

Computer Architecture
Lecture Week 4 (Part II)

Adder and Subtractor Circuits

Artem Burmyakov

September 26, 2025



INNOPOLIS
UNIVERSITY

Recap: Binary Values Summation

Recap: Binary Values Summation

$$\begin{array}{r} + \quad 0 \quad 0 \quad 1 \quad 1 \\ \quad 0 \quad 0 \quad 1 \quad 0 \\ \hline \end{array}$$

Recap: Binary Values Summation

Most Significant Bits (MSB)

Least Significant Bits (LSB)

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 0 \\ \hline \end{array}$$

Recap: Binary Values Summation

Most Significant Bits (MSB)

Least Significant Bits (LSB)

$$\begin{array}{r} + \quad \begin{array}{|c|} \hline 0 \\ \hline 0 \\ \hline \end{array} \quad 0 \quad 1 \quad \begin{array}{|c|} \hline 1 \\ \hline 0 \\ \hline \end{array} \\ \hline \end{array}$$



Computation starts with the least significant bits;
Excess is carried-out

Recap: Binary Values Summation

	0	0	1	1
+	0	0	1	0
				1

Recap: Binary Values Summation

“0” carried out

	0	0	1	1
+	0	0	1	0
				1

Recap: Binary Values Summation

“1” carried out

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 0 \\ \hline & & & 0 & 1 \end{array}$$

"1" carried out


$$\begin{array}{r} + \quad 0 \quad 0 \quad 1 \quad 1 \\ \quad 0 \quad 0 \quad 1 \quad 1 \\ \hline \quad \quad \quad 0 \end{array}$$

Recap: Binary Values Summation

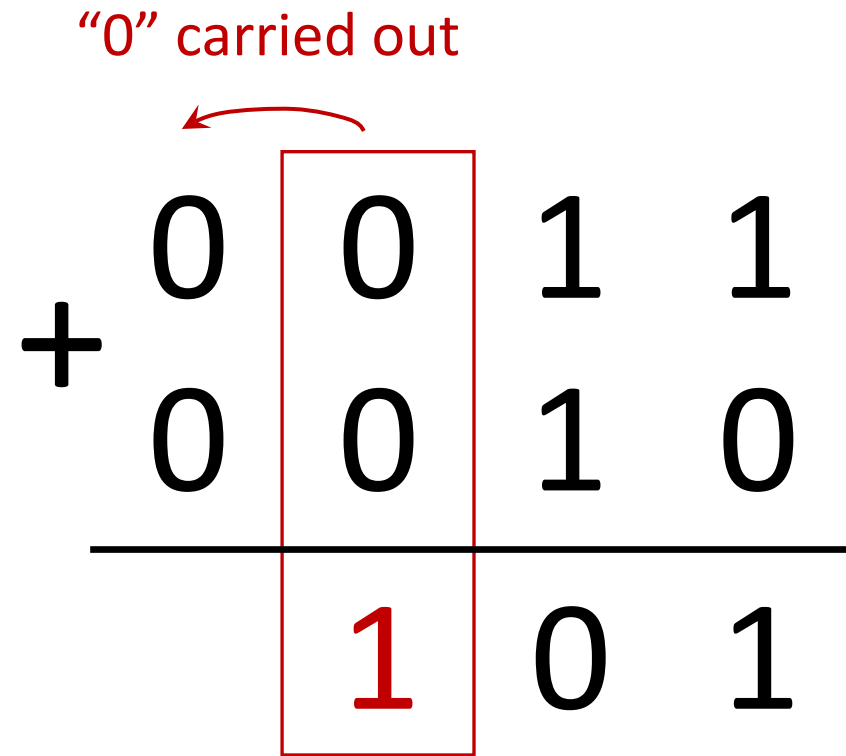
“1” carried out

The diagram illustrates the binary addition of two 4-bit numbers. The first number is 0011 and the second is 0010. The sum is 0101. A red box highlights the second and third columns, with a red arrow pointing from the third column to the second, indicating a carry-out of 1. The result '1' in the second column is also red.

	0	0	1	1
+	0	0	1	0
	1	0	1	

Recap: Binary Values Summation

“0” carried out


$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 0 \\ \hline & 1 & 0 & 1 & \end{array}$$

Recap: Binary Values Summation

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 0 & 0 & 1 & 0 \\ \hline & 0 & 1 & 0 & 1 \end{array}$$

Recap: Binary Values Summation

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & & & & \\ & 0 & 0 & 1 & 0 \\ \hline & 0 & 1 & 0 & 1 \end{array}$$

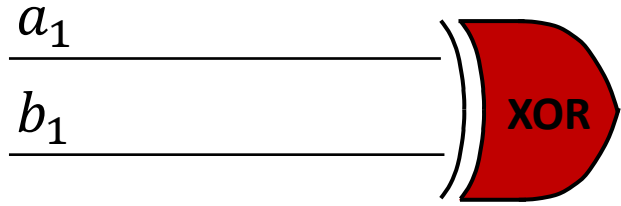
The result of summation:

Adder Circuit Design

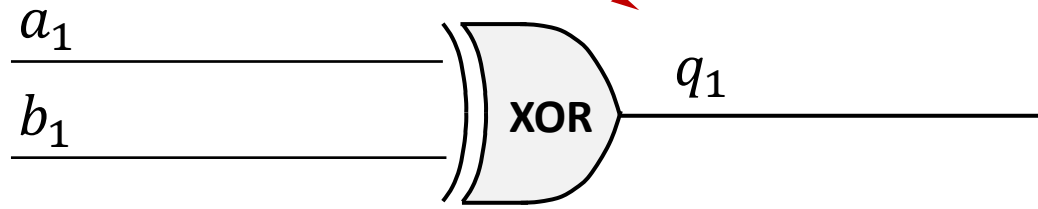
Adder Circuit Design

$$\begin{array}{r} a_1 \\ \hline b_1 \\ \hline \end{array}$$

Adder Circuit Design



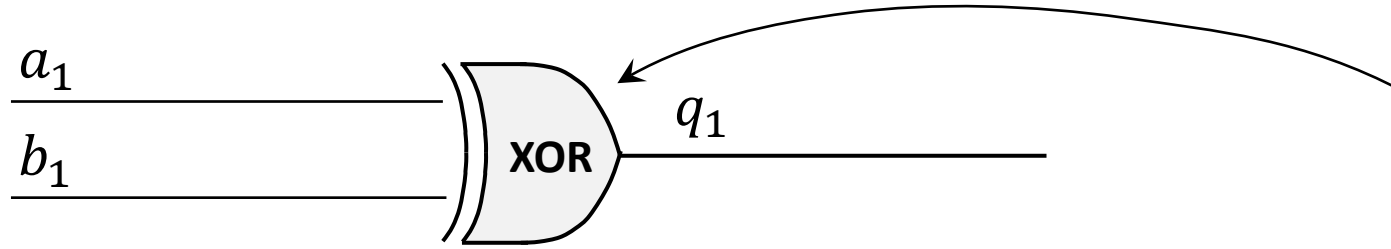
Adder Circuit Design



XOR Truth Table:

a	b	q
0	0	0
0	1	1
1	0	1
1	1	0

Adder Circuit Design



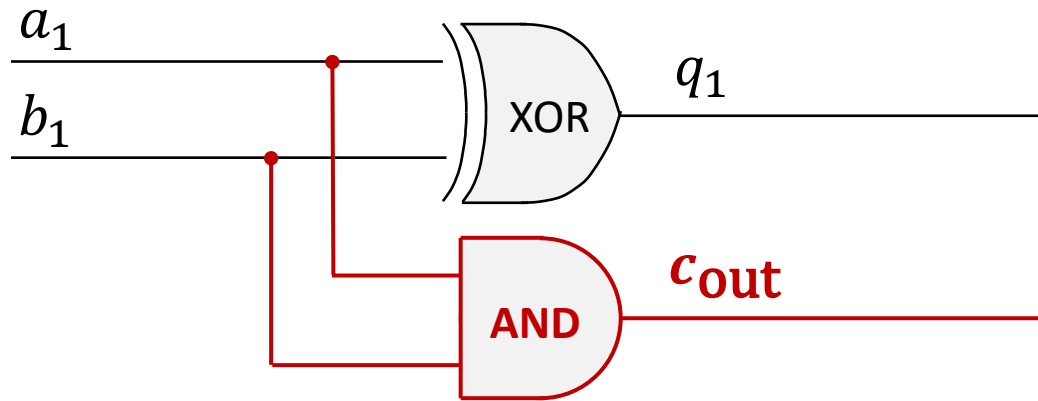
a	b	q
0	0	0
0	1	1
1	0	1
1	1	0

Carry-out case is not handled
(when both inputs are "1"s)

"1" carried out

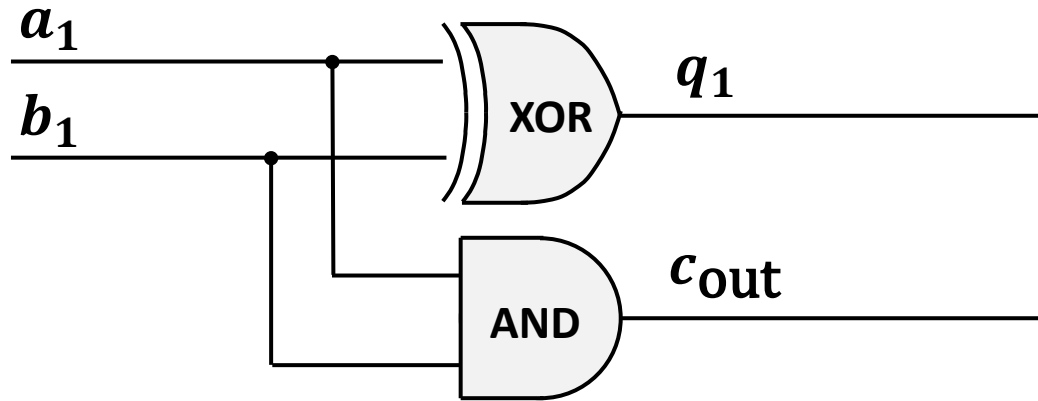
$$\begin{array}{r} + \quad 0 \quad 0 \quad 1 \quad 1 \\ \quad 0 \quad 0 \quad 1 \quad 0 \\ \hline \quad 0 \quad 1 \end{array}$$

Adder Circuit Design

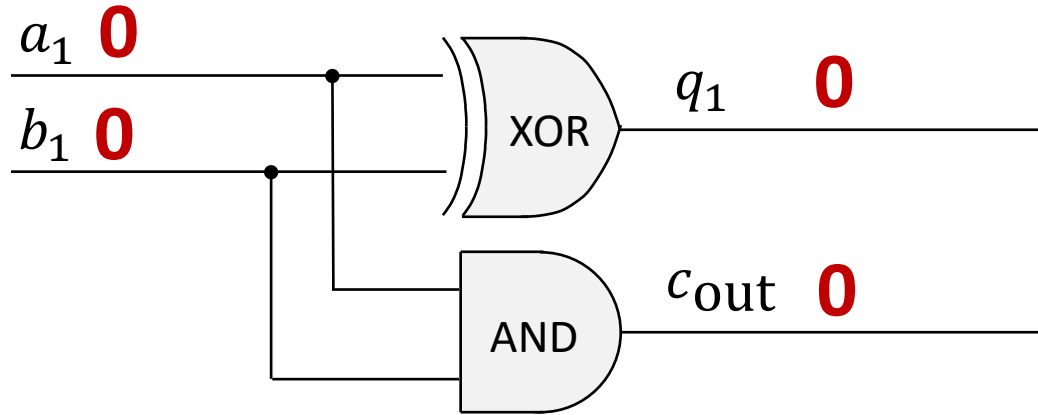


Carry-out case handling
(when both inputs are "1"s)

