LTE Cell Search Theory

Table of Contents

Table of Figures

Cyclic Prefix Correlation

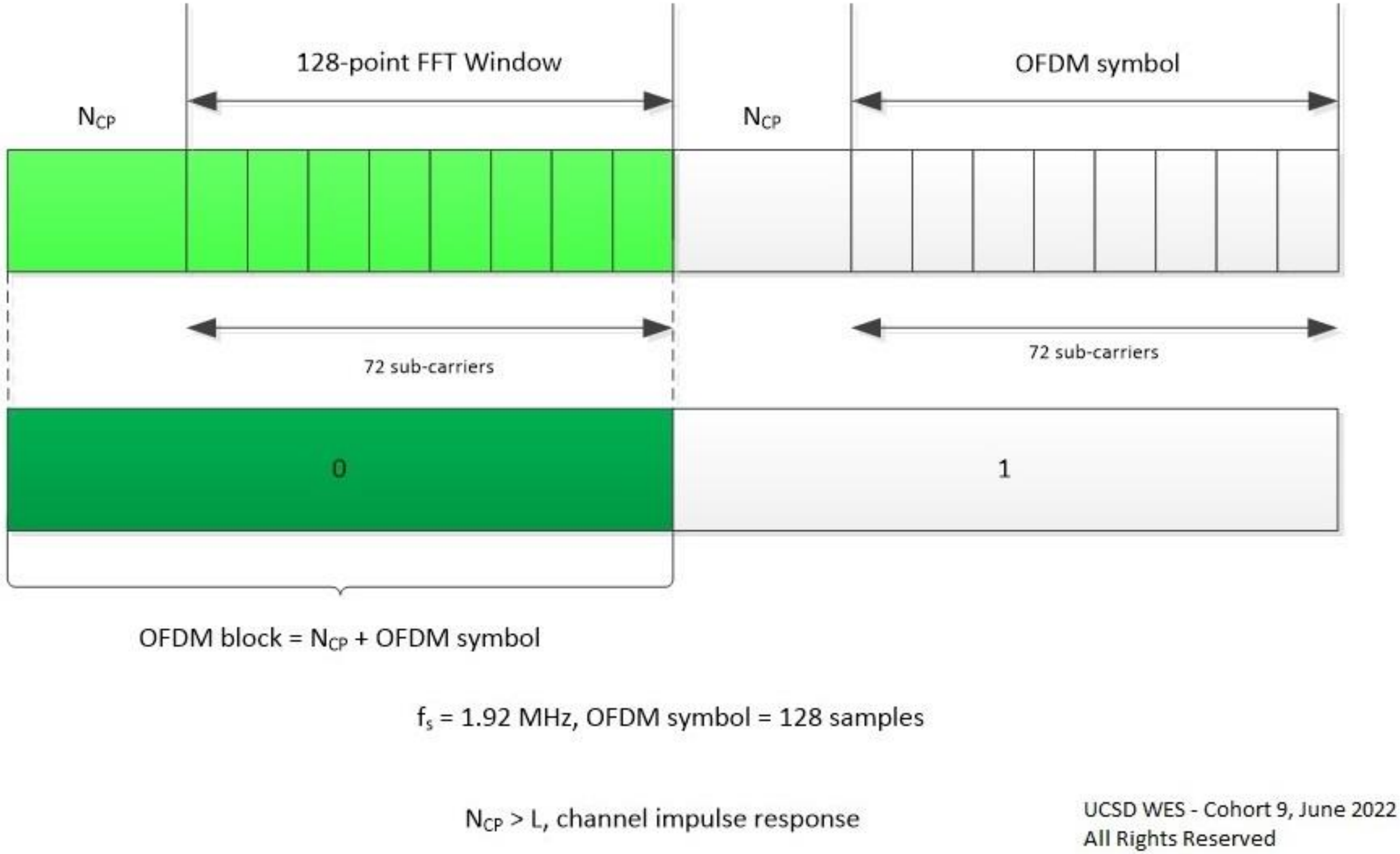


Figure 1: CP Correlation, FFT Window, OFDM Symbol, OFDM Block

LTE Synchronization Signals (PSS and SSS)

There are two downlink synchronization signals used by the User Equipment to obtain the cell identity and frame timing.

