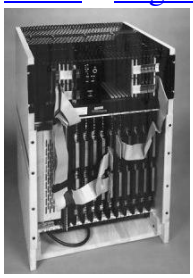
Home * Engines * **Belle**



Belle, 1977 [3]

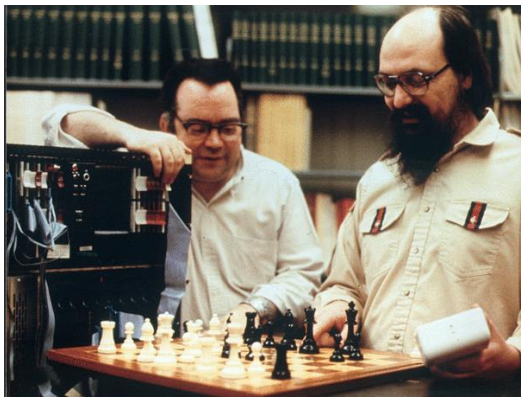
Belle,
the dominating chess machine in the late 70s and early 80s, was developed by Ken Thompson and Joe Condon [1] from Bell Laboratories. It was five times winner of the ACM North American Computer Chess Championship, the ACM 1978, ACM 1980, ACM 1981, ACM 1982, and ACM 1986, and won the Third World Computer Chess Championship 1980 in Linz [2].

Belle consists of a special-purpose hardware and associated software, and was pure brute-force. Belle started in the early 70s as a sole software approach, but more and more emerged to a hybrid chess computer, next using a move generator, a position evaluator, and a transposition table inside a special-purpose hardware. In its final incarnation, Belle was composed of a PDP-11/23, and further a LSI-11 processor with several custom boards. The speed increased from 200 nps from the software version to about 160,000 nps of the machine mentioned at the Advances in Computer Chess 3 conference in 1981.

Table of Contents

Photos & Games

Blitz 6.5



Belle, Joe Condon and Ken Thompson revisiting the 1978 ACM Blitz 6.5 - Belle game (1982) [4] [5]

```
[Event "ACM 1978"]
[Site "Washington D.C."]
[Date "1978.12.06"]
[Round "4"]
[White "Blitz 6.5"]
[Black "Belle"]
[Result "0-1"]

1.e4 e5 2.Nf3 Nc6 3.Nc3 Nf6 4.Bb5 Nd4 5.Bc4 Bc5 6.Nxe5 Qe7 7.Bxf7+ Kf8

8.Ng6+ hxg6 9.Bc4 Nxe4 10.O-
O Rxh2 11.Kxh2 Qh4+ 12.Kg1 Ng3 13.Qh5 gxh5
14.fxg3+ Nf3# 0-1
```

CHAOS

Belle vs CHAOS, WCCC 1980, Thompson, Friedel, Berman [6], Swartz, Donskoy [7]

```
[Event "3rd World Computer Chess Championship"]
[Site "Linz, Austria"]
[Date "1980.09.29"]
[Round "5 (playoff)"]
[White "Belle"]
[Black "CHAOS"]
[Result "1-0"]

1. e4 Nf6 2. e5 Nd5 3. d4 d6 4. Nf3 dxe5 5. Nxe5 g6 6. g3 Bf5 7. c4 Nb
4
8. Qa4+ N4c6 9. d5 Bc2 10. Qb5 Qd6 11. Nxc6 Nxc6 12. Nc3 Bg7 13. Qxb7
O-O
14. Qxc6 Qb4 15. Kd2 Be4 16. Rg1 Rfb8 17. Bh3 Bh6+ 18. f4 Qa5 19. Re1
f5
20. Qe6+ Kf821. b3 Bg7 22. Bb2 Bd4 23. g4 Rb6 24. Qd7 Rd6 25. Qa4 Qb6
26. Ba3 Bxc3+ 27. Kxc3 Rdd8 28. Rad1 Qf2 29. gxf5 Qc2+30. Kd4 gxf5 31.
 Qc6
Qf2+ 32. Ke5 Kg8 33. Rg1+ Kh8 34. Bxe7 Qb2+ 35. Rd4 Qg2 36. Qf6+ Kg8
37. Bxg2 Rxd5+ 38. Ke6 h6 39. Qxh6 Re5+ 40. fxe5 Rf8 41. Bf3# 1-0
```

