JPNT 3(3), 99-105 (2014)
http://dx.doi.org/10.11003/JPNT.2014.3.3.099

JPNT Journal of Positioning, Navigation, and Timing

Software-based Real-time GNSS Signal Generation and Processing Using a Graphic Processing Unit (GPU)

Sung-Hyuck Im[1], Gyu-In Jee[2†]

[1]Korea Aerospace Research Institute, Daejeon 305-806, Korea
[2]Department of Electronics Engineering, Konkuk University, Seoul 143-701, Korea

ABSTRACT

A graphic processing unit (GPU) can perform the same calculation on multiple data (SIMD: single instruction multiple data) using hundreds of to thousands of special purpose processors for graphic processing. Thus, high efficiency is expected when GPU is used for the generation and correlation of satellite navigation signals, which perform generation and processing by applying the same calculation procedure to tens of millions of discrete signal samples per second. In this study, the structure of a GPU-based GNSS simulator for the generation and processing of satellite navigation signals was designed, developed, and verified. To verify the developed satellite navigation signal generator, generated signals were applied to the OEM-V3 receiver of Novatel Inc., and the measured values were examined. To verify the satellite navigation signal processor, the performance was examined by collecting and processing actual GNSS intermediate frequency signals. The results of the verification indicated that satellite navigation signals could be generated and processed in real time using two GPUs.

Keywords: GPU, software defined radio, GNSS, CUDA

1. INTRODUCTION

As for the current satellite navigation signals, new satellite navigation systems such as Galileo, Beidou, Indian Regional Navigation Satellite System (IRNSS), and Quai-Zenith Satellite System (QZSS) have been developed and operated, along with the modernization of Global Positioning System (GPS) and GLObal NAvigation Satellite System (GLONASS). Accordingly, with the purpose of the research and development of new satellite navigation signals, specific hardware processing chips need to be newly developed and applied for the generation and processing of simulated signals, and thus, the required time becomes longer. On the other hand, software-based signal generation and processing can be developed using general programming language such as C/C++, and the generation and processing of new signals are possible within several months. However, a general purpose processor has a limitation in processing capacity, and thus, for real-time generation and processing of multiple satellite navigation signals, a special purpose processor, which has superior parallel processing performance compared to the existing general purpose processor, is required. A technique that is recently in the spotlight considering the compatibility with a general purpose processor is the parallel processing technique using a graphic processing unit (GPU). GPU includes hundreds of to thousands of special purpose processors for graphic processing, and aims to carry out programming so that the same function can be performed on multiple data. As the correlation and generation of satellite navigation signals involve the same calculation procedure for multiple sample and generation data, the use of GPU could maximize the efficiency.

In this study, the structure of a GPU-based Global Navigation Satellite System (GNSS) simulator for the generation and processing of satellite navigation signals was designed, developed, and verified. For the RF band conversion and sample of satellite signals, the vector signal generator and analyzer of National Instrument Corporation were used. For GPU, Tesla-K20 of Nvidia Corporation with

Received May 09, 2014 **Revised** July 28, 2014 **Accepted** July 28, 2014
†**Corresponding Author**
E-mail: gijee@konkuk.ac.kr
Tel: +82-2-450-3070 Fax: +82-2-3437-5235

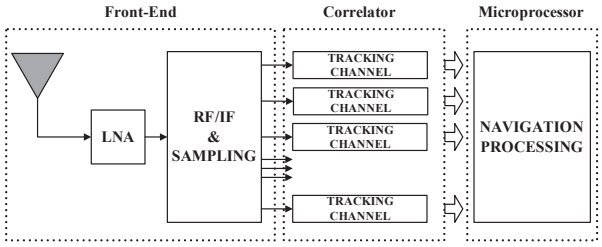

Fig. 1. Structure of a general GNSS receiver.

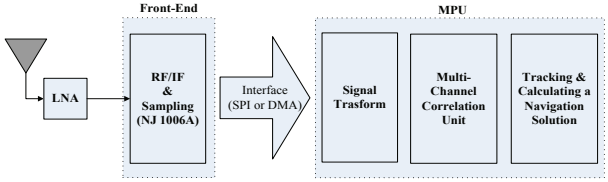

Fig. 2. Structure of a general software-based GNSS receiver.

