Communication Systems Design

Lab 6: Cell Search and MIB Recovery

(Part 1)

Dr. Wu Guang

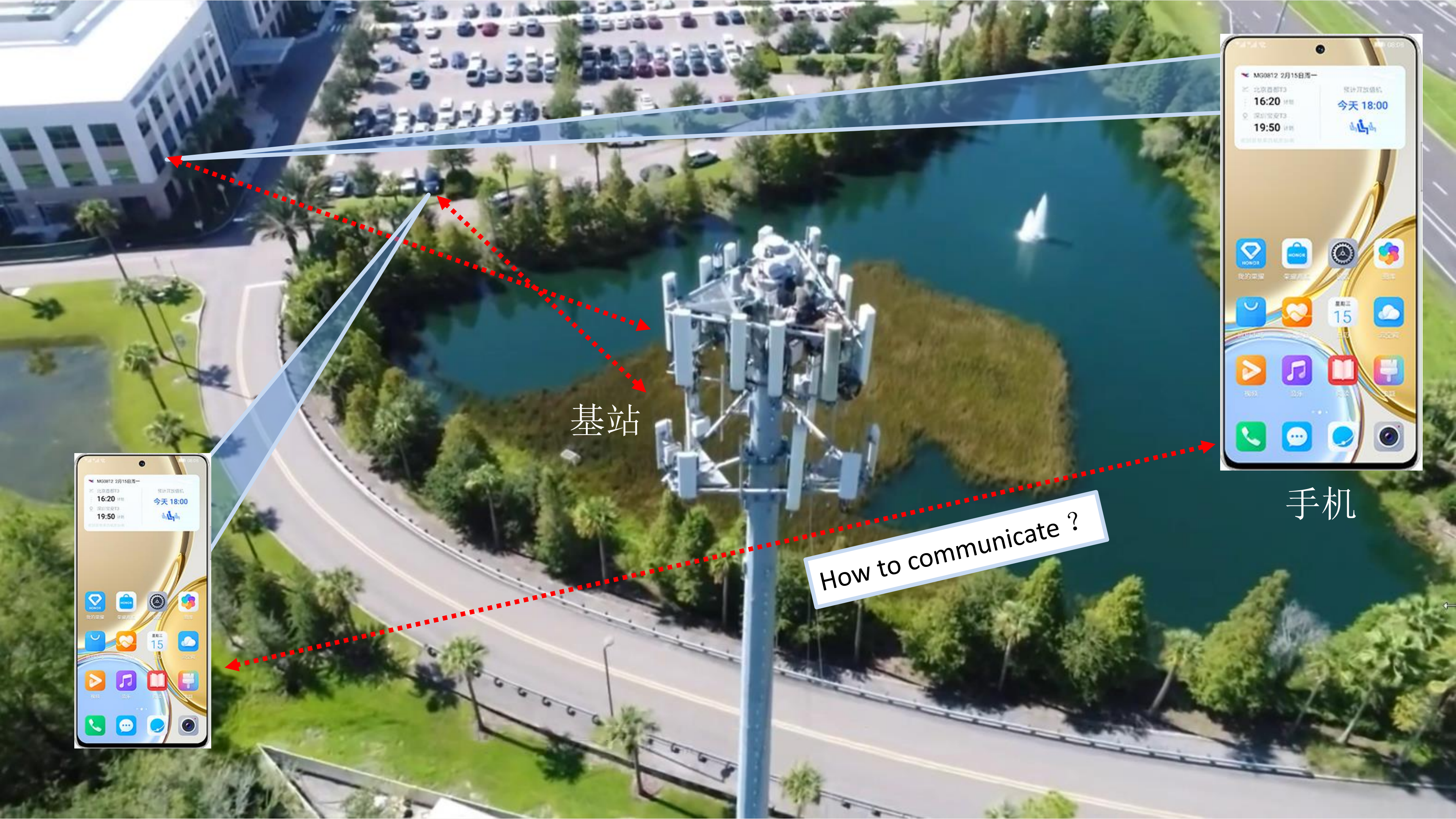
Email: wug@sustech.edu.cn

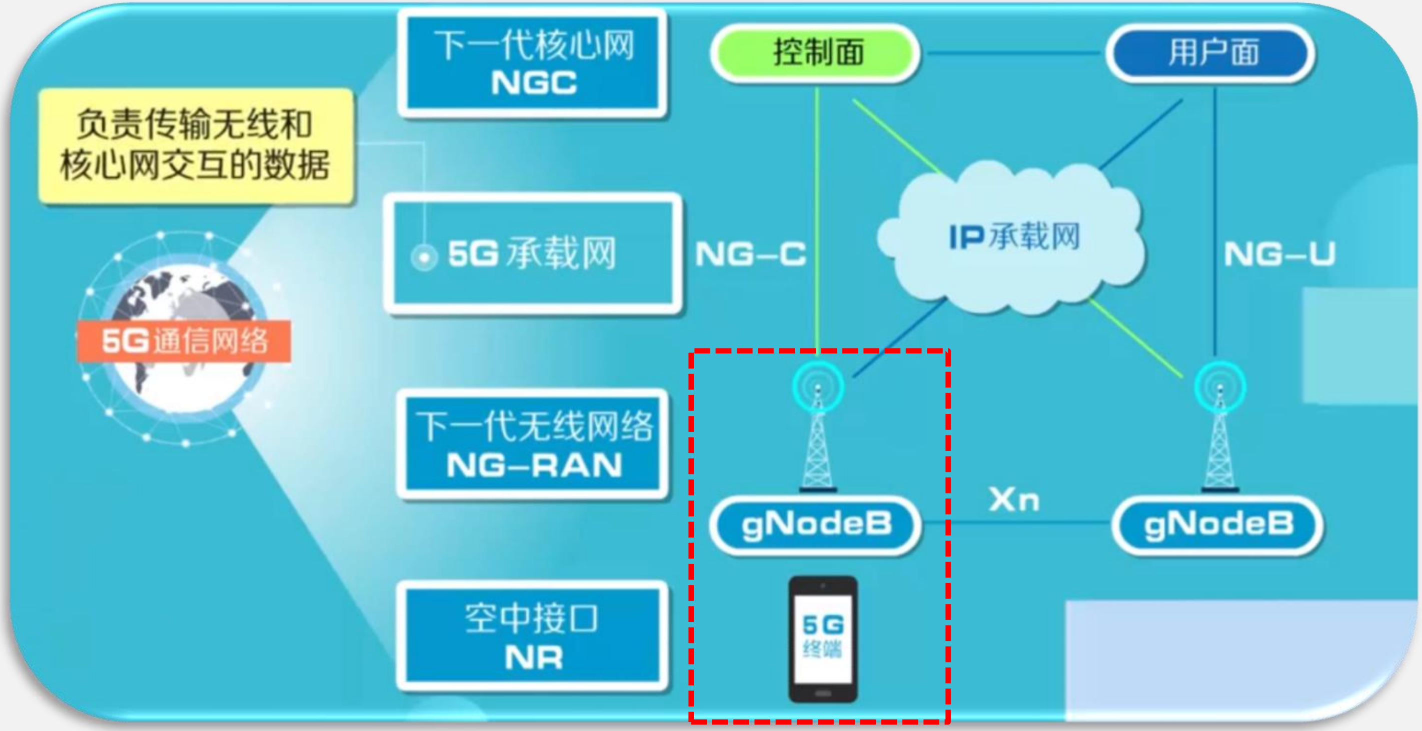
Electrical & Electronic Engineering

Southern University of Science and Technology

南方科技大学
SOUTHERN UNIVERSITY OF SCIENCE AND TECHNOLOGY

- 前沿通信系统设计（32学时）
  - 1 WiFi通信系统（9周）
    - 实验目标：利用USRP实现802.11a/n图像传输
    - 软件：MATLAB, 硬件：USRP
    - 授课内容：MATLAB通信编程、USRP文本传输、MIMO系统、802.11a/n仿真、802.11a/n图像传输
  - 2 5G/4G-LTE系统（4周）
    - 实验目标：利用USRP实现LTE图像传输
    - 软件：MATLAB，硬件：USRP
    - 授课内容：小区搜索过程、MIB/SIB解码过程、LTE图像传输、LDPC编解码过程、srsLTE系统
  - 3 无线网络传输系统（2周）
    - 实验目标：利用Telos实现无线多跳网络传输数据
    - 软件：TinyOS、NesC
    - 授课内容：TinOS编程、MICAZ平台介绍、无线多跳网络数据收集、无线信道建模、无线定位、路由和数据收集
  - 4 雷达感知系统（1周）
    - 实验目标：利用KerberosSDR实现测向
    - 软件：MATLAB，硬件：KerberosSDR、树莓派
    - 授课内容：MUSIC算法、空间谱估计、KerberosSDR原理，无线开源项目，课程Presentation