Hardware Design

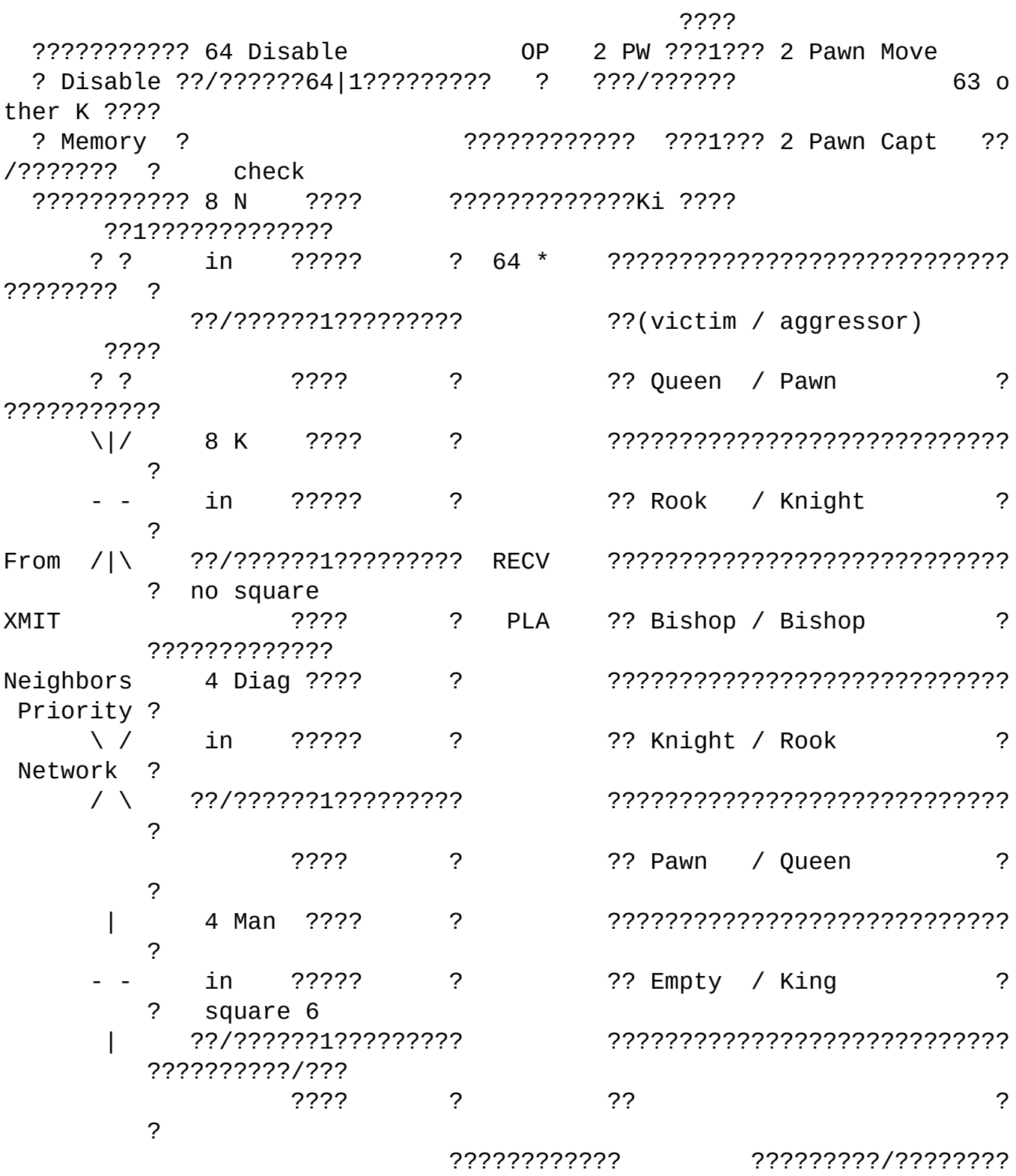
The find **victim** and find **aggressor** move generation was instrumented by Edward Fredkin, first applied in the Chess-orientated Processing System CHEOPS [8]. The overall architecture of Belle was used for the initial designs of ChipTest, the progenitor of Deep Thought which further evolved to Deep Blue [9], and more recently by FPGA projects, such as Marc Boulé's thesis project around MBChess [10], and notably Brutus [11] and its successor Hydra by Chrilly Donninger et al..

