FPGA Based Digital Signal Processing – Applications & Techniques

Nathan Eddy
Fermilab
BIW12 Tutorial

Outline

- Digital Signal Processing Basics
- Modern FPGA Overview
- Instrumentation Examples

Advantages of Digital Signal Processing (DSP)

- NO DRIFT – due to temperature or age
- ACCURACY – defined by number of bits
- PREDICTABILITY – from simulation
- PERFORMANCE
  - Linear Phase Response possible
  - Adaptability in terms of resources
- PRODUCTION – identical units, no tuning
- FLEXIBILITY – via firmware modifications
- DYNAMIC RANGE

For Beam Instrumentation

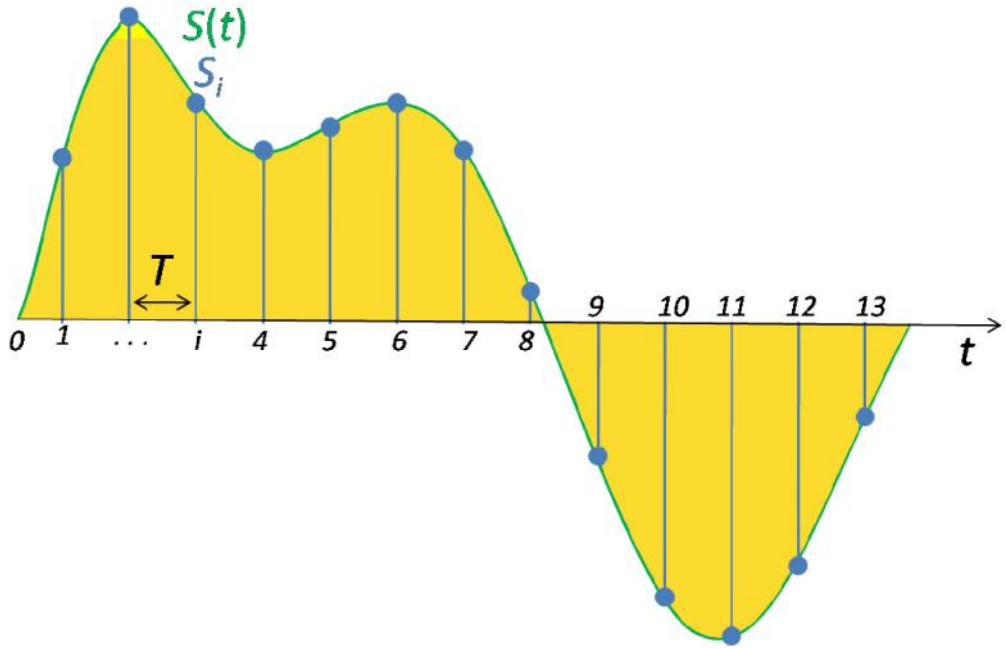
- Need to work with analog input signals
  - Beam pickups, Schottky detectors, Torroids, etc
  - Requires Analog to Digital Converters (ADCs)
- Need to produce analog output signals
  - To act on the beam – RF, kick signals, etc
  - Require Digital to Analog Converters (DACs)
- The effectiveness of FPGA solutions is largely dominated by the performance of the converters

Practical Limitations of Sampling

- Common sources of sampling error
  - Aliasing
  - Quantization
  - Sample Clock Jitter
- Can be characterized by their impact on the Signal to Noise Ratio (***SNR***) of the sampled signal
  - $\boldsymbol{SNR \sim log(S_a/N_a)}$
  - Typically expressed in decibels (***db***)

Discrete Time Sampling

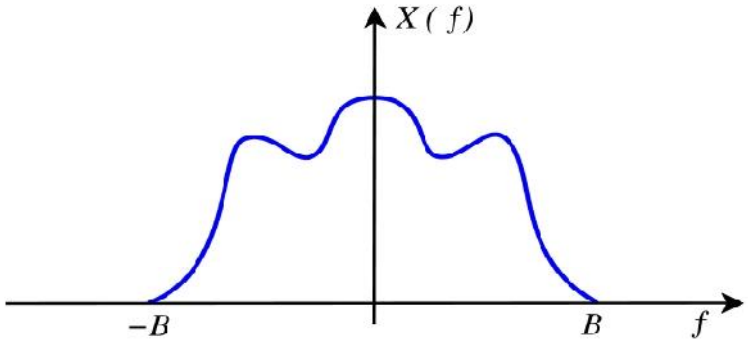
- $S_i = S(t) * \delta'(t)$ where $\delta'(t) = \Sigma\ \delta(t-nT)$



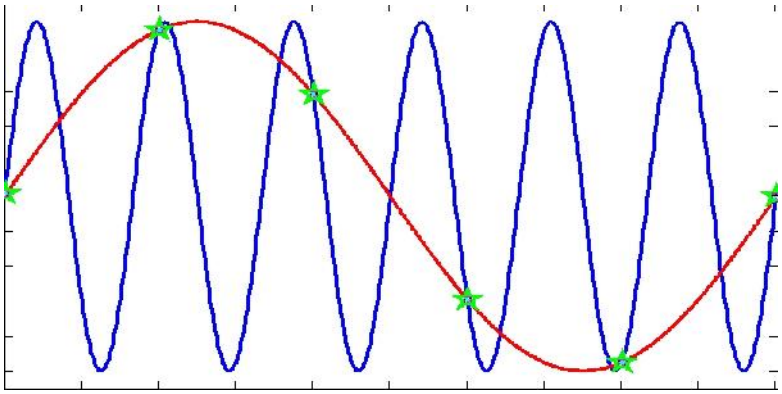
- The sequence S_i is the sampled version of $S(t)$
- The sampling frequency F_s is $1/T$

Sampling Theorem

- For ideal reconstruction $F_s > 2B$ where B is the highest frequency in the signal of interest

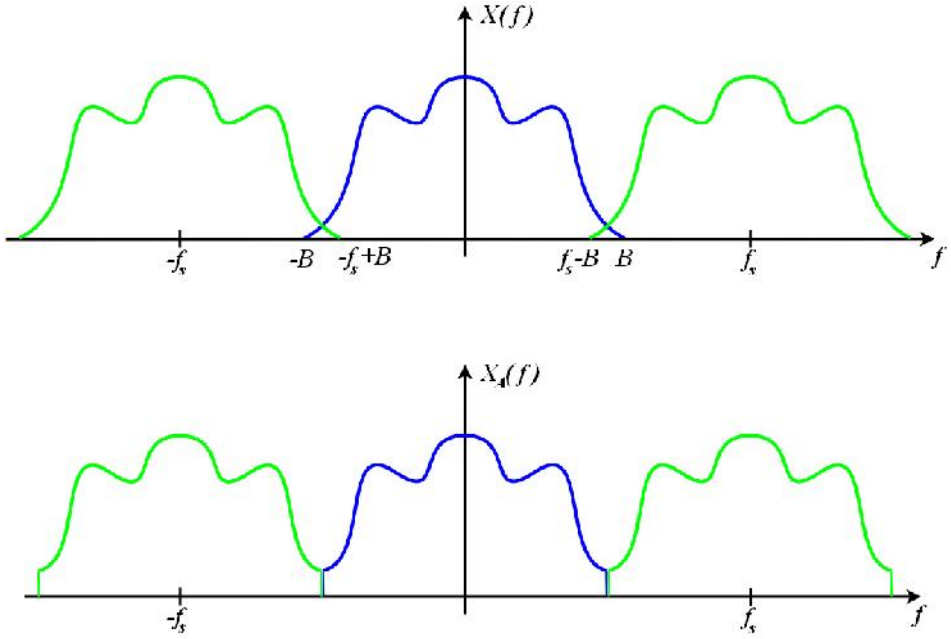


- Sampling ambiguity



Aliasing

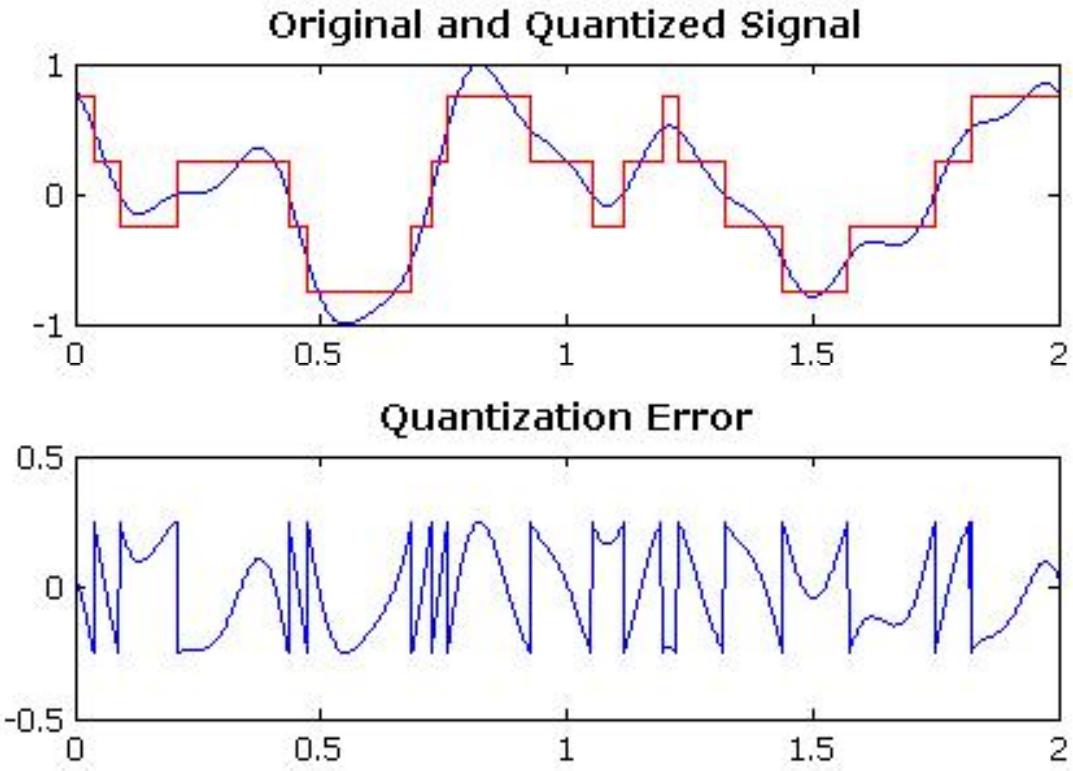
- Signals at frequencies larger $F_s/2$ than will "alias" into the first Nyquist band



- Undersampling makes use of this effect

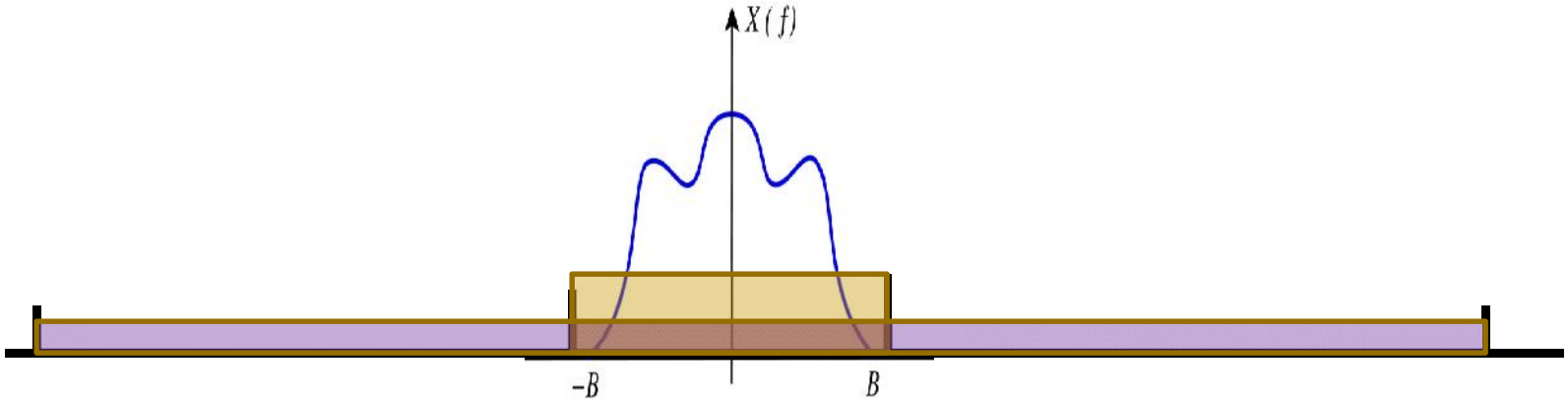
Quantization

- Results in signal noise



- $SNR_{db} = 1.76 + 6.02N$, N=number of bits

Oversampling Improves SNR



- Gain $SNR_{db} = 10log(R)$ where $R=F_s/2B$
  - F_s is the sampling Frequency
  - B is signal bandwidth
- Also relaxes the requirements on the anti-aliasing analog filter

Sampling Clock Jitter

- Another source of signal noise
  - Proportional to maximum signal frequency, f_{max}
  - RMS sampling clock jitter, T_a

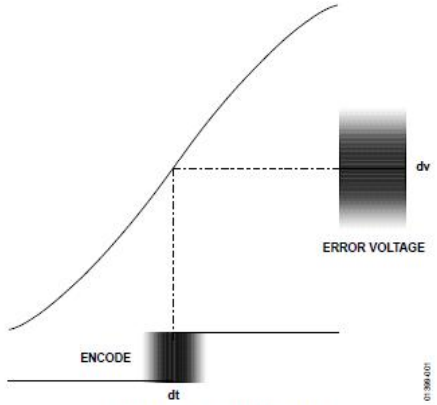
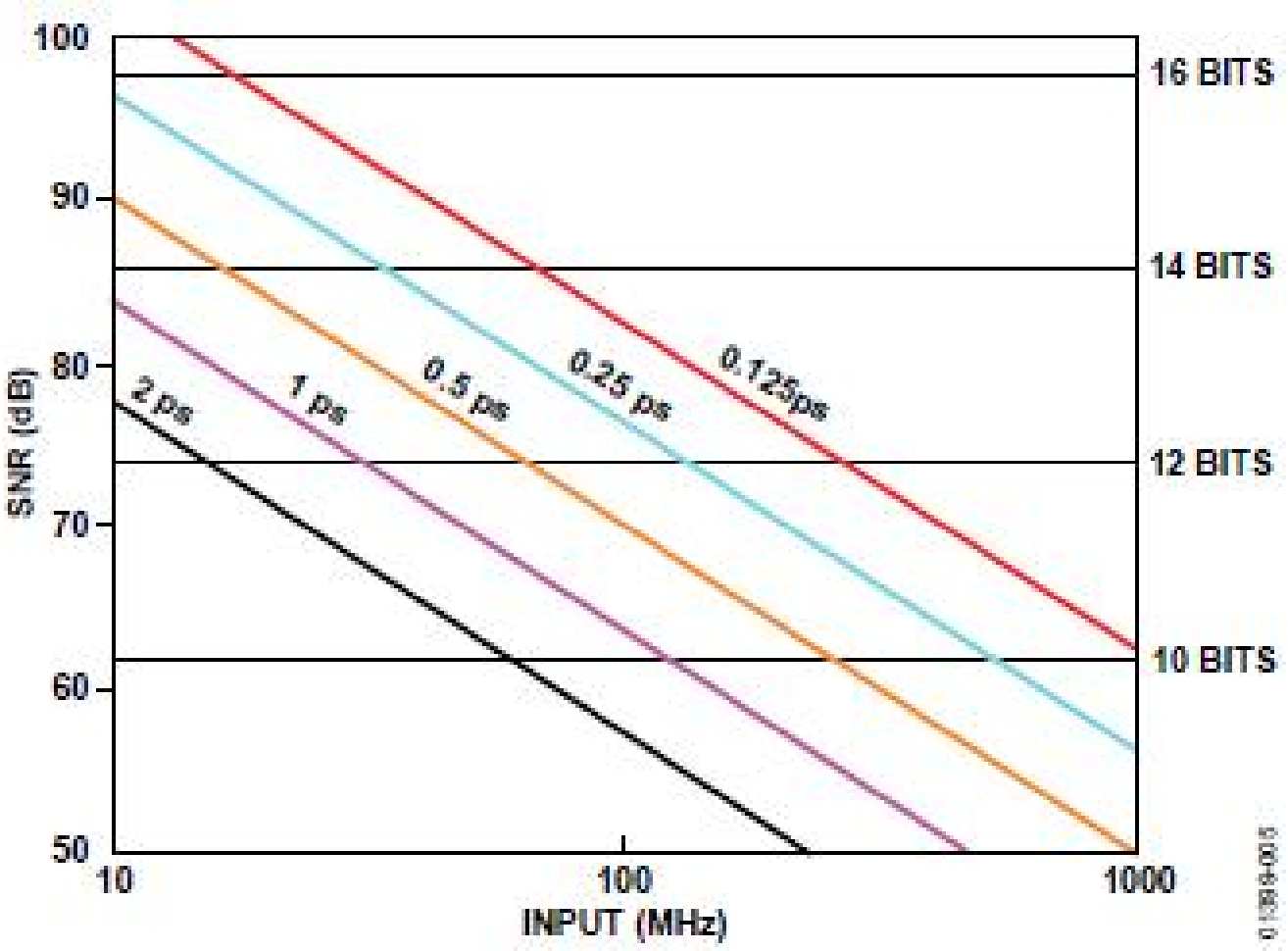


