Accelerating Compute with FPGAs
based Xilinx accelerator cards

Xinping Deng

10 September 2020, CASS Co-learnium
Kind of introduction ...
CPU, GPU and FPGA

A **central processing unit (CPU)** is the **electronic circuitry** within a **computer** that executes **instructions** that make up a **computer program**. It is not designed to process data in parallel.

A **graphics processing unit (GPU)** has highly **parallel structure** so that they are more efficient than general-purpose CPUs for **algorithms** that process large blocks of data in parallel. It is a popular hardware to accelerate compute, but its power consumption is high and high-end GPUs are extremely expensive.

A **Field Programmable Gate Array (FPGA)** is an **integrated circuit** designed to be configured by a customer or a designer after manufacturing. Its power consumption is low, but it is hard to program.
Why in CASS do we use FPGAs?

• They are mostly used in the telescope observing systems.
  • For example, digitizer, beamformer and correlator of ASKAP Phased Array Feed receiver
The problem with traditional FPGAs

• Need to be a hardware expert to program it.
• Need to arrange accessories connection
  ➢ either buy it or build it yourself.
Xilinx

• **Xilinx, Inc.** (ˈzaɪlɪŋks/ ZY-links) is an American technology company that develops highly flexible and adaptive processing platforms. The company invented the field-programmable gate array (FPGA).
What are Xilinx Accelerator Cards?

• Build on Xilinx FPGAs, for general developers;
• The first one was released at 16\textsuperscript{th} Oct. 2018;
• It is not as popular as GPU yet;

• High performance (comparing with GPU):
  ➢ More flexible on-chip memory;
  ➢ Higher internal bandwidth;
  ➢ Lower power consumption;
# Popular Xilinx Accelerator Cards

<table>
<thead>
<tr>
<th>Feature</th>
<th>Alveo U200</th>
<th>Alveo U250</th>
<th>Alveo U280</th>
<th>Alveo U50</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Width</td>
<td>Dual Slot</td>
<td>Dual Slot</td>
<td>Dual Slot</td>
<td>Single Slot</td>
</tr>
<tr>
<td>Form Factor, Passive</td>
<td>Full Height, ¾ Length</td>
<td>Full Height, ¾ Length</td>
<td>Full Height, ¾ Length</td>
<td>Half Height, ¼ Length</td>
</tr>
<tr>
<td>Form Factor, Active</td>
<td>Full Height, Full Length</td>
<td>Full Height, Full Length</td>
<td>Full Height, Full Length</td>
<td>Full Height, Full Length</td>
</tr>
<tr>
<td>Look-Up Tables</td>
<td>1,182K</td>
<td>1,728K</td>
<td>1,304K</td>
<td>872K</td>
</tr>
<tr>
<td>Registers</td>
<td>2,364K</td>
<td>3,456K</td>
<td>2,607K</td>
<td>1,743K</td>
</tr>
<tr>
<td>DSP Slices</td>
<td>6,840</td>
<td>12,288</td>
<td>9,024</td>
<td>5,952</td>
</tr>
<tr>
<td><strong>Logic Resources</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DDR Format</td>
<td>4x 16GB 72b DIMM DDR4</td>
<td>4x 16GB 72b DIMM DDR4</td>
<td>2x 16GB 72b DIMM DDR4</td>
<td>–</td>
</tr>
<tr>
<td>DDR Total Capacity</td>
<td>64GB</td>
<td>64GB</td>
<td>32GB</td>
<td>–</td>
</tr>
<tr>
<td>DDR Max Data Rate</td>
<td>2400MT/s</td>
<td>2400MT/s</td>
<td>2400MT/s</td>
<td>–</td>
</tr>
<tr>
<td>DDR Total Bandwidth</td>
<td>77GB/s</td>
<td>77GB/s</td>
<td>38GB/s</td>
<td>–</td>
</tr>
<tr>
<td>HBM2 Total Capacity</td>
<td>–</td>
<td>–</td>
<td>8GB</td>
<td>8GB</td>
</tr>
<tr>
<td>HBM2 Total Bandwidth</td>
<td>–</td>
<td>–</td>
<td>460GB/s</td>
<td>316GB/s/4</td>
</tr>
<tr>
<td><strong>Internal SRAM</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total Capacity</td>
<td>43MB</td>
<td>57MB</td>
<td>43MB</td>
<td>28MB</td>
</tr>
<tr>
<td>Total Bandwidth</td>
<td>37TB/s</td>
<td>47TB/s</td>
<td>35TB/s</td>
<td>24TB/s</td>
</tr>
<tr>
<td><strong>PCI Express®</strong></td>
<td>Gen3 x16</td>
<td>Gen3 x16</td>
<td>Gen3 x16, 2xGen4 x8, CCIX</td>
<td>Gen3 x16, 2xGen4 x8, CCIX</td>
</tr>
<tr>
<td>Network Interface</td>
<td>2x QSFP28</td>
<td>2x QSFP28</td>
<td>2x QSFP28</td>
<td>U50 2 - 1x QSFP28</td>
</tr>
<tr>
<td><strong>Thermal Cooling</strong></td>
<td>Passive, Active</td>
<td>Passive, Active</td>
<td>Passive, Active</td>
<td>Passive</td>
</tr>
<tr>
<td>Typical Power</td>
<td>100W</td>
<td>110W</td>
<td>100W</td>
<td>50W</td>
</tr>
<tr>
<td>Maximum Power</td>
<td>225W</td>
<td>225W</td>
<td>225W</td>
<td>75W</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clock Precision</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>IEEE Std 1588</td>
</tr>
<tr>
<td><strong>Vitis™ Developer</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Environment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

