About Me

- 1st year PhD student at Communications CDT, University of Bristol
- Enjoying RTL, SoC, microarch work since 2010.
- Research title: Machine Analysis for SoC Behaviour Optimization
  - Understanding Complex Architectures and Systems
- Sponsored by UltraSoC Technologies Ltd

Research Problem

The most difficult problem is knowing what questions to ask, and what the answers should look like.

- Why does/did that happen?
- What might happen next?

Behaviour can be described with statements.

- "DDR access occurs just after cache miss."
- "The faster you get into idle, the less power you use."

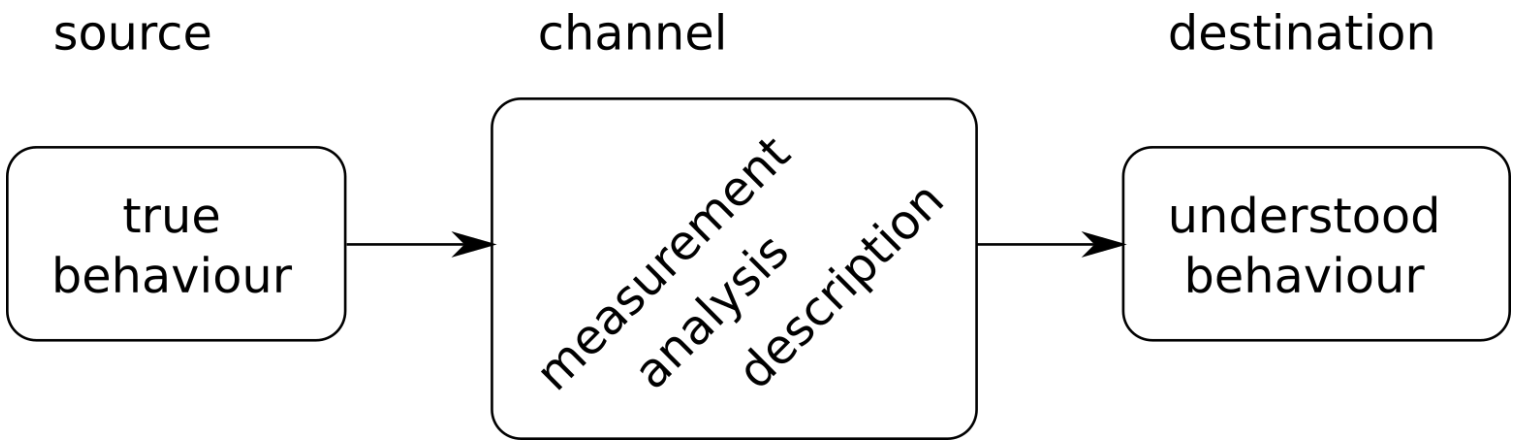
Assertion: Understanding is a complete set of precisely worded statements.

Application to HPC and Manycore

- Better, stonger, faster. Cheaper, smaller, thriftier.
- Specialization and optimization.
- Measurable aspects:
  1. Off-chip, real value
     - Temperature
     - Voltage
  2. Off-chip, event
     - DDR/SPI/I2C access
     - Peripheral powerdown
  3. On-chip, real value
     - Memory usage
     - Cache hit rate
  4. On-chip, event
     - Cache hit/miss
     - Instruction fetch/complete/cancel
- All are connected through complex relationships.

