Transistor operation

Ramil Aleskerov

(Dated: November 18, 2019)

This paper explores the history and the structure of transistors and their uses in circuits. The first part focuses on the design and underlying physics of both Bipolar Junction (BJT) and Field Effective (FET) transistor types. We then show how the introduction of transistors revolutionized the world of electronics by making the creation of logic gates not only possible but relatively cheap. We will demonstrate transistor circuits for several logic gates (AND, OR, NAND, NOR) thus underlying their importance, and also show how Boolean algebra can be used as a useful tool that helps to understand and design transistor circuits.

INTRODUCTION

Transistors

The history of transistors goes back to the early 20th century. During that time, vacuum tubes were used to amplify radio signal used in telephony. The device however was not only fragile but also consumed large amounts of power. Most importantly, it was unable to amplify high frequencies.

In 1947, John Bardeen, Walter Brattain and William Shockley began experimenting with silicon. They discovered that it was comprised of a region favoring positive current flow and one region favoring negative current flow, thus discovering the P-N junction. They later speculated that by adding a third electrode to the semiconductor, they could not only amplify but also control the amount of current through the silicon. While additional technical difficulties arose for silicon, they eventually achieved an amplification factor of 330 for germanium, which was then used to build the first point-contact transistor[1].

Logic gates

The history of logic gate goes back to 1924, when Walther Bothe has invented the "AND" gate in order to measure coincidences for electrons in Compton scattering. He used two point discharge counters and then recorded the coincidences on a moving photographic film with a time resolution of 1 millisecond [2]. It was a first example of an "AND" coincidence circuit, for which he shared a Nobel prize.

From 1934-1937, Akira Nakashima and Claude E. Shannon showed how switching circuits could be used to replicate Boolean algebra[3]. These methods are now widely used in MOS logic, which uses MOSFET type transistors.

THEORY

A transistor is a semiconducting device used to amplify an input signal, producing an output signal larger than the input signal by some factor, β. In this lab we are going to examine the two major types of transistors: Bipolar junction transistors (BJT) and Field effective transistors (FET). BJT transistors control the current with a current, while FET transistors control the current with a voltage.

BJT transistors are of two types: pnp and npn. Focusing on the npn type, it has three terminals: collector, C, base, B and emitter, E as shown in Fig. 1.

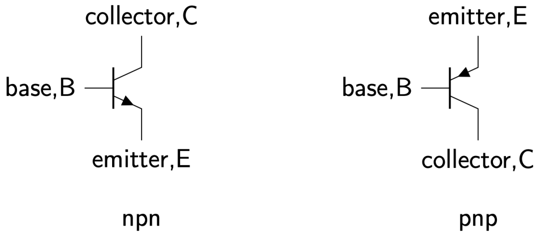

FIG. 1. Schematic symbols for npn and pnp transistors.

The structure of the npn transistor can be better seen from Fig. 2.

When the voltage, $V_B > V_E$ is applied, emitter and base act as a diode with current I_B. Therefore electrons of the n++ region would move to and fill the holes in the p region. Because the p-doped part of the transistor is narrow, most of these electrons will then move to the n+ region due to the positive potential V_C which forces electrons from n+ region to move. Thus we get a current passing through the transistor. We define the amplification constant of the transistor as the ratio $\beta = I_C/I_B$, which, generally speaking, is an unreliable parameter since it varies with currents and applied voltages. We can also notice that $V_B \approx V_E + 0.6V$ due to the pn++ region acting as a diode [4, p. 31].

The second important type of transistors are Field ef-

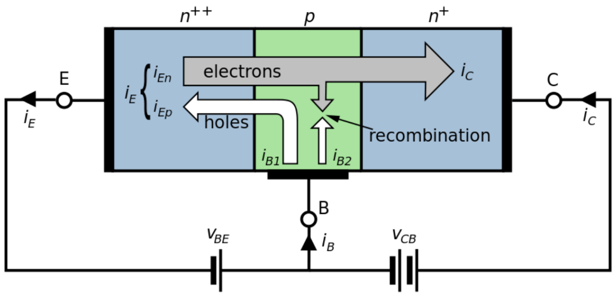

FIG. 2. Diagram breaking down voltage drops and currents over the npn transistor. When $V_{BE} = V_B - V_E > 0$, a current cause the holes of the p region shift to n++, and due to the small size of p region electrons can now freely move from emitter to collector.

fect transistors (FET). Each FET can be either n or p channel, be in a depletion or enhancement mode, and be either of Metal-Oxid-Semiconductor (MOSFET) or Junction FET (JFET) type (only 5 of 8 combinations are possible). The difference in channels (n or p) or modes (depletion or enhancement) just reverses the signs in equations or change the operation point and thus do not produce interesting physics. However, MOSFET and JFET do differentiate in their structure.

