USB 1.1 Receiver Basics

Preparation for final design project

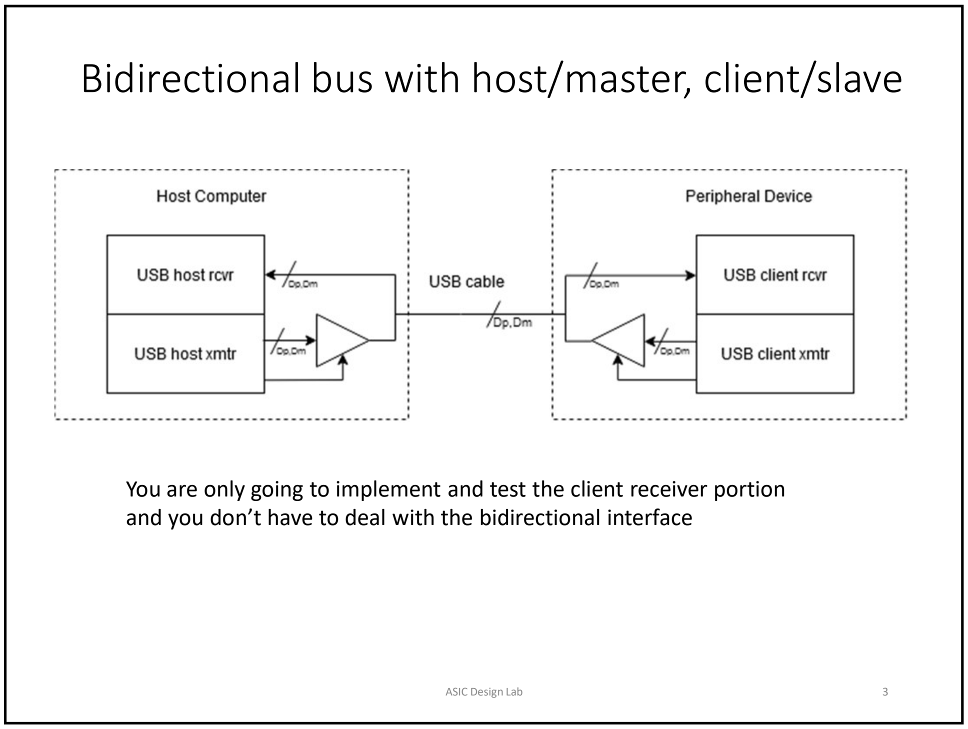
Details will be in lab manual and lab discussion notes

1

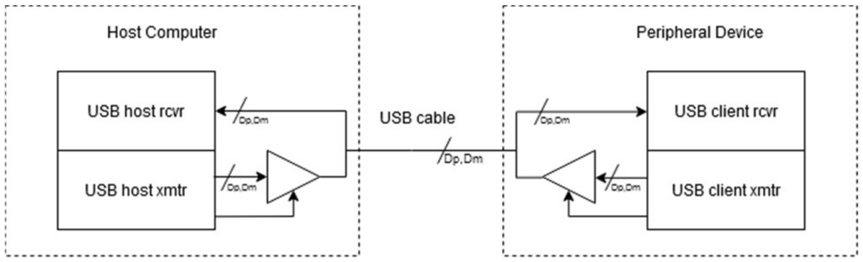
Key concepts

- Bidirectional busses with host/master, client/slave
- Complementary/balanced signal transmission
- Framing of USB 1.1 packets (review from HW 1)
  - Sync byte
  - EOP
- Bit timing recovery (review from HW 1)
- Bit stuffing (you won't be required to support)
- NRZ encoding
- Minimal USB 1.1 receiver architecture
- Coming later
  - CRC: Cyclical Redundancy Checksum
  - Packet types, PID (Packet ID)

