

CaribouLite FPGA / SMI

- Overview
- Architecture
- Firmware ↔ Software
- Firmware top level
- SMI I/F

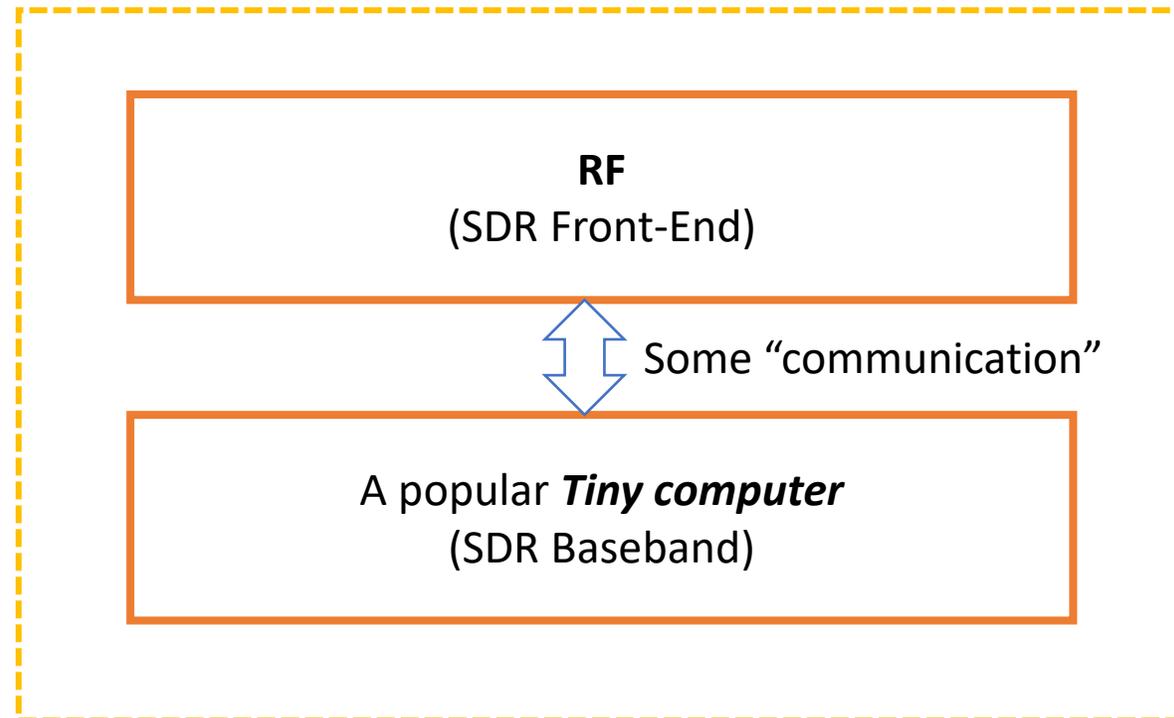


Portable SDR?

We wanted to make a portable RF board that connects to an existing compute unit

>70% of the RF board (the front-end) should be RF components!

Small and self-contained SDR

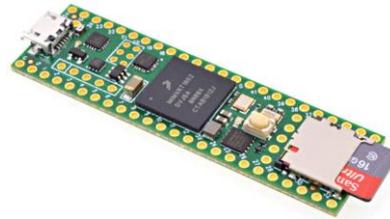


Portable SDR?

The platform problem

- We want to focus on RF, not Compute
- So, which is the best platform?