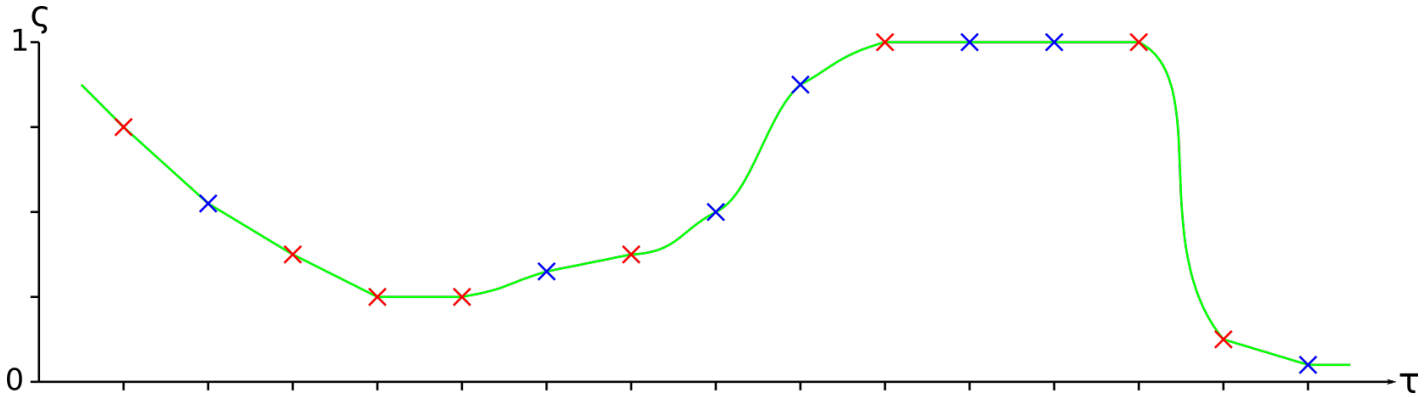
Practicallity

- Harmony with existing tools.
  - Cross-platform, easy install, few environment assumptions.
  - VCD/waveforms, Python/Numpy, web browsers, text files.
- Understanding is only meaningful when it's in your head.
  - Intuitive maths.
  - Explainable results.

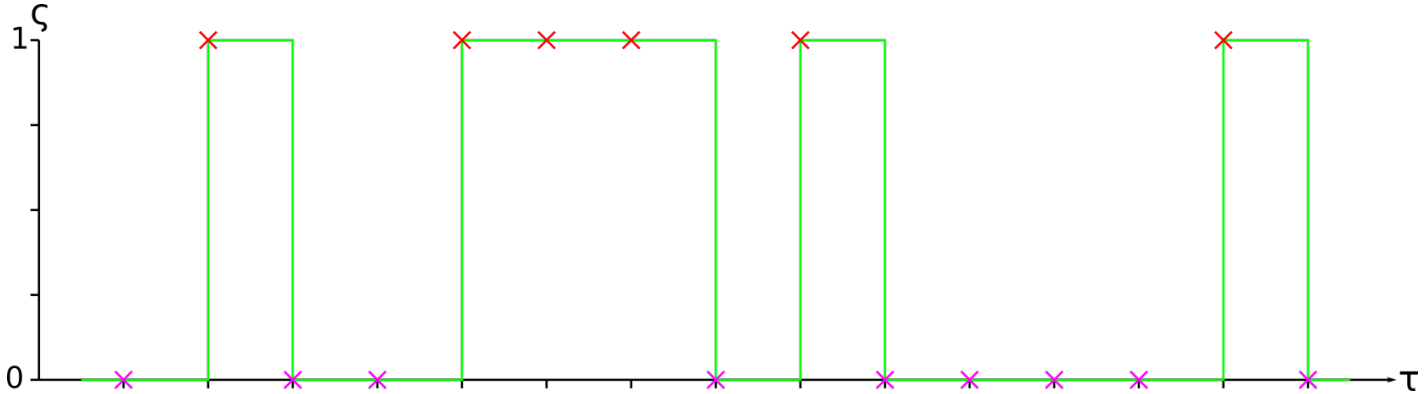


Model

$$e := (\psi \in [1, m],\ \tau \in \mathbb{N},\ \varsigma \in [0, 1],\ \beta \in [-1, +1]) \tag{1}$$



(a) Function of an evaluation, such as "voltage".



(b) Function of an event, such as "cache miss".

Method (1) - A Language for SoC Behaviour

Adapting the language of Probability.

$$Pr(X) = \frac{1}{N} |X| \tag{2}$$

$$Pr\Big(f_x;\ t \in [u, v)\Big) := \frac{1}{v-u} \int_u^{v-1} f_x \ dt \tag{3}$$

$$Pr(X \mid Y) = \frac{Pr(X \cap Y)}{Pr(Y)}, \quad \text{if } Pr(Y) \neq 0 \tag{4}$$

$$Pr\Big(f_x \mid f_{y_{\langle\delta\rangle}};\ t \in [u, v)\Big) = \frac{\int_u^{v-1} f_x * f_{y_{\langle\delta\rangle}} \ dt}{\int_{u+\delta}^{v-1+\delta} f_y \ dt} \tag{5}$$

