FPGA Acceleration for High Dimensional Inverse Kinematics

Yike Li, Lu Li, Howie Choset

ShanghaiTech University, Carnegie Mellon University Robotic Institute

Introduction

For complex structured robots with 6 or more degrees of freedom, there are infinite solutions to inverse kinematics problem. It usually takes a long time to consecutively compute the joint angles. It increases robots' response time and the dependency of high performance computer, which decrease robots' mobility. We designed an FPGA based inverse kinematics method to accelerate the processing. The key features are:

- Direct memory access(DMA) to connect external memory and FPGA acceleration core
- Ability to handle high-speed streaming data
- Easy to customize the design via C++ code

Methods

The robot needs to minimize the distance($\boldsymbol{e}$) between end-effector($\boldsymbol{s}$) and target($\boldsymbol{t}$) as best as it can.(D_{max} is the maximum distance in a single move of the end-effector.)[1]

$$\boldsymbol{e} = f_{ClampMag}(\boldsymbol{t} - \boldsymbol{s}, D_{max}), \tag{1}$$

where

$$f_{ClampMag}(\boldsymbol{w}, d) = \begin{vmatrix} \boldsymbol{w} & \text{if } \|\boldsymbol{w}\| \leq d \\ d\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} & \text{otherwise} \end{vmatrix} \tag{2}$$

Levenberg-Marquardt method [1] :

$$\Delta\boldsymbol{\theta} = (J^T J + \lambda^2 I)^{-1} J^T \boldsymbol{e} \tag{3}$$

To reduce the dimension of the second term,

$$\Delta\boldsymbol{\theta} = J^T (J J^T + \lambda^2 I)^{-1} \boldsymbol{e} \tag{4}$$

Discussion

Future work:

- Avoid singularity problem.
- Increase ARM core's functionality to increase the cooperation between FPGA and ARM core.
- Implement this method to real robot and test the performance in real scenario.

