Keysight Technologies Measurement Leadership in Modular

Modular Products Market Development Manager

April 2015

Robert Hood





Modular Advantages

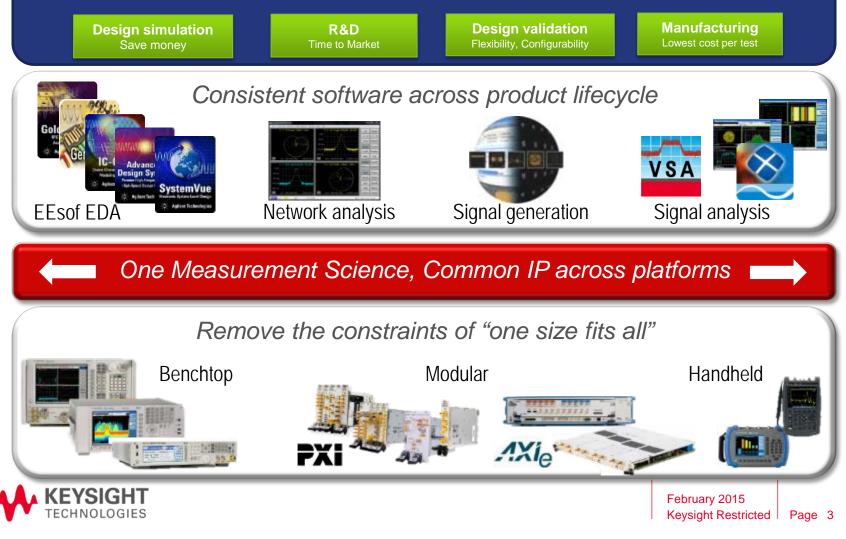




Keysight in Modular

Providing confidence in measurement results across development

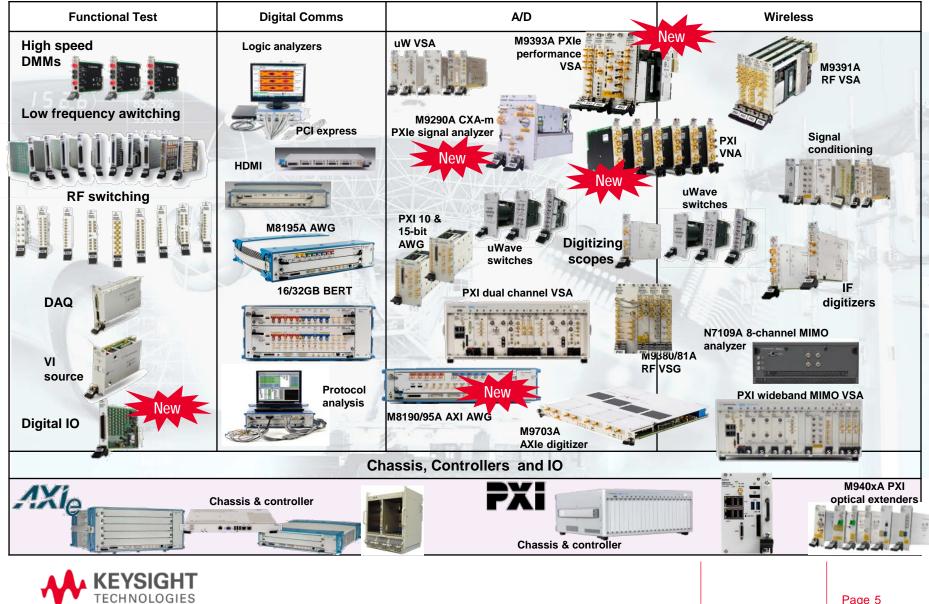
Keysight's extensive measurement expertise is applied across form factors, software, and solutions for design through manufacturing applications.



