# 1 Homework 3

## 1.1 Boolean Algebra

1. Simplify the following expressions using Boolean algebraic laws. Give each step of your simplification and denote which laws you're using for each step. Do not skip or combine steps!

   (a) $A * (\overline{A} + B * B) + (\overline{B + A}) * (\overline{A} + B)$

   **Work:**

   $$A * (\overline{A} + B) + (\overline{B + A}) * (\overline{A} + B) \text{ // Idempotent law}$$
   $$A * B + (\overline{B + A}) * (\overline{A} + B) \text{ // Redundancy law}$$
   $$A * B + \overline{B} * \overline{A} * (\overline{A} + B) \text{ // Demorgan's law}$$
   $$\boxed{\textbf{Answer: } A * B + \overline{B} * \overline{A}}$$

   (b) $\overline{C * B} + (A * B * C) + \overline{A + B + \overline{B}}$

   **Work:**

   $$\overline{C} + \overline{B} + (A * B * C) + \overline{A + C + \overline{B}} \text{ // Demorgan's law}$$
   $$\overline{C} + \overline{B} + (A * B * C) + \overline{A} * \overline{C} * B \text{ // Involution Law}$$
   $$\overline{C} + \overline{B} + A * B * C \text{ // Absorption Law}$$
   $$\overline{C} + \overline{B} + A * B \text{ // Absorption Law}$$
   $$\overline{C} + \overline{B} + A \text{ // Absorption Law}$$
   $$\boxed{\textbf{Answer: } \overline{C} + \overline{B} + A}$$

   (c) $(A + B) * (\overline{A} + C) * (\overline{C} + B)$

   **Work:**

   $$(\overline{A} + C) * (\overline{C} + B) * A + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Distributive Law}$$
   $$(\overline{C} + B) * A * \overline{A} + (\overline{C} + B) * A * C + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Distributive Law}$$
   $$0 + (\overline{C} + B) * A * C + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Complement Law}$$
   $$(\overline{C} + B) * A * C + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Identity Law}$$
   $$A * A * \overline{C} + A * C * B + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Distributive Law}$$
   $$0 + A * C * B + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Complement Law}$$
   $$A * C * B + (\overline{A} + C) * (\overline{C} + B) * B \text{ // Identity Law}$$
   $$A * C * B + B * \overline{A} * \overline{C} + B * \overline{A} * B + (\overline{C} + B) * B * C \text{ // Distributive Law}$$
   $$A * C * B + B * \overline{A} + (\overline{C} + B) * B * C \text{ // Absorption Law}$$
   $$B * (A * C + \overline{A}) + (\overline{C} + B) * B * C \text{ // Distributive Law}$$
   $$B * (C + \overline{A}) + (\overline{C} + B) * B * C \text{ // Absorption Law}$$
   $$B * C + B * \overline{A} + (\overline{C} + B) * B * C \text{ // Distributive Law}$$
   $$B * C + B * \overline{A} \text{ // Absorption Law}$$
   $$\boxed{\textbf{Answer: } B * C + B * \overline{A}}$$

2. Find all solutions of the following Boolean equations without using the truth tables:

(a) $(\overline{A}+C)*(\overline{B}+D+A)*(D+A*\overline{C})*(\overline{D}+A)=1$

**Work:**

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{A}*A+\overline{B}*C+C*D+C*A)*(D+A*\overline{C})*(\overline{D}+A)=1$ //Distributive

$(\overline{A}*\overline{B}+\overline{A}*D+0+\overline{B}*C+C*D+C*A)*(D+A*\overline{C})*(\overline{D}+A)=1$ //Complement

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{B}*C+C*D+C*A)*(D*\overline{D}+\overline{C}*A\overline{D}+\overline{C}*A)=1$

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{B}*C+C*D+C*A)*(0+A*D+\overline{C}*A*\overline{D}+A*\overline{C}*A)=1$

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{B}*C+C*D+C*A)*(D*A+A*\overline{C}*\overline{D}+A*\overline{C}*A)=1$ //Identity

