Locating Error Sources in Digital Circuit Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Modern integrated circuits are increasing in size and complexity — the latest Intel microprocessor exceeds 2.5 billion transistors, whilst NVIDIA's most advanced graphics accelerators are now at 3 billion transistors. As a result, more and more verification effort is required to ensure that these state-of-the-art devices are free from design errors. Building on previous work which attempts to predict the source of a malicious rumour in a network (or alternatively, locating the originator of a virus in an epidemic) using a maximum-likelihood estimator, this paper investigates applying such techniques to the problem of finding the source of a realistic design error inside a network-representation of an integrated circuit.

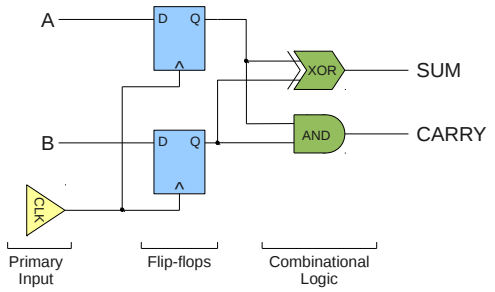
1 Introduction

Since their inception in the late 1950s, integrated circuits have come to infiltrate the lives of those across all spectrums of human society. As predicted (or perhaps, driven) by Moore's Law, these devices have made huge leaps and strides in integration — combining 3 billion transistors into an area no bigger than a postage stamp, as well as in performance and cost. To put this into perspective, had a $900 commercial flight from New York to Paris in 1978 evolved at the same rate, such a flight would now cost about a penny and take less than second [1]. This progress has enabled integrated circuits to be accessible to much of the world: the United Nations Foundation reports that with over 7 billion people residing on our fair planet, there are now more than 5 billion mobile phone subscriptions [2], for which each phone contains at least one, but likely more than one, integrated circuit.

However, "with great power comes great responsibility" — besides the technological marvel of being able to manufacture such capable yet intricate devices, more and more expense and effort is also required to ensure that these ICs are designed correctly. A recent example of this going awry is the design flaw in Intel's Sandy Bridge motherboard chipset that was only discovered after the product had shipped — necessitating a costly recall that analysts have estimated to hurt the company in the region of $700 million dollars [3].

The objective behind debugging is to locate the root cause of any erroneous behaviour. By considering the digital circuit as a network, this problem can be translated into a fascinatingly similar problem from another domain: how to infer the source of a malicious rumour in a (social) network, or alternatively, finding the originator of a virus in a population epidemic. Shah and Zaman first proposed a solution to this problem at NIPS 2009 [4], which was extended into a journal paper [5].

Using knowledge of the "infected" nodes, and information about the underlying network structure, the authors of this work constructed an estimator for the rumour source based on the notion of its rumour centrality. The authors then show that the node which maximizes this estimator corresponds to the maximum-likelihood estimate for the rumour source in an undirected, regular, tree network. However, rarely are networks in the real world tree structures, let alone regular ones (i.e. in which every node has the same number of neighbours) and so the authors finish by extending their rumour estimator to general networks and validate it through simulation.



A	B	CARRY	SUM	DEC
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	2

Figure 1: Representing digital circuits as a network, example circuit: half-adder

2 Rumour-source estimator

This section summarizes work from Shah and Zaman presented in [4, 5]. Within, the authors assume that rumours spread through a population according to the susceptible-infected SI epidemic model, which is a variant of the common SIR model with the recovered category removed. In the former model, infected individuals can pass the disease (rumour) onto those in the susceptible category, but those who are infected are not allowed to recover. Furthermore, the authors assume that infected individuals can only transmit the rumour to its immediate neighbours in the network, indicated by the existence of an edge. The probability of transmission is also assumed to be independent and identically distributed.

