FPGA Implementation of an OFDM PHY

Dr Chris Dick
DSP Chief Architect
Director, Signal Processing Systems Engineering

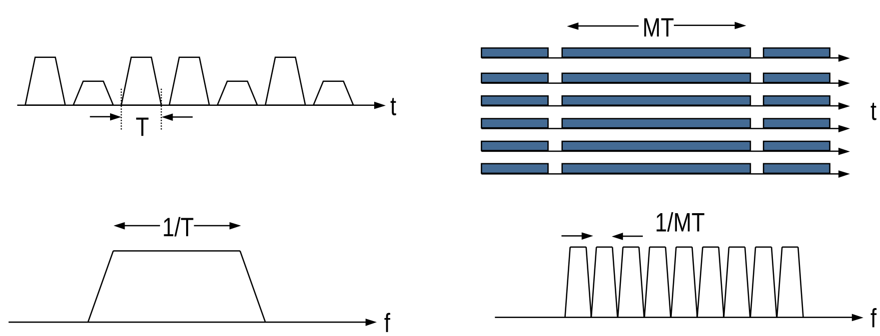
Introduction

- Orthogonal frequency division multiplexing (OFDM) increasingly found in wired and wireless communication systems
  - Digital television
  - High-speed wired connections
  - Wireless local area networks
  - 4G wireless … if we ever recover from/deliver 3G
- Key signal processing technology moving forward in many spaces
  - Mil/aero
  - commercial

OFDM Primer

- In conventional single carrier systems data symbols are transmitted serially using some modulation scheme and the spectrum of each symbol is allowed to occupy the entire channel bandwidth

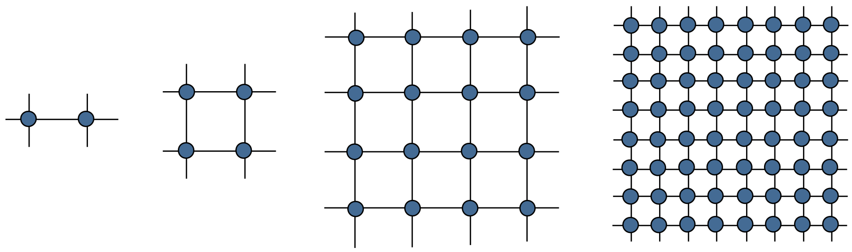
Single Carrier Vs Multi-Carrier Systems



Sequential transmission of waveforms	Parallel transmission of waveforms
Waveforms are short duration T	Waveforms are long duration MT
Waveforms occupy full transmission bandwidth 1/T	Waveforms occupy 1/M-th of system bandwidth 1/T

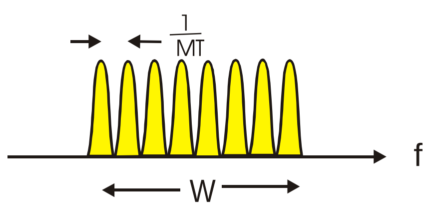
OFDM Systems

- OFDM modulation consists of multiplexing QAM data symbols over a large number of orthogonal carriers



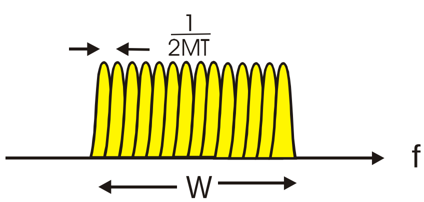
BPSK QPSK 16-QAM 64-QAM

OFDM Dense Multichannel System



Conventional Multichannel System
Non Overlapping Adjacent Channels.

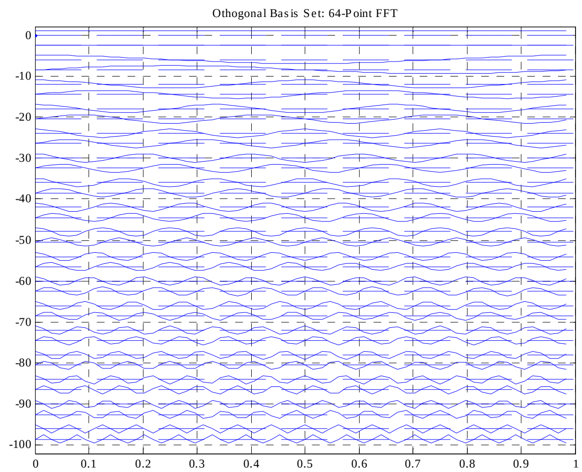
Channels separated by More
Than Their Two Sided bandwidth



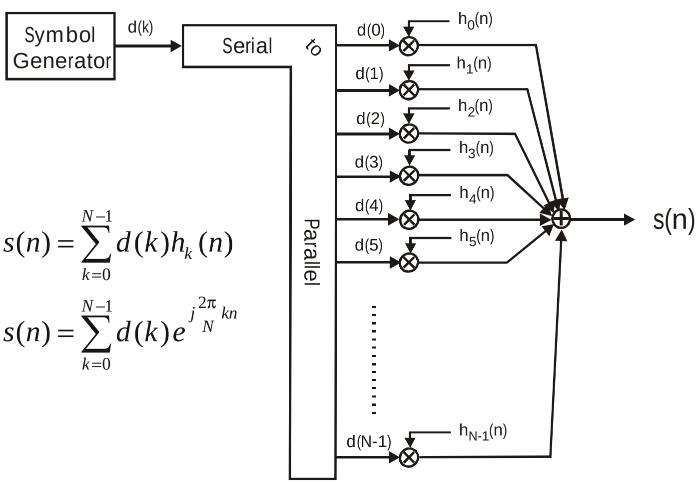
OFDM Multichannel System
50% Overlap of Adjacent Channels
Available bandwidth is Used Twice

Channels separated by Half
Than Their Two Sided bandwidth

Orthogonal Basis Functions



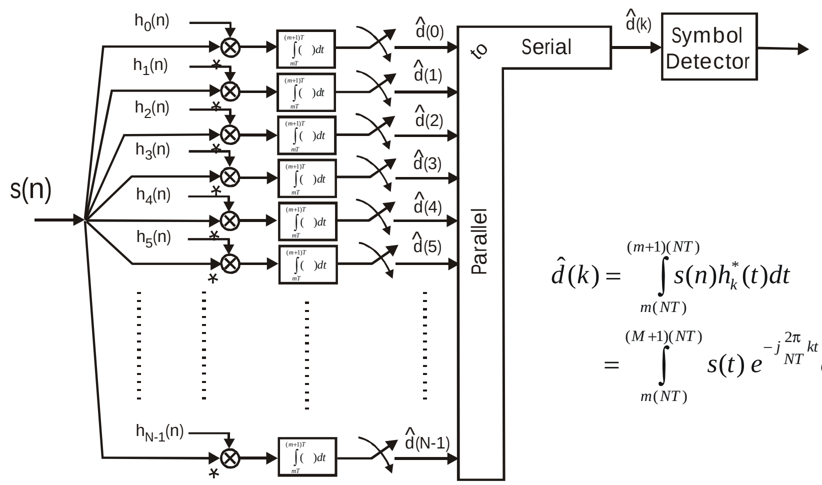
OFDM Modulator



$$s(n) = \sum_{k=0}^{N-1} d(k) h_k(n)$$

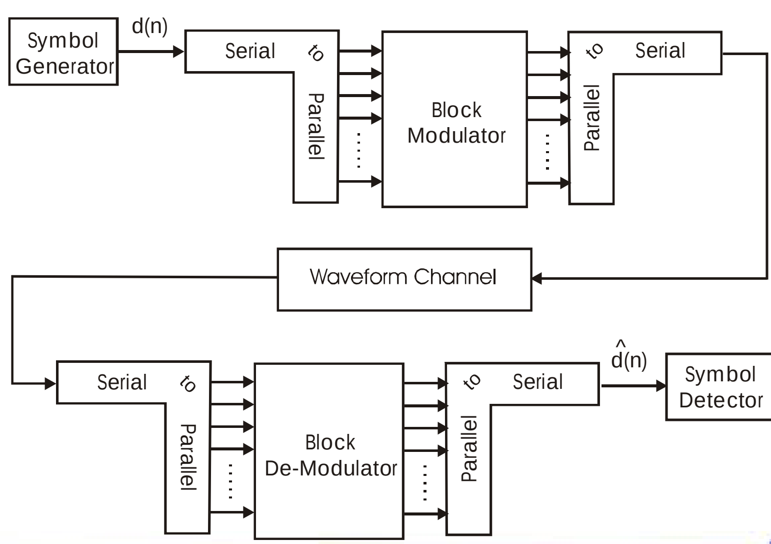
$$s(n) = \sum_{k=0}^{N-1} d(k) e^{j\frac{2\pi}{N}kn}$$