*Notes: U50 DD - Passive, U50 DD - 2x SFP-DD.*
# Comparing two Xilinx Accelerator cards with GPUs

## BENCHMARKS

<table>
<thead>
<tr>
<th>AREA</th>
<th>PARTNER WORKLOAD</th>
<th>ALVEO ACCELERATION VS CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database Search and Analytics</td>
<td>BlackLynx Unstructured Data Elasticsearch</td>
<td>90X</td>
</tr>
<tr>
<td>Financial Computing</td>
<td>Maxeler Value-at-Risk (VAR) Calculation</td>
<td>89X</td>
</tr>
<tr>
<td>Machine Learning</td>
<td>Xilinx Real-Time Machine Learning Inference</td>
<td>20X</td>
</tr>
<tr>
<td>Video Processing / Transcoding</td>
<td>NGCodec HEVC Video Encoding</td>
<td>12X</td>
</tr>
<tr>
<td>Genomics</td>
<td>Falcon Computing Genome Sequencing</td>
<td>10X</td>
</tr>
</tbody>
</table>

CPU Comparisons: Xeon c4.8xlarge AWS | Xeon E5-2643 vs4 3.4GHz | Xeon Platinum c5.18xlarge AWS | Dual Socket E5-2680 v3 2.5GHz | Xeon f1.8xlarge

### Increase Real-Time Machine Learning* Throughput by 20X

![Image showing throughput comparison between CPU, GPU, and CPU+Alveo]

### Reduce ML Inference Latency by 3X

![Image showing latency comparison between CPU, GPU, and CPU+Alveo]

*GoogleNet V1: Accelerating DNNs with Xilinx Alveo Accelerator Cards White Paper

CPU+GPU: Nvidia P4 + Xeon CPU E5-2690 v4 @2.60GHz (64 Cores)
CPU+Alveo: Alveo U200 or U250 + Xeon CPU E5-2686 v4 @2.3GHz (8 Cores)
Who is using these new cards?

Built for Any Server, Any Cloud

Deploy anywhere – from the private data center to the public cloud

On-Premise

Dell EMC

Inspur

Cloud

AWS

Alibaba Cloud

Baidu Cloud

HUAWEI

Nimbix

Tencent Cloud
Why is CASS interested in the new card?

- It is easier to program:
  - No hardware description language required
- It is ready to use:
  - Plug it into a computer and use it.
Why I am interested in this card?

