Deep Blue

Deep Blue is the chess machine that defeated then-reigning World Chess Champion Garry Kasparov in a six-game match in 1997. There were a number of factors that contributed to this success, including:

- a single-chip chess search engine,
- a massively parallel system with multiple levels of parallelism,
- a strong emphasis on search extensions,
- a complex evaluation function, and
- effective use of a Grandmaster game database.

This paper describes the Deep Blue system, and gives some of the rationale that went into the design decisions behind Deep Blue.

Deep Thought 2 had four improvements: Medium-scale multiprocessing, Enhanced evaluation hardware, Improved search software and Extended book.

Deep Blue I *based on a single-chip chess search engine lost to Kasparov.* Deep Blue II was designed with a significantly enhanced, chess chip with a completely redesigned evaluation function, going from around 6400 features to over 8000. Secondly, more than double the number of chess chips in the system, and use the newer generation of SP computer to support the higher processing demands thereby created. Thirdly, development of a set of software tools to aid in debugging and match preparation *Subsequently - Deep Blue defeated Garry Kasparov in the 1997 match by a score of 3.5–2.5.*

Deep Blue is a massively parallel system designed for carrying out chess game tree Searches organized in three layers. One of the SP processors is designated as the master, and the remainder as workers. The master searches the top levels of the chess game tree, and then distributes “leaf” positions to the workers for further examination. The workers carry out a few levels of additional search, and then distribute their leaf positions to the chess chips, which search the last few levels of the tree.

There were challenges observed during the quiescence search, iterative deepening, transposition tables and NegaScout implementations - Large searching capacity, Hardware evaluation function implementation, Hybrid software/hardware search, Massively parallel search.

Deep Blue is a massively parallel system, with over 500 processors available to participate in the game tree search. The chess chip divides into three parts: the move generator, the evaluation function, and the search control -

- move generator - A reasonable move ordering, preferably as close to best-first as possible, is an important consideration for efficient search in game trees.
- evaluation function - there were two implementations “fast evaluation” and a “slow evaluation”
- search control - is a state machines to implement null-window alpha-beta search

SW Search we designed on these principles – Extend forcing/forced pairs of moves, Forced moves are expectation dependent, Fractional extensions, Delayed extensions, Dual credit and Preserve the search envelope.

The following *Credit generation mechanisms* were introduced - Singular, binary, trinary, Absolute singular Threat, mate threat, Influence and Domain dependent.

The hardware search is fast, but is relatively simple in the Deep Blue system configuration. To strike a balance between the speed of the hardware search and the efficiency and complexity of the software search, they limit the chess chips to carry out only shallow searches. Some of the parameters used were - Depth of search, Depth of offset searches, Endgame rules assertions off or on, Number of "mating" checks allowed for each side in the quiescence search etc.

Parallel search implementation
The early iterations of the Deep Blue parallel search are carried out on the master node. There is not much parallelism in the first few iterations, and the master is fast enough (it has 16 chess chips) that there is little to be gained by attempting to further parallelize the search. As the search gets deeper, jobs get allocated throughout the system. There were three major issues that need to be addressed: Load balancing, Master overload and Sharing between nodes

Parallel search performance - The results varied widely depending on the tactical complexity of the position searched. For positions with many deep forcing sequences speedups averaged about 7, for an observed efficiency of about 30%. For quieter positions, speedups averaged 18, for an observed efficiency of 75%. The non-deterministic nature of the search, particularly in tactical positions, makes it more difficult to conduct these measurements.

Evaluation function - The initialization of the feature values is done by the "evaluation function generator", a sub-program which was run on the master node of SP system.

Opening book, extended book and end game databases were created to study the thousands of moves.

Conclusion –
The large searching capability, non-uniform search, and complex evaluation function were all critical. However other factors also played a role, e.g., endgame databases, the extended book, and evaluation function tuning.

The hardware search and evaluation could have been made more efficient and flexible with the addition of an external FPGA.