A real-time GPS software receiver running on the GPU

Thomas Hobiger[1], Tadahiro Gotoh[1], Jun Amagai[1], Tetsuro Kondo[1,2] and Yasuhiro Koyama[1]

(1) Space-Time Standards Group, National Institute of Information and Communications Technology, Japan

(2) Ajou University, Suwon, Korea

Contact: hobiger@nict.go.jp

GPU TECHNOLOGY CONFERENCE

Summary

Global positioning service (GPS) software radios have become an attractive and cheap alternative to hardware receivers. Such software implementations can be easily adopted to handle new signals and allow for rapid testing of new algorithms. Although modern CPUs are more and more evolving to multi-core processors, the performance from a single chip is usually not sufficient to fulfill the needs of real-time GPS tracking. On the other hand, Graphic Processing Units (GPUs) have undergone a tremendous development in the recent years and deployed electronic components have become powerful parallel systems which can be utilized for a variety of applications. We have utilized this parallel computing power and implemented a real-time GPS software receiver running on an off-the-shelf graphics board, using digitized GPS IF signals.

Hard- and software components

Software receivers require that the RF signals are down-converted and digitized by hardware components before they can be processed on the PC. Moreover, since the software developed in this study is not only dedicated to GPS but also useable for time-transfer applications using PRN-code like signals, flexible and robust hardware parts have been deployed. Several components which have been originally developed for radio astronomy are used in addition to other off-the-shelf components. After A/D conversion the data stream is sent to the PC via USB. Ring-buffers, realized in host-memory, are able to hold these data, before being copied to the GPU.

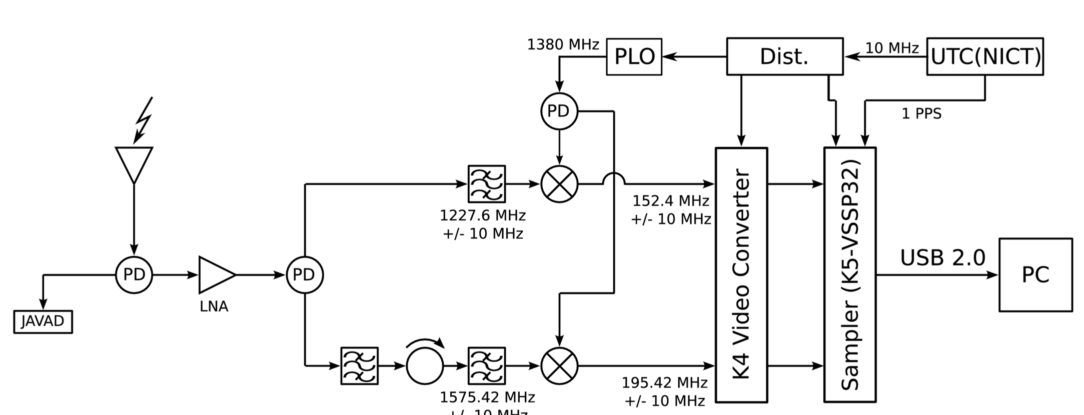

***Figure 1**: Schematics of the hardware components which are used for down conversion and digitization of the GPS signals (abbreviations: power divider (PD), phase looked oscillator (PLO), frequency distributor (Dist.), low noise amplifier (LNA)).*

	CPU	GPU
	Intel Core 2 / Q9450	NVIDIA GeForce GTX 280
Cores	4	240
Clock	2660 MHz	1296 MHz
Memory	4 GB	1 GB
Compiler	gcc	NVCC 2.3 (2.2)
OS	Fedora 10 (64bit)	

***Table 1**: Utilized soft- and hardware.*

Choices for the correlation engines

Conventional hardware-receivers as well as several software radios implement the "classical" early/prompt/late scheme for the code tracking loop. As for our solution we have decided to follow another strategy, computing a wider and finer correlation function, which has several advantages concerning multi-path mitigation and tracking of weak signals. For the realization of such an approach two different strategies exist.

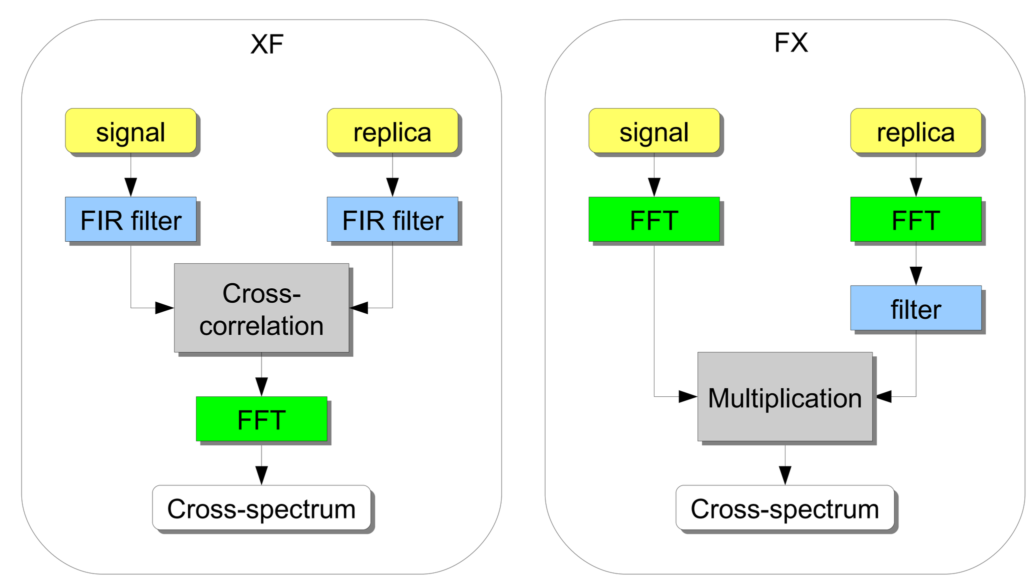

***Figure 2**: XF(left) vs. FX(right) approach.*

Performance of XF and FX architectures

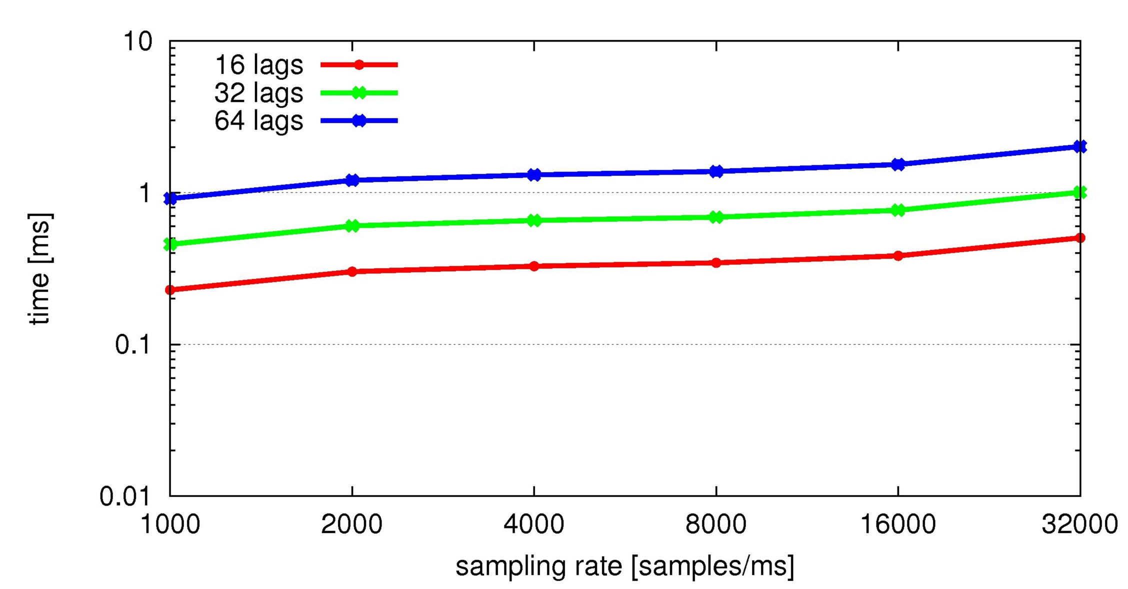

***Figure 3**: Performance of a single channel XF cross-correlation implementation on the GPU using different