HW 3 - JA - Karnaugh Maps

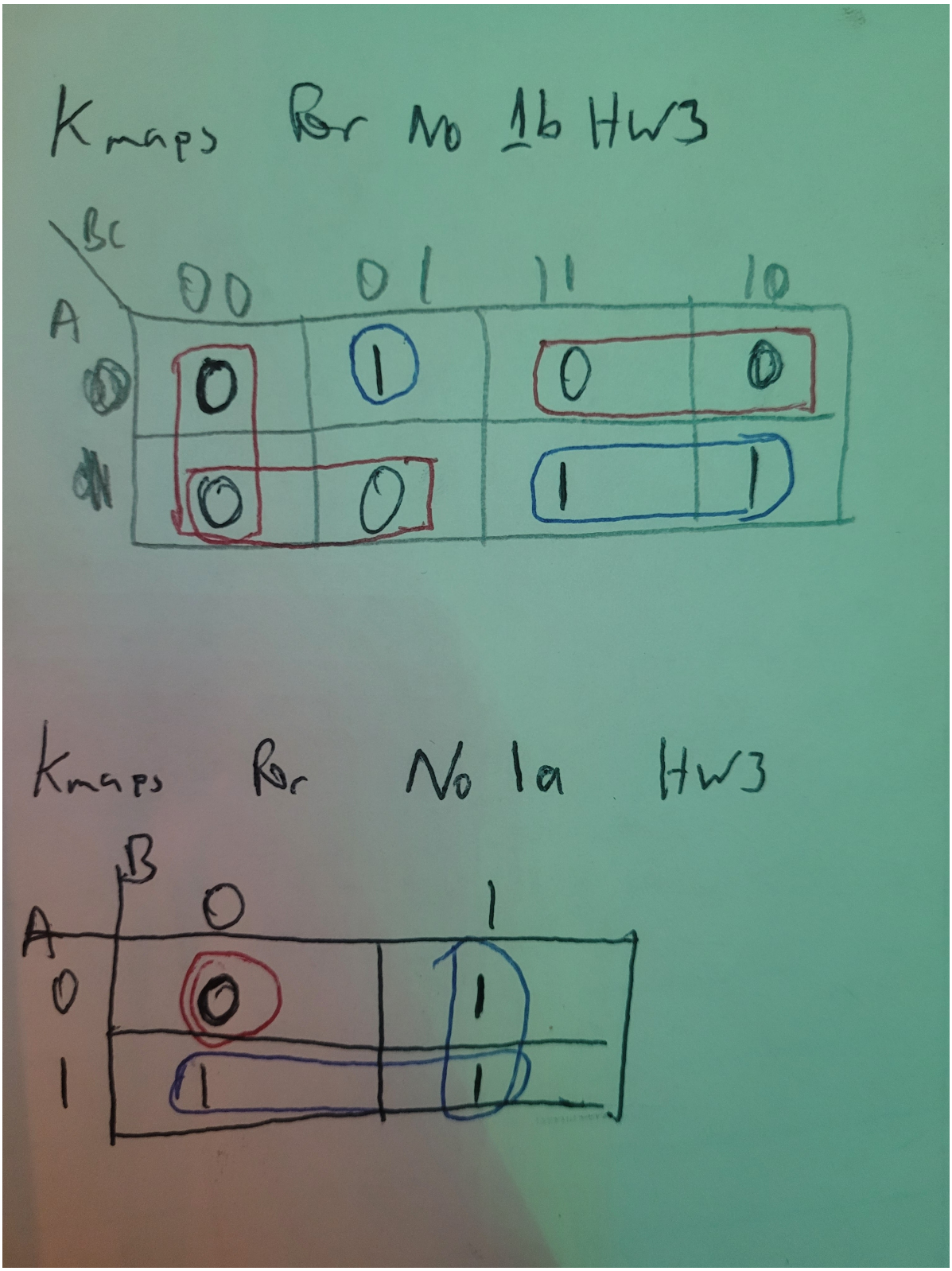
Owner	John Akujobi
Type	Homework
Created time	@September 5, 2023 10:06 PM
Due Date	@September 13, 2023
Status	In progress

homework3.docx

homework3.pdf

Q1

1. Calculate the minimal sum-of-products (SOP) and product-of-sums (POS) using Karnaugh Maps for the truth tables


Kmaps For No 1b HW3
BC
00
01
11
10
A
0
1
0
1
0
0
0
0
1
1
Kmaps For No 1a HW3
B
0
1
A
0
1
0
1
1
1

Q1 A

1.

