Cross platform specification to code generation for IP/SoC with IDesignSpec-NG

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Amanjyot Kaur
(Host)

Azhar Saifi
(Presenter)

Neena Chandawale
(Presenter)

Agenda

- Typical Chip design
- Components of typical SoC
- Overview of IDesignSpec™
- Challenges of single platform tool
  - Plugin/Add-in
  - Enterprise Features
- Solution – IDesignSpec-NextGen
  - Single IDE Tool
- Features and Power of IDE (IDS-NG)
- How to get Started with IDS-NG
- Quick Demo
- Conclusion

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

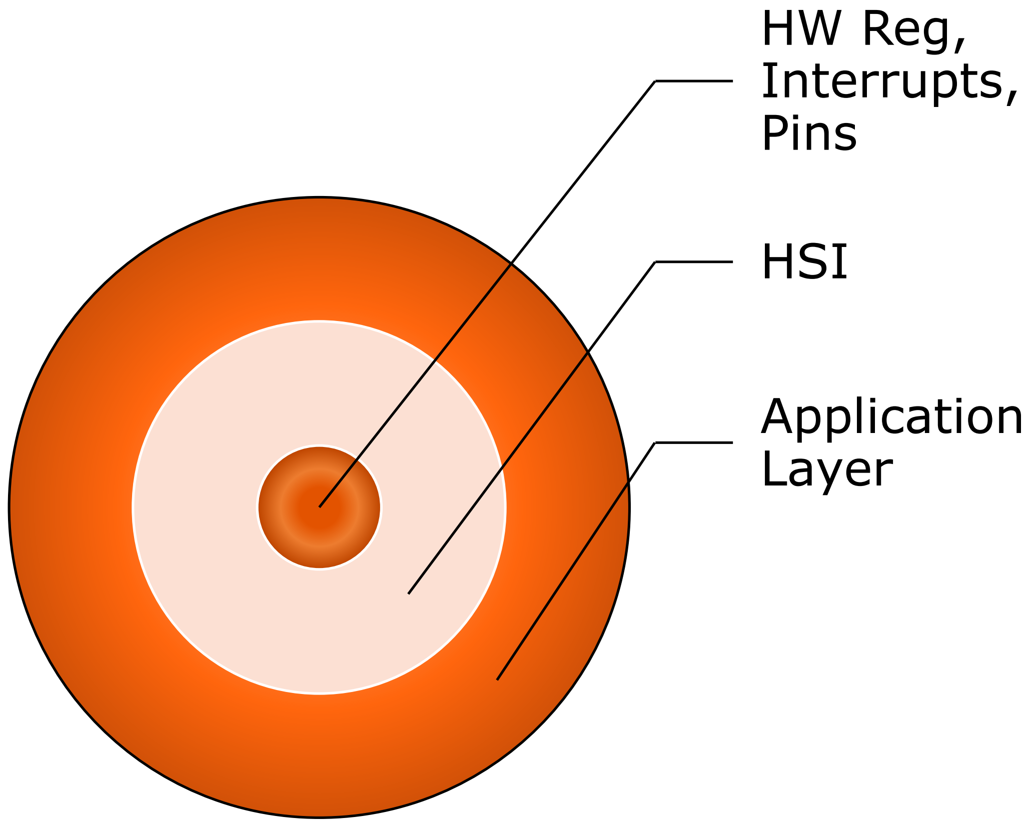
Typical chip design

- Hardware of the SoC is designed by HW team
    - But used by
        - Verification/Emulation team
        - Firmware team
        - Validation team
        - Software team

- How does the software interact with the IPs?
    - Through the Hardware Software Interface (HSI)

- Hardware is at the core and software API is around it

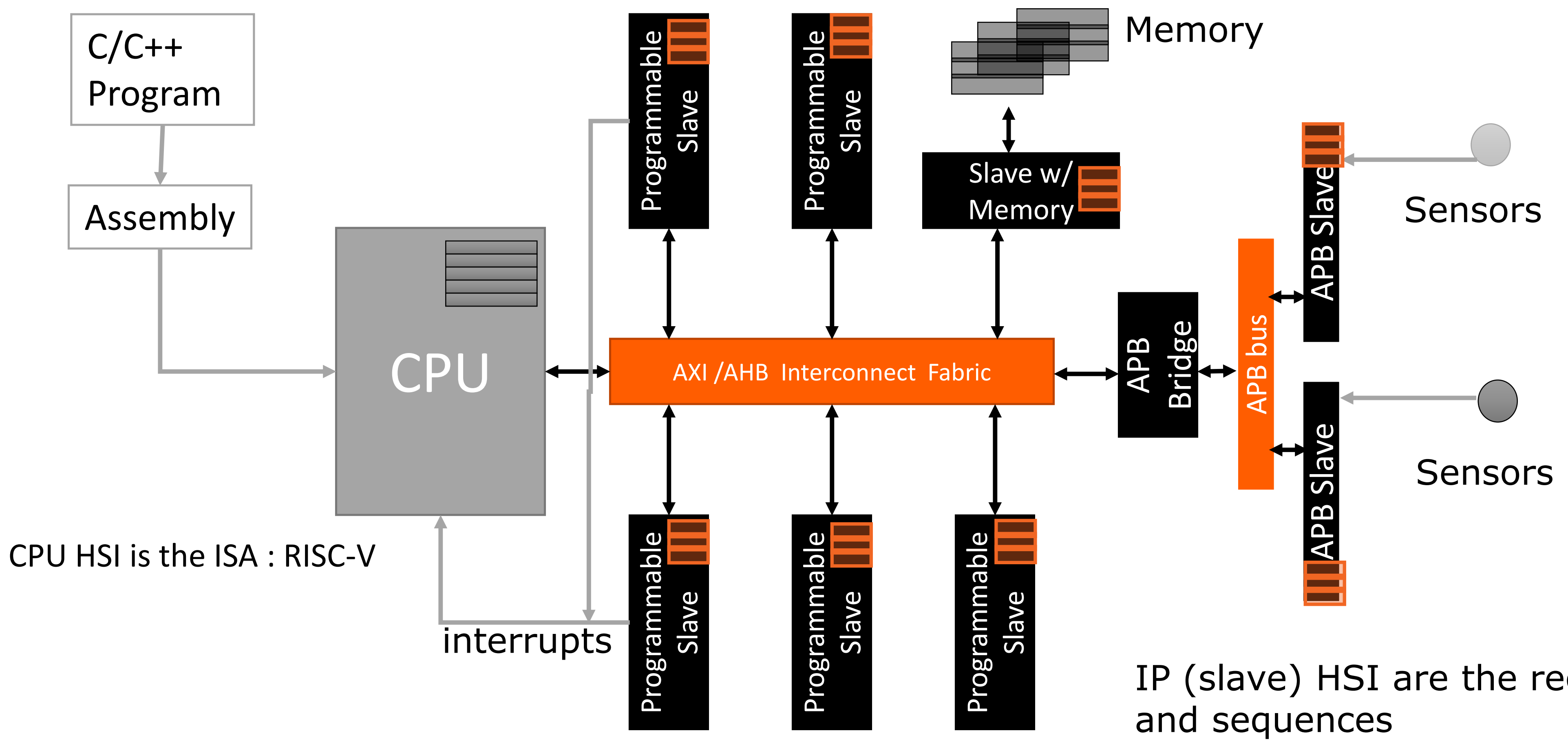
- Device Driver(part of the HSI) are tedious to create
    - They are created in C and Assembly



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Components of a typical SoC

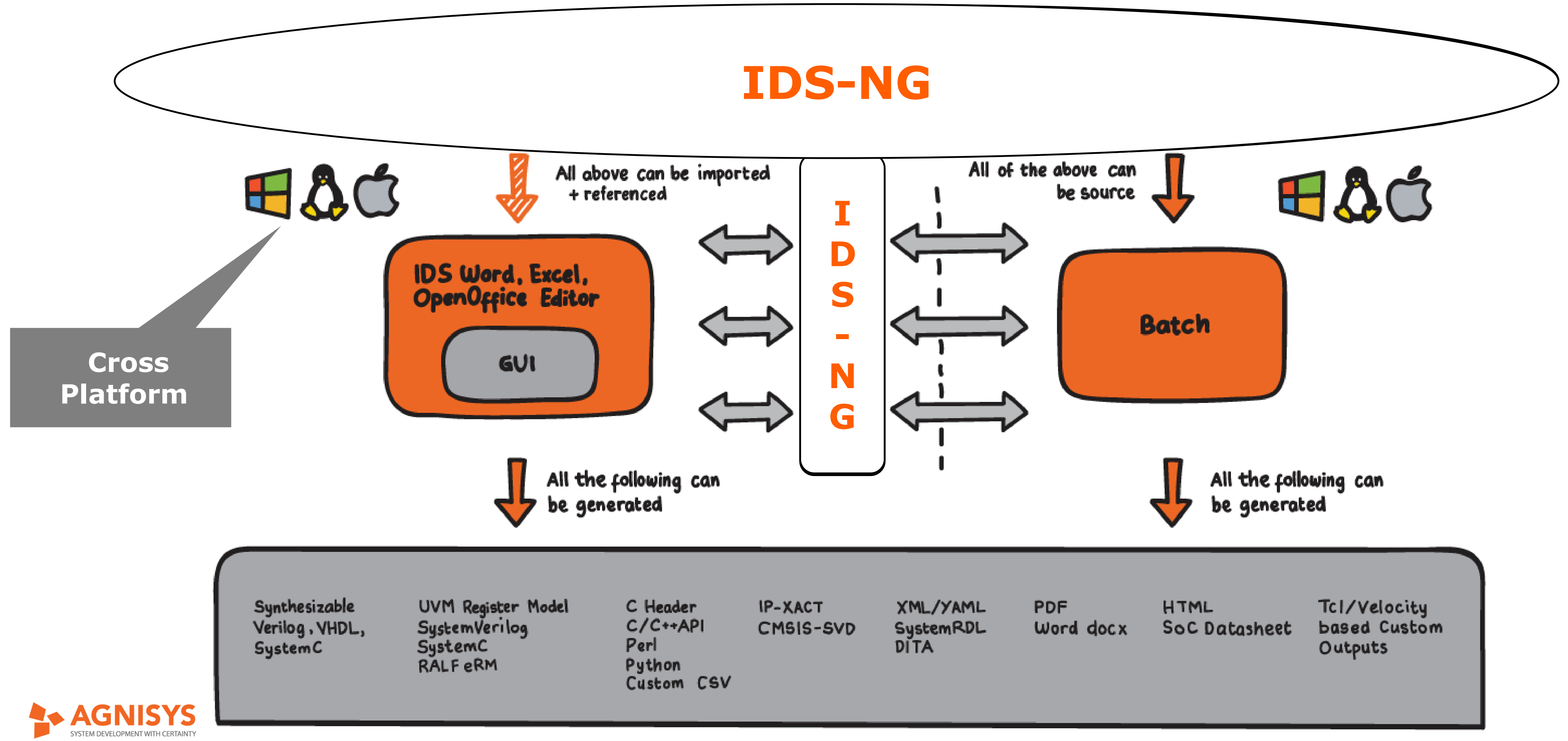
The slaves are programmed by reading/writing to the embedded register