lag and data sizes. Measurements represent mean values from 100,000 runs and do not include data-transfer between CPU and GPU.*

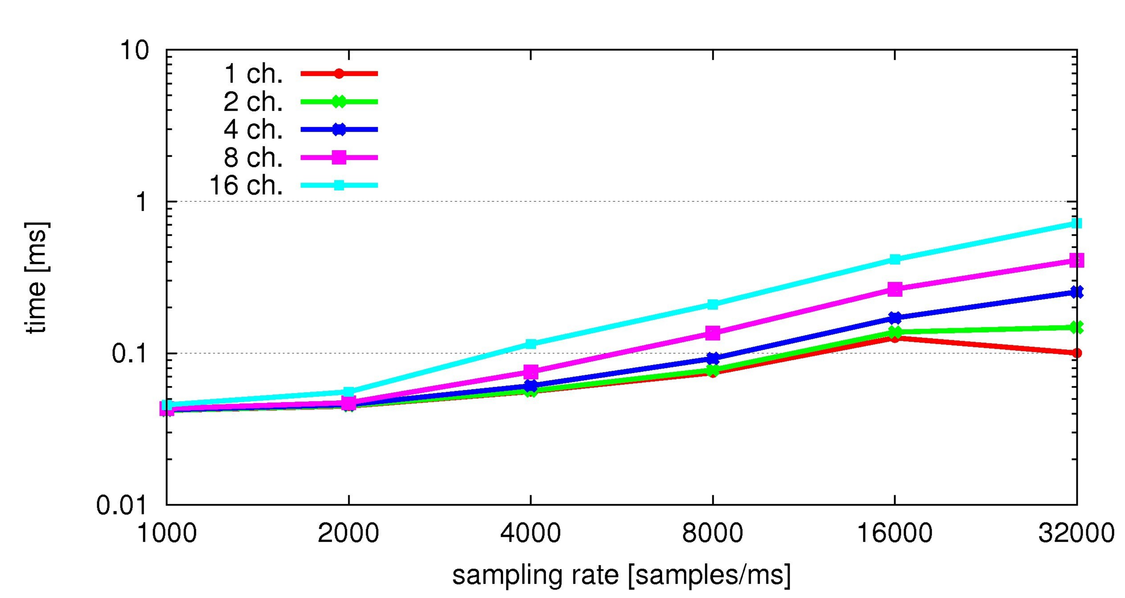

***Figure 4**: Parallel FX performance for a different number of channels and varying data sizes.*

Conclusion:

Only FX type allows multi-channel real-time operation on the GPU

GPS receiver architecture on the GPU

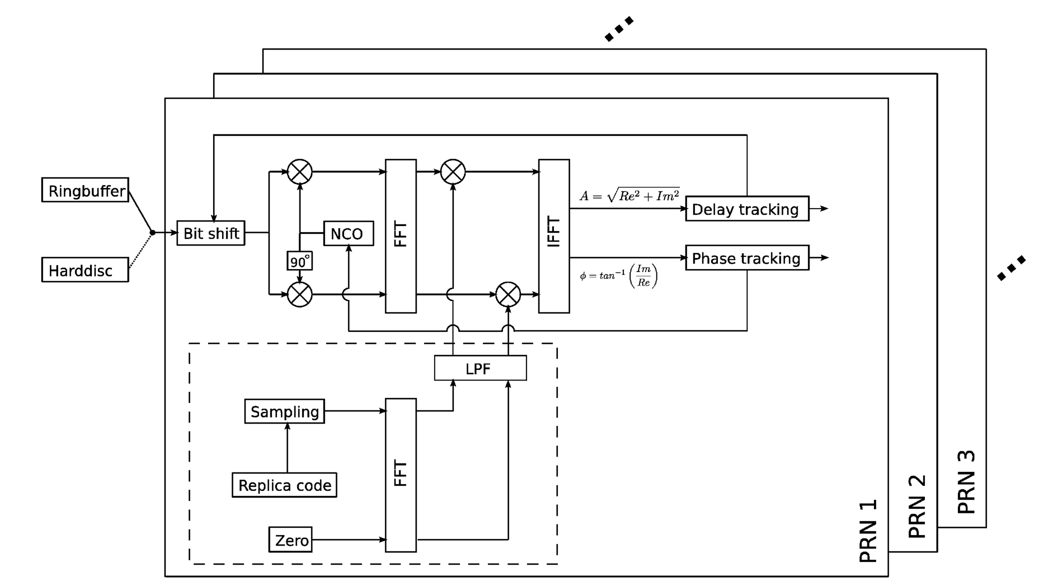

***Figure 5**: Schematics of the multi-channel real-time software receiver, running on the GPU. The numerically controlled oscillator (NCO) is updated via the Doppler tracking loop, ensuring a continuous tracking of the carrier phase. Delay tracking is performed via proper variation of the read-pointer which feeds the FX engine with data.*

Real-time performance of the receiver

$$\text{Processing factor } \kappa = \frac{\text{Processing time}}{\text{Time-span of data}}$$

(Real-time criteria: $\kappa < 1$)

