Chapter 1

Introduction

Computer vision, the understanding of images by a computer, is both an exciting component of artificial intelligence and a daunting computational task considered as one of the Grand Challenge problems of parallel processing. The design of a parallel computer for vision is an attractive problem for a computer architect, as the problem domain lends itself to optimizations that promise orders of magnitude performance over general-purpose parallel computer.

This thesis proposes to design, implement, and analyze a VLSI-based architecture tailored to the computing demands of early vision. Although the field has received a great deal of academic attention, this work targets a considerably higher speed design than previous efforts, and as a result encounter system-level consequences of a high speed parallel computer. In the body of this section we will the describe the problem domain and the detailed thesis approach.

Computing Requirements of Vision. Computer vision deals with the understanding of imagery by a computer, in contrast to image processing, which generally enhances images for human inspection. The field can be broadly categorized into three levels: low (or early) vision, which extracts physical properties of viewed surfaces, such as distance, orientation, and velocity; middle vision, which manipulates geometric quantities and performs simple recognition; and high level vision, which uses symbolic reasoning to disambiguate objects. Boundaries between each of these levels are rather blurred, especially as recent object recognition research has started to operate at the image level. Of the three, early vision processing is the best understood and therefore the best candidate for acceleration with a specialized architecture. The others are still highly experimental and require flexible general-purpose architectures.

The need for a specialized computer becomes apparent when the computational requirements of early vision are considered. A moderately sized image contains approximately 65,000 pixels, each of which may be used a hundred times in a computation. A single vision algorithm may require ten iterations per image, and as many as fifty algorithms may be required to extract essential information such as depth, motion, shapes, and shadows. All of this processing must be repeated thirty times per second to sustain real time response. The aggregate sustained computing power is therefore on the order of 100 billion operations per second (GOPS). It is clear that parallel processing is required to deliver this level of performance.

The question now becomes: what should the parallel system look like? There have been hundreds of proposed parallel system designs. These systems can be broadly categorized as Single-Instruction Multiple Data (SIMD) machines that consist of identical processing elements (PEs) executing the same instructions in lockstep, and Multiple-Instruction Multiple-Data (MIMD) machines, whose PEs execute independent programs and use explicit communication to synchronize operations. Historically, SIMD machines have been built with large numbers of simple PEs, while MIMD machines relied on a small number of complex PEs.

At the time of writing, the most prevalent parallel architectures are MIMD machines consisting of collections of workstation-class RISC processors. Indeed, only one major SIMD vendor, MasPar, still exists, while there are many MIMD vendors, including Cray, Convex, IBM, and Intel. This overwhelming commercial success is due to the constant improvement in microprocessor performance, driven by marketplace pressure and fueled by enormous investment.

Parallel MIMD machines are excellent general–purpose computing and development platforms, since they come equipped with the software inherited from their workstation heritage. As with the uniprocessors on which they are based, these machines excel at computations with irregular and unpredictable instruction streams. Unfortunately, the price of this flexibility is reduced performance. On problems with a data–parallel, highly regular structure, such as image processing, these processors continue to be unnecessarily flexible at every instruction step. When additional domain constraints such as low-precision arithmetic and 2-dimensional mesh communication are considered, a specialized architecture promises order-of-magnitude improvements over general purpose systems.

The proposed thesis develops a design for Abacus, a SIMD machine specialized for variable-precision integer operations on a dense 2-dimensional mesh, and targeted for a high-speed VLSI implementation. Predicted performance of Abacus parts is indeed approximately 50 times higher than that of uniprocessors with comparable silicon resources.

Reconfigurable Bit-Parallel Processing. The Abacus architecture can be classified as a reconfigurable bit-parallel (RBP) machine. Reconfigurable processors provide a smooth transition between the two extremes of a slow, small bit-serial PE and a complex, fast bit-parallel PE. The tradeoff between the two is that simple one-bit PEs are capable of arbitrary precision and can run at a high clock speed, but require many cycles per operation. Hardware bit-parallel PEs require few cycles, but run slower and waste valuable silicon when computing with data narrower than their ALU width. RBP designs retain bit-serial flexibility and bit-parallel performance.

The key idea of a RBP design is to provide programmable interconnected to link multiple PEs into a single Processing Site (PS) that can operate on several bits of a data word simultaneously. Reconfiguration allows precision to be matched to the need of each algorithm. Even more significantly, on-chip memory capacity at each processing site can also be adjusted by changing the number of PEs in a PS. Since off-chip memory references can be a factor of 20 slower than on-chip accesses, decreasing the miss rate at the cost of ALU capacity may lead to substantial performance increases for many algorithms. There is some overhead associated with RBP, such as the time to combine or distribute information from the entire word, as in conditional operations. As with hardware bit-parallel ALUs, operations on data narrower than the PS size (such as boolean flags) are inefficient.

Thesis Approach. This thesis proposes to analyze, quantify, and demonstrate the performance advantages of a high-speed VLSI RBP SIMD architecture on early vision tasks. The overall research approach will be to:

1. Develop an analytical performance model accounting for both architectural and algorithmic effects.

2. Design, implement, and test a VLSI chip containing Abacus processing elements. A working chip will both provide architectural parameters for an analytical model and unearth any inaccurate assumptions about high speed SIMD operations.

3. Implement a simulator and code a number of vision algorithms including a standard image understanding benchmark suite to determine algorithmic parameters.

4. Complete a system-wide (paper) design to ensure that there are no unexpected system-level bottlenecks.

5. Use the IC implementation experience and the analytical model to suggest a design for a second generation architecture.

The expected contributions of this thesis can be grouped under three categories: implementational, analytical, and cultural. A successful implementation will demonstrate a SIMD chip with the highest clock rate to date. It will force the development of system interfaces generally ignored in the academic field. This development is absolutely essential as input/output interfaces may prove to be the bottleneck in high speed designs.

The analytical model will allow the selection of an optimal point in implementation space as technology parameters vary. For example, novel high-bandwidth devices such as Rambus memory components may allow reduction in the amount of on-chip memory. The model will allow quantification of these tradeoffs. Also, in the course of obtaining algorithmic parameters, the vision algorithms will be systematically characterized, which may lead to improved intuition about optimizing algorithms for RBP architectures.

Hopefully, the thesis will also have several positive cultural effects. It will to provide a convincing demonstration of the effectiveness of SIMD processing. We believe important commercial applications of SIMD technology are near but may be hampered by the unconvincing performance and architectural deficiencies of existing SIMD processor arrays. Vision-related tasks, such as MPEG coding, OCR, paper document processing, digital picture manipulation, medical imagery, face recognition, vehicle collision avoidance, and automatic vehicle guidance all provide important practical applications which can be solved using SIMD techniques. An existing system design will allow the construction of a cheap tera-op level supercomputer for vision research. Such a platform will allow important image, signal, and text processing applications to be prototyped. A successful design will encourage research on specialized architectures, which has been decreasing recently due to the success of RISC-based MIMD machines. Finally, the project will demonstrate that high performance design can be done in a university environment.

Reconfiguration is the ability to dynamically configure memory organization, ALU width, or communication network topology to match the needs of the computational problem. *Parallel computers based on reconfigurable bit-slice architectures can be an order of magnitude faster on important applications than those based on conventional architectures.* This dissertation will validate the thesis with an implementation-based approach.

The proposed research will develop a reconfigurable architecture optimized for high speed VLSI, complete an aggressive implementation of the architecture, and evaluate the performance on a standard computer vision benchmark suite. Parameters from the implementation will be used to investigate an analytical performance model. The existence of a concrete instantiation of the architecture will allow empirical evaluation of the costs and benefits of reconfiguration.

The organization of the thesis essentially follows. Chapter 2 presents an overview of the substantial body of previous work relating to the topics of this thesis, from SIMD machines to modern architectural alternatives: the SRC PIM chip, Berkeley's PADDI-2 DSP, and FPGA based computing platforms. After demonstrating the the SIMD approach is competetive if the clock speed can be raised above previous levels, the discussion turns to issues of high-speed SIMD system designs. The chapter concludes with a review of related architectural performance evaluations.

Chapter 3 describes in detail the motivations for an RBP organization, RBP arithmetic algorithms, and specifics of the Abacus design.

Chapter 4 describes the implementation of the Abacus-1 chip at a fairly detailed level, as well as chip test results.

Chapter 5 describes the system-level design of the Abacus machine, including our approach to issues of high-speed SIMD operation.

Chapter 6 is the performance evaluation. It introduces the benchmark algorithms used to evaluate the architectural tradeoffs. Evaluates architectural tradeoffs such as local memory size, ALU width, off-chip memory bandwidth and network bandwidth based on instruction traces from the parallel algorithms.

Chapter 7 Describes the lessons learned from the design of the Abacus-1 architecutre, and the set of changes required to produce a smaller yet more powerful design. Looking farther ahead, it introduces modifications that allow a redesigned Abacus element to function in multiple-SIMD mode, as a systolic processor, or to emulate random logic circuits effectively.

Chapter 8 concludes with the implications of what has been learned from the design and suggests avenues for future research.