

Intel FPGA Cloud Services

Larry Landis, Senior Manager University Outreach, Intel Programmable Solutions Group



- Larry Landis – University Outreach Senior Manager, Intel Programmable Solutions Group
- 31 years in semiconductor industry – design, sales, marketing, training for ASIC, FPGA
- Adjunct Lecturer for Digital Electronics Course – Santa Clara University



BSEE



MSEE



Adjunct Lecturer



Intel® FPGA Academic Ecosystem



- Train the next generation of FPGA Designers
- Increase Intel® FPGA presence in academia



- Academic access to the latest generation of Intel FPGAs



- Nurture the talent pipeline for Intel and our customers
- Engage research on Intel FPGAs

Undergraduate Teaching Resources for Computer Engineering Profs and TAs



intel[®]

INTEL® FPGA EE Undergrad Coursework Offerings

- Undergraduate
 - Digital Logic
 - Digital Systems
 - Computer Organization
 - Embedded Systems

What's included?

Tutorials on tool usage

Semester worth of labs

<http://fpgauniversity.intel.com>



Remote Learning Tools for Undergrad Coursework

■ DESim

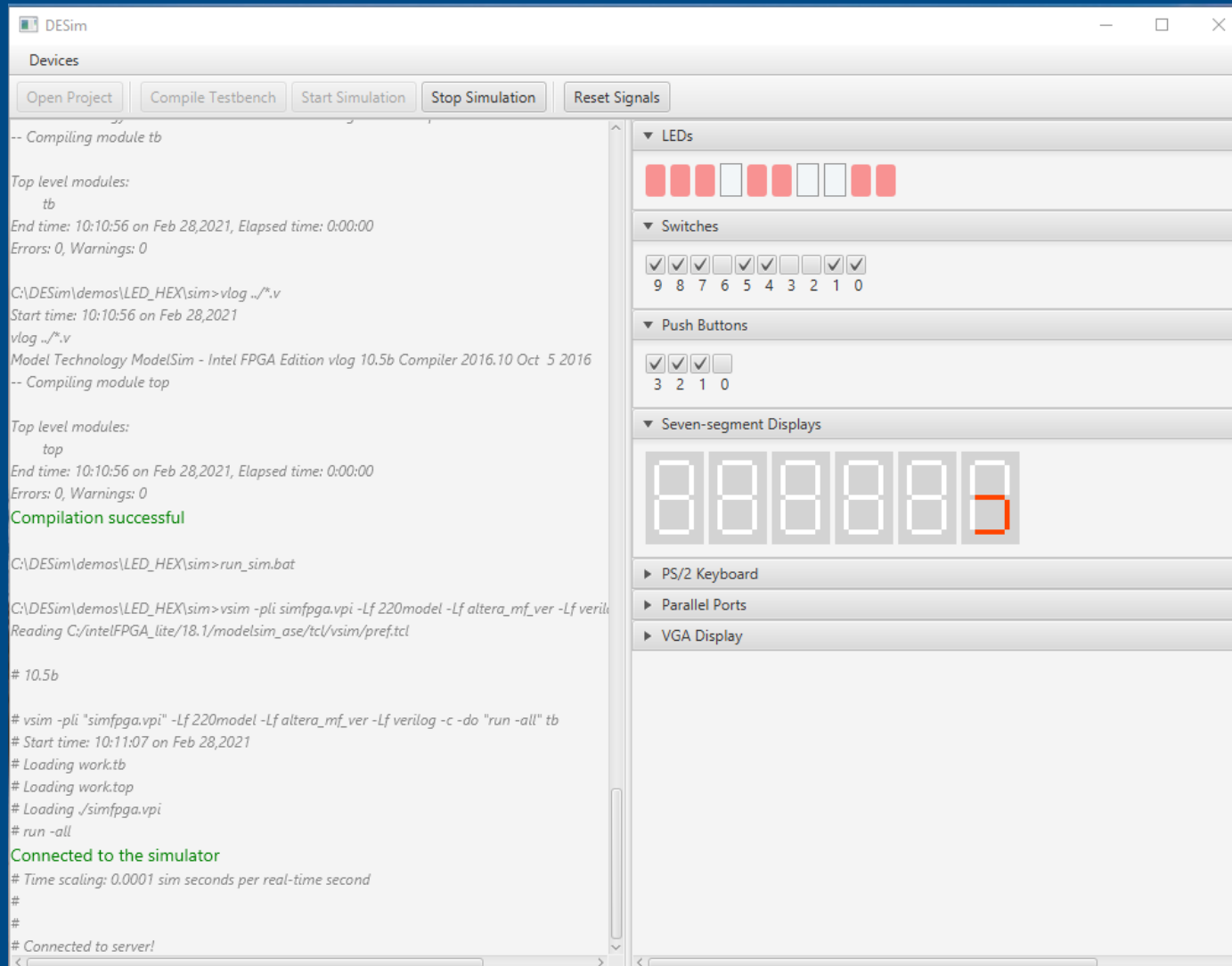
- Devkit like GUI that that runs Modelsim under the hood. Great for first time learners of Verilog/VHDL for stimulus/response type labs. No hardware required.