Figure 1. RMS Jitter vs. RMS Noise

- $SNR_{db} = 20log(1/(2\pi f_{max} T_a))$
- A significant effect for undersampling applications

Theoretical SNR Comparison



The Z domain

- A sampled sequence, ***x(n)*** can be represented

$$X(z) = \sum_{n=0}^{\infty} x(n)z^{-n}$$

- Where z^{-1} is the unit delay related to the sample period (T)
- ***X(z)*** is the z-transform of ***x(n)***

Difference Eq & Transfer Function

- A constant coefficient difference equation is a recursive relationship where-by the output of a discrete time system can be calculated using a combination of past output values and past and present input values

$$\sum_{k=0}^{N} a_k y[n-k] = \sum_{k=0}^{M} b_k x[n-k]$$

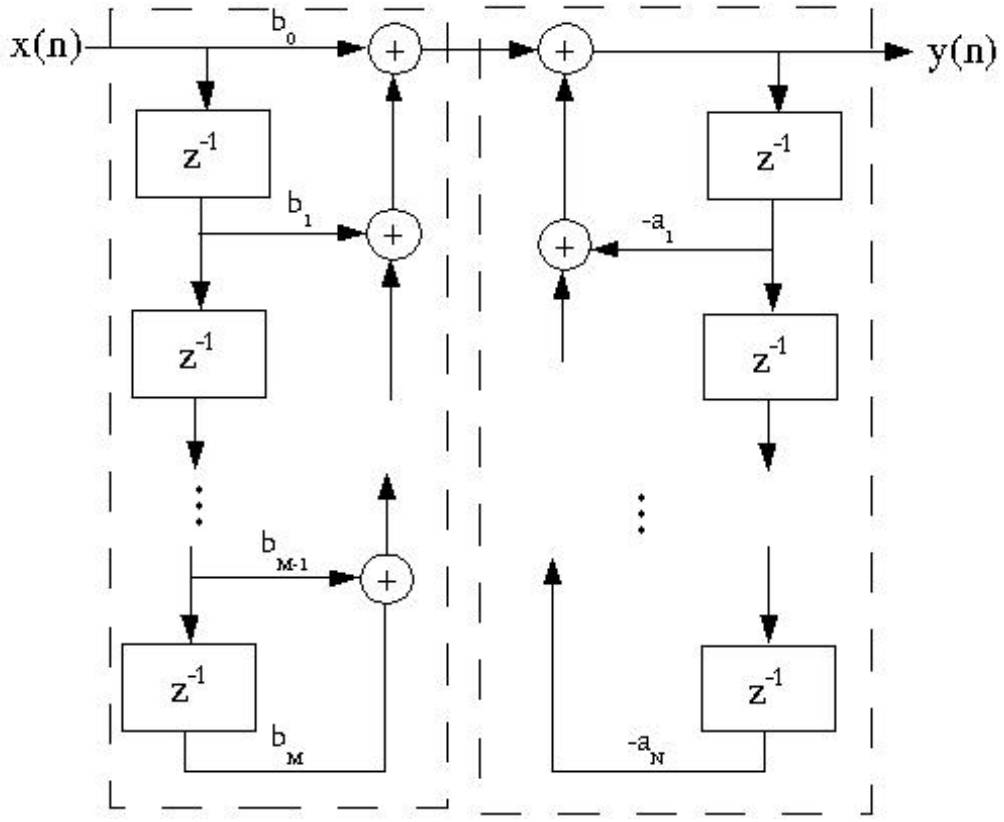
- Gives z-domain transfer function, ***H(z)***

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{N} b_k z^{-k}}{\sum_{k=0}^{M} a_k z^{-k}}$$

- Digital filters are specified by ***a,b*** coefficients

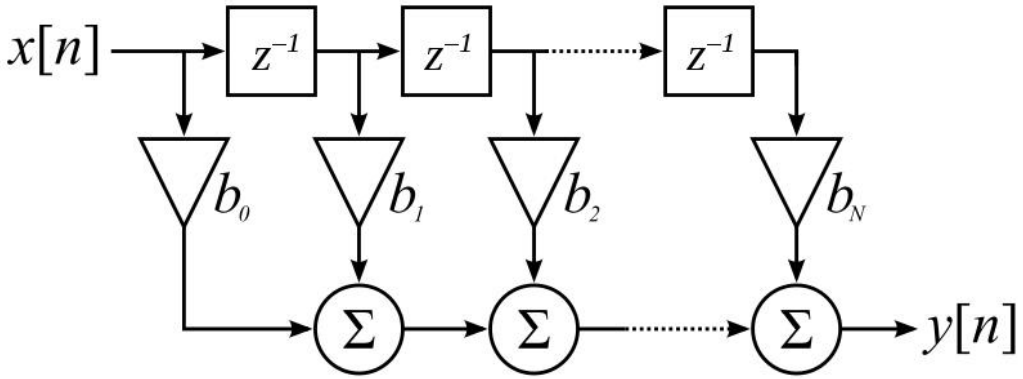
Infinite Impulse Response (IIR) Filter

- Feedback provides infinite impulse response
  - a_n non-zero for $n \geq 1$
- Properties
  - Very efficient
  - Non-linear Phase
  - Can be unstable



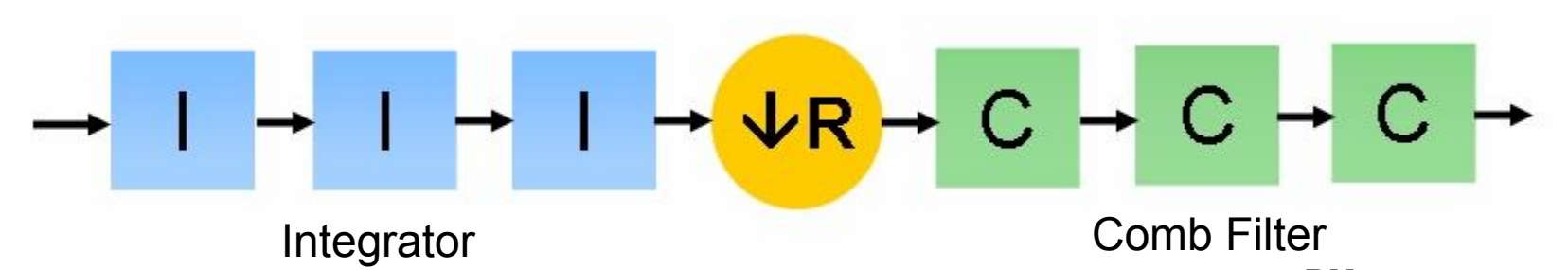
Finite Impulse Response (FIR) Filter

- Nth Order FIR filter



- Just the feedforward block
  - a_0=1, all others zero
- Has linear phase if coefficients are symmetric
  - That is $\{b_0,\ldots,b_N\} = \{b_N,\ldots,b_0\}$
  - No analog equivalent

Cascade Integrating Comb (CIC) Filter



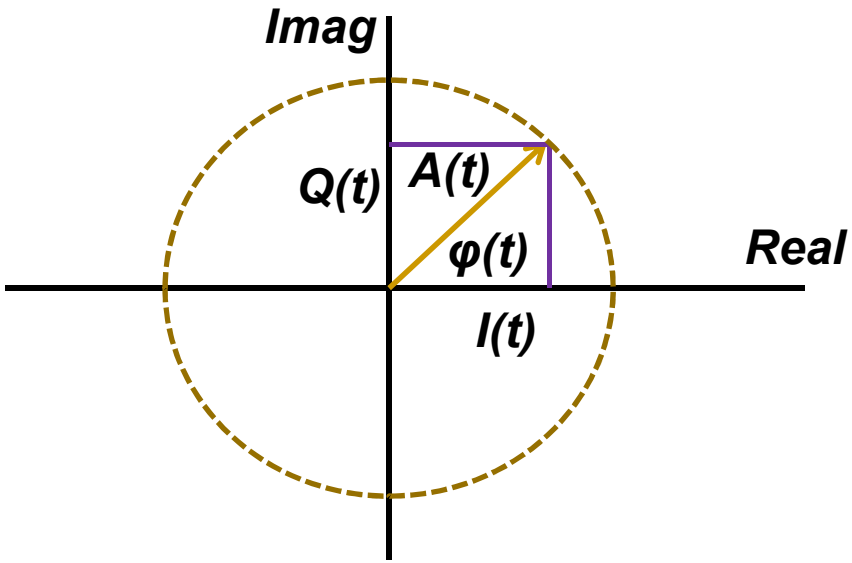
- For an Nth Order CIC Filter, the transfer function is

$$H(z) = \frac{(1-z^{-RM})^N}{(1+z^{-1})^N} = \left(\sum_{k=0}^{RM-1} z^{-k}\right)^N$$

  - Equivalent to N FIR filters with unit coefficients -> symmetric
  - Linear Phase even though it has infinite response filter sections
- Used as very efficient way to filter and change rate
  - Can be used as Interpolation filter by reversing I & C sections

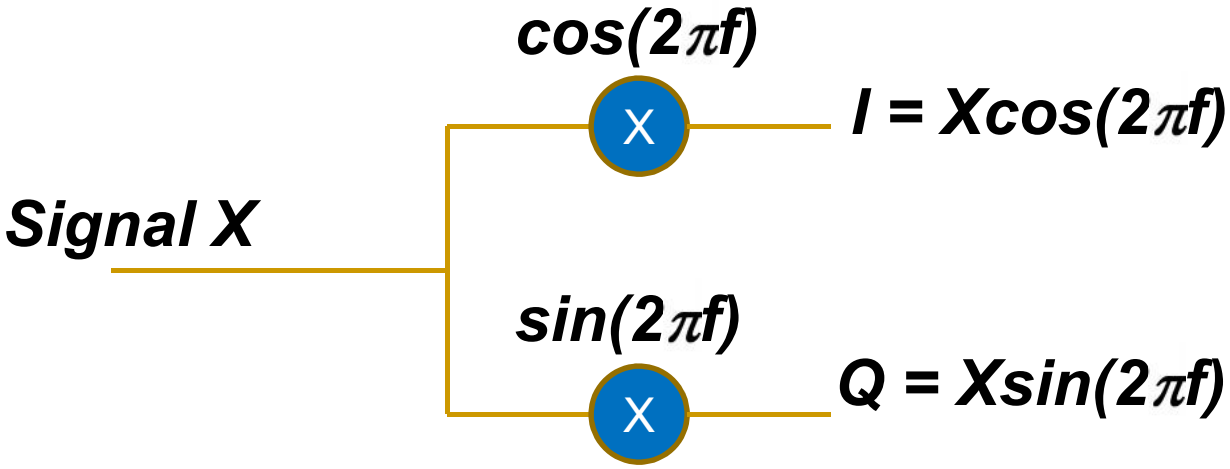
Quadrature Signals, *I* & *Q*

- Complex signal representation



- In-Phase (real) & Quadrature (imaginary)
  - ***I(t) = A(t)cos(φ(t))***
  - ***Q(t) = A(t)sin(φ(t))***

Digital Down-Conversion



- Multiply a signal ***X*** by ***sin(2πf)*** & ***cos(2πf)*** where ***f*** is the the down-conversion frequency
  - Shifts the spectral components of **X** down in frequency by ***f***
- Commonly followed by decimating low pass filter
  - Remove sum frequency components
  - Process narrow bandwidth signals at lower rate in baseband

Discrete Fourier Transform (DFT)

- Discrete Fourier Transform of length N

$$X(k) = \sum_{n=0}^{N-1} x(n)\,[\cos(2\pi nk/N) + i\sin(2\pi nk/N)]$$

- ***X(k)*** determines the complex signal contribution of the frequency in the composition of ***x(n)***
- Phase or direction (forward, inverse) given by the sign of the imaginary term

Fast Fourier Transform (FFT)

- Works as a bank of band-pass filters
  - The output magnitude from each filter is proportional to the input energy in each band
- An ***N*** point DFT requires N^2 operations
- Can compute the ***N*** point DFT as two ***N/2*** point DFTs
  - Extrapolate to the limit where ***N*** is a power of 2 for an ***N*** point FFT
- An ***N*** point FFT requires $N/2\log_2(N)$ operations
- Note, FFT are almost always implemented as a power of 2 but can be any prime factor

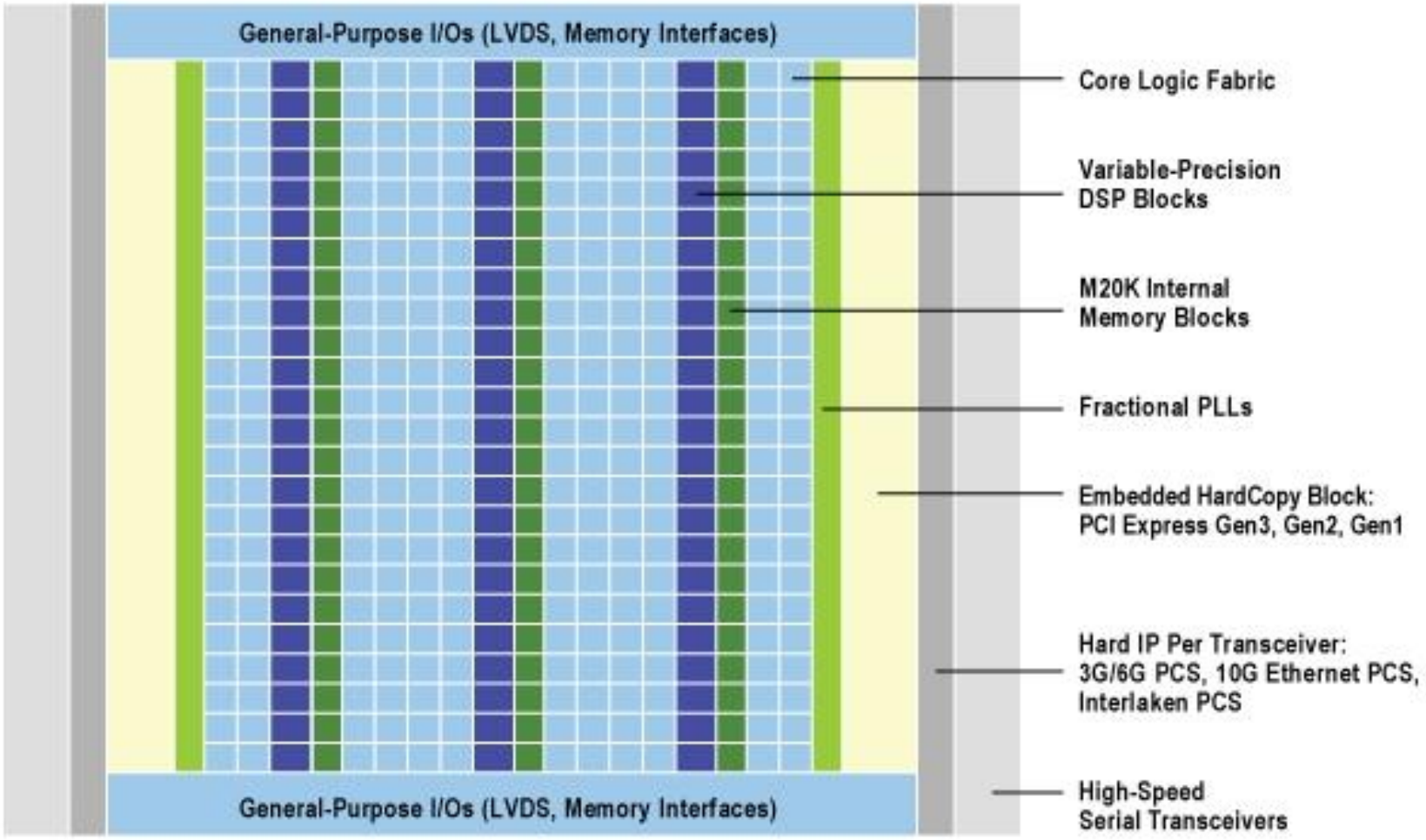
Outline

- Digital Signal Processing Basics
- Modern FPGA Overview
- Instrumentation Examples

FPGA Advantages for DSP

- Parallel processing power (Speed)
  - Able to consume large quantities of data
  - Alleviate bandwidth bottlenecks in front-ends
- Pipeline architecture
  - Efficient digital filter implementation
  - Perform operations under strict timing (latency) control
- Dedicated DSP (multiplier) blocks
- Flexibility
  - Clocking synchronous with accelerator
  - Easy integration with ADCs and DACs
  - Easy to modify algorithms and functionality

High End FPGA Architecture

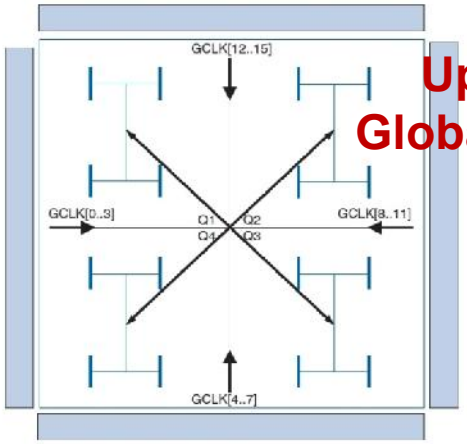


Flexible I/O Options

- Dedicated hardware support for almost all standard I/O protocols
- Transceivers with equalization for fast differential serial I/O
  - PCI Express, RapidIO, etc
  - Support for over 20GBps links!
- External memory interface support
  - DDR, DDR2, DDR3
- Speed and quantity of available I/O -> $$

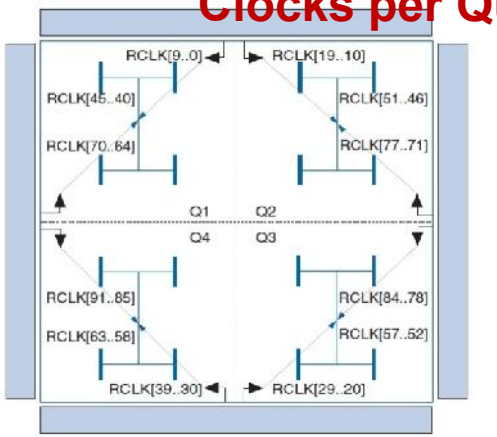
Clock Domains and Distribution Networks

- Multiple PLLs and DLLs for sophisticated clock management
  - Support for run-time reconfiguration
  - Fractional PLLs for arbitrary clock synthesis
- Dedicated global and regional clock distribution networks
  - Support high rate synchronous design
  - Support parallel & pipelined design



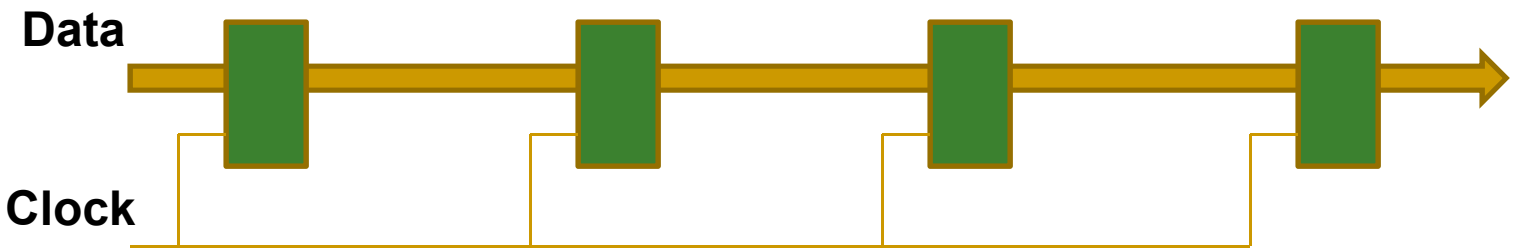
Up to 16 Global Clocks

Up to 23 Regional Clocks per Quadrant



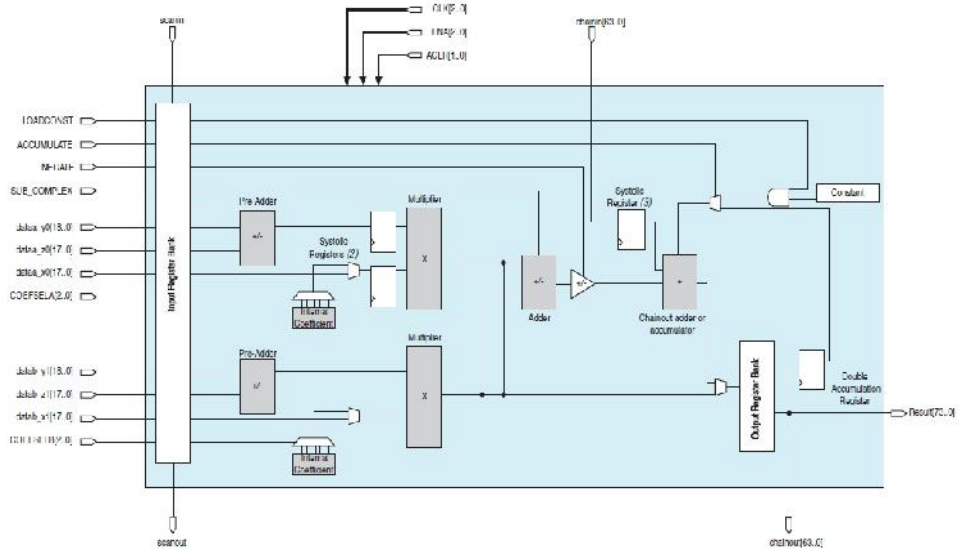
Synchronous Design - Pipelining

- Reliable designs require synchronous design principles
  - Increase speed and bandwidth
  - Trade resources and latency
- Simple Delay Pipeline



Digital Signal Processing (DSP) Blocks

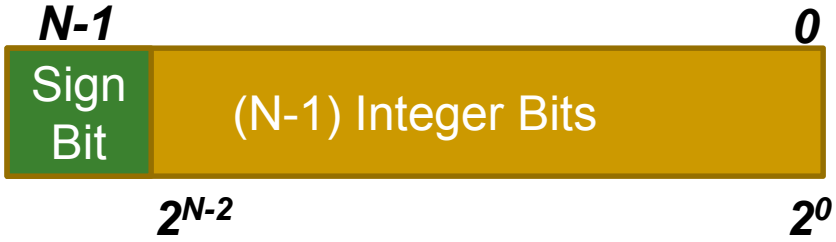
- Based upon dedicated multiplier blocks



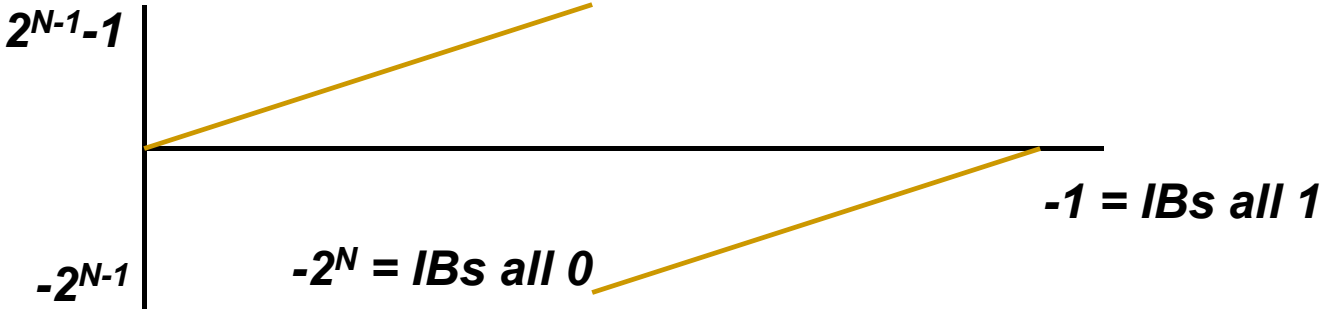
- Provide efficient flexibility and power
  - Fully customize, dedicated support for FIRs, FFT, etc to optimize for speed and efficiency
- Modern blocks provide capability for 32 or 64 bit floating point operations

Integer Representation

- 2s Complement ***N*** Bit representation

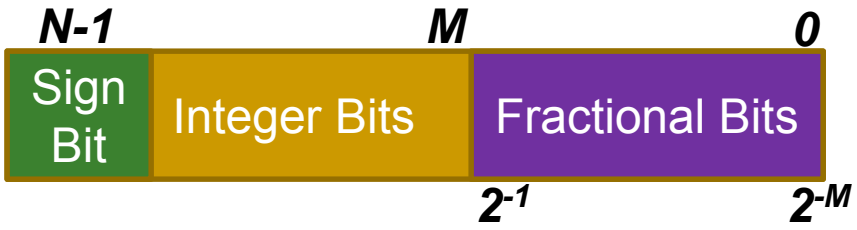


  - Positive numbers just simple binary representation
  - Negative numbers are binary number that when added to a positive number of the same magnitude equals zero
- Convenient Properties
  - Simple arithmetic – addition & subtraction the same



Fixed Point Signal Processing

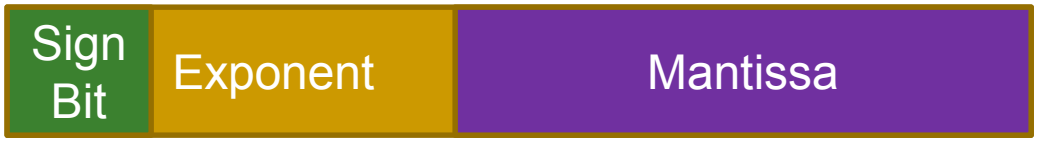
- Bit widths double on each multiply!
- Need to control bit widths for efficient resource use
  - Truncation or Rounding
  - Saturation or Roll-over
- Fixed Point N Bit Representation



  - Can facilitate keeping track of bit widths

Floating Point Representation

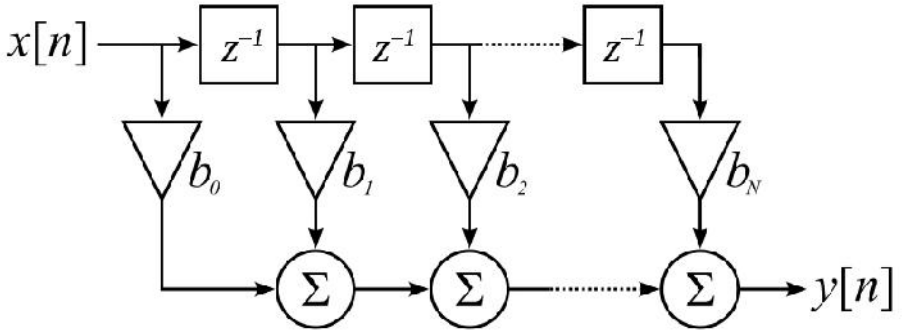
- IEEE-754 Standard Followed



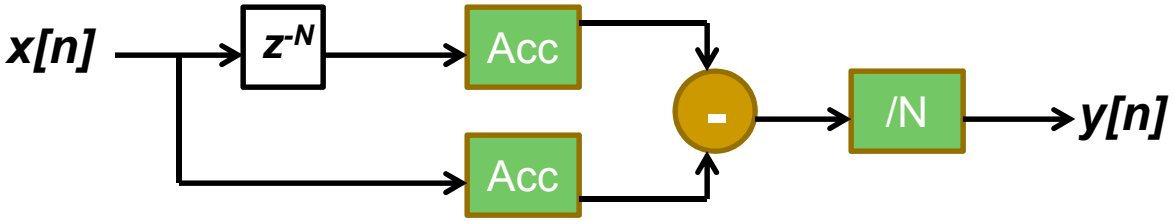
- 32 bit – Single Precision
  - 8 bit Exponent & 23 bit Mantissa
- 64 bit – Double Precision
  - 11 bit Exponent & 52 bit Mantissa
- Single-Extended Precision
  - Exponent and Mantissa widths are not fixed
  - Minimum 11 Exponent Bits (Exponent<Mantissa)
  - Minimum 31Mantissa Bits
  - Total Bits at least 43 up to 64

Running Average Example

- Can implement an ***N*** sample running average as an FIR with ***N*** taps with coefficients of ***1/N***



- Can be efficiently implemented using two accumulators and a delay line



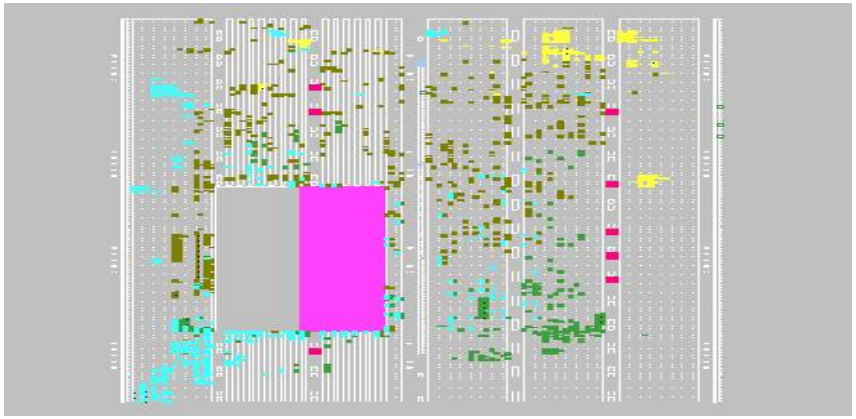
Intellectual Property (IP) Cores

- Functional cores which can be used to greatly speed up and simplify the design process
- Each FPGA manufacturer provides cores for all basic components
  - PLLs, RAM, FIFOs, Flip-Flops, etc
  - Accumulators, Add, Sub, Multiply, Divide, etc
  - Take advantage of chip resources
- Advanced cores available for almost any task
  - Simple implementation via GUI parametrization

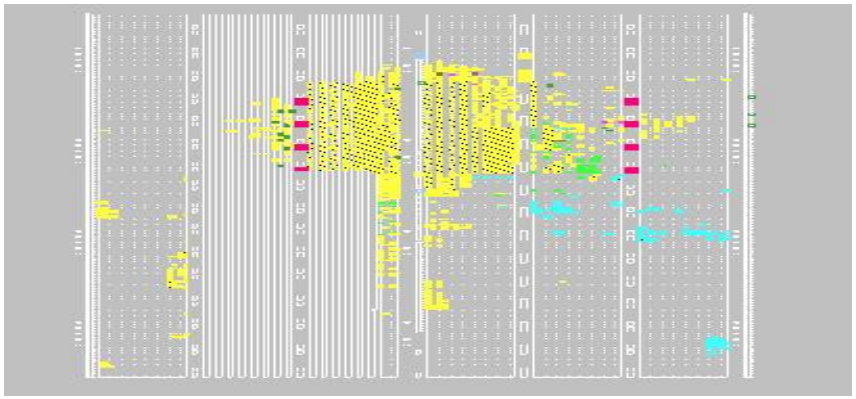
Embedded Systems

- Hard Core Embedded processor is a dedicated physical component of the chip, separate from the programmable logic
  - 2-4 times faster than Soft Core
  - More efficient if you need a processor
- Soft Core Embedded processor is built out of the programmable logic on the chip
  - A 32 bit RISC processor uses about few percent of total resources
  - Have option to re-allocate resources if processor not needed

