**раздел «Арифметические основы»**

**1.1 Системы счисления**

Любая система счисления должна обеспечивать:

− возможность представления любого числа в рассматриваемом диапазоне величин;

− единственность этого представления;

− простоту выполнения операций над числами.

Непозиционная система счисления – система, для которой значение символа не зависит от его положения в числе. (римская сс)

Смешанная система - каждое число в ней представляется как линейная комбинация. (время)

Позиционной системой счисления которой значение цифры зависит от ее положения в числе, т. е. веса. (десятичная 2 4 8..)

**1.2 Критерии выбора системы счисления**

Простота технической реализации, элемент будет тем проще, чем меньше состояний требуется для запоминания цифры числа.

Наибольшая помехоустойчивость кодирования цифр, наиболее вероятна ошибка в устройствах, для которых используется система счисления с наибольшим основанием.

Минимум оборудования, надо найти такую систему счисления, которая имеет минимальное количество цифроразрядов (D), где D=кол-во цифр в числе \* кол-во разрядов.

Простота арифметических действий, чем меньше цифр в системе счисления, тем проще арифметические действия над ними.

Наибольшее быстродействие, Максимальное время сложения получается в случае если выбрана система счисления с основанием 2.

Простота аппарата для выполнения анализа и синтеза цифровых устройств.

Удобство работы с ЭВМ.

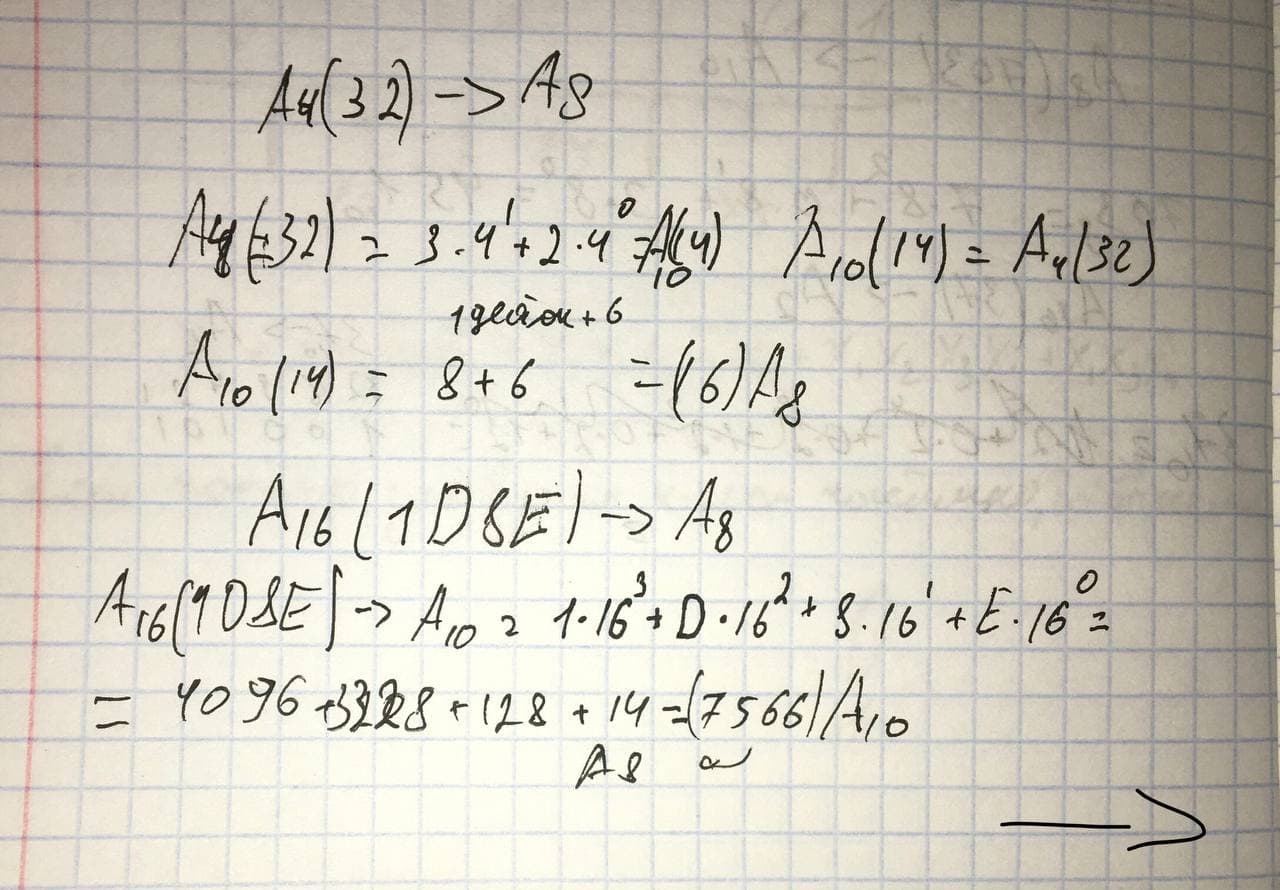
Возможность представления любого числа в рассматриваемом диапазоне величин.

Единственность представления числа.

**2. Перевод целых чисел из одной системы счисления в другую**

Метод подбора степеней основания:

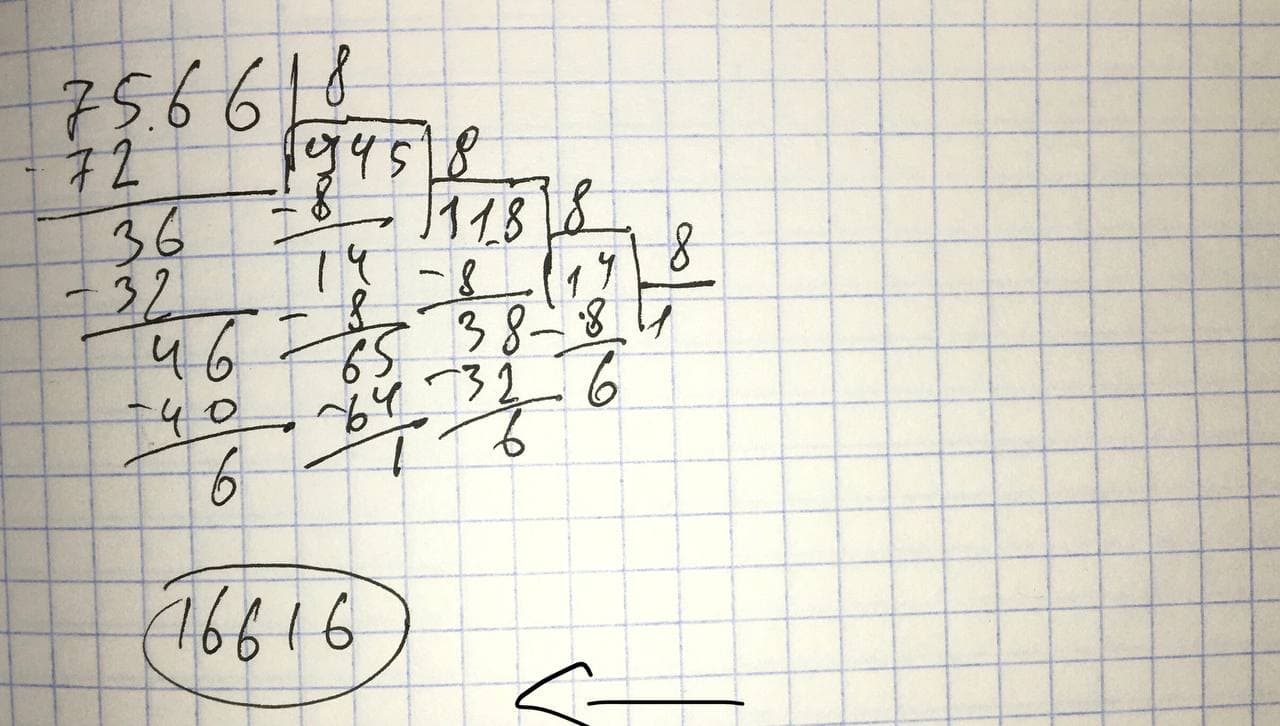
****



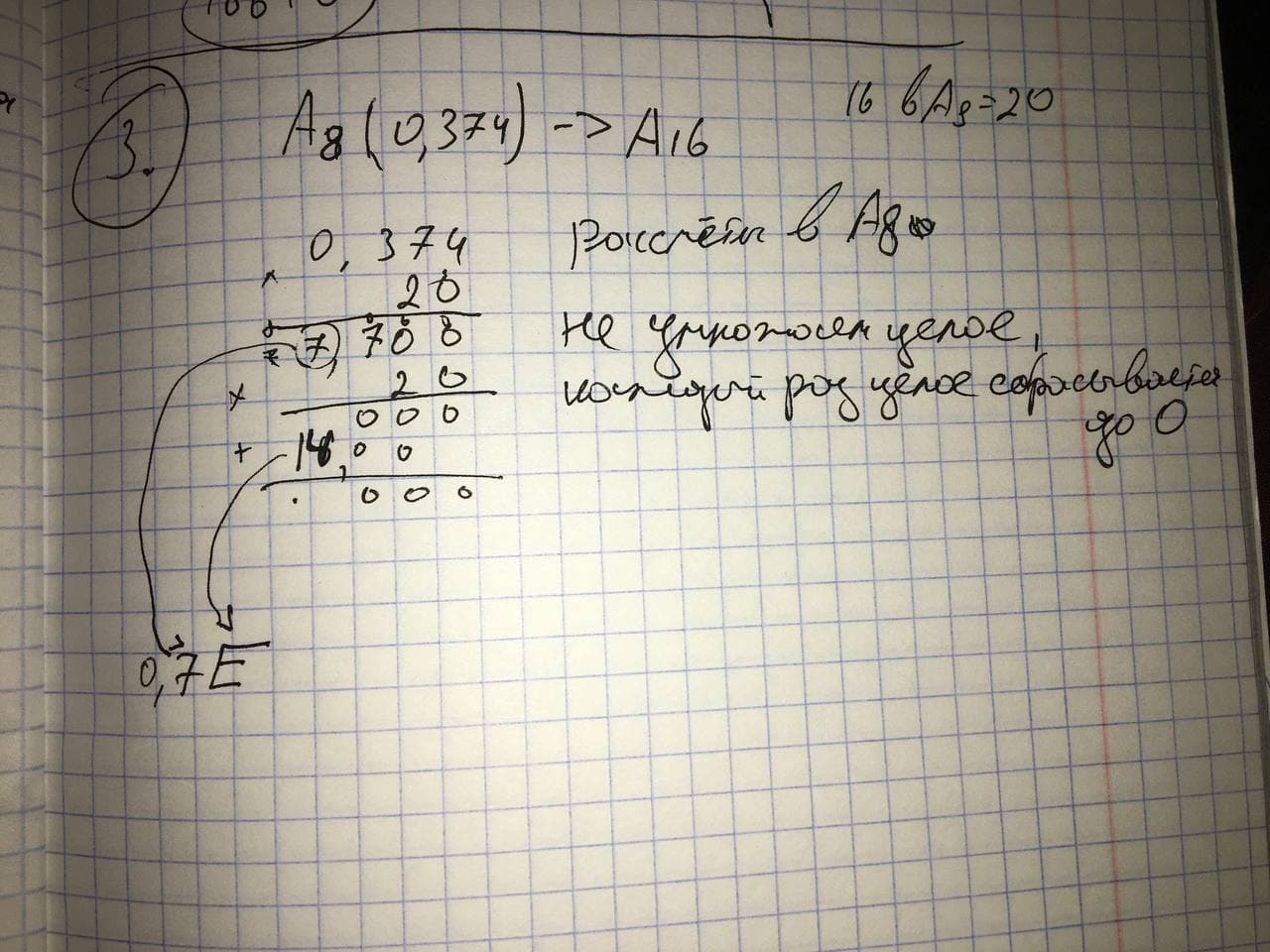
Метод деления на основание системы счисления:



И делим на r2 получая остатки an и целые части



**3. Перевод дробных чисел из одной системы счисления в другую**



**4. Кодирование чисел (прямой, обратный, дополнительный код)**

**5.1 Переполнение разрядной сетки. Причины и признаки переполнения**

Причиной переполнения может служить суммирование двух чисел с одинаковыми знаками, которые в сумме дают величину, большую или равную единице.

**5.2 Модифицированные коды**

Знак числа кодируется двумя разрядами

**6. и 7. Формы представления чисел в ЭВМ  (числа с фиксированной и плавающей запятой)**



**8. Округление**

Aо– часть числа, не вошедшая в разрядную сетку.

