Real-Time, Software-Based Characterization of Receiver Dynamic Channel Performance on an SDR Development Platform

Ivan Aguayo

Office of the Chief Technologist

Coherent Logix, Inc.

Photograph Courtesy NASA

Coherent Logix™

Enabling Low Power Software Defined Systems™

Talk Organization

- **Introduction**
  - HyperX™ Technology
  - HyperX Hardware Application Development System (hxHADS™)
- **System Design**
  - Architecture and Components
  - Channel Emulator
- **Adaptive Testing**
  - How to achieve meaningful results as soon as possible.
- **Example Test Case with Results**
- **Future Work**

Overall Project Goal

- **Enhance user productivity for Radio Waveform Design, Development, and Implementation.**

- **Specifically, provide tools to aid the evaluation of**
  - Functionality
  - Time performance
  - Power performance
  - "Characterization" (domain specific)

  as early in the process as possible.

- **Enable evaluation *continually* throughout the process.**

Receiver Characterization

- **What is it? Performance evaluation of a radio receiver under static and dynamic environmental conditions.**

- **Allows the designer to determine how "good" the receiver is under certain channel conditions such as:**
  - Multipath fading
  - Doppler (mobility)
  - Noise and signal strength

- **This talk concentrates on the generation of bit error rate (BER) and packet error rate (PER) results.**

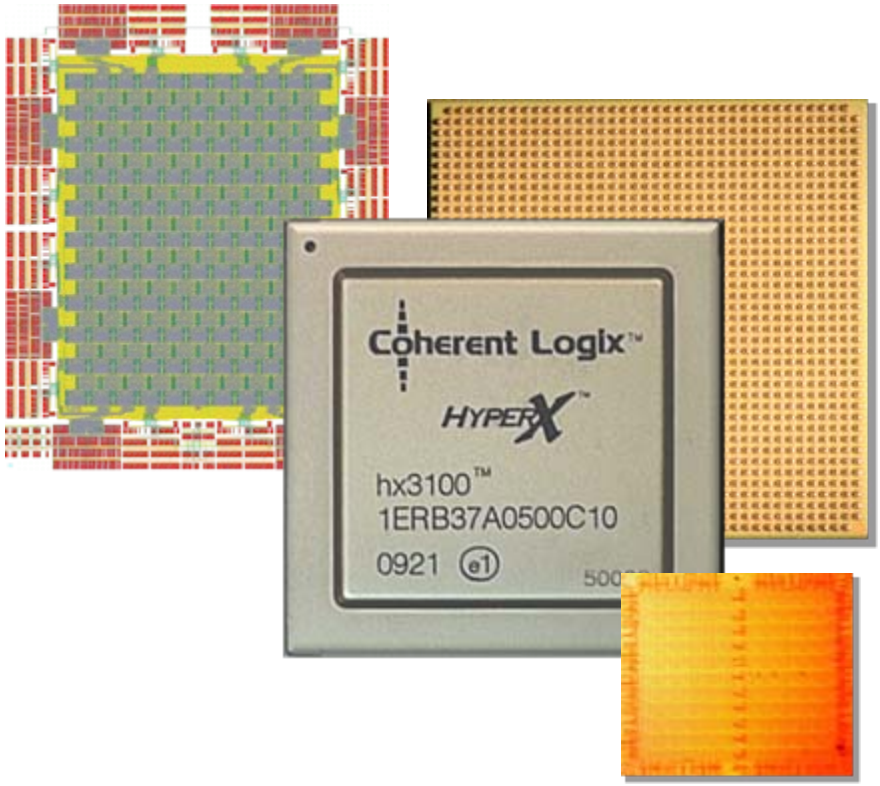
So What is the Problem?

- **Early design development can generally only be simulated.**
  - Statistically meaningful BER and PER results requires millions of tests.
  - Simulation proves to be too slow, requiring years of CPU time to get meaningful results.
  - Will eventually need to run the design/implementation on target hardware, where performance results may be different.

- **Use of dedicated hardware for Receiver Characterization**
  - It is costly.
  - Limited to the functionality and flexibility of the test hardware.
  - Requires a (semi) complete design to begin using the test hardware.

Our Approach

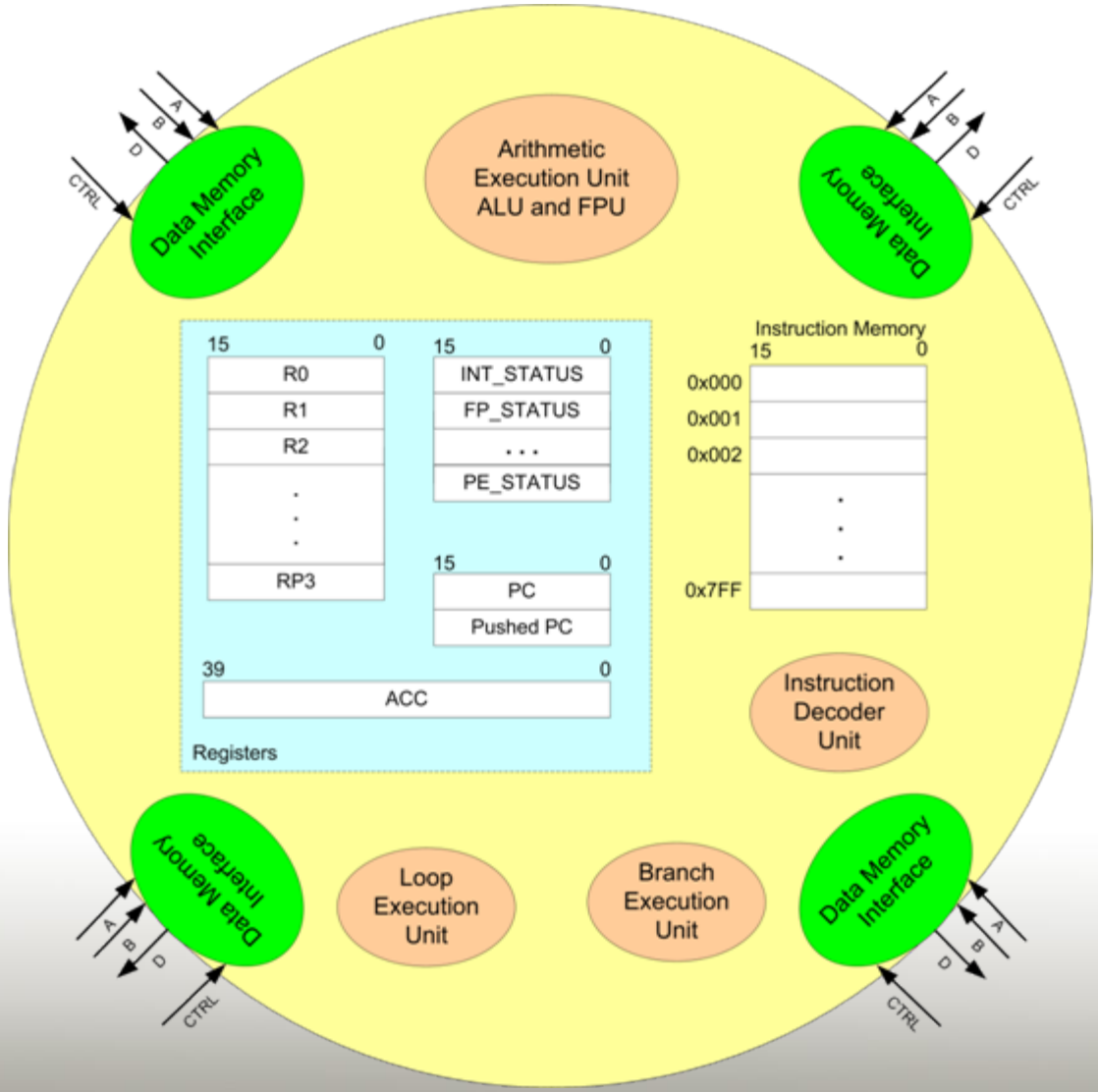
- **Build on the Coherent Logix HyperX technology.**
- **Provide a software-based, integrated tool flow for maximum flexibility.**
- **Support rapid prototyping with the hxHADS development system.**
- **Organize all components into a complete package, the RWDS (Radio Waveform Development System).**
- **Focus this talk on one component of the RWDS, the real-time, characterization of receiver dynamic channel performance.**



HyperX Technology

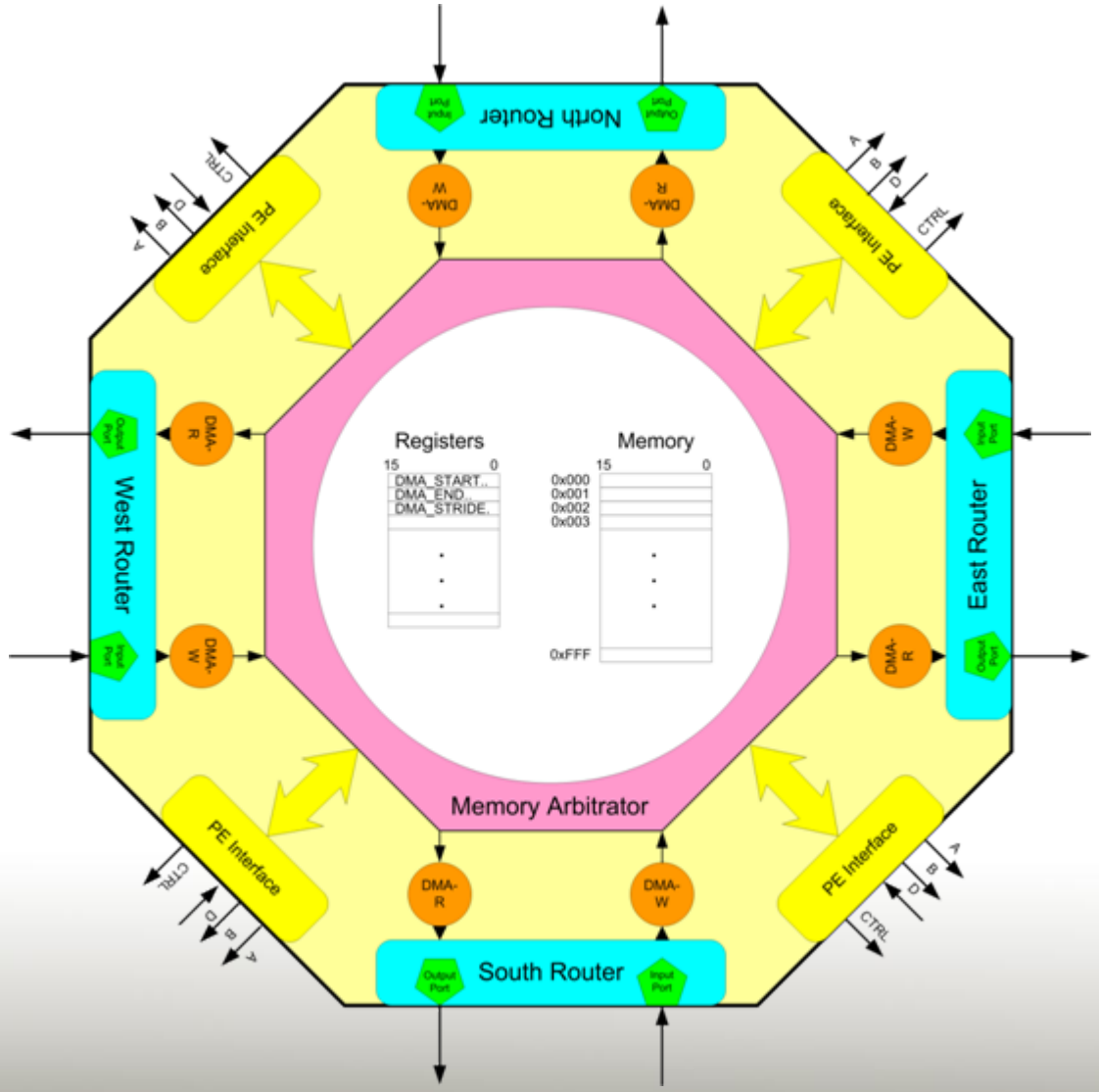
HyperX Architecture: Processing Element (PE)

