



### Features

- CCSDS 131.0-B-1 compatible
- Integrated channel coding solution
- Viterbi Decoder
  - Constraint length 7
  - Nominal code rate 1/2
  - Punctured rates 2/3, 3/4, 5/6, 7/8
- Attached Sync Marker removal
- Data de-randomizer
- Reed-Solomon decoder
  - $n = 255, k = 239, t = 8$
  - 8-bit symbols
  - Interleaving
  - Field generator polynomial:  
 $F(x) = x^8 + x^7 + x^2 + x + 1$
  - Generator polynomial:  
 $g(x) = \prod (x - \alpha^{1j})$
- Low latency
- Simple handshake protocol for reliable interfacing
- Fully synchronous design
- High speed operation > 100 Mbps
- Comprehensive verification plan provided

### General Description

The SAL40400D consists of verilog IP for implementing the CCSDS channel decoding.

The device consists of a Viterbi decoder, Attached Sync Marker (ASM) deletion, data randomizer, and an 8 -error - correcting Reed Solomon decoder.

The device consists of multiple modules, each capable of being bypassed. Modules that are bypassed are disabled to save power.

Incoming data are first (optionally) decoded by the Viterbi decoder with optional de-puncturing, based on the specified convolutional code rate. An Attached Sync Marker is then (optionally) detected and removed from the data, to allow data frame synchronization. The data are then (optionally) passed to the data de-randomizer module, which unscrambles the data. Finally, the data are (optionally) passed to the Reed-Solomon decoder module, which further decodes the data stream.

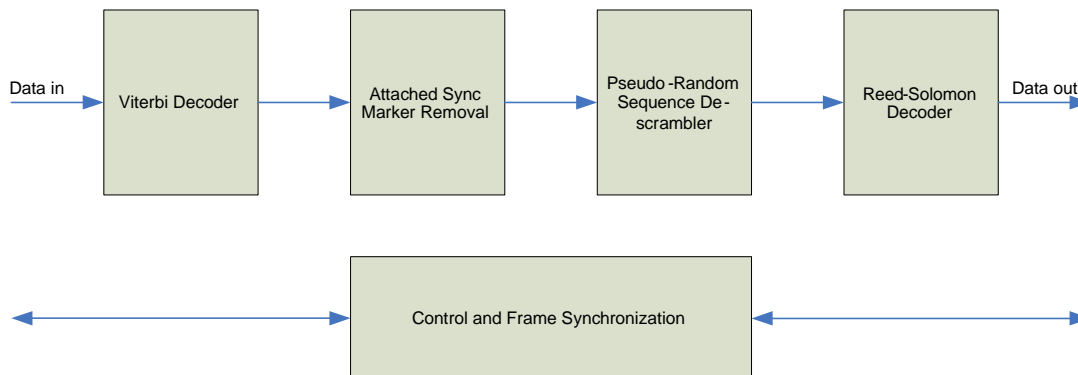


Figure 1. Block diagram

## Theory of Operation

### Viterbi Decoder

The input data are first presented to a CCSDS-compatible (7,1/2) Viterbi decoder. The device accepts 4-bit soft input data samples formatted as shown in table 1. The data are optionally depunctured depending on the code rate going into the device, inserting zero samples where necessary to re-make the nominal rate  $\frac{1}{2}$  code, as shown in table 2 below.

The incoming data are assigned a set of branch metrics based on different assumptions of the current state of the code trellis. These branch metrics are then passed to the path metric unit and are added to the accumulated path metric and the most likely path metric is selected and stored for subsequent comparison.

After a traceback depth's worth of data are captured, the traceback unit works backward through the trellis, pulling the most likely bits from the survivor memory as it goes. The data are stored in a Last-In-First-Out (LIFO) buffer and made available to the next signal processing step at the end of the traceback sequence.

in_mode[1:0]	Input format
00	2's compliment
01	sign-magnitude
10	binary
11	hard input
Table 1. input mode	

