



Hands-on Tutorial



RapidWright: Unleashing the Full Power of FPGA Technology with Domain-Specific Tooling

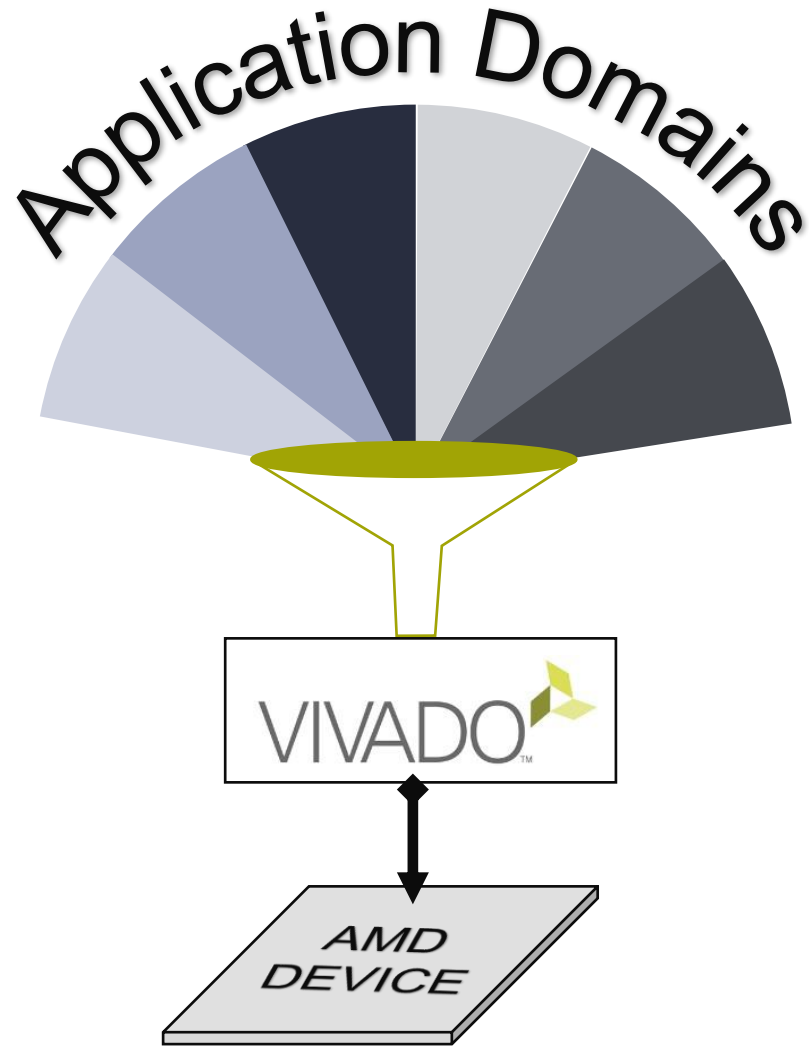
Chris Lavin, chris.lavin@amd.com

Eddie Hung, eddie.hung@amd.com

AMD Research and Advanced Development

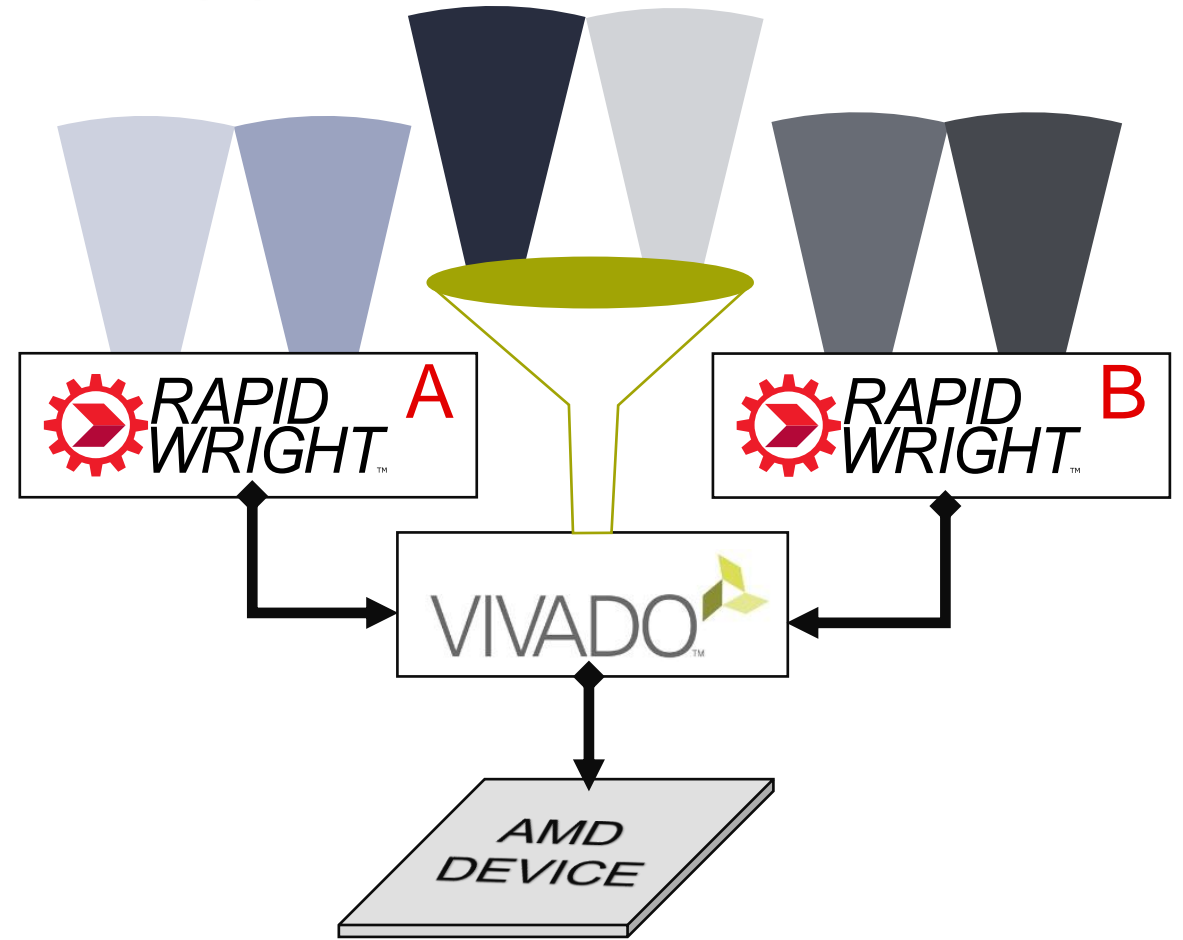
November 1, 2023
IEEE/ACM ICCAD 2023
San Francisco, CA USA

