Multithread accelerators on FPGAs: a Dataflow–based Approach

Francesco Ratto[1], Luigi Raffo[1], Francesca Palumbo[2]

[1]University of Cagliari (IT), [2]University of Sassari (IT)

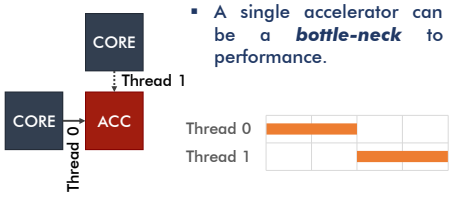
francesco.ratto@unica.it

Abstract

Multithreading is a well-known technique to deliver performance gain, raising resource efficiency by exploiting underutilization periods. In this work, we describe a model-based approach for designing custom multithread hardware accelerators targeting reconfigurable fabric. This approach exploits dataflow models of applications and tagged tokens to let the resulting hardware support concurrent threads. Results highlight that the proposed accelerators achieve a valuable tradeoff between a set of parallel single-thread accelerators and a single-thread accelerator multiplexed in time. The ongoing and future work to validate and improve the design approach are presented.

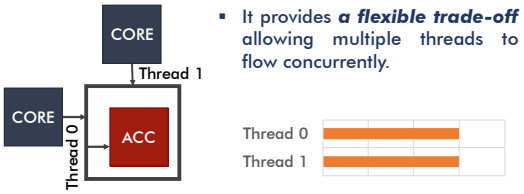
Motivation and goal

a) Single-thread accelerator



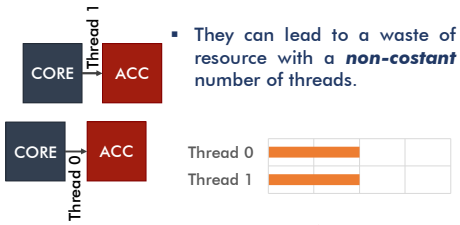
- A single accelerator can be a ***bottle-neck*** to performance.

b) Multithread accelerator

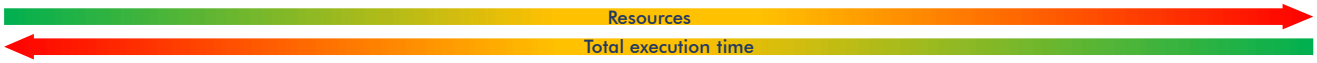


- It provides ***a flexible trade-off*** allowing multiple threads to flow concurrently.

c) Replicated accelerators

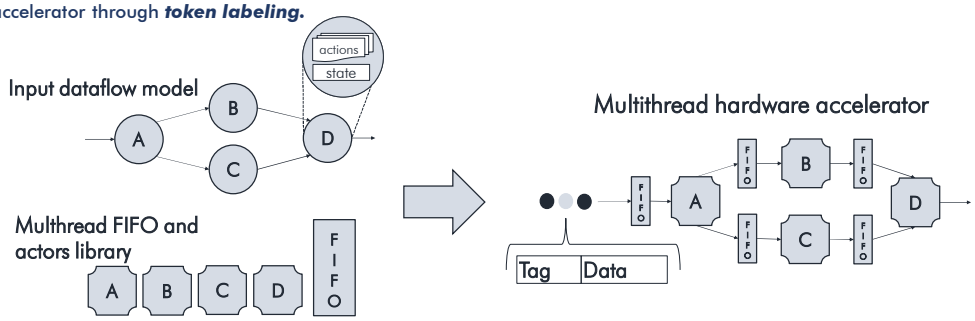


- They can lead to a waste of resource with a ***non-costant*** number of threads.



Design approach

The proposed ***model-based approach**** allows, starting from the single-thread ***dataflow specification*** of an application and ***without*** explicit need of ***data synchronization***, to design a corresponding multithread hardware accelerator through ***token labeling.***

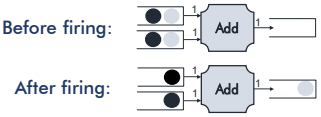


Developed components that meet the above functional requirements:

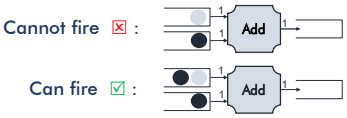
- Multithread FIFO ***interface*** to allow selecting the reding thread (Req 3) and to know the status of the FIFO with respect to each thread (Req 2).
- ***Actor*** architecture with ***shared logic*** to process tokens. ***The state is replicated*** for each thread ***and multiplexed*** depending on the tag of the input token.
- Two ***FIFO*** architectures. One with a ***dedicated memory per thread*** and a simple control logic. And one with ***two shared memories***, one for storing tokens and one for their order, but a more complex control logic.

Functional requirements

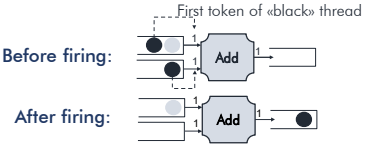
1. A firing actor must ***tag the output tokens*** with the same tag of the input ones



2. The firing rules must be adjusted so that only ***matching tokens*** can fire the execution.



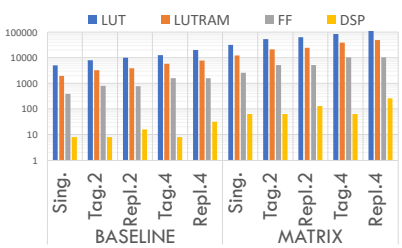
3. ***FIFOs must provide semi-out-of-order read***, letting the reading actors choose among the first token of each flow of execution.



Results

The design approach has been ***tested*** on two versions (Baseline and Matrix) of a ***two-stage filter*** adopted in the motion compensation phase of the HEVC standard.

Performance gain is obtained ***exploiting idle periods*** in the single-thread execution and allowing light threads to ***"overtake"*** heavier one.



The resource utilization of the multithread accelerators stands in between the ***single-thread*** ones and the corresponding ***set of replicas***.

Ongoing and future work

A complete host processor – accelerator environment with OS support and an API for thread instantiation is under development.

Validation of the approach on a heterogenous platform processing LTE workloads in collaboration with CC Chair at the CFAED of TU Dresden.

Integration of the proposed design approach with HLS tool to make it available to developers and speed up the design process.

* Ratto, F., Esposito, S., Sau, C., Raffo, L., & Palumbo, F. (2022). Multithread Accelerators on FPGAs: A Dataflow-Based Approach. In Proceedings of PARMA-DITAM 2022. Schloss Dagstuhl-Leibniz-Zentrum für Informatik..