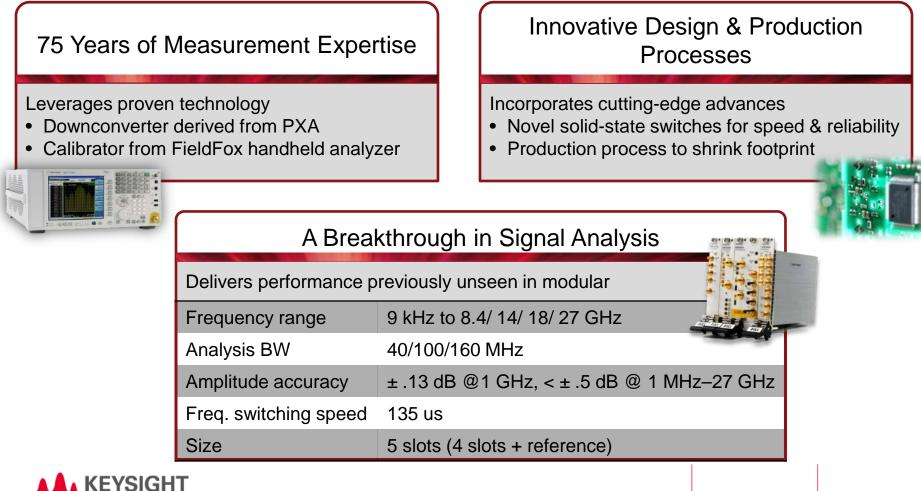




Keysight in Modular- A Growing Portfolio



The performance edge in PXI, M9393A VSA Validate true device performance at µW frequencies





CXA-m PXIe Signal Analyzer, M9290A Deploy a smaller microwave footprint up to 26.5 GHz

Get accurate, repeatable test results with best-in-class fully specified performance (*preliminary*)

- -163 dBm displayed average noise level
- -110 dBc/Hz phase noise (@ 1 GHz, 10 kHz offset)
- +17 dBm third-order intercept (TOI)

Count on trusted X-Series measurement science and calibration routines

- Analog demodulation, phase noise, noise figure
- Accelerate spur and harmonic searches with swept and FFT modes in any frequency span







PXIe Vector Network Analyzer, M937XA

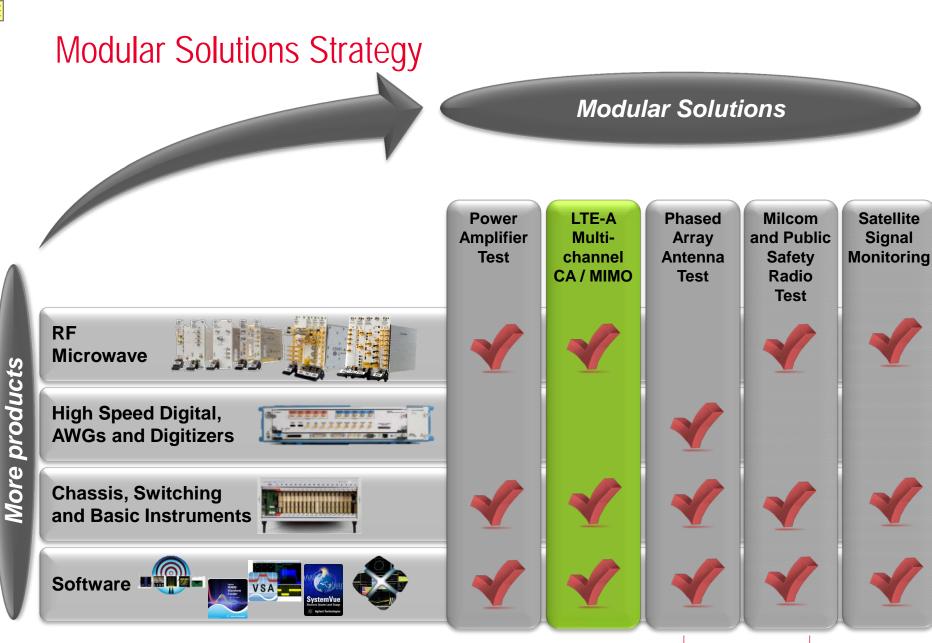
Driving down the size of test

- Full two-port network analyzer in just one slot
- Widest available frequency range:300 kHz to 4, 6.5, 9, 14, 20, 26.5 GHz
- Best PXI VNA performance in four key areas:
 - Speed: 28-33 msec across 401 points
 - Dynamic range: > 116 dB (9 GHz),
 > 98 dB (20 GHz)
 - Trace noise: < 0.001 dB
 - Stability: 0.005 dB/°C



