PROJECT REPORT

IIIT NAGPUR

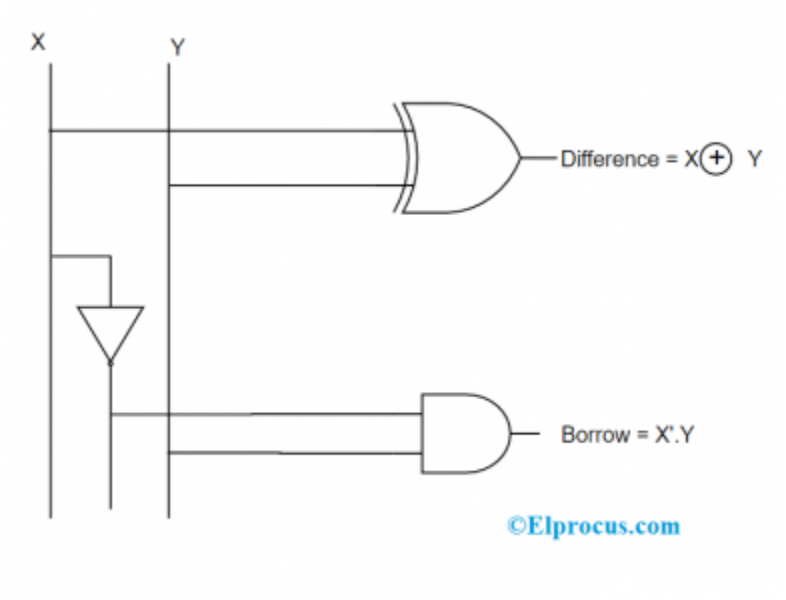
VISHAL PATEL (BT18ECE042)

TOPIC = 2BIT-BINARY SUBTRACTOR

SUBJECT -CMOS

Half Subtractor : Circuit

LOGIC GATE



Truth Table

First Bit	Second Bit	Difference (EX-OR Out)	Borrow (NAND Out)
0	0	0	0

1	0	1	0
0	1	1	1
1	1	0	0

Applications of Half Subtractor

The applications of half subtractor include the following.

- Half subtractor is used to reduce the force of audio or radio signals
- It can be used in amplifier to reduce the sound distortion
- Half subtractor is used in ALU
- It can be used to increase and decrease operators and also calculates the addresses
- Half subtractor is used to subtract the least significant column numbers. For the subtraction of multi-digit numbers, it can be used for the LSB.

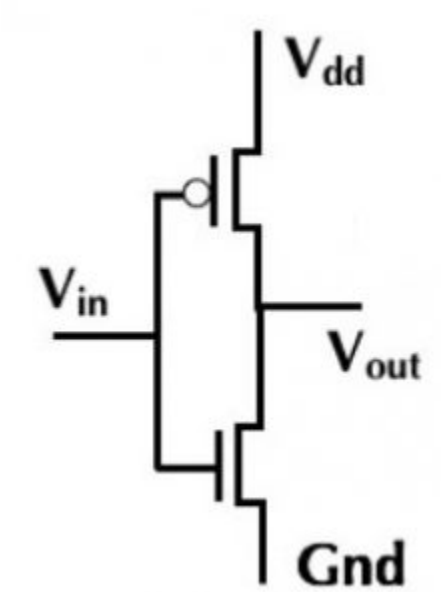
Require only 3 gates

1 NOT(inverter)

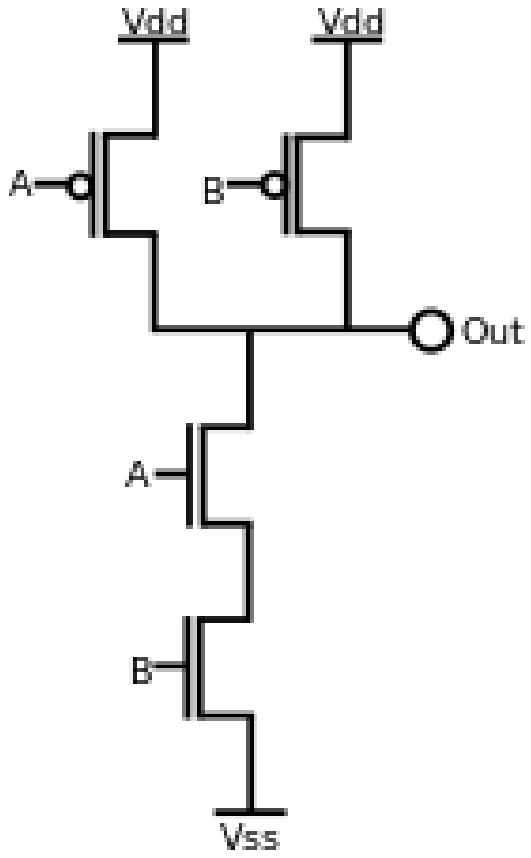
2 AND(NAND+INVERTER)