- **Registers**
  - 16x 16-bit GP
  - 40-bit Accumulator
  - 3x 16-bit Pointer
  - 3x 16-bit Index
- **16-bit ALU**
- **24-bit Multiplier**
- **32-bit Floating Point**
  - Add, Subtract, Multiply, Reciprocal, Reciprocal Square Root
- **40-bit Barrel Shifter**
- **SIMD (2x 8-bit)**
- **Variable length instructions**
- **Zero-overhead looping**
- **Conditional move**

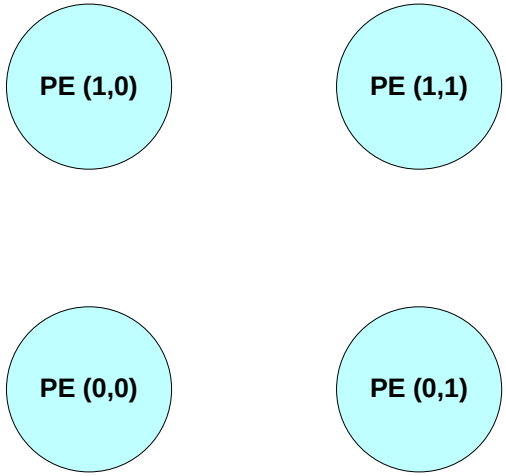


HyperX Architecture: Data Memory and Routing Unit (DMR)

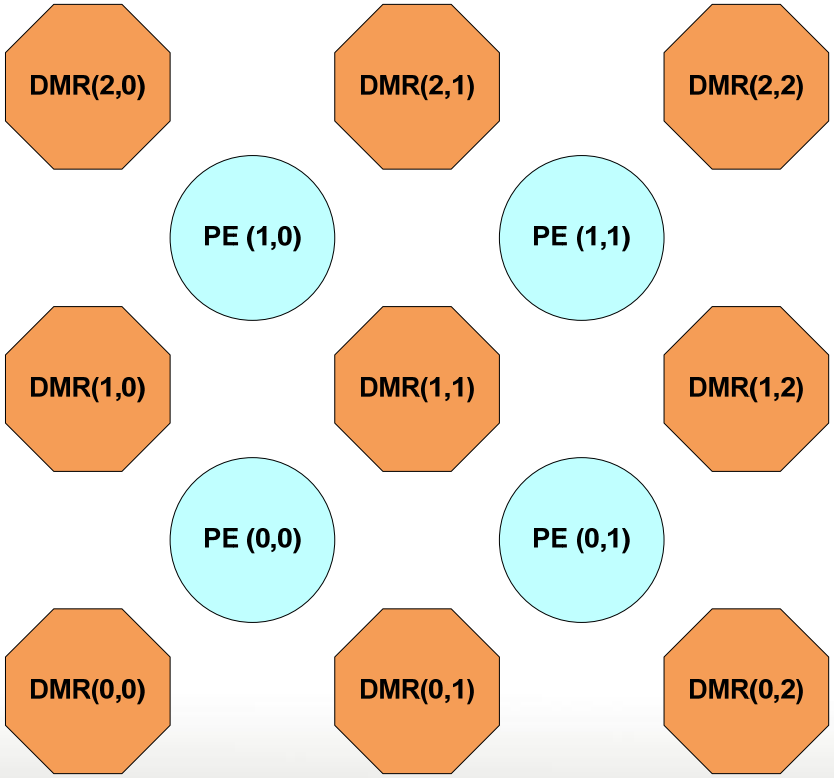
- Memory accessible by neighboring PEs
- Network connections to neighboring DMRs
- Software controlled access priority
- 20 simultaneous data operations possible
- Data transmission independent of PE activity
- Blocking and non-blocking transfers



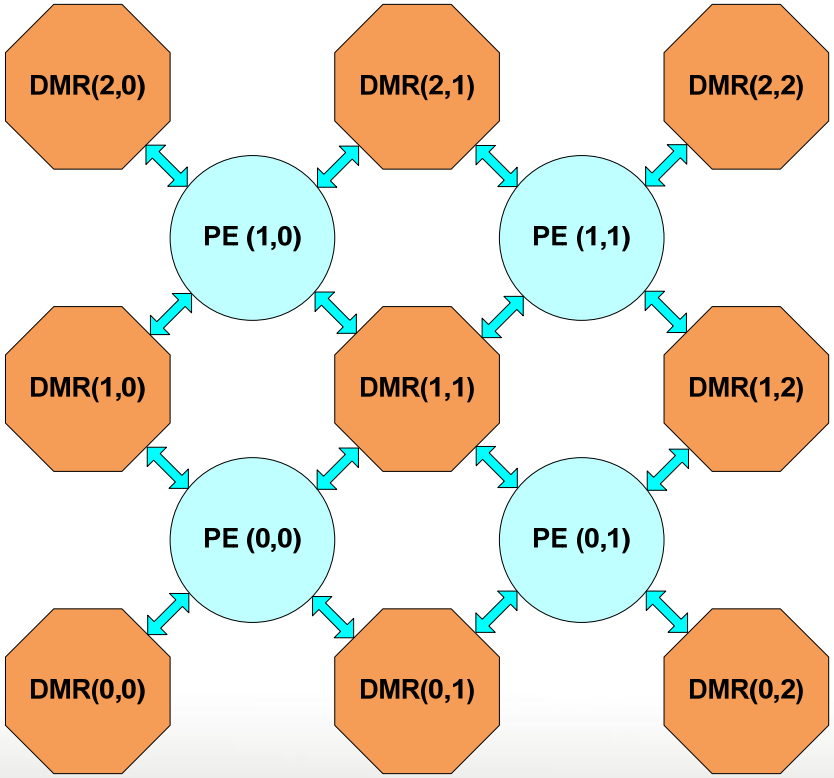
HyperX Architecture: PE Array



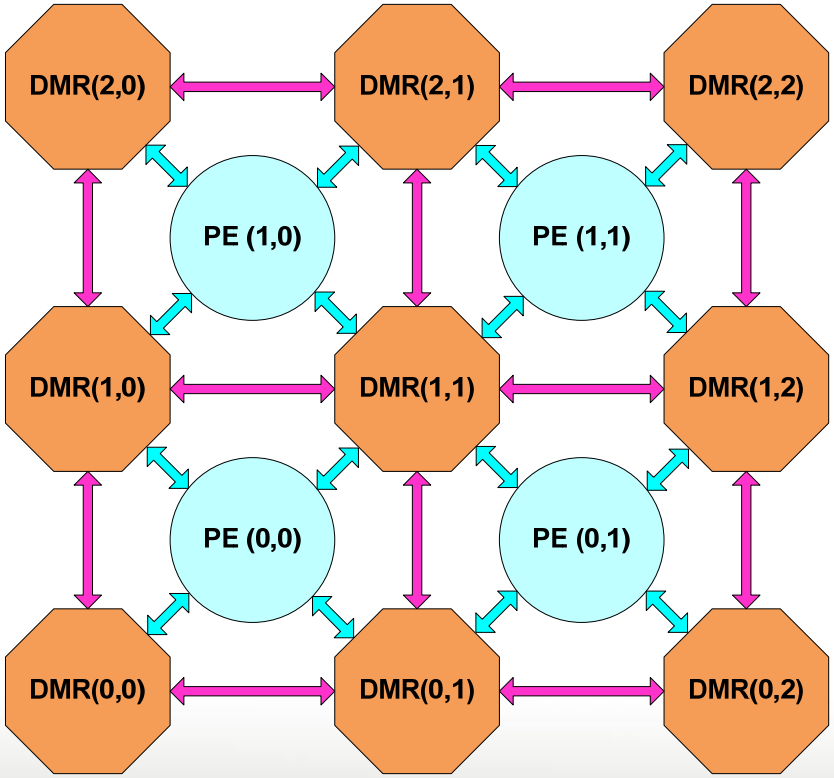
HyperX Architecture: PE / DMR Array



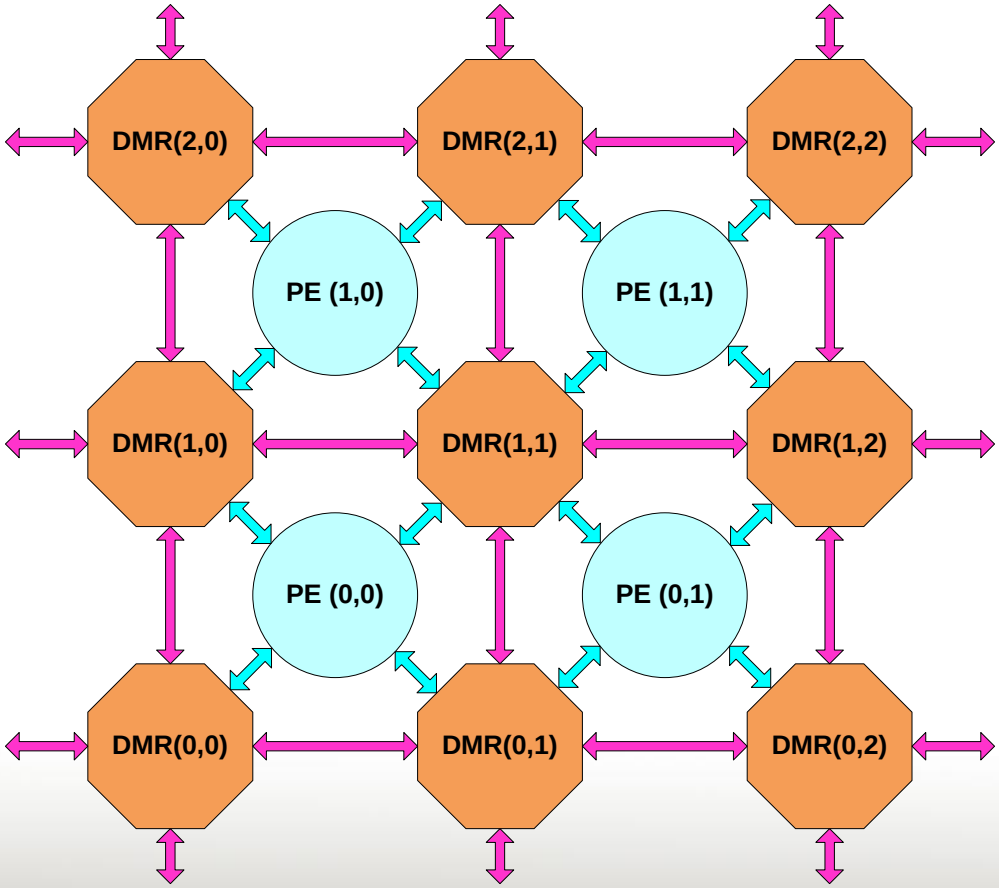
HyperX Architecture: PE / DMR Communications