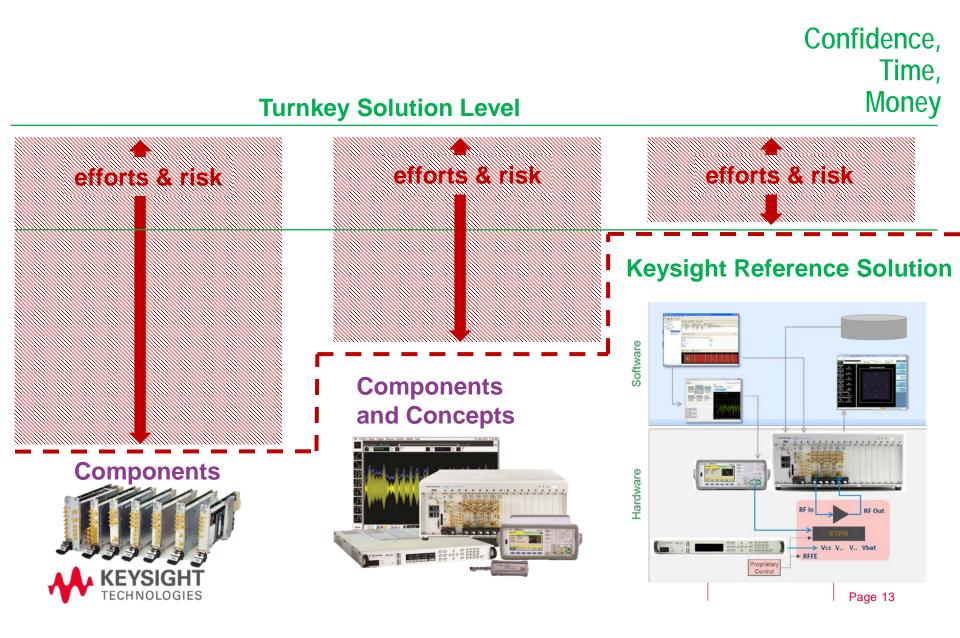




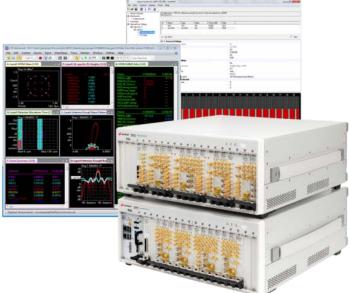




The Value of a Reference Solution



LTE-A Multi-Channel Reference Solution M9381A PXIe Vector Signal Generator M9391A PXIe Vector Signal Analyzer

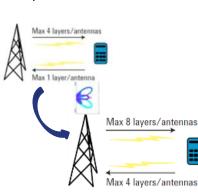


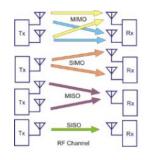


Multi-Channel Trends and Challenges Number of antennas per device increasing

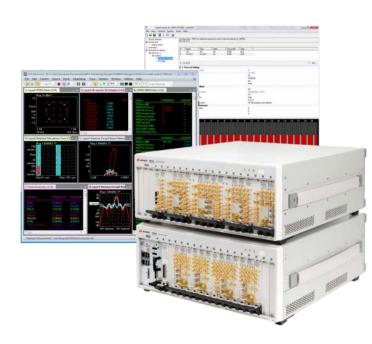
- More multi-channel techniques being employed:
 - Diversity, MIMO (spatial multiplexing), Beamforming, Carrier Aggregation
 - To drive:
 - Higher data rates
 - More capacity (number of users)
 - Quality of service
- Application examples:
 - Commercial wireless communications like LTE-Advanced, 802.11ac
 WLAN and emerging 5G research
- Test challenges:
 - Ability to visualize and verify complex MIMO and beamforming signals
 - Complex higher order MIMO configurations
 - Physical size and cost of test







