First thank you for your hard work in reviewing this paper. I find the review comments are very helpful because they show me many corner cases not considered before. I have taken these comments into consideration, and polish my paper significantly.

Almost all reviewers say that this paper miss many details about the previous work, on which this paper may depend on. I agree on this, but I need to give a little bit explanation. This is the first time for me to write for TODAES, whose latex template seems to be very sparse. The same content that covers only 10 pages or so in IEEE double column may cover almost 20 pages in TODAES. This makes me a little bit nervous and eager to reduce the page number. So some details of previous work are unfortunately omitted. And now they have been filled back in. Hope this ease your job.

I will first summarize the major modifications to my paper here. And then reply to your comments one by one.

My major modification are:
1. Redraw the Figure 1 and rewrite the introduction section to make it easier to be understood
2. Rewrite the subsection2.4 to introduce our previous work on halting algorithm to determine whether an input can be uniquely determined. More structure, intuitive explanation and figures are given.
3. Subsection 3.2 is added to describe new algorithm based on incremental SAT to speed up identifying flow control vector.
4. Add more subsubsection structures, figures and intuitive explanations into subsection 4.2 and 4.3, to make them easier to be understood.
5. Add section 5 to describe how to minimize p, l and r to reduce decoder area.
6. Add more benchmarks from our previous paper and other researcher's paper in section 7.
7. Add subsection 7.5 – 7.8 to compare between different algorithms .
8. Add subsection 8.4 and 8.5 to describe related works on Satisfying Assignments Enumeration and Logic synthesis with Craig interpolation.
9. Polish the references.

In the remainder of this file, I will reply to your comments. I will show your original comments in black color, and my response in **blue color and bold font**. All the modifications on my manuscript are marked with **bold font**.

---------- Reviewer Comments ----------

Referee: 1

Comments to the Author
This paper extends prior work on complementary synthesis by proposing a way of synthesizing decoders for encoders involving some flow control mechanism. The synthesis procedure consists of three steps: 1) obtaining the flow control variables, 2) finding the conditions when data variables can be inferred from the flow control variables, and 3) synthesizing the decoding functions for the flow control and data variables. Experiments are performed on three industrial standards.

The presented method seems to be technically sound, and enhances the practicality of decoder synthesis for industrial applications. However, the presentation requires further revision.

Detailed comments:

P2. The statement, “such an encoder with flow control mechanism may visit the non-unique state set infinitely often, that is, its input variables i cannot always be uniquely determined by a bounded sequence of o” is confusing. Change “that is” to “and therefore”.
P2. Please define the “unique state set”.
RESPONSE: I find concepts of “the unique state set” and “the Non-unique state set” are unnecessary here, I replace Figure 1 with a new figure showing the structure of a high speed communication system with flow control mechanism. I also rewrite the 1st to 6th paragraphs of Page 2 to explain the flow control mechanism and how will we handle it.

P2. In Fig. 1 b), please explain the “predefined special control symbol”. In your computation, how is the predefined special control symbol determined?
RESPONSE: I am sorry that I have use two different terms in this paper for the same thing , “predefined special control symbol” and “the invalid data”. Now I have replace all of them by a new term “idle symbol” to express that such a symbol can be safely discard by the decoder. I also rewrite the 1st to 6th paragraphs of Page 2 to explain the flow control mechanism and how will we handle it.

In Section 2.2, please specific how initial states are treated, and explain its connection to the parameter “p”.
RESPONSE: Thank you for your suggestion.
Because of structure adjust, it is now in subsection 2.4.
Actually we ignore the initial state of the encoder, and use p to propagate the constraint on input variables into the encoder's state variables. In this way, we can avoid the need to compute reachable state set, and thus significantly speed up our algorithm.
I have rewritten the last three paragraphs of page 5 and add Figure 2 to explain F_{PC} in more details.
Furthermore, I add the first four paragraphs in page 6 to explain the advantages and disadvantages to ignore initial state.

How does interpolation help to characterize the set of assignments that satisfy a Boolean relation (Algorithm 3 CharaterizingFormulaSAT), say, in comparison to enumerating all satisfying assignments? Please assess the usefulness of interpolation through experiments.
RESPONSE: Thank you for your suggestion. I think it is a widely accepted today that interpolation is much more efficient than simply enumerating all assignments. And in section 4.1 paragraph 3, I have also stated that the complexity of simply enumerating is exponent. So I think the reader may not be interested in reading again how interpolation defeat simply enumerating.

