PERFORMANCE COMPARISON OF MICROPROCESSORS FOR SPACE-BASED NAVIGATION APPLICATIONS

Sergio De Florio(1), Eberhard Gill(2), Simone D'Amico(1) and Andreas Grillenberger(1)
(1) DLR, Space Flight Technology Department, D-82230 Wessling, Germany
e-mail: sergio.deflorio@dlr.de, Phone: +49 (0) 8152 28 2357
(2) Chair of Space Systems Engineering, TUDelft, Faculty of Aerospace Engineering,
Kluyverweg 1, 2629 HS Delft, The Netherlands
e-mail: E.K.A.Gill@tudelft.nl, Phone: +31 (0) 15 27 87458

Abstract

In this paper the absolute and relative run-time performance of different space-borne microprocessors are compared. A realistic test scenario which comprises the propagation of a state vector and its transition matrix for a satellite in Low Earth Orbit (LEO) has been used to perform the tests. In addition, the computation time required for a Kalman filter time update with respect to the order and degree of the Earth's gravity field has been evaluated. The microprocessors considered are typical platforms for space-based applications. In particular detailed benchmark testing is conducted on the ARM7TDMI (Phoenix GPS receiver on the X-SAT mission), on the PowerPC823e (BIRD spacecraft onboard computer), LEON (PRISMA spacecraft onboard computer) and NiosII (NAMURU board for GPS applications). The benchmark software has been integrated as sole application into the respective operating and software system of the microprocessors. The analysis has been performed in the framework of the Swedish led PRISMA formation flying mission in which the German Aerospace Center (DLR) provides various key contributions which comprise a redundant GPS hardware architecture for the two spacecraft, a real-time navigation software to support formation flying during all phases, and dedicated experiments for absolute and relative orbit control.

1. Processors Description

A set of four microprocessor types has been selected for benchmarking. These are the Arm7TDMI, the PowerPC823e, NiosII and LEON. The PowerPC823e has already been used for space-borne navigation applications in the framework of the BIRD satellite mission (Ref. [1] and [2]). In the upcoming X-SAT mission of the Nanyang Technological University of Singapore, the Arm7TDMI will be employed for GPS tracking and autonomous navigation (Ref. [3]). The LEON Field Programmable Array (FPGA) based processors are employed in the PRISMA Formation Flying Mission (Ref. [4], [5] and [6]) of the Swedish Space Corporation (SSC). The NiosII is neither space qualified nor tested in terms of thermal vacuum and radiation tolerance and is under development as platform for spaceborne GNSS applications (Ref. [7] and [8]).

1.1 ARM7TDMI Processor

ARM7TDMI is a 32 bit embedded RISC microprocessor kernel, implementing the ARM Thumb instruction set. An ARM7TDMI processor kernel is also employed in the GP4020 baseband processor that forms the core of DLR's Phoenix GPS receiver (Ref. [9]). The ARM7TDMI microprocessor does not comprise a floating point unit. The clock rate of the processor may be selected by software. For the particular test conducted, a clock rate of 30 MHz has been configured. The microprocessor unit considered for testing is part of a Phoenix GPS receiver unit of DLR/GSOC. The receiver's GPS tracking software has been augmented by autonomous navigation software to be flown as part of the X-SAT satellite mission of the Nanyang University of Technology, Singapore (Ref. [10]).

1.2 PowerPC823e Processor

The PowerPC823e is an industrial microprocessor operated at 48 MHz clock rate without floating point support which provides a performance of about 66 MIPS. The processor was the core of the onboard computer for the German BIRD microsatellite mission, which comprised a GPS-based onboard navigation system. The performance test described in this note was executed on an integrated breadboard which has been built by the Institute for Computer Architecture and Software Technology (FIRST) of the Fraunhofer Gesellschaft (FhG). The real-time operating system

BOSS developed by FhG/FIRST separates the kernel run-time system and a hardware dependant layer, which allows the emulation on standard Linux workstations as well as an easy adaptation to different processors. BOSS is a pre-emptive multitasking operating system well suited for real-time and onboard applications.