First, Shah and Zaman construct a rumour-source estimator for a regular, undirected, tree network and go on to prove that it corresponds to the maximum-likelihood estimate (given that no prior knowledge exists). As most networks are not regular trees, the authors extend their estimator to work with general trees and networks. This generalized estimator is given by:

$$\hat{v} \in \max_{v \in G_N} P(\sigma_v^* | v) R(v, G_N) \tag{1}$$

where $\hat{v}$ represents the rumour-source node estimate, G_N the graph of N infected nodes, $P(\sigma_v^*|v)$ the likelihood of a legal breadth-first infection path (i.e. the *spanning tree* of nodes through which all nodes in G_N were infected if node v was the source) and $R(v, G_N)$ the rumour centrality, which captures the number of possible infection paths. For the exact definition of each of these terms, we refer readers to references [4, 5], but simply-speaking these quantities have the following relationship:

$$P(\sigma_v^*|v) \propto \frac{1}{f(\sigma_v^*)} \tag{2}$$

$$R(v, G_N) \propto \frac{1}{g(G_N)} \tag{3}$$

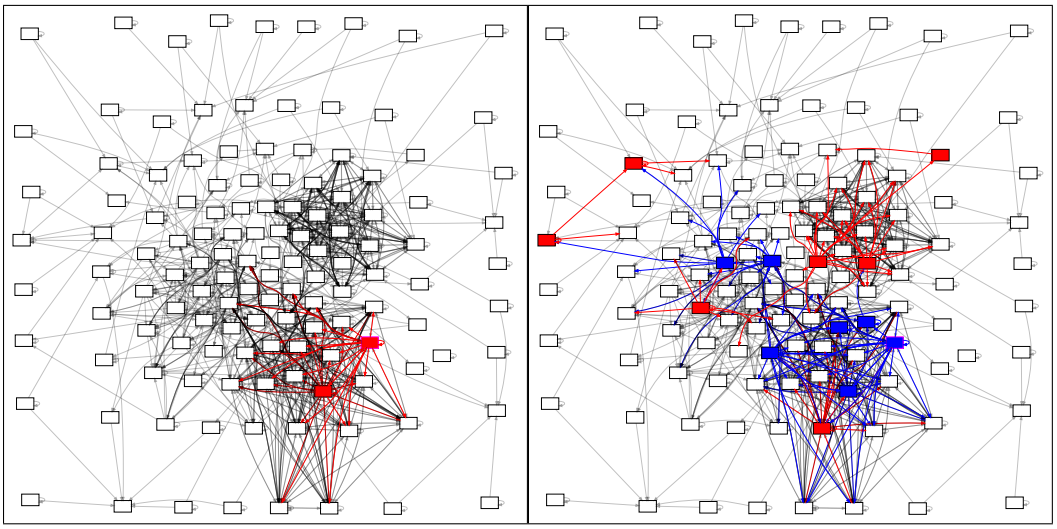
where $f(\sigma)$ is a function of the degree of each node in a legal infection path — used to capture the intuition that nodes with more neighbours are less likely to infect any *one* particular neighbour; $g(G_N)$ is a function of all possible sub-trees that exist in G_N — which seeks to reflect the infection path "complexity" and allow "deep" trees to be penalized over broader ones. Importantly though, this estimate can be computed in linear time making it extremely suitable for application to large problems.

These general graphs eliminate some of the simplifications that could previously be exploited in a regular tree (for example, that all infection paths were equally likely) explaining why this estimator is no longer the maximum-likelihood estimate. However, Shah and Zaman discovered that despite this flaw, their estimator performed very well in both synthetic and real generalized networks.

3 Digital circuits as a network

Digital circuits can be represented as a network, or graph, consisting of nodes that represent primary-inputs and -outputs through which the circuit exclusively interacts with the external environment, combinational elements (that are purely a function of their inputs, such as AND/OR/NOT gates) and sequential elements (which samples its input onto its output once per clock period: flip-flops). An example circuit, a half-adder, is shown in Figure 1.

