

# 7 GPU mining rig using Debian Linux as OS

## Summary

To satisfy my interest in blockchain and mining technologies, I build a 7 GPU mining rig using a PRIME Z270-A motherboard and Debian Linux as operating system. Since I met some quite interesting challenges to get all 7 GPUs working fine, I decided to share my experience. I hope this story is of use for others building their own mining rigs.

## Hardware

After some experiments converting old PCs, with 3 PCI Express slots, into cheap mining rigs, I decided to build one from scratch using new components only.

The shopping list for my mining rig is:

1. One PRIME Z270-A motherboard with 7 PCI Express slots
2. One Intel Pentium G4400
3. One Artic Silver 5 heat transfer paste for good heat transfer from the cooling body to the processor
4. One Crucial standard 8 GB DIMM DDR4-2133 SDRAM
5. One Cooler Master MasterWatt Lite 700 Watt power supply
6. One Kingston A400 SSD 120 GB drive
7. Seven GPUs type GeForce GTX 1050 Ti OC 4G
8. Six riser sets for placing 6 of the 7 GPUs next to the motherboard.

The motherboard was chosen because of the large number of 7 PCI Express slots. Small disadvantage is that the lay-out of the motherboard is such that only one GPU can be placed on it. All other GPUs need to be placed elsewhere using risers but since these are very cheap if bought through, for example, AliExpress, this is quit acceptable.

The processor could also have been a cheaper Intel Celeron but the Intel Pentium G4400 was chosen for to make the system sufficiently fast to use it as a desktop computer during mining.

The power supply is rated 700 Watt, which is slightly overrated to increase its lifetime. After overclocking the 7 GPUs, the total system uses roughly 600 Watt and the power supply stays cool.

The 8 GB SDRAM and 120 GB SSD drive were chosen to speed up the system so that it can be used as Desktop computer during mining.

The GeForce GTX 1050 Ti OC 4G GPUs are definitely not the fastest ones for mining. On ethereum they mine about 10.5 MH/s (windows) to about 11.5 MH/s (Linux). After overclocking they appear to mine a very stable 13.4 MH/s. Since these GPUs are relatively cheap and have relatively low power consumption figures, they appeared to produce the highest mining profit per invested euro in The Netherlands, November 2017 as compared to other commercially available GPUs.

## Building the rig

There are plenty of youtube videos on how to connect power supply, processor, SDRAM, SSD Drive and riser sets to a motherboard so I'll skip this step.

Figure 1 shows a photo of the rig that was just built on a wooden plank.



Figure 1. Photo of the mining rig

### Configuration of the motherboard

Correct configuration of the BIOS appears to be crucial for making the system to work well with 7 GPUs. It was observed that, out of the box, the motherboard only supported 3 GPUs at the same time and connecting more GPUs resulted in a system that hangs before starting up the operating system. This issue occurs both in Windows and Linux.

A very good tutorial for configuring the PRIME Z270-A motherboard can be found on youtube  
<https://www.youtube.com/watch?v=DR-FkpU2KfI>

A short action list to configure the motherboard:

1. Remove all the USB 3.0 cables from the 7 PCI Express slots so that the GPUs are not connected to the motherboard.
2. Start-up the motherboard by pressing the start button on the motherboard and go into the BIOS. If no operating system is installed yet, this happens automatically, otherwise press the del button during startup until you end up in the BIOS.
3. Goto Advanced -> Tools ->Asus EZ Flash 3 utility and select via Internet (while the network cable is connected to the motherboard) -> next -> Flash update -> follow instructions and update the firmware through DHCP connection. During the process the motherboard will reboot. The firmware update for this rig was: PRIME-Z270-A-ASUS\_109.zip. If any future updates cause trouble, download this one from ASUS, save it on a USB stick and update the BIOS from USB stick.
4. After updating, enter the BIOS again, press F7 to enter the advanced mode -> advanced -> system agent SA conf. -> DMI/OPI conf. -> DMI max link speed must be set to gen2
5. Go back one level and select PEG Port conf. -> PCIE16-1 link speed -> choose gen2, then PCIE16-2 link speed -> chose gen2, then PCIe spread spectrum clocking -> choose disabled.
6. Go back 2 levels and select PCH configuration -> PCI Express conf. -> PCIe speed -> choose gen2.
7. Go back 2 levels and choose On board device configuration -> HD audio controller -> choose disabled.