2



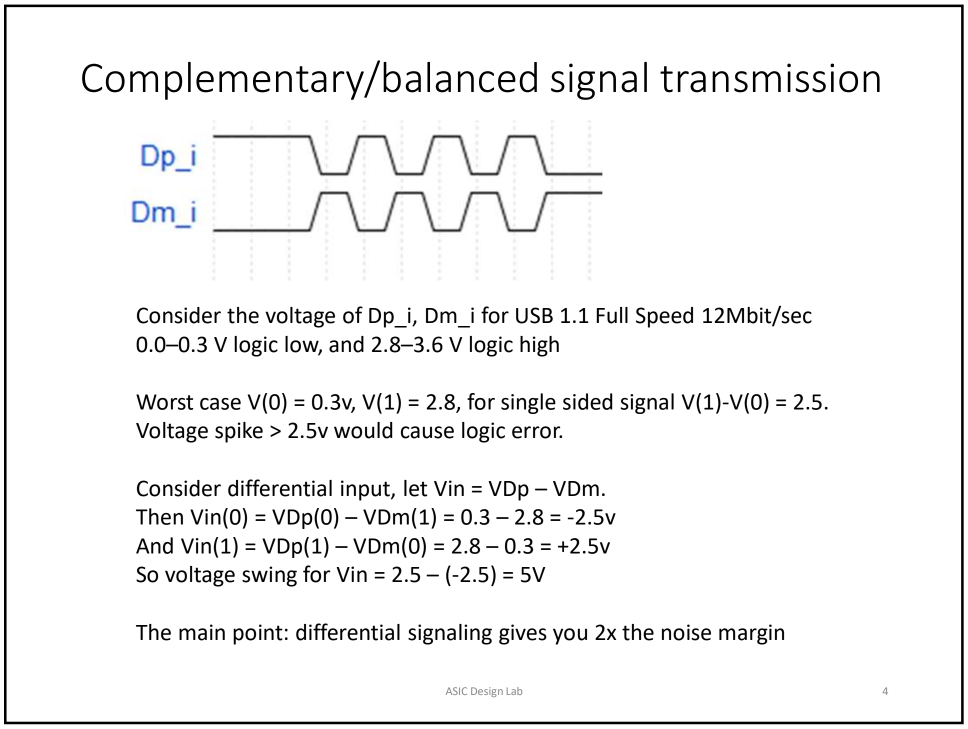
Bidirectional bus with host/master, client/slave



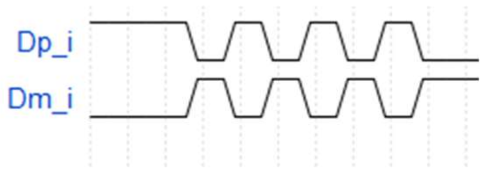
You are only going to implement and test the client receiver portion and you don't have to deal with the bidirectional interface

ASIC Design Lab 3

3



Complementary/balanced signal transmission



Consider the voltage of Dp_i, Dm_i for USB 1.1 Full Speed 12Mbit/sec
0.0–0.3 V logic low, and 2.8–3.6 V logic high

Worst case V(0) = 0.3v, V(1) = 2.8, for single sided signal V(1)-V(0) = 2.5.
Voltage spike > 2.5v would cause logic error.

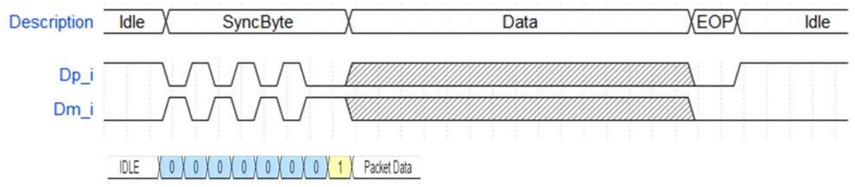
Consider differential input, let Vin = VDp – VDm.
Then Vin(0) = VDp(0) – VDm(1) = 0.3 – 2.8 = -2.5v
And Vin(1) = VDp(1) – VDm(0) = 2.8 – 0.3 = +2.5v
So voltage swing for Vin = 2.5 – (-2.5) = 5V

The main point: differential signaling gives you 2x the noise margin

ASIC Design Lab 4

4

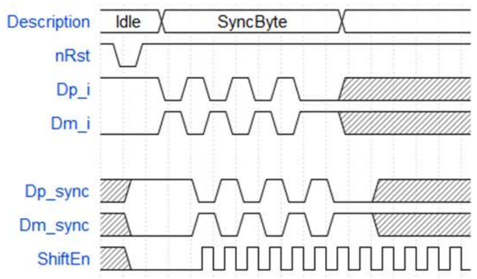
Framing of USB 1.1 packets



- In between packets bus is idle
- Beginning of packet must match sync byte pattern
  - Any thing else is invalid, ignore the packet
- At end of packet (EOP), Dp_i = Dm_i = 0.
- In between is data.
- Learn later – data includes metadata, checksums, and actual payload