Moreover, it would be interesting to compare the applied interpolant generation (from proofs) method to the following method without proofs.
Hana Chockler, Alexander Ivrii, Arie Matsliah: Computing Interpolants without Proofs. Haifa Verification Conference 2012: 72-85
RESPONSE: Thanks you for your suggestion. I read Hana's paper carefully and find that its ideas are:
1) Enumerating assignments of formula A in outer loop.
2) And then enlarging each such assignment in inner loop by dropping literals.
A similar approach have been tried in my first paper about complementary synthesis “Synthesizing complementary circuits automatically”appear in ICCAD09, which enumerate satisfying assignment by enumerating assignments and trying to drop literals by testing unsatisfiability.
The experimental results in both Hana's and my ICCAD09 paper indicate that such an

approach can not beat Craig interpolation from proof.
I add subsection 8.4 to discuss varies methods to enumerate satisfying assignments, including Hana's paper.
I am not sure what to do other than this, can you give more suggestion?

P7. "…, which covers and only covers …" is not followed by the proposed computation. Essentially interpolation makes the computed set only an over-approximation.
RESPONSE: Yes, you are right, interpolation of unsatisfiable formula A∧ B covers only an over-approximation of A. But in the special case that A and B are complement to each other, that is, every assignment must satisfy either A or B but not both, then this interpolation must cover and only cover A.
We have explained this in the first paragraph of P11. It first says that the interpolation cover an over-approximation. And then it says: "At the same time, IT P (a) ∧ R(a, A(b), 0) is unsatisfiable, so IT P (a) covers nothing that can make R(a, A(b), 0) satisfiable. Thus,IT P (a) covers exactly the set of valuations of a that can make R(a, A(b), 1) satisfiable.".

P8. The statement "ITP(a) covers exactly the set of valuations of a that can make R(a, A(b), 1) satisfiable" is not accurate. ITP is an over-approximation and may not be exactly the set.
RESPONSE: please refer to above response.

"Mcmillian" -> "McMillan"
RESPONSE: corrected, thank you.

In equations (5) and (7) of Section 4.2, is "=" different from "\equiv"? Please clarify.
RESPONSE: I am sorry that I have not capture your meaning, so I try to explain all the symbols related to equivalent here, and only modify my paper after I got your reply in next review.
For "=" with "def" over it, it means defining a new variable on its left to be an formula on its right.
For "=" after conjunction, it means enumerating a list of values, and conjunct all the resulting equation on its right.
For "\equiv", it is a predicate that means the formula on its left is equal to the one on its right.
As we have already used ":=" to represent assignment, so I have replace all "=" with "def" over it to ":=".

In Lemma 4.3, the implication "F'pc(p',l',r',1) => F'pc(p,l,r,1)" needs further elaboration.
RESPONSE: Thank you for your suggestion. This statement is not accurate. I have change the proof a little bit.

In Algorithms 2 and 4, the parameters p, l, and r are incremented simultaneously. It might seem too conservative and make formulas unnecessarily complex. There seems room for improvement. It would be nice to compare different strategies and their effects on the quality of synthesized decoders.
RESPONSE: Yes, you are right.
Actually algorithm 1 "Section 2 preliminaries" come from my TCAD 11 paper "A Halting Algorithm to Determine the Existence of the Decoder". In that paper, they actually include an additional step to reduce the value of p, l and r. But including all these detail here may make the "Section 2 preliminaries" too long. So in my previous version of this TODAES paper, I chose to omit these details and refer the readers to my TCAD 11 paper at the end of "Section 2 preliminaries".
But since you have asked, I add a new section 5 to reduce the value of p,l and r.

In Sections 5.1 and 5.2, the way of using interpolation for logic synthesis is not new. Citation to prior works would be helpful to contrast the relative novelty of the proposed method.
RESPONSE: Sure. I add the Subsection 8.5 to talk about this.

In the experimental results, what are the values of the corresponding parameters p, l, and r?
RESPONSE: I have all p,l and r to subsection 7.2 7.3 7.4 7.6 and 7.7.

P14. “the way it handle…” -> “the way it handles”
RESPONSE: thank you, I have corrected it.

