ADS Cosimulation

Cosimulation with Keysight's Advanced Design System

VPIphotonics DESIGN AUTOMATION

How to contact VPIphotonics GmbH

www.VPIphotonics.com	Web site
info@VPIphotonics.com	General information
sales@VPIphotonics.com	Sales, ordering information
support@VPIphotonics.com	Technical support, license key requests

Document Part Number: TC0-WP05-05 19041

Overview

This white paper describes how to run the Advanced Design System (ADS) electronic design automation software from Keysight Technologies (former Agilent Technologies) in simulations with VPIphotonics VPItransmissionMaker™/VPIcomponentMaker™ *(abbreviated here as "VPI-TC")*.

> Note: More details about configuring ADS simulations, such as working with external or encrypted libraries, can be found in ADS User's manual.
> See "Home > Simulation > Simulation-Analog RF > Using Circuit Simulators > ADS Simulator Input Syntax".

The built-in ***ADS_Interface*** module enables seamless modeling of optical communications systems including detailed electrical and electronic circuit models.

This document describes the interface, and provides an application example that demonstrate the value of the interface between ADS and VPI-TC.

Before starting, both VPItransmissionMaker™/VPIcomponentMaker™ and ADS (version 2012 or later) must be properly installed on your computer.

The following environment variables must be specified in the system variable area:

- `SIMARCH=win32_64`
- HPEESOF_DIR=*<ADS_install_dir>*
  By default this is set to `C:\Program Files\Keysight\ADS2015_01`

> Note: For simulation with Remote Simulation Server, the `ADS_LICENSE_FILE` variable must be set instead of `HPEESOF_DIR`.
>
> The `ADS_LICENSE_FILE` variable should specify the path to the license file (or license server) with a value such as
> `C: \Program Files\Keysight\ADS2015_01\data\licenses\license.lic`
> (or `27000@licenses.example.com`).
>
> You can verify the value of this variable using the ADS License Manager program available under Environment in the side menu (*License Manager* is accessible from the ADS Main Window through Tools > License Manager).

Description

Motivation

The performance of electronic components in optical systems and electro-optical devices have a significant impact on the performance of the system as a whole. In most optical communications systems simulations, electrical components (such as drivers, coders, receivers, clock/data recovery) are represented with system-level models denoting the behavior of the electronics as equivalent noise, nonlinearity, gain and frequency-dependent functions. However, a better understanding of interactions between the electronic and photonic parts in a link can be obtained by incorporating full electronic circuit models directly into the simulation of the whole system. Having full electronic models also saves the time required for abstracting the electronic model to a behavioral model at each design iteration.

Keysight provides a high competitive electronic design automation (EDA) software for RF, microwave, and high speed digital applications. VPIphotonics sets the industry standard for end-to-end photonic design automation (PDA) of optical components, systems and networks. Thus, VPIphotonics and Agilent have cooperated to provide a cosimulation interface between their tools. The interface calls the ADS simulator executable directly from VPI-TC, so that the best technologies for photonic and electronic design work seamlessly together.

Working with the Interface

Data Transfer

The bridge between VPI-TC schematics and ADS circuit diagrams is the Keysight Ptolemy signal-processing environment, which is part of the ADS package.

The Agilent Ptolemy environment uses **timed signals**, which are time-sampled signals with timing information. These carry essentially the same information as the **electrical signals** in VPI-TC, and so are used to transfer electrical signals between ADS and TC[1].

In ADS input, timed signals are interpreted as ideal voltage sources (referred to ground) and output timed signals are interpreted as open voltage sources. To obtain a current source in any ADS schematic, a voltage-controlled current source can be used to convert the input timed signal.

In VPI-TC, electrical signals are defined in 'arbitrary units'; that is, they can represent voltage (V) or current (A) waveforms. For example, drives to semiconductor lasers, and the outputs of photodiodes are usually currents, whereas drives to Mach-Zehnder and electro-absorption modulators and the outputs of transimpedance amplifiers are usually voltages. If voltage reflections are present (that is, backward waves), they are represented by an independent 'wire' in VPI-TC.

Furthermore, Keysight Ptolemy allows analog/RF designs to be integrated into signal processing schematics as subnetworks. This completes the interface between electrical signals in VPI-TC and the circuit-diagram schematics of ADS.

1. Specifically, Time-Domain Waveform Data (.tim) files in MDIF ASCII format are used. For more details, see the Agilent Ptolemy Simulation manual.

Design Procedure

A working ADS analog/RF schematic should be created, which will become part of a VPI-TC simulation.