TEENSY 4.1



+++ Cheap
++ Popular
+ NXP-FlexIO

--- Performance
--- No Linux

Raspberry Pi



++ Performance
+++ Popular (community)
+++ Cheap - Affordable

~~- I/Q Interface?~~



SMI

Jetson Nano



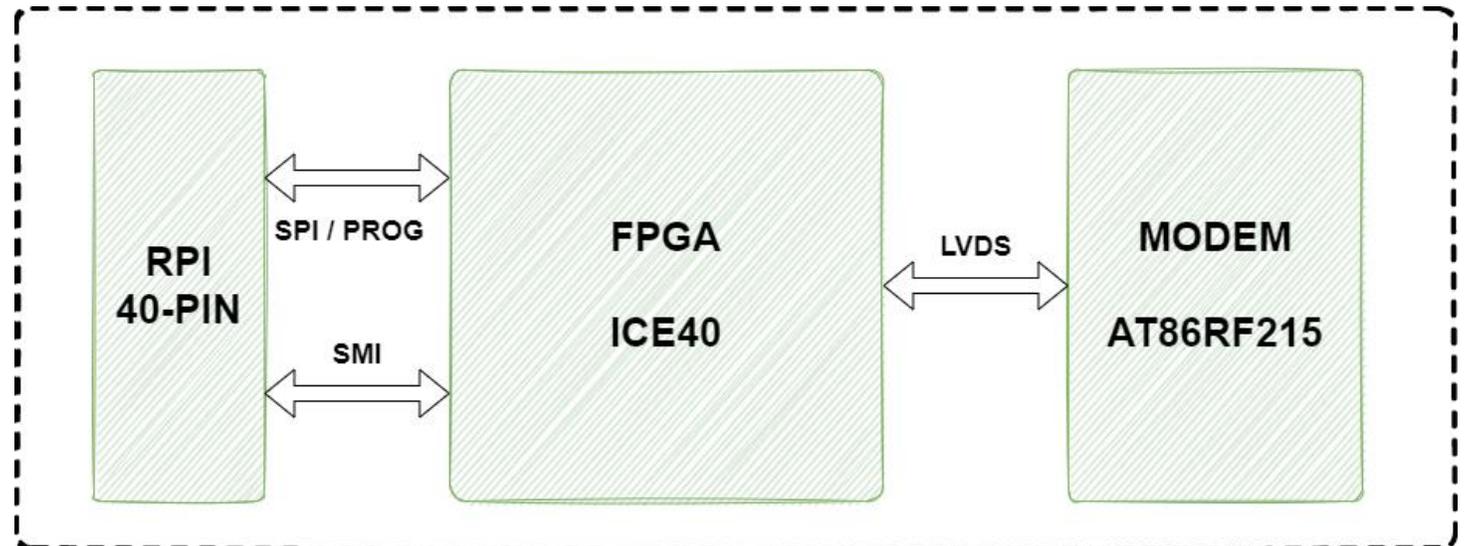
+++ Performance
++ Popular
+ AI

- Expensive
-- I/Q Interface?



Top Structure

- FPGA mediates between LVDS and SMI
- SPI – controlling FPGA functionality
- FPGA programming – done by RPI (Lattice SPI)

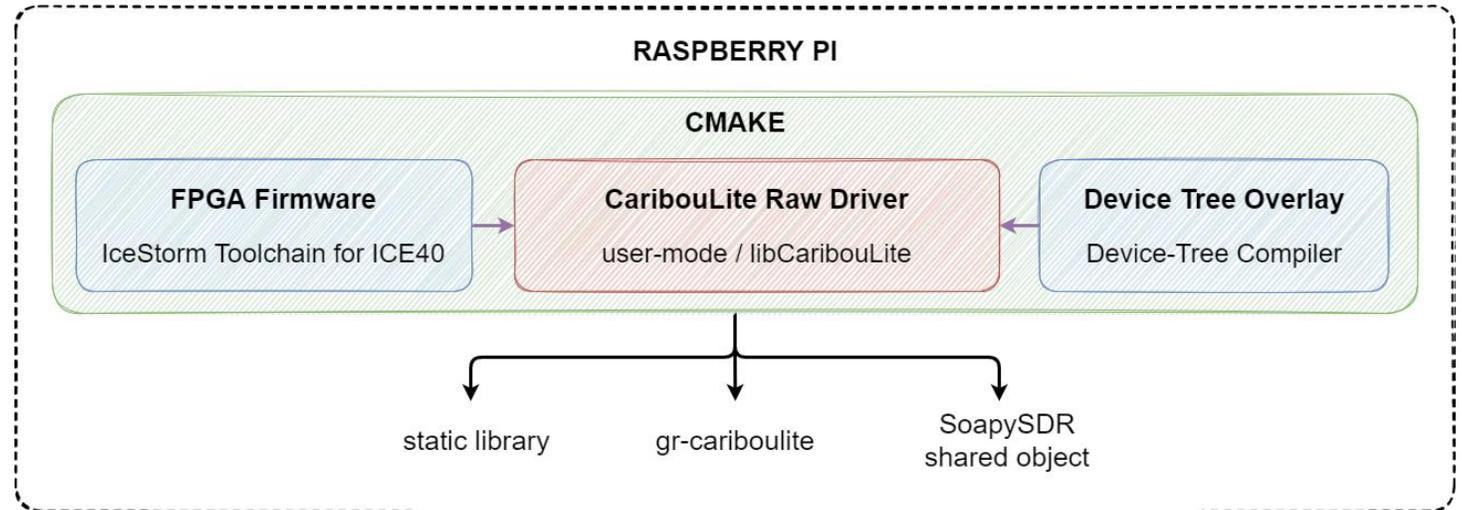


Firmware

↔

Software

Integration

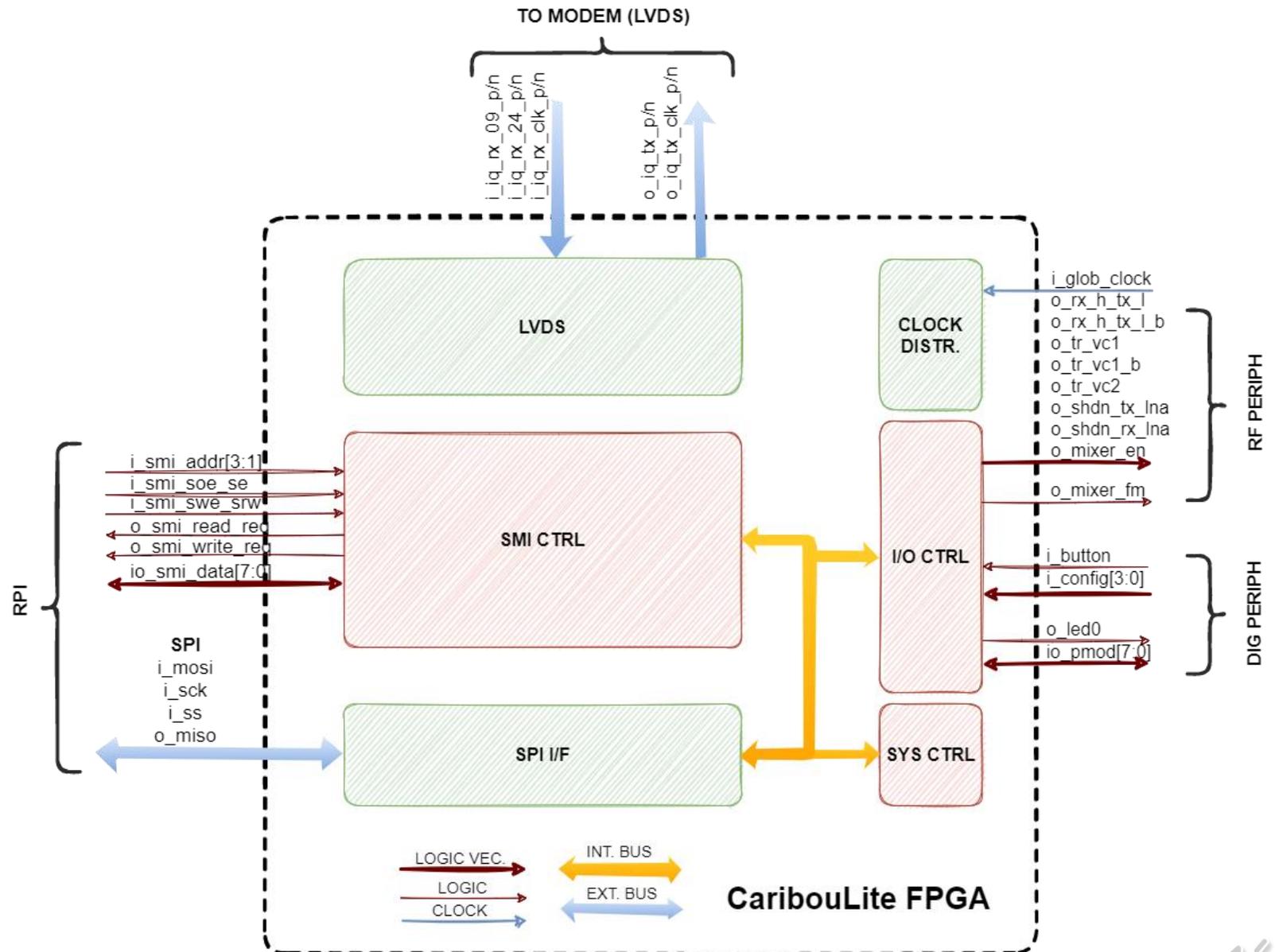


- **Verilog** – Yosys + NextPNR → code blob in C/C++
 - **Device tree** – DTC → code blob in C/C++
