Design Assignment 3 – Fixed Point Oscillator

Josh Tobia

Linh Ngo

Brian Dawson

ELC 411 - Embedded Systems

The College of New Jersey

Department of Electrical and Computer Engineering (ECE)



**Date Submitted: 11/15/19**

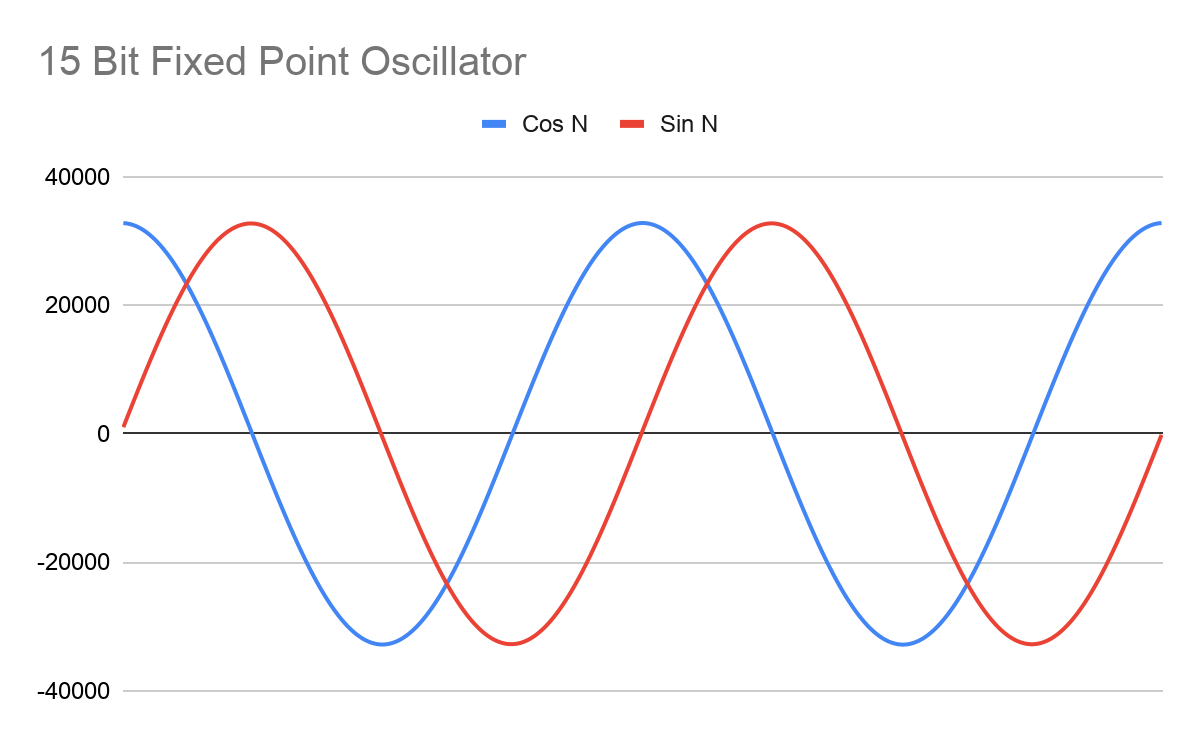
Group Members:

Josh Tobia - Worked on creating code and project

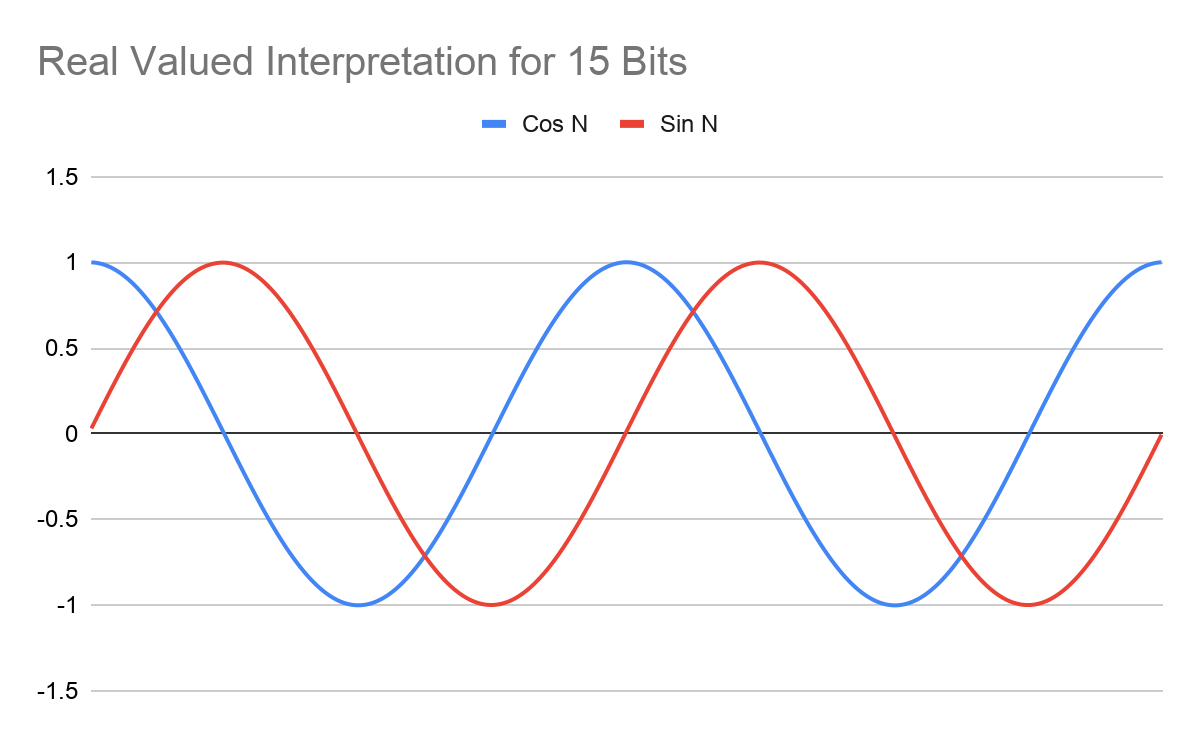
Linh Ngo - Worked on creating code and project

Brian Dawson - Worked on creating code and project

**Fixed N of 15:**

****

*Figure 1: 15 Bit Fixed Point Oscillator*

****

*Figure 2: 15 Bit Real Valued Interpretation*

What is the range of values?

The range of values for the real value interpretation for 15 bits is from -1.00061037

to 1.000549333.

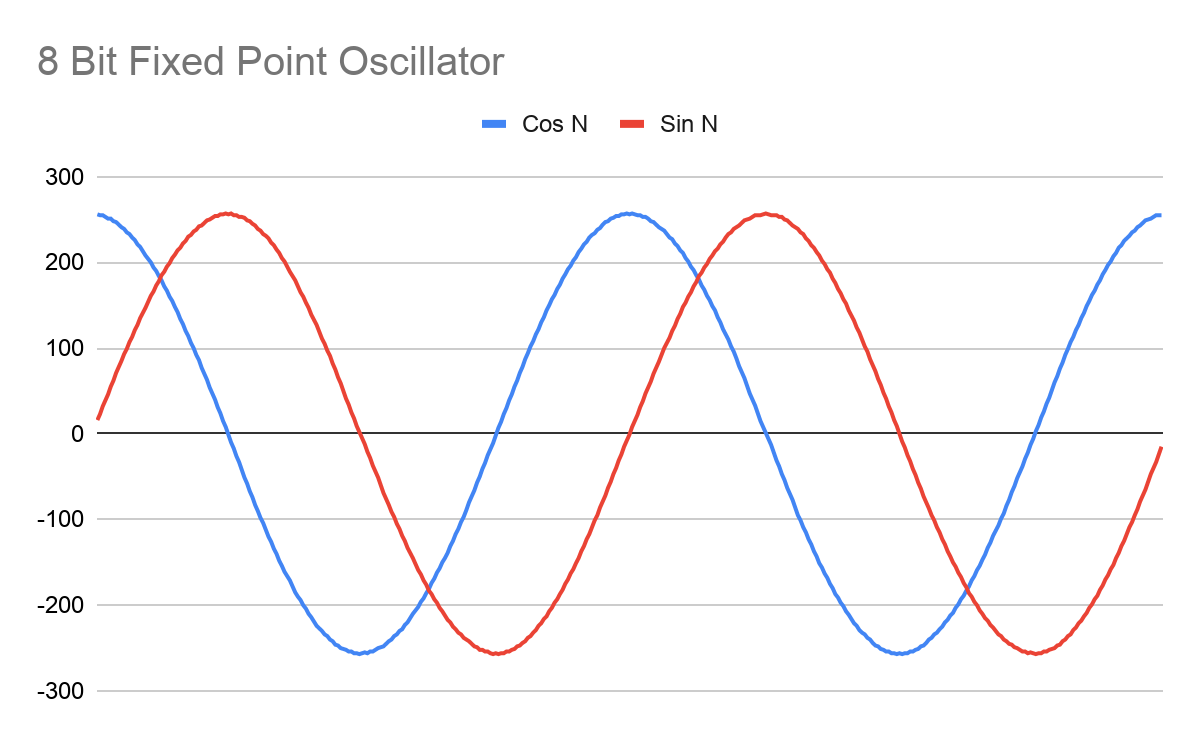
Is this what you would expect?

