RESCUE
A RADIATION HARD CCSDS REED-SOLOMON AND CONVOLUTIONAL ENCODER DEVICE

Sandi Habinc

European Space Agency, TOS-ESM, Postbus 299, NL-2200 AG Noordwijk, The Netherlands
sandi@ws.estec.esa.nl

Abstract

With increasing telemetry data rates, there is a need for faster Reed-Solomon and Convolutional on-board encoders than currently exist, both as self standing components and as integrated parts of future single chip telemetry systems. A radiation hard CCSDS Reed-Solomon and Convolutional Encoder (RESCUE) device has been developed along this line to meet current and future demands, being based on previously developed encoder cores.

1 INTRODUCTION

A telemetry channel is normally encoded to gain a higher data throughput at the same bit error rate as for uncoded transmission, but with less energy expended per information bit. The encoding allows the ground segment to perform error detection and correction on down-link data. Reed-Solomon and Convolutional encoding have been used widely on European spacecraft and are endorsed by European Space Agency (ESA) and Consultative Committee for Space Data Systems (CCSDS) standards. The same standards are now being revised also to allow Turbo encoding, being mostly foreseen for scientific deep-space missions but also for some specific near Earth communication scenarios. However, cascaded Reed-Solomon and Convolutional encoding will remain the main coding technique to be used on ESA spacecraft.

The MA1916 Reed Solomon and Convolutional encoder device (Ref. 7) from GEC Plessey Semiconductor (GB) has been flown on several spacecraft (e.g. SOHO) and will further be used on XMM and Integral. With the manufacturing of the MA1916 device being discontinued, several ongoing spacecraft developments were left without an alternative replacement. The development of a new encoder named RESCUE was therefore initiated. This new device is planned to be utilised both on commercial and scientific spacecraft such as Proba.

The intention with the radiation hard RESCUE device development has not only been to provide a replacement for the discontinued MA1916 device, but also to increase the functionality and raise the performance. The objective has not been to achieve full functional or timing compatibility between the two devices, but rather to develop a successor that could in most cases replace the MA1916 device with no or only minor modifications to existing board designs.

The RESCUE device is based on two encoder cores previously presented in (Ref. 11). These cores were modelled in synthesisable VHDL (Very high speed integrated circuit Hardware Description Language). The objective of the encoder core development was to create technology independent Reed-Solomon and Convolutional encoders adhering to ESA and CCSDS standards. The intention now is to use the encoder cores as a virtual second source for the RESCUE device, or other devices that might incorporate them, allowing fast transfer to another foundry if necessary. The initial encoder cores were pre-developed by ESA before being finalised by industry.

The RESCUE design was performed by SMARTECH (FIN) under the prime contractor Dornier Satellitensysteme (D). The device is manufactured by MITEL Semiconductor (S) in their radiation hard CMOS/SOS5 technology.

2 IMPLEMENTED STANDARDS

To provide the reader with an understanding of what the RESCUE device implements, the relevant ESA and CCSDS standards have been summarised hereafter.

The ESA Telemetry Channel Coding Standard (Ref. 1) specifies a Reed-Solomon block code with 8 bits per symbol, 255 symbols per codeword: first 223 symbols containing the information symbols and last 32 symbols constituting the check symbols. For transfers with more than 223 octets, codeblocks formed by interleaving of 2 to 5 or 8 codewords are specified. Shortened codeword lengths may be obtained using virtual fill. The CCSDS Telemetry Channel Coding standard (Ref. 2) allows only interleave depths 1 to 5.

The ESA Packet Telemetry Standard (Ref. 3) allows the standard codeblock lengths of 892, 1115 and 1784 octets when using Reed-Solomon encoding, corresponding to the interleave depths 4, 5 and 8. Shortened codeword lengths may be obtained using virtual fill. The CCSDS Packet Telemetry Blue Book (Ref. 4) allows however the same interleave depths 1 to 5 as (Ref. 2).

The CCSDS Advanced Orbiting Systems Blue Book (Ref. 5) allows interleave depths 1 to 5. It also allows shortened codeblock lengths to accommodate compatibility with 32-bit microprocessor systems, where the number of suppressed symbols must be a multiple of the interleave depth, i.e. all codewords are of equal length. The MA1916 device used to support interleave depths 1, 4 and 5 as per (Ref. 7). The Virtual Channel Multiplexer (VCM) device supports interleave depths 1, 2, 4 and 5 as per (Ref. 8).