CPU HSI is the ISA : RISC-V

IP (slave) HSI are the registers, interrupts, and sequences

IDesignSpec™ - Centralized Register Design/Verification from a Golden Spec


IDS-NG
All above can be imported + referenced
All of the above can be source
IDS Word, Excel, OpenOffice Editor
GUI
I D S - N G
Batch
Cross Platform
All the following can be generated
All the following can be generated
Synthesizable Verilog, VHDL, SystemC
UVM Register Model
SystemVerilog
SystemC
RALF eRM
C Header
C/C++API
Perl
Python
Custom CSV
IP-XACT
CMSIS-SVD
XML/YAML
SystemRDL
DITA
PDF
Word docx
HTML
SoC Datasheet
Tcl/Velocity based Custom Outputs

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

IDesignSpec™

Register Design Entry

- Importing SystemRDL, IP-XACT, CSV, XML, RALF, YAML
- Add-in to Word, Excel, OpenOffice Calc
- Register Templates

Code Generation

- Synthesizable VHDL/Verilog/SystemVerilog/SystemC, UVM Model, C Headers
- General Properties, RTL Properties, UVM Properties, C/C++ Properties, SV Header Properties
- Parameterization
- Bus Protocols
- Verification Plan
- Batch Utility, Perl, Python

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

IDesignSpec™

Documentation

- HTML Alt 1, HTML Alt 2, PDF, Custom PDF, Word, SVG
- SoC Datasheet Generation

Special Registers

- Lock, Alias, Trigger-Buffer, Interrupt registers, Shadow registers, Indirect registers, Counters
- FIFO registers, RO-WO Pair, Paged, Virtual, Multi-Dimensional, RegAlign, Wide, Triple Module Redundancy (TMR)

Advanced Features

- Clock Domain Crossing, Low Power RTL, Custom Circuitry, Parity/CRC, Power/Performance Circuitry
- Variants, Special Control Signals, SW/HW Access Types
- SystemRDL UDPs
- Velocity Template and TCL-API

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Challenges

- Multiple tools required for different specification formats
  - MS-Word, OpenOffice for Documents
  - MS-Excel, Calc, OpenOffice for Spreadsheets
  - RDL Editor & Compiler for System RDL
  - XML Editors for IPXACT
- Manual and tedious setup of development infrastructure using add-ins, third party tools, home grown scripts, etc.
- Solutions are usually platform specific customized for a specific platform
- Requirement of different tools adds to the licensing expenses
- Dependency on other tools for tracking the larger specification and version control

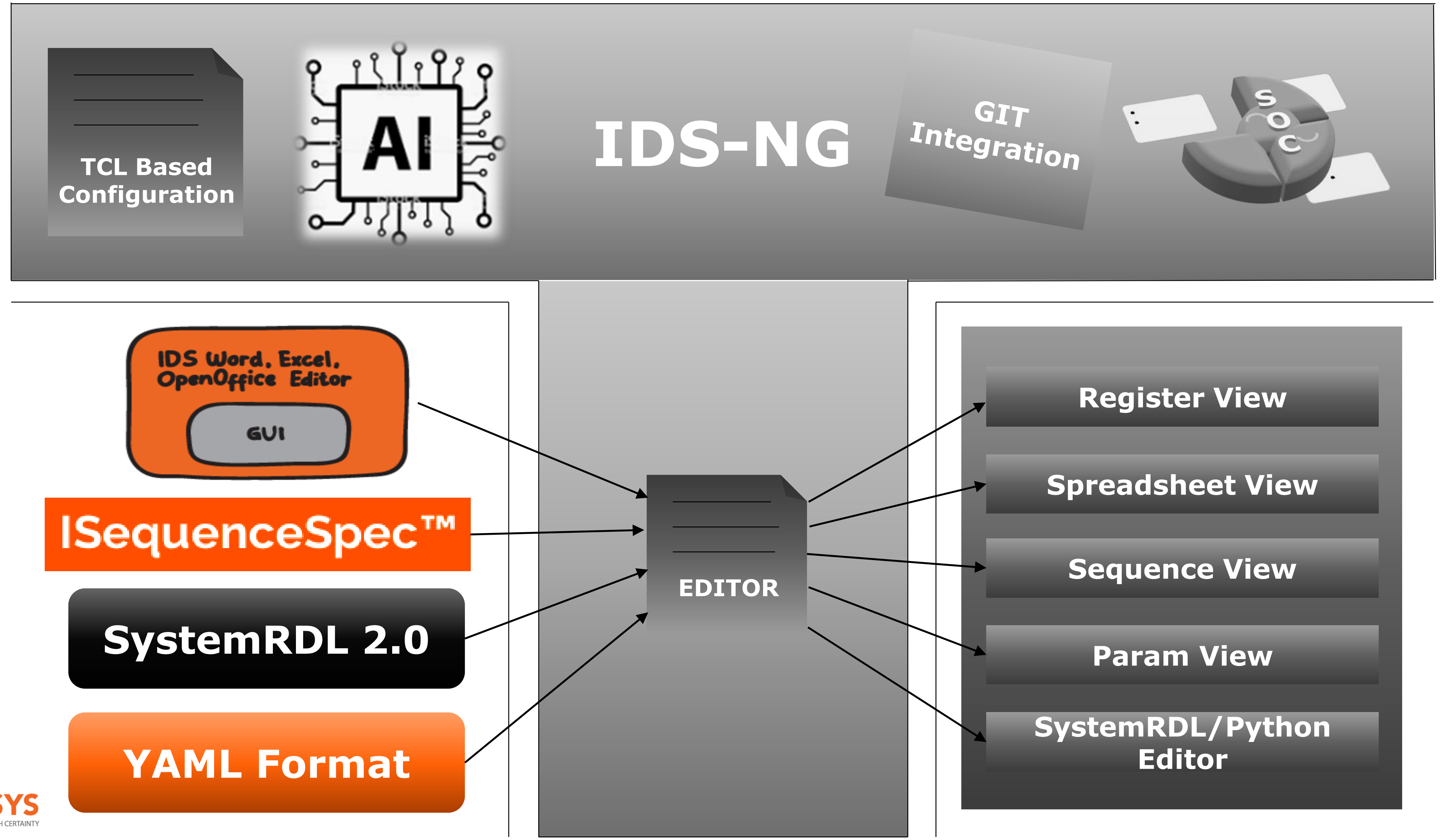
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

