Gmail **vikas rao <vikaskrao25387@gmail.com>**

23rd European Test Symposium - Notification of decision for paper #49

becker@informatik.uni-freiburg.de <becker@informatik.uni-freiburg.de> Tue, Feb 13, 2018 at 9:06 AM
Reply-To: "becker@informatik.uni-freiburg.de" <becker@informatik.uni-freiburg.de>
To: VIKAS KUMAR RAO <vikas.k.rao@utah.edu>
Cc: "utkarshgpt71@gmail.com" <utkarshgpt71@gmail.com>, "iilioaea1@student.gsu.edu" <iilioaea1@student.gsu.edu>, "kalla@ece.utah.edu" <kalla@ece.utah.edu>, "fenescu@gsu.edu" <fenescu@gsu.edu>

This message is very important. Please confirm its reading by clicking on the following link
http://welcome.molesystems.com/tttc/ETS/2018/mailReceipt.php?code=OSWBLDCWQJAIEYLEFVJSIBFXGAXPONJU

Without this confirmation, the server will send this email 3 times.
Should you have any questions, please contact the 23rd European Test Symposium Program Chair (becker@informatik.uni-freiburg.de)
==

Dear Rao Vikas,
We regret to inform you that your paper

Resolving Unknown Component In Finite Field Arithmetic Circuits Using Computer Algebra Methods
by Vikas RAO, Utkarsh GUPTA, Irina ILIOAEA, Priyank KALLA, Florian ENESCU

was not selected for inclusion in the technical program of the 23rd IEEE European Test Symposium (ETS’18).

Please find the reviewers’ comments attached at the bottom of this e-mail. The scores range from a high 5 to a low 1.

This year we received a large number of submissions and the acceptance rate has been 25%.

The ETS’18 Program Committee sincerely hopes that you will continue to support ETS in the future and that you will attend ETS’18 in Bremen.

Thank you again for your interest in ETS’18.

Best regards,

Bernd Becker
ETS’18 Program Chair

=======================

Reviewer #1

Relevance of the paper: 4
Relevance of the topic: 4
Technical merit: 3
Level of innovation: 3
Paper organization and language: 2
Overall score: 2
Comment for the authors:
This paper proposes a new method based on the symbolic computer algebra to resolve unknown gate problem in Galois field multipliers when only a single gate is the target component.
The authors have taken advantage a fully formal method to find the correct polynomial for unknown gate. However, there are many ambiguities in the paper making it hard to follow.
Here are some concerns about the paper:

- As far as there are just four types of gate in the used library, it is possible to exhaustively replace gates in the circuit, and then verify the design using SAT or symbolic computer algebra method. What are the advantages of the proposed method in comparison with the exhaustive approach?

- The process of finding unknown gate in subsection III.A is vague. For example, it is not clear why h'_i should be divided by constant h_i before reducing.

- A complete flow of the proposed method to clearly illustrate different phases is missing in the paper.

- The experimental results in Section IV do not verify the capability of the proposed method in resolving unknown gates for different multipliers. The authors have taken advantage of just one type of Galois field multiplier. Nevertheless, there are other finite field arithmetic circuits which could have been employed as benchmark.

- It is not clear how many possible solutions (P) are generated for the unknown gate problem in different multipliers with different sizes. The authors should report it in the experimental result section for different testcases.

- The authors should clarify the relation between the location of the unknown gate and the run-time of their proposed method. Why is the unknown gate resolving much more time-consuming for the location near to the primary output?

- There are some labels (e.g. #Gates, K, and M) in the caption of Table I which they have not been used in the table.

Reviewer #3

Relevance of the paper: 4
Relevance of the topic: 3
Technical merit: 2
Level of innovation: 3
Paper organization and language: 1
Overall score: 2
Comment for the authors:
Summary:
The submitted paper focuses on debugging of circuits. Given a faulty circuit and either its high level specification or a correct, golden implementation, find a gate to replace the fault such that the circuit matches its specification. The location of the (single) fault must be known before hand.
The authors propose to use symbolic computer algebra with focus on Gröbner Basis (GB) to solve this problem. The function f_i faulty gate is modeled as f_i=x_k+P, where x_k is its output variable. Using GBs, the authors find a substitution for P, such that the circuit implements the correct functionality.

Positive feedback:
The problem is interesting and computer algebra techniques seem to be an appropriate approach to tackle it. The experimental results further support that symbolic computer algebra is the way to go.

Necessary rework:
Even though I did really like the approach proposed in this paper, there are several weaknesses:

- the paper proof reading (preferable by someone, who's native language is English), since the paper is full of grammar mistakes. Especially, words like "the", "a" and "an" are missing.

- the paper also has some structural issues and inconsistencies. For example:
-> Structural: In the paragraph "Ideal and Variety operations" the authors use varieties before they introduce what a variety actually is. For a reader that is unfamiliar with symbolic computer algebra, this will be very hard to follow.
-> Structural: You use an Mastovito multiplier as an example in Section 3, but only explain what it is in Section 4. A reader who is not familiar with Mastovito multipliers might expect an
-> below Eq. 5, the authors suddenly use h_i', without introducing what they are. Above Eq. 5, you state that g denotes the left hand side of the equation. Then you substitute h_i'= P. The resulting equation would be g= h_i'*h_i+h_{i-1}f_{i-1}+.... You never defined what h_i' and the like is. This is a major issue.
-> The next part is also not clear: You say that you do some reduction of h_i', if h_i is constant. However, why would h_i being constant be an issue? Applying your methodology to a circuit consisting of a single gate or would result in P=y-a*b (with y being the output and a and b being the inputs) and thus h_i being equal to 1. I don't see any issues here. It is just not clear what you do. A running example would be nice (maybe just a simple half adder, instead of a multiplier).
-> you use the term "elimination term order" without ever defining it. You refer to II.2 (which could either be Theorem II.2

or Definition II.2, be more precise here). However, this links to the Elimination Theorem, where only an Elimination Ideal is defined. What is a "elimination term order"?
-> In your example with the Mastrovito multiplier, it is not clear why you actually need to consider the variables A,B and Z, if they are just a linear combination of a0,a1, b0 and b1. Why not just substitute them for the sake of simplicity? They are not part of the circuit.Specially, the varibale order is strange. If A is only dependent on a0 and a1, how can it be of higher order than z0 and z1?
-> In the experiments, you state that an MR block computes A*B*R^{-1}. From this formulation, I'd expect an MR block to have 3 inputs (A,B,R). In Figure 2 however, the number of inputs seems to be 2.
-> After denote the MR blocks as A,B,C,D, you never reference to these expressions.

- There are some notational issues in this paper: For example, you do introduce RTTO in Section 2. However throughout the text, you denote it as RTTO>. What is ">"? Do you want to emphasize that it is a monomial order?

- It is not clear, why the additional ideal J0 is needed. Specially not, why it must consist of all variables. If I understand the problem definition correctly, the gate is not placed in the wrong location, but only the gate function is chosen wrong. Is that correct? If that is not the case, then you should somehow emphasize this in the text. But if this is really not the case, then the RTTO might actually be wrong. Can this be an issue?

Conclusion:
While I like the considered problem and the approach to solve it, I have to recommend to reject this paper. The explanation of the approach needs significant improvements and the overall language and structure of the paper requires an intensive revision. Also, and this is a major point, you repeatedly use expressions without defining them.

Reviewer #4

Relevance of the paper: 3
Relevance of the topic: 4
Technical merit: 2
Level of innovation: 3
Paper organization and language: 3
Overall score: 2
Comment for the authors:
This paper addresses the debugging problem for galois field multipliers as they are heavily used for cryptographic applications. In previous work the authors have put attention to the equivalence verification problem on the same class of circuit designs. Computer algebra (CA) techniques can take advantage of the underlying arithmetic nature of the verification problems. Together with clever term orderings derived from circuit topologies CA techniques have shown useful in attacking these problems. In equivalence verification for galois field multipliers circuits of realistic bit-width of 128 bits and more could be handled for the first time.

In this paper, the authors try to extend their work towards automatic correction or completion of partial designs. This lifts the overall problem from a SAT-problem towards a QBF-problem increasing its worst case complexity from NP to PSPACE. This is reflected by the experimental results of the paper which indicate that the presented approach only marginally outperforms existing SAT based techniques. In fact, the proposed approach can solve 13 bit instances rather than 12 bit instances solved by previous work. This still is at least one order of magnitude too small to be of practical relevance.

The presentation of the key algorithm is rather weak. The authors spend considerable ammount of space to recap state of the art terminology and algorithms from CA (section II). The space occupied by this well-known content is then missing to elaborate more precisely on the new contribution and the techniques introduced in this paper.

