Optimization of GPS L1 Acquisition using Radix-4 FFT

Swarna Ravindra Babu#1, Panneer Selvam#2, Gottapu Sasibhushana Rao*3, Jinling Wang&4

#Advanced Technology Labs, Wipro Technologies
105 Anna Salai, Guinday, Chennai-32, India
1,2swarna.babu@wipro.com
3College of Engineering, Andhra University, Vizag, India
4University of New South Wales, Australia

Abstract— **The implementation of an acquisition loop for software GPS receiver is still a challenge because of the large number of arithmetic operations involved in an FFT algorithm. Oversampling of the composite signals further compounds this problem. A Radix-4 FFT algorithm was proposed to improve the performance and the results were compared with Radix -2 FFT. To validate the algorithms a software GPS baseband processor was developed in Simulink and tested with real satellite data set. The analysis shows that the Radix-4 FFT is about 23% more efficient than the Radix-2 algorithm.**

Keywords— **GPS Acquisition, FFT, Radix -4, Simulink**

I. INTRODUCTION

Modern applications have brought a paradigm shift in the GPS receiver design. With these receivers increasingly being integrated with the mobile devices, the speed, memory and the processing power of the embedded platforms become far more critical in the overall design. The design of the algorithms does not have that luxury it used to have when the systems were built for specific applications. The convergence of Mobile, GPS, Audio and Video devices in a single embedded platform is bringing a lateral thinking in the fundamental design procedure itself. Moreover, the evolution of the SDR (Software Defined Radio) technology has further pushed the frontiers of the design with the critical signal processing algorithms running on a 'reconfigurable and flexible' FPGA's and the application related algorithms running on embedded/DSP platforms.

The GPS receiver design is also undergoing tremendous changes due to the ongoing developments in the satellite navigation market. Several countries have begun launching their own satellite based navigation systems to preserve their 'autonomy' and also address their safety critical applications. It is envisioned that by the end of this decade there should be about 120 satellites serving only the navigation, position and timing markets. Therefore, the 'GPS receiver' is slowly replaced by 'GNSS receiver' which accommodates signals from multiple constellations. The GNSS (Global Navigation Satellite System) receiver designer faces a plethora of challenges – multiple RF bands, substantial increase in the correlator channels, time and position synchronization across multiple constellations in the navigation domain, limited processing power, size and power consumption limitations. However, as the primary objective of the 'futuristic' receivers is to increase the availability of satellites at the antenna the design procedures are slowly evolving to accommodate as many constellations as possible.

The GPS receiver should determine the precise estimates of Doppler frequency and PRN (Pseudo Random Noise) code phase to decode the navigation data which are then used for user position computation. Two loops work concurrently to provide these estimates – Acquisition and Tracking. While the acquisition loops provide coarse estimates, the tracking loops refine them to finer values. Of these two loops the Acquisition is the most challenging one as the amount of uncertainties in determining the Doppler and PRN code is large and also the search process should complete within a short duration. A stationary receiver will have about +/-6KHz Doppler uncertainty and +/-0.5 chips code phase offset. The acquisition loops will have to search these entire regions to determine the presence of absence of the signals. Conventionally, time domain approach using FLL (Frequency Locked Loop) and DLL (Delay Locked Loop) was used for the acquisition [3][10]; however with the advent of software receivers the frequency domain approach based on FFT has become more common [2][5]. Literatures have also shown that FFT algorithms acquire the signals faster [6][7][8][9].

Although there are several variants of FFT implementation, radix-2, radix-4, radix-8 and split-radix (2/4, 2/8), this paper proposes the radix-4 approach from both the implementation complexity and efficiency perspectives. However, it should be noted that this concept can be easily extended to other radix algorithms. The paper is organized as follows: Section II presents the GPS Baseband Processor Model explaining the mathematical model, time and frequency domain approaches. Section III provides details on the DFT, FFT and Radix-4 algorithms. Section IV demonstrates the operation of the Radix-4 algorithm in the acquisition loops using Simulink models.

