Тема 6. Логические основы ЭВМ

Основные вопросы

6.1. Основные сведения из алгебры логики
6.2. Законы алгебры логики
6.3. Понятие о минимизации логических функций
6.4. Техническая интерпретация логических функций

6.1.Основные сведения из алгебры логики

Теоретической основой построения ЭВМ являются специальные математические дисциплины. Одна из них — алгебра логика или булева алгебра, названная по имени основоположника этой дисциплины Дж. Буля — английского математика 19-го столетия. Аппарат алгебры логики широко используют для описания схем ЭВМ, их оптимизации и проектирования.

Вся информация в ЭВМ представляется в двоичной системе счисления. Поставим в соответствие входным сигналам отдельных устройств ЭВМ соответствующие значения x_i $(i=1,n)$, а выходным сигналам — значения функций y_j $(j=1,m)$ (см. рис. 6.1).

Рис.6.1. Представление схемы ЭВМ

В этом случае зависимостями

$$y_j=f_i(x_1,x_2,\ldots,x_i,\ldots,x_n), \qquad (5.1)$$

где x_i — i-й вход; n — число входов; y_j — j-й выход; m — число выходов в устройстве, можно описывать алгоритм работы любого устройства ЭВМ. Каждая такая зависимость y , является “булевой функцией, у которой число возможных состояний и каждой ее независимой переменной равно двум” (стандарт ISO 2382/2-76), т.е. функцией алгебры логики, а ее аргументы определены на множестве $\{0,1\}$. Алгебра логика устанавливает основные законы формирования и преобразования логических функций. Она позволяет представить любую сложную функцию в виде композиции простейших функций. Известно, что количество всевозможных функций N от n аргументов выражается зависимостью: $N = 2^{2^n}$.

Рассмотрим наиболее употребительные из них.

При n=0 можно определить две основные функции *(N=2),* не зависящие от каких-либо переменных:

y_0 , тождественно равную нулю (**y_0=0**), и

y_1 , тождественно равную единице (**y_1=1**).

Технической интерпретацией функции y_1=1 может быть **генератор импульсов**. При отсутствии входных сигналов на выходе этого устройства всегда имеются импульсы (единицы). Функция y_0=*0* может быть интерпретирована **отключенной схемой**, сигналы от которой не поступают ни к каким устройствам.

При n=1 число различных функций N=4. Представим зависимость значений этих функций от значения аргумента x в виде специальной таблицы истинности (табл. 6.1).

Таблица 6.1

Таблица функций от одной переменной

Y / x	Y_0	Y_1	Y_2	Y_3
0				
1	0	1	1	0

Таблицы истинности получили такое название, потому что они определяют значение функции в зависимости от комбинации входных сигналов. В этой таблице, как и ранее, y_0=0 и y_1=1. Функция y_2=*x,* а функция y_3=⌐*x* — (инверсия *x*). Этим функциям соответствуют определенные технические аналоги. Схема, реализующая зависимость **y_2=x**, называется **повторителем**, а схема **y_3=⌐x - инвертором**.

При n = 2, N = l6, т.е. от двух переменных, можно построить 16 различных функций. В табл. 6.2 представлена часть из них, имеющая фундаментальное значение при построении основных схем ЭВМ.

Таблица 6.2.

Таблица функций от двух переменных

Y_i / **X_1 X_2**	**Y_0**	**Y_1**	**Y_2**	**Y_3**	**Y_4**	**Y_5**	**Y_6**	**Y_7**	**Y_8**	**Y_9**	**Y_{10}**	**.**	**Y_{15}**
00	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**		
01	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**1**		
10	**0**	**1**	**1**	**0**	**1**	**0**	**0**	**1**	**0**	**1**	**0**		
11	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**		

Y_4 — логическое сложение, дизъюнкция **$Y_4 = X_1 \vee X_2$**

Y_5 — инверсная функция к Y_4 — «не-или» , «стрелка Пирса», **Y_5 = ⌐Y_4 = $X_1 \circ X_2$**

Y_6 — логическое умножение — конъюнкция **$Y_6 = X_1$ & X_2**

Y_7 — инверсная функция к Y_6 — «не-и» , «штрих Шеффера», **$Y_7 = X_1 / X_2$**

Y_8 — логическая равнозначность или эквивалентность **$Y_8=X_1 \sim X_2$**

Y_9 — логическая разноименность или сложение по модулю 2, **Y_9 =⌐ Y_8,** Y_9=**$X_1 \Delta X_2$**