With the RESCUE device supporting interleave depths 1 to 8 and virtual fill with the number of suppressed symbols being a multiple of the interleave depth, the above standards are all covered by a single radiation hard device.

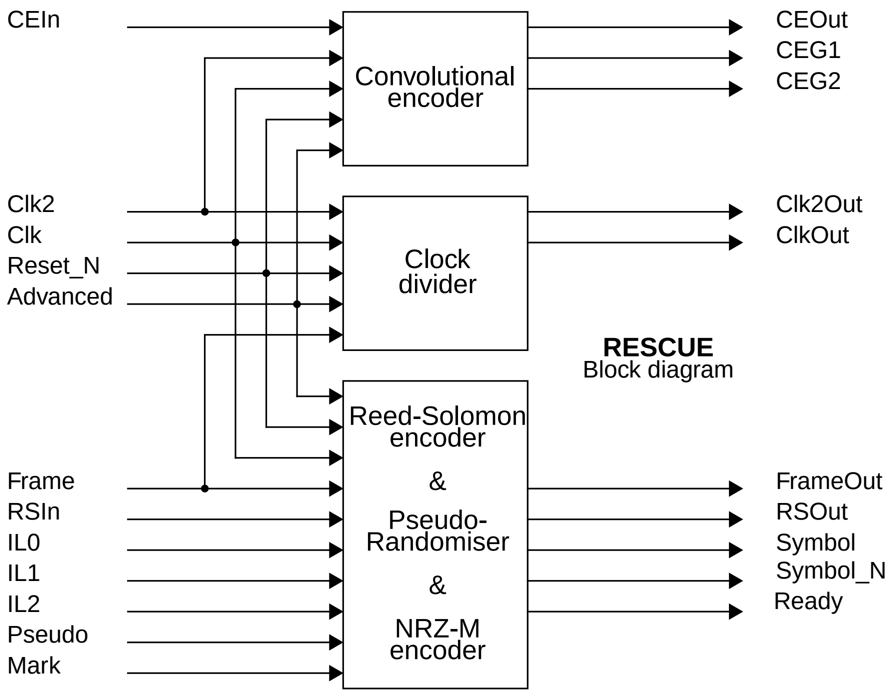


Figure 1: *Block diagram of the RESCUE device*

3 RESCUE DEVICE OUTLINE

The Reed-Solomon and Convolutional Encoder (RESCUE) device encodes telemetry channel data according to the ESA Telemetry Channel Coding standard (Ref. 1). The CCSDS Telemetry Channel Coding standard (Ref. 2) is implicitly supported. The RESCUE device can be viewed as the four separate but interconnected encoders as presented in figure 1. The encoding chain shown in figure 2 can be realised with the RESCUE device.

The Reed-Solomon encoder generates codeblocks by receiving information symbols which are transmitted unmodified while calculating the corresponding check symbols which in turn are transmitted after the information symbols. The check symbol calculation is disengaged during reception and transmission of data not being related to the encoding. The calculation is independent of any previous codeblock and is perform correctly on the reception of the first information symbol after a reset.

The Reed-Solomon encoder is targeted towards systems with low to medium data rates. It is also suitable for deep-space missions since is supports great interleave depths.

The Pseudo-Randomiser generates a bit sequence according to (Ref. 1) which can optionally be XOR-ed with the Reed-Solomon encoder output. This function allows the required bit transition density to be obtained on a channel to enable a receiver on ground to maintain bit synchronisation.

The Reed-Solomon output is nominally Non-Return to Zero Level (NRZ-L) encoded, but can optionally be Non-Return to Zero Mark (NRZ-M) encoded according to (Ref. 6).

The Convolutional encoder encodes data according to (Ref. 1). For each input bit two symbol bits are generated and can be output both serially and in parallel.

The RESCUE device comprises a clock divider that can be used together with the above encoders.

The RESCUE device can operate in a Basic mode where it is largely compatible with the MA1916 device, and in an Advanced mode with enhanced capabilities such as support for additional interleave depths, pseudo-randomising and NRZ-M encoding.

The RESCUE device does not generate the Attached Synchronisation Marker (ASM) as specified in (Ref. 1). Instead, it has a means for bypassing the check symbol calculation while receiving and transmitting unmodified data. This is compatible with the Virtual Channel Multiplexer (VCM) (Ref. 8).

