

Hands on with FPGA's: Module 7

Venkat Rangan

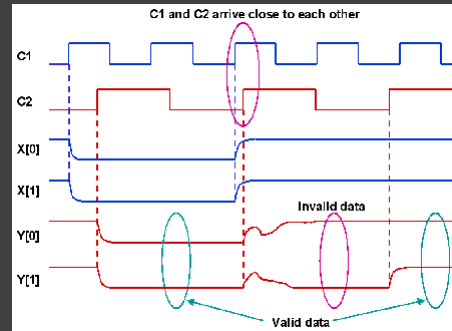
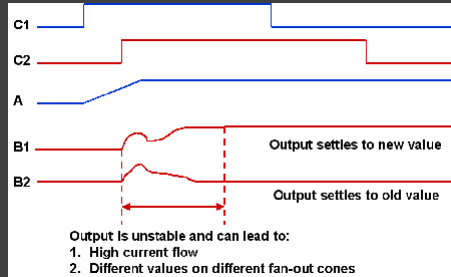
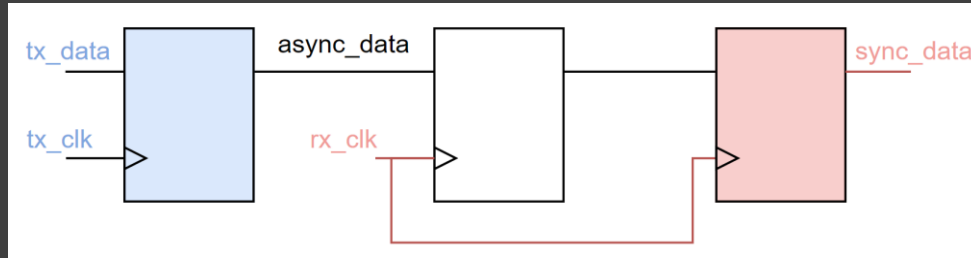
Questions on Module 6

Topics

- Pre-class: Open floor for questions
 - Module 6
- Clock Domain Crossing
 - Why?
 - How to deal with it?
- IP Reuse
 - FuseSoC
 - LiteX
- IDE's: tips/tricks
- Open discussion

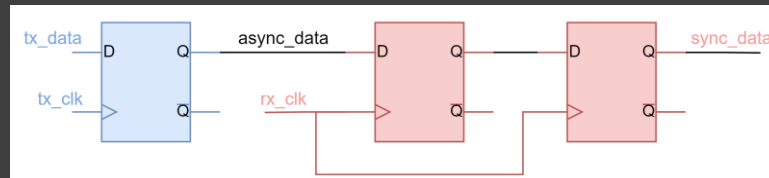
Clock Domain Crossing

- Metastability



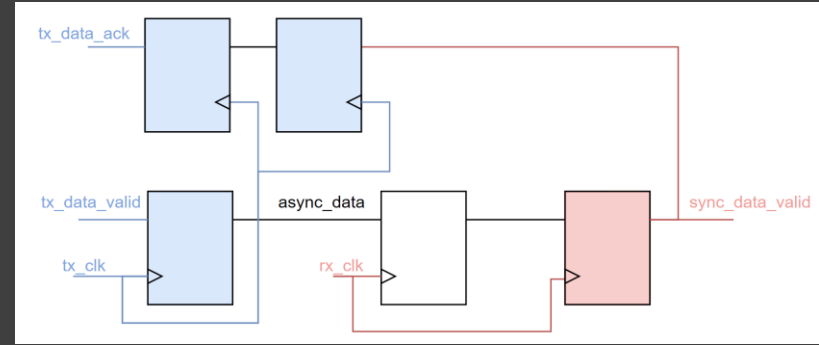
Techniques

- Double flop receiving signal
- Register sending signal
- What if tx_clk is much faster than rx_clk?
 - Assume a 1 clock wide pulse on tx_data



Safe Solution for CDC

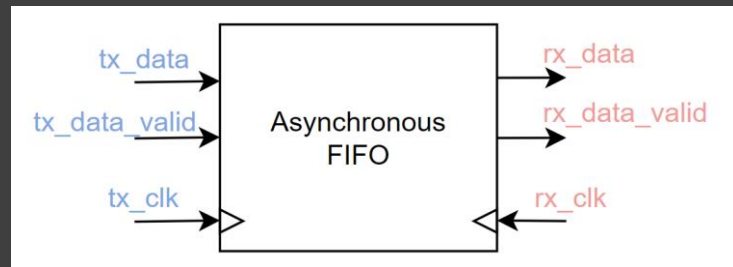
- Use Acknowledge



- Pros:
 - Reliable transfer across clock domain
- Cons:
 - Significantly larger synchronization delay

Other CDC techniques...

- Asynchronous FIFO's for multiple bits
 - Can be just 2 deep FIFO
- Gray encoding
 - Only 1 transition between successive counts



Decimal	Binary	Gray	Decimal of Gray
0	0000	0000	0
1	0001	0001	1
2	0010	0011	3
3	0011	0010	2
4	0100	0110	6
5	0101	0111	7
6	0110	0101	5
7	0111	0100	4
8	1000	1100	12
9	1001	1101	13
10	1010	1111	15
11	1011	1110	14
12	1100	1010	10
13	1101	1011	11
14	1110	1001	9
15	1111	1000	8

https://en.wikipedia.org/wiki/Gray_code

Lets build an SoC!

- Use [LiteX](#)
- Follow instructions in the repo to install
 - For windows, use WSL!
 - Will not work well in the Powershell/Command prompt

IDE's: Tips/tricks

- Emacs: Verilog mode:
 - Demo of auto completion

Module 7:

- Challenge:
 - Create your own version of a processor using LiteX or FuseSoC
 - Locate the I2C and SPI cores in it
 - Can you program your I2C controller to produce some outputs?
 - Hook up any I2C device to the FPGA and try to read/write registers

Open Discussion