

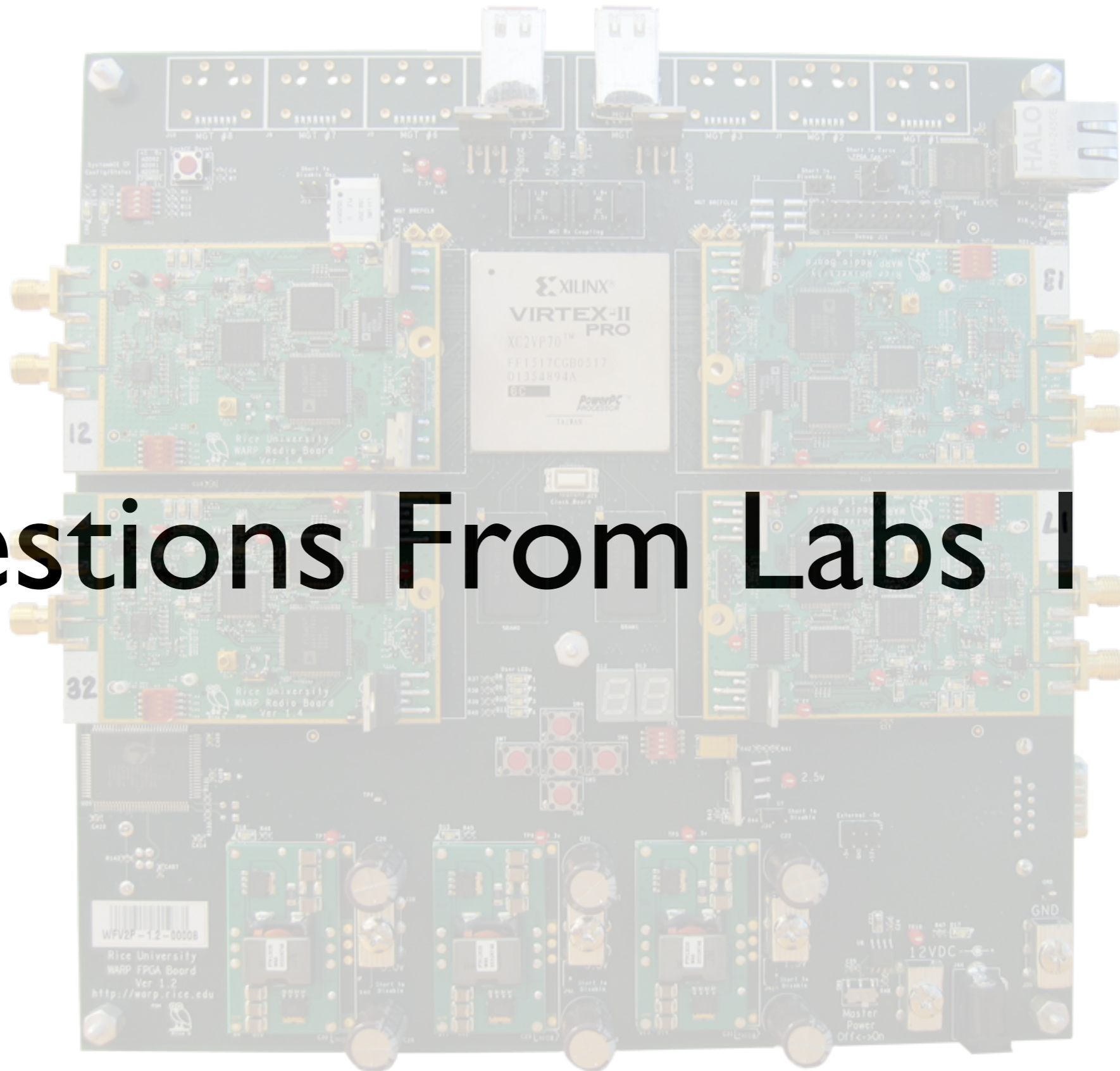
WARP: Physical Layer Design

Patrick Murphy & Siddharth Gupta

WARP Workshop
Rice University
July 14, 2008



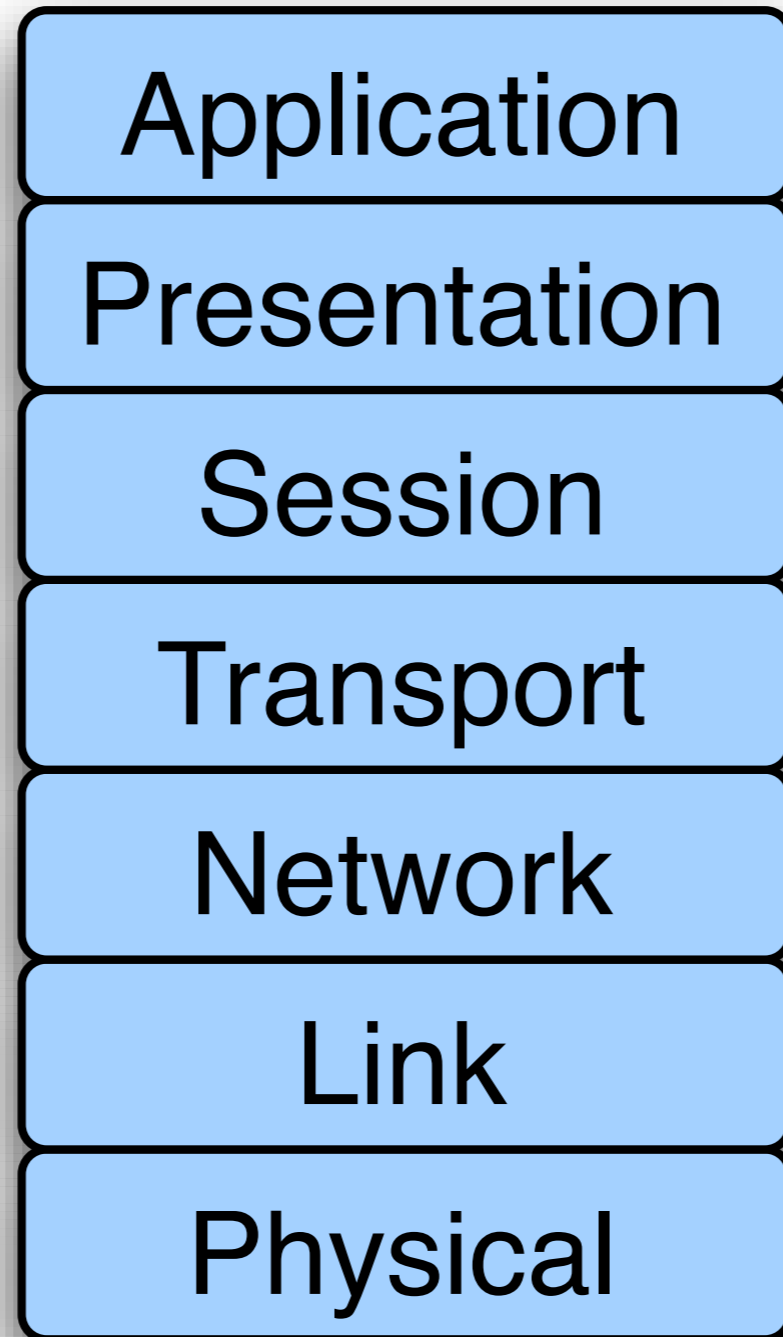
Questions From Labs 1 & 2?



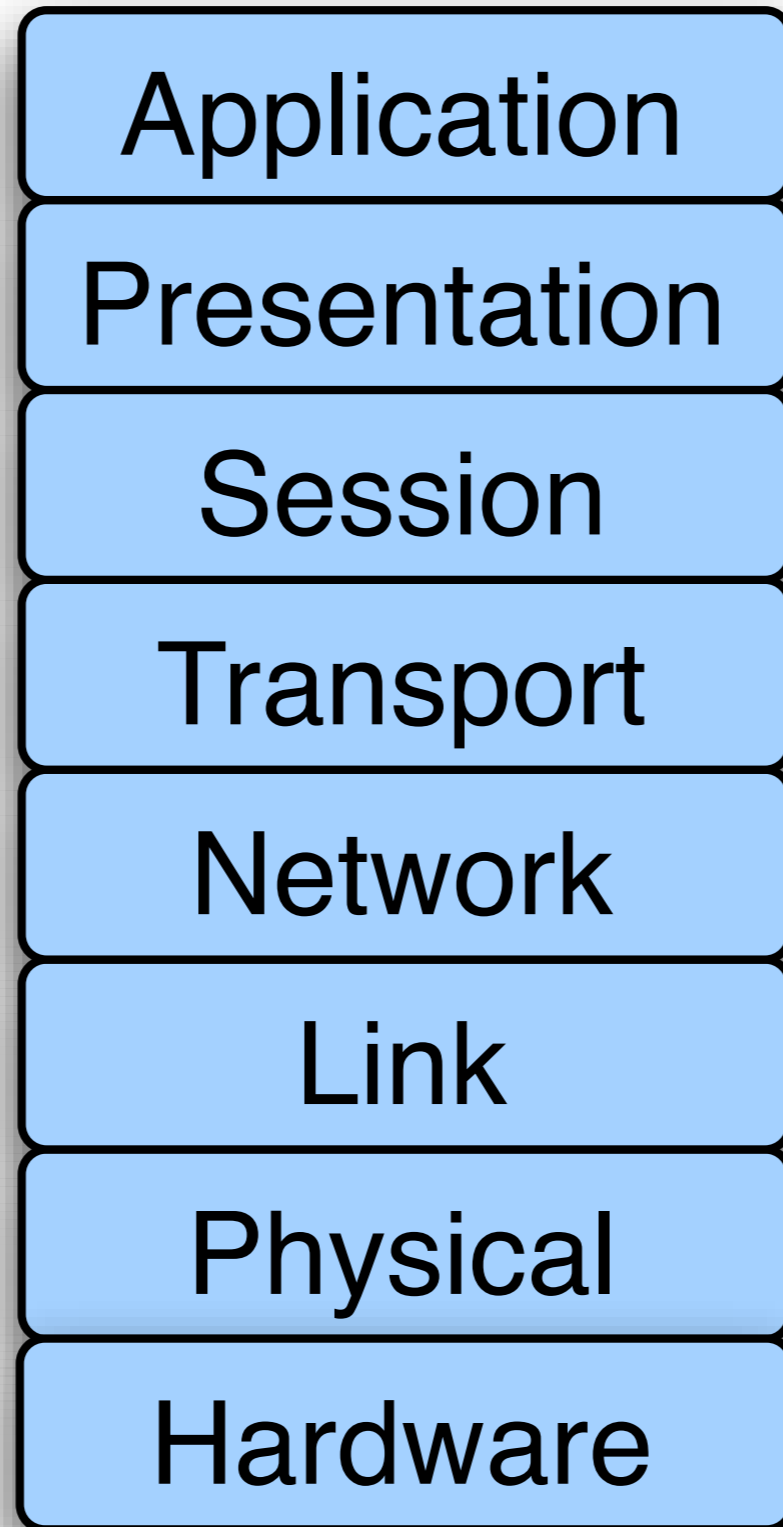
PHY Design - Outline

- Physical Layer Basics
- Real-time Physical Layer Design Flow
- Lab 3: Building a Simple Transmitter

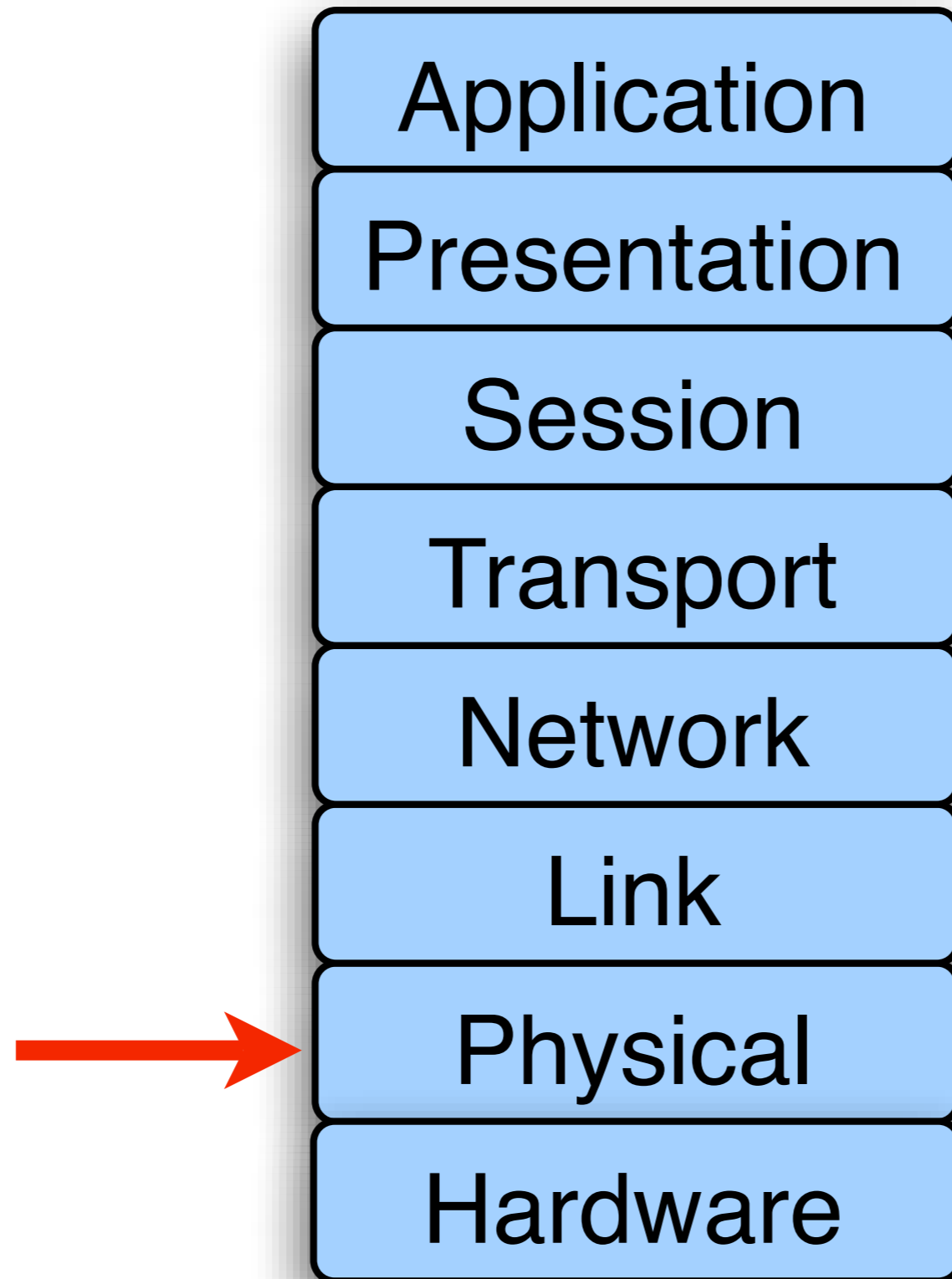
Physical Layer Basics



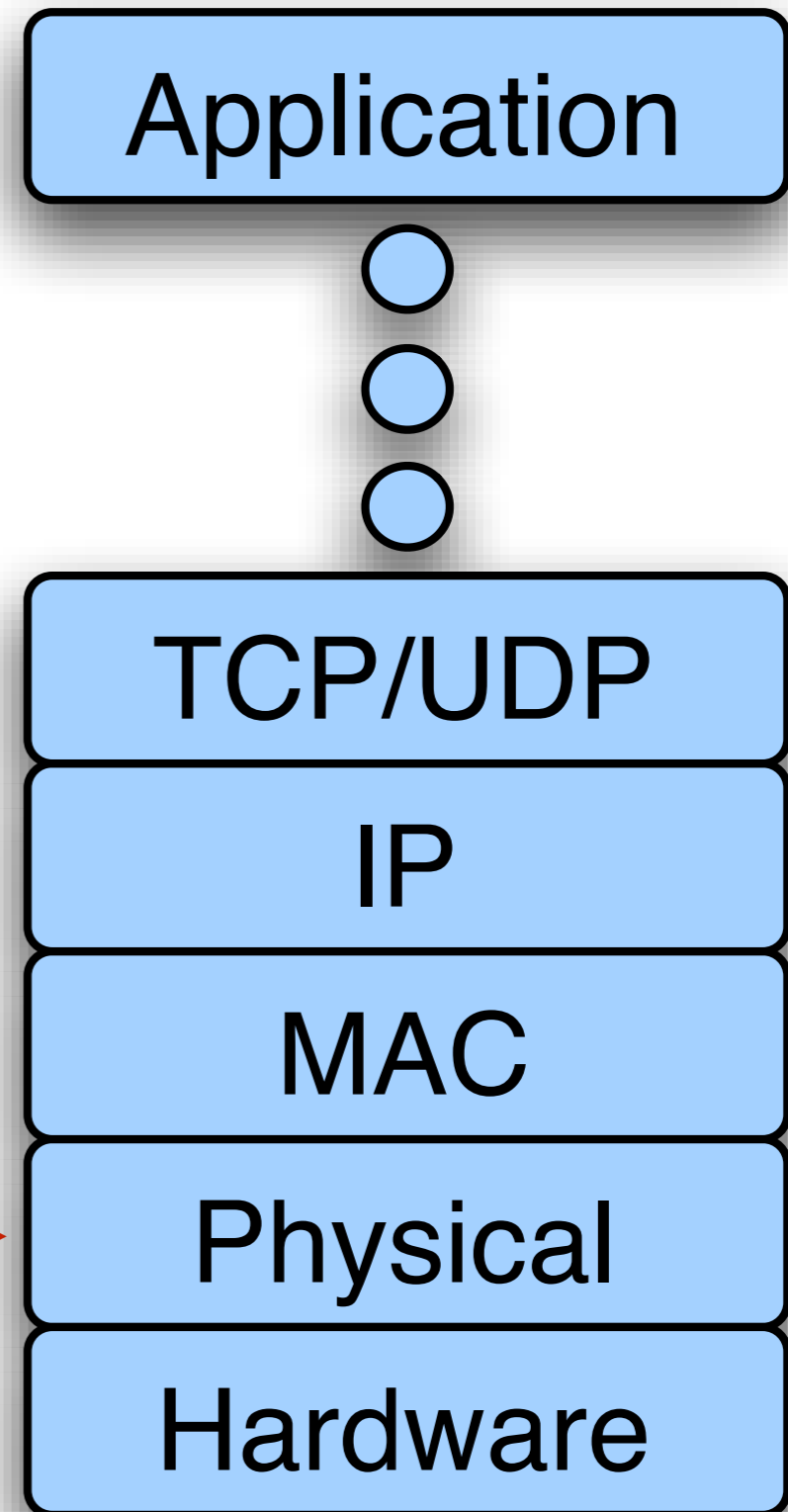
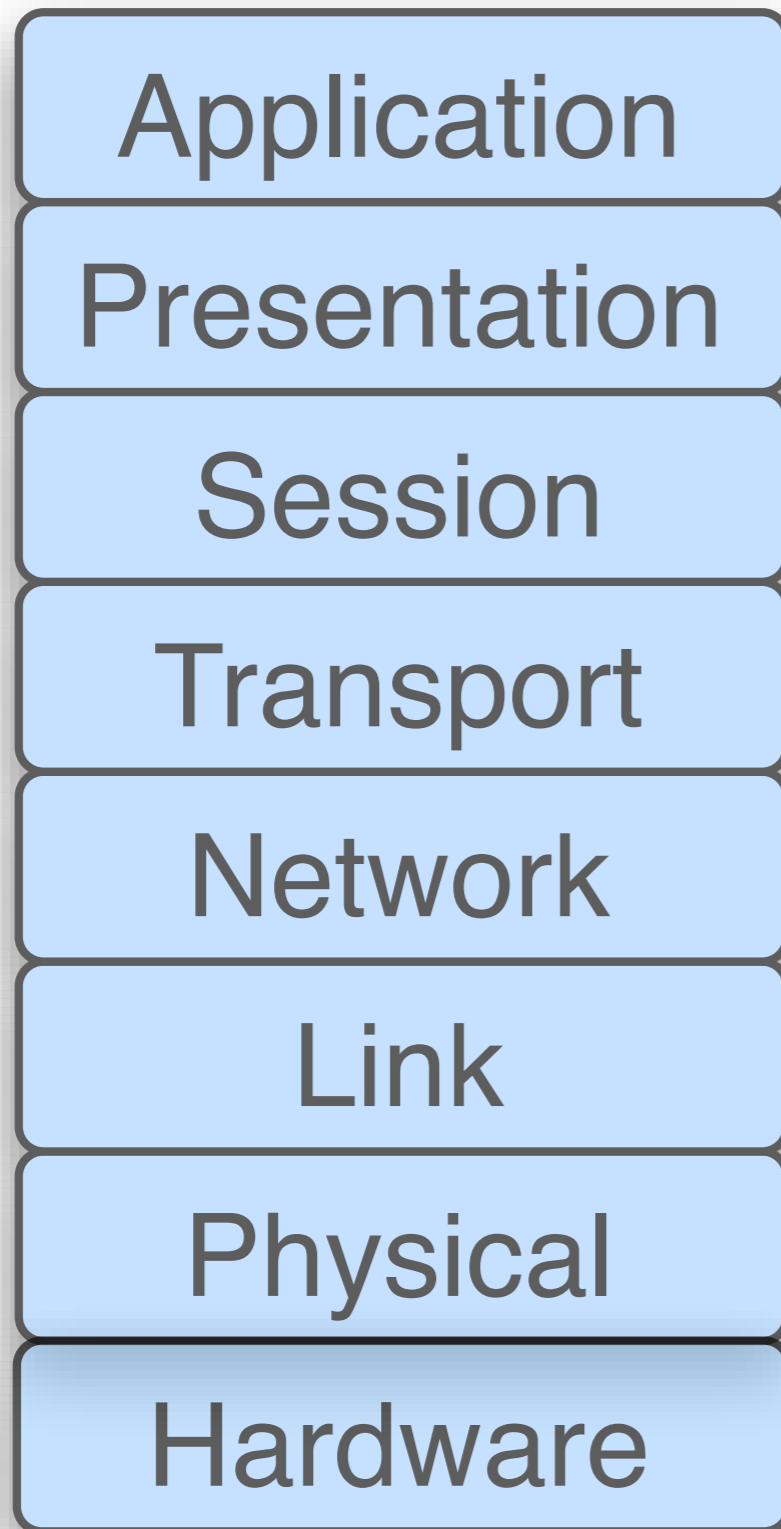
Physical Layer Basics



