An Autonomous CCSDS Packet Generating ASIC for Analog and Digital Telemetry

D. Reddy, F. O’Brien, M. Flynn, C. McSweeney, J. Damron, G. Maguire, B. Kinsella
Silicon Systems Ltd., 32-34 Harcourt St., Dublin 2, Ireland
tel: +353-1-4025800, fax +353-1-4025811

P. Sinander, C. Smith
Control, Data & Power division (TOS-ES), ESA, PO. Box 299, NL-2200 AG Noordwijk, The Netherlands
tel: +31-71-5656565, fax +31-71-5654295

Abstract

The Packetised Essential Telemetry Retrieval ASIC (PETRA) is a mixed analog/digital chip for retrieving sensor data on-board spacecraft, designed by Silicon Systems Ltd. One single PETRA can connect up to 40 digital and/or analog sensors which are periodically sampled. CCSDS Telemetry packets containing the data are automatically generated and can be delivered directly to the transfer frame generator or to an on-board processor. The packets are output serially, at a nominal baud rate of 9600bps or 115.2kbps, with the interface being directly compliant with both the Virtual Channel Multiplexer (VCA) and RS-232 type asynchronous serial interfaces.

The PETRA is a 0.8 μm double-metal, double-poly CMOS mixed signal device with a die size of 26 mm^2. The dynamic power dissipation is 80mW with a static dissipation of 30mW.

This paper introduces the advantages of using the PETRA in a spacecraft’s Data Handling System (DHS) compared to a traditional implementation. An overview of the PETRA functionality is presented, followed by a description of the device architecture. An outline of the design methodology is then followed by a discussion of the PETRA demonstration system.

1.0 Introduction

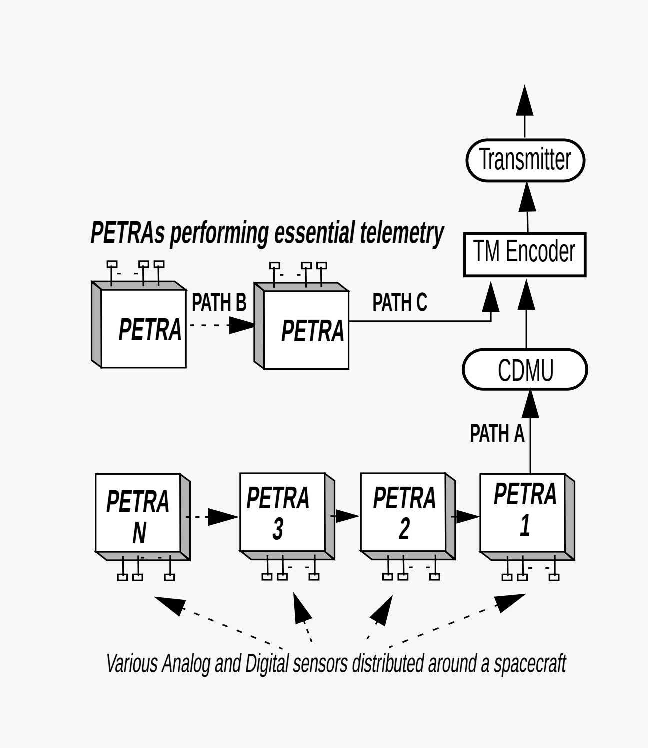
The spacecraft Data Handling System [1] plays a central role in the interaction between on-board sub-systems and the ground station. A standard DHS includes a Central Data Management Unit (CDMU), a data-handling bus, a Telemetry Encoder, a Telecommand Decoder, a receiver and a transmitter [2]. To acquire telemetry data, the CDMU communicates with Remote Terminal Units which can have large numbers of analog and digital sensors (up to a thousand) connected to them. The CDMU formats the data from each Remote Terminal Unit into packets which are transmitted to a ground station as telemetry frames via a radio-frequency link. Such a system requires a large amount of wiring routed from sensors all over the spacecraft to the centrally located RTUs. In addition, the system does not allow for easy access to the spacecraft status in the event of abnormal conditions (e.g. power disruption, CDMU not operational, etc.).

