How We Won The Computer Chess World's Championship

by Harry Nelson

[This article is an excerpt from a talk given at the DAS Computer Science Colloquium.]

About nine years ago I listened at this seminar as Ed Kozdrowicki of U.C. Davis gave a talk on "How we lost the world computer chess championship". Fortunately I am able to present better news this time.

Over the weekend of October 22-25, 1983, at the Association for Computing Machinery (ACM) tourney, the program Cray-Blitz established itself as the World Champion at computer chess. Basically, four enhancements not available a year earlier were responsible for this victory.

These four items are:

1. CRAY X-MP computer
2. Many more subroutines rewritten in CAL
3. Multiprocessing
4. New ideas in the opening library

(5. Luck)

Before looking in detail at each of these, let me review some of the history of computer chess and the Cray-Blitz program.

Around 1950 Claude Shannon wrote about how a computer might be programmed to play chess. The first complete games were played by 1957, but another ten years passed before any program had reached a level at which it could be rated in human terms.

The worldwide chess rating system has the following categories:

Under 1200	Class E
1200-1399	Class D
1400-1599	Class C
1600-1799	Class B
1800-1999	Class A
2000-2199	Expert
2200-2399	Master
2400-up	Senior Master

Within the Master categories there are International Master and Grand Master, which have separate requirements. In 1973 Bobby Fischer held the highest rating ever achieved, about 2750. Victor Korchnoi today is rated 2670.

In 1967 Mac Hack 6, written by MIT's R. D. Greenblatt, played in enough official U.S. Chess Federation tournaments to achieve a rating of just over 1400. Encouraged by Mac Hack's performance, several professors of computer science came out with pronouncements that computers would soon be able to win against the best human players.

But in 1968, David Levy, an English International Master rated around 2350, who had made a study of computer chess methods, decided that programs were not going to improve much soon, made a bet that no computer program would be able to beat him in a serious match within the next ten years, and staked 500 British pounds on himself.

This challenge led directly to the organization of annual computer chess tournaments under the sponsorship of ACM to determine a current North American champion, this champion being selected to play Levy for "The Honor of Computer Science".

After several of these ACM tournaments had been held, it was decided to have, in 1974 and every three years thereafter, a "World's Championship". Levy himself acted as Master of Ceremonies and Tournament Director for the first such event, which was held in Stockholm, Sweden, and was won by Kaissa, the entry from the Soviet Union, developed under the leadership of Mikhail Botvinnik, a former human world champion. Kaissa lost to Levy, and in fact through 1977 all challengers were decisively defeated by Levy, who won every game.

In 1977, Chess 4.6, written by David Slate and Larry Aikin of Northwestern University, won the Second World Championship in Toronto, Canada, running on a CDC Cyber 176, a 7600-class machine. Its successor, Chess 4.7, lost to Levy in the last match held during the ten years of his wager, although it did win one game (of six), the first computer to do even that.

After the 1978 match, Levy was pressured to renew his bet for another ten years, but he declined, saying that the progress shown in the preceding decade made him uncertain that he would be able to win every match over the next ten years; however, he did agree to extend for five years, until 1984. Backers were quickly found on both sides, and there is at present $10,000 at stake.

In 1980 the winner of the Third World Championship in Linz, Austria, was Belle, the Bell Labs entry produced by Ken Thompson, the programmer (who won this year's Turing award for contributions to Unix), and Joe Condon, the hardware specialist. Belle was the first successful single-purpose hardware box designed specifically to play chess, and as such was the mother of all the microprocessor "Chess Challengers" of today. (The Belle programmers think of their machine as "Ma Belle" and always use the feminine gender in speaking about her.)

From then through 1982 Belle consistently won outright or tied for first in every computer chess event she entered. Moreover, Belle has played hundreds of rated games against human opponents and in August, 1983, became the first program to win the rating of "Master" by gaining 8-1/2 points out of 12 at the U.S. Open in Pasadena, finishing with a rating of 2203. Belle is capable of examining over 100,000 positions (or nodes) per second (NPS), but does practically no analysis at each node, except to count pieces. This speed allows Belle to look ahead in a typical situation to a depth of 8 ply (four full moves). Studies with Belle seem to indicate that a program advances by about 200 rating points with each additional ply searched in a given time.

