TELEMETRY ACQUISITION BOARD INCLUDING REED-SOLOMON FPGA DECODER FOR SPACE APPLICATIONS

François Lassère
Centre National d'Etudes Spatiales
18 Avenue Edouard BELIN
31401 TOULOUSE Cedex 4

Max Ferréol
Centre National d'Etudes Spatiales
18 Avenue Edouard BELIN
31401 TOULOUSE Cedex 4

Jean-Pierre Rocher
Centre National d'Etudes Spatiales
18 Avenue Edouard BELIN
31401 TOULOUSE Cedex 4

ABSTRACT

In order to reduce the ground segment equipment cost for small space missions, the French national space center (CNES) had the need to develop a CCSDS down-link interface board for low telemetry rates (< 1.5 Mb/s).
This board performs frame synchronization and Reed-Solomon decoding.
An important part of this design was the Reed-Solomon decoder development. In order to maintain low recurrent cost for this board, this decoder was realized in FPGA technology. Reed-Solomon decoding function, interleaving function (from 1 to 5) and virtual fill management are included in the same component. All set-up parameters are software programmable via the PCI bus, data and status are also available via the PCI bus under windows NT operating system.
This paper presents the main features of this board and an overview of the Reed-Solomon decoder development.

KEY WORDS

CCSDS, Reed-Solomon, FPGA, PCI bus.

INTRODUCTION

The aim of this realization is to propose a low cost CCSDS front-end equipment for our future ground segment developments.

To achieve this goal the design phase tried to:

- integrate the down-link interface functions in base-band telemetry, into data-processing environments as PC compatibles or workstations using PCI (Peripheral Component Interconnect) standard bus,

- use to the maximum electronic programmable components technologies (FPGA), less expensive and more evolutionary than ASIC technologies.

This, in order to provide to end users, an easy to operate interface that allows them to meet different or changing space mission requirements.

It was thus designed as an open system allowing the development of different kind of applications, for that this telemetry interface board fulfills the obligatory basic functions (frame synchronization) or time CPU expensive functions (Reed-Solomon decoding), before transferring the telemetry data, via PCI bus, towards the computer which will process the secondary treatments. The application developed today is installed on a PC platform under Windows NT environment.
This development was made in cooperation with a small French company (ELFES electroniques - 10 impasse Borde-Basse 31240 l'Union - France)

FUNCTIONAL REQUIREMENTS

The functional requirements of the board are as follows :

- input signals adaptation,
- frame synchronization,
- data de-randomization,
- CRC check,
- Reed-Solomon decoding,
- PCI bus interface.

All functions are programmable via PCI bus.

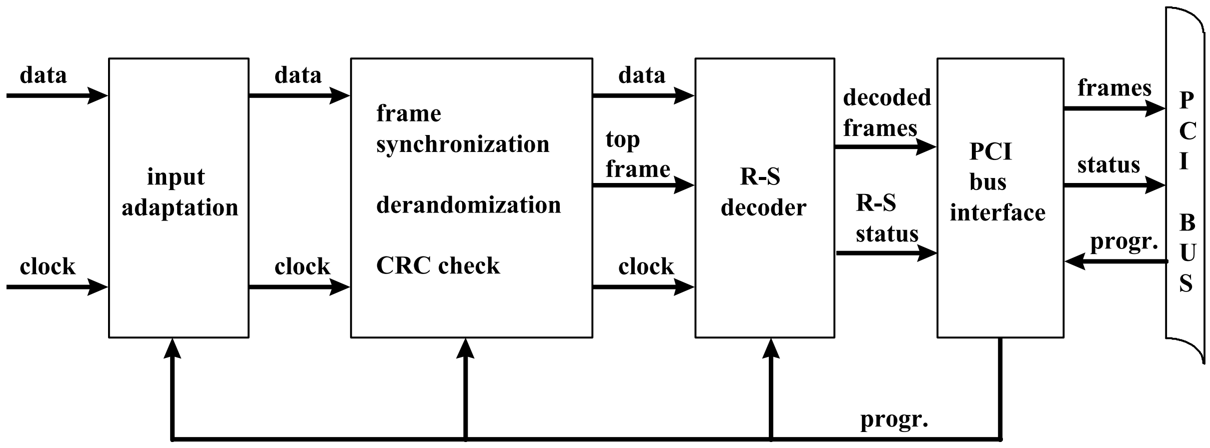

figure 1 : Board block diagram

INPUT ADAPTATION

The input impedance is adaptable and clock phase is configurable. The input adaptation module performs if necessary the NRZ-M to NRZ-L conversion.