Hard Core Resource Allocation



Soft Core Resource Allocation



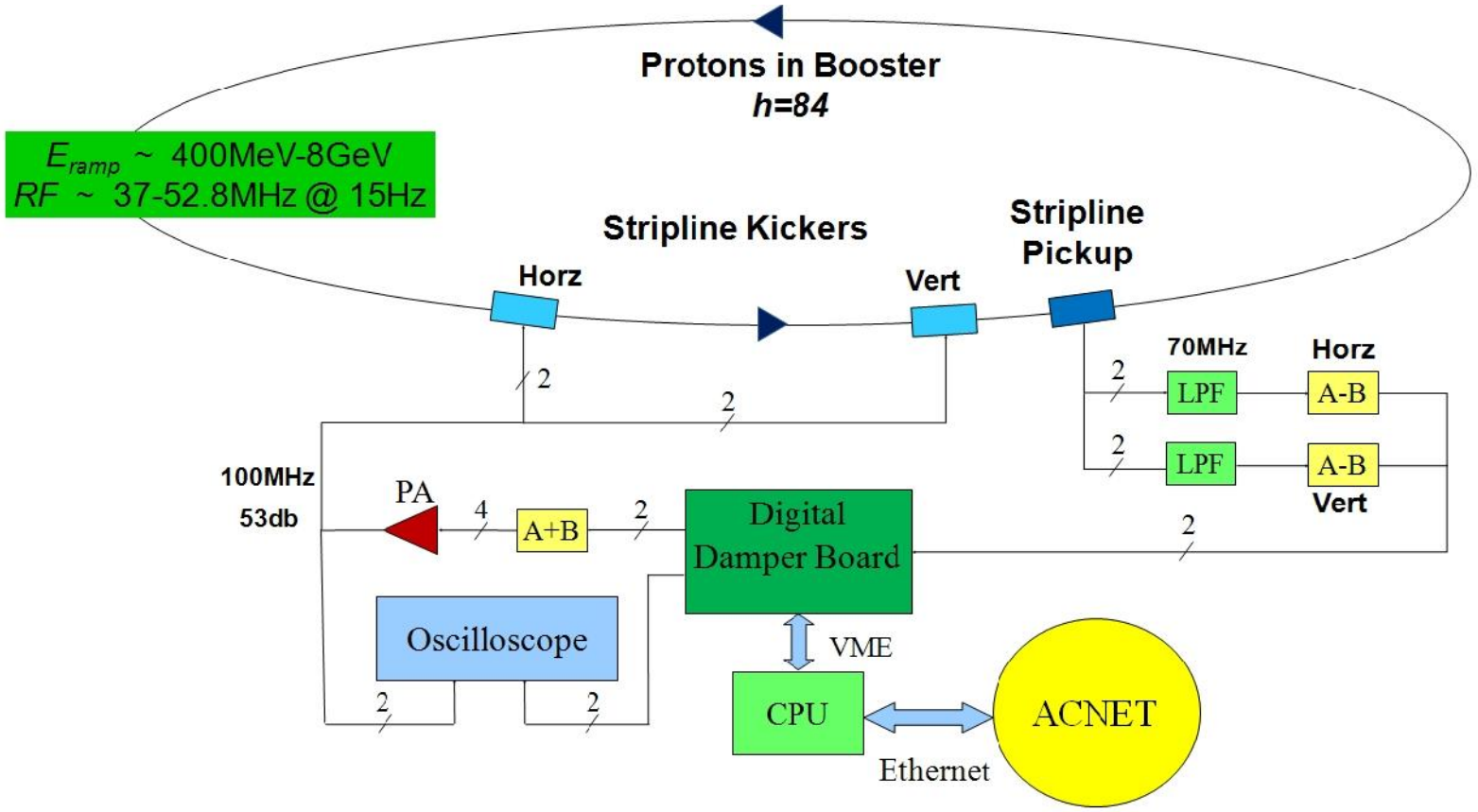
System On a Programmable Chip (SOPC)

- A wide variety of design tools and options available
  - Pure HDL entry – still possible
- Focus upon developing System On a Programmable Chip
  - Aim to simplify the design process as chip architectures become more complicated
- Tightly couple design and simulation at all levels

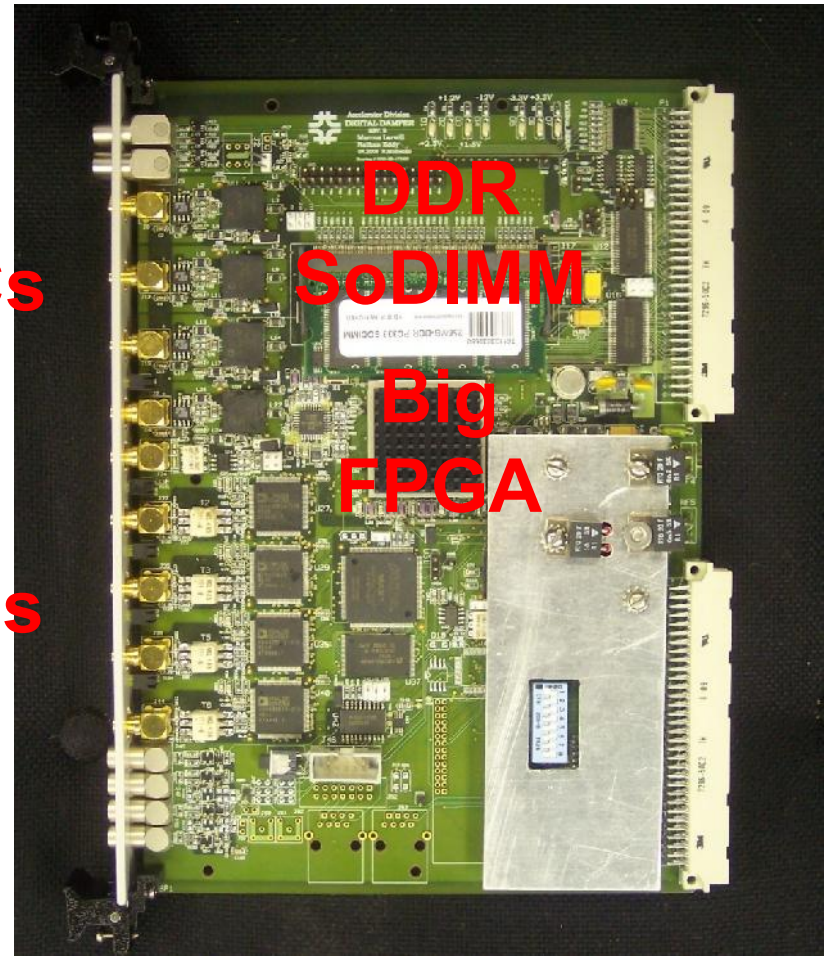
Outline

- Digital Signal Processing Basics
- Modern FPGA Overview
- Instrumentation Examples

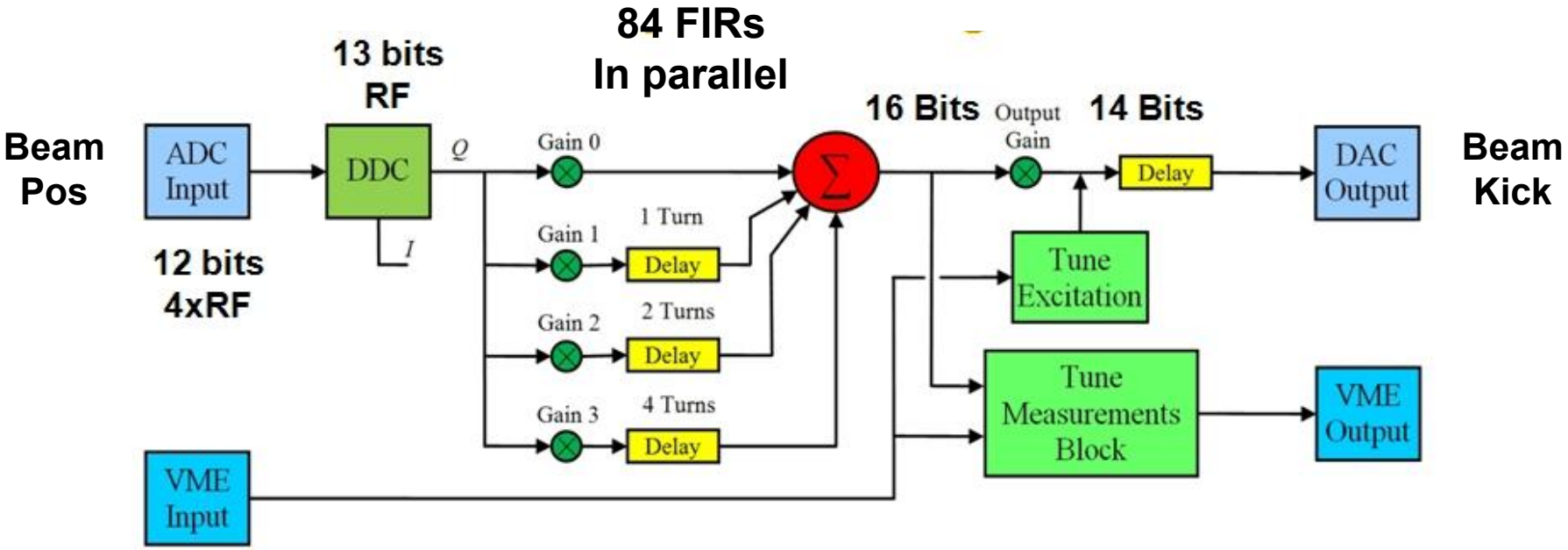
Fermilab Booster Digital Damper



Digital Damper Board

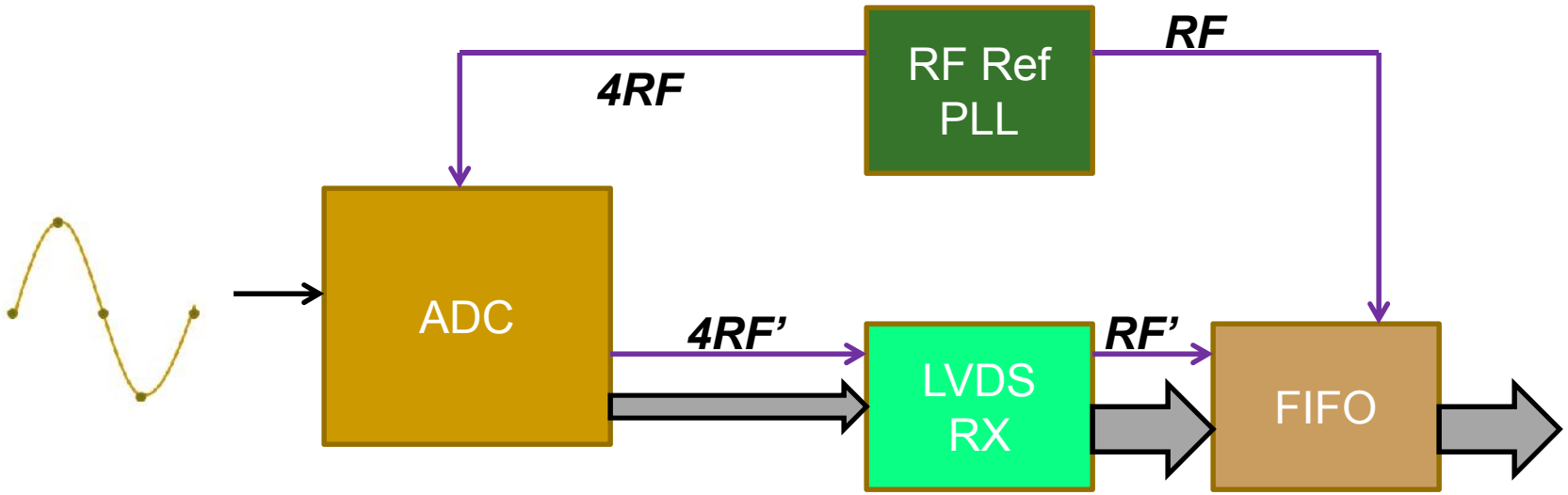


Firmware Overview



- Design operates synchronous with Booster RF sweeping from 37MHz to 53MHz
- Provides bucket by bucket damping on all 84 Booster bunches

ADC Input clocked at 4*RF

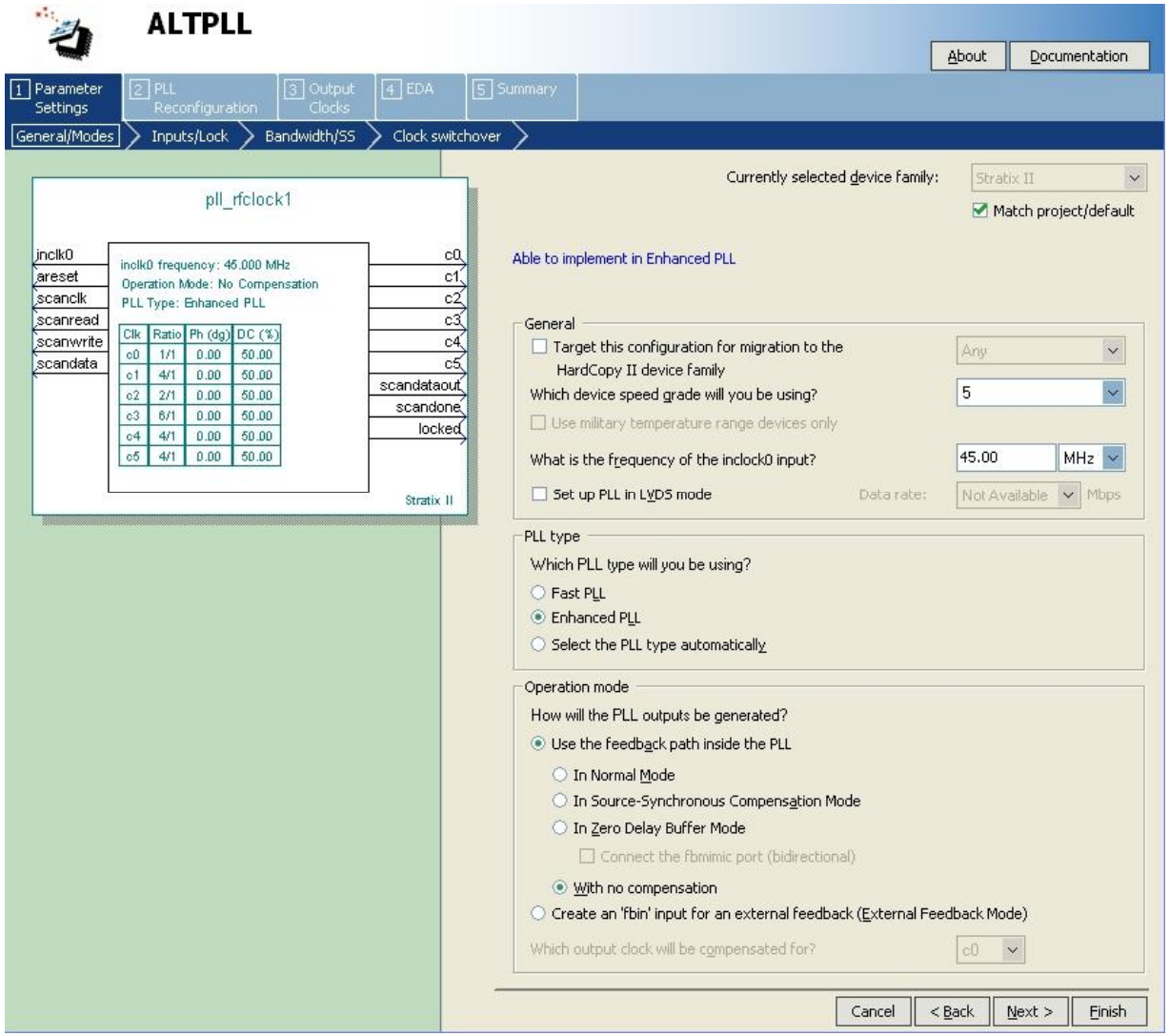


- ADC clock is shifted to provide ideal sampling
- FIFO used to shift data from ADC clock to internal clock

