Aswin Raj K (N18913561, ar7997@nyu.edu)

Note: Zoom in to see the plots clearly

RC5 Encryption and Decryption

RTL design for RC5-32/12/16 is implemented and validated using test bench. The Encryption module is written in VHDL and the decryption module is written in Verilog. The test bench provides the module with 100 different random inputs as test cases. The design uses active low synchronous clear.

Design

The design for RC5 encryption and decryption mainly consists of the ROM containing the expanded keys and the left/right data dependent rotate. The architectural diagram for the modules is as shown in figure 1.

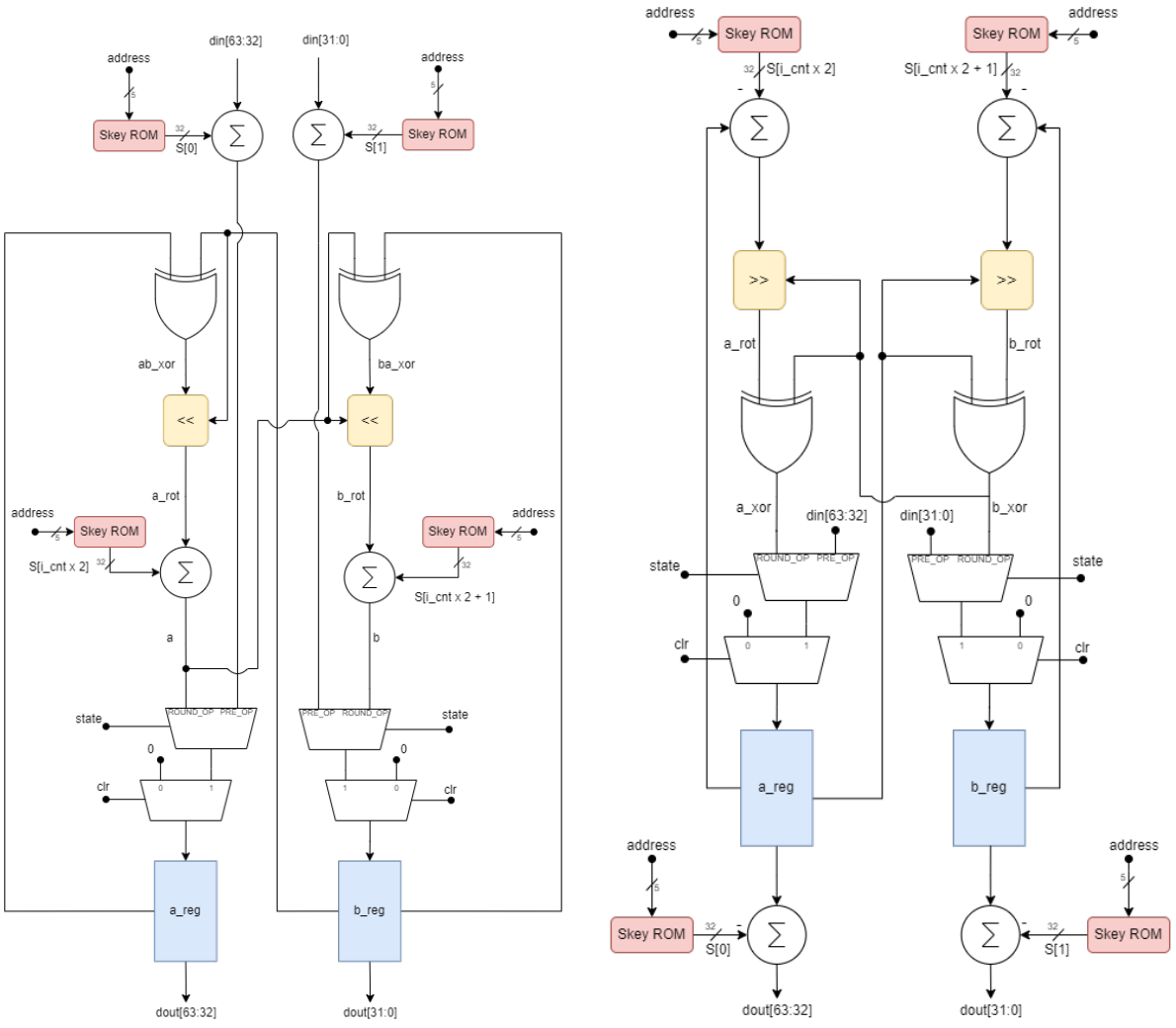
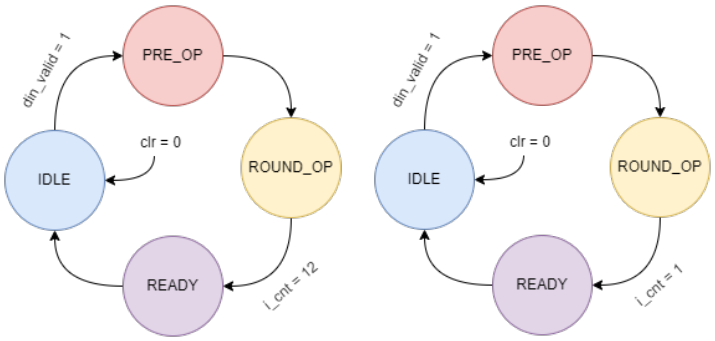


