Content:

1. Embedded Linux System
   - Embedded Linux Development Environment
   - FSBL+Uboot
   - Kernel
   - Root FileSystem
2. DMA
   - New Development SoC Platform
   - DMA interface
   - DMA performance
3. “Full-Chain” Bandwidth
   - File systems
   - Results
4. Specific Development for AMchip
   - JTAG communication
5. Future works
   - AMchip
     - Complete DMA communication with Ali’s IP
   - “Full-Chain” Bandwidth
   - Documentation

Embedded System:

AMchip Interfacing and driving, R&D

M. Vincent Voisin
CNRS, Software Engineer
LPNHE,
ATLAS group, AMchip team
Contact:

vvoisin@lpnhe.in2p3.fr

New development platform based on ZYNQ-FPGA and AMchip for different applications

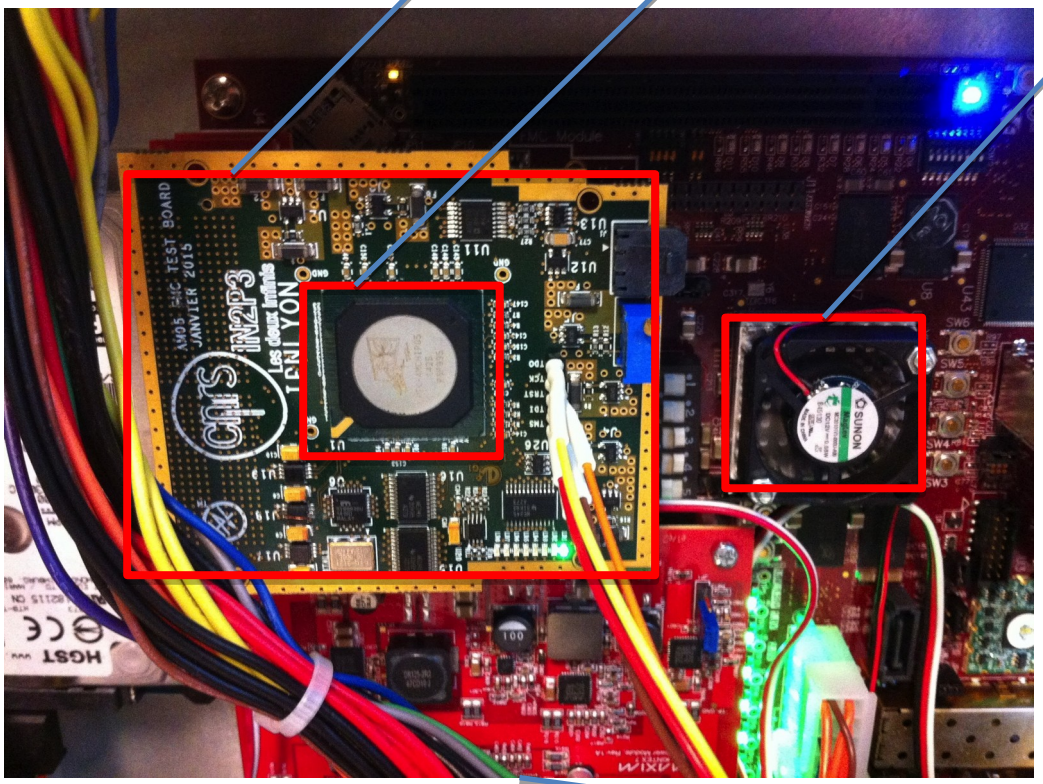
ZYNQ SOC

AMchip board

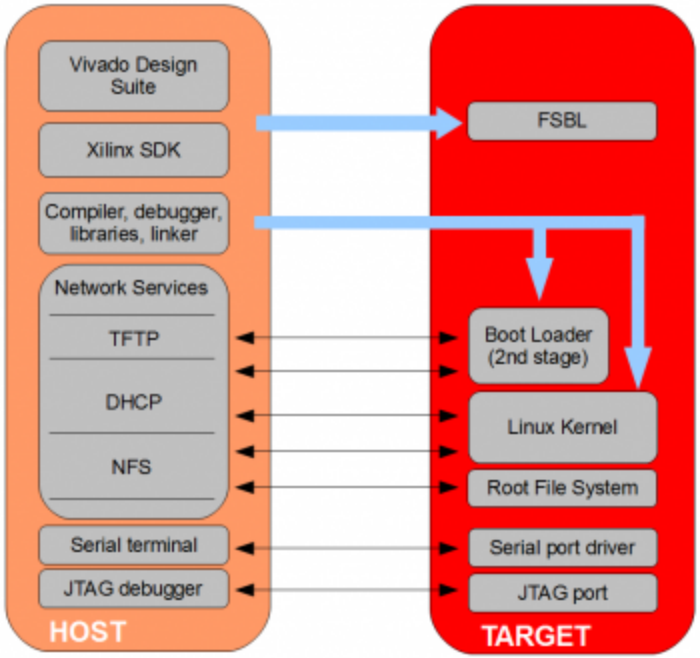
AMchip

New AMchip test-bench and development platform

Hardware of The Development Platform



Embedded Linux Development Environment



Boot-loaders

- First Stage Boot-loader
  - Setup the PS and load the second stage Boot-Loader
- U-boot (The second stage Boot-Loader)
  - A standard on embedded system
  - Execute the linux kernel
  - https://github.com/Xilinx/u-boot-xlnx

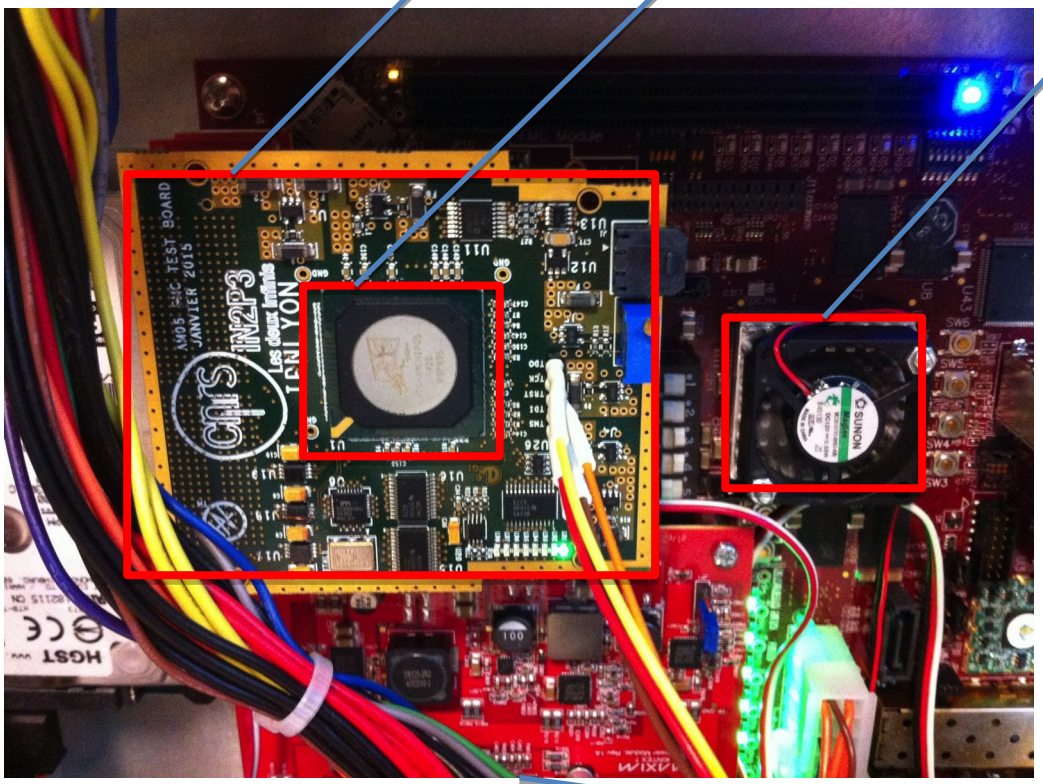
The kernel

- The Official Linux Kernel from Xilinx
  - Very recent Kernel: Linux v4.4.x
  - https://github.com/Xilinx/linux-xlnx
  - The kernel is on a TFTP server

The Root File System

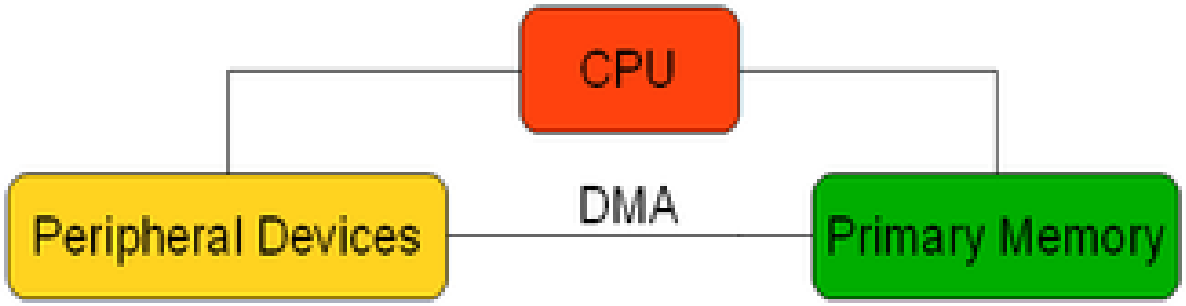
- Based on Ubuntu Core 16.04.02
  - Very recent version and LTS support
  - The RFS is on a NFS file system on the Host PC
  - http://cdimage.ubuntu.com/ubuntu-base/releases/16.04.2/release/ubuntu-base-16.04-core-armhf.tar.gz