8. Go back to advanced settings, choose APM configuration -> restore power loss -> choose power on
9. Go back to advanced settings -> boot section ->fast boot > choose enabled, Next boot -> choose fast boot, Post delay time -> choose 0 sec, Above 4G decoding -> choose enabled.
10. Now save these settings and reboot and your motherboard is prepared for dealing with 7 GPUs.

These motherboard settings make sure that all GPUs are recognized, and that the rig restarts automatically after power loss.

## **Installing the Debian Linux OS and software**

It is noted that in November 2017, Debian Stretch is the most recent distribution. However, it appeared that under Debian Stretch, issues occur with the NvidiaGraphicsDrivers. The drivers install correctly following the procedure on <https://wiki.debian.org/NvidiaGraphicsDrivers> but issues were encountered during configuration and overclocking of the GPUs. In order not to loose time, Debian Jessie 8.9 was installed and under this OS the drivers worked like a charm.

How to install Debian Linux from a USB stick is described in detail elsewhere. Here, I just mention a few headlines:

1. Download the netinstall debian-8.9.0-amd64-netinst.iso from a repo.
2. Copy it on a freshly formatted USB stick e.g., using dd if= debian-8.9.0-amd64-netinst.iso of=/dev/sdb bs=1M conv=noerror  
Note: in this case sdb is the device id of the usb stick
3. Put the USB stick in the motherboard, restart the system and enter the BIOS
4. Choose the USB stick as primary boot device (it may be the case that this step is not required) and save and exit the BIOS.
5. Choose Graphical installation and follow instructions.
6. Once the software is to be selected, choose MATE as favorite desktop and uncheck the others, also check the last option which is something like "install basic functionality"
7. Start-up the system and login as superuser by typing su and root password
8. Now add your username to the sudoers group: sudo usermod -aG sudo user
9. Make sure no password is asked for when using sudo: type su visudo and edit the file. Look to the line: %sudo ALL=(ALL:ALL) ALL and change it into ALL=(ALL:ALL) NOPASSWD:ALL
10. Realize autologin for the MATE GUI: edit the file /etc/lightdm/lightdm.conf and in this file search for the lines:  

```
#autologin-user=
#autologin-user-timeout=0
```

Remove the # from these lines and type your username after the = of the first line.
11. Connect 1 GPU to the slot closest to the rim of the motherboard and connect the monitor to this GPU. This is to make sure that a GPU is present during driver installation. It should also work without doing this but I observed with some motherboards that this is required to make sure that the GPUs are recognized after driver installation.
12. Install the drivers for Debian Jessie following the instructions on <https://wiki.debian.org/NvidiaGraphicsDrivers> Just for your own reference: Nvidia drivers Version 375.66 (via jessie-backports)
13. Install nvidia-cuda-toolkit by: sudo apt-get install nvidia-cuda-toolkit
14. It appears that after installing, one package is missing: libcurl4-openssl-dev. Install this manually by typing: sudo apt-get install libcurl4-openssl-dev
15. If you now connect 1 to 5 GPUs (always during power OFF), they will be recognized. You can check this by typing nvidia-settings.
16. If you connect more than 5 GPUs the system will hang before start-up. This has something to do with the max number of screens that is supported. So we will have to introduce virtual screens. For this purpose (and later for overclocking), nvidia-xconfig must be installed. You can

