Design of a Real-Time GPS Satellite Signal Simulator on the MULTISHARC Parallel Processing System

S. V. M. K. PRASAD, V. VAIDEHI, P. V. RAMAKRISHNA, C. N. KRISHNAN, T. ANANDA SARANGARAM, G. BOOPALAN, N. NAGALAKSHMI, and S. NAPPINNAI
Department of Electronics Engineering, Madras Institute of Technology, Anna University, Chennai, India

Received October 2002; Revised April 2003

ABSTRACT: *A GPS signal simulator is a versatile instrument used for testing and performance evaluation of a GPS receiver during its design and manufacture. A real-time GPS signal simulator (RTGPSSS) outputs the L1 signal of GPS satellites as received by the GPS receiver at its antenna. The generation of the GPS L1 signal for our design specification requires the transmission of a coarse acquisition (CA)-code, which is a pseudorandom noise (PRN) code at a rate of 1.023 Mchip/s, and the calculation of Doppler frequency at a rate of 19.55 μs under high receiver dynamics. These critical timing requirements necessitate parallel processing techniques. In this paper, the design of a 12-channel RTGPSSS using a digital signal processor (DSP)–based parallel processing system is presented. The effects of receiver dynamics, receiver clock drift, multipath, and atmosphere anomalies (ionospheric and tropospheric delay) are also incorporated in the simulated GPS signal.*

INTRODUCTION

The GPS [1–3] all-weather, all-time facility available worldwide enables one to know one's position fix and time very accurately whenever and wherever desired. GPS provides spread-spectrum signals that can be processed in a GPS receiver, enabling the receiver to compute position, velocity, and time. At least four satellites are used to compute position in three dimensions and the time offset in the receiver clock [4–6].

A real-time GPS signal simulator (RTGPSSS) is an instrument capable of simulating a high-fidelity L1 coarse acquistion (C/A)-code signal, which is fed at the radio frequency (RF) front end of the GPS receiver. These signals are similar to the one that exists at the antenna of the GPS receiver in real time. RTGPSSS test facilities are used to test the software and hardware of GPS receivers and their augmentation systems. Almost any type of GPS receiver test procedure requires the use of an RTGPSSS. These simulators replicate transmitted RF signals from satellites visible at a receiver's location. The simulators usually have a trajectory profile generator for dynamic receiver applications.

GPS signals are modeled for static and dynamic user environments. The generated signal is incorporated with all the atmospheric effects, such as ionospheric and tropospheric delays.

The signal generated by the GPS signal simulator is an L1 C/A-code signal given by

$$S(t) = \sum_{j=1}^{N} s_j(t) \tag{1}$$

where N visible satellites are present, and the simulator is to output S(t) in real time.

The simulated signal from satellite j would have the structure [7–9]

$$S_j(t) = \sum_{i=1}^{M} \sqrt{(2P_{ci})} D_j(t - \tau_i(t)) G_j(t - \tau_i(t)) \cdot \cos\left(2\pi f_{L1}(t - \tau_i(t)) + \phi_i(t)\right) \tag{2}$$

where M multipath components exist; P_{ci} is the received signal power from the j^{th} satellite in the i^{th} multipath: $D_j(t)$ is the 50 Hz navigation data of the j^{th} satellite; $G_j(t)$ is the length-1023 pseudorandom noise (PRN) code of the j^{th} satellite with a chipping rate of 1.023 Mchip/s; $\tau_i(t)$ is the delay in the i^{th} multipath component from the j^{th} GPS satellite to the receiver (inclusive of the unknown propagation delays, ionospheric and tropospheric delays, etc.); f_{L1} is the carrier frequency of the signal from the j^{th} satellite (1575.42 MHz); and $\phi_i(t)$ is the phase shift encountered by the signal.

In reality, RTGPSSSs are RF channel simulators. The commercially available RTGPSSSs are presented

NAVIGATION: *Journal of The Institute of Navigation*
Vol. 50, No. 1, Spring 2003
Printed in the U.S.A.

in [11–13]; however, their internal details are not available in the literature. The schemes presented in [14] and [15] use MATLAB toolboxes for simulating the signal in digital form (digital samples) and store it in memory. These samples are then modulated onto an RF carrier to provide digital construction of the simulated signal. However, these schemes require huge amounts of memory. Moreover, signal simulation depends upon the volume of data generated by the MATLAB toolboxes.

