SYNOPSYS®
Silicon to Software™

SpecToSVA: Circuit Specification Document to SystemVerilog Assertion Translation

Document Intelligence Workshop @KDD - 2021

Ganapathy Parthasarathy, Saurav Nanda, Parivesh Choudhary, and Pawan Patil

July 31, 2021

Background Work

Verification in EDA

Design Verification is time-consuming due to huge manual effort

"Industry studies reveal that as much as 50% of the total schedule is being spent in verification."

Damiano, Robert, et al., DATE (2004)

"By consensus within the electronics industry, design verification burns 70% or more of the development cycle."

Raynaud, Alain. ElectronicDesign (2013)

"… up to 60% of the design cycle of a modern SoC consists of verification activities."

Harris, Christopher B. and Harris, Ian G., DAC (2016)

Automation in defining verification collateral can reduce manual effort

Generate automatically checkable formal properties from natural language specifications

SpecToSVA - Overview

Problem and Proposed Solution

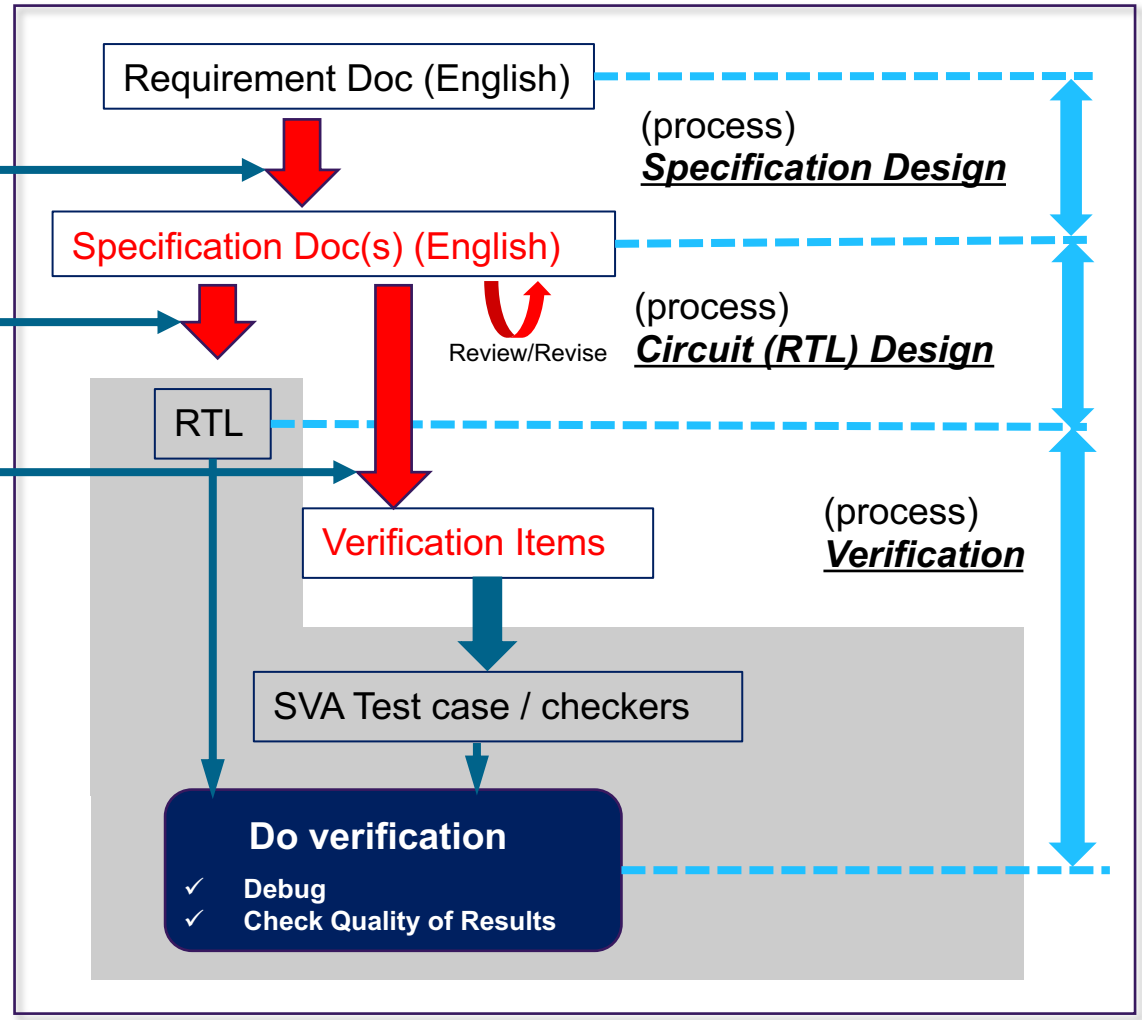
Problem

IP engineers create “Verification Items” manually.
- From Requirements Doc to Specification Doc
- From Specification Doc to RTL/Verification Items

Huge time and effort to :
- Create/review “IP verification Items”
- Errors in human translation causes re-spin/delays

Solution

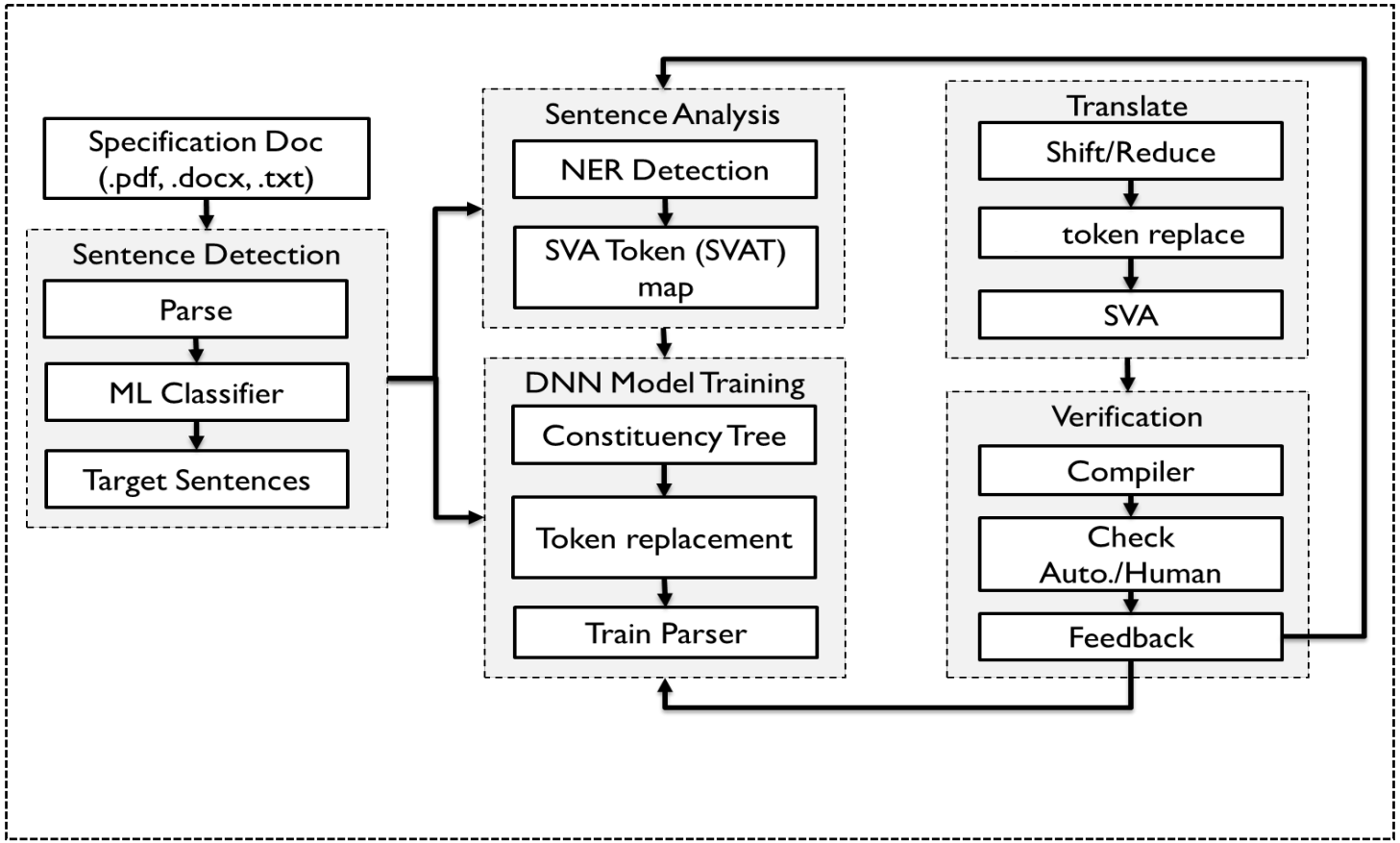
ML/Natural Language Processing-based solution to:
- Automatically synthesize formally checkable properties
- Do so with real-world QOR metrics
- Provide work-flow to iteratively improve QOR
- Deliver as secure cloud-solution (SaaS model)



