Waveform Generation for Software Defined Radio

November 1, 2000
The Beaches
Rome, NY

Maurice Schiff, Ph.D.
Chief Scientist
Elanix, Inc.
maury@elanix.com
www.elanix.com

Waveform Generation for SDR

Elanix Inc. Westlake Village CA.
Founded May 1991

SystemView by Elanix
launched May 1996

2000 seats in industry worldwide

200 universities using as primary
teaching tool for digital communications
and DSP signal processing

Objective of Presentation

Introduce SystemView, especially with regard to it's application to SDR.

Initiate a dialog with the SDR community on specific ways SystemView might aid system development.

Waveform Generation for SDR

Complete end to end simulations

Modulation Demodulation

Channel impairments
noise multipath

Bit true DSP operations

Distortion true analog modeling
noise figure, IP1, IP3

Extensive post processing

Waveform Generation for SDR

Communication Waveforms in SystemView

Virtually any waveform is possible. Most already exist in our libraries, or can be generated by a small number of primitive tokens.

User code (c code) generation option allows for addition of highly specialized signals.

Waveform Generation for SDR

Communication Waveforms in SystemView

Analog

AM (arbitrary m)
- dsb
- ssb
- ssb/sc

FM (arbitrary m)

Communication Waveforms in SystemView (cont.)

Digital

MPSK- arbitrary order BPSK, QPSK, etc.

MFSK-arbitrary number of tones
- switched oscillators
- reset phase
- continuous phase

CPM-arbitrary u
- MSK (u = .5)
- GMSK

QAM-any constellation

OFDM

PI/4 DQPSK

PN codes (maximal length)

Walsh codes

Arbitrary tap PN codes

Gold codes

Waveform Generation for SDR

Signal Processing Options

Bit error rate analysis

Eye diagrams

FFT –spectral analysis

Constellation generation

Interference / multipath

Base band pulse shaping –raised and root raised cosine filters

Error correcting codes- block (BCH, RS, Walsh), convolutional (arbitrary), turbo product

System Requirements

- 166 MHz Pentium
- 100 Mbytes of free hard disk space
- 64 Mbytes of RAM, 128 Mb Recommended
- Windows 95/98, Windows NT, or Windows 2000.

SystemView Simulation Engine

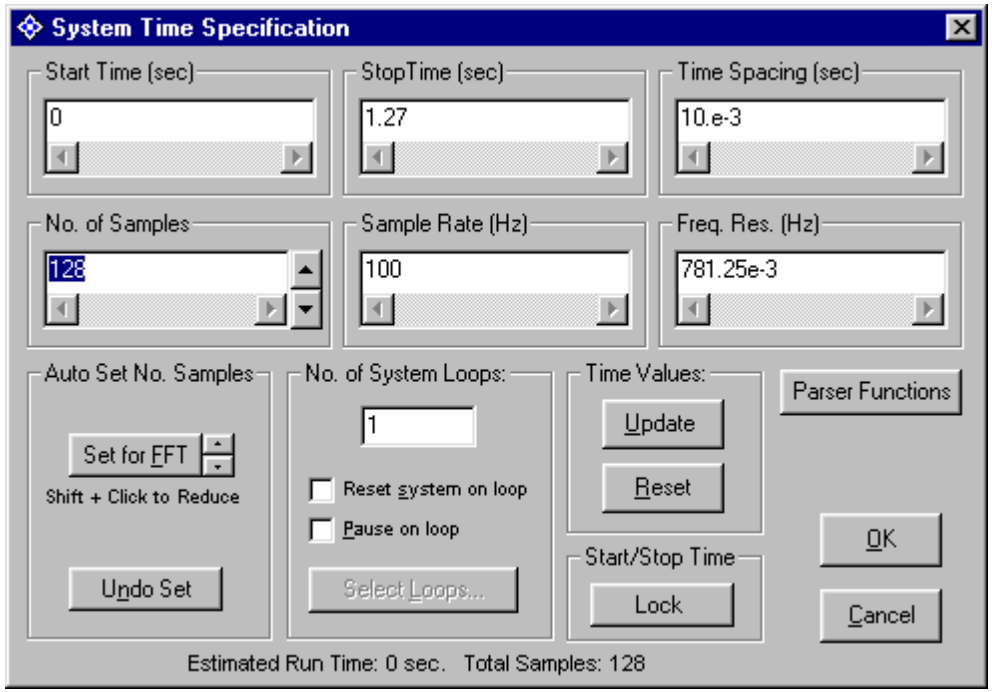
- SystemView provides a visually oriented, dynamic system model
- SystemView uses symbolic tokens (functional models) to represent processes, and a time base to represent system sampling characteristics
- Time based simulator. User can control the start time, stop time, and the system sample rate
- System or subsystem simulations are designed in System window, where tokens are selected from various libraries and connected together in System Design area
- Parameters for these tokens are defined through user-friendly dialog boxes or by direct entry of values