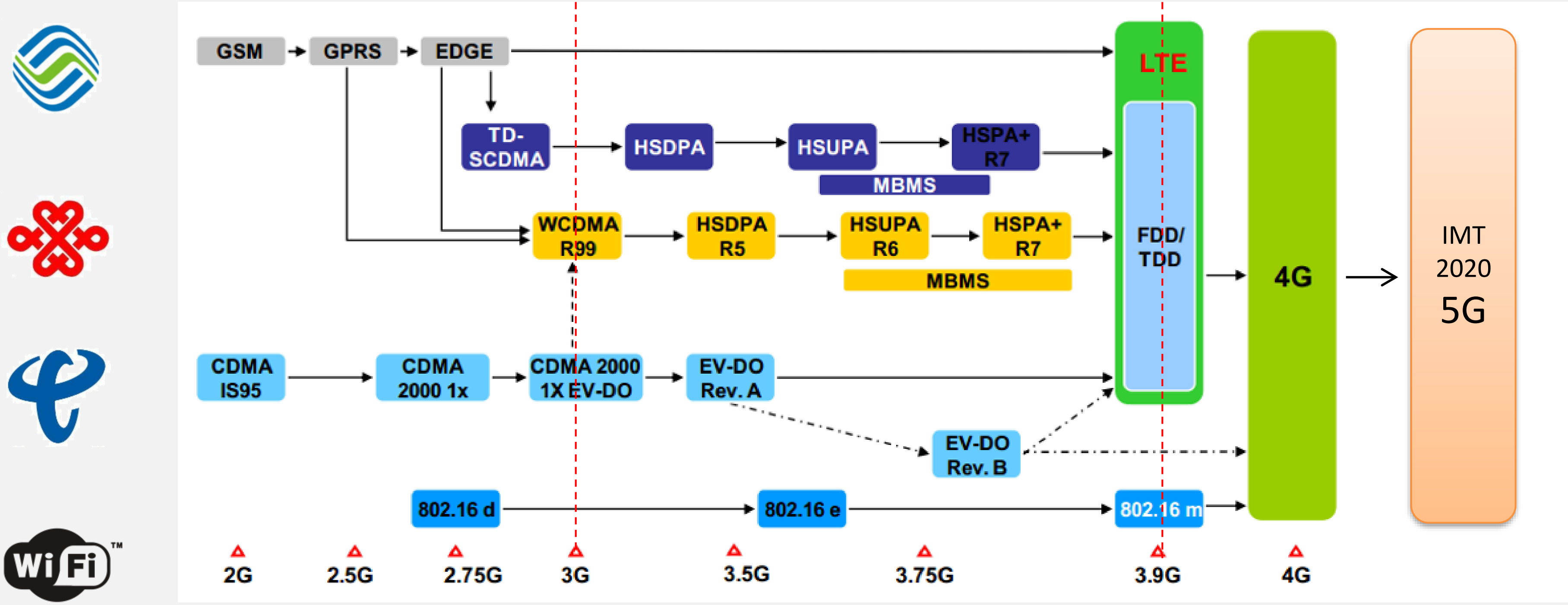

基站
手机
How to communicate？



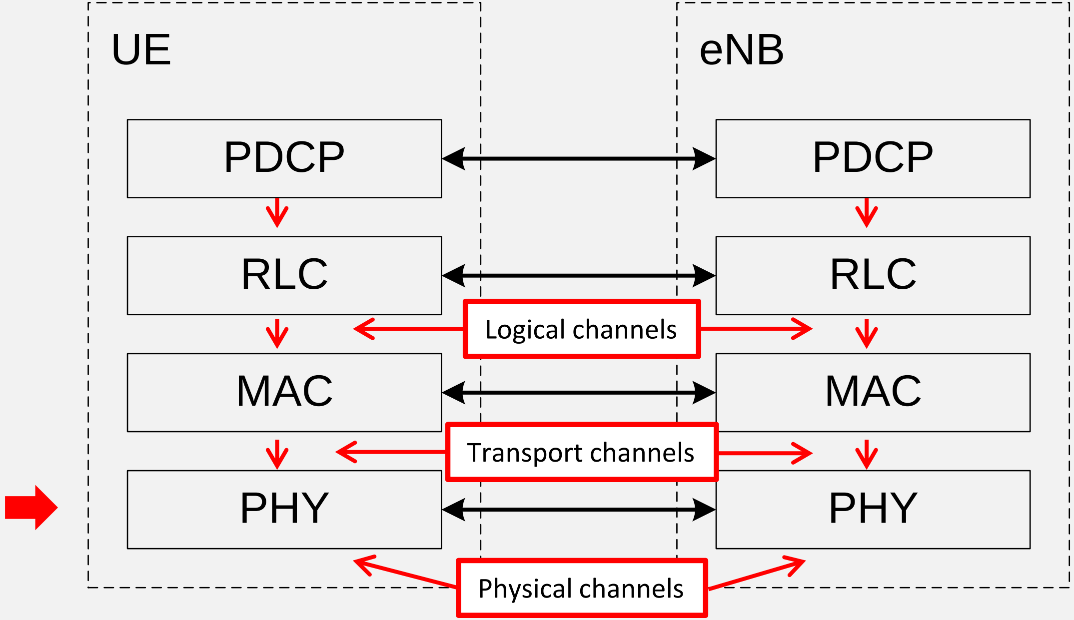
下一代核心网
NGC
控制面
用户面
负责传输无线和
核心网交互的数据
5G承载网
NG-C
IP承载网
NG-U
5G通信网络
下一代无线网络
NG-RAN
gNodeB
Xn
gNodeB
空中接口
NR
5G
终端

下午2:40
选取运营商
自动选择网络
自动选择首选网络
中国移动 4G
CMCC
中国移动 2G
CHINA MOBILE
中国联通 4G
CHN-UNICOM（禁用）
中国电信 4G
CHN-CT（禁用）

Evolution of Wireless Standards



Understanding LTE: Physical layer



1. **Frame structure**
2. **LTE Channel**

LTE System Toolbox

Simulate the physical layer of LTE and LTE-Advanced wireless communications systems

LTE System Toolbox™ provides standard-compliant functions and apps for the design, simulation, and verification of LTE and LTE-Advanced communications systems. The system toolbox accelerates LTE algorithm and physical layer (PHY) development, supports golden reference verification and conformance testing, and enables test waveform generation. With the system toolbox, you can configure, simulate, measure, and analyze end-to-end communications links. You can also create and reuse a conformance test bench to verify that designs, prototypes, and implementations comply with the LTE standard.

Examples
Functions
Release Notes
PDF Documentation

Getting Started
Learn the basics of LTE System Toolbox

LTE Modeling Basics
Resource grids, resource extraction, FDD and TDD duplexing modes, parameter structures

Downlink Channels
Physical signals and channels for transmit and receive, transport, control information, OFDM modulation

Uplink Channels
Physical signals and channels for transmit and receive, transport, control information, SC-FDMA modulation

Physical Layer Subcomponents
Downlink and uplink subcomponents for transport channels, physical channels, and physical signals

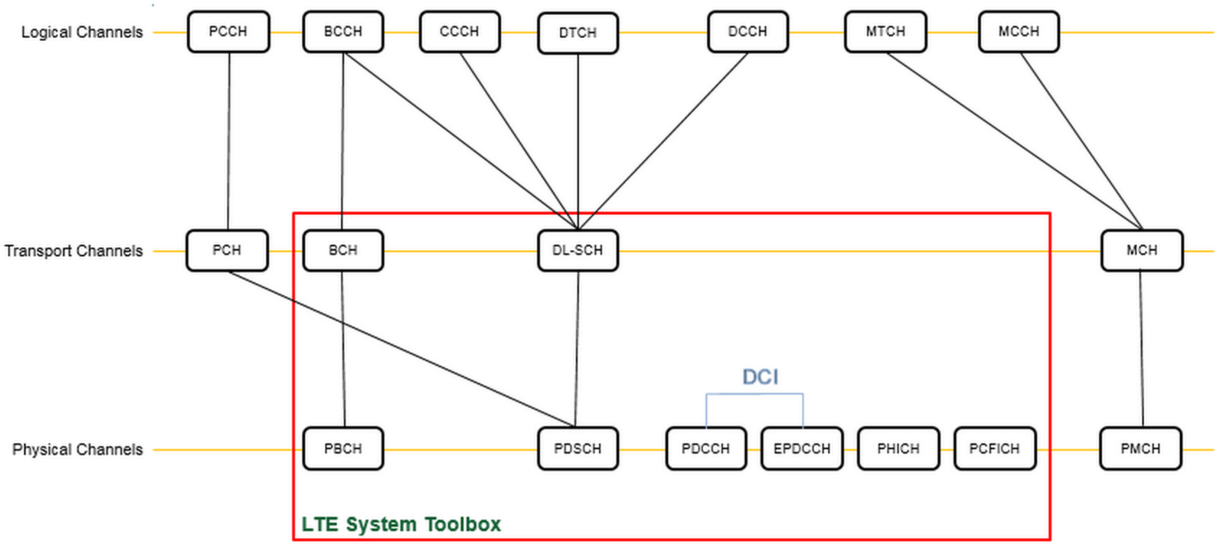
Signal Reception and Recovery
Frame synchronization, channel estimation, ZF and MMSE equalization, cell identity search, MIB decoding, SIB1 recovery, HARQ combining