SpecToSVA Architecture

System Architecture and execution flow for each sub-problem

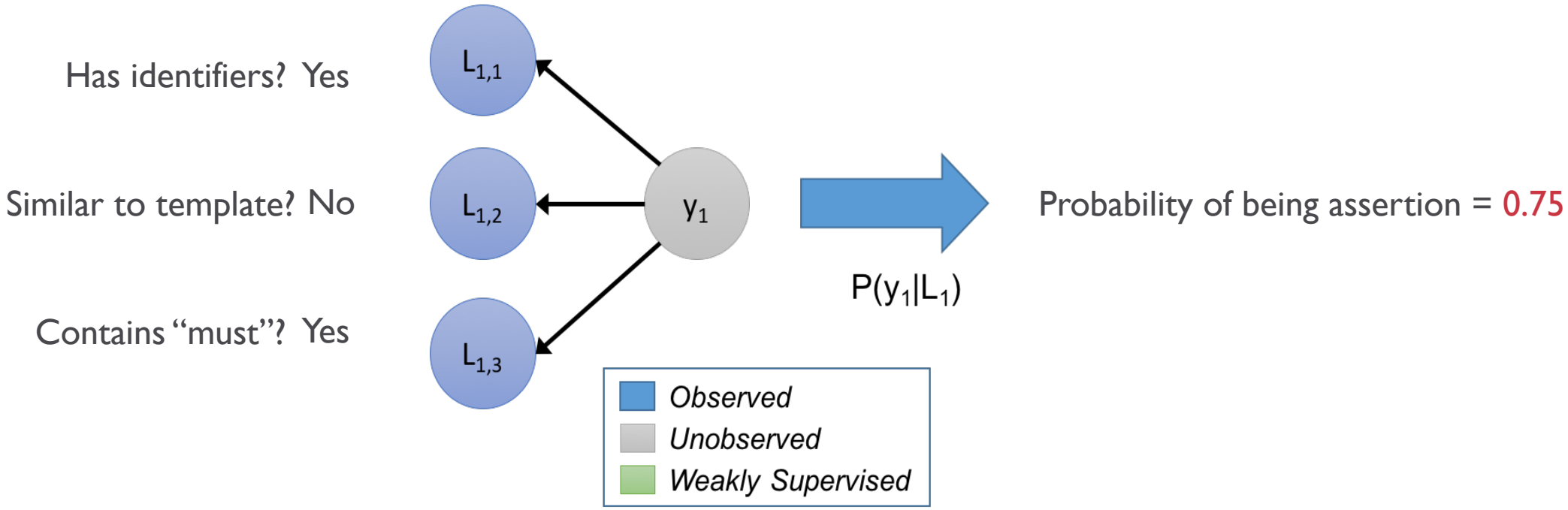
- Automatically synthesize verification items.
- Provide work-flow to iteratively improve QOR.
- Constituency Tree is enhanced using a custom NER model.
- SVAT: NER correspondence between English and SVA languages.
- Token Replacement: OOV markers.



