AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AI Based Sequence Detection for Verification and Validation of IP/SoCs

Neena Chandawale
(Host)

Asif Ahmad
(Presenter)

Agenda

- Introduction
  - History of EDA tools and design methodologies
  - Design abstraction
  - Issues faced by designers
- An insight into ML/AI
- Introduction to NLP
- Deep Learning
- Recurring Neural Network
- RNN Based Long Short Term Memory (LSTM) usage
- Specification to code
- What’s a sequence ?
- Challenges faced with sequences
- Solution : auto sequence detection
- Implementation
- Example and usage in IDSCloud
- Limitations
- Summary
- Future scope
- References

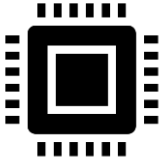
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Introduction

Introduction : History of EDA tools and design methodologies

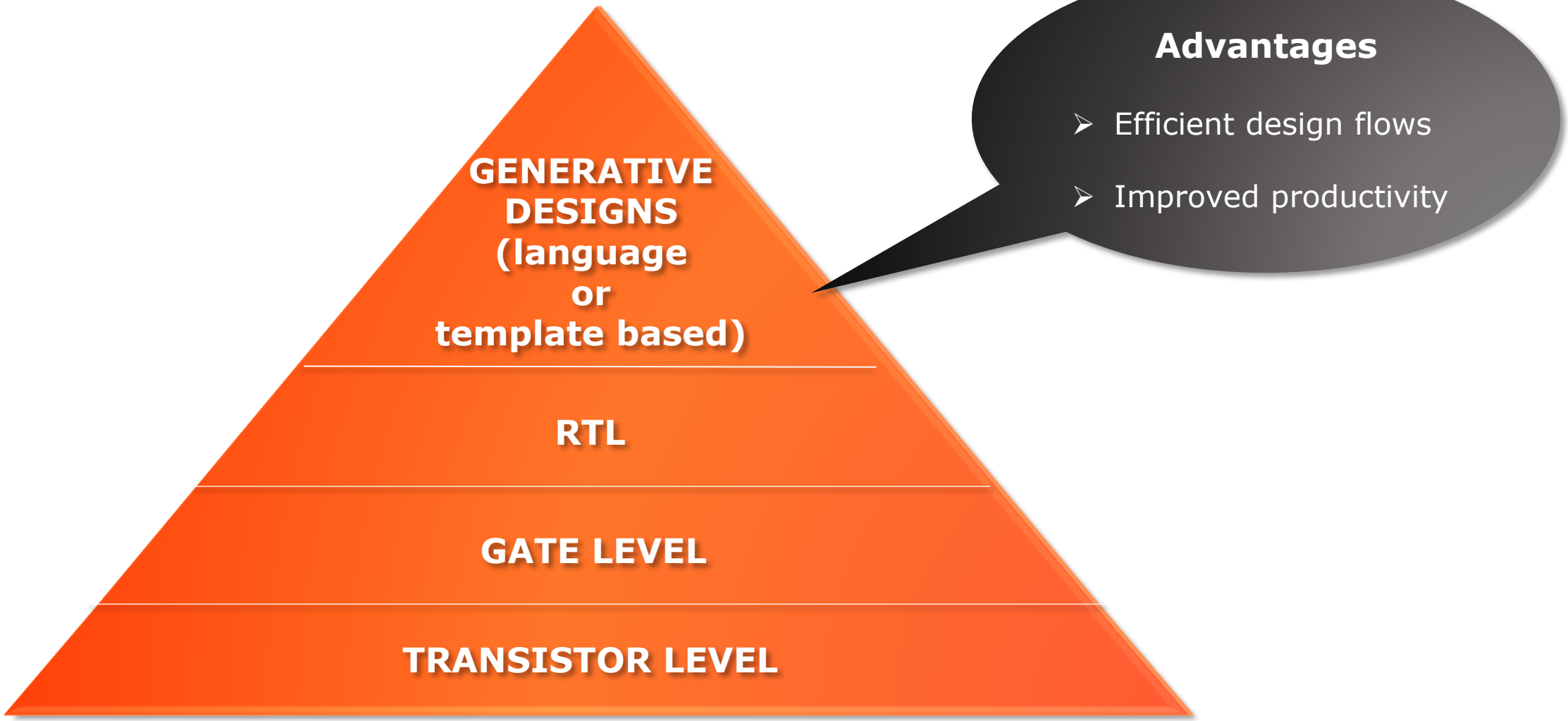
- Designs evolved from hundreds of transistors to hundreds of billions over last several decades



- Shrinking transistor size and increasing transistor count drove various design and verification methodologies

- Application of Machine Learning (ML) or Deep Learning (DL) techniques have also multiplied in aspects of register automation such as generation of some special registers, their assertions, etc.

