
BLOCK NAME

Verification Plan

Author:

Katrin Nekhin - 318647906

Yuval Saar - 304999261

Revision History

Rev	Description	Done By	Date
1.0	First Draft	Katrin Nekhin	

Table of Content:

1.	General.....	4
2.	Block description.....	5
2.1.	DUT High Level Description	5
2.2.	DUT Block Diagram.....	5
2.3.	DUT Interface.....	6
2.4.	Special Notes.....	Error! Bookmark not defined.
3.	Testbench	7
3.1.	Testbench Concept	7
3.2.	Testbench Diagram	7
3.3.	TB Blocks detailed description	8
3.3.1.	Agents	Error! Bookmark not defined.
3.3.1.	Sequencers	Error! Bookmark not defined.
3.3.2.	Sequences	Error! Bookmark not defined.
3.3.3.	Collectors & Monitors.....	Error! Bookmark not defined.
3.3.4.	Scoreboard.....	Error! Bookmark not defined.
3.3.5.	Reference Module	Error! Bookmark not defined.
3.4.	Resets & clocks	9
3.4.1.	Reset	9
3.4.2.	Clocks.....	9
3.5.	Pass / Fail mechanism.....	9
3.6.	Start / End test mechanisms	9
3.7.	Coverage	Error! Bookmark not defined.

1. GENERAL

The purpose of this document is to describe the Verification Plan of the ENC_DEC_ECC device, including all of its sub blocks and components, including DUT description, Testbench diagram, detailed mechanism description as well as coverage and checkers tables.

The Verification Plan will especially try to address the following guidelines:

- Allow as much as possible Test / Pattern generation to be re-usable from platform to platform
- Verification will be coverage driven, as to allow smart & fast bring up as well as allow visibility & measurability of the verification status

2. BLOCK DESCRIPTION

2.1. DUT High Level Description

ECC_ENC_DEC is a device with the capabilities of encoding and decoding data.

- Features:
 - Encode Operation
 - Decode Operation
 - Full Channel Operation (encode and decode with added noise)

