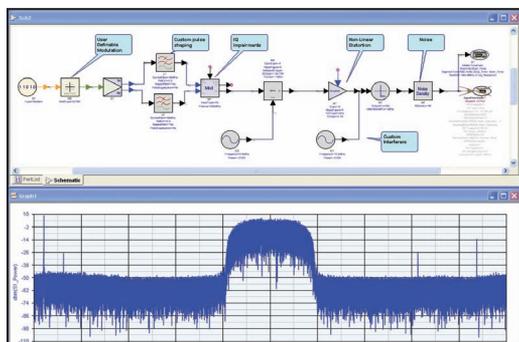


Microwave Journal



CREATING HIGH-PERFORMANCE SDR ARCHITECTURES

Developers of Software Defined Radio (SDR) waveforms are fortunate to have a myriad of high-performance technologies and targets available for waveform implementation in heterogeneous DSP and RF architectures. The challenge is how to co-design RF architectures together with baseband signal processing to create high performance and flexible SDR architectures that can achieve the critical performance specifications necessary in the operational environment.

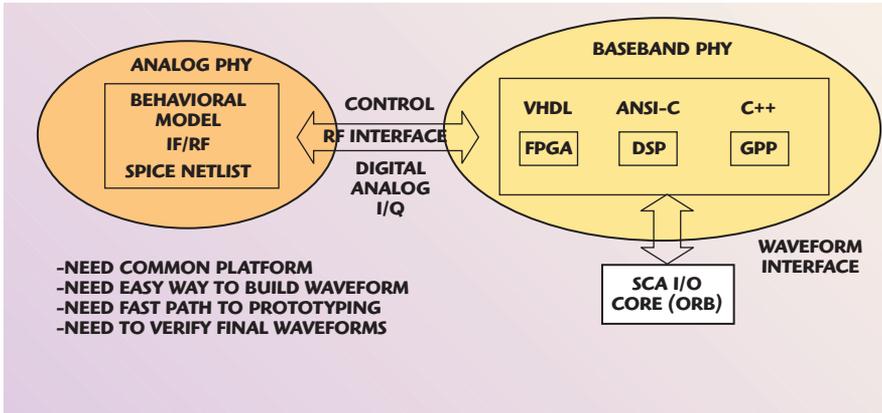
SDR waveform architectures found in military and commercial applications (see **Figure 1**) incorporate broadband high dynamic range RF subsystems, which act as the interface between the baseband processing engine and the real transmission channel. These analog/RF subsystems and associated transmission channels are never ideal and always introduce noise, distortion and other non-ideal impairments that limit the performance of the overall physical layer (PHY). In real systems, the non-ideal effects of these “real” analog/RF subsystems are either accounted for by over-design or through pain-staking, after-the-fact manipulation of the baseband algorithms to correct for these impairments. Being able to model and

account for these RF/channel impairments along with the DSP functions is critical, but many existing design methodologies simply ignore them or use greatly simplified math functions to model these impairments.

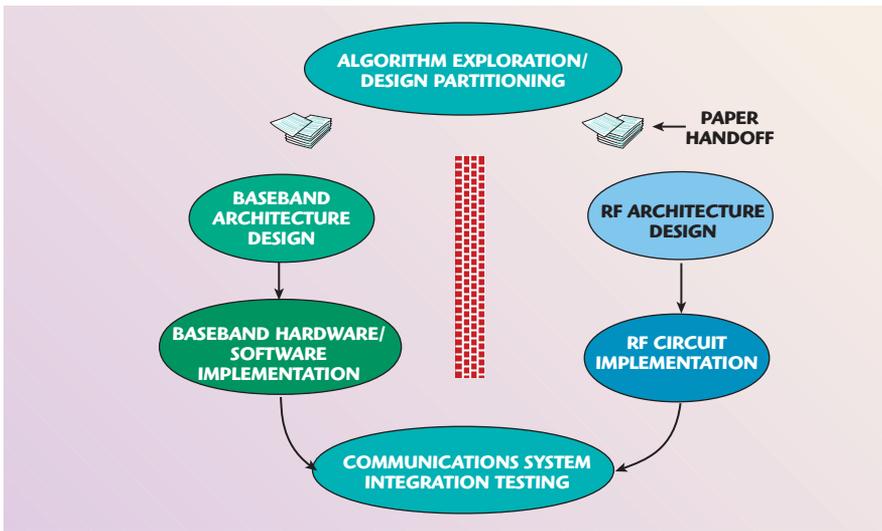
For the baseband designer, FPGA architectures and embedded DSP/GPP engines have reached new reference points in size, performance and power consumption, giving developers of military and airborne systems and lower volume applications plenty of target choices for the creation of high performance, flexible and re-programmable SDR waveform architectures. Taking advantage of the horsepower in these advanced HW targets continues to stretch the development methodologies that have served the SDR community for many years.