3 XOR

1:INVERTER



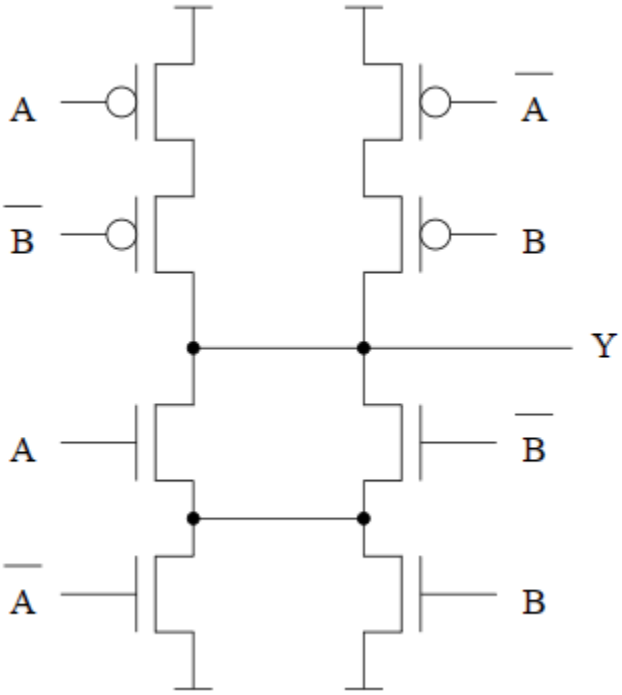
2: AND (NAND+INVERTER)



Output of NAND given to inverter and you get AND logic ;

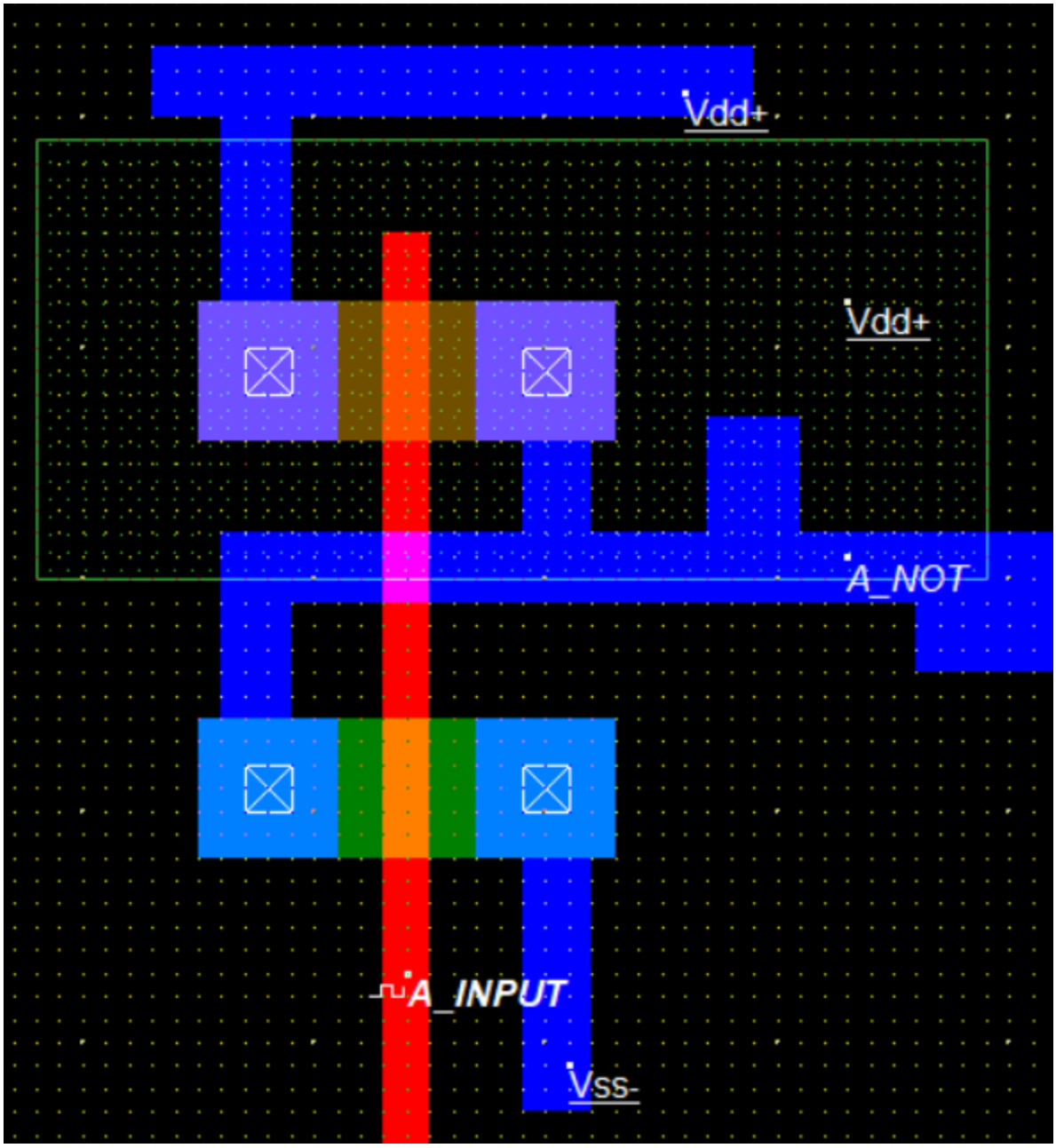
DIFFERENCE =XOR OF A AND B;
BORROW =NOT A AND B

