Ricky Perez
Github root directory: https://github.com/RickyPerez79/da_submissions_2019

Date Submitted: **11/02/19**

Task 01:

Youtube Link: https://youtu.be/RrRJDH7ZEb4

Screenshots:

Memory Browser

Memory Browser

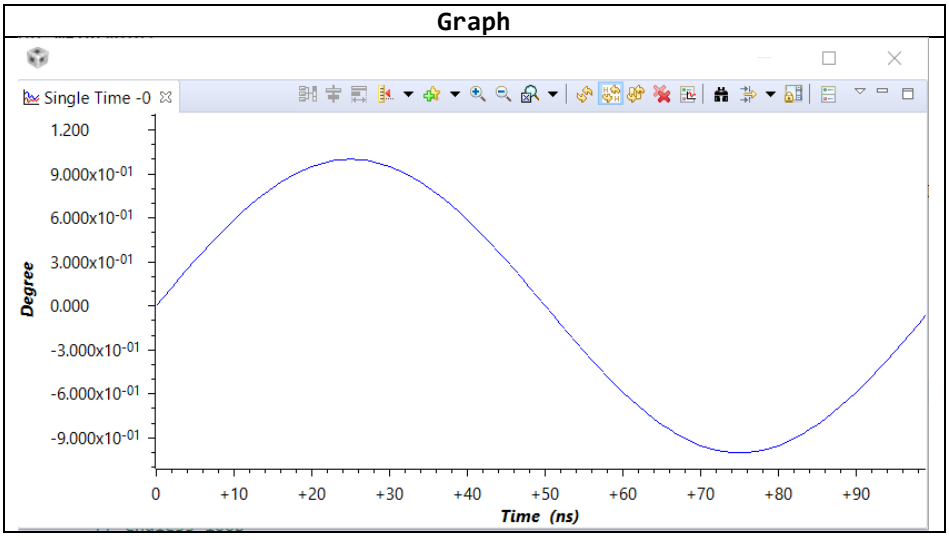
gSeriesData

0x20000200 - gSeriesData <Memory Rendering 1>

32-Bit Floating Point

0x20000200	gSeriesData		
0x20000200	0.0	0.0627905205	0.125333235
0x2000020C	0.187381327	0.248689905	0.309017003
0x20000218	0.368124574	0.425779313	0.481753707
0x20000224	0.535826862	0.587785244	0.637424052
0x20000230	0.684547186	0.72896868	0.770513296
0x2000023C	0.809017003	0.844327927	0.876306713
0x20000248	0.904827118	0.92977649	0.95105648
0x20000254	0.968583226	0.982287288	0.992114663
0x20000260	0.998026729	1.0	0.998026729
0x2000026C	0.992114663	0.982287228	0.968583107
0x20000278	0.95105648	0.92977643	0.904826999
0x20000284	0.876306593	0.844327807	0.809016883
0x20000290	0.770513117	0.728968501	0.684547126
0x2000029C	0.637423992	0.587785244	0.535826743
0x200002A8	0.481753588	0.425779164	0.368124396
0x200002B4	0.309016794	0.248689666	0.187381074
0x200002C0	0.125332966	0.0627904534	-8.74279067e-08
0x200002CC	-0.0627906322	-0.125333369	-0.187381476
0x200002D8	-0.248690069	-0.309017211	-0.368124783
0x200002E4	-0.425779551	-0.481753916	-0.535826862
0x200002F0	-0.587785304	-0.637424052	-0.684547186
0x200002FC	-0.72896874	-0.770513356	-0.809017122
0x20000308	-0.844328046	-0.876306832	-0.904827237
0x20000314	-0.929776609	-0.951056659	-0.968583226
0x20000320	-0.982287288	-0.992114782	-0.998026788
0x2000032C	-1.0	-0.998026729	-0.992114663
0x20000338	-0.982287288	-0.968583167	-0.95105648
0x20000344	-0.92977643	-0.904826999	-0.876306593
0x20000350	-0.844327807	-0.809016824	-0.770513058
0x2000035C	-0.728968382	-0.684546828	-0.637423694
0x20000368	-0.587784946	-0.535826445	-0.48175329
0x20000374	-0.425778866	-0.368124098	-0.309016496
0x20000380	-0.248689353	-0.187381208	-0.125333101
0x2000038C	-0.0627903715		