- do this by downloading the corresponding .deb file or from the repository. To do the latter, edit the file /etc/apt/sources.list and add the following line: deb http://ftp.de.debian.org/debian jessie main contrib
17. After the previous step type sudo apt-get update and then sudo apt-get install nvidia-xconfig  
Now the nvidia-xconfig will be installed.
  18. Now power off the computer, install 5 GPUs and login.
  19. Type: nvidia-xconfig-a --cool-bits=28 --allow-empty-initial-configuration  
This will automatically generate a new xorg.conf file in /etc/X11 and define a virtual screen per extra GPU
  20. Restart the computer, login and shut it down again. Now connect GPU 6 and GPU 7, restart the computer and you will see that now all GPUs are recognized. Now again type:  
nvidia-xconfig-a --cool-bits=28 --allow-empty-initial-configuration and a new xorg.conf configuration file will be made for all 7 GPUs.
  21. Now check with nvidia-settings that all GPUs are recognized and if yes then congratulations, you successfully installed 7 GPUs on one motherboard.
  22. Because of the virtual screens, the menu buttons on the task bars and the time may appear 7 times. You can click on one of them, then attach them to the task bar, then click on the attached one again and select remove from task bar. After this all 7 buttons disappear. After this you can add the button again and this time only 1 button will appear, as it should.

## **Installing mining software**

I used the claymore dual miner for dual mining of ethereum and siacoin in nanopool, see <https://eth.nanopool.org> for more info and downloading the required software.

Just for the sake of reproducibility: I used the following version of mining software that you can download at nanopool:

Claymore.s.Dual.Ethereum.Decred\_Siacoin\_Lbry\_Pascal.AMD.NVIDIA.GPU.Miner.v9.6.-.LINUX.tar.gz

Obviously you can also install your favorite mining software.

## **Autostarting the miner after power failure**

This appears to be less straightforward as it seems when you use the claymore mining software (and probably also other software). What you would like to happen after a power failure is that the computer automatically restarts, logs in and starts the miner.

The automatic restart was realized during configuration of the motherboard. The autologin during installation of the Debian Jessie software, so we only need to make sure that the miner automatically starts.

What we would like it to do is open a terminal, start the miner and send the output generated during the mining process to the terminal in order to monitor what is going on. However, to realize this, we should start the miner after logging into the GUI and not before that.

An obvious way to start the miner after each reboot would be to define a cronjob. However, this is executed before logging into the GUI. There are some tricks to go around this, but several issues were encountered.

A very simple procedure to deal with this is to go to System -> Control Center -> Startup Application and to activate a bash script there that starts the mining process. This appears to work using xterm by adding a start-up program using the menu and by typing following command line in the program:

xterm -hold -e sudo bash /home/user/Desktop/claymore/miningscript.bash

-hold makes sure that the terminal does not close. miningscript.bash is the mining script and it is assumed that this script is present in a claymore directory placed on the Desktop of user with name "user".

## **Overclocking of the miner**

During software installation, we have prepared the miner for overclocking. The overclocking procedure is also not straightforward since overclocking using the nvidia-settings functionality does not work because the software was intentionally developed to "forget all settings" when the computer is restarted.

In order to deal with this a bash script was made that overclocks the GPUs each time the computer is restarted:

```
nvidia-settings -a [gpu:0]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:0]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:1]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:1]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:2]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:2]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:3]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:3]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:4]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:4]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:5]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:5]/GPUMemoryTransferRateOffset[2]=1300  
nvidia-settings -a [gpu:6]/GPUGraphicsClockOffset[2]=175  
nvidia-settings -a [gpu:6]/GPUMemoryTransferRateOffset[2]=1300
```

This script overclocks the 7 GPUs thereby increasing the hasrate on ethereum from roughly 11.5 MH/s to 13.4 MH/s. The total hasrate of the system was most of the time 94 MH/s.

Now suppose the above script is in a file called overklokuuh.bash

You can make sure that this script is executed each time on start-up by adding a start-up program in System -> Control Center -> Startup Application and typing following command line in the program:  
xterm -e sudo bash overklokuuh.bash

In the above case, it is assumed that the file overklokuuh.bash is present in the user's home directory  
~/

**Cheers and have fun!!!**

**Mateo Mayer**