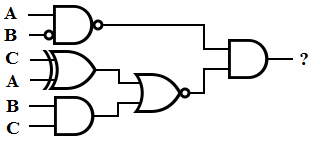
**CSC 1500 – Exam 1**

(1) For the following logical circuit, perform two actions. FIRST, convert the circuit into a logical statement. SECOND, create a truth table based on the circuit/statement. (*10 pts. each for statement and truth table.*)

A picture containing font, text, diagram, line

Description automatically generated

(A NAND !B) AND ( (A XOR C) NOR (B AND C))

A NAND !B is just A->B. Eh, too complicated to write down.

0s and 1s get confusing, true and false are friendlier.

A picture containing text, screenshot, number, line

Description automatically generated

Obviously this is much simpler. Light grey means a logic gate, darker grey means two logic gates, darkest grey means 3 logic gates.

(2) Use logical equivalences to perform a direct proof that the two following logical statements are equivalent. (*20 pts.*)

**Statement 1: !(P OR (!P AND Q))**

**De Morgan’s Law: !P AND !(P AND Q)**

**De Morgan’s Law: !P AND (P OR !Q)**

**Negation law: !P AND (!Q)**

That is to say that !P and P cannot exist at the same time, therefore including it in the or statement is redundant, his simplifies the statement down to !Q

**Statement 2: !P AND !Q**

(3) Prove the following statement using the Direct Method. (*20 pts.*)

If A < B, and B < C, then A < C

A< B

B < C

A < C

A number is inherently greater than another number when it is equal to that number + some positive number. Two positive numbers added together are equal to some other positive number.

B = A + some positive number (N)

C = B + some other positive number (K)

C = A + N + K

N + K are both positive numbers, and are therefore equal to some positive number.

Because C is equal to some A + some positive number, C is greater than A.

(4) Convert the following from the base indicated, to the base requested. (*10 pts. each*)

(4.1) 1410 = ??3

14 % 3 = 2

4 % 3 = 1

1 % 3 = 1

112

(4.2) B216 = ??2

Step 1: base 16 is confusing and I don’t like it, convert to base 10.

B = 11

11\*(16^1) + 2 \*(16^0) = 176 + 2 = 178

Convert the base 10 number into binary the way saint modulus intended.

178 % 2 = 0

89 % 2 = 1

44 % 2 = 0

22 % 2 = 0

11 % 2 = 1

5 % 2 = 1

2 % 2 = 0

1 % 2 = 1

10110010

(5) Add the following two binary numbers together. Take that result, and XOR it with the shown binary number. Then take those results, and NOR it together with the last binary number. (*20 pts.*)

Step 1: 1001101 + 1010

1001101

+

0001010

=

1010111

Step 2: XOR 1011001

XOR means mismatch is one, match is 0

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NUM1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 |
| STEP2 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| XOR | 0 | 0 | 0 | 1 | 1 | 1 | 0 |

0001110

Note: length of the initial string is not reduced in terms of logic comparison.

Step 3: NOR 110110

NOR means only two 0s produce a 1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| NUM1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| STEP3 | 0 | 1 | 1 | 0 | 1 | 1 | 0 |
| NOR | 1 | 0 | 0 | 0 | 0 | 0 | 1 |