IvyTown Xeon + FPGA: The HARP Program

Xeon+FPGA tutorial @ ISCA 2016

David Sheffield / Intel Labs/ david.b.sheffield@intel.com
Legal Disclaimer

Copyright (C) 2008-2016 Intel Corporation All Rights Reserved.

The source code contained or described herein and all documents related to the source code ("Material") are owned by Intel Corporation or its suppliers or licensors. Title to the Material remains with Intel Corporation or its suppliers and licensors. The Material contains trade secrets and proprietary and confidential information of Intel or its suppliers and licensors. The Material is protected by worldwide copyright and trade secret laws and treaty provisions. No part of the Material may be copied, reproduced, modified, published, uploaded, posted, transmitted, distributed, or disclosed in any way without Intel's prior express written permission.
Prototype Xeon+FPGA* system disclaimer

This talk is about prototype hardware and software which has been made available to universities in the HARP program.

Details of production Xeon+FPGA systems will be made available at a later date.

Results and details in this presentation were generated using pre-production hardware and software, and may not reflect production or future systems.

*System described today is officially known as the “Intel QuickAssist QPI FPGA Platform”
IvyTown Xeon + Stratix V FPGA

Accelerating Workloads using Xeon and coherently attached FPGA in-socket

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Intel® Xeon® E5-26xx v2 Processor</td>
</tr>
<tr>
<td>FPGA Module</td>
<td>Altera Stratix V</td>
</tr>
<tr>
<td>QPI Speed</td>
<td>6.4 GT/s full width</td>
</tr>
<tr>
<td>Memory to FPGA Module</td>
<td>2 channels of DDR3 (not used on HARP platform)</td>
</tr>
<tr>
<td>Features</td>
<td>Configuration Agent, Caching Agent, (optional) Memory Controller</td>
</tr>
<tr>
<td>Software</td>
<td>Accelerator Abstraction Layer (AAL) runtime, drivers, sample applications</td>
</tr>
</tbody>
</table>
Anatomy of an IvyTown Xeon+FPGA solution

User is free to program in any language with C++ bindings

AAL provides C++ API for FPGA resource management

Linux kernel module manages FPGA page table

10-core Intel Xeon E5-2680 v2

Intel S2600CP motherboard with custom BIOS

Intel cache coherent interconnect

Virtual memory support and transaction reordering

Intel provides hardware template, user defined accelerator RTL (assuming HDL flow)

User logic (FPGA-based accelerator)

Color key

User defined

Intel provided
IvyTown Xeon + FPGA tool flow

HDL Programming

OpenCL™ Programming

- C/C++
- AAL SDK / compiler
- Verilog
- Quartus
- bit-stream

- Intel® Xeon®
- FPGA

- OpenCL host
- Compiler / Run-time
- OpenCL kernels
- OpenCL for FPGA
- exe
- bit-stream
Agenda

• An overview of the IvyTown + FPGA system
  • Hardware details
• HARP program overview
• Current status
• What’s next?
HARP System – IvyTown Xeon + Stratix V FPGA

Standard 2U server – FPGA module is the only custom hardware!
HARP FPGA module

Top -- StratixV FPGA (5SGXEA7N1F45C1)

Bottom – Socket R (LGA 2011)
System Logical View

Accelerator Function Units (AFUs) can access coherent cache on FPGA
- AFU written by end-user

Intel® Quick Path Interconnect (Intel® QPI) IP participates in cache coherency with Xeon CPUs
IvyTown Xeon + FPGA microarchitecture

**PHY** – Implements the Intel QPI PHY 1.1 (Analog/Digital)

**Intel QPI Link layer** - provides flow control and reliable communication

**Intel QPI Protocol** – implements Intel QPI Cache Agent + Configuration Agent

**Cache Controller** – Cache hit/miss determination and generates Intel QPI protocol requests.

**Cache Tag** – Tracks state of cacheline (MESI + internal states for tracking outstanding requests)

**Coherency Table** – Programmable table that implements coherency protocol rules

**System Protocol Layer (SPL2)** – Implements Address translation functionality. Can provide up to 2GB device virtual address space to AFU. SPL2 cannot handle page faults.

**AFU** – User designed Accelerator Function Unit
SPL2/AFU2 details

Xeon and FPGA can share pointer-based data-structures with CCI-E
# CCI-S vs CCI-E

<table>
<thead>
<tr>
<th>Capability</th>
<th>CCI-S</th>
<th>CCI-E with SPL2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Req Header/Address width</td>
<td>32-bits</td>
<td>58-bits</td>
</tr>
<tr>
<td>Address granularity</td>
<td>64B</td>
<td>64B</td>
</tr>
<tr>
<td>Addressing Mode</td>
<td>Physical Addressing</td>
<td>Virtual Addressing</td>
</tr>
<tr>
<td>Maximum workspace size</td>
<td>Function of OS. Upto 4MB per workspace possible.</td>
<td>2GB workspace</td>
</tr>
<tr>
<td># of workspaces</td>
<td>&gt;1</td>
<td>1 (current restriction)</td>
</tr>
<tr>
<td>Read response reorder buffer</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Write ordering</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>
Cache protocol flows

For IvyTown + FPGA latency and bandwidth statistics, see academic literature:
AFU Interface

- 2 TX channels:
  - Ch0 for Memory Read
  - Ch1 for Memory Write
- Independent flow control of each Tx channel. Will accept up to 4 Requests after Almost Full asserted
- MData field: to identify requests and responses
- No flow control on Rx channel: AFU must provision Rx buffer before request is sent
- SoftReset_n must clear all AFU state