End-to-End Simulation
Propagation channel models; RMC, FRC, and E-TM configuration and waveform generation; link-level BER and conformance test

Test and Measurement
Verification, waveform generation, EVM, and ACLR calculation

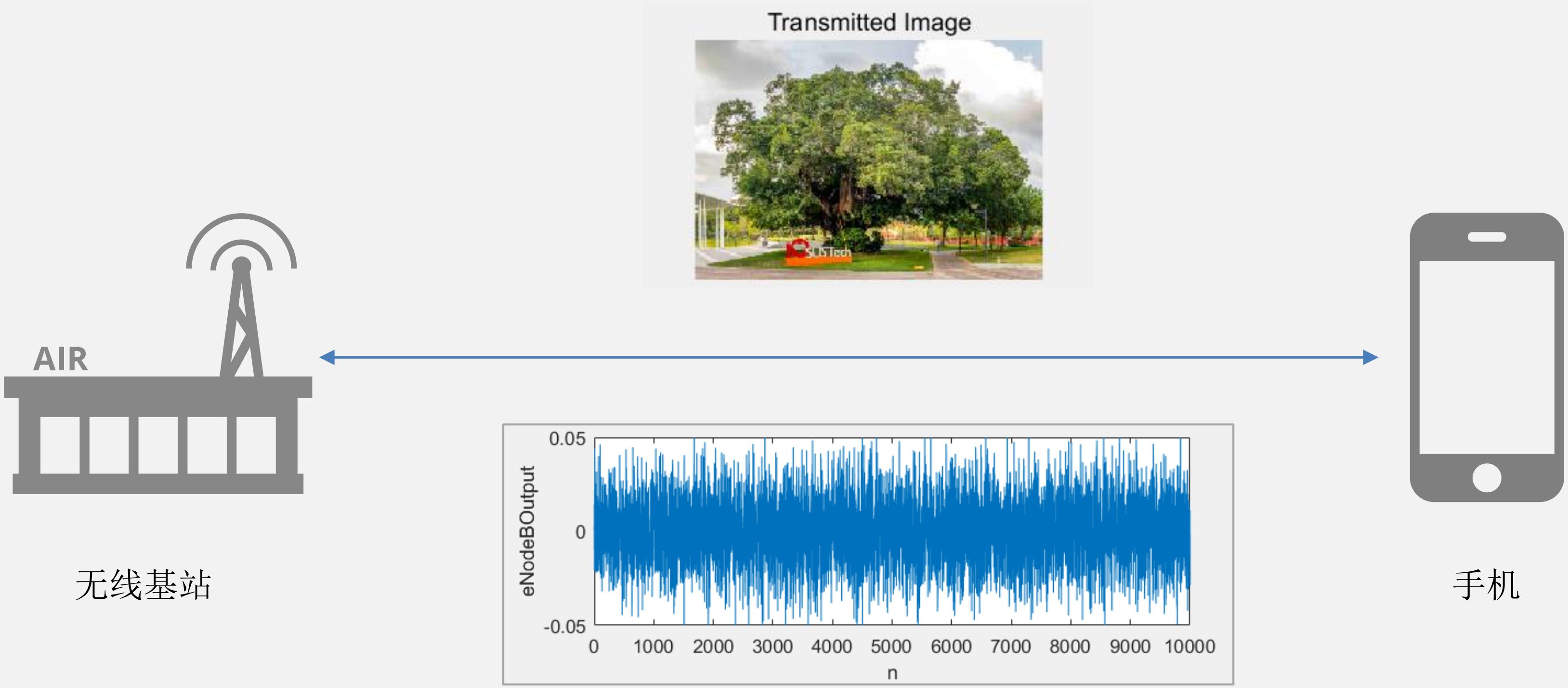
UMTS Test and Measurement

System downlink data follows the indicated mapping between logical channels, transport channels, and physical channels. The red outline contains LTE System Toolbox downlink functionality for physical channels, transport channels, and control information.



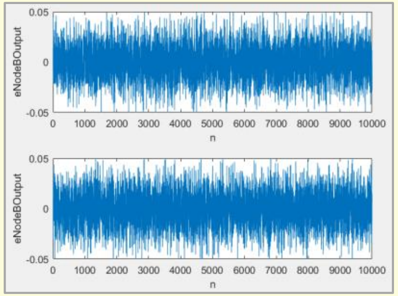
For more downlink channel information, see the channel or category of interest:

- PBCH
- PDSCH
- PDCCH
- PCFICH
- PHICH


Transmitted Image
AIR
无线基站
手机
eNodeBOutput
0.05
0
-0.05
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
n

```matlab

%% 2.1 导入图像，生成二进制码流
fileTx = 'tree.png';    % --------------------------------------------------------------> 定义图像文件名

scale = 0.4;        %-------------------------------------------------------------------> 缩放因子
[fData_Resize] = ResizeImage(fileTx,scale);  %------------------------------------------> 图像缩放

imsize = size(fData_Resize);     % -----------------------------------------------------> 新图像的尺寸

binData = dec2bin(fData_Resize(:),8); %-------------------------------------------------> 转换二进制数

trData = reshape((binData-'0').',1,[]).'; %---------------------------------------------> 创建二进制比特流

figure(1);  %---------------------------------------------------------------------------> 显示需要传输的图像
subplot(211);
    imshow(fData_Resize);
    title('Transmitted Image');
subplot(212);
    title('Received image will appear here...');
    set(gca,'Visible','off');
    set(findall(gca, 'type', 'text'), 'visible', 'on');
```

```
%% 2.2 设置DL-SCH下行链路参数
txsim.RC = 'R.7';          % RMC：Reference Measurement Channel-------------------------------------> 参考测量信道
txsim.NCellID = 88;        % --------------------------------------------------------------------> 小区标识
txsim.NFrame = 700;        % --------------------------------------------------------------------> 系统帧号
txsim.TotFrames = 1;       % --------------------------------------------------------------------> 初始化系统帧数
txsim.DesiredCenterFrequency = 2.45e9; % --------------------------------------------------------> 中心频率
txsim.NTxAnts = 1;         % --------------------------------------------------------------------> 发射天线数量

rmc = lteRMCDL(txsim.RC);  %---------------------------------------------------------------------> 创建RMC对象

trBlkSize = rmc.PDSCH.TrBlkSizes;
txsim.TotFrames = ceil(numel(trData)/sum(trBlkSize(:)));  %--------------------------------------> 计算所需要的LTE帧数（3个）

% 2.3 产生LTE复基带波形
[eNodeBOutput,txGrid,rmc] = LTEWaveformGenerator(rmc,txsim,trData);

figure(2)
subplot(2,1,1)
plot(real(eNodeBOutput)); axis([0 10000 -0.05 0.05]); xlabel('n') ;ylabel('eNodeBOutput')
subplot(2,1,2)
plot(imag(eNodeBOutput)); axis([0 10000 -0.05 0.05]); xlabel('n') ;ylabel('eNodeBOutput')
```



```
%% 2.2 设置DL-SCH下行链路参数
txsim.RC = 'R.7';        % RMC: Reference Measurer                  -----------------> 参考测量信道
txsim.NCellID = 88;      % ---------------------                    ----------------
txsim.NFrame = 700;      % ---------------------                    ----------------
txsim.TotFrames = 1;     % ---------------------                    ----------------
txsim.DesiredCenterFrequency = 2.45e9; % -------                    ----------------
txsim.NTxAnts = 1;       % ---------------------                    ----------------

rmc = lteRMCDL(txsim.RC);  %--------------------                    ----------------

trBlkSize = rmc.PDSCH.TrBlkSizes;
txsim.TotFrames = ceil(numel(trData)/sum(trBlkSi                    ----------------

% 2.3 产生LTE复基带波形
[eNodeBOutput,txGrid,rmc] = LTEWaveformGenerator

figure(2)
subplot(2,1,1)
plot(real(eNodeBOutput)); axis([0 10000 -0.05 0.                    tput')
subplot(2,1,2)
plot(imag(eNodeBOutput)); axis([0 10000 -0.05 0.                    tput')
```

```
K>> rmc

rmc =

                 RC: 'R.7'
              NDLRB: 50
           CellRefP: 1
            NCellID: 0
       CyclicPrefix: 'Normal'
                CFI: 2
        PCFICHPower: 0
                 Ng: 'Sixth'
       PHICHDuration: 'Normal'
              HISet: [112x3 double]
         PHICHPower: 0
             NFrame: 0
          NSubframe: 0
       TotSubframes: 10
          Windowing: 0
         DuplexMode: 'FDD'
              PDSCH: [1x1 struct]
   OCNGPDCCHEnable: 'Off'
    OCNGPDCCHPower: 0
   OCNGPDSCHEnable: 'Off'
    OCNGPDSCHPower: 0
          OCNGPDSCH: [1x1 struct]
```

```
K>> rmc.PDSCH

ans =

          TxScheme: 'Port0'
        Modulation: {'64QAM'}
           NLayers: 1
               Rho: 0
              RNTI: 1
             RVSeq: [0 0 1 2]
                RV: 0
      NHARQProcesses: 8
       NTurboDecIts: 5
            PRBSet: [50x1 double]
        TrBlkSizes: [1x10 double]
   CodedTrBlkSizes: [1x10 double]
