REMOTE PROCEDURE CALL COMPILER FOR FIELD PROGRAMMABLE GATE ARRAYS

Abstract
RPC-FPGA implements the Open Network Computing (ONC-RPC) remote procedure call protocol specification for use in FPGA accelerators. The compiler generates a High-Level Synthesis (HLS) interface to call a hardware procedure and to stream the data between the processor and the FPGA. The major benefits of RPC-FPGA are:

- hardware procedures running on an FPGA accelerator become accessible for any client in the network,
- the development time of the communication interface between the processor and the FPGA is greatly reduced.

Using RPC-FPGA a speedup of 17 to 20 is demonstrated on a square matrix multiplication of N=4096 with network speeds 100 Mbps and 1 Gbps respectively.

Motivation
High-Level Synthesis (HLS) languages define the interface between Host and FPGA using proprietary pragmas (Vivado HLS), language constructs (OpenCL) or message passing (MPI). Some limitations of current host/FPGA interfaces are:

- the hw/sw interface has to be defined and managed by the user
- DMA data transfer requires explicit coding
- data type serialization support is limited
- interfacing leads to substantial programming overhead

Methodology
The interface is defined in a protocol description file (.ad) consisting of the arguments declaration and the procedure prototype, following the ONC-RPC syntax. Example see the protocol description of a dot product.

RPC-Model
We use a 2-stage Remote Procedure Call:

1. 1st stage: connect network client and server via ONC-RPC
2. 2nd stage: connect server with attached FPGA via RPC-FPGA

RPC extension to multiple dimensions
HLS optimizations operate on multidimensional arrays.
ONC-RPC supports only one-dimensional arrays.

RPC protocol file
RPCGEN generates
interface definition
Declarations
Stream input
Calculate
Stream output

Generated HLS hw stub for the programmable logic (PL)

Server code linking network to FPGA

- service procedure on the server calls the hardware procedure on behalf of the ONC-RPC client
- (IO) arguments are sent as unit 32 or 64-bit words from DDR (PS) to BLOCKRAM (PL) using DMA.

RPCCGEN handling of variable length multidimensional arrays

Results
- Ease study: dense N x N square matrix multiplication
- Zynq 7020 SoC Zedboard ARM (PS) + FPGA (PL)

PL performance
- Available BlockRAM limits matrix size to N=128
- Maximum from HLS report: 5.08 GFLOPS (only PL)
- Measured on Petalinux: 2.09 GFLOPS (includes driver overhead)

Block matrix computation
PS+PL algorithm
- Increases matrix size up to N=4096 on Zedboard
- Block matrix multiply on PL (blocks of 50x50)
- Summation of blocks on PS

Speedup comparison
1. PS+PL: execution on ARM+FPGA under Petalinux
2. RPC+PS+PL: execution on network using RPC call
3. Local I7 : execution on Intel(R) Core(TM) i7-2600K processor

Performance analysis
PL = block matrix multiplication in FPGA
PS = sending and receiving block matrices between PS and PL, summation of blocks matrices in PS
RPC = sending and receiving matrix between network client and server

References