GateOS: A minimalist Windowing Environment and Operating System for FPGAs

A thesis presented in partial fulfilment of the requirements

for the degree of

Master of Engineering

In

Computer Systems Engineering

At Massey University, Palmerston North,

New Zealand

Andreas Buhler

2007

ABSTRACT

In order to debug and tune stand-alone FPGA image processing configurations, it is necessary for a developer to also create the required debug tools and to implement them on the FPGA. This process takes both time and effort that could be better spent on improving the image processing algorithms. The Gate Array Terminal Operating System (GateOS) is proposed to relieve the developer of the need to construct many of these debugging tools. In GateOS we separate the image processing algorithms from the rest of the operating system. GateOS is presented to the developer as a Handel-C library, which can be customised at compile-time, to facilitate the creation of windows and widgets. Several types of widgets are described that can manipulate the parameters of image processing algorithms and enable the end-user to dynamically rearrange the position of a window on the VDU. An end user is able to interact with GateOS with both a keyboard and a mouse.

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Associate Professor Donald Bailey, for all the guidance given to me over the previous year. Without his sound advice, GateOS would not be what it is today. I thank him for being brutally honest with me whenever I became sidetracked and deviated from dealing with the core issues of GateOS. I am also grateful for the considerable amount of time and patience expended on his behalf proofreading this thesis. Personally, I am astounded at how he was able to do this on his very tight schedule. I am thankful to Donald and Massey University for allowing me to use and take home various FPGA development boards. I have appreciated all the advice given to me by Chris Johnston and Kim Gribbon during the year. Their ideas for GateOS, recorded in a conference paper (Bailey et al, 2006) have proved invaluable in formulating the requirements of GateOS and to some extent, its design. I would like to thank Massey University for allowing me to complete the research for my thesis using their equipment in the computer labs.

I also acknowledge the role of Xilinx in providing me with licences to the ISE suite of software tools, which were required for the implementation of GateOS.

Finally, I would like to thank Roger Gook of Celoxica who has provided me with licences for the DK suite of software, so that I was able to use Handel-C to implement GateOS. He has also been very helpful by providing me with the source code for various board level drivers.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES