ACCELERATED FINITE STATE MACHINE TESTING USING GPUS

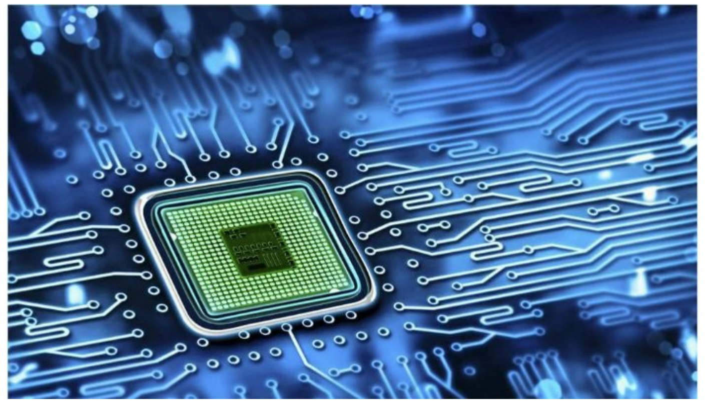
VANYA YANEVA

CDT in Pervasive Parallelism (cohort 2015)
PPar Lunch Series
4 April 2018

THE UNIVERSITY *of* EDINBURGH
informatics

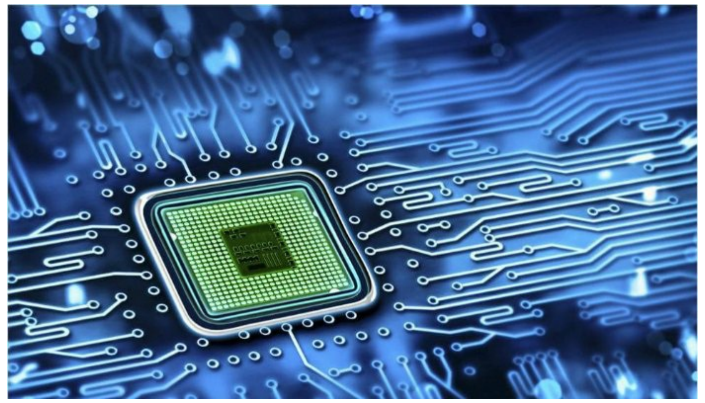
EPSRC Centre for Doctoral Training in
Pervasive Parallelism

SOFTWARE IS EVERYWHERE



- SAFETY AND CORRECTNESS ARE CRUCIAL
- TESTING IS CRITICAL

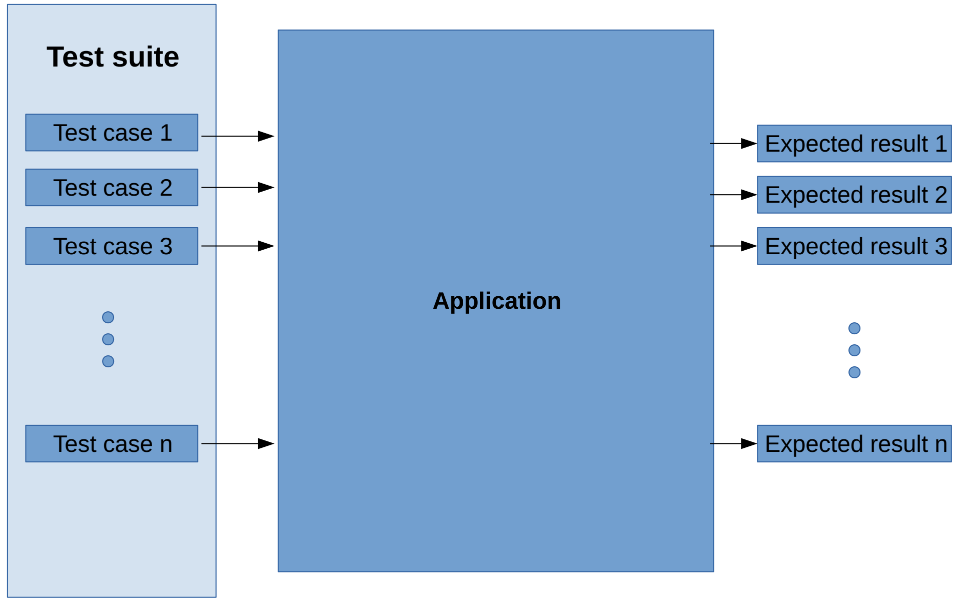
SOFTWARE IS EVERYWHERE



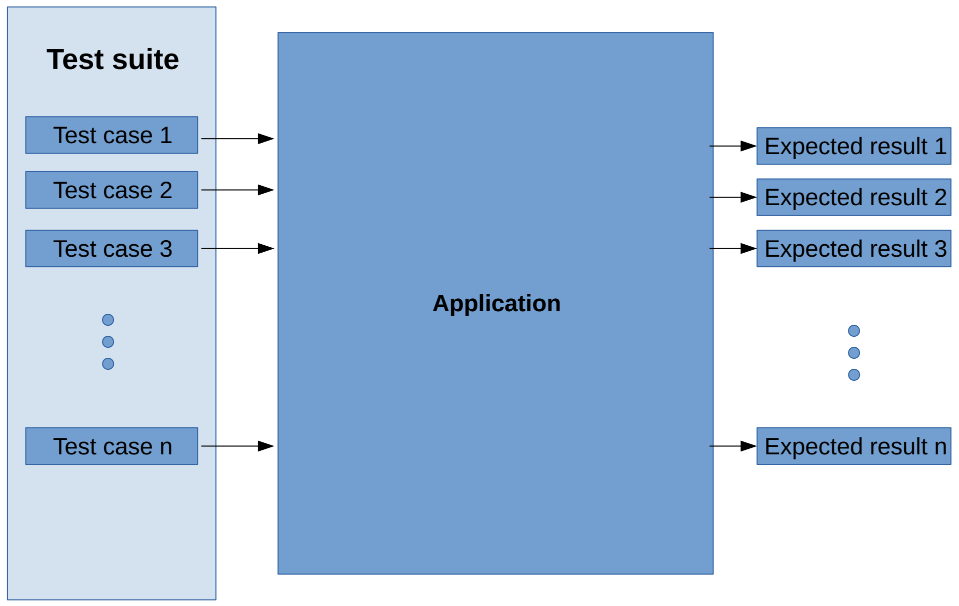
- SAFETY AND CORRECTNESS ARE CRUCIAL
- TESTING IS CRITICAL

BUT TESTING CAN BE EXTREMELY TIME-CONSUMING

FUNCTIONAL TESTING

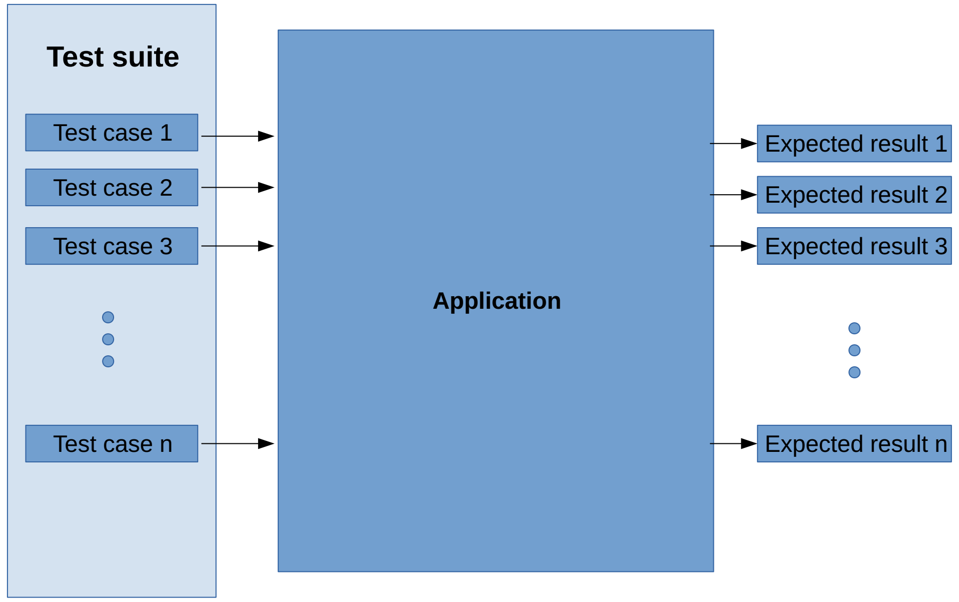


FUNCTIONAL TESTING



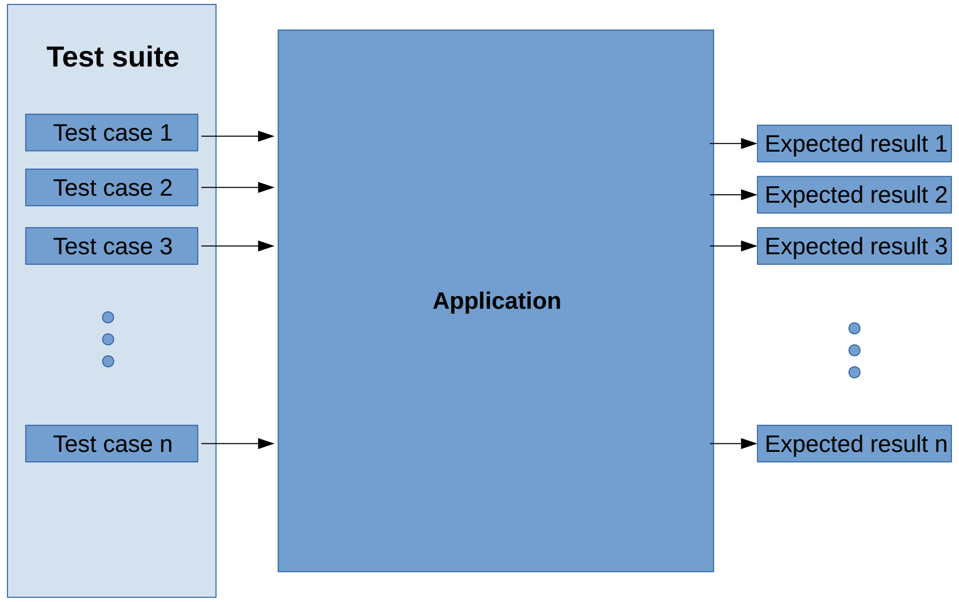
- Test cases are data parallel

FUNCTIONAL TESTING



- Test cases are data parallel
- Test executions are independent

FUNCTIONAL TESTING



- Test cases are data parallel
- Test executions are independent

Testing is an ideal candidate for parallelisation

EXECUTE TESTS IN PARALLEL

CPU SERVERS

- Expensive
- Do **not** scale easily as test suites grow
- Can be extremely underutilised

EXECUTE TESTS IN PARALLEL

CPU SERVERS

- Expensive
- Do **not** scale easily as test suites grow
- Can be extremely underutilised

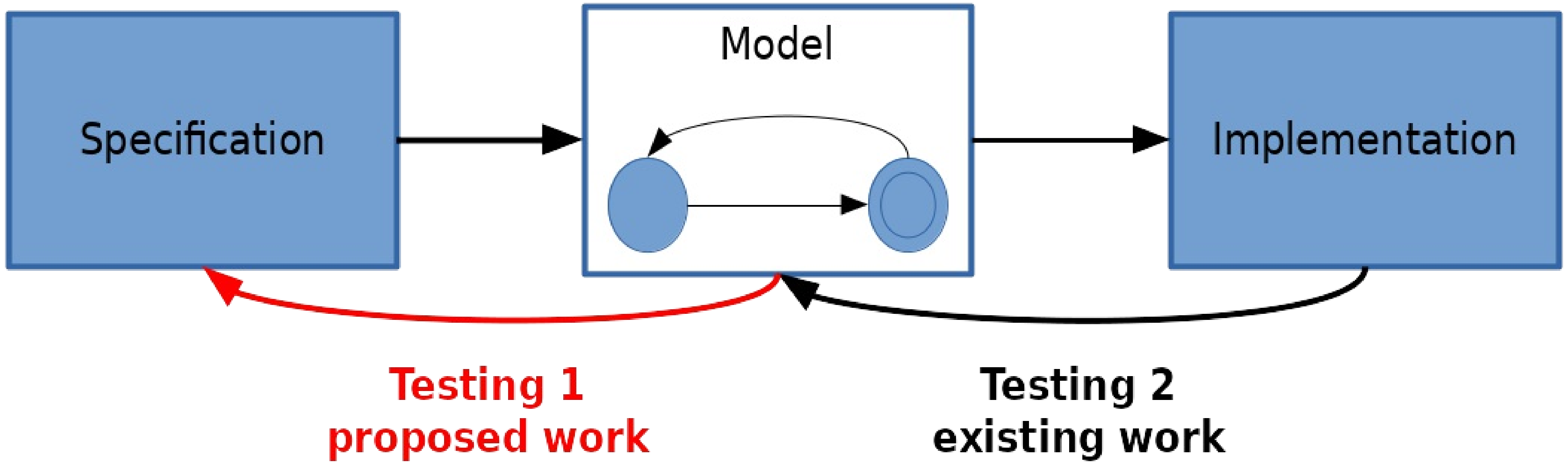
GPUS

- Cheap and widely available
- Large-scale parallelism, thousands of threads
- SIMD architecture suited to functional testing

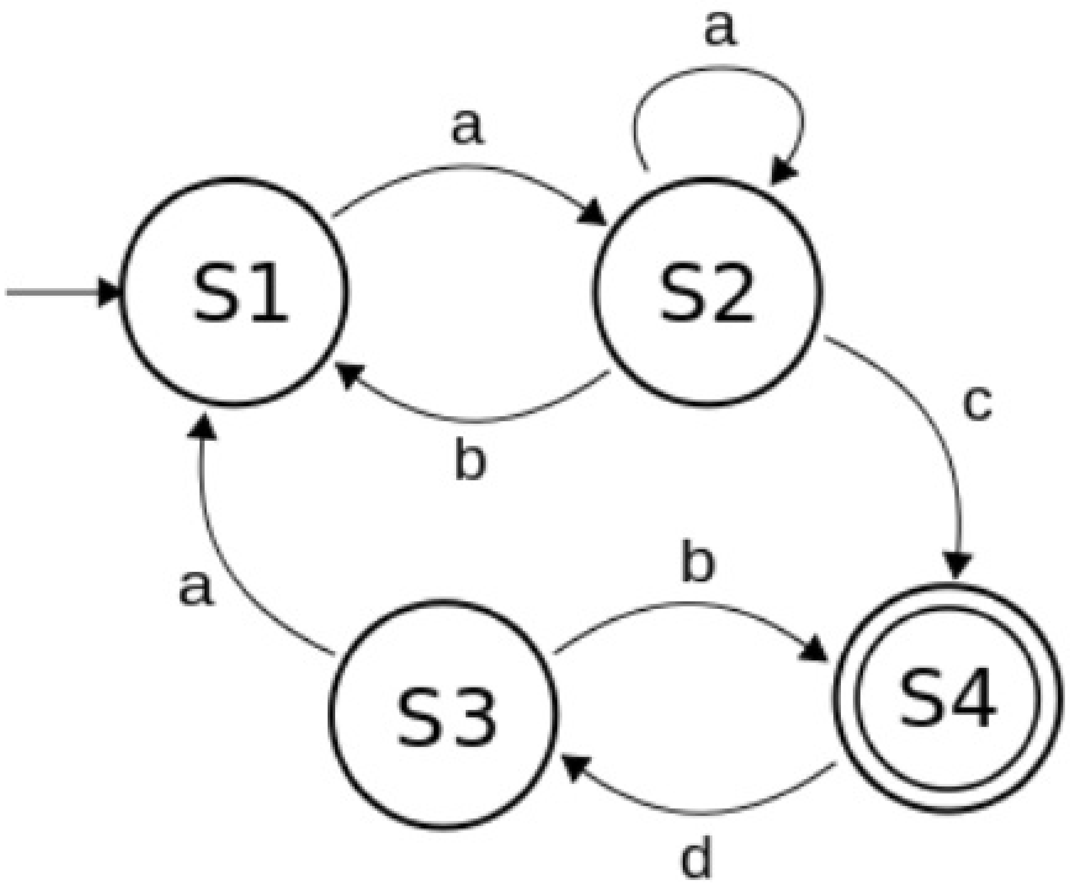
TALK OUTLINE

- **Domain:**
  Finite State Machine testing for model-based development
- **Case study:**
  Keysight Technologies
- **Proposed solution:**
  Execute tests in parallel on the GPU threads
- **Challenges**

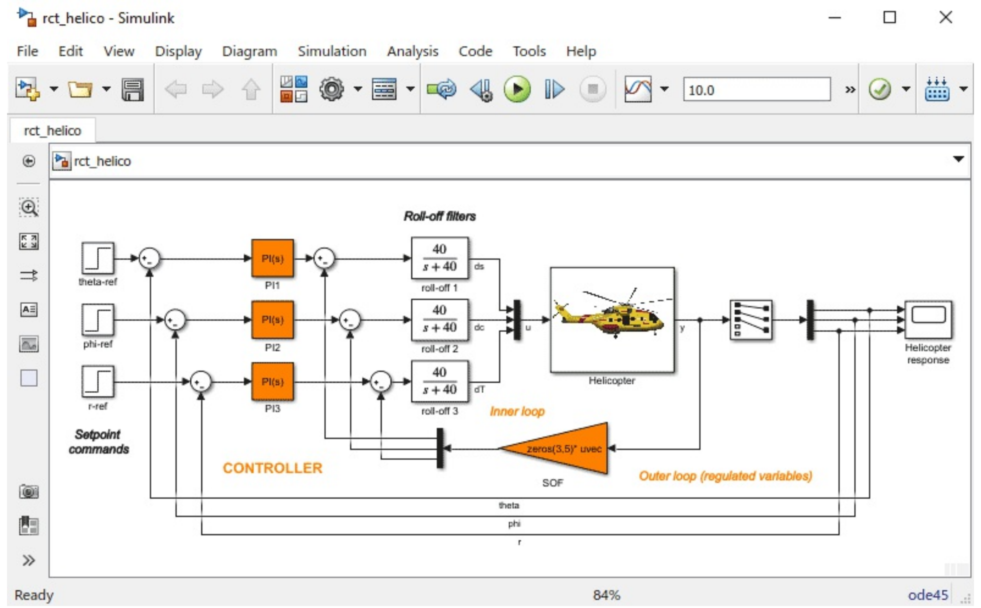
MODEL-BASED DEVELOPMENT

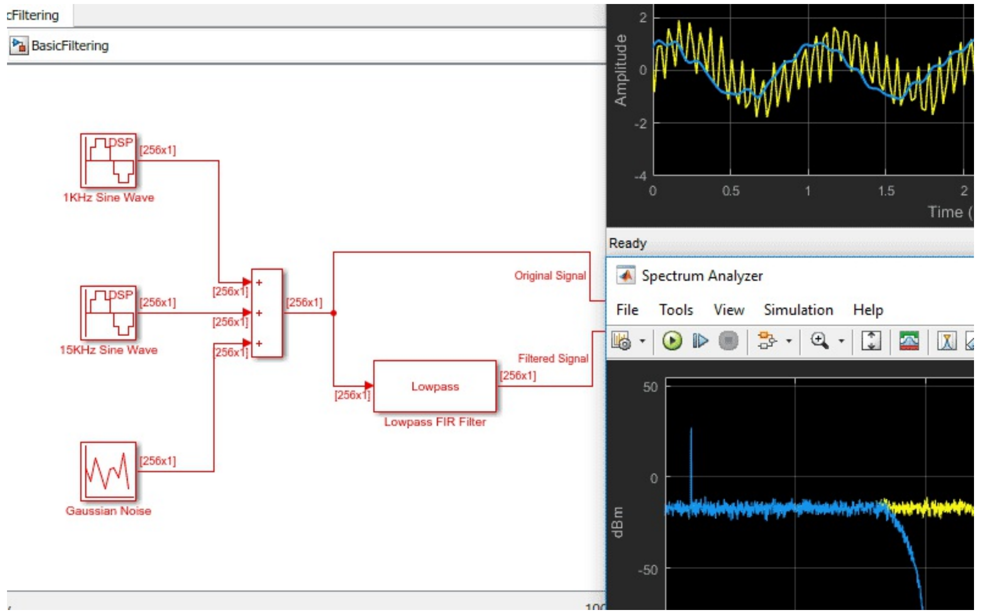


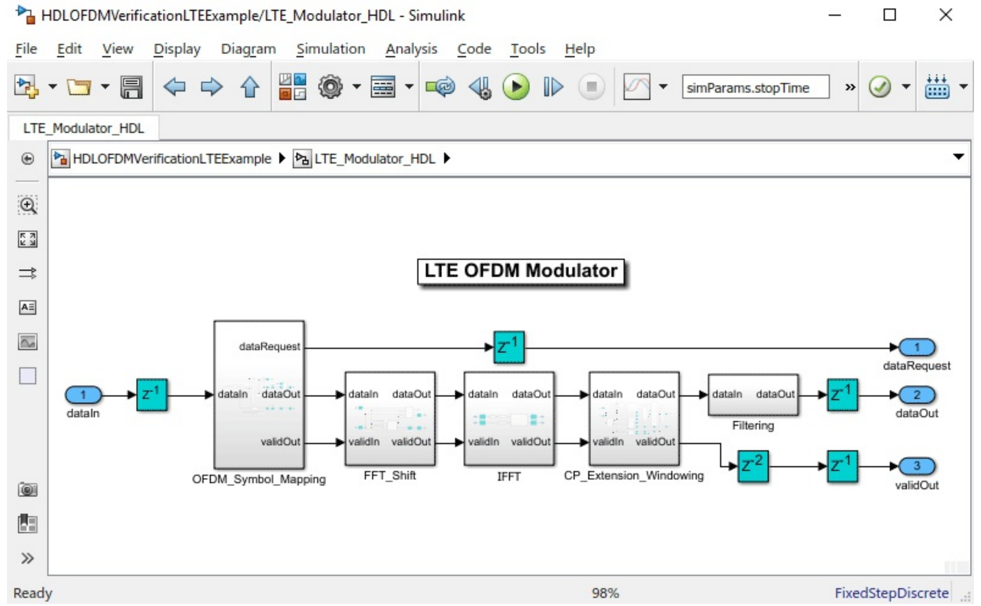
FINITE STATE MACHINES