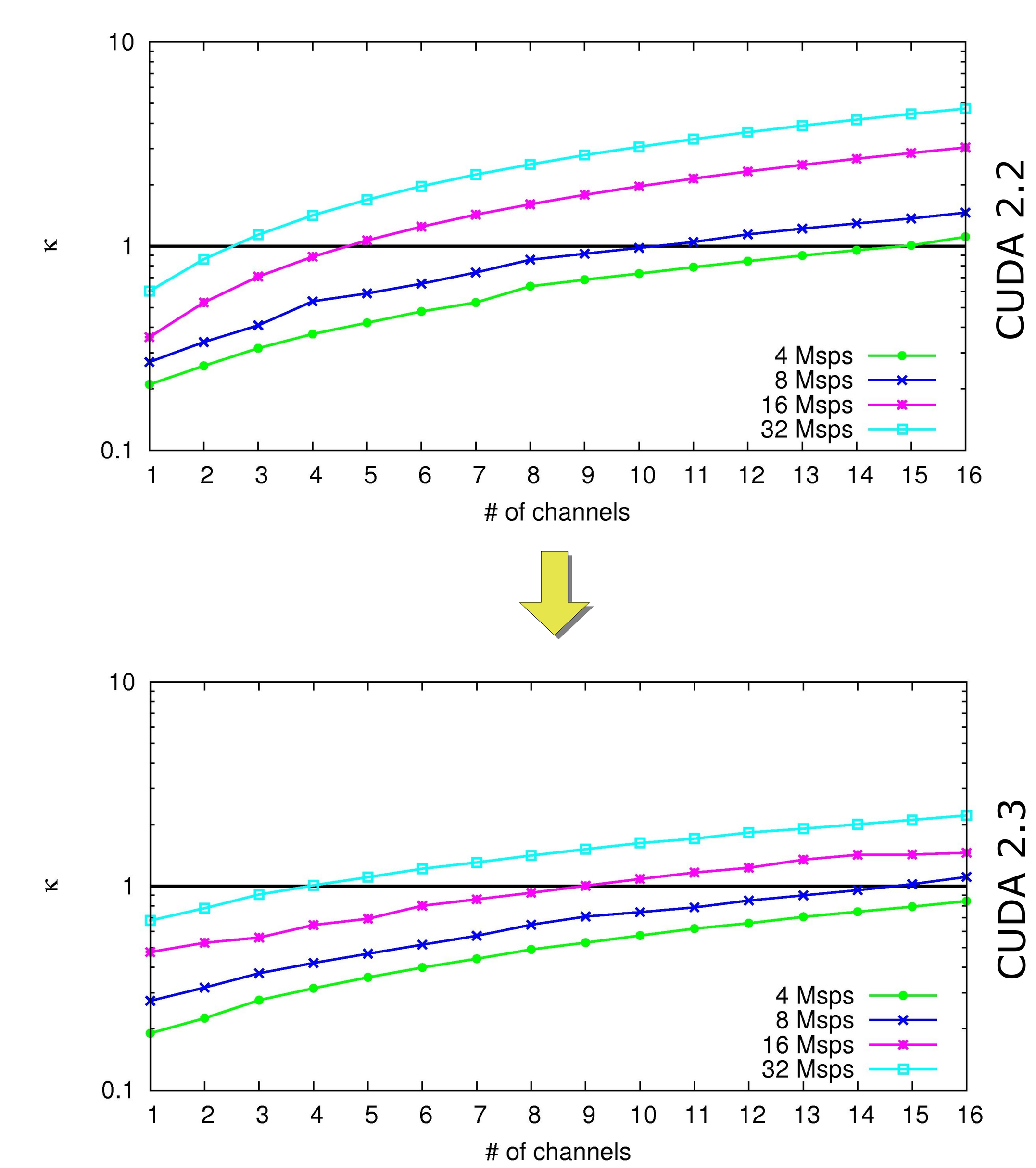

***Figure 6**: Processing factors for the receiver in dependency of the number of parallel channels for different sampling rates. The solid line represents the real-time criteria. The upper plot shows results based on CUDA 2.2 and the lower plot depicts performance of the receiver when being compiled with CUDA 2.3. Recent improvements of the CUFFT library allow to handle more channels (i.e. satellites) in real-time.*

