

Table of Contents

[Move Generation](#)

[Evaluation](#)

[Selected Games](#)

[Bebe](#)

[Cray Blitz](#)

[Sun Phoenix](#)

[See also](#)

[Publications](#)

[External Links](#)

[References](#)

[What links here?](#)

[Home](#) * [Engines](#) * **ChipTest**

ChipTest was a chess program running on a [Sun-3 workstation](#) using a high speed [move generator](#) in [hardware](#). It was the predecessor of [Deep Thought](#), later emerged to [Deep Blue](#). The project started in 1985 by two students at [Carnegie Mellon University](#), [Feng-hsiung Hsu](#) who did the chip design of the move generator ^[1], and [Thomas Anantharaman](#). Later in 1986 they were joined by former [HiTech](#) member [Murray Campbell](#). ChipTest played two [ACM North American Computer Chess Championships](#), [ACM 1986](#) and [ACM 1987](#), and it won the latter with a perfect score. At ACM 1986, ChipTest searched about 100K [positions per second](#) ^[2], in 1987 500K ^[3], already employing [singular extensions](#) ^[4].

Move Generation

The [move generator](#), a [combinational logic](#) 8x8 [array](#) is effectively a silicon chessboard in [VLSI design](#). The basic move generation algorithm is the same as in the [Belle move generator](#), where a disable-stack implements the bookkeeping of victims and per victim, of aggressors. In ChipTest and its successors, the last move searched from the position, which need to be stored for [unmake move](#) anyway, implies the priority levels of the last victim and the last attacking piece is used for the [sequential logic](#) to compute the

disable-mask on the fly, using distinct square priorities to discriminate equal valued victims and aggressors [\[5\]](#) [\[6\]](#).

```
disable = 0;
while ( ( to = findMVV(disable)) >= 0 ) {
    while ( ( from = findLVA(disable)) >= 0 ) {
        make(from, to);
        ....
        unmake(from, to);
        disable = allLessAndEqualValuableAttackers(from);
    }
    disable = allMoreAndEqualValuableVictims(to);
}
```

In software, the cost of recomputing the mask exceeds the cost of retrieving it. But in hardware, the reverse holds true. The disable-stack needs decoders to operate anyway. The modified decoder operates at a speed comparable to the disable-stack decoders and takes less space. This method avoids using the disable-stack, which probably needs to be at least 64 levels deep.

Evaluation

While features such as [material](#), [piece placement](#), and some [pawn structures](#) are easier to evaluate in an [incremental](#) way, certain more subtle features require a whole board approach to do the [evaluation](#) in hardware - aggregated [square control](#) aka [mobility](#) of a [Chess 4.5](#) like approach [\[7\]](#), and [pins](#) [\[8\]](#).

Selected Games

Bebe

[ACM 1986](#), round 1, [Bebe](#) - [ChipTest](#) [\[9\]](#)

```
[Event "ACM 1986"]
[Site "Dallas USA"]
[Date "1986.11.02"]
[Round "1"]
[White "Bebe"]
[Black "Chiptest"]
[Result "1-0"]
```

1.e4 e6 2.d4 d5 3.Nd2 dxe4 4.Nxe4 Qd5 5.Nc3 Bb4 6.Nge2 Qd8 7.a3 Bd6
8.g3 Nf6 9.Bg2 Nc6 10.O-
O Bd7 11.Bg5 h6 12.Bxf6 Qxf6 13.Ne4 Qf5 14.d5 exd5
15.Nxd6+ cxd6 16.Nf4 O-O-
O 17.Re1 d4 18.Bxc6 Bxc6 19.Qxd4 g5 20.Qd3 Qf6
21.Nd5 Qxb2 22.Rab1 Qg7 23.Ne7+ Kc7 24.Nxc6 Rd7 25.Qb5 b6 26.Nxa7 Qd4
27.Qc6+ Kb8 28.Qxd7 Qd5 29.Nc6+ Qxc6 30.Qxc6 d5 31.Rxb6+ Ka7 32.Qb7# 1
-0

Cray Blitz

[ACM 1987](#), round 3, [Cray Blitz](#) - [Chiptest M](#) ^[10]

```
[Event "ACM 1987"]  
[Site "Dallas, USA"]  
[Date "1987.10.26"]  
[Round "3"]  
[White "Cray Blitz"]  
[Black "Chiptest M"]  
[Result "0-1"]
```

1.e4 d5 2.exd5 Qxd5 3.Nc3 Qa5 4.d4 c6 5.Nf3 Nf6 6.Bc4 Bg4 7.h3 Bh5
8.Qe2 Nbd7 9.Bd2 Qc7 10.g4 Bg6 11.O-O-O O-O-O 12.Ng5 e5 13.Bxf7 exd4
14.Na4 Ne5 15.Bxg6 Nxg6 16.Ne6 Re8 17.Rhe1 Qd6 18.g5 Nd7 19.Qg4 b5
20.Nac5 Nge5 21.Nxf8 Rhxf8 22.Ne4 Qd5 23.Qg2 Re6 24.Kb1 Nf3 25.Qg4
Nxe1 26.Rxe1 Ne5 27.Qd1 Nf3 0-1

Sun Phoenix

[ACM 1987](#), round 4, [Chiptest M](#) - [Sun Phoenix](#) ^[11]

```
[Event "ACM 1987"]  
[Site "Dallas, USA"]  
[Date "1987.10.27"]  
[Round "4"]  
[White "Chiptest M"]  
[Black "Sun Phoenix"]  
[Result "1-0"]
```