Domain Specific Era Needs Domain Specific Backends



Vivado must **generalize** solutions

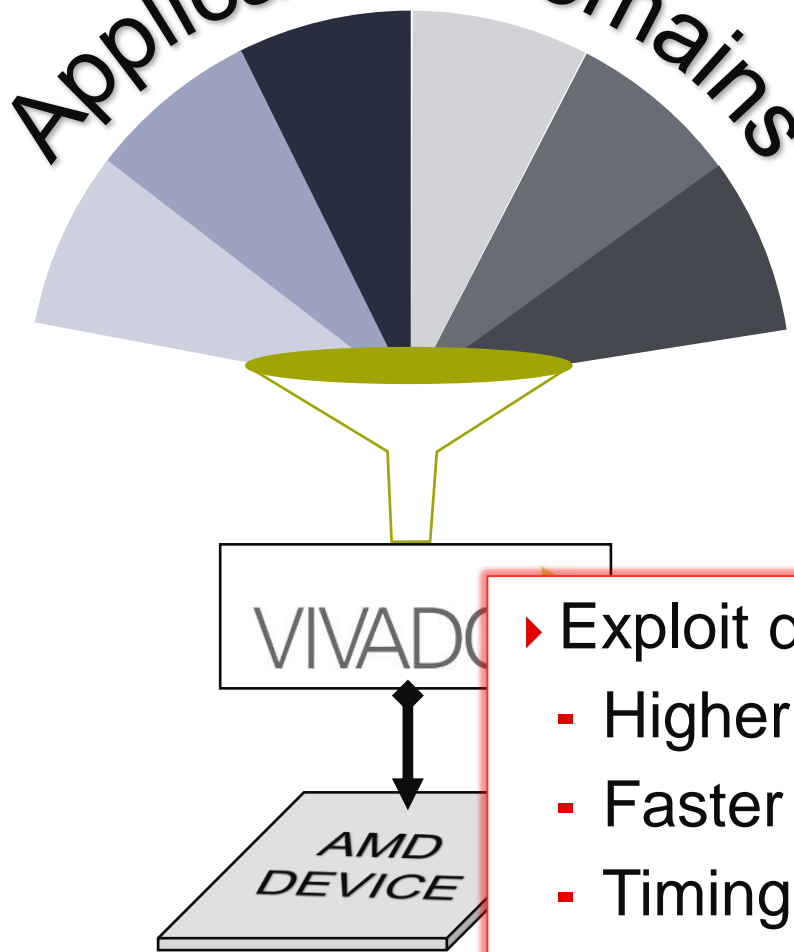
Application Domains



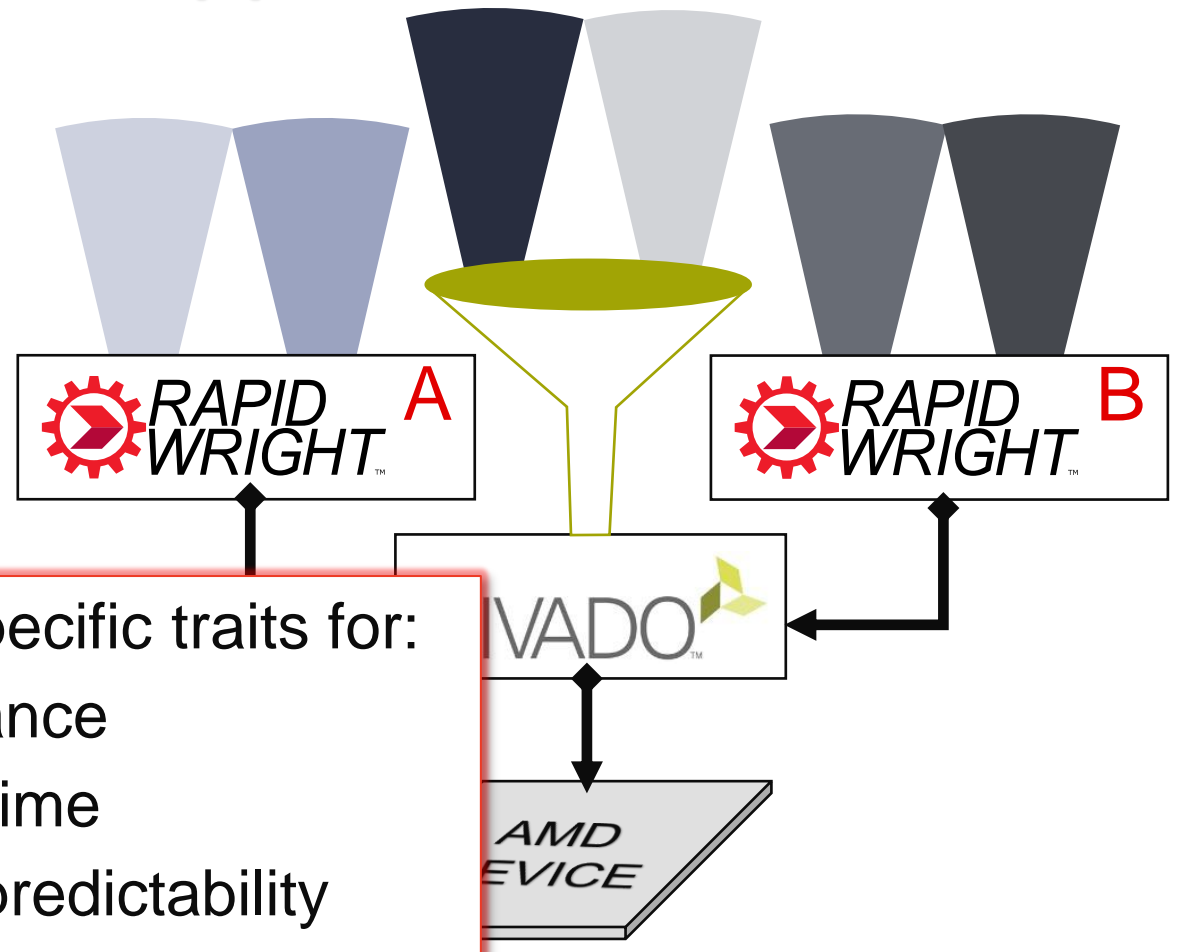
RapidWright can **specialize**

Domain Specific Era Needs Domain Specific Backends

Application Domains



Application Domains



- ▶ Exploit domain-specific traits for:
 - Higher performance
 - Faster compile time
 - Timing closure predictability

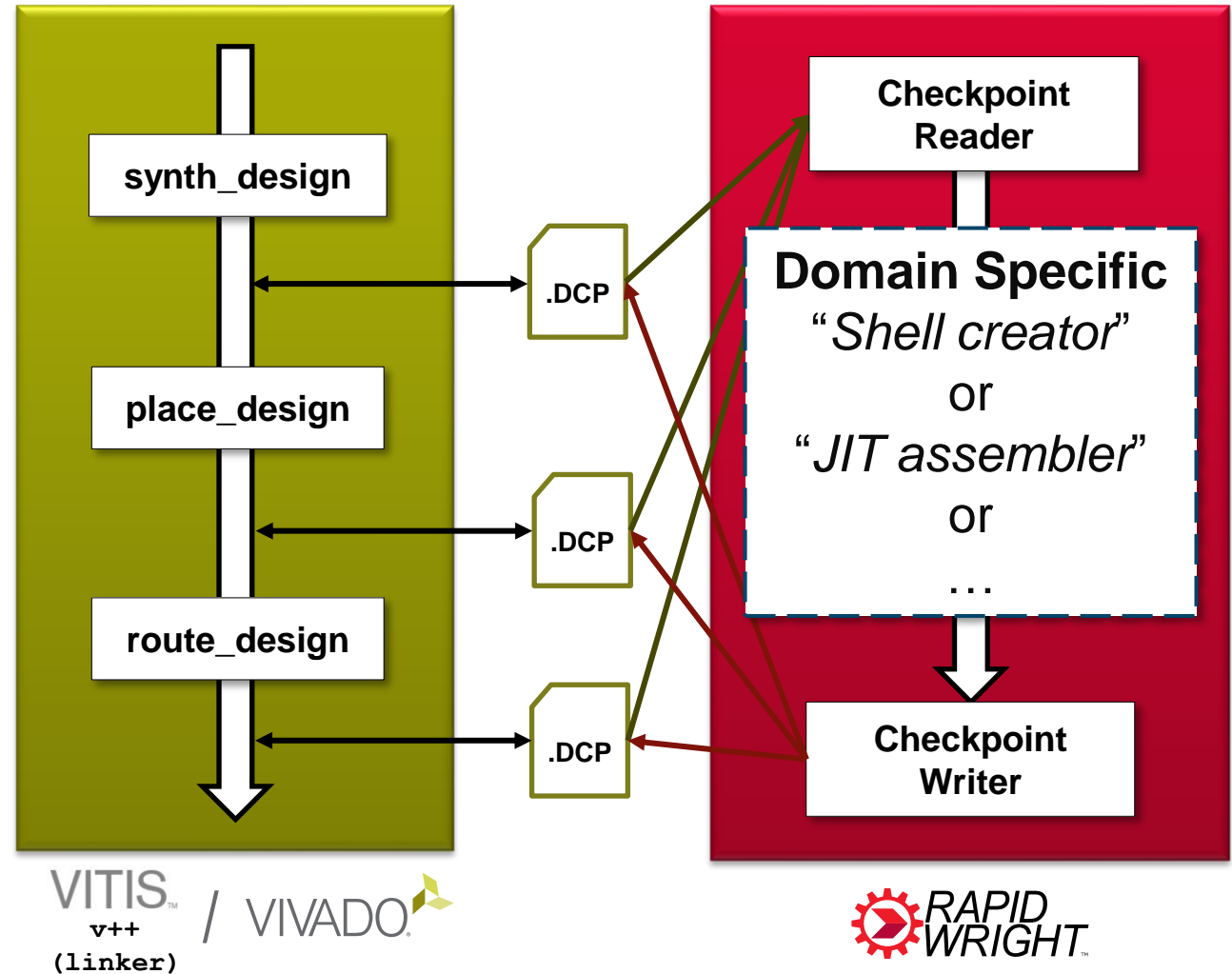
Vivado must **generalize** solutions

RapidWright can **specialize**

What is RapidWright?

- ▶ Companion framework for Vivado
 - Fast, light-weight, open source
 - Java code, Python scripting
 - All public devices supported
- ▶ Enables targeted solutions
 - Reuse & relocate pre-implemented modules
 - Create shells & overlays
- ▶ Power user ecosystem
 - Customized bottom-up flows
 - Rapid prototyping of CAD concepts

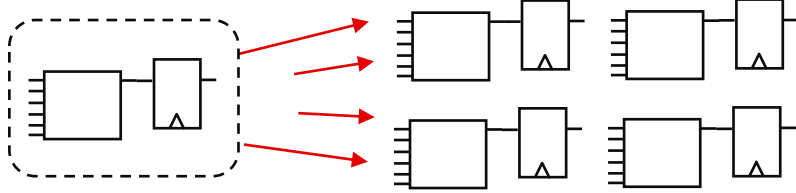
```
pip install rapidwright
```



DCP = Design Checkpoint (Netlist, Placement & Routing Data)

