GPS RECEIVER IMPLEMENTATION USING SIMULINK

C.Abhishek[1], A.Charitha[2], Dasari Goutham[3]

[1]Student, SCSVMV University, Kanchipuram
[2]Student, kl university, Vijayawada
[3]Student, SVEC college, Tirupati

Abstract: The Global Positioning System (GPS) is a space-based satellite navigation system that provides location and time information in all weather conditions, anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. GPS (Global Positioning System) is the one of the most pre-dominantly used in most parts of the world In the past, the implementation of a complete GPS receiver was divided into two parts. The first part is implemented on ASIC or FPGA. This part includes the acquisition and tracking phases, where the algorithms of the both are written by the HDL programming language. The second part is the navigation solution part that is implemented on DSP by writing its algorithm by C/C++, FORTRAN, or assembly. This means that we have to deal with three environments to implement a complete GPS receiver. The three environments are the Simulation, FPGA, and DSP environments. Moreover, using a text programming languages in writing such long and complicated algorithms makes the process difficult to debug, modify. Simulink was used in the implementation of such receiver; there by introducing a new look for the technology that can be implemented through a graphical user interface environment. This makes every part in the receiver architecture very clear and easier to understand, follow, modify and debug. This can be considered a step added on the route of an open source GPS receiver. Using the same environment in both the simulation and implementation stages makes the designer's mind dedicated most of the time in developing and enhancing the algorithm through rapid prototyping and experimentation and less time on the coding. In general, this article can be considered as introducing a new look for designing, simulating, and implementing the typical GPS receiver using a graphical programming language, which is SIMULINK.

Keywords: GPS receiver, MATLAB, SoftwareDefinedReceiver, Simulink, acquisition.

I. INTRODUCTION

Any GPS receiver that deals with the Standard Positioning Service (SPS) transmitted on the line carrier of 1.57542GHz using the Coarse Acquisition (C/A) code has four phases to extract the position and time from the received signal. These phases are: the front-end, the acquisition phase, the tracking phase, and the navigation solution phase. The implementation of a complete 8-channels single frequency GPS receiver using the C/A code on the line carrier using Matlab as the coding language. This represents the receiver algorithm and a record for a real GPS signal collected from a specially designed ASIC-based front- end.

In general, the algorithm of a complete GPS receiver is considered complicated and long, because it contains different stages that deals with the processing of RF, IF, and baseband signals. Previously, to write either a part or a complete GPS algorithm you must be professional in C/C11 programming, assembly, or Matlab. These are the programming languages that users used in the simulation of the GPS receivers. To write a complete algorithm for a GPS receiver; a teamwork is needed. Every person writes the algorithm of a certain part and delivers the output to the next person to write the next part of the algorithm. Using a GUI environment instead of the hand written programming languages was a far idea due to the algorithm's complexity and length. A Single-Frequency Approach are converted to only 5 Simulink models, after modifying the algorithm of some parts to conform with the GUI environment and make the simulation time faster. This article adds a step on the route toward a clear, easy, and transparent simulation of GPS receivers implemented with the SDR technology. The implementation of a complete 8-channel GPS receiver is accomplished via a graphical environment, which is Simulink. In general, the graphical environment makes the relation between the system modeling, simulation, and implementation clearer and easier to debug, because we can put test points anywhere and obtain immediate results. Also, using a graphical programming language eliminates the need to a team work and enables only one person to finish the simulation of the complete system in a relatively short time. In general, the success of using a graphical programming language such as SIMULINK in simulating a complicated system such as the GPS receiver will open the route for other complicated systems to be simulated by the same way. This will facilitate the education, the modification and the debugging of many of the complicated electronic systems.

II. BUILDING the DIFFERENT STAGES of GPS RECEVIER

The Simulink models of acquisition, tracking, pseudo range, and position solution are tested and developed by a real GPS signal that was collected from the SE4110 ASIC-based front end. Its functional block diagram is shown in reference. The same signal is used in testing and developing the Matlab algorithms.

.

A. Representing the Data Collected from the Front- End in the Different Domains

The digital samples that were collected from the frontend can be represented in the different domains using the different types of scopes available in Simulink. Fig. 1 shows the Simulink model for data representation. In this figure the first branch gives the time domain representation of 1,000 samples . The signal is a noise like and this is the nature of the CDMA signal. The second branch gives a histogram of 1,048,576 samples that is shown in Fig. 3. It is clear that the 4- bit samples are translated to 16 levels. The third branch gives the frequency domain representation of 1,048,576 samples that is shown in Fig. 2. It is clear that the symmetry is around the IF frequency

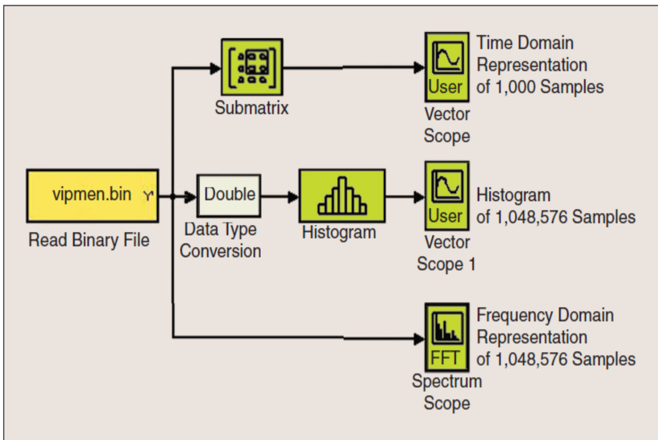
Submatrix
Time Domain Representation of 1,000 Samples
User
Vector Scope
vipmen.bin
Read Binary File
Double
Data Type Conversion
Histogram
Histogram of 1,048,576 Samples
User
Vector Scope 1
Frequency Domain Representation of 1,048,576 Samples
FFT
Spectrum Scope

Figure 1. Simulink model to represent the collected data in the different domains.

Figure 1. Simulink model to represent the collected data in the different domains

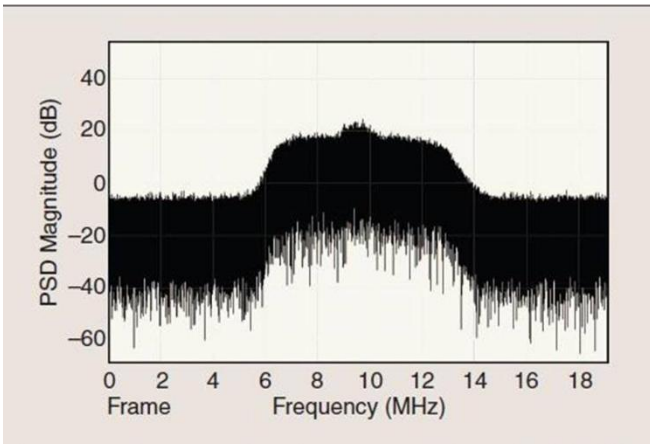

Figure 2 Frequency domain representation of 1,048,576 samples.

B. GPS Receiver

The operation of a single channel GPS receiver is based on the structure shown in Fig. 3. In acquisition we determine one of the visible satellites and determine both the carrier frequency and code phase roughly. In tracking we track any changes to these values to ensure correct data decoding. After that both the pseudo range and position are calculated. Each block in Fig. 3 is translated into a Simulink model, as shown in Fig. 4. In Fig. 4 there are two additional models; one provides the demodulation Carrier and the other provides the dispreading code. Both the demodulation and dispreading models must be simulated first to feed lookup tables existed in the acquisition model. These tables contain all the possibilities of the demodulating carrier and dispreading codes according to a specified sampling frequency that is a modifiable value.

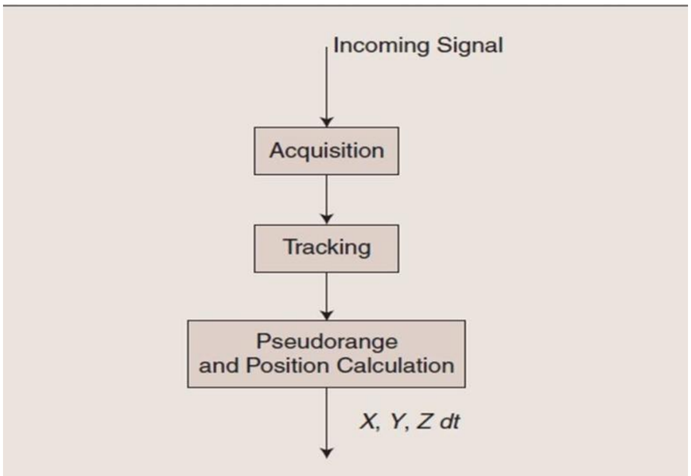

Figure 3 Structure of a single channel GPS receiver.

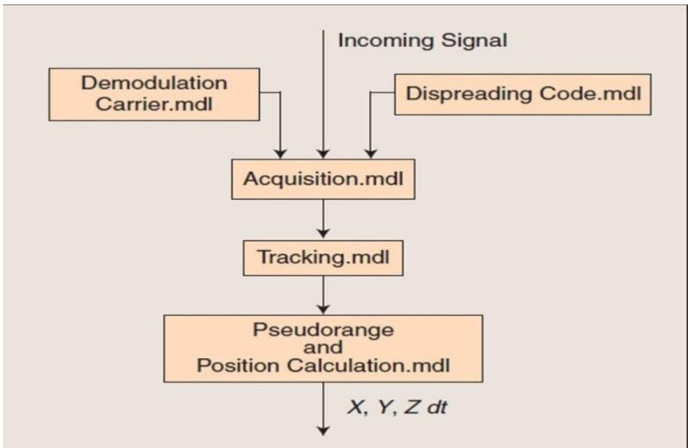

Figure 4 SIMULINK implementation of a single channel GPS receiver.

