SOLUTION BRIEF

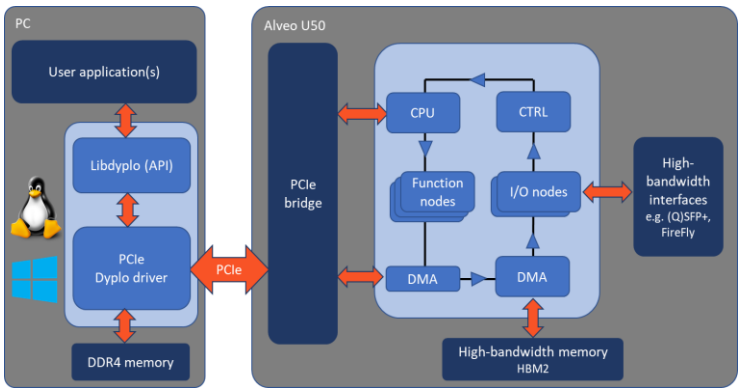
Dyplo® – FPGA-centric software acceleration made easy

INTRODUCTION

Applications can benefit from Dyplo® when FPGA type of algorithmic acceleration is required to achieve software performance goals or manage FPGA implemented dynamic processing pipelines. Using Dyplo®, you reduce the complexity of programming FPGA fabric to the level of programming GPU devices using OpenCL coding style. The usage of FPGA fabric for software programmers feels like working with software threads.

PRODUCT OVERVIEW

Applications integrating functionality of both the processors as well as the FPGA fabric, require data to be exchanged/shared between the two entities. This is typically where expertise from different disciplines is needed: writing Linux kernel drivers, construction of proper DMA based data exchange mechanisms, high-performance FPGA interfaces according to strict bus protocols and software programming skills. Here multiple programming disciplines meet.



To address this, TOPIC developed Dyplo®, a Dynamic Process Loader. On the FPGA side, Dyplo® forms a Network-on-Chip (NOC), wrapping fixed and dynamically exchangeable FPGA function blocks. On the processor side, Dyplo® is a Linux kernel driver/API that interfaces with the Dyplo® NOC using file based data streams. The third aspect of Dyplo® is the implementation flow to transform a software defined function block into a Dyplo® wrapped FPGA function block.

Software-like threading on FPGA fabric

- Easy-to-use software API
- Hustle-free integrated DFX support
- Seamless utilization of Vivado HLS
- C/C++ and VHDL/Verilog support
- Zynq 7000/Ultrascale+ support
- Alveo U50/U200/U250/U280 support
- Deterministic NOC on FPGA fabric

DYPLO® Powered by TOPIC

Adaptable. Intelligent.

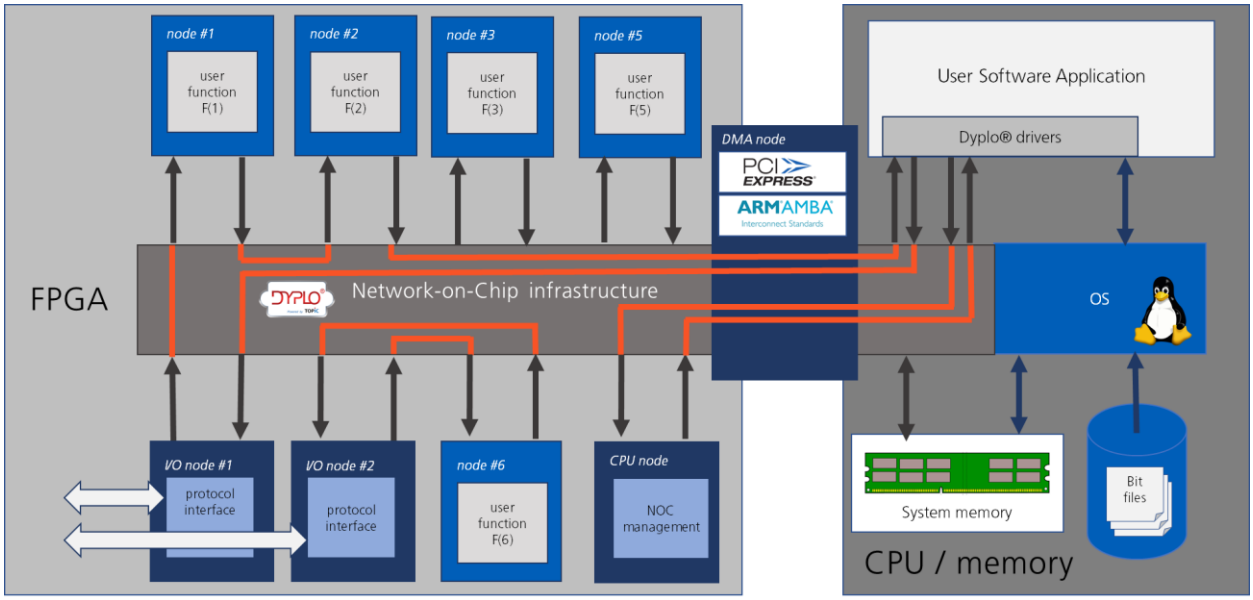
SOLUTION OVERVIEW

The Dyplo® concept is based on streaming data transport. The FPGA communication infrastructure is loosely inspired by Kahn Processing Networks (KPN). Nodes (accelerator regions) are interacting via a special buffering concept. This synchronizes operations between nodes and matches computational performance with the available communication bandwidth. In the software application, the data from and to the FPGA are available as regular data streams.

Dyplo® in 3 steps

Step 1: Dyplo® Network-on-Chip (NOC)

The FPGA part of Dyplo implements a deterministic NOC on the FPGA fabric, wrapping reconfigurable function regions with a high-performance communication infrastructure. The number and size of DFX regions and the position on the fabric are configurable, as well as the performance of the communication backbone. The DMA data transfer engine provides a ready-to-use low-latency link between PL and PS.



Step 2: Linux BSP integration on PC and Zynq 7000/Ultrascale+

The Linux driver for Dyplo® creates a software abstraction of the NOC data interfaces. It manages loading of the reconfigurable nodes as well as the on-the-fly configurable routes between the nodes. The data streams are compatible with e.g., Gsteamer and regular Linux file I/O.

Step 3: Accelerator management

The Dyplo® GUI automates the process to get your C/C++ or VHDL/Verilog code (semi) automatically mapped on the reconfigurable regions using the Vivado HLS and Xilinx DFX technology. You just need to develop/focus on the software functionality. Synthesis, place & route and creation of the FPGA partial bitstreams is taken care of by Dyplo®.

TAKE THE NEXT STEP

Learn more about Xilinx Alveo accelerator cards
Learn more about TOPIC Embedded Systems or Dyplo®
Reach out to TOPIC sales