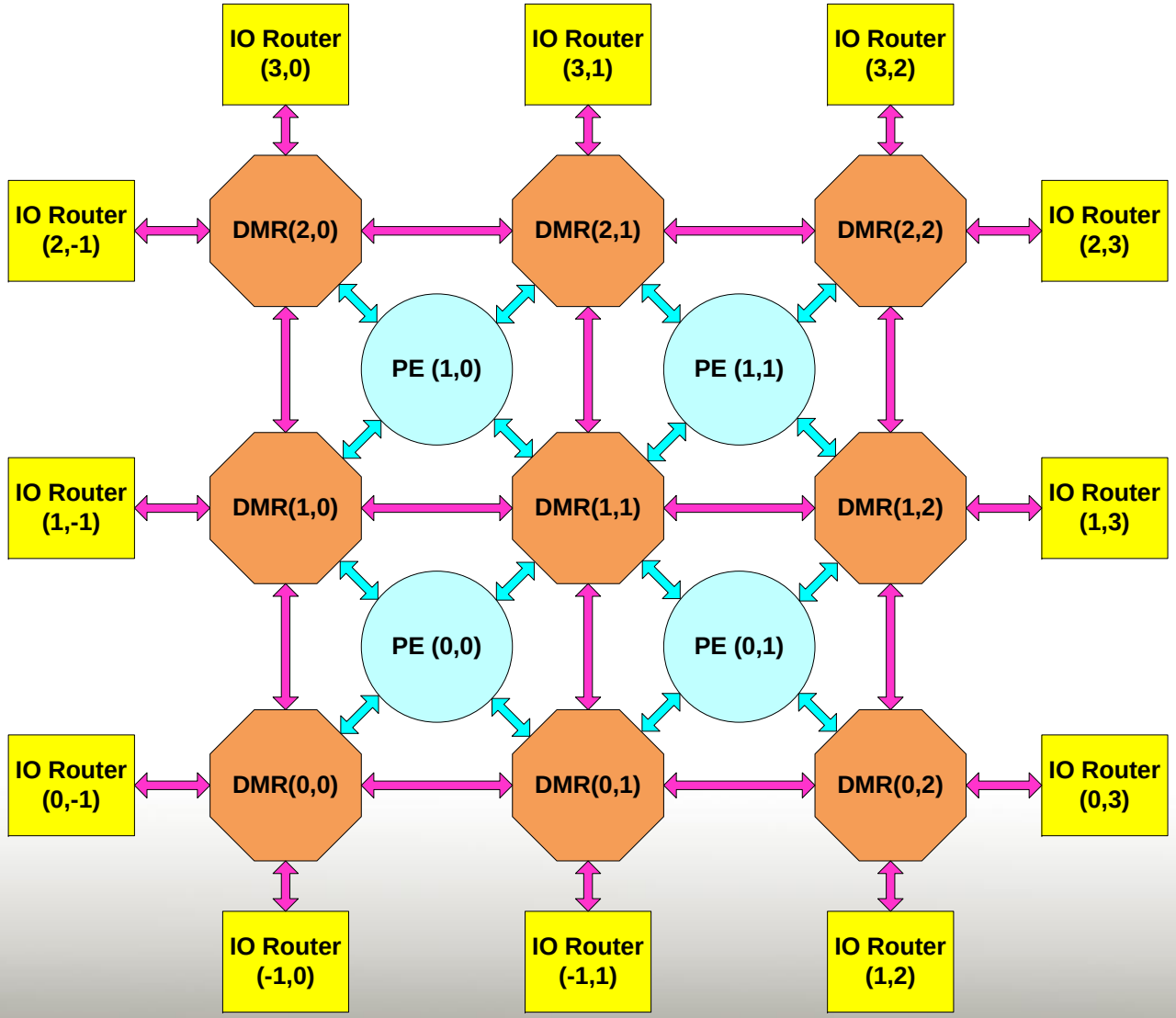
HyperX Architecture: PE / DMR Fabric



HyperX Architecture: Fabric External Connections

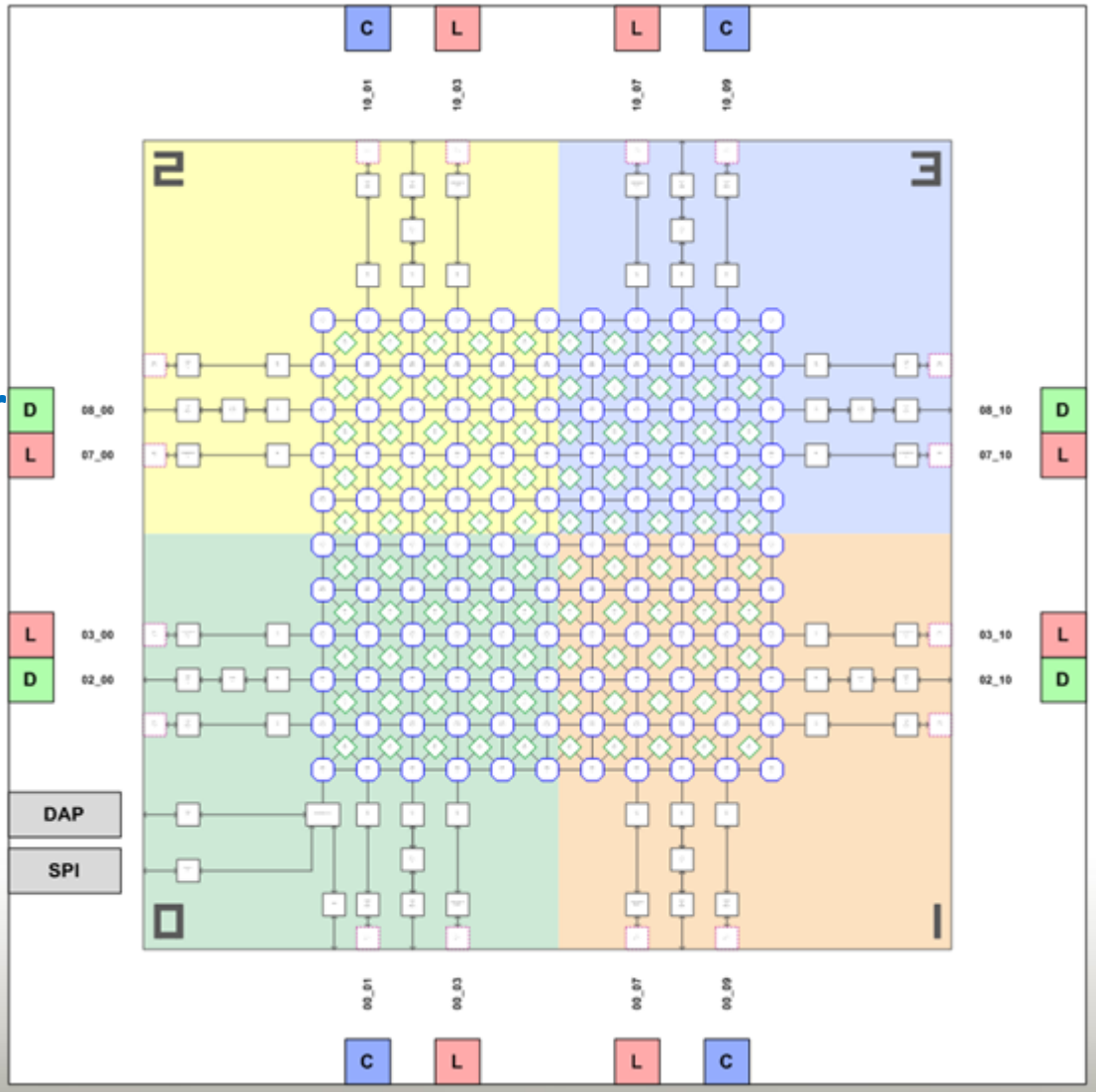


HyperX Architecture: Fabric including HyperIO™



HyperX Architecture: hx3100 (*"Generic" Package*)

- **Massively Parallel**
  - 100 PEs (10x10 grid)
  - 121 DMRs (11x11 grid)
- **Four Independent Power Quadrants**
- **Data Ports (bonded out)**
  - 8x LVDS, 4x CMOS
  - 4x DDR2
- **Control Ports**
  - Boot / Debug / JTAG



HyperX Hardware Application Development System

hxHADS

HyperX Hardware Application Development System

- **Open hardware development platform to facilitate full hardware and software system development**

- **Provide a clear path from an engineering hardware and software development platform …**

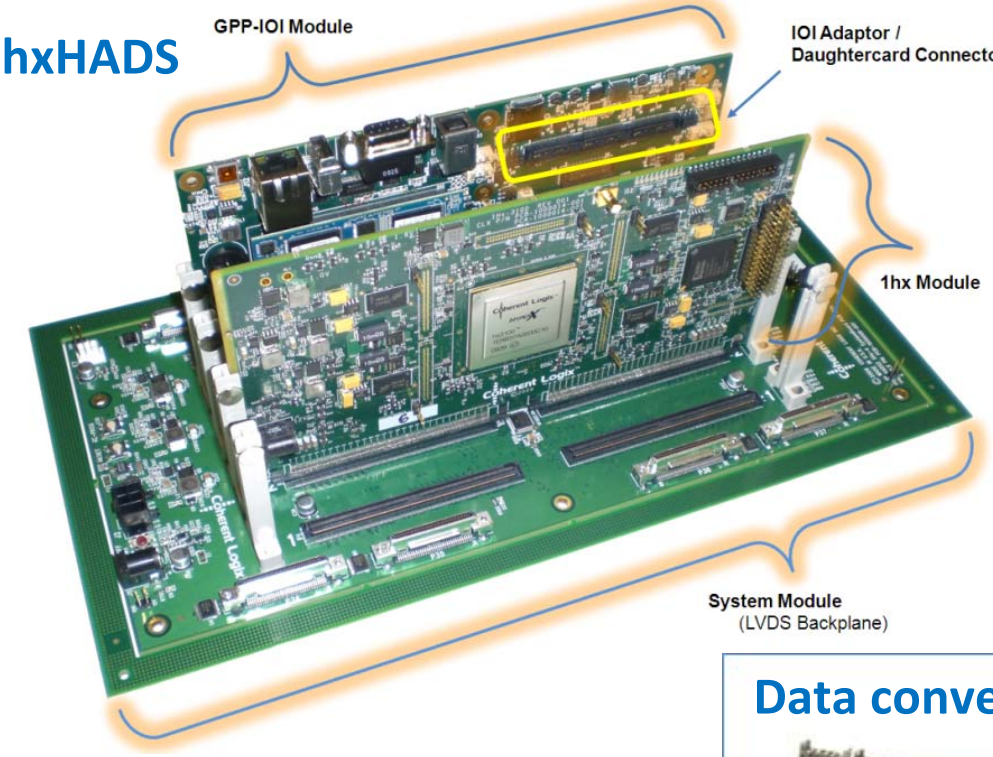
  … to …

  … final form factor product

  – Preserve full software stack “as is”
  – Preserve hardware architecture, remove unnecessary glue logic
  – Provide real-time analysis, characterization, instrumentation, etc.