The state of a circuit can be considered as being held in its flip-flops (which, for example, may implement the register bank of a microprocessor); with complete knowledge of all flip-flops, it is possible for a designer to compute the value of all combinational signals within. Hence, the structure



(a) G_N at t=673616ns (b) G_N at t=681656ns

Figure 2: Error snapshots for oc_i2c

of an integrated circuit can be simply represented by considering the connectivity between its flip-flops, and this results in a graph as illustrated by Figure 2. Each node represents one flip-flop inside a I^2C communication controller (oc_i2c) and each edge represents the existence of a logical connection from one flip-flop to another, possibly through combinational logic. We chose to use this oc_i2c circuit because it was sufficiently small to visualize and to allow the technique to be validated by hand, though we see no reason why these methods cannot be applied to larger circuits.

Because each wire in a digital circuit can only contain one driver, each edge is unidirectional making this a directed graph: flip-flop A may drive a signal which passes through multiple combinational gates to affect the value of flip-flop B, but this does not necessarily mean flip-flop B has a reciprocal effect on flip-flop A (unless of course, a path through a different set of combinational logic exists). More concretely, this graph captures the *dependency* relations between flip-flops, with one key difference: the graph is not guaranteed to be acyclic — that is, a path may exist between a node and itself — thereby creating an apparent circular dependency. However, as in the original work [5], we also make the assumption that errors propagate in a breadth-first fashion which means that cycles will not exist in any infection path σ_v^*. Consequently, we have modified the rumour-source estimator to work with directed graphs by reducing the degree contribution of each node to $P(\sigma_v^*|v)$.

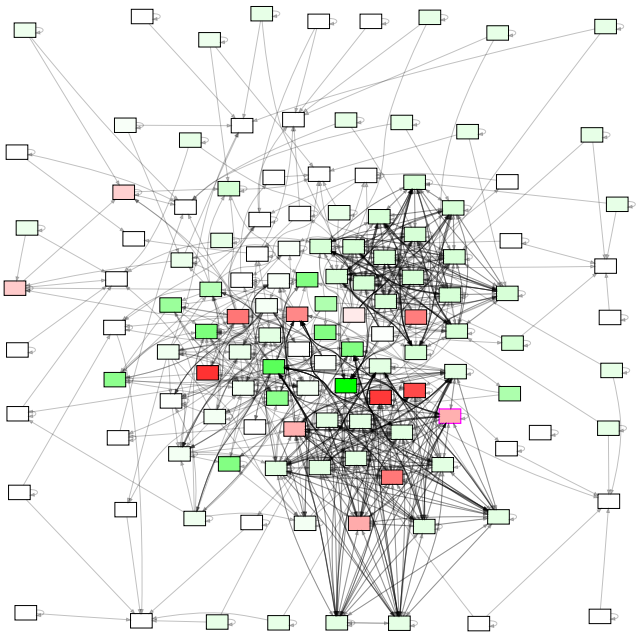
4 Results

In order to evaluate the application of the rumour-source estimator to a digital circuit network, we first simulated the correct circuit using a set of designer-supplied test vectors to realistically exercise it, and traced the state of all flip-flops in each clock cycle to create a golden reference. Next, we artificially broke the circuit to mimic a bug introduced during development. One common mistake that we have personally suffered at the hands of is not fully correcting a line that had been copy-and-pasted:

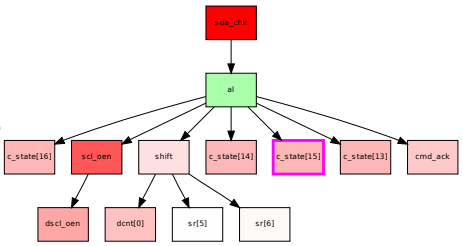
```
        parameter [16:0] wr_b    = 17'b0_0100_0000_0000_0000;
        parameter [16:0] wr_c    = 17'b0_1000_0000_0000_0000;
-       parameter [16:0] wr_d    = 17'b1_0000_0000_0000_0000;
+       parameter [16:0] wr_d    = 17'b0_1000_0000_0000_0000;
```

Here, we have purposefully corrupted the assignment for one state in the state machine: state wr_d is now indistinguishable from state wr_c. Following this, the buggy circuit was re-simulated and the value of each flip-flop recorded as the erroneous trace.