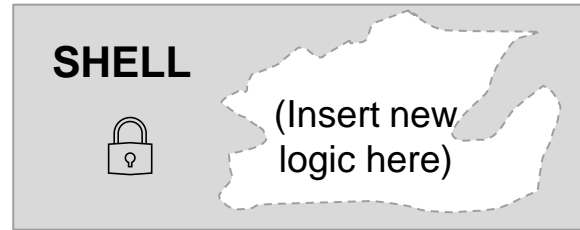
RapidWright Differentiators

REUSE



Copy & Paste

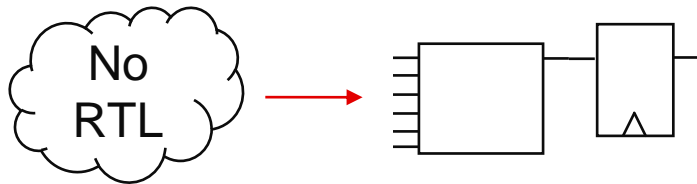
Replicate and relocate placed & routed circuits



Reuse Timing Closure

Create non-standard shells to preserve difficult-to-close circuits

SPEED



Circuits in Seconds

Generate well-formed P&R circuits from scratch in seconds



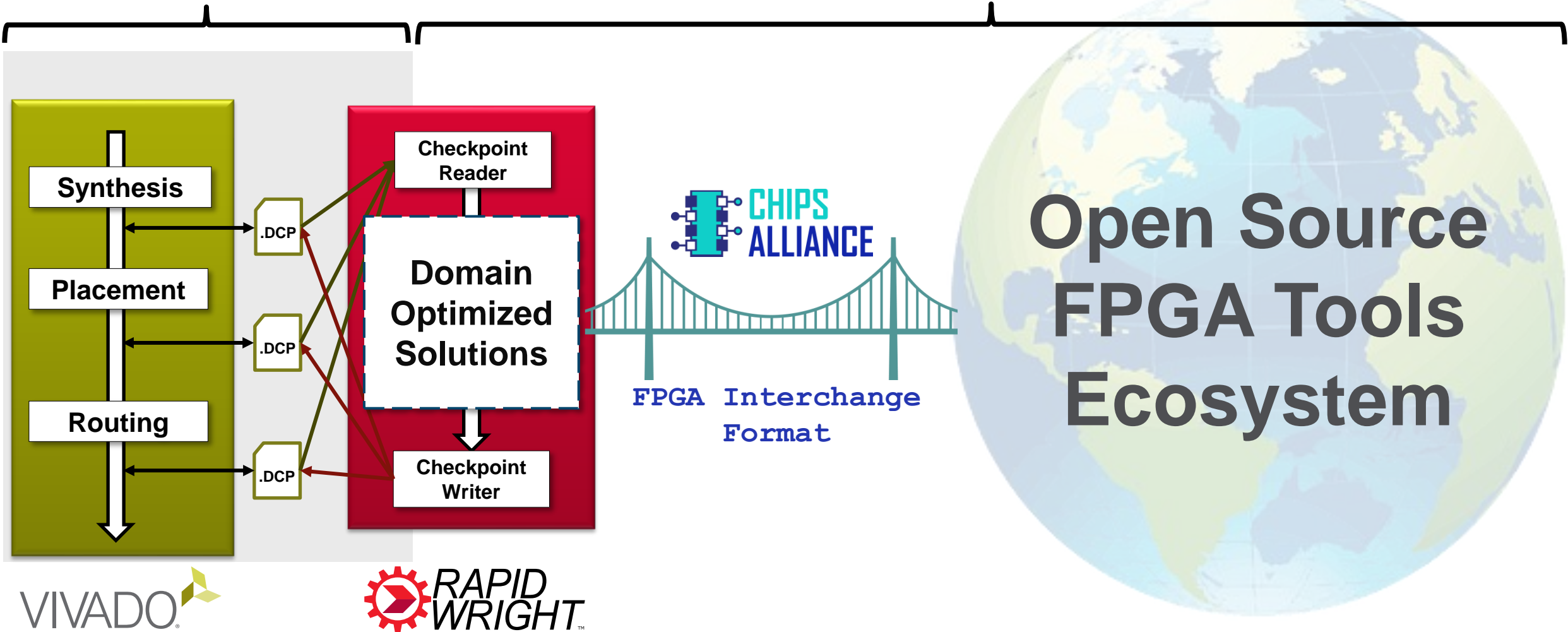
Greater Agility

Explore new P&R algorithms, faster design & ECO changes

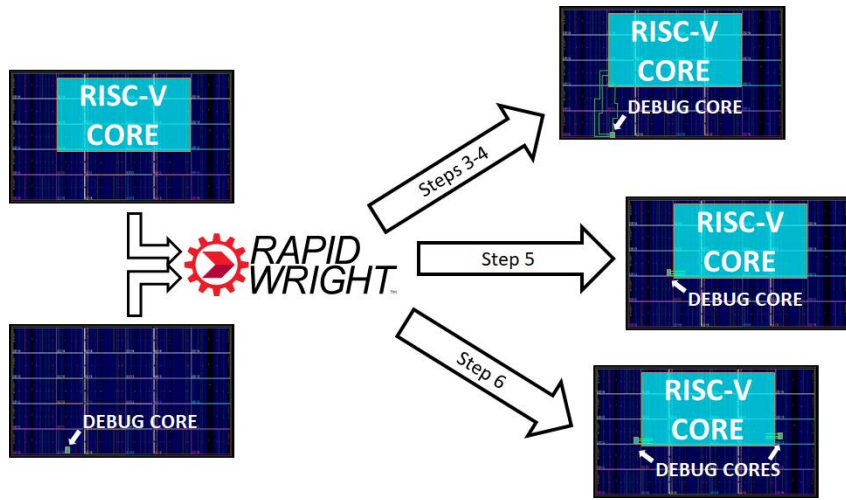
Backend Open Source & Open Standards Efforts

