CLICK-TO-NETFPGA

TEEMU RINTA-AHO (TEEMU.RINTA-AHO@ERICSSON.COM)
NOMADICLAB, ERICSSON RESEARCH FINLAND

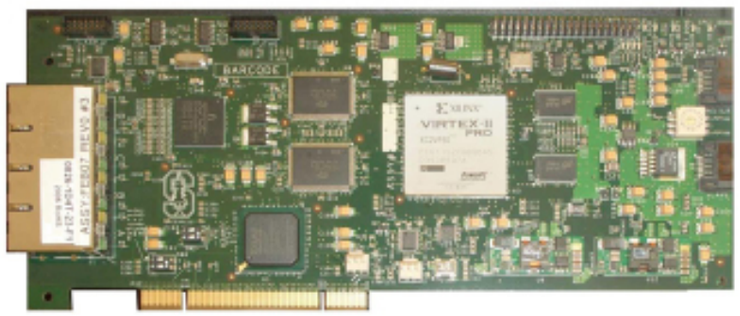
FI PIZZA, AALTO UNIVERSITY
DECEMBER 7, 2011

CLICK-TO-NETFPGA

- We have explored the possibilities of High Level Synthesis (HLS) in packet processing domain
  - –HLS means transformation of software into hardware
- Using open source components, we have created a prototype toolchain that allows
  - –Defining Click configurations using existing and new elements,
  - –Writing new Click elements in C++, and
  - –Compiling them to hardware, to be run on NetFPGA
- The main components are Click, NetFPGA, LLVM and AHIR

STANFORD NETFPGA

- A PCI network interface card with an FPGA
  - –4 x 1G Ethernet interface
- Line-rate, flexible, and open platform
- For research and classrooms



- More than 1,000 NetFPGA systems deployed
- A few open-source, Verilog-based reference designs
- A newer, NetFPGA 10G card is also available
  - –4 x 10G Ethernet interface, bigger FPGA, faster PCI interface

LOW LEVEL VIRTUAL MACHINE (LLVM)

- Open source, from UIUC
- Set of tools and optimizers
- Easy to write new compiler passes
- Easy to write backends (and, maybe, frontends)
- Represents code as SSA (Single Static Assignment)
  - An abstract, assembler-like form, with unlimited registers
- Outperforms GCC in many (but not all) ways
- Can perform global optimizations (after linking)

AHIR
"A HARDWARE INTERMEDIATE REPRESENTATION"

- LLVM backend for generating VHDL
  - To-be open source, by IIT Bombay (India)
  - Factorises the system into control, data, and storage
    - Supports scalable optimisations and analyses
  - Current limitations: no recursion or function pointers, otherwise full C
  - Generates a VHDL module out of each LLVM IR function
- Design = Set of modules with I/O channels
  - I/O through a simple VHDL "library", resembling Unix pipes