The references are not presented in a consistent style. Please clean them up.
RESPONSE: Thanks. corrected

Referee: 2

Comments to the Author
Synthesizing decoder from an encoder with a flow control is non-trivial.
The flow control makes it hard to infer input variables uniquely from a bounded sequence of output variables, as it invalidates data into the encoder's input sequence.
This paper discuss how to synthesis such a decoder from given encoder with flow control.
It identifies the flow control variables, the predicate over these variables controlling the validity of the data, and decoder function over the input variables.

Identification of flow control variables are done through iterative unrolling over the bounded sequence of outputs in a unique state path. It assumes that flow control variables can be uniquely identified from input variables.
The predicate over the flow control variable is found using interpolation technique to quantify b variables from a Boolean function R(a,b,1) where R(a,b,1) \and R(a,b,0) is unsat for all values of a,b.

Experimental results on real designs show the efficacy of the approach.
The reported runtimes were less than a minute.

Overall, the paper is bit notation heavy, and often use of symbols are done without giving adequate explanation. For example, before detailing the equations for FSAT_PC and FSAT_LN in (page 9), authors should give intuition behind the approach. I have to read some sections at least more than once to get the complete intentions.
RESPONSE: you are right, I have add the first and second paragraphs of Subsection 4.2 to introduce the intuition behind this complex algorithm.
Besides that, I also partition this subsection into subsubsection 4.2.1 and 4.2.2, each describe FSAT_PC and FSAT_LN respectively.

Besides that, I also have some doubts about the proposed solutions.
Please try to give explanation as needed.

1. Algo 1 checks sequences starting from initial state. Is this state a fixed state or arb. state?
RESPONSE: Actually it is an arbitrary state. I have discussed the advantage of such a choice in the third paragraph of page 6.

If the sequence is unsat at depth some (p,l,r), can it be satisfiable at higher depth? If so, then I think your decode logic may be wrong as it may be satisfiable at higher depth.
RESPONSE: Actually, when the sequence is unsat at some (p,l,r), then it will always be unsat for larger (p,l,r). Informally, if we can uniquely determine the input with a sequence of output, then we can certainly determine it with longer output.
I have explained this in the last two paragraphs at page 5 and the first two paragraphs at page 6.

Note, you only found a bounded proof from initial state, and not an unbounded proof. If there is some assumption made, please state it upfront.
RESPONSE: Yes, the proof we found from the unsat of F_PC and sat of F_LN are all bounded. But they can be generalize to unbounded case. Please refer to the last two paragraphs at page 5 and the first two paragraphs at page 6 for PC case, and the 3rd and 4th paragraphs for the LN case.

2. Algo 2 can also include non-control input variables (i.e., d variables) in the set returned at line 13. How do you ensure the set contains only and all flow control variables? (are you assuming that all and only flow control variables will have this property of uniqueness? If so, please state that upfront.)
RESPONSE: Yes, it can recognize some data variables as flow control variables. But that do not hurt our general framework. Because the decoder's Boolean function can still be correctly characterized.
I have add a new paragraph at the end of Subsection 3.1 to explain this.

3. Algo 3: typo at line 2. R(a,b,t) should be R(a,b,1).
RESPONSE: Wow, that is a bug hidden so deep. Thank you very much. I have corrected it.

4. Eqn 5: how do you find d? So far in algo 2, you only found f, but it is not clear whether it includes d or not? It is not clear if you found all f in algo 2?
RESPONSE: It is my mistake that have not say about this. Actually d=i-f. That is, all input variables are either f or d. After getting the full list of f, we take all left i as d.
I have also modified Algoritm 2 and 3 to find f and d at the same time.

5. Experiments. Is there way to estimate the quality of the synthesis, by comparing with an existing decoder?
RESPONSE: Sure. I have add Subsection 7.8 to compare the area of decoders synthesized by our algorithm and that of manually designed decoders.

Referee: 3

Comments to the Author
Dear Authors,
The topic discussed is very relevant to TODAES and with improvements should make for an interesting solid work that enables automation of practical processes.
However, in the way that the work is presented, it is either not explained well enough or the work was not rigor enough to convince that the presented work is all that is claimed to be - i.e. correct by construction algorithms.
RESPONSE: Thank you for your hard work. As I have mentioned at the head of this documentation, This is the first time for me to write for TODAES, whose latex template is so sparse that the same content cover about 10 pages in IEEE double column now cover more than 20 pages in TODAES. So I am a little bit eager to reduce the page number by omitting

something that I think unimportant. But obviously this have significantly complicate the reviewing, so I have filled them back in. Hope this time it is easier for reviewing.