AMD Tools

Open Source



Inserting and Routing a Debug Core As An ECO



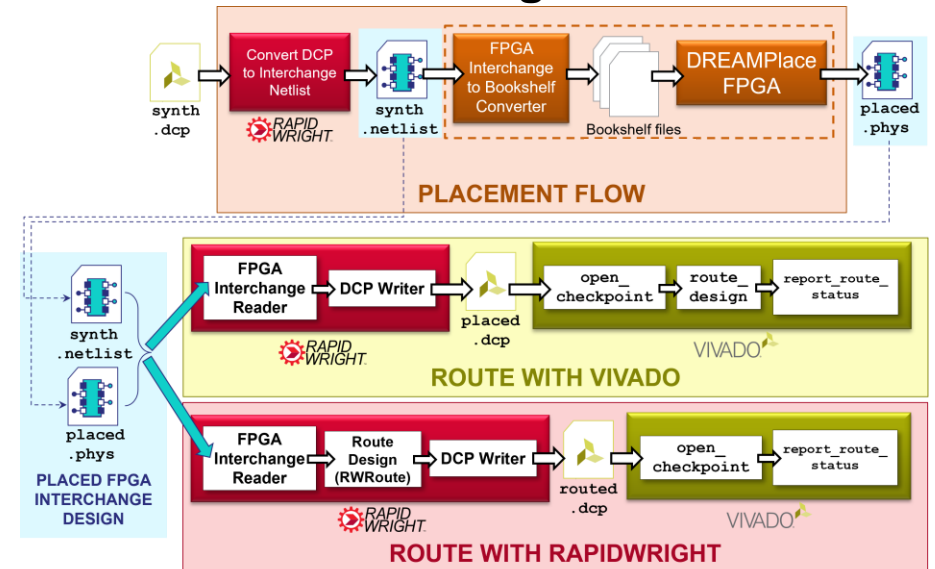
Reuse Timing-closed Logic As A Shell

The screenshot shows a Netlist window with device information and a diagram of static and dynamic logic. The diagram is titled "STATIC LOGIC" and "DYNAMIC LOGIC". The static logic examples include Ethernet MAC, PCIe, and DDR Controller. The dynamic logic examples include Accelerators, custom logic, and modules under development. A note indicates "* DONT_TOUCH = 'TRUE' *". The device name "top_cell" is visible at the bottom.

Pre-implemented Modules (Part I & II): PicoBlaze Array

The left screenshot shows a PicoBlaze array with a yellow box highlighting the "Pblock Boundary" and pink arrows pointing to "Side Loads (Used Stubs)". The right screenshot shows a "396 Replicated PicoBlaze Array" with a grid of green and blue blocks.


Use DREAMPlaceFPGA to Place a Netlist via FPGA Interchange Format



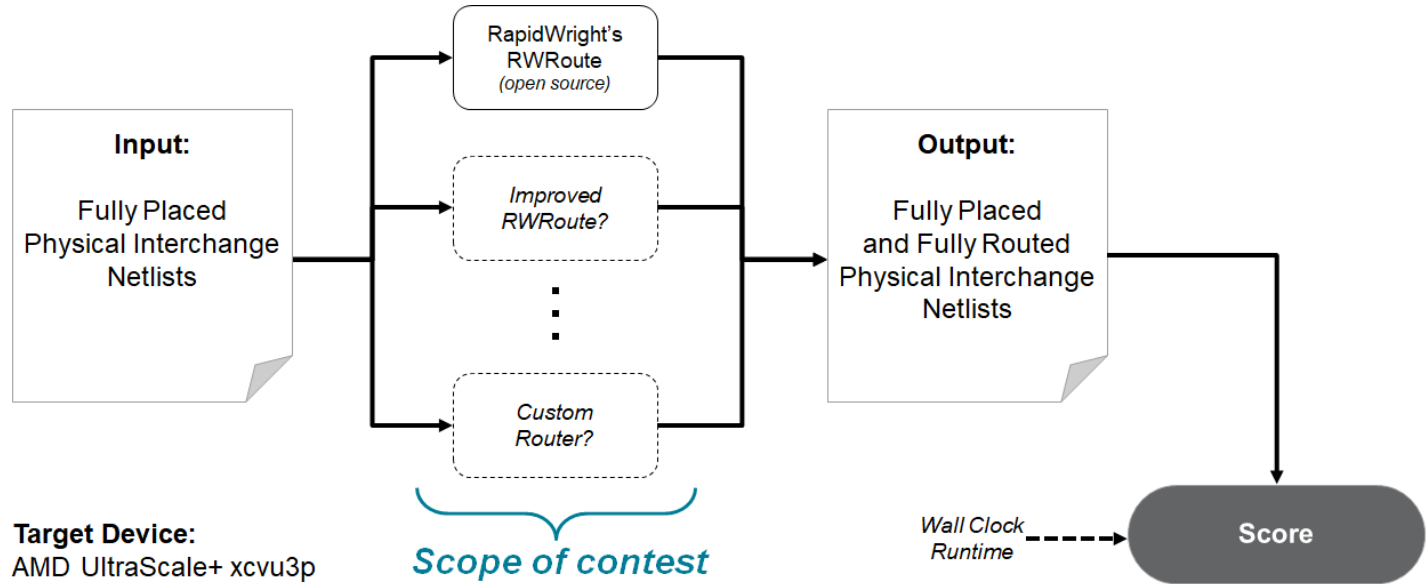
ISFPGA'24 Runtime-first FPGA Interchange Routing Contest

How fast can we route an FPGA if we pull out all the stops?

Built on:



FPGA Interchange Format



Rank	Prize (USD)
1st	\$2500
2nd	\$1500
3rd	\$1000
4th & 5th	\$500

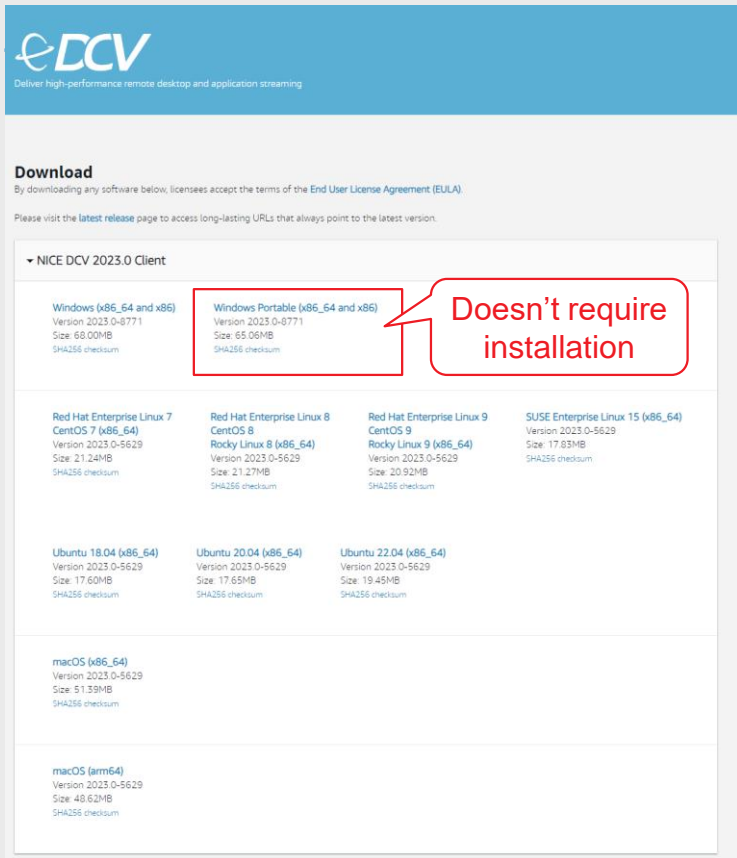
Target Device:
AMD UltraScale+ xcvu3p

Registration open until Nov 20:

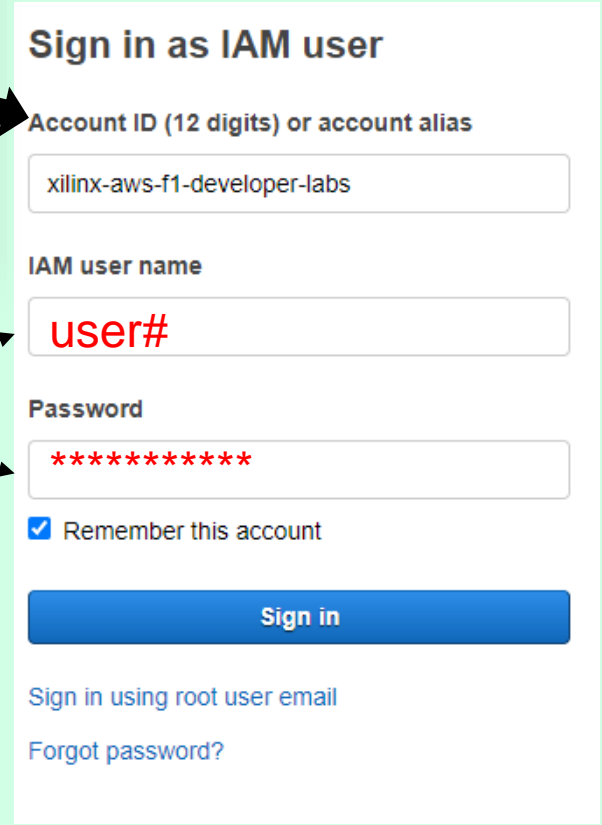
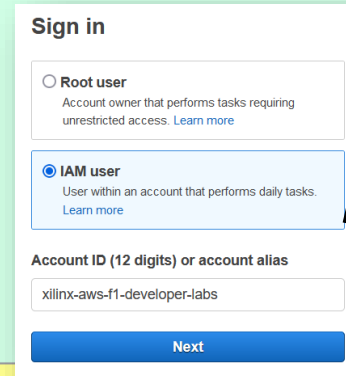
https://xilinx.github.io/fpga24_routing_contest

1 & 2: Install Client & Log in to AWS Console

1. Download & extract/install NICE Desktop Cloud Visualization License Client (to access AWS instance) <https://download.nice-dcv.com/>




