Frascati Physics Series Vol. XXXVII (2004) pp. 101–106
FRONTIER SCIENCE 2004, PHYSICS AND ASTROPHYSICS IN SPACE
Frascati, 14-19 June, 2004

THE CENTRAL PROCESSING UNIT OF PAMELA EXPERIMENT

F. ALTAMURA[a], A. BASILI[a], M. CASOLINO[a], M.P. DE PASCALE[a], M. MINORI[a], M. NAGNI[a], P. PICOZZA[a], O. ADRIANI[b], G. CASTELLINI[b,c], P. PAPINI[b], P. SPILLANTINI[b], M. BOEZIO[d], F. SEBASTIANI[e], G. ALFARANO[e], S. TASSA[e]

[a] *INFN, Sezione di Roma II - Dipartimento di Fisica, Università di Roma "Tor Vergata", Via della Ricerca Scientifica, Roma, Italy*

[b] *INFN sez. Firenze-University of Firenze, Firenze.*

[c] *IFAC-CNR, Firenze.*

[d] *INFN sez. Trieste-University of Trieste, Trieste.*

[e] *Nergal s.r.l., Roma*

Abstract

The Pamela experiment aims to measure with great precision the antimatter present in our galaxy in the form of high energy particles; also galactic, solar and trapped components of cosmic rays will be studied. The experiment will be placed on a Russian Resurs-DK1 satellite and launched into a 350 * 600 km orbit with and inclination of 70.4 degrees in the year 2005. The processing unit of the experiment is based on a ERC-32 architecture (a SPARC v7 implementation) running a real time operating system (RTEMS). The main purpose of the unit is to handle slow control, acquisition and store data on a 2 GB Mass Memory.

Communications between Pamela and the main satellite are done via a 1553B bus. Data acquisition from the detectors (TRD, Time of Flight, Magnetic Spectrometer, Electromagnetic Calorimeter, Anticoincidence system, Neutron detector, Bottom scintillator) is performed via a 2Mbyte/s interface. Download from the Pamela memory to satellite main storage is handled by a 16Mbyte/s bus. The daily amount of data transmitted to ground will be up to 20 GB/day. In this work we describe the CPU of the experiment and the general software scheme.

1 Introduction

The main objectives of PAMELA are the accurate measurements of the antiproton and positron fluxes with a sensitivity and statistics out of the reach of balloon-borne experiments. The energy range goes from below 100 MeV to above 100 GeV and the search of antihelium with a sensitivity better than 10^{-7} in antihelium to helium ratio[1]. The PAMELA telescope will be installed on board of the Russian Resurs DK-1 satellite and will be launched in the year 2005. The experiment is designed in an hierarchical - modular structure where the subdetectors handle all fast acquisition and data suppression before sending data to the acquisition board which interfaces them to the CPU. Data are then stored in the mass memory before transfer to the satellite. The intermediate boards have all their redundancy; all detectors are divided in section designed to continue working - even though with degraded performances - even in case of partial failures.

2 The Central Processing Unit

The Central Processing Unit of Pamela, is manufactured by Laben S.P.A. and handles all slow control, interaction with the satellite, data acquisition, storage and downlink. The CPU box is a space qualified system composed by:

- a central processor (CPU) ERC32 SPARC V7 with a clock of 24 MHz using a PROM of 128 KB for the boot, 4 MB of RAM, two banks x 512 KB of EEPROM.

- an avionics standard 1553B BUS interface board from/to the Resurs satellite. The CPU executes all the commands received from the Resurs satellite through this bus; these commands can be used to set the acquisition modes, reprogram the detectors or the CPU itself. This interface also acts as monitor for the Resurs to asynchronously control that the PAMELA experiment is properly running.

- a Pamela InterFace (PIF) board: a high speed board which transfers data from the detectors to the Mass Memory. Furthermore it contains two fast bus for Data Command communication with the detectors;

- a multipurpose Telemetry and Control board (TMTC) containing various digital, ADC/DAC interfaces to interact with the detectors and the subsystems of the experiment.

3 Software

The SPARC/ERC32 CPU is running RTEMS 4.0[2], a hard real time operative system, with microkernel model, that provides all primitives for multitasking, interrupt manager, timers, events, mailboxes, semaphores, signals, clock handling. The PAMELA Software Application is running on the top of the RTEMS Executive. RTEMS doesn't require system reserved tasks to run for itself. The Software handles communication from and to peripherals Front-Ends (FE) plugged to the IDAQ for the physical data event acquisition: format and send FE commands, wait for the answers, write them to the mass memory and, if needed, perform some check; this is what is typically done during the Initialization (include the program loading for the DSPs), Calibration and Acquisitions. The Software also manages changing of operative modes of peripherals according to the orbit position (proper macrocommands are provided by the satellite). Beside, the CPU software handles the programming of all subsystems and peripheral boards via a dedicated Housekeeping Board: Relay Board, Gas System Board and all acquisition boards: IDAQ, AC, Tracker sensor, Calorimeter, S4. The software also check and handles the power management: turn on and off power supplies of all boards and FE, manage main/spare switching and check for anomalies.

