

Features

- For use in 3GPP (UMTS, WCDMA) systems. Fully compliant with the 3GPP standard ETSI 25.212 V4.5.0
- Double Buffered input enables high speed operation
- Generic memory interface for easy ASIC integration
- Simple handshake protocol for reliable interfacing
- Fully synchronous design
- Full 3GPP compliant hardware interleaver. No external calculation required.
- Interleaver frame size selectable on a frame-by-frame basis
- Buffer-ready interrupt provided
- Comprehensive verification plan provided
- Full block size range of 40 – 5114 bits supported

General Description

The SAL20100E consists of verilog IP for implementing the parallel concatenated convolutional (turbo) encoder as defined by the 3GPP 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 3GPP specification using the interleave address generator.

The encoder outputs 3 bits for every input bit: an unencoded copy of the data, a parity bit from the first convolutional coder, and a parity bit from the second convolutional coder.

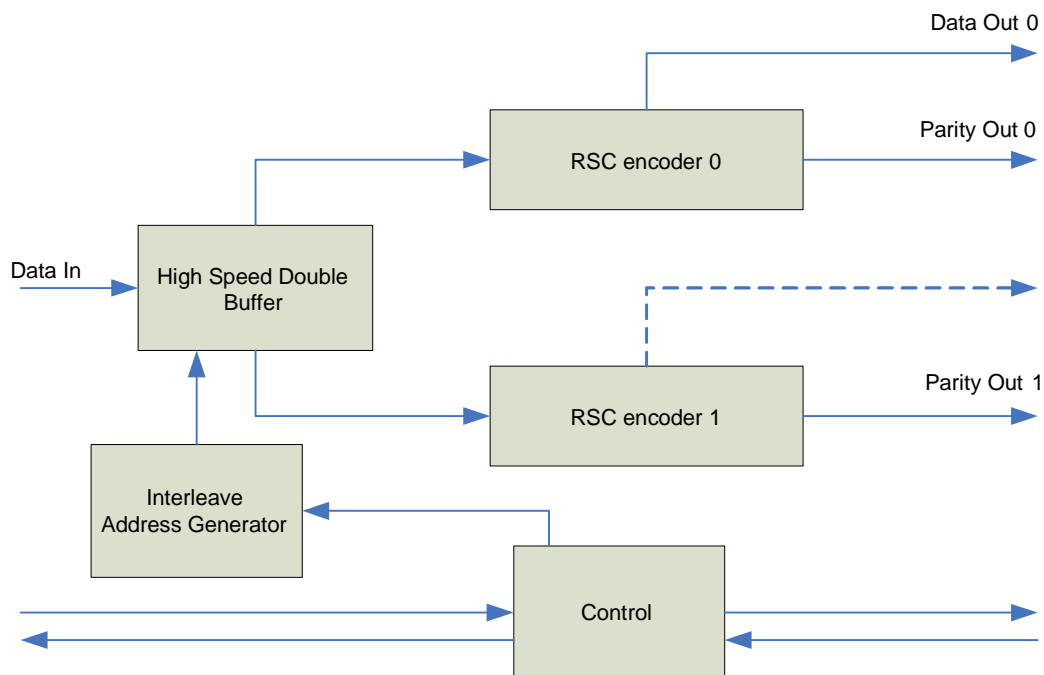


Figure 1: Encoder Block Diagram

Theory of Operation

The RSC encoder is shown in more detail in figure 2 below. It consists of a shift register with feedback connections. An unencoded copy of the data appears at the output of the

coder. Encoders with this property are called “systematic”. The feedback and feed-forward connections represent the operation of binary addition modulo 2.

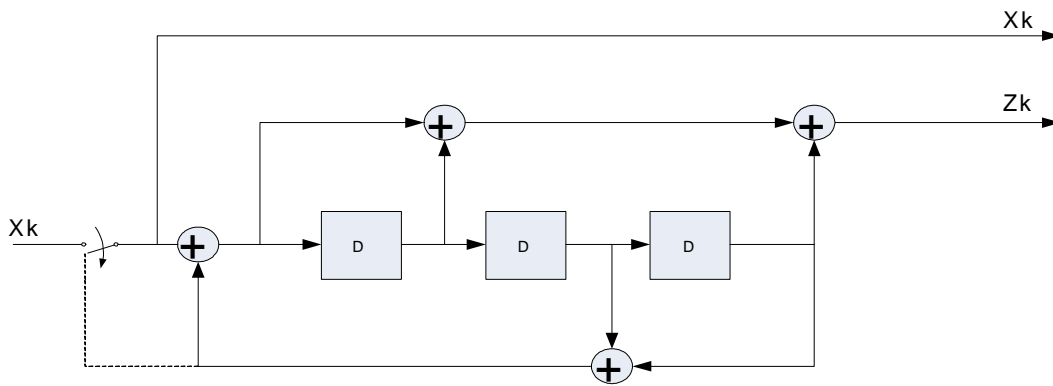


Figure 2 Constituent RSC Coder

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.

