MemecCore™ Product Line
3721 Valley Centre Drive
San Diego, CA 92130 USA
Americas: +1 800-752-3040
Europe: +41 (0) 32 374 32 00
Asia: +(852) 2410 2720
E-mail: actel.info@memecdesign.com
URL: www.memecdesign.com/actel

Product Summary

Intended Use

- Data communication channels
- DTV/HDTV broadcast
- Data storage systems
- Satellite communications

Key Features

- Core design is customized using the following specs
  - Primitive polynomial
  - Generator polynomial
  - Number of parity symbols
  - Symbol size
  - Symbol clock rate
  - System clock rates
- Message Block length configured by end user
- Continuous or burst mode operation
- Supports high speed applications (>900Mbps)
- Simple core interface for ease of integration
- Encoder available with or without Variable Parity Check Bytes

Targeted Devices

- Axcelerator Family
- ProASIC^{PLUS} Family

General Description

Reed-Solomon coding is a method of forward error correction in the form of block coding. Block coding consists of calculating a number of parity symbols over a number of message symbols. The parity symbols are appended to the end of the message symbols forming a codeword. Reed-Solomon coding is described in the form RS(n,k), where k is the number of message symbols in each block and n is the total number of symbols in the codeword. The value t defines the number of symbols that can be corrected by the Reed-Solomon code, where t=(n-k)/2 and the number of parity symbols is equal to 2t.

Reed-Solomon codes are calculated in a finite field of elements, or Galois fields GF(2^m). The Galois field is defined by a primitive polynomial P(x). The degree of the primitive polynomial m defines the number of bits per data symbol (m = bits per symbol) and the maximum length of the Codeword (2^m-1=maximum Codeword length n). The generator polynomial G(x) further defines the field of Codewords that the parity symbols are derived from.

MC-ACT-RSENC is a "core" logic module specifically designed for Actel FPGAs. The logic symbol of this core is shown in Figure 1.

Core Deliverables

- Netlist Version
  - Compiled RTL simulation model, compliant with the Actel Libero environment
  - Netlist compatible with the Actel Designer place and route tool
- RTL Version
  - Verilog Source Code
  - VHDL Source Code
- All
  - User Guide
  - Test Bench

Synthesis and Simulation Support

- Synthesis: Synplicity
- Simulation: ModelSim
- Other tools supported upon request

Verification

- Test Bench
- Test Vectors

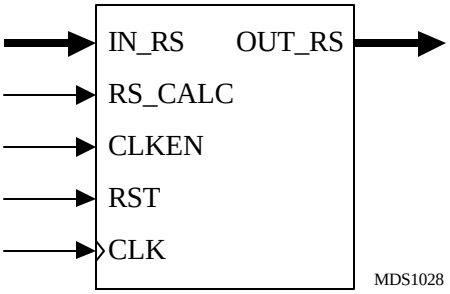

Figure 1: Logic Symbol

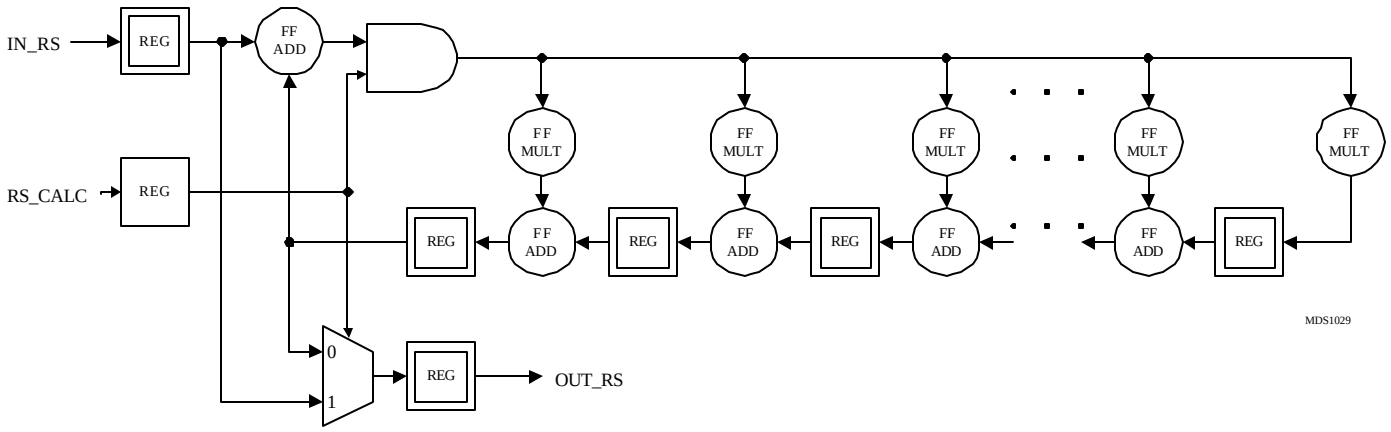

Figure 2: Block Diagram

Functional Description

The Reed-Solomon encoder core is partitioned into modules as shown in Figure 2, and described below.

