



Synopsys Users Group
UK 2012



Synopsys Users Group
UK 2012

Converting Existing Software to Hardware using SCC: A Case Study of an Open Source Connect-6 Solver

Sumanta Chaudhuri, Peter Y. K. Cheung
Imperial College, London

THES Ranking: 3rd in Europe and 8th in World

Facts & Figures

- Founded 1907
- 13,964 full-time students
- Students from 126 countries
- 242 taught courses
- Research Grants & Contracts Income 300 M£ (2010)

Outline

- **Goals of This Case-Study**
- **SynphonyCC**
 - A Brief Overview
- **The Game of Connect-6**
 - The Game
 - The Algorithm
- **Design**
 - Manual Intervention
 - Exposing Parallelism
- **Verification**
- **Results**

Goal of This Case-Study

- **Evaluate the ease of use.**
 - Push Button ?
 - Semi-Automatic ?
 - Manual Digging ?
- **Assess the quality of hardware generated.**
(Power & Performance, Slack Histogram, Interconnect Density....) ?

Potential Applications

- **Automatic Translation of Existing C Libraries to Hardware IP**

e.g.

- C64X-IMGLIB from TI,
 - OpenCV
-
- **Accelerated Time-to-Market**

- **A high level synthesis (HLS) tool
Based on PICO (Program In Chip Out)**

Kathail, V.; Aditya, S.; Schreiber, R.; Ramakrishna Rau, B.; Cronquist, D.C.; Sivaraman, M.; , "**PICO: automatically designing custom computers**," *Computer* , vol.35, no.9, pp. 39- 47, Sep 2002 doi: 10.1109/MC.2002.1033026

"Synphonycc,"

<http://www.synopsys.com/Systems/BlockDesign/HLS/Pages/SynphonyCompiler.aspx>.

SynphonyCC OverView

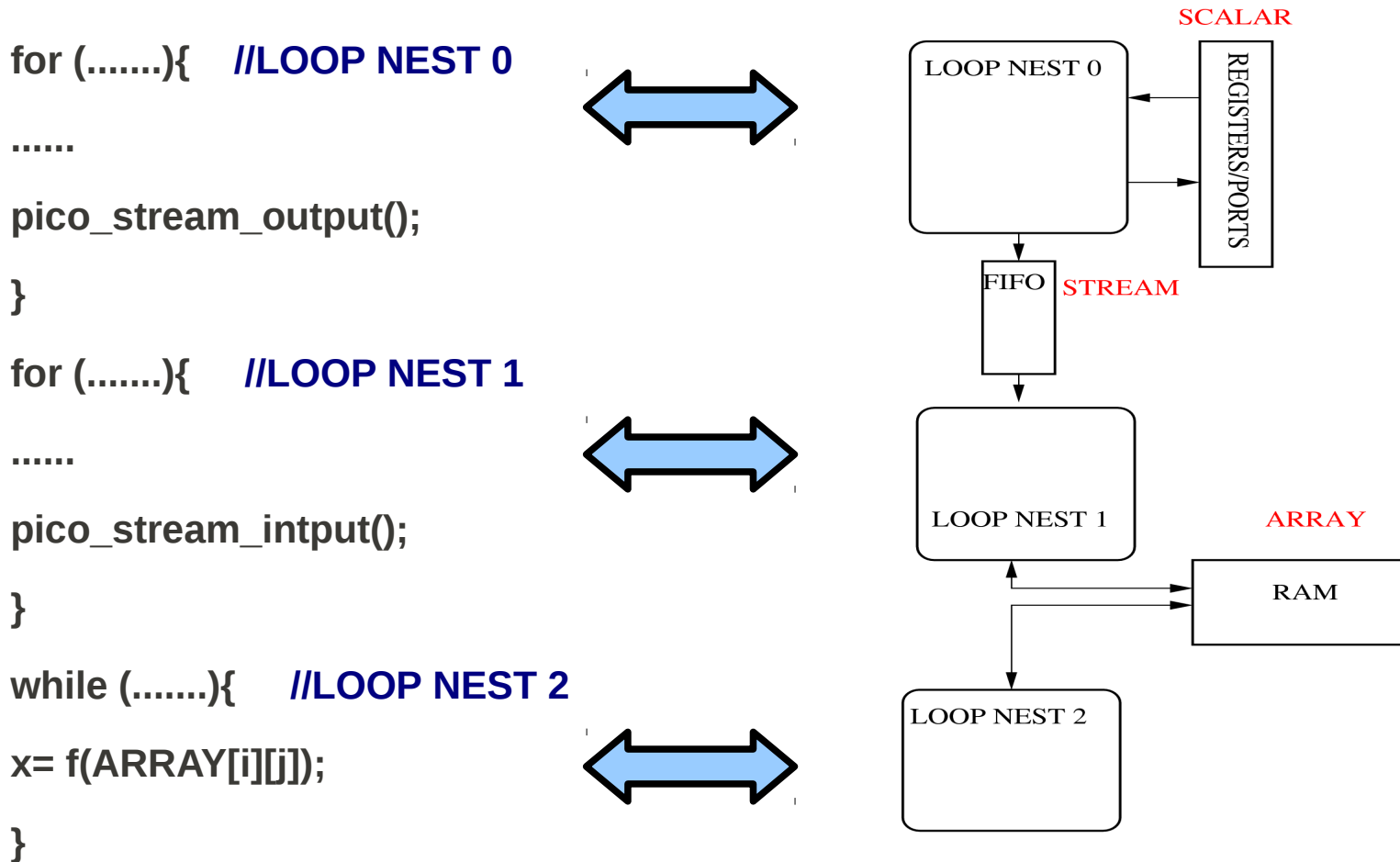


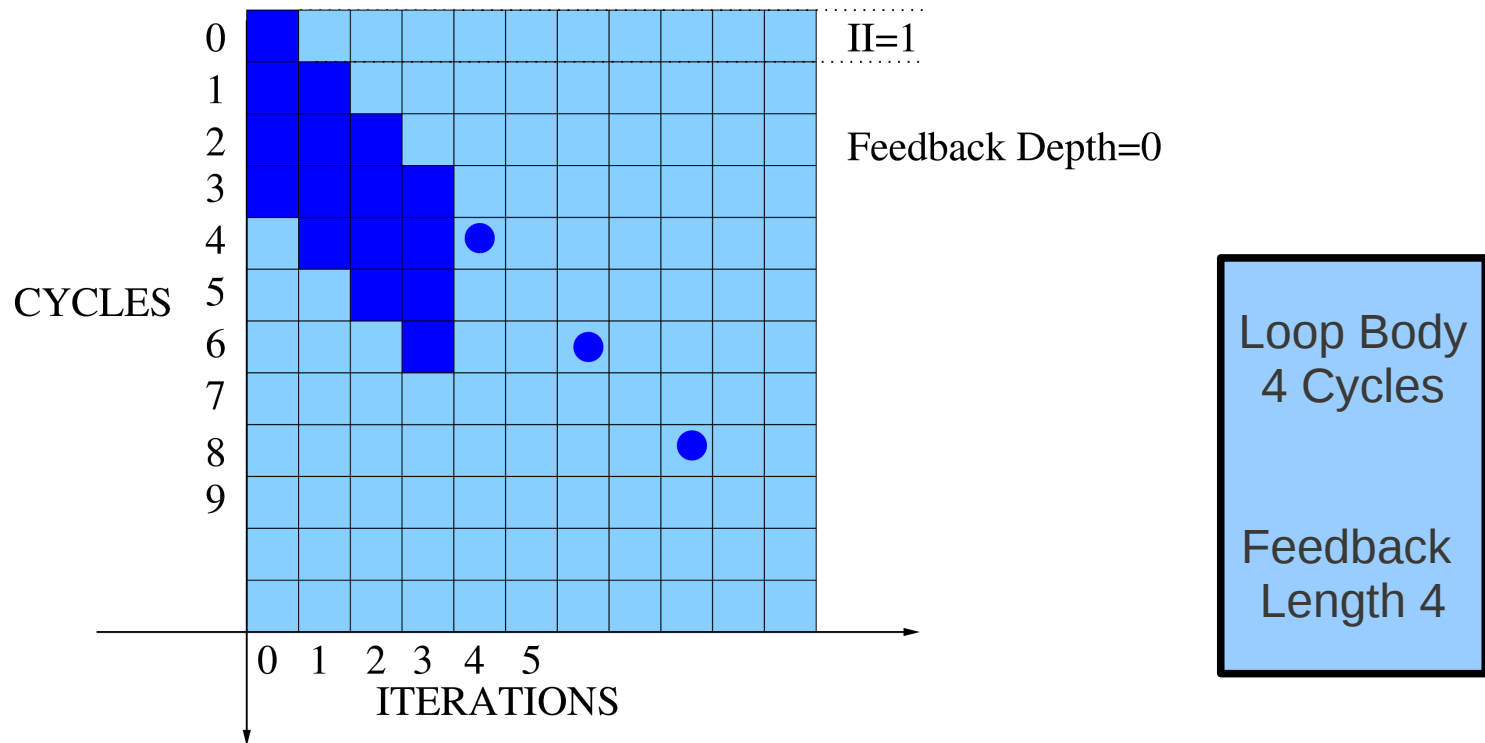
Figure 1. Untimed loopnests are scheduled and synthesized to cycle-accurate hardware, ILP within loopnest is extracted automatically

LoopNests

Parameters

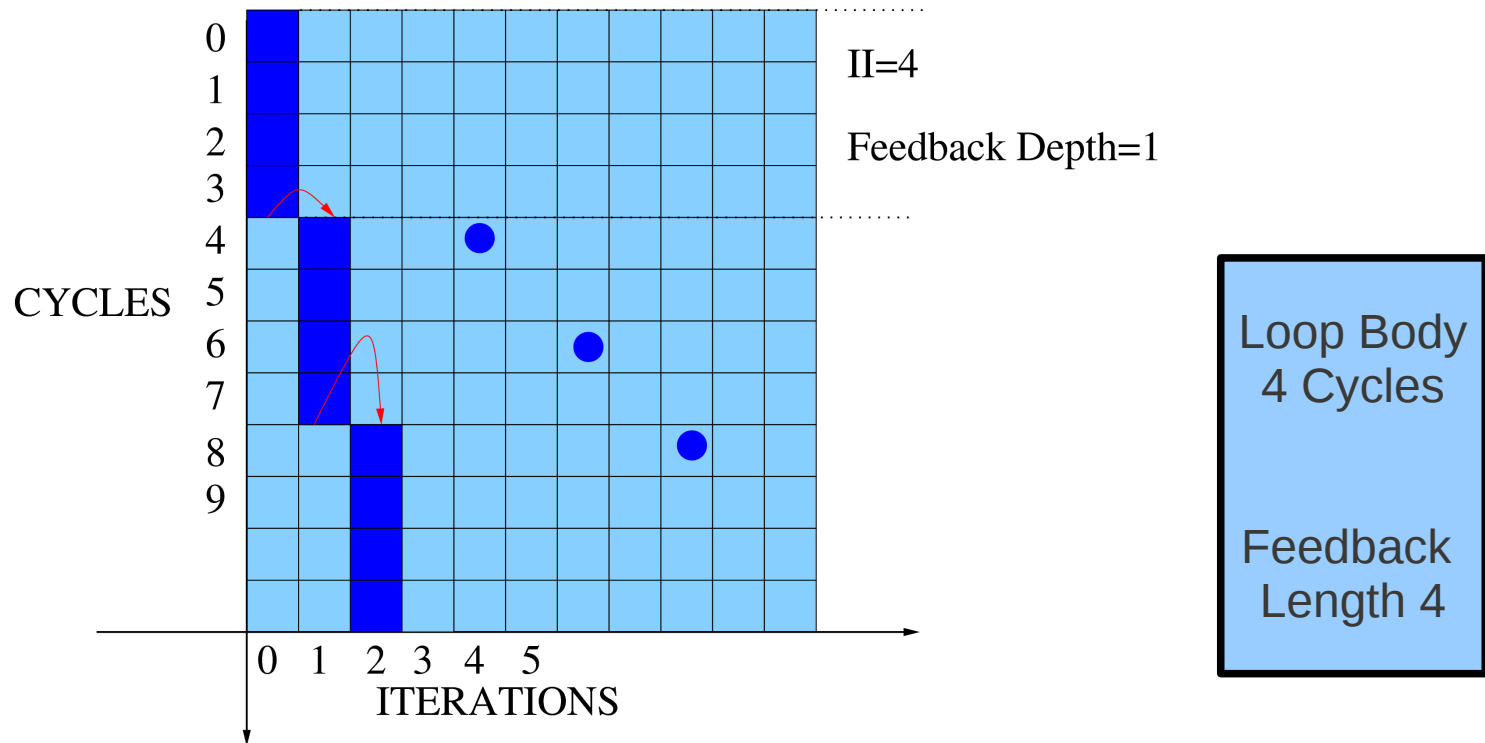
- Clock Frequency T (Constraint from User)
- Initiation Interval (II) (Constraint from User)
- Iteration Count (N) (Specified in the C code)
- Feedback Depth (Specified in the C code)
 - e.g $x[i]=x[i-2]+y[i]$; feedback depth=2
- Feedback Length (Characteristic of the Technology)
 - Sum of Operation delay around the feedback path
- Unroll Factor (Specified with Pragma)

LoopNests Parameters



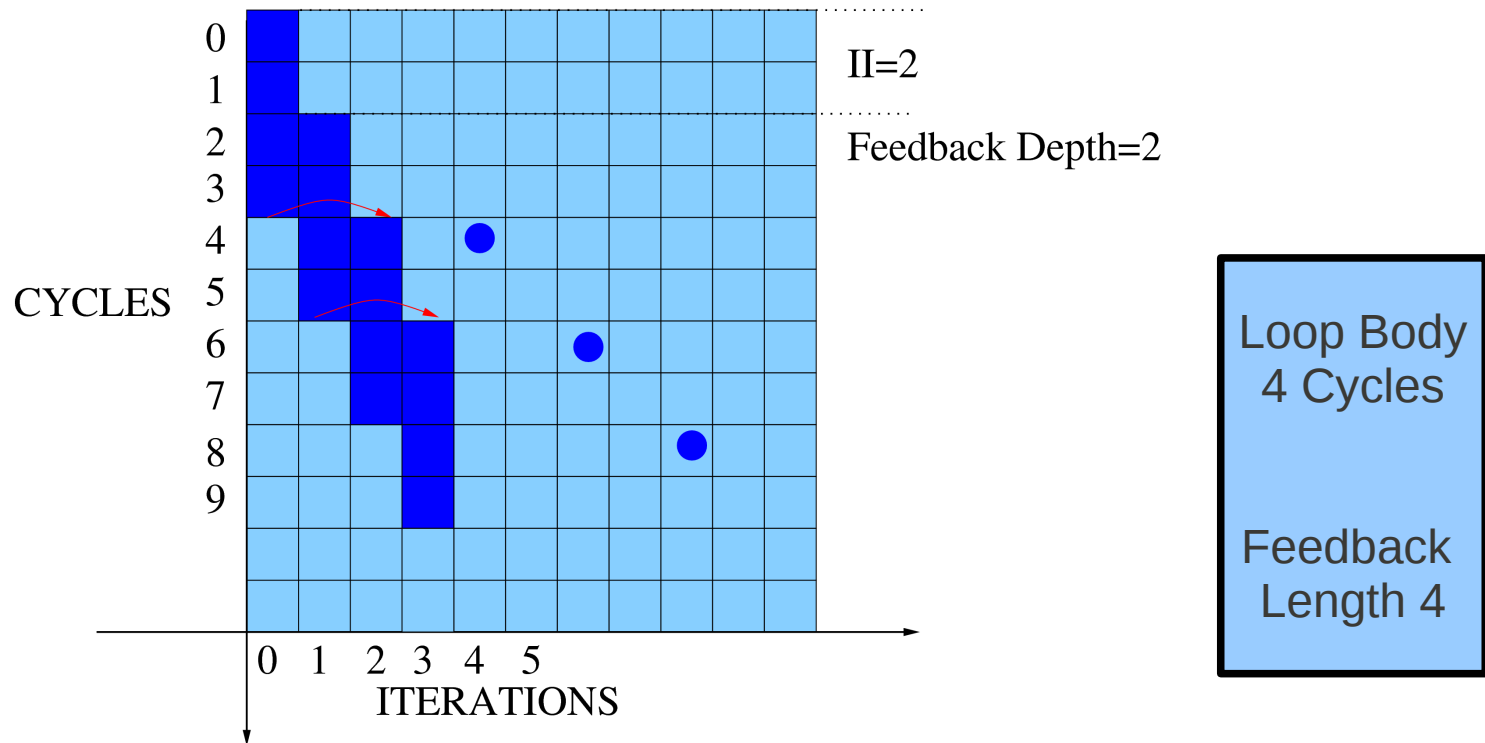
- Total Execution Time = $N * II$
- Condition to avoid feedback failure
 - (Total delay around feedback path) \leq (Feedback depth) * II

LoopNests Parameters



- Total Execution Time = $N * I$
- Condition to avoid feedback failure
 - (Total delay around feedback path) \leq (Feedback depth) * II

LoopNests Parameters

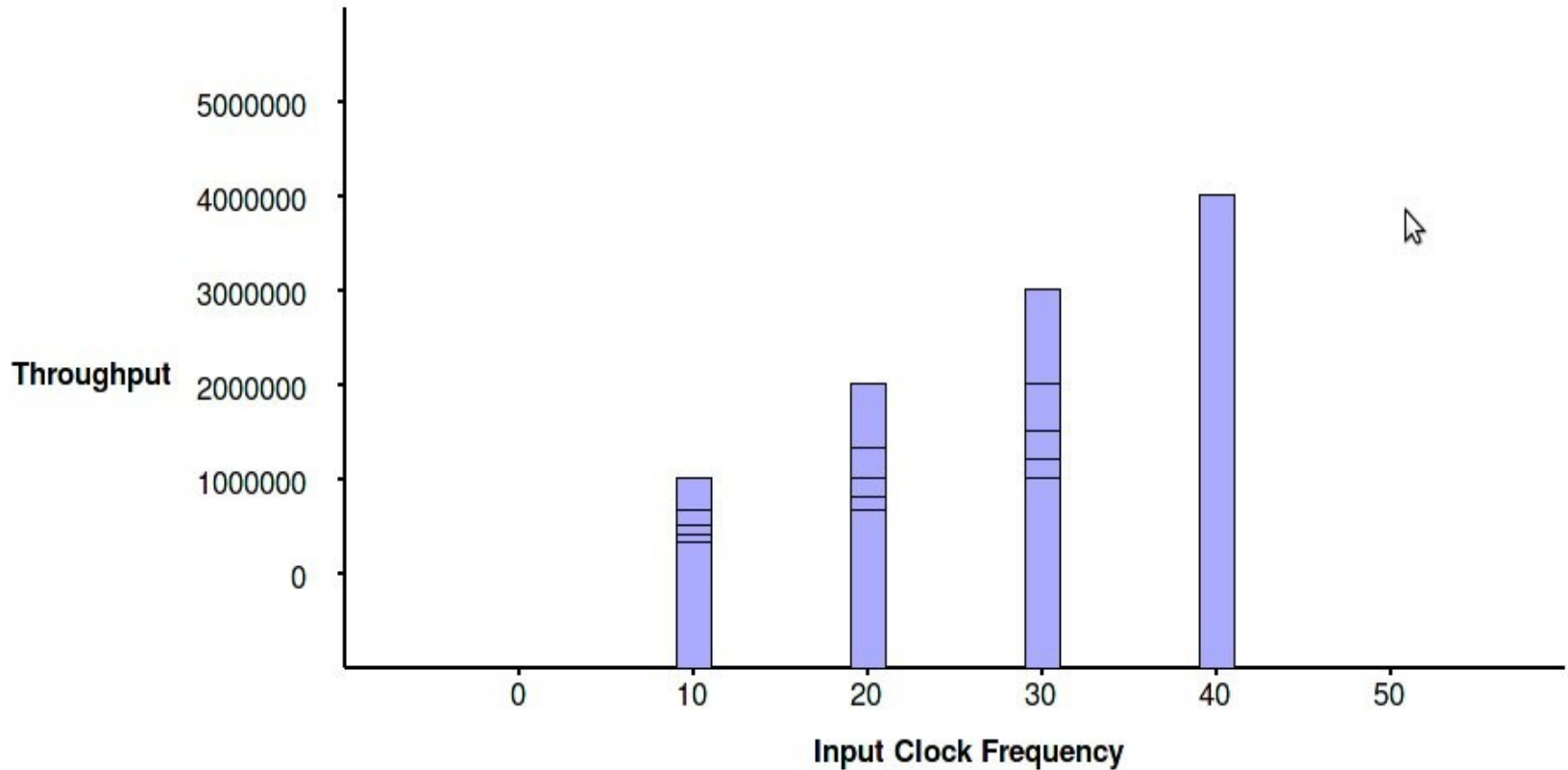


- Total Execution Time = $N * II$
- Condition to avoid feedback failure
 - (Total delay around feedback path) \leq (Feedback depth) * II

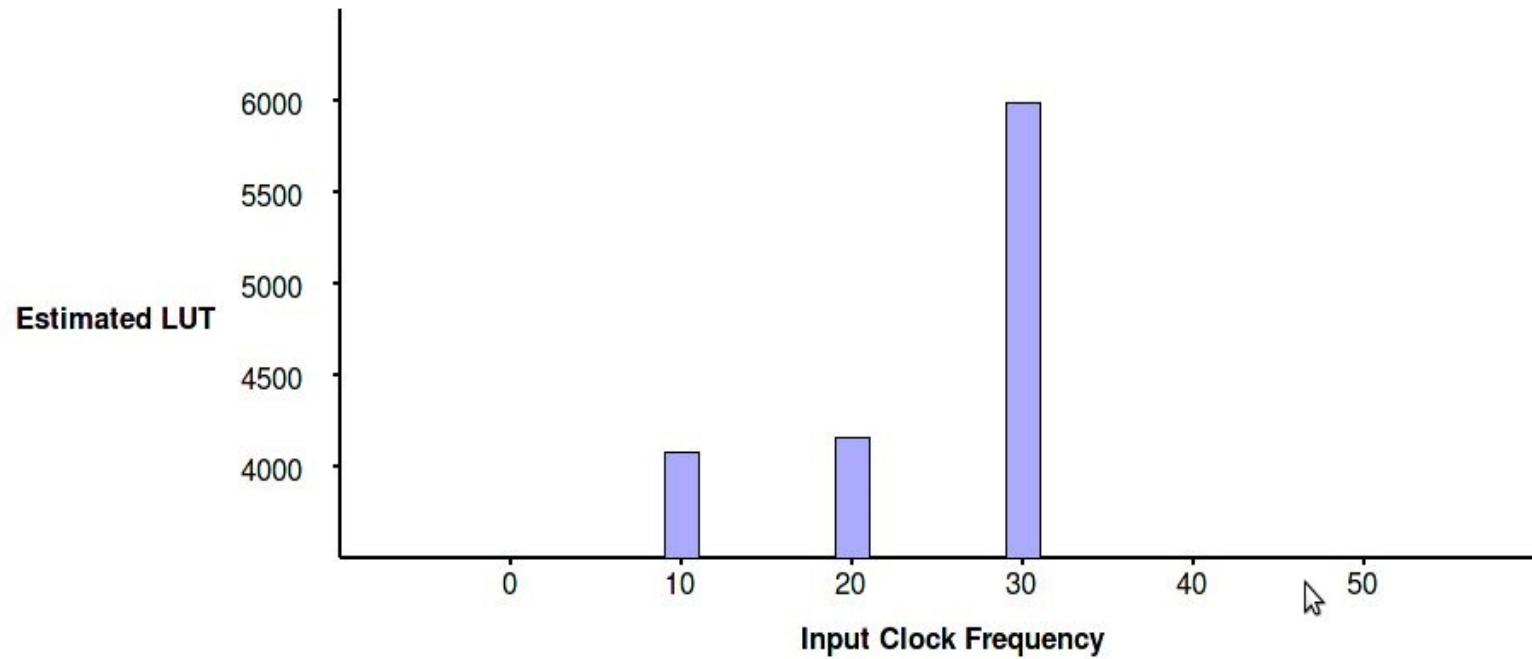
LoopNests

- An exploration of the loop parameters is possible from both GUI and Tcl scripts.
- Bit-Accurate ness has to be specified by user through Pragmas.
- Variable Trip count Loops:
 - Need to specify Max, Min and average no. of iterations.
 - `#pragma num_ iterations(<min>,<avg>,<max>)`
- Synthesized in independent stall domain