OFDM Demodulator



$$\hat{d}(k) = \int_{m(NT)}^{(m+1)(NT)} s(n) h_k^*(t)\,dt = \int_{m(NT)}^{(M+1)(NT)} s(t)\, e^{-j\frac{2\pi}{NT}kt}\,dt$$

OFDM is a Block Process



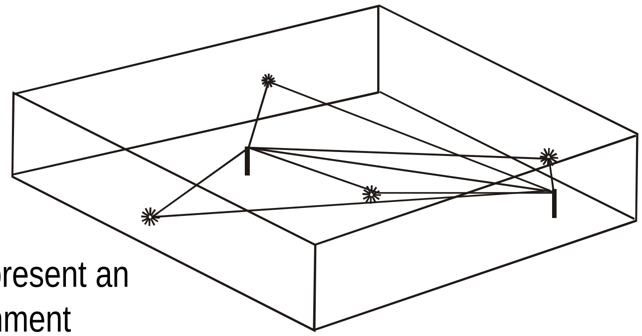
Advantages of OFDM

- Efficiently deals with multipath fading
- Efficiently deals with channel delay spread
- Enchanced channel capacity
- Adaptively modifies modulation density
- Robust to narrowband interference

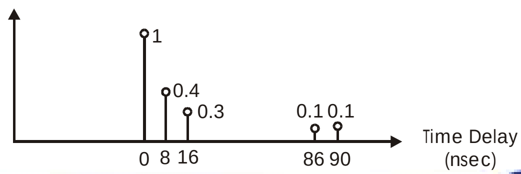
Disadvantages of OFDM

- Sensitive to small carrier frequency offsets
- Exhibits high peak-to-average power ratio
- Sensitive to high frequency phase noise
- Sensitive to sampling clock frequency offsets

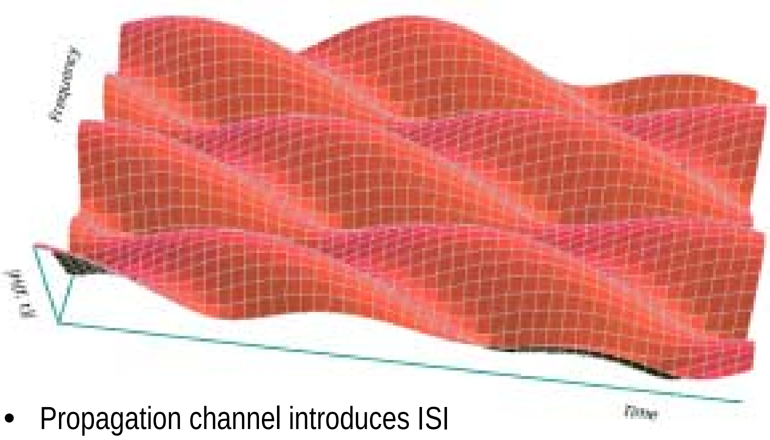
Multipath Model/Impulse Response



- This could represent an indoor environment



Radio Propagation Channel

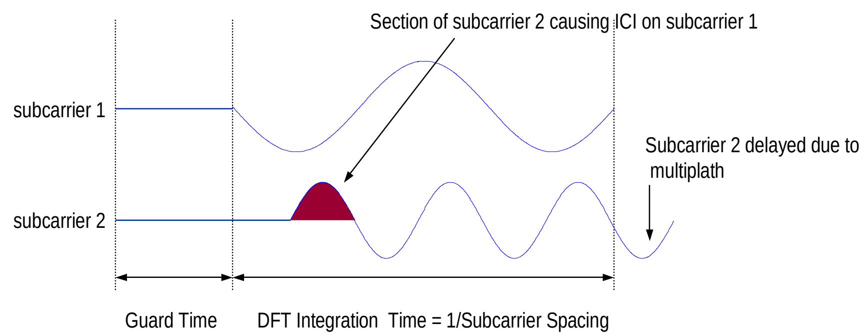


- Propagation channel introduces ISI
- Complex response a function of time and frequency
- OFDM is very robust against frequency selective fading channels

Dealing with ISI

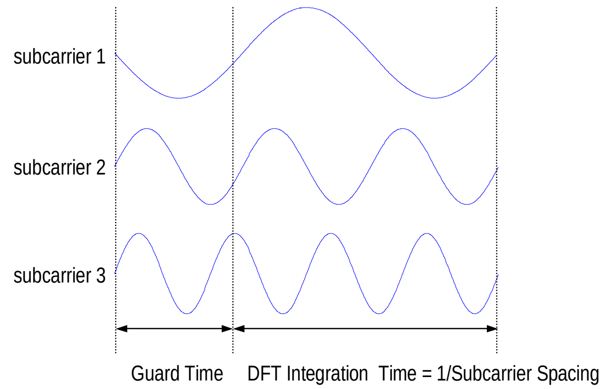
- Designing receivers to accommodate ISI is an important aspect of narrowband and wide band systems
- QAM receiver deals with ISI in the time domain using an adaptive equalizer
  - Typically a fractionally spaced
    - FFE
    - FBE
    - FFE & FBE (usual case)

Dealing with ISI in OFDM



- Non integer number of cycles difference between subcarrier 1 and 2
  - Loss of orthogonality

Cyclic Extension of FFT Data

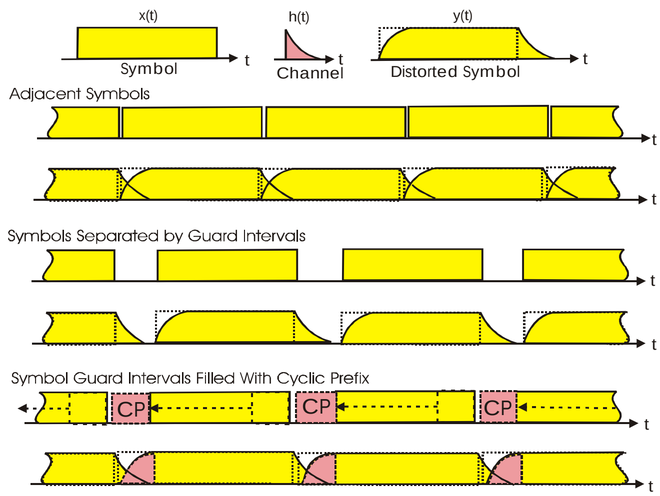


- Periodic extension of data ensures that there is always an integer number of cycles of each subcarrier within the FFT integration window

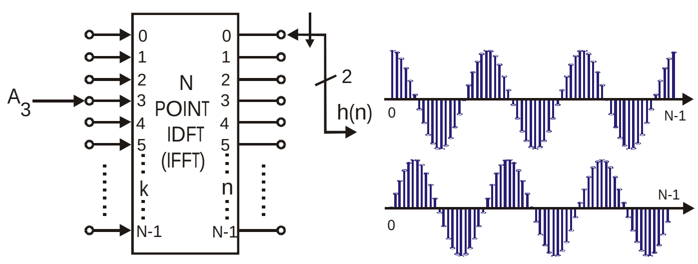
Cyclic Prefix

- Guard interval will now be called the cyclic prefix
- Cyclic prefix (CP) is a crucial feature of OFDM used to combat inter-symbol interference (ISI) and inter-(sub)carrier interference (ICI)
- Build a periodic extension of the FFT symbol by preappending a block of samples from the end of the FFT frame to the beginning
- Duration of the guard period should be longer than the worst case channel delay spread