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{B}*C+C*D+C*A)*(D*A+A*\overline{C}*\overline{D}+A*\overline{C})=1$ //Idempotent

$(\overline{A}*\overline{B}+\overline{A}*D+\overline{B}*C+C*D+C*A)*(D*A+A*\overline{C}*(\overline{D}+1))=1$ //Distributive

$(\overline{A}*\overline{B}+\overline{B}*C+C*D+C*A)*(D*A+A*\overline{C})=1$ //Identity Law

$(\overline{A}*\overline{B}*D*A+\overline{A}*D*D*A+C*\overline{B}*D*A+C*D*D*A+C*A*D*A)+\overline{A}*\overline{B}*\overline{C}*A+\overline{A}*D*\overline{C}*A+C*\overline{B}*\overline{C}*A+C*D*\overline{C}*A+C*A*\overline{C}*A=1$ //Distributive

$(0+0+C*\overline{B}*D*A+C*D*D*A+C*A*D*A)+0+0+C*\overline{B}*\overline{C}*A+C*D*\overline{C}*A+C*A*\overline{C}*A=1$

$(0+0+C*\overline{B}*D*A+C*D*D*A+C*A*D*A)+0+0+0+0+0=1$ //Complement

$C*\overline{B}*D*A+C*D*D*A+C*A*D*A=1$ //Identity Law

$C*\overline{B}*D*A+A*C*D+C*A*D=1$ //Idempotent Law

$C*\overline{B}*D*A+A*C*D=1$ //Idempotent Law

$(\overline{B}+1)*A*C*D=1$ //Distributive Law

$A*C*D=1$ //Identity Law

> **Answer:** $A=1,C=1,D=1$
> *B can be anything*

(b) $(((\overline{K}*L*N)*(L*M))+((\overline{K}+L+N)*(K*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})=1$

**Work:**

$(((\overline{K}*L*L*N)+\overline{K}*L*N*M)+((\overline{K}*K*\overline{L}*\overline{M}+L*K*\overline{L}*\overline{M}+N*K*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})=1$ //Distributive Law

$(((\overline{K}*L*L*N)+\overline{K}*L*N*M)+((0+0+K*N*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})=1$ //Complement

$(((\overline{K}*L*L*N)+\overline{K}*L*N*M)+((K*N*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})=1$ //Identity Law

$(\overline{K}*L*L*N+\overline{K}*L*N*M+N*K*\overline{L}*\overline{M})*(\overline{K}+\overline{N})=1$ //Commutative Law

$(\overline{K}*L*L*N+\overline{K}*L*N*M*\overline{K}+K*N*\overline{L}*\overline{M}*\overline{K})+\overline{K}*L*L*N*\overline{N}+\overline{K}*L*N*M*\overline{N}+N*K*\overline{L}*\overline{M}*\overline{N}=1$ //Distributive Law

$(\overline{K}*L*L*N+\overline{K}*L*N*M*\overline{K}+0)+0+0+0=1$ //Complement Law

$(\overline{K}*L*N+\overline{K}*L*N*M+0)+0+0+0=1$ //Idempotent Law

$(\overline{K}*L*N+\overline{K}*L*N*M)=1$ //Identity Law

$(\overline{K}*L*N(1+M))=1$ //Distributive Law

$\overline{K}*L*N=1$ //Identity Law

> **Answer:** $K=0,L=1,N=1$
> *M can be anything*

3. Simplify the following expression by first constructing a truth table, using that truth table to construct a K-map, and then using that K-map to simplify.
$Q = \overline{X} * \overline{Y} * Z + X * Y * \overline{Z} + \overline{X} + Y * \overline{Z} + X * \overline{Y} * \overline{Z}$
**Work:**

**Truth Table:**

| X | Y | Z | Q |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

**K-Map:**

| **X\YZ** | **00** | **01** | **11** | **10** |
|---|---|---|---|---|
| **0** | 0 | 1 | 0 | 1 |
| **1** | 0 | 0 | 1 | 0 |