LoopNests



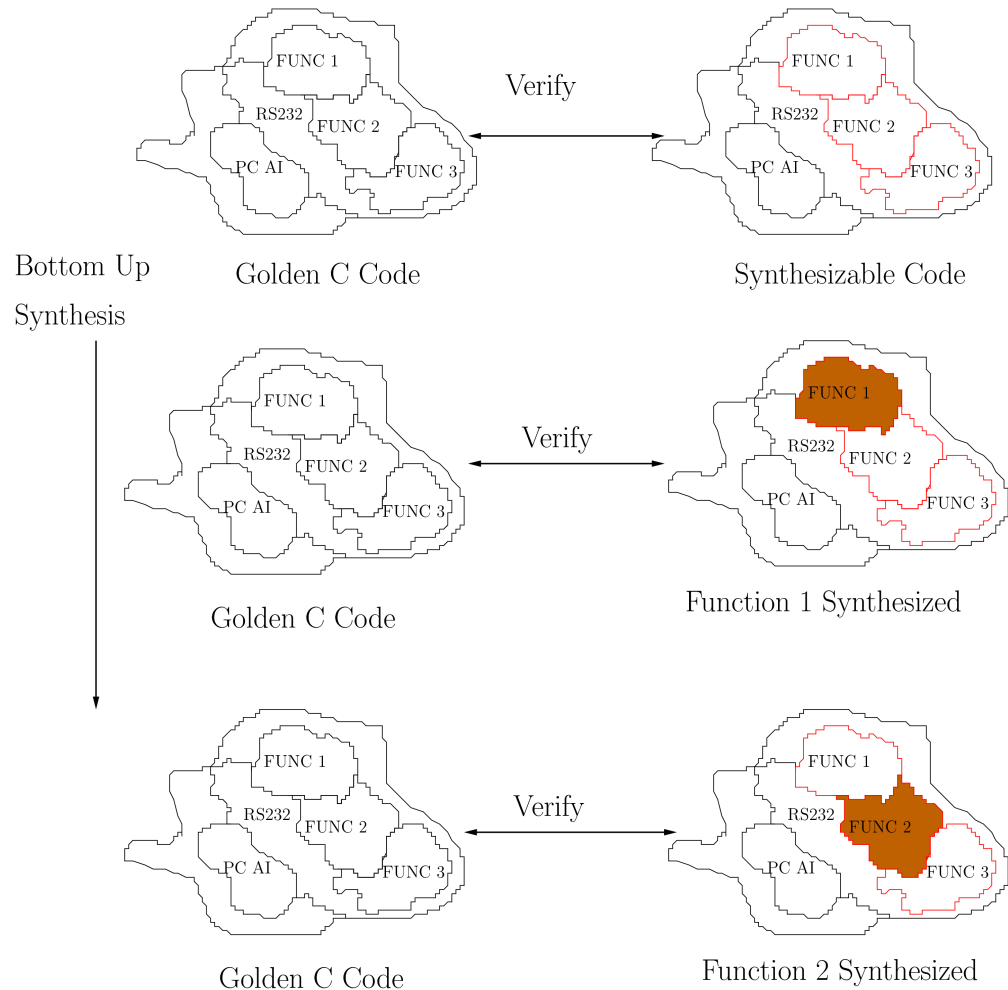
LoopNests



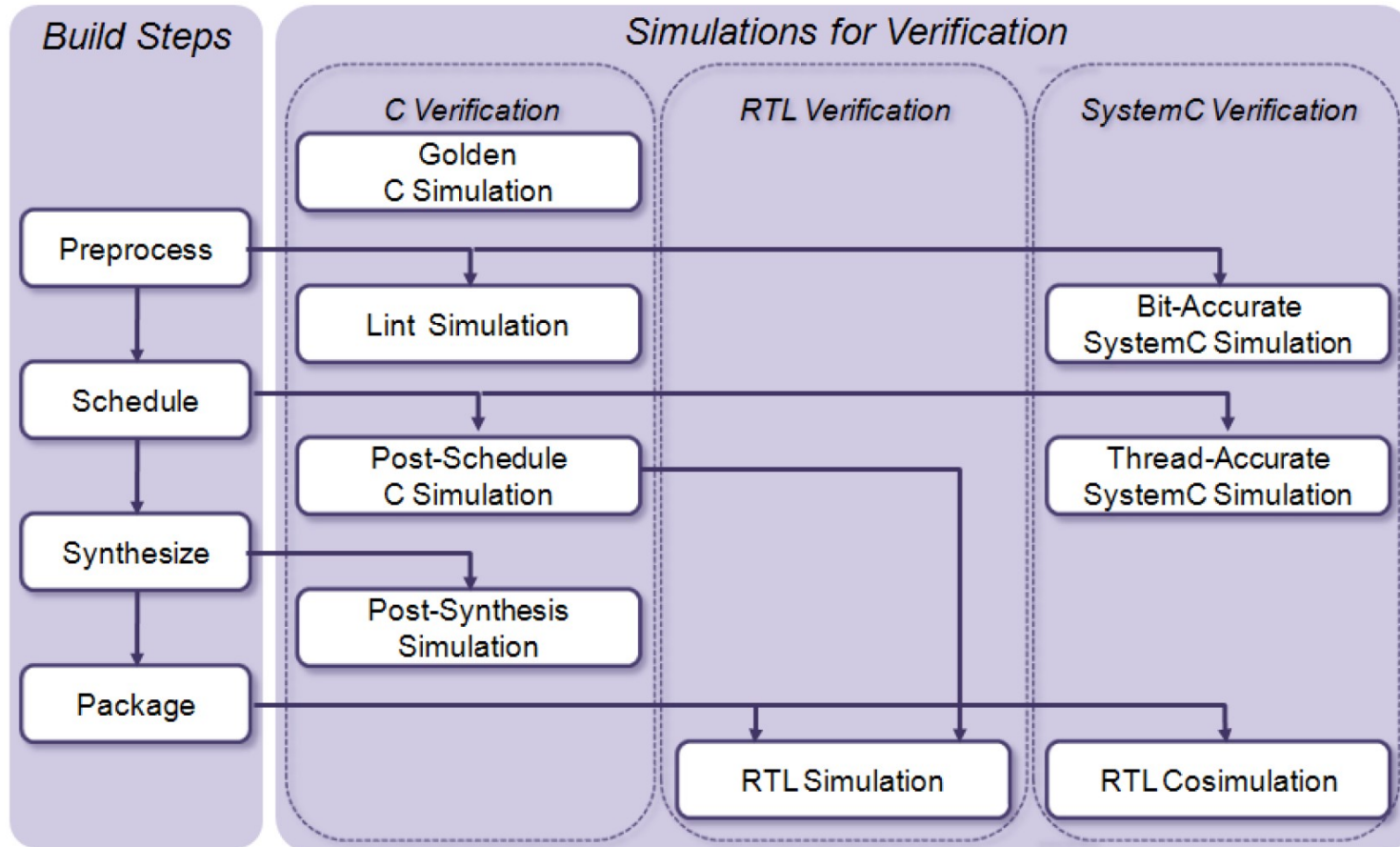
Hierarchical Synthesis

TCABs

- TCAB
 - Tightly Coupled Accelerator Blocks
- Can be imported into top level design as a block
- Considerably reduces the design time
- Facilitates Bottom-up verification (incremental).



SynphonyCC Build Steps

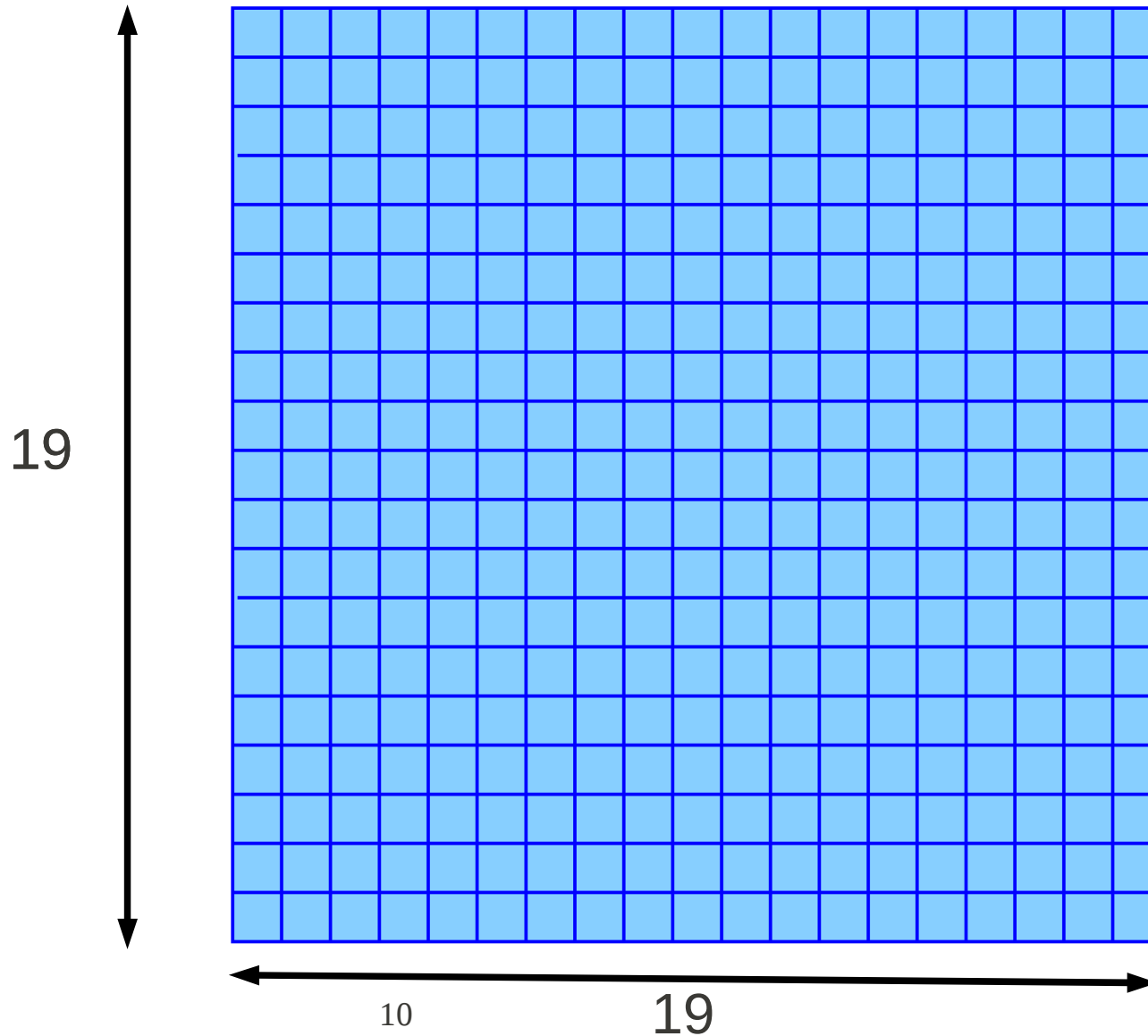


backend:
ise
quartus
synplify

sim:
modelsim
VCS

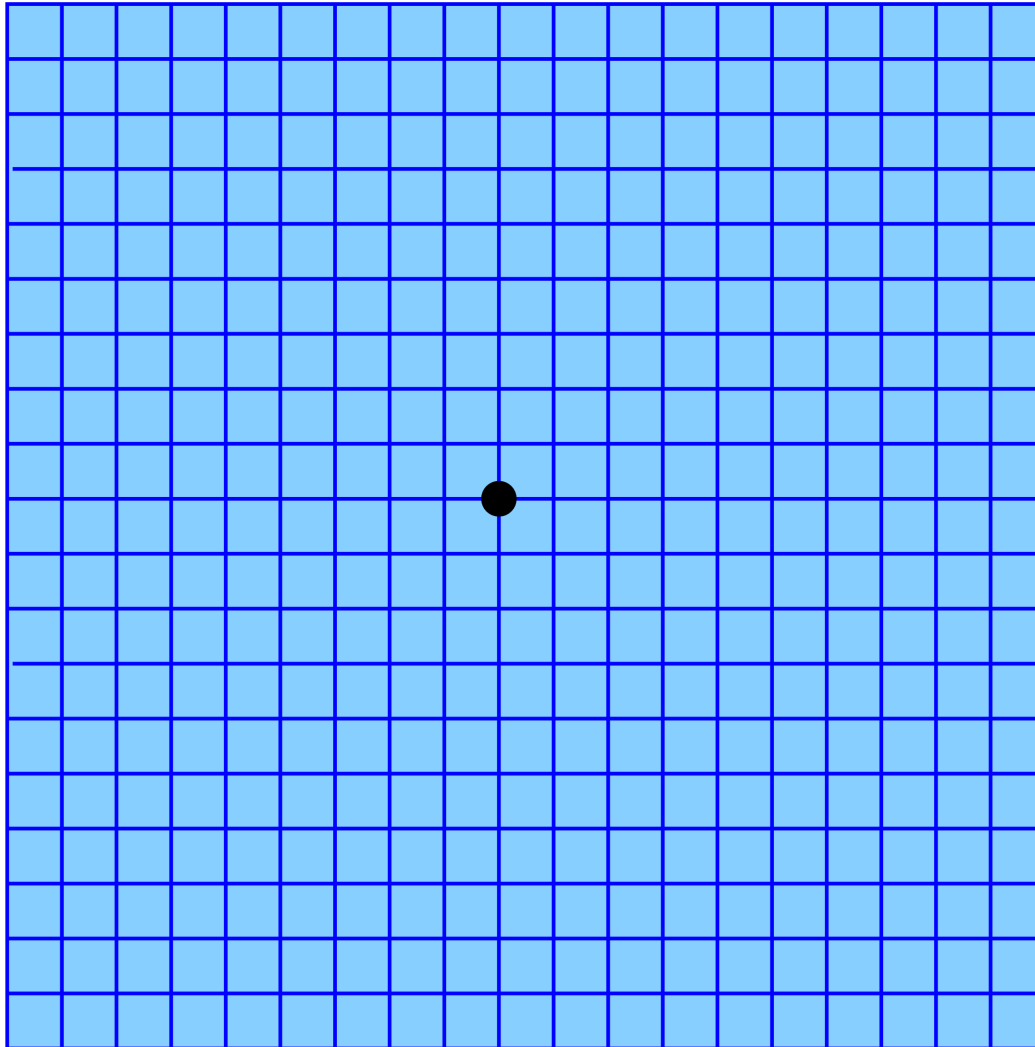
Figure 2. Build and Verification Steps. Taken from SCC User Guide (March 2012) p.84