Combatting ISI with the CP



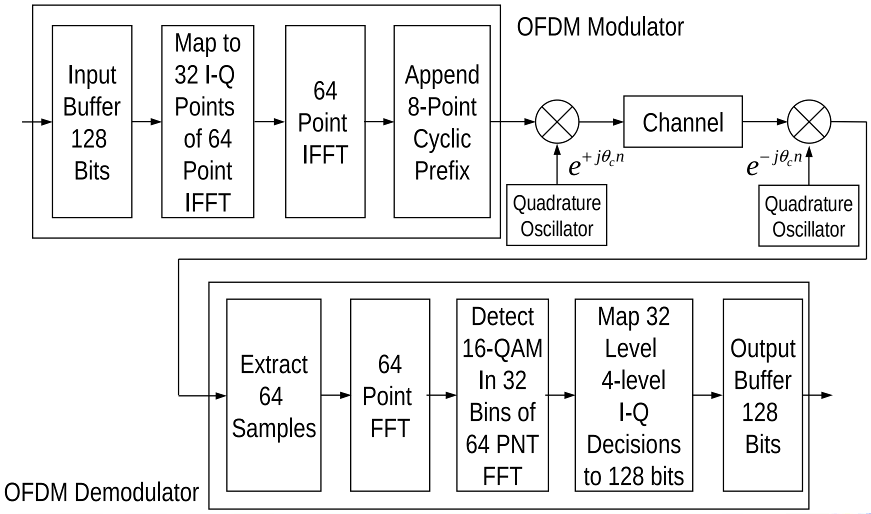
FFT as Sig Gen And Modulator



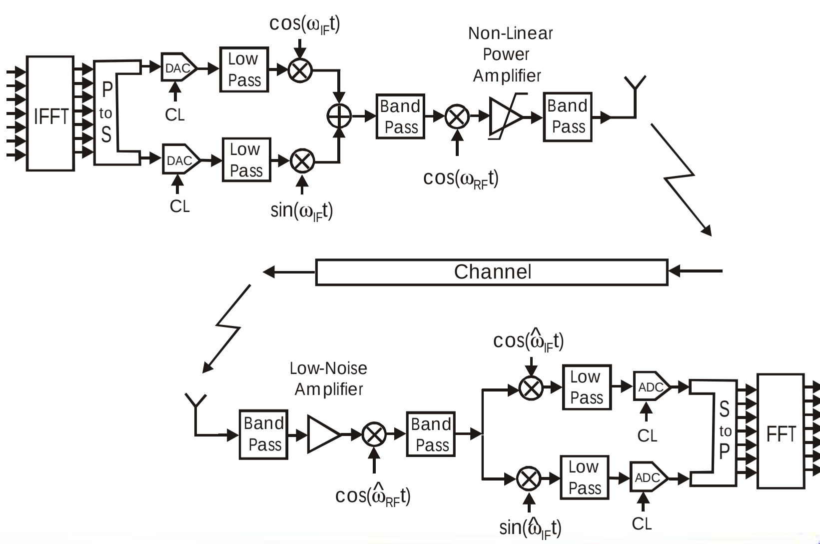
$$h(n) = \frac{1}{N}\sum_{n=0}^{N-1} H(k)\, e^{j\frac{2\pi}{N}nk} \;:\quad n = 0,1,2,\ldots,N-1$$

- FFT is one of the key kernals in the OFDM mod (IFFT) and demod (FFT)
- Viewed as the signal generator and the (de-)modulator

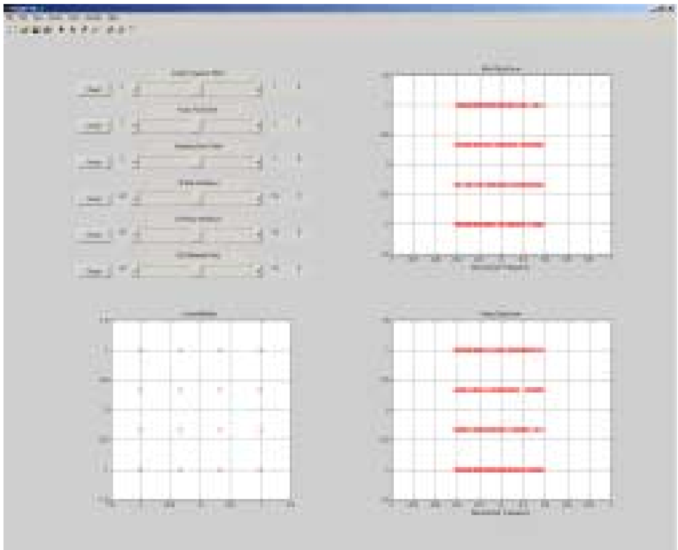
OFDM: Top Level View



OFDM Transceiver



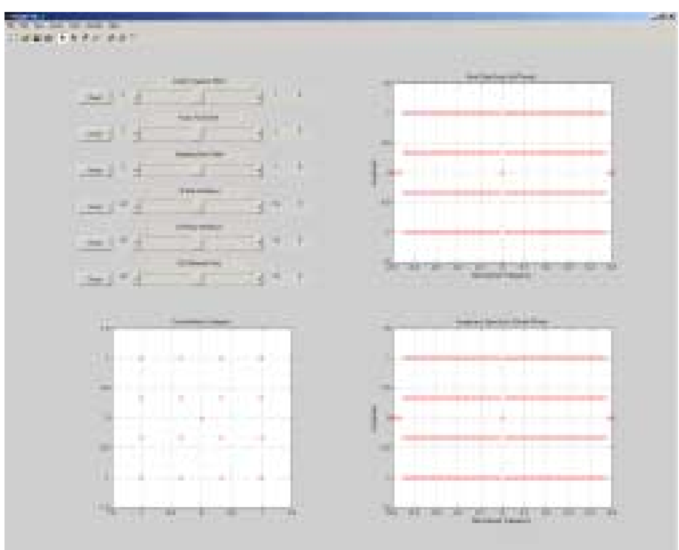
Simulation Baseline



- The Perfect World

OFDM Signal Processing 23

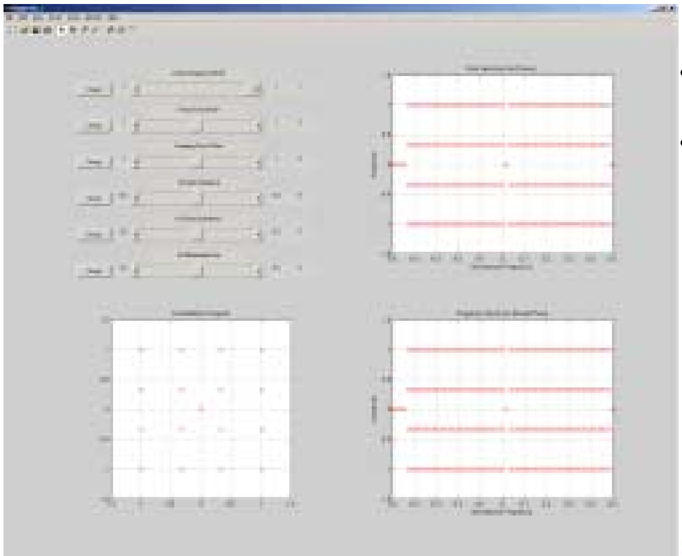
Some Subcarriers Unused



- Don't tx DC
- Possibly reserve edge frequencies to accommodate analog signal processing

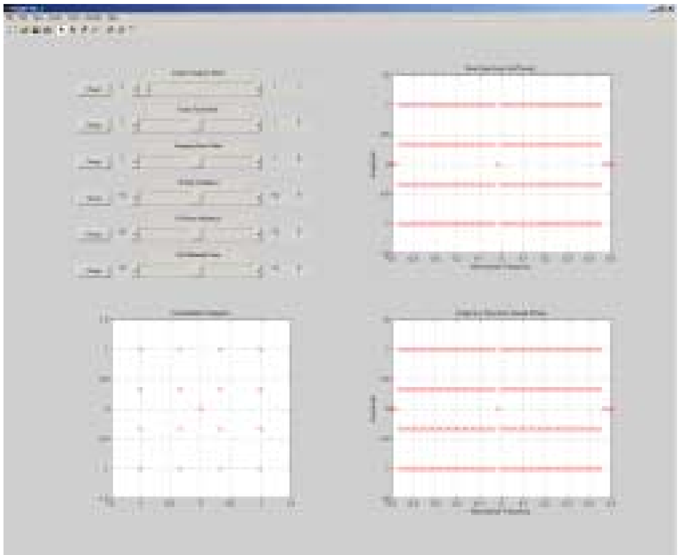
OFDM Signal Processing 24

Carrier Frequency Offset (CFO)



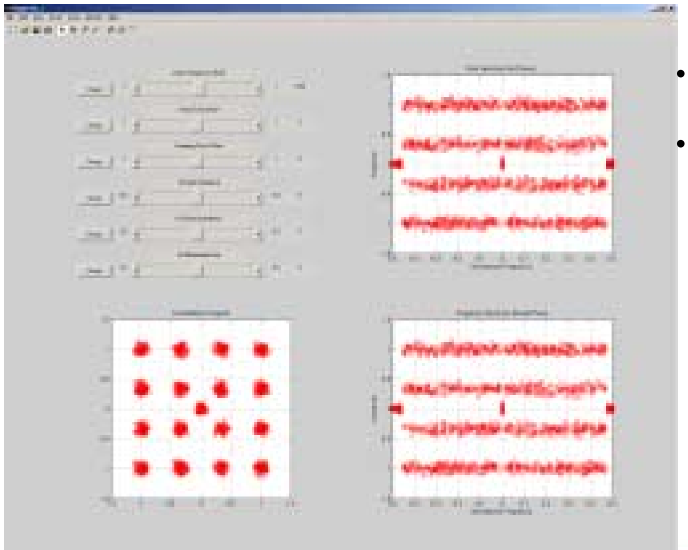
- CFO = multiple of the subcarrier spacing
- If not corrected will lead to detection errors