FRAME SYNCHRONISER

The frame synchronization uses a classical 3 states strategy : search, verify and lock. For each synchronized frame, the associated status are sent via PCI bus.
It is possible, either to select the automatic polarity search, either to fix this polarity in normal or opposite mode.

DATA DE-RANDOMISER

A configuration parameter allows to select or not this function. This function is exactly the CCSDS [1] recommended function. The de-randomization function is applied immediately after the frame synchronization: a reference pseudo-random sequence is applied to each bit of the transfer frame through an exclusive OR (starting with the first bit of the transfer frame following the synchronization word).

CRC CHECK

A configuration parameter allows also to select or not this function. The generating polynomial of this CRC is defined by CCSDS [1] standard.
This control is generally used when the R-S decoding function is not retained.

REED-SOLOMON DECODER

This function was realized in a single programmable component (FPGA from ALTERA) which, in addition to the error correction function itself, carries out the operations of de-interleaving at the component input and re-interleaving at the output (for depths from 1 to 5), as well as the virtual fill if the block lengths are non-standard.

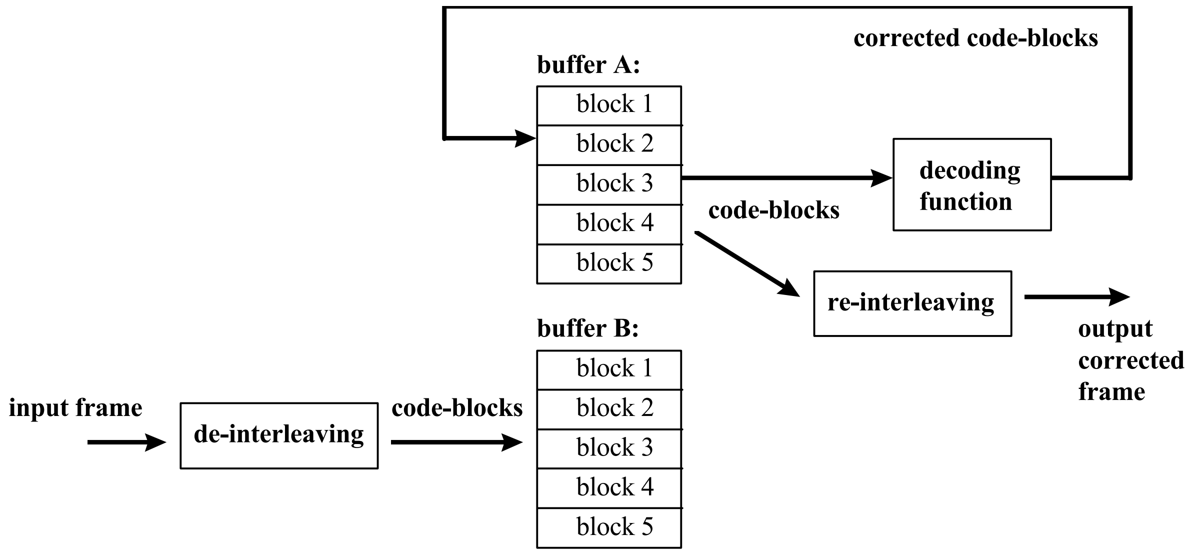

figure 2 : Component block diagram

INPUTS/OUTPUTS:

The memory zone for inputs/outputs is organized in 2 identical parts, each part being divided into 5 pages of 255 bytes each one. This organization allows, during the acquisition of input data, to reconstitute each block of data (code-block), by taking interleaving into account and by gathering the block data in a memory page which will be used as input of the R-S decoding function. Each code-block, once corrected (if necessary), will be rewritten in its corresponding memory page.
After each frame decoding, the output frame is rebuilt starting from the different code-blocks constituting it.
So, a half of the memory is used for acquisition of the frame presented to the circuit input, while other half is used for the processing of the previous received frame.

DECODING FUNCTION :

The main functions necessary to the realization of Reed-Solomon decoding are:

- syndromes calculation,
- determination of the error locator polynomial and error evaluator polynomial,
- search of the roots for these polynomials, these roots are used to determine the error addresses and values,
- errors correction.

