

Adder developed and constructed using only switches and connections

Abstract. *Subject* For several decades, digital electronics has been one of the fastest-growing branches of science and technology. During this time, the number of parts in each chip has grown exponentially and has recently passed the 1 trillion transistor milestone. With the purpose of understanding the principles of building complex systems, it is essential to study the mathematical foundations of computer engineering. The subject of the development was the adder for binary addition. *Objectives* The purpose of the work was to develop and construct an adder using only switches and connections to make the principles of building complex computers understandable even for schoolchildren. *Methodology* The Boolean algebra mathematical apparatus has been implemented for hardware representation of the adder on switches and light bulbs. A bitwise operation of exclusive OR function has been applied in order to get a circuit of the alternate switch for an arbitrary number of switches. For the synthesis of the adder single-bit circuit, the equivalent function has been implemented. *Results* As a result of the work, the adder working model using only switches and connections has been developed and constructed. This model can add numbers from 0 to 1111_b (15_d). It was successfully demonstrated at a school conference, which confirmed that the usage of this model is available even to schoolchildren. *Conclusion* The adder based only on switches and connections promotes the understanding of performance principles of modern computer engineering and can be useful during classes on computing science in the study of the binary numeral system.

Keywords: adder, Boolean algebra, equivalent function, exclusive OR, keys, switch.

Introduction

Computer engineering widely uses adders. In particular, the adder is the main element of the arithmetic and logic unit of the processor and is part of the random-access memory of the computer. As is commonly known, the adder converts information signals into a single one, equivalent to the sum of these signals id est adds up the input signals. In order to understand the principles of building complex systems, it is essential to study the mathematical foundations of computer engineering. One of the devices that demonstrate these foundations is a parallel binary adder described in the monograph [1] and the Wikipedia article [2]. There are ready-made adder implementations with a single chip, for example, a 4-bit binary full adder SN7483 (K155IM3) demonstrated in the description [3]. You can easily find the circuits where the adder is implemented with the help of elementary logic chips or with the relay as in descriptions [4, 5]. However, there are no adder circuits at all on the Internet, assembled on switches and connections without other elements. The purpose of this work was to develop and construct an adder using only switches and connections so that the principles of computer design can be understood even by the schoolchildren, who have the most superficial knowledge in electrical engineering.

Materials and methods

Parallel adder

In terms of mathematics, a parallel adder is a discrete device without memory, according to monograph [1]. It means that after the input signaling, the output signals are immediately generated. In the case of the adder, the input signals are the bits of the first term and addend, and the output signals are the sum result. The principle of the adder operation is very similar to the columnar addition. The functional flow diagram of the 4-bit parallel adder is presented in Figure 1.

