FINN Setup

Caution: It might take 1 day to setup FINN using VM
(Since VM is slow & need to install Vivado again)

Speaker: Hua-Yang Weng
Date: 2022/4/06

https://finn.readthedocs.io/en/latest/getting_started.html

Overview

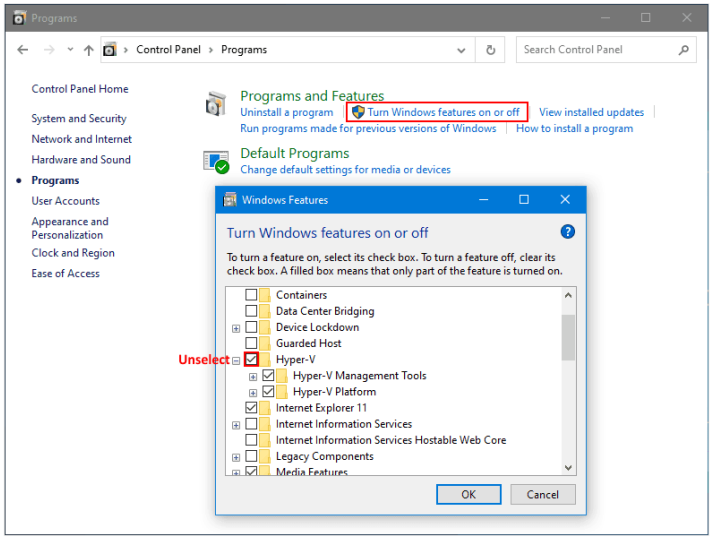
- FINN System Requirements
- VM
- Docker

FINN System Requirements

- **Version 0.7 :**
  - Vivado_hls 2020.1 (Not 2020.2)
  - Vivado 2020.1 (Not 2020.2)
- **OS: (mainly determined by Vivado tools)**
  - Ubuntu 18.04.1 ~ 18.04.4 (Not 18.04.5 or 18.04.6)
  - See below for details https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug973-vivado-release-notes-install-license.pdf
- Note:
  - Currently, FINN github has a branch “vitis_hls” for vitis_hls tool flow, since Vivado_hls will no longer update new version.
  - However, branch “vitis_hls” is not stable. This update is in the FINN v0.8 road map (See: https://github.com/Xilinx/finn/projects/1)

FINN System Requirements

- Skip VM if you have a valid ubuntu version
- VM requirements
  - RAM: > 6G (6G may be okay, but 8G is safer)
  - Space: 80G up (Tested with 100G)
    - 60G -> vivado + vivado_hls
    - 2G -> FINN
- For VM speed-up
  - Disable Hyper-V in windows
  - CPU >= 2 (1 will be very slow)

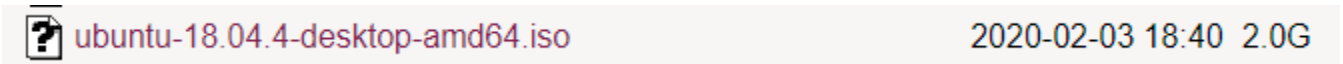


VM: Oracle VM Virtual Box (1/3)

- Download VM: https://www.virtualbox.org/

- Download Ubuntu 18.04.4

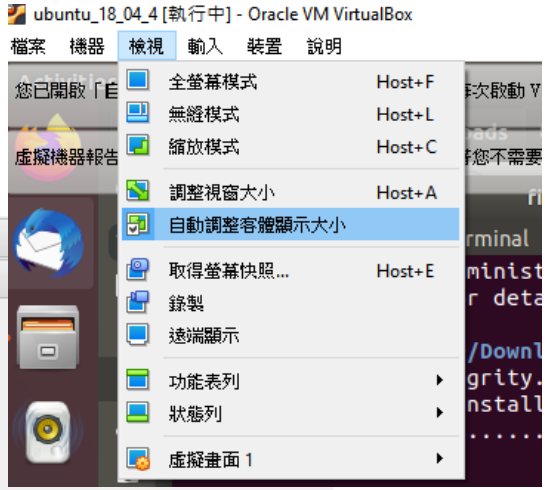
https://old-releases.ubuntu.com/releases/18.04.4/

