SYCL for Xilinx Versal ACAP AIE CGRA
Keryell, Ronan; Gozillon, Andrew; Harnisch, Gauthier; Kwon, Hyun; Chakaravarthy, Ravikumar; Wittig, Ralph

Accepted/In press: 27/04/2021

Document Version
Early version, also known as pre-print

Link to publication on the UWS Academic Portal

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy
If you believe that this document breaches copyright please contact pure@uws.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.

Download date: 05 May 2021
SYCL for Xilinx Versal ACAP AIE CGRA

Andrew Gozillon² Gauthier Harnisch³ Ronan Keryell¹ Hyun Kwon¹ Ravikumar Chakaravarthy¹ Ralph Wittig²
¹Xilinx ²University of the West of Scotland

Abstract

SYCL is a single-source C++ DSL targeting a large variety of accelerators in a unified way by using different backends.
Xilinx Versal ACAP is a new system-on-chip (SoC) device integrating various computing resources like various CPUs, an FPGA, a coarse-grain reconfigurable array (CGRA), etc. interconnected by different network-on-chip (NoC).
The AIE CGRA is an array of 400 VLIW DSP operating on 512-bit vectors with their own neighborhood distributed memory (32 Kib data, 16 Kib instructions). We expose architectural details to the programmer through some SYCL extensions and extend SYCL with a geographical collective model.
The SYCL implementation targeting the AIE CGRA by merging 2 different open-source implementations, the SYCL implementation targeting the AIE CGRA by merging 2 different open-source implementations, geographical collective model. We expose architectural details to the programmer through some SYCL extensions and extend SYCL with a geographical collective model.

Weaving heterogeneous tiles and memories with meta-programming

Type-safe access to heterogeneous neighbor memories (PDE, stencil...)

Different multi-level implementation/emulation for hardware-software co-design

First approach: expose geographical view of the chip

SYCL with individual tile model: 1 tile = 1 sub-device

Second approach: entangled mode by weaving tiles and memories

Implementation

Conclusion

References

1. Retargetable to other CGRA

Source File

```cpp
#include <iostream>

# include "tipe:tile.cpp"

int

```