WLAN erklärt: Wie aus Funksignalen Ethernet Pakete werden

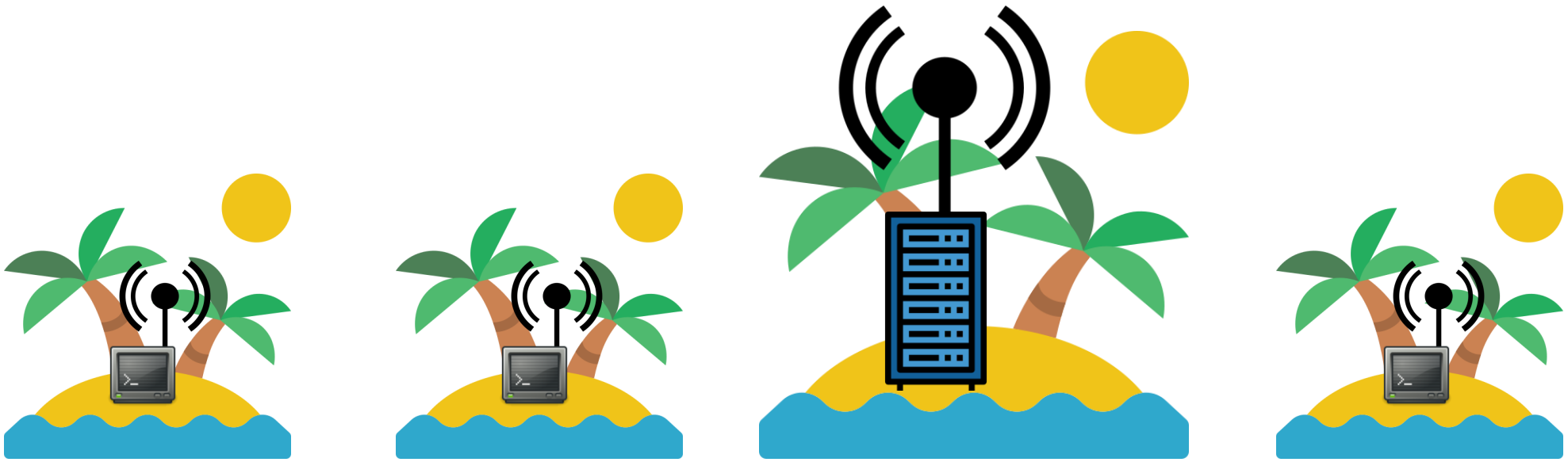
Source: https://zeptobars.com/en/read/AD9361-SDR-Analog-Devices-DAC-ADC-65nm

Wifi Chip as Blackbox

Signals In

Internet out?

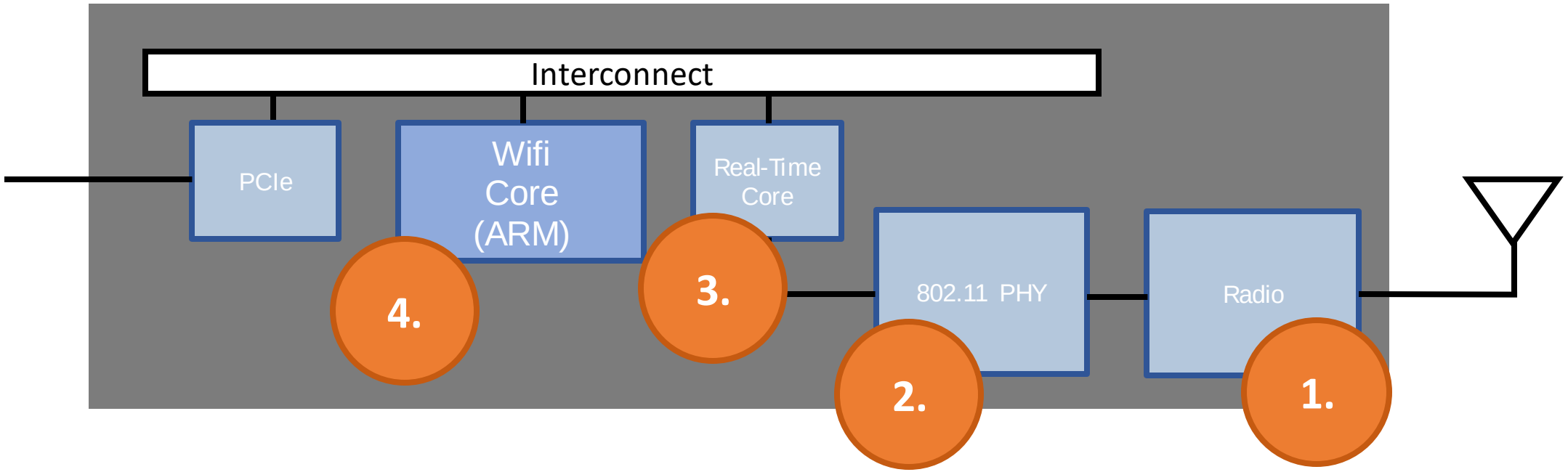
History: ALOHAnet



Wifi Standards

Year of Adoption	IEEE Standard	Generation Name
1999	802.11**a**	Wi-Fi 2
2003	802.11**g**	Wi-Fi 3
2008	802.11**n**	Wi-Fi 4
2014	802.11**ac**	Wi-Fi 5
2019	802.11**ax**	Wi-Fi 6
2020	802.11**ax + 6GHz**	Wi-Fi 6e
2024	802.11**be**	Wi-Fi 7

Building blocks of a Wifi Chip

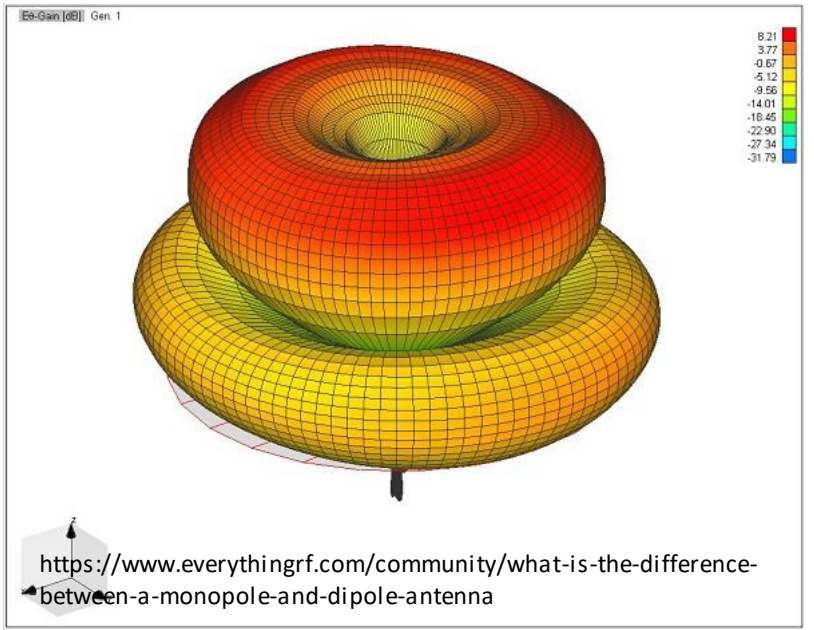


Wave length

- Antenna needs to resonate with the frequency we need
- 2.4 GHz for Wifi at channel 6:

$$\lambda = \frac{v}{f} \quad \lambda = \frac{299.792.458 \frac{m}{s}}{2.437.000.000 \frac{1}{s}} = 0.12 \text{ m}$$

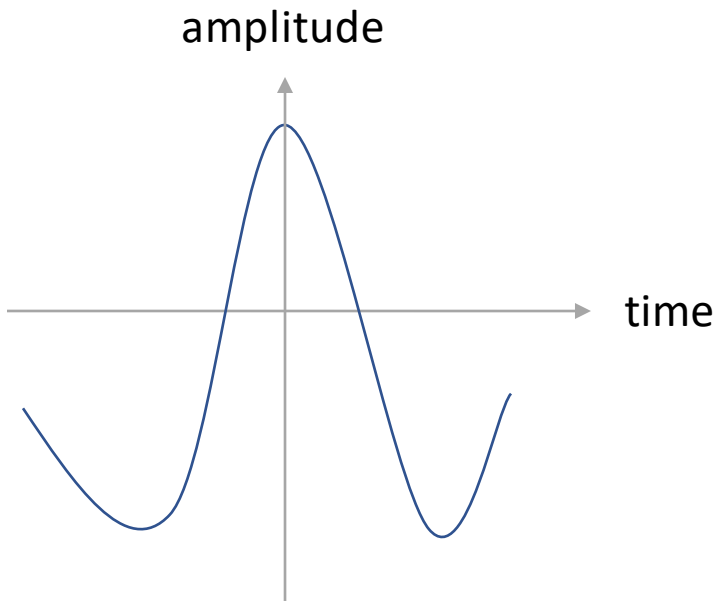
- Antenna length can also be **half** or **quarter** the wave length
- Antenna **orientation** is important! Keep Antennas of sender and receiver on the same polarization.

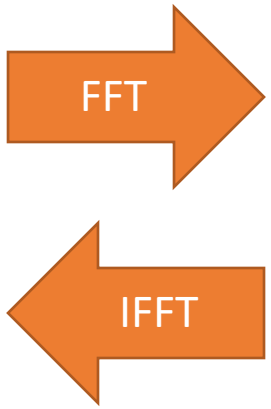


https://www.everythingrf.com/community/what-is-the-difference-between-a-monopole-and-dipole-antenna

