Deep Blue, IBM Watson team

The paper describes and justify both the hardware and the software that lead to the victory of the chess computer Deep Blue 2.
From the start, the deep blue system cannot be considered as a program as it heavily relies on hardware specific infrastructure. The overall set up is a very classic search (alphabeta search) with evaluation function. However, the evaluation function and some search are programmed in a hardware which makes the evaluation faster but also less flexible.

The general method consists in an alpha beta search with iterative deepening within the negamax framework and with quiescence search performed on chess chip (hardware search). The search is actually a succession of search both software and hardware.
Software search is a selective search which extends in smart ways by selecting branches. The game of chess could otherwise never end and the search complexity would be much larger leading to a large computational time. This search makes use of various principle used in chess such as the fact of accepting to lose on one round in order to win bigger in future round and the so called forced pairs of moves.
The hardware search is directly implemented on the chess chip is a simple Quiescent search. These chips are used to perform search of 4 to 5 ply. The algorithm is much simpler than the one used in software search and is therefore used for shallow search as it may extend for too many branches and become slow for deeper search.
Both implementations make use of pruning techniques to limit number of branches expanded meaning that branches leading to equivalent scores are pruned if a better branch was found first. Therefore, these nodes will not be search further.

Deep blue makes use of extensive evaluation functions mainly implemented on hardware chips. About 8000 patterns are entered. The patterns were mostly determined expertly. These are the evaluations function that are coded in the hardware for computational efficiency. This is due to the fact the number of calls grows exponentially with the depth of the search.

Obviously, the system is parallel in order to make use of the fact that a search can very easily be seen as independent recursive calls on either a software search or a hardware search.

Finally, in addition to this extensive algorithm the system makes use of an opening table in order to allow avoid very long and meaningless search at the beginning of the game of around 4000 opening moves. An extended book is also available in case the opening moves are not in the opening book. The information in that book is more summarize and is here to complement of the alpha beta search rather than a fully fledge book of moves. It allows the system to measure that no bad move is made because of a search not deep enough. Similarly, it makes use of a closing book when the number of pieces when only 4-5 pieces are remaining.

Results
Main result is a victory against Gary Gasparov. The selective search allows searching an average depth of 12.2 within a 3 minute-time period.

All this was possible due to a highly selective method to search or not. A well-designed evaluation function taking into account 8000 patterns. Use of opening and closing books. And the overall implementation via hardware and parallelism made it possible to perform such a number of calculations.