**Экзамен ЦУиМП**

Оглавление

[1. Предметодье (как предисловие) 4](#_Toc131316580)

[2 Комбинационная логика 5](#_Toc131316581)

[2.1 Цифровые элементы (О, ТИ, АВ) 6](#_Toc131316582)

[2.2 Арифметические операции с числами 7](#_Toc131316583)

[2.3 Вещественные числа. Хранение вещественного числа в компьютере 8](#_Toc131316584)

[2.4 Перевод чисел 9](#_Toc131316585)

[2.5 Таблицы истинности и алгебраические выражения 10](#_Toc131316586)

[2.6 Карты Карно 11](#_Toc131316587)

[2.7 Метод Квайна 12](#_Toc131316588)

[2.8 Базис 2И-НЕ 13](#_Toc131316589)

[2.9 Базис 2ИЛИ-НЕ 14](#_Toc131316590)

[2.10 Шифратор (О, ТИ, АВ) 15](#_Toc131316591)

[2.11 Дешифратор 16](#_Toc131316592)

[2.12 ССИ 17](#_Toc131316593)

[2.13 Приоритетный шифратор (О?, ТИ) 18](#_Toc131316594)

[2.14 Дешифратор матричного типа 19](#_Toc131316595)

[2.15 Каскадное включение дешифраторов 20](#_Toc131316596)

[2.16 Мультиплексор (О, ТИ, АВ) 21](#_Toc131316597)

[2.17 Демультиплексор (О, ТИ, АВ) 22](#_Toc131316598)

[2.18 Мажоритарный элемент (О?, ТИ, АВ) 23](#_Toc131316599)

[2.19 Сумматор и полусумматор 24](#_Toc131316600)

[2.20 Полувычитатель и вычитатель 25](#_Toc131316601)

[2.21 Умножители. Умножитель 2х2, 3х3, 4х4 26](#_Toc131316602)

[2.22 Компаратор (О, ТИ, АВ) 27](#_Toc131316603)

[2.23 Последовательный сдвиговый регистр (Barrel shifter) 28](#_Toc131316604)

[2.24 АЛУ 29](#_Toc131316605)

[2.25 Статический сбой (static hazart) 30](#_Toc131316606)

[2.26 Вопросы на подумать 31](#_Toc131316607)

[2.26.1 ИСКЛ ИЛИ на 4 элементах (базис 2И-НЕ). ИСКЛ ИЛИ в базисе 2ИЛИ-НЕ 32](#_Toc131316608)

[2.26.2 Вычитатель на сумматоре 33](#_Toc131316609)

[2.26.3 Сложите 4-5 одноразрядных чисел. Есть восьмиразрядное число, определить количество единиц в нём 34](#_Toc131316610)

[2.26.4 Компаратор на сумматорах. 35](#_Toc131316611)

[2.26.5 Вариант 1 РК1 36](#_Toc131316612)

[2.26.6 Вариант 3 РК1 37](#_Toc131316613)

[2.26.7 Вариант 4 РК1 38](#_Toc131316614)

[2.26.8 Вариант 7 РК1 39](#_Toc131316615)

[3 Последовательстная логика (логика с памятью) 40](#_Toc131316616)

[3.1 RS-триггер, статический, асинхронный, одноступенчатый 41](#_Toc131316617)

[3.2 RS-триггер, синхронный, статический, одноступенчатый 42](#_Toc131316618)

[3.3 D-триггер (D flip-flop), синхронный, статический, одноступенчатый 43](#_Toc131316619)

[3.4 Динамический D-триггер, синхронный, двухступенчатый 44](#_Toc131316620)

[3.5 T-триггер (счётный) 45](#_Toc131316621)

[3.6 Асинхронный счётчик на D-триггерах. 46](#_Toc131316622)

[3.7 Динамический RS-триггер 47](#_Toc131316623)

[3.8 Вычитающий счётчик (асинхронный) 48](#_Toc131316624)

[3.9 Реверсивный счётчик 49](#_Toc131316625)

[3.10 Счётчик с предустановленными значениями 50](#_Toc131316626)

[3.11 Кольцевой счётчик (синхронный) 51](#_Toc131316627)

[3.12 Кольцевой счётчик Джонсона 52](#_Toc131316628)

[3.13 Кольцевой счётчик (ходит только одна единица) 53](#_Toc131316629)

[3.14 Синхронный счётчик вариант 1 54](#_Toc131316630)

[3.15 Синхронный счётчик вариант 2 55](#_Toc131316631)

[3.16 Регистры. Последовательно-параллельные 56](#_Toc131316632)

[3.17 Регистры. Параллельно-последовательные 57](#_Toc131316633)

[3.18 Регистры параллельно-параллельные 58](#_Toc131316634)

[3.19 Регистры. Циклический регистр. Выбор направления. Предустановка значений. 59](#_Toc131316635)

[3.20 Динамический одноступенчатый D-триггер 60](#_Toc131316636)

[3.21 Динамический D-триггер с асинхронным сбросом, установкой 61](#_Toc131316637)

