



Features

- For use in CCSDS systems. Fully compliant with the CCSDS standard 131.0-B1
- Double Buffered input enables high speed operation
- Generic memory interface for easy ASIC integration
- Simple handshake protocol for reliable interfacing
- Fully synchronous design
- Full CCSDS compliant hardware interleaver. No external calculation required.
- Interleaver frame size selectable on a frame-by-frame basis
- Comprehensive verification plan provided
- All frame sizes supported
- All rates (1/2, 1/3, 1/4, 1/6) supported

General Description

The SAL40100ES consists of verilog IP for implementing the parallel concatenated convolutional (turbo) encoder as defined by the CCSDS standard. Refer to Fig. 1 below.

As shown in the figure, the encoder consists of two identical constituent Recursive Systematic Convolutional (RSC) coders operating in parallel on the input data frame. One RSC coder operates on the data in natural order and the other operates on a permuted version of the input data. The permutation is accomplished as described in detail in the CCSDS specification using the interleave address generator.

The encoder outputs a number of bits for every input bit, depending on the code rate: an unencoded copy of the data, parity bits from the first convolutional coder, and parity bits from the second convolutional coder.

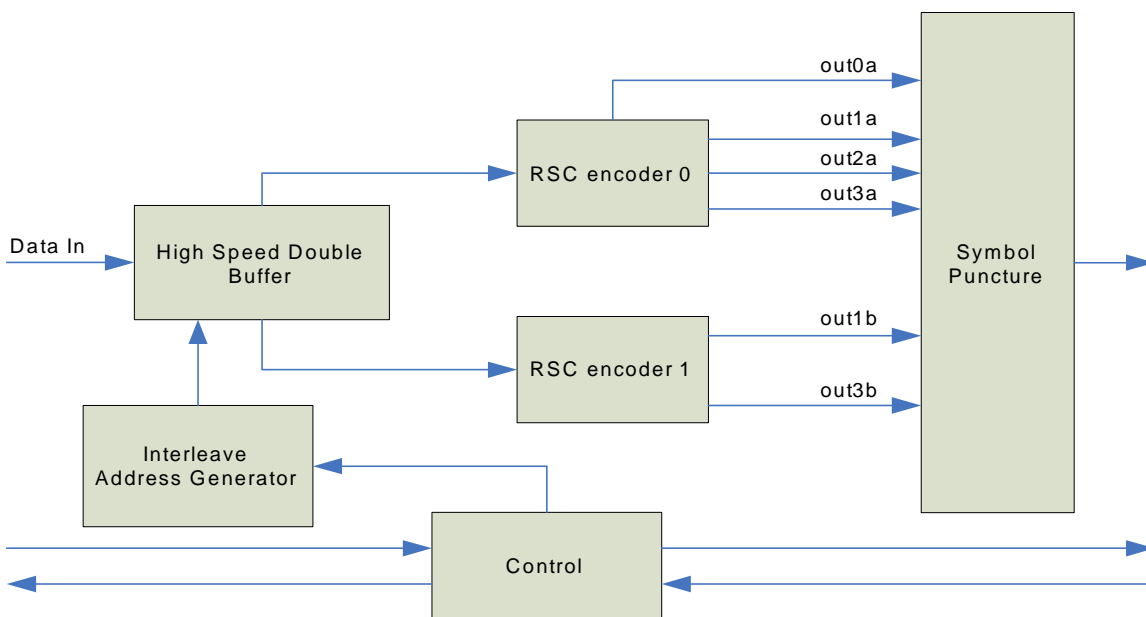


Figure 1: Encoder Block Diagram

Theory of Operation

The encoder is shown in more detail in figure 2 below. It consists of two RSC encoders, one of which receives the data in normal sequential order (at in_a), and one which receives the data in interleaved order (at in_b).

Connections

The feedback connection vector for both constituent codes is $G_0 = 23_8$ (in octal notation). The forward connection vector for rate 1/2 and rate 1/3 is $G_1 = 33_8$. Puncture (deletion) of every other parity symbol is necessary for the rate 1/2 code. No other code rates are punctured.

