семейства криптографических преобразований (криптоалгоритмов), и получает ШТ(*криптограмму)* **С = Ек (М)**, ктр отправляет получателю. Получатель, приняв шифртекст С, расшифровывает его с помощью **обратного преобразования D = ЕК-1** и получает исходное сообщение в виде ОТ М: **DК (С) = ЕК-1(ЕК(М)) = М**.



К - секретный симметричный ключ, используется для шифрования и расшифрования. Т.е. это параметр, с помощью которого выбирается отдельное используемое преобразование.

Криптографическая система - однопараметрическое семейство Ек обратимых преобразований ОТ в шифрованный. Ек: М →С из пространства М сообщений ОТ в пространство С (ШТ).

**Симметричная криптосистема** – криптосистема, в которой для шифрования и расшифрования используется один ключ.

**Шифр -** совокупность обратимых преобразований множества открытых данных на множество зашифрованных данных, задаваемых К и Ek.

В практических шифрах используют 2 общих принципа: рассеивание и перемешивание.

*Рассеивание* - распространение влияния одного знака ОТ на много знаков ШТ, что позволяет скрыть статистические свойства ОТ.

*Перемешивание* - использование таких шифрующих преобразований, ктр усложняют восстановление взаимосвязи статистических свойств ОТ и ШТ ов.

Примеры:

-Алгоритм шифрования данных DES;

-Алгоритм IDEA ;

-ГОСТ 28147-89;

-стандарт шифрования с депонированием ключа EES

**Стандарт шифрования данных DES.**

**Основные достоинства алгоритма DES:**

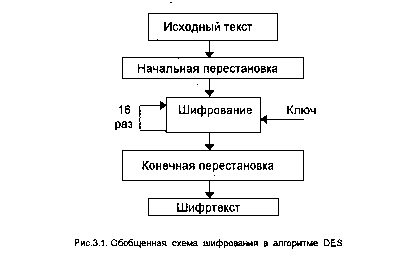
• используется только один ключ длиной 56 бит; 16 раундов при шифровании

• зашифровав сообщение с помощью одного пакета программ, для расшифровки можно использовать любой другой пакет программ, соответствующий стандарту DES;

• относительная простота алгоритма обеспечивает высокую скорость обработки;

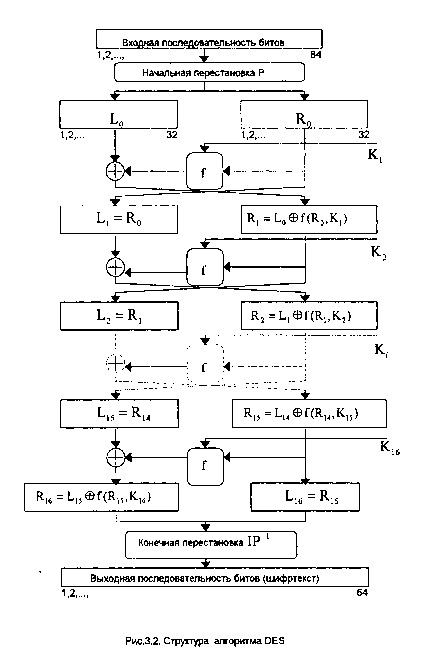
• достаточно высокая стойкость алгоритма.

Алгоритм DES использует комбинацию подстановок и перестановок. DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит - проверочные биты для контроля на четность). Дешифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности.



Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке битов.

Обозначения:



L и R - последовательности битов (левая и правая);

LR - конкатенация последовательностей L и R, т.е. такая последовательность битов, длина которой равна сумме длин L и R; в последовательности LR биты последовательности R следуют за битами последовательности L; ⊕ - операция побитового сложения по модулю 2.

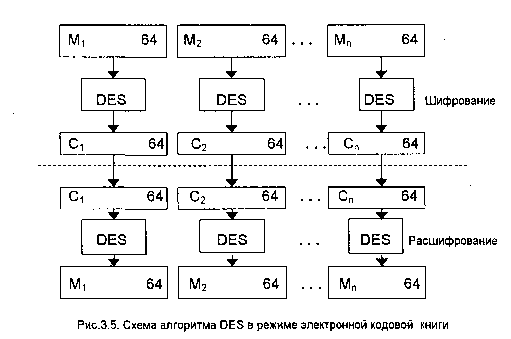
Пусть из исходного текста считан очередной 64-битовый блок T. Этот блок T преобразуется с помощью матрицы начальной перестановки IP. Биты переставляются в соответствии с этой матрицей IP. Полученная последовательность T0 разделяется на две последовательности: L0 – левый (старшие биты), R0 – правые (младшие) по 32 бита. Затем выполняется итеративный процесс шифрования, состоящий из 16 шагов (циклов). Пусть Тi – результат i-й итерации: Тi = Li Ri, где Li = t1 t2 ... t32 (первые 32 бита); Ri = t33 t34 ... t64 (последние 32 бита). Тогда результат i-й итерации описывается следующими формулами: Li = Ri–1, i = 1, 2, ..., 16; Ri = Li–1 ⊕ f (Ri–1, Ki), i = 1, 2, ..., 16.

Аргументами функции f являются последовательность Ri–1, получаемая на предыдущем шаге итерации, и 48-битовый ключ Кi, ктр является результатом преобразования 64-битового ключа шифра К.

На последнем шаге итерации получают последовательности R16 и L16 (без перестановки местами), которые конкатенируются в 64-битовую последовательность R16 L16. По окончании шифрования осуществляется восстановление позиций битов с помощью матрицы обратной перестановки IP^-1

**Режим «Электронная кодовая книга».**

Длинный файл разбивают на 64-битовые блоки. Каждый из этих блоков шифруют независимо с использованием одного и того же ключа шифрования.

