Chapter 5
Discussion and Conclusion

The preceding chapter shows how the Hopfield neural network and Boltzmann machine can be modified to solve the shortest path computation problem. It has introduced and utilized two neural shortest path computation algorithms from each type of neural network. These algorithms can be incorporated in ATM network for optimizing the computation of the shortest path.

5.1 Conclusions

This study has examined the Hopfield neural network and the Boltzmann machine for shortest path computation. The shortest path computation is not only used for routing purpose in computer network; it can also be used in VLSI circuit design. In the case of routing in computer network, the shortest path computation can be used for route determination in ATM PNNI network. The neural algorithm for shortest path computation is specially built for the route determination model in PNNI switching system (refer to the PNNI switching system reference model in figure 2.8). The neural algorithm for shortest path computation calculates the shortest path, based on the link cost generated from the data stored in the topology database. Since this study focuses on the shortest path computation, the link cost generation is eliminated.

The major problem with Hopfield neural network is setting the correct parameters. Simulation result shows that the parameters are optimal for shortest computation in a 5-node computer network but these parameters may not be optimal for shortest path computation in a larger computer network.

The Boltzmann machine was built using the concepts from [29]. The problem with Boltzmann machine is to find the relationship of the following:

1. The weight between neurons in the same row.
2. The weight between neurons in the same column.
3. The weight between neurons in different row and column.
4. The self-connected weight.

In this study, each of the above four types of weights were defined. The quality of the result is highly dependent on the relationship between the above four types of weights. The stopping condition of the Boltzmann machine may also need to be modified in order to achieve better results.

In conclusion, the comparison of simulation results for simulation model HOPFIELD1 and BOLTZMANN1 in shown in the following table:

Table 5.1: Comparison of simulation results between HOPFIELD1 and BOLTZMANN1

Possible Path	Total Cost	HOPFIELD1	BOLTZMANN1
0-1-4	1.402130	0%	14%
0-1-2-4	2.041090	0%	13%
0-1-3-4	1.805823	0%	15%
0-1-2-3-4	2.620460	0%	14%
0-2-4 (the shortest path)	1.308245	100%	20%
0-2-3-4	1.887615	0%	24%

Since link costs in the simulation model of HOPFIELD2 and BOLTZMANN2 are randomly generated, the shortest path is different in each iteration of the shortest path computation. The percentage of getting the shortest path in HOPFIELD2 in 100 time of shortest path computation is 100%, whereas is only 19% for the BOLTZMANN2 to get the shortest path.

5.2 Further Study

The research here can be further investigated by considering various aspects as stated below:

- *Cost function*: the cost function is to generate link cost for each link. Since each link is described by several parameters stored in the topology database, thus the purpose of the cost function is to generate a single value to describe the link. There might be several types of cost functions for difference traffic classes and each type of the cost function might not use all the parameters to generate link cost.
- *The Energy function*: the energy function in this study is derived from [28]. Other terms that further refine the requirement or constraint may be added into the energy function to express the shortest path problem.
- *Kohonen neural network*: the Kohonen neural network is another type of neural network that can be used for calculating the shortest path. Unlike Hopfield network and Boltzmann Machine, Kohonen neural network utilized its self-organizing feature to solve the shortest path routing problem.