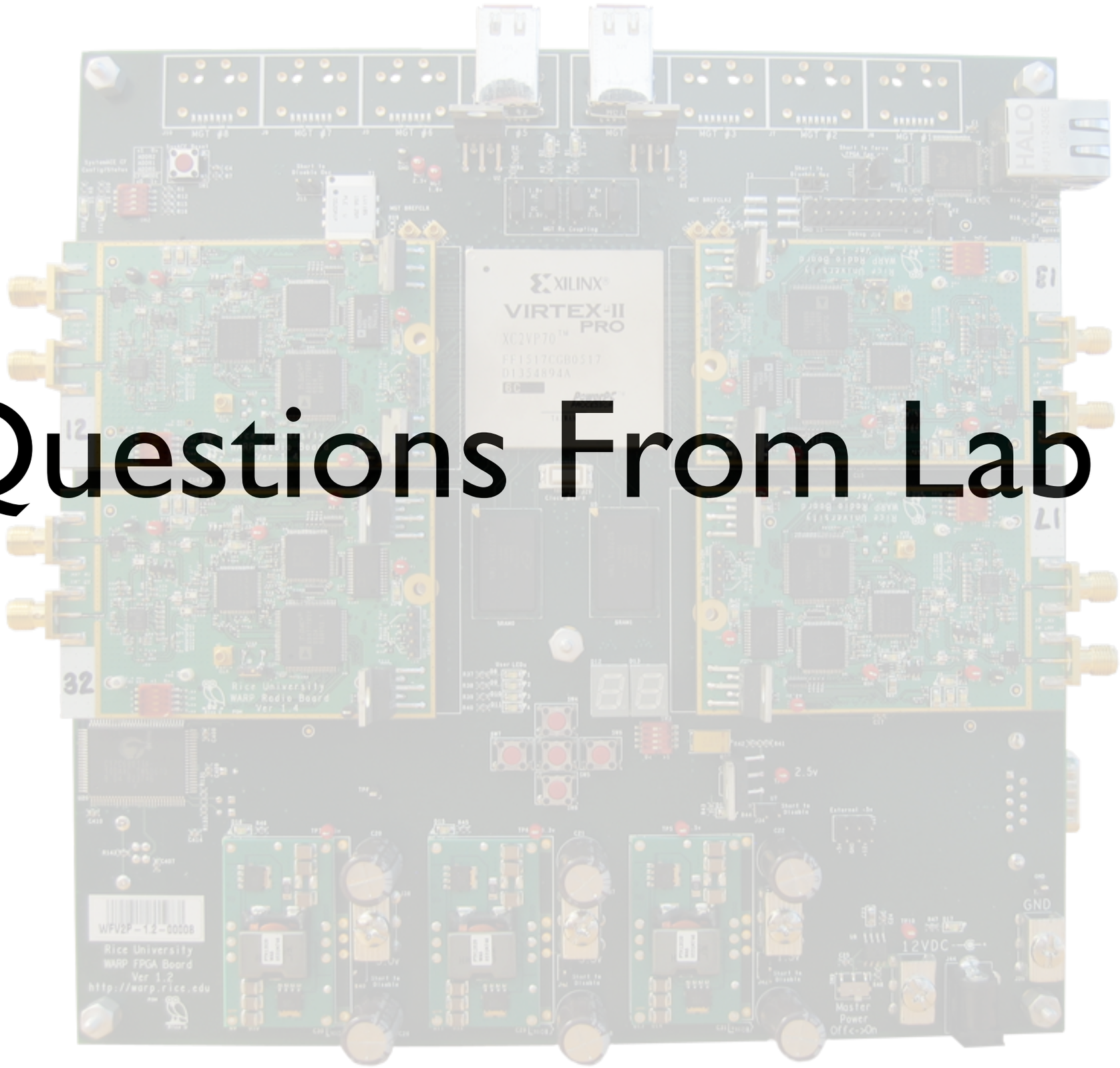
WARP: Physical Layer Design

Patrick Murphy
Rice University

WARP Workshop
March 22, 2007

warp.rice.edu

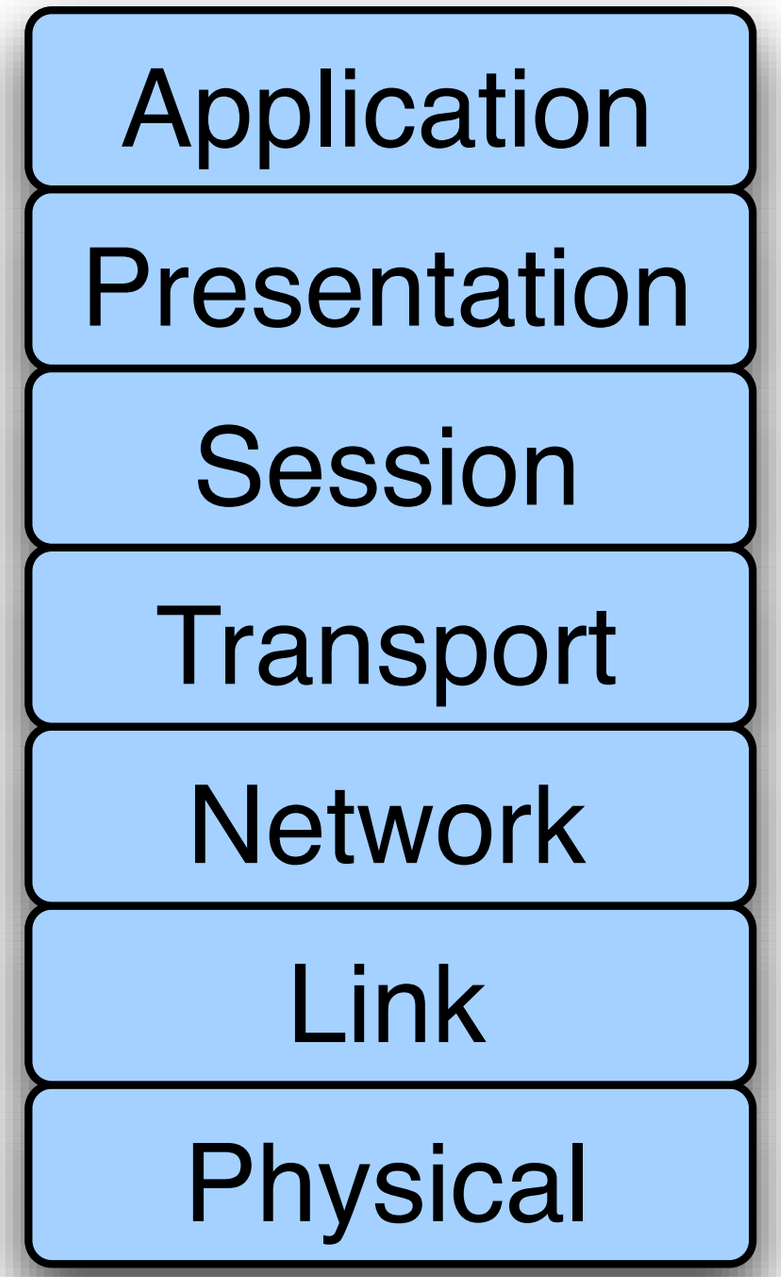


Questions From Lab1?

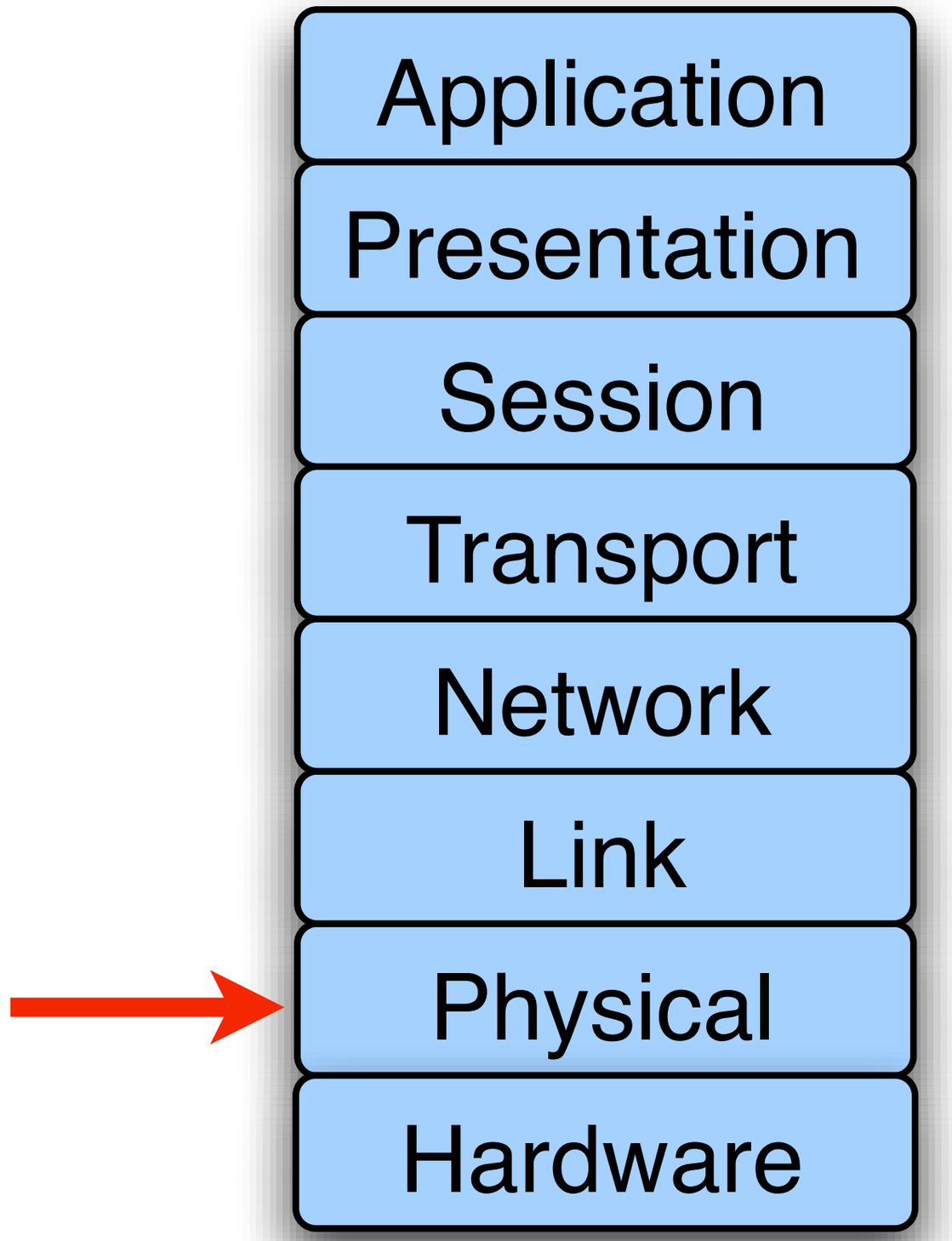
PHY Design - Outline

- Physical layer basics
- Key PHY design decisions
- PHY Example: OFDM Transmitter
- Hooking it all up in hardware
  - Radio considerations
  - Using platform support packages
- Lab 2: Building a transmitter