This is what you would expect as the numbers are slightly above 1 and slightly below -1.

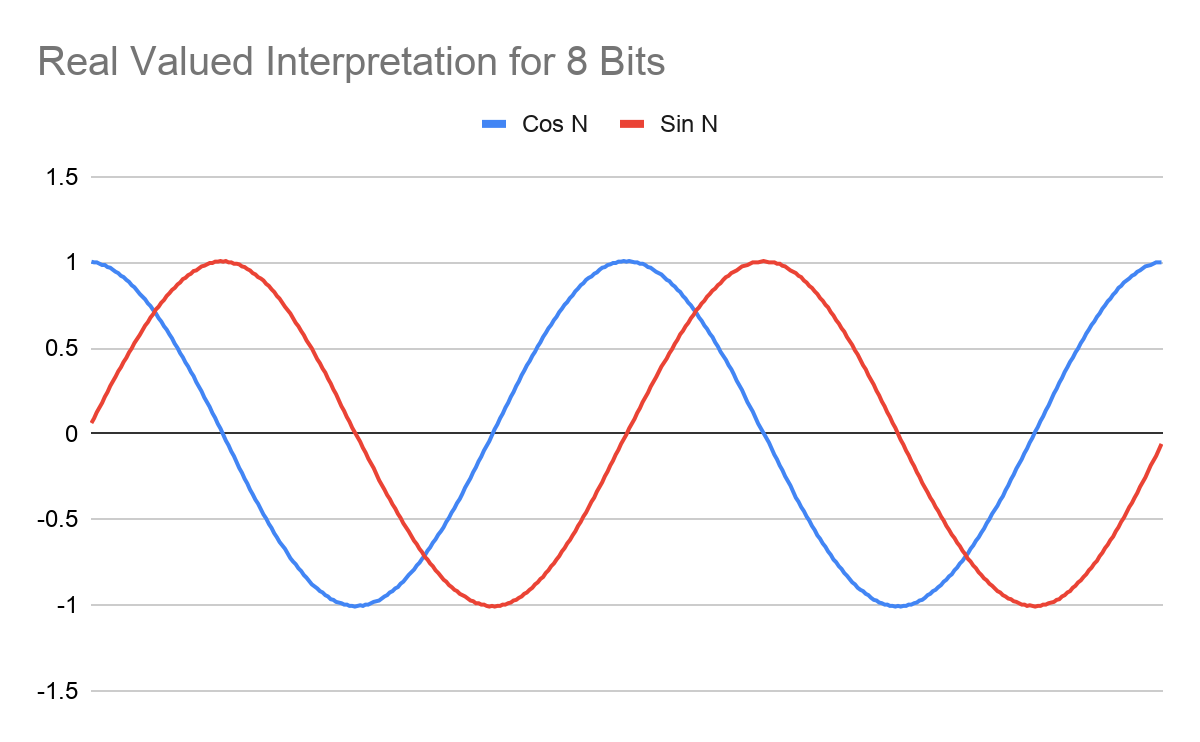
What is the minimum number of integer bits that could be used, and what would the total number of bits for fixed point be in that case?

You would need a minimum of 1 integer bit and 15 bits for fixed point.

**Fixed N of 8:**

****

*Figure 3: 8 Bit Fixed Point Oscillator*

****

*Figure 4: Real Valued Interpretation of 8 Bits*

What is the range of values?

The range of values for the real value interpretation for 8 bits is from -1.007843137 to 1.007843137.