End-user develops AFUs
Simulation environment - ASE

CPU
- Host Application
- Service API
- Virtual Memory API
- Physical Memory API
- QPI Link, Protocol, & PHY

FPGA
- Accelerator Function Units (AFU)
- Add Translation & Response re-ordering
- QPI Link, Protocol, & PHY

- QPI Link, Protocol, & PHY

HW deployment

ASE simulation environment

True HW/SW co-development environment for Intel Xeon+FPGA
Demo – Sudoku solver on IvyTown + FPGA

• Took Sudoku solver RTL from OpenCores and added SPL interface
  • http://opencores.org/project,sudoku
• Simple application built in C++ with AAL
  • Optimized Xeon CPU implementation
Demo
Agenda

• An overview of the IvyTown + FPGA system
• HARP program overview
• Current status
• What’s next?
Recap from HW overview: Xeon+FPGA unique features

FPGA with coherent low-latency interconnect:

Simplified programming model
- Support for virtual addressing
- Data caching on FPGA

Enables new classes of algorithms for acceleration with:
- Full access to system memory
- Support for efficient irregular data pattern access

Remapping of algorithms from off-load model to hybrid (CPU+FPGA) processing model
- Fine grained interactions
Intel Xeon+FPGA academic program goals

Win academic mindshare by providing

1. Early access to prototype Xeon+FPGA systems
2. Generate community around Xeon+FPGA
3. Technology leadership through world-class FPGA research

Nurture early users to demonstrate value and accelerate future Xeon+FPGA adoption
HARP - call for proposals (early 2015)

Intel® Altera Heterogeneous Architecture Research Platform (HARP) Program

Intel® Corporation and Altera® Corporation are pleased to announce the Heterogeneous Architecture Research Platform (HARP) program, which will provide faculty with computer systems containing Intel microprocessors and an Altera Stratix® V FPGA module that incorporates Intel® QuickAssist Technology in order to spur research in programming tools, operating systems, and innovative applications for accelerator-based computing systems.

http://www.sigarch.org/2015/01/17/call-for-proposals-intel-altera-heterogeneous-architecture-research-platform-program/
HARP program – areas of interest

- OS and middleware research
- Architecture research
- Programming systems research
- Application research
- Algorithm research
HARP call for proposals generated buzz

“The project is called HARP, short for Heterogeneous Architecture Research Platform. Mars and his team are applying to receive and test one of these boards. Although the world is moving toward thing like FPGAs, traditional server chips will continue to be used as well, and these boards explore ways of marrying the two. “It’s a very good move for Intel” Mars says.”

http://www.wired.com/2015/03/intel-exploring-biggest-takeover-ever/
Awardees & Research Areas

Received 100+ proposals, awarded 25 systems

- Application evaluation - 16 proposals
- Architectural tools and frameworks – 8 proposals
- Programming tools and methodology – 6 proposals
- OS and networking techniques – 3 proposals

More accepted proposals than awarded systems (some research groups sharing a system)
HARP program – history

Call for proposals early 2015
Notified awardees mid spring 2015
Full day tutorial @ Intel in June of 2015
• Intel provided full day of tutorials and hands-on activities
• 50 attendees
Systems shipped to universities mid-summer 2015
Agenda

• HARP program overview
• Current status
• What’s next?
Winning Academic Mindshare with HARP

1. Academics are focusing on novel hybrid CPU—FPGA use cases
   - Before: what can I offload to FPGA?
   - Now: what’s CPU great at? what’s FPGA great at? how to collaborate?
   - E.g., Genomics, Database, Graph/irregular, Sort

2. Academics are rethinking hybrid CPU – FPGA systems
   - FPGA is becoming 1st class citizen, tighter integration to CPU
   - What technologies needed to best take advantage of hybrid CPU-FPGA systems?
   - E.g., JIT to FPGA, SPARK cloud + FPGA, OpenMP for FPGA
Winning Academic Mindshare with HARP (cont.)

3. Academics are publishing on top FPGA conferences using Xeon-FPGA
   - ISFPGA (Feb 2016): 1 out of 20 full papers use HARP
   - FCCM (May 2016): 2 out of 18 full papers use HARP
Publications using HARP platform

UCLA
- “The SMEM Seeding Acceleration for DNA Sequence Alignment” – FCCM16
- “A Quantitative Analysis on Microarchitectures of Modern CPU-FPGA Platforms” – DAC16

CMU
- “A Study of Pointer-Chasing Performance on Shared-Memory Processor-FPGA Systems” – ISFPGA16

ETHZ
- “Runtime Parameterizable Regular Expression Operators for Databases” – FCCM16

USC
- “High Throughput Large Scale Sorting on a CPU-FPGA Heterogeneous Platform” – RAW 2016

Even more papers in submission!
Agenda

- HARP program overview
- Current status and outcomes
- What's next?
Where do we go with HARP?

“While at the ACM FPGA Conference in Monterey, we met with several students and professors who do not have access to the program who were just generally interested in how things were going.” – USC feedback

We continuously get email from well-known researchers asking if systems are still available

We want to grow the HARP program!
Where to go with HARP -- thoughts

1. Increase productivity of existing HARP community
   - Share apps, tools, and IP repository
   - Regular online symposiums / seminars. Workshop at Intel
   - Cloud Xeon-FPGA setup to put systems/tools within reach

2. Grow community by attracting more experts to join
   - Tutorial / workshops @ premier conferences
     - Please contact me if you’re interested in participating
Questions

Thank you for attending!

If you are interested in getting access to the SDK/HDK, please email me @ david.b.sheffield@intel.com