The recent increase in the performance of digital signal processors (DSPs) and field programmable gate arrays (FPGAs) greatly reduces the complexity of implementing a channel simulator. Hence the implementation of channel simulators using DSP processors and FPGAs has been the topic of recent research [16, 17]. However, the RF channel simulators on the market today are complex and costly [18]. In this paper, a low-cost solution for the design of RTGPSSSs using DSPs and FPGAs is considered.

The generation of a GPS signal requires the computation of signal parameters, such as amplitude, frequency, and phase, at different update rates. The calculation of these signal parameters requires a range of computations within a short time period. It is not possible to run all the above tasks with the required update rates on a single processor. Therefore, a multiprocessing system having suitable link ports for hardware interfacing is needed for the design of an RTGPSSS. Moreover, the GPS signal simulator should output the data on 12 channels simultaneously, and these outputs must be integrated with hardware to receive the intermediate frequency (IF) signal.

Examples of popular DSP-based parallel processing systems are the TI 320C40 processor and the ADSP SHARC-21060 processor [19]. However, the ADSP SHARC processor has a node speed of 120 MFLOPS and a link speed of 40 Mbytes. These parameters are better than those of the TMS320C40, and hence the SHARC-based multiprocessing (MULTISHARC) system was selected for this design. Simultaneous generation of signals in 12 ports is feasible using the MULTISHARC system as it supports parallel updating of data in the links using a built-in direct memory access (DMA) controller.

RTGPSSS specifications for this design are as follows [9]: velocity (v) = 10,000 m/s, acceleration (a_0) = 500 m/s^2, jerk (a_1) = 500 m/s^3, and pseudorate error = 0.04 m/s. Considering these specifications as maximum values, the corresponding Doppler frequency (f_d) is calculated as follows:

$$v = a_0 t + a_1 \cdot t^2/2 < 10{,}000 \text{ m/s} \qquad (3)$$

By substituting $a_0 = 500$ m/s^2 and $a_1 = 500$ m/s^3 in equation (3), we get $t = 5.5$ s. Doppler frequency $f_d = f_0 * v/c = f_0 * (a_0 t + a_1 * t^2/2)/c$, $df_d/dt = f_0(a_0/c + a_1 * t/c) = a_0/\lambda + a_1 * t/\lambda = 16.25$ Hz/ms, where $\lambda = f_0/c = 1575.42$ MHz/300000000 = 0.2 m.

Hence maximum frequency error is 8.125 Hz/ms, and the corresponding velocity error is given by $\lambda * 8.125$, which is equal to 1.625 m/s. If the frequency error is 0.2 Hz/ms, the corresponding velocity error is 0.04 m/s. Frequency update is the rate at which the frequency of the L1 signal should be calculated. To obtain a frequency error of 0.2 Hz/ms, the frequency update for every calculation should be done every 24 μs. Using these specifications, it is found that frequency update every 19.55 μs will result in a pseudorange rate error of less than 0.04 m/s.

The design of the RTGPSSS [20] consists of both software [21] and hardware [22] design. The software design deals with the computation of frequency, phase, and amplitude for all 12 channels. These parameters are given to the hardware to generate S(t). This paper presents the design and implementation of a 12-channel RTGPSSS on Spectrum's parallel processing system (Darlington SHARC3000 PCI motherboard), consisting of 10 SHARC (ADSP-21060) processors connected in a three-level tree topology. This design produces the parameters required for IF signal generation. This paper addresses the issues involved in the software design only. For a detailed description of the hardware design, see [22].

This paper first describes the implementation of the RTGPSSS in the DSP-based parallel processing system. Subsequently, a parallel algorithm for the computational block and results are presented.

IMPLEMENTATION DETAILS

A complete block diagram of the RTGPSSS is shown in Figure 1. In this figure, the HOST system and MPS (Multiprocessing System) relate to the software design, while the IF and RF blocks relate to the hardware design. Since the MPS does all the computation for signal generation, it is often referred to as the computational block.

Host Interface

In this design, a Pentium-based system acts as a host system. Receiver parameters, such as receiver initial position, receiver dynamics, and date and time of simulation, are input by the user. These parameters are provided to the MPS for signal generation. The host also displays the data coming from the MPS in a graphical user interface (GUI) environment.