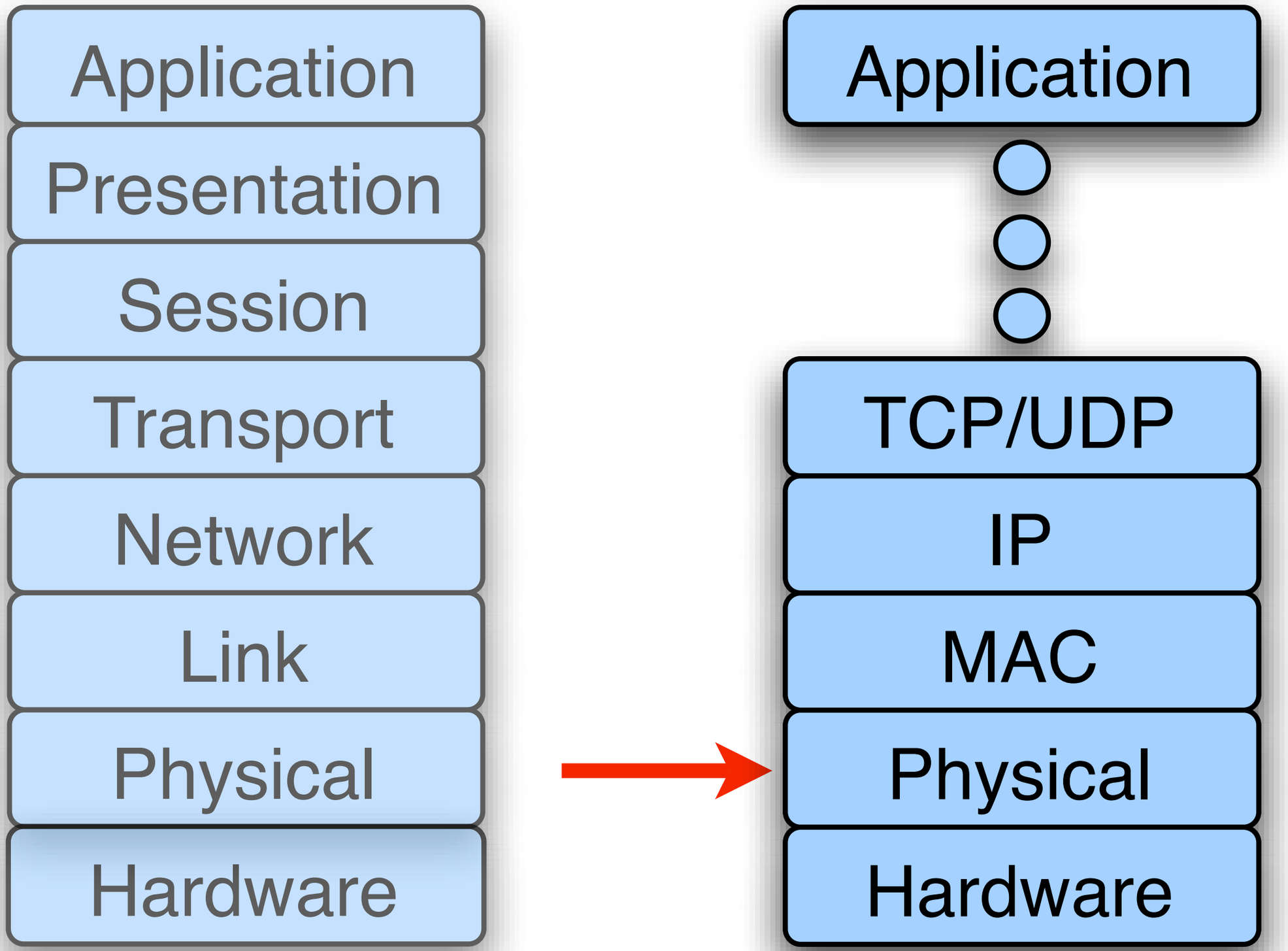
Physical Layer Basics



Physical Layer Basics

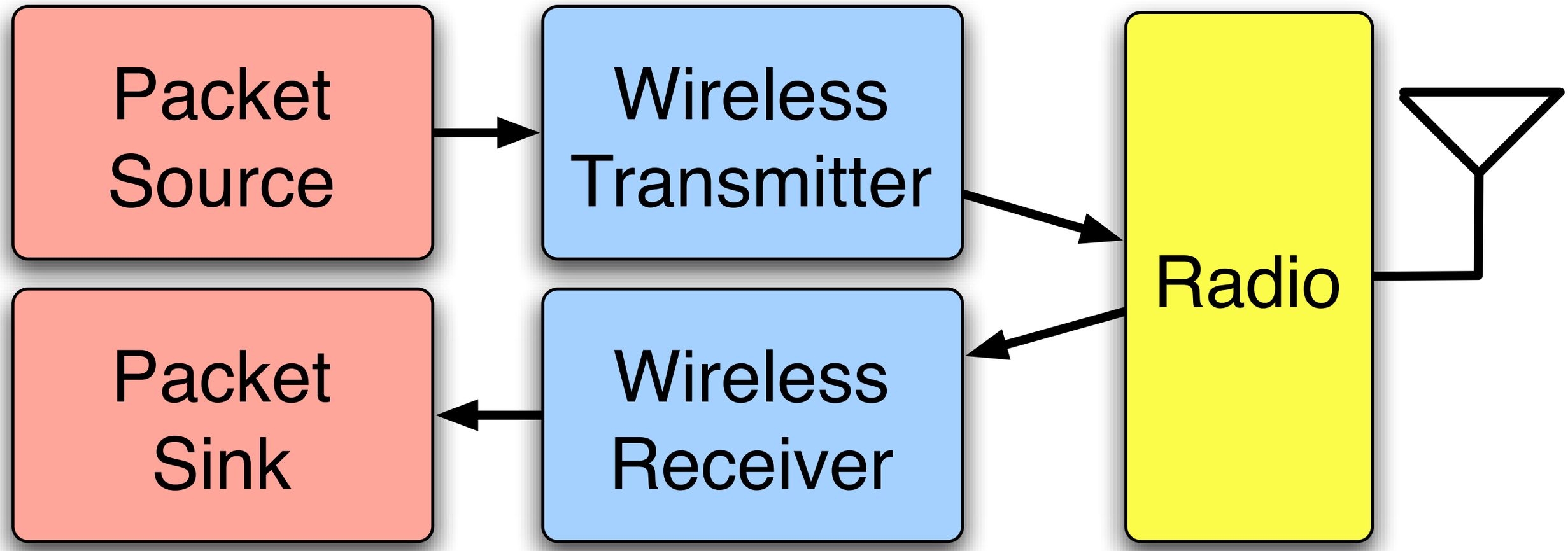


Physical Layer Basics



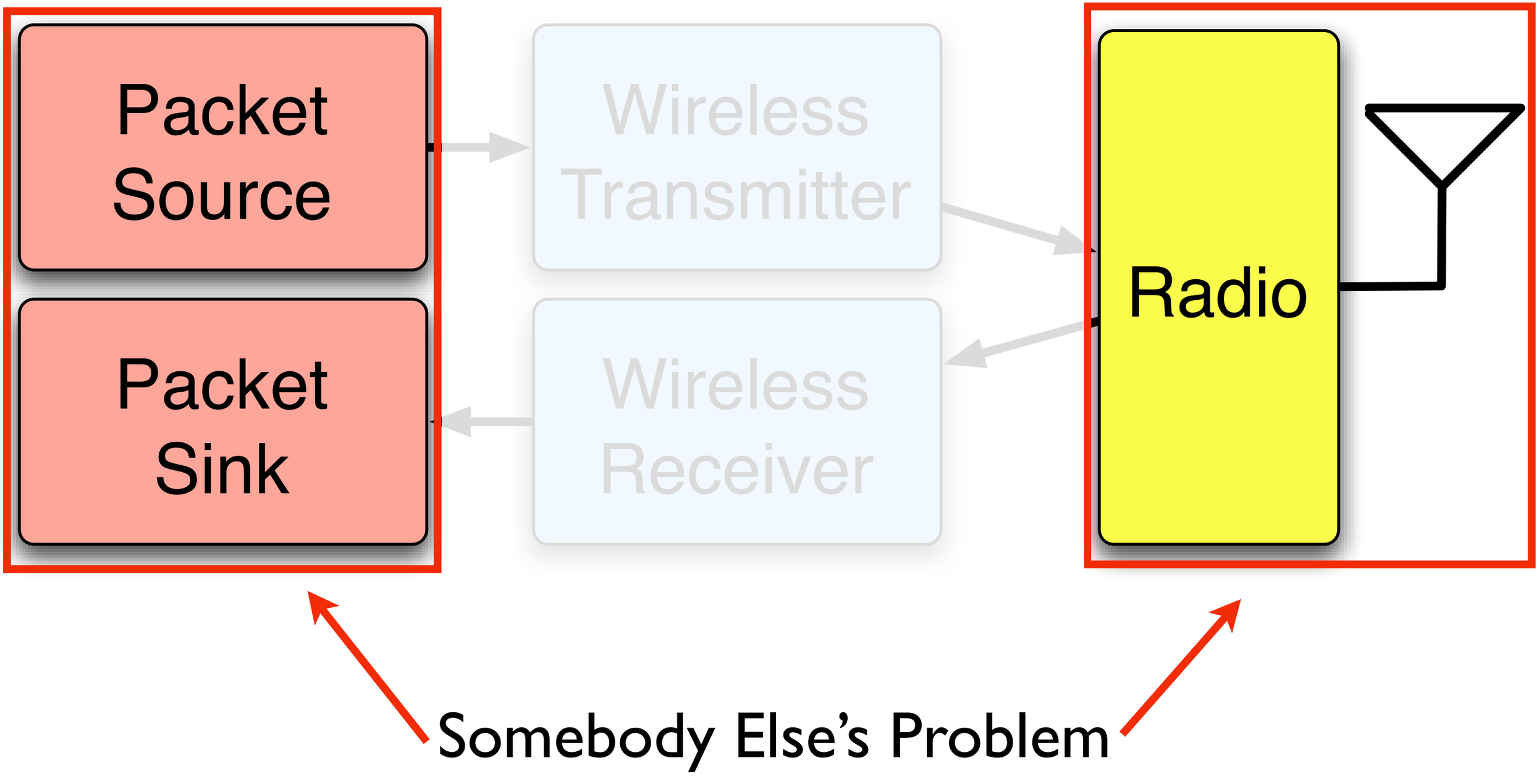
Physical Layer Basics

Simple Wireless Node



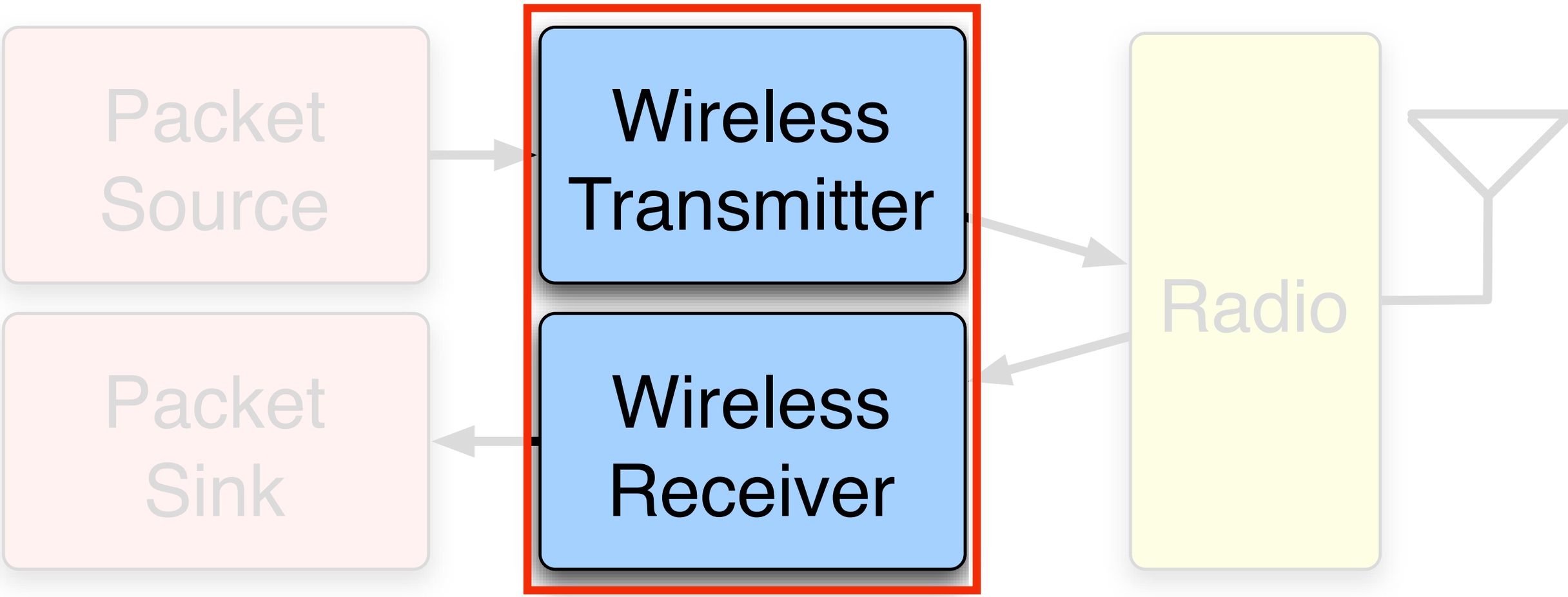
Physical Layer Basics

Simple Wireless Node



Physical Layer Basics

Simple Wireless Node



Physical Layer

PHY Design Decisions

PHY Design Decisions

- Packet-based or streaming?

PHY Design Decisions

- Packet-based or streaming?
- Wideband or narrowband?
  - Pipelined vs. iterative
  - Think clock cycles per channel sample

PHY Design Decisions

- Packet-based or streaming?
- Wideband or narrowband?
  - Pipelined vs. iterative
  - Think clock cycles per channel sample
- Complete synchronization or “cheating”?
  - Independent vs. common reference clocks
  - Packet detection vs. shared timing signals
  - Real channel vs. coax cable

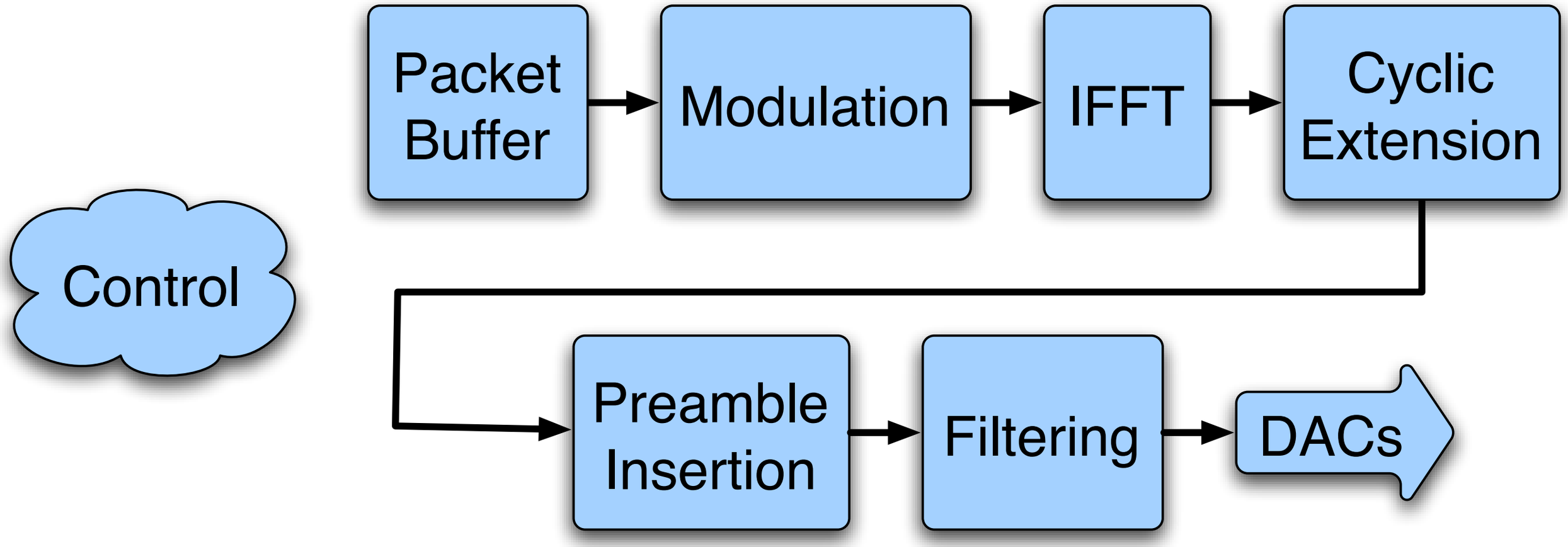
PHY Design Decisions

- Packet-based or streaming?
- Wideband or narrowband?
  - Pipelined vs. iterative
  - Think clock cycles per channel sample
- Complete synchronization or “cheating”?
  - Independent vs. common reference clocks
  - Packet detection vs. shared timing signals
  - Real channel vs. coax cable
- Runtime vs. hard-coded parameters

PHY Example: OFDM

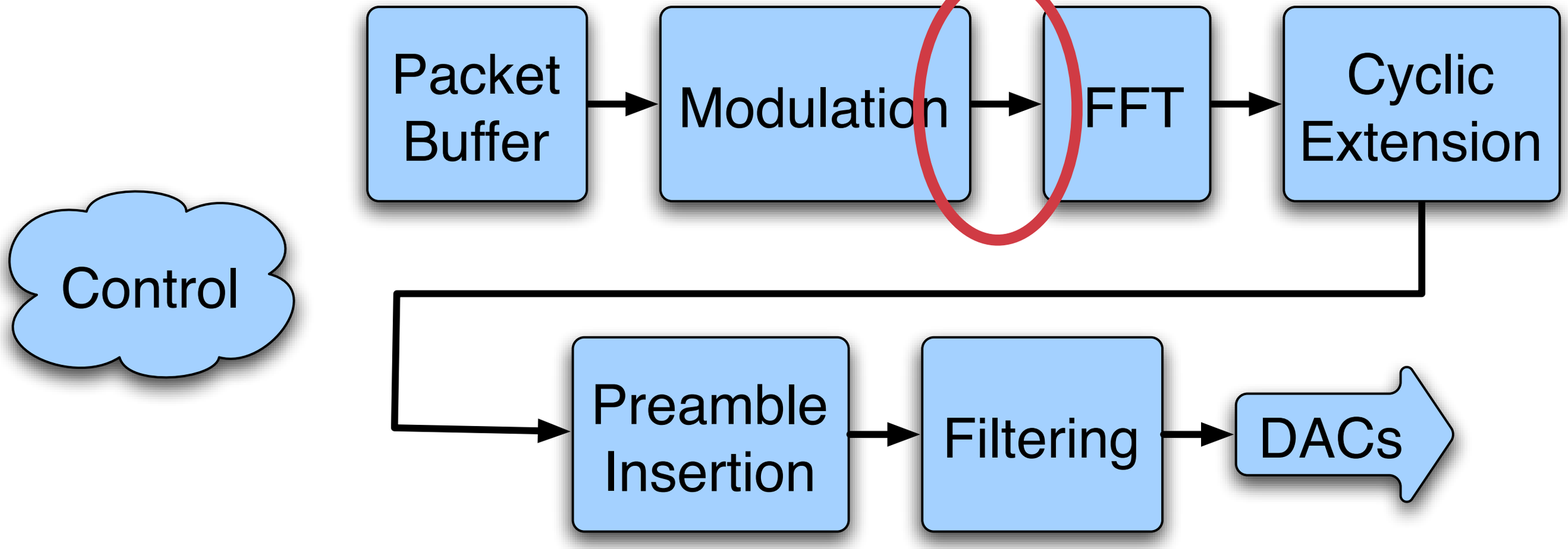
- Designed for wireless networking
  - Modeled on 802.11a (but not compliant)
- Packet-based OFDM transceiver
  - Packets source/sink in PowerPC code
- Wideband, real-time design
  - 5 cycles per sample
  - 10 MHz bandwidth at 50 MHz clock
- Full synchronization for standalone operation
- Implemented entirely in System Generator

PHY Example: OFDM Tx

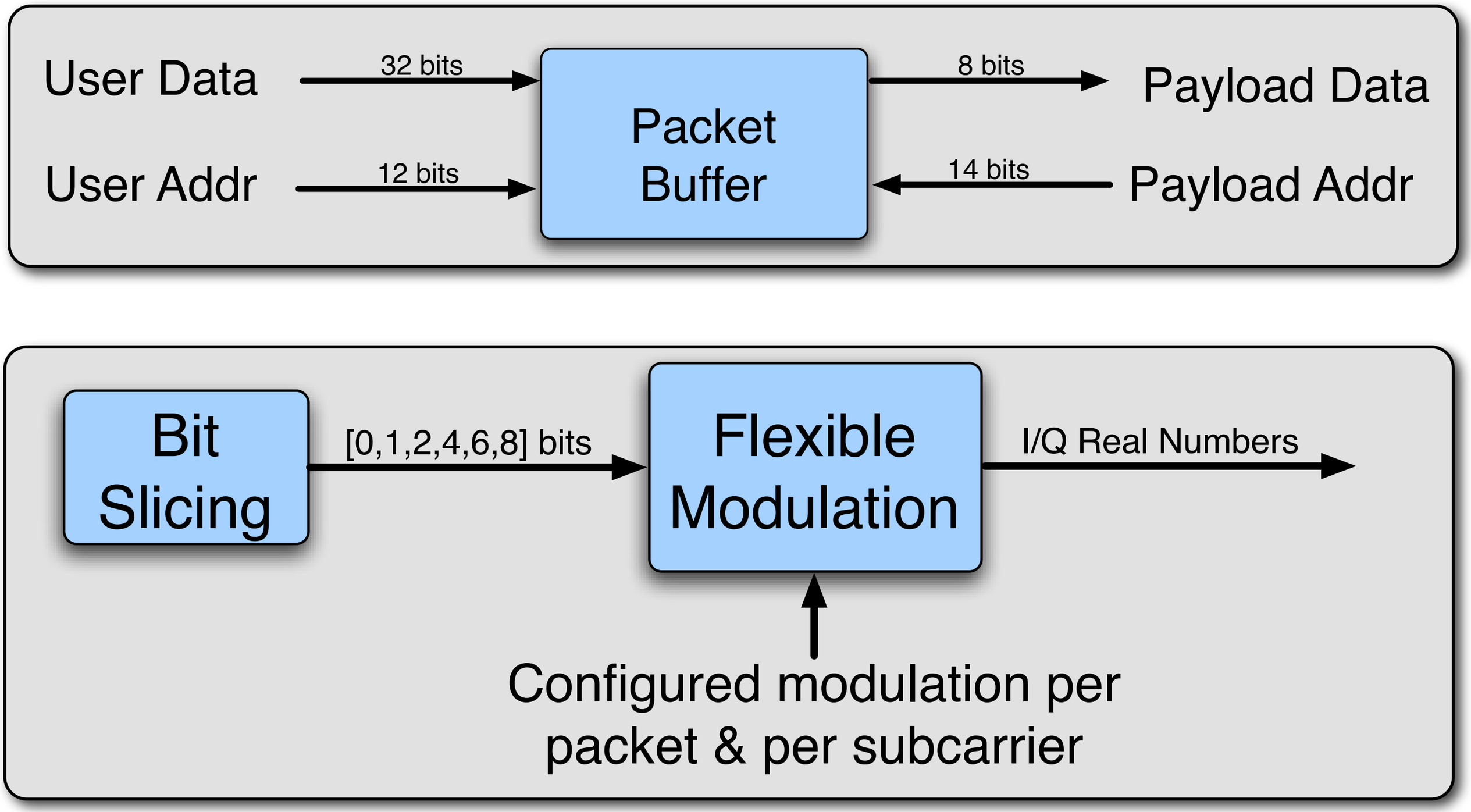


PHY Example: OFDM Tx

No Serial/Parallel Conversion!