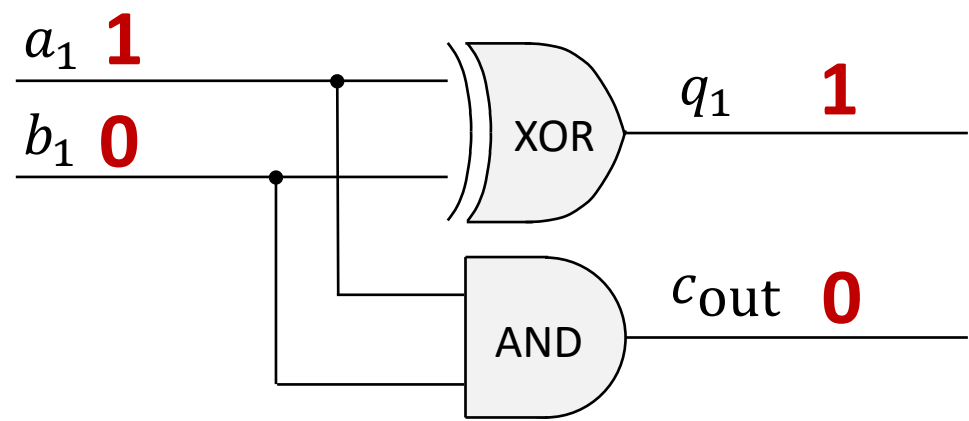
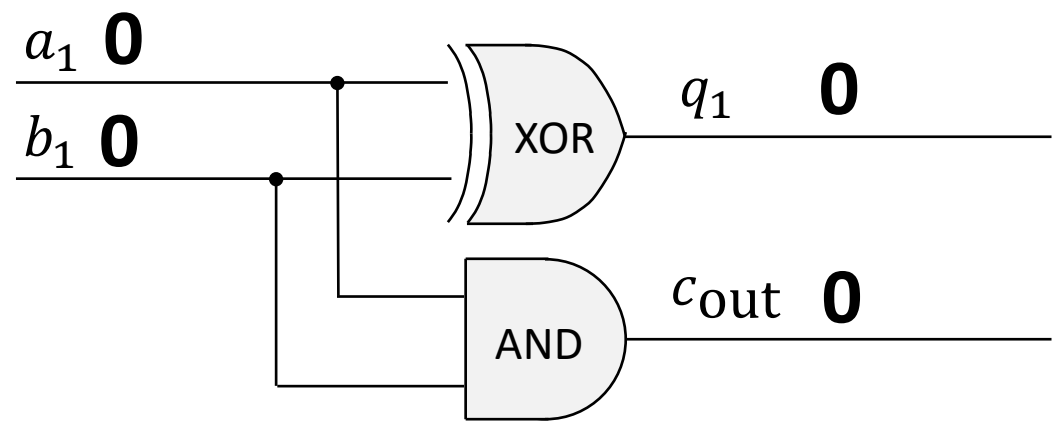
Half-Adder Circuit: Least Significant Bits (LSB) Summation



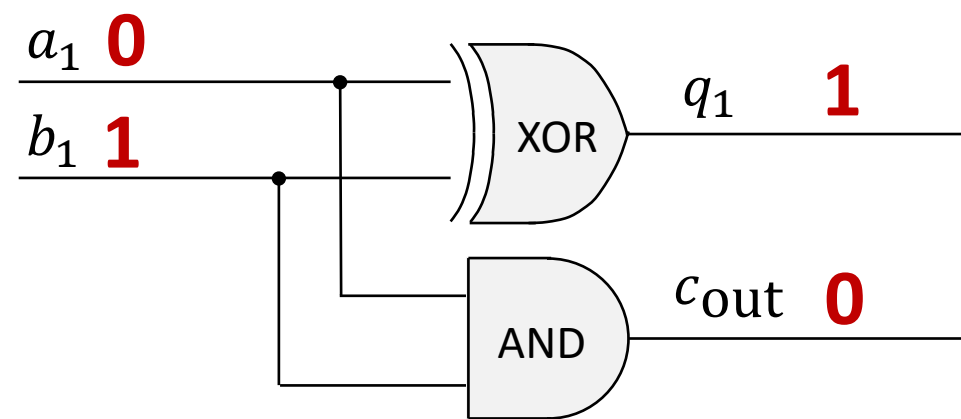
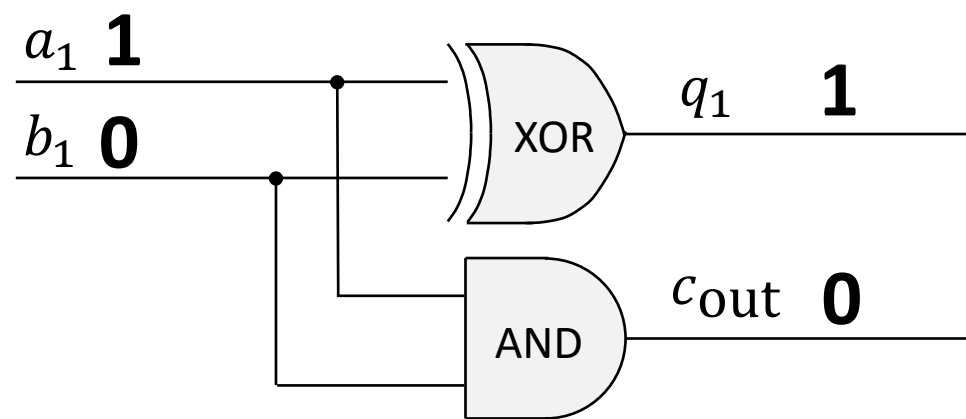
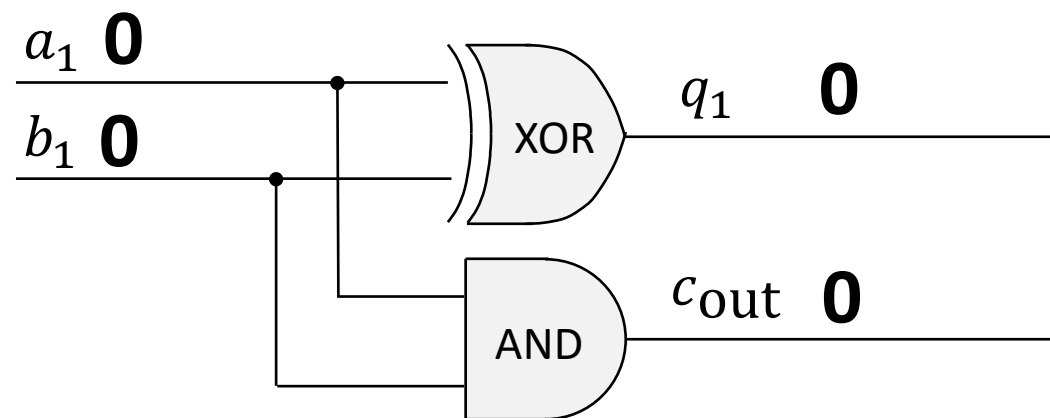
Half-Adder Circuit: Design Verification



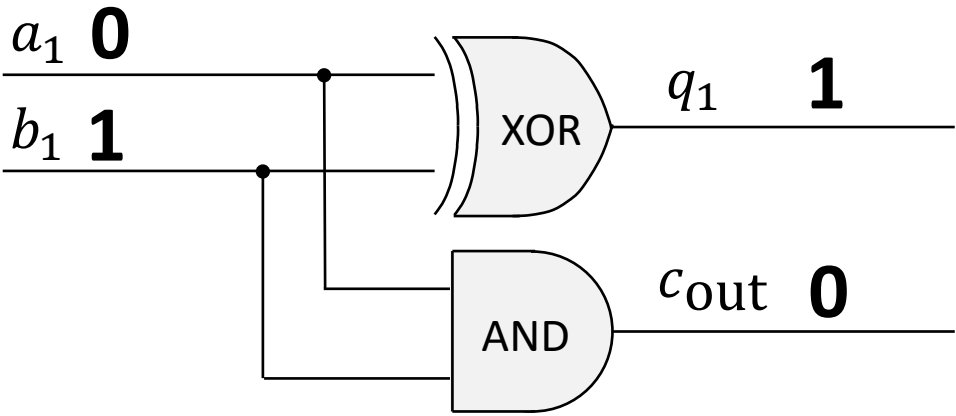
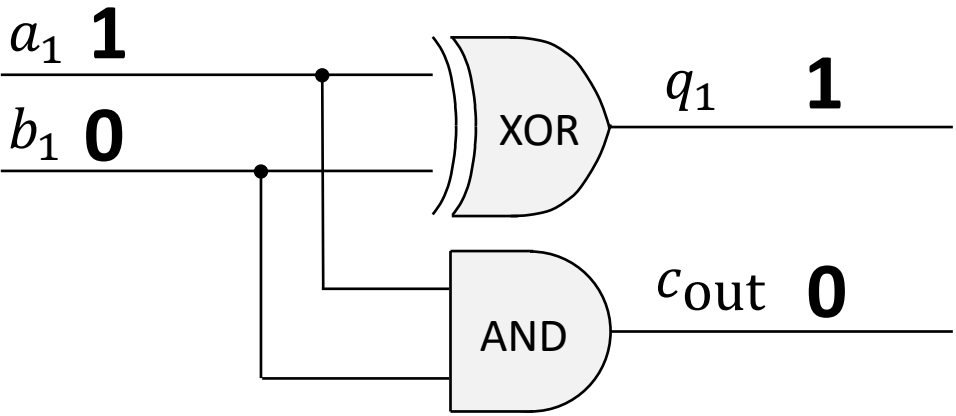
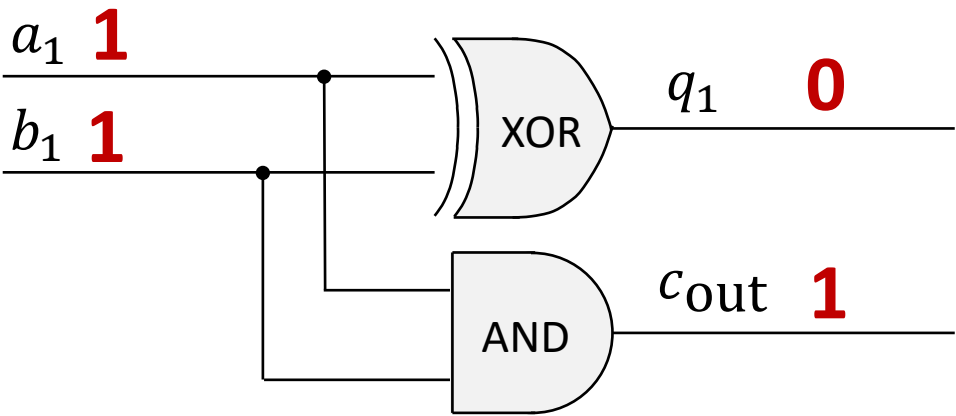
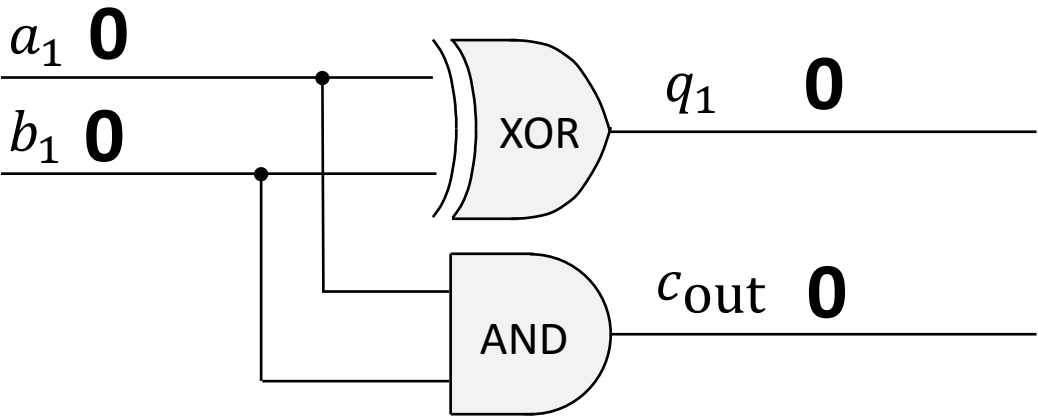
Half-Adder Circuit: Design Verification