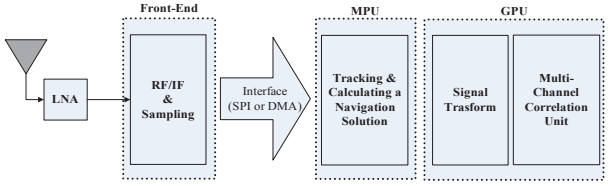

Fig. 3. Structure of a software-based GNSS receiver using GPU.

a Kepler architecture was used. A workstation was used for the control of the entire system. To verify the developed satellite navigation signal generator, generated signals were applied to the OEM-V3 receiver of Novatel Inc., and the measured values were examined. To verify the satellite navigation signal processor, the navigation solution performance was examined. The results of the verification indicated that satellite navigation signals could be generated and processed in real time using two GPUs.

2. STRUCTURE OF A SOFTWARE-BASED GNSS RECEIVER

2.1 Structure of a general GNSS receiver

A general hardware-based GNSS receiver consists of front-end, analog to digital (A/D) converter, correlator, and microprocessor (Kaplan & Hegarty 2005). Fig. 1 shows the structure of a general hardware-based GNSS receiver. A GNSS signal is received by an antenna, passes through a low noise amplifier, is converted to an intermediate frequency signal, and goes through A/D conversion. Then, it is entered into each correlator, and a measured value is generated through signal acquisition and tracking. Using the measured value generated by the correlator, the microprocessor calculates a position by performing a navigation algorithm. In the case of a hardware-based receiver, to process a new signal structure, it is essential to replace the correlator that performs the demodulation of signals (Im et al. 2007).

2.2 Structure of a general software-based GNSS receiver

Fig. 2 shows the structure of a general software-based satellite navigation receiver (Akos 1997). Unlike a hardware-based receiver, the correlator part is performed by the microprocessor. Thus, all the processing excluding the RF part (antenna, front-end, and A/D converter) is performed by the microprocessor. Due to its structural limit, a microprocessor has a limitation in processing high-capacity data (tens of millions of samples per second) in real time. Therefore, to resolve this problem, a processor that is capable of real-time parallel processing of high-capacity data is required (Im et al. 2007).

2.3. Software-based GNSS receiver using a GPU

GPU has been used as a processor for graphics. However, it has recently been used for scientific calculation based on the fact that its structure is efficient for parallel calculation. GPU is designed to have a structure where a lot of special purpose processors can be simultaneously operated. Thus, the efficiency can be maximized when the same calculation is performed on multiple data. Fig. 3 shows the structure of a software-based receiver using GPU. In this structure, the conversion and correlation of signals are performed by GPU; and the control of GPU and signal tracking and the calculation of a navigation solution are performed by the microprocessor. GPU can be substituted by field programmable gate array (FPGA). However, GPU has been used in the existing general purpose personal computer, and thus, the interface is easy. Also, for the programming method, development using C or C++ is possible in the existing development tool. In the case of FPGA, VHSIC Hardware Description Language (VHDL) or Verilog is used, and thus, the conversion of an algorithm that has been developed using the existing C or C++ is required. Also, due to the nature of FPGA, there are difficulties in implementing high-complexity calculations, and it is not appropriate for implementing floating point calculations (Thor 1999). In addition, it is essential to understand an entire algorithm (signal processing), and to optimize it for FPGA. Therefore, FPGA has lower flexibility than GPU.

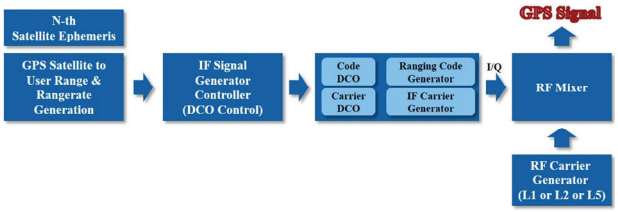

Fig. 4. Structure of single channel GNSS signal generation.