Let us consider JFETs. Unlike BJT, a FET controls the passing current with the voltage, or electric field more specifically. By applying a small voltage at the gate, we are able to manipulate the depletion region as to make in either narrower or wider and thus control the current (Fig. 3). One side effect of such structure is the fact that JFET can only be produced in a depletion mode, meaning that a zero input voltage corresponds to the ON mode.

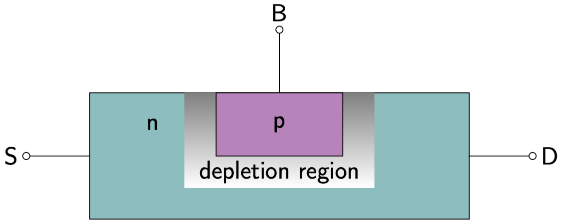

FIG. 3. Diagram showing the internal structure of JFET transistor. As the voltage applied to the base (sometimes labeled as gate instead) increases, so does the depletion region, which leaves less region in the n-region for the current to pass through cause the decrease in the current (and vice versa).

MOSFET works on a similar principle (Fig. 4). When $V_{GS} > 0$, the holes close to gate are repelled. When $V_{GS} > V_{TH}$ the voltage V_{DS} can now drive the current. The current voltage dependence is approximately linear for small V_{DS} and then is flat, saturated for larger V_{DS}. Unlike JFET, one advantage of MOSFET is that due to the layer of Metal oxide, a MOSFET's impedance ($10^{14}\Omega$) is much higher than JFET's ($10^{9}\Omega$) [4, Chapter 3]. Further, it can be produced in both modes, which is why it's generally more popular.

TABLE I. AND gate Boolean logic representation

input A	input B	output
0	0	0
1	0	0
0	1	0
1	1	1

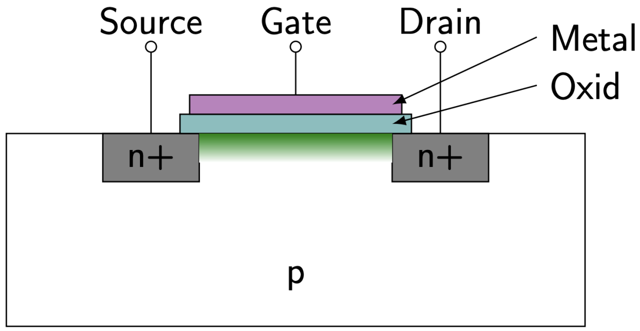

FIG. 4. Internal structure of MOSFET transistor.

APPLICATION

Now that we are familiar with the general structure of a transistor, we will show how they can be used to build logic gates. One of the key reasons why logic can now be easily implemented is because transistor can be thought of as an active circuit component that allows current to pass only given the proper base signal.

In order to organize the further analysis, we will use Boolean logic tables. A Boolean logic table is simply a way to iterate through all possible digital inputs and correspond them with the wanted binary outputs.

A circuit to study is the AND gate circuit. We know that since transistors allow current only given the proper base signal, putting two transistors in series will pass current only if both inputs meet the threshold (Fig. 5).

We can write the corresponding truth table as in Table I. In our case it's pretty simple since the output should be non-zero only if both inputs are non-zero. This is why we denote such operation as multiplication, $Output = AB$.