Multiprocessing System (MPS)

To achieve high computational power, a DSP-based parallel processing system with 10 ADSP SHARC-21060 processors on board (Spectrum Darlington SHARC 3000 PCI) is used [23]. Each ADSP SHARC-

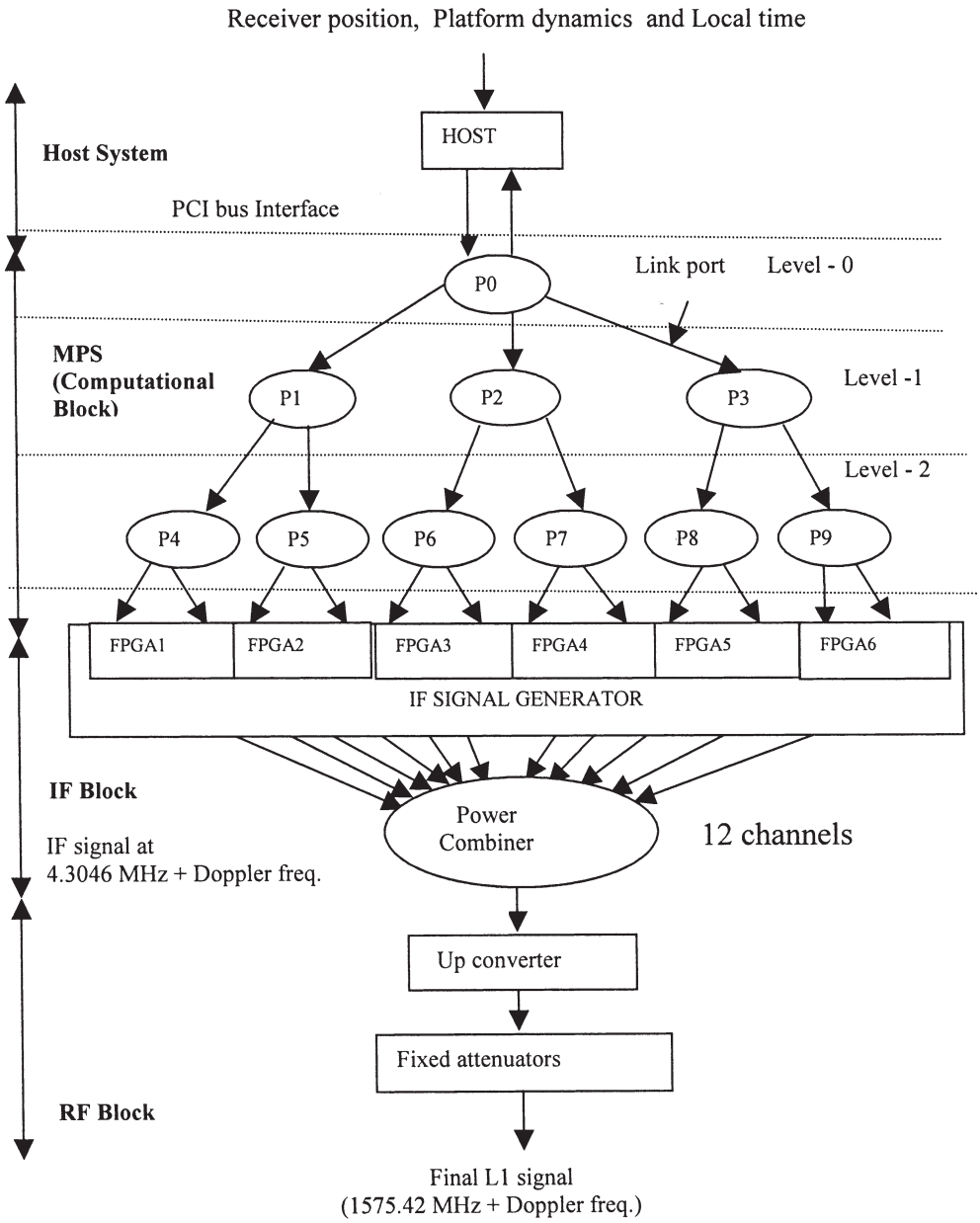

Fig. 1–Overall Block Diagram of 12-Channel Real-Time GPS Signal Simulator

21060 processor is capable of a computational speed of 120 MFLOPS, and easy interprocessor communication via six link ports with input/output (I/O) bandwidth of 40 Mbyte/s/link port [19].

The processors in the MULTISHARC system are configured in the following way: the root processor is named the level-0 processor (P0), while the other processors, which are connected to the root processor directly, are named level-1 processors (P1, P2, P3). The processors that are not connected directly to the root processor are named level-2 processors (P4, P5, P6, P7, P8, P9). The data from the root processor can reach a level-2 processor in two hops and vice versa.