[3.22 JK-триггер, синхронный, статический 62](#_Toc131316638)

[3.23 Динамический JK-триггер 63](#_Toc131316639)

[3.24 Динамический JK-триггер на D-треггерах 64](#_Toc131316640)

[3.25 T-триггер на JK-триггере 65](#_Toc131316641)

[3.26 Синхронный счётчик на JK-триггере 66](#_Toc131316642)

[3.27 Сброс триггера. Асинхронные, синхронные входы. 67](#_Toc131316643)

[3.28 Задачи на подумать. 68](#_Toc131316644)

[3.28.1 Счётчик в базисе 2И-НЕ 69](#_Toc131316645)

[3.28.2 Имеется кольцевой счётчик. Как загнать «1» или несколько «1» и двигать их по кругу? 70](#_Toc131316646)

# Предметодье (как предисловие)

*Я никогда не учу своих учеников. Я только даю условия, при которых они могут сами учиться*

(Эйнштейн)

*Без примеров невозможно ни правильно учить, ни успешно учиться*

(Пацанский паблик ВК)

Целью данной методы не является создать методу ради методы, хочется искренне всё понять, а это возможно только в том случае, если сделать всё самому. Поэтому при создании вашей части методы старайтесь «научить» читателя чему-то, заставить его попробовать реализовать что-то самостоятельно, ведь только в этом случае можно проникнуться всей красотой цифровой электроники и микропроцессоров, и сделать шаг к торжеству автоматизации рутинных задач и процессов.

Обозначения и сокращения:

* О – обозначение
* ТИ – таблица истинности
* АВ – алгебраические выражения
* ВД – временная диаграмма.

# Комбинационная логика

Шо це такое и зачем вообще нужно?

## Цифровые элементы (О, ТИ, АВ)

Инвертор, 2или, 2и, XOR (разные виды представления XOR)

## Арифметические операции с числами

что такое позиционные и непозиционные коды? Прямой, обратный, доп код, перевод из одного в другой, для каких чисел какие коды существуют, умножение, сложение, вычитание бинарых чисел.

## Вещественные числа. Хранение вещественного числа в компьютере

1. Алгоритм представления вещественного числа (числа с плавающей точкой) в компьютере:

1) Перевод числа в двоичную (бинарную) систему счисления.

2) Представление числа в нормализованной записи.

3) Нахождение степени.

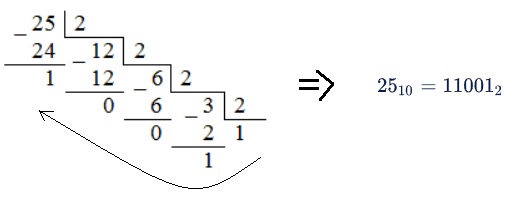
4) Размещение знака, степени и мантиссы в соответствующие разряды.

Рассмотрим этот процесс на примере.

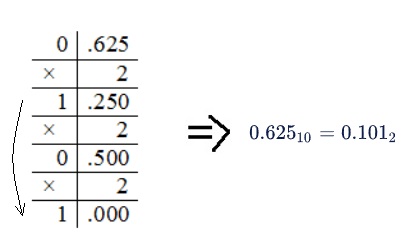
Дано число в десятичной системе счисления -26.625

1) Перевод в bin систему счисления:

а) Переведем целую часть числа:



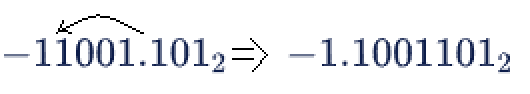
б) Переведем дробную часть числа:



в) Полученное число в бинарном виде:



2) Необходимо сместить точку к «первой» единице в числе:

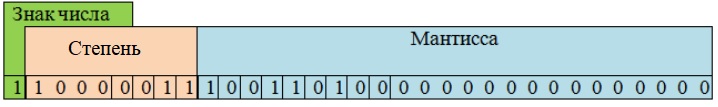


Отсюда получаем , т.к. сместили точку на 4 знака. Мантиссой будет являться часть полученного числа после точки.

3) Степень находим из следующего выражения:

Представляем степень тоже в бинарном виде:

4) Распределяем знак, степень и мантиссу в соответствующие разряды.

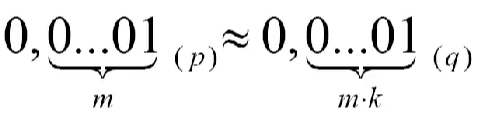


2. Рассмотрим следующую задачу: Задано число, нужно с точностью до Х знаков перевести в двоичную систему счисления. Как узнать число двоичных чисел после запятой, которых будет достаточно для заданной точности?

При переводе числа из десятичной системы счисления с двоичную необходимо ограничить число знаков после запятой при этом сохранив необходимую точность. Согласно теории погрешностей, точность числа считается равной единице младшего знака (разряда).

