**FPU Lab**

**Links to videos:**

**Task 1:** http://screencast.com/t/Zl5JhQxIjY

**Task 2:** http://screencast.com/t/qtZQC20a

**Task 1: Adding comments to original code.**

//video: http://screencast.com/t/Zl5JhQxIjY

#include <stdint.h>

#include <stdbool.h>

#include <math.h>

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/fpu.h"

#include "driverlib/sysctl.h"

#include "driverlib/rom.h"

#ifndef M\_PI //define pi if not already defined

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 100 //length of array to hold sine values is 100

float gSeriesData[SERIES\_LENGTH]; //array of floats to hold sine values

int32\_t i32DataCount = 0; //variable to store counter value for loop

int main(void)

{

float fRadians; //variable to store radian values

FPULazyStackingEnable(); //enable lazy stacking

FPUEnable(); //enable floating point unit (FPU)

SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //set clock to 50 MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //2PI\*length = radians

while(i32DataCount < SERIES\_LENGTH) //loop according to size of floating point array

{

gSeriesData[i32DataCount] = sinf(fRadians \* i32DataCount); //populate array with sine values

i32DataCount++; //increment count

}

while(1) //infinite loop

{

}

}

**Task 2: Modify code to cosine function with 1000 sample points and determine the time for FPU operation:**

//video: http://screencast.com/t/qtZQC20a

#include <stdint.h>

#include <stdbool.h>

#include <math.h>

#include "inc/hw\_memmap.h"

#include "inc/hw\_types.h"

#include "driverlib/fpu.h"

#include "driverlib/sysctl.h"

#include "driverlib/rom.h"

#ifndef M\_PI //define pi if not already defined

#define M\_PI 3.14159265358979323846

#endif

#define SERIES\_LENGTH 1000 //length of array to hold cosine values is 1000

float gSeriesData[SERIES\_LENGTH]; //array of floats to hold sine values

int32\_t i32DataCount = 0; //variable to store counter value for loop

int main(void)

{

float fRadians; //variable to store radians

FPULazyStackingEnable(); //turn on lazy stacking

FPUEnable(); //enable the floating point unit (FPU)

SysCtlClockSet(SYSCTL\_SYSDIV\_4 | SYSCTL\_USE\_PLL | SYSCTL\_XTAL\_16MHZ | SYSCTL\_OSC\_MAIN); //set clock to 50 MHz

fRadians = ((2 \* M\_PI) / SERIES\_LENGTH); //2PI/length = radians

while(i32DataCount < SERIES\_LENGTH) //loop according to size of floating point array

{

gSeriesData[i32DataCount] = cosf(fRadians \* i32DataCount); //populate array with cosine values

i32DataCount++; //increment count

}

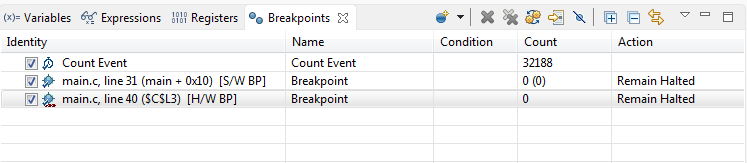
while(1) //infinite loop

{

}

}

**Time for FPU operation:** According to the following table, it took roughly 320 clock cycles to run each calculation. With a system clock of 50 MHz, that translates to a time of 6.4 micro seconds, and because there were 1000 samples, the total time would be 6.4 milliseconds.

****