

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Architecture without explicit locks for logic simulation on SIMD machines

M. Chimeh P. Cockshott

Department of Computer Science
University of Glasgow

UKMAC, 2016

Contents

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

1 Importance Of Simulation

2 Simulation Algorithms

3 Circuit Representation

4 SIMD Simulation

5 Machines

6 Results

- Setup
- Parallelism
- Comparisons
- Compilers

The Importance Of Simulation

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Using models to replicate the behaviour of an actual system is called **simulation**. A **model** is a simpler and abstract version of a desired system. In general, simulation refers to time evolution of a computerized version of a model.

Due to the growth of design size and complexity, design verification is an important aspect of the Integrated Circuit (IC) development process. The purpose of verification is to validate that the design meets the system requirements and specification. This is done by either functional or formal verification.

The most popular approach to functional verification is the use of simulation based techniques.

Cycle based vs Event Based simulation

Cycle based

Evaluates all logic gates during every simulation cycle

- Handles synchronous designs
- Suitable for circuits with high activity rate
- Performs unnecessary simulations (extra computation)

Event based

Evaluates only logic gates with a change on their inputs

- Handles both synchronous and asynchronous designs
- Suitable for circuits with low activity rate
- Requires a centralized scheduler that may cause large amount of overhead
- Maintaining queue for the list of events is challenging

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Cycle based simulation algorithm can be used to accelerate the simulation of synchronous design that is composed of combinational blocks and latches.

Cycle Based Algorithm

```
initialize each flop flop to zero
while there is more input
  read inputs
  for pd = 0 to critical path depth
    simulate each logic function at depth = pd
  update flip flops
```

Levelisation

- Step 1. form set of all signals feeding the latches or outputs.
- Step 2. push gates whose outputs generate this set onto a stack
- Step 3. form set of all signals feeding the set of gates on the top of the stack
- Step 4. if this set is empty goto step 5 otherwise goto step 2
- Step 5. set $n=0$
- Step 6. pop the stack and label all gates with level n
- Step 7. if stack empty terminate, otherwise set $n=n+1$ and goto step 6

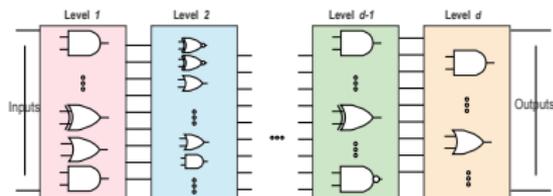


Figure: Levelisation example in a circuit, each of the coloured blocks

Circuit Representation

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

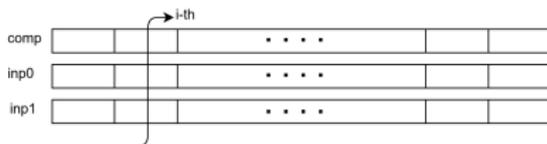


Figure: Vectors to hold the circuit specification

The `comp` array hold the type of logic gate. The `inp0` and `inp1` arrays points to a location in state array that signal values are stored.



Figure: Signal state vector

The state array contains all the signal values. Output signals of logic gates at the same level are stored adjacent to each other.

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

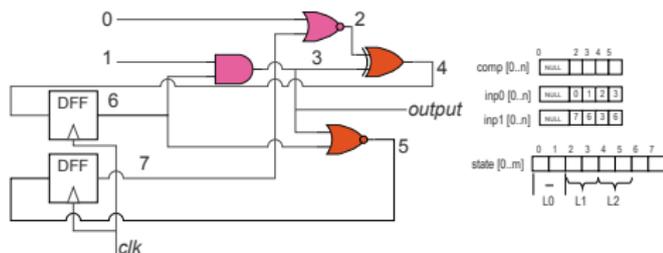


Figure: An example of a circuit with label

Logic gates of the same level are shown in the same color.