- Primary synchronization signal (PSS)
- Secondary synchronization signal (SSS)

The division into two signals is aimed to reduce the complexity of the cell search process.

https://www.mathworks.com/help/lte/ug/synchronization-signals-pss-and-sss.html

https://www.mathworks.com/help/wireless-hdl/ug/lte-hdl-cell-search.html

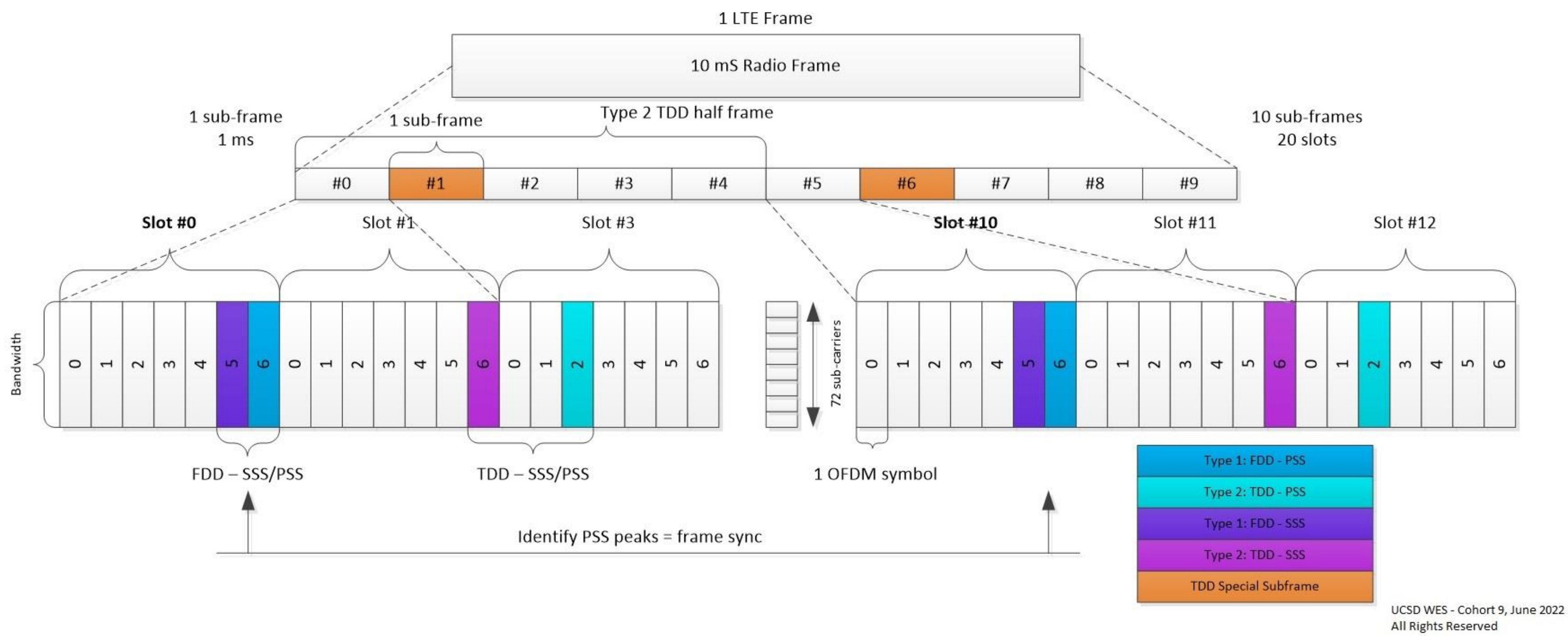


Figure 2: LTE Frame Details: Relative Symbol Position of SSS and PSS in Frame | Sub-frame | Slots.