Introduction : Design abstraction



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Introduction : Issues faced by designers

- In an ideal world, users would rather use plain and simple English text to describe the sequences rather than encode in various languages

- Natural, plain English is still the hallmark of specifications and a lot of useful and actionable information is embedded in the natural language specification text

- Numerous translations happen when the architect/designer creates a specification in English language and the hardware/software/firmware engineer manually converts it into code

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

An insight into ML/AI

An insight into ML/AI

- Artificial Intelligence (AI) has become a catch-all term that refers to any computer program that automatically does something
- AI works at its best by combining large amounts of data sets with fast, iterative processing and intelligent algorithms
- AI is basically a broader term, whereas ML is a subset of it and consists of more advanced techniques and models that enable computers to figure things out from the data and deliver AI applications.
- Their usage has increased by many folds in diverse fields such as face detection technique, object detection technique, language translation, chatbots, spam detection, data scraping, etc.

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

An insight into ML/AI

- Through AI, rule-based applications have taken a back seat as there have been many algorithms invented that are capable of defining their own rules or creating classifiers such as linear Regression, Logistic Regression, Trees, SVM, etc.

- Along with the algorithms, what is really important is the data used to train the model for these algorithms

- Their applications have translated into shorter turnaround times and provide greater flexibility in analysis and simulation coverage, shifting a step ahead towards broader automation

An insight into ML/AI

- In EDA, the application of ML/AI techniques enables the modeling and the simulation with an unprecedent levels of insight, to expect greater efficiency and accuracy from design tools

- It can help in identifying patterns to optimize design, allowing the designers and the testers to model more complex design in simpler way and in lesser time

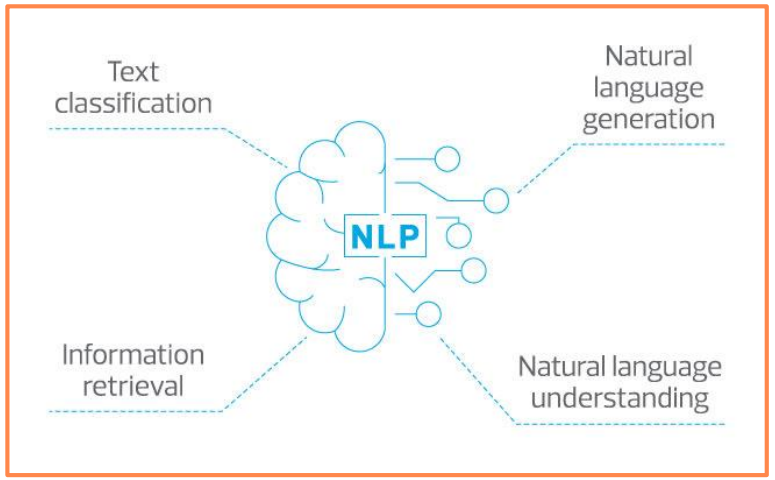
- ML/AI-generated models can provide better feedback to the designers and other engineers by indicating whether the design would live up to the expected performance at each step of the development process

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Introduction to NLP

Introduction to NLP (Natural Language Processing)

- The usage of NLP for manipulation of natural language text to capture sequences is now possible using deep learning techniques
- Applications such as machine translation, speech recognition, conversational chatbots, and POS (parts of speech) tagger have been most popular among the NLP applications
- This field grew out of the field of linguistics and has succeeded above expectations so far
- It can be used for the development of deep learning models for classification of text, translation of text, and more
- It is sometimes also referred to as 'linguistic science' in order to include both classical linguistics as well as modern statistical methods

