

[Home](#) \* [Engines](#) \* [Lachex](#)



Los Alamos <sup>LI</sup> Chess Experiment

### **Lachex,**

an Acronym for [Los Alamos](#) Chess Experiment, is the chess program developed by [Burton Wendroff](#) and [Tony Warnock](#). Lachex was a [mainframe program](#), playing on a [Cray X-MP 48](#), and was written in [Fortran](#) and [Assembly](#), performing some kind of full width [parallel principal variation search](#) inside an [iterative deepening framework](#). Also, Lachex already implemented presumably none [recursive null move pruning](#), as mentioned in the [description](#) from the [WCCC 1989](#) booklet.

## **Table of Contents**

[Tournament Play](#)

[Selected Games](#)

[Description](#)

[Board Representation](#)

[BCH Hashing](#)

[See also](#)

[Publications](#)

[External Links](#)

[References](#)

[What links here?](#)

## **Tournament Play**

Lachex played two [World Computer Chess Championships](#) <sup>[2]</sup>, the [WCCC 1986](#) and [WCCC 1992](#), and four [ACM North American Computer Chess Championships](#), [ACM 1985](#), [ACM 1986](#), [ACM 1987](#) and [ACM 1991](#). It was runner up in 1986 only losing the eventual winner [Belle](#).

## Selected Games

[WCCC 1986](#), round 1, [Ostrich](#) - [Lachex](#) <sup>[3]</sup>

```
[Event "WCCC 1986"]
[Site "Cologne, Germany"]
[Date "1986.06.11"]
[Round "1"]
[White "Ostrich"]
[Black "Lachex"]
[Result "0-1"]
```

```
1.e4 e5 2.Nf3 Nc6 3.Bb5 a6 4.Bxc6 dxc6 5.d4 exd4 6.Qxd4 Qxd4 7.Nxd4 Nf
6 8.O-O
Bc5 9.c3 O-O 10.f3 Bd6 11.Bg5 c5 12.Bxf6 gxf6 13.Ne2 Be6 14.Nd2 Rfd8 1
5.Rfd1 Kh8
16.a4 b5 17.axb5 axb5 18.Rxa8 Rxa8 19.g3 f5 20.Kf2 Ra2 21.Rb1 Bd7 22.K
e3 Ra8
23.Re1 Kg8 24.f4 Re8 25.Kd3 fxe4+ 26.Nxe4 Bf5 27.Nc1 Bf8 28.b3 c4+ 29.
bxc4 bxc4+
30.Kxc4 Bxe4 31.Rd1 c6 32.Rd2 Bd5+ 33.Kd3 Bc5 34.Re2 Rb8 35.c4 Be6 36.
Kc3 Bb4+
37.Kc2 Bxc4 38.Re5 f6 39.Rf5 Rd8 40.Rxf6 Rd2+ 41.Kb1 Bc3 0-1
```

## Description

from the [WCCC 1989](#) booklet <sup>[4]</sup>:

Lachex is specifically designed for the architecture of the Cray XMP and YMP series of machines. The highly repetitive parts of the program are written in [assembly language](#), the rest in [Fortran](#). Low level parallelism is achieved by extensive use of vector functional units and [pipelining](#). High level parallelism is obtained by means of multiple independent processors splitting up the search using a self-scheduling algorithm and communicating with each other through a large common [memory](#).

The search is basically [alpha-beta](#) with [iterative deepening](#). In the initial depth one search each [root](#) move is actually [scored](#) and the list of moves ordered accordantly. [Best moves](#) at subsequent iterations are moved to the top of the list. [Scouting](#) is used at [ply](#) one only - the first move in the list is scored and the remaining moves are tested with [minimal window](#). [Forward pruning](#) is done with a positional estimator at nodes below the horizon and with the [null move algorithm](#) above. Moves out of [check](#) above the horizon [extend](#) the [search depth](#) for that path by one, but by two if the check is [discovered](#) or [double](#). [Selective searches](#) below the horizon include [captures](#), [promotions](#), [castling](#), and some checking moves.

Lachex spends 1/3 of its time [generating moves](#), 1/3 doing bookkeeping, and 1/3 [evaluating leaf nodes](#). The evaluation function is symmetric wherever possible. [Mobility](#), [pawn structure](#), [king safety](#), [piece placement](#) and other features make up the evaluation function. Some strategy is incorporated at the root by shifting the minimal window to bias certain types of moves. There is a [transposition table](#) which can be as big as 32 million positions, on a 64 million word machine.

## Board Representation

Lachex used following [piece enumeration scheme](#) from the [initial position](#), which does not change during the cause of the game: the a1-rook was labeled with 1, b1-knight with 2, a2-pawn with 9, the a8-rook with 17 and the h7-pawn with 32. Beside a [bitboard board-definition](#) using 12 piece [bitboards](#) and [occupancy](#) as union set, Lachex used a redundant [8x8 board array](#), containing those 1..32 piece-codes, but zero for empty squares, and [piece-arrays](#) containing [squares](#) and associated [piece-types](#) or zero if the piece is missing. A so called *INC-Array* indexed by [origin](#), [target square](#) and [piece type](#) (excluding color, but white and black pawn as distinct types) was used in direct [move generation](#), similar as described in [In Between and Attacked by Piece on Square](#).

Further, something which reminds on [fill algorithms](#) like [Dumb7Fill](#) was used as described by Wendroff <sup>[5]</sup> :

There are several methods for generating the moves of the long range pieces. The method we have had the most success with on Cray machines preceding the X-MP/48 finds the to-squares closest to the home square, and then by a complicated sequence of shifts and boolean operations simultaneously continues these moves in the appropriate directions.

## BCH Hashing

To index and lock [search tables](#), especially the [transposition table](#), Lachex utilized signatures of chess positions by [BCH Hashing](#) based on [BCH-Code](#) as used in [Error detection and correction](#), otherwise [incrementally updated](#) similar than signatures from [Zobrist Hashing](#). Lachex used a BCH signature length of 81 (or more) bits to protect 16 bits from the full position string of 749 ( $64 \cdot 12 - 2 \cdot 2 \cdot 8 + 4 + 8 + 1$ ) bits, which is sufficient to avoid [collisions](#) within an 8 ply search. In their paper on *Search Tables* <sup>[6]</sup>, Warnock and Wendroff further elaborate on [alpha-beta inconsistencies](#), and that with the introduction of search tables, depending on the implementation, alpha-beta may not be order-independent. They refer implementations given by [Tony Marsland](#) <sup>[7]</sup> and [Harry Nelson](#) <sup>[8]</sup>.

## See also

- [ACM 1991 | Mephisto - Lachex](#)
- [Acronym](#)

## Publications

- [Burton Wendroff](#) (1985). *Attack Detection and Move Generation on the X-MP/48*. [ICCA Journal](#), Vol. 8, No. 2
- [Tony Warnock](#), [Burton Wendroff](#) (1988). *Search Tables in Computer Chess*. [ICCA Journal](#), Vol. 11, No. 1

## External Links

- [Lachex's ICGA Tournaments](#)
- [Lachex chess games](#) from [365Chess.com](#)

## References

1. <sup>^</sup> [Los Alamos National Laboratory from Wikipedia](#)
2. <sup>^</sup> [Lachex's ICGA Tournaments](#)
3. <sup>^</sup> [Cologne 1986 - Chess - Round 1 - Game 3 \(ICGA Tournaments\)](#)
4. <sup>^</sup> [Kings Move - Welcome to the 1989 AGT World Computer Chess Championship](#). Edmonton, Alberta, Canada, Courtesy of [Peter Jennings](#), from [The Computer History Museum](#), pdf
5. <sup>^</sup> [Burton Wendroff](#) (1985). *Attack Detection and Move Generation on the X-MP/48*. [ICCA Journal](#), Vol. 8, No. 2
6. <sup>^</sup> [Tony Warnock](#), [Burton Wendroff](#) (1988). *Search Tables in Computer Chess*. [ICCA Journal](#), Vol. 11, No. 1
7. <sup>^</sup> [Tony Marsland](#) (1986). *A Review of Game-Tree Pruning*. [ICCA Journal](#), Vol. 9, No. 1
8. <sup>^</sup> [Harry Nelson](#) (1985). *Hash Tables in Cray Blitz*. [ICCA Journal](#), Vol. 8, No. 1

## What links here?

Page	Date Edited
<a href="#">ACM 1985</a>	Jun 21, 2017
<a href="#">ACM 1986</a>	Dec 5, 2017
<a href="#">ACM 1987</a>	Dec 5, 2017
<a href="#">ACM 1991</a>	Nov 25, 2017
<a href="#">BCH Hashing</a>	May 8, 2017
<a href="#">Bebe</a>	Dec 23, 2017
<a href="#">Burton Wendroff</a>	Nov 21, 2017
<a href="#">Cray X-MP</a>	Jul 26, 2017
<a href="#">Delicate Brute</a>	Jun 7, 2016
<a href="#">Engines</a>	Mar 10, 2018
<a href="#">Lachex</a>	Jan 7, 2016
<a href="#">Los Alamos National Laboratory</a>	Sep 16, 2015
<a href="#">Mathematician</a>	Apr 9, 2018
<a href="#">Netherlands-vs-Computers-1989</a>	Mar 10, 2017
<a href="#">Persistent Hash Table</a>	Dec 31, 2017
<a href="#">Pieces</a>	Feb 19, 2018
<a href="#">Promotions</a>	Jun 6, 2017
<a href="#">Singular Extensions</a>	Jan 9, 2018
<a href="#">Tony Warnock</a>	Jan 7, 2016
<a href="#">Vaxchess</a>	Jan 7, 2016
<a href="#">WCCC 1986</a>	Jul 27, 2017
<a href="#">WCCC 1992</a>	Feb 26, 2018
<a href="#">Young Brothers Wait Concept</a>	Apr 22, 2018

[Up one level](#)