Although winning an occasional game, Belle also has been unsuccessful in several matches against Levy.

The Blitz chess program was begun in 1975 by Robert Hyatt, an instructor at the University of Southern Mississippi in Hattiesburg. It was written in standard Fortran and drew heavily from the papers of Slate and Aikin for its methods. Hyatt continued to work on the program through the 1970s, adding his own innovations and keeping it current with the latest knowledge in the field. The program played in several rated events in the Hattiesburg-New Orleans area and held a rating around 1700, using USM's SIGMA-9. During this period, Albert Gower, a music professor, correspondence chess player, and fellow member of the chess club at USM, began helping on the program, especially in the "openings book" area.

After the 1980 ACM tournament, where Blitz running on a VAX performed in a mediocre fashion, Hyatt and Gower could see that without a faster computer, Blitz was going nowhere. Hyatt then approached Cray Research, which agreed to support him with CRAY-1 machine time. The code was converted to compile with CFT and renamed Cray-Blitz (C-B). By 1981 it had improved its speed by a factor of 10 and its rating by about 400 points, winning the Summer 1981 Mississippi State Open Tournament and defeating, in the finals, a human Master player.

Around this time, I became interested in the program, having heard about it from Cray Research people, who were able to obtain a copy for me. I was mainly interested in comparing its performance against Chess 4.5, the 1976 version of Slate and Aikin, which was available here on the CDC 7600. I found that Blitz was slower—3000 NPS compared to 4000 NPS for Chess 4.5—and sometimes better, sometimes worse in chess "ability".

A position may help illustrate this. Here, WHITE's winning method consists in chasing black's king with the Queen, forcing it into a corner, winning the rook and unprotected pawns when black plays to h3, stepping the King to f2, pushing black's king back into a corner, and after pg1 is forced, capturing the remaining men, then (finally) enforcing mate.

The winning moves begin:

	WHITE	black
1.	Qc5	ka6
2.	Qb4	rh3
3.	Qd6+	ka5
4.	Qd2+	ka6
5.	Qd3+	kb6
6.	Qd8+	?

```
              black

 -  *  -  *  -  *  -  *   8
 *  k  *  -  *  -  *  -   7
 -  *  -  *  -  *  -  *   6
 Q  -  *  -  *  -  *  -   5
 -  *  -  *  -  *  -  p   4
 *  -  *  -  *  -  p  -   3
 -  *  -  *  -  *  p  r   2
 *  -  *  -  *  -  K  b   1

 a  b  c  d  e  f  g  h
              WHITE
```

Because C-B used a transposition table to save the value of each position looked at and 4.5 didn't, C-B could find these first few correct moves in a few seconds, while 4.5 would need hours. But having won the rook, C-B was unable to find the winning continuation, while Chess 4.5, because of its better endgame technique, could then proceed to the win given as little as five seconds per move.

It seemed odd that the CRAY program should run slower, so I used a timer-tally analysis on it and found that, for example, 25% of the time was being spent in one 20-line subroutine ATTACK, which checks to see if a given square is under attack, and that the code had numerous other inefficiencies.

I rewrote ATTACK in CAL in September 1981, reducing its time by 75% and offered it to Hyatt, who was pleased to accept. He invited me to come as an observer to the ACM tourney being held in Los Angeles the next month.

In that tournament C-B tied for second, losing to Belle in the final round, after winning three straight games prior to that. The crucial position against Belle arose after C-B, playing WHITE, had made its 27th move.

```
      black (belle)

 -  *  -  *  -  r  -  k   8
 *  -  *  -  *  -  p  p   7
 -  n  -  *  Q  *  -  *   6
 *  -  *  -  p  -  *  -   5
 -  *  -  *  -  *  -  *   4
 *  -  *  -  *  -  *  P   3
 P  P  -  *  -  *  P  K   2
 R  -  B  -  q  -  *  -   1

 a  b  c  d  e  f  g  h
      WHITE (C-B)
```

On her 27th move belle offered to sacrifice the knight on b6 and C-B fell for it. The game proceeded:

	WHITE	black
27.	...	ph6
28.	Qxb6	rf1
29.	Qd8+	kh7
30.	Qd3+	pe4
31.	Qxf1	qxf1.