II. GPS BASEBAND PROCESSOR MODEL

A. *GPS Baseband Mathematical Model*

Let us consider the mathematical model of the digitized IF signal to be (due to its orthogonality with the C/A (Coarse Acquisition) code and its classified properties the P (Precise) code can be conveniently removed from the analysis)

$$y_{IF}(t) = AC(t)D(t)\cos(w_{IF}(t-\tau)+\phi)+\eta \qquad (1)$$

where A $=\sqrt{2P}$, P is the power of the signal; $C(t)$ is C/A code sequence; $D(t)$ is Navigation data; ϕ is the carrier phase; τ is propagation delay between the satellite and receiver; w_{IF} is the intermediate frequency; η is the Gaussian noise. The propagation delay is expanded to be

$$\tau = \frac{|X_s - X_u|}{c} \qquad (2)$$

where X_S is the GPS Satellite position, X_u is the user position, and *c* is the velocity of light. After multiplying the $y_{IF}(t)$ with the local signals and integrating them, the I (In-Phase) and Q (Quadrature) signals are generated which are given by

$$I = \int_{kT}^{(k+1)T} \sin(\hat{w}t + \hat{\phi})\hat{C}(t)[y_{IF}(t)]\, dt \qquad (3)$$

$$Q = \int_{kT}^{(k+1)T} \cos(\hat{w}t + \hat{\phi})\hat{C}(t)[y_{IF}(t)]\, dt \qquad (4)$$

where $\hat{w}$, $\hat{\phi}$ and $\hat{C}$ are the local frequency, phase and code estimates respectively. When the signals are detected and tracked successfully the *I* channel provides the 50Hz navigation data whereas the *Q* channel will contain only noise.

B. *Time domain algorithm*

The GPS signals are received at the L1 frequency of 1575.42MHz and converted into digitized IF (Intermediate Frequency) signals by the RF-down converter and ADC. The digitized signals are then sent to the baseband processor, constituting the acquisition and tracking loops, for navigation data demodulation. A typical baseband architecture working in time domain is shown in the Fig. 1. The carrier NCO (Numerically Controlled Oscillator) generates orthogonal signals, sin and cos, and the code NCO generates three signals, Early, Prompt and Late, each separated by 0.5 chips. Both the carrier and the code signals are multiplied with the incoming satellite signal to generate I and Q signals. An I & D (Integrate and Dump) block integrates these signals over the correlation period and compared with a threshold. If a satellite signal is present in the data a correlation peak will be generated that will transcend the pre-defined threshold. If the correlation peak is less than the threshold then the carrier and code NCO's are shifted by a certain amoun t determined by the discriminator algorithms and then the process is repeated.

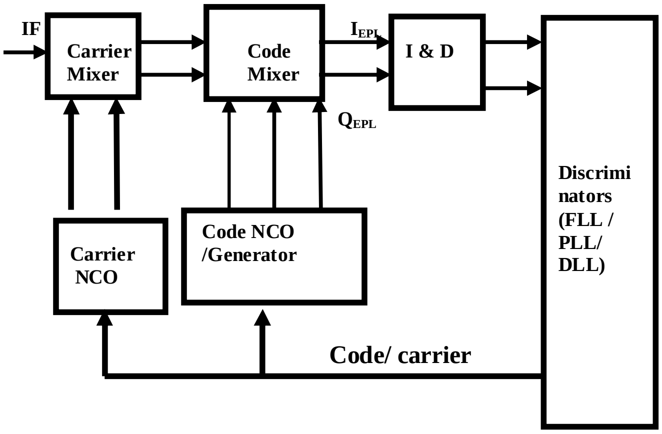

Fig.1 Time domain acquisition architecture

C. *Frequency Domain Algorithm*

Signal processing theory states that the correlation operation in time domain is equivalent to multiplication in the frequency domain [1]. Therefore, in the frequency domain approach, both the locally generated signals and the incoming satellite signals are first converted into the frequency domain using DFT or FFT. Before multiplying these two signals, one of the signals is conjugated because in the frequency domain only circular correlation is used. An IFFT algorithm then converts this data back into the time domain and the output is compared with a threshold for the signal detection .