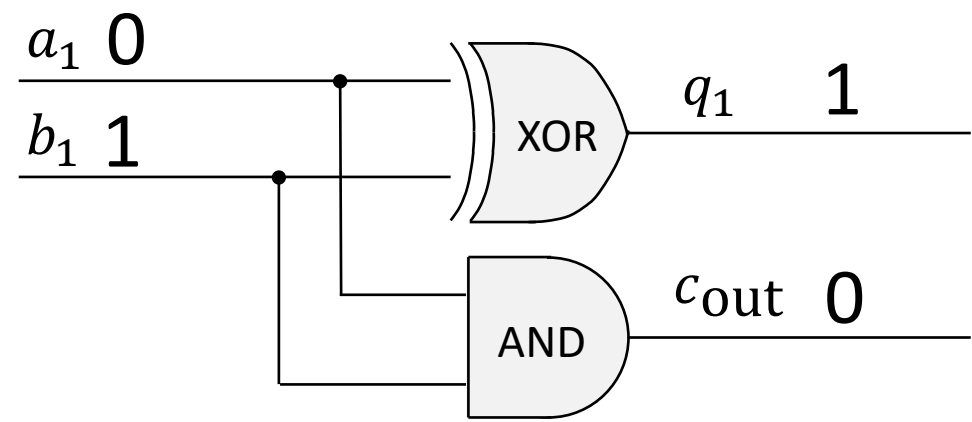
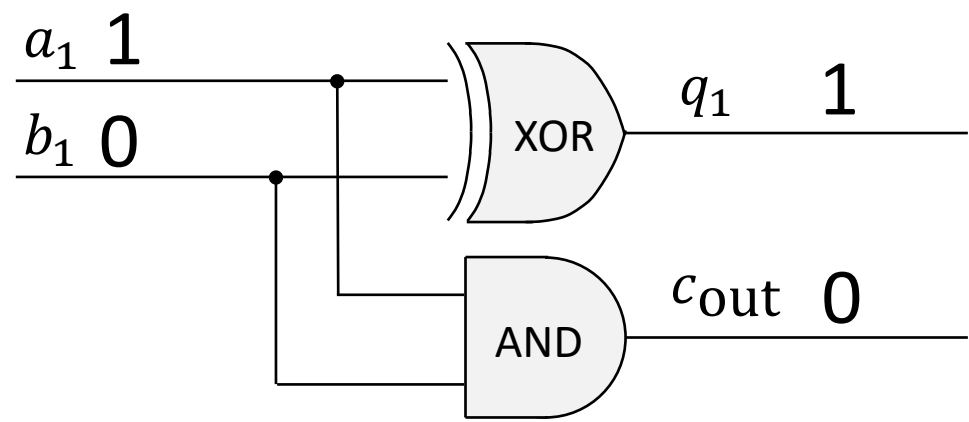
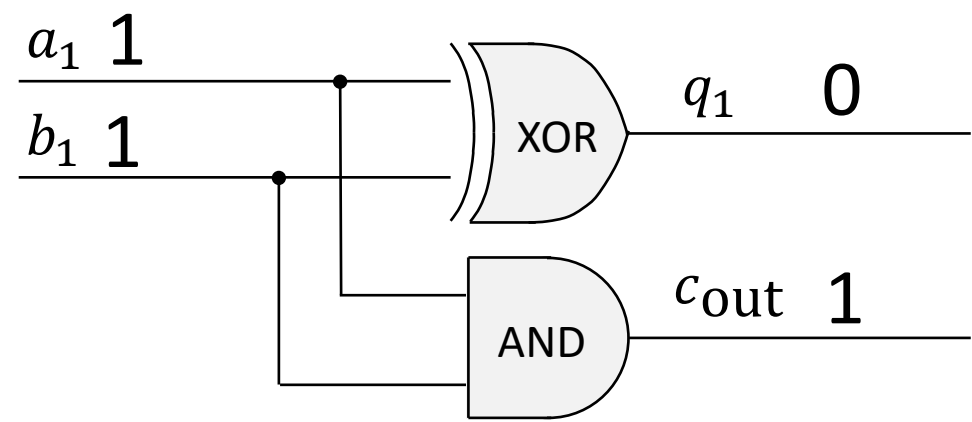
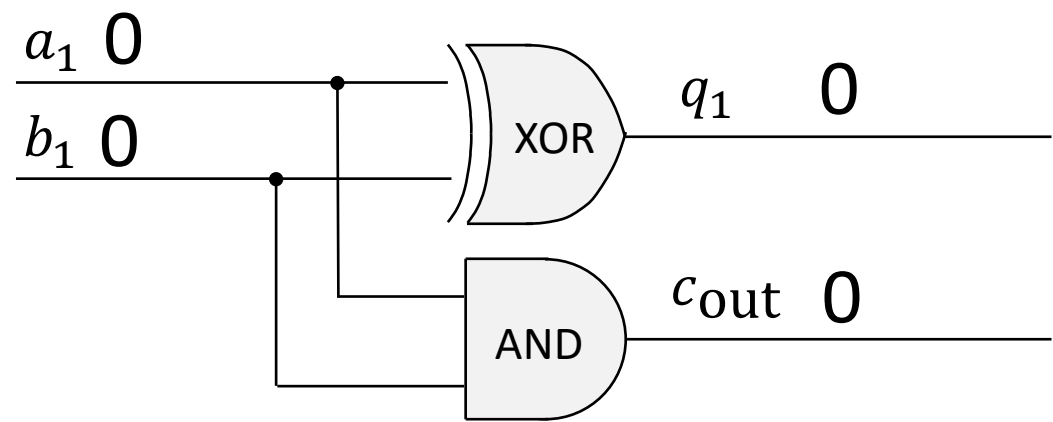
Half-Adder Circuit: Design Verification



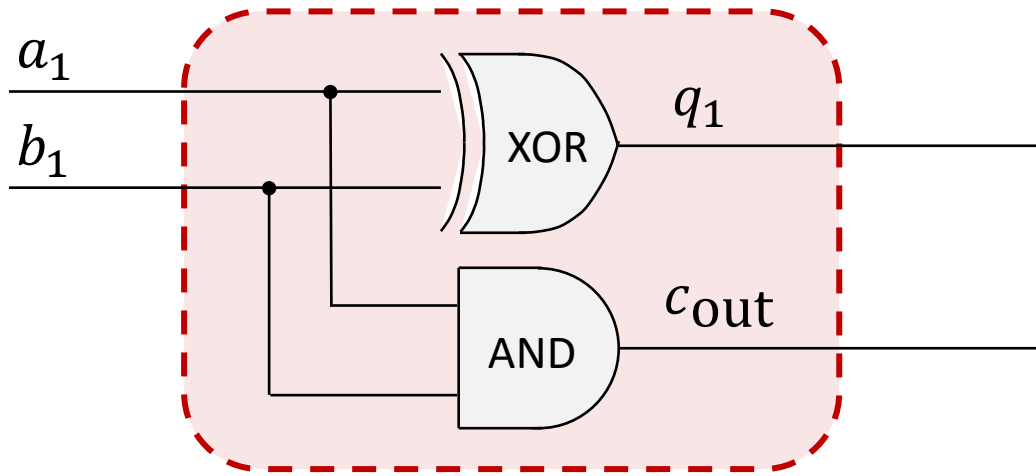
Half-Adder Circuit: Design Verification



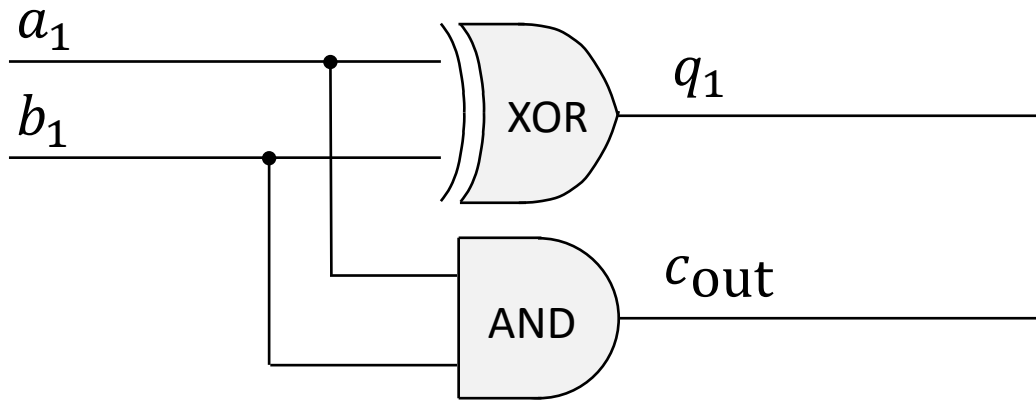
Half-Adder Circuit



Adder Circuit Design: Continuation



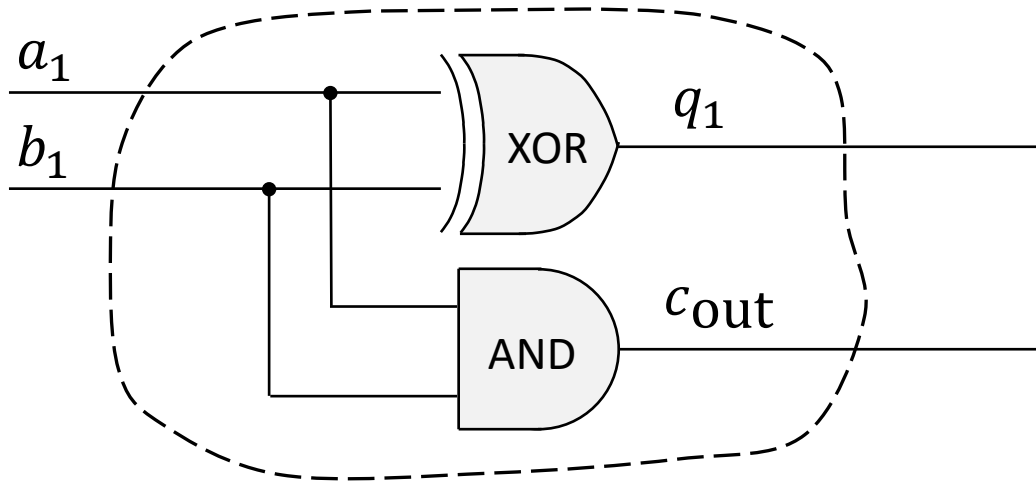
Adder Circuit Design



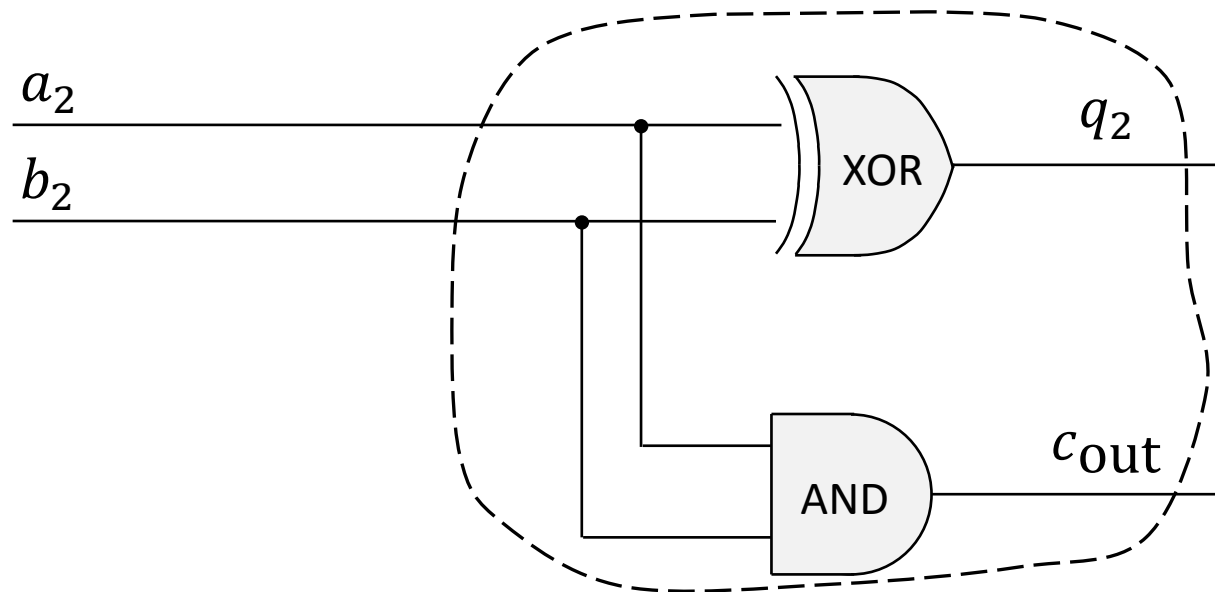
a_2

b_2

Adder Circuit Design



Identical



Adder Circuit Design:

