USC Viterbi
School of Engineering
Center for Cyber-Physical Systems and the Internet of Things

CCI

SOFTWARE DEFINED RADIO

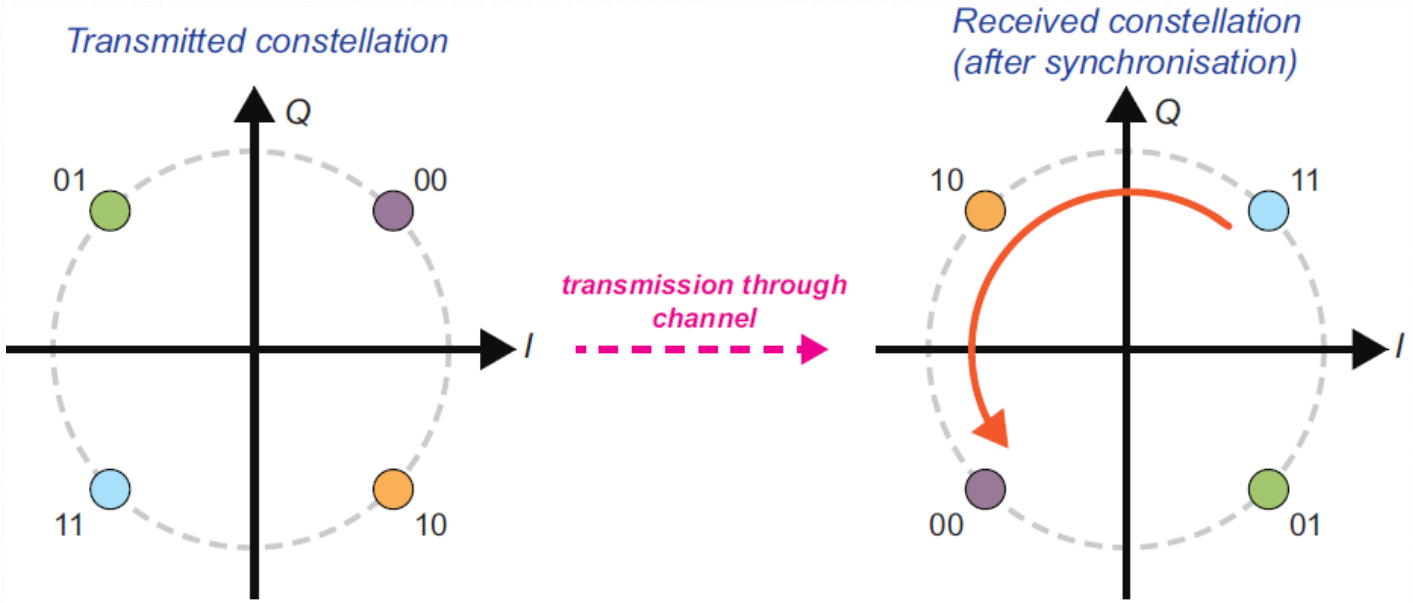
USR SDR WORKSHOP, SEPTEMBER 2017

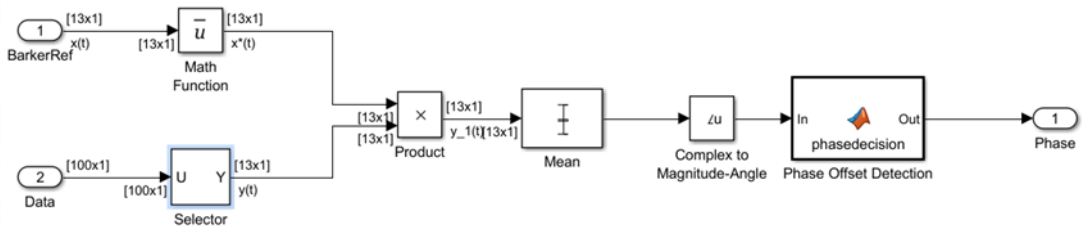
PROF. MARCELO SEGURA

SESSION 5: QPSK TX/RX IMPLEMENTATION

PHASE AMBIGUITY

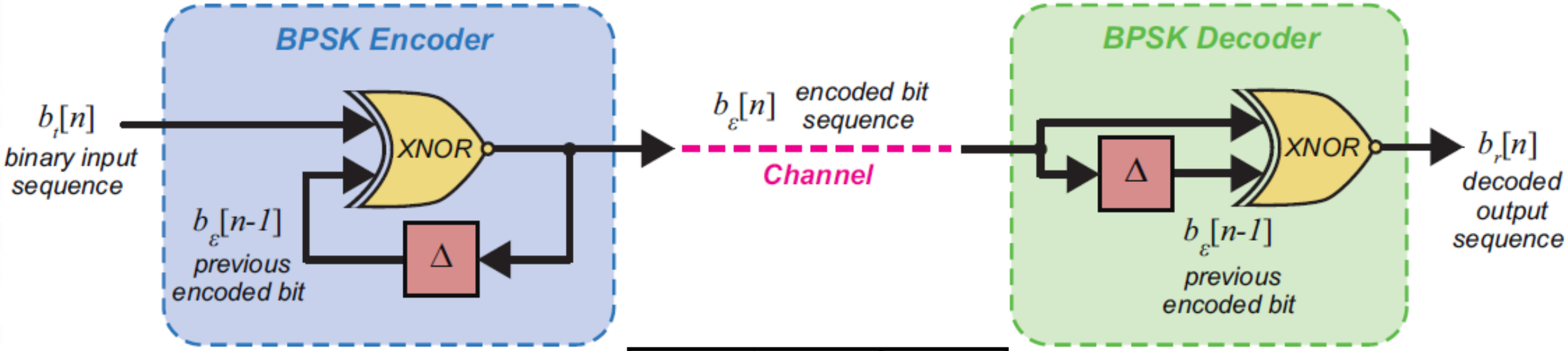
- In the case M-PSK modulation, due to rotational symmetry, the synchronization could lock to any of the M phases.
- To solve this situation, differential encoding can be used or Data Aided rotation.





DIFFERENTIAL ENCODING

- Differential encoding maps the data to the phase shift instead to the absolute phase.
- For **BPSK**, M=2, is very simple implementation using an XNOR



$b_t[n]$	$b_\varepsilon[n-1]$	$b_\varepsilon[n]$
0	0	1
0	1	0
1	0	0
1	1	1

DIFFERENTIAL ENCODING: BPSK

- If we have a phase error of π, the encoded and decoded data is as follow:
- ENCODER

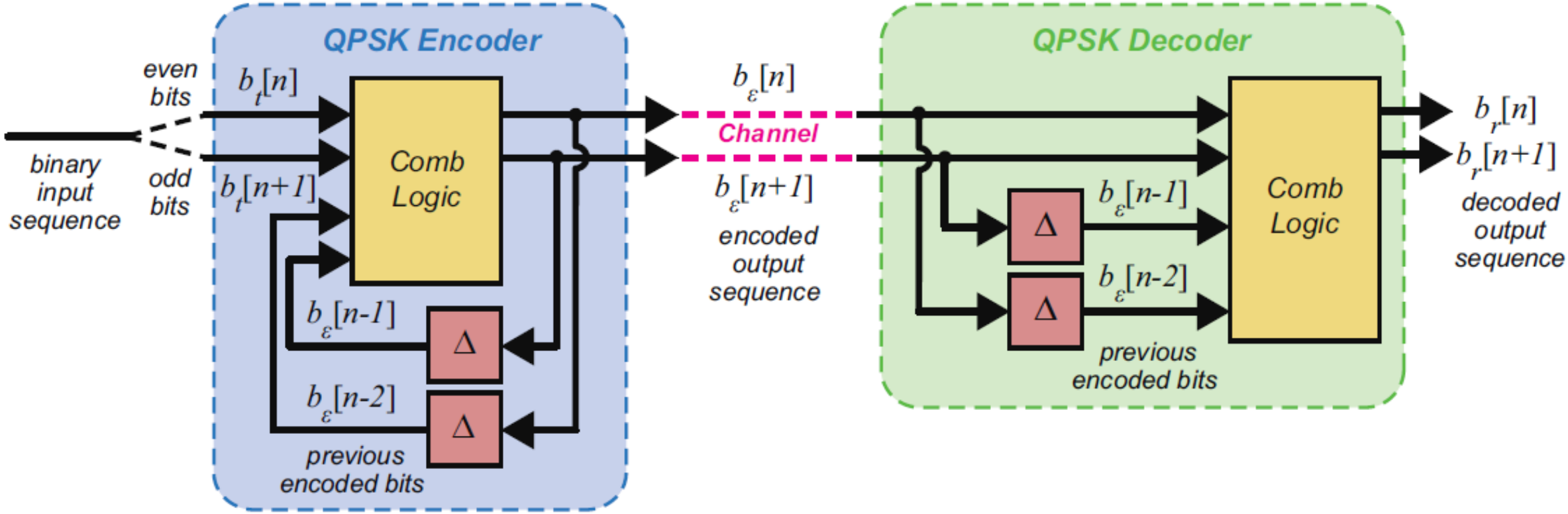
n	0	1	2	3	4	5	6	7
$b_t[n]$	0	0	1	0	1	1	0	1
$b_\varepsilon[n-1]$	0	1	0	0	1	1	1	0
$b_\varepsilon[n]$	1	0	0	1	1	1	0	0

- DECODER

n	0	1	2	3	4	5	6	7
$b_\varepsilon[n]$	0	1	1	0	0	0	1	1
$b_\varepsilon[n-1]$	0	0	1	1	0	0	0	1
$b_r[n]$	1	0	1	0	1	1	0	1

DIFFERENTIAL ENCODING: QPSK

- M=4



DIFFERENTIAL ENCODING: QPSK

- Example $b_t = \{0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0, 1\}$
- ENCODER

n	0	1	2	3	4	5	6	7
$b_t[n]$	0	0	1	0	1	1	0	0
$b_t[n+1]$	1	0	1	1	1	0	0	1
$b_\varepsilon[n-2]$	0	0	0	1	0	1	0	0
$b_\varepsilon[n-1]$	0	1	1	0	0	1	1	1
$b_\varepsilon[n]$	0	0	1	0	1	0	0	1
$b_\varepsilon[n+1]$	1	1	0	0	1	1	1	1