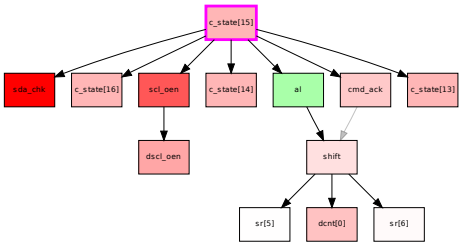
Figures 2a and 2b show how this error propagates through the network from the first moment that c_state[15] from the buggy trace corrupts and starts deviating from the golden reference, until some time later *(this and subsequent figures are best viewed in colour)*. A red node indicates an immediate mismatch from its reference value (an infected node) whereas a blue node indicates a node



(a) Rumour-source estimate visualization



(b) σ_v^* for $v = \texttt{sda\_chk}$



(c) σ_v^* for $v = \texttt{c\_state[15]}$

Figure 3: Rumour-source estimation and predicted infection paths (*with* oracle knowledge)

that had previously deviated, but had now re-synchronized itself with the correct trace. Similar to the SIR epidemic model, this represents a node that had previously been infected but had since recovered; we make the modification that such nodes do not acquire any immunity (but can be detected as having recovered) and are hence susceptible to re-infection (making it closer to an SIS model).

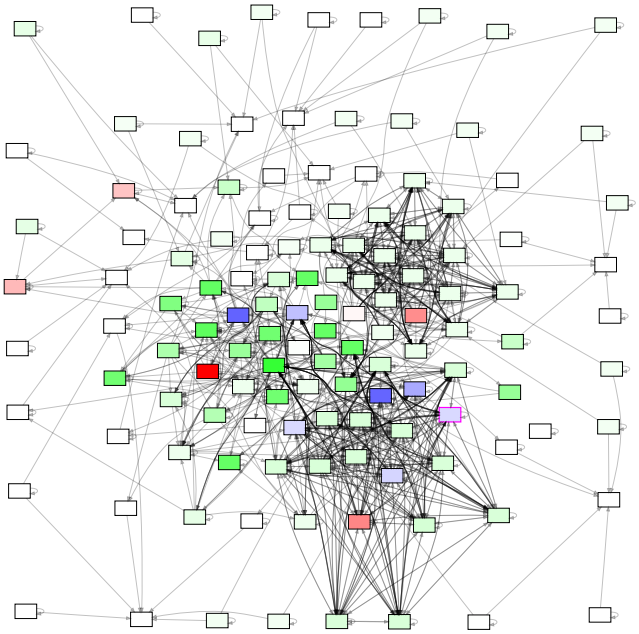
In the first erroneous instance, two bits differ from their correct values: bits 15 and 16 of signal `c_state` (corresponding to states `wr_c` and `wr_d`). For simplicity, we assume that `c_state[15]` is the unique source of the error.

4.1 Rumour-source estimation with oracle knowledge of recovered flip-flops

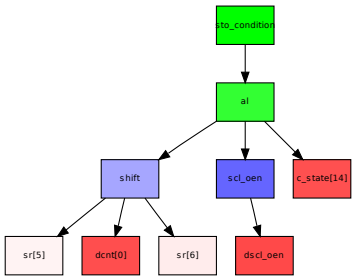
With knowledge of the complete trace of flip-flops over time, it becomes trivial to locate the source of any error by stepping (forwards or backwards) through a buggy trace in order to find the first instance at which it deviates from the reference. We assume that such a full trace is not available, and that the designer only becomes aware that an error has occurred at some time after this first instance. This is a realistic assumption as it may take some amount of time for the error to propagate to an observed node in the circuit (for example, a primary output). However, in this first study we also make the unrealistic assumption that "recovered" nodes (i.e. nodes that were once mismatched with the reference but are now in agreement) can be identified and treated as infected nodes, as doing so requires the complete trace that we supposedly don't have.