System Time

System Time

SystemView is a true dynamic system simulator. This means your system will behave in SystemView just as it would in the ‘‘real world.’’ Filters have transients, feedback loops exhibit the proper acquisition characteristics, non-linear elements produce the proper time and spectral effects, and so on. Thus, the definition of system time is very important.

Timing is controlled by clicking the System Time button  on the toolbar.



Main Libraries

Library Button
Click this button to toggle between the standard SystemView libraries and the optional SystemView libraries.

Source Token
This token type has outputs only, and represents a library of Sources (including data from external files) used to create inputs to systems.

MetaSystem Token
This token allows you to add hierarchy to your design. The MetaSystem token is used to import a predefined MetaSystem into the current system under design. Refer to Chapter 7 for details.

Adder Token
This token outputs the sum of its inputs.

MetaSystem I/O Token
This token defines the input and output nodes within a MetaSystem.

Operator Token
This token represents a library of Operators that use the input data as the argument for the operation.

Function Token
This token represents a library of Functions that use the input data as the argument to the function.

Multiplier Token
This token outputs the product of its inputs.

Sink Token
This token represents a library of Sinks for collecting, displaying, analyzing, and outputting (to an external file, if desired) the data within a system. Sink tokens have inputs only.

Optional Libraries

Custom Libraries — This library allows you to build your own custom token library. Your token functions are written in C and inserted into the provided template. The module (DLL) may be compiled using any commercial C compiler. Your custom library is automatically integrated into SystemView, and displayed and used just like the built-in token libraries.

Communications Library — This library includes error correcting encoders and decoders (BCH, RS, Golay, Viterbi, Gray), channel models (Jakes, Rayleigh, Rician, Rummler, Multipath), modulators, demodulators, bit-to-symbol, symbol-to-bit, BER counters, frequency dividers, PLLs, Costas loops, and more.

DSP Library — This library includes C4x standard and extended math, conventional or IEEE standard and extended math, mixed radix FFTs, Cosine, Sine, and Hadamard transforms, mean and trend removal, NxN multiply-accumulate, FIR and IIR filter design, fixed bit adders and multipliers, variable-rate data buffers, and more.