IDesignSpec™-NextGen

- The specialized integrated development environment for large IP/SoC focused on HSI
- Platform independent, available for Windows, Linux and MAC
- Captures registers in multiple views:
    - Register view, Spreadsheet view, Param view and System RDL Editor for registers
    - Sequence view and Python Editor for sequences
- Data saved as text
    - GIT Integration – Changes and merges possible
- AI based sequence detection from natural language

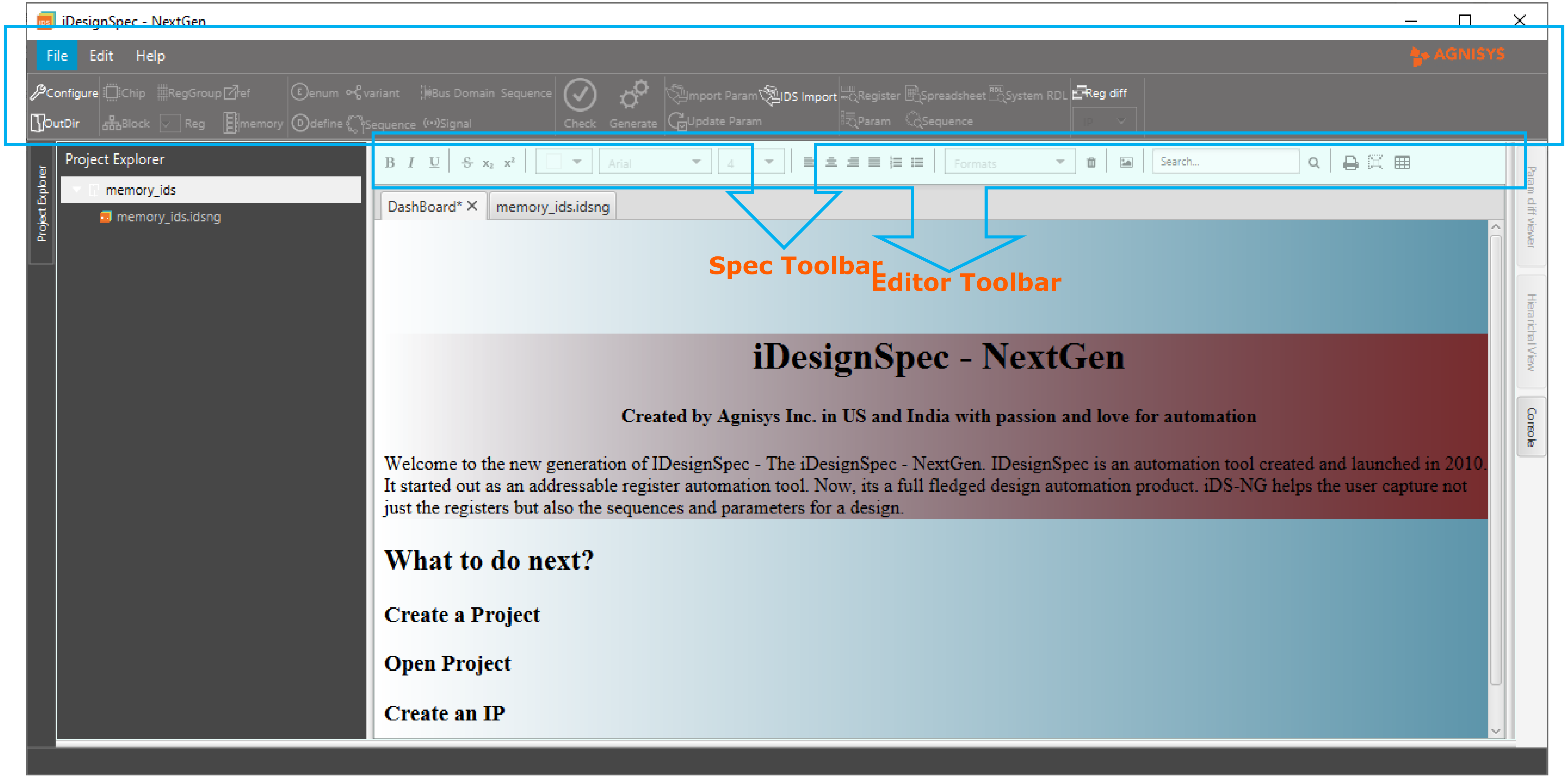
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Cross platform IDE



AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

GUI of IDS-NG



iDesignSpec - NextGen

Created by Agnisys Inc. in US and India with passion and love for automation

Welcome to the new generation of IDesignSpec - The iDesignSpec - NextGen. IDesignSpec is an automation tool created and launched in 2010. It started out as an addressable register automation tool. Now, its a full fledged design automation product. iDS-NG helps the user capture not just the registers but also the sequences and parameters for a design.