Так, например, точность для исходного числа 0,02 равна 0,01. Чтобы узнать какой же знак двоичной записи примерно равен 0,01 можно воспользоваться выражением . Отсюда . Отсюда следует что необходимо сохранить 7 знаков после запятой, чтобы точность представления числа не уменьшалась.

В общем случае:

**

Где m – требуемая точность *p*-ичной дроби;

k – показатель, характеризующий степенную зависимость между *p* и *q*, т.е. .

В нашем случае . Отсюда следует Округлив, получаем 7, что и было получено выше.

## Перевод чисел

Hex, bin, dec. Почему hex код удобен?

## Таблицы истинности и алгебраические выражения

Что такое комбинационная схема? Что такое СКНФ и СДНФ? Принципы составления СКНФ И СДНФ по таблице истинности. Как составить таблицу истинности по алгебраическому выражению? Основные законы алгебры логики (минимизации).

## Карты Карно

Причём тут код Грэя? Перевод бинарных чисел в код Грэя и в обратную сторону. Принцип составления карт Карно для СКНФ и СДНФ. Как можно объединять 1 (0) в ячейках? Что даст диагональное расположение 1 (0)?

## Метод Квайна

Смысл метода, инструкция

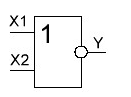
## Базис 2И-НЕ

Зачем нужны базисы? Основные цифровые элементы в базисе 2И-НЕ. Перевод АВ в базис 2И-НЕ.

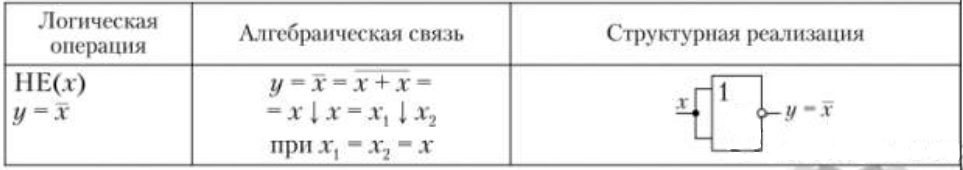
## Базис 2ИЛИ-НЕ

Необходимость базисов заключается в следующем: цифровое устройство, выполняющее любую логическую функцию, можно построить, имея элементы только одного вида («ИЛИ-HE» либо «И-НЕ»). При интегральной технологии с точки зрения надежности и стоимости использование однотипных элементов весьма предпочтительно.

Элемент 2ИЛИ-НЕ:



Реализация простейших логических операций в базисе 2ИЛИ-НЕ:



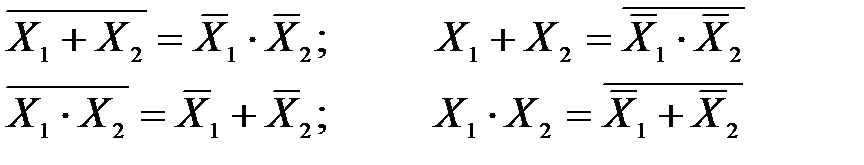


Для перевода АВ в базис 2ИЛИ-НЕ необходимо:

1) Наложить двойную инверсию.

2) Применить теорему де Моргана.

Теорема де Моргана заключается в следующем:



## Шифратор (О, ТИ, АВ)

Что это и зачем нужно? Можно ли использоваться комбинацию входных сигналов с несколькими 1? Обобщённая схема. Схема в базисе 2ИЛИ-НЕ, 2И-НЕ

## Дешифратор

Что это и зачем нужно. Обобщённая схема. Схема в базисе 2ИЛИ-НЕ, 2И-НЕ

## ССИ

Устройство, ТИ, Схемы с ОК и ОА. Пример составления выражения для одного индикатора, его минимизации и перевод в базис (любой), схема для индикатора.

