IBIS-AMI and Co-Optimization

Todd Westerhoff, Walter Katz, and Mike LaBonte

January 2015

Optimization

Mathematical optimization

From Wikipedia, the free encyclopedia
(Redirected from Optimization)

> "Optimization" and "Optimum" redirect here. For other uses, see Optimization (disambiguation) and Optimum (disambiguation).

In mathematics, computer science, economics, or management science, **mathematical optimization** (alternatively, **optimization** or **mathematical programming**) is the selection of a best element (with regard to some criteria) from some set of available alternatives.[1]

In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. The generalization of optimization theory and techniques to other formulations comprises a large area of applied mathematics. More generally, optimization includes finding "best available" values of some objective function given a defined domain (or a set of constraints), including a variety of different types of objective functions and different types of domains.
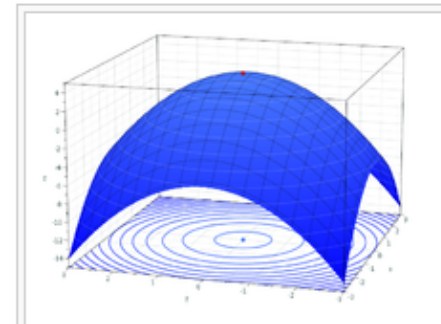
Graph of a paraboloid given by $f(x, y) = -(x^2 + y^2) + 4$. The global maximum at (0, 0, 4) is indicated by a red dot.

Other Terms …

The page "Co-Optimization" does not exist.
see whether the topic is already covered.
For search help, please visit Help:Searching.

Co-Optimization

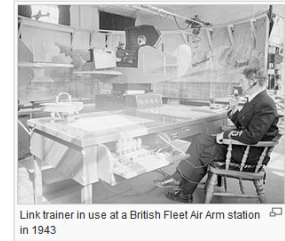## Link Trainer

From Wikipedia, the free encyclopedia
(Redirected from Link Training)

The term **Link Trainer**, also known as the "Blue box" and "Pilot Trainer"[1] is commonly used to refer to a series of flight simulators produced between the early 1930s and early 1950s by the **Link Aviation Devices, Inc**, founded and headed by Ed Link, based on technology he pioneered in 1929 at his family's business in Binghamton, New York. These simulators became famous during World War II, when they were used as a key pilot training aid by almost every combatant nation.

The original Link Trainer was created in 1929 out of the need for a safe way to teach new pilots how to fly by instruments. A former organ and nickelodeon builder, Link used his knowledge of pumps, valves and bellows to create a flight simulator that responded to the pilot's controls and gave an accurate reading on the included instruments. More than 500,000 US pilots were trained on Link simulators,[2] as were pilots of nations as diverse as Australia, Canada, Germany, United Kingdom, Israel, Japan, Pakistan and the USSR.

Link trainer in use at a British Fleet Air Arm station in 1943

The Link Flight Trainer has been designated as a *Historic Mechanical Engineering Landmark* by the American Society of Mechanical Engineers.[2] The Link Company, now the Link Simulation & Training division of L-3 Communications, continues to make aerospace simulators.[3]

Link Training

Backchannel (disambiguation)

From Wikipedia, the free encyclopedia

A **backchannel** is a real-time online conversation using networked computers that takes place alongside live spoken remarks.

Backchannel may also refer to:

- Return channel, a low-speed, or less-than-optimal, telecommunications transmission channel in the opposite direction to the main channel
- Backchanneling, the method a malicious malware program uses to secretly communicate to command and control servers from a compromised computer
- Track II diplomacy, an unofficial channel of communication between states or other political entities
- Backchannel (linguistics), listener responses that can be both verbal and non-verbal in nature
- Backchanneling, an organizational practice in business that involves bypassing recognized or official chains of command in order to create vulnerability[clarification needed] at the level(s) skipped

Back Channel

User Scenario #1:
Co-Optimization in Hardware **and** Models

- SerDes hardware uses run-time back channel communication to co-optimize TX & RX settings
  - AMI models implement hardware optimization protocol as closely as possible to predict how hardware will converge
  - Implies that results are optimization path-dependent and provide a local (possibly non-global) optimum
  - User programs hardware to self-optimize and expects that the system will converge to settings predicted by simulation

- Designer's Questions:
  - Will this link converge?
  - Do we need specific presets to ensure convergence?
  - What will the trained margins be?

SiSoft™
We Are Signal Integrity

User Scenario #1 Requirements

1. TX & RX models emulate hardware protocol
   - Models must communicate at simulation run-time
2. Report optimized margins (eye height, width, etc.)
3. Cross-vendor support
4. Report optimized IP settings (taps, etc.)
5. Constrain solution based on IP capabilities
6. Init and GetWave-based optimization
7. Support hardware starting point (presets)
8. Support multiple protocols with one model
9. Support private protocols
10. Probes work correctly (TX EQ from TX)

SiSoft™
We Are Signal Integrity

User Scenario #2:
Co-Optimization in Models **Only**

- SerDes hardware **does not** support hardware run-time optimization
  - AMI models perform co-optimization **beyond** what hardware can do at run-time
  - Implies that results are optimization path-independent and seek to provide a global optimum
  - User programs hardware based on simulation results and expects that the system will exhibit predicted margins

- Designer's Questions:
  - Can this link work with this IP?
  - How should IP be configured?
  - What will the margins be?