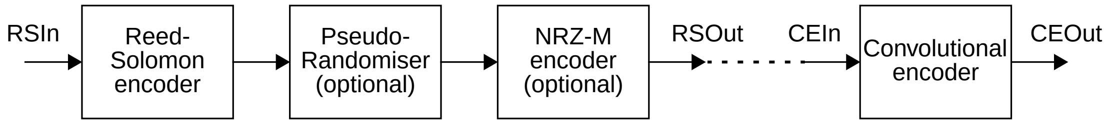


Figure 2: *Encoding chain in the RESCUE device*

4 REED-SOLOMON ENCODER

The Reed-Solomon encoder is implemented using a bit serial architecture to minimise die area consumption, except for the multiplication which is performed in parallel. All operations are made in the Galois field defined by the field polynomial described below. All symbols are transferred serially in a dual basis representation. The multiplier operates directly on the dual basis representation and there is no explicit base conversion before or after the check symbol encoding. The encoder has a serial input and output data interface. Data and control signals of the encoder are directly compatible with the VCM output interface, reducing design effort and required interface circuitry when integrating both in a board design.

The encoder does not require a reset of the check symbol memory before encoding a codeblock, since the internal feedback from the check symbol memory and the product from the multiplier are suppressed when assumed to be zero. Check symbol calculation is therefore independent of any previous codeblocks and correct encoding can begin immediately at the reception of the first information symbol after power-up. The two symbol delimiter outputs are re-synchronised for each codeblock.

To circumvent the effects of heavy particles the control logic is reset and re-synchronised for each codeblock. This, together with the above implementation features, will confine any errors due to Single Event Upsets (SEU) to the affected codeblock. Furthermore, the encoder does not require the input symbol value to be zero while check symbols are transmitted, being the case with some previous encoder designs.

The memory elements of the Pseudo-Randomiser are re-initialised to all-ones before each codeblock. The generated sequence can optionally be XOR-ed bitwise with the Reed-Solomon codeblock. The codeblock comprises the information and check symbols, but not the ASM or any other unmodified data that is transferred while not transmitting information or check symbols. If a gap is inserted between the last bit of the check symbols and the first bit of the ASM, the transferred data stream will consequently not be XOR-ed with the pseudo-random sequence. The XOR-ing of the Pseudo-Randomiser sequence and the Reed-Solomon encoder output is performed before the optional NRZ-M encoding.

The coding algorithm is compliant with (Ref. 1):

- 8 bits per symbol;
- 255 symbols per codeword;
- the encoding is systematic: the first 223 symbols transmitted are information symbols, and the last 32 symbols transmitted are check symbols;
- the code can correct up to 16 symbol errors per codeword;
- the field polynomial is

$$f(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$$

- the code generator polynomial is

$$g(x) = \prod_{i=112}^{143} (x + \alpha^i) = \sum_{j=0}^{32} g_j \cdot x^j$$

  with the highest power of x being transmitted first;
- interleaving is supported for depth I in range 1 to 8, where information symbols are encoded as I codewords with symbol numbers $i + j*I$ belonging to codeword i {where $0 \le i < I$ and $0 \le j < 255$};
- shortened codeword lengths are supported, allowing suppression of a number of information symbols equalling to any multiple of the selected interleave depth (maximum 221 suppressed symbols per interleave depth), where such suppressed symbols are assumed to be in the beginning of the codewords;
- the encoder input and output data are in a representation specified by the following transformation matrix

$$\begin{bmatrix}\iota_0 & \iota_1 & \iota_2 & \iota_3 & \iota_4 & \iota_5 & \iota_6 & \iota_7\end{bmatrix} = \begin{bmatrix}\alpha_7 & \alpha_6 & \alpha_5 & \alpha_4 & \alpha_3 & \alpha_2 & \alpha_1 & \alpha_0\end{bmatrix} \times \begin{bmatrix} 0&1&1&1&1&0&1&1 \\ 0&1&1&1&1&0&0&1 \\ 0&0&1&0&1&0&1&1 \\ 0&0&1&1&1&1&1&1 \\ 0&0&0&0&1&0&0&1 \\ 1&0&0&0&0&1&1&1 \\ 0&1&0&1&1&1&1&1 \\ 0&0&1&1&0&1&1&1 \end{bmatrix}$$

where ι_0 is transferred first, and with the following matrix specifying the reverse transformation

$$\begin{bmatrix}\alpha_7 & \alpha_6 & \alpha_5 & \alpha_4 & \alpha_3 & \alpha_2 & \alpha_1 & \alpha_0\end{bmatrix} = \begin{bmatrix}\iota_0 & \iota_1 & \iota_2 & \iota_3 & \iota_4 & \iota_5 & \iota_6 & \iota_7\end{bmatrix} \times \begin{bmatrix} 1&1&1&0&1&1&0&1 \\ 0&1&0&1&1&1&1&1 \\ 0&0&0&1&0&1&1&1 \\ 0&1&0&1&1&0&1&0 \\ 1&0&0&0&1&0&0&0 \\ 0&1&0&1&0&1&1&0 \\ 0&0&0&0&0&0&1&1 \\ 1&0&0&1&1&0&0&0 \end{bmatrix}$$

- the Pseudo-Randomiser polynomial is

$$h(x) = x^8 + x^7 + x^5 + x^3 + 1$$

5 CONVOLUTIONAL ENCODER

The Convolutional encoder developed encodes data according to (Ref. 1). The code has a constraint length of 7 bits and a code rate of 1/2 bit per symbol. It is generated by the two connection vectors G1=1111001 and G2=1011011, with symbol inversion on the output path of G2 and with G1 associated with the first symbol.

The Convolutional encoder receives data bitwise synchronously with an input clock. For each input bit, two symbols are generated. When output serially, they are multiplexed on one output with a rate twice the input frequency. When output in parallel, they are output with a rate equal to the input frequency.

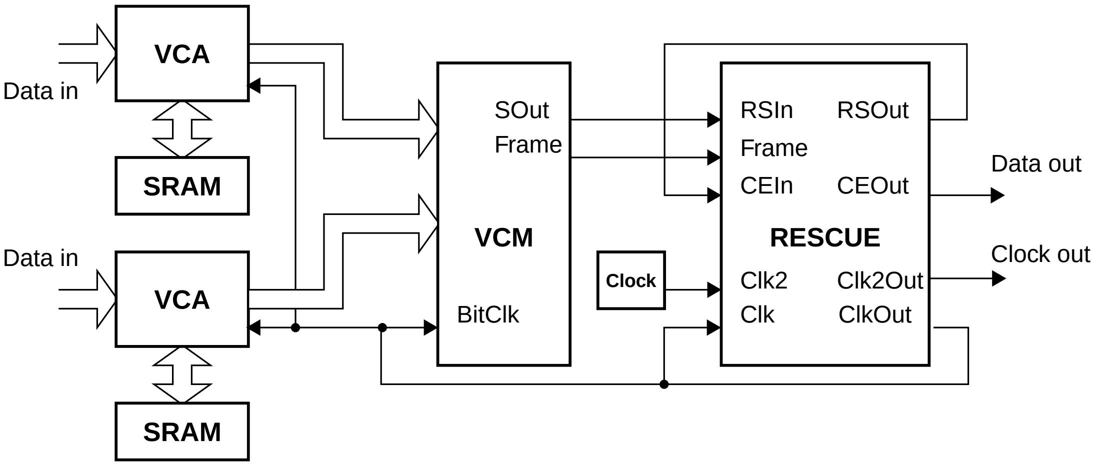


Figure 3: *Telemetry encoder implemented with VCA, VCM and RESCUE devices*

6 DESCRIPTION OF A TYPICAL SYSTEM USING THE RESCUE DEVICE

The RESCUE device is foreseen to be used in telemetry encoders on ESA and commercial spacecraft, replacing the discontinued MA1916 device. The principal functions of an telemetry encoder as per (Ref. 12) is to receive data from the on board data handling system that are formatted and output. The data can be received on virtual channels, where each virtual channel can be seen as an independent communication link. Since there is only one physical output from the telemetry encoder, the data from the virtual channels are time multiplexed on a per frame basis. The telemetry encoder generates the header and the optional trailer of the Transfer Frame. An ASM is inserted in front of the Transfer Frame and the Reed-Solomon check symbols are optionally attached. The data stream can then optionally be fed through the Convolutional encoder.