Carrier Frequency Offset (CFO)



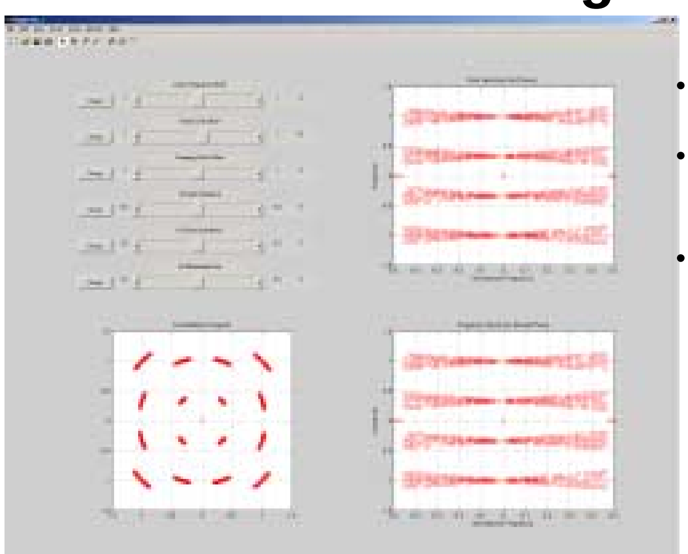
- CFO = multiple of the subcarrier spacing
- CFO in opposite direction to previous example
- If not corrected will lead to detection errors

Carrier Frequency Offset (CFO)



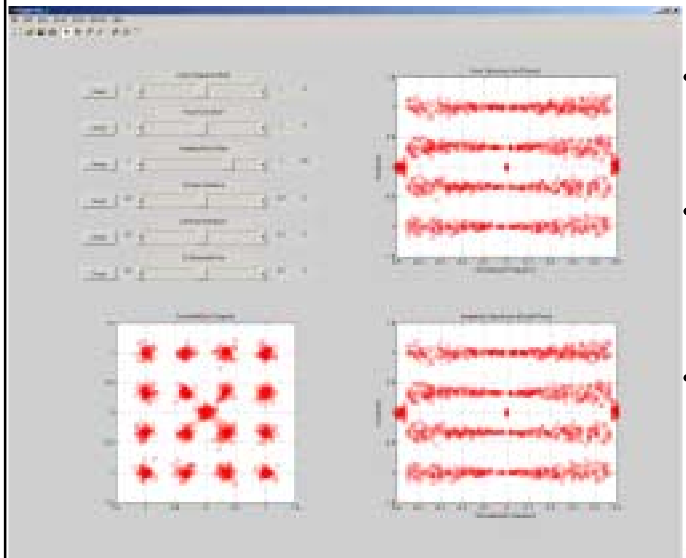
- CFO = fraction of the subcarrier spacing
- If not corrected will lead to detection errors due to degraded margin in detector

Packet Framing Error



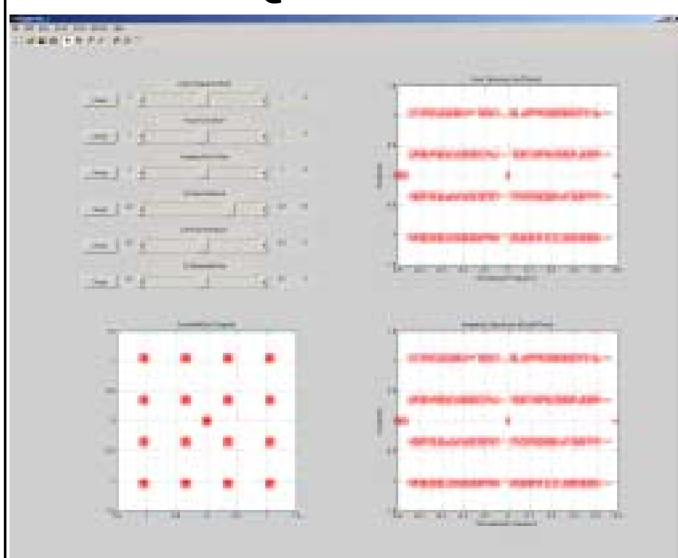
- Incorrect OFDM symbol boundary detection
- Manifests itself as what looks like phase noise in the constellation
- If not corrected will lead to detection errors due to degraded margin in detector

Sample Clock Frequency Error



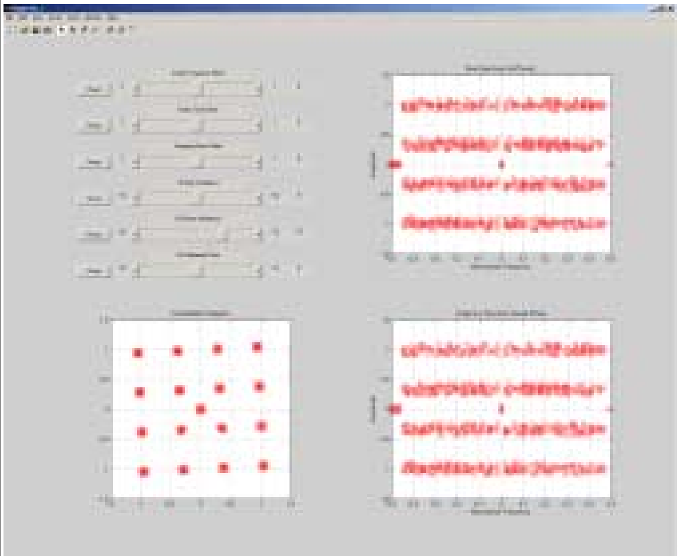
- CFO results in a constant phase rotation from one symbol to the next
- Sample clock offset produces a phase offset that that increases as a function of the subcarrier index
- If not corrected will lead to detection errors due to degraded margin in detector

I-Q Mixer Imbalance



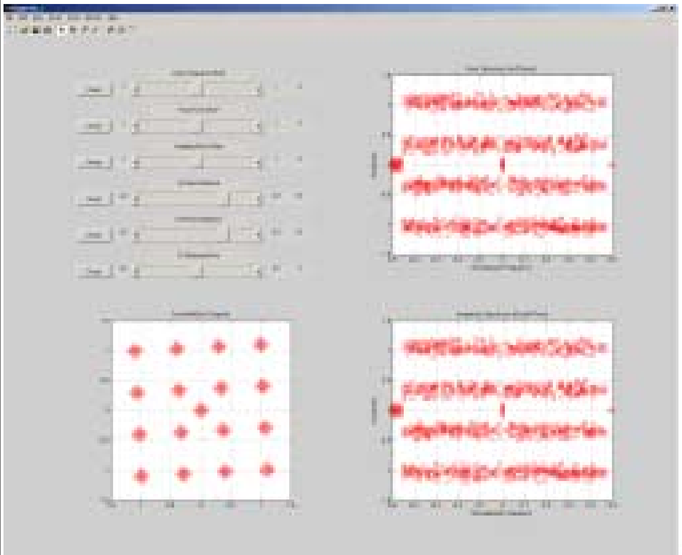
- Gain mismatch in I-Q mixers
- If not corrected will lead to detection errors due to degraded margin in detector

I-Q Mixer Imbalance



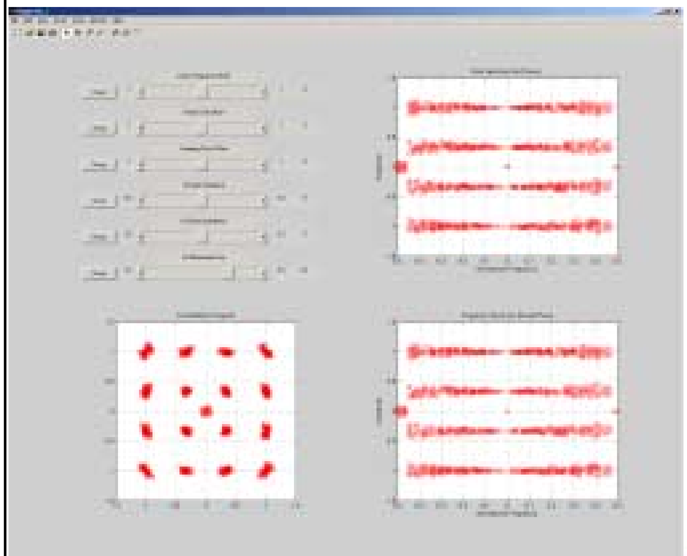
- Phase mismatch in I-Q mixers
- If not corrected will lead to detection errors due to degraded margin in detector