Connect-6 The Game



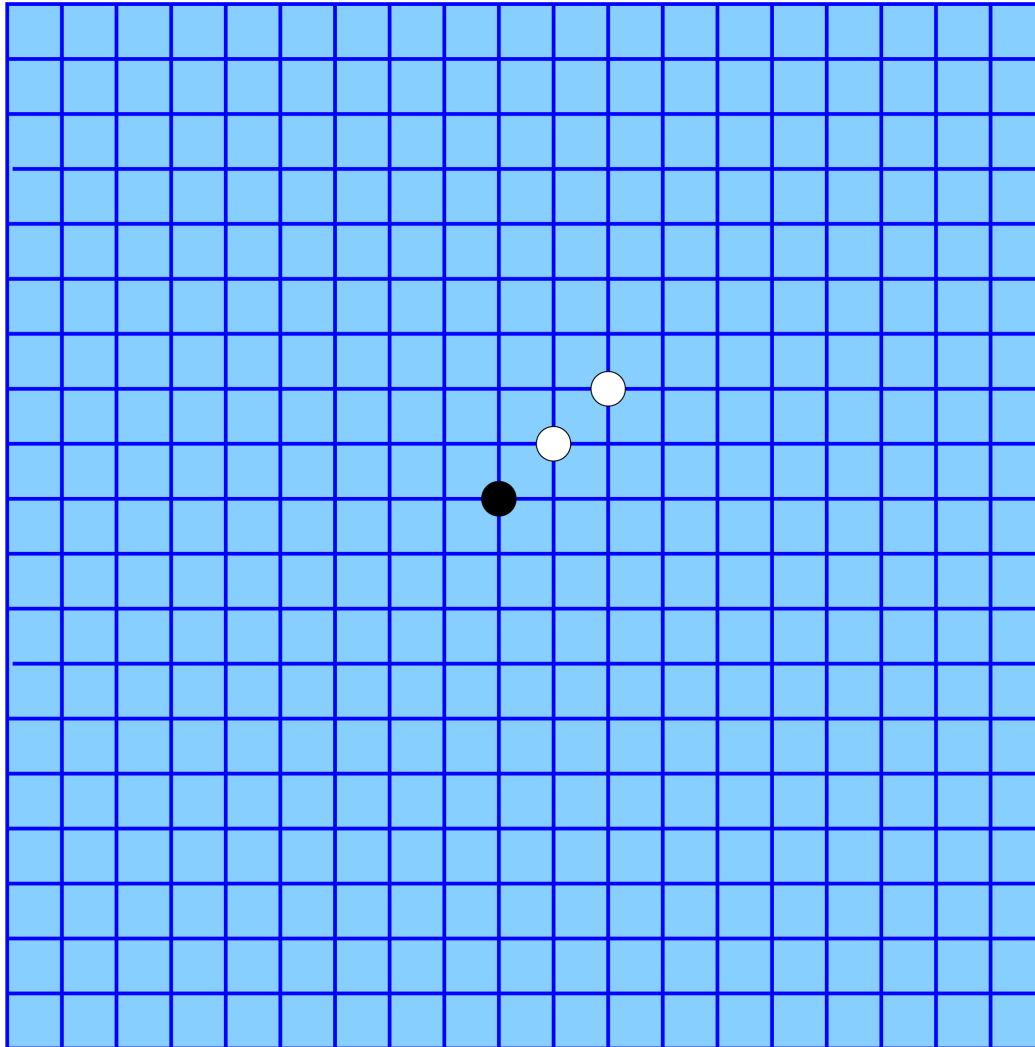
Played on a
19x19 GO Board

Connect-6 Example



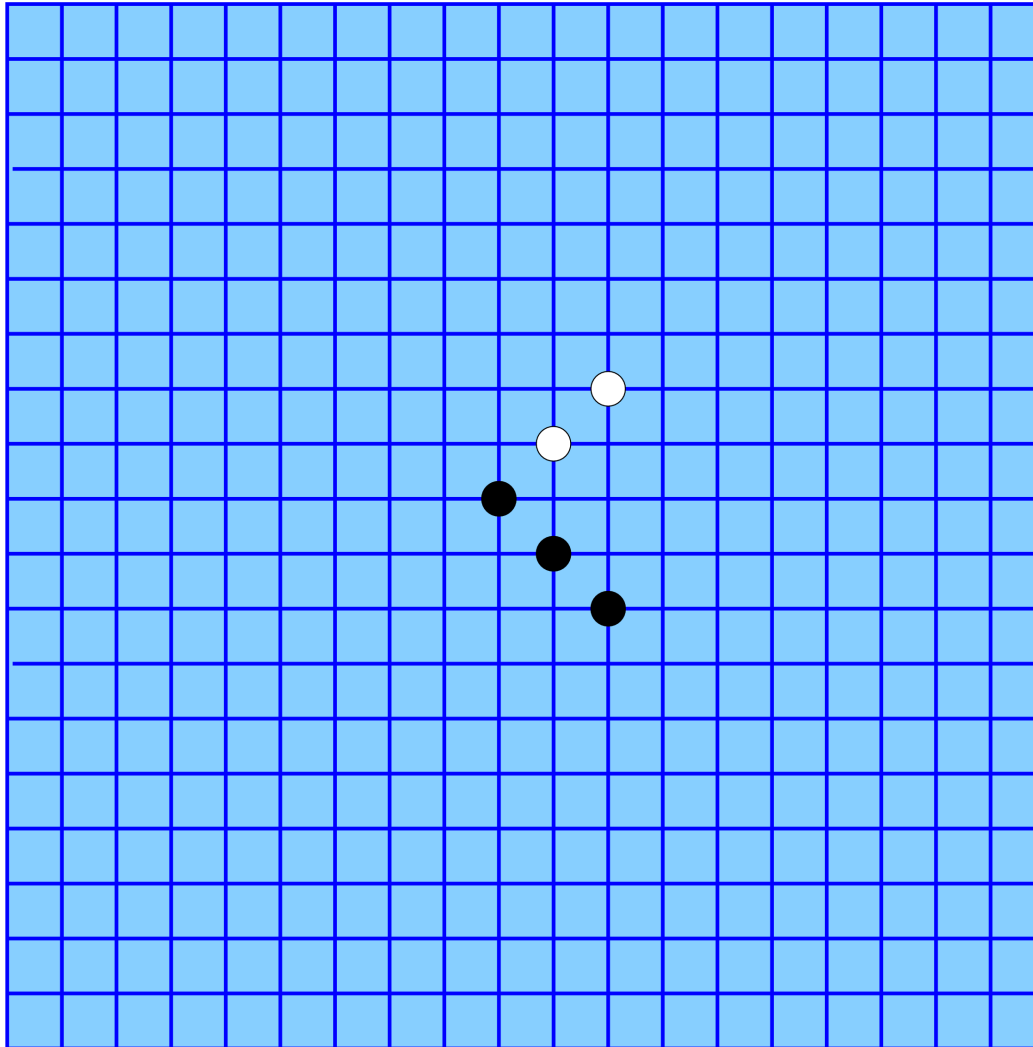
Black begins with
only one move

Connect-6 Example

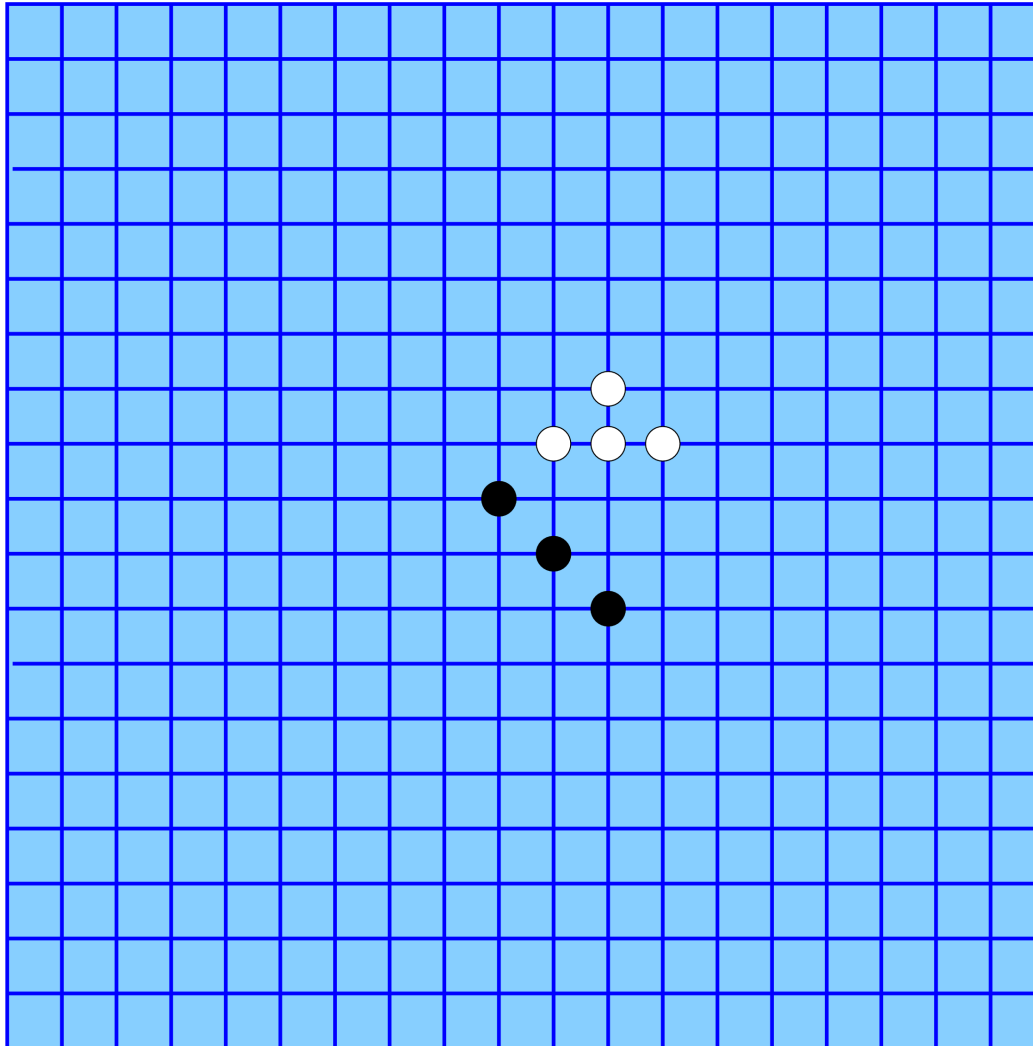


After that each player makes 2 moves per turn

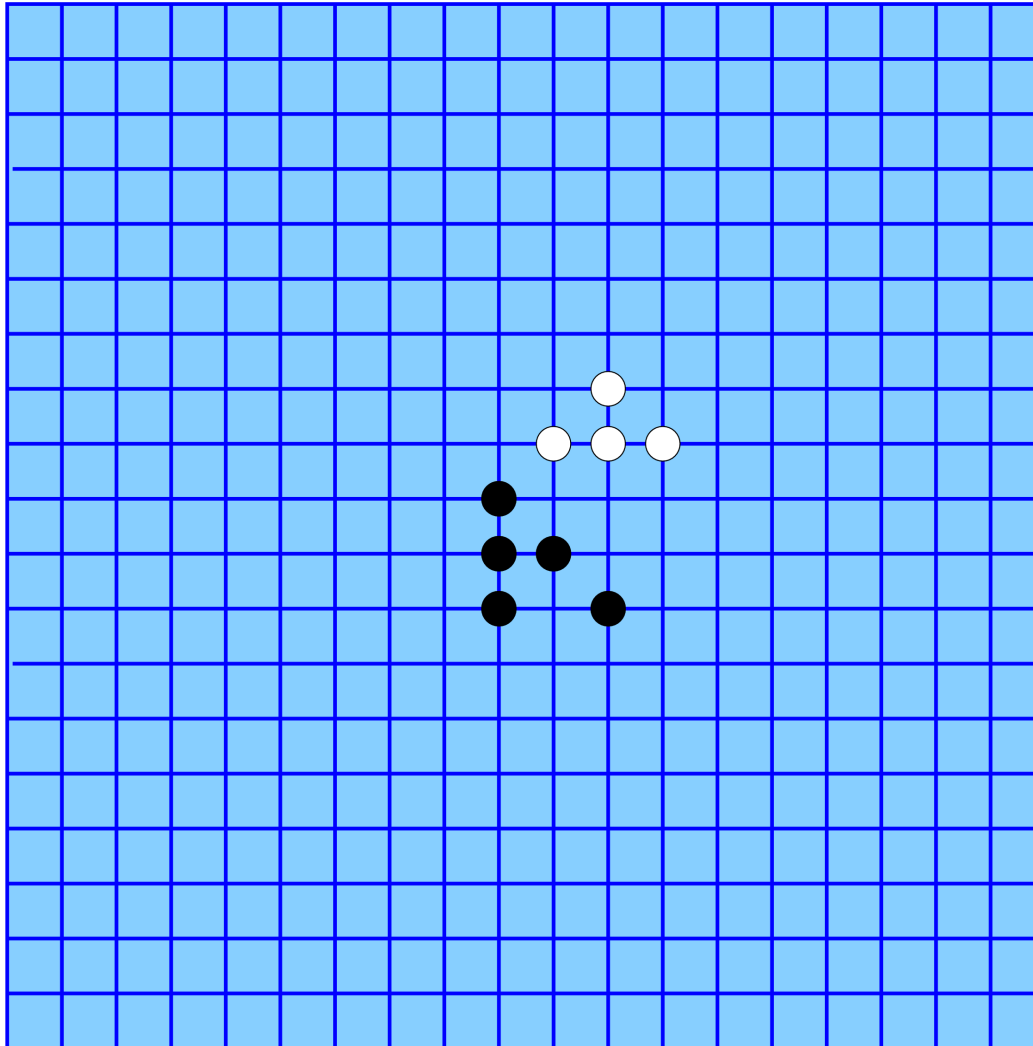