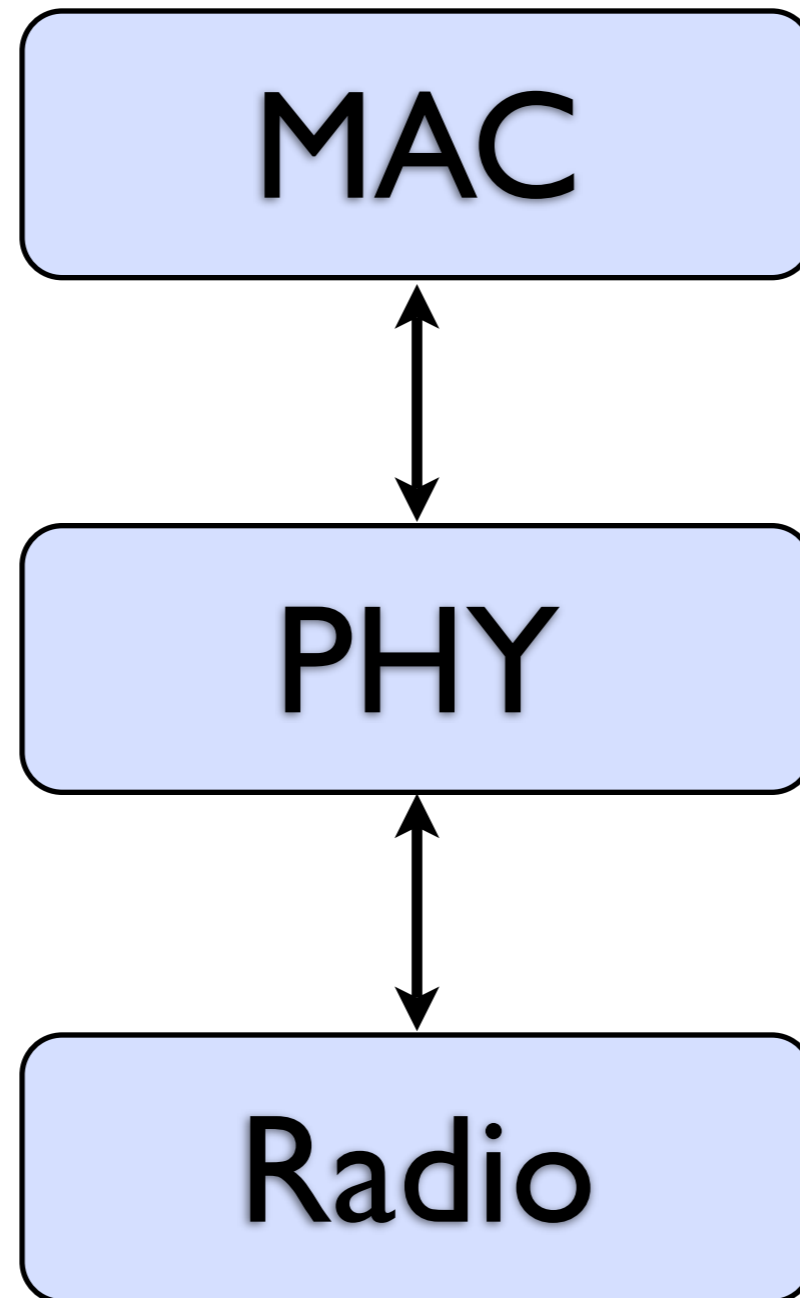
Physical Layer Basics



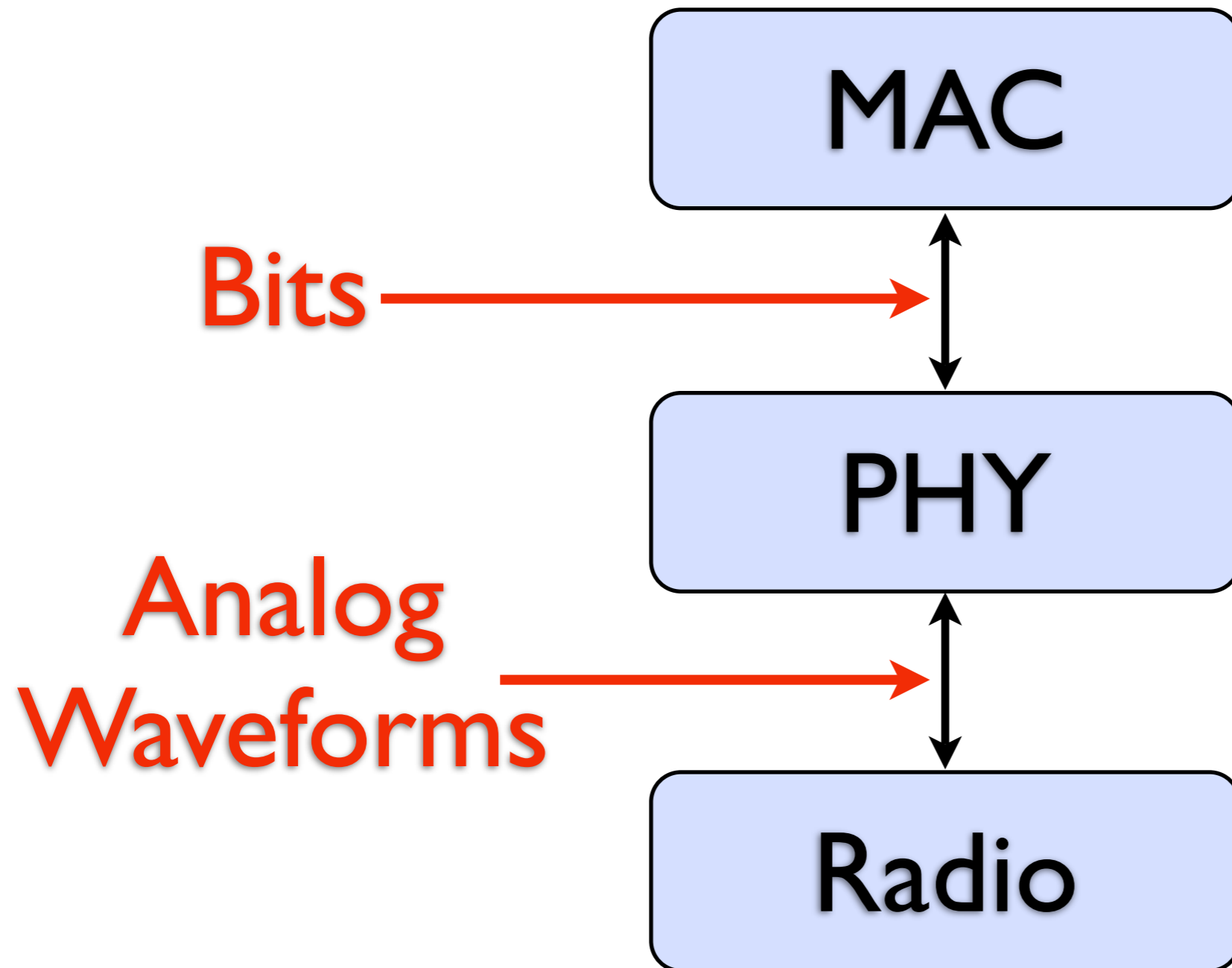
Physical Layer Basics



Physical Layer Basics

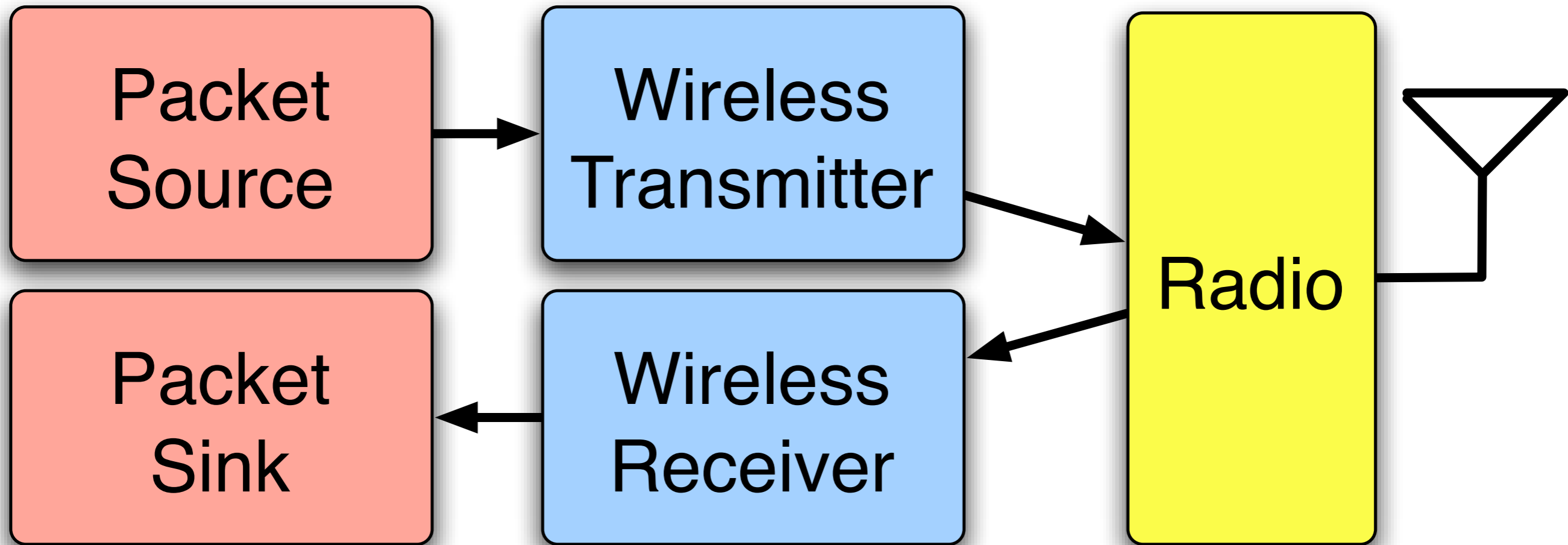


Physical Layer Basics



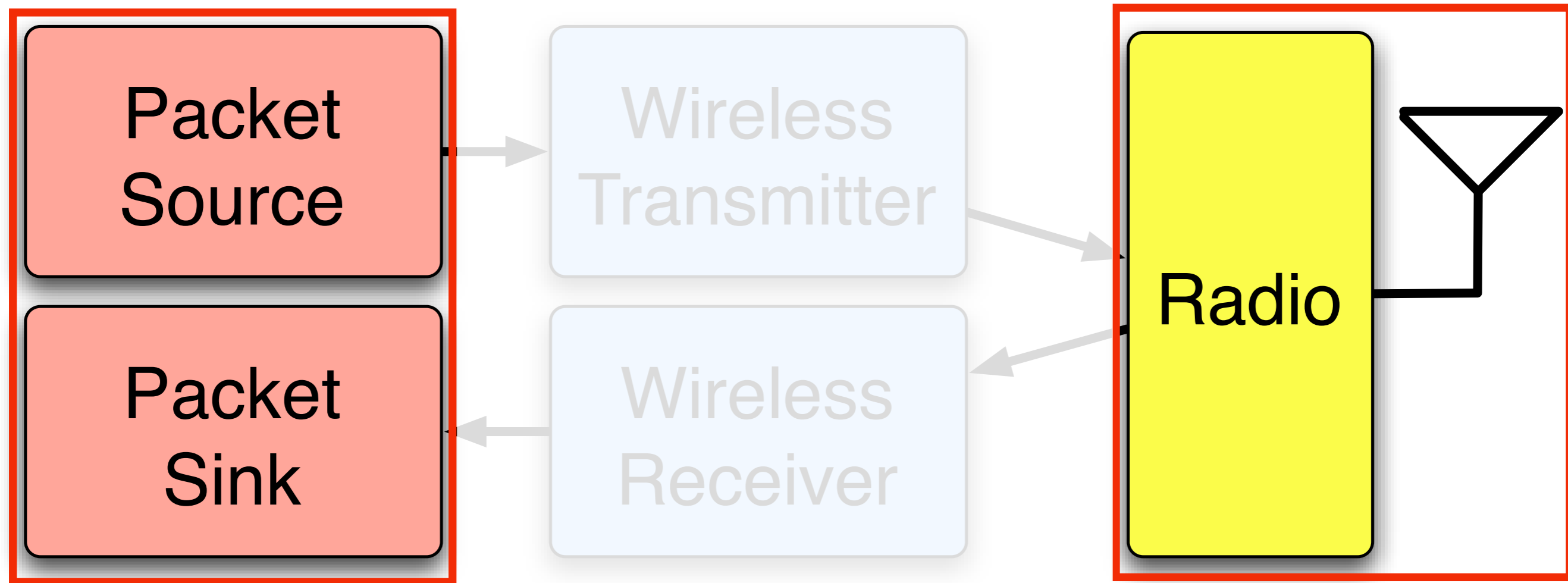
Physical Layer Basics

Simple Wireless Node



Physical Layer Basics

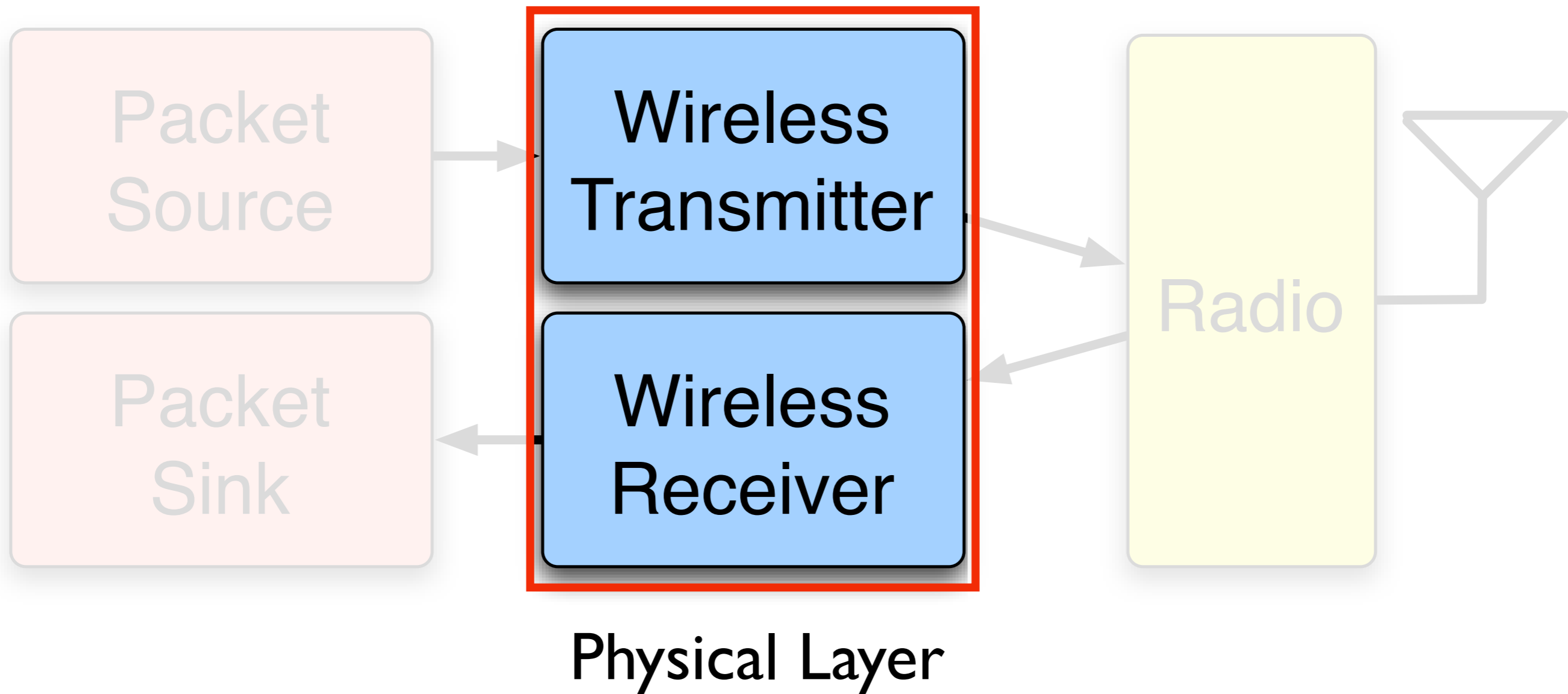
Simple Wireless Node



Somebody Else's Problem

Physical Layer Basics

Simple Wireless Node



PHY Design Flows

- WARPLab
 - MATLAB↔WARP Link
 - Very rapid prototyping of PHY algorithms
- Real-time PHY design
 - Low-level FPGA design
 - Putting it all together

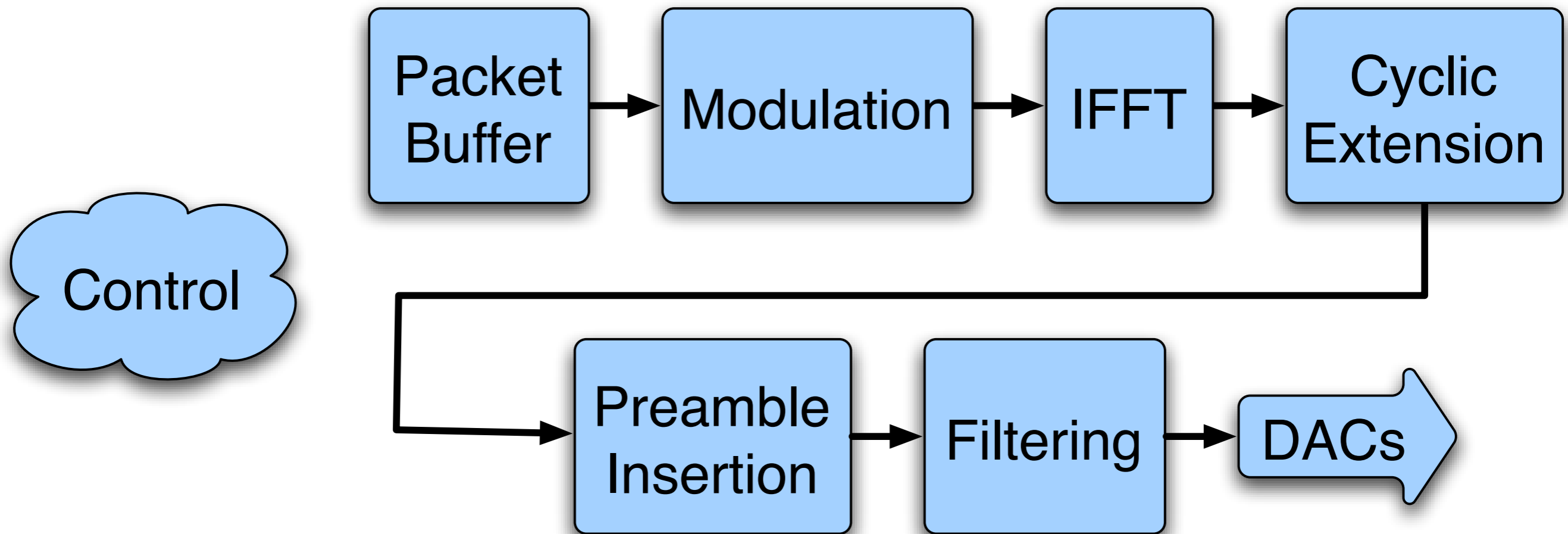
PHY Design Flows

- WARPLab
 - MATLAB↔WARP Link
 - Very rapid prototyping of PHY algorithms
- Real-time PHY design
 - Low-level FPGA design
 - Putting it all together