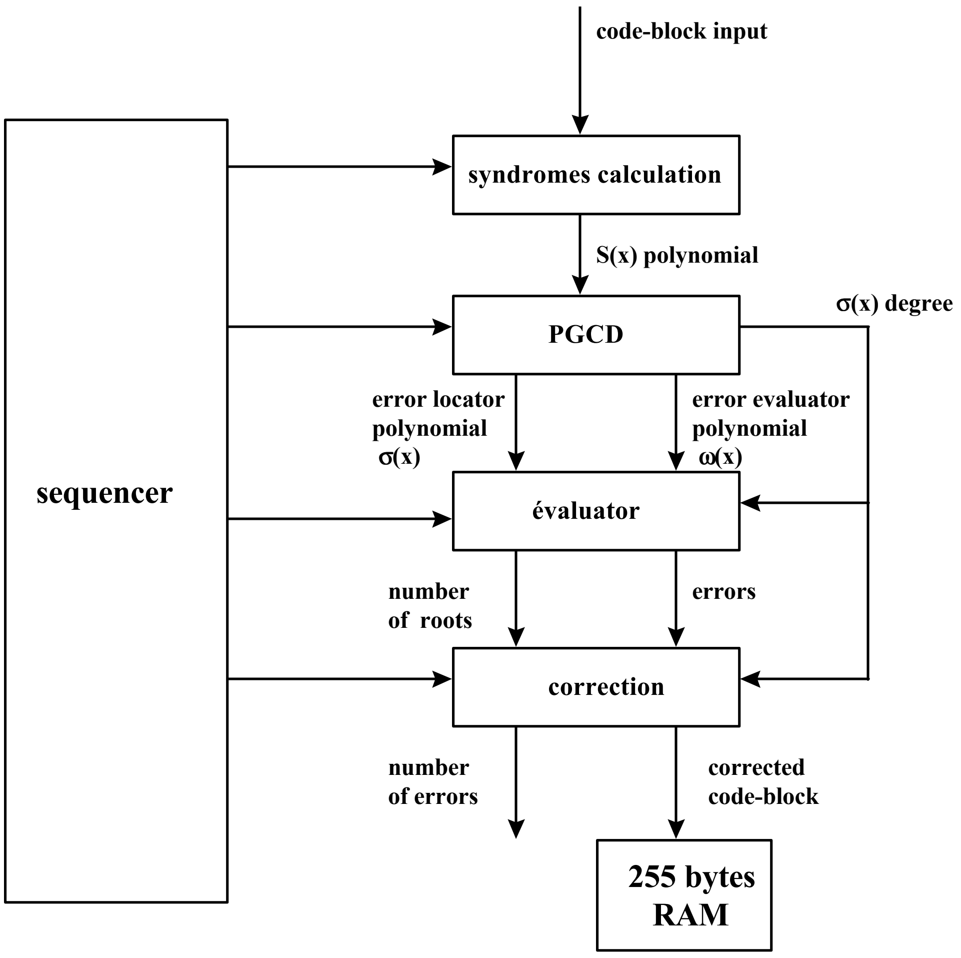

figure 3 : decoding function architecture

A sequencer controls the course of calculations in each phase of the process.

Role of the various functions:

Syndromes calculation :
calculation of the 32 syndromes and counting of the null syndromes number in order to respect the Reed-Solomon code correction capacity.

PGCD calculation :
determination of the error locator and evaluator polynomials ($\sigma(x)$ and $\omega(x)$), by means of the algorithm known as PGCD algorithm, then calculation and storage of the $\sigma(x)$ degree.

Evaluator:
Roots determination for $\sigma(x)$ and $\omega(x)$ and calculation of the error values. During the evaluation process, the roots number of the error locator polynomial is entered. This number is compared with the degree of this polynomial (provided by the PGCD calculation module). The block is regarded as being able to be corrected only if these two values are identical.

Correction:
counting of the roots number, errors correction and storage of the corrected block in a 255 bytes temporary buffer.

Cases when the block is not corrected:
- the polynomial degree $\sigma(x)$ is different from its roots number,
- the number of null syndromes is between 16 and 31, (it corresponds to an error byte number superior to 16 and thus exceeds the correction code capacity),
- block without error (32 null syndromes).

After correction the values are systematically stored in a buffer. If the requirements to authorize the correction of a code-block are met, the content of this buffer will replace the code-block received initially in the input/output memory, in the opposite case the correction buffer content is not used and the block initially acquired is not modified.