Y_{10} — логическая импликация, $Y_{10} = X_1 \rightarrow X_2$

В левой части таблицы перечислены всевозможные комбинации входных переменных (наборы значений), а в правой — возможные реакции выходных сигналов. В табл.6.2. представлены функции y_0—y_3, полностью соответствующие функциям табл.6.1, а также новые, часто используемые и интересные функции y_4—y_{10}. При этом местоположение функций и их нумерация в таблице особого значения не имеют. По данной таблице нетрудно составить аналитическое выражение (зависимость) для каждой функции от двух аргументов вида (6.1). Для этого наборы переменных, на которых функция принимает значение единицы, записываются как конъюнкции (логическое умножение) и связываются знаками логического сложения. Такие формы функций получили название **дизъюнктивных нормальных форм (ДНФ)**. Если в этих функциях конъюнкции содержат все без исключения переменные в прямом или инверсном значениях, то такая форма функций называется **совершенной или СДНФ**.

Рассмотрим по отдельности логические функции с y_4 *по* y_9.

Функция y_4 представляет собой функцию **логического сложения, дизъюнкцию**. Функция y_4 принимает значение единицы, если значение единицы имеет хотя бы одна переменная x_1 или x_2:

$$y_4 = x_1\bar{x}_2 \vee \bar{x}_1 x_2 \vee x_1 x_2 = x_1 \vee x_2$$

Тождественность приведенных аналитических зависимостей можно установить, пользуясь законами алгебры логики, приведенными ниже.

Функция y_5 является **инверсной функцией по отношению к y_4**:

$$y_5 = \overline{y_4} = \overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$$

Она имеет название “отрицание дизъюнкции”. Иногда в литературе встречается ее специальное название “стрелка Пирса”, по фамилии математика, исследовавшего ее свойства.

Функция y_6 является функцией **логического умножения**. Она очень похожа на операцию обычного умножения и принимает значение единицы в тех случаях, когда все ее переменные равны единице:

$$y_6 = x_1 \wedge x_2$$

Функция y_7 является **инверсной функцией по отношению *к* y_6***:*

$$y_7 = \overline{y_6} = \overline{x_1 \vee x_2} = \overline{x_1} \wedge \overline{x_2}$$

Она называется “отрицание конъюнкции” или “штрих Шеффера”. Функция y_8 называется **логической равнозначностью**, она принимает значение единицы, если все ее переменные имеют одинаковое значение (или 0 или 1):

$$y_8 = \overline{x_1} \wedge \overline{x_2} \vee x_1 \wedge x_2$$

Функция y_9 **является инверсной по отношению к y_8**:

$$y_9 = \overline{y_8}$$

$$y_9 = \overline{\bar{x}_1 \wedge \bar{x}_2 \vee x_1 \wedge x_2} = \overline{x_1} \wedge x_2 \vee x_1 \wedge \bar{x}_2$$

Она принимает значение единицы, если ее переменные имеют противоположные значения. Ниже будет показано, что функции y_8 и y_9 являются основой для построения сумматоров, так как они соответствуют правилам формирования цифр двоичных чисел при сложении (вычитании).

Из перечисленных функций двух переменных можно строить сколь угодно сложные зависимости, отражающие алгоритмы преобразования информации, представленной в двоичной системе счисления. Алгебра логики устанавливает правила формирования логически полного базиса простейших функций, из которых могут строиться любые более сложные.

Наиболее привычным базисом является набор трех функций:

{инверсия — ⌐ ,дизъюнкция — **v**, конъюнкция — **Λ** или **&**}.

Работа с функциями, представленными в этом базисе, очень похожа на использование операций обычной алгебры.

Алгебра логики устанавливает, что существуют и другие комбинации простейших логических функций, обладающих свойством логической полноты. Например, наборы логических функций {**инверсия, дизъюнкция**} и {**инверсия, конъюнкция**} также являются логически полными. Наиболее интересны минимальные базисы, включающие по одной операции {“отрицание дизъюнкции (¬∨)”} и {“отрицание конъюнкции (¬**Λ**)”}. Однако работа с функциями, представленными в указанных базисах, требует от специалистов по проектированию ЭВМ определенных навыков.