Connect-6 Example



Connect-6 Example

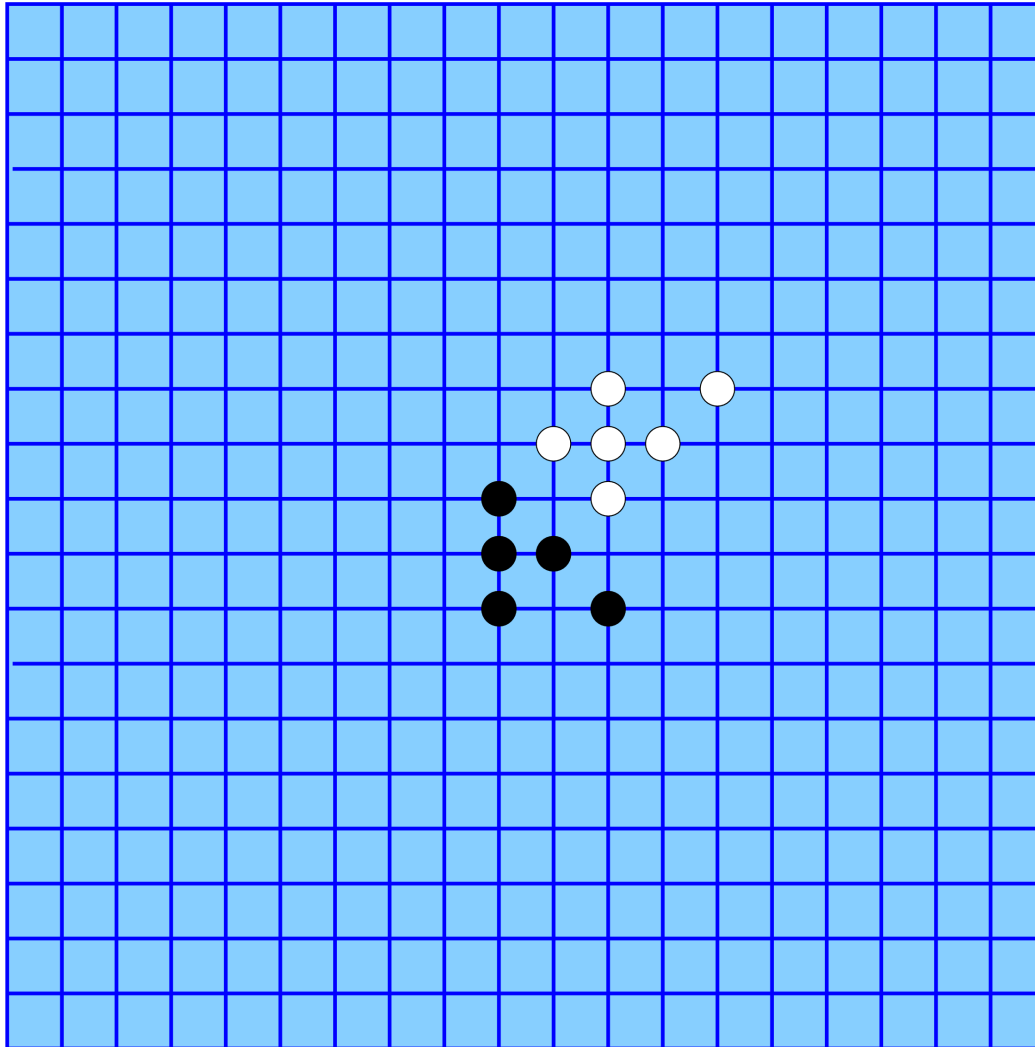


Connect-6 Example

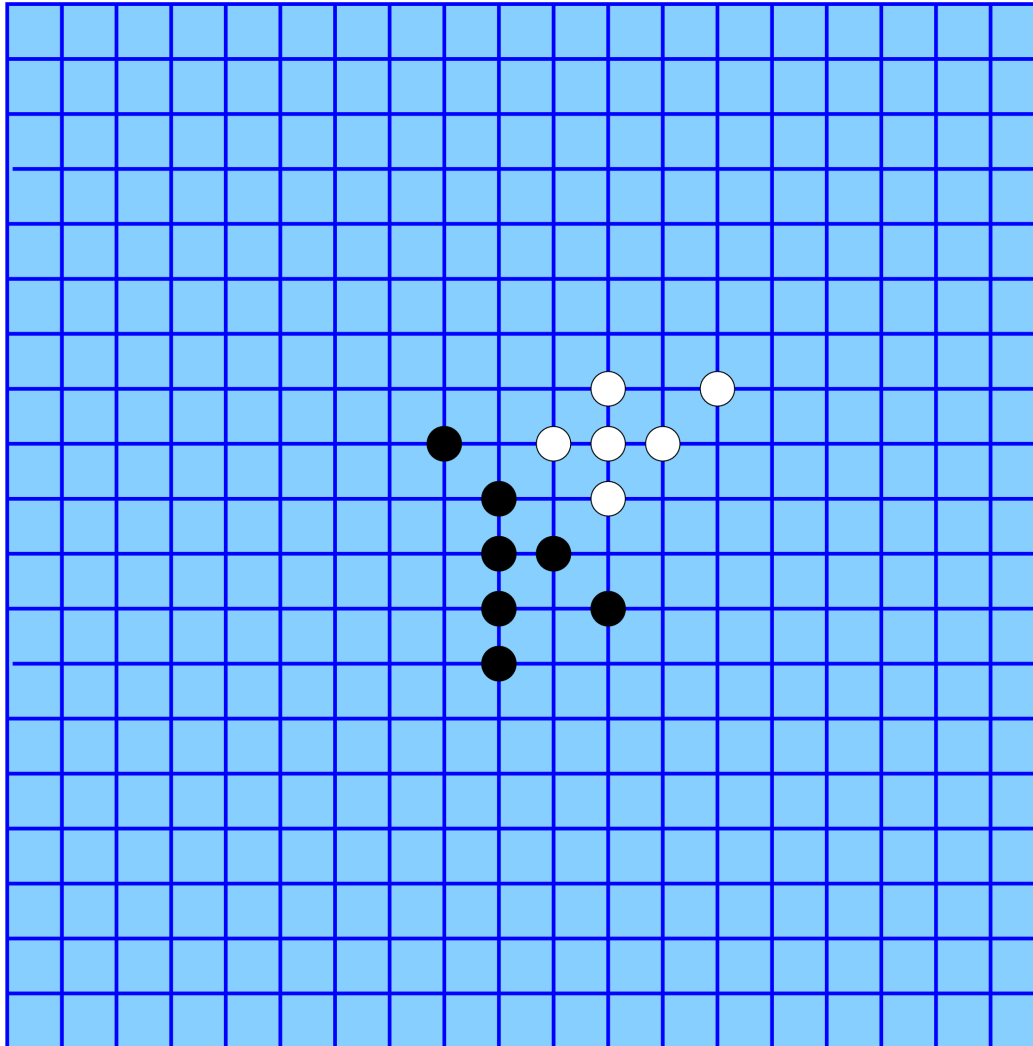


Connect-6

Example

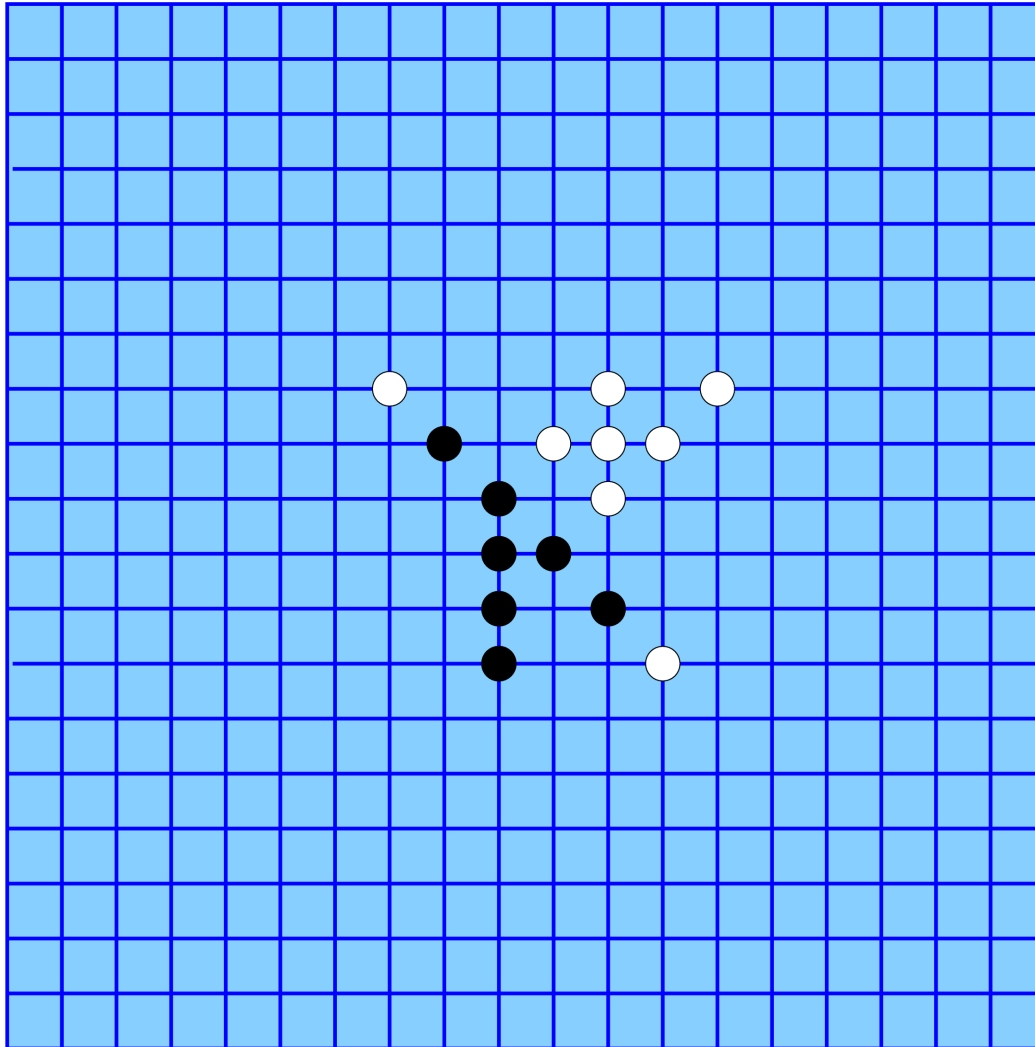


Connect-6 Example

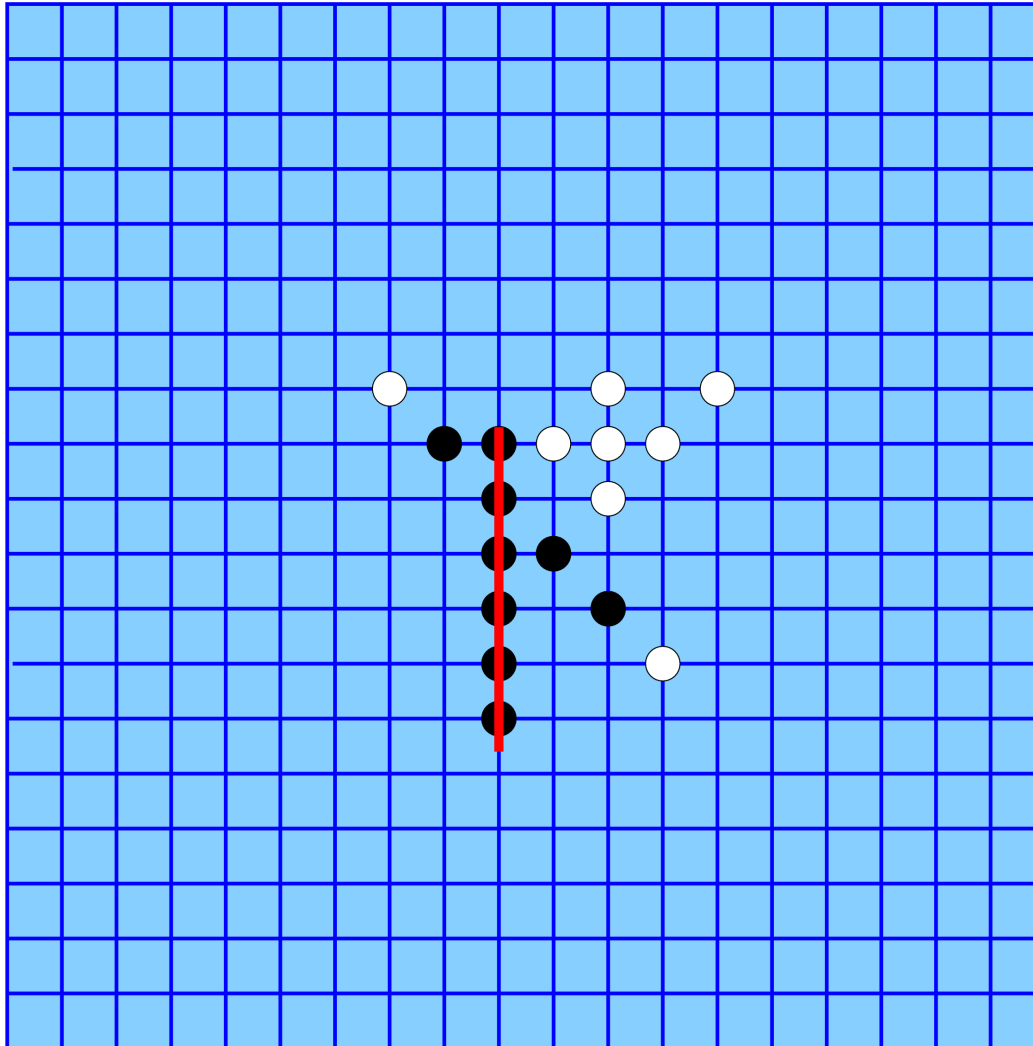


Black poses a
Double threat !!