PSS Peak Correlation

https://www.mathworks.com/help/lte/ug/synchronization-signals-pss-and-sss.html#bt0y_j2-1

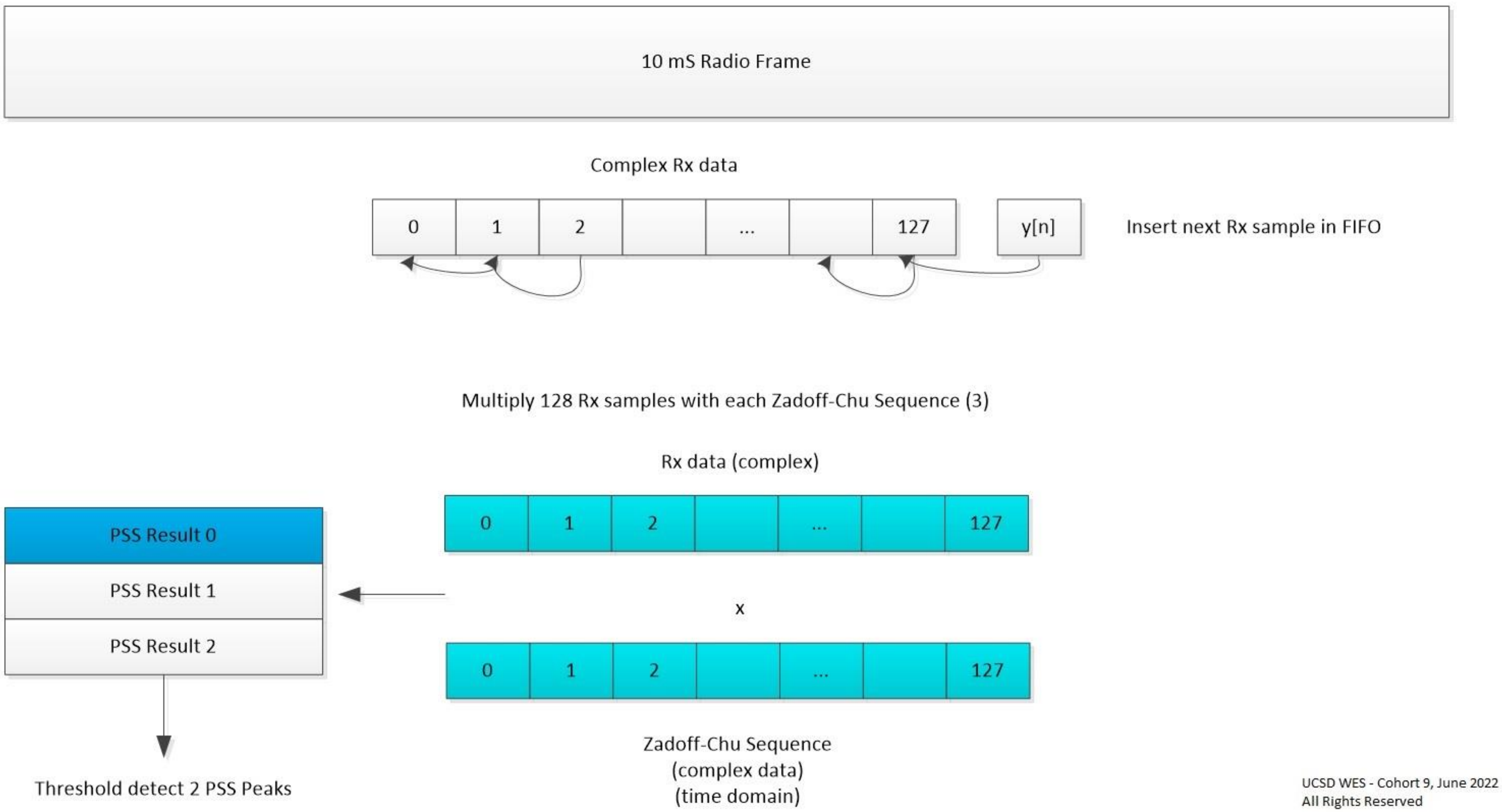


Figure 3: Calculations for Primary Synchronization Signal (PSS) Detection at Receiver.

