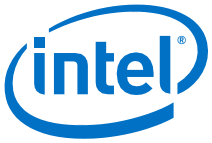
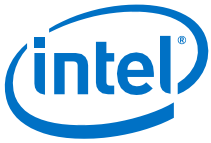


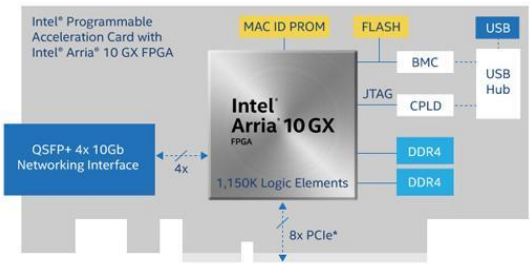
Intel SR-IOV FPGA Driver and Tools for VMware Hypervisor - Version 1.2.0

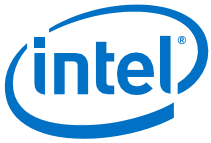


Contents

Introduction

Intel SR-IOV FPGA driver and tools for VMware Hypervisor for Intel Xeon® CPU with FPGAs, a collection of SR-IOV driver, firmware, and tools developed by Intel to make it easier to deploy Intel FPGAs for workload optimization in the data center. This PCI Express* (PCIe*)-based FPGA accelerator card for data centers offers both inline and lookaside acceleration. , Intel Arria® 10 GX FPGA provides the performance and versatility of FPGA acceleration and is one of several platforms supported by the Acceleration Stack for Intel® Xeon® CPU with FPGAs. The versatile Intel Programmable Acceleration Card with Intel Arria® 10 GX FPGA (Intel PAC with Intel Arria 10 GX FPGA) can be implemented in many market segments, such as big data analytics, artificial intelligence, genomics, video transcoding, cybersecurity, and financial trading.





Product Overview

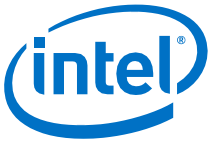
The release of the Intel SR-IOV FPGA driver and Tools for VMware Hypervisor includes SR-IOV FPGA driver and a set of management and monitoring tools supporting Intel programming accelerator card with Intel Arria® 10 GX FPGA on VMware Hypervisor version ESXi 6.7 U2.

Download

Download the Intel® Acceleration Stack SR-IOV driver for VMware Hypervisor from VMware. Navigate to https://www.vmware.com/resources/compatibility/search.php?deviceCategory=io and select the I/O Device Type "Hardware Acceleration" and click Update and View Results. Under Search Results, click "Intel Arria® 10 GX FPGA". Under the device driver name and version, click the link to download the driver.

The set of FPGA management and monitoring tools (PACDs) are packaged into an additional partner signed VIB and is available from Intel Corporation. The package can be obtained by sending request to vfpga-support@intel.com.

The guest VM runs the Intel® Acceleration Stack Version 1.2 downloadable from - https://www.intel.com/content/www/us/en/programmable/solutions/acceleration-hub/acceleration-stack.html



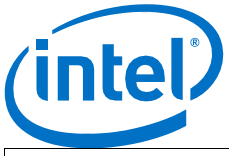
Acceleration Acronym List

Use the following table as a reference when reviewing the release notes.

Tale 1. Acronyms

Acronyms	Expansion	Description
AFU	Accelerator Functional Unit	Hardware Accelerator implemented in FPGA logic which offloads a computational operation for an application from the CPU to improve performance.
AF	Accelerator Function	Compiled Hardware Accelerator image implemented in FPGA logic that accelerates an application. An AFU and associated AFs may also be referred to as GBS (Green-Bits, Green BitStream) in the Acceleration Stack installation directory tree and in source code comments.
FIM	FPGA Interface Manager	The FPGA component containing the FPGA Interface Unit (FIU) and external interfaces for memory, networking, etc. The FIM may also be referred to as BBS (Blue-Bits, Blue BitStream) in the Acceleration Stack installation directory tree and in source code comments. The Accelerator Function (AF) interfaces with the FIM at run time.

continued...



Acronyms	Expansion	Description
VOB	VSphere Observation	PACD (sensor-monitor) sends VOB message to vSphere in case of thermal breach
BMC	Board Management Controller on the FPGA PAC	Provides on boards sensor data. It also supports firmware update
PR	Partial Reconfiguration	The ability to dynamically reconfigure a portion of an FPGA while the remaining FPGA design continues to function.

Installation details

Following assumptions are made for the rest of the document

Prompt #> for ESXi host SSH session

Prompt $> for Virtual Machine SSH session

Installation:

- Copy VMware FPGA SR-IOV driver VIB package (ifpga-1.2.0-1OEM.670.0.0.xxxxxxx.x86_64.vib) to ESXi host
- #> esxcli software vib install –v <path to vib> --maintenance-mode
- fpga software stack will be installed under /opt/intel/fpga
- Toggle passthrough if passthrough is enabled for the PF device.
- Execute /opt/intel/fpga/scripts/fpga-sriov-enable.sh
- Refresh host UI
- Make sure VFs is visible in hostUI (manage/pci Device) to be able to add them to guest VM
- Make sure ifpga driver is loaded by invoking the following command. ifpga should be listed

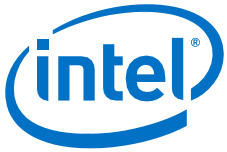
  #>vmkload_mod –l |grep ifpga

- Make sure FPGA VFs are enabled. For each Rush Creek (a10) PAC card there should be one VF enabled.

  #>lspci |grep acc

  0000:18:00.0 Processing accelerators: Intel Corporation FPGA DCP
  0000:18:00.1 Processing accelerators: Intel Corporation Device 09c5 [PF_0.24.0_VF_0]
  0000:af:00.0 Processing accelerators: Intel Corporation FPGA DCP
  0000:af:00.1 Processing accelerators: Intel Corporation Device 09c5 [PF_0.175.0_VF_0]
- source init_env.sh i.e.

  #>source /opt/intel/fpga/init_env.sh



FPGA CLI Tools

The set of FPGA management and monitoring tools (PACDs) are available through a separate partner signed VIB package from Intel Corporation and can be obtained by sending request to vfpga-support@intel.com.

Commented [BL1]: Insert Intel link here

Once the CLI Tools VIB package is installed then all the fpga management and monitoring tools are available under /opt/intel/fpga/tools. We have three tools as shown below -

#> ls /opt/intel/fpga/tools

- ifpga-cli
- fpgainfo
- fpgaflash

Each of the above tools usage is described next.

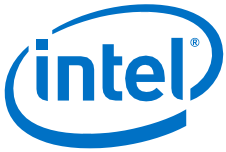
Ifpga-cli (/opt/intel/fpga/tools/ifpga-cli)

- The tool allows us to read/write various fpga attributes. All attributes are checked by the driver if they are valid before allowing either read/write. Any invalid attribute input will be rejected by the driver.
- Ifpga-cli read –l to get the list of read attributes
- Ifpga-cli read <attribute name>
- Ifpga-cli write –l to get the list of write attributes
- Ifpga-cli write <attribute name> <value>
- We also perform partial reconfiguration (i.e. programming an AFU with a GBS image) with this tool.

  usage: ifpga-cli pr [-h] [-l] [-p PORT] [endpoint] [file]

  - To get the list of valid endpoints invoke – ifpga-cli pr -l
  - –p can be ignored as currently there is only one port (i.e. 0 which is default)
  - Also, make sure Virtual machine is not using the AFU (i.e. VF) while performing partial reconfiguration. You can also disable SRIOV by invoking fpga-sriov-disable.sh. Following successful PR operation, you can re-enable SRIOV by invoking fpga-sriov-enable.sh

  Only valid endpoints (fpga fme) are accepted. Any invalid endpoint will be checked and rejected by the driver. Also, input GBS (green bit stream) file will be checked by the driver for compatibility. Only GBS file compatible with current FPGA will be accepted by the driver.



fpgainfo (/opt/intel/fpga/tools/fpgainfo)

Usage: fpgainfo [-h] [{errors}] {bmc,fme,port}

- This tool reads data from various fpga hardware components such as fme (fpga management engine), bmc (board management controller on the fpga card), AFU port (accelerator function unit)

fpgaflash (/opt/intel/fpga/tools/fpgaflash)

usage: fpgaflash [-h] {user, factory, bmc_bl, bmc_app} dev_name

- This tool allows to update BBS and BMC image on the fpga card. The images are read and some basic check is performed before flashing them onto the fpga hardware.
- A power cycle is needed following successful update. To get the fme dev name invoke ifpga-cli pr –l
- BBS image corresponding to Intel® Acceleration Stack Version 1.2 1.2 package (described later) is stored at /opt/intel/fpga/hw
- BMC firmware and bootloader image are located at /opt/intel/fpga/fw

PACD (Programmable accelerator Card Daemon)

We have two pacds available at /opt/intel/fpga/pacd. We need a separate SSH session for each pacd to be continuously running.

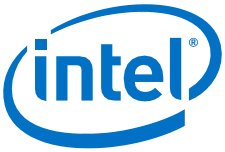
Graceful Thermal Shutdown/Thermal monitoring (opt/intel/fpga/pacd/sensor-monitor)

#>/opt/intel/fpga/sensor-monitor -h

Usage: sensor-monitor <options>

-g,--default-bitstream <file> Default bitstream (for thermal shutdown handling)
-p,--default-threshold <default threshold> Percentage of Default threshold (for thermal shutdown handling)

- To avoid sudden shutdown of PAC card by on-board BMC due to breaching of any unrecoverable sensor (temperature, current, volts, etc.) Thresholds (either Upper or Lower) this tool preemptively takes action before BMC to avoid ESXi platform PSOD.
- The tool continuously read BMC sensor data from all fpga hardware. The tool assumes a default 90% of actual configured non-recoverable threshold for temperate, current, voltage and power programmed on the BMC sensor to trigger shutdown VMs and prevent fpga usage until sensor



reading becomes normal. In addition, the tool programs the nlb_mode_0.gbs on the affected card. nlb_mode_0.gbs is stored at /opt/intel/fpga/gbs

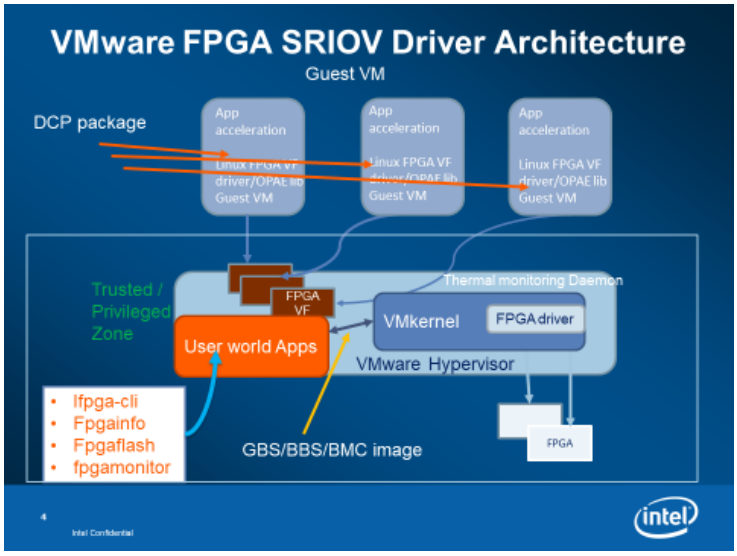
- Default 90% threshold could be changed with –p config parameter.
- A vob message is sent to vSphere indicating thermal breach on the affected fpga device. Administrator can setup an alert for the fpga vob message.
- A similar message is also displayed on the ESXi SSH console.
- Sensor readings are logged into /tmp/sensort.log which is periodically reset.