■ Remote Console

- “Video game” like GUI that connects to Terasic development kits
- Enables remote hosting of boards – ideal for work from home environments



DESIm



■ DESIm

- Student prepares Verilog/VHDL code
- Prof provides testbench and student can observe behavior through GUI – great for new learners to visualize logic design behavior

<https://github.com/fpgacademy/DESIm/releases/tag/v1.0.1>

Remote Console



Linux or Windows*
Server



USB



Method 1: Intel Quartus® Prime Software Hosted on Server

-or-

Method 2: Intel Quartus® Prime Software Hosted on Student's PC

Does not require 1 Devkit per Server – use USB port replicator

Install setup at university engineering department cluster

Host on Windows or Linux server running Intel Quartus Prime Programmer

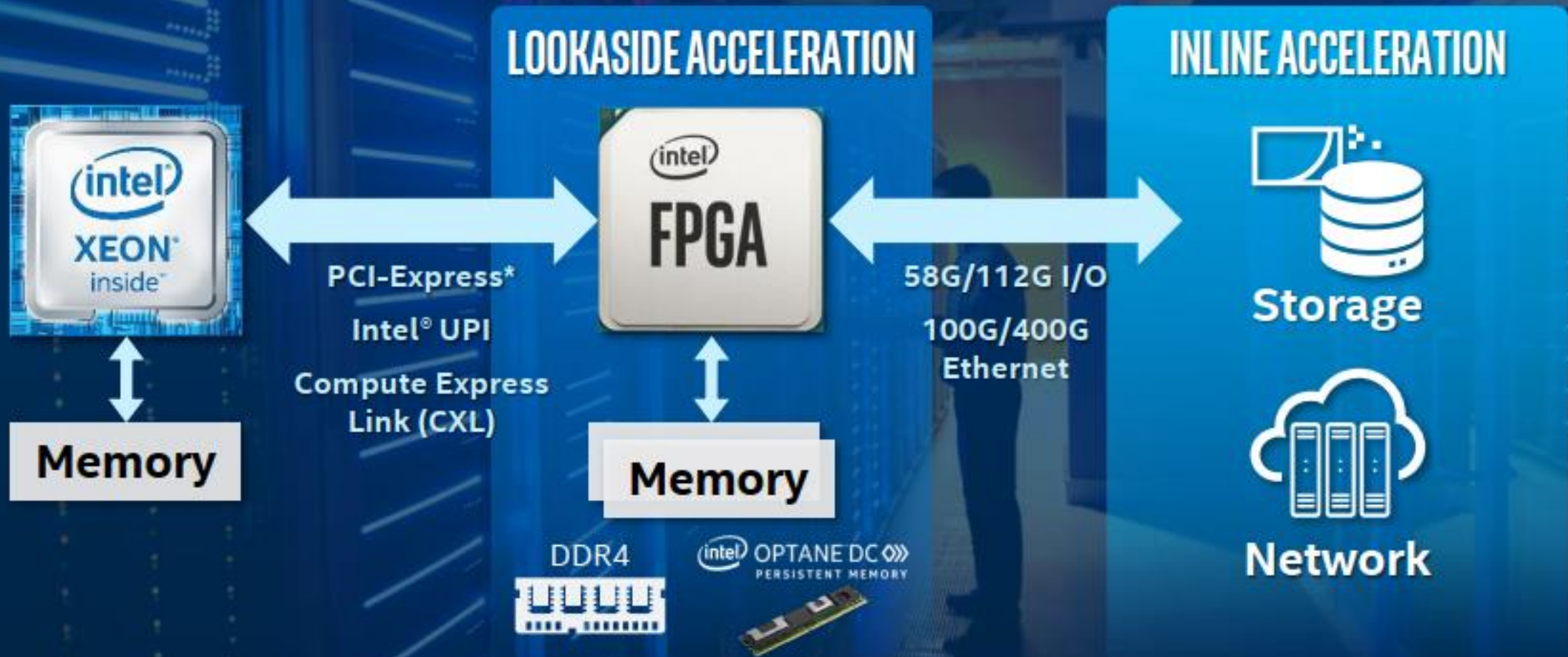
<https://github.com/intel/FPGA-Devcloud/tree/master/main/HandsFree>

