XRT – Acceleration RunTime

Sonal Santan
Xilinx, Inc
Agenda

● XRT Architecture
  ○ Modular design
  ○ Feature full user space API

● XRT Tools
  ○ xbutil, xbmgmt, xclbinutil

● Cloud Deployment
  ○ Plugin based integration

● Security Model
  ○ Platform and XRT

● Open Source Engagement
  ○ Development on GitHub
  ○ Open documentation
XRT Architecture
XRT Modular Stack

- XRT Common API across end-point to edge to cloud
  - Support for OCL, C/C++, ML, Video, and Storage
  - Multi-process/thread safe core APIs for heavy lifting of target solution stack
  - Common API across platforms and operating systems

- Device management and loading
  - Dynamic Function Exchange (partial reconfiguration)
  - In band BMC and device management
  - Powerful command line tools

- Memory management and data synchronization
  - Memory manager for various use models
  - Integrated DMA operation
  - CPU to CPU communication

- Scheduling and Execution
  - Execution framework with multi-process support
  - Integrated debug/profile and emulation
Overall Flow Background

- Host code compiled by gcc/g++
- Host code may be in Python; XRT ships with python binding
- Device code may be in C/C++/Verilog or VHDL
- Device code compiled by Xilinx v++ compiler
- Compiled binary links with XRT and loads device binary and interacts with device using XRT APIs
XRT Kernel Space (PCIe)

IOCTLS
- reset
- bitstream
- download
- reboot
- frequency scale
- error info

SYSFS
- xclbin id
- feature rom
- sensors (temp, power)

Sub-devices
- ICAP
- XVC
- XMC
- DNA
- SYSMON
- FEATURE ROM
- FIREWALL
- MIG ECC
- HBM CTRL
- MAILBOX

APP

XRT User Space

IOCTLS
- create context
- create buffers
- map buffers
- sync buffers
- read/write buffers
- copy buffers
- exec cmd buffers
- Wait for command completion
- info

SYSFS
- xclbin id
- feature rom
- scheduler stats
- mem, ip, conn
- topology

Sub-devices
- ICAP
- XVC
- GEM
- KDS
- FEATURE ROM
- FIREWALL
- XDMA
- QDMA
- MIG
- MAILBOX

Management Tools User Space

XCLMGMT

MGMT PF

XSA

MAILBOX

USER PF

SYSFS
- xclbin id
- feature rom
- scheduler stats
- mem, ip, conn
- topology
XCLMGMT Driver (PCIe)

- **Device Identification**
  - Feature ROM
  - IDCODE
- **Device Configuration**
  - ICAP programming
  - Clock Scaling
  - Load firmware from xsabin (device tree, embedded CPU firmware)
  - FPGA_MGR integration
- **In-band Sensors**
  - Temp, Voltage, Power, etc
  - Hwmon integration
- **Interrupt Handling**
  - AXI Firewall, MailBox
- **Device Reset**
  - PCI hot reset/rescan
  - Auto reset on Device hang
- **Hardware MailBox**
  - Communication between User PF and Mgmt PF
- **Flash Programmer**
  - Update Shell Image in PROM
- **AXI Firewall**
  - Protect PCIe from AXI bus hangs/violations inside the device
  - Notify host of violations
- **Device DNA Validation**
  - DRM enforcement
XOCL Driver (PCIe)

- Memory Management
  - Abstracted buffer allocation with page pinning support
  - Memory management of dynamic device memory topologies backed by Linux DRM MM
  - DMA-BUF backed buffer import/export between processes
  - PCIe peer-to-peer buffer mapping and sharing

- MM DMA Engine (XDMA)
  - Migrate buffer between host and device memory: DDR and PL RAM

- Streaming DMA Engine (QDMA)
  - Streaming queues between host and device kernels
  - AIO Integration

- Multi-process Aware Context Management
  - Shared/Exclusive Context creation on one or more compute engines (kernels)

- Built-in Multi-process Execution Model
  - HLS Start/stop execution model
  - HLS Data Flow execution model

- Interrupt Handling
  - DMA
  - Kernel Completion
  - MailBox

- Device Usage Metrics
  - Memory, Contexts, DMA, Execution, etc.
Zocl Driver (Edge)

- Heavy lifting for MPSoC and Zynq devices
  - Accelerator support for edge devices
- Memory manager
  - CMA buffer allocation
  - Coarse-grained SVM with SMMU integration
  - Support for partitioned memory space for PCIe hybrid use case
  - On demand remap of partitioned memory buffers allocated by PCIe host into PS Linux memory space
- ERT
  - Integrated ERT for edge and PCIe hybrid use case
- FPGA Manager integration
  - Send partial bitstreams to FPGA Manager for download
- Context and execution management
  - Similar to xocl functionality
XRT User Space APIs