         DCIFormat: 'Format1'
       PDCCHFormat: 2
        PDCCHPower: 0
           CSIMode: 'PUCCH 1-1'
           PMIMode: 'Wideband'
```

```
%% 3.1 定义接收机对象结构体参数
rxsim = struct;    % ------------------------------------------------------------> 定义接收机对象结构体参数
rxsim.RadioFrontEndSampleRate = rmc.SamplingRate; % ---------------------------------> 设置采样率
rxsim.RadioCenterFrequency = txsim.DesiredCenterFrequency;  % ----------------------> 设置接收机中心频率
rxsim.NRxAnts = txsim.NTxAnts;    % ---------------------------------------------> 设置天线数量
rxsim.FramesPerBurst = txsim.TotFrames+1;     %-----------------------------------> 设置系统帧数
rxsim.numBurstCaptures = 1;
samplesPerFrame = 10e-3*rxsim.RadioFrontEndSampleRate; % %--------------------------> 单个系统帧的采样点数

rx.BasebandSampleRate = rxsim.RadioFrontEndSampleRate;
rx.CenterFrequency = rxsim.RadioCenterFrequency;
rx.SamplesPerFrame = samplesPerFrame;
rx.OutputDataType = 'double';
rx.EnableBurstMode = true;
rx.NumFramesInBurst = rxsim.FramesPerBurst;
rx.ChannelMapping = 1;

burstCaptures = zeros(samplesPerFrame,rxsim.NRxAnts,rxsim.FramesPerBurst);
```

```
rx:

    BasebandSampleRate: 15360000
       CenterFrequency: 2.4500e+09
       SamplesPerFrame: 153600
        OutputDataType: 'double'
       EnableBurstMode: 1
      NumFramesInBurst: 4
        ChannelMapping: 1
```

```matlab
%% 3.2. 初始化ENodeB
enb.PDSCH = rmc.PDSCH;      % -------------------------------------------------------> 初始化eNodeB
enb.DuplexMode = 'FDD';     % -------------------------------------------------------> 设置复用方式
enb.CyclicPrefix = 'Normal';    % ---------------------------------------------------> 判断循环前缀方式
enb.CellRefP = 4;

% Bandwidth: {1.4 MHz, 3 MHz, 5 MHz, 10 MHz, 20 MHz}   % ---------------------------> 判断小区带宽
SampleRateLUT = [1.92 3.84 7.68 15.36 30.72]*1e6;
NDLRBLUT = [6 15 25 50 100];
enb.NDLRB = NDLRBLUT(SampleRateLUT==rxsim.RadioFrontEndSampleRate);
if isempty(enb.NDLRB)
    error('Sampling rate not supported. Supported rates are %s.',...
        '1.92 MHz, 3.84 MHz, 7.68 MHz, 15.36 MHz, 30.72 MHz');
end
fprintf('\nSDR hardware sampling rate configured to capture %d LTE RBs.\n',enb.NDLRB);

% 3.2. 信道估计结构体配置
cec.PilotAverage = 'UserDefined';  % -----------------------------------------------> Type of pilot symbol averagin
cec.FreqWindow = 9;                % -----------------------------------------------> Frequency window size in REs
cec.TimeWindow = 9;                % -----------------------------------------------> Time window size in REs
cec.InterpType = 'Cubic';          % -----------------------------------------------> 2D interpolation type
cec.InterpWindow = 'Centered';     % -----------------------------------------------> Interpolation window type
cec.InterpWinSize = 3;             % -----------------------------------------------> Interpolation window size
```

```
enb:
           PDSCH: [1x1 struct]
      DuplexMode: 'FDD'
    CyclicPrefix: 'Normal'
        CellRefP: 4
           NDLRB: 50
```

```
% *Signal Capture and Processing*
enbDefault = enb;

while rxsim.numBurstCaptures
    % Set default LTE parameters
      enb = enbDefault;
      rxWaveform  = eNodeBOutput;

    % Show power spectral density of captured burst
    hsa.SampleRate = rxsim.RadioFrontEndSampleRate;
    step(hsa,rxWaveform);

    % Perform frequency offset correction for known cell ID
    frequencyOffset = lteFrequencyOffset(enb,rxWaveform);
    rxWaveform = lteFrequencyCorrect(enb,rxWaveform,frequencyOffset);
    fprintf('\nCorrected a frequency offset of %i Hz.\n',frequencyOffset)

    % Perform the blind cell
    %   Use 'PostFFT' SSS detection method to improve speed
    cellSearch.SSSDetection = 'PostFFT'; cellSearch.MaxCellCount = 1;
    [NCellID,frameOffset] = lteCellSearch(enb,rxWaveform,cellSearch);
    fprintf('Detected a cell identity of %i.\n', NCellID);
    enb.NCellID = NCellID; % From lteCellSearch

    % Sync the captured samples to the start of an LTE frame, and trim off
    % any samples that are part of an incomplete frame.
    rxWaveform = rxWaveform(frameOffset+1:end,:);
    tailSamples = mod(length(rxWaveform),samplesPerFrame);
    rxWaveform = rxWaveform(1:end-tailSamples,:);
    enb.NSubframe = 0;
    fprintf('Corrected a timing offset of %i samples.\n',frameOffset)
```