Sentence Classification

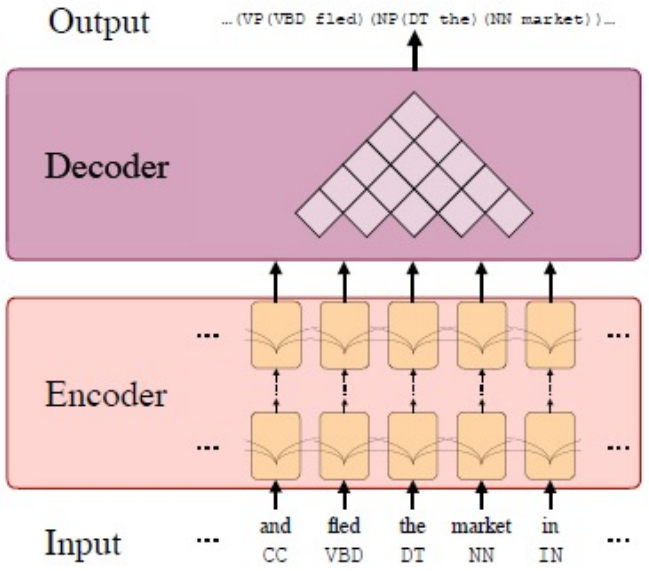
Explanation with an example

Sentence Classification [1] technique uses disagreeing heuristics to estimate marginals

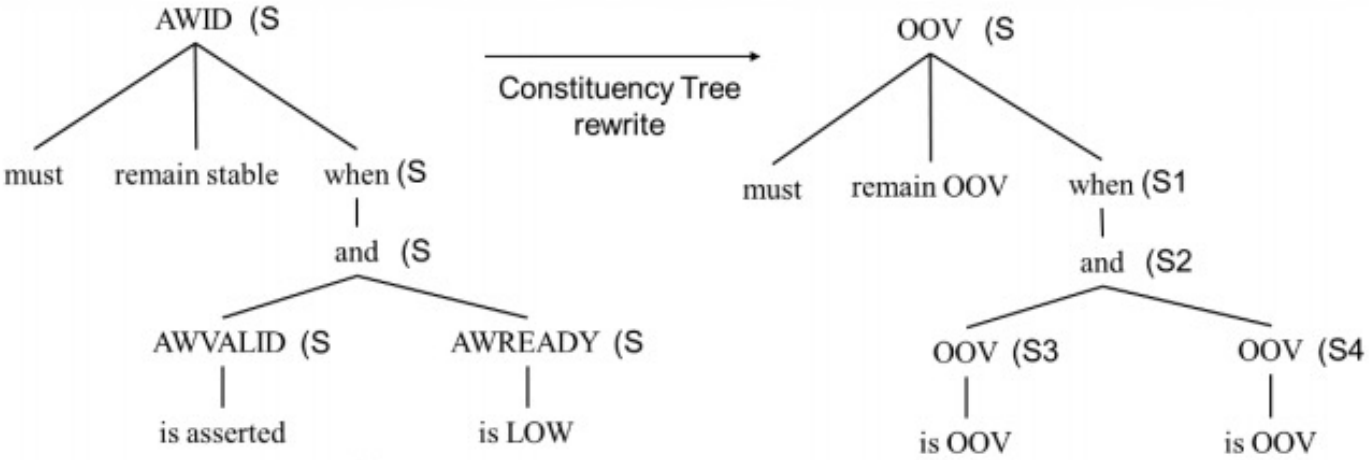


Sentence Analysis

NER Detection and Enhanced Constituency Tree Generation



Self-attentive parser [2]