PSS Correlation Peak Detection

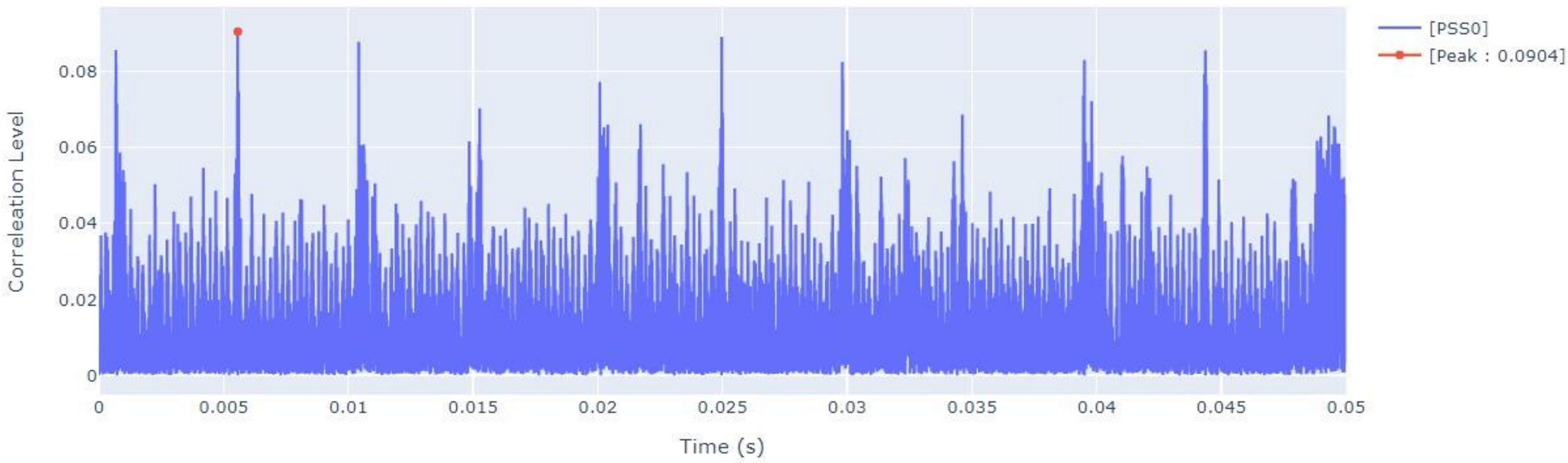


Figure 4: PSS Peak Detection: 50 ms Capture: 5 LTE Frames – 10 PSS Correlation Peaks

PSS Process

LTE data carried in frames (10 ms) and sub-frames (10) divided into 20 slots - structured for synchronization

Goal: Identify PSS and demodulate (extract info for Cell ID)

Don't know which type of frame. Need to correlate input with known Zadoff-Chu sequences - identify peaks (2 symbol locations within frame)

Programmable Logic (PL) Implementation:

- Complex Multiply: Rx input sequence * PSS (time domain)
- Identify 2 peaks over 10 slots (1 frame)
  - Identify known sequence in Rx data - 1 of 3 Zadoff-Chu sequences
- Keep searching for pair of peaks to maintain sync using threshold detect - maintain sync
- Part 1 of getting Physical Cell ID (504 Unique IDs)

Primary synchronization signal (PSS): linked to the cell identity within the group.

