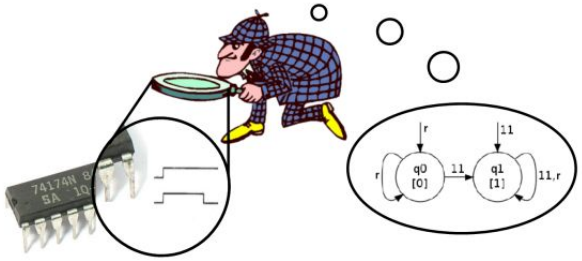
Specialized Strategies for Learning Integrated Circuits using Angluin L* and Rivest/Shapire Homing Inference



Tanya Braun, Arne Wichmann, Sibylle Schupp

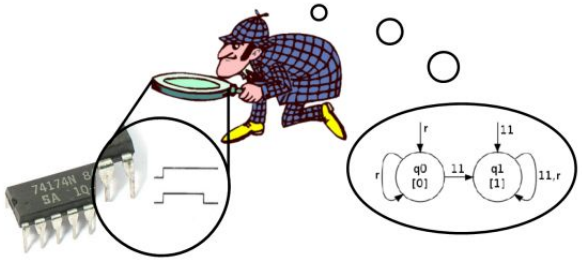
Institute for Software Systems, Hamburg University of Technology

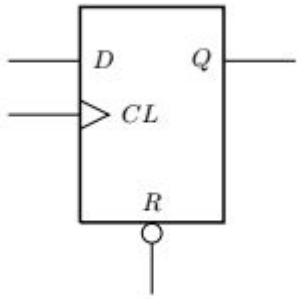
STS Software Technology Systems

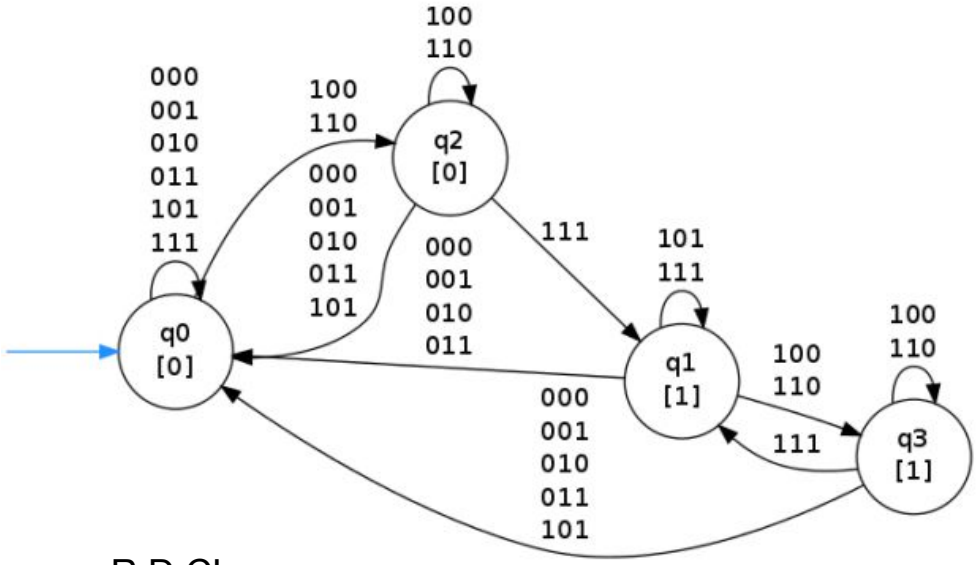
Blackbox Learning Digital ICs

Angluin L*: Learning.

- Stimulate, watch, learn.
- Check and terminate on equivalence.

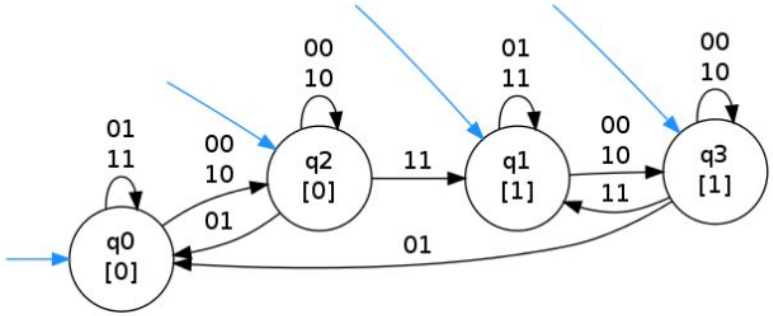


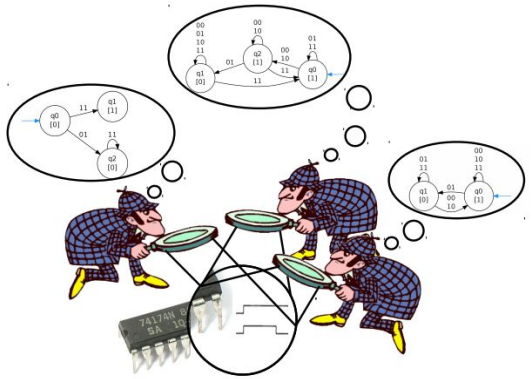




R D CL

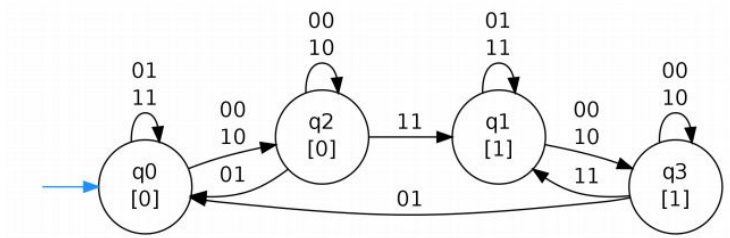
Blackbox Learning Digital ICs



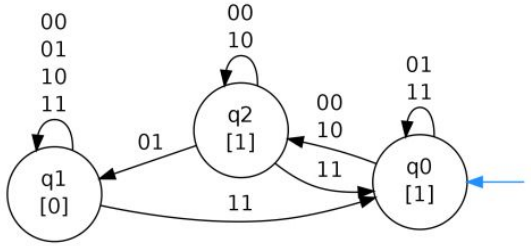


Rivest/Shapire homing: Missing reset.

- Use homing sequence to recognize learner to be updated.



Learner 0



Learner 1

Homing sequence: $h = 01$

Blackbox Learning Digital ICs

Problem:

- Blackbox implies approximative equivalence.

Use specialized strategies to check equivalence.