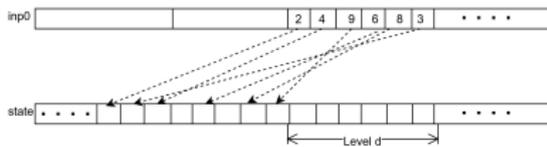


Figure: Illustration of input value retrieval from the state array

SIMD Simulation Requirement

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

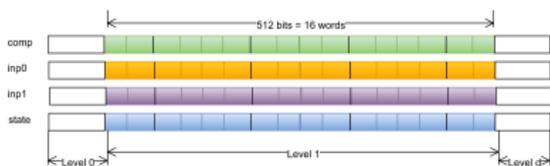


Figure: Example of performing SIMD operation on 512-bits of data in the integer array

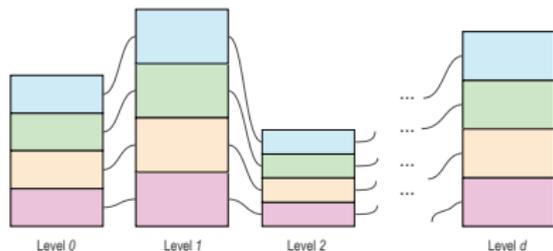


Figure: An example of workload among the threads per level simulation. The curved lines in the figure symbolized the synchronization between threads.

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

- Lookup Table vs Direct Logic
- Bit Packing vs Word Packing

Bit Packing vs Word Packing

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

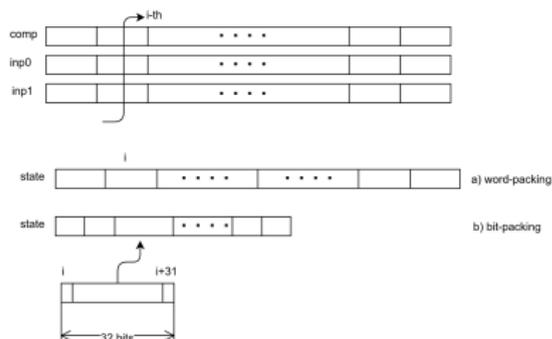


Figure: Signal Representation using a)word packing b)wbit packing

The state vector can either store each signal as 1 bit or use a whole word for each signal. The `inp0`, `inp1` vectors are unaffected by this choice, but the `comp` vector can be discarded when using bit packing.

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

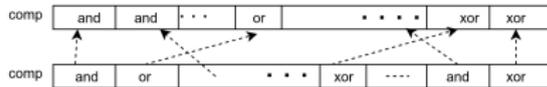


Figure: Re-arrangement of logic gates in a circuit in Bit packing Technique

This illustrates the re-arranged logic gates in `comp` array. Logic gates of the same type are stored next to each other. The rest of arrays are organized accordingly. The top is a re-arranged, and the bottom array is a normal array. This allows CPU AND, OR, NOT instructions to be used 32 bits at a time.

Xeon Phi

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Parameter	Intel Xeon Phi	Intel Xeon
	Coprocessor 5110P	Processor E5-2620
Core, Threads	60, 240	6, 12
Clock Speed	1.053 GHz	2 GHz
Memory Capacity	8 GB	16 GB per socket
Memory Speed	2.75 GHz (5.5 GT/s)	667 MHz (1333 MT/s)
Memory Channels	16	4 per socket
Memory Data Width	32 bits	64 bits
Peak Memory Bandwidth	320 GB/s	42.6 GB/s per socket
Vector Length	512 Bits (Intel IMCI)	256 Bits (Intel AVX)
Data Caches	32 KB L1, 512 KB L2 per core	32 KB L1, 256 KB per core, 15 MB L3 per socket