Click NetFPGA target library

Click element source

llvm-g++

Click element IR

llvm-g++

Click element object

llvm-link

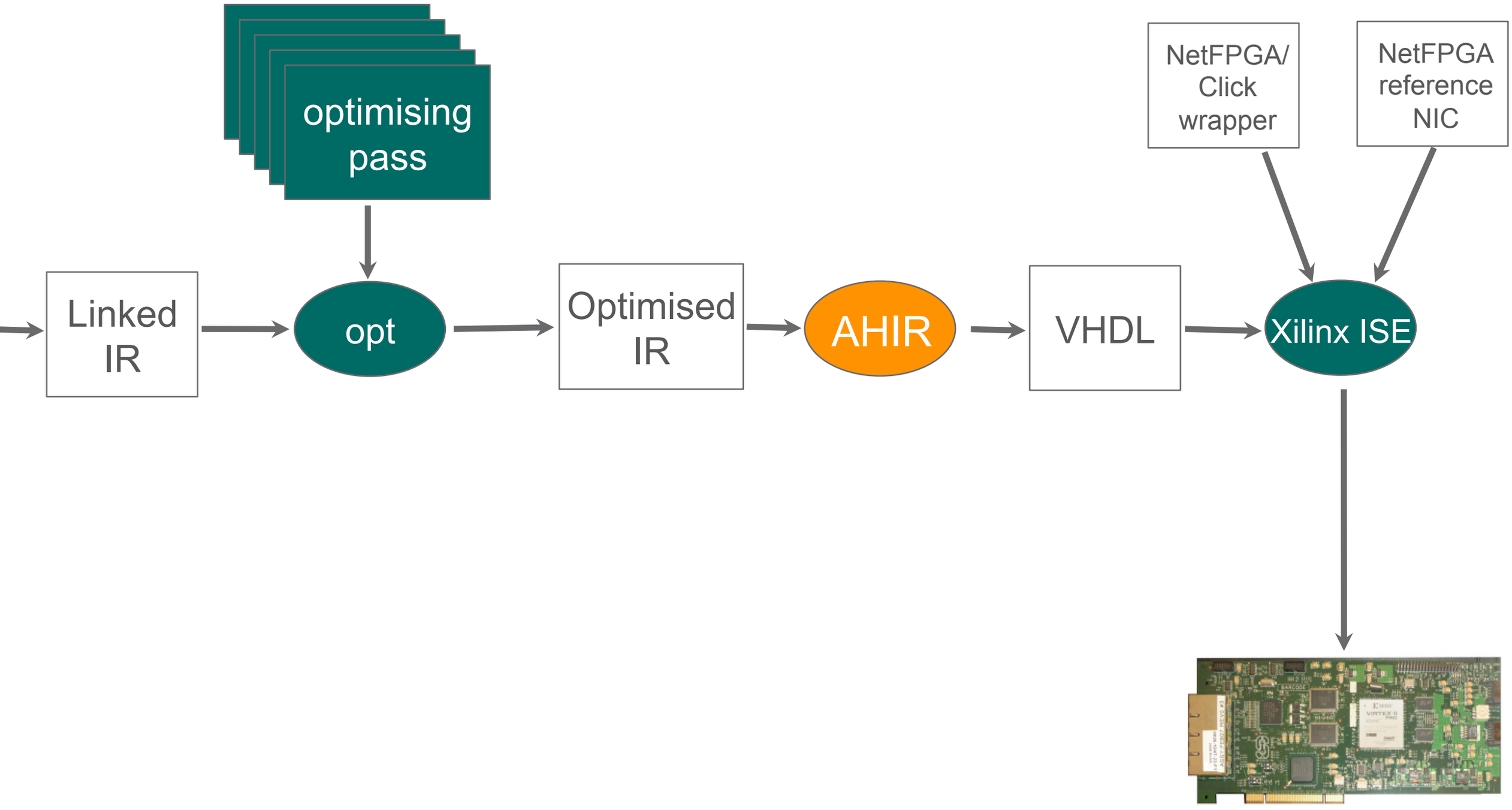
Linked IR

Click host target library

Click2-LLVM

Click config IR

Click config source


optimising pass
Linked IR
opt
Optimised IR
AHIR
VHDL
NetFPGA/ Click wrapper
NetFPGA reference NIC
Xilinx ISE

NetFPGA reference NIC design

User module (modified `user_data_path.v + netfpga_module.vhdl + ahir_system.vhdl`**)**

Generated part (click2llvm + AHIR generated `ahir_system.vhdl`**)**

Control path

Data path

wrapper _input

FromFPGA *Click element* (name: src)

CheckIPHeader *Click element* (name: chk)

IPLookup *Click element* (name: rtt)

ToFPGA0 *Click element* (name: to0)

ToFPGA1 *Click element* (name: to1)

wrapper _output

Control path

Data path

W R W R R R R R R

fromfpga_in0 chk_in0 rtt_in0 to0_out0 tofpga_out0 to1_out0

free_queue

pkt1 pkt4 pkt6

BRAM for packets

Memory subsystem

NOTE: Click elements are running in parallel, each processing a different packet, while software Click operates in run-to-completion mode.

CONCLUSION

- Writing a toolchain that transforms a complex software system into a hardware system *is possible*
  - –However, more research is required to create a toolchain that creates hardware which runs *faster* than the original software *with all the software features* (such as reconfigurability)
- Our results should encourage further research on the identified problems/subtopics
  - –We will submit a paper on the results so far to USENIX 2012