



### Features

- $n = 64,800 / 16,200$ , 1-bit symbols
- Capable of correcting (12, 10, or 8) 1-bit symbols
- Soft-decision decoder
- Support of shortened frames
- Errors only
- High-speed, low latency
- Uses the Berlekamp-Massey algorithm
- Simple handshake protocol for reliable interfacing
- Fully synchronous design
- Low speed / low power operation
- Comprehensive verification plan provided

### General Description

The SALxx553D consists of verilog IP for implementing a t-error-correcting BCH forward error correction decoder.

Incoming data are treated as coefficients of a polynomial over the field  $GF(2)$  generated by the product of minimal polynomials over consecutive powers of a field element belonging to  $GF(2^{16})$  or  $GF(2^{14})$

The device uses the known roots of the generator polynomial to construct a set of syndromes which result when the code word is evaluated at each of its  $2t$  roots. Those syndromes are then used to derive a polynomial whose roots are the locations of all the detected errors in the received word. The locations of the errors are then determined using the Chien root locator. The error locations are then used to fix the symbols found to be in error.

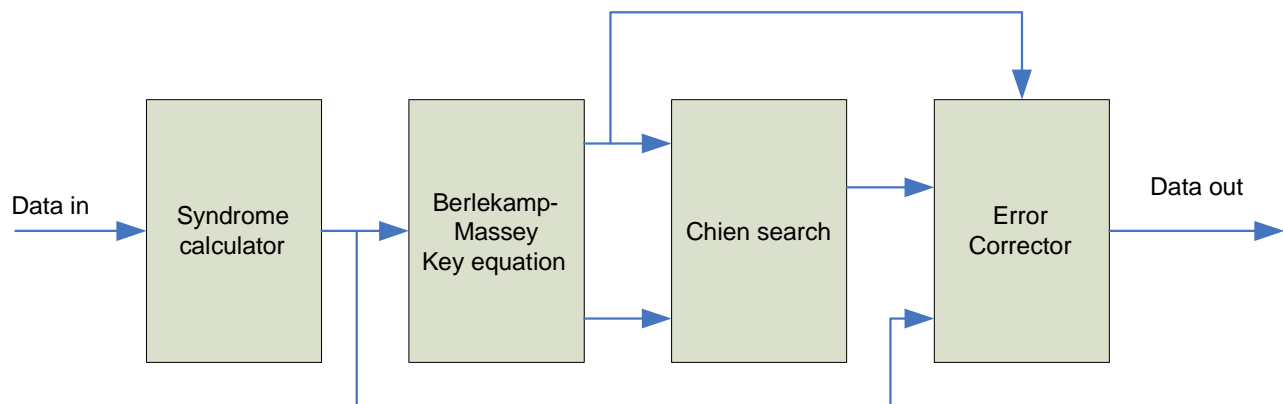


Figure 1: Decoder Block Diagram

## Theory of Operation

The SALxx553d decoder device is capable of decoding BCH codewords of maximum length 64,536 or 16,384. Shortened codewords are supported using the code\_len signal input to the device. There are three possible values for the error correcting capability 't', namely 12, 10, and 8.

The code generator polynomial is determined internally as the product of the first 't' polynomials in table 1, for the 64,800 length code, or the first 't' polynomials in table2, for the 16,200 length code.

$g_1(x)$	$1 + x^2 + x^3 + x^5 + x^{16}$
$g_2(x)$	$1 + x + x^4 + x^5 + x^6 + x^8 + x^{16}$
$g_3(x)$	$1 + x^2 + x^3 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{11} + x^{16}$
$g_4(x)$	$1 + x^2 + x^4 + x^6 + x^9 + x^{11} + x^{12} + x^{14} + x^{16}$
$g_5(x)$	$1 + x + x^2 + x^3 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{12} + x^{16}$
$g_6(x)$	$1 + x^2 + x^4 + x^5 + x^7 + x^8 + x^9 + x^{10} + x^{12} + x^{13} + x^{14} + x^{15} + x^{16}$
$g_7(x)$	$1 + x^2 + x^5 + x^6 + x^8 + x^9 + x^{10} + x^{11} + x^{13} + x^{15} + x^{16}$
$g_8(x)$	$1 + x + x^2 + x^5 + x^6 + x^8 + x^9 + x^{12} + x^{13} + x^{14} + x^{16}$
$g_9(x)$	$1 + x^5 + x^7 + x^9 + x^{10} + x^{11} + x^{16}$
$g_{10}(x)$	$1 + x + x^2 + x^5 + x^7 + x^8 + x^{10} + x^{12} + x^{13} + x^{14} + x^{16}$
$g_{11}(x)$	$1 + x^2 + x^3 + x^5 + x^9 + x^{11} + x^{12} + x^{13} + x^{16}$
$g_{12}(x)$	$1 + x + x^5 + x^6 + x^7 + x^9 + x^{11} + x^{12} + x^{16}$

