Информатика

Лекция 6

Метод Квайна

- способ представления функции в ДНФ или КНФ с минимальным количеством членов и минимальным набором переменных

- Преобразование функции можно разделить на два этапа:
- на первом этапе осуществляется переход от канонической формы (СДНФ или СКНФ) к так называемой **сокращённой форме**;
- на втором этапе — переход от сокращённой формы к **минимальной форме**.

Первый этап (получение сокращённой формы)

Представим, что заданная функция f представлена в СДНФ. Для осуществления первого этапа преобразование проходит два действия:

1. *Операция склеивания*;
2. *Операция поглощения*.

Операция склеивания сводится к нахождению пар членов, соответствующих виду $w \cdot x$ или $w \cdot \overline{x}$, и преобразованию их в следующие выражения: $w \cdot x \vee w \cdot \overline{x} = w \cdot (x \vee \overline{x}) = w$. Результаты склеивания w теперь играют роль дополнительных членов. Необходимо найти все возможные пары членов (каждый член с каждым).

Получение сокращённой формы

Потом выполняется *операция поглощения*. Она основана на равенстве $w \vee w \cdot z = w \cdot (1 \vee z) = w$ (член w поглощает выражение $w \cdot z$). Вследствие этого действия из логического выражения вычёркиваются все члены, поглощаемые другими переменными, результаты которых получены в *операции склеивания*.

Обе операции первого этапа могут выполняться до тех пор, пока это может быть осуществимо.

Пример

Пусть есть таблица истинности

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$$\text{СДНФ}(f) = \overline{x}_1\overline{x}_2x_3 \vee x_1\overline{x}_2\overline{x}_3 \vee x_1\overline{x}_2x_3 \vee x_1x_2\overline{x}_3 \vee x_1x_2x_3$$

После операции склеивания наша функция принимает вид:

$$f = x_1\overline{x}_2 \vee x_1x_2 \vee x_1x_3 \vee x_1\overline{x}_3 \vee \overline{x}_2x_3$$

Результаты упрощения

Получившиеся пять членов представляют собой результаты склеивания. Далее можно приступить как к поглощению, так и произвести новые склеивания:

$$f = x_1 \vee x_1\overline{x}_2 \vee x_1x_2 \vee \overline{x}_2x_3$$

Теперь действительно наступает этап поглощения:

$$f = x_1 \vee \overline{x}_2x_3$$

В такой форме функция состоит из так называемых простых импликант функции. Это наиболее простое представление функции по сравнению с её СДНФ.

Второй этап (табличный)

- Рассмотренный выше пример уже удовлетворяет определению минимальной формы, однако далеко не всегда после первого этапа сокращённая форма совпадает с минимальной.
- Ещё могут оставаться члены, чьё удаление не изменяет конечный результат.
- На данном этапе требуется удалить лишние переменные.

Пример

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

x_1	x_2	x_3	x_4	$f(x_1, x_2, x_3, x_4)$
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

Получение СДНФ

СДНФ

$$(f) = \overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4 \vee \overline{x}_1\overline{x}_2\overline{x}_3x_4 \vee \overline{x}_1\overline{x}_2x_3\overline{x}_4 \vee$$

$$\overline{x}_1x_2x_3\overline{x}_4 \vee x_1x_2x_3\overline{x}_4 \vee x_1x_2x_3x_4$$

После первого этапа функция принимает вид:

$$f = x_1x_2x_3 \vee \overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}_1\overline{x}_2\overline{x}_4 \vee \overline{x}_1x_3\overline{x}_4 \vee x_2x_3\overline{x}_4$$

Обработка импликантной матрицы.