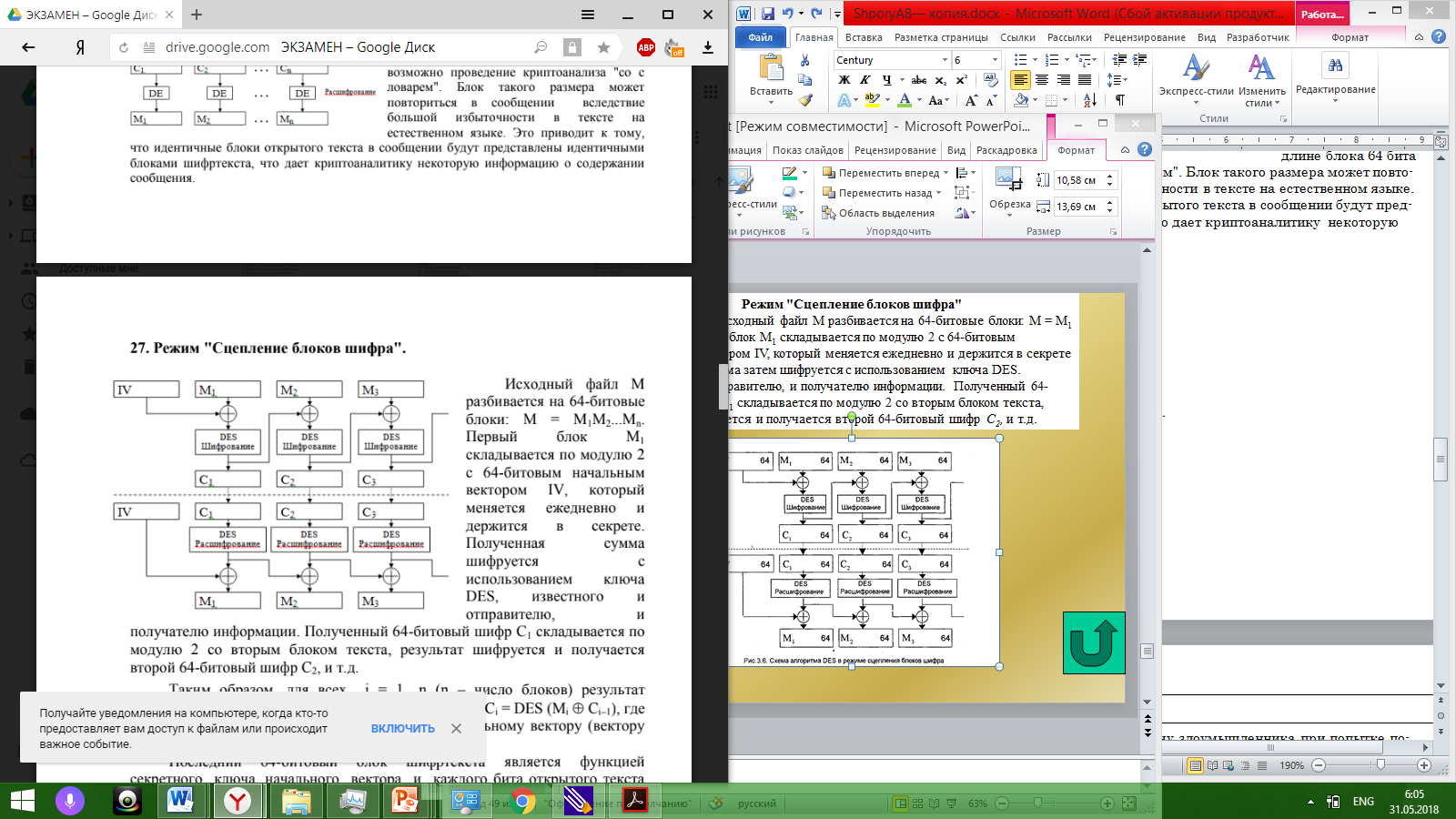


«+» простота реализации.

«-» относительно слабая устойчивость против квалифицированных криптоаналитиков. Из-за фиксированного характера шифрования при ограниченной длине блока 64 бита возможно проведение криптоанализа "со словарем". Блок такого размера может повториться в сообщении вследствие большой избыточности в тексте на естественном языке. Это приводит к тому, что идентичные блоки ОТ в сообщении будут представлены идентичными блоками ШТ, что дает криптоаналитику некоторую информацию о содержании сообщения.

**Режим «Сцепления блоков шифра».**

Исходный файл М разбивается на 64-битовые блоки: М = M1,M2...Mn. Первый блок M1 складывается по модулю 2 с 64-битовым начальным вектором IV, который меняется ежедневно и держится в секрете. Полученная сумма затем шифруется с использованием ключа DES, известного и отправителю, и получателю инф-ции. Полученный 64-битовый шифр C1 складывается по модулю 2 со вторым блоком текста, результат шифруется и получается второй 64-битовый шифр *С2,* и т.д.



Для всех i = 1...n (n – число блоков): Сi = DES (Мi ⊕ Ci–1), где С0 = IV – начальное значение шифра, равное начальному вектору (вектору инициализации). Последний 64-битовый блок ШТ - функция секретного ключа, начального вектора и каждого бита ОТ независимо от его длины. Этот блок шифртекста называют кодом аутентификации сообщения (КАС).

КАС м/б легко проверен получателем, владеющим секретным ключом и начальным вектором, путем повторения процедуры, выполненной отправителем. Посторонний не может осуществить генерацию КАС, который воспринялся бы получателем как подлинный, чтобы добавить его к ложному сообщению, либо отделить КАС от истинного сообщения для использования его с измененным или ложным сообщением.

«+» Не позволяет накапливаться ошибкам при передаче. Блок Мi является функцией только Сi–1 и Сi. Поэтому ошибка при передаче приведет к потере только двух блоков исходного текста.

**Режим «Обратная связь по шифру».**

Размер блока м/б отлична от 64 бит. Файл, подлежащий шифрованию (расшифрованию), считывается последовательными блоками длиной k битов (k =1...64).

