

# MxGPU Setup Guide for VMware®



**DISCLAIMER**

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of non-infringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD's products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

©2016 Advanced Micro Devices, Inc. All rights reserved. AMD, the AMD arrow, FirePro, and combinations thereof are trademarks of Advanced Micro Devices, Inc. in the United States and/or other jurisdictions. OpenCL is a trademark of Apple, Inc. and used by permission of Khronos. PCIe and PCI Express are registered trademarks of the PCI-SIG Corporation. VMware is a registered trademark of VMware, Inc. in the United States and/or other jurisdictions. Other names are for informational purposes only and may be trademarks of their respective owners.



Table of Contents

1. Overview ..... 4

2. Hardware and Software Requirements..... 4

    2.1 Hardware Requirements.....4

        2.1.1 Host/Server.....4

        2.1.2 Client.....4

    2.2 Software Requirements.....5

3. MxGPU Setup ..... 6

    3.1 Installing and Configuring amdgpuv using MxGPU Setup Script ..... 6

    3.2 Reconfiguring amdgpuv using MxGPU Setup Script ..... 9

    3.3 VF Pass Through ..... 10

    3.4 Assigning VFs Automatically Using MxGPU Setup Script..... 11

    3.5 Specifying Eligible Virtual Machines..... 13

4. Appendix..... 15

    4.1 Host Server Configuration..... 15

    4.2 Host Server Configuration..... 17



## 1. Overview

This setup guide details the advanced steps necessary to enable MxGPU on the AMD FirePro™ S7150 and S7150x2 family of products. The guide uses VMware® products as an example setup. These products include VMware ESXi™ as a hypervisor, the VMware vSphere® client and VMware Horizon® View™. The basic setup steps for the VMware software is detailed in the companion document to this one.

## 2. Hardware and Software Requirements

The sections below lists the hardware and software that are required for setting up the VMware environment.

### 2.1 Hardware Requirements

#### 2.1.1 Host/Server

Graphics Adapter: AMD FirePro™ **S7100X, S7150, S7150x2** for MxGPU and/or passthrough

\*\*\*note that the AMD FirePro™ S7000, S9000 and S9050 can be used for passthrough only

Sample of Certified Server Platforms:

- Dell PowerEdge R730 Server
- HPE ProLiant DL380 Gen9 Server
- SuperMicro 1028GQ-TR Server

Additional Hardware Requirements:

- CPU: 2x4 and up
- System memory: 32GB & up to 1TB; more guest VMs require more system memory
- Hard disk: 500G & up; more guest VMs require more HDD space
- Network adapter: 1000M & up

#### 2.1.2 Client

Any of the following client devices can be used to access the virtual machine once these VMs are started on the host server:

- Zero client (up to 4 connectors) with standard mouse/keyboard and monitor
- Thin client with standard mouse/keyboard and monitor running Microsoft® Windows® Embedded OS
- Laptop/Desktop with standard mouse/keyboard and monitor running with Microsoft® Windows® 7 and up



## 2.2 Software Requirements

| Product                 | Type                            | Install On         | Section | Version/Download Location                                                                                                                                                                                     |
|-------------------------|---------------------------------|--------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| AMD FirePro™ VIB Driver | Hypervisor Driver               | Host (Server)      | 3.1     | <a href="http://support.amd.com/en-us/download/workstation?os=VMware%20vSphere%20ESXi%206.0#catalyst-pro">http://support.amd.com/en-us/download/workstation?os=VMware%20vSphere%20ESXi%206.0#catalyst-pro</a> |
| AMD VIB Install Utility | Script                          | Host (Server)      | 3.1     | <a href="http://support.amd.com/en-us/download/workstation?os=VMware%20vSphere%20ESXi%206.0#catalyst-pro">http://support.amd.com/en-us/download/workstation?os=VMware%20vSphere%20ESXi%206.0#catalyst-pro</a> |
| PuTTY                   | SSH client                      | Host Admin. System |         | <a href="http://www.putty.org/">http://www.putty.org/</a>                                                                                                                                                     |
| SSH Secure Shell        | SSH Client and Download Utility | Host Admin. System | 3.1     |                                                                                                                                                                                                               |

**Table 1 : Required Software for Document**

(Links to non-AMD software provided as examples)



### 3. MxGPU Setup

The following sections describe the steps necessary to enable MxGPU on the graphics adapter(s) in the host. Before proceeding, refer to the **Appendix** to ensure that the host system is enabled for virtualization and SR-IOV. Once virtualization capabilities are confirmed for the host system, follow the steps in the next two sections to program the graphics adapter(s) for SR-IOV functionality and to connect the virtual functions created to available virtual machines.

