|  |  |  |
| --- | --- | --- |
| G:\nsu-logo.png  **North South University**  Department of Electrical & Computer Engineering    **LAB REPORT**  Course Name: **CSE231L**  Experiment No: 05     |  | | --- | | Experiment Name: **Binary Arithmetic** |   Experiment Date: 08 December 2020  Report Submission Date: 14 December 2020  Section: 05 | |
| Student Name:Md. Masruf Jaman | Score |
| Student ID: **1813625642** |  |
| Remarks: |

**LAB-05: Binary Arithmetic**

**Objectives:**

1. Understand the concept of binary addition and subtraction.
2. Learn about half and full binary adders.
3. Perform binary addition and subtraction using IC7483.
4. Understand the concept of BCD addition and implement a BCD adder using IC7483

**Apparatus:**

* 2 x IC 7483 4-bit binary adder
* 1 x IC 7486 quadruple 2-Input XOR gates
* 1 x IC 7408 quadruple 2-Input AND gates
* 1 x IC 4075, 3-Input OR gates

**Theory:**

Digital computers play out an assortment of data handling tasks. Among the capacities experienced are the different arithmetic operations. The most essential number crunching activity is the addition of two binary digits. This straightforward addition comprises of four potential rudimentary activities: 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, and 1 + 1 = 10. The initial three activities produce a sum of one digit, yet when both augend and numbers to be added bits are equivalent to 1, the binary sum comprises of two digits. The higher significant bit of this outcome is known as a carry. At the point when the augend and numbers to be added numbers contain more significant digits, the carry acquired from the addition of two bits is added to the next higher order pair of significant bits. A combinational circuit that performs the addition of two bits is called a half adder. One that performs the addition of three bits is a full adder. The names of the circuits come from the way that two half adders can be utilized to execute a full adder.

**Circuit Diagram:**

Attach the following Circuit Diagrams Screenshots.

(***After Discussion***)

1. **E.1 Report** Simulate a 4-bit adder in Logisim **using basic logic gates**. Provide a screenshot of the Logisim circuit schematic and truth table with your report
2. **E.1 Report** Construct the 4-bit adder-subtractor circuit of **Figure D.1.1**. Using **logic IC’s**
3. **E.2 Report** Draw the **IC** logic diagram for the 8-bit ripple-through-carry adder using logic **ICs**.
4. **E.3 Report** - Derive the circuit for the BCD adder

**Data Tables:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **M** | **A** | **B** | **C4** | **S4 S3 S2 S1** |
| 7 + 5 | 0 | 0111 | 0101 | 0 | 1100 |
| 4 + 6 | 0 | 0100 | 0110 | 0 | 1010 |
| 9 + 11 | 0 | 1001 | 1011 | 1 | 0100 |
| 15 + 15 | 0 | 1111 | 1111 | 1 | 1110 |
| 7 – 5 | 1 | 0111 | 0101 | 1 | 0010 |
| 4 – 6 | 1 | 0100 | 0110 | 1 | 1110 |
| 11 – 2 | 1 | 1011 | 0010 | 1 | 1001 |
| 15 – 15 | 1 | 1111 | 1111 | 1 | 0001 |

**F.1 Experimental Data (Binary Adder-Subtractor):**

**F.2 Experimental Data (Ripple-Through-Carry Adder):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operation** | **A** | **B** | **Overflow Carry** | **Sum** |
| 7 + 5 | 00000111 | 00000101 | 0 | 00001100 |
| 18 + 19 | 00010010 | 00010011 | 0 | 00100101 |
| 72 + 83 | 01001000 | 01010011 | 0 | 10011011 |
| 129 + 255 | 10000001 | 11111111 | 1 | 10000000 |

**Table F.2.1**

**F.3 Experimental Data (BCD Adder):**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Decimal Value |  | B | inary Sum | |  |  |  | BCD SUM |  |  |
| K | Z3 | Z2 | Z1 | Z0 | C | S3 | S2 | S1 | S0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 5 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 |
| 7 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 10 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| 11 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| 12 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| 13 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| 14 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 16 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| 17 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 18 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 |
| 19 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |

**Table F.3.1**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Operation** | **A**  (In Binary) | **B**  (In Binary) | **Overflow Carry** | **Sum**  (In Binary) | **SUM**  **(**Decimal**)** |
| 9 + 0 | 1001 | 0000 | 0 | 1001 | 9 |
| 9 + 1 | 1001 | 0001 | 1 | 0000 | 0 |
| 9 + 2 | 1001 | 0010 | 1 | 0001 | 1 |
| 9 + 3 | 1001 | 0011 | 1 | 0010 | 2 |
| 9 + 4 | 1001 | 0100 | 1 | 0011 | 3 |
| 9 + 5 | 1001 | 0101 | 1 | 0100 | 4 |
| 9 + 6 | 1001 | 0110 | 1 | 0101 | 5 |
| 9 + 7 | 1001 | 0111 | 1 | 0110 | 6 |
| 9 + 8 | 1001 | 1000 | 1 | 0111 | 7 |
| 9 + 9 | 1001 | 1001 | 1 | 1000 | 8 |

**Table F.3.2**

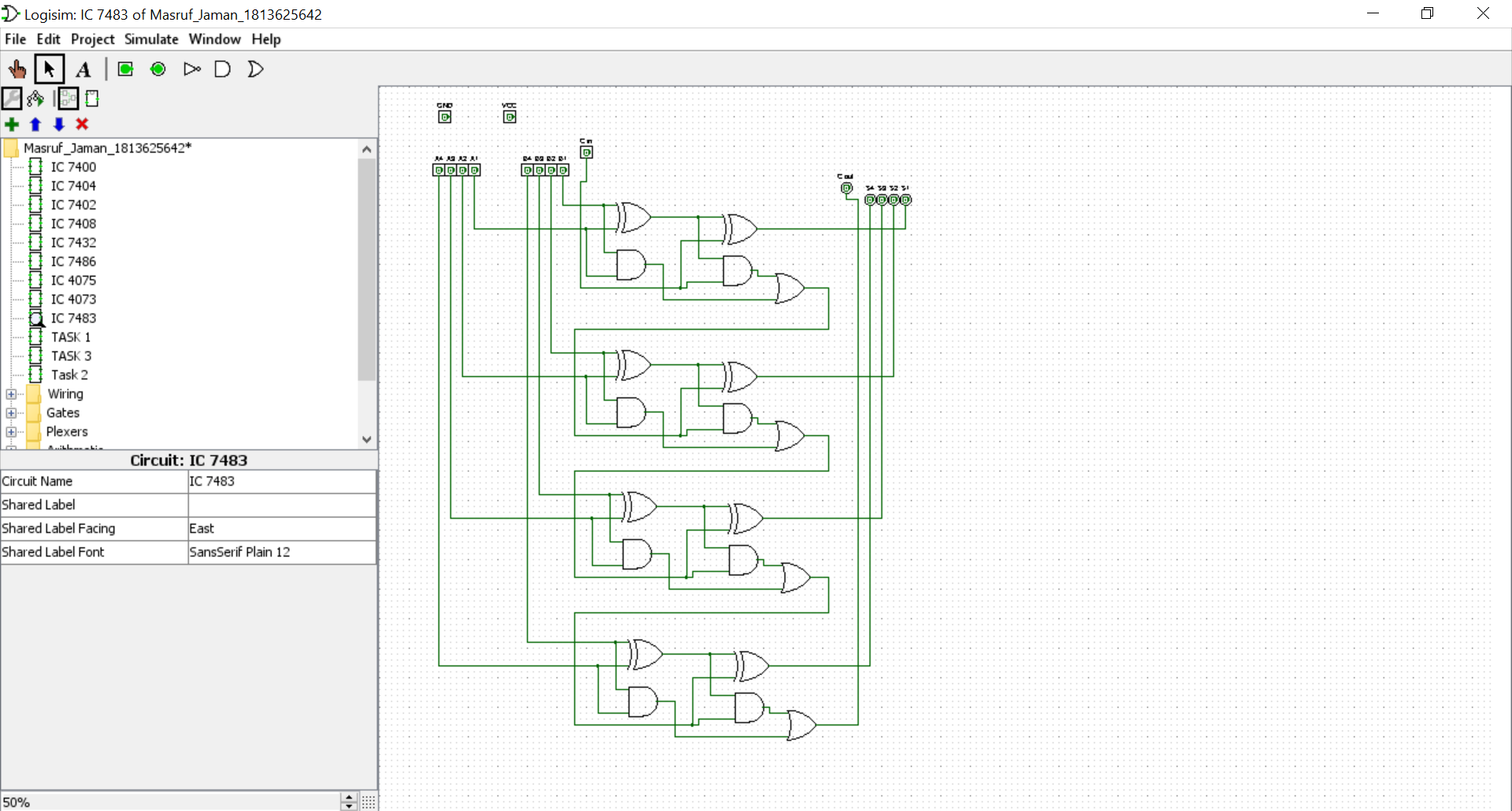
**Question Answering:**

* Comment on the use of the XOR gates and the M bit of the 4-bit adder-subtractor.

