Homework 3

Danny Zou

11-09-2023

Boolean Algebra

1

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

$A * (\overline{A} + B * B) + (\overline{B} * \overline{A}) * (\overline{A} + B)$ DeMorgan's
$A * (\overline{A} + B) + (\overline{B} * \overline{A}) * (\overline{A} + B)$ Idempotent Law $(B * B = B)$
$(A + (\overline{B} * \overline{A}))(\overline{A} + B)$ Distributive Law (Inversed)
$(A + (\overline{A} * \overline{B}))(\overline{A} + B)$ Commutative Law
$(A + \overline{B})(\overline{A} + B)$ Redundancy Law $(A + (\overline{A} * \overline{B}) = A + \overline{B})$
$A(\overline{A} + B) + \overline{B}(\overline{A} + B)$ Distributive Law
$(A * \overline{A} + A * B) + \overline{B}(\overline{A} + B)$ Distributive Law
$(0 + A * B) + \overline{B}(\overline{A} + B)$ Inverse Law $(A * \overline{A} = 0)$
$(A * B) + \overline{B}(\overline{A} + B)$ Identity Law $(0 + A * B = A * B)$
$(A * B) + (\overline{B} * \overline{A} + \overline{B} * B)$ Distributive Law
$(A * B) + (\overline{B} * \overline{A} + 0)$ Inverse Law $(\overline{B} * B = 0)$
$(A * B) + (\overline{B} * \overline{A})$ Identity Law $(\overline{B} * \overline{A} + 0 = \overline{B} * \overline{A})$

$\boxed{A * B + \overline{B} * \overline{A}}$

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

$\overline{C} + \overline{B} + (A * B * C) + \overline{A + C + \overline{B}}$ DeMorgan's
$\overline{C} + (A * B * C) + \overline{B} + \overline{A + C + \overline{B}}$ Commutative Law
$\overline{C} + (C * A * B) + \overline{B} + \overline{A + C + \overline{B}}$ Commutative Law
$\overline{C} + (A * B) + \overline{B} + \overline{A + C + \overline{B}}$ Redundancy Law $(\overline{C} + (C * A * B) = \overline{C} + (A * B))$
$\overline{C} + \overline{B} + (A * B) + \overline{A + C + \overline{B}}$ Commutative Law
$\overline{C} + \overline{B} + (B * A) + \overline{A + C + \overline{B}}$ Commutative Law
$\overline{C} + (\overline{B} + A) + \overline{A + C + \overline{B}}$ Redundancy Law $(\overline{B} + (B * A) = \overline{B} + A)$
$\overline{C} + (\overline{B} + A) + \overline{A} * \overline{C} * \overline{\overline{B}}$ DeMorgan's
$\overline{C} + (\overline{B} + A) + \overline{A} * \overline{C} * B$ Double Negation Law
$\overline{C} + (\overline{A} * \overline{C} * B) + (\overline{B} + A)$ Commutative Law
$\overline{C} + (\overline{C} * \overline{A} * B) + (\overline{B} + A)$ Commutative Law
$\overline{C} + (\overline{B} + A)$ Absorption Law $(\overline{C} + (\overline{C} * \overline{A} * B) = \overline{C})$

$\boxed{\overline{C} + \overline{B} + A}$

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

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

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

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

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

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

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

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

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

$((0*\overline{C})+(A*C*\overline{C})+(B*\overline{A}*\overline{C})+(B*C*\overline{C}))+$
$((A*\overline{A}*B)+(A*C*B)+(B*\overline{A})+(B*C))$ Inverse Law

$((0*\overline{C})+(A*0)+(B*\overline{A}*\overline{C})+(B*C*\overline{C}))+$
$((A*\overline{A}*B)+(A*C*B)+(B*\overline{A})+(B*C))$ Inverse Law

$((0*\overline{C})+(A*0)+(B*\overline{A}*\overline{C})+(B*0))+$
$((A*\overline{A}*B)+(A*C*B)+(B*\overline{A})+(B*C))$ Inverse Law

$((0*\overline{C})+(A*0)+(B*\overline{A}*\overline{C})+(B*0))+$
$((0*B)+(A*C*B)+(B*\overline{A})+(B*C))$ Inverse Law

$((0)+(0)+(B*\overline{A}*\overline{C})+(0))+$
$((0)+(A*C*B)+(B*\overline{A})+(B*C))$ Law of Zeros (Too much to do them one by one ...)

$(B*\overline{A}*\overline{C})+(A*C*B)+(B*\overline{A})+(B*C)$ Identity Law (Too much to do them one by one...)

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

$(B*\overline{A})+(A*C*B)+(B*C)$ Absorption Law $((B*\overline{A})+(B*\overline{A}*\overline{C})=(B*\overline{A}))$

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

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

$(B*\overline{A})+(B*C)$ Absorption Law $((B*C)+(B*C*A)=(B*C))$

$B*(\overline{A}+C)$ Distributive Law (Inversed)

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

2

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

$(\overline{A}+C)*(\overline{B}+D+A)*(D+\overline{\overline{A}+\overline{\overline{C}}})*(\overline{D}+A)$ DeMorgan's
$(\overline{A}+C)*(\overline{B}+D+A)*(D+\overline{\overline{A}+C})*(\overline{D}+A)$ Double Negation Law
$(\overline{A}+C)*(D+\overline{\overline{A}+C})*(\overline{B}+D+A)*(\overline{D}+A)$ Commutative Law
$(\overline{A}+C)*(\overline{\overline{A}+C}+D)*(\overline{B}+D+A)*(\overline{D}+A)$ Commutative Law
$((\overline{A}+C)*D)*(\overline{B}+D+A)*(\overline{D}+A)$ Redundancy Law $((\overline{A}+C)*(\overline{\overline{A}+C}+D)=(\overline{A}+C)*D)$
$(\overline{A}+C)*D*(D+A+\overline{B})*(\overline{D}+A)$ Commutative Law
$(\overline{A}+C)*D*(\overline{D}+A)$ Absorption Law $(D*(D+A+\overline{B})=D)$
$(\overline{A}+C)*(D*A)$ Redundancy Law $(D*(\overline{D}+A)=D*A)$
$A*(\overline{A}+C)*D$ Commutative Law
$A*C*D$ Redundancy Law $(A*(\overline{A}+C)=A*C)$
$A*C*D=1$

A = 1, C = 1, D = 1, B = 0 or 1

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

$((\overline{K}*L*(L+M)*N)+((\overline{K}+L+N)*(K*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})$ Commutative Law
$((\overline{K}*L*N)+((\overline{K}+L+N)*(K*\overline{L}*\overline{M})))*(\overline{K}+\overline{N})$ Absorption Law $(L*(L+M)=L)$
$((\overline{K}*L*N)+(K*(\overline{K}+L+N)*\overline{L}*\overline{M}))*(\overline{K}+\overline{N})$ Commutative Law
$((\overline{K}*L*N)+(K*(L+N)*\overline{L}*\overline{M}))*(\overline{K}+\overline{N})$ Redundancy Law $(K*(\overline{K}+L+N)=K*(L+N))$
$((\overline{K}*L*N)+(K*\overline{L}*(L+N)*\overline{M}))*(\overline{K}+\overline{N})$ Commutative Law
$((\overline{K}*L*N)+(K*\overline{L}*N*\overline{M}))*(\overline{K}+\overline{N})$ Redundancy Law $(\overline{L}*(L+N)=L*N)$
$N*((\overline{K}*L)+(K*\overline{L}*\overline{M}))*(\overline{K}+\overline{N})$ Distributive Law (Inversed)
$N*(\overline{K}+\overline{N})*((\overline{K}*L)+(K*\overline{L}*\overline{M}))$ Commutative Law
$N*(\overline{N}+\overline{K})*((\overline{K}*L)+(K*\overline{L}*\overline{M}))$ Commutative Law
$N*\overline{K}*((\overline{K}*L)+(K*\overline{L}*\overline{M}))$ Redundancy Law $(N*(\overline{N}+\overline{K})=N*\overline{K})$
$(\overline{K}*L*N*\overline{K})+(K*\overline{L}*\overline{M}*N*\overline{K})$ Distributive Law
$(\overline{K}*\overline{K}*L*N)+(K*\overline{L}*\overline{M}*N*\overline{K})$ Commutative Law
$(\overline{K}*L*N)+(K*\overline{L}*\overline{M}*N*\overline{K})$ Idempotent Law
$(\overline{K}*L*N)+(K*\overline{K}*\overline{L}*\overline{M}*N)$ Commutative Law
$(\overline{K}*L*N)+(0*\overline{L}*\overline{M}*N)$ Inverse Law
$(\overline{K}*L*N)+(0)$ Law of Zeros
$(\overline{K}*L*N)$ Identity Law
$\overline{K}*L*N=1$