C code generation

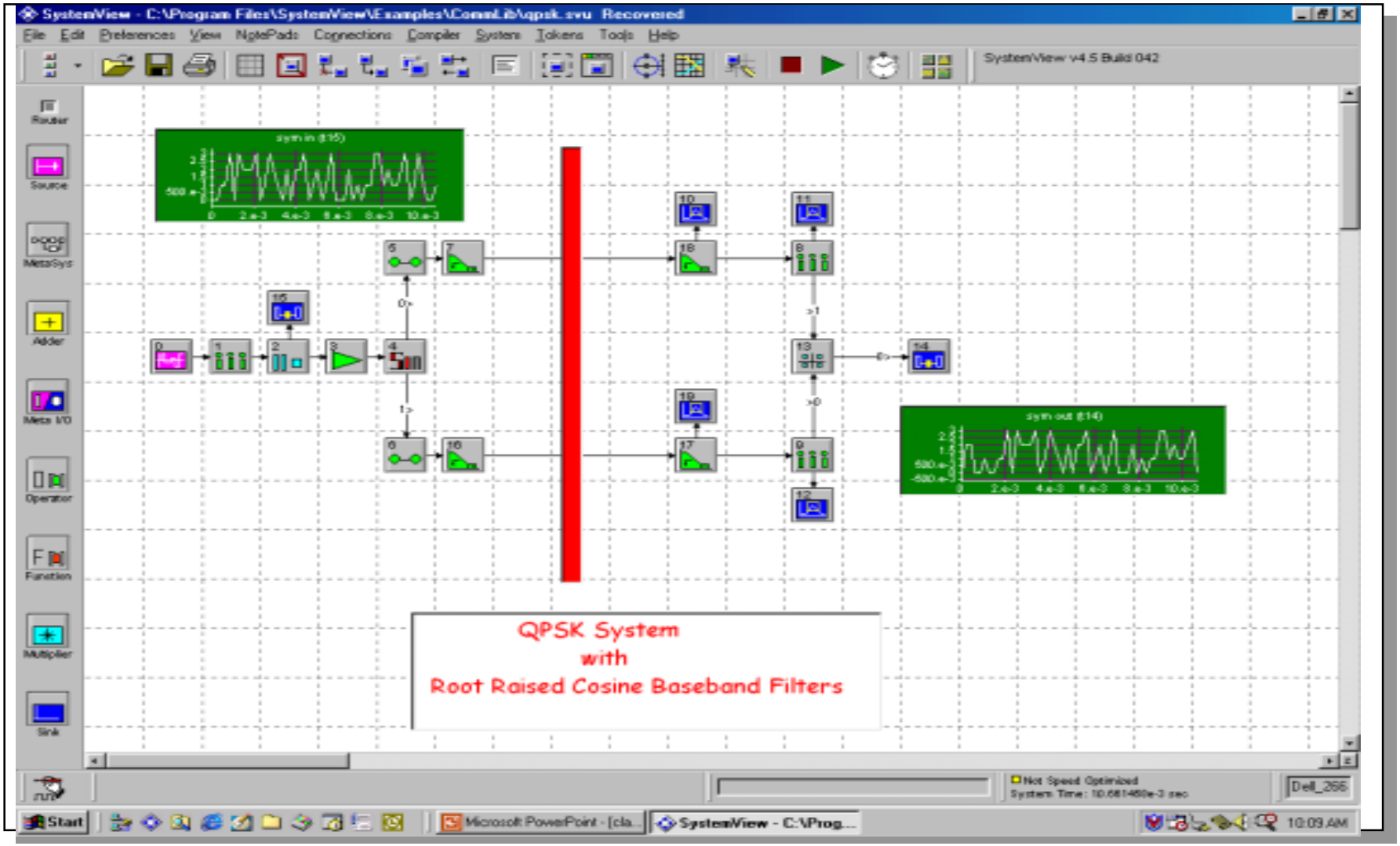
RTDA link thru TI code composer.

Logic Library — This library includes ADCs, DACs, flip-flops (D, JK), counters, dividers, encoders and decoders, and more.

RF/Analog Library — This library includes RF amplifiers, variable log-gain-control amplifiers, op-amp circuits, passive and active mixers, power splitters and combiners, diodes (including Zener diodes), resistor-capacitor-inductor circuits (filters), and more.