- Мы вновь получили дизъюнкцию простых импликант, на этот раз в количестве пяти штук.
- Чтобы получить минимальную форму, воспользуемся импликантной матрицей.
- Столбцы в ней соответствуют членам СДНФ, а строки — членам сокращённой формы.
- Отмечаются столбцы членов СДНФ, которые поглощаются отдельными простыми импликантами:

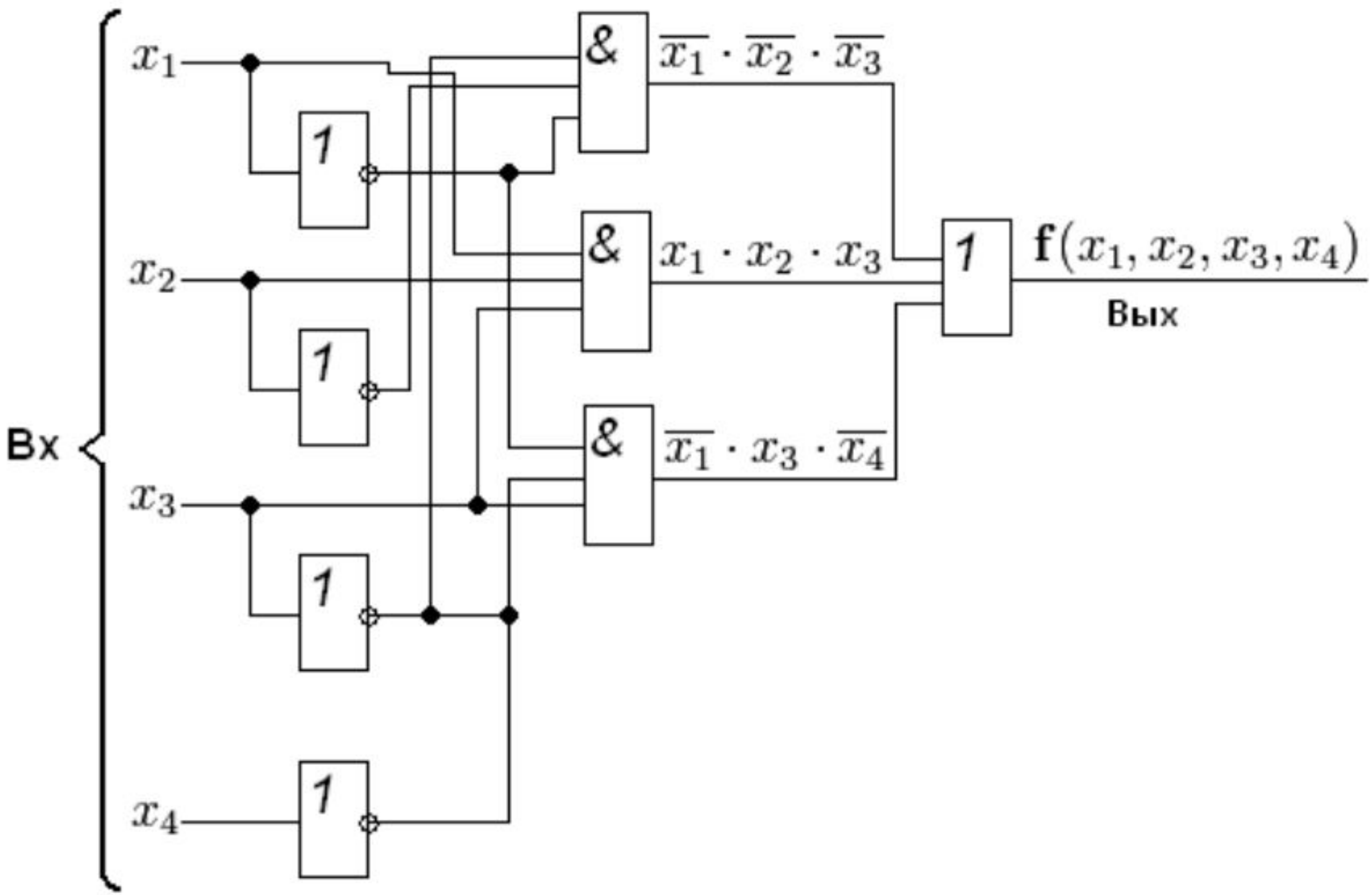
	$\overline{x}_1\overline{x}_2\overline{x}_3\overline{x}_4$	$\overline{x}_1\overline{x}_2\overline{x}_3x_4$	$\overline{x}_1\overline{x}_2x_3\overline{x}_4$	$\overline{x}_1x_2x_3\overline{x}_4$	$x_1x_2x_3\overline{x}_4$	$x_1x_2x_3x_4$
$\overline{x}_1\overline{x}_2\overline{x}_3$	$\times$	$\times$				
$\overline{x}_1\overline{x}_2\overline{x}_4$	$\times$		$\times$			
$\overline{x}_1x_3\overline{x}_4$			$\times$	$\times$		
$x_2x_3\overline{x}_4$				$\times$	$\times$	
$x_1x_2x_3$					$\times$	$\times$