SiSoft™
We Are Signal Integrity

Why User Scenario #2 ?

- Provide a starting point for lab validation

- Optimize settings for individual links at the system level
  - As opposed to "bucketing by length"

- Eliminate need to run "blind sweep" simulations

SiSoft™
We Are Signal Integrity

User Scenario #2 Requirements

1. Report optimized IP settings and margins
2. Cross-vendor support
3. Provable methodology
   - Ability to validate results independently
4. High performance optimization
   - Optimize 4,000 links overnight
5. Constrain solution based on IP capabilities
6. Support private protocols
7. Probes work correctly
8. User-selectable optimization criteria
   - Defined objectives and constraints

SiSoft™
We Are Signal Integrity

User Scenario #3:
Co-Optimization with Legacy AMI Models

- Optimize TX & RX settings for SerDes IP with legacy SerDes IBIS-AMI models
  - No hardware backchannel
  - No additions to DLL functionality beyond IBIS 6.0
  - Optimization executed by EDA tool
  - Additional meta-data supplied to help EDA tool understand TX models that do not support training communication

- Designer's Questions:
  - Can this link work with this IP?
  - How should IP be configured?
  - What will the margins be?

SiSoft
We Are Signal Integrity

Why User Scenario #3 ?

- Support optimization without recompiling DLLs

- Support EDA tool exploration and optimization of model settings without running many separate simulations

- Provide a starting point for lab validation

- Optimize settings for individual links at the system level

SiSoft
We Are Signal Integrity

User Scenario #3 Requirements

1. Legacy model DLLs supported
   - NOTE: Requires TX models that can be configured to use exactly the settings passed in, no self-optimization changes
   - NOTE: Legacy Tx models need to supply ASCII meta-data to map tap coefficients to AMI equalization settings.
2. Report optimized IP settings and margins
3. Cross-vendor support
4. Provable methodology
   - Ability to validate results independently
5. High performance optimization
   - Optimize 4,000 links overnight
6. Constrain solution based on IP capabilities
7. Probes work correctly
8. User-selectable optimization criteria
   - Defined objectives and constraints

SiSoft™
We Are Signal Integrity

Terminology

- **Adaptation** – When AMI models change their behavior (EQ, clock recovery) on a bit by bit basis in time-domain simulation
- **Eye Quality Metric (EQM)** – Computed measure of eye quality at the RX decision point, such that different scenarios can be evaluated and the best case identified
- **Self-Optimization** – When an RX adjusts it internal behavior based on an internal EQM to optimize its settings
- **Co-Optimization** – Simultaneous adjustment of TX / RX settings to optimize EQM, usually at the RX decision point
- **Co-Optimization by proxy** – Where one device provides the EQ that would normally be provided by another. In AMI this normally occurs with TX EQ disabled and the RX providing the TX (LTI) equalization in the TX's place

SiSoft™
We Are Signal Integrity

Key Questions

- What is being optimized?
- Who does the optimizing?
- What optimization algorithm?
- Local or global optimum?