**Simplified Answer:** $Q = \overline{X} * Z + X * \overline{Y} * \overline{Z}$

## 1.2 Logical Circuits

4. Convert the following truth table into its sum of products representation:

| A | B | C | Output |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |

**Work:**

$$0 \quad 0 \quad 0 \;\big|\; 1 = \overline{A} * \overline{B} * \overline{C}$$

$$0 \quad 1 \quad 0 \; \Big| \; 1 = \overline{A} * B * \overline{C}$$

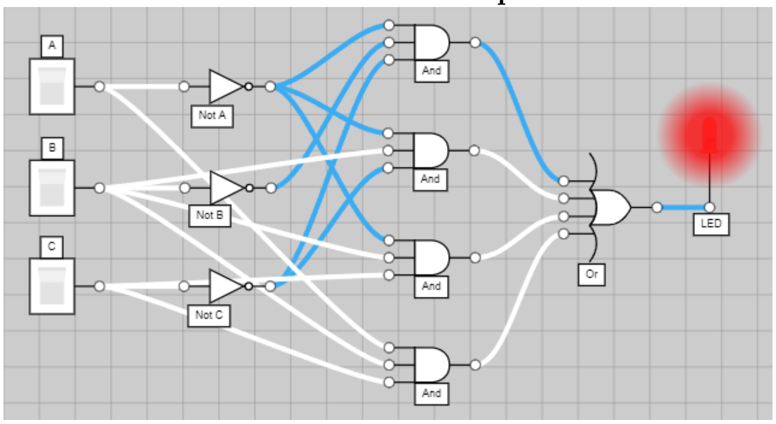
$$0 \quad 1 \quad 1 \; \Big| \; 1 = \overline{A} * B * C$$

$$1 \quad 1 \quad 1 \; \Big| \; 1 = A * B * C$$

$$\boxed{\begin{array}{c} \overline{A} * \overline{B} * \overline{C} + \overline{A} * B * \overline{C} + \overline{A} * B * C + A * B * C \\ \textbf{Simplified Answer: } \overline{A} * \overline{C} + B * C \end{array}}$$

5. Draw a logical circuit diagram that represents the above sum of products expression using OpenCircuits (https://opencircuits.io/). Clearly label all inputs/outputs and all components. Make sure you connect appropriate input components (e.g., buttons, switches, clocks, etc.) and output components (e.g., LEDs, displays, etc.) to facilitate testing of your circuit.Download your diagram using OpenCircuits' "Download" feature, rename it to hw3_SOP.circuit, and submit on Submitty along with your hw3.pdf file.

**Answer: The Circuit below was uploaded to Submitty**

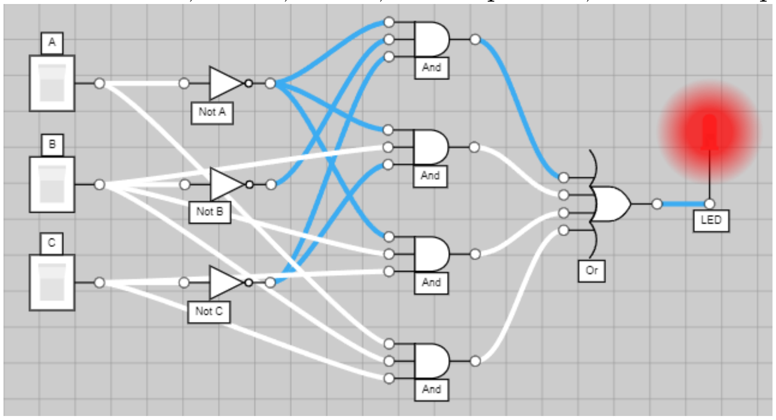



6. Test you circuit by supplying appropriate inputs and observing the expected values of the output. Explain why your set of tests is sufficient to prove that your logical circuit does in fact implement the required Boolean function. For each test, provide a picture (snapshot) of your circuit. Insert all such pictures in the hw3.pdf PDF file. You can download pictures (PNG, JPEG, or PDF) of your circuit diagram using OpenCircuits' "Export Image" feature.

