2006 IEEE Conference on
Systems, Man, and Cybernetics
October 8-11, 2006, Taipei, Taiwan

Acquisition of GPS Software Receiver Using Split-Radix FFT

W. H. Lin, W. L. Mao, H. W. Tsao, F. R. Chang, and W. H. Huang

***Abstract*—Acquisition is the first step in the signal processing section of a Global Position System (GPS) receiver. It detects the presence of a GPS signal received by antenna and provides estimates of code phase and Doppler frequency to the tracking loops. Three acquisition methods in frequency-domain processing are compared, i.e., namely radix-2 FFT, split-radix-2/4 FFT and. The expectancy of the split-radix algorithm is increasing the acquisition efficiency. It is shown that the split-radix-2/8 FFT can save 25% of data loads and stores compared with split-radix-2/4 FFT. In order to simulate the overall system in an efficient way, the acquisition functions are developed in software architecture running on PC.**

I. Introduction

Global positioning system (GPS) receivers must detect the presence of GPS signal to track and decode the information which enables position computation. Tracking of the signal is possible only after they have been acquired, so acquisition is the first step in GPS signal processing scheme. The acquisition must ensure that the signal is acquired at the correct code phase and carrier frequency.

Software receivers provide flexibility, reconfigurability and high fidelity among many other advantages. There has been a variety of software based receiver architectures for GPS and other navigational system

The fast algorithm for computing Discrete Fourier Transform (DFT) has been studied for many years. The split-radix FFT approach is proposed by Duhaamel and Hollmann in [16] differs from that of Cooley-Tukey in decomposition process. For example, in the Cooley-Tukey radix-2 FFT, a radix-2 index map is used to factor both the even and odd index term.

The split-radix fast Fourier transform FFT algorithm [16] is known as the FFT algorithm that uses the lowest total number of operation. The split-radix idea can be extended to other radix pairs. When we want to enhance the speed of GPS software receiver, we use the algorithm of FFT that is radix-2, split-radix-2/4, and split-radix-2/8. We will compare their stores, loads, additions, and multiplications.

The rest of paper is organized as follows. Section II presents the architecture of GPS System. In Section III, we will introduce Split-radix 2/8 fast Fourier transform and other algorithms about FFT. In the section IV, we will show the simulation results and some table that compare each algorithms of FFT. Final is our conclusions about the GPS software receiver using FFT algorithm.

W. H. Lin, H. W. Tsao, and F. R. Chang are with the Graduate Institute of Communication Engineering and Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan. (e-mail: roger1109@gmail.com, frchang@cc.ee.ntu.edu.tw, tsaohw@cc.ee.ntu.edu.tw)

M. L. Mao is with the Graduate Institute of Electrical Engineering and Department of Electrical Engineering, Mingchi University of Technology, Taipei county, Taiwan (e-mail: wlmao@ccsun.mit.edu.tw)

W. H. Huang is with Evermore Inc, Hsinchu City, Taiwan (e-mail: maped@emt.com.tw).

II. The Architecture of GPS System

The basic GPS receiver discussed in this paper is show in Figure 1. The signals transmitted from the antenna. Through the radio frequency (RF) chain the input signal is amplified to proper amplitude and the frequency is converted to desired output frequency. An analog-to-digital converter (ADC) is utilized to digitize the output signal. The antenna, RF chain and ADC are the hardware used in receiver

After the signal is digitized, software is used to process it, and that is why this paper has taken a software approach. Acquisition means to find the signal of a certain satellite. The tracking program is used to find the phase transition of the navigation data. In conventional receiver, the acquisition and tracking are performed by hardware. From the navigation data phase transition the subframes and navigation data can be obtained. Ephemeris data and pseudorange can be obtained from the navigation data. The ephemeris data are used to obtain the satellite positions. Finally, the user position can be calculated for the satellite positions and the pseudoranges. Both the hardware used to collect digitized data and software used to find the user position.