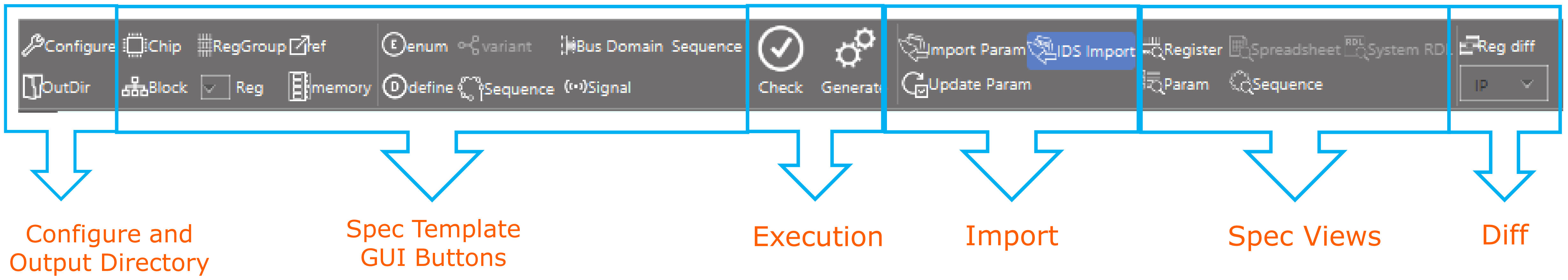
What to do next?

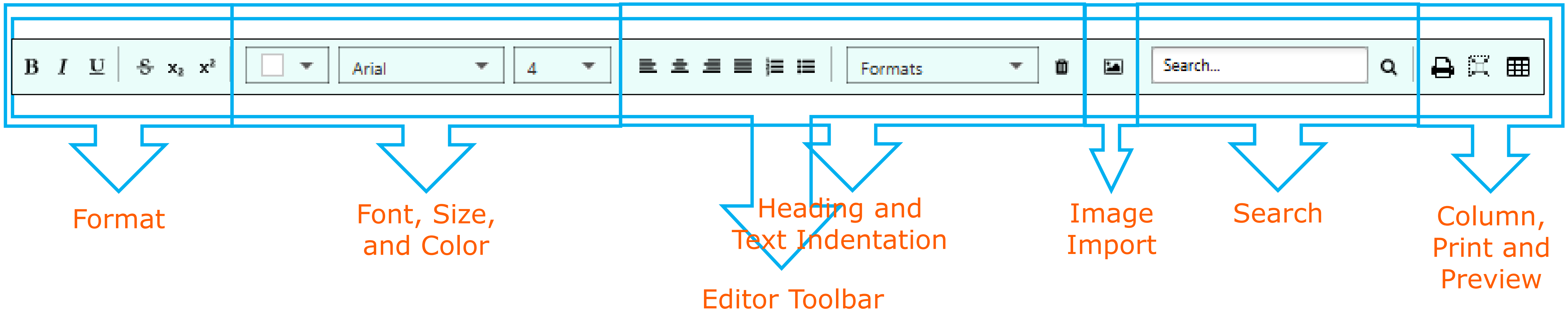
Create a Project

Open Project

Create an IP

IDS-NG toolbar



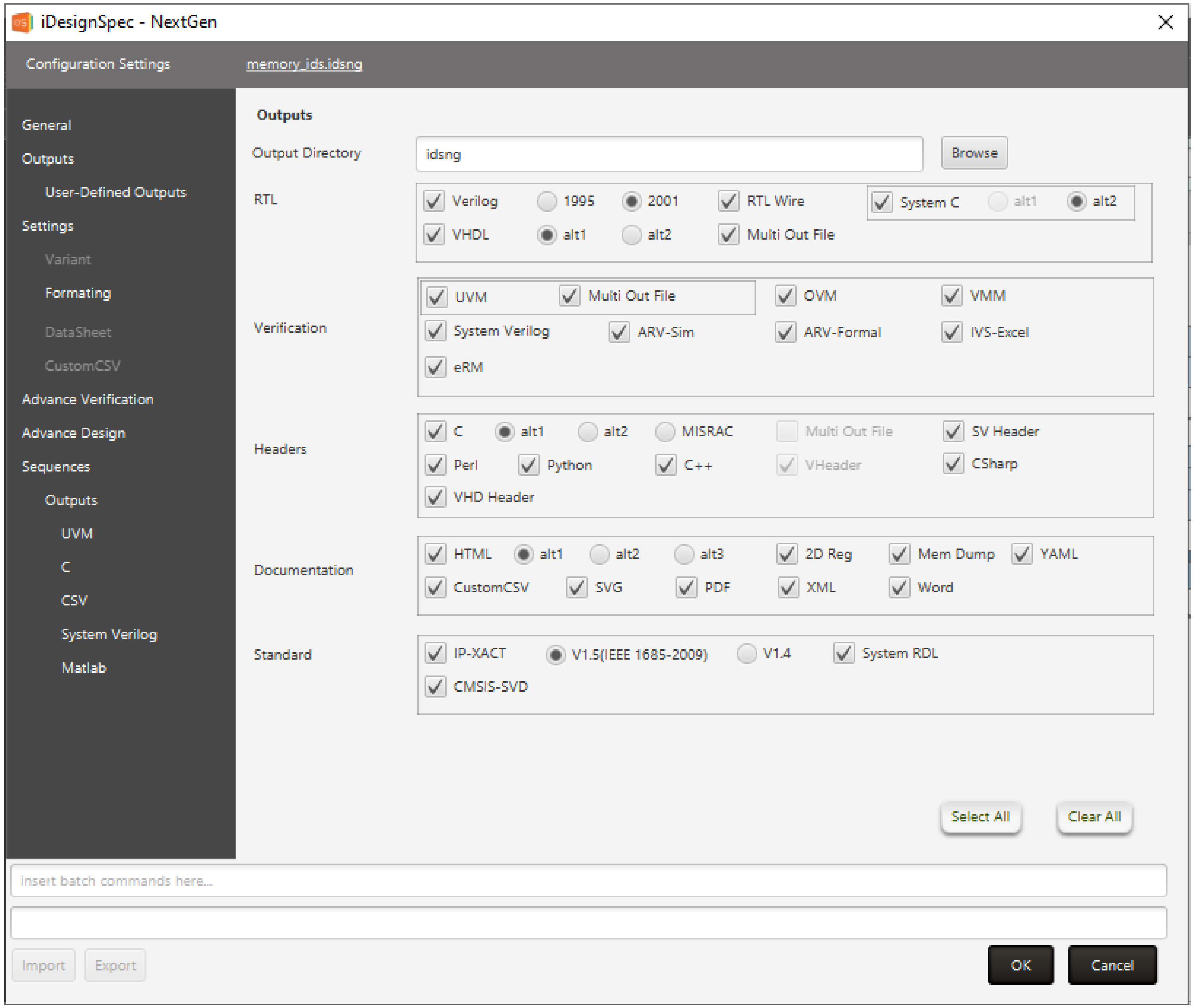


AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Register Outputs

IDesignSpec-NextGen can generate various configurable register outputs:

- Synthesizable RTLs.
- Verification Methodologies.
- Firmware outputs.
- Documentation level outputs.
- Industry standard outputs.



iDesignSpec - NextGen

Configuration Settings   memory_ids.idsng

General
Outputs
  User-Defined Outputs
Settings
  Variant
  Formating
  DataSheet
  CustomCSV
Advance Verification
Advance Design
Sequences
  Outputs
    UVM
    C
    CSV
    System Verilog
    Matlab

Outputs

Output Directory: idsng [Browse]

RTL:
[✓] Verilog ( ) 1995 (●) 2001 [✓] RTL Wire [✓] System C ( ) alt1 (●) alt2
[✓] VHDL (●) alt1 ( ) alt2 [✓] Multi Out File

Verification:
[✓] UVM [✓] Multi Out File [✓] OVM [✓] VMM
[✓] System Verilog [✓] ARV-Sim [✓] ARV-Formal [✓] IVS-Excel
[✓] eRM

Headers:
[✓] C (●) alt1 ( ) alt2 ( ) MISRAC [ ] Multi Out File [✓] SV Header
[✓] Perl [✓] Python [✓] C++ [✓] VHeader [✓] CSharp
[✓] VHD Header

Documentation:
[✓] HTML (●) alt1 ( ) alt2 ( ) alt3 [✓] 2D Reg [✓] Mem Dump [✓] YAML
[✓] CustomCSV [✓] SVG [✓] PDF [✓] XML [✓] Word

Standard:
[✓] IP-XACT (●) V1.5(IEEE 1685-2009) ( ) V1.4 [✓] System RDL
[✓] CMSIS-SVD

[Select All] [Clear All]

insert batch commands here...

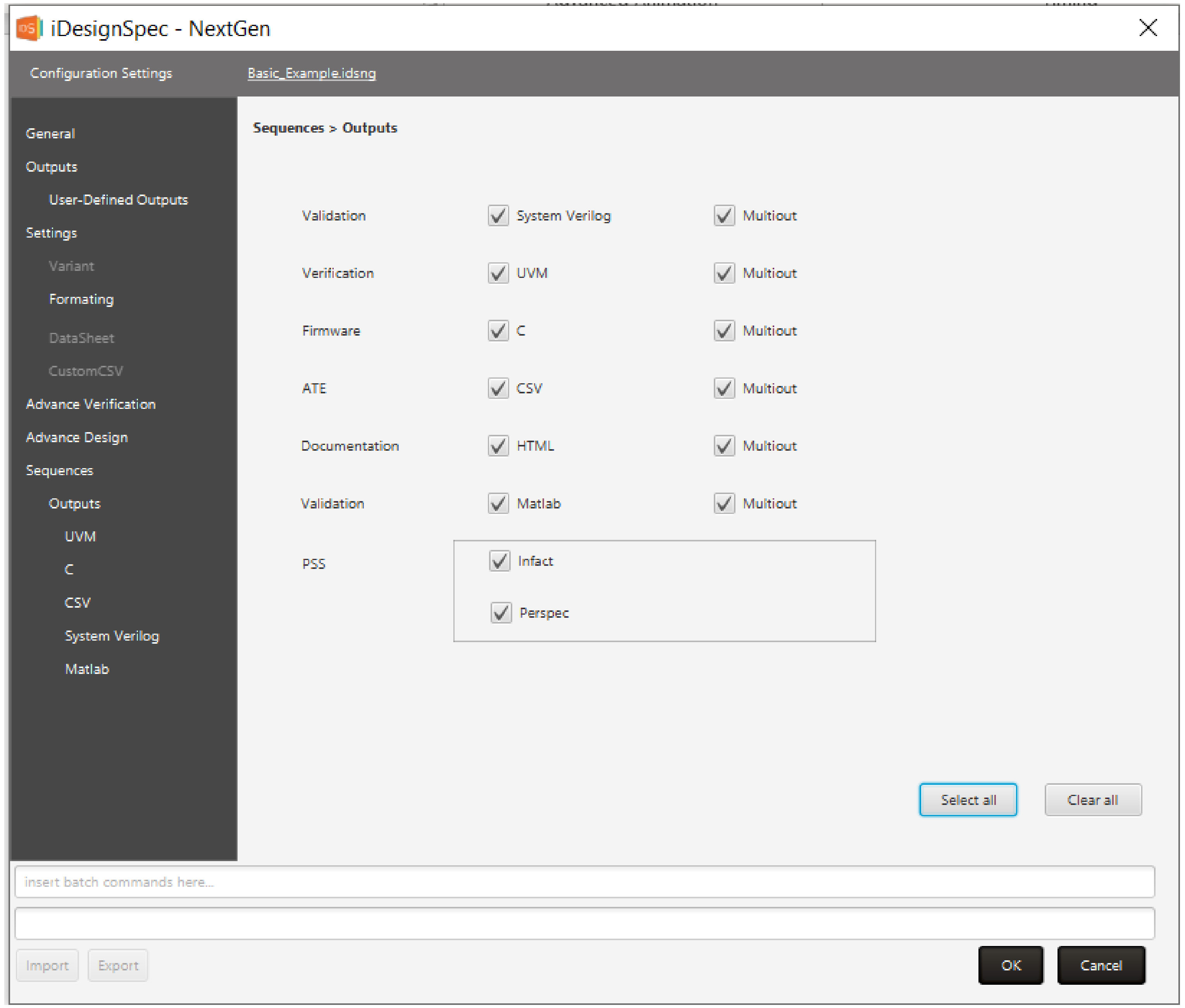
[Import] [Export] [OK] [Cancel]

