Platforms for a Future GNSS Receiver

A Discussion of ASIC, FPGA, and DSP Technologies

Since introduction of the first GPS receivers more than a quarter century ago, GNSS equipment has changed profoundly – from racks of computers and 25-pound "manpacks" into tiny integrated circuit chipsets suitable for inclusion in mobile phones and other portable devices. But the evolution of GNSS form factors is far from ended. Indeed, the appearance of new GPS and GLONASS signals and the arrival of Galileo has injected new vitality into design of GNSS products. This installment of Working Papers traces the trajectory – past, present, and future – of that technological evolution.

GUENTER W. HEIN, THOMAS PANY,
STEFAN WALLNER, JONG-HOON WON
UNIVERSITY FAF MUNICH

GNSS receiver technology has changed dramatically since the first reception of a GPS signal. It evolved from complex electrical circuits — partly analog — tracking only one satellite at a time to today's sophisticated, small multichannel receivers. The core of a modern receiver is contained in one or more highly sophisticated chips that perform all the receiver's tasks, starting with signal processing, followed by positioning, and often ending at application processing.

The technology to build these chips is called application-specific integrated circuit (ASIC) technology. Following this approach of receiver design, a manufacturer completely designs the chip from scratch, having the maximum flexibility in the design but also facing tremendous development efforts and costs. By selling a large amount of chipsets, the manufacturer hopes to recover those development costs, enabling the company to offer affordable receivers while still making a profit. However, redesign of a receiver ASIC remains a major task and can only be afforded once every several years.

The appearance of new GPS and GLONASS signals and the arrival of Galileo has injected new vitality into design of GNSS products.

Nowadays, improvements in software and hardware technology seem to promise reductions in future receiver development costs by using field programmable gate arrays (FPGAs), digital signal processors (DSPs), or even general purpose processors to realize a complete GNSS receiver. The receiver makes use of these predefined hardware structures, which can be configured (in case of an FPGA) or programmed (in case of a processor) by means of software.

For such projects a large number of elaborate tools are available, providing the engineer with a convenient development environment. This, in turn, will lead to greater design productivity and lower development costs. Furthermore, the software approach makes it possible to run field upgrades of the receiver. The obvious question thus arises —with which technology will the future GNSS receiver be built: ASIC, FPGA, DSP, or even a general purpose central processing unit (CPU)?

In this article we try to answer this question as completely as possible by describing future GNSS hardware platforms, focusing on different techniques to realize GNSS tracking and using our background in receiver technology from the algorithmic and application points of view.

ASIC Technology

For more than 25 years ASIC technology has now been successfully used in the GNSS industry for designing and building GNSS receivers. Varieties of GNSS chipsets based on ASIC technology have been widely introduced into the global marketplace. The general approach used in ASIC-based receivers has provided a well-engineered partitioning of the required computations across fixed and programmable logic. High-speed digital correlation operations are performed in ASIC with parallel hardware-based digital signal processing manner.

In typical ASIC-driven GNSS receiver design, a programmable microprocessor is responsible for controlling code and carrier tracking loops as well as for decoding and processing the navigation data bit stream to provide the user's position solution. Additional correlators increase receiver sensitivity and lower the CPU load. Current ASIC technology makes it possible to integrate tens of thousands of parallel hardware correlators into a single, small chip. This obsoletes the need for separate acquisition and tracking stages.

The currently used development cycle for ASICs is as follows: algorithm development, simulation and initial tests using FPGA, and ASIC implementation.