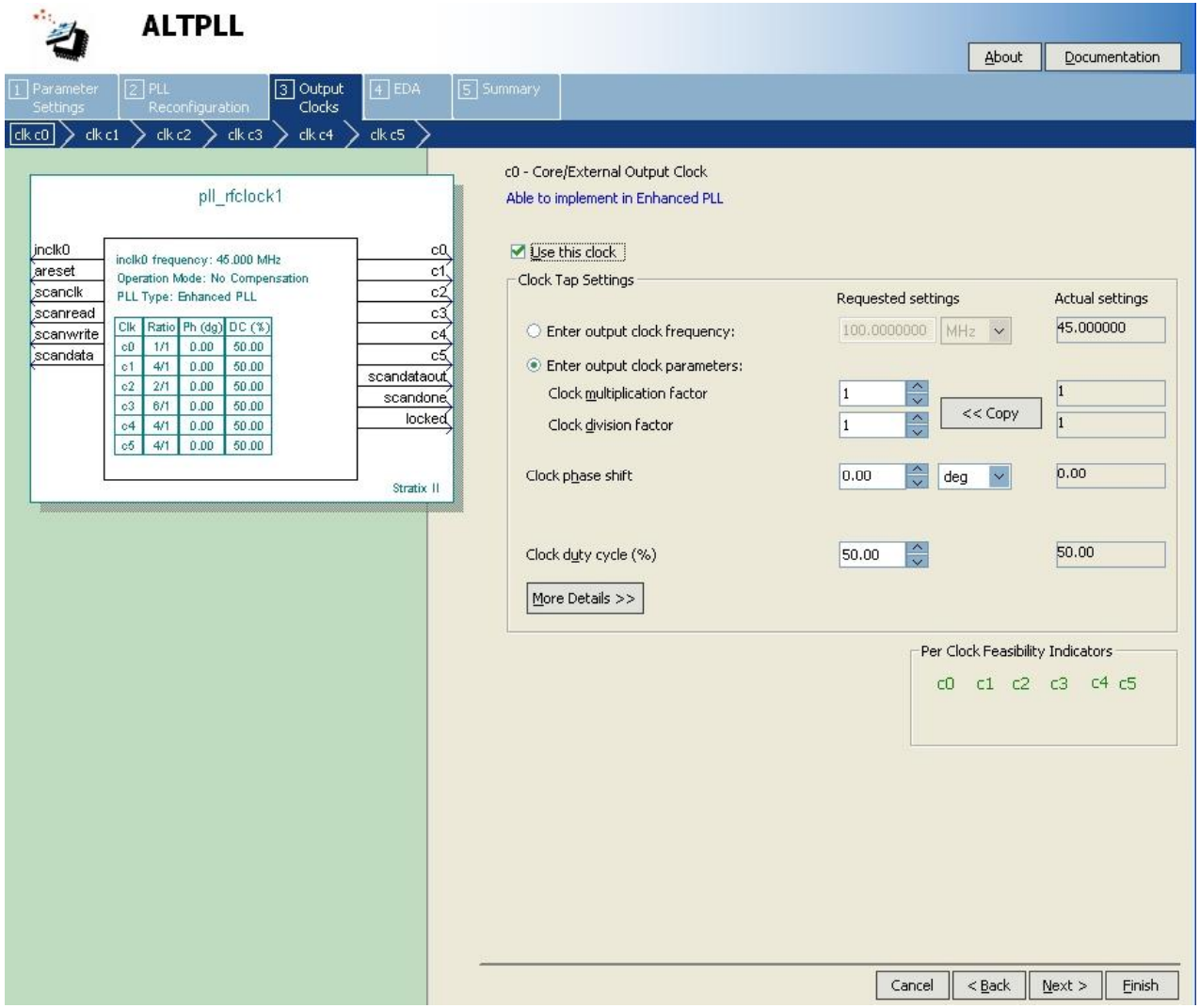
RF Reference PLL

- Use one of the Enhanced PLLs
  - Dedicated clock pin for RF Reference
  - Directly connected to clock output pins for each ADC clock
- Real-time reconfigurability
  - Use to adjust phase of ADC clock in 15º increments (~200ps @ 50Mhz)
- Instantiated with parametrized IP

PLL Instantiation



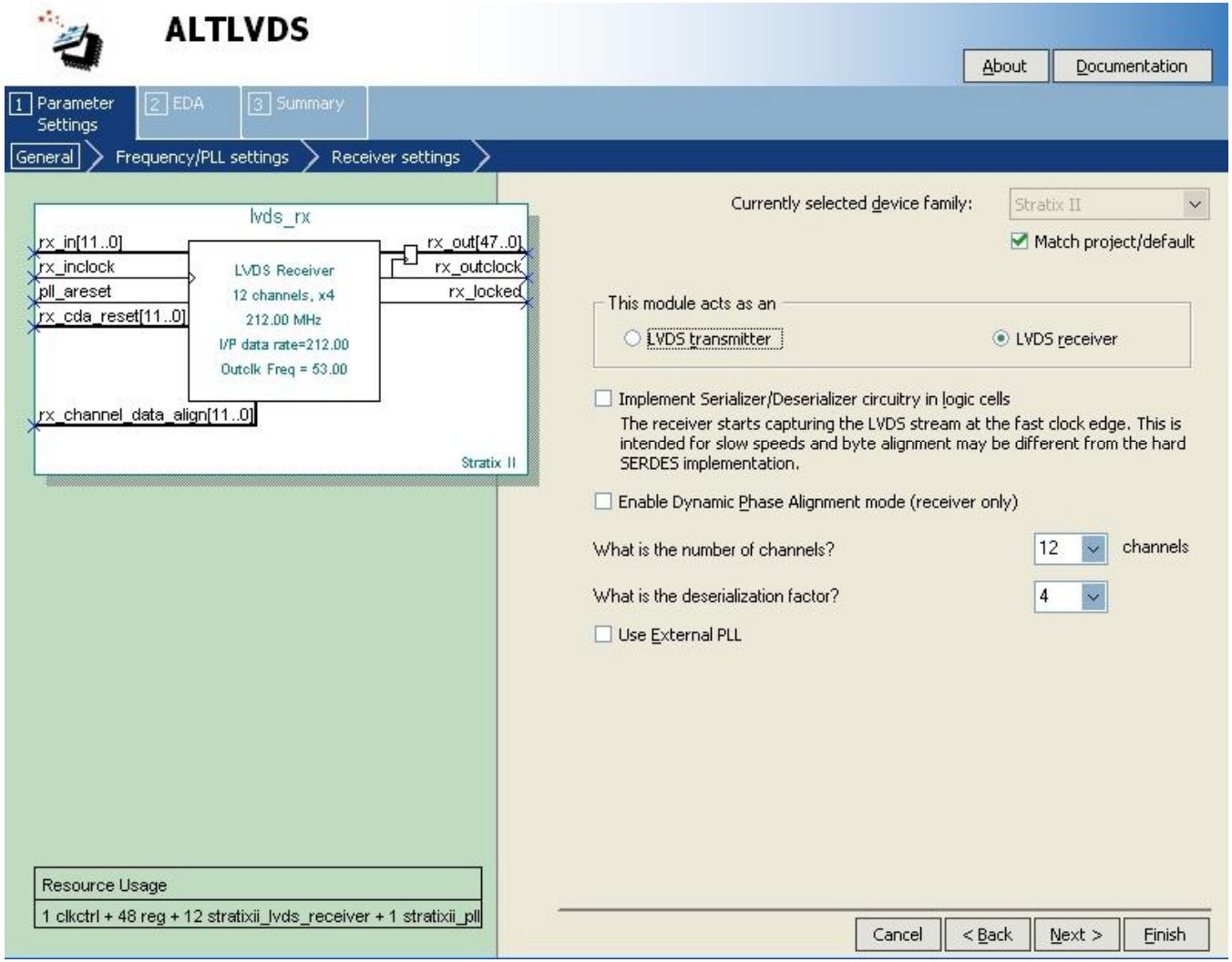
PLL Instantiation



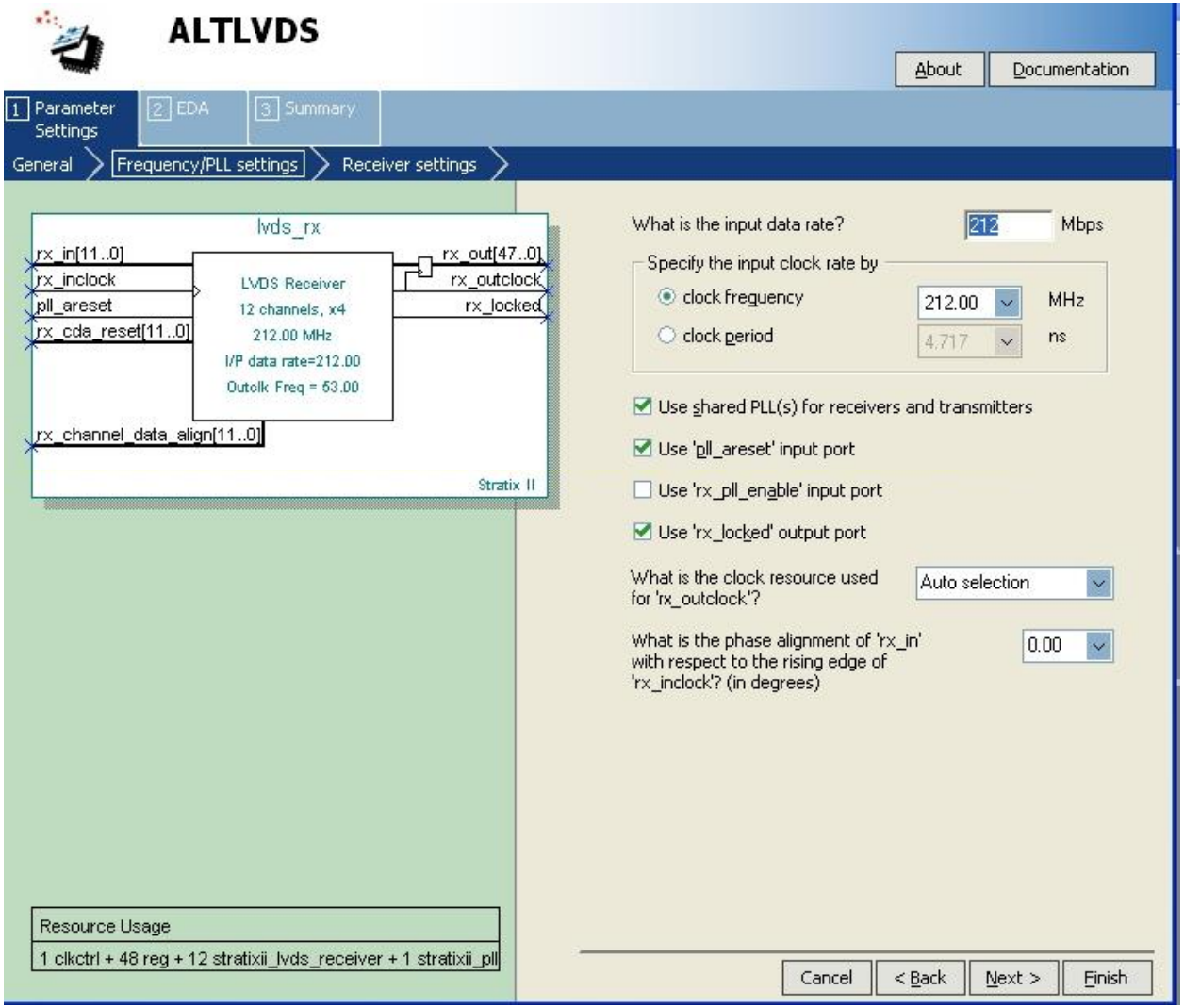
LVDS Receiver

- Instantiated with parametrized IP
- Able to take full advantage of built in RX blocks
  - Dynamic Phase Alignment circuitry
  - Automatically de-serialize data from 4*RF rate to 4 time multi-plexed samples at RF rate
- No need to worry about timing constraints!