I am attaching a list of concerns that caught my attention. Please note that I started with a detailed review, but once I lost faith in the validity of the arguments I did not review the remainder of the paper at the same level of detail, hoping that the paper will go over a major review. Please take the spirit of the comments made and apply it to the later sections of the paper.
RESPONSE: Thank you for your hard work again. Of course I will rewrite all contents that I think may not be clear enough. I am still looking forward to your valuable suggestion again from your next review.

Major points for consideration:

The problem statement is not clear. The objectives are not clear. It is also not clear what the inputs of the problem are and what the expected outputs are.
RESPONSE: I am sorry that I have not describe this paper's purpose clear. I have restructure the first paragraph of section 1. Actually we are trying to generate the decoder automatically from the encoder.

Are we designing a decoder? If so, what do we know of the specifications of the encoder?
RESPONSE: Yes, we are designing a decoder by automatically generating it from the specification of an encoder. We have many ways to know the specification of the encoder, verilog or vhdl. But our paper is not limited by this. I have restructure the first paragraph of section 1 to talk about this.

It is not clear why we are trying to estimate functions of the encoder that one may assume that they are known.
Taking into account that this is likely to be mostly an issue with presentation more than anything else that can be resolved with a proper introduction to the proposed solution.
RESPONSE: I am sorry that I don't understand the meaning of " to estimate ...". Can you please explain that?

Many explanations are missing, such as the idea of the unique and non unique state sets that are used but not explained. The author explained these concepts in other papers but there are no references to these explanations.
RESPONSE: Actually the concepts of unique and non-unique state set are only be used once in our FMCAD 10 paper, but not after. Actually these two concepts are not accurate and sometimes misleading. So I remove them and rewrite Section 1.

The approach taken in the solution draws from formal methods but does not appear to be solid, or explanations are missing to show that it is. A few of examples that illustrate this issue:
o In Equation 1 F PC (p,l,r) is defined as a conjunction of a few terms, one of which is $i \neq i'$, and a conclusion is made that if the conjunction is unsatisfiable then $i = i'$.
Logically, there are other ways for this conjunction to fail. It may very well be that they all lead to the same conclusion, but this is not discussed or even mentioned.
RESPONSE: You are right, I add the 4 th paragraph in page 5 to explain this.

o In Equation 2, on the other hand another conjunction is presented, and this time the conclusion is derived on the case that the conjunction is satisfiable. It is not clear, presented or discussed why the conclusion here is different.

RESPONSE: I am sorry that I have not present a clear structure for Subsection 2.3. Now I have change it into a clear structure.
According to the last paragraph of page 4, Equation (1) is an under-approximate approach to determine whether an input variable can be uniquely determined, while Equation 2 is an over-approximate approach.
That is, when Equation 1 say YES, then the final answer is YES. When Equation 2 says NO, then the final answer is NO.
These two approaches are presented in Subsubsection 2.4.1 and 2.4.2. And we show in Subsubsection 2.4.3 that they will eventually converge and give conclusive answer.

o Equation 2, the idea of loops is also presented though not discussed. It is not clear why the loops are divided into three different sections or why the definition is such that they need to appear in the same places along the two paths.
RESPONSE: I highlight one paragraph in the end of Subsection 2.4.2 which starts with "More importantly...". Basically, these three loops can be unrolled to generalize the proof of not uniquely determining to all larger valuation of p, l and r.

o Algorithm one assumes that p, l and r all have the same values, probably for simplicity and without a significant effect, but the difference from the definitions and equations that do not require this property is not mentioned.
RESPONSE: Actually other reviewer also have notice this problem. We always have another step to reduce these values. Please refer to Section 5.
This step is too simple and is repeatedly mentioned in all my previous paper, so I think it is better to omit it to shorten this paper, because it already contains almost 20 pages. I have never written so long a paper.

Such issues make the validity of the arguments very weak. I would suggest that either a full formal approach is taken, with all the laborious work of getting that right, but with the reward of the robustness of the solution, or an engineering approach can be taken that is weaker but easier to adopt. The mix created an illusion of robustness that is not scientific.
RESPONSE: You are right, the current approach is not in good sharp, from both the formal and engineering aspect. I will try my best to improve both the intuitive and formal presentation. You may have notice that all complex contents now start with intuitive introduction.

