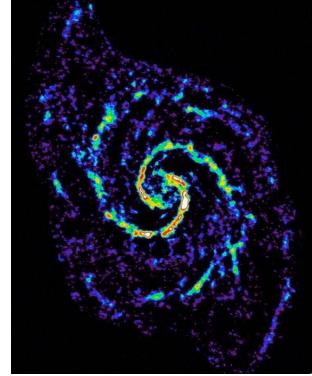


FPGA-based Digital Signal Processing



David Hawkins
dwh@ovro.caltech.edu



1

Overview

- Introduction
- Application Example
- DSP Review
- FPGA Architectural Features
- DSP Components
- Example Designs

FPGAs; What are they?

- Abbreviations;
 - FPGA = **Field-Programmable** Gate Array
 - CPLD = **Complex** Programmable Logic Device

- Vendors;
 - Major; Altera, Xilinx, Lattice
 - Minor; Actel (Microsemi), Achronix, SiliconBlue, Tabula

3

FPGAs; What are they?

- '**Field-Programmable**'
 - Fully-customizable and reprogrammable by the user

- '**Complex**' - Fairly steep learning curve;
 - Hardware description languages (HDL); Verilog, VHDL
 - Scripting; Tcl
 - Vendor-specific;
 - Tools
 - IP configuration
 - Constraints
 - Board-level integration;
 - Power/Thermal design
 - Dynamic reconfiguration
 - System-level communications

4

FPGAs; What's inside one?

- Logic elements
 - 4 or 6 input LUT (eg., 531k S4GX530)
- DSP blocks
 - multipliers, adders (eg., 1024 18bx18b)
- Embedded memory blocks
 - Dual-ported, FIFOs (eg., ~3MB)
- External memory controllers
 - DDR1/2/3, QDRII/II+, RLDRAMII/III

5

FPGAs; What's inside one?

- Transceivers
 - 8.5Gbps (Stratix IV) to 12.5Gbps (Stratix V)
 - Hard-IP serializer/deserializer (SERDES)
 - PCIe, 10/40/100G Ethernet, Hypertransport, Interlaken, etc.
- LVDS
 - 1.25Gbps with hard-IP SERDES
- Parallel I/O
 - Hundreds of user I/O pins
 - Multiple banks can use different voltage standards

6

FPGAs; What's inside one?

■ Embedded Processors

■ Softcore

- Altera NIOS II
- Xilinx Microblaze
- ARM Cortex-M1

■ Hardcore

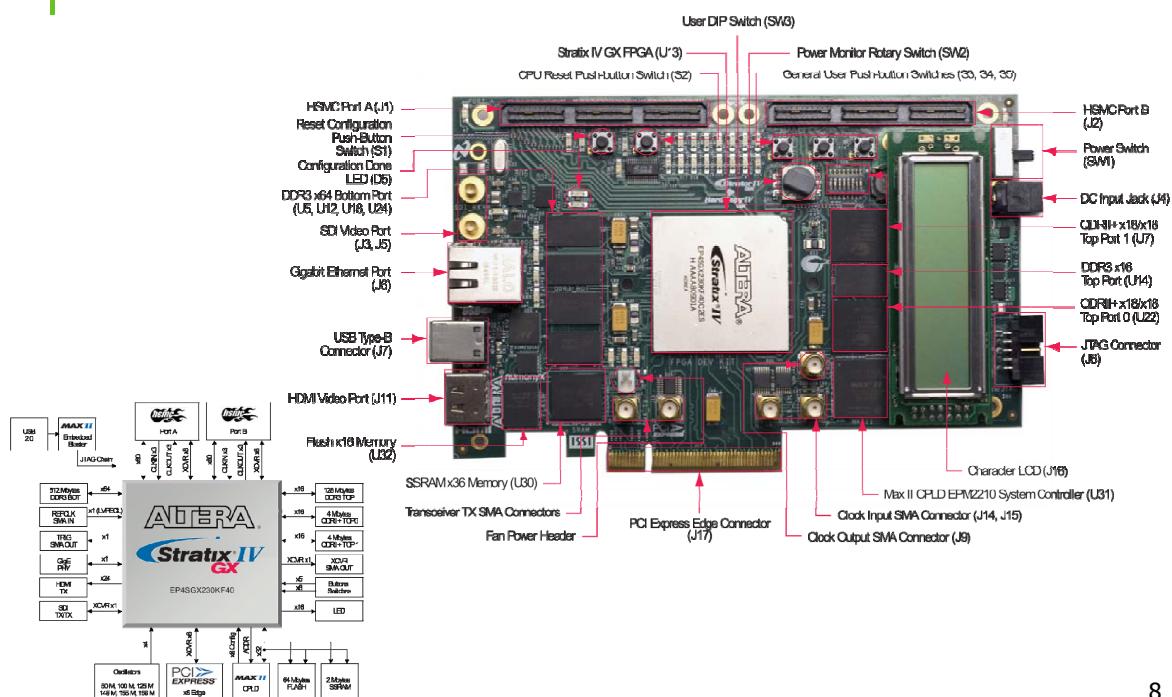
- Xilinx PowerPC
- Newer ARM cores coming

■ Embedded Software

- Linux
- uCOS/II RTOS

7

FPGAs; What do they look like?



8

FPGAs; How do I get one?

- Low-cost (under \$100) USB-Sticks

- Terasic DE0-nano (Altera Cyclone IV)
 - 22k LEs, 66 DSP blocks (18x18-bit)
- Arrow BeMicro-SDK (Altera Cyclone IV)
 - 22k LEs, 66 DSP blocks (18x18-bit)
- Avnet Microboard (Altera Spartan-LX9)
 - 9k LCs, 16 DSP48A1 blocks (18x18-bit)

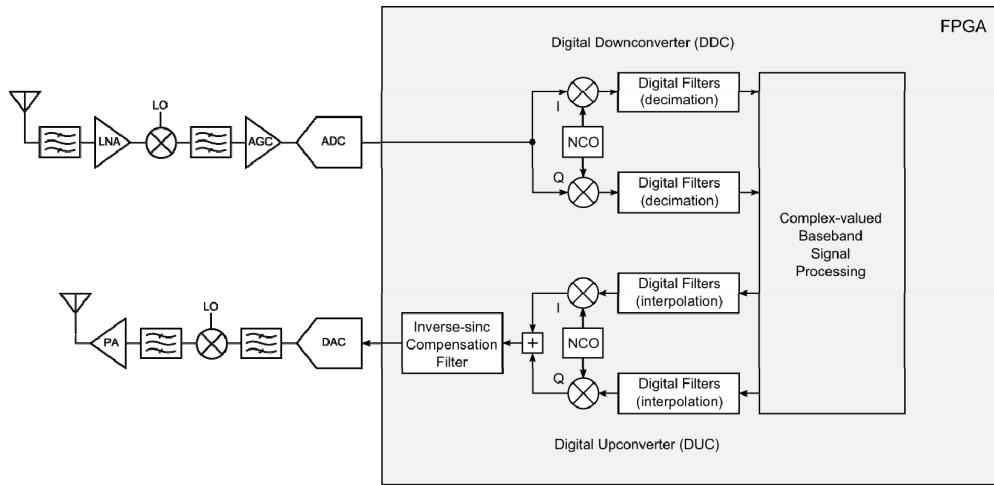


9

FPGAs; How do I get one?

- High-end (several \$1000) kits
 - DSP kits from Altera and Xilinx
 - Altera kits from Terasic, eg. DE4
 - DSP companies (Altera or Xilinx boards)
- My development kits;
 - Stratix II DSP Development kit
 - Stratix IV GX Development kit
 - USB-Sticks
 - Custom board designs

An FPGA-based DSP system



11

Application Example

Why am I interested in FPGA-DSP?
What do I do with it?

12

Radio Astronomy

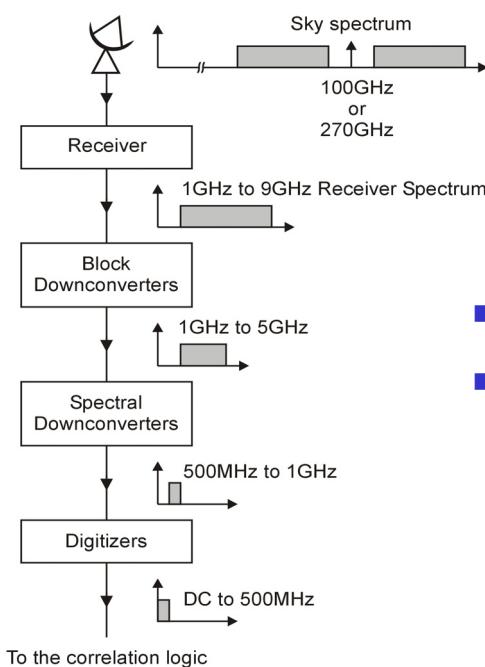


The CARMA + SZA radio interferometers



13

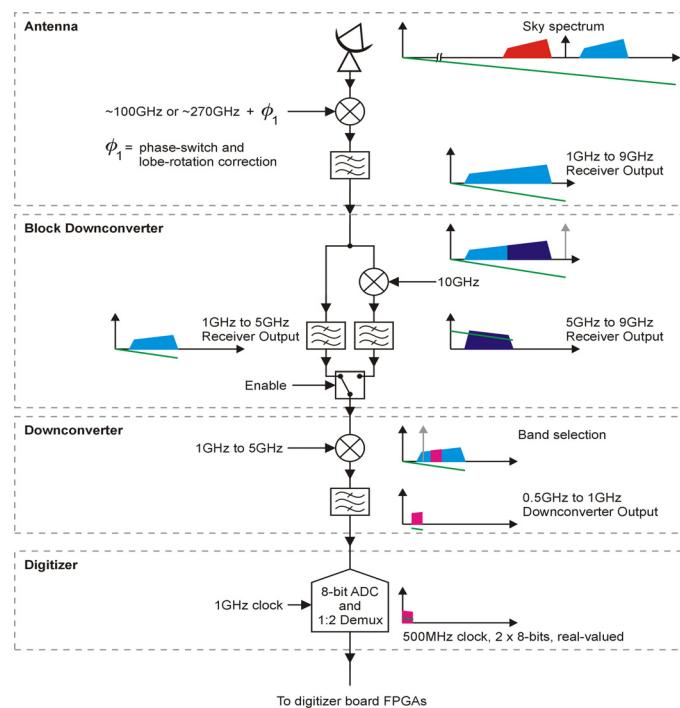
Antenna Signal Path



- 23-antennas
- 8-bands of up to 500MHz each

14

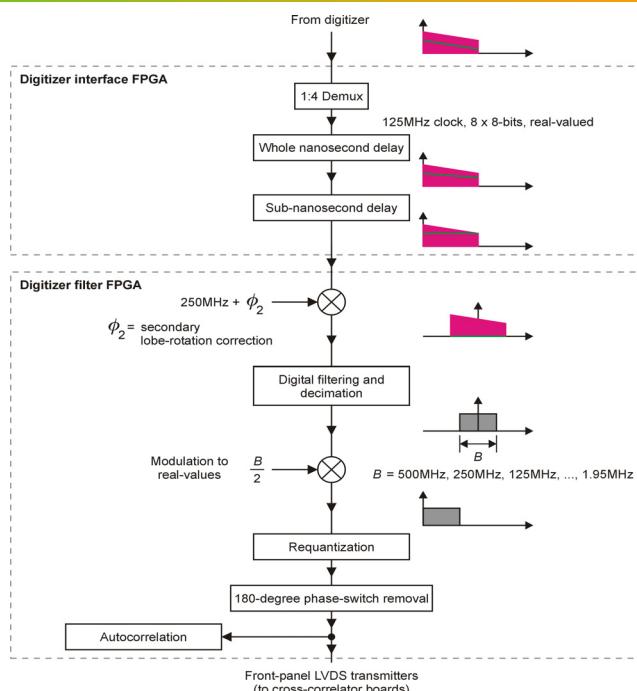
Antenna Signal Path



Using 20GHz ADCs,
we will convert to
an 'all digital' IF

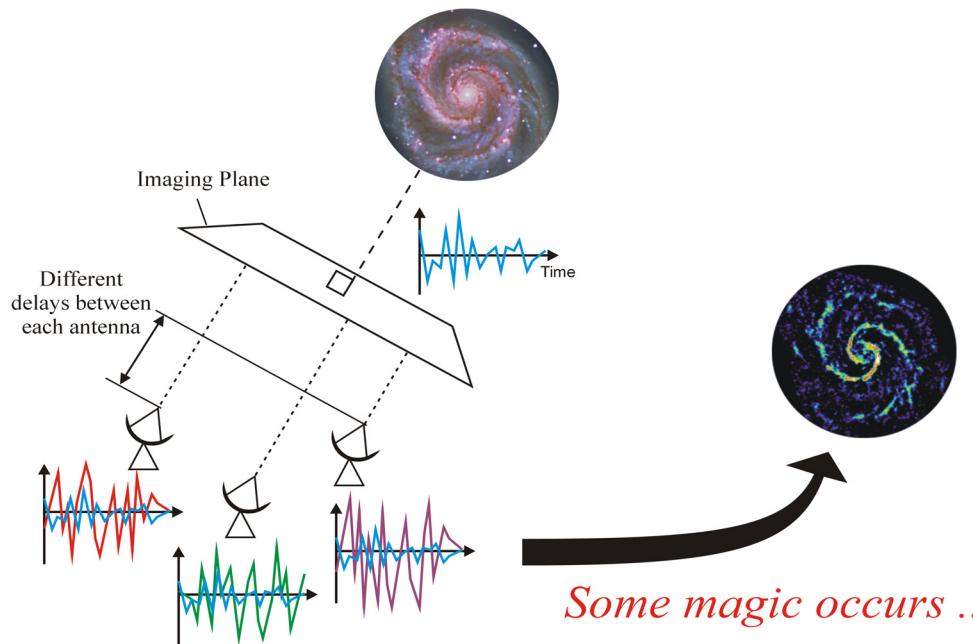
15

CARMA Antenna Signal Path



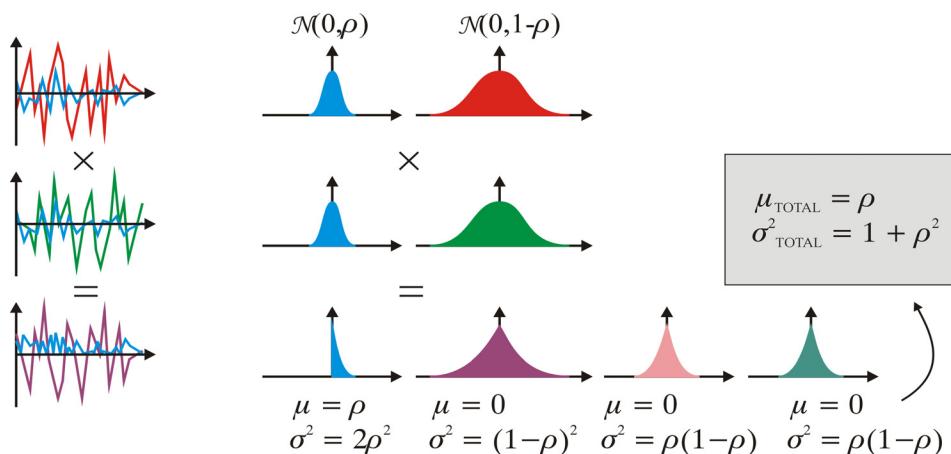
16

Power-spectra (correlation)

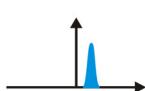


17

Cross-correlation



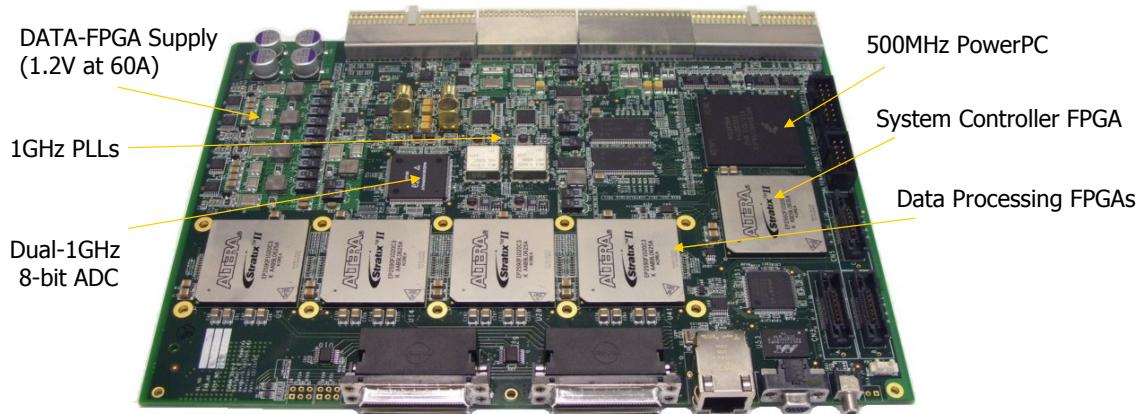
K-sample average:



$$\mathcal{N}\left(\rho, \frac{(1+\rho^2)}{K}\right)$$

18

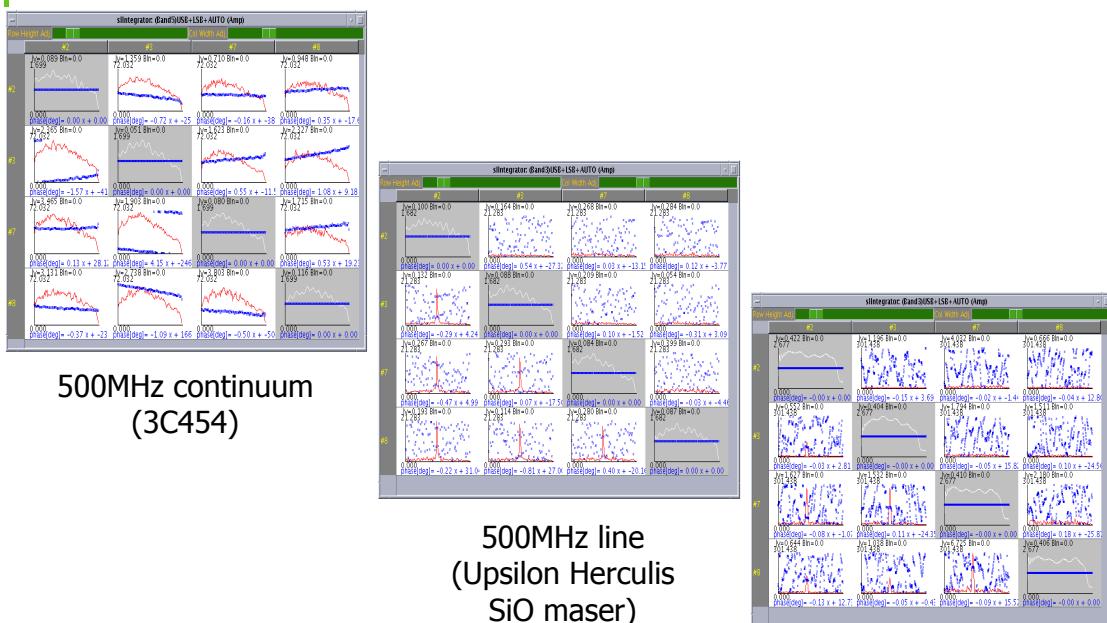
CARMA Digitizer/Correlator Board



- 2-antennas input
- 500MHz bandwidth each input
- FIR filtering
- Digital Downconversion
- Digital Delay
- Auto/Cross Correlation
- DATA-FPGAs average correlation data
- SYS-FPGA coordinates real-time DMA of averaged data to PowerPC memory
- PowerPC runs Linux
- Processes the data in 'sustained real-time', i.e., it just has to keep up!

19

Radio Source Spectra

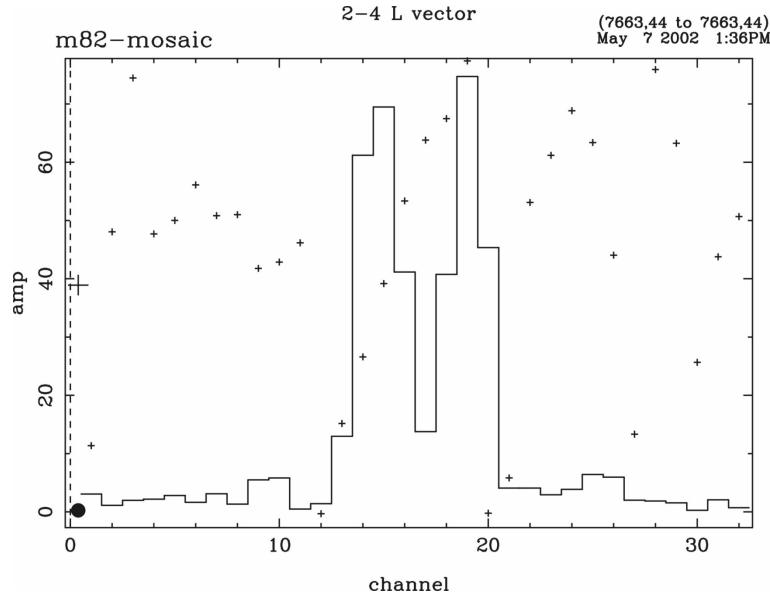


62MHz line

20

Radio Source Spectrum

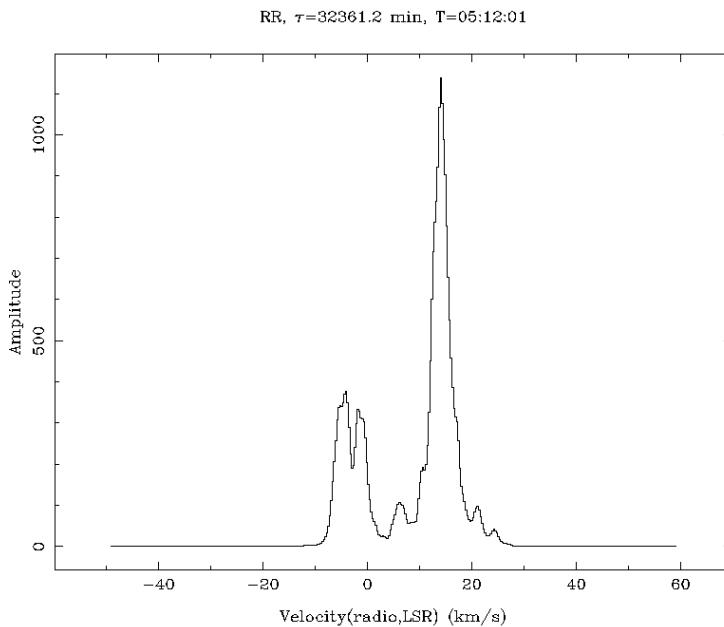
Galaxy M82 SiO Maser Spectrum (~30 channels)



21

Radio Source Spectrum

Orion SiO Maser Spectrum (~300 channels) Moore's Law in Action!