ADS Schematic

1. Create a subnetwork of the component and add ports for all input and output signals. Input signals to ADS are interpreted as ideal voltage sources by the circuit simulator. Note that only open voltages can be made output signals. If your circuit needs currents as inputs or outputs you can use ideal voltage-controlled current sources to transform an input voltage to an input current or ideal current-controlled voltage sources to transform an output current to an output voltage.
2. Set the start time to zero and the stop time to the value of the global parameter **`TimeWindow`** in VPI-TC. The max time step must be less than or equal to 1/**`SampleRate`** in VPI-TC.
3. Create a wrapper setup in ADS containing a ***TimedDataRead*** and a ***TimedDataWrite*** module for each input and output port, respectively.

> Note: ***TimedDataRead*** and ***TimedDataWrite*** modules are part of ADS signal processing library.
> More details about the ***TimedDataRead*** module can be found in ADS documentation under Home > Components > Components-Signal Processing > Timed Components > Timed Sources.
> For more details about the ***TimedDataWrite*** module, refer to ADS documentation under Home > Components > Components-Signal Processing > Sinks.
>
> One easy way to create an ADS schematic is to reuse an existing ADS workspace provided together with the cosimulation demonstrations available at *Optical Systems Demos > Simulation Techniques > Cosimulation > ADS.*

4. Set the **`FileName`** parameters to the names of the corresponding input or output files.

   The input files generated by the interface follow the naming convention:
   `TC_ADSInSignal`<*Port_number*>`.tim`.
   The output files generated by ADS must follow the naming convention:
   `TC_ADSOutSignal`<*Port_number*>`.tim`.
   (For details refer to the *Photonic Modules Reference Manual* entry for the ***ADS_Interface*** module.)
5. Set the **`ControlSimulation`** parameter to `YES` for all sources and sinks.
6. Set the ***TimedDataRead*** parameters **`T_Step`** and **`Period`** to zero so that the values calculated from the file data are used.
7. Wire the wrapper setup and insert a data flow controller.
8. Set the wrapper setup parameters **`DefaultTimeStart`** to zero and **`DefaultTimeStop`** to the value of the VPI-TC global parameter **`TimeWindow`**.

> Note: To ensure the **`DefaultTimeStop`** parameter is always up-to-date, pass it as a parameter from VPI-TC by setting the **`ADS_Parameter`** of the ***ADS_Interface*** module to `DefaultTimeStop={TimeWindow}`.
>
> Set the **`Stop`** parameter in the ***TimedDataWrite*** module(s) to `DefaultTimeStop`.

9. Try to run the wrapper setup in order to generate the `netlist.log`. Though it will break with an error message because it cannot find the input files yet, the required `netlist.log` file will be generated.

VPItransmissionMaker™/VPIcomponentMaker™ Schematic

1. Place an instance of the ***ADS_Interface*** module in the schematic. (This module can be found in the *Cosimulation* folder under TC Modules).
2. In the parameter editor (double-click on the module) set the parameters **`N_Inputs`** and **`N_Outputs`** to the number of input and output ports of the electrical circuit in ADS.

> Note: There must be at least one input and one output for correct scheduling. If an input port is not required, a dummy input port must be created and fed with a dummy signal, e.g., from a DC_Source. Likewise, an unused output port can be fed into the Ground modules.

3. Connect the input and output ports of the ***ADS_Interface*** to the neighboring modules. If the electrical component has multiple inputs or outputs, use buses and the bus manipulation modules to form and split buses. Those are located in the *Wiring Tools* folder.
4. Set the **`ADS_ProjectDirectory`** parameter to the location of the ADS workspace containing the schematic of the electrical component.
5. Set the **`SampleRate`** parameter to the sample rate of the input and output signals. All interface signals must have the same sample rate.
6. Set the galaxy parameter **`TimeWindow`** in the ***ADS_Interface*** module to the value of the VPI-TC global parameter **`TimeWindow`**.
7. Specify any parameter that should be passed to ADS in the galaxy parameter **`ADS_Parameter`**.

   Use the following syntax:
   `<ads_component_id>.<parameter_name>= <value>`
   and
   `<ads_global_parameter_name>=<value>`
   or
   `global.<ads_global_parameter_name>=<value>`

   Multiple assignments must be separated with semicolons. Names wrapped in braces { } are substituted with the values of the corresponding global parameter in VPI-TC.

Now the simulation can be started in VPI-TC. It will call ADS when needed, and use the `netlist.log` file to define the electronic circuit.