PHY Example: OFDM

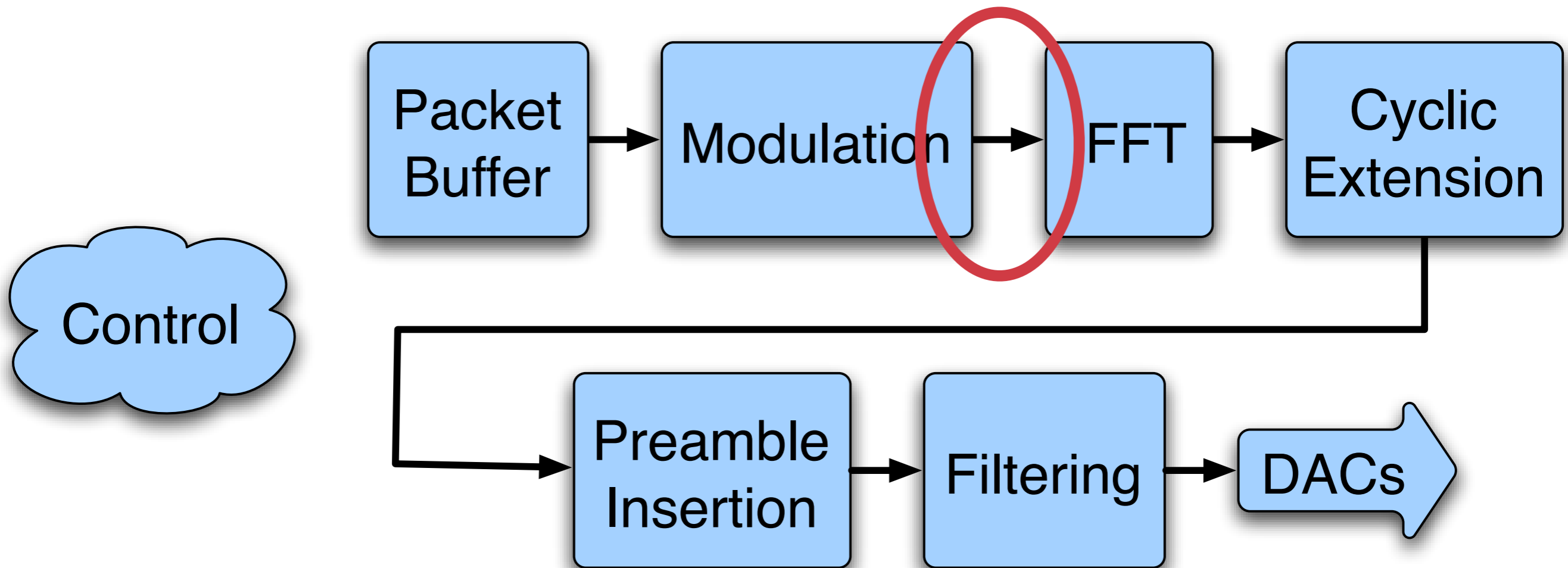
- Designed for wireless networking
 - Modeled on 802.11a (but not compliant)
- Packet-based OFDM transceiver
 - Packets source/sink in PowerPC code
- Wideband, real-time design
 - 4 cycles per sample
 - 10 MHz bandwidth at 40 MHz clock
- Full synchronization for standalone operation
- Implemented entirely in System Generator

PHY Example: OFDM Tx

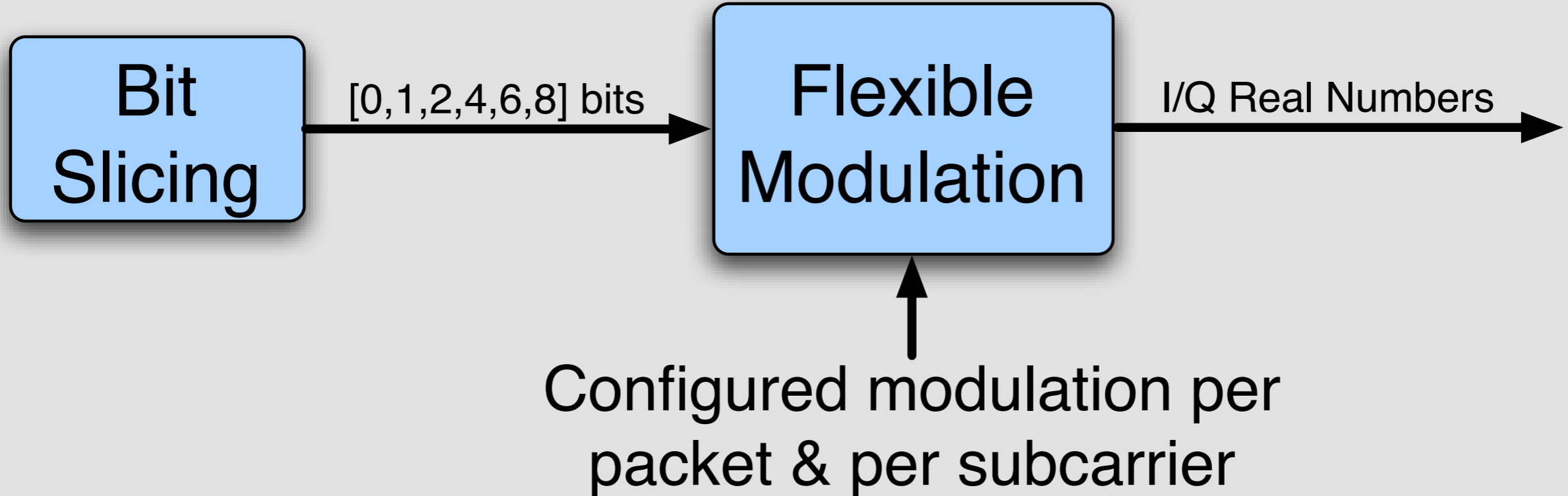
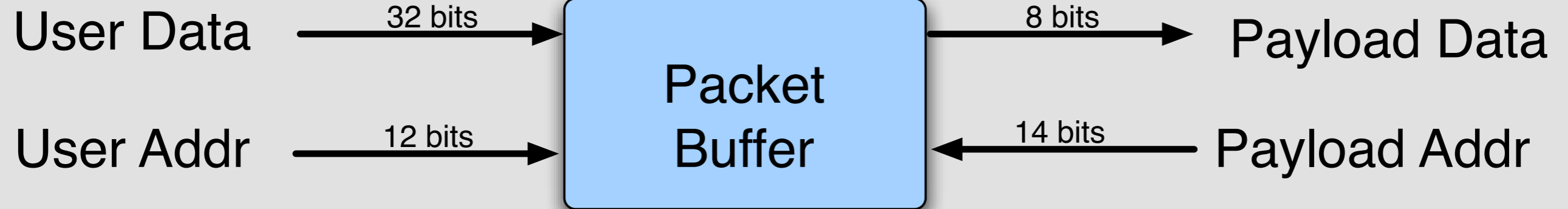


PHY Example: OFDM Tx

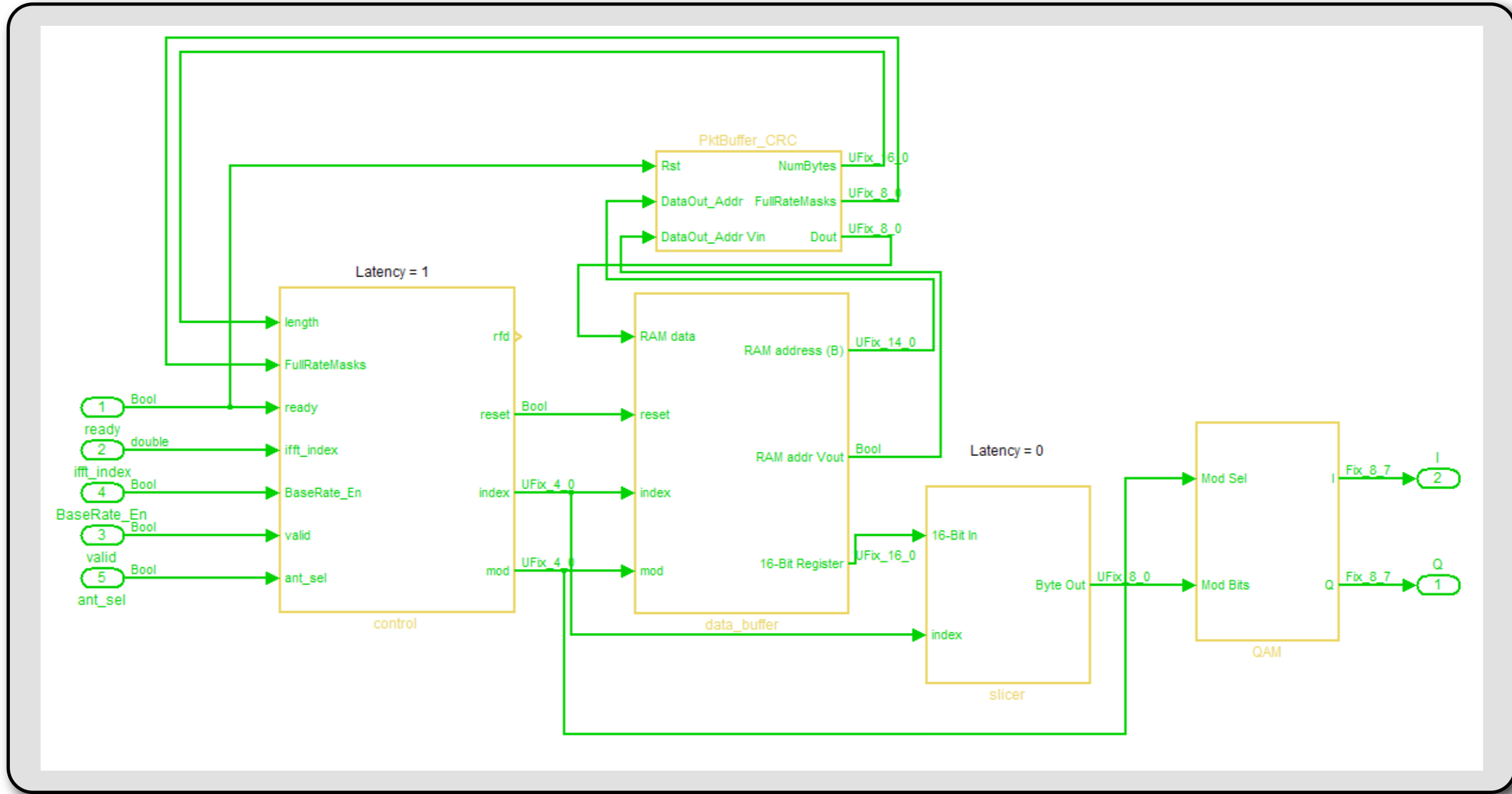
No Serial/Parallel Conversion!