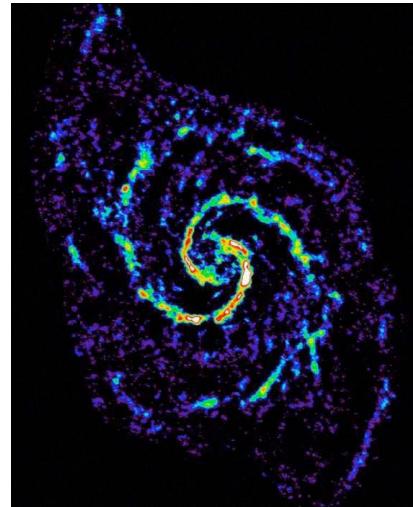
22

Optical and Radio Images

The Whirlpool Galaxy (M51)



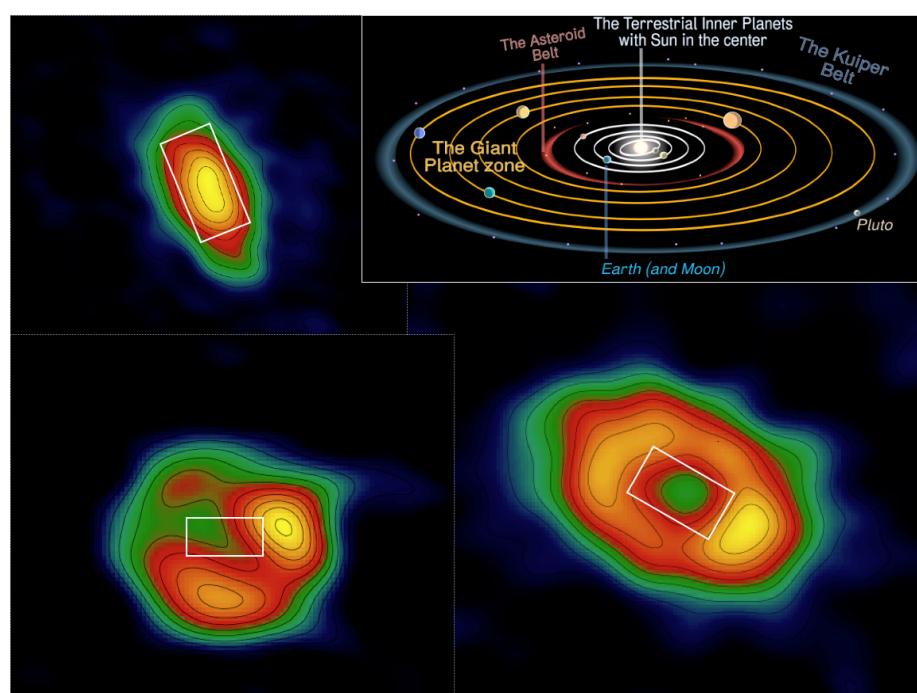
(a) Hubble Space Telescope Image



(b) CARMA Radio Image

23

Solar Systems – Just like ours?



24

The Correlator Systems



SZA Correlator
16 bands x 8-telescopes x fixed
500MHz bandwidth

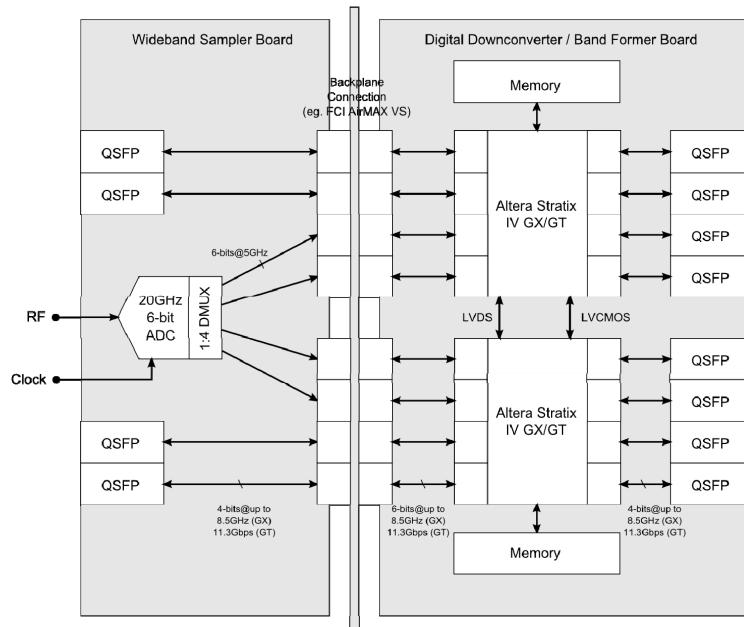


CARMA Correlator
8 bands x 15 or 30-receivers x adjustable LO x
500MHz down to 2MHz bandwidth 25

Data Volumes

- Sampled signal bandwidth;
 - 8GHz analog bandwidth; 16 x 500MHz bands
 - 8-bits at 1GS/s per 500MHz input
 - 23 antennas
 - 16 bands x 8Gbps per band x 23 antennas = **2.94Tbps**
- Processing bandwidth
 - 23 auto + 253 cross-correlations
 - 1024-lags per correlation (one MAC per lag)
 - 16 x 1GS/s x 1024 x 276 = **4522 TMAC/s**

Current project



27

Digital Signal Processing

What is it? How do you do it?
What is important for FPGAs?

28

DSP Review

- Numeric Formats
 - In an FPGA **you** control it, so **you** must understand it
- Complex Numbers
- Transforms
 - Fourier
 - Laplace
 - Z-Transform
- Discrete Signals
 - Sampling
 - Aliasing
 - Quantization

29

Numeric Formats

- Floating-point format
- Fixed-point format
 - Unsigned and signed
 - Integers and Fractional integers
 - Qm.n format
 - signed
 - m-bits whole
 - n-bits fraction
 - m+n+1 bits to represent

30

Fixed-point Addition/Multiplication

- 4-bit unsigned ($N = 4$)
 - 0000b to 1111b (0 to 15)
 - Largest sum; **(N+1)-bits required**
 - $15 + 15 = 30$ (11110b)
 - Largest product; **2N-bits required**
 - $15 \times 15 = 225$ (1110_0001b)

31

Fixed-point Addition/Multiplication

- 4-bit signed ($N = 4$)
 - The MSB represents 0 or -8
 - LSBs [2:0] represent 0 to +7
 - 1000b to 0111b (-8 to 7) **(asymmetric)**
 - Most positive sum; **(N+1)-bits required**
 - $7 + 7 = 14$ (0_1110b)
 - Most negative sum; **(N+1)-bits required**
 - $-8 - 8 = -16$ (1_0000b)
 - Largest product; **2N-bits required**
 - $-8 \times -8 = 64$ (0100_0000b)

32

Fixed-point Addition/Multiplication

■ Problems with signed numbers

- Negation of the largest negative value requires an extra bit to store the result
- The product of the most negative value requires one extra bit relative to all other products
- Use **signed symmetric** instead
 - eliminate the most negative number
 - Eg., a 4-bit integer with range **-8 to +7**, has any -8 values changed to -7, and the range becomes **-7 to +7**

33

3-bit Unsigned Multiplication

	0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0 000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000
1 001	0 000000	1 000001	2 000010	3 000011	4 000100	5 000101	6 000110	7 000111
2 010	0 000000	2 000010	4 000100	6 000110	8 001000	10 001010	12 001100	14 001110
3 011	0 000000	3 000011	6 000110	9 001001	12 001100	15 001111	18 010010	21 010101
4 100	0 000000	4 000100	8 001000	12 001100	16 010000	20 010100	24 011000	28 011100
5 101	0 000000	5 000101	10 001010	15 001111	20 010100	25 011001	30 011110	35 100011
6 110	0 000000	6 000110	12 001100	18 010010	24 011000	30 011110	36 100100	42 101010
7 111	0 000000	7 000111	14 001110	21 010101	28 011100	35 100011	42 101010	49 110001

3-bit x 3-bit unsigned multiplication gives a 6-bit result

34

3-bit Signed Multiplication

The only positive product with the MSB set

	-4 100	-3 101	-2 110	-1 111	0 000	1 001	2 010	3 011
-4 100	16 010000	12 001100	8 001000	4 000100	0 000000	-4 111100	-8 111000	-12 110100
-3 101	12 001100	9 001001	6 000110	3 000011	0 000000	-3 111101	-6 111010	-9 110111
-2 110	8 001000	6 000110	4 000100	2 000010	0 000000	-2 111110	-4 111100	-6 111010
-1 111	4 000100	3 000011	2 000010	1 000001	0 000000	-1 111111	-2 111110	-3 111101
0 000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000	0 000000
1 001	-4 111100	-3 111101	-2 111110	-1 111111	0 000000	1 000001	2 000010	3 000011
2 010	-8 111000	-6 111010	-4 111100	-2 111110	0 000000	2 000010	4 000100	6 000110
3 011	-12 110100	-9 110111	-6 111010	-3 111101	0 000000	3 000011	6 000110	9 001001

No 3-bit code for $-(-4) = 4$ Eliminate the use of -4, and the product MSB is available for a multiply-add

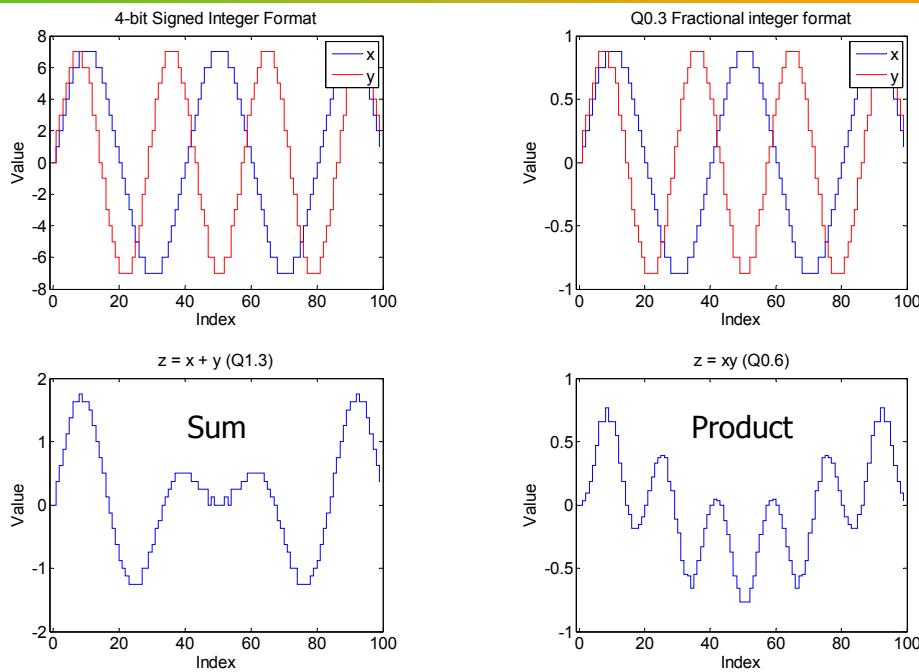
35

Fractional Integer Format

- (N+1)-bit signed value used as Q0.N
 - 8-bit signed integer; (-128) -127 to 127
 - Divide by $2^7=128$ to get fractional format Q0.7
 - 8-bits fractional integer;
 - $(-128/128)$ -127/128 to 127/128
 - (-1.0) -0.992 to +0.992
 - i.e., approximately -1.0 to 1.0
- Each sum or difference requires 1 more MSB
 - Q1.N format
- Products are in (Q1.2N) Q0.2N format
 - Convert to Q0.N by retaining sign and first N fraction bits

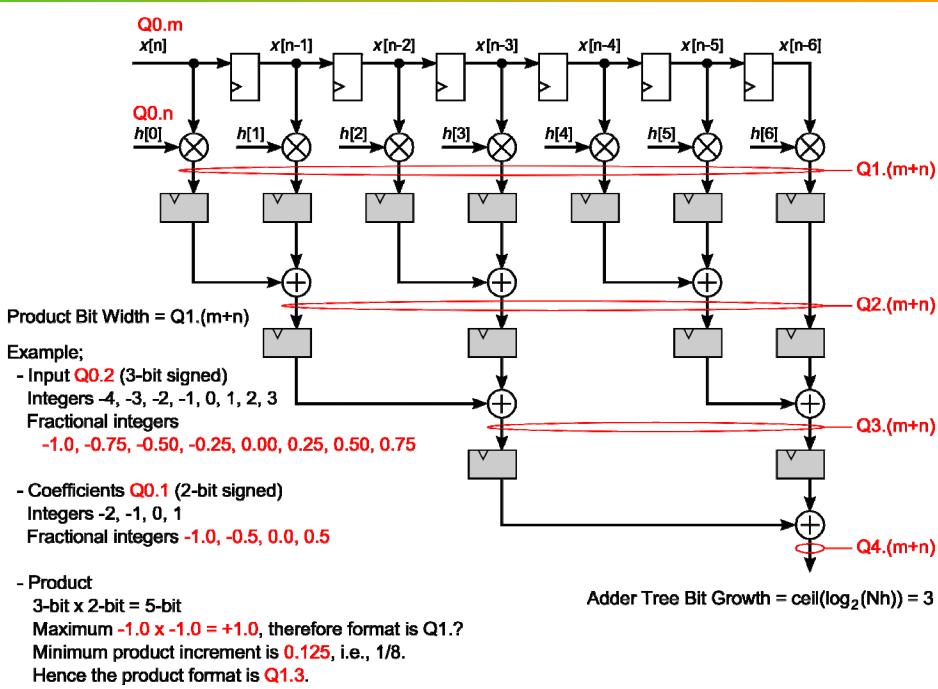
36

4-bit Fractional Integer Format



37

Fractional Integers and Bit-Growth



38

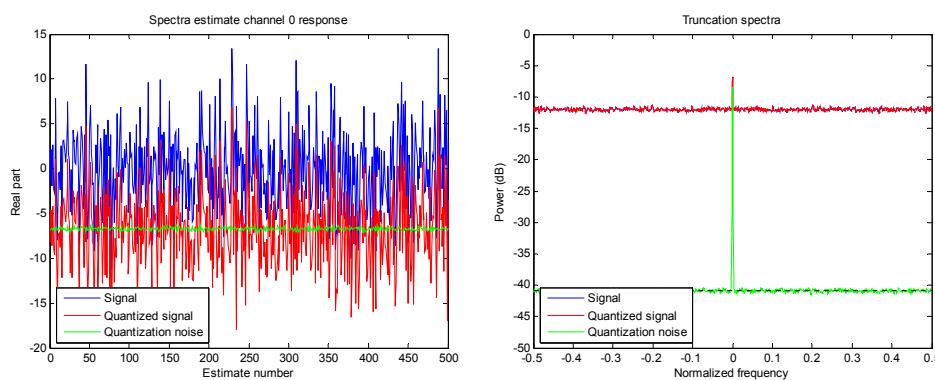
Numeric Terminology

- DSP operations increase the number of bits required to represent results
- Conversion of results to fewer bits;
 - Truncation
 - Throw away the LSBs
 - Rounding
 - Round-to-nearest (add 0.5 and then truncate)
 - Convergent (round 0.5 to even)
 - Overflow
 - Value exceeds the maximum +/- and wraps around
 - Saturation
 - Values that exceed the maximum +/- representation are 'saturated' to the maximum value

39

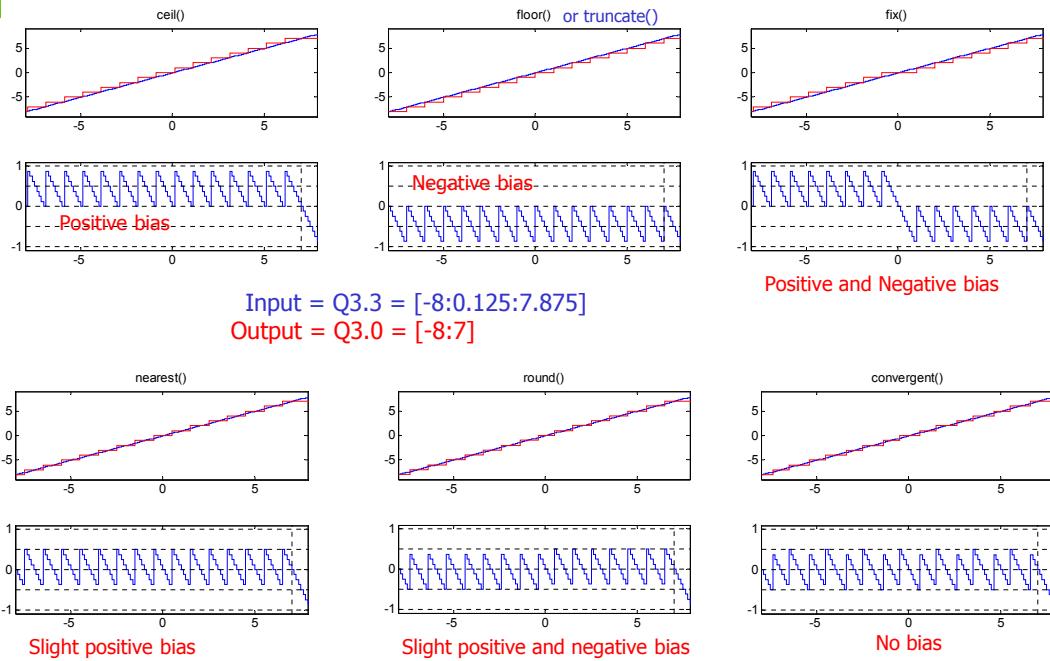
Truncation

- Truncation = throw away the LSBs
 - A quantization operation, where the quantization noise is distributed between -1 and 0 (of the new binary code)
 - This quantization noise has a DC bias
 - For example, 6-bit quantization of a wideband noise signal



40

Common Rounding Methods



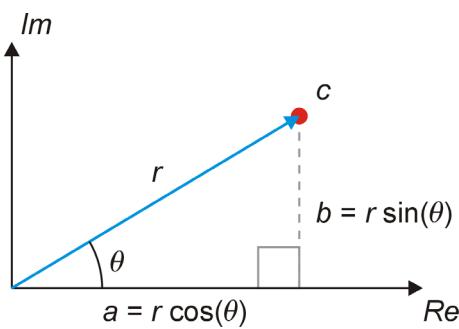
41

Complex Numbers

- Used to represent signals that have two components:
 - Dealing with complex numbers is not **complex** (hard), there is nothing **imaginary** (non-existent) about either signal component, both components are **real** (exist) and need to be represented.
- Complex-valued signals are manipulated in FPGAs as two separate data paths (buses)

42

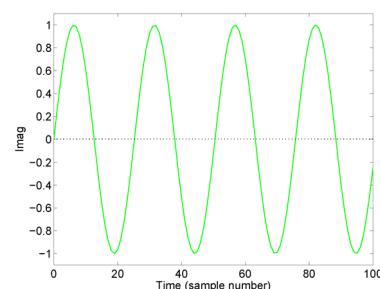
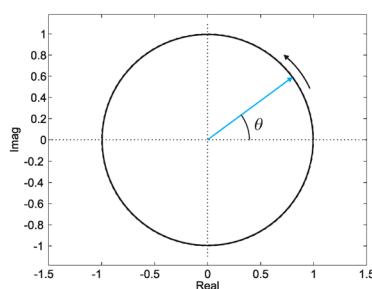
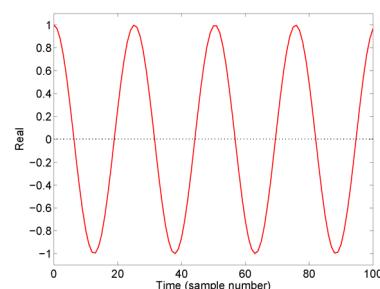
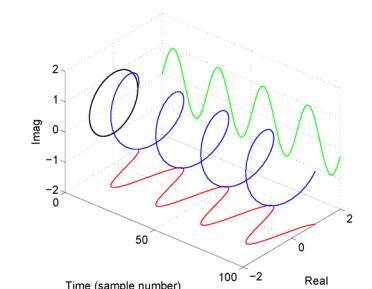
The Complex Plane



- Cartesian format;
 - $c = a + jb$
- Polar format;
 - $c = r\{\cos(\theta) + j\sin(\theta)\} = r\exp(j\theta)$

43

Complex Exponentials



Complex exponential = a convenient way to manipulate a cosine and a sine

44

Complex Exponentials

- Complex-valued samples are treated together as they are related by **time**.
- In Fourier transforms, the real- and imaginary components relate to symmetry
 - The even-symmetric cosine functions determine the real-part the Fourier transform
 - Eg., a FIR filter with an even symmetric tap response will have a real-valued Fourier transform
 - The odd-symmetric sine functions determine the imaginary-part of the Fourier transform
 - Eg., a Hilbert transform FIR filter with odd-symmetric tap response has an imaginary-valued Fourier transform

45

Transforms

- Fourier Transform
 - Decomposes a signal into cosines and sines (complex exponentials) of different frequencies
- Laplace Transform
 - Adds exponentials to the analysis
 - S-plane shows locations of;
 - Poles – increased response
 - Zeros – decreased response
 - Used in analog filter design and control theory
- Z-Transform
 - Discrete analog of the Laplace Transform
 - Z-plane shows poles and zeros of digital filters

46

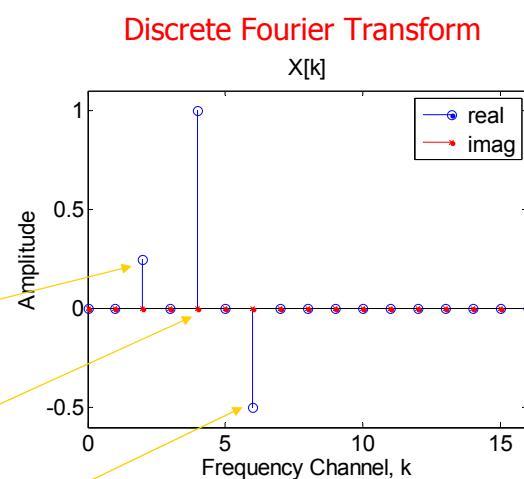
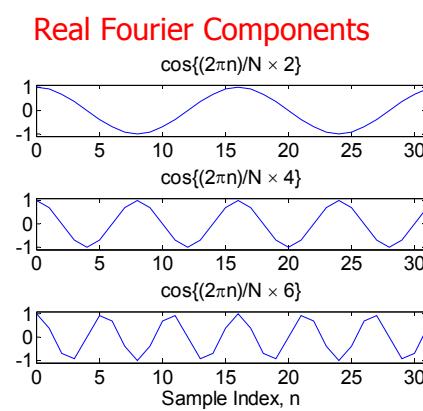
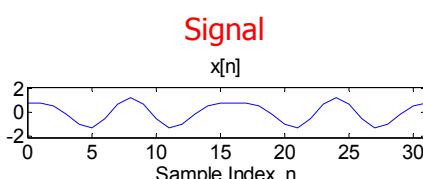
Discrete Fourier Transform

$$X[k] = \sum_{n=0}^{N-1} x[n] \exp\left(-j \frac{2\pi n}{N} k\right)$$

- Represents signal samples $x[n]$ in terms of cosine and sinusoid functions
- Provides an alternative view of a signal (and its components)
- Efficient computation via the Fast Fourier Transform

47

Discrete Fourier Transform



The DFT was normalized by $N/2$ (the expected peak of a cosine)

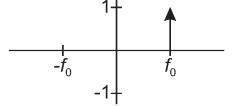
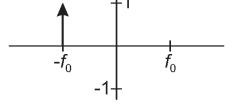
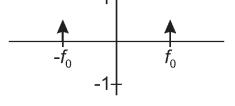
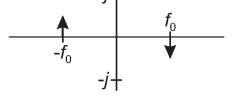
48

Fourier Theorems

Theorem	Time \Leftarrow Frequency
Time shift	$x(t - t_0) \Leftarrow X(f) \exp(-j2\pi f t_0)$
Frequency shift	$x(t) \exp(j2\pi f_0 t) \Leftarrow X(f - f_0)$
Time-domain convolution (Frequency-domain multiplication)	$x(t) * y(t) \Leftarrow X(f)Y(f)$
Time-domain multiplication (Frequency-domain convolution)	$x(t)y(t) \Leftarrow X(f) * Y(f)$
Time-domain correlation	$x(t) \star y(t) \Leftarrow X(f)Y^*(f)$