In contrast, the PETRA autonomously acquires telemetry data, without being dependent on a centralized DHS. The single-chip PETRA allows data acquisition to be performed locally, resulting in a reduction of mass, wiring and power consumption. The PETRA is ideal for gathering essential telemetry data, ensuring that the data is provided to the ground station even during emergency situations. However, the PETRA is not only intended for emergency operation; it is also suitable for telemetry data acquisition during normal operation, with the sensor data being provided via the CDMU.

The PETRA allows for three forms of data communication (see Figure 1):

- communication with the DHS using the RS-232 type serial interface. In this case the PETRA acts like a Packet Remote Terminal, providing the data to the DHS which reformats it and transmits it to ground, or alternatively consumes it on-board. (Path A)
- serial communication with other PETRAs using the VCA compatible interface. An arbitrary number of devices can then be cascaded in a chain, increasing the amount of sensor data provided over one serial link. (Path B)

- direct communication with the Telemetry Encoder ensuring immediate determination of the spacecraft status after power on. In this case the PETRA retrieves “essential telemetry” independently of the CDMU, and only the Telemetry and Transmitter subsystems need to be operational for the spacecraft status to be determined on ground. The essential telemetry should for example include power status, critical system status and whether telecommands are being received and being acted upon. (Path C)



Transmitter
PETRAs performing essential telemetry
TM Encoder
PETRA
PATH B
PETRA
PATH C
CDMU
PATH A
PETRA N
PETRA 3
PETRA 2
PETRA 1
Various Analog and Digital sensors distributed around a spacecraft

Figure 1: The PETRA in the Data Handling System

2.0 Functional Overview

The PETRA is essentially a multiplexer for analog and digital signals which presents the sampled data in the format of an ESA/CCSDS telemetry packet [3] as shown in Figure 3. It has 40 discrete inputs which can be configured as digital (maximum of 40) or analog (maximum of 32) with selectable voltage ranges of 0-1V or 0-4V. All enabled inputs are scanned in a fixed sequence and can be sampled over one of twelve different *scan periods* (10ms, 20ms, 50ms, 100ms, 200ms, 500ms, 1s, 2s, 5s, 10s, 50s). A scan period is defined as the periodic rate at which each input is scanned once. Selection of the scan period and baud rate is programmed by setting the BAUD_SEL and SCAN_PER[0:2] inputs pins. At the nominal frequency of 4 MHz, data is output in serial format, at a baud rate of either 9600 bps or 115.2 kbps. All digital samples are sampled together at the beginning of a scan period, whereas analog sampling is evenly distributed over this time, as shown in Figure 2.