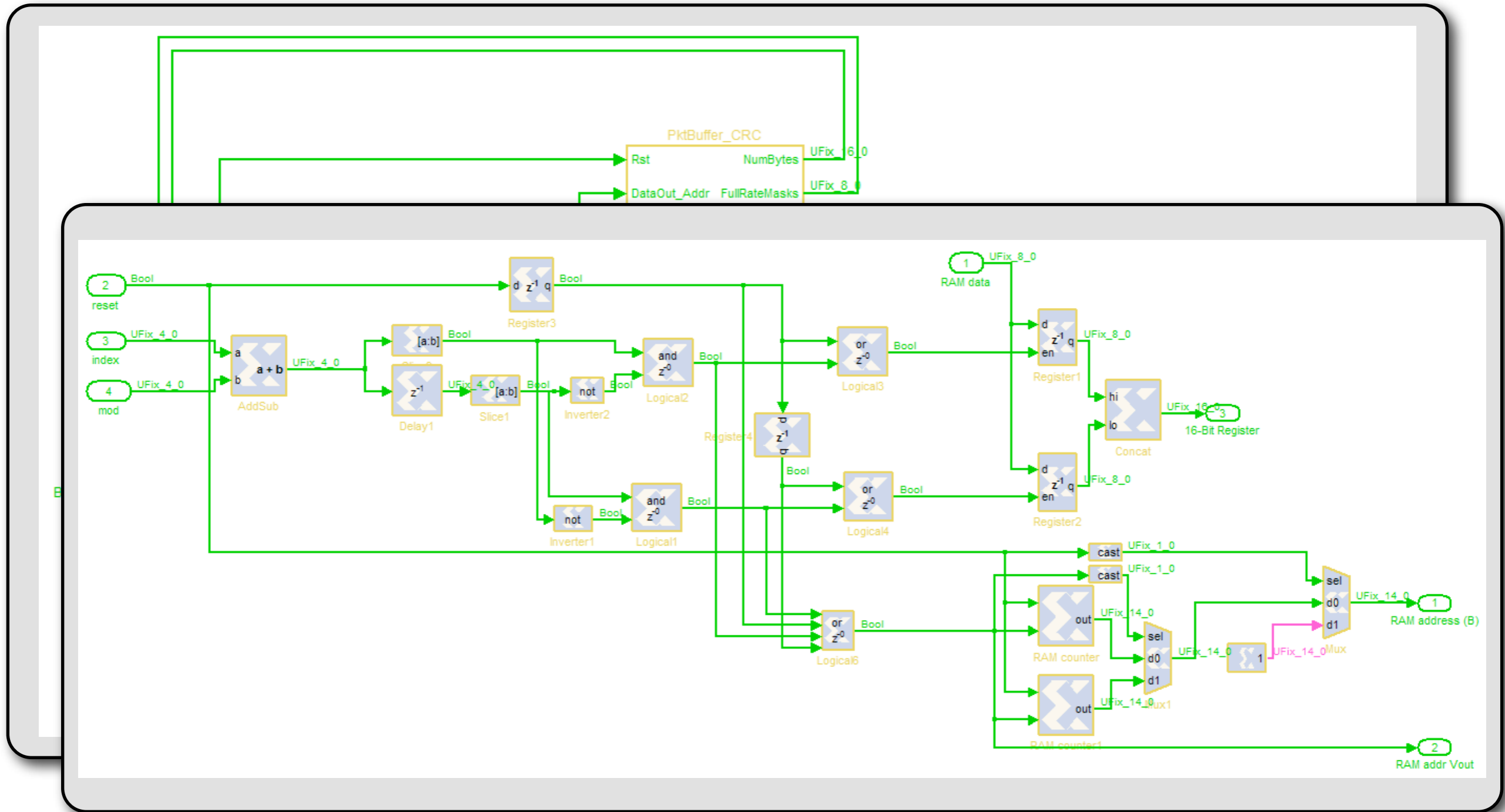
PHY Example: OFDM Tx



PHY Example: OFDM Tx

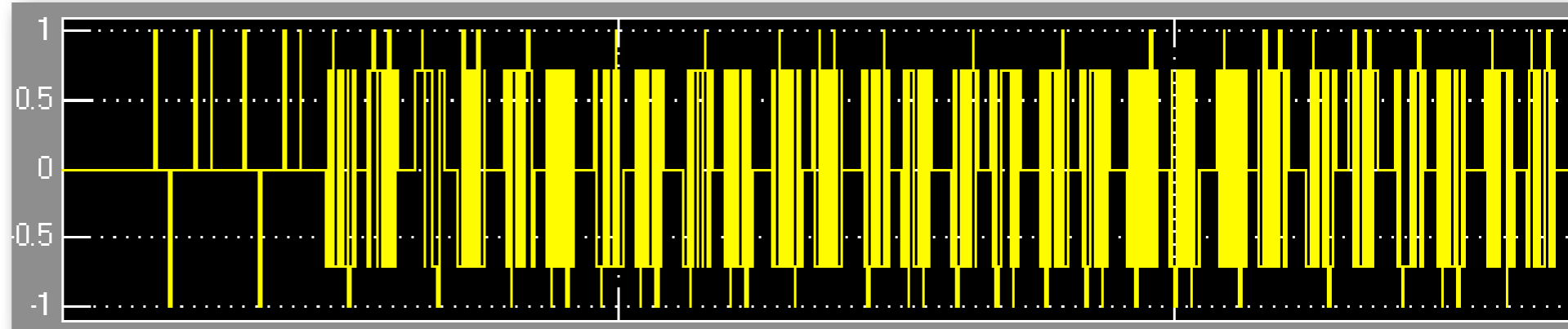


PHY Example: OFDM Tx

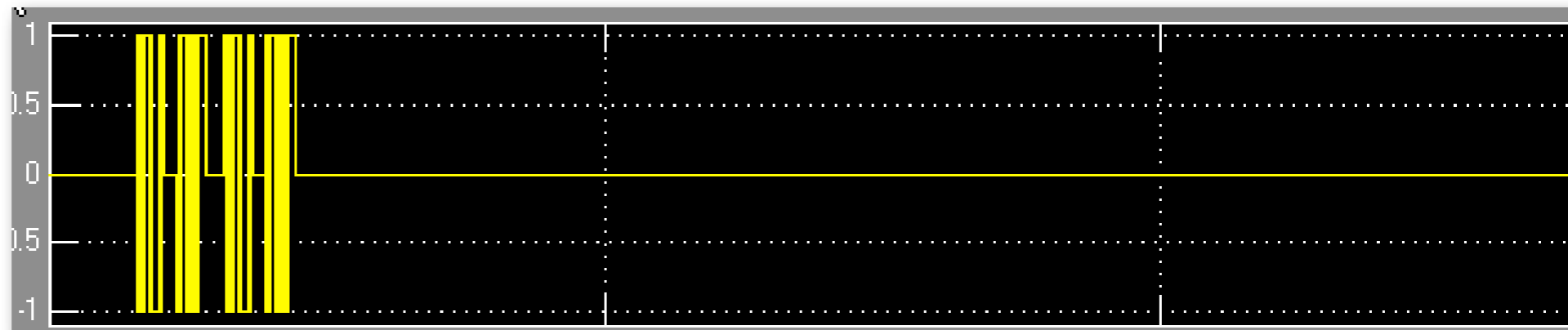


PHY Example: OFDM Tx

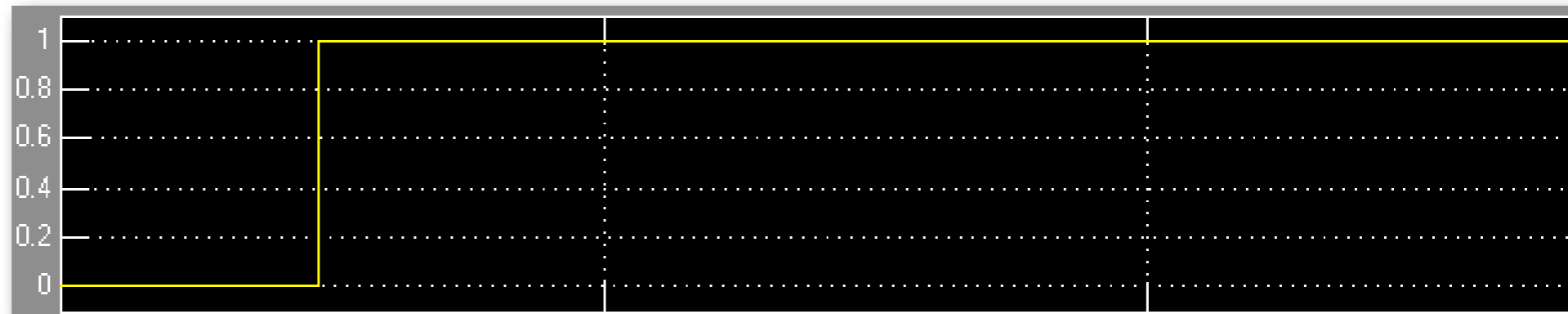
Modulator
Output



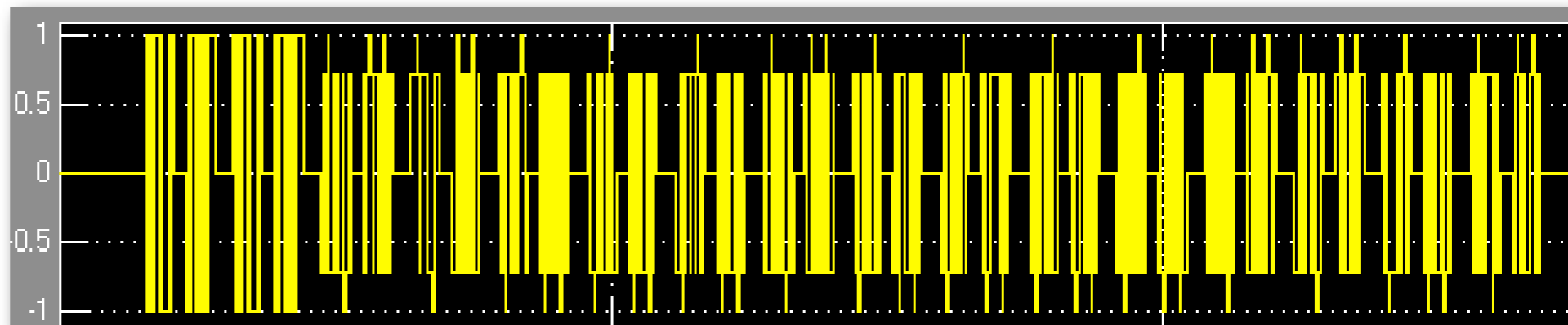
Stored Training
Sequence



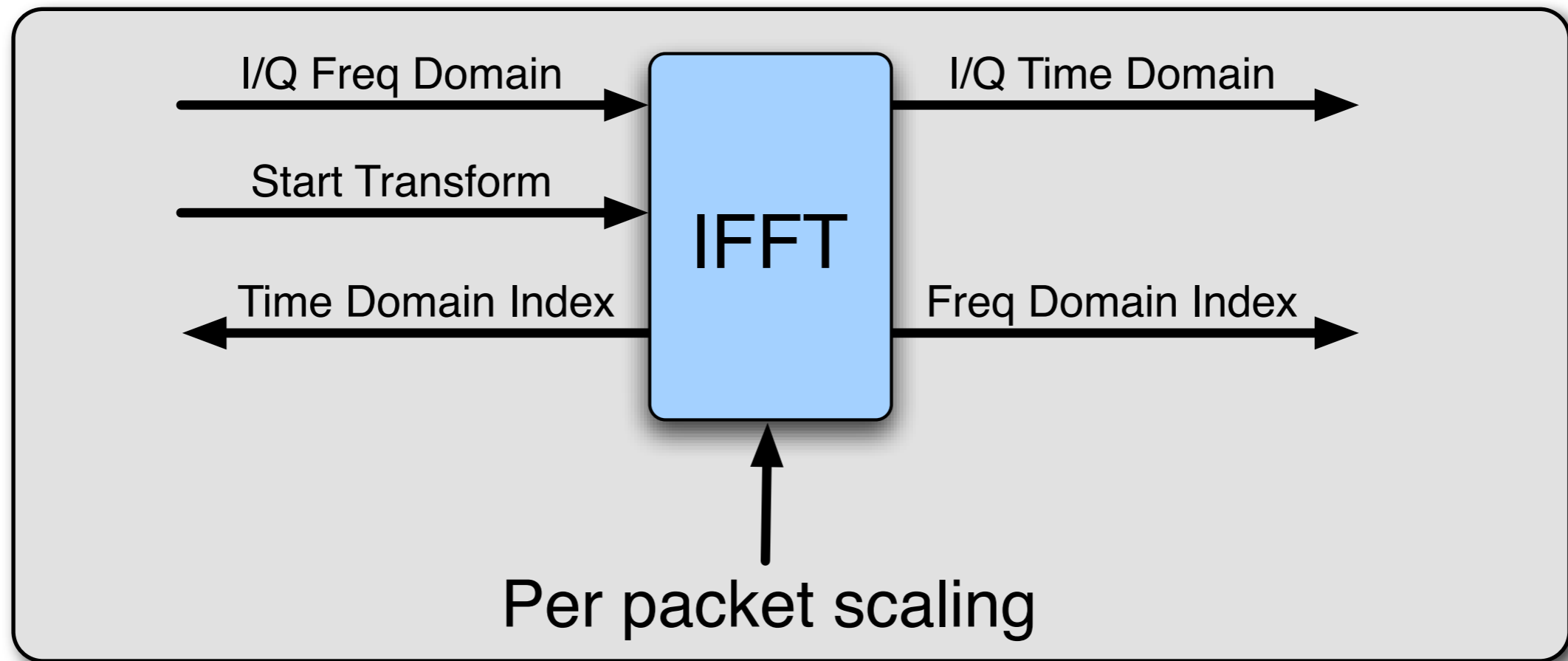
Source Mux
Select



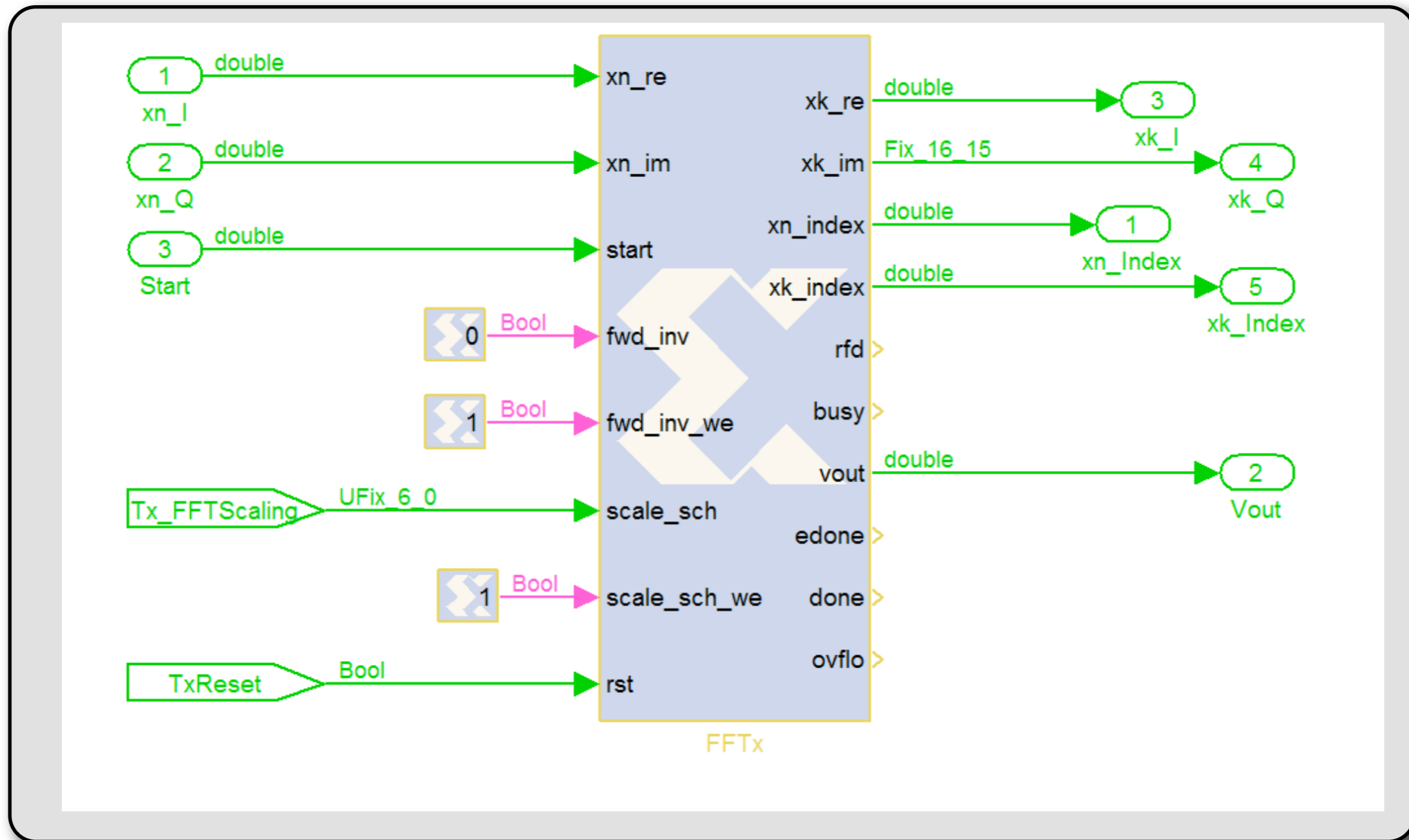
Input
IFFT



PHY Example: OFDM Tx



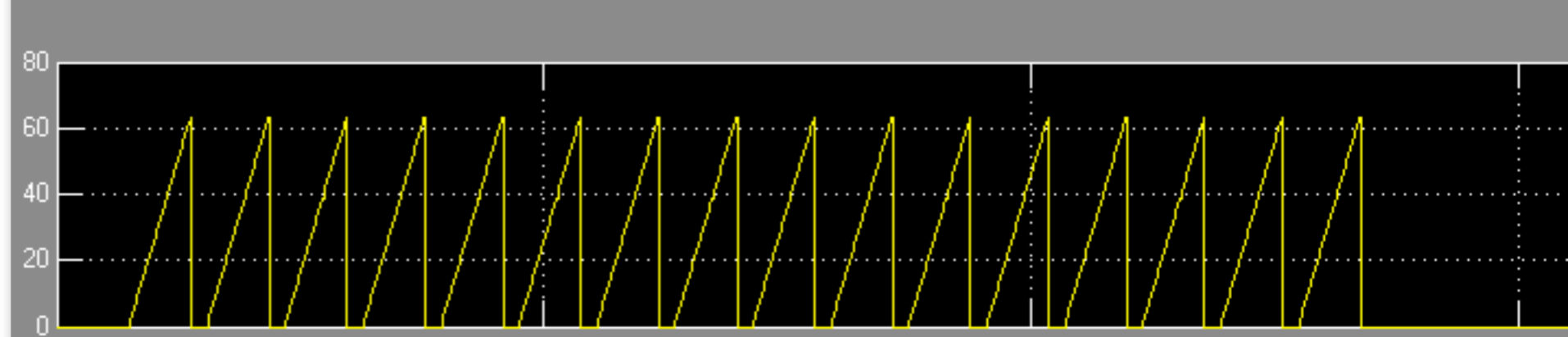
PHY Example: OFDM Tx



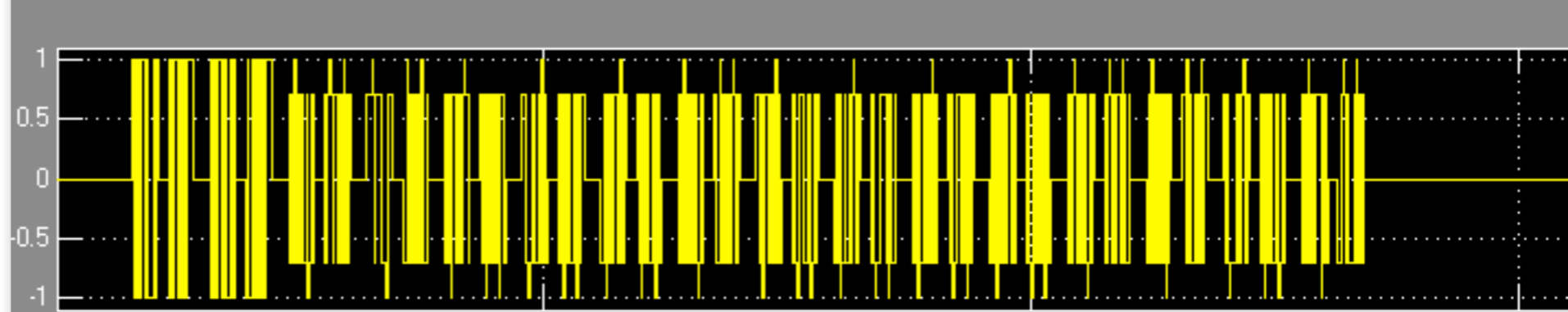
IFFT Start



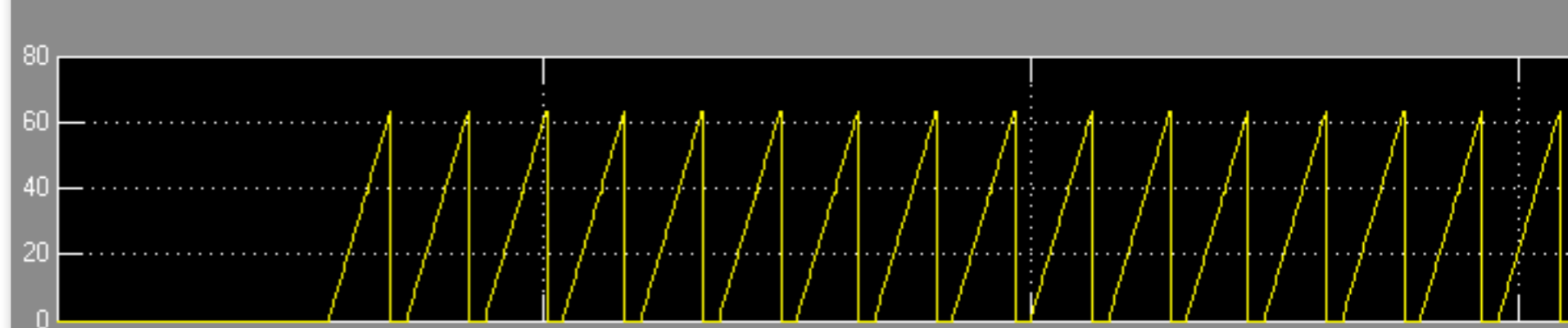
Input Index



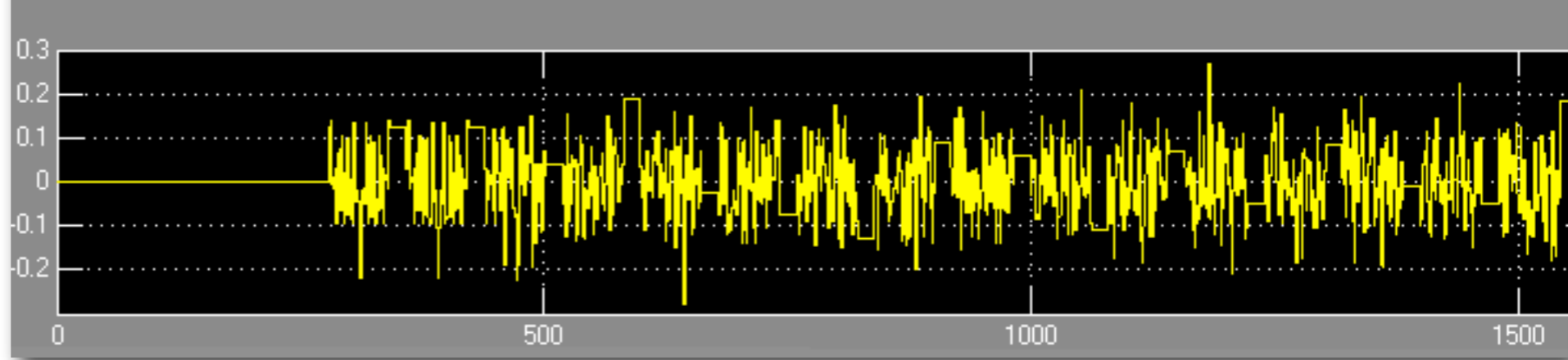
Input Data



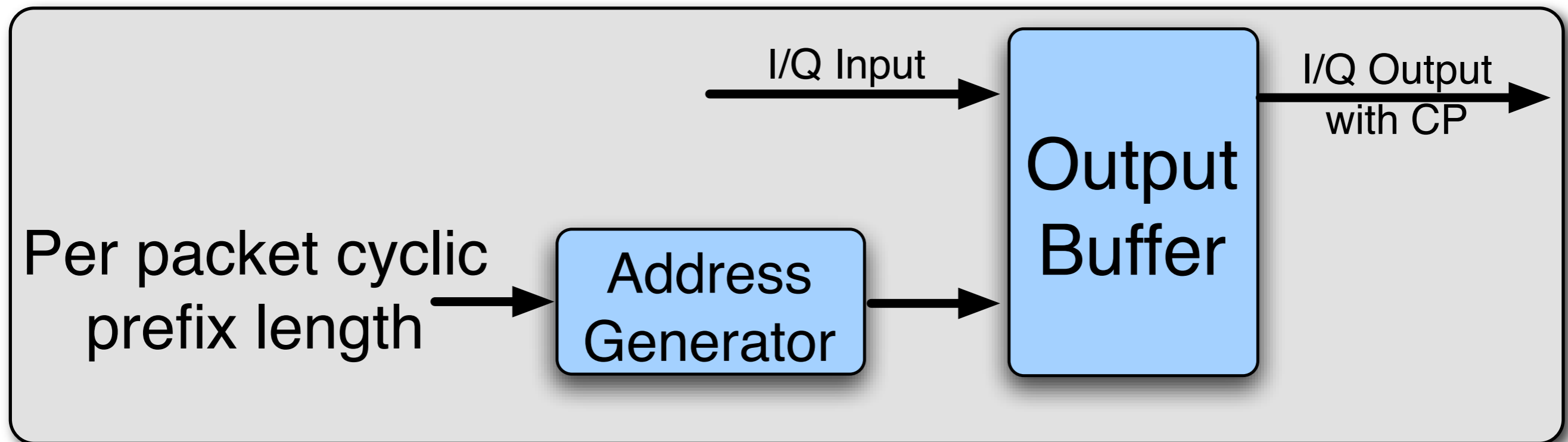
Output Index



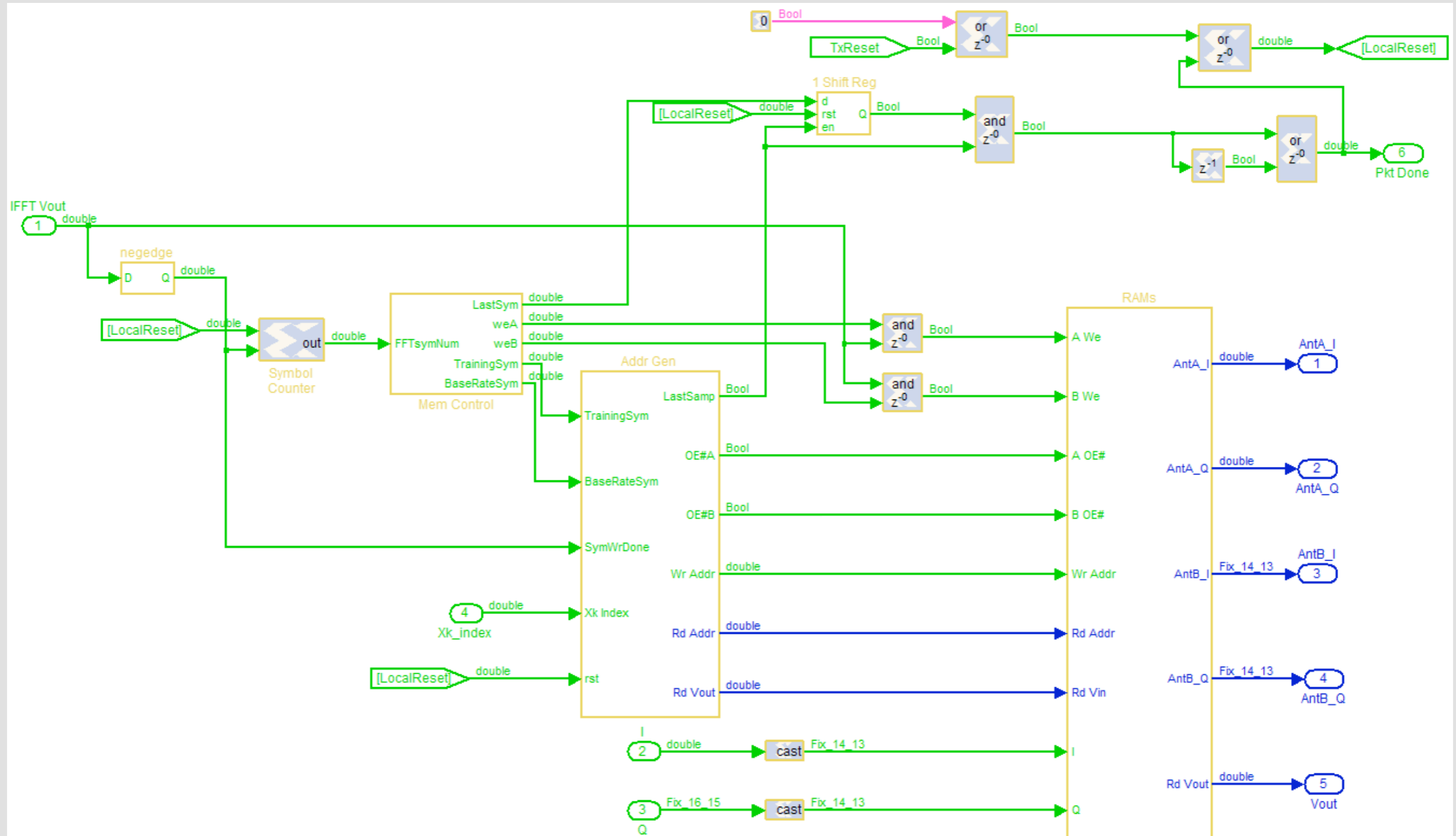
Output Data



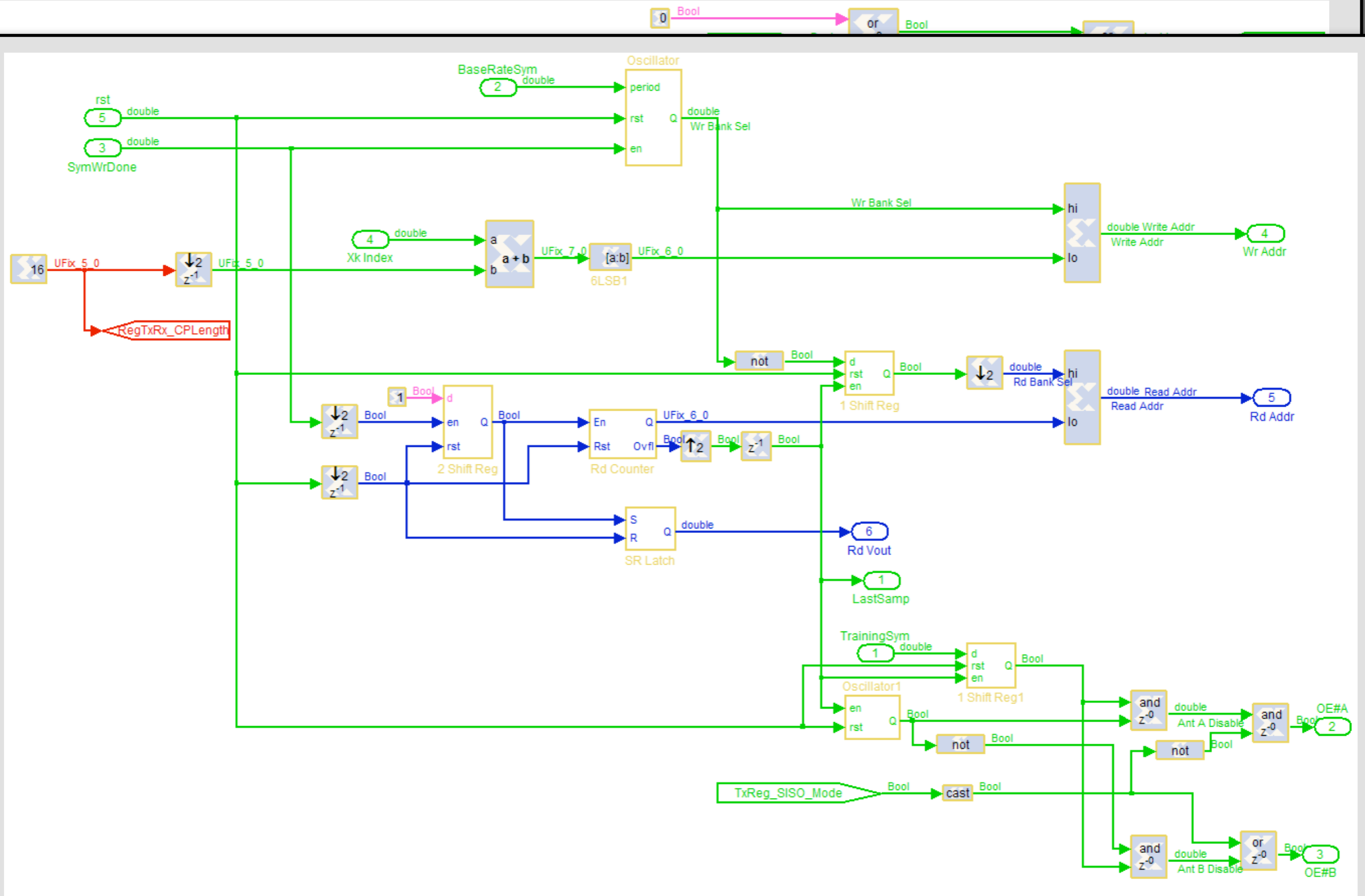
PHY Example: OFDM Tx



PHY Example: OFDM Tx

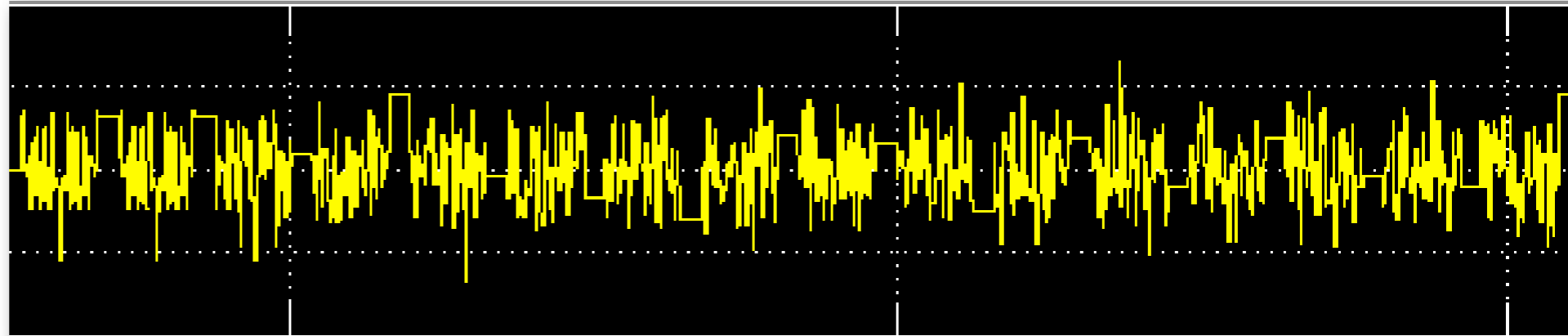


PHY Example: OFDM Tx

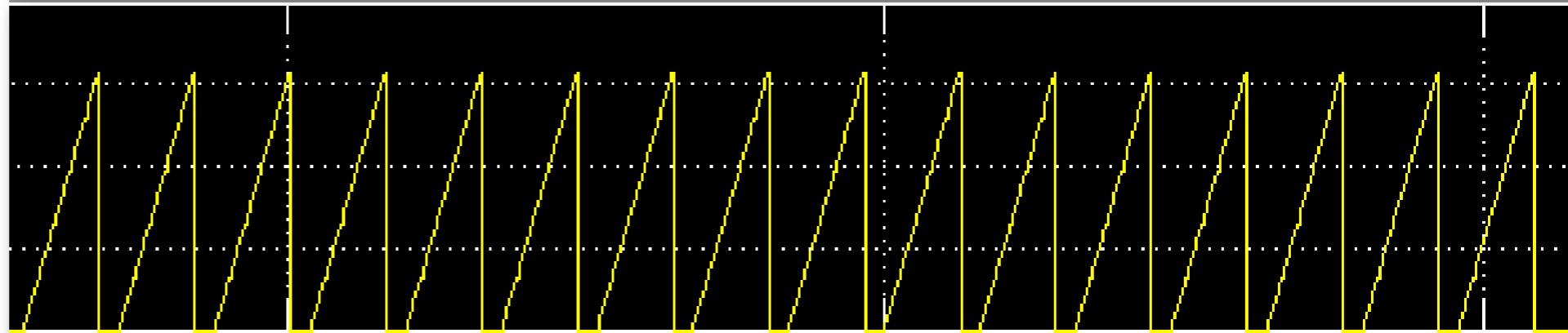