LTE-Advanced Multi-Channel Reference Solution Accelerate LTE-A designs and gain deeper insight faster



Reference Solution includes:

- M9391A PXIe VSA(s)
- M9381A PXIe VSG(s)
- 89600 VSA software
- Signal Studio software
- Multi-channel configuration & correction utilities

- Easily configure complex MIMO systems with LTE/LTE-Advanced signal generation or analysis software
- MIMO toolkit accelerates multi-channel hardware synchronization and software setup
- Correction utility for accurate time & phase measurements at the device-under-test
- 2x2, 4x4 or 8x8 MIMO for spatial multiplexing/MIMO/ beamforming and carrier aggregation applications
 - < 1 ns time synchronization between channels ensures accurate chan-chan timing analysis
 - < 1 deg phase jitter for precise generation and analysis of beamforming patterns
 - Carrier aggregation with independently tuned channels for inter-band & up to 160 MHz bandwidth for intra-band support
 - Simultaneous UL & DL measurements

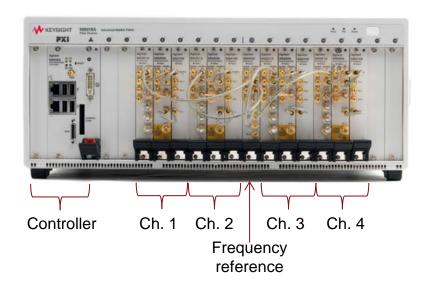


Easily Scale to Multi-Channel Instruments Up to 8-channel, phase-coherent PXIe VSG & VSA

Target Performance Channel-to-Channel Synchronization		
Timing jitter	≤ 1 ns	
Timing repeatability	≤ 1 ns	
Phase jitter	≤ 1°	
Phase repeatability	≤ 1°	

Configuration:

- Fit a 4-ch VSA or VSG in 1 chassis
- Multi-chassis configurations for up to 8 ch.
- 1 M9300A freq. reference per chassis
- 1 controller for entire system
- M9021A PCIe interface modules & cables to connect chassis
- Optional phase coherency capability









Simplifying System Setup

Multi-channel configuration and correction utilities

- Automatically set up backplane triggers and hardware configurations and aliases
- Verify correct software revisions
- Perform self test of MIMO hardware
- Automated correction routine time and phase alignment between channels