1.4 NiosII Processor

The Namuru V2 Receiver has been developed at the University of New South Wales (UNSW) in Sydney, Australia. The receiver uses an Altera CycloneII 2c50c6 FPGA chip for the integration of the baseband processing module and the microprocessor. Both frontends use the Zarlink GP2015 RF chip for GPS L1 at 1575.42MHz, one frontend can be configured to GPS L2 by using an onboard up-converter circuit. The board offers one USB 2.0 and two RS232 interfaces, 64MB SDRAM and 8MB flash memory. The baseband processor and the NiosII soft-core CPU are implemented in the FPGA. This allows changing the baseband processor and the CPU and adapting it to specific needs. The baseband processor has been designed by UNSW and is public domain.

The hardware is neither space qualified nor tested in terms of thermal vacuum and radiation tolerance, current testing is limited to a lab environment using signal simulators. Table 1 collects the main features of the two NiosII board configurations considered for the tests. The used FPGA is an Altera Cyclone II 2c50c6 on Namuru II board with a total of 50528 Logic Elements (LE), 594432 memory bits and 86 DSP slices.

Table 1. Main features of NiosII board configurations.

Component	Big – config	Small - config
CPU	Nios 2/f, 16k data cache, 16k instruction cache, hardware multiplier/divider	Nios 2/s, no cache, no hardware multiplier/divider
LE	5157 - 10.21 % of total	2355 - 4.66 % of total
MEM	296704 bits - 49.91 % of total	14336 bits - 2.41 % of total
DSP	4 - 4.65 % of total	0 – 0 % of total

1.5 LEON Processors

LEON2 and LEON3implements a 32-bit processor compliant with the SPARC V8 architecture. The LEON board is designed for embedded applications with the following features on-chip (Ref. [11]): separate instruction and data caches, hardware multiplier and divider, interrupt controller, two 24-bit timers, two UARTs, power-down function and watchdog, 16-bit I/O port, flexible memory controller. To simplify initial software development, Gaisler Research is providing LECCS, a free C/C++ cross-compiler system based on the GNU C Compiler (gcc) and the Real-Time Executive for Multiprocessor Systems (RTEMS) real-time kernel (Ref. [12]). LECCS allows cross-compilation of single or multi-threaded C and C++ applications for the LEON board. Main difference between the LEON2 and LEON3 is that the last one has a Faul Tolerant (FT) memory controller. Table 2 collects the main features of the two LEON board configurations considered for the tests.

Table 2. LEON2 and LEON3 boards main features

Component	LEON2	LEON3
CPU	LEON2, SPARC V8 processor 40 Mhz, win8, hwbp 4, V8 mul/div, lddel 1	LEON3, SPARC V8 processor 24 Mhz, win8, hwbp 4, itrace 128, lddel 1
GRFPU	Icache 2*8 kbyte, 32 byte/line lrr dcache 2*8 kbyte, 32 byte/line lrr	Icache 1*8 kbyte, 32 byte/line lrr dcache 1*4 kbyte, 16 byte/line lrr
Memory Controller	LEON2 Memory Controller (ver 0) 64-bit prom 0x000000000 32-bit sdram: 1*64 Mbyte 0x40000000, col 9, cas 2, ref 7.8 us	Gaisler Research FT Memory Controller (ver 0) 32-bit prom 0x000000000 32-bit sdram: 1*64 Mbyte 0x40000000, col 9, cas 2, ref 7.7 us
Operating System	RTEMS	RTEMS

1.6 Summary of Processors

A summary of the key parameters of the considered processors is given in Table 3.

Table 3. Basic characteristics of selected microprocessors for spaceborne navigation applications.