- Импликанты, не подлежащие исключению, образуют ядро.
- Такие импликанты определяются по вышеуказанной матрице.
- Для каждой из них имеется хотя бы один столбец, перекрываемый только этой импликантой.
- Исключить все остальные члены сокращённой формы, однако, нельзя, так как это может привести к превращению какой-либо другой импликанты в нелишнюю.

- Выбор остальных импликант, что войдут в минимальную форму, сводится к нахождению минимального набора неядровых импликант, которые покроют столбцы, не перекрываемые ядровыми.
- Применительно к рассматриваемому случаю это будут третий и четвёртый столбец.
- То, что можно исключить остальные импликанты, легко проверяется. На соответствующих наборах мы получаем значение 1, которая получается и на удалённых импликантах.

$$f = \overline{x}_1\overline{x}_2\overline{x}_3 \vee x_1x_2x_3 \vee \overline{x}_1x_3\overline{x}_4$$

Структурная схема, при минимизации функции методом Квайна



Использование метода для получения минимальной КНФ

Для получения Минимальной конъюнктивной нормальной формы (МКНФ), используя метод Куайна, вводятся следующие критерии:

- для минимизации берётся не СДНФ, а СКНФ функции;
- склеиваемые пары членов меняются на: $w \vee x$ или $w \vee \overline{x}$;
- правило операции поглощения выглядит следующим образом:

$$z \cdot (z \vee y) = z \vee z \cdot y = z \cdot (1 \vee y) = z$$

Метод Квайна—Мак-Класки

1. Нахождение первичных импликант;
2. Эти импликанты разбиваются на группы, в каждую группу входят импликанты с равным количеством единиц (нулей);
3. Производится попарное сравнение эквивалентов (термов) в соседних группах, с целью формирования термов более низких рангов;
4. Составляется таблица, где строкам соответствуют первичные импликанты, а заголовкам столбцов — термы низких рангов;
5. Расставляются метки, отражающие поглощение термов высших рангов (исходных термов).
6. Находятся существенные импликанты;
7. Вычёркиваются лишние столбцы;
8. Вычёркиваются лишние первичные импликанты;
9. Выбирается минимальное покрытие максимальными интервалами.

Пример

y=f(0011,0100,0101,0111,1001,1101,1110,1111)

- Положим, что функция записана в виде СДНФ. Тогда первичными импликантами будут 010~, 0~11, 1~01, 111~, ~1~1.

	0011	0100	0101	0111	1001	1101	1110	1111
010~		×	×					
0~11	×			×				
1~01					×	×		
111~							×	×
~1~1			×	×		×		×

- Существенными импликантами будут те, где в соответствующих столбцах стоит только одна метка.
- Без любой из них не будет полного покрытия исходных минтермов.
- Итого получается, что в данном примере существенные импликанты — 0~11, 010~, 1~01, 111~.

- Убираем лишние строки (в данном случае только одну) и выберем такую совокупность первичных импликант, что включает метки во всех столбцах. Предпочтение отдаётся минимальной выборке.
- Получаем искомую форму

$$y = \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_3 x_4 \vee x_1 \bar{x}_3 x_4 \vee x_1 x_2 x_3$$

Достоинства метода

- Его можно применять на большом количестве переменных в САПР, с использованием ЭВМ для минимизации полностью или частично определённых функций;
- Не принципиально, задана функция в СДНФ или СКНФ;
- Удобно минимизировать системы булевых функций в связи с простым выделением общих частей реализуемой системы ФАЛ;
- Данный метод — алгоритмически систематический, он легко формализуется и алгоритмизируется. Не зависит от навыков разработчика;
- Позволяет последовательно осуществить все этапы минимизации (склеивание и выявление лишних импликант, получение минимальных покрытий).