3. STRUCTURE OF A SOFTWARE-BASED GNSS SIGNAL GENERATOR

3.1 Hardware-based GNSS signal generator

Fig. 4 shows the structure of hardware-based single channel satellite navigation signal generation. The geometric relation of a satellite and a receiver is calculated, and code and Doppler are generated by applying an intermediate frequency signal control value. Then, a single satellite signal is generated through mixing with a carrier. For example, for the generation of 12 satellite navigation signals, 12 satellite navigation signals are generated using a 12-channel hardware-based signal generation module, and they are converted to a single output using an RF combiner. A hardware-based satellite navigation signal generator guarantees accuracy and real-timeness, and is easy to use. However, it is difficult to manufacture, is expensive, and requires a long period for development. Also, a commercial hardware-based satellite navigation signal generator has a limitation in performing simulations that are required by a user, due to the limit of provided scenarios.

3.2 Software-based GNSS signal generator

Fig. 5 shows the structure of software-based satellite navigation signal generation. The geometric relations of desired number of satellites and receiver are calculated, and intermediate frequency signals for discrete samples are generated by software. Then, the signals of the entire satellites are generated via mixing. The generated satellite signals are made to have a desired RF band using digital to analog (D/A) conversion and RF up-conversion, which completes the satellite navigation signal generation. As signals for discrete samples are generated, the generation accuracy is determined by the density of the sample interval. The code generation accuracy using the developed software-based satellite navigation signal generator is one over dozens meter and the carrier generation accuracy is one over thousands meter. Thus, it satisfies the performance for a simulation at a commercial receiver level. However, although it has higher flexibility than a hardware-based satellite navigation signal generator, for a simulation at a high-precision receiver level, the code generation accuracy (one over hundreds meter) and carrier generation accuracy (one over thousands ~ one over tens of thousands meter) of a commercial hardware-based signal generator are

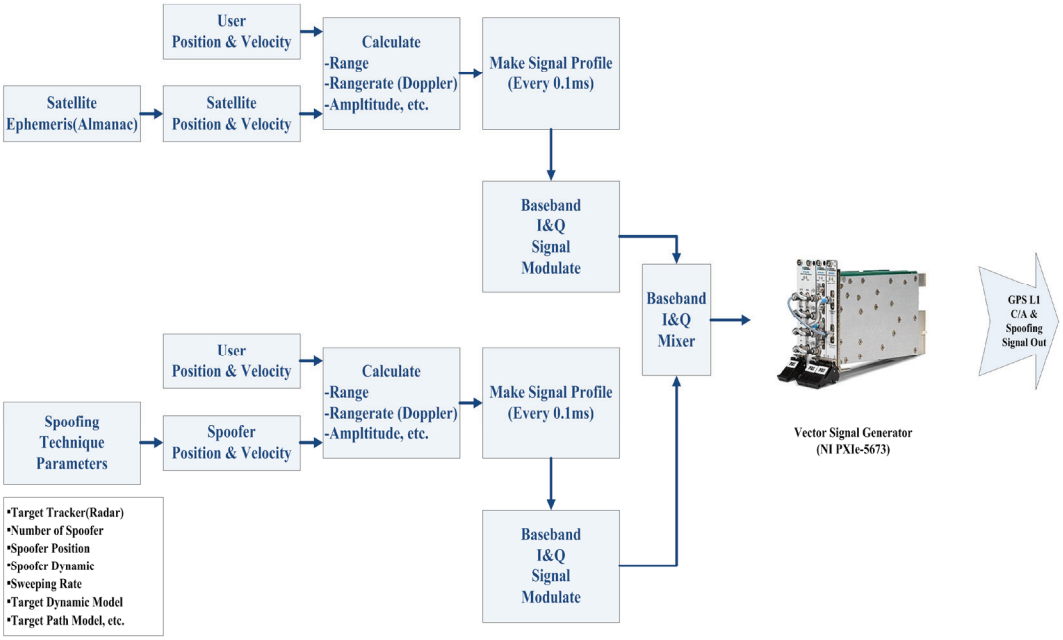

Fig. 5. Structure of a software-based GNSS signal generator.

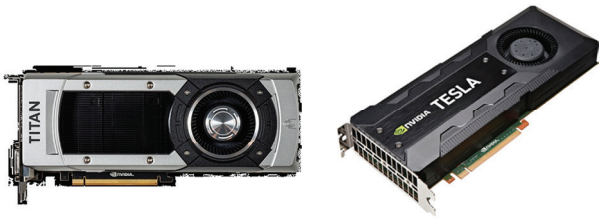

Fig. 6. Graphic processor (left: GeForce, right: TESLA) (Nvidia 2014).

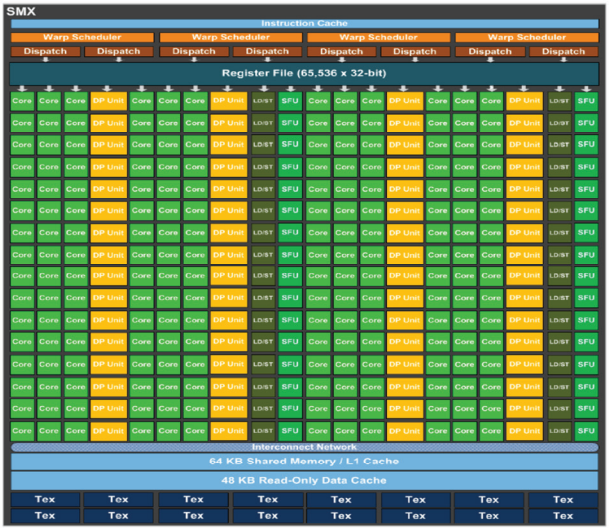

Fig. 7. Structure of a streaming multiprocessor: SMX (Nvidia 2014).

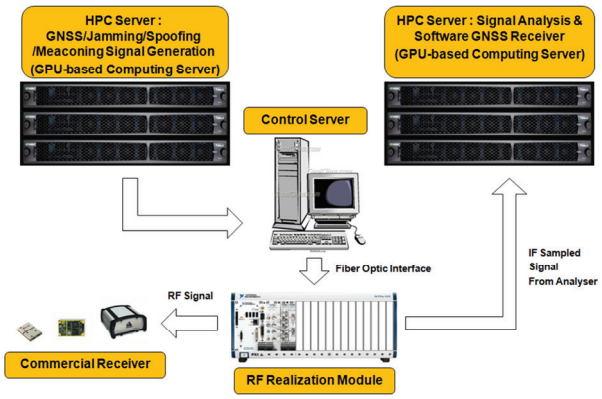

Fig. 8. GPU-based GNSS signal simulator (generation and processing).

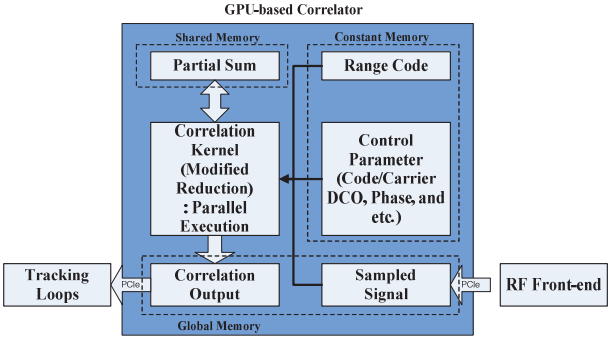

Fig. 9. GPU-based correlator.

required. Therefore, further research is needed.

4. STRUCTURE OF GPU

The GPU used in this study was Tesla-K20 of Nvidia Corporation. For GPU-based programming, OpenCL and DirectX Compute as well as compute unified device architecture (CUDA) can be used. However, for low-level programming, application of the Nvidia product family using CUDA could increase the efficiency of development. As for the difference between the Tesla product family and the GeForce product family, Tesla is GPU for calculation, has limited or no display function, and includes a double precision floating point calculation function; and thus, it has superior double precision floating point calculation performance compared to the GeForce product family.

Fig. 6 shows the GeForce Titan of Nvidia Corporation and the Tesla module for calculation (K20). The K20 module is equipped with 2,496 CUDA cores. Therefore, 2,496 calculations can be performed per one cycle. The differences between a general graphic processor and Tesla-K20 include the double precision floating point calculation function and the adoption of error correcting code (ECC) memory for preventing calculation error due to memory error. For satellite navigation signal processing, the processing performance of the GeForce product family with more cores is generally superior to that of the Tesla product family. However, for satellite navigation signal generation, it is desirable to use the Tesla product family because the use of double precision floating point is frequent.