Graph



Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Modified Code:

```
// Ricky Perez
// CpE 403
// Lab 9
// Task 1
// Submit a comprehensive commented file of the original code.

#include <stdint.h>
#include <stdbool.h>
#include <math.h> // this program uses sinf() so we need this library
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" // support for Floating Point Unit
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"


// If the variable M_PI is not define when executing
// this will define it
#ifndef M_PI
#define M_PI        3.14159265358979323846  // value of Pi
#endif

#define SERIES_LENGTH 100 // depth of our data buffer
float gSeriesData[SERIES_LENGTH]; // an array of floats SERIES_LENGTH long

int32_t i32DataCount = 0;// counter

int main(void)
{
    float fRadians; // will be used to calculate sine

    ROM_FPULazyStackingEnable(); // turn on Lazy Stacking
    ROM_FPUEnable(); // turn on the FPU
    // set the system clock for 50MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);
    // A full sine wave cycle is 2*pi radians.
    // Divide (2*pi) by the depth of the array.
    fRadians = ((2 * M_PI) / SERIES_LENGTH);

    while(i32DataCount < SERIES_LENGTH)
     {
        // calculate the sine value for each of the 100 values
        // of the angle and place them in our data array
         gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);
         i32DataCount++; // increment counter by one
    }

    while(1)
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
        {
        // endless loop
        }

}
```

Task 02:

Youtube Link: https://youtu.be/y0GBMkazCCw

Modified Schematic (if applicable):

Memory Browser

Memory Browser

gSeriesData

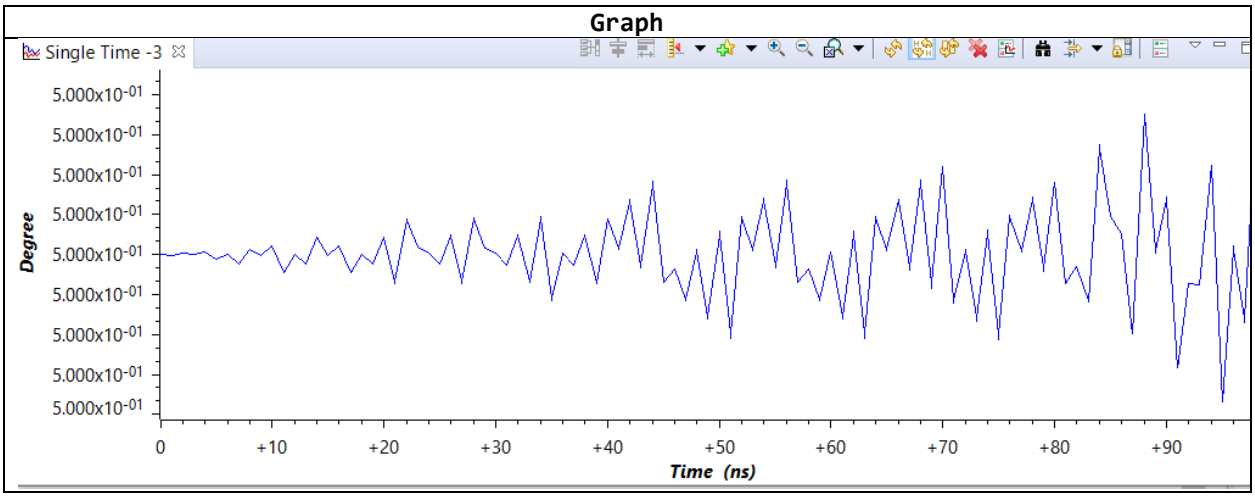
0x20000200 - gSeriesData <Memory Rendering 4>