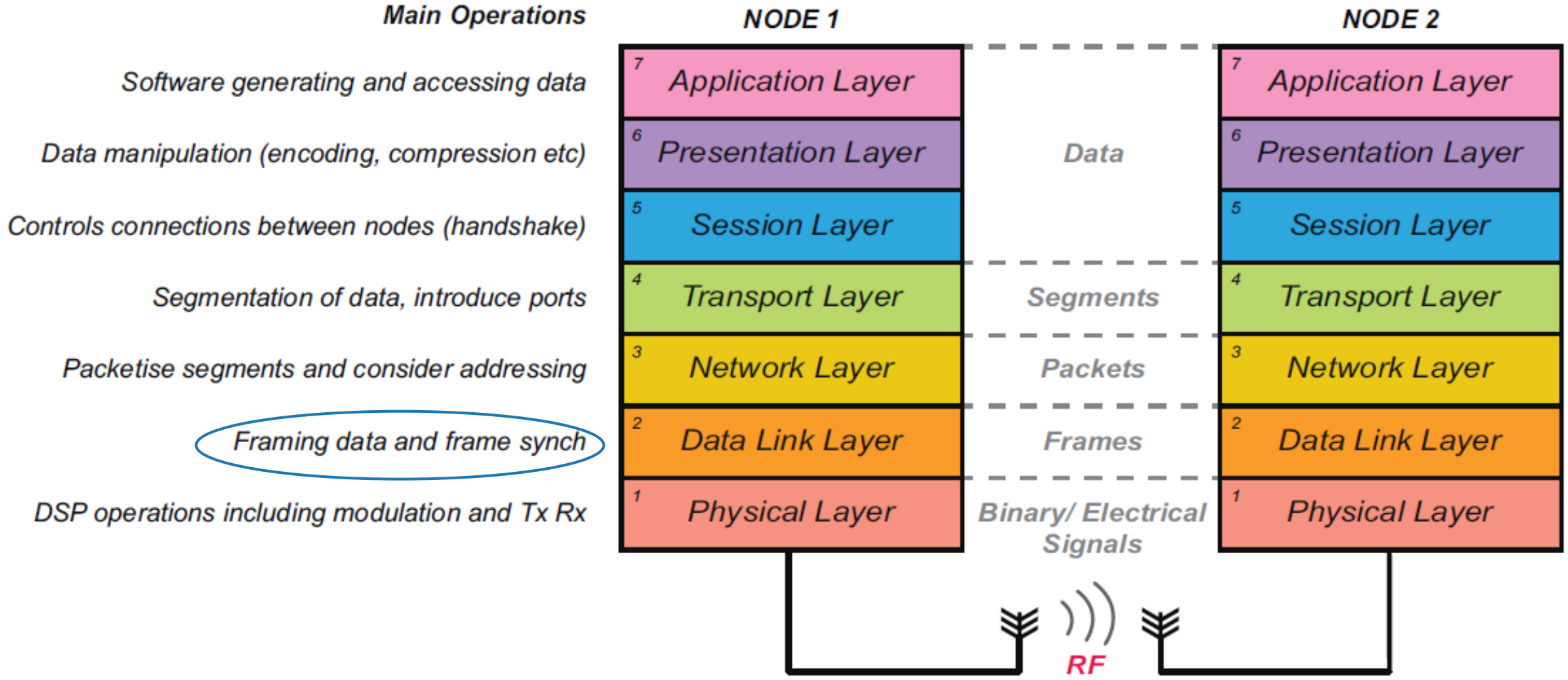
Phase error π

- DECODER

n	0	1	2	3	4	5	6	7
$b_\varepsilon[n-2]$	0	1	1	0	1	0	1	1
$b_\varepsilon[n-1]$	0	0	0	1	1	0	0	0
$b_\varepsilon[n]$	1	1	0	1	0	1	1	0
$b_\varepsilon[n+1]$	0	0	1	1	0	0	0	0
$b_r[n]$	1	0	1	0	1	1	0	0
$b_r[n+1]$	0	0	1	1	1	0	0	1

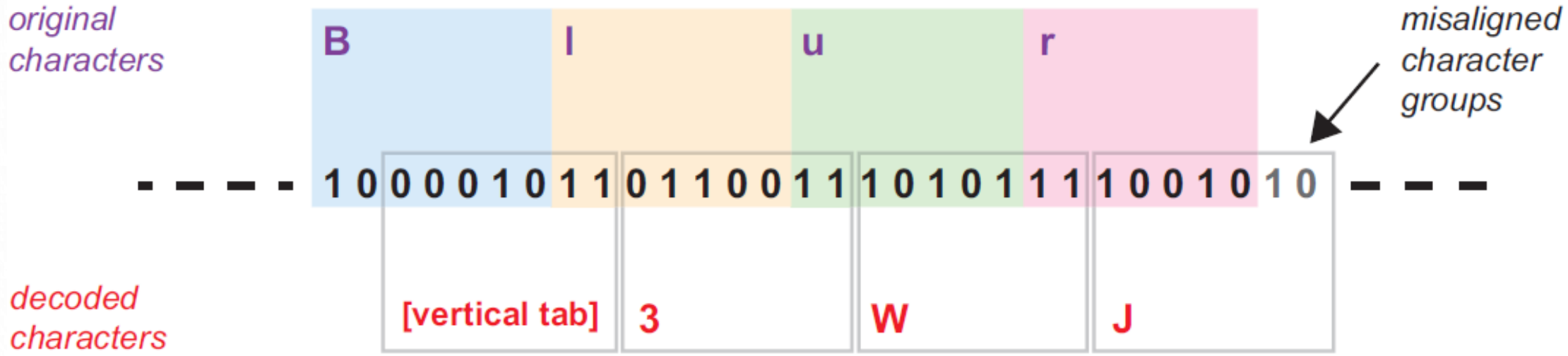
COMM PROTOCOL

- From now we just work on the **PHY** layer, then we need to move to the Data Link Layer.



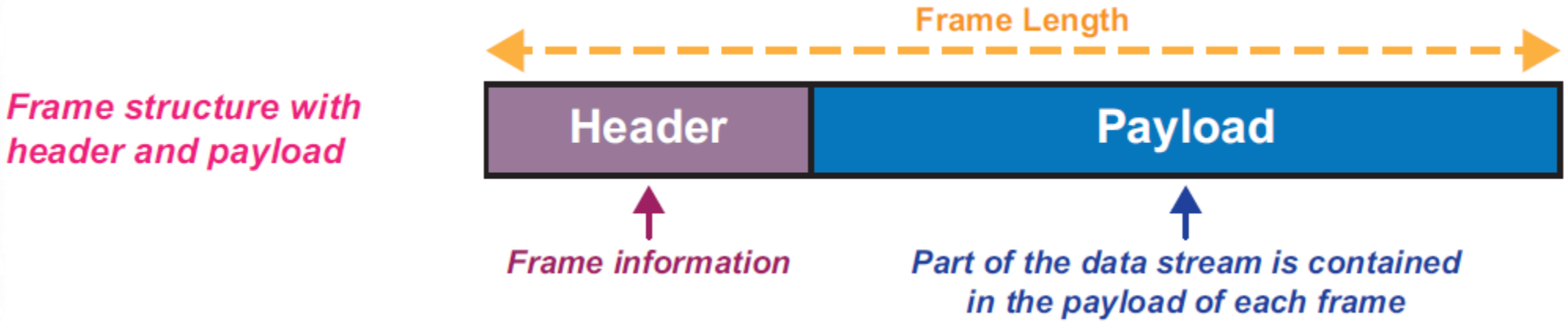
FRAME SYNC

- The basic task is to detect the start of a signal information. If we are transmitting text, encoded as ASCII, we need to detect the sequence start to decode properly.

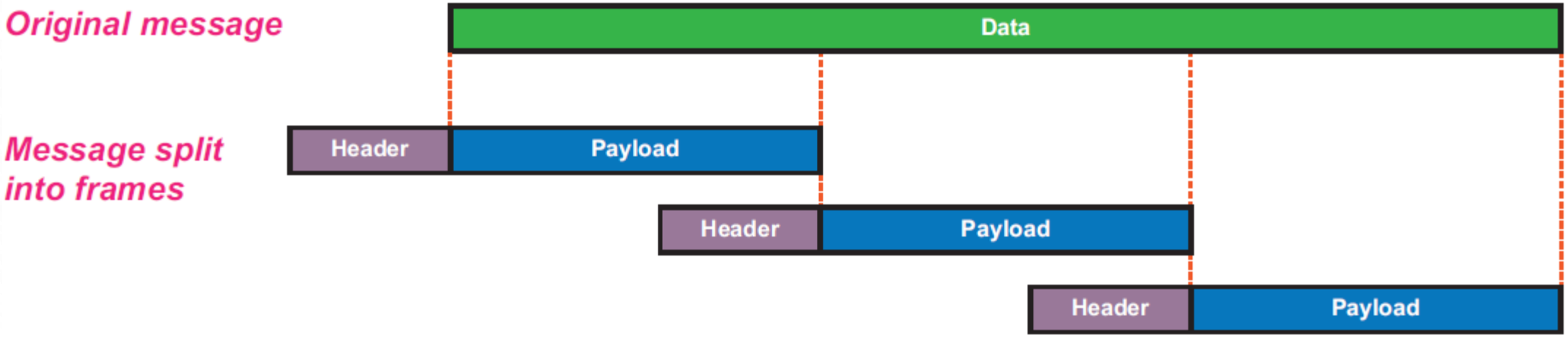


FRAME SYNC

- The Header transport the information useful for synchronization.

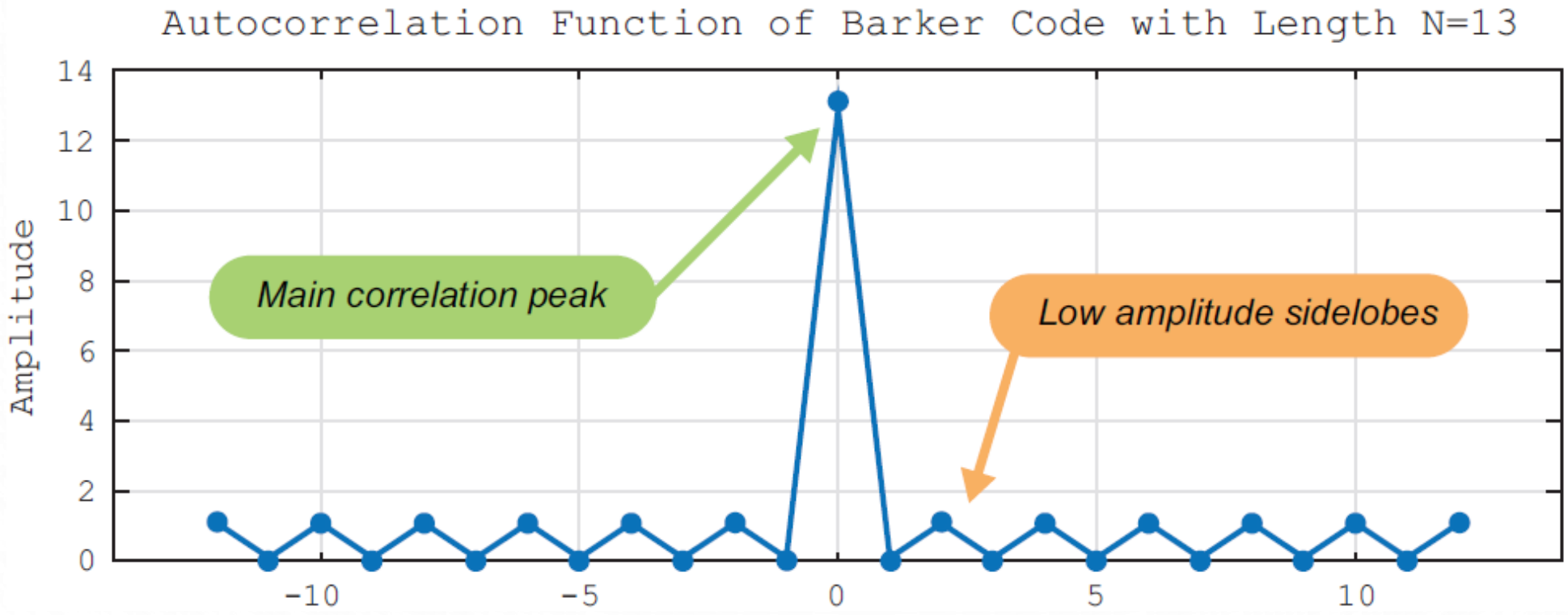


- Advantages: in cases of error just re-transmite the lost part.