Application Example

Modulator Driver with Preamplifier and Travelling Wave Amplifier

This example shows an example of a RF broadband amplifier, employed for driving a MZM to generate a PSK signal. The RF amplifier, designed in ADS, is composed by a preamplifier and a travelling wave amplifier.

Simulation Setup in VPItransmissionMaker™/VPIcomponentMaker™

Figure 1 shows the simulation setup in VPI-TC. A data stream is generated at 40Gb/s. Electrical data+ and data- are generated by the code drivers and sent to the ***ADS_Interface*** module, representing the RF amplifier. Data signals to the amplifier are typically in the order of several milivolt. The driver amplifier adjusts the signals for driving the MZM, which requires voltages of several volt. The amplified signals are used as inputs for the dual-drive MZM, which genenerated the PSK-modulated optical signal.

The receiver consists of a MZI and two PIN photodiodes (only considering noise).

The ADS schematic was adopted from the existing file provided as a Design Guide in ADS. The parameters passed to the amplifier are: the temperature of the amplifier, the simulation time and the step time.

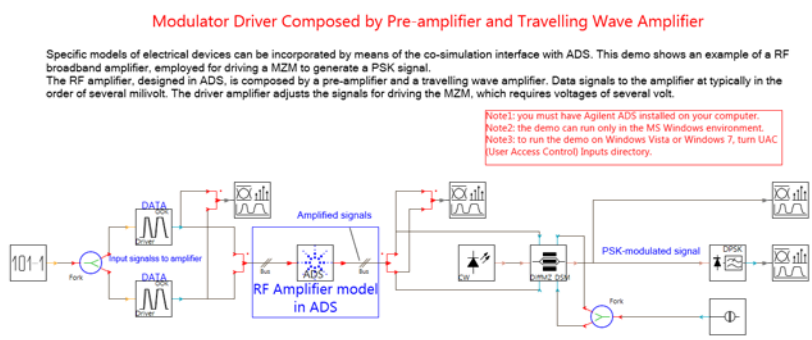

Figure 1 *PSK system using a RF driver amplifier in ADS*

Note: There is a restriction in the current implementation of the Cosimulation interface that makes it impossible to pass logical information through the DLL. For BER measurements, it is necessary to pass reference information separately around the DLL and use the clock recovery and BER modules that have a reference input – see the demo ‘Direct Detection Example’.

ADS Workspace `ADS-Driver_wrk`

The ADS workspace contains a hierarchy of circuits (cells) used to design the amplifier. Figure 2 shows the schematic of the cell `Laser_Driver`, which contains a preamplifier, a travelling wave amplifier and the transient simulator controller. Figure 3 shows the transient analysis settings.

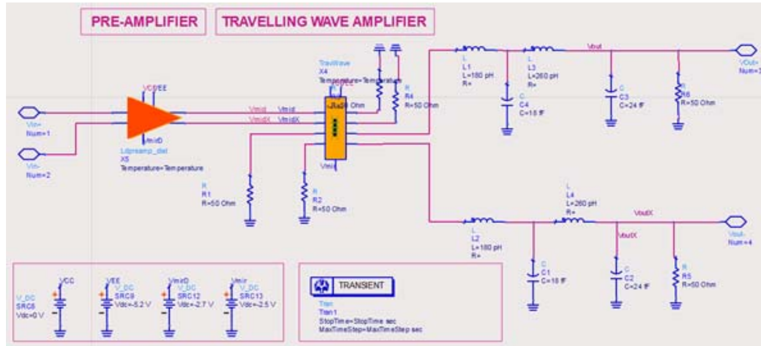

Figure 2 *ADS setup of TIA as subnetwork*

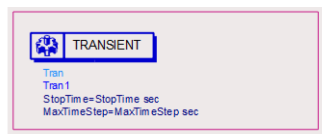

Figure 3 *Transient controller parameter settings and voltage-to-current conversion in the TIA example*

The `Laser_Driver` cell needs to be included as a subnetwork in a wrapper setup. This wrapper setup provides a standardized environment for reading in the electrical input signals from VPI-TC and for writing to files the electrical output signals of the ADS circuit design, which will be read by VPI-TC.

Figure 4 shows the wrapper setup.