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Sequence Outputs

IDesignSpec-NextGen can generate multiple configurable sequences outputs:

- Validation outputs
- Verification outputs.
- Firmware outputs.
- ATE outputs.
- Documentation outputs
- PSS outputs.



Sequences > Outputs

Validation	✓ System Verilog	✓ Multiout
Verification	✓ UVM	✓ Multiout
Firmware	✓ C	✓ Multiout
ATE	✓ CSV	✓ Multiout
Documentation	✓ HTML	✓ Multiout
Validation	✓ Matlab	✓ Multiout
PSS	✓ Infact	
	✓ Perspec	

Conclusion

- IDS NextGen is a very powerful IDE which helps generate accurate code for not only just registers but also sequences in one integrated environment.

- It is a cross platform enterprise class product that is an indispensable tool for design, verification, software, firmware and technical documentation teams.

- It reduces the verification time by generating the entire UVM, SV and C output sequences.

- It speeds up the time to market and quality at reduced overall costs.

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
  - 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, 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

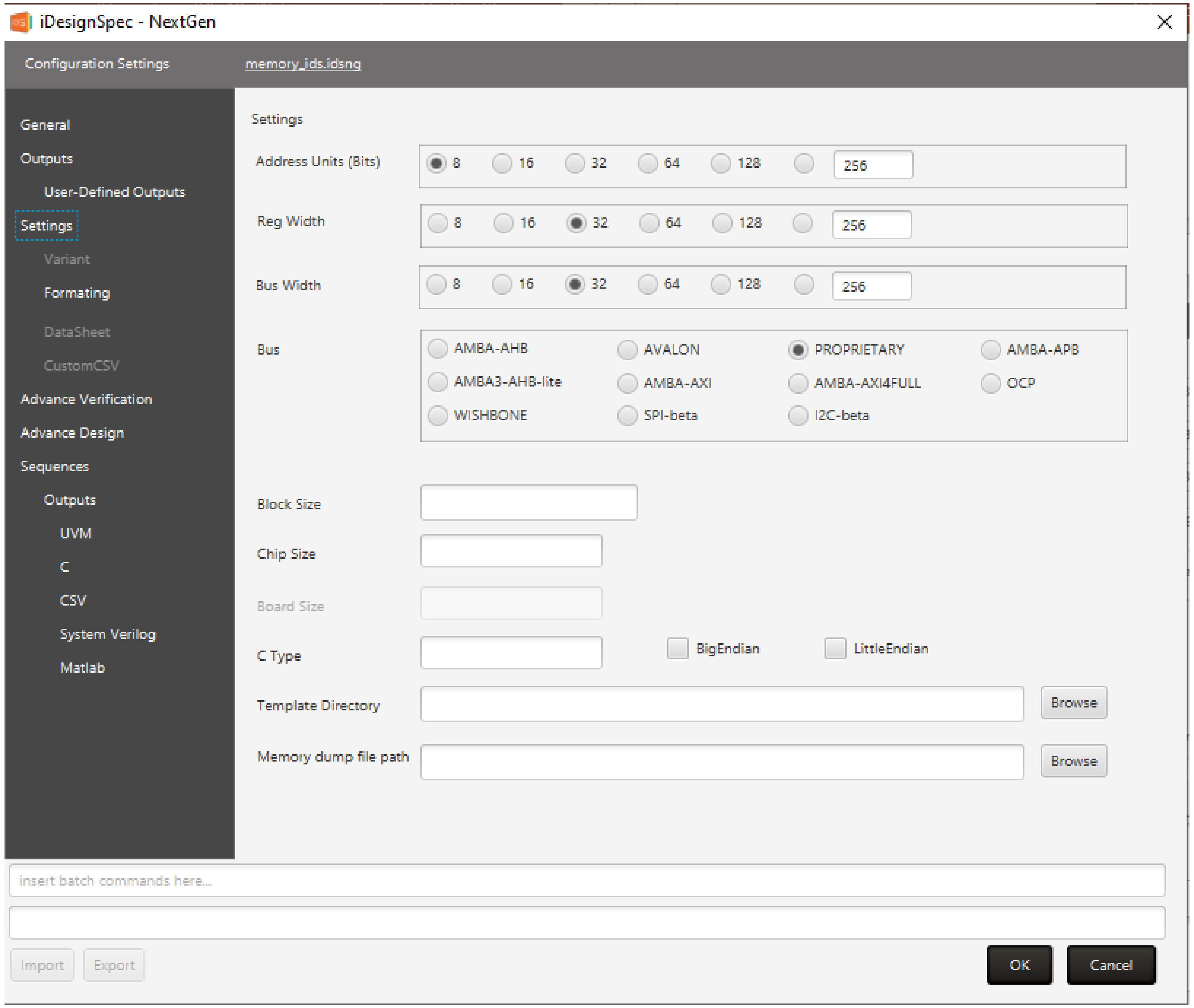
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