In the experimental results, the authors mention that they are not applying their algorithms to the benchmark of another group because it does not contain control flow mechanism. If I understand this correctly, the simpler problem still falls within the scope of this work. Hence choosing not to use it for experimentation is not justified. At least some comparison to the works of others should be presented.
RESPONSE: Thank you for your suggestion.
I have add subsection 7.5 to compare my algorithm with another one on all encoders without flow control mechanism.
I also add subsection 7.8 to compare the decoder generated by us to manually written decoders.

Minor points for consideration:

Use of Wikipedia reference is not acceptable. The referencing system is built on accountability for information whereas Wikipedia is the opposite.
RESPONSE: thank you, I have replaced them with reference to books and articles.

“Determining that the design of the encoder-decoder is one of the most difficult jobs” – it’s a big statement. Either provide motivation for it, cite others that explain this, or don’t make a big statement.
RESPONSE: Thank you for your suggestion. May be change to “one of the most important and error-prone job”?

Level of explanations is not consistent: on the one hand the authors bother to explain the mining of the ∈ sign as well as other standard Set theory definitions that are taught at an undergraduate level. On the other hand some more complicated and relevant matters are not explained at all. Such as the meaning of the states and transitions in Figure 1a, and the meaning of “path 2” visiting states.
RESPONSE: this figure and its path are unnecessarily and now removed.

Line 21 on page 2: “this algorithm assumes...” which algorithm is this referring to?
RESPONSE: Actually it should be “all these algorithms assume..”. But now I have rewritten this paragraph.

Line 49 there: “none of the current algorithms” – the specific algorithms under consideration should be listed. It is (in theory) possible that the authors are not aware of ALL current algorithms.
RESPONSE: I have listed them in the 6th paragraph of page 2.

Lines 50-52 thee: this may be a terminology issue, but there appears to be a contradiction in the description. Either the algorithms assume that something is not possible, or they ensure that it doesn’t happen. If they assume then there is no need to ensure.
RESPONSE: Actually these algorithms assume that the encoder's input should always be uniquely determined. But the encoders always contain some redundant logic that fail this assumption, not only flow control mechanism in this paper. So before applying all these algorithms , the user need to write an assertion to disable these logic, and this assertion should be applied to all steps. I have add two entries to the end of Equation 1 to describe this assertion.

What does ‘pc’ stand for in F PC ? (the same question applies to other names used in the paper)
RESPONSE: “pc” stands for parameterized complementary. And “ln” stands for loop non-complementary. I have add these to the paragraphs below Equation 1 and 2.

The perspective of equation 1 is not clear. From a decoder’s point of view the o’s are known and the i’s are not. Yet the equation assumes that we know what i is. Perhaps it should be explained first where the i's are coming from.
RESPONSE: We don't assume whether i or o are known. We just constrain that the two sequences of o are equal, while the two i are not.

Page 5 line 9 - Equation 1 is not explained well enough, “line 2 is a copy of it” – why is there a copy of it?
RESPONSE: I have redrawn Figure 2 to show Equation 1 intuitively. These two paths are all shown in it. And I also explain in the paragraph below Equation 1 that “while Line 2 corresponds to the right path in Figure 2. These two paths are of the same length”

Algorithm 1: perhaps this is a matter of terminology, but it is not clear what the inputs of the algorithm are, or why it is designed in this manner. It appears that the actual inputs to the algorithm should be the o's, or perhaps o's and i's, but not only i.
RESPONSE: This algorithm is called for every input variable i. So it is run several times with different i. So to emphasis this fact, I use i as the only input. For output variables o,it is always the same for every run of this algorithm, so I do not include it here.

Is there an assumption in Algorithm 2 that the inputs were already uniquely identified?
RESPONSE: As we have add a new Algorithm 2, so this algorithm is now Algorithm 3.
For each input vector, Algorithm 3 check whether it is uniquely determined. So we don't have such assumption before running Algorithm 3, actually it is the result of this algorithm.

Section 4.1: the logic is not complete. The fact that R(a,b,0) and R(a,b,1) does not necessarily imply that a and b determine t (and should perhaps be written determine the value of t that satisfies R). R may not be satisfiable for any a and b.
RESPONSE: I am sorry I have not explained it more clear.
I add the last sentence "We further assume that..." at the end of the first paragraph of Subsection 4.1.
The way we construct R in Equation 14 and 18 guarantee that R is satisfiable.

It is possible that section 4.1 should become an appendix and only use its results for the paper. It is a diversion from the problem on discussion.
RESPONSE: Thank you for your suggestion. But I think it is short and leave it there is not a problem.