The Virtual Channel Assembler (VCA) (Ref. 9) and the Virtual Channel Multiplexer (VCM) are radiation-hard devices that together with the RESCUE device can be used for implementing a telemetry encoder. A block diagram of a typical telemetry encoder is shown in figure 3, comprising one VCA and SRAM pair per virtual channel, a VCM and a RESCUE device. In this example the RESCUE device is operating in the Advanced mode and is directly interfaced to the VCM, generating Reed-Solomon symbols (with optional pseudo-randomising and NRZ-M encoding) and convolutionally encoding the data.

7 RESCUE AND MA1916 COMPARISON

As mentioned earlier, there are minor differences between the RESCUE and MA1916 devices as described hereafter.

The following enhancements over the MA1916 device are implemented for both the Basic and Advanced modes:

- symbol clock is re-synchronised for each frame (only done on the first frame after reset for MA1916);
- virtual fill and shortened codeword lengths, supporting 1 to 221 suppressed symbols per interleave depth;
- synchronisation markers, information and check symbol output are deterministic after a reset (MA1916 generates an invalid frame after a reset);
- errors due to SEUs are contained within one codeblock on account of two mechanisms: control logic is re-initialised and re-synchronised for each frame, and the encoding is independent of any previous codewords;
- increased data rates for both encoders.

Clock divider

The CLK input of MA1916 is internally divided in a clock divider which drives the Reed-Solomon encoder with a bit clock half the CLK frequency. The CE_CLKS input is used for delimiting input data to the Convolutional encoder.

The internal clock feed between the clock divider and the Reed-Solomon encoder has been avoided for the RESCUE device since it would had precluded separte use of the encoders in Advanced mode. Instead, the Clk input directly drives the Reed-Solomon encoder. The user has therefore to connect the ClkOut and Clk pins whenever the clock divider is to drive the Reed-Solomon encoder (which is always the case in Basic mode, and therefore being mandatory). In Basic mode, an active clock has to be provided on the Clk2 input, without which none of the encoders can operate.

Convolutional encoder

In Basic mode, the ClkOut output clock will begin toggling when the Frame input is sampled asserted on the rising Clk2 edge after a reset. For sake of simplicity, both the sampling and asserting has been done on the same Clk2 edge for the RESCUE device. In contrast, the MA1916 device samples the SMC input on the falling CLK edge, and asserts the CE_CLKS output on the subsequent rising CLK edge.

The selected clocking scheme in Basic mode is compatible with the MA1916 device assuming that the external clock driving the Frame source is derived and divided from the rising Clk2 edge, and that Frame (and RSIn) is clocked out on the falling edge of this external clock (as is the case when connecting a RESCUE or MA1916 device to a VCM).

First Attached Synchronisation Marker

The first Attached Synchronisation Marker (ASM) is not transmitted by the RESCUE device when in Basic mode, since the clock driving the Reed-Solomon encoder is not operating until the first rising Frame edge is sampled. This is because the clock divider needs to synchronise the phase of the internal symbol clock with that of the external bit clock. Therefore, the first ASM will be ignored since the Frame input will not have yet been assert when the ASM is to be transmitted. Zeros are output instead.

The MA1916 device will transmit the first ASM since it asynchronously transfers input data to the output, and is not dependent on the Reed-Solomon clock at that point in time.

Difference in input/output signal relations

With the MA1916 device, the RSE_OUT, SYZ, SZY and ST1 outputs and the SMC input can all be captured on the rising CLKS output clock edge. This is possible since data on the MSG input are asynchronously transferred to the RSE_OUT output while being received. The SMC input can thus be used as a frame delimiter, sampled on the same clock edge as the outputs listed above.

In Basic mode, data on the RSIn input are sampled on the rising Clk edge and are clocked out on the RSOut output on the subsequent falling Clk edge to emulate the timing of the MA1916 device. All outputs are therefore delayed by half a Clk clock period compared to the input data stream. The FrameOut output has therefore been provided as an output frame delimiter, being a delayed version of the Frame input, suitable for being sampled on the rising ClkOut edge together with the RSOut, Symbol, Symbol_N and Ready outputs. The user will however have to sample the RESCUE outputs one Clk period later than for the MA1916 device.