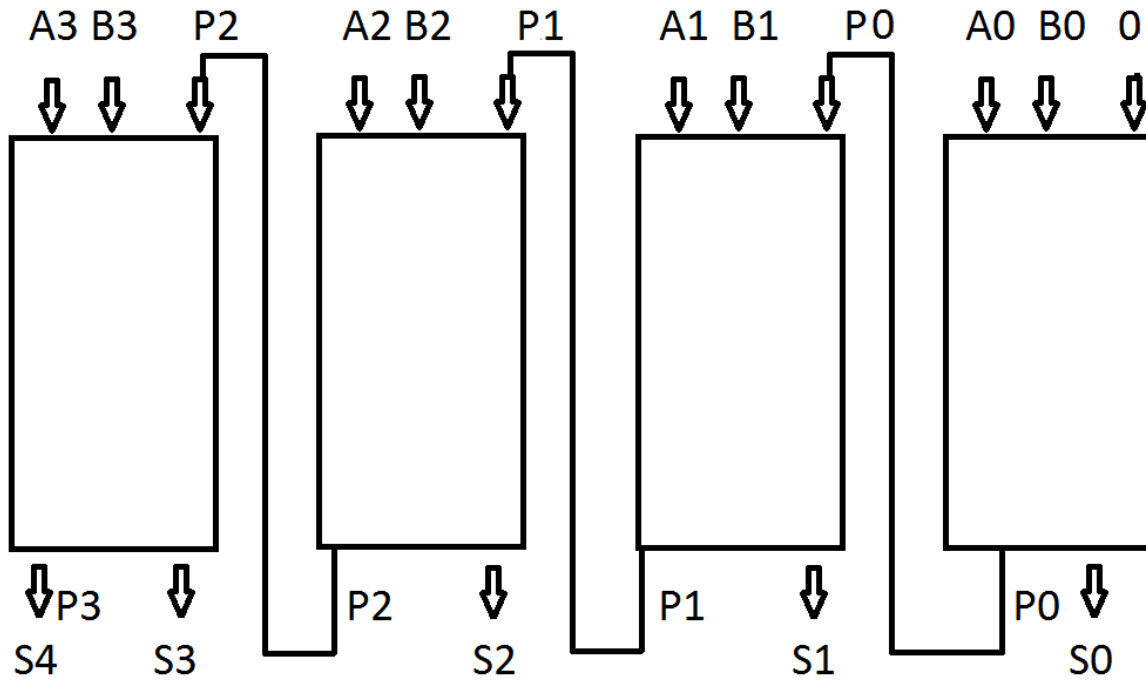


Fig. 1: Functional flow diagram of the 4-bit parallel adder

where $A_0...A_3$ are bits of the first term, $B_0...B_3$ are bits of the addend, and $P_1...P_3$ are the carry into 1...3 bits.

In this case, the carry in the least significant bit is equal to 0, and the carry from the most significant bit is the most significant bit of the sum. Each bit has three inputs and two outputs. The corresponding bits of summands and the carry of the previous bit are fed to the input. The output is the values of the sum and the carry. The truth table (Table 1) for one bit is presented below.

Table 1: Truth table for one bit

<i>Input</i>			<i>Output</i>		
<i>a</i>	<i>b</i>	<i>p</i>	<i>s</i>	<i>P</i>	\bar{P}
0	0	0	0	0	1
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	0	1	0

1	0	0	1	0	1
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	1	1	0

From the truth table it is easy to see (even without the disjunctive normal form DNF) that

$$s = a \oplus b \oplus p \quad (1)$$

Let us write the PDNF (principal disjunctive normal form) for P and simplify it:

$$P = \bar{a}bp \vee a\bar{b}p \vee ab\bar{p} \vee abp = \bar{a}bp \vee a\bar{b}p \vee ab = \bar{a}bp \vee ap \vee ab = bp \vee ap \vee ab \quad (2)$$

Due to the peculiarities of our hardware implementation, we need another function \bar{P} , which is described by the following expression:

$$\bar{P} = \bar{a}\bar{b}\bar{p} \vee \bar{a}b\bar{p} \vee a\bar{b}\bar{p} \vee ab\bar{p} = \bar{a}\bar{b} \vee \bar{a}b\bar{p} \vee a\bar{b}\bar{p} = \bar{b}\bar{p} \vee \bar{a}\bar{p} \vee \bar{a}\bar{b} \quad (3)$$

Hardware implementation of Boolean algebra on switches and light bulbs

For the task implementation, we will use double-pole switches for inputs (variables) – keys. One position will be marked "0", another – "1". As an output, we shall use an incandescent light bulb. The switched-on incandescent light bulb means "1" and the switched-off incandescent bulb means "0". The circuit for the implementation of constants 0, 1 and one-place functions A and \bar{A} is presented in Figure 2.

