ЦИФРОВЫЕ УСТРОЙСТВА

Появление импульсных устройств создало материальную базу для разработки цифровых измерительных приборов, систем передачи цифровой информации, ЭВМ. Вся эта техника осуществляет операции над цифровыми сигналами. Такие сигналы принимают лишь два значения "0" или "1". Их называют состояниями. Число состояний *m* = 2. Физически состояния задаются определенным уровнем напряжения, например "0" – напряжением , "1" – напряжением .

Сообщениями часто служат цифры. Совокупность цифр образуют алфавит *L*. Количество цифр от 0 до 9 определяют объем алфавита, т. е. *L* = 10. Передать десять цифр двумя состояниями нельзя. Поэтому каждой цифре ставят в соответствие не один, а несколько импульсов – *n*.

Совокупность из *n* импульсов называют кодовой комбинацией. Импульсы в кодовой комбинации называют разрядами. Число разрядов – *n* называют длиной кодовой комбинации. Так как каждый разряд может принимать одно из двух состояний, то совокупность из *n* разрядов позволяет создать  различных кодовых комбинаций. Если , то такой код может обеспечить передачу *L* цифр. Для *L* = 10 . В качестве примера можно поставить следующее соответствие цифр и кодовых комбинаций.

0 – 0000;

1 – 0001;

2 – 0010;

3 – 0011;

4 – 0100;

5 – 0101;

6 – 0110;

7 – 0111;

8 – 1000;

9 – 1001.

В приведенном примере каждой цифре соответствует четырехразрядная кодовая комбинация. Появление единицы последовательно в каждом из разрядов соответствует цифрам 8; 4; 2; 1. Эти цифры называются весами разрядов, а рассмотренный код – кодом с весом 8-4-2-1. Каждому из разрядов кода могут быть присвоены и другие веса, например 4-2-2-1 или 2-4-2-1. Цифрам могут быть поставлены в соответствие другие кодовые комбинации, например код избытком три. Принцип формирования кодовых комбинаций может быть иным. Например, если каждая кодовая комбинация отличается от соседних состоянием только одного из разрядов, то получаем код Грея:

0 – 0000;

1 – 0001;

2 – 0011;

3 – 0010;

4 – 0110;

5 – 0111;

6 – 0101;

7 – 0100;

8 – 1100;

9 – 1101.

Приведенные примеры показывают, что количество кодов велико. Наиболее широко применяется код 8-4-2-1.

Любое число десятичной системы счисления *N* можно представить двоичным кодом в виде

,

где *n* – число двоичных разрядов;

*Ki* – коэффициент, определяющий состояние *i*-го разряда: 0 или 1.

Например, число 258 в двоичной системе имеет вид:



Однако наиболее удобна двоично-десятичная система. В такой системе цифре каждого десятичного разряда соответствует кодовая комбинация кода 8-4-2-1. Например, число 258 в двоично-десятичной системе имеет вид:

0010 0101 1000.

Формирование цифровой информации может быть различным. В ЭВМ информация вводится в виде цифр. В измерительных приборах измеряемая величина преобразуется, например, в уровень напряжения, который затем преобразуется в код, определяющий результат измерения числом. В системах связи непрерывный сигнал дискретизируется по времени, каждый дискретный отсчет квантуется по уровню, а затем уровень каждого дискретного отсчета преобразуется в код. Такое преобразование выполняется аналого-цифровыми преобразователями.

1. **Основные операции и элементы алгебры логики.**

Основой построения любого устройства, использующего цифровую информацию, являются элементы двух типов: логические и запоминающие. Логические элементы выполняют простейшие логические операции над цифровыми сигналами. Запоминающие элементы служат для хранения цифровой информации (состояния разрядов кодовой комбинации).

Логическая операция состоит в преобразовании по определенным правилам входных цифровых сигналов в выходные. Математически цифровые сигналы обозначают поразрядно символами, например *x*1, *x*2, *x*3, *x*4. Их называют переменными. Каждая переменная может принимать значение "0" или "1". Результат логической операции часто обозначают *F* или *Q.* Он также может иметь значение "0" или "1". Математическим аппаратом логики является алгебра Буля. В булевой алгебре над переменными "0" или "1" могут выполняться три основных действия: логическое сложение, логическое умножение и логическое отрицание.

Логическое сложение (дизъюнкция или операция ИЛИ) записывается в виде



Правила выполнения операции ИЛИ заключаются в следующем:

|  |  |
| --- | --- |
| 0 + 0 = 0; 1 + 0 = 1;  0 + 1 = 1; 1 + 1 = 1. | (17.1) |

Логические схемы, реализующие операцию ИЛИ; называют ячейками ИЛИ. Их схемное обозначение приведено на рис. 17.1а. Простейшая реализация логической ячейки ИЛИ на диодах приведена на рис. 17.1б. Напряжение на выходе схемы будет равно *E* (*F=*1), если хотя бы на один из входов будет подан единичный сигнал.

Логическое умножение (конъюнкция или операция И) записывается в виде



Правила выполнения операции И заключаются в следующем

  (17.2)

Логические схемы, реализующие правила (17.2), называются ячейками И. Их схемное обозначение приведено на рис. 17.2а. Простейшая реализация



логической ячейки И на диодах приведена на рис. 17.2б. Напряжение на выходе  только в том случае, если все диоды будут закрыты, т. е. на всех входах будет потенциал *Е* (логическая 1). В противном случае открывшийся диод шунтирует нагрузку и .

