TENSORFLOW LITE MODEL BOOTED ON ZEPHYR RTOS RUNNING ON FPGA UPON LITEX VEXC RISC-V SOC

1-INSTALL ZEPHYR:

1.1 Install dependencies:

- sudo apt install git ninja-build gperf ccache dfu-util device-tree-compiler wget python3-dev python3-pip python3-setuptools python3-tk python3-wheel xz-utils file make gcc gcc-multilib g++-multilib libsdl2-dev

- The current minimum required version for the main dependencies are:

Tool	**Min. Version**
CMake	3.20.0
Python	3.6
Devicetree compiler	1.4.6

- Install west, and make sure ~/.local/bin is on your **PATH** environment variable:

```
pip3 install --user -U west
```

```
echo 'export PATH=~/.local/bin:"$PATH"' >> ~/.bashrc
```

```
source ~/.bashrc
```

- Get the Zephyr source code:

```
west init ~/zephyrproject

cd ~/zephyrproject

 west update
```

- Export a Zephyr CMake package. This allows CMake to automatically load boilerplate code required for building Zephyr applications.

```
west zephyr-export
```

- Zephyr’s scripts/requirements.txt file declares additional Python dependencies. Install them with pip3.

```
pip3 install --user -r ~/zephyrproject/zephyr/scripts/requirements.txt
```

1.2 Install Zephyr’s Toolchain:

Download and verify the latest Zephyr SDK bundle:

```
cd ~
wget
    https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.14.1/zephyr-sdk
    -0.14.1_linux-x86_64.tar.gz
wget -O -
    https://github.com/zephyrproject-rtos/sdk-ng/releases/download/v0.14.1/sha256.su
    m | shasum --check --ignore-missing
```

- Extract the Zephyr SDK bundle archive:

```
tar xvf zephyr-sdk-0.14.1_linux-x86_64.tar.gz
```

- It is recommended to extract the Zephyr SDK bundle at one of the following locations:
  1. $HOME
  2. $HOME/.local
  3. $HOME/.local/opt
  4. $HOME/bin
  5. /opt
  6. /usr/local

- The Zephyr SDK bundle archive contains the zephyr-sdk-0.14.1 directory and, when extracted under $HOME, the resulting installation path will be $HOME/zephyr-sdk-0.14.1.
  (preferred path is $HOME/bin)

- Run the Zephyr SDK bundle setup script:

```
cd zephyr-sdk-0.14.1
./setup.sh
```

This may not be an updated installation guide for updated guide (https://docs.zephyrproject.org/latest/develop/getting_started/index.html)

2-GENERATE AND LOAD THE BITSTREAM:

- 2.1 Requirements:
- Riscv Toolchain (refer to https://github.com/merledu/porting-docs/blob/main/QEMU%20Documentation.pdf)

Synthesis tool: like vivado

2.2 Download LiteX:

```
wget https://raw.githubusercontent.com/enjoy-digital/litex/master/litex_setup.py
    chmod +x litex_setup.py
    ./litex_setup.py init
    ./litex_setup.py install
```

2.3 Generating Litex Vexriscv bitstream :

```
cd litex-boards/litex_boards/targets/

./digilent_arty.py --build --csr-json csr.json

./digilent_arty.py --load
```

2.4 FOR BUILDING AN APPLICATION :

Directory tree of an app :

```
App___src/
      |___trj.conf
      |___CMakeLists.txt
```

All the C/C++ files will be in src directory

2.5 To build an Application:

```
cp csr.json <path to app>
```

For example:

```
cp csr.json ~/zephyr/samples/modules/tflite-micro/person-classification-app/

cd ~/zephyr/samples/modules/tflite-micro/person-classification-app/
```

- JSON to DTS Conversion

```
~/<path-to-litex-directory>/litex/litex/tools/litex_json2dts_zephyr.py --dts
    litex_vexriscv.dts --config litex_vexriscv.config csr.json
```

```
west build -p auto -b litex_vexriscv . -DDTC_OVERLAY_FILE=litex_vexriscv.dts
    -DCONFIG_UART_LITEUART=y -DCONFIG_LITEX_TIMER=y
    -DCONFIG_ETH_LITEETH=n -DCONFIG_SPI_LITESPI=n
    -DCONFIG_I2C_LITEX=n

litex_term /dev/ttyUSB1 --kernel build/zephyr/zephyr.bin --serial-boot
```

Now press the reset button on FPGA