Carry-in is to be handled

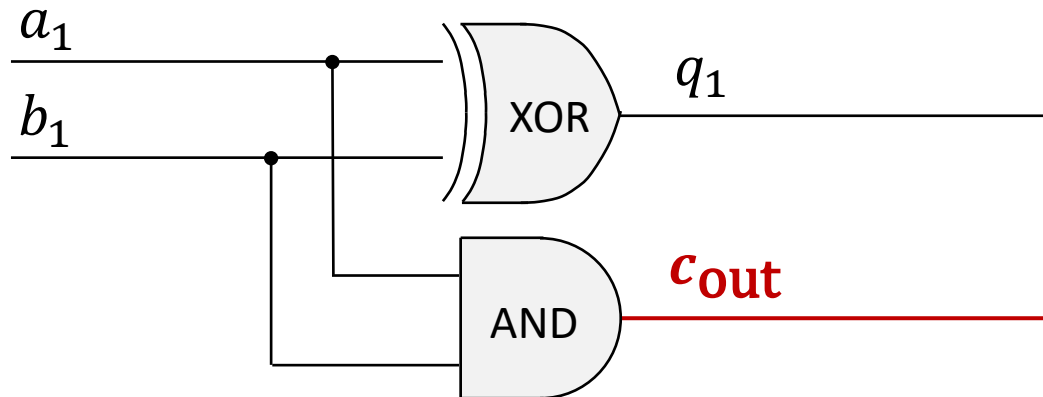
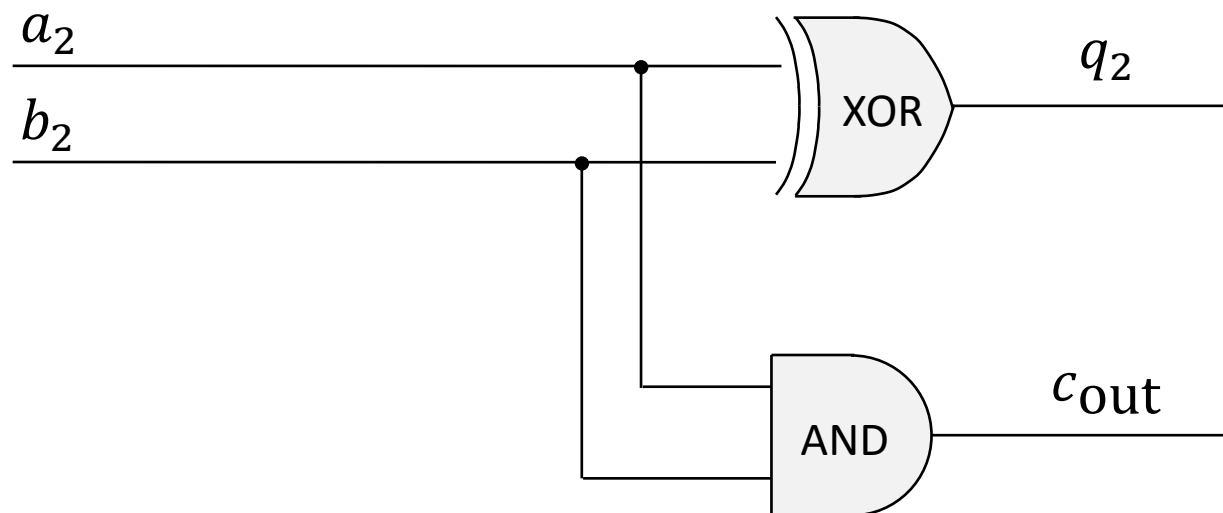
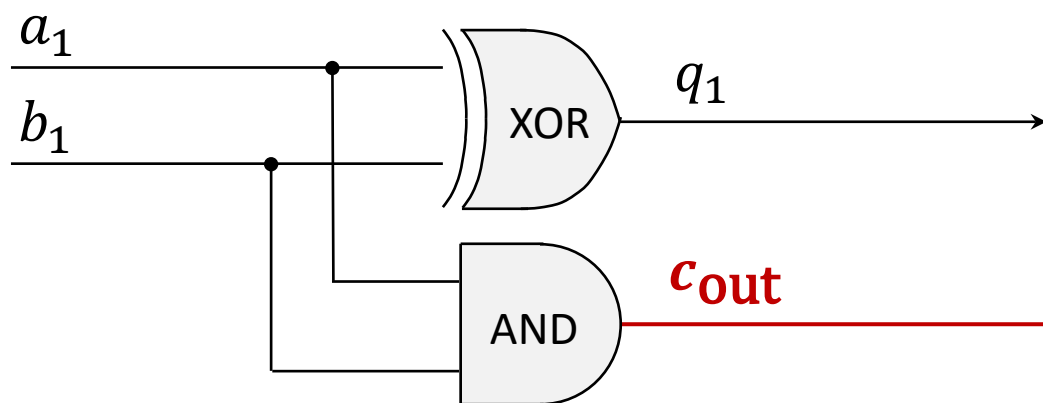


Diagram illustrating the carry-in handling for the second stage. The carry-out from the first stage (c_{out}) is carried into the second stage as a carry-in. The diagram shows the addition of two 3-bit numbers, resulting in a 4-bit sum. The carry-in is indicated by an arrow labeled "1" carried out from the first stage.