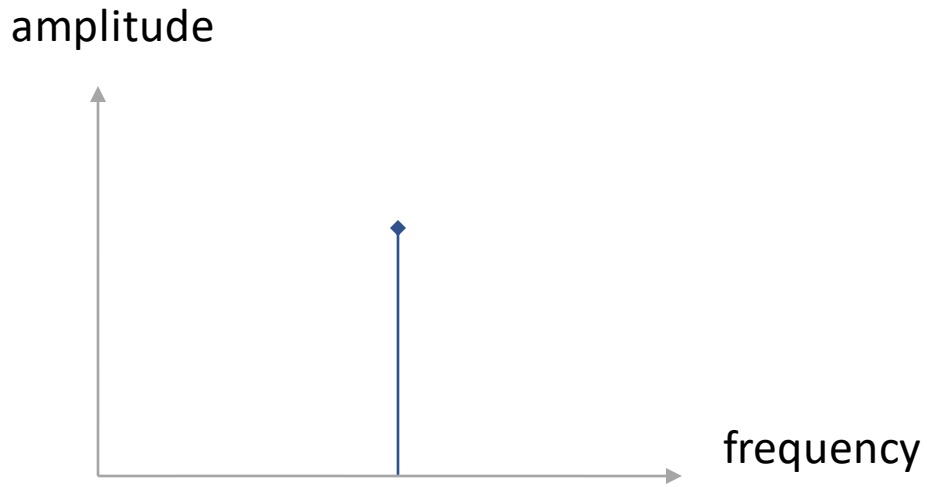
Signals

- Time Domain



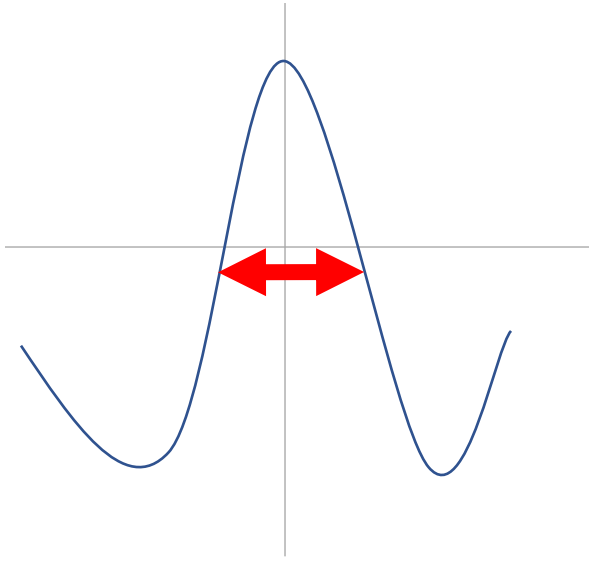


- Frequency Domain

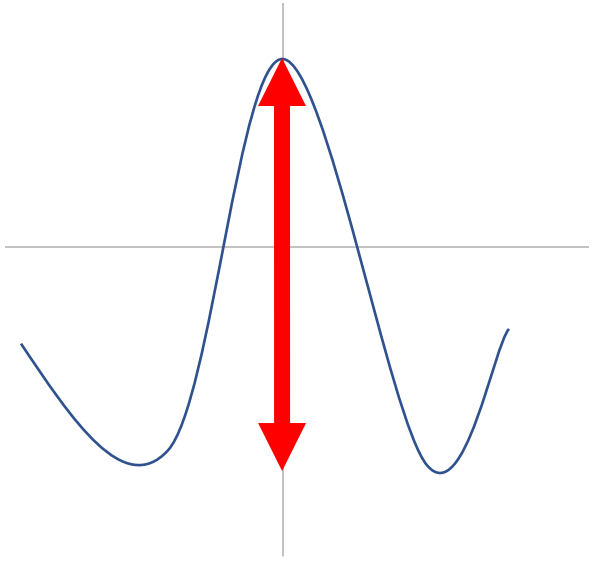


Ways to encode data in wireless signals

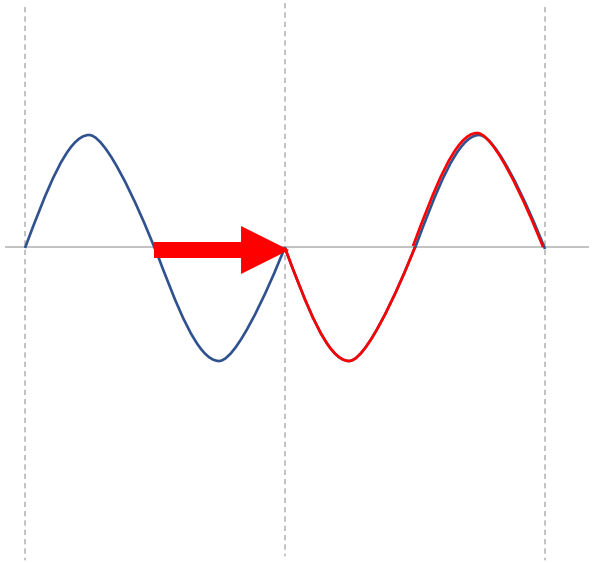
- Frequency



- Amplitude



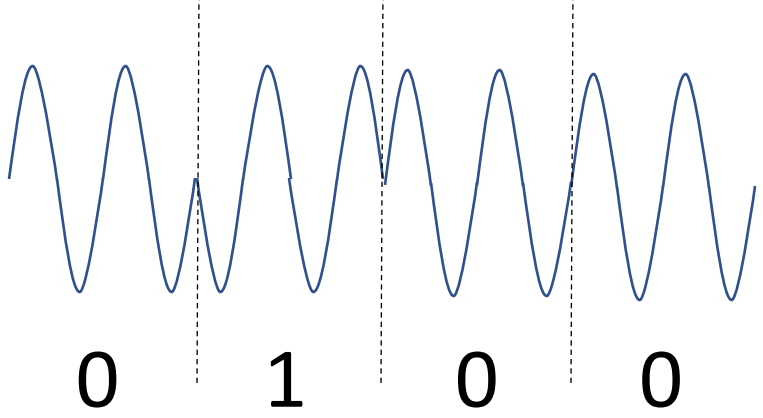
- Phase



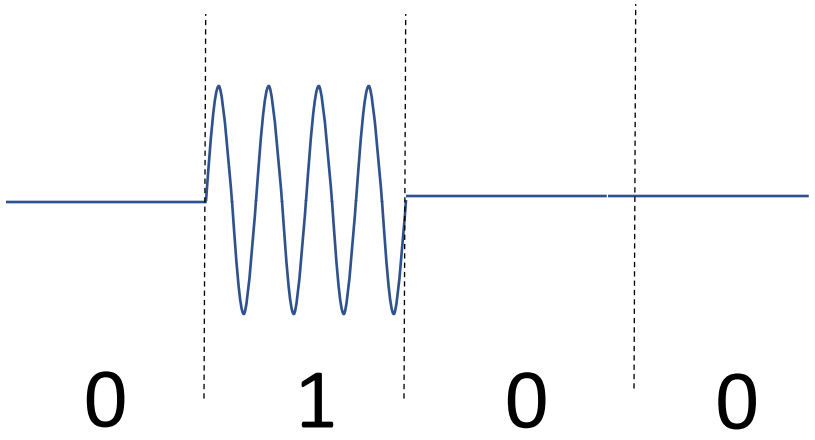
For WIFI

Modulation

- Phase Modulation



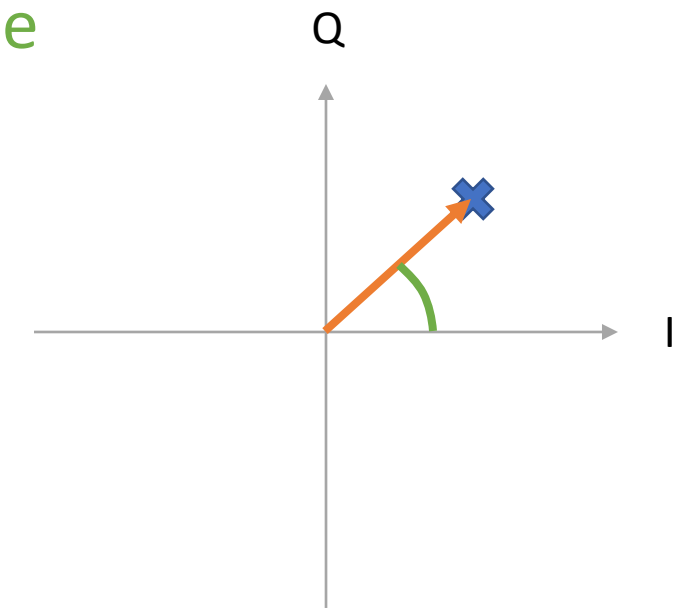
- Amplitude Modulation

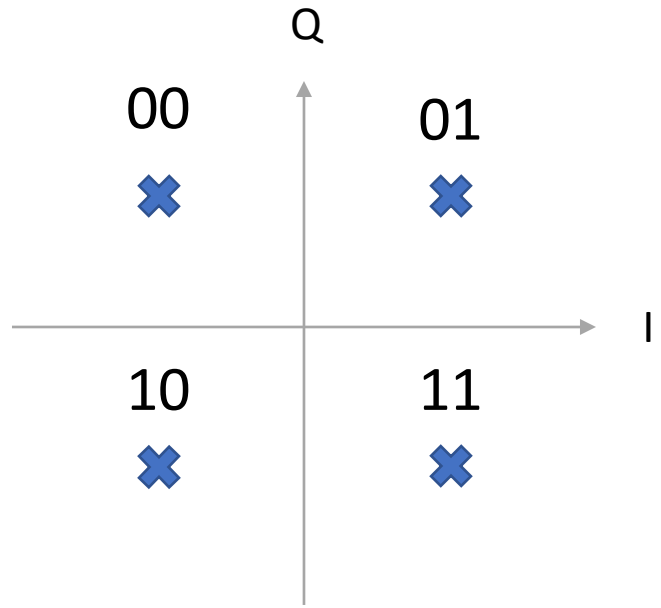


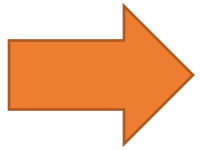
I and Q: Constellation Diagram

Vector:

- Length: Amplitude
- Angle: Phase



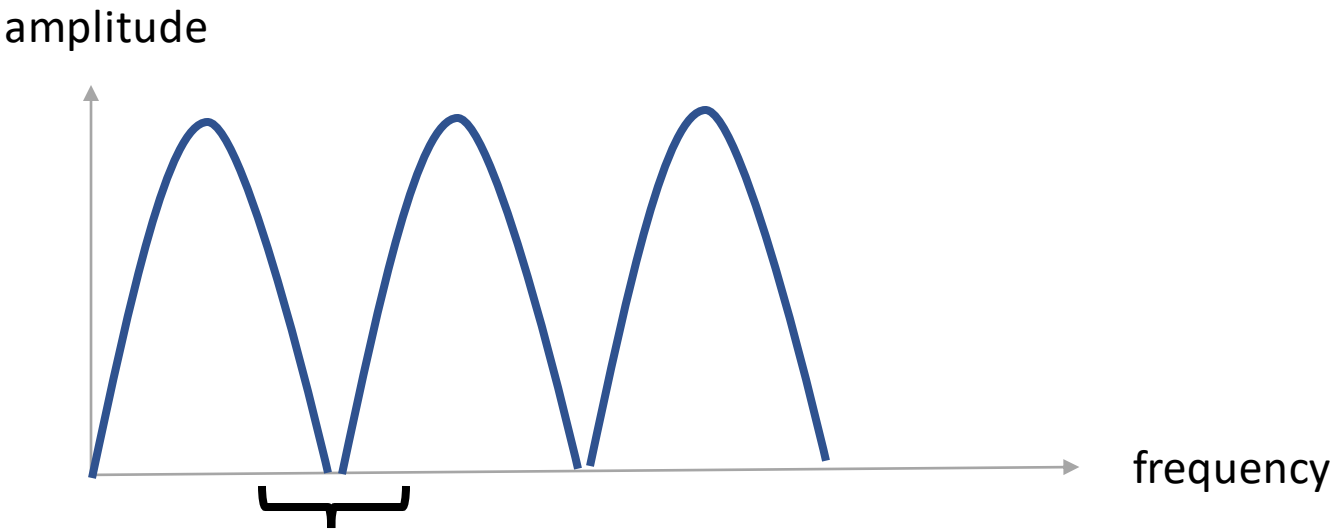




QPSK

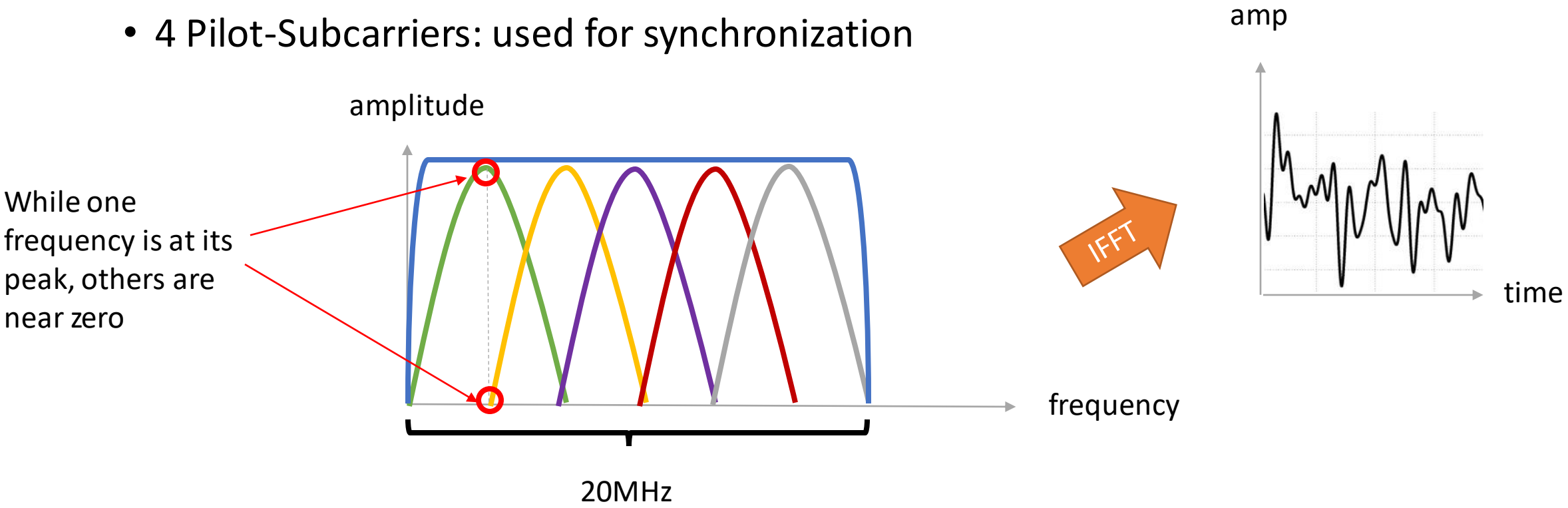
OFDM (Orthogonal Frequency Division Multiplexing)

- Sending multiple carriers at once



OFDM (Orthogonal Frequency Division Multiplexing)

- Subcarriers can be close together **without spacing**
- 52 Subcarriers in total for 802.11a
  - 48 Data-Subcarriers
  - 4 Pilot-Subcarriers: used for synchronization



Why 54MBit?

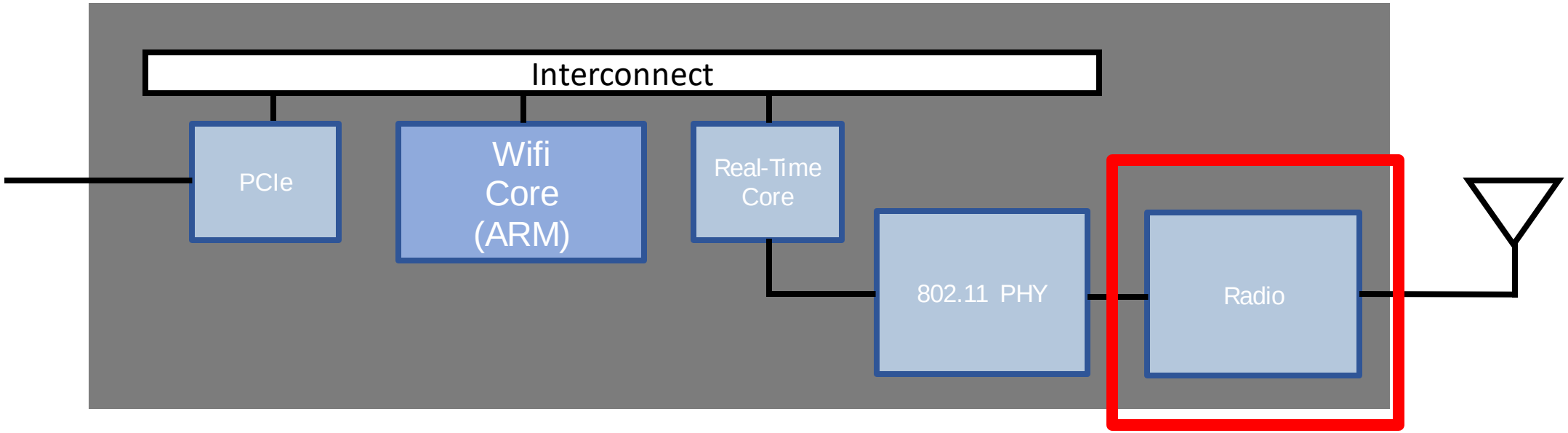
- Data Rate: $\frac{bits\ per\ symbol * Number\ of\ subcarriers}{OFDM\ symbol\ duration} * encoding$:

64-QAM:
6 bits per symbol

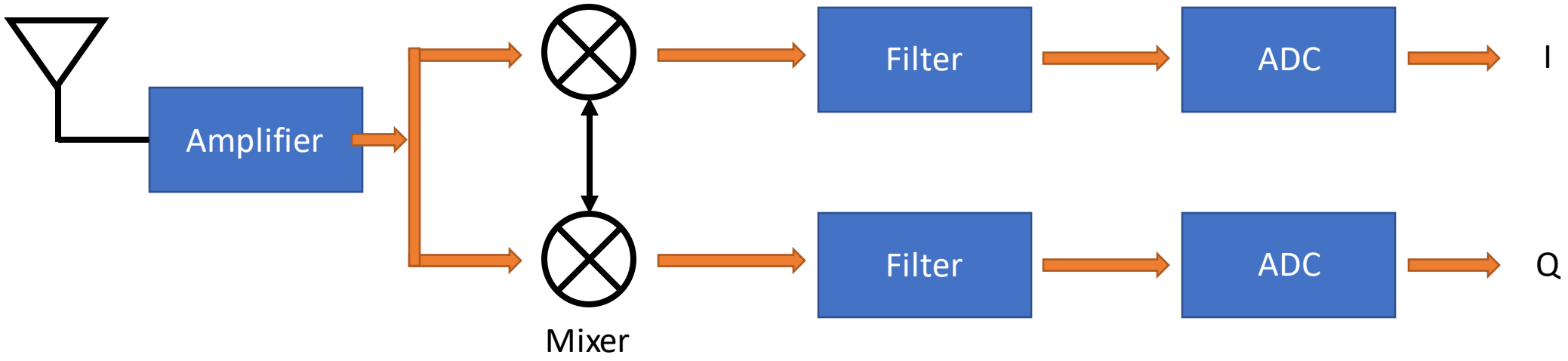
$$\frac{6 * 48}{4 * 10^{-6}} = 72Mbps * \frac{3}{4} = 54Mbps$$