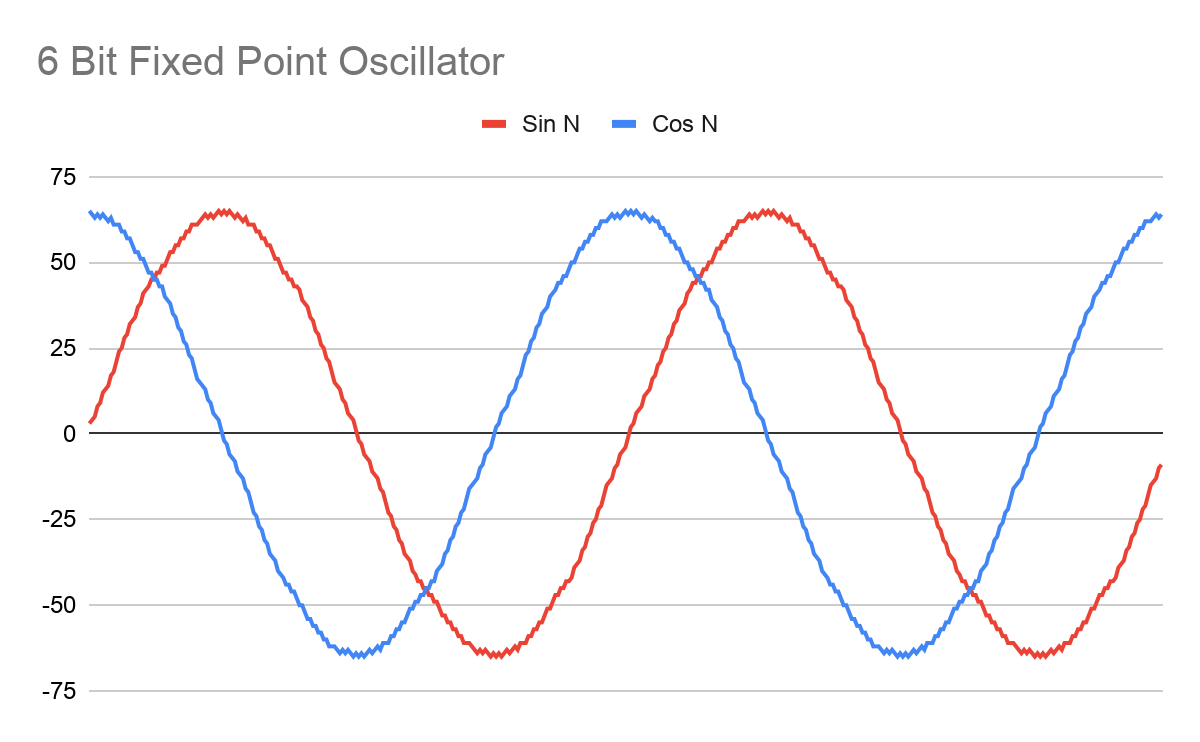
Is this what you would expect?

This is what you would expect as the numbers go from slightly below -1 to slightly above 1.