Graduate Level and Research: Intel Devclouds and HARP



intel[®]

INTEL® FPGAS ACCELERATING THE CLOUD & ENTERPRISE



Intel FPGA University Graduate Level Coursework

- Graduate level FPGA coursework focuses on AI, Machine Learning, Heterogeneous Computing using C++ extension languages: HLS, OpenCL, DPC++ and OpenVino for Visual Inferencing
- Run a web search on Intel FPGA Training for various topics
- Partner university links for FPGA devcloud coursework through Mindshare Grants:
 - U of Florida – [RTL AFU](#)
 - U Mass Lowell – [OpenCL](#) / [OpenAPI](#)
 - UC Davis - [OneAPI](#)
- Intel offers **free** FPGA HARP (aka vlab) for Research or Devcloud (Teaching) cloud services



Intel FPGA Academic Clouds

Cloud access to Intel servers with FPGAs for academics

FPGAs/SW tools already installed. Just login remotely. Ready to use!

HARP (vlab) for long-term research

Hardware accelerator research program (HARP), originally offered cloud access to integrated (MCP) Xeon+FPGA

Now expanded to offer servers with FPGAs cards, hosted in Intel's Academic Compute Env.

Exclusively for long-term academic research (e.g., 1+ year PhD research)

DevClouds for teaching and beyond

Offers servers with FPGA cards

Suitable for teaching (e.g., lab projects) and short-term research efforts. Move to HARP when research grows

And for short-term development projects in general (academic and industry)

What's available

FPGA flows/framework	Devcloud	HARP
Traditional RTL flow	Y	Y
HLS Compiler	Y	Y
FPGA SDK for OpenCL	Y	Y
DPC++ (part of OneAPI)	Y	Upon Request
OpenVino (AI framework)	Y	Upon Request

FPGA hardware

- Intel Xeon with Arria 10 Programmable Acceleration Cards (PAC)
- Intel Xeon with Stratix 10 Programmable Acceleration Cards (PAC)
- Integrated Xeon+FPGA systems (HARP only)

HARP: Getting access

To get access

Send email to: IL_Academic_Res_Env@intel.com

Put email subject “[HARP] new account request”.

Include a short (1 page max) research proposal

Include the type of workloads you are planning to run

More details

The following website offers information of available FPGA systems in HARP. It also provides detailed tutorials and examples on how to get started.

<https://wiki.intel-research.net/FPGA.html>

Intel Devclouds –Teaching and Short-term Research Focus

FPGA Instance:

RTL
OpenCL
CPU, FPGA

OneAPI Instance:

OneAPI (DPC++)
CPU, GPU, FPGA

Edge Instance:

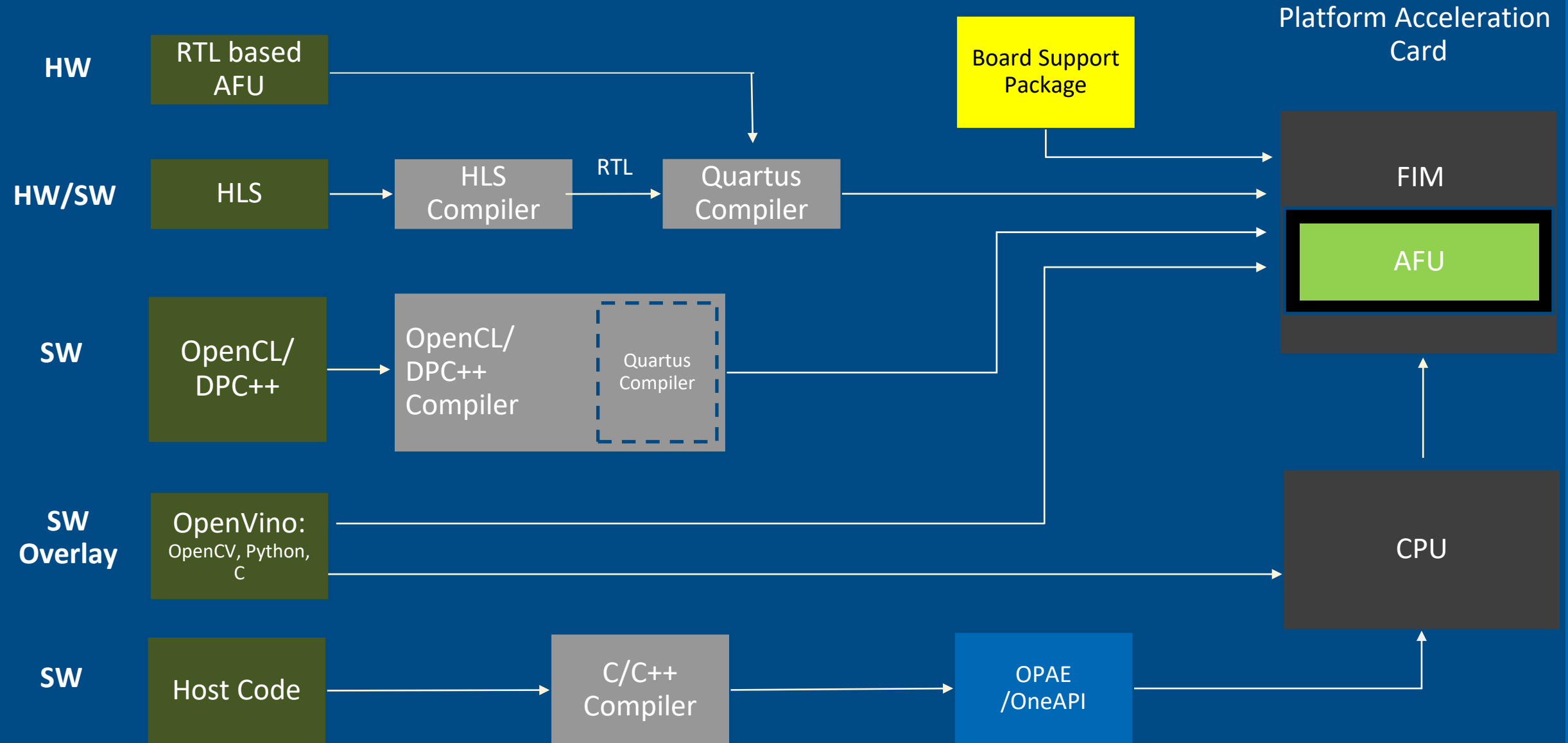
OpenVino
CPU, GPU, VPU, FPGA



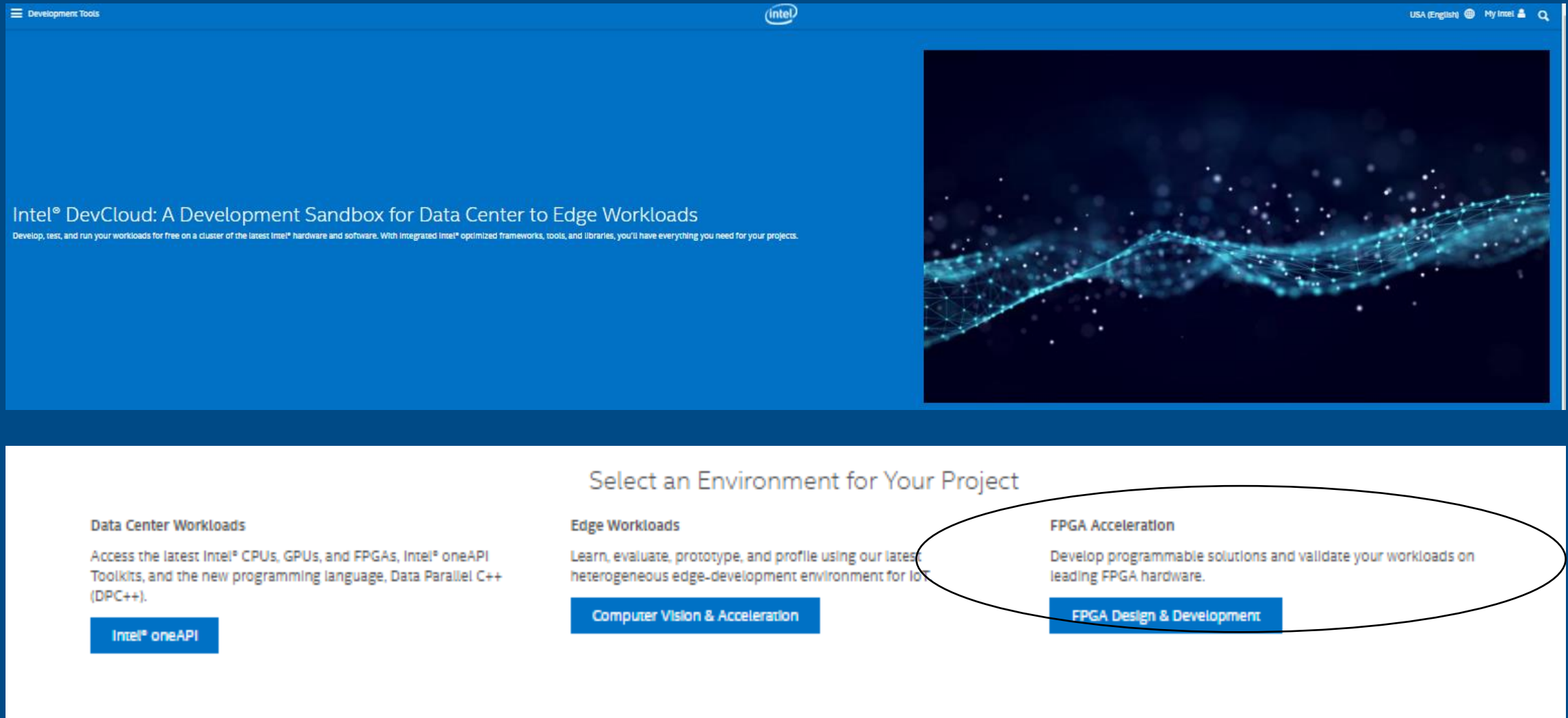
Shared Login

*Free Access!

Working in the FPGA Clouds: Development Flows



Website Access for the Intel Devclouds



The screenshot shows the Intel DevCloud website. The header includes 'Development Tools', the Intel logo, and 'USA (English) My Intel'. The main content area has a blue background with the text 'Intel® DevCloud: A Development Sandbox for Data Center to Edge Workloads' and a subtext 'Develop, test, and run your workloads for free on a cluster of the latest Intel® hardware and software. With integrated Intel® optimized frameworks, tools, and libraries, you'll have everything you need for your projects.' To the right is a graphic of a glowing blue wave. Below this is a white section titled 'Select an Environment for Your Project' with three columns:

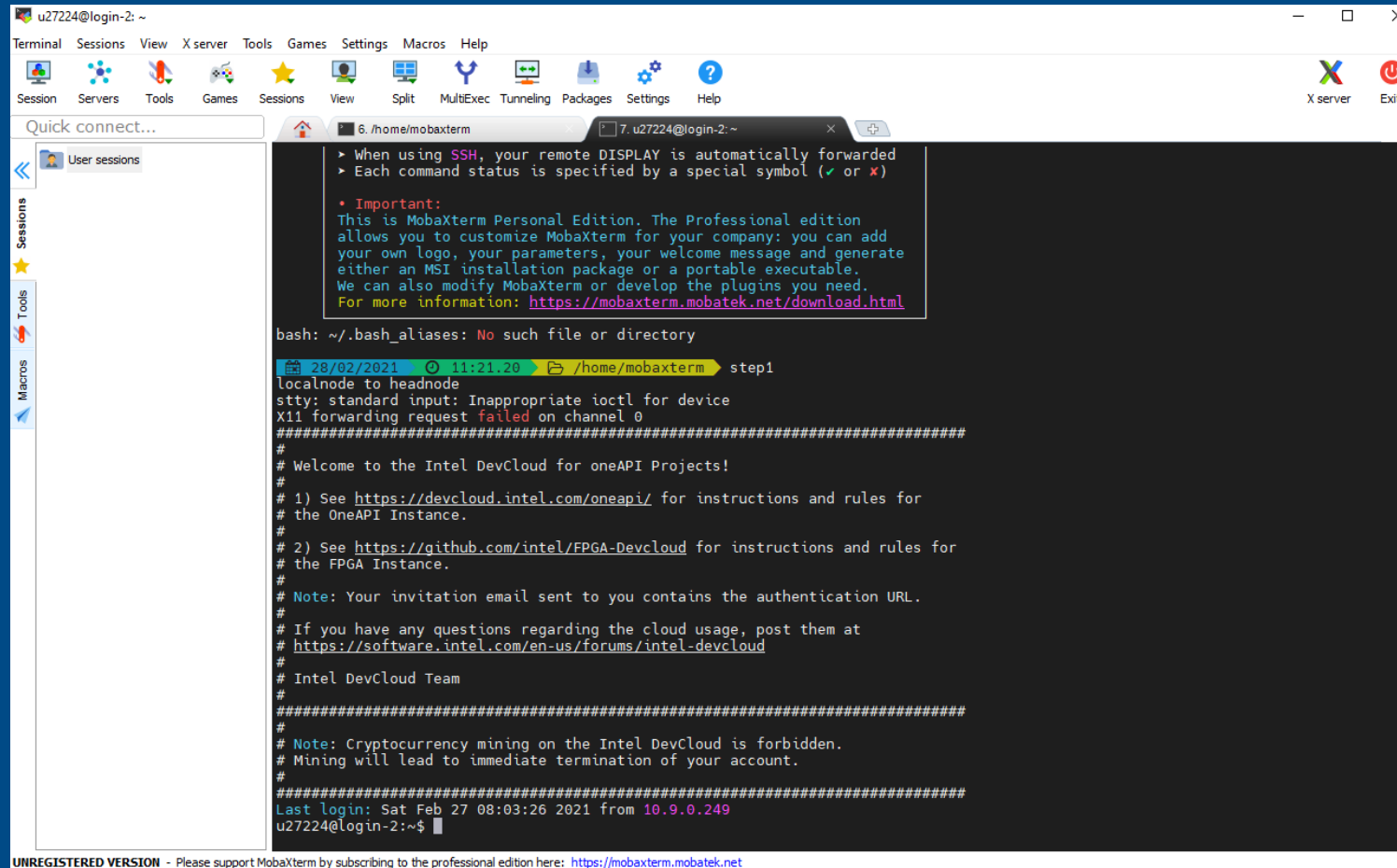
- Data Center Workloads**
Access the latest Intel® CPUs, GPUs, and FPGAs, Intel® oneAPI Toolkits, and the new programming language, Data Parallel C++ (DPC++).
[Intel® oneAPI](#)
- Edge Workloads**
Learn, evaluate, prototype, and profile using our latest heterogeneous edge-development environment for IoT.
[Computer Vision & Acceleration](#)
- FPGA Acceleration**
Develop programmable solutions and validate your workloads on leading FPGA hardware.
[FPGA Design & Development](#)

The 'FPGA Acceleration' column is circled in the image.