Логическое отрицание (инверсия или операция НЕ) записывается в виде



и читается: *F* равно не *x*. Правила выполнения операции НЕ заключаются в следующем

  (17.3)

Логические схемы, реализующие правило (17.3) называются ячейками НЕ. Их графическое обозначение приведено на рис. 17.3. Операция НЕ может быть реализована схемой транзисторного ключа.

Рассмотренные логические правила и схемы позволяют реализовать сколь угодно сложную логическую функцию. Например, функция



реализуется пятью логическими элементами, в том числе два элемента И, два элемента НЕ и один элемент ИЛИ (см. рис. 17.4).

 Все логические элементы выпускаются в микросхемном исполнении. Они входят в состав всех серий цифровых микросхем и имеют следующие условные обозначения:

* элементы "ИЛИ" – ЛЛ;
* элементы "И" – ЛИ;
* элементы "НЕ" – ЛН.

Например, микросхема К555 ЛИ1 имеет в своем составе 4 элемента "И" на два входа каждый.

1. **Основные теоремы алгебры логики.**

Теоремы для одной переменной охватывают все операции над переменной *x* и константами "0" и "1":

|  |  |  |
| --- | --- | --- |
|  |  |  |

Теоремы для двух или более переменных – *x* и  *y*:

1. Переместительный закон:

 

1. Сочетательный закон:





1. Распределительный закон:





Доказательство:



Здесь к скобке применена теорема 2.

1. Закон поглощения:



Доказательство:



1. 



Доказательство:



1. Закон склеивания:



Доказательство:



1. Закон отрицания (теорема де-Морана)

|  |  |
| --- | --- |
|  |  |

1. **Булевы функции (функции логики).**

Результат выполнения логических операций над двоичными переменными называется булевой функцией *F*. Она может принимать только два значения – "0" или "1". Задать булеву функцию – значит указать ее значение при всех возможных комбинациях переменных (аргументов). Если число переменных равно "*n*", то число возможных комбинаций равно . Когда значение функции известно для всех комбинаций, она называется полностью определенной. В противном случае – частично определенной.

Булевы функции необходимы для синтеза цифровых устройств, содержащих только логические элементы. Для представления булевых функций часто применяют словесное описание, табличное и алгебраическое представление.

Словесное описание функции должно однозначно определять все случаи, в которых выходные сигналы принимают значение "1" или "0". Например: Спроектировать устройство с тремя входами *x*1, *x*2, *x*3, на выходе которого сигнал *F* = 1 в случае, если на любые два или на все три входа подан сигнал "1".

Табличное представление – это перечисление всех возможных комбинаций входных сигналов. Для устройства, заданного приведенным выше словесным описанием, таблица значений имеет вид

# Таблица 17.1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | *x*1 | *x*2 | *x*3 | F |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 1 |
| 6 | 1 | 1 | 0 | 1 |
| 7 | 1 | 1 | 1 | 1 |

Такая таблица называется таблица истинности.

Алгебраическая форма представления булевых функций используется для минимизации (упрощения формул) и для построения логических схем. Существует две формы алгебраических функций – дизъюнктивная и конъюнктивная. Дизъюнктивная нормальная форма представляет собой сумму элементарных произведений аргументов, например



Если каждое слагаемое содержит все аргументы или их отрицания, то получаем совершенную дизъюнктивную нормальную форму (СДФН), например



Совершенная конъюнктивная нормальная форма (СКНФ) представляет собой логическое произведение элементарных логических сумм, причем каждая сумма содержит все аргументы или их отрицания, например



Для перехода от таблицы истинности к СДНФ учитываются только те состояния, для которых функция равна 1. Для каждого такого состояния записывается элементарное произведение всех аргументов. Если аргумент имеет значение "0", то записывается его отрицание. Для приведенного примера СДНФ имеет вид

 (17.4)

### Для перехода от таблицы истинности к СДНФ учитываются только те состояния, для которых функция равна "0". Для каждого такого состояния записывается элементарная сумма аргументов. Если аргумент имеет значение "1", то пишется его отрицание. Для приведенного примера СКНФ имеет вид

 (17.5)

На основании полученных формул (17.4) или (17.5) можно построить логическую схему, состоящую из элементов "ИЛИ", "И", "НЕ". Для функции (17.4) сначала изображаются инверторы, затем ячейки "И" и потом ячейки "ИЛИ" (см. рис. 17.5).

Схемы рис. 17.4 и рис. 17.5 содержат все типы логических элементов. При проектировании всегда стремятся номенклатуру элементов. В связи с этим созданы логические элементы, способные выполнить простейшую функцию двух аргументов "ИЛИ-НЕ", а также "И-НЕ". С помощью каждого из этих элементов можно выразить все основные операции булевой алгебры, а значит реализовать любую логическую функцию. Покажем это.

#### Для элемента "ИЛИ-НЕ"

|  |  |  |
| --- | --- | --- |
| операция "НЕ" | операция "ИЛИ" | операция "И" |



#### Для элемента "И-НЕ"

|  |  |  |
| --- | --- | --- |
| операция "НЕ" | операция "ИЛИ" | операция "И" |



В микросхемном исполнении элементы "ИЛИ-НЕ" обозначаются индексами ЛЕ, элементы "И-НЕ" – индексами ЛА. Например, микросхема К555 ЛЕ1 имеет в своем составе четыре элемента "ИЛИ-НЕ" на два входа каждый.