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Results

Experimental Setup

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

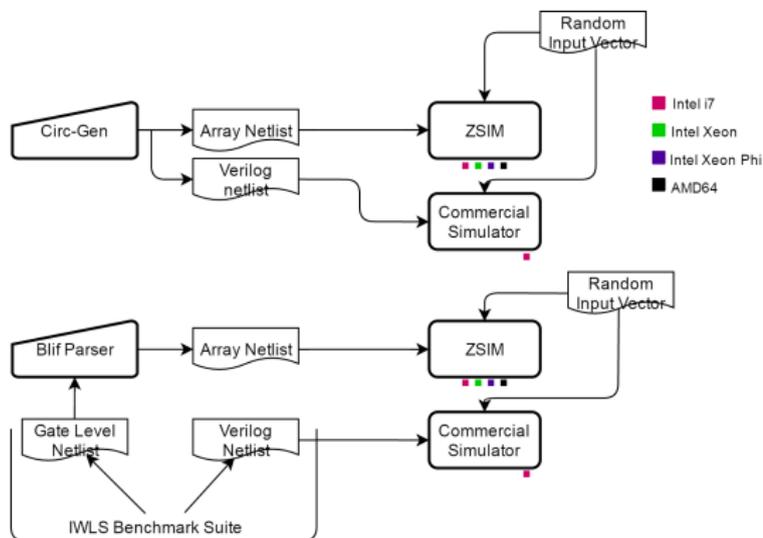
Machines

Results

Setup

Parallelism
Comparisons
Compilers

Summary



Note that our SIMD algorithm was implemented in both Pascal and C++. ZSIM was compiled with three different compilers (Intel C, Gcc, Vector Pascal)

Vectorization and Multicore Performance

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup

Parallelism

Comparisons

Compilers

Summary

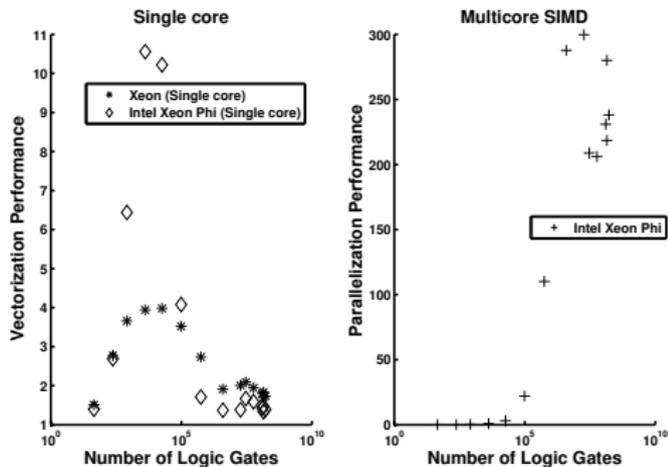


Figure: Performance comparison of single and multicore SIMD with single core sequential code on Intel Xeon Phi and Xeon. Left plot shows the speed on both machines using single core. Acceleration gain falls off for larger circuits that do not fit in 1 core's cache. Right plot shows the speedup when 240 threads SIMD were used on Intel Xeon Phi.

Performance Comparison to Xilinx Commercial Simulator

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup

Parallelism

Comparisons

Compilers

Summary

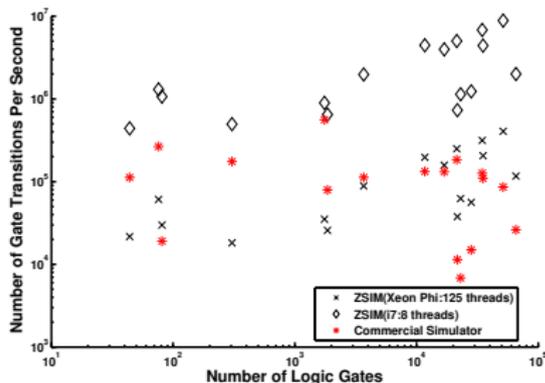


Figure: Log/Log plot of gate transitions per second for the Xilinx Simulator ISIM (on Intel i7), and the SIMD ZSIM running on both Intel i7 and Xeon Phi for circuits from IWLS suite

Performance Comparison to Xilinx Commercial Simulator

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup

Parallelism

Comparisons

Compilers

Summary

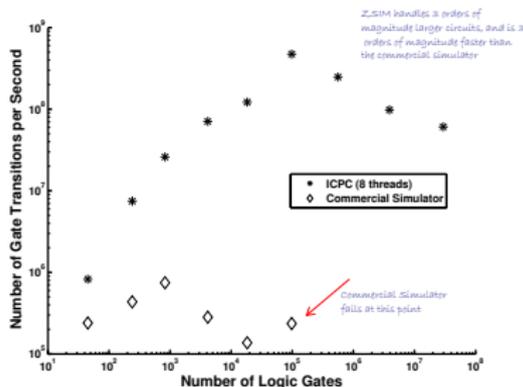


Figure: Number of gate transitions per second between the Commercial Simulator and SIMD ZSIM both running on Intel i7 for synthetic circuits (with inputs from any level)