Недостатки метода

- Затруднительна ручная минимизация функций с шестью и более переменных;
- Метод Куайна — Мак-Класки алгоритмически неинвариантен: время работы растёт экспоненциально с увеличением входных данных.
- Можно показать, что для функции от n переменных верхняя граница количества основных импликант $3^n/n$. Если $n = 32$ их может быть больше чем $6.5 * 10^{15}$.

Трансвычисли́тельная зада́ча

- в теории сложности вычислений задача, для решения которой требуется обработка более чем 10^{93} бит информации.
- Число 10^{93}, называемое «пределом Бремерманна»

Сумматоры

- **Сумматор логический** операционный узел, выполняющий **арифметическое** сложение кодов двух чисел.
- При арифметическом сложении выполняются и другие дополнительные операции: учет знаков чисел, выравнивание порядков слагаемых и тому подобное.
- Указанные операции выполняются в арифметическо-логических устройствах (АЛУ) или процессорных элементах, ядром которых являются сумматоры.

По числу входов и выходов

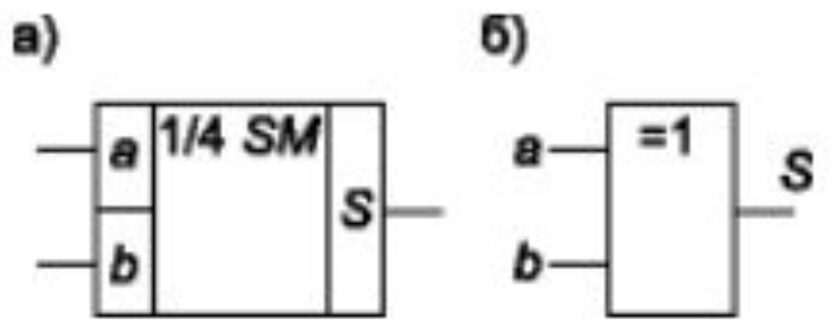
- четвертьсумматоры;
- полусумматоры;
- полные одноразрядные двоичные сумматоры

По количеству одновременно обрабатываемых разрядов складываемых чисел:

- одноразрядные,
- многоразрядные.

Четвертьсумматор

$$S = \bar{a}b + a\bar{b} = a \oplus b$$



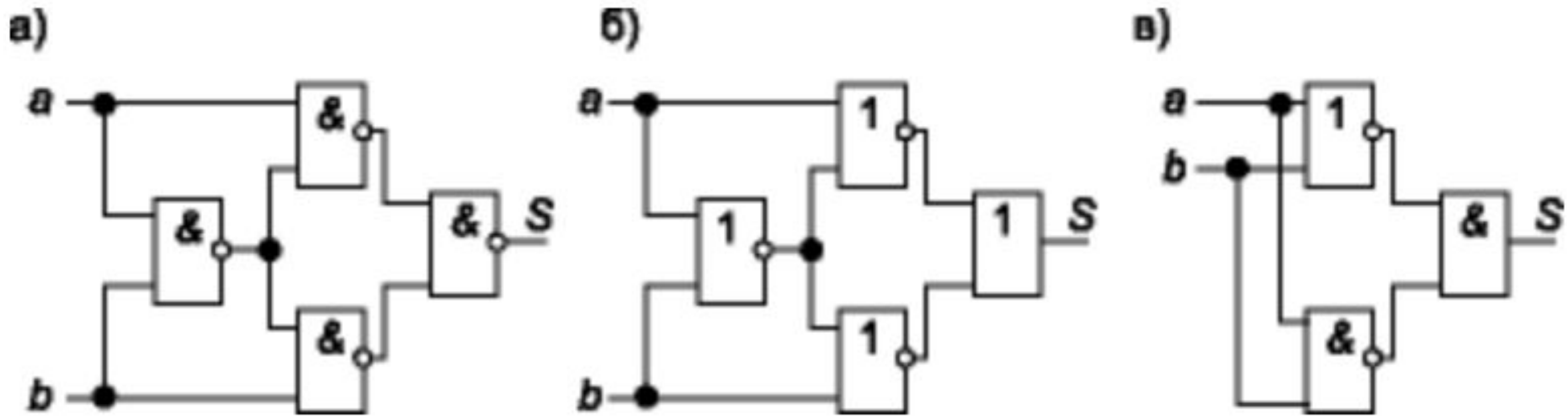
Реализация сумматора

$$S=\bar{a}b+a\bar{b}=\bar{a}a+\bar{a}b+\bar{b}b+a\bar{b}=$$
$$=a(\bar{a}+\bar{b})+b(\bar{a}+\bar{b})=a\overline{ab}+b\overline{ab}=\overline{\overline{a\overline{ab}}\bullet\overline{b\overline{ab}}}$$

