

Implementation of a Two's Complement Circuit*

*Note: Sub-titles are not captured in Xplore and should not be used

Nan Lin
SPEIT
Shanghai Jiao Tong University
Shanghai, China
lins_brandon@sjtu.edu.cn

Yanxu Meng
SPEIT
Shanghai Jiao Tong University
Shanghai, China
email address or ORCID

Abstract—This document is a model and instructions for L^AT_EX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

Index Terms—component, formatting, style, styling, insert

I. QUESTION 1-5

A. Question 1

The two's complement of a binary number is obtained by inverting its digits and adding one to the least significant bit. This method allows for straightforward binary arithmetic and simplifies the design of digital circuits for arithmetic operations.

A half-adder is a fundamental component in digital arithmetic. It takes two single-bit binary inputs and produces a sum and carry output. The truth table for a half-adder is shown below:

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

TABLE I

TRUTH TABLE FOR A HALF-ADDER

The corresponding half-adder circuit diagram is illustrated in Figure 1.

To create an 8-bit two's complement circuit, we combine multiple half-adders. The process involves inverting each bit of the input number and then adding one using a series of adders. The detailed circuit diagram is shown in Figure 2.

B. Question 2

To analyze the depth and complexity of the two's complement circuit for 8 bits, we divide the process into two stages.

The first stage is the **inversion stage**. Each bit requires a NOT gate, thus:

- Depth: 1 (because all the NOT gates operate simultaneously)
- Complexity: 7 (NOT gates)

Identify applicable funding agency here. If none, delete this.

Component: Half Adder

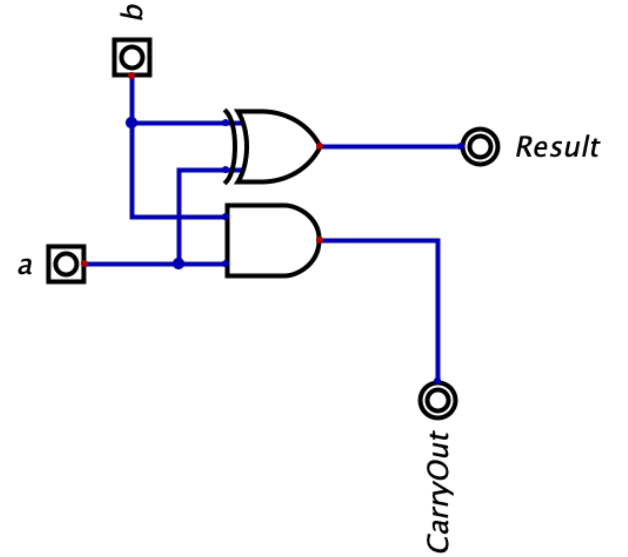


Fig. 1. Half-Adder Circuit

The second stage is the **addition stage**. For each bit, a half-adder is applied. The half-adder of the highest bit besides the sign bit (here, it is c_6) requires all the preceding adders to compute the result to operate. Therefore, the depth is 7 and the number of gates is $2 \times (8 - 1) = 14$.

In conclusion, for an 8-bit two's complement, the depth is $1 + 7 = 8$ and the complexity is $8 + 14 = 22$.

We expand the results to any 2^p -bit ($p \in \mathbb{N}$) two's complement circuit.

- 1) In the **inversion stage**, the depth is 1 due to parallelization, and the complexity is $2^p - 1$.
- 2) In the **addition stage**, the depth is $2^p - 1$ and the number of gates required is $2 \times (2^p - 1)$.

Conclusion: For a 2^p -bit two's complement circuit, the depth is 2^p , and the complexity is $3(2^p - 1)$.