SiSoft
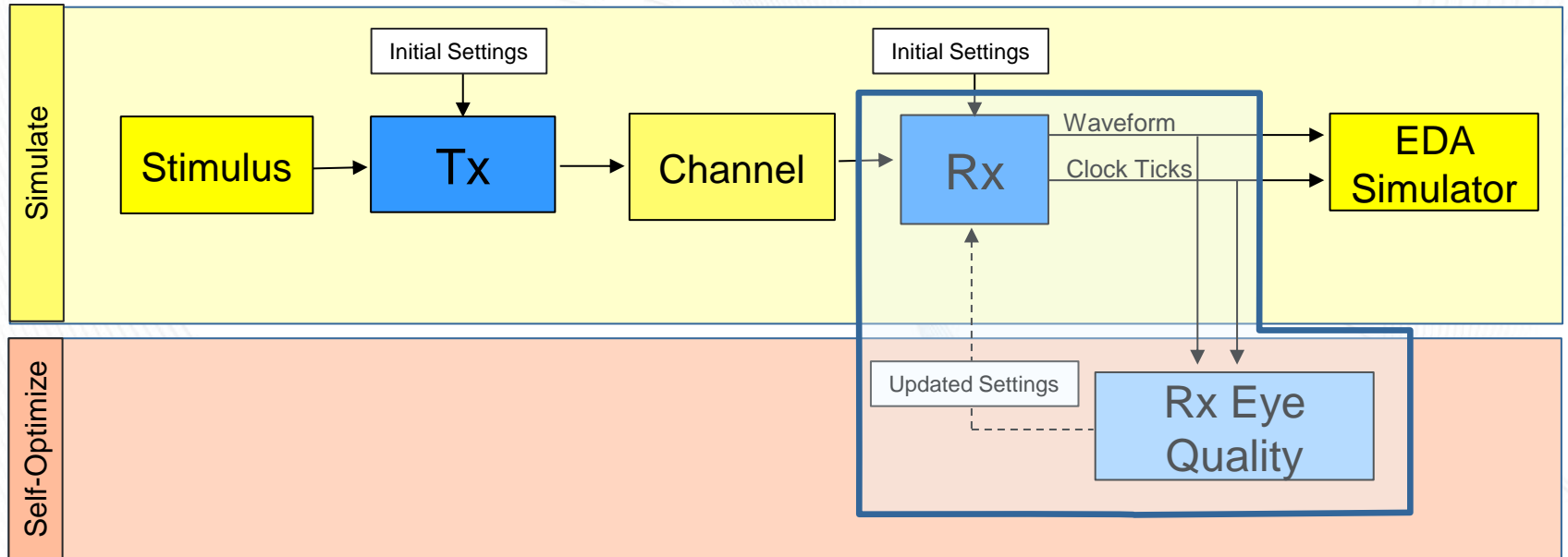We Are Signal Integrity

No Optimization

Legend: EDA (yellow) | AMI (blue)
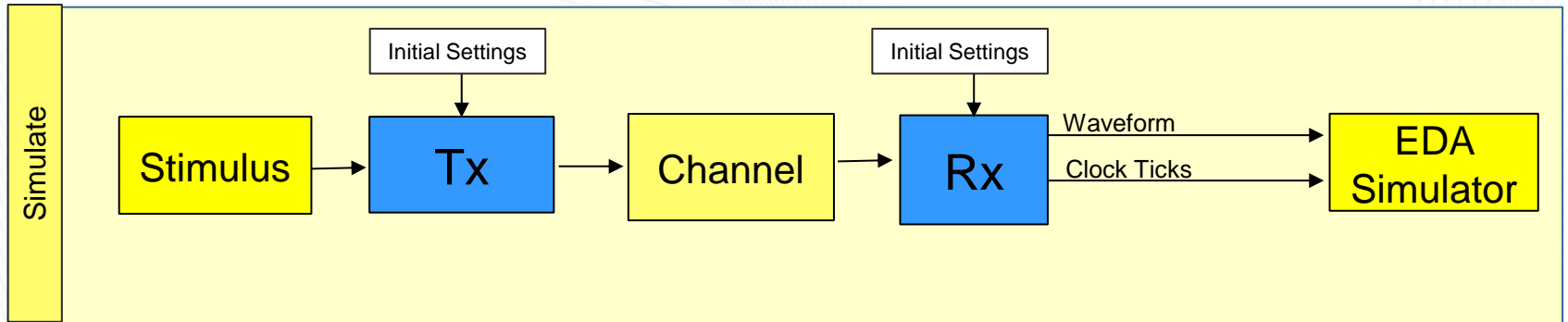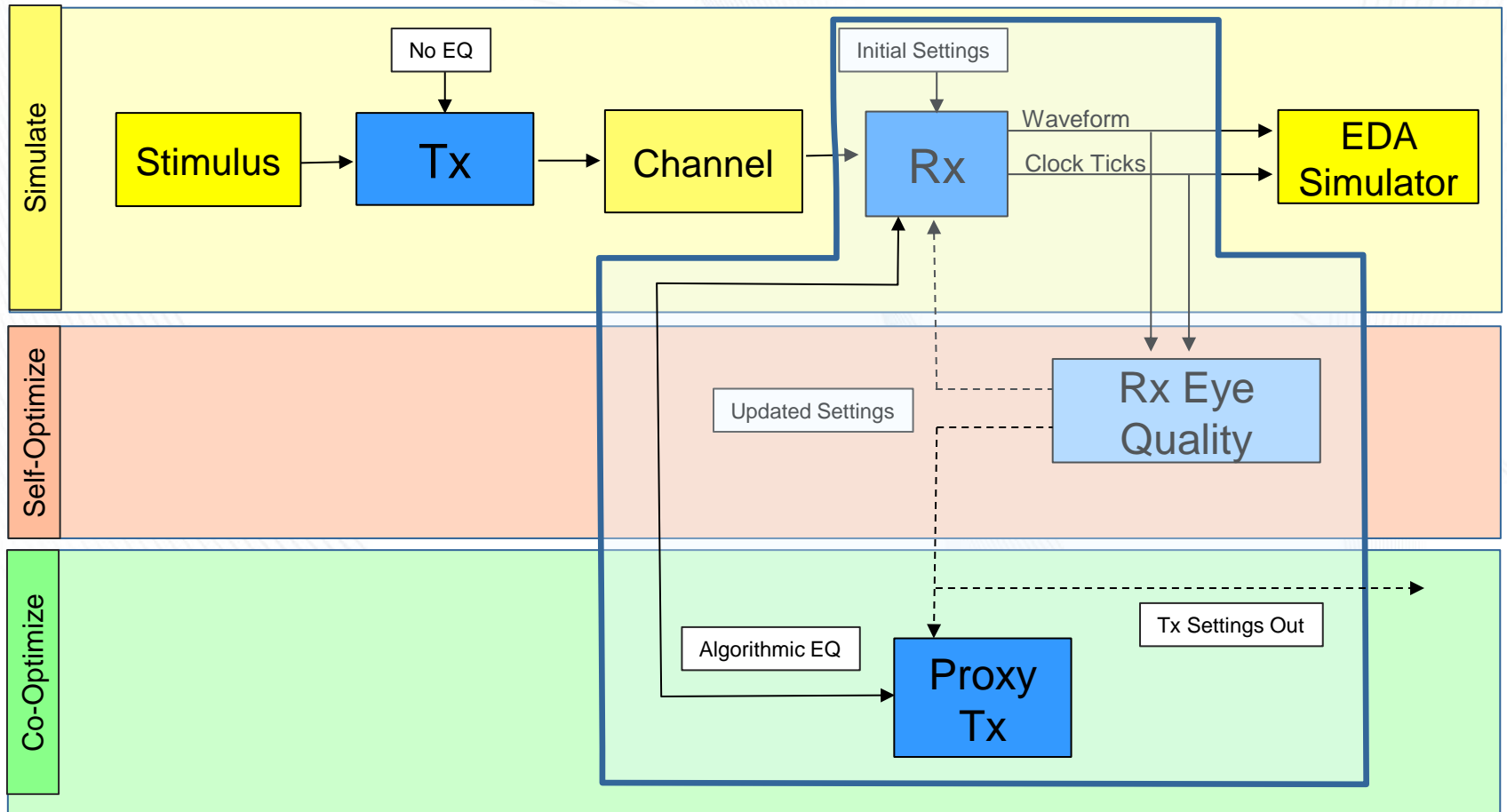
Self-Optimization

