Name: Caden Roberts **Last 4 ID:** 2015 **CSE 122/222A**

Due: January 26 (midnight) **HW 2**

Instructions: Complete the problems below **in your own handwriting**. Show any work. Circle your answers where necessary. If the work or answer is not readable, you will not receive credit.

Also, submit the following required file as described in the problems to your github repository:

- hw2/README.md

You should go through the tutorials at

- https://vlsida.github.io/chip-tutorials/git.html
- https://vlsida.github.io/chip-tutorials/klayout.html
- https://vlsida.github.io/chip-tutorials/spice.html

While we will not check that you did this, you need to learn the tools well enough to do this assignment. For now, you can skip the DRC, LVS, and Edit sub-pages of KLayout until HW3. You will need to use gitlab to do the last problem and to submit your future homeworks. Several of these problems use SPICE netlists and that tutorial will help you understand the format.

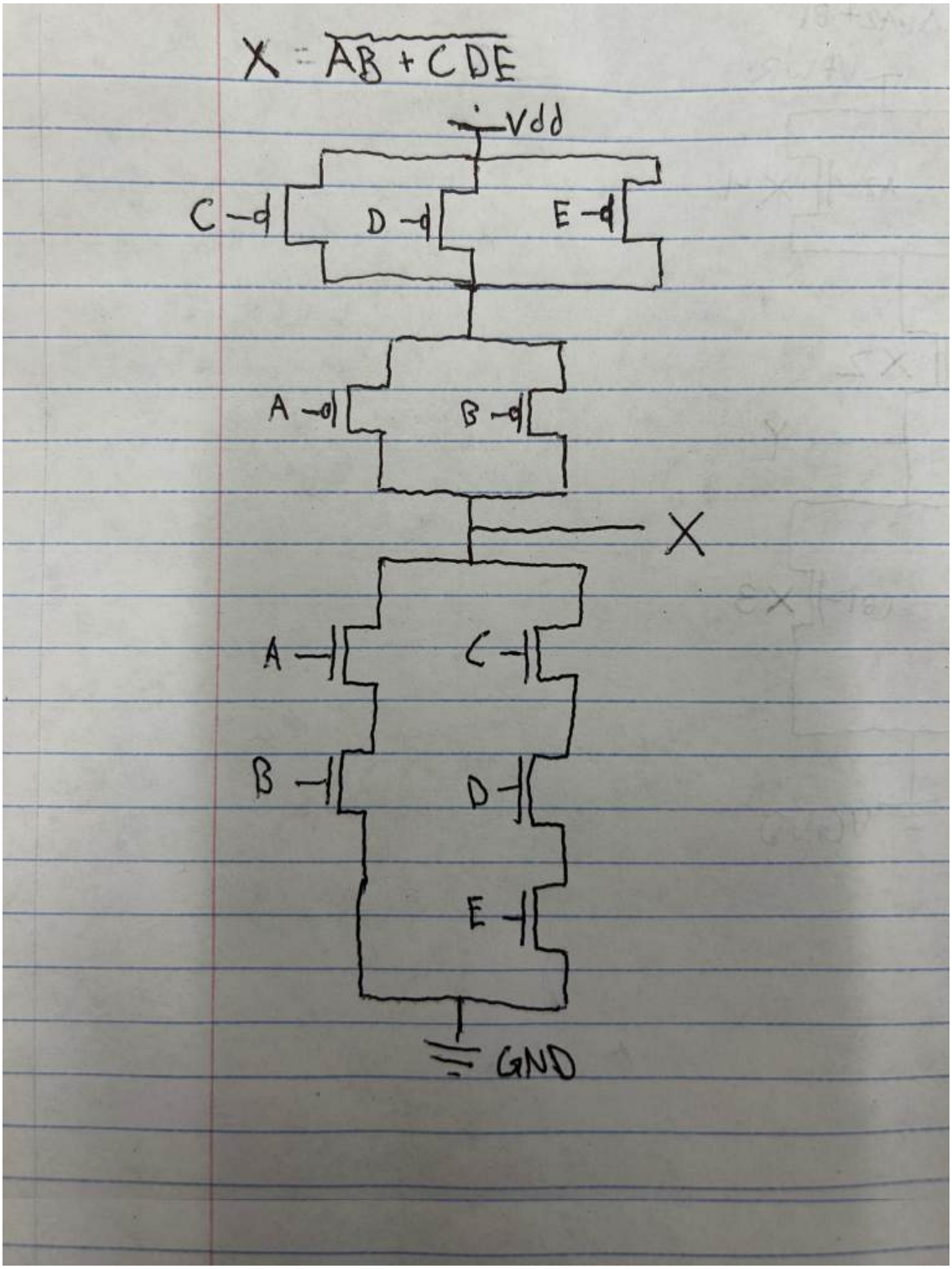
The PDK contains the spice files for the gates you will need to analyze. You can find the spice library in your home directory at:

```
~/.volare/sky130A/libs.ref/sky130_fd_sc_hd/spice/sky130_fd_sc_hd.spice
```

This contains all the library cells (i.e., subcircuits for each gate) in the Sky130 HD (high-density) library. If you do not have this, make sure to install the PDK as described in the SPICE tutorial.