32-Bit Floating Point

0x20000200	gSeriesData		
0x20000200	0.5	0.499999911	0.500000179
0x2000020C	0.49999997	0.500000358	0.499999315
0x20000218	0.50000006	0.499998659	0.500000715
0x20000224	0.49999994	0.500001311	0.499997377
0x20000230	0.50000006	0.499998629	0.500002682
0x2000023C	0.499999851	0.500001371	0.499997318
0x20000248	0.500000119	0.499998569	0.500002682
0x20000254	0.499996006	0.500005245	0.500001073
0x20000260	0.500000179	0.49999854	0.500002742
0x2000026C	0.499995977	0.500005305	0.500001013
0x20000278	0.500000238	0.49999845	0.500002801
0x20000284	0.499995917	0.500005364	0.499993354
0x20000290	0.500000298	0.49999845	0.500002861
0x2000029C	0.499995887	0.500005424	0.500000954
0x200002A8	0.500007987	0.499998391	0.50001055
0x200002B4	0.499995828	0.499997824	0.499993265
0x200002C0	0.500000358	0.499990702	0.500002921
0x200002CC	0.499988139	0.500005484	0.500000834
0x200002D8	0.500008047	0.499998301	0.50001061
0x200002E4	0.499995738	0.499997914	0.499993175
0x200002F0	0.500000477	0.499990612	0.50000304
0x200002FC	0.499988049	0.500005603	0.500000775
0x20000308	0.500008166	0.499998212	0.500010729
0x20000314	0.499995649	0.500013292	0.499993086
0x20000320	0.500000596	0.499990523	0.500003159
0x2000032C	0.49998796	0.500005722	0.500000656
0x20000338	0.500008225	0.499998093	0.500010788
0x20000344	0.499995559	0.499998122	0.499992996
0x20000350	0.500015914	0.500005662	0.500003219
0x2000035C	0.49998787	0.50002104	0.500000596
0x20000368	0.500008345	0.499982744	0.499995649
0x20000374	0.49999544	0.500013471	0.499977618
0x20000380	0.500000775	0.499990344	0.500018597
0x2000038C	0.50000304		

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

Ricky Perez
Github root directory: https://github.com/RickyPerez79/da_submissions_2019



Modified Code:

```
// Insert code here
// Ricky Perez
// CpE 403
// Lab 9
// Task 2
// Modify the code to implement the equation:
sin(2π*50t)+0.5*cos(2π*200t)
// to generate a frequency of 5 Hz.
// Display the equation for 1 sec.

#include <stdint.h>
#include <stdbool.h>
#include <math.h> // this program uses sinf() so we need this library
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/fpu.h" // support for Floating Point Unit
#include "driverlib/sysctl.h"
#include "driverlib/rom.h"


// If the variable M_PI is not define when executing
// this will define it
#ifndef M_PI
#define M_PI        3.14159265358979323846  // value of Pi
#endif

#define SERIES_LENGTH 100 // depth of our data buffer
float gSeriesData[SERIES_LENGTH]; // an array of floats SERIES_LENGTH long

int32_t i32DataCount = 0;// counter

int main(void)
{
    float fRadians; // will be used to calculate sine
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.

```
    float fRadians_2;
    ROM_FPULazyStackingEnable(); // turn on Lazy Stacking
    ROM_FPUEnable(); // turn on the FPU
    // set the system clock for 50MHz
    ROM_SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_XTAL_16MHZ |
SYSCTL_OSC_MAIN);
    // A full sine wave cycle is 2*pi radians.
    // Divide (2*pi) by the depth of the array.

    // 2π*50t; exclude "t" until loop
    fRadians = ((2 * M_PI* 50) / SERIES_LENGTH);
    // 2π*200; exclude "t" until loop
    fRadians_2 = ((2 * M_PI* 200) / SERIES_LENGTH);
    while(i32DataCount < SERIES_LENGTH)
     {
        // calculate the sine value for each of the 100 values
        // of the angle and place them in our data array
         //gSeriesData[i32DataCount] = sinf(fRadians * i32DataCount);

        //sin(2π*50t)+0.5*cos(2π*200t)
        gSeriesData[i32DataCount] = 1.0*sinf(fRadians * i32DataCount) +
(0.5*(cosf(fRadians_2 *i32DataCount)));

        i32DataCount++; // increment counter by one
    }

    while(1)
        {
        // endless loop
        }

}
```

Grading scheme: 30% Coding, 30% Documentation, 40% Execution/Video.