EECS 281: Homework #1 Problem 1 (1a SPICE)

Convert the following schematic to (a) SPICE, (b) truth table, (c) logic gates (d) logic expression.

```plaintext
.SUBCKT GATE1X 1 2 3 4 5 6
*M .SUBCKT GATE1X A1=1 B1=2 C1=3 D1=4 Q1=5 VDD=6 *
* Mname DRAIN GATE SOURCE SUBSTRATE MODEL WIDTH LENGTH *
* NODE NODE NODE NODE NAME MICRONS MICRONS *
------- ----- ----- ------ --------- ----  ------- -----
MP0   10  3  6  6    PCH W=20U L=2U
MP2   7  1 10  6    PCH W=20U L=2U
MN0   7  2  8  0    NCH W=5U L=2U
MN1   8  1  0  0    NCH W=5U L=2U
MP1   10  4  6  6    PCH W=20U L=2U
MP3   7  2 10  6    PCH W=20U L=2U
MN2   7  3  9  0    NCH W=5U L=2U
MN3   9  4  0  0    NCH W=5U L=2U
MP4   5  7  6  6    PCH W=20U L=2U
MN4   5  7  0  0    NCH W=6U L=2U

.ENDS GATE1X

*MODEL NAME TYPE; 2 Micron process technology
*------- ----- -----
.model NCH nmos LEVEL=1 KP=48E-6 LAMBDA=0.032 VTO=0.88 GAMMA=0.66 PHI=0.7
.model PCH pmos LEVEL=1 KP=16E-6 LAMBDA=0.044 VTO=-0.85 GAMMA=0.69 PHI=0.7

Compare this with with the CMOS AND-OR-INVERT gate Fig. 3-20 in Wakerly's book.
```
Another way to view CMOS logic is the box flow model.
1b: flow model

Convert the following schematic to (a) SPICE, (b) truth table, (c) logic gates (e) logic expression.

Rewrite the circuit as a flow model

Notice these connections
**1b: flow model**

Another way to view this (where x=1 or 0=Don't Care):

- **W₁** becomes 1 when C=0 & A=0 otherwise Z
  
  \[ \Rightarrow AxCx \Rightarrow 0x0x \Rightarrow 0000, 0001, 0100, 0100 \]

  Also, **W₁** becomes 1 when D=0 & A=0 & otherwise Z
  
  \[ \Rightarrow AxD \Rightarrow 0xx0 \Rightarrow 0000, 0010, 0100, 0110 \]

- **W₂** becomes 1 when (D=0 & B=0) OR (C=0 & B=0) otherwise Z
  
  \[ \Rightarrow xBxD \Rightarrow x0x0 \Rightarrow 0000, 0010, 1000, 1010 \]

  \[ \Rightarrow xBCx \Rightarrow x00x \Rightarrow 0000, 0001, 1000, 1001 \]

OVERLAP: notice **W₃** and **W₄** overlaps when ABCD is equal to 1111 that is ok because the result **W₄** = 1 **W₄** = 1

it's when the differ, it's bad!

- **W₄** becomes 0 only when D=1 & C=1
  
  \[ \Rightarrow xxCD \Rightarrow xx11 \Rightarrow 0011, 0111, 1011, 1011 \]

- **W₃** becomes 0 only when A=1 & B=1
  
  \[ \Rightarrow ABxx \Rightarrow 11xx \Rightarrow 1100, 1101, 1110, 1111 \]
1b: Truth Table (behavior)

**W₁** is Z otherwise is 1 when

=> Ax*Cx => 0x0x => 0000, 0001, 0100, 0101

=> AxxD => 0xx0 => 0000, 0010, 0100, 0110

**W₂** is Z otherwise is 1 when

=> xBxD => x0x0 => 0000, 0010, 1000, 1010

=> xBCx => x00x => 0000, 0001, 1000, 1001

**W₃** becomes 0 when A=1 & B=1 otherwise Z

=> ABxx => 11xx => 1100, 1101, 1110, 1111

**W₄** becomes 0 when C=1 & D=1 otherwise Z

=> xxCD => xx11 => 0011, 0111, 1011, 1111

W = W₁ OR W₂ OR W₃ OR W₄

Q = NOT(W) = NOT(W₁ OR W₂ OR W₃ OR W₄)

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>W₁</th>
<th>W₂</th>
<th>W₃</th>
<th>W₄</th>
<th>W</th>
<th>Q</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Z</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Z</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>Z</td>
<td>Z</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Cases of ABCD

- **W₁,W₁,W₂,W₂**: 0x0x, 0xx0, x0x0, x00x
- **W₁,W₂**: 0x0x, x00x
- **W₁,W₂**: 0xx0, x0x0
- **W₄**: xx11
- **W₁,W₁**: 0x0x, 0xx0
- **W₂**: 0x0x
- **W₁**: 0xx0
- **W₄**: xx11
- **W₂**: x0x0, x00x
- **W₂**: x00x
- **W₂**: x0x0
- **W₄**: xx11
- **W₃**: 11xx
- **W₃**: 11xx
- **W₃**: 11xx
- **W₃,W₄**: xx11, 11xx
### 1c: Logic Gate Level

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>Q</th>
<th>SOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>A B C D</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>A B C D</td>
</tr>
</tbody>
</table>

![Logic Gate Diagram](image-url)
1d: Logic Expression

\[
Q = \left( \sim A \& \sim B \& C \& D \right) \\
| \left( \sim A \& B \& C \& D \right) \\
| \left( A \& \sim B \& C \& D \right) \\
| \left( A \& B \& \sim C \& \sim D \right) \\
| \left( A \& B \& \sim C \& D \right) \\
| \left( A \& B \& C \& \sim D \right) \\
| \left( A \& B \& C \& D \right);
\]
2a: Logic Expression

Re-write the following schematic as two logic expressions, sum=? And cout=? (b) as VHDL (c) and convert schematic using only NORs.

\[ \text{sum} = \text{cin} \land (a \land b) = \text{cin} \lor (a \lor b); \]

\[ \text{VHDL: } \text{cout} = \left( ( (a \land b) \lor (a \land \text{cin}) ) \lor (b \land \text{cin}) \right); \]

\[ \text{C/C++: } \text{cout} = \left( ( (a \land b) \lor (a \land \text{cin}) ) \lor (b \land \text{cin}) \right); \]
2b: VHDL

Re-write the following schematic as two logic expressions, sum=? And cout=? (b) as VHDL (c) and convert schematic using only NORs.

ENTITY P2 IS
    PORT ( a, b, cin: IN std_logic;
           Sum, Cout: OUT std_logic
    );
END P2;

ARCHITECTURE P2A OF P2 IS
BEGIN
    sum <= cin XOR (a XOR b);
    cout <= ((a AND b) OR (a AND cin))
        OR (b AND cin);
END P2A;
2c: AND-to-NOR, NOT-to-NOR, OR-to-NOR NOR Transforms

Re-write the following schematic as two logic expressions, sum=? And cout=? (b) as VHDL (c) and convert schematic using only NORs.

DeMorgan's Law:

\[ X = \text{NOT} (\text{NOT} x) \]

\[ \text{NOT}(X) = \text{NOT}(X \text{ OR } X) = X \text{ NOR } X; \]
2c: XOR Transform

Truth Table: behavior

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>f_{xor}</th>
<th>XOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>~x &amp; y</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>x &amp; ~y</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

XOR = (~x & y) | (x & ~y)

Gate Level

DeMorgan's Law

SOP

XOR = (~x & y) | (x & ~y)
2c: XOR Transform

Replace every XOR with the following structural circuit: