REAL-TIME ACQUISITION AND TRACKING FOR GPS RECEIVERS

Abdulqadir Alaqeeli, Janusz Starzyk, Frank van Graas

School of Electrical Engineering and Computer Science
Ohio University
Athens, OH 45701

ABSTRACT

Current GPS receivers spend much time in base-band processing performing acquisition and tracking. This is due to the large number of required operations in the software-based signal processing. This paper presents a novel signal acquisition and tracking method that reduces the number of operations, simplifies hardware implementation and decreases the acquisition time. The implementation of this method in an FPGA provides very fast processing of incoming GPS samples that satisfies real-time requirements.

1. INTRODUCTION

The Global Positioning System (GPS) has been widely used in civilian and military positioning, velocity, and timing applications. GPS transmits its standard positioning service (SPS) on the L1 carrier frequency of 1575.42 MHz. Each GPS satellite (or transmitter) has a unique spreading gold code (C/A code) that is orthogonal to all the other satellites' codes. GPS receivers, on the other hand, must search for these C/A codes to know which satellites are available to the user. For each code, a receiver must perform a 2-D search for carrier frequency offset and code shift, or in other words acquire the C/A code. Then it should track (or lock in) the signal. Acquisition is the most time consuming operation in the GPS receiver [1-2]. A fast search algorithm was presented by Van Nee in [1]. This method searches all possible code shifts in one step using the well-known FFT-based correlator. It performs the correlation using frequency domain multiplication [1].

When a SPS GPS signal is received, it goes through many steps of filtering and amplification. Then the signal is digitized using at least a 5MHz sampling rate which obeys Nyquist's law for the C/A code, which has a bandwidth of 2.046 MHz [3]. This leaves the GPS signal as 5,000 samples of C/A code for each 1-ms period of the code, which could be downconverted to an intermediate frequency of 1.25 MHz [4]. From this point, the GPS receiver starts its digital signal processing (DSP) algorithms to find the C/A codes. Different search algorithms were presented in [1][3-5]. Assuming that the receiver knows the C/A code and the exact intermediate frequency, it needs to do a carrier wipe-off before performing the FFT-based correlation function with the locally-generated C/A code.

The FFT-based correlator is chosen for the implementation of the acquisition process as part of the block processing method [6]. This way, the gained advancements in block processing could be used. Building power-of-two-based FFTs is achieved by using a uniform butterfly structure. However, 5,000-point FFTs are required to implement the C/A code correlator. The size of this FFT is not a power-of-two, so it will be very difficult to implement since it requires a mixed-radix algorithm that also includes non–power-of-two small FFTs [7]. The use of smaller FFTs is needed to map the acquisition process in the FPGA. A novel method for solving this problem was first presented by Starzyk and Zhen in [8]. This method is called acquisition of C/A code using averaging correlators, and it uses FFTs with a similar size as the C/A code (1,023 instead of 5,000-point FFTs).

2. AVERAGING CORRELATOR

The averaging method averages the incoming 5,000 samples to become 1,023 averaged samples. If the starting point of the averaging operation is chosen in the right place, the 1,023 new samples may represent the original chips of the C/A code. In other words, one can say that a full C/A code is presented by 5,000 samples and that its chips are represented by either 4 or 5 samples each. Since the C/A code is circular, therefore, the right averaging starting point is one of 5 successive samples. This will generate 5 of the 1,023 averaged-samples code. One of the generated sequences is considered a good approximation of the original C/A code (or the best recovery) [8]. This best recovered averaged sequence contains the strongest peak among the other four and estimates the code phase in chips (1/1023 ms). The resultant five correlation functions are a good approximation of the 5,000-point correlation function and thus can be used with a triangle fitting to refine the code phase.

This method did not reduce the peak to the second peak value[8]. As a result, detection probability is not affected by replacing the 5,000-point FFT-based method with the averaging method. The acquisition time is reduced by using this method, because calculating five 1,023-point FFTs and IFFTs requires less time (in software and in hardware) than the 5,000-point FFTs and IFFTs. However, implementing 1,023-point FFT (or IFFT) is not an easy task since it is not a power of two. A zero-padding-based solution was introduced by Zhen in [9]. The signal energy loss on average was acceptable. However, the computation time or the necessary hardware resources are increased to a certain level that makes it useless, especially when real-time acquisition is required. One possible solution to the problem of size mismatching of the code and the available Xilinx's 1024-point FFT core is presented in the next section. The new approach will show a fast acquisition method that can be implemented much easier than the 5,000-point FFT-based method.

0-7803-7761-3/03/$17.00 ©2003 IEEE

3. MODIFIED C/A CODE

The size incompatibility of C/A code and the available FFT core can be solved by changing the down-sampling rate from 1,023 to 1,024. So, the 5,000 samples will be down sampled (or averaged) to 1,024 points. A similar procedure will be done to the local code. Therefore, the local code will be up-sampled to 5,000 and then down sampled to 1,024 points. Therefore, the averaging correlator here will use 1,024 averaged samples and 1,024-point modified C/A code. The resultant code has changed from a binary code to a multilevel code which contains values as ±1.0, ±0.8, ±0.6, ±0.5, ±0.4, ±0.2, and 0.0. However, this modified C/A code is still considered a unique code related to the selected C/A code. It cannot be generated from a different C/A code.

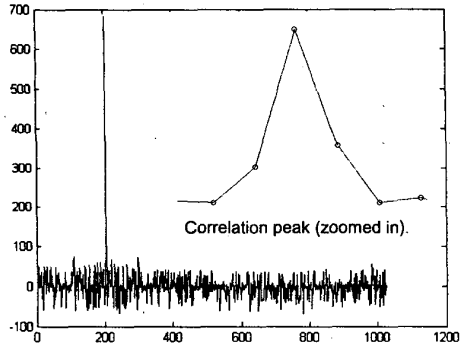

Figure 1. Winner autocorrelation function of modified C/A code

As stated above, the averaging correlator needs to be applied five times with five successive starting samples. When the five correlation functions for this averaged C/A code are generated, the correlation function that contains the strongest peak keeps the important information necessary for acquisition. Fig-1 shows the winner autocorrelation function of the modified C/A code. The peak is not based on equilateral triangle, as what it should be in a normal C/A code, but it is good enough to roughly estimate the code phase for acquisition. One may notice that each point in this correlation function is one out of 1,024 modified chips. Each modified chip is a 1/1024 ms while the original chip width is a 1/1023 ms. To estimate the code phase, one needs to consider this change in chip width and should correct the value of the code phase. For example, if the peak is at location 120 out of 1,024, then the estimated code phase equals to (120/1024)*1023 original chips, or in other words, the code phase is (120/1024)*1ms.

The averaged-code correlator needs five times 2*1024*10 plus 5,120 multiplications and five times 2*1024*10 additions. Therefore, the total required number of operations equals to 107,520 multiplications and 102,400 additions. Also, the averaging computation requires about 25,000 additions and 5,120 divisions. The division operations will not be counted because they can be avoided. Therefore, the total number of operations is 107,520 multiplications and 127,400 additions. This does not reduce the computation time much compared to the 5,000-point FFT-based correlator, but the implementation of 1,024-point FFT is much simpler than the 5,000-point FFT. This will lead to significant simplification in the hardware implementation.

In order to use this method, the effects on signal-to-noise ratio and the other characteristics should be studied. The next section presents the characteristics of the modified-code averaging correlation method in terms of signal-to-noise ratio (SNR) loss, code phase accuracy and carrier phase accuracy.

4. CHARACTERISTICS OF MODIFIED CODE AVERAGING CORRELATION

In the previous section, the modified code averaging correlation method was described. However, in order to use it for the acquisition process and the tracking loops, it is necessary to check its effect on signal power and on calculation accuracy of code and carrier phases. For this reason, a 200ms of GPS data is used to test the acquisition using the averaged-code averaging method. The peak-to-peak ratio was chosen as a measure of signal strength. Matlab simulation showed that the average loss in peak-to-peak ratio is about 12.7%, as shown in Fig-2. This loss is about 0.5dB, and is acceptable in most of the cases, except in the case of weak signal acquisition.

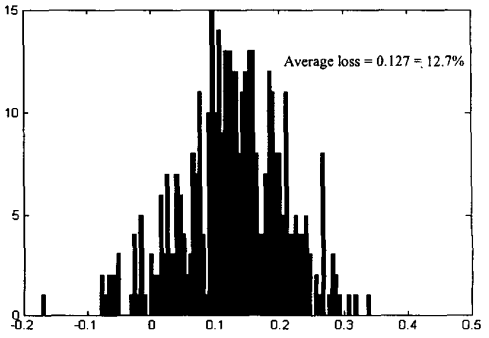

Figure 2. peak-to-peak loss using modified C/A code for 400ms

For the acquisition process, code phase estimation accuracy needs to be within ± ½ chip. The averaging method was able to produce the right code phase with accuracy of 100ns (± 0.1 of a chip) as seen in Fig-3a. Therefore, the averaging method is acceptable for acquisition. However, tracking requires that the block-processing produce refined code phase with high accuracy. For this reason, the triangle fitting approach was applied to the peaks of the averaging method. This approach was able to refine the code phase estimation and the average error in code phase estimation using triangle fitting was 42.7ns. A modified triangle fitting with acquisition history feedback was developed and showed an improvement over triangle fitting (with average error of 10ns) in the code phase calculation accuracy. However, none of these results has sufficient accuracy to replace the real-time tracking. Moreover, the carrier phase estimation based on the block processing using averaging method was tested. The accumulated carrier phase error was 0.03 radians when computed from Fig-3b. The accumulated carrier phase error was considered acceptable for C/A code tracking. However, the limiting factor is the code phase error, which was found not acceptable. Therefore, the modified-code averaging correlator

method was selected for the implementation of the acquisition process only.

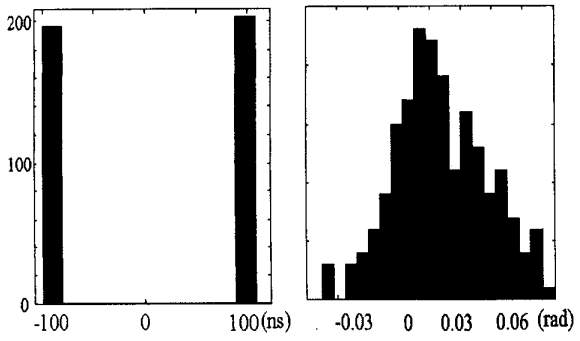

Figure 3. (a) code phase error (b) carrier phase error

5. IMPLEMENTATION OF BASE-BAND PROCESS OF GPS RECEIVER

In the previous section, we showed that the computational effort of the averaging correlator with the modified-code approach is small compared to regular 5,000-point FFT-based correlators. However, the implementation of 5 1024-point FFTs is much simpler than designing a single 5000-point FFT. Therefore, the averaging correlator is our choice for the implementation of the acquisition process.

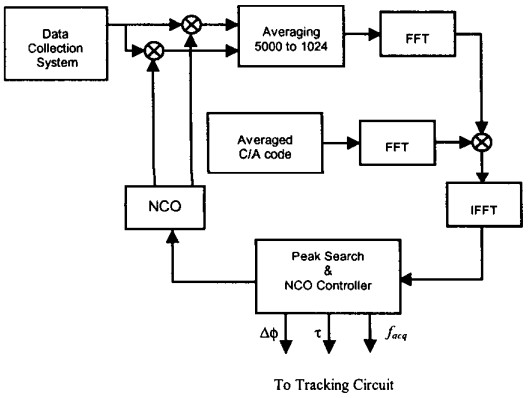

Figure 4. Acquisition using averaging-correlator

A block diagram of the averaging-correlator-based acquisition is shown in Fig-4. First, the samples are multiplied by in-phase and quad-phase components of the carrier signal. The I and Q channels are each averaged to 1,024 points. Then they are converted to frequency domain using 1,024-point complex FFT and multiplied by the conjugate of the FFT of the averaged-local-code. A 1,024-point complex IFFT is used then to return back to the time domain. A peak searcher inspects the 1,024 outputs of the IFFT and stores the location of the peak and its value. This process is repeated four times more, each with a different starting point, as described in the previous section. After checking all five loops, the peak searcher compares the peak value to a threshold to decide if the peak (or a GPS code) is detected. If the searcher does not detect a peak, then a new search cell with different frequency bin is inspected using the above- described process. This is repeated until a true peak is found or until all frequency search bins are completed. When the GPS signal is acquired, the frequency is selected and the search is conducted only in the code-phase dimension.

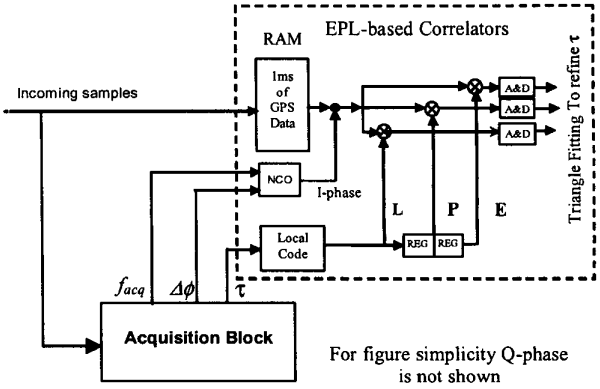

Figure 5. Block diagram of the implemented base-band

Every 1ms the acquisition process will produce estimate of the code phase, carrier phase, frequency, and peak value. These values are accurate enough to guide three 5000-point serial correlators for the early-prompt-late (EPL) implementation of delay-locked-loop (DLL) [2-3]. Fig-5 shows the EPL-based correlators and how they communicate in the whole system. They provide the three correlation points around the true peak and then used with triangle fitting technique to refine code and carrier phases. This method enhances the code phase accuracy to few nano-seconds [6]. One of the reasons behind using the serial correlators is that they occupy small silicon area or fit easily in an FPGA. These correlators do not require closed loops, like the ones in the standard tracking loops, because the acquisition component provides all the necessary information to keep the serial correlators in-lock with the signal.

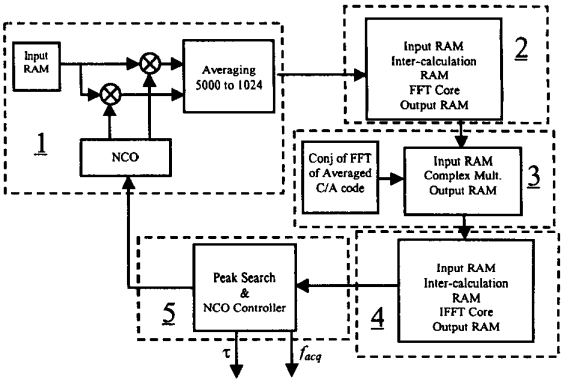

Figure 6. Partitions of Acquisition Process

The FPGA platform we used has a Virtex FPGA that provides 800k logic gates. This FPGA cannot implement the whole base-band system. Therefore, the system was partitioned into smaller parts as shown in Fig-6. Each block was mapped to the FPGA and

was tested separately. These blocks were also tested in sequence with real GPS data and acquisition and tracking were achieved. Fig-7 shows the hardware simulation for the correlation functions of the averaging-based acquisition. Five peaks are shown and they are identical to Matlab simulation using fixed-point operations.