After which C-B quickly lost, as the pawn at e4 cannot be stopped from promoting without giving up a piece.

The other second-place finisher was Nuchess, a rewrite of Chess 4.9 by Slate (Aikin having dropped out), which had drawn with Belle in the next-to-last round. Another high finisher was the Michigan U. entry, Chaos.

After the tourney, Hyatt came to LLNL and spoke at the DAS seminar. I was unable to attend, but decided to keep helping out by redoing a few more routines in CAL, in my "spare time". Though the CFT version contained no automatically vectorizable code, using CAL I was able to partially vectorize ATTACK and MOVGEN, the routine which generates legal moves from a position, although the longest vector was only of length 7. This effort, together with some re-Fortranning that Hyatt did at my suggestion, gave us a factor-of-2 improvement during 1982, so that instead of 3000 NPS we were getting 6000 NPS, without lessening the amount of analysis for each position. Because of its large transposition table and sophisticated cutoff methods, C-B gains about 1 ply of look-ahead for each factor-of-3 increase in speed, so we hoped to finish higher in the ACM tourney.

I was unable to attend the 1982 tournament in Dallas, but the result was that Belle, Nuchess, Chaos, and C-B finished in a four-way tie for first, with Belle winning on the basis of the tie-breaking procedure.

The chart below shows the cumulative scores of the top finishers. The same letter means those two programs played in that round. (A draw counts 1/2.)

Round	1	2	3	4
Belle	1/2	(a) 1 1/2	(d) 2 1/2	(f) 3
Nuchess	1	(b) 2	(d) 2	(g) 3
Cray Blitz	1	(c) 1 1/2	(e) 2 1/2	(f) 3
Chaos	1	(a) 1	2	3
Bebe	1	(c) 1 1/2	2 1/2	(g) 2 1/2
Fidelity X	1	(b) 1	(e) 1	2

While considering what to do to improve the code more for 1983, we knew that additional CAL coding could be done, but the payoff was diminishing. In addition, we thought that we could probably get the use of the CRAY X-MP which would help. Also, I suggested and implemented a redesign of the transposition storage table to enable a doubling of its capacity. Hyatt, meanwhile, decided to rewrite the basic-search mechanism to utilize the latest theoretical advances he had heard about in Dallas. (At an ACM tournament the various programmers get together, give papers, and share information about the changes they have made during the previous year. However, <u>during</u> the previous year, they are very secretive about it, hoping to spring surprises at the tournament.) Finally, Bert Gower had determined that the openings library we were using was not suitable for play against Belle, in particular, and computers in general, so he would spend the year reworking that with Belle as the target.

In July 1983 we decided that it was time to try out the latest version, and, being able to get weekend time on the hardware-development CRAY-M at Chippewa Falls, Wisconsin, we entered the speed chess event, a one-night tournament starting Saturday, August 13, at the U.S. Open in Pasadena. Belle was in Pasadena for the regular tournament along with over 800 human competitors, but she did not play in the speed chess event.

The speed tourney session had about 170 entries and was divided up into 9 round-robins of 18 or 19 players each, with the top two from each round-robin advancing to the final 18-player round-robin. Several Grand Masters and Senior Masters were entered. The upshot was that the C-B code won 13 and lost 4 in its section, finishing in a three-way tie for second with two humans, each rated over 2450 and each of whom had lost to the program. However at this point, due to the fact that we had exhausted our scheduled machine time, not to mention Hyatt's typing muscles and the CRAY's operator, who was still standing by long after her shift had ended, we withdrew, and did not play in the final round-robin.

Just for the interest of chess players, here is the order of finish in the speed tourney.