All registers are driven by common clock (CLK), clock enable (CLKEN), and asynchronous reset (RST) signals. This provides ultimate flexibility in integration of the core into larger systems. The registers are clocked on the rising edge of CLK, enabled by a high on CLKEN, and asynchronously reset by a high on RST.

The RS_CALC signal controls whether the core is calculating the parity (high) or shifting out the calculated parity (low). Logic external to the core is required to generate this control signal. This allows the user to vary the message block length as desired for shortened codes. The RS_CALC pin must be held at a low for 2t enabled clock cycles to ensure all parity register values are shifted out and the registers are cleared.

FFADD Blocks perform modulo 2 addition of the input symbols.

FFMULT Blocks perform a finite field multiplication, over the Galois field, of a constant and the input symbol.

Timing

Figure 3 shows a simple timing diagram of the RSENC interfaces for a Reed-Solomon block code of RS(204,188).

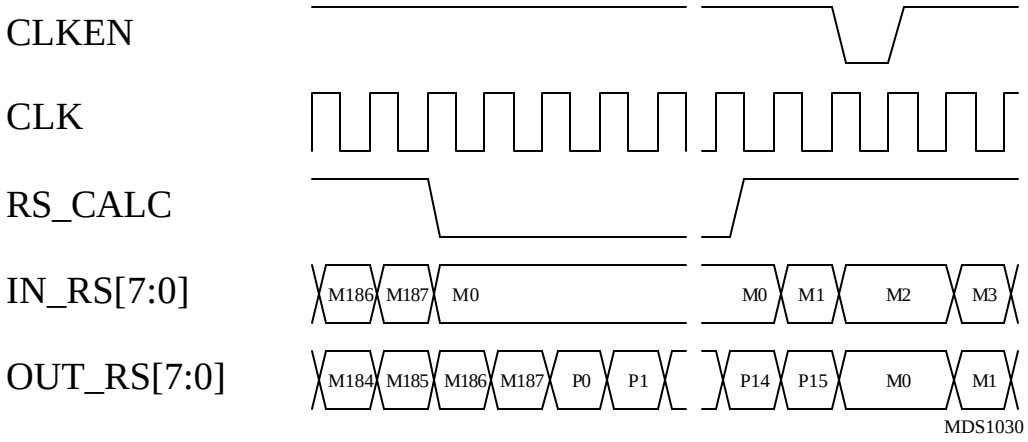

Figure 3: RSENC Timing

Reed-Solomon Encoder Variations

Various forms of the core have been developed. One supports high data rates where area is a secondary concern. Another supports low data rates where a high-speed system clock is available and area is a primary concern. The cores work for any valid codeword length and can be customized to support either fixed or variable values of parity symbols. The core has been developed in two different approaches. One approach is optimized for encoders with a single value of t. The other approach is optimized for encoders with selectable values of t.

The RSENC-INTELSAT is a precustomized core that calculates a selectable 14, 16, 18, or 20 parity symbols (t = 7, 8, 9, or 10) and is compliant with the IESS-308 Intelsat standard. The primitive polynomial and generator polynomial implemented is $P(x)= x^8+x^7+x^2+x+1$ and $G(x)= (x-\alpha^{120})(x-\alpha^{121})\ldots(x-\alpha^{119+2t})$ over a Galois field of GF(256). A 2-bit input control signal selects between four different codeword/message lengths: RS(126,112), RS(194,178), RS(219,201), and RS(225,205). The RSENC-INTELSAT core is delivered as Verilog RTL source code. Detailed timing and pin descriptions of the RSENC-INTELSAT can be found in the User's Guide available from MemecCore.

