Questions From Labs 1 & 2?
PHY Design - Outline

• Physical Layer Overview
• Integrating Physical Layer Designs
• Lab 3: Building a Simple Transceiver
Physical Layer Basics

- Physical Layer
- Link
- Network
- Transport
- Session
- Presentation
- Application
Physical Layer Basics

- Application
- Presentation
- Session
- Transport
- Network
- Link
- Physical
- Hardware
Physical Layer Basics

- Physical
- Hardware
- Link
- Network
- Transport
- Session
- Presentation
- Application
Physical Layer Basics

Hardware

Physical

Link

Network

Transport

Session

Presentation

Application

TCP/UDP

IP

MAC

Physical

Hardware

Application
Physical Layer Basics

INFORMATION

PHY

Radio
Physical Layer Basics

Application Dependent
(Bytes, Packets, Waveforms, etc.)

INFORMATION

PHY

Radio
Physical Layer Basics

**INFORMATION**

**PHY**

**Radio**

*Always Waveforms & Control*
Physical Layer in Hardware

- Processor
- Physical Layer Design
- Radio Controller
- WARP Radio Board

Connections:
- I/Q From ADCs
- I/Q To DACs
- Tx/Rx Gains
- RSSI From ADC
- Radio Control
Radio interface is real-time

PHY/radio interface uses FPGA logic

Many options for “information” source
PHY-Radio Interface

- Two 14-bit samples (I/Q) at 40MSps
- Direct sampling of radio I/Q outputs
- Radio settings are very important
• Two 16-bit samples (I/Q) at 40MSps
• Direct sampling to radio I/Q inputs
- Four variable-gain amplifiers in Tx/Rx paths
- All under FPGA control
PHY-Radio Interface

Tx Gains

- Antenna Switch
- Control Registers
- PLL
- RF Variable Gain Amplifier (VGA)
- Baseband Amplifiers

TX OUTPUT POWER vs. GAIN SETTINGS

(MAX2829 datasheet pg. 17)
PHY-Radio Interface

Rx Gains

<table>
<thead>
<tr>
<th>RX VOLTAGE GAIN vs. BASEBAND GAIN SETTINGS</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAIN SETTINGS</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>LNA = LOW GAIN</td>
</tr>
<tr>
<td>LNA = MEDIUM GAIN</td>
</tr>
<tr>
<td>LNA = HIGH GAIN</td>
</tr>
<tr>
<td>LNA = LOW GAIN</td>
</tr>
</tbody>
</table>

(MAX2829 datasheet pg. 15)
PHY-Radio Interface

Rx Gains

RX EVM vs. $P_{IN}$

- LNA = HIGH GAIN
- LNA = LOW GAIN
- LNA = MEDIUM GAIN

(MAX2829 datasheet pg. 16)
PHY-Radio Interface

Rx Gains

RX EVM vs. PIN

(MAX2829 datasheet pg. 16)
PHY-Radio Interface

- 10-bit samples at 10Mspses
- Direct sampling of radio’s RSSI output
PHY-Radio Interface

RX RSSI OUTPUT vs. INPUT POWER

(MAX2829 datasheet pg. 16)
PHY-Radio Interface

RX RSSI OUTPUT vs. INPUT POWER

(MAX2829 datasheet pg. 16)
PHY-Radio Interface

**RSSI**

RF Transceiver (MAX2829)

A/D Converter (AD9200)

0.5v $\Rightarrow$ RSSI_ADC[9:0] = 0
2.5v $\Rightarrow$ RSSI_ADC[9:0] = 1023

RX RSSI OUTPUT vs. INPUT POWER

LNA = LOW GAIN
LNA = MEDIUM GAIN
LNA = HIGH GAIN

(Max2829 datasheet pg. 16)
PHY-Radio Interface

**RSSI**

- **RF Transceiver (MAX2829)**
- **A/D Converter (AD9200)**

0.5v $\Rightarrow$ RSSI_ADC[9:0] = 0
2.5v $\Rightarrow$ RSSI_ADC[9:0] = 1023
- PHY is a memory mapped peripheral
- Uses both registers and memory blocks
- Bus interface is generated automatically
PHY-Processor Interface

Software sees 32-bit memory space

Hardware sees 32 wires per address
PHY-Processor Interface

Example

![Diagram of Carrier Sensing Logic]

CS_THRESH

RSSI

Running Average

BUSY

Carrier Sensing Logic
PHY-Processor Interface

Example

PHY uses memory-mapped wires for:

0x80000000

---

Carrier Sensing Logic

CS_THRESH

10

RSSI

10

Running Average

> BUSY

PHY uses memory-mapped wires for:
PHY-Processor Interface

Example

PHY uses memory-mapped wires for:

- Parameters

Carrier Sensing Logic

CSI_THRESH

0x80000000

RSSI

Running Average

BUSY

10

10
PHY-Processor Interface

Example

PHY uses memory-mapped wires for:

- Parameters
- Feedback

Carrier Sensing Logic

- CS_THRESH
- CS_BUSY

0x80000000

Running Average

\[ \text{RSSI} > \text{BUSY} \]
PHY-Processor Interface

**Example**

PHY uses memory-mapped wires for:
- Parameters
- Feedback
- Control

![Diagram showing PHY-Processor Interface](image)
PHY-Processor Interface

• Hardware/software ≠ PHY/MAC
• Some MAC functions in logic
• Some PHY control in software
• Parameterization vs. complexity is tradeoff
• Easier to iterate on software
• Designing flexible hardware is tricky
Transceiver Examples

- WARPLab
- MIMO OFDM
Transceiver Examples

WARPLab Transceiver

- Application provides digital waveforms
- Transceiver handles radio data I/O
- Software handles radio control
- Application provides packets & parameters
- PHY processes waveforms and radio I/O
- All processing is real-time
Transceiver Examples

**OFDM Transmitter**

Packet Buffer → CRC Calculation → Modulation → Training Symbols → IFFT → Preamble → Cyclic Extension → Interpolation → I/Q to DACs

**OFDM Receiver**

RSSI from ADC → PHY Control → Packet Detection → AGC → Rx Gains → CFO Correction → Cyclic Prefix Removal → FFT → Channel Estimation → Equalization → Demodulation → Packet Buffer
Questions?
(Lab 3 is next)
Lab 3: Simple Transceiver

- Build a simple transmitter in Sysgen
- Convert the model to a PLB peripheral
- Connect the Tx core to the radio bridge
- Test the model at RF
- Add an RSSI-based receiver
- Update the hardware and test