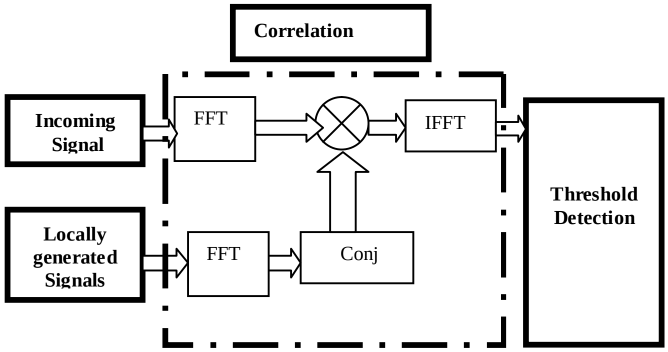

Fig.2 Frequency domain acquisition architecture

The FFT based acquisition algorithm is shown in the Fig. 2. Assuming $x_1(m)$ is the satellite signal and $x_2(m)$ is the locally generated signal, the correlation can be expressed in time domain as $r(n)=\sum_{m=0}^{N-1} x_1(m)x_2(m+n)$ and in the frequency domain as $R(k)=X_1(k)X_2(k)$. The inverse Fourier function of $R(k)$ yields the correlation peak, i.e., $R(k)=X_1^{-1}(k)X_2(k)$. For real signals, the magnitude of $R(k)$ can be computed as $\left|R(k)\right|=\left|X_1^*(k)X_2(k)\right|$ where * represents the conjugate operation.

III. FREQUENCY DOMAIN ALGORITHMS

A. DFT and FFT

The DFT algorithm is fundamental to the processing of the signals in the frequency domain. It converts a sequence of data $x(n)$ of length N into a sequence $X(k)$ of length N, according to the equations (5) and (6)

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{kn} \qquad 0 \le k \le N-1 \qquad (5)$$

$$W_N = e^{-j2p/N} \qquad (6)$$

where W_N is the twiddle factor. For each sample of k, the computation of $X(k)$ involves N complex multiplications and $N-1$ complex additions, and for the entire N samples it takes N^2 complex multiplications and $N(N-1)$ complex additions. The inefficiency of this direct approach primarily arises from the fact that it does not exploit the symmetry and periodicity properties of the phase component W_N as given in the equation (6). These properties are given in the equation (7) below:

$$\begin{aligned} &\mathit{Symmetry}\ \Pr\mathit{operty}: \quad W_N^{k+N/2}=-W_N^k \\ &\mathit{Periodicity}\ \Pr\mathit{operty}: \quad W_N^{k+N}=W_N^k \end{aligned} \qquad (7)$$

By exploiting these two properties the FFT algorithm efficiently performs the frequency conversion. By splitting the N points into two $N/2$ sequences, $x(2n)\ and\ x(2n+1)$ where $n=0,1,2...N/2-1$ the $X(k)$ can be simplified into

$$X(k)=\sum_{m=0}^{(N/2)-1} x(2m)W_N^{2mk}+\sum_{m=0}^{(N/2)-1} x(2m+1)W_N^{(2m+1)k} \qquad (8)$$

By using the property $W_N^2=W_{N/2}$ the equation (8) can be simplified into

$$X(k)=X_1(k)+W_N^k X_2(k) \qquad k=0,1,2........N-1 \qquad (9)$$

where $X_1(k)$ and $X_2(k)$ are the N/2 DFT sequences. Therefore, the total number of calculations is reduced to $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions.

B. Radix-4 FFT

The radix-4 algorithm requires the samples to be a power of 4, i.e. $N=4^n$. A decimation-in-time algorithm will be analysed here as it was used for the implementation. The N point data signals are split into four different sequences:

$$x(4n), x(4n+1), x(4n+2), x(4n+3) \qquad where\ n=0,1,2.....(N/4)-1.$$

The algorithm is given in the equations (10) to (14)

$$X(p,q)=\sum_{l=0}^{3}\left[W_N^{lq}H(l,q)\right]W_4^{lp} \qquad (10)$$

$$H(l,q)=\sum_{m=0}^{(N/4)-1} x(l,m)W_{N/4}^{mq} \qquad (11)$$

$$p=0,1,2,3;\ l=0,1,2,3;\ q=0,1,2....N/4-1 \qquad (12)$$

$$x(l,m)=x(4m+1) \quad and \quad X(p,q)=X\left(\frac{N}{4}p+q\right) \qquad (13)$$

The four N/4 DFT sequences $H(l,q)$ are combined to give the N point DFT. The procedure for combining the four DFT's is the decimation-in-time butterfly architecture, which can be expressed in matrix format as

$$\begin{bmatrix} X(0,q) \\ X(1,q) \\ X(2,q) \\ X(3,q) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix} \begin{bmatrix} W_N^0 H(0,q) \\ W_N^q H(1,q) \\ W_N^{2q} H(2,q) \\ W_N^{3q} H(3,q) \end{bmatrix} \qquad (14)$$

The number of complex multiplications and additions required to compute a N point DFT using radix-4 algorithm are $(3/8)N\log_2 N$ and $N\log_2 N$ respectively [4]. The number of complex additions remains the same in both the radix-2 and radix-4 algorithms.