Tesla-K20 includes 13 streaming multi-processors (SMX) shown in Fig. 7. Each SMX consists of 192 single precision CUDA cores, 64 double precision units, 32 special function units (SFU), and 32 load/store units (LD/ST); is controlled by one instruction cache, four warp schedulers, and eight dispatch modules; and includes 64KB shared memory and 48KB read-only data cache. Therefore, Tesla-K20 includes 2,496 CUDA cores (Nvidia 2014).

5. GNSS SIGNAL GENERATION AND PROCESSING USING GPU

Fig. 8 shows a software-based satellite navigation simulator using GPU. The generation and processing of

http://dx.doi.org/10.11003/JPNT.2014.3.3.099

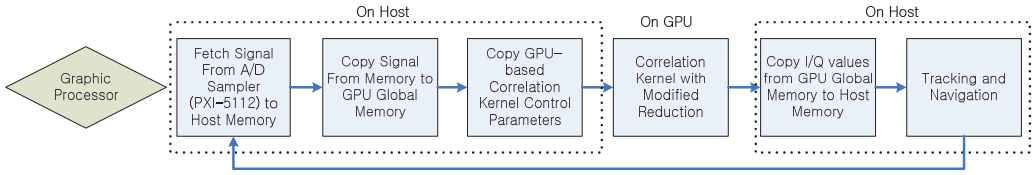

Fig. 10. Flowchart of the signal processing of a GPU-based receiver.

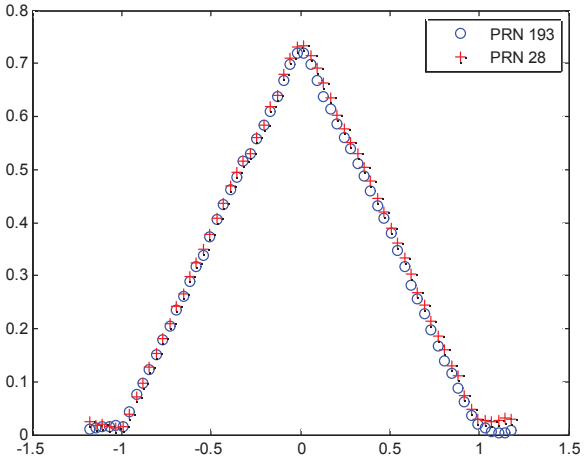

Fig. 11. Correlation result for signal monitoring (64 track-arm).

signals are performed by the GPU module (or server), and the transmission and control of data are performed by the control server (or workstation). For the collection and generation of RF signals, the PXIe-5673 (vector signal generator) and PXIe-5663 (vector signal analyzer) of National Instrument Corporation were used. For low-priced implementation, Universal Software Radio Peripheral (USRP) or Blade-RF module can be used for the collection and generation of RF signals although the accuracy and applicability would slightly decrease.

5.1 GPU-based GNSS signal processing

Fig. 9 shows a GPU-based correlator. Variables that are the most frequently used in the processing of satellite navigation signals are the Doppler values of code and carrier. Thus, if these frequently used variables are included in constant memory (Sanders & Kandrot 2010), and a partial sum technique using shared memory is used for correlation sum (Kirk & Hwu 2010), a real-time software correlator with 64 tracking-arms could be implemented up to about 30 channels. Fig. 10 shows the flowchart of the processing of a satellite navigation receiver using GPU.

Fig. 11 shows the correlation result of GPS PRN 28 satellite and MSAS PRN 193, using the implemented correlator. Fig. 12 shows the comparison between the position solution calculated using the implemented real-time GPU-based software satellite navigation receiver and the position solution obtained by the OEM-V3 of Novatel Inc.