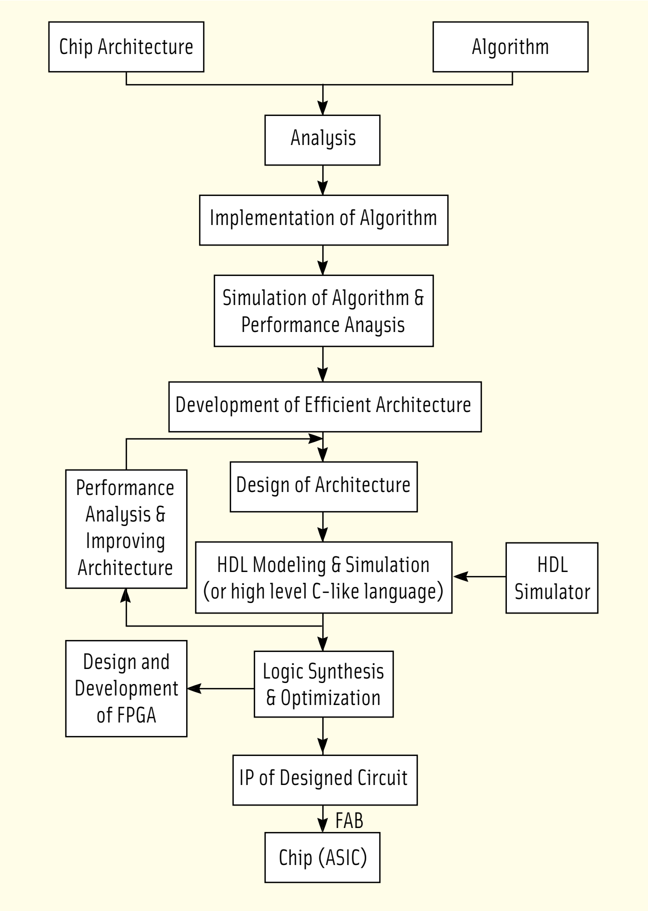

Chip Architecture
Algorithm
Analysis
Implementation of Algorithm
Simulation of Algorithm & Performance Anaysis
Development of Efficient Architecture
Performance Analysis & Improving Architecture
Design of Architecture
HDL Modeling & Simulation (or high level C-like language)
HDL Simulator
Design and Development of FPGA
Logic Synthesis & Optimization
IP of Designed Circuit
FAB
Chip (ASIC)

FIGURE 1. Design flow of ASIC Development

As shown in **Figure 1**, the architecture design begins with analyzing existing chip architectures and signal processing algorithms. The designed architecture is implemented and tested using FPGA. During this stage, hardware description languages (HDLs) may be used to define the electronic circuitry. Also, designers may conduct timing and function simulation based on a custom FPGA.

A board-level receiver solution next might be developed to test an entire hardware/software solution, including the real-time operating system (RTOS) and low-level software modules. After logic synthesis and optimization for the designed architecture, the intellectual property (IP) core of the designed circuits is achieved. Finally, a chip-level solution, which is based on the design IP, is produced and brought to the mass market. This is aimed at dramatically reducing the overall fabrication cost and time of customized high-performance semiconductor chips.

Several new design trends in the ASIC approach have emerged in recent years. The first is chip-level integration, which provides a variety of benefits to manufacturers and system integrators, for example, reducing size and power consumption, improving reliability and performance, and saving cost. The goal of chip-level integration is development of a single-chip solution that includes the RF and digital processing components. Nonetheless, chip-level integration may require a certain amount of development time and cost.

Palinuro, a proprietary ST Microelectronics product, offers a one-chip GPS solution from RF input to NMEA position output. The radio is surrounded by an isolation zone, and care is taken with all connections crossing that zone to ensure that they cannot take noise into the radio. The GPS-only version is already under development, and the hardware has been implemented in FPGA. A GPS+Galileo version will be built in a further phase of development, also in FPGA.

The second ASIC trend is the design of a GNSS receiver as a form of IP (for example, RF IP and baseband IPs for each GNSS). This design approach is considered as one of the potential solutions for mobile applications in the form of PDAs, cell phones, and so on. Successful adoption of GNSS IP into cell phones requires that the technology meets new demands such as indoor operation with low power consumption.

Efforts to eliminate the dedicated CPU and incorporate the GNSS software into a cell phone's existing CPU have, however, been limited by the high CPU load and high interrupt rate associated with traditional GNSS processing. The mission to optimize GNSS for cellular application is leading towards a new architectural design that substantially increases signal processing capability so as to be suitable for shared processor implementation. This new architecture achieves massive parallel correlation and long integration times in dedicated logic, relieving the processor of computationally intensive tasks.

The third trend that we will discuss is system on a chip (SoC) technology. Today's SoC designs are hardware/software systems on silicon rather than just gates on a chip. SoC-based system development combines tightly integrated software and hardware using ASIC or FPGA technology as the main hardware platform. This approach is currently helping to narrow the gap between the design methods of software and hardware designers.

Currently, software designers use off-the-shelf RTOS emulators or development tools for writing high-level applications but are forced to wait for a hardware prototype or real target hardware to finish their development. On the other hand, SoC (ASIC and FPGA) designers are accustomed to working with the details of the hardware design. Their concern is to verify the hardware with a cycle-accurate, and ultimately timing-accurate simulation.