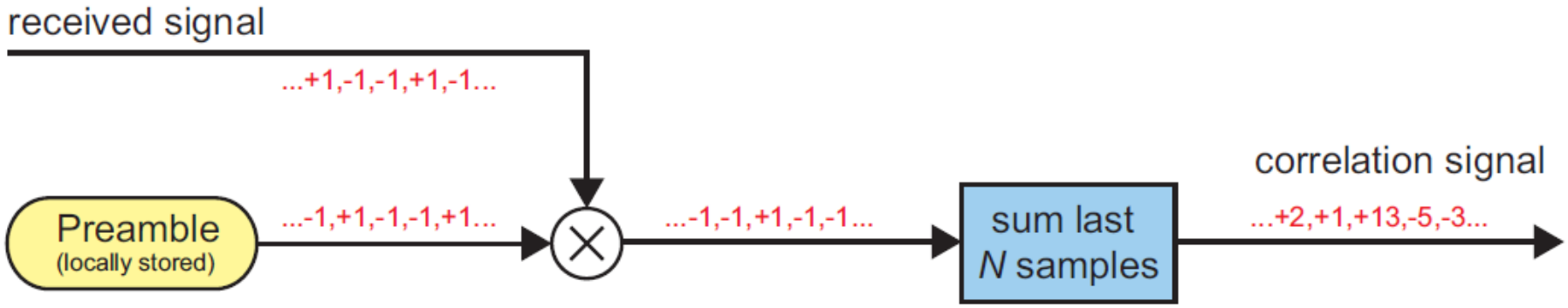
FRAME SYNC: PREAMBLE

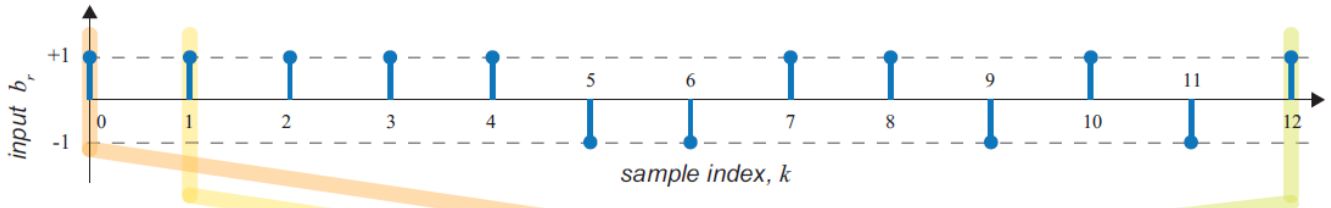
- Usually the Header contain a Preamble, that is a sequence with good autocorrelation properties.
- The preamble sequence is very important for start detection.
- **BARKER sequence** is one with have this characteristics.



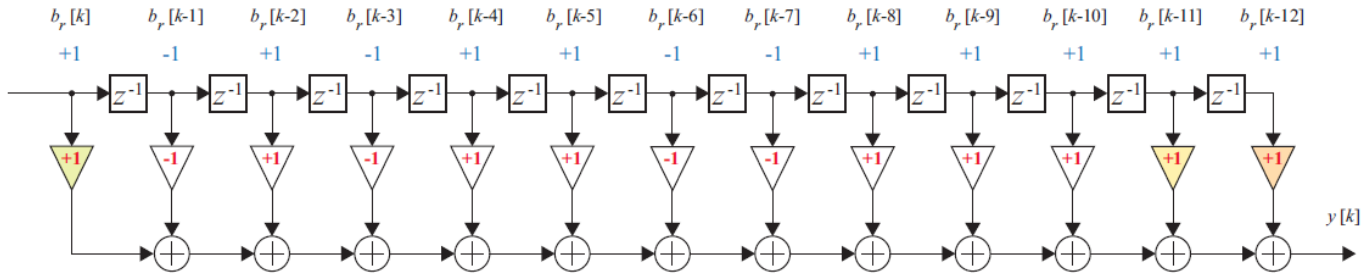
FRAME SYNC: CORRELATOR

- The receiver implement a correlator, FIR with inverted coefficients, to detect the frame start, usually called Start Frame Delimiter (SFD).



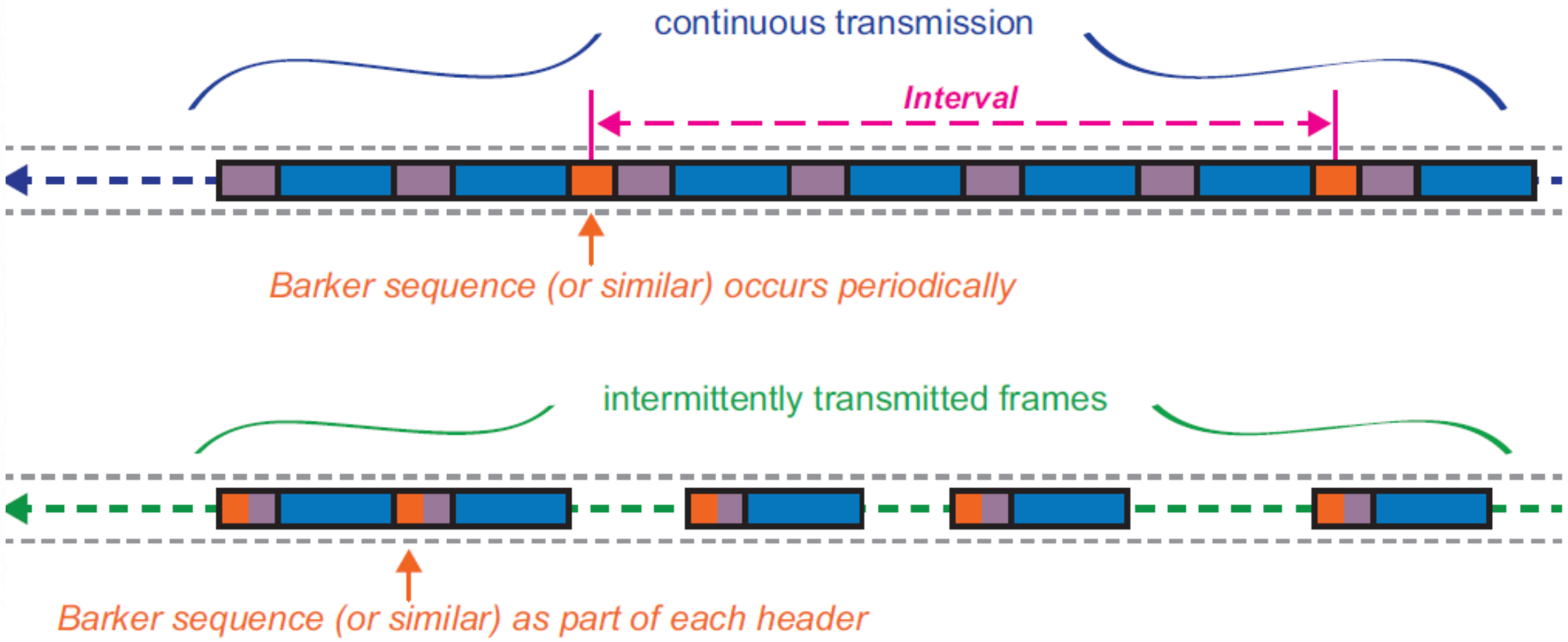


For $n = 12...$



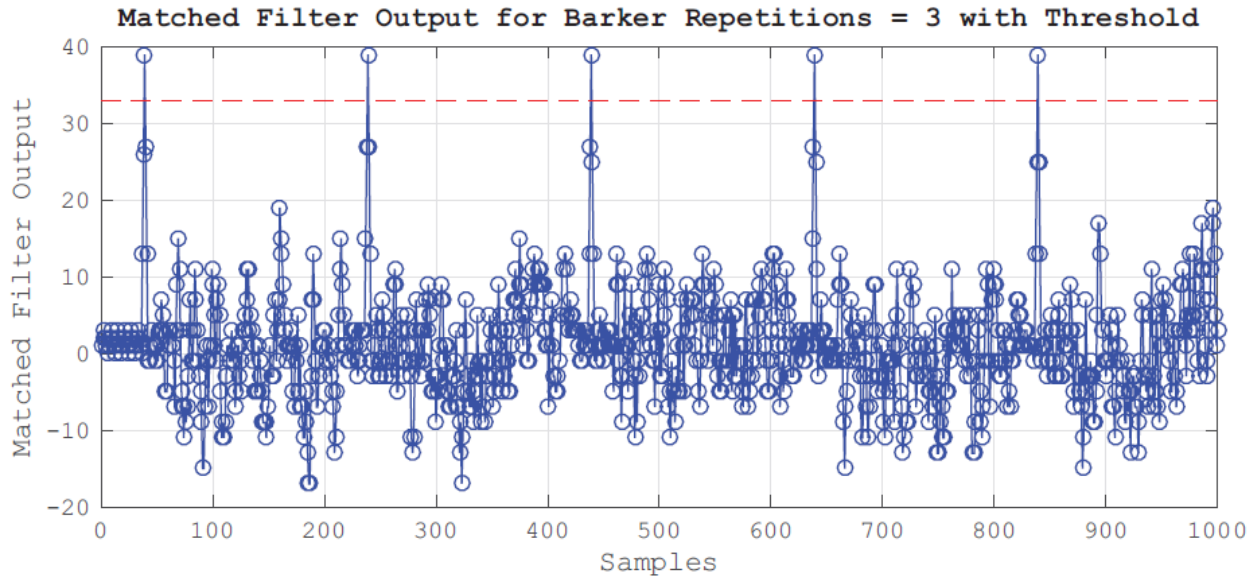
FRAME SYNC

- Depends on the type of communication systems, will be the load that header will introduce.