Position	Name	Rating	Result
1.	L. Alburt	2612	13
2.	K. Shirazi	2563	12
3.	J. Benjamin	2605	11 1/2
4.	L. Gutman	2497	11
5.	D. Gurevich	2522	9 1/2
6.	J. Tarjan	2631	9 1/2
7.	Thinnsen	2408	9
8.	N. Defirmian	2599	8 1/2

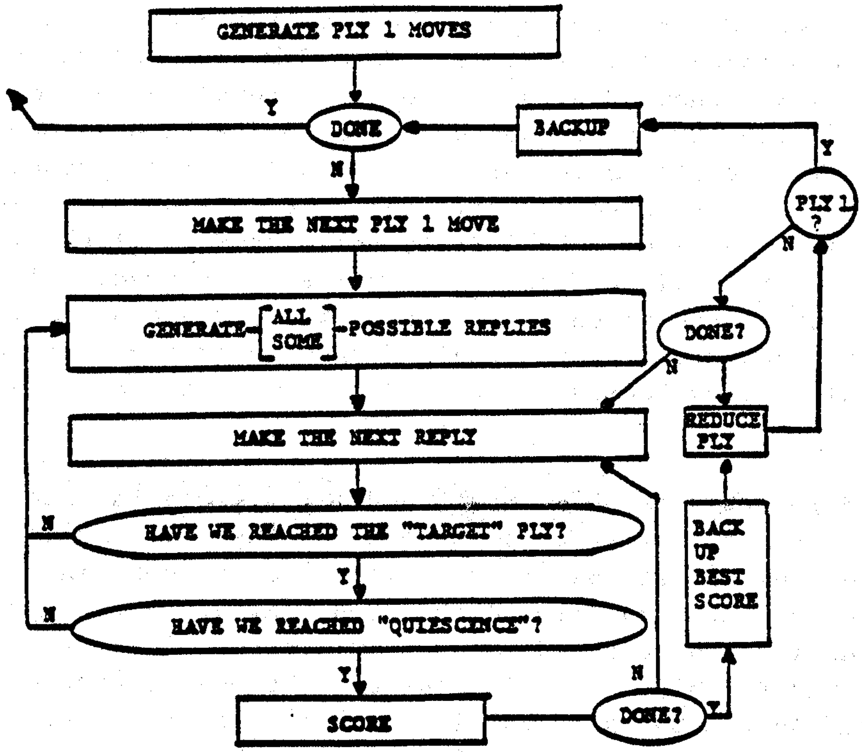
The authors of Belle have noted that in speed chess, humans perform about 200 points lower than their rating, relative to machines.

While Hyatt and I were in Los Angeles for the tournament, I pointed out that Cray Research had just released an experimental version of their two-processor, multiprocessing operating system for the X-MP, and that since we were supposed to get the X-MP for the ACM tourney, we might be able to figure out how to use both processors together to get another nice speed-up. So Hyatt decided to look into this, while I elected to rewrite the new basic-search in assembly language and to make the necessary changes to the other CAL subroutines to allow for multiprocessing.

With a lot of help from the Cray system people in Minnesota, who generally were quick to fix the bugs we found in their experimental operating system, we got everything brought together (including Gower's revised openings book) in early October. Unfortunately, although it *was* fast (25,000 NPS), nearly all of our test problems failed to run correctly and the system sporadically crashed whenever we tried to use it. My CAL changes made to use the "new calling sequence" seemed to be incompatible with Cray's multiprocessing library.

Two weeks later, and a week before the tournament, I decided to go to Minnesota to work more closely with the Cray experts to try to resolve the remaining problems. I did find and have fixed two system-related difficulties and corrected numerous program bugs, but when I flew to New York on Saturday morning, the beginning day of the tourney, the best code we had running *correctly* was a pure Fortran version. This version worked okay on one processor and *seemed* to be able to multiprocess all right, but we couldn't be sure since the code was designed to be nondeterminative. For this code, even while multiprocessing, our NPS rate stayed under 10,000.

Let me explain how the work is divided up between the two processors. In a given position, with the machine to move, there are generally a few dozen possible moves available. Normally the search procedure is as follows:



This is called a "depth first, iterative deepening, full-width, alpha-beta, minimaxing" procedure. All possible moves are considered up to the target ply, after which only checks, captures, and promotions are searched further.

The only changes made for dual processing are that the ply-1 moves are shared by the two processors, after which they do the deepening on their own, finally sharing the backed-up score when returning to ply-1.

As far as the actual coding was concerned, it was only necessary to replicate subroutines involved in the search procedure (99% of the time is consumed there), form two threads of routines from them, giving one set new subroutine names and new common block names, and write two new routines, one to initially share the ply-1 position information, and one to finally share the backed-up scores.

