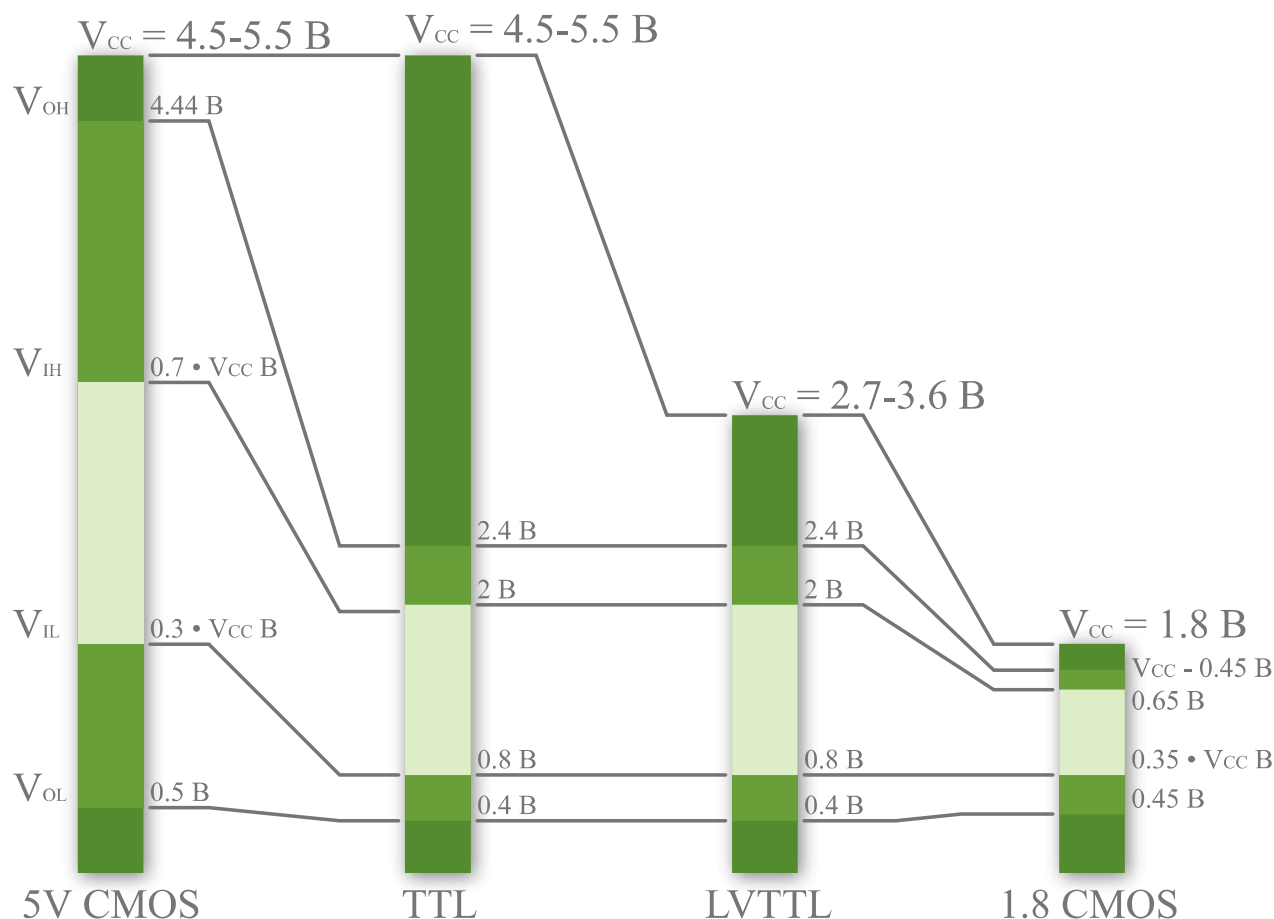


## Логические уровни и операции

Вы наверняка уже слышали о таком понятии, как напряжение (англ. voltage), которое измеряется в ~~Алессандре~~ Вольтах? Напряжение может быть разным: 220 В, 110 В, 5 В, 3 В и даже 3,3 В. Однако в цифровой технике нам не нужен такой большой разброс значений. Если объяснять совсем просто: когда мы подадим на светодиод, скажем, 5 вольт (так называемая «логическая единица»), он начнет светиться. Если же мы подадим 0 вольт (так называемый «логический ноль»), то светодиод гореть не будет. У чисел «1» и «0» есть и альтернативные названия в зависимости от области их применения: «истина» (англ. true) или «ложь» (англ. false), «высокое состояние» (англ. high) или «низкое состояние» (англ. low).