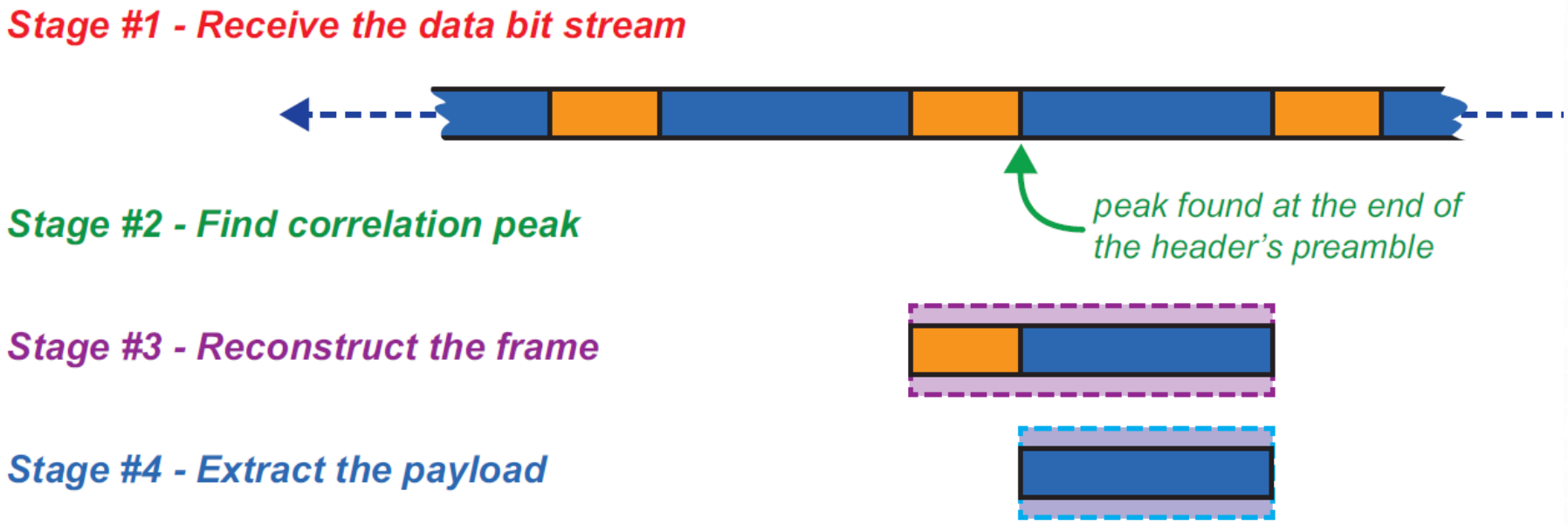
FRAME SYNC

- In order to **get better correlation peaks**, the sequence can be repeated, this also reduce efficiency.
- Ex Barker N=26, Frame=200, Payload=174, efficiencies 174/200=87%



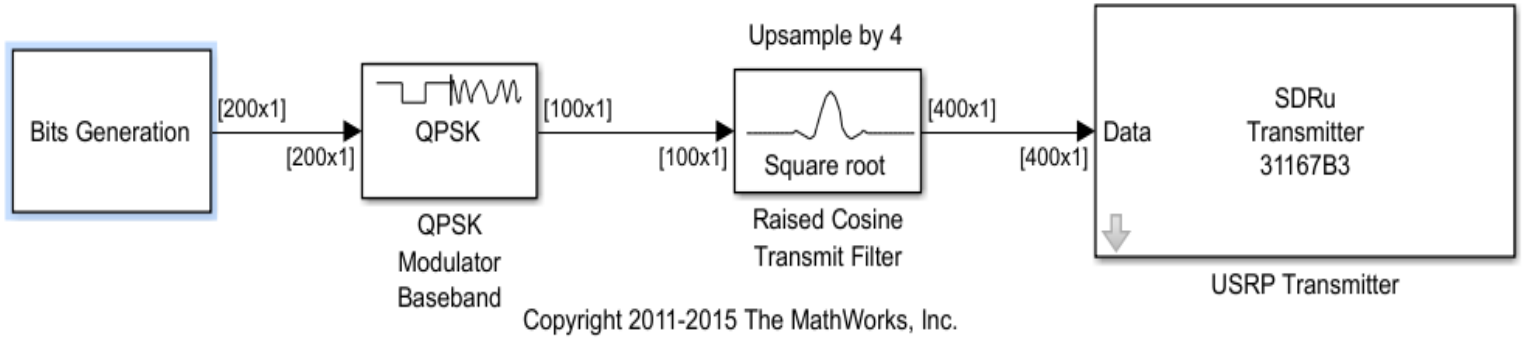
FRAME SYNC

- A threshold should be selected to capture the payload when the correlator output cross that level.



QPSK TX

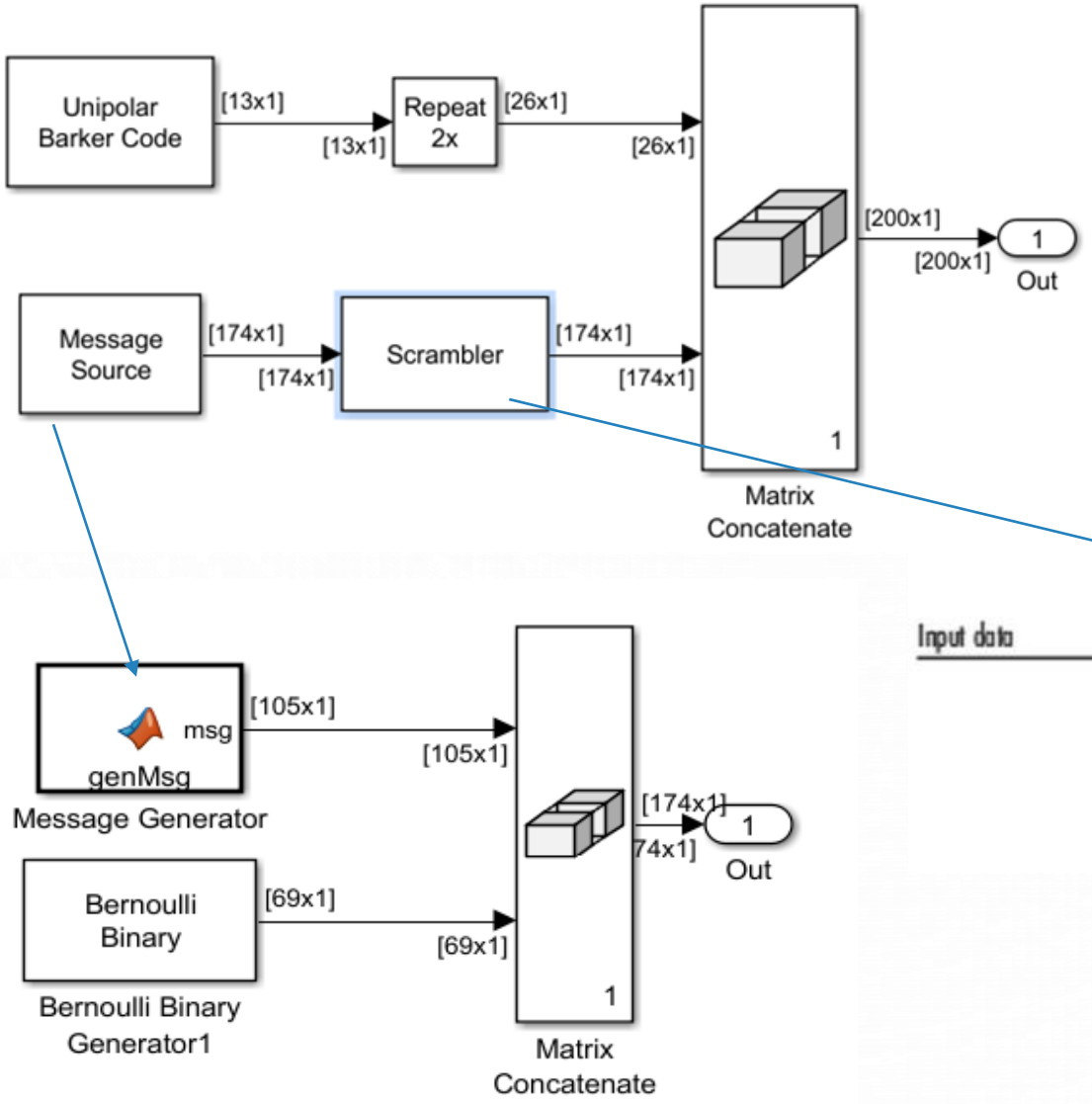
- Tx Model

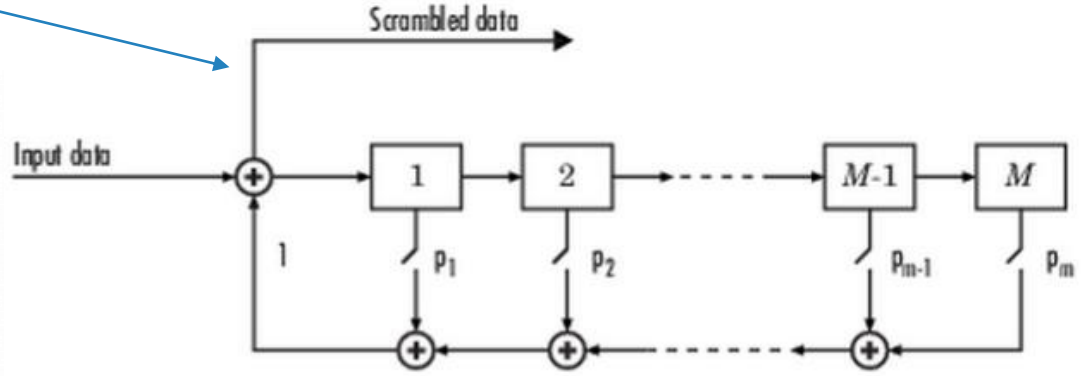


- **Bit Generation block:**
  - Data Frame = 200 bits
  - Header= 2 Baker 13 sequences (26 bits)
  - DataLenght= 200-26= 174bits
  - MessageLength=7bits (Ascci) x 15 (Hello World xxx)= 105 bits
  - Random bits to improve Scramble= 69 (Bernoulli Gen)
  - Scramble: use to reduce the number of consecutives '1' or '0', helps the synchronization.