To have a hierarchical data flow structure, a tree-connected architecture is derived, as shown in Figure 1. The data flow structure is based on the required update rate, which is the rate at which the parameter is calculated in that particular node. It has been found by extensive analysis that this tree interconnection topology of the processors is the best method for our design as it decreases the communication overhead, reduces the number of links required to interconnect all the processors, and provides an easy way to interact with the hardware.

Tasks in Level-0 Processor (Root)

The tasks assigned to the root processor, along with the computational load, are given in Table 1. The tasks to be executed in the root processor are detailed below.

Table 1 — Tasks and Computational Load of Level-0 Processor

Task	Computational Load in Machine Cycles (M/c)
Constellation generation, visibility check	6,500,000
Iono, tropospheric delay calculation	864
Receiver earth-centered, earth-fixed (ECEF) coordinates and north-east-up (NEU) vector calculation	15,625
Host and level-1 processor communication	1,500
Channel allocation	1,000
Satellite vehicle (SV) clock error calculation	48
Total	6,519,037

The total time needed for the root processor under worst-case conditions is 6,519,037/40 MHz = 0.163 s with a processor speed of 40 MHz.

Satellite Vehicle (SV) Constellation Generation and Visibility Check

Satellite vehicle (SV) constellation generation [7, 8] is the task in which the earth-centered, earth-fixed (ECEF) position of all 32 satellites is determined for the given receiver position, time, and ephemeris data. The visible satellite list is obtained by calculating user ECEF position, user north–east–up (NEU) unit vectors, the up vector between the user and the satellite, and the satellite elevation angle. The NEU local frame is defined by taking receiver position as the origin.

The simulator has a capability to enter the user-required mask angle, which could be either a positive or negative value. The up vector is calculated between the receiver and the satellite in the NEU coordinate system. The satellite is said to be visible if and only if the elevation angle is greater than the mask angle (typically +5 deg).

Ionospheric and Tropospheric Delay Calculations

As a result of the propagation of the GPS signal through the ionosphere and troposphere [7, 8], delays are introduced in the received signal, because the electromagnetic (EM) wave sees a different path length when it propagates through the ionosphere. The model used in this work is that of Klobuchar [7], which is the cosine approximation of the ionospheric delay.

Troposphere is a nondispersive medium for radio waves up to 15 MHz, and the delay introduced by this medium is denoted as the tropospheric delay. The model used here is the Saastamoinen model [7].

SV Clock Error Calculation

Although the SV clock placed is an atomic clock, this clock is affected by a small error. The SV clock error is calculated using the SV clock correction parameters present in the ephemeris data.

The update rate at each level depends on processor speed, processor computational load, and the accuracy of the position fix in the receiver.

If the receiver position is static, the ECEF and NEU unit vectors of the receiver remain constant throughout the simulation. Thus for the static case, it is enough to update the constellation every 10 s.

If the receiver position is dynamic, i.e., if the receiver is moving with certain velocity or with acceleration and jerk, receiver position must be updated more frequently than in the static case. It has been found by analysis that an update period of 1 s is needed for the dynamics specified in this design.

Each level-1 processor handles 4 channels. The root processor sends the details of 4 channels to each level-1 processor, along with receiver parameters and delays, every 1 s.

Tasks in Level-1 Processors

The tasks and the computational load of level-1 processors are given in Table 2. There are three level-1 processors, each controlling 4 channels. The complex computations required for signal generation are performed in these processors. The tasks to be executed in a level-1 processor are described below.

Satellite Relative Velocity and Carrier Doppler Frequency Calculation

The range-rate computation requires calculation of the range between the satellite and the receiver, which is calculated from the satellite position and receiver position. Knowing the satellite position and receiver position at two different times, the relative velocity between receiver and satellite is computed as

$$\text{Relative velocity } (V_{rel}) = (R2 - R1)/(T2 - T1) \quad (4)$$

where R2 is the range between receiver and satellite at time T2, and R1 is the range between receiver and satellite at time T1. Using V_{rel}, the carrier Doppler frequency (Fd) is calculated as follows:

$$Fd = (V_{rel}/C) \cdot F_{L1} \quad (5)$$

where F_{L1} is the L1 frequency (1575.42 MHz), and C is the velocity of light in m/s.