There are basically two types of signals: the coarse/ acquisition (C/A) and precision (P) codes. The actual P code is not directly transmitted by the satellite, but it is modified by a Y code, which is often referred to as the P(Y) code. The P(Y) code is not available to civilian user and primarily used by the military. In general, in order to acquire the P(Y) code, the C/A code is usually acquired first.

The GPS signal contains two frequency components: link 1 (L1) and link 2 (L2). The center frequency of L1 is at 1575.42 MHz and L2 is at 1227.6 MHz. These frequencies are coherent with a 10.23 MHz clock. These two frequencies can be related to the clock frequency as

$$L1 = 1575.42MHz = 154 \times 10.23MHz \quad (1)$$

$$L2 = 1227.6\ MHz = 120 \times 10.23MHz \quad (2)$$

The signal structure of the satellite may be modified in the future. However, at the present time, the L1frequency contains the C/A and P(Y) signals, while the L2 frequency contains only the P(Y) signal. The C/A and P(Y) signals in the L1 frequency are quadrant phase of each other and they can be

1-4244-0100-3/06/$20.00 ©2006 IEEE

written as:

$$S_{L1} = A_p P(t) D(t) \cos(2\pi f_1 t + \phi) + A_c C(t) D(t) \sin(2\pi f_1 t + \phi) \quad (3)$$

where S_{L1} is the signal at L1 frequency, A_c is the amplitude of the P code, P(t) = ±1 represents the phase of the P code, D(t) = ±1 represents the data code, f_1 is the L1 frequency, ϕ is the initial phase, A_c is the amplitude of the C/A code, C(t) = ±1 represents the phase of the C/A code. In this equation the P code is used instead of the P(Y) code. The P(Y), C/A, and the carrier frequencies are all phase locked together. The role of acquisition is to provide a coarse estimate of the code phase and the Doppler to the tracking loops.

Acquisition performs the task of detecting a GPS signal from the signals received by the antenna by exploiting the autocorrelation and cross correlation properties of GPS coarse/acquisition (C/A) codes.

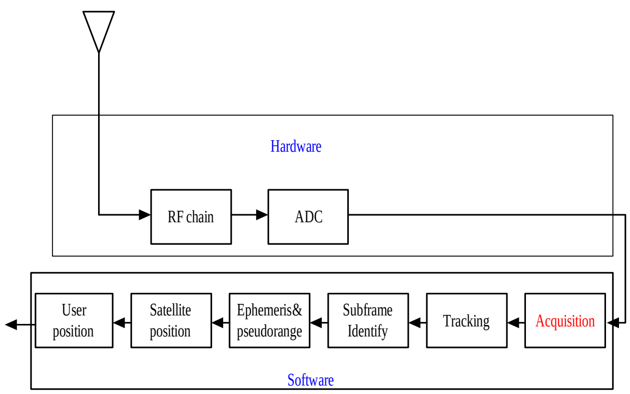


Fig 1: The architecture of GPS system

A. Acquisition

The purpose of acquisition is to determine visible satellite and coarse values of carrier frequency and code phase of the satellite signal.

The satellites are differentiated by 32 different PRN sequences. The second parameter, code phase, is the time alignment of PRN code in current block data. The code phase is necessary to be able to generate a local PRN code that is perfectly aligned with incoming data. Only when this is the case, the incoming code can be removed from the signal. The third parameter is the carrier frequency, which in case of down conversion corresponds to the IF. The IF should be known from the L1 carrier frequency of 1575.42MHz and from the expected value. The line of sight velocity of the satellite causes a Doppler effect resulting in a higher or lower frequency. In the worst case, the frequency can deviate up to ± 5kHz. It is important to know the frequency of the signal to be able to generate a local carrier signal carrier signal. The signal is used to remove the incoming carrier from the signal .In most case it is sufficient to search the frequency in steps of 250Hz.

