

# Weekly Report

March 20, 2019

## 1 Targets

### 1.1 Urgent

- Randomly selecting a node, compute its care set, change its care set, synthesize the local circuit, evaluate its area and error rate, accept it with a certain probability.

### 1.2 Important

- Use approximate confidence interval / hypothesis testing of Bernoulli experiments to evaluate the accuracy of error rate.
- Trade off the accuracy of batch error estimation for speed (even directly use Su's equation to update Boolean difference), perhaps use hypothesis testing to evaluate the accuracy.
- Combine the simulation of circuits with the simulation of Monte Carlo Tree Search. In other words, in one loop of Monte Carlo Tree Search, merge logic simulation and playout (only simulate circuit once and playout once).
- Represent circuit with AIG because of more potential LAC candidates. For each round, select one or more input wires and replace them with constant 0 or 1. Consider how to combine Wu's method (choose a subset of input wires and substitute).
- Accelerate Approximate Logic Synthesis Ordered by Monte Carlo Tree Search: reuse the result of batch error estimation in playout.
- How DC affects BDD
- Induce DC with PLA files

### 1.3 Worth Trying

- Enhance default policy with greedy approach or field domain knowledge.
- In expansion process of MCTS, expand more than one layers.

- Tune parameters in MCTS.
- Perform greedy flow on leaves of the final Monte Carlo Search Tree.
- Combine beam search and MCTS.

## 1.4 Potential Topics

- Relationship between power simulation and logic simulation.
- Combine Binarized Neural Network with approximate computing.
- Relationship between Boolean network and Bayesian Network.
- Approximate TMR.

## 2 Progress

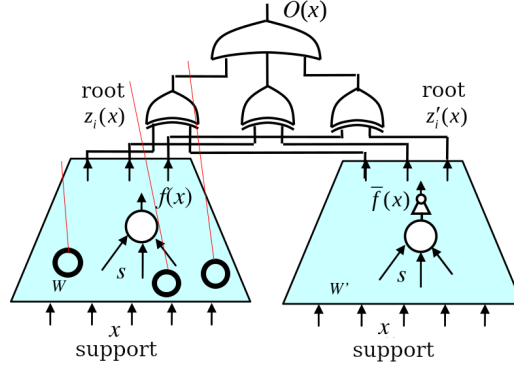
### 2.1 Count #Literals for Su’s work

See “./single-selection-result-20190319.xls”

### 2.2 Attempt on Don’t Care Based ALS

Currently, I modify the “mfs” command. The framework is listed below:

1. Select a node  $f$ . (Currently in topological order)
2. Collect the window for  $f$ , whose inputs are primary inputs  $\mathbf{x}$ , outputs  $\mathbf{z}$  are nodes in  $f$ ’s fanout cone.
3. Build a miter  $O(\mathbf{x})$  for  $f$ .
4. Set approximate don’t care assignments by change the miter. (which will be discussed later)
5. Build the SAT problem.
6. Solve SAT, find interpolation and change the circuit.
7. If exists other unprocessed node, return to 1.



## 2.3 How to Add Approximate DC assignments

For a miter,

$$O(\mathbf{x}) = 0, \mathbf{x} \in DC \text{ set}$$

$$O(\mathbf{x}) = 1, \mathbf{x} \in \text{care set}$$

We want to set an assignment  $\hat{\mathbf{x}} = x_1 x_2 \dots x_n$  as don't care, where  $x_i = 0/1$ . Then for  $\mathbf{x} = \hat{\mathbf{x}}$ ,  $\hat{O}(\mathbf{x}) = 0$ , for  $\mathbf{x} \neq \hat{\mathbf{x}}$ ,  $\hat{O}(\mathbf{x}) = O(\mathbf{x})$ . So

$$\hat{O}(\mathbf{x}) = \bar{\hat{\mathbf{x}}} O(\mathbf{x})$$

## 2.4 Test on c880

Since the approximate DCs are directly added on primary inputs, the number of approximate DCs  $m$  decides an upper bound of error rate.

$$er \leq m/2^n$$

where  $n$  is the number of primary input.

For c880,  $n = 26$ .

m	er	area	time/s
0	0	607	0.189
1	0.000000014	589	0.207
10	0.000000149	594	0.241
100	0.000001490	553	0.595
1000	0.000014901	532	5.452
10000	0.000149012	408	70.104

It seems to be promising, compared to area of 506 under 5% error rate in Wu's result.

Time increases as  $m$  grows because of the number of clauses in SAT solver increases. I have two techniques to solve it:

- Synthesize the approximate DCs clauses before adding to a solver.
- Limit the input of the miter window. (But how to assess the error rate)