SSS Sequence Calculations and Correlation

https://www.mathworks.com/help/lte/ug/synchronization-signals-pss-and-sss.html#bt03428-1

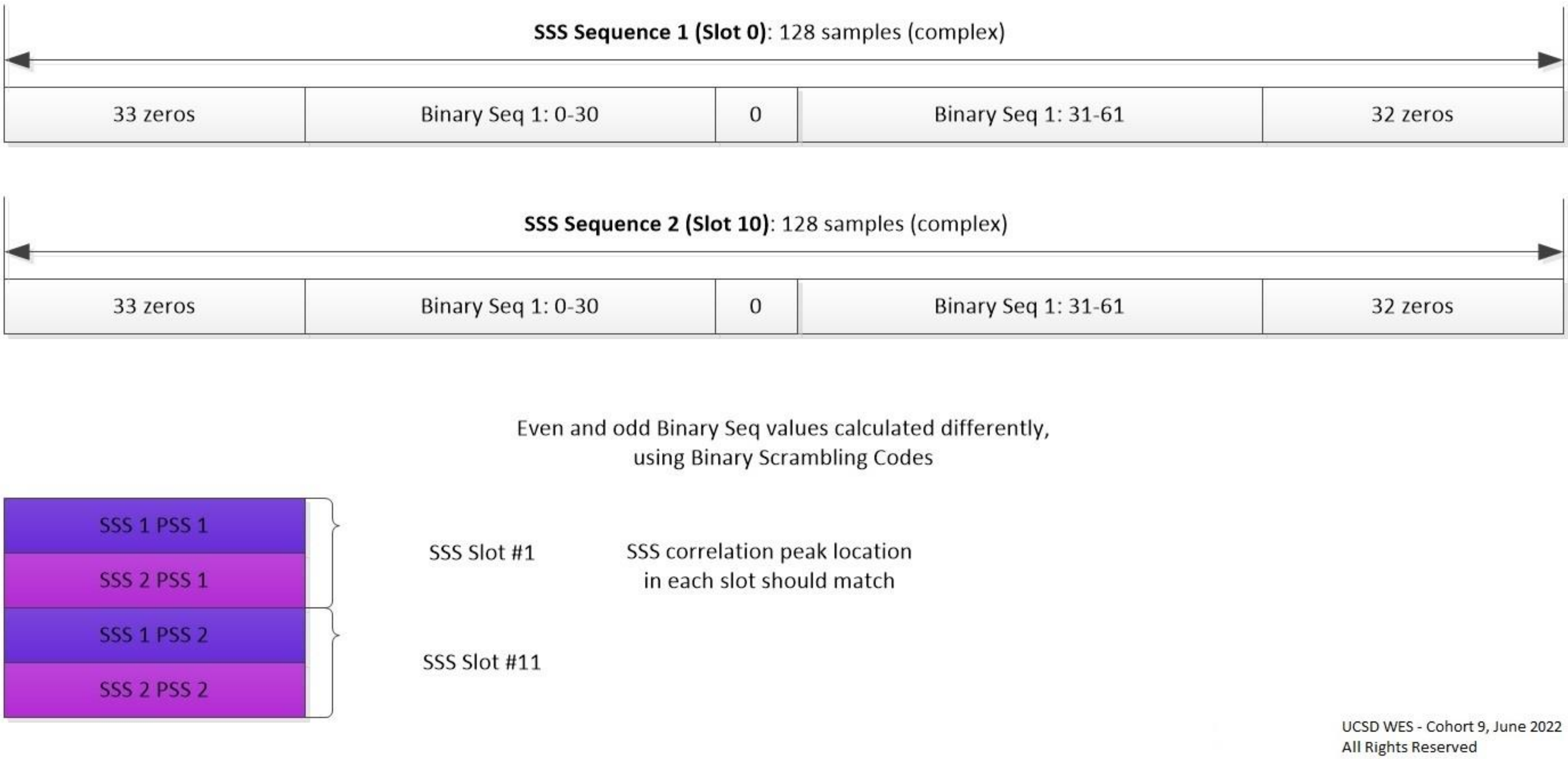


Figure 5: SSS Sequences: SSS Correlation.