Processor Type	Mission	Clock rate [MHz]	Floating Point Unit
ARM7TDMI	X-SAT	30	No
PowerPC823e	BIRD	48	No
NiosII	-	80	No
LEON2	PRISMA	40	Yes
LEON3	PRISMA	24	Yes

2. Test Case Specification

To provide a simple yet typical test case for a performance evaluation, an orbit propagation scenario for a satellite in Low Earth Orbit (LEO) has been selected. The considered dynamic model described in Table 4 comprises the Earth's gravity field, taking into account a spherical harmonic expansion complete to degree and order 10 (Model a) or 20 (Model b), respectively. In addition, the Sun and Moon third body forces are accounted for as well as the solar radiation pressure and the atmospheric drag, based on a modified Harris-Priester model for the atmospheric density. An Earth-fixed frame has been adopted as reference system for the integration of the state vector y (position, velocity) and the transition matrix $\boldsymbol{\Phi}$ (partial derivatives of state vector at time t_{i+1} w.r.t. state vector at time t_i). Thus, a proper consideration of the respective Coriolis and centrifugal forces is required in the equations of motion and the variational equations. While the Coriolis and centrifugal terms have been properly treated in terms of precession, nutation and sideral time, simplified transformations have been applied for the perturbation forces.

Table 4. Overview of dynamical models.

Item	Description
Earth Gravity field	GGM01S 10x10 (Model a) and GGM01S 20x20 (Model b)
Tides	None
Luni-solar gravity	Analytical series expansion of luni-solar coordinates
Solar radiation pressure	Canon-ball model, conical Earth shadow
Atmospheric drag	Modified Harris-Priester model
Empirical forces	None
Numerical integration method	RK4R: Runge-Kutta 4th-order extended by Richardson extrapolation

The integration of the state vector and the transition matrix is accomplished by means of a Runge-Kutta (RK) 4th-order numerical integrator, extended by Richardson extrapolation (RK4R), which is adequate for spaceborne navigation applications (Ref. [1]). Each Runge-Kutta step covers a fixed step size of 30 s.
The test comprises two basic cases:
The integration of the state vector y using a precise force model
The integration of the state vector y using a precise force model and the additional integration of the state vector y' and transition matrix $\boldsymbol{\Phi}$ based on a simplified force model (Keplerian motion plus J_2 of the Earth's gravity field).
The latter case basically corresponds to a complete time update step of the Kalman filter in an orbit determination process. Each of the above cases is performed for Model a and b in order to assess the dependency on the complexity of the force model. In addition, to verify the linearity of the CPU performance with the number of steps, the above test cases are executed for 1, 10, and 20 calls of the integrator, respectively. The executed computation time on each processor has been determined from built-in CPU time monitoring functions and checked by external timing based on control outputs of the sample application. Furthermore, the test software had been configured in such a way, that additional tasks were either completely eliminated or reduced to a necessary minimum to avoid computational overhead affecting the results. The software for the benchmark test was written in C++.

3. Results of the Benchmark Test

Each of the tests provided a value for the run time of the application in units of seconds. The absolute run time values for the three microprocessors are given in Table 5. The table comprises a total of 8 tests for each microprocessor, split into four tests for Model a and Model b, respectively.

Table 5. Results of run time benchmarking in absolute terms [s].

Absolute [s]		ARM7TDMI		PPC823		NiosII – big config	
	Number of steps	t(y)	t(y)+t(ϕ)	t(y)	t(y)+t(ϕ)	t(y)	t(y)+t(ϕ)
Model a	1		1.9		4.0		2.7
	10		19.6		40.0		26.8
	20	22.0	39.2	46.6	80.1	30.8	53.5
Model b	1		2.4		5.3		3.5
	10		24.7		52.8		35.4
	20	32.0	49.4	71.2	105.6	48.1	70.8
Absolute [s]		NiosII – small config		LEON2		LEON3	
	Number of steps	t(y)	t(y)+t(ϕ)	t(y)	t(y)+t(ϕ)	t(y)	t(y)+t(ϕ)
Model a	1		74.7		0.44		1.3
	10		746.6		4.4		12.9
	20	872	1493	4.3	8.8	12.6	25.7
Model b	1		101.2		0.46		1.4
	10		1012.3		4.7		14.5
	20	1403	2025.0	4.8	9.3	15.1	29.0