2. Log in to the AWS EC2 Console <https://console.aws.amazon.com/ec2/>

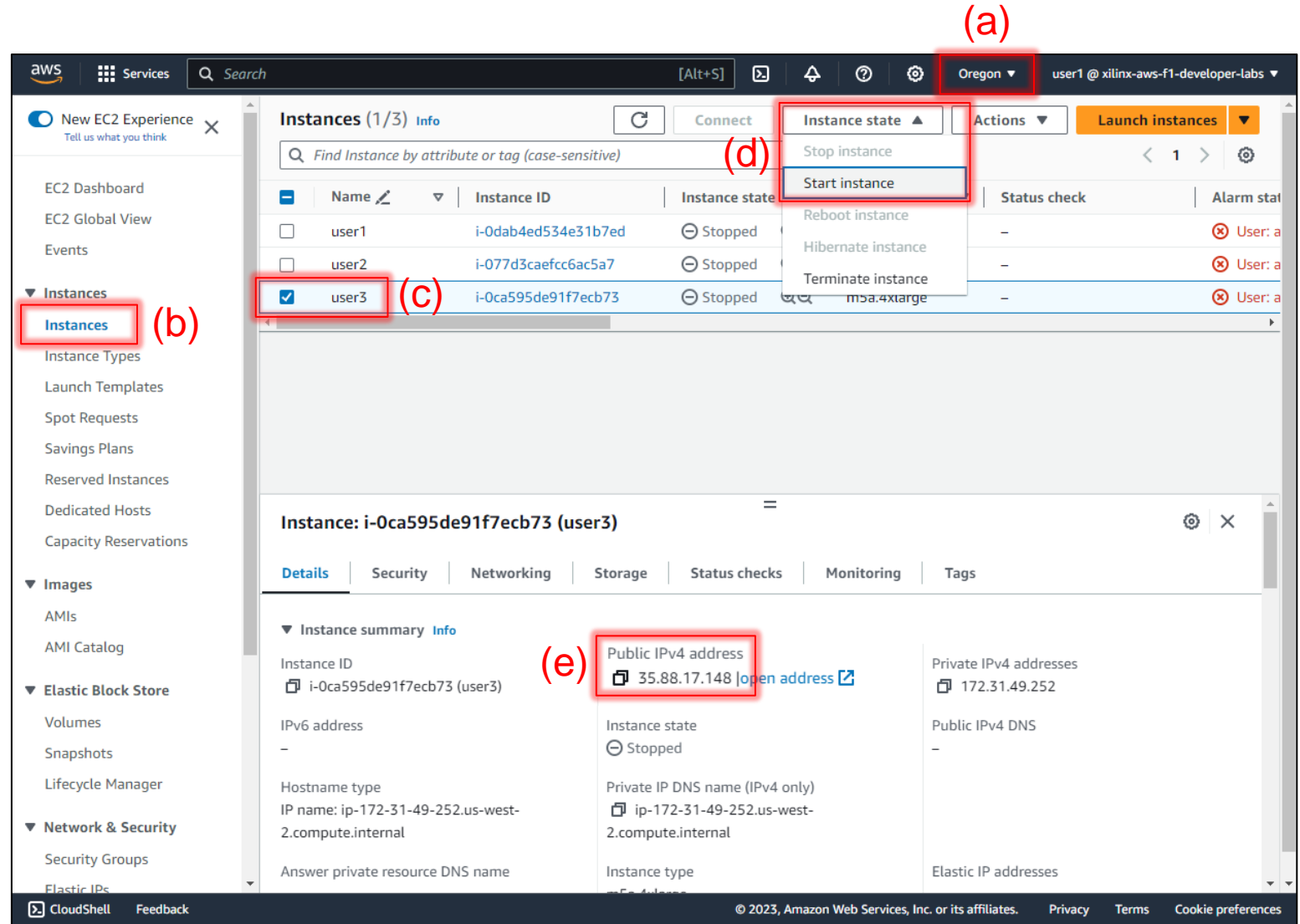
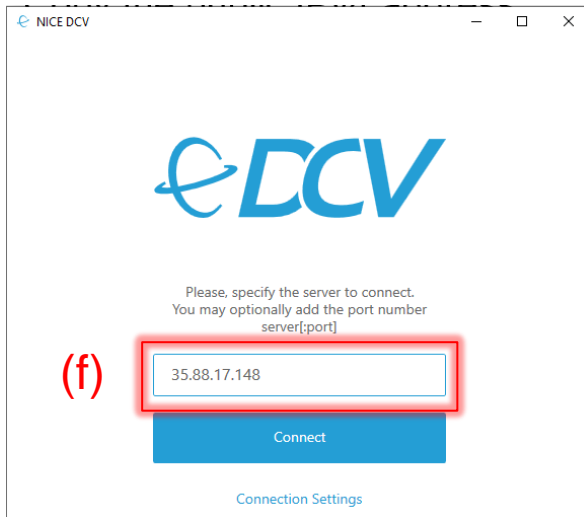


Please see Eddie or Chris if you did not receive this yet



3. Start AWS EC2 Instance and Connect

- a) Ensure “Oregon” region
- b) Click on “Instances”
- c) Select your named user instance (e.g., ‘user3’)
- d) Select “Start instance” from the “Instance state” menu
- e) Copy the public IPv4 address (click the  icon)
- f) Open NICE DCV Client, Paste IP address in window, click Connect




The screenshot shows the AWS Management Console. The top navigation bar shows the "Oregon" region selected. The left sidebar has the "Instances" menu item highlighted. The main content area shows a table of instances. The instance "user3" with ID "i-0ca595de91f7ecb73" is selected. A context menu is open over this instance, and the "Start instance" option is highlighted. Below the table, the details for instance "i-0ca595de91f7ecb73 (user3)" are shown. The "Public IPv4 address" is listed as "35.88.17.148" with a copy icon. A red box highlights the public IPv4 address and the copy icon. The label "(e)" is placed to the left of the address. Other labels (a, b, c, d) are placed on the console interface to indicate other steps in the process.

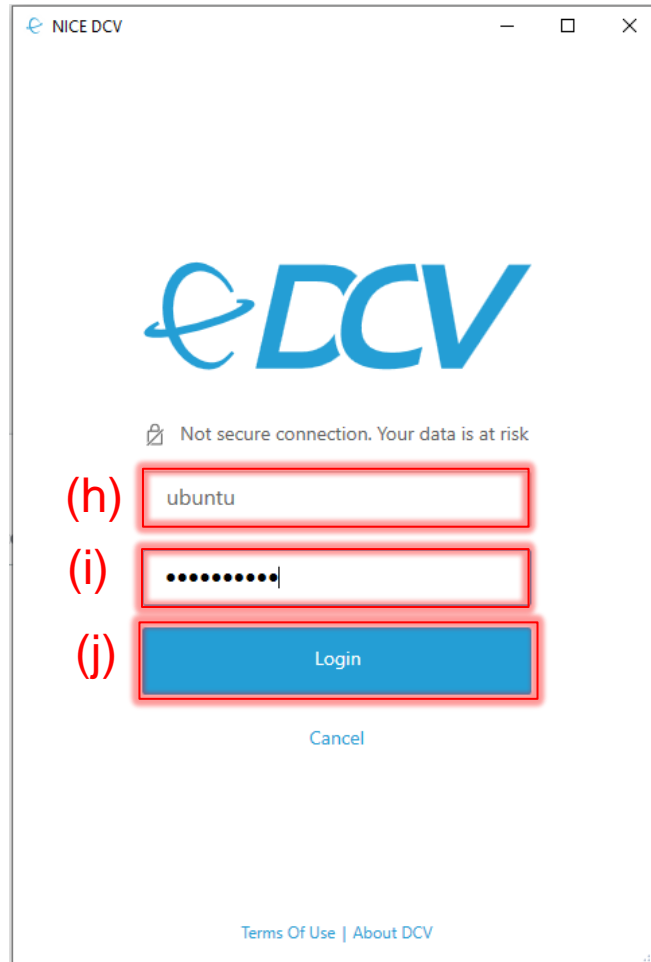
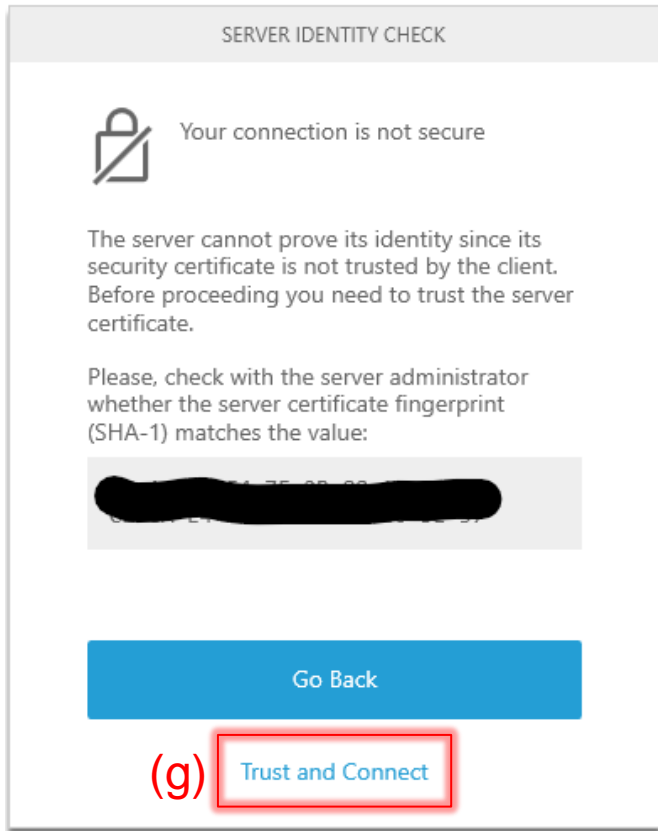
Name	Instance ID	Instance state	Status check	Alarm status
user1	i-0dab4ed534e31b7ed	Stopped	-	User: a
user2	i-077d3caefcc6ac5a7	Stopped	-	User: a
user3	i-0ca595de91f7ecb73	Stopped	-	User: a