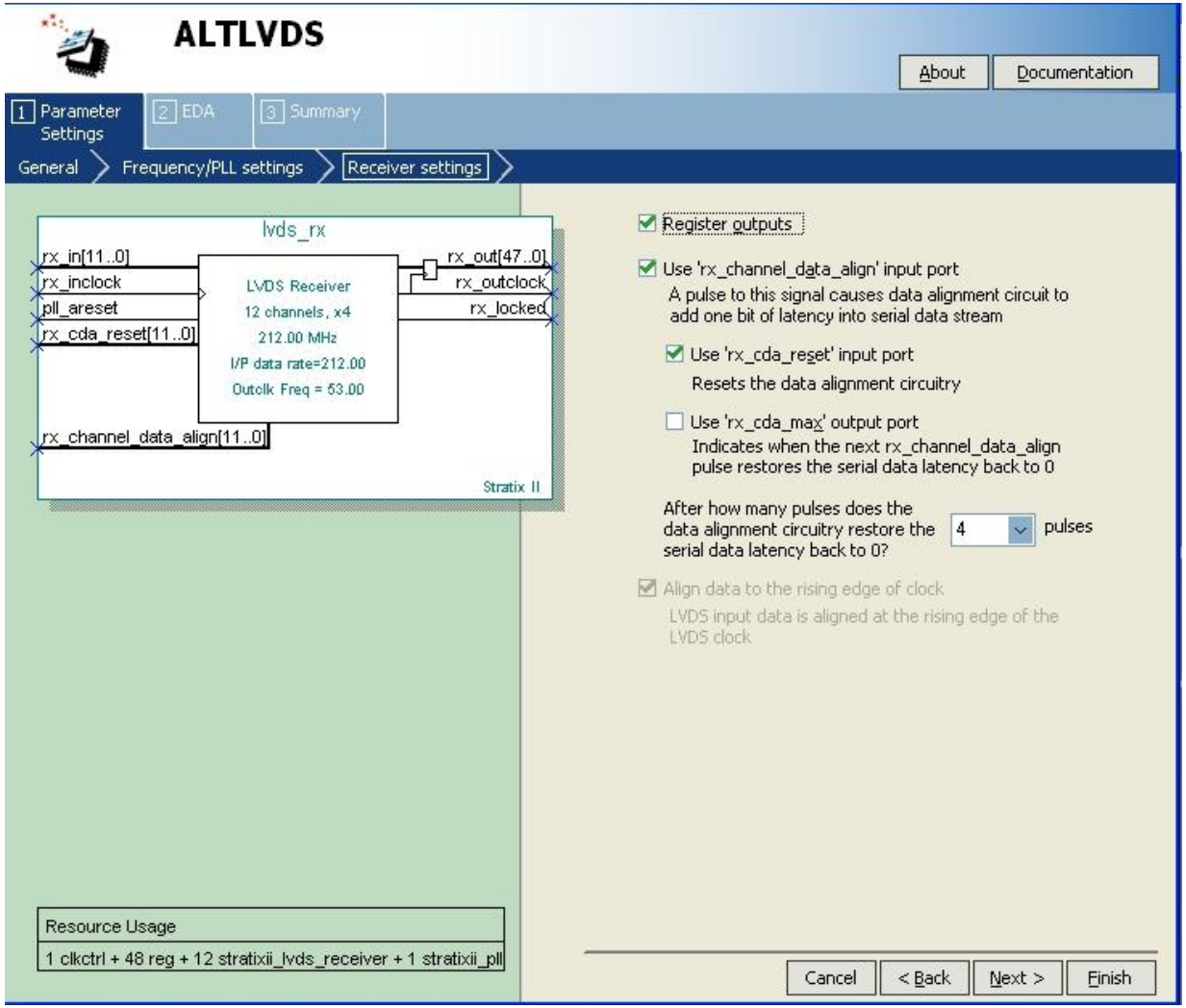
LVDS Receiver IP



LVDS Receiver IP

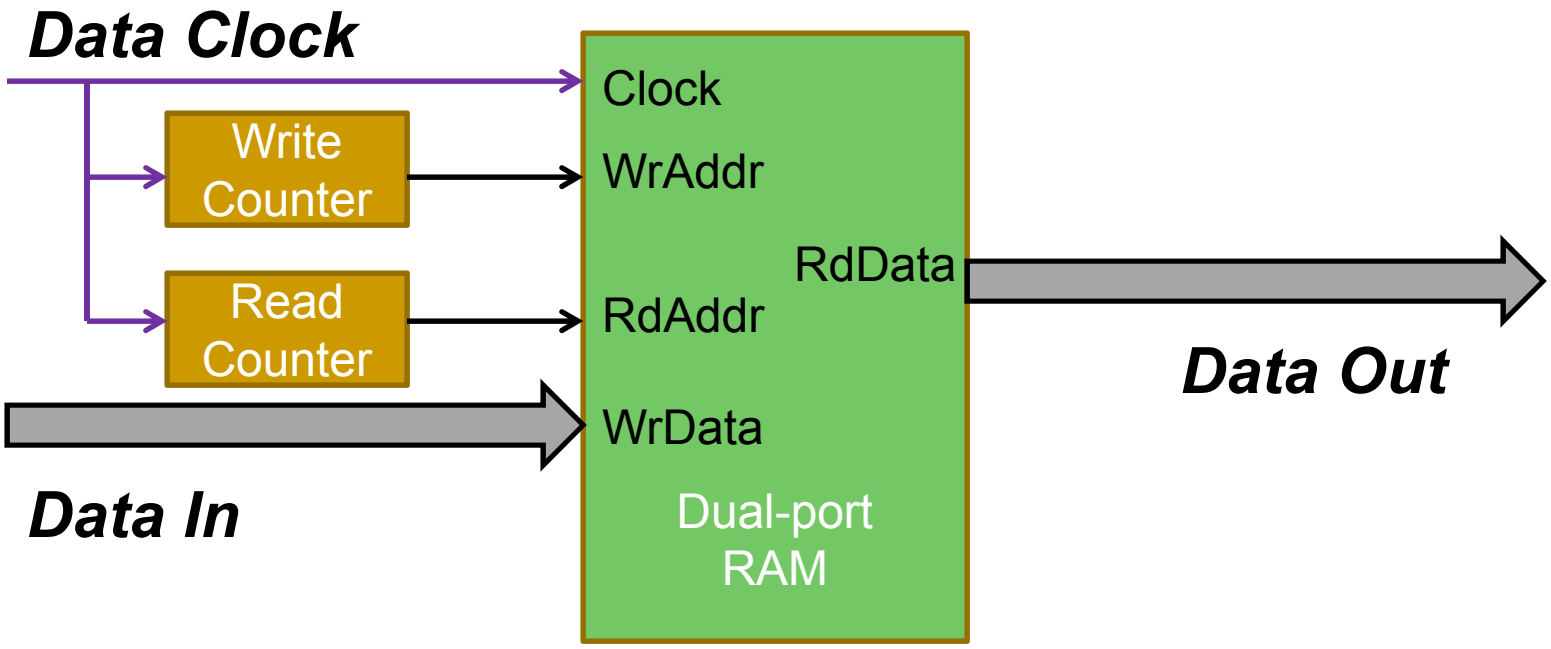


LVDS Receiver IP



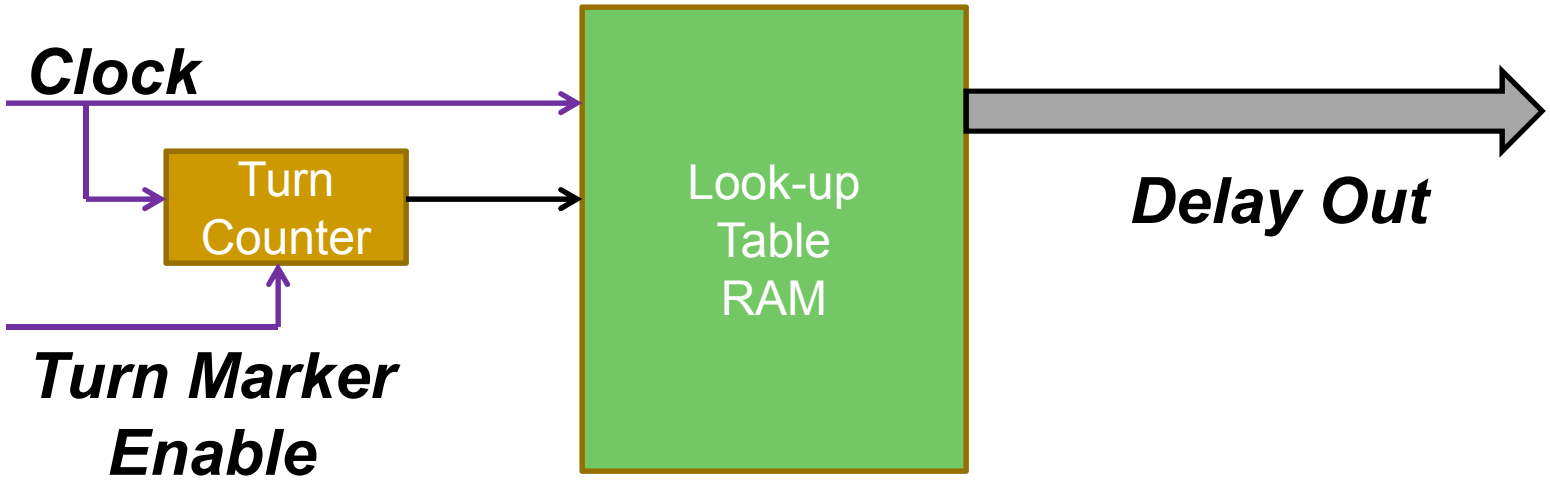
Programmable Delay

- Dual-port RAM to provide programmable delay
  - Offset read & write pointer to requested delay value
    - Reset Read Counter to zero when Write Counter = Delay-1
  - Maximum delay set by the depth of the RAM



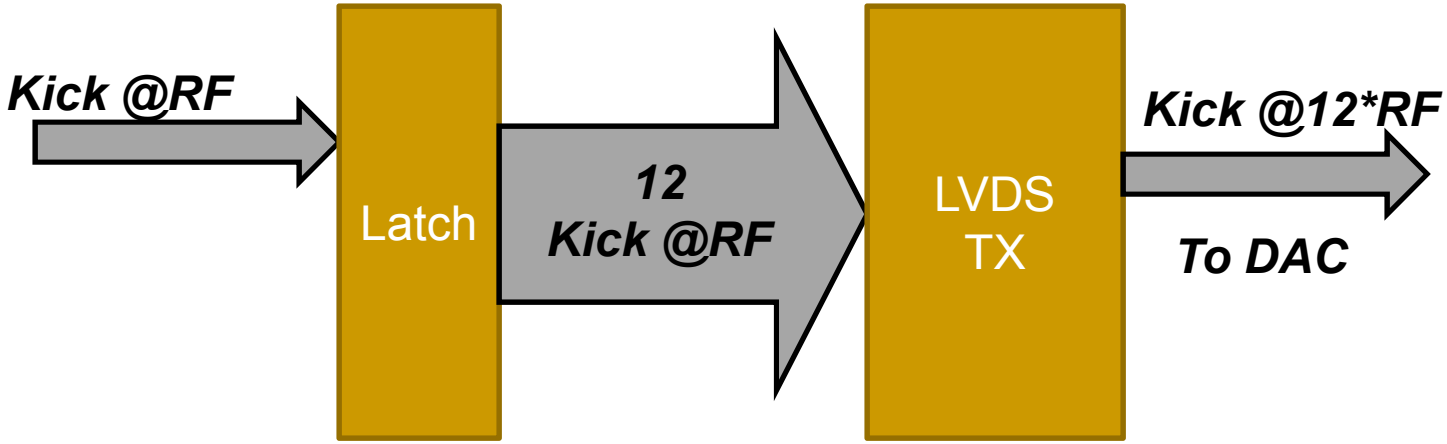
Handling Frequency Sweep

- Digital Portion of design is operating locked to the Booster RF and sweeps along with it
- Need to adjust output delay to account for fixed input/output cable & amplifier delays
- Use lookup table to specify output delay



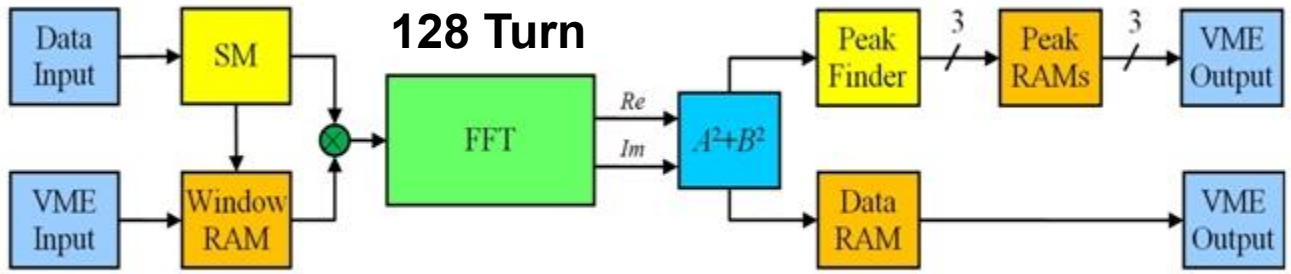
Implementation of "Fine" Delay

- Filter produces kick for each RF bucket
- Operate DAC at 12*RF clock rate



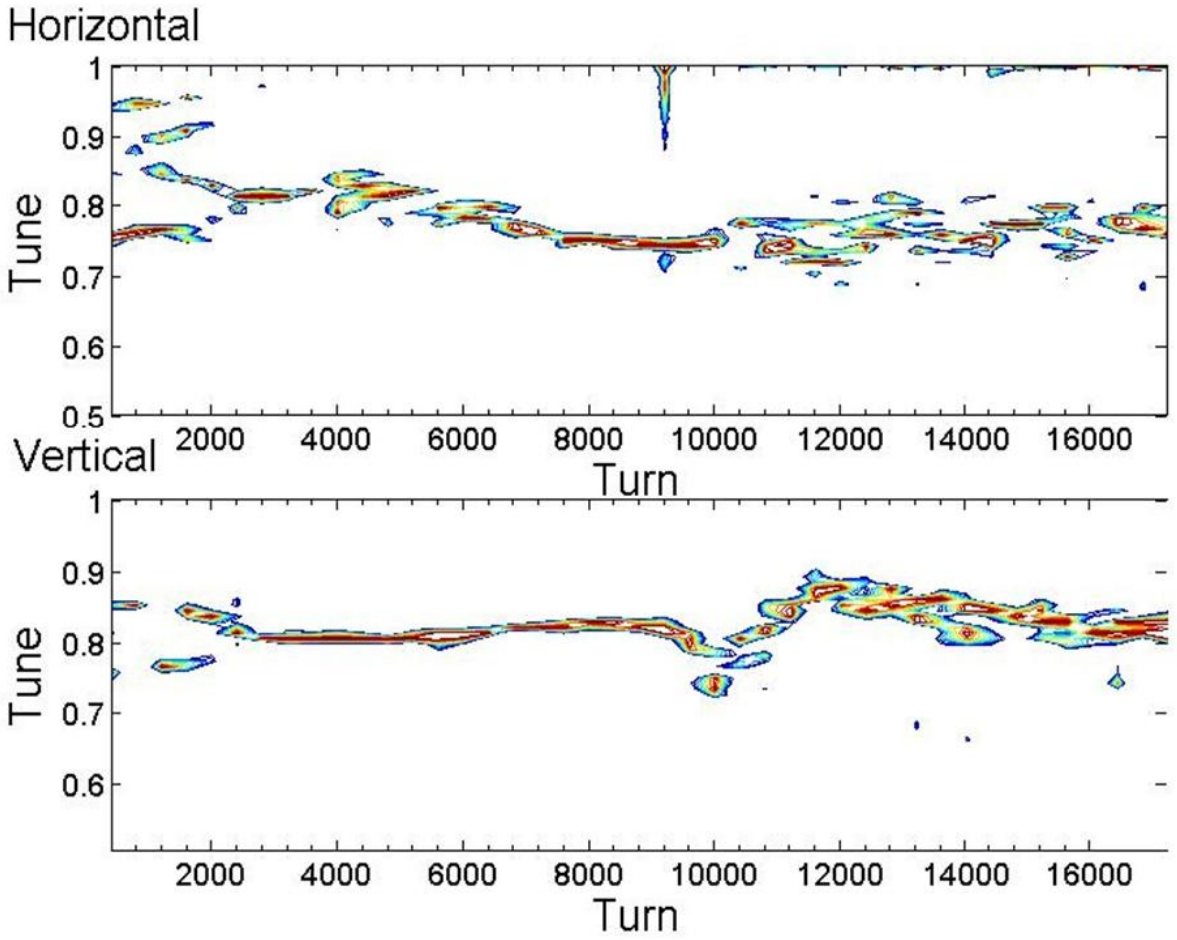
- Can shift data fed to LVDS Transmitter to provide RF/12 delay resolution

Tune Measurement

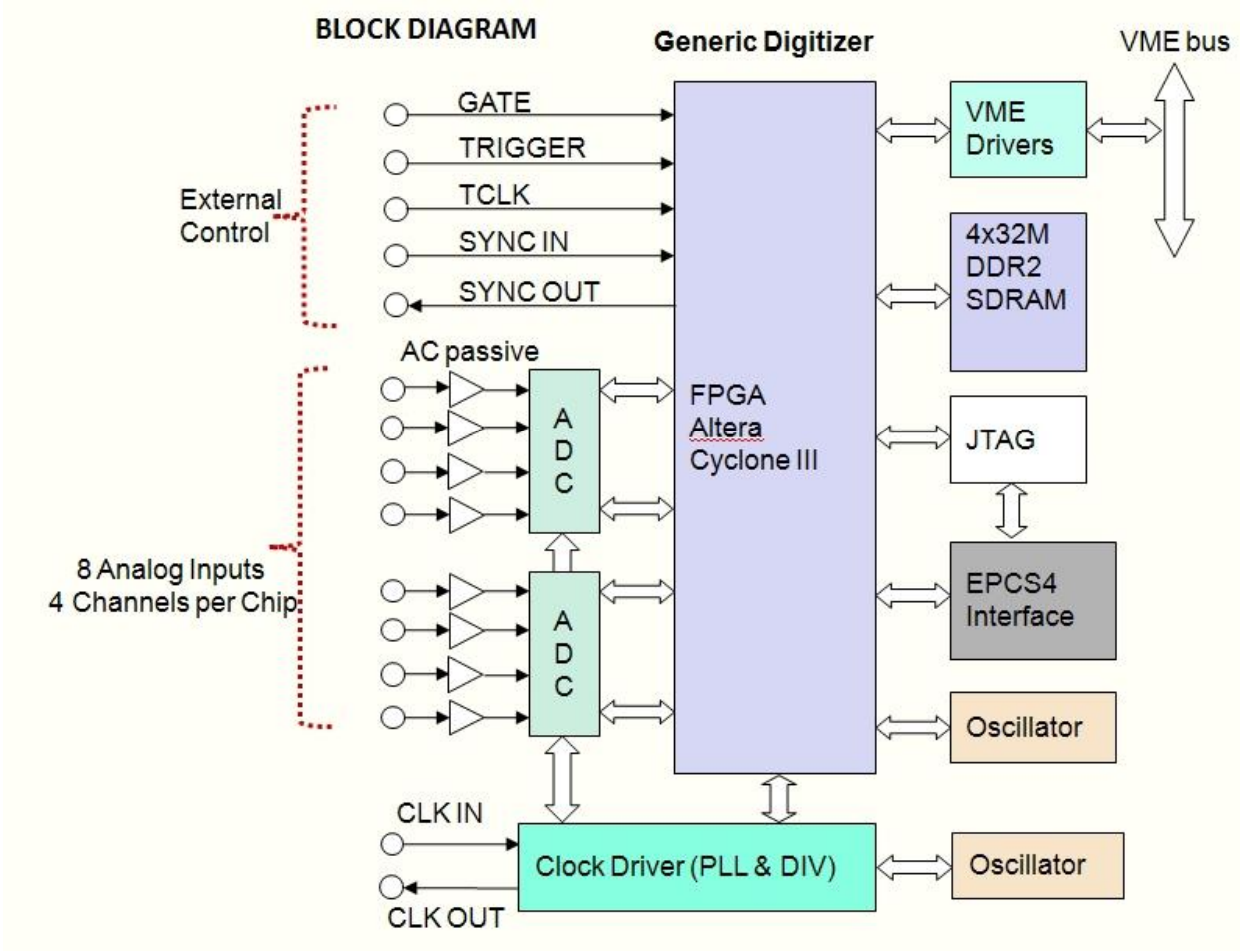


- Excite single bunch with noise via the damper
  - User programmable excitation, noise or anti-damping
  - Can vary gain and duration of excitation at each measurement point
- Simple State Machine to Control Measurement
  - Up to 64 measurements at selectable time (turn) within the machine cycle
  - Select bunch position to pass to FFT engine
- Measure response of single bunch over 128 turns
  - Instantiate fixed point FFT using commercial IP

