**Design and Implementation of 4-bit Parallel Binary Adder**

**a) Truth Table of half and full Adder:**

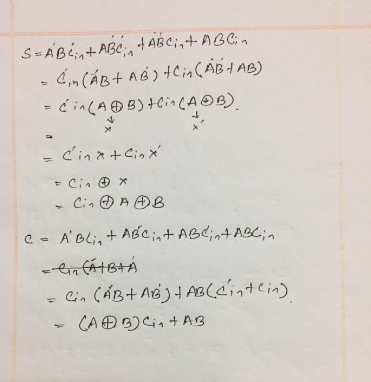
**Half Adder:**

|  |  |  |  |
| --- | --- | --- | --- |
| **X** | **Y** | **C** | **S** |
| **0** | **0** | **0** | **0** |
| **0** | **1** | **0** | **1** |
| **1** | **0** | **0** | **1** |
| **1** | **1** | **1** | **0** |

**Full Adder:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | Y | Z | C | S |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |

**b) Simplification of the boolean equation of Full Adder (By Hand. Attach Image to the report):**

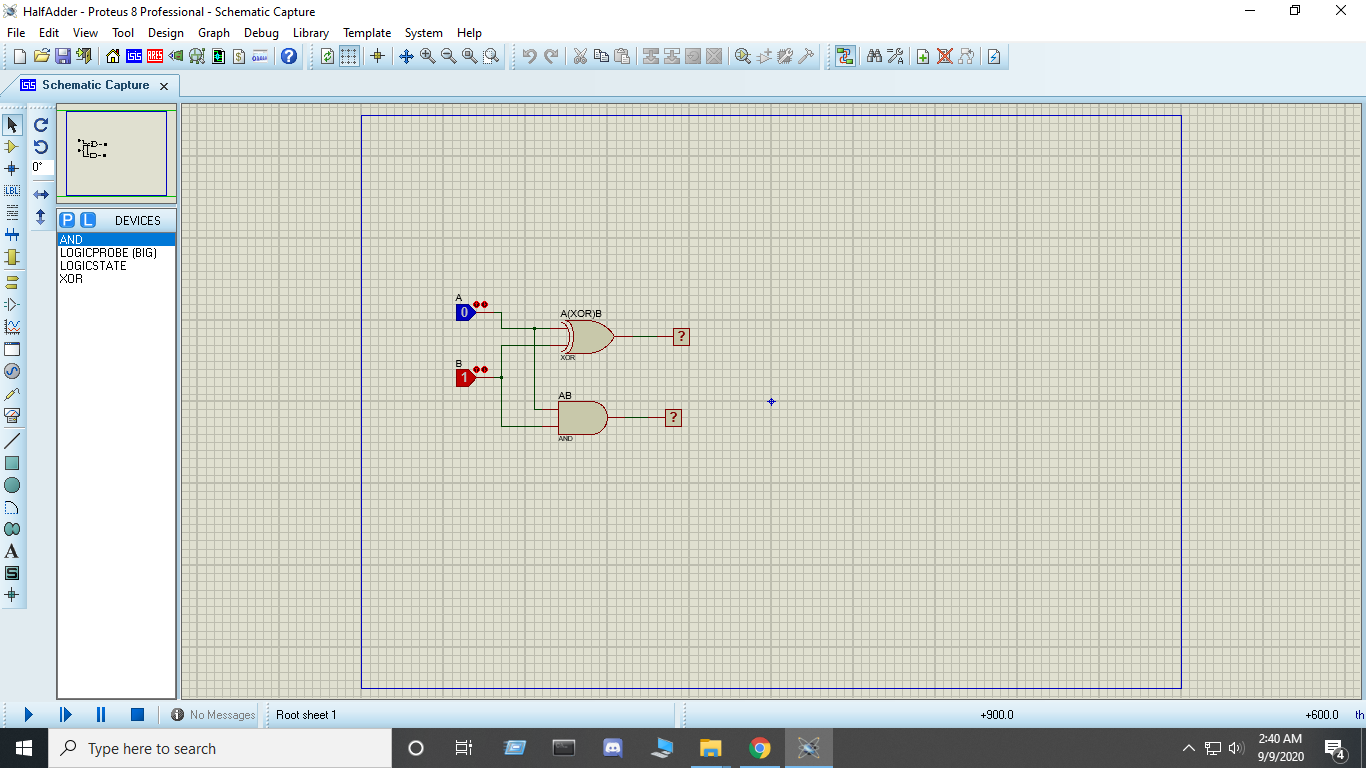