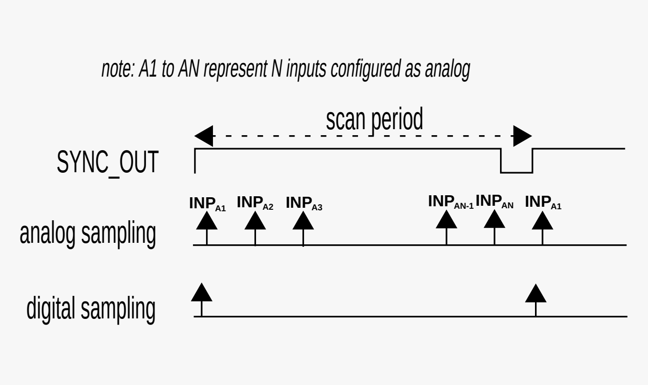


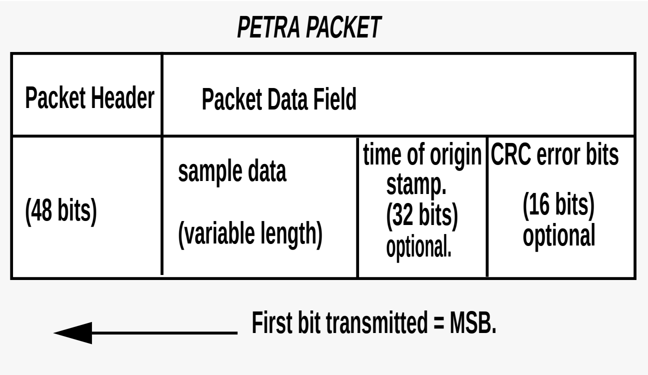
Figure 2: Data sampling over a scan period

The PETRA has an on-chip 10-bit ADC for conversion of the analog samples over the 0-4V input range. Data acquired in a scan period is placed in an ESA/CCSDS compliant packet when being transmitted. This packet is transmitted via the Virtual Channel Assembler (VCA) [4] and RS-232 type output interfaces simultaneously. The VCA is a member of the TM Encoder chip set.

As a spacecraft generally requires a large number of sensors, several PETRAs can be connected in a chain configuration to increase the number of inputs, with each PETRA providing a separate packet or part of a packet. Every PETRA in a chain is associated with an unique Application Identifier which can be one of two fixed values or programmable. This Identifier is unique for each separate process or application being monitored in the chain. A PETRA is in *merge mode* if its MERGE input is asserted and if it receives a packet with an application ID which matches its own. It will then append its own data to complete the packet for that application/process. This process of merging packets for the same application reduces the overhead of separate packet headers.

In addition, to further reduce overheads on the TM bandwidth, applications being monitored which rarely change can be monitored in *event driven mode*. With this option the scan period is fixed to 10ms and the high baud rate is selected. However, a packet is only output when any one of the digital sample inputs have changed with respect to their previously saved value and, by default, once every 4096 scan periods (approximately every 40s). This allows high time resolution when events occur, combined with low packet generation rate during quiet periods. This is particularly useful for reducing TM bandwidth previously clogged up with data which never changes and also if this information is being used on board, it would reduce the load on the the CDMU.

The PETRA supports an option for generating a Sample Packet Error Control Field at the end of every packet transmitted. This is a 16 bit CRC code calculated over the whole packet including the header. When enabled, this error control field occupies the last two octets of the packet.



PETRA PACKET

Packet Header	Packet Data Field		
(48 bits)	sample data (variable length)	time of origin stamp. (32 bits) optional.	CRC error bits (16 bits) optional

Figure 3: A PETRA Packet

Inherent in all packet systems are buffering delays. To calculate the exact sampling time of the data the PETRA has an option to produce a "time of origin" stamp in each packet. The PETRA incorporates its own basic 32-bit time counter, allowing it to provide time stamps even when the central spacecraft time subsystem is not operative. This time count corresponds to the absolute time from the occurrence of a time strobe (provided by the transfer frame generator) until the start of the next scan period for the data in the packet, allowing the absolute time for the start of this scan period to be determined.

3.0 Architecture

Without plunging into all the details of the architecture and design, this paper will look at the basic data flow through the PETRA (see Figure 4) under the following headings:

- Internal Packet Generation
- External Packet Parsing
- Output Interfaces
- Control and Arbitration.

