



**AN EFFICIENT CHINESE CHECKERS IMPLEMENTATION:
RANKING, BITBOARDS, AND BMI2
PEXT AND PDEP INSTRUCTIONS**

Nathan R. Sturtevant
University of Alberta
Canada CIFAR Chair, Amii



MOVING AI

Computers and Games 2022



GAME-PLAYING PROGRAMS

- Speed = Performance
 - Faster game engine = more wins
- Important when solving a game
- Simplify/speed move generation with bit operations

SELECTED BITBOARD HISTORY

- Chess
 - Adel'son-Vel'skii et al, 1970
 - Frey, 1983
- Shogi
 - Grimbergen, 2007
- Arimaa
 - Carlini et al, 2008
- General Overview
 - Browne, 2014

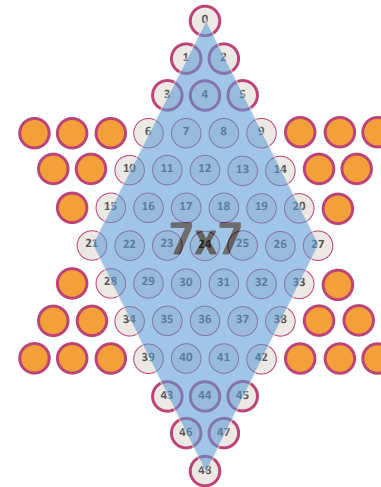
CHINESE CHECKERS

- Chinese Checkers Implementation
 - Original C++ code circa 1998
 - Slowly updated made more efficient
- Now: strongly solving game
 - Can we get even more performance?