 - **One CMAKE project** → SoapySDR, GnuRadio, static-lib
-
- **All inside Raspberry Pi**



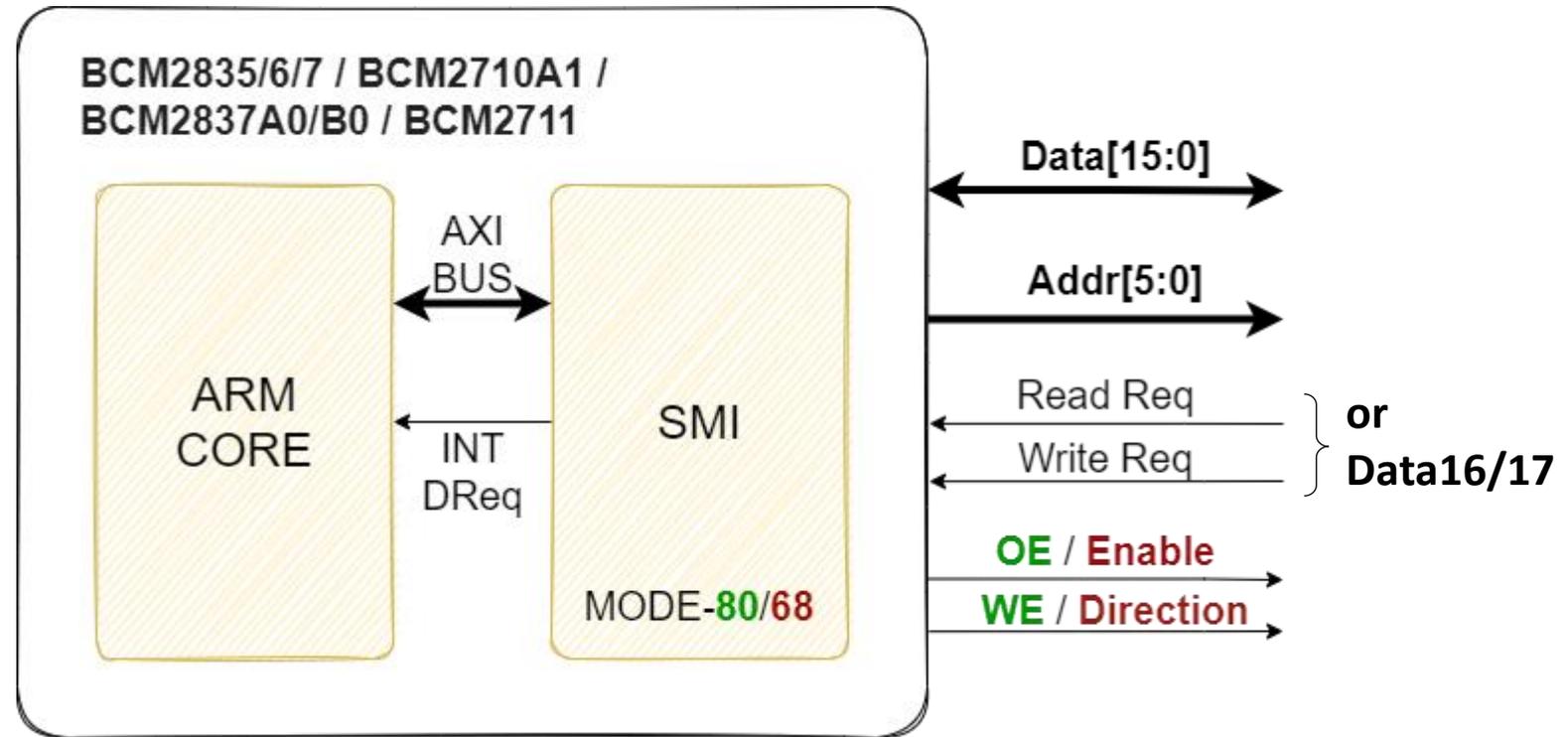
Firmware Top-Level

- Three controllers with bus
 - Sys-Ctrl
 - I/O-Ctrl
 - SMI-Ctrl
- SPI I/F controlling all
- Digital clock: 125 MHz