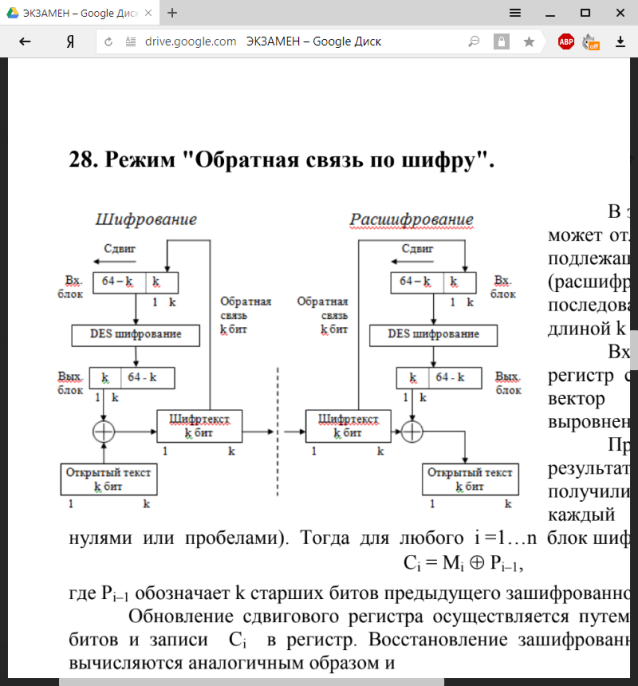


Схема шифрования:

1. Исходное сообщение (файл) разбивается на блоки длиной k битов каждый (остаток дописывается нулями или пробелами)

2. Входной блок (64-битовый регистр сдвига) вначале содержит вектор инициализации, выровненный по правому краю.

3. Входной блок подвергается DES шифрованию.

4. Полученный после шифрования блок разделяют на k старших и 64-k бит. k старших бит склад. по mod 2 с k битами ОТ. Блок, полученный после сложения, является k битовым блоком ШТ.

5. k битовый блок ШТ участвует в обновлении регистра сдвига. Из входного блока удаляются k старших битов, остальные биты сдвигаются влево и на освободившееся место записывается k битовый блок ШТ.

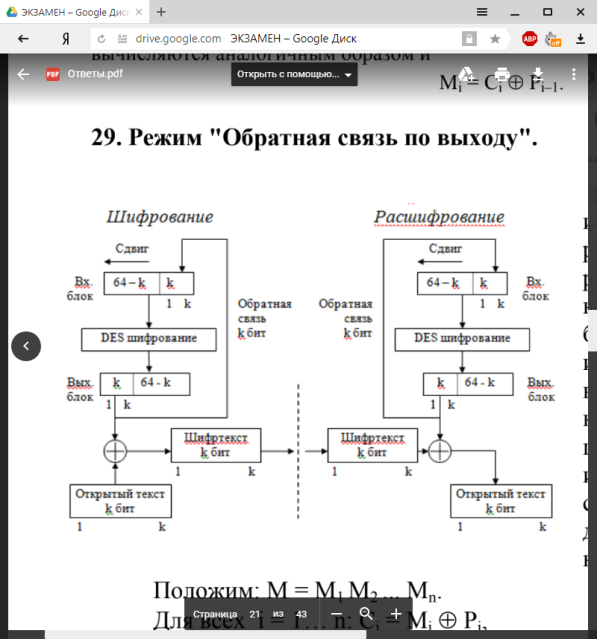
6. Процесс шифрования заканчивается, когда будут зашифрованы все блоки ОТ.

Тогда для любого i =1...n блок ШТ: Сi = Mi ⊕ Pi–1, где Рi–1 обозначает k старших битов предыдущего зашифрованного блока.

Восстановление зашифрованных данных: Рi–1 и Сi вычисляются аналогичным образом и Мi = Сi ⊕ Рi–1.

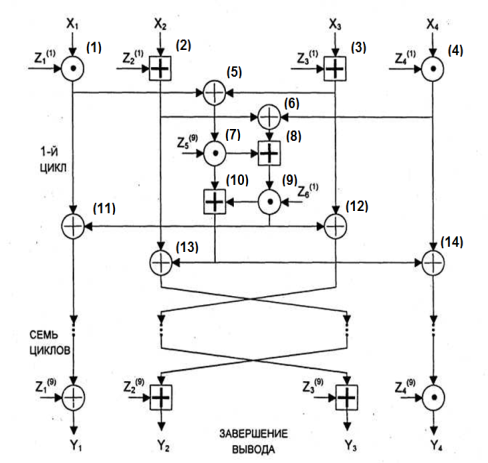
**Режим «Обратная связь по выходу».**

Алгоритм см. предыдущий. Отличие от режима ОС по шифру состоит в методе обновления сдвигового регистра. Это осуществляется путем отбрасывания старших k битов и дописывания справа Рi.



**Алгоритм шифрования IDEA -** блочный шифр. Оперирует 64-битовыми блоками ОТ. «+» Длина ключа 128 бит. Один и тот же алгоритм для шифрования и расшифрования.

**Используются мат. операции**:



• поразрядное сложение по модулю 2 ⊕;

• сложение беззнаковых целых по модулю 216 [+];

• умножение целых по модулю (216+1), рассматриваемых как беззнаковые целые, за исключением того, что блок из 16 нулей рассматривается как 216 ⊙.

• никакая пара из этих трех операций не удовлетворяет ассоциативному закону а[+](b⊕c) ≠ (a[+]b) ⊕ с; и дистрибутивному закону а[+](b⊙с) ≠(а[+] b)⊙(а[+]c).

64-битовый блок данных делится на четыре 16-битовых **субблока**. Эти четыре субблока становятся входом в первый цикл алгоритма. В каждом цикле имеет место следующая последовательность операций на рисунке ().Всего выполняется восемь циклов. Выход цикла – четыре субблока, ктр как результаты пыполнения шагов (11), (12), (13), (14). Между циклами второй и третий субблоки меняются местами (за исключением последнего цикла).