Table 1: BCH Polynomials for n = 64,800

$g_1(x)$	$1 + x + x^3 + x^5 + x^{14}$
$g_2(x)$	$1 + x^6 + x^8 + x^{11} + x^{14}$
$g_3(x)$	$1 + x + x^2 + x^6 + x^9 + x^{10} + x^{14}$
$g_4(x)$	$1 + x^4 + x^7 + x^8 + x^{10} + x^{12} + x^{14}$
$g_5(x)$	$1 + x^2 + x^4 + x^6 + x^8 + x^9 + x^{11} + x^{13} + x^{14}$
$g_6(x)$	$1 + x^3 + x^7 + x^8 + x^9 + x^{13} + x^{14}$
$g_7(x)$	$1 + x^2 + x^5 + x^6 + x^7 + x^{10} + x^{11} + x^{13} + x^{14}$
$g_8(x)$	$1 + x^5 + x^8 + x^9 + x^{10} + x^{11} + x^{14}$
$g_9(x)$	$1 + x + x^2 + x^3 + x^9 + x^{10} + x^{14}$
$g_{10}(x)$	$1 + x^3 + x^6 + x^9 + x^{11} + x^{12} + x^{14}$
$g_{11}(x)$	$1 + x^4 + x^{11} + x^{12} + x^{14}$
$g_{12}(x)$	$1 + x + x^2 + x^3 + x^5 + x^6 + x^7 + x^8 + x^{10} + x^{13} + x^{14}$

Table 2: BCH Polynomials for n = 16,200

## 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. Because *rst\_n* is designed to be connected to a system-wide synchronous reset, there is an additional soft reset signal, *init*.

### Control signals

Two signals control flow of data into the device, *din\_rdyin\_n* and *din\_rdyout\_n*. The *din\_rdyout\_n* signal indicates that the device is ready to receive data. The *din\_rdyin\_n* signal indicates that data into the device is valid. Valid data is being shifted into the device when both *din\_rdyin\_n* and *din\_rdy\_out\_n* are asserted (low).

Two signals control flow of data out of the device, *dout\_rdyin\_n* and *dout\_rdyout\_n*. The *dout\_rdyout\_n* signal indicates that data out of the device is valid. The *dout\_rdyin\_n* signal indicates to the device that it's OK to shift data out. Valid data is being shifted out of the device when both *dout\_rdyin\_n* and *dout\_rdy\_out\_n* are asserted (low).