The next simplest logical circuit is an OR gate. We want the current to pass if at least one of the gates is

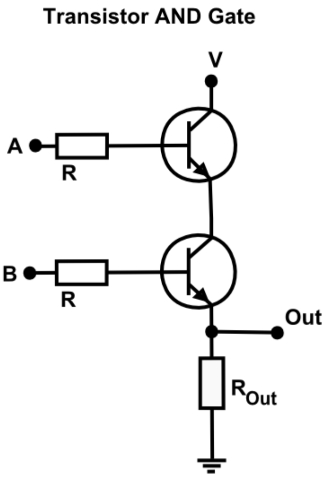

FIG. 5. Diagram showing the circuit representation of an AND gate. A and B are the input signals passed into the base of a transistor. The current will pass through both if and only if both channels have a proper base signal (ex. exceed threshold voltage).

true (the boolean term for 1). Therefore it makes sense to place them in parallel as shown in Fig. 6, with truth values shown in Table II.

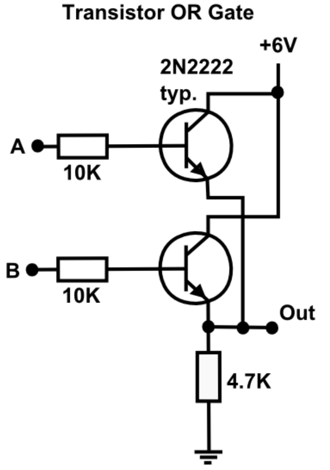

FIG. 6. Diagram showing the circuit representation of a OR gate. A and B are the input signals passed into the base of a transistor. The current will pass through both if and only if at least one of the inputs is at (or above) threshold voltage

TABLE II. OR gate Boolean logic representation

input A	input B	output
0	0	0
1	0	1
0	1	1
1	1	1

Following the analogous procedure other logical gates were designed (NAND, NOR, XOR) and frequently the design was in large guided by manipulations of some of the simpler gates.

An interesting question one could ask if it's possible to construct all possible logic gates just out of one. In turns out it is possible, and in fact there are two such commonly used gates: NOR and NAND (the negation of OR/AND) [5, p. 500].

In order to prove this theorem (we will prove for NAND since the proof for NOR is analogous) we will use another commonly used algebraic notations for logic gates, which are derived from their logic tables shown on Table III.

TABLE III. Logic gates - Algebraic representation

Gate	Algebraic representation B
OR	$A + B$
AND	$A \times B$
Negation	$\overline{A}$
NAND	$\overline{A \times B}$
XOR	$A \oplus B$
NOR	$\overline{A + B}$

Finally, we will use Morgan's Theorem to simplify some operations, which states that

$$\overline{A \times B} = \overline{A} + \overline{B} \tag{1}$$

$$\overline{A + B} = \overline{A} \times \overline{B} \tag{2}$$

First thing to show is the negation, which is quiet simple since

$$NAND(A, A) = \overline{A \times A} = \overline{A}$$

Therefore negation is "NAND-complete". Therefore AND which is a negation of NAND is also "NAND-complete".

In order to construct OR (and therefore NOR) we will use first Morgan's relation for two negated input channels, $\bar{A}, \bar{B}$.

$$\overline{\bar{A} \times \bar{B}} = \bar{\bar{A}} + \bar{\bar{B}} = A + B$$

The most difficult is the construction of XOR (exclusive OR). One can refer to Table IV for the Boolean truth-table.

TABLE IV. XOR gate Boolean logic representation

input A	input B	output
0	0	0
1	0	1
0	1	1
1	1	0

After looking at the table, a brute force way to approach XOR is to represent it using other gates:

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

Since all of the used operations can be represented solely with NAND, the XOR itself can be represented with NAND, making NAND universal.

CONCLUSION

In this paper we discussed the historical development of transistors, described the structures of both FET and BJT type transistors, and then showed how their invention allowed for an efficient implementation of Boolean logic, which now is in the foundation of any modern computer.

[1] "November 17 - December 23, 1947: Invention of the First Transistor," `https://www.aps.org/publications/apsnews/200011/history.cfm`.

[2] L. Bonolis, American Journal of Physics **79**, 1133–1150 (2011).

[3] R. S. Stanković, J. T. Astola, C. Moraga, and S. Stanković, Computer Aided Systems Theory - EUROCAST 2013 Lecture Notes in Computer Science , 258–265 (2013).

[4] P. Horowitz and W. Hill, *The Art of Electronics* (Cambridge University Press, 2015).

[5] J. Bird, *Engineering mathematics* (Routledge, 2010).