How to perform cell search with PSS/SSS here ?

```
K>> enb

enb =

        PDSCH: [1x1 struct]
   DuplexMode: 'FDD'
 CyclicPrefix: 'Normal'
     CellRefP: 4
        NDLRB: 50
```

```
enb =

        PDSCH: [1x1 struct]
   DuplexMode: 'FDD'
 CyclicPrefix: 'Normal'
     CellRefP: 4
        NDLRB: 50
      NCellID: 88
```

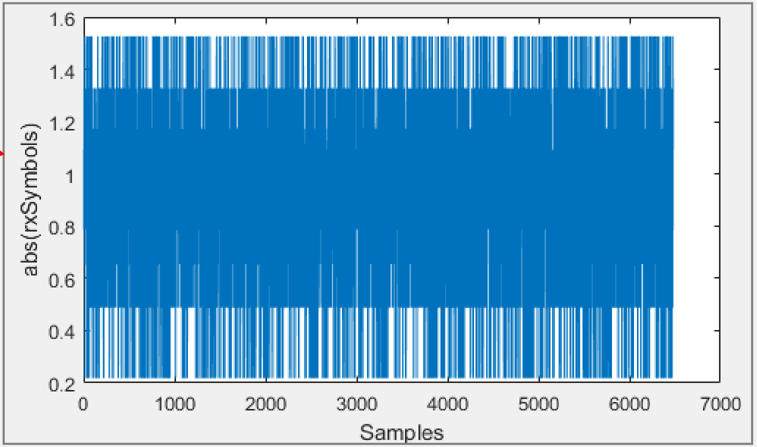
```matlab
% For each frame decode the MIB, PDSCH and DL-SCH
for frame = 0:(numFullFrames-1)
    fprintf('\nPerforming DL-SCH Decode for frame %i of %i in burst:\n', ...
        frame+1,numFullFrames)

    % Extract subframe #0 from each frame of the received resource grid
    enb.NSubframe = 0;
    rxsf = rxGrid(:,frame*LFrame+(1:Lsf),:);
    hestsf = hest(:,frame*LFrame+(1:Lsf),:,:);

    % PBCH demodulation.
    enb.CellRefP = 4;
    pbchIndices = ltePBCHIndices(enb);
    [pbchRx,pbchHest] = lteExtractResources(pbchIndices,rxsf,hestsf);
    [~,~,nfmod4,mib,CellRefP] = ltePBCHDecode(enb,pbchRx,pbchHest,nest);

    % If PBCH decoding successful CellRefP~=0 then update info
    if ~CellRefP
        fprintf('  No PBCH detected for frame.\n');
        continue;
    end
    enb.CellRefP = CellRefP; % From ltePBCHDecode

    % Decode the MIB to get current frame number
    enb = lteMIB(mib,enb);

    % Incorporate the nfmod4 value output from the function
    % ltePBCHDecode, as the NFrame value established from the MIB
    % is the system frame number modulo 4.
    enb.NFrame = enb.NFrame+nfmod4;
    fprintf('  Successful MIB Decode.\n')
    fprintf('  Frame number: %d.\n',enb.NFrame);

    % The eNodeB transmission bandwidth may be greater than the
    % captured bandwidth, so limit the bandwidth for processing
    enb.NDLRB = min(enbDefault.NDLRB,enb.NDLRB);
```

```
enb =

          PDSCH: [1x1 struct]
     DuplexMode: 'FDD'
   CyclicPrefix: 'Normal'
       CellRefP: 4
          NDLRB: 50
        NCellID: 88
      NSubframe: 0
```

```
enb =

          PDSCH: [1x1 struct]
     DuplexMode: 'FDD'
   CyclicPrefix: 'Normal'
       CellRefP: 1
          NDLRB: 50
        NCellID: 88
      NSubframe: 0
   PHICHDuration: 'Normal'
             Ng: 'Sixth'
         NFrame: 700
```

