

Deep Blue System Overview

Feng-hsiung Hsu, Murray S. Campbell, and A. Joseph Hoane, Jr.
IBM T. J. Watson Research Center

Abstract

One of the oldest Grand Challenge problems in computer science is the creation of a World Championship level chess computer. Combining VLSI custom circuits, dedicated massively parallel $\alpha\beta$ search engines, and various new search algorithms, Deep Blue is designed to be such a computer. This paper gives an overview of the system, and examines the prospects of reaching the goal.

1 Introduction

Charles Babbage, back in 1840s, contemplated the possibility of playing chess using his Analytical Engine. Claude Shannon, at the dawn of the modern computer age, presented his seminal paper [10] on computer chess in 1949 at a convention in New York. Across the Atlantic, Alan Turing hand simulated his program and played possibly the first computer chess game on record around the same time. In 1957, Herbert Simon gave his keynote speech [11] predicting a computer would be the World Chess Champion before 1967 in the annual meeting of Operational Research Society of America.

Despite the early enthusiasm, computers were still playing at rank beginner level in 1966, when Richard Greenblatt brought up a program that started to play at low Class C (about a weak amateur) level on the USCF (United State Chess Federation) scale. Ten years later, the top program, Chess 4.5 [12], reached Class A level (strong amateur), playing on a Cyber 176 supercomputer. This is more than one full class stronger than when the same program was playing on its regular machine, which was ten times slower.

By the late 1970s, it became clear that faster hardware speed was the single most effective way to improve program performance, when used in conjunction with brute force searching techniques. Joe Condon and Ken Thompson [2] of Bell Laboratories proceeded to build the famous Belle special-purpose chess automaton. In 1982, searching about 8 plies (4 moves by White, and 4 moves by Black) in chess middle games, it became the first computer to play at USCF National Master level and thus claimed the first Fredkin Intermediate Prize. Thompson's self-play experiments [13, 4] with Belle, however, showed that the performance increase started to seriously taper off when the search depth exceeded 8 plies. At shallower depths, Belle gained about 200 rating points (one full chess Class) for each additional ply, or close to 100 rating points for each doubling of search speed. (To search one extra ply requires a six-fold increase in search speed on average.) The basic slope measured by Thompson was later confirmed in both machine vs. machine and machine vs. human play. His observed tapering off effect, on the other hand, did not actually happen when faster chess automata finally appeared.

The quest for faster hardware continued unabated during the 1980s. First, there was Cray Blitz [8] running on the Cray supercomputers. Then there was the Hitech chess automaton from Carnegie Mellon University, using a 64-chip (one chip for each square on the chess board) VLSI move generator [3]. The first author, Hsu, started a different machine in 1985, also at Carnegie Mellon, using a one-chip move generator based on a refined Belle move generator design, which later evolved into the Deep Thought chess automaton [6, 7, 5]. Deep Thought won the second Fredkin Intermediate Prize by being the first machine to achieve Grandmaster level performance over 25 consecutive rated games in 1988.

Deep Thought, and, to a lesser degree, Hitech were both playing at well above what the Belle self-play curve predicted for their search speed. It is difficult to answer why this is so. One conjecture is that Belle

Permission to make digital/hard copies of all or part of this material without fee is granted provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication and its date appear, and notice is given that copyright is by permission of the Association for Computing Machinery, Inc. (ACM). To copy otherwise, to republish, to post on servers or to redistribute to lists, requires specific permission and/or fee.
ICS '95 Barcelona, Spain © 1995 ACM 0-89791-726-6/95/0007..\$3.50

was not optimized for greater search speed, and the other two machines were. *Selective* searching programs did not perform well back in Belle's heyday when computers were much slower, while by late 1980s, all of the top programs had significant selective searching components. It is also plausible that deep searchers require different types of chess knowledge to be incorporated into their evaluation functions.

2 Project History

The first two authors, Hsu and Campbell, joined IBM in 1989, having previously worked together on the Deep Thought machine. Shortly after, in October 1989, the first ever exhibition match between a reigning human World Chess Champion and a computer was played in New York city. The players were Gary Kasparov and Deep Thought. Kasparov won handily as predicted by the roughly 350 rating point difference of the two players. In fact, having previously studied all the computer's games, Kasparov could very well have been effectively playing another 200 points higher than the machine. To counter the 550 potential rating point difference would need 3 additional plies, assuming that the rating increase per ply can be maintained. Deep Thought was searching about 700,000 Pops (positions per second). To reach equity with Kasparov, if the extrapolation holds, requires a machine searching at least 150 million Pops.