Sequencer:
signals generation controlling the various stages of decoding.

SET-UP :
Operating conditions definition (depth of interleaving, presence of virtual fill, decoding function active or not), requires to program some parameters. For that, a configuration programming byte must be applied to a specific component input (PROG). There are 2

programming addresses defined by signal CONF_ADR, according to the value of CONF_ADR the bits significance for the configuration byte is the following one:

CONF_ADR = 0	bit 7	:	0 : RS active 1 : RS inactive
	bit 2 to bit 0 :		interleaving depth value: (I-1) bits 3 to 6 are unused.
CONF_ADR = 1	bit 7 to bit 0 :		virtual fill zone size in each code-block.

Virtual fill :
When the frame length is shorter than the standard length (I x 255), during the calculation of the redundancy bytes, each code-block size is brought back to the standard size (255 bytes) while adding null bytes at the beginning of the code-block, these bytes not being transmitted, it is necessary to temporarily add them during the decoding process: this is the virtual fill.
The value programmed here corresponds to the virtual fill size zone for a code-block.
This value is computed starting from the frame length, in the following way:

(standard frame length - real frame length) / I

which can be also written:

255 - real frame length / I
Note: The use of virtual fill zones reduces the component performances since the processing time remains the same, whereas the acquisition time is reduced.

STATUS :
A status byte is provided after the last byte of each frame at the component output. This byte can take the following hexadecimal values:

0	: frame without error.
7F	: frame including code-blocks which could not be corrected.
80	: the input clock data rate is too high compared to the internal processing rate of the component.

Value range between **0 and 7F**: total number of corrected errors in the frame (sum of the corrected bytes in each block composing the frame).

PERFORMANCES:
For a 255 bytes code-block, received at a given rate, one needs : 255 x 8 = 2040 cycles of the input data clock to acquire a complete code-block.
The time needed by the decoding process of this code-block is : 8079 cycles of the internal processing clock.
It is obvious that to have enough time to decode a block during the acquisition of another block, one needs an internal processing clock 4 times faster than the input acquisition clock .

We made the choice of a processing clock at 5 Mb/s, therefore the maximum rate acceptable for telemetry at the component input will be 1.25 Mb/s.
Though this design can accept higher rates, these performances are sufficient for the projects interested by this component today.
If, for future projects, the need for higher rates appears, it will be necessary to study the possibilities for improvement of these performances.

Note: The treatment time for one frame includes the decoding time for each code-block composing the frame and, at the end of decoding process, the time necessary to emit this frame at the component output. In order to reduce the treatment time as much as possible, the frames at the component output are emitted byte by byte at internal processing clock rate (5 Mo/s). The frames thus appear at the component output in a discontinuous way (each data frame being separated from the following by the correction cycle time). Redundancy bytes are also provided at component output, so the user has the choice to use or not these bytes (example : case of double decoding at ground level).

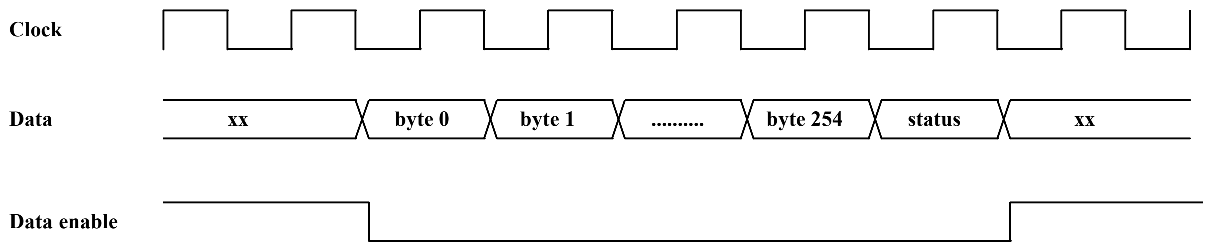

figure 4 : output data timing

CONCLUSION

With this development we have today a low cost and small size interface board for CCSDS telemetry. This board is able to satisfy the needs for a great number of our space projects.
This approach shows in particular the interest of technologies such as programmable electronic components even for the realization of voluminous and complex functions like Reed-Solomon decoding.

REFERENCES

[1] Telemetry channel coding, Consultative Committee for Space Data Systems, CCSDS 101.0-B-3, Blue book, May 1992.