Error Interrupt Monitoring (/opt/intel/fpga/pacd/error-interrupt-monitor)

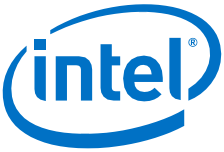
- This pacd captures any error interrupt (due to fatal/non-fatal/pci errors) for any of the PAC in the system. The error needs to be cleared to be able to receive any further error interrupt.
- The error is cleared by inspecting the error outputs on the console and writing the same value to the error attribute with ifpga-cli write <error attribute > < error value received on console>

FPGA usage under SR-IOV

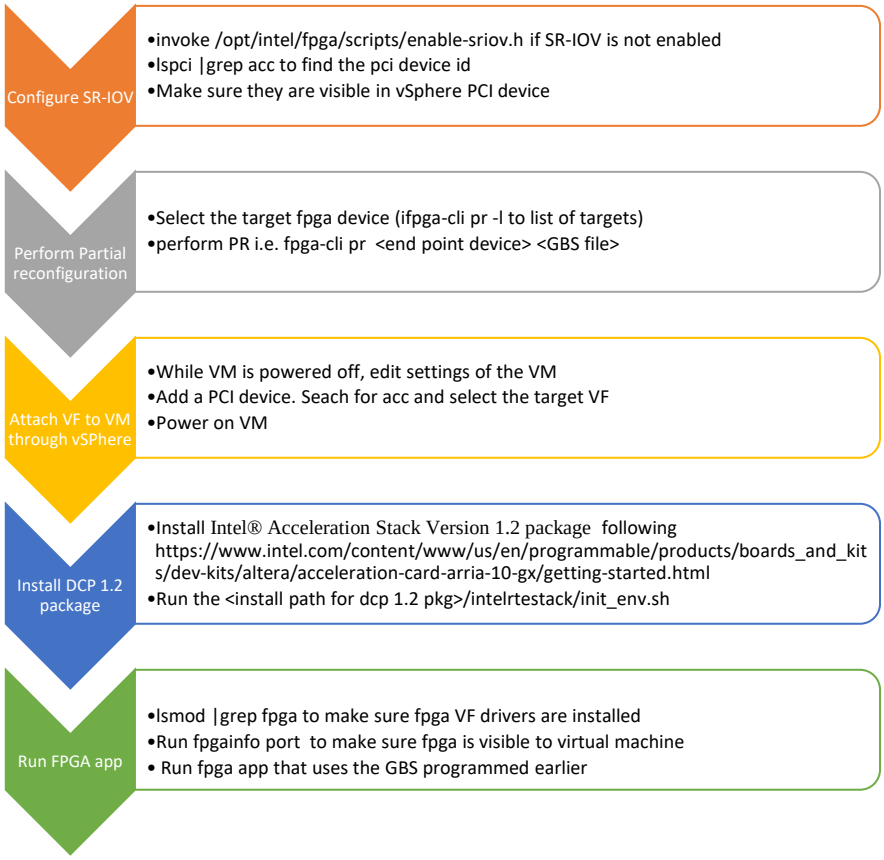
FPGA SR-IOV Driver architecture is shown below. ESXi fpga PF driver and tools installation steps are described earlier. The guest VM runs the Intel® Acceleration Stack Version 1.2 downloadable from - https://www.intel.com/content/www/us/en/programmable/solutions/acceleration-hub/acceleration-stack.html



Driver Architecture and trusted zone

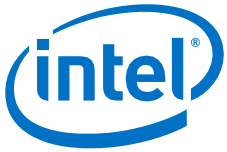


To use FPGA acceleration from guest virtual machine follow the steps below -



Running OpenCL fpga application within virtual machine

- Setup the guest virtual machine with OpenCL run-time SDK as per the Intel® Acceleration Stack Version 1.2 getting started guide
- Since, we cannot program AFU from virtual machine, we would need to extract GBS file from OpenCL kernel aocx file. Follow the steps below to extract the GBS file within VM.
- $> aocl binedit <opencl kernel>.aocx get .acl.fpga.bin fpga.bin
- $>aocl binedit fpga.bin get .acl.GBS.gz acl.GBS.gz