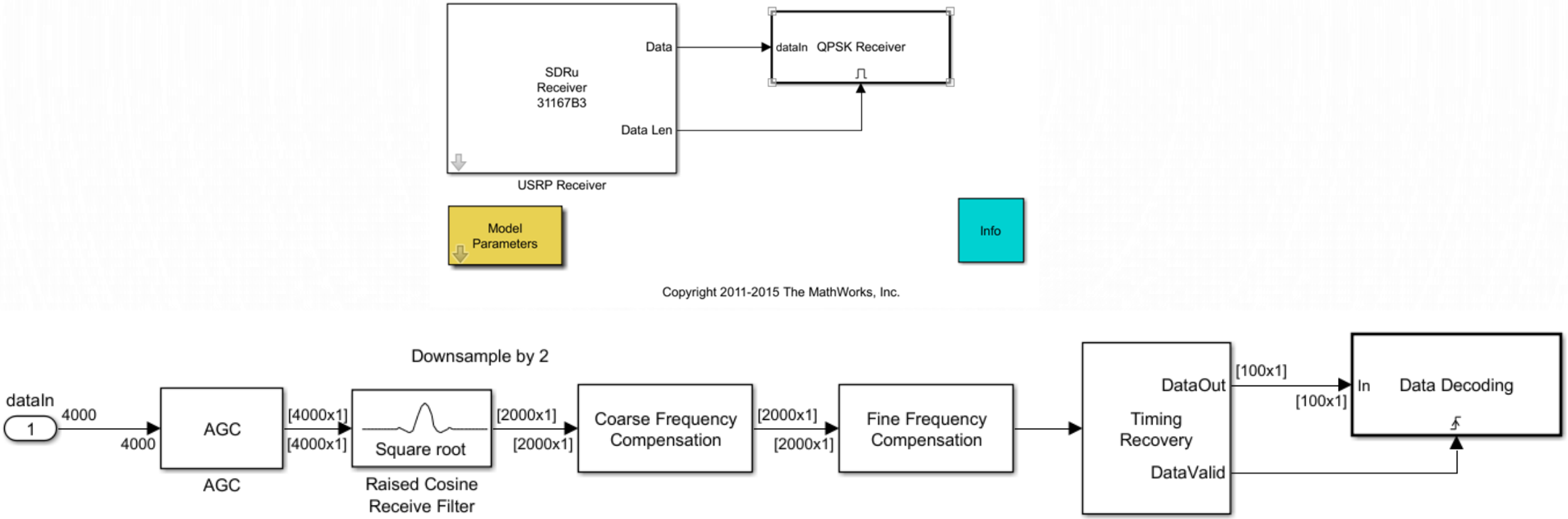
QPSK TX

- Bit generator





QPSK RX

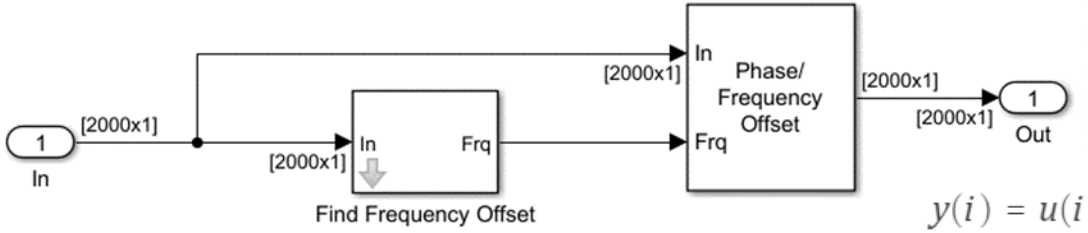


AGC: The AGC is placed before the Raised Cosine Receive Filter so that the signal amplitude can be measured with an oversampling factor of four.

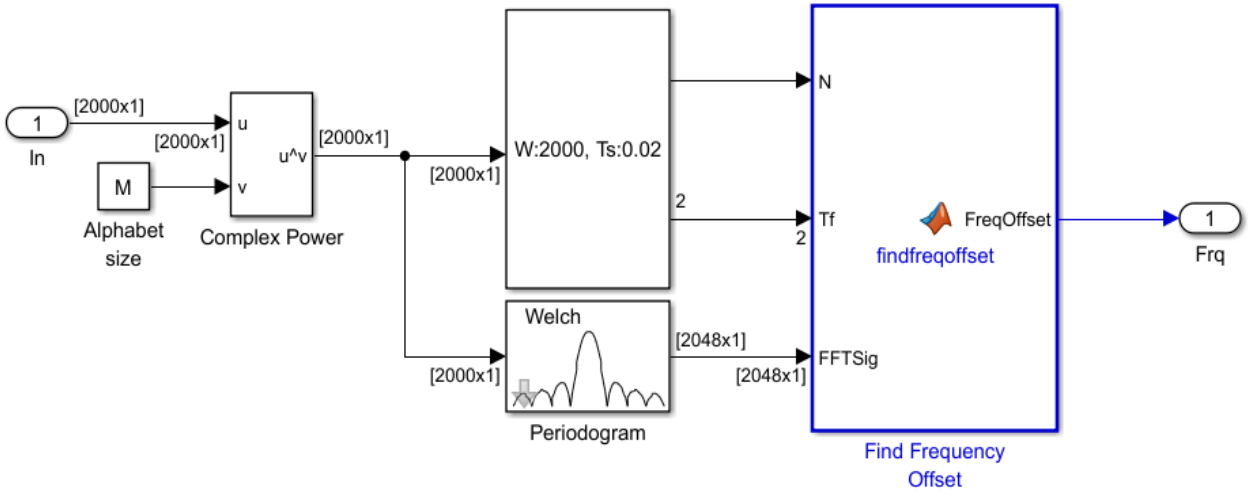
RRC: in this implementation decimate by 2

QPSK RX: COARSE FREQUENCY

- FFT implementation

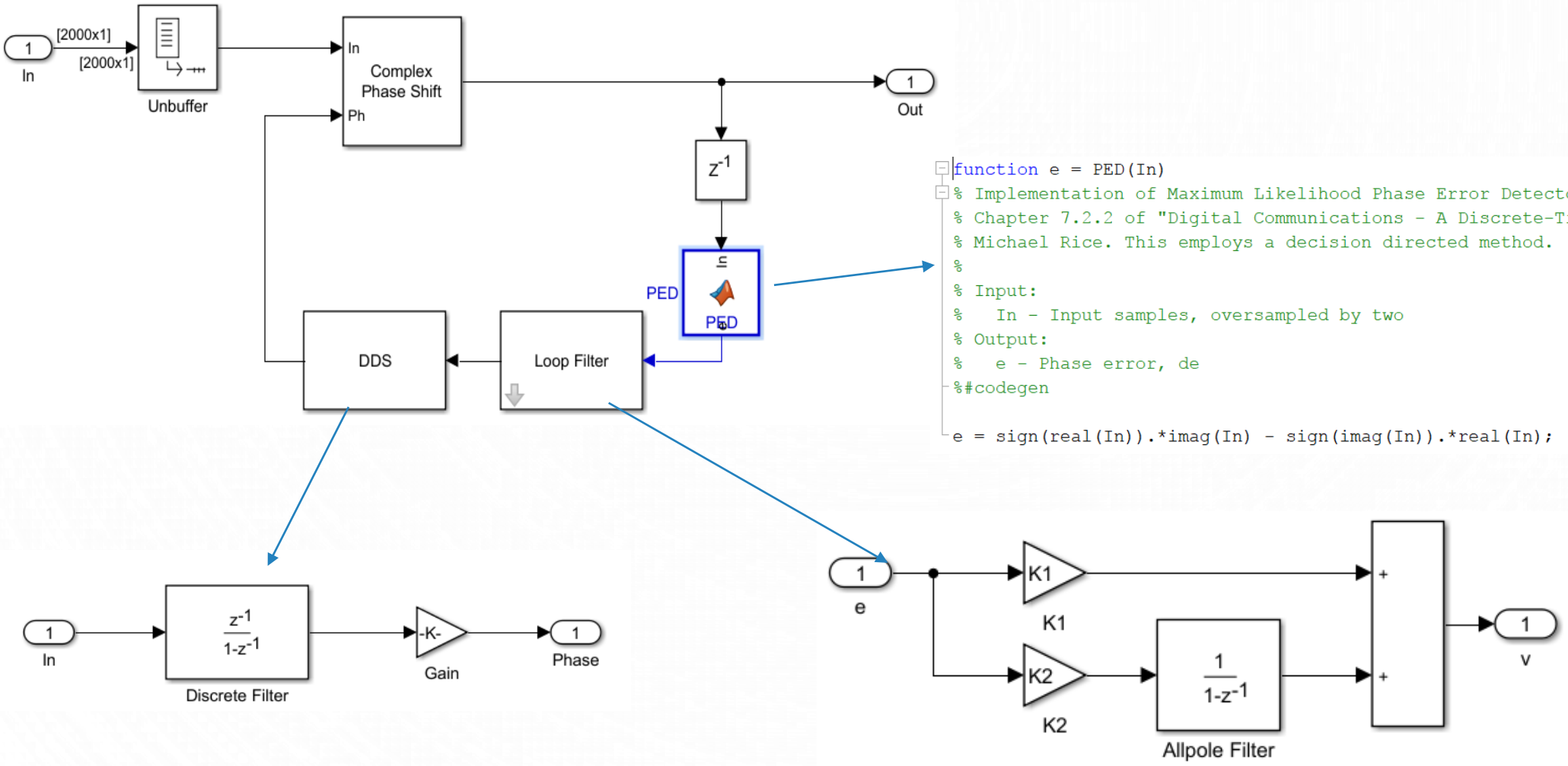


$$y(i) = u(i)\left(\cos\left(2\pi\sum_{n=0}^{i-1} f(n)\Delta t + \varphi(i)\right) + j\sin\left(2\pi\sum_{n=0}^{i-1} f(n)\Delta t + \varphi(i)\right)\right)$$