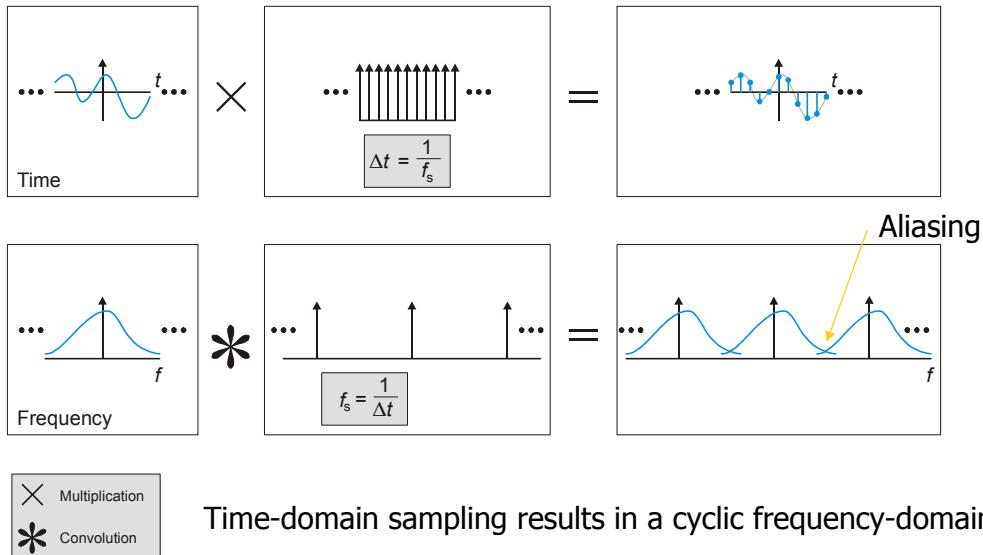
49

Fourier Pairs

Time \Leftarrow Frequency	Spectra
$\exp(+j2\pi f_0 t) \Leftarrow \delta(f - f_0)$	
$\exp(-j2\pi f_0 t) \Leftarrow \delta(f + f_0)$	
$\cos(2\pi f_0 t) \Leftarrow \frac{1}{2} \{\delta(f + f_0) + \delta(f - f_0)\}$	
$\sin(2\pi f_0 t) \Leftarrow \frac{j}{2} \{\delta(f + f_0) - \delta(f - f_0)\}$	

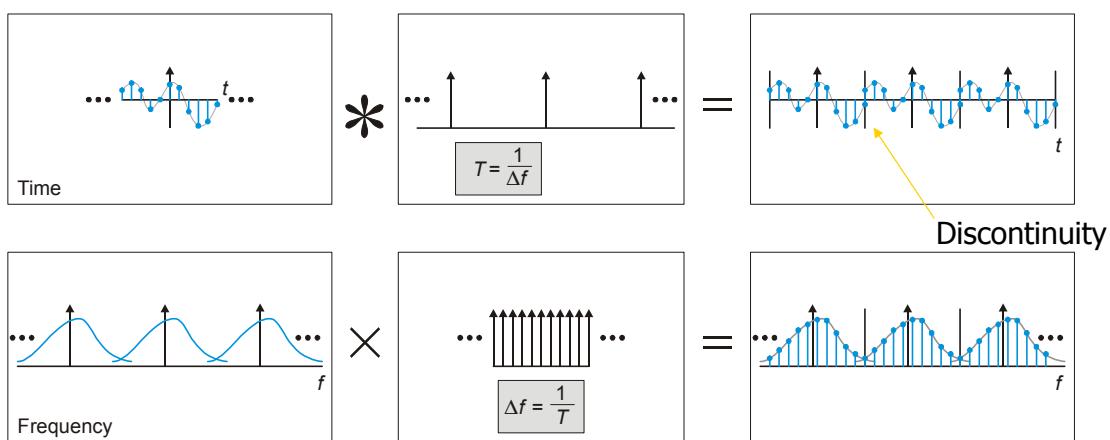
50

Time-Domain Sampling



51

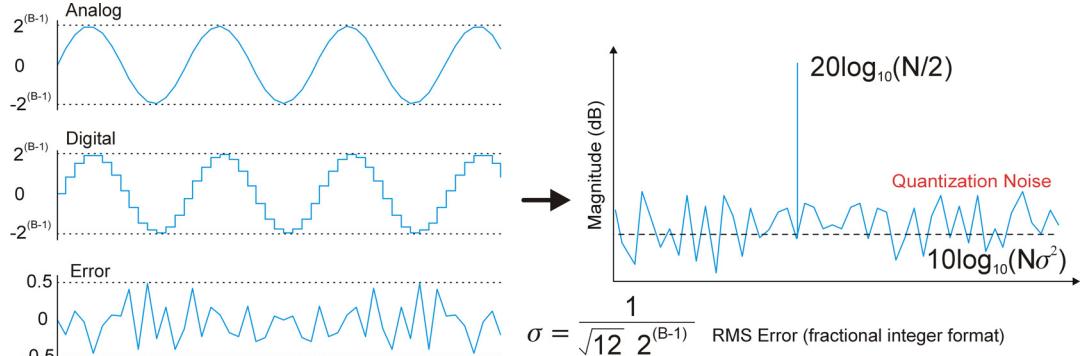
Frequency-Domain Sampling



Frequency-domain sampling results in a cyclic time-domain

52

Quantization Noise



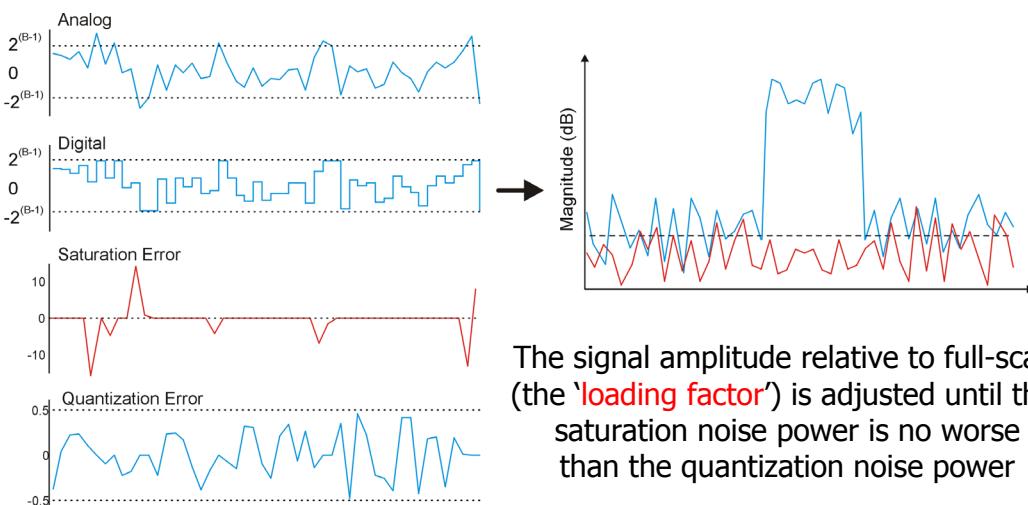
$$\frac{\text{SIGNAL}}{\text{NOISE}} = \left(\frac{N}{2}\right)^2 \frac{12}{N} 2^{(2B-2)} = 2^{2B} \frac{3}{2} \frac{N}{2}$$

$$\text{SNR (dB)} = 6.02B + 1.76 + 10\log_{10}(N/2)$$

Quantization noise also occurs when reducing the bit-width of digital signals

53

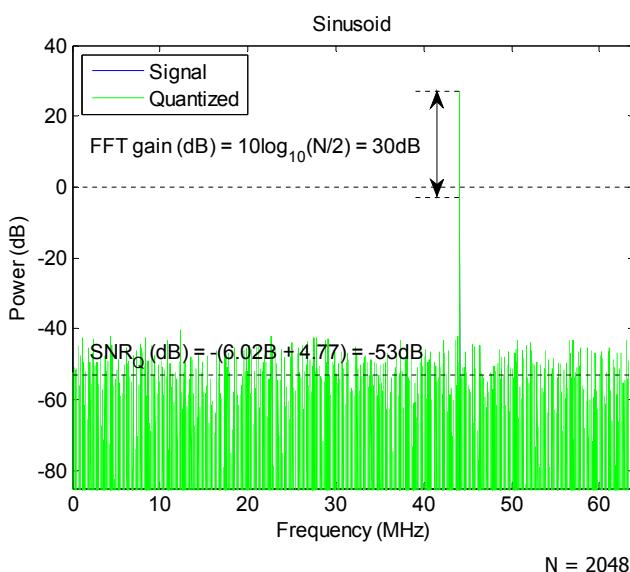
Saturation Noise



The signal amplitude relative to full-scale (the 'loading factor') is adjusted until the saturation noise power is no worse than the quantization noise power

54

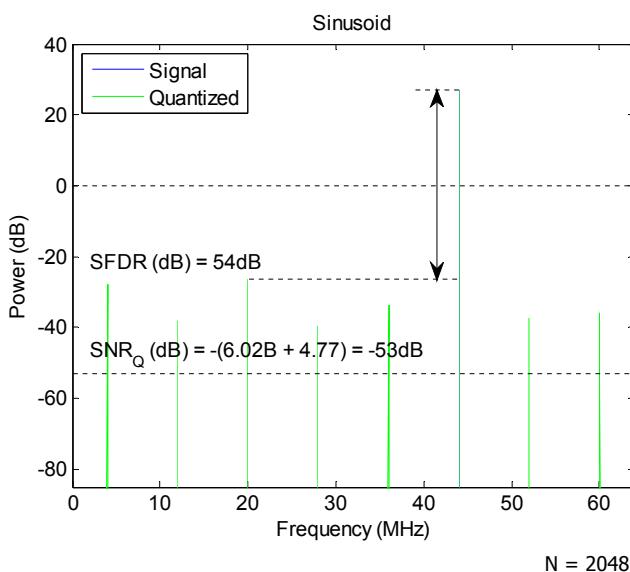
Quantization Noise; Sinusoids



- 8-bit data
- Full-scale (-3dB RMS) sinusoid
- The RMS (averaged) noise level is at -80dB relative to the peak
- **WARNING:** the tone frequency affects the response!
 - A tone frequency that is an odd frequency channel results in more uniform quantization noise at it exercises the most quantization codes.

55

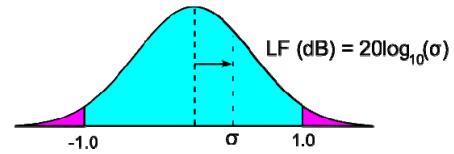
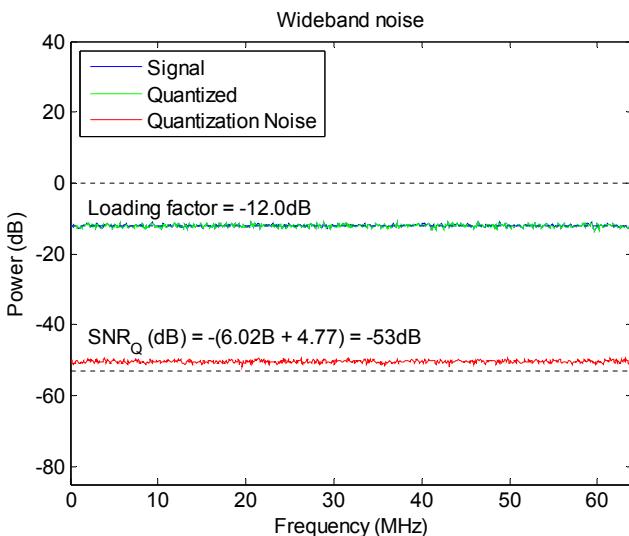
Quantization Noise; Sinusoids



- 8-bit data
- The tone is located at an even frequency channel. The quantization noise is concentrated at harmonics.
- **Numerically Controlled Oscillators** (NCOs) will show this issue.
- The **Spurious Free Dynamic Range** (SFDR) of the sinusoid is limited by the harmonics. The harmonics generate replicas during modulation/demodulation.

56

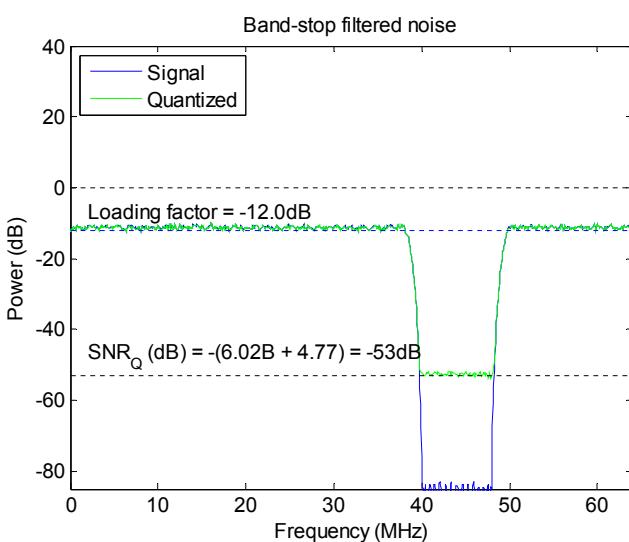
Quantization Noise; Wideband Noise



- 8-bit data
- Quantization increases the 'noise' (variance) in the power estimate
- In correlator systems, this corresponds to a loss in efficiency, since you have to average longer to reach the required signal-to-noise

57

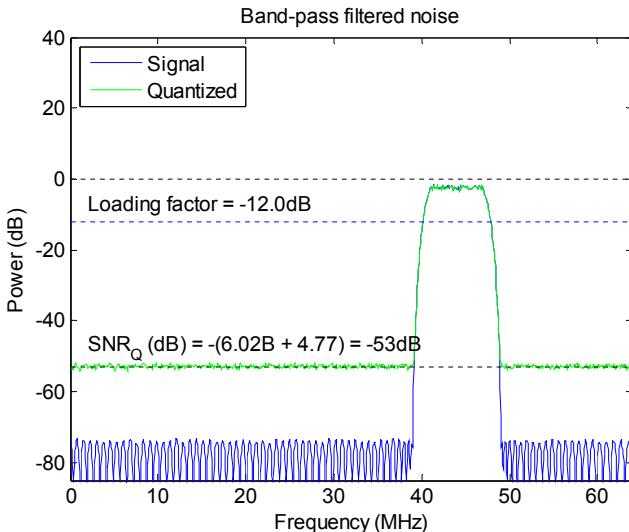
Quantization Noise; Band-Stop Noise



- 8-bit data
- Pass-band noise floor at -41dB
- **Noise Power Ratio (NPR)** is the depth of the passband to the quantization noise floor in a narrow bandwidth notch.
- NPR is a wideband ADC performance metric.

58

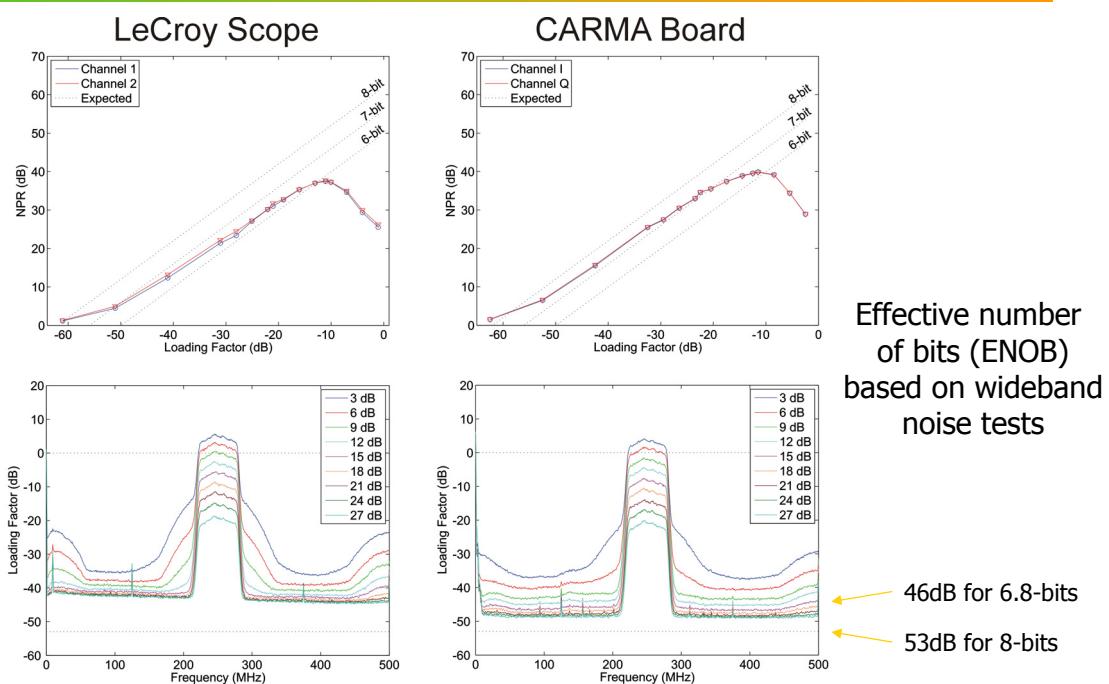
Quantization Noise; Band-Pass Noise



- 8-bit data
- Pass-band noise floor at -50dB
- **Over-sampling:**
 - SNR within the band is improved 6dB (1-bit) for every 4x over-sampling
 - After demodulation and decimation, the new signal will have higher power, so bit-growth is needed in the MSBs (or LSBs if data is scaled down)

59

Digitizer testing; Noise Power Ratio



60

Quantized Signals/Components

- Quantization of data and filter coefficients
- Floating-point to integer
 - round, ceil, floor, fix, ...
- Optimization for fixed-value integer coefficients;
Multipliers can be implemented via shift-and-add
 - Nearest power-of-2
 - Nearest number with fewest bits
 - Canonic Signed Digit (CSD)
 - Eg, $15 = 01111b$ or $16 - 1 = 1000\underline{1}b$
 - The underline, 1, means negative

61

FPGA Architectural Review

- Input/Output
- Phase-locked Loops (PLLs)
- Logic Elements
- Embedded Memory
- Hardware Multipliers and Accumulators

62

Input/Output

- LVCMOS
 - 1.5V/1.8V/2.5V/3.3V
- Programmable I/O delays
 - Meet external setup/hold and output requirements
- PCB space-saving features;
 - Bus hold, weak pull-ups, downs, on-chip terminations

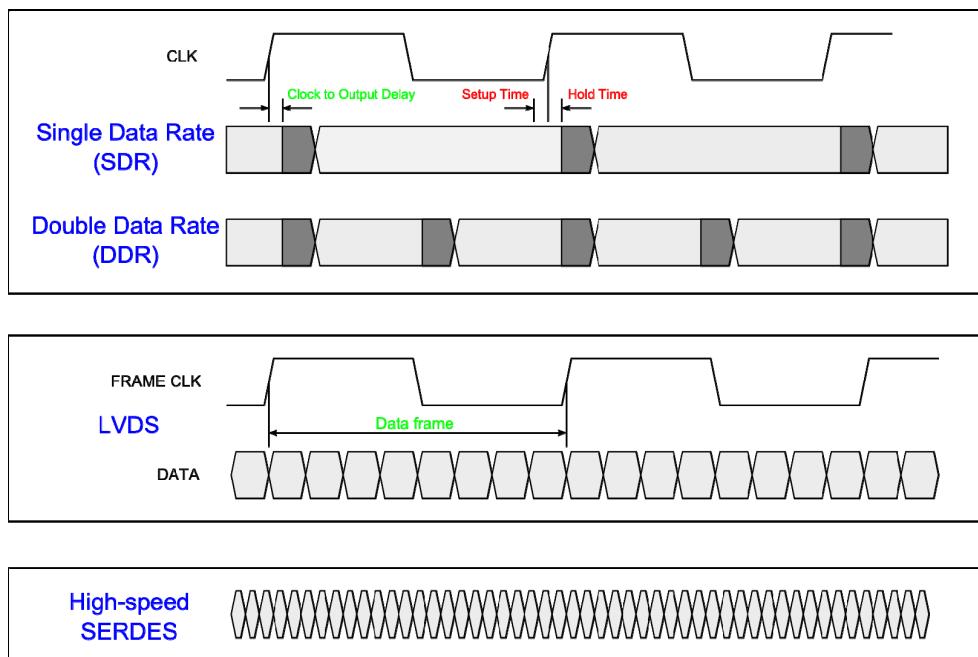
63

Input/Output

- High-speed memory support
 - DDR1/2/3
- LVDS serializer/deserializer (SERDES)
 - 1.6Gbps data rates
- Clock-and-data recovery (CDR)
 - 11.3Gbps data rates
 - JESD204A high-speed ADCs/DACs

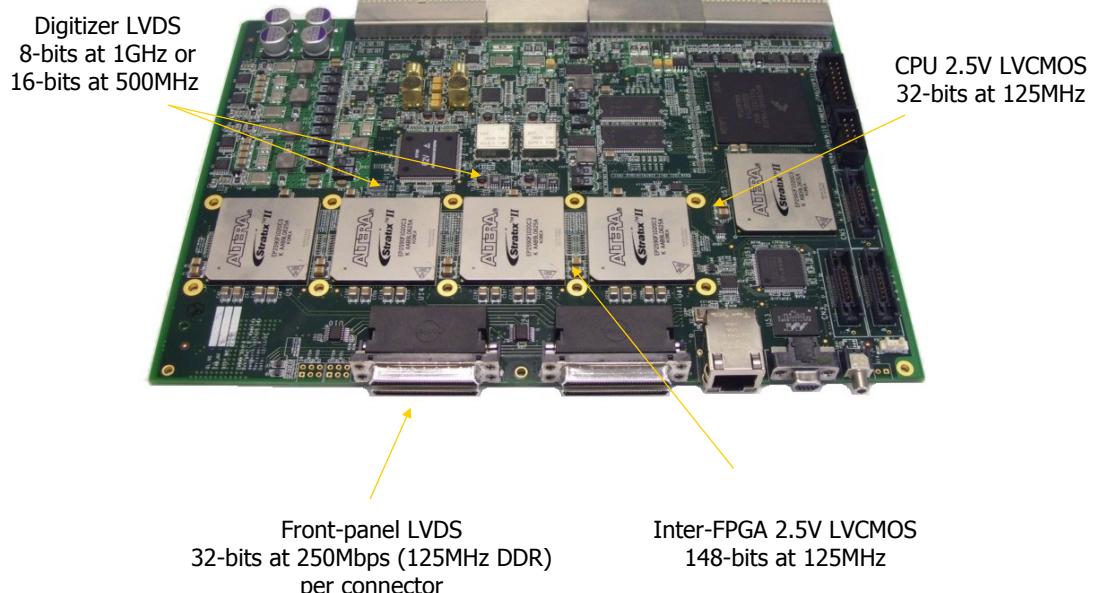
64

Input/Output



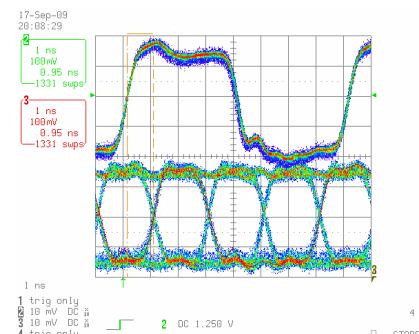
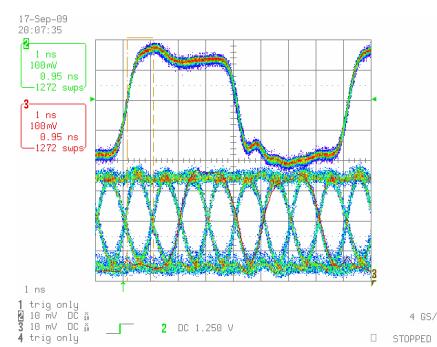
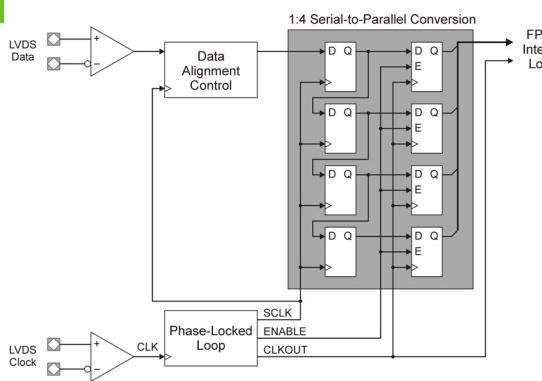
65