A two-phased approach was adopted. First, an interim machine, Deep Thought 2, was completed in 1991 with both more knowledgeable new chess processors, and completely new software. Deep Thought 2's chess processors are comparable to Deep Thought's chess processors in speed (500,000 Pops when running stand-alone), but had more chess knowledge built in, and up to 24 processors can be running simultaneously (Logically, up to 32, but only 24 were built). The software for Deep Thought 2 underwent several revisions. With a 14-processor configuration, Deep Thought 2 searches 3 to 5 million Pops and plays at close to 2600 international rating, or about 200 points weaker than Kasparov.

Experience with Deep Thought 2 offered two valuable lessons. Long range chess knowledge gaps have to be explicitly filled in the hardware and the deeper the searcher, the more selective the search should be. These lessons, along with other findings were applied during the design of a new single chip chess processor in the second phase of the project.

The second phase of the project went into full swing after the first version of Deep Thought 2 software was completed in 1991 and Joseph Hoane, who joined the group in late 1990, took over the maintenance and the revisions of the search code. Hsu started to work on the

Table 1: Chip Characteristics

Transistors	about 1 Million
Logical	450,000
Memory	550,000
Technology	0.6 μ m 3 metal CMOS
Package	144-pin PGA
Clock Freq	30-40 MHz
Speed	3-5 Million Pops

new chess chip, while Campbell and Hoane continued to improve Deep Thought 2 from average Grandmaster strength to close to Super Grandmaster (2600+ international rating) strength by 1995.

3 Hardware

3.1 Chess Chip

The single-chip chess move generator used in Deep Thought and Deep Thought 2 was designed in about 6 months. The new chip took over 3 years to complete. Unlike the move generator chip, the new chip is a full-fledged chess machine on its own, containing a new move generator, a sophisticated hardware evaluation function, and an $\alpha\beta$ search controller. It contains over one million transistors versus the 36,000 or so in the old move generator. A single new chess chip searches 3-5 million Pops, or about the same as the entire Deep Thought 2, but since both the evaluation function and the search would be much improved, a system using a single new chess chip can be expected to play at or above Super Grandmaster level when properly optimized. To implement the equivalent function of one chip on a general purpose computer would require a 10,000 MIPs machine, assuming coding efficiency of 2,000-3,000 instructions per position searched. A rough description of the chip is given here, further details will be presented in a future paper. Table 1 summarizes the basic design characteristics of the chip.

Of the three components of the chip, the *evaluation function* is by far the biggest, occupying two thirds of the chip area. The $\alpha\beta$ *search controller* is the smallest, using only about 5% of the chip area, with the balance going to the *move generator*. The evaluation function contains over half of the logical transistors, and more than 80% of the memory transistors, distributed over 50 odd RAMs and ROMs of various sizes.

The core part of the move generator is an 8 by 8 combinatorial array, one cell per square of the chess board, following the same basic design used in the Deep Thought move generator [7]. The rules of the chess game are embedded in the wiring patterns between

the cells. Unlike the old move generator which can only generate pseudo-legal moves on demand, the new move generator core can generate check evasion moves or checking moves as well. Additional support to detect hung pieces and to measure the forcefulness of a check is also included. Special circuitry outside the core keeps track of the presence of special moves (en passant and castling) and generates the special moves at the appropriate time in the move generation sequence. This new move generator allows the search to be more selective than was possible with the old move generator, and the move ordering is also slightly improved.