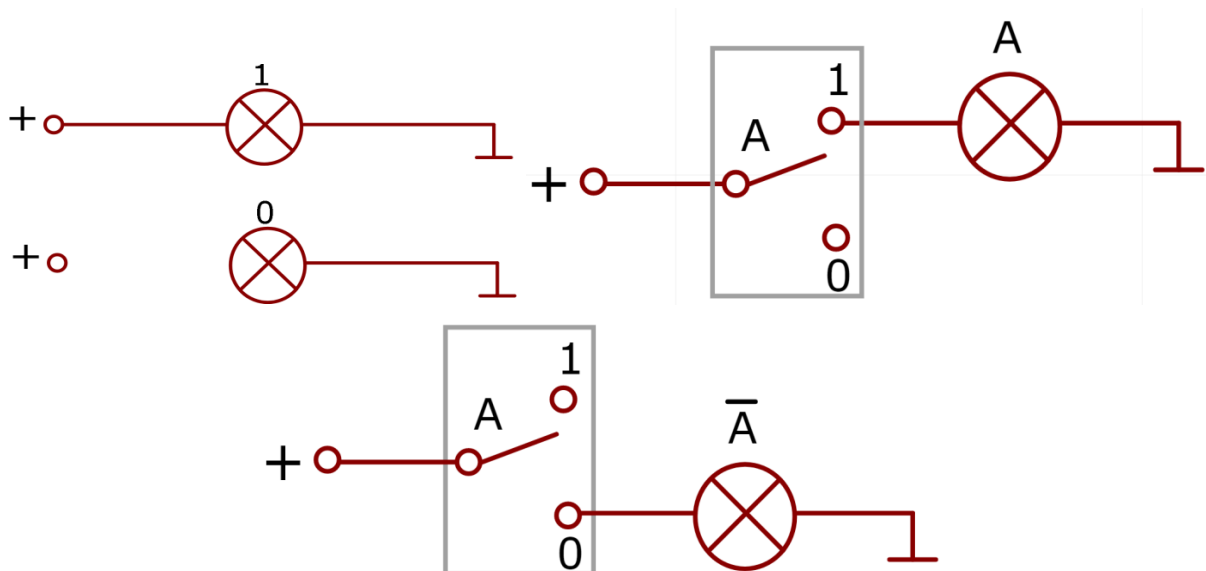


Figure 2: Circuit for implementation of constants 0, 1 and one-place functions A and \bar{A}

Figure 3 presents the implementation of some binary functions.

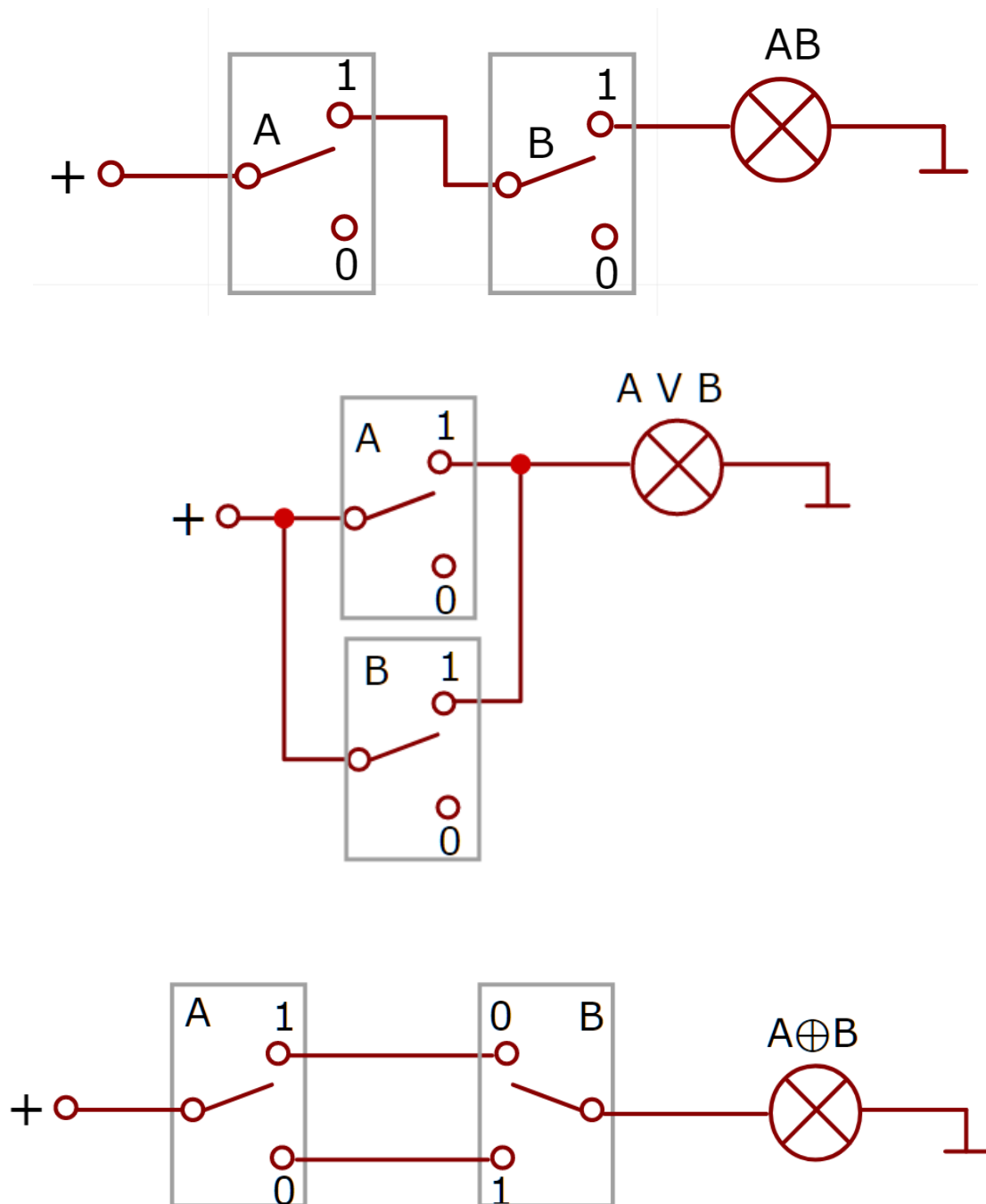


Figure 3: Implementation of some binary functions

In more complicated cases, we will need double-pole and even multi-pole switches. It means that we control two or more contact groups using one switch.

Alternate switches (Traveler system)

The implementation of the exclusive-OR function has a wide practical application. The matter is that this is a well-known circuit of the alternate switch. A group of switches controlling one lamp is referred to as the alternate switch so that the change of state of each switch leads to a change of state of the lamp, which corresponds to the logic of "exclusive OR". This allows the lamp to be switched off or switched on by any switch in the group, as described in the Wikipedia article [6].

The state transition is described by the following expression:

$$\overline{a \oplus b} = \bar{a} \oplus b = a \oplus \bar{b} \quad (4)$$

That is, when changing (switching) any variable (switch), the result (lamp status) changes to the opposite one.

Let us generalize the expression (4) to an arbitrary number of variables (switches):

$$\overline{x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n} = \bar{x}_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_n = x_1 \oplus \bar{x}_2 \oplus x_3 \oplus \dots \oplus x_n = x_1 \oplus x_2 \oplus \bar{x}_3 \oplus \dots \oplus x_n = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus \bar{x}_n \quad (5)$$

Let us denote $X_i = x_1 \oplus x_2 \oplus x_3 \oplus \dots \oplus x_i$

Then, X_n is the desired result.

Let us implement $X_i = X_{i-1} \oplus x_i = X_{i-1}\bar{x}_i \vee \overline{X_{i-1}}x_i \quad (6)$

The implementation circuit using the double-pole switch is presented in Figure 4.

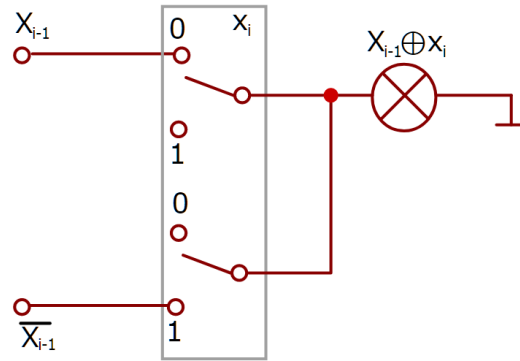


Figure 4: Implementation circuit using the double-pole switch

It is easy to see that the circuit can be simplified by combining the keys. A simplified circuit of the adder implementation with a double-pole switch is presented in Figure 5.

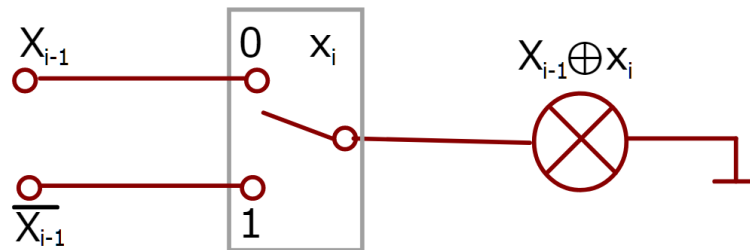
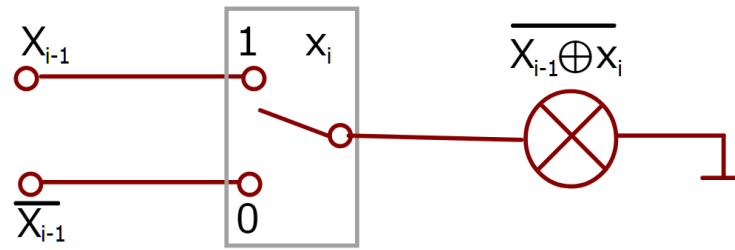


Figure 5: Simplified circuit of adder implementation with a double-pole switch

It is significant that at every step (except the last one), we need both the result of the previous step and its negation. Let us implement $\bar{X}_i = X_{i-1} \oplus \bar{x}_i$ (Figure 6).

Figure 6: Implementation of $\overline{X_i} = X_{i-1} \oplus \overline{x_i}$

Let us combine two last circuits. The combined circuit is presented in Figure 7.