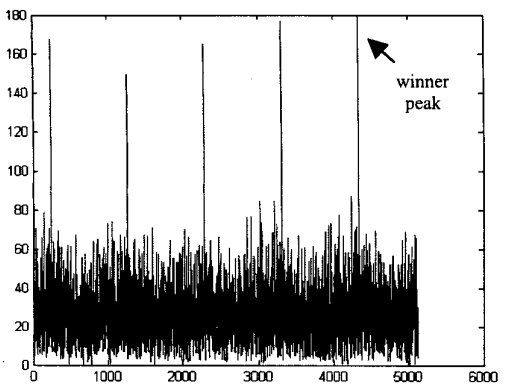

Figure 7. Averaging Correlation Hardware Results

The required FPGA resources (or implementation area) for each part of the design are shown in Table-1. In order to estimate the required FPGA resources for the whole system, we cannot simply add of the numbers of the logic slices used and the used block RAMs of all the blocks. This is because each block contains data collection, intermediate calculation handling, and result storage that can be reduced for the complete system implementation to avoid redundancy. Therefore, estimation of the required FPGA resources of the whole system which implements acquisition and tracking is about 8,000 programmable logic slices and 114 Block RAMs. With the availability of such resources in one FPGA such as the VirtexE1600, the implementation of such system is possible. The whole design may take about 50 % of the available slices and 80 % of the available Block RAMs. This leaves more room for routing that can directly affect the maximum clock speed. Routing in FPGA is one of the main problems since it is not predictable. However, based on the already implemented blocks and the above estimations, a system clock of 45MHz may be used with no timing problems. The whole system requires about 45,000 clock cycles. If the clock speed is equal to or more than 45MHz, the averaging correlation of 1-ms GPS data is computed in 1-ms or less. Assuming that the carrier frequency is known, the acquisition is conducted in the code phase dimension and therefore processing blocks of 1-ms GPS data can be achieved in real time.

Table 1. Implementation Cost (Virtex Resources)

	Acquisition						Serial Correlators (zooming)
	Part.#	1	2	3	4	5	
Configurable Logic Slices (max. 9408)		704	2288	649	2288	862	697
Block RAMs (max. 28)		25	24	18	24	8	17

6. Summary

This paper presented the current challenges for designing a real-time base-band block processor for a GPS receiver using FPGA Technology. The two main processes, acquisition and tracking, were described. Speeding up the GPS block processing was the goal of this paper. A fast and simple-to-implement acquisition algorithm based on averaging-correlation method was presented. This gave acceptable accuracy to guide serial correlators to zoom in for better accuracy similar to block processing technique. The algorithm was implemented using acquisition-driven-correlators architecture for block processing and was mapped to an FPGA. The implementation architecture satisfies real-time processing and tracking accuracy that makes it a useful core in a GPS receiver for real-time applications.

7. REFERENCES

[1] Van Nee, D. and Coenen, A., "New fast GPS code-acquisition technique using FFT," *Electronics Letters*, vol. 27, no. 2 pp. 158-160, January 17, 1991.

[2] M. Braasch and A. J. Van Dierendonck, "GPS Receiver Architectures and Measurements," *Proc. of The IEEE,* vol. 87, no. 1, pp. 48-64, January 1999.

[3] Akos, D., "A Software Radio Approach to Global Navigation Satellite System Receiver Design," *Ph.D. Dissertation*, Ohio University, August 1997.

[4] Tsui, J., *Fundamentals of Global Positioning System Receivers: A Software Approach*, John Wiley & Sons, Inc., 2000.

[5] D. Lin and J. Tsui, "Comparison of Acquisition Methods for Software GPS Receiver," *ION GPS-2000*, pp. 2385-2390, Salt Lake City, UT, September 2000.

[6] G. Feng, and F. van Graas, "GPS Receiver Block Processing", *Proc. ION GPS-99*, pp. 307-316, Nashville, September 1999.

[7] Sanjeev, G., "Feasibility Study for the Implementation of Global Positioning System Block Processing Techniques In Field Programmable Gate Arrays," *MS. Thesis*, Ohio University, June 2000.

[8] J. Starzyk and Z. Zhu, "Averaging Correlation for C/A code Acquisition and tracking in Frequency Domain," *MWSCS,* Fairborn, Ohio August 2001.

[9] Zhu Z., " Averaging Correlation for Weak GPS Signal Processing," *MS. Thesis*, Ohio University, April 2002.