Chapter 11

Conclusion and Future Work

In this chapter we summarize our conclusions and point to the important areas for future research along the lines of this thesis.

11.1 Summary of Contributions

In this thesis, we have attempted to comprehensively survey the entire range of current knowledge about reversible computing techniques, with an emphasis on the use of these techniques to make computers more efficient in a variety of ways. In the course of my own research in this area, I have discovered a substantial number of original and very interesting results, the most important of which we summarize here:

- In the context of *traditional* models of computation, purely reversible models appear to be asymptotically sub-optimal when both time and space costs are considered. (§3.4)

- However, given our understanding of the fundamental constraints and opportunities for information processing implied by well-established laws of physics (chs. 2,4), it becomes clear that those traditional models of computation lead to scaling predictions that are either suboptimal or physically unrealistic. Based on the constraints of physics, one can conceive of an *ultimate* physical model of computation that gives exactly the asymptotically best scaling for all problems that is permitted by physics. We conjecture that the ultimate model must take the form of some class of reversible 3-D mesh (either time-proportionately reversible, ballistic, or quantum coherent). (Chapter 5)

- We show that any of the proposed varieties of reversible 3-D mesh are asymptotically *strictly* faster per unit area or per unit mass than *any* irreversible physical computing architecture. The advantage per unit area grows with the square root of the reversible mesh thickness. However, the advantage per unit mass of the non-quantum approaches is only a very small polynomial, growing at best only with the 18th root of the number of processors. (Or 9th root, if ballistic computation is possible.) (Chapter 6)

- We demonstrate that an asymptotically optimal reversible 3-D mesh could be built today using existing commercial VLSI processes, by giving a complete circuit design for a proof-of-concept universal reversible mesh processing element (FLATTOP). Unfortunately, we also find that despite the superior asymptotic scaling, reversible processing in VLSI is not competitive for supercomputing applications at feasible cost levels. Tipping the scale would require the development of much lower-resistance transistors. (Chapter 7)

- Fortunately, in the long term, a wide variety of advanced superconducting and nano-scale logic device technologies have been proposed that would not only be much faster than traditional VLSI, but would also entail much larger advantages for reversible processing, since they are much more nearly ballistic than are systems based on highly-resistive MOSFETs. Only a very thin layer (microns to millimeters thick) of these future reversible devices would be needed in order for the resulting machine to be faster per unit area than *any possible* irreversible technology of *any thickness*. The advantages would continue to increase with the square root of further increases in machine thickness. (Chapter 8)

- In order to take full advantage of the long-term efficiency benefits offered by reversible computing, it is required to use new microprocessor instruction sets and new high-level programming languages, in order to permit the optimal reversible algorithms to be expressed and executed with their intended efficiency. However, the necessary changes are fairly straightforward. We presented some important principles for reversible instruction sets, and a simple proof-of-concept C-like programming language. Algorithms must in general also be redesigned to take best advantage of the reversible paradigm. We give examples, including an efficient reversible algorithm for simulating quantum mechanics. (Chapter 9)

- In addition to its thermodynamic advantages, pure reversible computing may conceivably have applications in other areas such as audit trails, transaction rollback, and backwards debugging. However, after considering a number of such possibilities, we have not found any very convincing benefits in areas other than in increasing computational efficiency. For other applications, there seem to be equally good solutions that do not require complete reversibility. (Chapter 10)

The upshot of all this is that substantially reversible computing techniques do not seem to be immediately practical through the next 10 years or so of semiconductor technology, but in the much longer term, as we move to a nano-scale computing technology, the issues and techniques described in this thesis will be not only practical, but essential in order to make good use of the physical resources available for computing. We can confidently predict that meter-scale and larger machines composed of good nano-scale reversible components will be much faster, at many types of problems, than *any physically possible* mostly-irreversible computer of *any size*. Designing and programming these superior machines will require processor architectures and algorithms along the lines we have discussed.

Although the manufacturing technology does not yet exist to produce many of the proposed future logic devices, with a bit of optimism, one can look at the present rate of progress of technology, and see that it is at least a fairly good bet that at some point, probably a few decades hence, that technology will exist, and we will need to build computers with it. Based on that projection, it is worthwhile for computer scientists to start thinking now about how to architect and program those machines, and for device physicists to start thinking about how to improve their components with the future reversible revolution in mind.