	0	0	1	1
+	0	0	1	1
				0

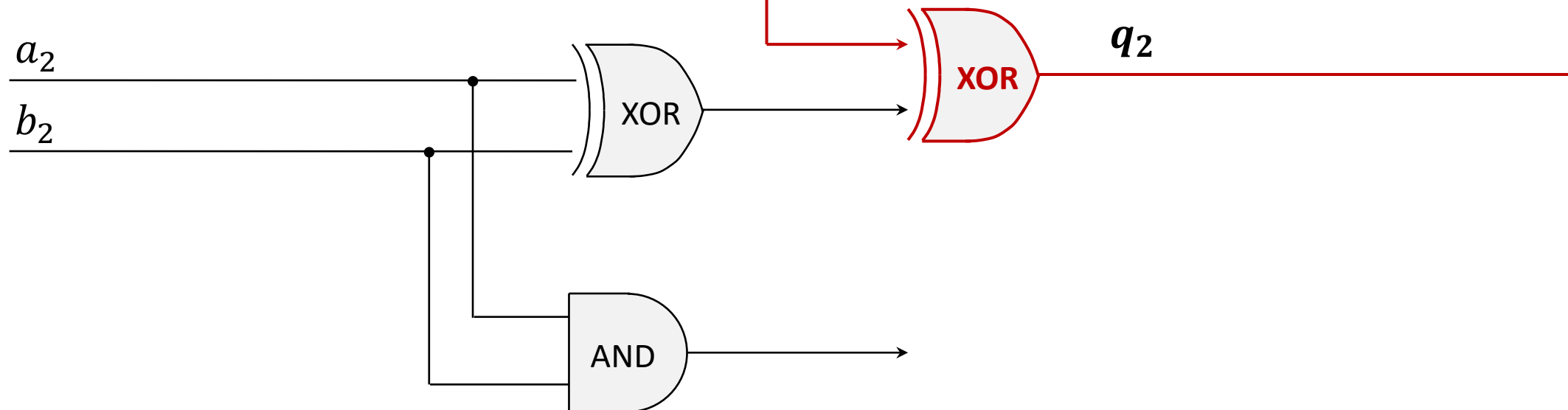


Adder Circuit Design: Carry-in Handling

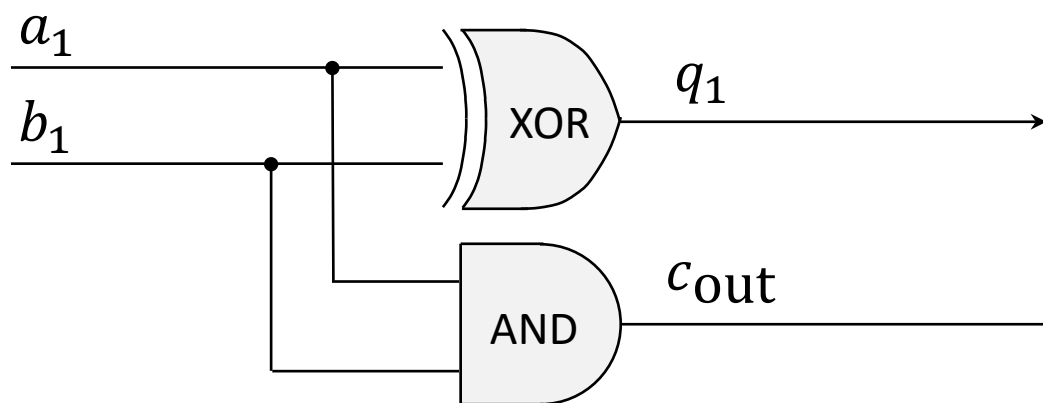


"1" carried out

	0	0	1	1
+	0	0	1	1
				0

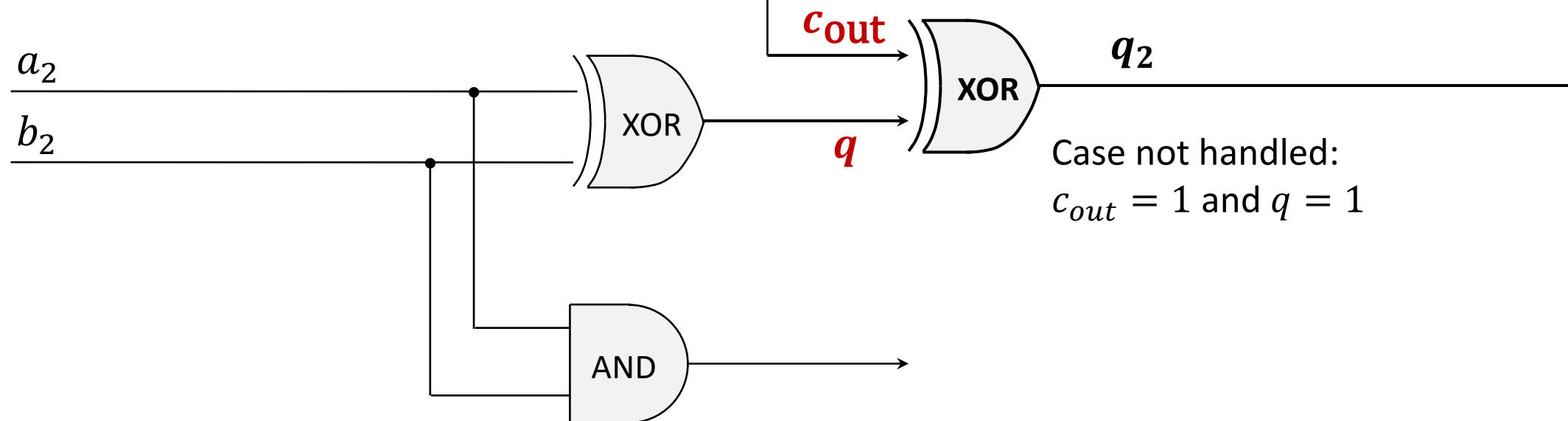


Adder Circuit Design: Carry-in Handling



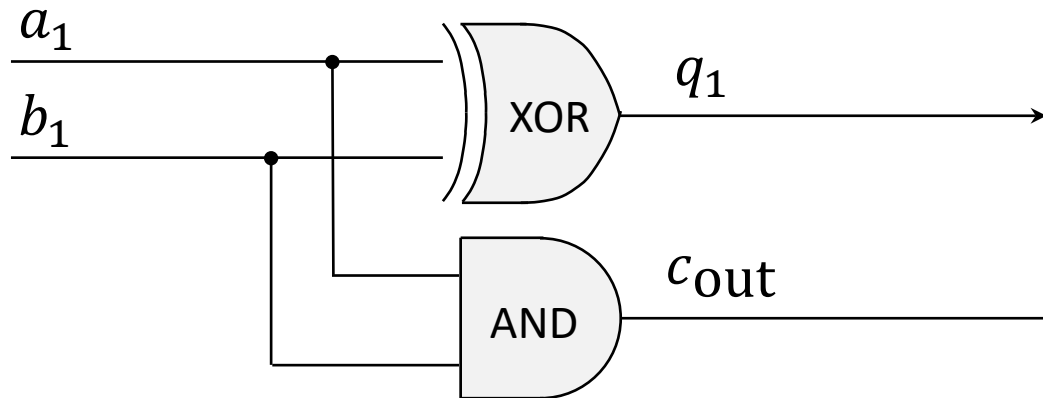
"1" carried out

0	0	1	1
0	0	1	1
			0



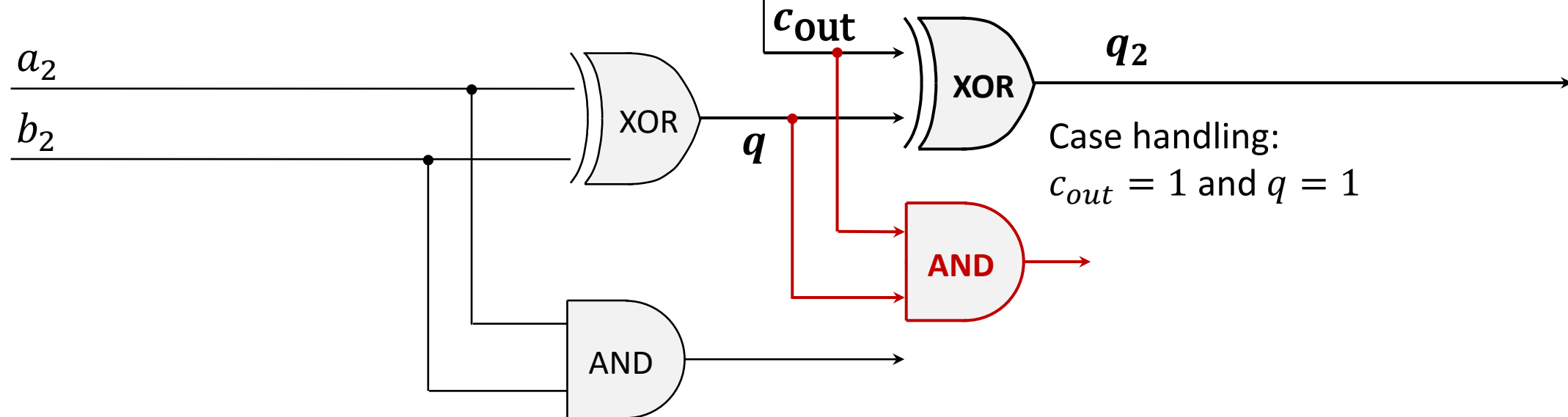
Case not handled:
 $c_{out} = 1$ and $q = 1$

Adder Circuit Design: Carry-in Handling

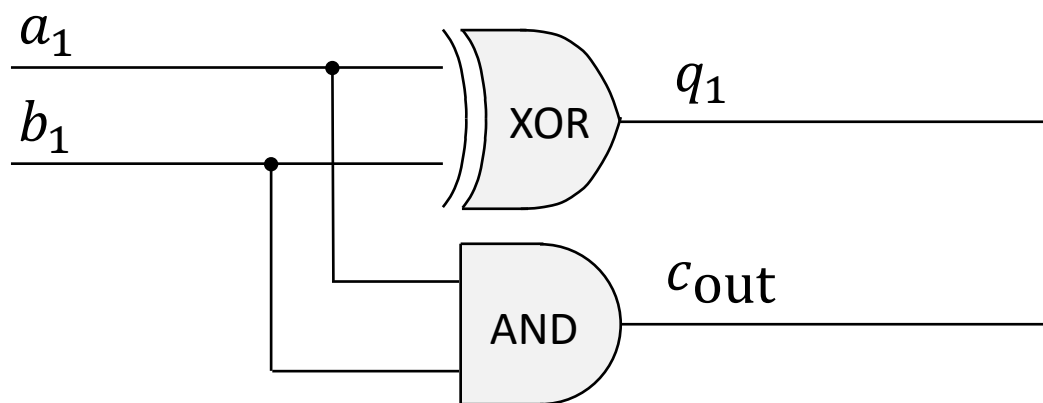


"1" carried out

	0	0	1	1
+	0	0	1	1
				0

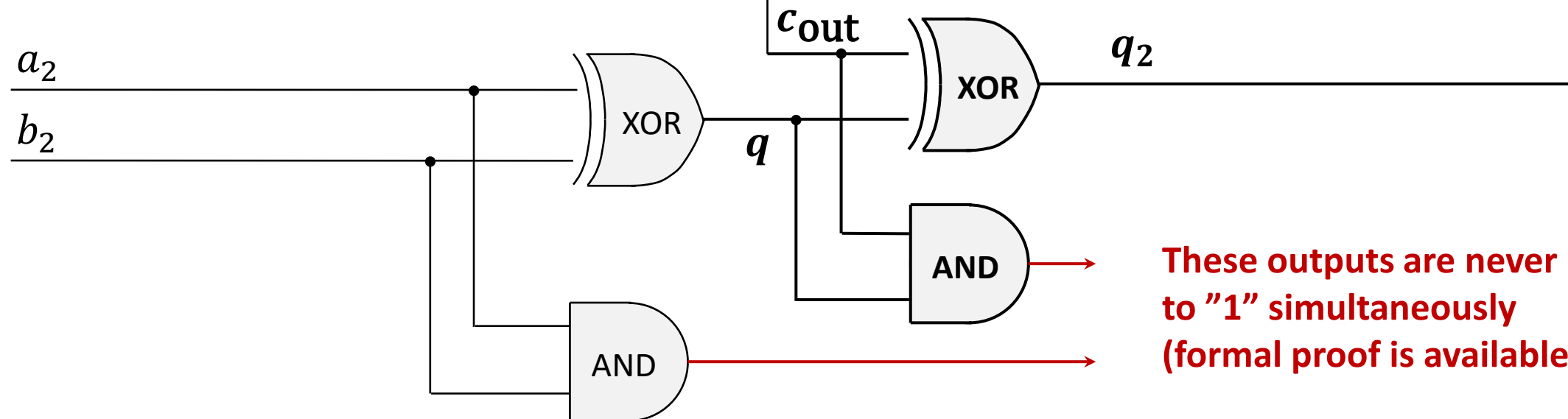


Adder Circuit Design: Carry-in Handling



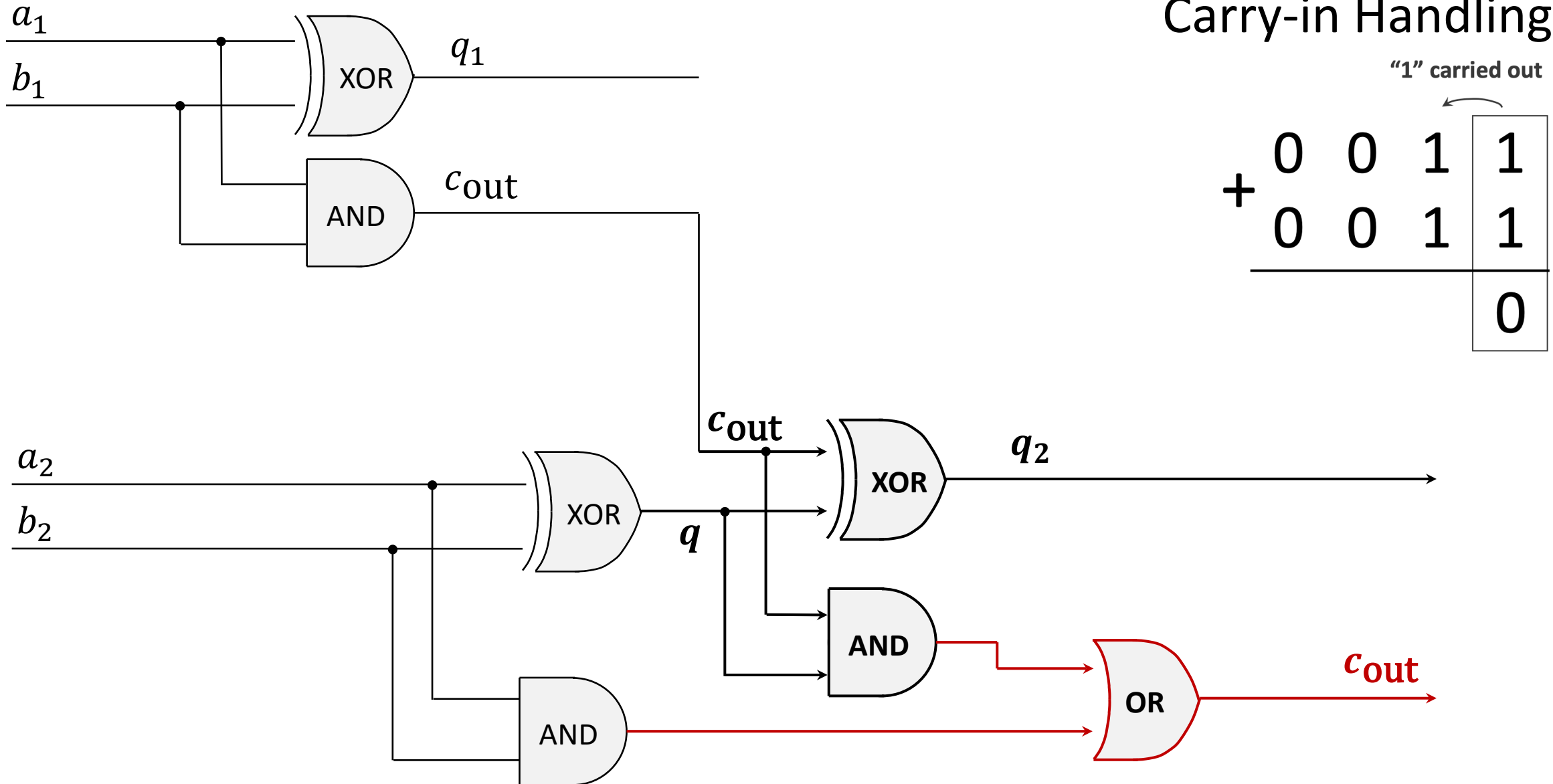
"1" carried out

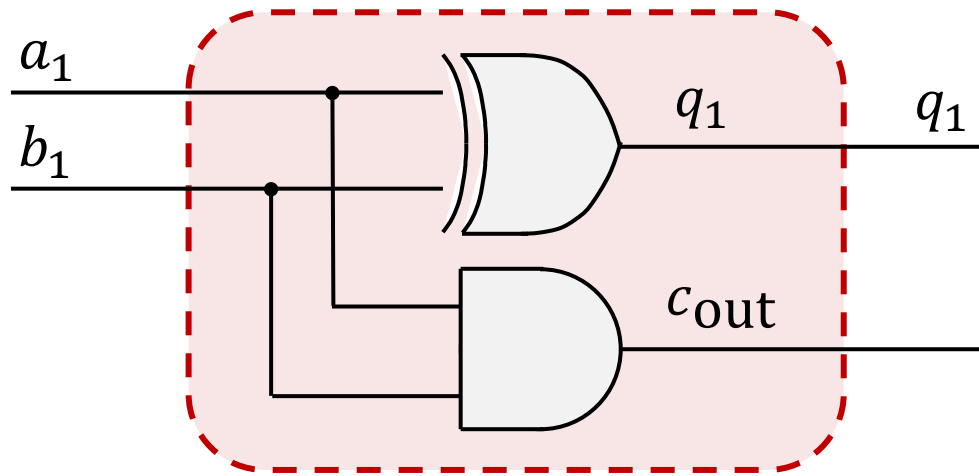
	0	0	1	1
+	0	0	1	1
				0



These outputs are never equal
to "1" simultaneously
(formal proof is available)

