Anant Singh
Advanced Hardware Design
November 11, 2021

Homework 5

1. Architecture and FSM

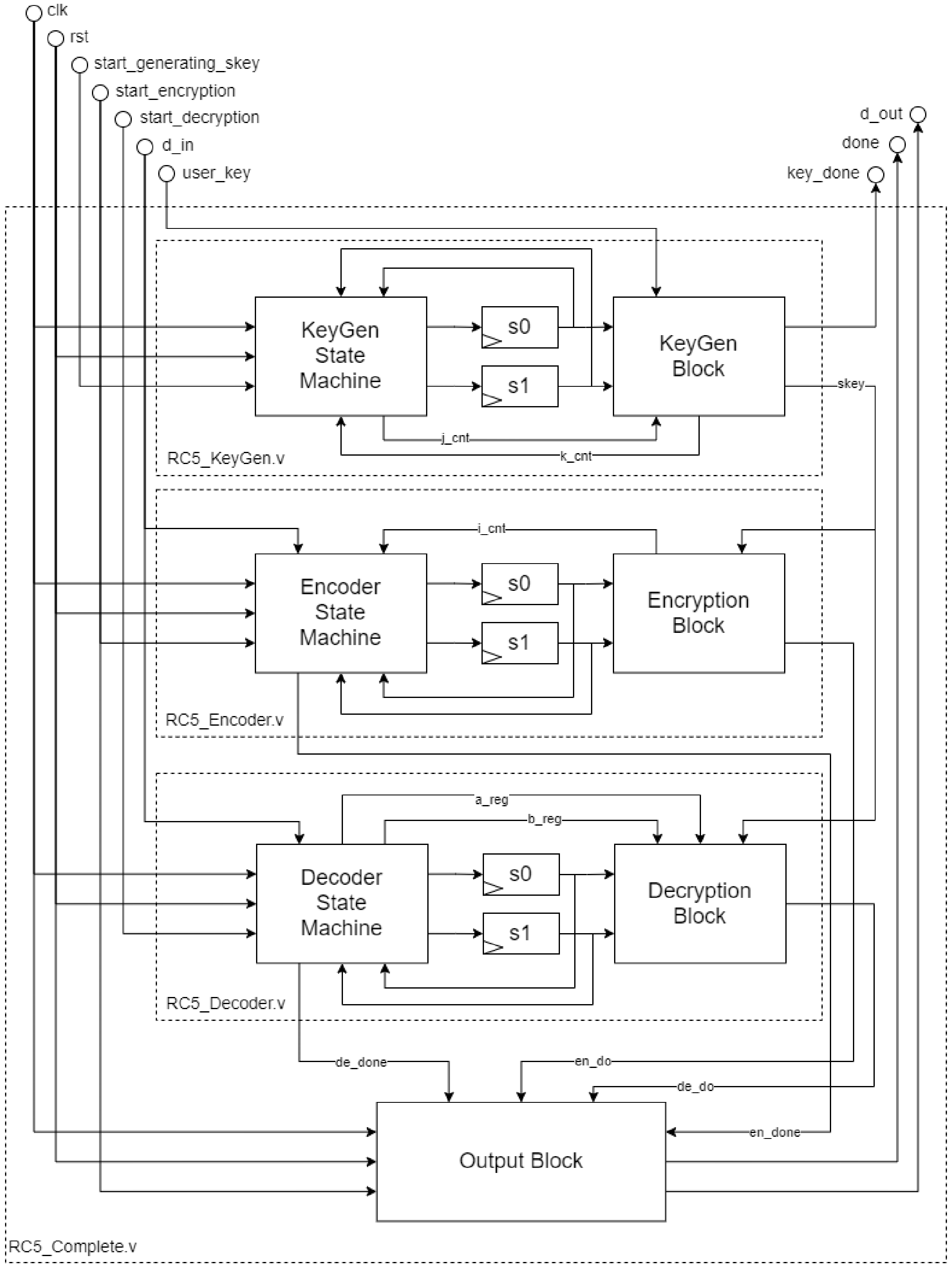