3: XOR Gate

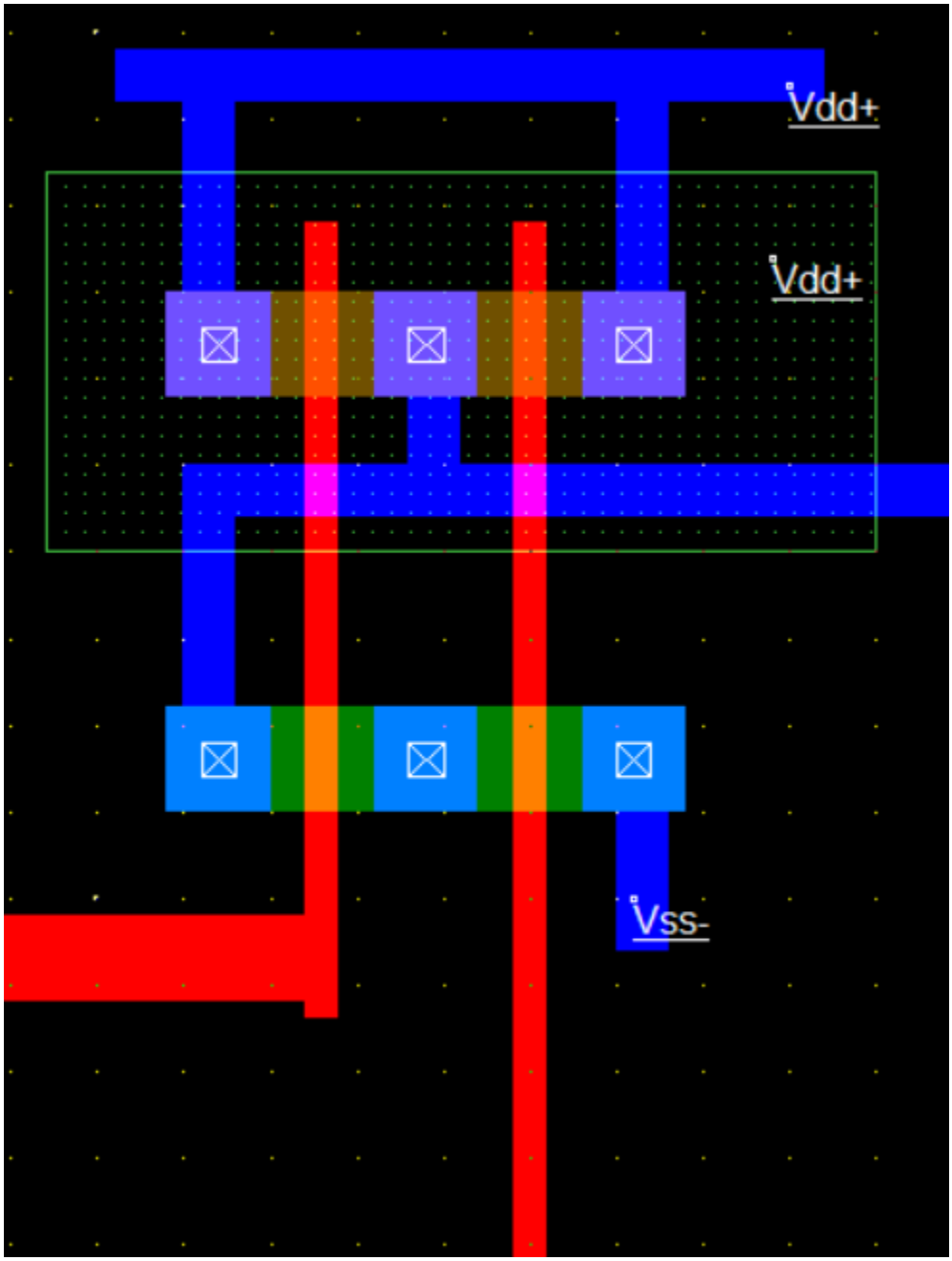


MICROWIND DESIGN

