# https://eng.asu.edu.eg/download?sid=ZyszcL41aMPUCgmWWlKiyWXL2GTF85a5MsjJsJNZZuk%3D

Ain Shams University,

Faculty of Engineering,

Computer and Systems Engineering

**Digital Design and Verification**

**For**

**SerDes System**

**Team members**

Hassan Khaled Hassan

Abdelmagid Mohamed Abdelmagid

Alaa Salah Abd-Elfattah

**Under Supervision of: Sponsored By:**



Dr. Ahmed M. Zaki

Associate professor,Ain-shams University, Egypt

**(https://ic-pedia.com)**

Contents

[1](#_Toc155890371)

[Table of Content 3](#_Toc155890372)

[ABSTRACT 6](#_Toc155890373)

[Introduction 7](#_Toc155890374)

[Elastic Buffer: 8](#_Toc155890375)

# Table of Content

1. USB-PHY
   1. TX
      1. PCS
         1. Gasket
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         2. Line\_8\_10\_coding
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         3. FSM\_RD
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         4. Encoding
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
      2. PMA
         1. PMA
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
   2. RX
      1. PCS
         1. Elastic Buffer
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         2. Comma Detection
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         3. Decoder
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         4. RX\_Status
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         5. Gasket
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
      2. PMA
         1. CDR
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]
         2. Serial To Parallel
            1. Specs.
            2. RTL
            3. Schematic
            4. Report Area, Power
            5. Test Plan
            6. Gasket UVM Environment
            7. Snippets Wave Form
            8. Coverage Reports [Functional & Code]

# ABSTRACT

This document is the graduation project report prepared by senior year’s students in Computer and System department of Faculty of Engineering Ain Shams to discuss the USB (universal serial bus) Superspeed also known as USB 3.0, it introduces significantly higher data transfer rates and enhanced capabilities compared to its predecessors. The USB SuperSpeed standard employs advanced serializer and deserializer (SerDes) technologies to enable faster communication between USB devices and hosts.

Serializer and deserializer components play a crucial role in USB SuperSpeed by converting parallel data into serial data during transmission and vice versa during reception. This enables efficient utilization of the available bandwidth and facilitates high-speed data transfer rates.

This document also features Functional verification for USB chip to chip layer using SV(system verilog) and UVM (Universal Verification Methodology) with its phases, hierarchy and components to verify the modules and the functionalities implemented.

# Introduction

A Serializer/Deserializer (SerDes) is a pair of functional blocks commonly used in high-speed communications to compensate for limited input/output. These blocks convert data between serial data and parallel interfaces in each direction. The term "SerDes" generically refers to interfaces used in various technologies and applications. The primary use of SerDes is to provide data transmission over a single line or a differential pair to minimize the number of I/O pins and interconnects. The basic SerDes function is made up of two functional blocks: the Parallel In Serial Out (PISO) block (aka Parallel-to-Serial converter) and the Serial In Parallel Out (SIPO) block (aka Serial-to-Parallel converter). There are 4 different SerDes architectures: (1) Parallel clock SerDes, (2) Embedded clock SerDes, (3) 8b/10b SerDes, (4) Bit interleaved SerDes.

### Elastic Buffer:

**Introduction**

The elastic buffer is a crucial component used to avoid the timing errors between the read and write operations. The elastic buffer is mainly an asyncrounous fifo with additional functionality where it is used to deal with SKP to avoid losing data. Normally the System clock and receive clock, which is used in the elastic buffer, should be the same which is 5G

However there there can be slight changes which ranges from -5300 ppm to 300ppm.

The elastic buffer implemented uses Nominal Half Full Buffer where the elastic buffer should always have the buffer half full and adding the SKP or removing it is done to keep the elastic buffer half full.

**Elastic Buffer Design:**

Figure 1 elastic buffer

Memory Unit

Write pointer control unit

Read pointer control unit

Synchronous Unit

Threshold monitor

The elastic buffer is mainly designed to have 5 main blocks:

* Memory unit

It is mainly a fifo where data is stored and read from it. The elements stored are the 10 bits data received which can be data or command like SKP. It determines the position to read from or write to using the read and write pointer which is received from other blocks.

* Write pointer control unit

This module produces the binary and gray write pointer, which is used for the selection of the correct address of the memory unit while taking into consideration if a SKP needs to be deleted. It generates the overflow signal by comparing between the read and write address. . The gray code of read pointer is synchronized to the Recovered Clock Domain and compared with the gray code of the write pointer , then the unit produces the full Flag Read pointer control unit

* Read pointer control unit

This module produces the binary and gray read pointer, which is needed to read from the correct address in the memory unit and it generates the request signal of SKP add after being compared with the write pointer. The gray code of write pointer is synchronized to the local Clock Domain and compared with the gray code of the read pointer , then the unit produces the empty Flag.

* Threshold unit

This module is used to check wether the buffer exceeded the limit in comparision with the size of half the array. So if the number of elements bigger than 8 skp remove request is made

* Synchronous Unit

The function of this module is to synchronize the gray code of the write pointer and the read pointer so the read and write ointer checks on them and produce the empty and full flag

FF

FF

read pointer control unit

FF

FF

Write pointer control unit

Gray code write pointer

Gray code read pointer

Gray code is used as the change between a number and the following one is in only one bit so by using Gray code and synchronizers errors is avoided even when dealing with two clock domains, Recovered Clock Domain and Local Clock Domain.

**Flow charts for SKP Delete and SKP add Operations**

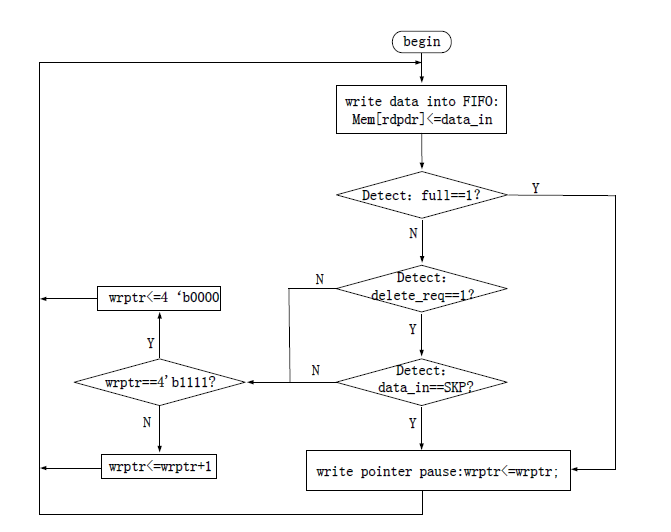


Figure flow chart for SKP deleting operation

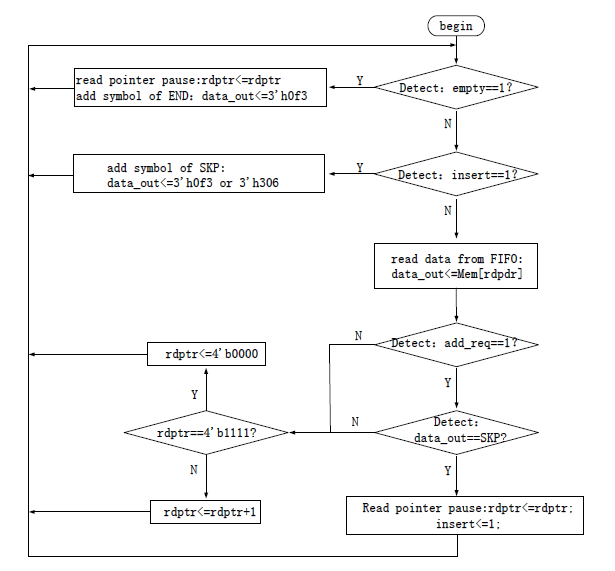


Figure flow chart for SKP adding operation