#### 3.1 Installing and Configuring amdgpuv using MxGPU Setup Script

1. Download and unzip the vib and MxGPU-Setup-Script-Installer.zip from **Table 1**.
2. Copy MxGPU-Setup-Script-Installer.zip and amdgpuv vib to a persistent directory  
(Recommended : /vmfs/volumes/datastore1)
3. Navigate to the directory containing the script and vib file  
(Example : `# cd /vmfs/volumes/datastore1`)
4. Unzip the MxGPU-Setup-Script-Installer.zip file  
`# unzip MxGPU-Setup-Script-Installer.zip`
5. Open the “vms.cfg” file and specify VMs that will use the AMD virtual GPUs
  - a. Note: If all registered VMs on the host should be considered eligible, skip to step 6
6. Edit vms.cfg file to include VMs that should be considered for auto-assign
  - a. Examples are provided in the vms.cfg configuration file to help specify eligible VMs. See **3.5 Specifying Eligible Virtual Machines** for more information.
7. Change permissions of mxgpu-install.sh to be executable `# chmod +x mxgpu-install.sh`
8. Run command: `# sh mxgpu-install.sh` to get a list of available commands
9. Run command: `# sh mxgpu-install.sh -i <amdgpuv...vib>`
  - a. Alternatively, running `# sh mxgpu-install.sh -i` , the script will find the amdgpuv vib
    - i. If the “-i” option is used without a specified vib, ensure that there is only one amdgpuv vib in the current directory and all of its subdirectories.
    - ii. If there is more than one amdgpuv vib, then the script will exit
  - b. The script will check for system compatibility before installing the driver
  - c. After confirming system compatibility, the script will display all available AMD adapters
  - d. If any GPU’s are in passthrough mode, the script will provide the user with the option to remove them from passthrough mode. Note: any GPU’s in passthrough mode will not be virtualized.



10. Next, the script will show three options: Auto/Hybrid/Manual.

1) **Auto:** configures the same number of VFs to all GPUs

- The script first prompts for the number of virtual functions desired (per GPU) and sets all settings accordingly (frame buffer, time slice, etc...)
- The settings are applied to all AMD GPUs available on the bus
- The script will add the pciHole parameters to eligible VMs that are specified in the vms.cfg file
- A reboot is required for the changes to take effect
- Any GPU's still in passthrough mode will not be virtualized

2) **Hybrid:** configure the number of VFs for each GPU

- The script first prompts for the number of virtual functions desired (per GPU) and sets all settings accordingly (frame buffer, time slice, etc...)
- The settings are applied to the selected AMD GPU; the process repeats for every GPU
- The script will add the pciHole parameters to eligible VMs that are specified in the vms.cfg file
- A reboot is required for the changes to take effect
- Any GPU's in passthrough mode will not be virtualized; any configurations on these GPU's will not have any effect

3) **Manual:** Manually configure each GPU (Advanced)

- The script prompts the user to enter number of VFs, Frame Buffer size per VF (in MB) and time slice (in  $\mu$ s)
- The settings are applied to selected AMD GPU; the process repeats for the next GPU
- The script will add the pciHole parameters to eligible VMs that are specified in the vms.cfg file
- A reboot is required for the changes to take effect
- Any GPU's in passthrough mode will not be virtualized; any configurations on these GPU's will not have any effect