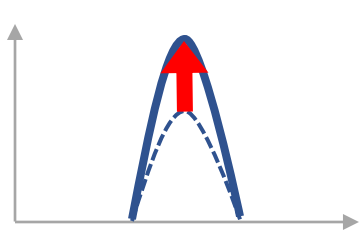
4 µs includes 0.8 µs guard interval

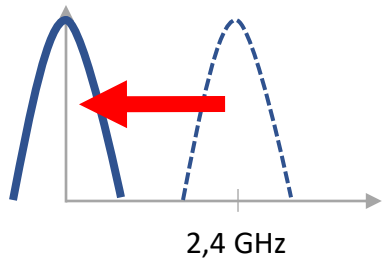
Building blocks of a Wifi Chip



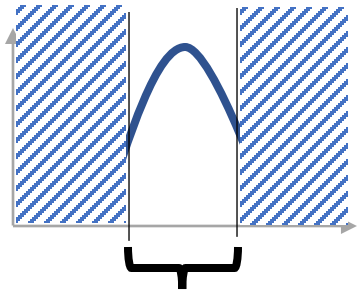
Hardware to get I and Q



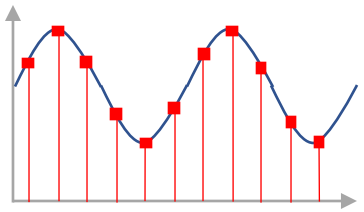




Down conversion



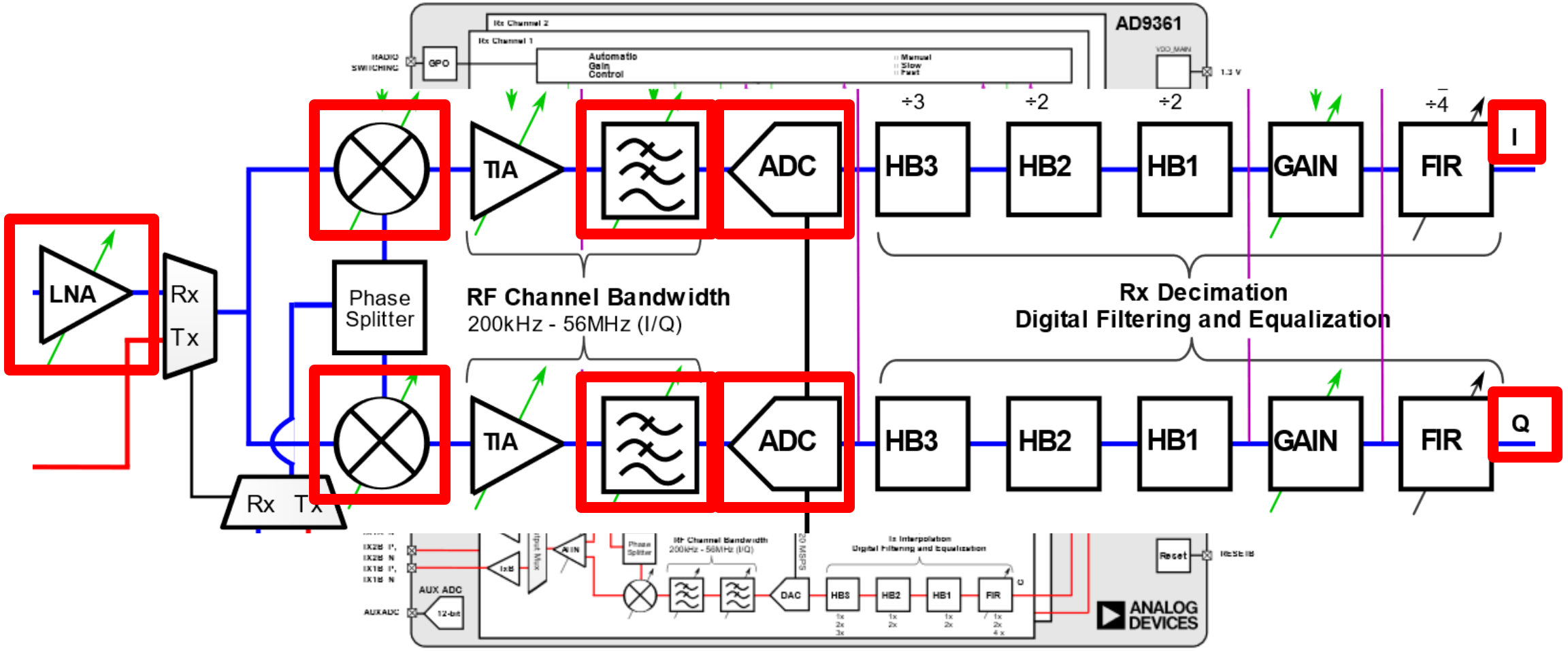
Pass



Sampling

1001101001…

I/Q using SDR: Analog Devices AD9361

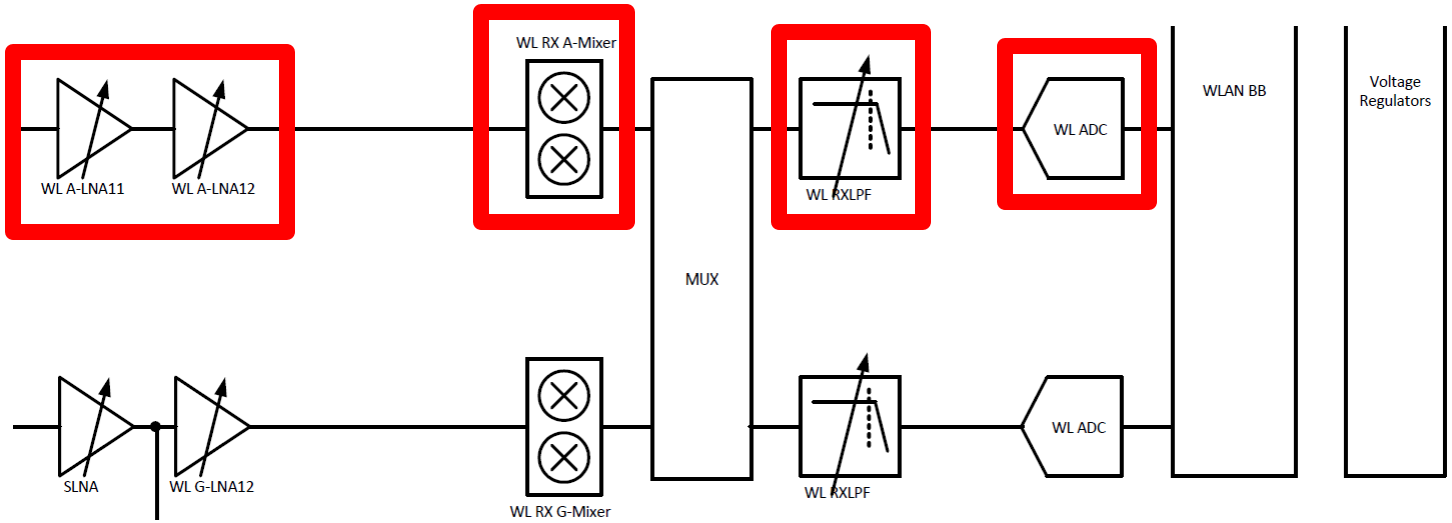


https://wiki.analog.com/resources/eval/user-guides/ad-fmcomms2-ebz/ad9361

I/Q in Broadcom Wifi chips

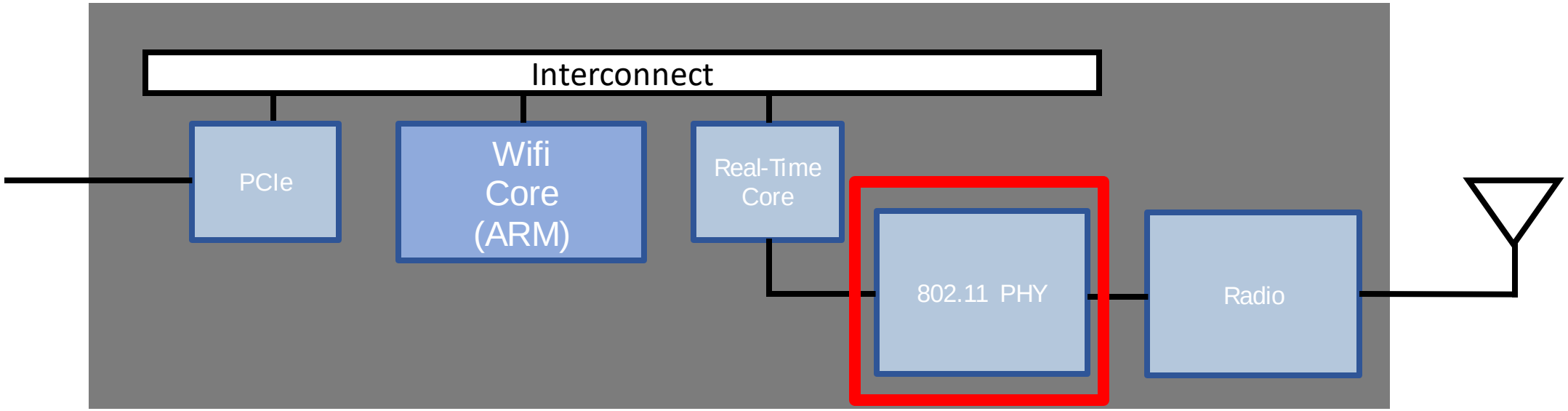
RX

- LNA: Low Noise Amplifier
  - 2.4 GHz shared between BT and WIFI
  - 5GHz dedicated
- LPF: Low Pass Filter



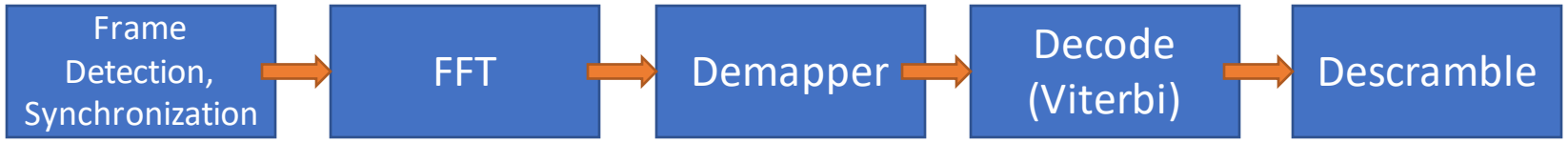
https://www.infineon.com/dgdl/Infineon-CYW4339_Single_Chip_5G_WiFi_IEEE_802.11ac_MAC_Baseband_Radio_with_Integrated_Bluetooth_4.1-DataSheet-v10_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee1e7d367fd