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

POS

- Y = A+B
- Cost = 3

SOP

- Y = A + B

Q1 B

A	B	C	Y	
0	0	0	0	
0	0	1	1	A’.B’.C
0	1	0	0	
0	1	1	0	
1	0	0	0	
1	0	1	0	
1	1	0	1	A.B.C’
1	1	1	1	A.B.C

SOP:

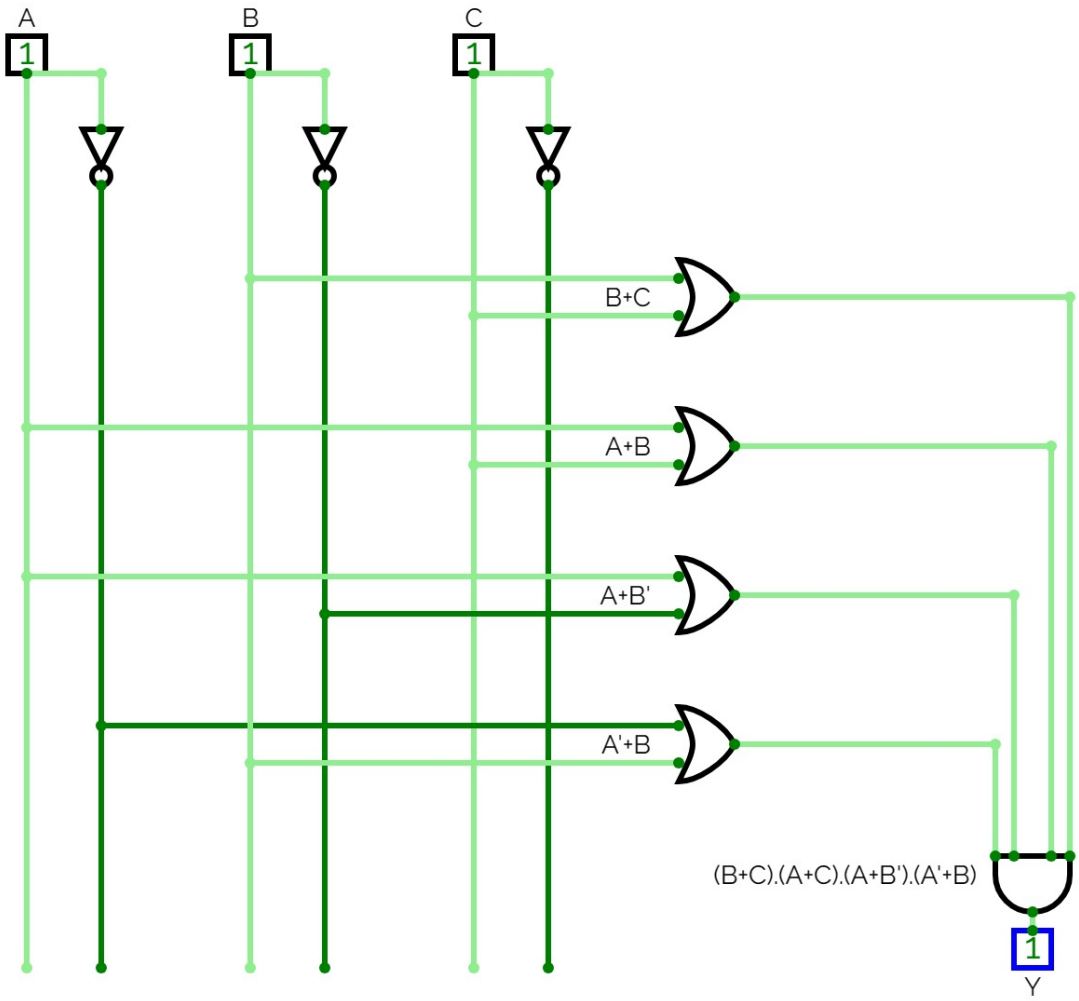
- Y = A.B + (A’.B’.C)
- Cost = 11

POS:

- Y = (B+C).(A+C).(A+B’).(A’+B)

- Cost = 17

Karnaugh Maps HW 1b.cv



POS Circuit

c. What is the cost of the two circuits? State which is cheaper, SOP or POS.

Q1a

- `SOP = 3`

- `POS = 3`
- `They were the same`

Q1b

- SOP =
- POS =
- was cheaper tha

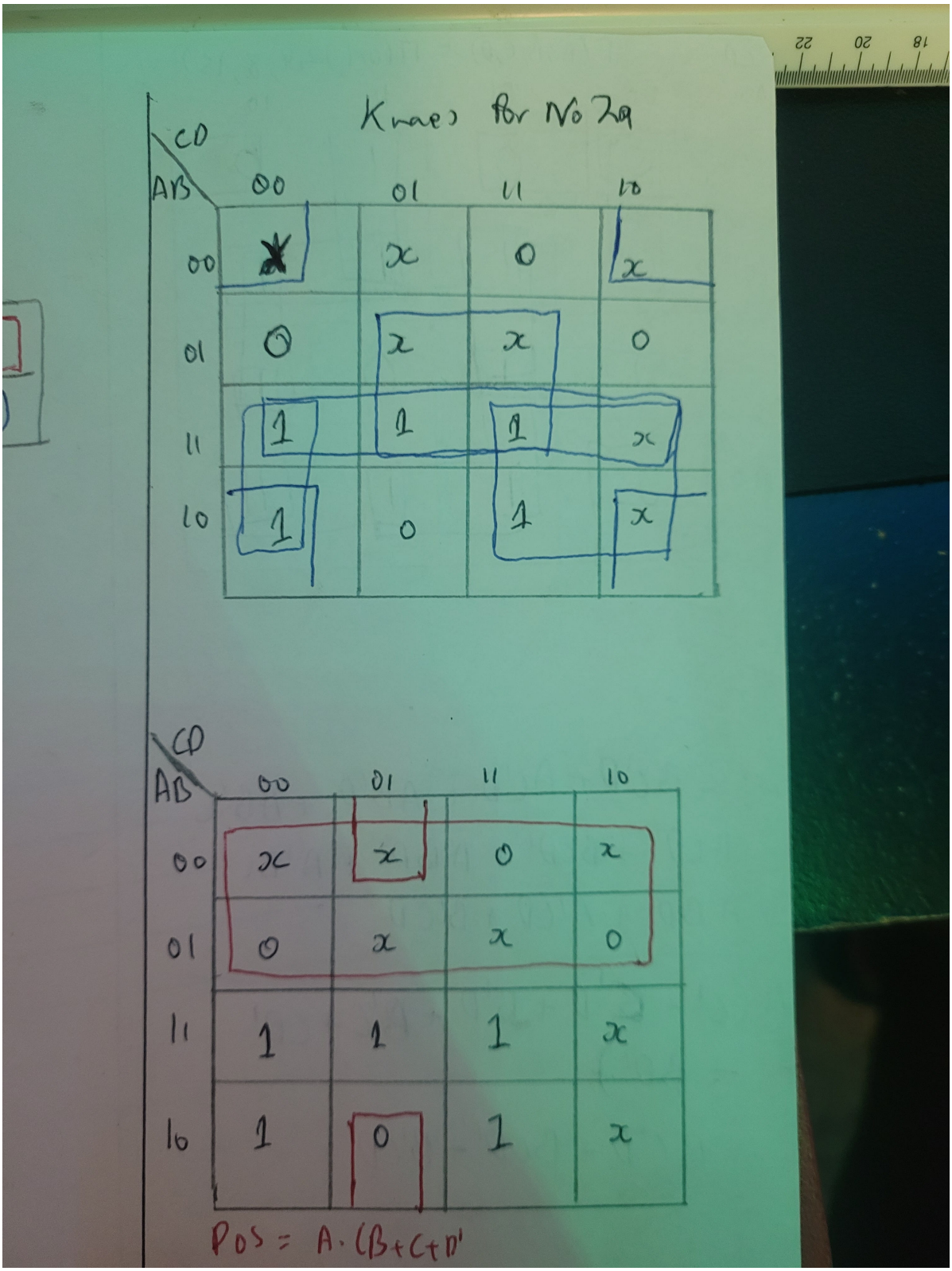
Q 2

2. Find minimal Boolean equations for the truth table below using both SOP and POS forms using K-maps.

A	B	C	D	Y
0	0	0	0	X
0	0	0	1	X
0	0	1	0	X
0	0	1	1	0
0	1	0	0	0
0	1	0	1	X
0	1	1	0	0
0	1	1	1	X
1	0	0	0	1
1	0	0	1	0
1	0	1	0	X
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	X