Performance Comparison to Blue Gene/L Supercomputer

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism

Comparisons
Compilers

Summary

Table: Characteristic comparison of Intel Xeon phi and IBM Blue Gene/L

Parameter	IBM Blue Gene/L	Intel Xeon phi
Cores	1024	60
Clock Speed	700 MHz/core	1.053 GHz/core
Price	\$0.8m - \$1.3m	\$1600.00 - \$2649.00
Size	2m height x 1m width	24.61cm x 11.12cm x 3.86cm

Table: Comparison of number events per second (IBM Blue Gene/L vs. Intel Xeon Phi)

Machine	Number of gates	Cores/Threads	Event rate (millions/sec)
Blue Gene/L	\simeq 216 million	512	60
		1024	116
Xeon Phi	\simeq 160 million	125	76.8
		240	142

1 Xeon Phi thread is as powerful as 4 Blue Gene/L

Performance Comparison Across Compilers

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

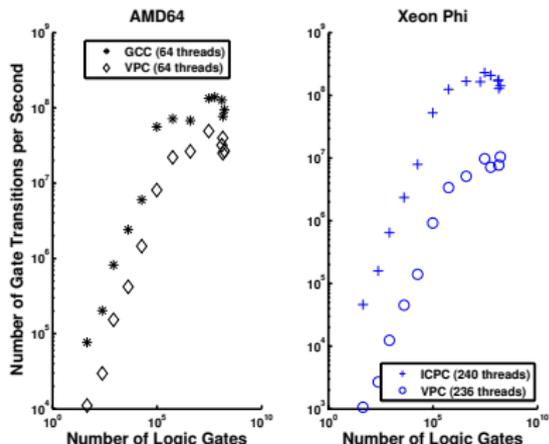


Figure: Comparison of number of transitions per second of the parallel simulator across different compilers on both AMD Opteron and Xeon Phi machine

Performance Comparison Across Compilers

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

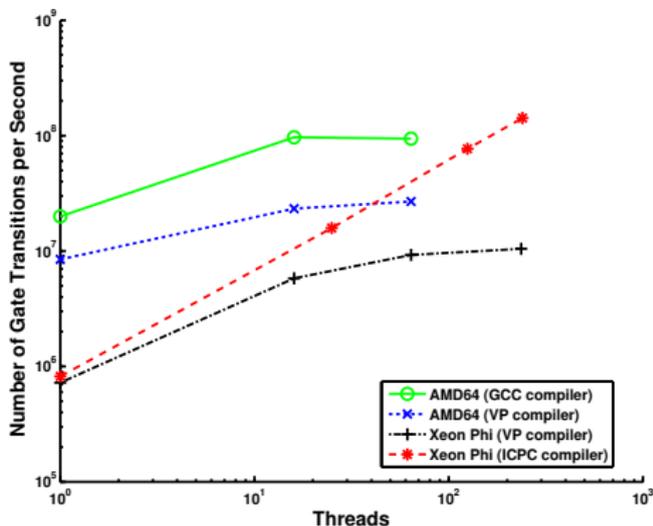


Figure: Comparison of number of transitions per second of parallel simulator on both Intel Xeon Phi and AMD Opteron, compiled by both Vector Pascal and Intel compiler for circuit size of 170M

Summary

SIMD simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

- Verified that the data structures used allow SIMD acceleration, particularly on machines with gather instructions.
- Verified that, on sufficiently large circuits, substantial gains could be made from multi-core parallelism.
- Showed that a simulator using this approach outperformed an existing commercial simulator on a standard workstation.
- Showed that the performance on a cheap Xeon Phi card is competitive with results reported elsewhere on much more expensive super-computers.

SIMD
simulation

M. Chimeh,
P. Cockshott

Importance Of
Simulation

Simulation
Algorithms

Circuit
Representation

SIMD
Simulation

Machines

Results

Setup
Parallelism
Comparisons
Compilers

Summary

Thank You