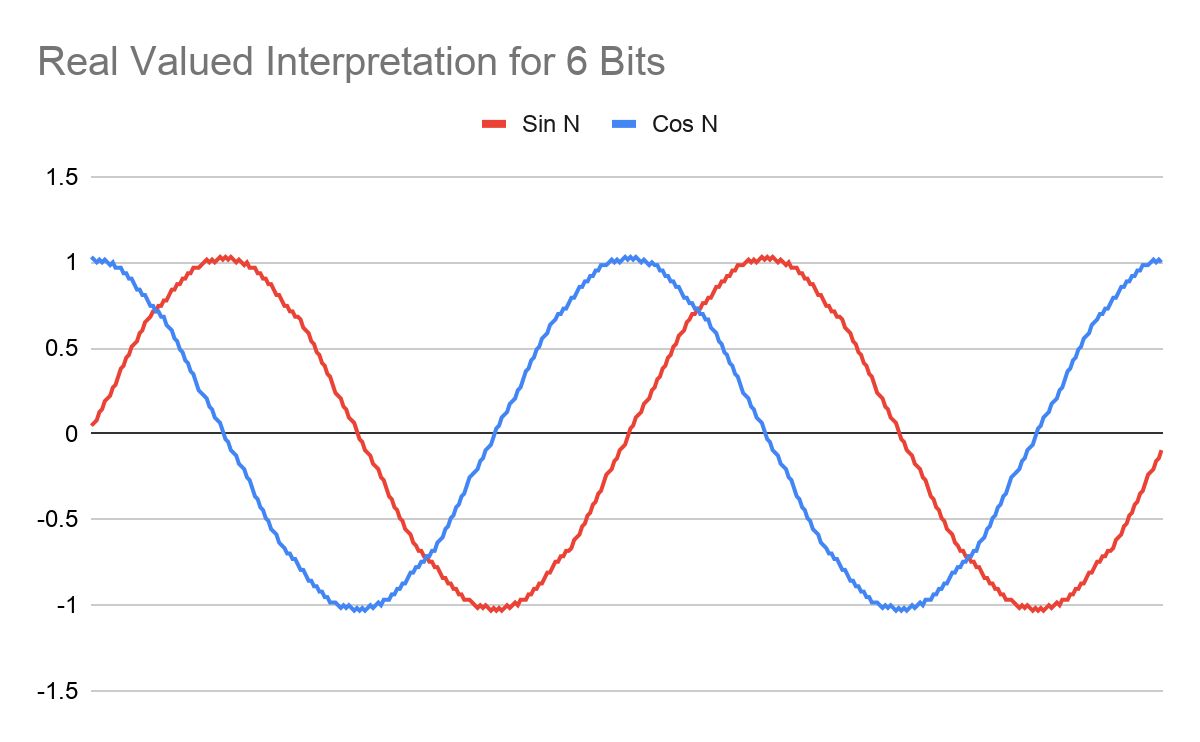
What is the minimum number of integer bits that could be used, and what would the total number of bits for fixed point be in that case?

The minimum number of integer bits would be 1 and the total number of bits for fixed point would be 8.

**Fixed N of 6:**

****

*Figure 5: 6 Bit Fixed Point Oscillator*

****

*Figure 6: Real Valued Interpretation of 6 Bits*

What is the range of values?

The range of values for real valued interpretation for a 6-bit range is -1.30746032 to 1.30746032.

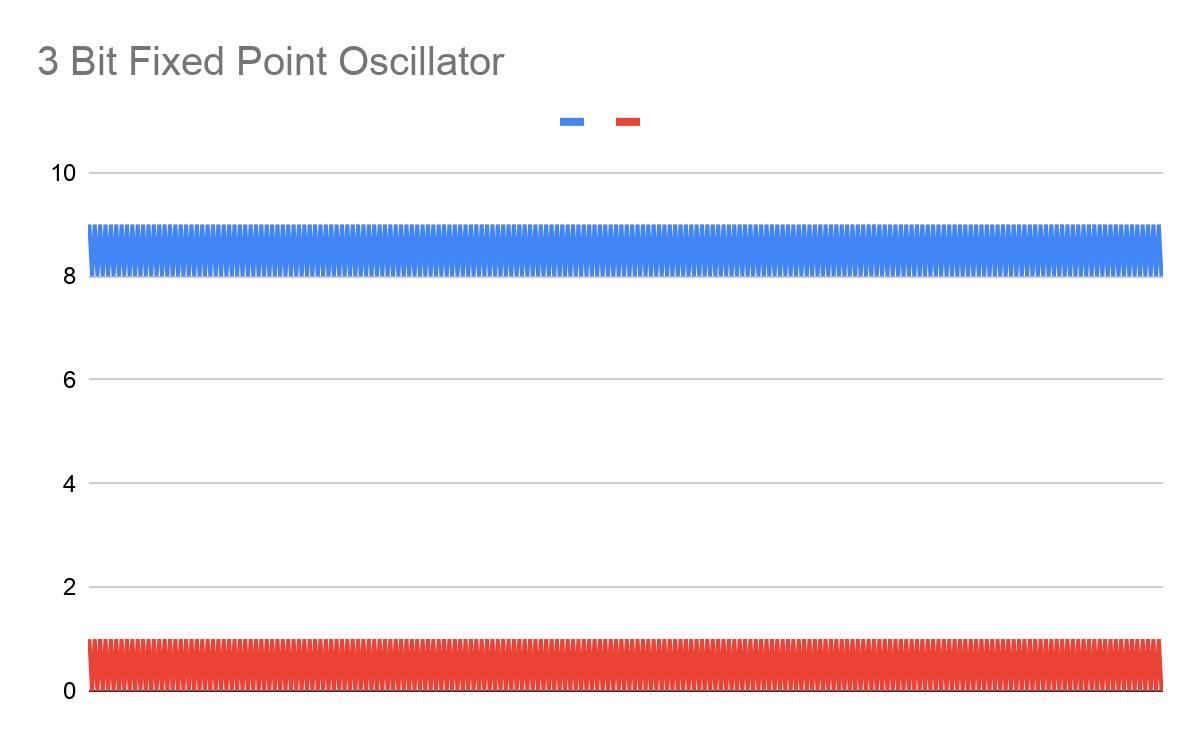
Is this what you would expect?

