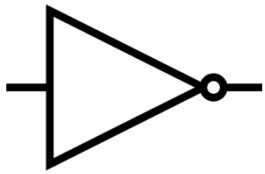
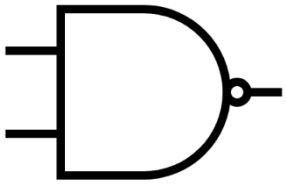
Primitive Gates

For each primitive logic gate shown, draw the transistor-level schematic.

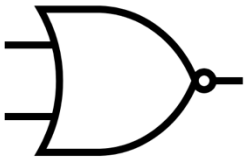
a. NOT



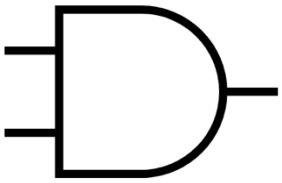
b. NAND2



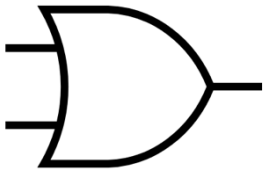
c. NOR2



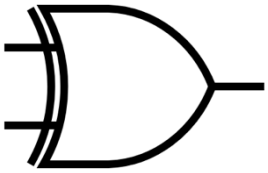
d. AND2



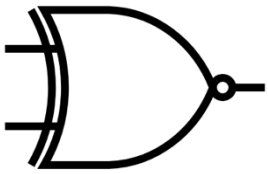
e. OR2



f. XOR2



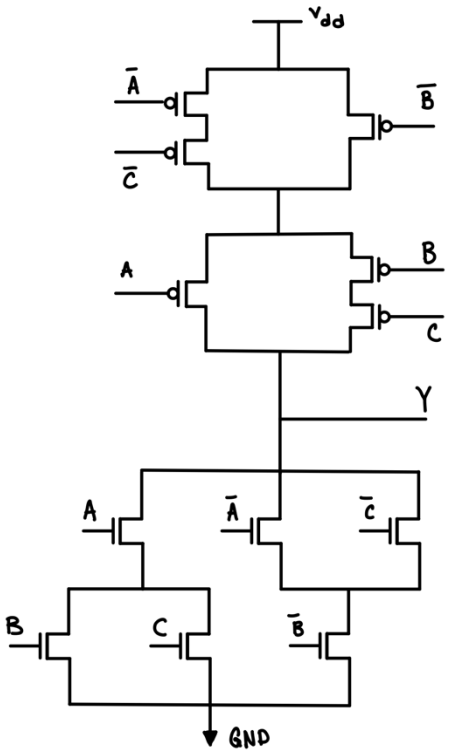
g. XNOR2



From Digital Circuits to Truth Tables

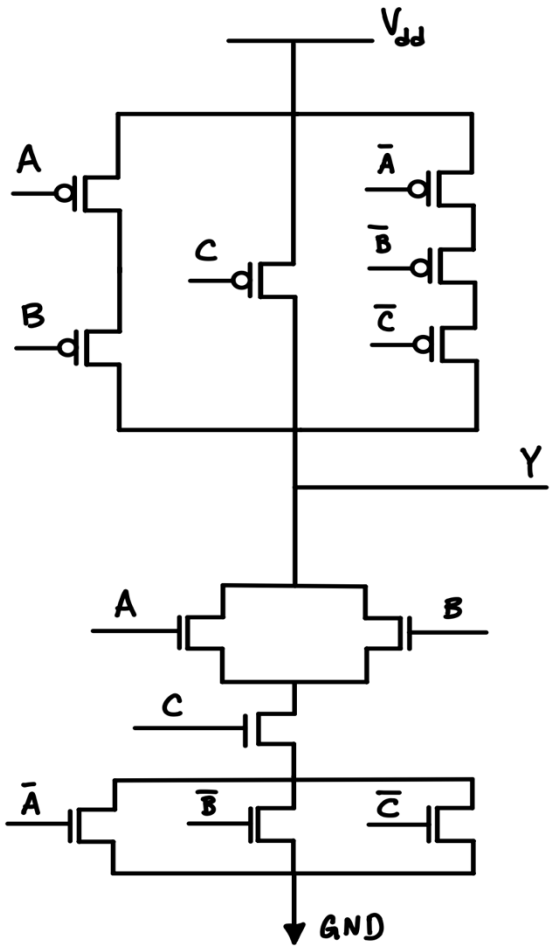
For the following complementary digital circuits fill out their truth table.