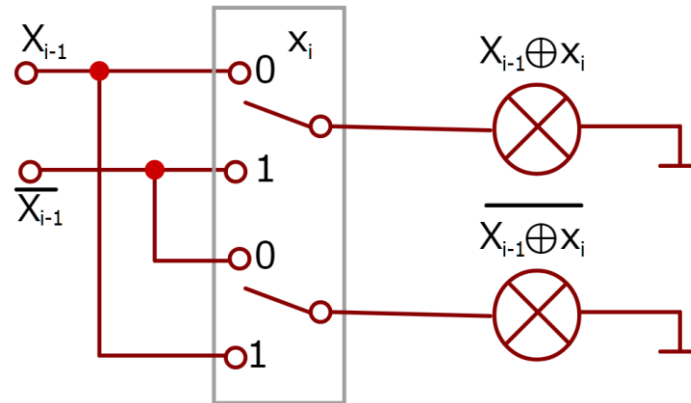


Figure 7: Combined circuit

Results and discussion

From the abovementioned approach, it becomes clear how you can obtain a circuit of the alternate switch for an arbitrary number of switches. For example, let us present the circuit for six switches (Figure 8). The first and last switches are single pole because for the first switch, negation is obtained on the second contact, and for the last switch, negation is not necessary.

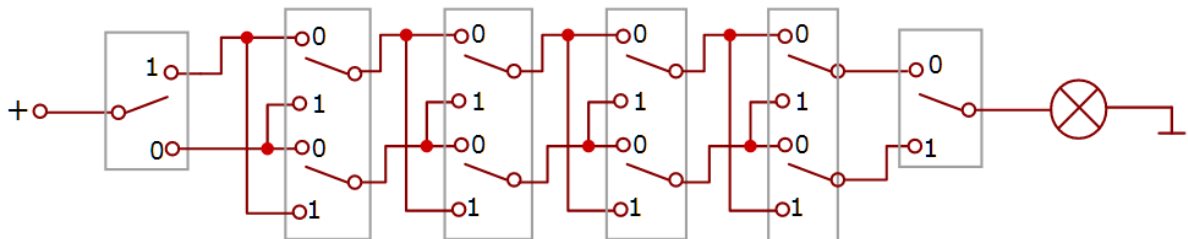


Figure 8: Circuit of the alternate switch for six switches

Circuit synthesis of adder's one bit

For each bit of the summand, 6-pole rotary switch SR25P-1-6-2 was used. The formula for the sum value is as follows:

$$s = p \oplus a \oplus b \quad (7)$$

The circuit for the sum bit is presented in Figure 9.

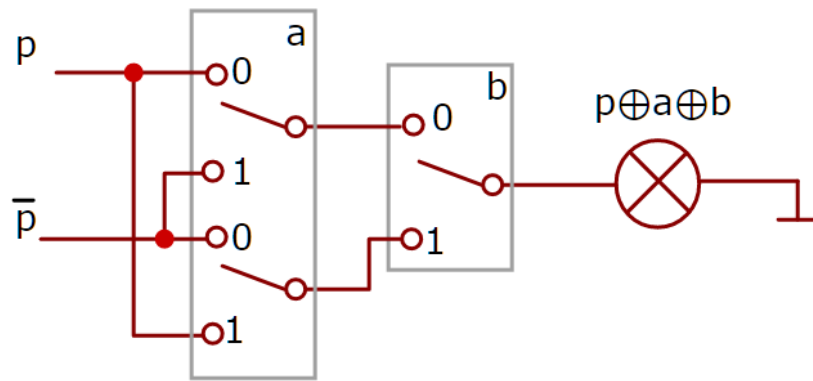


Figure 9: Circuit for the sum bit

The carry-over function is expressed by the next formula:

$$P = ab \vee pa \vee pb \quad (8)$$

The circuit for the implementation of such function by the brute-force approach is presented in Figure 10.

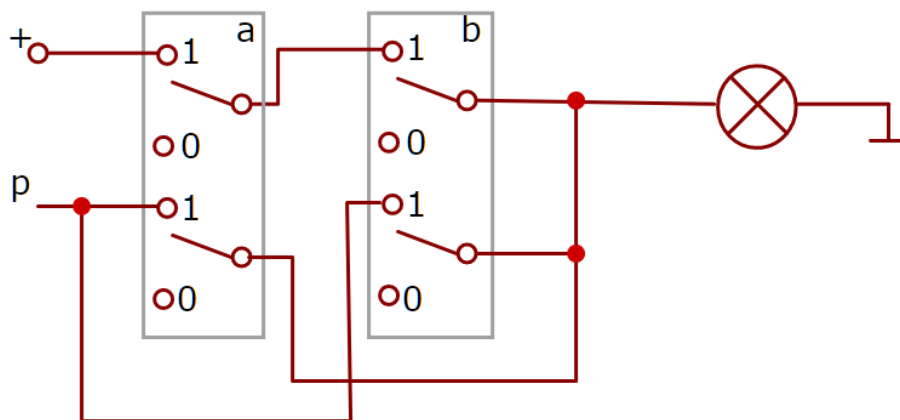


Figure 10: Circuit for carry-over function implementation by the brute-force approach