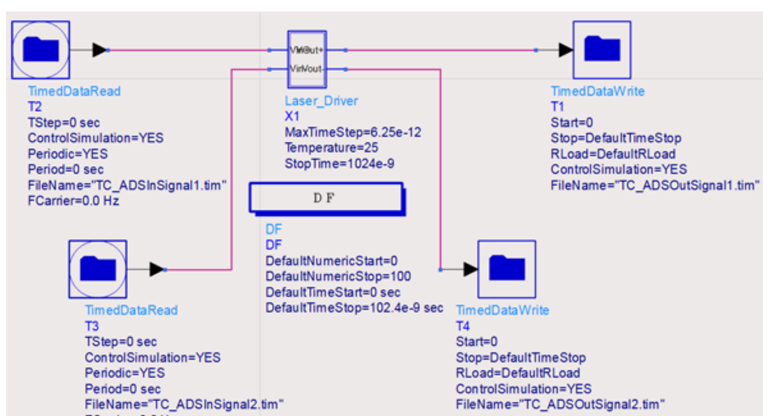

Figure 4 *Wrapper setup for the laser driver*

Note: Before each simulation run, the global variables **`MaxTimeStep`**, **`StopTime`** and the **`Temperature`** parameter of the ADS setup `Laser_Driver` are replaced with the parameter values passed from VPI-TC through the interface.

Passing Parameters from VPI-TC to ADS

The *ADS_Interface* module used in the VPI-TC setup uses the following string in the **`ADS_Parameter`**:

```
DefaultTimeStop= {TimeWindow};
X1.MaxTimeStep=1/{SampleRateDefault};X1.Temperature={T}
```

where `X1` is the identifier of the `Laser_Driver` cell in the ADS schematic, *`{TimeWindow}`*, *`{T}`* and *`{SampleRateDefault}`* are global parameters in VPI-TC, and **`DefaultTimeStop`** is a global variable in ADS.

> Note: Only the values of parameters are replaced in the net list. Optionally following units in the net list remain unchanged. If you specify a parameter without a unit in the **`ADS_Parameter`** string, it is interpreted in the units of the corresponding default parameter setting in the ADS schematic. For example, if the parameter **`DefaultTimeStop`** is set to `1024 nsec` in ADS and **`ADS_Parameter`** is set to `DefaultTimeStop=1024e-9` in VPItransmissionMaker, the result is `DefaultTimeStop = 1024e 9 nsec`, which is not what was intended. Therefore, always use MKS (meters, kilograms, seconds) SI units in ADS for parameters that should be modified by VPI-TC. That is, do not use kilometers, nanoseconds, or gigahertz, etc.

Running the Simulation in VPItransmissionMaker™/VPIcomponentMaker™

Run the setup for different temperatures to see the effect on the resultant signal.

The results for some of the temperatures are displayed below in Figure 5.

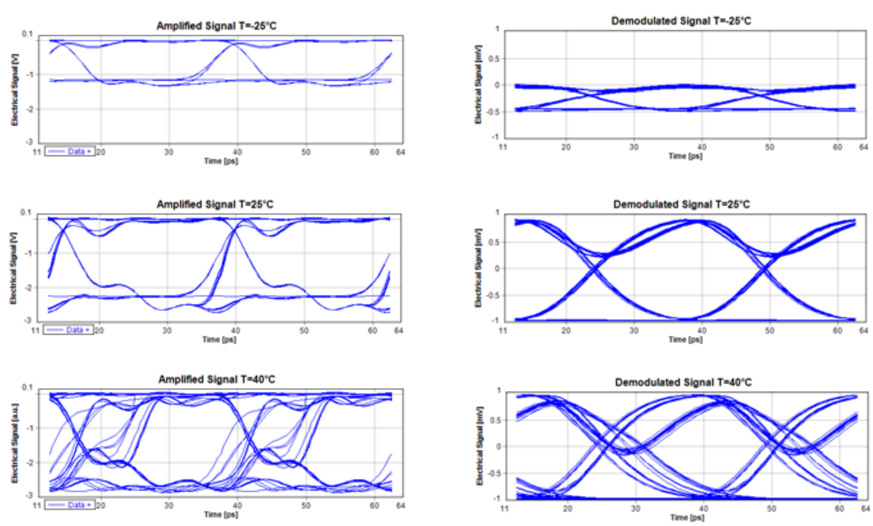

Figure 5 *Electrical eye diagram at different temperatures. Left: Amplified signal after the RF driver. Right: Signal after photodetection.*

Figure 5 shows eye diagrams of the amplified and demodulated electrical signal for temperatures of -25°C, 25°C and 40°C. The performance is limited at low powers by the noise in the system (mainly originating from the optical amplifiers and the photodiode), and at high powers by the nonlinearity of the RF amplifier.