The evaluation function can be partitioned into three parts: piece placement evaluation, endgame evaluation, and slow evaluation. The piece placement evaluation assigns a value for each piece on every square of the chess board, usually precomputed by software based on the game position. When a piece captures another piece, the moving piece's from-square value and the victim piece's to-square value are subtracted, and the moving piece's to-square value is added. The endgame evaluation maintains the counts of various chess pieces still on the chess board, the XORed locations of all the pieces of a given piece type, and a bitmaps of all the pawns. Simple endgame assertions such as rule draws, insufficient material, bishops of opposite colors and so on are computed based on the piece counts. For certain endgames, the XORed piece locations are used to look up the possible outcome from ROM tables. King centralization bonus is also computed as part of the endgame evaluation. The pawn bitmap is embedded into an 8 by 8 combinatorial array, which is somewhat similar to the move generator core but much smaller. The pawn array computes whether a given pawn is beyond the reach of opponent's king and whether it has a clear path to promote. A few more complicated functions related to passed pawns are also computed by the pawn array. Both the piece placement evaluation function and the endgame evaluation are computed in one single cycle. The slow evaluation is the single most complicated element on the chip, occupying about half of the chip core, and takes 10 cycles to compute. At very shallow depths, the slow evaluation has to be computed for about half of the positions searched, but at the realistic search depths, the percentage drops to around 15%. The slow evaluation has a 3-stage pipeline, starting with a 8 by 1 systolic array that runs for 8 cycles, one cycle per file. This array is followed by 40 odd synchronous RAMs, and then an adder tree that accumulates the results. The slow evaluation sequence can be stopped on the fly by the controller to reduce power consumption. The slow evaluation computes values for chess concepts such as square control, pins, xrays, king safety, pawn structure, passed pawns, ray control, outposts, pawn majority, rook on the 7th, blockade, restraint, color complex,

trapped pieces, development and so on. It is perhaps more complicated than any chess evaluation function ever described in computer chess literature.

The search control does not really implement the regular $\alpha\beta$ search algorithm [9]. Rather, it implements a *minimum-window* $\alpha\beta$ search algorithm. This eliminates the need for a value stack. The search control contains a 16-bit datapath and three state machines controlling different sections of the datapath. Two of the state machines also control the move generator indirectly. The datapath uses multiple cascaded adder/subtractor's to compute the conditional flags needed by the search algorithm in as few cycles as possible.

At the system level, the chip appears as a 32-bit device with a 17-bit address space. One of the possible addresses, when written, initiates a search from the current position on the chip, usually for 4 or 5 plies beyond the software search depth. The host processor can then be freed up to do housekeeping chores, or initiate a search on another chip.

3.2 Initial System

The initial system is configured similar to Deep Thought 2. Up to 4 microchannel cards, each with 8 chess chips, can be plugged into an IBM RS/6000 workstation. Currently, the initial system is not expected to use more than 16 processors at a time, although, in theory, a 16-processor IBM SP2 supercomputer can be used to support up to 512 chess chips. There is no plan to do so at the moment. Each chess chip is about the same speed as the entire Deep Thought 2, and Deep Thought 2 is only about 200 points from Kasparov. It is conceivable that a single card, 8-chip, system could be right at Kasparov's level, when *properly optimized* and against an *unprepared* Kasparov. The first release of the software is adapted from Deep Thought 2 software, which almost certainly is not optimal with respect to the new hardware.

The purpose for building this initial system is mainly for debugging and preparing for the chip production run which will be used to build the final system. It also should eat Deep Thought 2 for lunch. [For non-native English speakers, it should beat Deep Thought 2 consistently.]

3.3 Final System

The design of the final system is not yet complete. The system is to be mounted on a rack about 6 feet tall, and has a footprint about the same as an office refrigerator. A PowerPC based host inside the rack controls a 21-slot 9U full depth VME box which contains the chess chips and a hardware hash table. The current estimate is that up to 8 boards of chess chips, 128 chips

per board for a total of 1,024 chips, up to 4 boards for the hardware hash table and a host interface board would be inside the VME box. The total power dissipation is expected to be around 2-3 KW. The aggregate raw speed would be 3-5 billion Pops. Assuming a 30% parallelization efficiency, the effective speed would be around 1 billion Pops, or greater than one thousand times the speed of Deep Thought. A general purpose machine used to perform the same task would need at least one Teraops.

4 Software

Since the design of the the final system is incomplete at the moment, the descriptions of the software here are limited to the software of the initial system. It is essentially the same as the Deep Thought 2 software in basic structure.

The software supports up to 32 chess chips as does the initial system hardware. The chess chips are used as hardware routines by the software. For an 12-ply search, the software searches the first, say, 8 plies, and then gives the positions to the the chess chips to search the remaining 4 plies. Each 4-ply search takes about the order of 1 ms, which gives the host processor some time to process the search results and initiate new searches. The software can adjust the chip search parameters to change the behavior of the hardware searches. Most of the selectivity of the search is in the software part of search. The software search uses a revised version of *singular extensions* [1], which has greater selectivity than the original algorithm. The parallelization efficiency is around 80% when doing a simple brute force search. For chess positions requiring very long sequences of calculations, the parallelization efficiency can drop to around 50%. For typical positions, with a brute force search depth of 12 plies the average maximum depth reached is around 36 plies.