## Приоритетный шифратор (О?, ТИ)

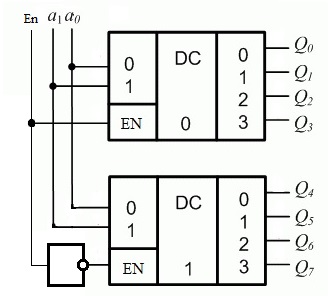
Схема

## Дешифратор матричного типа

Что это, зачем и схема

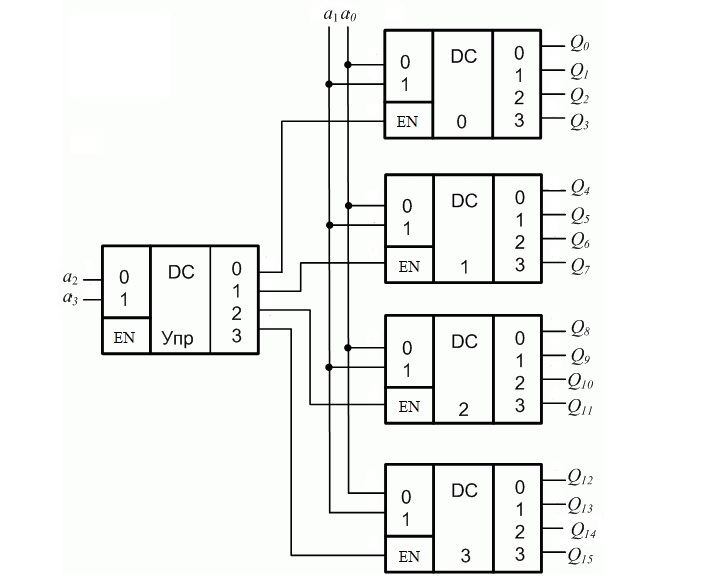
## Каскадное включение дешифраторов

Необходимость каскадного соединения нескольких дешифраторов возникает в том случае, когда разрядность одной ИС оказывается недостаточной для адресации большого количества различных устройств. Например, в нашем распоряжении схемы дешифратора на 2 входа и, соответственно, 4 выхода. А необходимо организовать возможность обращения к 8 цифровым устройствам. Тогда можем использовать схему каскадного включения двух таких дешифраторов:



Как видим, с помощью EN мы можем выбрать четверку устройств из которых мы будем производить свой выбор в дальнейшем при подаче *a0* и *a1* .

Рассмотрим случай, когда нам необходимо иметь возможность обращения к 16-ти цифровым устройствам, тогда можем использовать следующую схему каскадного включения:



В данном случае используется 5 дешифраторов, один из которых (управляющий) используется для выбора четверки устройств, из которых будет производиться дальнейший выбор с помощью оставшихся дешифраторов.

## Мультиплексор (О, ТИ, АВ)

Что это, зачем? Обобщённая схема, схема в базисах 2И-НЕ, 2ИЛИ-НЕ

## Демультиплексор (О, ТИ, АВ)

Что это, зачем? Обобщённая схема, схема в базисах 2И-НЕ, 2ИЛИ-НЕ

## Мажоритарный элемент (О?, ТИ, АВ)

Что это, зачем? Обобщённая схема, схема в 2И-НЕ, 2ИЛИ-НЕ.

## Сумматор и полусумматор

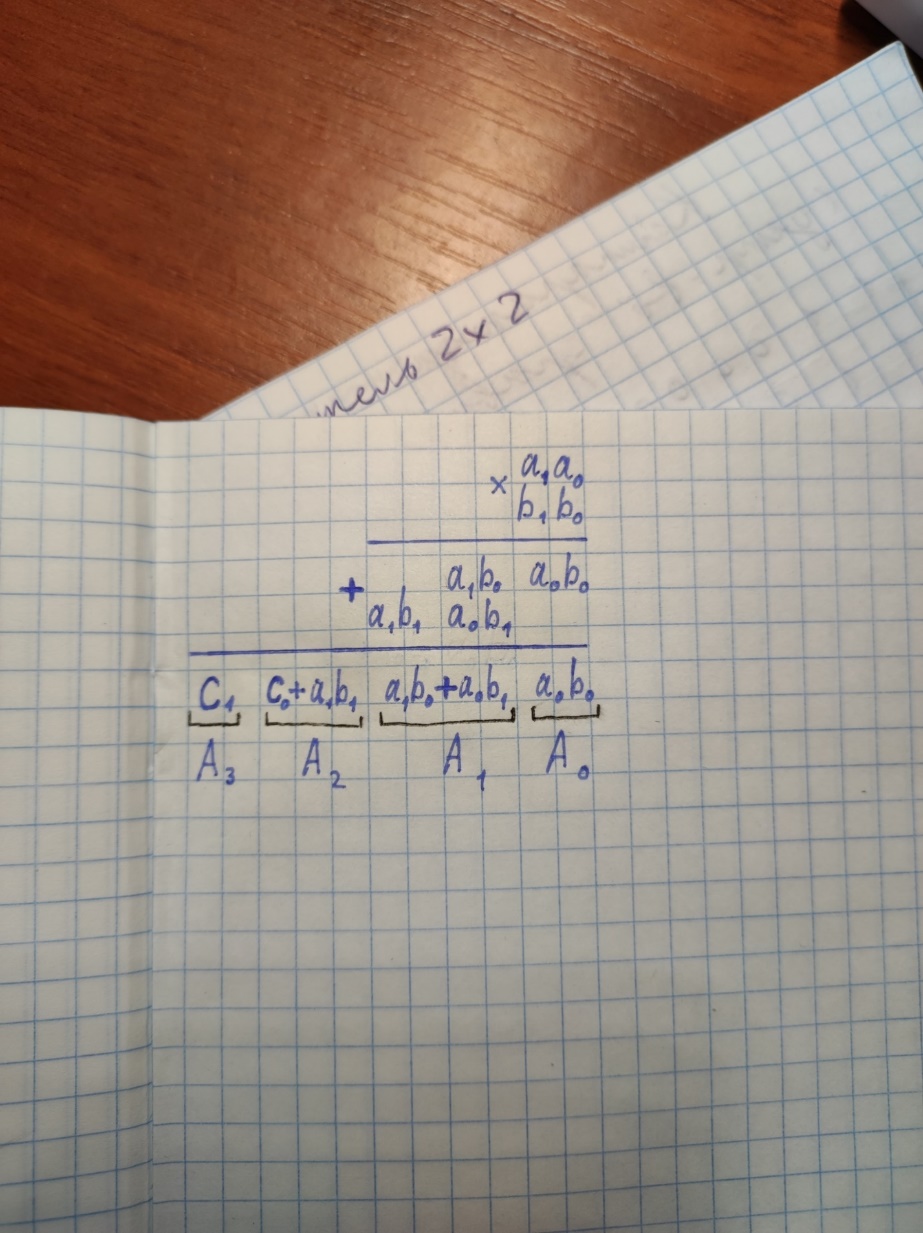
В чем разница (Н-р, с помощью сложения двух бинарных чисел столбиком)? О, ТИ, АВ, обобщённая схема, схема в базисах 2И-НЕ, 2ИЛИ-НЕ. Сумматор из полусумматоров? Каскадное включение сумматоров.