2.2. DUT Block Diagram

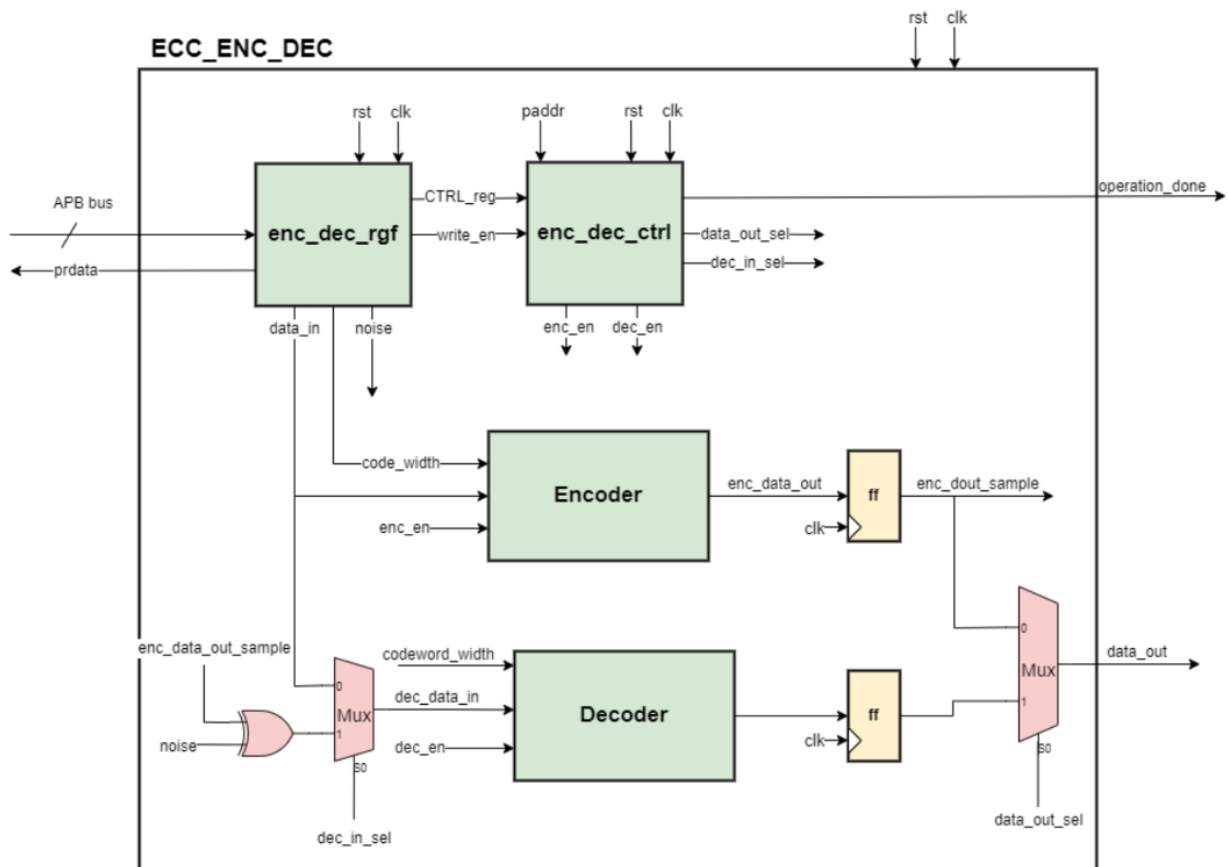


Figure 1 : Block Diagram

2.3. Interface

Interface Parameters:

- AMBA_ADDR_WIDTH
 - Width of the APB address lines
 - Default value 32
- AMBA_WORD
 - Width of the data being sent by APB
 - Default value 32
- DATA_WIDTH
 - Width of a codeword
 - Default value 32

Interface Signals:

- operation_done
 - 1 bit
 - Goes from the DUT to the checker
- data_out
 - 8/16/32 bit
 - Goes from the DUT to the checker
- Clk, rst
 - 1 bit each
 - Goes into every sub-module
- Paddr, pwrdata, penable, psel, pwwrite
 - APB input signals, with different sizes
 - Goes from stimulus into the DUT
- Prdata
 - APB output signals, with size AMBA_WORD
 - Goes from the DUT to the checker
- Num_of_errors
 - 2 bit
 - Goes from the DUT to the checker
- Type_of_work
 - 2 bit
 - Goes from the stimulus to the checker
 - Indicates what mode is being used (encode, decode, full channel)

3. TESTBENCH

3.1. Testbench Concept

The TB handles the following main tasks:

1. Data
 - a. For every work mode (Encode/Decode/Full), randomize 10 data inputs and their noise inputs (if in full channel, distributed equally between 0, 1 or 2-bit noise).
2. Configuration
 - a. Given every data input, configure the registers via APB.
 - b. Send the data input to the golden model.
 - c. The DUT handles the input and sends the output to the checker.
3. Checker:
 - a. Compare the output with that of the golden model.
 - b. Write report.
4. Reset event

