Chapter 1

Introduction and background

In this chapter, we describe (§1.1) and motivate (§1.2) the topic of this thesis, outline some of the history of the body of research upon which this work builds (§1.3), summarize the major contributions of this thesis (§1.4), and give a brief overview of the contents of the later chapters (§1.5).

1.1 What this thesis is about

This thesis is a detailed study of the advantages (and disadvantages) of the use of *reversibility* in computing. What do we mean by reversible computing? For our purposes, there are two important meanings:

Logical reversibility. First, a computational operation can be *logically reversible*, meaning that the logical state of the computational device just prior to the operation (its *input* state) is uniquely determined by its state just after the operation (its *output* state). Computing in a logically reversible fashion implies that no information about the computational state of the system can ever be lost; any earlier state can always be recovered by computing backwards from a given point. Another way to understand logical reversibility is that the system is deterministic looking *backwards* in time.

In chapter 10 we will see that logical reversibility, in itself, has some interesting computational applications. Chapter 9 will discuss how to program logically reversible computers. But the larger emphasis of this dissertation will not be on logical reversibility by itself, but on the benefits to be gained from using logical reversibility to enable another important kind of reversibility, namely, *physical reversibility*.

Physical reversibility. A *physically reversible* process is a process that dissipates no energy to heat, and produces no *entropy*. It seems that absolutely *perfect* physical reversibility is technically unattainable in practice in a complex, controlled dynamical system, simply because there will always be some nonzero probability for a random event to occur (*e.g.*, the impact of a cosmic ray, or an asteroid) that is sufficiently energetic that it will interfere with even the most carefully-controlled and well-isolated of systems. Nevertheless, physical reversibility is a useful concept, because (as we will see in ch. 7) even with present-day electronic technology, we can already make logic devices that are *almost* physically reversible, and we do not yet know of any fundamental limits to how close we can get to perfect reversibility, as technology improves.

As we will see in §2.5, logical reversibility is necessary in order to approach complete physical reversibility. Chapters 6 and 7 will focus on the study of computing devices that are *both* logically and physically reversible, and on their resulting implications for the potential efficiency of computation. Usually when we speak of reversibility in this thesis, we will be referring to this combination of logical and physical reversibility, rather than to just logical reversibility by itself.

1.2 Motivation

Why study reversible computing? Aside from pure academic interest, we feel that the study of reversible computing can be motivated in a fairly strong way, in terms of the long-term goals of society in general and the field of computer science in particular.

For the productivity of society, and the growth of the economy, efficient information processing is critical. A relatively small improvement in the speed and power of computers facilitates progress in virtually every industry. The great value of information processing has motivated the enormous technological investments fueling Moore's law, the trend of exponential improvement in computer speed and cost-efficiency that has been maintained over the last half-century.

It is in society's interest that computer technology continue to improve rapidly for as long as possible. Therefore, it is important to identify various potential obstacles to further improvements far enough in advance so that the research community has time to develop solutions before such an obstacle has a chance to stall the rate of progress. Or, if some truly insurmountable barrier to further improvement can be identified early on, at least society will have time to prepare for the consequences.

The semiconductor electronics industry is well aware of a variety of potential obstacles to further improvements of its technology over the next 10 to 15 years [145]. Even if these obstacles are overcome, we can expect that eventually a point will be reached where it is technically or economically impossible to refine semiconductor technology further. At that point, perhaps alternative computing technologies will eventually emerge and supersede semiconductors. (We discuss several potential alternatives in ch. 8.)

However, in the longer term, we can foresee a variety of more fundamental limits to the improvement of computer technology, limits that are qualitatively independent of the particular technology used (such as semiconductors), and whose existence depends only on well-established fundamental laws of physics. These fundamental limits will become increasingly important as computer technology improves, whatever path it takes. If we can, right now, identify some techniques that will allow technology to perform as well as possible given these ultimate physical limits, then we will be well prepared to cope with these limits once they become dominant concerns in computer engineering. (We discuss this research philosophy in more detail in ch. 5.)

