



NTNU

TDT4295 - COMPUTER DESIGN PROJECT

Project Report

Convolution team

Peter Aaser

Mattis Spieler Asp

Truls Fossum

Fredrik Haave

Øyvind Kjerland

Arnstein Kleven

Mathias Ose

October 22, 2015

Abstract

Contents

1	Introduction	1
2	Description & Methodology	2
2.1	FPGA	2
2.2	Data in	2
2.3	Convolver	2
2.3.1	Datafeeder	2
2.4	Data out	3
3	Results	5
3.1	Discussion	5
4	Evaluation of assignment	6
5	Conclusion	7

1 | Introduction

2 | Description & Methodology

2.1 FPGA

The very heart of our computer is our custom made architecture implemented on our FPGA. In this section we will first describe the overall architecture of our convolution engine before we will examine our implementation.

2.2 Data in

We used an EBI bus!

2.3 Convolver

The two major parts of the convolution engine is the data feeder and the executing unit. By decoupling these we ensure a modular design allowing us to try a range of different approaches in our executing unit.

2.3.1 Datafeeder

The focal point of the convolver is the data delivery conveyor belt. This belt is a set of rows containing data, each row corresponding to a row in a convolution kernel. Each cycle the convolver expects a datum from a FIFO queue which will be inserted into the input feeder tree. Each row has a feeder tree, however the lower rows is fed by the output of the row above. In our figure we show a feeder belt for a kernel of size 3*3 using nine columns. The green registers correspond to the register that is currently being extracted. Each timestep the requested data is the register to the right of the previously requested register, wrapping around in a toroid fashion when the rightmost register was previously registered. In order to explain the purpose of our request pattern we will focus on the middle row, specifically the elements in this row that is surrounded by as many elements required by the convolution kernel. In fig TODO these pixels are colored purple, and the surrounding pixels required to convolute the three first pixels are drawn with arrows pointing to the middle pixel. When an operation is described "in context of a pixel", this refers to one of the center pixels. For instance, when a pixel in the bottom row is requested it is read in three different contexts, one for each neighbouring middle row pixel, as a lower left, lower, and lower right pixel. To motivate our reading pattern we will use a read from the lowest row as an example. The pixel being read on the lower row in this cycle, colored green, will be read in three contexts:

1. The rightmost lower pixel in context of the middle left pixel, marked as 1
2. The middle lower pixel in context of the middle pixel, marked as 2
3. The leftmost lower pixel in context of the middle right pixel, marked as 3

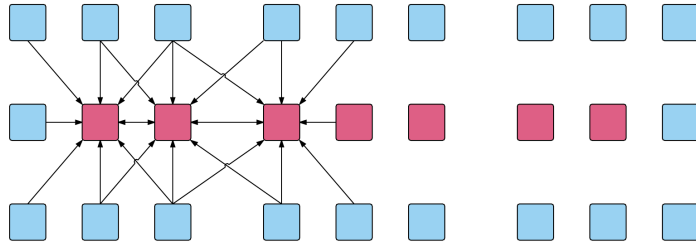


Figure 2.1: For each middle pixel, the contexts it applies to neighbouring pixels are denoted by arrows

2.4 Data out

aaa

We used HDMI!

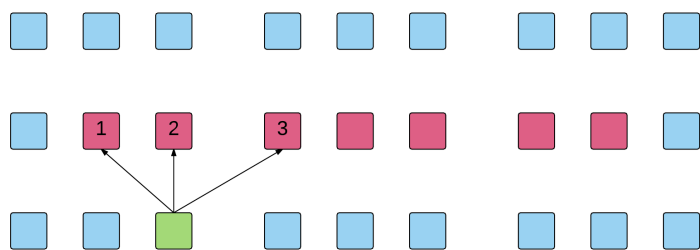


Figure 2.2: For a specific read, marked as green, the contexts it is read in are marked with arrows

3 | Results

3.1 Discussion

4 | Evaluation of assignment

5 | Conclusion

References

- [1] Edgar Xavier Ample. Foo. Technical report, Foxford University, 3000.
- [2] Mathias Ose. ma.thiaso.se. <http://ma.thiaso.se/>, 2014.