In order to accomplish the search in a short time, the bandwidth of the searching means taking many steps to cover a wide bandwidth filter will provide relatively poor sensitivity.

In order to track and decode the information in GPS signal, an acquisition method must be used to detect the presence of the signal. Once the signal is detected, the necessary parameters must be obtained and passed to tracking program. From tracking program information such as the navigation data can be obtained.

There are three acquisition methods. First, serial search acquisition, parallel frequency space search acquisition, and parallel code phase search acquisition. Comparison with three methods, we will use the third method. It is to provide a better estimate of frequency to the carrier loop, and the parallel code phase search algorithm has better execution time and parameter estimation.

B. Parallel Code Phase Search Acquisition

In the same way as with the order acquisition methods, the implementation of this one is very straight-forward, as it can be implemented directly based on the block diagram in Fig. 2.

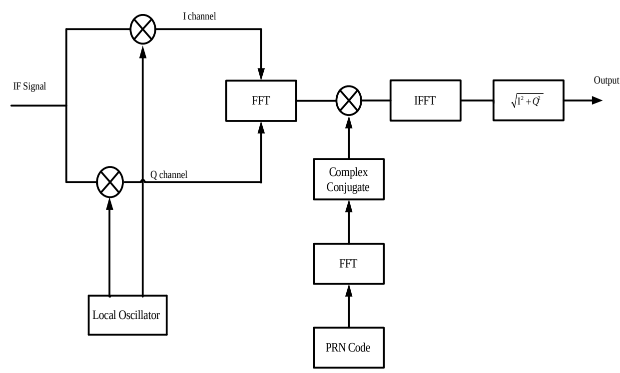


Fig 2: Parallel Code Phase Search Acquisition

One difference in this method is that only one PRN code should be generated for each acquisition. That is, it is not necessary to take the 1023 different code phase into account.

The first step of the method is multiply the incoming signal with locally generated cosine and sine carrier wave respectively , giving an I and a Q signal component. These two are combined as a complex input to the Fourier transform. The result of the Fourier transform is multiplied with the result of lower branch of the block diagram in Figure.2.This signal is created as follow. The PRN generator generates a code with no code phase. As in the implementation of the other acquisition algorithm, the code generation is performs a Fourier transform of the PRN code, and the result is complex conjugated.

The results of the before mentioned multiplication are given as input to an inverse Fourier transform. The function has properties similar to FFT function regarding execution time. As mentioned in implementation of parallel frequency space search acquisition method above, the output from an FFT is complex .this is also the case for the IFFT, so the output needs the same processing as in that case. The absolute value is calculated for all components.

The goal of acquisition is to perform a correlation with the incoming signal and a PRN code. It is more convenient to make a circular correlation between the input and the PRN code without shifted code phase.

The cross-correlation between two finite length sequences x(n) and y(n) both with length N is calculated as:

$$z(n)=\sum_{m=0}^{N-1} x(n)\, y(n+m) \qquad (4)$$

The convolution between the two finite length sequence x(n) and y(n) is calculated as:

$$z(n)=x(n)\otimes y(n)=\sum_{m=0}^{N-1} x(n)\, y(n-m) \qquad (5)$$

Equation (4) and (5) show only difference between cross-correlation and convolution between finite length sequences is the sign y(n+m). The convolution can be transferred to frequency domain through Fourier transform.

$$\begin{aligned} Z(K) &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(m)\, y(n-m)\, e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N-1} x(m) e^{-j2\pi kn/N} \sum_{n=0}^{N-1} y(n-m) e^{-j2\pi k(n-m)/N} \\ &= X(k)Y(k) \qquad (6) \end{aligned}$$

The combination of equation (5) and (6) give the DFT of cross-correlation between x and y:

$$\begin{aligned} Z(K) &= \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} x(m)\, y(n+m)\, e^{-j2\pi kn/N} \\ &= \sum_{m=0}^{N-1} x(m) e^{j2\pi kn/N} \sum_{n=0}^{N-1} y(n+m) e^{-j2\pi k(n-m)/N} \\ &= X^{*}(k)Y(k) \qquad (7) \end{aligned}$$

Therefore, the (6) and (7) are almost similar to each other. The only difference is that the Fourier transform of one of the two input sequences should be complex conjugated before multiplication. Fig.2 is a block diagram of the parallel code phase search algorithm.