## Полувычитатель и вычитатель

В чем разница (Н-р, с помощью вычитания двух бинарных чисел столбиком)? О, ТИ, АВ, обобщённая схема, схема в базисах 2И-НЕ, 2ИЛИ-НЕ. Вычитатель из полувычитателей? Каскадное включение вычитателей.

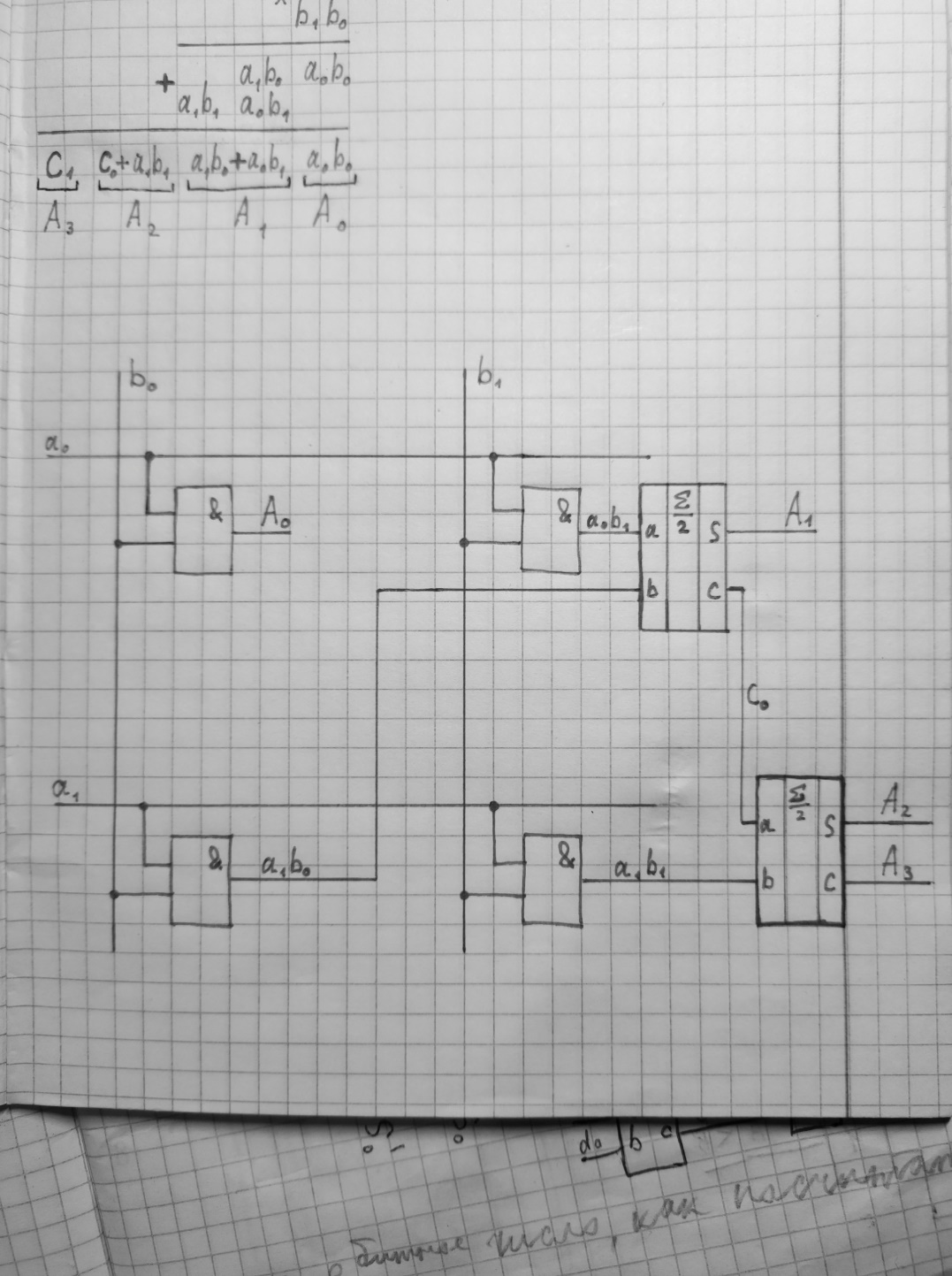
## Умножители. Умножитель 2х2, 3х3, 4х4

Рассмотрим умножитель 2x2.

