TaPaSCo
The Task-Parallel System Composer

Build heterogeneous FPGA systems with ease!

Key Features
- Builds complete FPGA SoC-designs from HLS kernels or custom HDL cores
- Automates Design-Space Exploration to determine best system composition
- Supports wide variety of Xilinx platforms
- Includes software API for dispatching compute tasks to FPGA

Architecture
- Standardized interface for cores
- Processing Cluster aggregates core of same kind
- Platform-independent Architecture combines clusters
- Platform abstracts over platform-specific components

Use Cases
- OpenMP Device Offloading to FPGAs
- Network Acceleration
- RISC-V Softcores
- Heterogeneous System Architecture (HSA) on FPGAs

Supported Platforms

Datacenter
- Xilinx Alveo AU250
- Xilinx Virtex Ultrascale+ VCU1525
- Xilinx Virtex Ultrascale+ VCU118
- Xilinx Virtex Ultrascale+ VCU108
- Digilent NetFPGA-SUME
- Xilinx Virtex 7 VC709

Edge Devices
- Xilinx Zynq UltraScale+ MPSoC ZCU102
- Xilinx Zynq 7000 ZC706
- AVNET ZedBoard
- AVNET Ultra96v2
- Digilent PYNQ-Z1

Usage

Hardware Design Flow
- Easy-to-use build flow to create FPGA SoC
- Program with HLS C/C++ or embed custom HDL core
- Automatically connects core to memory and host

Software API
- Software API for dispatching tasks to accelerator
- API available in C and C++
- Platform-independent - write once, run everywhere

C++ API example:
```cpp
Tapasco tapasco;
auto a_wrapped = makeWrappedPointer(a.data(), a.size());
auto b_wrapped = makeWrappedPointer(b.data(), b.size());
auto job = tapasco.launch(CORE_ID, makeInOnly(a_wrapped), makeOutOnly(b_wrapped));
job(); // Wait for completion
processResult(b);
```

References


https://github.com/esa-tu-darmstadt/tapasco
tapasco@esa.tu-darmstadt.de