Structured ASICs — a kind of SoC — have been combining the benefits of normal cell-based ASICs and FPGA. They are faster, consume less power and have higher density than FPGAs. Fur-

The software designed radio approach is expected to become more and more important, particularly in the design of multi-mode GNSS receivers.

thermore it is easier to use design tools because they are more configurable and the execution time is shorter than for ASICs (for example, the Force524 Aircraft Navigation GPS of Trimble was designed by using RapidChip of LSI Logic, USA.)

FPGA Technology

More than 20 years ago, programmable devices consisted of programmable array logic (PAL) and complex programmable logic devices (CPLDs), which were essentially planes of NAND/NOR logic gates plus a few registers to realize a digital electrical circuit. Those chips contained the equivalent of a few hundred gates of logic. They were used to replace glue logic and were reprogrammable, allowing the correction of design mistakes quite easily. For example, a state machine could be realized.

About 15 years ago static reserve access memory (SRAM)-based FPGA devices were introduced that contained 1,000–5,000 logical gates or more. At that time configuring those devices was a cumbersome task and often not all gates could be used, because only a small number of development tools were available. To describe in detail the functions of an FPGA would be beyond the scope of this article, but we should mention that FPGAs are composed of configurable logic blocks (CLBs) as shown in Figure 2. They are connected by programmable connections, a process called routing. The number and structure of the CLBs and the flexibility in routing are the key performance parameters of a FPGA device.

A modern FPGA contains the equivalent of 1.5 million logic gates subdivided into 200,000 logic cells. Design engineers can optionally place up to two hardcore processors on the same chip, or the FPGA can load several CPU IP's of very different complexity ("softcore" processors) on its logical devices. In addition high performance input/output hard-IP's (such as Rock IO multi-gigabit transceivers) and even DSP slices are available on several high-performance FPGAs. A very high-performance FPGA provides 256 giga multiply and accumulate commands per second (GMACS).

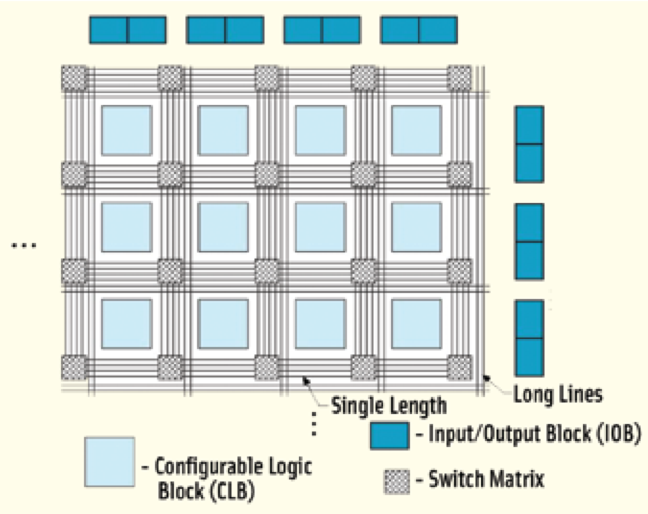


FIGURE 2. Layout of an FPGA

Apart from the hardware, software development tools may represent an even more important cornerstone in FPGA development. An FPGA is configured through use of a netlist, which is either synthesized from an HDL or directly generated by a schematic editor. (An HDL is a language to describe any digital electrical circuit which can be a complete computer or even a simple logical gate.)

The described hardware is then realized on the FPGA, but can also be transferred later to an ASIC. Depending on the abstraction layer, many different grammars are available, such as Very High Speed Integrated Circuit (VHSIC) hardware description language (VHDL), Verilog (low-level grammars), and SystemC, ImpulseC, or Handel-C. Often a GNSS algorithm is already available in C++ or as a Matlab model, which eases the transfer to the FPGA by using one of the high-level languages. This finally may result in an HDL model, too, but the coding process is simplified. On the other hand, use of such high-level FPGA languages results in a less effective use of the chip's resources.

Apart from further increase of the number of logical gates and processing speed, future FPGA development will most likely bring more and more powerful hardcore processors. Softcore processors will be used for less-complex designs and hardcore processors, for more complex designs and to support legacy code. Several very small softcore processors may be used to perform locally specialized tasks in an FPGA instead of using a large central CPU. Eventually hybrid devices will show up making use of FPGA technology on an ASIC. The ASIC part will be frozen while the FPGA part can be changed in the field. It may also be possible that future general purpose processors used in a PC will contain some small FPGA sections for high-speed signal processing.