The chess chip hardware supports a fairly complicated evaluation function, but not all of the evaluation is done in hardware. The software precomputes the evaluation tables for each game position and then downloads them onto the chips in parallel. Furthermore, in the software part of the search, a complementary software evaluation is also computed. The software evaluation is used mainly for long range planning. The software also accesses the on-line 5-piece endgame databases when a 5-piece endgame position is reached in the software part of the search tree.

An opening book created from a chess game database with 300,000 games makes up the rest of the software. Only the most frequent moves with good results are included in the opening book. The opening book is constantly revised. It is expected that the opening book will have to be changed dramatically when the

final system is completed and before the Match takes place.

5 Conclusion

Charles Babbage never did get his Analytical Engine to play chess, settling for the much simpler goal of playing tic-tac-toe. Herbert Simon's prediction never did come true. Very famous computer scientists have been known to make too optimistic predictions about computer chess.

Gary Kasparov, the human World Chess Champion, said in 1988, the very same year that Deep Thought scored the first victory by a computer over a Grandmaster in a tournament game, that computers will never beat a Grandmaster before the year 2000. Very famous chess players have been known to make too optimistic predictions about computer chess.

The truth probably lies somewhere in between. Eventually and inevitably, computers will overtake even the human World Champion. The authors believe that the time is at hand and Deep Blue could be the machine to do it.

Acknowledgements

The authors would like to thank Randy Moulic, C. J. Tan, Zeev Barzilai and Maurizio Arienzo for their support of the project, and Gershon Brody for providing engineering assistance.

References

- [1] Thomas Anantharaman, Murray Campbell, and Feng-hsiung Hsu. Singular extensions: Adding selectivity to brute force searching. In *1988 AAAI Spring Symposium, Computer Game Playing*, pages 8-13, March 1988.
- [2] J. H. Condon and K. Thompson. Belle chess hardware. In M. R. B. Clarke, editor, *Advances in Computer Chess 3*, pages 45-54. Pergamon Press, 1982.
- [3] Carl Ebeling. *All the Right Moves: A VLSI Architecture for Chess*. PhD thesis, Carnegie Mellon University, April 1986.
- [4] J. H. Gondor and Ken Thompson. Belle. In Peter W. Frey, editor, *Chess Skill in Man and Machine*, pages 201-210. Springer-Verlag, 2nd edition, 1983.
- [5] F. Hsu, T. Anantharaman, M. Campbell, and A. Nowatzky. Deep thought. In T. A. Marsland

- and J. Schaeffer, editors, *Computers, Chess, and Cognition*, pages 55–78. Springer-Verlag, 1990.
- [6] Feng-hsiung Hsu. A two million moves/sec cmos single chip chess move generator. In *1987 ISSCC Digest of Technical Papers*, page 278, February 1987.
 - [7] Feng-hsiung Hsu. *Large Scale Parallelization of Alpha-Beta Search: An Algorithmic and Architectural Study*. PhD thesis, Carnegie Mellon University, February 1990.
 - [8] Robert M. Hyatt. Parallel chess on the cray xmp/48. *International Computer Chess Association Journal*, pages 90–99, 1985.
 - [9] Donald E. Knuth and Ronald W. Moore. An analysis of alpha-beta pruning. *Artificial Intelligence*, 6:293–326, 1975.
 - [10] C.E. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 41:256–275, 1950.
 - [11] H. A. Simon and A Newell. Heuristic problem solving: The next advance in operations research. *Operations Research*, 6(1):1–10, 1958. Address given at the 1957 meeting of Operations Research Society of America.
 - [12] David J. Slate and Lawrence R. Atkin. Chess 4.5—the northwestern university chess program. In Peter W. Frey, editor, *Chess Skill in Man and Machine*, pages 82–118. Springer-Verlag, 1977.
 - [13] K. Thompson. Computer chess strength. In M. R. B. Clarke, editor, *Advances in Computer Chess 3*, pages 55–56. Pergamon Press, 1982.