Gaining access to the FPGA Devcloud

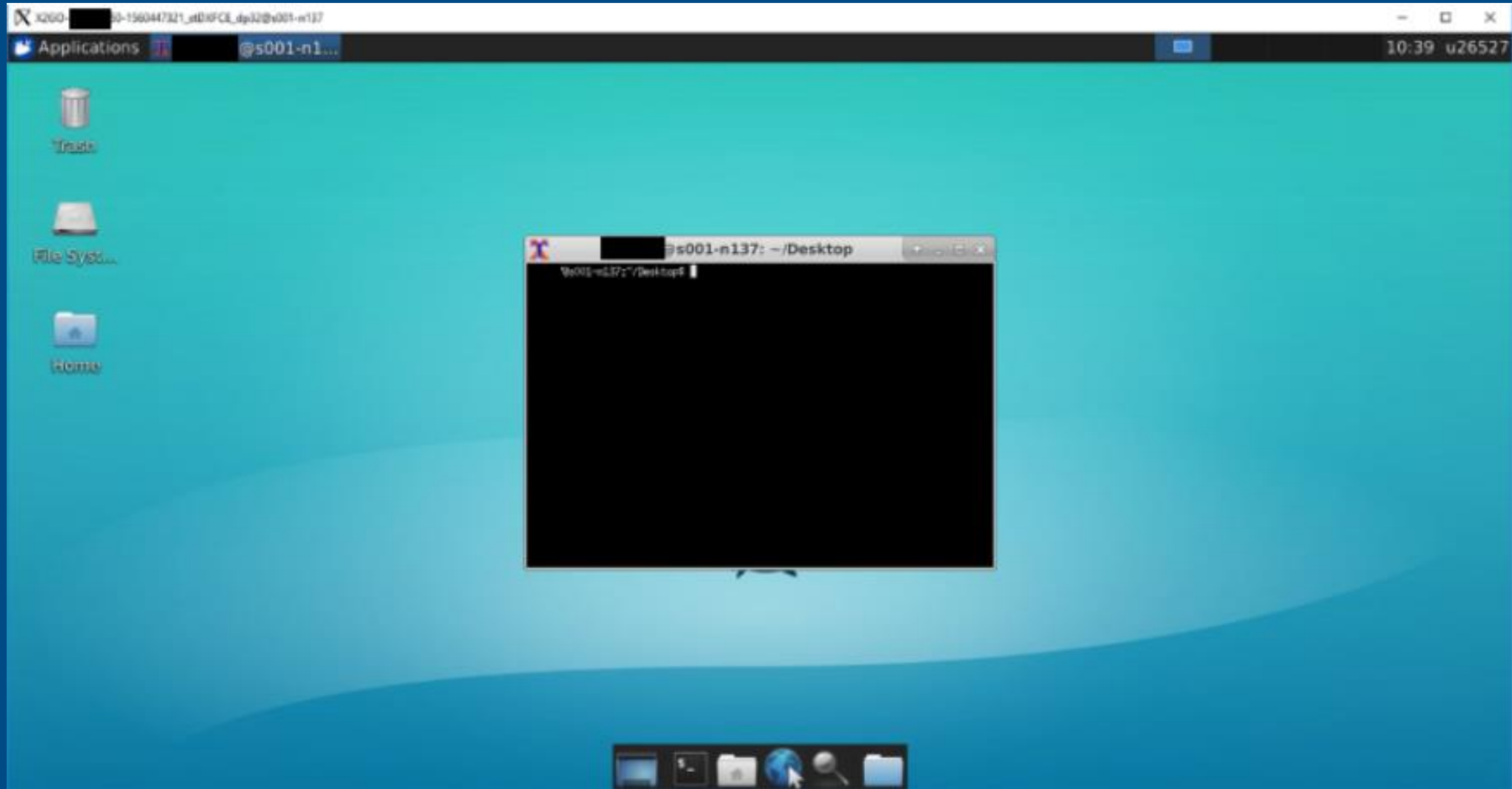
- Signup @ <https://intelsoftwaresites.secure.force.com/fpgadevcloud>
- Check out our github site for instructional material: <https://github.com/intel/FPGA-Devcloud>
- Quickstart guides for a variety of flows: <https://github.com/intel/FPGA-Devcloud/tree/master/main/QuickStartGuides>
- You get access for 3 months, look for cancellation notices and extend access. Data is lost if you fail to extend access
- Use batch mode for longer compiles (timeout after 6 hours)! Use walltime extends beyond 6 hour FPGA compiles