### Data signals

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

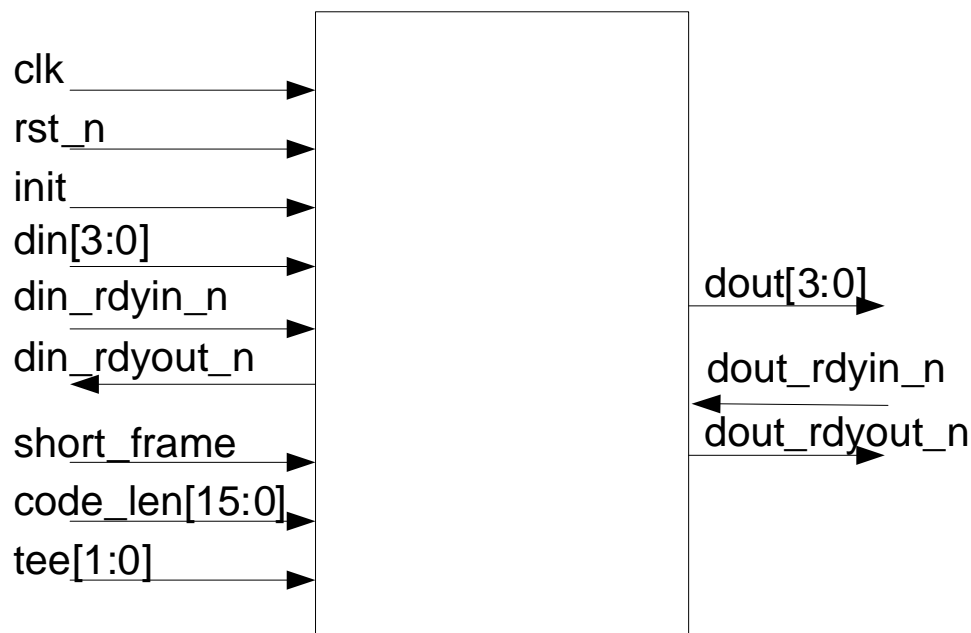


Figure 2: Component pinout

Pin	Sense	Width	Description
clk	in	1	clock
rst_n	in	1	synchronous reset
init	in	1	device "soft" reset
din	in	4	serial data (code word symbol) in
short_frame	in	1	when asserted, frame is based on 16,200, else 64,800
din_rdyin_n	in	1	data in is valid
din_rdyout_n	out	1	device is ready to receive data
dout	out	4	data out
dout_rdyin_n	in	1	interface is ready to receive data from device
dout_rdyout_n	out	1	data out is valid
code_len	in	16	code length
tee	in	2	00 = 12, 01 = 8, 10 = 10, 11 = 12
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 data (as opposed to parity) is being shifted into the device.