Forward connection vectors for the rate 1/4 code are $G_2 = 25_8$, $G_3 = 37_8$, (1st component code) and $G_1 = 33_8$ (2nd component code).

Forward connection vectors for the rate 1/6 code are $G_1 = 33_8$, $G_2 = 25_8$, $G_3 = 37_8$, (1st component code) and $G_1 = 33_8$ and $G_3 = 37_8$ (2nd component code).

The encoded symbols are multiplexed from top to bottom along the output line for the selected code rate to form the Turbo Codeblock (as defined in the spec). The output sequence is repeated $k + 4$ bit times, where k is the number of bits in the information sequence to be encoded.

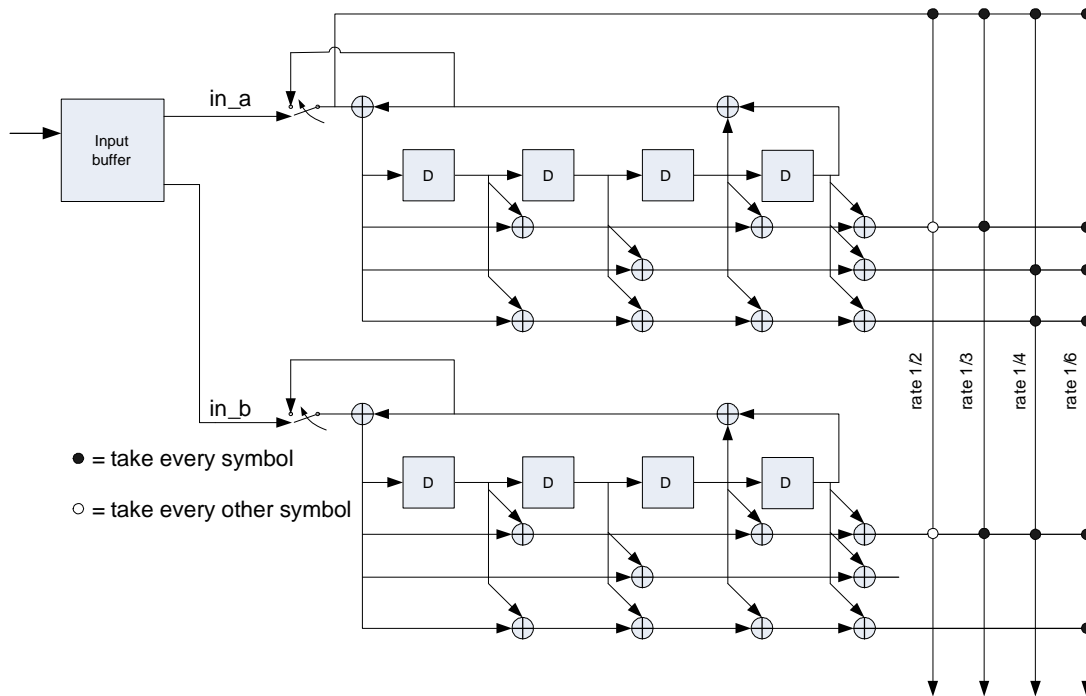


Figure 2 Encoder details

Specifying the code rate

The code rate is specified using the rate signal according to table 1 below.

rate signal	code rate
00	1/2
01	1/3
10	1/4
11	1/6

Table 1. Code rate

block length (bits)	notes
1784	223 x 1 octet
3568	223 x 2 octets
7136	223 x 4 octets
8920	223 x 5 octets
16384	2048 octets

Table 2. Block lengths

Trellis termination

Both constituent encoders in figure 2 are initialized with '0's in all registers. To assist in the decoding, the encoders should be returned to the all zeros condition. This is accomplished by moving the input switches to the upper position to receive feedback from the shift registers. This causes all four registers to become filled with zeros. The code rate is thus reduced slightly from the nominal rate due to these "tail" bits added to the information stream.