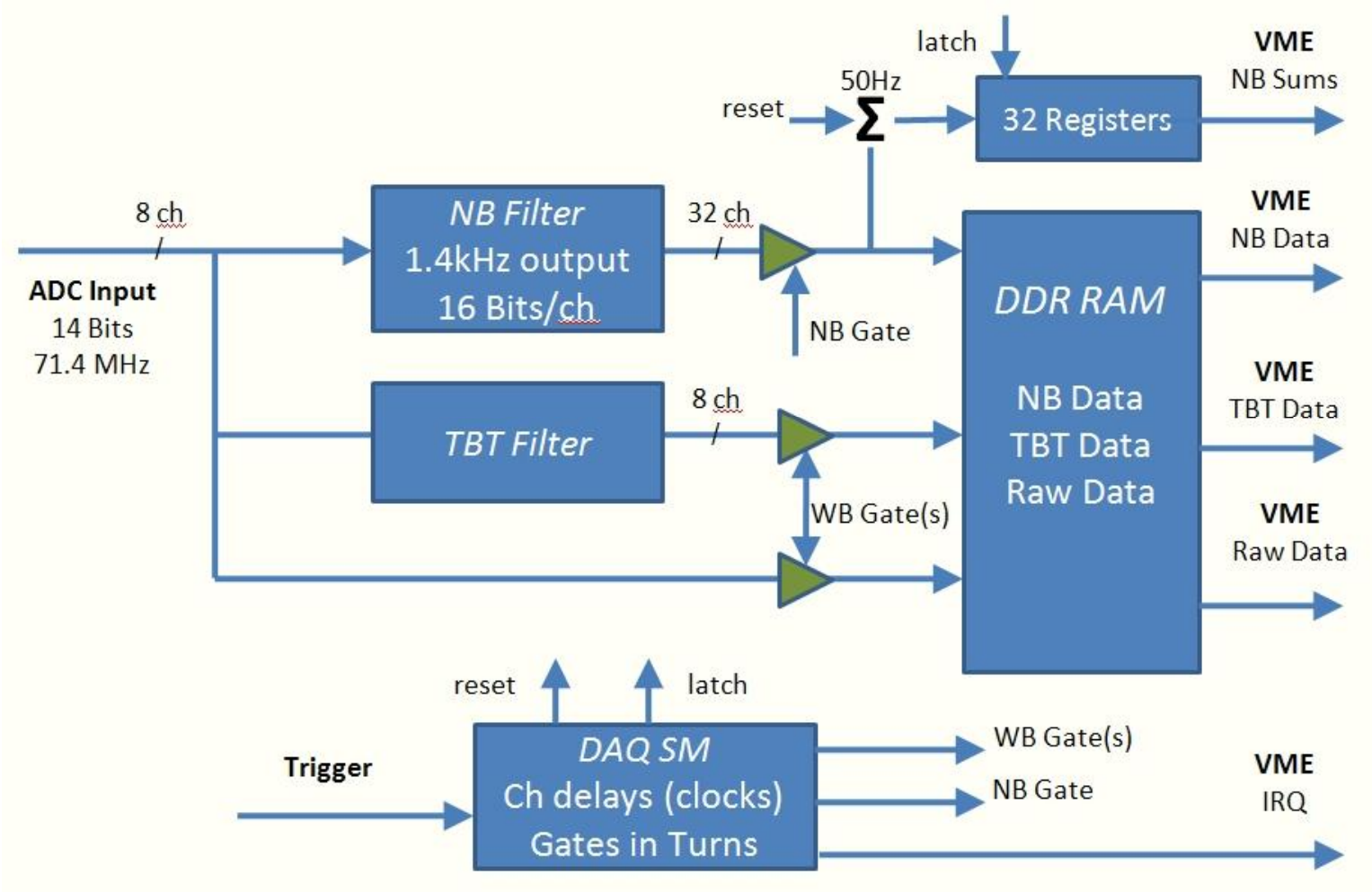
Tune Measurement



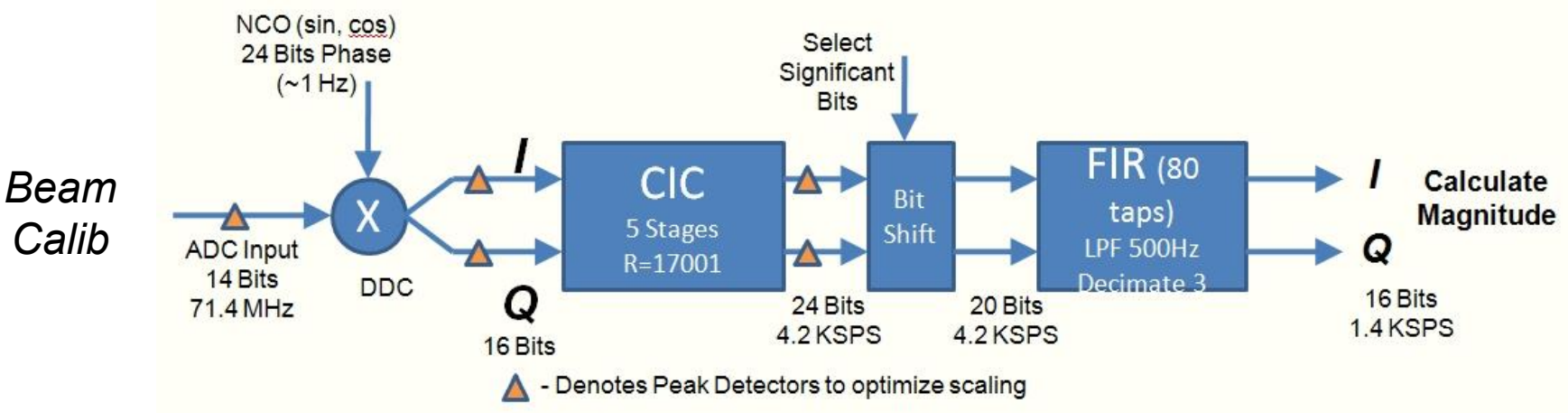
8 Channel 125MS/s Digitizer



Digital BPM Receiver Firmware



Narrowband (Closed Orbit) Filter

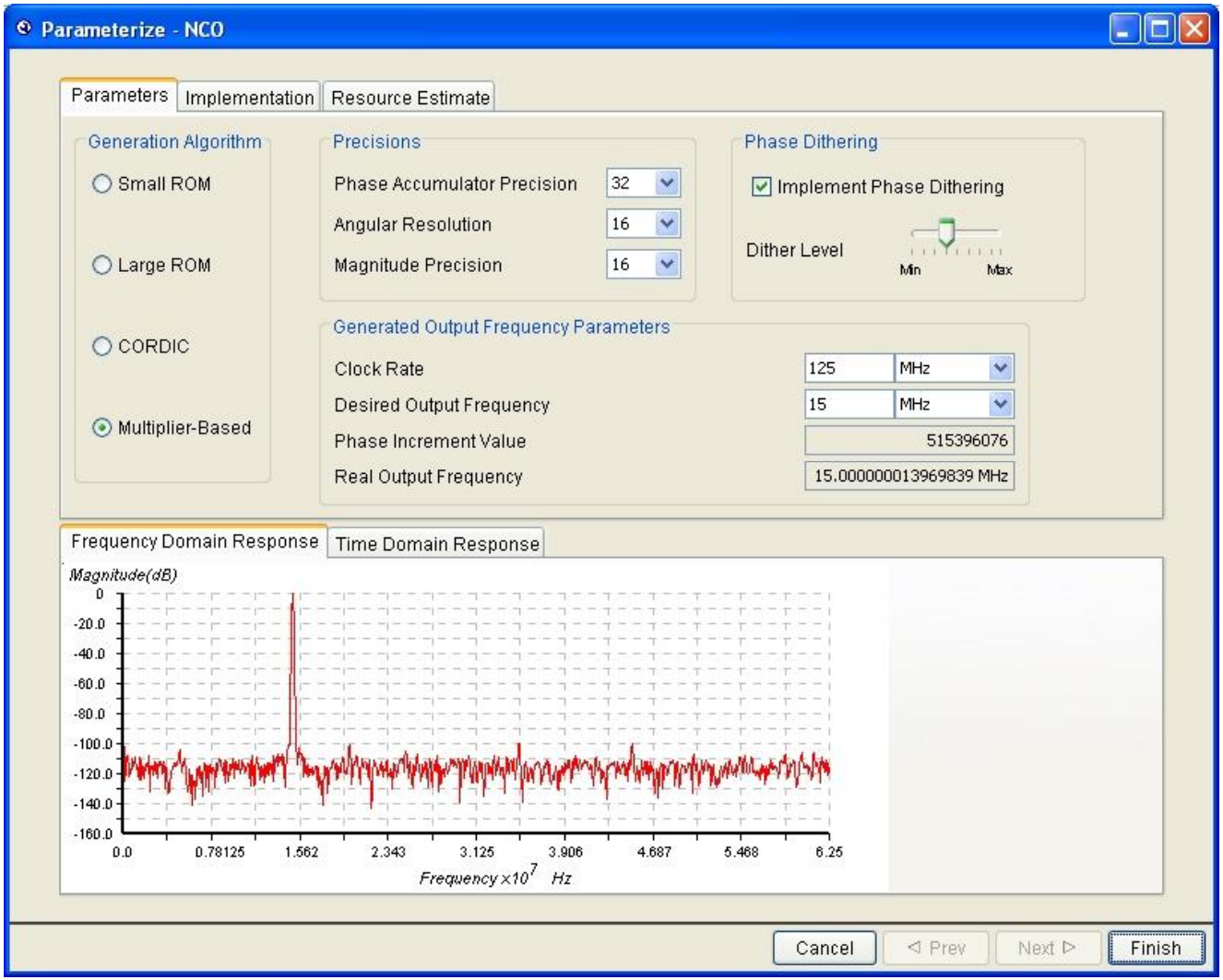


- Digital Down Converter (DDC) Section
  - 32 separate filter paths simultaneously
  - 8 channels, I&Q, 2 frequencies (beam, calibration)
- CIC filters operating in parallel at 71MHz
- Serial FIR filter at 4.2KHz

Numerically Controlled Oscillator (NCO) IP

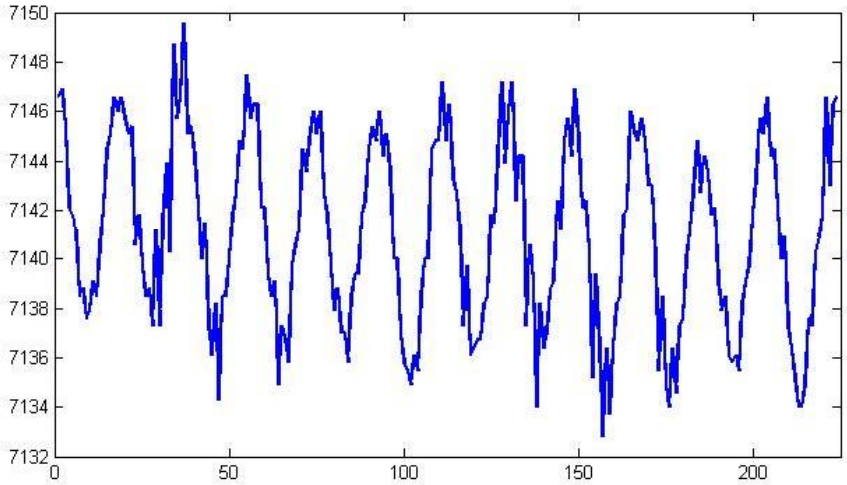
- Generate Sin & Cos signals at requested frequency
  - Frequency Output = $F_{CLK}Phi/2^N$
  - Frequency Resolution = $F_{CLK}/2^N$

NCO IP


Parameterize - NCO
Parameters
Implementation
Resource Estimate
Generation Algorithm
Small ROM
Large ROM
CORDIC
Multiplier-Based
Precisions
Phase Accumulator Precision
32
Angular Resolution
16
Magnitude Precision
16
Phase Dithering
Implement Phase Dithering
Dither Level
Min
Max
Generated Output Frequency Parameters
Clock Rate
125
MHz
Desired Output Frequency
15
MHz
Phase Increment Value
515396076
Real Output Frequency
15.000000013969839 MHz
Frequency Domain Response
Time Domain Response
Magnitude(dB)
0
-20.0
-40.0
-60.0
-80.0
-100.0
-120.0
-140.0
-160.0
0.0
0.78125
1.562
2.343
3.125
3.906
4.687
5.468
6.25
Frequency x10^7 Hz
Cancel
Prev
Next
Finish

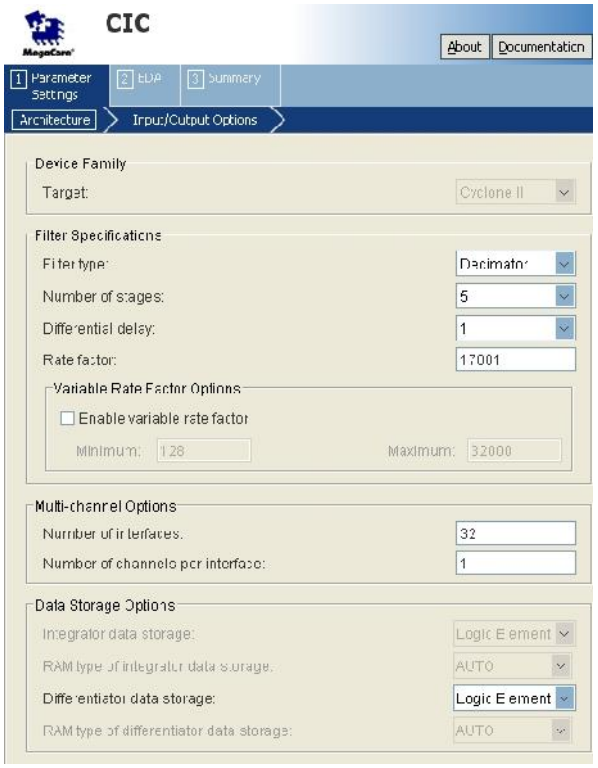
DDC Quantization Error

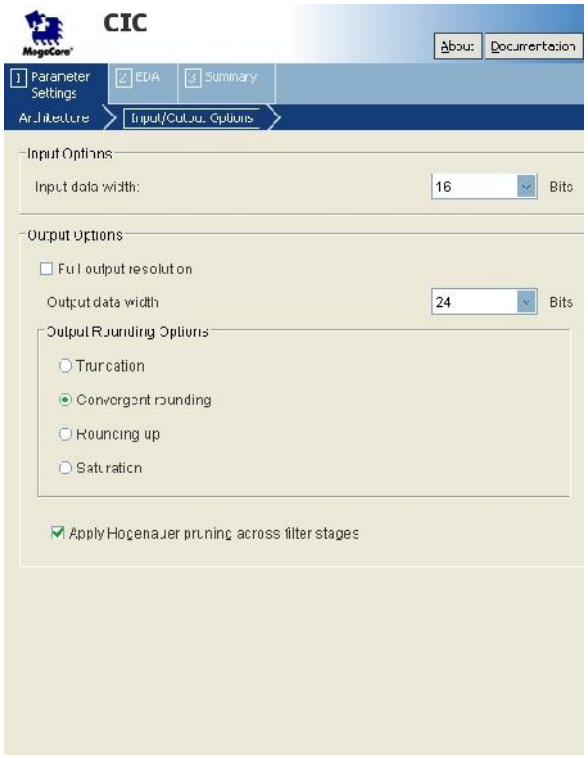
- Will always have a slight offset between the NCO frequency and the beam frequency