Input/Output



66

LVDS Deserializer



- LVDS deserializer
 - Serial-to-parallel conversion
- Oscilloscope traces;
 - 1GHz clock rate ADC
 - 125MHz 'frame' clock
 - 8-bits or 4-bits per frame

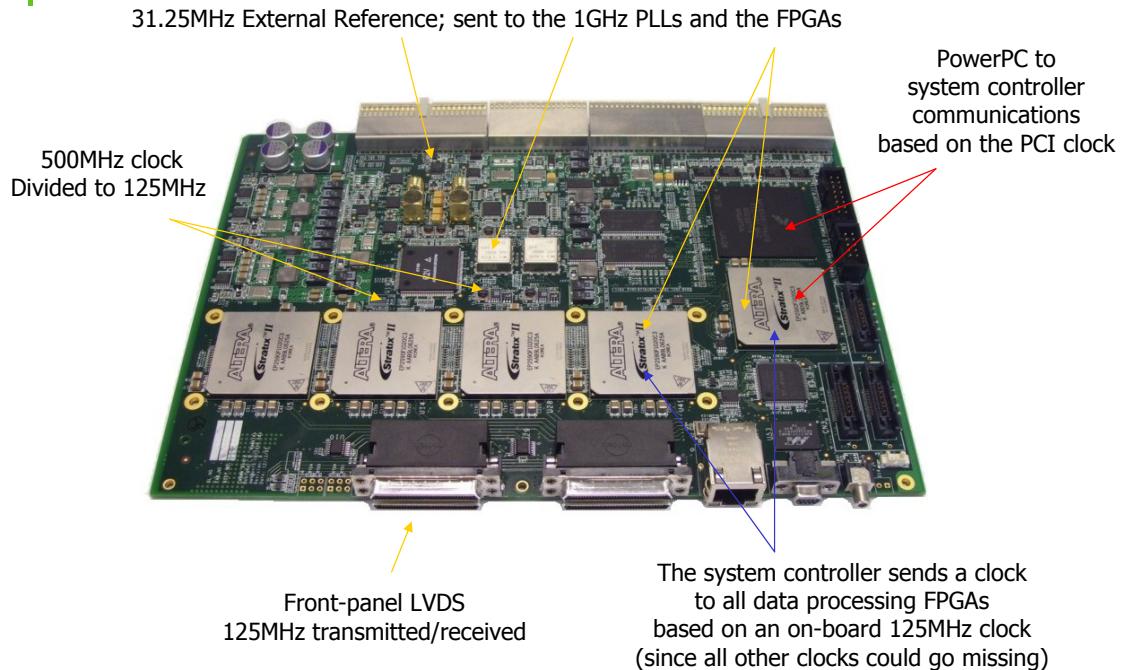
67

Phase-Locked Loops

- Used to;
 - generate multiple internal/external clocks
 - with user-defined phase relationships
- Allows system- and board-level distribution of a much slower reference clock
- Adjusting the timing of the generated clocks provides flexibility in meeting external device interface timing requirements

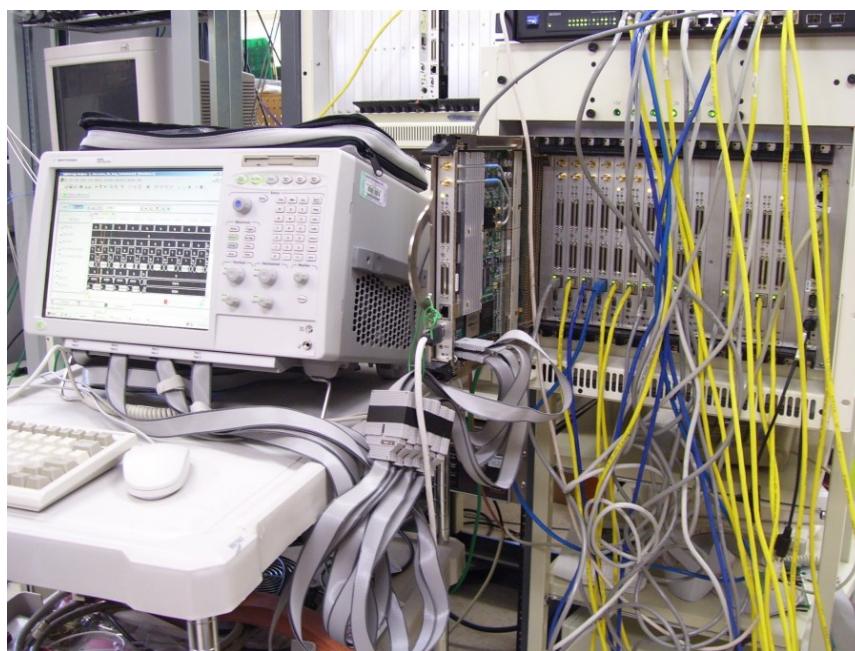
68

Phase-Locked Loops



69

Lab test/debug



70

Logic Elements

- The 'work horse' of the FPGA
- Architecture;
 - Basic logic element is a 4- or 6-input LUT with register
 - Hierarchical groupings of logic elements into blocks
 - Routing resources to implement a huge number of logic functions (at high-speed)
- Example uses;
 - Combinatorial logic, registers, finite state machines, counters, adders, FIFOs, and soft-core processors.

71

Embedded Memory

- Example sizes (Altera Stratix II);
 - M512 (512-bit), M4K (4K-bit), M-RAM (64kB)
 - Takes up a lot of FPGA die space
- Example uses;
 - FIFOs and dual-ported RAM (for clock-domain crossing)
 - Data width and rate changing
 - cos/sin table storage for NCOs
 - Coefficient storage for FIR filters
 - Look-up tables
 - RAM-based multipliers
 - Shift-registers
 - Cache memory for soft-core processors

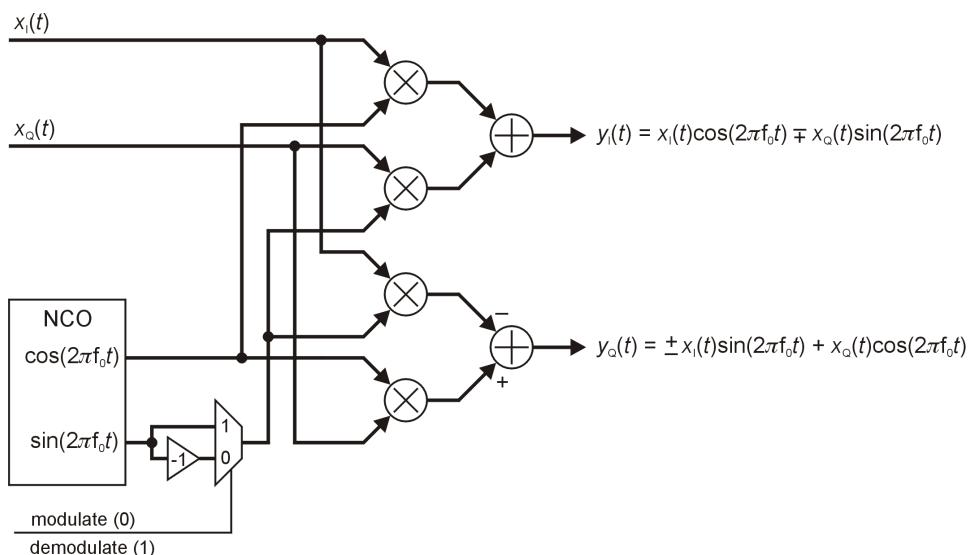
72

- DSP operations;
 - Multiply, Multiply-add, Multiply-accumulate
 - Complex-valued multiply
 - Real x complex (demodulation)
 - Re{ complex x complex } (modulation)
- FPGA DSP blocks;
 - 9x9, 18x18, 36x36 multipliers (Altera devices)
 - Sum/difference/accumulate
 - Register pipelines
 - Complex multiplies fit in a single DSP block
 - FIR/IIR filters structure map directly to DSP blocks

73

Complex Multiplication

$$(a + jb) \times (c + jd) = (ac - bd) + j(ad + bc)$$



74

FPGA-DSP Components

75

DSP Components

- Linear Feedback Shift Registers (LFSRs) and Pseudo-Random Binary Sequences (PRBSs)
- Numerically controlled oscillators (NCOs)
- Digital Upconversion/Downconversion
- FIR Filters
- IIR Filters

76

Linear Feedback Shift Registers

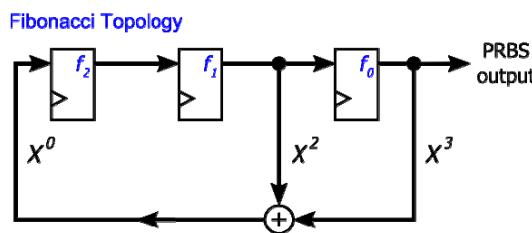
- Shift-registers with feedback taps implemented using XOR gates
- The output bit has 'random' noise properties, but is deterministic
- Applications;
 - Code division multiple access (CDMA)
 - SONET for data scrambling
 - SERDES for scrambling and bit error rate tests
 - High-speed ADCs (though JESD204A uses 8/10B)
 - Parallel output PRBS can be used for uniform and Gaussian noise sources.

77

Linear Feedback Shift Registers

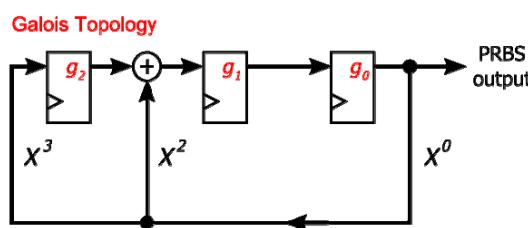
Generates a 1-bit output sequence with 'noise'-like properties

PRBS [3,2], $X^3 + X^2 + 1$, 1101b



f_2	f_1	f_0	
1	1	1	7
0	1	1	3
0	0	1	1
1	0	0	4
0	1	0	2
1	0	1	5
1	1	0	6
1	1	1	7

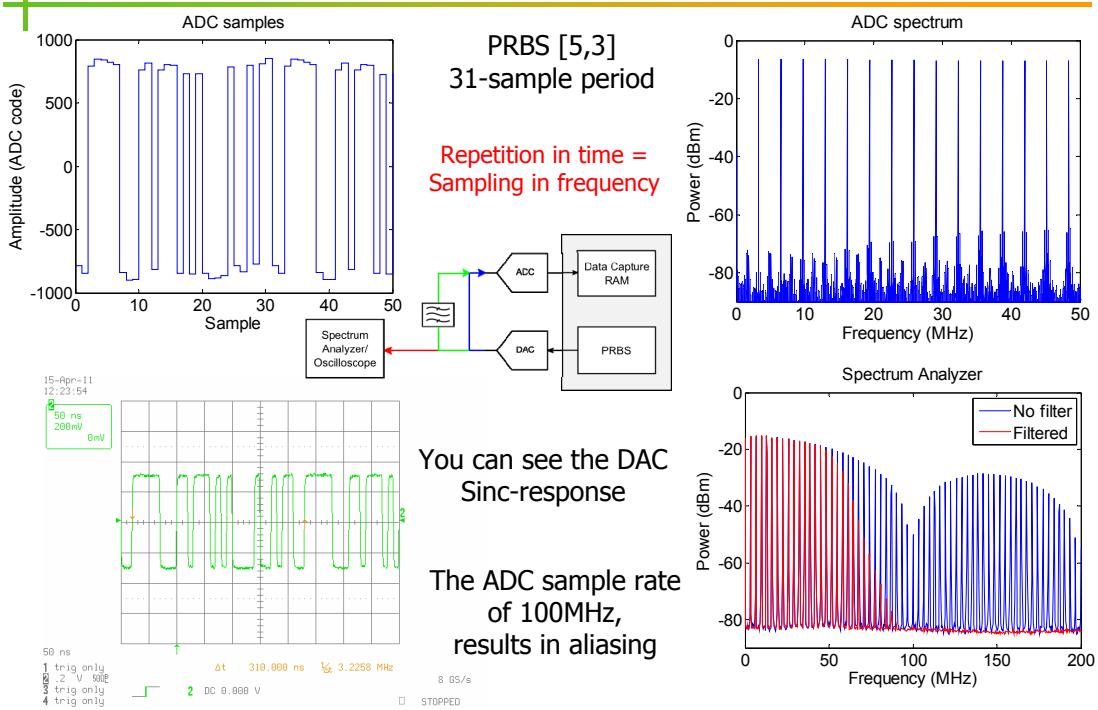
Same binary sequence



g_2	g_1	g_0	
1	1	1	7
1	0	1	5
1	0	0	4
0	1	0	2
0	0	1	1
1	1	0	6
0	1	1	3
1	1	1	7

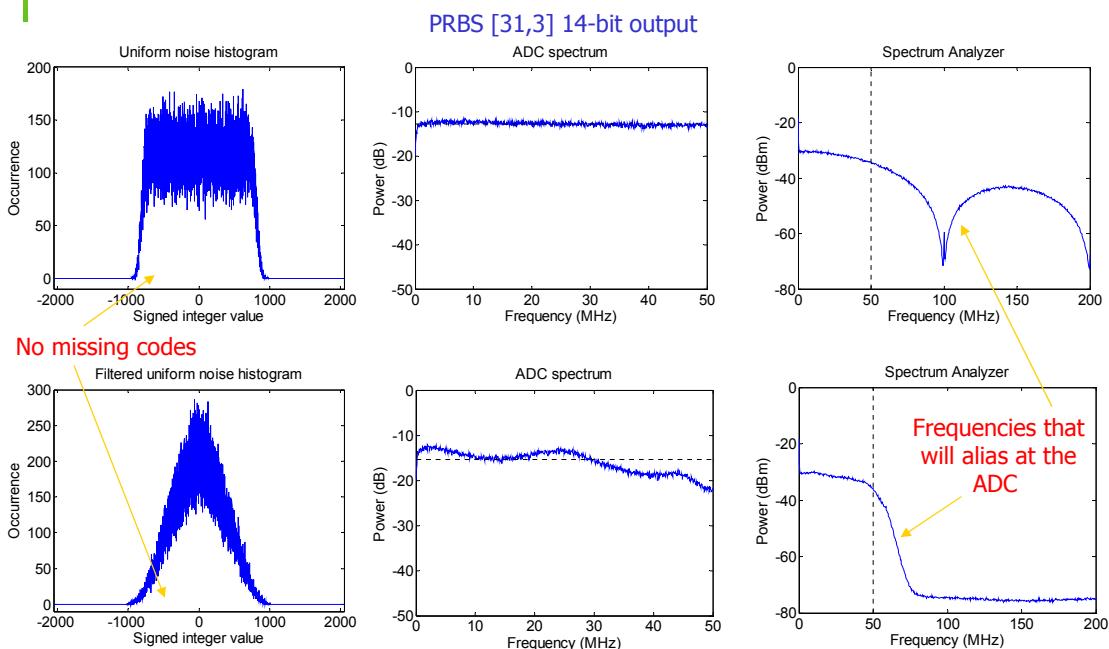
78

DAC/ADC tests using PRBS 'noise'



79

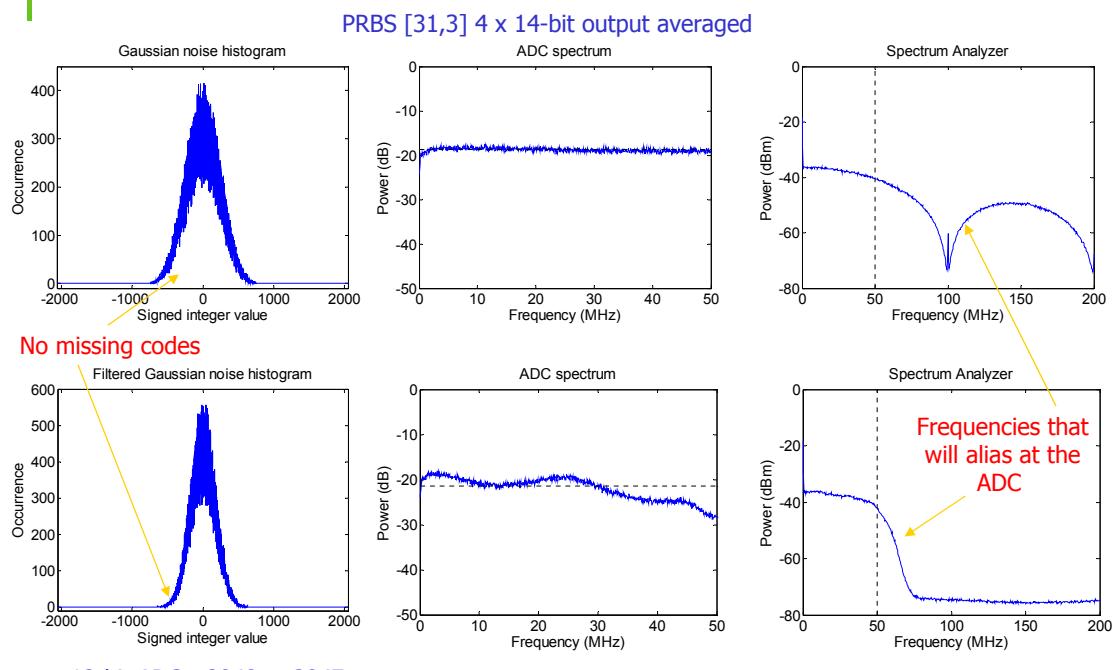
Parallel output PRBS 'uniform noise'



12-bit ADC; -2048 to 2047

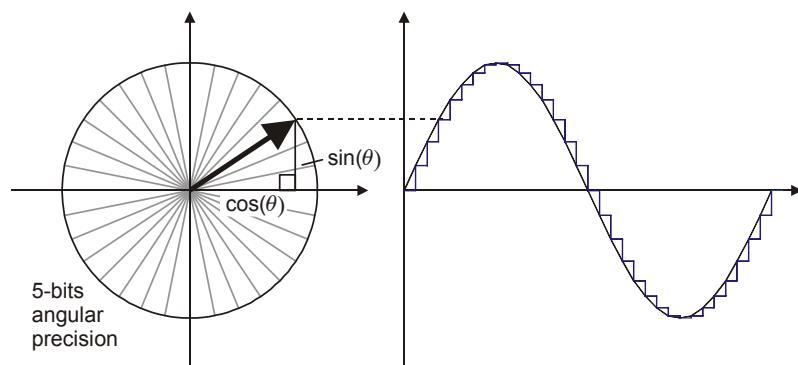
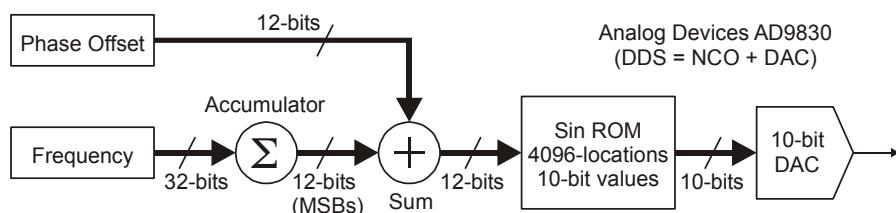
80

Parallel output PRBS 'Gaussian noise'



81

Numerically Controlled Oscillators



82

Numerically Controlled Oscillators

$$FTW = \text{round} \left(\frac{f_{out} \times 2^{B_{acc}}}{f_{clk}} \right)$$

- The NCO output frequency is determined by the frequency tuning word (FTW)
- The FTW depends on the NCO clock rate, and the number of bits in the accumulator
- Warnings:
 - Rounding of the FTW generates a phase/timing error
 - FTWs with fractional parts will generate harmonics

83

Numerically Controlled Oscillators

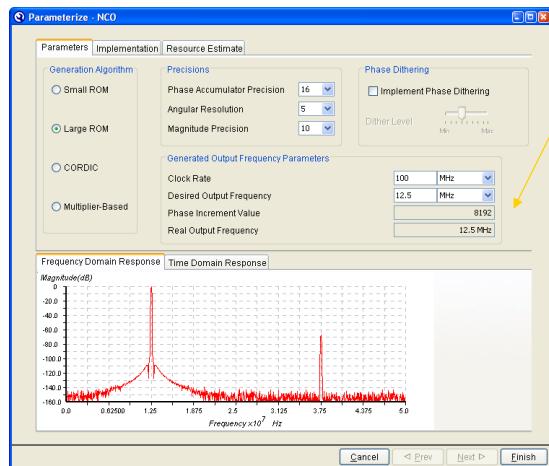
- Example:
 - NCO clock frequency of 100MHz
 - Output frequencies of 12.5MHz and 1.25MHz
 - 16-bit accumulator
 - ROM with 5-bits address (32 entries) and 10-bit data (amplitude quantization)

84

Altera Quartus NCO Design Tool

$$FTW = \frac{12.5 \times 2^{16}}{100} = 2^{13} = 8192$$

$$\begin{aligned} 2000h &= [00100] \bullet [000000000000] \\ &= [4.0] \end{aligned}$$



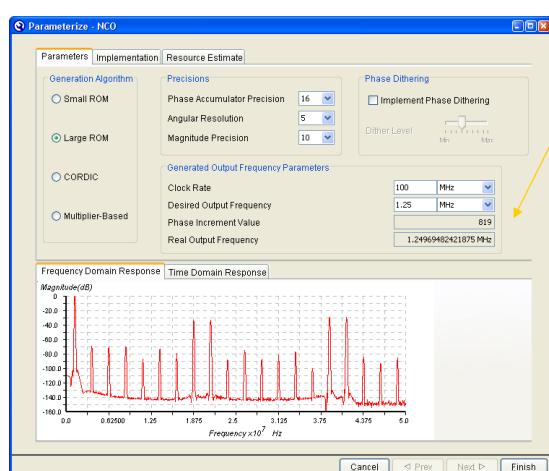
Quartus v10.1 NCO Compiler

85

Altera Quartus NCO Design Tool

$$FTW = \frac{1.25 \times 2^{16}}{100} = \frac{2^{13}}{10} = 819.2$$

$$\begin{aligned} 0333h &= [00000] \bullet [01100110011] \\ &= [0.3999] \end{aligned}$$



- Rounding required
- ROM address increment is fractional
- SNR limited by harmonics

86

Digital Upconversion/Downconversion

- Frequency Shift Fourier pair:

$$x(t) \exp(j2\pi f_0 t) \Leftrightarrow X(f - f_0)$$

- Euler's Formula:

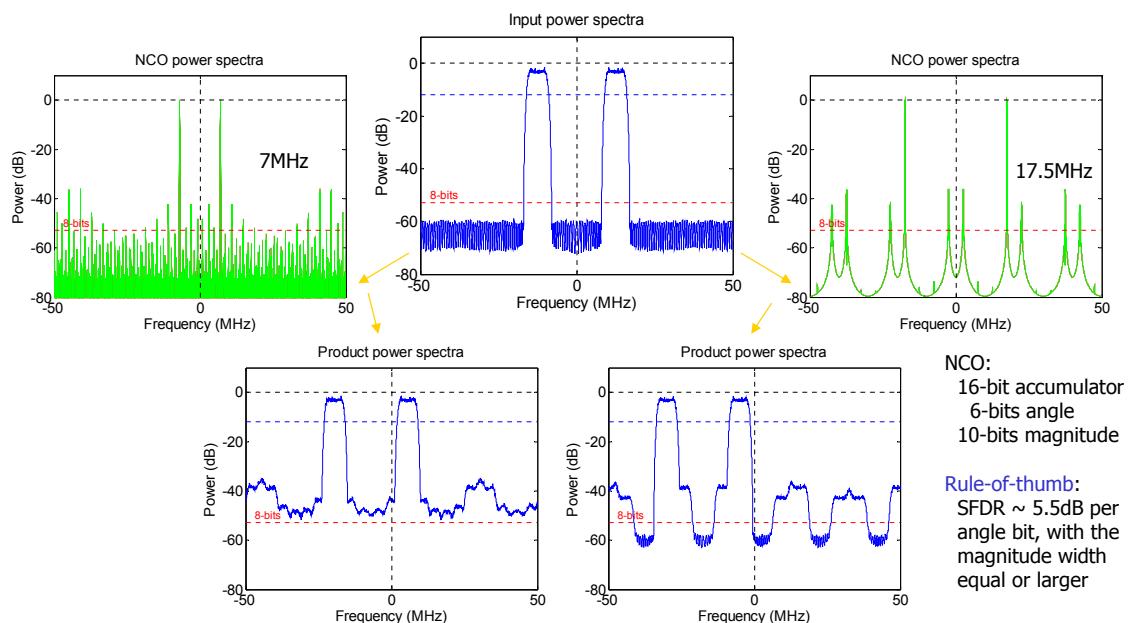
$$\exp(j2\pi f_0 t) = \cos(2\pi f_0 t) + j \sin(2\pi f_0 t)$$