$$S=\bar{a}b+a\bar{b}=\bar{a}a+\bar{a}b+\bar{b}b+a\bar{b}=$$
$$=\overline{\overline{\bar{a}(a+b)}}+\overline{\overline{\bar{b}(a+b)}}=\overline{a+\overline{a+b}}+\overline{b+\overline{a+b}}$$

$$S=\bar{a}b+a\bar{b}=\bar{a}a+\bar{a}b+\bar{b}b+a\bar{b}=$$
$$=\bar{a}(a+b)+\bar{b}(a+b)=(a+b)(\bar{a}+\bar{b})=(a+b)\overline{ab}$$

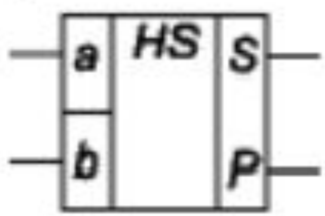
Реализация сумматора



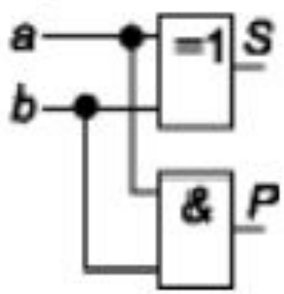
Полусумматор

$$\left.\begin{aligned} &S=\bar{a}b+a\bar{b}=a\oplus b \\ &P=ab \end{aligned}\right\}$$

а)

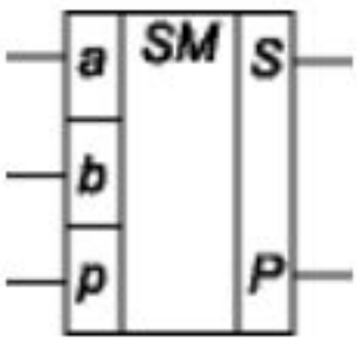


б)