- Easy solution is to offset the NCO frequency to get an integer number of periods in the averaging window

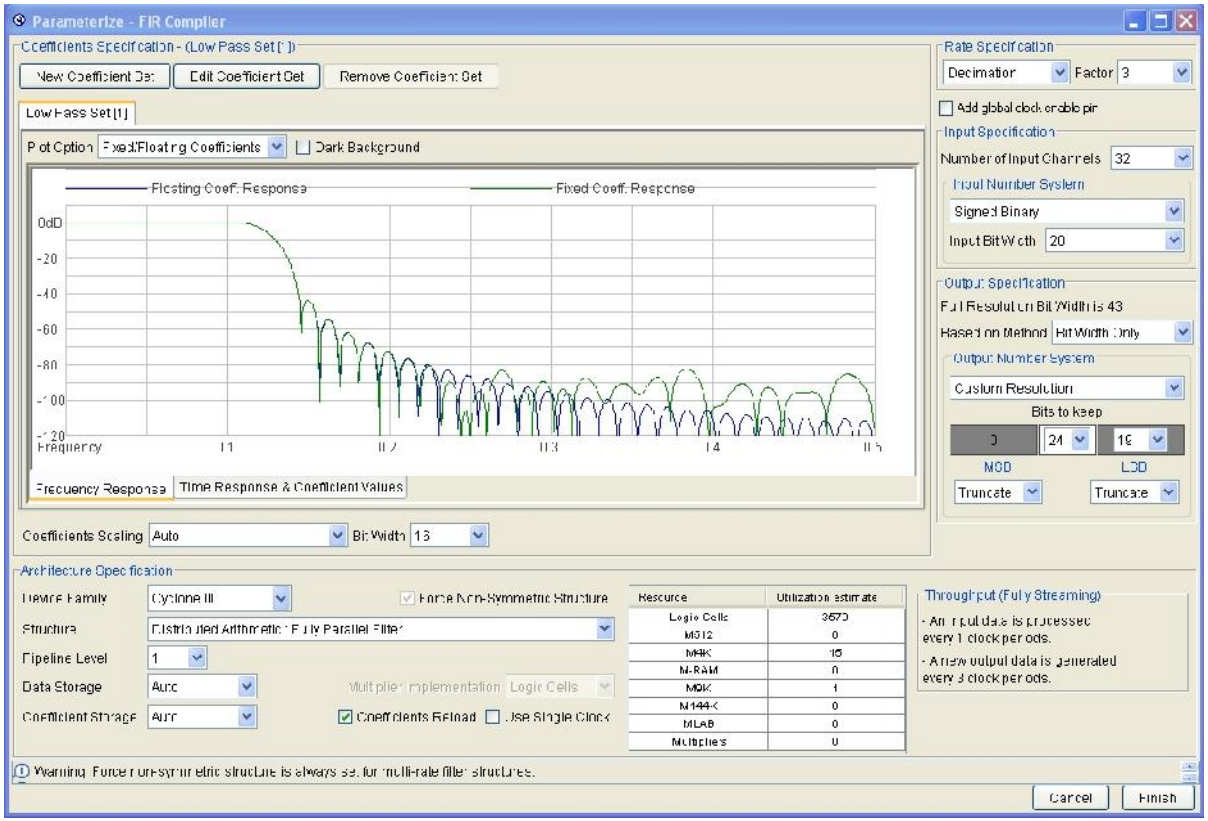
CIC Filter IP





- Commercial IP allows for single instantiation for all 32 CIC filters needed
  - Provides standard serial data output which can be directly interfaced to serial FIR filter

FIR Filter IP

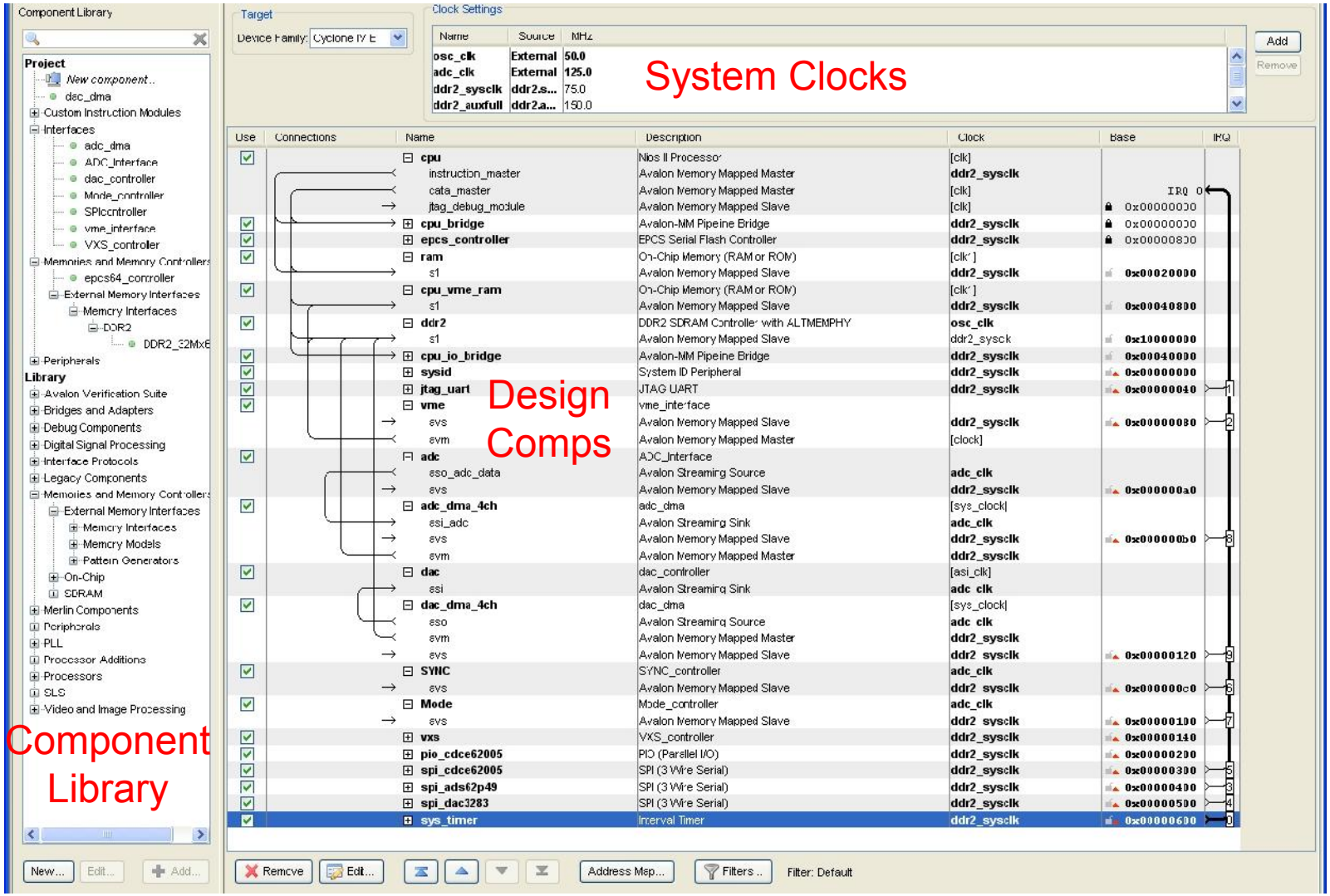


- Provides simple filter design tool or ability to import filter coefficients
- Option to allow modification of coefficients

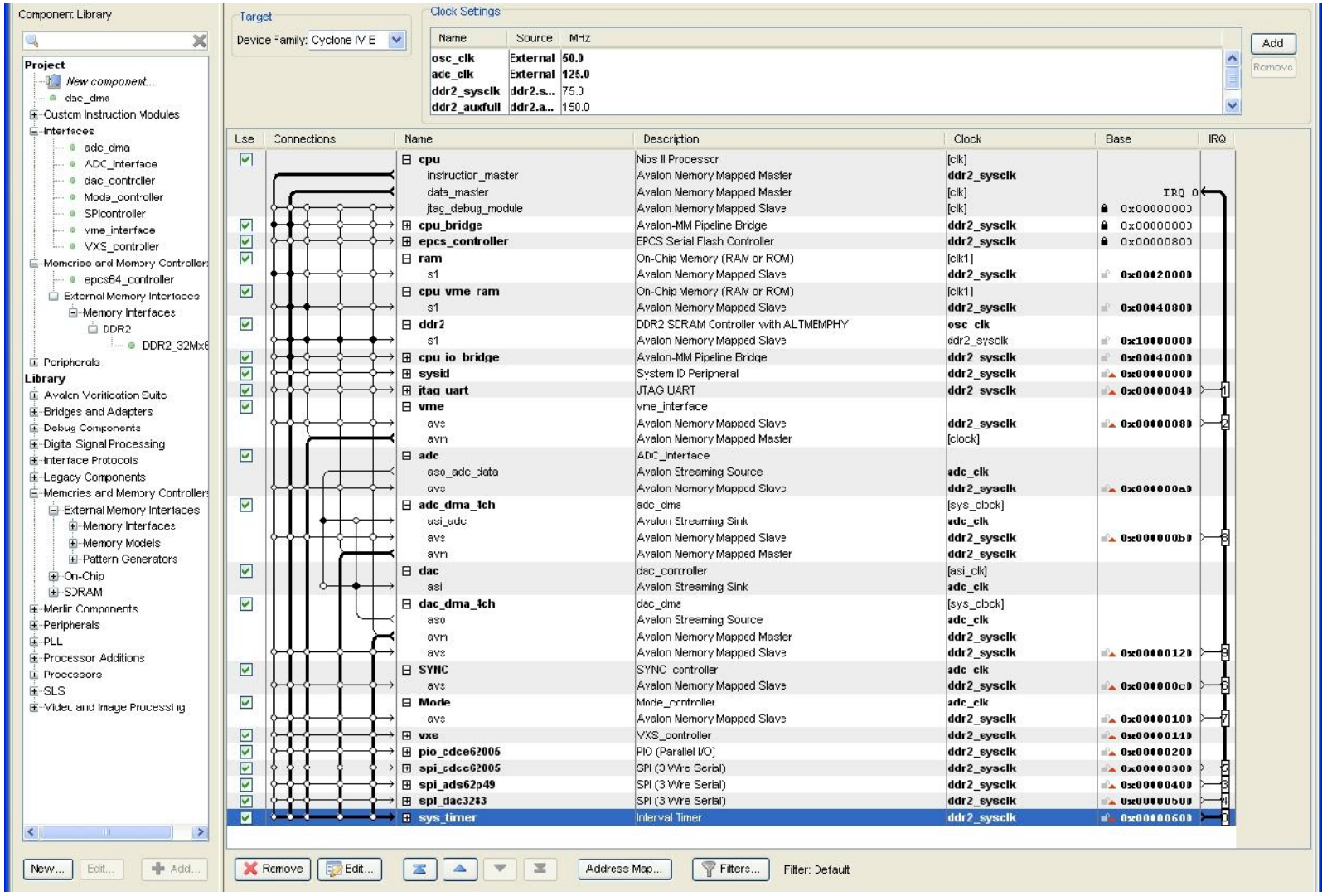
System Integration Tools

- Streamline system integration and design
  - Connect standard interfaces
    - Internal memory, external memory, configuration devices, etc
  - Connect custom interfaces
  - Easy CPU integration
    - Handles addressing & interrupts
    - Even generates drivers for system components!
- Use well defined interfaces
  - Generates all logic for system interconnects!
  - Handles all the timing - clock domains, multiplexing, etc
  - Built in error checking at compile time
- Facilitates implementing re-usable HDL blocks and group design methodologies

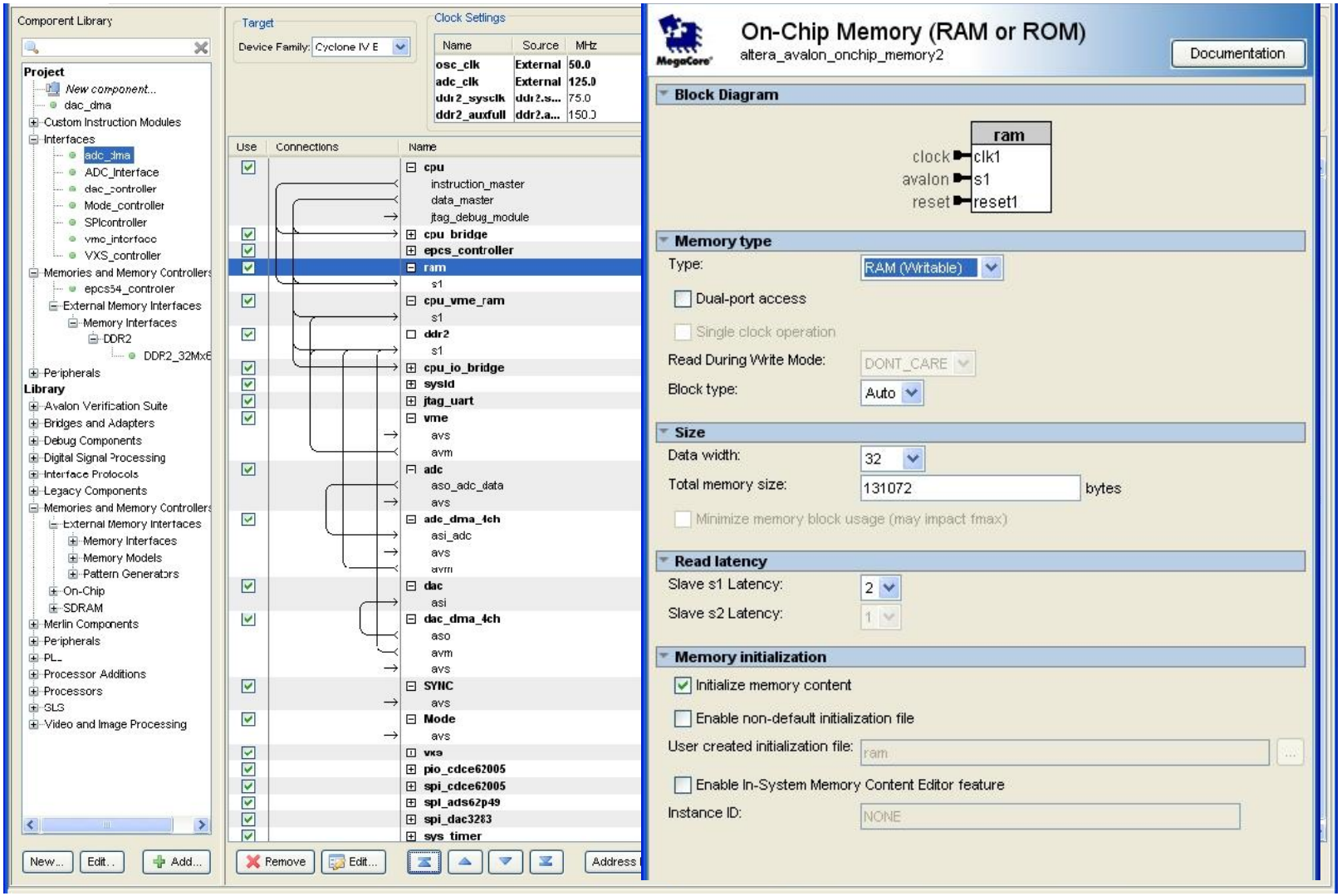
System Integration Tool Example



System Integration Tool Example



System Integration Tool Example



Summary - the FPGA Pitch

- Sure things…
  - FPGA are now the acknowledged leader of cutting edge fast DSP applications where speed and flexibility are needed
  - Accelerator Control and Instrumentation is already using FPGAs to implement fast online applications, especially feedback & control
  - The size, speed, and feature sets continue to grow by leaps and bounds
  - Today's mid level chips are offering features only available in high end chips just a few years ago at a fraction of the cost
  - Design tools are getting closer to traditional programming and becoming easier to use
- Looks promising…
  - Use of FPGA's to implement online orbit measurements and optic calcuations which could be used for realtime feedback
  - The next step is cluster and mesh architectures using FPGAs to further increase the processing power
- It could happen..
  - FPGA based co-processors for dedicated calculations
  - FPGA based super computers which configure their hardware to optimize the performance for the algorithms being used

Thanks for Your Attention!