Design and Implementation of Logical Scrambler Architecture for OTN Protocol

Chethan Kumar M, Praveen Kumar Y G, Dr. M. Z. Kurian.

Abstract **- This paper describes design and implementation of logical scrambler architecture for OTN protocol and specifications of their logical resources. The logical scrambler architecture can be designed using serial scrambler architecture, in which registers are connected in parallel to achieve high data transfer rate in OTN protocol. The whole design will be developed using Xilinx ISE 12.2 and will be simulated using Modelsim 6.3c and will be implemented on Virtex 4 FPGA.**

Index terms**: OTN, Logical Scrambler, Serial Scrambler.**

I. INTRODUCTION

The optical transfer networks (OTN) are standards for data transmission over fiber optic links. Due to long sequences of consecutive digits from incoming data streams, the clock signals become low and lead to delay in clock signal. This decrease the data transfer rate in OTN system. Hence complexity increase in OTN system and leads to over consumption of logic resources. Hence a need for clock recovery at the receiver, which in turn requires a guaranteed minimum number of transitions in the incoming serial data stream. The mechanism to achieve this transition density is known as scrambling. And this uses pseudorandom bit sequence (PRBS) circuit to perform scrambling. The basic architecture to achieve this scrambling uses serial scrambler architecture. [5]

This paper presents the suitable solution by implementing logical scrambler architecture. The logical scrambler architecture can be implemented using basic architecture of serial scrambler in which registers are connected in parallel.

II. SERIAL SCRAMBLER

In digital transmission systems, there are always scramblers to scramble the transmission data. In general, multiples of base rate signals are multiplexed and then scrambled before transmission which is descrambled and demultiplexed after reception. Scrambling used to be done serially.

The scrambler diagram shown in Fig.1 shall generate a continuous stream of output bits at the same rate as the transmitted bit rate.

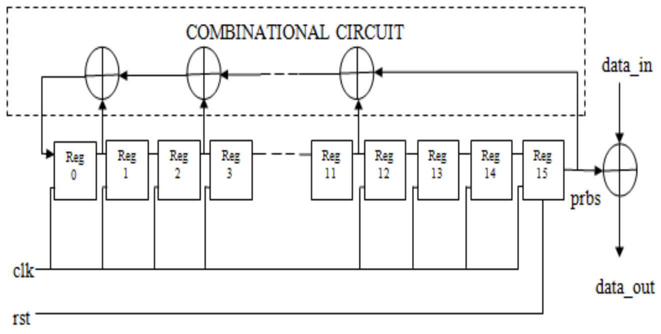

Fig.1 Serial scrambler block diagram [6]

The scramblers transform the input data stream by applying a pseudo-random binary sequence (PRBS) (by modulo-two addition). Sometimes a pre-calculated PRBS stored in the Read-only memory is used, but more often it is generated by a linear feedback shift register (LFSR).In order to assure a synchronous operation of the transmitting and receiving LFSR (that is, scrambler and descrambler), a sync-word must be used. A sync-word is a pattern that is placed in the data stream through equal intervals (that is, in each frame). A receiver searches for a few sync-words in adjacent frames and hence determines the place when its LFSR must be reloaded with a pre-defined initial state.

III. LOGICAL SCRAMBLER

Fig.2 describes a general block diagram of a logical scrambler. The architecture is implemented using register set which are feedback through a combinational circuit [2][3].

The pseudorandom signal generated by the circuit feed the output bus called ***prbs***[(L-1)..0], where L represents the number of output bits.

The simplest example for this type generator shown in Fig. 1, where the output of the random sequence has only one bit.

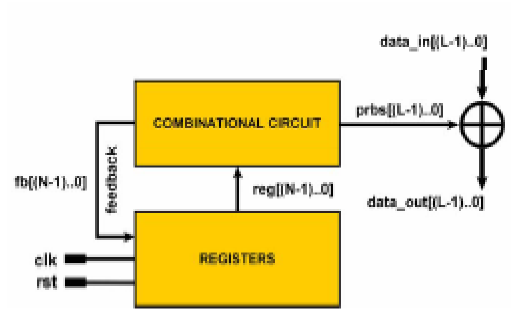

Fig.2. Logical scrambler block diagram [3].

In this serial scrambler, the initialization block forces the reset of all registers that keep a high logic level in their outputs. As per observation, in each clock cycle the register data are shifted to the right and the least significant bit is calculated from the combined output of the register ***reg***(0),***reg***(2), ***reg***(11) and ***reg***(15). This process continues until 2^{16}-1 values are generated, and the sequence is then restarted. If an FPGA operates at 150 MHz clock frequency, the PRBS produces one bit for each clock cycle, hence, the data rate is 150 Mbit/s[3].

An efficient way to double the data rate of the PRBS generator is to modify the PRBS circuit to generate two bits simultaneously instead of just one. In other words, working with the two output bits in the PRBS module, the data rate reaches 300 Mbit/s for the same 150 MHz frequency clock shown in Fig.3.

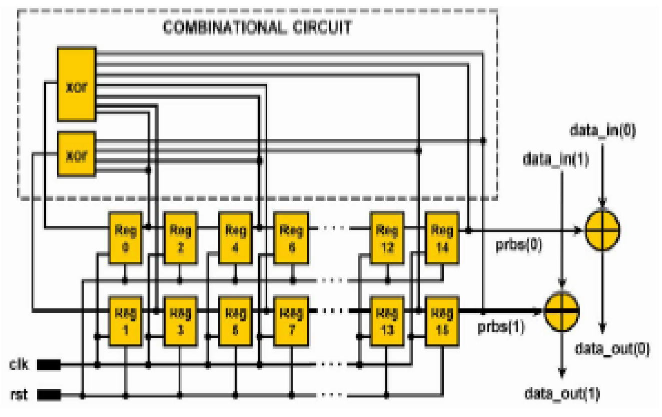

Fig.3.Logical scrambler for 300 Mbit/s [3].

IV. IMPLEMENTATION AND RESULTS

A. Design of 300 Mbit/s logical scrambler:

The implementation of 300Mbit/s logical scrambler is described in this section. This module is implemented using Verilog HDL. Fig.4 Top view of logical scrambler architecture shows the inputs and outputs of the logical scrambler. Since it has clk, rst and 2 data inputs din1, din2 are inputs and 2 data outputs dout1, dout2 are outputs. The feedback loop is given as input to the scrambler. Here two feedback loops are used to perform parallel operation to increase the data transfer rate.

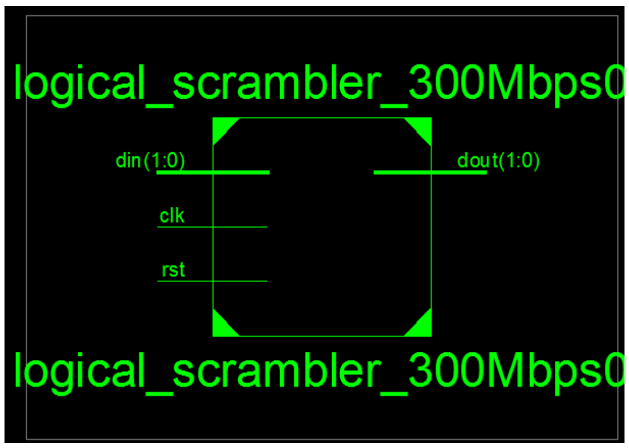

Fig.4 Top-view of 300Mbit/s logical Scrambler

B. Simulation Result:

The logical scrambler architecture was implemented using Verilog hardware description language. These descriptions were then processed by standard Xilinx ISE 12.2 design tool suite, which performed synthesis, placement, routing, and bitstream (FPGA physical programming information) generation. The bitstream generated was dumped onto xc3s50-4pq208VP30 device. Fig.5 shows the simulation results.

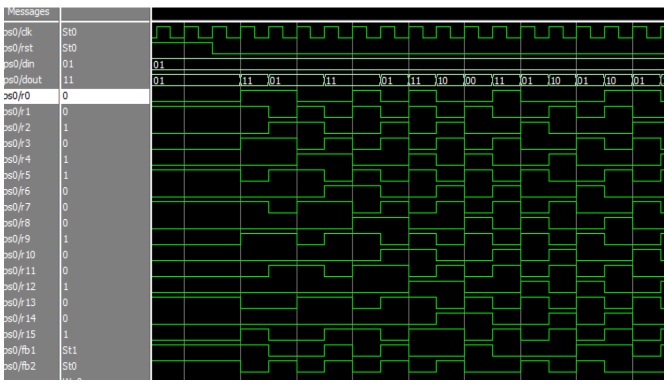

Fig.5 Simulation Result of 300Mbit/s logical Scrambler

The above simulation result shows the logical scrambler architecture for 300Mbit/s application. It uses same register number as in serial scrambler architecture and uses two feedbacks, inputs and output. Hence the speed increases and area reduces by using this architecture. The above result is only for 2bits parallel data in order to achieve high throughput it is necessary to increase the number of bits.

C. Performance and Resource utilization:

The below Table.1 shows a utilization and performance report. In order to compare this architecture with any other scrambler architecture these parameters very use full. The tool called Xilinx XPower Analyzer is used to measure the power consumption of the device according to implemented logic. The power is more important because it effect to system performance. According to following result the through achieved here is for 2 bits data is 1.2Gbit/s.

Resource Utilization	Slices	17
	Register	17
	LUTs	6
Performance	Max Frequency	644MHz
Power supply	Consumption	360mw

Table.1 Utilization and Performance report

V. CONCLUSION AND FUTURE WORK

The architecture of the logical scrambler architecture has been designed. The logical scrambler can achieve higher performance with low FPFA resource occupation shown in table.1.This architecture will be implemented in verilog HDL independently and then integrated as per design. The complete design will be verified using Test benches. Finally power consumption also less in this architecture. In future high data transfer rate is possible by increasing the number of bits in architecture.

REFERENCES

[1]. Win C. H., Chen C. N., Wang Y. J., Hsiau J.Y., Jou s. J., “Parallel Scrambler for High Speed Applications”, Jul. 2006.

[2]. Guilherme Guindani Frederico Ferlini Jeferson Oliveira Ney Calazans Daniel Pigatto Fernando Moraes “A 10 Gbps OTN Framer Implementation Targeting FPGA Devices”, Dec. 2009.

[3]. Arley Salvador, Valentino Corso “100 Gbit/s Scrambler Architectures for OTN Protocol: FPGA Implementation and Result Comparison”, Aug.2012

[4]. ITU-T “G.870: Terms and definitions for optical transport networks (OTN)”. Available at: http://www.itu.int/rec/TREC-G.870-200803-I/en, Apr. 2009.

[5]. Chethan Kumar M, Praveen Kumar Y G, Dr. M. Z. Kurian ,“Design and Implementation of Serial Scrambler Architecture for OTN Protocol”, Apr 2014

[6]. Sawyer N. “SONET and OTN Scramblers/Descramblers”, Application Note XAPP 651, Nov. 2002.

First Author
Chethan Kumar M
PG Student, Sri Siddhartha Institute of Technology, Tumkur, Karnataka India.

Second Author
Praveen Kumar Y.G
Asst. Prof., dept. of E&C, Sri Siddhartha Institute of Technology,Tumkur, Karnataka,India,

Third Author
Dr. M. Z. Kurian
HOD, Dept of ECE, Sri Siddhartha Institute of Technology, Tumkur, Karnataka, India,