HyperX Hardware Application Development System

hxHADS

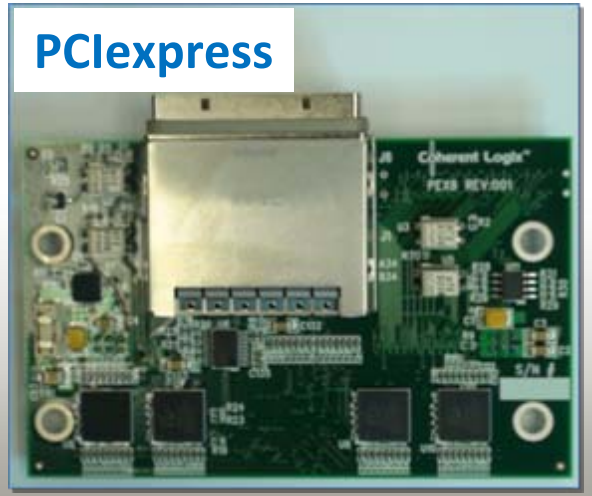


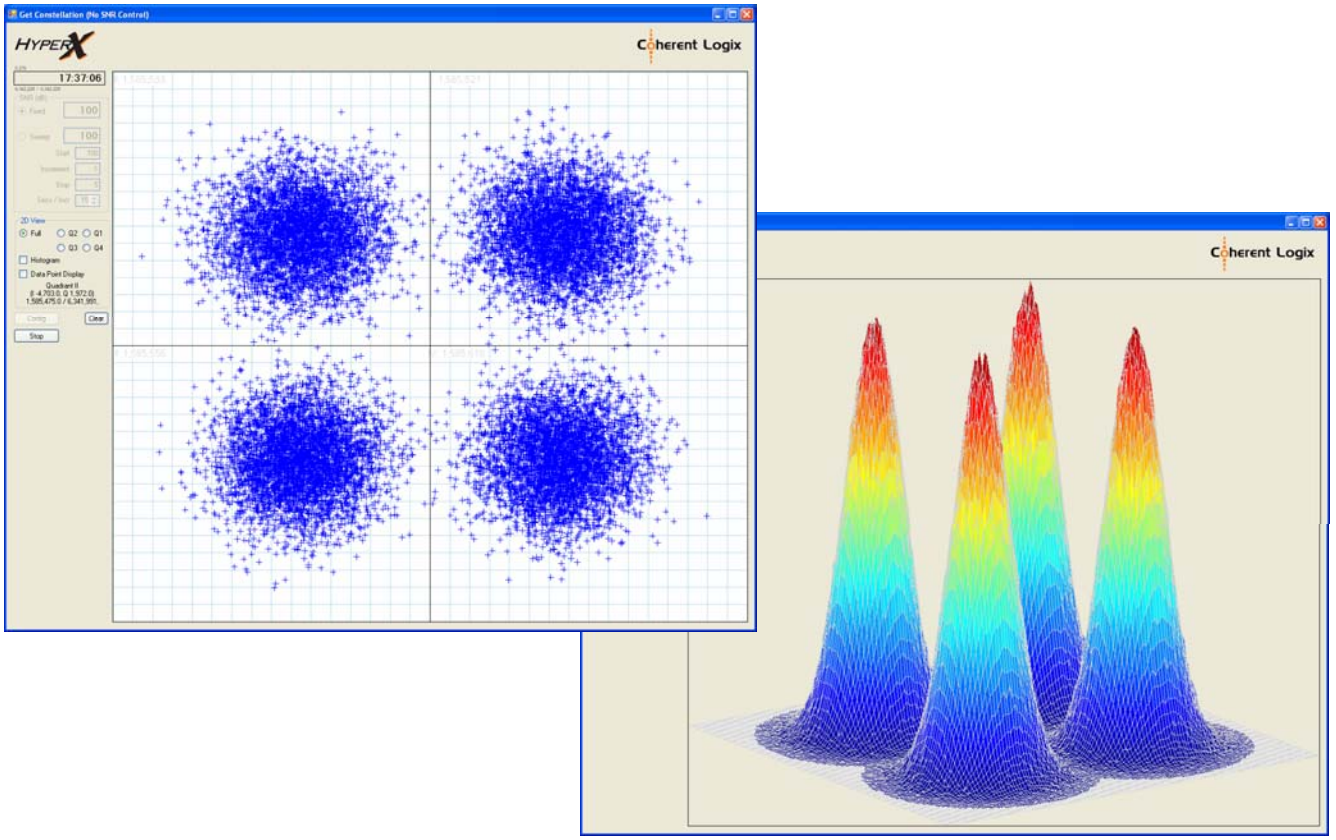
- Fully modular and customizable
- Plug-and-play capability with the HyperX ISDE
- Clear path to form factor product
- Supports user-developed cards

RF front ends

Data conversion

PCIexpress





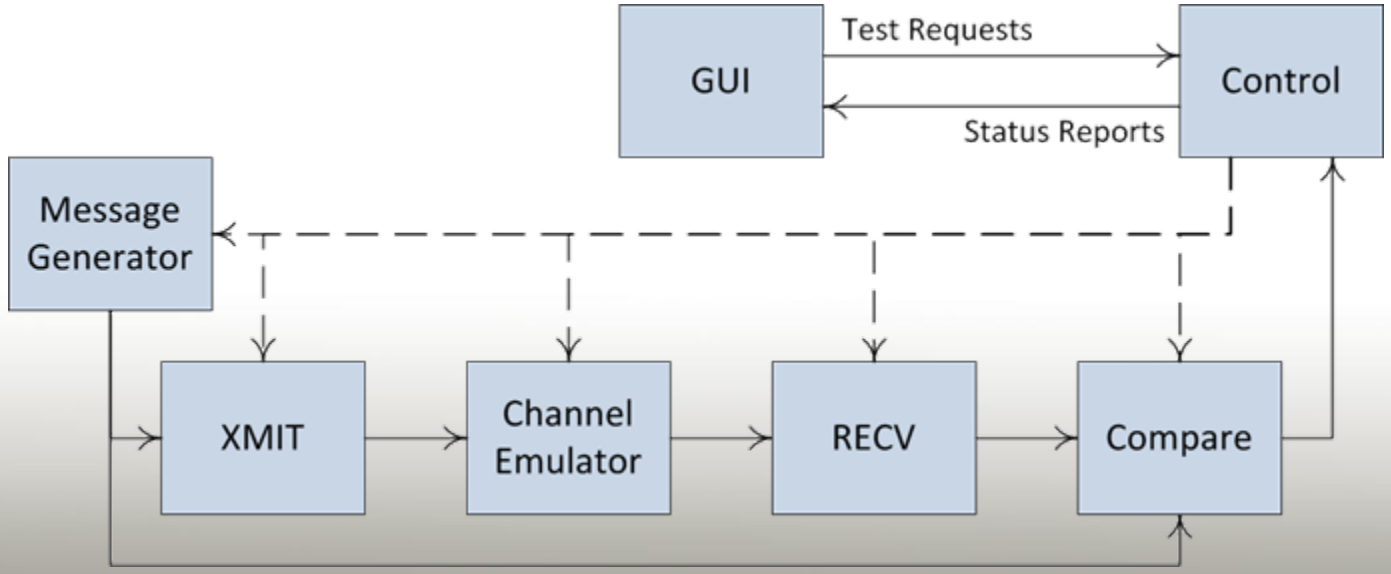
System Design

Architecture and Components

System Design Overview

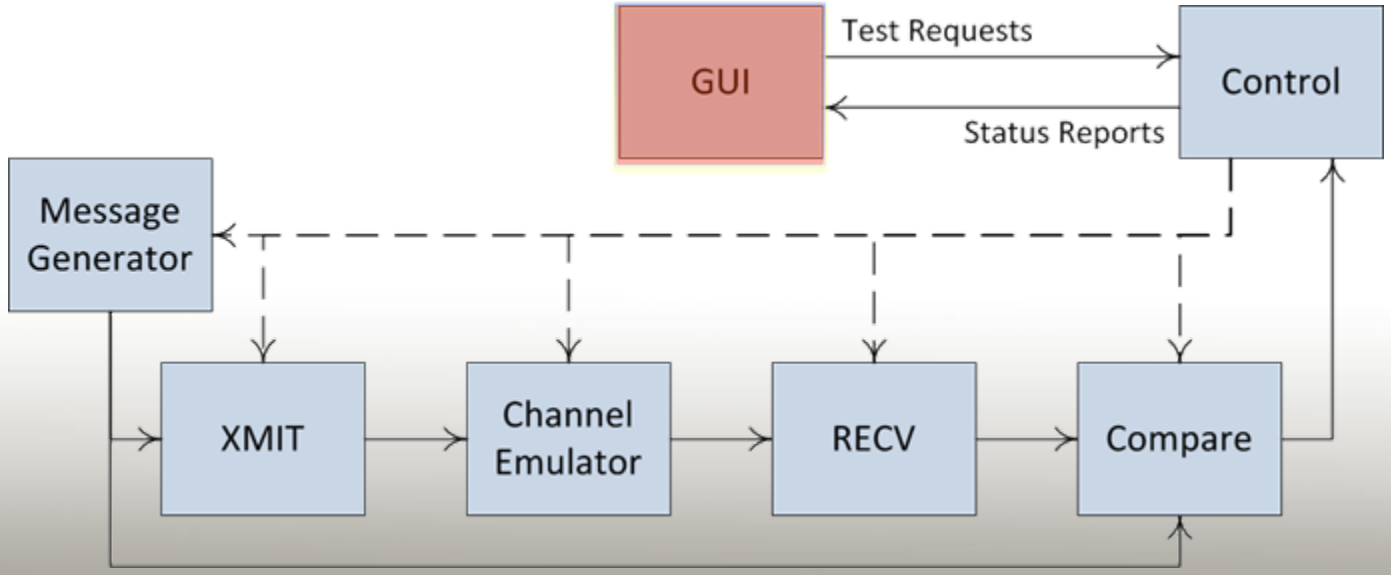
- **Architecture composed of closed-loop setup with control to all functional.**

- **GUI to provide test setup interface and feedback.**



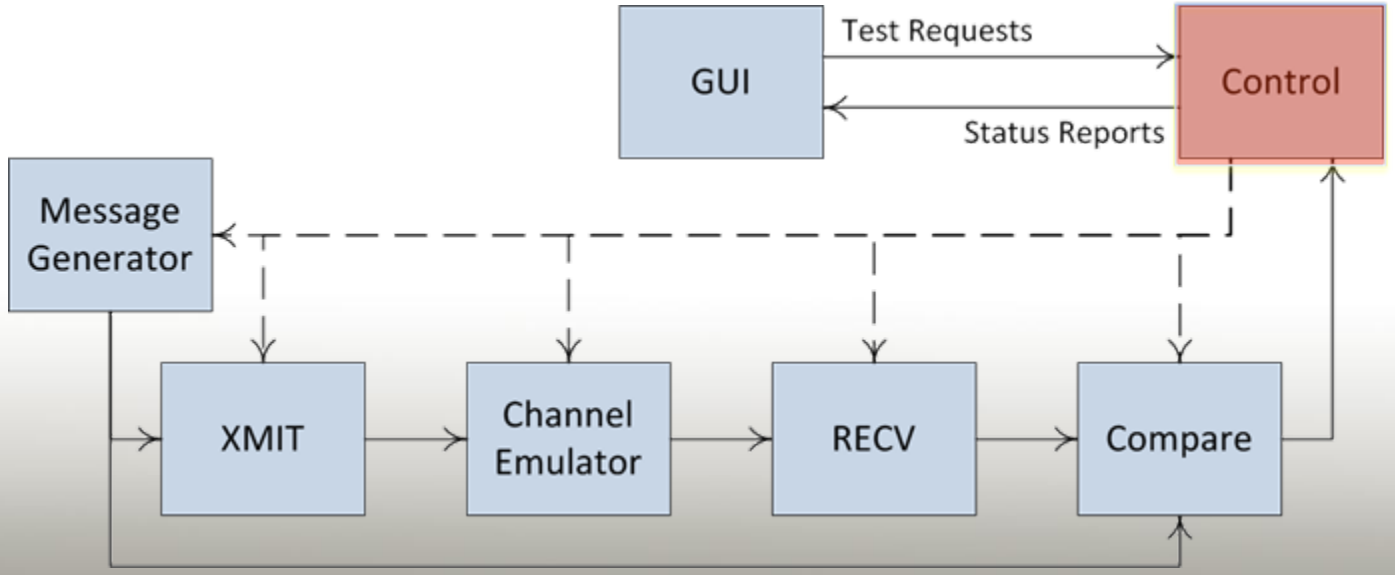
System Design - GUI

- **Accepts characterization scenario definitions from the user.**
- **Constructs packets to send test requests to the control unit.**
- **Logs and maintains statistics on the test results.**
- **Presents graphical feedback to the user.**