K = 0, L = 1, N = 1, M = 0 or 1

3

Truth Table

X	Y	Z	Q
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

K-Map

Z	**XY**			
	00	**01**	**11**	**10**
0	0	1	1	1
1	1	0	0	0

Z	**XY**			
	00	**01**	**11**	**10**
0	0	1	1	1
1	1	0	0	0

The merge shown by the yellow means "if Z is 0 and Y is 1, it doesn't matter what the value of X is."
In the function, it expresses – $X * Y * \overline{Z} + \overline{X} * Y * \overline{Z} = Y * \overline{Z}$

Z	**XY**			
	00	**01**	**11**	**10**
0	0	1	1	1
1	1	0	0	0

The merge shown by the yellow means "if Z is 0 and X is 1, it doesn't matter what the value of Y is."
In the function, it expresses – $X * Y * \overline{Z} + X * \overline{Y} * \overline{Z} = X * \overline{Z}$

After applying those two reductions, we get

Original : $\overline{X} * \overline{Y} * Z + X * Y * \overline{Z} + \overline{X} * Y * \overline{Z} + X * \overline{Y} * \overline{Z}$

Reduced:

$$\boxed{Y * \overline{Z} + X * \overline{Z} + \overline{X} * \overline{Y} * Z}$$

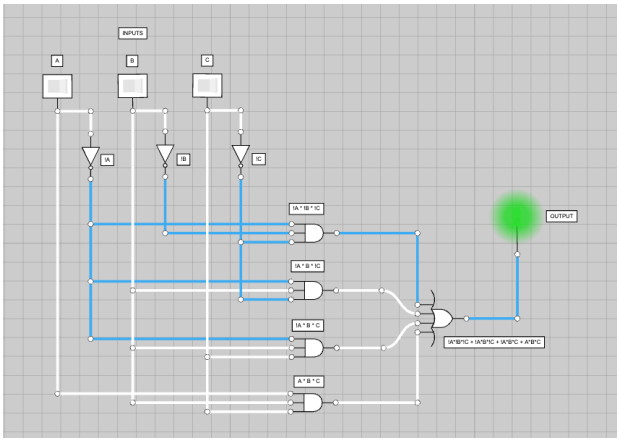
Logical Circuits

4

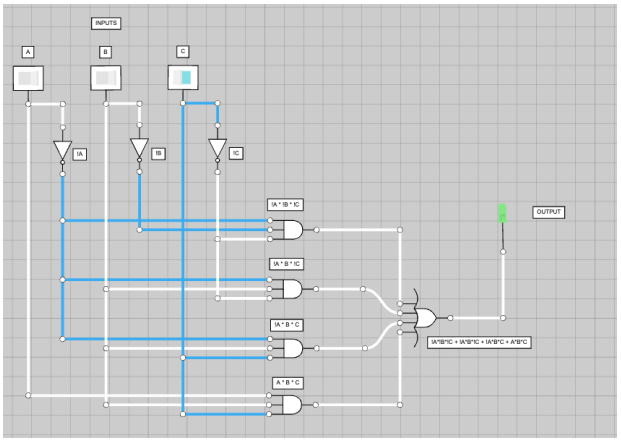
To obtain the sums of products representation, we look for the rows where the output is 1 and then form a product term for each row.
From the truth table, we get

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

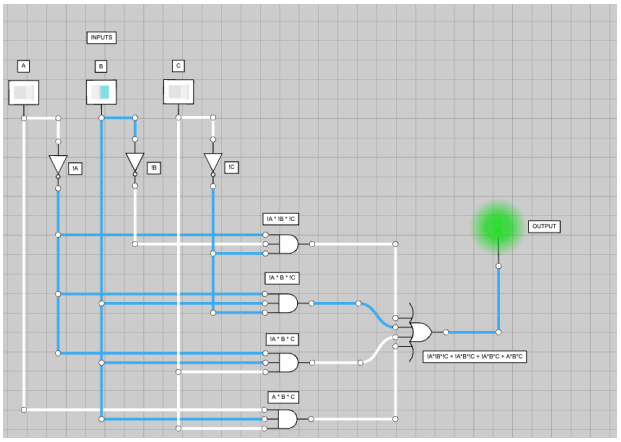
6



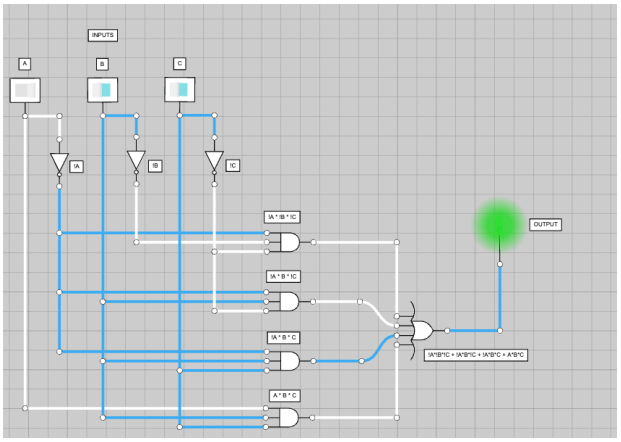
When input A = 0, B = 0, C = 0, the output is 1, like in the truth table (Also satisfies expression $(\overline{A} * \overline{B} * \overline{C})$ as it should)



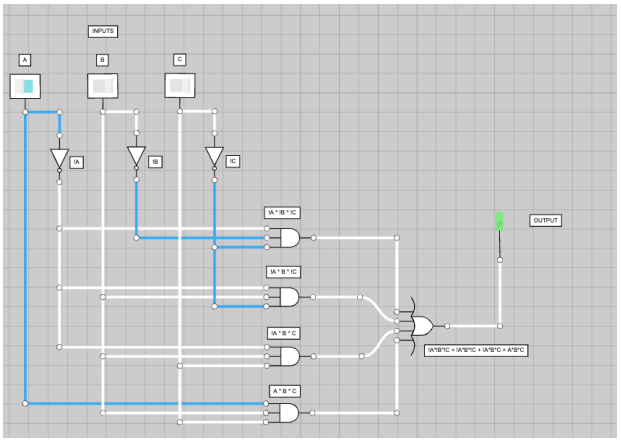
When input A = 0, B = 0, C = 1, the output is 0, like in the truth table (Doesn't satisfy none of the expressions)



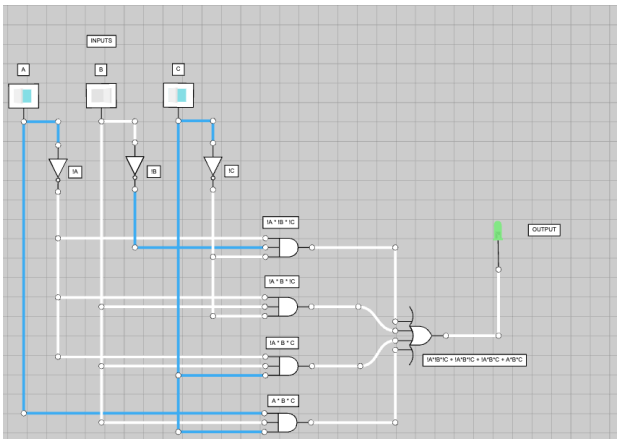
When input A = 0, B = 1, C = 0, the output is 1, like in the truth table (Also satisfies expression $(\overline{A} * B * \overline{C})$)



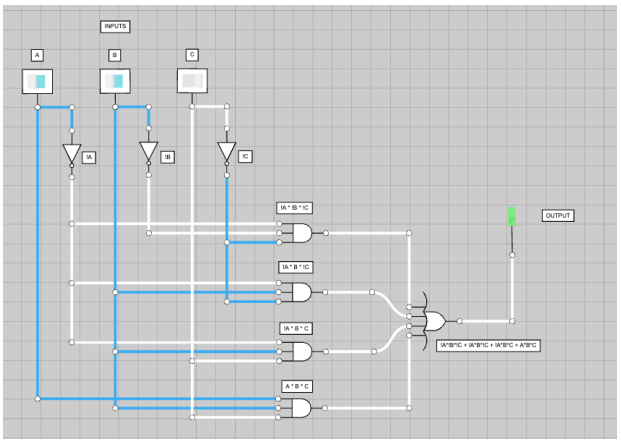
When input A = 0, B = 1, C = 1, the output is 1, like in the truth table (Also satisfies expression $(\overline{A} * B * C)$)