These methodologies, which rely heavily on general purpose design and math modeling, are now becoming a bottleneck for efficient SDR design. Many engineers continue to use general purpose Register Transfer Level (RTL) design and math modeling tools, with simple and mostly ideal models of the RF subsystems or utilize

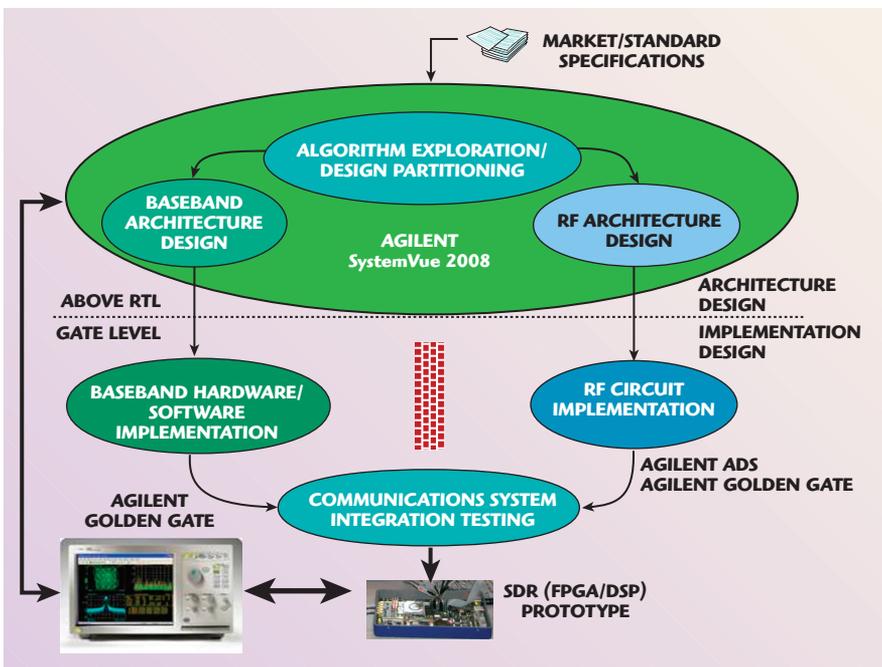
AGILENT EESOF EDA
Santa Clara, CA



▲ Fig. 1 Typical SDR architecture for military and commercial waveform implementation.



▲ Fig. 2 Discontinuous system architecture flow.



▲ Fig. 3 SystemVue 2008 connected design methodology.

in-house, hand-coded methodologies that tend to create discontinuities in the overall task of system architecture design and algorithm implementation. **Figure 2** shows a discontinuous system architecture flow.

With the emergence of electronic-system-level (ESL) design methodologies there are new classes of design automation tools available today to help with older strained methodologies. However, most of these ESL solutions are focused on high-level design synthesis of digital systems, allowing higher modeling abstractions. Often these ESL tools ignore the analog/RF systems, providing value only for the digital baseband and SW engineers. Agilent Technologies EDA division has recently announced its new SystemVue 2008 product, which is focused on the task of rapid development of SDR waveform architectures and algorithm development. The software unifies the disjointed architectural and algorithm design flow, connecting high level algorithm design with lower level HW architectural design for both baseband HW and analog/RF architectural design. This new tool bridges the architectural design “gap” allowing seamless integration of algorithms, HW descriptions and analog/RF behavioral modeling all in a simple and intuitive graphical design environment. **Figure 3** shows the connected design methodology afforded by Agilent SystemVue 2008.

Most algorithm developers prefer textual-based modeling and debugging over graphical design tools. SystemVue 2008 provides algorithm developers with an integrated design environment (IDE) for developing and debugging text-based algorithm models in familiar M-code (math) and C/C++ formats. These integrated code development interfaces allow complete mathematical and algorithmic coding, including the ability to model algorithms in fixed point using industry standard SystemC fixed point class. Developers can quickly model algorithmic behavior with immediate creation of simulation models that can be wired graphically into the rest of the design to form the basis of the design architecture.

Figure 4 is a screen shot of the M-code modeling interface. From

this interface M-code algorithms can be written and debugged. This interface is also used for bringing real world data into the environment or

scripted to control a simulator.