The left column, denoted $t(y)$, indicates the results from a pure state vector prediction, while the right column, denoted $t(y)+ t(\Phi)$, indicates the results from state vector and transitions matrix computation. As expected, the resulting run-time performance is strictly linear with the number of integration steps, indicating that no other computation intensive processes are running which might degrade the results. For the state vector prediction, the increase in size of the gravity field by a factor of two leads to an increase of the run time by factors of only 1.45, 1.53, 1.56, 1.6, 1.11 and 1.19 for ARM, PPC, NiosII (big and small configurations), LEON2 and LEON3 respectively. Considering only the computation of the Earth's gravity field, an increase in run-time of a factor of 4.0 might be expected from theory, while a dedicated test has indicated a factor of 2.8. Smaller values thus indicate that significant computation time is required for tasks not directly related to the Earth's gravity field. For a time update step of a Kalman filter (see also Ref. [13]), the increase in size of the gravity field by a factor of two leads to an increase of the run time by a factor of 1.26, 1.31, 1.32, 1.35, 1.1 and 1.12 for ARM, PPC, NiosII (big and small configurations), LEON2 and LEON3 respectively. These figures are smaller than those for a pure state propagation, since the computational time for the state transition matrix is not affected by an increase in the gravity field size (limited to J_2). Regarding the relative run time performance of the tested microprocessors in terms of the ratio of respective CPU times, in general it is found, that the ARM processor outperforms the PPC in CPU time by a factor of two. The LEON2 processor on the other hand, outperforms the ARM processor by a factor of 4-5. Thus, the LEON2 processor is faster by a factor of 8-12 than the PPC. The LEON2 outperforms the LEON3 by a factor 3.1 and the NiosII by a factor of 10 (big configuration) and 292 (small configuration). The different performances depend mainly on the different CPU clock rates and the use of an FPU (cf. Table 3).

4. Performance of the LEON3 Board for the PRISMA Mission

The results showed in Sec. 3 have been used in selecting a proper board for the PRISMA mission. PRISMA is a micro-satellite mission created by the Swedish National Space Board (SNSB) and Swedish Space Corporation (SSC), which serves as a test platform for autonomous formation flying and rendezvous of spacecraft. The formation comprises a fully maneuverable micro-satellite (MANGO) as well as a smaller sub-satellite (TANGO) which are launched together in a clamped configuration and separated in orbit after completion of all checkout operations The design orbit of the satellites is sun-synchronous, near-polar at approximately 700 km altitude with local time of the ascending node at 6.00 or 18.00. The mission schedule foresees a launch in fall 2009 of the two spacecraft with a targeted lifetime of at least eight months. The PRISMA mission primary objective is to demonstrate in-flight technology experiments related to autonomous formation flying (Ref. [5]) to which DLR will contribute with the Autonomous Formation Control (AFC) experiment. The backbone navigation sensor is based on GPS receivers on both satellites