SMI Signals

- **Two operation modes**
 - Mode-80
 - Mode-68
- **CaribouLite uses:**
 - 8-bit data (out of 16)
 - 3-bit address (out of 6)
 - Read-Req. / Write-Req.
 - Mode-80 signaling



SMI Performance

- **Standard configuration:**

SOE strobe: $T_{soe,stroke} = 2\text{ cc}$

Bus Freq.: $F_{bus} = 125\text{ MHz}$

Data Len.: $L_{data} = 8\text{ bit}$

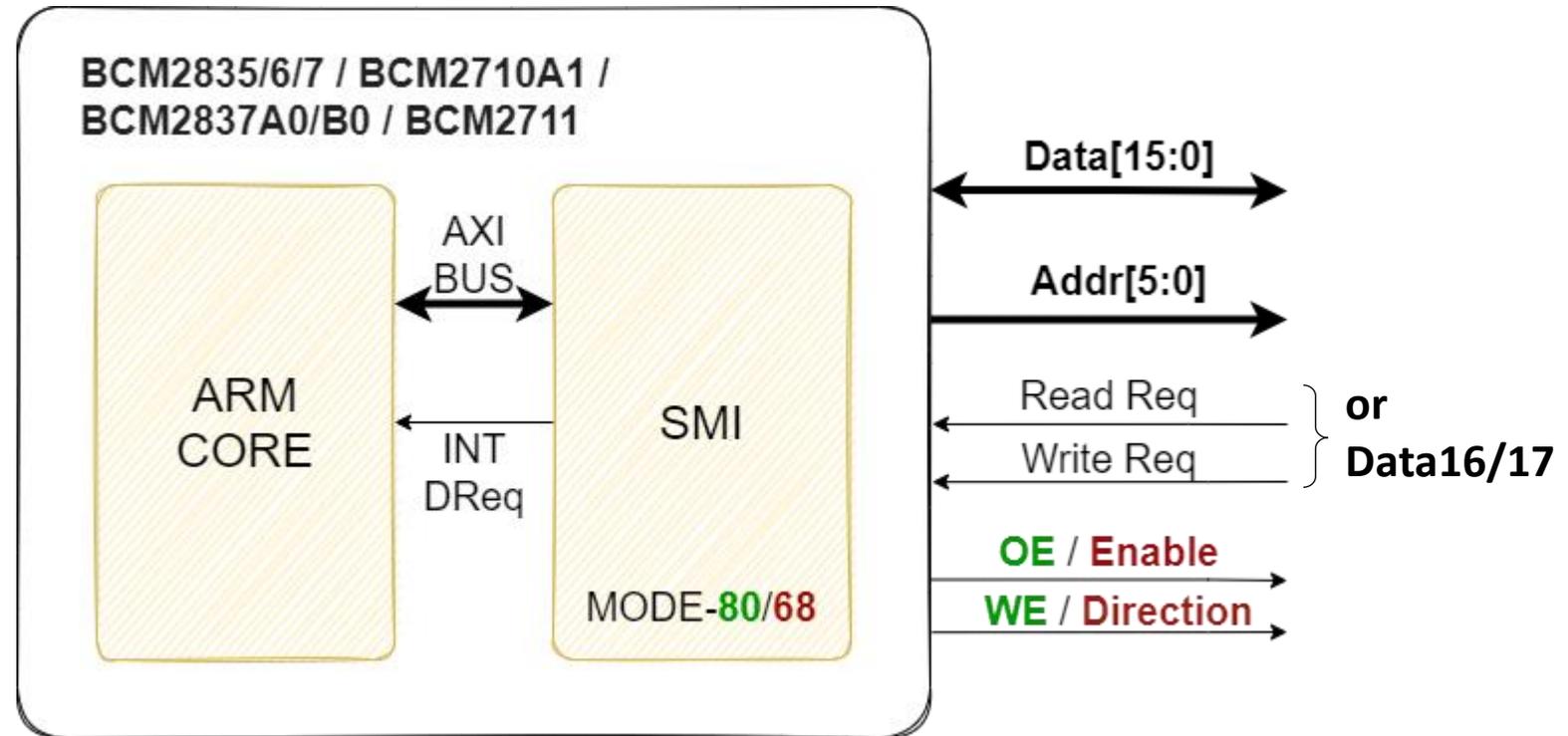
Potential:

$$S = F_{bus} / (T_{soe,stroke} + 1) \cdot L_{data}$$

$$\Rightarrow S = \frac{125}{3} * 8 = 333\text{ Mbit/sec}$$

- **BUT:**

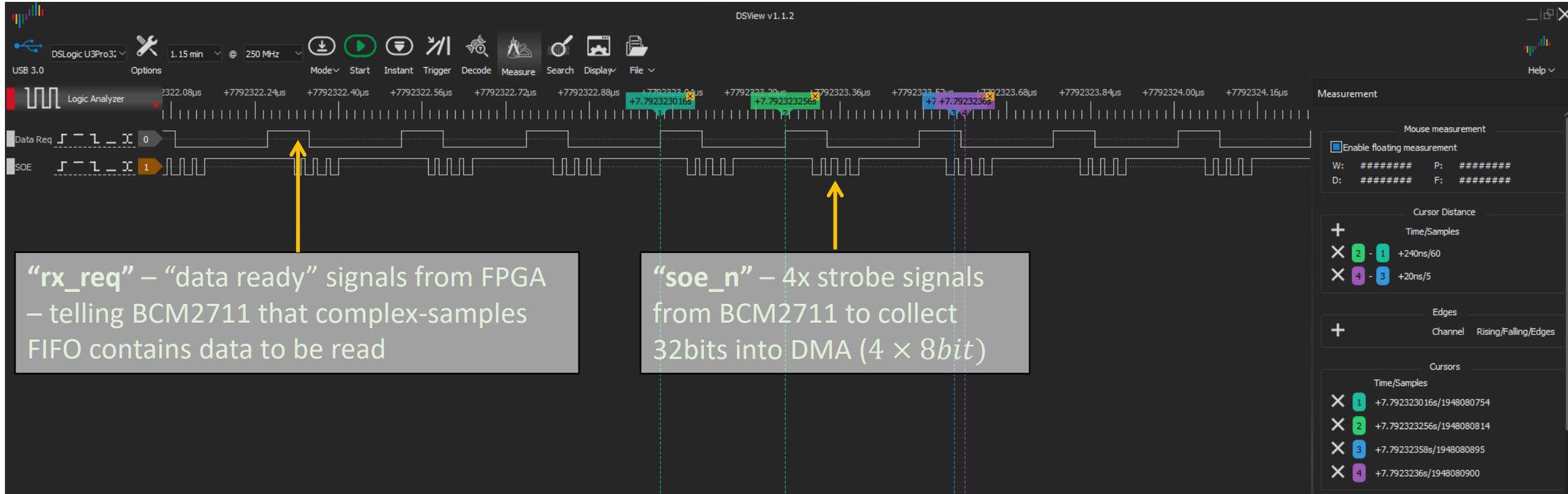
1. Read and write share Data[7:0]
2. Channel config takes time (address)
3. We use only 8-bit data!
4. Next => 16-bit data



Thank you,

- van Loo et.al., 2017, Secondary Memory Interface (SMI)
- Lean2, 2020, <https://iosoft.blog/2020/07/16/raspberry-pi-smi/>

SMI Performance



- Actual speed can be software-scalable to control EMI
- Faster?
 - FPGA (DPRAM) reaction time dependent
 - Next evolution – 16 bit → 666 Mbit/sec



Data-Path

- Dual-port RAM
 - Complex-Numbers queue.
 - Back pressure
- Two clock regions –
 - LVDS from modem
 - Digital from RPI side

