

OpenCAPI platform (IBM P9 IC922 with Alphadata 9H3 with XCVU35P)

Documentation

[Alphadata 9H3 with XCVU35P](#)

<https://opencapi.github.io/oc-accel-doc/>

<https://github.com/OpenCAPI/oc-accel>

End User Guide

This is a condensed version of what is provided in <https://github.com/OpenCAPI/oc-accel>. Assuming all tools are already installed, which is the case.

Compiling a bitfile on hal-fpga-x86

ssh into hal-fhga-x86 with -X to enable X forwarding. This also implies your client can receive X forwarding.

```
source /opt/xilinx/Vitis/2019.2/settings64.sh
git clone https://github.com/OpenCAPI/ocse.git
git clone https://github.com/OpenCAPI/oc-accel.git
cd oc-accel
make snap_config ## this uses an opensource Kconfig menu
```

When a SNAP Config GUI shows up, select 3 items:

1. Card Type (OpenCAPI3.0: AlphaData 9H3-35 (incl. VU35P with HBM) (under development))
2. Action Type (HLS HelloWorld 1024b (Action to Host I/F =1024b))
3. (~/ocse) OCSE_ROOT: Path to OpenCAPI Simulation Engine Root

Do not forget to save the configuration into config file and then exit.

Check file snap_env.sh script to make sure it has "export OCSE_ROOT=~/.ocse".

```
make sim ## (or make sim_tmux if no xterm available)
# in the terminal that appears, run
echo "Hello world_1024. This is my first CAPI SNAP experience. It's real fun." > /tmp/t1
snap_helloworld_1024 -i /tmp/t1 -o /tmp/t2
cat /tmp/t2
```

You should see "HELLO WORLD_1024. THIS IS MY FIRST CAPI SNAP EXPERIENCE. IT'S REAL FUN."

This concludes the software simulation test.

To compile bitfile for FPGA

```
cd ~/oc-accel
make image
```

This will take an hour+ and eventually will produce an image in ~/oc-accel/hardware/build/Images/. Copy this image to the corresponding location on hal-fpga server, if the location already exists (see next section).

```
# scp ~/oc-accel/hardware/build/Images/F <userid>@hal-fpga.ncsa.illinois.edu:~/oc-accel/hardware/build/Images/.
```

Executing on hal-fpga

ssh into hal-fpga.

```
git clone https://github.com/OpenCAPI/oc-accel.git
cd oc-accel
make apps
```

At this point, scp compiled image from hal-fpga-x86:

```
mkdir -p hardware/build/Images
scp <userid>@hal-fpga-x86.ncsa.illinois.edu:~/oc-accel/hardware/build/Images/* ~/oc-accel/hardware/build
/Images/.
```

And finally load bitfile and run the example:

```
# check the exact name of the bitfile, e.g.,
sudo oc-flash-script hardware/build/Images/oc_2023_0202_1645_25G_hls_helloworld_1024_noSDRAM_OC-
AD9H335_1_primary.bin hardware/build/Images/oc_2023_0202_1645_25G_hls_helloworld_1024_noSDRAM_OC-
AD9H335_1_secondary.bin

sudo ./actions/hls_helloworld_1024/tests/hw_test.sh
```



Note that sudo is required for working with the card. Need to ask Yan to either add relevant users to sudoers, or to enable execution of specific tools with su privileges.