a	b	P	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

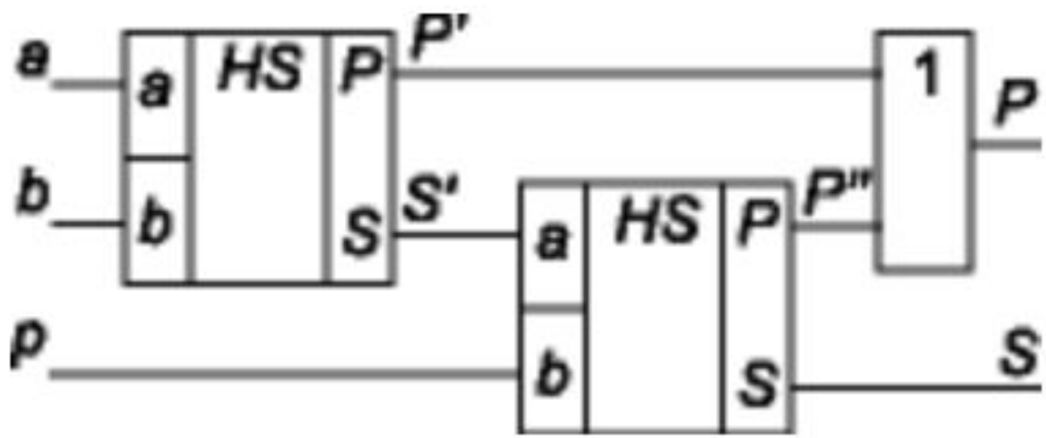
Полный одноразрядный двоичный сумматор



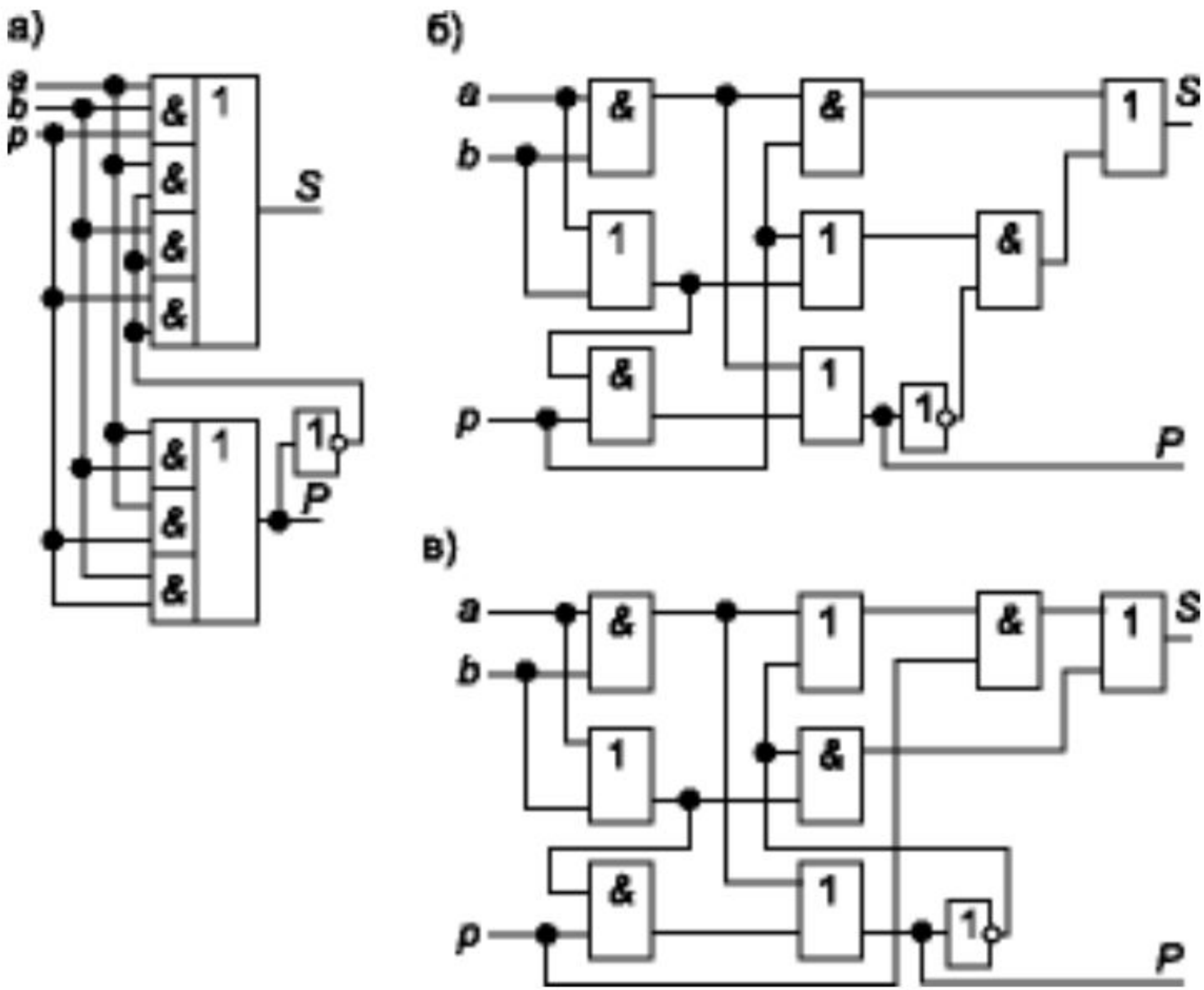
$$\left.\begin{aligned} S &= \bar{a}\bar{b}p + \bar{a}b\bar{p} + a\bar{b}\bar{p} + abp \\ P &= \bar{a}bp + a\bar{b}p + ab\bar{p} + abp \end{aligned}\right\}$$

