OpenCL Streaming Platform on FPGA

Armin Weiss / Matthias Rosenthal
Institute of Embedded Systems
High Performance Multimedia Group
Zurich University of Applied Sciences
Agenda

- OpenCL on FPGA
- Parameter Update
- Measurement Results
Processing of Streaming Data

- Audio
- Video
- Sensor Data

Signal Processing
Fast and simple way to program the signal processing
What is OpenCL?

- Initially developed by Apple in collaboration with AMD, IBM, Intel and Nvidia
- OpenCL is very similar to C
- Basic Idea: Use GPU power for non-graphic applications
OpenCL on various Architectures

SIMD (Single Instruction Multiple Data)
# Parallelism by Pipelining

## Clock Cycle

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

## Kernel Threads

- A
- B
- C
- D
OpenCL on FPGA: Addition Example

```
__kernel void sum(__global const float *a,
                  __global const float *b,
                  __global float *answer)
{
    int gid = get_global_id(0);  // Thread ID
    answer[gid] = a[gid] + b[gid];
}
```

8 threads for vector add example

Thread IDs

OpenCL kernel used as hardware accelerator

```c
__kernel void sum(__global const float *a,
                  __global const float *b,
                  __global float *answer)
{
    int gid = get_global_id(0);    // Thread ID
    answer[gid] = a[gid] + b[gid];
}
```
Streaming Data Flow

Hardware Accelerator Use:

Global Memory

Kernel 1
Read
Write

Kernel 2
Read
Write

Kernel 3
Read
Write

Kernel 4
Read
Write


Streaming Application Use:

Global Memory

Input
Read
Kernel
Write
Output
Color Space Converter Reference Design

DE1-SoC

Cyclone V

HPS
Linux (Yocto)

Parameter

Test Pattern Generator

OpenCL Color Space Converter

VGA Controller

Video

Monitor
1. *Method: Shared Buffer*

- Global memory interconnect requires FPGA logic and memory
- Memory access may stall
- OpenCL v1.2 does not define shared memories
2. Method: Update Interface

- No stalls
- Requires limited local FPGA memory
Total System Resource Usage

- Resource usage is independent from the resolution
- Higher resolutions have a higher occupancy

<table>
<thead>
<tr>
<th>Resolution</th>
<th>640x480</th>
<th>800x600</th>
<th>1280x1024</th>
<th>1706x1365 (Theoretical)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALUTs</td>
<td>9089</td>
<td>9091</td>
<td>9087</td>
<td>-</td>
</tr>
<tr>
<td>M10K Memory Blocks</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>-</td>
</tr>
<tr>
<td>Occupancy</td>
<td>13.1% (44.2 MB/s)</td>
<td>20.7% (68.7 MB/s)</td>
<td>56.3% (189.1 MB/s)</td>
<td>100% (335.9 MB/s)</td>
</tr>
</tbody>
</table>
Kernel Duplication

FPGA

Huge Image Source → Top → Bottom → OpenCL Kernel

OpenCL Kernel
Conclusion

- OpenCL on FPGA Architecture
  - No cores but pipelines

- New Framework for Streaming Applications
  - For streaming data (Video, Audio, Sensor-Data)
  - FPGA based solution without CPU processing
  - New method for parameter updates
  - Measurements

- Reference Design available from ZHAW
Thank you for your attention!

Web:  http://www.zhaw.ch/ines  
Blog:  https://blog.zhaw.ch/high-performance/  
Github:  https://github.com/InES-HPMM  
E-Mail:  Armin Weiss (weii@zhaw.ch), Matthias Rosenthal (rosn@zhaw.ch)