See **Appendix** for recommended settings for Frame Buffer Size and Time Slice duration. Auto and Hybrid configuration modes apply the recommended settings.



```

10.6.139.74 - PuTTY
[root@ /vmfs/volumes/571bdee7-2f300cf5-8e6f-6cc21739d6a0] chmod +x mxgpu-install.sh
[root@ /vmfs/volumes/571bdee7-2f300cf5-8e6f-6cc21739d6a0] sh mxgpu-install.sh
** is not a valid argument.

Usage: sh mxgpu-install.sh [-acr] [-i VIBFILE] [-u VIBFILE]
Install, configure, update, auto-assign, or remove AMD amdgpv driver

-i <VIB name>* Install and Configure amdgpv driver
-c          Configure amdgpv driver
-a          Automatically assign or unassign VFs to VMs
-u <VIB name>* Update to specified amdgpv driver
-r          Remove amdgpv driver and pciHole parameters
-ap         Add pciHole parameters
-rp         Remove pciHole parameters

*: Optional VIB can be specified, otherwise it will look for an amdgpv VIB in the current directory.
[root@ /vmfs/volumes/571bdee7-2f300cf5-8e6f-6cc21739d6a0] sh mxgpu-install.sh -i amdgpv-1.00-1OEM.600.0.0.2494585.x86_64-CL1263512.vib
Checking environment.....
+++++Installing driver
+++++Installation Completed Successfully!
Detected 4 SR-IOV GPU
0000:0b:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:0f:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:89:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:8d:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
Start configuration....
Enter the configuration mode([A]uto/[H]ybrid/[M]anual,default:A)Auto
Please enter number of VFs:(default:4): 4
Configuring the GPU 1 ...
0000:0b:00.0 Display controller: AMD Tonga S7100X
adapter1_conf=11,0,0,4,1920,5000,1
Configuring the GPU 2 ...
0000:0f:00.0 Display controller: AMD Tonga S7100X
adapter2_conf=15,0,0,4,1920,5000,1
Configuring the GPU 3 ...
0000:89:00.0 Display controller: AMD Tonga S7100X
adapter3_conf=137,0,0,4,1920,5000,1
Configuring the GPU 4 ...
0000:8d:00.0 Display controller: AMD Tonga S7100X
adapter4_conf=141,0,0,4,1920,5000,1
Setting up SR-IOV settings...
Done
pciHole.start = 2048
pciHole.end = 3584
Eligible VMs:
Win7-
- Win7-
Win10-
Win7L
Win7-
- Win7-Baseline
- Win7-
- Win7-
marcus-
win7L
Win10-
PCI Hole settings will be added to these VMs. Is this OK?[Y/N]Y
Completed
The configuration needs a reboot to take effect

To automatically assign VFs, please run "sh mxgpu-install.sh -a" after system reboot
[root@ /vmfs/volumes/571bdee7-2f300cf5-8e6f-6cc21739d6a0]