However, here are disadvantages in our circuit technique because the output can affect the input. It is easy to see that in the case of $a=1$ and $b=1$, the current bit carry will be equal to 1 (even if it must be equal to 0). At first glance, the problem could be easily solved with the diode application, but it turned out that this problem cannot be solved in such a way. The output is an implementation of the equivalent function, which appears as follows:

$$P = \bar{a}bp \vee a\bar{b}p \vee ab \quad (9)$$

The circuit for the equivalent function implementation is presented in Figure 11.

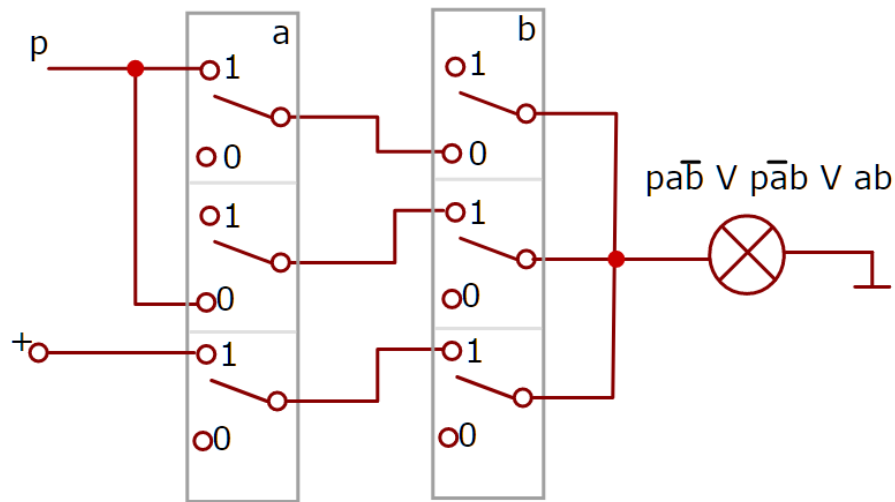


Figure 11: Circuit for the equivalent function implementation

Due to unused contacts, this circuit can be simplified. The simplified circuit of equivalent function implementation is presented in Figure 12.

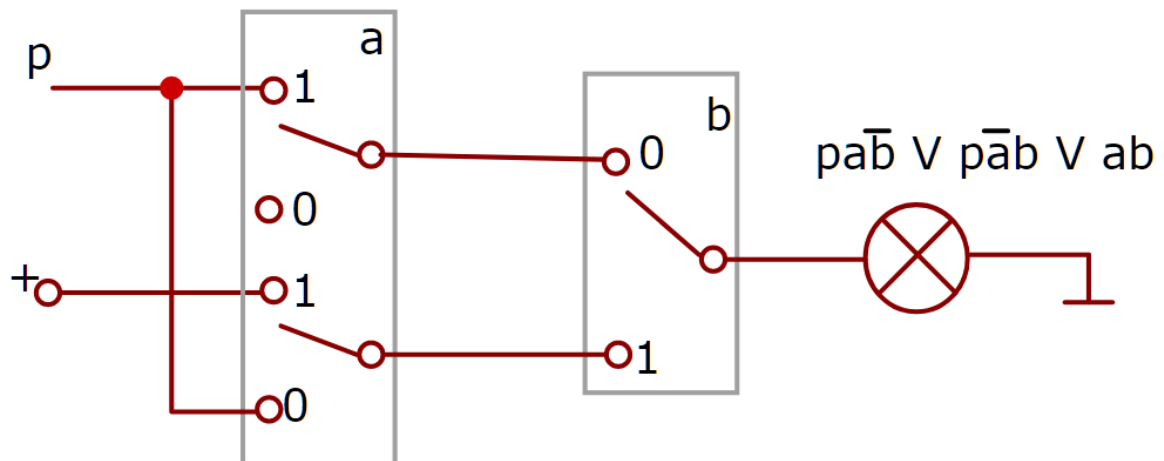


Figure 12: Simplified circuit of equivalent function implementation

The function \bar{P} , whose expression is presented in formula (3), is implemented in a similar manner:

$$\bar{P} = \bar{a}\bar{b} \vee \bar{a}b\bar{p} \vee a\bar{b}\bar{p} \quad (10)$$

Figure 12 demonstrates the circuit for the implementation of the function \bar{P} .