I-Q Mixer Imbalance



- Phase and gain mismatch in I-Q mixers
- If not corrected will lead to detection errors due to degraded margin in detector

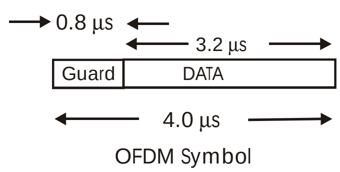
I-Q Differential Time

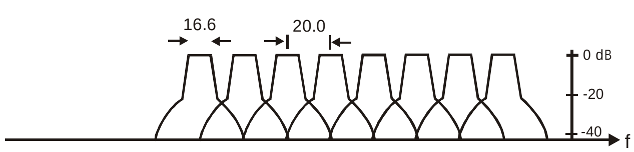


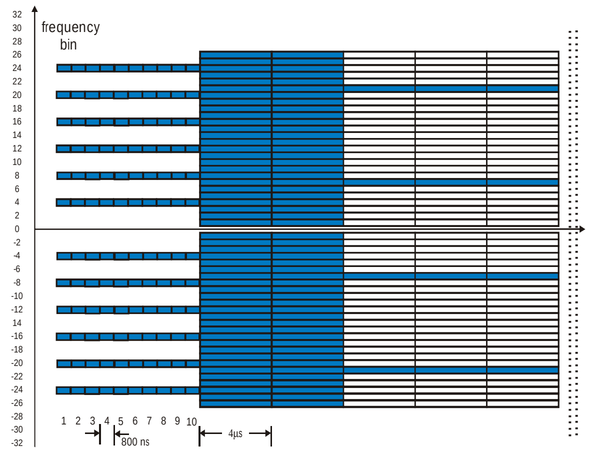
- Pracitcal I-Q mixers introduce a time offset between the I and Q signal components
- Differential time
- Could be due to signal propagation path differences in a mixed-signal RF device
- If not corrected will lead to detection errors due to degraded margin in detector

Example: 802.11a

OFDM SYMBOL DURATION	**4 µsec**
Guard Interval	**800 nsec**
Orthogonal Tone Duration	**3.2 µsec**
Spacing Between Tone Centers	**312.5 kHz {1/3.2 µsec)**
Number of Data Subcarriers	**48**
Number of Pilot Subcarriers	**4**
Total Number of Subcarriers	**52**
Total Bandwidth	**16.56 MHz {(52+1)*312.5 kHz)}**
Modulation	**BPSK, QPSK, 16-QAM, 64-QAM**
Coding Rate	**1/2, 2/3, 3/4**
Coded Data Rates	**6, 9, 12,18, 24, 36, 48, 54 Mbps**
Channel Spacing	**20 MHz**

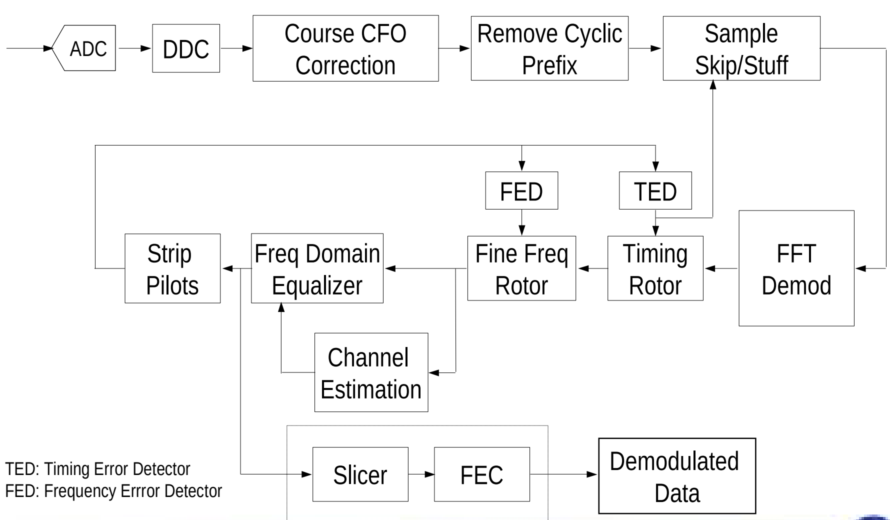






Time-Frequency (Bin) Profile of 802.11a OFDM Subcarriers
Gray Subcarriers Contains Known Pilot Signals

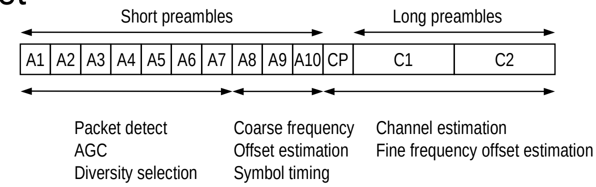
OFDM Receiver Architecture



TED: Timing Error Detector
FED: Frequency Errror Detector

Preamble for Packet Detection

- Preambles of 802.11a and HyperLAN/2 have been designed to help the detection of the start of a packet



- A1-A10 are short training symbols all identical and 16 samples in duration

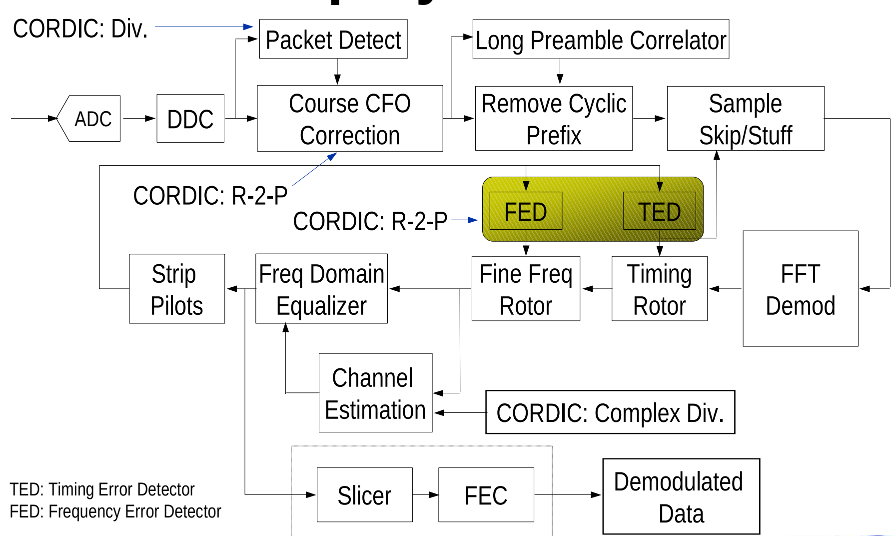
OFDM Arithmetic Resourcing (1)

- Several OFDM functions require arithmetic resourcing for processes other than
  - Multi-rate filters
  - (I)FFT
- Synchronization/Equalization require
  - Division – Real
  - Division – Complex
  - Rectangular-to- Polar transformation

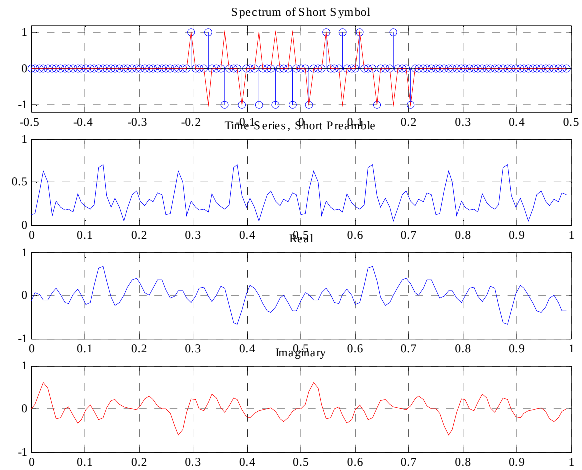
OFDM Arithmetic Resourcing (2)

- Peak-to-average power ratio (PAR) control in mod
  - Complex vector magnitude
  - Rotation of PAR control time-series
- Best method to support these arithmetic functions in FPGAs?
  - CORDIC particularly useful
  - Add/Sub/Shift requirements supported very well in the FPGA device architecture