Software Defined Radio

In part, because the ASIC approach limits the flexibility of receiver architecture for many applications of interest, a new trend in the GNSS receiver design process, the software defined radio (SDR) approach, has evolved. SDR incorporates digitization closer to the receiver antenna front end so as to develop systems that work at increasingly higher frequencies and wider bandwidth.

Let's take a moment to compare the characteristics of SDR with ASICs. SDR accomplishes all digital signal processing using a programmable microprocessor, such as a DSP or a general purpose CPU, rather than an ASIC. This completely separates analog signal conditioning in hardware from all digital signal processing in software and results in significant gains, most notable flexibility. That means the analog signal conditioning components are all very wide-band and capture an exceptionally wide frequency span containing many transmissions. As a result, the particular frequency band of interest is digitally filtered and the resulting samples decimated to accommodate the desired subset of the sampled frequency band.

The SDR approach is expected to become more and more important,

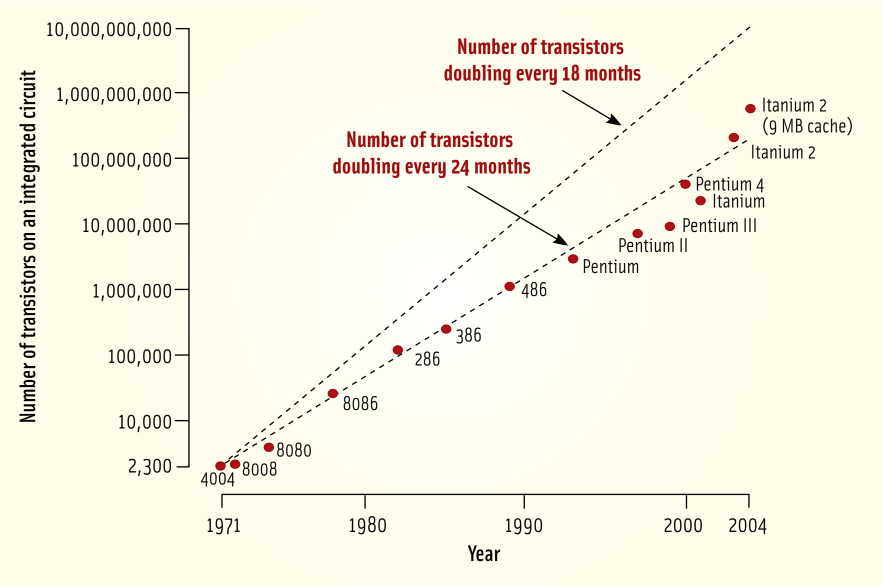


FIGURE 3. Moore's Law

particularly in the design of multi-mode GNSS receivers (GPS+Galileo and so on). We anticipate that a programmable down-converter in a single-chip design will reach the commercial market in the near future. This will enable implementation of a flexible multi-mode GNSS receiver without the need to change the RF portion.

However, the ultimate ideal SDR consisting of an RF part and a microprocessor part capable of operating with ultra-wide bandwidth over a larger frequency range remains unrealistic. Current technology does not allow for a single hardware platform to process the wide variety of all existing RF signals. The ADC limits the spectrum and dynamic range that can be captured and available microprocessors cannot handle the very high data rates that would be required.

Achieving the required MIPS rate (million instructions per second) to process additional GNSS signals with the same amount of cost is a hot issue in designing modernized ASICs or SDRs. Moore's law predicts "the doubling of transistors in IC [the doubling of computational power] every 18 to 24 months," (see Figure 3). Assuming that MIPS of current digital processing technology can fully cover GPS L1 CA code signals and that the required processing power for each modernized GNSS signal is about the same as for GPS L1 CA code signal, then a dual-mode GNSS ASIC or SDR (GPS+Galileo) will be available in commercial market in about two years.

In order to face to an evolution of new GNSSes in the near future, flexibility is a crucial factor in designing GNSS receivers. The SDR approach provides ultimate flexibility to designers and researchers, although ASIC-based GNSS receivers are expected to be a major part in commercial market because of its cost-effectiveness for the next several years.

Digital Signal Processors (DSPs)

A modern desktop computer's CPU is a generalist in terms of the variety of tasks with which it must deal and might be compared to a decathlete; a DSP, on the other hand, is a highly trained specialist similar to a sprinter. As in athletics both hardware components have the same origins but are developed to fulfill different purposes. In addition to the pure processing of data, a CPU handles with its huge repertoire of functions the interaction of all devices of computer hardware, beginning from the management of the hard disk to providing the data for the graphical output on the display.