Pinout

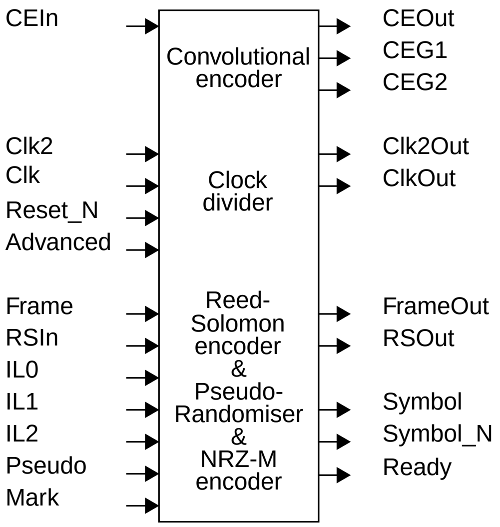
The RESCUE device does not implement the test generator and the production test structures available in the MA1916, but has instead its own production test structures. This affects the following inputs which have to be held statically at logical zero during normal operation in Basic mode; IL2, Pseudo, Mark, Advanced. This also affects the following outputs that have a different behaviour compared to the corresponding MA1916 pins; FrameOut, Unused0, Unused1, CEG1 and CEG2. As explained above, the user has to connect externally the ClkOut output with the Clk input whenever the clock divider is to drive the Reed-Solomon encoder.

Pin 7 and pin 20 are specified as N/C for MA1916, but are used for VDD and VSS, respectively, for RESCUE. A comparison between the RESCUE and the MA1916 pinouts is shown in table 1.

Pin number	I/O type	RESCUE pin name	MA1916 pin name	Basic mode	Advanced mode	MA1916 compatible	Remarks
1	I	Advanced	T2	tie to logical 0	tie to logical 1	partially	mode selection
14	I	Reset_N	n_RST	as n_RST	as for Basic mode	yes	reset
17	I	Clk2	CLK	as CLK	as for Basic mode	no	does not drive Reed-Solomon encoder
27	O	Clk2Out	CLK_OUT	as CLK_OUT	as for Basic mode	yes	Clk2 buffered
22	O	ClkOut	CLKS	as CLKS	always toggling	yes	Clk2 divided by 2
3	I	Clk	CE_CLKS	as CE_CLKS	as for Basic mode	no	drives Reed-Solomon encoder
10	I	IL0	SEL_A	as SEL_A	selects interleave depth	yes	supports 1-8 in advanced mode
12	I	IL1	SEL_B	as SEL_B	selects interleave depth	yes	supports 1-8 in advanced mode
19	I	IL2	T0	tie to logical 0 or 1	selects interleave depth	partially	supports 1-8 in advanced mode
23	I	Pseudo	T3	tie to logical 0	enables pseudo-randomiser	partially	(production test when logical 1)
2	I	Mark	T1	tie to logical 0	enables NRZ-M encoding	partially	(production test when logical 1)
26	I	Frame	SMC	as SMC	as for Basic mode	yes	compatible with VCM
15	O	FrameOut	SMC_OUT	delayed Frame	delayed Frame	no	test generator not supported
18	I	RSIn	MSG	as MSG	as for Basic mode	yes	compatible with VCM
25	O	RSOut	RSE_OUT	as RSE_OUT	as for Basic mode but on rising Clk edge	yes	
28	O	Unused1	MSG_OUT	logical 0	as for Basic mode	no	test generator not supported
16	O	Unused0	READY	logical 0	as for Basic mode	no	test generator not supported
24	O	Symbol	SYZ	as SYZ	as for Basic mode but on rising Clk edge	yes	
13	O	Symbol_N	SZY	as SZY	as for Basic mode but on rising Clk edge	yes	
4	I	CEIn	CE_IN	as CE_IN	as for Basic mode	yes	
5	O	CEOut	CE_OUT	as CE_OUT	as for Basic mode	yes	
8	O	Ready	ST1	as ST1	logical 1	yes	emulates ST1
6	O	CEG1	TEST_POINT	logical 0	G1 output	no	old production test not supported
11	O	CEG2	ST2	logical 0	G2 output (with inverter)	no	old production test not supported
7	P	VDD	N/C			no	old N/C pin
20	P	VSS	N/C			no	old N/C pin
9	P	VSS	VSS			yes	
21	P	VDD	VDD			yes	