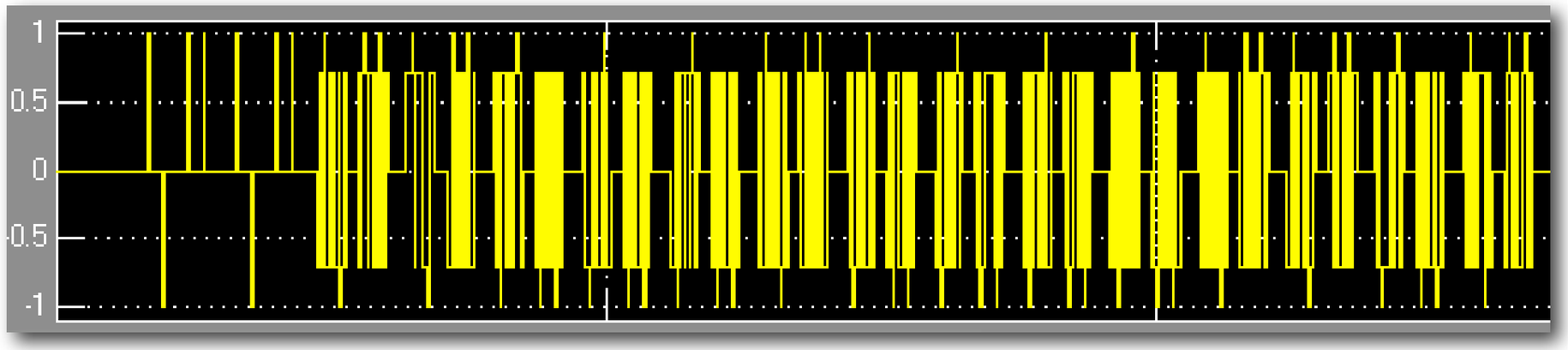


PHY Example: OFDM Tx



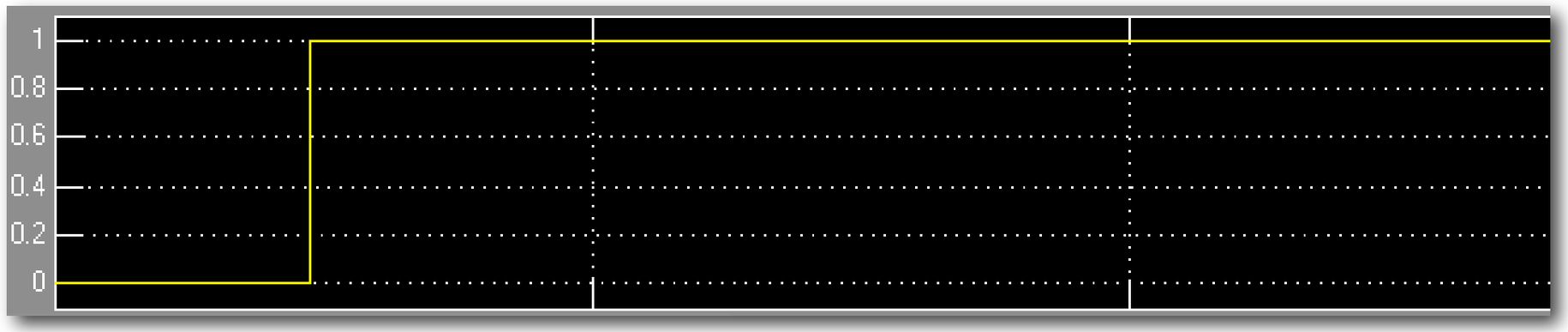
PHY Example: OFDM Tx

Modulator Output

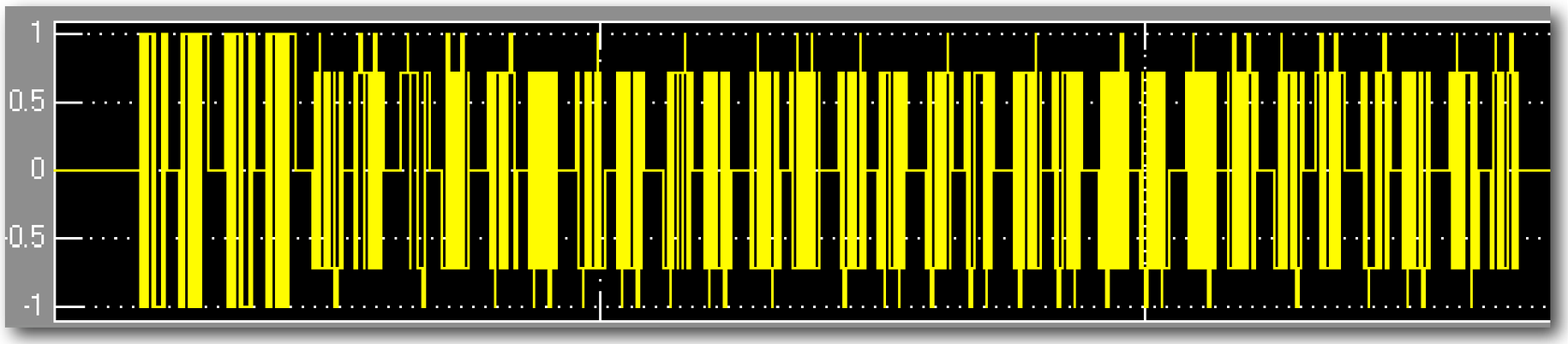


Stored Training Sequence

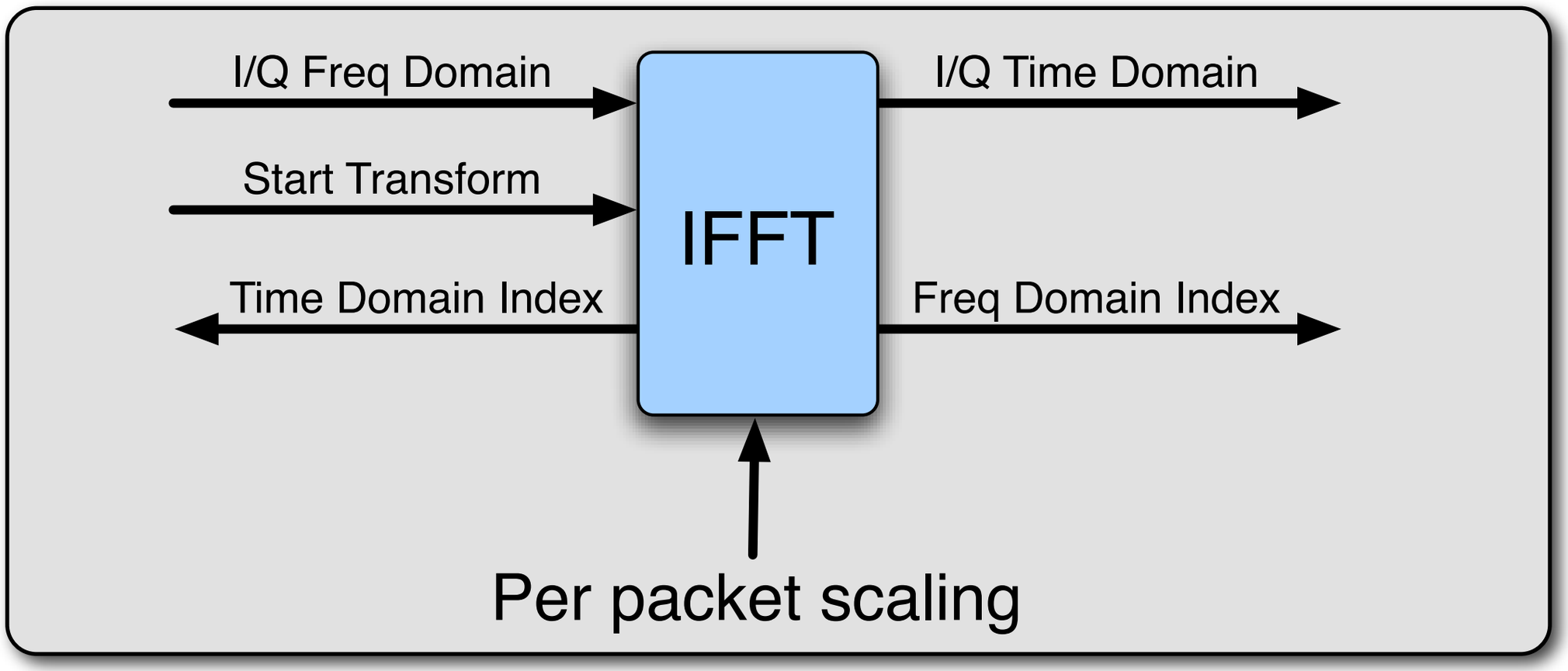
Source Mux Select



Input IFFT



PHY Example: OFDM Tx



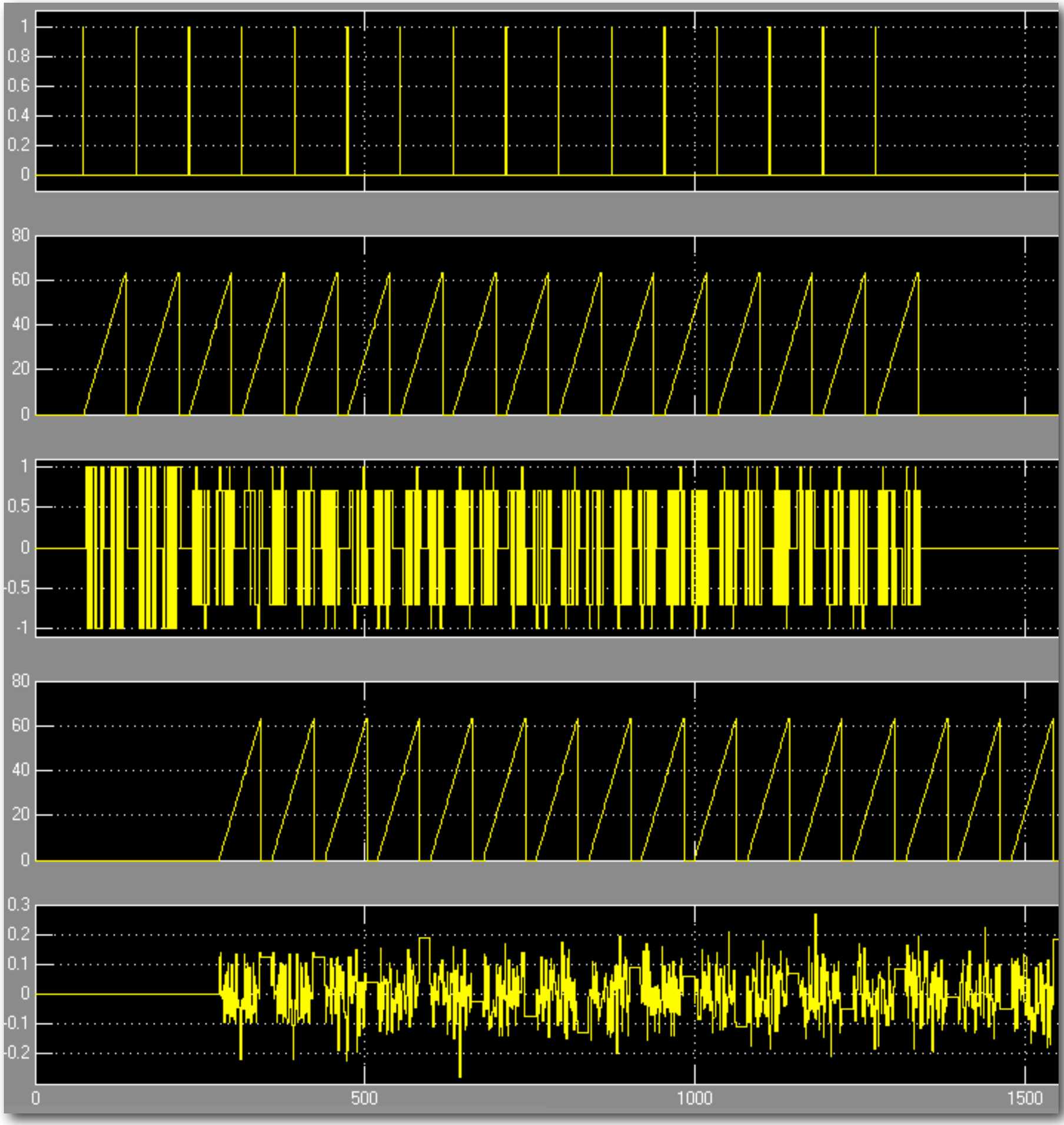
IFFT Start

Input Index

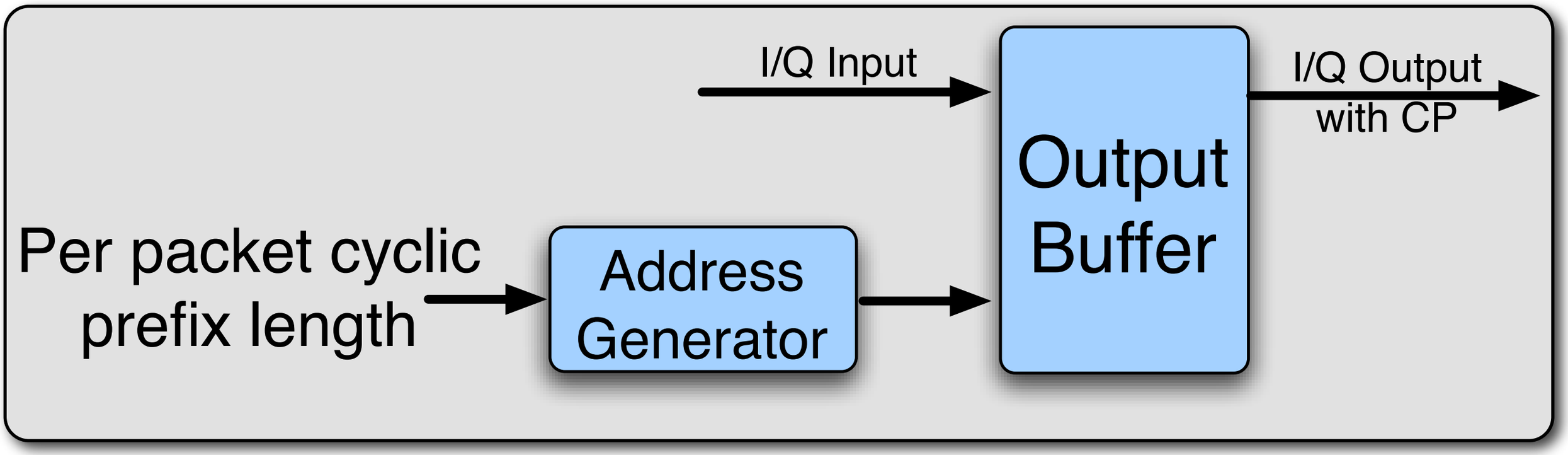
Input Data

Output Index

Output Data

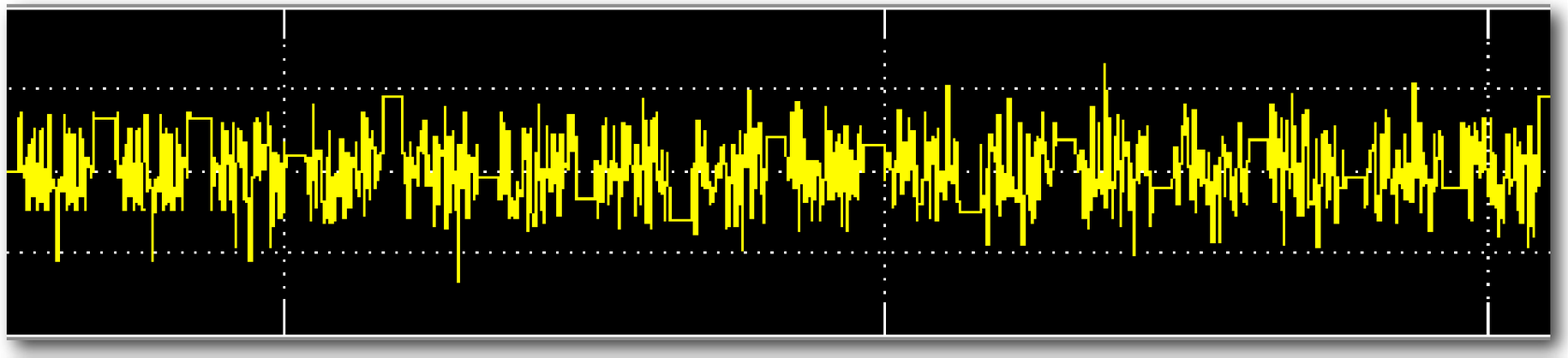


PHY Example: OFDM Tx

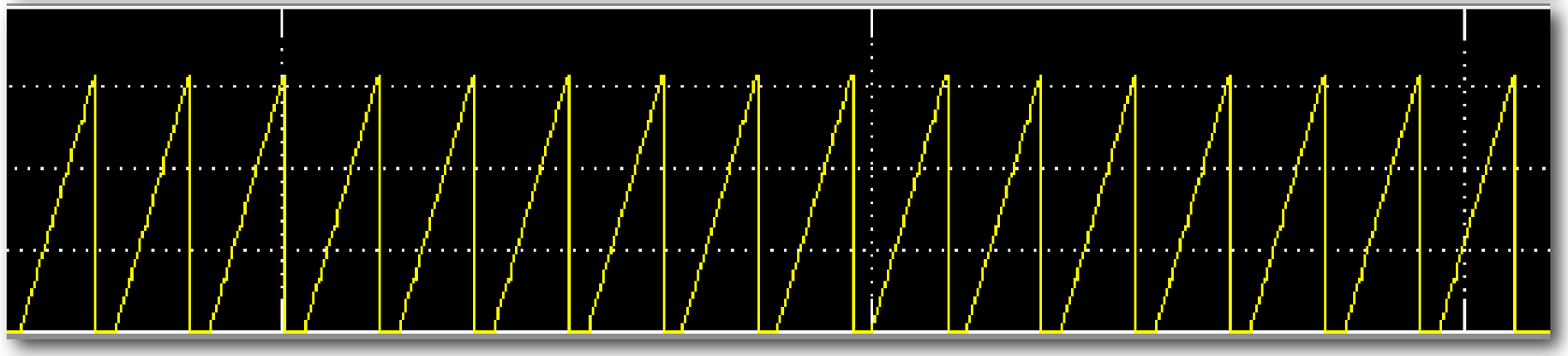


PHY Example: OFDM Tx

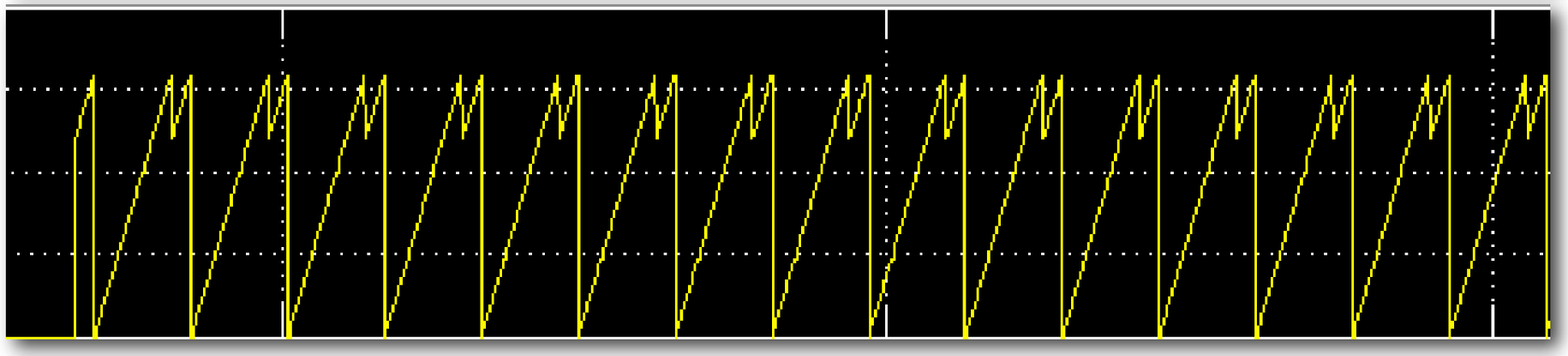
IFFT Output



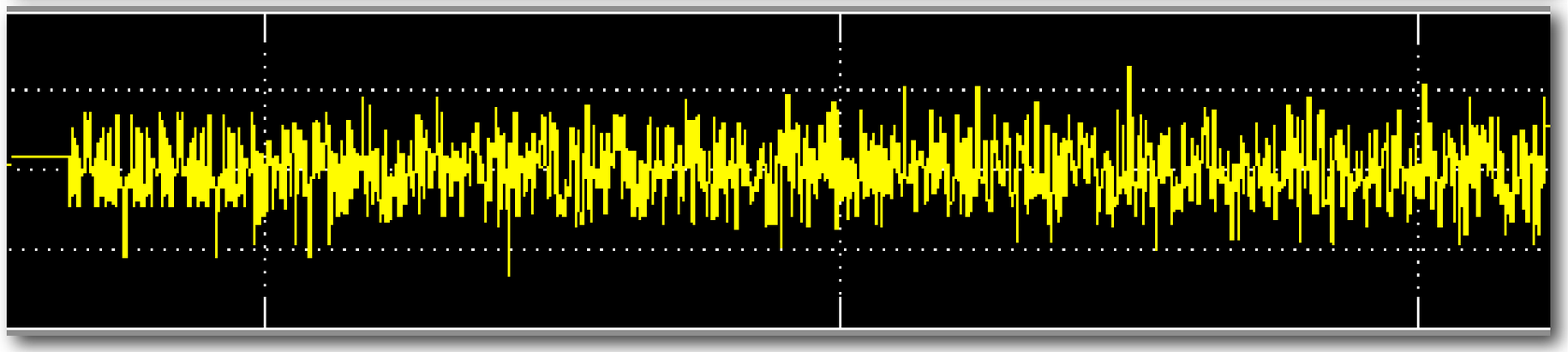
RAM Write Address



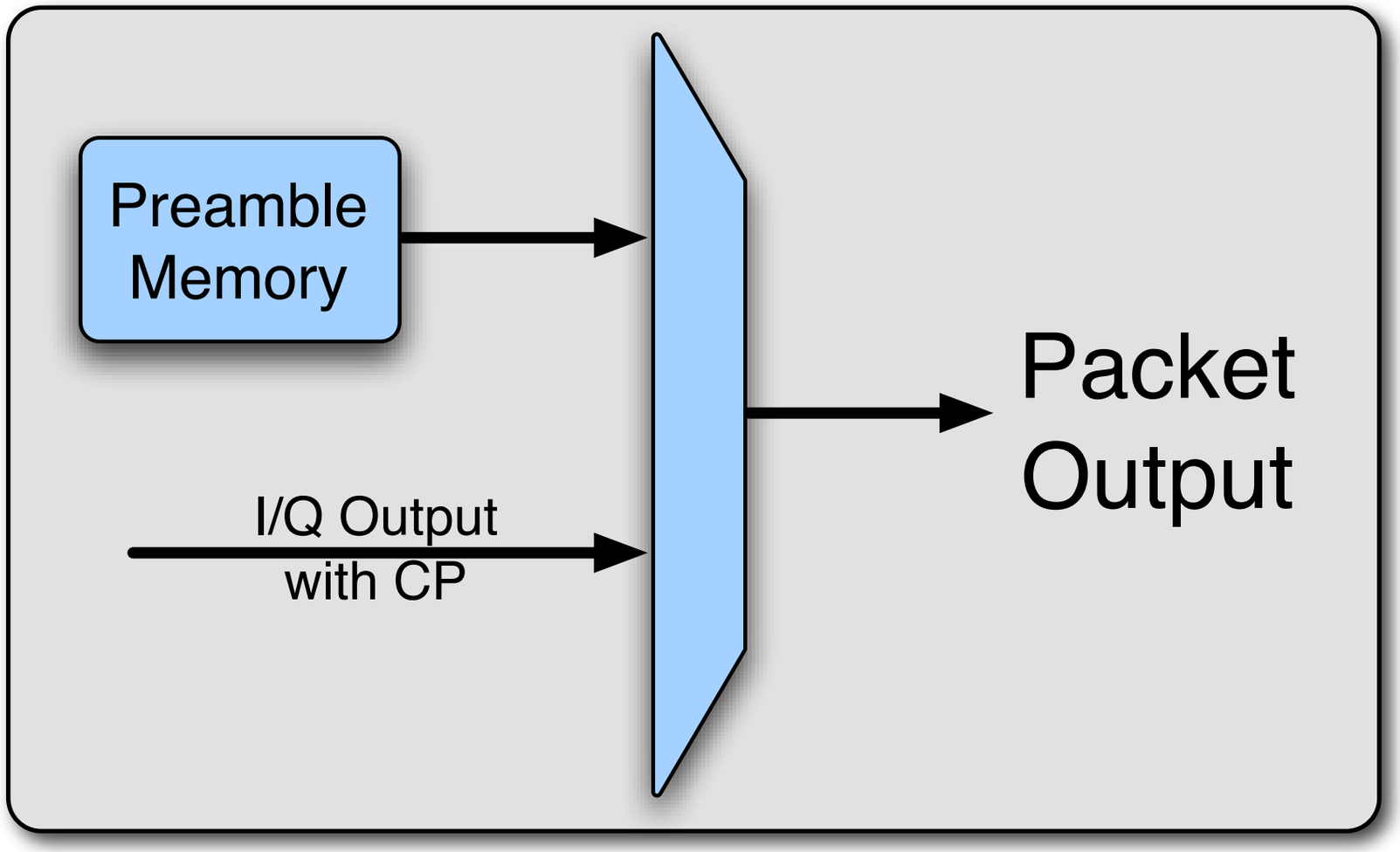