Legend: EDA AMI

Co-Optimization by Proxy

- Special case using matched pair models:
  - TX equalization is disabled
  - RX provides TX equalization <u>in place of</u> TX
  - RX co-optimizes itself <u>and</u> proxy TX settings
  - Optimized TX settings are reported
  - Subsequent analysis used to verify TX settings
- Requires no changes to existing AMI flows
- RX has detailed knowledge of TX EQ
  - Only works for paired TX / RX models
  - Limited / no cross-vendor support

SiSoft
We Are Signal Integrity

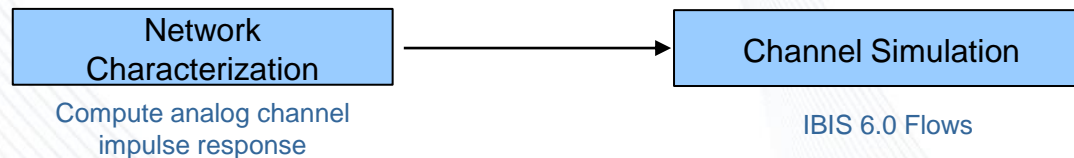# Co-Optimization by Proxy

Legend: EDA   AMI
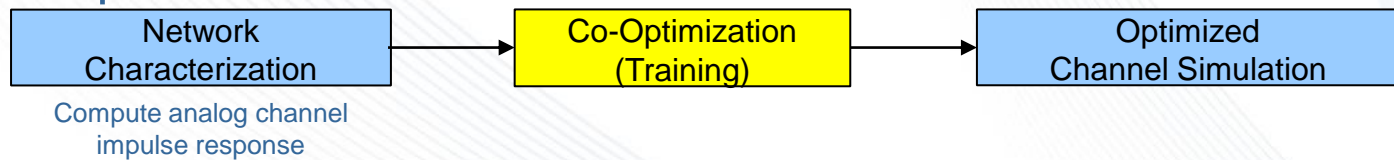
SiSoft™
We Are Signal Integrity

Co-Optimization - Analysis Flow

- In hardware, co-optimization (link training) precedes normal system operation.

- We believe that simulation-based co-optimization should follow the same pattern