Although I consider myself an expert in this field, I had a hard time working through the two pages of the main section (III), in order to get a glimpse of the underlying idea. This is mainly due to the fact that an algorithmic overview on the new algorithm in form of some pseudo-code is missing. Instead the reader is confronted with a lengthy example and needs to reverse-engineer the underlying algorithm form the steps taken in the example, which include long mechanical reductions of polynomials.

If I correctly understood the approach, the key idea is to change the term order into an elimination term order. This imposes the need to actually compute a Gröbner-Basis rather than deriving it for free from the topology of the circuit. The paper does not provide any analysis on the cost of these computations but I would guess that this is the reason why the presented approach does not scale as nicely to higher bit widths as the previous work on equivalence verficiation.

In summary, there may be some interesting ideas in this paper but the paper would need major revision before it can be published in a high rated conference like ETS.

Reviewer #5

Relevance of the paper: 3
Relevance of the topic: 4
Technical merit: 4
Level of innovation: 4
Paper organization and language: 4
Overall score: 4
Comment for the authors:
The paper looks into the interesting question of error correction
in arithmetic circuits. Most existing approaches to error correction
are based on SAT and SAT is not very effective for arithmetic circuits.
The currently most efficient methods for verification of arithmetic circuits
are based on computer algebra instead of SAT. So it seems to be reasonable
to look into computer algebra methods for error correction of arithmetic
circuits as well. The paper presents an error correction method in the following
setting: A location for the error is already known and the task is to find an
alternative implementation that - inserted at this error location - corrects
to error. The error correction circuit is not restricted to a single gate, but
may be an arbitrary circuit described by a polynomial.
As arithmetic circuits the paper considers finite field multipliers.

The approach is makes use of the theory of Groebner bases of multivariate polynomial rings
F_q[x_1, ..., x_n] that are based on a finite field F_q.
For the error location at a signal x_k an (unknown) polynomial
P is computed such that the specification is fulfilled. Roughly speaking, this question
is reduced to the question whether the specification polynomial f is included in the ideal with all
gate polynomials (including x_k + P) as a Groebner base (which is equivalent to the condition
that the specification polynomial f vanishes when reduced by the gate polynomials
using polynomial division).

The paper gives a good review of the underlying maths and presents a sound theory
of the approach. The critical issue is applicability in practice.
Unfortunately, the computed solution polynomial P is not unique. Usually there are many
possible solution polynomials and not all of them are suitable for implementation.
The paper discusses this issue only in part.
The paper discusses only the case that the solution polynomial is based on signals
that are not in the fan-in cone of the error signal x_k and discusses how to fix this issue by changing the
topological order. This case is not too dangeroeus however, since the approach presented in the paper
guarantees that the signals on which P depends are before x_k in a topological order of the circuit,
i.e., no circular dependencies are introduced.
What is even worse though: All comptations are performed in the polynomial ring F_{2^m}[x_1, ..., x_n]
with finite field F_{2^m} (or GF[2^m]). Therefore coefficients of the solution polynomials P
may be in GF[2^m] instead of GF[2].
What is needed from a practical point of view is not an arbitrary solution polynomial
P, but a simple one that is easily implementable by a circuit. Thus it would be desirable that
both the variables in P should be GF[2]-variables and the coefficients should be also in GF[2].
See for instance h_2'' in the example on p. 5 which is a valid solution, but with coefficients from
GF[2^2].
Thus, one has to look in the complete solution space for solution polynomials and select simple solutions.
The paper does not provide a systematic, goal-driven search method for simple solution polynomials implementable
by a simple circuit.

The paper only discusses some special cases where a simpler solution polynomial can be found
(case h_2 is constant on p. 5, col. 1 (or in general h_i constant on p. 4, col. 1)) in which
an alternative simpler solution can be found. The paper also discusses a preliminary method
to generate larger *sets* of solution polynomials based on quotients of ideals, but this method
does not seem to generate all possible solution polynomials, but many trivial solutions based
on a single solution, see e.g. the example on p.5: Instead of the solution
a_1 * b_1 one can also use a_1 * b_1 exor (s_2 exor a_1 * b_0) e.g.. This is trivial however,
since the signal s_2 is exactly computed by a_1 * b_0, i.e, s_2 exor a_1 * b_0 = 0.

