On-Boarding Help

Table of contents

List of Figures

FAQs

1. Where is the HW BOM?

   The complete qualified ARC-OTA BOM is located in the Procure the Hardware chapter of the Installation Guide.

2. Does the platform support MU-MIMO?

   ARC-OTA does not offer MU-MIMO integrated interop, however the same platform is capable of adding software features for MU-MIMO.

3. Which frequency bands does the platform support?

   ARC-OTA Cloud offers tested solution in n78 band.

4. How can I apply for an experimental license in United States?

   Review the application located at https://apps2.fcc.gov/ELSExperiments/pages/login.htm. If you have a program experimental license (https://apps.fcc.gov/oetcf/els/index.cfm), you can also use it for the Innovation Zone areas (Boston and the PAWR platforms) by submitting a request on that website.

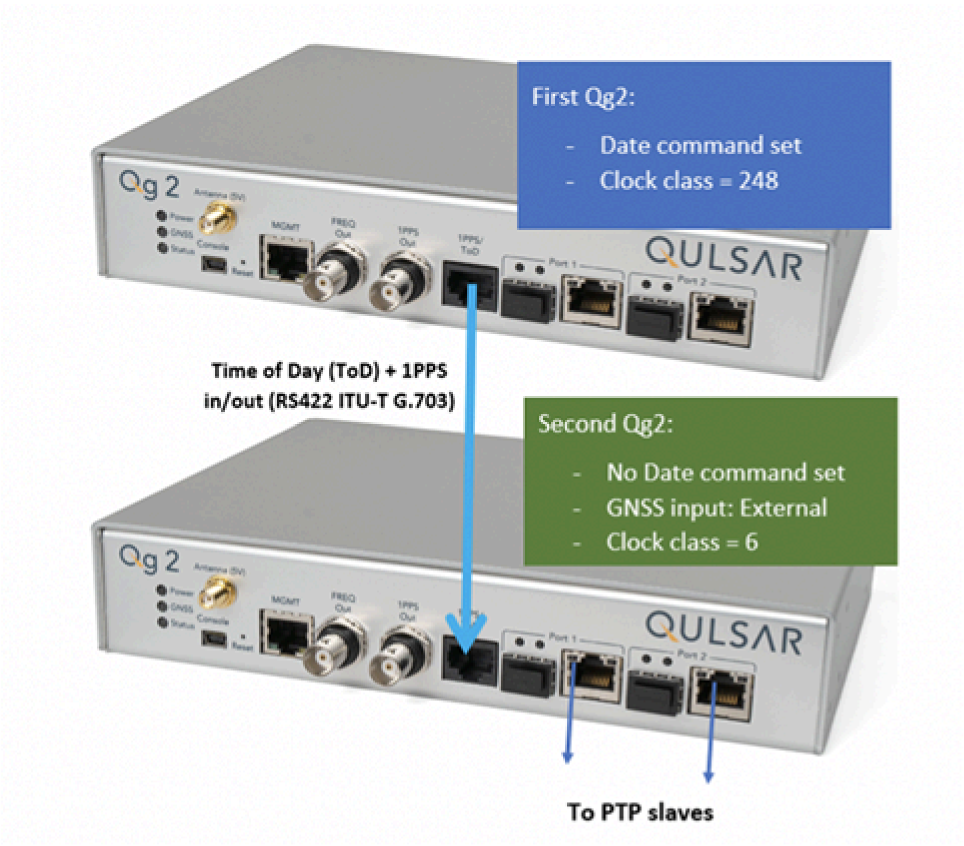
5. Where can I find utility RF tools and calculators?

   See Tools for RF Wireless.

6. Is GPS needed?

   Yes a GPS signal is necessary to drive precision timing for 5G networks. In case GPS signal is in-accessible, the date command can be used as a workaround. This command is useful for those deployments where there is no timing reference (like GNSS) but needs Qg 2 to act as a Grandmaster to propagate time and synchronization over PTP to slave units.

   When using two GMs, you can manually set the date and time on the first one and connect it’s 1PPS/ToD output to the 1PPS/ToD port (configured as input) of the second GM. The second GM then outputs PTP messages with a clock class=6.



First Qg2:
- Date command set
- Clock class = 248

Time of Day (ToD) + 1PPS
in/out (RS422 ITU-T G.703)

Second Qg2:
- No Date command set
- GNSS input: External
- Clock class = 6

To PTP slaves

7. Check OS and kernel version and configuration:

```
cat /proc/cmdline BOOT_IMAGE=/boot/vmlinuz-5.4.0-65-lowlatency
root=UUID=d0bc583b-6922-4e70-af14-92b624fe1bbe ro default_hugepagesz=1G
hugepagesz=1G hugepages=16 tsc=reliable clocksource=tsc intel_idle.max_cstate=0
mce=ignore_ce processor.max_cstate=0 intel_pstate=disable audit=0 idle=poll
isolcpus=2-21 nohz_full=2-21 rcu_nocbs=2-21 rcu_nocb_poll nosoftlockup
iommu=off intel_iommu=off irqaffinity=0-1,22-23 uname -a Linux dev01 5.4.0-65-
lowlatency #73-Ubuntu SMP PREEMPT Mon Jan 18 18:17:38 UTC 2021 x86_64
x86_64 x86_64 GNU/Linux
```

8. For your convenience, you can use the following shell script to build cuBB and create the script in `/opt/nvidia/cuBB`:

```
#!/bin/bash SCRIPT=$(readlink -f $0) SCRIPT_DIR=$(dirname $SCRIPT) echo running
$SCRIPT echo running $SCRIPT_DIR export cuBB_SDK=${SCRIPT_DIR}
insModScript=${SCRIPT_DIR}/cuPHY-CP/external/gdrcopy/ echo $insModScript cd
$insModScript && make && ./insmod.sh && cd - export
```