Move Generation

The hardware move generator consists of 64 combinatorial circuits similar for each square, controlled by appropriate latches representing the board, and receiving and transmitting signals from/to neighboring squares to propagate piece attacks. Two single wired op-codes, find **victim** and find **aggressor**, locate the highest valued piece (including empty square) under attack and the lowest valued piece attacking it in MVV-LVA manner. Move making and unmaking perform incremental updates not only on the piece registers, but also on a Zobrist like 48-bit hashkey and evaluation registers.

Block Diagram

Transmitter (XMIT), receiver (RECV), piece register, and ray- and direction multiplexer for each square [12]. The opcode (OP) input is either find victim or aggressor. It controls the transmitters and the order of the receiver outputs as input of the two level priority encoding tree.




64 *
63x6 others
P4
Piece Register
4 Man in
From X
MIT Neighbors
OP WTM
4 Diag in
Manhattan
Man out 4
To RECV and XMIT
Diagonal
RAY
Diag out 4
Neighbors
MUX
Empty
XMIT
control
To RECV Neighbors
ROM
Knight
N out 8
King
K out 8
Pawn
Move out 1
Capt out 2
DIR MUX
WTM
control
or

Find Victim

Find **victim** causes each transmitter (XMIT) to activate signals corresponding to the piece of the side to move on each square. Empty squares activate the sliding ray signals from their neighbors in appropriate directions. All 64 receivers (RECV) are now supplied with signals from their correspondent neighbors, and six disjoint piece (capture target) or empty flags per square are feed into a priority network, which finally outputs the square of the most valuable victim - the target-square. All this takes place simultaneously in propagation delay time of the combinatorial logic of all 64 squares. A king victim on any square, caused by an illegal move is indicated by the chess output flag, ored (>=1) over all 64 square receivers.

Find Aggressor

The find **aggressor** opcode causes only the addressed victim transmitter to radiate as the union of all other side to move pieces, the super piece, and otherwise applies the same logic as in the find **victim** cycle. All appropriate pieces of the side to move which receive attacks of the super-piece are potential from-squares of a pseudo-legal move, and piece disjoint signals in reversed order from pawn to king are feed into the priority network to get the from-square of the least valuable aggressor first.

Bookkeeping

Without further sequential logic subsequent find victim and aggressor cycles would always leave the same victim and same aggressor. A stack of 64-bit disable words is used for bookkeeping per ply, keeping the move generation state by consecutively disabling victims, and per victim, its aggressors. After making, processing and unmaking move, the aggressor's from-square of the current victim (to-square) is disabled, so that the next aggressor square is found in consecutive find aggressor cycles, until all from-squares are exhausted. Then, the victim to-square is disabled, and all origin squares of own pieces - always disjoint from their to-squares - are enabled again, to continue with a new find victim cycle until no more are found. In C like pseudo code with Bitboards the control flow looks as follows:

```
disable[ply] = 0;
while ( ( to = findMVV(disable[ply])) >= 0 ) {
  disable[ply] &= ~ownPieces;  /* enable all possible from-squares */
  while ( ( from = findLVA(disable[ply])) >= 0 ) {
    make(from, to);            /* ply++ */
    ....
    unmake(from, to);          /* ply-- */
    disable[ply] |= 1 << from; /* disable from-square */
  }
  disable[ply] |= 1 << to;     /* disable to-square */
}
```