Connect-6 Example



Connect-6 Example



Connect-6 Algorithm

- Source code available in <http://risujin.org/connectk/>
- Move Selection Core
 - Select possible moves with proper weights.
- Game-tree Search Minimax Algorithm with alpha-Beta pruning
 - Finds the best move by looking ahead

Connect-6 Algorithm

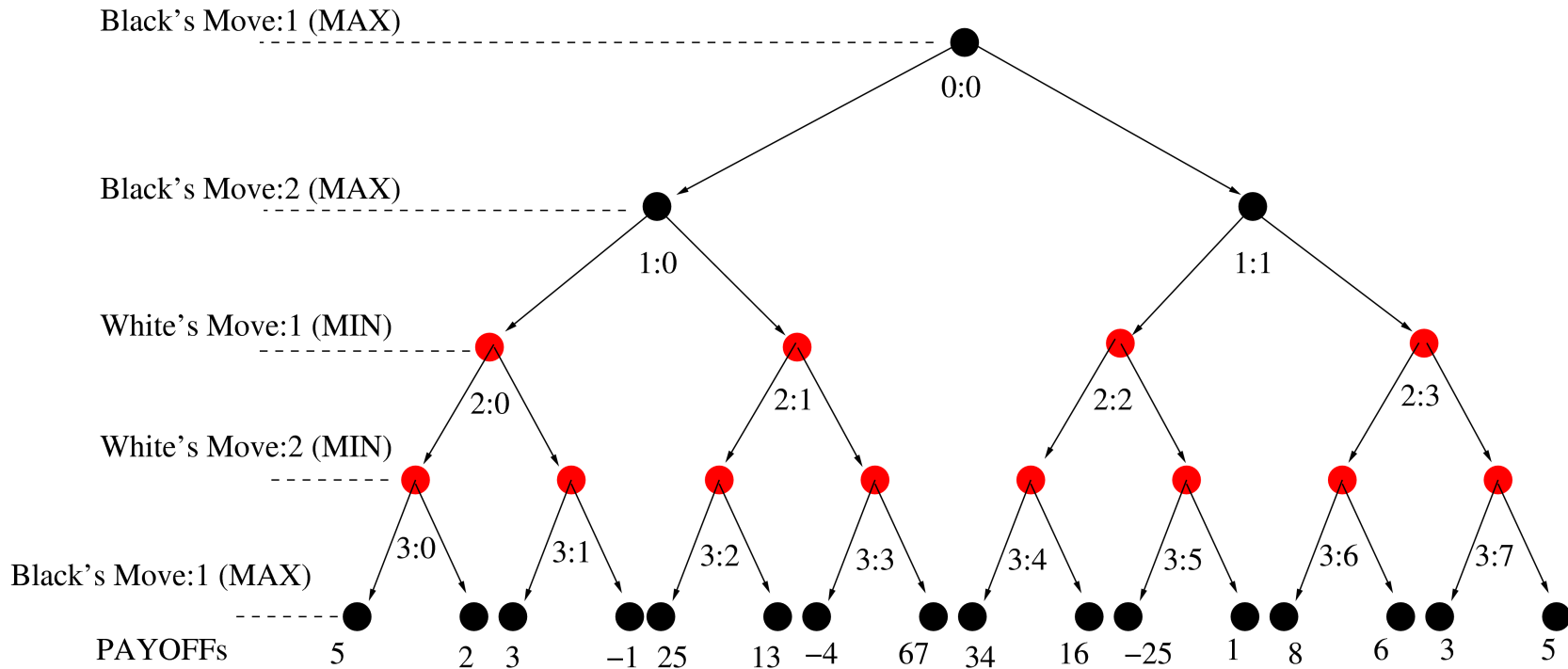


Figure 3. An example Game-Tree for Black with Depth=4 and Branching Factor=2.

Connect-6

Some Facts

- A Variation of K-in-a-row Games
 - e.g Tic-Tac-Toe, Go-Moku
- The first Move exception is added to increase fairness.
- Game-Tree Complexity (assuming average game length of 30)
 - Branching Factor $300C_2$ (Chess has a branching factor of 37)
 - Game tree complexity $(300C_2)^{30} = 10^{140}$
- A part of the Computer Olympiad
 - Tilburg'11: 6 Participants, Winner: Cloudict.connect6 from China

Design

Manual Preprocessing

- Including the functions called from C Libraries into the code.
- Changing Malloc() calls by statically assigning the memory. (fixed length array)
- Code sinking: putting as much code as possible inside for/while loops.

Design

Manual Preprocessing

```
/* Horizontal lines */
for (i = 0; i < board_size; i++)
    u_sum += threat_line(0, i, 1, 0,&b);

/* Vertical lines */
for (i = 0; i < board_size; i++)
    u_sum += threat_line(i, 0, 0, 1,&b);

-
-
-

/*SW diagonals */
for (i = 1; i < board_size-connect_k + 1; i++)
    u_sum += threat_line(board_size1,i, 1,1,&b);
```

```
for (i = loop_begin; i < loop_bound; i++){
    /*begin Canonical for loop*/
    switch(j){
        case 0:
            {
                arg1=0;arg2=i;arg3=1;arg4=0;break;
            }
        case 1:
            {
                arg1=i;arg2=0;arg3=0;arg4=1;break;
            }
        default:{
            Break;
        }
    }

    u_sum+=threat_line(arg1, arg2, arg3,arg4,&b);
}/*end Canonical for loop*/
```

Design Runtime Management

Method	Runtime
Hierarchical Synthesis (TCAB Based) Constraint : 50 MHz	37 m 33 sec
Flat Synthesis Constraint: 1 MHz	127 m 44 sec
Flat Synthesis Constraint: 50 MHz	Out of Memory

Table.1 Runtimes for Flat and TCAB based synthesis on a Intel Core i5 PC (3.33 GHz, 4GB RAM)

Design Execution Profile

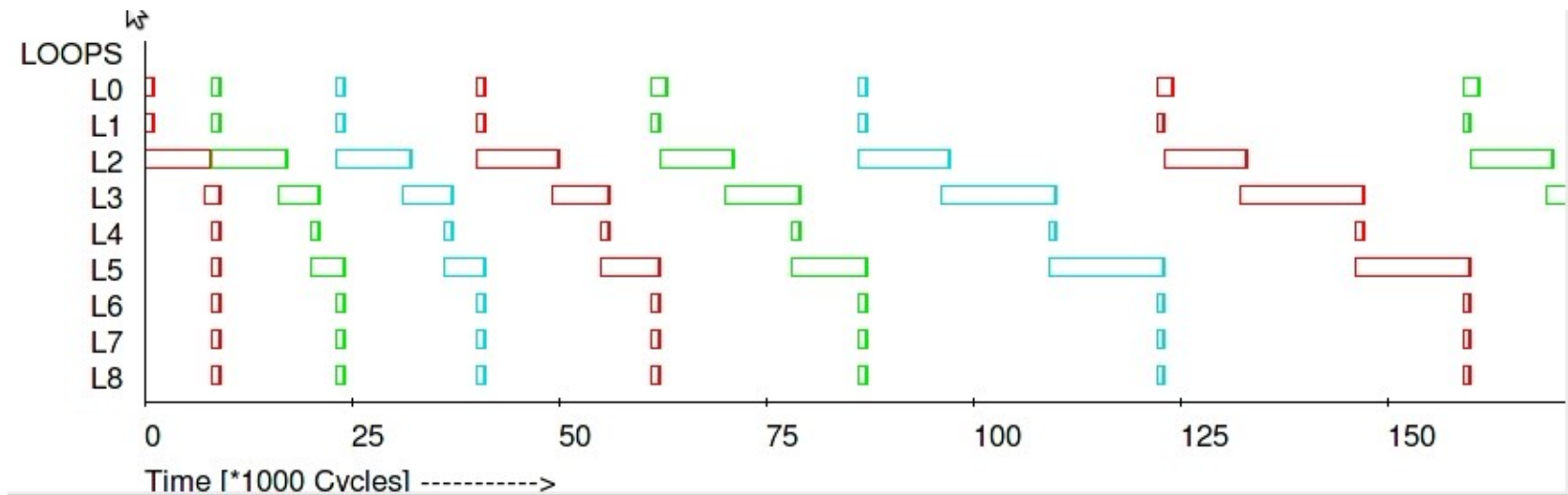


Figure 4. The sequential execution of un-optimized hardware, L2: Move scoring; L3: Preprocessing the Moves Array; L5: Sorting the Moves Array

Design Memory Porting

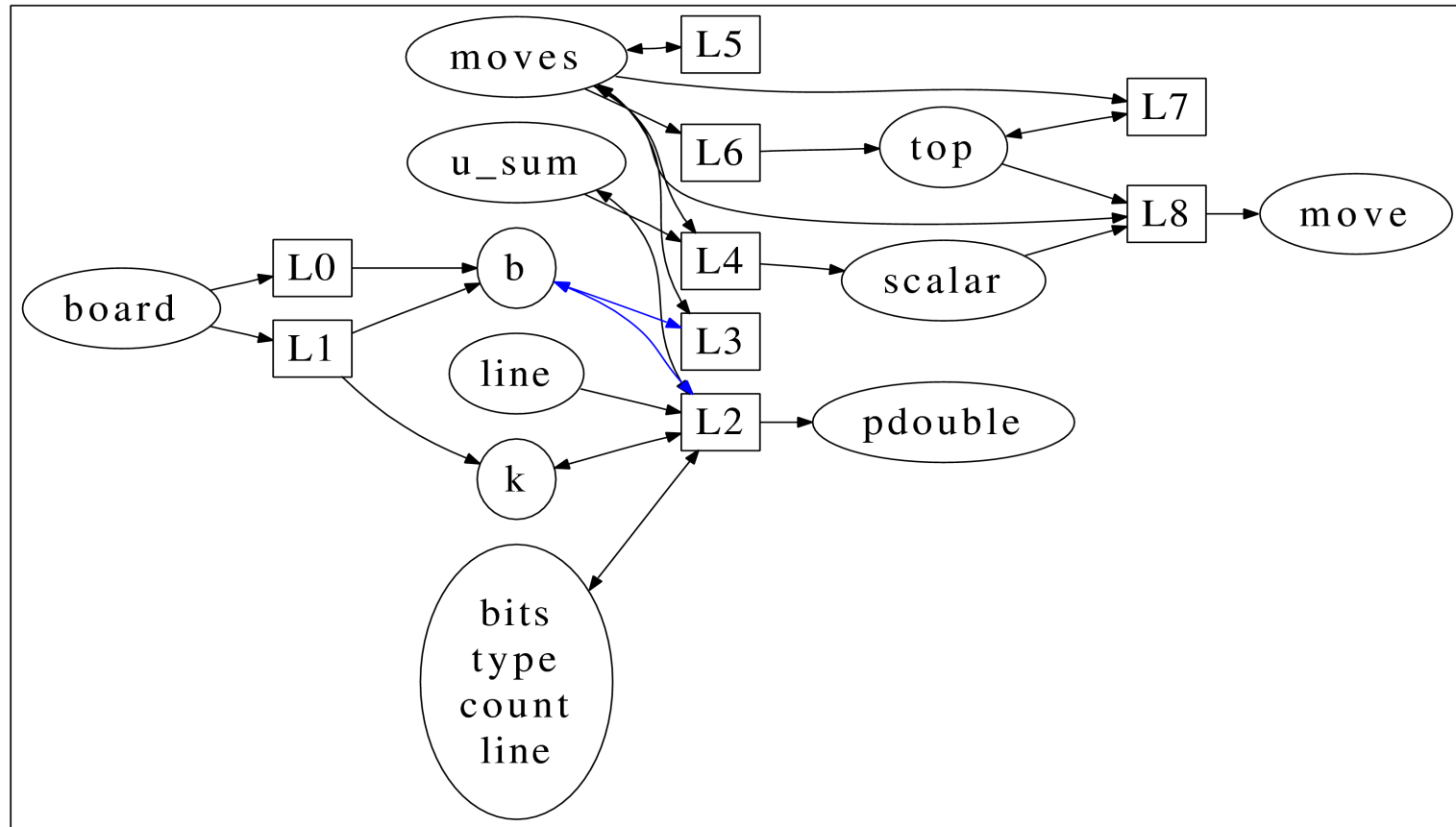


Figure 5. Board b has 2 read ports and 2 write ports, Can't Implement in Cyclone II.