6.2. Законы алгебры логики

Из определения вышеприведенных функций можно установить целый ряд простейших свойств:

$$x \vee 1 = 1 \qquad x \cdot 0 = 0$$

$$x \vee \overline{x} = 1 \qquad x \cdot 1 = x \qquad x \vee x \vee \ldots \vee x = x$$

$$x \vee 0 = x \qquad x \cdot \overline{x} = 0 \qquad x \cdot x \cdot \ldots \cdot = x$$

$$x \vee x = x \qquad x \cdot x = x$$

В алгебре логики установлен целый ряд законов, с помощью которых возможно преобразование логических функций (ЛФ):

- ***коммутативный*** (переместительный)

$x_1 \& x_2 = x_2 \& x_1$

$x_1 \vee x_2 = x_2 \vee x_1$;

- ***ассоциативный*** (сочетательный)

$(x_1 \& x_2) \& x_3 = (x_1 \& x_3) \& x_2 = x_1 \& (x_2 \& x_3)$

$\left(x_1 \vee x_2\right) \vee x_3 = x_1 \vee \left(x_2 \vee x_3\right)$.

Эти законы полностью идентичны законам обычной алгебры;

- ***дистрибутивный*** (распределительный)

$$x_1 \cdot (x_2 \vee x_3) = x_1 \cdot x_2 \vee x_1 \cdot x_3$$

$$x_1 \vee x_2 \cdot x_3 = (x_1 \vee x_2)(x_1 \vee x_3).$$

- ***Закон поглощения.*** В дизъюнктивной форме ЛФ конъюнкция меньшего ранга, т.е. с меньшим числом переменных, поглощает все конъюнкции большего ранга, если ее изображение содержится в них. Это же справедливо и для конъюнктивных форм:

$$x_1 \vee x_1 \cdot x_2 = x_1$$

$$x_1 \cdot (x_1 \vee x_2) = x_1.$$

- ***Закон склеивания***

$$x_1 x_2 \vee x_1 \overline{x}_2 = x_1 \quad (x_1 \vee x_2)(x_1 \vee \overline{x}_2) = x_1,$$

$$Fx \vee F\overline{x} = F \qquad (x \vee F)(\overline{x} \vee F) = F,$$

где F — логическая функция общего вида, не зависящая от переменной x.

- ***Закон свёртки***

$$x \vee \overline{x}F = x \vee F \quad x(\overline{x} \vee F) = xF$$

- ***Правило де Моргана***

$$\overline{x_1 \vee x_2} = \overline{x}_1 \cdot \overline{x}_2 \quad \overline{x_1 \cdot x_2} = \overline{x}_1 \vee \overline{x}_2.$$

Убедиться в тождественности приведенных зависимостей можно путем аналитических преобразований выражений или путем построения таблицы истинности для логических функций, расположенных в левой и правой частях.

Используя данные зависимости, можно преобразовывать исходные выражения в более простые (минимизировать их), а далее по упрощенным выражениям построить техническое устройство, имеющее минимальные аппаратурные затраты.

6.3. Понятие о минимизации логических функций

Проблема минимизации логических функций решается на основе применения законов склеивания и поглощения с последующим перебором получаемых дизъюнктивных форм и выбором из них оптимальной (минимальной длины). Существует большое количество методов минимизации ЛФ. Все они отличаются друг от друга спецификой применения операций склеивания и поглощения, а также различными способами сокращения переборов. Среди аналитических методов наиболее известным является метод Квайна-Маккласки, среди табличных — метод с применением диаграмм Вейча. Графические методы минимизации отличаются большей наглядностью и меньшей трудоемкостью. Однако их применение эффективно при малом числе переменных n<5.

Рассмотрим последовательность действий минимизации ЛФ на примере.

Пример 1. Найти минимальную дизъюнктивную форму функции, заданной таблицей истинности, представленной в таблице 5.3.

Таблица 6.3

Таблица истинности функции $Y=f(X_1,X_2,X_3)$

X_1	X_2	X_3	Y
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Эта функция имеет несколько минимальных форм. По данным таблицы запишем аналитическое выражение:

$$y = \bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3$$

Пунктирными линиями в этом выражении отмечены пары конъюнкций, к которым можно применить операцию склеивания типа $FX \vee F\overline{X} = F$. Особенно это видно при использовании диаграммы Вейча, в которой “склеиваемые” конъюнкции находятся по соседству друг с другом. Диаграмма Вейча просто по-другому интерпретирует таблицу истинности (табл. 6.4).

Таблица 6.4.