This is what you would expect as the numbers go from below -1 to above 1. However, the range is greater than previous examples.

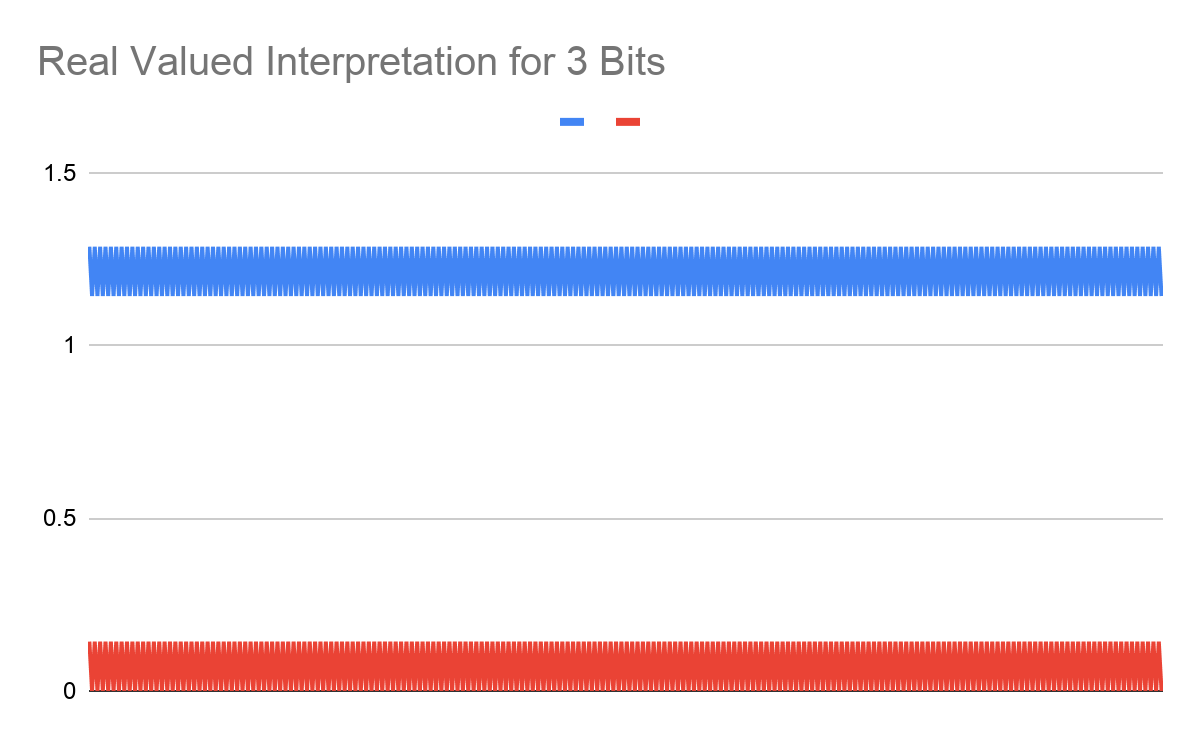
What is the minimum number of integer bits that could be used, and what would the total number of bits for fixed point be in that case?

The minimum number of integer bits is 1 and the total number of bits for fixed point would be 6.

**Fixed N of 3**

****

*Figure 7: 3 Bit Fixed Point Oscillator*

****

*Figure 8: Real Valued Interpretation of 3 Bits*

What is the range of values?

The range of values for real valued interpretation for a 3-bit range is 0 to 1.285714286

Is this what you would expect?

This is not what we would expect. The reason that the data and graphs are not representative of a sine/cosine wave is because the function is collapsing.

What is the minimum number of integer bits that could be used, and what would the total number of bits for fixed point be in that case?

The minimum number of integer bits would be 1 and the total number of bits for fixed point would be 3.