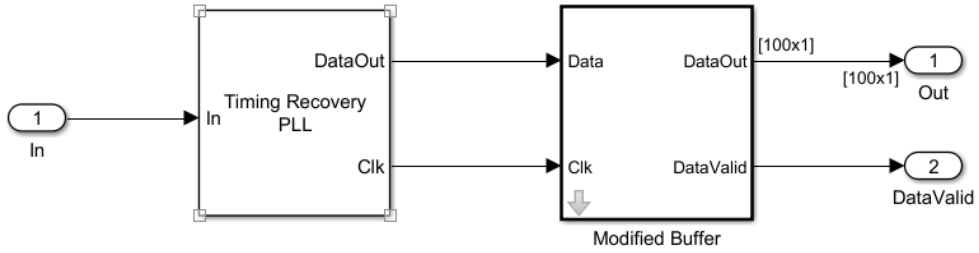

```
%  Estimate frequency offset.
%  findfreqoffset(N,Tf,fvec,params) returns the estimated frequency offs
%  given time-domain frame length N, sample time vector Tf containing
%  [frameTime,offsetTime], complex FFT vector FFTsig, and modulation
%  order M.

% Find index of max magnitude, convert to offset symmetric about zero.
% Return a negative integer offset to be interpreted as a compensation.
[~,maxIdx] = max(FFTSig);
FFTSize = size(FFTSig,1);
if maxIdx > FFTSize/2
    OffsetIdx = FFTSize+1-maxIdx;
else
    OffsetIdx = 1-maxIdx;
end
% Map offset index to a frequency value.
% Compute offset using modulation alphabet size params.M.
Ts = Tf(1)/N; % Sample time
delta_f = 1/(Ts*FFTSize); % Frequency resolution of FFT
FreqOffset = OffsetIdx * delta_f / M;
```

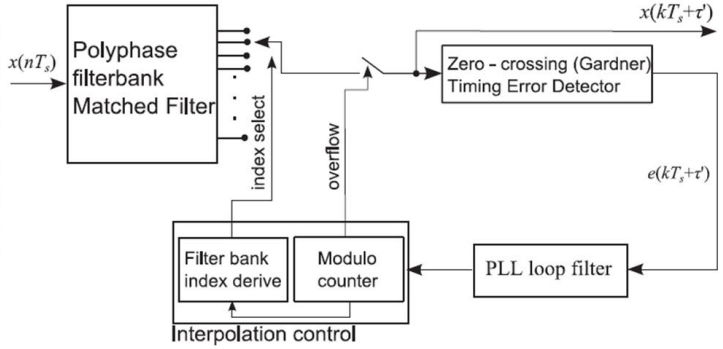
QPSK RX: FINE FREQUENCY

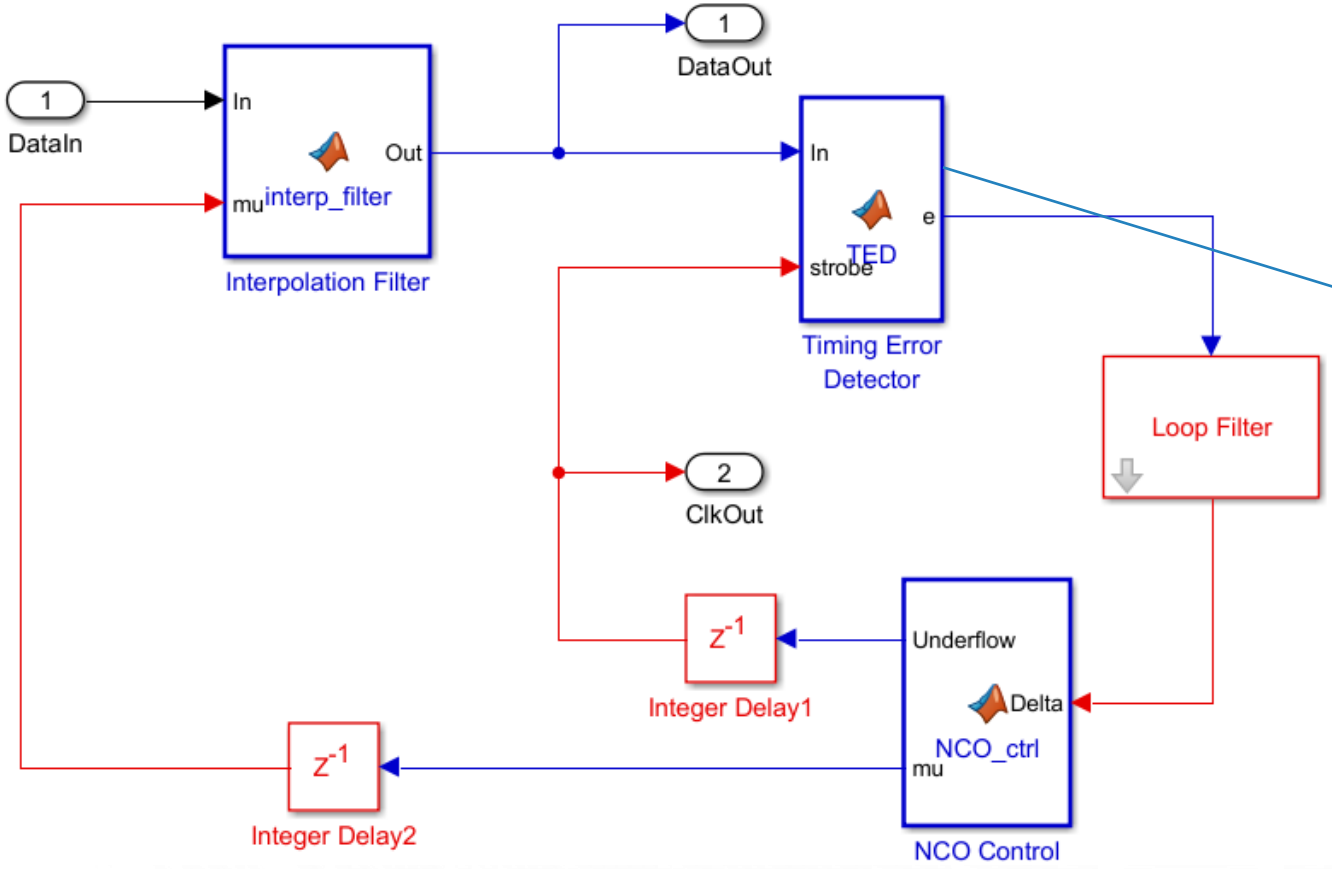



```
function e = PED(In)
% Implementation of Maximum Likelihood Phase Error Detector listed in
% Chapter 7.2.2 of "Digital Communications - A Discrete-Time Approach" by
% Michael Rice. This employs a decision directed method.
%
% Input:
%   In - Input samples, oversampled by two
% Output:
%   e - Phase error, de
%#codegen

e = sign(real(In)).*imag(In) - sign(imag(In)).*real(In);
```

QPSK RX: TIME RECOVERY





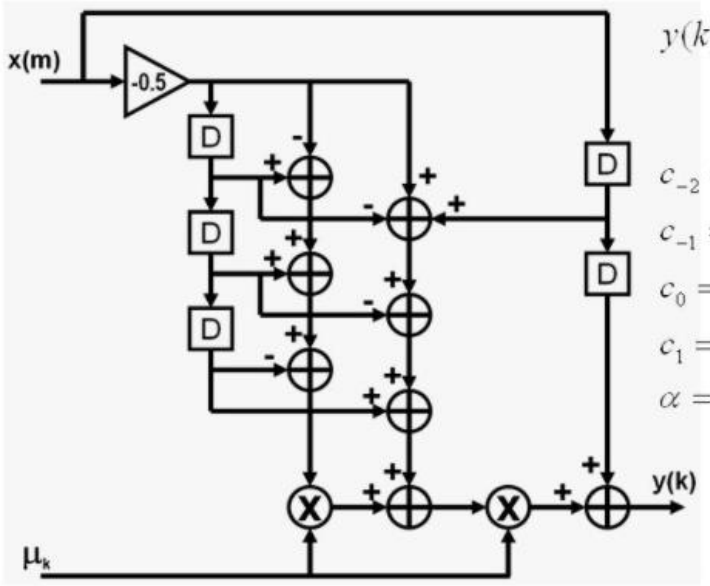


$$e(k) = x((k-1/2)T_s + \hat{\tau})\left[\hat{a}(k-1) - \hat{a}(k)\right]$$

```
if strobe==1 && delayStrobe~=strobe
    e = real(delay1) * (sign(real(delay2)) - sign(real(In))) + ...
        imag(delay1) * (sign(imag(delay2)) - sign(imag(In)));
else
    e = 0;
end

if delayStrobe~=strobe
    % Shift contents in delay register
    delay2 = delay1;
    delay1 = In;
elseif strobe==1
    % Two consecutive high strobes
    delay2 = 0; % Stuff missing sample
    delay1 = In;
end
```

QPSK RX: TIME RECOVERY

- Interpolation Filter



$$y(k) = c_{-2} \cdot x(m_k - 2) + c_{-1} \cdot x(m_k - 1) + c_0 \cdot x(m_k) + c_1 \cdot x(m_k + 1)$$

$$c_{-2} = \alpha\mu_k^2 - \alpha\mu_k$$

$$c_{-1} = -\alpha\mu_k^2 + (\alpha + 1)\mu_k$$

$$c_0 = -\alpha\mu_k^2 + (\alpha - 1)\mu_k + 1$$

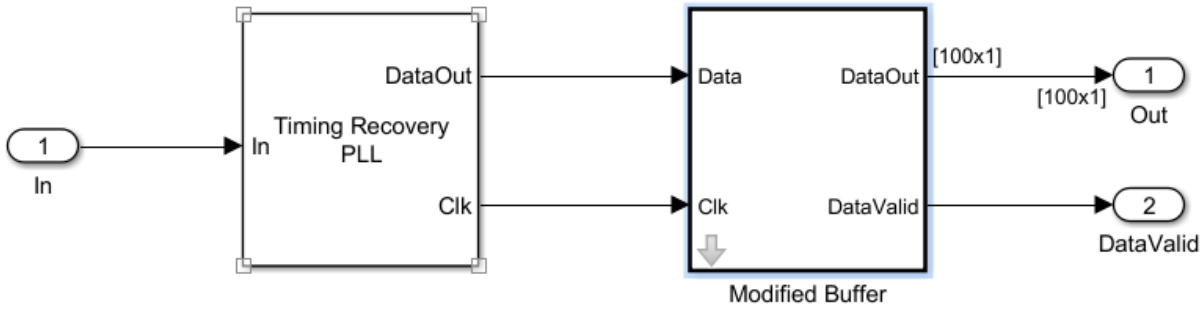
$$c_1 = \alpha\mu_k^2 - \alpha\mu_k$$

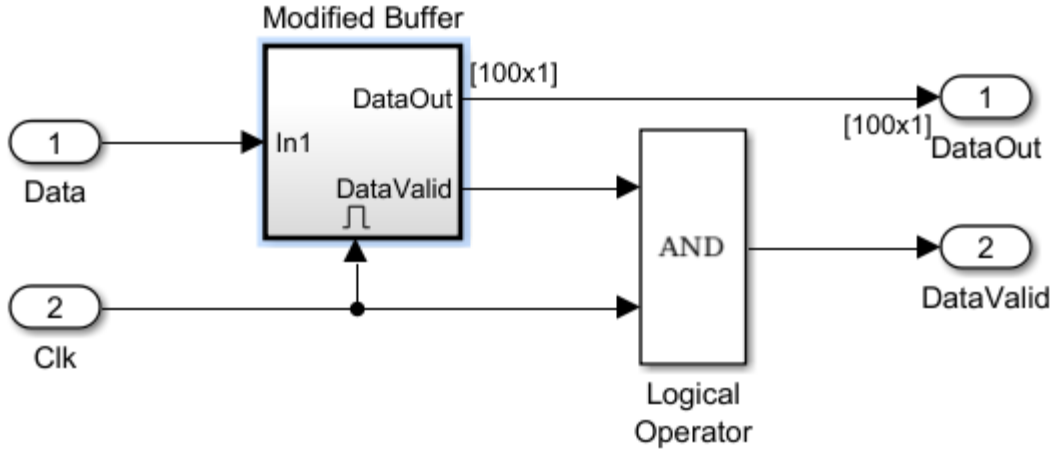
$$\alpha = -0.5 \quad for \quad Farrow \quad Structure$$