PHY Example: OFDM Tx

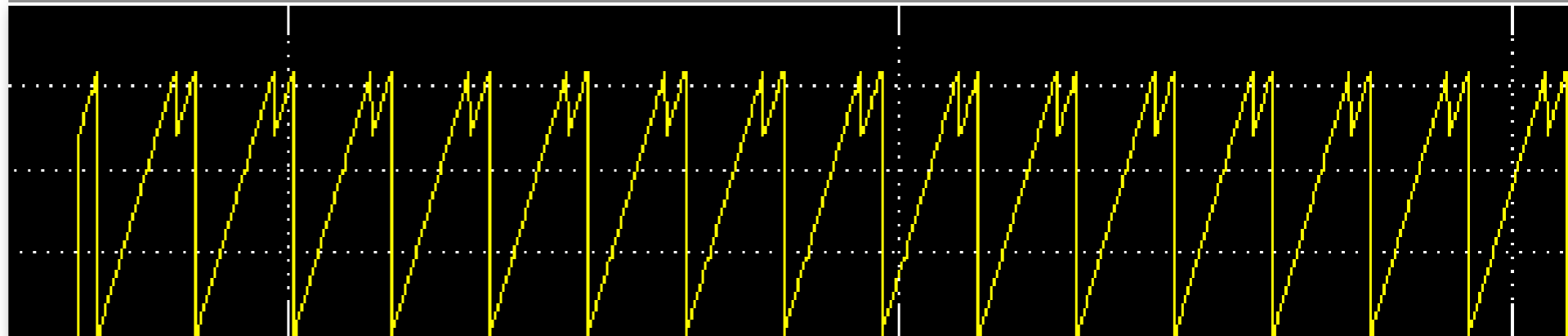
IFFT Output



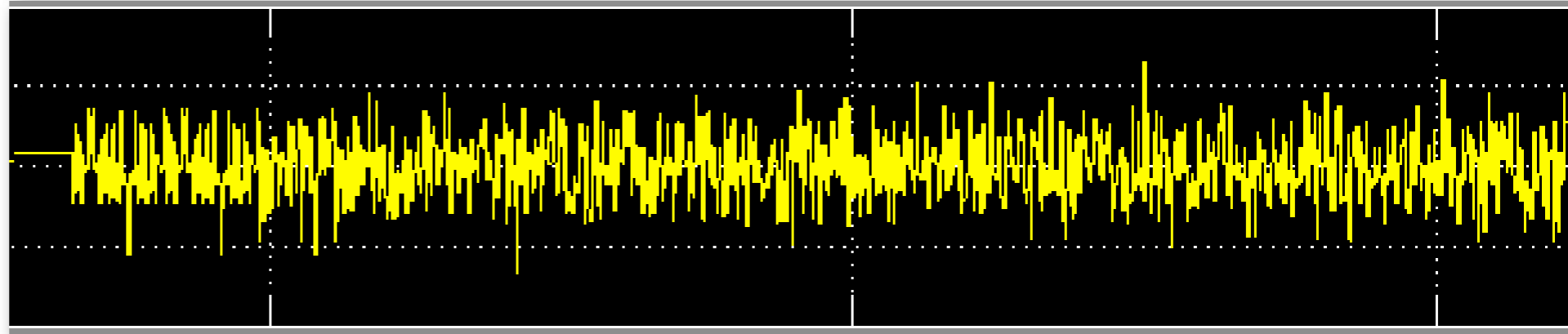
RAM Write
Address



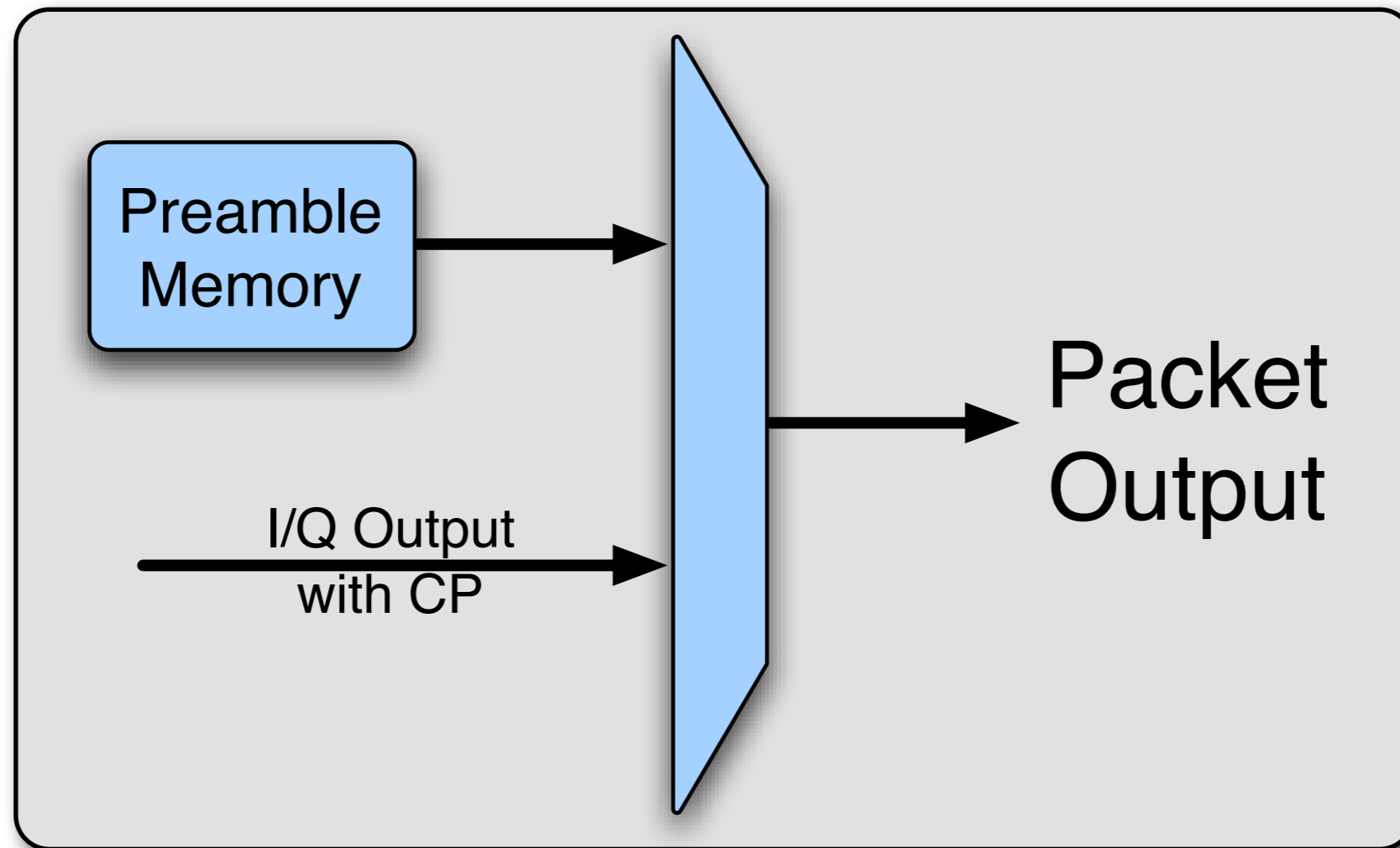
RAM Read
Address



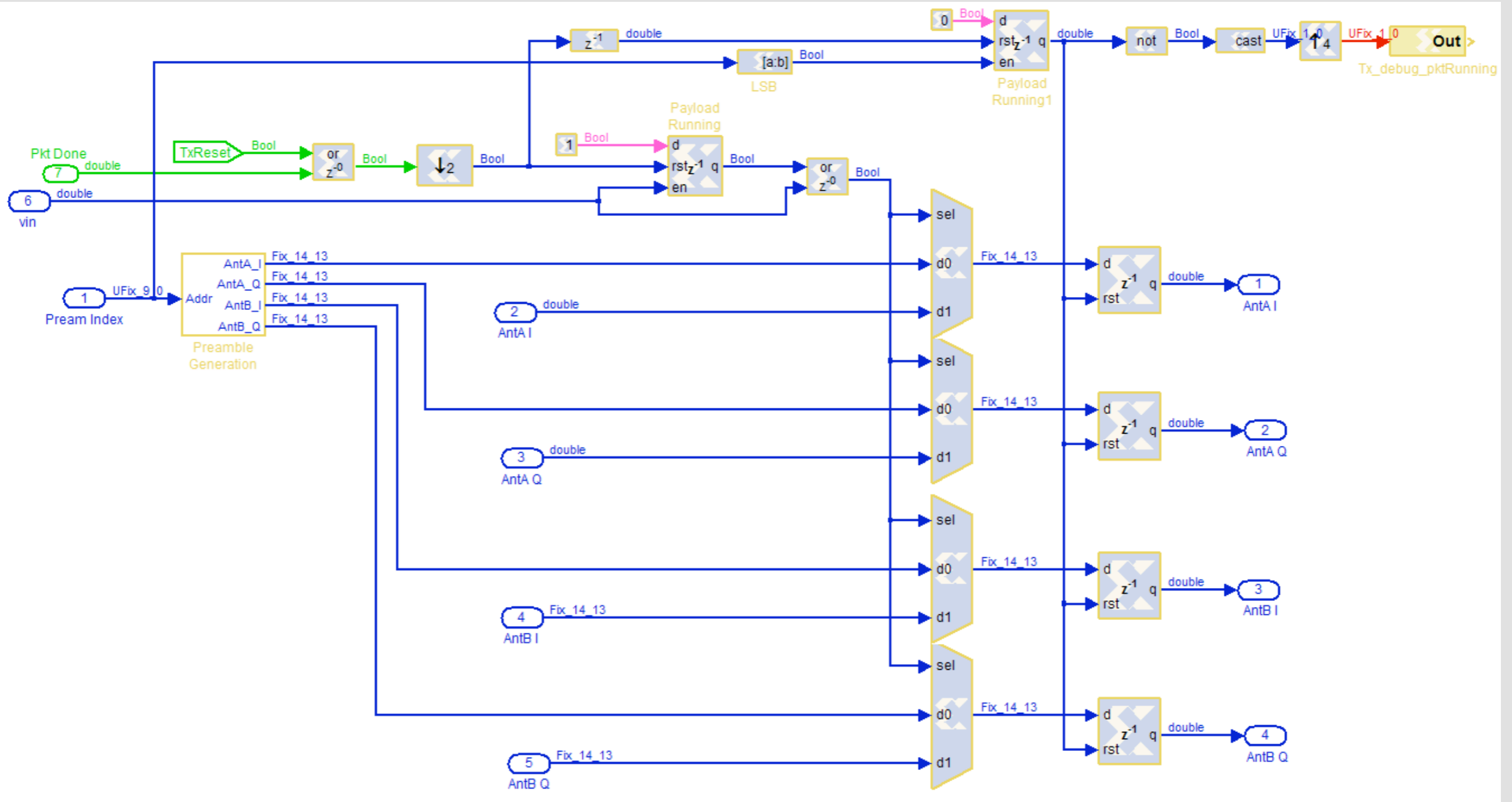
Cyclically
Extended



PHY Example: OFDM Tx



PHY Example: OFDM Tx



PHY Example: OFDM Tx

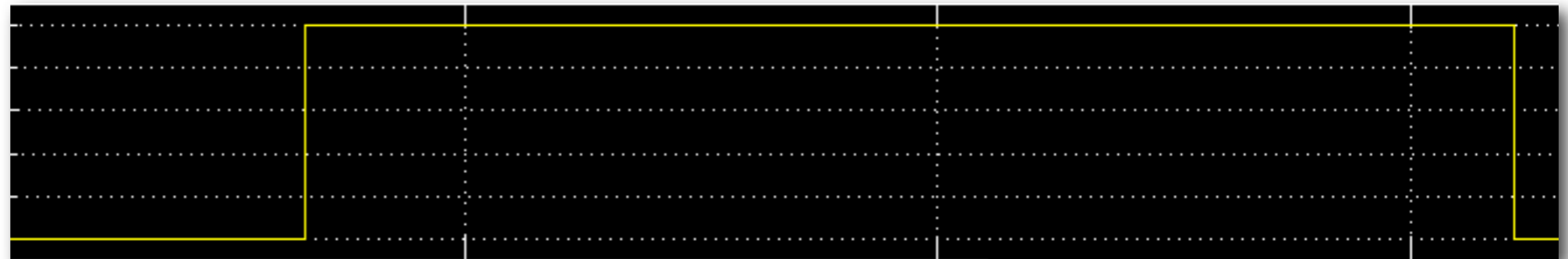
Stored
Preamble



OFDM
Output



Output Mux
Selection



Final Output
to DACs

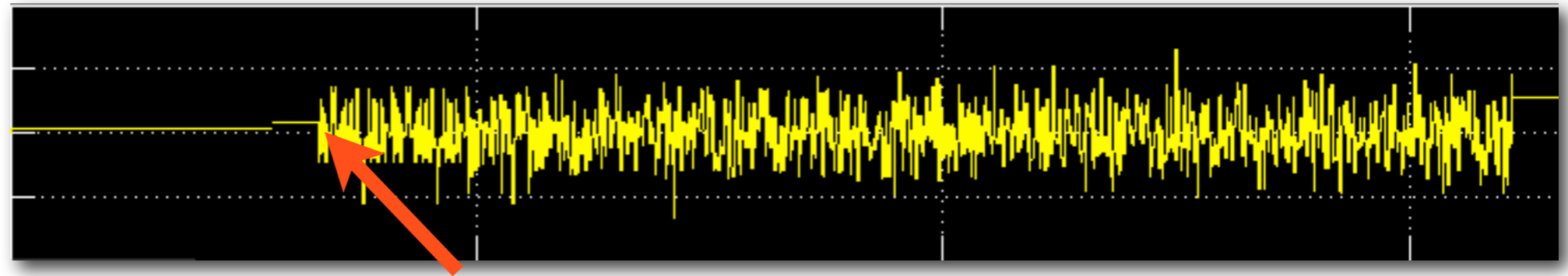


PHY Example: OFDM Tx

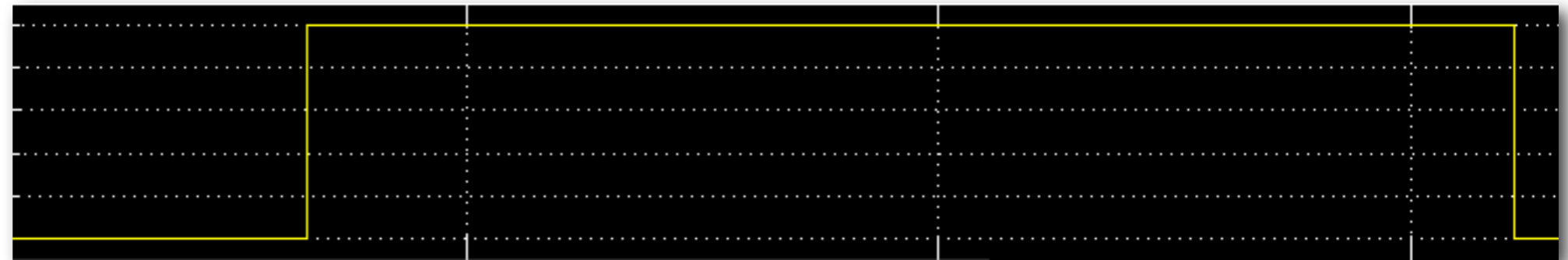
Stored
Preamble



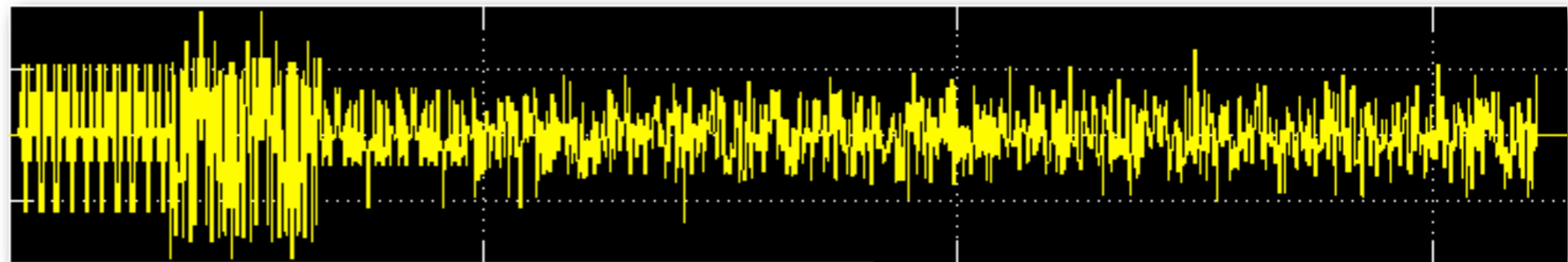
OFDM
Output



Output Mux
Selection



Final Output
to DACs

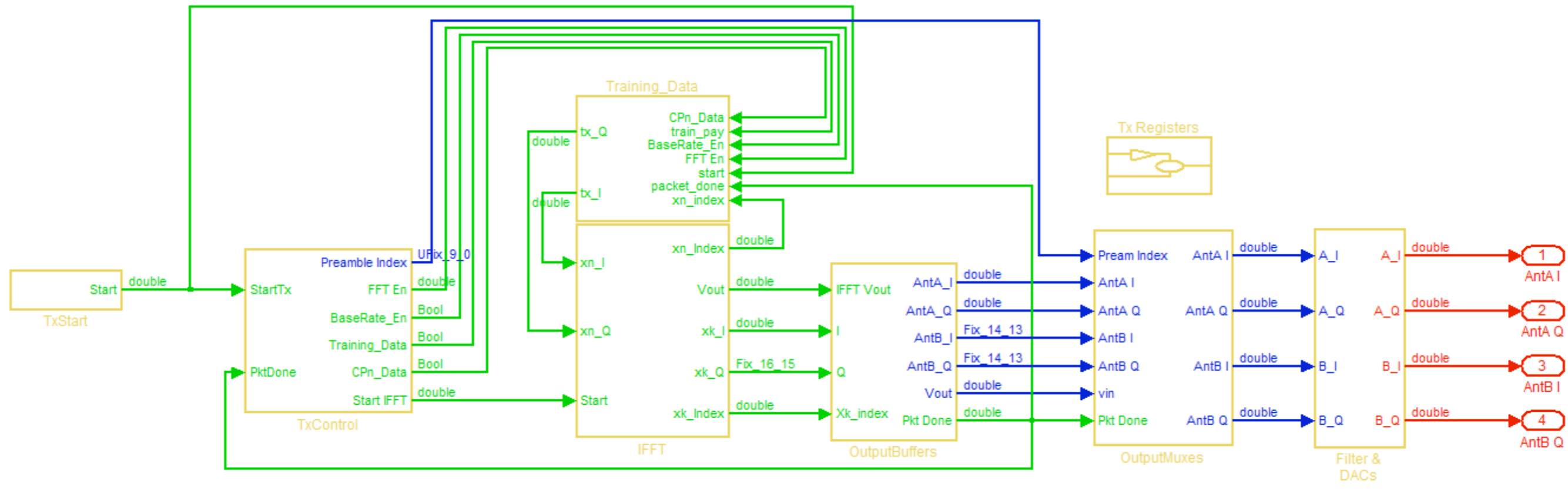


PHY Example: OFDM Tx



- Per packet configuration drives control system
 - Number of training symbols
 - Number of bytes
 - Modulation choices
- Block specific control blocks
 - IFFT start signal
 - Memory address generation
- Triggers & status between core and PowerPC