Allowable information block lengths

There are only 5 information block lengths, as shown in the table below. They were chosen for compatibility with corresponding Reed-Solomon interleaving depths.

Signal Descriptions

The module pin out 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.

Control signals

Init, *rate*, and *start_encode* are control signals whose operation is described in table 3 below.

Data signals

The serial data are shifted in on *din*. The encoded data are shifted out on *out_0a* – *out_3b*, as shown in figure 2, and as shown in the waveforms below.

Input Flow Control

Two signals control the flow of data into the device. *Din_rdyin_n* is an active low signal indicating to the device that valid data is ready on *din*. *Din_rdyout_n* is an active low signal supplied by the device that indicates that it's ready to accept data in on *din*. Valid data are flowing into the device whenever both *din_rdyin_n* and *din_rdyout_n* are low.

Output Flow Control

Two signals control the flow of data out of the device. *Dout_rdyin_n* is an active low signal indicating to the device that the interface can receive data from the outputs. *Dout_rdyout_n* is an active low signal supplied by the device that indicates that it has valid data to shift out. Valid data are flowing out of the device whenever both *dout_rdyin_n* and *dout_rdyout_n* are low.

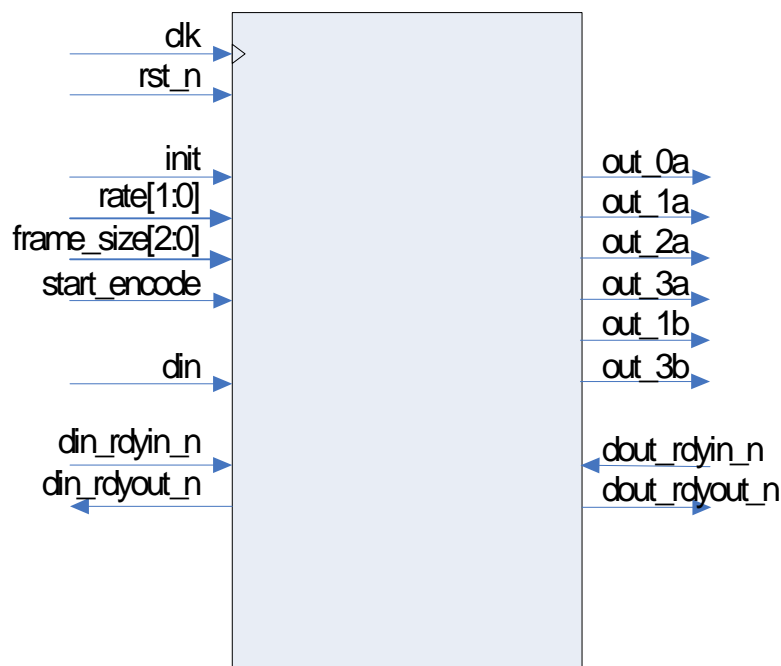


Figure 3 Component Pinout

Pin	Sense	Width	Description
clk	in	1	Clock
rst_n	in	1	Synchronous reset
din	in	1	Serial data (message) in
din_rdyin_n	in	1	Indicates serial data in is valid
din_rdyout_n	out	1	Indicates module ready to accept data (not all input buffers full)
out_0a	out	1	Systematic data out
out_1a	out	1	Data parity out from 1 st constituent coder
out_2a	out	1	Data parity out from 1 st constituent coder
out_3a	out	1	Data parity out from 1 st constituent coder
out_1b	out	1	Data parity out from 2 nd constituent coder
out_2b	out	1	Data parity out from 2 nd constituent coder
dout_rdyin_n	in	1	Indicates to the module that it's ok to shift data out
dout_rdyout_n	out	1	Indicates that data is available to be shifted out
init	in	1	Initializes the interleaver and loads the block size
rate	in	2	Code rate: 00 = 1/2, 01 = 1/3, 10 = 1/4, 11 = 1/6
frame_size	in	3	Indicates size of current block. Initialized with "init" signal.
start_encode	in	1	Triggers the encode operation