III.TRACKING &NAVIGATION

The main purpose of tracking is to refine the acquisition results, track any changes that occur to these results with time, and demodulate and dispread the incoming signal to obtain the
50 bits/s navigation data. The data size that is being processed in each tracking cycle is a complete C/A code plus or minus the delay or lag that is being calculated during tracking. Tracking starts if the number of the detected PRN codes at the acquisition phase ≥ 4.In general, tracking is implemented as a carrier tracking loop that works in consistency with a code tracking loop. The carrier and code tracking loops that is
Used usually in GPS receivers are the Costas Phase Lock Loop (PLL) and the Delay Lock Loop (DLL). Fig. 5 shows the block diagram of both Costas PLL and the DLL. Usually the two loops are combined in one loop to increase the control on the generated carrier frequency and the code phase and to reduce the number of multipliers. This helps in reducing the tracking time.

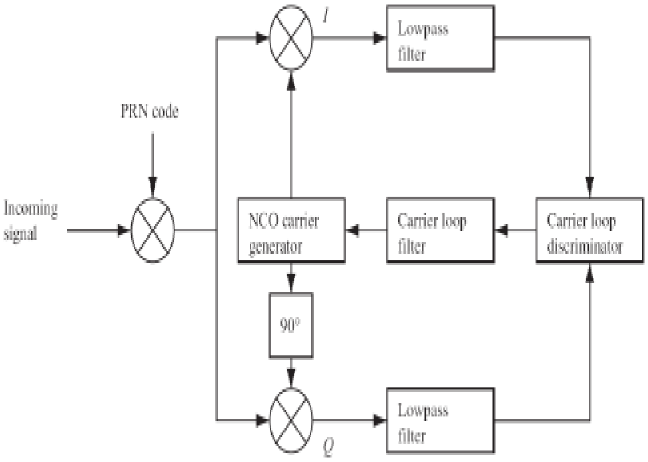

Fig. 5: Costas phase locked loop

In this phase the 50 Hz navigation bits are decoded according to ICD-GPS-200 (1991) to obtain the pseudo range, the receiver position, and the receiver clock offset. The navigation solution phase is simulated according to the block diagram.

The first step in the navigation solution phase is the Bit Synchronization. In this step we find the time where bit transitions occur and then each 20ms are replaced by a single value that represent a navigation data bit. The second step is determining the beginning of each sub frame that is consisting of an 8-bit long preamble and that is performed by correlating the tracking output with a locally generated preamble.

The third step is to determine the transmission time of the first detected sub frame by the aid of decoded TOW word. The TOW corresponds to the transmission time of the next sub frame. To obtain the transmission time of the first sub frame we multiply the TOW by 6 and subtract 6s from the result. All the navigation bits are ready now to be decoded. The ephemeris parameters are decoded. Pseudo range calculations are accomplished on two steps: the first step is to find initial pseudo ranges. The second step is to keep track of the initially calculated pseudo ranges. The initial pseudo ranges are used in calculating the receiver position (X Y Z) and the receiver clock offset (dt). The Least- Squares method is usually used in calculating the receiver position for ≥ 4 visible satellites.
The Simulink models of acquisition, tracking, and navigation solution phases were combined in one model that represents the simulation stage of a complete GPS receiver. Constructing the receiver in such a way makes its architecture very clear and very easy to debug, modify, and separate any part of it and obtain immediate results.

VI. CONCLUSIONS

In this article we introduced the simulation and implementation of a complete GPS receiver on through a graphical programming language, which is Simulink. Also we introduced the idea of modifying both the IF and Fs in the Simulink GPS receiver and the corresponding results can be easily analyzed. Using a graphical environment in both the simulation and implementation stages makes the designer mind dedicated most of the time in developing and enhancing the algorithm and less time on the coding. Moreover using a graphical programming language makes the algorithm very clear and can be easily modified and debugged.

REFERENCES:

[1] H. A. David, Order Statistics, 2nd ed, New York: Wiley, 1981.

[2] S. Diggavi, M. Grossglauser, and D. N. C. Tse, “Even one-dimensional mobility increases ad hoc wireless capacity,” in Proc. IEEE Int. Symp. Information Theory (ISIT), Lausanne, Switzerland, June 2002.

[3] W. Feller, An Introduction to Probability Theory and Its Applications, 2nd ed, New York: Wiley, 1971, vol. II.

[4] A. Goldsmith and P. Varaiya, “Capacity of fading channel with channel side information,” IEEE Trans. Inform. Theory, vol. 43, pp. 1986–1992, Nov. 1997.

[5] P. Gupta and P. R. Kumar, “The capacity of wireless networks,” IEEE Trans. Inform. Theory, vol. 46.