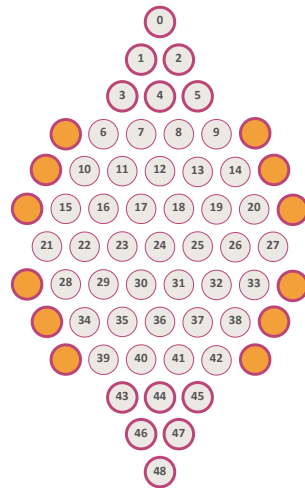
REMAINING TALK OUTLINE

- Chinese Checkers Move Generation
 - Step
 - Hop
- Experimental Results

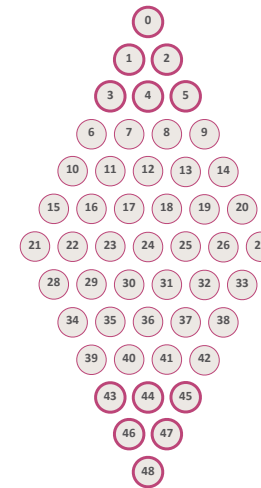
BITBOARD



BITBOARD



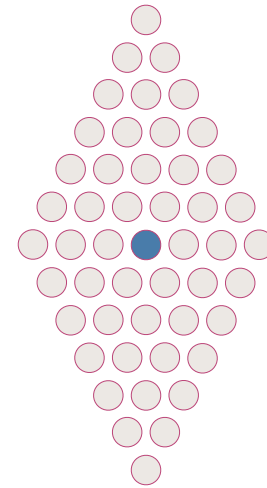
BITBOARD



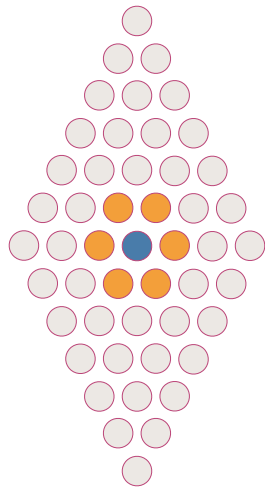
BOARD/MOVE REPRESENTATION



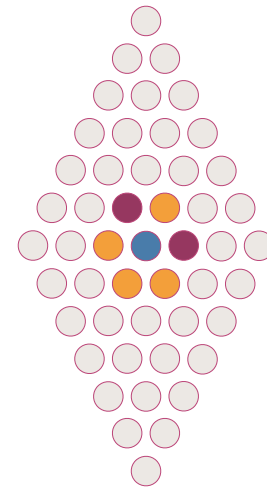
CHINESE CHECKERS STEP MOVE



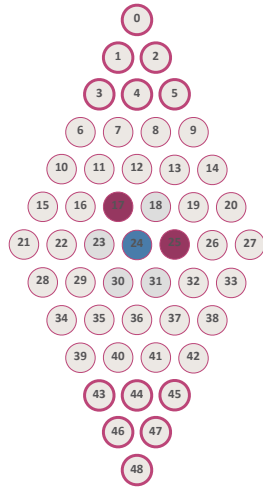
CHINESE CHECKERS STEP MOVE



CHINESE CHECKERS STEP MOVE



CHINESE CHECKERS STEP MOVE

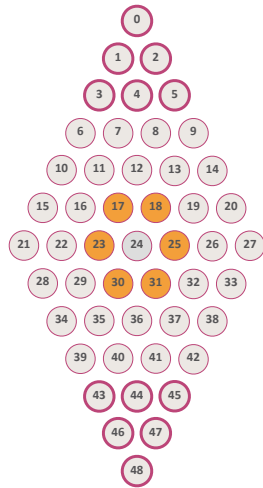


BOARD/MOVE REPRESENTATION

Pieces



CHINESE CHECKERS STEP MOVE MASK



BOARD/MOVE REPRESENTATION

Piece Mask



BOARD/MOVE REPRESENTATION

Pieces



Piece Mask



~Piece & Piece Mask



BOARD/MOVE REPRESENTATION

Piece



Piece Mask



~Piece & Piece Mask

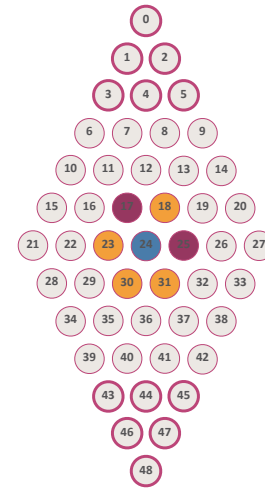


BOARD/MOVE REPRESENTATION

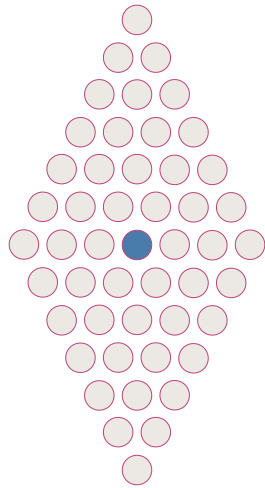
Final Moves (one uint64_t per piece)