3.2. Testbench Diagram

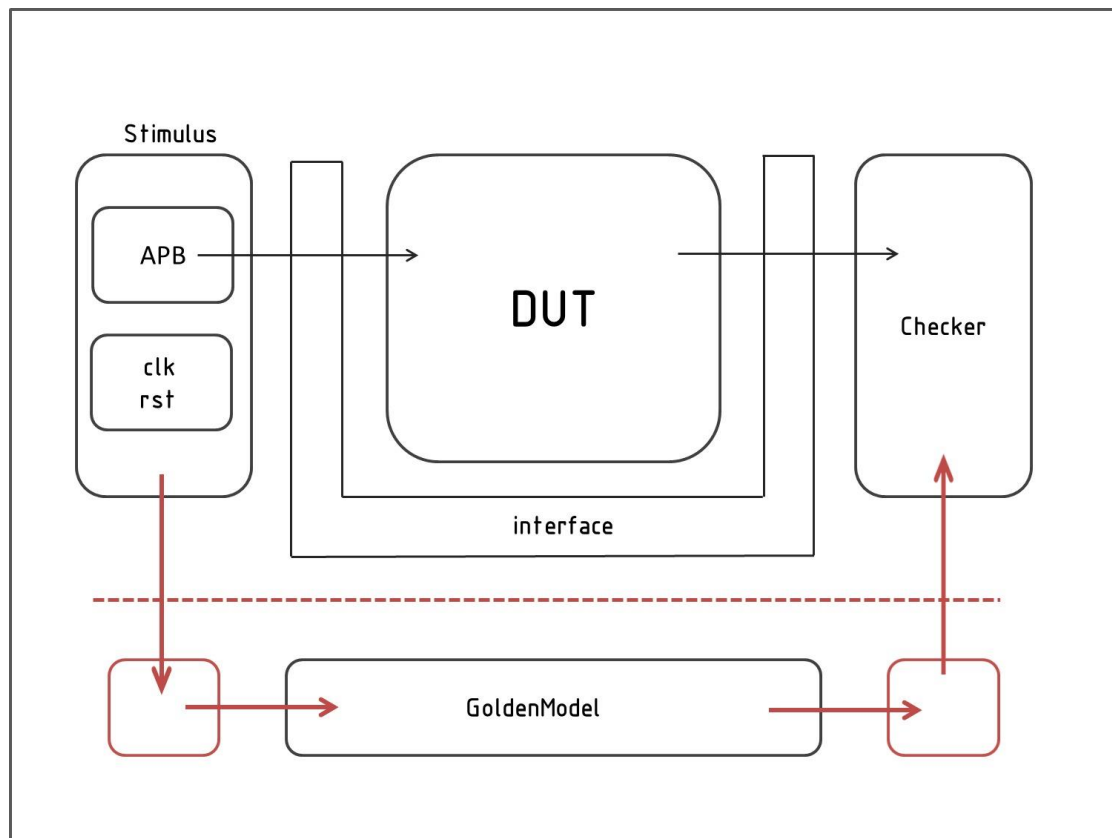


Figure 2: TestBench Active Diagram

3.3. TB Blocks detailed description

3.3.1. Stimulus

Upon initialization the stimulus generates a list of input data vectors using randomization. The list is composed of 10 vectors for each mode.

For full channel mode also the noise register vector is randomized between 0, 1 and 2 bit error vectors.

The stimulus then writes the list to a file for the golden model to read, waits for a while for the golden model to finish calculation and then starts sending the vectors to the DUT, one input vector at a time. Once all of them are sent the stimulus is terminated.

3.3.2. Checker

The checker waits for the golden model to finish calculations and then reads from the output file, one output at a time.

It then compares the output read to the output received from the DUT and writes the result to a report file.

Checker Table:

Condition	Event	Expected Result	Scenario
rst == 1	rst	data_out == 0 && num_of_errors == 0	Standard
type_of_work == 1 && operation_done == 1	Encode Operation	data_out == encoded data_in	Standard
type_of_work == 2 && operation_done == 1 && stim_num_of_errors == 0	Decode Operation No Noise	data_out == decoded data_in && num_of_errors == 0	Standard
type_of_work == 2 && operation_done == 1 && stim_num_of_errors == 1	Decode Operation With 1-bit Noise	data_out == decoded data_in && num_of_errors == 1	Standard
type_of_work == 2 && operation_done == 1 && stim_num_of_errors == 2	Decode Operation With 2-bit Noise	num_of_errors == 2	Standard
type_of_work == 3 && operation_done == 1	Full Channel Operation	data_out == data_in	Standard

3.3.3. Golden Model

The golden model is a MATLAB application which reads data vectors from a text file, performs encoding/decoding on the files according to the operation needed, and writes the results to an output file.

3.4. Coverage

Coverage Table:

Function	Event	Coverage Point	Bins	Scenario
Reset	rst	rst	0,1	Standard
Control	posedge clock	ctrl	0:2	Standard
Input data	posedge clock	data_in	0,[1:255],[256:65535],[65536:\$]	Standard
CODEWORD_WIDTH	posedge clock	codeword_width	0:2	Standard
Add noise	posedge clock	add_noise	0,1	Standard
Num of errors	posedge clock	err_num	0,1,2	Extreme

3.5. Resets & clocks

3.5.1. Reset

- rst_n
 - Active low “cold_reset”, synchronized to apb_clk, resets registers (and datapath?).

3.5.2. Clocks

- APB clock
 - “always on”
 - Operational clock
 - Register configuration

3.6. Pass / Fail mechanism

Simulation will fail if one of the following events will occur:

- Data does not match expected data – Either by checker compare or by post run script
- Assertions fail

3.7. Start / End test mechanisms

- Start of test will be out of reset
- End of test will be one of:

- No more transactions
- Error
- Timeout by built in TB timeout