As algorithms move through base-band architecture and HW implementation, libraries of architectural building blocks support fixed point design for implementation in FPGA or ASIC. The fixed point simulation offers advanced analysis and optimization of finite precision math, displaying histograms of numerical overflow and underflow, along with the ability to manipulate word-length to

maximize performance and minimize implementation size.

Specific GUI features allow users to associate alternate model views for different architectural building blocks, allowing floating-point schematic, floating-point M-code or C++, fixed-point C++, fixed-point schematic and fixed-point user-supplied HDL model views for each block. This simplifies the task of having separate environment and test harnesses to validate fixed-point and HDL functionality against early algorithmic models. SystemVue 2008 also integrates Mentor Graphics ModelSim co-simulation, allowing simple inclusion of user supplied HDL to be simulated using this industry standard simulator.

```

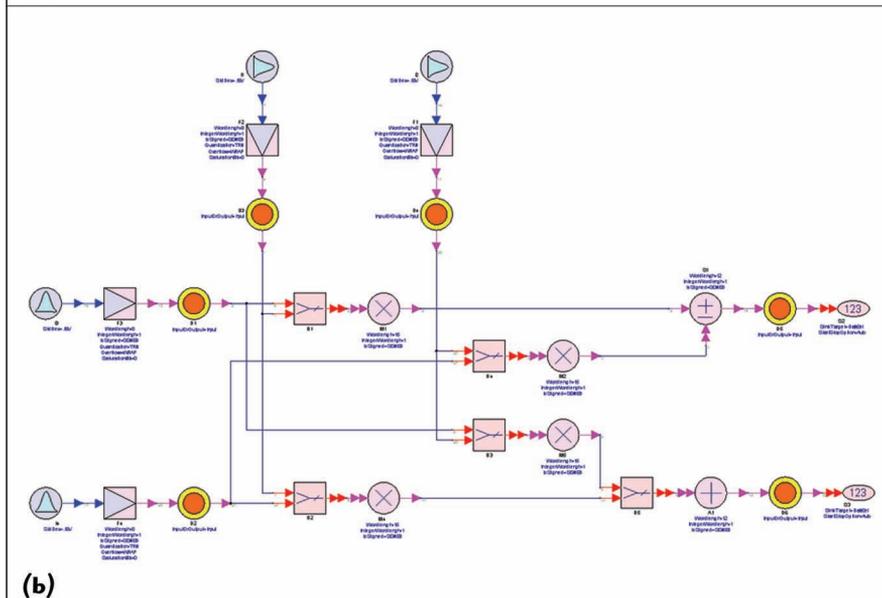
1 % get the generator polynomials,
2 Turbo_G1 = [1 0 1 1; 1 1 0 1];
3 Turbo_G2 = [1 0 1 1; 1 1 0 1];
4
5
6 Output = zeros(1, m_Output_Len);
7 NumIter = TC_Iteration;
8 Algorithm = 0; %log_map
9 EbN0 = 2;
10 rate = 2;
11 tb_idx = find(TransBlockSize > 0);
12
13 sum_in = 0;
14 sum_out = 0;
15 for i=1:1:numel(tb_idx)
16     if m_C(i) == 1
17         Kc = m_Kplus(i);
18         map_intv = zeros(1, Kr);
19         [map_intv] = LTE_TurboIntv_Map(Kr);
20         Output(sum_out+1:sum_out+Kr) = turbo_decoder(Inp
    
```

▲ Fig. 4 The floating point M-code.

```

1
2 -- Automatically generated VHDL code for non-primitive component
3 -- ComplexMult.vhd
4
5
6 library IEEE;
7 use IEEE.std_logic_1164.all;
8 use IEEE.numeric_std.all;
9 library work;
10 use work.p_fxp.all;
11
12
13 -- declare complexMult entity
14
15 entity ComplexMult is
16
17
18
19
20 port
21 (
22     D1 : in std_logic_vector(8-1 downto 0);
23     D2 : in std_logic_vector(8-1 downto 0);
24     D3 : in std_logic_vector(8-1 downto 0);
25     D4 : in std_logic_vector(8-1 downto 0);
26     D5 : out std_logic_vector(12-1 downto 0);
    
```

(a)



(b)

▲ Fig. 5 A fixed-point GFSK modulator built using SystemVue 2008 and resulting VHDL.