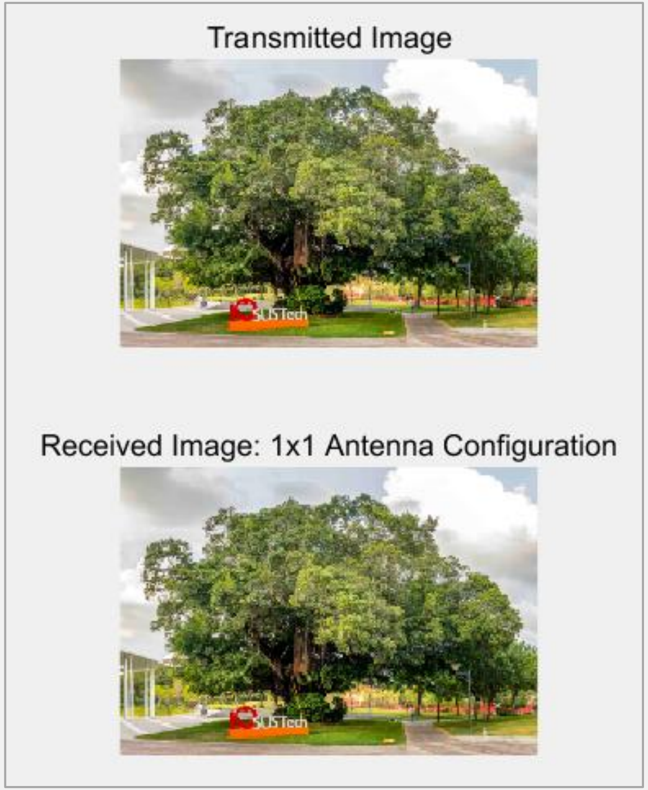
```matlab
% PCFICH demodulation. Extract REs corresponding to the PCFICH
% from the received grid and channel estimate for demodulation.
pcfichIndices = ltePCFICHIndices(enb);
[pcfichRx,pcfichHest] = lteExtractResources(pcfichIndices,rxsf,hestsf);
[cfiBits,recsym] = ltePCFICHDecode(enb,pcfichRx,pcfichHest,nestsf);

% CFI decoding
enb.CFI = lteCFIDecode(cfiBits);

% Get PDSCH indices
[pdschIndices,pdschIndicesInfo] = ltePDSCHIndices(enb, enb.PDSCH, enb.PDSCH.PRBSet);
[pdschRx, pdschHest] = lteExtractResources(pdschIndices, rxsf, hestsf);

% Perform deprecoding, layer demapping, demodulation and
% descrambling on the received data using the estimate of
% the channel
[rxEncodedBits, rxEncodedSymb] = ltePDSCHDecode(enb,enb.PDSCH,pdschRx,...
                               pdschHest,nestsf);

% Append decoded symbol to stream
rxSymbols = [rxSymbols; rxEncodedSymb{:}]; %#ok<AGROW>

% Transport block sizes
outLen = enb.PDSCH.TrBlkSizes(enb.NSubframe+1);

% Decode DownLink Shared Channel (DL-SCH)
[decbits{sf+1}, blkcrc(sf+1)] = lteDLSCHDecode(enb,enb.PDSCH,...
                                outLen, rxEncodedBits);  %#ok<SAGROW>

% Recode transmitted PDSCH symbols for EVM calculation
%   Encode transmitted DLSCH
txRecode = lteDLSCH(enb,enb.PDSCH,pdschIndicesInfo.G,decbits{sf+1});
%   Modulate transmitted PDSCH
txRemod = ltePDSCH(enb, enb.PDSCH, txRecode);
%   Decode transmitted PDSCH
[~,refSymbols] = ltePDSCHDecode(enb, enb.PDSCH, txRemod);
```