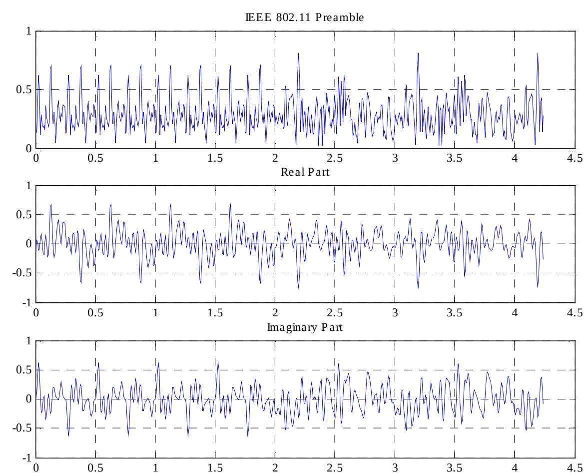
CORDIC Deployed in OFDM PHY



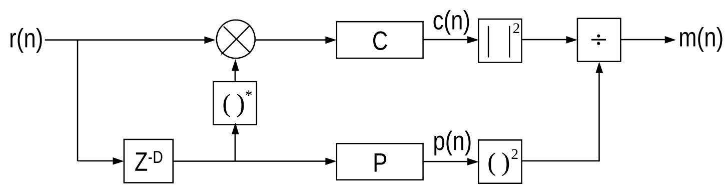
Short Preamble



Long Preamble



Schmidl and Cox Correlator[1]



- Delay and correlate algorithm
- Exploits periodicity of short training symbols
- C is a cross-correlation between r(n) and r(n-D) (D=16 for 802.11a)
- P calculates the received signal energy during the cross-correlation window

[1] T. M. Schmidl, D. C. Cox, "Low-Overhead, Low Complexity (Burst) Synchronization for OFDM", *IEEE International Conference on Communications,* Vol. 3., 1996, pp. 1301-1306.

Schmidl and Cox Correlator

$$c(n)=\sum_{k=0}^{L-1} r(n+k)r^*(n+k+D)$$

$$p(n)=\sum_{k=0}^{L-1} r(n+k+D)r^*(n+k+D)=\sum_{k=0}^{L-1}\left|r(n+k+D)\right|^2$$

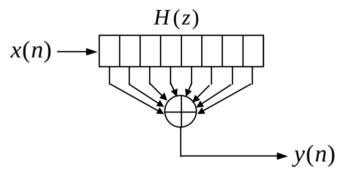
Decision statiestic $m(n)$ is calculated as

$$m(n)=\frac{\left|c(n)\right|^2}{p(n)^2}$$

- c(n) and p(n) are sliding windows so a recursive procedure can be used to reduce the computation load

Packet Detector Implementation

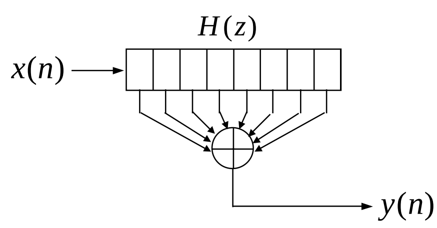
- The C and P correlators in the packet detector utilize boxcar filters



- Recursive procedure for computationally efficient implementation

Packet Detector Implementation

- The C and P correlators in the packet detector utilize boxcar filters



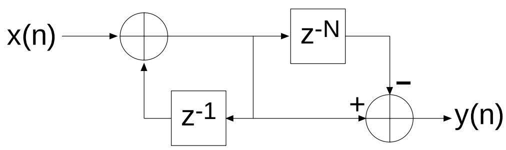
- Recursive procedure for computationally efficient implementation

Packet Detector Implementation

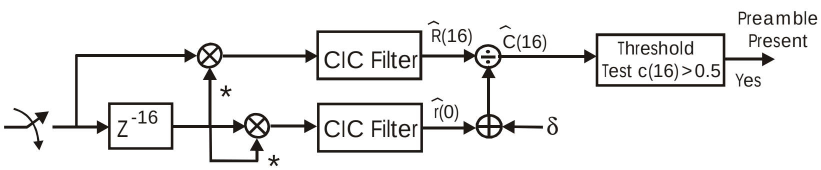
$$H(z)=\frac{Y(z)}{X(z)}=\sum_{n=0}^{N-1} z^{-n}=\frac{1-z^{-N}}{1-z^{-1}}$$

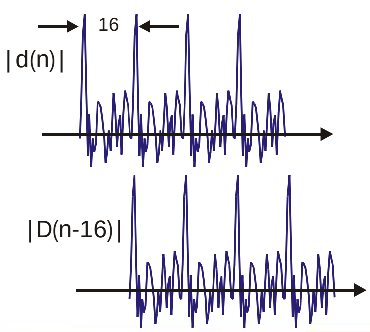
$$=\underbrace{\frac{1}{1-z^{-1}}}_{\text{INTEGRATOR}}\cdot\underbrace{\frac{1-z^{-N}}{1}}_{\text{COMB FILTER}}$$

- Implement average in correlators as CIC



Preamble Detector Using Repeated Short Symbols



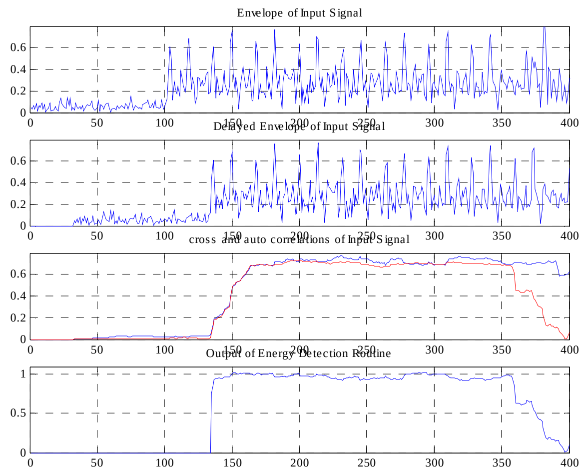


$$\hat{r}(0)=\sum_{k=n}^{15} d(n+k)d^*(n+k)$$

$$\hat{r}(16)=\sum_{k=n}^{15} d(n+k)d^*(n-16+k)$$

$$\hat{c}(16)=\frac{\hat{r}(16)}{\hat{r}(0)+\delta}$$

Packet Detection



OFDM Arithmetic Resourcing

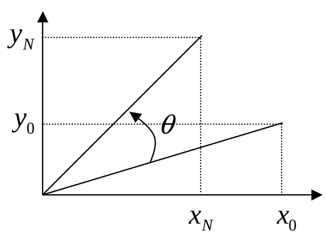
- Several OFDM functions require arithmetic resourcing for processes other than
  - Multi-rate filters
  - (I)FFT
- Synchronization/Equalization in demod
  - Division for computing decision statistic in delay-and-correlate packet detection routine
  - Constructing database from channel probe used in frequency domain equalizer
- Peak-to-average power ratio (PAR) control in mod
  - Complex vector magnitude
  - Rotation of PAR control time-series

CORDIC[1] Algorithm

- COoRdinate Digital Computer (CORDIC) algorithm
- Useful for computing
  - linear functions
  - circular/inverse circular functions
    - sin,cos, atan(y/x)
  - hyperbolic/inverse hyperbolic functions
    - sqrt(a)
    - ln(a)

[1] J. E. Volder, “The CORDIC Trigonometric Computing Technique”, *IRE Trans. On Electronic –Computers,* Vol. EC-8, 1959, pp. 330-334.

CORDIC: Rotate Mode



$$i = 0$$

$$z_i = \theta$$

$a_i = -1$ if $z_i < 0$ otherwise $a_i = +1$

$$x_{i+1} = x_i - a_i y_i 2^{-i}$$

$$y_{i+1} = y_i + a_i x_i 2^{-i}$$

$$z_{i+1} = z_i - a_i \tan^{-1}\left(2^{-i}\right)$$

$$i = i + 1$$

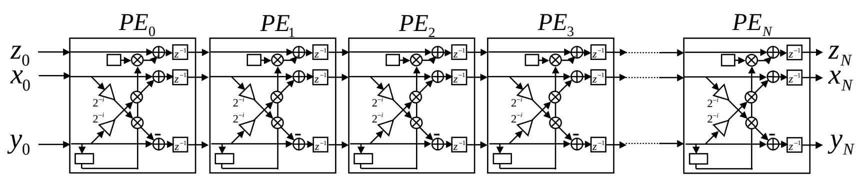
After the final iteration

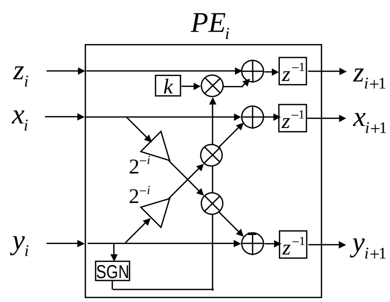
$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = K \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix}$$

$$z_N = 0$$

$$K = \prod_{i=0}^{N} \sqrt{1 + 2^{-2i}}$$

Unrolled CORDIC





Parallel CORDIC implementation (vector mode)
One result/clock
Does not include output scaling