Table 3. Module Pinout

Waveforms

Initialization

The frame size defaults to 1784 bits (the minimum number specified in the CCSDS spec). Use the *init* signal and the *frame_size* signal as shown in fig. 4 below to specify a new frame size (in bits) for the component. To ensure proper operation of the device, make sure the input buffers are empty before specifying a new frame size, as the interleaver parameters are computed each time a new value is specified for the frame size. Thus the *init* signal can also be used to perform a device “soft” reset by asserting *init* without changing the *frame_size*.

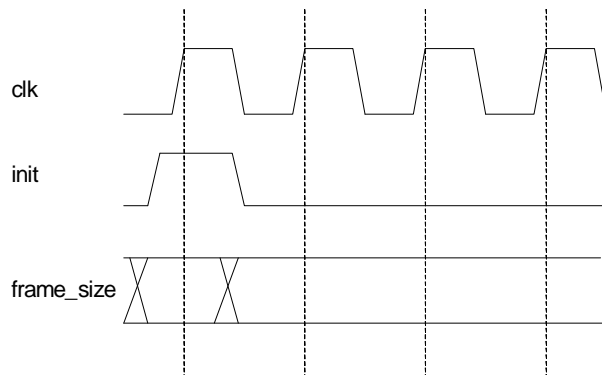


Figure 4. Initialization

In order to read and operate on both normal and interleaved data, an entire frame should be loaded into the input buffer before the encoding process will start. The encoding will start when the *start_encode* signal is asserted. This signal can also be used to re-send a data frame, or to force early termination of a data frame.

If a data frame is terminated with *start_encode* before the number of bits indicated by *frame_size* have been loaded, the device will automatically pad the end of the data frame with zeros.

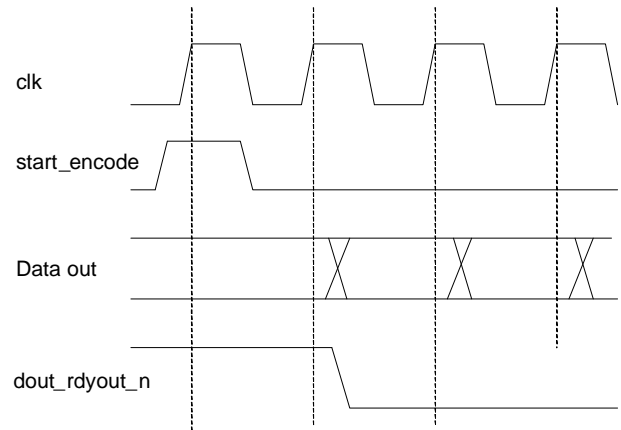


Figure 5. Start Encode

Input Flow Control

The input flow control signals consist of *din_rdyin_n* and *din_rdyout_n*. These signals accompany the data in signal *din*. A ‘0’ on the *din_rdyin_n* indicates to the device that valid data is being shifted into the device on *din*. The device uses *din_rdyout_n* to indicate that it’s able to accept more data. Proper input operation is simple: valid data is flowing into the device’s input buffers when both *din_rdyin_n* and *din_rdyout_n* are logic ‘0’. Because there is an input “pre-buffering” circuit that allows zero-wait-state input operation, the *din_rdyout_n* signal can effectively be ignored in the middle of loading a frame into the device.