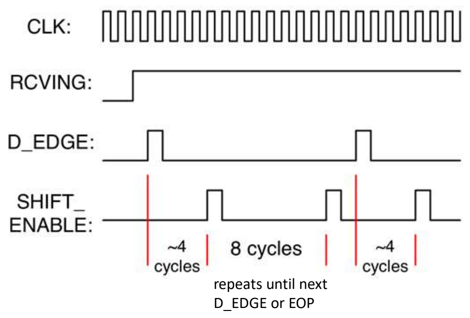
Timing Recovery

Here is how it was presented for HW 1

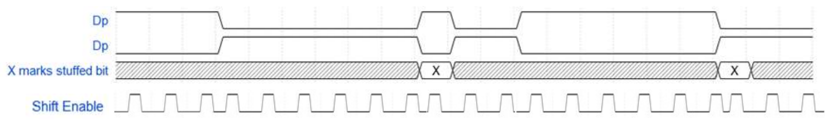


Edge detector circuit makes a pulse after every transition on Dp (or Dm).

D_EDGE triggers a Counter to produce a pulse after 4 CLK cycles then repeat every 8 cycles



Bit Stuffing



USB relies on transitions to restart (resynchronize) the shift enables

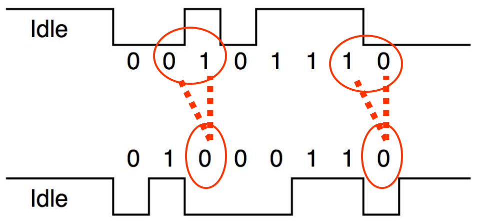
This example: incoming data slightly faster than receiver data rate

If too many bits in a row without a transition, shift enable timing drifts too far from incoming bit rate.

Notice: at each input transition, shift enable pulses resynchronize to incoming bits

NRZ Encoding/Decoding 0's

NRZ-Encoded data:



Recovered Data:

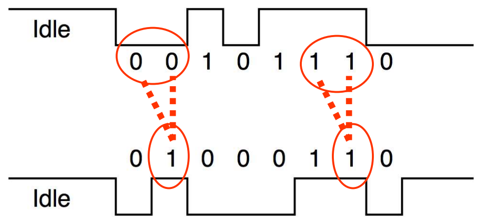
Original Data		Encoded data
0	**=>**	**0->1 or 1->0 transition**
1	=>	no transition

To decode: have to compare current and previous bit.
If same, data = 1.If different, data = 0.

NRZ Encoding/Decoding 1's

NRZ-Encoded Data:

Recovered Data:



Original Data		Encoded data
0	=>	0->1 or 1->0 transition
1	**=>**	**no transition**

To decode: have to compare current and previous bit.
If same, data = 1.If different, data = 0.

9

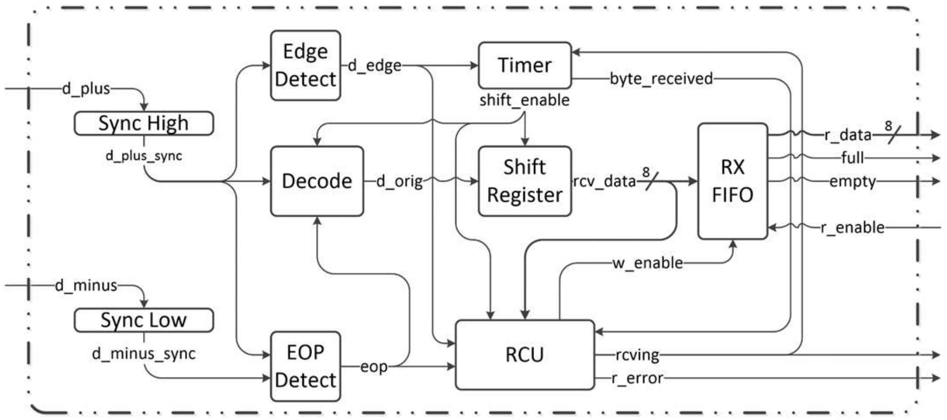
How to decode

- Reverses NRZ encoding
- Done inline with data stream to shift register
- Synchronously compare current value and immediate past data bit