RAM Read Address



Cyclically Extended



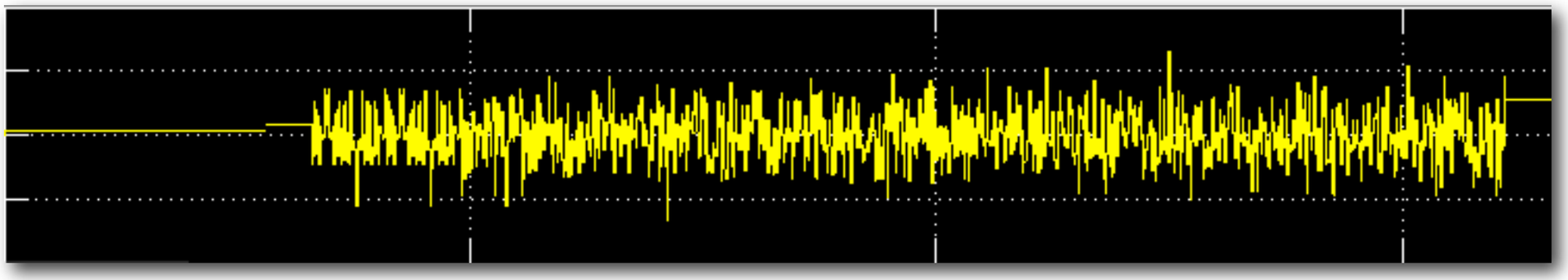
PHY Example: OFDM Tx



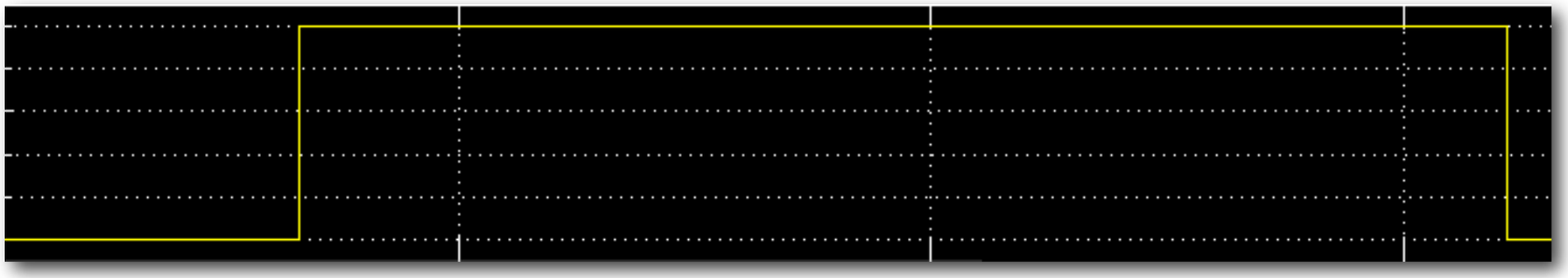
PHY Example: OFDM Tx

Stored Preamble

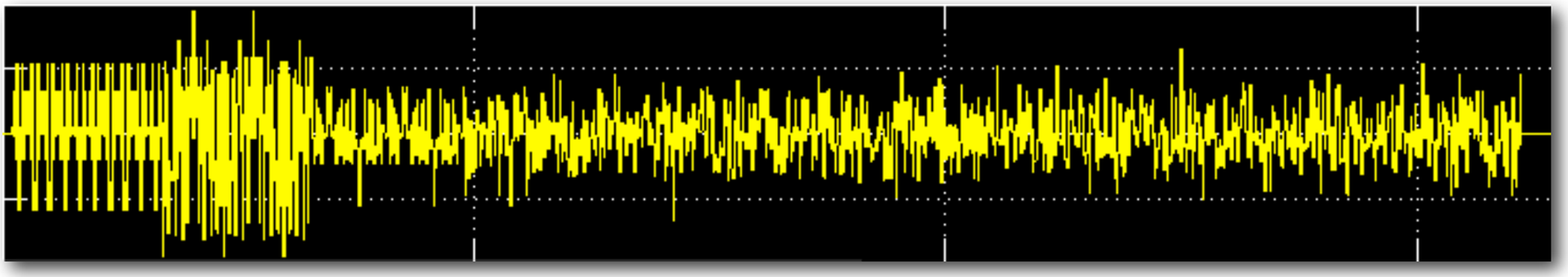
OFDM Output



Output Mux Selection

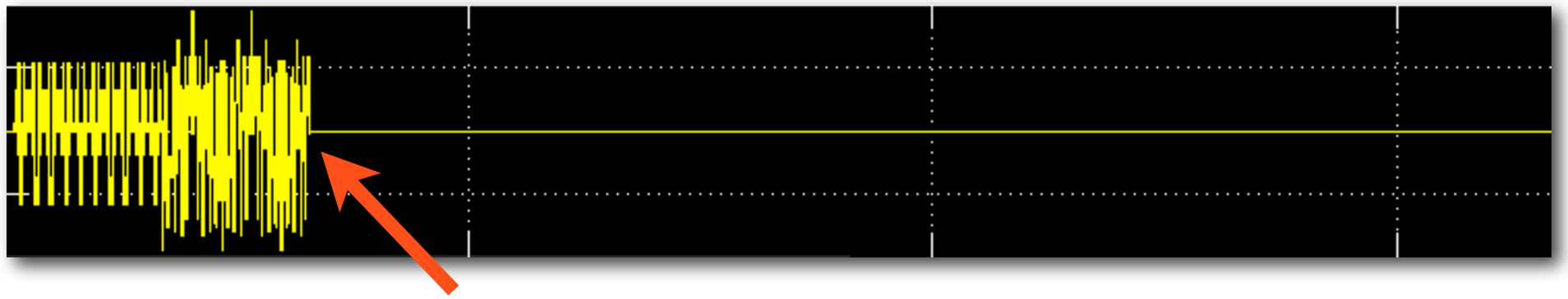


Final Output to DACs

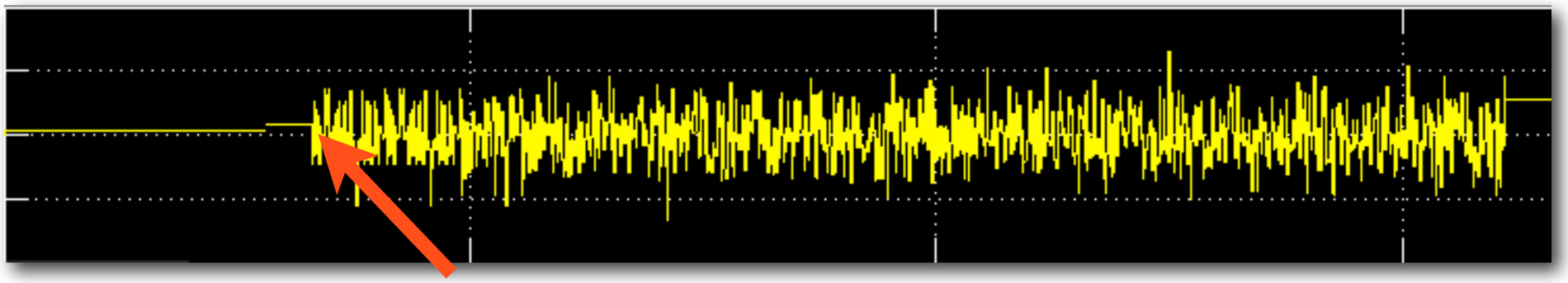


PHY Example: OFDM Tx

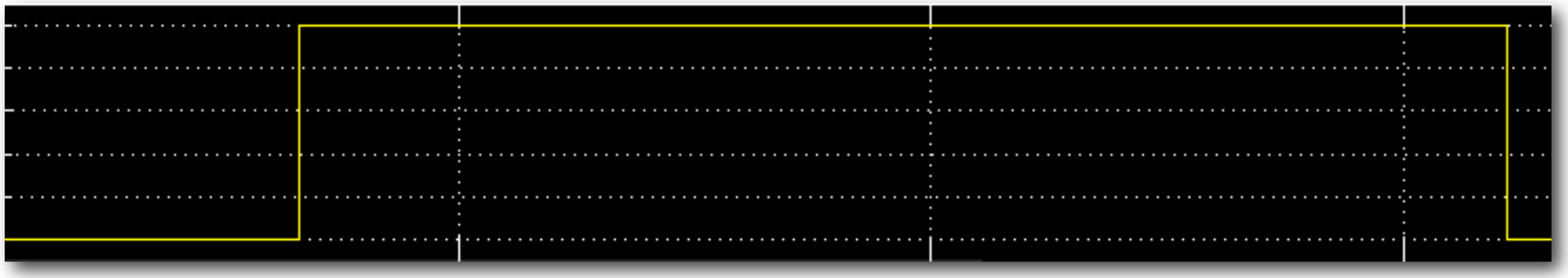
Stored Preamble



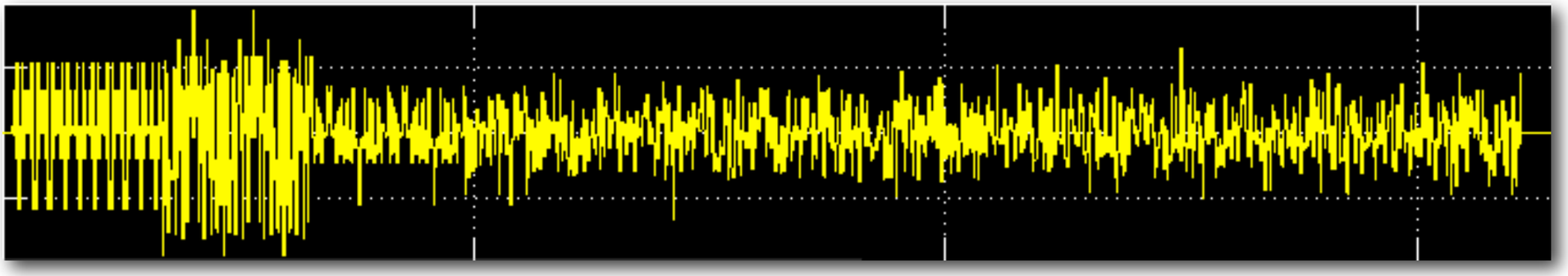
OFDM Output



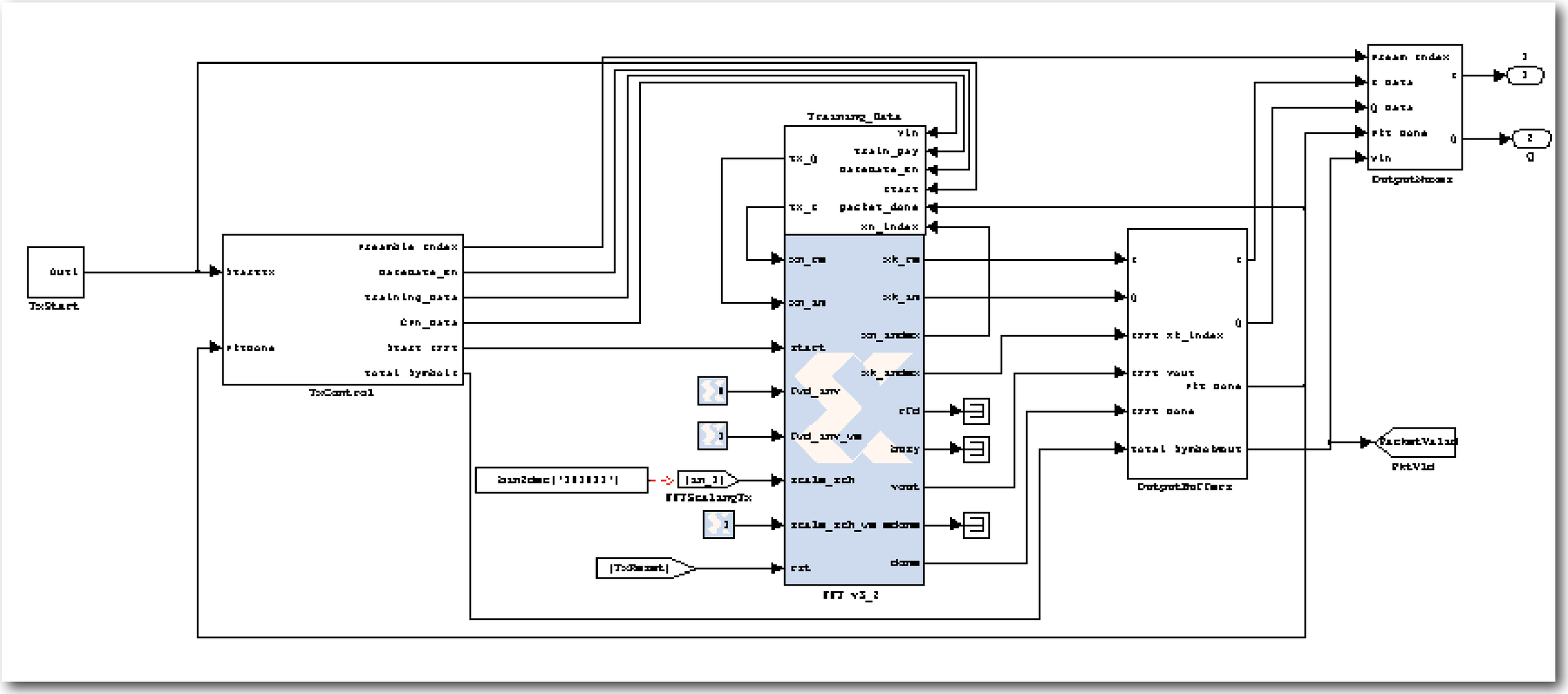
Output Mux Selection



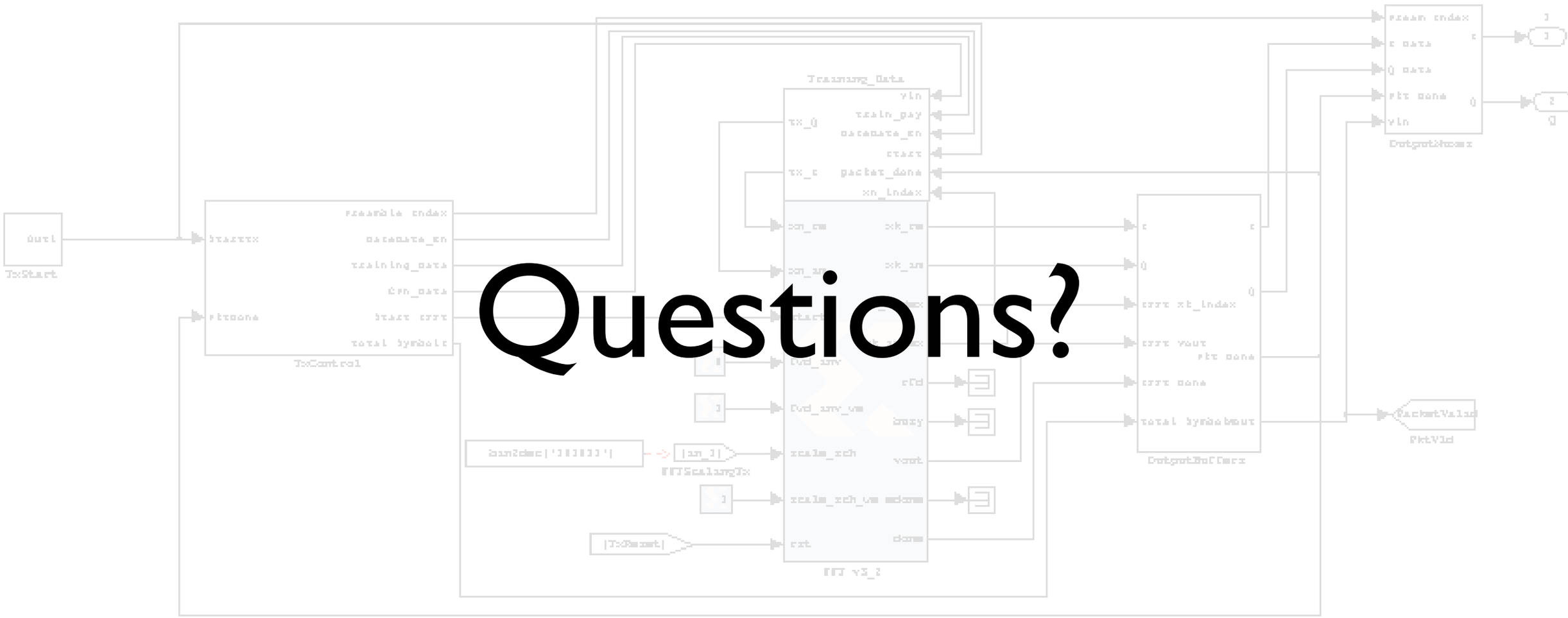
Final Output to DACs



PHY Example: OFDM Tx

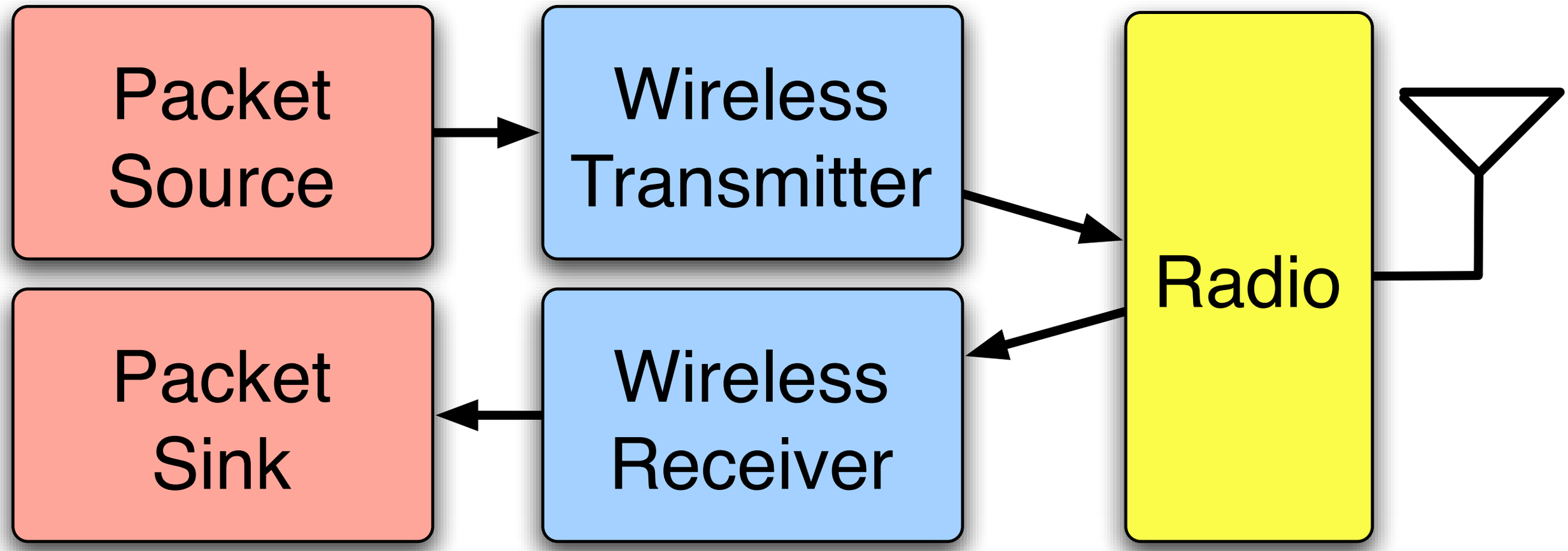


Complete model is available in the WARP repository


Questions?