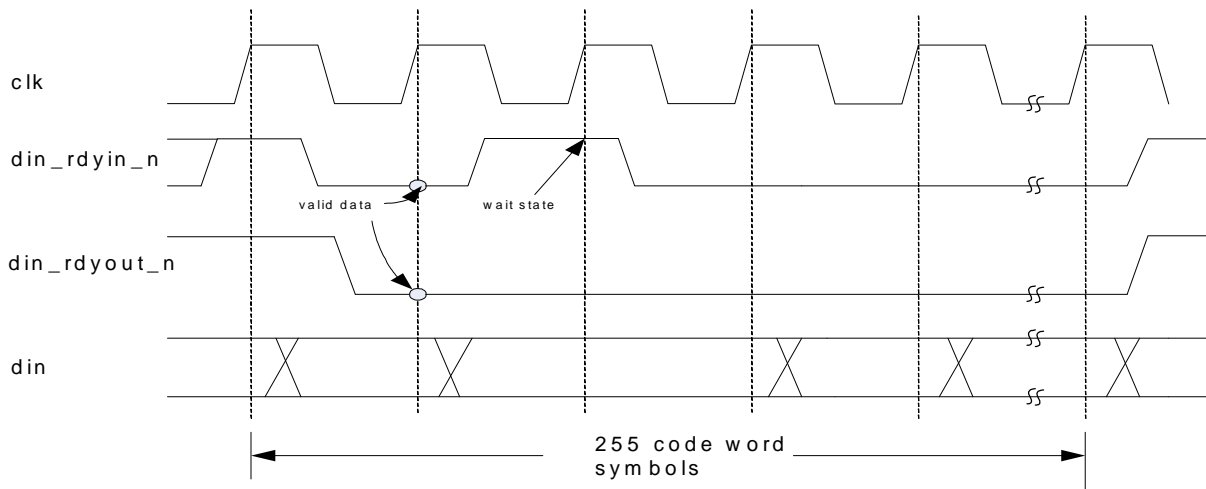


Figure 3: Input timing

### 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 signal the device that it's OK to shift data out.

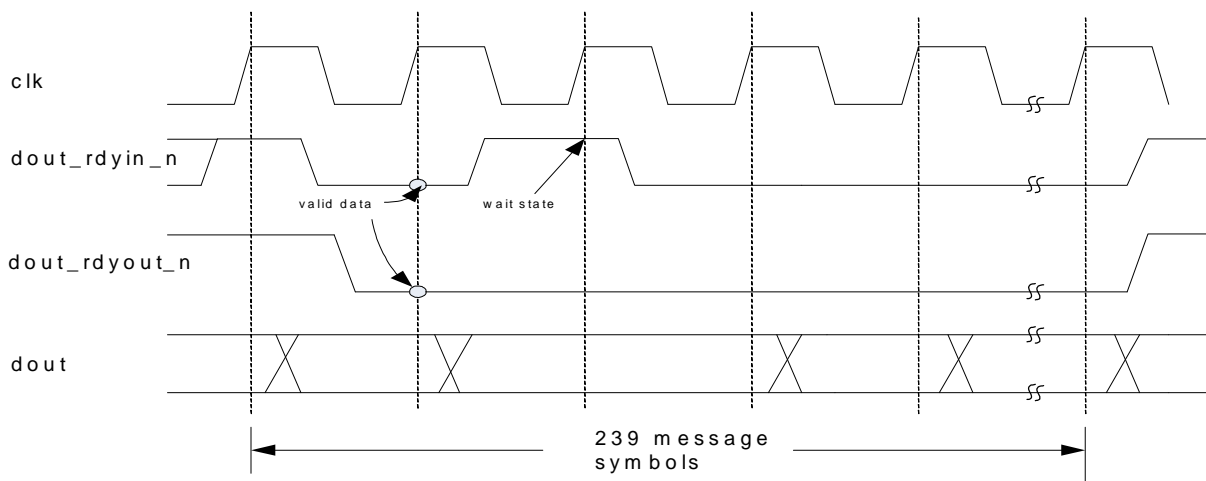


Figure 4. Output timing

## Module Verification

The SALxx553D 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

