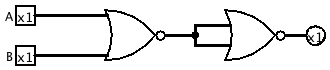
CPSC 121 Assignment #1

1. a) The following circuit simulates ~ using NOR gates.  
NOT with NOR.png  
Here is the truth table for this circuit.

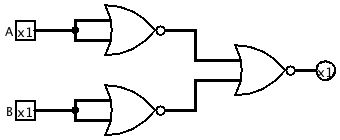
|  |  |  |
| --- | --- | --- |
| A | A | A NOR A |
| F | F | T |
| F | T | not possible |
| T | F | not possible |
| T | T | F |

This is identical to the truth table for ~, so this circuit is logically equivalent to ~.

b) The following circuit simulates ∨ using NOR gates.  
  
Here is the truth table for this circuit.

|  |  |  |
| --- | --- | --- |
| A | B | ~(A NOR B) |
| F | F | F |
| F | T | T |
| T | F | T |
| T | T | T |

This is identical to the truth table for ∨, so this circuit is logically equivalent to ∨.

c) The following circuit simulates ∧ using NOR gates.  
  
Here is the truth table for this circuit.

|  |  |  |
| --- | --- | --- |
| A | B | ~A NOR ~B |
| F | F | F |
| F | T | F |
| T | F | F |
| T | T | T |

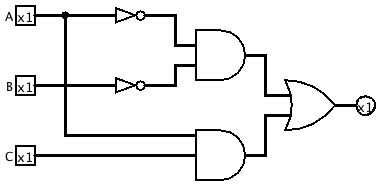
This is identical to the truth table for ∧, so this circuit is logically equivalent to ∧.

d) Any truth table can be translated into a statement using ∨ to connect rows that result in a TRUE. Within each row the inputs are connected with ∧, and where necessary preceded with ~.   
  
For example a truth table with five inputs (A, B, C, D, and E) that contains a row that outputs as TRUE such as T, T, F, T, F would be translated as (A ∧ B ∧ ~C ∧ D ∧ ~E).

Thus, using only ~, ∨ and ∧ a logic function can be created for a truth table with an arbitrary number of inputs, and this can be implemented with a circuit using only NOR gates, using the above circuits for ~, ∨ and ∧.

2. a) The statement (~A ∧ ~B) ∨ (A ∧ C) corresponds to the truth table given. Here is the full truth table for this statement.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| A | B | C | (~ A | ∧ | ~B) | ∨ | (A ∧ C) |
| F | F | F | T | T | T | T | F |
| F | F | T | T | T | T | T | F |
| F | T | F | T | F | F | F | F |
| F | T | T | T | F | F | F | F |
| T | F | F | F | F | T | F | F |
| T | F | T | F | F | T | T | T |
| T | T | F | F | F | F | F | F |
| T | T | T | F | F | F | T | T |

Here is the circuit corresponding to the logical statement (~A ∧ ~B) ∨ (A ∧ C).  


3. a) and c) are equivalent as shown below.  
 (c ∧ a) → (c ∧ b) ~a ∨ (c → b)  
 ~(c ∧ a) ∨ (c ∧ b) IMP ~a ∨ ~c ∨ b IMP  
 ~c ∨ ~a ∨ (c ∧ b) DM  
 ~a ∨ ~c ∨ (c ∧ b) COM  
 ~a ∨ [(~c ∨ c) ∧ (~c ∨ b)] DIST  
 ~a ∨ [T ∧ (~c ∨ b)] NEG  
 ~a ∨ ~c ∨ b I