- Higher level domain APIs
- Interoperability between APIs
- Thread and multi-process safe
- Core API defined in `xrt.h`

```c
device = xclOpen(...);
xCllLoadXclbin(device, ...);
xCllOpenContext(device, uuid, ipIndex);

argbuf = xclAllocBO(device, flags, size,...);
argdata = xclMapBO(argbuf,...);
xCllSyncBO(device, argbuf,...);

cmdbuf = xclAllocBO(device, flags,...);
cmddata = xclMapBO(cmdbuf,...);
while (work) {
    cmd = getNextcmd();
    xclExecBuf(device, cmd);
    ++load;
}
while (load)
    while (!xclExecWait(device,...));
    --load;
}
```

XRT API

- `xclOpen`
- `xclLoadXclbin`
- `xclOpenContext`
- `xclAllocBO`
- `xclMapBO`
- `xclSyncBO`
- `xclExecBuf`
- `xclExecWait`
XRT Memory Management

● Abstracted view of memory with Buffer Objects (BO)
  ○ Non Unified Memory with PCIe devices
  ○ Unified Memory with CMA/SVM* for MPSoC
  ○ Buffers shared with PCIe host and PCIe-MPSoC hybrid devices
  ○ BOs with attributes: PCIe peer-to-peer, DMA-BUF, device only, etc.

● Advanced design
  ○ Page pinning for better performance
  ○ Integrated support for multiple contexts (process/thread)
  ○ Zerocopy (UserPtr) feature
  ○ Partitioned device memory topology for PCIe-MPSoC hybrid devices

● Addressing/Translation
  ○ Host uses virtual addressing
  ○ Typically device uses physical addressing

● Data driven Memory manager initialization and tear down
  ○ Read memory topology from xclbin

MM APIs

- xclAllocBO
- xclAllocUserPtrBO
- xclMapBO()
- xclFreeBO()
- xclSyncBO()
- mmap()
- munmap()
- xclImportBO()
- xclExportBO()
- xclCopyBO()
Asynchronous Execution Model

- Execution loop
  1. Fill the contents of command buffer
  2. Submit execution jobs to queue -- xclExecBuf()
  3. Wait for jobs to finish -- xclExecWait()
  4. Check the status of the submitted command buffers
- Support for single run (start/done) execution and data-flow (start/ready/done) execution
- Command buffer objects defined in `ert.h`
- Command buffer objects shared between user space and kernel space
- Status field in command buffer updated by XRT as command state changes

**Execution APIs**
- `xclAllocBO`
- `xclMapBO()`
- `xclOpenContext()`
- `xclExecBuf()`
- `xclExecWait()`
PCle peer-to-peer Model

- PCle peer-to-peer to transfer technology to move data between FPGA and other PCle devices
  - Expose FPGA DDR memory on PCle BAR
  - Use DMA engine of peer PCle device to directly read/write data from/to FPGA
  - Bypass host memory altogether leading to lower latency and high scalability

- XRT Features
  - Allocate buffer objects in FPGA DDR which are directly visible to host system and CPU
  - Mmap function to obtain system address for device resident buffer

- Host controlled data flow across devices
Open FPGA binary – xclbin

- ELF style sections based binary container
  - Sections for bitstream, metadata, ELF files for embedded processors, clock frequencies, etc.
  - Defines memory topology, IP instantiations, IP connectivity
  - UUID to distinguish between xclbins
  - Unique signature of target shell for compatibility check

- Well documented C-struct based design
  - xclbin.h header file

- Xclbinutil command line tool
  - Modify/Upgrade/query a xclbin binary
  - Sign an xclbin binary
  - Used by v++ to generate xclbin
XRT Tools
Command Line Tools

xbmgmt
- flash
- config
- scan

xbutil
- query
- program
- reset
- list/scan
- top
- validate
- dmatest
- m2mtest
- clock

xclbinutil
- sign
- package
- inspect
Xbutil query

INFO: Found total 1 card(s), 1 are usable

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
XRT
Version: 2.2.0
Git Hash: 4965cd43b50c47f78597c9bb5194878d2afeee90
Git Branch: master
Build Date: 2019-01-18 21:48:58

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
DSA  FPGA  IDCode
xilinx_u200_xdma_201830_2   xcu200-fsgd2104-2-e   0x14b37093
Vendor         Device      SubDevice      SubVendor
0x10ee          0x5000      0xe            0x10ee
DDR size     DDR count      Clock0      Clock1
80 GB         5            250          500
PCIE DMA chan(bidir) MIG Calibrated  P2P Enabled
GEN 2x2        2            true          false

Temperature(C)
PCB TOP FRONT   PCB TOP REAR    PCB BTM FRONT
39              32              39
FPGA TEMP       TCRIT Temp      FAN Speed(RPM)
0               0               1185