Not to keep the reader in suspense, one fundamental physical limit, known since at least 1961 (Landauer, [97]), is that for every bit's worth of computational information that is discarded within a computer, at least one bit's worth of new physical entropy must be generated. Moreover, due to basic thermodynamic principles, this entropy cannot simply be destroyed, but must instead be physically moved out of the computer, if one is to keep the machine from eventually overheating. (We will explain these constraints in more detail in chapter 2.)

In chapter 6 of this thesis, we establish that in order for a scalable computer architecture to be as efficient as possible in the face of these constraints, the machine must contain the capability to perform computations in a logically and physically reversible manner, which minimizes the production of unnecessary entropy, and the overhead of its removal from a densely-packed machine. This suggests a framework for algorithm design in which information is considered as a *conserved* material-like thing, embedded in 3-D space. As we will see, this is what information really is like. The expert programmer should not mind expanding his expertise to working with such a model, because it allows designing the best algorithms that are physically possible.

The capability of reversibility is completely lacking from today's processor designs. But the technology now exists to remedy this situation, and Part II of this thesis discusses how to design and program machines that use reversibility to achieve asymptotically optimal efficiency. The high-level concepts of reversible circuits in chapter 7 are described in terms of existing semiconductor technology, but are not dependent on it: they can be applied equally well to a wide range of future logic-device technologies that might emerge.

Near-term benefits. Present-day technology is far from the fundamental physical limits of computation, but reversibility offers some of the same benefits today that it will offer in the limiting technology. We now know how to build approximately physically reversible computers using today's electronic technology. These techniques may have benefits in the near-term, in applications where energy dissipation is of paramount importance (see §7.10). There may even be some near-term uses for logical reversibility by itself, regardless of physical reversibility, as discussed in chapter 10.

But one must be careful: chapter 3 reveals some of the theoretical inefficiencies incurred when using logical reversibility by itself, in situations where saving energy and minimizing entropy production are unnecessary.

In summary, motivations for studying reversible computing include both short and long-term applications; the long-term ones being more fundamental. The major motivation lies in the economic value of making computers more efficient through reversibility, under any of a variety of measures of efficiency that are influenced by energy dissipation. This thesis focuses on exploring how this can be done.

1.3 Brief history of reversible computing

In this section we briefly summarize some of the history of reversible computing research. This is not intended to be a complete account. Some additional historical information about particular sub-areas will be provided in later chapters. A more comprehensive review of the early history of part of the field is provided in Bennett 1988 [18].

1.3.1 Early thermodynamics of computation

The study of thermodynamically and logically reversible computational processes has historically been motivated by concerns in fundamental physics. For example, the proper resolution of the famous "Maxwell's Demon" paradox of thermodynamics (see the papers in [100]) required understanding that the means of disposal of unwanted information can be important when considering the thermodynamics of a system.

The first connection between computation and fundamental thermodynamics was apparently made by John von Neumann ([182], p. 66). In a December 1949 lecture at the University of Illinois, he reportedly performed a calculation of the thermodynamical minimum energy that is dissipated "per elementary act of information, that is, per elementary decision of a two-way alternative and per elementary transmittal of 1 unit of information." He quantified this energy as $k_{\mathrm{B}}T\ln\mathcal{N}$, where k_{B} is Boltzmann's constant, T is the temperature, and $\mathcal{N}=2$ is the number of alternatives to be decided between. Unfortunately, there is apparently no existing complete record of this lecture, or of any corresponding written analysis by von Neumann, so it is difficult to determine exactly how he explained this analysis, how seriously he took it, and whether it was actually original to him.

Rolf Landauer (1961, [97], §4) was apparently the first person to explicitly state the argument establishing that the irreversible erasure of a bit of computational information inevitably requires the generation of a corresponding amount of physical entropy (namely 1 bit $= \ln 2$ "nats" $= k_{\mathrm{B}}\ln 2 \approx 9.57\times10^{-24}$ J/K). In that paper, Landauer also recognized that reversible operations need not incur such dissipation, and that any irreversible computation can be performed via a sequence of reversible operations by saving a history of all the information that would otherwise be irreversibly dissipated. However, Landauer then proceeded upon the mistaken assumption that the space occupied by this history record would have to be irreversibly cleared in order to be reused, and concluded that therefore, reversible operations could not avoid the fundamental unit dissipation incurred by each computational step, but could only postpone it until the memory needed to be reused. To his credit, Landauer realized that this argument was not rigorous, and did not present it as such.

1.3.2 Development of reversible models of computation

Landauer's error was not caught until Charles Bennett (1973, [16]) discovered that the reversibly-recorded history of an irreversible computation could also be cleared in a logically reversible way, leaving only the input and the desired computational output in memory. This refuted Landauer's argument that each useful computational step must incur, in the long run, at least about $k_{\mathrm{B}}T$ energy dissipation. With Bennett's trick, the amount of memory that would need to be irreversibly cleared between runs could be smaller, by an arbitrarily large factor, than the number of useful irreversible computation steps that are reversibly simulated during the course of the computation.

Bennett described his technique using a formal Turing machine model, but later researchers showed that Landauer's trick of recording a history could also be applied to permit other models such as cellular automata (Toffoli 1977 [160]) and logic circuits (Toffoli 1980 [161], Fredkin & Toffoli 1982 [74]) to operate reversibly as well. Indeed, the Laundauer/Bennett techniques seem to apply generally to "reversiblize" any model of computation.

1.3.3 Development of physically reversible logic devices

However, showing that logically irreversible operations can be avoided in useful computations is only part of the problem of demonstrating that reversible computing can save energy. The other part requires showing that physically reversible primitive logic devices can actually be built. Bennett's 1973 paper [16] suggested the possibility of an enzymatic reversible computer using biomolecules, and in later papers such as (1982, [17]) he described a clockwork mechanical Turing machine powered by Brownian motion. Meanwhile, Fredkin and Toffoli had described an electronic implementation (1978, [73]), and an idealized model based on the ballistic motion of rigid spheres (1982, [74]), which we will describe in more detail in §7.7.1, p. 201. Konstantin Likharev showed in 1982 [108] that superconducting Josephson junction circuits could be used to compute in a reversible fashion.

Later reversible device proposals (see ch. 8) include various mechanical and electronic proposals by the pioneering molecular nanotechnologists Drexler and Merkle (Drexler 1992 [51], ch. 12; Merkle 1993 [123, 124]; Merkle & Drexler 1996 [126]), and a single-electron system analyzed by Likharev and Korotkov (1996, [111]).

So at present, there is no shortage of reversible device ideas. Moreover, in the years since Fredkin & Toffoli's 1978 proposal [73] it has become quite feasible and economical to build reversible devices using conventional VLSI electronic fabrication techniques (*cf.* Athas *et al.* 1994 [5], Younis & Knight 1994 [193]); we will review those developments in more detail in chapter 7.

1.3.4 Previous reversible computing theory

Independently of the type of reversible devices that are used, there are algorithmic issues involved in performing large computations using logically reversible primitives. For example, Bennett's original reversible simulation technique is limited by the fact that the algorithm requires an amount of temporary storage space that is proportional to its run-time. In contexts where digital storage is expensive and energy is cheap, one might do better by just discarding the bits instead.

So, in 1989, Bennett developed a more space-efficient version of his algorithm [19]. Unfortunately, it incurs a polynomial slowdown factor that cannot be made arbitrarily close to linear without making the space usage exponentially large (Levine and Sherman 1990 [103]). Similarly, in 1997, Lange, McKenzie, and Tapp [98] gave a general algorithm for reversible simulation of irreversible computations using *no* extra space, but with exponentially inflated run-times. It remains an important open problem to prove whether or not there is a *single* reversible simulation technique that incurs overheads in *neither* space nor time, but, as we will prove in §3.4, any such technique cannot be totally general, in the sense of applying to any conceivable model of computation.

1.3.5 Optimal scaling of physical machines

This thesis takes the study of reversible computing beyond the traditional focus on devices and classical complexity theory; chapter 6 introduces a new area of study, namely of how reversibility affects the scaling behavior of the most powerful physically possible computers, based on fundamental physical arguments.

The optimal scaling of computation within physically realistic constraints is an issue that has been studied previously (*cf.* Vitányi 1988 [180], Bilardi & Preparata 1993 [24], Smith 1995 [152]), but never before with particular attention to how the reversibility of physics allows reversible computation to improve physical scaling behavior. The research reported in this thesis is, to our knowledge, the first work that explores this new angle.

1.3.6 Programming reversible machines

We will save our review of the history of this area until chapter 9.

1.4 Major contributions of this thesis

The primary novel, original contributions of this thesis are the following:

- **Chapter 2** gathers together and presents in an organized form a variety of known fundamental physical constraints on information processing, that are expected to apply to any physically possible computing technology, at least in the non-relativistic regime. We conjecture that this is the first such listing that is sufficiently complete that it encompasses all the fundamental physical constraints (within that regime) that determine the maximum asymptotic scalability of computers and algorithms.

- **Chapter 3, section 3.4** (work done with Josie Ammer) underscores the overheads for reversibility in traditional models of computation by proving, for the first time, that any completely general transformation of irreversible machines to reversible ones must sometimes increase either the asymptotic computational time or space requirements for solving some problems. It gives lower bounds on the amount of increase required. The proof applies to cases where there is reversible access to an external black-box ROM or oracle. It is conjectured to also be true for pure models with no external black box. The proof might be extensible to that realm if it assumes that one-way functions exist, as is frequently assumed in cryptography.

- **Chapter 5** presents a novel physically-realistic model of computation (the R3M or "reversible 3-D mesh") and conjectures a "tight Church's thesis" claiming that this model is asymptotically as powerful as is physically possible given the constraints from ch. 2, within a constant factor.

- **Chapter 6** proves that the proposed R3M model is asymptotically strictly more powerful than any irreversible model of computation, by small polynomial factors in the machine size. Specifically, reversible machines of physical diameter D are shown to be asymptotically faster than diameter-D irreversible machines, by a factor of $\mathbf{\Theta}(\sqrt{D})$. Also, reversible machines of mass M are both faster and more hardware-efficient than mass-M irreversible machines by a factor of $\mathbf{\Theta}(\sqrt[18]{M})$. These bounds are shown to apply to a wide class of parallel computations that require sufficiently tight communication, but that need not be inherently reversible.

I consider the previous item to be the central, most important contribution of the thesis.

- **Chapter 8** uses the scaling results from ch. 6 together with parameters of present-day and proposed future technologies to show that with present-day technology, reversibility becomes advantageous at a reasonable scale, and in future technologies, it will be advantageous at just about any scale.

- **Chapter 7, section 7.6** does some novel analysis showing how to choose speeds, voltages and temperatures so as to minimize energy dissipation in one form of reversible electronics.

- **Chapter 7, section 7.7** and **appendix A** present the design (for which I was primarily responsible) of the world's first ever fabricated reversible parallel processor, which in principle obeys the scaling results of chapter 6 and thus is asymptotically faster than all previous parallel processing architectures, which are irreversible.

- **Chapter 9** and **appendices B through E** present examples of reversible instruction sets, programming languages, and algorithms. Similar efforts have been undertaken before by other researchers, so this area of contribution is not completely novel. However, much of my work proceeded independently of the earlier efforts. This reinvention helps underscore my point that reversible programming concepts are not difficult to master.

That completes our summary of the major contributions of the thesis. We will revisit this list once again in chapter 11.

1.5 Overview of thesis chapters

Here we summarize the contents of the various chapters of this thesis.

Chapter 2 surveys what is currently known about the fundamental constraints that known physics places on the potential capabilities of computing systems. We describe limits on the speed at which information can travel, the density at which it can be stored, and the rate at which it can cross a surface. We also review recent fundamental limits from Margolus and Levitin (1996, [118]) on the rate at which a computer can change state. We discuss the meaning and the computational implications of physical reversibility and the second law of thermodynamics.