C. Question 3

We propose a new version of the two's complement circuit for an 8-bit machine by employing a method that uses two

Question 1

Test

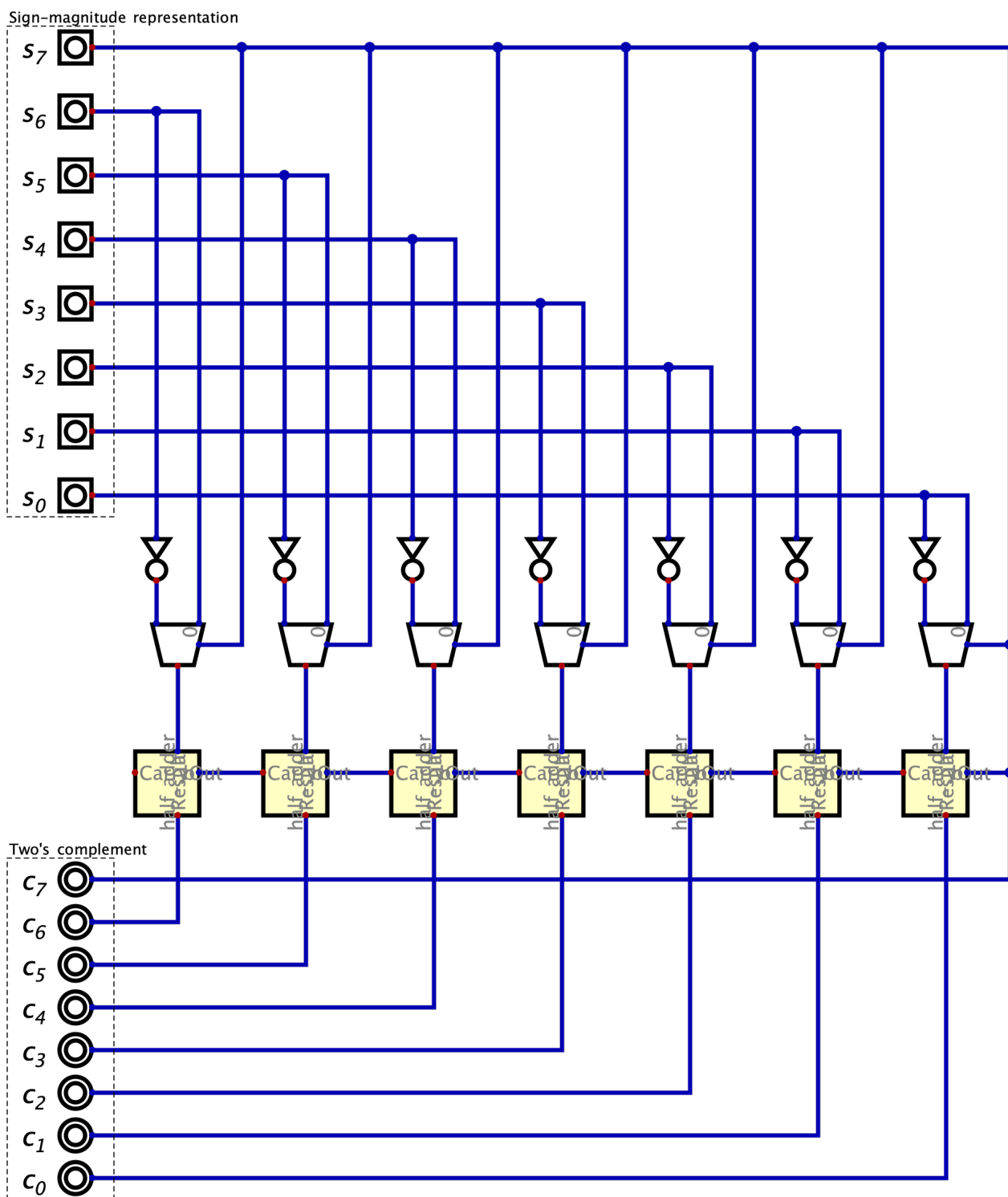


Fig. 2. 8-bit Two's Complement Circuit

4-bit lookahead carry adders in series. This approach allows for the calculation of four bits and their carry at once, then passing the carry to the next set of four bits. This reflects the divide and conquer strategy.