Method (2) - More Interesting Statements

- Appearance and disappearance of dependencies over time is interesting.

$$Pr(X \cap Y) = Pr(X)Pr(Y) \iff X \perp\!\!\!\perp Y \tag{6}$$

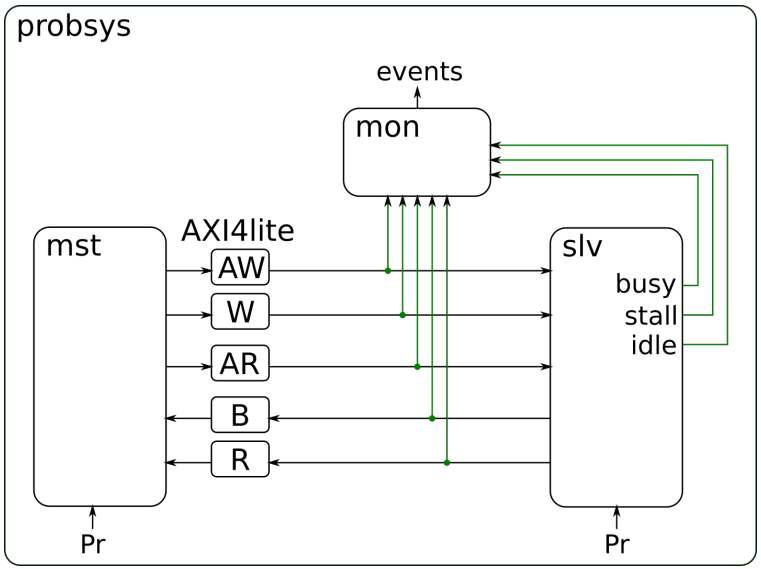
$$Pr(f_x \mid f_y) = Pr(f_x) \iff f_x \perp\!\!\!\perp f_y \tag{7}$$

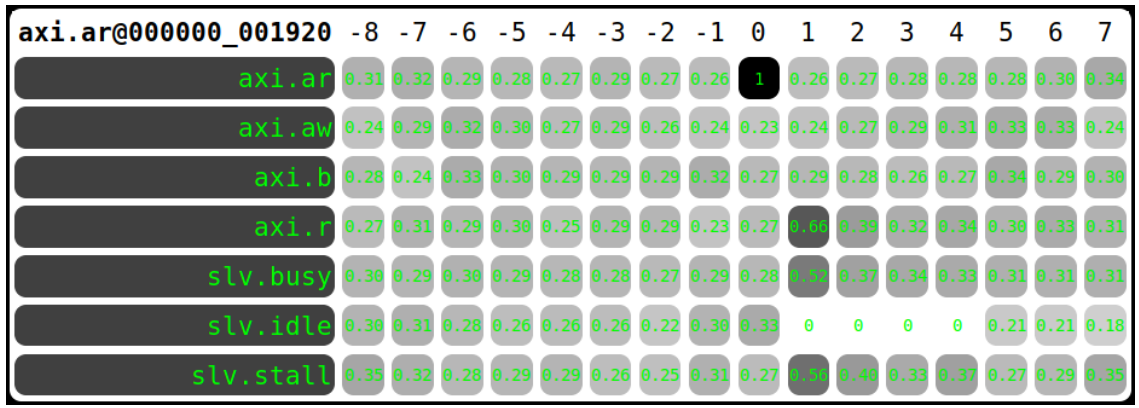
- Complex relationships defined by functions.

$$Pr(f_x \mid f_{y_1 \circ y_2 \circ \cdots \circ y_n}) \tag{8}$$

Example (1)