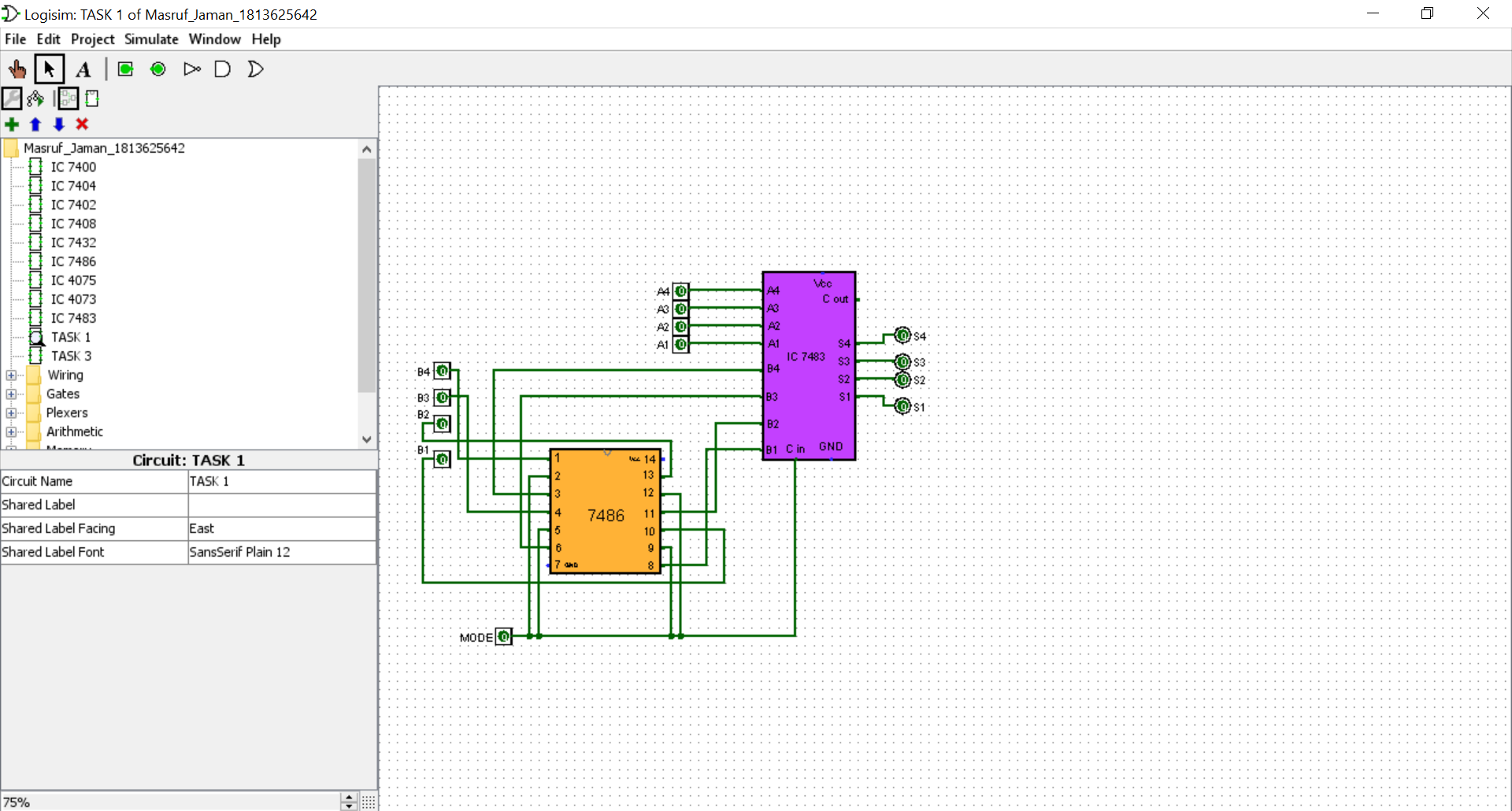
**Discussion:**

The name of our experiment was Binary Arithmetic. We covered the topic Half adder and Full adder in this experiment. we used an apparatus IC 7483. In first experiment we implement a 4-bit adder subtractor. And our second experiment was Ripple Through Carry Adder. We implement here an 8-bit Ripple Through Carry Adder. Third and last experiment was BCD Adder.

