VERTIPH:
A Visual Environment for Real-Time Image Processing on Hardware

A thesis presented in partial fulfilment of the requirements for the degree of

Doctor of Philosophy

in

Computer Systems Engineering

at Massey University, Palmerston North,
New Zealand.

Christopher Troy Johnston

2009

Abstract

This thesis presents VERTIPH, a visual programming language for the development of image processing algorithms on FPGA hardware. The research began with an examination of the whole design cycle, with a view to identifying requirements for implementing image processing on FPGAs. Based on this analysis, a design process was developed where a selected software algorithm is matched to a hardware architecture tailor made for its implementation. The algorithm and architecture are then transformed into an FPGA suitable design. It was found that in most cases the most efficient mapping for image processing algorithms is to use a streamed processing approach. This constrains how data is presented and requires most existing algorithms to be extensively modified. Therefore, the resultant designs are heavily streamed and pipelined.

A visual notation was developed to complement this design process, as both streaming and pipelining can be well represented by data flow visual languages. The notation has three views each of which represents and supports a different part of the design process. An architecture view gives an overview of the design's main blocks and their interconnections. A computational view represents lower-level details by representing each block by a set of computational expressions and low-level controls. This includes a novel visual representation of pipelining that simplifies latency analysis, multiphase design, priming, flushing and stalling, and the detection of sequencing errors. A scheduling view adds a state machine for high-level control of processing blocks. This extended state objects to allow for the priming and flushing of pipelined operations.

User evaluations of an implementation of the key parts of this language (the architecture view and the computational view) found that both were generally good visualisations and aided in design (especially the type interface, pipeline and control notations). The user evaluations provided several suggestions for the improvement of the language, and in particular the evaluators would have preferred to use the diagrams as a verification tool for a textual representation rather than as the primary data capture mechanism.

A cognitive dimensions analysis showed that the language scores highly for thirteen of the twenty dimensions considered, particularly those related to making details of the design clearer to the developer.

To my family

Acknowledgements

Thanks to my supervisors Donald and Paul for all the advice and encouragement that you gave me.

Thanks to my office mates for the interesting discussions, especially Kim for being a sounding board for ideas.

Thanks to my friends and flatmates for keeping me sane and making sure I got out once in a while, especially Andy, Jess, Øyvind, Reuben and Sarah

I would like to thank my family for everything, without you I would not be the person I am today. Though you did not live to see me achieve this work thanks to Grandma and Granddad for the Lego that encouraged me into engineering and to Nana for encouraging me to think and go as far as I could. Thank you Pop for letting me build things and take them apart and for all the wood over the past years to keep me warm in winter.

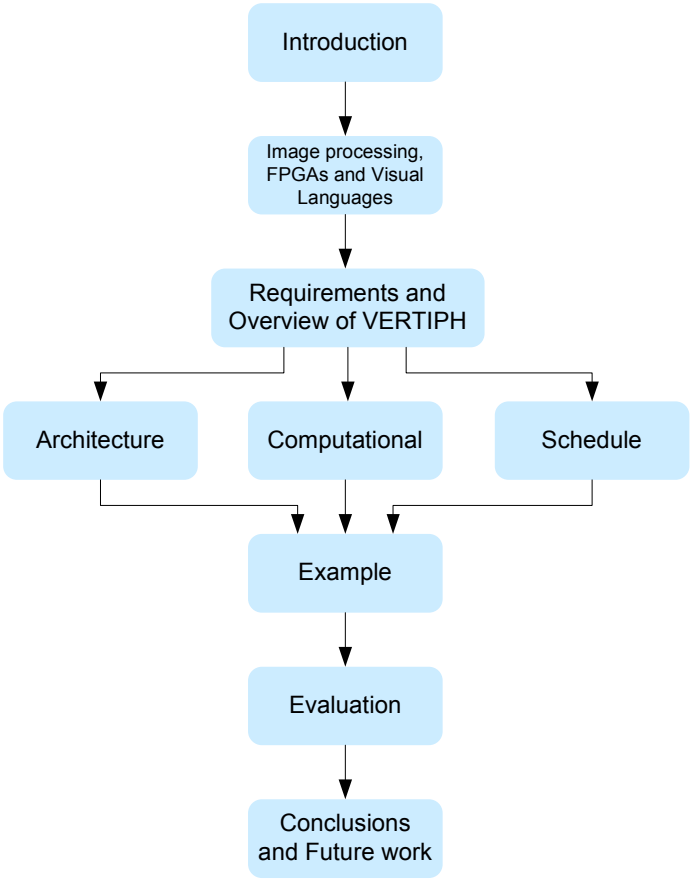
To Nic and Bridget thanks for the dinners and proof reading. AJ for just being you.