Puncture pattern 1 = accept bit 0 = insert '0'	Code Rate	input Sequence
C1: 1 0 C2: 1 1	2/3	c1(1), c2(1), c2(2), ...
C1: 1 0 1 C2: 1 1 0	3/4	c1(1), c2(1), c2(2), c1(3), ...
C1: 1 0 1 0 1 C2: 1 1 0 1 0	5/6	c1(1), c2(1), c2(2), c1(3), c2(4), c1(5), ...
C1: 1 0 0 1 0 1 C2: 1 1 1 1 0 1 0	7/8	c1(1), c2(1), c2(2), c2(3), c2(4), c1(5), c2(6), c1(7), ...
Table 2. Puncture patterns		

## Attached Sync Marker

Synchronization of the data transferred across a CCSDS compliant communication system is achieved by using a stream of fixed length data blocks separated by a specific known bit pattern between them. This bit pattern is known as an Attached Sync Marker, or ASM. The data blocks are known as Codeblocks if Reed-Solomon (or turbo) coding is used on the data prior to randomization. Otherwise, the data block is known as a Transfer Frame. After the attachment of ASM, the data block is known as a Channel Access Data Unit.

Although not optional according to the CCSDS recommendations, the ASM is optional in this device to provide maximum flexibility for the user. The default behavior is for the ASM to be prepended to the RS-coded and randomized Codeblock or the Transfer Frame, if Reed-Solomon coding is bypassed.

This module finds and removes the ASM from the data stream, working in conjunction with the Viterbi unit to synchronize the incoming data.

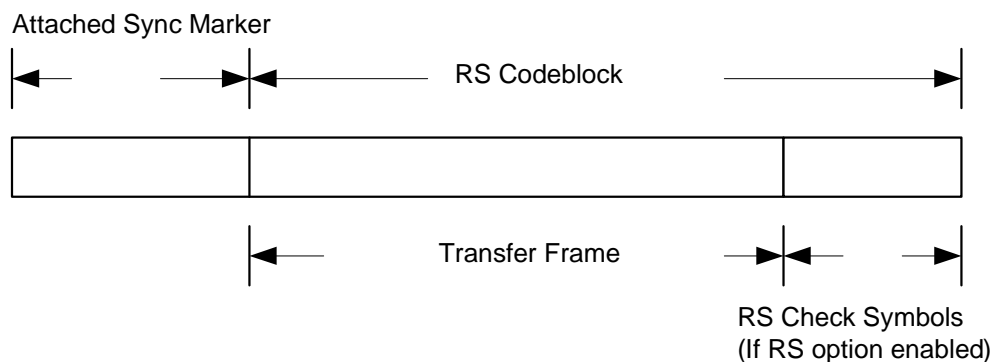


Figure 2. Sync Marker format

## Pseudo-Randomizer

In order to ensure proper receiver operation, the data stream must be sufficiently random. The pseudo-randomizer in the SAL40400 products is the preferred method to ensure sufficient randomness for all combinations of CCSDS-recommended modulation and coding schemes. The presence or absence of pseudo-randomization is fixed for a physical channel, and its existence is understood *a priori* by the receiver.

Randomization is achieved by an exclusive OR of each bit of the Codeblock or Transfer Frame with a standard pseudo-random sequence. On the receiving end (i.e., this device) it is again applied to de-randomize the data after convolutional coding (if used) and Codeblock synchronization.

The pseudo-random sequence is generated using the circuit shown below, which implements the following polynomial:

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

This sequence begins at the first bit of the Codeblock or Transfer Frame and repeats after 255 bits, continuing repeatedly until the end of the Codeblock or Transfer Frame.

The sequence generator is initialized to all-ones state at the start of each Codeblock or Transfer Frame.

The first 40 bits of the pseudo-random sequence from the generator are:

1111 1111 0100 1000 0000 1110 1100 0000  
1001 1010 ....

The leftmost bit is the first bit of the sequence to be XOR'd with the first bit of the Codeblock or Transfer Frame, etc.