WIMO Config Ubility (ver. 1.2.0) Missing 1.3 (a) 3.5 Properties Actions Help Status Acto-Config Ubility (ver. 1.2.0) Properties Action is Help Missing 1.3 (a) 3.5 (b) 45(3) 14(1) Status //5et up sync ref 10 for M3300A1_chassing_islot10 M3300A1_chassing_islot2_sint1 (b) 45(3) 14(1) M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M3300A1_chassing_islot10 M330A1_chassing_islot10 M3300A1_chassing_islot10 M330A1_chassing_islot10 M3300A1_chassing_islot10 M330A1_chassing_islot11_13 //Set up sync ref 10 for M3300A1_chassing_islot10 M330A1_chassing_islot25 M330A1_chassing_islot10 M330A1_chassing_islot2 //Set up sync ref 10 for M3300A1_chassing_islot10 M330A1_chassing_islot35 //Set up sync ref 10 for M3300A1_chassing_islot35 (b) 40(2) 40(2) 5 //Set up s	 Generate source code 		Select a VSA System Master
MIMO Config Utility (ver. 1.2.0) Properties Actions Help Status Auto-Config Output Self-Test Output Code Snippet Properties Image: Configure Auto-Config Output Self-Test Output Code Snippet Properties Image: Configure Auto-Config Output Self-Test Output Code Snippet Presessel_select what devices are part of a MIMO configuration, and what devices will run independently. Image: Configure Auto-Config Output Self-Test Output Code Snippet Presessel_select what devices are part of a MIMO Devices? Image: Configure M300A1_chassis1_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3091A1_chassis1_slot10_slot1_S17 Image: Mage:			
Properties Actions Help Status Configure //Set up sync ref 10 for M3300A1_chassis1_slot10 M3300A1_chassis1_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis1_slot15_17 M3300A1_chassis2_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot15_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot15_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot15_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot15_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot1_13 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot1_5_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3331A1_chassis2_slot1_5_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3381A2_chassis3_slot1_5_17 M3300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M3381A2_chassis3_slot1_1_14 //Set up sync ref 10 for M3300A1_chassis1_slot10 M3381A2_chassis3_slot1_1_14 //Set up sync ref 10 for M300A1_chassis1_slot10 M3381A2_chassis3_slot1_slot_6_5_9 M3300A1_chassis		🖳 Select MIMO Devices 🔤	
Status Auto-Config Dutput Self-Test Dutput Code Snippet Configure //Set up sync ref 10 for M9300A1_chassis1_slot10 M9300A1_chassis1_slot10.ReferenceB ase2.BackPlaneReferenceEnabled = true; M9391A3_chassis1_slot11_13 M9300A1_chassis2_slot10.ReferenceB ase2.BackPlaneReferenceEnabled = true; M9391A3_chassis1_slot15_17 M9300A1_chassis2_slot10.pdpl(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis2_slot10.pdpl(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis2_slot10.pdpl(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceB ase2.BackPlaneReferenceEnabled = true; M9391A4_chassis2_slot5_8 M9300A1_chassis4_slot10.Appl(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.pdpl(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.Appl(); //Set up sync ref 10 for M9300A1_chassis1_sigueEnumAppl(); //Set up sync	🖳 MIMO Config Utility (ver. 1.2.0)		