**c) Experimental Setup of 4-bit Parallel Adder cum Subtractor. (Step by Step description with a brief summary):**

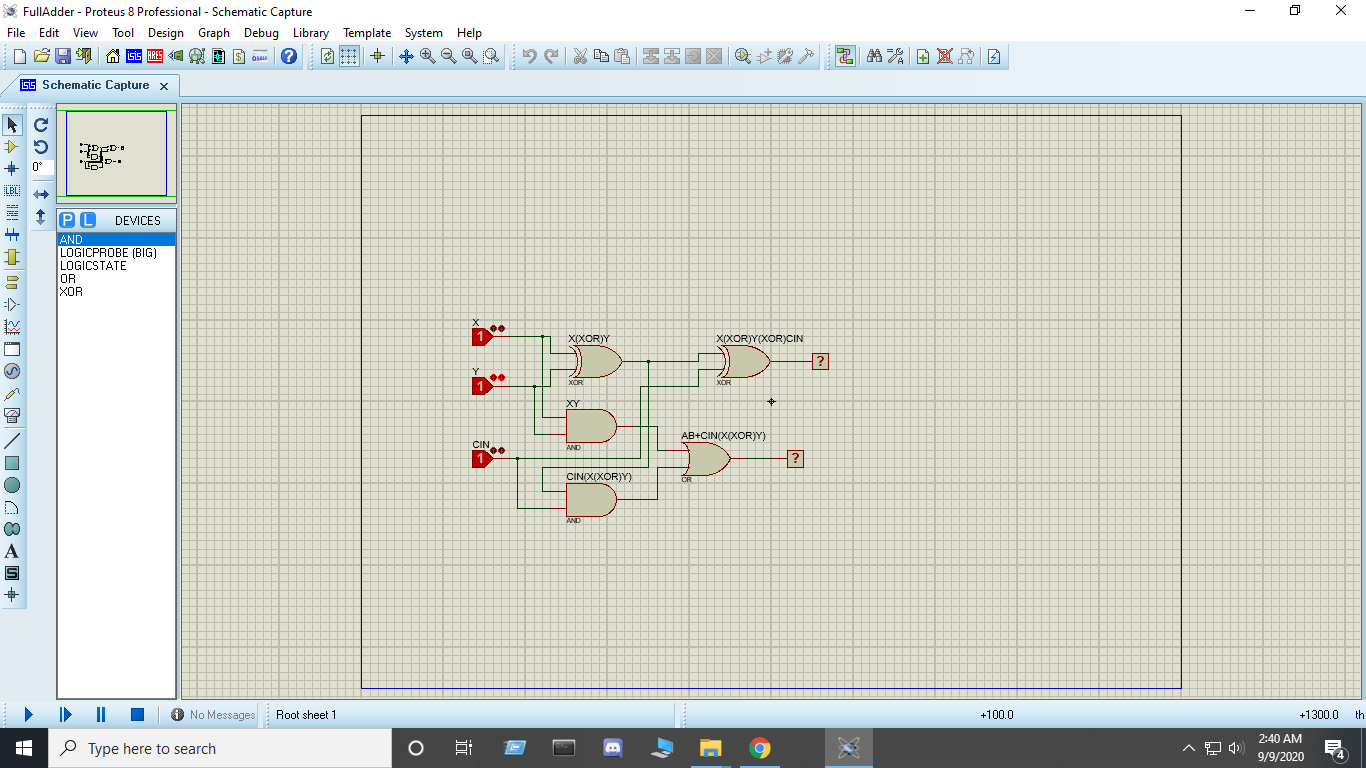
First I went to Parts and than simulator primitive. From there I picked the XOR gate. I also picked logic states and logic probe(big) and 7483 from parts. I placed 4 logic states to take four inputs of A0, A1, A2, A3. I placed four XOR gates and for its input, I took B0, B1, B2, B3 on one side. For the other input, I placed a logic state X and took input from there. I added C0 with X. And finally i added a logic probe in C4 to see my expected output.