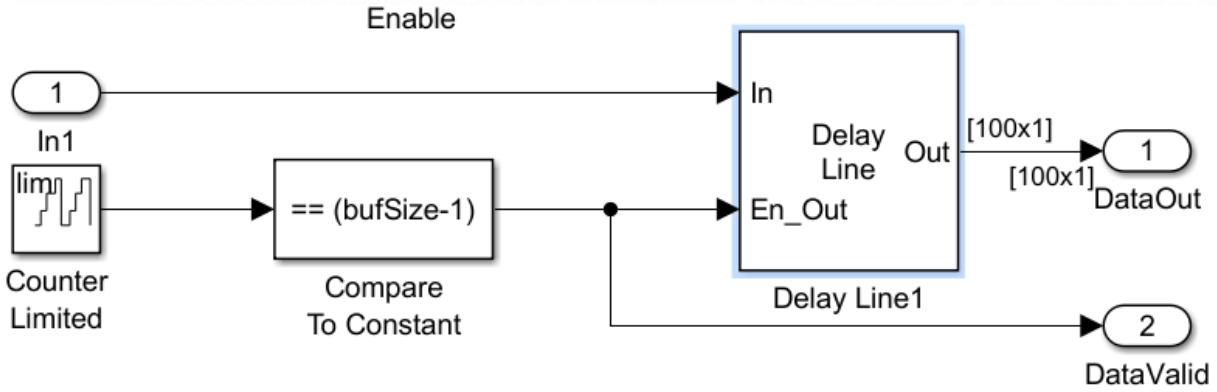
```
persistent delay1 delay2 delay3; % Input delayed by 1, 2 and 3 samples
if isempty(delay1)
    [delay1, delay2, delay3] = deal(complex(0,0));
end

K = -0.5;

Out = delay2 + mu.*(K.*(In+delay2+delay3)+(1-K).*delay1)...
      + K.*(delay1+delay2-In-delay3).*mu.^2;

% Update delay buffers
delay3 = delay2;
delay2 = delay1;
delay1 = In;
```

