In the name of God

ASIC/FPGA Chip Design: Project #1

Due on April 13, 2020 at 12:00pm

Professor Mahdi Shabani

Alireza Shirzad
95101847
ee.sharif.ir/∼alireza.shirzad

Assumptions

In this phase it is assumed that the data is going to come and go from TX and RX module in a serial manner. And also the architecture and FSMs are exactly the same in both TX and RX except that the scrambler in the TX is replaced by the Descrambler.

State Machine

4 states are defined in both TX and RX:
-IDLE: the modules are in the idle state when there are nothing to transmit and receive.
-INITIALIZE: With the first signal of request line and the 12 bits of length should be also ready in the input to be flopped. Then the machine goes to INITIALIZE state. the scrambler and descrambler are reset to their initial values and ready to receive data.
-SCRAMBLING: The machine goes to the state of scrambling and stays there for L clocks which L is the length register specified in the INITIALIZE state. in this state the bits are serially fed to the scrambler/descrambler until the counter reaches L. during these L clocks the ready signal is 1 meaning there is an stream of data coming out of the module.
-FINISH: After scrambling the machine goes to finish state and sets the ready signal to 0. after this state the machine goes to IDLE state.

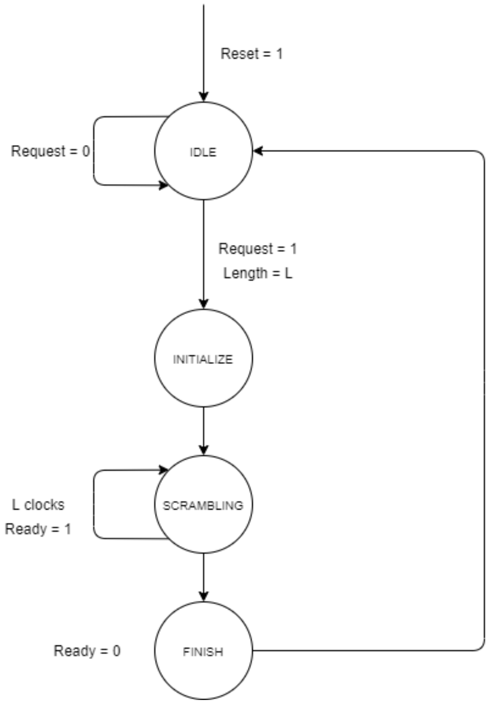


Figure 1: TX state machine

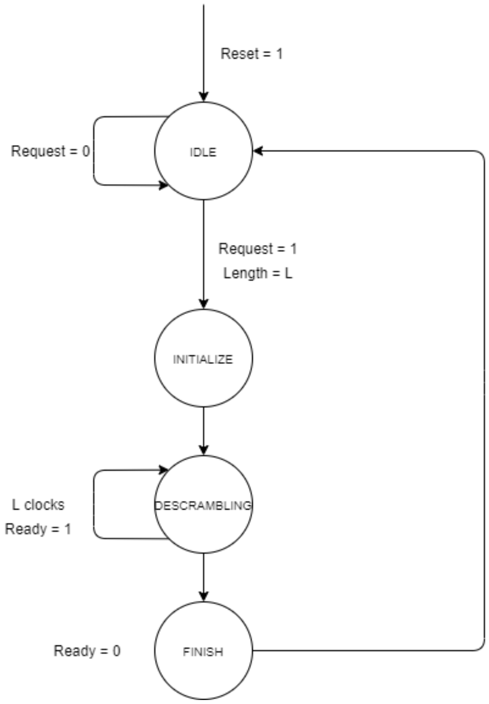
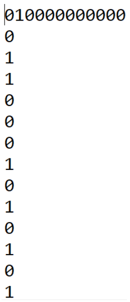


Figure 2: RX state machine

Testbench

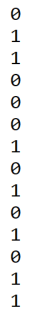
In the testing phase we create a file using MATLAB in the following manner:



```
010000000000
0
1
1
0
0
0
1
0
1
0
1
0
1
```

Figure 3: An example of test file generated by matlab and fed to the testbench

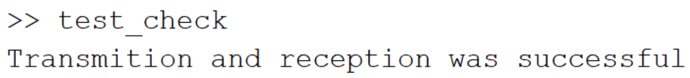
The first line is the length vector and there is L lines bellow that containing the serial data. These data are fed to TX module and the the output of the TX module are fed to the input of the RX module. The result of receiver are save in the result file. this result file is a simple stream of output illustrated in the following figure:



```
0
1
1
0
0
0
1
0
1
0
1
0
1
1
```

Figure 4: An example of result file generated by testbench and fed to the MATLAB

Then this result is checked line by line, by a MATLAB script and if the received data is the same as the transmitted data then MATLAB outputs the following line:



```
>> test_check
Transmition and reception was successful
```

Figure 5: An example of MATLAB final output

Module view

The final module architectures are depicted in the following figures:

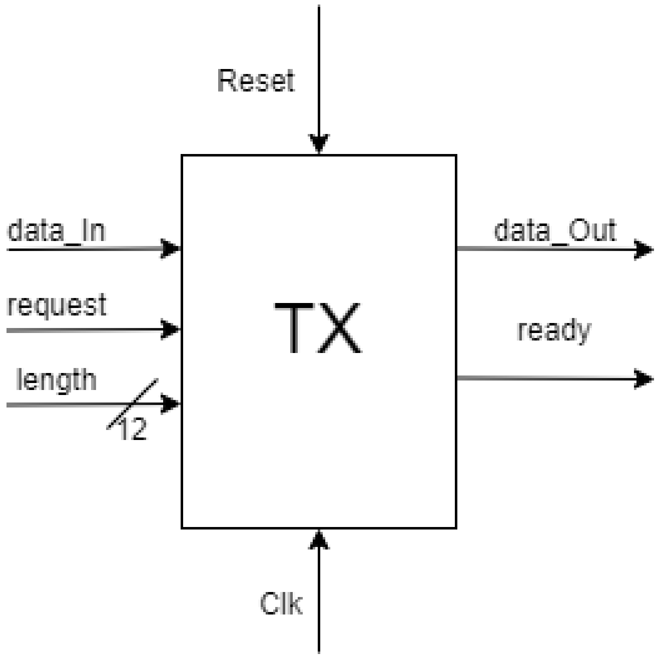


Figure 6: Transmitted module view

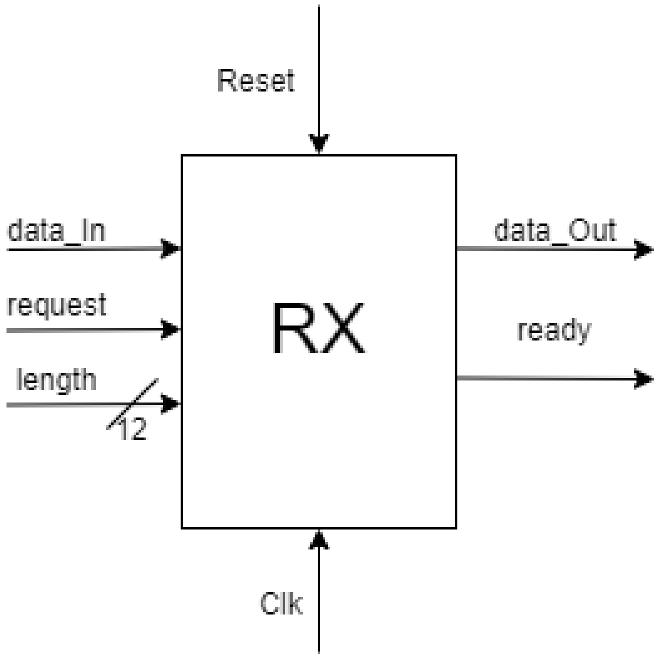


Figure 7: Receiver module view