

УДК 519.7

ИСПОЛЬЗОВАНИЕ АЛГЕБРЫ ЛОГИКИ В УСТРОЙСТВЕ ПРОЦЕССОРА

М.В. Кондратьев, В.В. Костенко, студенты гр. ИБс-231, III курс
Е.А. Николаева, к. ф.-м. н., доцент, зав. кафедрой математики
Кузбасский государственный технический
университет имени Т.Ф. Горбачёва, г. Кемерово

Современный процессор – это невероятно сложное устройство, которое обеспечивает работу компьютеров, мобильных телефонов, телевизоров и многой другой техники. За более чем 50 лет с момента создания первых процессоров, прогресс ушел далеко вперед и, на данный момент, разобраться в их устройстве с нуля – практически невыполнимая задача. Но, как процессор, созданный в 70-х годах прошлого века так и тот, что находится в любом современном компьютере, оба работают по одним и тем же законам, законам алгебры логики [1].

Алгебра логики – раздел математики, сформированный Джорджем Булем в середине XIX века. Он оперирует значениями «истина» (1) и «ложь» (0), что идеально соответствует двоичной системе счисления, применяемой в цифровой технике. Использование всего двух операндов, а также небольшого количества законов, Булева алгебра позволяет преобразовывать простую последовательность сигналов в сложные инструкции, по которым и работает процессор.

Основным интерпретатором алгебры логики в процессоре является транзистор. Транзистор – электронное устройство, необходимое для преобразования тока в цепи в сигналы [2]. На самом деле, процессор не воспринимает именно нули и единицы, как принято считать. Подобная формулировка служит как некое упрощение для понимания устройства его работы. В реальности, сигнал всего один, и у него есть два состояния – ток есть или же он отсутствует (1 и 0 соответственно). И именно транзисторы позволяют преобразовывать последовательность тока в цепи в сигналы, а потом в инструкции.

В дальнейшем, в этой статье будет использоваться термин «бит». Бит – это единица измерения количества информации [3]. Он, как было сказано выше имеет два состояния – 1 и 0.

Процессор выполняет обработку данных, используя три базовые логические функции булевой алгебры: инверсию, конъюнкцию и дизъюнкцию. На их основе строятся любые более сложные преобразования, запись которых выглядит следующим образом:

1. Инверсия (НЕ / NOT)

Операция отрицания меняет значение бита на противоположное.

$$f(A) = \bar{A}$$

2. Конъюнкция (И / AND)

Результат истинен только тогда, когда оба входных сигнала равны 1.

$$f(A, B) = A \cdot B$$

3. Дизъюнкция (ИЛИ / OR)

Результат истинен, если хотя бы один из входных сигналов равен 1.

$$f(A, B) = A + B$$

Из этих функций складываются логические вентили. Вентиль представляет собой логический элемент, который принимает одни двоичные значения и выдает другие в зависимости от своей реализации. Например, выполнение логического умножения (рис 1.).

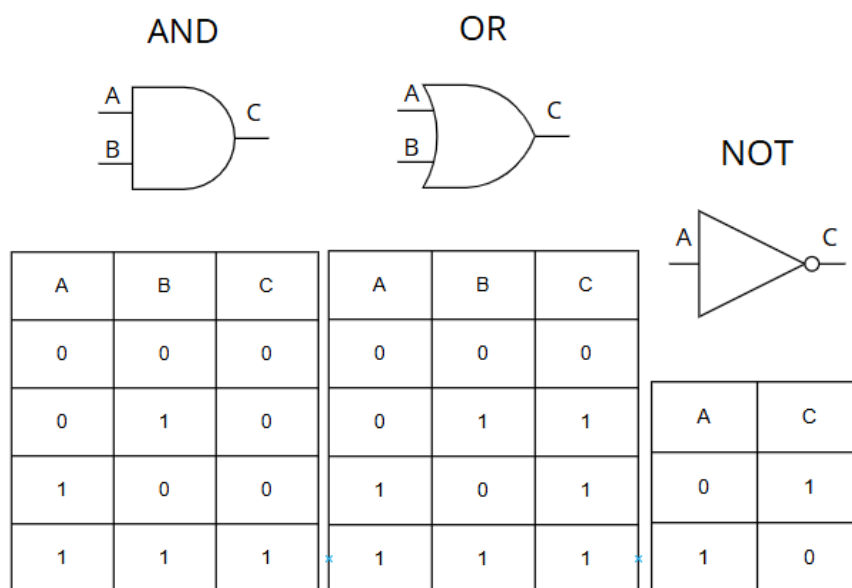


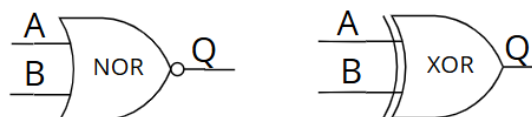
Рисунок 1. Вентили: И (AND), ИЛИ (OR) и НЕ (NOT).

Существуют комбинированные и универсальные вентили, которые выполняют более сложные операции. Примеры подобных конструкций представлены на рис 2.

На основе подобных универсальных элементов можно создать любую, сколь угодно сложную схему.

Логические вентили объединяются в функциональные блоки процессора, этот процесс называется синтезом сложных узлов центрального процессора (ЦП).

Одним из таких блоков является сумматор. Он преобразует входящие сигналы в сигнал, эквивалентный сумме этих сигналов. Иными словами, производит сложение.



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

Рисунок 2. ИЛИ-НЕ (NOR), Исключающее ИЛИ (XOR).

Операция сложения начинается с полусумматора (рис.3), который вычисляет сумму двух бит (S) и перенос в следующий разряд (C).

Логические выражения для полусумматора:

1. $S = A \oplus B$ (Сумма через XOR)
2. $C = A \cdot B$ (Перенос через AND)
- 3.

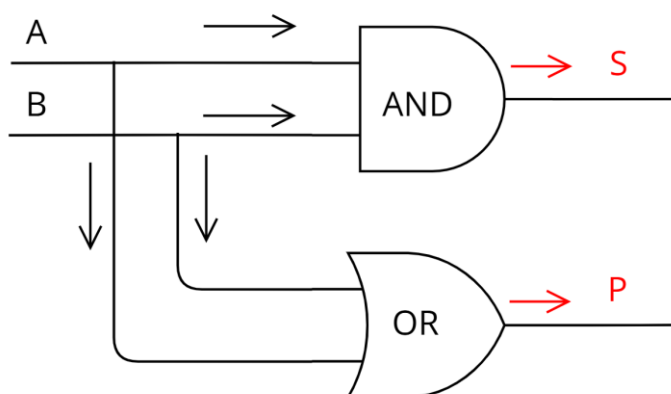


Рисунок 3. Схема полусумматора

Используется полусумматор, когда нужно сложить два бита без учета переноса с предыдущего разряда (поэтому, входных канала всего два, разряд не учитывается).

Для более сложных вычислений, учитывающих перенос разряда, используются полные сумматоры (рис 4.). В таблице 2 представлена работа полного сумматора.

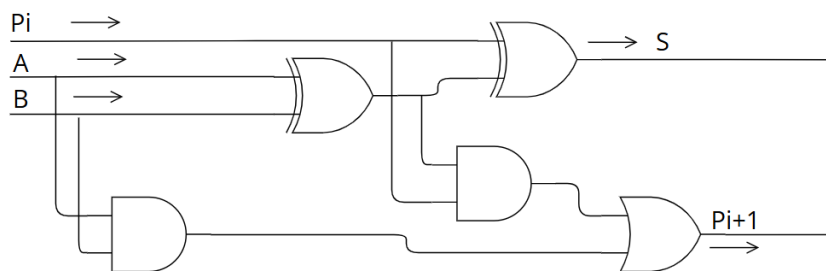


Рисунок 4. Схема сумматора
В Таблице 1. Представлена работа полного сумматора.

A	B	P	Сумма(S)	Перенос(P)
0	0	0	0	0
1	1	0	0	1
1	1	1	1	1

Таблица 1.

Важнейшим функциональным блоком является арифметико-логическое устройство (АЛУ).

АЛУ – ядро процессора, его «мозг». Оно выполняет как математические операции (сложение, вычитание), так и логические. В современных процессорах это огромная сеть из тысяч логических вентилях, управляемых сигналами выбора операции (рис. 5).

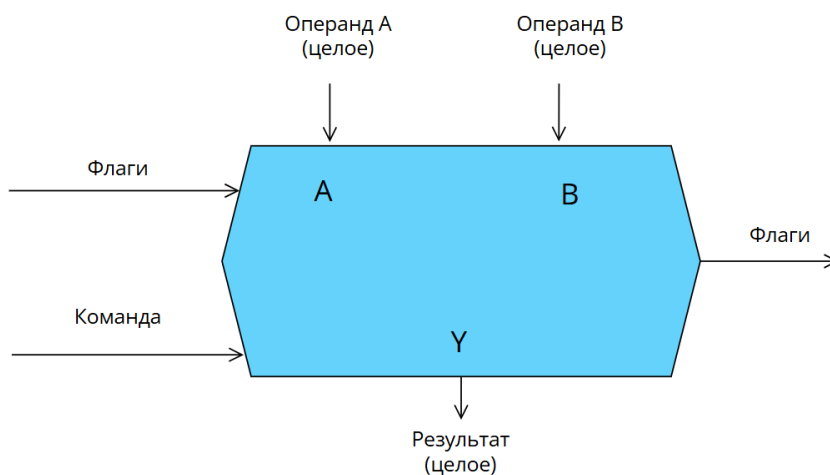


Рисунок 5. Упрощенная схема АЛУ

Современный процессор содержит огромное количество логических вентилях и функциональных блоков, а, следовательно, и транзисторов. Поэтому, необходимо минимизировать их количество для оптимизации работы устройства.