Configure //Set up sync ref 10 for M300A1_chassis1_slot10 M300A1_chassis1_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis2_slot10 M300A1_chassis2_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis3_slot10 M300A1_chassis2_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis3_slot10 M300A1_chassis2_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis3_slot10 M300A1_chassis3_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis3_slot10 M300A1_chassis3_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis3_slot10 M300A1_chassis3_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis4_slot10 M300A1_chassis3_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis4_slot10 M300A1_chassis3_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis4_slot10 M300A1_chassis4_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis1_slot13 //Backplane trigger routing for chassis1 (M300A1_chassis4_slot10.Apply[; //Set up sync ref 10 for M300A1_chassis1_slot2 //Backplane trigger routing for chassis1 (M300A1_chassis4_slot10.Apply[; //Backplane trigger routing for chassis1	Properties Actions Help	Which are MIMD Devices?	
Image: Commit // 3900A1_chassis1_slot10.ReferenceEase2.BackPlaneReferenceEnabled = true; M9300A1_chassis1_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis2_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis2_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis2_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis1_slot10 M9300A1_chassis3_slot0.Apply(); //Set up sync ref 10 for M9300A1_chassis1_slot10 M9300A1_chassis4_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis1_rigBus1ro2To3); chassis1_riggerBus_connect(Status Auto-Config Output Self-Test Output Code Snippet	M9391A1_chassis1_slot3_5	
//Set up sync ref 10 for M9300A1_chassis2_slot10 M9300A1_chassis2_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis2_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis3_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis3_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.Apply(); //Backplane trigger routing for chassis1 chassis1.TriggerBus_Connect(1, AgM9018TrigBus1To2To3); chassis1.TriggerBus_Connect(3, AgM9018TrigBus1To2To3); chassis1.TriggerBus_Connect(3, AgM9018TrigBus1To2To3); chassis1.TriggerBus_Connect(3, AgM9018TrigBus1To2To3); chassis1.TriggerBus_Connect(3, AgM9018TrigBus2To1); chassis1.TriggerBus_Connect(3, AgM9018TrigBus2To1); chassis1.TriggerBus_Connect(3, AgM9018TrigBus2To1); chassis1.TriggerBus_Connect(3, AgM9018TrigBus2To1); <t< td=""><td>M9300A1_chassis1_slot10.Apply();</td><td>W9391A3_chassis1_slot11_13</td><td></td></t<>	M9300A1_chassis1_slot10.Apply();	W9391A3_chassis1_slot11_13	
<pre>//Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis3_slot10.Apply(); //Set up sync ref 10 for M9300A1_chassis4_slot10 M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis4_slot10.Apply(); //Backplane trigger routing for chassis1 chassis1_TriggerBus_Connect(1, AgM9018TrigBusEnum.AgM9018TrigBus1To2To3); chassis1_TriggerBus_Connect(3, AgM9018TrigBusEnum.AgM9018TrigBus1To2To3); chassis1_TriggerBus_Connect(3, AgM9018TrigBusEnum.AgM9018TrigBus2To1);</pre>	//Set up sync ref 10 for M9300A1_chassis2_slot10 M9300A1_chassis2_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300A1_chassis2_slot10.Aerbit0;	 M9391A2_chassis2_slot6_8 M9391A3_chassis2_slot11_13 	E Cancel
M9300Å1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true; M9300Å1_chassis4_slot10.Apply(); OK Cancel //Backplane trigger routing for chassis1 chassis1.TriggerBus.Connect(1, AgM9018TrigBusEnum.AgM9018TrigBus1To2To3); chassis1.TriggerBus.Connect(3, AgM9018TrigBusEnum.AgM9018TrigBus2To1);	//Set up sync ref 10 for M9300A1_chassis3_slot10 M9300A1_chassis3_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true;	M9381A1_chassis3_slot2_5 M9381A2_chassis3_slot6_9	
Run Sample Waveform chassis1.TriggerBus.Connect(1, AgM9018TrigBusEnum.AgM9018TrigBus1To2To3); chassis1.TriggerBus.Connect(3, AgM9018TrigBusEnum.AgM9018TrigBus2To1);	M9300Å1_chassis4_slot10.ReferenceBase2.BackPlaneReferenceEnabled = true;		cel
	Run Sample Waveform chassis1. TriggerBus.Connect(1, AgM9018TrigBusEnum.AgM9018TrigBus1To2To3); chassis1. TriggerBus.Connect(3, AgM9018TrigBusEnum.AgM9018TrigBus2To1);	•	
KEYSIGHT February 2015 TECHNOLOGIES Keysight Restricted Page 18			3