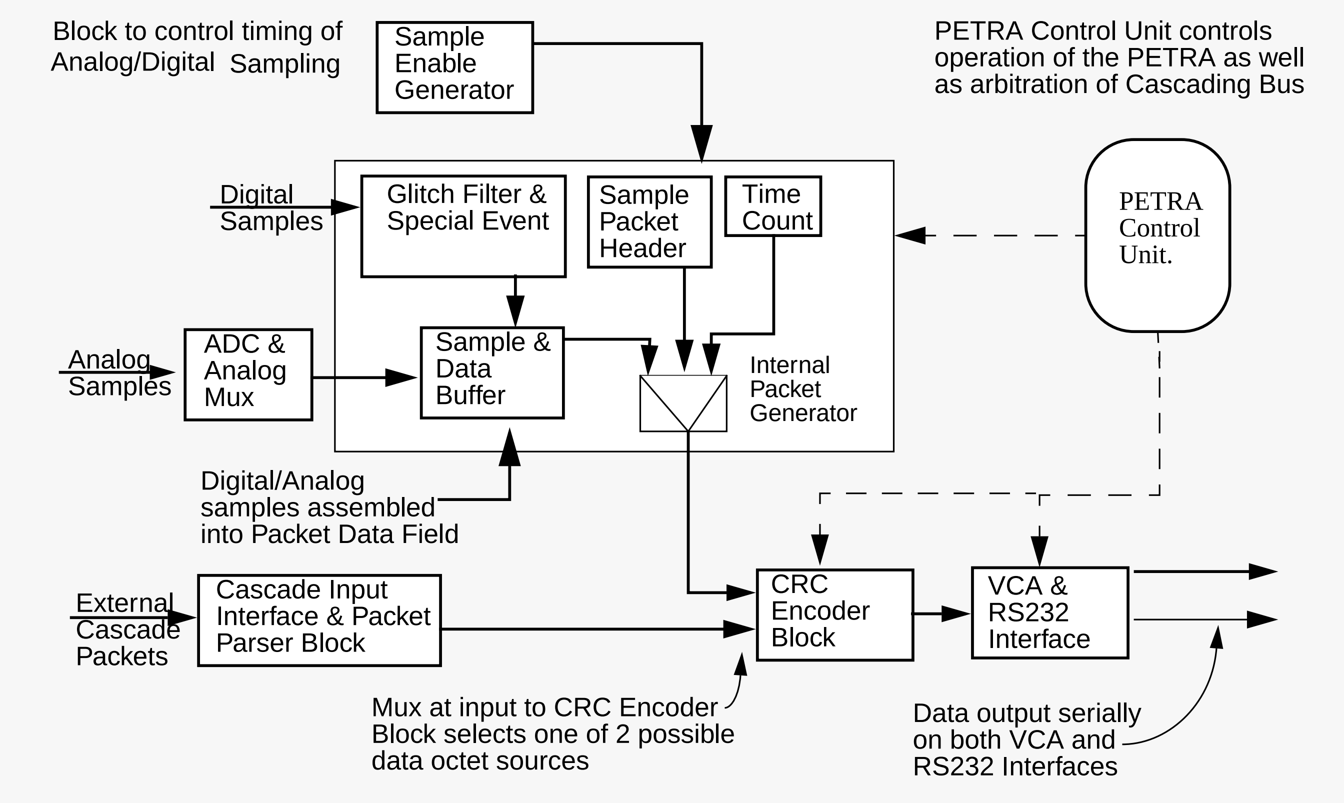
3.1 Internal Packet Generation

This section of the PETRA is responsible for sample gathering (analog and digital) and packet generation. The sample enable generator controls the exact times at which digital and analog sampling is carried out. Sample data is acquired in one scan period and is transmitted at the beginning of the next. A PETRA's scan period can be viewed externally on SYNC_OUT, which is an indication that the sampling process is ongoing. During a scan period, all the digital inputs are passed through anti-glitch filters before sampling the data into the *Sample Buffer*. Analog samples are selected using a 32:1 analog multiplexer before being converted in an A/D into digital form and then saved in the sample buffer. The A/D converter is a 10 bit device that can potentially handle up to 12 bit operation. The Internal Packet Generator combines the Analog and Digital samples (i.e. the Packet Data Field) and stores them in the *Sample & Data buffer* in correct order ready for transmission.