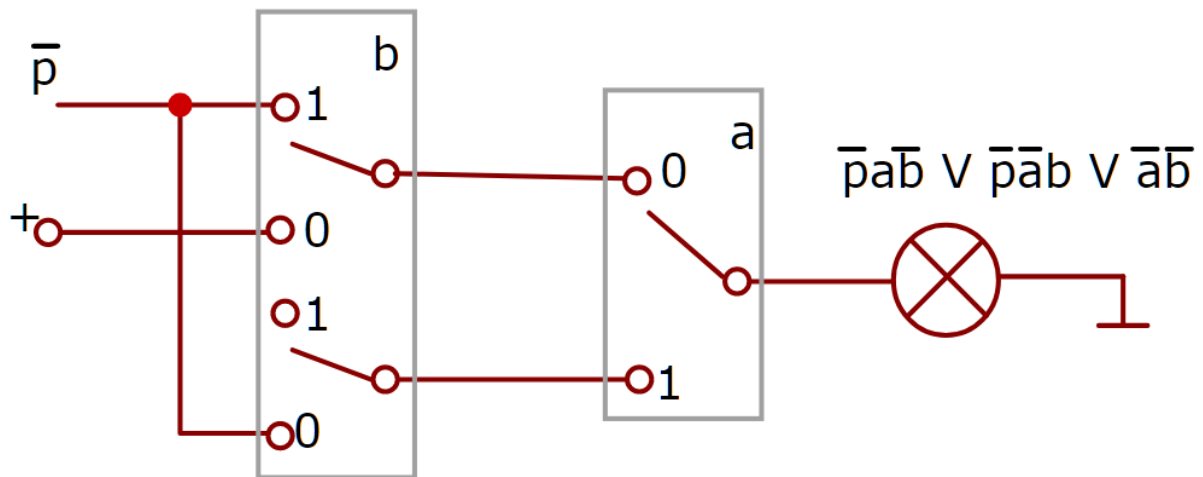


Figure 12: Circuit for the implementation of function \bar{P}

For uniform consumption of keys in the implementation of the function \bar{P} , let us use two keys from b . Some keys (6 in total) have remained unimplemented, and it has been decided to use them for the input indication. Figure 13 presents the resulting circuit of the adder's one bit.