5.2 GPU-based GNSS signal generation

Using the GPU and CUDA explained earlier, a baseband signal generation part, which is part of a signal generator, was implemented. Fig. 13 shows the flowchart of the baseband signal generation for satellite navigation signal generation. It separately shows a case using CPU and a case using GPU. In Fig. 14, the parts that are processed by GPU

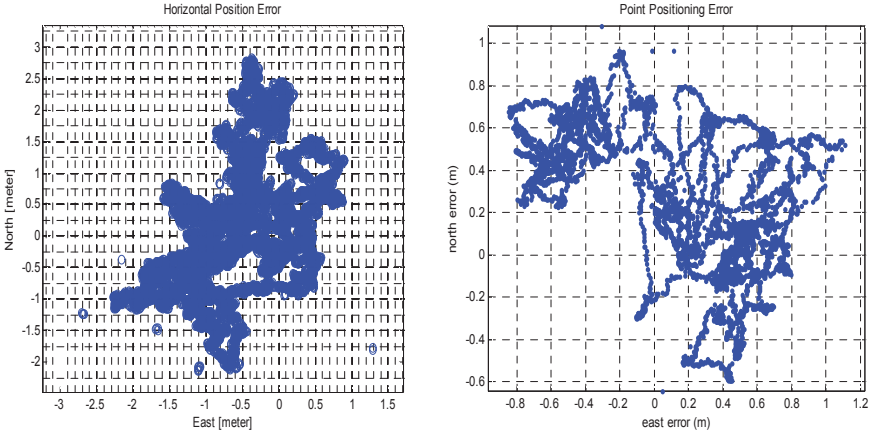

Fig. 12. Horizontal position error (left: software GPS, right: Novatel OEM-V3).

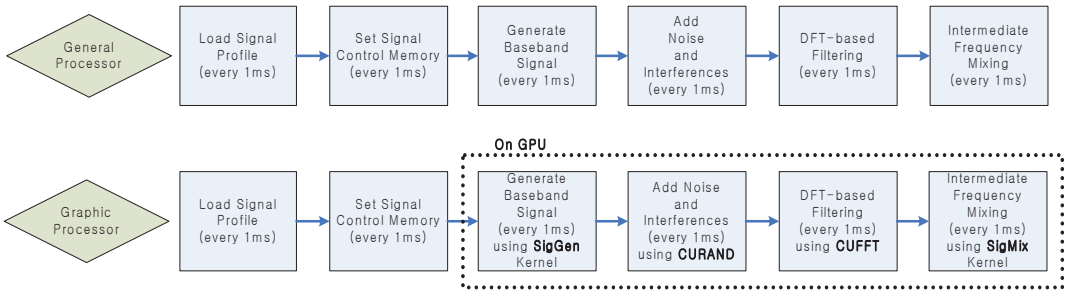

Fig. 13. Flowchart of GPU-based signal generation.

