



Bridge of Life
Education

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

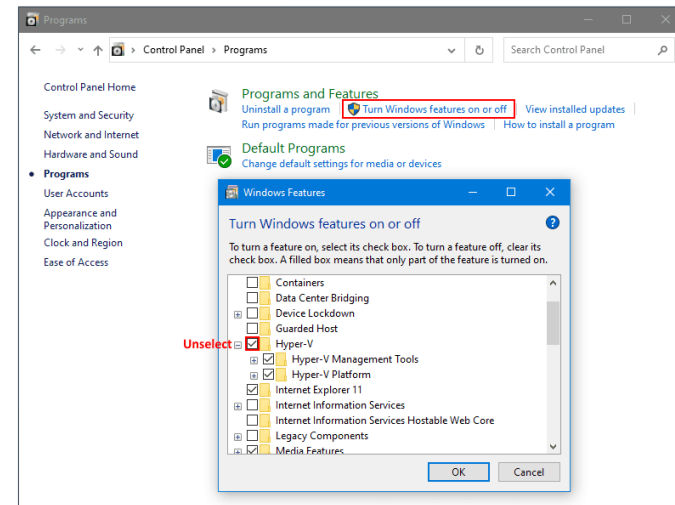
- FINN System Requirements
- VM guide
- Docker

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 OS choice details
https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug973-vivado-release-notes-install-license.pdf
 - Ubuntu 20.04 is okay, however if you install it by yourself, there are some installation issues (See: https://wiki.archlinux.org/title/Xilinx_Vivado#Vivado_2020.1_installer_does_not_start)
- **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: 70G up (Tested with 150G)
 - 50~60G -> vivado + vivado_hls
 - 2G -> FINN
 - If you use method (II) for installing Vivado, then you need 150 G (see P.14)
- For VM speed-up
 - Disable Hyper-V in windows
 - CPU >= 2 (1 will be very slow)
 - Larger video memory



VM: Oracle VM Virtual Box (1/4)

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



The screenshot shows the 'Create Virtual Machine' wizard in Oracle VM VirtualBox. The window title is '建立虛擬機器'. The left sidebar has buttons for '新增(N)', '設定(S)', '捨棄', and '啟動(T)'. The '一般' (General) tab is selected. The '名稱和作業系統' (Name and Operating System) section is active, with instructions: '請為新的虛擬機器選擇描述性名稱和目的地資料夾，並選取要在其上安裝的作業系統類型。您選擇的名稱將在整個 VirtualBox 中使用，以標識這部電腦。' The fields are: '名稱' (Name) set to 'FINN', '機器資料夾' (Machine Folder) set to 'C:\Users\Win10-PC\VirtualBox VMs', '類型(T)' (Type) set to 'Linux', and '版本(V)' (Version) set to 'Ubuntu (64-bit)'. The '記憶體大小' (Memory Size) section is partially visible on the right, with instructions: '選取配置到虛擬機器的記憶體量 (RAM)，單位 MB。建議的記憶體大小為 1024MB。' A slider shows the memory set to 8192 MB, with a range from 4 MB to 8192 MB. The '顯示' (Display) section is also partially visible, with a slider for '視訊記憶體(M)' (Video Memory) set to 128 MB, with a range from 0 MB to 128 MB.

- Download Ubuntu 18.04.4

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

 ubuntu-18.04.4-desktop-amd64.iso

2020-02-03 18:40 2.0G

VM: Oracle VM Virtual Box (2/4)

硬碟檔類型

請選擇新的虛擬硬碟希望使用的檔案類型。 如果不需要用在其它虛擬化軟體，您可以保留這個設定不變更。

- ☒ VDI (VirtualBox 磁碟映像)
- ☐ VHD (虛擬硬碟)
- ☐ VMDK (虛擬機器磁碟)