Hardware of The Development Platform

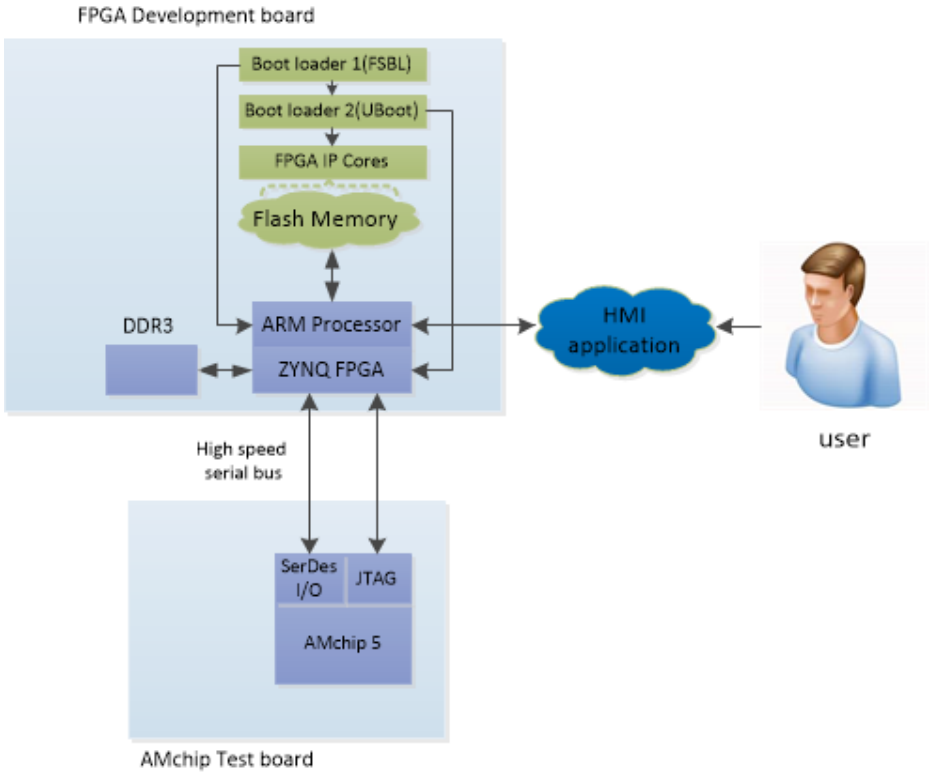


Direct Memory Access

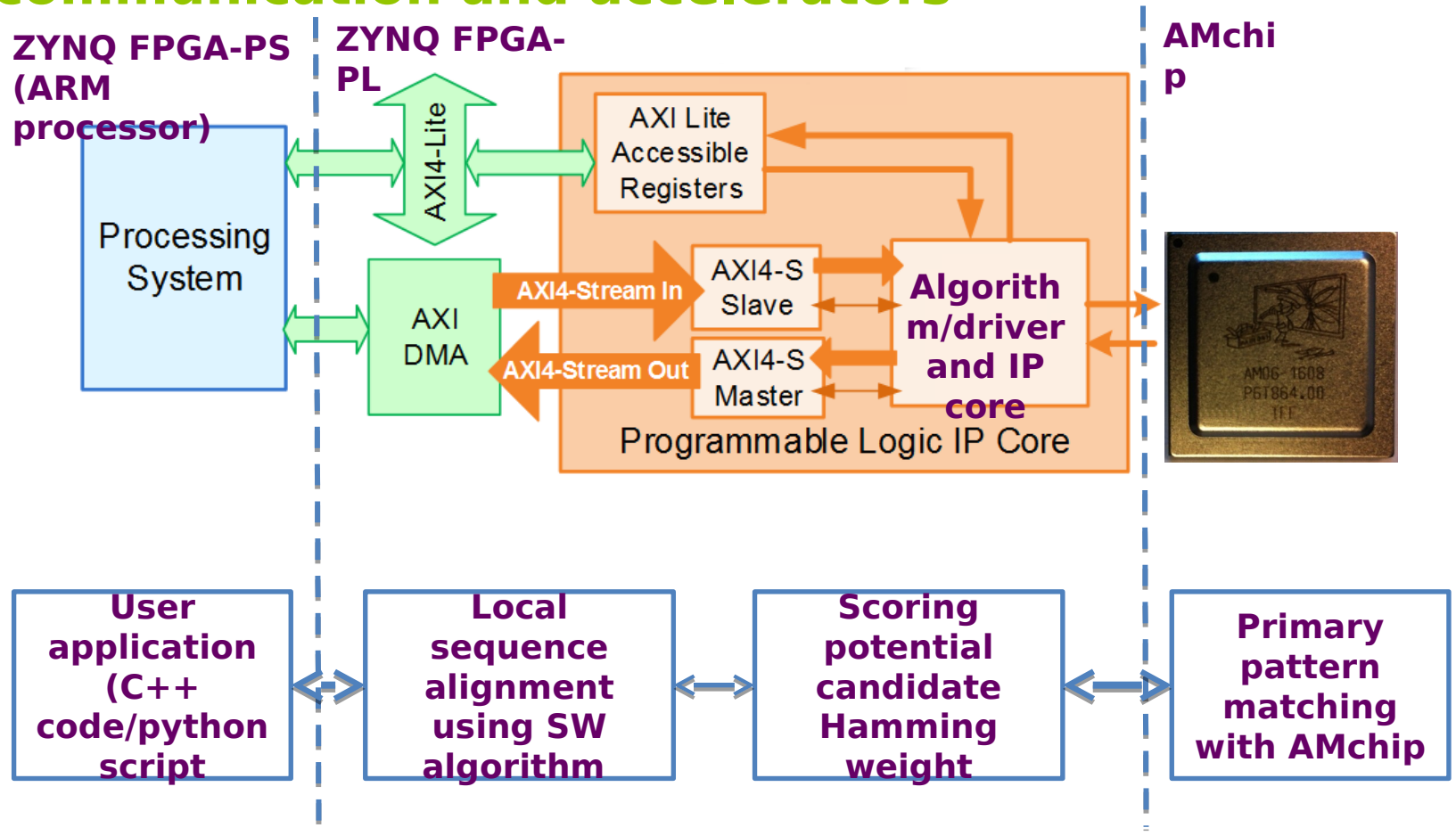
- The purpose of using DMA :
  - Transfer a large amount of data to/from the Programmable Logic to UserSpace Memory
  - Avoid the use of CPU load for transfer of large amount of data