Table 2—Maximum Computational Load of Level-1 Processor

Module	M/c Needed
Receiver position updating	12,102
Satellite relative velocity calculation	111,992
Frequency and change in frequency calculation	80
Signal power calculation	122
Code Doppler and chip period calculation	50
Approximate total for a single satellite	124,346

Each level-1 processor has to handle 4 channels. Total computational requirement = 124,346 · 4 ≈ 498,000 M/c. For a 40 MHz processor, computational time = 498,000 × 10^6 ≈ 12.45 ms.

Frequency and Change in Frequency Calculation

The total IF (F_t) is calculated by eliminating Doppler frequency:

$$\text{Total frequency } F_t = F_{IF} - Fd \quad (6)$$

where F_{IF} is the IF (4.3046 MHz). F_t is up converted to RF in the hardware.

The change in frequency for 19.55 μs is calculated as

$$F_{change} = ((Acc * F_{L1})/C) \cdot 19.55\ \mu s \quad (7)$$

where Acc is the change in relative velocity (acceleration) between receiver and satellite. Acc is computed by knowing the relative velocity betwen receiver and satellite at two different times:

$$Acc = (V_{rel(T1)} - V_{rel(T2)})/T2 - T1 \quad (8)$$

Code Doppler Calculation

The code Doppler shift is calculated as is done for carrier Doppler frequency:

$$Fcd = V_{rel} \cdot Fc/C \quad (9)$$

where F_c is the PRN code chipping rate (1.023 Mchip/s).

Signal Power Calculation

The signal power of all 32 satellites is set by the user at the start of the simulation (S0 [in dBm]) and will be varied as the simulation progresses.

$$S1 \text{ (in dBm)} = S0 + 10 \log (R0^2/R1^2) \quad (10)$$

S0 and S1 are the powers of the GPS signal at times t and t1, and R0 and R1 are the range between receiver and satellite at the times t and t1, respectively. S0 is assigned a value input by the user at the start of the simulation.

Receiver Position Updating

If the receiver is dynamic, receiver position has to be updated periodically. Here the receiver position is updated in ECEF [7] coordinates every 20 ms.

Signal Transmission Time Calculation

From the given receiver time and the calculated delays, the signal transmission time is calculated as

$$T_{trans} = t - t_{IONO} - t_{TROPO} - t_{PD} + t_{svclock} \quad (11)$$

where t is the receiver local time (in GPS seconds), t_{IONO} is ionospheric delay, t_{TROPO} is tropospheric delay, d(t) is the range between receiver and satellite at time t, t_{PD} is the propagation delay between receiver and satellite at time t (d(t)/C), and $t_{svclock}$ is the SV clock error correction.

In the above equation, however, relativistic errors [7] are considered negligible. The level-1 processor sends the calculated data (T_{trans}, F_t, and F_{change}) of 2 channels to each level-2 processor every 20 ms.

As per the computational load given in Table 2, the frequency word calculation requires a total computational time of 12.5 ms. Hence the update period of level-1 processors was fixed at 20 ms.

Tasks in Level-2 Processors

All level-2 processors store 32 satellite PRN codes in memory in a look-up table format (a single satellite requires 1023 locations) and navigation data with a length of 1 superframe [9]. Navigation data is generated by the host processor and provided to all the level-2 processors before the start of the simulation. After the existing navigation data (12.5 min) has been placed in memory, the level-2 processor updates it in a background process. Each level-2 processor handles 2 channels.

The update rate is chosen as 19.55 μs because the scheme is implemented in such a way that a level-2 processor downloads 20 PRN chips to the hardware at a time (20 * single PRN code chip duration ≈ 19.55 μs, where the single PRN code chip duration is 1/1.023 MHz); also, as per the above calculations, the frequency must be updated every 24 μs. Since frequency and phase are downloaded to the hardware at the same time, 19.55 μs (minimum of the above two) is chosen as the update rate. The tasks assigned to level-2 processor are described below.

Frequency Updating

F_{change} is used here to update the frequency of the signal to be transmitted every 19.55 μs:

$$F_{transmit} = F_t + F_{change} \quad (12)$$

The new F_t and F_{change} come from a level-1 processor every 20 ms.

Calculating the Starting Phase of the Signal

From the signal reception time, the level-2 processor calculates the starting phase of the signal for the assigned satellites by calculating the number of superframes, subframes, bits, epochs, and chips elapsed in the transmission time. It also finds the starting chip and the navigation bit to be transmitted.