(Ref. [6]). One of the PRISMA secondary objectives is the demonstration of autonomous orbit keeping of the MANGO spacecraft which will be performed at the end of the mission, through the Autonomous Orbit Keeping (AOK) experiment (Ref. [16]). One of the main challenges of the PRISMA formation flying is the realization of an on-board navigation system for all mission phases which is robust and accurate even for various spacecraft orientations and frequent thruster firing for orbit control. The goal of the absolute and relative orbit determination is to achieve an accuracy of 2 m and 0.1 m, respectively (1 σ) and to provide continuous position and velocity data of the participating spacecraft at a 1 Hz rate for guidance and control purposes. The PRISMA On-Board Software (OBS) consists of two main layers, Basic Software (BSW) and an Application Software (ASW). The ASW consists of a number of application-cores implementing guidance, navigation and control, thermal control, power control, payload control functionalities etc. All these application-cores are executed through a real-time monotonic scheduler, i.e. they all have a specified sample time and their priority depends on the sample time: the smaller the sample time, the higher the priority. The DLR software contribution to the PRISMA OBS consists of specific application cores within the BSW and ASW. The GPS-based navigation system is split into three modules located in different OBS levels and running at different sample rates. The GPS interface (GIF) is part of the BSW, runs at 1 s sample time and is directly fed with GPS messages issued by the Phoenix GPS receivers on-board MANGO and TANGO. The GPS-based Orbit Determination (GOD) and GPS-based Orbit Prediction (GOP) are embedded in the ASW layer as part of the ORB core (30 s sample time) and the GNC core (1 s sample time), respectively. GOD implements an extended Kalman filter to process GRAPHIC observables as well as single difference carrier phase measurements from MANGO and TANGO. Attitude data from both spacecraft are applied to correct for the GPS receivers antenna offset with respect to the spacecraft center of mass. Furthermore, a history of maneuver data is provided to GOD and taken into account in the orbit determination task. GOD performs a numerical orbit propagation which is invoked after the measurement update and provides orbit coefficients for interpolation to GOP for both spacecraft.

PRISMA OBS runs on a LEON3 board under the operating system RTEMS. The flight software is validated in real-time through the inclusion of hardware in the loop (for a detailed description of DLR hardware-in-the-loop test environment see Ref. [17]). The preliminary evaluation of the memory usage and computational load of the DLR's flight software is performed on a LEON3 microprocessor FPGA board which is representative of the MANGO spacecraft on-board computer (cf. Table 2). The execution time of the DLR's flight software on the LEON3 board is dominated by the GOD block. Table 6 collects the execution times of the sub-processes in which the GOD software module is subdivided. These results are obtained from the execution of the max-path test defined for GOD. A max-path is defined as a unit test that reproduces the conditions of maximum computational load of a software module via the provision of a minimum quantity of constant inputs (Ref. [14] and [15]).

Table 6. GOD execution times on the LEON3 board

Step		0	1	2
Execution times [s]	Process input and data editing	0.026	0.07	0.09
	Trajectory and filter initialization	0	3.3	0
	State interpolation	0	0	0.853
	Measurement update	0	0	3.5
	Time update	0	0	3.1 = 2 x 1.55
	Process output	0	453	453
	GOD block	32	3.9	7.9

The main contribution to the execution time of GOD is given, as expected, by the measurement update section. The orbit perturbation model implemented for the time update is similar to Model b of Table 4. The time update execution time of Table 5 (1.4 s) is fully consistent with the LEON3 test result of Table 6 (3.1 s) considering that in GOD each task is performed two times (two satellites).

5. Conclusions

The computing performances of four microprocessor types, namely the ARM7TDMI, the PowerPC823, the NiosII (in two different configurations) and the LEON (in two different configurations), have been compared for spaceborne navigation tasks. To that end, a realistic test scenario has been established which comprised the propagation of a state vector and its transition matrix for a satellite in LEO. In addition, the required computation time for a Kalman filter time update with respect to the order and degree of the Earth's gravity field has been evaluated. Here, it turned out that doubling the order and degree of the Earth's gravity led to an increase of the computation time of about 1.45, 1.53, 1.56, 1.6, 1.11 and 1.19 for ARM, PPC, NiosII (big and small configurations), LEON2 and LEON3 respectively. Performance results of the LEON3 board are fully consistent with the execution times analyses of the actual PRISMA on-board navigation software confirming the validity of the test here considered, in selecting a proper board for a satellite mission.

REFERENCES

[1] E. Gill, O. Montenbruck, K. Briess, *GPS-Based Autonomous Navigation for the BIRD Satellite*, 15th International Symposium on Space Flight Dynamics, France 26-30 June (2000).