System Design – Control Unit

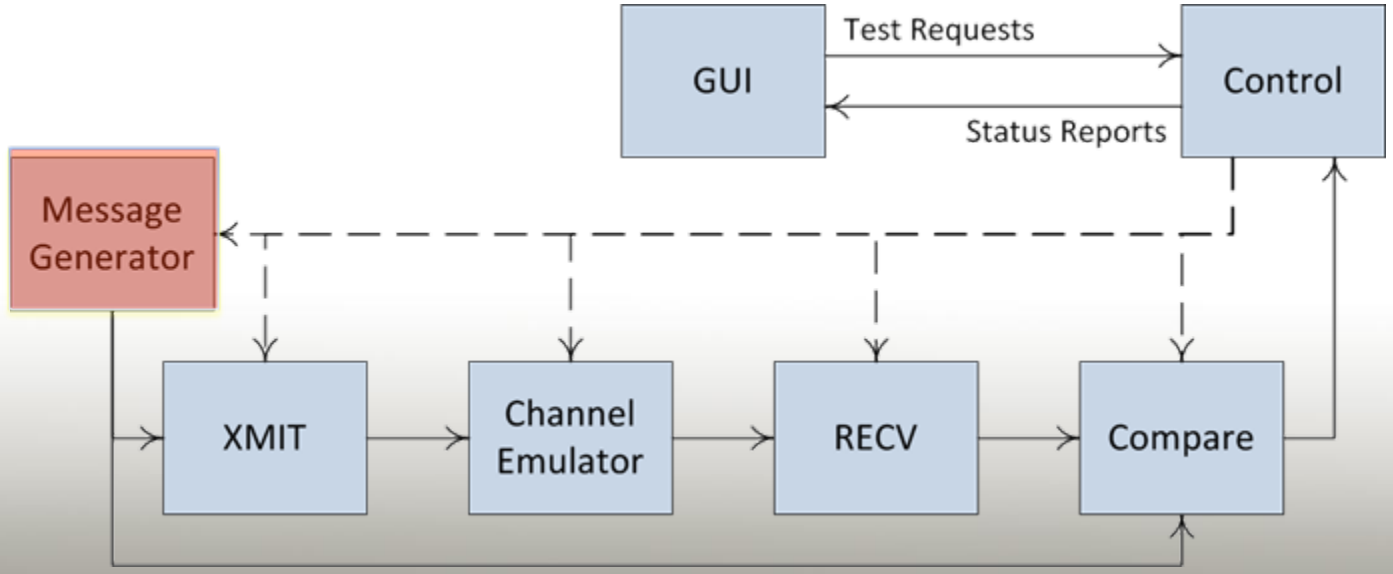
- **Accepts test requests from the GUI.**
- **Generates the configuration information appropriate to carry out the test for the other units of the characterization environment (dashed lines).**
- **Collects the result of the test and returns status information to the GUI.**



System Design – Message Generator

- **Creates random content data messages of the requested size and appropriate format.**

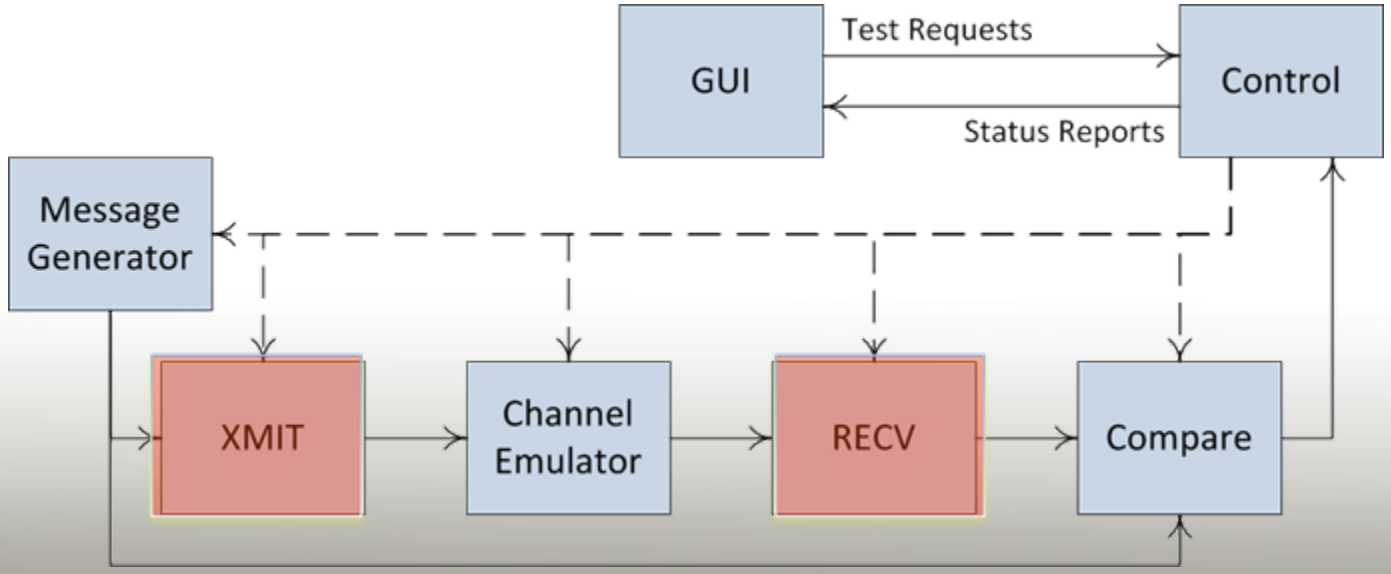
- **Use of long-period Tausworthe generators to produce uniform random numbers (2^115 bytes periodicity).**



System Design – XMIT and RECV Units

- **The XMIT and RECV units implement the transmitter / receiver pair being characterized.**

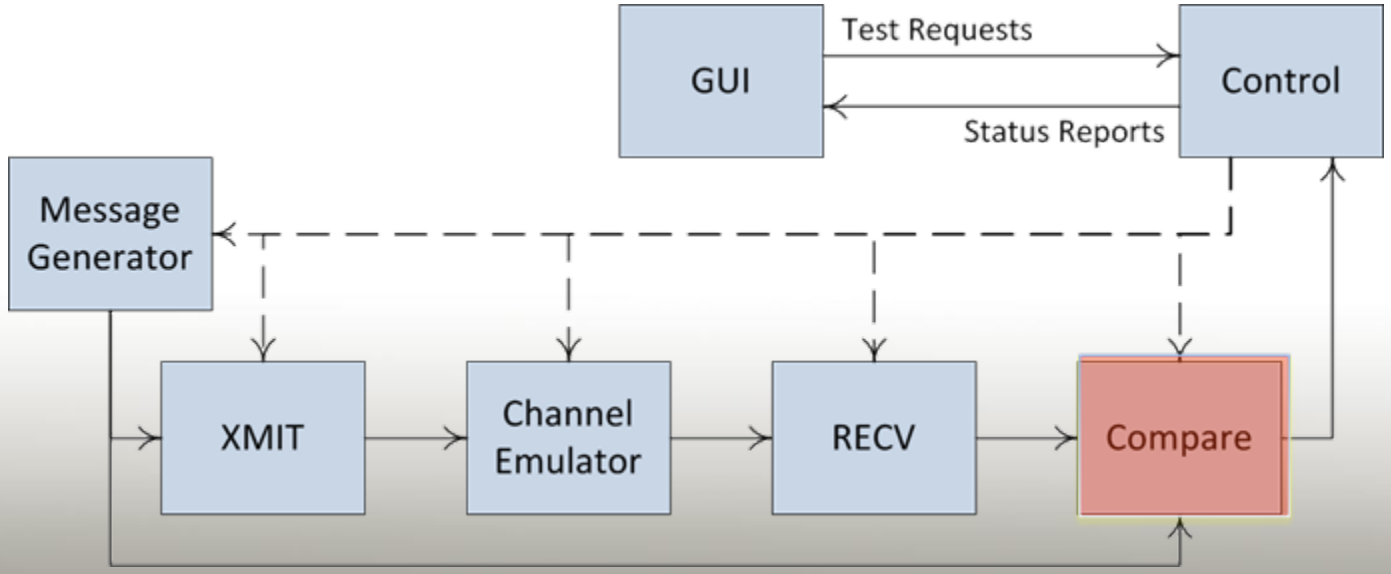
- **The XMIT and RECV units can grow in complexity as they are being developed, without modifying the rest of the design, allowing real-time characterization at an early stage.**



System Design – Compare Unit

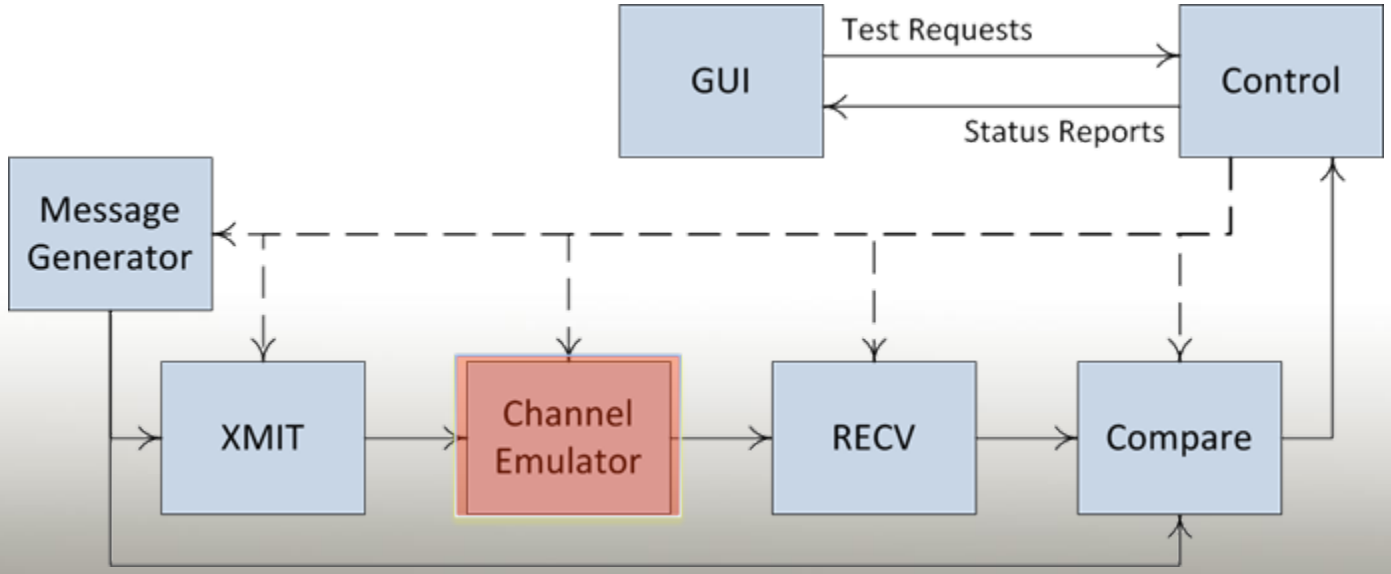
- **Compares the original message with the received/decoded message and tabulates discrepancies.**

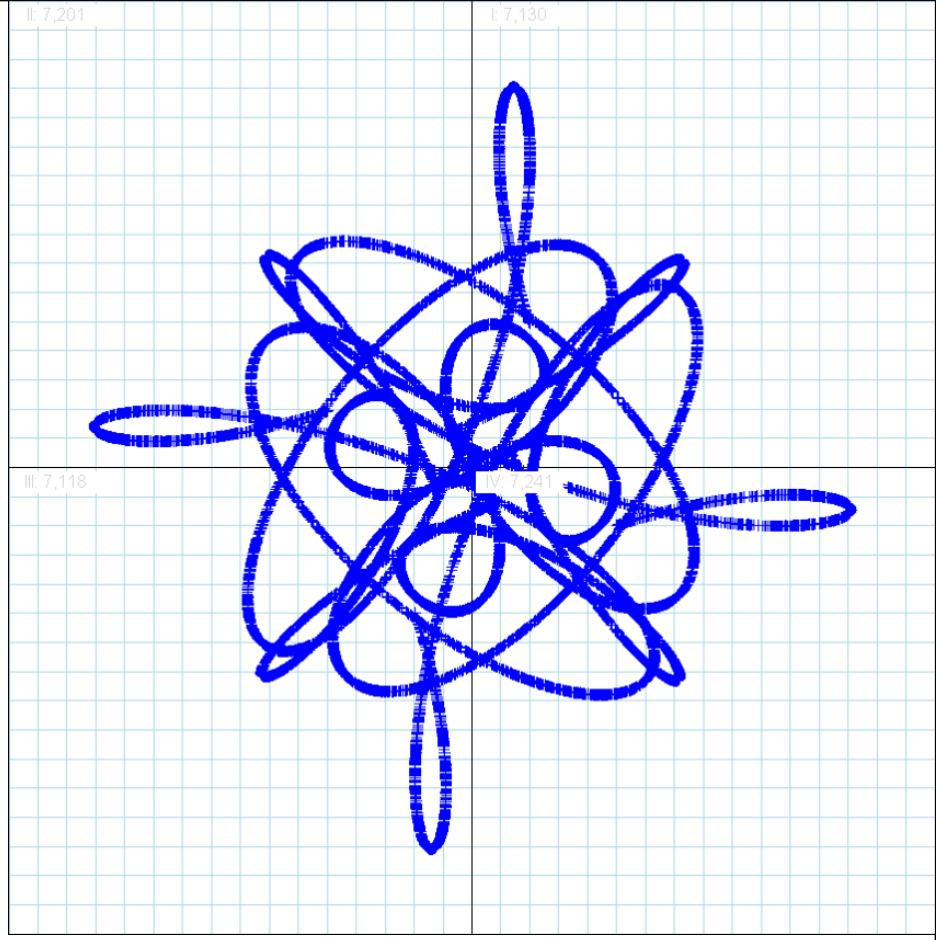
- **Reports discrepancies (BER, PER) to the Control Unit.**