The implementation involves the following steps:

- 1) **Inversion Stage:** Each of the 8 bits is inverted using NOT gates.
- 2) **Addition Stage:** The 8-bit addition is divided into two 4-bit additions using lookahead carry adders.
 - **First 4-bit Adder:** Computes the result and the carry for the first four bits.
 - **Second 4-bit Adder:** Computes the result for the next four bits, including the carry from the first adder.

The 4-bit lookahead carry adder is shown in Figure 5.

In digital circuits, the Generate-Propagate (GP) logic is used to speed up the carry calculation in adders. For two binary inputs A_i and B_i , the generate (G_i) and propagate (P_i) signals are defined as:

$$G_i = A_i \cdot B_i \quad (1)$$

$$P_i = A_i + B_i \quad (2)$$

In this implementation, two components are utilized. The first one is a full adder displayed in Figure 3. The second one is the GP generator discussed above and displayed in Figure 4. The GP generator is used to generate respectively the AND and OR result of two variables.

We then combine the two 4-bit adder in series. The final two's complement circuit is shown in Figure 6.

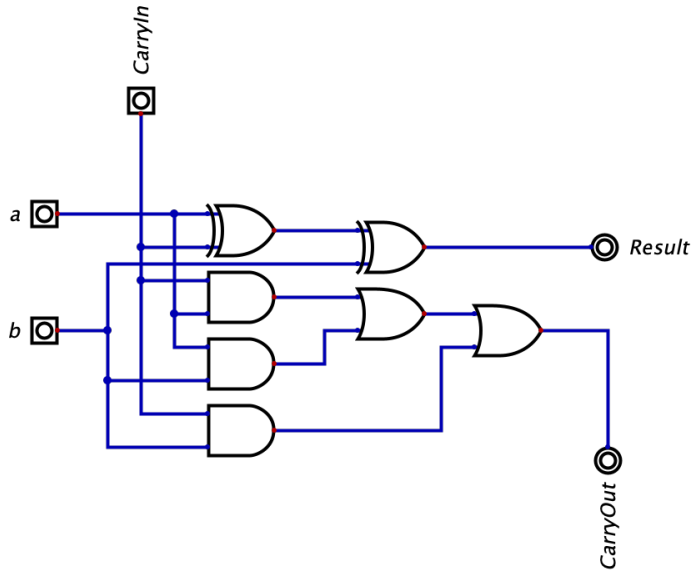


Fig. 3. Full-Adder Circuit

D. Question 4

For a 8-bit adder, to calculate the depth and complexity of the circuit, we divide the circuit into two parts:

- 1) In the **inversion stage**, all components are inverted simultaneously, thus depth is 1 and amount of NOT gates required is 7.
- 2) In the **addition stage**, the two 4-bit lookahead carry adder one after another. They are connected in series.

Component: Generation of signal G_i and P_i

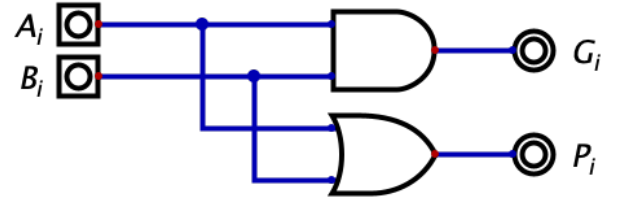


Fig. 4. GP Generator Circuit

For a single 4-bit lookahead, the number of gates utilized is:

$$\underbrace{4}_{\text{GP Generator}} + \underbrace{4 \times 7}_{\text{Full Adder}} + \underbrace{14}_{\text{In the middle layer}} = 46 \quad (3)$$

And the depth is:

$$\underbrace{1}_{\text{GP Generator}} + \underbrace{2}_{\text{In the middle layer}} = 3 \quad (4)$$