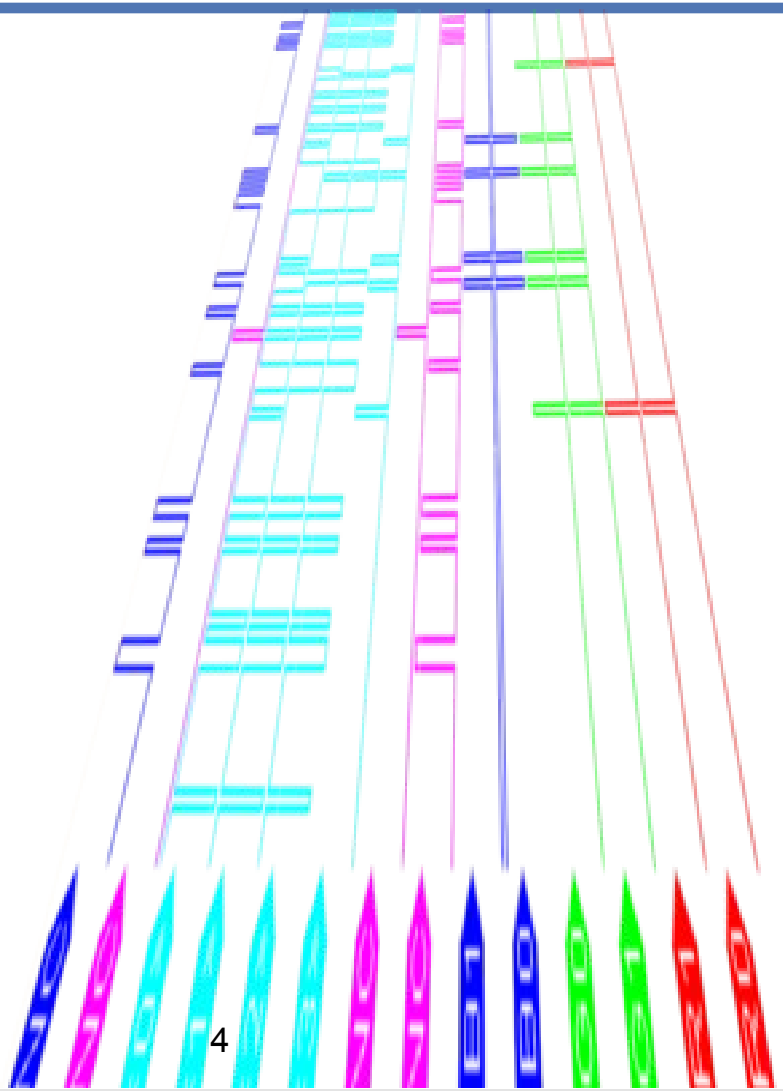
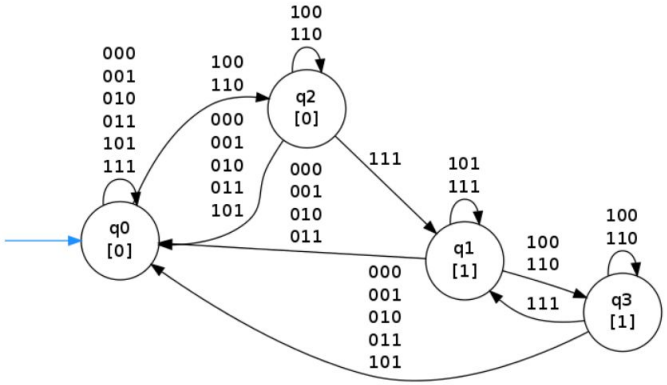
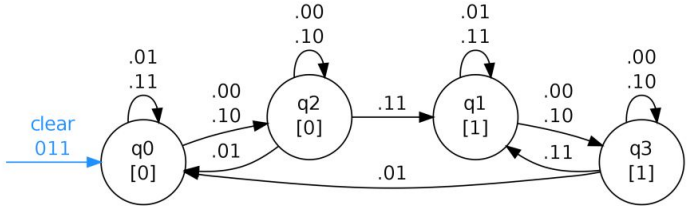


Figure based on Image By Mhinner (Own work) [Public domain], via Wikimedia Commons