The incoming signal is multiplied by locally generated carrier signal. Multiplication with the carrier signal generates the I-arm signal, and multiplication with a 90°phase shifted version of the carrier signal generates the Q-arm signal. The two I and Q signals and combined to form a complex input signal x(n)=I(n)+jQ(n) to DFT function. The generated PRN code is transformed into the frequency domain and result is complex conjugated.

The Fourier transform of the input is multiplied with the Fourier transform of RPN code. The result of multiplication is transform into the time domain by an inverse transform. The absolute value of the output of the inverse Fourier transform represents the correlation between the input and the PRN code. If a peak is present in the correlation, the index of this peak marks the PRN code phase of the incoming signal.

C. Tracking

One might think that basic method of tracking a signal is to build a narrow-band filter around an input signal and follow it. In other word, while the frequency of the input signal varies over time, the center frequency of the filter must follow the signal. In the actual tracking process, the center frequency of the narrow-band filter is fixed, but a locally generated signal follows the frequency of the input signal. The phase of the input and locally generated signal follows the frequency of input signal. The phase of the input and locally generated signals are compared through a narrow-band filter. Since the tracking circuit has a very narrow bandwidth, the sensitivity is relatively high in comparison with the acquisition method.

The acquisition loop gives a coarse estimate of carrier Doppler and PRN code offset of incoming signal. Control is then handed over the variation in the carrier Doppler and code offset due to line of line of sight dynamics between the satellite and the receiver.

The unknown parameters of interest in incoming GPS signal are the carrier Doppler and carrier phase and the RPN code Doppler and code phase. These parameters are function of time because of time the relative motion between the satellite and user receiver. Hence it is important to track these parameters and get a very good estimate of the same. Another important function of the tracking loops is demodulated the navigation data form the incoming GPD signal. The PRN code phase can be used to determine the pseudorange whereas the carrier phase is used to accurately determine the small changes in pseudorange.

III. SPLIT-RADIX 2/8 FAST FOURIER TRANSFORM

Discrete Fourier transform DFT) is an important tool in digital signal processing. Many fast algorithms have been proposed to compute the discrete Fourier transform of signals. The split-radix-2/4 fast Fourier transform algorithm and the split –radix-2/8 fast Fourier transform are known as the FFT algorithms that use the lowest total number of operations. But the split-radix-2/8 FFT saves 25% of data loads and stores compared with the split-radix -2/4 fast Fourier transform algorithm.

A. Fast Fourier Transform

The radix-2 fast Fourier transform algorithm decomposes the N-point FFT into two (N/2)-point FFT. The decimation-in-frequency version is shown as follows:

$$X_{2k} = \sum_{n=0}^{(N/2)-1} \left(x_n + x_{n+(N/2)}\right)\cdot W_{\frac{N}{2}}^{nk} \qquad (8)$$

$$X_{2k+1} = \sum_{n=0}^{(N/2)-1} \left(x_n - x_{n+(N/2)}\right)\cdot W_N^{n}\cdot W_{\frac{N}{2}}^{nk} \qquad (9)$$

$$where \quad k=0,\cdots,(N/2)-1$$

The butterfly of radix-2 FFT algorithm is shown in Fig.3

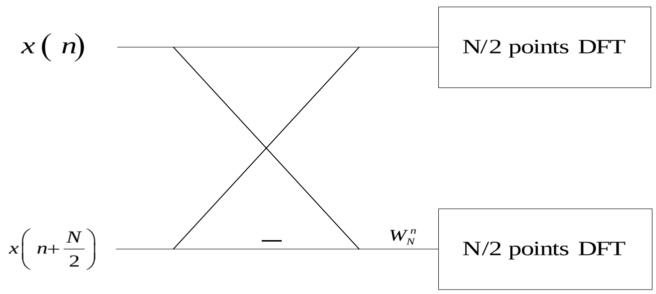


Fig.3 The butterfly of radix-2 FFT

B. *Split-radix-2/4 Fast Fourier Transform*

The split-radix-2/4 fast Fourier transform algorithm developed by Duhamel and Hollmann uses radix-2 decomposition for even part and radix-4 for odd part. This method decomposes the N-point FFT into one (N/2)-point FFT and two (N/4) –point FFT. Using the split-radix -2/4 FFT method, the formula for the even part of FFT is as

Even Part:

$$X(2k)=\sum_{n=0}^{\frac{N}{2}-1}(x(n)+x(n+\frac{N}{2}))W_{\frac{N}{2}}^{nk} \quad (10)$$

Odd Part:

$$\left.\begin{cases} X(4k+1)=\sum_{n=0}^{\frac{N}{4}-1}[(x(n)-x(n+\frac{N}{2}))-j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))]W_N^{n}W_{N/4}^{nk} \\ X(4k+3)=\sum_{n=0}^{\frac{N}{4}-1}[(x(n)-x(n+\frac{N}{2}))+j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))]W_N^{3n}W_{N/4}^{nk} \end{cases}\right\} (11)$$

Fig.4 shows the structure of split-radix-2/4 FFT algorithm. The total numbers of operation for this algorithm is lower the radix-2 FFT algorithm.

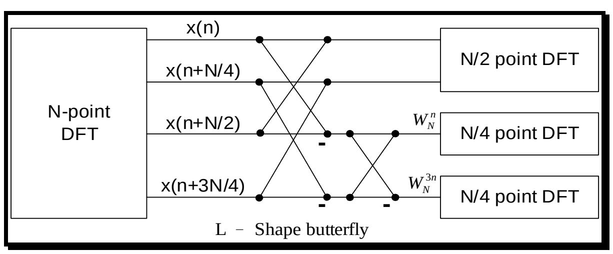


Fig.4 The structure of the split-radix-2/4 FFT algorithm

C. *Split-Radix-2/8 Fast Fourier Transform*

Takahashi has developed the split-radix-2/8 fast Fourier transform algorithm. This algorithm is an extension of split-radix-2/4 FFT algorithm. The split-radix-2/8 FFT algorithm decomposes the even part using radix-2 method while using radix-8 for the odd part. Thus, he formula for even part is still the same:

Even Part:

$$X(2k)=\sum_{n=0}^{\frac{N}{2}-1}(x(n)+x(n+\frac{N}{2}))W_{\frac{N}{2}}^{nk}$$

$$,k=0,1,2,...,(N/2-1) \quad (12)$$

Odd part:

$$X(8k+1)=\sum_{n=0}^{\frac{N}{8}-1}\{[(x(n)-x(n+\frac{N}{2}))-j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))] +\frac{1}{\sqrt{2}}[(1-j)(x(n+\frac{N}{8})-x(n+\frac{5N}{8}))-(1+j)(x(n+\frac{3N}{8})-x(n+\frac{7N}{8}))]\}W_N^{n}W_N^{8nk} \quad (13)$$

$$X(8k+3)=\sum_{n=0}^{\frac{N}{8}-1}\{[(x(n)-x(n+\frac{N}{2}))+j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))] -\frac{1}{\sqrt{2}}[(1+j)(x(n+\frac{N}{8})-x(n+\frac{5N}{8}))-(1-j)(x(n+\frac{3N}{8})-x(n+\frac{7N}{8}))]\}W_N^{3n}W_N^{8nk} \quad (14)$$