In contrast, the sole purpose of a DSP is to modify and manipulate the numbers in a digital data stream, and it is "trimmed" to perform these tasks the quickest fashion possible. Consequently, a DSP's instruction set is much smaller than that of a microprocessor, but those functions that are present are geared towards maximal performance. Hence, the DSP can perform its tasks by relying on many fewer transistors than a CPU, which translates into much lower power consumption, and thus makes DSPs ideal for mobile devices that have to be powered by batteries or power cells.

This trend has been characterized by Gene Frantz, Texas Instruments DSP Business Development Manager, in a formula known as Gene's Law: the power usage of integrated circuits decreases exponentially every 18 months, leading to reductions in the size of devices built around these chips and to longer battery life. The evolution of DSPs in the last two decades can be seen in Table 1 and Figure 4.

As the processing capabilities of DSPs steadily increased at the same time that their prices were dropping, they eventually reached the point where they could affordably be used in GPS receivers, as was shown by CEVA in May 2004 with its first DSP-based GPS solution "Xpert-GPS." According to CEVA the costs of implementing GPS using their hardware does not exceed US$3 per unit. By implementing this GPS receiver unit into cell phones, further reduction of costs can be achieved while supporting all wireless standards such as GSM, GPRS, and UMTS.

As we look into conventional DSP technology, we see that the multiply and accumulate command (MAC) block is one of the bottlenecks of DSPs, because each bit requires a separate operation, and the dot-product is one of the most common operations performed in a GNSS receiver (signal correlation for tracking and acquistion). Rather than simply raising the clock speed, another way has to be found to improve the processing power significantly. The limiting conditions, which are set by the MACs, are counteracted in enhanced DSPs by using multiple MACs, but this approach is also limited to the small number of these blocks — four to eight — that are able to operate in parallel. These

	1982	1992	2002
Die size (mm)	50	50	50
Technology size (microns)	3	0.8	0.18
MIPS	3	40	5000
MHz	20	80	500
Price (US $)	150	15	1.50
Power dissipation (mW/MIPS)	150	12.5	0.1
Transistors	5e4	5e5	5e6

TABLE 1. Evolution of DSPs in the last decades (data extracted from Gene Frantz, "Digital Signal Processor Trends", *IEEE Micro*, Vol 20, No. 6, 2000, pp. 52-59

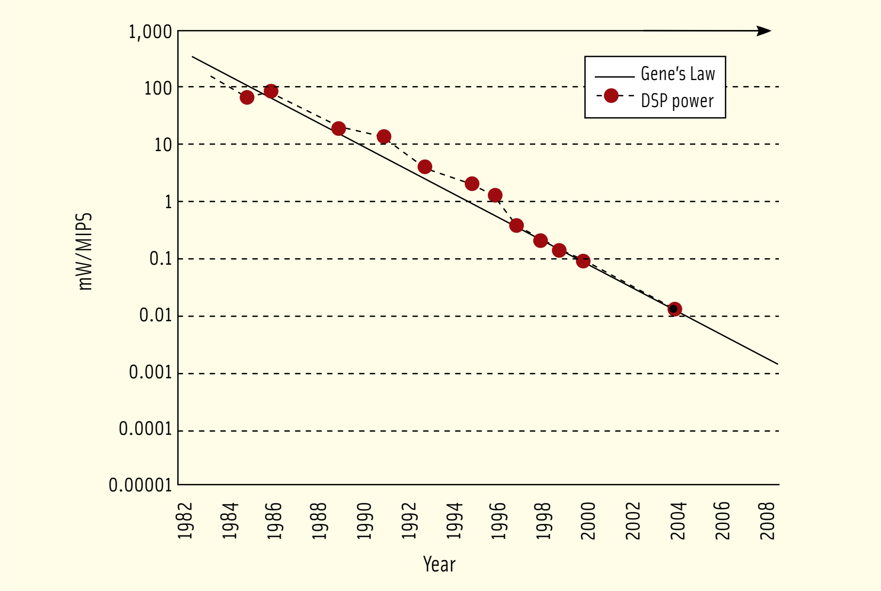


FIGURE 4. Evolution of power consumption relative to processing power during the last two decades [from Gene Frantz, "Digital Signal Processor Trends" *IEEE Micro*]

enhanced DSPs provide improved processing power, because they allow more operations to be carried out every cycle step, but they still suffer from the drawbacks of conventional DSPs.

