18-224 Proposal First Draft

Rudy Sorensen

April 24, 2025

1 Project Title: USB Hub

$$\lim_{x \to \infty} \frac{1}{x} = 0$$

This project will implement a USB Hub with 1 Type-A Downstream Facing Port (DFP), 2 Type-C DFPs, and a Type-C Upstream Facing Port (UFP). It will only be USB 1.1 compliant due to hardware limitations (maximum clock speed of 66 MHz and required clock speed of 480 MHz). For a short description of its functionality, it will respond to SETUP packets, report updates on connections to DFPs, perform the necessary measures for a low-speed connection, handle the sending of packets upstream, and otherwise act transparent while packets are sent downstream.

2 Technical Details:

- As according to spec, operates in full-speed mode (12 Mbps) and supports devices that operate in low-speed mode (1.5 Mbps)
- Will operate at 24 MHz to properly sample the asynchronous signal coming from the Host into the Hub's clock domain
- Architecture shown below, sourced from section 11.1.1 of the USB 1.1 Specification. The Hub Controller mostly generates control signals for the Hub Repeater to act upon

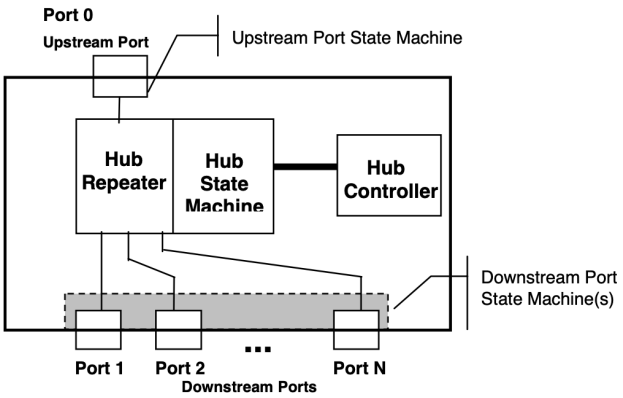

Figure 11-1. Hub Architecture

2.1 Hub Controller

"...provides the mechanism for host-to-hub communication. Hub-specific status and control commands permit the host to configure a hub and to monitor and control its individual downstream ports."

- Responds to USB setup packets for the initialization procedure, communicates device information
- Detects Host communications and enables Hub connections to full-speed Devices
- Detects PRE packets from Host in order to enable propagation to low-speed devices
- Detects reset requests from the Host and enables the reset of that specific device

2.2 Hub Repeater

"...responsible for connectivity setup and tear-down. It also supports exception handling, such as bus fault detection and recovery and connect/disconnect detect."

- Acts as a link from the Host to Devices, connecting the UFP to the DFPs when necessary
- Inverts next packet from Host and drives low-speed lines after a PRE packet is detected

3 I/Os:

- 8 bidirectional pins for DPs and DMs, these are standard pins in the USB protocol for serial communication
- 1 output pin for VCC
- 1 output pin for GND

4 Hardware Peripherals:

I intend to create a PCB that has the connectors for the Devices and Host to connect to. This PCB would also "trick" the Type-C ports into permanently being in either Host or Device mode by tying their Configuration Control (CC) pins to VCC or GND. This is necessary since they typically decide their roles through a complicated protocol over the CC pins introduced in USB 3.0 that I cannot perform.

5 Module Header:

```
module top (
  input  logic clk,
  input  logic rst_n,

  // USB Devices
  inout logic dp0,
  inout logic dm0,
  inout logic dp1,
  inout logic dm1,
  inout logic dp2,
  inout logic dm2,

  // USB Host
  inout logic dp_H,
  inout logic dm_H,
);
```