Electrical(mV|mA)
12V PEX         12V AUX         12V PEX Current 12V AUX Current
11799          11879            1217          1396
3V3 PEX         3V3 AUX         DDR VPP BOTTOM DDR VPP TOP
3338           3331            2500          2500
SYS 5V5         1V2 TOP         1V8 TOP      0V85
5485           1193            1853          851
MGT 0V9         12V SW          MGT VTT
906            11795           1199
VCCINT VOL      VCCINT CURR     DNA

Board Power
30 W

Firewall Last Error Status
Level 0 : 0x0(GOOD)

Memory Status
Tag         Type        Temp(C)  Size    Mem Usage       BO count
[ 0] bank0       MEM_DDR4    35       16 GB   0 Byte          0
[ 1] bank1       MEM_DDR4    36       16 GB   0 Byte          0
[ 2] bank2       MEM_DDR4    43       16 GB   0 Byte          0
[ 3] bank3       MEM_DDR4    38       16 GB   0 Byte          0
[ 4] PLRAM[0]    **UNUSED**  N/A      128 KB  0 Byte          0
[ 5] PLRAM[1]    **UNUSED**  N/A      128 KB  0 Byte          0
[ 6] PLRAM[2]    **UNUSED**  N/A      128 KB  0 Byte          0

DMA Transfer Metrics
Chan[0].h2c:  54144 MB
Chan[0].c2h:  22784 MB
Chan[1].h2c:  17280 MB
Chan[1].c2h:  22784 MB

Xclbin UUID
8518bdfc-3931-4e95-b4d1-7bc2380be3be

Compute Unit Status
CU[ 0]: bandwidth1:kernel_1 @0x1c000000 (IDLE)
CU[ 1]: bandwidth2:kernel_2 @0x18000000 (IDLE)

INFO: xbutil query succeeded.

dx4300:~>
XRT Cloud Support
Streamlined Cloud Support

- Virtualization
  - Management PF hidden in Host
  - PCIe pass-through of User PF to Guest
- Common code base across all cloud vendors
  - MailBox for control messages
  - Mgmt Interfaces abstracted by Management Proxy Daemon (MPD) and Management Service Daemon (MSD)
- MPD and MSD are systemd managed daemons
  - Load vendor specific plugin

Sample flow for xclbin download in cloud runtime environment
XRT Security Model
Security – Platform Topology

Security – Platform

● PCIe Bus Safety/Security
  ○ On boot shell is loaded from PROM and can be optionally encrypted
  ○ Shell prevents user compiled binary from accessing PCIe bus
  ○ AXI Firewall in data paths in the shell protects the shell itself from errant role

● Two Physical Function Architecture
  ○ Mgmt Physical Function responsible for trusted operations
  ○ User Physical Function responsible for end-user operations
  ○ Trusted components in the Shell can only be accessed by Mgmt Physical Function
Security – Software

● OS Level Security
  ○ XRT RPM/DEB install packages are signed by Xilinx
  ○ XRT drivers support UEFI secure boot with DKMS signed drivers.
  ○ XRT supports loading of signed xclbin with authentication using installed certificates
  ○ XRT code is continuously validated by Coverity static code analyzer for any security vulnerability

● Role authentication/Ingestion
  ○ xclbins can be signed by administrator in a secure environment
  ○ FaaS infrastructure allows xclbin downloads only from secure host and not from untrusted VM

● Multi-tenancy
  ○ Multi-process is supported between trusted applications which share the same xclbin
  ○ One role cannot be shared between multiple VMs
Open-source Development
XRT Development and Delivery

● Open development on GitHub
  ○ CMake based build system
  ○ Auto-generation of documentation using Sphinx
  ○ Agile development model
  ○ 72K LOC Linux driver
  ○ 127K LOC userspace

● Distributed as versioned RPM/DEB packages

● Accepting pull requests from the community

● Seeking to upstream XRT drivers

● Keen to collaborate with Linux distros

https://github.com/Xilinx/XRT
https://xilinx.github.io/XRT
mailto:runtime@xilinx.com
Integration With Linux Standards/Components

● Linux Kernel
  ○ Uses FPGA_MGR, DRM MM, HWMON, DMA_BUF, etc.
  ○ Extensive sysfs node hierarchy
  ○ Modular platform driver architecture; Device Tree based Shell/platform peripheral discovery

● Systemd integrated daemons

● Build And Delivery
  ○ Cmake based build system; DKMS integration
  ○ RPM/DEB deployment packages
  ○ Sphinx-doc and reStructuredText based documentation system
  ○ Yocto recipes for edge platforms

● Development Model
  ○ Open GitHub based development model
  ○ Integration with Travis CI
Summary

- XRT is modular acceleration architecture which supports end-point to edge to data center platforms
- XRT abstracts many advanced and complex technologies into simple API calls
- XRT is open source and seeking to build a vibrant community around the technology
Thank you

Join Linaro to accelerate deployment of your Arm-based solutions through collaboration

contactus@linaro.org