Telemetries (temperature, power, contact closure) coming from Telemetry interface are also periodically sampled and stored in the Mass Memory (MM) for post analysis on ground and in the telemetries format region of the 1553B interface. Beside, CPU also checks telemetry values in order to point out anomalies to perform certain actions. The Gas System software has to perform some main task: periodic container purge, periodic TRD purge , check for pressure and temperatures values to be in some ranges and beside handle all safety and recovery procedures. In the software eight tasks are running for low-level purpose (i.e. History Area, Macrocommand Dispatcher, File Manager (mass memory), House Keeping Manager, PIF Manager) running at high priority. In the application level we have four tasks at lower priority: Run Manager (acquisition and all interaction with IDAQ-PIF-MM), Pamela Manager (general control), Slow Control Manager (housekeeping and alarms) and Gas System Manager. The message exchange programming paradigm is used: tasks wait for a message from its own mailbox that other tasks or

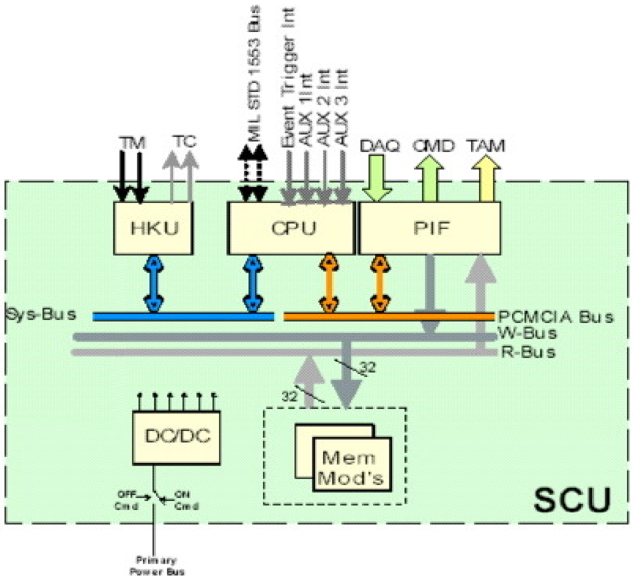

Figure 1: Scheme of the Pamela CPU and its block scheme with the Housekeeping board (HKU), the fast acquisition board (PIF), the memory module.

interrupts send. All the software, including the OS, is written in C, cross-compiled under an ordinary Linux/Intel PC using the ”sparc-rtems” porting of the GNU compiler and GNU tools (gdb,ld,mn,...). Remote debugging, loading and firmwaring could be made using gdb. Many other utilities and languages have been used in the development: UNIX programming, Perl, C++, CVS.

The PAMELA software is based on a multitask architecture. It largely uses the communication and synchronization features provided by the operating system. Data transfer and synchronization between cooperating tasks and/or ISRs is made using RTEMS tools like semaphores, events and message queues. The PAMELA Application is divided in three logical layers:

1. Drivers and Modules: this layer is in charge of managing the protocol of boards, the interrupt, communication bus 1553B, MM drivers, File System, etc..

2. Low Level Tasks layer: those tasks handle the high priority process like dispatching of the macrocommand and the accessing to shared resources like TMTC board.

3. High Level (or Application) layer: those task are in charge to handle the main flow of the application: manage the acquisition and FE's configuration, handle error and alarm condition, power supplies management, Gas System handles.

3.1 Tasks

PAMELA software uses a typical paradigm for multitasking[3]: operating system objects are all created at the software initialization (as soon as after bootstrap) and never destroyed. For each task the system creates the task objects with its own priority and its entry point (a function pointer) and the corresponding mailbox for this task that implements a message queue. A message queue (FIFO) allows the passing of messages among tasks and ISRs. It can contain a variable number of messages.

Low Level Tasks

Low level tasks in PAMELA interact with some resource providing a transparent functionality.

1. HKManager: provides interface to the TMTC and through it provides a high level interface to the Housekeeping Board.

2. ReportGenerator: reporting status of the software for the 1553B chip.

3. HistoryArea: provides logging functionalities to store datas in 1553B shared memory read by Resurs and while in debugging mode used for logging output to serial ports.

4. DiagSupervisor: the lowest priority tasks that make a scrubbing of the RAM in order to detect parity error and active the EDAC mechanism of the CPU.

5. MCMDDispatcher: the 1553B interrupt receives a macrocommand and sends it to this task that analyze the data structure of the macrocommand, dispatching it to the proper task.

6. PatchDumpManager: used for patching the software on the EEPROM and dumping arbitrary RAM or EEPROM data from ground.

7. MMSUManager: manage access to the MM registers in particular to program the WRITE DMA between PIF and MM.

High Level Tasks

Four high level task manages the main application process.

- PamManager: it is a task that receives most of the macrommands from the MCMDDispatcher and manages them. Beside it holds the main state machine of the PAMELA software and controls the behavior of the other two tasks: Run Manager and SCManager. As soon as one of these task has completed the requested operation it notifies to the PamManager the end of the operations with a return code and an optional error code.

- RunManager: it is the task for the acquisition process. Its goal is to assemble command queues for different FEs, store them into the PIF, send them to the Intermediate Board, wait for the answer, store results into the MM, detect some FE error condition and locally manage some of them.

- SlowControlManager: this task manages all monitoring signal, power supply, checks temperature, voltage, via TMTC and Housekeeping Board. Besides it checks and sets the correct power supply configuration after the boot process of PAMELA, handles the power on and power off of PAMELA and of FEs and manages recovery and error handling in case of anomaly.

- GasSystemManager: The Gas system is an electromechanical system whose operative modes are controlled by a driver board which is programmed and handled by the PSCU through the software gas task. This separate task is necessary because of the long duration of operations ($\simeq$hours) which involves filling or purging a container with gas, opening and closing electromechanical valves, measuring pressures and temperatures related to the various system components.

References

[1] Simon, M., XXVIII ICRC, OG 1.5, 2117 Tsukuba, Japan, 2003.

[2] RTEMS official documentation: http://www.rtems.com/onlinedocs/releases/4.0.0/doc

[3] Straumann, T. Open Source Real-Time Operating Systems Overview, "8th International Conference on Accelerator and Large Experimental Physics Control Systems", San Jose, USA, 2001.