Building blocks of a Wifi Chip



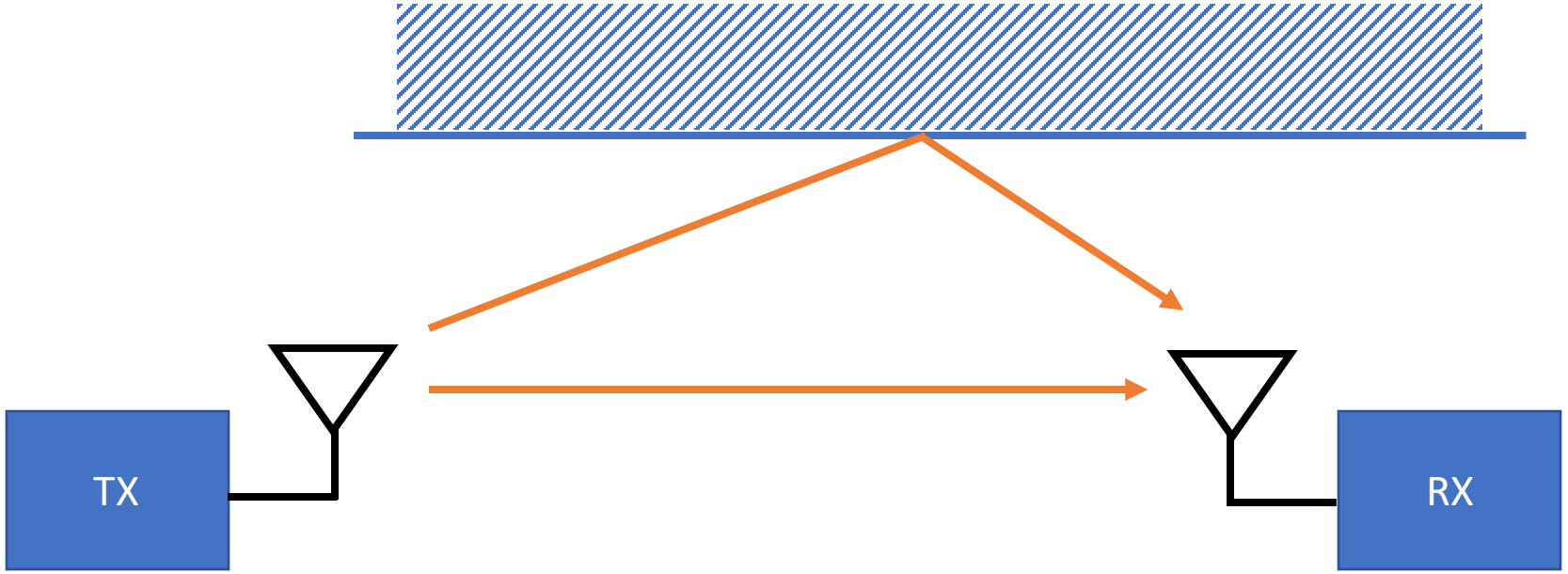
Pipeline IQ and Bit Processing

- RX



- TX

Multipath Effects

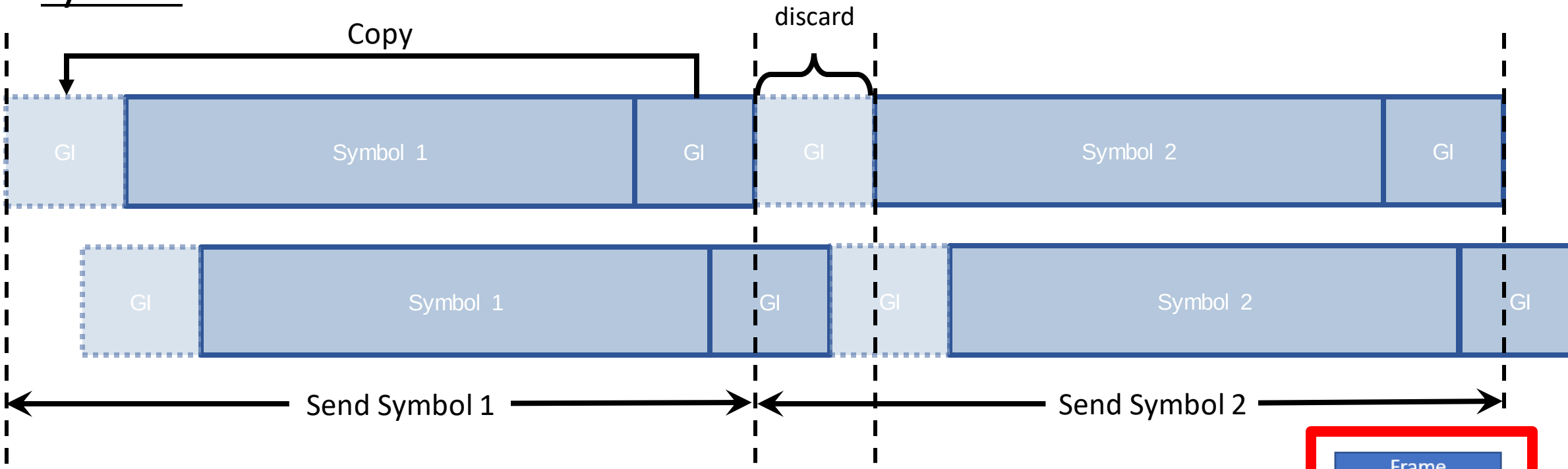


Frame Detection, Synchronization

https://helpfiles.keysight.com/csg/89600B/Webhelp/Subsystems/wlan-ofdm/content/ofdm_80211-overview.htm

Multipath Effects – Guard Interval

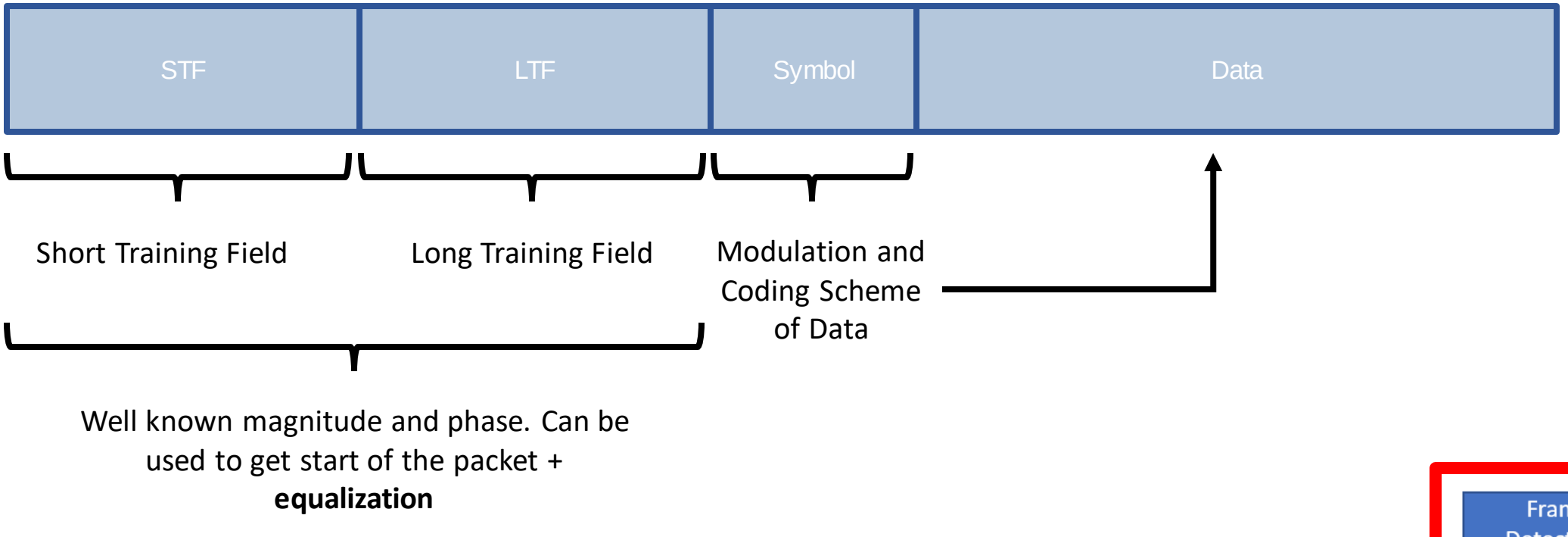
- Guard Interval or Cyclic Prefix protects against interference with next symbol



https://helpfiles.keysight.com/csg/89600B/Webhelp/Subsystems/wlan-ofdm/content/ofdm_80211-overview.htm

Frame Format with Preamble

- Frames begin with a Preamble (here shown for OFDM in 802.11a)

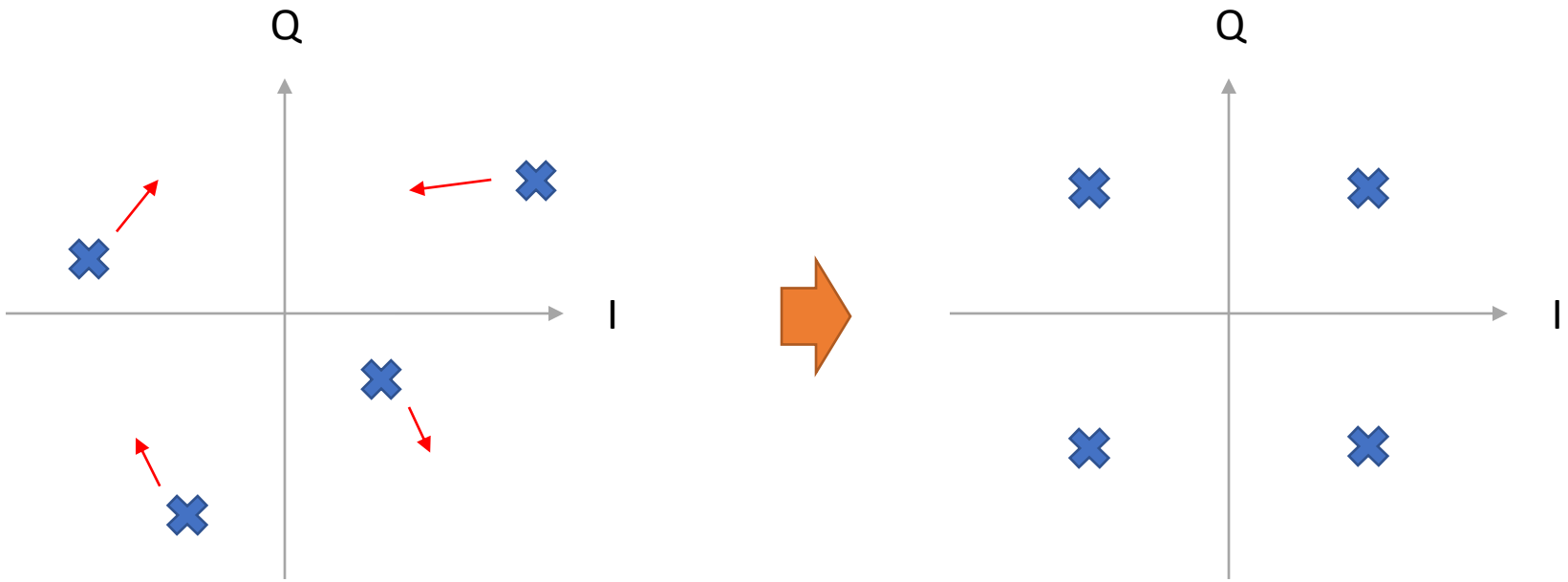


Frame Detection, Synchronization

https://helpfiles.keysight.com/csg/89600B/Webhelp/Subsystems/wlan-ofdm/content/ofdm_80211-overview.htm

Preamble - Equalization

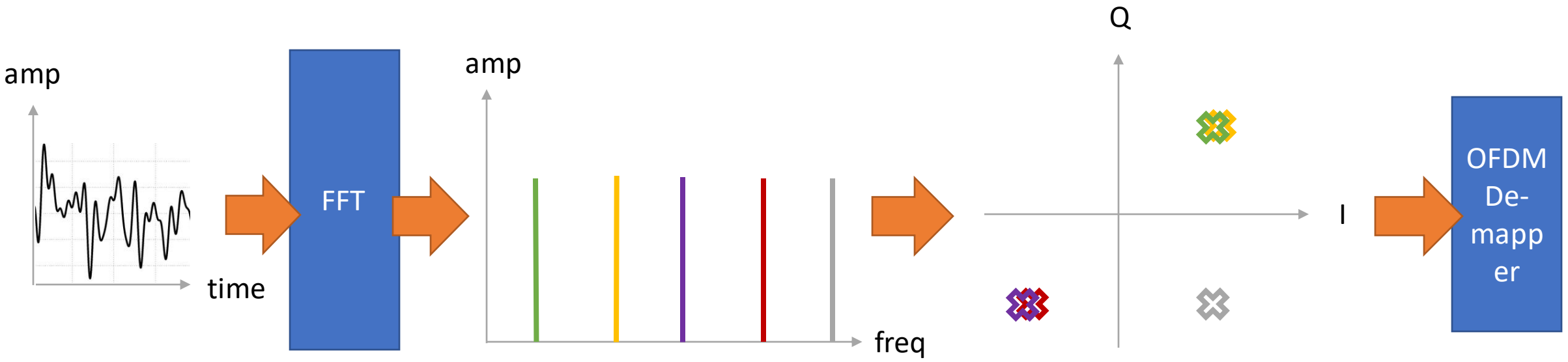
- Fix amplitude and phase offsets introduced by channel

