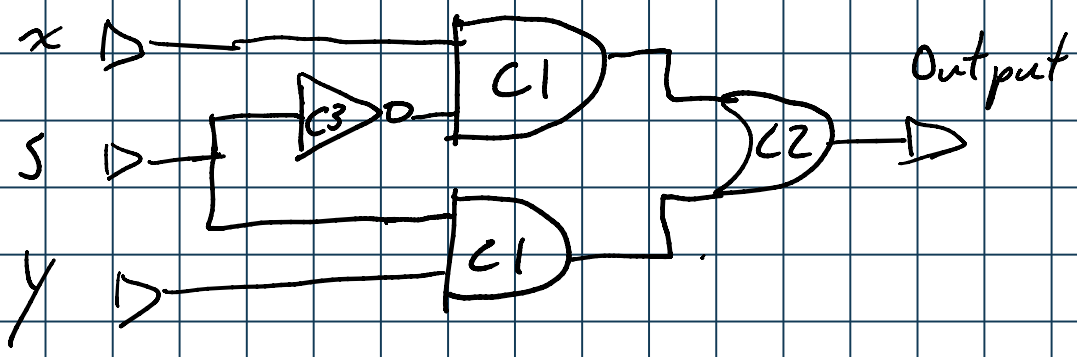
Lab 1

September 21, 2020 4:03 PM

Part I

1.



C1: 7408

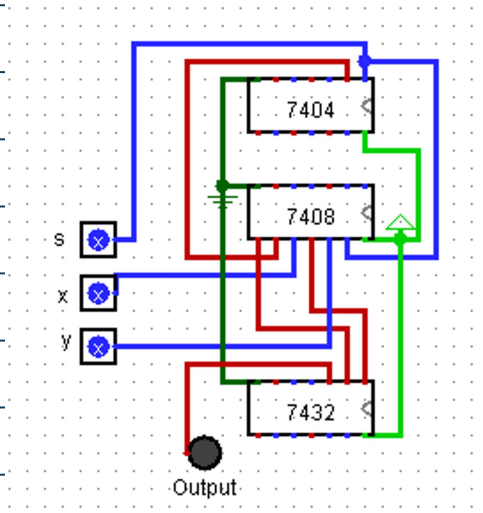
C2: 7432

C3: 7404

2. : $x\bar{s} + ys$

x	y	s	Out
0	0	0	0
0	0	1	0
0	1	0	0
1	0	0	1
0	1	1	1
1	1	0	1
1	0	1	0
1	1	1	1

3.



4. There is no cheaper design, as the multiplexer uses at least 1 of each operation (AND, OR, NOT), and each 7408 and 7432 chip contains 4 gates, we cannot go lower than the 1 chip, or we exclude the operation entirely.

1 NOT Gate < 6 NOT Gates on 7404
2 AND Gates < 4 AND Gates on 7408
1 OR Gate < 4 OR Gates on 7432

lower bound for # of chips is met.

Part II

6. $\bar{a}b\bar{c} + \bar{b}(\bar{a}\bar{c} + ad) + \bar{a}\bar{b}d$

$\bar{a}b\bar{c} + \bar{a}\bar{b}\bar{c} + ab\bar{d} + \bar{a}\bar{b}d$

$(\bar{a}\bar{c}) + (\bar{b}d)$

*recall $b + \bar{b} = 1$, $a + \bar{a} = 1$

∴ Yes there is a cheaper implementation

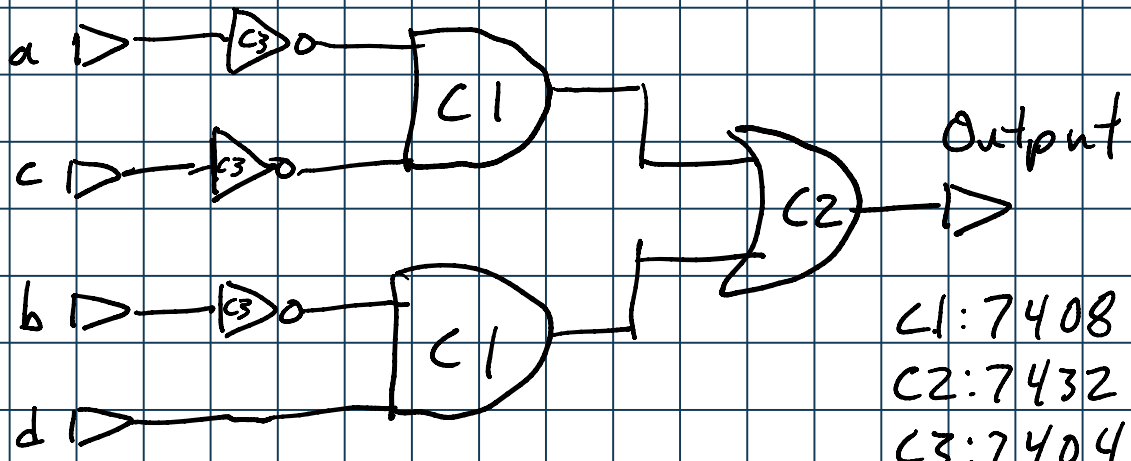
Original amount of gates to chips

AND: 8 -> 2 7408s
OR: 4 -> 1 7432
NOT: 3 -> 1 7404

Optimized amount of gates to chips

AND: 2 -> 1 7408
OR: 1 -> 1 7432
NOT: 3 -> 1 7404

1.