Chapter 3 examines various formal theoretical models of reversible computing, and describes all the known results in the area. Then the chapter focuses on proving an important new conjecture in the theory of reversible computing: namely that in ordinary, nonphysical models of computation, imposing reversibility on the model must cause either the space or time complexity of some problems to increase. We prove that the conjecture is indeed true in a model of computation that invokes a contrived (but computable) oracle, and we establish lower bounds on the resulting increase in complexity. This proof implies that if there is an algorithm for simulating irreversible machines on reversible ones with perfect efficiency, then that algorithm cannot be totally general (relativizable to all oracles), in contrast to all the reversible simulation algorithms that are known currently.

Chapter 4 reviews the possibility of computation using large coherent superpositions of states (quantum computation). Quantum computers are inherently reversible.

Chapter 5 introduces the concept of "ultimate" physical models of computing, which are designed to accurately capture the true asymptotic complexity of all computational problems under the laws of physics. Then the chapter outlines the form that we will argue such models must take—namely, some sort of three-dimensional mesh of potentially reversible processors.

Chapter 6 discusses how the use of reversibility affects the scaling behavior of computers in several important respects. Due to their unavoidable generation of entropy which must be removed, irreversible computers turn out to ultimately be limited to processing rates that are only proportional to their surface area. In contrast, if a computer uses devices that are reversible, even in a limited sense that takes frictional effects into account, then it can perform $\boldsymbol{\Theta}(\sqrt{d})$ times more operations per second within a physical space of diameter d. Even if we do not constrain the physical area of the computer, but only its mass (number of processors), reversible computers are still faster at some problems by a factor that grows as $\boldsymbol{\Theta}(\sqrt[18]{n})$ where n is the number of processors.

Chapter 7 describes and analyzes in detail some known reversible circuit technologies, how they perform as various parameters are scaled, how they compare to traditional circuits, and how to design processors based on these techniques that realize the scaling benefits described in the previous chapter. We describe a very simple example of such a processor that we designed.

Chapter 8 reviews a variety of advanced logic technologies that have been proposed for use when the limits of traditional VLSI are reached. Then, we use our scaling results from chapter 6, together with parameters of the proposed technologies, to show that if we assume reasonable limits on future cooling systems, then any computers of macroscopic size that are built using these future technologies will be considerably faster if their logic elements are operated reversibly.

Chapter 9 illustrates in detail how to program reversible computers.

Instruction sets. We start with a description of some properties that a good reversible microprocessor machine instruction set needs to have, and how we achieved these properties in our group's Pendulum instruction set architecture.

High-level languages. Next we describe important issues in the design of high-level programming languages for reversible processors. Special programming languages are required in order to permit optimum efficiency on reversible processors. We describe the simple reversible programming language "R" which we designed and wrote a compiler for.

Algorithms. Finally, we describe some good reversible algorithms for a number of problems, including sorting, searching, arithmetic, matrix operations, graph problems, and simulations of physical systems.

Chapter 10 briefly discusses some potential alternative applications for reversible computing, aside from the energy dissipation issues. These include applications in hardware error detection, protecting against accidental or malicious data destruction, program debugging, transaction processing and database rollback, and speculative execution in multiprocessors.

Chapter 11 summarizes the progress in reversible computing achieved in the thesis, and points out the main areas where future work is needed.

Appendix A shows circuit schematics and VLSI layouts for the proof-of-concept parallel reversible processing element we describe in chapter 7.

Appendix B gives program-level specifications for PISA, the instruction set architecture for PENDULUM, our group's reversible RISC processor design.

Appendix C gives a complete account of "R," the simple C-like reversible programming language we developed.

Appendix D describes our compiler, written in Common Lisp, which translates R source programs into reasonably efficient PISA assembly code.

Appendix E gives the detailed derivation and code for our reversible program for simulating the Schrödinger wave equation of quantum mechanics (our illustration of an efficient reversible physical simulation).

Appendix F gives tables of mathematical units, constants, and notations used in the text, for easy reference.

1.6 Overall message of thesis

The overall message of this thesis is that (1) reversible computing techniques are not very different from or more difficult than ordinary computing techniques, and (2) they will definitely be a necessary part of the long-term future of computing.

It is hoped that this thesis will help to convince the larger computing community of these very important points, and thus help to spur further research in this field.