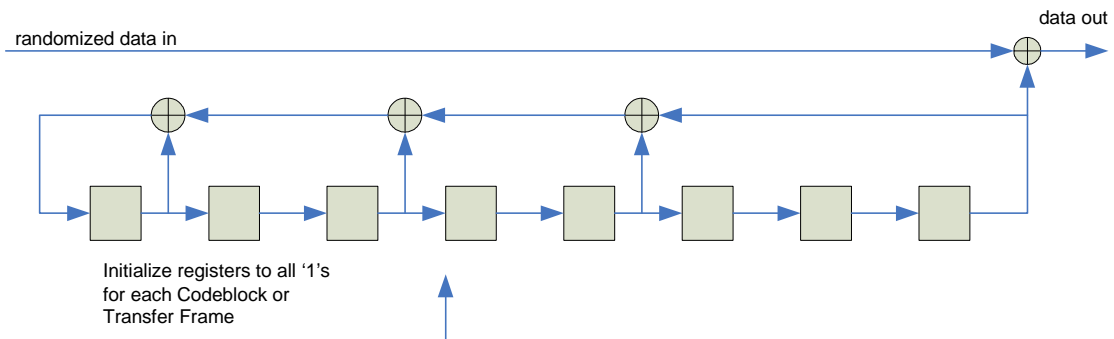


Figure 3. Randomizer block

## Reed-Solomon Decoder

The CCSDS recommendations specify two Reed-Solomon constructions, differing in their error correction ability. As one might expect, the greater the error correcting ability, the greater the overhead in terms of redundant symbols. The specification additionally outlines an interleaving scheme, which allows a larger data block to be defined. The SAL40400D device supports all the options defined in the CCSDS spec:

- $n = 255, k = 239, t = 8$
- 8-bit symbols
- Interleaving
- Field generator polynomial:  
 $F(x) = x^8 + x^7 + x^2 + x + 1$
- Generator polynomial:  
 $g(x) = \prod (x - \alpha^{1j})$

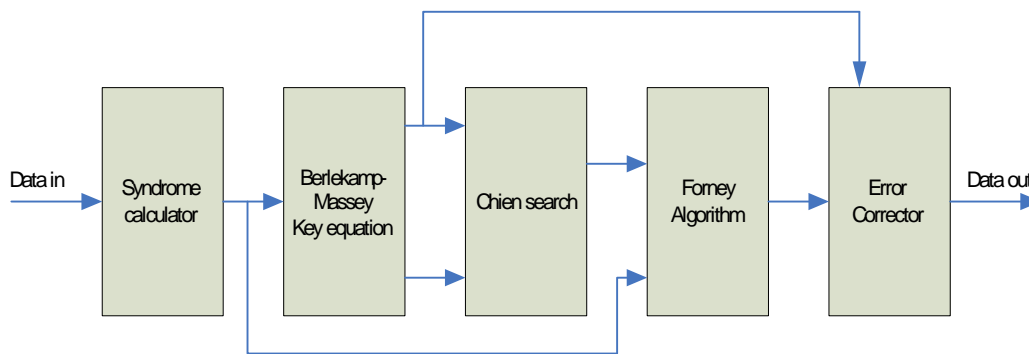


Figure 4. Reed-Solomon block

## Signal Descriptions

The module pinout is shown in the figure below, and in table 1. The signals are conveniently organized into functional groups as follows:

### Clock and Reset

The design is fully synchronous with a single clock signal. The reset signal is synchronous and needs to be asserted for at least one full clock cycle to reset internal logic.

### Flow Control signals

Three signals control flow of data into the device, *din\_rdyin\_n* and *din\_rdyout\_n*. The *din\_rdyin\_n* signal indicates that data into the device is valid. The *din\_rdyout\_n* signal indicates that the device is ready to receive data. Valid data is flowing into the device if *din\_rdyin\_n* and *din\_rdyout\_n* are both asserted (low).

Two signals control flow of data out of the device, *dout\_rdyin\_n* and *dout\_rdyout\_n*. The *din\_rdyin\_n* signal is used to indicate to the device that the external interface is ready to receive data. The *dout\_rdyout\_n* signal indicates that data out of the device is valid. Valid data is flowing out of the device if *dout\_rdyin\_n* and *dout\_rdyout\_n* are both asserted (low).

### Data signals

The data are clocked in on *din* and clocked out on *dout*.

### Configuration signals

In addition to the above defined signals, there are a number of mode bits which are defined in table 3 below.