$$X(8k+5)=\sum_{n=0}^{\frac{N}{8}-1}\{[(x(n)-x(n+\frac{N}{2}))-j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))] -\frac{1}{\sqrt{2}}[(1-j)(x(n+\frac{N}{8})-x(n+\frac{5N}{8}))-(1+j)(x(n+\frac{3N}{8})-x(n+\frac{7N}{8}))]\}W_N^{5n}W_N^{8nk} \quad (15)$$

$$X(8k+7)=\sum_{n=0}^{\frac{N}{8}-1}\{[(x(n)-x(n+\frac{N}{2}))+j(x(n+\frac{N}{4})-x(n+\frac{3N}{4}))] +\frac{1}{\sqrt{2}}[(1+j)(x(n+\frac{N}{8})-x(n+\frac{5N}{8}))-(1-j)(x(n+\frac{3N}{8})-x(n+\frac{7N}{8}))]\}W_N^{7n}W_N^{8nk} \quad (16)$$

The structure of the split-radix-2/8 FFT algorithm is shown in Fig.5.

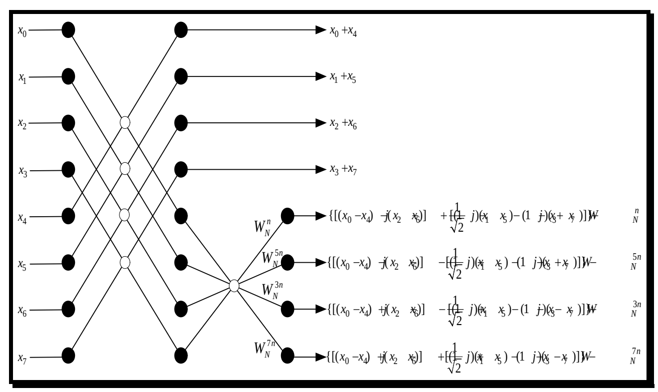


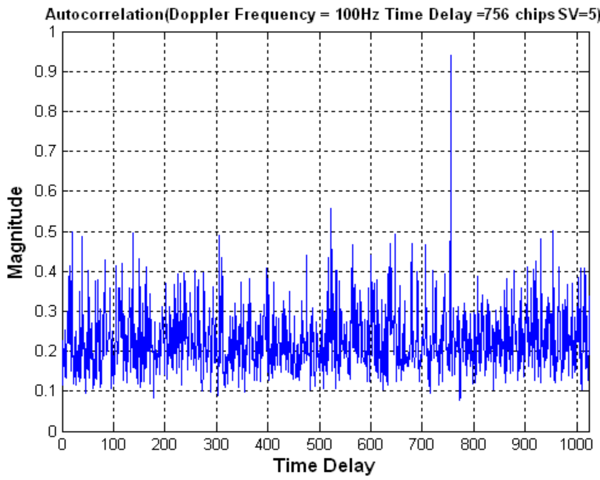
Fig.5 The structure of the split-radix-2/8 FFT algorithm

IV. Simulation Results

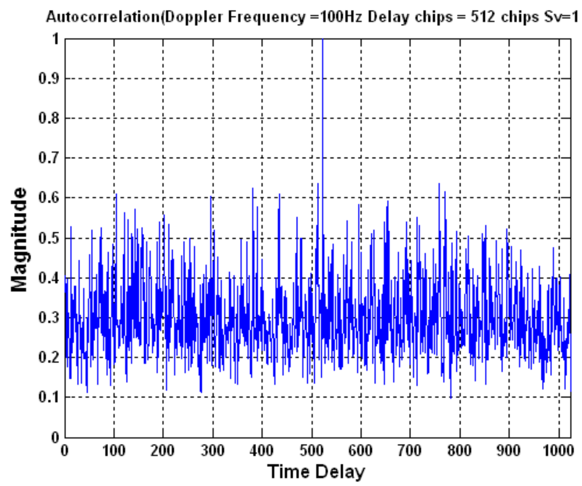
Now, an important problem is discussed before acquisition process: sampling rate problem. An important factor in selecting the sampling frequency is related to the C/A code chip rate. The C/A code chip is 1.023MHz, and the sampling would be a multiple number of the chip rate. For our example, a sampling frequency of 16.384 MHz is used.