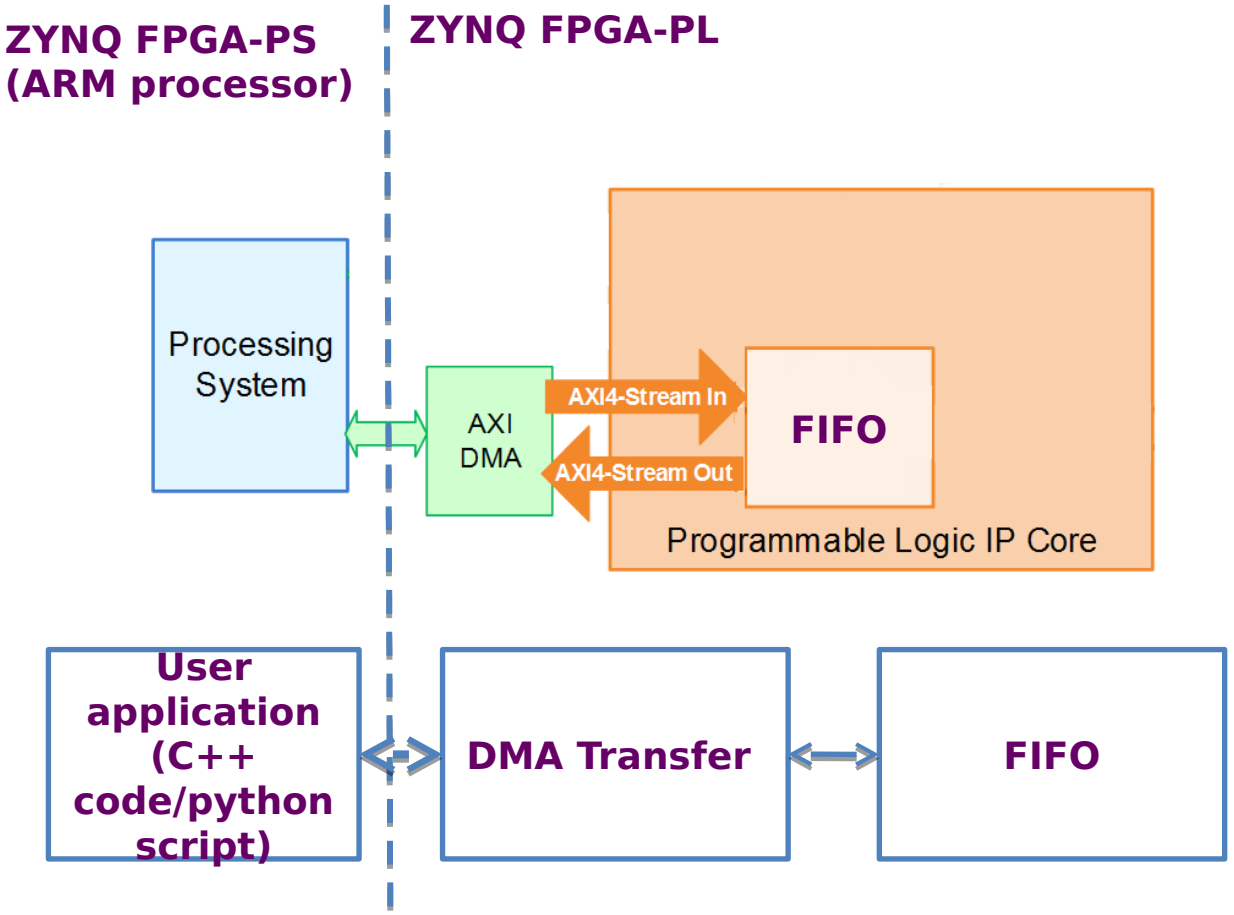
New Development Platform/Architecture based on ZYNQ-SOC



Full-chain high-speed link for data communication and accelerators



Full-chain high-speed link for data communication

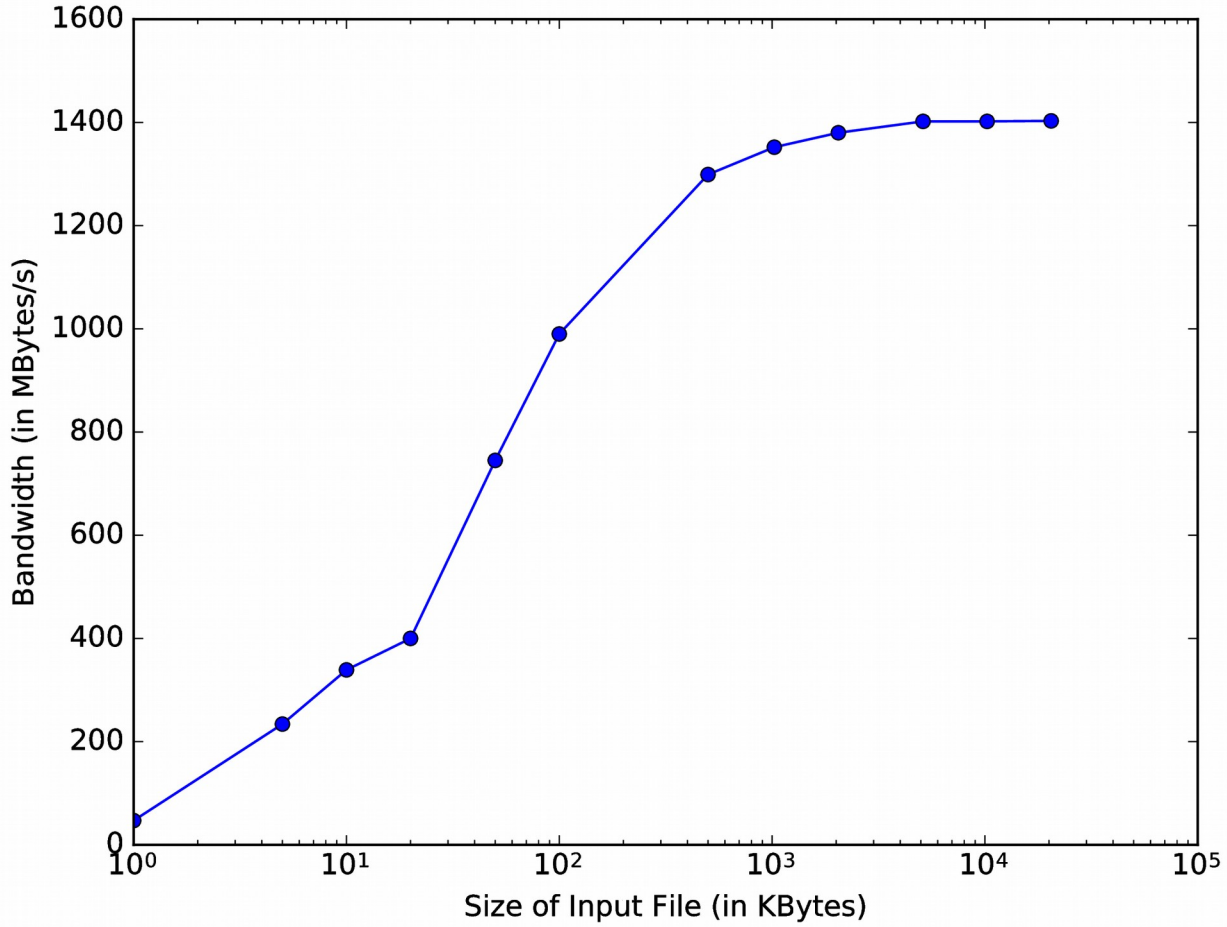


The libgannet solution

- A complete framework to create DMA interface
  - Programmable Logic
  - Softwares (linux driver + user-space library to handle DMA)
- https://gitlab.com/SmartAcoustics/libgannet
- Python wrappers were created to make the development easier

DMA Bandwidth

Max = 1.4 GBytes/s



“Full-Chain” bandwidth

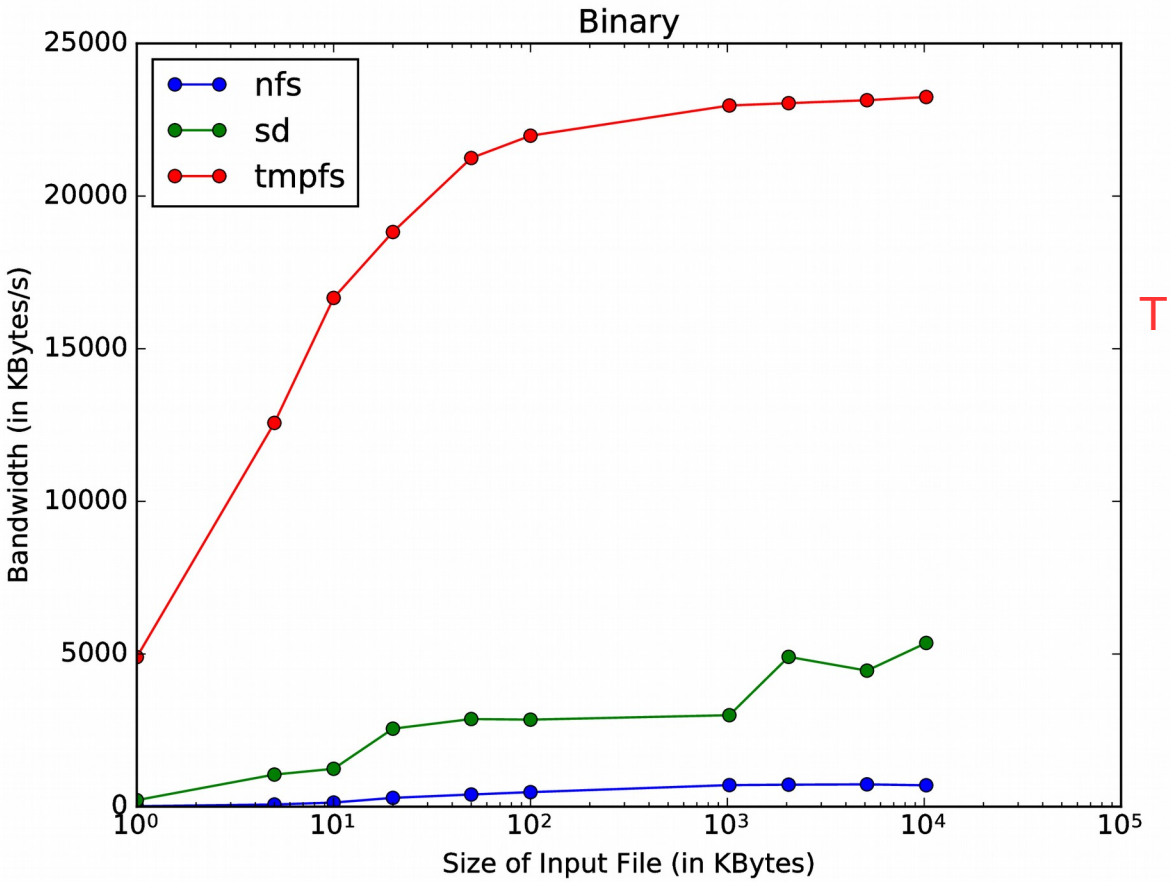
- Evaluation of “full-chain” bandwidth:
  - Data are read from a file and put in memory
  - Data in memory are send to PL (here a FIFO) through DMA
  - Data are read back to Memory from FIFO
  - Data in memory are saved in another file