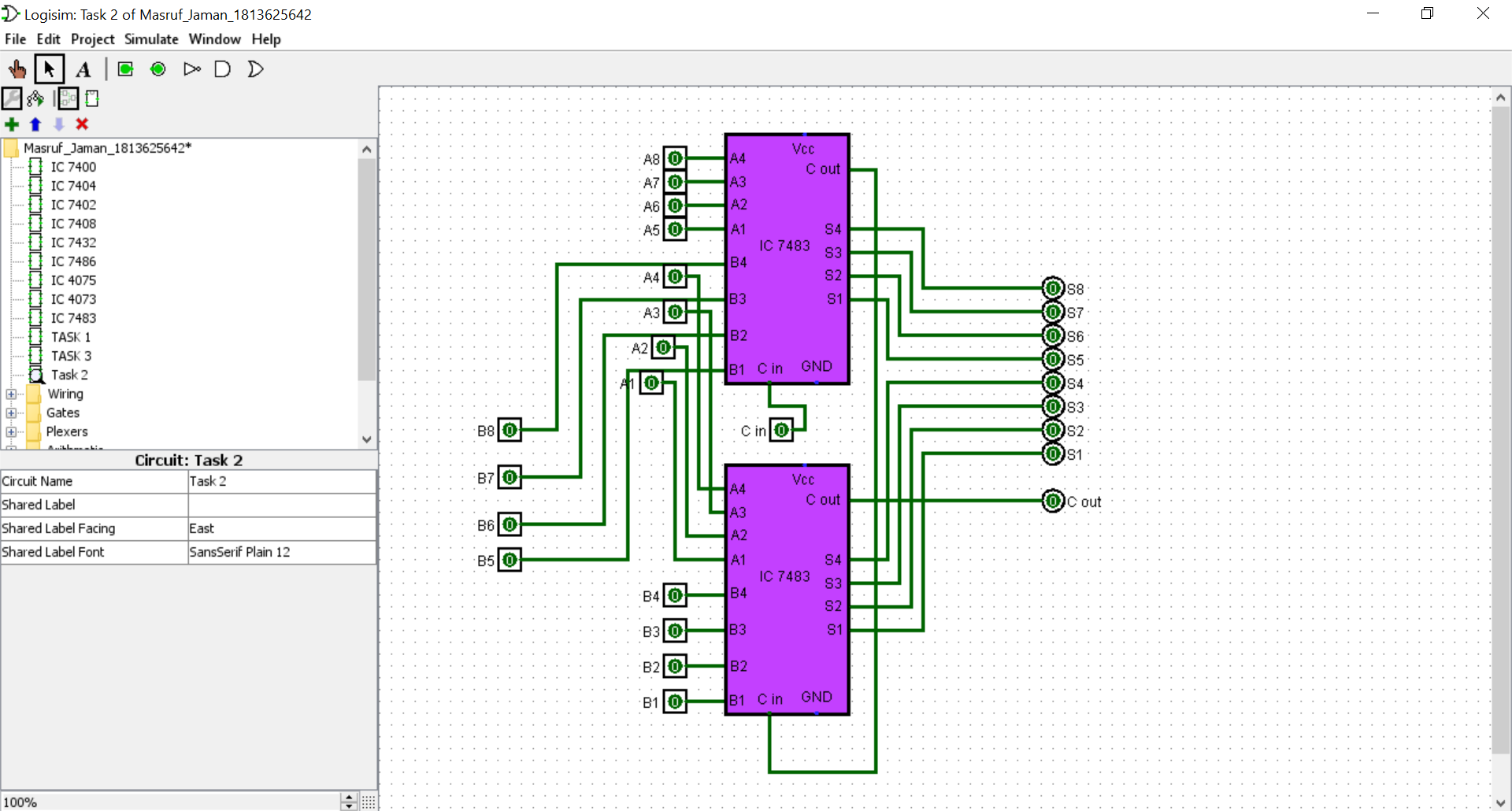
**No 1:**



**No 2:**

****

**No 3:**

****

**No 4:**