Figure 1: Data Path for a) Encryption b) Decryption

The design makes use of FSM. The FSM diagram is as shown in figure 2.



(a) (b)

Figure 2: FSM for a) Encryption b) Decryption

Test Cases

Test cases for the design validation is created using Python. It contains 100 text and its encrypted values. The first and second element of each line in the 'testCases.mem' is the text and its encrypted text respectively. The test bench for the encryption feeds in the text to be encrypted into the encryption module and checks its output against the encrypted text. Similarly, the test bench for the decryption feeds in the encrypted text into the decryption module and checks its output against the text. If all the test cases are passed it prints out the result in the Vivado Tcl console window.

Simulation Results

Data Dependent Left Rotate

The simulation result for the encryption is as shown in figure 3.

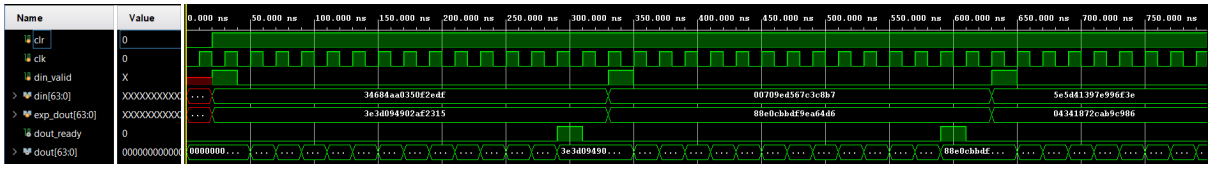


Figure 3: Encryption simulation result for the first two test cases

The simulation result for the Decryption is as shown in figure 4.

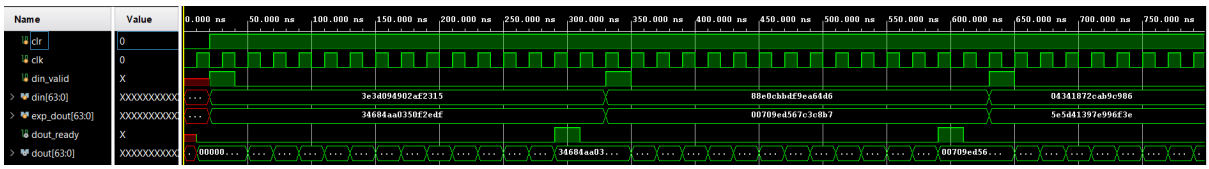


Figure 4: Decryption simulation result for the first two test cases

It is clear from figure 3 and 4 that the dout_ready goes high few clock cyles after a valid input, indicating the ouput is ready. The result obtained for the encryption and decryption module is as expected. Each of the module passes all the 100 test cases contained in the 'testCases.mem' file.

Summary

- Encryption module is written in VHDL and the Decryption module is written in Verilog.
- testCases.mem contains 100 text and its cipher text for the design validation.
- The designed encryption and decryption module works as expected.

Resources

Click on this link for the folder containing Working video, Verilog, VHDL files and Python files (used for generating the test cases).