Для этого инженеры используют законы Буля, упрощающие логические выражения:

1. Закон поглощения: $A + (A \cdot B) = A$
2. Закон де Моргана: $\overline{A \cdot B} = \bar{A} + \bar{B}$
3. Закон склеивания: $(A \cdot B) + (A \cdot \bar{B}) = A$

Примененные подобных законов позволяет добиться снижения энергопотребления и количества транзисторов, что позволяет повысить итоговую мощность процессора и понизить его расходы электроэнергии.

Рассмотрим практическое применение законов алгебры логики на примере полного сумматора.

Исходные данные:

- A, B — складываемые биты,
- C — бит переноса с предыдущего разряда,
- S — бит суммы,
- P — бит переноса в следующий разряд

$$S = A \oplus B \oplus C$$

$$P = (A \cdot B) + (C \cdot (A \oplus B))$$

Реализация схемы без оптимизации:

Основываясь на схеме полного сумматора (рис. 4), нам понадобятся следующие вентили:

- XOR — 8 транзисторов, 2 вентиль,
- AND — 4 транзистора, 2 вентиль,
- OR — 4 транзистора, 1 вентиль

Таким образом, общее количество транзисторов:

$$8 \cdot 2 + 4 \cdot 2 + 4 \cdot 1 = 28$$

Без использования законов Буля для реализации схемы полного сумматора требуется 28 транзисторов.

Реализация схемы с использованием основных законов Буля:

Основная формула для подсчета P (бит переноса в следующий разряд) содержит в себе сложение по модулю 2 ($\oplus$). Его можно выразить через базовые операции:

$$A \oplus B = A \cdot \bar{B} + \bar{A} \cdot B$$

Подставим полученное выражение в исходную формулу для подсчета P :

$$P = (A \cdot B) + (C \cdot (A \oplus B)),$$

$$P = (A \cdot B) + (C \cdot (A \cdot \bar{B} + \bar{A} \cdot B)),$$

$$P = A \cdot B + C \cdot A \cdot \bar{B} + C \cdot \bar{A} \cdot B$$

Основываясь на выражении $A = A \cdot (B + \bar{B})$, преобразуем нашу новую формулу по расчету P :

$$A \cdot B = A \cdot B \cdot (C + \bar{C}) = A \cdot B \cdot C + A \cdot B \cdot \bar{C},$$

Подставим это выражение в исходное:

$$A \cdot B \cdot C + A \cdot B \cdot \bar{C} + C \cdot A \cdot \bar{B} + C \cdot \bar{A} \cdot B,$$

Воспользуемся группировкой:

$$A \cdot B \cdot C + C \cdot A \cdot \bar{B} = A \cdot C,$$

$$A \cdot B \cdot C + C \cdot \bar{A} \cdot B = B \cdot C,$$

Таким образом, полученное после всех преобразований выражение имеет вид:

$$P = A \cdot B + A \cdot C + B \cdot C$$

Это выражение для оптимизированной схемы полного сумматора, оно содержит 2 вентили OR и 3 вентили AND, общее количество транзисторов:

$$4 \cdot 3 + 4 \cdot 2 = 20,$$

Благодаря примененным законам Буля мы смогли сделать схему эффективнее на 8 транзисторов. Эта прибавка не кажется такой большой, но в масштабах миллиардов транзисторов в одном процессоре такой прирост невероятно высок.

Алгебра логики является фундаментом для развития цифровых технологий. Без применения ее законов было бы невозможно создание даже самой простой микросхемы. И хоть инженеры идут по пути усложнения логических связей, принципы преобразования сигналов остаются такими же, какими они были 50 лет назад. Поэтому, такой раздел математики, как Булева алгебра является необходимым для понимания логических операций, обязательных для создания эффективного ПО.

Список литературы

1. Новиков Ф. А. Дискретная математика для программистов: Учебник для вузов. — СПб.: Питер, 2009. — 384 с. URL: <https://stugum.wordpress.com/wp-content/uploads/2014/03/novikov/>
2. Таненбаум Э., Уодхолл А. Архитектура компьютера. — СПб.: Питер, 2013. — 816 с. URL: <https://www.gstu.by/sites/default/files/files/resources/2015/01/tanenbaum/>
3. Харрис Д. М., Харрис С. Л. Цифровая схемотехника и архитектура компьютера. — Morgan Kaufmann, 2013. URL: https://is.ifmo.ru/books/2016/digital-design-and-computer-architecture-russian-translation_July16_2016.pdf/