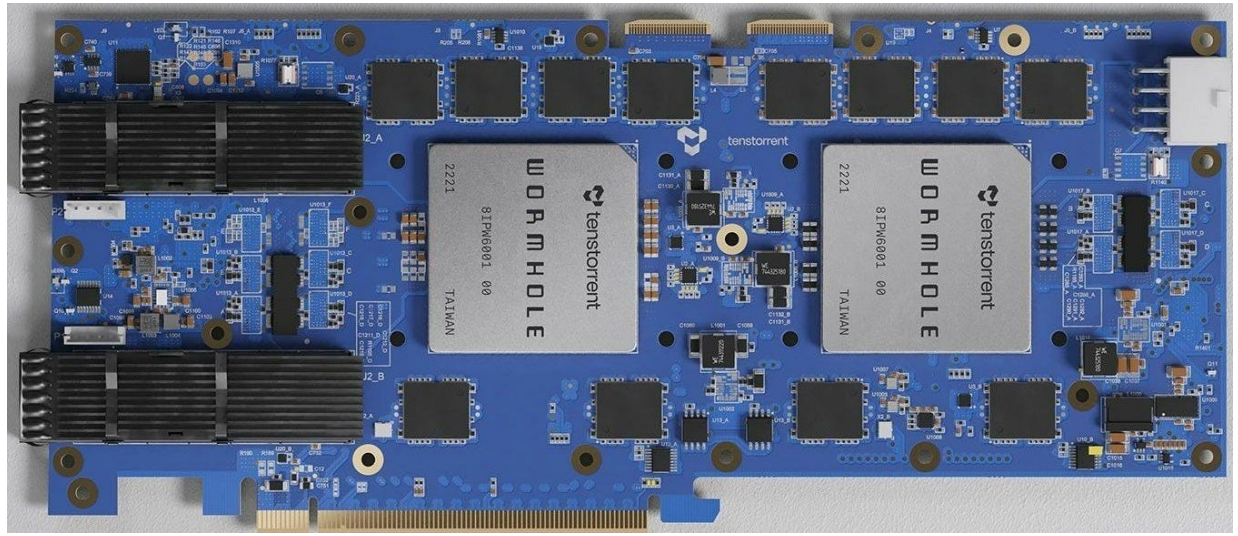
Accessing the Tenstorrent device



epcc

tenstorrent

Using the EPCC RISC-V testbed

- We host a testbed system that enables developers to experiment with RISC-V
  - https://riscv.epcc.ed.ac.uk

- Lots of different RISC-V technologies, but this also contains Wormhole and Blackhole PCIe accelerators
  - This week and next our compute facility is offline due to power maintenance works, so we are currently using a single Wormhole for the tutorial and access instructions are slightly different than they normally are

Everyone has a visitor account on the machine

- We will provide the details of this to you now
  - Everything is set up in this account so that you can work with the practicals

- You can then use Putty (on Windows) or your inbuilt SSH client (on Linux or MacOS) to SSH into the remote machine

Let's access the machine

- Step one – using the visitor account assigned to you, access the login node

```
ssh user-id@riscv-login.epcc.ed.ac.uk
```

  - You are now logged into the login node

- Step two – SSH to tenstorrent Wormhole machine

```
[user-id@riscv-login ~]$ ssh tenstorrent1
```

Let one of the tutors know if you are struggling with this

We have set some files up for you already….

```
[user-id@tenstorrent1 ~]$ ls
tt-metal-0.62.2    tt-tutorial
```

- The *tt-metal-0.62.2* directory contains the Tenstorrent software stack
  - You don't need to worry too much about this, but if you issue *export* you will see that there are some paths (e.g. *TT_METAL_HOME*) set up which point into this

- We will be working with the *tt-tutorial* directory today which contains the practicals

```
[user-id@tenstorrent1 ~]$ cd tt-tutorial
[user-id@tenstorrent1 ~/tt-tutorial]$ ls
Lectures    LICENCE    practical    README
[user-id@tenstorrent1 ~/tt-tutorial]$ cd practical
[user-id@tenstorrent1 ~/tt-tutorial/practical]$ ls
five   four   general   one   README.md   three   two
```

- These correspond to practicals one through five that we will work with today

A really useful tool

- Tenstorrent System Management Interface (TT-SMI) is a command line utility that enables us to interact with Tenstorrent devices connected to a host

```
[user-id@tenstorrent1 ~]$ tt-smi
```

- Brings up an ncurses dialog with information about the card, including telemetry (e.g. power draw, temperature, clock frequency)

- Most usefully this can be used to reset the device if it hangs/gets stuck
  - Please don’t use this during the tutorial (unless it really gets stuck!) as multiple people will be running on the same device, so you might reset their run!

```
[user-id@tenstorrent1 ~]$ tt-smi -r
```