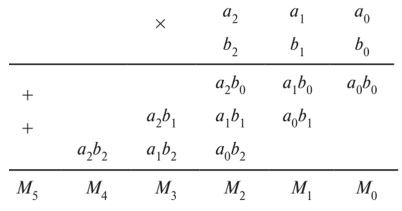


Переносы *c1* и *c0* появляются благодаря суммированию при получении элементов *А2* и *А1* соответственно.

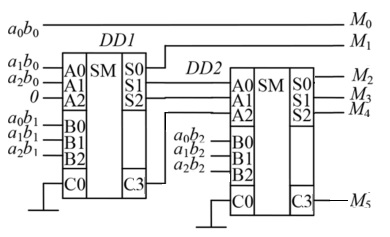
Составим схему соответствующего умножителя 2х2:



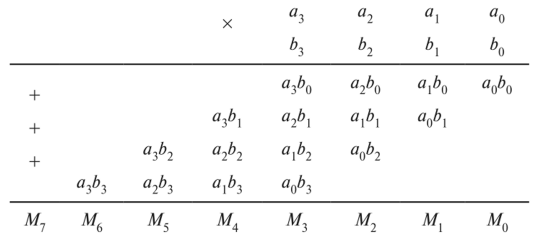
Рассмотрим алгоритм умножения двух трехразрядных чисел:



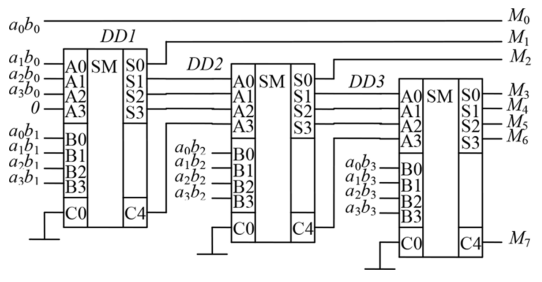
Построим схему умножителя 3х3 на трехразрядных сумматорах:



Рассмотрим алгоритм умножения двух четырехразрядных чисел:



Построим схему умножителя 4х4 на четырехразрядных сумматорах:



## Компаратор (О, ТИ, АВ)

Зачем нужен, принцип работы? Последовательный компаратор. А есть параллельный (на лекциях ничего не написали, нужно почитать)?

## Последовательный сдвиговый регистр (Barrel shifter)

## АЛУ

Написали про них мало, но возможно стоит поискать более подробную общую информацию, красиво все описать, а не как в лекции одной картинкой, в которую еле всё поместилось и ниче не понятно.

## Статический сбой (static hazart)

Пример на какой-то схеме, н-р, как в лекции, но если будут другая, то будет только +

## Вопросы на подумать

Просьба не сразу выдавать решение, а сначала написать некий алгоритм рассуждений, на основе которого можно самому попытаться построить цифровую схему.

При решении вариантов можно не делать задания, которые уже присутствовали в вопросах до этого, если только конечно вы не знаете какой-то изощрённый и отличающийся от представленного в методе ранее способа решения.

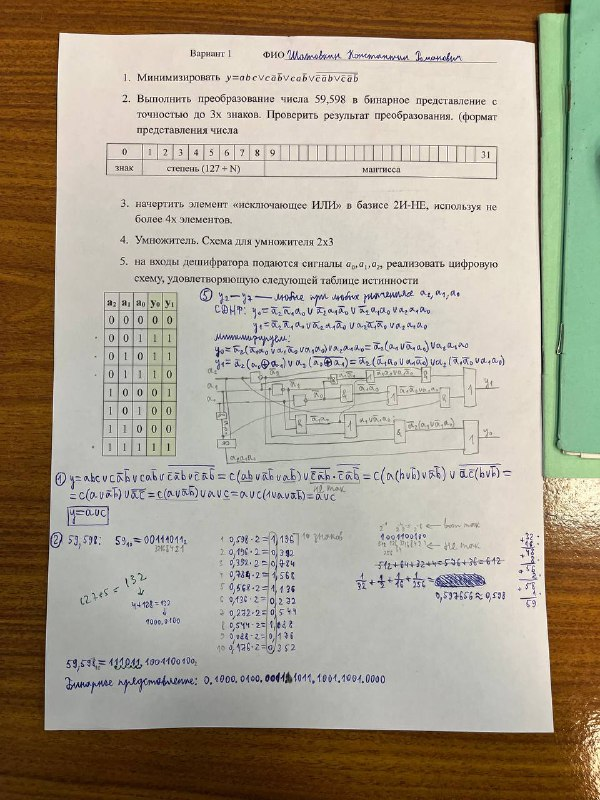
### ИСКЛ ИЛИ на 4 элементах (базис 2И-НЕ). ИСКЛ ИЛИ в базисе 2ИЛИ-НЕ