Обозначения: Xi – 16-битовый субблок ОТ i=1…4; Yi – 16-битовый субблок ШТ i=1...4; Zj^(r) – 16-битовый подключ (субблок ключа) j=1...6, r=1...9. Четыре субблоаа Y1...Y4 объединяются для получения блока ШТ.

Создание подключей Zj несложно.

Алгоритм использует всего 52 подключа (по шесть для каждого из восьми циклов и еще четыре для преобразования выхода).

Сначала 128-битовый ключ делят на восемь 16-битовых подключей. Это - первые восемь подключей для алгоритма (шесть подключей для первого цикла и первые два подключа - для второго цикла).

Затем 128-битовый ключ циклически сдвигается влево на 25 бит и снова делится на восемь подключей. Первые четыре из них используют во втором цикле; последние четыре - в третьем цикле.

Ключ снова циклически сдвигается влево еще на 25 бит для получения следующих восьми подключей и т.д., пока выполнение алгоритма не завершится.

«+» в сравнение с DES: 128-битовый ключ, внутренняя структура алг-ма IDEA обеспеч. лучшую устойчивость к криптоанализу.

1. Структурная схема асимметричной криптосистемы. Примеры шифров и их характеристики.
2. Шифры перестановки. Определение и примеры реализации.

При шифровании перестановкой символы ШТ переставляются по определенному правилу в пределах некоторого блока этого текста.

«+» Простота и легкость реализации.

«-» Сохраняется частота появления той или иной буквы в ОТ и ШТ, тем самым, проще получить ОТ из ШТ.

При достаточной длине блока, в пределах ктр осуществляется перестановка, и сложность неповторяющемся порядке перестановки можно достигнуть приемлемой стойкости шифра.

**Шифр перестановки "скитала"**

На стержень цилиндрической формы (скиталу) наматывали спиралью (виток к витку) полоску пергамента и писали на ней вдоль стержня несколько строк текста сообщения. Затем снимали со стержня полоску пергамента с написанным текстом. Буквы на этой полоске оказывались расположенными хаотично.

Такой же результат можно получить, если буквы сообщения писать не подряд, а через определенное число позиций до тех пор, пока не будет исчерпан весь текст. Для расшифрования такого ШТ нужно знать правило шифрования и обладать ключом в виде стержня определенного диаметра.

**Шифрующие таблицы** задают правила перестановки букв в сообщении.

Ключом в шифрующих таблицах используются:

• **размер таблицы;**

• **слово или фраза, задающие перестановку;**

• **особенности структуры таблицы.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Т | Р | А | Ф | О |
| Е | Е | П | А | Р |
| О | М | И | Г | А |

**Простая перестановка.** Ключ - размер таблицы. 3X5.

Сообщение записывается в таблицу по столбцам. Считывая символы из таблицы по строкам, получаем ШТ.

ОТ: ТЕОРЕМА ПИФАГОРА.

ШТ: ТРА ФОЕ ЕПА РОМ ИГА.

Объединение букв шифртекста в 3-и буквенные группы не входит в ключ шифра и осуществляется для удобства записи несмыслового текста. При расшифровании действия выполняют в обратном порядке.

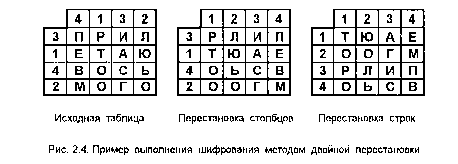
**Одиночная перестановка по ключу.** Столбцы таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| П | Р | А | В | О |
| 4 | 5 | 1 | 2 | 3 |
| Т | Р | А | Ф | О |
| Е | Е | П | А | Р |
| О | М | И | Г | А |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| А | В | О | П | Р |
| 1 | 2 | 3 | 4 | 5 |
| А | Ф | О | Т | Р |
| П | А | Р | Е | Е |
| И | Г | А | О | М |

В верхней строке таблицы записан ключ, а номера под буквами ключа определены в соответствии с естественным порядком соответствующих букв ключа в алфавите. Если бы в ключе встретились одинаковые буквы, они бы были пронумерованы слева направо. Затем записывается ОТ по столбцам, как в методе простой перестановки. В правой табл. столбцы переставлены в соответствии с упорядоченными номерами букв ключа. Считывание содержимого по строкам правой табл.

**Метод двойной перестановки.** (уточнить, записать по столбцам, считываем построчно



Перестановки определяются отдельно для столбцов и отдельно для строк. Сначала в таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, а затем строки. При расшифровании порядок перестановок должен быть обратным.

Ключ к шифру двойной перестановки – послед-ть номеров столбцов и номеров строк исходной таблицы (4132 и 3142).

**Решетка Кардано.** Формируется таблица – шаблон, состоящая из нулей и единиц. В ячейки с единицами последовательно записываются символы текста. Затем таблица поворачивается и снова записывается текст. Всего три поворота (по 90 град). В итоге получается полностью заполненная таблица. (пояснит рис.)

**Подстановка степени n.** Формируется таблица, в которой 2 строки, количество столбцов – n, где n – длина текста. В одну строку записываются случайные, неповторяющиеся числа от 1 до n, во вторую строку записывается текст. Затем столбцы переставляются в соответствии с первой строкой.

**Применение магических квадратов.** *Магический квадрат -*  квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, каждой строке и каждой диагонали одно и то же число.

Шифруемый текст вписывали в магические квадраты в соответствии с нумерацией их клеток. Если затем выписать содержимое такой таблицы по строкам, то получится шифртекст, сформированный благодаря перестановке букв исходного сообщения.