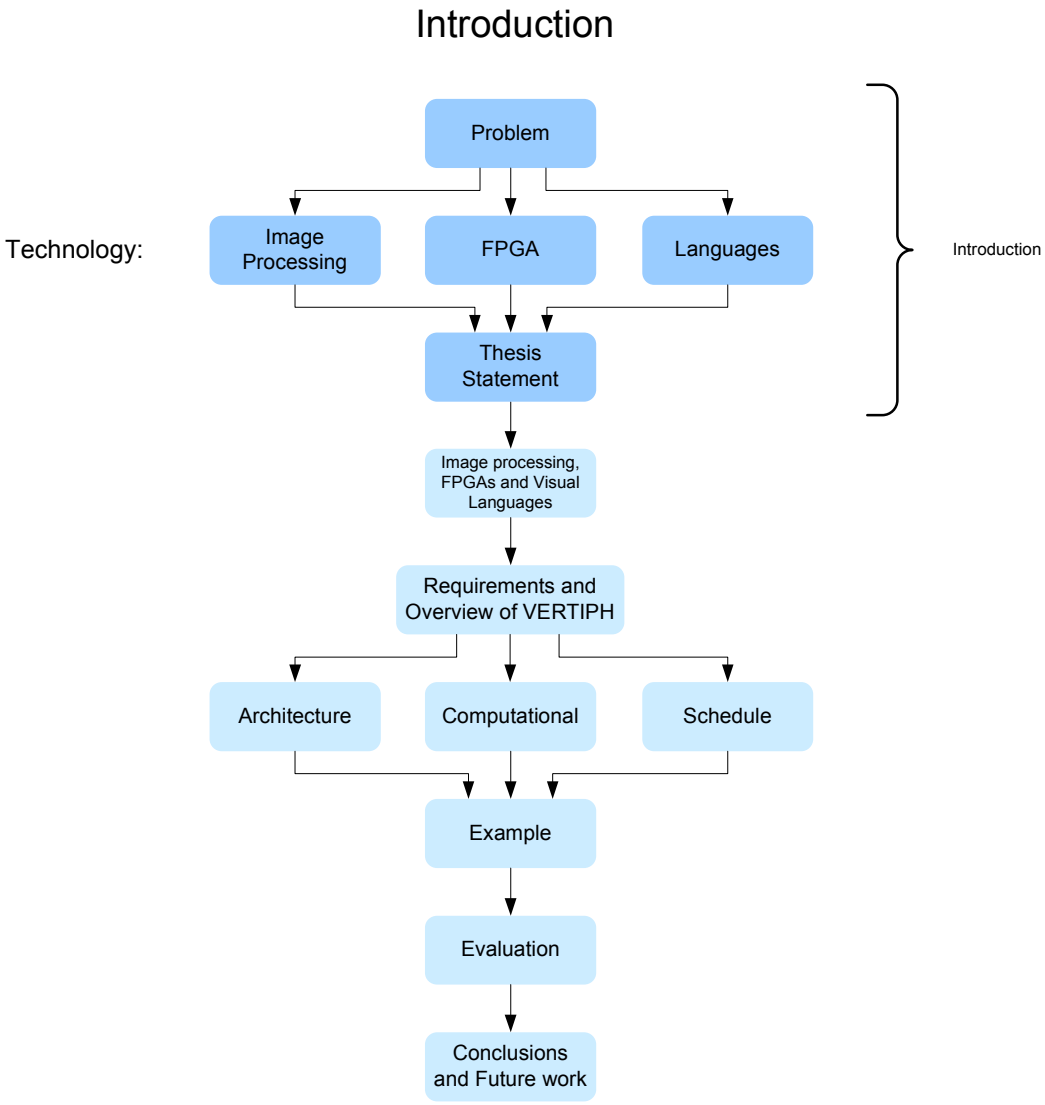
To Mum and Dad thank you for all the support and encouragement with out it I would not have got this far.

Table of Contents

List of Figures

Thesis Structure




Introduction
Technology:
Problem
Image Processing
FPGA
Languages
Thesis Statement
Image processing, FPGAs and Visual Languages
Requirements and Overview of VERTIPH
Architecture
Computational
Schedule
Example
Evaluation
Conclusions and Future work
Introduction