Evaluation

Lazy

The fast incremental lazy evaluation was composed of a material register, a piece position register containing sum from hard wired piece-square tables, and king position registers, in conjunction with 9-bit registers indicating existence of friendly pawns on each wing, the weighted sums of king safety and king centralization, selected by the stage of the game. A conservative fast evaluation could quite often cause a beta-cutoff, without the need for a full evaluation.

Full

The slow and asynchronous full evaluation took eight cycles, and like the move generator, it consists of 64 similar circuits, to evaluate pawn structure and mobility along rays, also considering outposts and surrounding king squares.

Transposition Table

The Transposition Table was implemented within a separate microprocessor system with 1MByte of memory viewed as 128K 8-byte positions for the table, addressed by 16 of the incremental updated 48-bit hashkey bits and White to move. Four of the eight byte entries hold the remaining 32 hash code-bits to resolve index collisions.

Microcode

The microcode was implemented on a 1K x 64-bit microprocessor to perform an alpha-beta search using the move generator, evaluation and transposition table. A value bus was used to perform the minimal calculations, to implement the alpha-beta search and to gate 16-bit values to and from a stack, transposition table, evaluators and LSI-11 processor, with the sole operations of add and compare. A board address and piece bus were used to transfer board memories in the move generator, incremental- and full evaluator. Getting out of check did not count as a ply.

PVS

Recently [13], John Philip Fishburn gave some interesting internals on Belle's principal variation search:

```
However when I went to work at Bell Labs in 1981, Ken Thompson told me
```

that he had read the SIGART paper [14], and had sped up Belle by 1.5x with null windows.

The PVS algorithm in Belle did not do a second search at the root until a **second** fail high occurred. I don't know whether or not this idea appears in the literature or not. I would hope it does, but I haven't been following the literature for about 25 years. In other words, Belle is cleverly going for broke: it knows it's got a high failure, which is the best move so far, but as long as it doesn't get a second high failure, the first high failure remains the best move, and it can still avoid doing any more full searches.

Miscellaneous

Most components were Low Power Schottky (74LS) TTL logic, slightly slower than Schottky, but draws much less power, reducing power supply and cooling. The complete machine fits in a rack of 46cm x 38cm by 71cm tall. It weighs about 60kg and was portable.

In 1982, Belle was confiscated by the US State Department at Kennedy Airport when heading to the USSR to compete in a computer chess tournament. Its shipping was considered to be an illegal transfer of advanced technology to a foreign country [15] [16]. It took over a month and a $600 fine to get Belle out of customs [17].

See also

- Berkeley Chess Microprocessor
- Brutus
- CHEOPS
- ChipTest
- Deep Thought
- FPGA
- HiTech
- Hydra
- MBChess
- Thompson's Databases

Selected Publications

[18]

- "Belle" wurde auch US-Champion 1980: Frecher Schachzwerg beweist Kaltblütigkeit, January 23, 1981, Computerwoche 3/1981 (German) » ACM 1980
- Kevin O'Connell (**1981**). *Belle, Beller, Bellest*. Personal Computer World, July 1981 [19]
- Joe Condon, Ken Thompson (**1982**). *Belle Chess Hardware*. Advances in Computer Chess 3, reprinted (**1988**) in Computer Chess Compendium
- David Levy, Monroe Newborn (**1982**). All About Chess and Computers. 2nd edition, Springer, Postscript 1978-80 and Belle
- Joe Condon, Ken Thompson (**1983**). *BELLE*. Chess Skill in Man and Machine

Forum Posts

- Bebe and Belle by Joshua Lee, CCC, October 04, 1999 » Bebe
- Chip design project & another request for Belle/DT/DB info by Tom Kerrigan, CCC, January 27, 2000 » FPGA, Deep Thought, Deep Blue
- Chess 4.0 vs Belle 1973 by Joshua Lee, CCC, May 31, 2000 » Chess
- How high would the Rating of Belle realy be by Marc van Hal, CCC, September 22, 2001 [20] [21]
- Question about Program called Belle by O. Hall, CCC, September 01, 2008

External Links

- Belle (chess machine) from Wikipedia
- Belle's ICGA Tournaments
- Middle Game: Computer Chess Comes of Age - Brute Force vs Knowledge from The Computer History Museum
- The chess games of Belle (Computer) from chessgames.com
- QUEEN vs. ROOK by Warren Stenberg and Edware J. Conway, reprinted from the January, 1979 issue of the Minnesota Chess Journal, The Usenet Oldnews Archive, Compilation Copyright (C) 1981, 1996 Bruce Jones, Henry Spencer, David Wiseman » ACM 1978, Endgame Tablebases, Walter Browne
- BELLE, Baczynskyj, and Bisguier by Bruce Till, August 06, 2009, Chess.com » Boris Baczynskyj
- Ritchie & Thompson - Belle for Chess from I Programmer - programming, reviews and projects, January 26, 2011
- My Games Against World Champions (Part 1) by Dana Mackenzie, Dana Blogs Chess, April 24, 2012
- Classic Computer Chess - ... The programs of yesteryear by Carey, hosted by the Internet Archive [22]

References

1. ^ Creator of Belle computer chess dies at 76, NJ.com, March 02, 2012
2. ^ 3rd World Computer Chess Championship
3. ^ Bellechess-playing computer, Date: 1977 from The Computer History Museum
4. ^ Dennis Ritchie (**2001**). Ken, Unix, and Games, ICGA Journal, Vol. 24. No. 2
5. ^ Editor (**1979**). *Will Blitz be next year's champ?* Personal Computing, Vol. 3, No. 7, pp. 80, July

1979

6. ^ Militärischer Wert Der Spiegel 24/1982, pdf (German)
7. ^ Photo by Monroe Newborn from The Computer History Museum
8. ^ John Moussouris, Jack Holloway, Richard Greenblatt (**1979**). CHEOPS: A Chess-orientated Processing System. Machine Intelligence 9 (Jean Hayes Michie, Donald Michie and L.I. Mikulich editors) Ellis Horwood, Chichester, 1979, pp. 351-360. Reprinted (**1988**) in Computer Chess Compendium
9. ^ Feng-hsiung Hsu (**1999**). *IBM's Deep Blue Chess Grandmaster Chips*. pdf
10. ^ Marc Boulé (**2002**). *An FPGA Move Generator for the Game of Chess*. Masters thesis, McGill University, (Supervisor: Zeljko Zilic, Co-Supervisor: Monty Newborn), pdf
11. ^ Chrilly Donninger, Alex Kure, Ulf Lorenz (**2004**). *Parallel Brutus: The First Distributed, FPGA Accelerated Chess Program*. IPDPS'04, pdf
12. ^ Joe Condon, Ken Thompson (**1982**). *Belle Chess Hardware*. Advances in Computer Chess 3, Reprinted (**1988**) in Computer Chess Compendium
13. ^ John Philip Fishburn, note in September 2010
14. ^ John Philip Fishburn (**1980**). *An optimization of alpha-beta search*, SIGART Bulletin, Issue 72
15. ^ Militärischer Wert Der Spiegel 24/1982, pdf (German)
16. ^ Re: Microprocessors for Russian Shuttle, etc by Bruce C. Wright @ Duke University, net.space, June 22, 1982
17. ^ Belle (chess machine) from Wikipedia
18. ^ ICGA Reference Database (pdf)
19. ^ Publication Archive from Chess Computer UK by Mike Watters
20. ^ Jan Hein Donner vs Belle (Computer) (1982) from chessgames.com
21. ^ Jan Hein Donner from Wikipedia
22. ^ Re: Old programs CHAOS and USC by Dann Corbit, CCC, July 11, 2015

What links here?

Page	Date Edited
ACM 1973	Jan 19, 2018
ACM 1974	Jan 19, 2018
ACM 1978	Dec 23, 2017
ACM 1979	Apr 3, 2017
ACM 1980	Dec 25, 2017
ACM 1981	Jan 19, 2018
ACM 1982	Jul 19, 2016
ACM 1984	Jul 19, 2016
ACM 1986	Dec 5, 2017
ACM 1987	Dec 5, 2017
ACM 1990	Jun 7, 2016
ACM North American Computer Chess Championship	Jun 13, 2015
Alexander Szabo	Nov 10, 2012
Awit	Dec 22, 2017
Barend Swets	Jan 7, 2016

Page	Date Edited
Bebe	Dec 23, 2017
Bell Laboratories	Jun 20, 2017
Belle	Jan 18, 2018
Berkeley Chess Microprocessor	May 22, 2013
Bitboard Serialization	Dec 24, 2014
Blitz	Dec 23, 2017
Boris Baczynskyj	Oct 1, 2016
Brutus	Dec 24, 2016
CHAOS	Dec 28, 2017
CHEOPS	Apr 18, 2015
Ches	Jan 3, 2018
Chess (Program)	Dec 22, 2017
Chess Challenger	Dec 27, 2017
Chess Skill in Man and Machine	Nov 12, 2014
ChipTest	Jun 19, 2016
Combinatorial Logic	Apr 6, 2017
Computer Chess Compendium	Dec 29, 2015
Computerworld	Sep 9, 2017
Cyrus	Jul 22, 2014
Dana Mackenzie	Jan 18, 2018
Depth	Feb 25, 2018
Dieter Steinwender	Jul 4, 2016
Duchess	Dec 22, 2017
Edward Fredkin	Oct 27, 2013
Endgame Tablebases	Mar 6, 2018
Engines	Mar 10, 2018
FPGA	Jul 9, 2016
Fred Popowich	Jul 25, 2017
Fred Swartz	Jul 19, 2016
Frederic Friedel	Jan 6, 2018
Frieder Schwenkel	Dec 23, 2017
Ga Tech CP	Oct 8, 2014
Hardware	Jan 20, 2018
Harry Shershow	Dec 27, 2017
History	Jan 2, 2018
ICGA	Jul 8, 2017
Ira Ruben	Jul 19, 2016
Jaap van Oosterwijk Bruyn	May 22, 2016
Joe Condon	Jan 7, 2016
Ken Thompson	Sep 8, 2017
Kevin O'Connell	Dec 25, 2017
L'Excentrique	Jan 7, 2016
L. Stephen Coles	Jan 5, 2017
La Regence	Jul 23, 2015
Lachex	Jan 7, 2016

Page	Date Edited
Mathematician	Apr 9, 2018
MBChess	Sep 29, 2012
Move Generation	Jan 29, 2018
MVV-LVA	Oct 26, 2017
Netherlands-vs-Computers-1989	Mar 10, 2017
Nuchess	Apr 3, 2017
Null Window	Nov 8, 2016
Opening Book	Feb 26, 2018
Parabelle	Jul 28, 2017
PDP-11	Jan 5, 2016
Perft	Sep 26, 2017
Personal Computer World	Apr 12, 2013
Personal Computing	Dec 27, 2017
Piece Recognition	Sep 8, 2017
Principal Variation Search	Oct 22, 2017
Repetitions	Jan 16, 2018
Sequential Logic	May 10, 2017
Thompson's Databases	Dec 24, 2014
United States Open Computer Chess Championship	Dec 2, 2017
USOCCC 1985	Sep 1, 2017
WCCC 1977	Dec 22, 2017
WCCC 1980	Dec 25, 2017
WCCC 1983	Jan 20, 2018
Who's Who	Sep 6, 2017
World Computer Chess Championship	Mar 6, 2018

Up one Level