⊥ наб.	a	b	p	P	S
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

Реализация на двух полусумматорах и одном элементе ИЛИ



Схемы реализации



Полувычитатель

$D_i = \bar{A}_i B_i + A_i \bar{B}_i = A_i \oplus B_i;$

$V_i = \bar{A}_i * B_i .$

Входы		Выходы	
A_i	B_i	D_i	V_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0
A_i - B_i			

Реализация полувычитателя

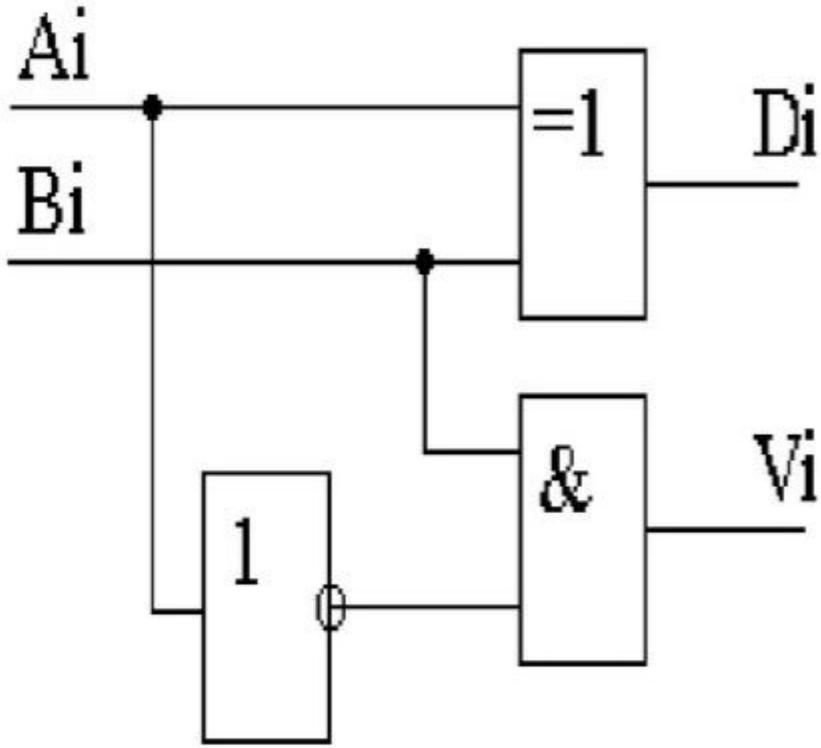
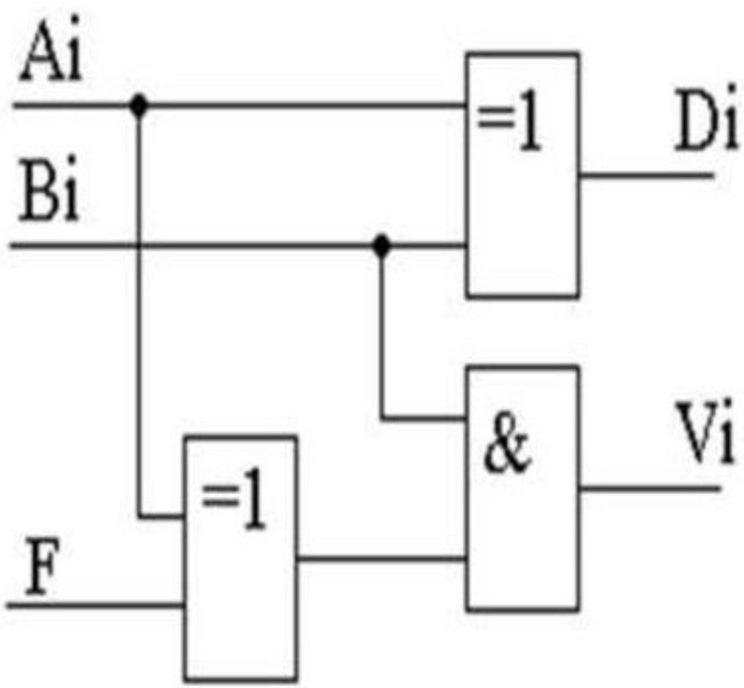


Схема полувычитателя отличается от схемы полусумматора только наличием инвертора по сигналу А.

Универсальное устройство



F = “1” – полусумматор;
F = “0” - полувычитатель.