FINITE STATE MACHINES







CASE STUDY

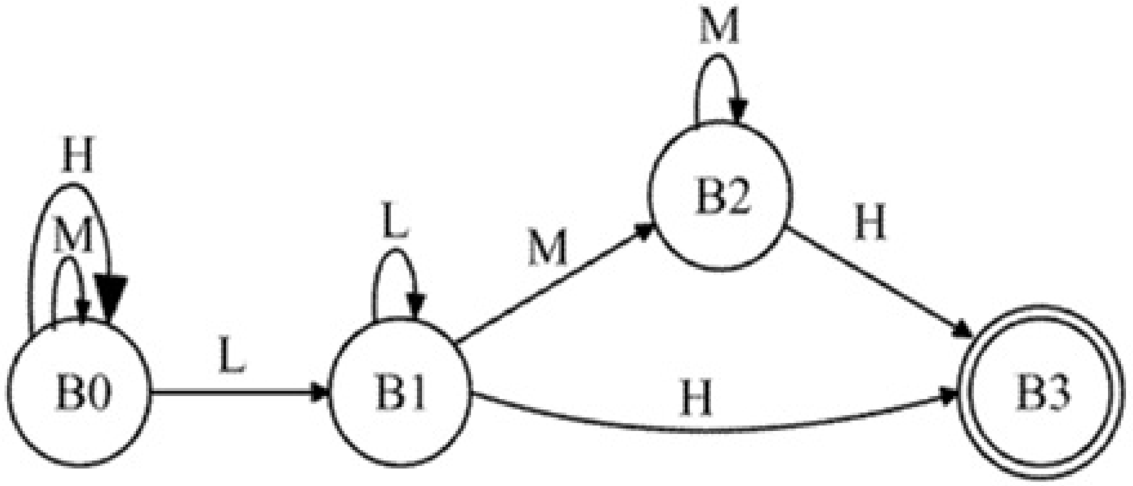
KEYSIGHT
TECHNOLOGIES

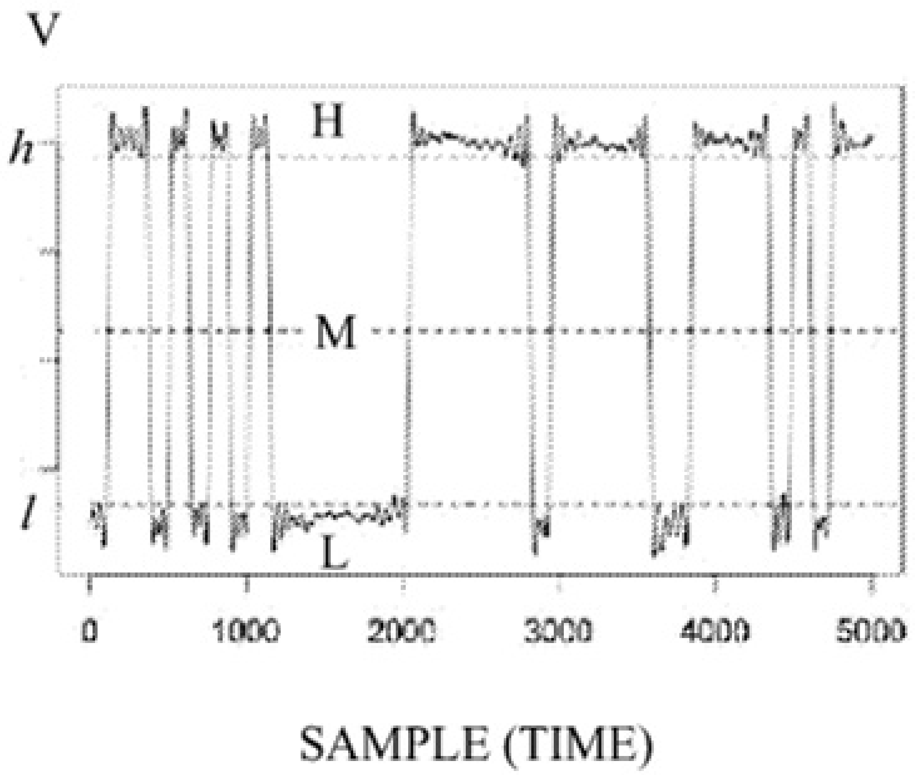
Measuring tools for:

- semiconductor
- aerospace
- wireless communications

CASE STUDY

KEYSIGHT TECHNOLOGIES





Lehane A.R., Kirkham A.J.A. and Barford L.A., *Digital Triggering Using Finite State Machines*, 2016, US Patent App. 14/957,491

PROPOSED SOLUTION

Execute tests in parallel on the GPU threads

1. Read the FSM and its test cases
2. Transfer it to GPU memory
3. Execute test cases in parallel
4. Transfer results to CPU memory

CHALLENGES

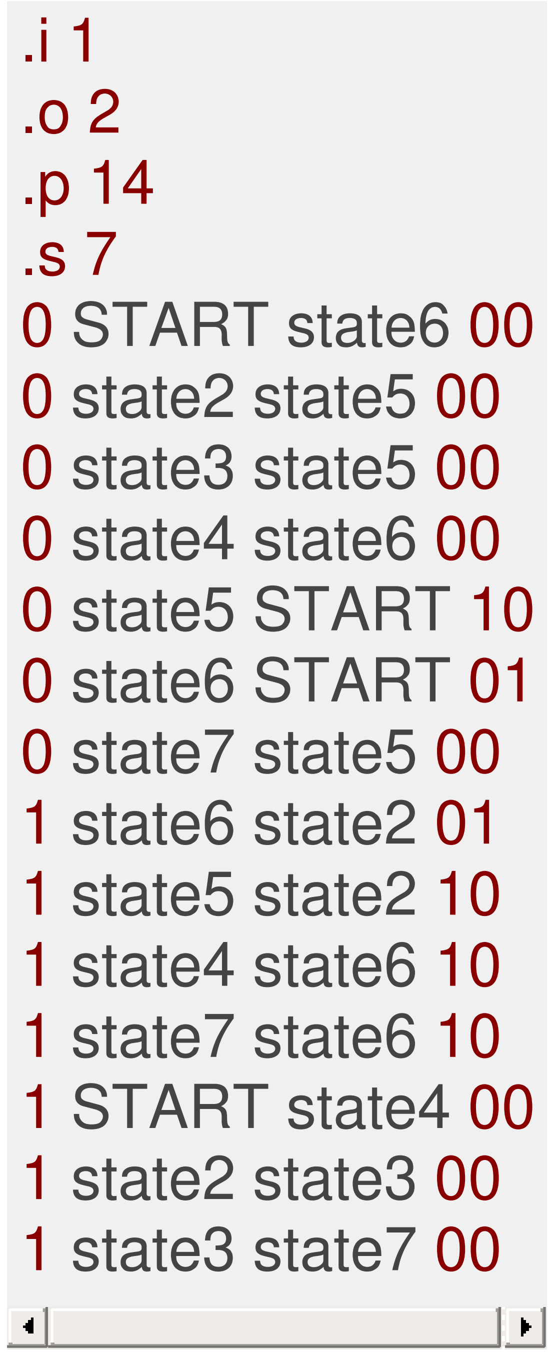
1. What FSM representation should we use?
   Large FSMs need to fit onto GPU memory.

2. What subject FSMs can we find for evaluation?

3. How do we generate representative test cases?

4. Performance?

FSM REPRESENTATION



```
.i 1
.o 2
.p 14
.s 7
0 START state6 00
0 state2 state5 00
0 state3 state5 00
0 state4 state6 00
0 state5 START 10
0 state6 START 01
0 state7 state5 00
1 state6 state2 01
1 state5 state2 10
1 state4 state6 10
1 state7 state6 10
1 START state4 00
1 state2 state3 00
1 state3 state7 00
```

```
struct transition {
  char input[2];
  short current_state;
  short next_state;
  char output[3];
};
```

SUBJECT FSMS

- ACM SIGDA Open Source Benchmarks

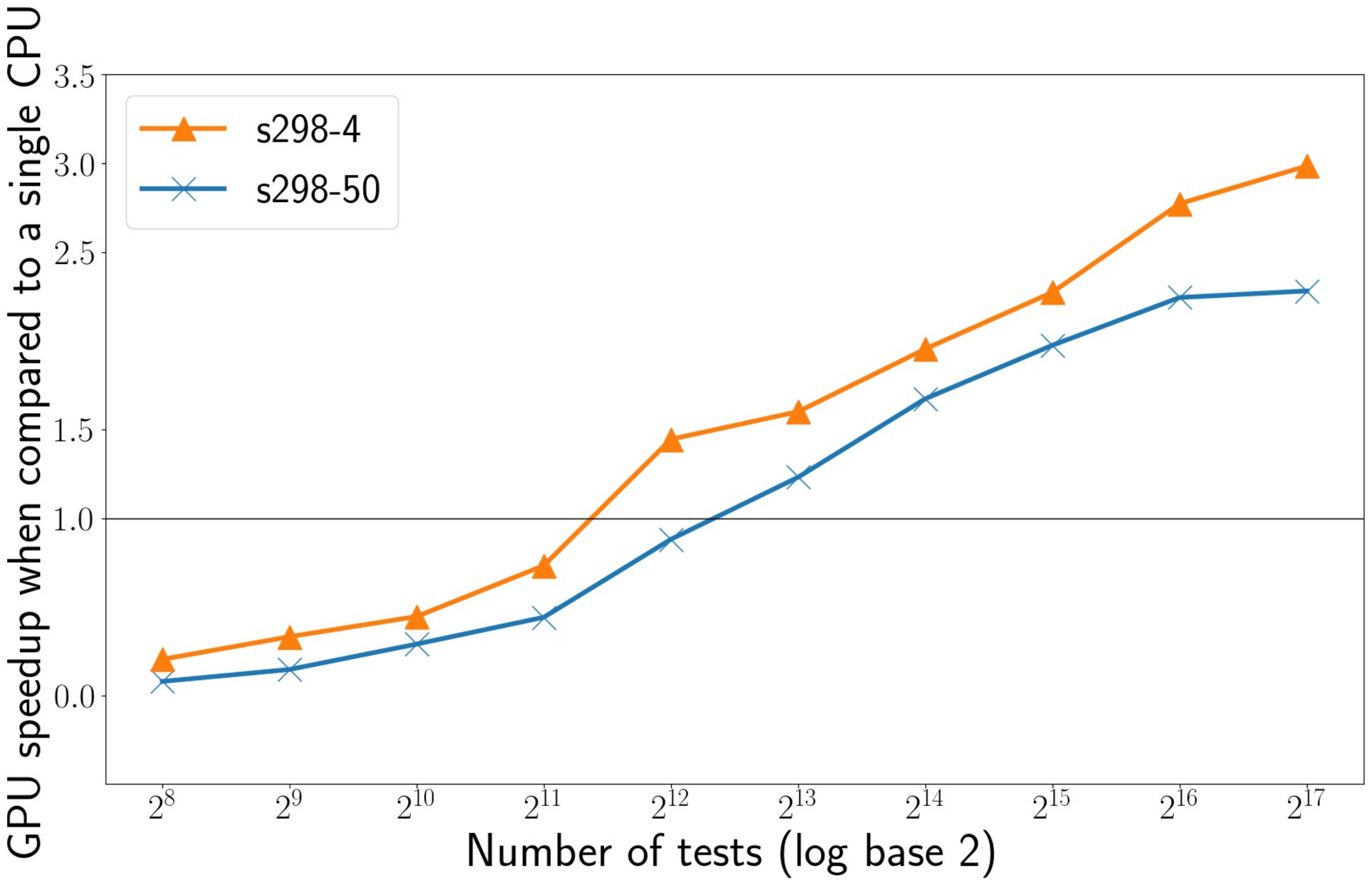
  52 FSMs for circuit design, 4 to 218 states, 10 to 1096 transitions, binary i/o of different lengths
- Examples from Keysight
- FSMs used in existing research (?)

REPRESENTATIVE TEST CASES

- Request them from Keysight and other researchers.
- Generate them based on state and transition coverage.
- The length and number of test sequences could have implications for performance.

PERFORMANCE

s298 - 218 states and 1096 transitions; GPU - NVidia Tesla K40m, CPU - Intel Xeon



- Shorter test sequences are better
- Frequent accesses to global memory are a performance bottleneck
- We can group test cases based on length and similarity

CONCLUSION

Finite state machines are important for model-based development.

GPUs provide exciting possibilities for the acceleration of their testing.

I am still at the preliminary stages, but I hope to have solutions soon.