Normal

| Network Characterization | → | Channel Simulation |

Compute analog channel impulse response

IBIS 6.0 Flows

Co-Optimization

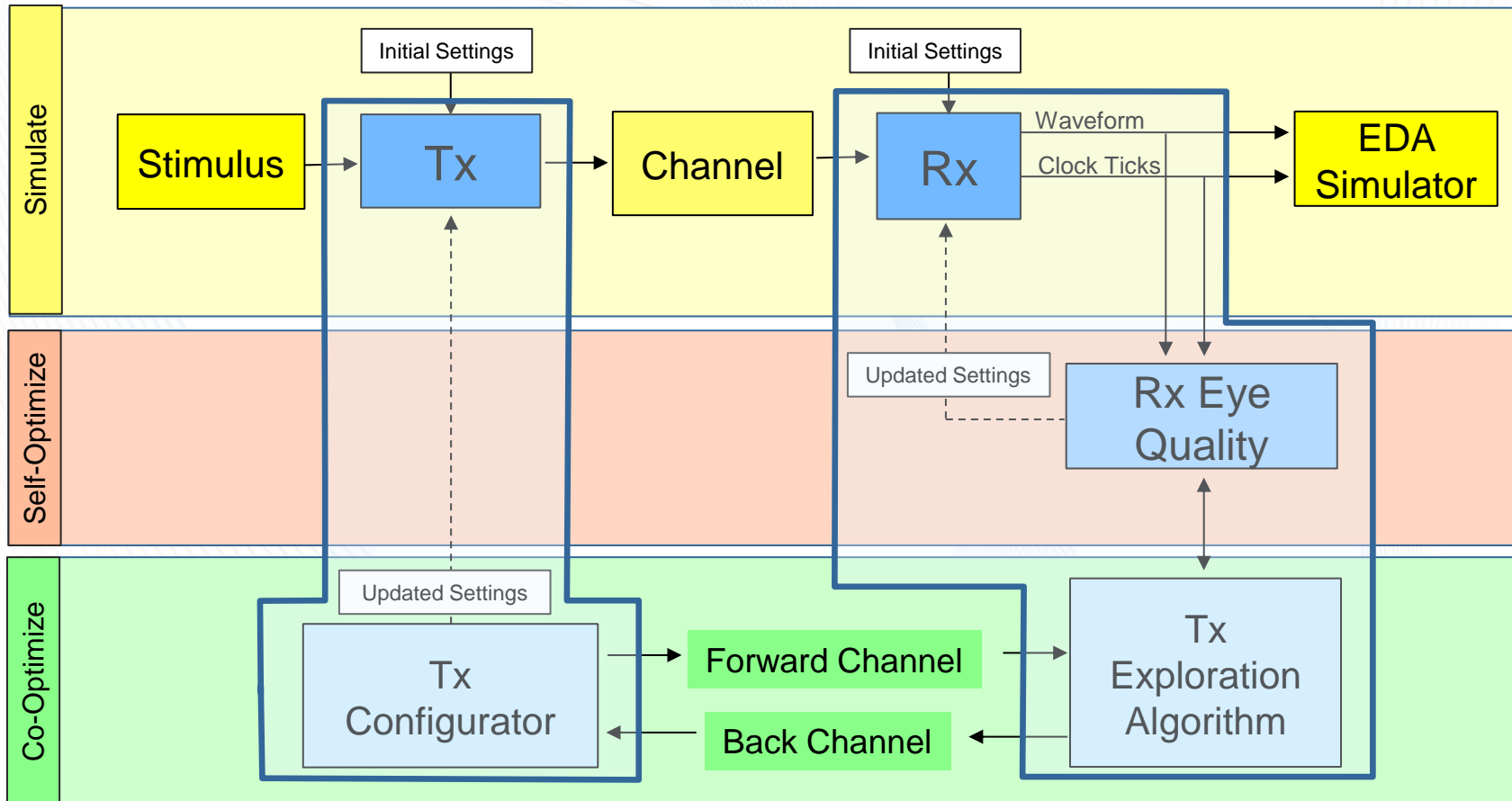| Network Characterization | → | Co-Optimization (Training) | → | Optimized Channel Simulation |