Compiling programs that are written in higher languages (e.g. C) is sometimes impossible, and they have to be programmed in assembly language. Multi-issue DSPs use very simple instructions that typically encode a single operation, so that a high level of parallelism can be achieved. In contrast to executing one instruction per time, these DSPs allow small parallel groups to be dealt with together.

Texas Instruments was the first to implement this multi-issue technology in commercial DSP's with its TMS320C6sxx and it was dramatically faster than the previous generations of DSPs. This substantial progress in processing performance results from the simpler, more regular architecture and instruction set of the multi-issue DSPs, which is also reflected in a very efficient code generation by the compiler. However, when comparing multi-issue DSPs to their ancestors — conventional and enhanced DSPs — they clearly suffer from higher power consumption because they were trimmed with an emphasis on processing performance, not energy efficiency.

When solving the drawbacks of DSPs some designers are using the same strategy as for speeding up CPUs. Instead of steadily increasing the clock rate of CPUs, which generates more heat problems, designing multi-core chips has become the trendy solution. But this raises the issue of creating software able to exploit the advantages of parallel DSPs. Fortunately the multi-channel concept employed in a GNSS receiver is parallel in nature and can be readily transferred to a multi-core chip.

On the DSP side, designers focus on the hybridization of DSPs and FPGAs as co-processors, which enables highly parallel DSP processing. According to a Xilinx Inc. "roadmap" for DSP/FPGA technology (See citation 23 in the Additional Resources section at the end of this article.), this trend offers a reduction of power consumption by 50 percent compared to pure FPGA solutions and an increase in terms of performance by a factor of 10 when compared to stand alone DSPs.

This strategy of combining the advantages of DSPs and FPGAs can be found in different areas, where huge amount of data streams have to be handled. One field is the video coding and processing, where enormous quantities of data have to be manipulated as the users' demand for quality (video and audio) is steadily increasing. Also in the field of GPS applications developments are towards a hybrid FPGA/DSP board, as was shown at last year's ION GNSS conference in a paper presented by F. Covis and colleagues describing a GPS/Galileo hybrid FPGA/DSP board (see Additional Resources).

This approach exploits the high computational power of FPGAs for high data rates together with the signal processing and mathematical capabilities of DSPs.

Today, general purpose processors – looking especially at the high-end CPUs – are able to outdistance the fastest DSPs in terms of processing performance of typical DSP tasks from themselves. At first glance this statement appears quite contradictory to the premise at the beginning of this section regarding the advantages of "specialist" DSPs. However, one has to keep in mind that the performance advantage of CPUs is totally based on superior clock speed. While Intel is pushing at the 4 GHz barrier with it's Pentium 4 processor, the fastest high-performance Texas Instruments DSP is running at 1 GHz. But this doesn't mean the end of DSPs as they are still the best choice when focusing on a balance between price, performance, and power consumption.

General Purpose CPU

Although a general purpose CPU does not represent a primary platform for GNSS receiver development, this situation has been changing for some years, due to the continual increases in processing power driven in part by computer

multimedia applications. For example, most modern PC CPUs (Intel or AMD) support the SSE2/3 instruction set with which one operation (for example, multiplication) can operate on several arguments (single instruction, multiple data or SIMD).

The CPU also provides special commands to calculate dot products efficiently and today it can achieve about 20,000-30,000 MIPS. Even more astonishing is the computational power of new CPU architecture like that used in the cell multiprocessor. (See the articles on that subject, citations 14 and 4 in the Additional Resources section.) It combines a "conventional" power processor with up to eight newly architected synergistic processor elements (SPEs). The SPE implements a new instruction-set architecture optimized for power and performance on computing-intensive and media applications. Prototype implementations show a processing performance of 200 GFLOPS per CPU.

Depending on the algorithm, a GNSS receiver may need six operations to correlate one intermediate frequency (IF) sample per channel. Thus, for a high-end receiver with 36 channels, each operating at 40 MHz, 8.64 billion operations need to be computed per second, a figure which at the first glance is easily achieved by those CPUs. However, that calculation assumes that all data is available to the processor with no latency when, in practice, the CPUs interface to the main memory often provides a major bottleneck.

Future GNSS Receiver

In the past the GPS market has been driven more by applications than advances in the infrastructural technology, because changes in the space segment of GPS were quite marginal. Core technologies such as signal processing and navigation processing have shown only medium-level development and remained virtually fixed before GPS modernization and the development of Galileo. Furthermore, present day GPS tracking and positioning algorithms (before introduction of the second civil signal, L2CS) are near their theoretical limits. Most developments are done at the application level based on integrating OEM GPS receivers or chipsets.