System Design – Channel Emulator

- **Provides the required channel impairment.**
  - AWGN
  - Multipath Fading
  - Doppler shift





Channel Emulator

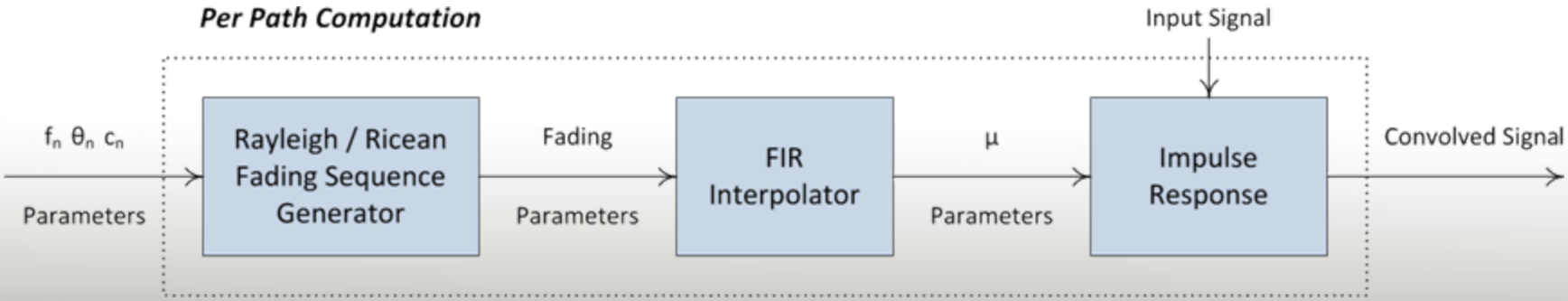
Our Channel Emulator

- **Based on "MEDS" (Method of Exact Doppler Spread) [Pätzold, 2002].**
- **Excellent compromise between computational complexity and accuracy of results.**
- **Real and Imaginary components of the fading sequences are uncorrelated.**
- **Discrete Doppler Frequency and Coefficients Generator**
  - Used to shape a Gaussian process into the desired Doppler PSD
  - Design flexibility to support multiple types of Doppler PSDs

Channel Emulator Scalability

- **Per path units**
  - Rayleigh / Ricean Fading Sequence Generator
  - FIR Interpolator
  - Impulse Response Convolver

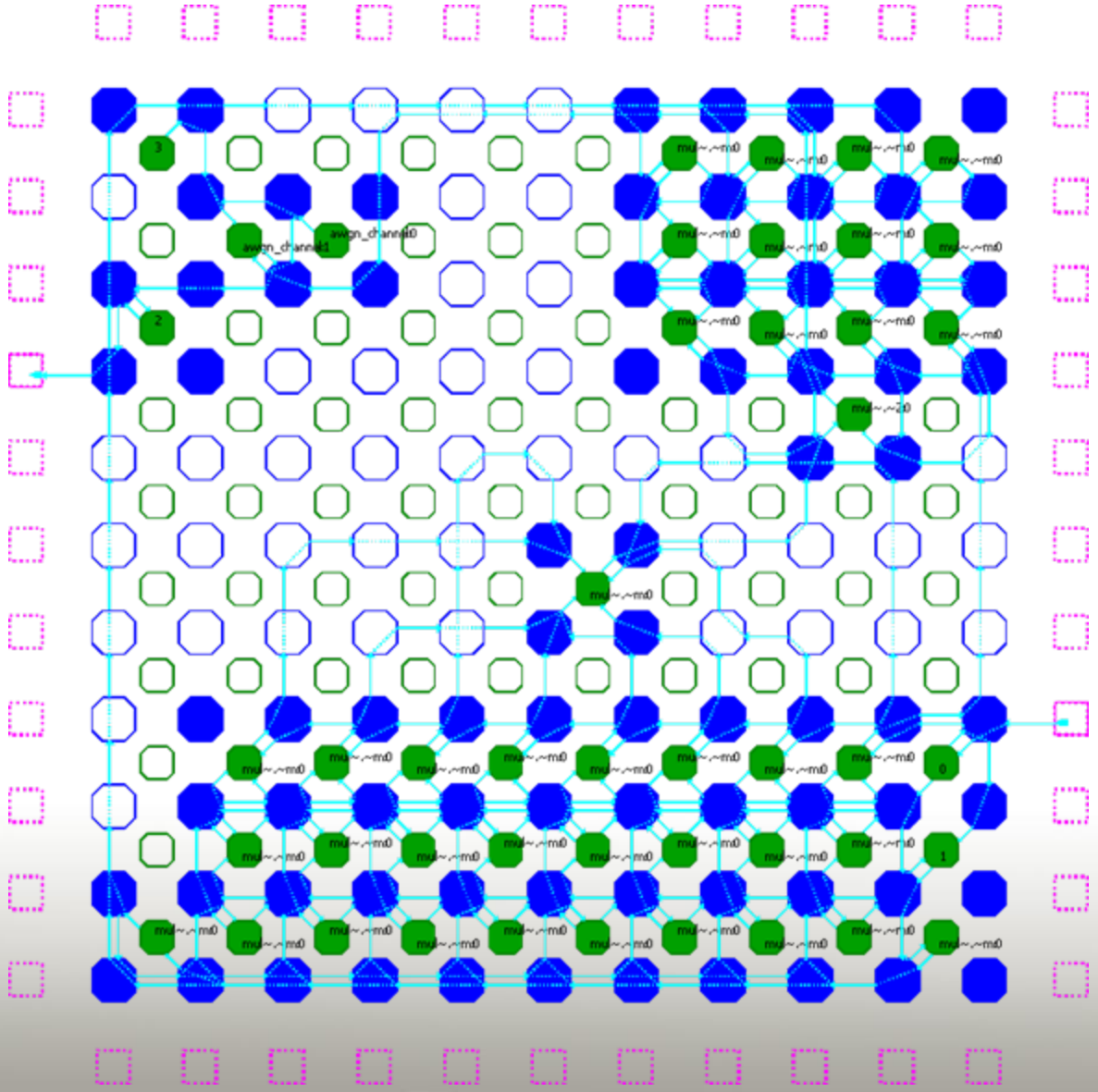
- **Each Path instantiates the same source code**
  - Each computes in parallel with the others (use of separate PEs)
  - Can add as many paths as desired



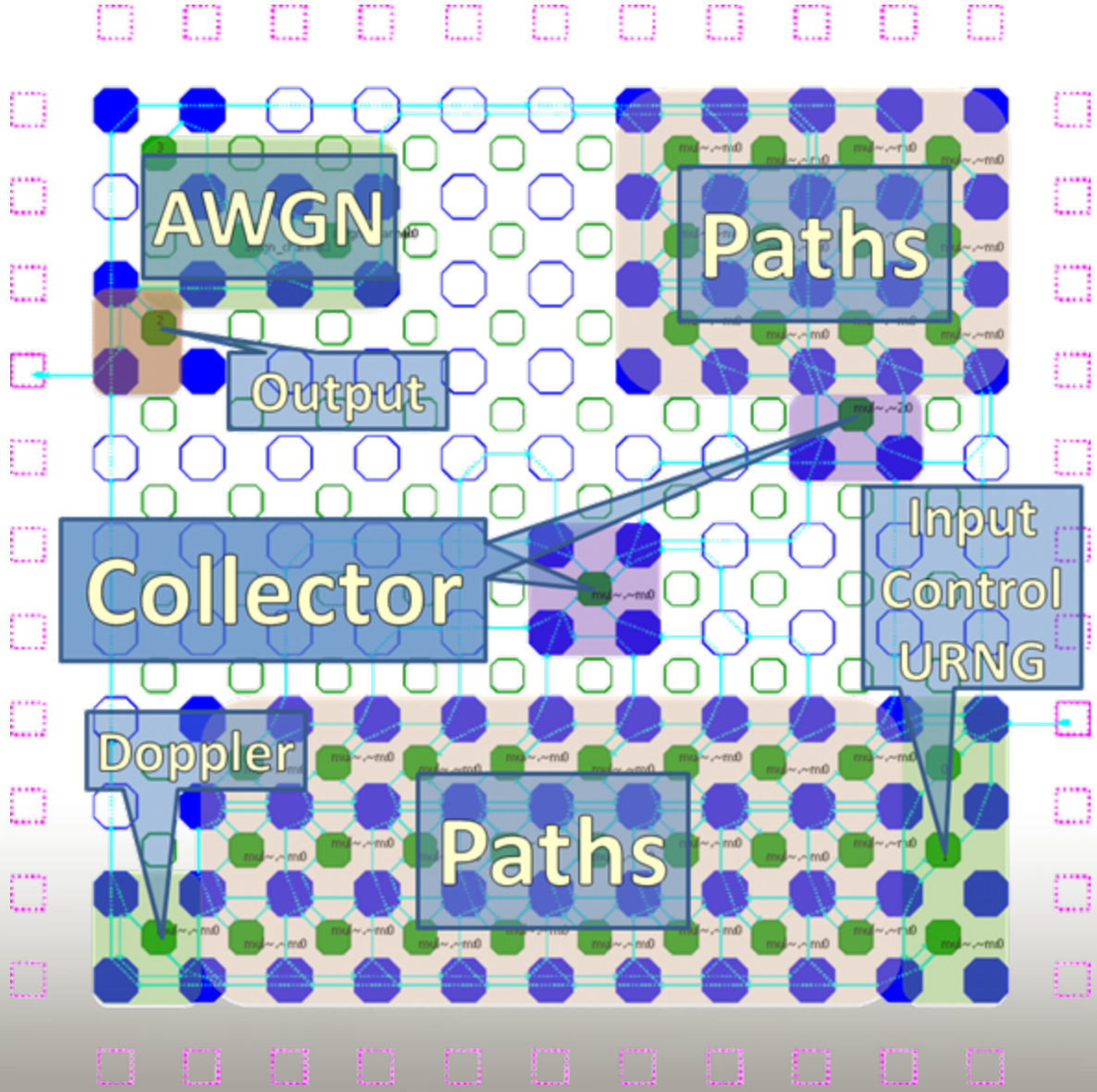
Capacity and Performance

- **Supports up to 12 independent, uncorrelated paths.**
  - Each path Rayleigh / Ricean / Static selectable.
  - Per path configurable parameters include: path gain (dB), Ricean K-factor (linear) (if applicable), path delay (µs), maximum Doppler frequency (Hz), number of harmonic functions (int).
- **20 M complex samples / second at system typical clock rate (500 MHz) with 25 I / 26 Q harmonic functions per fading path.**
- **AWGN alone supports 30 M complex samples / second at system typical clock rate (500 MHz).**
- **No model-based limitation on path delay spread.**
  - Current implementation supports delays ≤ 400 µs.