Image 1: Architecture RC5_Complete.v

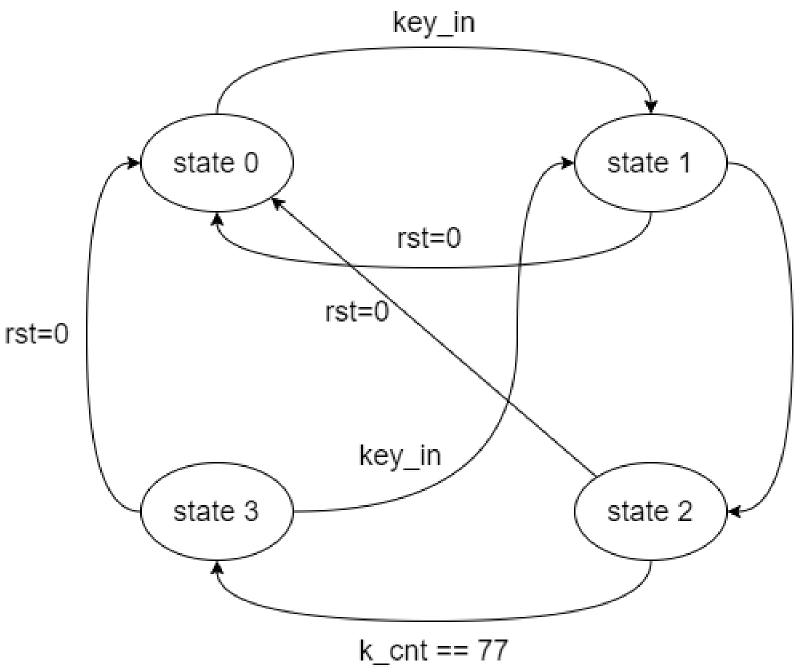
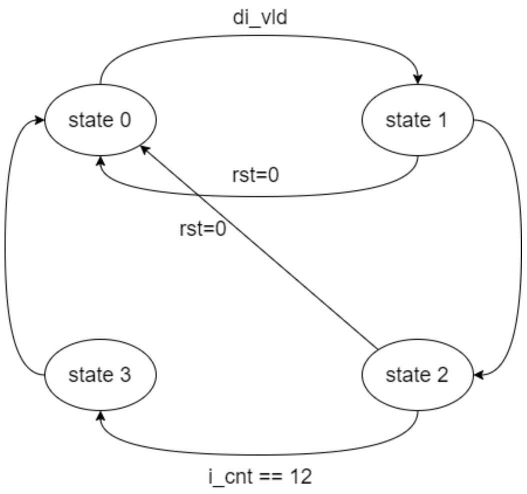