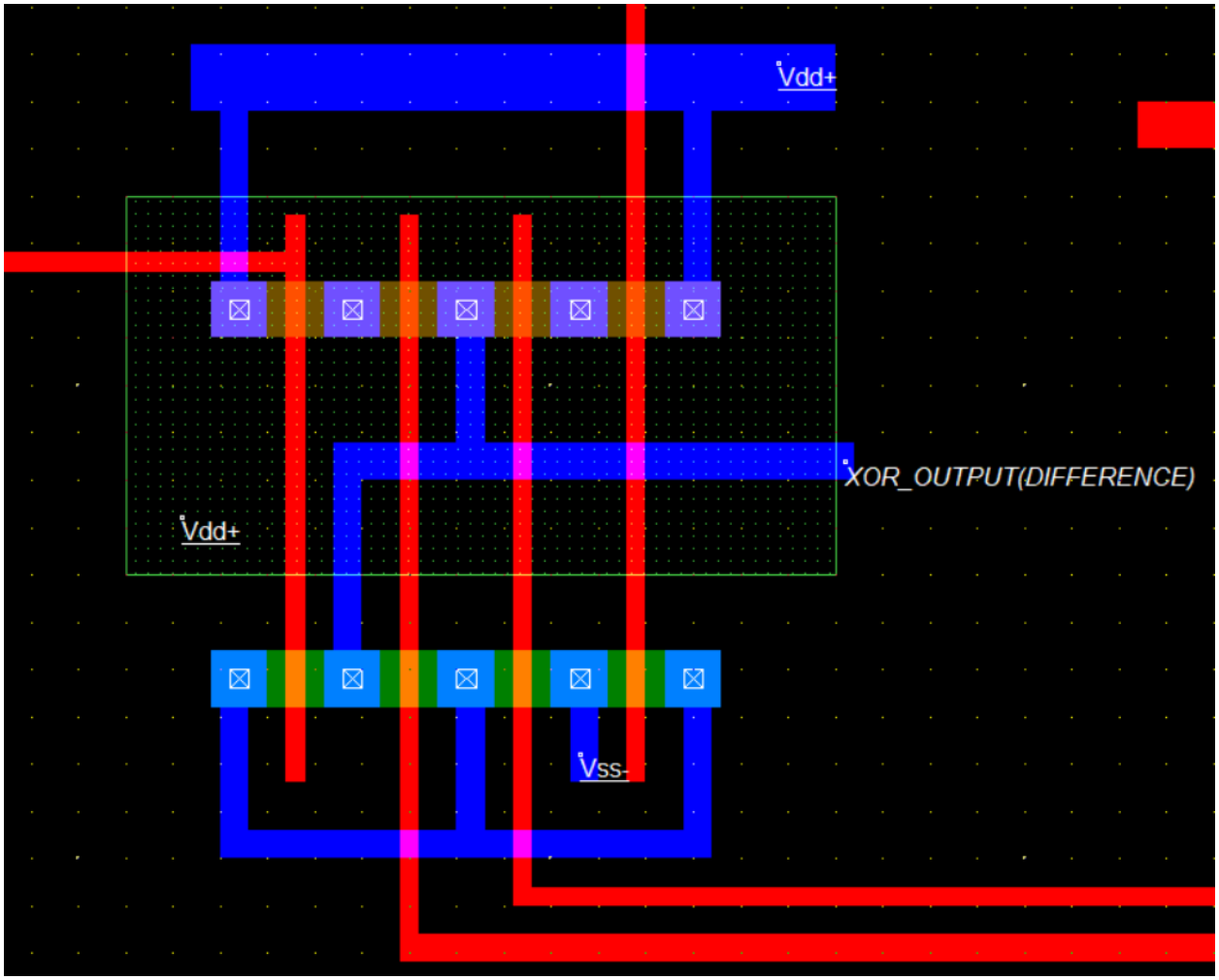
INVERTER DESIGN
FOR IMPLEMENT
AND and XOR LOGIC