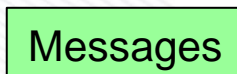Compute analog channel impulse response

SiSoft™
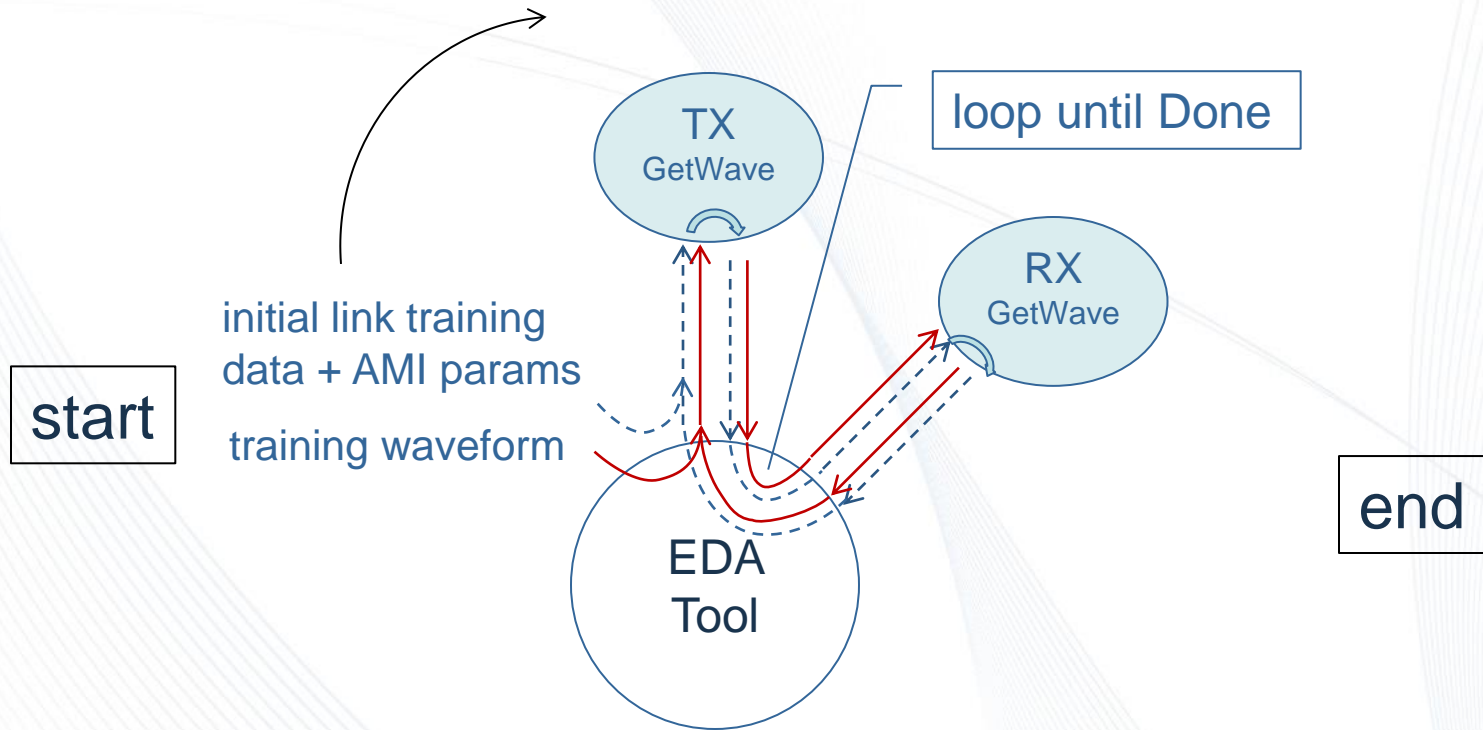We Are Signal Integrity

Co-Optimization

Key Questions

- ## What is being optimized?
  - Eye Quality –quantitative metric determined by the RX model based on looking at its own behavior. Does not have to be rigorously defined or even reported.

- ## Who does the optimizing?
  - The RX model

- ## What optimization algorithm?
  - Scenario #1 – an algorithm internal to the RX model that follows the hardware protocol as closely as possible
  - Scenario #2 – an algorithm internal to the RX model

- ## Local or Global optimum?
  - Scenario #1: Local
  - Scenario #2: Local or Global

Analysis Mode Requirements

- Emulate hardware training protocol as closely as possible (Scenario 1, requirement 1)
  - Requires training be performed using bit by bit waveform processing (GetWave)
- High performance Co-Optimization (Scenario 2, requirement 4)
  - Requires training be performed using impulse response processing (Init)

SiSoft™
We Are Signal Integrity
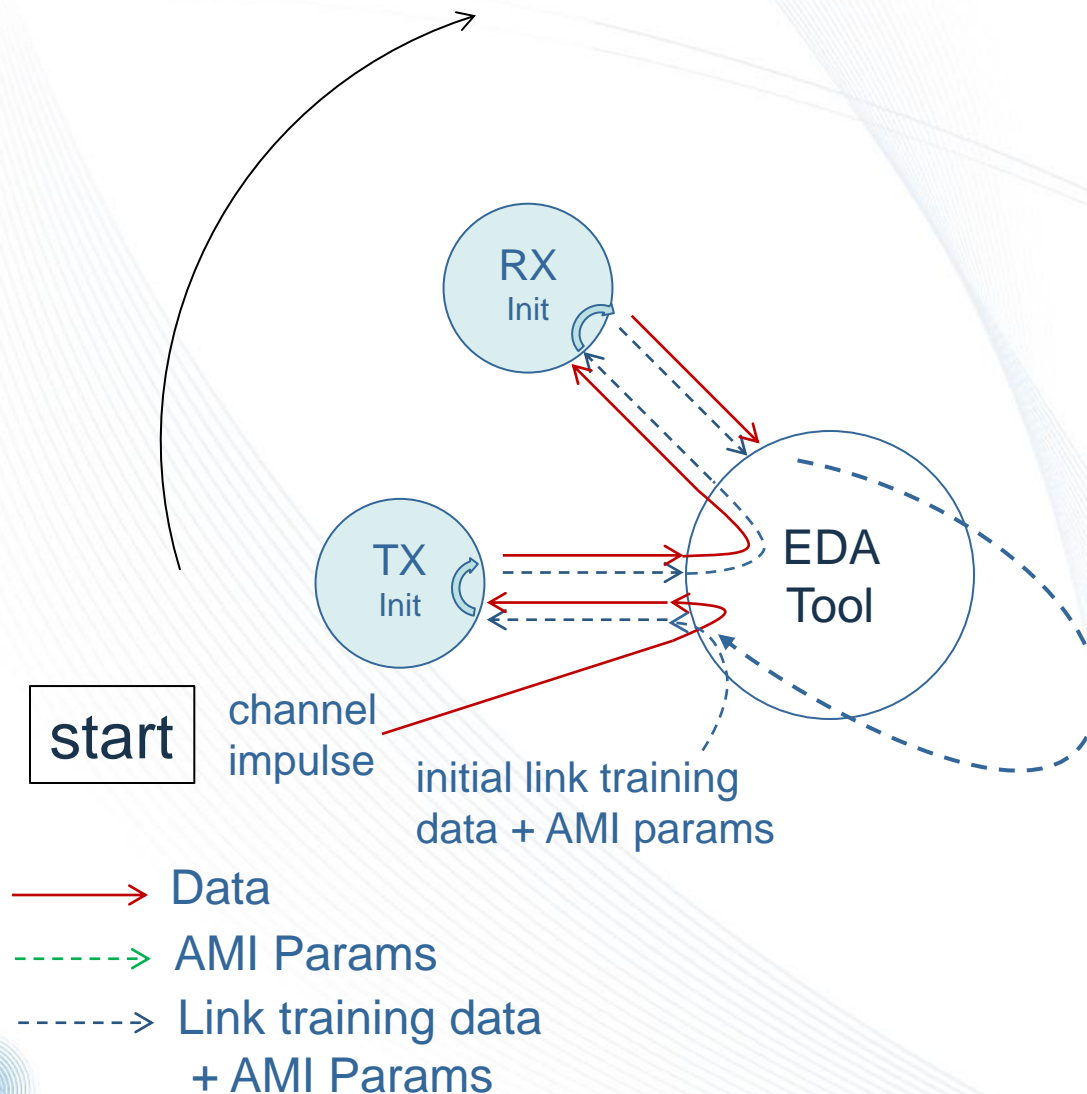
Time Domain Link Training