SSS Process

LTE data carried in frames (10 ms) and sub-frames (10) divided into 20 slots - structured for synchronization
Goal: Identify SSS and demodulate (extract info for Cell ID)

Know location of PSS in frame. 2 frame types determine relative location of SSS sequences.
Programmable Logic (PL) Implementation:

- Start with 2 locations of PSS and take 128 point fft (and fft shift to put samples in order for analysis)
- Analyze PSS locations @ **slot 0 and slot 10**
- Generate 3 binary sequences: s[31],c[31],z[31] - polynomial based on Galois field
- Iterate over the set of 168 cell identity groups, NID[(1)].
    - Create binary sequences with binary scrambling codes - depend on NID[(2)]
- Generate 2 sequences length 62 from binary sequence: even and odd indices calculations different
- Insert 2 length 62 sequences in 128 samples centered around 0 at middle (DC) (See Figure 5 above)
- Iterate over 128 samples @ PSS locations: Multiply and accumulate input samples with generated sequences (separate real and imaginary components): 2 complex inputs from 2 PSS locations
- Fill 4 complex data buffers, length 168, with accumulated values:
    - 2 sequences for **slot 0** and 2 sequences for **slot 10**
    - 2 sequences since SSS interleave improve sync performance
- Analyze the 4 complex data buffers (only need 2 since redundant)
    - Identify peaks and determine physical layer identity group (location of peak is group -> Cell ID

Secondary Synchronization Signal (SSS): linked to the cell identity within the group + physical layer cell identity group.

PSD of LTE Signal Received

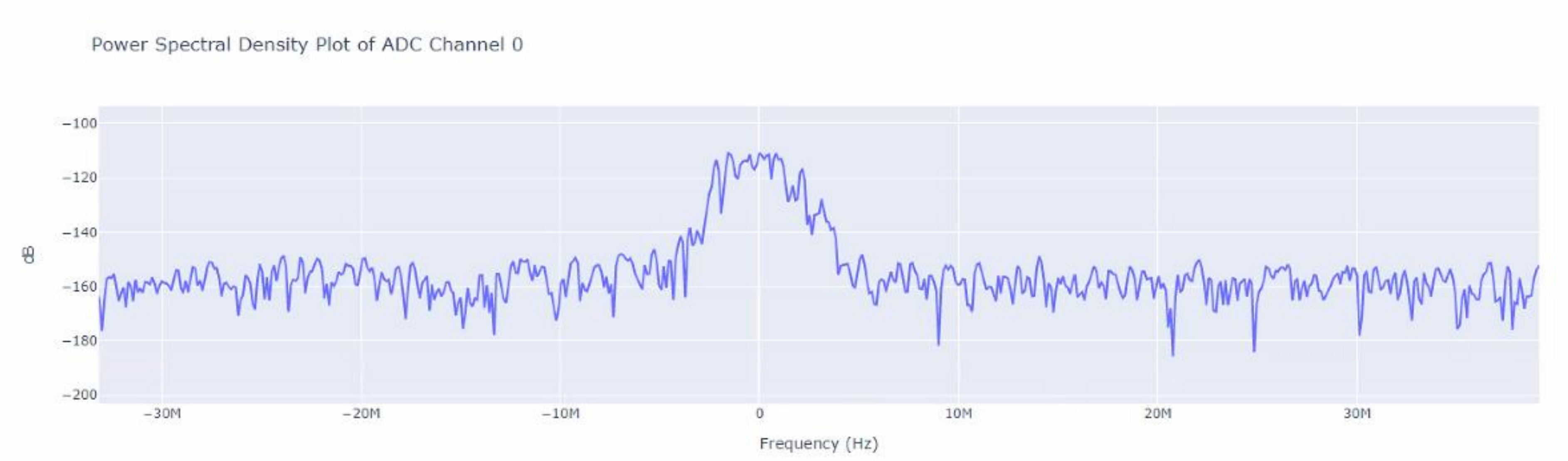


Figure 6: Received Power Spectral Density (PSD) of Recorded LTE Signal Transmitted from the Pluto.