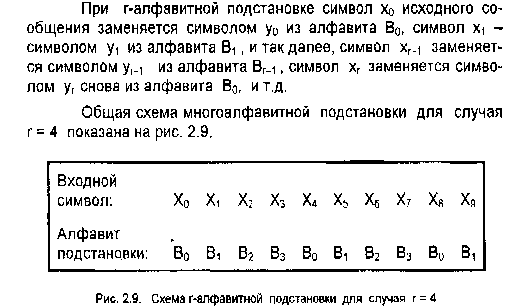
1. Шифры подстановки. Определение и примеры реализации.

В шифрах подстановки - символы ШТ заменяются символами того же или другого алфавита в соответствии с заранее обусловленной схемой замены. Они делятся на: шифры простой и сложной замены.

Шифры **простой замены** называют шифрами одноалфавитной подстановкой, т.к. каждый символ исходного текста заменяется символами того же алфавита одинаково на всем протяжении текста.

Шифры **сложной замены** называют шифрами многоалфавитной подстановки, т. к. для шифрования каждого символа исходного сообщения применяют свой шифр простой замены. Многоалфавитная подстановка последовательно и циклически меняет используемые алфавиты.

Эффект использования многоалфавитной подстановки - обеспечение маскировки естественной статистики исходного языка, т. к. конкретный символ из исходного алфавита А м/б преобразован в несколько различных символов шифровальных алфавитов Bj. **Степень обеспечиваемой защиты теоретически пропорциональна длине периода *r* в последовательности используемых алфавитов Вj.**

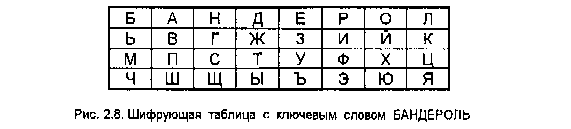


**Полибианский квадрат.**

Сначала генерируется матрица, заполненная символами используемого алфавита в случайном порядке. Далее, каждый символ исходного текста заменяется на символ, который в матрице находится ниже исходного в том же столбце. Если исходный символ находится в самой нижней строке матрицы, то он заменяется символом в самой верхней строке того же столбца.

**Шифрующие таблицы Трисемуса.**

В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. При шифровании находят в этой таблице очередную букву ОТ и записывают в ШТ букву, расположенную ниже ее в том же столбце. Если буква текста оказывается в нижней строке таблицы, тогда для шифртекста берут самую верхнюю букву из того же столбца.



Табличные шифры называются монограммными, так как шифрование выполняется по одной букве.

**Биграммный шифр Плейфейра.**

Шифрующие таблицы позволяют шифровать сразу по две буквы - шифры ***биграммные.*** Основой шифра Плейфейра является шифрующая таблица со случайно расположенными буквами алфавита исходных сообщений. Для удобства запоминания шифрующей таблицы отправителем и получателем сообщений можно использовать ключевое слово (или фразу) при заполнении начальных строк таблицы. Структура шифрующей таблицы системы Плейфейра полностью аналогична структуре шифрующей таблицы Трисемуса.

Процедура шифрования:

1. Открытый текст исходного сообщения разбивается на пары букв (биграммы). Текст должен иметь четное количество букв и в нем не должно быть биграмм, содержащих две одинаковые буквы. Если эти требования не выполнены, то текст модифицируется даже из-за незначительных орфографических ошибок.

2. Последовательность биграмм открытого текста преобразуется с помощью шифрующей таблицы в последовательность биграмм шифртекста по следующим правилам:

2а.Если обе буквы биграммы открытого текста не попадают на одну строку или столбец, тогда находят буквы в углах прямоугольника, определяемого данной парой букв (Последовательность букв в биграмме шифртекста должна быть зеркально расположенной по отношению к последовательности букв в биграмме открытого текста.)

2б.Если обе буквы биграммы открытого текста принадлежат одному столбцу таблицы, то буквами шифртекста считаются буквы, которые лежат под ними. Если при этом буква открытого текста находится в нижней строке, то для шифртекста берется соответствующая буква из верхней строки того же столбца.

2в.Если обе буквы биграммы открытого текста принадлежат одной строке таблицы, то буквами шифртекста считаются буквы, которые лежат справа от них. Если при этом буква открытого текста находится в крайнем правом столбце, то для шифра берут соответствующую букву из левого столбца в той же строке.

**Система шифрования Цезаря.**

Шифр Цезаря - частный случай шифра простой замены. При шифровании исходного текста каждая буква заменялась на другую букву того же алфавита по следующему правилу: заменяющая буква определялась путем смещения по алфавиту от исходной буквы на К=3 букв. При достижении конца алфавита выполнялся циклический переход к его началу.

Мат анализ

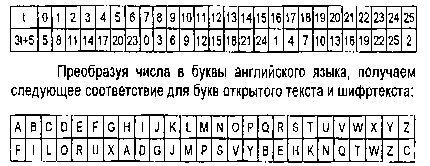
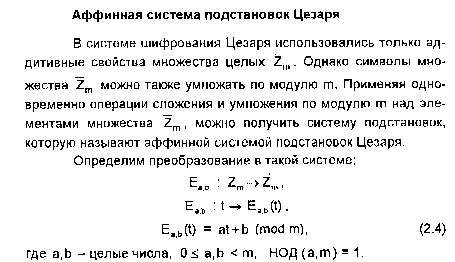
Система Цезаря - одноалфавитная подстановка, ктр шифрует n-грамму (x0,x1,x2,...,xn-1) ОТ в n-грамму (y0,y1,y2,...,yn-1) ШТ согласно следующему правилу: yi = ЕК(хi), 0  i < n, ЕК : j  (j + К) (mod m), 0  К < m, где j - числовой код буквы открытого текста; j + К - числовой код соответствующей буквы шифртекста. Система шифрования Цезаря образует по существу семейство одноалфавитных подстановок для выбираемых значений ключа К, причем 0  К < m.

«+»простота шифрования и расшифрования.