IV. SIMULATION EXPERIMENTS

The radix-4 algorithm mentioned in the previous section is implemented in the acquisition loop as shown by red blocks in the Simulink model in figure 3. The signal processing toolbox does not have a radix-4 block, therefore an Embedded Matlab function was used to develop this algorithm. The overall performance was analysed by testing this algorithm with a real data set provided by Tsui. The specifications of the data set are: IF frequency - 21.25MHz, and Sampling frequency - 5MHz. Therefore, in a 1msec correlation period the total number of samples were 5000. Two options were possible to process these data samples in the radix-4 algorithms – decimate the sequence to 4096 samples, or interpolate to 16384 samples. As the former approach involved SNR degradation due to bandwidth loss, the latter was adopted for this implementation. The local signals, however, are generated directly at the required sampling frequency.

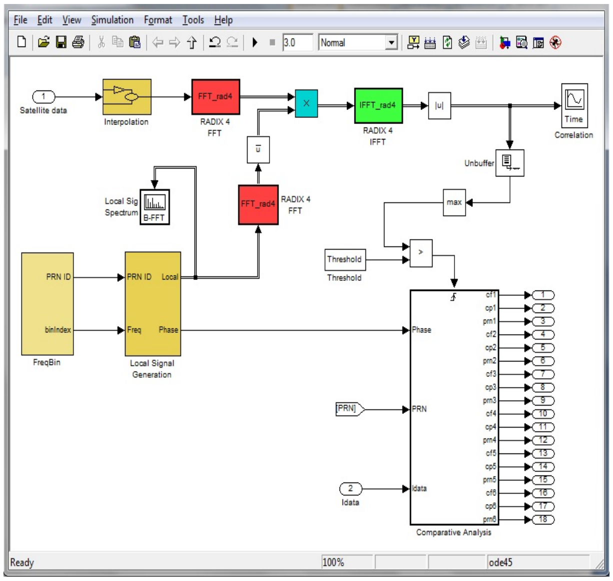

Fig.3 Simulink Model of Radix-4 FFT based acquisition

The PRN codes for all the 32 satellites were pre-computed and stored in a look-up table. As 16384 samples are required for the radix-4 algorithm, the C/A code is generated for so many samples, and then the logic given in Fig. 4 is used to read all the samples.

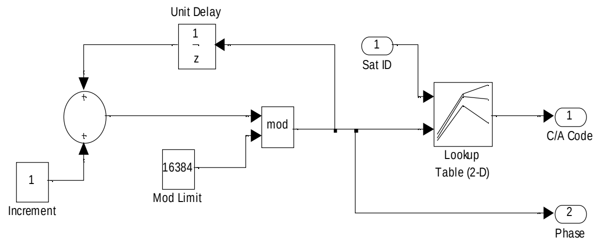

Fig.4 C/A Code Generation Logic using look-up table

The total Doppler with search range was +/-7KHz with incremental steps of 500Hz. Therefore, a total of 29 bins were searched to cover the entire range. The Simulink model to search the Doppler range and also to generate the IDs for all the 32 satellites is shown in the Fig. 5.

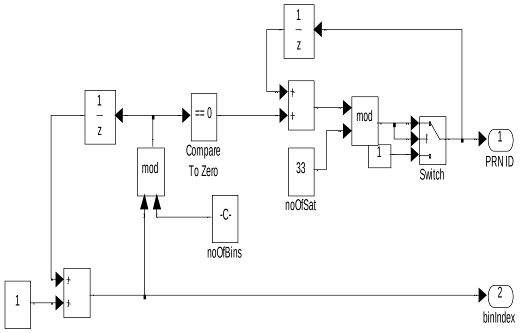

Fig.5 Doppler bins search logic and PRN ID generation

With the models shown in figures 3 to 5, the algorithm was able to detect 7 satellites. However, the results of only one satellite, i.e., PRN 6, are shown here in Table I. To verify the accuracy of the results, the estimates provided in Tsui's book are also given in the table. The results show that the estimated Doppler frequency and code phase values closely match the results given in the book.