Access method 1: MobaXterm (multi-tab console)



Linux based – makes PC look like Linux filesystem
Doesn't support GUI programs – use X2Go

Access method 2: x2go



Multi window system – using sparingly - for Quartus GUI

Access method 3: JupyterLab

The screenshot displays the JupyterLab web interface in a browser. The top navigation bar includes links to various applications like UP Sharepoint, Corporate, University, Devcloud, RefDes, Training, UNIX, Medical, Travel, Mgmt, Accel, Investing, Conferences, DX, HR, SCU, eASIC, and Maker. The main workspace is divided into three panes:

- File Browser (Left):** Shows a directory tree with files like `log_11102019.txt`, `machine_properties`, `mgls_v9-22_3-1-0.aol.tar.gz`, `MPI_with_OpenMP_or_DPCPP.tar.gz`, `nodecheck.txt`, `nodes.txt`, `oneAPI_Essentials.tar.gz`, `OpenMP_Offload.tar.gz`, `opt_lic`, `OSPRay_Essentials_Notebooks.tar.gz`, `qsubtest.txt`, `quartus_pro_setup.sh`, `quartus_setup_old.sh`, `quartus_setup_orig.sh`, `quartus_setup.sh`, `questa_sim-2020.2_2-online.bin`, `README_201.html`, `README.html`, `runquartus_sh.sh`, `serv_req_info.txt`, `setup_permissions.sh`, `Site_107705.txt`, `status`, `STDIN.e271726`, `STDIN.o271726`, `testnote`, `testqsub.txt`, `tools_sh.txt`, `transcript`, and `Welcome.ipynb` (selected).
- Terminal (Middle):** Shows a shell prompt with a script being executed. The script includes copyright information for Intel Corporation and instructions for setting up the environment, such as sourcing the `/data/intel_fpga/devcloudLoginToolSetup.sh` and running `tools_setup -t S1005`.
- Code Editor (Right):** Displays a Jupyter Notebook with a cell containing Python code to write a file, echo system information, and run `lscpu` to check CPU details. The code is as follows:

```
[ ]: %%writefile hello-world-example
cd $PBS_O_WORKDIR
echo "Hello world from compute server `hostname`!"
echo "The current directory is ${PWD}."
echo "Compute server's CPU model and number of logical CPUs:"
lscpu | grep 'Model name\\|\\^CPU(s)'
echo "Python available to us:"
which python
python --version
echo "The job can create files, and they will be visible back in the Note
sleep 10
echo "Bye"
# Remember to have an empty line at the end of the file; otherwise the last
```

The bottom status bar indicates "Saving completed" and shows the current position in the file: "Ln 10, Col 54 Spaces: 4 S10_opend_batch.sh".

Multi tab system through browser – use native editor, not vi; no Quartus GUI

Contacts for Intel FPGA Devcloud help

University Outreach – San Jose, California

Larry Landis
Sr Manager University Outreach
Intel Programmable Solutions Group
lawrence.landis@intel.com
+1 669 544-6956

Terry Barrette:
Academic Program Manager, CTO Office
Diversity and Inclusion Director
terry.barrette@intel.com
+1 669 544-6735

Also: fpgauniversity@intel.com

Lab – Focus is Access Methods, not language instruction

1. Sign-up (if not done yet – make sure to select FPGA instance)
2. MobaXterm download, ssh key, etc
3. `devcloud_login` and `tools_setup` commands for simplified access
4. Run OpenCL example with emulation
5. Run same example in batch mode
6. Jupyter Lab access

Notes for lab:

- Access to compute machines might get queued, if bottleneck, try interactive lab later
- Interactive sessions consume node for 6 hours, we show interactive mode, however we prefer you use batch processing!
- Walltime command can extend access for 24 hours – see github instructions