Linear CORDIC

Comment	Initial Condition $[x_0, y_0, z_0]$	Result Vector $[x_N, y_N, z_N]$
Drive z to 0	$[x_0, y_0, z_0]$	$[x_0, y_0 + x_0 z_0, 0]$
Drive y to 0	$[x_0, y_0, z_0]$	$[x_0, 0, z_0 + y_0 / x_0]$

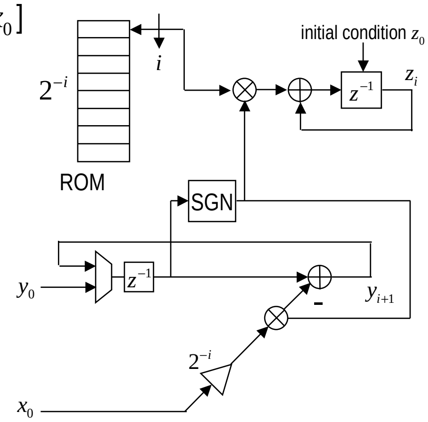
- The linear CORDIC mode can be used to construct a divider

Linear CORDIC

With initial conditions $[x_0, y_0, z_0]$
driving y to 0 gives $[x_0, 0, z_0 + y_0 / x_0]$

Linear Iterations (divide)

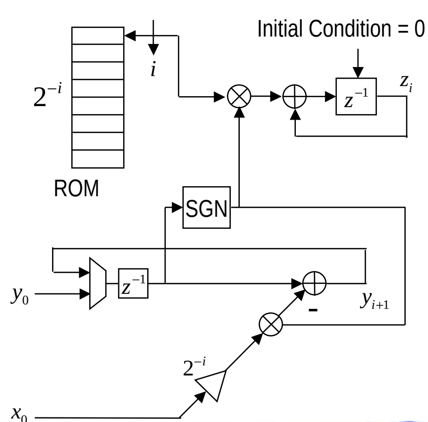
$$i = 0$$
$$d_i = \mathrm{sgn}(y)$$
$$x_{i+1} = x_i$$
$$y_{i+1} = y_i - d_i x_i 2^{-i}$$
$$z_{i+1} = z_i + d_i 2^{-i}$$
$$i = i + 1$$



CORDIC Divider

- The linear CORDIC mode can be used to construct a divider by setting z_0=0

With initial conditions $[x_0, y_0, 0]$
driving y to 0 gives $[x_0, 0, y_0 / x_0]$



Linear CORDIC

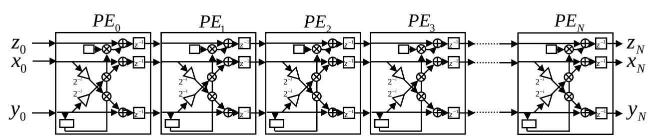
- Matlab script to implement a CORDIC divider

```
function [x,y,z] = cordic_div3(x,y,niters);
% Creation Date 03/14/02
% Author: Dr Chris Dick
% Synopsis: This routine employs the CORDIC algorithm to compute y/x

% Linear CORDIC configuration

z = 0;
for ii = 1:niters
   dn = sgn(y);    % sgn() returns the sign of the argument
   y = y - dn * x*2^-(ii-1);
   z = z + dn * 2^-(ii-1);
end
```

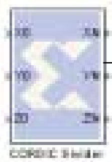

Programmatic Generates (1)



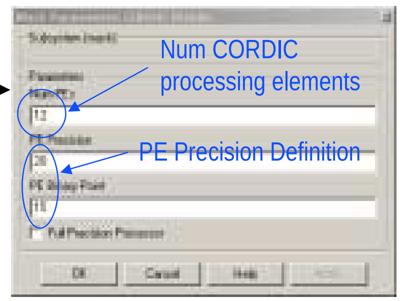
- For regular structures like the CORDIC algorithm the conditional generate functionality of Simulink is very useful
- Permits construction of a graph (in Simulink) based on
  - GUI
  - workspace variables

Programmatic Generates (2)

- The *add_*block(), add_*line*(), *delete_block*() and *delete_line*() Matlab function calls are employed to contruct the graph

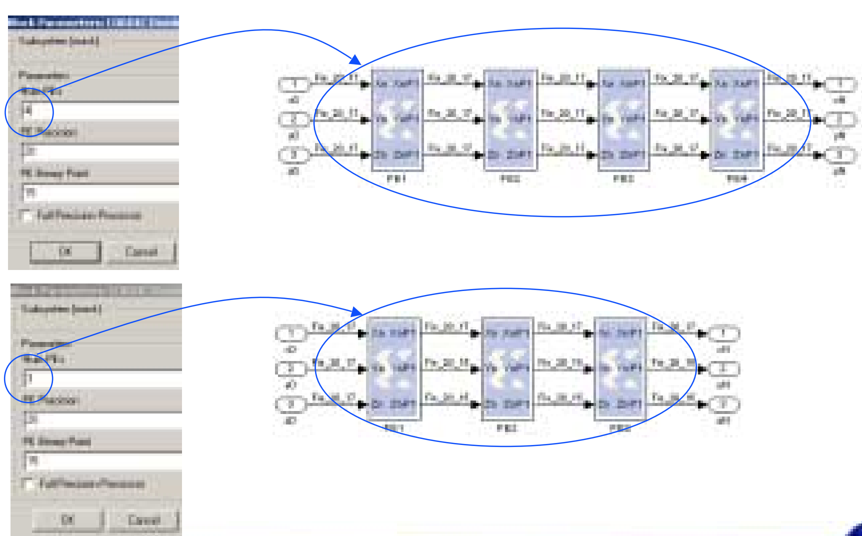


Customization Panel

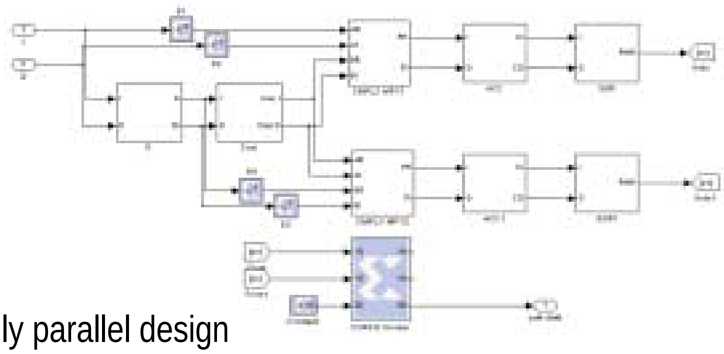


- This is not a component from the System Generator BlockSet
- It is a component from a customized libray that any user can construct

Programmatic Generates (3)



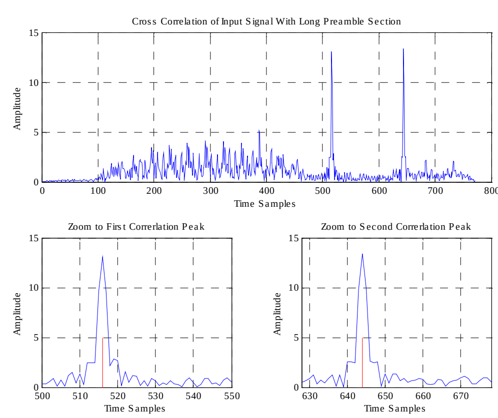
Implementation



- Fully parallel design
  - 473 slices
  - 12 embedded mpy
  - fclk ~200 MHz
  - Potential for hardware folding to reduce mpy count

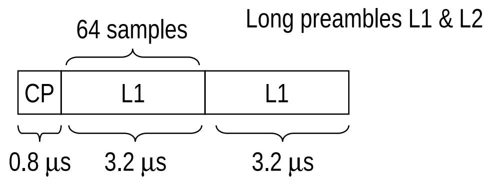
Long Preamble Correlator

- Full precision correlator
  - Requires large arithmetic resourcing (MACs)