Control signals

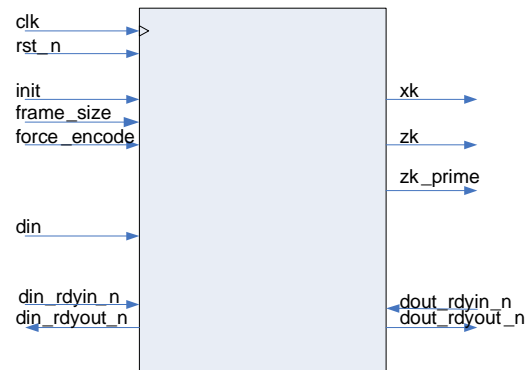
Init, frame_size, and force encoding are control signals whose operation is described below.

Data signals

The din, xk, zk, and zk_prime are the data signals into and out of the encoder.

Flow control signals

The “rdy_in_n” signals are active low indicators to the device that data is available on the input or that the external logic is able to accept data.



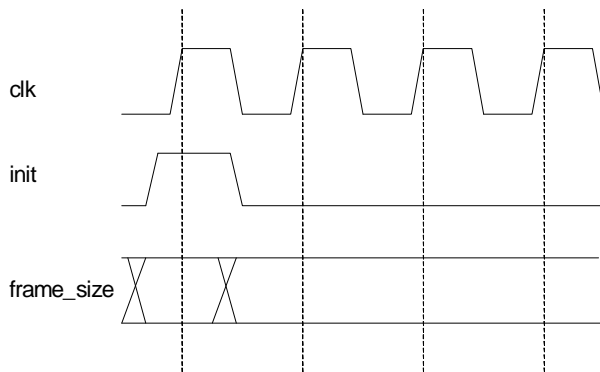
The “rdyout_n” signals are active low indicators from the device that it is ready to accept data on the input or to shift out data on the output.

Pin	Sense	Width	Description
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)
xk	out	1	Systematic data out
zk	out	1	Data parity out from 1 st constituent coder
zk_prime	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
frame_size	in	13	Indicates size of current block. Initialized with “init” signal.
force_encode	in	1	Forces the encode operation

Waveforms

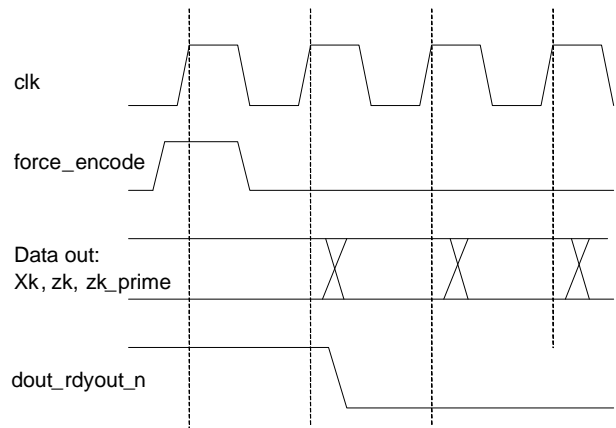
Initialization

The frame size defaults to 40 bits (the minimum number specified in the 3GPP 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*.



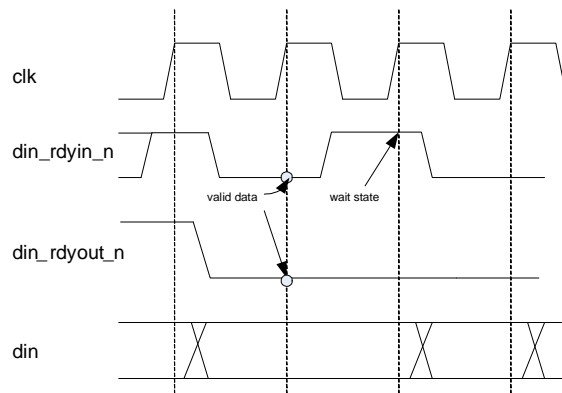
In order to read and operate on both normal and interleaved data, an entire frame must be loaded into the input buffer before the encoding process will start. The encoding will start automatically when the correct number of bits are shifted into the component. Alternatively, the *force_encode* signal can be used to force the encoder to begin encoding the data. This can be used to re-send a data frame, or to force early termination of a data frame. If a data frame is terminated with *force_encode* before *frame_size* number of bits have been loaded, the device will

automatically pad the end of the data frame with zeros.