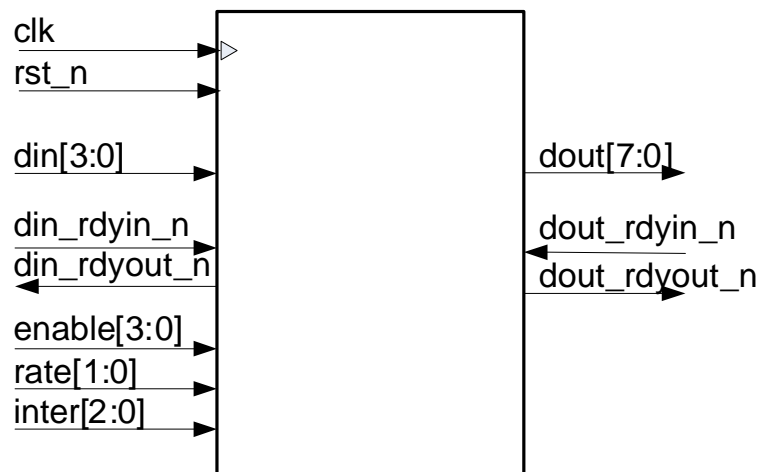


Figure 5: Component pinout

Pin	Sense	Width	Description
clk	in	1	Clock
rst_n	in	1	Synchronous reset
din	in	4	Serial data (message) in
din_rdyin_n	in	1	Indicates serial data in is valid
din_rdyout_n	out	1	Indicates device is ready to receive data in
dout	out	1	Data out
dout_rdyin_n	in	1	Indicates that it's OK to shift data out
dout_rdyout_n	out	1	Indicates that data is available to be shifted out
enable	in	4	Individual internal module enables: enable[3] = enable Reed-Solomon enable[2] = enable Pseudo-Randomizer enable[1] = enable Attached Sync Marker enable[0] = enable convolutional coding
rate	in	2	Convolutional code rate: 00 = 2/3, 01 = 3/4, 10 = 5/6, 11 = 7/8
inter	in	3	Reed-Solomon Interleave: 001/110/111 = 1, 010 = 2, 011 = 3, 100 = 4, 101 = 5, 000 = 8
Table 3. Component pinout			

## Waveforms

### Input

The input functional timing is shown below. *Din\_rdyin\_n* is used as an input data enable, *din\_rdyout\_n* is used to indicate when the device is ready to receive data.

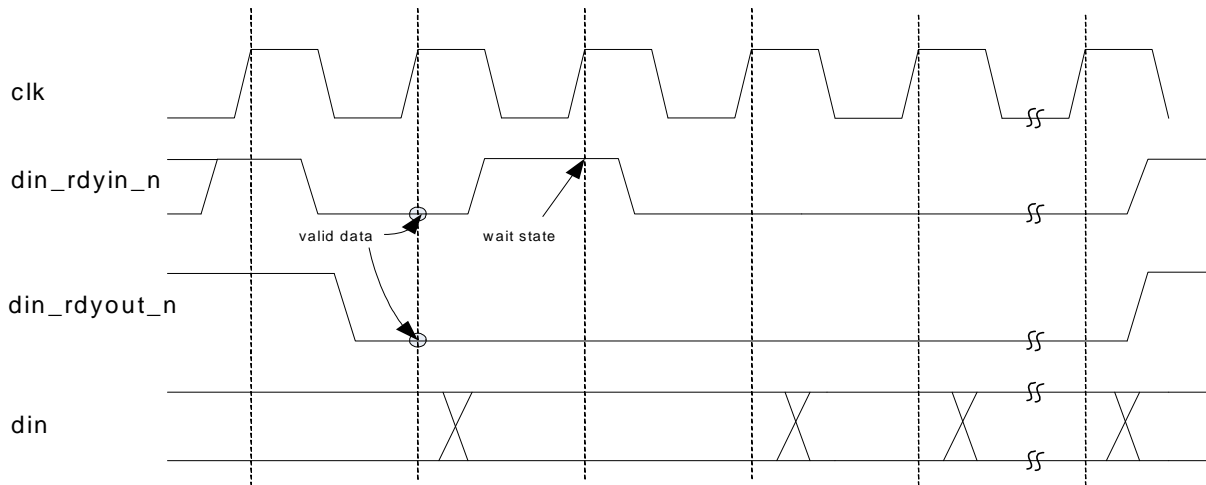


Figure 6. Input timing

### Output

The output functional timing is shown below. *Dout\_rdyout\_n* is used as an output data ready indication, *dout\_rdyin\_n* is used to indicate to the device that it's OK to shift data out.