Cell ID Detected from MATLAB IP

```
[74]: val = matlabIP.read(0x100) #extdatasel_in_Data
print("Ext Data Select 0x%X : " % val)
val = matlabIP.read(0x108) # start_in_Data
print("Start 0x%X : " % val)
val = matlabIP.read(0x110) #ncellid_Data
print("NCell ID 0x%X : " % val)
val = matlabIP.read(0x118) #freqest_Data
print("Freq Est. 0x%X : " % val)
val = matlabIP.read(0x11C) #celldetected_Data
print("Cell Detected 0x%X : " % val)
val = matlabIP.read(0x120) #cellsearchdone_Data
print("Cell Search Done 0x%X : " % val)
val = matlabIP.read(0x138) #mibdetected_Data
print("MIB Detected 0x%X : " % val)
val = matlabIP.read(0x13C) #miberror_Data
print("MIB Error 0x%X : " % val)

Ext Data Select 0x1 :
Start 0x0 :
NCell ID 0x4C :
Freq Est. 0x52 :
Cell Detected 0x1 :
Cell Search Done 0x1 :
MIB Detected 0x1 :
MIB Error 0x0 :
```

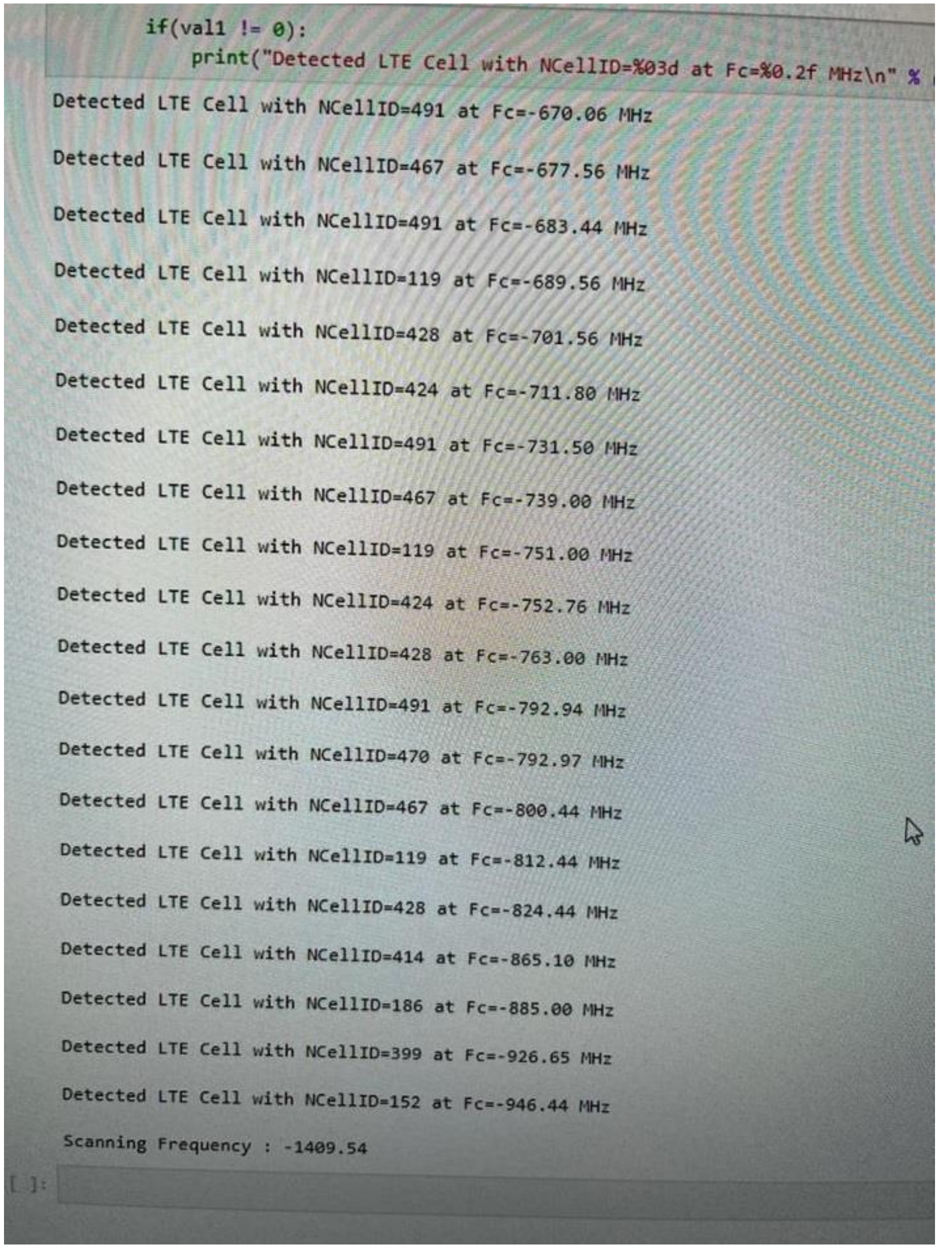
Frequency Estimation from Custom IP

```
[77]: val = vitisIP.read(0x10) #Data signal of OUT_F
print("Data Signal OUT_F %F : " % (val/1000.0), end='\r')

Data Signal OUT_F 509.182000 :
```