«-» подстановки не маскируют частот появления различных букв исходного ОТ; сохраняется алфавитный порядок в послед-ти заменяющих букв; при изменении значения К изменяются только начальные позиции такой последовательности; число возможных ключей К мало; легко вскрывается на основе анализа частот появления букв в шифртексте.

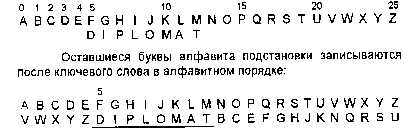
**Аффинная Система Цезаря.**

Буква, соответствующая числу **t**, заменяется на букву, соответствующую числовому значению (at+b) по модулю m. Преобразование Ea,b(t) - взаимно однозначное отображение на множестве Zm только, если НОД (а,m) = 1, т.е. а и m должны быть взаимно простыми числами. Пример, m = 26, а = 3, b = 5.НОД (3,26) = 1, тогда



**Система Цезаря с ключевым словом**

Одноалфавитная система подстановки. Особенность этой системы - использование ключевого слова для смещения и изменения порядка символов в алфавите подстановки.



Выберем некоторое число k, 0 ≤ k < 25 , и слово или короткую фразу в качестве *ключевого слова.*

Желательно, чтобы все буквы ключевого слова были различными. Пусть выбраны слово DIPLOMAT в качестве ключевого слова и число k = 5. Ключевое слово записывается под буквами алфавита, начиная с буквы, числовой код ктр совпадает с выбранным числом k:

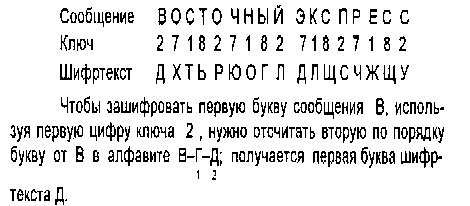
**Система омофонов.**

Шифруя букву исходного сообщения, выбирают случайным образом одну из ее замен. Замены (часто называемые омофонами) могут быть представлены трехразрядными числами от 000 до 999. Если омофоны (замены) присваиваются случайным образом различным появлениям одной и той же буквы, тогда каждый омофон появляется в шифртексте равновероятно.

**Шифр Гронсфельда.**

Шифр сложной замены. Есть модификация шифра Цезаря с числовым ключом. Для этого под буквами исходного сообщения записывают цифры числового ключа. Если ключ короче сообщения, то его запись циклически повторяют. ШТ получают примерно, как в шифре Цезаря, но отсчитывают по алфавиту не третью букву, а выбирают ту букву, которая смещена по алфавиту на соответствующую цифру ключа.

«-»: в числовом ключе каждая цифра имеет только десять значений, а значит, имеется лишь десять вариантов прочтения каждой буквы ШТ.

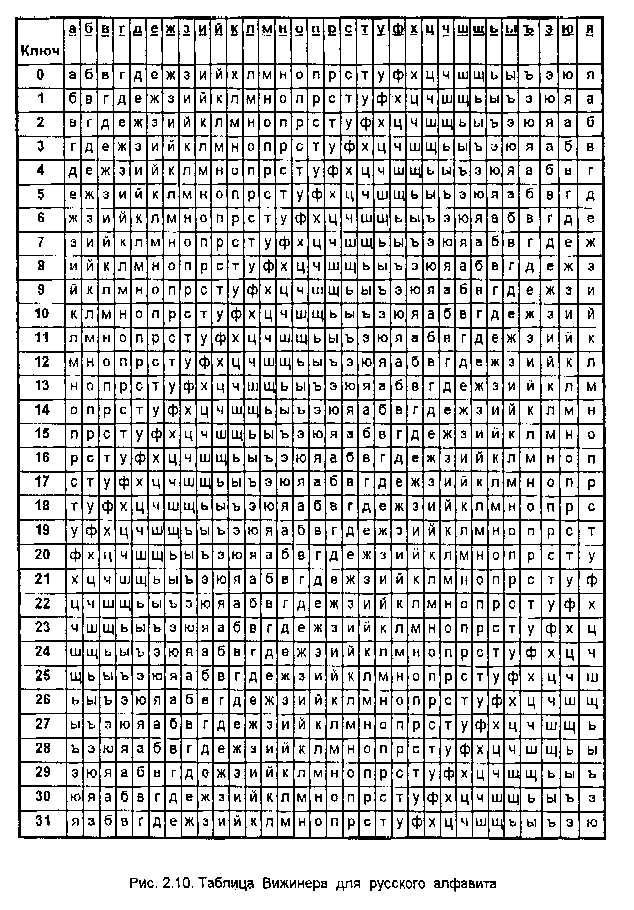


«+»: этот шифр допускает дальнейшую модификации, улучшающие его стойкость, в частности двойное шифрование разными числовыми ключами.

**Система шифрования Вижинера.**

Многоалфавитная. Она подобна системе шифрования Цезаря, у которой ключ подстановки меняется от буквы к букве.

Табл. Вижинера используется для зашифрования и расшифрования и имеет два входа:

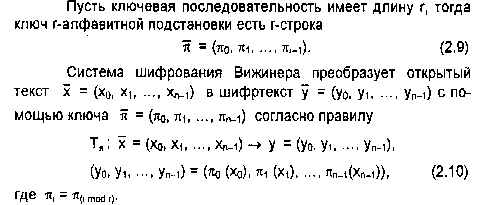


• верхнюю строку \_ символов, используемую для считывания очередной буквы исх-ого ОТ;

• крайний левый столбец ключа;

Последовательность ключей получают из числовых значений букв ключевого слова.