Table 1: *Comparison between the pinouts of the RESCUE and MA1916 devices*

8 KEY CHARACTERISTICS OF RESCUE



Compliance:	implementing ESA and CCSDS standards on channel encoding (for packet telemetry and advanced orbiting systems);
Compatibility:	largely compatible with the MA1916, input interface compatible with the VCM;
Performance:	Reed-Solomon encoder and pseudo-random generator with 17 Mbits/s data rate, Convolutional encoder with 17 Mbits/s input data rate and 34 Mbits/s output symbol rate (supporting up to 8 Mbit/s input data rate when interfacing the VCM);
Interleaving:	depth 1 to 8;
Technology:	MITEL 1.25 µm CMOS/SOS5 process (former ABB Hafo);
Radiation:	100 kRad (Si) guaranteed total dose tolerance, low SEU sensitivity and Latch-up immune;
Power:	+5 V supply, low power consumption;
Package:	28 pin flat package;
Schedule:	prototypes in 3Q98, commercial parts in 4Q98, ESA SCC 9000 level B parts in 4Q98;
Process status:	MITEL CMOS/SOS5 ESA Capability Domain Approval expected in 1998;
Support:	supported by foundry as an ASSP with a complete data sheet (Ref. 10);
Part number:	MITEL MS13544

9 POINT OF CONTACT

Direct technical questions regarding the RESCUE device directly to the author. Direct questions regarding RESCUE pricing and availability to:

MITEL Semiconductor AB
Bruttovägen 1
Box 520
SE-17 526 Järfälla Sweden
Tel.: +46 8580 24500

10 REFERENCES

1 Telemetry Channel Coding Standard, ESA PSS-04-103, Issue 1, September 1989, http://www.estec.esa.nl/wsmwww/components/pss_overview.html
2 Telemetry Channel Coding, CCSDS 101.0-B-3, Issue 3, May 1992, ftp://nssdc.gsfc.nasa.gov/pub/ccsds/pdf/CCSDS-101.0-B-3.pdf
3 Packet Telemetry Standard, ESA PSS-04-106, Issue 1, January 1988 (see 1 for web link)
4 Packet Telemetry, CCSDS 102.0-B-4, Issue 4, November 1995, ftp://nssdc.gsfc.nasa.gov/pub/ccsds/pdf/CCSDS-102.0-B-4.pdf
5 Advanced Orbiting Systems, CCSDS 701.0-B-2, Issue 2, November 1992, ftp://nssdc.gsfc.nasa.gov/pub/ccsds/pdf/CCSDS-701.0-B-2.pdf
6 Radio Frequency and Modulation Standard, ESA PSS-04-105, Issue 1, December 1989 (see 1 for web link)
7 GPS SOS Radiation Hard Handbook, July 1995, http://www.gpsemi.com
8 Virtual Channel Multiplexer (VCM), Preliminary Data Sheet, MITEL Semiconductors, HAF_12396, January 1994, ftp://ftp.estec.esa.nl/pub/vhdl/doc/HAF12396.ps
9 Virtual Channel Assembler (VCA), Preliminary Data Sheet, MITEL Semiconductors, HAF_12399, January 1994, ftp://ftp.estec.esa.nl/pub/vhdl/doc/HAF12399.ps
10 Reed-Solomon and Convolutional Encoder (RESCUE), Preliminary Data Sheet, SMARTECH, February 1998, http://www.smartech.fi/MS13544.pdf
11 *A CCSDS Reed-Solomon Encoder Core and Radiation Hard Device*, S. Habinc, ESA Workshop on Microelectronics for Space Applications, Noordwijk, December 1996, ftp://ftp.estec.esa.nl/pub/vhdl/doc/Rescue.pdf
12 *Components for Building Telecommand and Telemetry On-board Systems*, P. Sinander, On-board Data Management Symposium, Rome, January 1995, ftp://ftp.estec.esa.nl/pub/vhdl/doc/TmTc.ps

11 ACRONYMS AND ABBREVIATIONS

ASIC	Application Specific Integrated Circuit
ASM	Attached Synchronisation Marker
ASSP	Application Specific Standard Product
CCSDS	Consultative Committee for Space Data Systems
CMOS	Complementary Metal-Oxide Semiconductor
ESA	European Space Agency
N/C	Not Connected
NRZ-L	Non-Return to Zero - Level
NRZ-M	Non-Return to Zero - Mark
RESCUE	Reed-Solomon and Convolutional Encoder
RS	Reed-Solomon
SEU	Single Event Upset
SOS	Silicon On Sapphire
SRAM	Static Random Access Memory
VCA	Virtual Channel Assembler
VCM	Virtual Channel Multiplexer
VHDL	VHSIC Hardware Description Language
VHSIC	Very High Speed Integrated Circuits
XOR	Exclusive Or