INVERTER



NAND DESIGN



XOR DESIGN

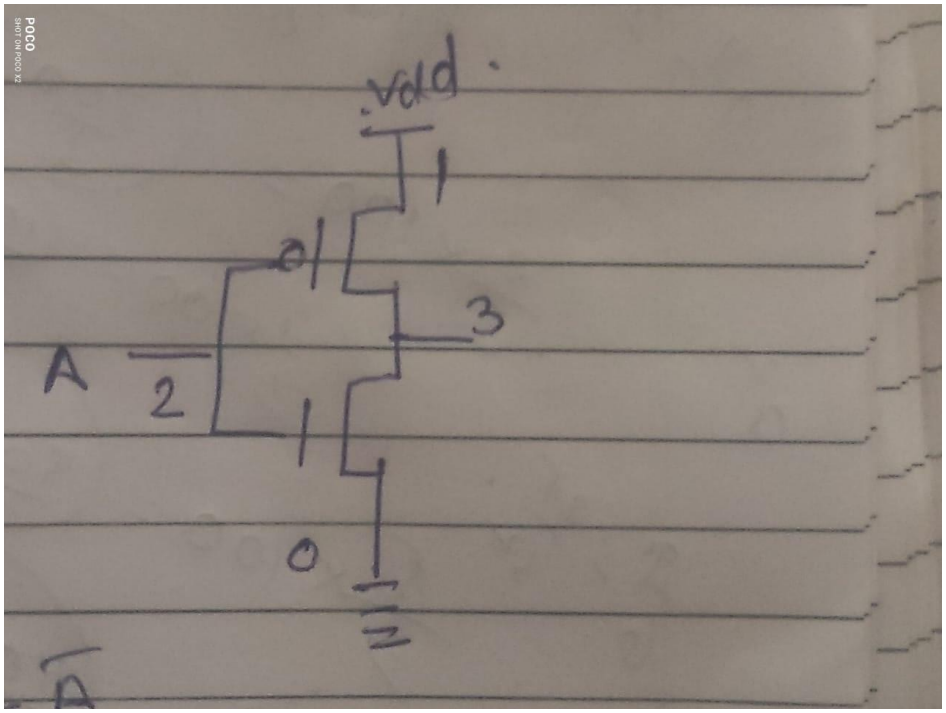
PSPICE CODE:

3 PART

1 INVERTER
2 NAND
3 XOR

DESIGN AS A SUB CIRCUIT

1 . IMAGE 1 FOR INVERTER



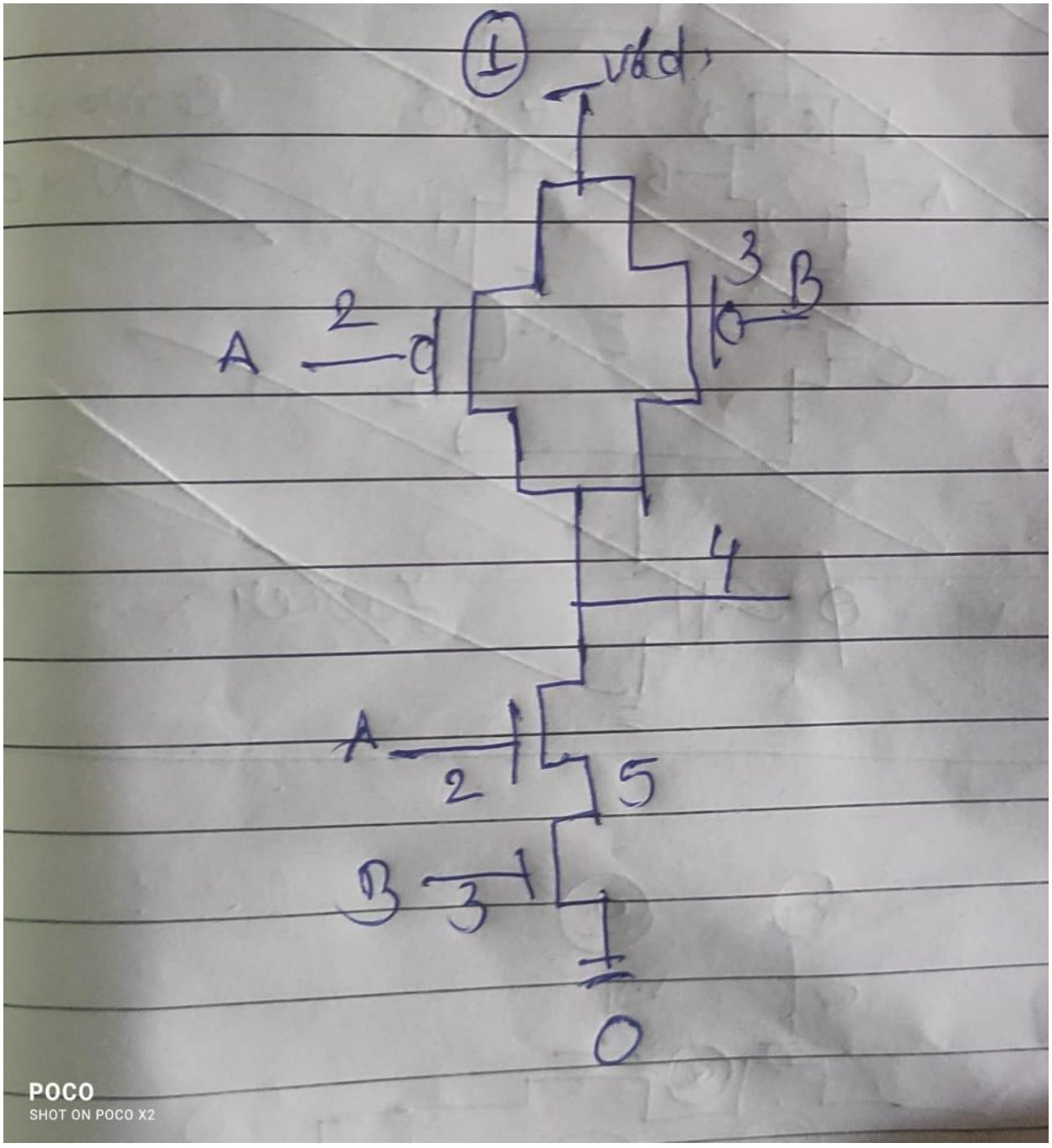
CODE OF INVERTER :

Take above image as reference to write below code:

```
.subckt inverter 2 3 1
mn 3 2 0 0 mod1 l=1u w=100u
mp 3 2 1 1 mod2 l=1u w=100u
.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends
```

2 =from where input have taken;
3 =return output to which node;
1 =location of vdd;

2. NAND

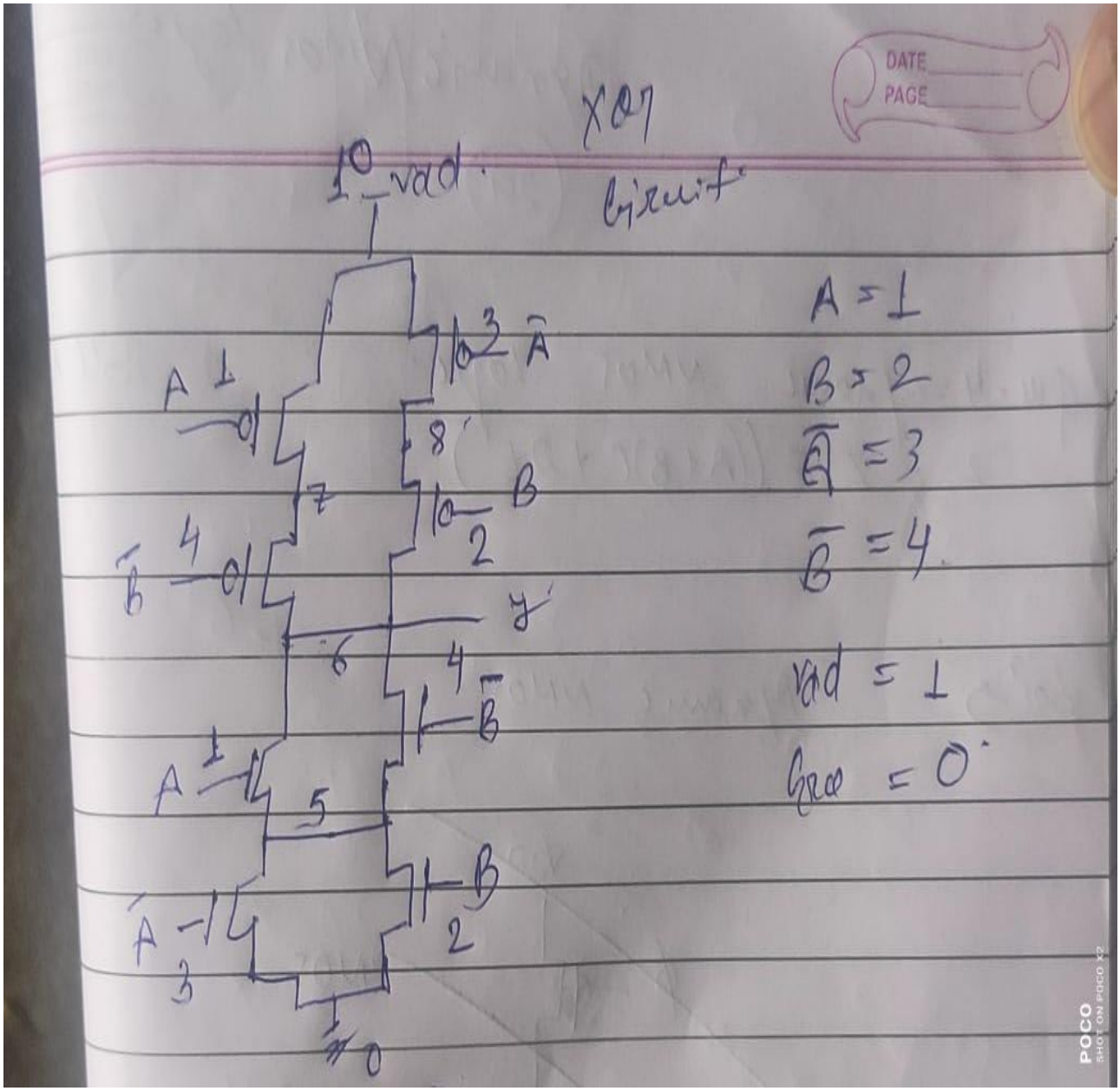


Code for nand gate

```
.subckt nand 2 3 4 1
mn1 5 3 0 0 mod1 l=1u w =100u
mn2 4 2 5 5 mod1 l=1u w =100u
mp1 4 2 1 1 mod2 l=1u w=100u
mp2 4 3 1 1 mod2 l=1u w=100u

.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends
```

2 =first input which we want
3 =second input which we want
4 =output where we want
1 =location of ground

3.XOR



CODE FOR XOR

```
.subckt xor 1 2 3 4 6 10
mn1 5 3 0 0 mod1 l=1u w =100u
mn2 5 2 0 0 mod1 l=1u w =100u
mn3 6 1 5 5 mod1 l=1u w =100u
mn4 6 4 5 5 mod1 l=1u w =100u
mp1 7 1 10 10 mod2 l=1u w =100u
mp2 6 4 7 7 mod2 l=1u w =100u
mp3 6 2 8 8 mod2 l=1u w =100u
mp4 8 3 10 10 mod2 l=1u w =100u

.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends
```