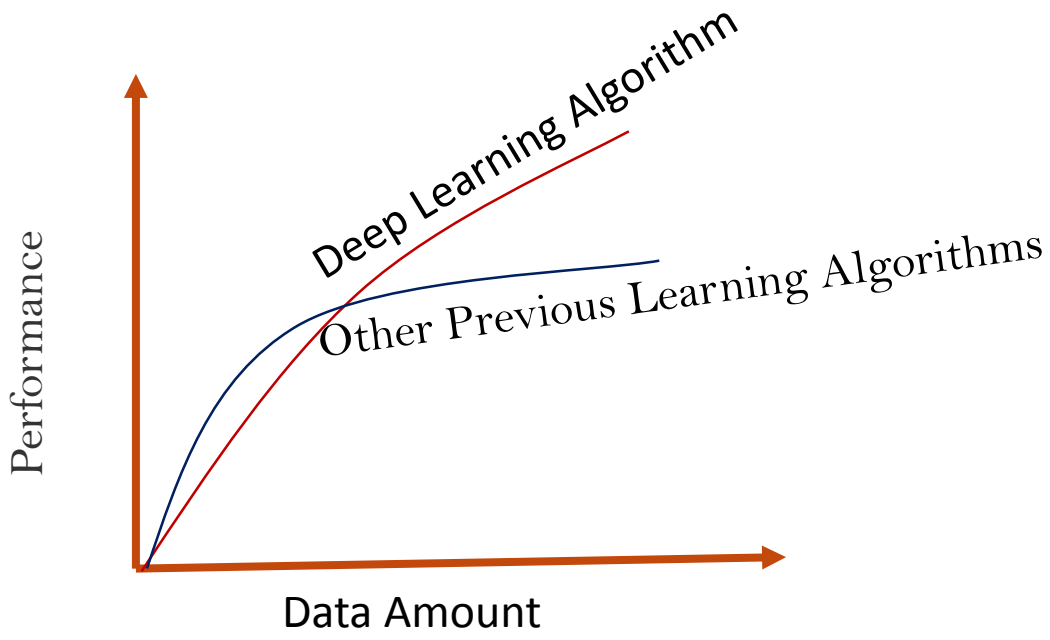


AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Deep Learning

Deep Learning

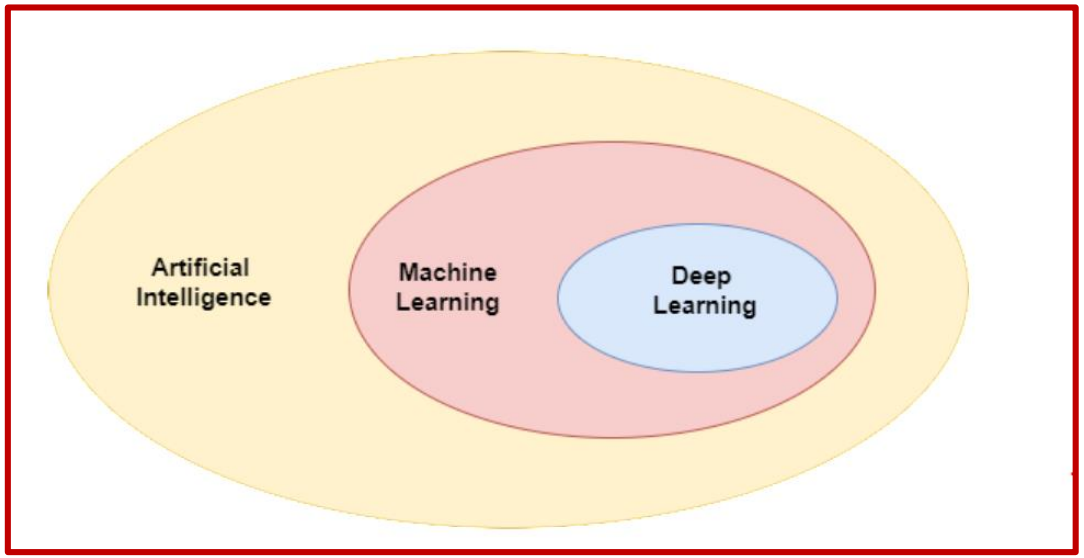
- Deep Learning is a subfield of machine learning that focuses on the algorithms inspired by the structure and function of the brain.
- These techniques have proved useful in solving challenging natural language processing problems.



Deep Learning

- Several neural network architectures have had great impact in addressing natural language processing tasks through deep learning.

