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.

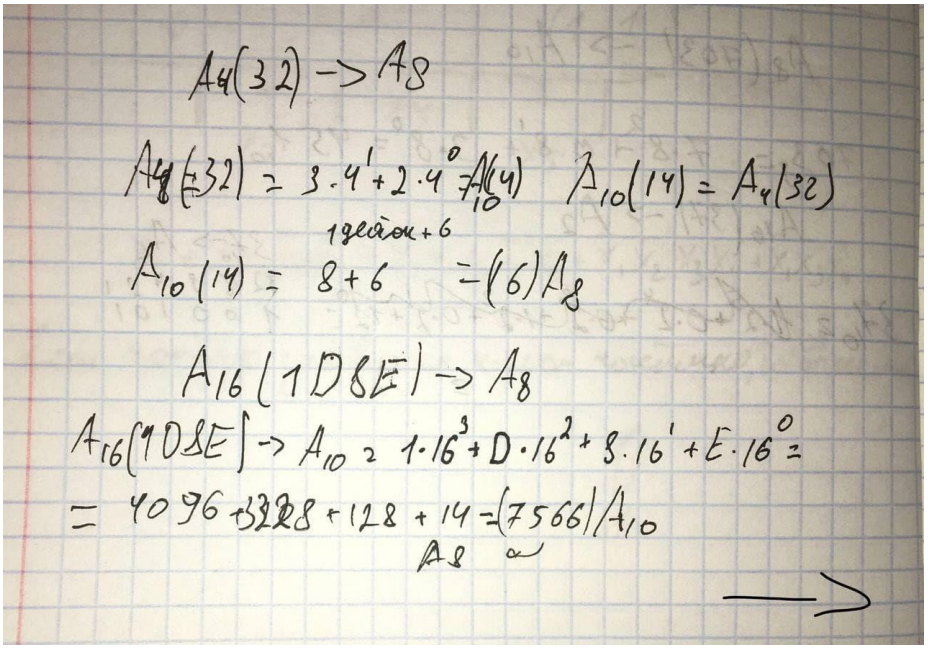
Возможность представления любого числа в рассматриваемом диапазоне величин.

Единственность представления числа.

2. Перевод целых чисел из одной системы счисления в другую

Метод подбора степеней основания:

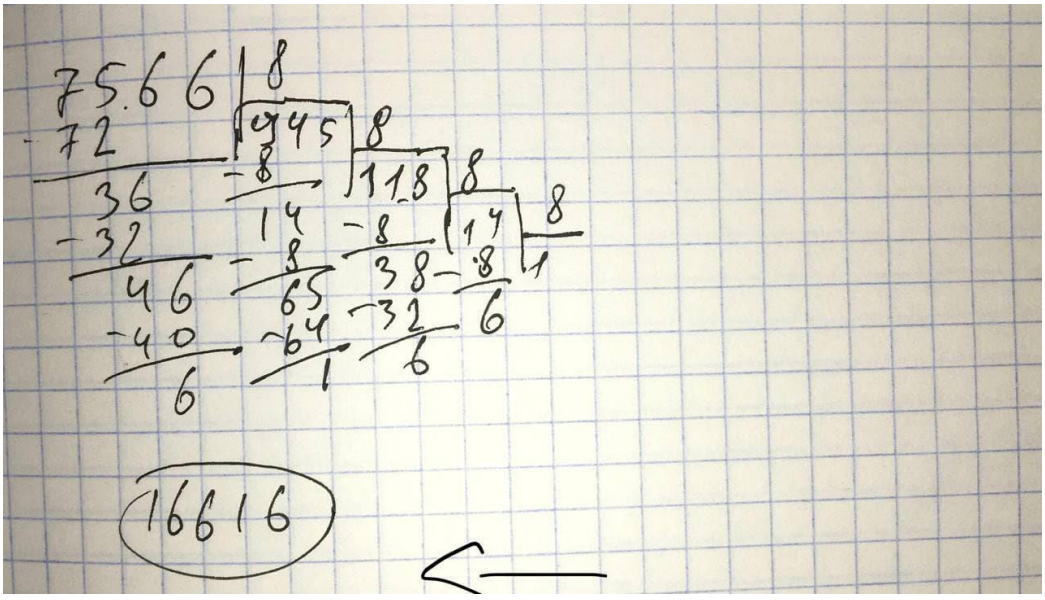
$$A_{r_2} = a_n r_2^n + a_{n-1} r_2^{n-1} + \ldots + a_1 r_2^1 + a_0 r_2^0$$



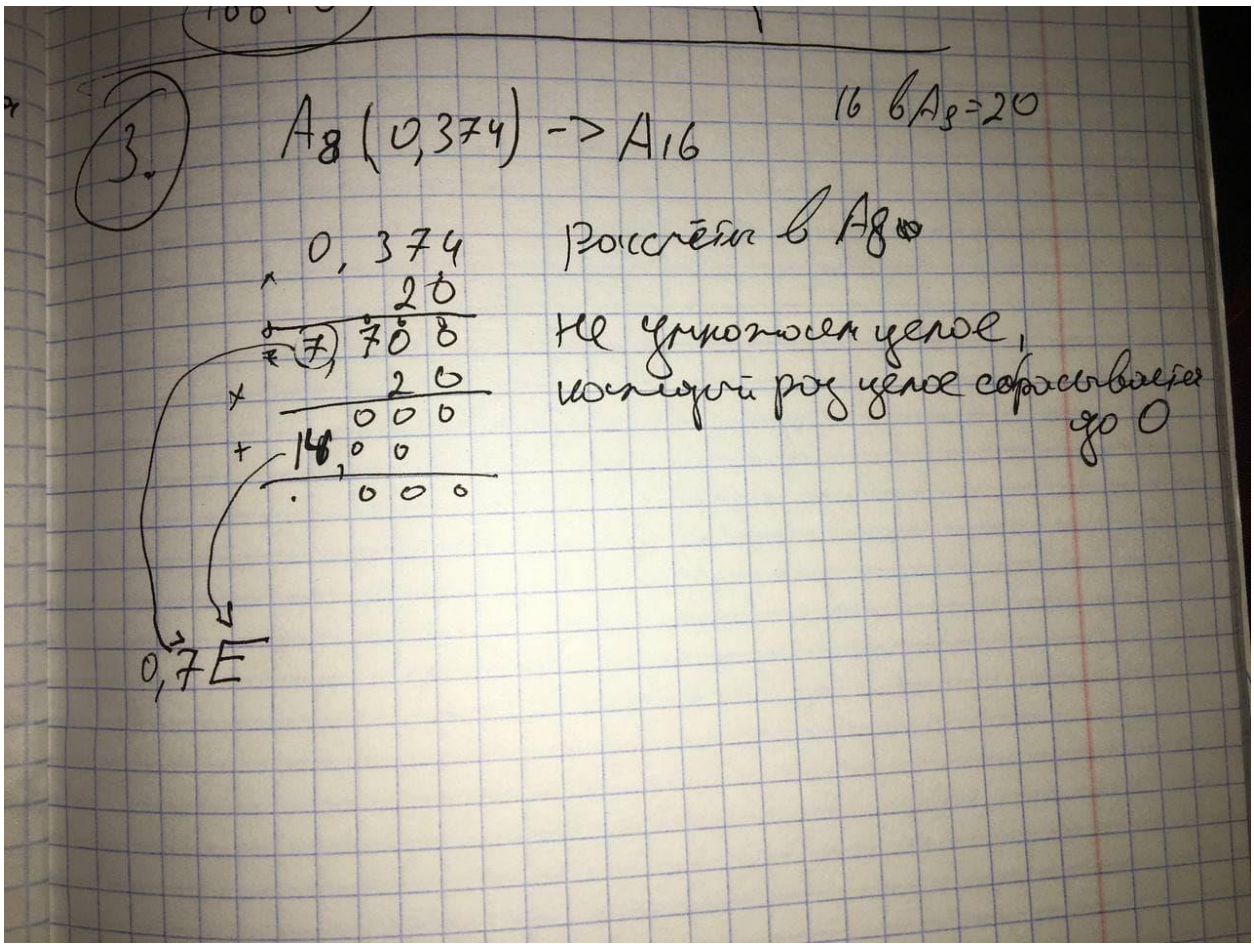
Метод деления на основание системы счисления:

$$A_{r_2} = (\ldots(a_n r_2 + a_{n-1}) r_2 + \ldots + a_1) r_2 + a_0$$

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

Ао– 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

A = [A] + ΔA и B = [B] + ΔB, здесь A и B – точное значение числа, [A] и [B] – приближенное их значение, ΔA и ΔB – абсолютная погрешность

Абсолютные погрешности действий:

A+B=ΔA + ΔB

A-B=ΔA − ΔB

A*B=[A]ΔB + [B]ΔA

A/B= ΔA/[B] – [A]ΔB/([B])^2

10. Последовательное сложение чисел. Последовательный сумматор

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

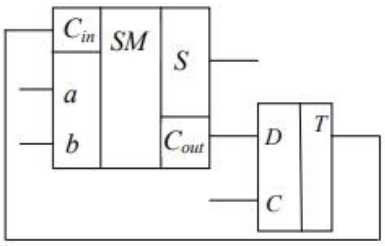


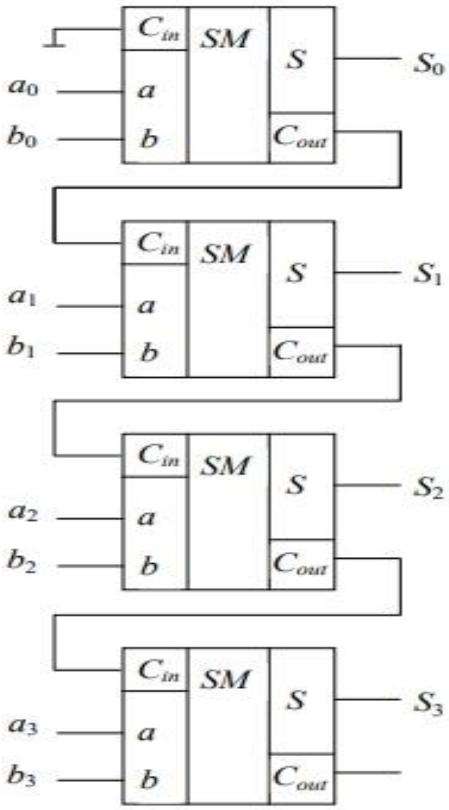
Рис. 4. Схема последовательного суммирования

Время, затрачиваемое на сложение n-разрядных чисел, в этом случае равно $T_{\text{сл(посл)}} \cong n\Delta t$, где Δt – время формирования сигналов переноса и суммы.

В этой схеме на входы a и b последовательно подаются попарно разряды слагаемых a_i и b_i. C_{in} и C_{out} – соответственно перенос из предыдущего (младшего) разряда и перенос в следующий (старший) разряд. На выходе S формируются s_i разряды суммы. Триггер введен в схему для хранения значения переноса до следующего такта (следующей пары разрядов).

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.

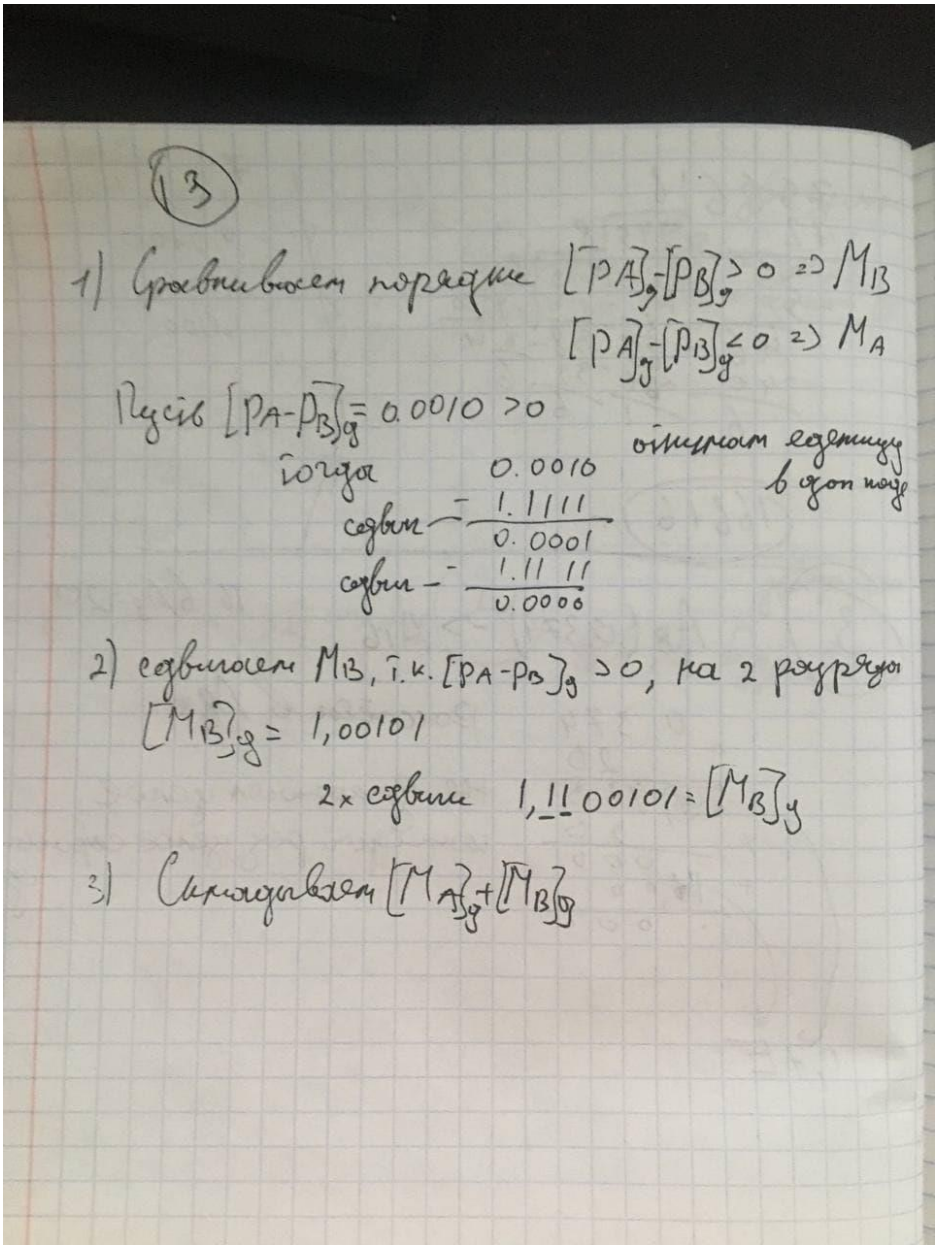
Простой сдвиг:

Исходная комбинация	Сдвиг влево	Сдвиг вправо
$0,a_1a_2\ldots a_n$	$a_1,a_2\ldots a_n0$	$0,0a_1a_2\ldots a_{n-1}$
$1,a_1a_2\ldots a_n$	$a_1,a_2\ldots a_n\alpha$	$0,1a_1a_2\ldots a_{n-1}$

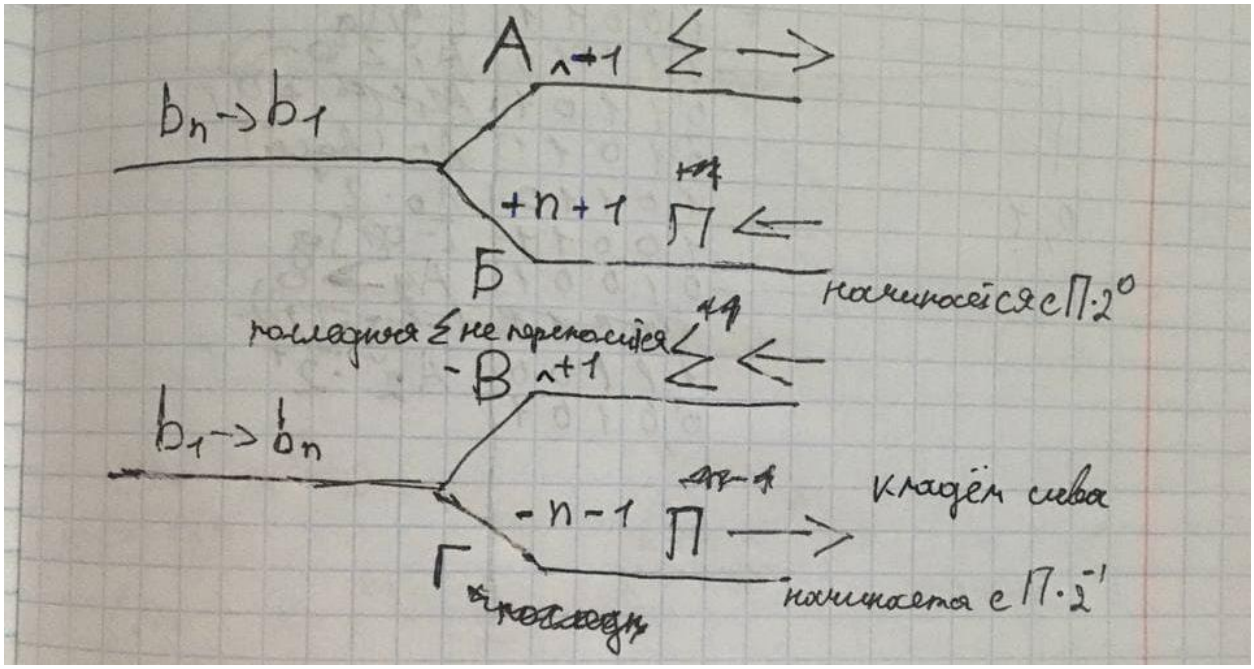
Модифицированный сдвиг − сдвиг, при котором в сдвигаемый разряд заносится значение, совпадающее со значением знакового разряда:

$00,a_1a_2\ldots a_n$	$0a_1,a_2\ldots a_n0$	$00,0a_1a_2\ldots a_{n-1}$
$01,a_1a_2\ldots a_n$	$1a_1,a_2\ldots a_n0$	$00,1a_1a_2\ldots a_{n-1}$
$10,a_1a_2\ldots a_n$	$0a_1,a_2\ldots a_n\alpha$	$11,0a_1a_2\ldots a_{n-1}$
$11,a_1a_2\ldots a_n$	$1a_1,a_2\ldots a_n\alpha$	$11,1a_1a_2\ldots a_{n-1}$

13. Сложение чисел с плавающей запятой



14. Умножение чисел в прямых кодах



15. Умножение чисел с хранением переносов

Пример. Мн = 0,1111
Мт = 0,1011

0,0000	$\Sigma_0^ч$
0,0000	регистр переносов
+ 0,1111	$П_1^ч = Мн \cdot b_4$
0,1111	$\Sigma_1^ч$
0,0000	регистр переносов
0,**0**111 1	$\Sigma_1^ч \cdot 2^{-1}$
+ 0,1111	$П_2^ч = Мн \cdot b_3$
0,1000 1	$\Sigma_2^ч$
0,0111	регистр переносов
0,**0**100 01	$\Sigma_2^ч \cdot 2^{-1}$
+ 0,0000	$П_3^ч = Мн \cdot b_2$
0,0011 01	$\Sigma_3^ч$
0,0100	регистр переносов
0,**0**001 101	$\Sigma_3^ч \cdot 2^{-1}$
+ 0,1111	$П_4^ч = Мн \cdot b_1$
1,0100 101	$\Sigma_4^ч$ (возникло переполнение)
0,**1**010 0101	$\Sigma_4^ч \cdot 2^{-1}$ (Мн · Мт)

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, Мн < 0,

Мт < 0, Мт < 0, то поправка [-Мн]доп

19. Умножение целых чисел в дополнительных кодах

Мн > 0, Мн < 0,

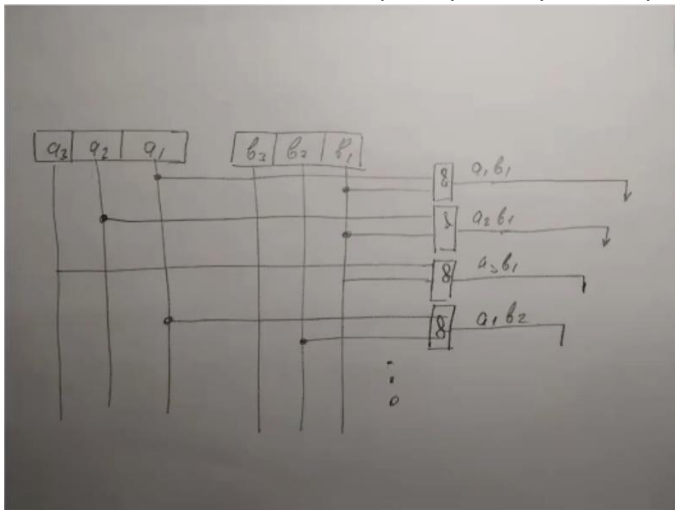
Мт < 0, Мт < 0, то поправка [-Мн]доп

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.