```

Figure 1 : Screenshot of MxGPU Setup Script Installation Flow

### 3.2 Reconfiguring amdgpuv using MxGPU Setup Script

1. Run command: `# sh mxgpu-install.sh -c`
2. The “-c” option allows you to quickly reconfigure your amdgpuv. For help with configuration options, see page 8.

```
To automatically assign VFs, please run "sh mxgpu-install.sh -a" after system reboot
[root@ /vmfs/volumes/571bdee7-2f300cf5-8e6f-6cc21739d6a0] sh mxgpu-install.sh -c
Detected 4 SR-IOV GPU
0000:0b:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:0f:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:89:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
0000:8d:00.0 VGA compatible controller Display controller: AMD Tonga S7100X
Start configuration....
Enter the configuration mode ([A]uto/[H]ybrid/[M]anual,default:A)Hybrid
Configuring the GPU 1 ...
0000:0b:00.0 Display controller: AMD Tonga S7100X
Please enter number of VFs:(default:4): 2
adapter1_conf=11,0,0,2,3840,15000,1
Configuring the GPU 2 ...
0000:0f:00.0 Display controller: AMD Tonga S7100X
Please enter number of VFs:(default:4): 3
adapter2_conf=15,0,0,3,2560,6000,1
Configuring the GPU 3 ...
0000:89:00.0 Display controller: AMD Tonga S7100X
Please enter number of VFs:(default:4): 4
adapter3_conf=137,0,0,4,1920,5000,1
Configuring the GPU 4 ...
0000:8d:00.0 Display controller: AMD Tonga S7100X
Please enter number of VFs:(default:4): 5
adapter4_conf=141,0,0,5,1536,6000,1
Setting up SR-IOV settings...
Done
pciHole.start = 2048
pciHole.end = 3584
Eligible VMs:
Win7-
    Win7-
    Win10
win7
Win7
    Win7-
    Win7-
    Win7-
marcus-
win7
W10-'
PCI Hole settings already exist on all eligible VMs.
The configuration needs a reboot to take effect

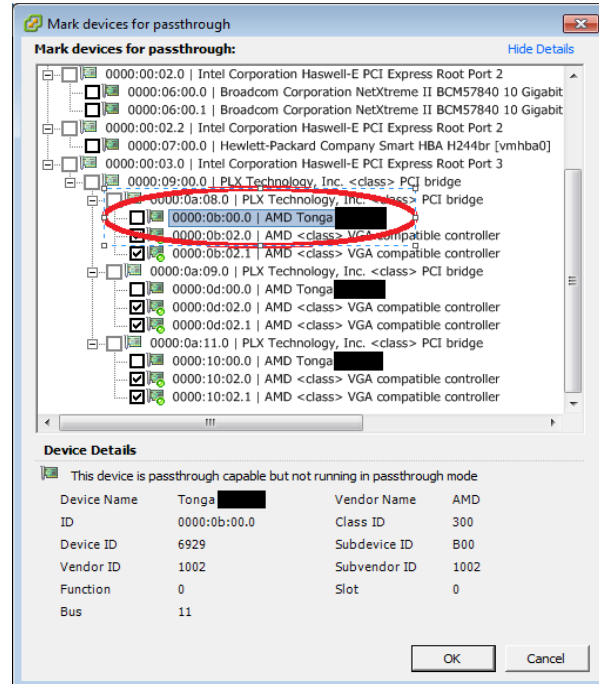
To automatically assign VFs, please run "sh mxgpu-install.sh -a" after system reboot
```