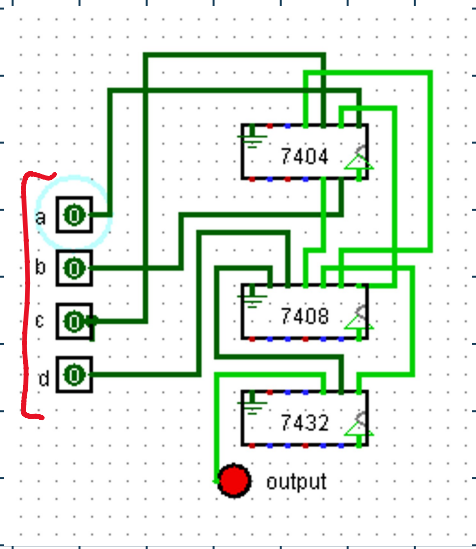


2.

a	b	c	d	Out
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	1	0	0	1
1	0	0	0	0
0	0	1	1	1
0	1	1	0	0
1	1	0	0	0
1	0	0	1	1
0	1	1	1	0
1	1	1	0	0
1	1	0	1	0
1	0	1	1	1
1	1	1	1	0

$\bar{a}\,\bar{c} + \overline{b\,d}$

3.

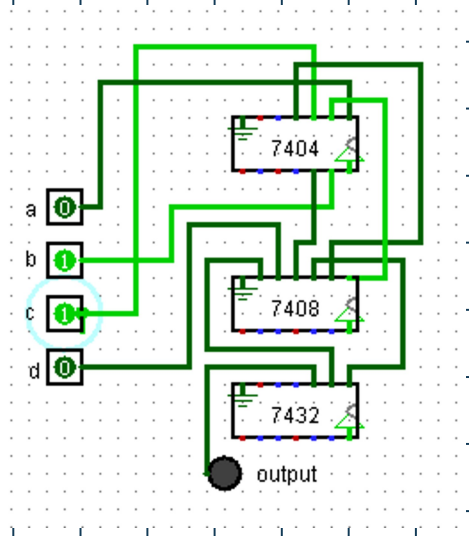


4.

5.

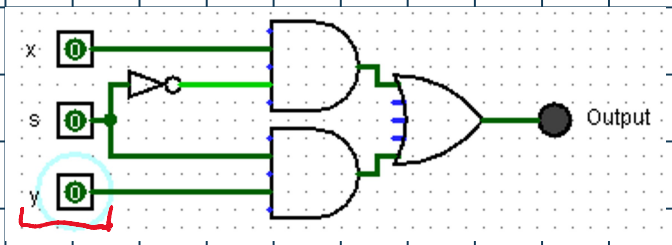
a	b	c	d	Out	
0	0	0	0	1	✓
0	0	0	1	1	✓
0	0	1	0	0	✓
0	1	0	0	1	✓
1	0	0	0	0	✓
0	0	1	1	1	✓
0	1	1	0	0	✓
1	1	0	0	0	✓
1	0	0	1	1	✓
0	1	1	1	0	✓
1	1	1	0	0	✓
1	1	0	1	0	✓
1	0	1	1	1	✓
1	1	1	1	0	✓

Example



Part III

1.



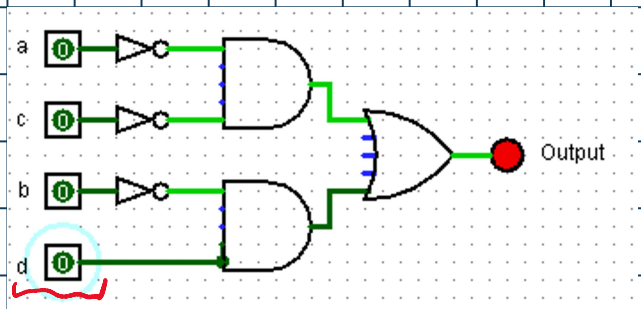
2.

3. The circuit behaves as expected. The output LED only lights up when (x and $\bar{s}$) or (s and y). The truth table from Part I, #2 is still true.

Part IV

Part IV

1.



2.

3. The circuit behaves as expected. The output LED only lights up when ($\bar{a}$ and $\bar{c}$) or ($\bar{b}$ and d). The truth table from Part I, #2 is still true.