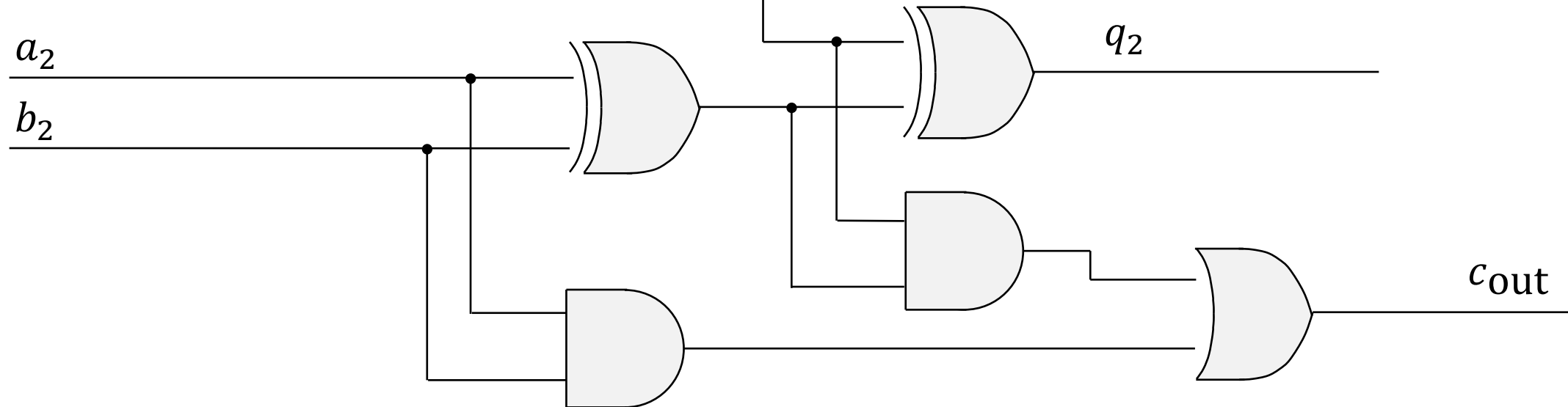
Adder Circuit Design: Carry-in Handling

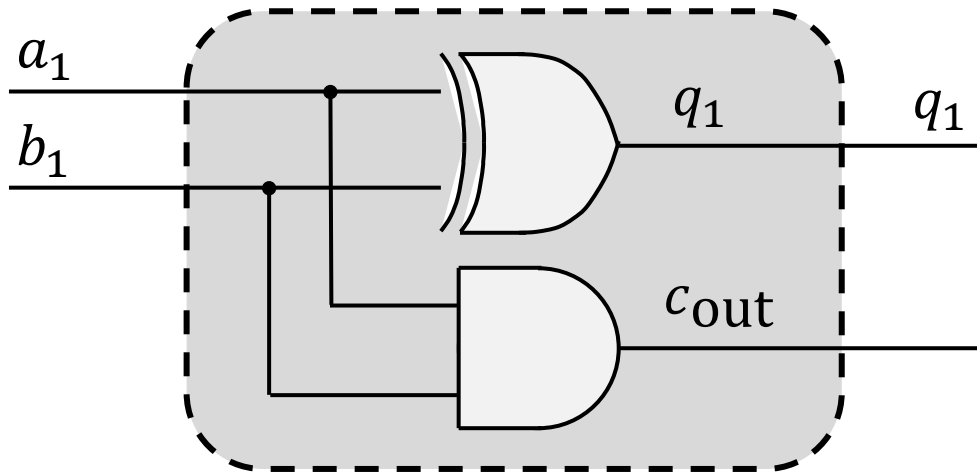




Half-Adder:

- No carry-in handling
- Least Significant Bits (LSB) summation only



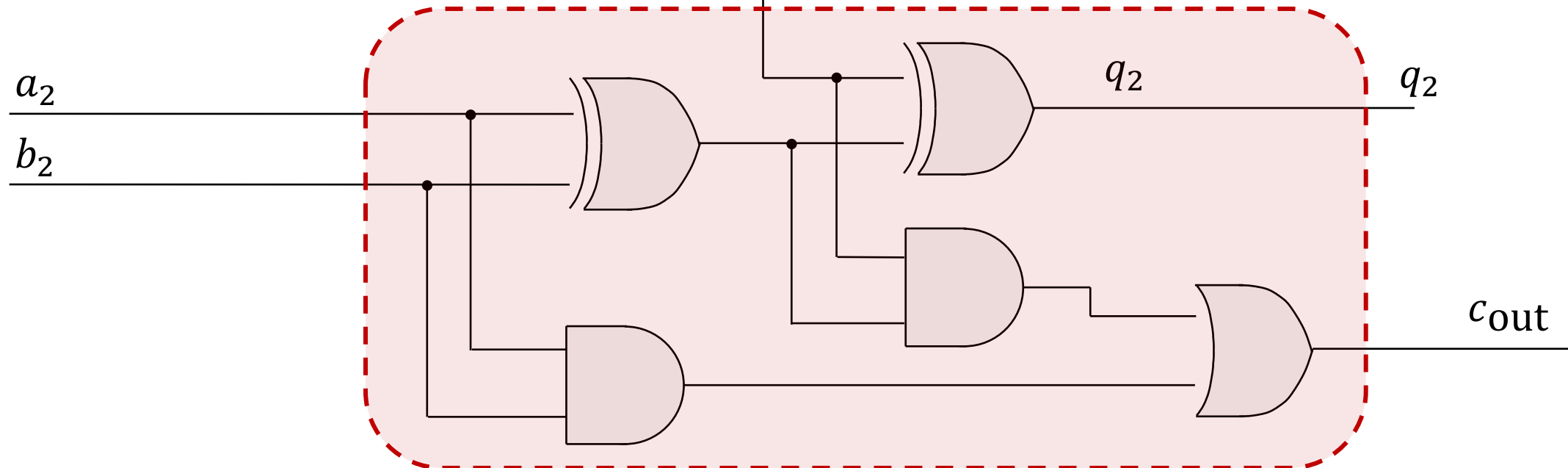


Half-Adder:

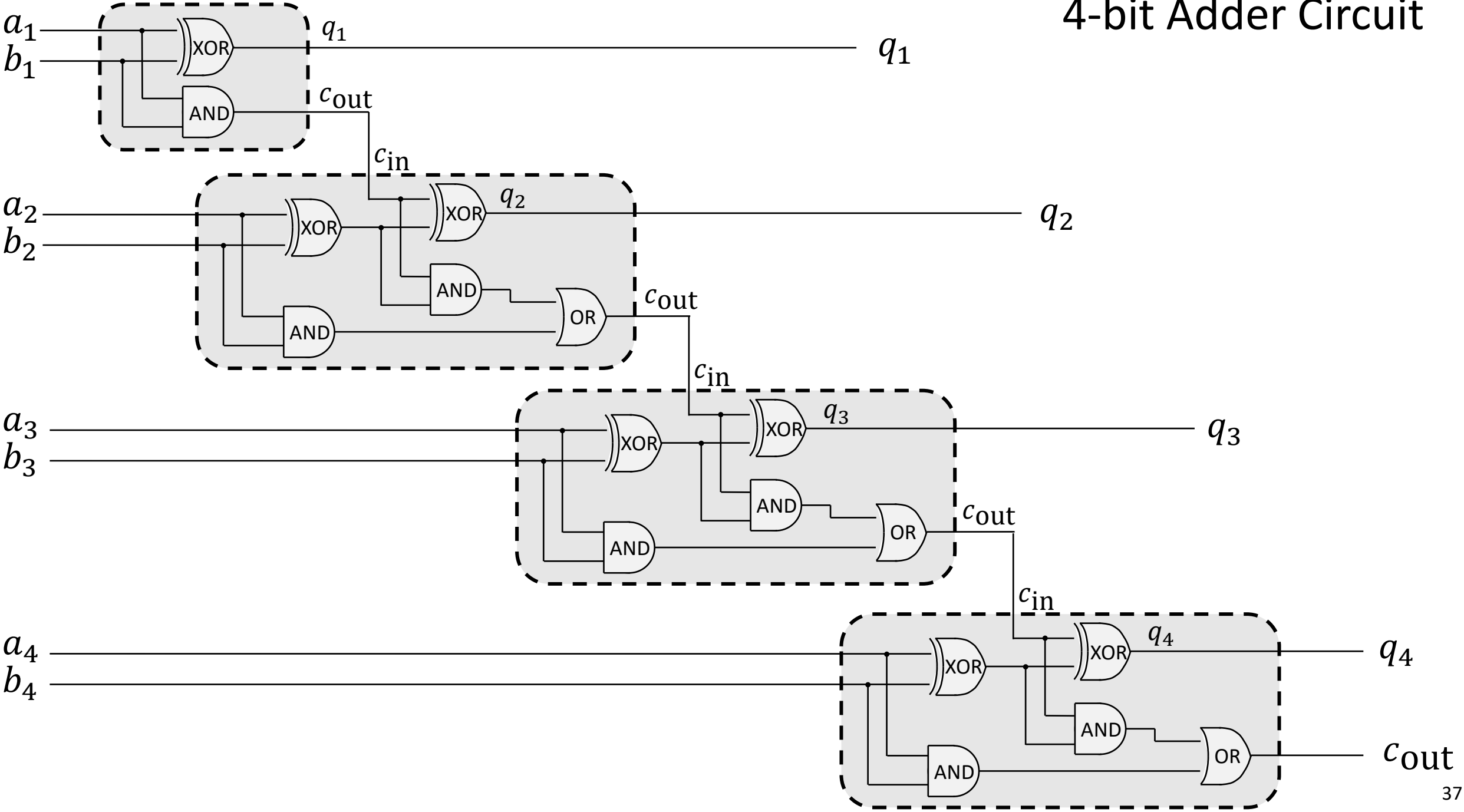
- No carry-in handling
- Least Significant Bits (LSB) summation only

Full-Adder:

- Carry-in handling

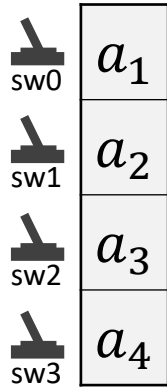


4-bit Adder Circuit

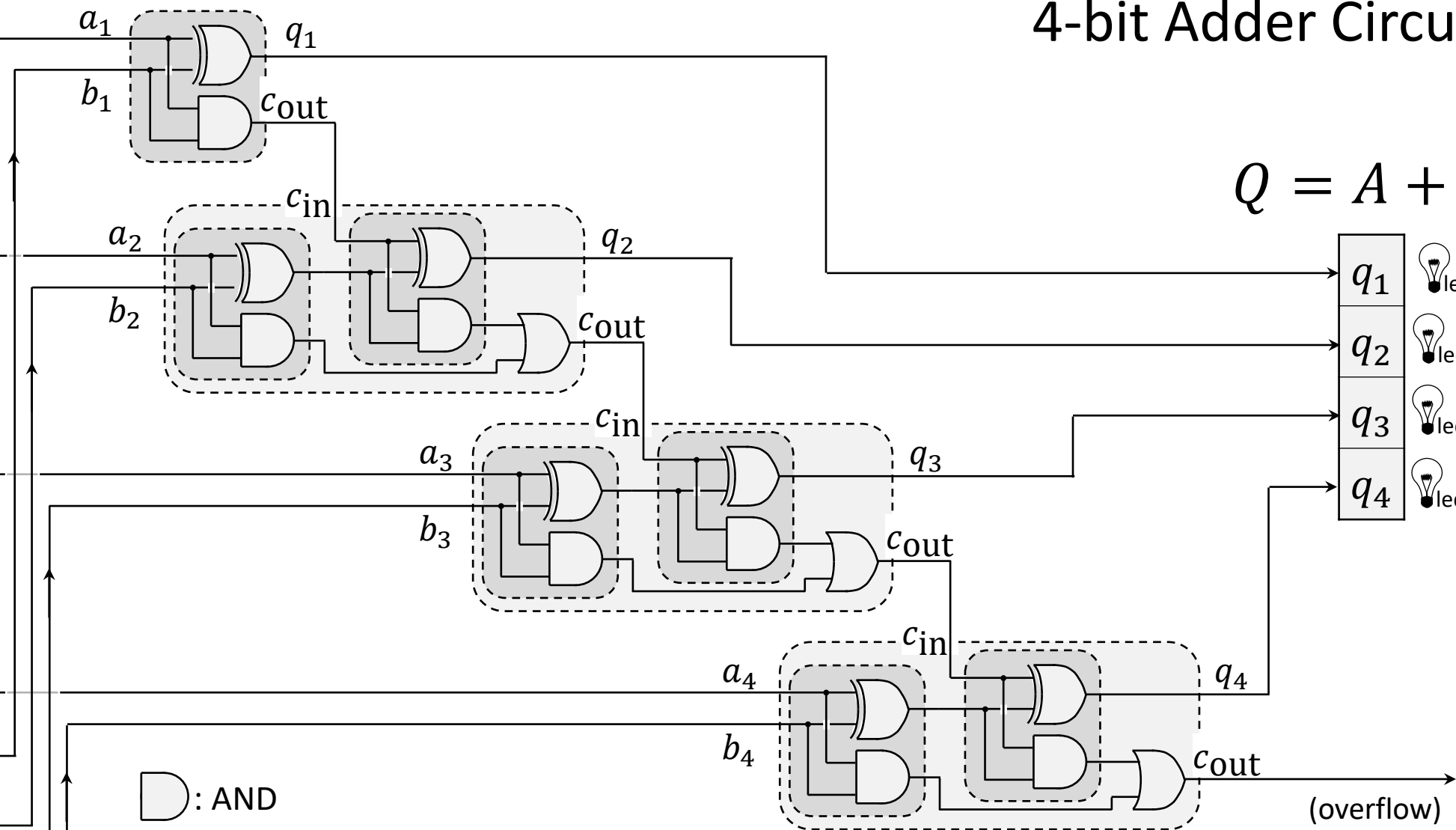
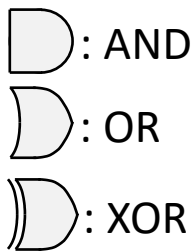
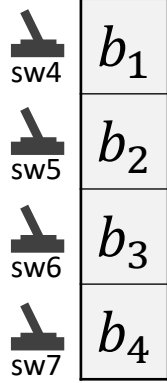