← 建立虛擬機器

硬碟

如果您希望能加入虛擬硬碟到新的機器。 可以建立新的硬碟檔或從清單選取一個或使用資料夾圖示選取另一個位置。

如果需要更多複雜存放裝置設定，可以略過此步驟，並在機器建立時進行變更機器設定。

建議硬碟的大小為 **10.00 GB**。

- ☐ 不加入虛擬硬碟(D)
- ☒ 立即建立虛擬硬碟(C)
- ☐ 使用現有虛擬硬碟檔案(U)

ubuntu_vm.vdi (標準, 15.00 GB)

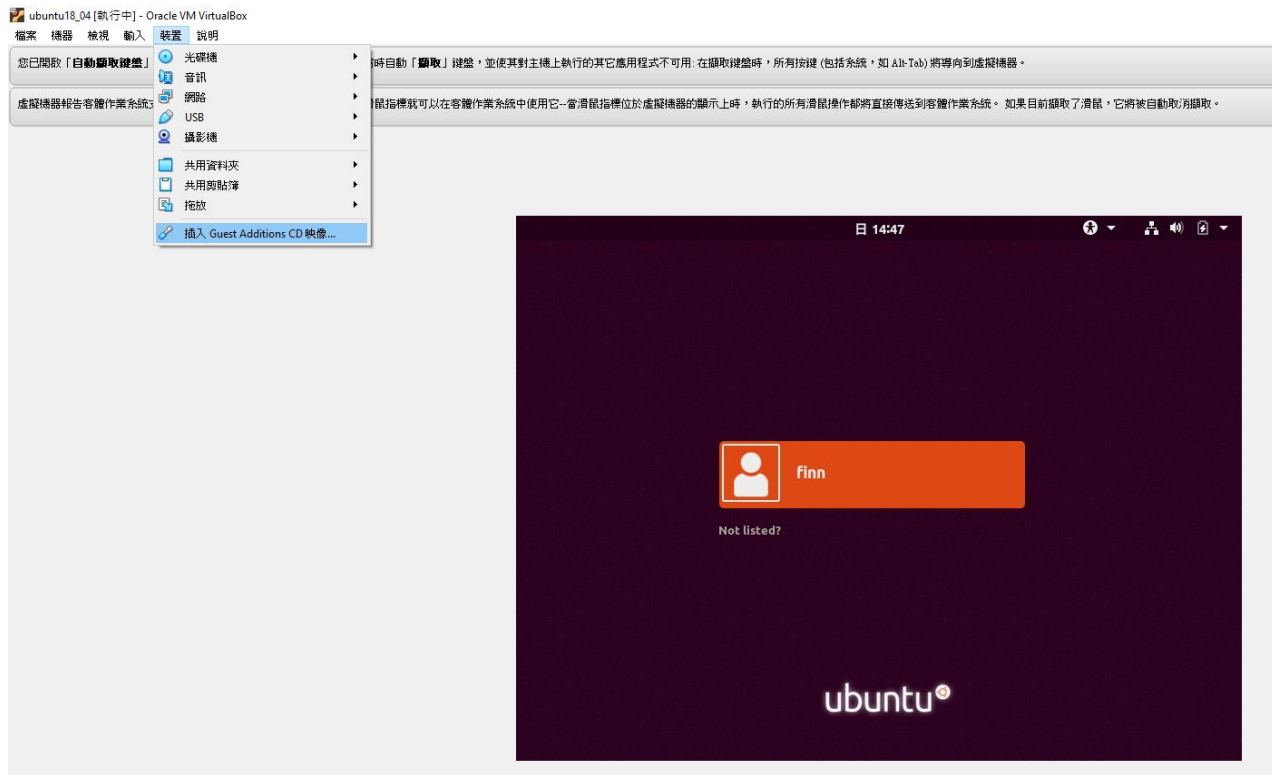
建立

取消



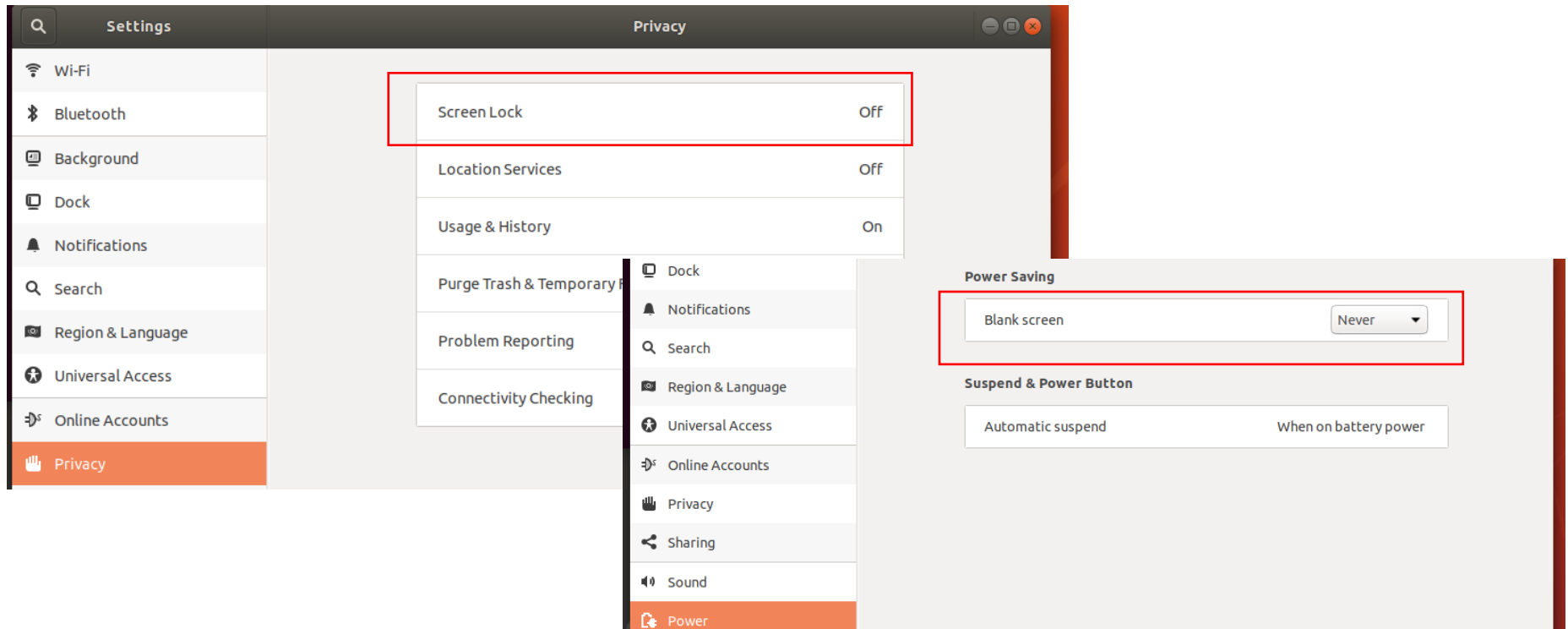
VM: Plugin Guest Additions (3/4)

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



VM: Install Vivado (4/4)

- Before installing Vivado, it is safer to close both the screen sleep & lock settings.



- Since screen lock might interrupt the download process

Two Methods: Installing Vivado

- Some people might fail downloading large file in VM
- Method (I) : Download & install all in VM
 - If downloading fails many times or failed at specific place, then use method (II)
 - Lesser disk space usage during installation
- Method (II): Download in host, install in VM
 - However, for this method, the VM disk memory size should be larger (tested with 150G)
 - Since download, unzip, install all takes more disk space

Method (I) : Download & install all in VM

- Download Vivado 2020.1 installer:
 - <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>


Vivado Design Suite - HLx Editions - 2020.1 Full Product Installation

Important Information


Vivado® Design Suite 2020.1 is now available:

- Ability to select the full image or selected products as part of Web installer
- Address map enhancements provide Realtime error highlighting and cross probing
- Nested DFX further extends the flexibility of DFX solutions
- Report QoR Suggestions predicts up to 3 custom strategies for Improved performance
- Power Rail based reporting now available