### Вычитатель на сумматоре

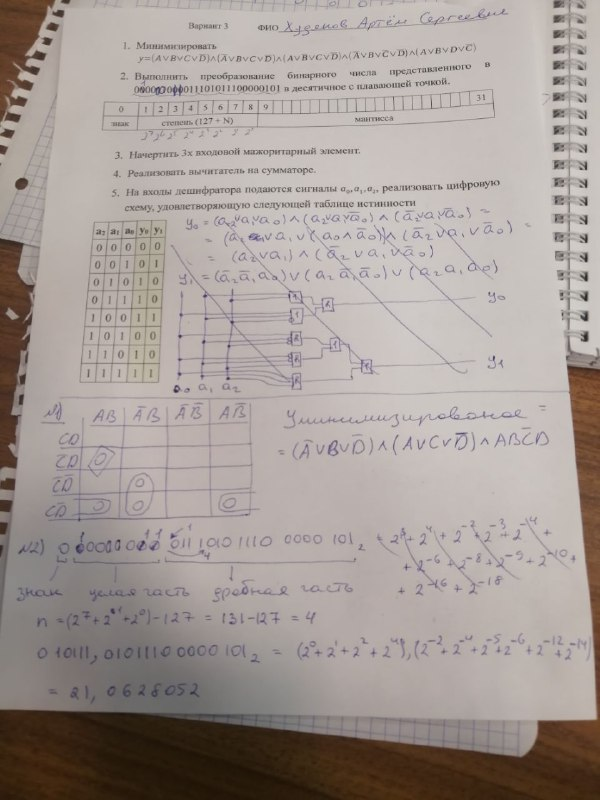
### Сложите 4-5 одноразрядных чисел. Есть восьмиразрядное число, определить количество единиц в нём