1=INPUT A
2=INPUT B
3 =INPUT NOT A
4=INPUT NOT B

6=FOR OUTPUT
10 =LOCATION OF GROUND

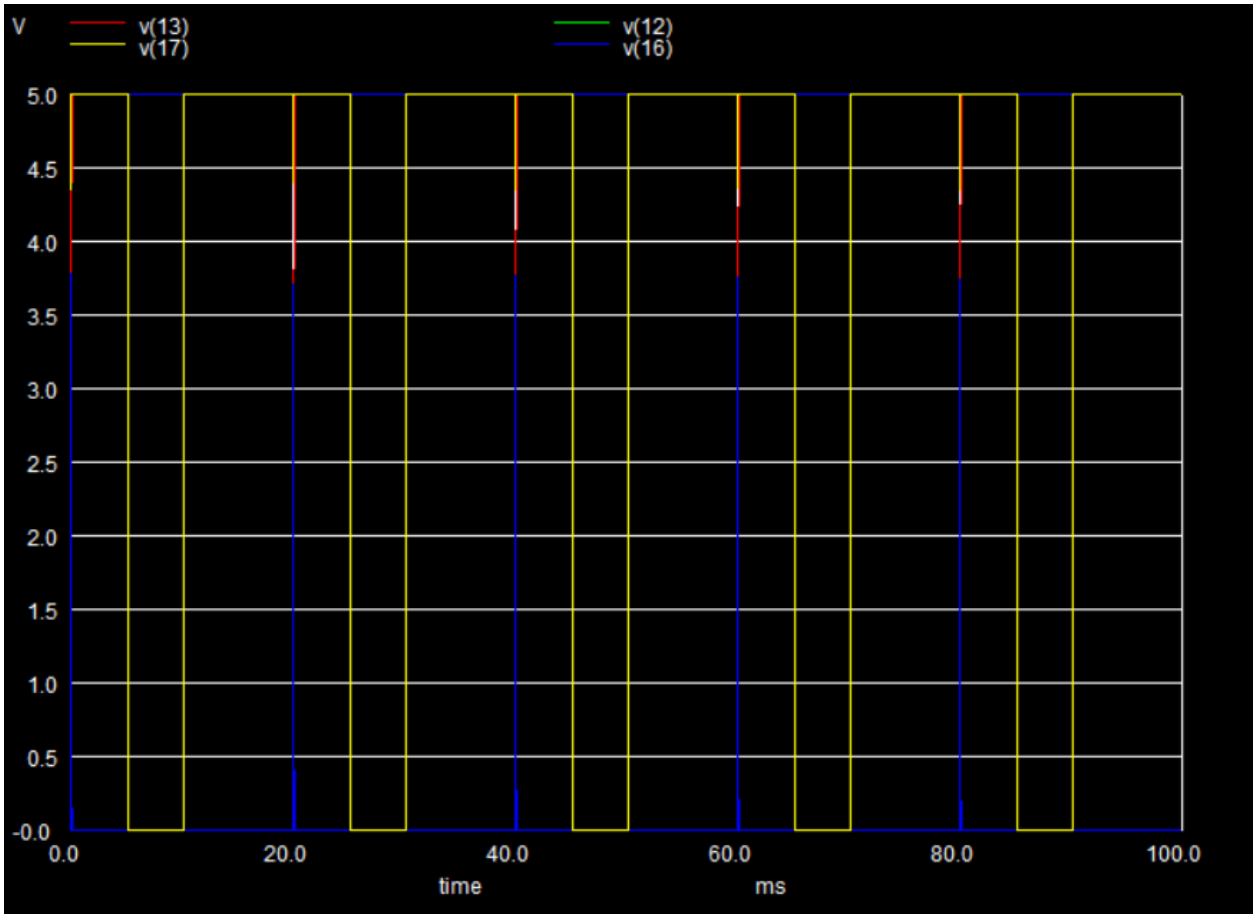
FINAL CODE

Project HALF SUBTRACTOR

```
.subckt inverter 2 3 1
mn 3 2 0 0 mod1 l=1u w=100u
mp 3 2 1 1 mod2 l=1u w=100u
.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends


.subckt nand 2 3 4 1
mn1 5 3 0 0 mod1 l=1u w =100u
mn2 4 2 5 5 mod1 l=1u w =100u
mp1 4 2 1 1 mod2 l=1u w=100u
mp2 4 3 1 1 mod2 l=1u w=100u

.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends

.subckt xor 1 2 3 4 6 10
mn1 5 3 0 0 mod1 l=1u w =100u
mn2 5 2 0 0 mod1 l=1u w =100u
mn3 6 1 5 5 mod1 l=1u w =100u
mn4 6 4 5 5 mod1 l=1u w =100u
```

```
mp1 7 1 10 10 mod2 l=1u w =100u
mp2 6 4 7 7 mod2 l=1u w =100u
mp3 6 2 8 8 mod2 l=1u w =100u
mp4 8 3 10 10 mod2 l=1u w =100u

.model mod1 nmos level=54 version=4.7
.model mod2 pmos level=54 version=4.7
.ends

Vdd 10 0 dc 5v

va 12 0 pulse(0 5 0 0 0 10ms 20ms) $ input a
vb 13 0 pulse(0 5 0 0 0 5ms 20ms)  $ input b
xinv_a 12 14 10 inverter          $ input not a
xinv_b 13 15 10 inverter          $ input not b

xxor_output_difference 12 13 14 15 16 10 xor
xborrow 12 15 17 10 nand


.tran 0.1ms 100ms
.control
run
plot    v(12) V(13) V(16) V(17)
.endc
.end
```

OUTPUT:


V
v(13)
v(17)
v(12)
v(16)
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
-0.0
0.0
20.0
40.0
60.0
80.0
100.0
time
ms