- Terminology;

- Modulation = Digital Up Conversion (DUC)
 - Demodulation = Digital Down Conversion (DDC)

87

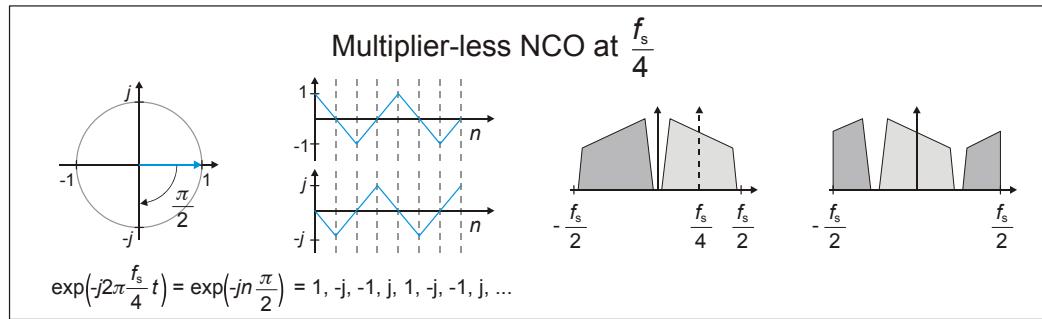
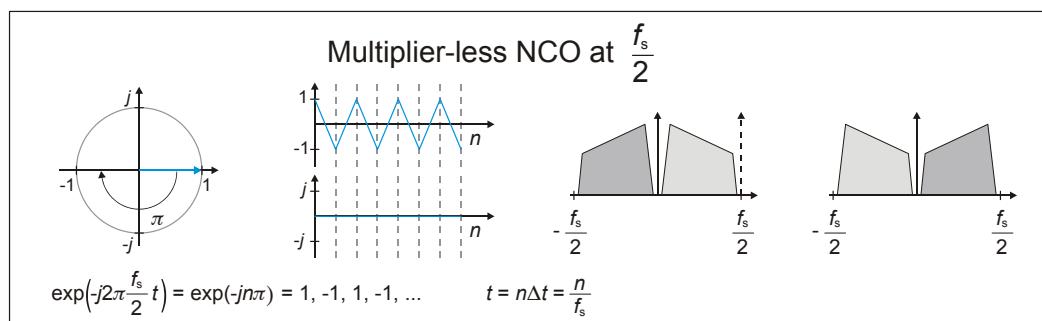
Digital Downconversion



The NCO harmonics are limiting the dynamic range; the NCO requires more angle bits

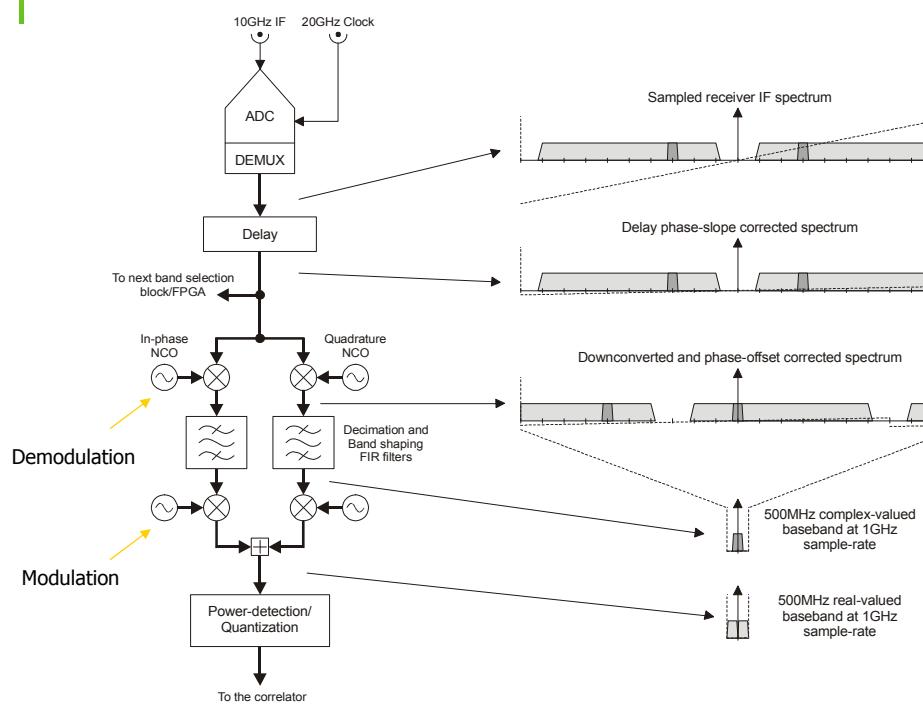
88

Digital Upconversion/Downconversion



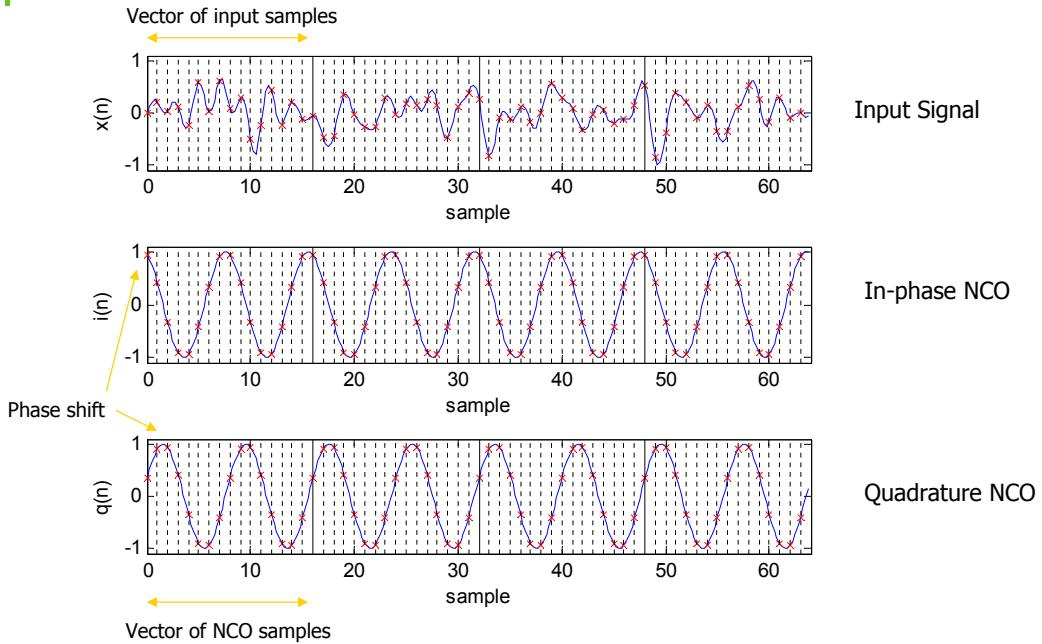
89

Digital Downconversion



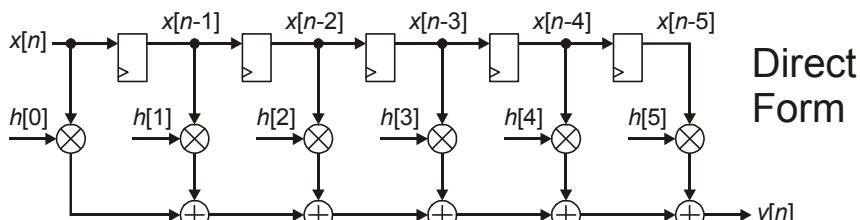
90

Digital Downconversion

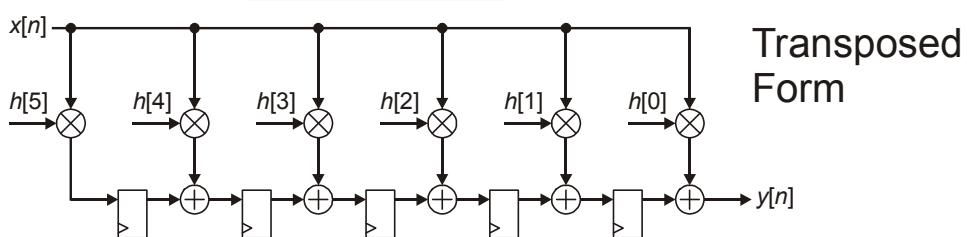


91

FIR Filters

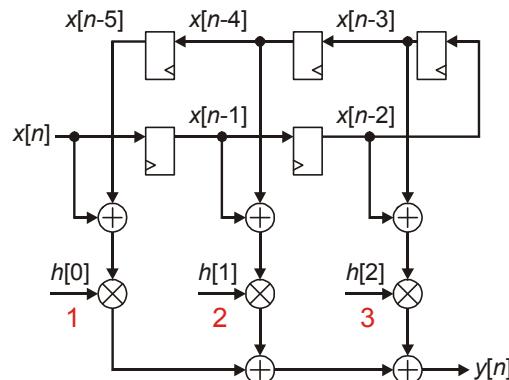
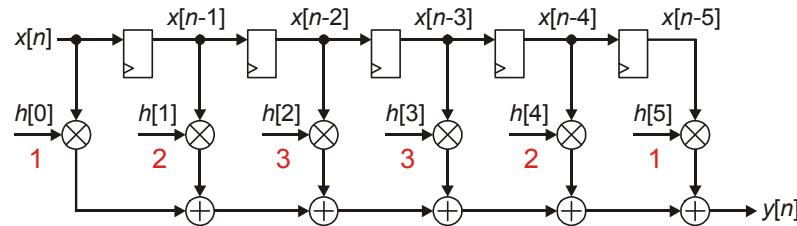


$$y[n] = \sum_{m=0}^{M-1} h[m]x[n-m]$$



92

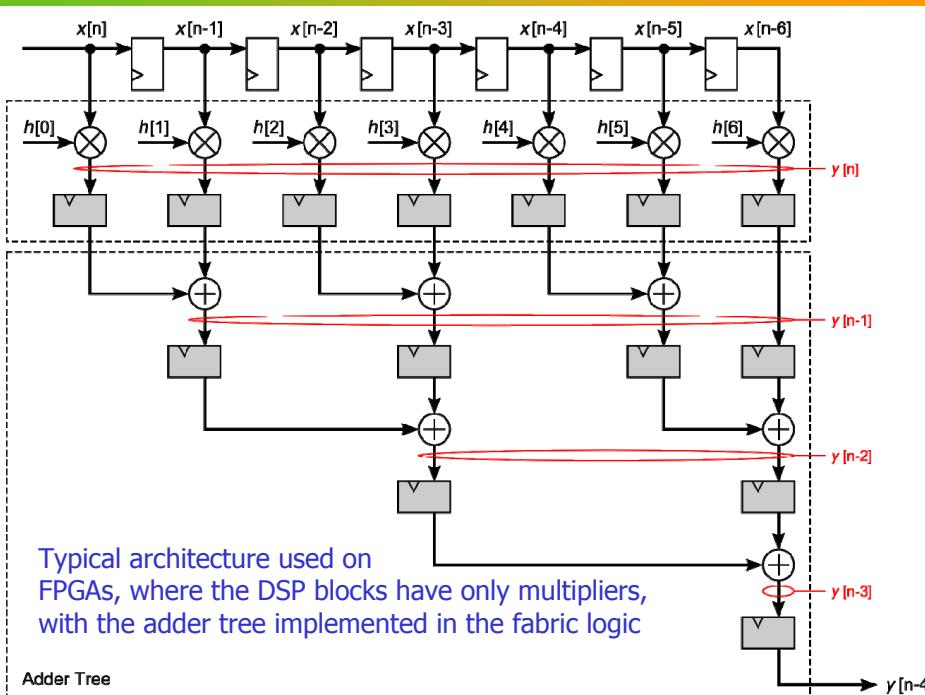
FIR Filters



Symmetry is used to eliminate multiplications

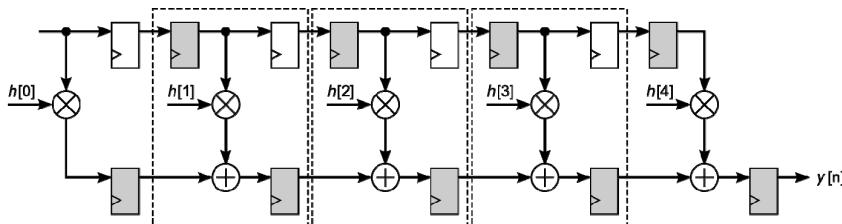
93

FIR Filter with Pipelined Adder Tree

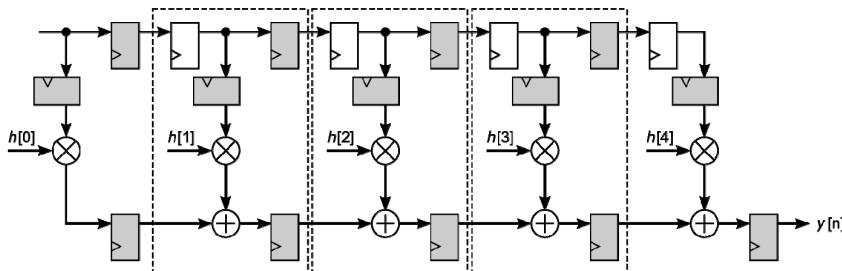


94

Systolic FIR Filters



Pipeline registers have been added

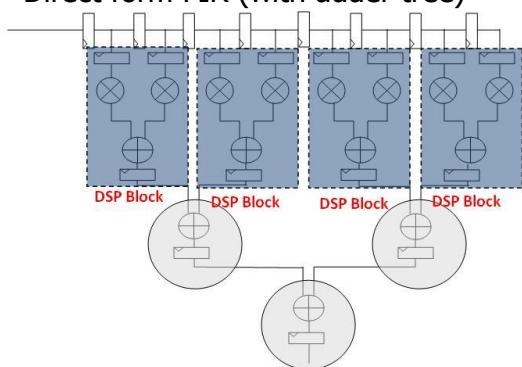


Pipeline registers can be added for the multiplier output too

95

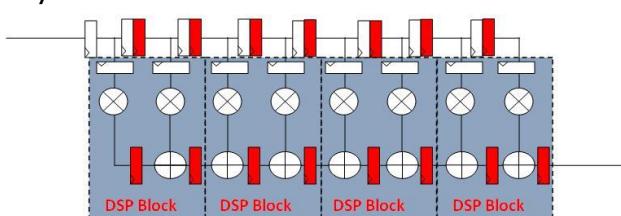
Which FIR architecture should you use?

Direct form FIR (with adder tree)



- The implementation depends on the **DSP block architecture**.
- The Direct form FIR filter with adder tree, implements the adder tree in the FPGA fabric, i.e., logic cells are used.
- The Systolic FIR sums are implemented using DSP Blocks

Systolic FIR



Stratix V DSP blocks
(images from the Altera Wiki)

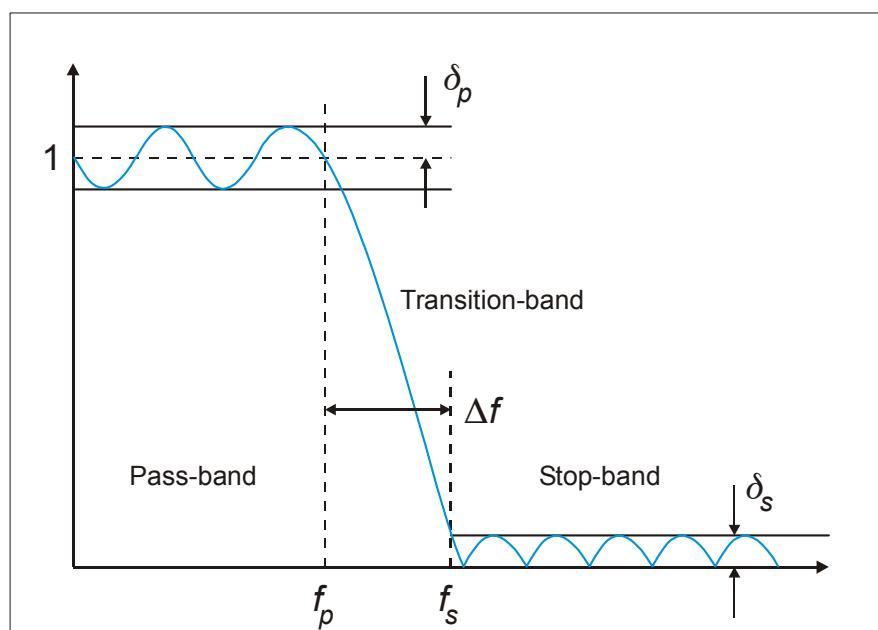
96

More FIR Filter Terminology

- Time-Division Multiplexed (TDM) FIR filters
- Multi-channel FIR filters;
 - The filter is clocked at a higher-rate than the data, and multiple channels are processed using the same filter logic
- Folded FIR filter
 - The entire filter is implemented with just one or a few DSP blocks (and RAM for taps and data)
 - Eg., Multiply-accumulate FIR (MACC-FIR)

97

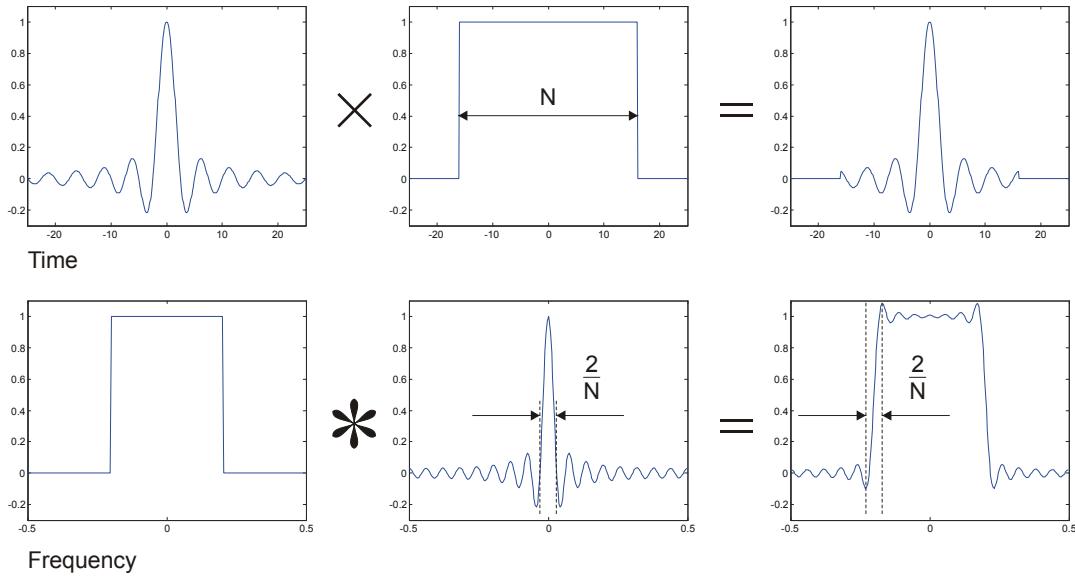
Filter Specification



98

Window-method FIR Filter Design

Time domain limiting produces Frequency domain ripples

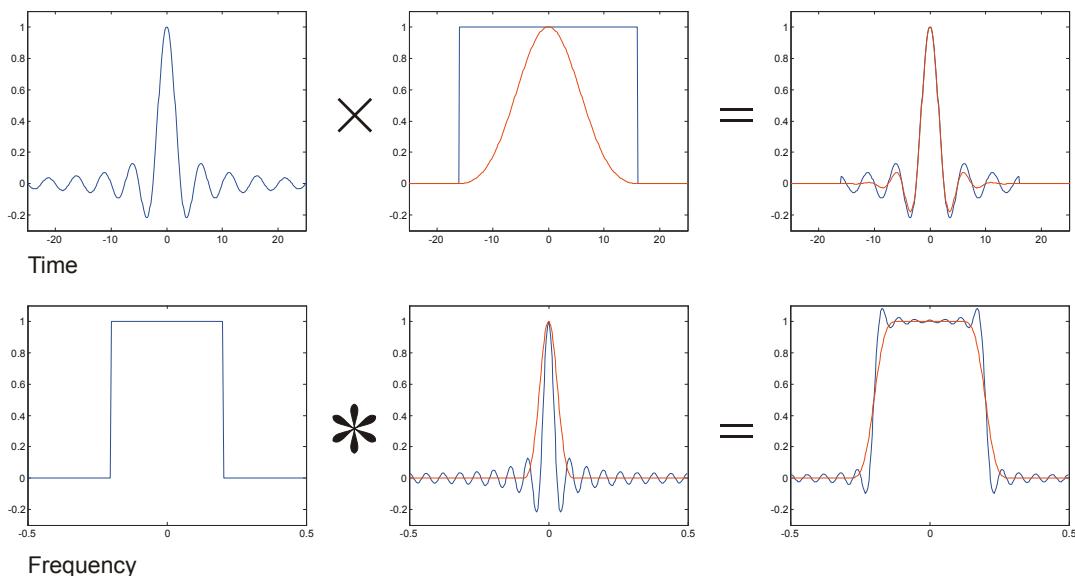


The transition width depends on the number of time domain samples, N

99

Window-method FIR Filter Design

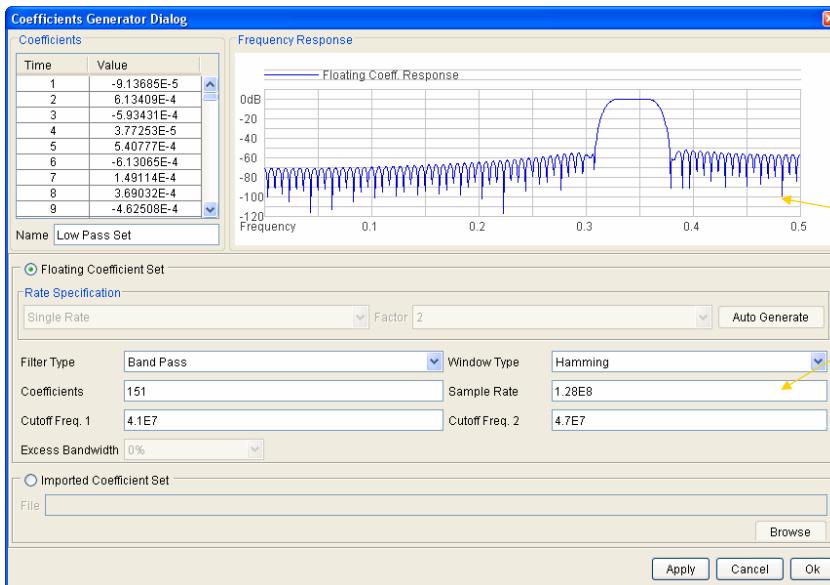
Time domain windowing reduces Frequency domain ripples