Figure 3a visualizes the rumour-source estimation using the error snapshot previously illustrated in Fig. 2b. Nodes that are now coloured in red correspond to both infected and recovered nodes (red and blue nodes from the previous figure). Green nodes indicate suspected nodes. The saturation of each node indicates its estimator value, as normalized to the maximum across all nodes.

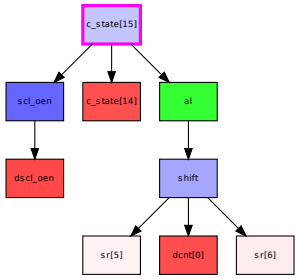
Disappointingly, the estimator performs rather poorly at predicting the source of the error. The results show that the `sda_chk` signal was predicted to be originator, with the predicted breadth-first infection path as shown in Figure 3b. In contrast, the predicted path for the correct source is shown in Fig. 3c, which differs from the actual infection path by only one edge (shown in grey). Upon further investigation, we found that although the rumour centrality values $R(\texttt{sda\_chk}, G_N) < R(\texttt{c\_state[15]}, G_N)$ because of the "deeper" nature of the former, their likelihoods differ by a much bigger factor $P(\sigma_v^2|\texttt{sda\_chk})) \gg P(\sigma_v^2|\texttt{c\_state[15]}))$ due to the significantly smaller node degree of `sda_chk`, leading to a higher probability of transmission as indicated by Eqn. 2. Interestingly, the true source `c_state[15]` was ranked joint-9th of 103, over



(a) Rumour-source estimate visualization



(b) σ_v^* for $v = \texttt{sto\_condition}$



(c) σ_v^* for $v = \texttt{c\_state[15]}$

Figure 4: Rumour-source estimation and predicted infection paths (*without* oracle knowledge)

the 129 nodes present. In the original work, Shah and Zaman measure the performance of their generalized estimator by counting the number of "hops" their estimate was from the true source. We have chosen not to use this measure as we felt it to be meaningless for our application.

4.2 Rumour-source estimation without oracle knowledge

A more realistic scenario would be to remove the previous assumption that oracle knowledge exists to allow "recovered" nodes to be identified. Figure 4 repeats the same experiments as in the previous subsection, but this time with the recovered nodes (blue) treated as being uninfected. The first of four nodes which returns the maximum rumour-source estimation is `sto_condition`, and its infection path predicts that it correctly passes through two recovered nodes, however, these nodes do not include the true source `c_state[15]`. As before, we see the same relationship where although the rumour centrality of the true source is higher than for the predicted source, this is outweighed by their likelihoods. This time around, `c_state[15]` is jointly ranked 24 from 103 estimates.

4.3 Analysis and future directions

A number of reasons exist for why the rumour-source estimator proposed by Shah and Zaman performs so poorly in this application. First and foremost, the prior work assumes that each infected node can transmit its rumour to any neighbouring node with equal probability — this is not true in our application: where it is more a case that "all nodes are equal, but some nodes are more equal than others" — in other words, some nodes are more susceptible to propagating errors depending on the combinational logic gates that exist between flip-flops. In some ways this is also true of real social networks — members of a population trust their connections by differing amounts, whilst some members are more gullible than others. Interesting future work would be to infer those probabilities from the underlying circuit logic, or perhaps through online learning, and to weight the graph appropriately although this would almost certainly impact the computational complexity of the estimator. Performance is also compounded by using a breadth-first search to predict the infection path: in Figure 3c the predicted path is agonizing close to the true infection path (differing by one edge, indicated in grey) which isn't selected only due to this breadth-first policy, in which ties are broken at random.