- Deep Learning techniques also enable the modeling and the simulation with an unprecedent levels of insight, hence one can expect greater efficiency and accuracy from design tools



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

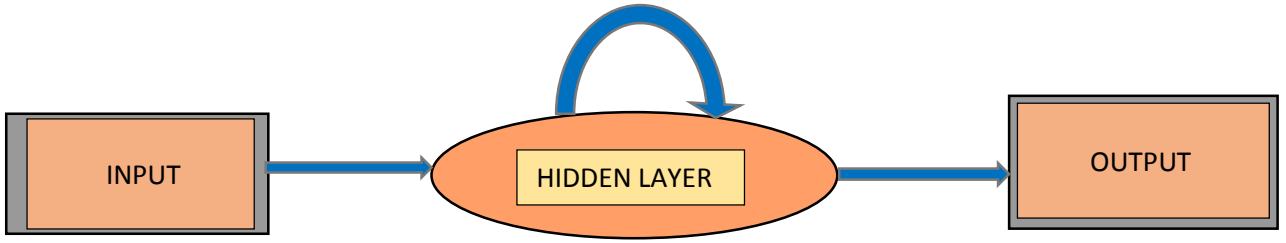
Recurring Neural Network

Recurring Neural Network (RNN)

- The RNN is a series of algorithms that attempts to recognize basic relationships in a set of data similar to a way that human brain operates
- It is designed to deal with real world problems such as machine translation, chatbot, etc.
- The inputs and outputs are connected as previous step output is fed as current step input to predict the new text
- The RNN comes with a hidden state that basically remembers some information about a sequence with the help of a memory in it
- It uses same parameters recursively for all inputs and performs the same task on them as well as the hidden layers to produce the output reducing the complexity of those parameters

Recurring Neural Network

- It is used in NLP to carry pertinent information from one input item to other in series, i.e., it can take a series of inputs without any predefined limit on size
- Every word gets transformed into machine readable vectors, and this sequence of vectors are processed one by one
- The processing is done by passing the hidden state to the very next step of the sequence and this hidden state acts as a memory of the neural network
- This makes it applicable to accomplish tasks such as unsegmented, connected handwriting recognition or speech recognition



A typical RNN unit with a recurrent hidden layer

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

RNN Based Long Short Term Memory (LSTM) usage

RNN Based Long Short Term Memory (LSTM) usage

- LSTM is a modified version of the RNN that makes it simpler to remember past data in the memory

- This trains the model using back propagation and, at every time step, attention is given to those words that direct towards the prediction of most part of the output

- This attention weight is calculated using an algorithm and formula for each time step

- The attention mechanism allows to focus on a certain part of the input sequence while predicting a certain part of the output sequence that ultimately enables easier learning and higher quality of prediction

- A final context vector is then built by the dot product of all attention values, which is also known as stacking

RNN Based Long Short Term Memory (LSTM) usage

Example: -

Let the context vectors be c1, c2, c3, ...

with h1, h2, h3, … as output vectors of the encoder

and α1, α2, α3, ... be their attention weights

So, the dot product would be -

$$c_i = \sum_{j=1}^{4} \alpha_{ij} h_j$$

- The output is fed word by word to the decoder of the LSTM network and at each time step the context vector is used to produce appropriate output
- This attention mechanism located between the encoder and the decoder enables improved performance

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

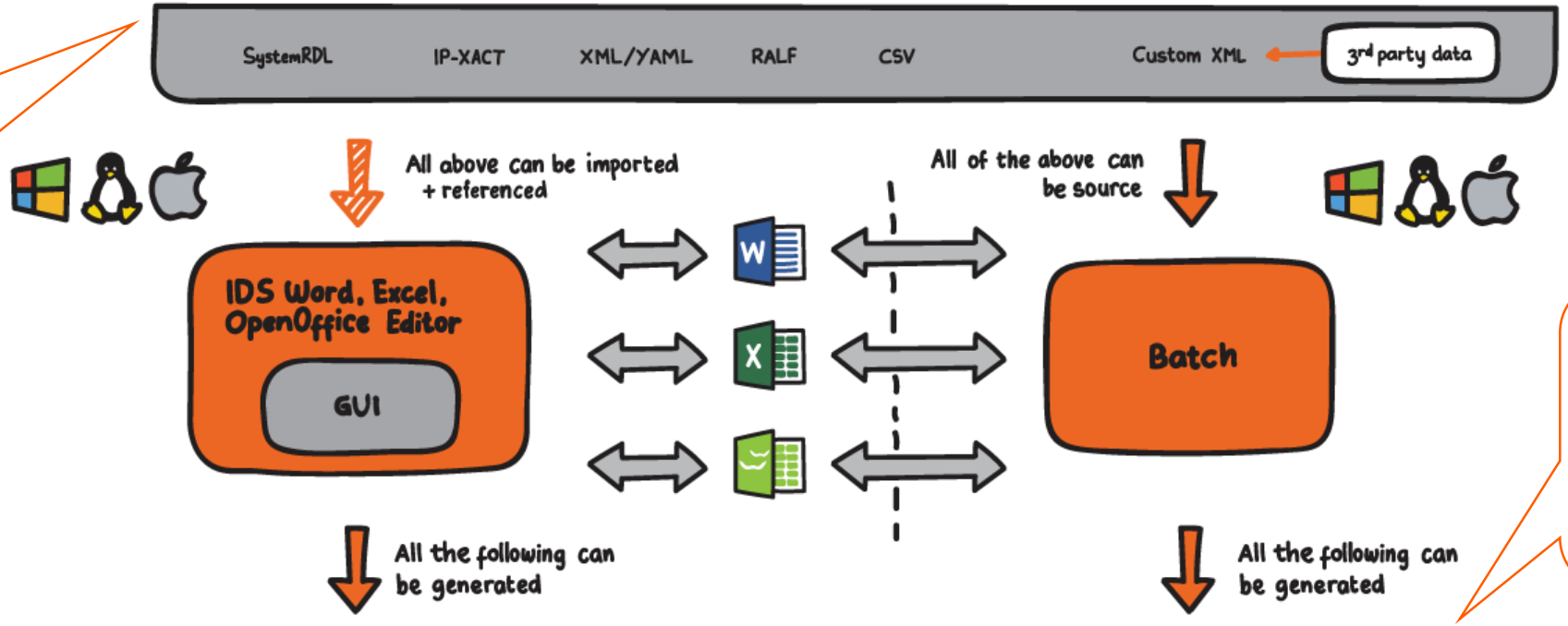
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Specification to code