1) Отбросить Ао

2) Симметричное - +1 к последнему разряду мантиссы

3) Округление по дополнению - переносится 1, если она есть из разряда за сеткой

4) Случайное - +1 или +0 к последнему разряду мантиссы

**9. Погрешность выполнения арифметических операций**

А = [A] + ∆A и B = [B] + ∆B, здесь А и В – точное значение числа, [А] и [В] – приближенное их значение, ∆A и ∆B – абсолютная погрешность

Абсолютные погрешности действий:

A+B=∆A + ∆B

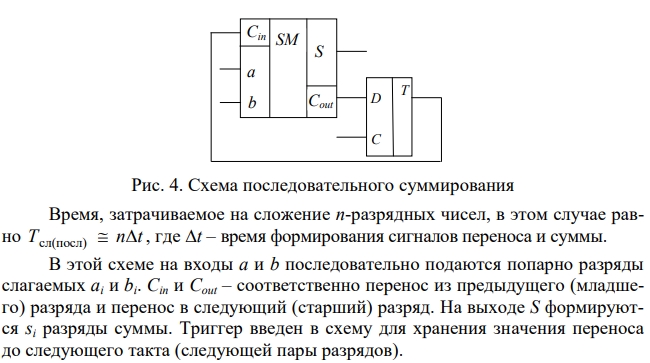
A-B=∆A − ∆B

A\*B=[A]∆B + [B]∆A

A/B= ∆A/[B] − [A]∆B/([B])^2

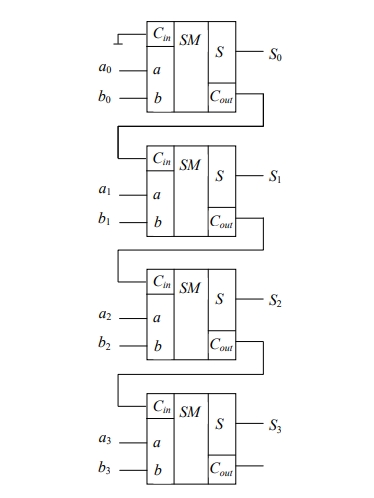
**10. Последовательное сложение чисел. Последовательный сумматор**

Параллельный способ передачи информации является более быстродействующим по сравнению с последовательным, но менее экономичным.



**11. Параллельное сложение чисел. Параллельный сумматор**

Быстродействие устройства определяется суммой задержек передачи сигнала переноса с младшего разряда на вход сумматора старшего разряда.



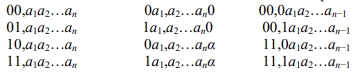
**12. Нормализация чисел**

Число называется нормализованным, если его мантисса меньше еденицы и больше нуля.

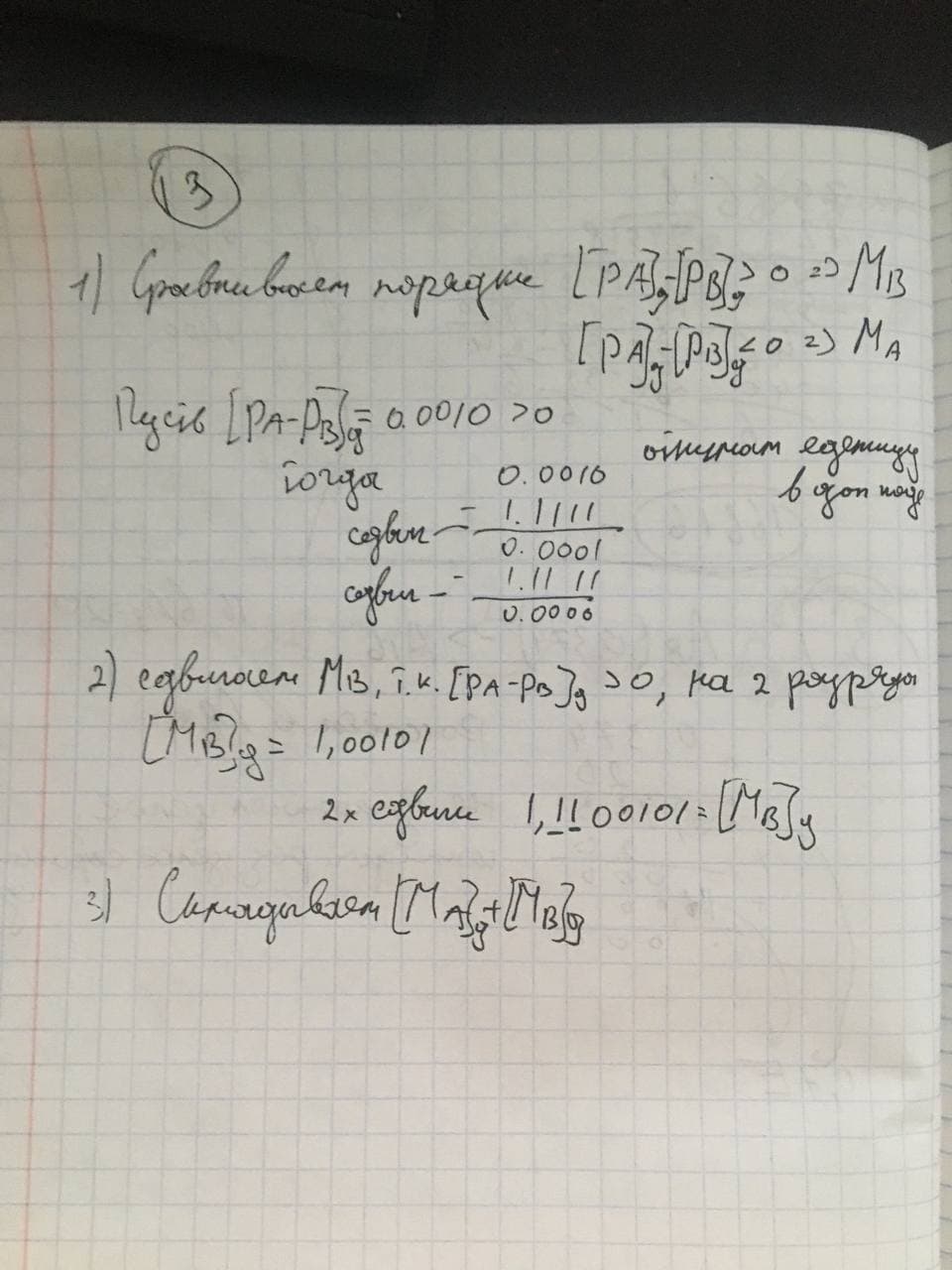
Простой сдвиг:



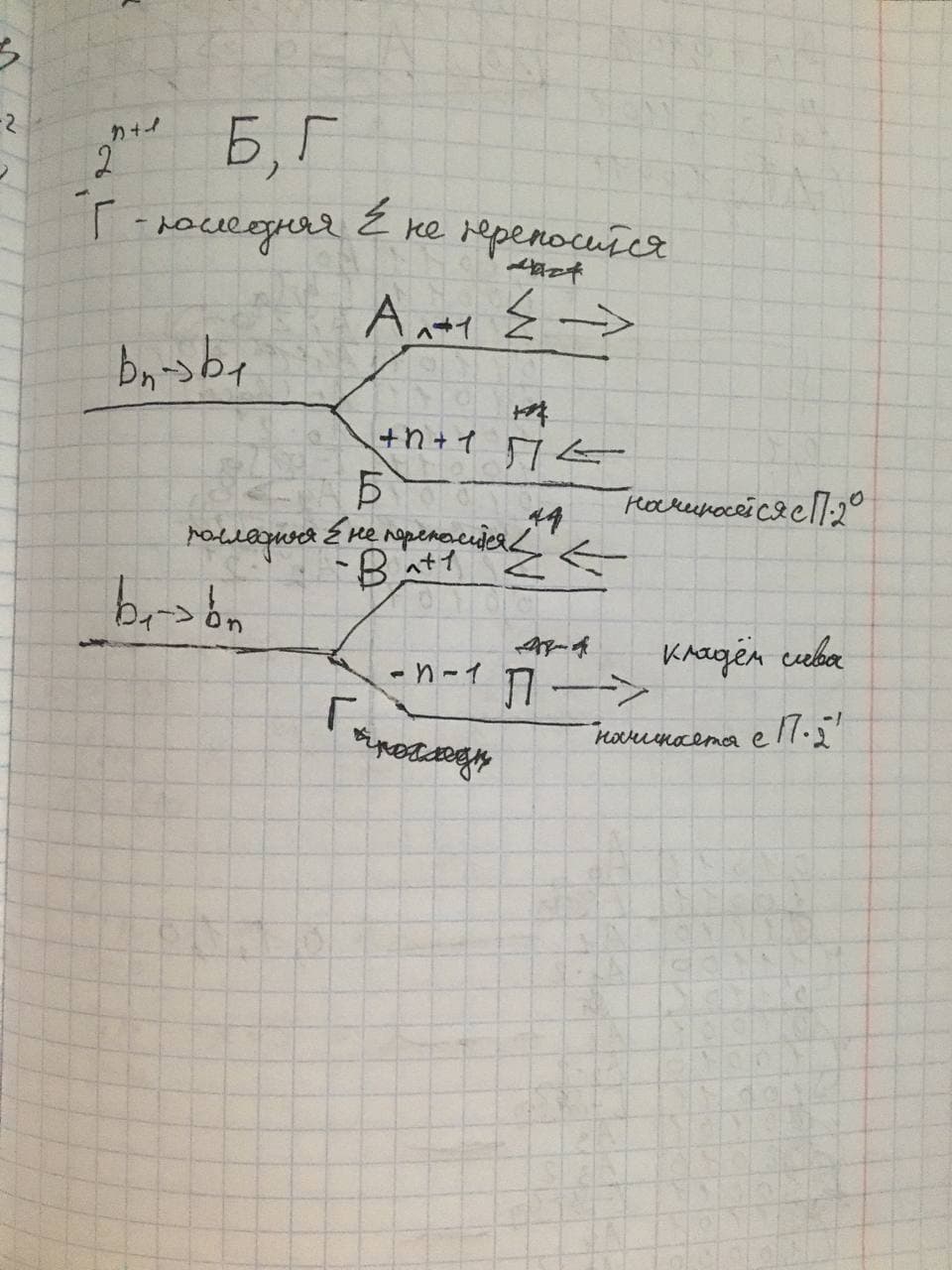
Модифицированный сдвиг − сдвиг, при котором в сдвигаемый разряд заносится значение, совпадающее со значением знакового разряда:



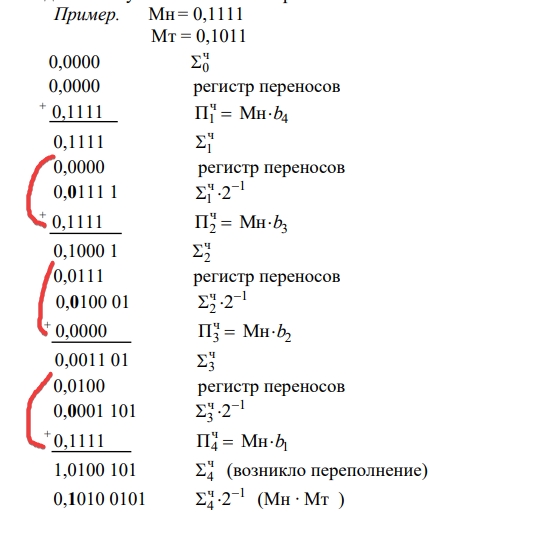
**13. Сложение чисел с плавающей запятой**



**14. Умножение чисел в прямых кодах**



**15. Умножение чисел с хранением переносов**



**16. Умножение на 2 разряда множителя одновременно в прямых кодах**

Разбиваем Мт на пары и преобразуем, если перенос, то сначала перенос, потом преобразование. Преобразуемые пары: 11

0**.0**101001 //добавляем до пары нуль

0**,**010110**0** //добавляем до пары нуль

**17. Умножение на 4 разряда множителя одновременно в прямых кодах**

Дает возможность осуществить сдвиг на четыре разряда за один такт, но понадобятся дополнительные регистры для хранения множителей.

**18. Умножение дробных чисел в дополнительных кодах**

Мн > 0, Mн < 0,

Mт < 0, Мт < 0, то поправка [-Мн]доп

**19. Умножение целых чисел в дополнительных кодах**

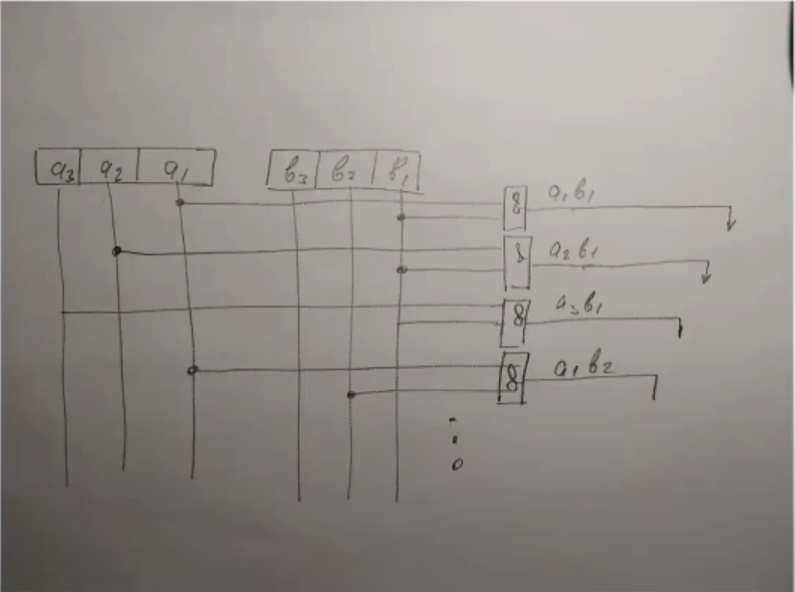
Мн > 0, Mн < 0,

Mт < 0, Мт < 0, то поправка [-Мн]доп

**20. Умножение на 2 разряда множителя одновременно в дополнительных кодах**

Разбиваем Мт на пары и преобразуем, если перенос, то сначала перенос, потом преобразование. Преобр. пары: 10 и 11

**21. Матричные методы умножения**

Произведение A\*B может быть получено, если суммировать элементы матрицы (по диагонали). Принцип основан на использовании матриц промежуточных результатов.