Figure 7: **Jupyter Notebook Output from Loopback Configuration**

Detected Cell ID



```
        if(val1 != 0):
            print("Detected LTE Cell with NCellID=%03d at Fc=%0.2f MHz\n" %
Detected LTE Cell with NCellID=491 at Fc=-670.06 MHz

Detected LTE Cell with NCellID=467 at Fc=-677.56 MHz

Detected LTE Cell with NCellID=491 at Fc=-683.44 MHz

Detected LTE Cell with NCellID=119 at Fc=-689.56 MHz

Detected LTE Cell with NCellID=428 at Fc=-701.56 MHz

Detected LTE Cell with NCellID=424 at Fc=-711.80 MHz

Detected LTE Cell with NCellID=491 at Fc=-731.50 MHz

Detected LTE Cell with NCellID=467 at Fc=-739.00 MHz

Detected LTE Cell with NCellID=119 at Fc=-751.00 MHz

Detected LTE Cell with NCellID=424 at Fc=-752.76 MHz

Detected LTE Cell with NCellID=428 at Fc=-763.00 MHz

Detected LTE Cell with NCellID=491 at Fc=-792.94 MHz

Detected LTE Cell with NCellID=470 at Fc=-792.97 MHz

Detected LTE Cell with NCellID=467 at Fc=-800.44 MHz

Detected LTE Cell with NCellID=119 at Fc=-812.44 MHz

Detected LTE Cell with NCellID=428 at Fc=-824.44 MHz

Detected LTE Cell with NCellID=414 at Fc=-865.10 MHz

Detected LTE Cell with NCellID=186 at Fc=-885.00 MHz

Detected LTE Cell with NCellID=399 at Fc=-926.65 MHz

Detected LTE Cell with NCellID=152 at Fc=-946.44 MHz

Scanning Frequency : -1409.54

[ ]:
```

Figure 8: **Jupyter Notebook Output from Over-the-Air Data (LNA + Antenna)**