- 3) Since there are two lookahead carry adder, the total depth is

$$1 + 3 \times 2 = 7 \quad (5)$$

and number of gates utilized is:

$$7 + 46 \times 2 = 100 \quad (6)$$

Component: 4-bit adder

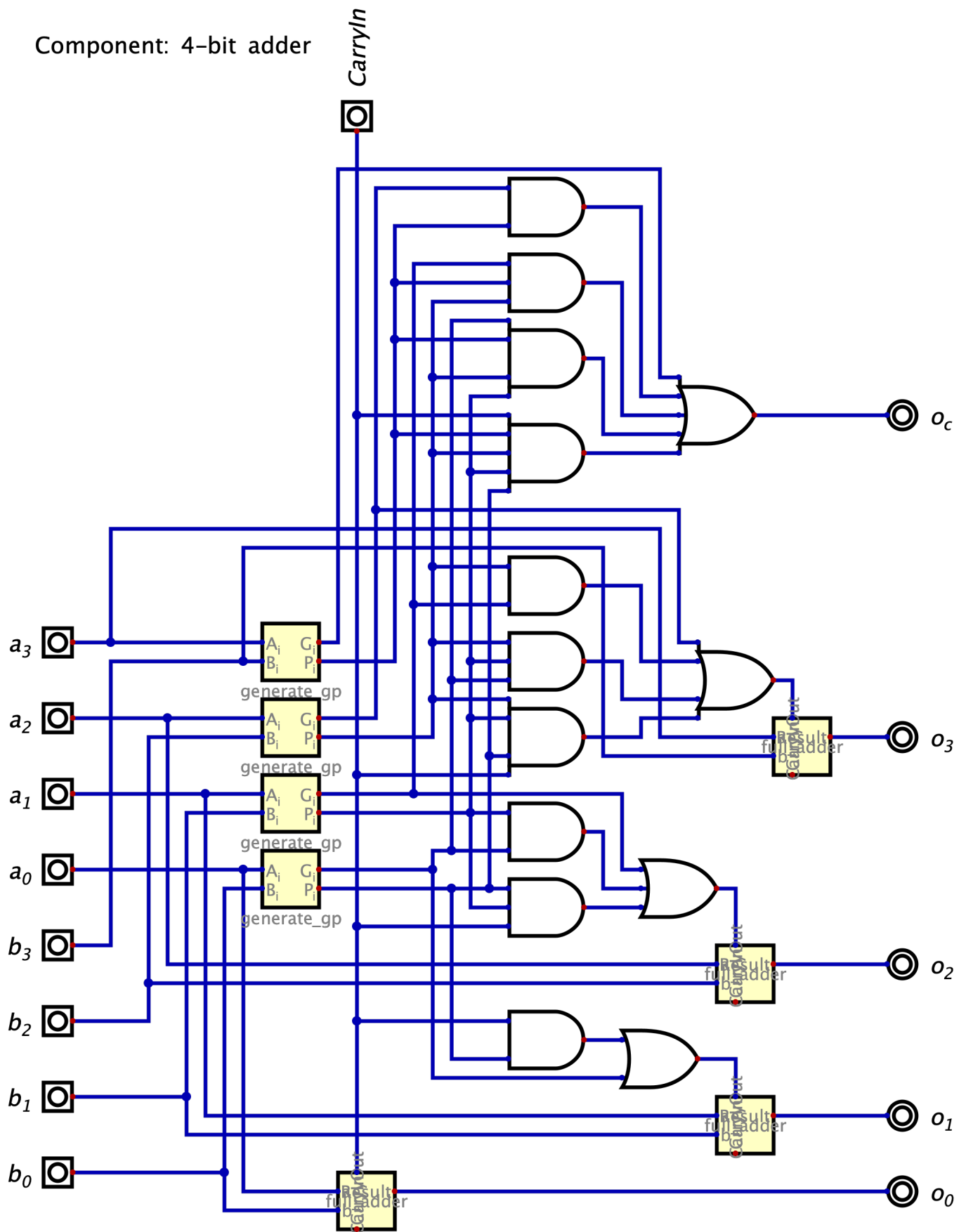


Fig. 5. 4-bit Lookahead Carry Adder

Question 3

Test

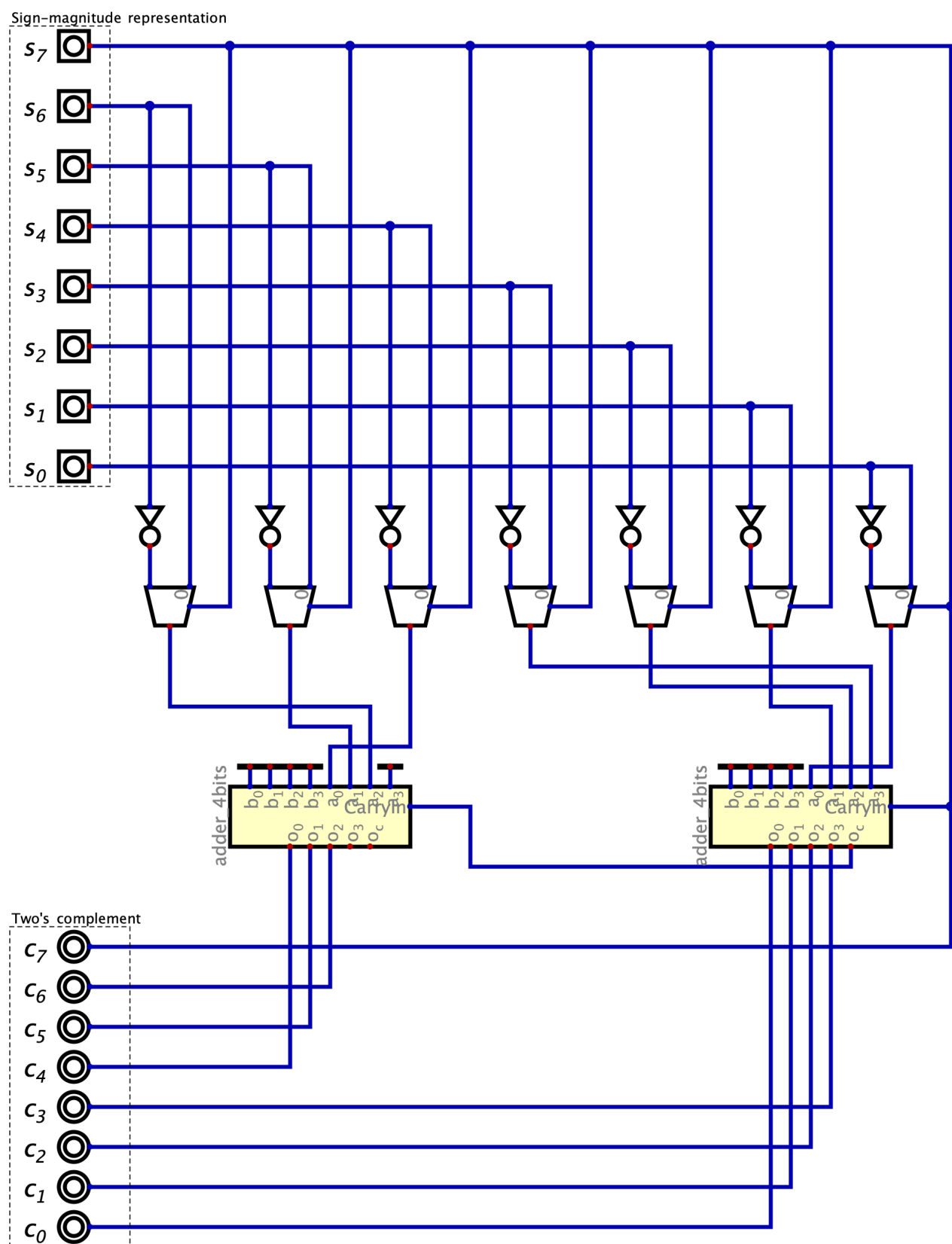


Fig. 6. 8-bit Two's Complement Circuit (Divide and Conquer Method)