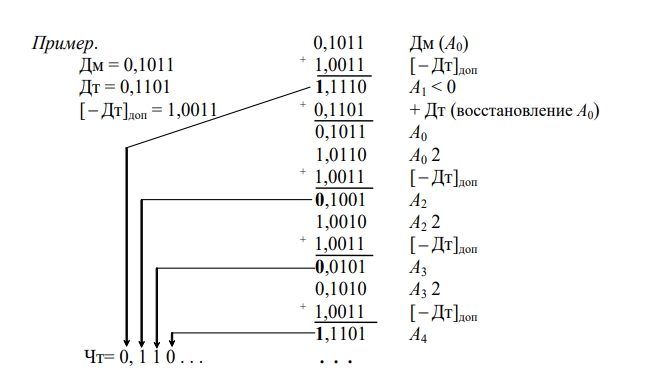
**22.1 Машинные методы деления**

Деление – простое многократное вычитание делителя вначале из делимого, затем из остатков. С восстановлением или без.

**22.2 Деление в прямых кодах**

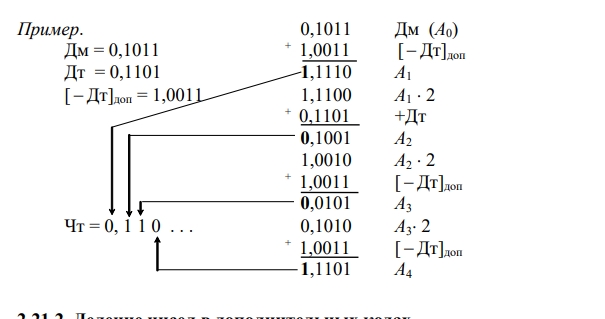
С восстановлением:

Дм + [-Дт]доп если А<0, то +Дт(восст) и ставим %0%, восст сдвигаем \* 2. Если А>0, то сдвигаем \* 2 и -Дт и ставим %1%



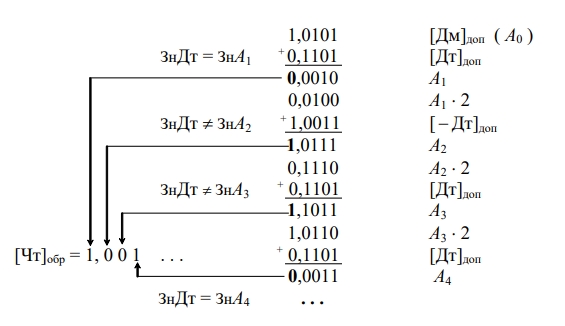
Без восстановления:

Дм + [-Дт]доп, А>0 ставим %1%, A<0 ставим %0% и сдвигаем \* 2. Если число положительное, то -Дт, если отрицательное, то +Дт.

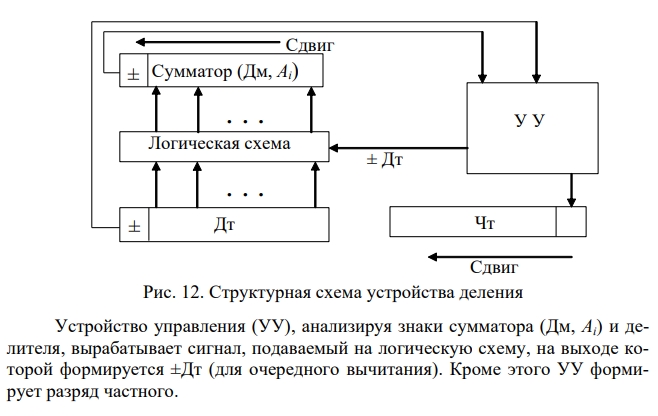


**23. Деление в дополнительных кодах.**

Если знак Дм = зн Дт, то -Дт. Если зн Дт = зн А, то 1 в ответ и дальше -Дт.



**24. Структурная схема операционного устройства выполняющего** **операцию деления**



**25. Методы ускорения деления**

Дт выбирается таким образом, чтобы он был нормализован. Это требует устройство управления делением и логическую схему, осуществляющую две функции:

1) сдвиг Дт и делимого до тех пор, пока после запятой не станет единица;

2) выявление остатков вида 0,0 или 1,1.

**26. Одноразрядный двоично-десятичныей сумматор**

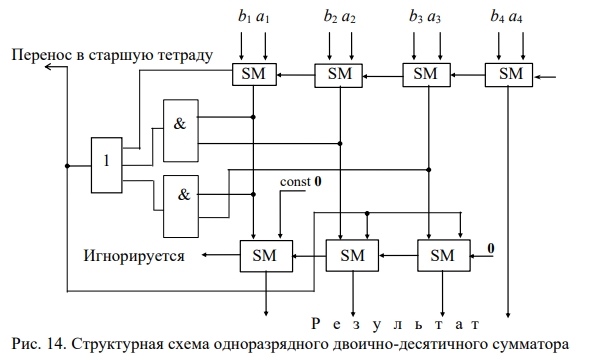
При обработке больших массивов десятичных чисел значительная часть времени расходуется на перевод чисел из одной системы счисления в другую. В этом случае целесообразно выполнять обработку данных непосредственно в десятичной системе счисления. При этом для представления десятичных чисел используют различные двоично-десятичные коды. Десятичные цифры представляются двоичными тетрадами. Сложение тетрад выполняется с помощью двоично-десятичных сумматоров.

**27. BCD-коды. Суммирование чисел с одинаковыми знаками**

1) Если двоичная сумма не более 9, то коррекция не требуется.

2) Если двоичная сумма принимает значение от 10 до 15, необходимо искусственно вызвать перенос в следующую тетраду. + 0110.

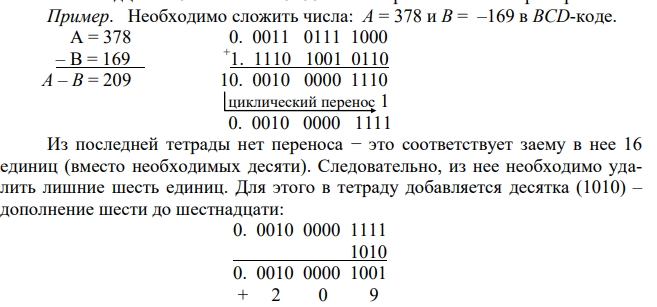
3) Если двоичная сумма принимает значение от 16 до 19, то возникает перенос из тетрады, который имеет вес, равный 16, и уменьшает значение суммы на 6. + 0110.