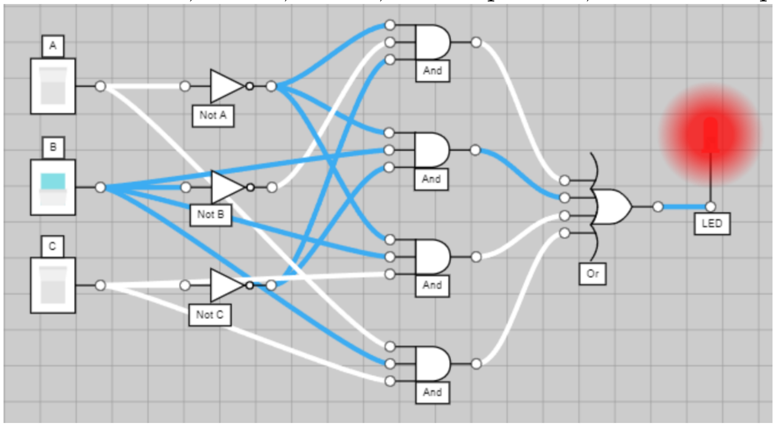
**Answer:**
Below are the images of the circuit being tested. There are a total of 8 test photos, and they cover all the possible cases.

**Images:**

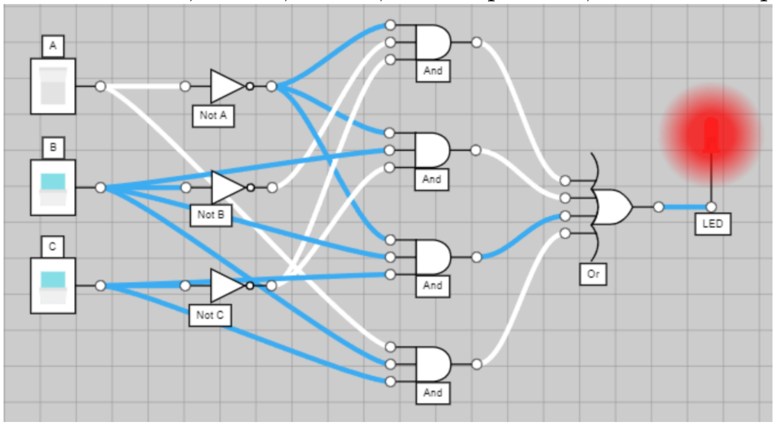
Case 1: A = 0, B = 0, C = 0; the output is 1, which corresponds to the boolean function



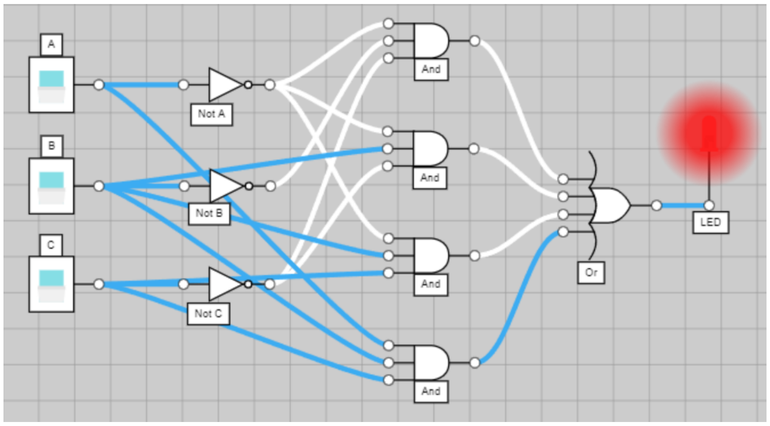

Case 2: A = 0, B = 1, C = 0; the output is 1, which corresponds to the boolean function



