A PROGRAMMING SYSTEM FOR THE IMAGINE MEDIA PROCESSOR

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Peter Mattson

March 2002

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Prof. William J. Dally
(Principal Adviser)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Prof. Dawson Engler

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

_____

Prof. Monica Lam

Approved for the University Committee on Graduate Studies:

_____

Abstract

Media processing applications, such as image-processing, signal processing, and graphics, motivate new processor architectures that place new burdens on the compiler. These applications demand very high arithmetic rates and data bandwidth, but lack data reuse. The Imagine Media Processor (Imagine) is a new architecture that introduces two innovations to meet these demands. To support a large number of arithmetic logic units (ALUs), it uses multiple register files connected to the ALUs by shared buses and register file ports. Instead of a cache, which relies on data reuse, it uses a large on-chip memory called the stream register file (SRF) to hold sequences of records called *streams*. The compiler is responsible for allocating the shared interconnect and SRF.

This thesis describes a programming system that enables efficient application development for Imagine and other architectures that include these innovations. It introduces an implementation of the stream programming model. The stream programming model divides a media processing application into a series if kernels, computation-intensive functions that operate on streams, and a stream program that defines the high-level control- and data flow between kernels. This thesis introduces KernelC, a C-like language used to write kernels. It describes a KernelC compiler that uses a new technique called communication scheduling to allocate the shared buses and register file ports and manage data flow between multiple functional units and register files. This thesis introduces a language extension called StreamC used with C++ to write stream programs. It describes a StreamC compiler that uses a new technique called stream scheduling to allocate the stream register file and determine when to load and store streams.

The Imagine programming system has been used to implement sophisticated, high-performance applications for Imagine such as stereo depth extraction, MPEG2 encoding, and polygon rendering. Experimental results for a set of image processing, signal processing, and graphics benchmarks show that communication scheduling delivers schedule lengths for an Imagine Media Processor with multiple partitioned register files and shared interconnect that are within 1% of the same architecture with one multi-ported register file that is estimated to be eight times larger. For these benchmarks, stream scheduling allocates the SRF as well or better than experienced Imagine programmers can by hand using assembly language.

Acknowledgements

This thesis would not have been possible without the support of many exceptional people. To them, my thanks:

My advisor, Professor Bill Dally, for a fascinating problem, good advice, constructive criticism, support, and flexibility.

The other members of my committee, Professor Monica Lam and Professor Dawson Engler, for their timely and constructive criticism.

The members of my research group for their persistent use of the Imagine programming system as it was being developed and their many suggestions on how to it could be improved.

The understanding people at Reservoir Labs for providing an interesting place to work, and for being patient.

Professor Yongmin Kim, for wonderful challenges which started me down this path as an undergraduate.

My family, without whom I would not be the person I am.

My wife, Allissa, for being a constant source of happiness and encouragement, and so much more.

Table of Contents

List of Figures