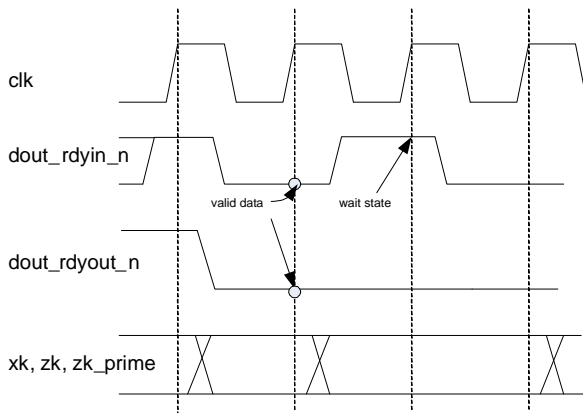
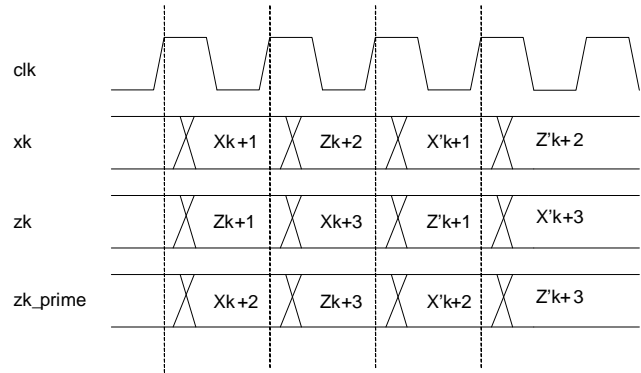
Input Flow Control

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.



Output Flow Control

The output flow control signals consist of *dout_rdyin_n* and *dout_rdyout_n*. These signals accompany the data signals *xk*, *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 tied low for fastest operation.

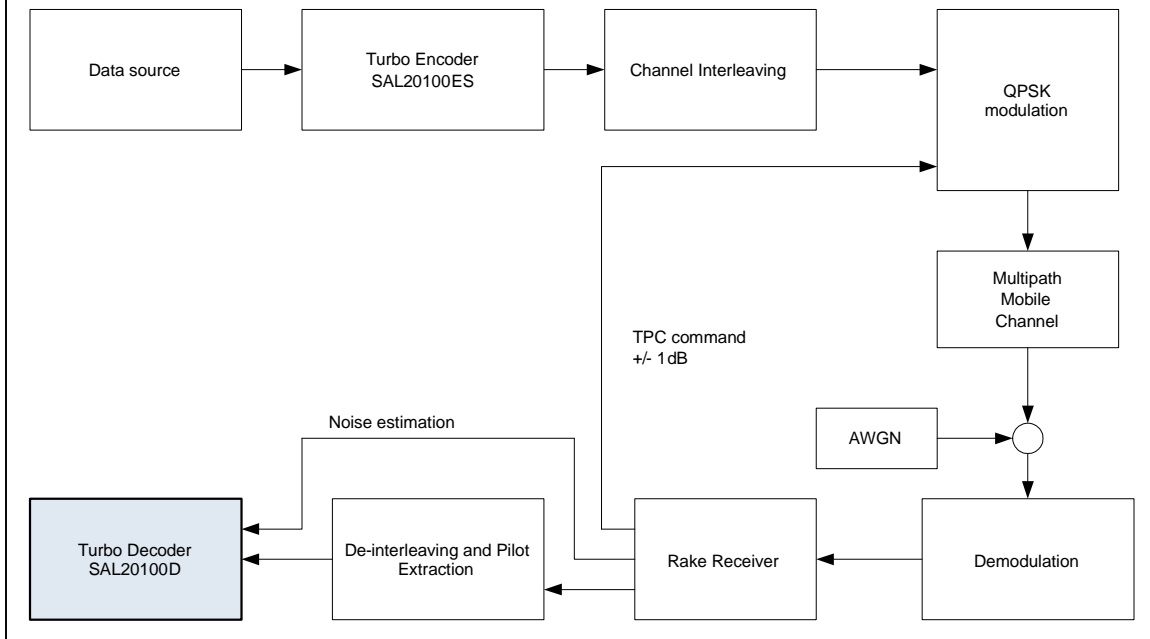


Trellis Termination

The encoding process is completed by forcing the constituent coders to a known state (the 000 state, in this case). The switch in figure 2 is placed in the lower position, thus forcing the input to the shift register to be '0'. The first constituent coder is clocked three times while the second constituent coder is disabled, then the second constituent coder is clocked three times, resulting in a total of twelve tail bits: X_{k+1} , Z_{k+1} , X_{k+2} , Z_{k+2} , X_{k+3} , Z_{k+3} , X'_{k+1} , Z'_{k+1} , X'_{k+2} , Z'_{k+2} , X'_{k+3} , Z'_{k+3} .

Application: UMTS

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 SAL20100E is designed specifically for use in UMTS systems.



Ordering Information

Salamander Error Correction currently has 1 3GPP-compatible turbo encoder IP module available:

SAL20100E 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