Figure 2 : Screenshot of MxGPU Setup Script reconfiguration

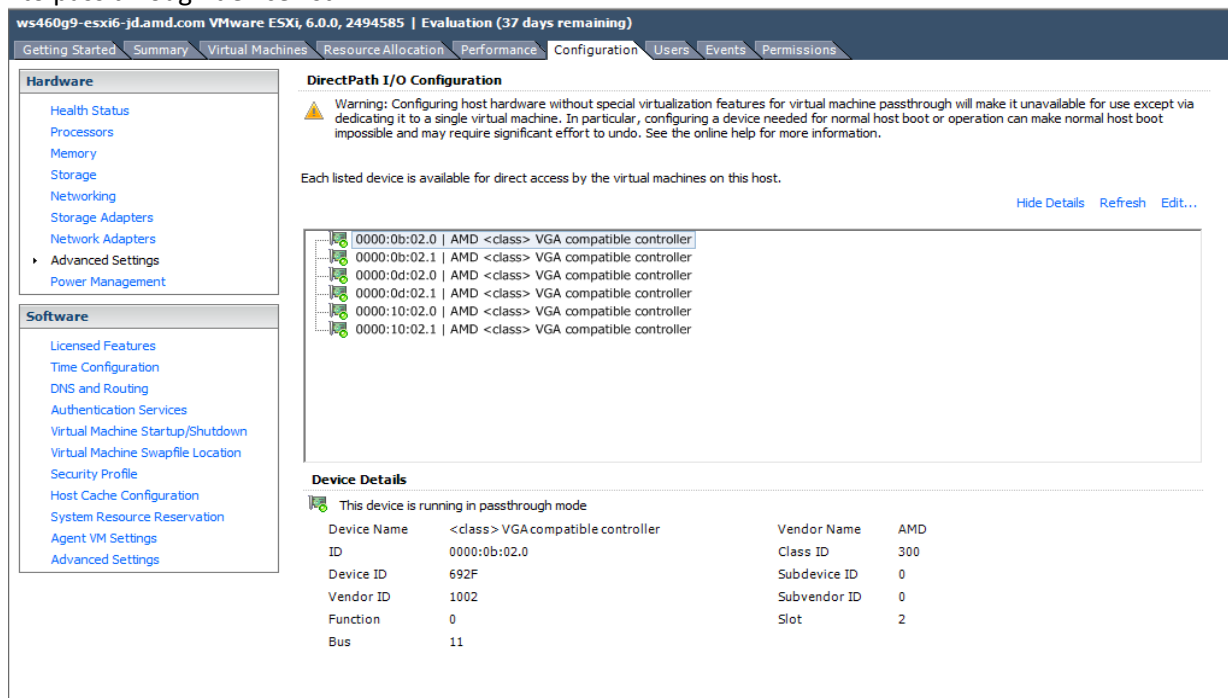


### 3.3 VF Pass Through

Once the VFs (virtual functions) are set up, then passing through these VFs follows the same procedure as passing through a physical device. To successfully pass through the VFs, the physical device CANNOT be configured as a passthrough device. If the physical device is being passed through to the VM, then the GPUV driver will not install properly. If that happens, the VFs will not be enabled and no VFs will be shown.



Once the VFs are enabled, they will be listed in the available device list for pass through, and the status of the PF will be changed to unavailable for pass through. No additional operation is needed to move VF into pass through device list.



### 3.4 Assigning VFs Automatically Using MxGPU Setup Script

1. After rebooting the system and the VFs are populated on the device list, navigate to the directory containing the mxgpu-install.sh script
2. Start the auto-assign option `# sh mxgpu-install.sh -a`
3. Select the command you want to execute:

| Command  | Description                                             |
|----------|---------------------------------------------------------|
| Assign   | Assign VFs to all eligible VMs, if there are enough VFs |
| Unassign | Unassign VFs from all eligible VMs                      |
| List     | List all VM and GPU configurations on the host          |
| Help     | Print a helpful message with these descriptions         |

- Assign will exit if the number of eligible VMs exceeds the number of VFs.
  - Unassign will only unassign VFs from eligible VMs.
  - By default, all VMs on the host are considered eligible. Modify vms.cfg to specify which VMs are eligible for auto-assign. See **3.5 Specifying Eligible Virtual Machines** for example configuration files.
4. Power on the VMs and refer to **Section 3.4** of the [Basic Virtual Environment Setup](#) for installing the GuestOS driver in a VM
  5. Known Issue: After reconfiguring amdgpudev and restarting, VMs show ghost VFs that are assigned but not actually available on the bus.

Workaround: Run auto-assign with the unassign option to unassign all the ghost VFs and then run auto-assign with the assign option to assign real VFs to VMs.



```
[root@:/vmfs/volumes/564c6bc4-f9eff6db-e29b-3863bb3305d4] sh mxgpu-install.sh -a
AMD MxGPU Multi-Assign Script
Usage : [assign|unassign|list|help]

assign : Assign VFs to selected VMs automatically. By default, it will read VM filters from vms.cfg
unassign : Unassign VFs from the selected VMs. By default, it will read VM filters from vms.cfg
list : Print vm and card infomation tables
help : Print this help

Note :
Edit vms.cfg to specify VMs that are eligible for Multi-Assign.
By default, vms.cfg will detect all registered VMs as eligible.

Please select an option [assign|unassign|list|help]: list
| VM ID | VM NAME | GPU ASSIGNED |
| 24 | win7vm1 | NONE |
| 25 | win7vm2 | NONE |
| 26 | win7vm3 | NONE |
| 27 | win7vm4 | NONE |
| 31 | win7vm8 | NONE |
| 32 | win7vm5 | NONE |
| 33 | win7vm6 | NONE |
| 34 | win7vm7 | NONE |
| 36 | win10vm1 | NONE |
| 37 | vm | NONE |

| CARD NAME | Total VFs | Used VFs |
| Tonga S7150 | 5 | 0 |
| Tonga S7150 | 5 | 0 |

Please select an option [assign|unassign|list|help]: assign
Eligible VMs:
win7vm1
win7vm2
win7vm3
win7vm4
win7vm5
win7vm6
win7vm7
win10vm1
vm
These VMs will be assigned a VF, is it OK?[Y/N]y
VM win7vm1 configure over
VM win7vm2 configure over
VM win7vm3 configure over
VM win7vm4 configure over
VM win7vm8 configure over
VM win7vm5 configure over
VM win7vm6 configure over
VM win7vm7 configure over
VM win10vm1 configure over
VM vm configure over
done
```