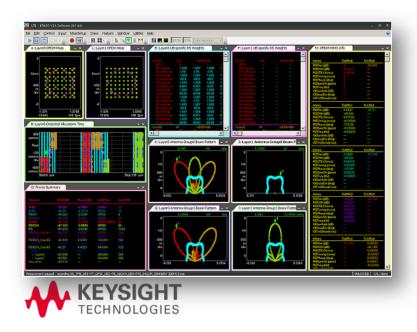
89600 VSA and WLA Software

Comprehensive multi-channel analysis



89600 VSA Software:

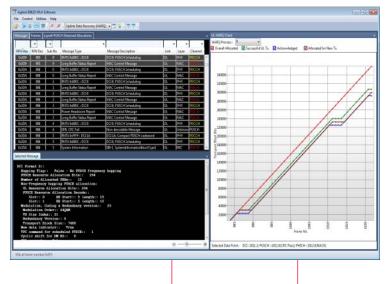
- Industry-leading with support for over 75 signal formats including MIMO, Beamforming
- In-depth RF troubleshooting tools
- Cross-channel analysis
- Analyze at any stage of design -- baseband to RF, simulation to design validation)





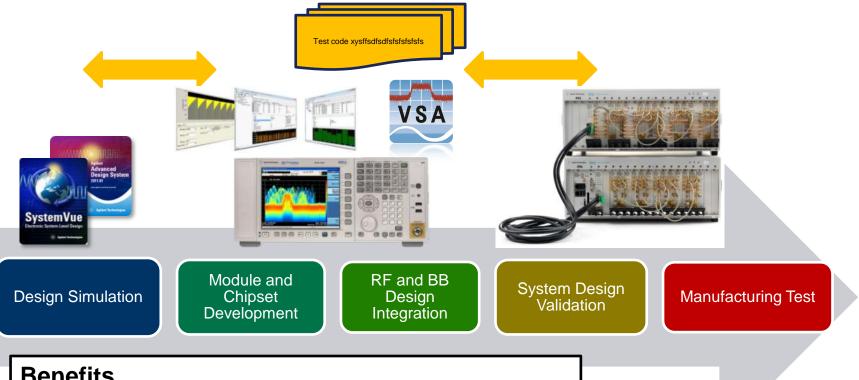
89600 WLA Software:

- MAC, RRC, RLC layer add-on to the 89600 VSA for UL & DL LTE FDD
- Enables multi-frame, multi-layer analysis
- LTE transmission modes 1-6 in DL





Leverage Across Platforms From design simulation to test



- **Benefits**
- Leverage from signal channel measurements
- Quickly validate multi-channel designs for emerging standards ٠
- Eliminate time spent implementing new standards ٠





For more information please visit

www.keysight.com/find/solution-lte www.keysight.com/find/pxi-mimo

Access to solution brochure, videos, configurations, application note



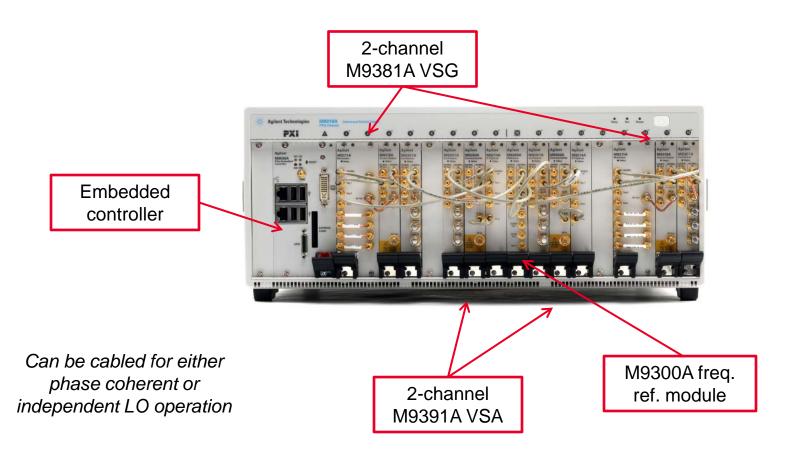
Backup



System Configurations

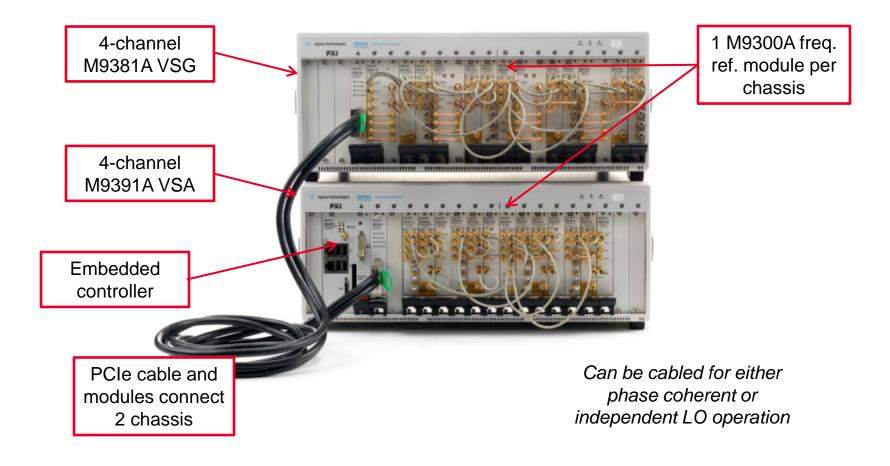


2x2 (2-channel source and 2-channel analyzer)



