Setting Up the ZCU102/104 Evaluation Board

The Vitis™ AI software is made available via docker hub
https://hub.docker.com/r/xilinx/vitis-ai/tags

Vitis AI consists of the following two docker images:

- xilinx/vitis-ai:tools-1.0.0-cpu
- xilinx/vitis-ai:runtime-1.0.0-cpu

Setting Up the Host

- Clone the Vitis AI repository:

On linux terminal : “git clone https://github.com/xilinx/vitis-ai”

- Set up Vitis AI to target Alveo cards (Only for systems with Alveo cards)
- Run the following commands:

cd Vitis-AI/alveo/packages

sudo su ./install.sh

Start the Docker Container.

a. Change directories to Vitis AI:

cd Vitis-AI/

b. Run one of the following command sets.

- For a CPU tools container:

./docker_run.sh xilinx/vitis-ai:1.0.0-cpu

- For a GPU-enabled tools container:

cd Vitis-AI/docker

./docker_build.sh

cd Vitis-AI

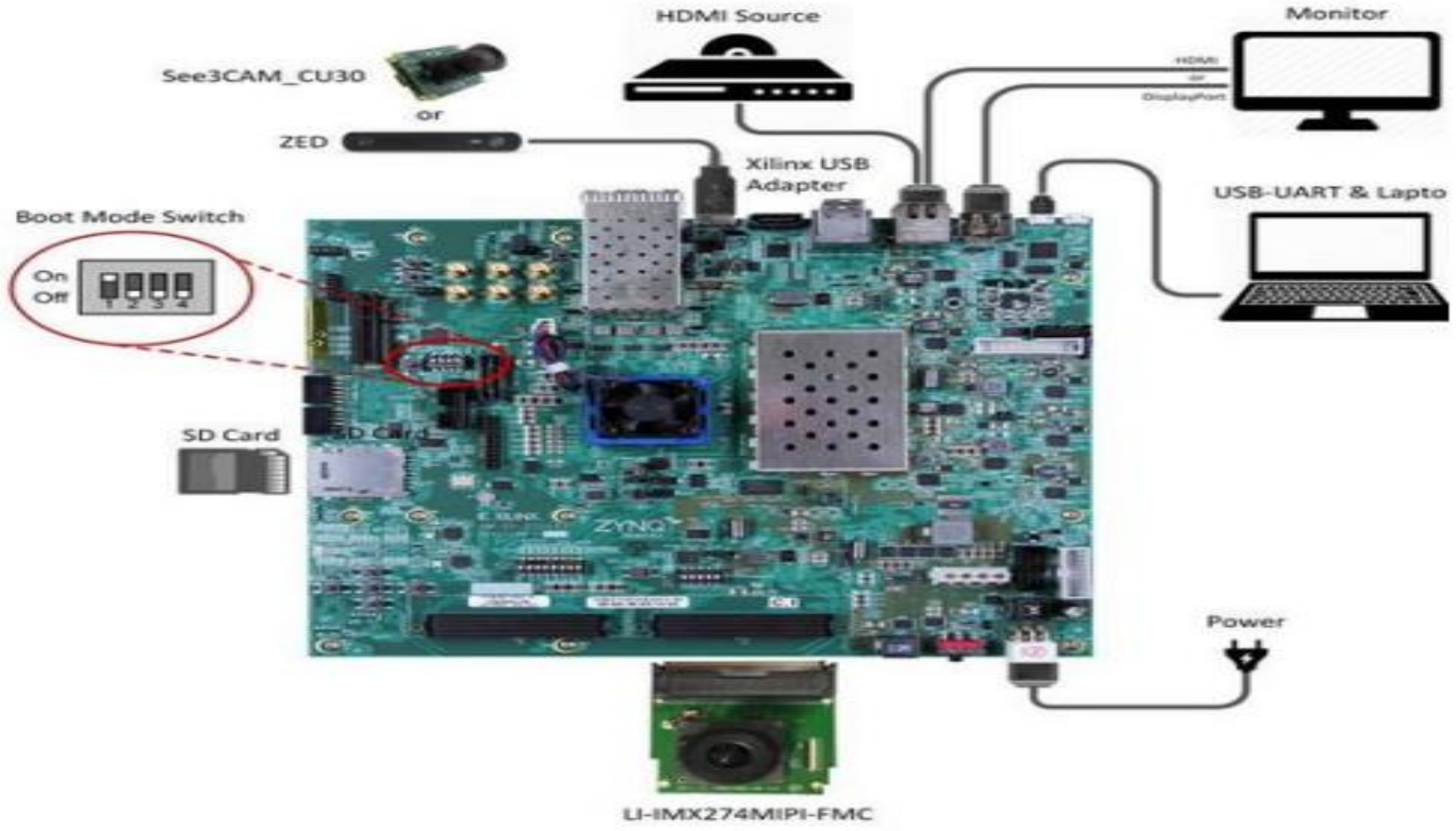
./docker_run.sh xilinx/vitis-ai:runtime-1.0.0-gpu