Figure 2 : Screenshot of Multi-Assign Usage

```
Please select an option [assign|unassign|list|help]: list
| VM ID | VM NAME | GPU ASSIGNED |
| 24 | win7vm1 | ASSIGNED |
| 25 | win7vm2 | ASSIGNED |
| 26 | win7vm3 | ASSIGNED |
| 27 | win7vm4 | ASSIGNED |
| 31 | win7vm8 | ASSIGNED |
| 32 | win7vm5 | ASSIGNED |
| 33 | win7vm6 | ASSIGNED |
| 34 | win7vm7 | ASSIGNED |
| 36 | win10vm1 | ASSIGNED |
| 37 | vm | ASSIGNED |

| CARD NAME | Total VFs | Used VFs |
| Tonga S7150 | 5 | 5 |
| Tonga S7150 | 5 | 5 |
```

Figure 3 : Screenshot of Multi-Assign “list” command after assigning VFs to VMs

### 3.5 Specifying Eligible Virtual Machines

This section will detail how to specify which Virtual Machines on your host will be considered eligible for adding and removing **PCI Hole Settings** as well as assigning and unassigning VFs in **MxGPU Auto-Assign**.

Use your preferred text editor to edit “vms.cfg”. This file is contained in MxGPU-Setup-Script.zip and should be present in the same directory as mxgpu-install.sh.

The examples in this section will assume the built-in text editor **vi**.

1. Use the command `# vi vms.cfg` to open the file. For help with using vi, an example can be found in the following [VMware article](#)
2. Remove the following line before making any changes as this line specifies all VMs on the host as eligible  

```
/ #REMOVE this line before making any changes to this file.
```
3. Methods to specify eligible VMs

#### I. Specify all VMs in a Directory or Datastore

Enter the path to a directory or datastore under the “Datastore/Directories” section. All VMs under that directory or datastore will be recursively considered an eligible VM. You can enter multiple directories in this section, one per line.

Example that specifies all VMs in datastore1 and all VMs in the directory “datastore2/Windows10” to be eligible.

```
#-----Datastores/directories-----#
#Enter the location of your VMs.
#
#/vmfs/volumes/datastore1/           This will find all VMs in datastore1
#/vmfs/volumes/datastore5 (1)/AMD/win7/ This will find all VMs in the subdirectory AMD/win7/
/vmfs/volumes/datastore1/
/vmfs/volumes/datastore2/Windows10/
#
#-----VM Names-----#
```



## II. Specify VMs by Name

Enter the name of VMs under the “VM Names” section, one per line. These VMs will be considered eligible.

Example that specifies a few VMs by name

```
-----VM Names-----#
#Enter the exact names of VMs
#NoReg: #DO NOT change or remove this line. Used for internal parsing.
#AMD-Win10-Pro          This will only match the VM with the exact name "AMD-Win10-Pro"
Win7-VM
Win10-VM
MxGPU-Performance VM
-----Regular Expressions-----#
```

## III. Specify VMs by Regular Expression

Enter a regular expression that can match multiple VMs that follow a similar naming pattern. This can be useful to specify a collection or pool of VMs.