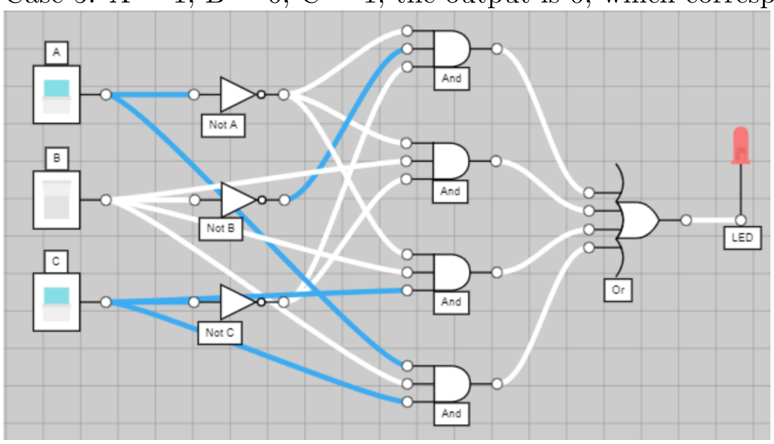

Case 3: A = 0, B = 1, C = 1; the output is 1, which corresponds to the boolean function



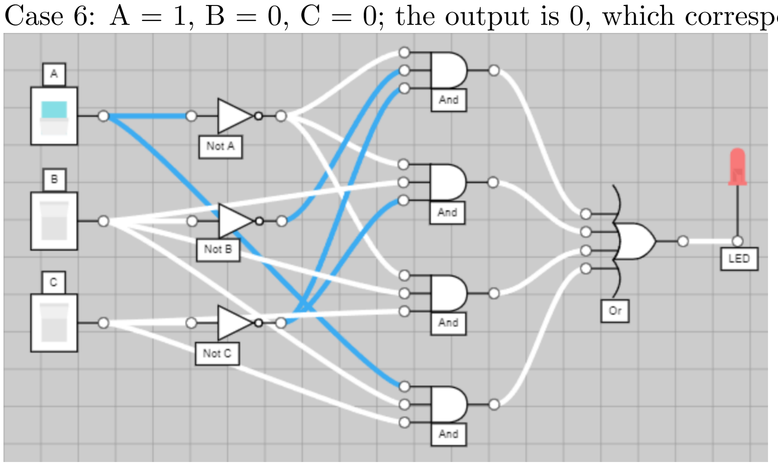

Case 4: A = 1, B = 1, C = 1; the output is 1, which corresponds to the boolean function



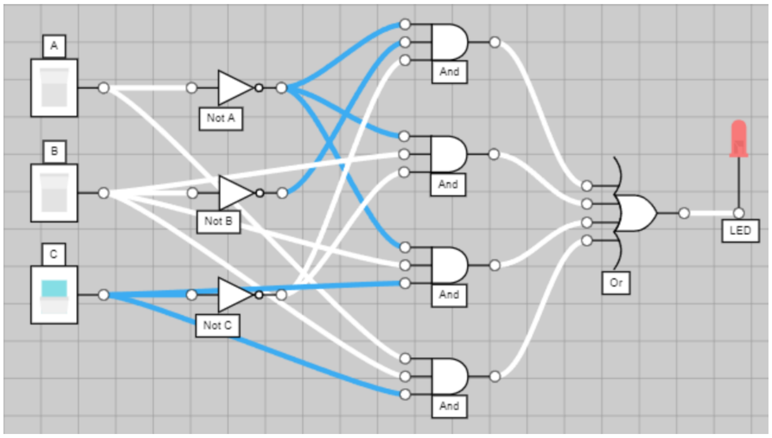

Case 5: A = 1, B = 0, C = 1; the output is 0, which corresponds to the boolean function