Логический сигнал — это сигнал, принимающий два возможных значения — «истина» или «ложь», «высокое» или «низкое» состояние, «0» или «1». Конечно, мы живем не в идеальном мире, и получить ровно 5 вольт мы не можем, как и не можем получить чистый ноль (мы лишь «договорились», что он является нулем относительно тех 5 вольт). Что же делать в таком случае? Абстрактные «0» и «1» сильно бы упростили нам жизнь. Уже из определения логического сигнала можно догадаться, что есть понятие «логический уровень». Существуют стандарты, определяющие допустимые уровни напряжений, логических сигналов. В зависимости от технологии исполнения (КМОП, ТТЛ) эти уровни не всегда совпадают. STM32 питается от напряжения 3,3 В, т.е. «высокий уровень» для нашего МК — 3,3 В. «Низкий уровень», соответственно — это 0 В. Следует обратить внимание на то, что в стандартах в качестве логических уровней приведены не одиночные значения напряжений, а диапазоны значений. А между низким и высоким уровнями есть буферная зона. Она нужна для того, чтобы избежать «дребезга состояний», который может возникнуть, если значение сигнала будет колебаться около граничного значения. Ниже приведены некоторые стандарты напряжений.



Микроконтроллер – это сложное устройство, он содержит множество триггеров, которые включают или отключают тот или иной участок цепи, отвечающий за определенную функциональность. Вся информация, которую понимает микроконтроллер, — это те самые нули и единицы, т. е. «высокое» напряжение или «низкое». Программа, которую мы будем писать, представляет собой последовательность нулей и единиц. Не более. О некоторых важных понятиях мы поговорим позже, а сейчас лишь отметим, что нам придется записывать нули и единицы в определенные участки памяти, тем самым включая и отключая части внутренних схем микроконтроллера. Для этого нам понадобятся логические операции.

Алгебра логики, или булева алгебра, оперирует логическими переменными и включает три базовых логических операции:

- логическое «И» (англ. and), конъюнкция или логическое умножение, обозначается  $\wedge$  или в языке Си `&` ;
- логическое «ИЛИ» (англ. or), дизъюнкция или логическое сложение, обозначается  $\vee$  или в языке Си `|` ;
- логическое «НЕ» (англ. not), изменение значения, инверсия или отрицание, обозначается  $\neg$  или в языке Си `~` .

Таблицы истинности для этих операций:

A	B	A & B	A   B	$\neg A$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Ниже приведены аксиомы булевой алгебры.

1. Коммутативность (переместительный закон)

$$A \& B = B \& A$$

$$A | B = B | A$$

2. Ассоциативность (сочетательный закон)

$$(A \& B) \& C = A \& (B \& C)$$

$$(A | B) | C = A | (B | C)$$

3. Дистрибутивность (распределительный закон)

$$(A | B) \& C = (A \& C) | (B \& C)$$

$$(A \& B) | C = (A | C) \& (B | C)$$

4. Свойства констант

$$A \& 0 = 0$$

$$A | 0 = A$$

$$A \& 1 = A$$

$$A | 1 = 1$$

$$A | A = A$$

5. Идемпотентность

$$A \& A = A$$

$$A | (\neg A) = 1$$

6. Комплементарность (дополнительный закон)

$$A \& (\neg A) = 0$$

7. Закон снятия двойного отрицания

$$\sim(\sim A) = A$$

## 8. Теорема де Моргана

$$\sim(A \mid B) = (\sim A) \& (\sim B)$$

$$\sim(A \& B) = (\sim A) \mid (\sim B)$$

В программировании существуют два вида логических операций. Когда оперируют числами (набором бит), то осуществляют побитовую логическую операцию, которая применяется независимо к каждому биту. В языке программирования Си логический оператор «И», как уже говорилось раньше, вызывают символом «&». Например, если `A = 0101`, а `B = 1100`, то

<b>A</b>	<b>0101</b>
B	1100
A & B	0100

Другой вид логических операций — с участием булевых значений. В Си булева переменная является «ложью», если равна 0, и «истиной», если равна любому ненулевому числу (стандартная величина для «истины», очевидно, 1). В Си булевый оператор «И» вызывают символом `&&`. Таким образом:

<b>A</b>	<b>1100</b>
B	0100
A && B	1

Нам потребуются операции первого типа (над набором битов) для настройки микроконтроллера, но об этом позже.

---

[Назад](#) | [Оглавление](#) | [Дальше](#)