4-bit Adder Circuit

A



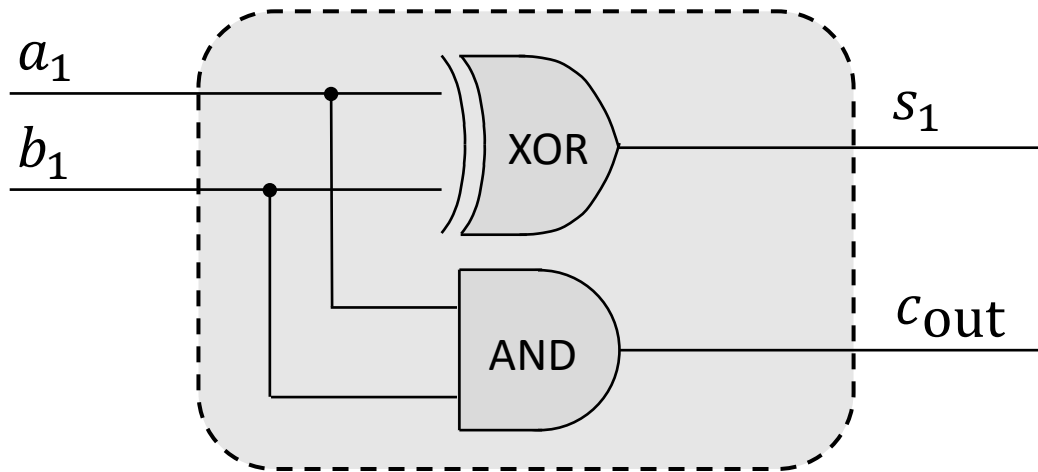
B



$Q = A + B$

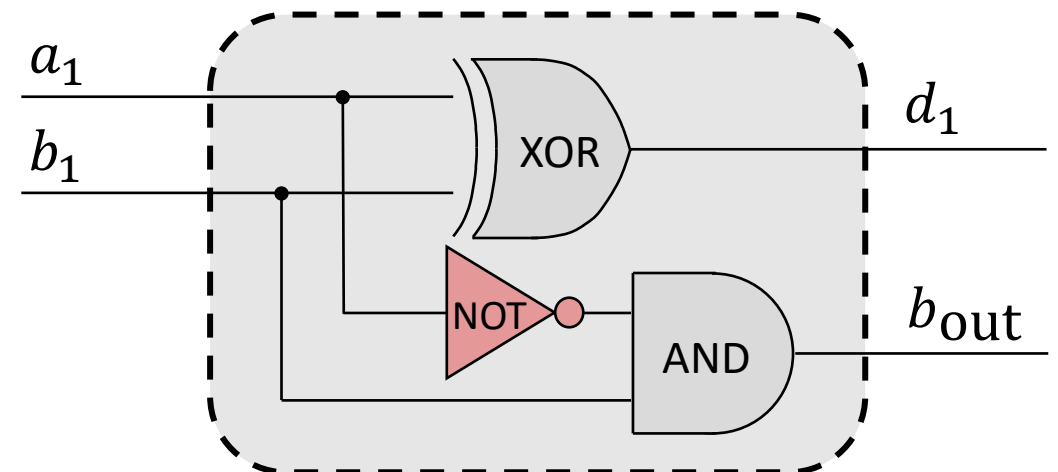
Half Adder:

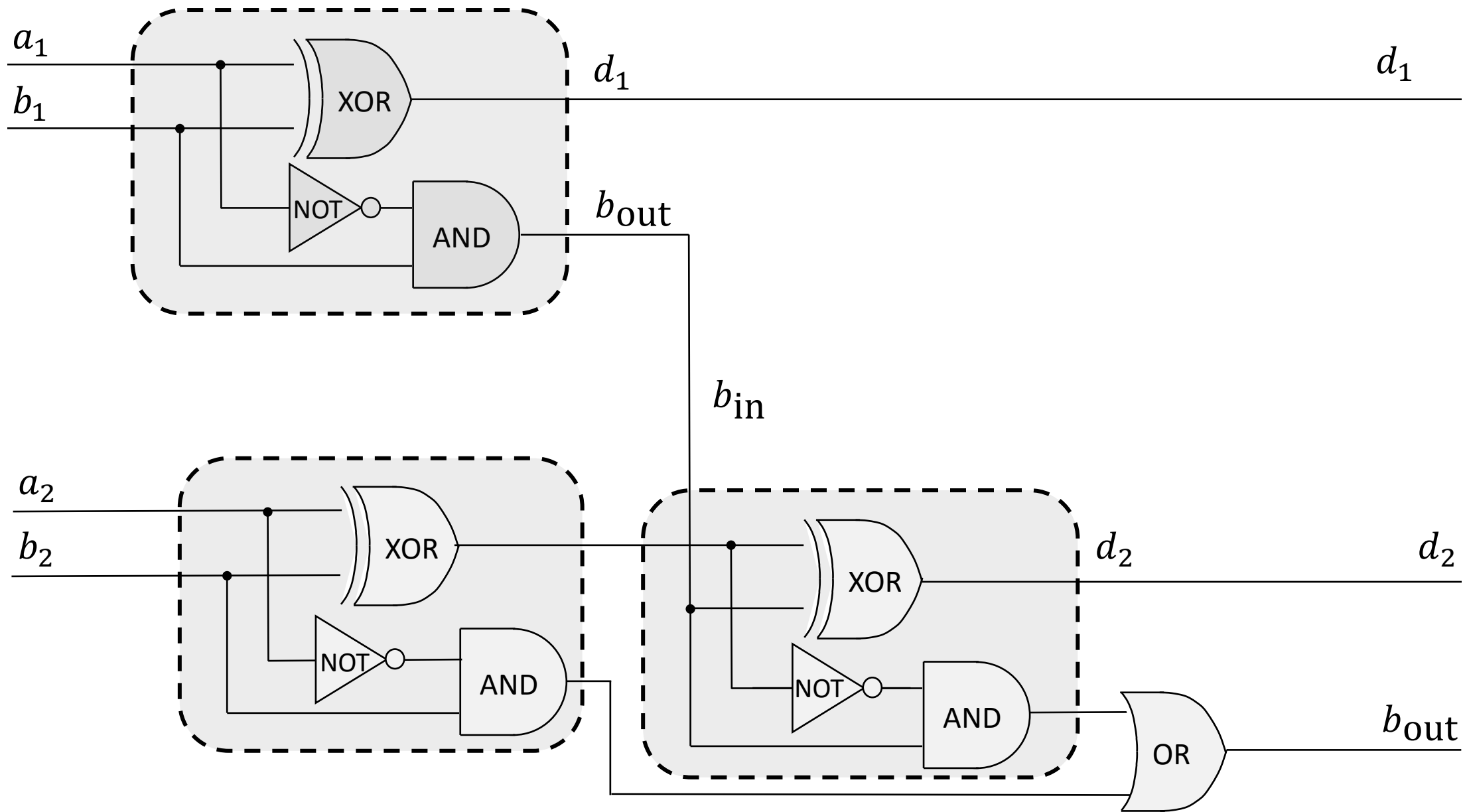
- No carry-in handling
- Summation of Least Significant Bits



Half Subtractor:

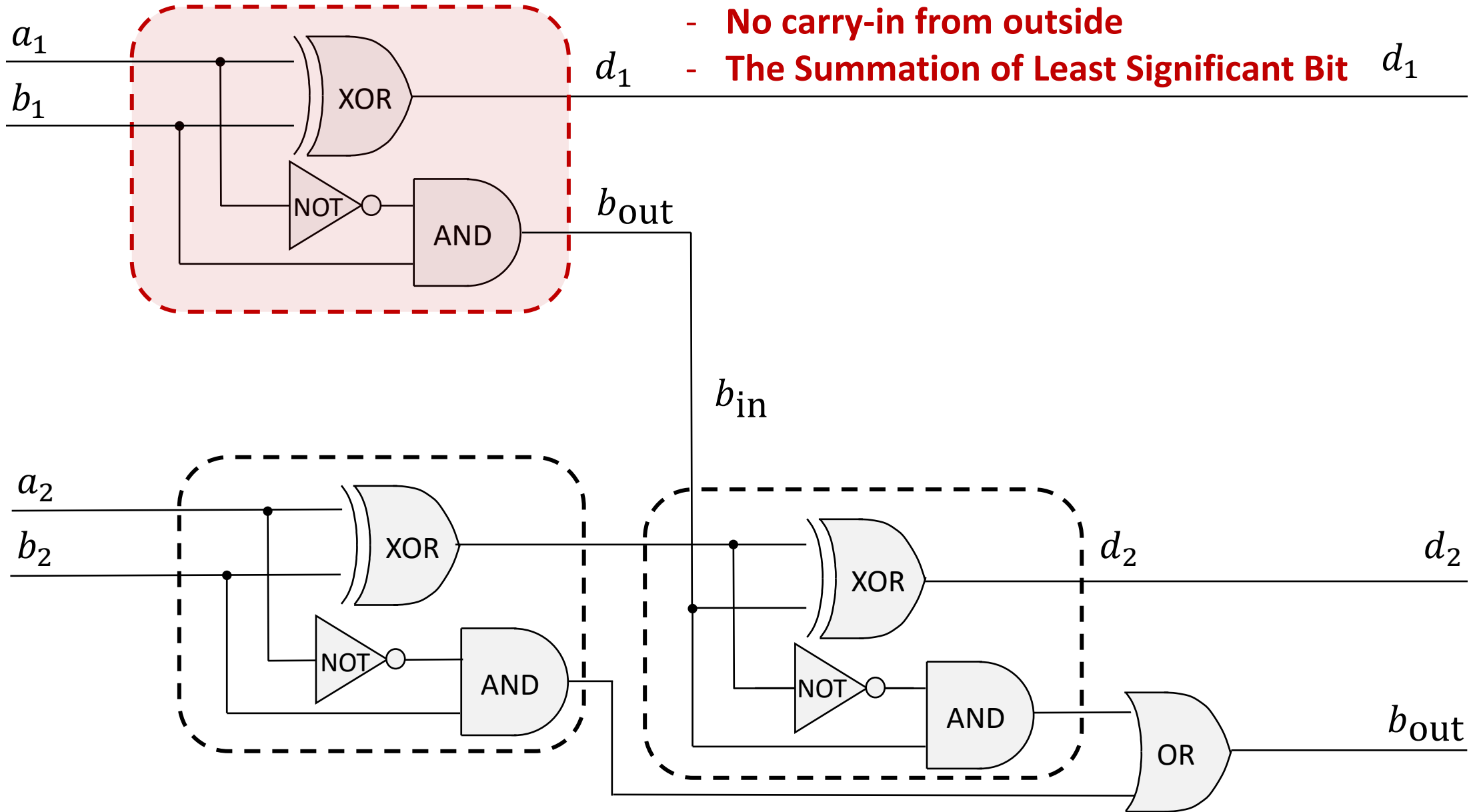
- No borrow-in handling
- Subtraction of Least Significant Bit