start

loop until Done

TX GetWave

RX GetWave

initial link training data + AMI params

training waveform

EDA Tool

end

Data

AMI Params

Link training data + AMI Params

SiSoft
We Are Signal Integrity

Statistical Link Training

start

channel impulse

initial link training data + AMI params

RX Init

TX Init

EDA Tool

→ Data

----→ AMI Params
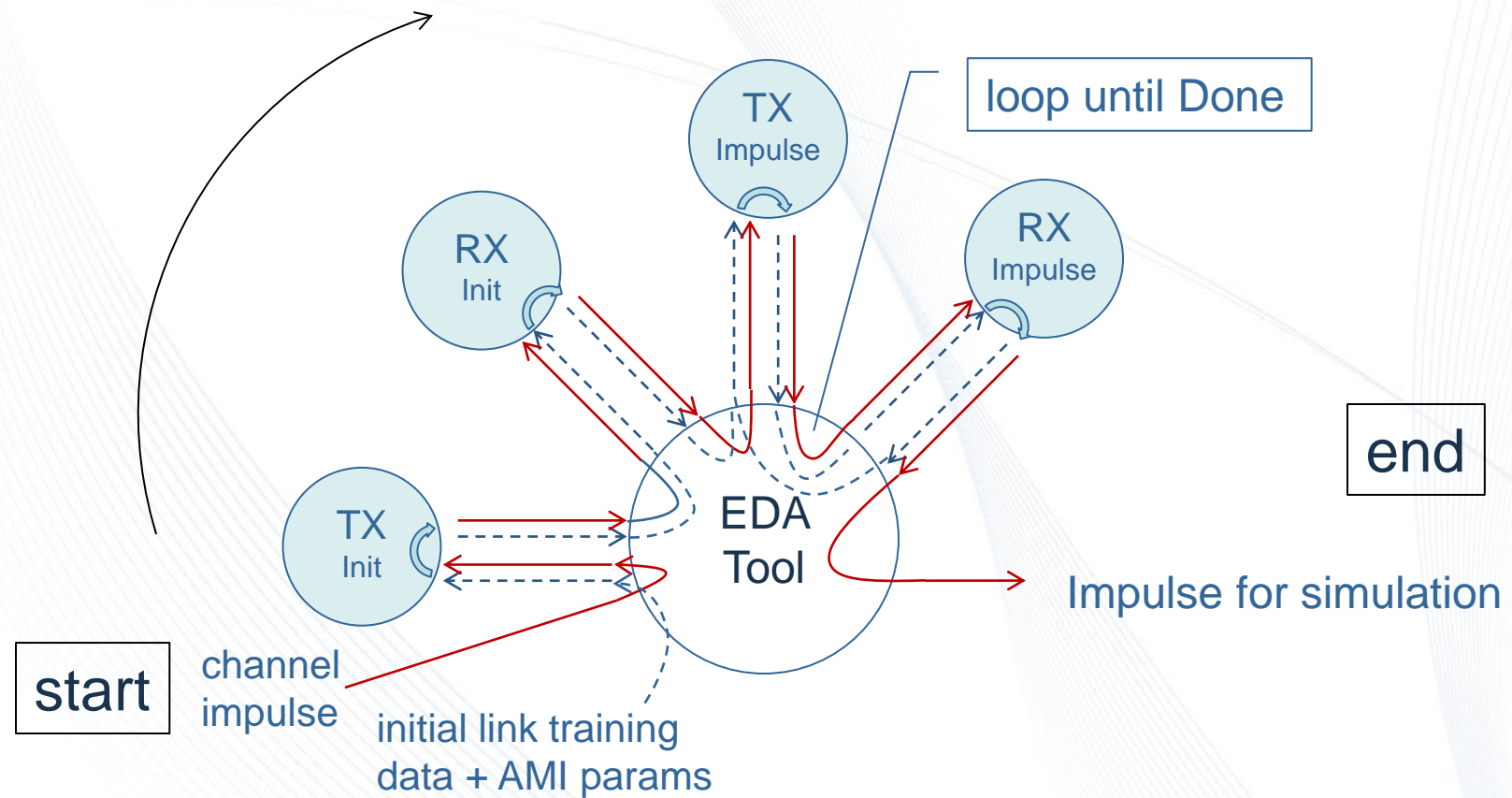
----→ Link training data + AMI Params

Problem: We need to call AMI_Init again, but it can be called only once.