Now, see our data specification. The chip rate is of 1.023MHz, so 16× chip rate is equal to 16.384MHz. But the sampling rate = 16.384 MHz, it means that in one millisecond, 16384 points will be chosen. But 16×chip rate = 16.368MHz means one millisecond, the picked up sample is 16368. Thus, zero must be inserted. Inserting zero is only sure process, because in tracking mode, synchronized circuit such as Costas PLL will be used.

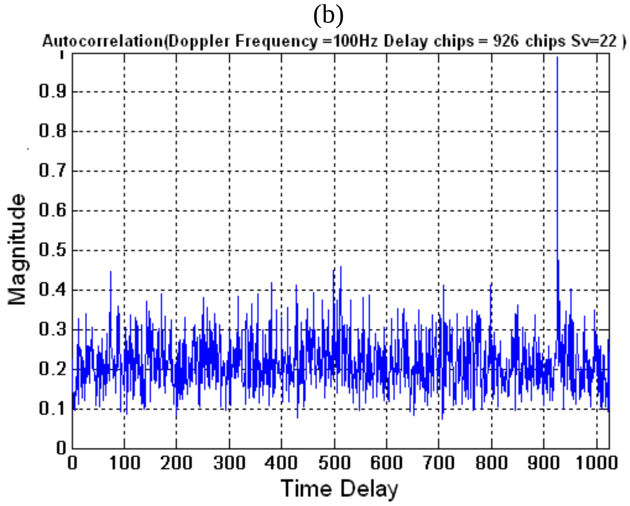
If signal strength is more than threshold, we will declare the signal is found. And we will send some information to tracking mode. When the autocorrelation function is calculated, how the program detects where is the peak value? Especially when the signal power has a Doppler frequency shift, the problem is hard to get an optimal solution.



(a)



(b)



(c)

Fig. 8 Autocorrelation function with 16K FFT using (a) split-radix-2, (2) split-radix-2/4, and (3) split-radix-2/8 methods

Simulation results show the chip delay and Doppler frequency for different GPS satellites. The input SNRs are set at -110 dBm. In Fig. 6, the autocorrelation functions are conducted using the split-radix-2, split-radix-2/4, and split-radix-2/8 methods to demonstrate the acquisition process. The split-radix 2/8 scheme can have the better computationally efficiency and preserve the information of code offset effectively.

Table I: Number of nontrival real multiplication and two real additions per complex multiplication for N complex number

	Radix-2		Split-radix 2/4		Split-radix 2/8	
	Mults	Adds	Mults	Adds	Mults	Adds
256	2316	5380	1656	5008	1660	5140
512	5644	12292	3988	11380	3932	11676
1024	13324	27652	9336	25488	9148	26180
2048	30732	61444	21396	56436	20892	57996
4096	69644	135172	48248	123792	46844	127200
8192	155660	294916	107412	269428	103900	276988
16384	344076	638980	236664	582544	228412	599076

Table II: Number of Loads, Stores Multiplications, and Additions for GENERAL BUTTERLIES Used in FFT Algorithm. The number of Loads includes loading all of the constants.

Algorithm	Loads	Stores	Mults	Adds
Radix-2	6	4	4	6
Split-radix-2/4	12	8	8	16
Split-radix-2/8	24	16	20	44

Table III: Number of Loads, Stores, Multiplications, and Additions. Use the $N \log_2 N$ FFT algorithm to compute an N point DFT. Lower order terms have been omitted.

Algorithm	Loads	Stores	Mults	Adds
Radix-2	3	2	2	3
Split-radix-2/4	2	4/3	4/3	8/3
Split-radix-2/8	3/2	1	5/4	11/4

From Table I II, and III the Split-radix-2/4 FFT and the Split-radix 2/8 FFT are compared. Obviously, the conventional split-radix -2/8 saves 25% of data loads and stores compared with the split-radix-2/4 FFT algorithm.

V. CONCLUSIONS

In this paper, three very useful algorithms have been presented in software GPS receiver design. The software modules, taking such IF samples as its input, perform the baseband signal processing and navigation solution computation. The proposed schemes for acquisition in frequency domain are the Fast Fourier Transform: radix-2, split-radix-2/4 and split-radix-2/8. It is more efficient and speedy to use the split-radix-2/8 algorithm than the split-radix-2/4 algorithm because of the less computation burdens and memory requirements. The acquisition of software receiver, running on a standard PC, is developed and evaluated in civil applications.

ACKNOWLEDGMENT

This research was supported by the Evermore INC., Hsinchu City, Taiwan, R. O. C.

REFERENCES

[1] J. B. Y. Tsui, *"Fundamentals of Global Positioning System Receivers, A software Approach*", Wiley Inter-Science, 2005.

[2] E. D. Kaplan, *"Understanding GPS: Principles and Applications"* Artech House, 1996..

[3] P. Rinder, and N. Bertelsen, *"Design of a single Frequency GPS software receiver* AAU 2004

[4] C. Ma,G. Lachapelle, and M. E. Cannon, *"Implement of a software GPS Reveiver",* ION GNSS 17th International Technical Meeting of Satellite Division, 21-24 Sept.2004 Long Beach, CA

[5] M. Djebbouri and D. Djebbouri, *"Fast GPS Satellite Signal Acquisition"*, Electronics Letters.com, no.1/10/2003.

[6] M. S. Braasch and A.J. Van Dierendonck, "GPS Receiver Architectures and Measurements", Proceeding of the IEEE, vol. 87, no.1, pp.48-64, Jan 1999.

[7] A. J. Van Dierendonck, P. Fenton, and T. Ford, *"Theory and performance of narrow correlator spacing in a GPS receiver,"* Navigation: J. Inst. Navigation, vol. 39, no. 3, pp. 265–283, Fall 1992.

[8] P. Ward, *"Satellite signal acquisition and tracking,"* in Understanding GPS: Principles and Applications, E. Kaplan, Ed. Boston: Artech House, ch. 5, pp. 119–208, 1996.

[9] B. W. Parkinson, and Jr. J. J. Spilker: *'Global positioning system: theory and applications'* (AAAI, USA, 1996.)

[10] S. M. Kay, 'Fundamentals of statistical signal processing: estimation theory', PTR Prentice Hall, New Jersey, 1993.

[11] A. Sadat, W. B. Mikhael, *"Fast Fourier Transform for High Speed OFDM Wireless Multimedia System"*, Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium, vol 2, pp. 938 – 942, Aug. 2001.

[12] Takahashi, D.; "An extended split-radix FFT algorithm", Signal Processing Letters, Vol 8, Issue 5, pp. 145~ 147, May 2001.

[13] S. Bouguezel, M. O. Ahmad,, M.M.S. Swamy, "*Arithmetic complexity of the split-radix FFT algorithms*", Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference, Vol 5, pp. 137-140, March 2005.

[14] G. Bi; Y. Q. Chen, "*Fast DFT algorithms for length $N=q*2^m$*", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol 45, Issue 6, pp. 685-690, June 1998.

[15] H. V. Sorensen, M. T. Heideman, and C. S. Burrus, "On computing the split-radix FFT," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-34, pp. 152–156, Feb. 1986.

[16] P. Duhamel and H. Hollmann, "Split-radix FFT algorithm," Electron. Lett., vol. 20, pp. 14–16, Jan. 1984.

[17] A. V. Oppenheim and R. W. Schafer, "*Discrete-Time Signal Processing"*, Englewood Cliffs, NJ: Prentice-Hall, 1989.