Results

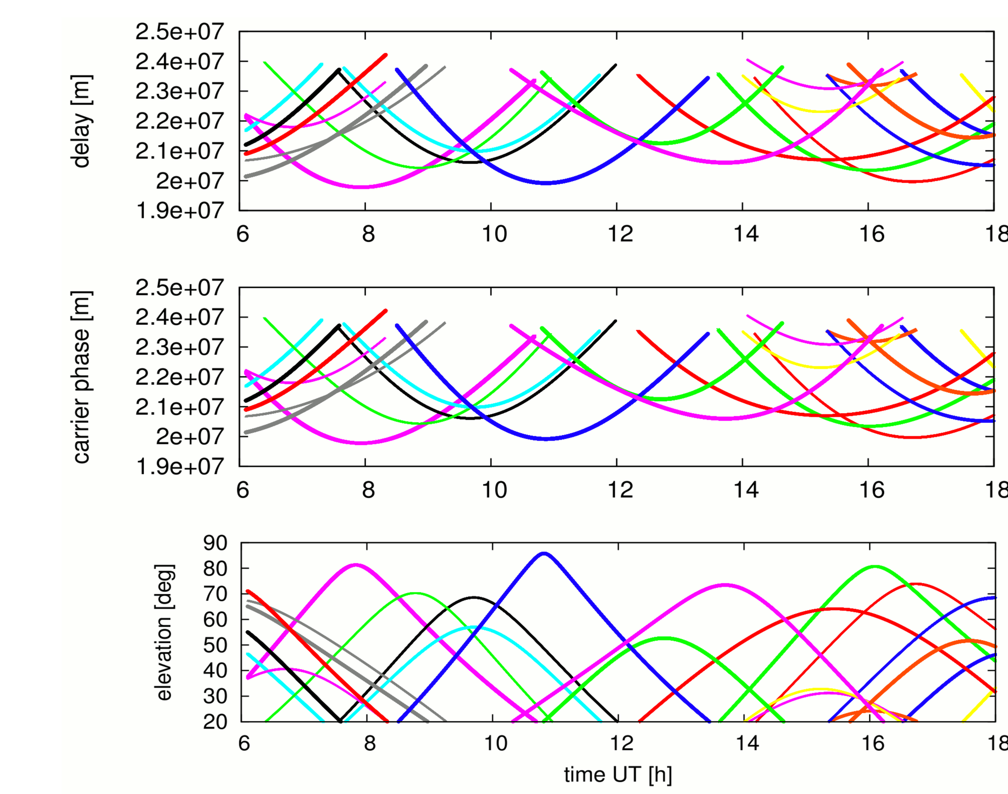

***Figure 7**: Obtained delays and phases as well as the corresponding elevation angles for a 12 hours test starting March. 25th, 2009 sampled with 8 Msps.*

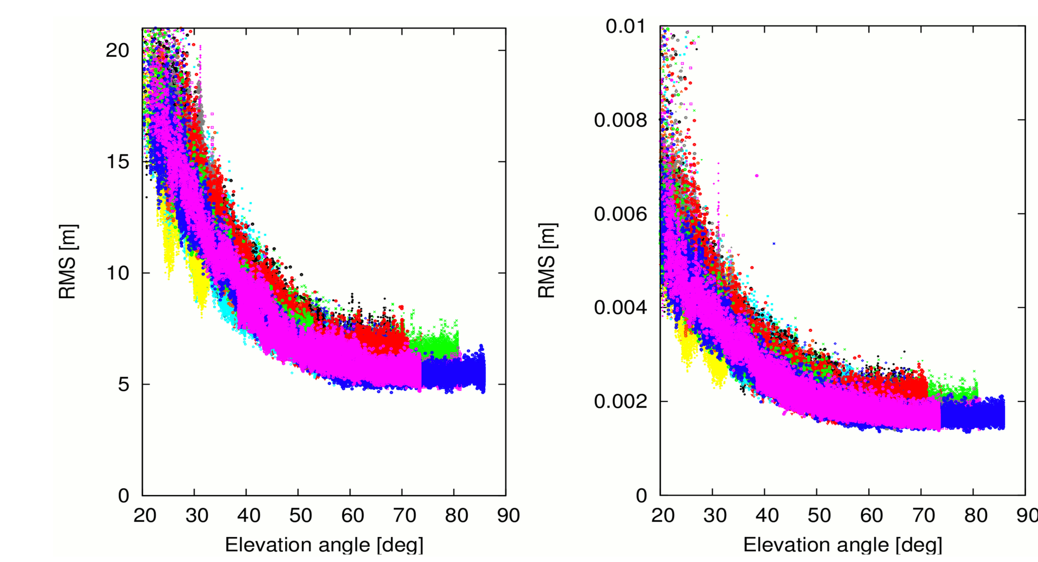

***Figure 8:** RMS of the delays (left) and carrier phases (right) w.r.t. elevation angle for the obtained observables displayed in Figure 7.*

Conclusions

It has been demonstrated (more details in [1]) that an efficient GPS software radio can be implemented on a GPU. For real-time processing the FX strategy seems to be more suitable than its counterpart, the XF architecture, since it takes advantage of the fast parallel FFT implementation which is available for the GPU. Thus, the performance of the FFT engines strongly determines the real-time capability of the receiver and restricts the number of channels when data-rates are exceeding 16 Msps. Future GPU cards and improved FFT implementations are expected to support even higher sampling rates, which enables tracking of weaker signals. Software receivers can be easily adapted to handle new signals and allow testing of new algorithms and signal processing strategies. Thereby, the usage of a GPU appears to be more advantageous than the CPU which offers significantly less computational power.

References

[1] Hobiger T., T. Gotoh, J. Amagai, Y. Koyama, T. Kondo, A GPU based real-time GPS software receiver, GPS Solutions, in print, 2009.