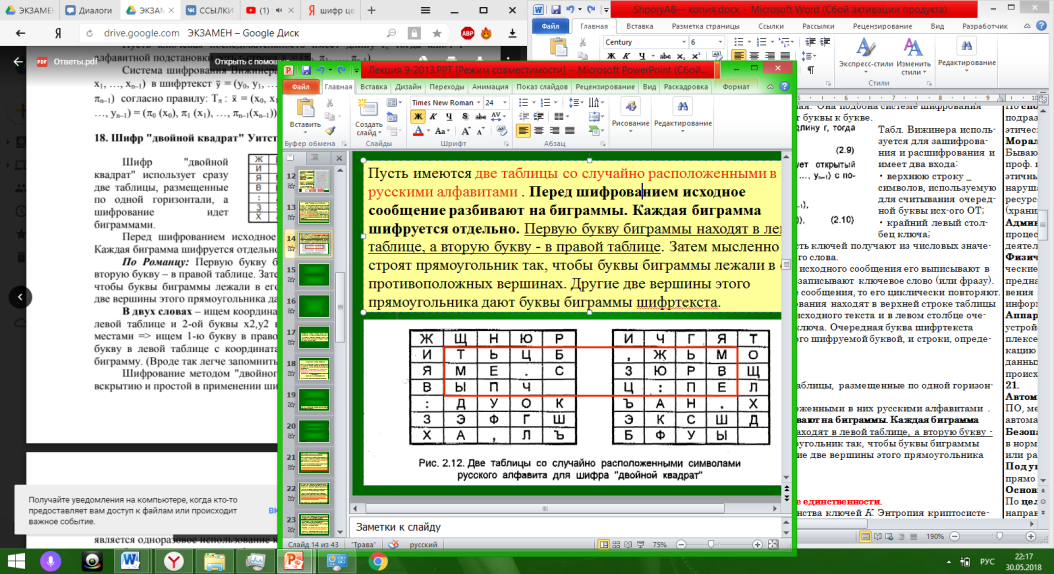
При шифровании исходного сообщения его выписывают в строку, а под ним записывают ключевое слово (или фразу). Если ключ короче сообщения, то его циклически повторяют. В процессе шифрования находят в верхней строке таблицы очередную букву исходного текста и в левом столбце очередное значение ключа. Очередная буква шифртекста находится на пересечении столбца, определяемого шифруемой буквой, и строки, определяемой числовым значением ключа.



**Шифр «двойной квадрат» Уитстона.**

Используется две таблицы со случайно расположенными в них буквами алфавитами, размещенные по одной горизонтали, а шифрование идет биграммами. **Перед шифрованием исходное сообщение разбивают на биграммы. Каждая биграмма шифруется отдельно.**

1) Первую букву биграммы находят в левой таблице, а вторую букву - в правой таблице. Строят прямоугольник так, чтобы буквы биграммы лежали в его противоположных вершинах. Другие две вершины этого прямоугольника дают буквы биграммы шифртекста, причем, первая буква - из правой таблицы, а вторая – из левой таблицы. ТЕ -> МЫ**.**



2) Если обе буквы биграммы сообщения лежат в одной строке. то и буквы шифртекста берут из этой же строки.

**Первую букву биграммы ШТ** берут из левой таблицы в столбце, соответствующем второй букве биграммы сообщения.

**Вторая же буква биграммы ШТ** берется из правой таблицы в столбце, соответствующем первой букве биграммы сообщения. TO -> БЖ.

***Через координаты***

1) Ищем координаты (столбец, строка) 1-ой буквы x1,y1 в левой таблице и 2-ой буквы x2,y2 в правой таблице => меняем x1 и x2 местами => ищем 1-ю букву в правой таблице с координатами x2,y1 и 2ю букву в левой таблице с координатами x1,y2 => получаем шифрованную биграмму.

2) Ищем координаты (столбец, строка) 1-ой буквы x1,y в левой таблице и 2-ой буквы x2,y в правой таблице (y=y1=y2) => меняем x1 и x2 местами => ищем 1-ю букву в левой таблице с координатами x2,y и 2ю букву в правой таблице с координатами x1,y => получаем шифрованную биграмму.

1. Структура автоматизированной системы обработки информации (АСОИ). Основные угрозы безопасности компонентам АСОИ. Понятие несанкционированного доступа.

**Автоматизированная система обработки информации (АСОИ)** – совокупность тех. средств, ПО, методов обработки информации и действия персонала, ктр обеспечивает выполнение автоматизированной обработки информации.

**Безопасность АСОИ** – её защищенность от случайного и преднамеренного вмешательства в нормальный процесс её функционирования, а также от попыток хищения, изменения или разрушения её компонентов.

**Под угрозой безопасности АСОИ** понимаются возможные воздействия на АСОИ, которые прямо или косвенно могут нанести ущерб её безопасности.

**Основные угрозы безопасности АСОИ**.

По **цели воздействия**:

1) угрозы нарушения **конфиденциальности** информации, ктр направлены на разглашение конфиденциальной или секретной информации. При реализации этих угроз информация становится известной лицам, которые не должны иметь к ней доступ;

2) угрозы нарушения **целостности** информации, хранящейся в компьютерной системе или передаваемой по каналу связи, направлены на её изменение или искажение, приводящее к нарушению её качества или полному уничтожению; они актуальны для систем ПИ;

3) угрозы нарушения **работоспособности** системы (отказы в обслуживании) направлены на создание таких ситуаций, когда определенные преднамеренные действия либо снижают работоспособность АСОИ, либо блокируют доступ к некоторым её ресурсам.

По **характеру воздействия**:

1) Случайные угрозы (отключение питания, отказы и сбои аппаратуры, ошибки в ПО, ошибки в работе обсуживающего персонала и пользователей, помехи в линиях связи из-за воздействий внешней среды);

2) Преднамеренные угрозы связаны с целенаправленными действиями нарушителя. Мотивы нарушителя и модеь см. екции.

По **компонентам АСОИ**, на которые угрозы нацелены:

1) аппаратные средства – ЭВМ и их составные части;

2) ПО – приобретённые программы, исходные модули, загрузочные модули и т.д;

3) Данные – хранимые временно и постоянно на магнит. носит., печатные, архивы и т.д;

4) Персонал – обслуживающий персона или пользователи;

По **расположению источника угроз:** внутри или вне рассматриваемой АСОИ.