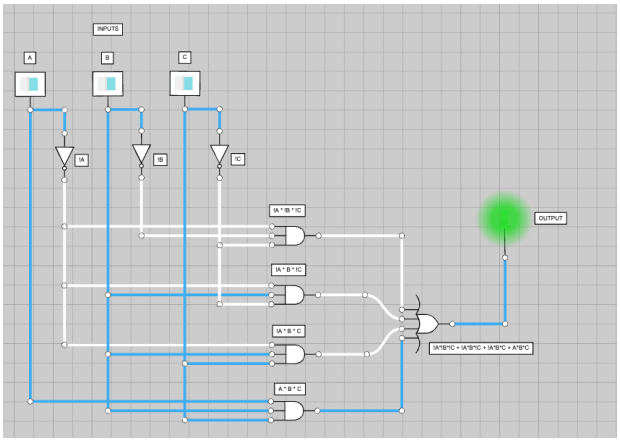
When input A = 1, B = 0, C = 0, the output is 0, like in the truth table (Doesn't satisfy none of the expressions)



When input A = 1, B = 0, C = 1, the output is 0, like in the truth table (Doesn't satisfy none of the expressions)



When input A = 1, B = 1, C = 0, the output is 0, like in the truth table (Doesn't satisfy none of the expressions)



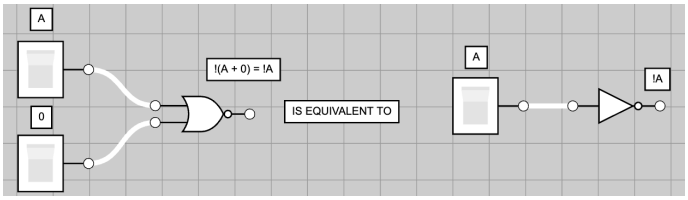
When input A = 1, B = 1, C = 1, the output is 1, like in the truth table (Also satisfies expression $(A * B * C)$)

After going through all possible combination of inputs, I find that the expected output with each different set of inputs is exactly like it is in the truth table. When the output is 1, the appropriate AND components also lights up, matching the appropriate expression in the boolean function (i.e., $(A * B * C)$ is an expression that lights up when A = 1, B = 1, C = 1). Through these findings, I believe there is sufficient proof that my logical circuit does in fact implement the required Boolean function, and that the boolean function also correctly represents the truth table.

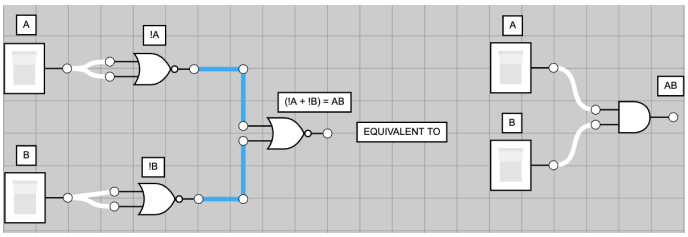
7

First of all

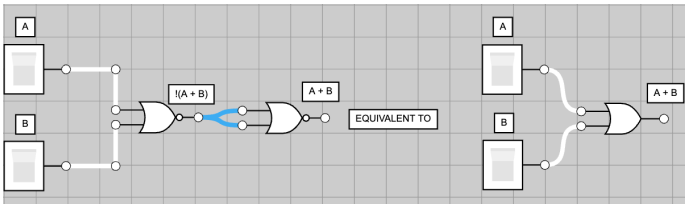
And



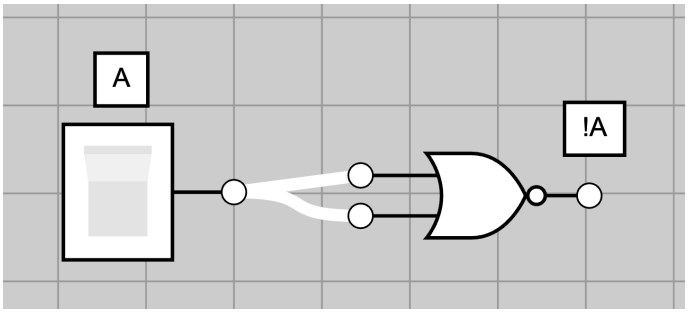
Using these logics, we can implement **AND{(A*B)}** using only NOR Gates by



We can also implement **OR{(A+B)}** using only NOR Gates by



And finally implement **NOT{$\overline{A}$}** simply as



Thus we have shown **NOR{$\overline{(A+B)}$}** is functionally complete by giving the logical circuit equivalents to each AND(A*B), OR(A+B) and NOT($\overline{A}$).