It is hoped that the research reported in this thesis will serve as an impetus to that work, and a guide for future researchers starting out in this important area.

11.2 Major areas for future research

The future research that will be needed in reversible computing spans a wide variety of disciplines.

Fundamental theoretical physics. In chapter 2 we saw that among the ultimate physical limits on computation, the fundamental limits on entropy density (and entropy flux) are currently not very well understood. (At least, I have not yet encountered a definitive statement of them.) For example, it is not clear to me what is the maximum density with which entropy can be stored within normal atomic matter at manageable temperatures, or whether there are other types of matter that might achieve higher densities and still be useful. Entropy density limits determine entropy flux limits, and thus are important for understanding the limitations of cooling systems.

Another important area for theoretical physics is in further elucidating the physics relating to quantum computing, devising better ways of avoiding decoherence, and so forth.

Of course, it would also be nice to have a simple, complete, unified theory of physics, rather than the somewhat incomplete picture we still have. It is possible that with a complete theory, we might see some surprising new implications for the fundamental limits of computation. Of course, pinning down a complete theory is the holy grail of theoretical physics, and physicists are already hard at work on this problem.

Nano-scale device physics. There is of course also much need for work on designing new physical devices that can be used as computational elements at the nanoscale, devices that have the appropriate physical properties so that they can be operated quickly in time-proportionately reversible fashion, with an entropy coefficient that is low enough so that the devices are nearly ballistic. (One would like to not have to start clocking the devices more slowly than their maximum speed until a very large scale of machines.)

More difficult than just designing new devices is designing *buildable* devices, or more broadly, designing economically feasible pathways for the development of future manufacturing technology that will eventually lead to the ability to build the desired devices. Fortunately, the entire field of nanotechnology (*cf.* [51, 41], and the journal *Nanotechnology*) is already working hard in this direction, since it can be seen that a flexible nano-scale manufacturing infrastructure would benefit society enormously in ways that go far beyond just making faster computers.

Semiconductor device physics. Meanwhile, one direction to try to make effective reversible supercomputers would be to try to find ways to make lower-resistance switches using more conventional fabrication technology. We saw that low-resistance switches benefit adiabatic circuit techniques because they would decrease entropy coefficients. However, they do not benefit traditional irreversible circuits as much because the CV^2 switching energy is independent of switch resistance, and so a dissipation-limited system will not go any faster beyond a certain point no matter how fast its switches are. One promising approach to making lower-resistance switches might be to make smaller, faster micro-electromechanical relays. Other approaches might be possible.

Resonant power supplies. Another element that would be needed for good, scalable adiabatic circuits is a good resonant power supply. Currently, we do not know of a resonant supply technique that has a high Q, the desired scaling properties, and can provide the desired waveforms. Becker and Knight's technique [12, 13] comes close, but the scaling seems not quite right. It might be that the desired scaling isn't possible, in which case adiabatic circuits overall might not scale as well asymptotically as reversible circuits based on other types of devices.

Adiabatic circuit designs. If lower-resistance switches and good resonant power supplies can be found, the adiabatic circuit approach to reversible computing might become attractive enough so that it is worth some effort to try to find an adiabatic logic circuit family that is simpler than SCRL, while still providing all of SCRL's desirable features. SCRL already seems pretty good, but a further improvement by a small constant factor might still be possible.

Reversible architectures. Independent of the precise logic technology used, work can already be done on designing better reversible processor architectures. Our proof-of-concept FLATTOP chip was not designed to be easy to program. Vieri's Pendulum architecture [179, 178] is much better, but further improvements in the efficiency of the design are likely possible. For use in parallel mesh architectures, a processor including hardware support for routing might be desirable.

Reversible programming languages. Our proof-of-concept R programming language was intentionally extremely simple. It would be interesting to have reversible versions of more sophisticated programming languages with more advanced features. However, one should be careful when choosing a programming language that it does not necessarily impose any asymptotic inefficiencies compared to programming in raw machine language. Constant-factor overheads are acceptable, but if the language is asymptotically inefficient, then to some degree it ruins the point of developing a reversible architecture to begin with.

Mesh programming languages. Traditional programming languages are tied to the idea of a uniprocessor architecture, and do not help much in expressing effective parallel algorithms. In order to concisely express reversible 3-D mesh algorithms, it would be nice to have a language in which one could express algorithms at the level of specifying where information is located in 3-D space, how it should flow around the system as it is computed and uncomputed, and so forth.