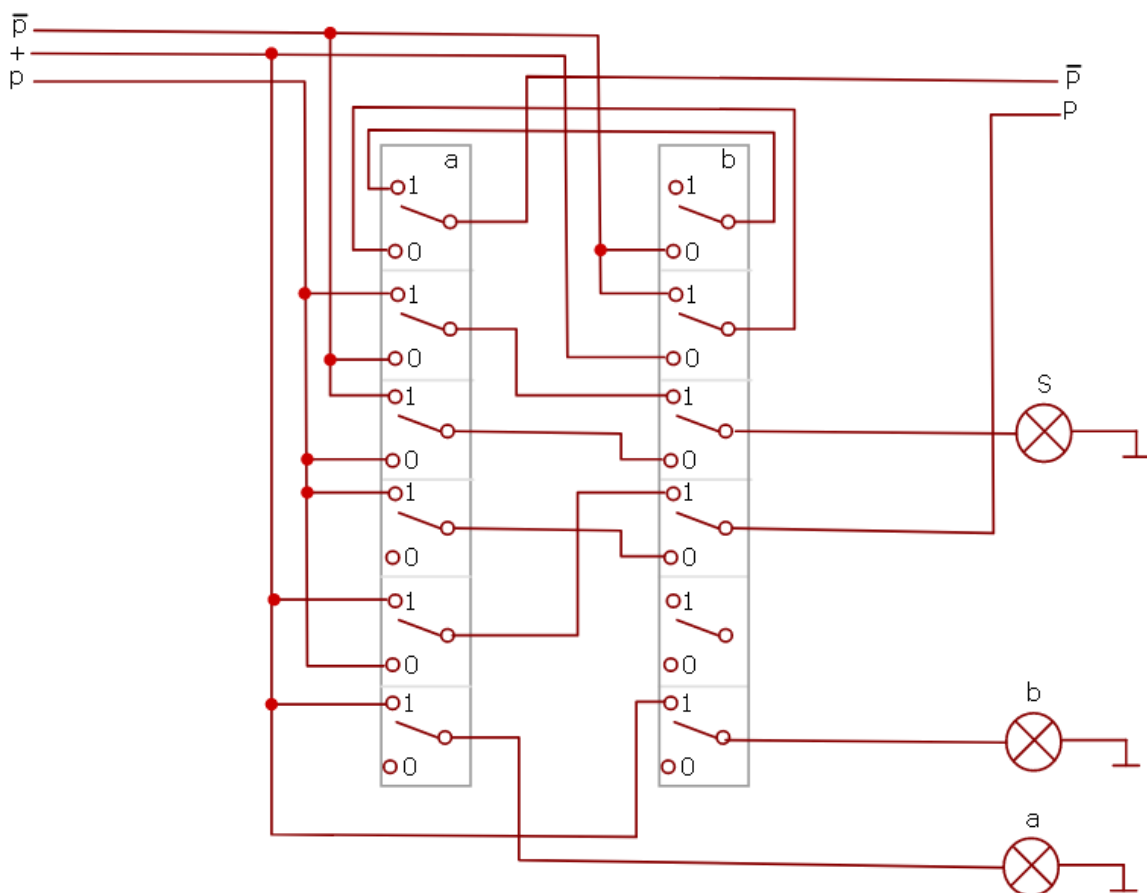


Figure 13: Resulting circuit of adder's one bit

For demonstration purposes, the working model of the adder has been assembled. The model is presented in Figure 14. It can add numbers from 0 to 1111_b (15_d). It is

interesting that the maximum reading value 11111_b (31_d) at the output cannot be obtained, which may be the subject of further research. The operation of the adder's working model was demonstrated at a conference for schoolchildren. As can be seen from the demonstration on video [8], even a schoolchild can use this adder.

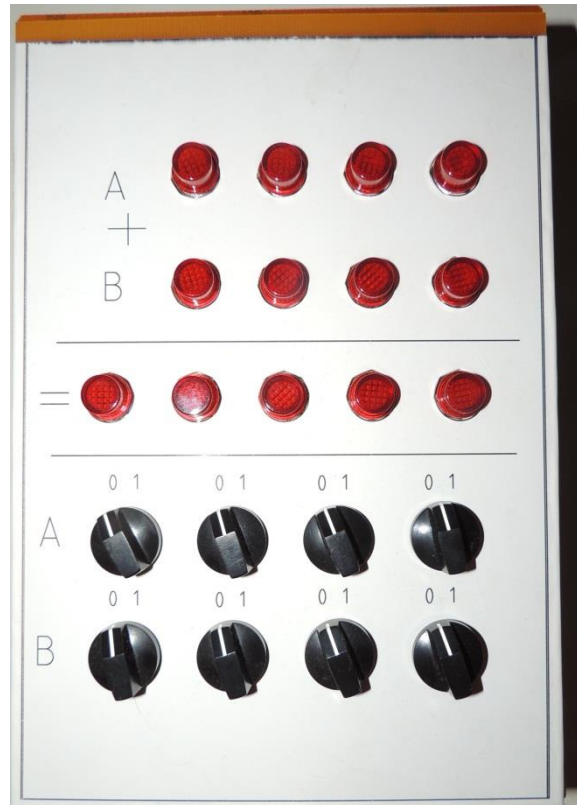


Figure 14: Working model of the adder

Conclusion

Unlike other types, the developed adder is constructed using switches and connections only. It makes the fundamentals of modern computer engineering more understandable and can be useful in classes on the basics of information and computer science at schools when studying the binary numeral system.

References

1. Sholomov L.A. Fundamentals of discrete logical and computing devices. 3rd revised edition. St. Petersburg: Lan' Publishing House; 2011, 432 p.
2. Adder. Available from: [https://en.wikipedia.org/wiki/Adder_\(electronics\)](https://en.wikipedia.org/wiki/Adder_(electronics))
3. 4-bit binary full adder with fast carry. Available from: <https://mil.ufl.edu/4712/docs/sn74ls283rev5.pdf>

4. Binary Adder. Available from: https://www.electronics-tutorials.ws/combinational/comb_7.html
5. Relay Full Adder Circuit. Available from: <http://www.andrewkingsolver.com/relay-full-adder-circuit/>
6. Multiway switching. Available from: https://en.wikipedia.org/wiki/Multiway_switching
7. Rotary Switch. Available from: https://en.wikipedia.org/wiki/Rotary_switch
8. Adder (binary numeral system). Video. Available from: <https://www.youtube.com/watch?reload=9&v=jJeoUozxwok>