Frame Check Sequence

Verilog implementation

Made by: Youssef Galal

Contents

1- Introduction 2

A) Generator Polynomial 2

B) Calculating CRC-16 2

C) Digital Implementation 2

2- Designing the system 3

A) Main parts 3

1- The Counter 3

2- The shift register 4

3- Testing the RTL 5

A) For Data Size between 64 and 1024 5

B) For Illegal Data Size 5

1- Introduction

Frame Check Sequence is an error detection field that is added to the transmitted data to be sent to the other end to determine whether the data was sent correctly or not. FCS Standard uses a 16-bit ITU-T CRC method to calculate this field.

A) Generator Polynomial

The transmit and receive both have a polynomial that they use to calculate this field. For 16 Bit CRC we need a polynomial of degree 16. In our case the polynomial is $G_{16}(x) = x^{16} + x^{12} + x^5 + 1$. For this polynomial we can only use it to detect errors for data sizes bigger than 16. In our case we are using it for data sizes that varies from 64 to 1024.

B) Calculating CRC-16

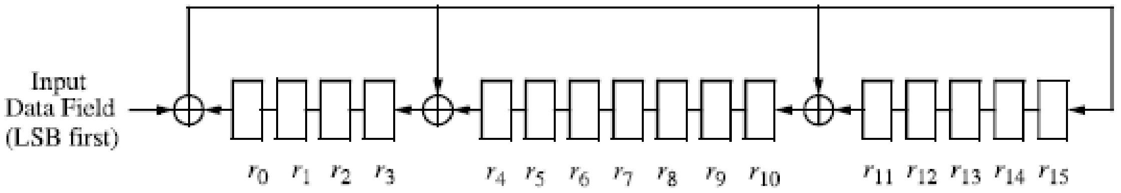
To calculate FCS, we can represent our array of bits as a polynomial

$M(x) = b_0 x^{k-1} + b_1 x^{k-2} + \cdots + b_{k-2} x + b_{k-1}$. where b_0 is the first bit transmitted and b_{k-1} is the last bit transmitted.

1- First, we multiply M(x) by x^{16}
2- Divide the output given from step 1 by G_{16}(x) but using modulo 2 division
3- The remainder of the division performed in step 2 is the FCS value.

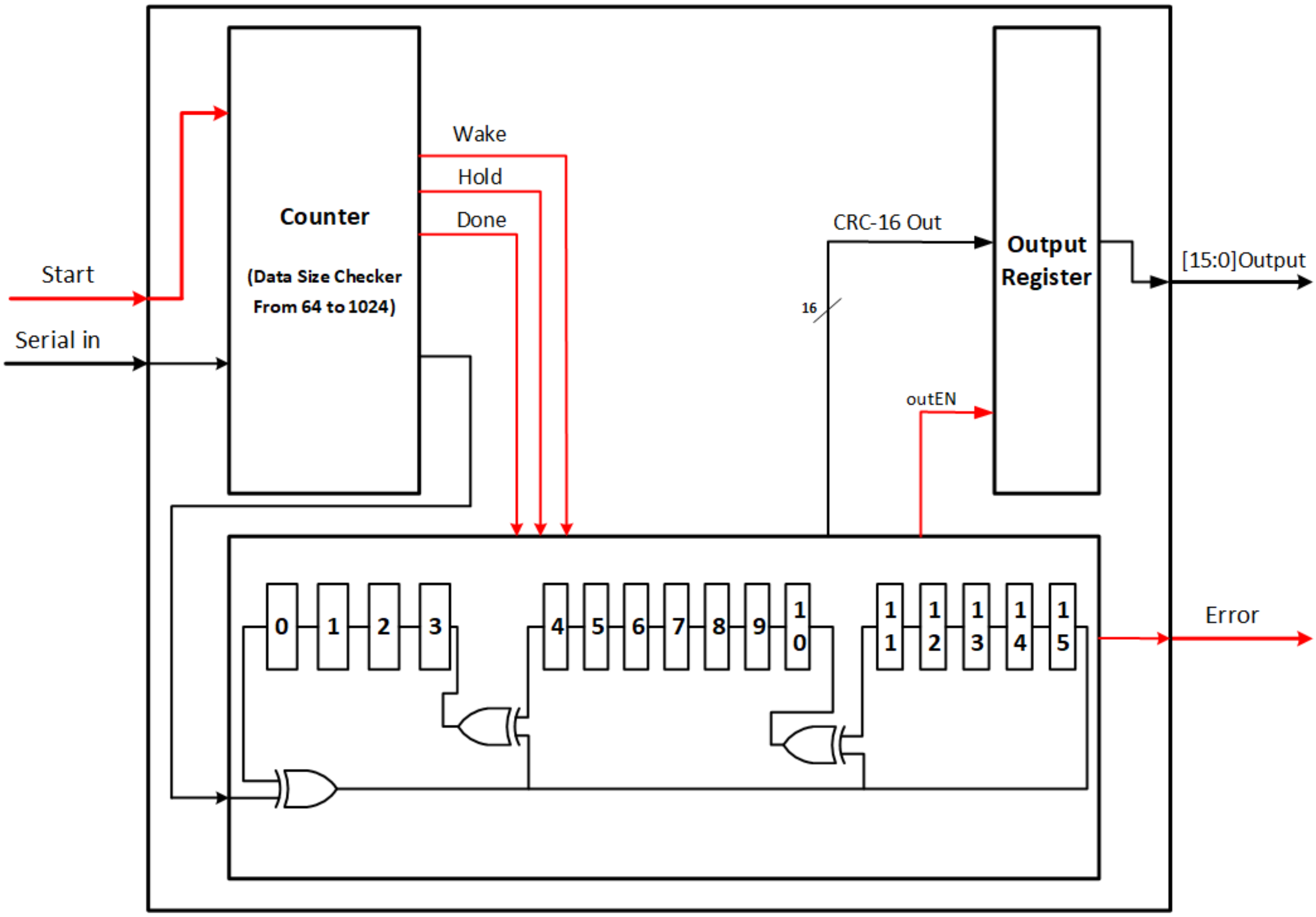
C) Digital Implementation

Implementation of CRC-16 and division can be easily implemented using XOR gates in the corresponding bits of the generator polynomial as shown in the figure below.



2- Designing the system

In the figure below, the design of the system with the input and output signals.



A) Main parts

The system consists of two main modules:

1- Counter
2- Shift register with XOR gates to implement the generator polynomial

1- The Counter

The counter is used to count the input bits to the system as they must be in the size of 64:1024 bits. If the size isn't of the required, it must produce and error

Inputs to this system are (Start) and serial in. Start signal is a signal that the user must put 1'b1 on it while sending the input bits in the Serial in line. The Start signal makes the counter start to count up to count the number of the input bits and also outputs the wake signal to wake the shift register to start calculating the FCS field. When the user finishes inputting the data, the start signal must be 1’b0. In this moment, the counter will compare the value inside it with 64 and 1024 to make sure that the data size lies between them. if the data size is ok, the done signal will be 1’b1 to tell the shift register that the input size is ok and it can output the value inside it.

2- The shift register

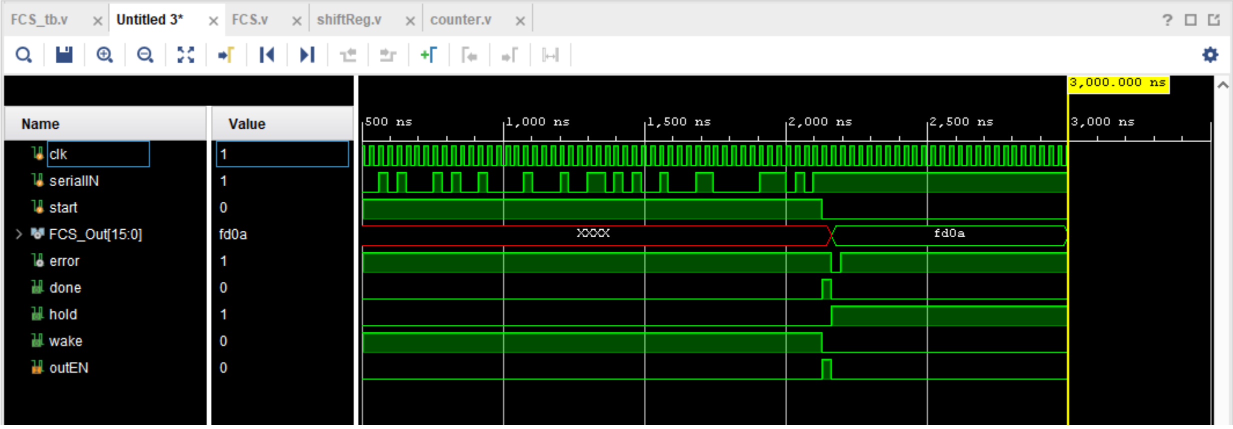
It has the full implementation of the generator polynomial and it receives the control signals from the counter to start or reset itself.

1- Hold signal is always 1’b1 if there are no inputs (Start is 1’b0). This signal keeps the shift register on reset state (all values are = 0).
2- Wake signal is always 1’b0 until the start signal goes 1’b1.
3- Done signal is a signal that counter produces when the input size is ok and we can output the 16 bits stored in the register.

3- Testing the RTL

A) For Data Size between 64 and 1024

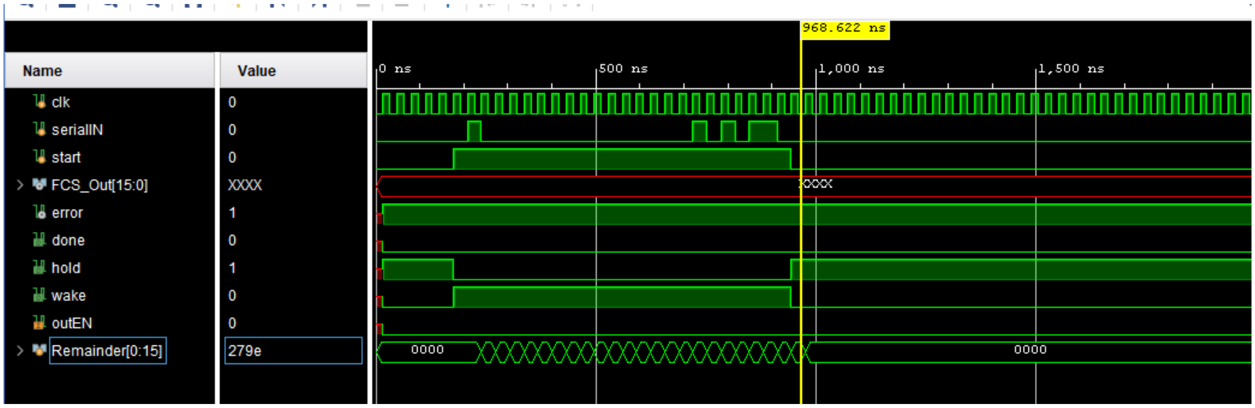
Trying 64 bits input = 0xDAB1452113523075



Error signal is always 1 until a valid data appears in the register to be read. Output is 0xFD0A.

B) For Illegal Data Size

Trying 24 bits input = 0x400056 (the same input used in the standard document)



Output is correct in the shift register (Remainder) 0x279E, but the output of the system didn't come out as the error signal is still 1 because of the wrong data size and also the outEN didn't become 1 to enable the register.