[2] E. Gill, O. Montenbruck, The Onboard Navigation System for the BIRD Small Satellite, GSOC FB 2002-06 (2002).

[3] T. Bretschneider, S. H. Tan, Goh C.H., K. Arichandran, W.E. Koh, E. Gill, *X-SAT Mission Progress*, 5th IAA Symposium on Small Satellites for Earth Observation IAA-B5-0504, Berlin (2005).

[4] S. Persson, P. Bodin, E. Gill, J. Har, and J. Jörgensen, *Prisma An Autonomous Formation Flying Mission*, Sardegna, Italy, ESA Small Satellite Systems and Services Symposium, 25-29 September 2006.

[5] E. Gill, S. D'Amico, and O. Montenbruck, *Autonomous Formation Flying for the PRISMA Mission*, Journal of Spacecraft and Rockets, Vol. 44, May 2007, pp. 671–681.

[6] S. D'Amico, E. Gill, M. Garcia and O. Montenbruck, *GPS-Based Real-Time Navigation for the PRISMA Formation Flying Mission*, Noordwijk, 3rd ESA Workshop on Satellite Navigation User Equipment Technologies - NAVITEC, 11-13 December 2006.

[7] P.J. Mumford, K. Parkinson, A.G. Dempster, *The Namuru Open GNSS Research Receiver*, 19th Int. Tech. Meeting of the Satellite Division of the U.S. Inst. of Navigation, Fort Worth, Texas, USA, 2006.

[8] A. Grillenberger, R. Rivas, M. Markgraf, P. Mumford, K. Parkinson, C. Rizos, *The Namuru Receiver as Development Platform for Spaceborne GNSS Applications*, 4th ESA Workshop on Satellite Navigation User Equipment Technologies, NAVITEC'2008, 10-12 December 2006, Noordwijk (2008).

[9] O. Montenbruck, E. Gill, M. Markgraf, *Phoenix-XNS - A Miniature Real-Time Navigation System for LEO Satellites*, 3rd ESA Workshop on Satellite Navigation User Equipment Technologies, NAVITEC'2006, 11-13 December 2006, Noordwijk (2006).

[10] O. Montenbruck, B. Nortier, S.Mostert, *A Miniature GPS Receiver for Precise Orbit Determination of the SUNSAT2004 Micro-Satellite*, ION National Technical Meeting, 26-28 Jan. 2004, San Diego, California (2004).

[11] J. Gaisler, *The LEON Processor User's Manual*, V 2.3.7 Gaisler Research (2001).

[12] *RTEMS C Users Guide - Edition 4.6.5, for RTEMS 4.6.5*, On-Line Applications Research Corporation (OAR) - http://www.rtems.com, 30 August 2003.

[13] S. De Florio and S. D'Amico, Comparison of Scalar and Vectorial Implementation of the Extended Kalman Filter, PRISMA-DLR-TN-19, 2008.

[14] S. De Florio and S. D'Amico, *First Tests of the PRISMA Flight Software on LEON2 Board*, PRISMA-DLR-TN-14, 2007.

[15] S. De Florio and S. D'Amico, *Max-Path Unit Tests of the PRISMA Flight Software v. 6.0.0*, PRISMA-DLR-TN-17, 2007.

[16] S. D. Florio, S. D'Amico, and M. G. Fernandez, *The Precise Autonomous Orbit Keeping Experiment on the Prisma Formation Flying Mission*, Galveston, USA, 18th AAS/AIAA Space Flight Mechanics Meeting, 27-31 January 2008.

[17] S. D'Amico S., S. De Florio, J. S. Ardaens, T. Yamamoto, Offline and Hardware-in-the-loop Validation of the GPS-based Real-Time Navigation System for the PRISMA Formation Flying Mission, 3rd International Symposium on Formation Flying, Missions and Technology, 23-25 April 2008, ESA/ESTEC, Noordwijk (2008).