Before transmission the PETRA generates a header (six octets long), which is carried out in the *Sample Packet Header* sub-block. The header contains information about the packet, i.e. the PETRA's Application ID, how many scan periods have occurred since power on, and the length of the packet. Also within the domain of the Internal Packet Generation is the generation of the time stamp associated with the current packet. This is carried out by a 32-bit time counter which increments by one for every system clock cycle after the assertion of the TIME input pin until the first sample of the next scan period has been taken. The data octets from the Time Count sub-block, Sample Packet Header Sub-block, and Sample & Data Buffer sub-block are then passed into a multiplexer. The octets are then chosen in correct order and output to the CRC encoder block before being output on the VCA and RS-232 type interfaces.

3.2 Cascading and Packet Merging

When two or more PETRAs are connected in a chain configuration they can communicate with each other via a Cascade Interface which is part of the *Cascade Input Interface and Packet Parser* block. Here external packets are input serially and are assembled into octets before the being parsed. The serial inputs SIN and SCLK_IN are registered twice for metastability reasons. The parser reads the header of the external packet and decides what must be done with it. In particular if the PETRA has MERGE asserted, it will compare the Application Identifier of the incoming packet



Block to control timing of Analog/Digital Sampling
Sample Enable Generator
PETRA Control Unit controls operation of the PETRA as well as arbitration of Cascading Bus
Digital Samples
Glitch Filter & Special Event
Sample Packet Header
Time Count
PETRA Control Unit.
Analog Samples
ADC & Analog Mux
Sample & Data Buffer
Internal Packet Generator
Digital/Analog samples assembled into Packet Data Field
External Cascade Packets
Cascade Input Interface & Packet Parser Block
CRC Encoder Block
VCA & RS232 Interface
Mux at input to CRC Encoder Block selects one of 2 possible data octet sources
Data output serially on both VCA and RS232 Interfaces

Figure 4: PETRA Functional Block Diagram

with its own Application ID. If the Application IDs match it will merge the data in the Sample & Data buffer with the external packet and adjust the length field of the external packet before transmission.

3.3 Output Interfaces

Once the internal packet has been generated and the External Packet has been parsed (if it exists), the serialised octets are passed to the CRC block. If the CRC input is asserted, this block will generate a 16-bit CRC code for the entire packet (Header Field, Data Field and Time count field if included). The final serialised octets are then passed to the *VCA & RS-232 type interfaces* from which they are transmitted through the VCA interface and the RS-232 type interface using one of two selectable baud rates. Data octets received at a lower baud rate than the PETRA's are output at the PETRA's own baud rate. For small percentage differences between the respective baud rates, the length of the 'stop' bit is adjusted to compensate.

3.4 Control and Arbitration

The *PETRA Control Unit* is responsible for co-ordinating the functions of the various blocks, arbitration on the cascading bus, error control on the Cascade Interface, and co-ordinating the internal and external data flow. Its main function is to ensure that data is transmitted only at the correct times and according to the rules of the arbitration scheme that the PETRA obeys. The arbitration scheme, built into each PETRA, determines if it is allowed to transmit its own (PETRA) packet, if it should give priority to a packet (Cascade Packet) arriving from an *upstream* source or if a *downstream* source (higher priority) should transmit first (see Figure 5). An upstream source is defined as a PETRA or other VCA compatible device which is further away from the TM Encoder than the current PETRA. The data flow is from the upstream device towards the TM Encoder. Devices closer to the TM Encoder are known as downstream devices.

This block also controls the switchover from internal to external transmission (and vice versa) in cascade and merge modes and controls the flow of telemetry data within the PETRA through generation of internal enable signals. Another important function of the *PETRA Control Unit* is to protect the PETRA against any errors which might occur at the Cascade Interface, and also highlighting the arrival at the Cascade Interface of the first and last octets of an external data packet.