• GPU’s on-chip memory is not flexible to run ASKAP coherent fast-radio-burst detection pipeline;
• Get experience on the new card to see if we can develop telescope signal processing system with it in future.
How to develop with it?
Development methodology

• Like GPU approach:
  - Host does memory management, communication control, etc.
  - Kernel does accelerations;
  - PCIe for data transfer;
• Host in C/C++/OpenCL;
• Kernel in RTL, C/C++ or OpenCL C;
  - RTL is unnecessary for kernel programming;
  - It is an option kept for traditional FPGA developers;
• General libraries provided;

Programming languages
**Development iteration**

- **Software emulation:** kernel runs on CPU, takes couple of minutes to compile and execute;
- **Hardware emulation:** kernel runs on CPU, but emulation the execution on hardware, takes about ten minutes to compile, couple of hours to execute;
- **Hardware execution:** kernel runs on FPGA, takes couple of hours to compile, take seconds to execute;

<table>
<thead>
<tr>
<th>Software Emulation</th>
<th>Hardware Emulation</th>
<th>Hardware Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host application runs with a C/C++ or OpenCL model of the kernels.</td>
<td>Host application runs with a simulated RTL model of the kernels.</td>
<td>Host application runs with actual hardware implementation of the kernels.</td>
</tr>
<tr>
<td>Used to confirm functional correctness of the system.</td>
<td>Test the host / kernel integration, get performance estimates.</td>
<td>Confirm that the system runs correctly and with desired performance.</td>
</tr>
<tr>
<td>Fastest build time supports quick design iterations.</td>
<td>Best debug capabilities, moderate compilation time with increased visibility of the kernels.</td>
<td>Final FPGA implementation, long build time with accurate (actual) performance results.</td>
</tr>
</tbody>
</table>
Acceleration with FPGA is easy
#PRAGMA HLS XXX
is all you need to know ...
Instruction level parallelism

void F (int A[2], int Z[2]) {
    # PRAGMA HLS PIPELINE
    add: for (i=1; i<=2; i++) {
        Z[i] = A[i] + 10;
    }
}

PRAGMA provides additional information to the compiler

A slide for experts!

One iteration at a time without pragma

Standard C code

Func Pipelining

Start the next iteration before finishing the previous one!
Instruction level parallelism

void F (int A[2], int Z[2]) {
    # PRAGMA HLS PIPELINE
    add: for (i=1; i<=2; i++) {
        Z[i] = A[i] + 10;
    }
}

default

PIPELINE

void F (...) {
    ...
    add: for (i=0; i<=3; i++) {
        #pragma HLS UNROLL
        b = a[i] + c;
    ...

Default: 4 cycles

Unroll: 1 cycle

A slide for experts!
Task-Level Parallelism

for (int i=0; i<N; i++)
{
    #pragma HLS DATAFLOW
    func1();
    func2();
    func3();
}

> Create custom dataflow pipelines
> Multiple tasks executing simultaneously
> Streaming programming paradigm
Sometimes acceleration may not be that easy

Not going to detail here...
ASKAP coherent FRB detection pipeline

- Developing a pipeline with Xilinx accelerator card U280
- Expect that this will improve ASKAP FRB detection sensitivity by a factor of 10

Simulated FRB candidate coming through the pipeline
FPGAs vs GPUs for astronomy data processing

• Should future systems be FPGAs or GPUs based?
• With experience in both GPU and FPGA programming and my thoughts are:
  Ø GPUs are relatively easy to program and faster to iterate your code;
  Ø New Xilinx system much easier than previous FPGAs, but still in development;
  Ø It is worth to follow and use these developments as:
    ✓ It is a more cost-efficient solution (both power consumption and price is lower than GPUs);
    ✓ Its flexible on-chip memory and integrated network interface are very attractive;
Want to try it out?

• Xilinx Nimble Cloud has Xilinx Vitis and accelerator card installed at [link](https://www.xilinx.com/xilinxtraining/assessments/portal/alveo/intro_nimbix_cloud/story.html), free trial is available.

• Come and chat with me. xinping.deng@csiro.au, room 84.
Thank you

Astronomy and Space Science
Xinpeng Deng
Research Engineer
E  xinpeng.deng@csiro.au