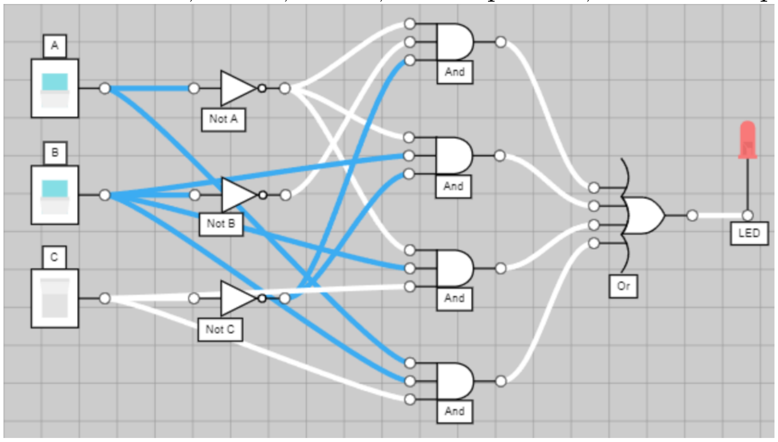

Case 6: A = 1, B = 0, C = 0; the output is 0, which corresponds to the boolean function




Case 7: A = 0, B = 0, C = 1; the output is 0, which corresponds to the boolean function



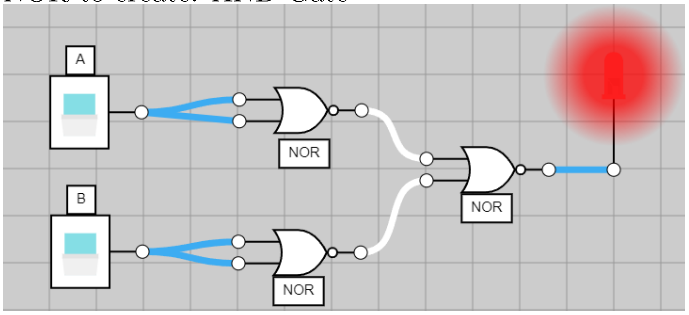

Case 8: A = 1, B = 1, C = 0; the output is 0, which corresponds to the boolean function



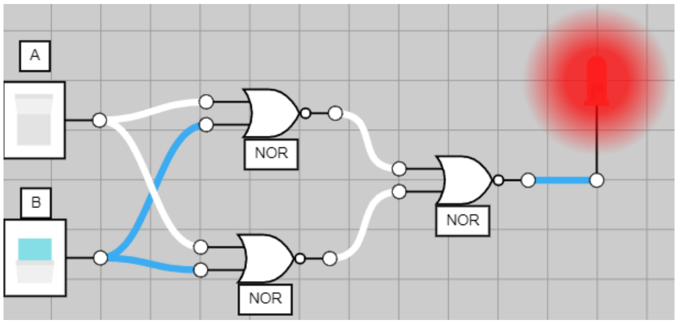

7. Given inputs A and B, show that NOR $\{(\overline{A+B})\}$ is functionally complete by giving logical circuits equivalent to AND $\{(A*B)\}$, OR $\{(A+B)\}$, and NOT $\{(\overline{A})\}$ using only NOR gates in their construction.

**Images:**

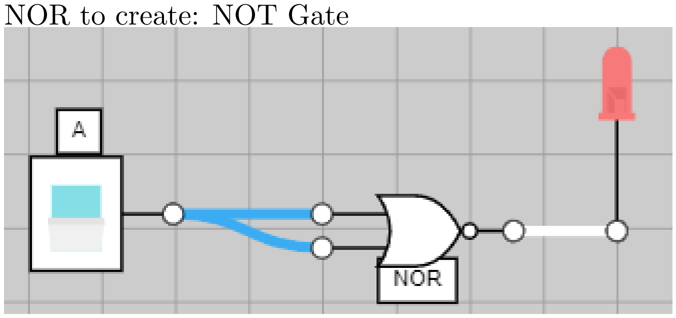
NOR to create: AND Gate




NOR to create: OR Gate




NOR to create: NOT Gate




## 1.3 Numerical Conversions and Arithmetic

8. For each of the following numbers, convert them to their closest single precision IEEE 754 floating point representation. First, denote the binary values of the sign, fraction, and exponent. Then provide a 32-bit hexadecimal value. Show your steps.