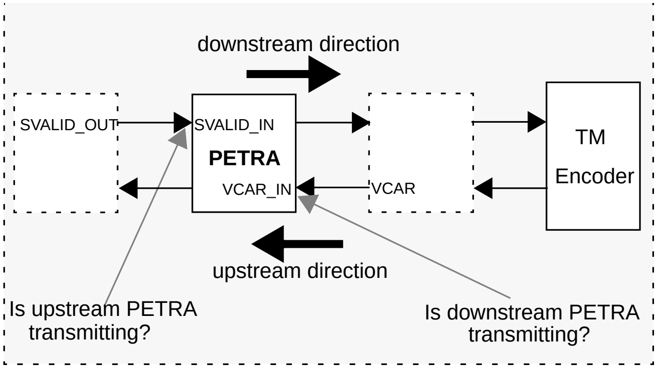


Figure 5: The PETRA data flow

4.0 Design Methodology

The PETRA was designed by SSL based on an initial specification [5] produced by the Microelectronics section at ESTEC (TOS-ESM), exploring the possibilities offered by mixed analog/digital technology. It is part of an ongoing ESA program to create a commercially available family of standard ASICs to reduce costs and development time while improving the performance of the data handling systems using the ESA/CCSDS standards. This initial specification was subsequently refined into the PETRA Functional Specification by SSL.

The digital part of the PETRA was implemented in VHDL RTL code and was kept as synchronous as possible. This code was simulated with the Cadence Leapfrog VHDL simulator. The chip was then laid out with Cell Ensemble. No floating nodes exist in the PETRA as the digital design was implemented with static standard cells. The entire analog front-end was implemented in Spectre and eldo. Verification of the analog and digital interface was carried out by running mixed mode simulations in SpectreVerilog. The full PETRA design was verified for design rule checks (DRC) and Layout versus schematic (LVS) match using the Cadence DIVA tool.

The design allows the possibility of scan insertion for production testing. The netlist was taken through scan insertion with Sunrise, but since the area increased by 15% scan testing was abandoned. A gate level verilog netlist of the design was generated and verifault was used to verify a 90% fault coverage without scan, which was considered acceptable for a demonstrator not intended for flight. At this point the digital design was also mapped to an ALTERA Flex 10K FPGA using Synopsys and functionality was tested and verified using the demonstration system.

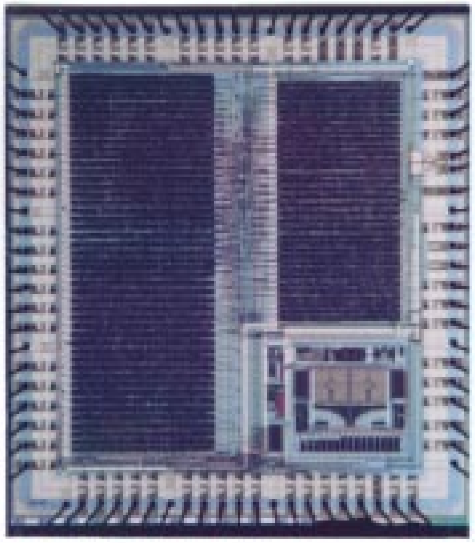


Figure 6: Photo of the PETRA die

After the PETRA Functional Specification had been finalized, the design from RTL implementation to final layout took eight months. The digital complexity is 13K gates. The PETRA is implemented in the AMS 0.8 µm double-metal, double-poly CMOS technology provided through the Europractice multi-project wafer service, with a die area of 26mm^2 (see Figure 6).

5.0 Demonstration System

SSL has also developed a PC-based demonstration system for the PETRA (see Figure 7). The overall requirement was for a user-friendly, flexible system capable of demonstrating as much of the PETRA functionality as possible in an environment emulating a real application. There are three basic elements to such a system:

- Physical hosting capability for the PETRA chip,
- Ability to stimulate analog and digital inputs,
- Software to acquire, analyse, store and display the packet data.