Design

Memory Porting

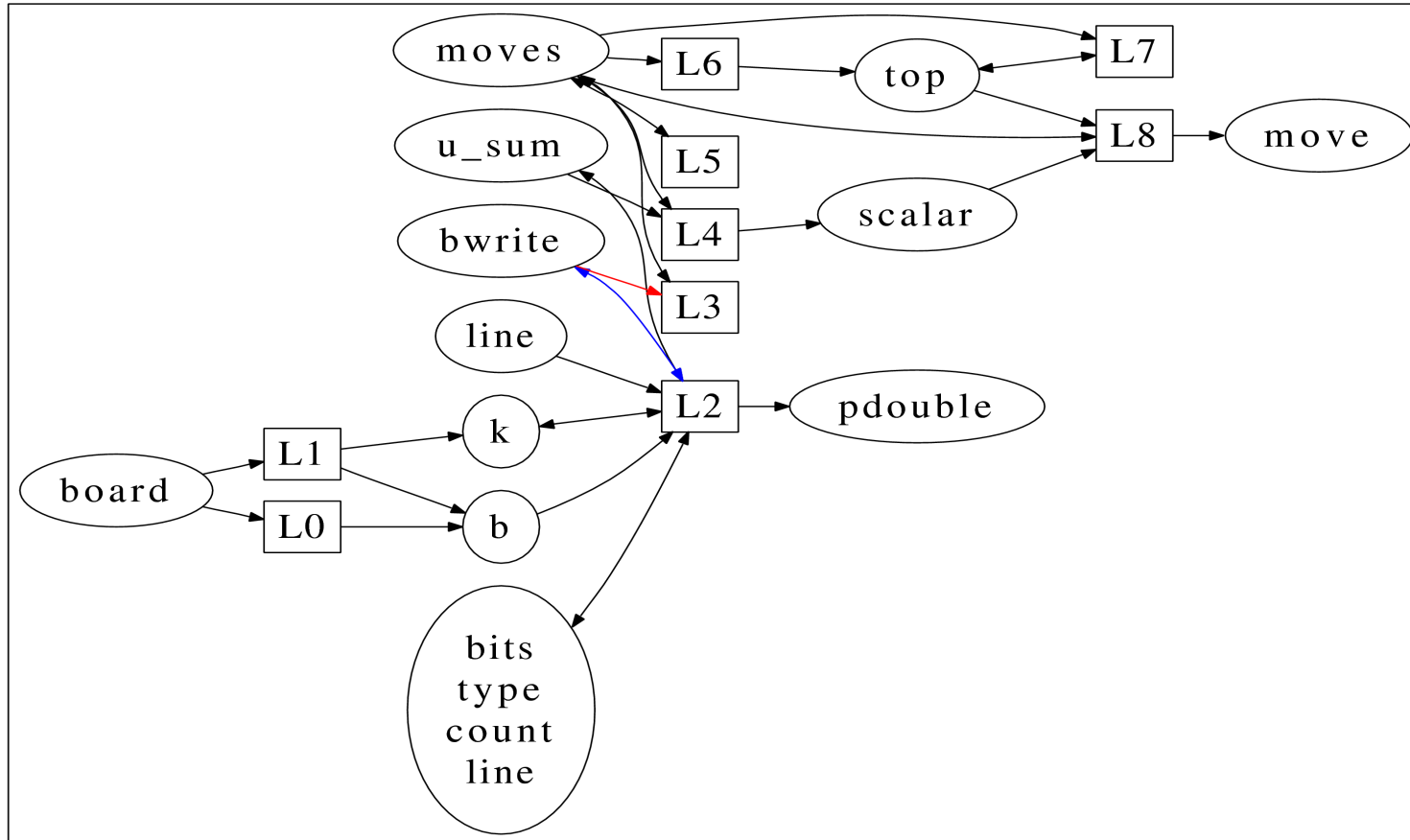
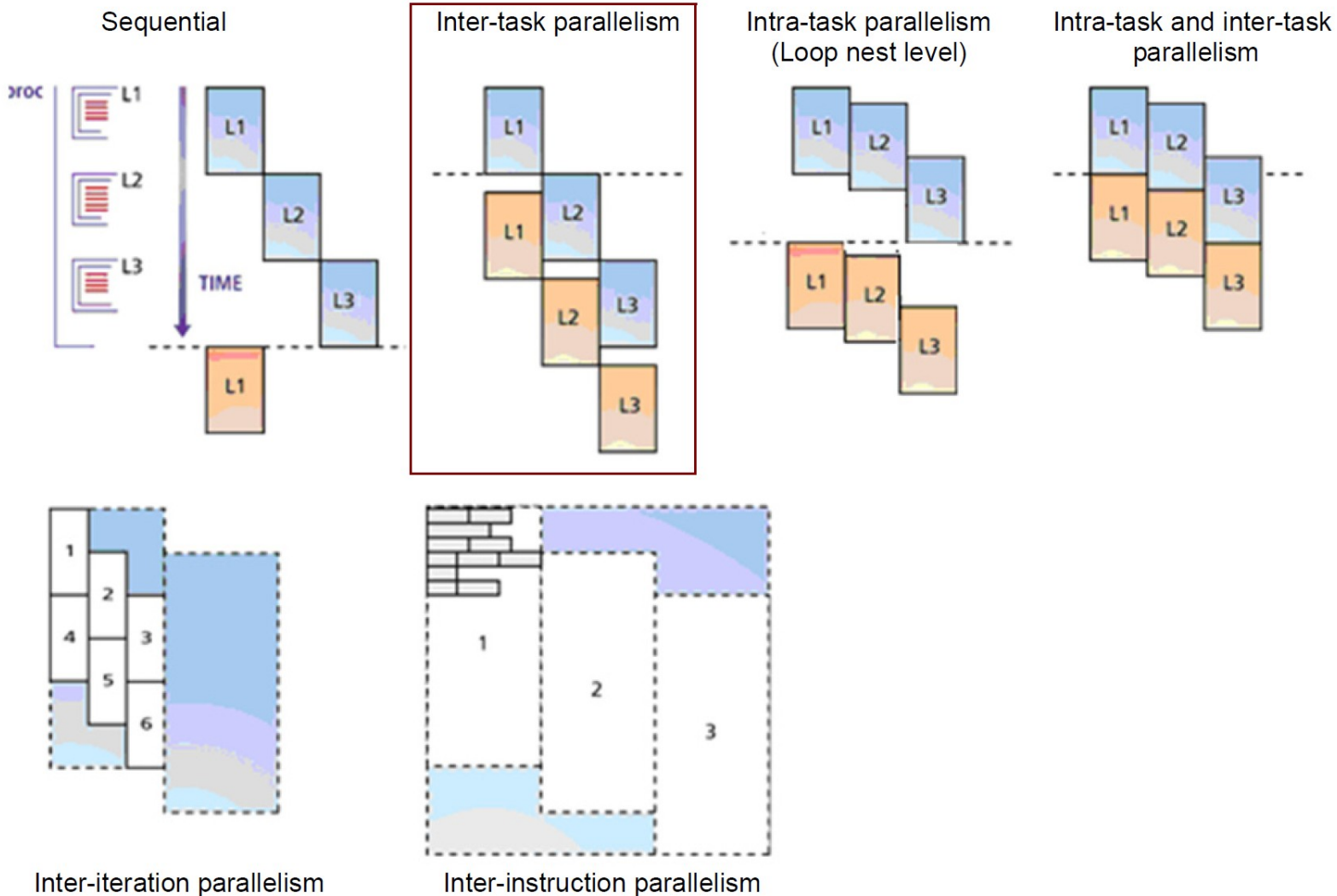


Figure 6. The board memory is split into b and bwrite to reduce porting.

Exposing Parallelism



Taken from SCC User Guide (Dec. 2011) p.267

Exposing Parallelism

Adding Streams

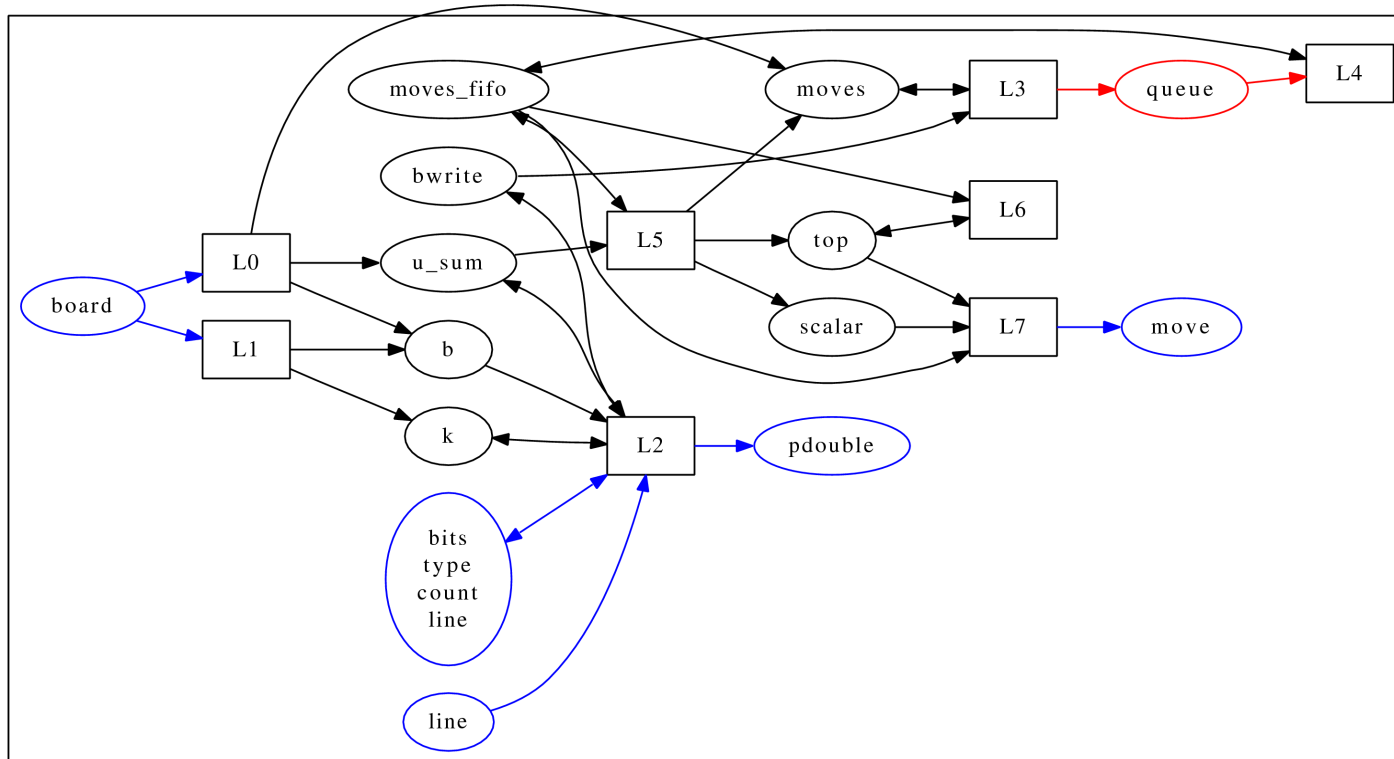


Figure 7. A stream (FIFO) is added between loopnests L3 and L5 so that the can operate in Parallel.

Exposing Parallelism

Adding Streams

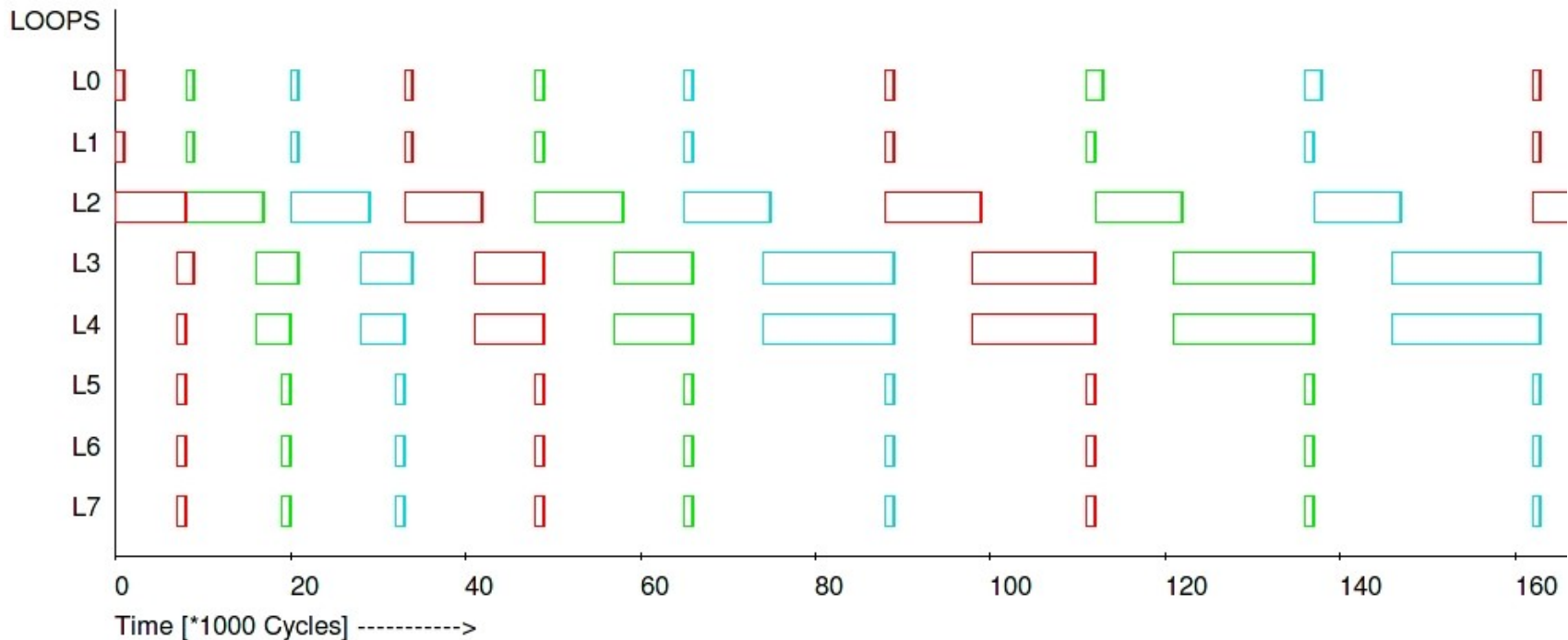


Figure 8. Inter-Loop Parallelism: L3 and L4 working in parallel. L3: Preprocessing the moves stream; L4 :sorting the moves stream (insertion sort)

Exposing Parallelism

Task Overlap

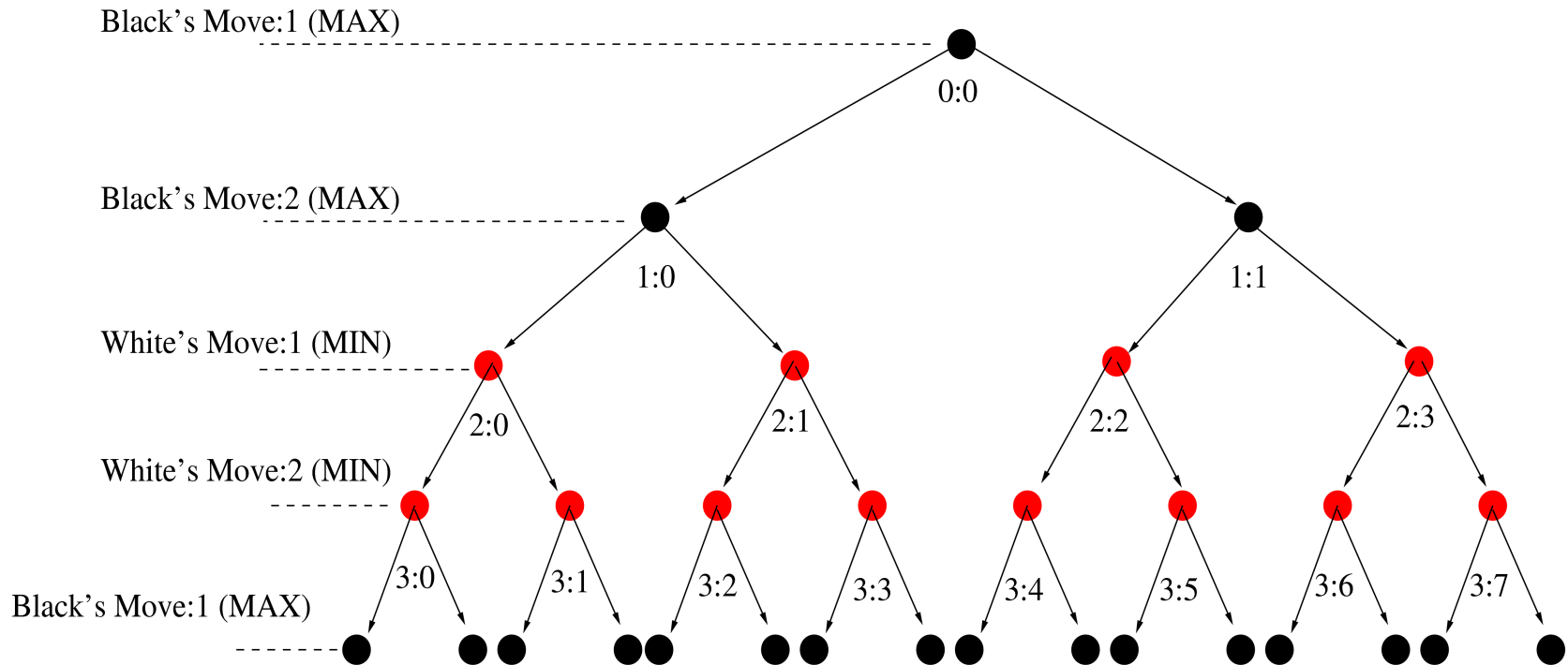


Figure 9. Inter-Task Parallelism: Move Selection on Boards 3:0 and 3:1 can be overlapped because they are stocked in two different arrays.