1 CMOS Logic Gates [15 points]

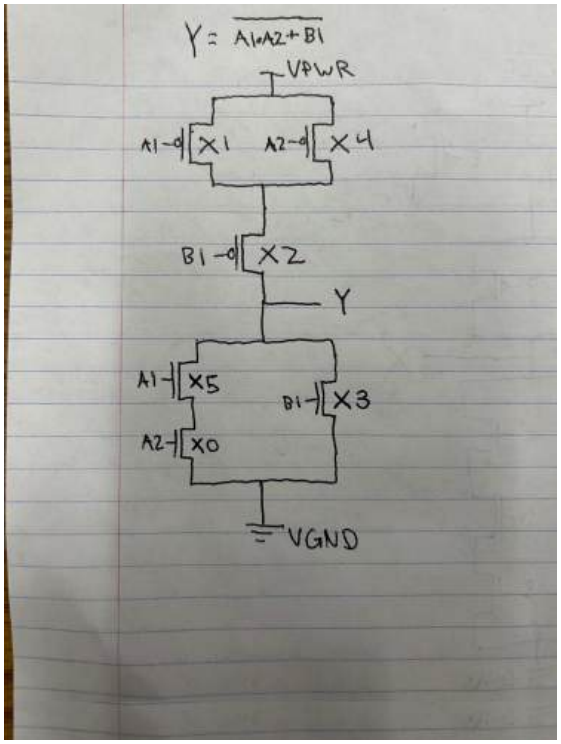
Draw a single CMOS logic gate that implements the function: $X = \overline{AB + CDE}$



2 Spice Netlists [20 points]

Draw the transistor-level schematic of the following Sky130 library gate: **sky130_fd_sc_hd__a21oi_1**. Label the inputs and output. What is the logic function?

Transistor Schematic:




```
.subckt sky130_fd_sc_hd__a21oi_1 A1 A2 B1 VGND VNB VPB VPWR Y
X0 a_199_47# A2 VGND VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X1 a_113_297# A1 VPWR VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X2 Y B1 a_113_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X3 VGND B1 Y VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X4 VPWR A2 a_113_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X5 Y A1 a_199_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
.ends
```

Equation: **Y = !((A1*A2) + B1)**

3 Transistor Sizes [15 points]

What are the instance name, width, length (in nm) of the PMOS and NMOS transistors connected to input A1 in the **sky130_fd_sc_hd__a221o_1** library gate? (Note this is a different gate than the previous problem.)

PMOS:

- Instance: **X0**
- W = **1000** nm
- L = **150** nm

NMOS:

- Instance: **X5**
- W = **650** nm
- L = **150** nm

```
.subckt sky130_fd_sc_hd__a221o_1 A1 A2 B1 B2 C1 VGND VNB VPB VPWR X
X0 VPWR A1 a_193_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X1 a_193_297# A2 VPWR VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X2 VGND a_27_47# X VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X3 a_205_47# B1 a_27_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X4 a_465_47# A2 VGND VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X5 a_27_47# A1 a_465_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X6 a_193_297# B1 a_109_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X7 a_27_47# C1 VGND VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X8 a_27_47# C1 a_109_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X9 a_109_297# B2 a_193_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X10 VGND B2 a_205_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X11 VPWR a_27_47# X VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
.ends
```

4 Transistor Operation [15 points]

For each transistor, indicate whether it is "ON" (conducting) or "OFF" (non-conducting) in the gate **sky130_fd_sc_hd__a22oi_1** with input values A1=0V, A2=1.8V, B1=1.8V, B2=1.8V, VPWR=1.8V, VGND=0.

- X0: ON
- X1: OFF
- X2: ON
- X3: ON
- X4: ON
- X5: OFF
- X6: OFF
- X7: OFF

```
.subckt sky130_fd_sc_hd__a22oi_1 A1 A2 B1 B2 VGND VNB VPB VPWR Y
X0 a_109_47# B1 Y VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X1 a_109_297# A2 VPWR VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X2 VPWR A1 a_109_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X3 a_381_47# A2 VGND VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X4 VGND B2 a_109_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X5 Y B2 a_109_297# VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
X6 Y A1 a_381_47# VNB sky130_fd_pr__nfet_01v8 w=650000u l=150000u
X7 a_109_297# B1 Y VPB sky130_fd_pr__pfet_01v8_hvt w=1e+06u l=150000u
.ends
```

Y = !(A1*A2+B1*B2) = !(LOW*HIGH+HIGH*HIGH) = !(LOW+HIGH) = !(HIGH) = 0V
a_109_47# = 0V, NFET X0 is ON because B1-Y=1.8V-0=1.8V so it pulls a_109_47# down to Y at 0V
a_109_297# = 1.8V, PFET X2 is ON because a_109_297# - A1 = a_109_297# - 0V, so when the voltage at the node starts rising X2 will be ON and pull a_109_297# to VPWR at 1.8V
a_381_47# = 0V, NFET X3 is ON because A2-VGND = 1.8V-0 = 1.8V, so X3 will pull a_381_47# to VGND at 0V

