

# USB 3.1 PHY Implementation

Amr Elhosiny

October 16, 2014

## 1 HDL Language

VHDL will be used for RTL coding. Verilog and/or VHDL will be used for test bench coding.

## 2 Package

A VHDL package will be created listing:

- All hard-coded packet types in the document will be defined with names to be easily compared and detected in the RTL.
- It will define needed functions.
- Packages and functions specific for test benches will be in separate packages.

## 3 Clock Generator

There will be a central clock generator module which generates divided clocks, gated clocks. It may instantiate FPGA specific modules like BUFGE's.

## 4 Packet Generation & Detection

There are many design alternatives listed below.

### 4.1 Packet Type Detector

Depending on packet type, some blocks or functions are enabled or disabled. There will be a central packet type detector module with an active high output bit for each packet type. For packets that have common effect on loaded modules or functions, more possible outputs may be grouped into less or single output bit. The type of packet will decide the behaviour of the datapath modules.

### 4.2 State Machine

Packet driver may be a state machine implementing some features of the link layer. It will transmit/receive certain sequences and implement limited link layer features. It will then produce required control signals defining the type of transmitted/received packet to control the behaviour of the datapath modules. Examples for the link layer features that may be implemented :

- A state machine will handle link initialization.
- Then sending data packet.

A set of assumptions will be made to simplify the state machines and the implementation for example implementing specific not all modes of operation.

### **4.3 Downstream & Upstream Ports**

The host machine will implement extra features not present in the USB device.

## **5 Scrambler**

Will have a toplevel consists of a scrambler core alongside with a controller. The controller will tell the core to enable or disable scrambling, advance the LFSR, XOR or do not XOR data.

### **5.1 Gen-1**

### **5.2 Gen-2**