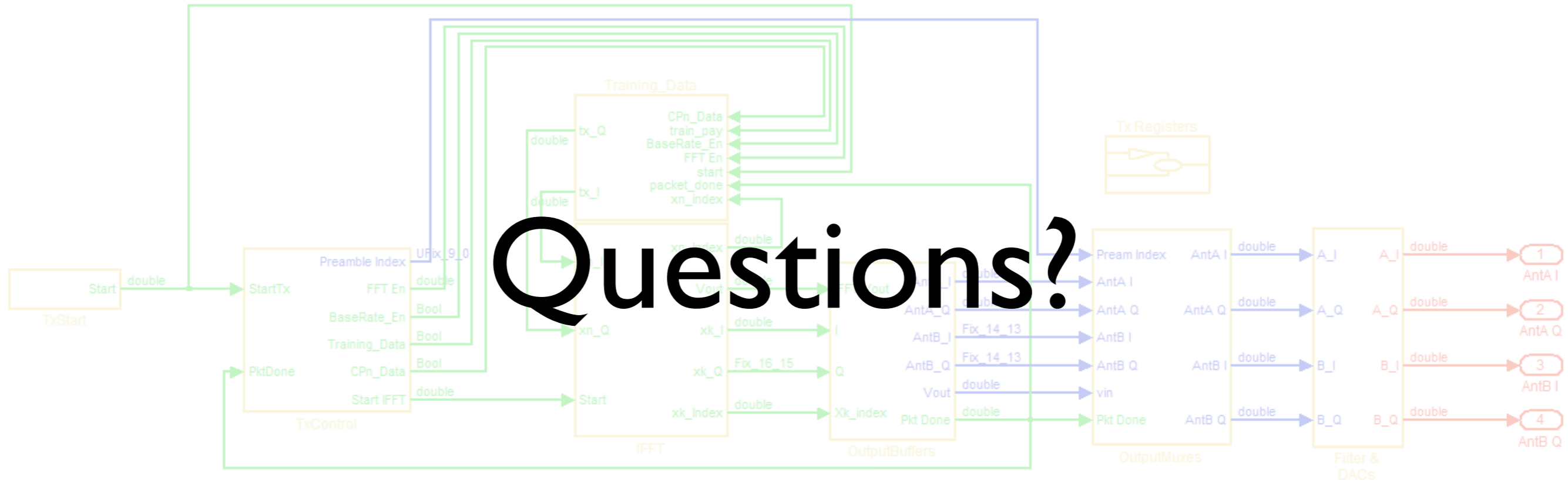
PHY Example: OFDM Tx



Complete model is available in the WARP repository

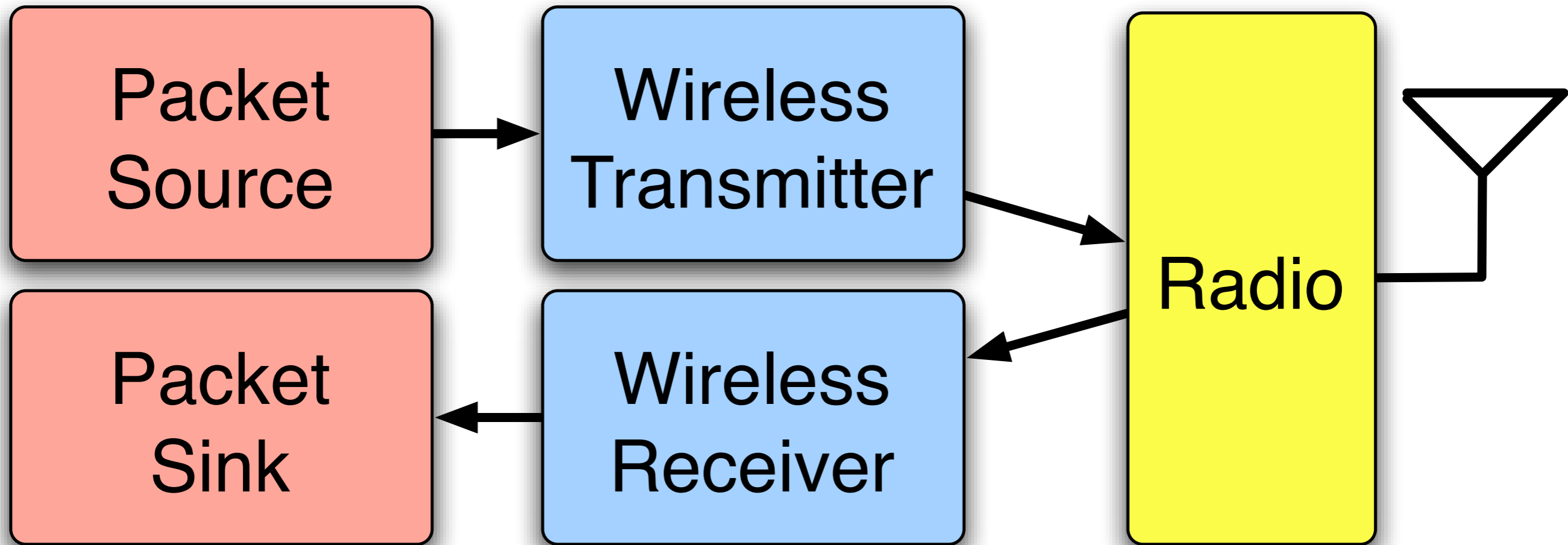
You'll use this model all day tomorrow

Questions?



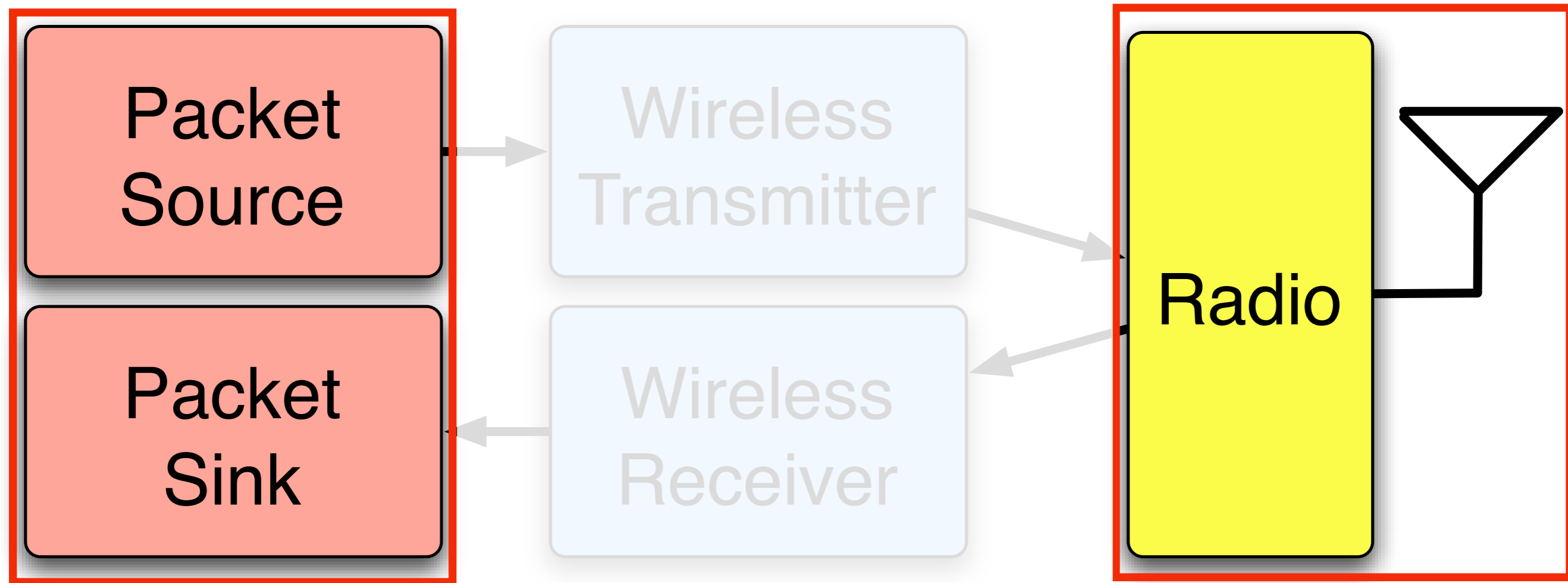
Physical Layer Basics

Simple Wireless Node



Physical Layer Basics

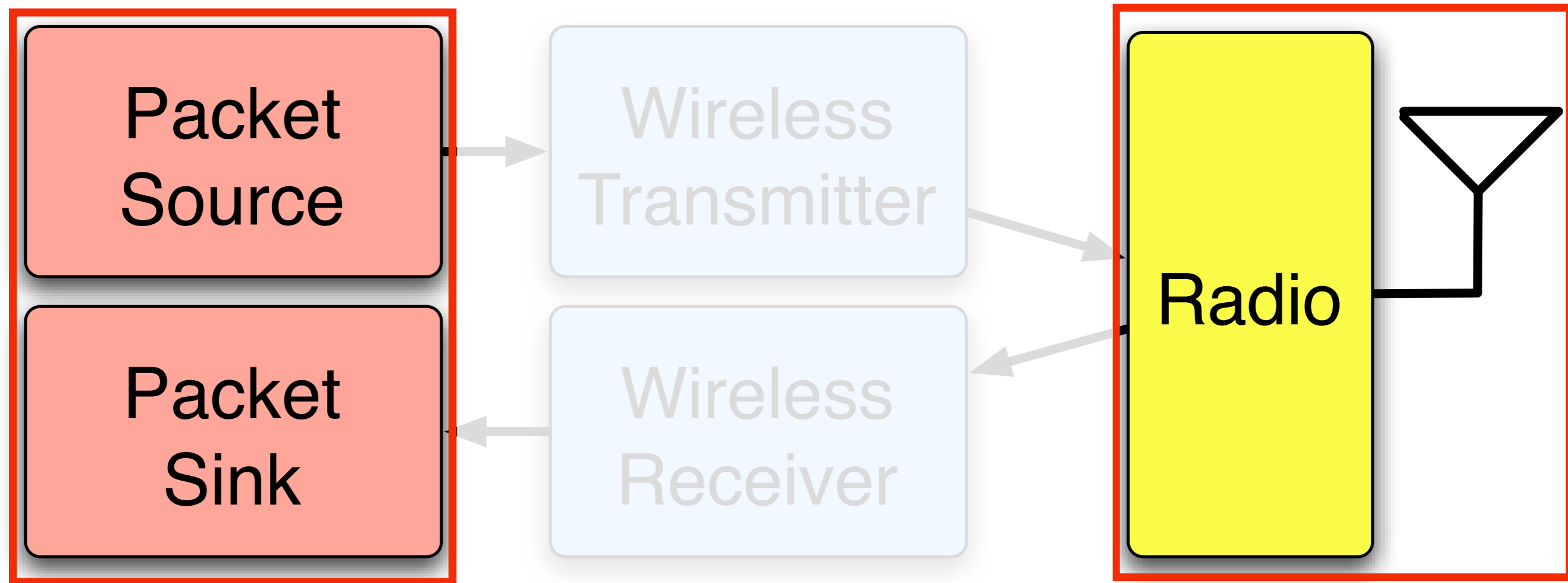
Simple Wireless Node



Somebody Else's Problem

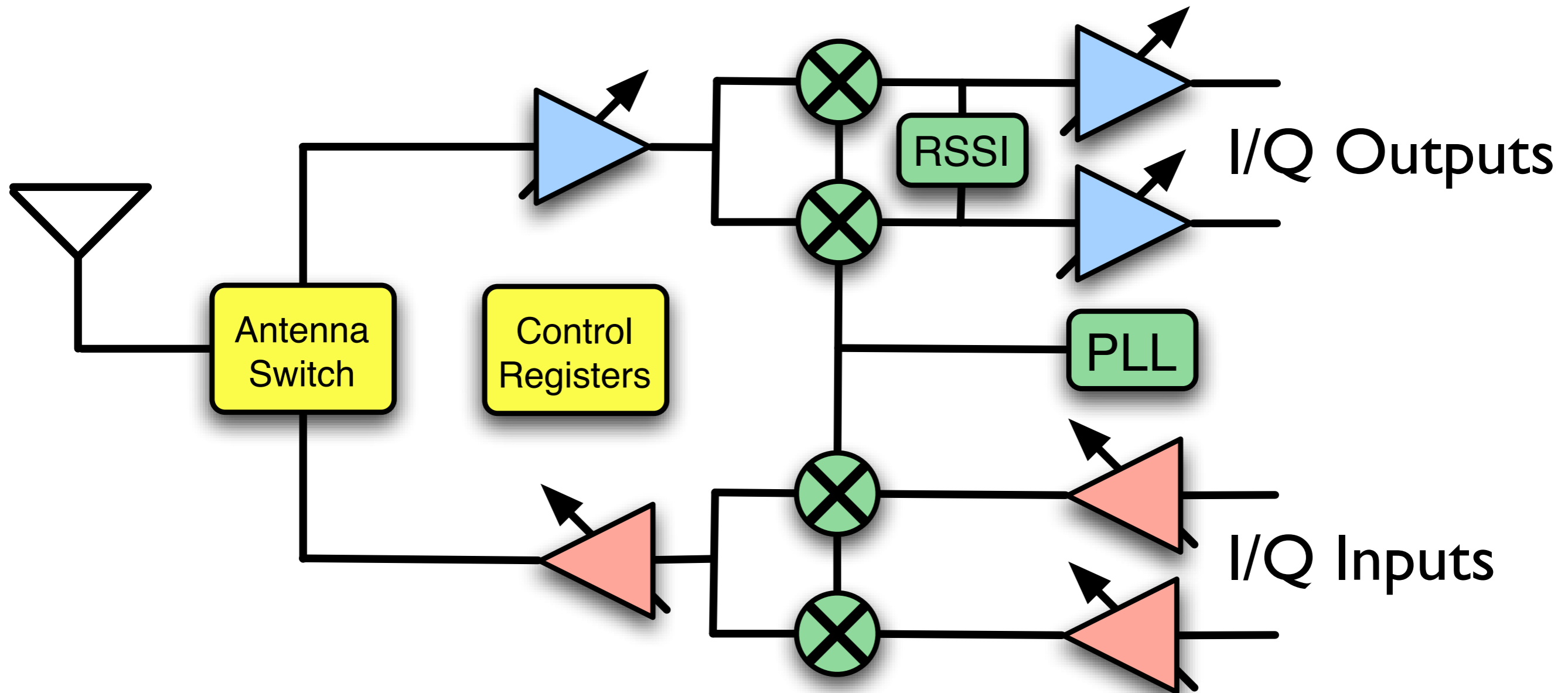
Physical Layer Basics

Simple Wireless Node

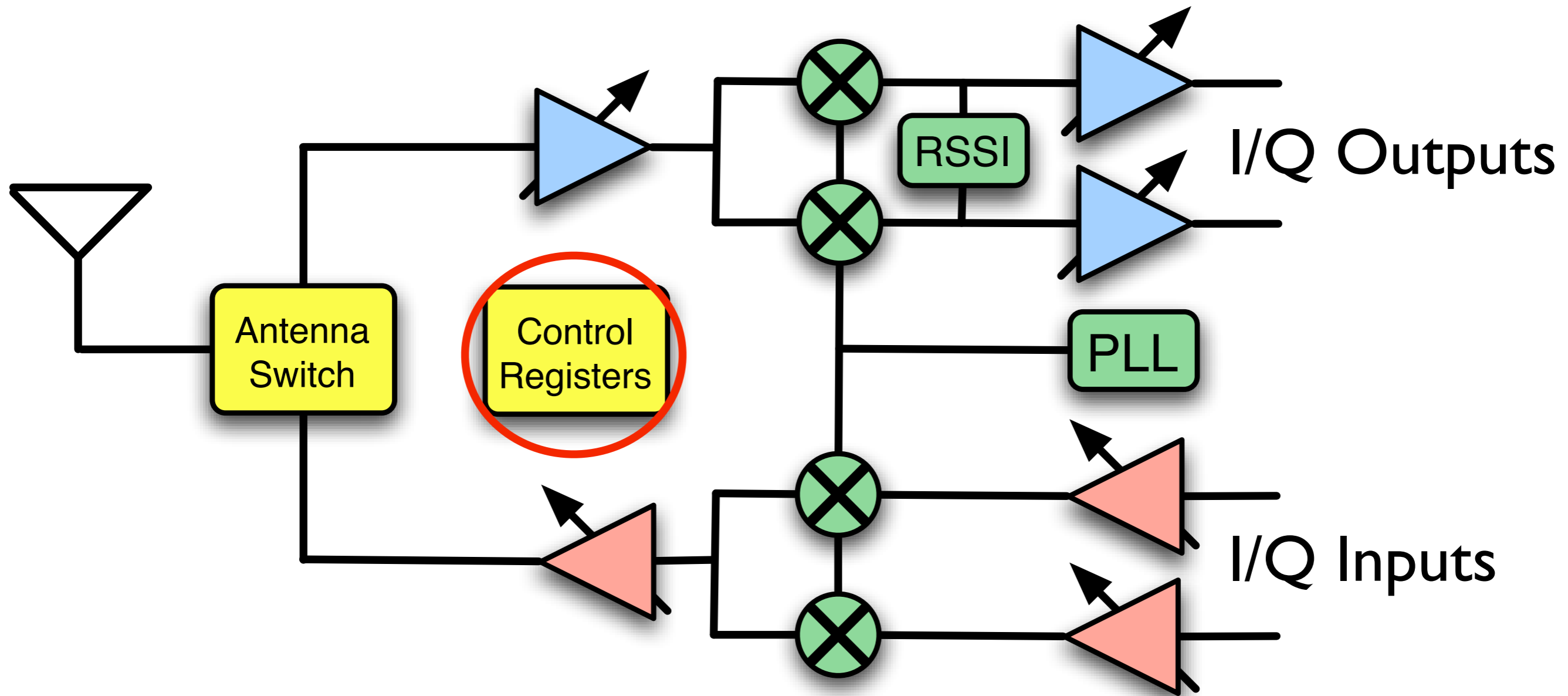


Now They're Our Problem

Radio Transceiver

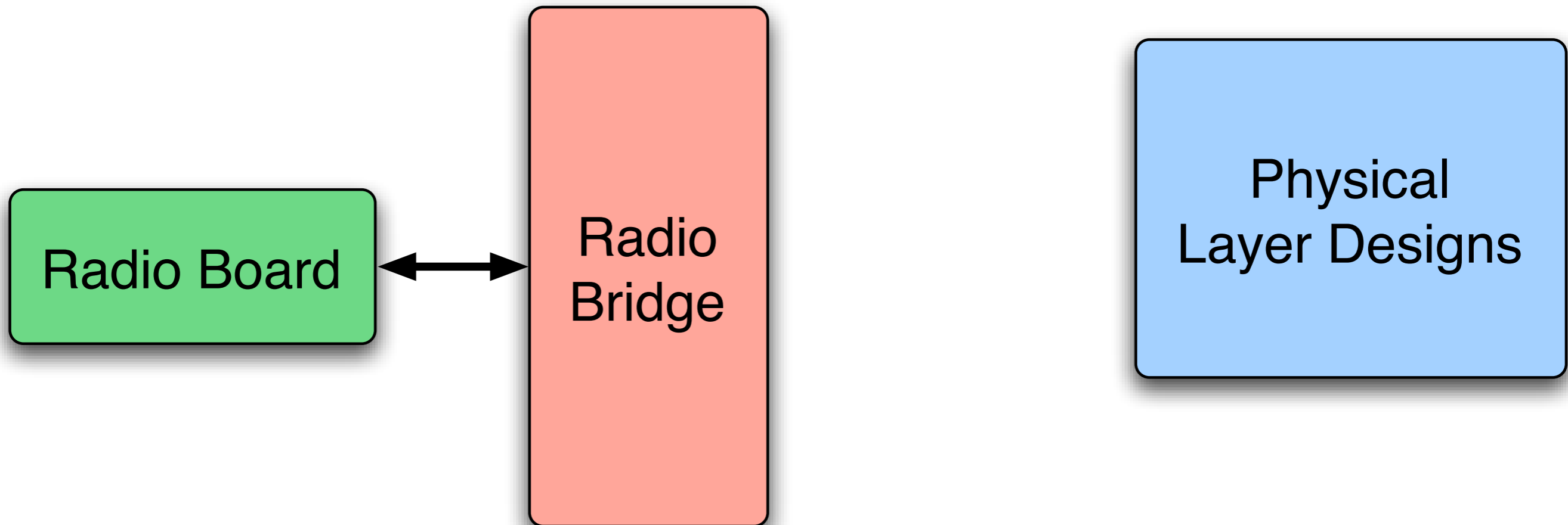


Radio Transceiver

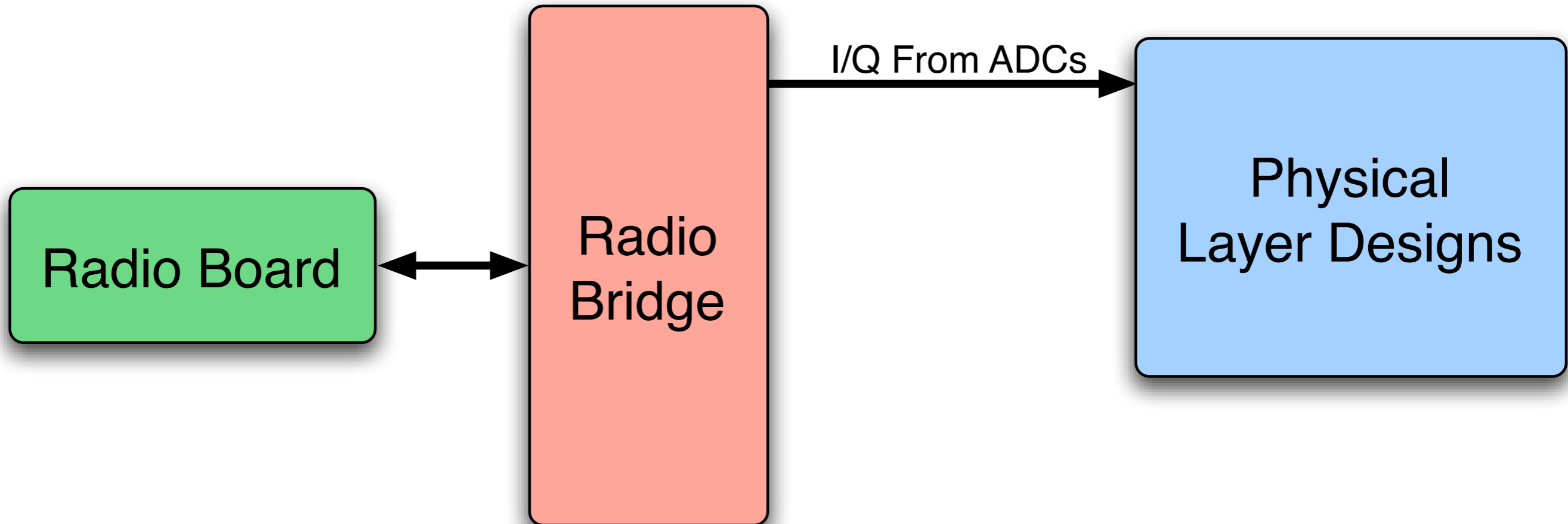


Register Bank
(controlled by SPI interface)