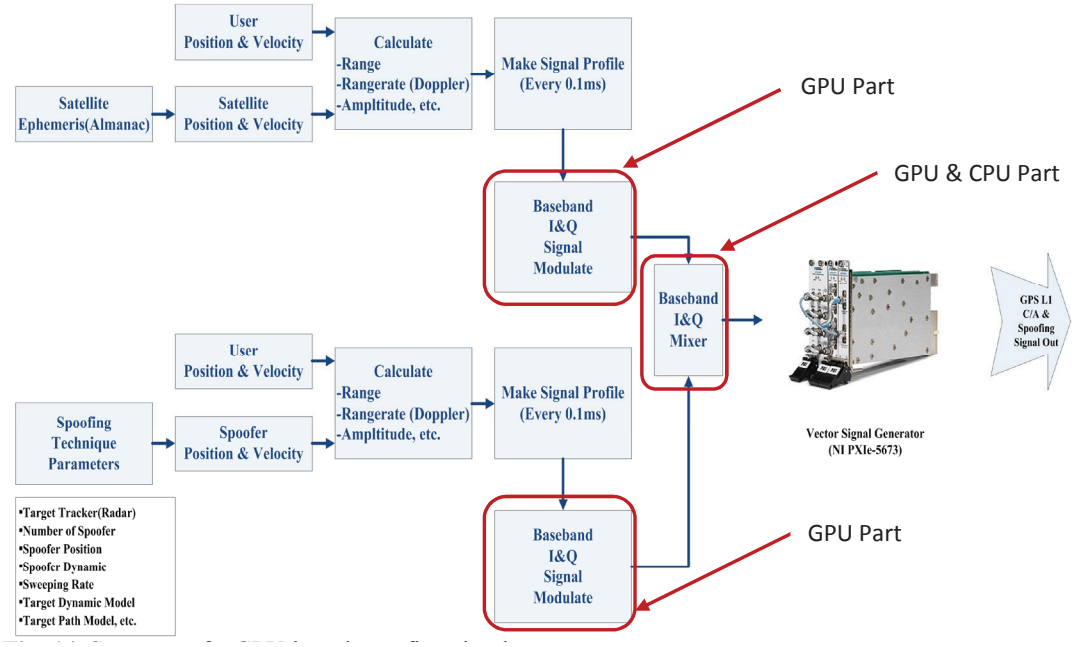

Fig. 14. Structure of a GPU-based spoofing simulator.

in Fig. 5 are shown in rectangles with rounded corners. GPU has low efficiency for complicated calculations such as the calculation of the geometric relation between a satellite and a receiver. Thus, CPU performs this complicated calculation; and based on this, GPU is appropriate for a part that generates signals corresponding to each sample, and a calculation that obtains the sum of each generated signal.

6. CONCLUSION

Currently, software-based satellite navigation signal generation and processing are widely used in satellite navigation signal studies due to the advantage of prompt development. In this study, considering this research and development trend, GPU was introduced, applied, and verified for the generation and processing of high-capacity satellite navigation signals, and the possibility of replacing the existing high-priced satellite navigation simulator was examined. If a hardware-based satellite navigation simulator is used in a satellite navigation signal spoofing study, limitations in generating various spoofing scenarios could be overcome, and a study that unifies real-time signal generation and processing of various spoofing scenarios is enabled.

However, in the case of software-based signal processing, pulse per second (PPS) generation is difficult; and in the case of signal generation, the accuracy is limited. Therefore, further studies for overcoming these problems are required.

ACKNOWLEDGMENTS

This research was supported by a grant from Transportation System Innovation Program (TSIP) funded by Ministry of Land, Transport and Maritime Affairs (MLTM) of Korean government.

REFERENCES

Akos, D. M. 1997, A software Radio Approach to Global Navigation Satellite System Receiver Design, PhD Dissertation, Ohio University

Im, S., Jee, G., Cho, S., & Ko, S. 2007, A novel software GPS receiver architecture using partial down-conversion, Proceedings of the 2007 National Technical Meeting of The Institute of Navigation, January 22-24, 2007, The Catamaran Resort Hotel, San Diego, CA, pp.702-707

Kaplan, E. D. & Hegarty, C. J. 2005 Understanding GPS: Principles and Applications. 2nd ed. (Norwood, MA: Artech House Publisher)

Kirk, D. B. & Hwu, W. W. 2010, Programming Massively Parallel Processors: A Hands-on Approach (Burlington, MA: Elsevier and Morgan Kaufmann)

Nvidia KEPLER Architecture, cited 2014 Jul 1, available from: http://www.nvidia.com/object/nvidia-kepler.html

Sanders, J. & Kandrot, E. 2010, CUDA by Example: An Introduction to General-Purpose GPU Programming (Boston, MA: Addison-Wesley)

Thor, J. 1999, Evaluation of a Reconfigurable Computing Engine for Digital Communication Applications, Master's Thesis, Lulea University

Sung-Hyuck Im is a senior researcher in the KARI(Korea aerospace research institute). He received Ph.D. degree from Konkuk University in 2011. He is interested in (Real-time) Software GNSS receiver, Generation and processing of navigation signals, Vector-based signal processing, Anti-Jamming/ Spoofing, Indoor positioning, Navigation sensor integration, etc.

Gyu-In Jee is a professor in the department of Electronics Engineering at Konkuk University in Seoul, Korea, since 1992. He received his Ph.D. in Systems Engineering from Case Western Reserve University in 1989. His research has been focused on GNSS, autonomous vehicle, and navigation system. He has worked on several research and development project: Autonomous ground vehicle system implementation, Indoor positioning, Software GNSS receiver, IEEE 802.16e based wireless location system, precise GNSS system, etc.