Code:

#include "project.h"

#include <stdio.h>

//#define FLOATING\_VER

#define FIXED\_N (15)

#define FIXED\_ONE (1<<FIXED\_N)

#define FIXED\_ONE\_HALF (FIXED\_ONE >> 1)

#define SQR(x) ((x) \* (x))

// Define w0 as PI/100

// cos(w0) ~ 1.0

// sin(w0) ~ w0, small angle approximation

#define COS\_0 (FIXED\_ONE)

#define SIN\_0 ( (int) (FIXED\_ONE \* 3.1415926535897 / 100 + 0.5))

#define DCOS\_0 (1.0)

#define DSIN\_0 (2 \* 3.1415926535897 / 100)

#ifdef FLOATING\_VER

double dx\_real = 1.0;

double dx\_imag = 0.0;

double drsq;

double disq;

#else

int32\_t x\_real=FIXED\_ONE; // Initial value of x (complex number) is 1+j0, in fixed point

int32\_t x\_imag=0; // Initial value of x (complex number) is 1+j0, in fixed point

int32\_t rsq; // Holds square of real component

int32\_t isq; // Holds square of imag component

#endif

int32\_t sgn\_real; // +/-1, based on x\_real > 0 or < 0

int32\_t sgn\_imag; // +/-1, based on x\_imag > 0 or < 0

int main(void)

{

int i;

char msg\_str[1024];

CyGlobalIntEnable; /\* Enable global interrupts. \*/

UART\_Start();

for(i = 0; i < 400; ++i)

{

#ifdef FLOATING\_VER

dx\_real = dx\_real\*DCOS\_0 - dx\_imag\*DSIN\_0;

dx\_imag = dx\_real\*DSIN\_0 + dx\_imag\*DCOS\_0;

sgn\_real = dx\_real >= 0 ? 1 : -1; // signum of x\_real

sgn\_imag = dx\_imag >= 0 ? 1 : -1; // signum of x\_imag

drsq = SQR(dx\_real);

disq = SQR(dx\_imag);

if (drsq+disq > 1.0)

{

dx\_real -= 0.001 \* sgn\_real;

dx\_imag -= 0.001 \* sgn\_imag;

}

else

{

dx\_real += 0.001 \* sgn\_real;

dx\_imag += 0.001 \* sgn\_imag;

}

sprintf( msg\_str, "%5d\t%5d\n\r", (int) (dx\_real\*1000), (int) (dx\_imag\*1000) );

UART\_PutString(msg\_str);

#else

x\_real = x\_real\*COS\_0 - x\_imag\*SIN\_0; // QM.N \* QM.N -> Q2M.2N

x\_real = (x\_real + FIXED\_ONE\_HALF) >> FIXED\_N; // convert back to QM.N, with rounding

x\_imag = x\_real\*SIN\_0 + x\_imag\*COS\_0; // QM.N \* QM.N -> Q2M.2N

x\_imag = (x\_imag + FIXED\_ONE\_HALF) >> FIXED\_N; // convert back to QM.N, with rounding

sgn\_real = x\_real >= 0 ? 1 : -1; // signum of x\_real

sgn\_imag = x\_imag >= 0 ? 1 : -1; // signum of x\_imag

rsq = SQR(x\_real); // QM.N \* QM.N -> Q2M.2N

rsq = (rsq + FIXED\_ONE\_HALF) >> FIXED\_N; // convert back to QM.N, with rounding

isq = SQR(x\_imag); // QM.N \* QM.N -> Q2M.2N

isq = (isq + FIXED\_ONE\_HALF) >> FIXED\_N; // convert back to QM.N, with rounding

// The goal is for 'x' to represent a value on the unit circle, i.e. complex magnitude

// should be unity. Because of quantization error (which will even occur with double

// precision, need some way of avoiding collapsing to zero, or growing to infinity

//

// My quick and dirty solution is to push both real and imaginary part down toward

// zero a smidgen if the magnitude is greater than one, and away from zero otherwise

if (rsq+isq > FIXED\_ONE)

{

x\_real -= sgn\_real;

x\_imag -= sgn\_imag;

}

else

{

x\_real += sgn\_real;

x\_imag += sgn\_imag;

}

sprintf( msg\_str, "%5d\t%5d\n\r", (int) (x\_real), (int) (x\_imag) );

UART\_PutString(msg\_str);

#endif

}

}