A second reason for poor results may also be our use of directed, as opposed to undirected, graphs. At first glance, this may appear to simplify the problem, but the existence of cycles in the directed

graph means that there are still a huge number of possibly infection paths to be considered (as almost all nodes in the graph can be reached from any other node). Furthermore, we consider a more realistic SIS (susceptible-infected-susceptible) epidemic model for our application, which allows nodes to recover and become re-infected, as opposed to the simpler SI model where nodes keep their disease forever. Again, this results in further magnifying the search space. Future directions on this front may be to allow multiple error snapshots to be taken into account, which is not unrealistic given that designers can often observe their circuit periodically, but not in real-time, to provide partial (but not oracle) knowledge of recovered nodes. Applying belief propagation techniques may be useful here.

Lastly, the buggy digital circuit that we apply our experiments to is not a "closed" system. Specifically, it continues to interact (unavoidably so) with the external environment through its primary-inputs and -outputs, meaning that errors can propagate from outside of the system. This violates yet another assumption made by the previous work: that there is only one rumour source in the network. Vigilant readers will have noticed that in Figure 2b, there are a total of 14 infected and recovered nodes, whilst the predicted infection paths in Fig. 3 contain only 12 nodes — this is because the two missing nodes are only reachable from (and hence infected by) a primary-input.

Future work would also include applying the rumour-source estimator on more error scenarios, and to investigate with significantly larger (and hence, more sparse) benchmark circuits in which errors may propagate more extensively and less ambiguously. Looking even further into the future, by considering that complete visibility into all flip-flops of a large design is often not feasible, how would an estimator perform when given *incomplete* knowledge of even the infected nodes? Following on from that, an even more interesting (and extremely relevant problem currently plaguing both academia and industry) is: given you can only observe a subset of flip-flops in the circuit, which ones should be selected to maximize the likelihood of estimating the correct source?

5 Conclusions

Due to the rapid progress made by semiconductor technology, modern circuits are integrating more and more functionality into each design, which is becoming harder and harder to verify. Prior work by Shah and Zaman [4, 5] proposed a solution for the compelling problem of how to locate the source of a malicious rumour as it spreads through a network of individuals (or equivalently, locating the originator of a virus in a population epidemic). By first considering the trivial case of estimating the source in an undirected, regular, tree-structured network, the authors constructed a rumour-source estimator which they subsequently proved to give the maximum-likelihood solution. Next, the authors then extended their solution to work with irregular trees and general, undirected networks; although they found that this no longer gave the maximum-likelihood estimate, their estimator was still found to produce convincing results when validated against simulations.

In this paper, we have investigated applying Shah and Zaman's rumour-source estimator onto a network-based representation of an actual digital circuit, using real extracted data, in order to automatically locate the source of a bug. We make the assumption that bugs in digital circuits appear as a corrupted value held by a state element (flip-flop) in the circuit, and that this corrupted bit can propagate onto other flip-flops. In contrast to previous work, our problem differs in that our network is a directed graph (with cycles) where each node in the graph does not have an identical probability of infecting neighbouring nodes, and that there may exist multiple error sources.

Disappointingly, we have found that the original rumour-source estimator does not perform as well when applied to our problem, which we believe is due to the relaxation of the assumptions mentioned above. However, we believe that this method does still show promise, and we have proposed several avenues of future work for adapting the estimator to these new assumptions.

References

[1] Intel. Moore's Law in perspective (Press Kit). `http://download.intel.com/museum/Moores_Law/Printed_Materials/Moores_Law_Perspective.pdf`, March 2005.

[2] UN Foundation. 7 Billion Reasons for mHealth. `http://www.unfoundation.org/news-and-media/press-releases/2011/7-billion-reasons-for-mhealth.html`, October 2011.

[3] Tom's Hardware. The Hidden Cost of Intel's $700 Million SB Recall. `http://www.tomshardware.com/reviews/cougar-point-recall-sata-6gbps,2896.html`, February 2011.

[4] Devavrat Shah and Tauhid Zaman. Rumours in a Network: Who's the Culprit? In *NIPS 2009 Workshop on Analyzing Networks and Learning with Graphs*, December 2009.

[5] Devavrat Shah and Tauhid Zaman. Rumors in a Network: Who's the Culprit? *IEEE Transactions on Information Theory*, 57(8):5163–5181, 2011.