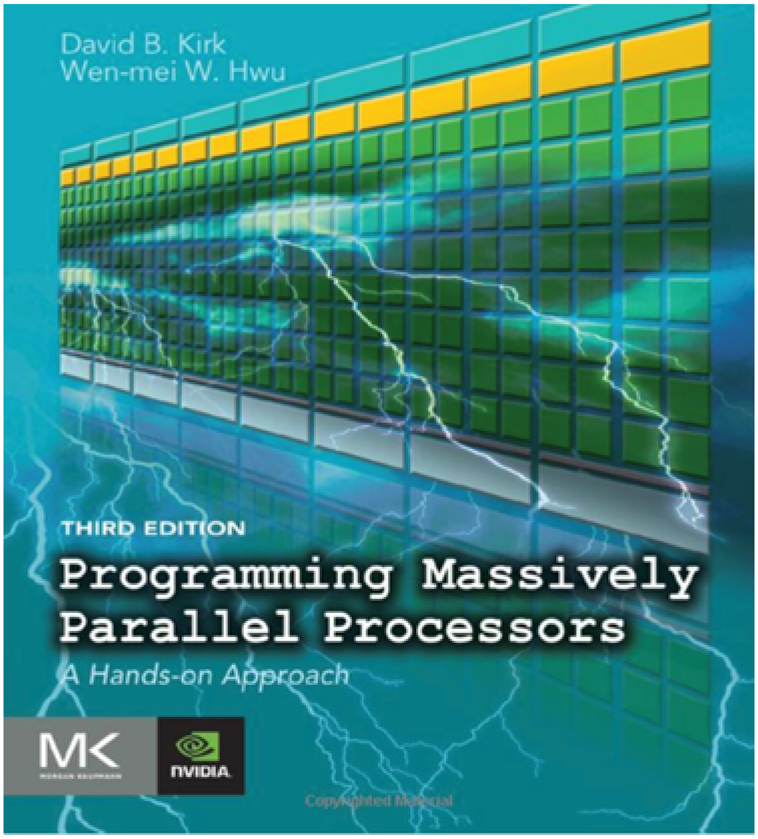
```
LD_LIBRARY_PATH=$cuBB_SDK/gpu-dpdk/build/install/lib/x86_64-linux-gnu echo
$LD_LIBRARY_PATH | sudo tee /etc/ld.so.conf.d/aerial-dpdk.conf sudo ldconfig echo
"perhaps you want to: " echo "mkdir build && cd build && cmake .. " mkdir -p build
&& cd $_ && cmake .. && time chrt -r 1 taskset -c 2-20 make -j
```

9. For your convenience, shell script to start cuphycontroller, create the script in `/opt/nvidia/cuBB`:

```
#!/bin/bash sudo nvidia-smi -pm 1 sudo nvidia-smi -i 0 -lgc $(sudo nvidia-smi -i 0 --query-supported-clocks=graphics --format=csv,noheader,nounits | sort -h | tail -n 1)
SCRIPT=$(readlink -f $0) SCRIPT_DIR=$(dirname $SCRIPT) echo running $SCRIPT
echo running $SCRIPT_DIR export cuBB_SDK=${SCRIPT_DIR}
insModScript=${SCRIPT_DIR}/cuPHY-CP/external/gdrcopy/ echo $insModScript cd
$insModScript && ./insmod.sh && cd - export LD_LIBRARY_PATH=$cuBB_SDK/gpu-dpdk/build/install/lib/x86_64-linux-gnu:$cuBB_SDK/build/cuPHY-CP/cuphydriver/src
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/mellanox/gdrcopy/src export
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/opt/mellanox/dpdk/lib/x86_64-linux-gnu:/opt/mellanox/doca/lib/x86_64-linux-gnu echo $LD_LIBRARY_PATH | sed
's/:/\n/g' | sudo tee /etc/ld.so.conf.d/aerial-dpdk.conf sudo ldconfig export
GDRCOPY_PATH_L=$cuBB_SDK/cuPHY-CP/external/gdrcopy/src export
CUDA_MPS_PIPE_DIRECTORY=/var export CUDA_MPS_LOG_DIRECTORY=/var sudo -E
nvidia-cuda-mps-control -d sudo -E echo start_server -uid 0 | sudo -E nvidia-cuda-mps-control sleep 5 echo "perhaps you want to: " echo "sudo -E
LD_LIBRARY_PATH=$cuBB_SDK/gpu-dpdk/build/install/lib/x86_64-linux-gnu/
./build/cuPHY-CP/cuphycontroller/examples/cuphycontroller_scf P5G_SCF_FXN"
sudo -E ./build/cuPHY-CP/cuphycontroller/examples/cuphycontroller_scf
P5G_SCF_FXN sudo ./build/cuPHY-CP/gt_common_libs/nvIPC/tests/pcap/pcap_collect
```

Recommended Reading Material

Programming Massively Parallel Processors by David B. Kirk and Wen-mei W. Hwu



This book can be obtained through`ScienceDirect <https://www.sciencedirect.com/book/9780128119860/programming-massively-parallel-processors>`_ or similar book retailers.

Hands-on CUDA-C

See the syllabus from NVIDIA/UIUC Accelerated Computing Teaching Kit, it outlines each module's organization in the downloaded Teaching Kit. It can be used to teach and learn CUDA.