GPS modernization and Galileo are now changing the environment of GNSS receiver design, and new signal processing algorithms are being developed. Thus, we are seeing an increased demand on development tools. During this transition period (which could be quite long, perhaps 6 to 10 years), commercial receivers may employ technology that can be changed in the field. Due to the continuous and flexible development this allows, such systems will have an advantage.

Normally this kind of transition phase ends when technology development stops making significant improvements. After the coming round of GNSS modernization reaches fruition, a similar situation will occur as nowadays where established chipsets, OEM modules, or receivers will dominate the market. However, depending on the outcome of the GPS III program or the eventual commencement of a Galileo II program, the transition phase will never stop creating a need or opportunity to continuously improve core GNSS receiver technology.

For mass market applications, an ASIC-based receiver will most likely always provide the smallest single-unit production costs and consequently serve as the technology of choice. FPGA and software based receivers may serve as development tools, but might not survive as mass-market applications. However, many GNSS applications are not building a numerically large market because only a limited number of units can be sold. Those areas include, for example, geodesy and surveying, aviation, reference stations, timing, and others.

As the processing capabilities of DSPs steadily increased at the same time that their prices were dropping, they eventually reached the point where they could affordably be used in GPS receivers.

Conclusions

The question stated in the introduction — which technology will be used for future GNSS development — can be rephrased as, "What is more important: flexibility, development costs, single unit production costs, or power consumption?"

Table 2 shows the benefits and drawbacks of the various platforms used for signal-processing in a GNSS receiver. An entry of "++" (or, respectively, "--") represents a major advantage/disadvantage of this technology in the corresponding category. An entry of "+/-" means that the technology is more or less suited for this category, but this judgment clearly contains our personal opinion.

The categories for development cost, power consumption, flexibility, and single unit costs have been already discussed in the main text. Regarding performance, it should be added that with each technology a high performance can be achieved just by increasing the number of chips or processors used.

For mass-market applications, ASIC technology may still dominate the scene in the future, simply because of the low production costs. For high-end applica-

Technology	Development Costs	Performance	Power Consumption	Single Unit Costs	Flexibility
ASIC	--	++	++	++	--
FPGA	-	++	+	-	+
DSP/CPU	++	+/++	+/--	+/-	++
Hybrid FPGA/CPU	+	++	+	-	+

TABLE 2. GNSS technology comparison

tions, ASIC technology might in the long run dominate again, but here an outcome is much more difficult to predict. Especially as the performance of FPGAs, DSPs, and general purpose CPUs continues to increase, they will create possibilities and applications making them competitors even to highly developed ASICs.

The intrinsic flexibility in these systems will bring them always one nose ahead. Nevertheless, those "new possibilities" have to be continuously found. For example, signal optimization in the space segment provides such an opportunity. A DSP-based software receiver has a good chance to be used in applications where a DSP already exists and processing time is available (e.g., in an embedded system). A highly sophisticated DSP or general purpose CPU might even be a cost-efficient platform for a high-end receiver. A genuine software receiver (running on a PC) is the ideal teaching and research tool and might prove to be valuable for monitoring applications due to extremely flexible data inspection, monitoring and logging possibilities. FPGA receivers will have their place in the GNSS control segment, where development costs are the dominant factor. Additionally the control segment receiver algorithms can be continuously updated resulting in an steadily improving overall GNSS performance.

Additional Resources

[1] BDTI, http://www.bdti.com/articles/slides.evolution.pdf, November 14, 1997

[2] Battson N., "Using domain-optimized FPGAs in high-performance DSP applications", http://www.dsp-fpga.com/articles/battson/, May 9, 2005

[3] Bruister J., "Bridging the Hardware/Software Design Gap", *SoC Solutions,* TechOnLine Publication, March 2001

[4] Cell based board prototype, http://techon.nikkeibp.co.jp/english/NEWS_EN/20050525/105050/?ST=english, May 2005

[5] CEVA, http://rfdesign.com/news/DSP-GPS-solution/, May 20, 2004

[6] Covis F., M. Spelat, P. Mulassano, C. Leone, "On the Tracking Performance of a Galileo/GPS Receiver Based on Hybrid FPGA/DSP Board", *Proceedings of the International Technical Meeting of the Institute of Navigation,* ION-GNSS 2005, Long Beach, September 2005