Specification to code : IDesignSpec™

- **IDesignSpec (IDS)** is a generator that takes a high-level specification and generates parameterizable code
- So, it helps IP/SoC design architects and engineers to create an executable specification for registers and automatically generate output for SW and HW teams

Specifications can be written in **MS Word, MS Excel, LibreOffice** or text-based industry standard formats such as **SystemRDL, RALF or IP-XACT**



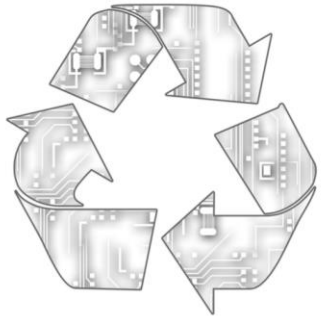
IDesignSpec captures simple as well as **special registers, signals, interrupts, and generates synthesizable RTL, UVM model, C/C++ headers, HTML or PDF**

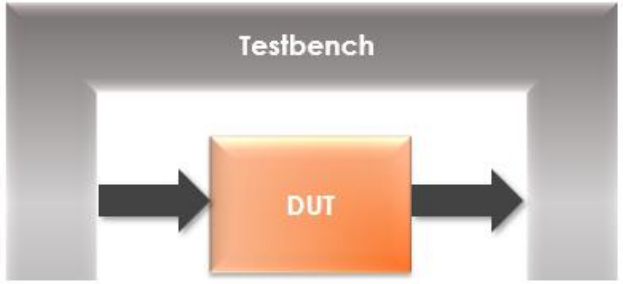
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

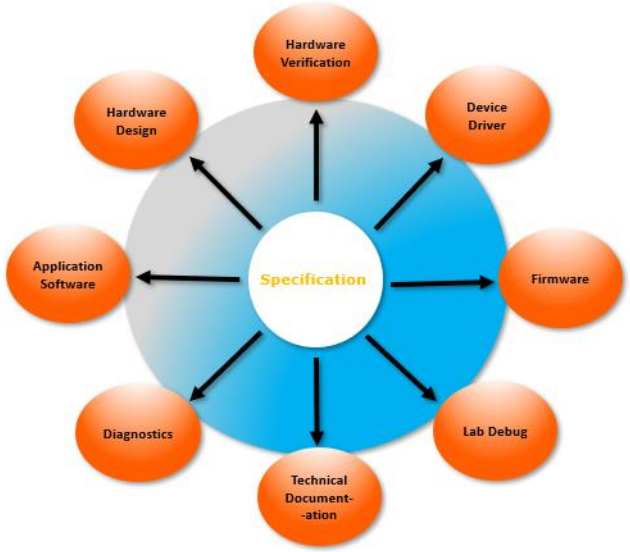
Specification to code : IDesignSpec™

Benefits and Capabilities

- Text/Template based approach
- Design reusability
- Eliminate inefficiencies
- Generate hardware and testbench
- A better feasible and productive approach