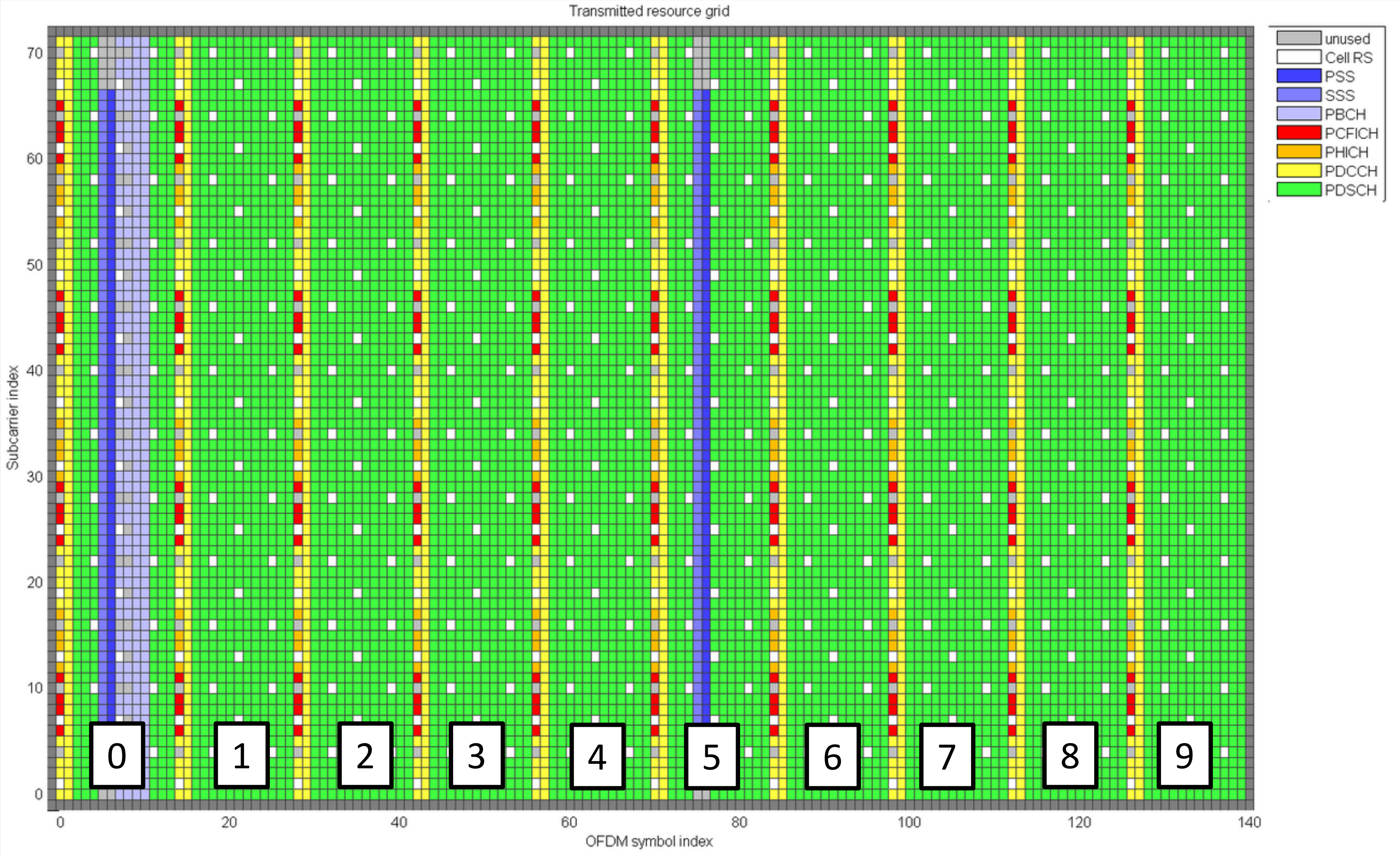
```
enb =

           PDSCH: [1x1 struct]
      DuplexMode: 'FDD'
     CyclicPrefix: 'Normal'
        CellRefP: 1
           NDLRB: 50
         NCellID: 88
       NSubframe: 0
    PHICHDuration: 'Normal'
              Ng: 'Sixth'
          NFrame: 700
             CFI: 2
```



Simulation results

```
Performing DL-SCH Decode for frame 1 of 3 in burst:
  Successful MIB Decode.
  Frame number: 700.
  Retrieving decoded transport block data.

Performing DL-SCH Decode for frame 2 of 3 in burst:
  Successful MIB Decode.
  Frame number: 701.
  Retrieving decoded transport block data.

Performing DL-SCH Decode for frame 3 of 3 in burst:
  Successful MIB Decode.
  Frame number: 702.
  Retrieving decoded transport block data.

Recombining received data blocks:
  EVM peak = 0.000%
  EVM RMS  = 0.000%
  Bit Error Rate (BER) = 0.00000.
  Number of bit errors = 0.
  Number of transmitted bits = 815184.

Constructing image from received data.
```





Transmitted resource grid
unused
Cell RS
PSS
SSS
PBCH
PCFICH
PHICH
PDCCH
PDSCH
Subcarrier index
OFDM symbol index
0
10
20
30
40
50
60
70
0
20
40
60
80
100
120
140
0
1
2
3
4
5
6
7
8
9

Assignments

- Read the example ‘**Transmit and Receive LTE MIMO Using a Single Analog Devices AD9361/AD9364**’ in LTE System Toolbox.

- Explain the functions of the following eight subcomponents respectively,

  (1) lteRMCDL.m

  (2) lteRMCDLTool.m

  (3) lteFrequencyOffset.m

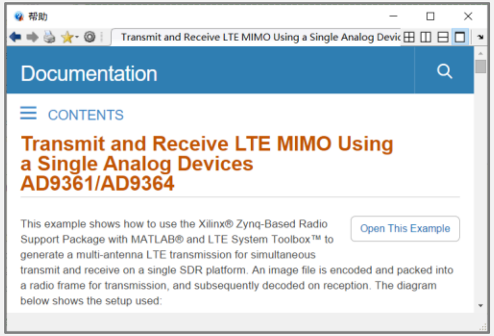
  (4) lteFrequencyCorrect.m

  (5) lteCellSearch.m

  (6) lteOFDMDemodulate.m

  (7) lteDLChannelEstimate.m

  (8) lteResourceGridSize.m



Questions