Если перенос из этой тетрады или запрещенная комбинация, то требуется коррекция. + 0110

**28. BCD-коды. Сложение чисел с разными знаками (результат > 0).**

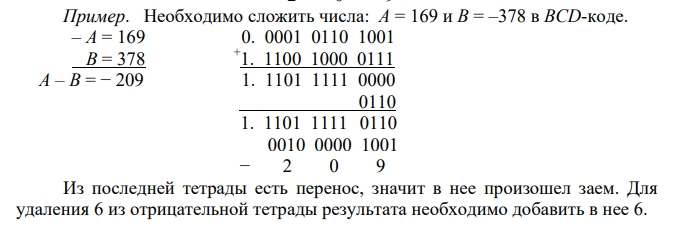
Если есть запрещенная комбинация, то требуется коррекция. + 1010



**29. BCD-коды. Сложение чисел с разными знаками (результат < 0)**

Если из тетрады есть перенос, то +0110

Если ответ отрицательный, то инверсируем все кроме знака.



**30.** **BCD-коды с избытком 3**

Если число <=9 то + 0011

Если число >=10 то -0011

Без инверсий

**раздел «Логические основы»**

**31. Основные понятия алгебры логики**

Алгебра логики – это математический аппарат, с помощью которого записывают (кодируют), упрощают, вычисляют и преобразовывают логические высказывания.

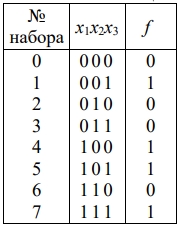
Двоичные (логические, булевы) переменные могут принимать только 0 и 1 − и обозначаются символами x1, x2. А также являются аргументами логических функций.

Логическая (булева) функция – это функция и аргументы и значение которой принадлежат множеству {0, 1}.

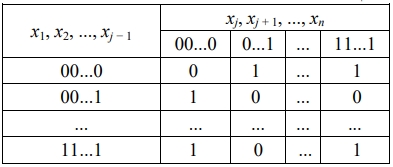
Для того чтобы задать функцию, достаточно выписать значения f(0, 0, ..., 0, 0), f(0, 0, ..., 0, 1), f(1, 1, ..., 0, 1), f(1, 1, ..., 1, 0). Этот набор называют вектором значений функции.

**32. Способы задания функций алгебры логики**

1) Табличный (матричный) для задания произвольной булевой функции:



2) Модификация таблицы истинности. Построение таблицы истинности для функции n переменных.



3) При аналитическом способе булева функция задается формулой, т. е. аналитическим выражением, построенным из операций булевой алгебры.

**33. Формы представления функций алгебры логики**

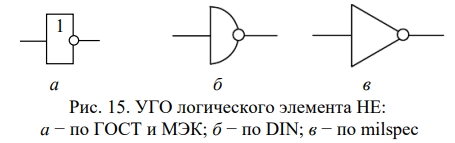
Существует несколько общепринятых стандартов условных обозначений логических элементов. Среди них наиболее распространенными являются американский стандарт milspec806B и стандарт МЭК 117-15 А, созданный Международной Электротехнической Комиссией. Часто в литературе используются обозначения элементов согласно европейского стандарта DIN 4070. В отечественной литературе УГО элементов, в основном, приводятся в ГОСТ 2.743−82.

Функций от одной переменной четыре: f0 = 0 , f1= 1, f2 = значение переменной, f3 = противоположное значение.

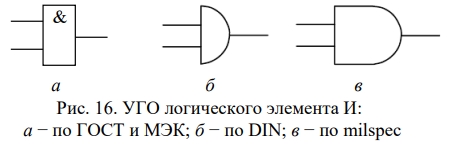
Фиктивные переменные - это те, значения которых не меняются при изменении добавочных переменных.

Основные операции булевой алгебры:

1) Логическое отрицание



2) Логическое умножение



3) Логическое сложение



Импликация − это высказывание, принимающее ложное значение только в случае, если x1 истинно, а x2 ложно

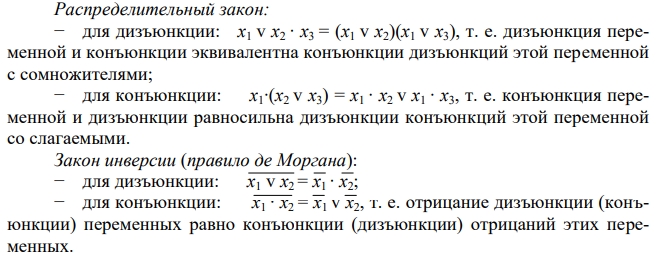
Суперпозиция функций одного аргумента порождает функции одного аргумента. Суперпозиция функций двух аргументов дает возможность строить функции любого числа аргументов. Суперпозиция булевых функций представляется в виде логических формул.

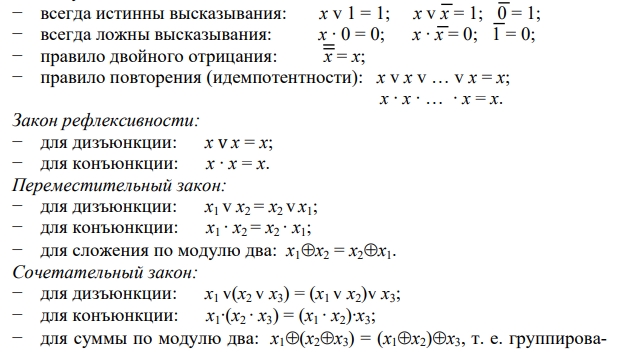
Однако:

- функция может быть представлена разными формулами

- каждой формуле своя суперпозиция и схема

**34. Основные законы и правила алгебры логики**





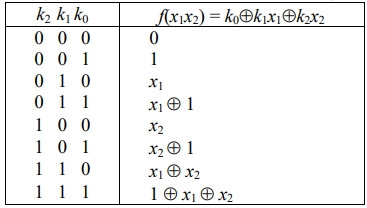
**35.1 Классы функций алгебры логики**

Полином Жегалкина

****

1) Класс линейных функций:





2) Класс функций, сохраняющих нуль

Функция сохраняет константу 0

f(0, ..., 0) = 0

3) Класс функций, сохраняющих единицу

Функция сохраняет константу 1

f(1, ..., 1) = 1

4) Класс монотонных функций

Функция называется монотонной, если при любом возрастании набора переменных значения этой функции не убывают.

Если f(0,0) ≥ f(0,1) ≥ f(1,1) или f(0,0) ≥ f(1,0) ≥ f(1,1), то функция f является монотонной.

5) Класс самодвойственных функций

булева функция называется самодвойственной, если на любых двух противоположных наборах она принимает противоположные значения.



**35.2 Функционально полные наборы**

Для того чтобы система булевых функций была полной, необходимо и достаточно, чтобы она содержала хотя бы одну функцию:

− не являющуюся линейной;

− не сохраняющую нуль;

− не сохраняющую единицу;

− не являющуюся монотонной;

− не являющуюся самодвойственной.

Минимальный базис – такой базис, когда удаление одной (любой) функции превращает систему булевых функций в неполную.

Функционально полный базис – это набор операций булевой алгебры, позволяющих построить любую булеву функцию.

Функционально полным базисом является базис И, ИЛИ, НЕ.