Full-Chain bandwidth

- Two kinds of file format
  - Data File is in a binary format
  - Data File is in a JSON format
- Three types of file systems
  - NFS (slow)
  - SDCard
  - tmpfs (fastest, useful to discover bottlenecks)

“Full-chain” Bandwidth

With files in binary format



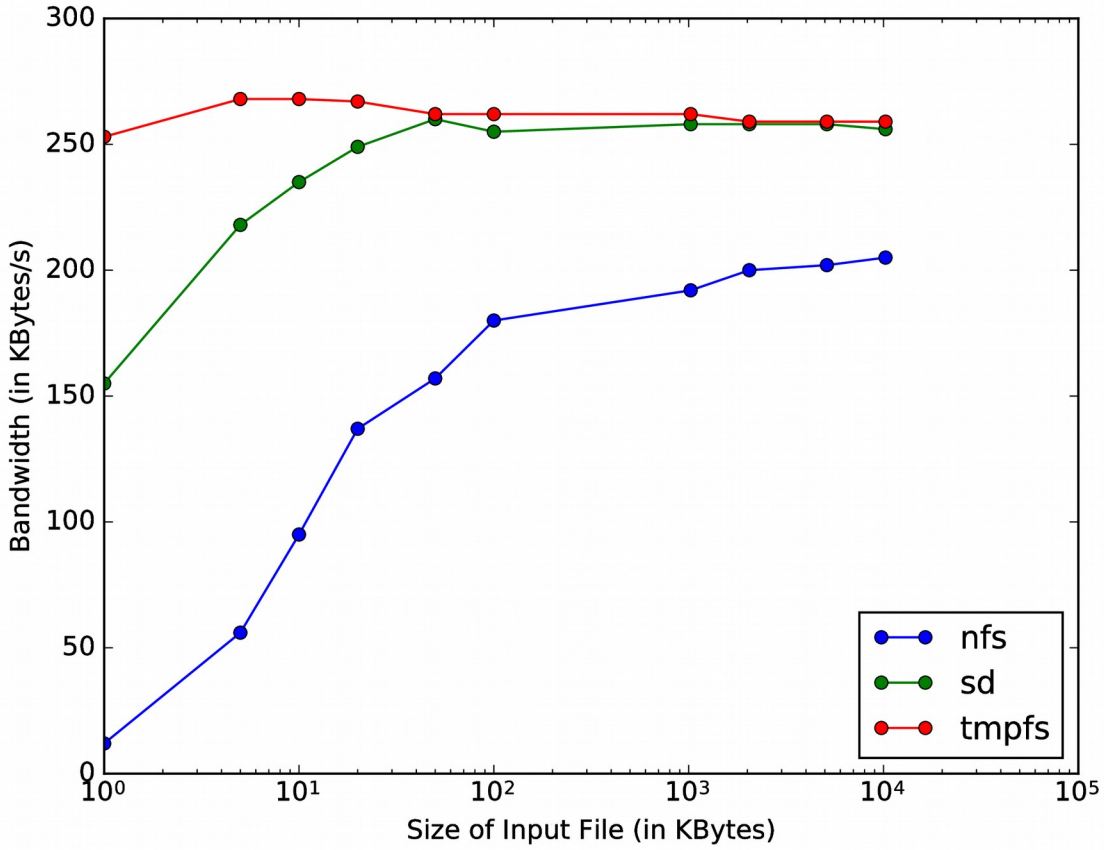
NFS = 730 KBytes/s

SD = 5.36 MBytes/s

TMPFS = 23.25 MBytes/s

“Full-chain” Bandwidth

With files in JSON format



NFS = 205 KBytes/s

SD = 256 KBytes/s

TMPFS = 259 KBytes/s

Publications of the results

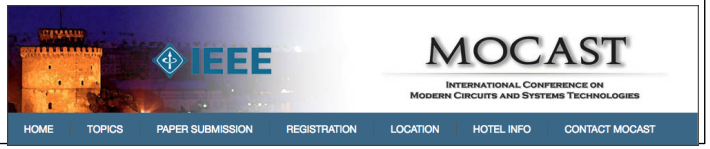
Heterogeneous computing system platform for high-performance pattern recognition applications