- Add shift enable to a synchronous edge detector

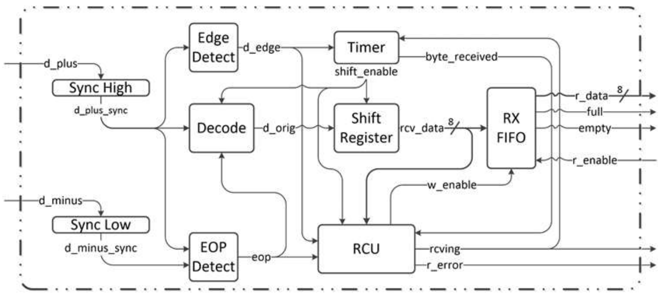
10

Minimal Architecture



11

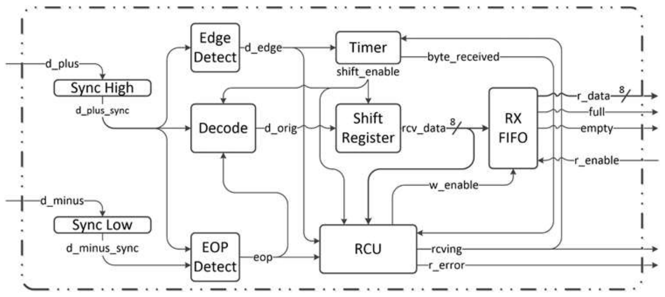
RCU



- Controls USB Receiver circuit
- Begins receiving when an edge is detected.
- Checks to see if the SYNC byte was received
  - Error condition if not
- Activate the storing of each data byte into the FIFO
  - Told by timer when a new byte is ready
- Stop receiving when the EOP is detected.

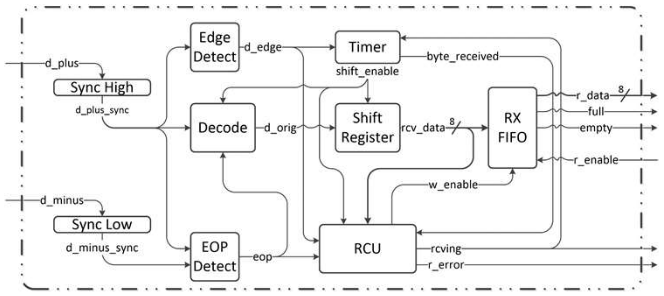
12

Sequence of Operation



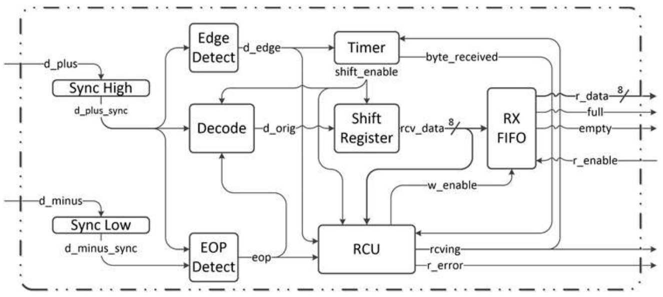
1. On reset, the RCU goes to the idle state
2. When an edge is first detected
   - Begin receiving data through the DECODE block to the SHIFT_REG.
   - Set the RCVING line high.

13



3. After the first byte is shifted in
   - Check that byte matches USB SYNC byte
   - If byte matches SYNC, do not store to FIFO, but begin receiving and storing data from next byte
   - If the byte does *not* match the SYNC byte
     - Set R_ERROR flag to 1
     - Disregard any input until the next EOP is reached
     - RCVING line should remain high until the EOP is reached
     - R_ERROR flag should remain high until the next packet begins

14



4. Continue receiving and storing data to FIFO until the EOP is detected
   - Then set the RCVING line low.
   - If an EOP is reached with an incomplete byte in the shift register
     - Set the ERROR flag high and do not store the last byte.
     - Leave the R_ERROR flag high until the next packet begins.
   - Note: Do not worry about FIFO overflow as it will not be graded

ASIC Design Lab 15

15