Given the ultimate physical nature of information, we suspect that a good parallel programming language should have a bit of the flavor of a language for specifying the detailed design and layout of a complex assembly line in a factory, in which physical objects (data structures) are assembled and disassembled (computed and uncomputed), and must physically move from place to place, at bounded speeds (no more than c), without ever occupying the same space as each other (consuming more memory than is available at a given node). Moreover, like in a factory, one has to route power to the subsystems, and provide ventilation and cooling systems to remove unwanted heat (entropy)—except that in a reversible mesh architecture, the entropy could be moved out of the system while it is still in digital form, if this is beneficial. Moreover, the production of unwanted entropy could be kept to a minimum by uncomputing objects that are no longer needed, rather than, say, "melting them down" to the "raw material" of free memory—which is an appropriate physical analogy for what we do when we overwrite memory that contains a data object.

Unlike in a factory, however, the structure of the manufacturing machines (programs) in the mesh computer would be relatively easy to reconfigure dynamically, under program control. They could replicate themselves at will, and move around freely from one part of the factory to another. At the programming level in the computer, one need not worry about factors such as gravity, strength of materials, vibrations, and wearing out of parts. (Hopefully, the bottom-level logic devices are sufficiently reliable enough and/or the architecture fault-tolerant enough that the programmer does not have to worry about failure of the underlying physical components.)

Reversible manufacturing. The above analogy between computing and manufacturing also suggests the intriguing idea that in future nanomechanical factories, one might also apply reversible computing principles to manufacturing. In a real nano-scale factory that is producing nano-scale components, from time to time, temporary structures may need to built. The system will be able to operate with less heat dissipation, and thus faster overall, if it uses a knowledge of how a temporary structure was built when disassembling it. If the structure of an atomically-precise object is known precisely, it can theoretically be disassembled to raw materials with no production of entropy. In contrast, disassembling the temporary object using a general approach with no knowledge of the object's structure is bound to be wasteful.

An analogy from everyday experience: when the face of a tall building is repaired, often the contractor will construct a temporary scaffolding, in a regular structure, to aid the work. When the work is finished, they carefully unscrew the pipes making up the scaffolding, in an appropriate order, dissassembling the scaffolding piece by piece. An alternative approach would be to just knock down the whole scaffolding, break it into pieces, melt it down, mold new pipes, and rebuild it next time from scratch. But of course this would be much less energy-efficient.

The same basic principle could be applied in manufacturing at all scales. For very fast nano-scale systems where heat removal is a major concern, this principle would be essential for allowing as many nano-manufacturing operations as possible per unit time, per unit of outer surface area.

Reversible mesh algorithms. Finally, even given a good programming language, one is still left with the task of designing good algorithms for all the computationally expensive problems that one encounters. As we saw in ch. 9, in general the best reversible algorithm for a problem might not be derivable from the best irreversible algorithm in a straightforward way. The same applies to mesh algorithms. If we wish to find the truly best algorithms for solving very hard problems, we should be working on algorithms for reversible 3-D mesh architectures. (And as we saw in ch. 4, perhaps the algorithms should be quantum coherent as well.) It would be very interesting and useful to compile a catalog of the best classical (and quantum) reversible 3-D mesh algorithms that can be devised for a variety of the hard, data-intensive problems that have historically motivated the development of massively parallel supercomputers. That way, once we have developed the technology to start building computers based on high-quality reversible nano-scale components, we will already have a good idea of how to use them effectively.

11.3 Final words

In closing, we hope that this thesis has shed a revealing new light on an fascinating area of computer science, reversible computing, an area that has previously been somewhat obscure, and not very well-understood. We hope that this work will contribute significantly to an eventual consensus in the computing community that it is worth the effort to design and build new types of nano-scale physical logic devices with reversible usage in mind, and to develop and study reversible algorithms to run on the massively parallel reversible meshes that we will someday be able to construct.

It is our fervent hope that such a consensus will arrive sooner rather than later, in order to facilitate a more rapid evolution towards the reversible computing revolution that must eventually occur. By enabling vastly more efficient computing, reversible computing techniques should greatly facilitate many amazing feats of technological and social progress that we expect and hope our civilization will accomplish, over the course of the new millenium that is about to dawn.