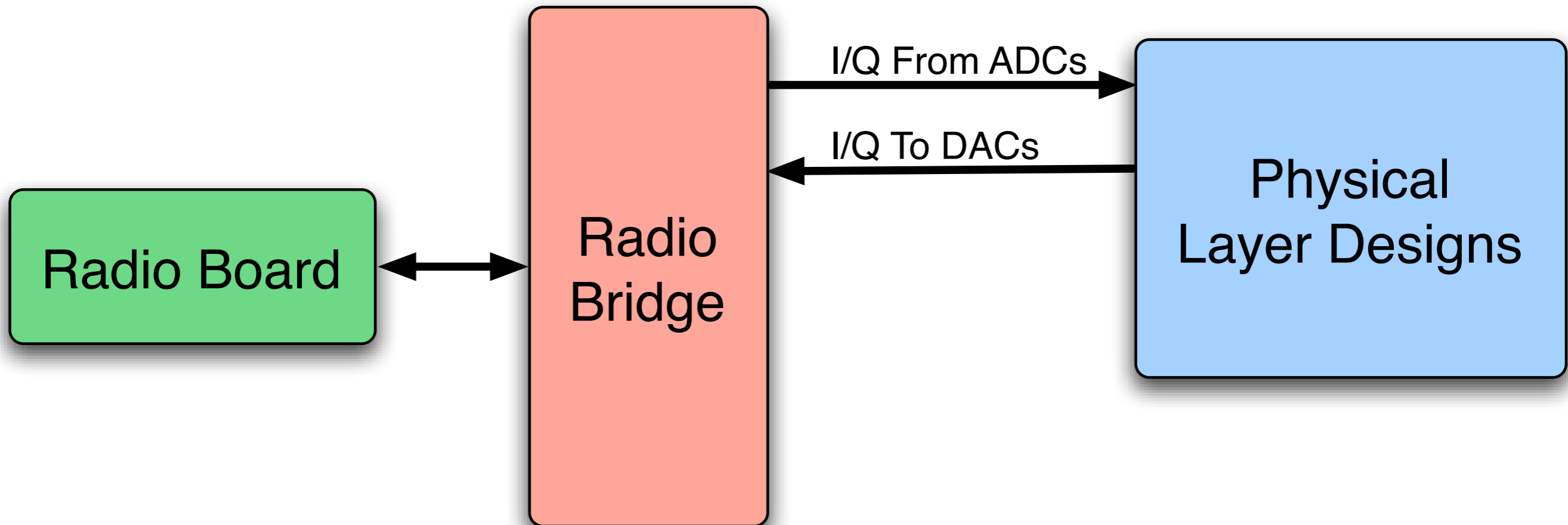
Physical Layer in Hardware



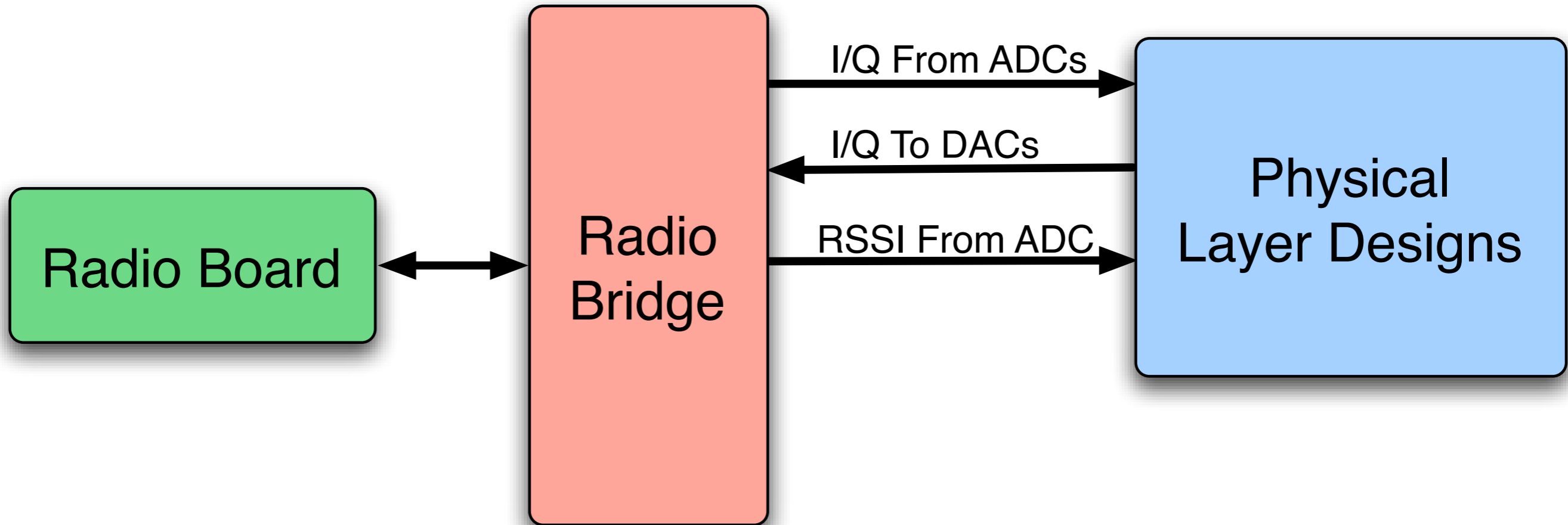
Physical Layer in Hardware



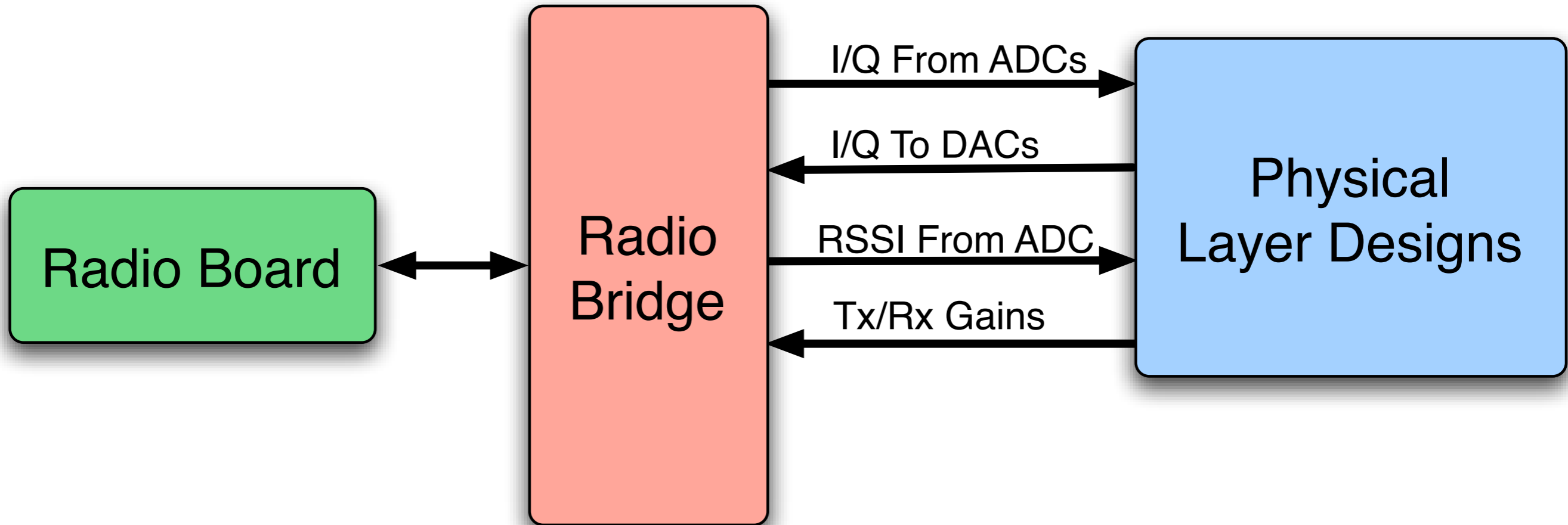
Physical Layer in Hardware



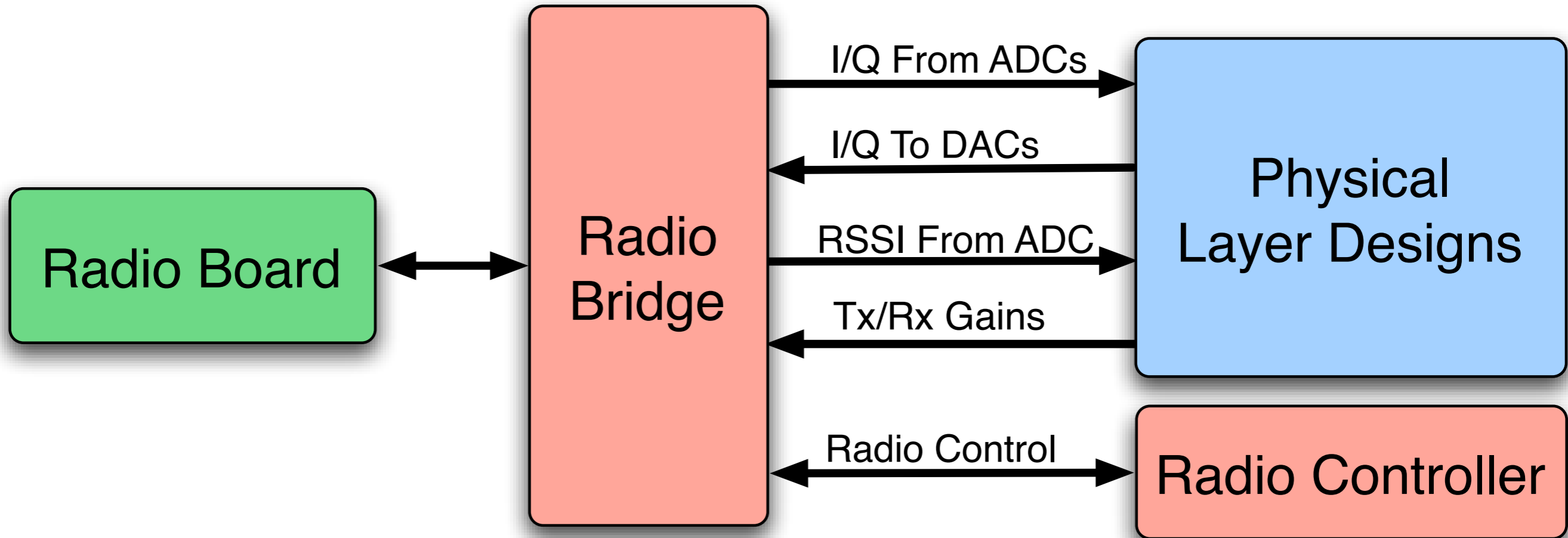
Physical Layer in Hardware



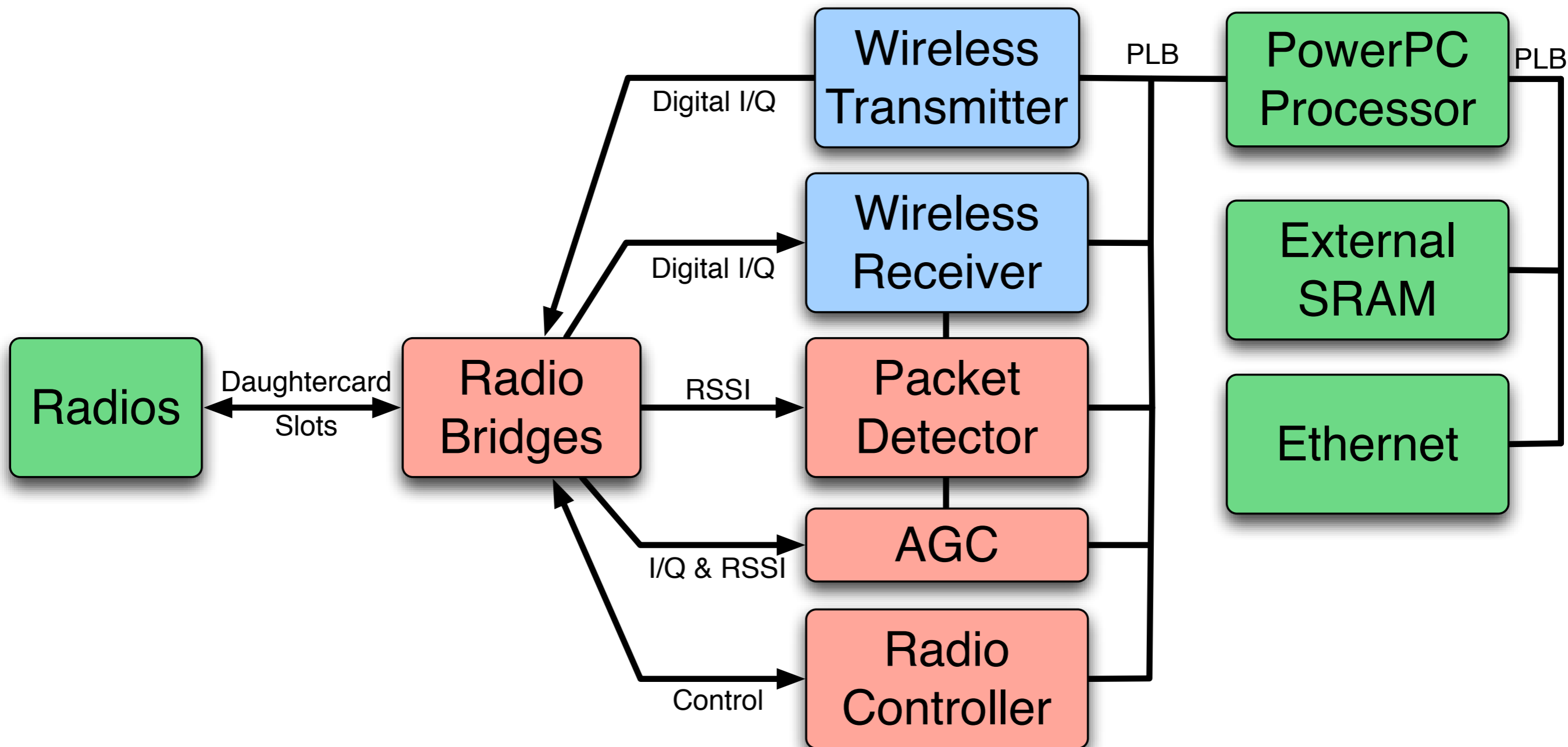
Physical Layer in Hardware



Physical Layer in Hardware

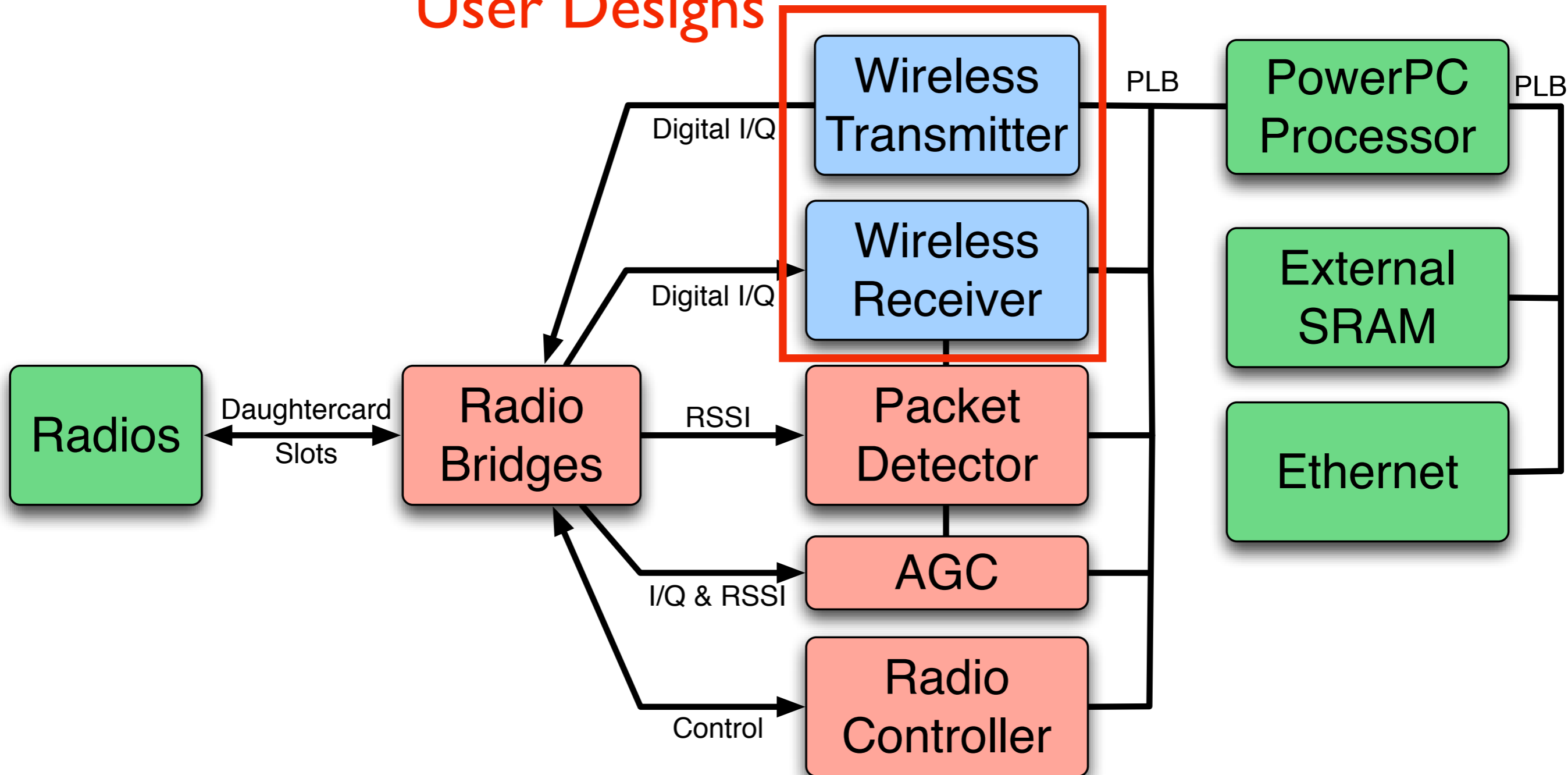


Physical Layer in Hardware

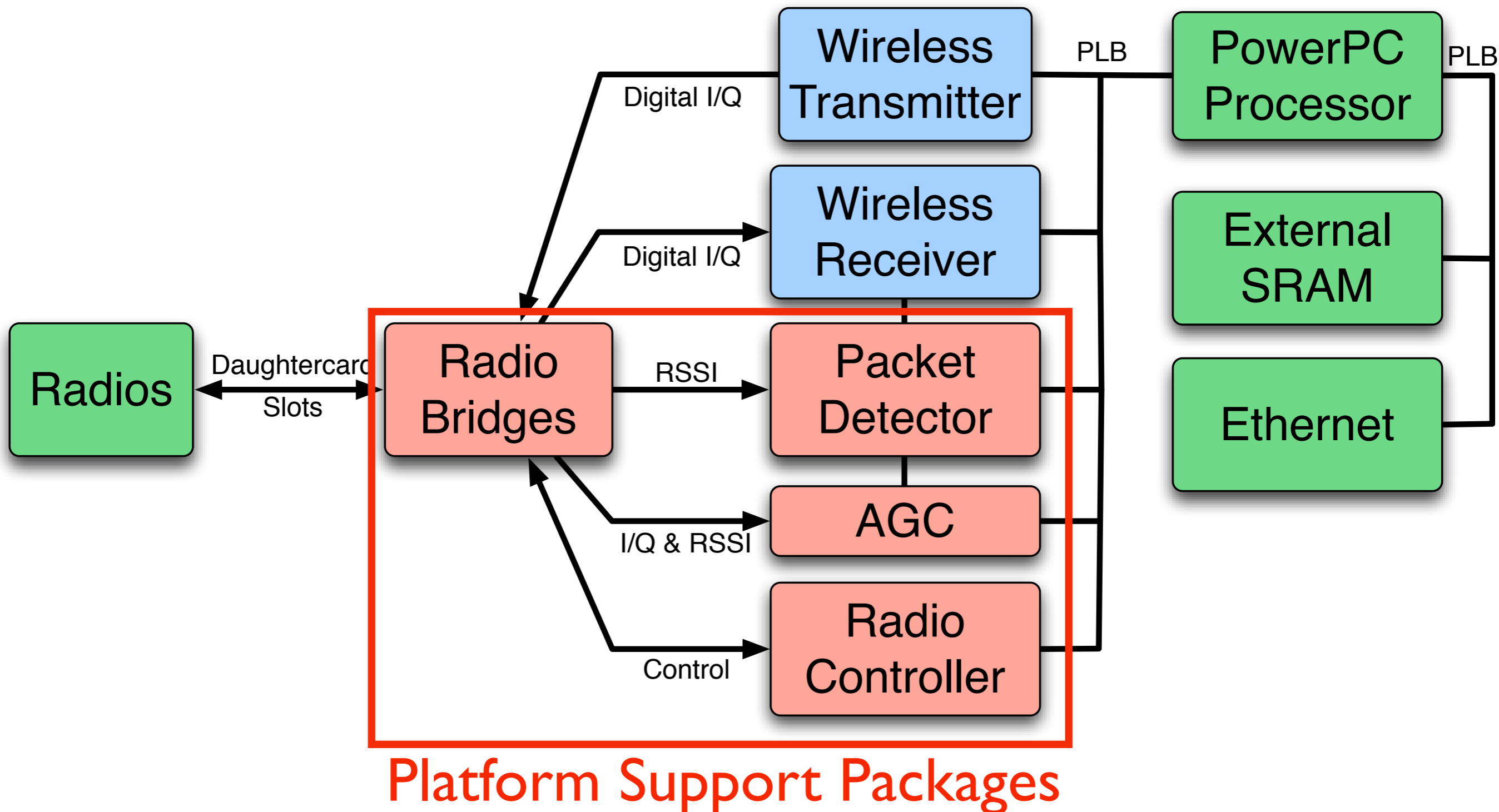


Physical Layer in Hardware

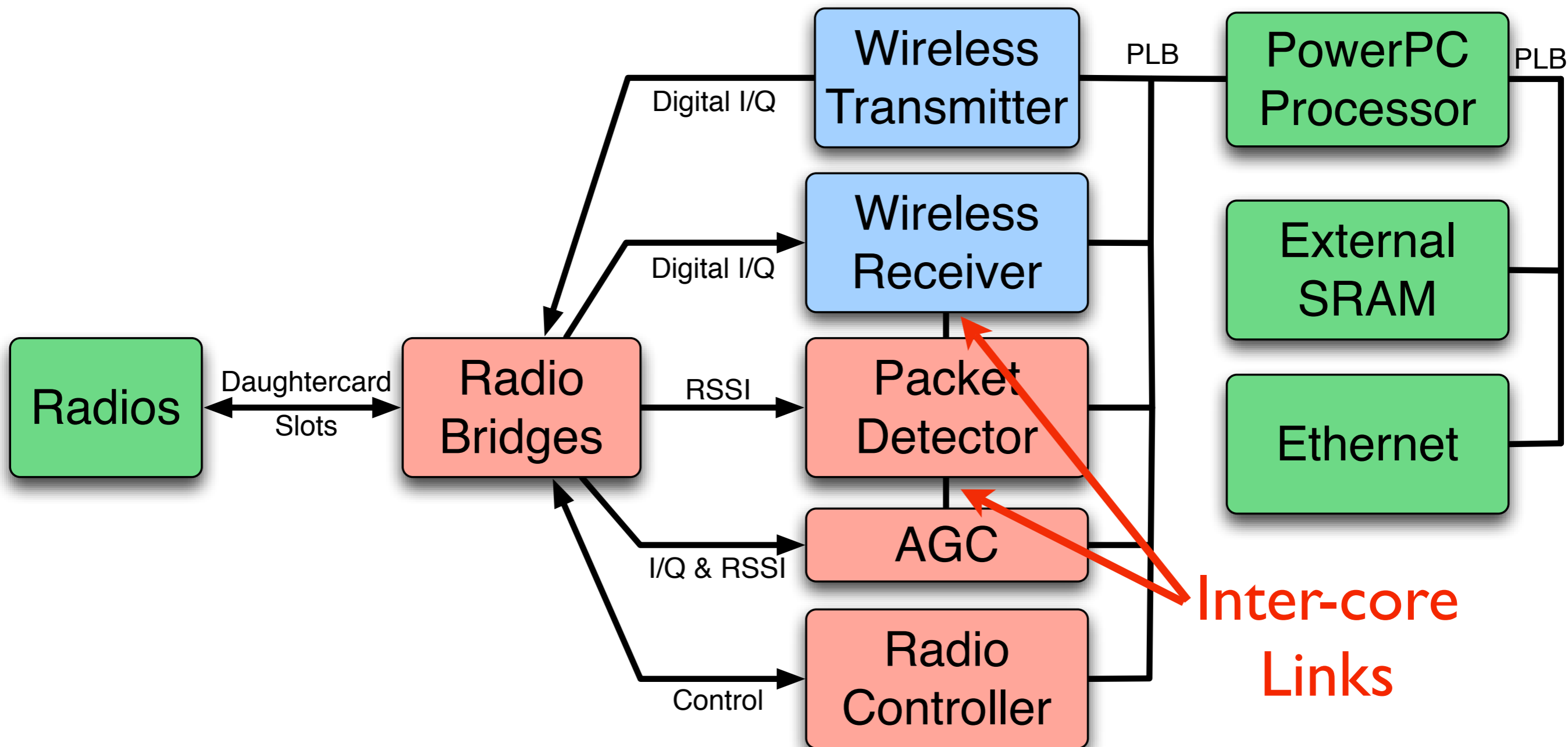
User Designs



Physical Layer in Hardware



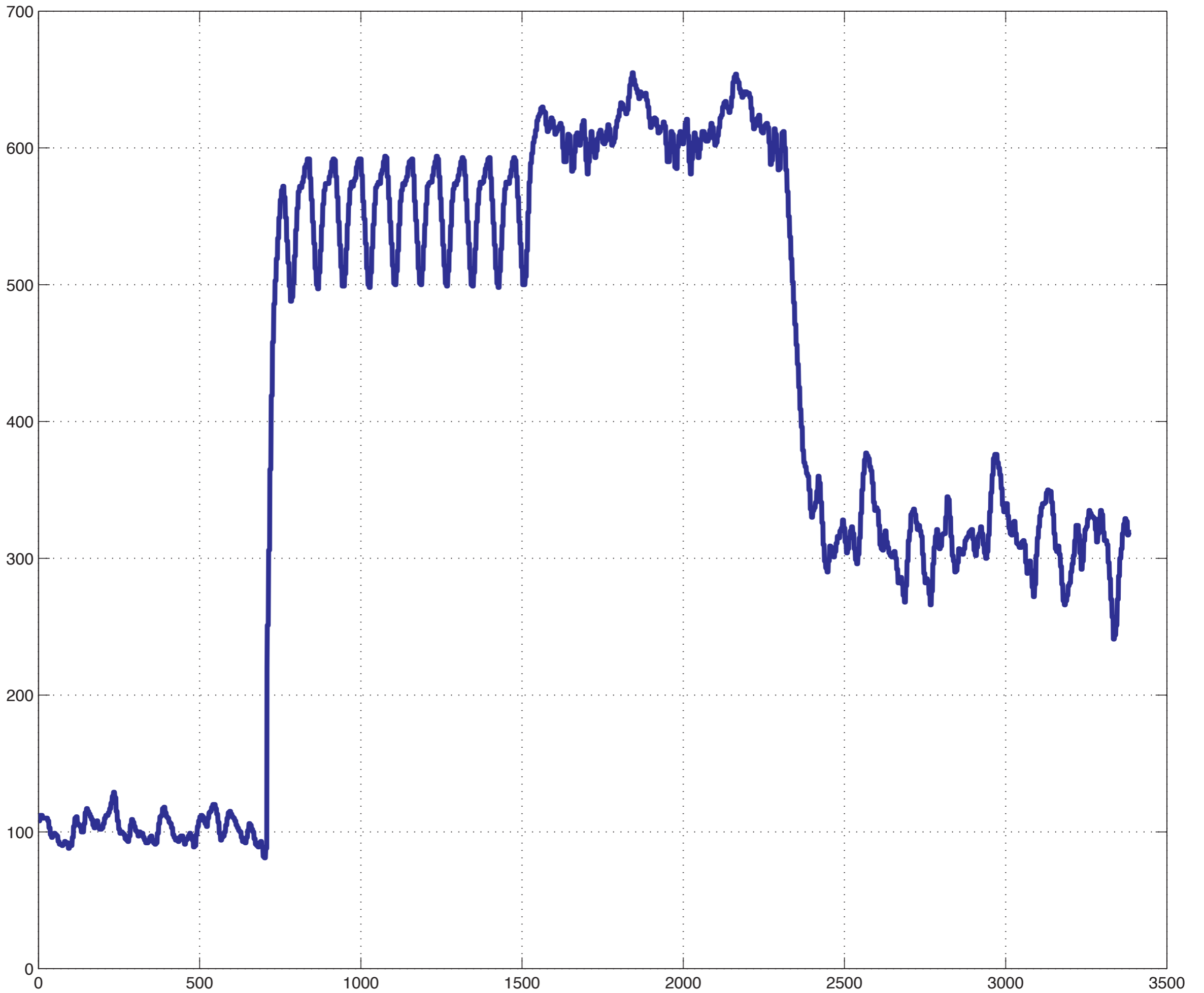
Physical Layer in Hardware



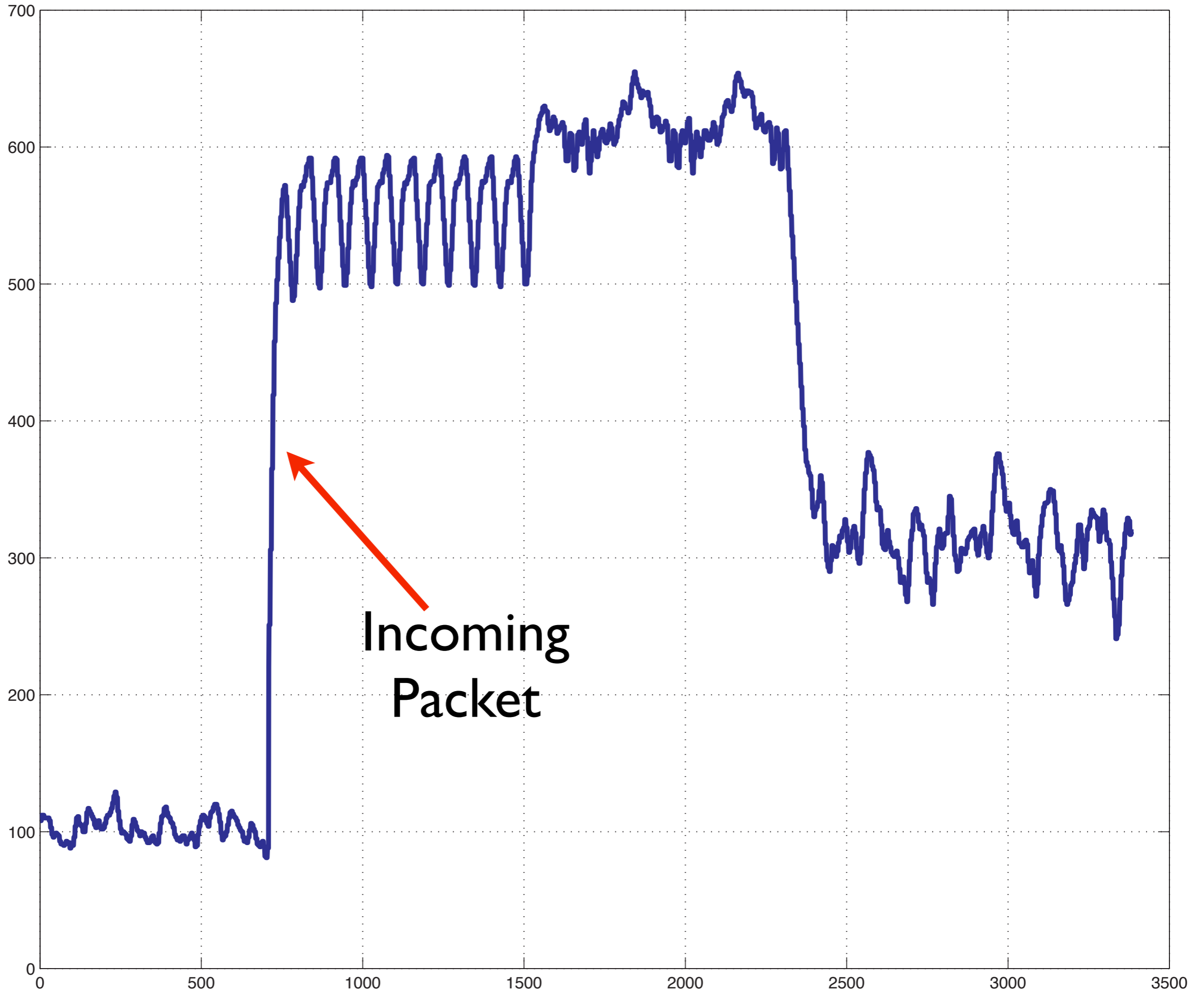
Packet Detection

- Triggers AGC & receiver models
- Detection based only on received energy
 - I/Q saturated and too corrupted
 - Gain adjusted *after* detection
- Detection confirmed/rejected by Rx PHY
 - Requires some data-aided detection
 - Correlates against every packet's preamble

RSSI



RSSI



Incoming Packet

Automatic Gain Control

- Receiver has ≈ 90 dB gain range
 - RF gain of 0, 15 or 30 dB
 - Baseband gain of 0...60 dB
- Amplifiers start max gain with each packet
- AGC reduces gain in first 5 μ s
 - RF gain set by RSSI
 - Baseband gain set by I/Q averages

Radio Controller

- Controller hardware
 - I/O registers & SPI controller
 - One core controls all 4 radios & DACs
- Controller software
 - Full C API for radio board control
 - All radio features controlled by C functions
 - Simple functions required
 - Advanced functions optional

Radio Controller API

WarpRadio_v1_Reset()

WarpRadio_v1_TxEnable()

WarpRadio_v1_SetCenterFreq2GHz()

WarpRadio_v1_BaseBandTxGain()

WarpRadio_v1_TxVGAGainControl()

WarpRadio_v1_24AmpEnable()

WarpRadio_RxEnable()

WarpRadio_RxLNAGainControl()

WarpRadio_RxVGAGainControl()

WarpRadio_RxLpfCornFreqCoarseAdj()

Radio Controller API

Full API online:

http://warp.rice.edu/WARP_API

WarpRadio_v1_Reset()

WarpRadio_v1_TxEnable()

WarpRadio_v1_SetCenterFreq2GHz()

WarpRadio_v1_BaseBandTxGain()

WarpRadio_v1_TxVGAGainControl()

WarpRadio_v1_24AmpEnable()

WarpRadio_RxEnable()

WarpRadio_RxLNAGainControl()

WarpRadio_RxVGAGainControl()

WarpRadio_RxLpfCornFreqCoarseAdj()

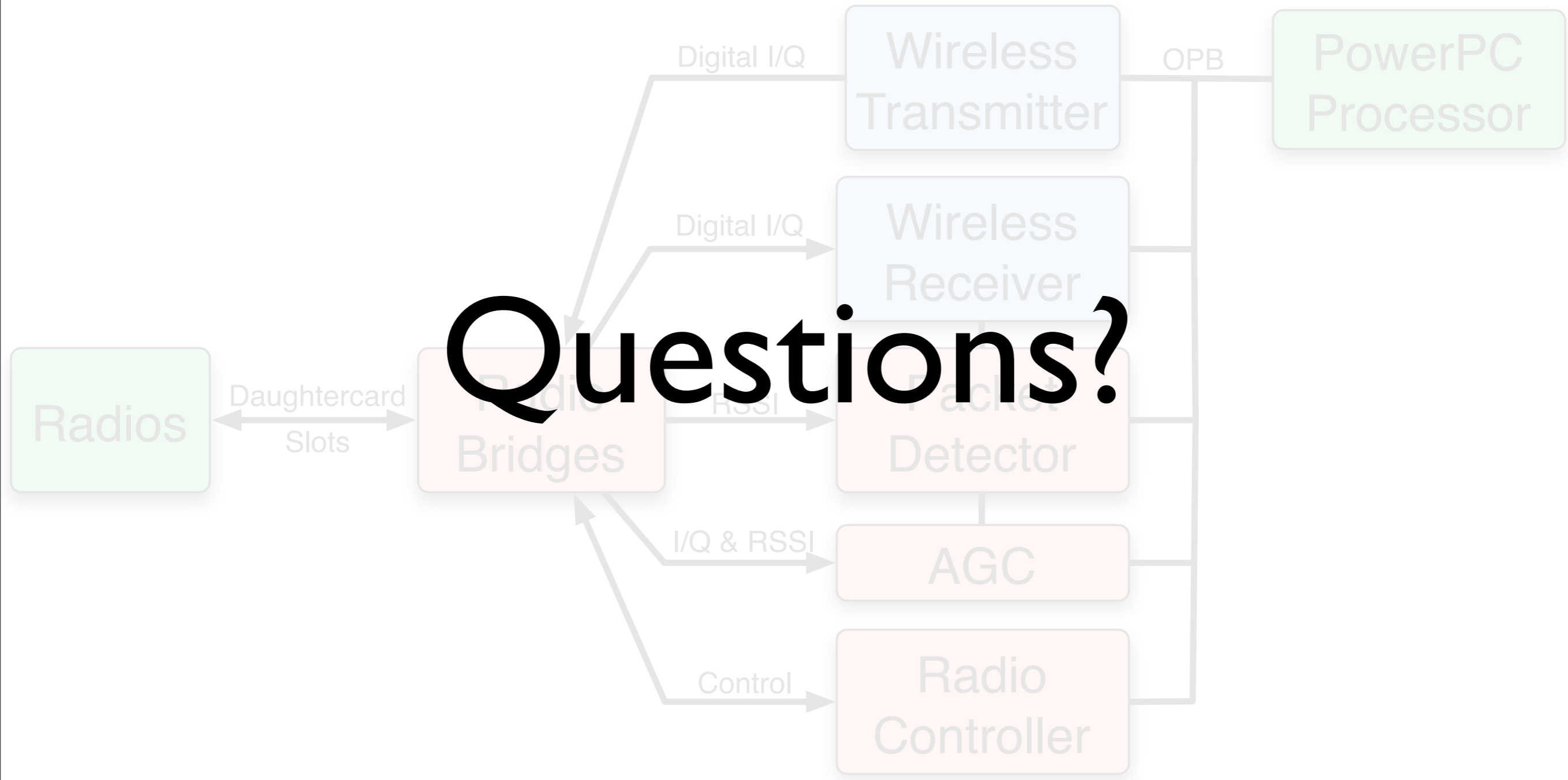
Radio Bridge

- Ties user designs to radio hardware
 - Ports for user signals (ADC, DAC, gains)
 - Ports for radio controller I/O
- Users instantiate one bridge per radio board
- All constraints & most links are automatic
- Custom Verilog peripheral

PHY Design Review

- Build & verify PHY in FPGA design tool
 - System Generator is a good choice
 - Make sure everything works in simulation
- Generate simple Tx/Rx peripherals
 - “Cheating” is good at first
- Hook up your core in the EDK
 - Use the radio bridges & controller
- Generate the platform & test it in hardware

Questions?



Lab 3: Simple Transmitter

- Build a sinusoid generator in Sysgen
- Convert the model to a PLB peripheral
- Connect the Tx core to the radio bridge
- Test the model at RF