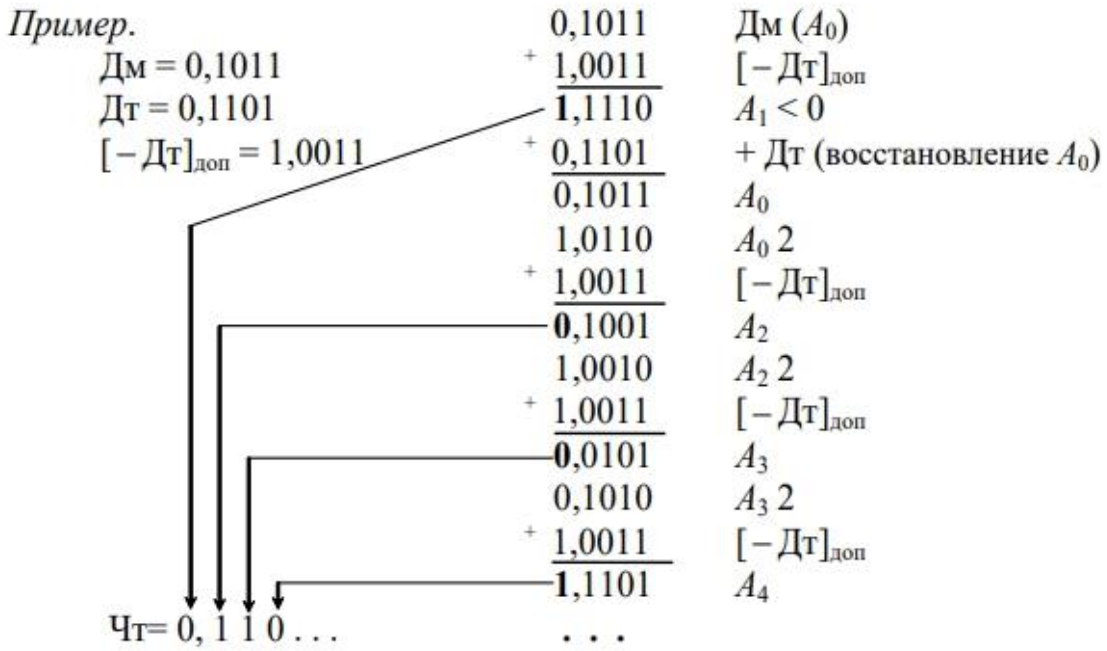
22.1 Машинные методы деления

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

22.2 Деление в прямых кодах

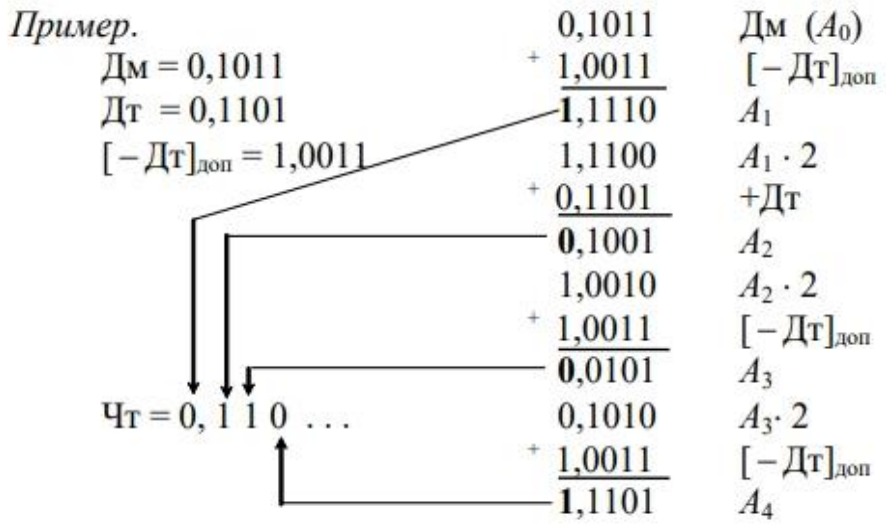
С восстановлением:

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



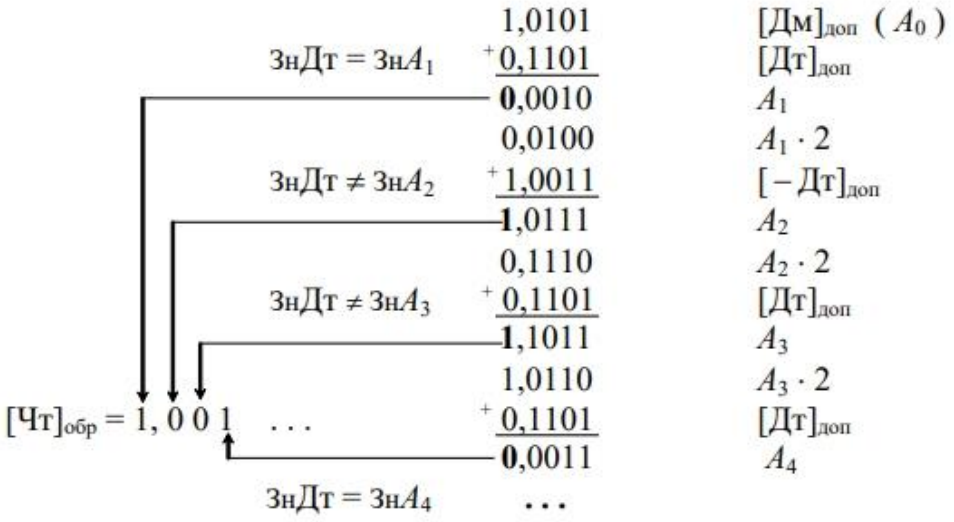
Без восстановления:

Дм + [-Дт]доп, A>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

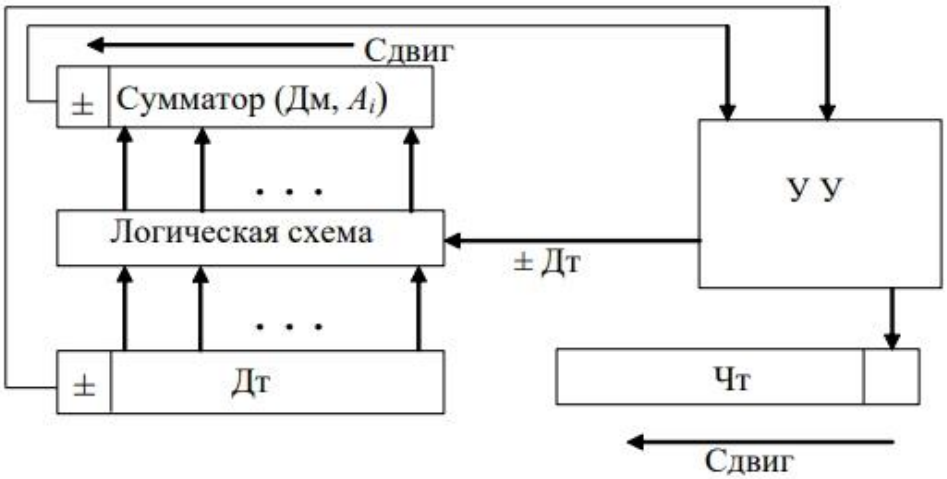


Рис. 12. Структурная схема устройства деления

Устройство управления (УУ), анализируя знаки сумматора (Дм, A_i) и делителя, вырабатывает сигнал, подаваемый на логическую схему, на выходе которой формируется ±Дт (для очередного вычитания). Кроме этого УУ формирует разряд частного.

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.

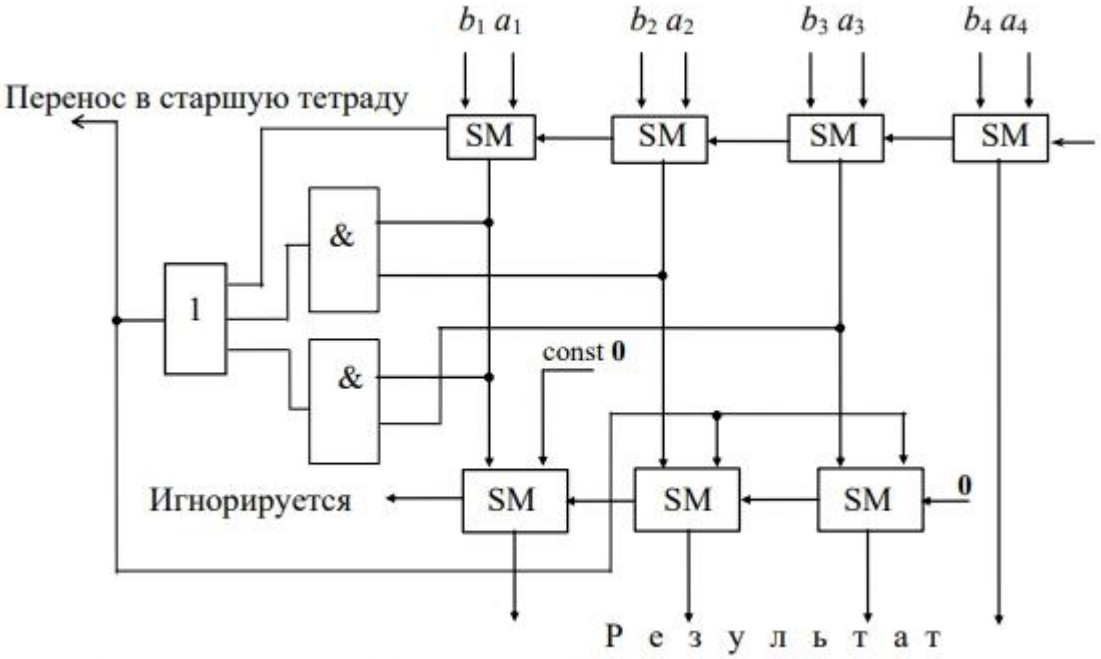


Рис. 14. Структурная схема одноразрядного двоично-десятичного сумматора

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

Пример. Необходимо сложить числа: $A = 378$ и $B = -169$ в *BCD*-коде.

A = 378	0. 0011 0111 1000
– B = 169	+1. 1110 1001 0110
$A - B = 209$	10. 0010 0000 1110
	циклический перенос 1
	0. 0010 0000 1111

Из последней тетрады нет переноса – это соответствует заему в нее 16 единиц (вместо необходимых десяти). Следовательно, из нее необходимо удалить лишние шесть единиц. Для этого в тетраду добавляется десятка (1010) – дополнение шести до шестнадцати:

	0. 0010 0000 1111
	1010
	0. 0010 0000 1001
	+ 2 0 9

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.

Пример. Необходимо сложить числа: $A = 169$ и $B = -378$ в *BCD*-коде.

$-A = 169$	0. 0001 0110 1001
$B = 378$	$^{+}$1. 1100 1000 0111
$A - B = -209$	1. 1101 1111 0000
	0110
	1. 1101 1111 0110
	0010 0000 1001
	− 2 0 9

Из последней тетрады есть перенос, значит в нее произошел заем. Для удаления 6 из отрицательной тетрады результата необходимо добавить в нее 6.

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:

№ набора	$x_1x_2x_3$	f
0	000	0
1	001	1
2	010	0
3	011	0
4	100	1
5	101	1
6	110	0
7	111	1

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

$x_1, x_2, \ldots, x_{j-1}$	$x_j, x_{j+1}, \ldots, x_n$			
	00...0	0...1	...	11...1
00...0	0	1	...	1
00...1	1	0	...	0
...	...	...	...	...
11...1	1	0	...	1

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

33. Формы представления функций алгебры логики

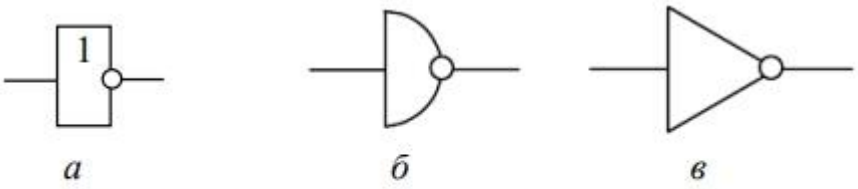
Существует несколько общепринятых стандартов условных обозначений логиче ских элементов. Среди них наиболее распространенными являются американский стандарт milspec806B и стандарт МЭК 117-15 А, созданный Международной Электротехнической Комиссией. Часто в литературе используются обозначения элементов 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:

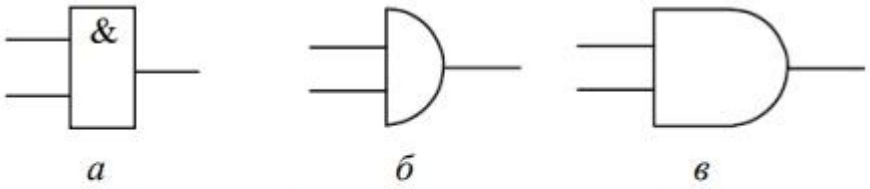
1) Логическое отрицание



а *б* *в*

Рис. 15. УГО логического элемента НЕ:
а – по ГОСТ и МЭК; *б* – по DIN; *в* – по milspec

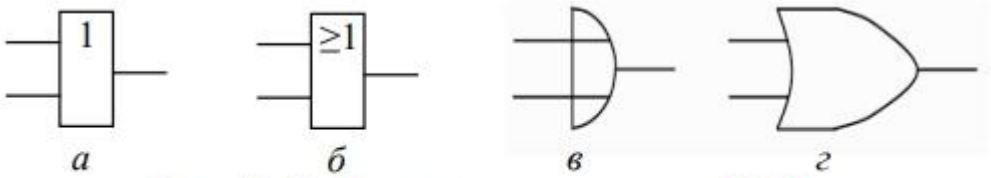
2) Логическое умножение



а *б* *в*

Рис. 16. УГО логического элемента И:
а – по ГОСТ и МЭК; *б* – по DIN; *в* – по milspec

3) Логическое сложение



а *б* *в* *г*

Рис. 17. УГО логического элемента ИЛИ:
а – по ГОСТ; *б* – по МЭК; *в* – по DIN; *г* – по milspec

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

Распределительный закон:

– для дизъюнкции: $x_1 \vee x_2 \cdot x_3 = (x_1 \vee x_2)(x_1 \vee x_3)$, т. е. дизъюнкция переменной и конъюнкции эквивалентна конъюнкции дизъюнкций этой переменной с сомножителями;

– для конъюнкции: $x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3$, т. е. конъюнкция переменной и дизъюнкции равносильна дизъюнкции конъюнкций этой переменной со слагаемыми.

Закон инверсии (*правило де Моргана*):

– для дизъюнкции: $\overline{x_1 \vee x_2} = \overline{x_1} \cdot \overline{x_2}$;

– для конъюнкции: $\overline{x_1 \cdot x_2} = \overline{x_1} \vee \overline{x_2}$, т. е. отрицание дизъюнкции (конъюнкции) переменных равно конъюнкции (дизъюнкции) отрицаний этих переменных.

– всегда истинны высказывания: $x \vee 1 = 1$; $x \vee \overline{x} = 1$; $\overline{0} = 1$;
– всегда ложны высказывания: $x \cdot 0 = 0$; $x \cdot \overline{x} = 0$; $\overline{1} = 0$;
– правило двойного отрицания: $\overline{\overline{x}} = x$;
– правило повторения (идемпотентности): $x \vee x \vee \ldots \vee x = x$;
$x \cdot x \cdot \ldots \cdot x = x$.

Закон рефлексивности:

– для дизъюнкции: $x \vee x = x$;
– для конъюнкции: $x \cdot x = x$.

Переместительный закон:

– для дизъюнкции: $x_1 \vee x_2 = x_2 \vee x_1$;
– для конъюнкции: $x_1 \cdot x_2 = x_2 \cdot x_1$;
– для сложения по модулю два: $x_1 \oplus x_2 = x_2 \oplus x_1$.

Сочетательный закон:

– для дизъюнкции: $x_1 \vee (x_2 \vee x_3) = (x_1 \vee x_2) \vee x_3$;
– для конъюнкции: $x_1 \cdot (x_2 \cdot x_3) = (x_1 \cdot x_2) \cdot x_3$;
– для суммы по модулю два: $x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3$, т. е. группирова-

35.1 Классы функций алгебры логики

Полином Жегалкина

$$f(x_1 x_2 \ldots x_n) = k_0 \oplus k_1 x_1 \oplus k_2 x_2 \oplus \ldots \oplus k_n x_n$$

1) Класс линейных функций:

$$f(x_1x_2...x_n) = k_0 \oplus k_1x_1 \oplus k_2x_2 \oplus ... \oplus k_nx_n.$$

$k_2\ k_1\ k_0$	$f(x_1x_2) = k_0 \oplus k_1x_1 \oplus k_2x_2$
0 0 0	0
0 0 1	1
0 1 0	x_1
0 1 1	$x_1 \oplus 1$
1 0 0	x_2
1 0 1	$x_2 \oplus 1$
1 1 0	$x_1 \oplus x_2$
1 1 1	$1 \oplus x_1 \oplus x_2$

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.

$$f_1(x_1, x_2, ..., x_n) = \overline{f_2}(\bar{x}_1, \bar{x}_2, ..., \bar{x}_n).$$

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 конъюнкцию без инверсии, если их значение координатах равно единице и с инверсией, если равно нулю.

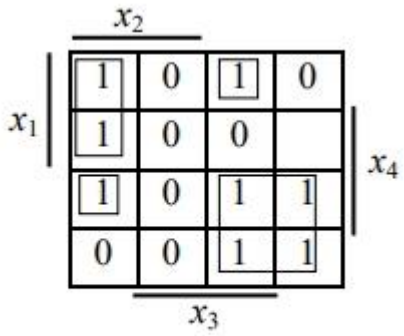


Рис. 27. Карта Вейча для $f_{ДНФ}$

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

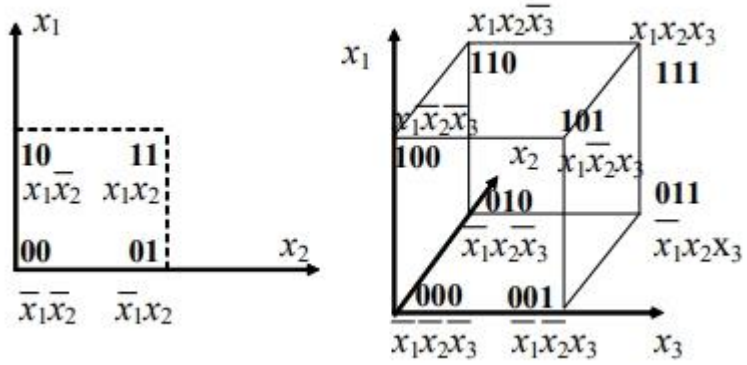
Звездочками на карте отмечаются наборы, на которых функция f не определена.

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 x1x2.

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 (x) 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

$f_{\text{СДНФ}}(x_1x_2x_3x_4) = \text{V}\,(2, 3, 4, 6, 9, 10, 11, 12)$.

