Hardware/Software Codesign for Wireless Systems (E168b)

Harris

Lab 7: Correlation in C

The objective of this lab and the next two is to learn to develop nontrivial signal processing code in C. You have already done similar work in Matlab and Verilog, so these labs will give you a chance to see how the algorithms port to C. In principle, the C code could be implemented on the PowerPC. In the interest of time, we will just run it on a Windows computer where the development environment is much better.

In this lab, you will write a function that performs correlations in C and you will use it to acquire satellite 5 from data. The operations should be very similar to what you have done in Lab 2 in Verilog and Lab 3 in Matlab.

For this lab, it is recommended that you use Microsoft Visual Studio in the MicroP's lab. To do this, you will need to create a new project on the C drive of the local machine. You can find all the necessary files in the class directory in the lab 7 folder. After you have copied all the files to a folder on the local machine, run Visual Studio and select file>>new>>project. Select visual c++ and then console application in the right window. Enter the name of the project and place it in the same folder as the files from the course directory. Once you have created the project, go to file>>add existing file. Add all of the files from your directory. These files include a main function, so you will need to delete the main function that was automatically generated when you created the project. It should have the same name that you gave to the project. You will notice that there are a number of source files and header files that are necessary. You will only be working on correlate.cpp.

Begin by opening gpsc_main.cpp. Notice that in your version, only the acquisition function is called; the others will be added in later labs. Now open acquisition.cpp. Scroll down until you see the assignment of svlist, which is the array of satellites that the code will attempt to acquire. In this lab, you will only be concerned with satellite 5. Because we already know the location of the satellite, the next piece of code performs a check at the correct Doppler frequency and offset to determine if the correlate function is finding satellite 5. This will be very useful as you debug your code. The next part searches through the Doppler bins first in a coarse search and then in a more refined search. Both use the correlate.cpp function that you will be writing.

The Correlate Function

Open correlate.cpp. You will notice that there is already code written and that the inputs and outputs have been defined. The PRN sequences are generated in this file and a sine lookup table is generated. This lookup table takes the phase as an input and returns the result of the sine function to two bits of resolution. This is sufficient because the data itself has only two bits of resolution. Cosine can be easily generated from this lookup table; it is simply a constant offset in phase. The inputs and outputs of the main function are described in the comments at the beginning of the function. The main difficulty of this lab will be to find the correct phase and to properly index the data in the correlations. It will be very useful to think in terms of the binary representations of the values, the correlation can be greatly simplified with bit operations. Your code will perform the correlation storing the results in sumi and sumq and it will also update the code phase and carrier phase as the correlation is performed. You may also want to insert a breakpoint at the end of the main function so you can see what is printed to the screen.

What to Turn in

1) Your C code for the correlate function
2) The results of the initial test for satellite 5
3) The final Doppler frequency and acquisition metric found with your function