Channel Emulator HyperX Layout

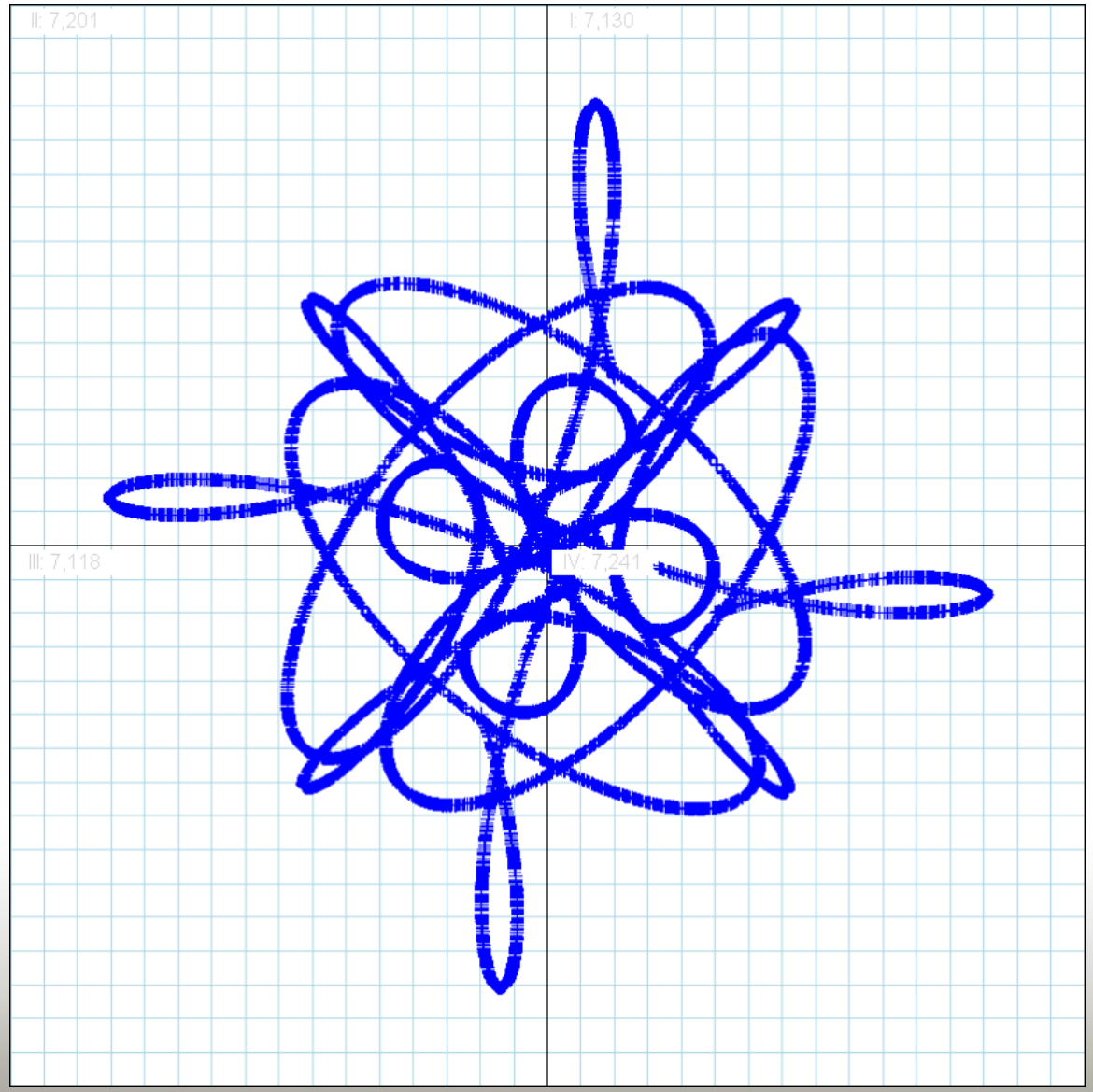


Channel Emulator HyperX Layout



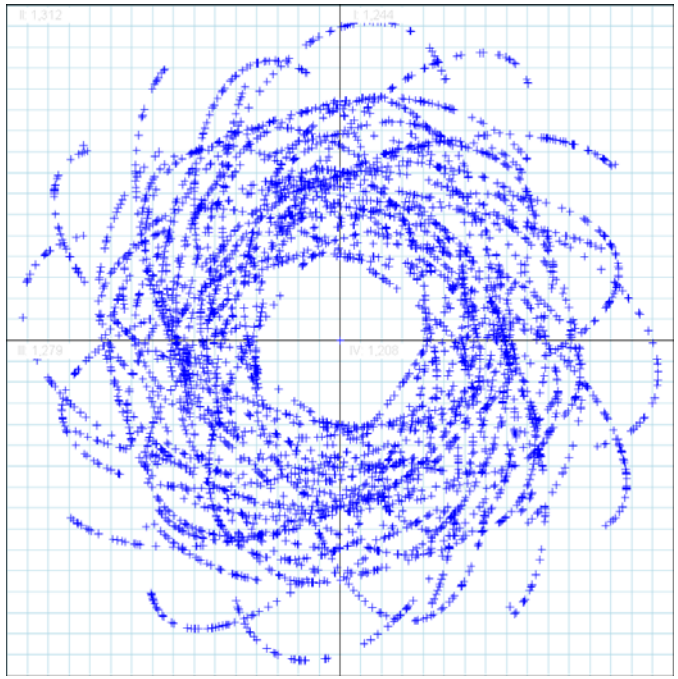
Sample Results

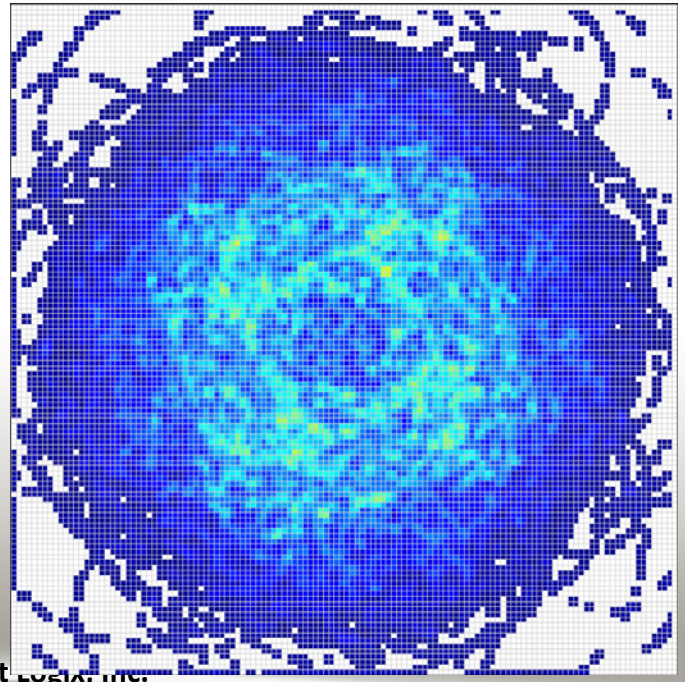
- **Source**
  - Step function
  - (Phasor trajectory)
- **Channel**
  - 1 Rayleigh Path
  - 0 dB Path Gain
  - 0 µs Path Delays
  - Fade Rate 80 Hz
  - 25 I / 26 Q Harmonics
  - No Gaussian Noise Added



Sample Results

- **Source**
  - Random Signal
  - QPSK (4-QAM) Modulated
- **Channel**
  - 3 Paths, 1x Ricean, 2x Rayleigh
  - Ricean K-factor of 5 in first path
  - Path Gains
    - 0 dB, -3 dB, -6 dB
  - Path Delays
    - 0 µs, 2.0 µs, 5.0 µs
  - Max Doppler Frequencies
    - 100 Hz, -200 Hz, 80 Hz
  - 25 I / 26 Q Harmonics
  - No Gaussian Noise Added





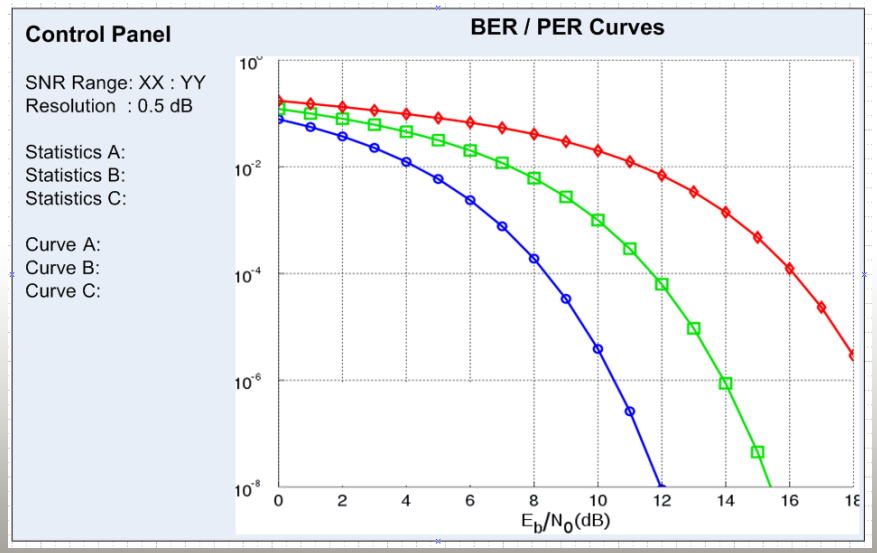
"Smart Allocation of Resources"

Adaptive Testing

Adaptive Testing

- **Observation: Lower SNR values require exponentially less tests than higher SNR values for equal statistical BER/PER accuracy.**

- **Generally, concentrate testing time and resources on higher SNR values to "drive the BER/PER curves down".**