Optional HDL code generation from fixed-point algorithmic descriptions supports fully synthesizable IEEE compliant VHDL and Verilog RTL generation as an added convenience to fixed-point algorithm developers. This allows designers to quickly generate fully synthesizable HDL from early algorithmic descriptions for early and rapid algorithm prototyping in FPGAs.

Figure 5a and 5b show a fixed-point GFSK modulator built from Agilent SystemVue 2008 with the resulting VHDL for the top level architecture. Because all SDR waveform architectures contain an analog/RF subsystem, real-world channel impairments will degrade the overall PHY performance upon final implementation of the system. It is imperative that early system architects and algorithm developers have a quick and easy way to prototype virtual RF architectures including the effects these impairments will have on the system and to explore algorithmic means for correcting or accounting for these impairments.

The underlying simulator in the SystemVue 2008 platform is uniquely designed to handle modeling and analysis of the real modulated signals propagating through a channel, providing, in some cases, orders of magnitude more computational efficiency than the leading “math” tools, without requiring the need for additional HW acceleration or compute farm support. Some of the unique models in SystemVue 2008 for analog and RF/channel impairments include:

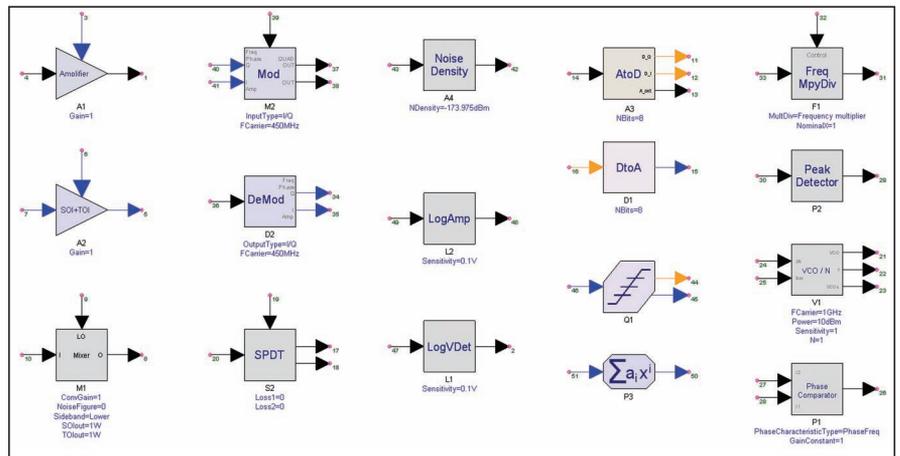
- RF and mixer models that support

both second- and third-order effects, allowing modeling of DC offsets in ZIF receiver architectures and image reject/alternate channel receiver degradation

- I/Q modulator and demodulators
- Complete library of RF communications and analog filters
- Data converters, quantizers and phase detectors

Figure 6 shows the collection of RF/IF/analog processing blocks that are standard with this package. Each of these blocks supports full envelope simulation of modulated signals and can impart bandpass impairments to the signal.

SystemVue 2008 is built on a new infrastructure that allows complete scripting of test instrument connectivity, allowing the program to link with the full array of Agilent instrumentation, including signal generators, signal analyzers, oscilloscopes, vector analyzers, logic analyzers and vector network analyzers. Seamless links to hardware and scripting ability allow the software to extend functional test to include performance metrics of BER/PER of fully coded systems, swept power/frequency characterization of receiver performance looking at digital output, and full algorithm implementation testing of designs in FPGAs and DSPs. SDR waveform ar-



▲ Fig. 6 RF/IF/analog processing blocks—standard with SystemVue 2008.

chitectures can now be verified at the HW prototype level and easily compared to early descriptions of waveform performance in simulation.

CONCLUSION

SystemVue 2008 brings together the needed design disciplines to accelerate the development of innovative heterogeneous SDR waveform architectures. By combining text-based code development methodologies with a GUI block editing environment, the new software can cut the time it takes to get ideas (algorithms) into real hardware. This brings real-world RF impairments into the hands of early ar-

chitects and algorithm developers, allowing for true high performance designs, and avoiding overdesign while reducing the chances of poor implementation choices.

SystemVue 2008 configurations start at US \$14,000 for full featured comms PHY focused capability that includes comms, DSP, logic and RF models, along with the fast simulator for modulated, multi-rate systems.

**Agilent EEsof EDA,
Santa Clara, CA 800-829-4444,
www.agilent.com/find/systemvue.**

RS No. 300

www.agilent.com

Product specifications and descriptions in this document subject to change without notice.

© Agilent Technologies, Inc. 2008
Printed in USA, November 25, 2008
5990-3301EN