Solution: New function ***AMI_Impulse*** with same signature as AMI_Init:

❌ Initialize

✔ Params In/Out

✔ Impulse response

SiSoft
We Are Signal Integrity

Statistical Link Training (again)

loop until Done

start

end

TX
Impulse

RX
Init

RX
Impulse

TX
Init

EDA
Tool

Impulse for simulation

channel impulse

initial link training
data + AMI params

——→ Data

-----→ AMI Params

-----→ Link training data
        + AMI Params

SiSoft
We Are Signal Integrity

Statistical / Time-Domain Link Training

TX Impulse

RX Init

RX Impulse

loop until Done

pass settings

training waveform

TX Init

EDA Tool

TX GetWave

loop until Done

start

channel impulse

initial link training data + AMI params

RX GetWave

end

Impulse for simulation

Data

AMI Params

Link training data + AMI Params

SiSoft
We Are Signal Integrity
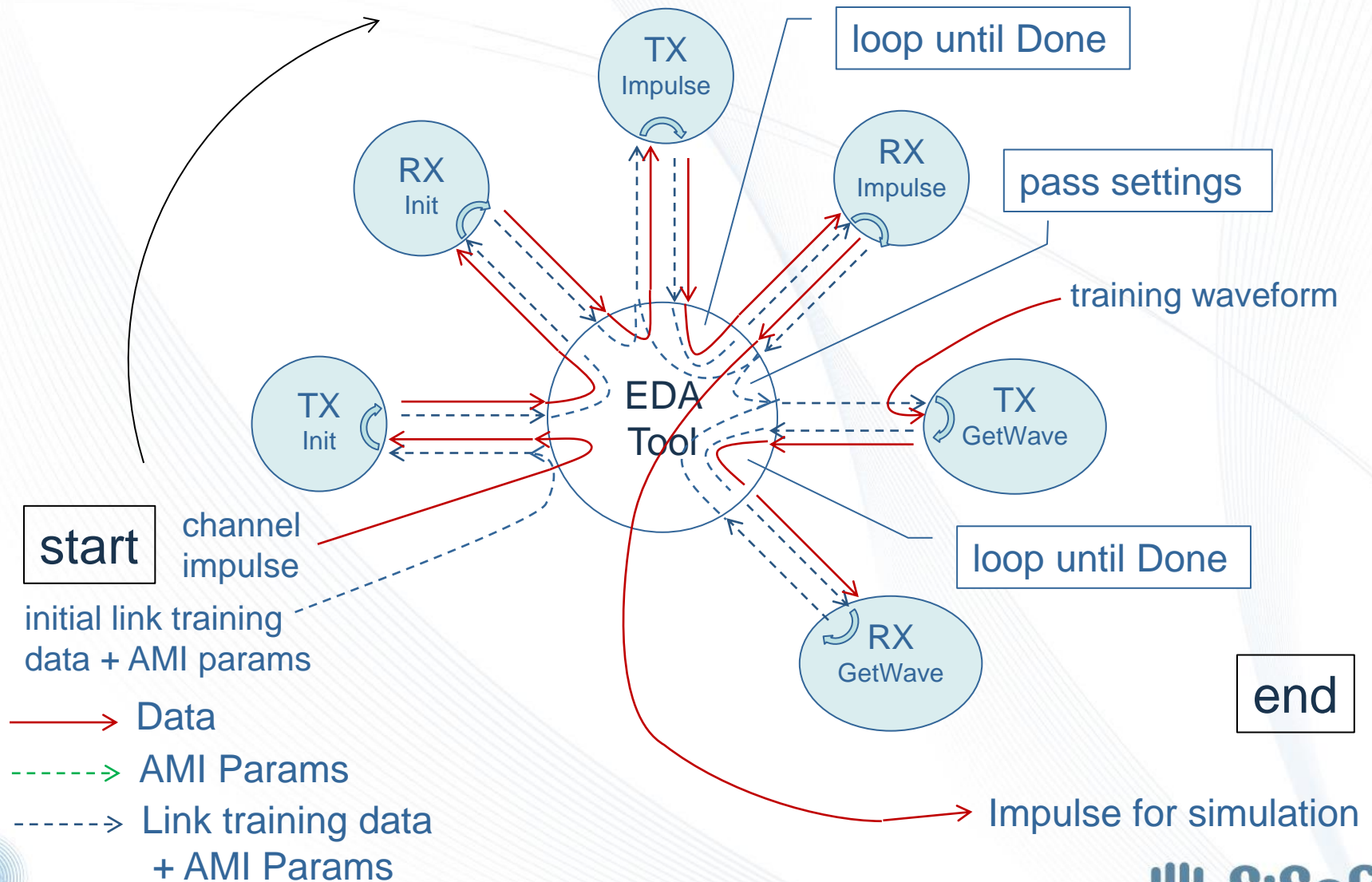
Enhanced Statistical Link Training

start

end

loop until Done

pass settings

training waveform

loop until Done

channel impulse

initial link training data + AMI params

Impulse for simulation

TX Impulse

RX Init

RX Impulse

TX Init

EDA Tool

TX GetWave

RX Impulse

RX GetWave

TX Impulse

——→ Data

-----→ AMI Params

-----→ Link training data + AMI Params

SiSoft™
We Are Signal Integrity