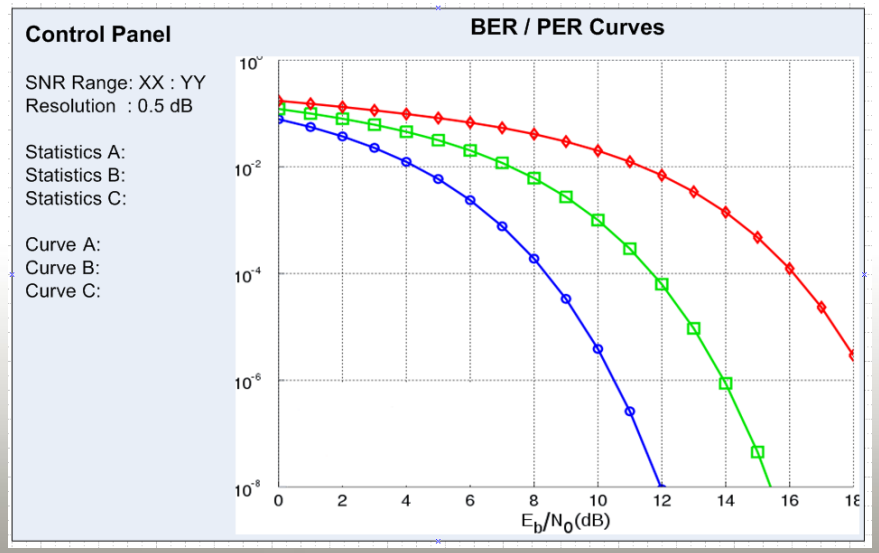


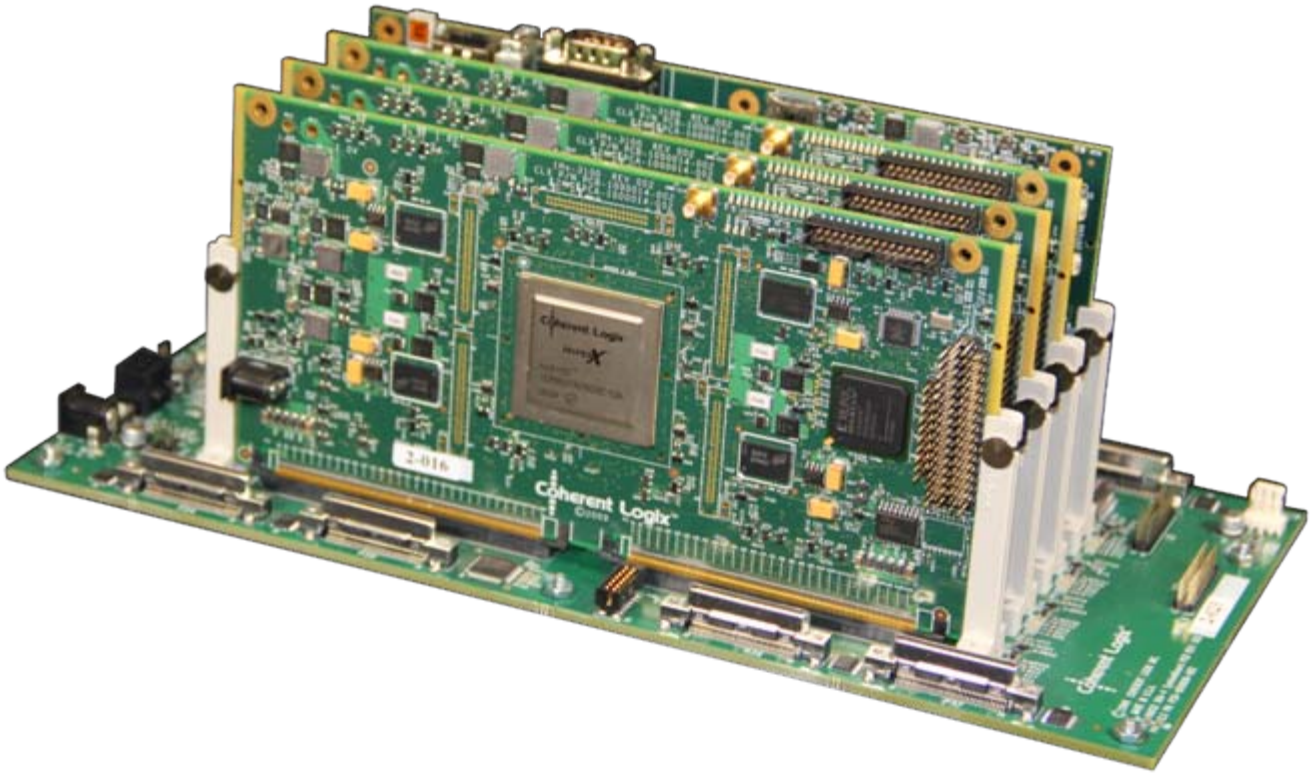
Adaptive Testing (cont.)

- **Adaptively schedule tests to run at SNR values such that one collects meaningful BER and PER statistics as rapidly as possible across the entire SNR range.**

- **Sweep SNR range from left-to-right until each SNR point reaches a certain number of packet failures.**

- **Expect the "number of tests per SNR point" to plot inversely proportional to the PER curve.**





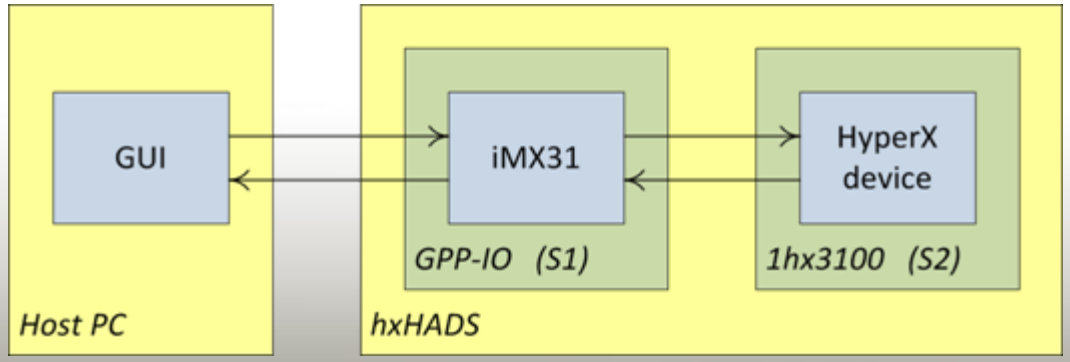
Example Test Case

Test Case Details

- **Simple transceiver design to demonstrate system architecture.**
  - Convolutional encoder (R = ¾, K = 7).
  - Soft-decision Viterbi decoder.
  - BPSK modulation and demodulation.
- **Channel Impairment comprises AWGN only.**
- **User parameter controls**
  - SNR range to test
  - SNR resolution
  - Message/packet size
  - Transmitter power

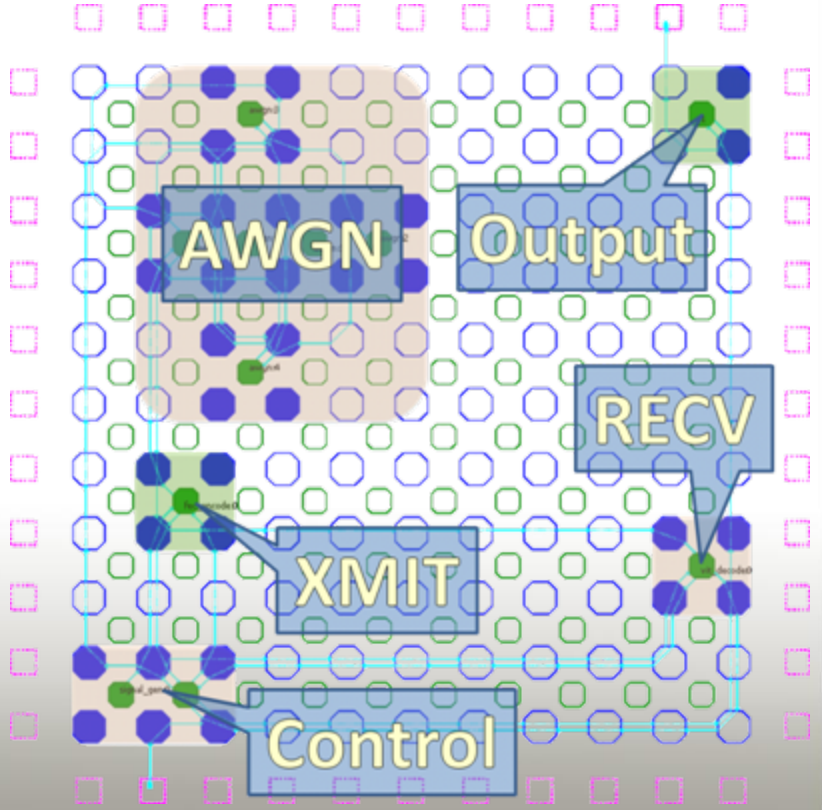
Implementation and Components

- **The GUI runs on a Host PC and communicates with a hxHADS development system through an Ethernet connection.**
- **The rest of the system runs on a single hx3100 HyperX device.**
- **The iMX31 processor on the hxHADS provides network connectivity for data I/O between the Host PC and the HyperX device.**



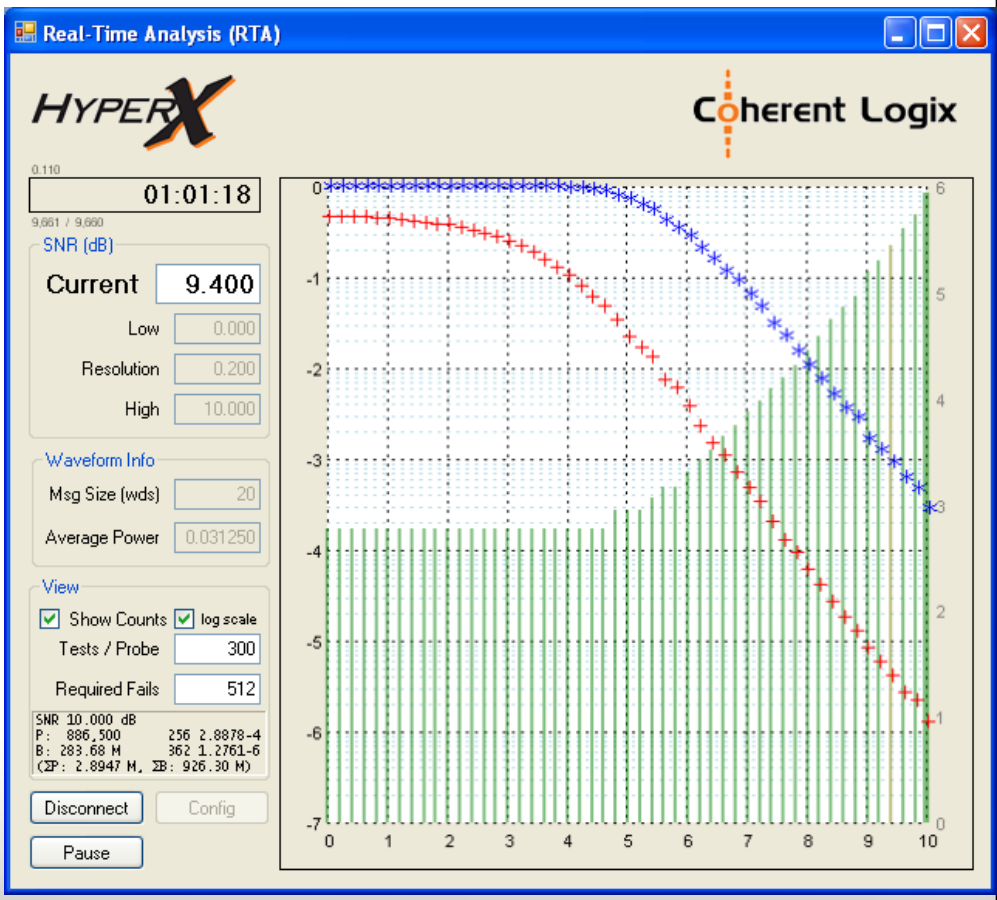
HyperX Implementation

- **Control Unit includes message generator, control, and compare.**
- **High performance AWGN channel.**
- **XMIT: FEC and Modulation**
- **RECV: Viterbi and Demodulation**
- **Total Resources: 11 PEs**



Test Parameters and Results

- **SNR range between 0 – 10 dB, at 0.2 dB resolution.**
- **320-bit messages.**
- **Approximately 2.7 M test messages in one hour (870 M bits).**
- **Adaptive testing effort concentrated on the higher SNR points.**
  - Green bars indicate number of tests per SNR point (log scale).



Test Results (cont.)

- **All SNR points had at least 256 packet errors in one hour.**
- **Amount of time to achieve each successive (power of 2) level of required fails is twice that of the previous level.**

Number of Packet Errors	HH:MM:SS
1	00:00:32
2	00:00:44
4	00:01:12
8	00:02:07
16	00:04:04
32	00:07:32
64	00:15:57
128	00:31:17
256	01:00:25

Conclusions

Summary

- **Entirely software-based, real-time characterization of receiver dynamic channel performance.**
  - Real-time characterization at an early stage of development.
  - Software-based allows for maximum design flexibility.

- **The computational capacity of the HyperX architecture allows for an integrated design and characterization setup.**
  - A single HyperX device may contain various components of the design and the characterization platform.
  - The hxHADS system allows expandability to multiple HyperX devices in order to separate the transceiver design from the characterization platform, while seamlessly communicating between the two.

What's Next

- **More comprehensive transceiver designs.**
- **Make use of the multi-slot capability of the hxHADS system. Transceivers and Channel Emulator in dedicated HyperX devices.**
- **Additional performance characterization measurements.**
  - I/Q constellation display (scatter plot)
  - Burst error statistics
  - Dynamic frequency response measurements
- **Integrate Channel Emulator to the characterization design.**

Thank you for your time and attention!

Ivan Aguayo

Office of the Chief Technologist

Coherent Logix, Incorporated

Office: +1-512-579-4970, Fax: +1-512-382-8941

aguayo@coherentlogix.com

Photograph Courtesy NASA

Note

- **The latest version of this presentation and the corresponding paper will be accessible from the Coherent Logix, Inc. website after the conclusion of the conference.**

- **Please visit**

www.coherentlogix.com

for more details.