Exposing Parallelism

Task Overlap

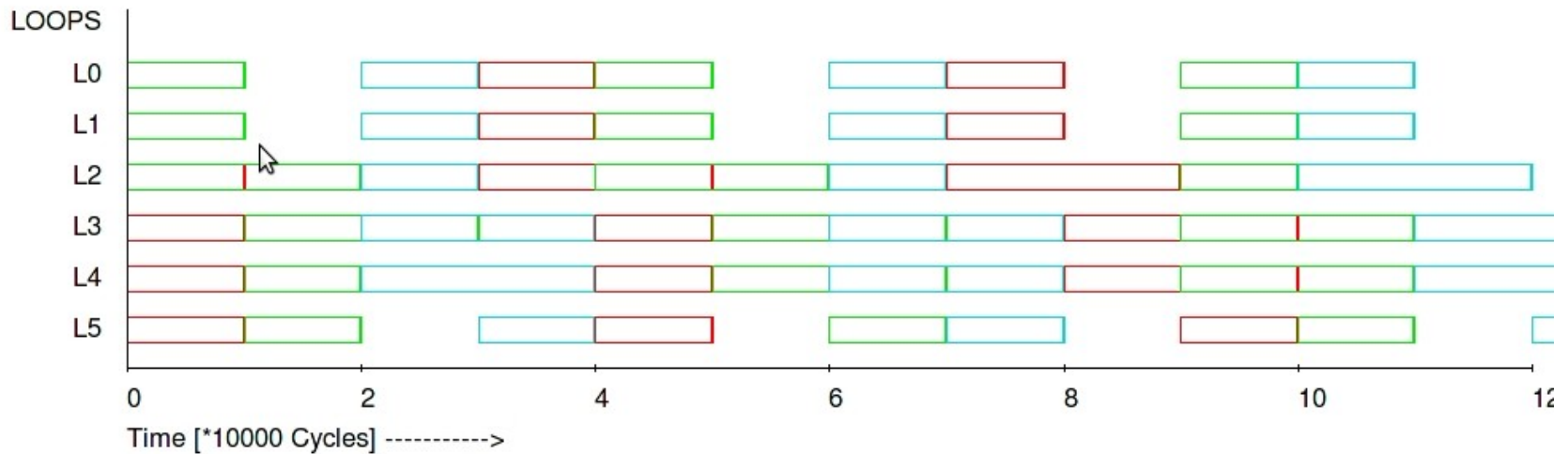


Figure 10. Inter-Task Parallelism: While L3 L4 hardware block operates on board 1:0, L1 hardware block starts operating on 1:1.

- Simulation Based
- Bottom up Synthesis with Non-Regression
- SystemC and Verilog testbenches are automatically generated from C. :-)

- Code Coverage for C simulation 99 % with a testbench where the AI makes 73 Moves
- Possibility to determine Code-Coverage for RTL as well
- Final design is verified via FPGA Emulation with 10000 random games for
 - The AI makes no invalid moves
 - The behaviour is exactly same as that of the software given the same random seed.

Results

Implementation	Avg. Cycles	Avg. Time to Make one Move
Intel i5 (3.33 Ghz, 4GB RAM)		0.56 ms
FPGA (Cyclone II, 38 MHz)	47984	1.26 ms
FPGA (Cyclone II 38 MHz) optimized	28982	0.76 ms

Results

Implementation	BRAM	DSP	FF	LUT
Game Tree (depth=1,branch=1)	24	14	9508	22530
Game Tree (depth=5,branch=2)	193	73	16536	42919

Results

Slack Histogram (Overall)

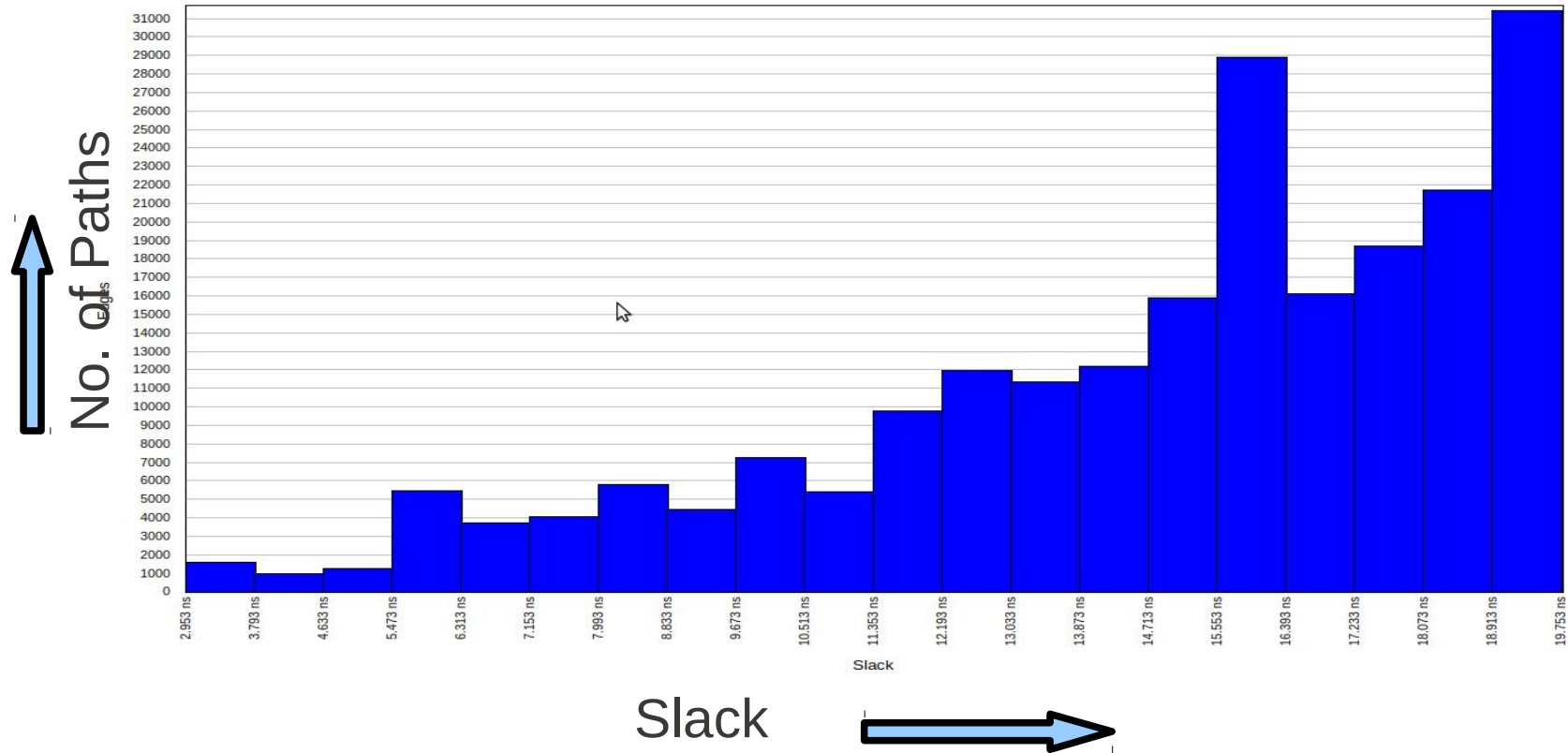


Figure 10. Slack Histogram for the complete design. Generated from Quartus tool (Altera) before P&R.

Results

Slack Histogram (TCAB)

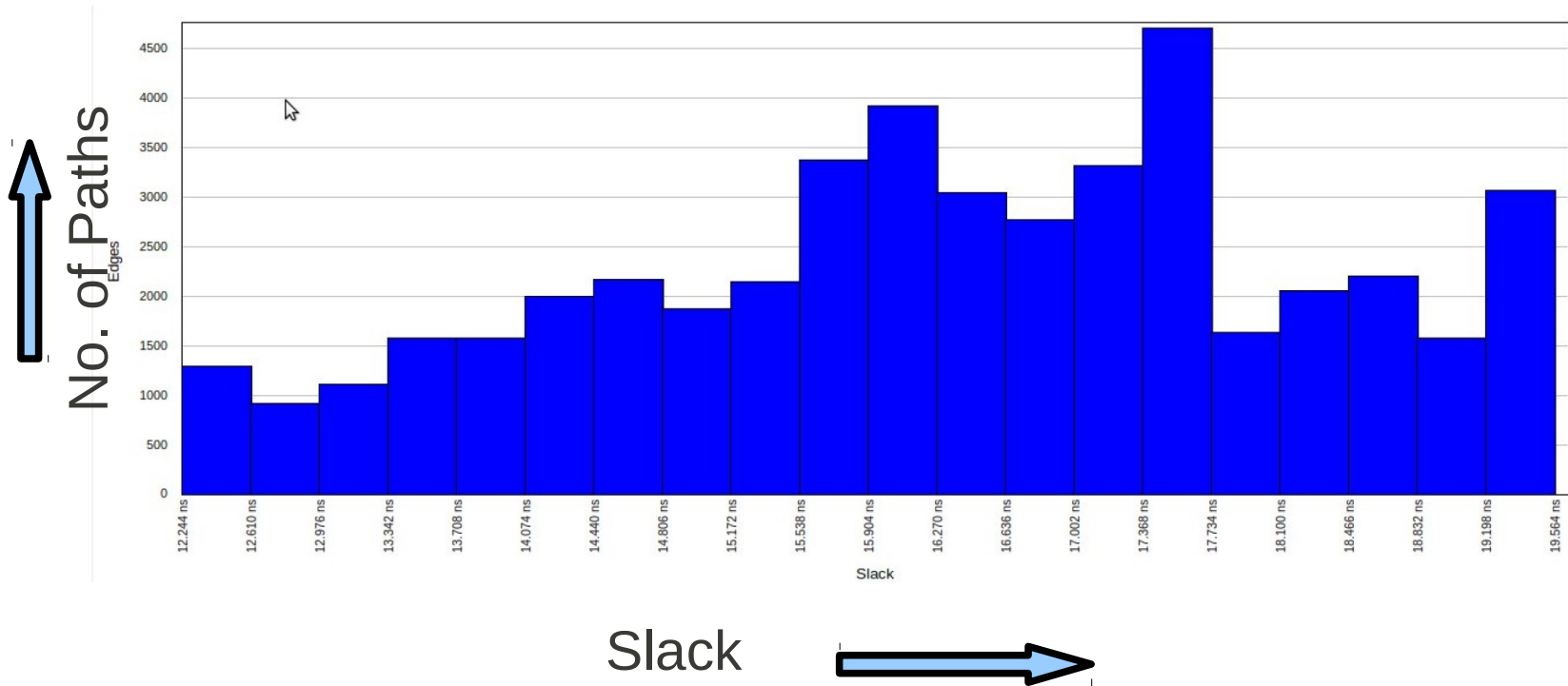


Figure 11. Slack Histogram for the TCAB Threat_window. Generated from Quartus tool (Altera), before P&R.

Conclusion

- **ease of use**
 - Push Button
 - Semi-Automatic ✓
 - Manual Digging
- **Highly integrated environment (C, C++, systemC)**
- **Good management of simulation based verification flow**

Limitations:

- **Runtime**
- **Extremely Memory Hungry**

Good Performance can be achieved by simply specifying pragmas, if the code is explicitly parallel.

A good strategy will be to synthesize the whole design, then recode the critical parts in systemc.

Possible Improvements:

- **Addition of multiple clock domain TCABs can be useful.**
- **Integration of VHDL/Verilog IPs as TCABs.**
- **Industry standard BUS specific wrappers can be useful.** (The wrapper and interfacing signals are predefined, need some tweaking to integrate into standard BUS architectures.)
- **More integrated formal verification can be useful.** (the user need to add the assertions in to the existing C code to formally capture the behaviour of C code.)



Synopsys Users Group
UK 2012



Synopsys Users Group
UK 2012

DEMO

SPARTAN Vs. CYCLONE

- **The processor speed is profiled with gprof, the actual walltime can be different dependent on the processor load.**
- **The Frequency of operation for FPGA is the worst-case frequency over a wide range, estimated by Quartus Tool.**
- **The software implementation uses a Depth-First Search with Alpha-Beta pruning, In hardware we used a statically scheduled breadth-first search**