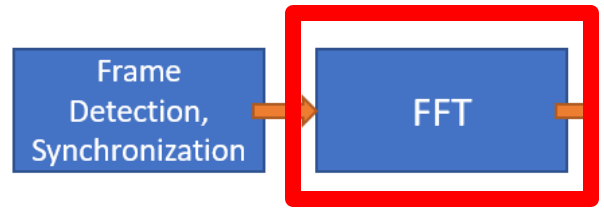


Frame Detection, Synchronization

FFT

- Use FFT to get phase and amplitude for each sub-carrier

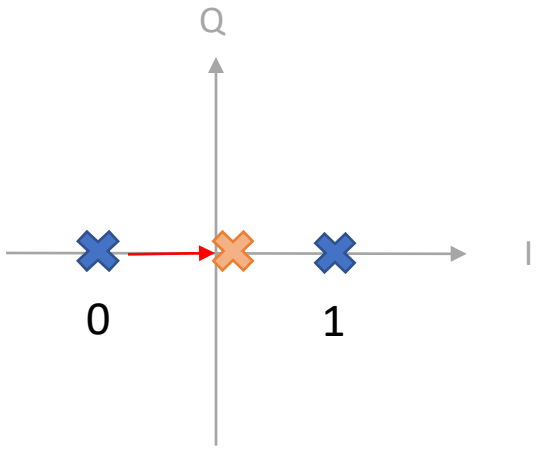




Demapper and Decoder

In case of errors, how can we know which bits are wrong?

1. Demap: Create probabilities (using Viterbi) of **how likely** it is that a **symbol is a certain value**
2. Decoder: Use probabilities to **figure out which bit is wrong** in case parity bit does not match

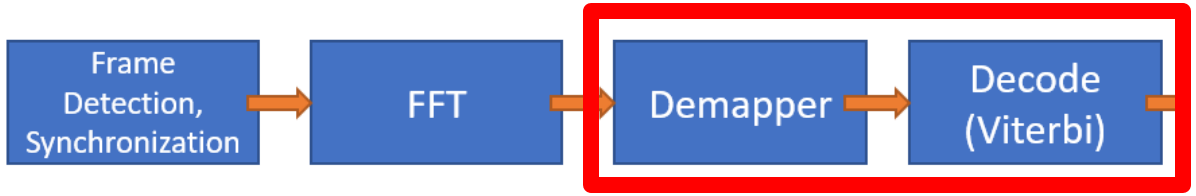


1, 0, 0, 1, X (parity)

0.98, 0.02, 0.52, 0.96

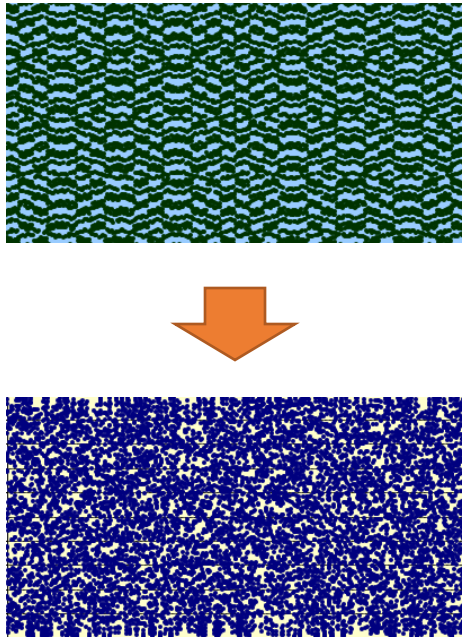
1, 0, 1, 1, X (parity)

1, 0, 0, 1

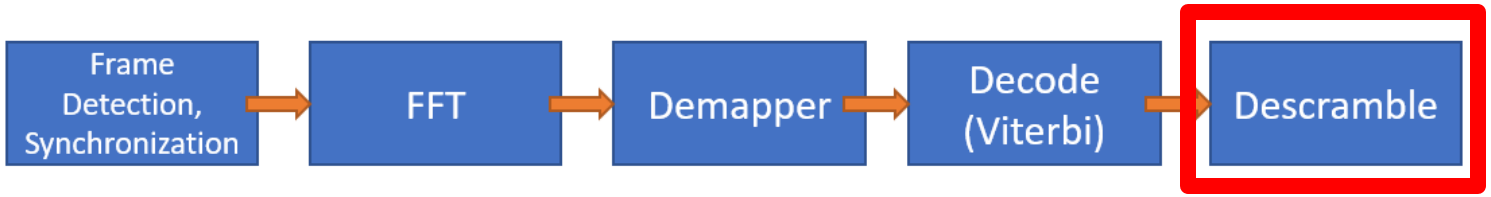


Descramble

- Reverse:
  - Create **even number** of **zeros and ones**
  - Avoid **long runs of zeros or ones**
    - spread power across spectrum
    - avoid interference with other channels
- Using LFSRs: **Linear Feedback Shift Registers**

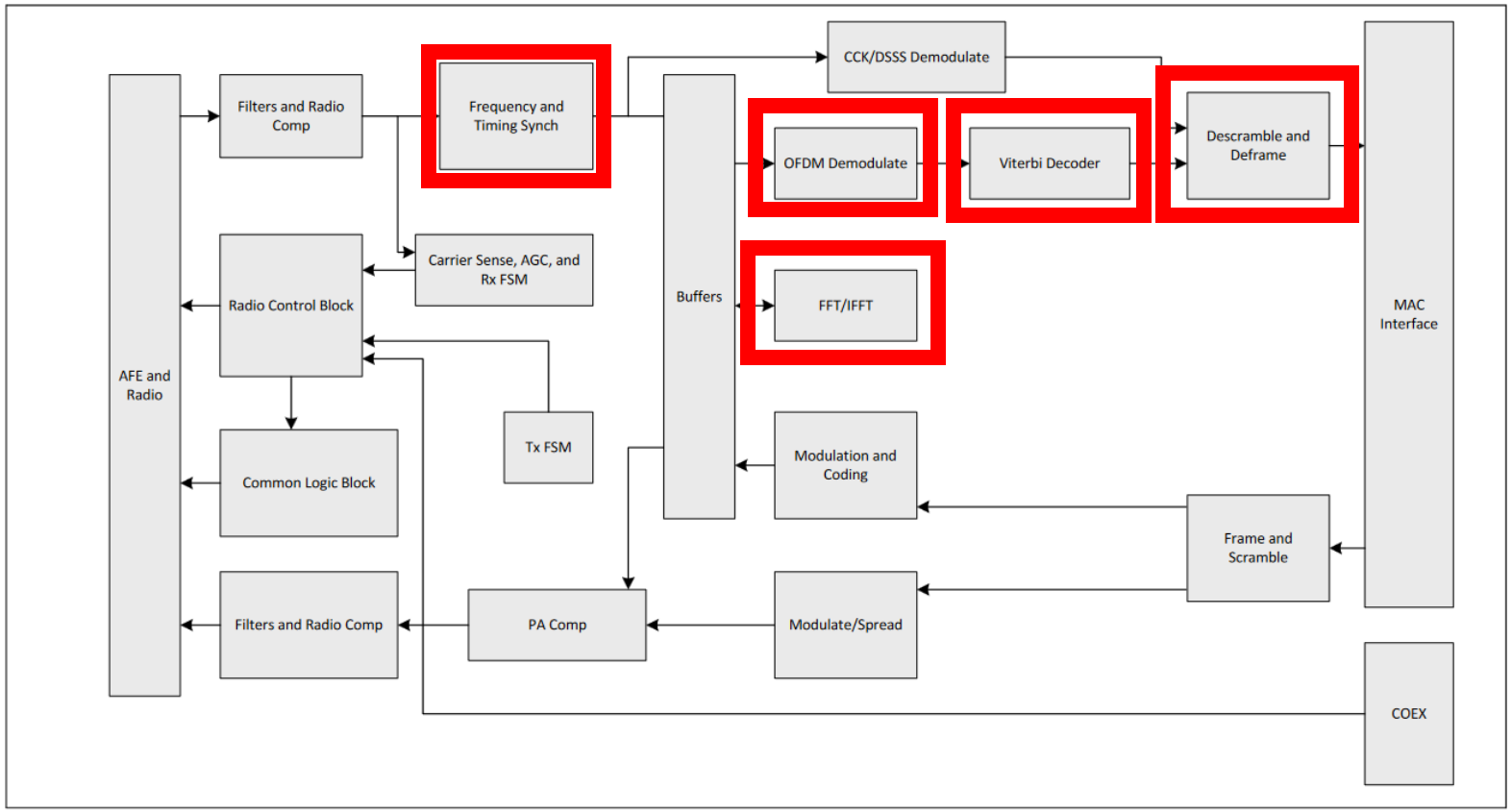


https://excelunusual.com/an-animated-linear-feedback-shift-register-lfsr-as-a-pseudo-random-pattern-generator-in-excel-2003-part4/



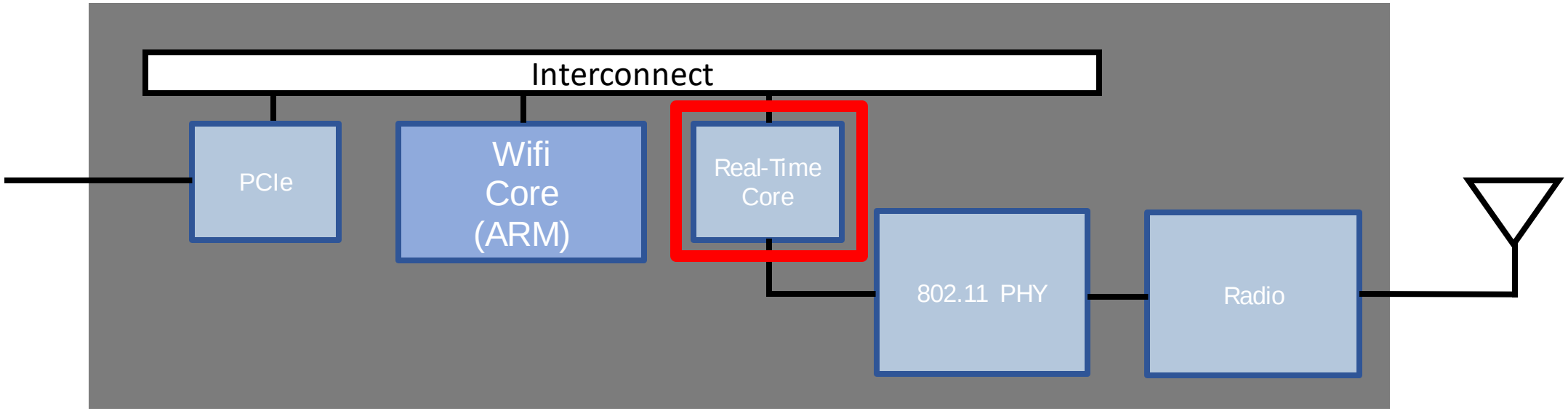
Pipeline IQ and Bit Processing

Figure 24. WLAN PHY Block Diagram



https://www.infineon.com/dgdl/Infineon-CYW4339_Single_Chip_5G_WiFi_IEEE_802.11ac_MAC_Baseband_Radio_with_Integrated_Bluetooth_4.1-DataSheet-v10_00-EN.pdf?fileId=8ac78c8c7d0d8da4017d0ee1e7d367fd