**НСД. Основные каналы несанкционированного доступа.**

**Доступ к информации** – возможность ознакомления с информацией, в частности копирование, модификация или уничтожение информации. Доступ к информации бывает санкционированным (СД) и несанкционированным (НСД).

**Несанкционированный доступ (НСД)** является наиболее распространённым и многообразным видом компьютерных нарушений. Суть НСД состоит **в получении пользователем (нарушителем) доступа к объекту в нарушение правил разграничения доступа**, установленных в соответствии с принятой политикой безопасности. НСД использует **любую ошибку в системе защиты** и возможен при **нерациональном выборе средств защиты, их некорректной установке и настройке**.

НСД может быть осуществлён как **штатными средствами АСОИ, так и специально созданными аппаратными и программными средствами**.

**Основные каналы НСД**, через которые нарушитель может получить доступ к компонентам АСОИ и осуществить хищение, модификацию и/или разрушение информации:

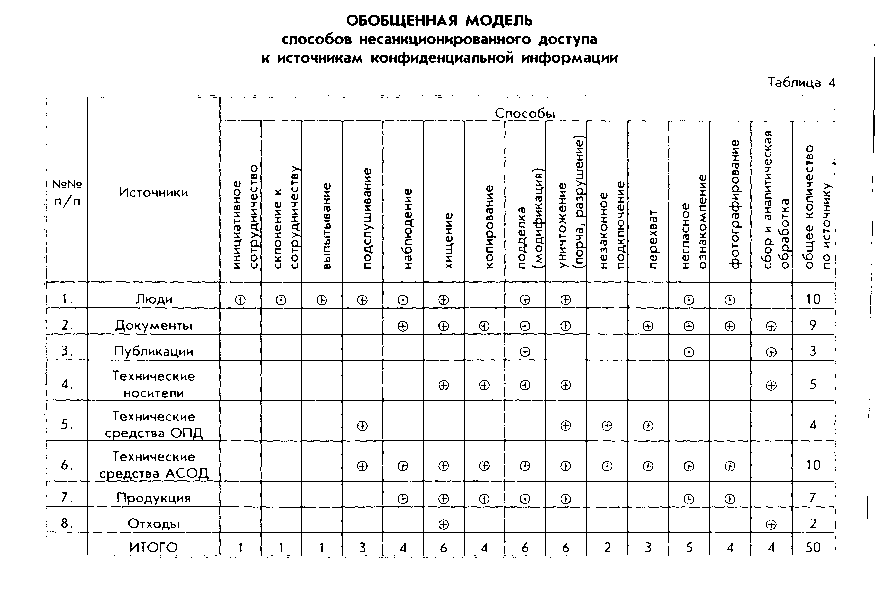
**1)** Все штатные каналы доступа к информации (терминалы пользователей, оператора, администратора системы; средства отображения и документирования информации; каналы связи) при их использовании нарушителями, а также законными пользователями вне пределов их полномочий.

**2)** Технологические пульты управления.

**3)** Линии связи между аппаратными средствами АСОИ.

**4)** Побочные электромагнитные излучения от аппаратуры, линий связи, сетей электропитания и заземления.

**Обобщен. модель способов НСД к источникам конфиденциальной инф-ции**



1. Структура системы безопасности АСОИ. Назначение и функции подсистем. Классы защищенности АСОИ.

**Автоматизированная система обработки информации (АСОИ)** – совокупность тех. средств, ПО, методов обработки информации и действия персонала, ктр обеспечивает выполнение автоматизированной обработки информации.

**Безопасность АСОИ** – её защищенность от случайного и преднамеренного вмешательства в нормальный процесс её функционирования, а также от попыток хищения, изменения или разрушения её компонентов.

**Структура системы без-ти АСОИ. Основ. функции подсистем без-ти**

Схему надо функции подсистем безопасности АСОИ:

1. Подсистема **управление доступом**: идентификация субъектов; проверка подлинности субъектов; контроль доступа субъектов к объектам доступа.

2. Подсистема **регистрации и учета**: регистрация входа субъектов в систему; запуск и завершение программ и процессов; изменение полномочий субъектов доступа; учет защищаемых объектов доступа.

3. Подсистема **криптографической защиты**: шифрование конфиденциальной информации; шифрование информации, принадлежащей разным субъектам доступа на разных ключах; использование криптографических средств.

4. Подсистема **обеспечения целостности**: обеспечение целостности программных средств и данных; физическая охрана средств ВТ и носителей информации; тестирование средств защиты информации на НСД; восстановление средств защиты информации на НСД.

**Классы защищенности АСОИ.**

**Признаки**, по ктр производится группировка АС в различные классы:

1) Наличие в АС инф-ии различного уровня конфиденциальности.

2) Уровень полномочий субъектов доступа АС на доступ к конфиденциальной инф-ции.

3) Режим обработки данных в АС коллективный или индивидуальный.

**Установлено 9 классов защищённости** АСОИ от НСД к информации, распределённых по 3 группам. Класс, соответствующий высшей степени защищенности для данной группы, обозначается индексом NА, где N – номер группы, далее NБ и т.д. три номера всего

**Третья группа** включает АС, в которых работает **один пользователь**, допущенный ко всей инф-ии АС, размещённой на носители одного уровня конфиденциальности. Группа содержит 2 класса 3Б и 3А.

**Вторая группа** включает АС, в которых пользователи имеют **одинаковые полномочия доступа** ко всей инф-ии, обрабатываемой и хранимой в АС на носителях различного ур-ня конфиденциальности. Группа содержит 2 класса 2Б и 2А.

**Первая группа** включает **многопользовательские** АС, в которых одновременно обрабатывается и хранится инф-ия разных ур-ней конфиденциальности. Не все пользователи имеют равные права доступа. Группа содержит классы 1Д, 1Г, 1В, 1Б, 1А.