Create a modified constituency tree from an English sentence

DNN Model Training

SPINN based DNN for Generator and Pointer Network

Generator Network

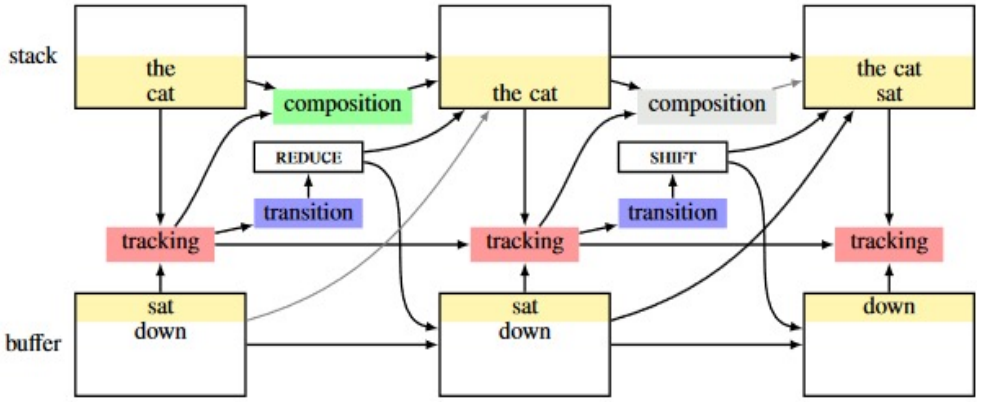
- Buffer and transitions (Encoder) and corresponding target lang (Decoder)
- Learns the structure of target language from the source lang
- Output has OOV (NER) tokens and SVA (ops/fn/const)
- Buffer has token positions of English language words

Pointer Network

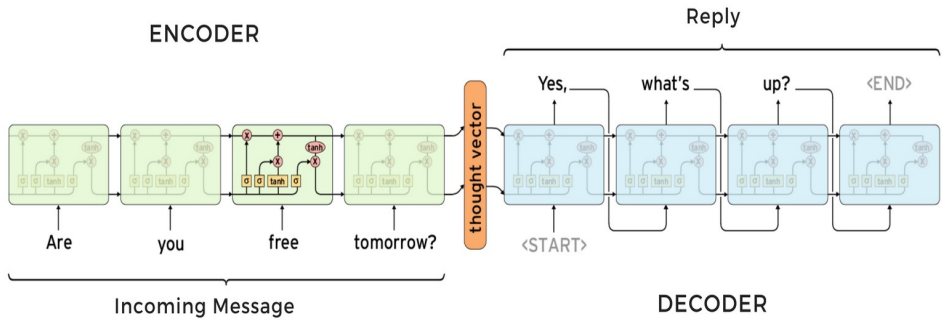
- Works similar to Generator Network
- It learns the relative positions of OOV (NER)
- Training data comprises of English and corresponding SVA
- Buffer has token embeddings of English language words

Transitions

- Post-order token sequence of updated constituency tree
- Every POS tag of constituency tree is a transition



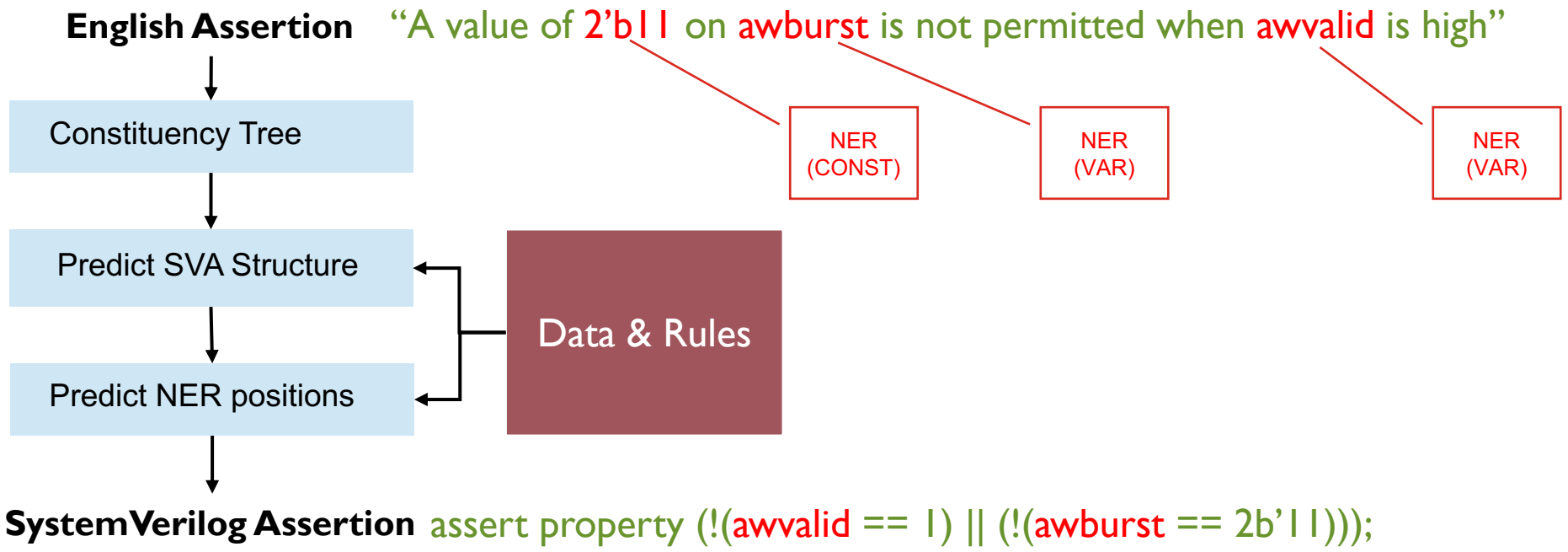
Probabilistic Parser (SPINN [3])



Generate translation tokens

Translation Process - Overview

Example to explain the translation process



Experimental Results

Example Translations

##	Input English	Output SVA
01	IRESP remains stable when IVALID is asserted and IREADY is LOW.	(ivalid && ! iready) |=>$stable (iresp)
02	IREQ is only permitted to change from HIGH to LOW when IACK is HIGH.	$fell (ireq) |->iack
03	When *transmit data is written to SPDR/SPDR_HA* and when *the transmit buffer of SPDR/SPDR_HA is empty (data for the next transfer is not set)*, the SPI writes data to the transmit buffer and clears the SPSR.SPTEF flag to 0.	((*def_wr_spdr_spdr_ha*) && (*user_def_spdr_spdr_ha_empty*) |-> (## *user_def_tim01*) $fell(sptef))

Examples of English sentences translated to SVA

Experimental Results

Performance and QoR

Dataset	Sentences	Classified	Translated	Precision
1	4896	4896	3605	73.63%
2	43	43	27	62.79%

Quality of results on internal data

Docs	Sentences	Classified	Translated	Precision
Doc-1	679	622	476	70.1%
Doc-2	1124	981	555	49.37%
Doc-3	12962	11745	8274	63.83%
Doc-4	19060	16933	12186	63.93%

Quality of results on commercial IC Specifications

Conclusion and Future Work

What do we have currently, and what we want to achieve in future?

- Conclusion
  - SpecToSVA provides a complete flow for IC designers and verification engineers to identify potential verification items.
  - Translate those verification items to formal language models i.e. assertions or checkers.
  - Experimental results support the utility of SpecToSVA
    - Different commercial/public IC design specification documents
    - Reasonable precision that significantly improved the productivity of verification engineers.
- Future Work
  - Find co-references between two or more sentences that can potentially be used for translation;
  - Combine and/or re-write those sentences that can be consumed by SpecToSVA for translation purposes.

References

[1] Alexander Ratner, Braden Hancock, Jared Dunnmon, Frederic Sala, Shreyash Pandey, and Christopher Ré. 2019. Training complex models with multi-task weak supervision. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33. 4763–4771.

[2] Nikita Kitaev and Dan Klein. 2018. Constituency Parsing with a Self-Attentive Encoder. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2676–2686.

[3] Samuel Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D Manning, and Christopher Potts. 2016. A Fast Unified Model for Parsing and Sentence Understanding. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 1466–1477.

Note: *Please refer to the paper for full list of references.*

SYNOPSYS®
Silicon to Software™

Thank You