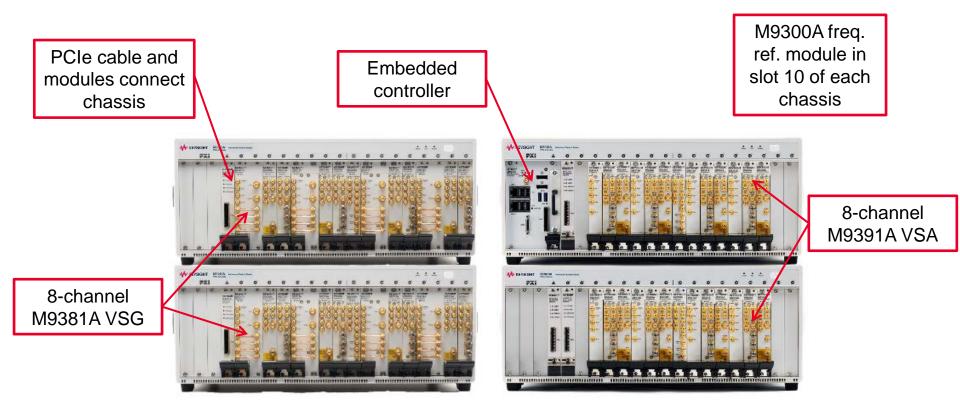


4x4 (4-channel source and 4-channel analyzer)





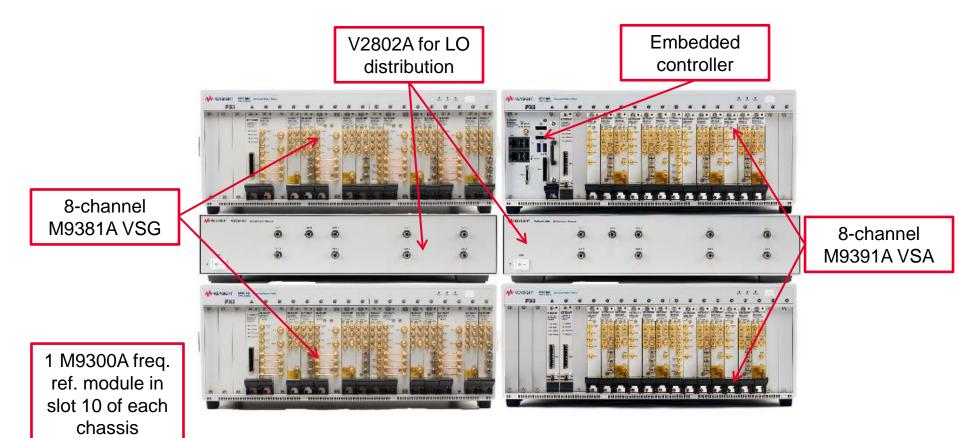
8-channel, time-synchronized source & analyzer



This 8x8 configuration is for independent LO operation for time synchronization only



8-channel, phase-coherent source & analyzer



This 8x8 configuration is for shared LO operation for phase coherency



Phase and Timing Calibration

Multi-channel corrections wizard

- Fixed time and phase skew need to be calibrated/corrected out prior to testing
 - Cables, temperature, module placement can also affect correction factor
 - Dependent on center frequency and span
- Wizard walks you through creating a corrections file step-by-step
- Corrections then available for application through wizard