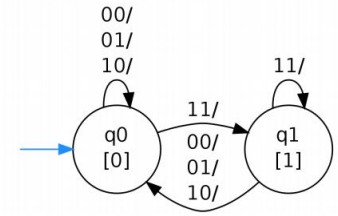
Approximating: Engineering Stuff



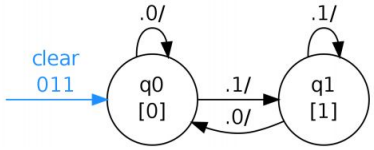
Without any prior knowledge



Clear pin known

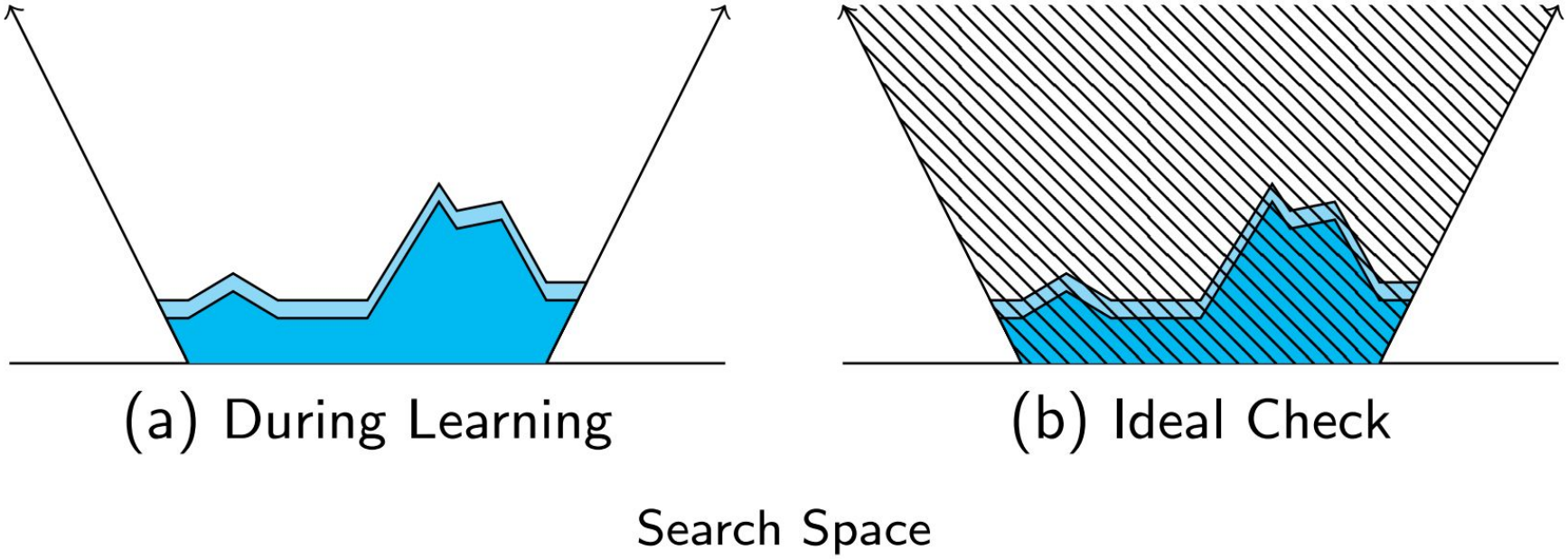


Clock pin known



Both pins known

Exploring State Space

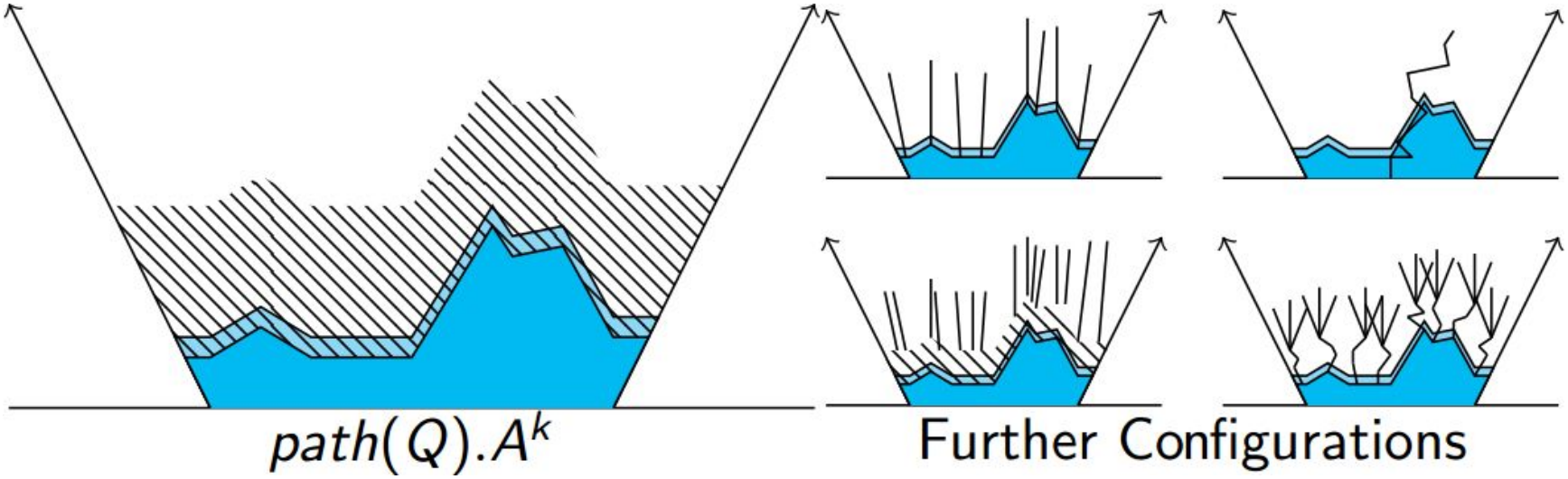


x-Axis: Input/alphabet.

y-Axis: Time/steps/progress (to infinity).

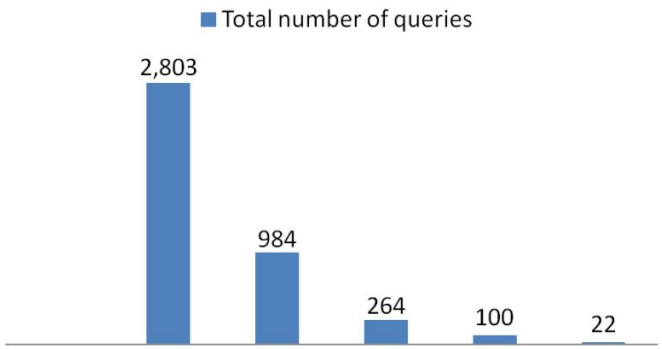
Baseline: Reset/homing/init.

Exploration Strategies

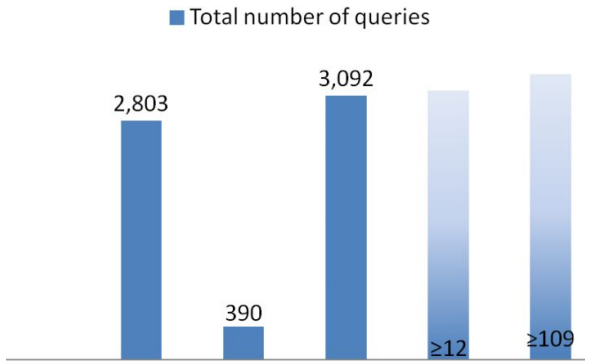


Automata	Strategies
