Less is more: efficient hardware design through Approximate Logic Synthesis

Università della Svizzera italiana

Dr. Ilaria Scarabottolo
USI Lugano, Switzerland

Partition and Propagate

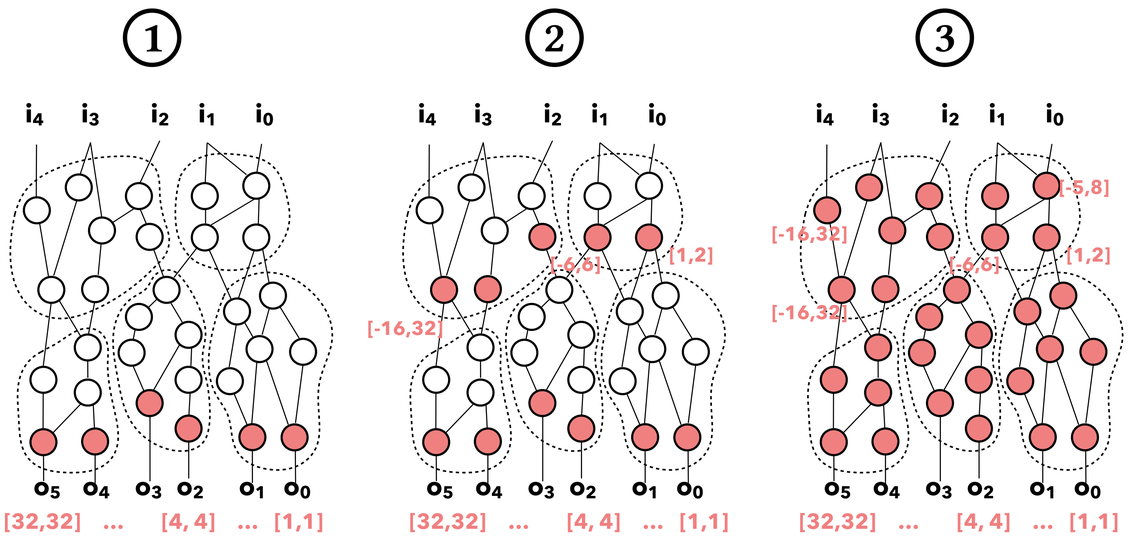
Approximate hardware design techniques require an accurate error model, which can be employed for guiding the ALS algorithm towards more efficient solutions, for instance, which gate should be pruned first.

Key idea: estimate the **maximum influence** of each circuit gate **on the final output**, **by partitioning** the circuit **and propagating error intervals** among (small) subcircuits.

Related Work

- gate-level error-derivation techniques:
  - ***sum weighting***[1] - **non-accurate**
  - ***whole-circuit simulation***[2] - **non-scalable**
  - ***SAT solver derivation***[3] - **non-scalable**
  - ***Monte Carlo input selection***[4] - **no upper bound**

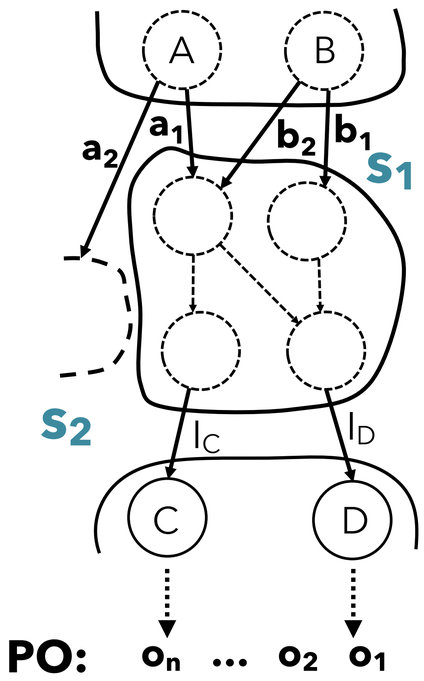
Intervals derivation



output intervals are set to the bit significance

intervals are propagated upwards exploiting subgraph semantics

subgraphs are simulated to derive internal intervals



a_1 B	C D
0 0	0 1
0 1	1 0
...	...

$I_C - I_D$

Propagation

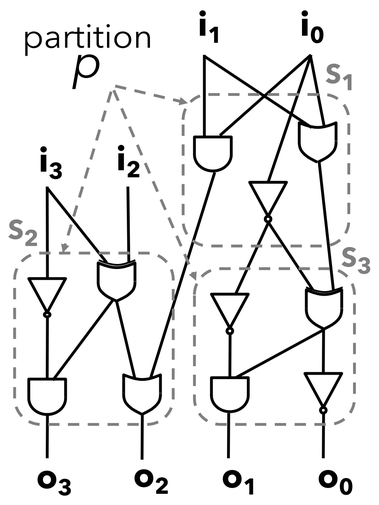
- intervals for nodes A and B, are computed observing the truth tables of $\mathbf{s_1}$ and $\mathbf{s_2}$
- Intervals in form $I_A = [l_A, u_A]$
- $\mathbf{I_B}$: when the value of B changes, C increases while D decreases,
  $I_B = \mathbf{I_C} - \mathbf{I_D} = [l_C, u_C] + [-u_D, -l_D]$
- $\mathbf{I_A}$: children belonging to different subgraphs. No single truth table, hence
  $I_A = I_{a1} + I_{a2}$
- **final bound** on maximum error:
  $w_B = \max(|l_B|, |u_B|)$