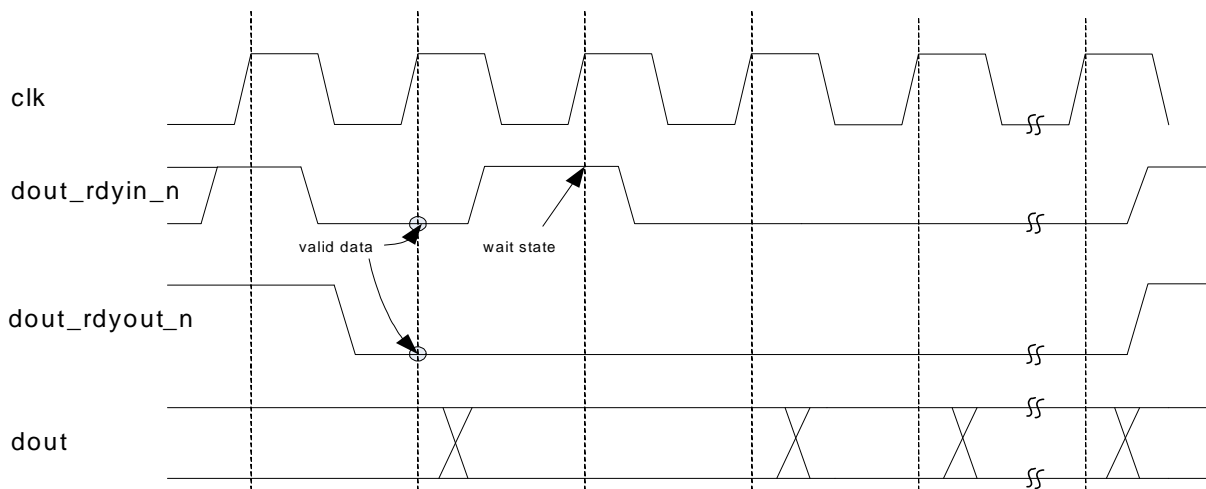


Figure 7. Output timing

## Module Verification

The SAL40400D has been subjected to extensive verification to ensure the highest quality product possible. A comprehensive test plan was implemented which included the following:

- High-quality random data source
- High-quality random noise source
- Extensive flow-control simulations
- Verification of operation against known data sequences