Numerical Conversions and Arithmetic

8

50.4375

$\underline{50}.4375$
$50 = \underline{25} * 2 + \mathbf{0}$
$25 = \underline{12} * 2 + \mathbf{1}$
$12 = \underline{6} * 2 + \mathbf{0}$
$6 = \underline{3} * 2 + \mathbf{0}$
$3 = \underline{1} * 2 + \mathbf{1}$
$1 = 0 * 2 + \mathbf{1}$
We get **110010**, which is 50 in binary representation

$50.\underline{4375}$
$.4375 * 2 = \mathbf{0}.875$
$.0875 * 2 = \mathbf{1}.75$
$.75 * 2 = \mathbf{1}.5$
$.5 * 2 = \mathbf{1}.0$
We get **0111**, which is .4375 in binary representation

Combining, we get – **110010.0111**
Convert to scientific notation, we get – $\mathbf{1.\underline{100100111}} * \mathbf{2^{\underline{5}}}$
Where **100100111** is the mantisa
And **5** is the exponent
Convert the exponent to the bias(127 + 5 = **132**) in binary, we get **1000 0100**
We have everything we need, lets list them out

Sign = **0**, because positive
Exponent Bias Binary = **1000 0100**
Mantisa/Fraction = **10010011100000000000000**
Combine to get Single Precision IEEE 754 Floating Point Representation
Where Sign = 1 bit, Exponent = 8 bits, Mantisa/Fraction = 23 bits

0100 0010 0100 1001 1100 0000 0000 0000

Get the hexidecimal value by converting each set of 4 binaries to respective value

32-bit Hexadecimal – 0x4249c000

0.0

Sign = **0**, because positive
Exponent = 0, Bias Binary = **0000 0000**
Mantisa/Fraction = **00000000000000000000000**
Combine to get Single Precision IEEE 754 Floating Point Representation
Where Sign = 1 bit, Exponent = 8 bits, Mantisa/Fraction = 23 bits

0000 0000 0000 0000 0000 0000 0000 0000

Get the hexidecimal value by converting each set of 4 binaries to respective value

32-bit Hexadecimal – 0x00000000

-Infinity

Sign = **1**, because Negative
Exponent = 128, Bias(128 + 127 = **255**) Binary = **1111 1111**
Mantisa/Fraction = **00000000000000000000000**
Combine to get Single Precision IEEE 754 Floating Point Representation
Where Sign = 1 bit, Exponent = 8 bits, Mantisa/Fraction = 23 bits

1111 1111 1000 0000 0000 0000 0000 0000

Get the hexidecimal value by converting each set of 4 binaries to respective value

32-bit Hexadecimal – 1xff800000

1.0000001

1.0000001
$1 = 0 * 2 + \mathbf{1}$
We get **1**, which is 1 in binary representation

1.0000001
$.0000001 * 2 = \mathbf{0}.0000002$
$.0000002 * 2 = \mathbf{0}.0000004$
$.0000004 * 2 = \mathbf{0}.0000008$
$.0000008 * 2 = \mathbf{0}.0000016$
$.0000016 * 2 = \mathbf{0}.0000032$
$.0000032 * 2 = \mathbf{0}.0000064$
$.0000064 * 2 = \mathbf{0}.0000128$
$.0000128 * 2 = \mathbf{0}.0000256$
$.0000256 * 2 = \mathbf{0}.0000512$
$.0000512 * 2 = \mathbf{0}.0001024$
$.0001024 * 2 = \mathbf{0}.0002048$
$.0002048 * 2 = \mathbf{0}.0004096$
$.0004096 * 2 = \mathbf{0}.0008192$
$.0008192 * 2 = \mathbf{0}.0016384$
$.0016384 * 2 = \mathbf{0}.0032768$
$.0032768 * 2 = \mathbf{0}.0065536$
$.0065536 * 2 = \mathbf{0}.0131072$
$.0131072 * 2 = \mathbf{0}.0262144$
$.0262144 * 2 = \mathbf{0}.0524288$
$.0524288 * 2 = \mathbf{0}.1048576$
$.1048576 * 2 = \mathbf{0}.2097152$
$.2097152 * 2 = \mathbf{0}.4194304$
$.4194304 * 2 = \mathbf{0}.8388608$
$.8388608 * 2 = \mathbf{1}.6777216$
We get $\overline{\mathbf{00000000000000000000001}}$, which is 0.0000001 in binary representation