- For MPSoC Runtime tools container:

./docker_run.sh xilinx/vitis-ai:runtime:1.0.0-cpu

c. Upon starting the container your current working directory will be mounted to: /workspace

Xilinx ZCU102 Evaluation Board and Peripheral Connections



Flashing the OS Image to the SD Card

- For ZCU102, the system images can be downloaded from here:

https://www.xilinx.com/bin/public/openDownload?filename=xilinx-zcu102-dpu-v2019.2.img.gz

The image file has to be flashed into the sd card

i. Download and install Etcher from: https://etcher.io/

ii. Eject any external storage devices such as USB flash drives and backup hard disks. Then, insert the SD card into the slot on your computer, or into the reader.

iii. Run Etcher. Choose the image file -> Choose the SD card ->Click Flash

Booting the Evaluation Board

i. Connect the power supply (12V ~ 5A).
ii. Connect the UART debug interface to the host and other peripherals as required.
iii. Turn on the power and wait for the system to boot.

Accessing the Evaluation Board

There are three ways to access the ZCU102 board: • UART port • Ethernet connection • Standalone

You generally need to connect the board via both UART port and Ethernet connection with the host PC

Download and install Putty or Teraterm in linux using linux Apps search bar

UART Port

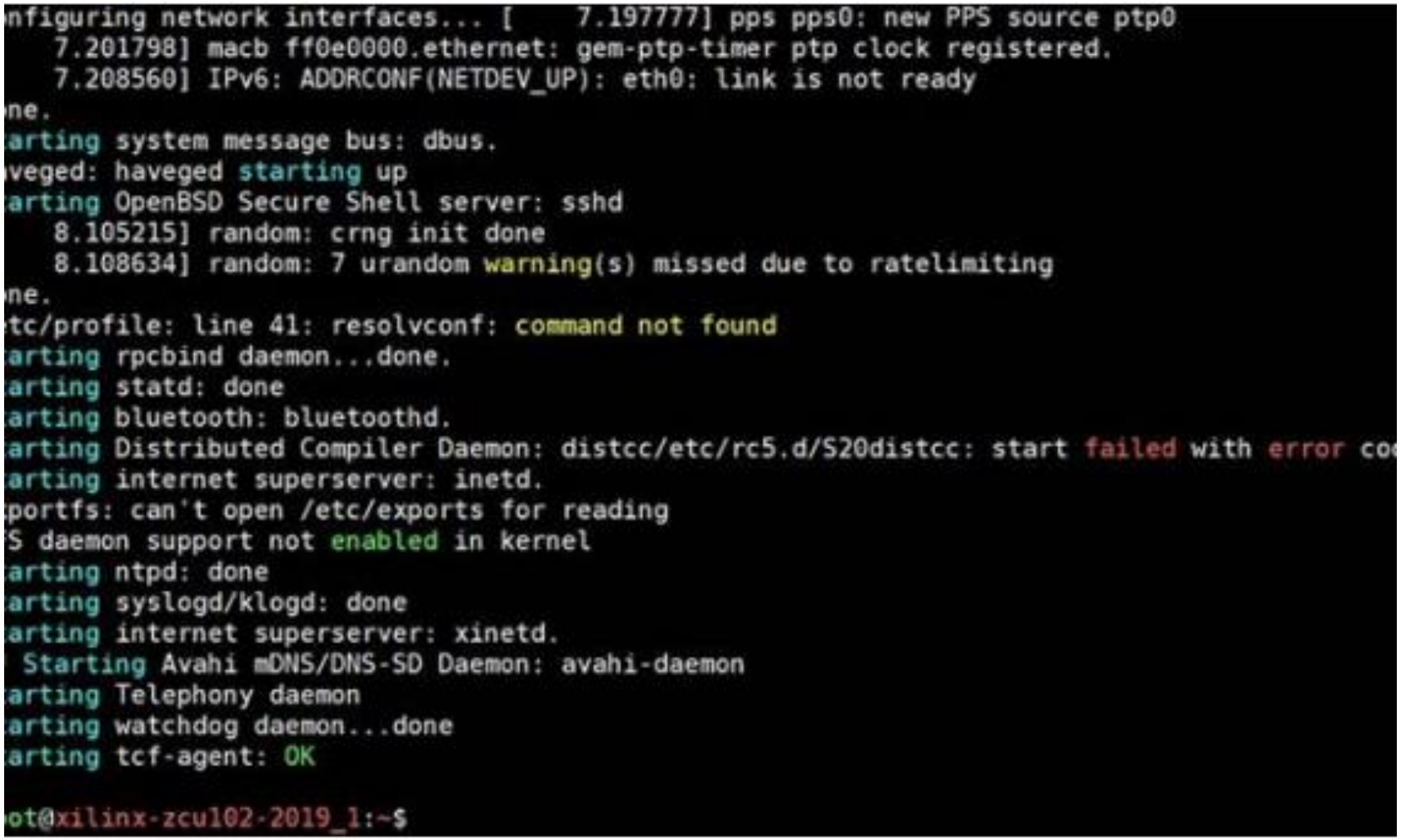
Open Putty by typing “sudo putty” -> select serial port -> Choose the USB port

(The ttyUSB number varies. I suggest you to try each ones. If you cant find ttyUSB in dropdown list box in putty, try google the issue. The linux connection port drivers might need an update. Try typing “lsusb” on linux terminal)

Select • baud rate: 115200 bps • data bit: 8 • stop bit: 1 • no parity

(Make sure you type the right baudrate. Once, the terminal is launched, press enter to get the cursor)

Example of UART Boot on Putty Terminal



Ethernet Port

- You need to assign appropriate ip address in both host pc and the target to establish Ethernet Connection
- On putty terminal type “ipconfig eth0 192.168.0.101”
- On linux terminal type “ipconfig –a” to know the name of the Ethernet port on host pc we are trying to connect. Typically it is eth0 or something like enx00…,
- On linux terminal type “sudo ipconfig eth0*(or enx00… or other name)* 192.168.0.100”
- Verify the Ethernet connection by typing “ping 192.168.0.101” on linux terminal. This will exchange packets between host and target. Press Ctrl+C to stop verifying

Installing Vitis AI Package on the Evaluation Board

- With an Ethernet connection established, you can copy the Vitis AI installation package from docker image vitis-ai-docker-runtime to the evaluation board and set up Vitis AI running environment for the ZCU102 board.
- On linux terminal “sudo scp -r /opt/vitis_ai/xilinx_vai_board_package root@192.168.0.101:~/”

(syntax: scp-r filetobecopied root@IPofboard:~/”

On the ZCU102 board, change to the ~/xilinx_vai_board_package/ directory and run install.sh. The Vitis AI runtime and utility tools will be installed into system automatically. You can now copy Vitis AI samples from docker image vitis-ai-docker-runtime to the evaluation board for evaluation.

Running Examples

- samples can be found at https://github.com/xilinx/vitis-ai
- The /alveo folder contains the sample for DPU-v1 on Alveo platform, and the folder mpsoc contains the samples for edge DPU on ZCU102 and ZCU104 boards
- If you are using Xilinx ZCU102 and ZCU104 boards to run samples, make sure to enable X11 forwarding with the command export DISPLAY=192.168.0.10:0.0 (assuming the IP address of host machine is 192.168.0.10) when logging in to the board using an SSH terminal since all the examples require Linux windows system to work properly.

- Vitis AI samples can be found in the following locations:
- ZCU102 board samples: https://github.com/Xilinx/Vitis-AI/tree/master/mpsoc/vitis_ai_samples_zcu102
- After downloading the samples, copy them into your /workspace/sample/ folder
- Test images can be found in /workspace/sample/vitis_ai_samples_zcu102/images/
- copying the whole directory /workspace/sample/vitis_ai_samples_zcu102/ to ZCU102 board directory /home/root/ is recommend.

- The launching command for each sample is listed in the following table. For Python samples, note that the absolute path for dpuv2_rundir should be specified.

ID	Example Name	Command
1	resnet50	./resnet50 dpuv2_rundir
2	resnet50_mt_py	python3 resnet50.py 3 /home/root/vitis_ai_samples_zcu102/ resnet50_mt_py/dpuv2_rundir/
3	inception_v1_mt_py	python3 inception_v1.py 3 /home/root/vitis_ai_samples_zcu102/inception_v1_mt_py/dpuv2_rundir/
4	pose_detection	./pose_detection video/pose.mp4 dpuv2_rundir
5	video_analysis	./video_analysis video/structure.mp4 dpuv2_rundir
6	adas_detection	./adas_detection video/adas.avi dpuv2_rundir
7	segmentation	./segmentation video/traffic.mp4 dpuv2_rundir

Legacy DNNDK Examples

- The legacy DNNDK C++/Python examples can be found at the following locations:
- ZCU102 examples: https://github.com/Xilinx/Vitis-AI/tree/master/mpsoc/dnndk_samples_zcu102

After downloading the samples, copy them into the /workspace/sample/ folder within the runtime container. These examples can be built with Arm GCC cross-compilation toolchains.

- The samples stay under the directory /workspaces/sample/dnndk_samples_zcu102/. After all the samples are built by Arm GCC cross-compilation toolchains within runtime container, it is recommended to copy the whole directory /workspaces/sample/dnndk_samples_zcu102/ to ZCU102 board directory /home/root/.

Running Examples

ResNet-50

$dnndk_sample_base/resnet50 contains an example of image classification using Caffe ResNet-50 model. It reads the images under the $dnndk_sample_base/dataset/image500_640_480 directory and outputs the classification result for each input image. You can then launch it with the./resnet50 command.

Video Analytics

An object detection example is located under the $dnndk_sample_base/video_analysis directory. It reads image frames from a video file and annotates detected vehicles and pedestrians in real-time. Launch it with the command ./video_analysis video/structure.mp4 (where video/structure.mp4 is the input video file).

ADAS Detection

An example of object detection for ADAS (Advanced Driver Assistance Systems) application using YOLO-v3 network model is located under the directory $dnndk_sample_base/adas_detection. It reads image frames from a video file and annotates in real-time. Launch it
with the ./adas_detection video/adas.avi command (where video/adas.mp4 is the input video file).

Semantic Segmentation

An example of semantic segmentation in the $dnndk_sample_base/segmentation directory. It reads image frames from a video file and annotates in real-time. Launch it with the ./segmentation video/traffic.mp4 command (where video/traffic.mp4 is the input video file).

Running Examples On Cloud

- Working with Vitis AI via cloud is fairly straightforward
- Refer to this youtube tutorial https://www.youtube.com/watch?v=FCP0sN_HInw