Сформируем кубический комплекс K, состоящий из кубов нулевой размерности:

$K = (0010, 0011, 0100, 0110, 1001, 1010, 1011, 1100)$.

Выполним разбиение комплекса K на группы, получим

$$K_1^0 = \begin{Bmatrix} 0010 \\ 0100 \end{Bmatrix}, \quad K_2^0 = \begin{Bmatrix} 0011 \\ 1100 \\ 0110 \\ 1001 \\ 1010 \end{Bmatrix}, \quad K_3^0 = \{1011\}.$$

Попарное сравнение можно проводить только между соседними по номеру группами кубов (K_1^i и K_2^{i+1}), так как кубы, порождающие новые кубы, должны иметь кодовое расстояние, равное единице. Если сравниваемые кубы различаются только в одном разряде, то в следующий столбец записывается код с символом x на месте этого разряда, т. е. производится операция полного склеивания, в результате которой формируется куб следующего ранга (ранг определяется количеством свободных координат x в нем). В результате сравнения кубов получим:

$$K_1^0 \text{ и } K_2^0 \Rightarrow K_1^1 = \begin{Bmatrix} 001x \\ 0x10 \\ x010 \\ x100 \\ 01x0 \end{Bmatrix}, \qquad K_2^0 \text{ и } K_3^0 \Rightarrow K_2^1 = \begin{Bmatrix} x011 \\ 10x1 \\ 101x \end{Bmatrix}.$$

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 (С – сумма, П – перенос в следующий (старший) разряд). Обозначением полного двоичного сумматора служат буквы SM.

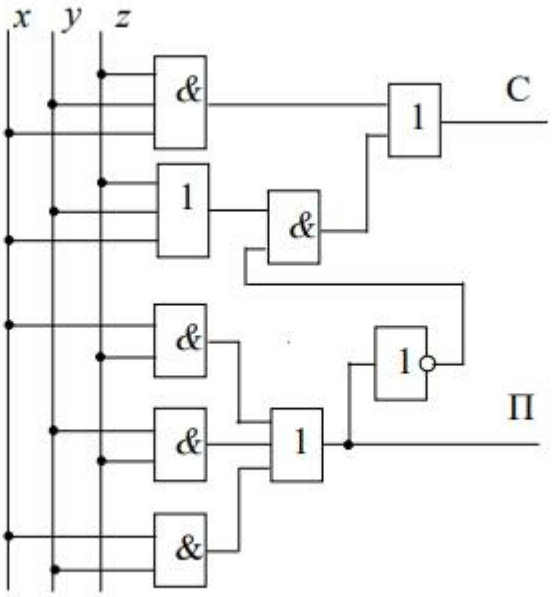
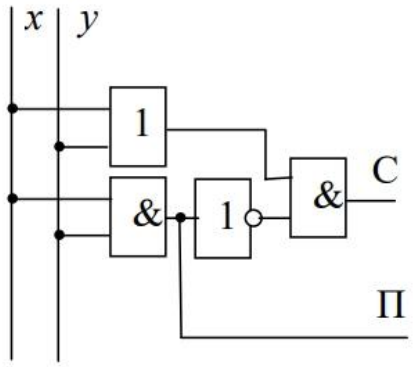


Рис. 47 Логическая схема полного комбинационного сумматора

43. Синтез одноразрядного комбинационного полусумматора

Полусумматор имеет два входа x и y для разрядов двух слагаемых и два выхода: С – 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

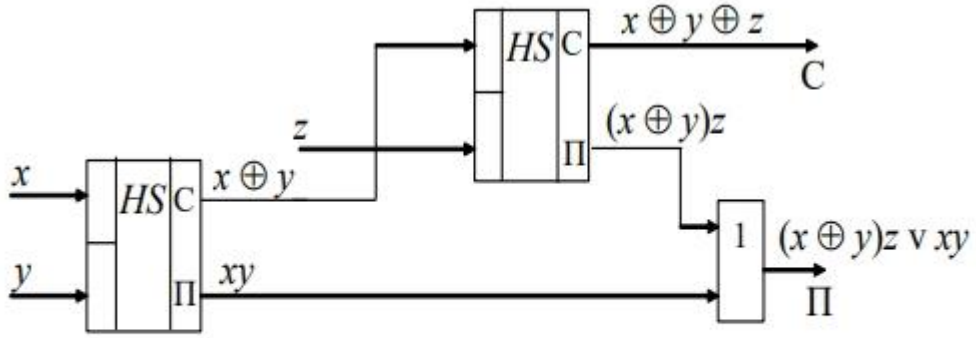
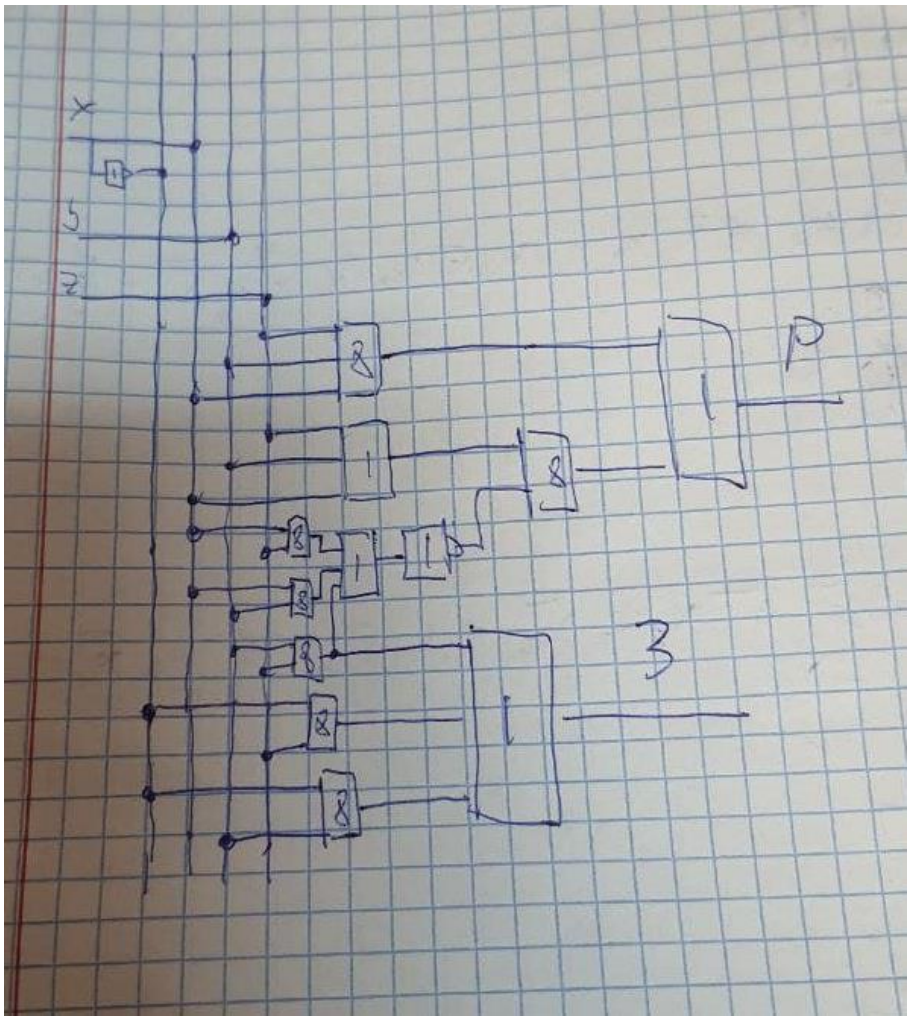


