SystemView 2005

James K Beard, Ph.D.
jkbeard@temple.edu
April 20, 2005

Topics

- Modeling and Simulation Overview
- SystemView Data Formats
- Use of Filters/Systems token collection
- Data Formats for Each Token
- Key Principles of Using SystemView
- SystemView Design and Development Issues
- Timing Issues
- Sample Rate Issues

Modeling and Simulation Overview

- Levels of modeling varies from concepts to details
  - Estimations – computed models (Frii's Equation, EADSIM)
  - Simulative – computations of estimated signal strengths and other quantities in the systems
  - Emulative – voltages, gates and the actual bits
- Data formats in models vary by level of modeling
  - Parameters – SNR, ranges, variances, probability distributions
  - Representations – computed estimations of
  - Simulations – the actual bits, the voltages, etc.
- Validation is necessary for serious use of any model
  - By measurement – open air range testing, etc.
  - By analysis – map results to real data
  - Self-validating – break down into simple examples of known algorithms

SystemView Does All of These

SystemView Data Formats

- Voltages
  - May be any floating point value
  - Usual range is -1 v to +1 v
- Integers
  - Signed – may be any integer in the 32-bit range
  - Unsigned
    - Zero to 2^{32}
    - Symbols are unsigned integers with specified number of bits
- Digital in one of two binary levels
  - Zeros and ones
  - -1s and +1s

Scale, Offset and Quantization Differ

Data Formats and Filters

- Filters/Systems is the default folder in the Operator library
- Appropriate use of filters
  - Voltages: Analog, Laplace
  - Signed integer: FIR, Custom, Z-Domain
  - Receiver matched filters: Comm
- Don't use filters on digital signals, unsigned integers or symbols

Data Formats for Each SystemView Token

- Data formats are implied in the documentation and token parameters
- Input
  - Threshold parameter (typical default 0.5 v) – an input is digital
  - Range parameter (typical default -1v to+1 v) – an input is a voltage
  - Symbol input – an input is an unsigned integer
- Output
  - No parameters – use the documentation, usually real
  - Integer, unsigned integer check boxes – will scale and shift from default real voltage

Data Formats are Usually Clear

Key Principles in Using SystemView

- Always understand
  - The input to every token
    - The data type and format
    - The sample rate
    - The voltage or logic range
  - The output of every token
  - The function of every token -- read all of the documentation carefully
- Look at the waveform and/or spectrum out of each token

Waveforms and Spectra – Seeing is Believing

SystemView Design and Development Issues

- The tokens don't convert or test for data type
- Missing inputs and unused outputs
  - Not tested
  - Inputs may default to zero
- Each token is a delay
  - One sample at the SystemView sample rate
  - The only exception is a bypass of this with the 1/z sample delay token
  - Make the system sample rate several times the largest sample rate in your design

Some Tokens are Sensitive to Timing

- Examples include
  - Bits to symbol decoder
  - Convolution and other FEC decoder
  - Some digital functions in the chip library
- Solving timing problems
  - Examine the input waveforms carefully
  - Use the 1/z sample delay token to force latency to an even number of character or message bits
  - Use the time delay token to set up digital chips

Some Tokens Change the Sample Rate

- Examples include
  - Symbol to bits – increases the sample rate
  - Bits to symbol – changes it back
  - FEC coding – increases
  - FEC decoding – decreases
- Take this into account in setting the system sample rate
- Watch out for additions up-sampling to the system sample rate
  - Never ignore the warning when you start your system
  - An error will usually result farther down the data flow

Decoding and Synchronization

- Bitstream symbol synchronization is left to the user
- Bit to symbol decoding
  - Sample rate decreases in proportion to the bits per symbol
  - The MSB must come in right at the beginning of the symbol decode
  - The [1/z] synchronization delay is in bit stream sample rate
  - Don't do too many bits

Symbol Synchronization is the Essence

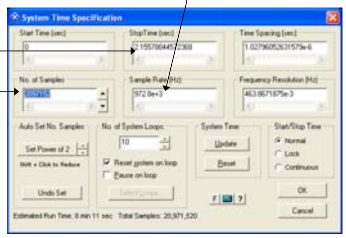
The SystemView System Sample Rate

- Intended for simulation of analog waveforms
- Your system sample rate
  - Must be at least Nyquist for your highest simulated data rate
  - Good practice is 4 to 8 times Nyquist
- Factors in your simulated data rate include
  - Input sample rate
  - Character to symbol ratio
  - Symbol to bit ratio
  - FEC code rate

The SystemView sample rate is that used to simulate analog signals and must be higher than the highest system sample rate

Watch the stop time; it's (no. of samples) over (sample rate)

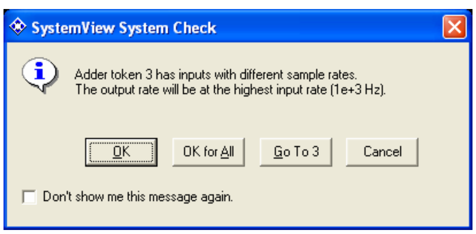
Set no. of samples in run with "Set Power of 2"



Sample Rate and Timing

- General principles
  - Analog signals are usually at the SystemView sample rate
  - Digital signals are usually sampled by your system at a lower rate
- When adding noise to a sampled signal
  - Sample the noise before adding
  - Never ignore a warning

Synchronize the Data



This Warning Always Precedes a Later Error

Bottom Line

- When you think your system is blocked out
  - Make a rough timing diagram
    - Symbol start ticks
    - Bit clock ticks
    - Delays in the tokens
  - Make a table of data formats between tokens
    - Analog, digital, integer
    - Range of voltage or level
- Examine the voltages in the analysis window

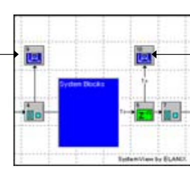
Seeing is Believing

Reconciling Decoder Timing

- Make the system run time only two or three data samples
- If practical, add a decoder on the output of its corresponding encoder
- Look at the waveforms
  - At the input to a working test decoder
  - At the input to your system decoder
- Synchronize by adding [1/z] delay
  - Use waveforms in the analysis window for timing
  - Synch beginning of first data block to the next available start of decoding

Synchronizing Decoder

Look at the bitstream waveform out of the encoder



Look at the waveform into the decoder

Timing Diagram