Long Preamble Correlator

- Use 802.11a as an example
- Preamble structure



- 64 samples in 3.2 µs corresponds to a channel data rate of 20 MHz

Long Preamble Correlator

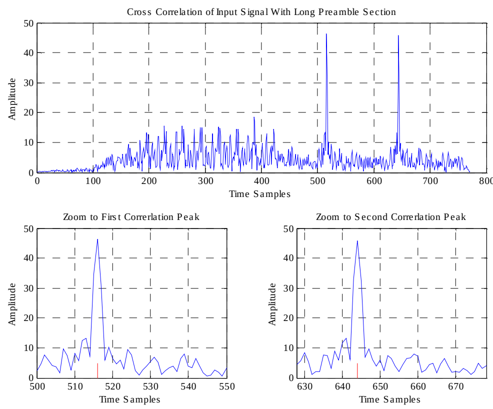
- Correlator workload
- Remember the data and reference waveform are both complex

$$\text{OPs}= \underbrace{20e6}_{\text{CHANNEL DATA RATE}} \times \underbrace{64}_{\text{CORRELATION SEQUENCE LENGTH}} \times \underbrace{4}_{\text{COMPLEX CORRELATOR}} = 5.12e9 \text{ GOPs}$$

- One way to place this number in context
  - 200% the real-time cycles of a 500 MHz 4-MAC DSP processor

Long Preamble Correlator

- Clipped correlator
  – Uses sign of data and sign of template samples

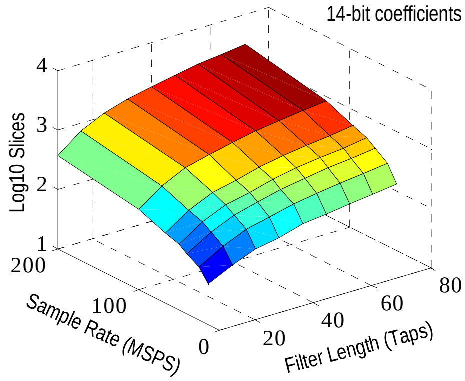


Clipped Correlator

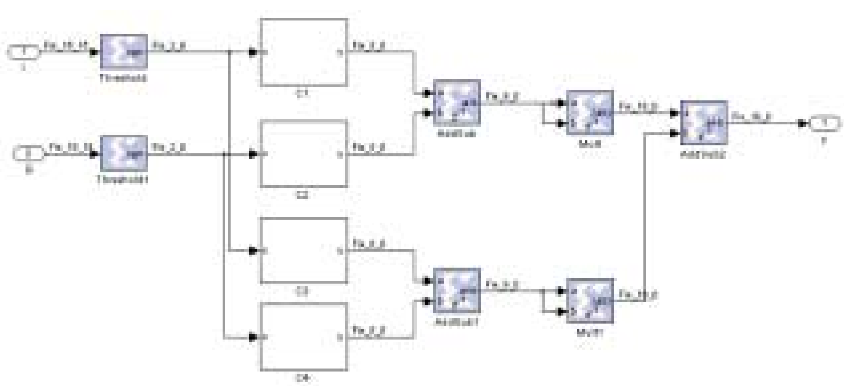
- Clipped correlator uses 1-bit MACs
  – No mpy's required
- FPGA implications
  – Assume FPGA fclk=100 MHz
  – 64-point correlator decomposed as 13 5-point correlators
  – Use 4 of these to implementation the complex correlator
- Classic example of datapath "right sizing"

Datapath Right-sizing

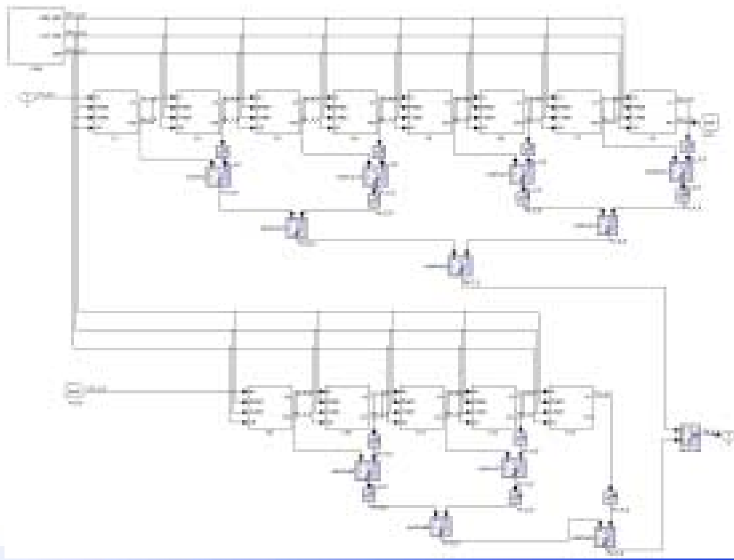
- DA FIR is malleable and provisions for
  - Si/sample rate matching
  - Datapath right-sizing
  - Actually unlike MPY based structures in this respect/capability
- Use optimum precisions at each node in the computation graph
- 'Right-size' the datapath
- design surface for a FIR filter: Area vs Sample Rate vs Length



Implementation



Correlator Arm (1 of 4)



Implementation

- 1100 slices
- 2 embedded multipliers
- 192 MHz[1]

$$\text{Correlator Ops} = \underbrace{4}_{\text{complex}} \times \underbrace{64}_{\text{integration interval}} \times \underbrace{192e^6}_{\text{FPGA Clock}} = 49\ \text{GOPs/s}$$

$$\frac{49\ \text{GOPs/s}}{1100\text{slices}} = 45\ \text{MMACs/slice}$$

1. Device speed data version: ADVANCED 1.69 2002-11-07

Channel Estimation/Equalization

- To achieve high data rates in OFDM coherent detection is employed
- Detector relies on phase and amplitude information on each flat fading subcarrier
- OFDM receiver must compensate for the effects of the propagation channel before making symbol decisions for each sub-carrier

Channel Estimation/Equalization

- Compensating for the radio channel is partitioned into two phases
  - Channel estimation (acquisition)
  - Channel equalization & tracking
- Channel estimation
  - Frequency domain approaches
  - Time domain approaches

Channel Estimation Using Training Data

- Long training symbols in WLAN applications can be used to estimate the frequency response for all the subcarriers
- Averaging multiple training symbols may be used to improve the estimate
- DFT at the Rx is a linear operation so that the average can be computed prior to the transform

Channel Estimation Using Training Data

- In single carrier systems the receiver is presented with a data stream that is the result of a linear convolution (tx data and channel)
- Of course it is not quite that simple
  - Multipath
  - AWGN
  - Other interference (e.g. narrowband sources)

Channel Estimation Using Training Data

- Cyclic extension of the FFT symbol makes the channel filtering process *look* like a cyclic convolution (as opposed to a linear convolution) … at least as far as the tx data is concerned
- Recall the convolution property of the DFT
  – Time-frequency duality

$$y(n) = h(n) \otimes x(n) \quad \longleftrightarrow \quad Y(z) = H(z)X(z)$$

where $\otimes$ denotes cyclic convolution

Channel Estimation Using Training Data

- The phase and amplitude distortion introduced by the channel can now be corrected using a single complex multiplication per subcarrier
  – 1 tap equalizer
- WLAN applications assume that the channel is quasistationary
  – Channel does not change during the packet transmission

Channel Estimation

- The received signal after the FFT is

  $Y(k) = C(k)X(k) + Z(k)$

  where k is the subcarrier index, C is the channel, X is the training data and Z is noise

  Channel estimate $\hat{C}(k)$ is $\hat{C}(k) = \frac{Y(k)}{X(k)}$
- The division is undesirable from an implementation perspective
- Use CORDIC algorithm for FPGA implementation

Channel Equalization

- Channel equalization in OFDM systems is performed in the frequency domain
- Realized as a complex multiplication for each spectral component recovered by the FFT in the demodulator
- For each packet compute the compensation term

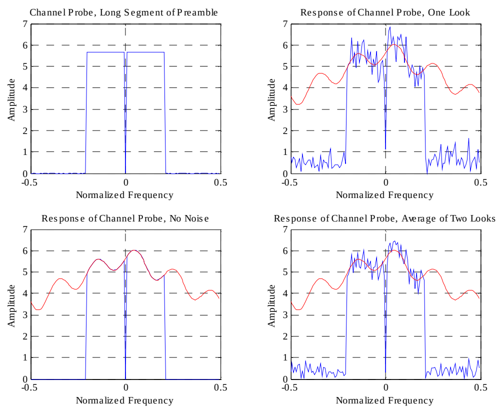
$$\frac{1}{\hat{C}(k)} = \frac{X(k)}{Y(k)}$$

and store for use by the equalizer as it reverses the channel impact for the OFDM data payload

- Receiver has access to the long preamble
- *X*(*k*) is computed offline and stored in the receiver for use by the channel estimator

Channel Estimation

- Wideband channel probe
  - Long preamble in WLAN applications



Implementation

- Fully parallel implementation of estimator and equalizer
- 776 slices
- 10 multipliers
- 4 block RAMs