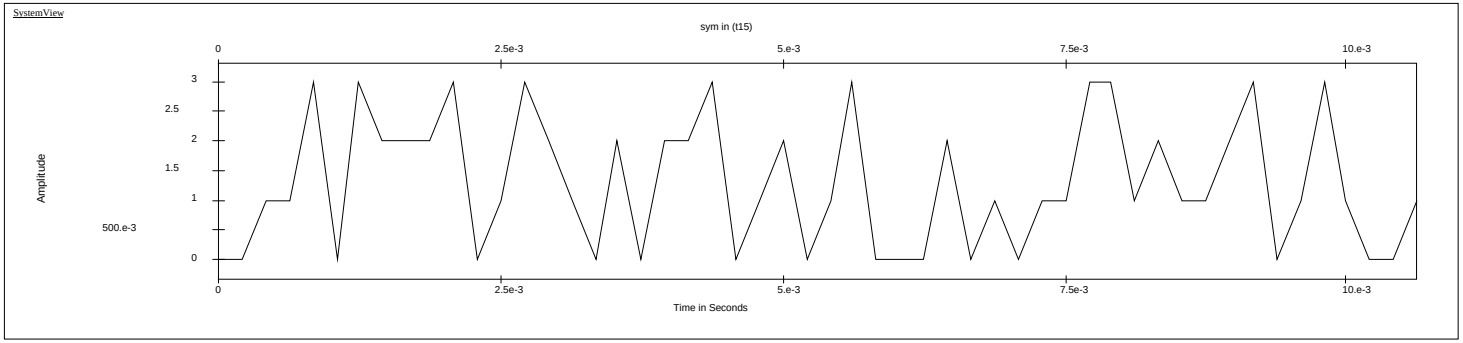
Noise figure, IP1, IP3 capability

Baseband QPSK System with RRC Filters

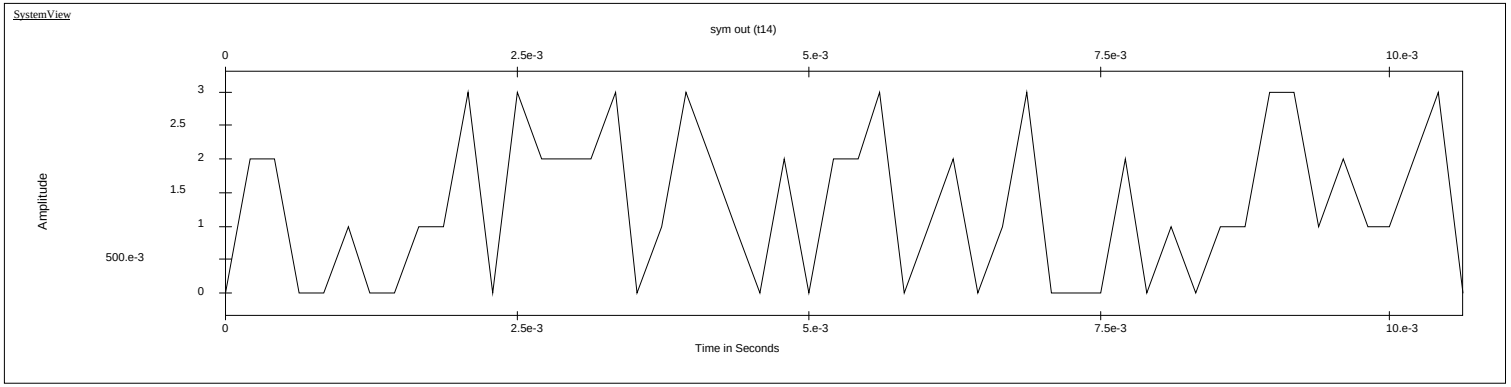


ELANIX INCORPORATED

Baseband QPSK System with RRC Filters