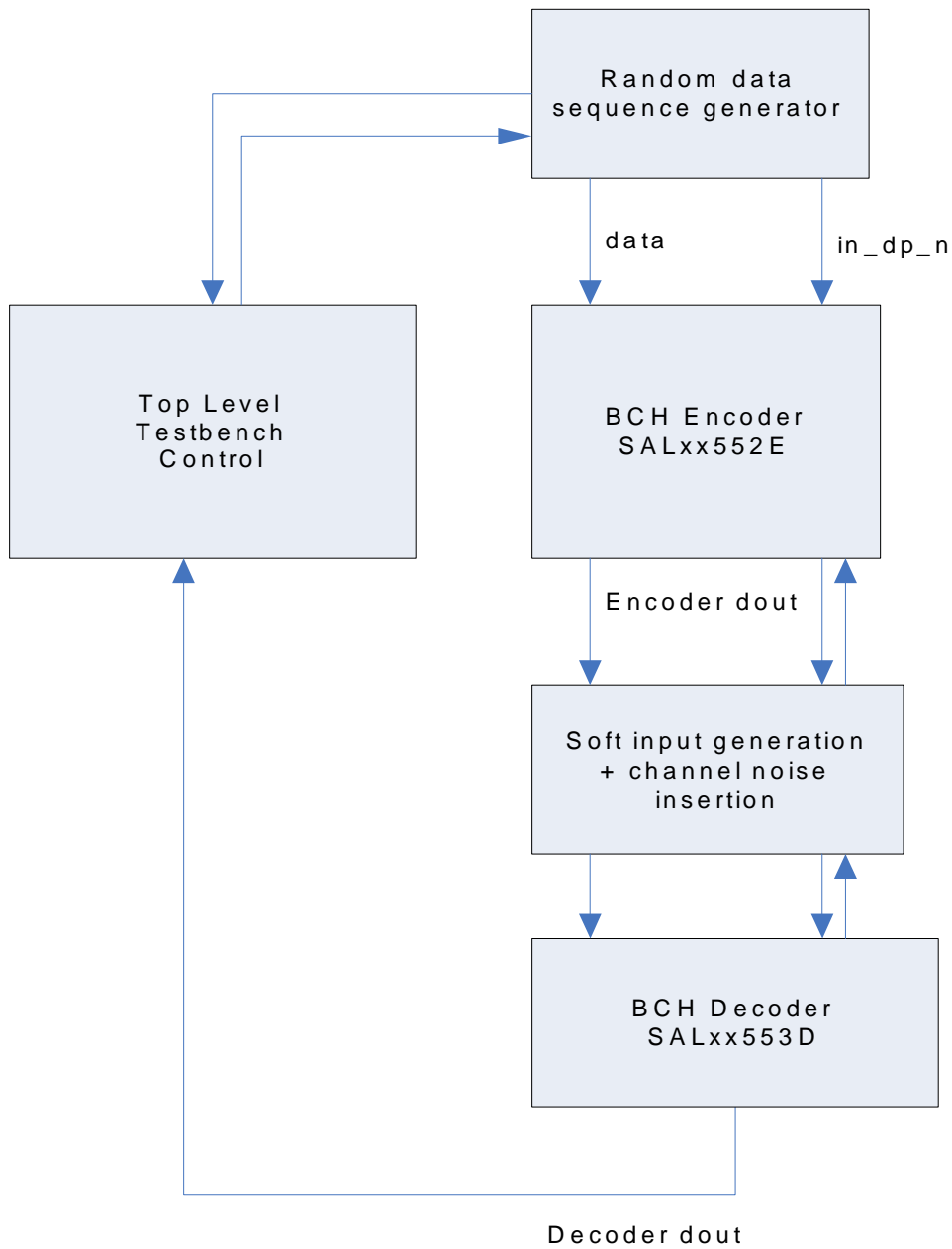
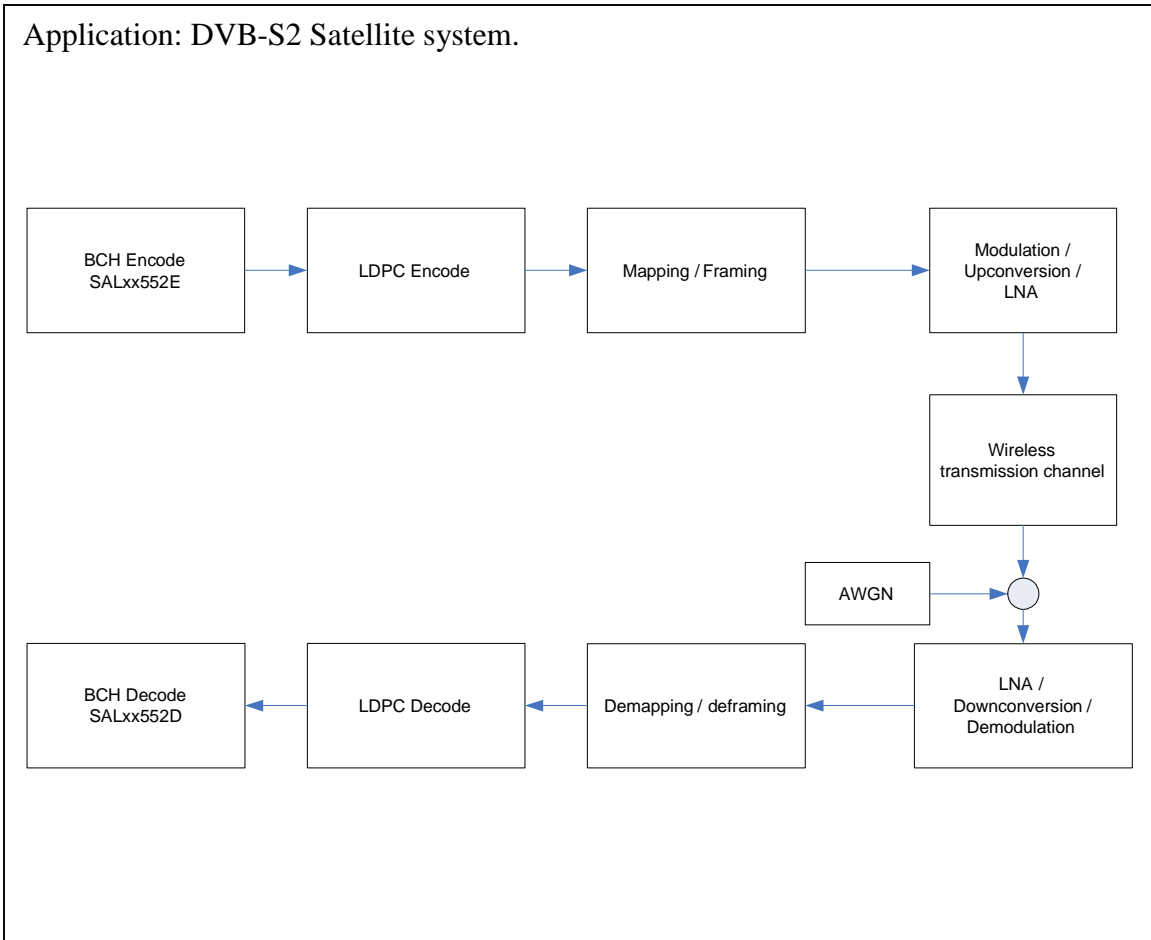


Figure 5: Testbench Block Diagram



## Ordering Information

SALxx553D BCH code for DVB-S2

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:  
 5330 Carroll Canyon Rd., ste 210  
 San Diego, CA 92121  
 Ph: (858) 373-2112  
[sales@salamander-ecc.com](mailto:sales@salamander-ecc.com)