a.



a	b	c	y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

b.



a	b	c	y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

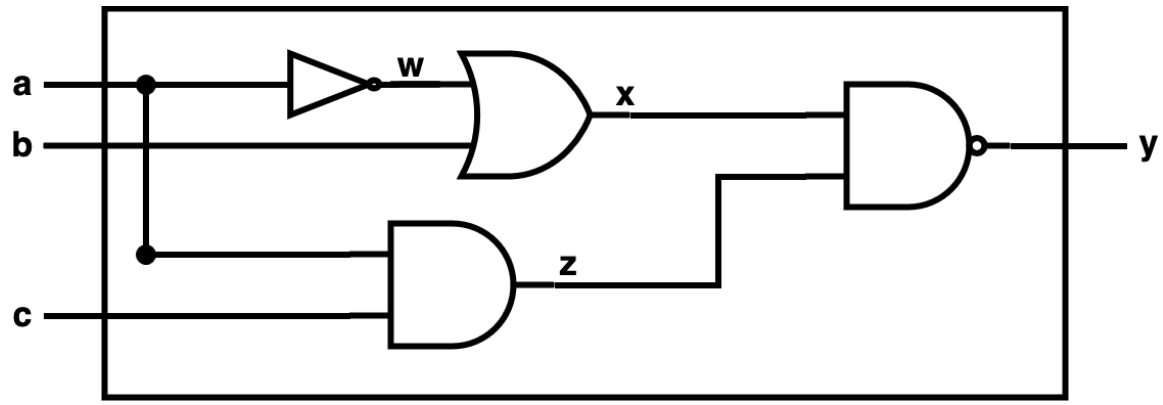
Gate-Level Networks, Simulation Tables, and Verilog

For each of the following problems you will be asked to fill out the truth table for a gate-level netlist. After, fill out the provided simulation table. Finally implement the gate-level network in Verilog. A module definition will be provided. Write Verilog coded in the provided space.

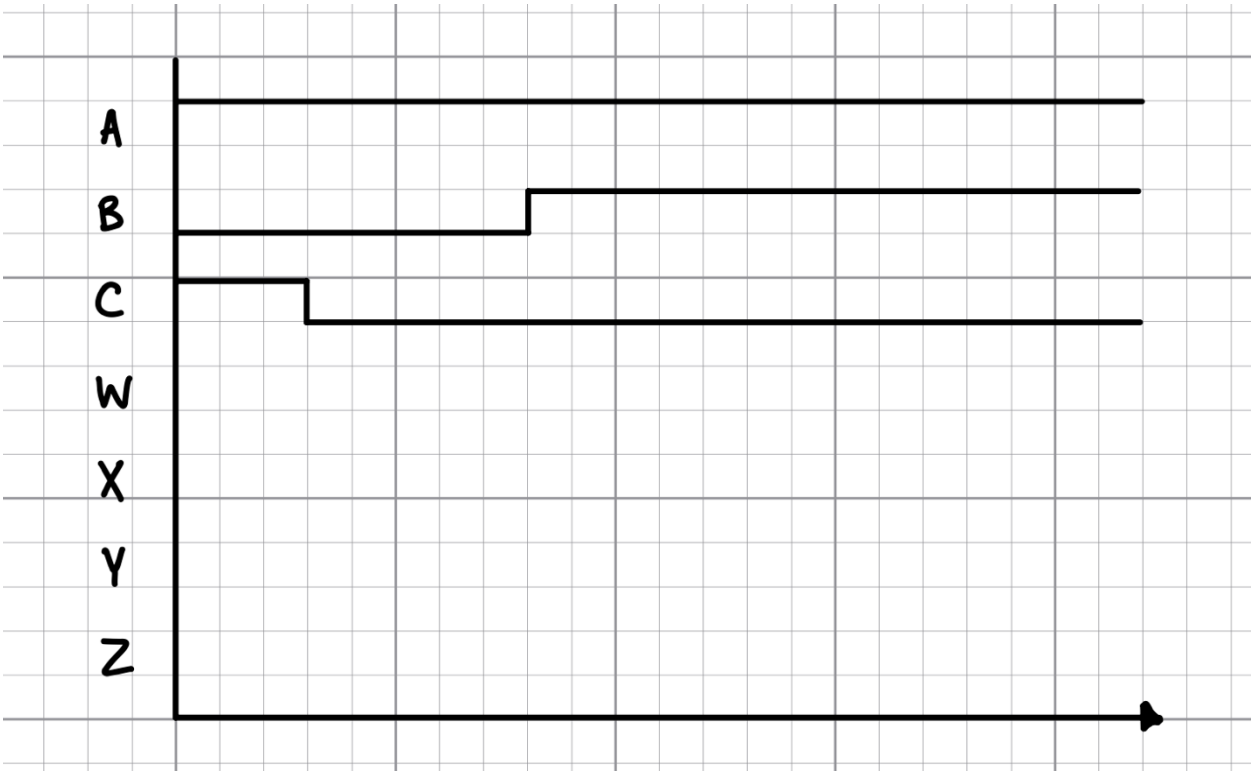
Assume we have the following gate level propagation delay and contamination delays:

Gate	t_{pd}	t_{cd}
NOT	1τ	1τ
AND2	3τ	1τ
OR2	4τ	2τ
NAND2	4τ	2τ
NOR2	5τ	3τ
XOR2	7τ	4τ

a.

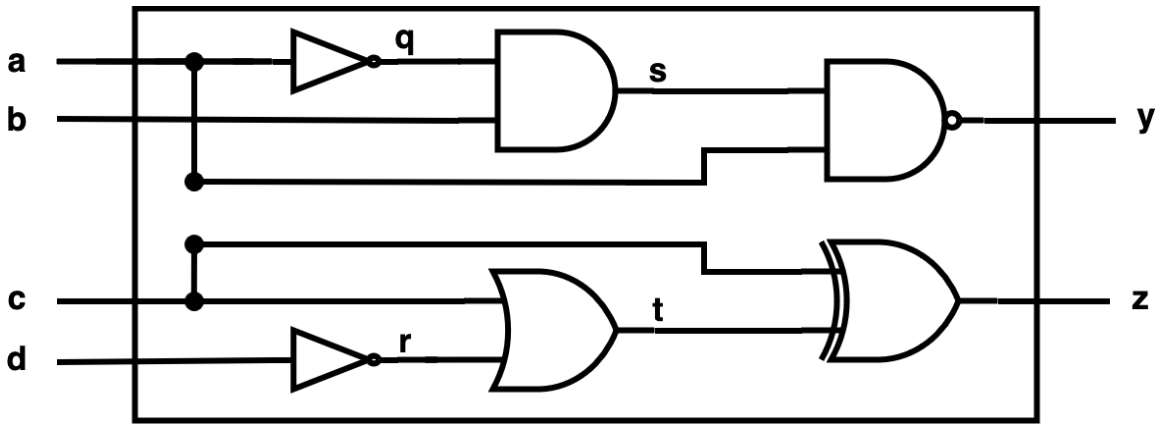


a	b	c	w	x	z	y
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

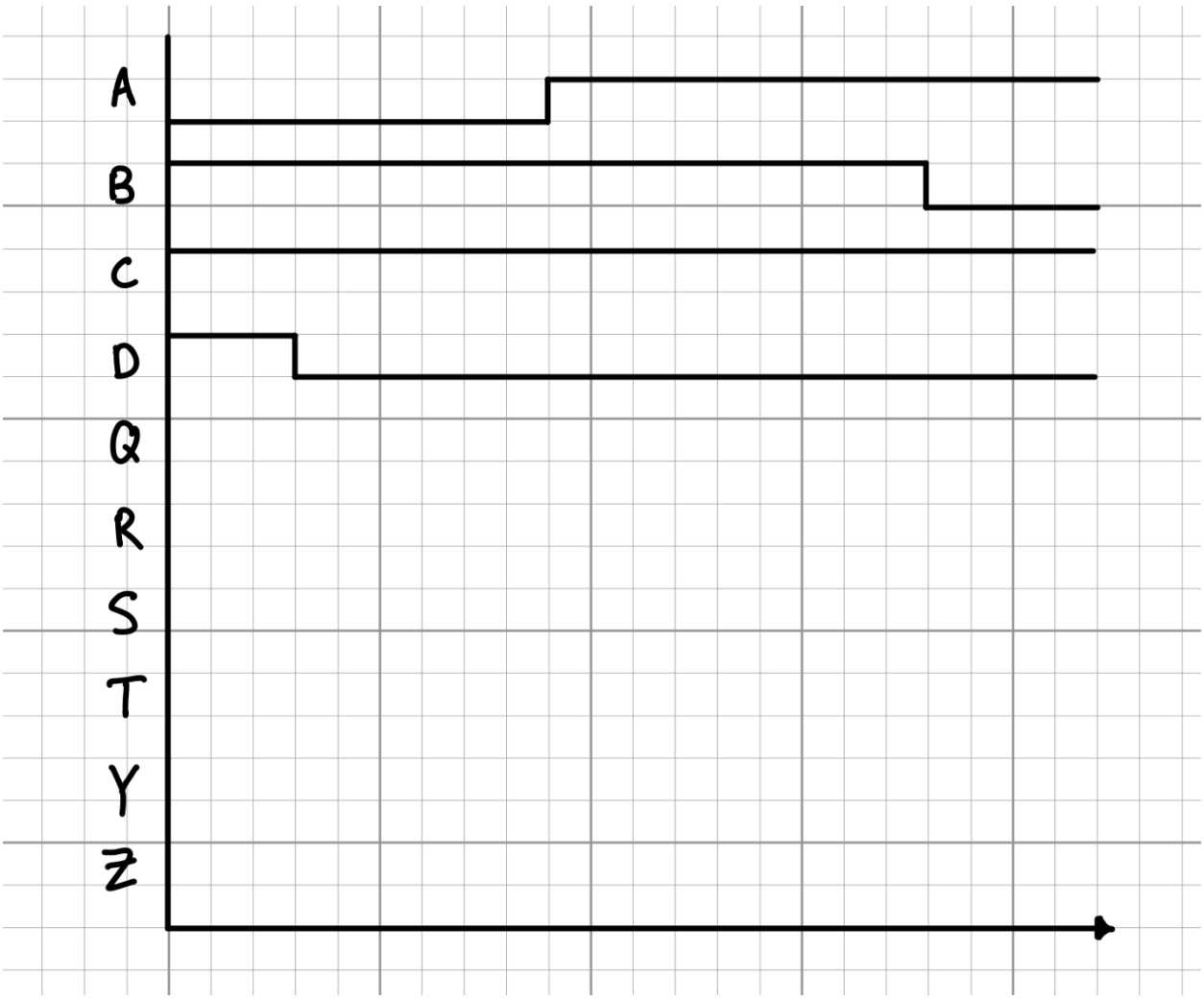

A
B
C
W
X
Y
Z

```
module LogicFun1
(
  input  wire a,
  input  wire b,
  input  wire c,
  output wire y
);
// Implement the Gate Level Netlist showed in the diagram.









endmodule;
```

b.



a	b	c	d	q	r	s	t	y	z
0	0	0	0						
0	0	0	1						
0	0	1	0						
0	0	1	1						
0	1	0	0						
0	1	0	1						
0	1	1	0						
0	1	1	1						
1	0	0	0						
1	0	0	1						
1	0	1	0						
1	0	1	1						
1	1	0	0						
1	1	0	1						
1	1	1	0						
1	1	1	1						


A
B
C
D
Q
R
S
T
Y
Z

```
module LogicFun2
(
  input  wire a,
  input  wire b,
  input  wire c,
  input  wire d,
  output wire y,
  output wire z
);
// Implement the Gate Level Netlist showed in the diagram.









endmodule;
```