Sharing the Wormhole: Your executable is running

```
[user-id@tenstorrent1 ~]$ ./ex_five
2025-09-01 18:38:54.566 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:54.579 | info     |          Device | Opening user mode device driver (tt_cluster.cpp:192)
2025-09-01 18:38:54.579 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:54.590 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:54.601 | info     |   SiliconDriver | Harvesting mask for chip 0 is 0x300 (NOC0: 0x300, simulated harvesting mask: 0x0).
(cluster.cpp:295)
2025-09-01 18:38:54.603 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:54.699 | info     |   SiliconDriver | Harvesting mask for chip 1 is 0x204 (NOC0: 0x204, simulated harvesting mask: 0x0).
(cluster.cpp:295)
2025-09-01 18:38:54.702 | info     |   SiliconDriver | Opening local chip ids/pci ids: {0}/[0] and remote chip ids {1} (cluster.cpp:157)
2025-09-01 18:38:54.705 | info     |   SiliconDriver | All devices in cluster running firmware version: 255.255.0 (cluster.cpp:138)
2025-09-01 18:38:54.705 | info     |   SiliconDriver | Software version 6.0.0, Ethernet FW version 6.9.0 (Device 0) (cluster.cpp:935)
2025-09-01 18:38:54.705 | info     |   SiliconDriver | Software version 6.0.0, Ethernet FW version 6.9.0 (Device 1) (cluster.cpp:935)
Completed successfully on the device, with 65536 elements
2025-09-01 18:38:58.053 | info     |          Device | Closing user mode device drivers (tt_cluster.cpp:383)
```

Tenstorrent’s kernel driver protects against users concurrently running, but this is a little rudimentary

Sharing the Wormhole: Conflict, you are not running

```
[user-id@tenstorrent1 ~]$ ./ex_five
2025-09-01 18:38:19.201 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:19.214 | info     |          Device | Opening user mode device driver (tt_cluster.cpp:192)
2025-09-01 18:38:19.214 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:19.224 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:19.234 | info     |   SiliconDriver | Harvesting mask for chip 0 is 0x300 (NOC0: 0x300, simulated harvesting mask: 0x0). (cluster.cpp:295)
2025-09-01 18:38:19.236 | info     |   SiliconDriver | Opened PCI device 0; KMD version: 1.29.0; API: 1; IOMMU: disabled (pci_device.cpp:197)
2025-09-01 18:38:19.341 | info     |   SiliconDriver | Harvesting mask for chip 1 is 0x204 (NOC0: 0x204, simulated harvesting mask: 0x0). (cluster.cpp:295)
2025-09-01 18:38:19.346 | info     |   SiliconDriver | Opening local chip ids/pci ids: {0}/[0] and remote chip ids {1} (cluster.cpp:157)
2025-09-01 18:38:19.347 | info     |   SiliconDriver | All devices in cluster running firmware version: 255.255.0 (cluster.cpp:138)
2025-09-01 18:38:19.347 | info     |   SiliconDriver | Software version 6.0.0, Ethernet FW version 6.9.0 (Device 0) (cluster.cpp:935)
2025-09-01 18:38:19.348 | info     |   SiliconDriver | Software version 6.0.0, Ethernet FW version 6.9.0 (Device 1) (cluster.cpp:935)
2025-09-01 18:38:20.081 | info     |           Metal | While initializing device 0, active ethernet dispatch core (x=21,y=17) detected as still running, issuing exit
signal. (metal_context.cpp:586)
2025-09-01 18:38:20.082 | info     |           Metal | While initializing device 0, active ethernet dispatch core (x=18,y=17) detected as still running, issuing exit
signal. (metal_context.cpp:586)
2025-09-01 18:38:20.082 | info     |           Metal | While initializing device 0, active ethernet dispatch core (x=25,y=17) detected as still running, issuing exit
signal. (metal_context.cpp:586)
Read unexpected run_mailbox value: 0x40 (expected 0x80 or 0x0)
2025-09-01 18:38:20.082 | critical |          Always | Read unexpected run_mailbox value from core (x=25,y=17) (assert.hpp:107)
2025-09-01 18:38:20.082 | warning  |          Always | Detected dispatch kernels still running but failed to complete an early exit. This may happen from time to time
following a reset, continuing to FW intialization... (metal_context.cpp:632)
2025-09-01 18:38:20.088 | info     |           Metal | While initializing device 1, active ethernet dispatch core (x=18,y=16) detected as still running, issuing exit
signal. (metal_context.cpp:586)
2025-09-01 18:38:20.088 | info     |           Metal | While initializing device 1, active ethernet dispatch core (x=25,y=16) detected as still running, issuing exit
signal. (metal_context.cpp:586)
Read unexpected run_mailbox value: 0x40 (expected 0x80 or 0x0)
2025-09-01 18:38:20.088 | critical |          Always | Read unexpected run_mailbox value from core (x=25,y=16) (assert.hpp:107)
2025-09-01 18:38:20.088 | warning  |          Always | Detected dispatch kernels still running but failed to complete an early exit. This may happen from time to time
following a reset, continuing to FW intialization... (metal_context.cpp:632)
```

- The output is longer, it will pause, and you can see *detected as still running, issuing exit*
- Ctrl-C (or wait for timeout after around 20 seconds) and then rerun