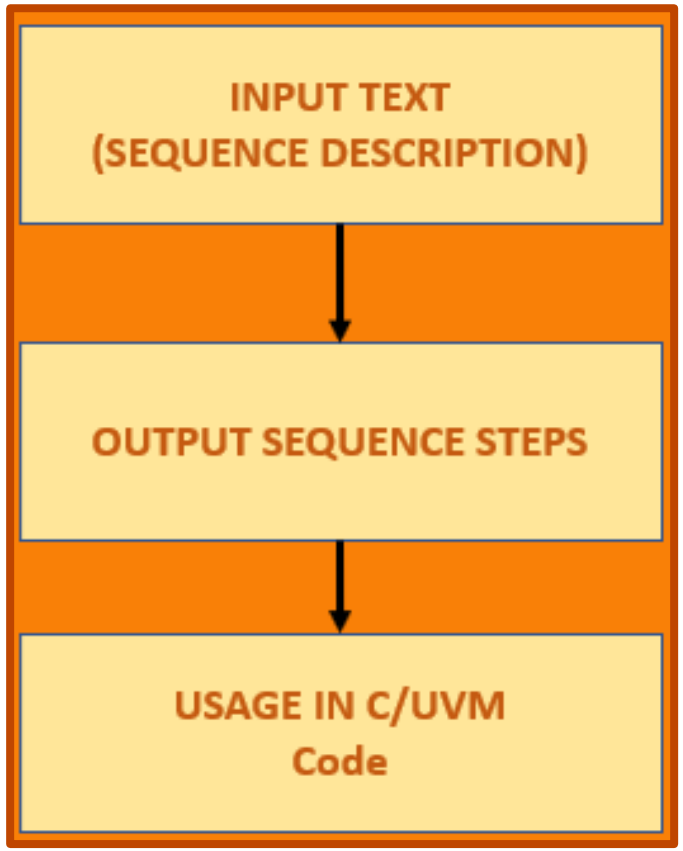


AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Specification to code

- With our continued strive for automation, we are taking a step further by helping designers automatically generate customized sequences from the defined English language specification

- These generated AI based sequences can be used in C/UVM code



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

What's a Sequence ?

What's a Sequence?

- A list of steps that need to be executed in order
- A flow-chart or an algorithm
- Used for configuration setting (programming) or test

Functional

Reset
Power up
Low power
Initialization
Functional modes
Filter coefficients
Eeprom settings
Memory descriptors

Test

Parameterized
Randomized
Constrained

- Sequences are created at various stages of the design Process
  - Frontend design verification — SystemVerilog/UVM
  - Embedded code — C/C++
  - Backend configuration of chips — C/C++
  - Backend testing and validation — C/C++

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

What's a Sequence?

Sequences are built on registers, memories, and pins

- Sequences contain
  - Register / Field writes
  - Register / Field reads
  - Pin manipulation commands
  - Wait / Function calls, subsequence calls

- Information about registers/memories can be in any format
  - IP-XACT
  - SystemRDL
  - Word / Excel
  - Text files

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Challenges faced with Sequences

Challenges faced with Sequences

Manual capturing of sequences from the documentation/English language specification

Difficulty faced by designers in writing verification/validation sequences with their correct syntax

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Solution : Auto Sequence detection

Solution : Auto Sequence detection

- Numerous translations happen when the architect/designer creates a specification in English language
- The hardware/software/firmware engineer must manually convert these specification into code
- With this new methodology, the specification writer’s original intent is converted into real, usable code
- RNN based network is used to read the input text (i.e., the sequence description text) word by word
- The order of the sentence formation is maintained so as to learn the meaning of input text, and each word is processed through an RNN unit

Solution : Auto Sequence detection

- These units process all the words one by one and hence generate output information as its result

- Bidirectional layer has been used, which reads the input text form both the directions (i.e., both forward and backward) improving the performance of the model on this sequence classification

- Focus on embedding is performed, which basically treats each input text word by making their vector forms and ultimately each word is represented with some fixed size vector of numbers

- Attention algorithm is used, which helps in predicting more closely expected outputs, i.e., the most probable expected output sequence

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Solution : Auto Sequence detection

- The data that is fed to the model for training should be as good as possible, i.e., the dataset should be appropriate to expect a greater performance from the model

- We have carefully created our corpus by getting references from actual used register programming sequences in the electronic design automation (EDA) industry and have also introduced a wide variety of cases including cases with augmented data or noise