### Компаратор на сумматорах.

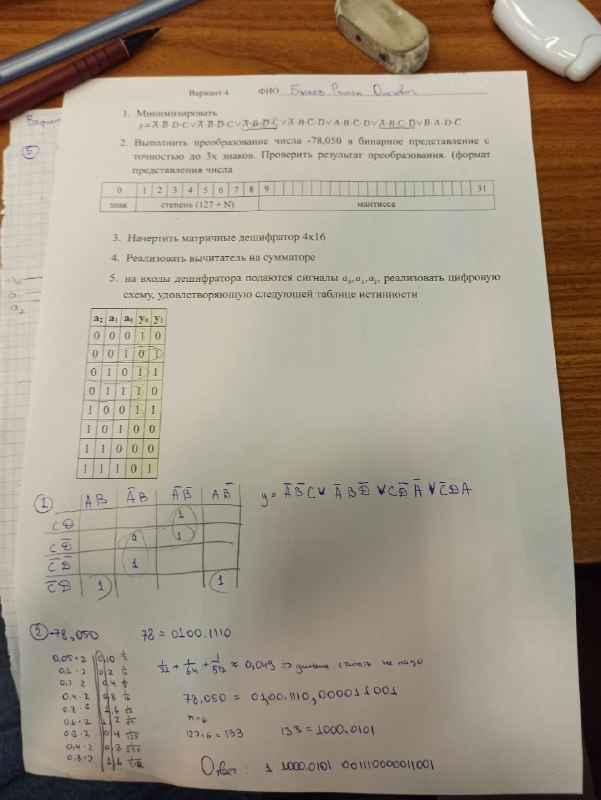
### Вариант 1 РК1



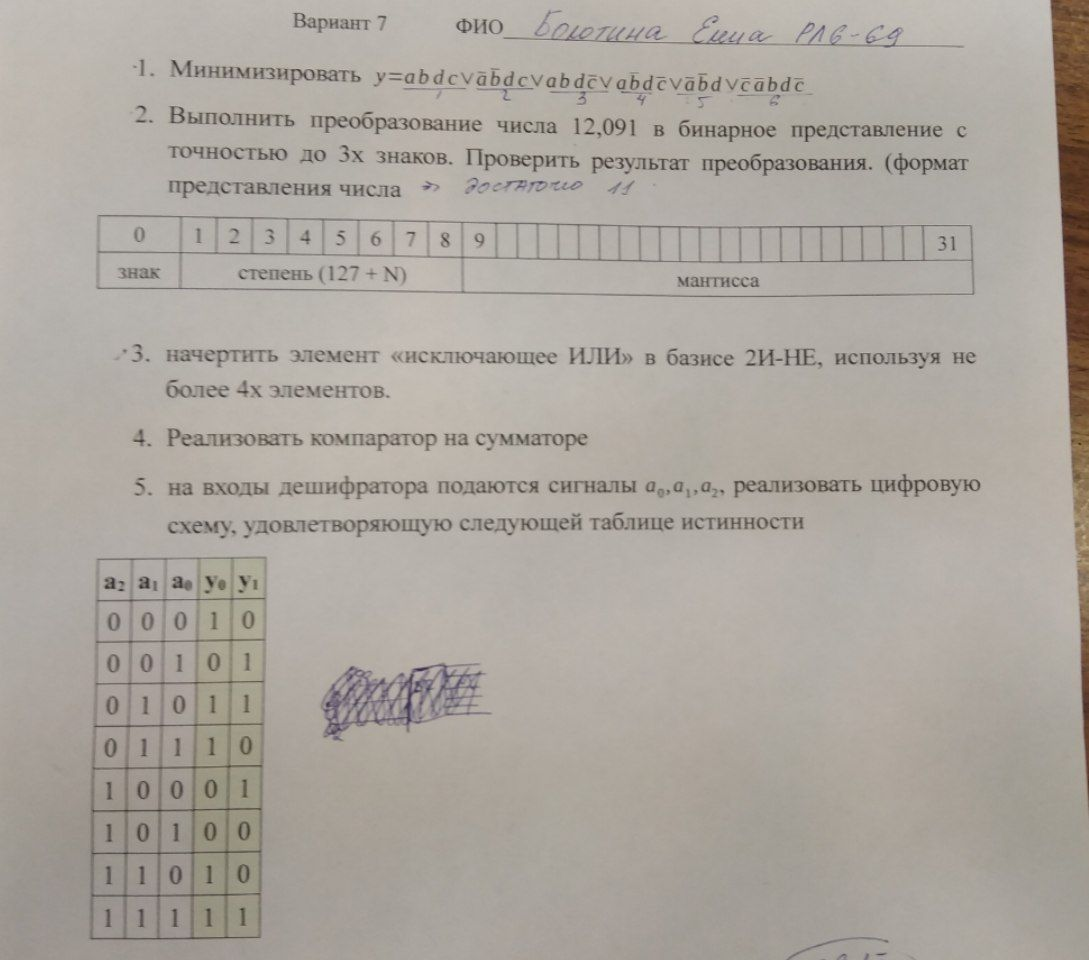
### Вариант 3 РК1



### Вариант 4 РК1



### Вариант 7 РК1



# Последовательстная логика (логика с памятью)

В чём отличие от комбинационной логики? Шо це таке?

## RS-триггер, статический, асинхронный, одноступенчатый

Пояснить статический, асинхр. О, ТИ, ВД. Построить на 2И-НЕ и 2ИЛИ-НЕ. В чём отличие? Как избежать запрещённого состояния?

## RS-триггер, синхронный, статический, одноступенчатый

Пояснить статический, синхр. О, ТИ, ВД. Построить на 2И-НЕ и 2ИЛИ-НЕ. В чём отличие? Как избежать запрещённого состояния?

## D-триггер (D flip-flop), синхронный, статический, одноступенчатый

О, ТИ, ВД. Всегда синхронный! Статический? Реализция D-триггера методом «чёрного ящика». На 2И-НЕ и 2ИЛИ-НЕ.

## Динамический D-триггер, синхронный, двухступенчатый

Переключение по заднему фронту, по переднему? О, ВД.

## T-триггер (счётный)

О, ВД.

## Асинхронный счётчик на D-триггерах.

Что значит Асинхронный? О, ВД. Как реализовать сброс счётчика?

## Динамический RS-триггер

ВД, О, схема. По переднему фронту, по заднему?

## Вычитающий счётчик (асинхронный)

Асинхронный? О, ВД

## Реверсивный счётчик

Принцип, ВД.

## Счётчик с предустановленными значениями

Принцип, ВД.

## Кольцевой счётчик (синхронный)

Принцип, ВД

## Кольцевой счётчик Джонсона

Принцип, ВД, как можно использовать для создания дешифратора?

## Кольцевой счётчик (ходит только одна единица)

Принцип, ВД, самовосстановление счётчика

## Синхронный счётчик вариант 1

Зачем нужны? Принцип работы, ВД. Реальные ВД для синхронного и асинхронного счетчиков, чтобы показать преимущества синхронного.

## Синхронный счётчик вариант 2

## Регистры. Последовательно-параллельные

## Регистры. Параллельно-последовательные

## Регистры параллельно-параллельные

## Регистры. Циклический регистр. Выбор направления. Предустановка значений.

## Динамический одноступенчатый D-триггер

Страшная схема

## Динамический D-триггер с асинхронным сбросом, установкой

Страшная схема со сбросом и с установкой

## JK-триггер, синхронный, статический

ВД, О

## Динамический JK-триггер

## Динамический JK-триггер на D-треггерах

## T-триггер на JK-триггере

## Синхронный счётчик на JK-триггере

## Сброс триггера. Асинхронные, синхронные входы.

## Задачи на подумать.

Просьба не сразу выдавать решение, а сначала написать некий алгоритм рассуждений, на основе которого можно самому попытаться построить цифровую схему.

### Счётчик в базисе 2И-НЕ

### Имеется кольцевой счётчик. Как загнать «1» или несколько «1» и двигать их по кругу?