ТИ полного вычитателя

Входы			Выходы	
A_i	B_i	P_{i-1}	D_i	V_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1
$A_i - B_i - V_{i-1}$				

Таблица истинности полного вычит

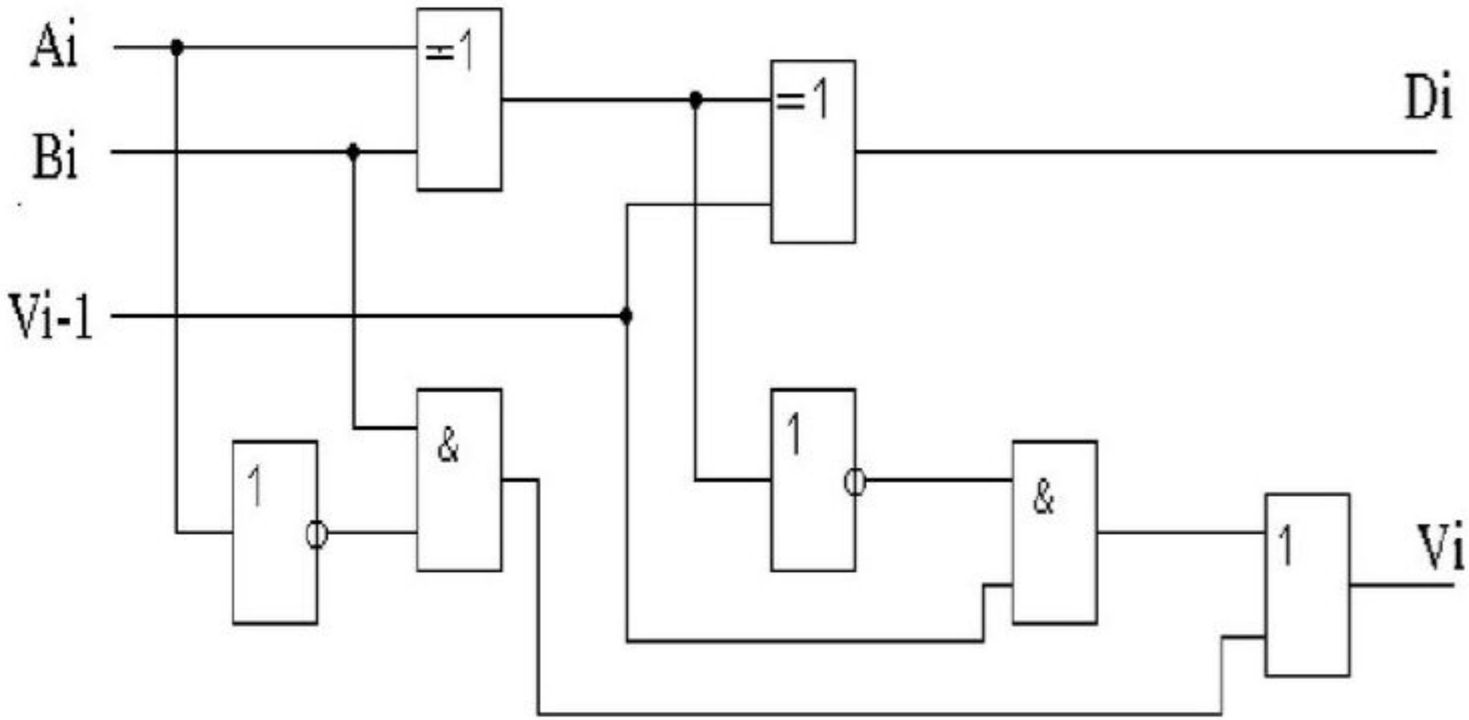
Как следует из таблиц истинности для сумматора и вычитателя выходные сигналы суммы и разности совпадают, т.е.

$$\mathbf{D_i = V_{i-1} \oplus A_i \oplus B_i.}$$

Выражение для заема, полученное по приведенной карте Карно, имеет вид:

$$\mathbf{V_i = \overline{A}_i\ B_i + V_{i-1}\,(\overline{A_i \oplus B_i}).}$$

Схема полного вычитателя



Универсальное устройство