Implementation

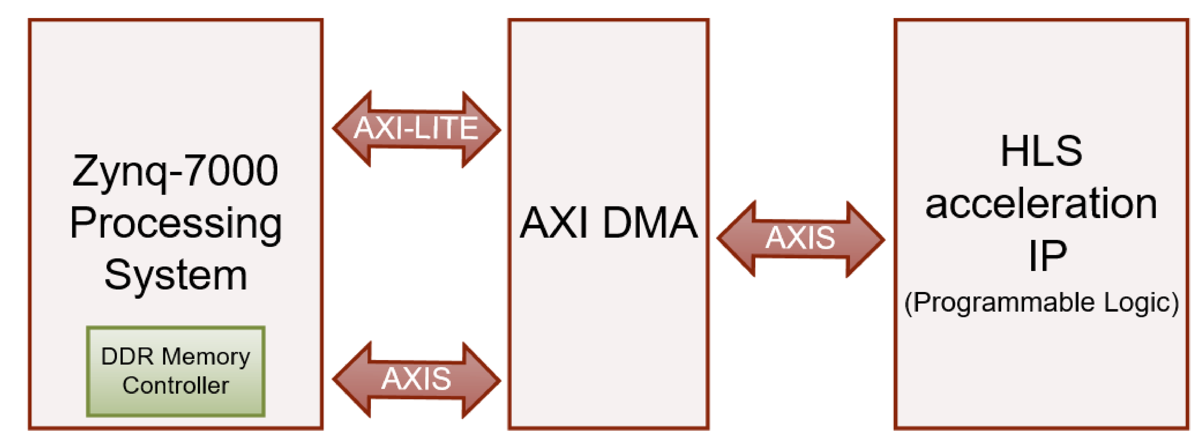

Figure 1: System design and connections between PS and PL

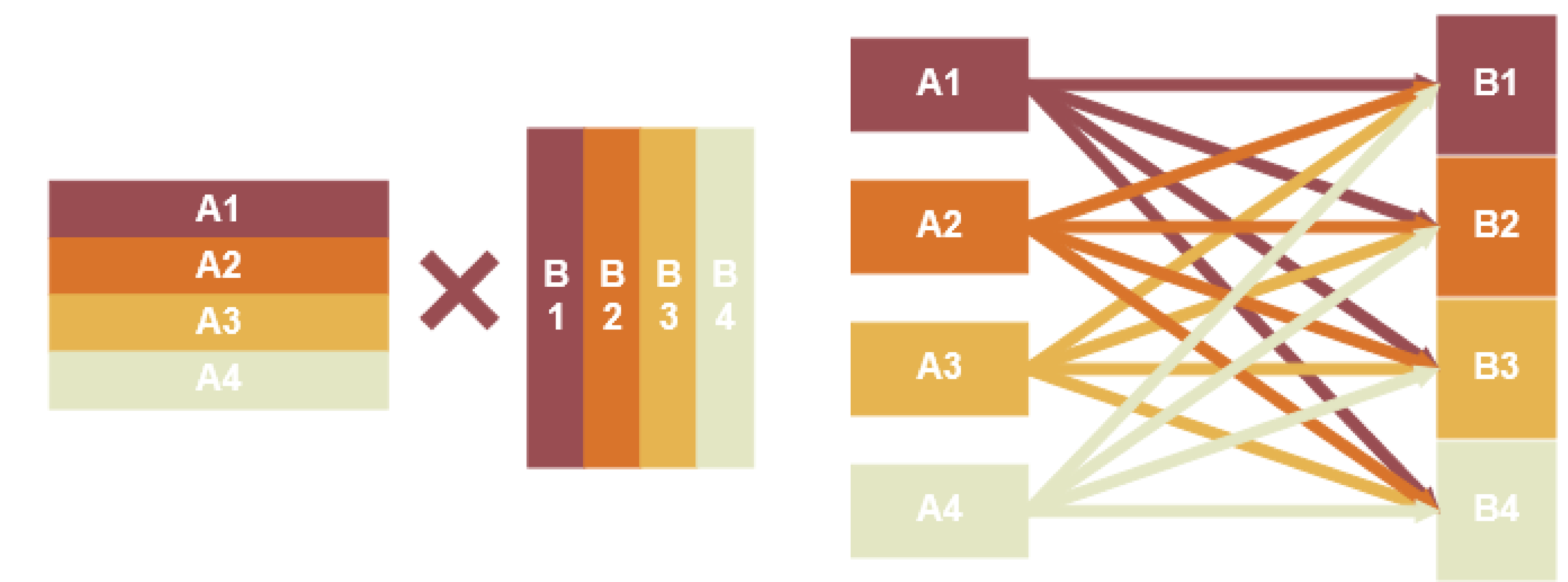

Figure 2: Working principle of FPGA. All the arrows are computed simultaneously

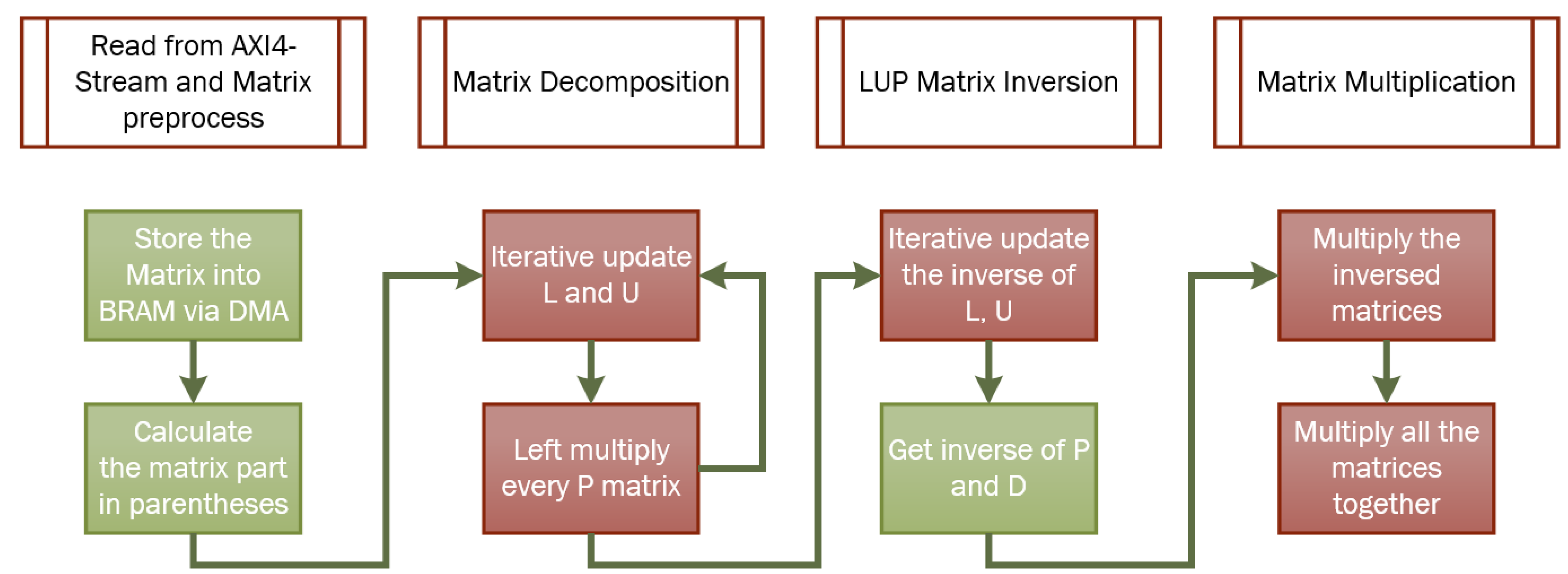

Figure 3: Details of the implementation. Accelerated blocks are indicated in red

Benchmark Test Results

Target: 32-bit floating point inverse kinematics problem for 16 DOF robot
Number of test: 1000
Performance:

- MATLAB on Interl(R) Core(TM) i7-5500U CPU @2.40GHz: 0.135ms
- Zynq-7000S device features a single-core ARM $Cortex^{TM}$-A9 processor @100MHz mated with Artix-7 based programmable logic: 0.357ms

Our board is 10 times cheaper, several times smaller. It works with lower frequency, but performs almost as good as a computer.

References

[1] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods."

Acknowledgment

Special thanks to Dr. Lu Li, Prof. Howie Choset, Rachel Burcin, John Dolan, Ziqi Guo, and RISS cohort and ShanghaiTech University.