100

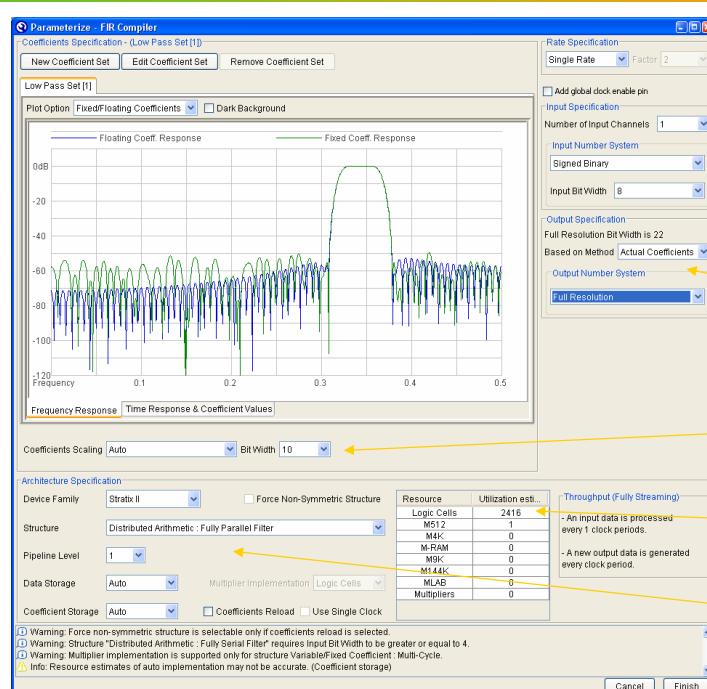
Altera Quartus FIR Design Tool

FIR design using the 'window-method'



101

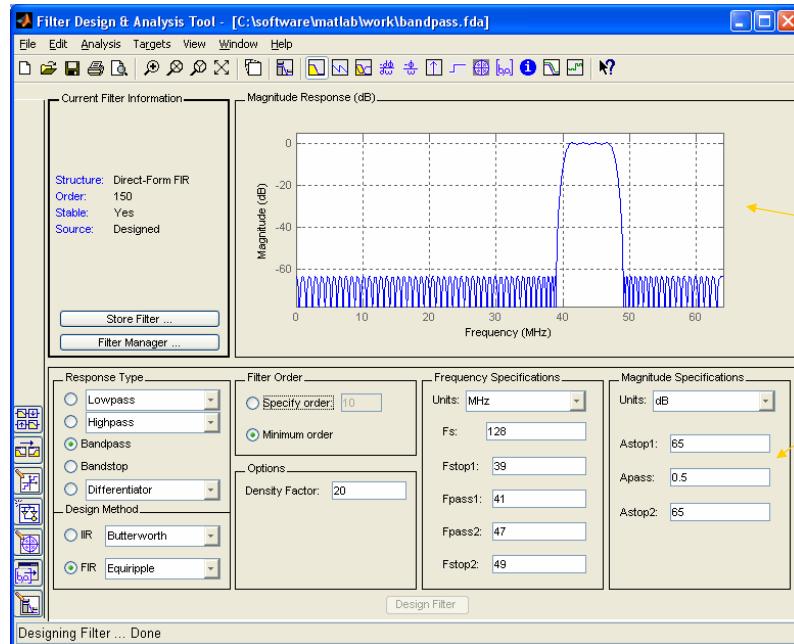
Altera Quartus FIR Design Tool



102

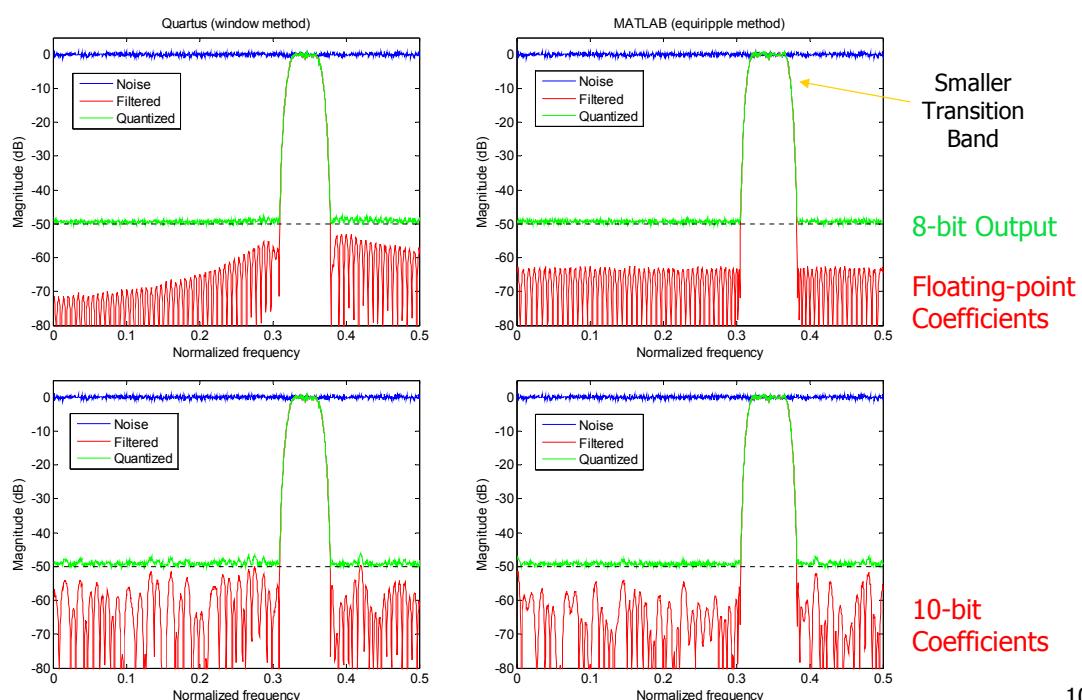
MATLAB FIR Design Tool

FIR design using the 'equiripple method'



103

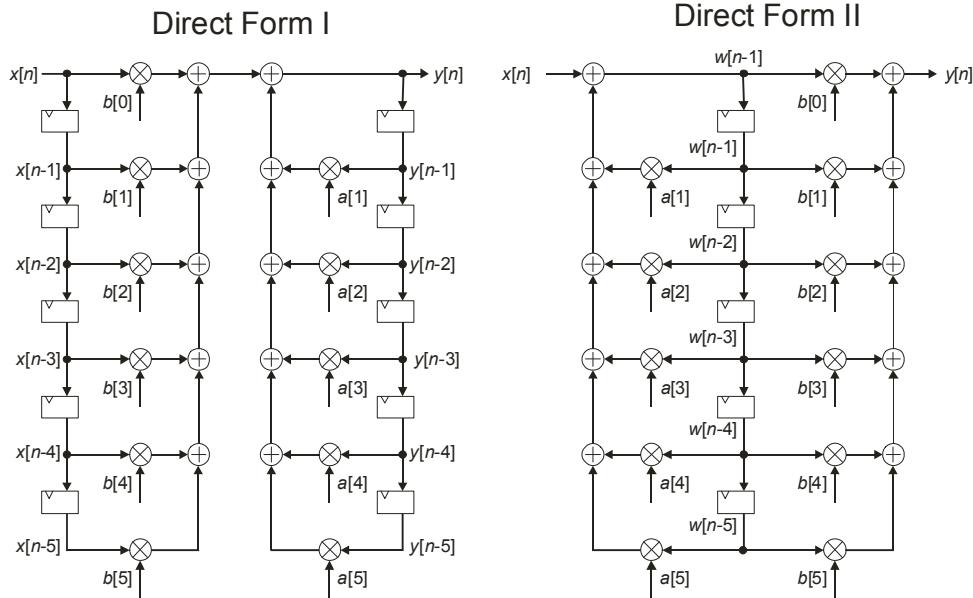
FIR Filtered Noise



104

IIR Filters

$$y[n] = \sum_{m=0}^{M-1} b[m]x[n-m] + \sum_{k=1}^{K-1} a[k]y[n-k]$$



105

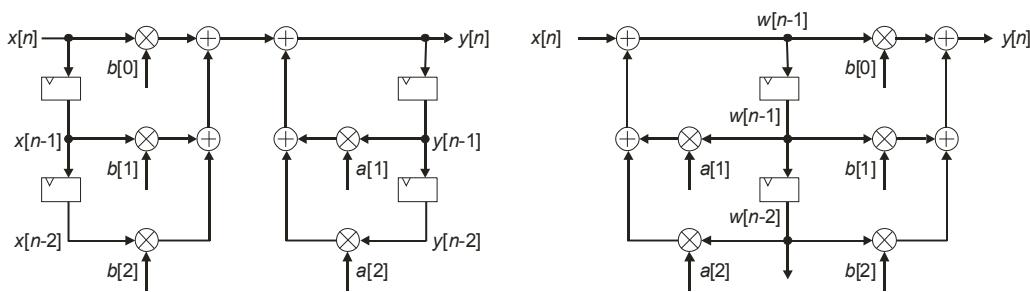
IIR Filters

- More ‘bang for the buck’ than an FIR
- Coefficient quantization can cause issues
- Phase-response;
 - Non-Linear phase is typical
 - Linear phase is possible (using all-pass filters)
- Design methods;
 - Directly in the Z-plane with poles and zeros
 - Product/sums of Biquadratics (Second-order-sections)
 - Parallel all-pass filters
 - Lattice Wave Digital Filters
 - Design an Analog Filter and bilinear transform

106

Biquadratic/Second-order-section

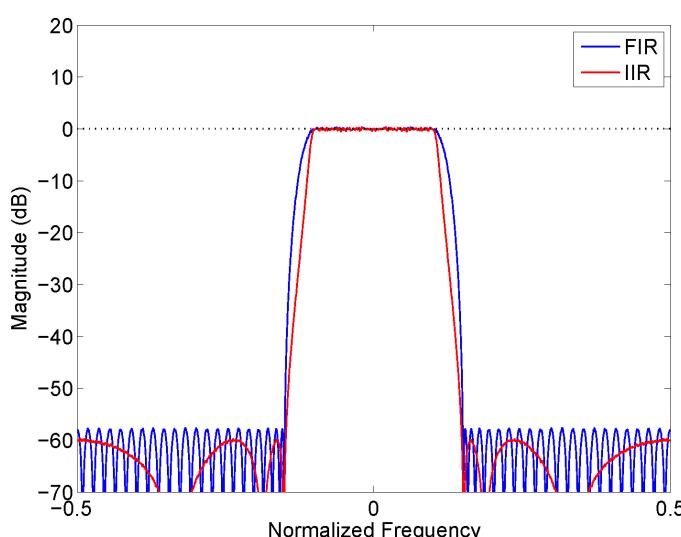
$$H(z) = \frac{b[0] + b[1]z^{-1} + b[2]z^{-2}}{1 - a[1]z^{-1} - a[2]z^{-2}}$$



Efficient implementation; Butterworth low-pass $b[n] = \{1,2,1\}$, and high-pass $\{1,-2,1\}$

107

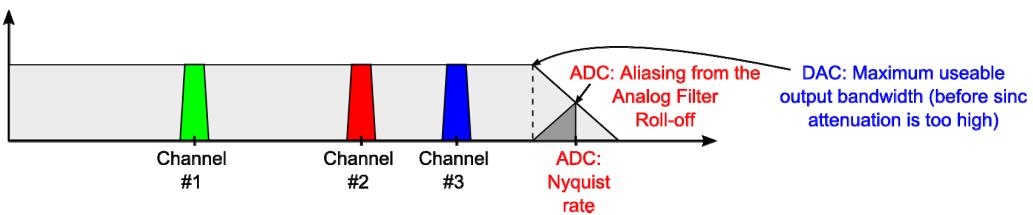
Low-pass FIR vs IIR



- Requirement:
 - Passband 0.1
 - Stopband 0.15
 - 0.1dB passband ripple
 - 60dB stopband attenuation
- FIR
 - 55 coefficients
 - Linear phase
- IIR
 - 3 SOS
 - Non-linear phase

108

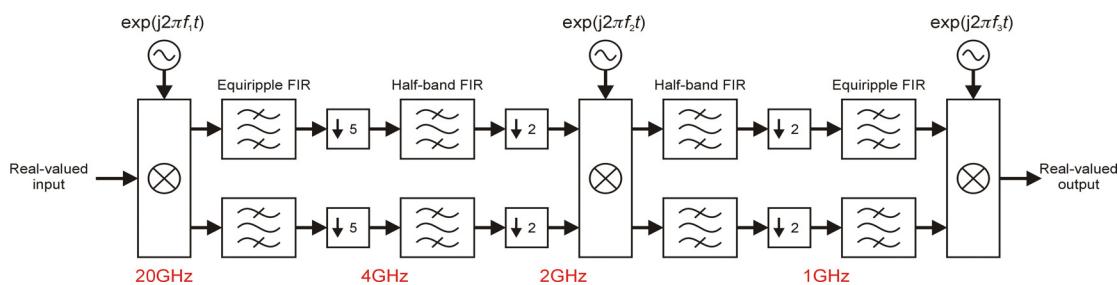
Multi-rate Processing



- Decimation;
 - Input from high-speed ADC
 - Digital downconvert (DDC) channel(s)
 - Resample to a lower sample rate
 - Process the channel(s)
- Interpolate;
 - Process the channel(s)
 - Resample to a higher sample rate
 - Digital up convert (DDC) and sum channels
 - Sinc correct
 - Output to high-speed DAC
- Resampling uses digital filters

109

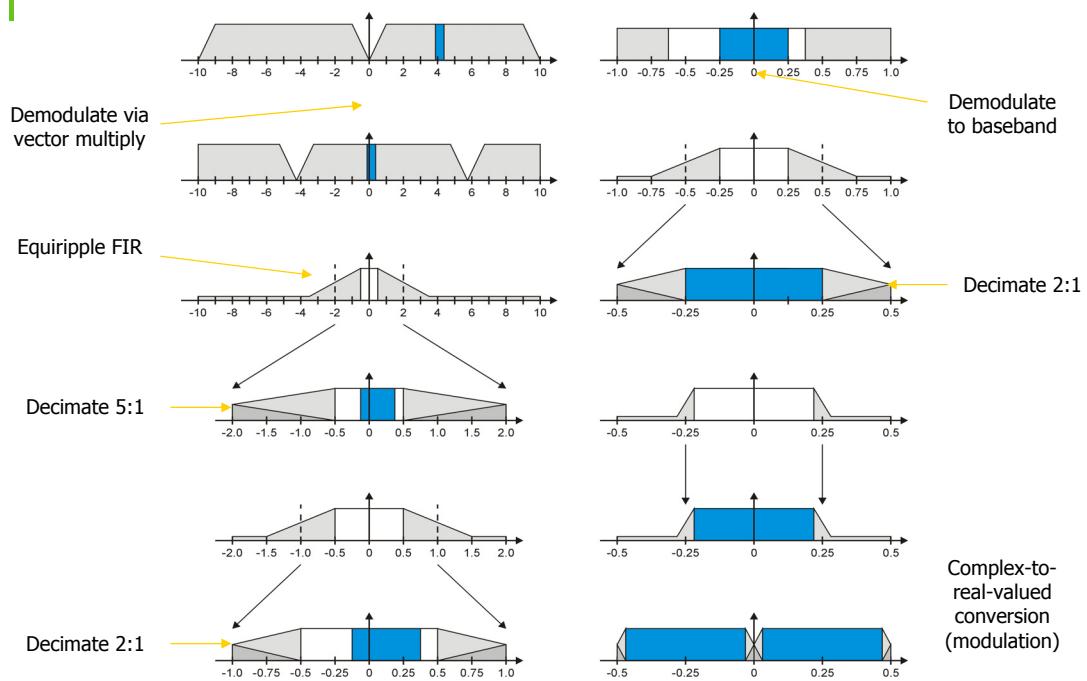
Multi-rate Downconversion/Decimation



- Micram ADC - 20GHz 6-bit sampler
 - Demux-by-4
 - 24-bits at 5GSps ADC output
- Decimate down to 1GHz sample rate

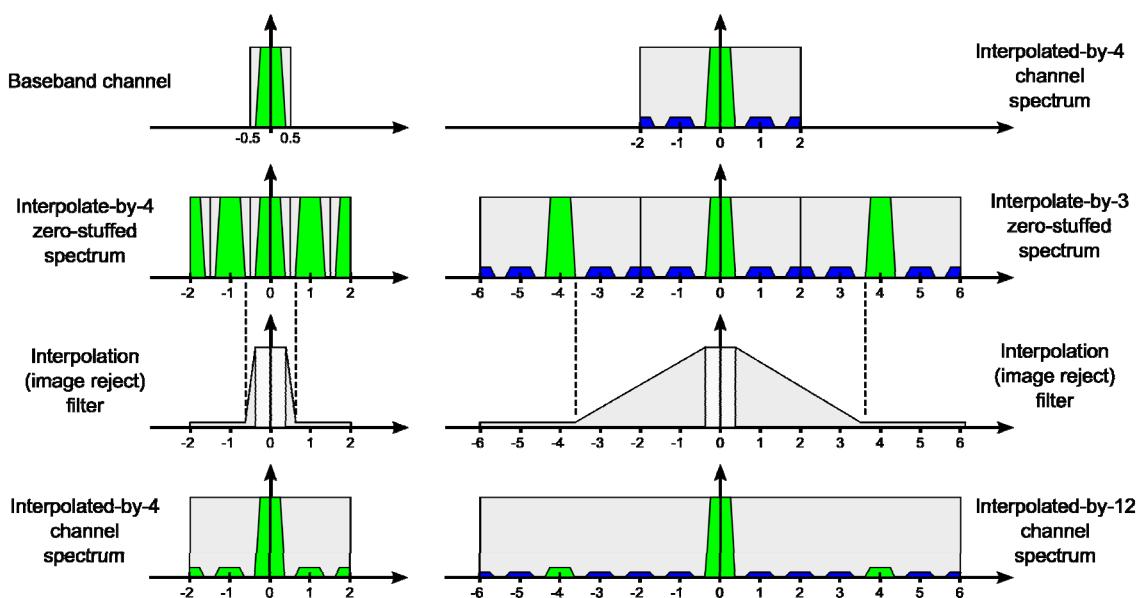
110

Multi-rate Downconversion/Decimation



111

Interpolation



112

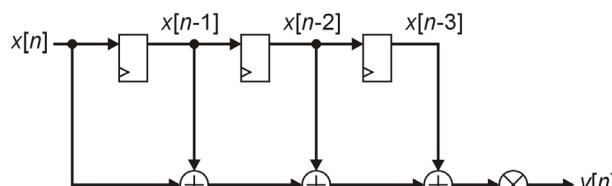
Example Designs

- Moving average FIR filter
- Half-band filter
- Simple IIR filter
- Bandpass processing
 - Aliasing digital downconversion
 - Multi-stage with multiplier-less NCOs
 - Wideband ADC
- Detailed FIR filter analysis

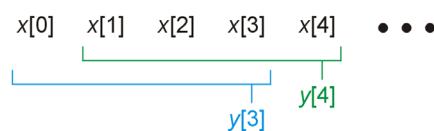
113

Moving Average Filter

4-sample moving average filter



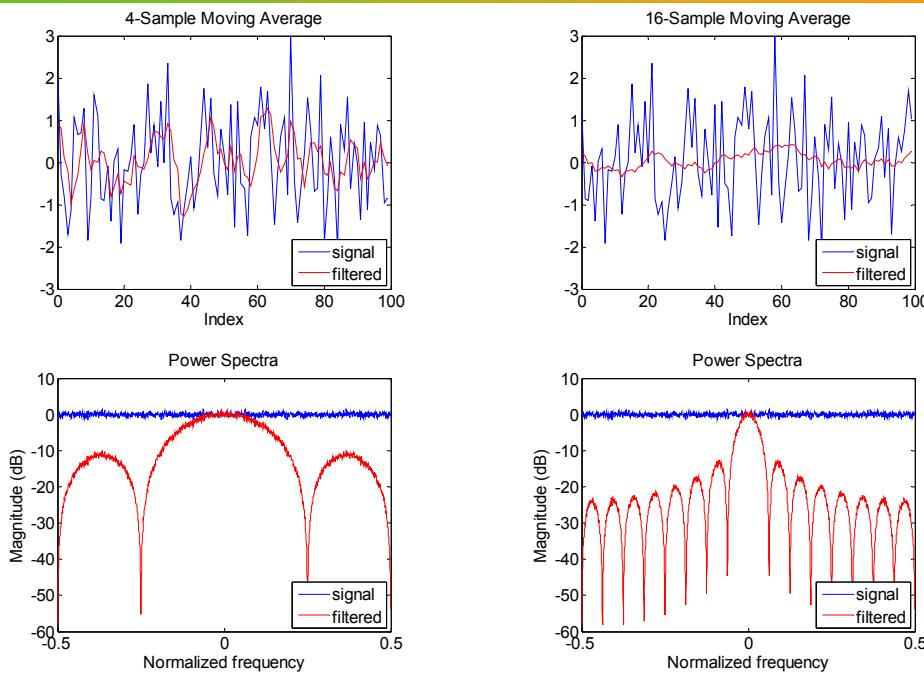
$$y[n] = \frac{1}{4} \sum_{m=0}^3 x[n-m]$$



Multiplier 'free' (for a power-of-2 average)

114

Moving Average Filter



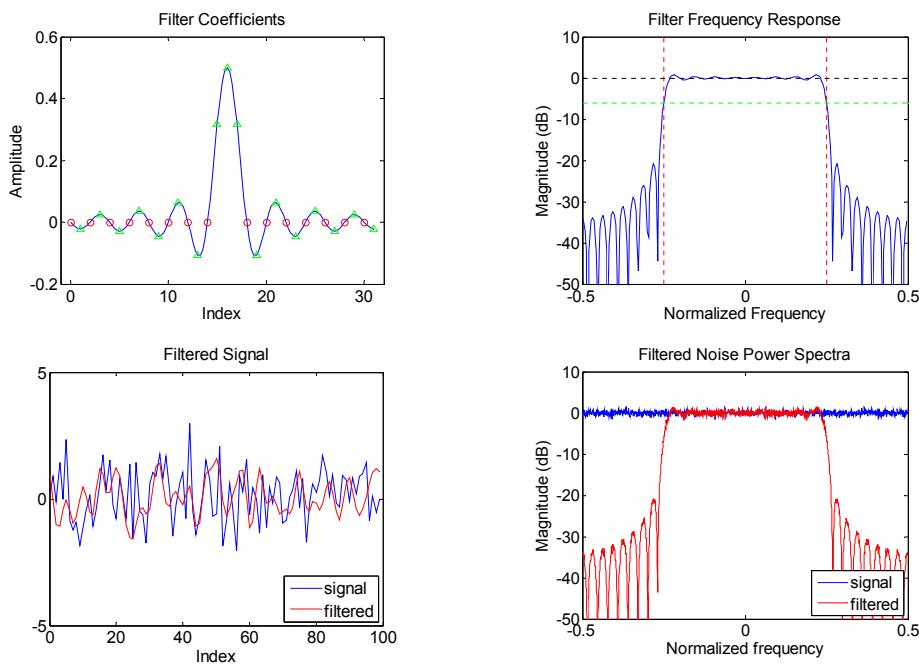
115

Moving Average Filter

- A FIR filter is generally designed for;
 - Flat pass-band
 - Fast pass-band to stop-band transition
 - Low stop-band sidelobes
- A moving average filter has none of these!
 - But, they are multiplier-free
 - They are used in multi-stage designs, where later filter stages are used to correct their shortcomings
- AKA Cascade Integrator Comb (CIC) Filters

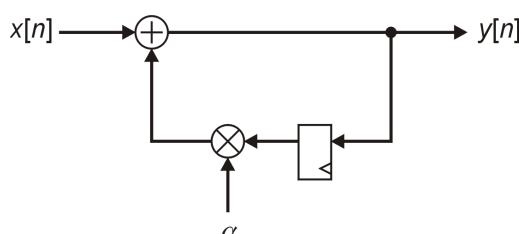
116

Half-band Filter



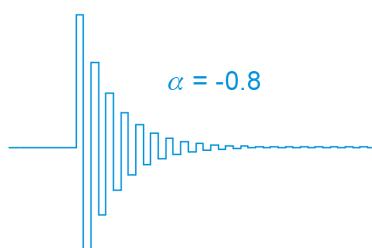
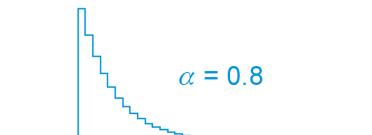
117