Densely connected.	Alphabet stimulation from known states.
Chain bridges.	Toggle to find critical pins.
Unknown.	Random

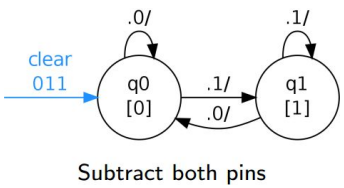
Example Flip-Flop



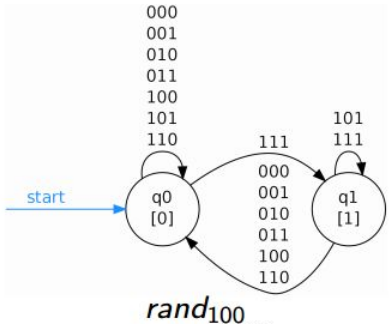
	Plain	Use clear word	Subtract clear pin	Subtract clock pin	Subtract both pins
Input pins	3	3	2	2	1
States	4	4	4	2	2
Learners	5	1	1	3	1



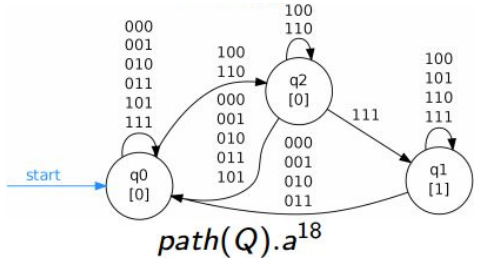
	Plain	path(Q). a^{18}	path(Q). A.a^{18}	path(Q). $rand_{10}$. a^{18}	$rand_{100}$
States	4	**3**	4	**≥1**	**≥1**



Subtract both pins

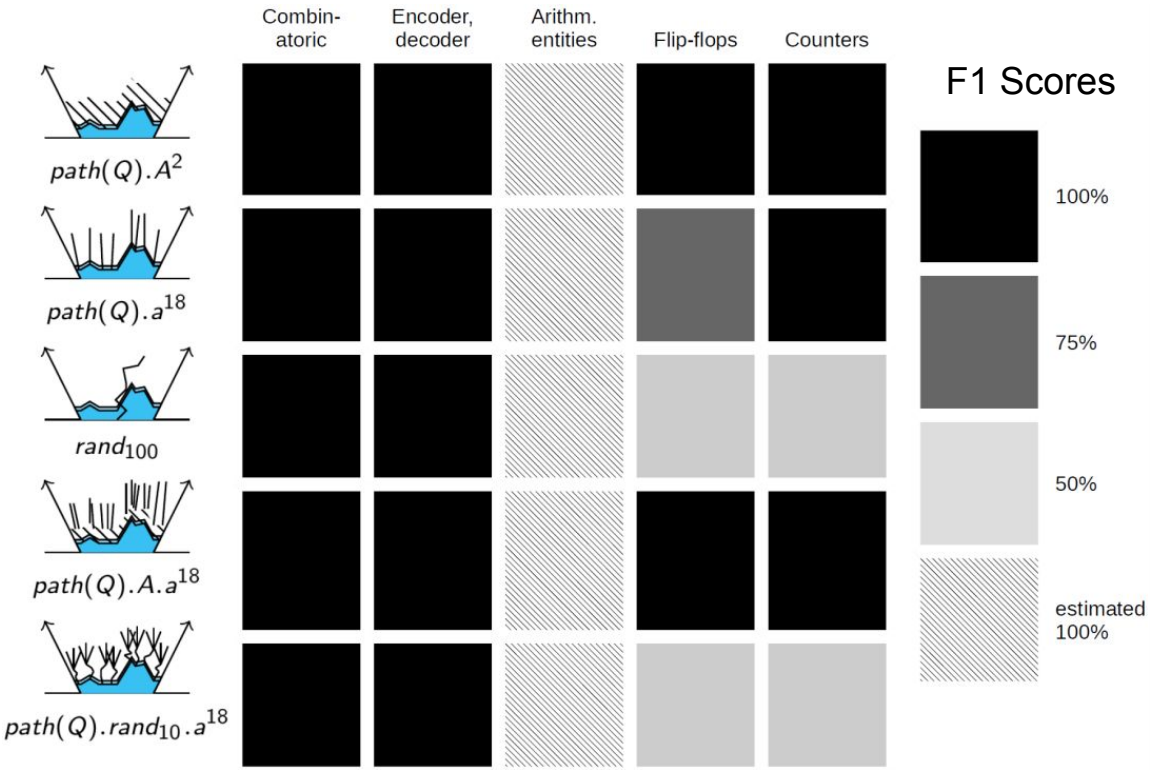


$rand_{100}$



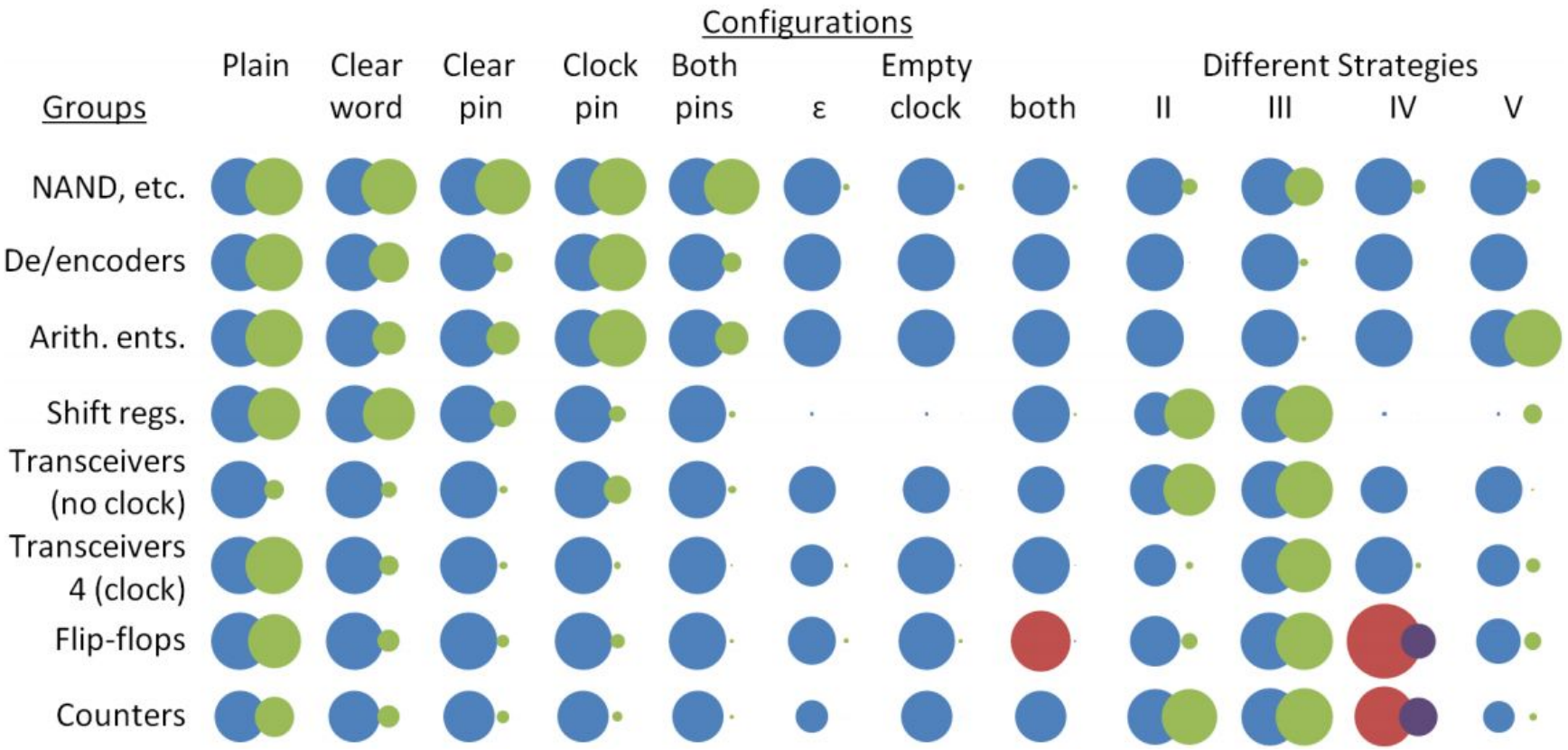
$path(Q).a^{18}$

Evaluation Results: Quality



- Evaluation using 116 VHDL models of the 7400 series ICs (Free Model Foundry).
- Case studies using real hardware.

Results: Quality and Cost



F1 Score (blue) and Costs (green) relative to the maximum cost within a group in terms of queries per configuration and group of ICs; the red-colored F1 score and violet-colored costs indicate inconsistent data.

Conclusion

- Strategies give good overall results.
  - Plain (Path + Alphabet) for the general case.
  - Counters need toggle for lookahead.
  - Engineering information (clear/clock) helps significantly.
- Random based strategies did not perform well. (They usually trigger a reset or automata growth.)
- Cost strongly depends on pin count/alphabet:
  - Abstract groups of pins to variables (arithm. ent.)?
  - Learning using sparse alphabets?