Image 2: KeyGen State Machine



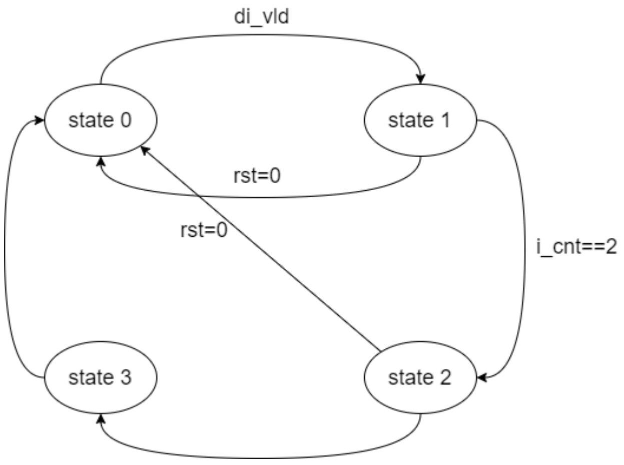


Image 3: Encoder State Machine

Image 4: Decoder State Machine

2. Code Description and Justification

Top Module working: This module initializes 3 submodules -1. RC5_KeyGen (Responsible for generating secret key from 128-bit user key) 2. RC5_Encoder (Responsible for Encryption) 3. RC5_Decoder (Responsible for Decryption). The modules are designed in such a way that start_generating_skey, start_encryption or start_decryption signals cannot be given at the same time or before the done signal, only one function is accessible either Key Generation or Encryption and Decryption. start_encryption or start_decryption signals can only be given once the key _done signal becomes high or the output values will be garbage. Depending upon the signal request i.e start_encryption or start_decryption , d_out and done are selected and correct outputs are assigned.

Key Generator working: Key generation starts working on receiving a start_generating_skey signal, and expects a 128-bit user_key on next clock cycle. KeyGen State Machine comes into action and after initiating the s_arr_tmp from magic constant and loading the user key into l_arr state again changes and 78 cycles of key generation begin. This process is controlled by 3 counters i (0-25), j (0-3) and k (0-77) and on the last cycle, state changes to Ready state. As soon as the ready state becomes active output key_rdy becomes high. A ram skey is updated with newly generated secret key. At the end RC5_KeyGen.v generates an array of 26 32-bit values for encryption.

Encoder working: Encoder has 4 state FSM, Idle, First-Encryption, Encryption and Encrypted state. Encryption starts when start_encryption signal is received, upon which Encoder State Machine changes state and encryption begins by initiating a_reg and b_reg with skey[0] and skey[1] values followed by 12 round key encryption cycles. State again changes on 12th cycle and state changes to encrypted where decryption is completed. The signal en_done becomes high and passed to top module’s output block along with the encrypted 64-bit en_do. Then finally State returns to Idle.

Decoder working: Decoder has 4 state FSM, Idle, Decryption, Last-Decryption and Decrypted state. Decryption starts when start_decryption signal is received, upon which Decoder State Machine changes state and decryption begins by initiating a_reg and b_reg with 64-bit input d_in divided in 2 halfs, followed by 12 round key Decryption cycles. State again changes on 2nd cycle and state changes to decrypted where decryption is completed. The signal de_done becomes high and passed to top module’s output block along with the encrypted 64-bit de_do. Then finally State returns to Idle.

Testbench working: Testbench includes 2 processes out of which on is to generate clock signal, the other process is to test the function of KeyGen, Encoder and Decoder modules. The testcases are loaded from text file using built-in library in Verilog. The testing process includes an extensive last test which is a loop to test 25 values for 4 different keys for all the three modules by comparing file input against the correct value of encryption/decryption. Each loop contains loading of Input and output from file to test d_out values. If All the Tests are passed All test Passed is displayed and simulation ends. Testbench includes other tests –

Test 1 - Encryption only, Test 2 - Decryption Only, Test 3 - Both Encryption and Decryption, Test 4 - Reset Testing, Test 5 - Reset in between of Key Generation cycle, Test 6 - Reset in between of Encryption, Test 7 - Reset in between of Decryption, Test 8 - Encryption in between of Decryption cycle, Test 9 - Decryption in between of Encryption cycle, Test 10 - Extensive Both Encryption and Decryption with different user keys.

New Features: Now the whole RC5 encryption is combined into a single module and it supports all the functions namely Key Generation, Encryption and Decryption. Since all the modules are now interconnected the complexity of Encoder and Decoder is increased. A FSM is new to Encoder and Decoder to control the process. Key generation modules is new, there were fixed secret key for Encoder and Decoder but now a new secret key can be generated by user key. There are extra control signals now for controlling the module which were not required earlier as the modules were separated.

Resource Utilization: As per the Resource Utilization report summary (for Nexys 4DDR FPGA) - LUT utilization is 1959 (3%), 1255 (1%) Flip Flops are being Utilized. Input Output ports are overutilized, there are only 210 ports but the design uses 263 (125%) IO ports. 1257 Registers are being used in total, most of which (1045) are being used in key generator.