QPSK RX: TIME RECOVERY

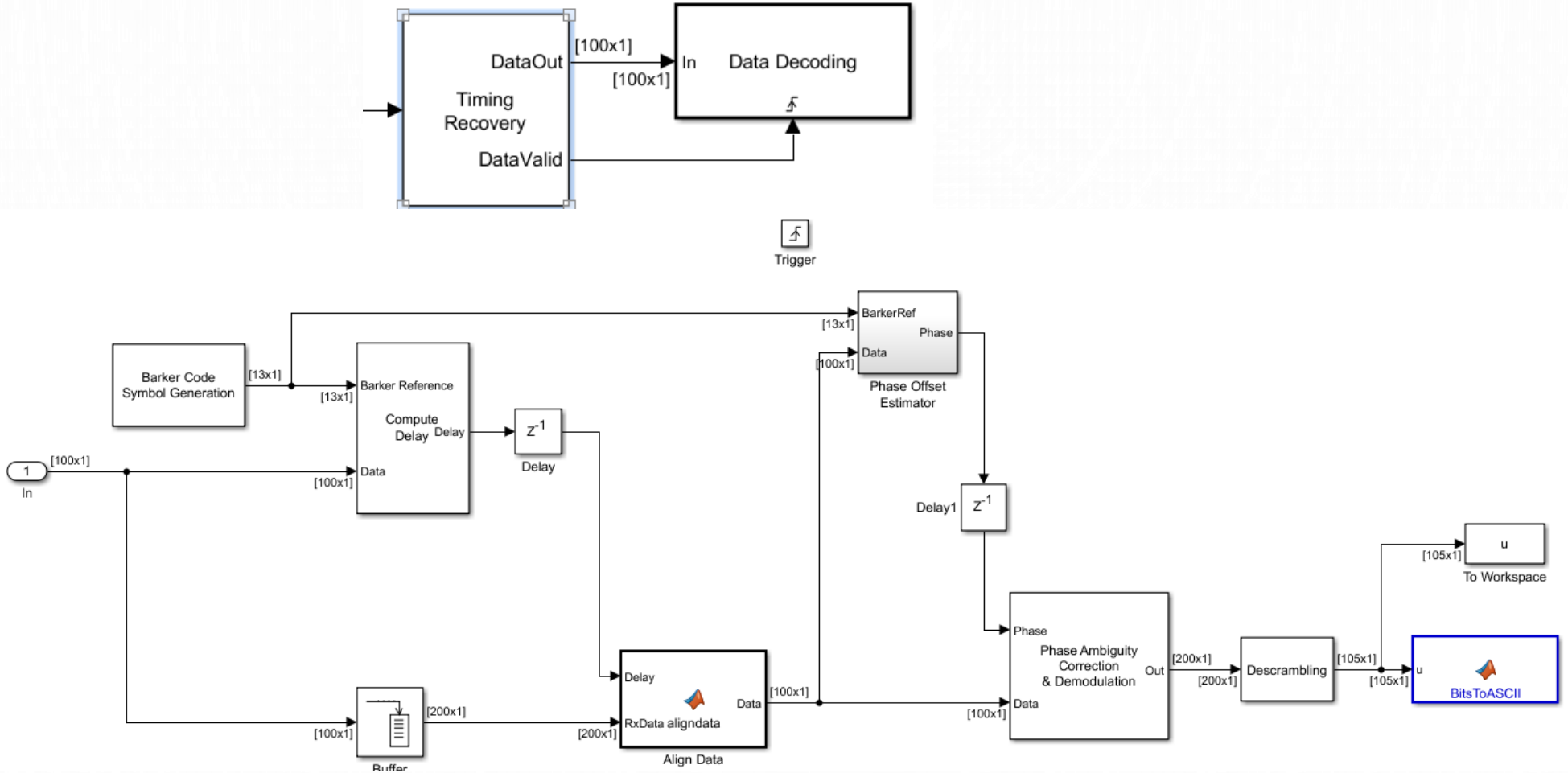
- SYMBOL RECOVERY





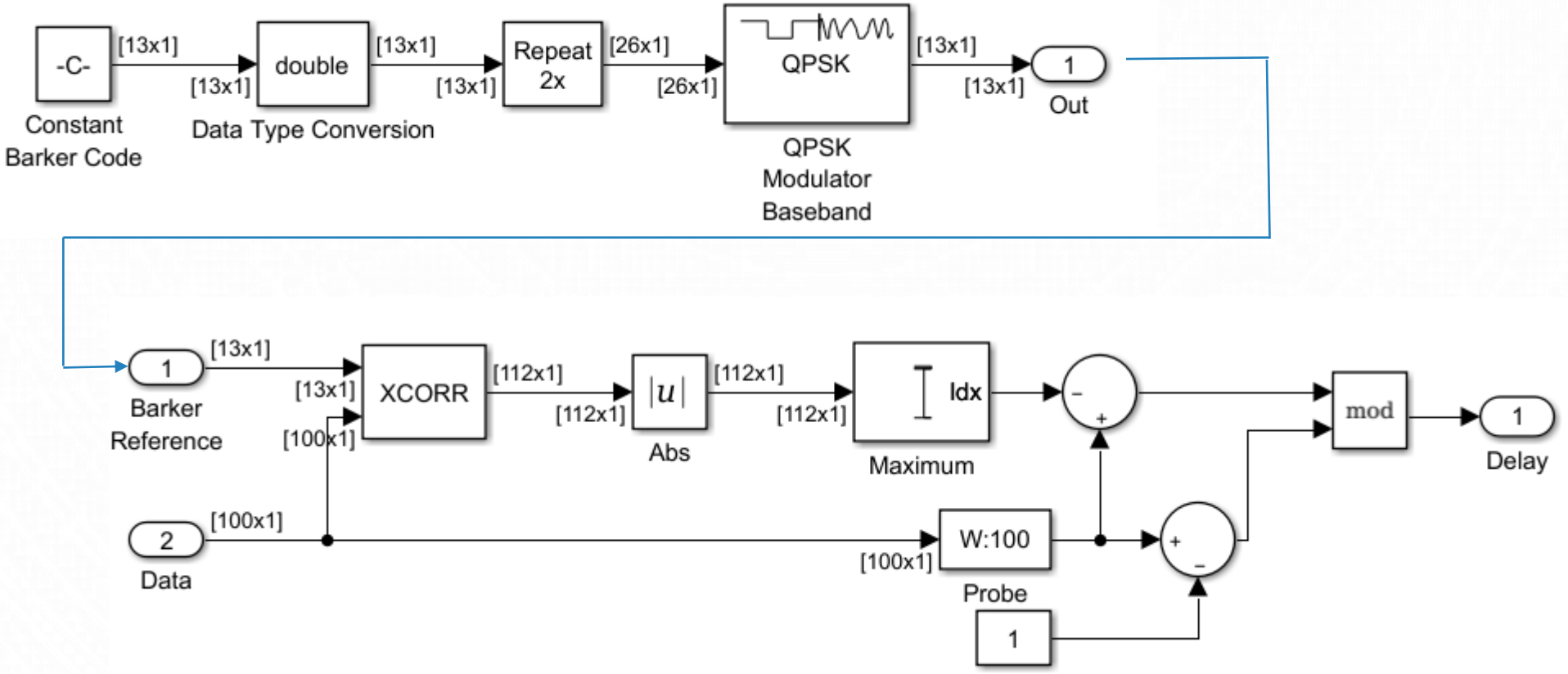


QPSK RX: DATA DECODING



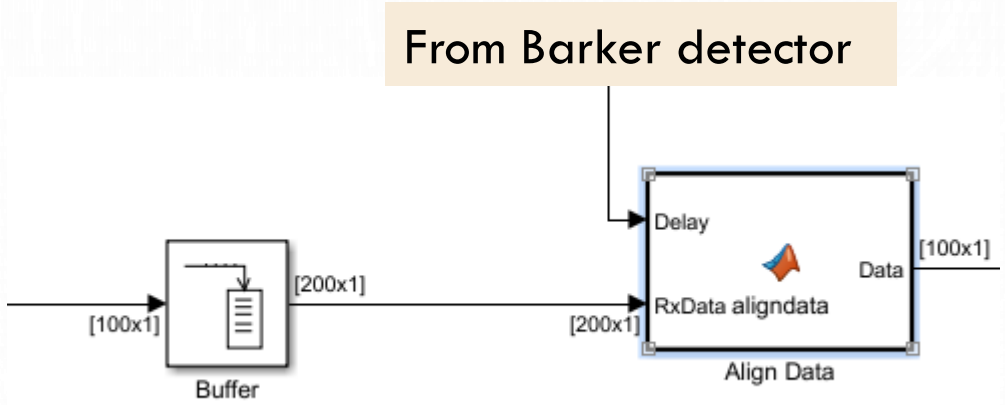
QPSK RX: DATA DECODING

- FRAME SYNC



QPSK RX: DATA DECODING

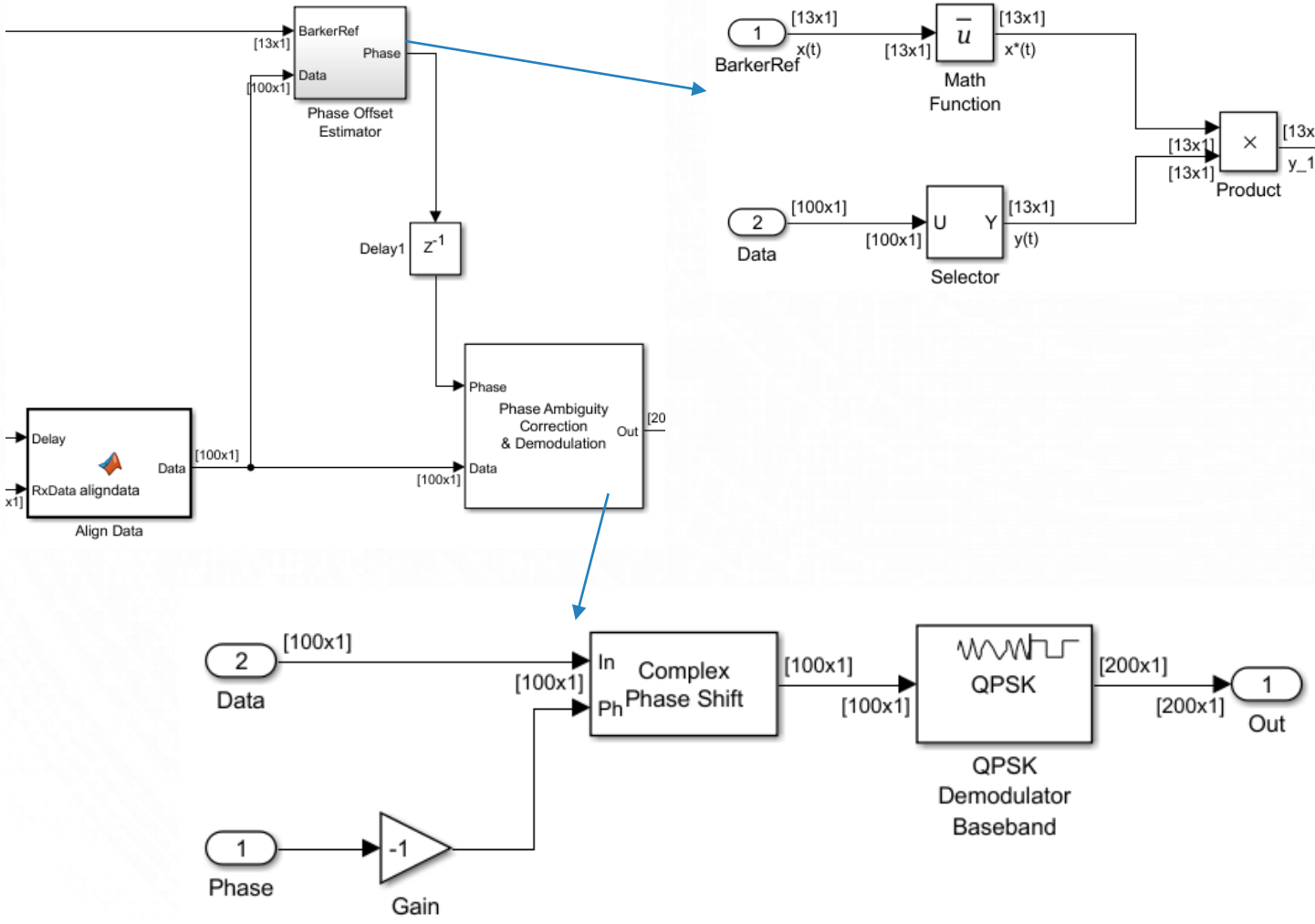
- Frame Sync

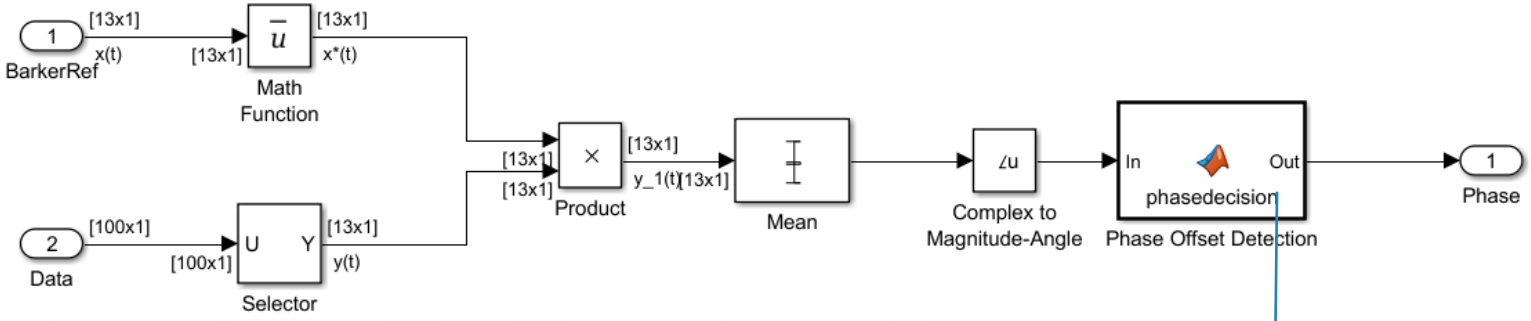



```
function Data  = aligndata(Delay, RxData)
% Extract data from the buffered RxData, using the
% start
%
% Input:
% Delay - Index of frame start
% RxData - Buffered symbol-spaced data
% Output:
% Data - Data with the Barker code at frame start
%#codegen
Data = RxData(Delay+1:Delay+length(RxData)/2);
```

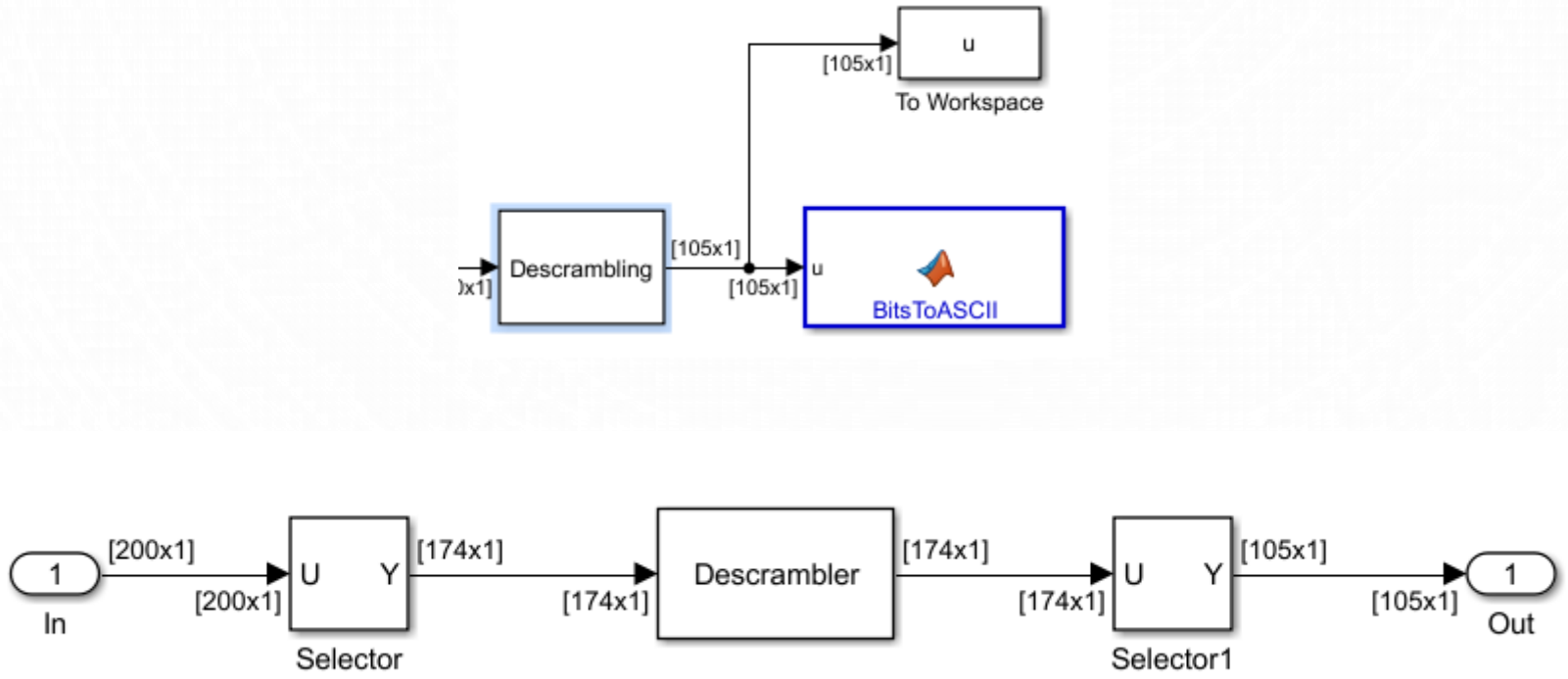
QPSK RX: DATA DECODING

- Phase ambiguity correction: Data aided






```
if In<pi/4 && In>=-pi/4
    Out = 0;
elseif In>=pi/4 && In<pi*3/4
    Out = pi/2;
elseif In>=pi*3/4 || In<-pi*3/4
    Out = -pi;
else
    Out = -pi/2;
end
```

QPSK RX: DATA DECODING

- Header remove & Descramble
- Send data to workspace for ASCII conversion