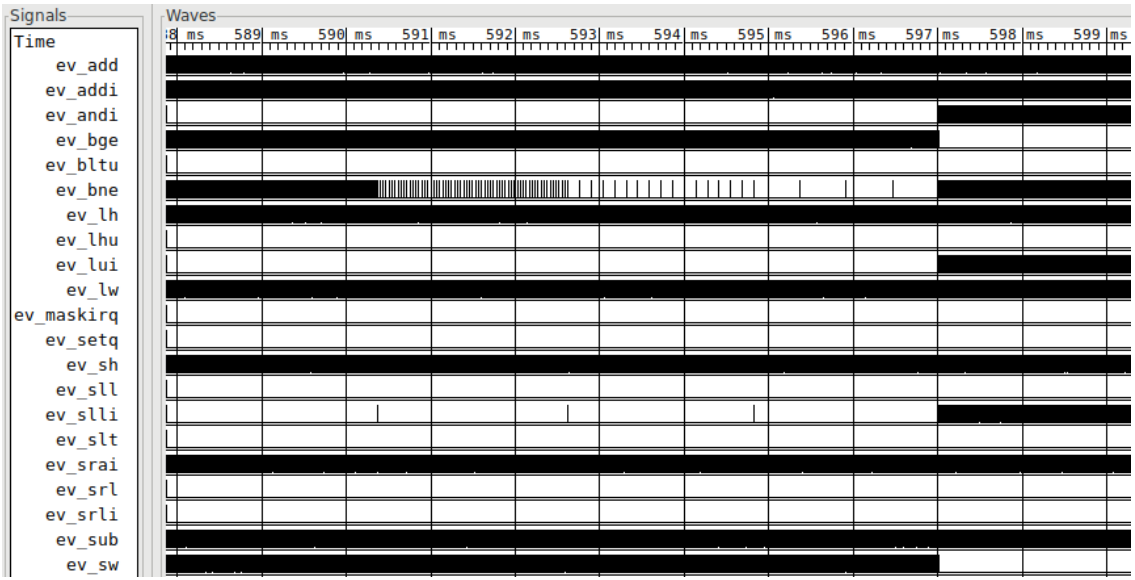
- Simple probabilistic single master, single slave system.
- Visualization of $Pr(f_x = \texttt{axi.ar} \mid f_y)$ for one time window.
- Behavioural features apparent from image.

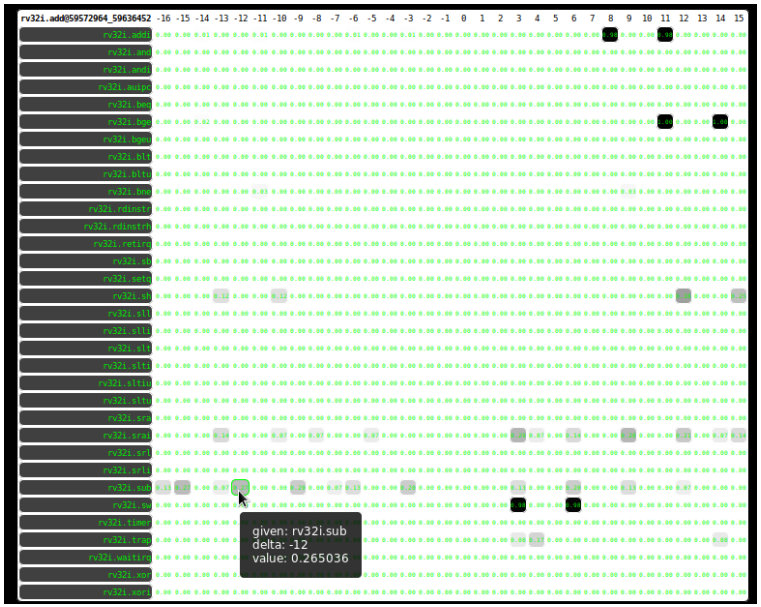




Example (2)

- picrorv32 - RV32IM running radix4 1024 point FFT
- Probe insn completions with existing signals in unmodified design.
- Visualization of $Pr(f_x = \texttt{add} \mid f_y)$ for one time window.





Limitations and Future Directions

- Conditional probability result space is large. $(\#\delta \times m^2 \times \#u)$
- Result space of complex relationships is even larger.
- Efficient calculation is important.
- Choosing what to calculate is more important, necessitating machine learning techniques rather than brute force.
- Activation functions based on β simulate human perception, match the analyst's intuition.
- Learning suitable thresholds for $f_x \in \mathbb{R}$ is high compute risk, high entropy payoff.
- Markov Chain discovery is ultimate goal.

Conclusion

- Questions?