Experimental results: As expected, the experimental resuls show run-time improvements compared to
a SAT-based error correction approach, but they neglect the question of easy implementability.

Altogether, the paper gives an interesting step into the right direction, presents a sound
theory nicely illustrated by an example, but it is still preliminary in the sense that the
computed correction polynomials are not guaranteed to be (easily) implementable.
It is not clear how to fix this issue easily.

Detailed comments to the authors:

- Please check your description of the solution process of the 2-level QBF from [4] (p.1, col. 2).
Your description is wrong whereas the original description in [4] is correct.

- p.2, col. 2:
"A Groebner basis (GB) G of ideal J is one such
set of polynomials G = GB(J) = ... that is a
canonical representation of the ideal."
This statement is wrong. Groebner bases do not
have to be canonical (see also reduced vs. non-reduced Groebner bases ...).

- p.2, col.1: "Farimah" --> "Farahmandi"

- p.2, col.1: "R = F_2[x_1, ..., x_n]" --> "R = F_q[x_1, ..., x_n]"?

- p. 4, col. 1: Here and in the following the notion is a little bit strange.
H is not an "arbitrary element from F_q", but a vector of elements (one for each (x_l^q - x_l)).
x_l^q - x_l also implicitly represents a vector of polynomials.

- p.4, col. 1: "We will also have cases, when h_i ends up being a constant, in which case lift returns g itself as a solution h'_i."
This formulation is a little bit strange (same also one page later in the example).
A correct solution polynomial is g * h_i^{-1} in this case, but there are simpler alternative
solutions which you can compute as given later (reducing g * h_i^{-1} by f_{i-1}, ..., f_1).
This should be the important message, not the technical information that the solution returned
by lift has to be divided by a constant in order to be correct.

- p.4, col. 2: "alpha is the root of primitive polynomial s.t. P(alpha) = 0."
Wouldn't it be simpler to write: "alpha is the constant in GF[2^2] that
corresponds to the polynomial X"?

- p.5, col. 1: In fp_3, ..., fp_7 the quotient-h_i's in [] and the divisor-f_i's in ()
have to be exchanged, e.g., [1](z_0 + s_0 + s_3) instead of [z_0 + s_0 + s_3](1) for fp_3.

- p.5, col. 1: "a implementable solution" --> "an implementable solution"

- p.5, col. 2:
"a1 * b1 - P' in <f_3, f_4, x_l^q - x_l> : h2;" instead of "a1 * b1 - P' in {f_3, f_4, x_l^q - x_l} : h2;"

- Sect. III.B. seems to be somewhat unsound.
Why do you need a miter with an XOR gate?
Why is the specification polynomial not simply x_n - y_m, i.e., the fact that the circuit polynomials
are 0 implies that the specification polynomial x_n - y_m = 0?
What does "f : t * (x_n - y_m) where, t is the final output of miter gate" mean here?

Reviewer #6

Relevance of the paper: 3
Relevance of the topic: 2
Technical merit: 3
Level of innovation: 3
Paper organization and language: 2
Overall score: 2

Comment for the authors:

Summary

The authors consider the problem of synthesizing a single unknown component in a digital circuit. They focus on arithmetic circuits and propose to apply methods from computational algebra to solve this problem. First results seem promising - compared to the approach [4] using incremental SAT-solving, it seems more efficient.

Evaluation

- The paper definitely needs more polishing before publication - there are many linguistic and typographic issues and minor errors.

- I think the paper is slighty off topic for the European Test Symposium. There are certainly conferences that will appreciate works in computational algebra more than ETS.

- The motivation of the paper - fixing a single gate in an arithmetic circuit which has accidently been replaced by a different gate - seems not really relevant. Please provide more information on this issue.

- The authors seem to be unaware of the works on black box model checking using QBF and DQBF solving. For example see:
  * Scholl, Becker: Checking Equivalence for Partial Implementations. DAC 2001
  * Gitina, Reimer, Sauer, Wimmer, Scholl, Becker: Equivalence checking of partial designs using dependency quantified Boolean formulae. ICCD 2013

- The method seems restricted to modules with a single output. Can it be extended to multiple outputs? Can you take into account the information which signals are read by the unknown module/gate? If you known that a certain gate is wrong, you typically know which signals are read by the module/gate.