A	B	C	D	Y
1	1	1	1	1

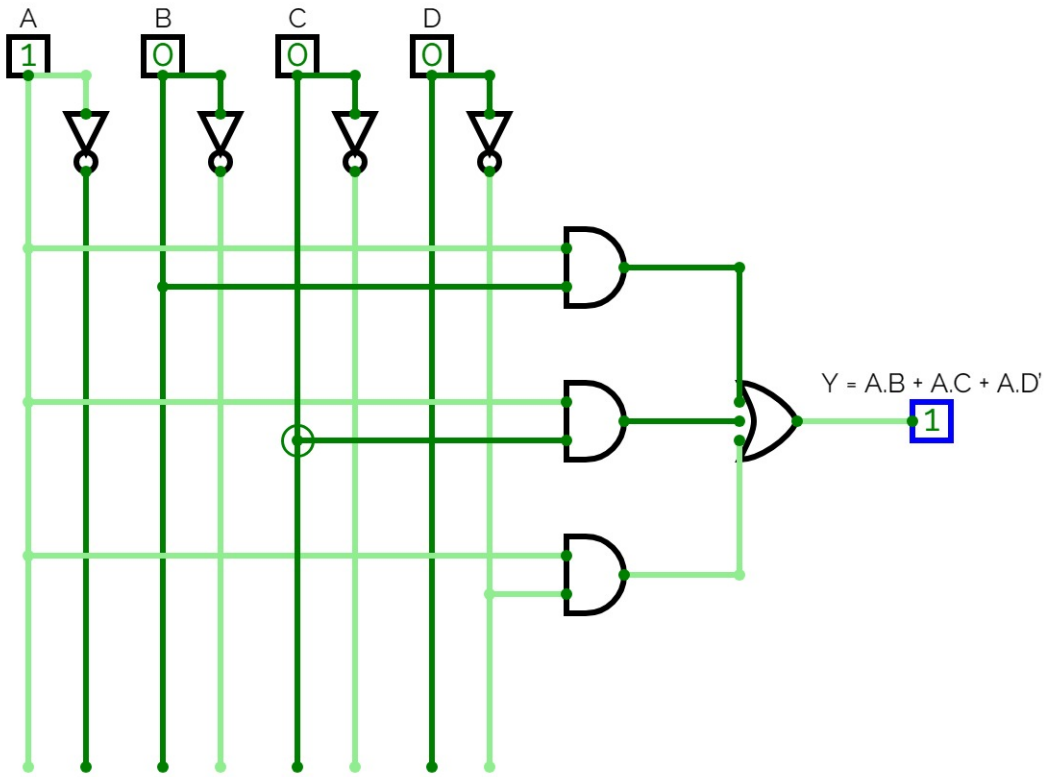
K-Maps



SOP

- `Y = AB + AC + AD' + BD + B'D'`
- `Y = AB + AC + AD' + ~~BD + B'D'~~`
- `Y = A.B + A.C + A.D'`
- Cost = 14

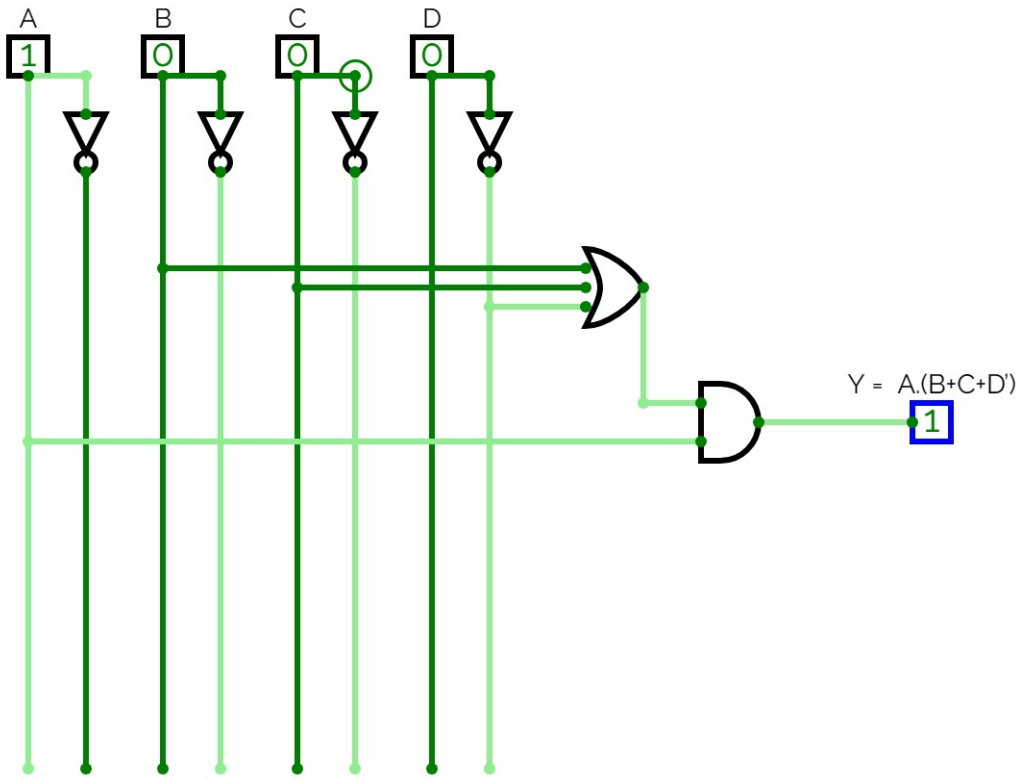
SOP Q2-1 HW3.cv



POS

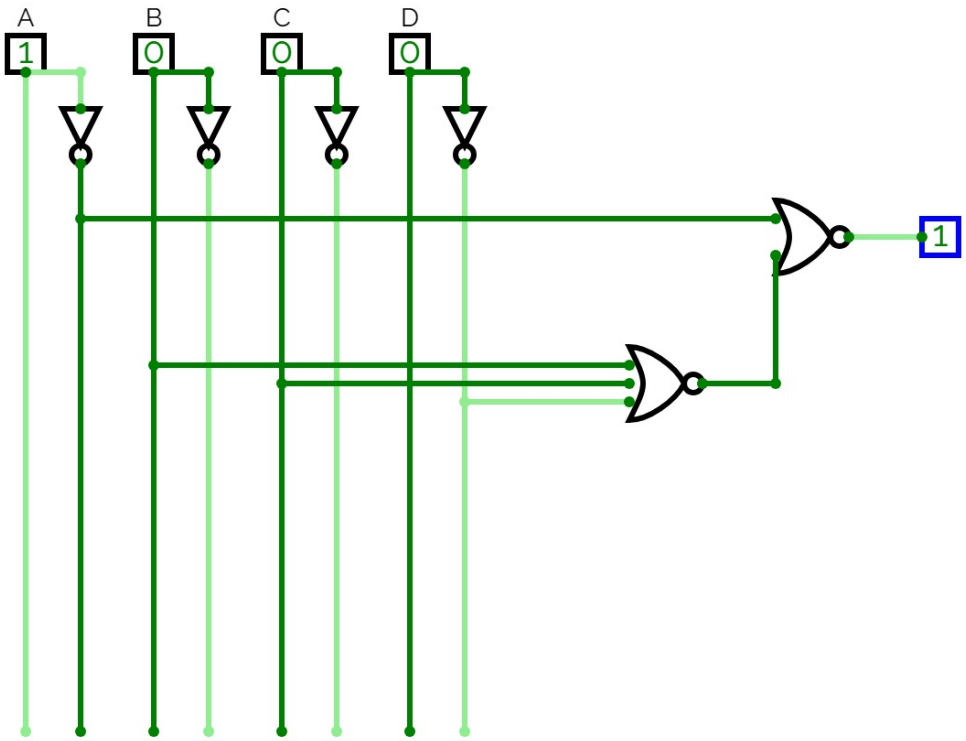
- `Y = A.(B+C+D')`

POS Q2-1 HW3.cv



1. Determine which circuit is cheaper
    - `The POS circuit is cheaper`
2. Draw the circuit for the **cheapest implementation** using only NAND gates (if sum-of-products form) or NOR gates (if product-of-sums form). You may also use inverters if needed.

NAND Gates of Q2A HW 3.cv



Using NOR gates

Q3

Calculate the minimal SOP and POS for the following function using K-maps:

1. Write two sentences on your solution to SOP; is it unique?

   F(A,B,C,D) = Π (0,1,2,4,8,15)

   ### POS:

   - `(A+B+C) . (A+B+D) . (A+C+D) . (B+C+D) . (A’+B’+C’+D’)`

   ### SOP:

   - `AC’ + BC’D + AC’D + AB’D + AB’C + ACD’ + BCD’ + ABD’ + A’BC + A’BD + A’CD + B’CD`
   - It looks like it can be simplified by bring groups together, but after trying multiple times, it looked like it could not be simplified more.

- Each of the groups had only one input that was inversed. Eg C in BC’D