Chip Period Calculation

The chip period gives the duration of the chip in samples:

$$\text{Chip period} = (\text{master clock frequency of the hardware board})/(Fc + Fcd) \quad (13)$$

The master clock frequency is selected as 29.667 MHz, which results in an integer chip period of 29 samples (with no Fcd). But in real scenarios, Fcd exists, and this results in fractional chip periods. Since hardware can accept only an integer number of samples, fractional chip period is accumulated to obtain an integer number of samples and added or subtracted from the original 29 samples, depending on positive or negative Fcd.

The F_t, signal power, 20 PRN chips, and chip period, which are used for calculating the frequency, phase, and amplitude of the signal, respectively, are transmitted by the level-2 processor to the hardware periodically with a period of 19.55 μs. At the start of the simulation, the level-2 processor downloads the starting chip and the number of samples present in the starting chip. The 20 chips are exlusive-or'ed with the navigation data bit before downloading to the hardware. After 20 chips have been exhausted, the hardware requests from the level-2 processor the next 20 chips. The level-2 processor downloads the next 20 chips, which are transmitted to the hardware, and the sequence continues.

Figure 2 shows the hardware implementation for a single channel. The FPGA collects all the data coming from the level-2 processor and in turn passes the data to the direct digital synthesizer (DDS) for IF signal generation; the IF signal is then up converted to the L1 frequency, as shown in Figure 2. More detail on the implementation of the FPGA and the hardware of the RTGPSSS is presented in [22].

The multipath signal for any of the visible satellites is simulated by delaying the original signal by a fraction of a chip and attenuating the signal. The multipath signal is fed into one of the simulator free channels. The free channels (not allocated to any satellite) can be switched on for multipath effect for any visible satellite. However, the effect of multipath is not investigated here.

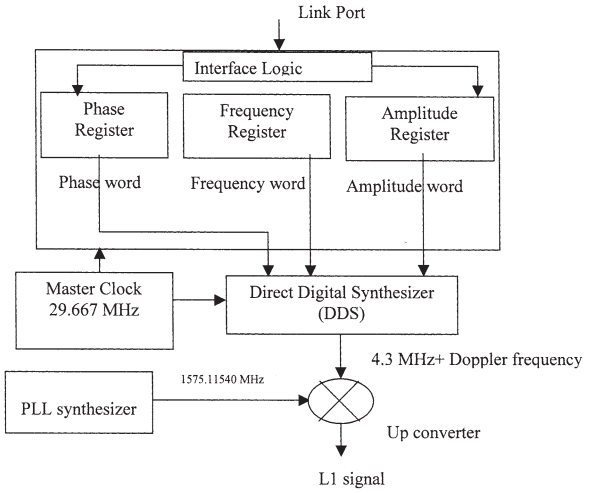

Fig. 2 – Hardware Implementation of RTGPSSS for Single Channel

All the modules at level-0 are running with 64-bit precision to achieve high-precision results. Most of the tasks at level-0 and level-1 are coded in C, and most of the modules at level-2 are coded in assembly language.

PARALLEL ALGORITHM

The tasks carried out by each processor in the parallel processing system are presented in the following algorithm.

Level-0 processor

```
REPEAT for every 1s
BEGIN
    Generate Constellation
    Find Visible Satellites
    Assign a channel for all Visible Satellites
    Compute Ionospheric, Tropospheric,
      Propagation delays and SV clock error
    Send channel details to all 3 Level-1
      processors, and Send data to HOST for
      display
END
END
```

Level-1 processors

```
Do FOR ALL Processors (P1, P2, and P3)
  BEGIN
    REPEAT for every 1 s
      BEGIN
      Get 4-channel details from root processor
      END
```

```
  END
   REPEAT for every 20 ms
    BEGIN
     FOR ALL enabled channels
      Compute V_rel and Acc
      Compute F_t and F_change
      Send details of 2 channels to each level-2
        processor
     END FOR
    END
END DO
```

Level-2 processors

```
Do FOR ALL Processors (P4, P5, P6, P7, P8 and P9)
  BEGIN
   REPEAT for every 20 ms
    BEGIN
     Get details of 2 channels from level-1
       processor
    END
   REPEAT for every 19.55 μs
    FOR ALL channels
     BEGIN
      Compute Ftransmit
      Send Ft, signal power, 20 PRN chips, and chip
        period to hardware
     END
  END
```

RESULTS AND DISCUSSION

All the modules mentioned above are implemented in the MULTISHARC system, and the results are verified using a GPS receiver development board. The host displays the data calculated by the level-0 processor—visible satellite list, channel allocated to each satellite, and position of each satellite in ECEF (X,Y,Z) coordinates. The host also displays the four best satellites selected using geometric dilution of precision (GDOP).