NFET X0 = B1 - Y = 1.8V - 0V = 1.8V ON
PFET X1 = VPWR - A2 = 1.8V - 1.8V = 0V OFF
PFET X2 = a109_297# - A1 = 1.8V - 0V = 1.8V ON
NFET X3 = A2 - VGND = 1.8V - 0V = 1.8V ON
NFET X4 = B2 - a_109_47# = 1.8V - 0V = 1.8V ON
PFET X5 = a_109_297# - B2 = 1.8V - 1.8V = 0V OFF
NFET X6 = A1 - a_381_47# = 0V - 0V = 0V OFF
PFET X7 = Y - B1 = 0V - 1.8V = -1.8V OFF

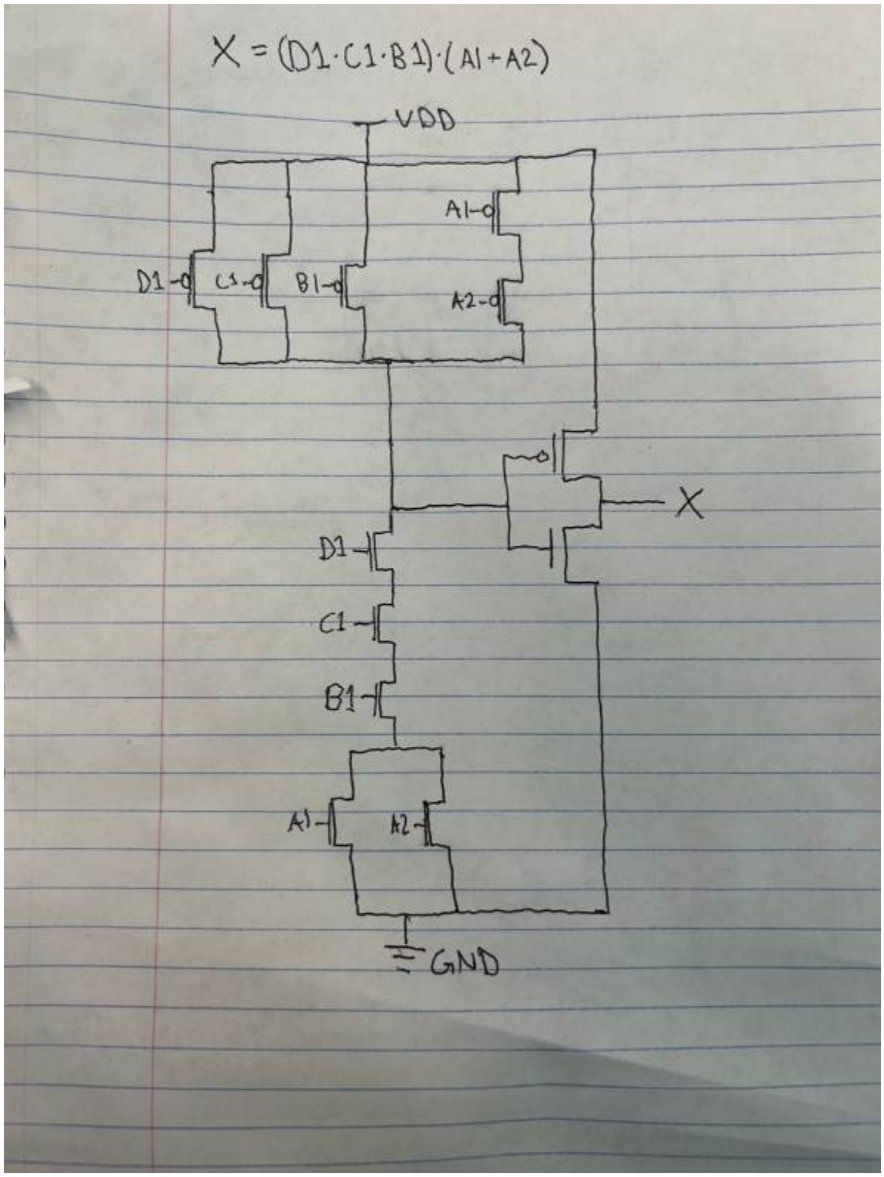
5 Layout to SPICE [30 points]

Depending on the first letter of your last name, select one of the following cells (available in Canvas):

- A-G: cell1.gds
- H-M: cell2.gds
- N-S: cell3.gds
- T-Z: cell4.gds

Open the GDS file in KLayout and draw the transistor-level schematic. Label the inputs and output according to the labels in the layout. What is the logic function?

Transistor Schematic:



Equation: X=(D1*C1*B1)*(A1+A2)

6 Gitlab [5 points]

Login to the UCSC gitlab server (git.ucsc.edu). In your CSE 122 project directory, add a file in a subdirectory called `hw2/README.md` with the following contents:

```
cruzid
```

where `cruzid` is your UCSC cruzid. I recommend that you do this via the command-line interface rather than the web interface to familiarize yourself with Git for future submissions.

Done and pushed through the command line.