- $>gzip -d acl.GBS.gz
- scp the GBS file onto ESXi host and program it against a target fpga device.
- Once the GBS is programmed from ESXi host using ifpga-cli tool, set the following environment variable in guest virtual machine to prevent PR from guest.
- $> export CL_CONTEXT_COMPILER_MODE_INTELFPGA=3
- Guest virtual machine now should be able to run OpenCL app acceleration using fpga.

Contents of ifpga vib –

size	dir/file
0	bin/
0	etc/
0	lib/
0	opt/
0	usr/
3813	bin/bin2hex.py
4672	bin/hex2bin.py
4537	bin/hex2dump.py
2845	bin/hexdiff.py
3657	bin/hexinfo.py
6202	bin/hexmerge.py
0	etc/vmware/
0	etc/vmware/default.map.d/
185	etc/vmware/default.map.d/ifpga.map
0	lib/python3.5/
0	lib/python3.5/site-packages/
0	lib/python3.5/site-packages/intelhex/
51252	lib/python3.5/site-packages/intelhex/__init__.py
1876	lib/python3.5/site-packages/intelhex/__main__.py
123	lib/python3.5/site-packages/intelhex/__version__.py
9290	lib/python3.5/site-packages/intelhex/bench.py
4955	lib/python3.5/site-packages/intelhex/compat.py
2175	lib/python3.5/site-packages/intelhex/getsizeof.py
67633	lib/python3.5/site-packages/intelhex/test.py
0	opt/
0	opt/intel/fpga/
0	opt/intel/fpga/docs/
0	opt/intel/fpga/fw/
0	opt/intel/fpga/gbs/
0	opt/intel/fpga/hw/

0	opt/intel/fpga/pacd/
0	opt/intel/fpga/scripts/
0	opt/intel/fpga/tools/
68	opt/intel/fpga/init_env.sh
845519	opt/intel/fpga/docs/vmware-fpga-sriov-driver-qs.pdf
281250	opt/intel/fpga/fw/a10sa4-26889-fw.hex
72267	opt/intel/fpga/fw/a10sa4_bootloader-26879-fw.hex
1.36E+08	opt/intel/fpga/gbs/nlb_mode_0.gbs
89685376	opt/intel/fpga/hw/dcp_1_2.rpd
30921	opt/intel/fpga/pacd/error-interrupt-monitor
180395	opt/intel/fpga/pacd/sensor-monitor
213	opt/intel/fpga/scripts/fpga-sriov-disable.sh
192	opt/intel/fpga/scripts/fpga-sriov-enable.sh
22061	opt/intel/fpga/tools/fpgaflash
38503	opt/intel/fpga/tools/fpgainfo
8412	opt/intel/fpga/tools/ifpga-cli
0	usr/lib/
0	usr/share/
0	usr/lib/vmware/
0	usr/lib/vmware/vmkmod/
229736	usr/lib/vmware/vmkmod/ifpga
0	usr/share/hwdata/
0	usr/share/hwdata/default.pciids.d/
515	usr/share/hwdata/default.pciids.d/ifpga.ids

Support Notes

VMware certification of the Intel SR-IOV FPGA Driver is limited to the testing of SR-IOV functionality of the driver only when used with VMDirectPath IO (passthrough) and to validate VF interoperability with ESXi. Support of device PFs and associated management tools is provided by Intel

Contact Information and References:
vfpga-support@intel.com

Disclaimer: VMware is not responsible for the reliability of any data, opinions, advice, or statements made on third-party websites. Inclusion of such links does not imply that VMware endorses, recommends, or accepts any responsibility for the content of such sites.