**ScreenShots:**

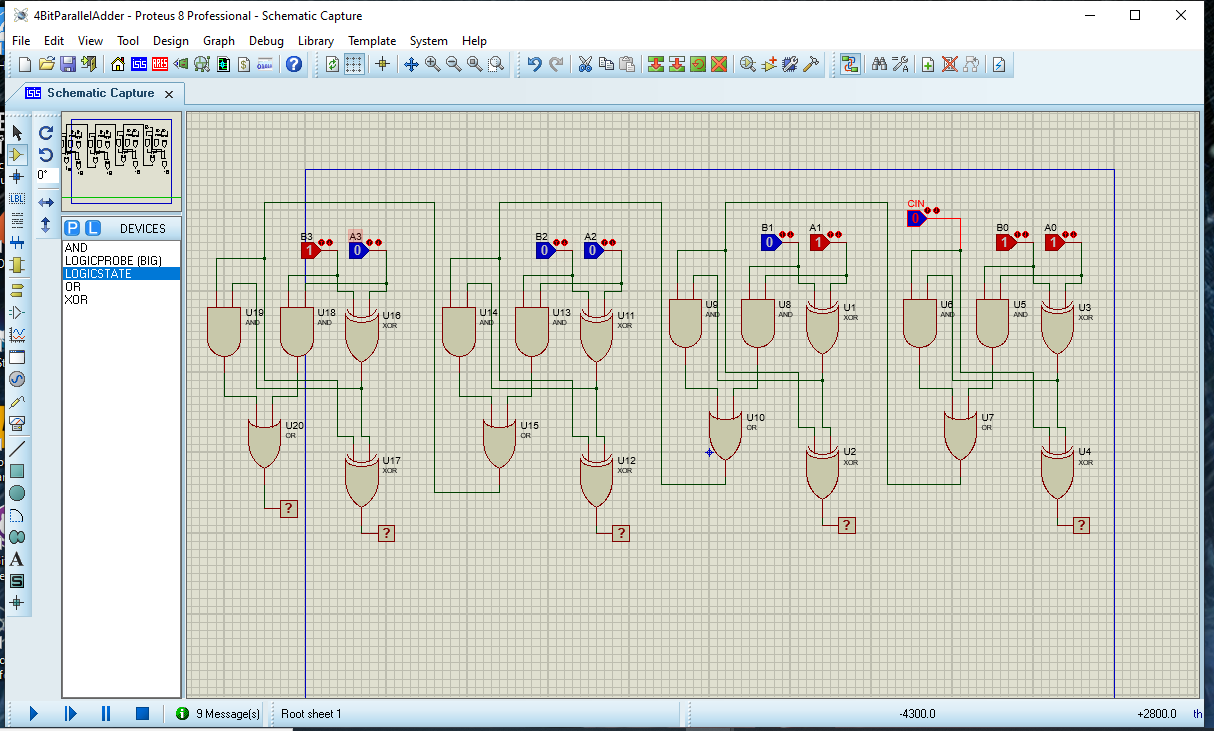
**a) Half Adder**

****

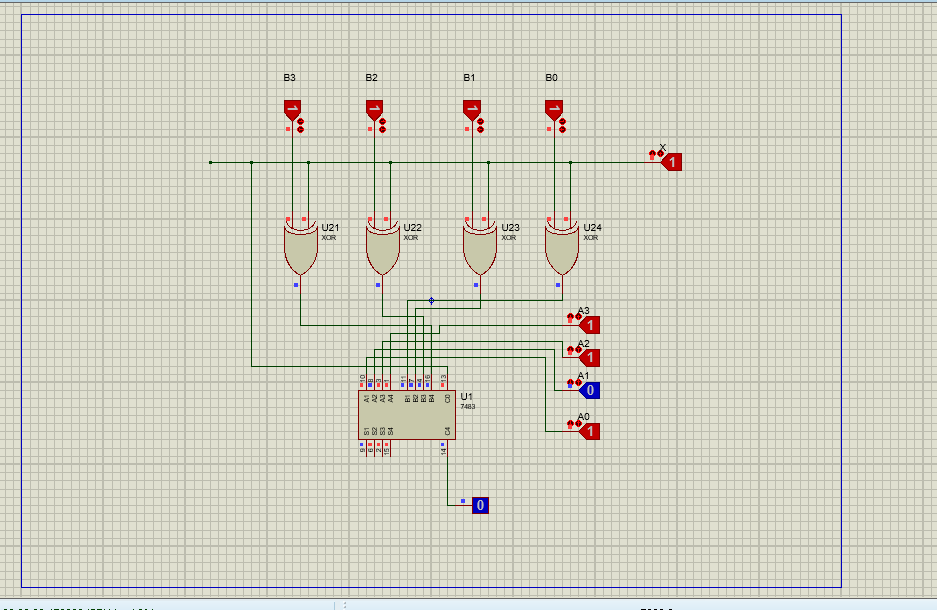
**b) Full Adder**

****

**c) 4 Bit Parallel Adder**

****

**d) 4 Bit Parallel Adder cum Subtractor**

****