Рис. 48. Схема сумматора на двух полусумматорах и ИЛИ

45. Синтез одноразрядного комбинационного вычитателя



46. Объединенная схема одноразрядного комбинационного сумматора-вычитателя

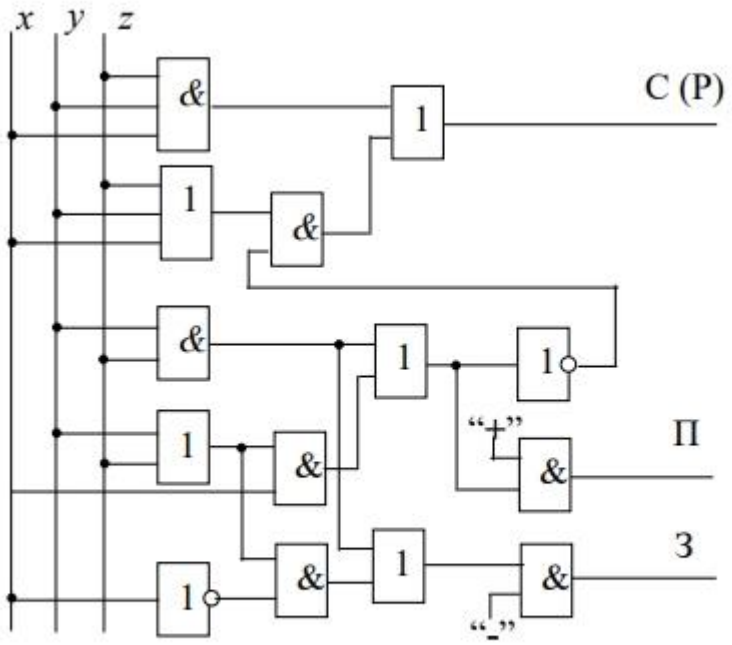


Рис. 49 Логическая схема полного комбинационного сумматора-вычитателя

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 x 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 Т-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.

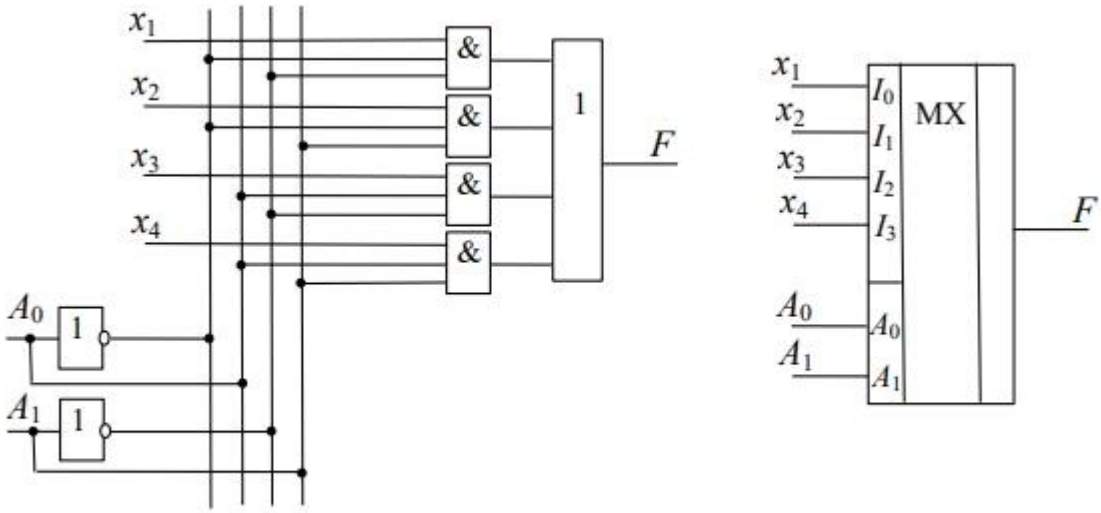


Рис. 52. Структура и УГО мультиплексора

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.

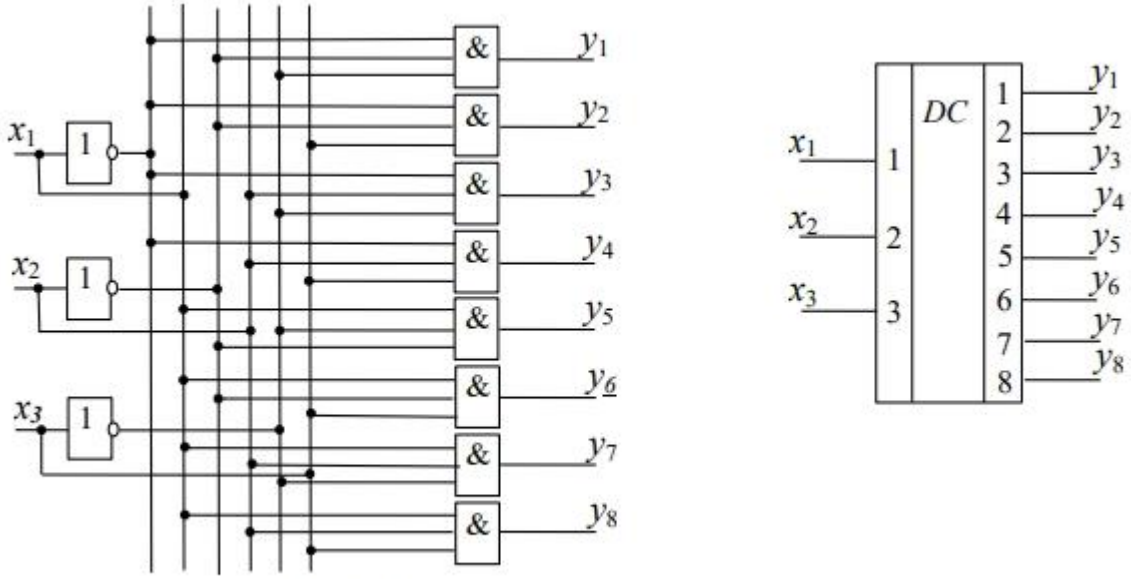


Рис. 51. Структура и УГО дешифратора