In the process we decided not to keep separate transposition tables for each thread but instead to share a common table. This leads to indeterminacy, since the score for a given position may come from a deep ply (found by the other processor and left in the table) or may be recalculated by the current one, depending on real time considerations. In fact, in test positions I have run and rerun, variations by as much as 20% in the time needed to reach a certain decision have been noted.

But back of the events of the Tournament. The format of the tournament was a five-round Swiss-style with 22 entries, 10 from the U.S., 3 from Canada, 3 from England, 3 from West Germany, and 1 each from Sweden, Austria and the Netherlands. The rules are about the same as for human tournaments. Both sides use a clock and are allowed 2 hours for the first 40 moves and 30 minutes for each 10 moves thereafter. All decisions must be made by the programs, but a human is allowed to type in the opponent's moves and operate the clock.

The four top seeds were Belle, Cray-Blitz, Nuchess and Bebe. Bebe is a powerful special purpose processor, like Belle. Both C-B and Nuchess were using Cray products. Nuchess was connected by phone to the hardware-development M-processor in Chippewa Falls, while we were using the Software-development X-MP in Mendota, Minnesota. Many of the machines being used were microprocessors, and over half of the computer hardware was physically present on site, including Belle and Bebe. Belle is about 2 by 2 by 3 feet, while Bebe is slightly larger. The latest "Chess Challenger," called Fidelity X, was also entered. Although invited, the Russians didn't send an entry, though Botvinnik did attend. He said that the programmers had not improved their program over the last three years and felt they would not be "competitive". He promised to be ready for the next tourney, though.

From New York on Saturday afternoon, Hyatt and I dialed up the machine in Minnesota. We were able to replace three important routines with their CAL versions and verify that the result was correct, in either thread, in uniprocessing mode. But 7 p.m., the time of the first match, came before we could go further. We decided to go ahead and use multiprocessing for the experience despite the uncertainty. One CAL routine, the move generator, which seemed to be okay in one thread but not the other, we decided to leave in Fortran. Fortunately, our opposing program, BCP (British Chess Program), was having trouble too, and due to a bug of theirs, used less than 5 seconds per move. We won the first game. Belle and Nuchess also won in the first round, but Bebe lost in an upset to Merlin, an Austrian entry of unknown strength, in 77 moves—a game which lasted till 3 a.m.

Immediately after our game ended, about 11 p.m., Hyatt and I went back to our debugging. We stayed on the X-MP till 5 a.m., and by then had fixed all the problems except for some (still unknown) bugs in the CAL SCORE routine, which slowed the code down to CRAY-1 speed in uniprocessor mode. However, the multiprocessing seemed to be working okay and we were pleased to see the program reporting 198% average CPU utilization on our test positions and during the previous match.

In the second round, on Sunday afternoon, we won against Fidelity X, Nuchess beat Merlin, and Belle also won. The code performed well and we were averaging over 20,000 NPS.

The third round was Sunday night. There were only four undefeated, untied programs, so Belle was matched against Nuchess while we played Advance 3.0, our third straight microprocessor. We won easily, while Nuchess beat Belle, her first loss in five years of ACM tournaments.

Here is a position from the Belle-Nuchess match. In this position both programs blundered, being unable to go deep enough to find the best moves, much to the consternation of the programmers. (An interesting sidelight to me is that most programmers of chess programs are much better players than their codes. During a match they are always seeing further ahead than the program and they moan every time a bad move is made. Our team has no such problems.)

```
        black (belle)

 -  *  -  *  -  *  -  *   8
 *  -  *  -  *  -  p  p   7
 p  *  -  *  -  k  -  *   6
 *  -  *  -  *  -  *  -   5
 -  p  -  K  -  P  -  *   4
 *  P  *  -  *  P  *  -   3
 -  P  -  *  -  *  -  P   2
 *  -  *  -  *  -  *  -   1

 a  b  c  d  e  f  g  h
       WHITE (NUCHESS)
```

After:

37. ... kf6

Nuchess played Ke4; later analysis showed that Kc5 wins, while it is possible for black to hold to a draw after Ke4 with ke6. At least that was the opinion of the interested onlookers, who included several Master players from the New York area as well as Botvinnik. Since our game was already over, we tried our program on both sides here and found that indeed C-B would have played Kc5 as WHITE, and ke6 as black in reply to Ke4, within the usual time.