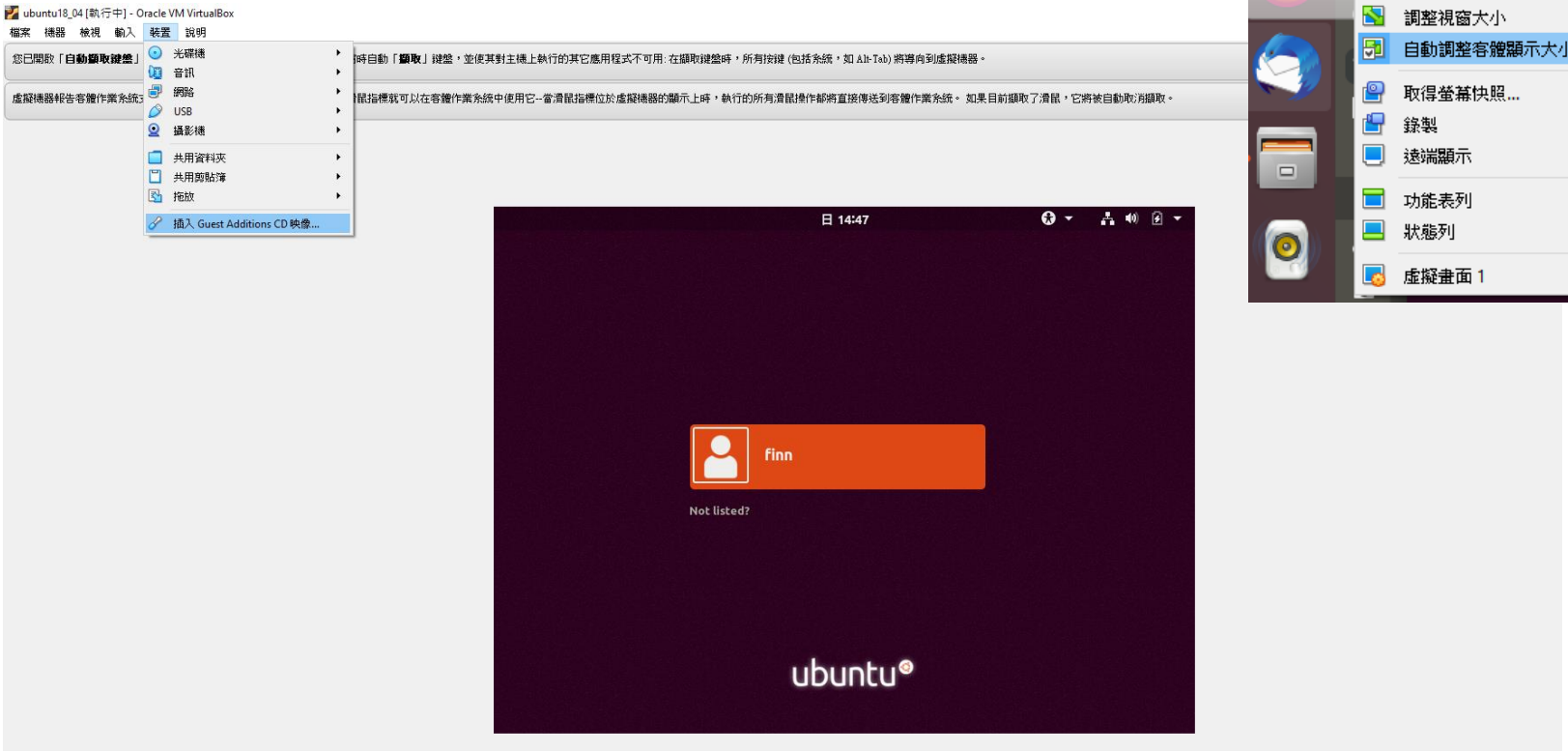


VM: Oracle VM Virtual Box (2/3)

VM: Plugin Guest Additions (3/3)