The L1 signal (1575.42 MHz) generated by the simulator is interfaced with the GPS receiver development board, incorporating low dynamics in the receiver position. The receiver is able to acquire, track, and demodulate the navigation data bits present in the signal. The receiver identifies and extracts the almanac and ephemeris data present in the navigation data. A sample of the test results for the receiver for a signal simulated for a stationary case is given in Figure 3.

The signal is simulated for the position 10 deg latitude, 80 deg longitude, and 1000 m altitude. From Figure 3, it can be observed that the receiver was able to give the three-dimensional position fix using four satellites. The corresponding speed, heading, horizontal dilution of precision (HDOP), and positional dilution of precision (PDOP) values are displayed. It can also be seen that 17, 05, 10, and 09

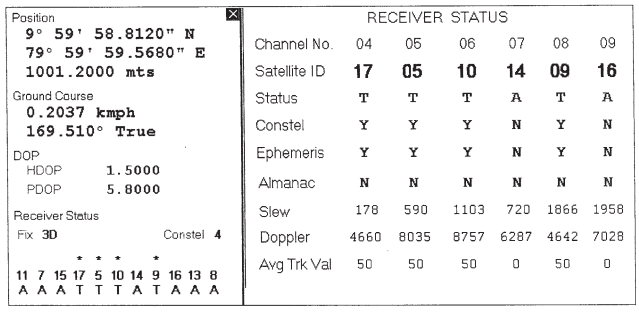

Fig. 3–Receiver Output

(satellite IDs) are the best four satellites selected by the GPS receiver to give the position fix, being tracked in channels 04, 05, 06, and 08, respectively. The corresponding status, constellation, ephemeris, almanac, slew, Doppler, and average track value are also shown in this figure.

Figure 4 shows a Gantt chart that presents the processor utilization with respect to time for all processors at each level. It is found that the processor utilization of each level-2 processor is 80 percent, that of each level-1 processor is 62 percent, and that of the level-0 processor is 16 percent.

CONCLUSION

The design of an RTGPSSS using a DSP-based multiprocessing system has been presented in this paper. The multiprocessing system generates phase word, amplitude word, and frequency word. These words are given to the FPGA-based hardware unit for generating the real-time GPS signal. The computational complexity of the GPS modules, the load on each processor, processor connectivity for efficient parallel implementation, and the rates at which the frequency, phase, and amplitude of the signal are to be downloaded to the FPGA have been studied and verified.

The RTGPSSS generates different platform motion profiles, stores them for repeated use, and calculates

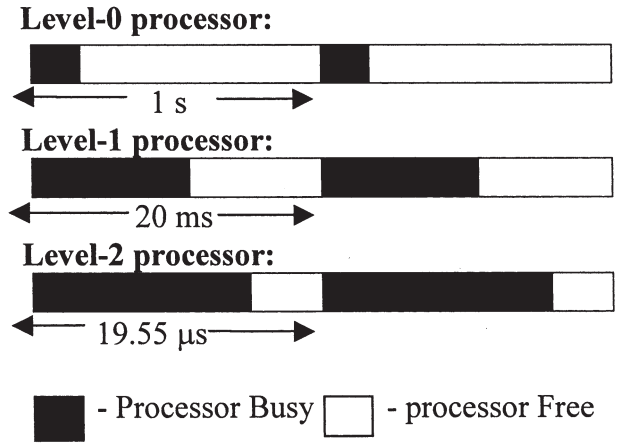

Fig. 4–Gantt Chart

the frequency of the signal every 19.55 μs. The overall efficiency of the parallel scheme is 66 percent, and the pseudorate error is 0.04 m/s^2. In this design, the user can select any position and time and generate the corresponding visible satellite signals. The designed simulator has been tested for the static and simple dynamic receiver configuration.

ACKNOWLEDGMENT

The authors acknowledge the Aeronautical Research and Development Board (AR&DB), Ministry of Information and Communication Technology (Department of Electronics), New Delhi, and Signals & Systems (SANDS), Chennai, for sponsoring this project. The authors also acknowledge the help given by Mr. Senthil Kumar of the Electronics Department.