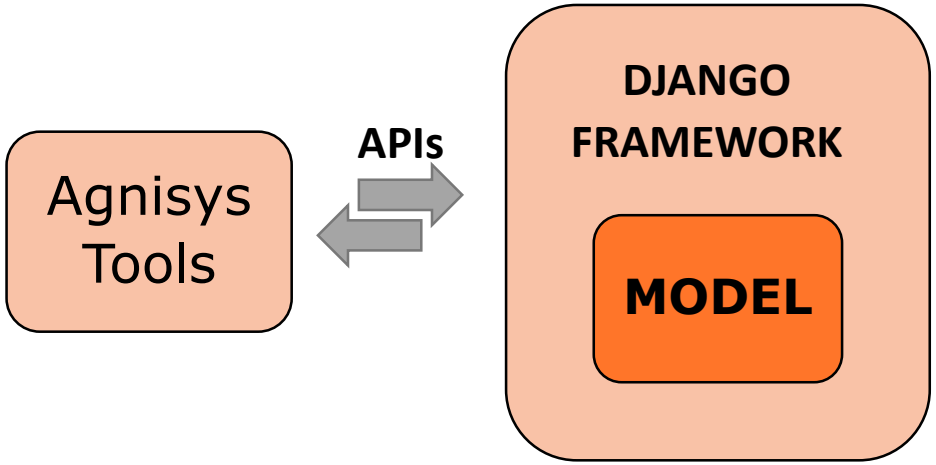
- This robust model has great accuracy covering almost all scenarios of sequences that can be used by a designer for description of an input text sequence

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Implementation

Implementation

- This model has been deployed using Django Framework to maintain communication between the model and our spec entry tools through various APIs handling multiple requests at a single time



COMMUNICATION FRAMEWORK

- The above communication network represents the interaction of a spec entry tool with the model through REST APIs

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

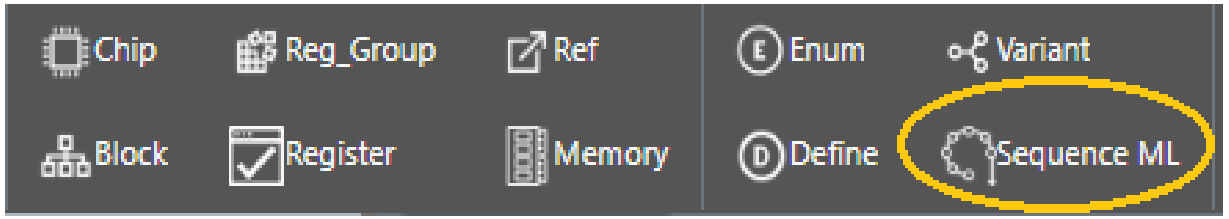
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Example and usage in IDSCloud

Example and usage in IDSCloud (web based spec entry tool)

Here are the steps to use the AI based sequence detection functionality in IDSCloud

- Login to “ids.agnisys.com”
- Create your Project in IDSCloud
- Create your IP/Specification
- Click on the “Check” button in the IDS tool bar to validate the legality of the specification
- Once the document is checked successfully, go to ‘Sequence view’
- In the ‘Sequence view’, click on the 'Sequence ML' button present in the IDS tool bar:



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Example and usage in IDSCloud

- In the ML Sequence template, write your sequence name in the cell provided below ‘sequence’ column, IP name in the cell provided below ‘IP(Design)’ column and then sequence description in the ‘Description’ column and the actual sequence with syntax will be automatically inserted in the ‘Sequence Steps’ column:

Sequence	IP(Design)
seq1	Example.idsng
Description	**Sequence Steps**
Update the values of all the fields of register Sbcs with value 0x1 and then monitor the field Sberror	write Sbcs = 0x1 read Sbcs.Sberror

If the register Sbcs's bits Sbacc is disabled then then wait for 100 time units and then program the value 0x5 on the register dma_controller.
Else if the field Sbacc of Sbcs is set to equal to or greater than 2 then monitor Sberror. and set dma_controller to 0x2. Else if dma_controller is set then wait for 5 time units and then reset dma_controller.
Else assert whether Sbcs is set.
	if (Sbcs.Sbacc == 0) {
	wait (100)
	write dma_controller = 0x5
	}
	else if (Sbcs.Sbacc >= 2) {
	read Sbcs.Sberror
	write dma_controller = 0x2
	}
	else if (dma_controller == 1) {
	wait (5)
	write dma_controller= 0
	}
	else{
	assert (Sbcs == 1)
	}

Set ULPIDATRD and then monitor register Sbcs.	write USB_OTG_HS_USBMODE.ULPIDATRD = 1 read Sbcs
Assert whether register dma_controller is set to low and then set dma_controller to high.	assert (dma_controller == 0) write dma_controller = 1