- For bigger screen resolution (Need to reboot)
- Click auto adjust display

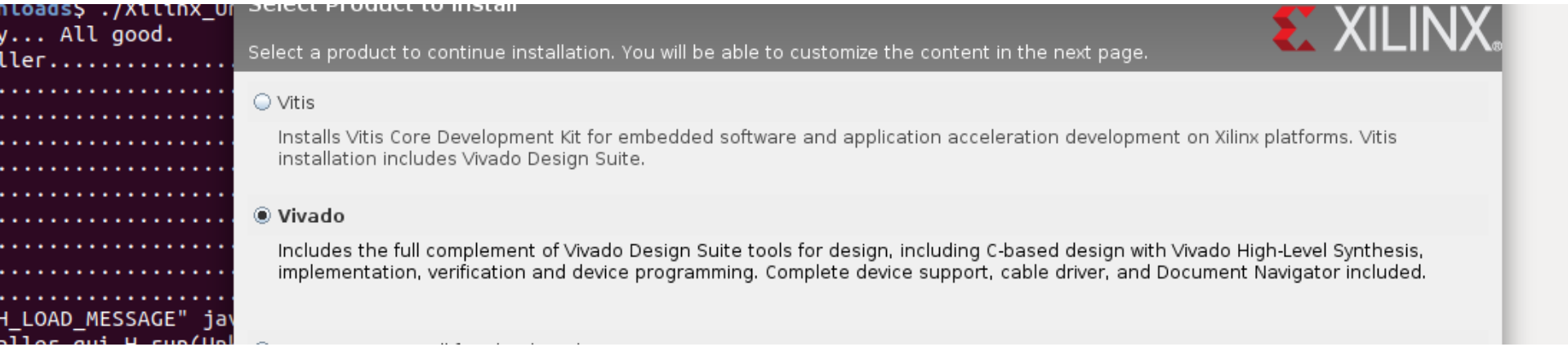




Vivado 2020.1 setup

- Before installing Vivado, it is safer to close both the screen sleep & lock settings.
  - Since screen lock might interrupt the download process
- Install Vivado 2020.1:
  - https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html
- >> sudo path/to/vivado/2020.1/installer

(sudo is for installing at /tools/Xilinx)



FINN steps

- Git clone FINN v0.7
  - >> git clone https://github.com/Xilinx/finn.git
- See here for details:
  - https://finn.readthedocs.io/en/latest/getting_started.html

Modify .bash.rc environment

```
yuoto@yuoto-All-Series:~$ vi ~/.bashrc
```

```
export FINN_XILINX_PATH=/home/yuoto/YuotoSSD/Xilinx/
export FINN_XILINX_VERSION=2020.1
```

Environment variables

Prior to running the *run-docker.sh* script, there are several environment variables you can set to configure certain aspects of FINN. These are summarized below:

- (required) `FINN_XILINX_PATH` points to your Xilinx tools installation on the host (e.g. `/opt/Xilinx`)
- (required) `FINN_XILINX_VERSION` sets the Xilinx tools version to be used (e.g. `2020.1`)
- (required for Alveo) `PLATFORM_REPO_PATHS` points to the Vitis platform files (DSA).
- (required for Alveo) `XRT_DEB_VERSION` specifies the .deb to be installed for XRT inside the container (see default value in `run-docker.sh`).
- (optional) `NUM_DEFAULT_WORKERS` (default 4) specifies the degree of parallelization for the transformations that can be run in parallel, potentially reducing build time
- (optional) `FINN_HOST_BUILD_DIR` specifies which directory on the host will be used as the build directory. Defaults to `/tmp/finn_dev_<username>`
- (optional) `JUPYTER_PORT` (default 8888) changes the port for Jupyter inside Docker
- (optional) `JUPYTER_PASSWD_HASH` (default "") Set the Jupyter notebook password hash. If set to empty string, token authentication will be used (token printed in terminal on launch).
- (optional) `LOCALHOST_URL` (default localhost) sets the base URL for accessing e.g. Netron from inside the container. Useful when running FINN remotely.
- (optional) `NETRON_PORT` (default 8081) changes the port for Netron inside Docker
- (optional) `PYNQ_BOARD` or `ALVEO_BOARD` specifies the type of PYNQ/Alveo board used (see "supported hardware" below) for the test suite
- (optional) `PYNQ_IP` and `PYNQ_PORT` (or `ALVEO_IP` and `ALVEO_PORT`) specify ip address and port number to access the PYNQ board / Alveo target
- (optional) `PYNQ_USERNAME` and `PYNQ_PASSWORD` (or `ALVEO_USERNAME` and `ALVEO_PASSWORD`) specify the PYNQ board / Alveo host access credentials for the test suite. For PYNQ, password is always needed to run as sudo. For Alveo, you can leave the password empty and place your ssh private key in the `finn/ssh_keys` folder to use keypair authentication.

Other variables: See https://finn.readthedocs.io/en/latest/getting_started.html#environment-variables

Execute run-docker.sh

- Before that, install docker first
- Run docker “without” sudo
  - https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user
- Check docker without sudo
  - >> docker run hello-world
- After all the steps are done, open jupyter notebook

>> bash run-docker.sh notebook