Диаграмма Вейча функции Y

	x_2		$\bar{x}_2$	
x_1	$x_1 x_2 \bar{x}_3$	$x_1 x_2 x_3$ *	$x_1 \bar{x}_2 x_3$ *	$x_1 \bar{x}_2 \bar{x}_3$ *
$\bar{x}_1$	$\bar{x}_1 x_2 \bar{x}_3$ *	$\bar{x}_1 x_2 x_3$ *	$\bar{x}_1 \bar{x}_2 x_3$	$\bar{x}_1 \bar{x}_2 \bar{x}_3$ *
	$\bar{x}_3$	x_3		$\bar{x}_3$

После выделения конъюнкций (отмечены звездочкой), видно, какие конъюнкции могут образовывать пары для склеивания.

В результате применения операций склеивания и поглощения можно получить другое аналитическое выражение:

$$y = \bar{x}_1 x_2 \vee x_2 x_3 \vee x_1 x_3 \vee x_1 \bar{x}_2 \vee \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_3,$$

в котором отсутствуют возможности дальнейших склеиваний и поглощений. Однако последнее выражение является избыточным, так как отдельные конъюнкции могут быть “лишними”, т.е. их “составные части” могут включаться в другие конъюнкции. У данной функции существует пять безызбыточных дизъюнктивных форм, из которых только две являются минимальными:

$$y_1 = \overline{x}_1 x_2 \vee x_1 x_3 \vee \overline{x}_2 \overline{x}_3;$$

$$y_2 = \overline{x}_1 x_2 \vee x_2 x_3 \vee x_1 \overline{x}_2 \vee \overline{x}_1 \overline{x}_3;$$

$$y_3 = \overline{x}_1 x_2 \vee x_1 x_3 \vee x_1 \overline{x}_2 \vee \overline{x}_1 \overline{x}_3;$$

$$y_4 = \overline{x}_1 \overline{x}_3 \vee x_2 x_3 \vee x_1 \overline{x}_2;$$

$$y_5 = \overline{x}_1 \overline{x}_3 \vee \overline{x}_1 x_2 \vee x_1 x_3 \vee x_1 \overline{x}_2$$

Из приведенных зависимостей видно, что только функции $\boldsymbol{y_1}$ и $\boldsymbol{y_4}$ являются минимальными формами функций, так как они содержат наименьшее число конъюнкций и имеют минимальный ранг этих конъюнкций.

Минимизация “вручную” возможна только для функций, зависящих от 4—5 переменных, так как трудоемкость переборов растет в квадратичной зависимости от числа переменных. Применение мощных ЭВМ для этих целей позволяет расширить границы до n=12—15. Если при этом учесть, что функции могут быть частично определены (значения функций на некоторых наборах переменных можно определять произвольно), а также что иногда приходится решать задачи совместной минимизации систем ЛФ, то минимизация ЛФ становится сложной инженерной, практической и научной проблемой.

6.4.Техническая интерпретация логических функций

По логическим выражениям проектируются схемы ЭВМ. При этом следует придерживаться следующей последовательности действий.

1. Словесное описание работы схемы.
2. Формализация словесного описания (выделение входных/выходных данных, запись функции таблицей истинности).
3. Запись функций в дизъюнктивной (конъюнктивной) совершенной нормальной форме по таблицам истинности.
4. Минимизация логических зависимостей с целью их упрощения.
5. Представление полученных выражений в выбранном логически полном базисе элементарных функций.
6. Построение схемы устройства.
7. Проверка работоспособности полученной схемы.

Покажем взаимосвязь перечисленных этапов на примере.

Пример 2. Спроектировать схему, фиксирующую появление “неправильной” тетрады в двоично-десятичном представлении чисел.

1. Каждая тетрада двоично-десятичного представления числа содержит десятичные цифры 0 — 9, что соответствует двоичным числам 0000 — 1001. Значения тетрады, соответствующие двоичным числам 1010 — 1111 (шестнадцатеричные цифры A — F), не должны появляться при представлении десятичных чисел.

2. Составим таблицу истинности функции (табл. 6.5), которая принимает значения, равные единице, при появлении “неправильных” тетрад. Разряды тетрады обозначим переменными *x*, *y*, *z*, *u*.

Таблица 6.5

Таблица истинности функции *F*

x	y	z	u	F	
0	0	0	0	0	*Разрешенные комбинации*
0	0	0	1	0	
0	0	1	0	0	
0	0	1	1	0	
0	1	0	0	0	
0	1	0	1	0	
0	1	1	0	0	
0	1	1	1	0	
1	0	0	0	0	
1	0	0	1	0	
1	0	1	0	1	*Неправильные тетради*
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	1	
1	1	1	0	1	
1	1	1	1	1	

3. Исходная совершенная дизъюнктивная нормальная форма записывается

$$F = x\bar{y}z\bar{u} \vee x\bar{y}zu \vee xy\bar{z}\bar{u} \vee xy\bar{z}u \vee xyz\bar{u} \vee xyzu.$$

4. Эта форма функции допускает упрощение, что видно по диаграмме Вейча (табл. 6.6). Этот же результат можно получить аналитически.

Таблица 6.6

Диаграмма Вейча для функции *F*

	y		$\bar{y}$		
x	$xyz\bar{u}$ •	$xy\bar{z}u$ •			$\bar{z}$
	$xyz\bar{u}$ •	$xyzu$ •	$x\bar{y}zu$ •	$x\bar{y}z\bar{u}$ •	
$\bar{x}$					z
					z
	$\bar{u}$	u		$\bar{u}$	

5. Минимальная форма функции *F* в логически полном базисе {&, v, ⌐ } будет иметь вид:

$$F = xy \vee xz = x(y \vee z).$$

Для представления этой же схемы в другом полном базисе, например {⌐,&}, воспользуемся правилом де Моргана:

$$F = xy \vee xz = \overline{\overline{xy \vee xz}} = \overline{\overline{xy \cdot xz}}.$$

6. По полученным зависимостям можно построить схемы фиксации “неправильных” тетрад (рис. 6.2).

7. Проверить работоспособность построенных схем можно путем задания различных комбинаций переменных ***x, y, z,*** и ***u*** определения реакции на выходе схемы *F*.

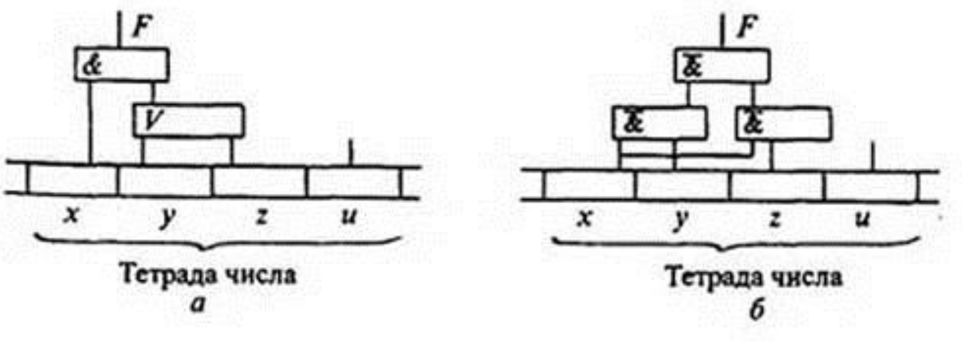
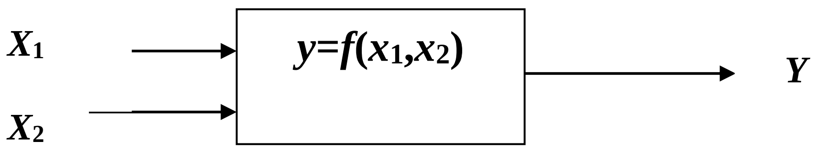


Рис.6.2. Схема фиксации «неправильных» тетрад:
а — схема в базисе (⌐, &, V), *б* — схема в базисе (⌐&).

Пример 3. Спроектировать схему устройства для обработки результатов решения студентами задач №1 и №2. При этом результатом считается решение одной из задач (минимальный результат) или решение двух задач.

1. Обозначим функцию решения $y=f(x_1;x_2)$ и представим структуру разрабатываемого устройства:



2. Составим таблицу истинности для поставленной задачи.

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	1

3. Запишем аналитическое выражение для функции *Y* в совершенной дизъюнктивной нормальной форме (СДНФ) по условию задачи:

$$Y = \overline{x_1} \wedge x_2 \vee x_1 \wedge \overline{x_2} \vee x_1 \wedge x_2$$

4. Выполним минимизацию, используя закон склеивания для 1-ой и 3-ей конъюнкций и 2-ой и 3-ей конъюнкций. В результате получим:

$$Y = x_2 \vee x_1$$

Полученная функция минимальна и реализуется в виде простой схемы «ИЛИ» (логического сложения).