   a. 50.4375

   **Work:**

   Sign: 0

   Exp: $5 + 127 = 132 = 10000100$

   Mantissa: $110010.0111 = 1.100100111 * 2^5$

   **Answer:** 0 10000100 10010011100000000000000

   b. 0.0

   **Work:**

   Sign: 0

   Mantissa: 0

   **Answer:** 0 00000000 00000000000000000000000

   c. $-Infinity$

   **Work:**

   Sign: 1 // *Since it's negative*

Exp: 1 *// All 1s - which in binary is 11111111*

Mantissa: 0

**Answer:** 1 11111111 00000000000000000000000

d. 1.0000001 **Work:**

Sign: 0 *// Since it's positive*

Exp: $01111111 = 127$

Mantissa: $1.0000001 = 1.0000001 * 2^0$

Normalized: 00000000000000000000001

**Answer:** 0 01111111 00000000000000000000001

9. For each of the following hexadecimal values, convert them from single precision IEEE 754 floating point representation to decimal rational numbers. You may leave large powers of two in the exponential form, and you may express your answer as a ratio (e.g., $-\frac{5}{8}, \frac{1}{2^{64}}$). Show your steps.

a. $0xc349a000$

**Work:**

In Binary Form: 11000011010010011010000000000000

Sign: 1 *// Since it's negative*

Exp: $10000110 = 134 = 2^1 + 2^2 + 2^7$

The rest of the bits are the mantissa:

$10011010000000000000000 = 2^{-1} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-10} = 589/1024$

Converting: $(-1^1) * (1 + 589/1024) * 2^{134-127} = -(1613/8)$

**Answer:** $-(1613/8)$

b. $0xffe00001$

**Work:**

In Binary Form: 11111111111000000000000000000001

Sign: 1 *// Since it's negative*

Exp: $11111111 = 255 = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^7$

The rest of the bits are the mantissa:

$11000000000000000000001 = 2^{-1} + 2^{-2} + 2^{-23} = 6291457/2^{23}$ Converting:

$(-1)^1 * (1 + 6291457/2^{23}) * 2^{255-127} = -(6291459/2^{104})$

**Answer:** $-(6291459/2^{104})$

c. $0x80000000$

**Work:**

In Binary Form: 10000000000000000000000000000000
Sign: 1 *// Since it's negative*
Exp: 00000000 = 0
The rest of the bits are the mantissa: 00000000000000000000000 = 0
Converting: $(-1)^1 * (1 + 0) * 2^{0-127}$

| **Answer:** $-(1/2^{127})$ |
|---|

d. $0x00400000$
**Work:**

In Binary Form: 00000000010000000000000000000000
Sign: 0 *// Since it's positive*
Exp: 00000000 = 0
The rest of the bits are the mantissa: 10000000000000000000000 $= 2^{-1} = 1/2$
Converting: $(-1)^0 * (1 + 1/2) * 2^{0-127}$

| **Answer:** $3/2^{128}$ |
|---|

10. Give a reason why we use 2's complement representations for negative numbers in computer arithmetic. Give an example of its usage.
**Answer:**

The 2's complement representation is favored in computer arithmetic for negative numbers because it makes the addition and subtraction of positive and negative numbers easier. This method enables the utilization of the same circuitry for both types of numbers, simplifying hardware implementation and boosting efficiency. Consequently, a single set of instructions suffices for performing arithmetic operations, whether they involve unsigned or signed numbers.

An example would be if we take the 8-bit bumbers 10 (00001010 in binary) and -6 (11111010 in binary, two's complement) and add them together, we get 4 (00000100 in binary). This is because 10 + (-6) = 4. Doing this, there is no special handling for negatives.