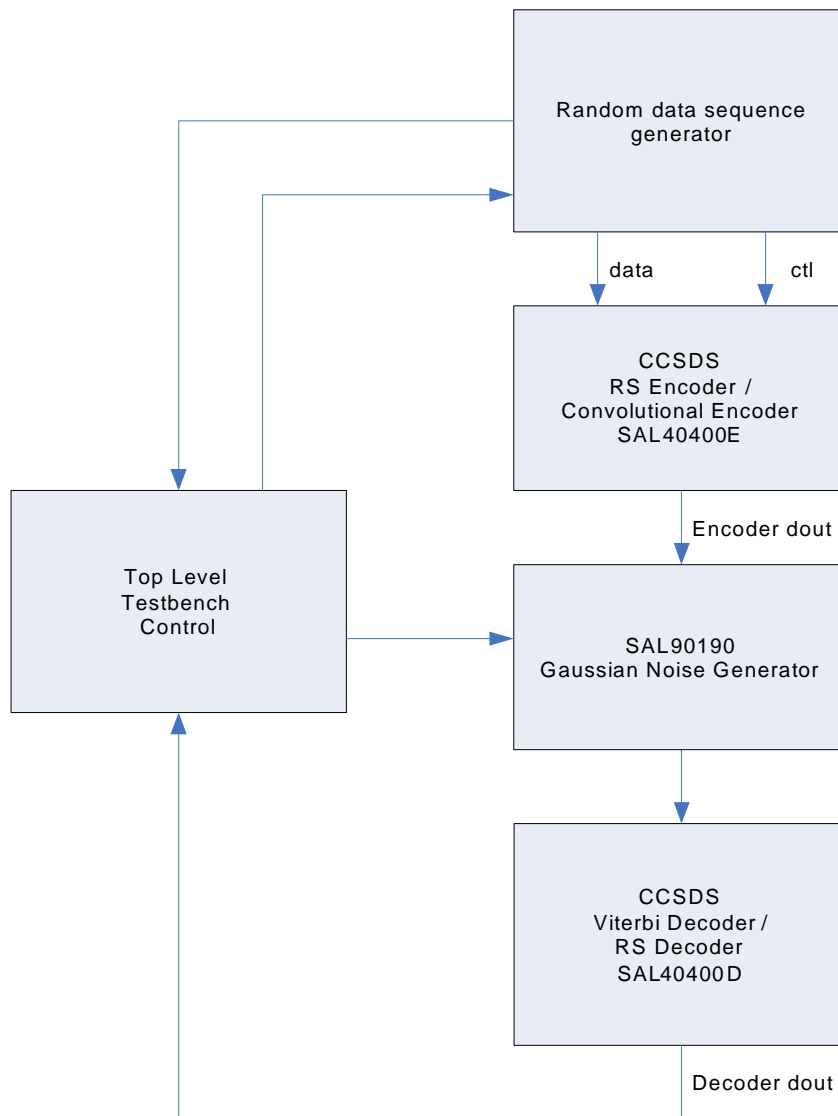
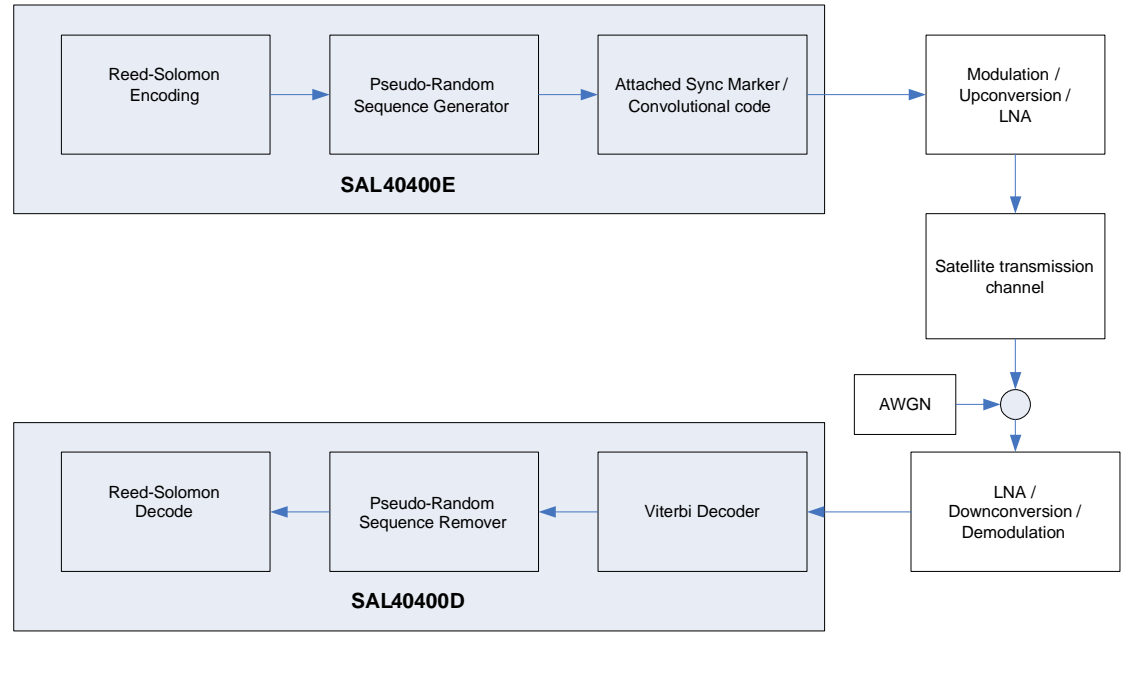


Figure 8: Testbench Block Diagram

## Application: Satellite System

The CCSDS channel coding forms an integral part of many satellite telemetry systems.



## Ordering Information

Salamander Error Correction currently has 2 CCSDS-compatible Viterbi / Reed-Solomon decoder IP modules available:

**SAL40400D** CCSDS Viterbi / RS decoder ( $t=8$ )

**SAL40401D** CCSDS Viterbi / RS decoder ( $t=16$ )

About Salamander:

**Salamander Error Correction** develops and sells error correction modules of the highest quality worldwide.

Salamander Error Correction is a division of Komodo Industries.

Salamander Error Correction:  
3364 Via Alicante  
La Jolla, CA 92037

[sales@salamander-ecc.com](mailto:sales@salamander-ecc.com)