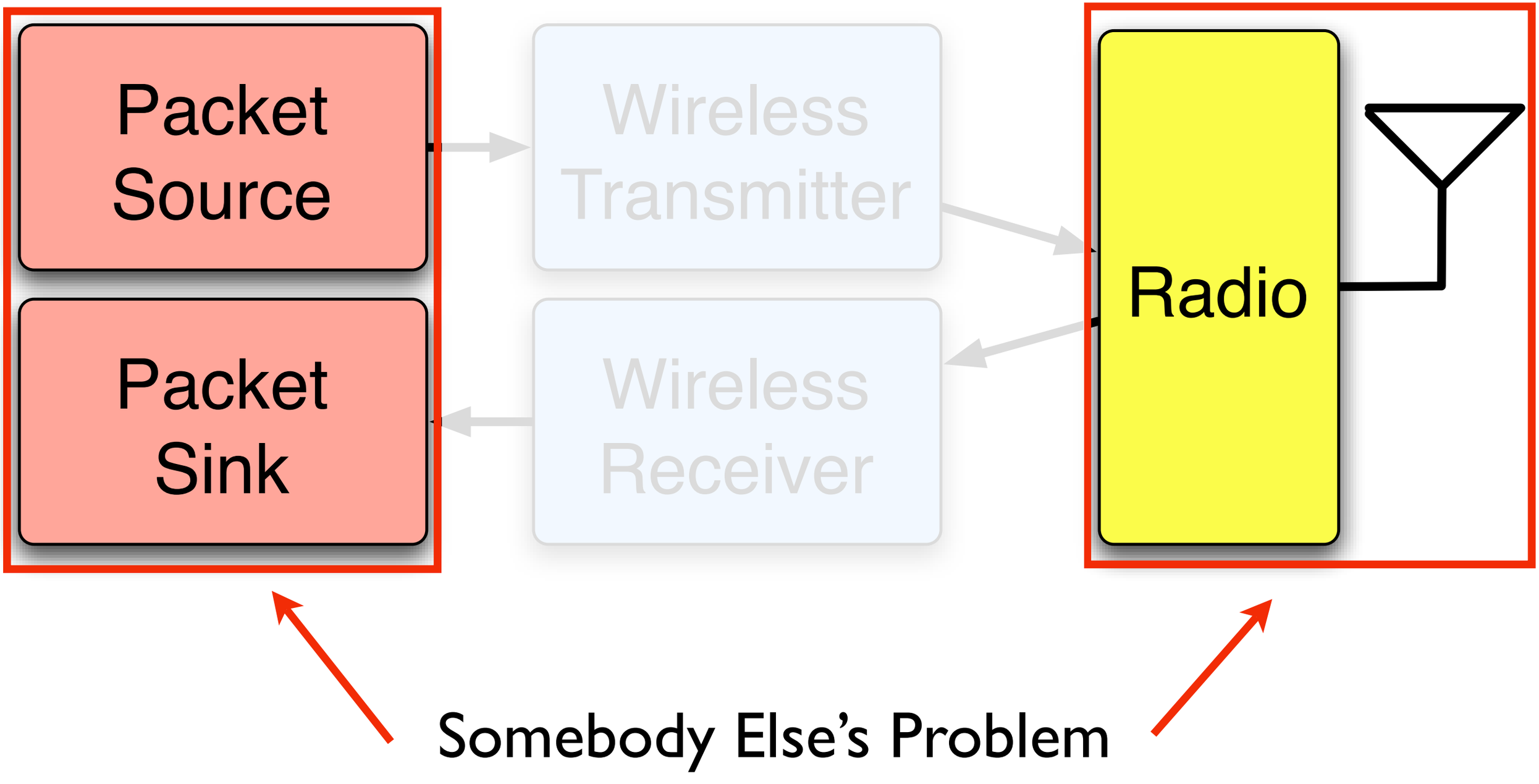
Physical Layer Basics

Simple Wireless Node



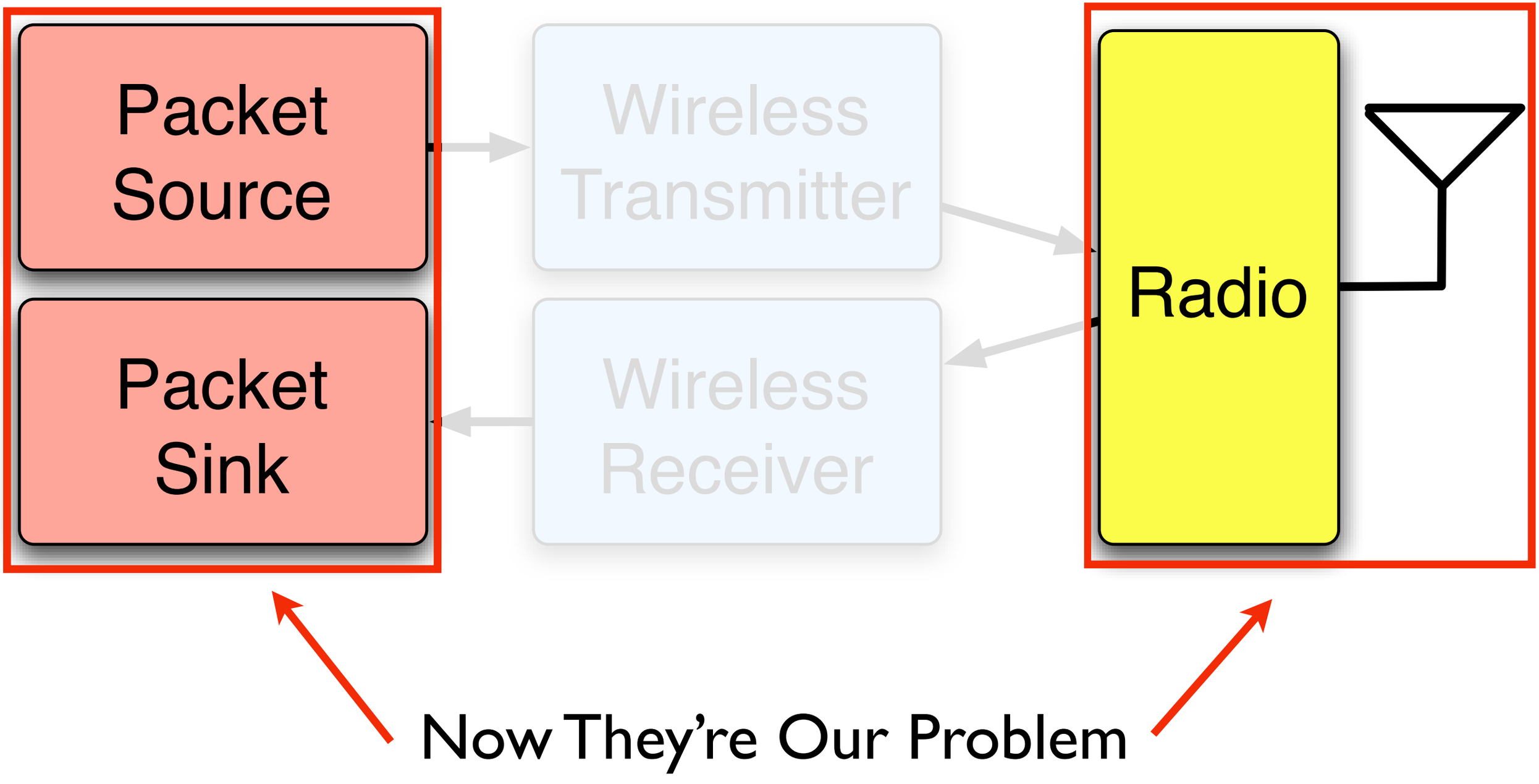
Physical Layer Basics

Simple Wireless Node

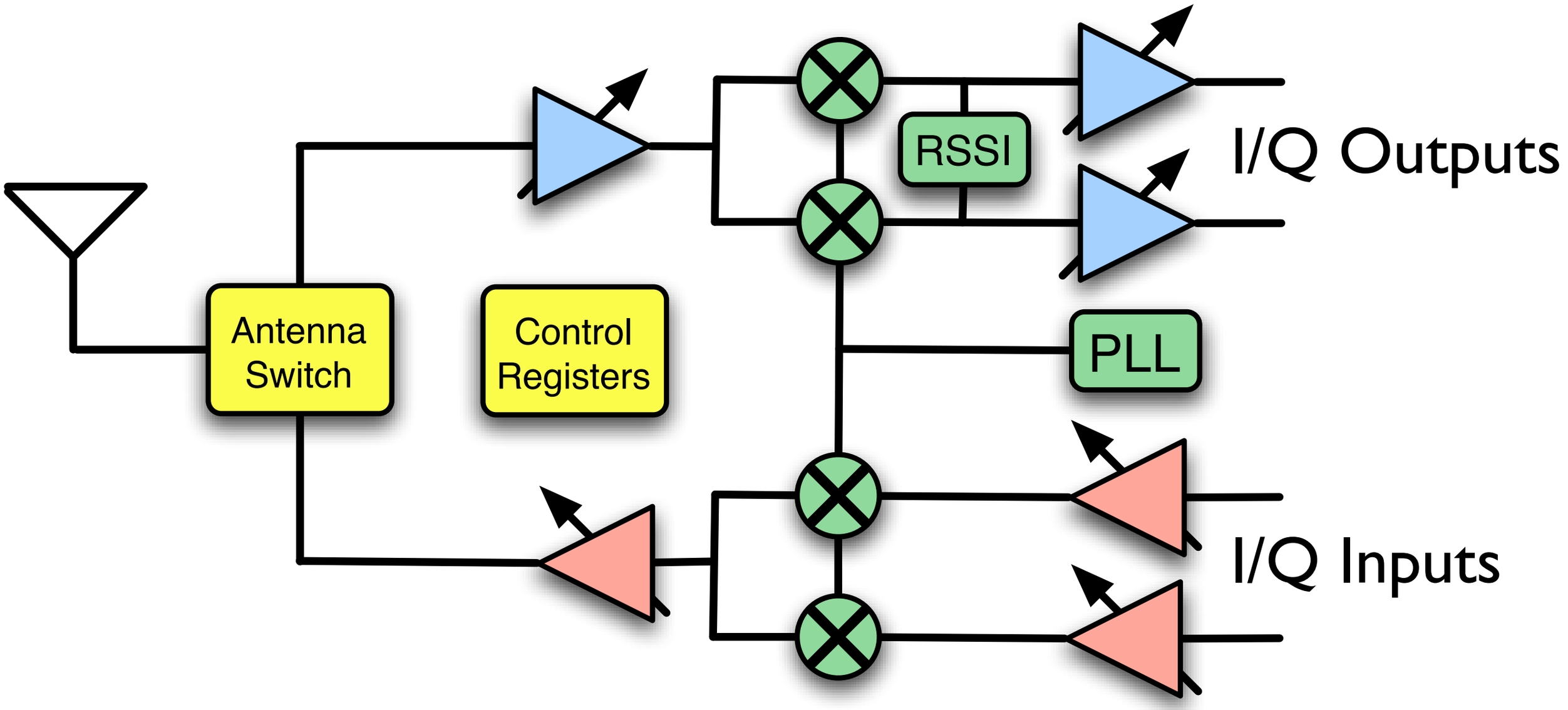


Physical Layer Basics

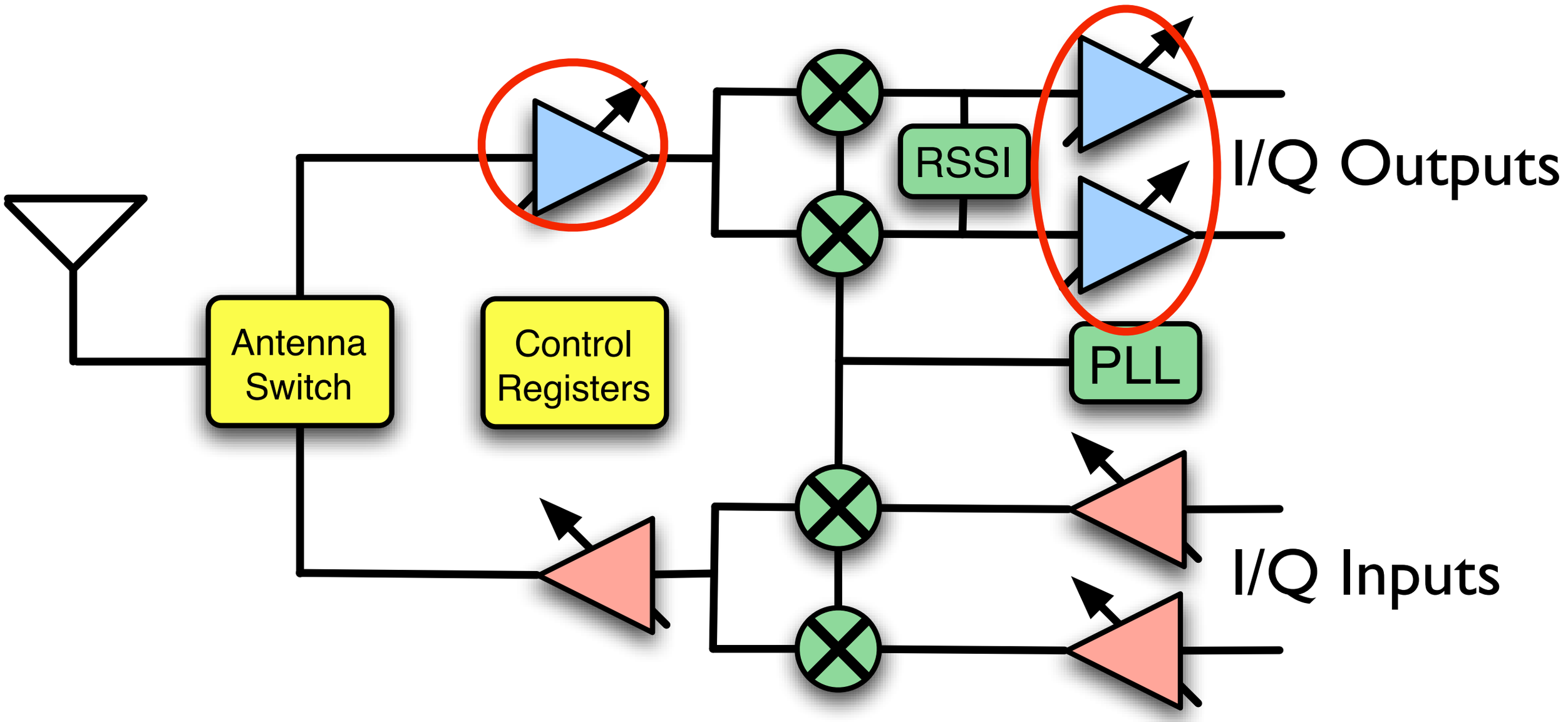
Simple Wireless Node



Radio Transceiver

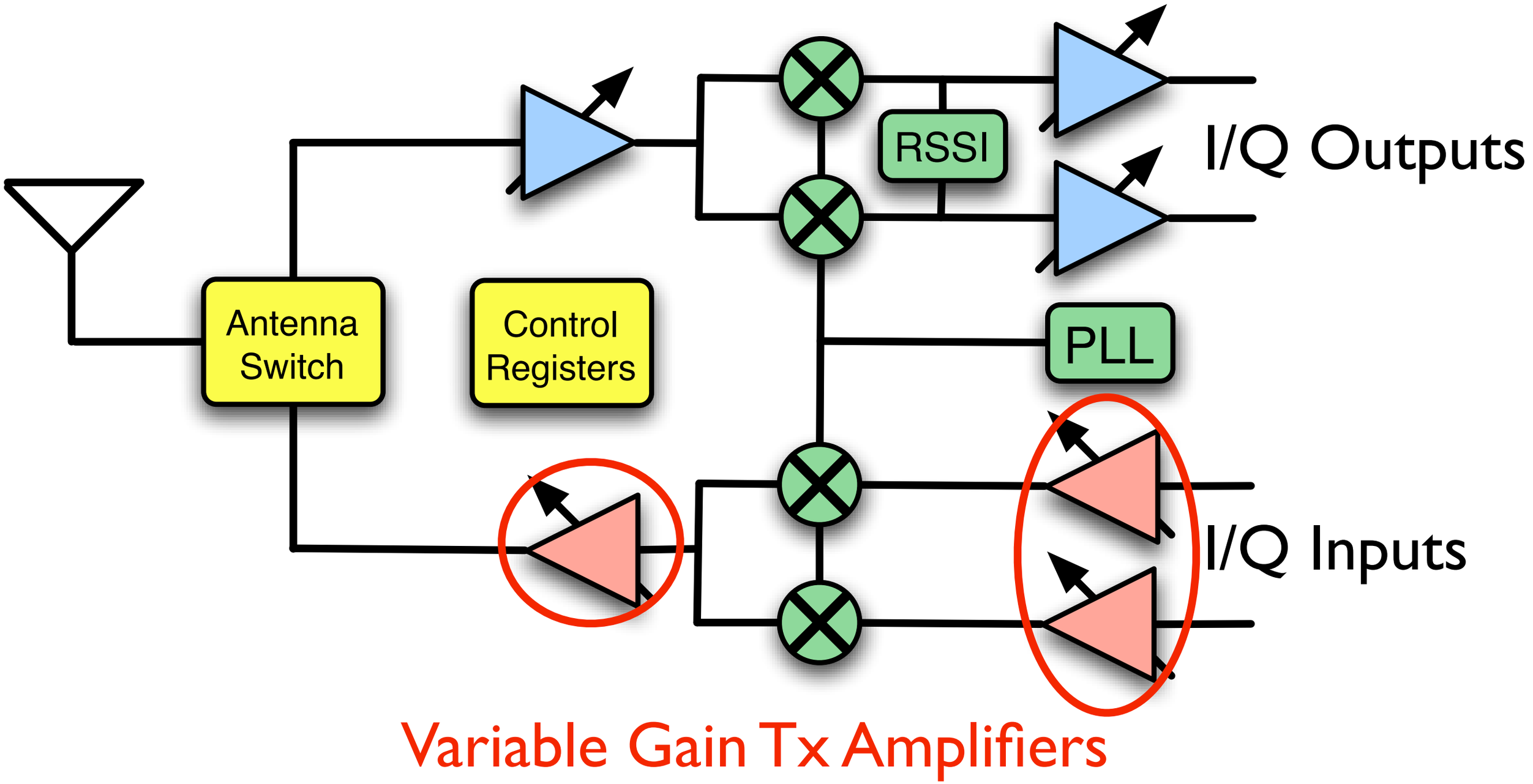


Radio Transceiver

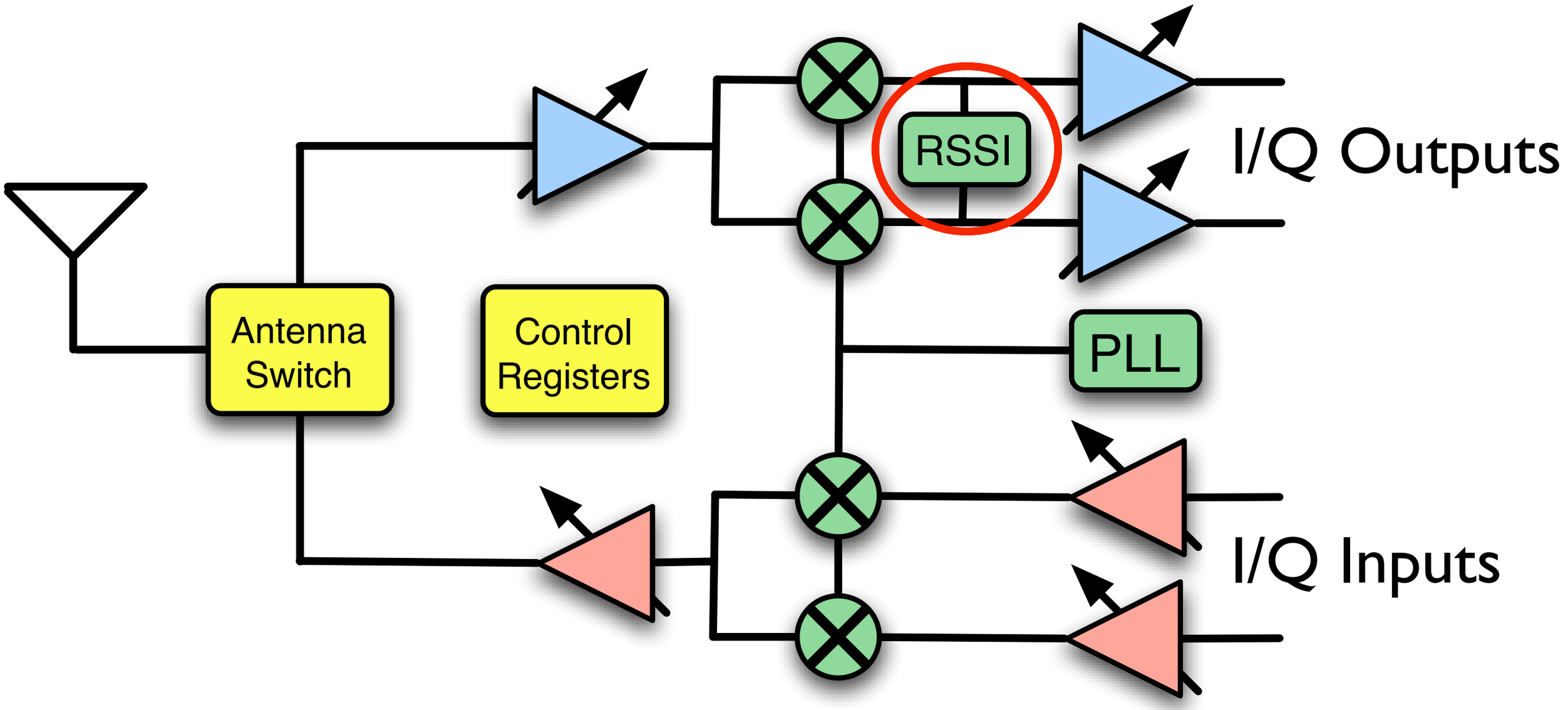


Variable Gain Rx Amplifiers

Radio Transceiver

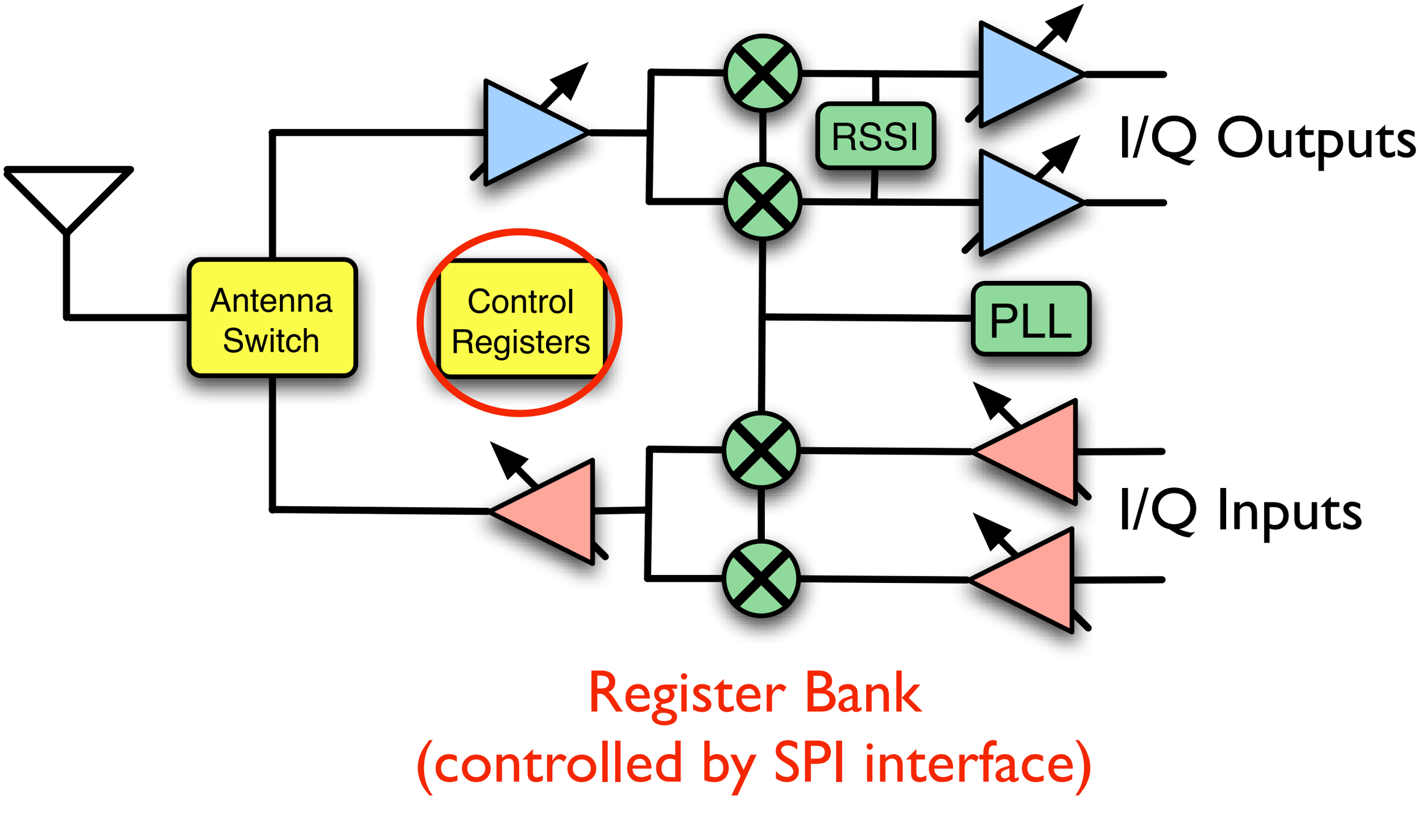


Radio Transceiver



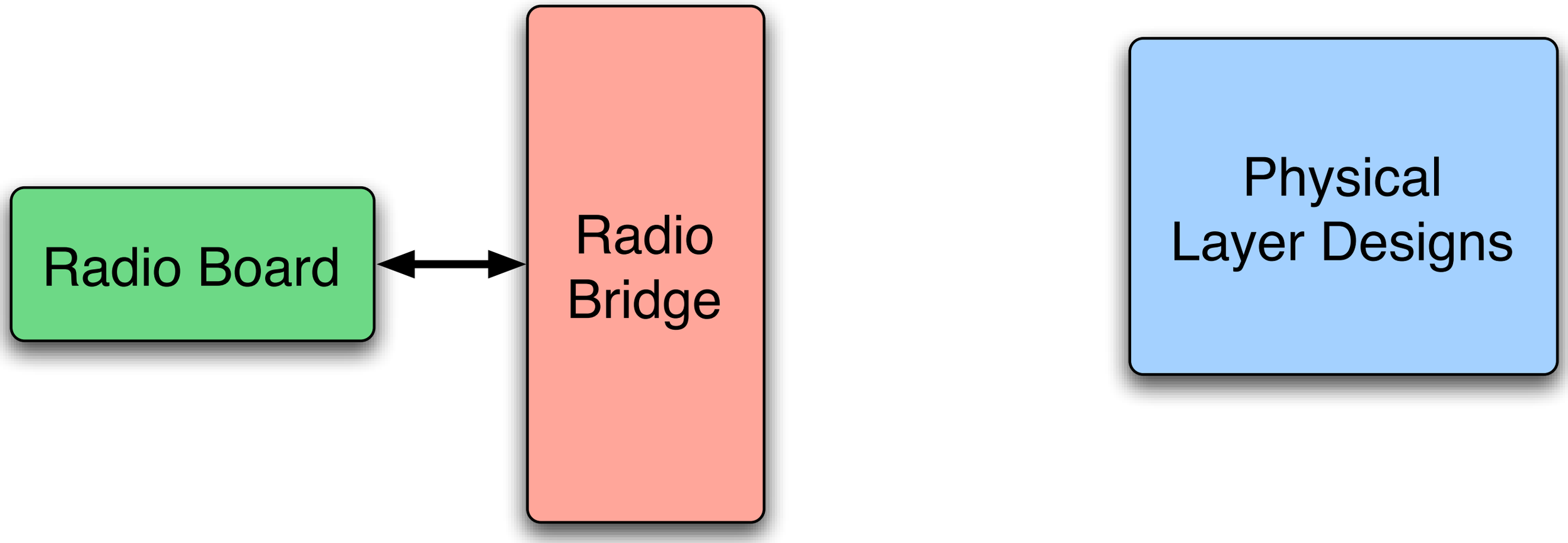
Received Signal Strength Indicator
After RF Gain

