Review of
From High-Level Modeling Towards Efficient and Trustworthy Circuits

by M. Jaber, M. Noureddine, and F. A. Zaraket. Submitted to Sosym

Summary:

The paper presents a 2-step translation (via the so called "one loop programs") of BIP models into circuits. This has not been done before, and the novelty of the approach is most welcomed. However, i am missing some of the underlying motivation as some of the criticism provided by the authors is not, in fact, plausible: it is mentioned in several places in the introduction that

> "DFinder [7] does not handle data transfer between components, and it does not support checking for invariants other than deadlock freedom"
>
> "ABC either proves correctness or produces a counter example where the system violates an invariant. This enabled us to find defects and prove systems that were not possible using DFinder";

this is not the case[1] and in fact these observations are, in a way, peculiarly contradicted also by the authors themselves when in the "Related work" section they write:

> "DFinder [8] is an automated verification tool for checking invariants on systems described in the BIP language. Given a BIP system S and an invariant I, DFinder operates compositionally and iteratively to compute invariants X of the interactions and the atomic components of S. It then uses the Yices Satisfiability Modulo Theory (SMT) solver [20] to check for the validity of the formula $X \wedge \neg I = false$. Additionally, DFinder checks the deadlock freedom."

While the authors count the generation of C code as one of the advantages of their approach, they should do right and add that also the current BIP engine generates C++ code; in this light, the authors could emphasise the pluses of their approach: is it more efficient? Why? Please note that the statements in

> "We differ in that, the system specific scheduler is a bit vector of interactions directly embedded in the implementation. The interaction bit vector evaluates in real-time and directly depends on the locations and the values of the variables of the input system. The system specific execution framework empirically reduces the space and time requirements for the C simulation and the FPGA execution"

are not clear enough: the current BIP engine is quite optimised, and it also uses bits vectors, for example.

The statement on pg 17:

> "BipSV (BIP Synthesis Verification) is a Java implementation of the translation from BIP to OLP described in Section 5, and, is part of the BIP distribution [38]."

is not accurate: there is no BIPSV at the BIP website.

By solely reading the paper, the unprepared reader hasn't even a clue what "FPGA" stands for. The synthetisation of FPGA is not at all justified. The authors are content with just stating:

> "we are the first to synthesize a BIP system directly into an FPGA."

but to me, this is not a good enough reason if it is not clear what the end result in itself is good for.

[1] See, for instance "Rigorous Component-Based System Design Using the BIP Framework" at `http://www-verimag.imag.fr/~sifakis/RecentPublications/2011/ieee-software.pdf`.

Technicality:

I find the description of the transformations BIP $\rightarrow$ OLP $\rightarrow$ AIC too sketchy to really understand them. It is also too low-level for the reader to have a global view and a better intuition of the choices the authors adopt. The more that conceptually there is quite a difference between BIP models and circuits.

There is no single line about the soundness of the approach: why is the intermediary translation to OLP needed? how can one be sure that a property valid in a BIP model is valid also in the generated circuit? Besides, the semantics of OLP is missing. Why present for the illustrative example only the corresponding OLP and not the circuit itself?

As for the choice of examples, why not use real-time BIP instead of modelling time as ticks?

Readability:

A part from:

- the description of the transformations themselves (too low-level, see my observations referring the technicality of the paper)
- the discussion of the related work (too cryptic for a reader not familiar with hardware software and too chaotic in structure: NuSMV is mentioned on pg 23, last paragraph, then follows some short descriptions of [33] and [3], and then again appears an observation about NuSMV, why? and why are [33] and [3] only vaguely described and not also compared to?)

the paper reads fine.

Minor comments:

- pg 2, line 31: the statements stating that D-Finder supports just deadlock freedom and does not handle data are not accurate; what is FPGA? strictly speaking, a NAND gate is not defined
- pg 3, line 40:
  - please correct "netlists [...] employs"
  - remove the 1st "and" in "and iteratively, and [...]"
  - what's the purpose of the whole enumeration? the keywords denote quite orthogonal topics and their simple enumeration risks to puzzle the reader instead of impressing him; are all these techniques incorporated in the ABC framework? at this introductory level, the non-specialist reader has no clue what are "semi-canonical Boolean netlists with memory elements"
- pg 3, line 5: as mentioned before, "This enabled us to find defects and prove systems that were not possible using DFinder" is not accurate
- pg 6: The def of interaction (Definition 4) deviates from the one in previous BIP articles (e.g., "Modeling Heterogeneous Real-time Components in BIP"); in fact, it borrows from that of a connector in "Runtime Verification of Component-based Systems" leaving the notion of *connector* undefined in Definition 5. I assume that the authors' intention was to simplify the framework, nevertheless this should be explicitly mentioned and the references to connectors replaced by interactions.
- pg 11, line 3: "as discussed in Section." misses the number of the section.
- pg 12, line 22: BipSV is a forward reference.
- pg 17, line 43: ", and," please remove the commas; line 50: what is "openpm"?
- pg 18, Fig. 4: in `init-list`: why is light.m = 5 and why timer.n = 10? why isn't it mentioned the correspondence between the names of locations in Fig. 3 and the ints used timer.l and light.l?

- pg 25, line 26: what are “conservative decisions to avoid interaction conflicts”?
- bibliography: please correct [38]

Overall recommendation: reject.

Though the article starts from a novel and possibly interesting idea, i find that, all in all, the contribution is too weak to recommend “revision”. To account all the encouraged structural, motivational and technical changes would boil down to an article quite different from the current one, thus i find it fair that the authors consider a resubmission once the article is worked out to a new shape.