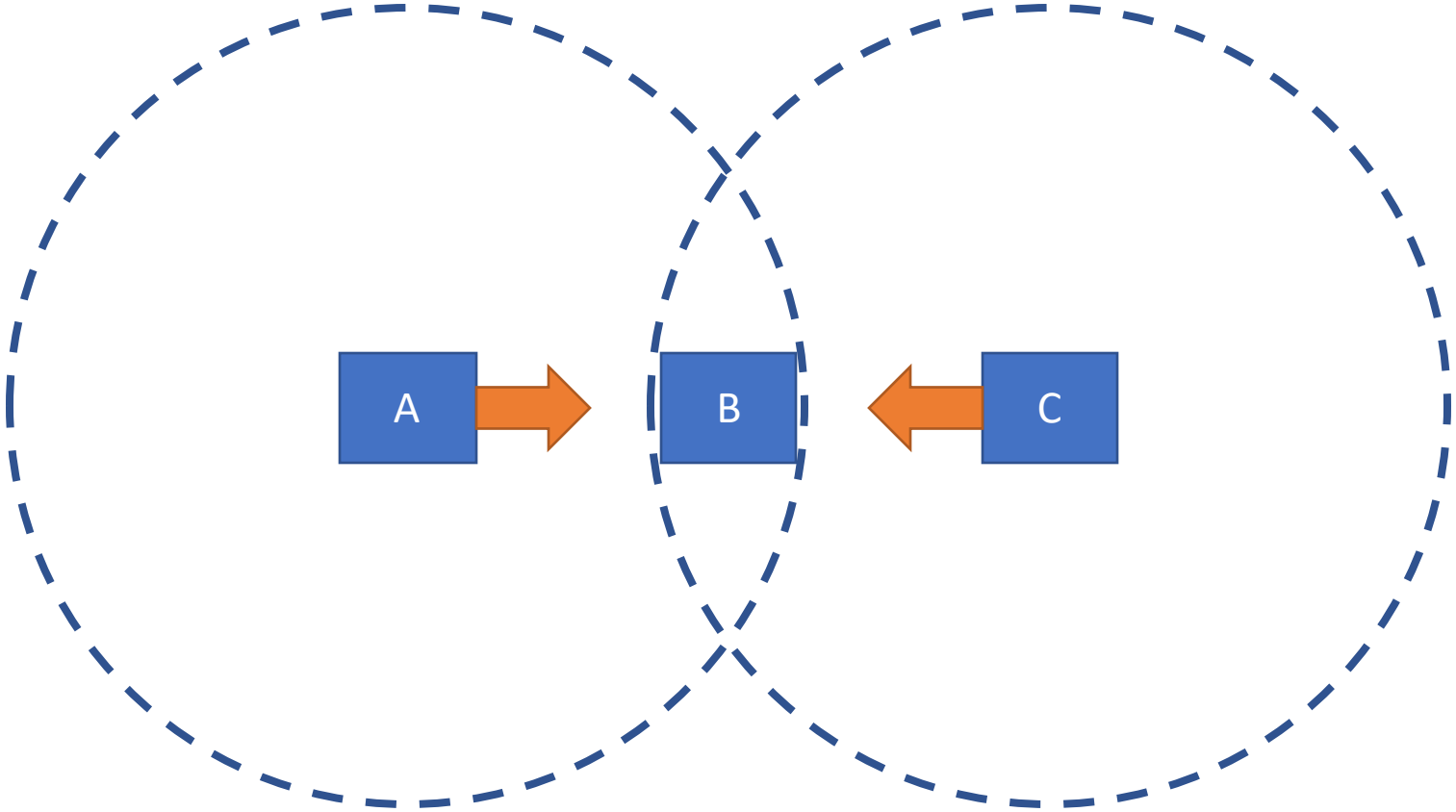
Building blocks of a Wifi Chip



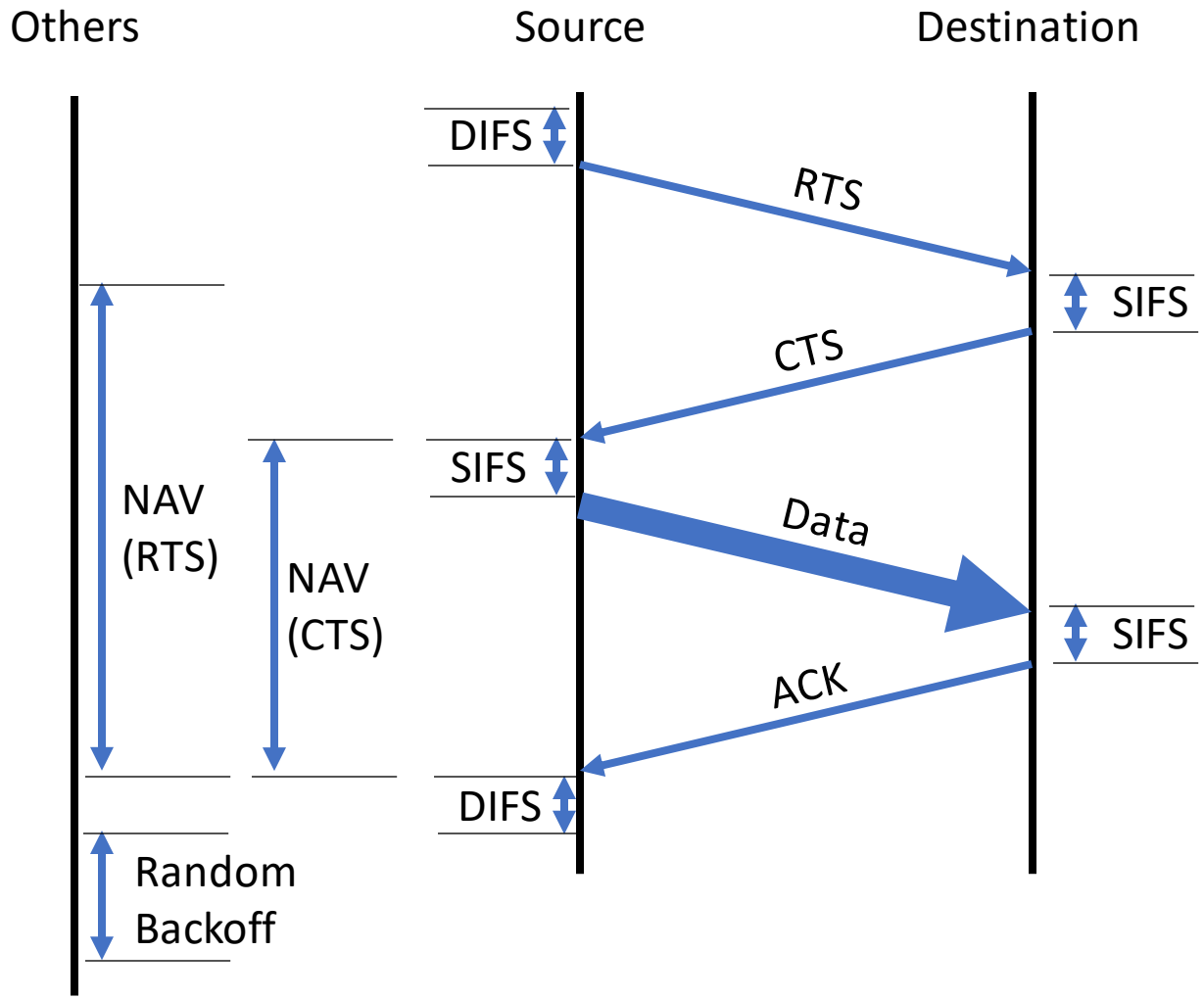
Can we send and Receive at the same time?

- No! Only sending or receiving possible at the same time with one transceiver ➔ Shared medium
- Ethernet: **Carrier-sense multiple access with collision detection (CSMA/CD)**
- Wifi: **Carrier-sense multiple access with collision avoidance (CSMA/CA)**

Hidden Terminal Problem

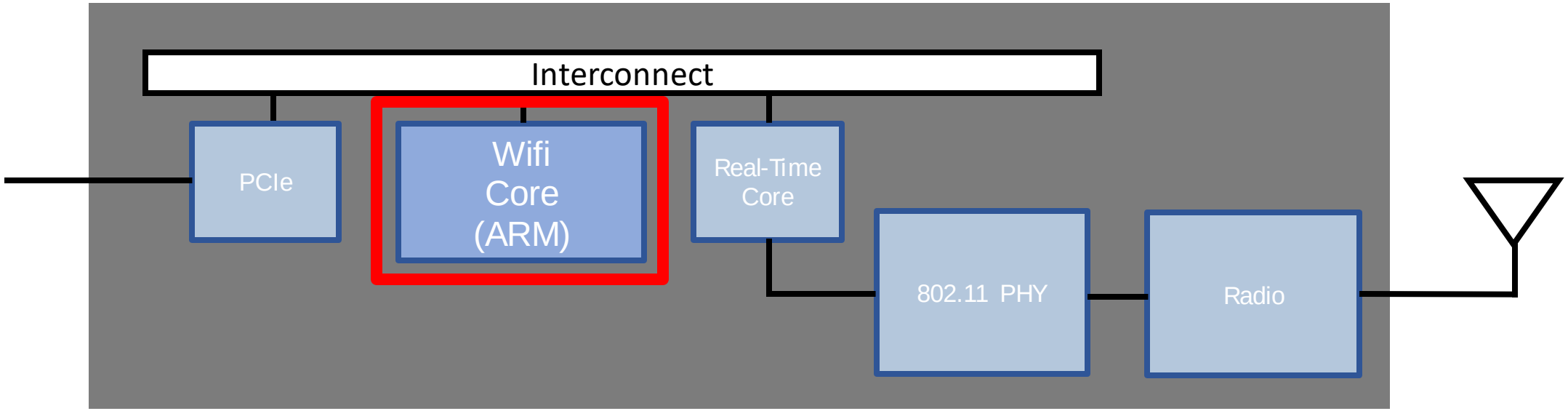


DCF: Distributed Coordination Function



DCF: Distributed Coordination Function
DIFS: DCF Interframe Space
SIFS: Short Interframe Space
RTS: Request To Send
CTS: Clear To Send
ACK: Acknowledgement
NAV: Network Allocation Vector