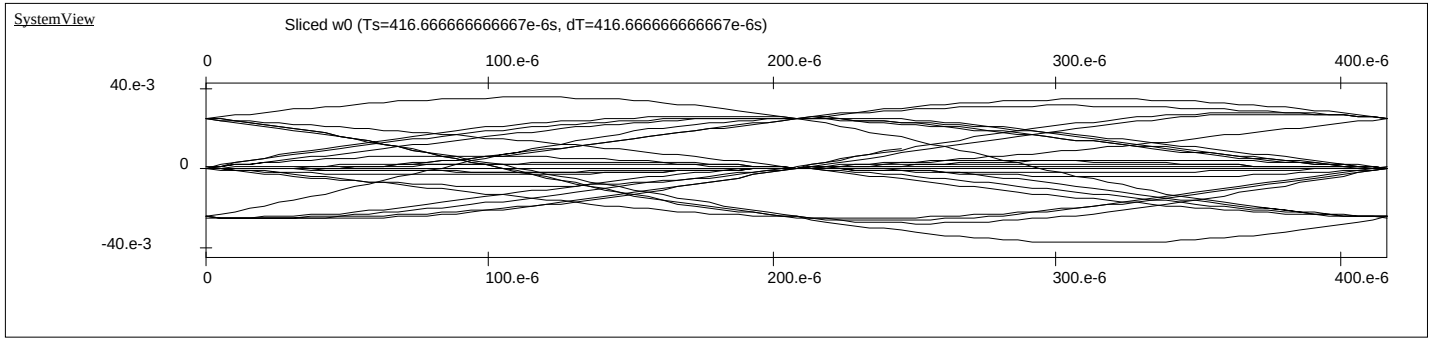


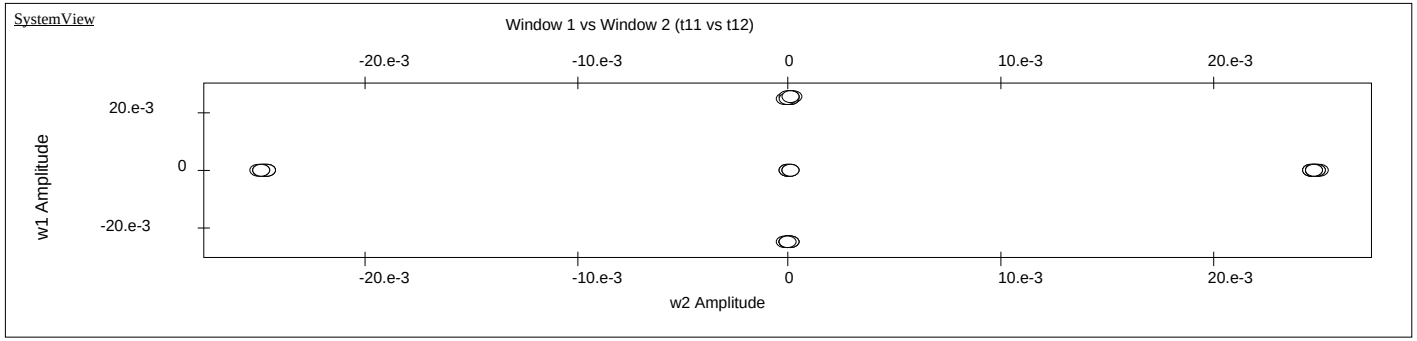
Symbols in

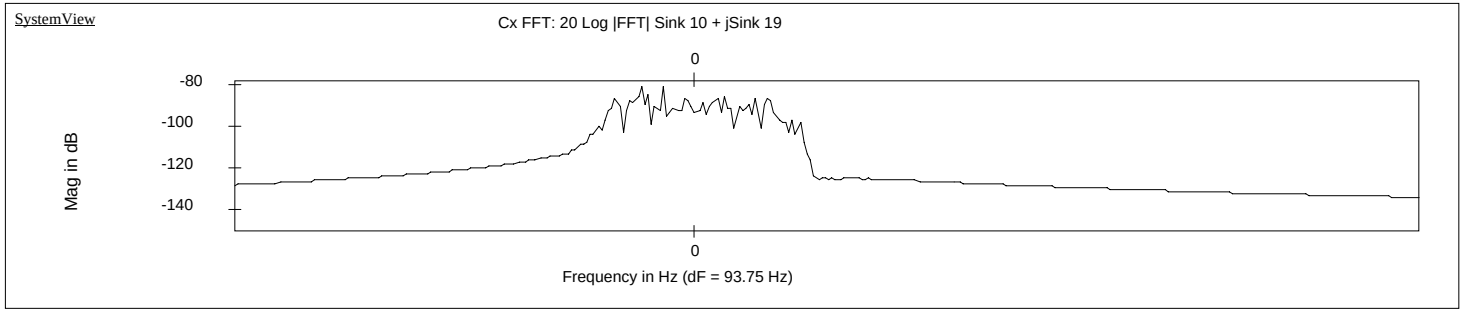


Symbols out

Eye Diagram-Constellation-Power Spectra







ELANIX INCORPORATED