Download Includes	Vivado Design Suite HLx Editions (All Editions)
Download Type	Full Product Installation
Last Updated	Jun 4, 2020
Answers	2020.x - Vivado Known Issues
Documentation	Release Notes

 **Xilinx Unified Installer 2020.1: Linux Self Extracting Web Installer (BIN - 116.89 MB)**

MD5 SUM Value : 1f21c8a5858b947c003f741826b5bce5

Download Verification 

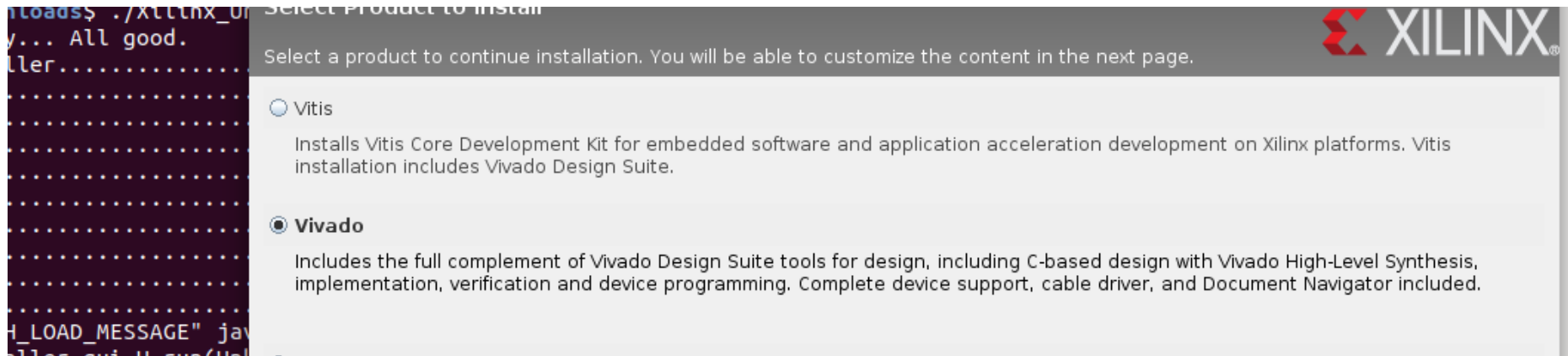
Digests

Signature

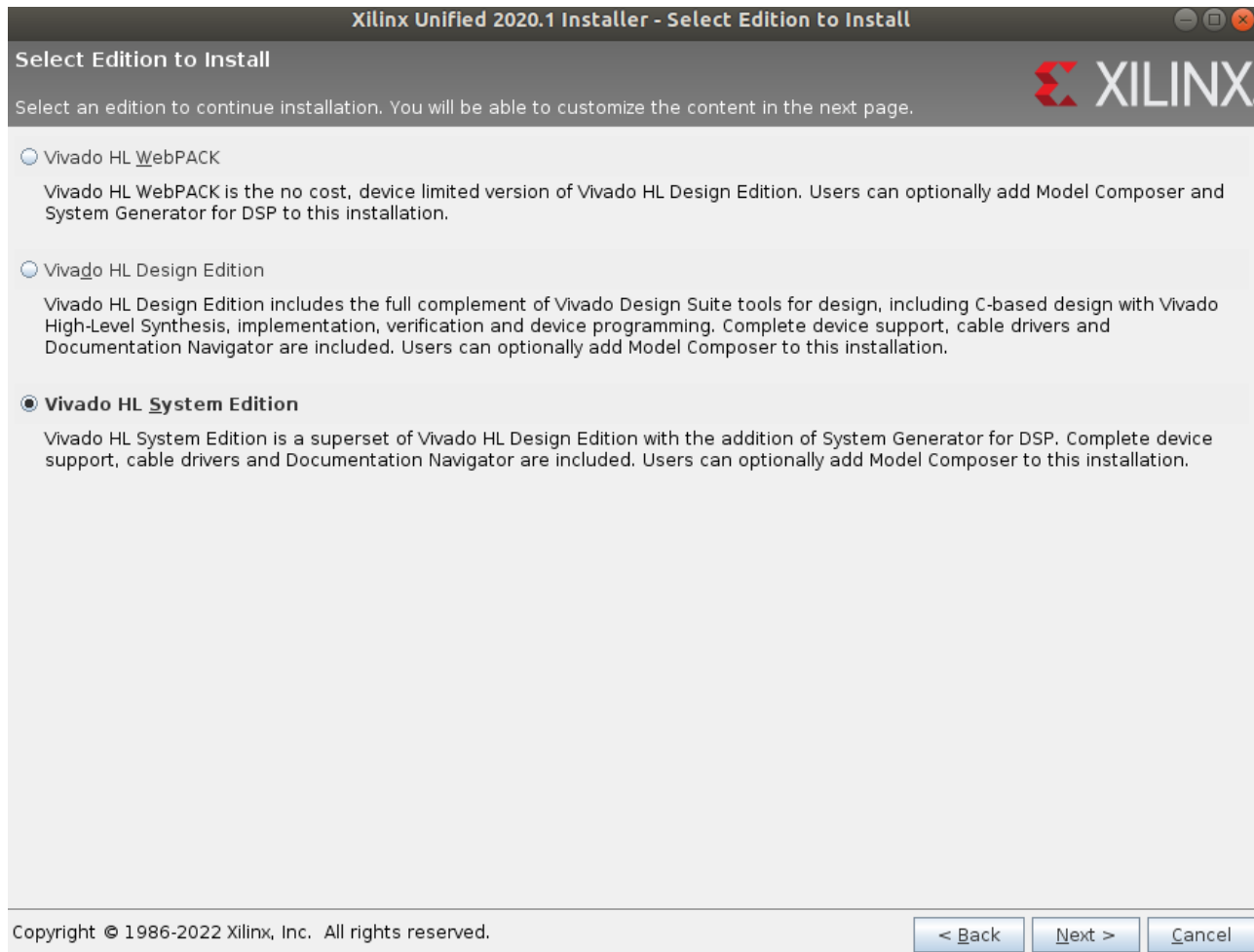
Public Key

Method (I) : Download & install all in VM

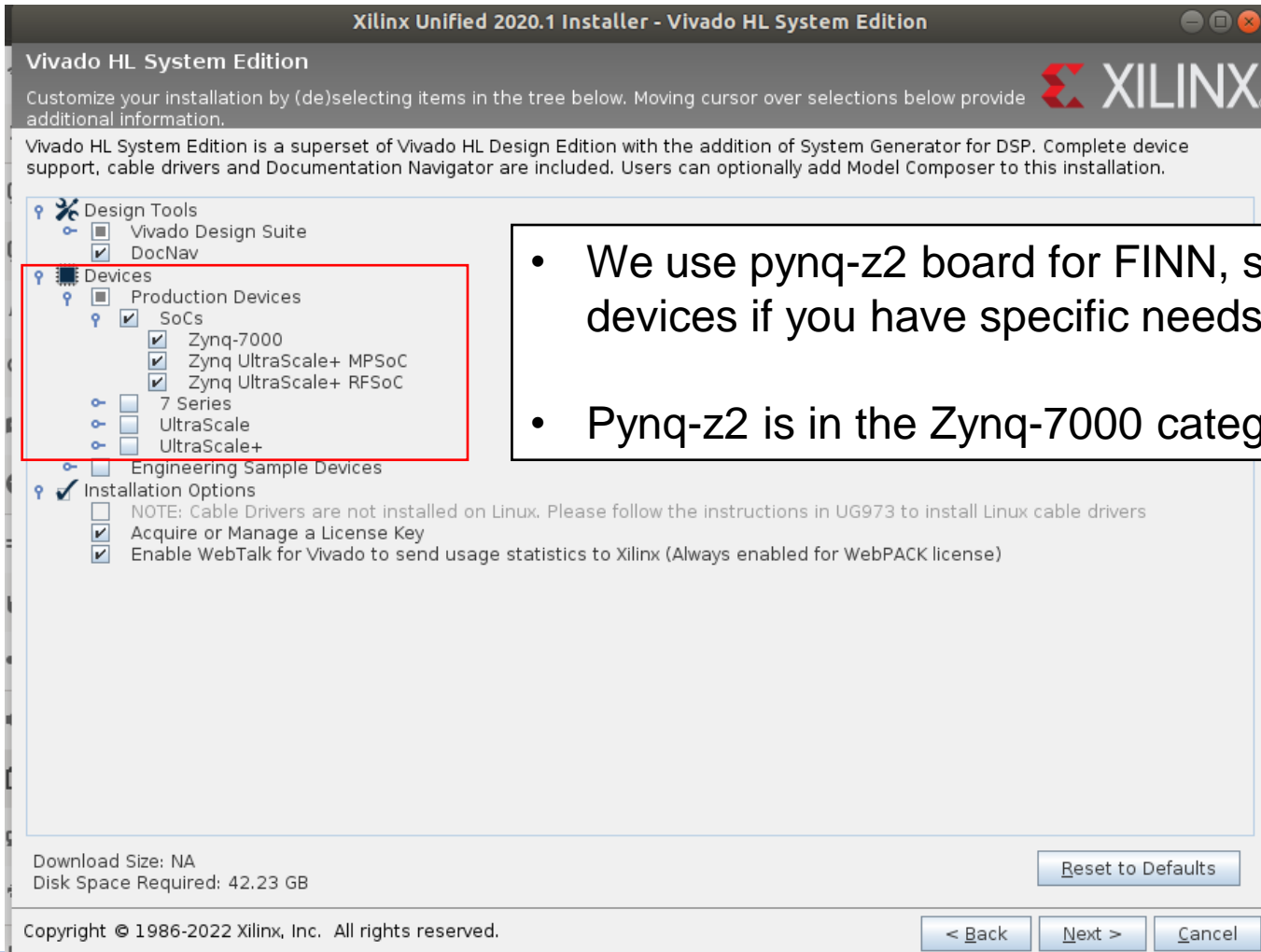
- >> sudo path/to/vivado/2020.1/installer
(sudo is for installing at /tools/Xilinx)



Method (I) : Download & install all in VM



Method (I) : Download & install all in VM



- We use pynq-z2 board for FINN, select more devices if you have specific needs.
- Pynq-z2 is in the Zynq-7000 category

Method (II): Download **in host**, install in VM

- Download Vivado 2020.1 **All OS** installer **@ host**:
 - <https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html>

Vivado Design Suite - HLx Editions - 2020.1 Full Product Installation

Important Information

Vivado® Design Suite 2020.1 is now available:

- Ability to select the full image or selected products as part installer
- Address map enhancements provide Realtime error highlighting cross probing
- Nested DFX further extends the flexibility of DFX solutions
- Report QoR Suggestions predicts up to 3 custom strategies: Improved performance
- Power Rail based reporting now available

Download Includes

Vivado Design Suite HLx Editions (All Editions)

Download Type

Full Product Installation

 **Vivado HLx 2020.1: All OS installer Single-File Download (TAR/GZIP - 35.51 GB)**

MD5 SUM Value : b018f7b331ab0446137756156ff944d9

Download Verification 

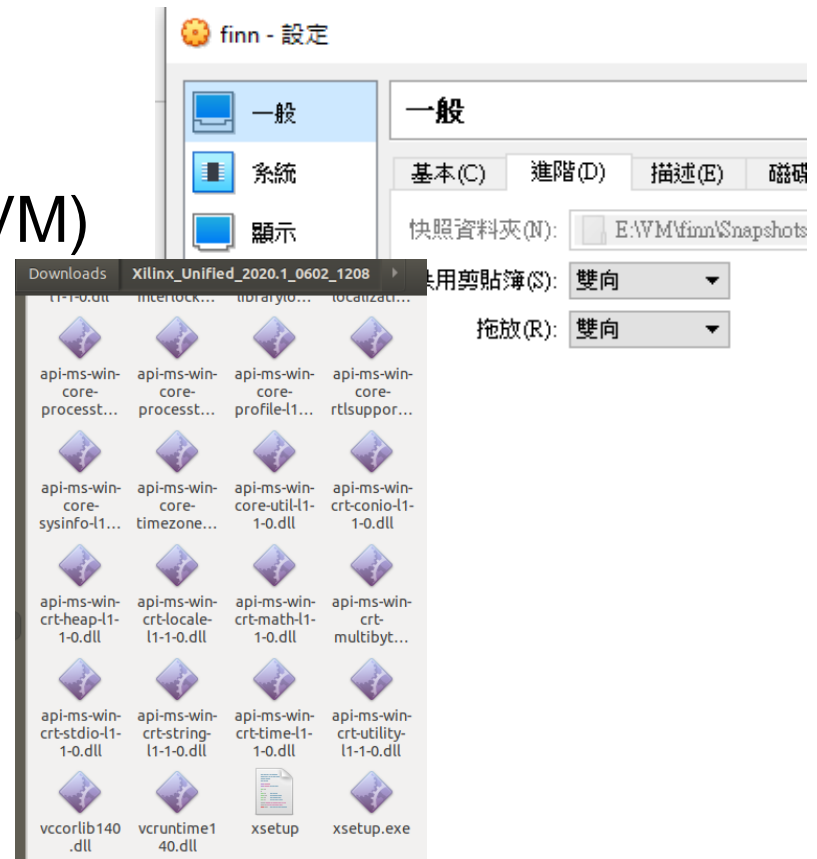
Digests

Signature

Public Key

Method (II): Download in host, install in VM

- Use any method to move the file to VM from host
 - E.g. simple drag-drop
 - (need to open drag-drop in VM)
- Run installation
- >> ./xsetup
 - The installation selection is similar to method(I)



Xilinx Bug patch (Both Methods)

- Due to year changing from 2021->2022, there is a bug in original installation:
 - Details: <https://github.com/Xilinx/finn/discussions/483>
 - Download this Patch:
https://support.xilinx.com/s/article/76960?language=en_US

+ Files (1) Download		
FILE NAME	SIZE	ACTION
+ y2k22_patch-1.2.zip	3.89 KB	▼

- Put the y2k22_patch directory to /tools/Xilinx/

Xilinx Bug patch (Both Methods)

All slides below (including) this are for both methods

- Run the following 2 commands (*or see README in y2k22_patch directory for details*)

```
>> export LD_LIBRARY_PATH = $PWD/Vivado/2020.1/tps/lnx64/python-2.7.16/lib/
```

```
>> Vivado/2020.1/tps/lnx64/python-2.7.16/bin/python2.7  
y2k22_patch/patch.py
```

- You will see something like below if succeed

```
[2022-01-07] INFO: This script (version: 1.2) patches Xilinx Tools for HLS Y2k22 bug for the following release:  
2014.*, 2015.*, 2016.*, 2017.*, 2018.*, 2019.*, 2020.* and 2021.*  
[2022-01-07] UPDATE: /tools/xilinx/Vivado/2021.1/common/scripts  
[2022-01-07] COPY: /tools/xilinx/y2k22_patch/automg_patch_20220104.tcl to  
/tools/xilinx/Vivado/2021.1/common/scripts/automg_patch_20220104.tcl  
[2022-01-07] UPDATE: /tools/xilinx/Vitis_HLS/2021.1/common/scripts  
[2022-01-07] COPY: /tools/xilinx/y2k22_patch/automg_patch_20220104.tcl to  
/tools/xilinx/Vitis_HLS/2021.1/common/scripts/automg_patch_20220104.tcl
```

[optional] Test for vivado & vivado_hls installation

- source /tools/Xilinx/Vivado/2020.1/settings64.sh
- Then you can use
- >> vivado
- or
- >> vivado_hls

anywhere

FINN steps

- For VM, install git using
 - >> sudo apt-get install git
 - If apt-get is locked due to slow VM daily update -> kill the daily update process with command
 - >> kill -9 xxxx (xxx denotes process PID)
- 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
```

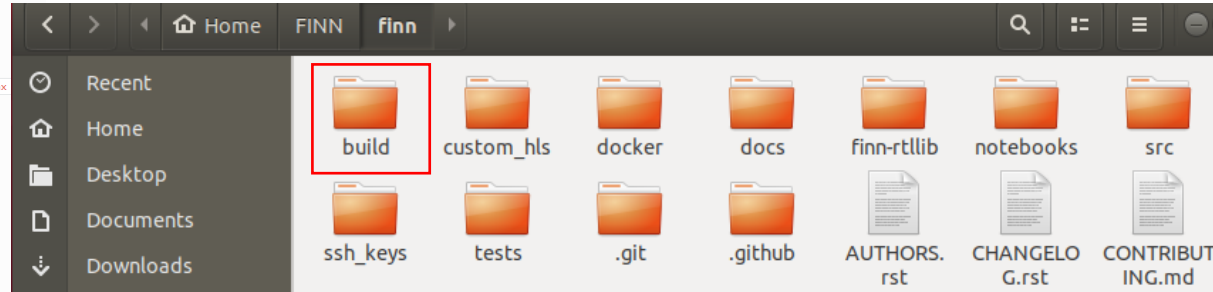
```
export PYNQ_BOARD=Pynq-Z2
export FINN_XILINX_PATH=/tools/Xilinx
export FINN_XILINX_VERSION=2020.1
export FINN_HOST_BUILD_DIR=path/to/finn/build/dir
```

- Add these lines
- For \$FINN_HOST_BUILD_DIR
Make a build directory at finn root directory

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_PASSWORD_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
- <https://docs.docker.com/engine/install/ubuntu/>

Set up the repository

1. Update the `apt` package index and install packages to allow `apt` to use a repository over HTTPS:

```
$ sudo apt-get update

$ sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

2. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

3. Use the following command to set up the **stable** repository. To add the **nightly** or **test** repository, add the word `nightly` or `test` (or both) after the word `stable` in the commands below. [Learn about nightly and test channels.](#)

```
$ echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

Install Docker Engine

1. Update the `apt` package index, and install the *latest version* of Docker Engine and containerd, or go to the next step to install a specific version:

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io
```

Execute run-docker.sh

- Run docker “without” sudo
 - <https://docs.docker.com/engine/install/linux-postinstall/#manage-docker-as-a-non-root-user>
- Check docker without sudo (VM might need reboot)
 - >> docker run hello-world
- After all the steps are done, open **jupyter notebook**
>> bash run-docker.sh notebook

If run-docker.sh Failed

- If you encountered this dependency issue

ImportError: cannot import name 'soft_unicode' from 'markupsafe'

- Modify finn/requirements.txt
- Add markupsafe==2.0.1

```
bitstring==3.1.7
clize==4.1.1
dataclasses-json==0.5.2
docrep==0.2.7
future==0.18.2
gspread==3.6.0
numpy==1.18.0
onnx==1.7.0
onnxoptimizer
onnxruntime==1.4.0
pre-commit==2.6.0
pyscaffold==3.2.1
scipy==1.5.2
setuptools-janitor>=1.1.2
toposort==1.5
vcdvcd==1.0.5
wget==3.2
markupsafe==2.0.1
```

- See here for details
 - <https://github.com/aws/aws-sam-cli/issues/3661>

Finally

- Ctrl + right click the link

```
1ebf730f1158cc0125aea16593cc44
[I 04:37:39.524 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[C 04:37:39.548 NotebookApp]

To access the notebook, open this file in a browser:
file:///tmp/home_dir/.local/share/jupyter/runtime/nbserver-6-open.html
Or copy and paste one of these URLs:
http://finn_dev_finn:8888/?token=6edf7d2db8646235d01ebf730f1158cc0125aea
16593cc44
or http://127.0.0.1:8888/?token=6edf7d2db8646235d01ebf730f1158cc0125aea1659
3cc44
```

- If you can open the notebook, you done the setups for FINN