M Ali Mirzaei*, Vincent Voisin*, Alberto Annovi‡, Guillaume Baulieu†, Matteo Beretta¶, Giovanni Calderini*, Saverio Citraro‡, Francesco Crescioli*, Geoffrey Galbit†, Valentino Liberali§, Seyed Ruhollah Shojaii§, Alberto Stabile§, William Tromeur†, and Sebastien Viret†
*LPNHE, IN2P3, CNRS, UPMC; 4 Place Jussieu, 75005 Paris, France; Email: mmirzaei@lpnhe.in2p3.fr
†IPNL, IN2P3, CNRS, UCBL; 4, Rue Enrico Fermi, 69622 Villeur., France
‡Università di Pisa - INFN Pisa; Largo B. Pontecorvo 3, 56127 Pisa, Italia
§Università di Milano - INFN Milano; Via Celoria 16, 20133 Milano, Italia
¶INFN Laboratori Nazionali di Frascati; Via Enrico Fermi 40, 00044 Frascati, Italia

***Abstract*—we present a system architecture made of a motherboard with a Xilinx Zynq System on Chip (SoC) and a mezzanine board equipped with an Associative Memory chip (AM). The proposed architecture is designed to serve as an accelerator of general purpose algorithms based on pipeline processing and pattern recognition. We present the open source software and firmware developed to fully exploit the available communication channels between the ARM CPU and the FPGA using Direct Memory Access (DMA) technique and the AM using Multi-Gigabit Transceivers (MGT). We report the measured performances and discuss potential applications and future developments. The proposed architecture is compact, portable and provide a large communication bandwidth between components.**

Result and publication:

This paper has already been accepted and will be presented by Vincent VOISIN in 4 - 6 May 2017 Thessaloniki Greece



JTAG Communication

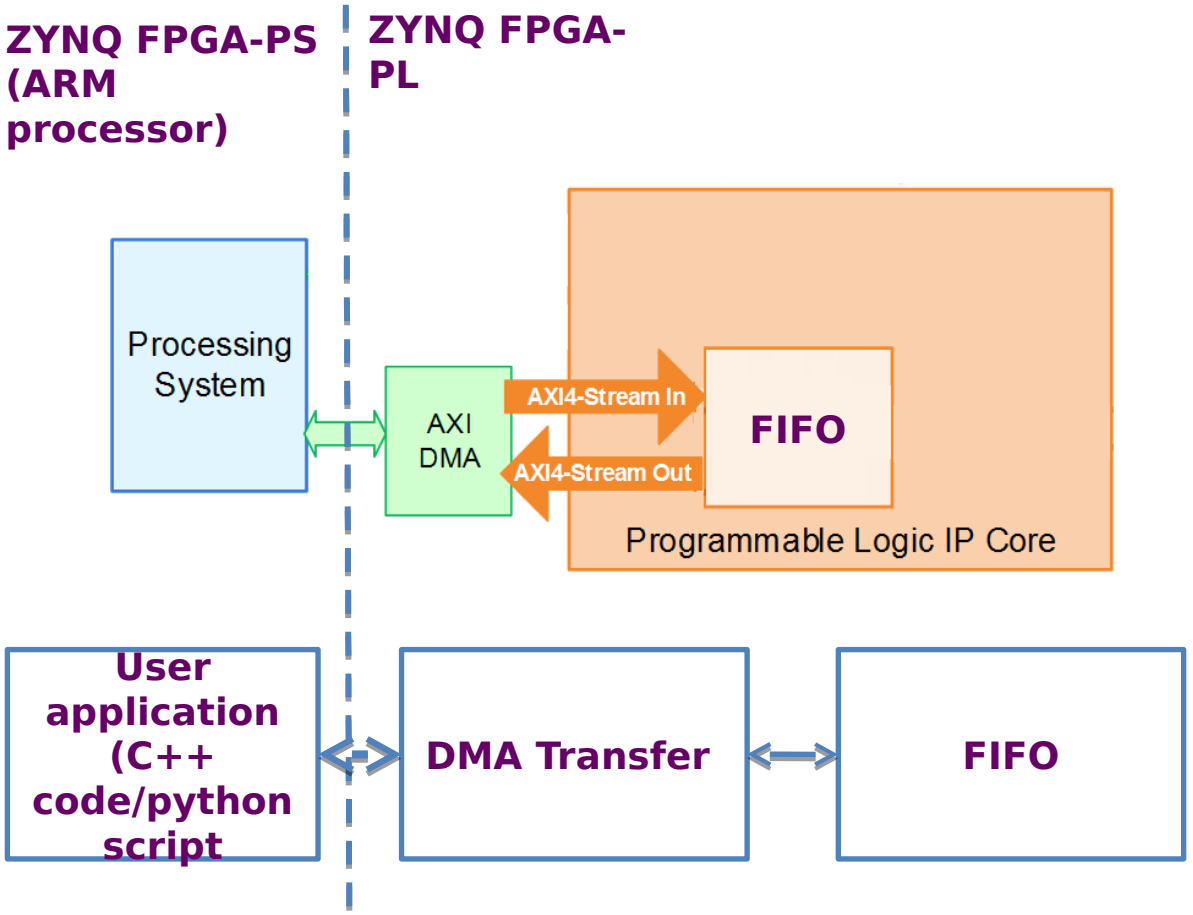
- To configure the AMchip we need to communicate through a JTAG port
  - No standard output JTAG on Zynq
  - A solution was found, we use “standard” ZYNQ GPIO to “emulate” a JTAG bus
    - Inspired by a library in Kovan-JTAG project: https://github.com/xobs/kovan-jtag
    - Simple but slow method