Simple IIR filter



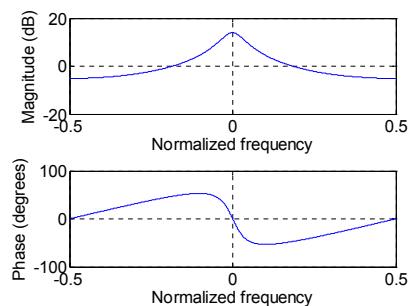
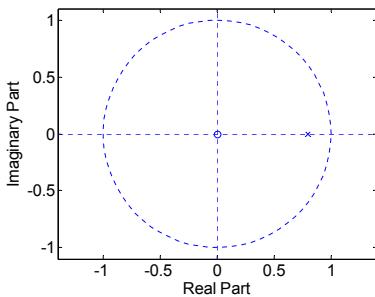
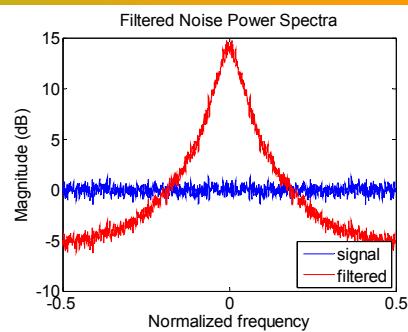
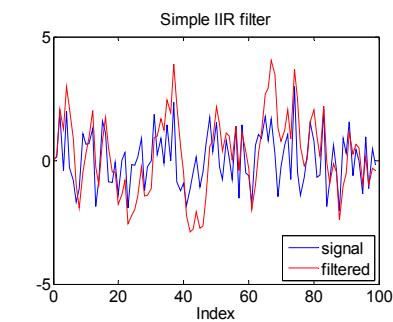
$$y[n] = x[n] + \alpha y[n-1]$$

$x[n]$	1	0	0	0	• • •
$y[n]$	1	α	α^2	α^3	• • •



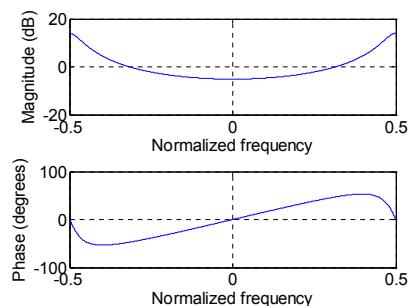
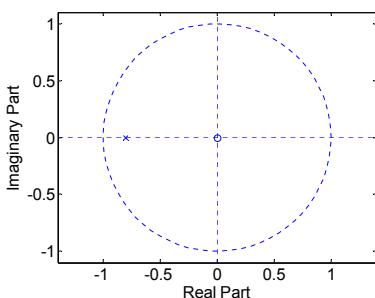
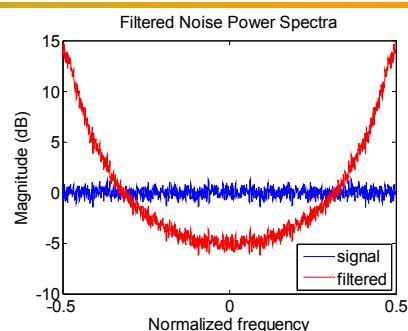
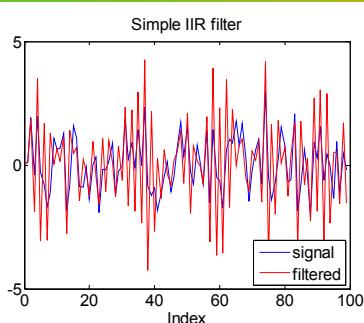
118

Low-pass IIR ($\alpha = 0.8$)



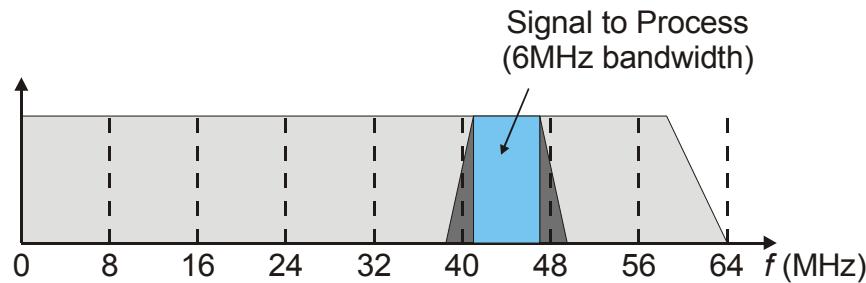
119

High-pass IIR ($\alpha = -0.8$)



120

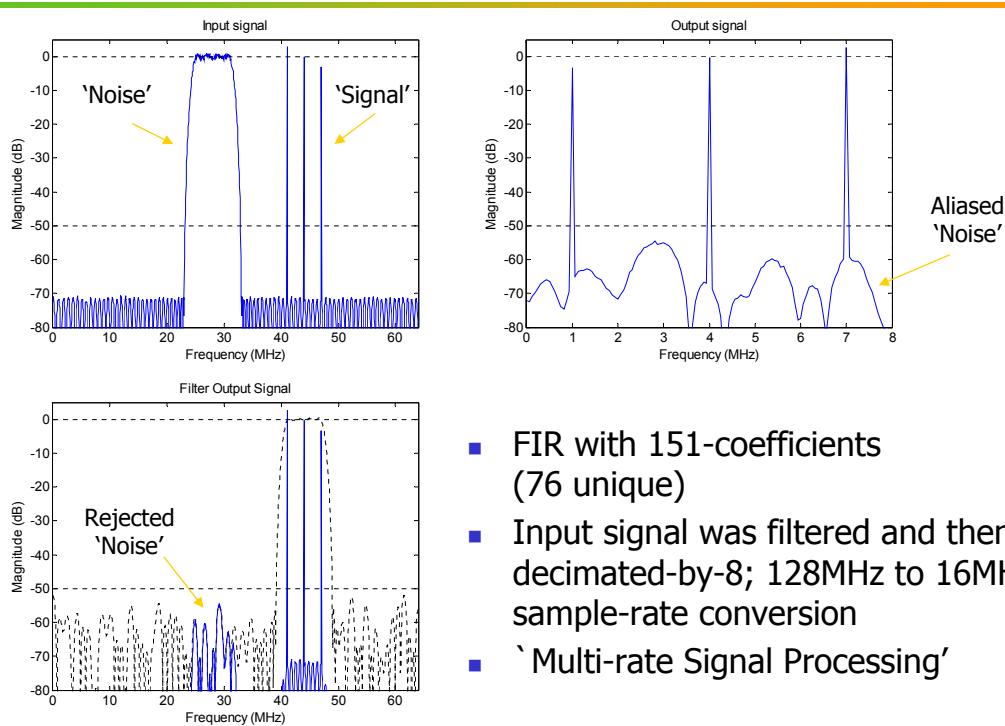
Bandpass Processing Example



- Process a 6MHz band from a signal sampled at 128MHz
- The band is located within a 'Nyquist zone' of a 16MHz sampling frequency (i.e., aligned to 8MHz)
- 'NCO-less' digital downconversion is possible

121

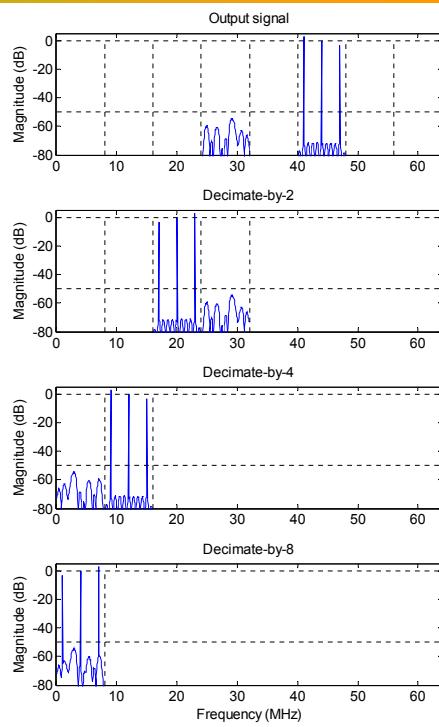
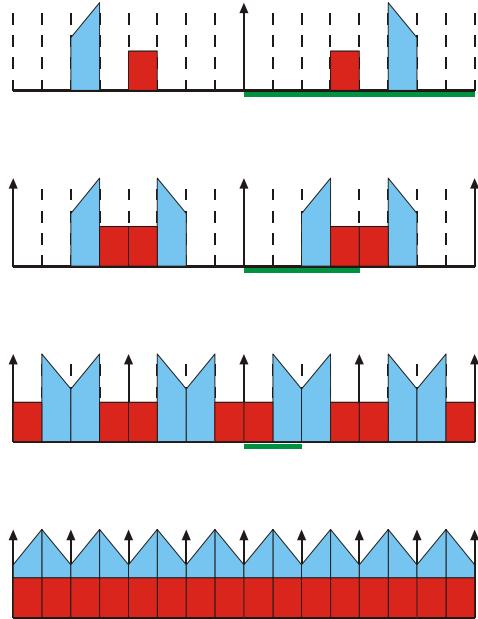
Aliasing downconversion



- FIR with 151-coefficients (76 unique)
- Input signal was filtered and then decimated-by-8; 128MHz to 16MHz sample-rate conversion
- 'Multi-rate Signal Processing'

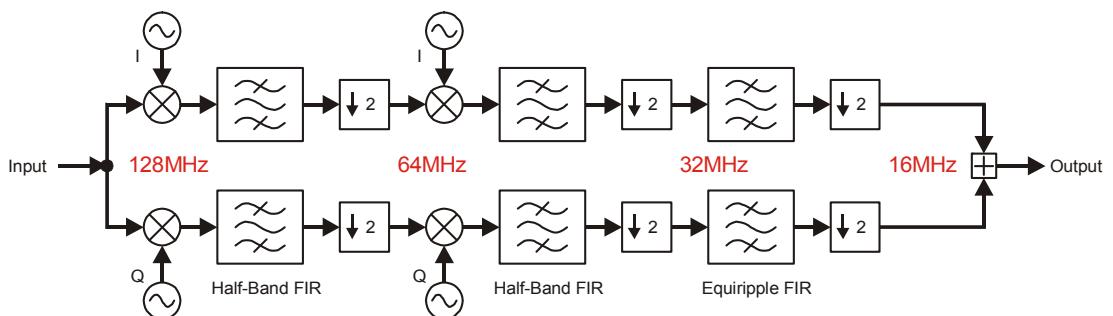
122

Aliasing downconversion



123

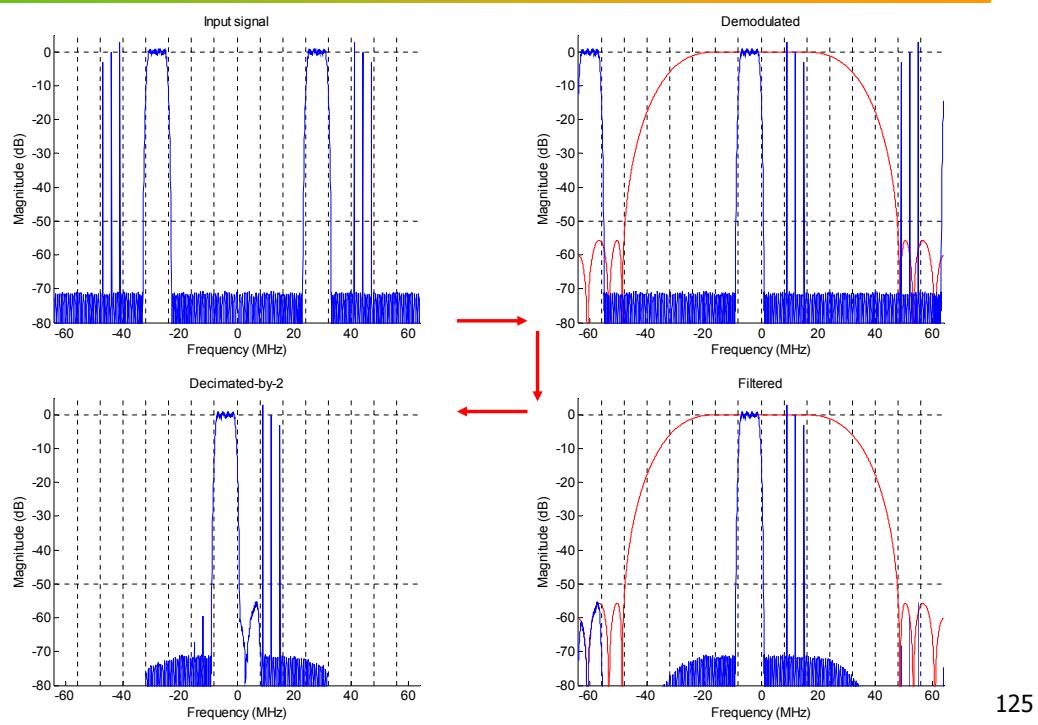
Multi-stage Multi-rate Processing



- Multiplier-less NCOs; $1, j, -1, -j, \dots$
- FIR filters;
 - Half-band FIR filter; 11 coeffs (4 unique)
 - Equiripple FIR filter; 38 coeffs (19 unique)
 - Multipliers; $2 \times (4 + 4 + 19) = 54$ ($54/76 = 71\%$)

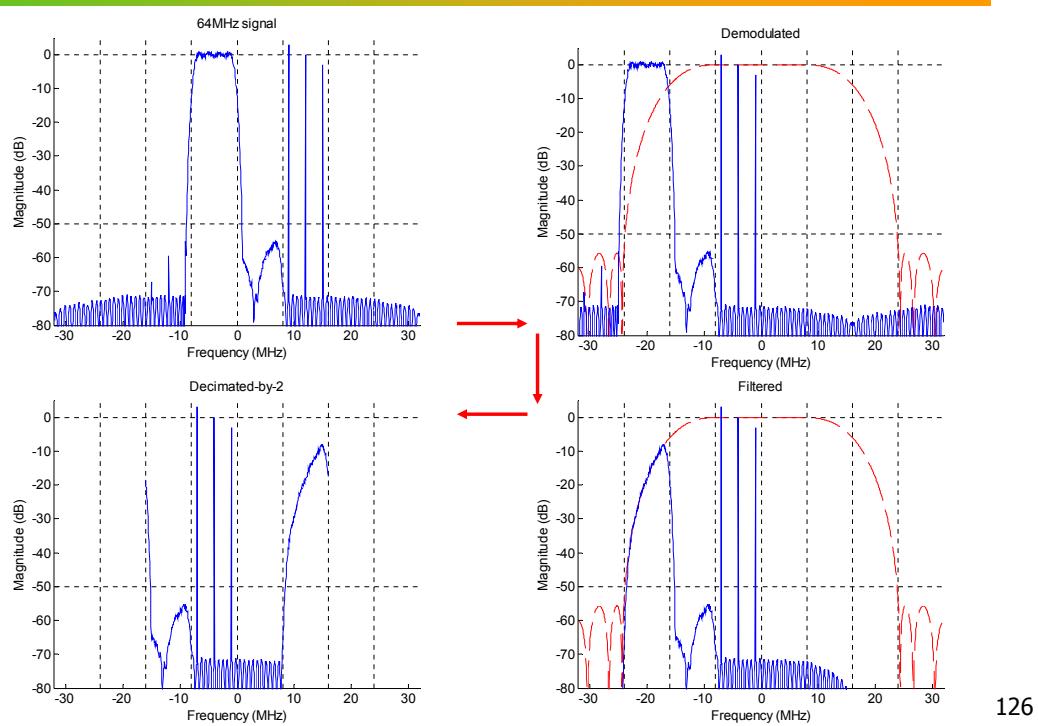
124

128MHz to 64MHz



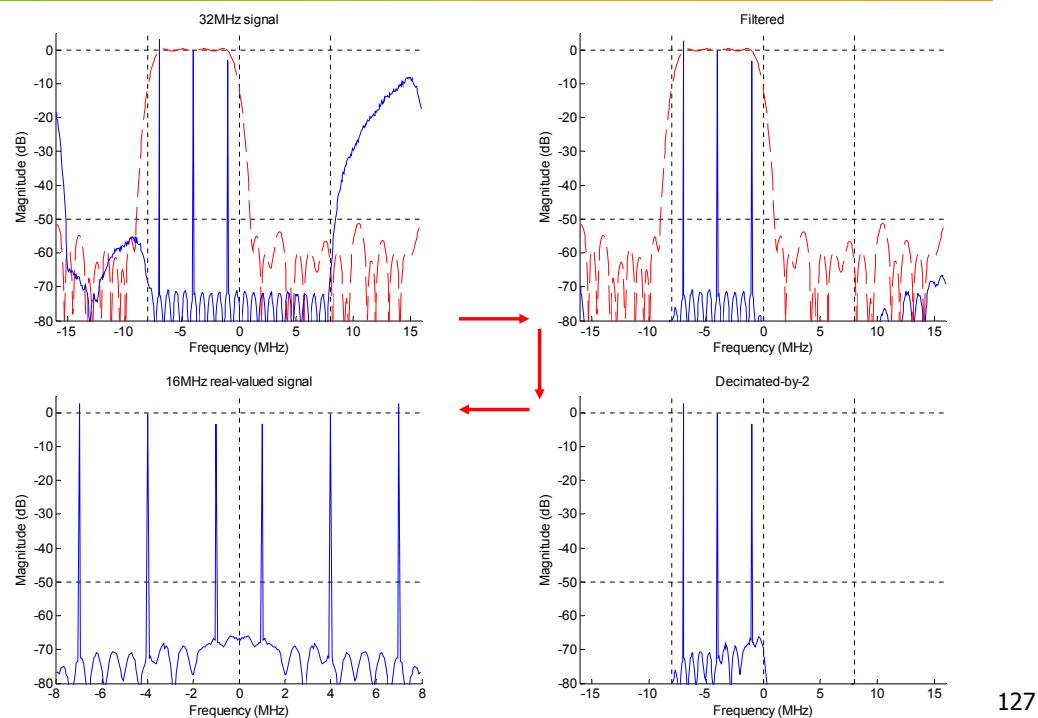
125

64MHz to 32MHz



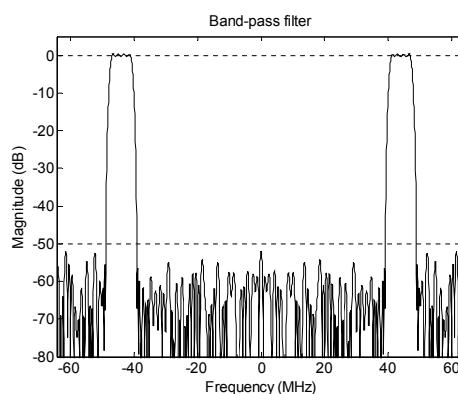
126

32MHz to 16MHz



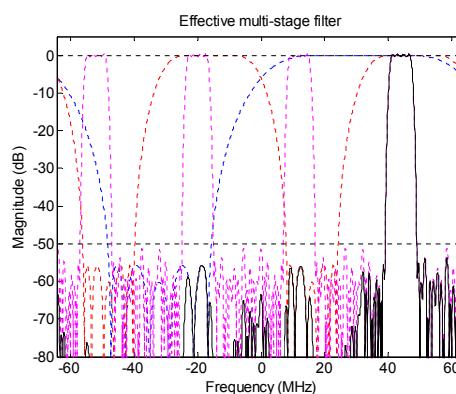
Filter comparison

Single-stage Filter



76-multipliers

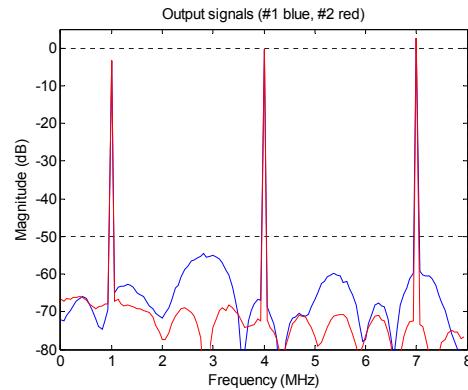
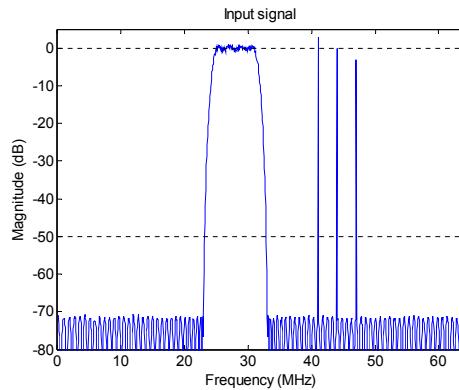
Multi-stage Filter



54-multipliers

$$54/76 = 71\%$$

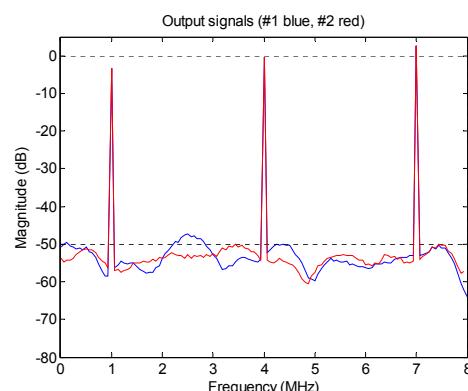
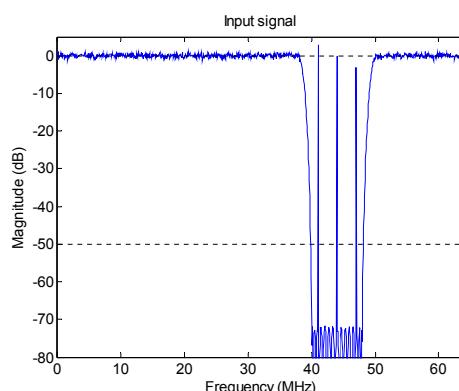
Signal comparison



- Signal;
 - 8MHz band (40MHz to 48MHz)
 - 3 sinusoid tones (41MHz, 44MHz, 47MHz)
- Noise;
 - Band-pass filtered (8MHz wide, 24MHz to 32MHz)

129

Out-of-band noise in all bands



- Signal;
 - 8MHz band (40MHz to 48MHz)
 - 3 sinusoid tones (41MHz, 44MHz, 47MHz)
- Noise;
 - Band-stop filtered (40MHz to 48MHz kept clear of noise)

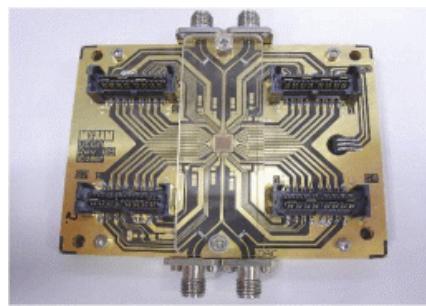
130

Wideband ADC Prototyping

Altera Stratix IV GX Development Kit



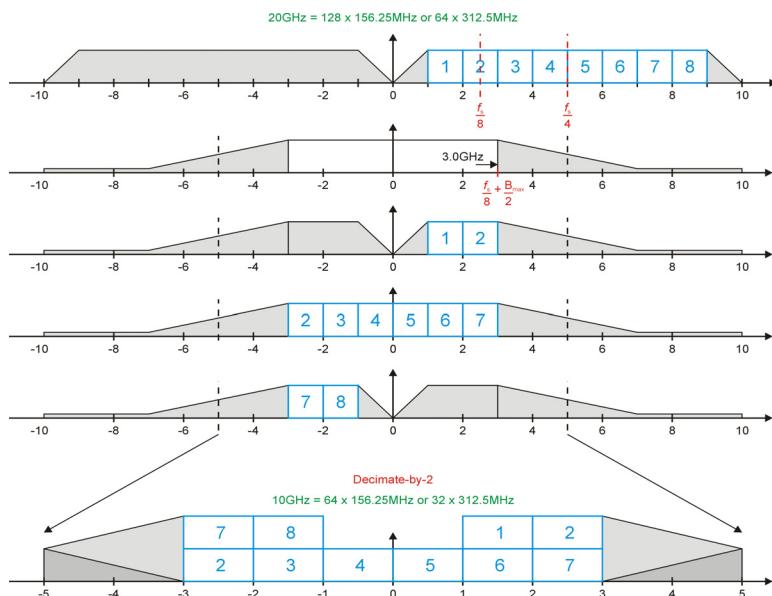
Micram 6-bit 20GSps VEGA ADC Module



(This is arriving next week! 5/9/11)