Our solution involved the design of a multi-layer PCB to host one PETRA device. DIP switches allow the PETRA to be configured into different operational modes. RS422 transceivers are provided to interface with an upstream and a downstream PCB (and their associated PETRAs). This allows the user to exercise an individual PETRA or a chain of PETRAs. There is also an RS-232 transceiver to send the data packets to a PC or other computer. In the demonstration system,

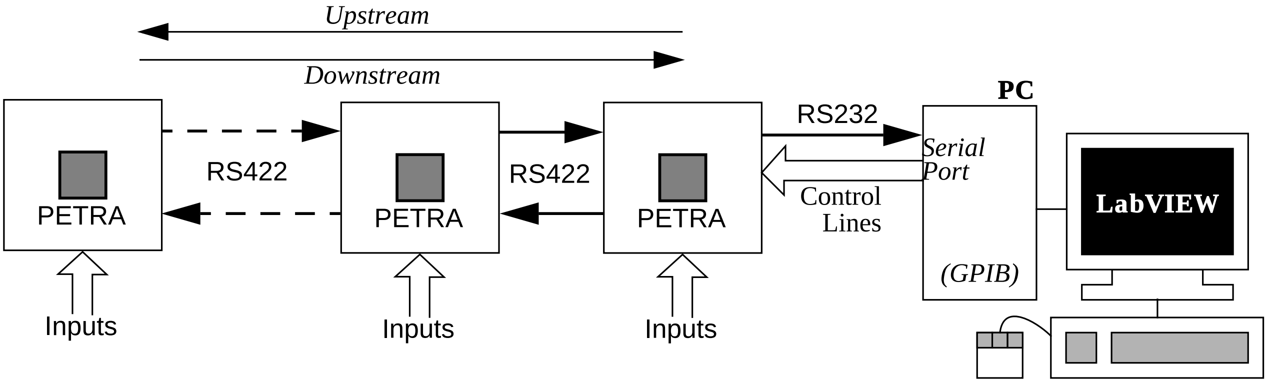


Figure 7: Demonstration of a chain of PETRAs

the PC replaces the TM Encoder, downlink, ground station and operator display.

The inputs to each PETRA may be driven directly through connectors on the board. A range of plug-in daughter-boards with switches, potentiometers and other sensors allows the user to stimulate the digital and analog data inputs.

LabVIEW® (from National Instruments) was used to create a Graphical User Interface for the demonstration system. Essentially, the software reads the packet data received via the RS-232 port and optionally saves it to disk. Once the user has entered the configuration of each PETRA, the software can automatically interpret the packets into their constituent header and data fields. The individual samples are then converted into engineering units, checked for alarm conditions, and displayed in a variety of formats.

6.0 Conclusions

A mixed signal prototype for data acquisition and packet generation has been implemented to meet CCSDS and ESA Packet Telemetry Standards. The chip requires no additional software or processor to operate and can reduce the load on on-board processors in *event driven mode*. The ability to communicate directly with the TM encoder allows the PETRA to operate immediately after power restoration and it is ideal for acquiring vital telemetry data without relying on a centralized Data Handling System. The PETRA was designed for full compatibility with the current generation of DHS systems while improving telemetry and sensory capabilities for future demands.

References

1. Chris Plummer, "*Data Handling System Standard,* ESA PSS-04-201", Draft 1.0f, European Space Agency, February 1996
2. Peter Sinander, "*Components for Building Telecommand and Telemetry On-board Systems"*, WDN/PS/1251, European Space Agency, 1994
3. Standards Approval Board (STAB) for Space Data Communications, *"Packet Telemetry Standard",* ESA PSS-04-106 Issue1, European Space Agency, January 1988
4. "*VCA HAF_12399 Preliminary Data Sheet"*, ABB Hafo (now MITEL Semiconductor), January 1994
5. Peter Sinander, "*PETRA, Packetised Essential Telemetry Retrieval ASIC, Initial Specification*", WSM/PS/140, Issue 1.1, European Space Agency, 29 March 1996
6. "PETRA Preliminary Data Sheet", rev 1.1, January 1998, Silicon Systems Limited