Radio Transceiver

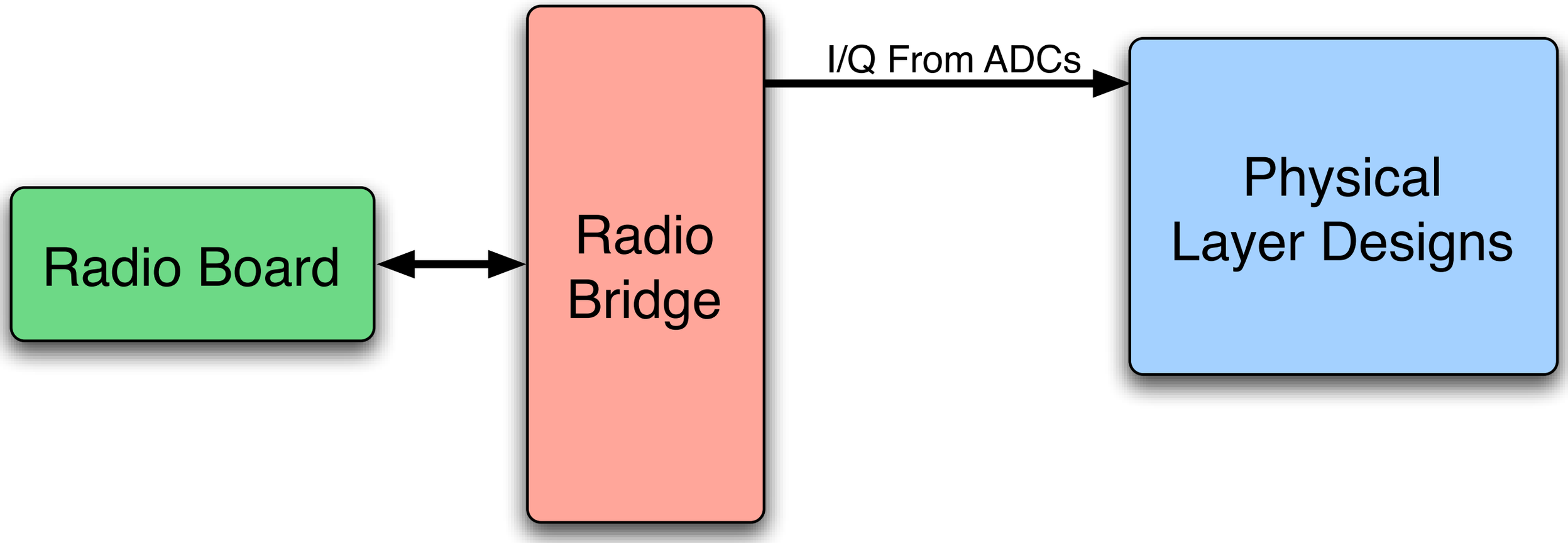


Register Bank
(controlled by SPI interface)