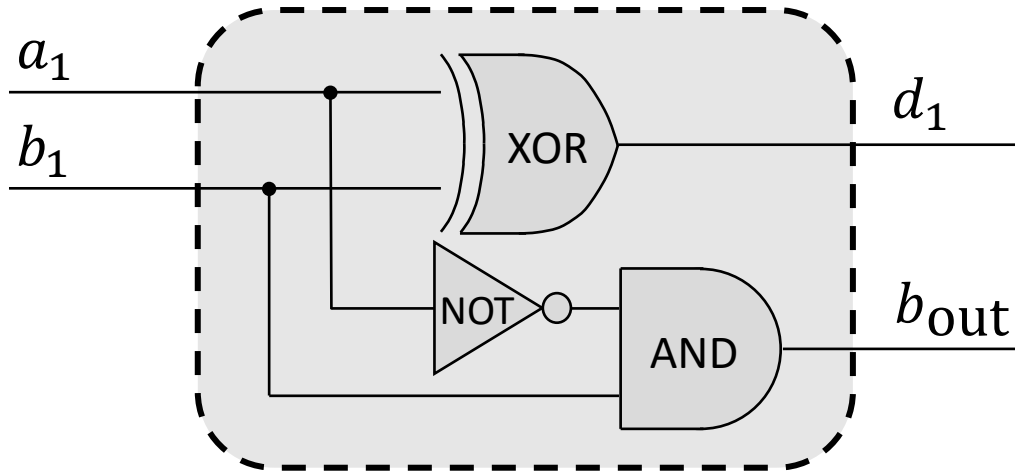
Half Subtractor Circuit:

- No carry-in from outside
- The Summation of Least Significant Bit d_1



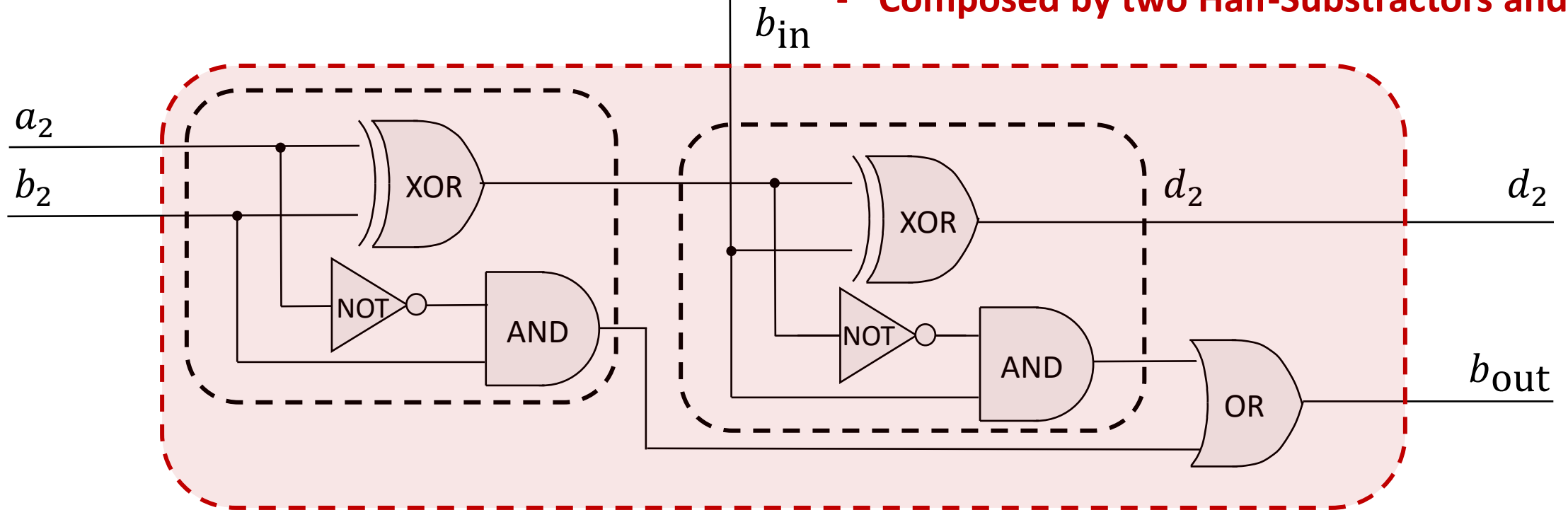
Half Subtractor Circuit:

- No carry-in from outside
- The Summation of Least Significant Bit d_1

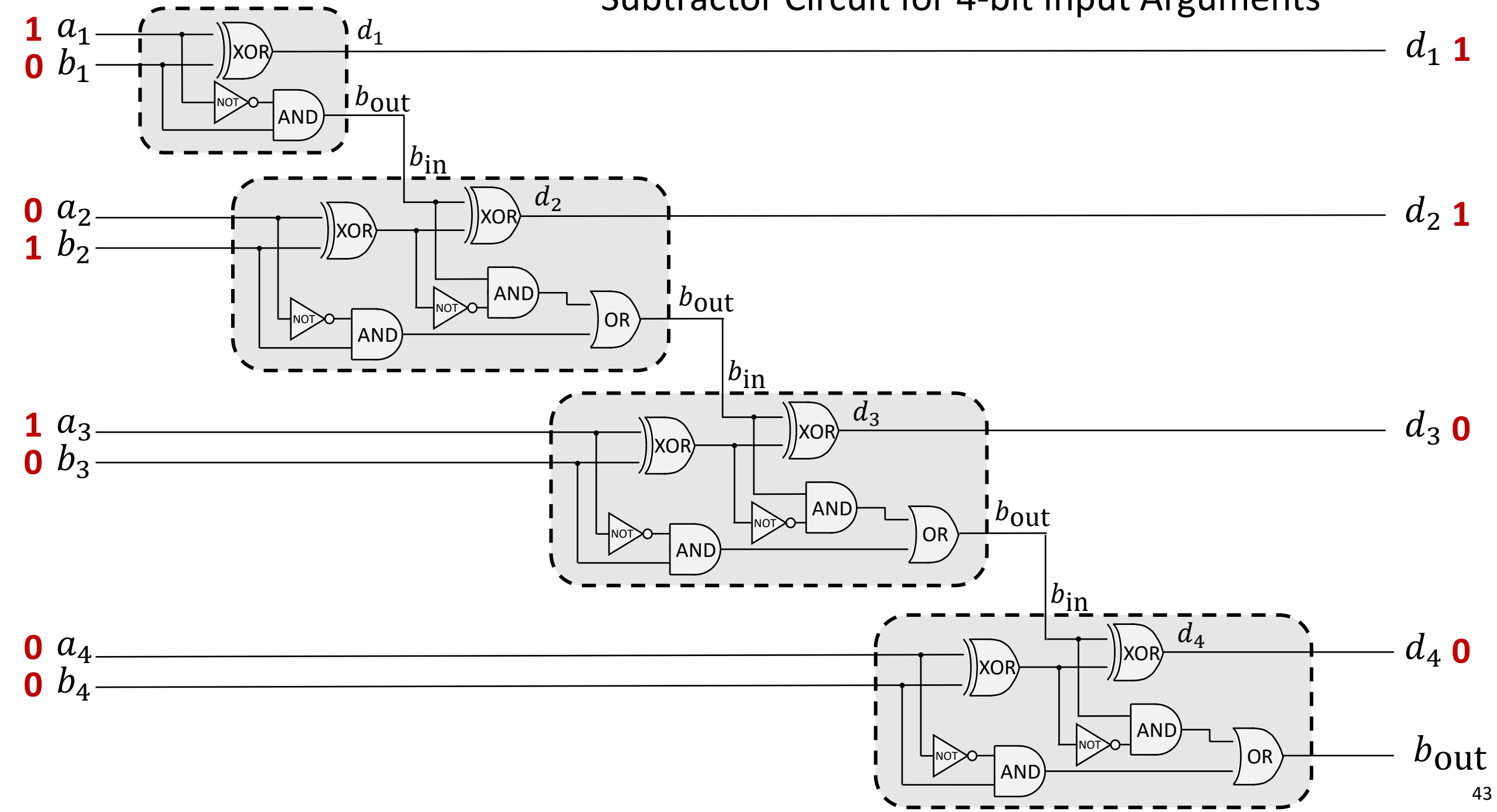


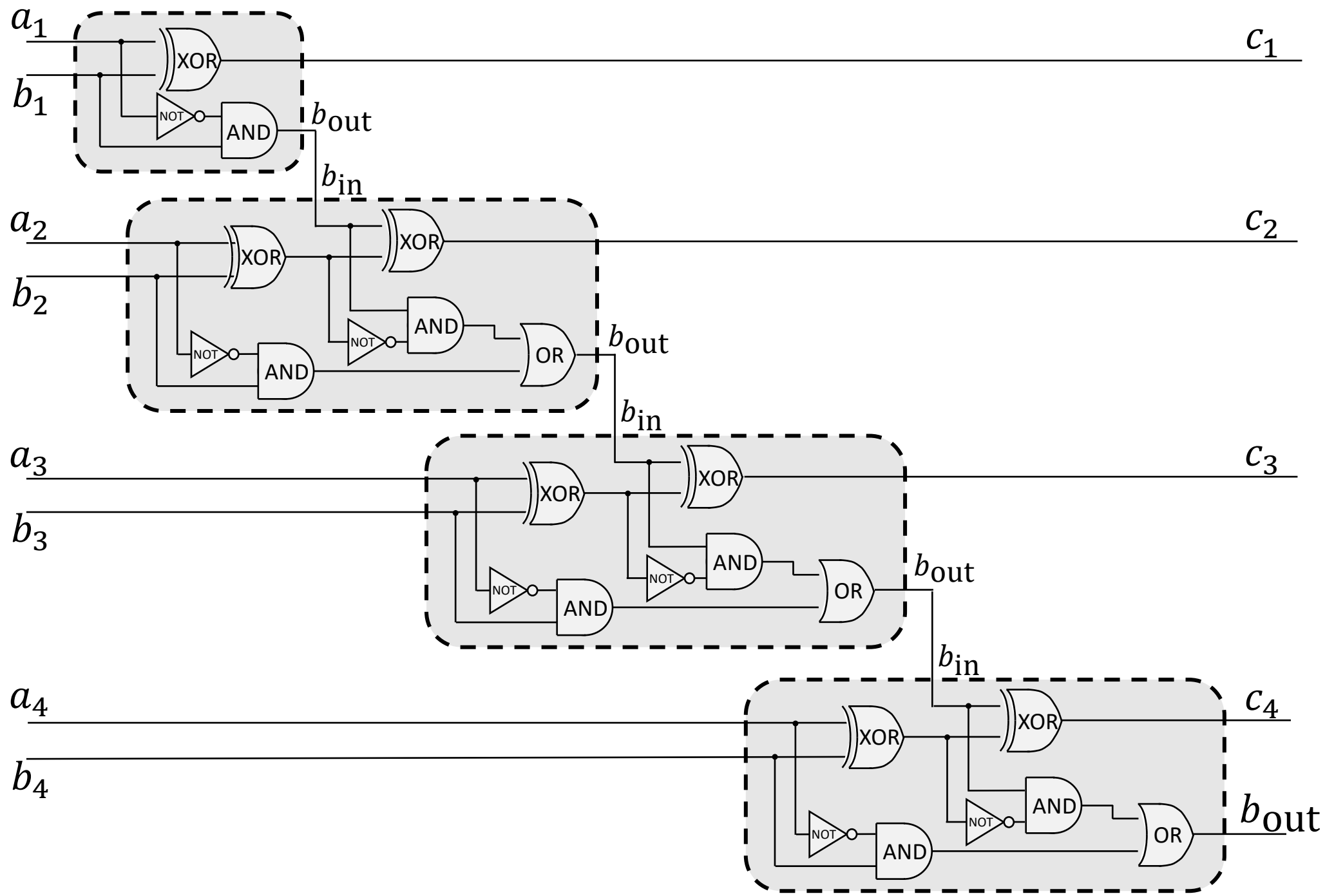
Full Subtractor Circuit:

- Takes borrow-in from outside
- Composed by two Half-Subtractors and OR



Subtractor Circuit for 4-bit Input Arguments





Multiplier Circuit for 2-by-2 Bits Input Arguments