[7] Daniels T., "Structured ASICs and Their Use in Embedded IP SOC Designs", LSI Logic Corporation, November 2005

[8] DSP-FPGA.com, http://www.dsp-fpga.com/articles/banks_and_chung/, September 19, 2005

[9] Eyre J., J. Bier, "The Evolution of DSP Processors", Berkeley Design Technology, 2000

[10] Franzt G., "Digital Signal Processor Trends", *IEEE Micro,* Vol. 20, No. 6, 2000, pp. 52-59

[11] Impulse C, http://www.impulsec.com/, February 2006

[12] Intel Technology Journal, vol. 8, no. 1, http://www.intel.com/technology/itj/2004/volume08issue01/art01_microarchitecture/p06_sse.htm, February 2004

[13] Intel, http://www.intel.com/products/processor/pentium4HTXE/index.htm

[14] Introduction to the Cell multiprocessor, http://researchweb.watson.ibm.com/journal/rd/494/kahle.html, September 2005

[15] MacGougan G., P. L. Normark, C. Ståhlberg, "The Software GNSS Receiver", *GPS World,* January 2005

[16] Mattos P. G., "A Single-Chip GPS Receiver and the Steps to Mass-Market Galileo", *GPS World,* October. 2005

[17] Pany, T., M. Irsigler, B. Eissfeller, K. Fürlinger, K, "Performance Assessment of an Under Sampling SWC Receiver for Simulated High-Bandwidth GPS/Galileo Signals and Real Signals", *Proceedings of the International Technical Meeting of the Institute of Navigation,* ION-GPS 2003, Portland, September 2003

[18] SystemC, http://www.systemc.org/, February 2006

[19] The future of programmable logic, http://www.us.design-reuse.com/articles/article6404.html, January 2006

[20] van Diggelen F., C. Abraham, "Indoor GPS: The No-Chip Challenge", *GPS World,* September. 2001

[21] Verilog, http://www.verilog.com/, February 2006

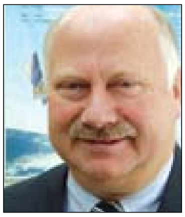
[22] XILINX Virtex-4 Multi-Platform FPGA, http://www.xilinx.com, February 2006

[23] XILINX, http://www.xilinx.com/prs_rls/dsp/05102dsp_roadmap.htm , February 2006

[24] Lück, T., M. Bodenbach, J. Winkel, F. Förster, "An Experimental Multi Band Receiver for the German Galileo Test Environment (GATE)", *Proceedings of the International Technical Meeting of the Institute of Navigation,* ION-GPS 2005, Portland, September 2005

Authors

"Working Papers" explore the technical and scientific themes that underpin GNSS programs and applications. This regular column is coordinated by **PROF. DR.-ING. GÜNTER HEIN**



Prof. Hein is a member of the European Commission's Galileo Signal Task Force and organizer of the annual Munich Satellite Navigation Summit. He has been a full professor and director of the Institute of Geodesy and Navigation at the University FAF Munich since 1983. In 2002, he received the United States Institute of Navigation Johannes Kepler Award for sustained and significant contributions to the development of satellite navigation. Hein received his Dipl.-Ing and Dr.-Ing. degrees in geodesy from the University of Darmstadt, Germany. Contact Professor Hein at <Guenter.Hein@unibw-muenchen.de>.

Stefan Wallner studied at the Technical University of Munich and graduated with a diploma in techno-mathematics. He is now a research associate at the Institute of Geodesy and Navigation at the University of the Federal Armed Forces Germany in Munich. His main topics of interest include the spreading codes and the Galileo Signal Structure.

Jong-Hoon Won received his B.S. and M.S. degrees in control and instrumentation engineering at Ajou University, Suwon, Korea. He also received his Ph.D. at the same university based on the design and development of a software-based GPS receiver and precise navigation algorithm. Since 2005, he has worked as a research associate with the Institute of Geodesy and Navigation of the University of Federal Armed Forces Munich, Germany. His current research interests are in the signal processing and navigation algorithms for GNSS receivers.

Thomas Pany has a Ph.D. in geodesy from the Graz University of Technology and a MS in Physics from the Karl-Franzens University of Graz. Currently he is working at the Institute of Geodesy and Navigation at the University of Federal Armed Forces Munich. His major areas of interest include GPS/Galileo software receiver design, Galileo signal structure and GPS science.