Example that specifies all VMs that start with “Win7-” and “Win10-”

```
-----Regular Expressions-----#
#Enter a regex that matches a set of your VMs names. Useful if you want to select a pool of VMs that have a predictable name
#
#.*          This will match all VMs in this machine. The '.' means any character including numbers, symbols and
#windows\..1-64x-\d+ This will match VMs with names such as 'win8.1-64x-1' or 'win8.1-64x-123'. The '\.' means a bare '.'
#Reg: #DO NOT change or remove this line. Used for internal parsing.
Win7-.*
Win10-.*
```

Note: All three methods can be used in combination to specify eligible VMs. For example, you can specify all VMs in datastore1 and all VMs that start with “Win7-”. If there is any overlap and there are “Win7-” VMs in datastore1, they will still be eligible.



## 4. Appendix

### 4.1 Host Server Configuration

To enable the MxGPU feature, some basic virtualization capabilities need to be enabled in the SBIOS. These capabilities may be configured from the SBIOS configuration page during system boot up. Different system BIOS vendors will expose different capabilities differently. Some may have one control that enables a number of these capabilities. Some may expose controls for some capabilities while hardcoding others. The following settings, taken from an American Megatrends system BIOS, provides a list of the minimal set of capabilities that have to be enabled :

➤ **Server CPU supports MMU**

➤ **Server chipset supports AMD IOMMU or Intel VT-d**

The option “Intel VT for Directed I/O (VT-d)” should be enabled

Example Path : IntelRCSetup ➔ IIO Configuration ➔ Intel(R) VT for Directed I/O (VT-d) ➔ Intel VT for Directed I/O (VT-d)

➤ **Server (SBIOS) support PCIe standard SR-IOV**

The option “SR-IOV Support” should be enabled.

Example Path : Advanced ➔ PCI Subsystem Settings ➔ SR-IOV Support

➤ **Server (SBIOS) support ARI (Alternative Routing ID)**

The option “ARI Forwarding” should be enabled.

Example Path : Advanced ➔ PCI Subsystem Settings ➔ PCI Express GEN 2 Settings ➔ ARI Forwarding

➤ **Server (SBIOS and chipset (root port/bridge)) supports address space between 32bit and 40bit**

If there is an “Above 4G Decoding” enable it.

Example Path : Advanced ➔ PCI Subsystem Settings ➔ Above 4G Decoding

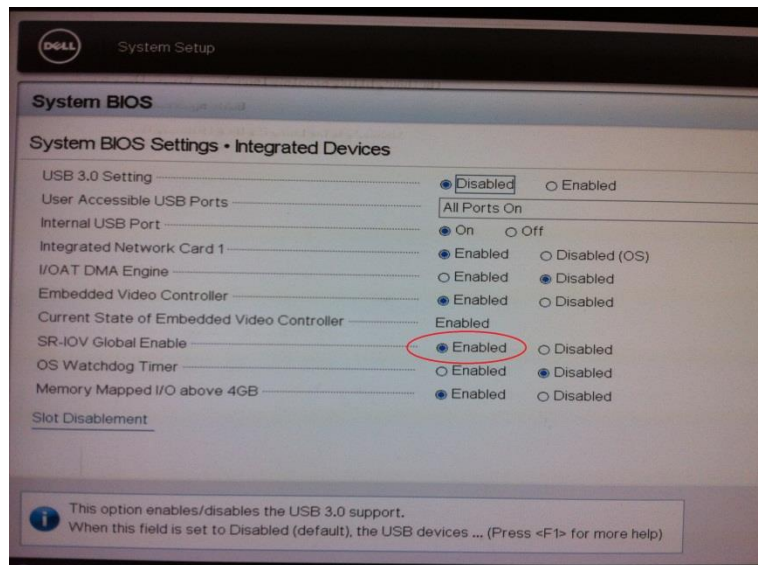
➤ **Server (Chipset (root port / bridge)) supports more than 4G address space**

There may be an option “MMIO High Size” for this function (default may be 256G).

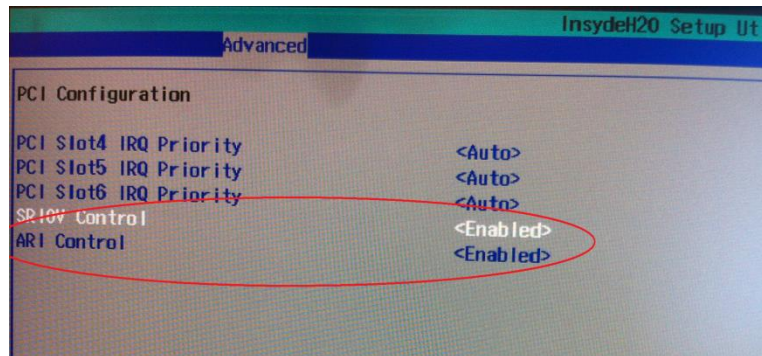
Example Path : IntelRCSetup ➔ Common RefCode Configuration ➔ MMIO High Size



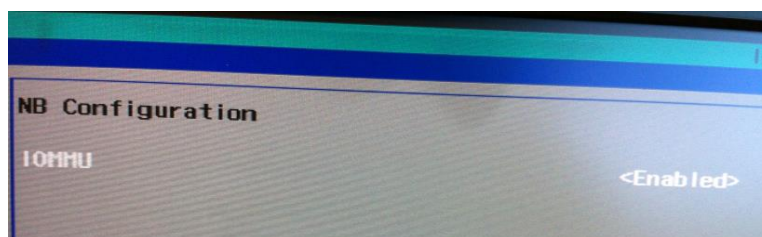
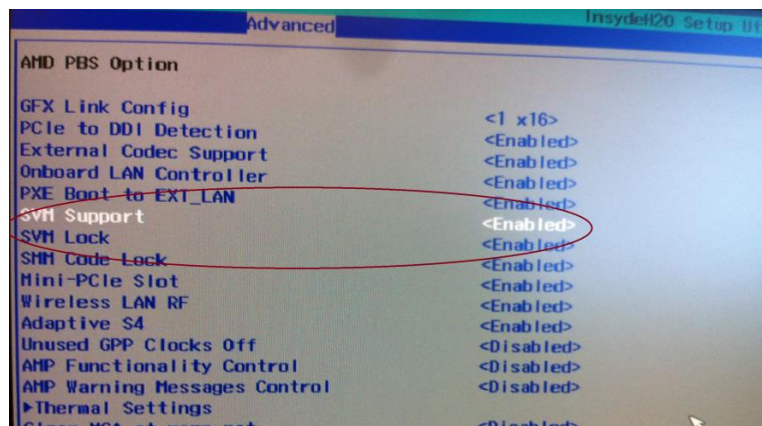
The following example shows how to enable SR-IOV on a Dell R730 platform.



On some platforms, the SBIOS configuration page provides more options to control the virtualization behavior. One of these options is the ARI (alternative reroute interface) as shown below.



In addition, some platforms also provide controls to enable/disable SVM and/or IOMMU capability. These options must be enabled on the platform.



## 4.2 Host Server Configuration

Default Configurations:

Please see the table below for the recommended Frame Buffer size and Time Slice for different VF configurations. These are the default settings when configuring MxGPU with the Auto and Hybrid modes. Recommended timeslices come from testing and it was found that the values below give the best performance.

| # of VFs | FrameBuffer Size (in MB) | Timeslice (in $\mu$ s) |
|----------|--------------------------|------------------------|
| 2        | 3840                     | 15000                  |
| 3        | 2560                     | 6000                   |
| 4        | 1920                     | 5000                   |
| 5        | 1536                     | 6000                   |
| 6        | 1280                     | 6000                   |
| 7        | 1097                     | 6000                   |
| 8        | 960                      | 6000                   |
| 9        | 853                      | 6000                   |
| 10       | 768                      | 6000                   |
| 11       | 698                      | 6000                   |
| 12       | 640                      | 6000                   |
| 13       | 590                      | 6000                   |
| 14       | 548                      | 6000                   |
| 15       | 512                      | 6000                   |
| 16       | 480                      | 6000                   |

Frame Buffer Size, also known as Memory size or VRAM size, specifies how much memory is available to each VF

Time Slice is the interval of time that each VF has access to the core resources of the GPU before switching to the next VF.

Note: MxGPU is limited to 15 VFs per GPU due to a limitation in VMWare ESXi 6.0.