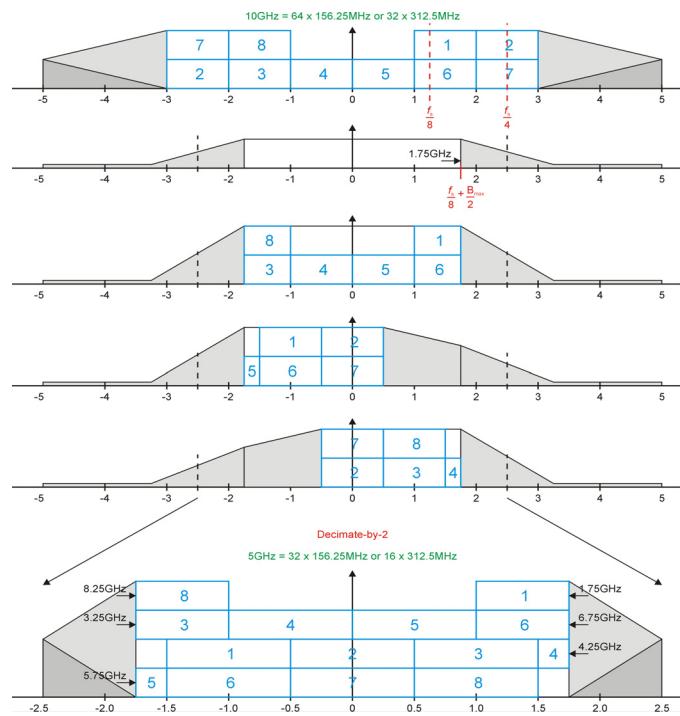
131

Multiplier-free Downconversion



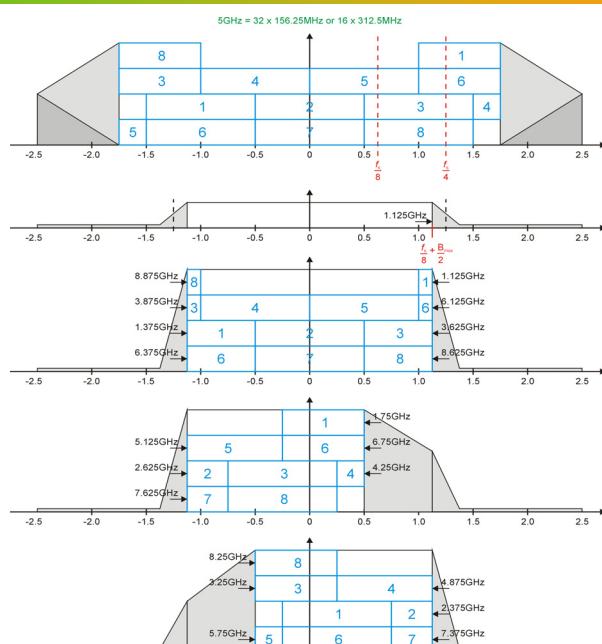
132

Multiplier-free Downconversion



133

Multiplier-free Downconversion



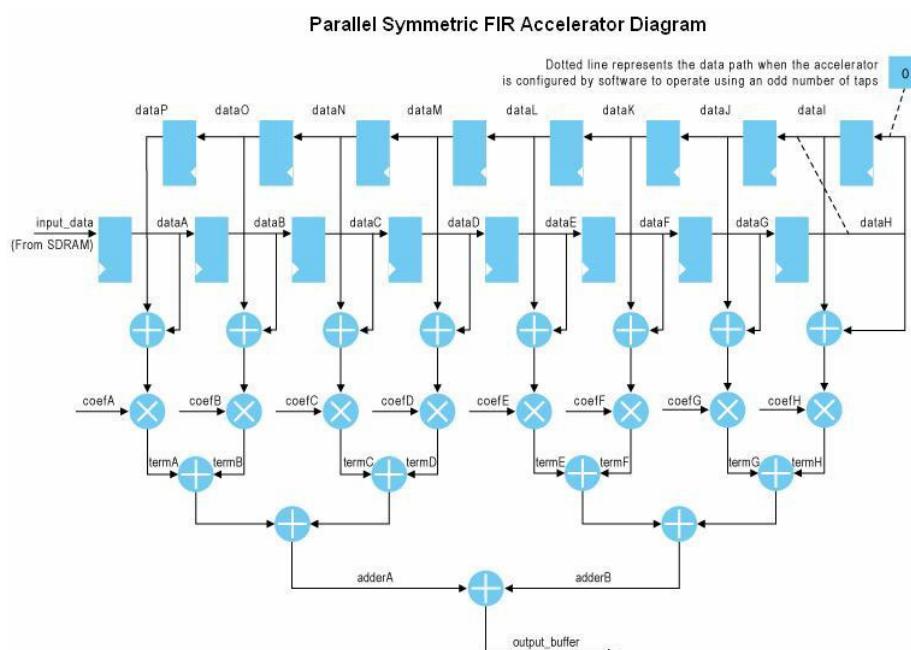
134

Detailed FIR filter analysis

- BeMicro-SDK FIR example:
 - Sample rate of 10MHz
 - Low-pass with cutoff of 1.25MHz
 - 15-coefficient FIR filter
 - Windowed design method
 - 12-bit input (Q0.11)
 - 14-bit coefficients (Q0.13)
 - 16-bit output (Q5.10)

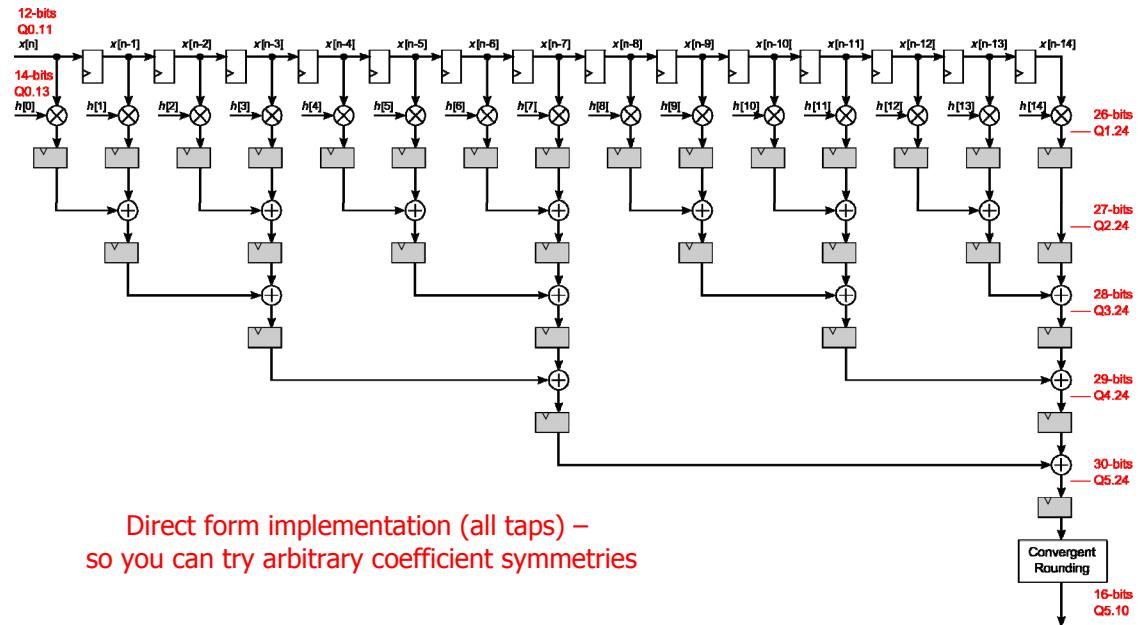
135

BeMicro-SDK FIR Example



136

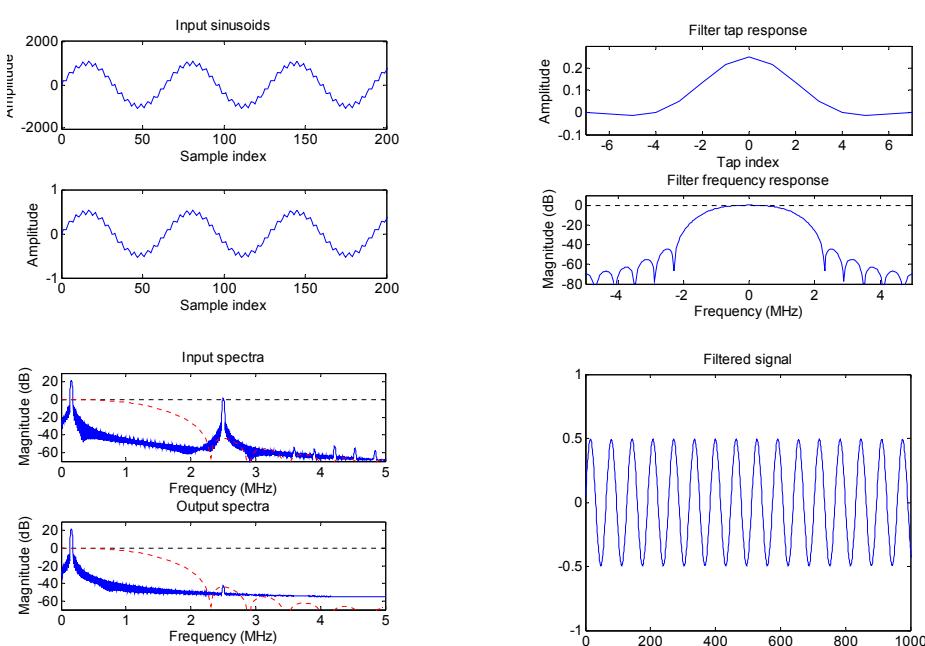
BeMicro-SDK FIR Example



Direct form implementation (all taps) –
so you can try arbitrary coefficient symmetries

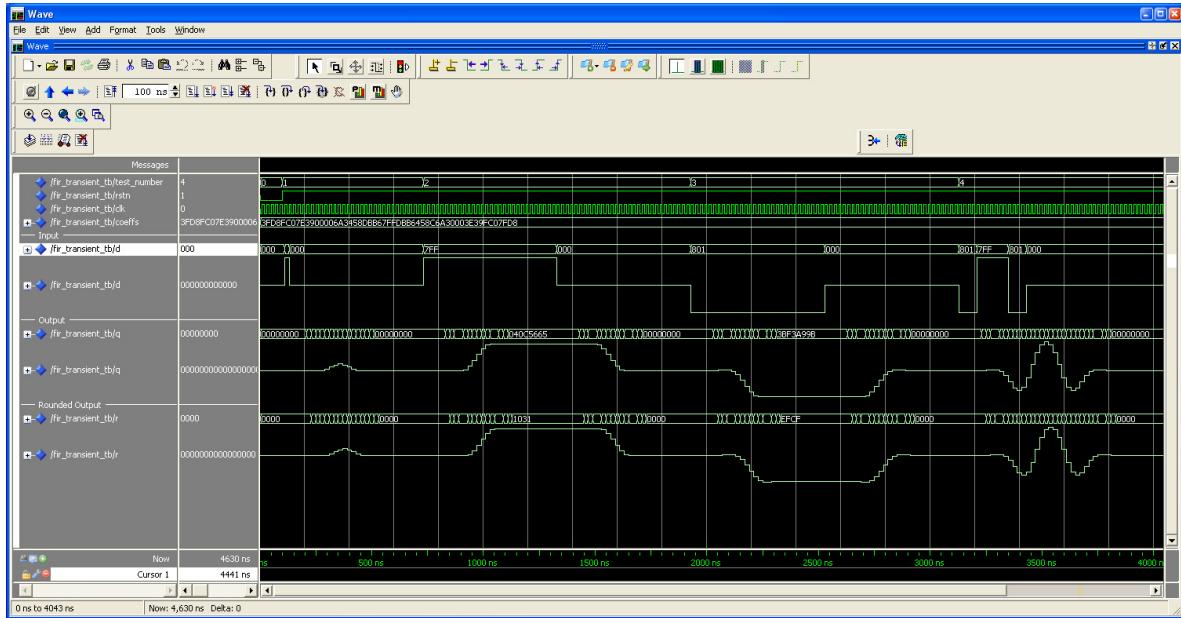
137

BeMicro-SDK FIR Example



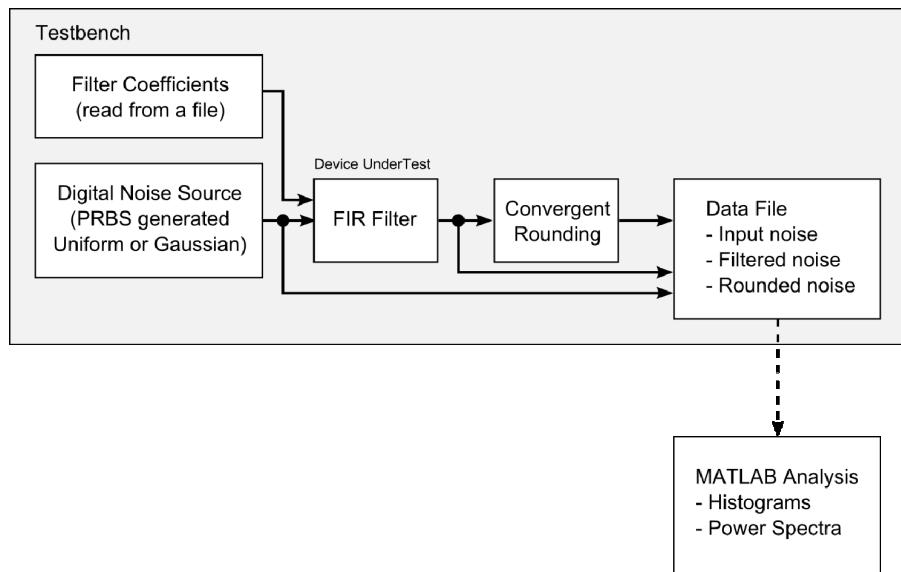
138

FIR Transient Response Testbench



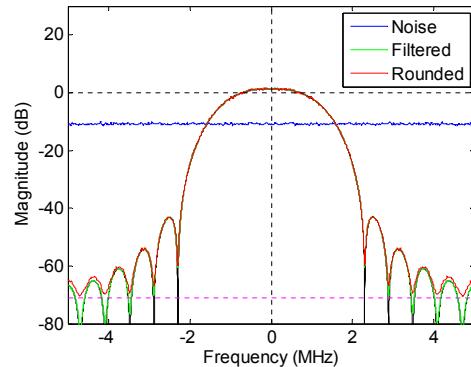
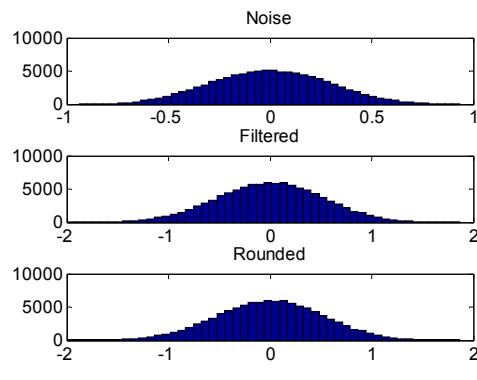
139

FIR Filtered Noise Testbench



140

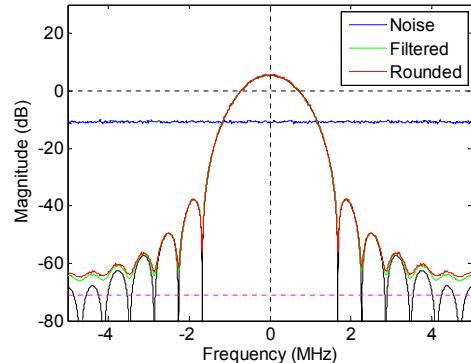
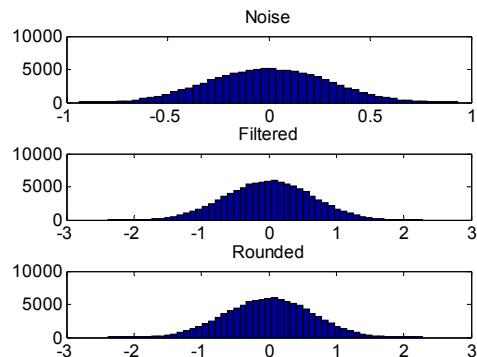
FIR Filtered Noise



- Cut-off frequency of 1.25MHz (2.5MHz bandwidth)
- Output passband at 10MHz/2.5MHz above the input signal, at 1.3dB

141

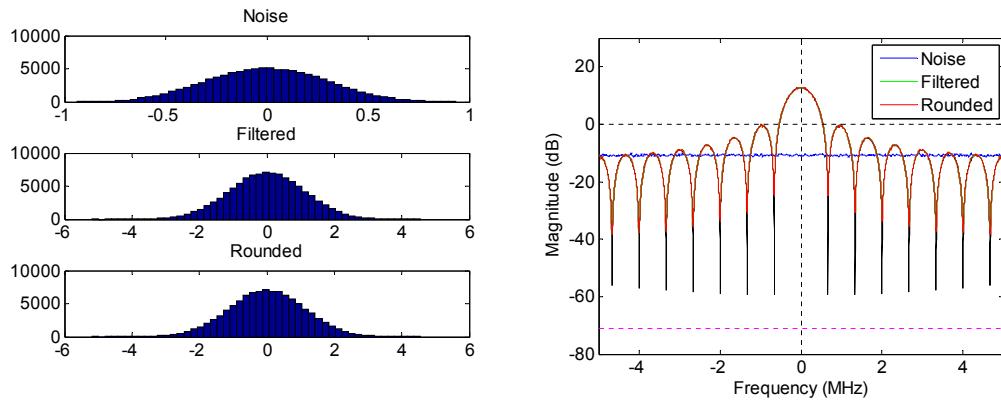
FIR Filtered Noise



- Cut-off frequency of 625kHz (1.25MHz/2)
- Output passband at 10MHz/1.25MHz above the input signal, at 7.3dB

142

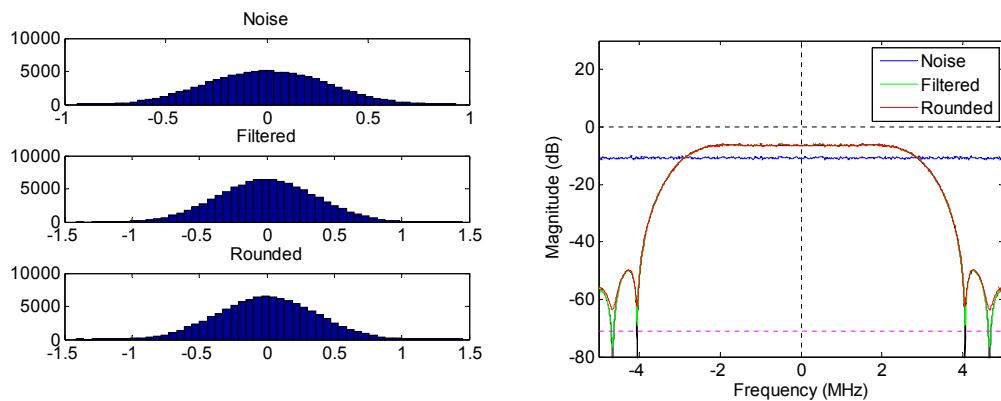
FIR Filtered Noise



- All coefficients set to full-scale (fractional integer 1.0)
- Results in a moving average, without the divide-by-15, so the passband is $20\log_{10}(15) = 23.5$ higher than the input, at 12.7dB

143

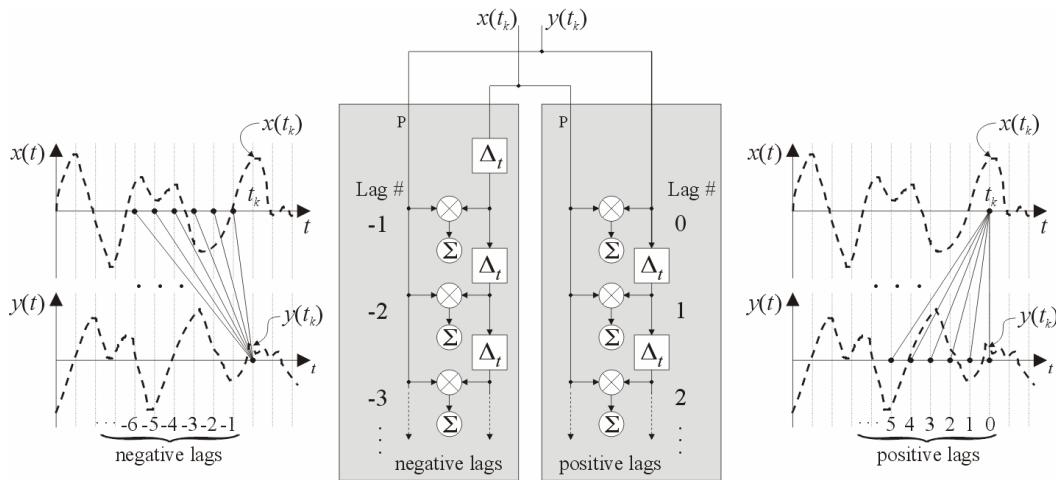
FIR Filtered Noise



- Cut-off frequency of 3MHz
- Output passband at 10MHz/6MHz above the input signal, at 6.4dB

144

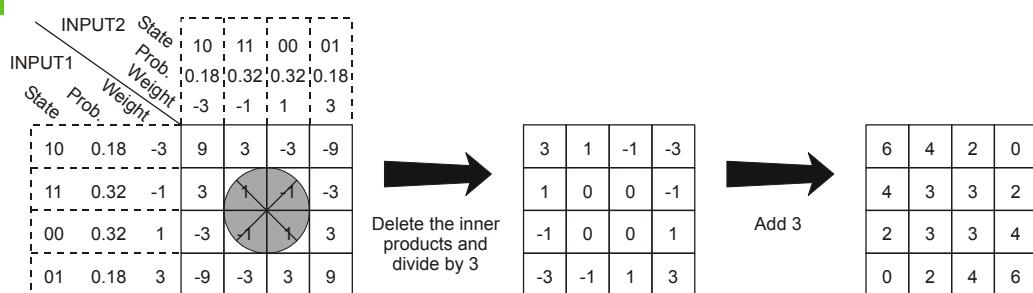
Correlation Processing



Looks similar to an FIR filter; except that two signals are being multiplied-and-averaged, rather than a signal and tap coefficients.

145

Correlation Processing



- 2-bit sampling (or filter output quantization)
- 2-bit by 2-bit multiplication
 - 4-input lookup table (LUT)
 - Nominal 4-bit multiplication result
 - Deleted inner product has 3-bit result
(25% logic reduction for 1% loss in SNR)

146

Summary

- Learned to speak 'FPGA-DSP'
- Seen what FPGAs are, and what they can be used for
- Lowered the intimidation level of using FPGA-based DSP

147

The End

Please fill out the evaluation forms, thanks!

Additional resources can be found at:
<http://www.ovro.caltech.edu/~dwh>

148