**36. Минимизации булевых функций методом Квайна**

1) Для получения минимальной формы булевой функции необходимо в СДНФ произвести все возможные склеивания и поглощения так, чтобы в результате была получена сокращенная ДНФ. Склеивание и поглощение выполняются до тех пор, пока имеются члены, не участвовавшие в попарном сравнении. Если какие то термы не учавствовали в склеиваниях и поглощениях, то это простые импликанты, которые включаются в сокращенную ДНФ.

2) Формирование тупиковой формы. Для выявления обязательных простых импликант и формирования на их основе минимального покрытия строится импликантная таблица. В строках импликантной таблицы записываются простые импликанты, а в столбцах − конституенты единицы. Звездочка ставится, если простая импликанта покрывает конституенту.

**37. Карты Вейча**

Один из наиболее удобных методов при небольшом числе переменных (до шести).

Переменные, обозначающие клетки диаграммы, расставляются таким образом, чтобы наборы, записанные в двух смежных клетках, имели кодовое расстояние, равное единице. Кодовым расстоянием между двумя наборами называется количество отличающихся координат (разрядов).

В клетку карты, соответствующую конституенте (элементарной конъюнкции) единицы, заносится 1, иначе − 0.

Минимизация заключается в объединении единичных клеток в контуры.

1) Внутри контура должны находиться только одноименные значения функции.

2) Количество клеток внутри контура должно быть равно степени двойки.

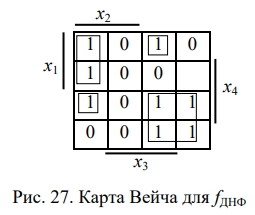
3) При проведении контуров крайние строки карты, а также угловые клетки считаются соседними.

4) Каждый контур должен включать максимально возможное количество клеток.

5) Все единицы в карте должны быть охвачены контурами. Любая единица может входить в контуры произвольное количество раз.

6) Число контуров должно быть минимальным.

7) Переменные булевой функции входят в элементарную коньюнкцию без инверсии, если их значение координатах равно единице и с инверсией, если равно нулю.



Можем объединять в контуры, если единицы симметричны!

**38. Карты Карно**

Карта Карно является координатным способом представления булевых функций. Переменные функции разбиваются на две группы так, что одна группа определяет координаты столбца карты, а другая − координаты строки.

Переменные в строках и столбцах располагаются так, чтобы соседние клетки карты Карно различались только в одном разряде переменных. Карта Карно для пяти переменных представляет собой две карты четырех переменных, зеркально отображенные относительно центральной вертикальной линии

**39. Минимизация не полностью определенных ПФ в дизъюнктивной и конъюнктивной формах**

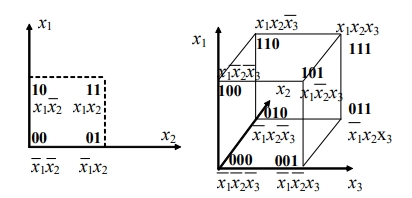
Звездочками на карте отмечаются наборы, на которых функция f не определена.

Реализация полученной f логической схемой с учетом неопределенных наборов (\*)будет «дешевле» по сравнению с реализацией функции представленной в форме без учета неопределенных наборов(\*).

**40. Кубическое задание ФАЛ**

Булева функция удобно интерпретируется с использованием ее геометрического представления. Функцию двух переменных можно интерпретировать как плоскость, заданную в системе координат х1х2.

Кубическое представление булевых функций наиболее пригодно для машинных методов их анализа, так как позволяет компактно представлять булевы функции от большого количества переменных. Для представления булевых функций в кубической форме используется кубический алфавит {0, 1, x}. Для функций, зависящих более чем от четырех переменных, предпочтительным является аналитическое представление булевых функций.



**41. Минимизации булевых функций методом Квайна-Мак Класки**

1) Сформируем кубический комплекс K, состоящий из кубов нулевой размерности. Выполним разбиение комплекса K на группы.

Попарное сравнение можно проводить только между соседними по номеру группами кубов. Если сравниваемые кубы различаются только в одном разряде, то в следующий столбец записывается код с символом x на месте этого разряда.

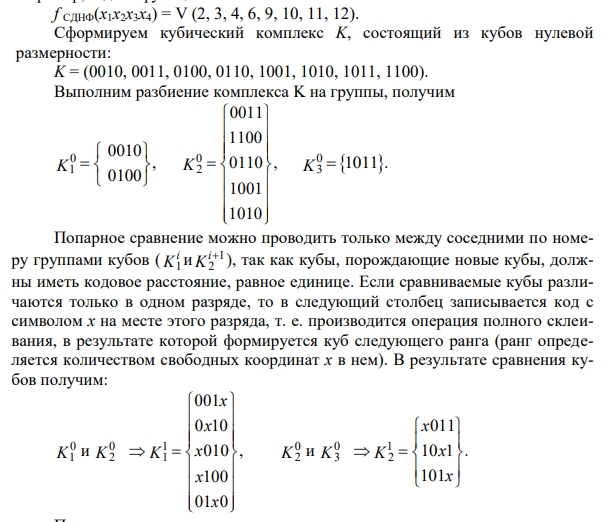
2) Полученные кубы первой размерности разобьем на группы кубов в зависимости от местоположения свободной координаты (х) в кубе.

3) Далее выполняется сравнение кубов внутри каждой из групп. В результате могут быть получены кубы второй размерности, которые аналогично кубам первой размерности будут объединены в группы по совпадению свободных координат и вновь будет выполнено сравнение.

Кубы, не участовавшие в образовании новых - исключаются.

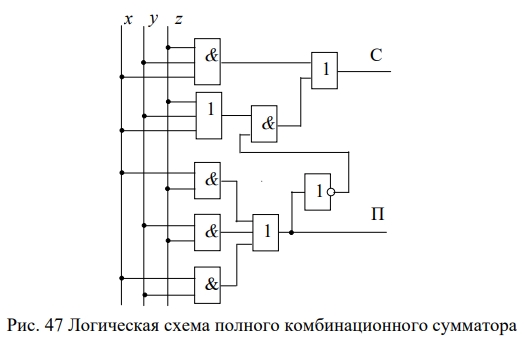
4) Таким образом, получено множество простых импликант.

5) Далее аналогично методу Квайна строится импликантная таблица



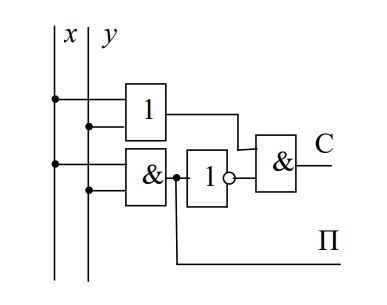
**42. Синтез одноразрядного полного комбинационного сумматора**

Полный одноразрядный двоичный сумматор (рис. 46) имеет три входа (x, y − для разрядов двух слагаемых и z − для переноса из предыдущего (младшего) разряда) и два выхода (C − сумма, П − перенос в следующий (старший) разряд). Обозначением полного двоичного сумматора служат буквы SM.