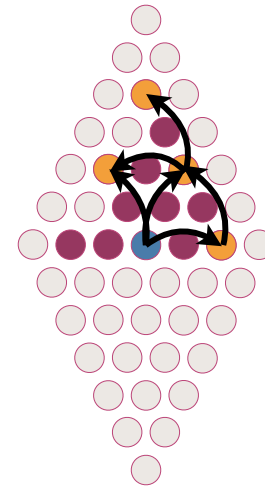
CHINESE CHECKERS STEP MOVE



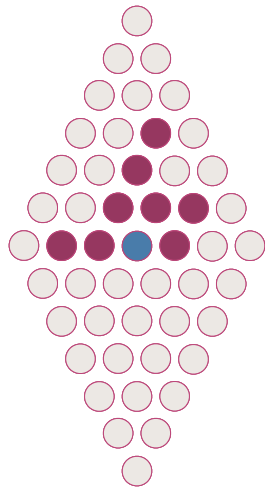
**CHINESE CHECKERS
HOP MOVE**



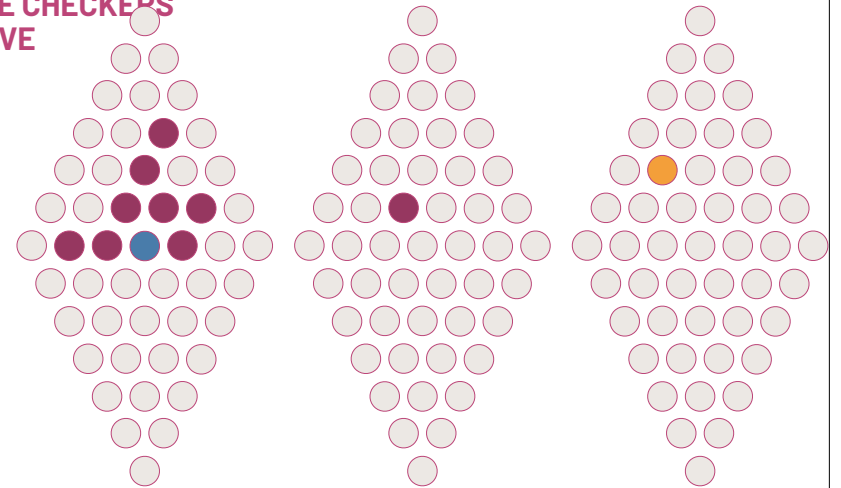
**CHINESE CHECKERS
HOP MOVE**



**CHINESE CHECKERS
HOP MOVE**



**CHINESE CHECKERS
HOP MOVE**



BOARD/MOVE REPRESENTATION

Board



Neighbors[24]



Jump To Locations[24]



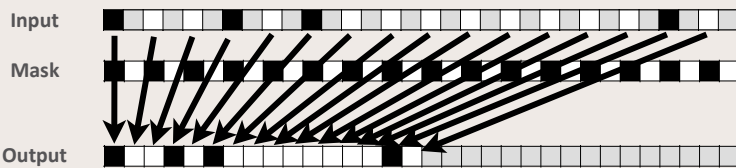
Final Moves



BMI2 - PDEP, PEXT

- BMI2 introduced in Haswell, 2013
 - Parallel Deposit
 - Parallel Extract
- Pack and unpack bits with selector mask

PARALLEL EXTRACT



OUTPUT ← PEXT(BOARD, NEIGHBORS[24])

Board



Neighbors[24]



Neighbor Output



OUTPUT ← PEXT(BOARD, NEIGHBORS[24])

Board



Neighbors[24]



Neighbor Output



OUTPUT ← PEXT(BOARD, JUMPTO[24])

Board



Jump To Locations[24]



Jump To Output



OUTPUT ← PEXT(BOARD, JUMPTO[24])

Board



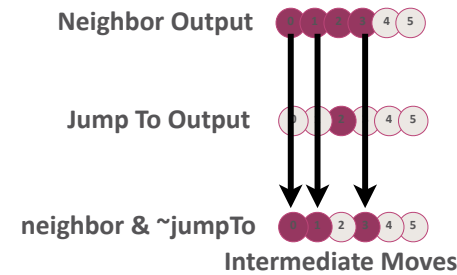
Jump To Locations[24]



Jump To Output



GET ALL JUMPS



MOVES ← PDEP(INTERMED., JUMPTO[24])

Board



Jump To Locations[24]



Intermediate Moves



Final Moves



MOVES ← PDEP(INTERMED., JUMPTO[24])

Board



Jump To Locations[24]



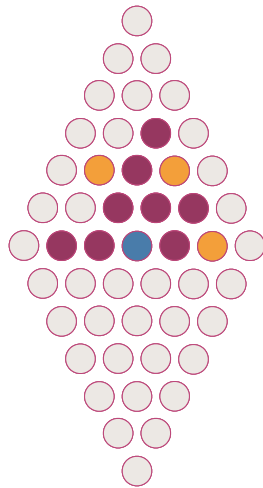
Intermediate Moves



Final Moves




CHINESE CHECKERS HOP MOVE



CHINESE CHECKERS BITBOARD OPERATION SUMMARY

- Get Winner - Goal Mask
- Get Moves - PDEP/PEXT
- Apply/Undo Move - XOR
- Ranking/Unranking (perfect hash)
 - Better combinatorial math $O(\text{pieces})$ vs $O(\text{board})$
 - PDEP/PEXT for second player



EXPERIMENTAL RESULTS

SPEEDUPS


Operation	Speedup
GetWinner	2.74
Succ/Apply	8.89
Rank P1	1.54
Rank P2	4.99
Unrank P1	2.48
Unrank P2	2.06

STRONG SOLVING TIME (7x7 BOARD)

Pieces	States	Symmetric States	Original	BitBoard	Speedup
3	559,352,640	141,219,540	204.51	74.12	2.8
4	63,136,929,240	15,822,357,347	43,187.58	21,597.82	2.0

CONCLUSIONS

- PDEP/PEXT instructions are useful/powerful
- Significant speedup in Chinese Checkers



MOVINGAI

<http://www.movingai.com/>