Graph partitioning

- circuit represented as DAG G(N,E)
- purpose: label nodes with intervals I_i, representing the influence of each gate on the approximate output (in terms of maximum and minimum induced error):
- Definition of partition function:

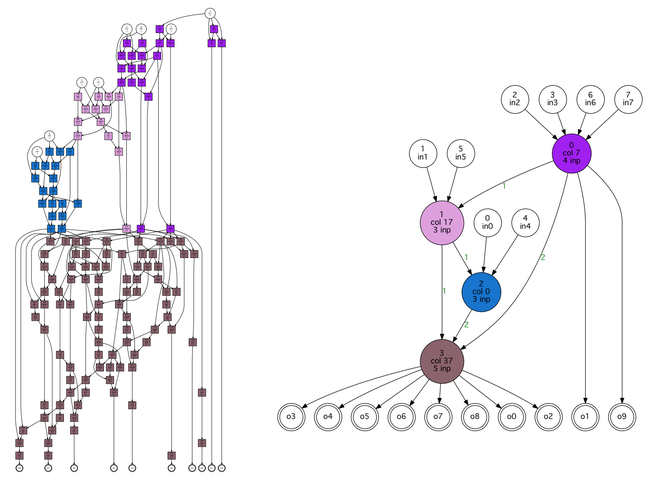
$$p: G(N,E) \mapsto S$$

$$S = \{\, s(N_s,E_s) \mid N_s \subseteq N,\ E_s \subseteq E \,\}$$

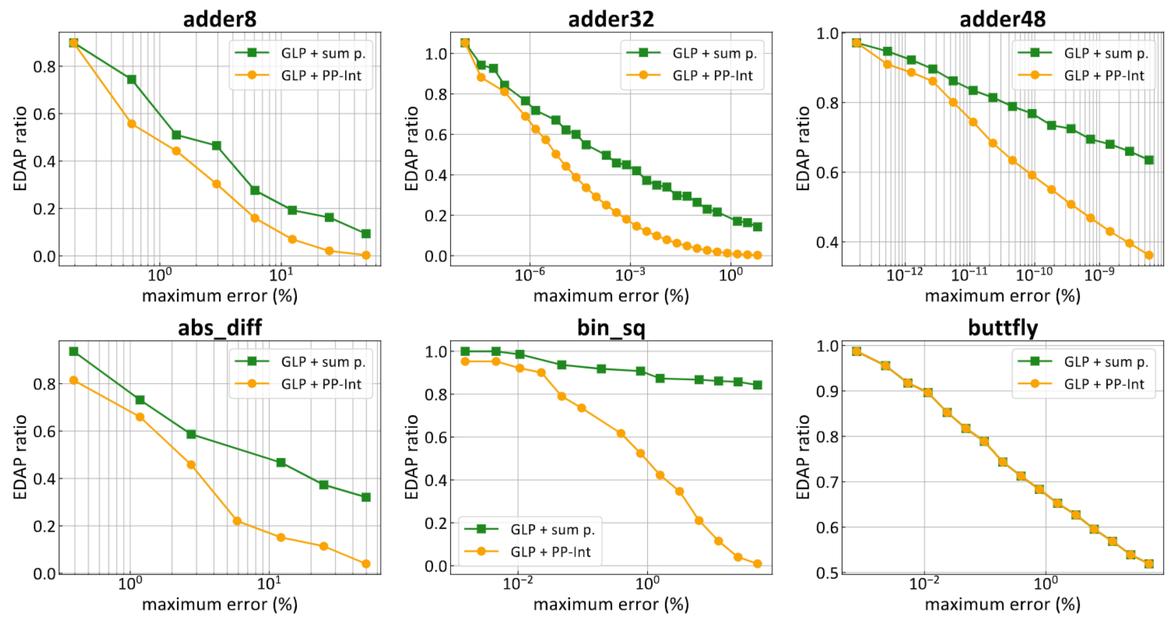


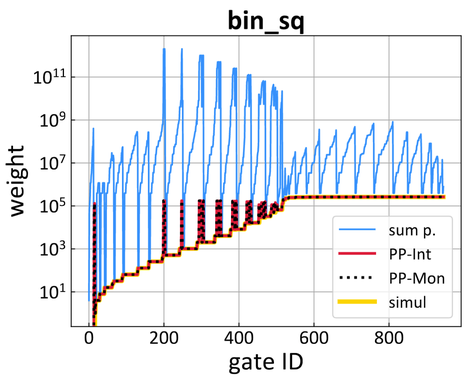
- each node $n_i \in N$ belongs to one and only one subgraph $s(N_s,E_s) \in S$, and $U(N_s) = N$.
- the number of input per subgraph must be smaller than a threshold T
- all children of the same node should belong to the same subgraph

Partition example of a 4-bit binomial squared: all children of the same node are coloured with the same colour



Experimental results





Time performance (N=400):

- Monte Carlo [4]: > **1h**
- SAT [3]: **8 m**
- Partition and Propagate: **18 s**

References

[1] J. Schlachter, V. Camus, K. V. Palem, and C. Enz. *VLSI*, 25(5):1694–1702, May 2017.
[2] I. Scarabottolo, G. Ansaloni, and L. Pozzi. DATE 2018, pages 545-550, March 2018.
[3] R. Venkatesan, A. Agarwal, K. Roy, and A. Raghunathan. ICCAD 2011. 667–673.
[4] S. Hashemi, H. Tann, and S. Reda. DAC 2018. 55:1–55:6.