If Sbcs's Sbacc is less than 0x3 then sample Sbcs.	if (Sbcs.Sbacc < 0x3) {
	read Sbcs
	}

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Example and usage in IDSCloud

- After writing and capturing your desired sequences in the Sequence template, go to the config window and select desired outputs including the sequence outputs (firmware and UVM)

- Click on the “Generate” button to generate the outputs (one can also download the zip file of the outputs generated)

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Limitations

Limitations

- An issue remains with words that are unknown in the vocabulary/dictionary of the model: For such words the model may produce unexpected outputs as those are totally out of context for the model

- In some cases, failure in data interpretation may occur because of insufficient data input text

- The model may produce unexpected results if not trained with enough data that is accurate, i.e., the training dataset needs to be large and accurate

- Computational power, inference time, etc. may be a hindrance for viable usage with larger architecture

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Summary

Summary

- We have been able to handle a wide variety of cases with an accuracy as good as more than 90% with no delay in inference time

- This model can effectively handle noise in the input text and thus gives attention to only the relevant part of the text that has any influence in the output sequence generating correct output sequences

- We have achieved this by improving on and working around the limitations such as sufficient and correct amount of dataset to train the model, and improving the inference time by reducing the complexity of the model architecture and other issues that we faced along the way

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Future scope

Future scope

- We further plan to make our model more robust, covering all legal text description of sequences with more diverse training
- Including more hierarchal and complex scenarios for output predictions
- Improving the accuracy of our model against the current accuracy of 90%
- Working more precisely on the pre-processing part for improved efficiency and better results

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

References

References

- https://www.agnisys.com/ids_nextgen/
- https://ids.agnisys.com/login
- https://content.riscv.org/wp-content/uploads/2018/05/15.55-16-30-UL-001906-PT-C-RISCV-Debug-Specification-Update-and-Tutorial-for-Barcelona-Workshop.pdf
- https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-vol-3b-part-2-manual.pdf
- https://www.analog.com/media/en/technical-documentation/user-guides/ADV7511_Programming_Guide.pdf

Agnisys Webinar Series

BRINGING THE LATEST AUTOMATION IN IP/FPGA/SOC TO YOUR HOME!

Time : 10:00 AM – 11:00 AM PDT

08-April-2020 Correct by construction SV UVM code with DVinsight - a smart editor	30-April-2020 Advanced UVM RAL - callbacks, auto-mirroring, coverage model, and more	28-May-2020 Steps to setup RISC-V based SOC Verification Environment
09-April-2020 Creating portable UVM sequences with ISequenceSpec	7-May-2020 Functional safety and security in embedded systems	04-June-2020 Automatic verification using Specta -AV - a boost to verification productivity
16-April-2020 Register automation from SystemRDL to PSS - Basic to Pro	14-May-2020 IP generators - the next wave of design creation	11-June-2020 AI based sequence detection for verification and validation of IP/SoCs
23-April-2020 Cross platform specification to code generation for IP/SoC with IDS-NG	21-May-2020 A flexible and customizable flow for IP connectivity and SoC design assembly	18-June-2020 Understanding clock domain crossings

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

About Agnisys

- The EDA leader in solving complex design and verification problems associated with HW/SW interface
- Formed in 2007
- Privately held and profitable
- Headquarters in Boston, MA
  - ~1000 users worldwide
  - ~50 customer companies
- Customer retention rate ~90%
- R&D centers (US and India)
- Support centers - committed to ensure comprehensive support
  - Email : support@agnisys.com
  - Phone : 1-855-VERIFYY
  - Response time within one day; within hours in many cases
  - Multiple time zones (Boston MA, San Jose CA and Noida India)

IDESIGNSPEC™ (IDS) EXECUTABLE REGISTER SPECIFICATION

Automatically generate UVM models, Verilog/VHDL models, coverage model, software model, etc.

AUTOMATIC REGISTER VERIFICATION (ARV)

ARV-Sim™ : Create UVM test environment, sequences, and verification plans, and instantly know the status of the verification project

ARV-Formal™ : Create formal properties and assertions, and coverage model from the specification

ISEQUENCESPEC™ (ISS)

Create UVM sequences and firmware routines from the specification

DVinsight™ (DVi)

Smart editor for SystemVerilog and UVM projects

IDS – Next Generation™ (IDS-NG)

Comprehensive SoC/IP spec creation and code generation tool

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

THANK YOU