- The formulation of definitions is sometimes a bit strange. Consider on page 3:

  Definition II.2. Reverse topological term order [14]:
  The computational complexity of Buchberger's algorithm is exponential in the number of variables n. As our work is focused on the circuits, we will describe a term order that renders the set of polynomials for the gates of the circuit, a Groebner basis itself. This term order is called Reverse Topological Term Order (RTTO).

  I would not consider this a definition! (A more reasonable definition follows without number).

- The core part of the paper in Sect. III is hard to follow. For me it was not clear how to obtain the polynomial h_i if P is unknown. We have that fin J+J_0 if there are P, h_1,... such that f = h_1f_1 + h_2f_2 + ...
  The transformations are clear, but you can only check if
  f - h_sf_s - ... - h_ix_k in <h_i,f_{i-1},...,f_1,x_l^q-x_l...> if you know h_i.

- The experimental evaluation should be done more carefully, e.g., the authors do not provide any information regarding the size (number of gates) of the considered circuits.

In summary, the paper describes a nice idea with (as far as one can judge) promising results, but the presentation definitely needs substantial improvement. In the present form, I would not accept the paper for publication. Therefore I vote with "weak reject".

Detailed comments

- p1, title: "Component" -> "Components"
- p1, left, line 1 of Introduction:
  "gate level" -> "gate-level"
- There should always be one space before references, between a
  word and the acronym in braces like in "Binary Decision Diagrams(BDDs)[1]"
  (p1, left, first paragraph). Please apply this throughout the paper.
- There should never be a linebreak immediately before a comma.
- p1, left, 1st paragraph: "Due to _the_ inherent algebraic nature ..."
- p1, left, 2nd paragraph, 1st line:
  "Within _a_ symbolic algebra environment ..."
- p1, right, "A. Previous work": many spaces missing
- p1, right, 1st paragraph of A
  "The solution to these variables _implements_ the desired logic function"
- p1, right, enumeration (1): "num-ber" (hyphen is missing)
- p1, right, enumeration (1): "initialized to _the_ empty set"
- p1, right, last paragraph: "Despite using state-of-the-art SAT solvers"
- p2, left, 1st word: "Techniques" -> "The technique"
- p2, left, 1st line: "et al." (the point is after "al." not after "et")
- p2, left, Sect. I.B, line 2f:
  "only one gate in the design incorrectly replaced" - weird sentence
- p2, left, "using Groebner basis based guided ideal membership testing, and elimination ideal"
  weird sentence
- p2, left, "approach to resolve _an_ unknown component"
- p2, left, last line before Sect. II
  "_a_ comparison to the _SAT-based_ approach"
- p2, left, Sect. II
  "is some exponent" -> "is some power"
  "R = F_2[x_1,...,x_n]" -> "R = F_q[x_1,...,x_n]"
- p2, right, line 3: "e.g." is strange in that sentence
- p2, right, insert before formula (2):
  "... such that the roots of the polynomials correspond to consistent assignements
  of the gate's signals" (or similar).
- p2, right, Algorithm 1, line 11: (u_1,...,u_s) should be a vector as the
  order of the elements matters and because you need to preserve duplicate values.
  The same applies to the text (paragraph below the algorithm)
- p2, right, "Polynomial Ideals:"
  {f_1,...,f_s}subseteq F_q[x_1,...,x_n]
- p2, right, Def. II.1, last line on page 2:
  iin{1,ldots,t} (instead of cdots)
  (same on p3, left, line 5)
- p2/3: on p2, right, you say: "A Groebner basis is one such set [...]
  that is a canonical representation of the ideal",
  on p3, left, only a reduced Groebner basis is a canonical representation. Please clarify this.
- p3, left, "Ideal and Variety operations"
  You should provide the definition of a variety before you define operations
  on varieties.
- p4, left, below Eq. (4):
  H is not a single polynomial, but we need one for each vanishing polynomial x_l^q - x_l for 1 <= l <= n.
- For multiplication, please use cdot instead of *.
- p4, right, below caption of Fig. 1: "2x2 Mastrovito multiplier" -> "$2times 2$ Mastrovito multiplier"
- p4, right, 1) Field construction:
  It is clear what F_4 is, and also what F_2[X] is. But what is F_4 = F_2[X] mod P?
  What is a primitive polynomial? It would be useful to say in the example in Sect. III.A which function
  a Matrovito multiplier computes and not only later in Sect. IV.A ...

[I skipped most of the typos, imprecisions etc. after page 2; there are just too many of them.]