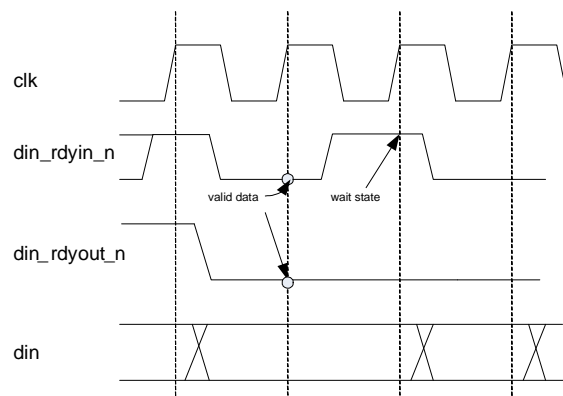


Figure 6. Input timing

Output Flow Control

The output flow control signals consist of *dout_rdyin_n* and *dout_rdyout_n*. These signals accompany the data signals *zk*, *zk*, and *zk_prime* out of the device. A '0' on the *dout_rdyin_n* indicates to the device that it's OK to shift data out of the device. The device uses *dout_rdyout_n* to indicate that it has valid data to send. Proper output operation is simple: valid data is flowing out of the device when both *dout_rdyin_n* and *dout_rdyout_n* are logic '0'. The *dout_rdyin_n* signal can be tied low for fastest operation.

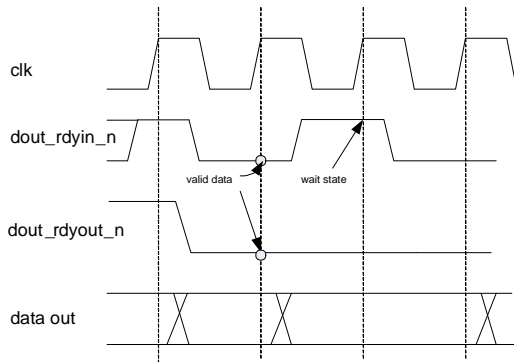


Figure 7. Output timing

Module Verification

The SAL40100ES has been subjected to extensive verification to ensure the highest quality product possible. A comprehensive test plan was implemented which included the following:

- All CCSDS turbo code frame sizes

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

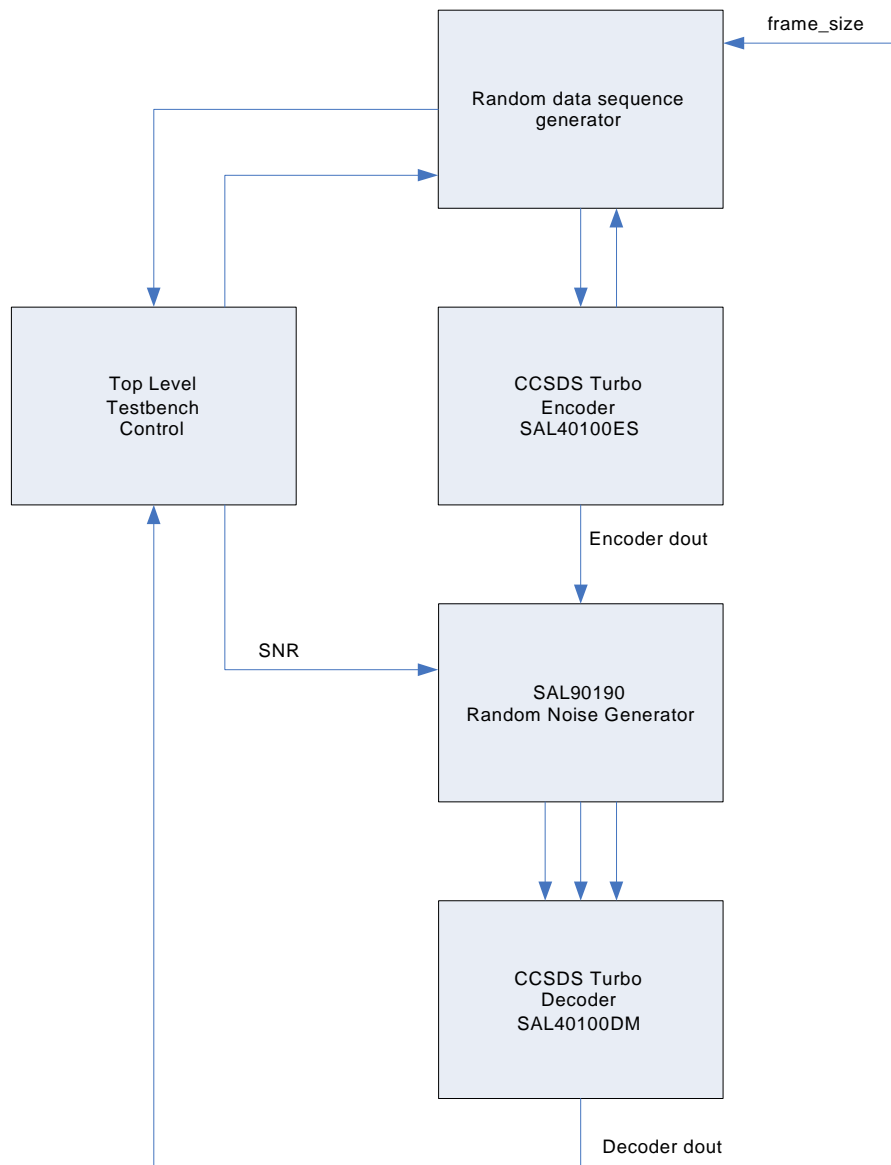
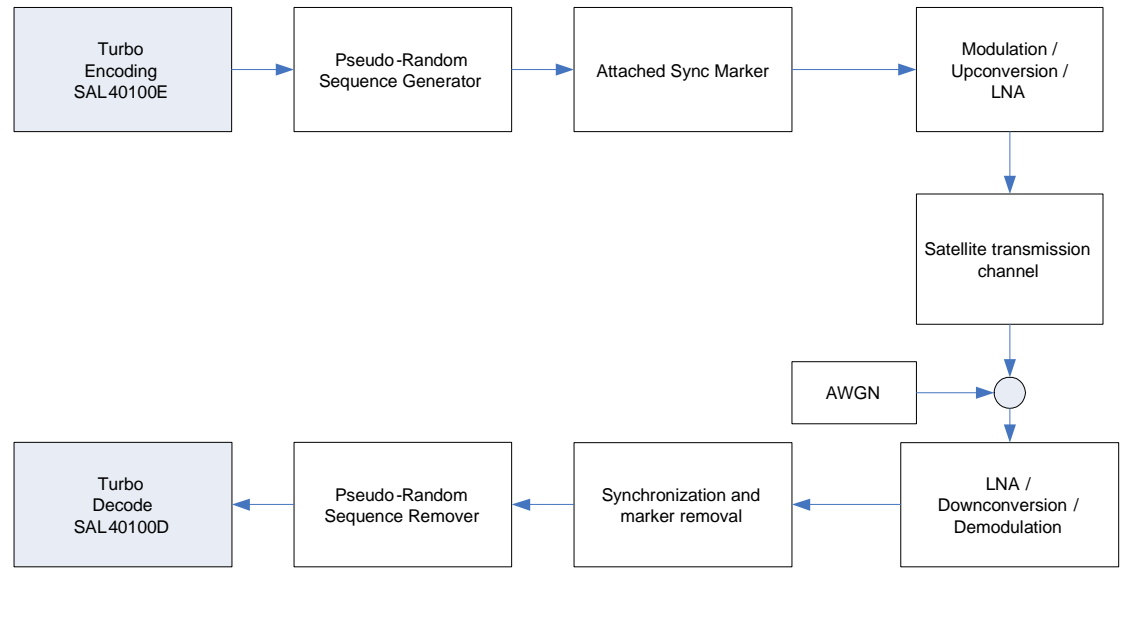


Figure 8: Testbench Block Diagram

Application: Satellite communications

Parallel Concatenated Convolutional Codes (PCCC) offer the best error correction performance of any known codes for code rates < 0.7 and moderate frame sizes (~ 1024 bits). The SAL40100ES is designed specifically for use in space systems.



Ordering Information

Salamander Error Correction currently has 1 CCSDS-compatible turbo encoder IP module available:

SAL40100ES Serial Interface

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, Inc.

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

sales@salamander-ecc.com