Instance: i-0ca595de91f7ecb73 (user3)

Public IPv4 address: 35.88.17.148

3. Log Into NICE DCV (cont.)

- g) Choose “Trust and Connect”
- h) Enter username: “ubuntu”
- i) Enter Password: <same as AWS Console>
- j) Click “Login”
- k) Click on  to launch the Tutorial Page



Introduction

Getting Started

FPGA Architecture Basics

Xilinx Architecture Terminology

RapidWright Overview

Design Checkpoints

Implementation Basics

Merging Designs

Bitstream Manipulation

FPGA Interchange Format

RapidWright Publications

A Pre-implemented Module Flow

▣ RapidWright Tutorials

RWRoute Timing-driven Routing

RWRoute Wirelength-driven Routing

RWRoute Partial Routing

RapidWright Report Timing Example

Reuse Timing-closed Logic As A Shell

Use DREAMPlaceFPGA to Place a Netlist via FPGA Interchange Format

Polynomial Generator: Placed and Routed Circuits in Seconds

Inserting and Routing a Debug Core As An ECO

Create Placed and Routed DCP to Cross SLR

Build an IP Integrator Design with Pre-Implemented Blocks

RapidWright PipelineGenerator Example

RapidWright PipelineGeneratorWithRouting Example

RapidWright ICCAD 2023 Hands-on Tutorial

Title: RapidWright: Unleashing the Full Power of FPGA Technology with Domain-Specific Tooling





Organizers: Chris Lavin and Eddie Hung

Where: Artisan Room, Hyatt Regency San Francisco Downtown SOMA, [ICCAD 2023](#)

When: Wednesday, November 1st, 2023, 11:00am PDT

- 11:00am - 11:05am : Machine Allocation
- 11:05am - 11:15am : Introduction and Overview
- 11:15am - 1:00pm : Hands-on, self-guided tutorials

Featured Tutorial Segments	Time	Description
Reuse Timing-closed Logic As A Shell	30 mins	Create a pre-implemented shell from an existing design without pblocks
Using DREAMPlaceFPGA to Place	25 mins	Use a 3rd party placer with the FPGA Interchange Format
Polynomial Generator	15 mins	Create placed and routed circuits from scratch in seconds
ECO Debug Core Insertion	35 mins	Add debug logic without changing existing placement and routing

Additional Tutorial Segments	Time	Description
Hello, World 	5 mins	Intro to RapidWright in Jupyter Notebooks
Create Netlist from Scratch 	10 mins	How to build a netlist from scratch
Pre-implemented Modules: Part I	15 mins	How to create a pre-implemented module
Pre-implemented Modules: Part II	15 mins	Use & relocate pre-implemented modules
SAT Router 	15 mins	Use SAT to solve hard routing congestion
Create and Use an SLR Bridge	20 mins	Combine Vivado & RapidWright circuits
Basic Routing 	20 mins	How to build a basic router in RapidWright

 = Jupyter Notebook Tutorial

URL:
rapidwright.io/docs/ICCAD23_Tutorial.html

COPYRIGHT AND DISCLAIMER

©2023 Advanced Micro Devices, Inc. All rights reserved.

AMD, the AMD Arrow logo and combinations thereof are trademarks of Advanced Micro Devices, Inc. Other product names used in this publication are for identification purposes only and may be trademarks of their respective companies.

The information presented in this document is for informational purposes only and may contain technical inaccuracies, omissions, and typographical errors. The information contained herein is subject to change and may be rendered inaccurate releases, for many reasons, including but not limited to product and roadmap changes, component and motherboard version changes, new model and/or product differences between differing manufacturers, software changes, BIOS flashes, firmware upgrades, or the like. Any computer system has risks of security vulnerabilities that cannot be completely prevented or mitigated. AMD assumes no obligation to update or otherwise correct or revise this information. However, AMD reserves the right to revise this information and to make changes from time to time to the content hereof without obligation of AMD to notify any person of such revisions or changes.

THIS INFORMATION IS PROVIDED 'AS IS.' AMD MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND ASSUMES NO RESPONSIBILITY FOR ANY INACCURACIES, ERRORS, OR OMISSIONS THAT MAY APPEAR IN THIS INFORMATION. AMD SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. IN NO EVENT WILL AMD BE LIABLE TO ANY PERSON FOR ANY RELIANCE, DIRECT, INDIRECT, SPECIAL, OR OTHER CONSEQUENTIAL DAMAGES ARISING FROM THE USE OF ANY INFORMATION CONTAINED HEREIN, EVEN IF AMD IS EXPRESSLY ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

AMD 