    At the moment it is slow and takes 2minutes to load reference patterns, however the high-speed solution is found and under development

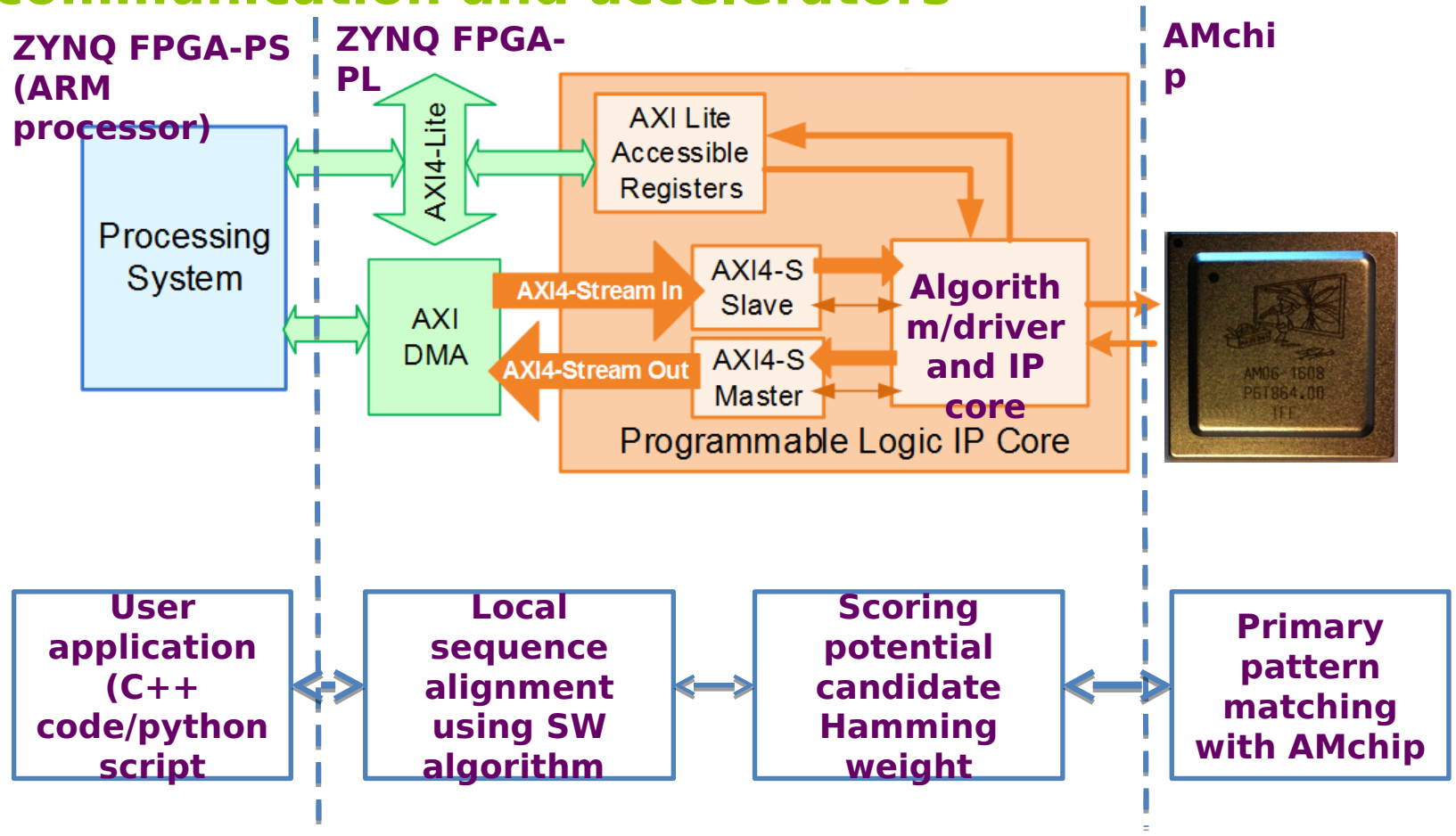
On AMchip

- Finish to develope a complete system to communicate with AMchip
  - JTAG is ready for AMchip configuration
  - DMA software is ready
  - Need to test Smith-Waterman Scoring IP core

Full-chain high-speed link for data communication



Full-chain high-speed link for data communication and accelerators



On AMchip

- Finish to develope a complete system to communicate with AMchip
  - JTAG is ready for AMchip configuration
  - DMA software is ready
  - Need to test Smith-Waterman Scoring IP core

On “Full Chain” Bandwidth

- Speed up the read/write of data
  - Use a database such MongoDB rather than files
  - Use a >1Gbits/s network adapter (speed up NFS)
  - Use a SATA disk
  - Receive/send data from an external PC through a >1Gbits/s using on-board SFP socket
  - Use OProfile to investigate and trace software performance and data communication

Documentation

- Need a lot of works to document the developments and distribute it
- Modify and improve this presentation for MOCAST Conference
- Add results to this presentation for MOCAST Conference