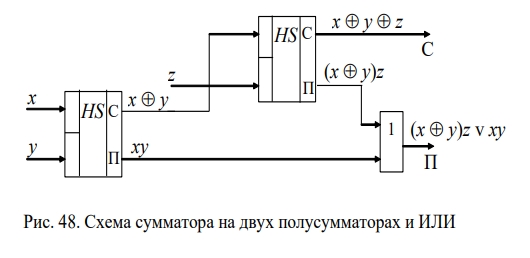


**43. Синтез одноразрядного комбинационного полусумматора**

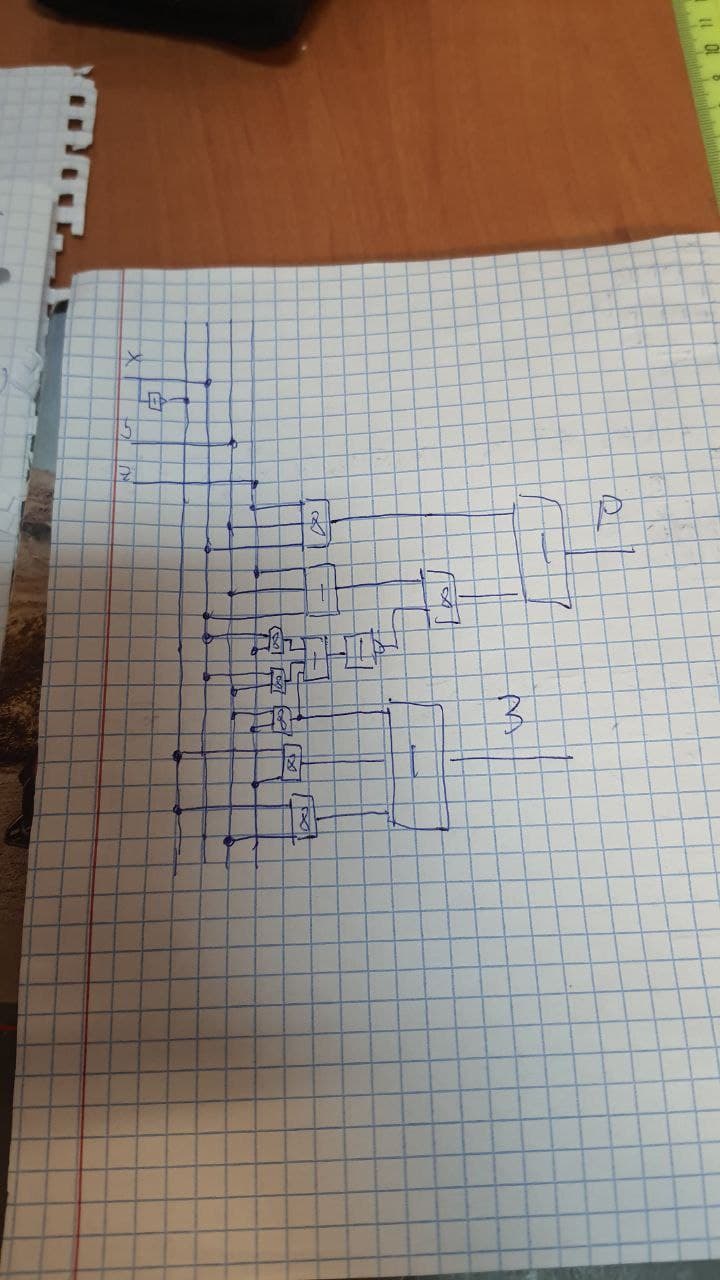
Полусумматор имеет два входа x и y для разрядов двух слагаемых и два выхода: C – сумма, П – перенос. Таким образом, это устройство предназначено для сложения разрядов двух чисел без учета переноса из предыдущего разряда.



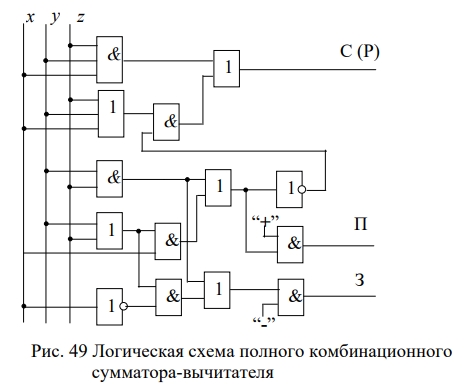
**44. Синтез одноразрядного полного комбинационного сумматора на 2 полусумматорах**



**45. Синтез одноразрядного комбинационного вычитателя**



**46. Объединенная схема одноразрядного комбинационного сумматора-вычитателя**



**47. Триггер со счетным входом как полный одноразрядный сумматор**

Триггером называется устройство, имеющее два устойчивых состояния и способное под действием входного сигнала переходить из одного устойчивого состояния в другое. Триггер – это простейший цифровой автомат с памятью, способный хранить один бит информации.

Триггер со счетным входом (Т-триггер) может быть рассмотрен как полный сумматор, работающий в три такта. В основе работы этого устройства лежит операция «сумма по модулю 2».

1) 1 такт: слагаемое х подается на вход триггера;

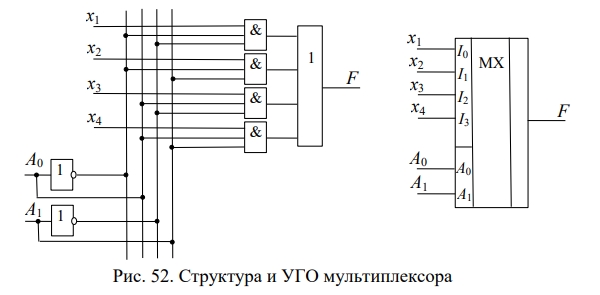
2) 2 такт: на вход триггера подается второе слагаемое y и формируется сумма по модулю 2 слагаемых x и y;

3) 3 такт: в третьем такте на вход триггера поступает значение, соответствующее третьему слагаемому z

4) на выходе T-триггера получена полная сумма трех слагаемых

**48. Стандартные функциональные узлы цифровой техники (мультиплексоры)**

Мультиплексором называется комбинационное цифровое устройство, способное передавать информацию с нескольких входов на один выход. Мультиплексор имеет несколько информационных входов, один или более управляющих (адресных) входов и один выход. Выбор входа осуществляется подачей соответствующей комбинации (номера этого входа) на управляющие входы мультиплексора.



**49. Стандартные функциональные узлы цифровой техники (дешифраторы)**

Дешифратор – это комбинационная логическая схема, преобразующая поступающий на ее входы код сигнал только на одном из выходов.

При подаче на вход устройства двоичного кода на выходе дешифратора появится сигнал на том выходе, номер которого соответствует десятичному эквиваленту двоичного кода.