The RSENC-DVB is a precustomized core that calculates 16 parity symbols and is compliant with the DVB standard. The primitive polynomial and generator polynomial implemented is $P(x)=x^8+x^4+x^3+x^2+1$ and $G(x)=(x-\alpha^0)(x-\alpha^1)\ldots(x-\alpha^{15})$ over a Galois field of GF(256). The core is adaptable to any message/codeword length RS(n,k) where n-k=16. Thus, the largest code supported by this core is R(255,239). The DVB standards specify a code of RS(204,188).

Parameter	Symbol	DVB	Intelsat
Primitive Polynomial	P(x)	$x^8 + x^4 + x^3 + x^2 + 1$	$x^8 + x^7 + x^2 + 1$
Generator Polynomial	G(x)	$(x-\alpha^0)(x-\alpha^1)(x-\alpha^2)(x-\alpha^3)\ldots(x-\alpha^{15})$	$(x-\alpha^{120})(x-\alpha^{121})\ldots(x-\alpha^{119+2t})$
Bits per symbol	m	8	8
Codeword Length	n	204 (Note 1)	126, 194, 219, 225
Message Word Length	k	188	112, 178, 201, 205 (Note 2)
Parity Symbols (n-k)	2t	16	14, 16, 18, 20

Table 1: Reed-Solomon Encoder Parameters

Notes:

1. Core is adaptable to any RS(n,k) where n-k=16 and largest code being RS(255,239).
2. Selected by 2-bit input control signal for RS(126,112), RS(194,178), RS(219,201), and RS(225,205).

Device Requirements

Family	Device	Utilization			Performance
		COMB	SEQ	Total	
Axcelerator	AX500-3	6%	10%	7%	119 MHz
ProASIC^{PLUS}	APA075	n/a	n/a	34%	51 MHz

Table 2: Device Utilization and Performance*

*Note: The data provided is for an encoder with RS(255,225), G(x)=120, P(x)=125, m=8.

Verification and Compliance

Functional and timing simulation has been performed on the RSENC using Verilog Test Benches. Simulation vectors used for verification are provided with the core. This core has also been used successfully in customer designs.

Signal Descriptions

The following signal descriptions define the IO signals.

Signal	Direction	Description
IN_RS[m-1:0]	Input	Message block input.
RS_CALC	Input	Parity Calculate
CLKEN	Input	Clock Enable that enables all registers (active high).
CLK	Input	System Clock that drives all registers.
RST	Input	Asynchronous reset (active high).
OUT_RS[m-1:0]	Output	Coded block output.

Table 3: RSENC I/O Signals

Recommended Design Experience

For the source version, users should be familiar with HDL entry and Actel design flows. Users should be familiar with Actel Libero v2.2 Integrated Design Environment (IDE) and preferably with Synplify and ModelSim.

Ordering Information

Part Number	Description
MC-ACT-RSENC-NET	RS Encoder Netlist
MC-ACT-RSENC-VLOG	RS Encoder Verilog
MC-ACT-RSENC-VHDL	RS Encoder VHDL
MC-ACT-RSENCU-NET	RS Encoder Netlist with Variable Parity Check Bytes
MC-ACT-RSENCU-VLOG	RS Encoder Verilog with Variable Parity Check Bytes
MC-ACT-RSENCU-VHDL	RS Encoder VHDL with Variable Parity Check Bytes

Table 4: Core Part Numbers

The CORE is provided under license from Memec Design for use in Actel programmable logic devices. Please contact Memec Design for pricing and more information.

Information furnished by Memec Design is believed to be accurate and reliable. Memec Design reserves the right to change specifications detailed in this data sheet at any time without notice, in order to improve reliability, function or design, and assumes no responsibility for any errors within this document. Memec Design does not make any commitment to update this information.

Memec Design assumes no obligation to correct any errors contained herein or to advise any user of this text of any correction, if such be made, nor does the Company assume responsibility for the functioning of undescribed features or parameters. Memec Design will not assume any liability for the accuracy or correctness of any support or assistance provided to a user.

Memec Design does not represent that products described herein are free from patent infringement or from any other third-party right. No license is granted by implication or otherwise under any patent or patent rights of Memec Design.

MemecCore products are not intended for use in life support appliances, devices, or systems. Use of a MemecCore product in such application without the written consent of the appropriate Memec Design officer is prohibited.

Datasheet Revision History

Version	Date	Description
Datasheet 1.0	January 10, 2003	First Release
Datasheet 1.1	April 24, 2003	Updated corporate address, added CompanionCore logo