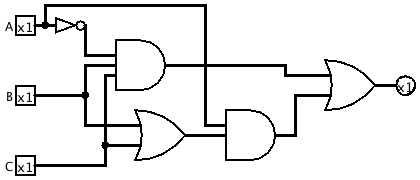
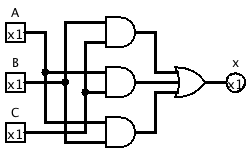
b) and f) are equivalent as shown below.  
 (a ∧ (b ∨ c)) → (a ∧ c) ~(a ∧ ~c) ∨ ~(b ∨ c)  
 ~(a ∧ (b ∨ c)) ∨ (a ∧ c) IMP ~a ∨ c ∨ (~b ∧ ~c) DM *x* 2  
 ~a ∨ ~(b ∨ c) ∨ (a ∧ c) DM ~a ∨ [(c ∨ ~b) ∧ (c ∨ ~c)] DIST  
 ~a ∨ (~b ∧ ~c) ∨ (a ∧ c) DM ~a ∨ [(c ∨ ~b) ∧ T] NEG  
 ~a ∨ (a ∧ c) ∨ (~b ∧ ~c) COM ~a ∨ c ∨ ~b I  
 [(~a ∨ a) ∧ (~a ∨ c)] ∨ (~b ∧ ~c) DIST   
 [T ∧ (~a ∨ c)] ∨ (~b ∧ ~c) NEG   
 ~a ∨ c ∨ (~b ∧ ~c) I   
 ~a ∨ [(c ∨ ~b) ∧ (c ∨ ~c)] DIST   
 ~a ∨ [(c ∨ ~b) ∧ T] NEG  
 ~a ∨ c ∨ ~b I

d) and e) are equivalent as shown below – only e) is simplified since d) is already simplified.  
 ~{[(a ∨ b) ∧ ~(a ∧ b)] ∧ [~a ∧ b]}  
 ~[(a ∨ b) ∧ ~(a ∧ b)] ∨ ~[~a ∧ b] DM  
 ~(a ∨ b) ∨ (a ∧ b) ∨ a ∨ ~b DM *x* 2  
 (~a ∧ ~b) ∨ (a ∧ b) ∨ a ∨ ~b DM  
 (~a ∧ ~b) ∨ (a ∧ b) ∨ ~b ∨ a COM  
 (~a ∧ ~b) ∨ [(a ∨ ~b) ∧ (b ∨ ~b)] ∨ a DIST  
 (~a ∧ ~b) ∨ [(a ∨ ~b) ∧ T] ∨ a NEG  
 (~a ∧ ~b) ∨ (a ∨ ~b) ∨ a NEG  
 (~a ∧ ~b) ∨ a ∨ (a ∨ ~b) COM  
 [(~a ∨ a) ∧ (~b ∨ a)] ∨ (a ∨ ~b) DIST  
 [T ∧ (~b ∨ a)] ∨ (a ∨ ~b) NEG  
 (~b ∨ a) ∨ (a ∨ ~b) I  
 (a ∨ ~b) ∨ (a ∨ ~b) COM  
 a ∨ ~b ID

4. a) From the definition we created the following truth table.

|  |  |  |  |
| --- | --- | --- | --- |
| a | b | c | “majority” |
| F | F | F | F |
| F | F | T | F |
| F | T | F | F |
| F | T | T | T |
| T | F | F | F |
| T | F | T | T |
| T | T | F | T |
| T | T | T | T |

Then, breaking the table into the top half we derived the statement ~a ∧ b ∧ c. Looking at the bottom half we derived the statement a ∧ (b ∨ c). We then combine the two statements as follows (~a ∧ b ∧ c) ∨ (a ∧ (b ∨ c)).

b) Here is the circuit corresponding to this logical statement.  
  
  
Using Logisim we were able to generate the somewhat simpler circuit following.  
  
This corresponds to the logical statement (a ∧ b) ∨ (a ∧ c) ∨ (b ∧ c).

5. A says “I don’t know” so that means that A sees either two black hats or one hat of each colour (if A had seen two white hats he would have known his hat was not white since there cannot be three white hats).  
  
B hears A, and draws the same conclusions, and then says “I don’t know” so B does not see a white hat since then he would know his hat was black. So, B must see a black hat.  
  
Thus C knows her hat must be black.  
  
We can explore all possible arrangements of the hats by looking at the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C |  |
| white | white | white | excluded by definition |
| white | white | black | possible |
| white | black | white | B would know his hat was black |
| white | black | black | possible |
| black | white | white | A would know his hat was black |
| black | white | black | possible |
| black | black | white | B would know his hat was black |
| black | black | black | possible |