TABLE I
ACCURACY OF ACQUISITION PARAMETERS USING RADIX-4

PRN 6	**Radix-4 FFT**	**Tsui Results**
Doppler Frequency (MHz)	21.2458 (only coarse acquisition)	21.24579 (includes fine acquisition)
PRN Code Phase Offset (chips)	2885	2884

The correlation plot acquiring the PRN 6 is shown in the Fig. 6. The horizontal axis is scaled for the correlation period in milliseconds.

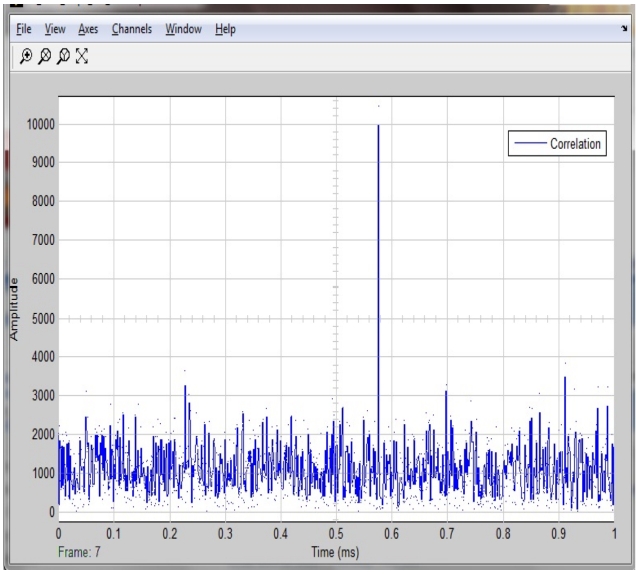

Fig.6 Correlation Plot for PRN 6

The number of arithmetic operations required to compute the 16384 point DFT using both radix-2 and radix-4 algorithms are obtained through the simulations results and tabulated in the Table II.

TABLE II
RADIX-2 VS RADIX-4 ARITHMETIC OPERATIONS

N = 16384	Radix-4		Radix-2	
	Simulations	Theory	Simulations	Theory
Complex Multiplications	89721	86016	117756	114688
Complex Additions	232144	229376	233846	229376

It is evident that the number of complex multiplications required for the radix-4 algorithm is comparatively less than the radix-2 algorithm. However, there is not a significant difference in the complex additions. From the results obtained the efficiency of the radix-4 algorithm is calculated to be about 23%.

V. CONCLUSION

The acquisition algorithms are the most challenging components of a GPS receiver because not only these loops detect the satellites signals by scanning the entire uncertain region but also they need to do it within a short duration. With the proliferation of SDR technology most of the modern receivers use the frequency domain approach because of the reduced computations and improved speeds. This paper proposed a radix-4 FFT algorithm to improve the efficiency of the conventional DFT algorithm for real-time implementations. Experiments with real data sets were conducted and the results show that the radix-4 approach is about 23% more efficient than the radix-2 algorithm.

REFERENCES

[1] S. K. Mitra, *Digital Signal Processing – A Computer Based Approach*, Tata McGraw-Hill Edition, India, ISBN 0-07-463723, 1999.

[2] J.B.Y. Tsui, Fundamentals of Global Positioning Receivers – A Software Approach, John Wiley & Sons, Inc., 2000.

[3] E.D. Kaplan, Understanding GPS: Principles and Applications, Artech House, M.A., 1996.

[4] Dougles F. Elliott & K. Ramamohan Rao, Fast Transforms: Algorithms, Analyses, Applications, Academic Press Inc., London, UK, 1982.

[5] Fran van Graas et.al., Comparison of two approaches for GNSS receiver algorithms: batch processing and sequential processing considerations, ION GNSS, 2005.

[6] D. Van Dee & A. Coenen, New Fast GPS code-acquisition technique using FFT, Electronics Letters, vol. 27, pp. 158-160, 1991.

[7] O'Driscoll, Cillian, Murphy, C. Colin, Performance Analysis of an FFT Based Fast Acquisition GPS Receiver, pp. 1014-1025, NTM, The Institute of Navigation, CA, 2005.

[8] M. Djebbouri & D.Djebbouri, Fast GPS Satellite Signal Acquisition, Electronics Letters.com, no. 1/10/2003.

[9] G.Bi & Y.Q. Chen, Fast DFT algorithms for length N=q*2, IEEE Transactions on Circutis and Systems II: Analog and Digital Signal Processing, Vol.45, Issue 6, pp. 685-690, 1998

[10] B.W. Parkinson & Jr. J.J. Spilker, Global Positioning System: Theory and Applications, AAAI, USA, 1996.