Combining, we get – $\mathbf{1}.\overline{\mathbf{00000000000000000000001}}$
Convert to scientific notation, we get – $\mathbf{1}.\underline{\mathbf{00000000000000000000001}} * \mathbf{2}^{\underline{\mathbf{0}}}$
Where **00000000000000000000001** is the mantisa/fraction
And **0** is the exponent
Convert the exponent to the bias(127 + 0 = **127**) in binary, we get **0111 1111**
We have everything we need, lets list them out

Sign = **0**, because positive
Exponent Bias Binary = **0111 1111**
Mantisa/Fraction = **00000000000000000000001**

Combine to get Single Precision IEEE 754 Floating Point representation
Where Sign = 1 bit, Exponent = 8 bits, Mantisa/Fraction = 23 bits

0011 1111 1000 0000 0000 0000 0000 0001

Get the hexidecimal value by converting each set of 4 binaries to respective value

32-bit Hexadecimal – 0x3f800001

9

0xc349a000

IEEE 7354 Representation – **1100 0011 0100 1001 1010 0000 0000 0000**
Sign = 1
Exponent Bias Binary = **1000 0110**
Mantisa/Fraction = **10010011010000000000000**
Exponent Bias Decimal Value = 134
Exponent = 134 - 127 = **7**
Fraction Value From Mantisa = $2^{-1} + 2^{-4} + 2^{-7} + 2^{-8} + 2^{-10} = \frac{589}{1024}$

Equation to plug into $-1^{Sign} * (1 + (FractionValue)) * 2^{Exponent}$
Plug in, we get
$-1^1 * (1 + \frac{589}{1024}) * 2^7 = -(\frac{1613}{1024}) * 2^7 = -201.625$

-201.625

0xffe00001

IEEE 7354 Representation – **1111 1111 1110 0000 0000 0000 0000 0001**
Sign = 1
Exponent Bias Binary = **1111 1111**
Mantisa/Fraction = **11000000000000000000001**
Because the exponent field is all 1s and the fraction field is non-zero, we can conclude what we are trying to find is **NaN**

NaN(Not a Number)

0x80000000

IEEE 7354 Representation – **1000 0000 0000 0000 0000 0000 0000 0000**
Sign = 1
Exponent Bias Binary = **0000 0000**
Mantisa/Fraction = **00000000000000000000000**
Both Fraction Value and Exponent Value is 0, so Decimal Value = -0, negative because sign is 1

-0

0x00400000

IEEE 7354 Representation – **0000 0000 0100 0000 0000 0000 0000 0000**

Sign = 0

Exponent Bias Binary = **0000 0000**

Mantisa/Fraction = **10000000000000000000000**

Exponent Bias Decimal Value = 0

Fraction Value From Mantisa = $2^{-1} = \frac{1}{2}$

Because Fraction Value is non-zero and the Exponent Value is 0, the number is **subnormal**

We can then use this formula $-1^{Sign} * (FractionValue) * 2^{-126}$

Plug in, we get

$-1^0 * (\frac{1}{2}) * 2^{-126} = \frac{1}{2} * 2^{-126} = \frac{1}{2^{127}}$ $\boxed{\frac{1}{2^{127}}}$

10

The use of 2's complement representation for negative numbers in computer arithmetic is primarily due to its simplicity and efficiency in performing arithmetic operations, especially addition and subtraction. One such reason is.

Using 2's complement simplifies arithmetic operations because addition and subtraction can be performed using the same hardware for both positive and negative numbers without the need for separate logic for subtraction.

Another reason is that 2's complement naturally supports modular arithmetic, making it easier to perform operations like addition, subtraction, and multiplication within a finite range.

For example: Let's add 3 and -5

3 in binary is **0000 0011**

-5 in binary is **1111 1011**

```
  0000 0011
+ 1111 1011
  1111 1110
```

This showcases how addition between positive and negative numbers is performed using the same binary arithmetic, making the process simpler and more efficient for computers.