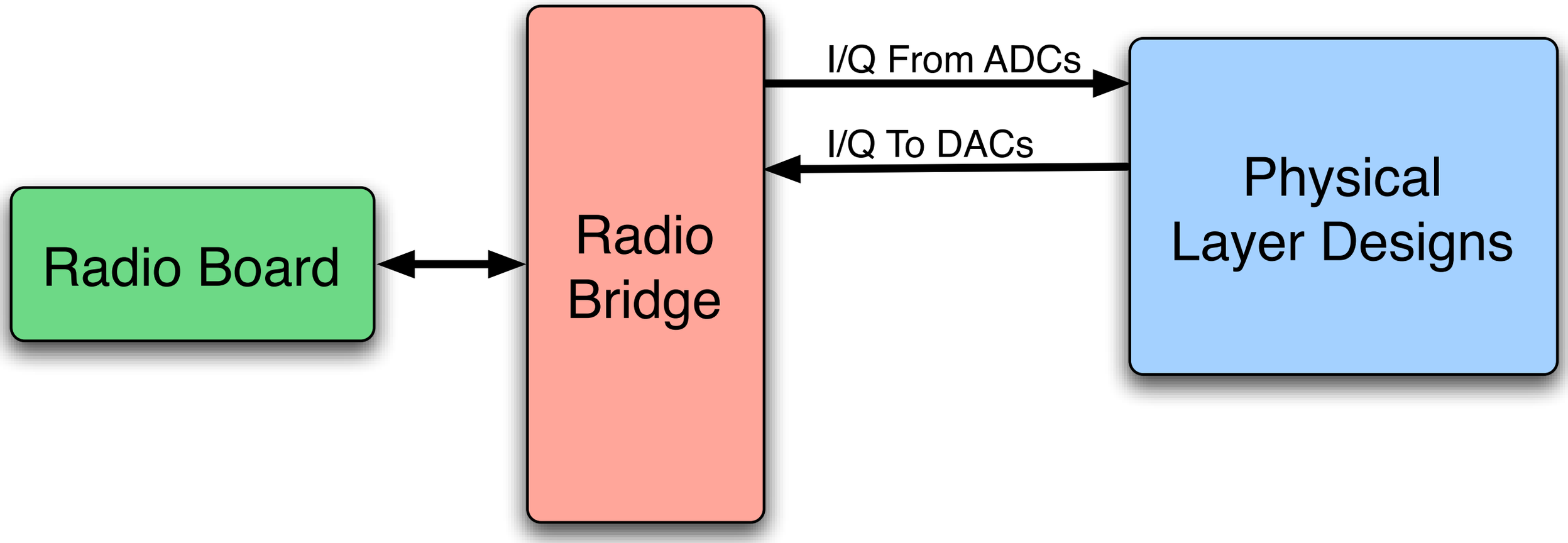
Physical Layer in Hardware



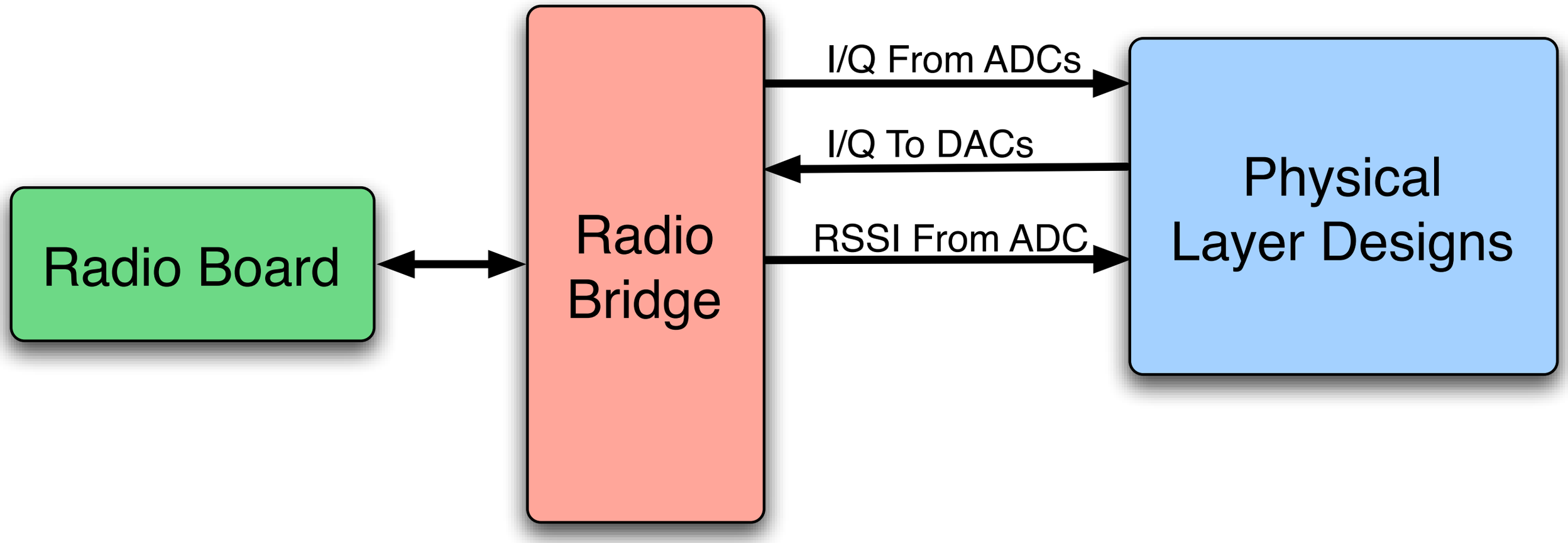
Physical Layer in Hardware



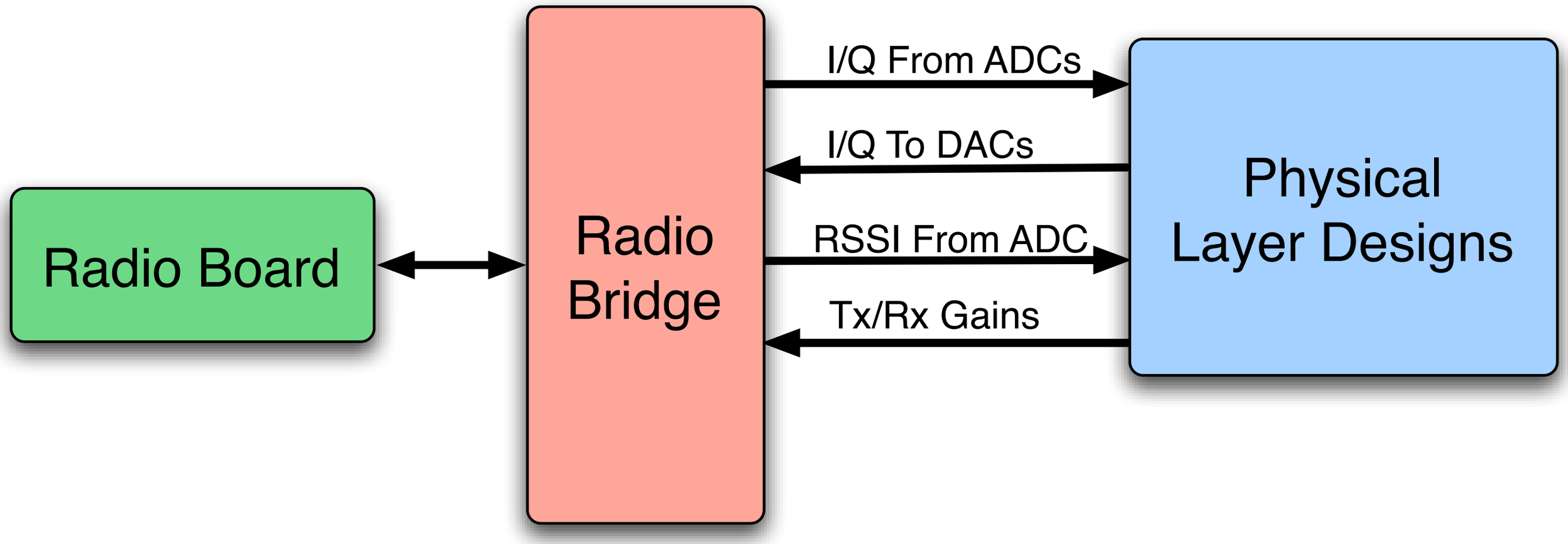
Physical Layer in Hardware



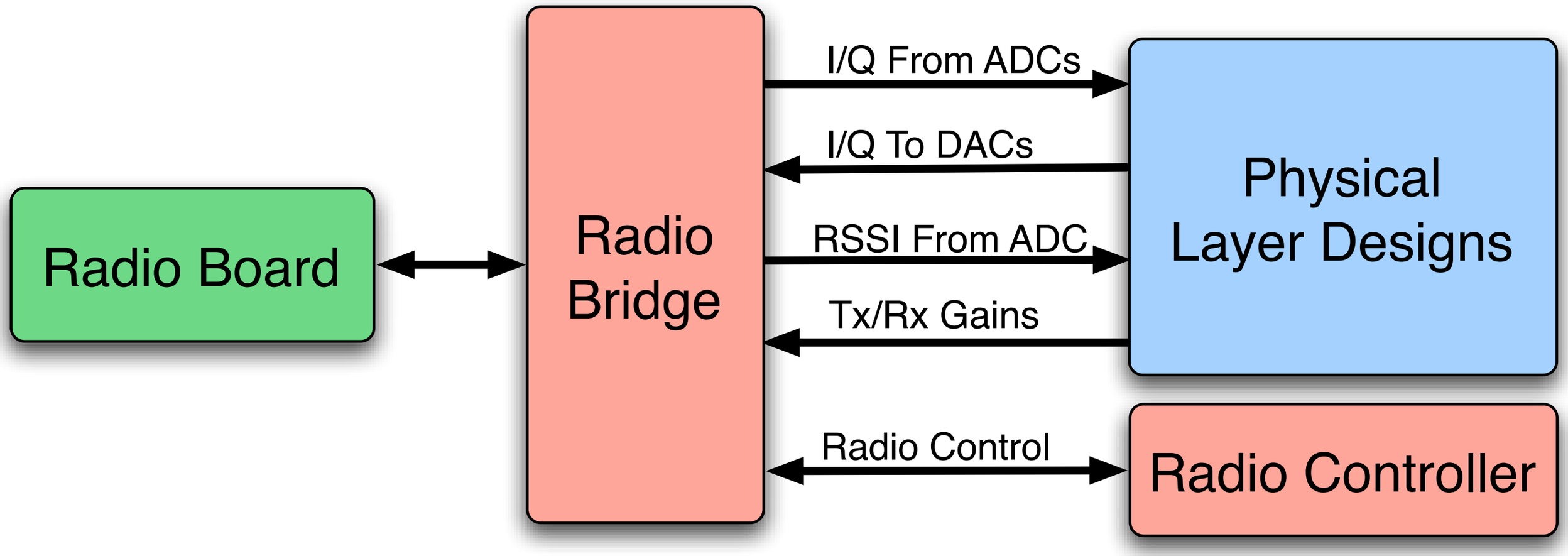
Physical Layer in Hardware



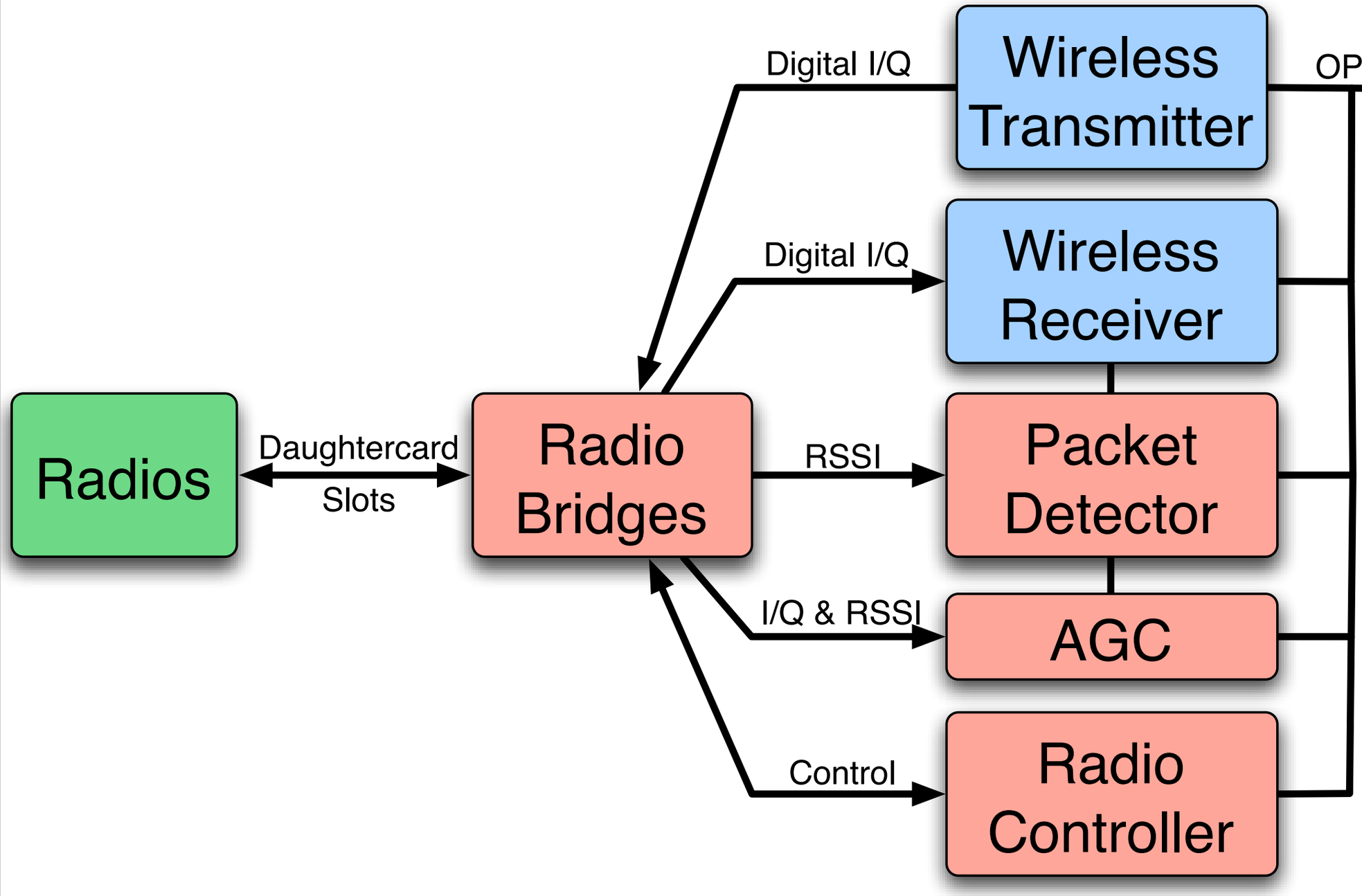
Physical Layer in Hardware



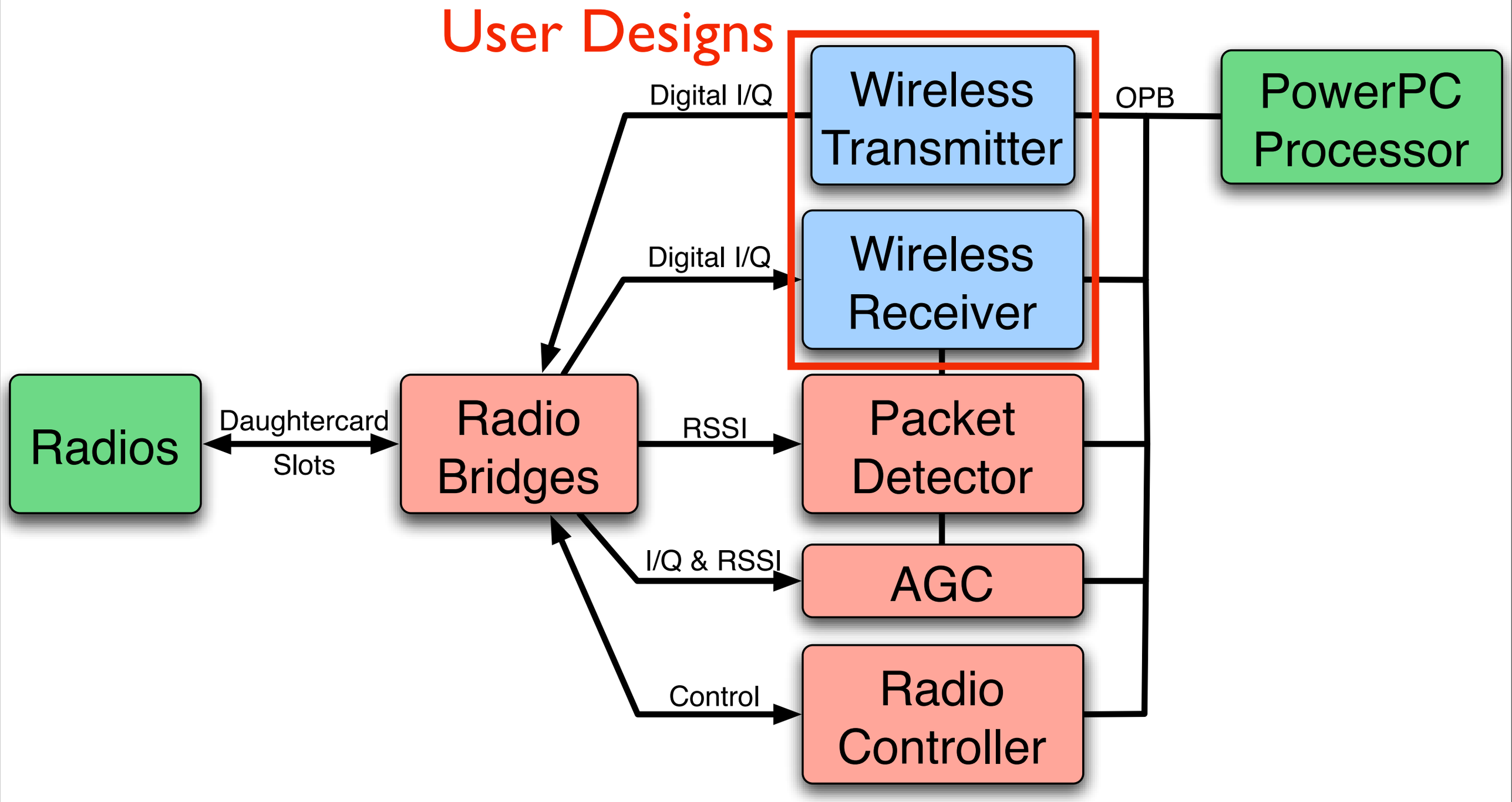
Physical Layer in Hardware



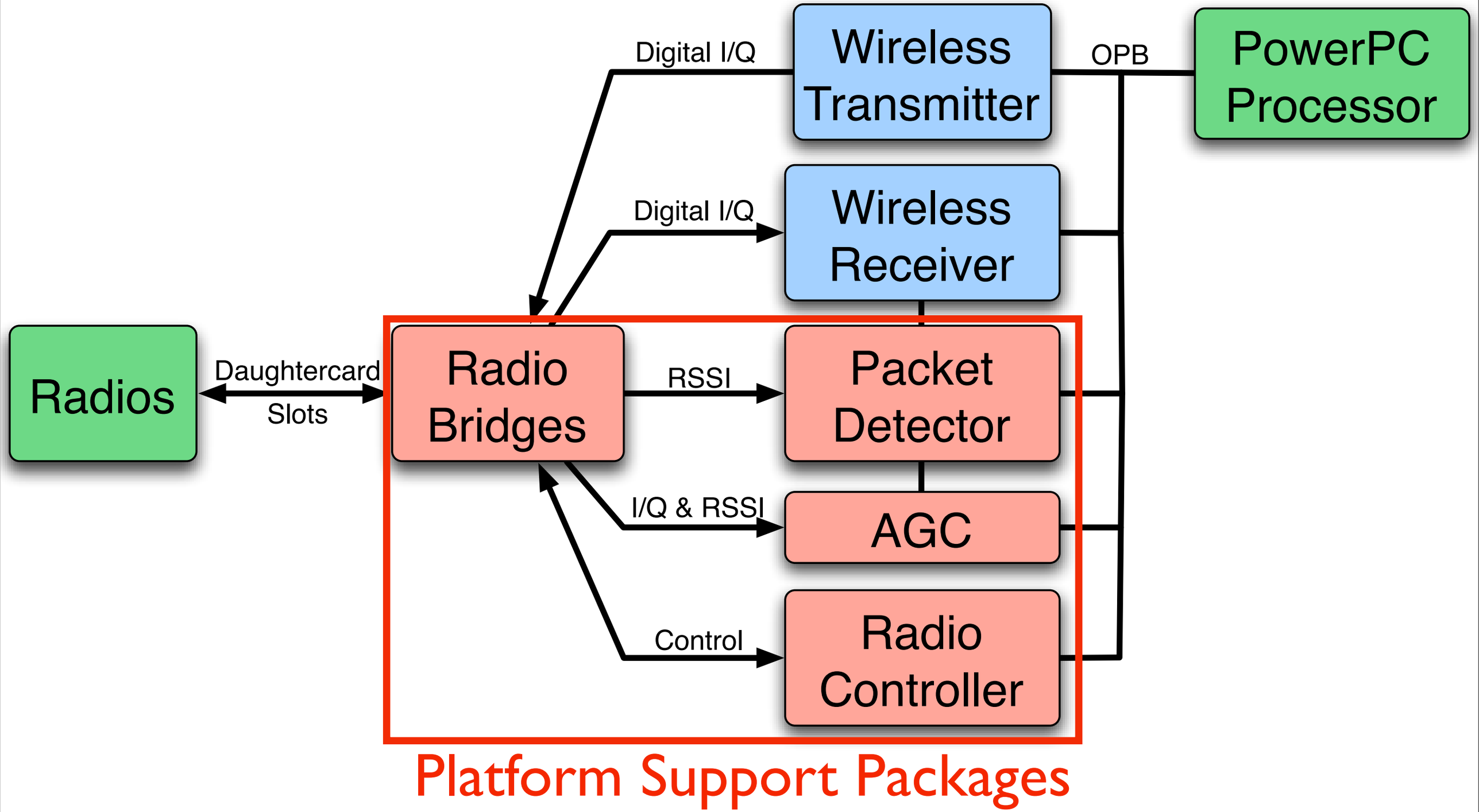
Physical Layer in Hardware



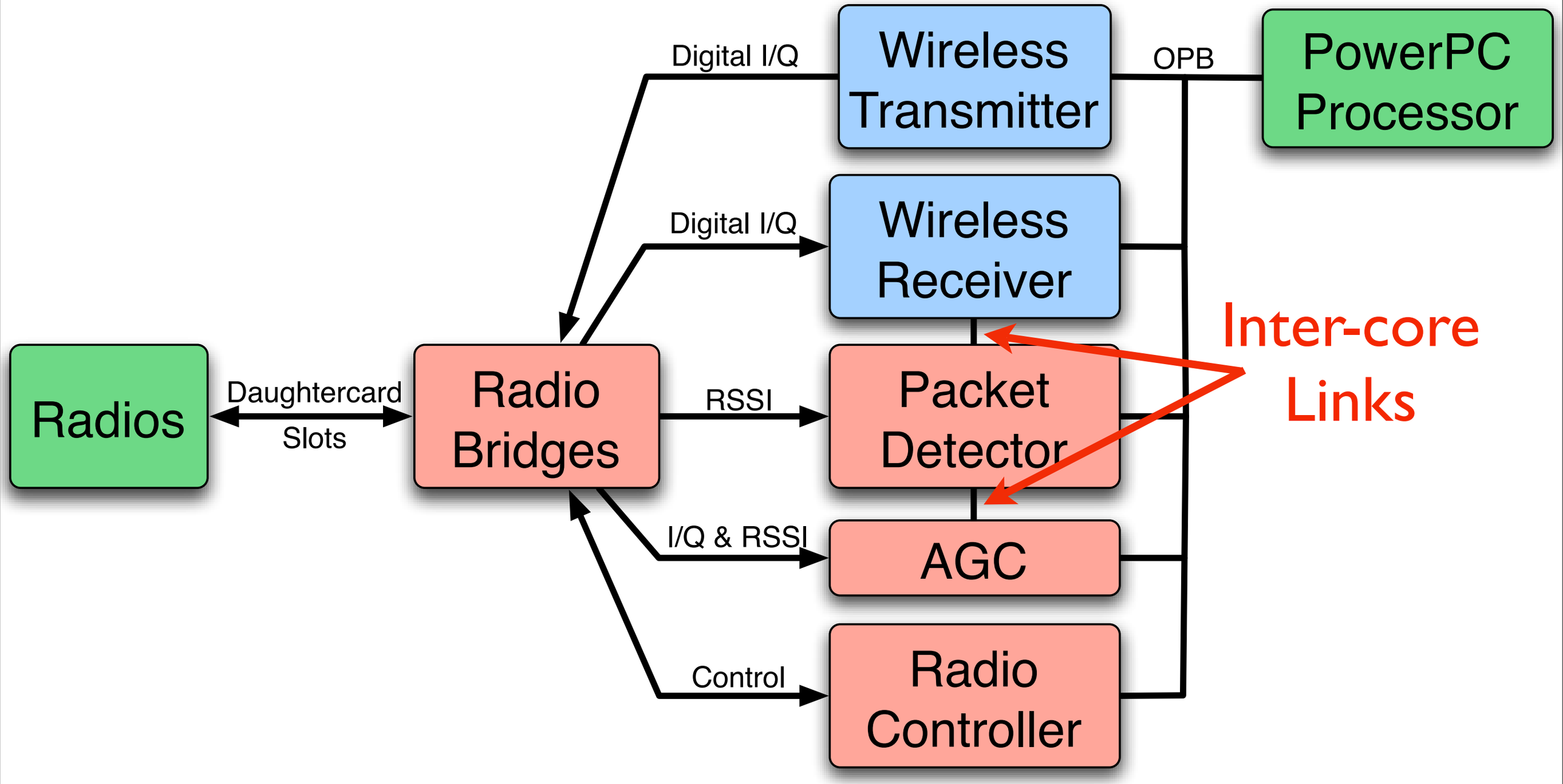
Physical Layer in Hardware



Physical Layer in Hardware



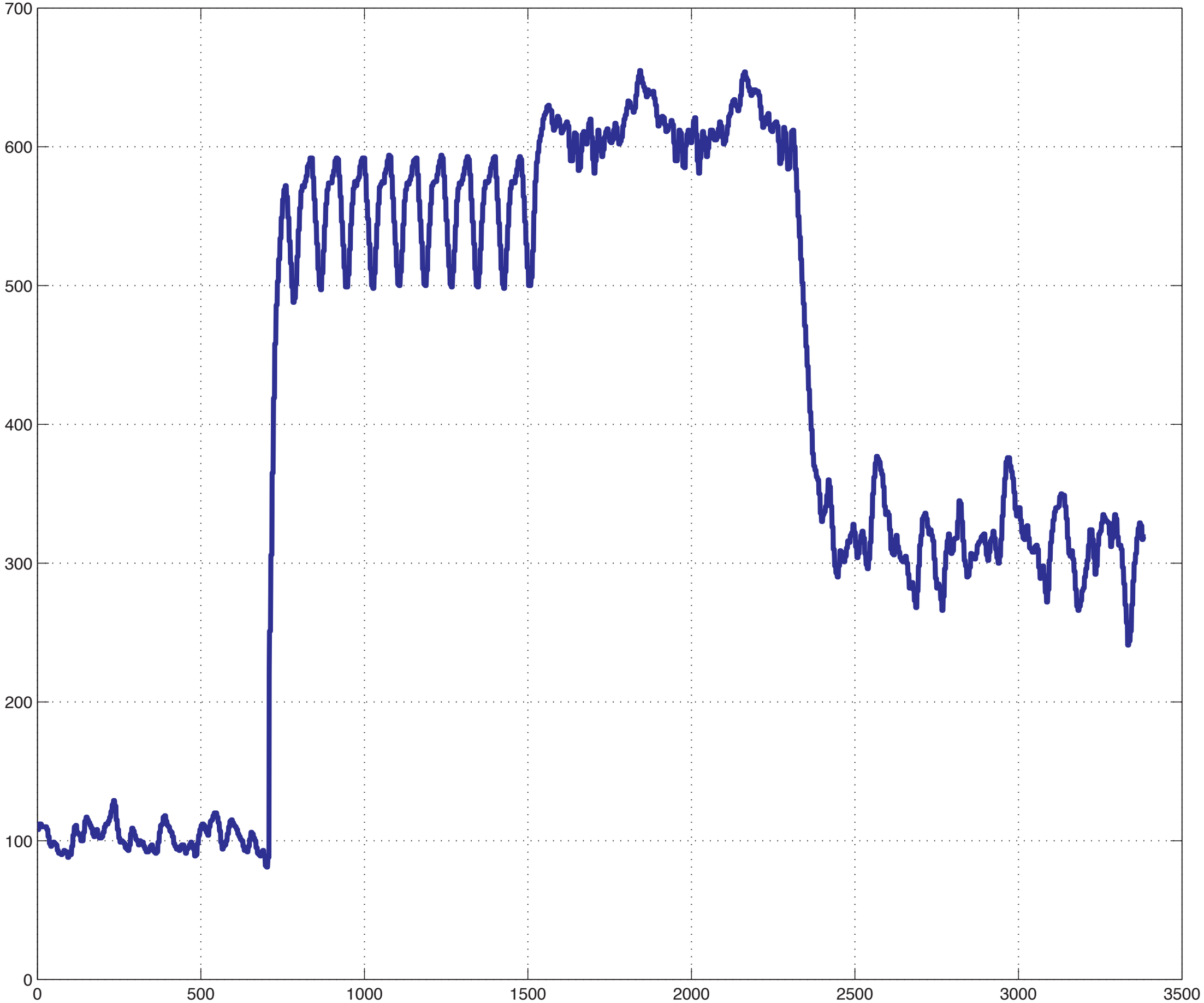
Physical Layer in Hardware



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


700
600
500
400
300
200
100
0
0
500
1000
1500
2000
2500
3000
3500