1.e4 c6 2.d4 d5 3.Nc3 g6 4.h3 Bg7 5.Nf3 Nf6 6.e5 Ne4 7.Nxe4 dxe4
8.Ng5 c5 9.dxc5 Qa5+ 10.c3 Qxc5 11.Qd4 Qxe5 12.Qxe5 Bxe5 13.Bc4 O-O

14.O-O Bd7 15.Rd1 Ba4 16.Re1 Nd7 17.Bd5 Nc5 18.Nxe4 Nd3 19.Re2 Bc6
20.Bxc6 Nxc1 21.Rxc1 bxc6 22.Nc5 Bf4 23.Rce1 Rfb8 24.g3 Bd6 25.Ne4
Bc7 26.f4 Rd8 27.Kg2 Rd5 28.c4 Rd4 29.c5 Ba5 30.Rf1 Rad8 31.Rff2 Bb4
32.a3 Ba5 33.Ng5 Re8 34.f5 Rd3 35.fxg6 fxg6 36.Nf3 Bc7 37.Re6 Bb8
38.Rxc6 Re3 39.g4 e5 40.Ng5 Rb3 41.Re6 Rxe6 42.Nxe6 h5 43.c6 e4
44.c7 Bxc7 45.Nxc7 1-0

See also

- [Belle](#)
- [Deep Thought](#)
- [Deep Blue](#)

Publications

- [Feng-hsiung Hsu](#) (1986). [Two designs of functional units for VLSI based chess machines](#). [Carnegie Mellon University](#), Computer Science Department. Paper 1566.
- [Feng-hsiung Hsu](#) (1987). *A Two-Million Moves/Sec CMOS Single-Chip Chess Move Generator*. IEEE J. of Solid-state Circuits, Vol. 22, No. 5, pp. 841-846
- [Thomas Anantharaman](#), [Murray Campbell](#), [Feng-hsiung Hsu](#) (1988). *Singular extensions: Adding Selectivity to Brute-Force Searching*. AAAI Spring Symposium, Computer Game Playing, pp. 8-13. Also published in [ICCA Journal, Vol. 11, No. 4](#), republished (1990) in [Artificial Intelligence](#), Vol. 43, No. 1, pp. 99-109. ISSN 0004-3702.

External Links

- [ChipTest from Wikipedia](#)
- [The chess games of Chiptest](#) from [chessgames.com](#)
- [Manu Dibango](#) - Sun Explosion, [YouTube](#) Video

References

1. [^] [Feng-hsiung Hsu](#) (1987). *A Two-Million Moves/Sec CMOS Single-Chip Chess Move Generator*. IEEE J. of Solid-state Circuits, Vol. 22, No. 5, pp. 841-846
2. [^] [The ACM's Seventeenth North American Computer Chess Championship and The Sixth World Microcomputer Chess Championship](#) from [The Computer History Museum](#), [pdf](#)
3. [^] [The ACM's Eighteenth North American Computer Chess Championship](#) from [The Computer History Museum](#), [pdf](#)
4. [^] [Thomas Anantharaman](#), [Murray Campbell](#), [Feng-hsiung Hsu](#) (1988). *Singular extensions: Adding Selectivity to Brute-Force Searching*. AAAI Spring Symposium, Computer Game Playing, pp. 8-13. Also published in [ICCA Journal, Vol. 11, No. 4](#), republished (1990) in [Artificial Intelligence](#), Vol. 43, No. 1, pp. 99-109. ISSN 0004-3702.

5. [^ Feng-hsiung Hsu \(1986\). Two designs of functional units for VLSI based chess machines. Carnegie Mellon University](#), Computer Science Department. Paper 1566.
6. [^ Feng-hsiung Hsu \(1999\). IBM's Deep Blue Chess Grandmaster Chips. IEEE Micro, Vol. 19, No. 2 \(Mar-Apr\), pp. 70-81. ISSN 0272-1732. pdf](#)
7. [^ David Slate, Larry Atkin \(1977\). Chess 4.5 - The Northwestern University Chess Program. Chess Skill in Man and Machine](#), reprinted (1988) in [Computer Chess Compendium](#)
8. [^ Feng-hsiung Hsu \(1986\). Two designs of functional units for VLSI based chess machines. Carnegie Mellon University](#), Computer Science Department. Paper 1566.
9. [^ PGN Download NACCC](#) from [Computerschaak/Downloads/Games](#) hosted by [CSVN](#)
10. [^ PGN Download NACCC](#) from [Computerschaak/Downloads/Games](#) hosted by [CSVN](#)
11. [^ Monty Newborn, Danny Kopec \(1988\). Results of ACM's Eighteenth Computer Chess Championship. Communications of the ACM](#), Vol. 31, No. 8, [pdf](#)

What links here?

Page	Date Edited
ACM 1986	Dec 5, 2017
ACM 1987	Dec 5, 2017
ACM North American Computer Chess Championship	Jun 13, 2015
Aegon 1989	Jan 22, 2017
Bebe	Dec 23, 2017
Belle	Jan 18, 2018
Carnegie Mellon University	Feb 12, 2018
ChipTest	Jun 19, 2016
Deep Blue	Aug 1, 2017
Deep Thought	Dec 5, 2016
Engines	Mar 10, 2018
Eval Tuning in Deep Thought	Jun 7, 2016
Feng-hsiung Hsu	Dec 27, 2016
Hardware	Jan 20, 2018
Harvard Cup 1989	Oct 14, 2015
Mate Threat Extensions	Apr 14, 2012
Mathematician	Apr 9, 2018
Move Generation	Jan 29, 2018
Murray Campbell	Jun 2, 2017
Netherlands-vs-Computers-1989	Mar 10, 2017
Singular Extensions	Jan 9, 2018
Sun	Jun 7, 2016
Thomas Anantharaman	Jan 27, 2016

[Up one Level](#)