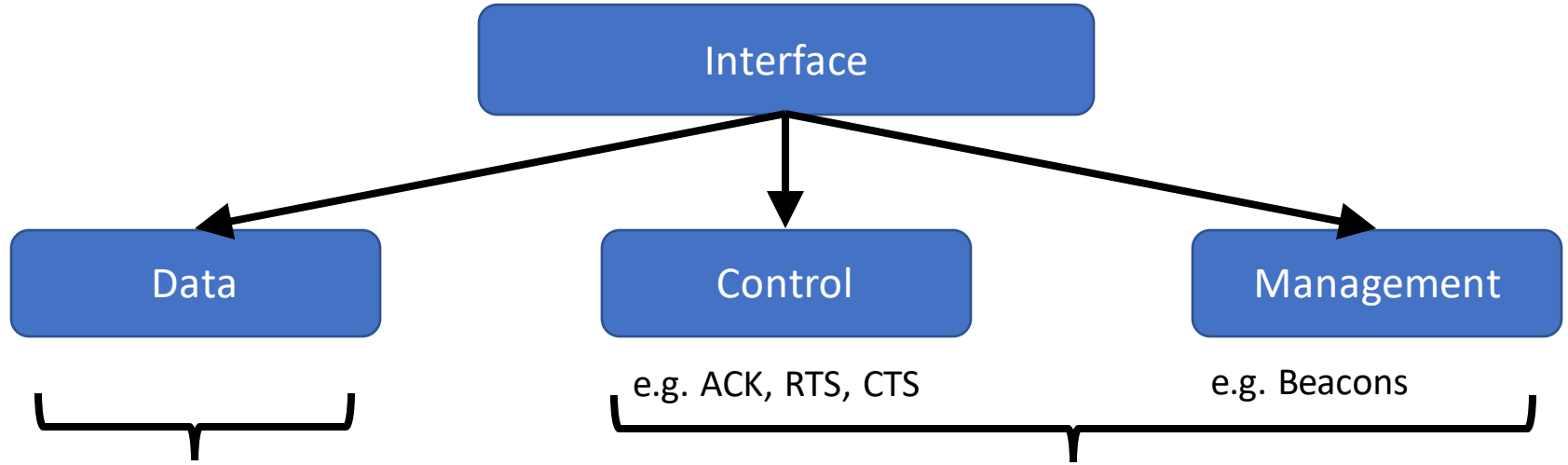
Building blocks of a Wifi Chip



What is the MAC layer responsible for?

- Frame aggregation and fragmentation
- Scanning
- Authentication + Association
- Power Saving
- Roaming
- Checksums

Frame Types

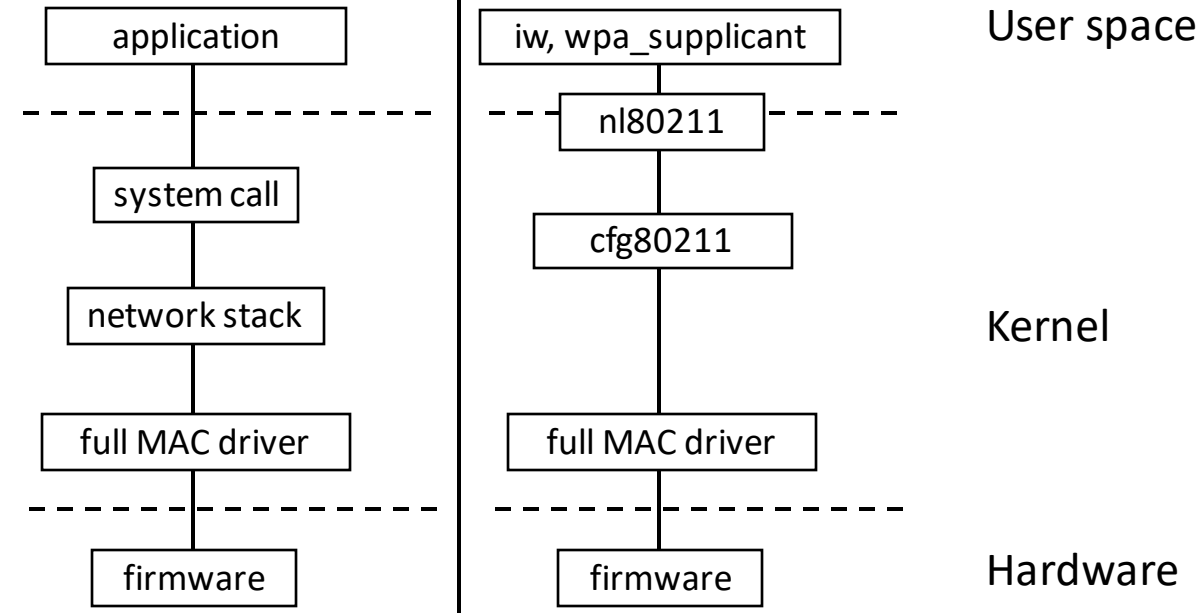


Only this is shown in Linux

Used by Wifi firmware

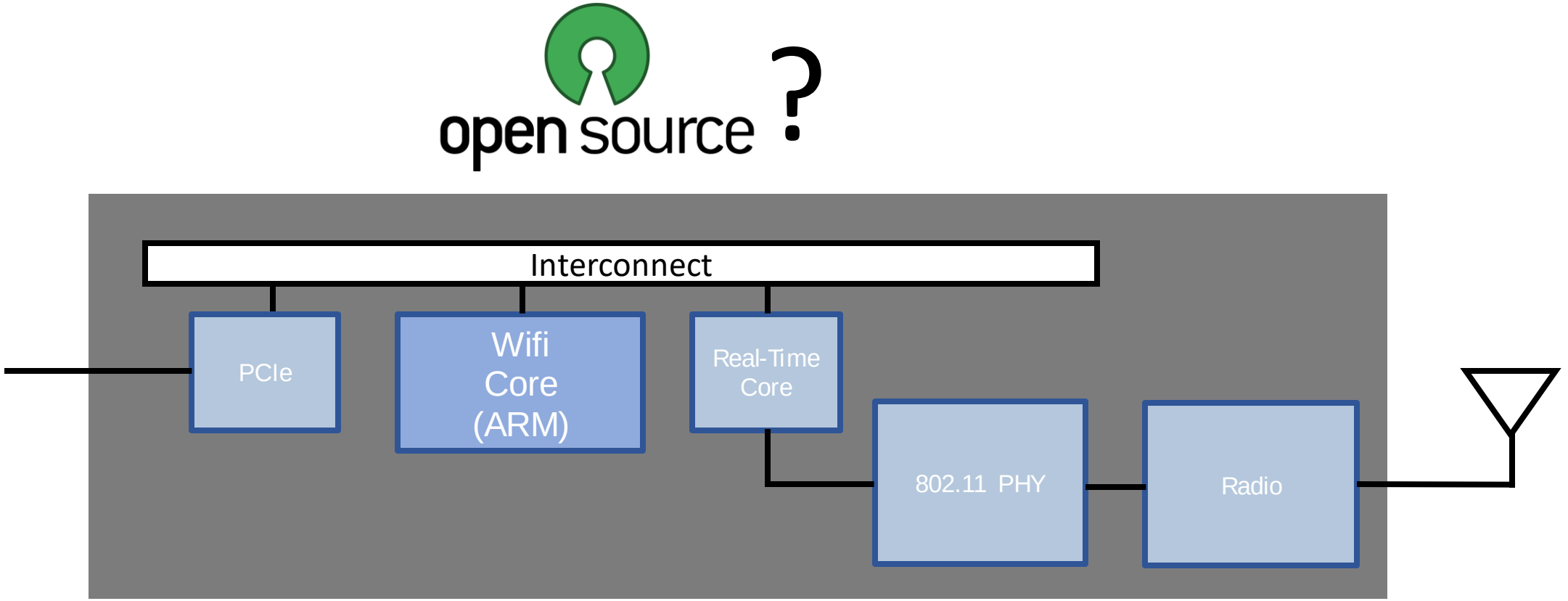
Wireless Data in Linux

Data and management/configuration move differently thorugh the Linux kernel



Flow of data (left) and management/configuration (right) through the Linux kernel

Building blocks of a Wifi Chip



Open Source Firmware - Problems

- HW initialization: HW Registers are not known
- Primitives for Sending and Receiving packets
- Tasks or processes need to be understood to run code independent of main loop
- Control “real time” part of FW
  →needed DCF: Sending ACKs (done by HW in ESP8266)
- HW packet filtering
- License: needs “clean room” documentation to develop FW which could be GPL licenced and be usable in Linux Kernel

Thank You!

Q&A

Links

- Analog Devices Course
  - https://www.analog.com/en/resources/analog-dialogue/articles/rf-signal-chain-discourse.html
  - https://www.analog.com/en/resources/analog-dialogue/articles/rf-signal-chain-discourse-part-2-essential-building-blocks.html
- I/Q Data
  - http://whiteboard.ping.se/SDR/IQ
  - https://towardsdatascience.com/mind-your-is-and-q-s-the-basics-of-i-q-data-d1f2b0dd81f4
- https://wirelesspi.com/
- https://www.ni.com/en/support/documentation/supplemental/15/labview-communications-802-11-application-framework-1-1-white-pa.html
- https://www.tek.com/en/documents/primer/wi-fi-overview-80211-physical-layer-and-transmitter-measurements
- Explanation videos on various digital signal processing algorithms and methods: https://www.youtube.com/@iain_explains
- SDR
  - Youtube Introduction Series using HackRF One: https://www.youtube.com/playlist?list=PL75kaTo_bJqmw0wJYw3Jw5_4MWBd-32IG
  - https://ajoo-github-blog-old.pages.dev/
- AD9361 datasheet: https://www.farnell.com/datasheets/2007082.pdf
- Projects
  - https://github.com/open-sdr/openwifi
  - https://github.com/esp32-open-mac/esp32-open-mac
  - Modify Broadcom Wifi Chip firmware: https://nexmon.org
- https://mcsindex.com/
- Open Source MATLAB alternative: https://octave.org/