RSSI

Incoming Packet

700
600
500
400
300
200
100
0

0
500
1000
1500
2000
2500
3000
3500

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 µs
  - 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 at

http://warp.rice.edu/trac/wiki/RadioControllerAPI

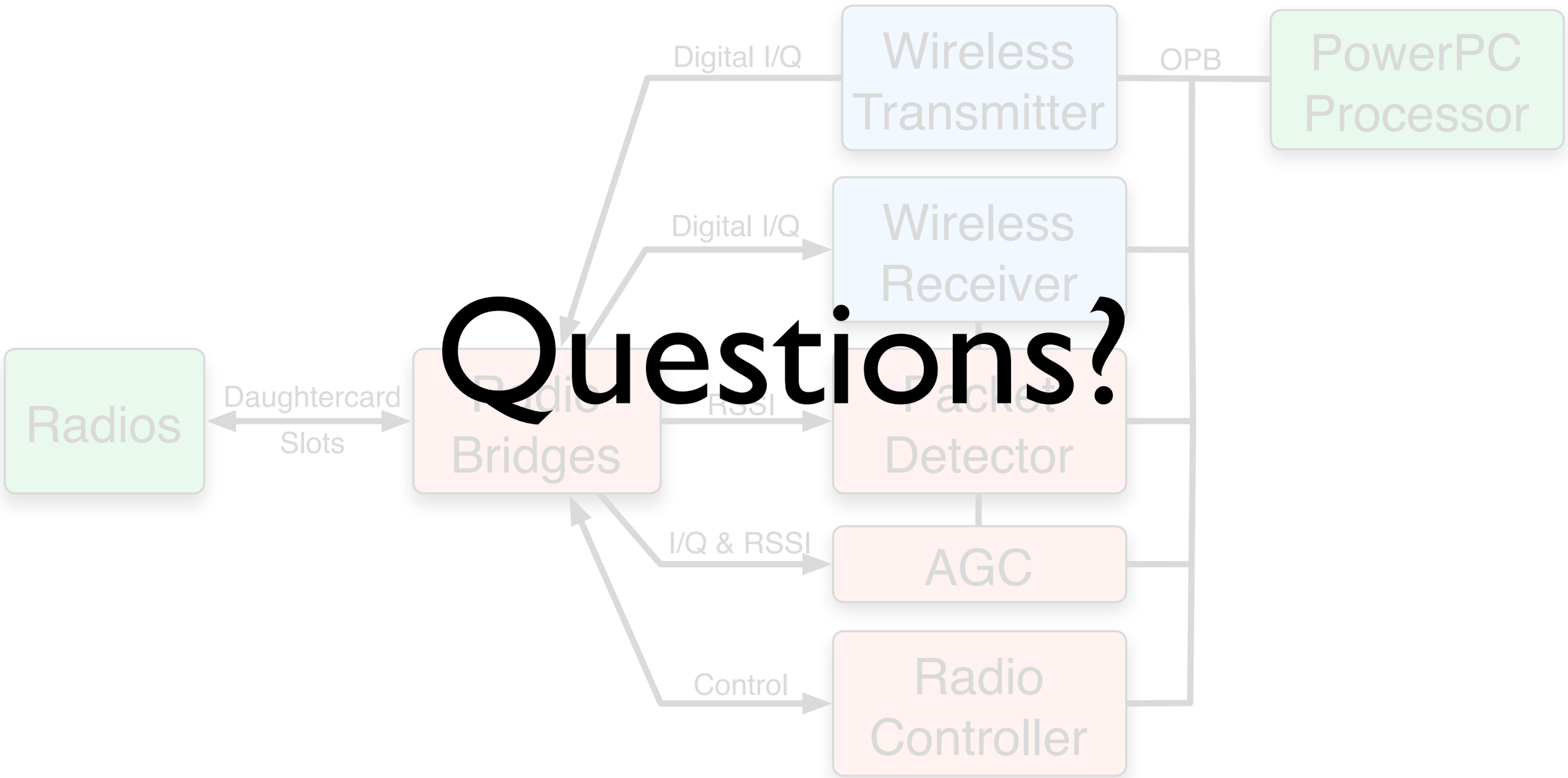
WarpRadio_v1_Reset()
WarpRadio_v1_TxEnable()
WarpRadio_v1_SetCenterFreq2GHz()
WarpRadio_v1_BaseBandTxGain()
WarpRadio_v1_TxVGAGainControl()
WarpRadio_v1_24AmpEnable()

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

Physical Layer in Hardware



Questions?

PHY Design Flow

- 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 our radio bridges & controller
- Generate the platform & test it in hardware

Lab 2: Simple Transmitter

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