So with some confidence and a good morning's sleep, we entered the fourth round on Monday evening against Nuchess, while Belle played Chaos, which had won two after an initial draw.

By the luck of the pairings, Nuchess was playing WHITE again while we were black. We started off well, but at move 17 we made a characteristic error. The game had proceeded as shown below, leaving the position at right:

1.	Pf4	pd5
2.	Nf3	nf6
3.	Pe3	bg4
4.	Pb3	nbd7
5.	Bb2	pe6
6.	Bd3	bd6
7.	Ph3	bxf3
8.	Qxf3	pe5
9.	Be2	o-o
10.	O-O	pxf4
11.	Pxf4	re8
12.	Nc3	pc6
13.	Qd3	nc5
14.	Qf3	pd4
15.	Nb1	re4
16.	Pg3	qb6
17.	Qf2	?

```
      black (cray-blitz)

 r  *  -  *  -  *  k  *   8
 p  p  *  -  *  p  p  p   7
 -  q  p  b  -  n  -  *   6
 *  -  n  -  *  -  *  -   5
 -  *  -  p  r  P  -  *   4
 *  P  *  -  *  -  P  P   3
 P  B  P  P  B  Q  -  *   2
 R  N  *  -  *  R  K  -   1

 a  b  c  d  e  f  g  h
       WHITE (NUCHESS)
```

At this point C-B played na4, figuring that after Ba3, bxa3; Nxa3, its knight could go to b2 where it would be "safe". *Unfortunately*, there is then no way to extract the knight, but its actual capture takes place too far ahead for C-B to discover it. *Fortunately*, although the game did go as forecast, when we actually reached the 19th move, C-B could "see" deep enough and simply retreated the knight back to c5, losing only a (valuable) "tempo".

Nothing worse happened during the game and finally at move 77 the programs played to the same position for the third time, forcing a draw. Again, after much analysis by the Masters present, they agreed that indeed the game would have been a draw, with best play, from the final position.

Meanwhile, Belle had cleaned up on Chaos, and Bebe won her third straight after the first-round loss.

So, on the last night, we finally met Belle, once again getting the black pieces, while Nuchess took on Bebe. We thought we had a good chance, although we had never beaten Belle, partly because Bebe had been playing 5-second games against the local Masters all day and had won almost every game, which meant she would be no pushover for Nuchess, and partly because we knew Belle's "book" and had an innovation prepared.

The result was that we won a Pawn at move 20, held it till move 38, survived a couple of program bugs, and finally won on move 54, as Belle resigned when our extra pawn went to the 7th rank, threatening to either promote or force Belle's King to a square where our queen would check and mate in 3.

Meanwhile, Bebe did knock off Nuchess, which had black for once, finishing second, while Awit, a Canadian entry, snuck into third, managing to avoid playing any top seeds, leaving Nuchess fourth, Chaos fifth, and Belle sixth. Cray-Blitz's 4-1/2 out of 5 gave it undisputed first place, as every other team lost at least once.

There are several plans for future development. One is to have a two-tier book, one for computers, another for humans. We will also generalize the multiprocessing coding so as to be ready for any (small) number of processors. There is a possibility that an X-MP-4 may be available by next October's ACM tournament. We expect to get the CAL SCORE routine working, someday, and there is still one heavily used subroutine which has not been hand-coded which I expect to work on. There are several weak areas for which simple fixes are known (at the expense of additional time), which will be considered, plus any new developments garnered from the conference papers.

The program is scheduled, tentatively, to play Levy somewhere, sometime in the first quarter of 1984. We are invited to play in the (human) U.S. Open in Texas next summer. There is also a large prize (the Fredkin award of $100,000) being offered for the first program to win in a match against the human World Champion, and there is the ultimate goal of playing perfect chess, which is certainly much harder than beating the Grandmasters.

Right now there is a lot of interest in computers playing in human tournaments, but I suspect this will drop to zero if machines actually become better than the best human competitors. I don't expect that to happen in the current century, myself, however.