REFERENCES

1. Getting, I. A. *The Global Positioning System*, IEEE Spectrum, December 1993.
2. Chandha, K., *The Global Positioning System: Challenges in Bringing GPS to Mainstream Consumers*, SiRF Technology, Inc., Santa Clara, CA, 1998.
3. Enge., P., and P. Misra, *Scanning the Special Issue/ Technology on Global Positioning System*, Proceedings of IEEE, Vol. 87, January 1999, pp. 3–15.
4. Lewandowski, W., J. Azoubib, and W. J. Klepczynski, *GPS: Primary Tool for Time Transfer*, Proceedings of IEEE, Vol. 87, No.1, January 1999, pp. 163–171.
5. Herring, T. A., *Geodetic Applications of GPS*, Proceedings of IEEE, Vol. 87, No. 1, January 1999, pp. 92–109.
6. Abbott, E., and D. Powell, *Land-Vehicle Navigation Using GPS*, Proceedings of IEEE, Vol. 87, No. 1, January 1999, pp. 144–161.
7. Hofmann-Wellnhof, B., H. Lichtenegger and J. Collins, *Global Positioning System: Theory and Practice*, Fourth edition, Springer-Verlag Wien, New York, 1997.
8. Parkinson, B. W., and J. J. Spilker Jr., *Global Positioning System: Theory and Applications*, Volumes 1 & 2, American Institute of Aeronautics and Astronautics, 1996.
9. Krishnan, C. N., P. V. Ramakrishana, and V. Vaidehi, *Real Time GPS Signal Simulator Development*, Interim Technical Report and Documentation, Electronics Engineering Department, Madras Institute of Technology, April 1999.
10. NAVSTAR, *GPS Constellation Simulator SCS2400* (Online).
11. NovATel's GPS Timing Simulator—Document from web.
12. BFGoodRich Aerospace, *GNS 743A GPS/GLONASS Satellite Simulator*, JcAIR Systems, Charlotte, NC, 1998.
13. Sprint Communications, *Low Cost Multi-channel GPS/SBAS Simulation System STR4500*, GSS, (Online).
14. Brown, A., N. Gerein, and K. Taylor, *Modeling and Simulation of GPS Using Software Signal Generation and Digital Signal Reconstruction*, Proceedings of the ION National Technical Meeting, Anaheim, CA, January 2000.
15. Brown, A., and N. Gerein, *Advanced GPS Hybrid Simulator Architecture*, Proceedings of the ION National Technical Meeting, Anaheim, CA, January 2000.
16. Papenfuss, J. R., and M. A. Wickert, *A TMS320C6701/ FPGA Based Frequency Selective RF Channel Simulator Using IF Sampling*, TI DSPS Fest, Houston, TX, August 2000.
17. Wickert, M. A., and J. Papenfuss, *Implementation of Real-Time Frequency Selective RF Channel Simulator Using a Hybrid DSP-FPGA Architecture*, IEEE Trans. on Microwave Theory and Techniques, Vol. 49, No. 8, August 2001, pp. 1390–1396.
18. Schweber, B., *RF Channel Simulators, Bring Reality's Challenges to Your Prototype*, EDN, September 11, 1998, pp. 52–64.
19. Analog Devices, *ADSP-2106x SHARC User Manual*, 2nd edition, 1997.
20. Krishnan, C. N., P. V. Ramakrishna, V. Vaidehi, G. Boopalan, S. V. M. K. Prasad, G. Ramya, U. Chandra, J. Gnana Prakash, T. A. Sarangaram, N. Kumar, and R. Ashok raj, *Development of Real-Time GPS Signal Simulator,* Proceedings of the Workshop on GPS Modernization and Augmentation Systems, Osmania University, Hyderabad, February 2001, pp. 129–136.
21. Prasad, S. V. M. K., T. A. Sarangaram, G. Boopalan, V. Vaidehi, C. N. Krishnan, and P. V. Ramakrishna, *Real-Time GPS Satellite Signal Simulator: Software Design on a Multi SHARC Parallel Processing System*, Proc. of NCC, IITB, Mumbai, 2002, pp. 553–557.
22. Boopalan, G., S. V. M. K. Prasad, V. Vaidehi, C. N. Krishnan, P. V. Ramakrishna, and L. R. Rajagopal, *Hardware Design for GPS Signal Simulators*, Proc. of NCC, IIT Bombay, Mumbai, 2002, pp. 84–69.
23. Spectrum Signal Processing, *Apex-Lite, Apex-Debug, Apex-Trace, Darlington SHARC3000 PCI Motherboard Manuals*, Revision 2.0, Burnaby, B. C., Canada July 1998.