THANK YOU

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Bus support in IDesignSpec-NG

- PROPRIETARY
- AMBA
  - AXI, AHB, APB, AXILite, AHBLite
- AVALON
- OCP
- WISHBONE
- I2C
- SPI
- TileLink-1.7 & 1.8



iDesignSpec - NextGen

Configuration Settings   memory_ids.idsng

General
Outputs
User-Defined Outputs
Settings
Variant
Formating
DataSheet
CustomCSV
Advance Verification
Advance Design
Sequences
Outputs
UVM
C
CSV
System Verilog
Matlab

Settings

Address Units (Bits): [✓ 8] 16 32 64 128 256

Reg Width: 8 16 [✓ 32] 64 128 256

Bus Width: 8 16 [✓ 32] 64 128 256

Bus: AMBA-AHB, AVALON, [✓ PROPRIETARY], AMBA-APB, AMBA3-AHB-lite, AMBA-AXI, AMBA-AXI4FULL, OCP, WISHBONE, SPI-beta, I2C-beta

Block Size:

Chip Size:

Board Size:

C Type: BigEndian   LittleEndian

Template Directory: Browse

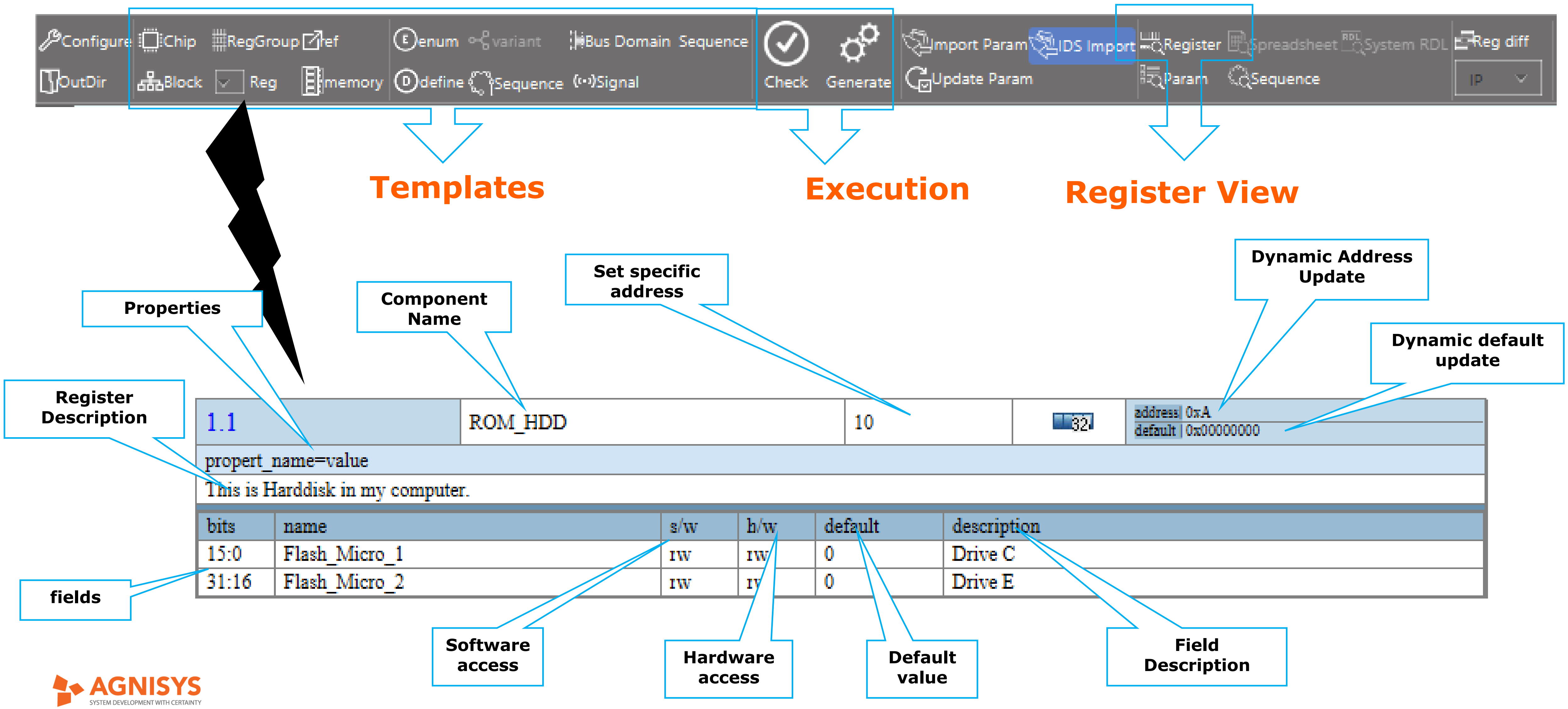
Memory dump file path: Browse

insert batch commands here...

Import   Export   OK   Cancel

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Register view



1.1	ROM_HDD	10	32	address 0xA default 0x00000000

propert_name=value

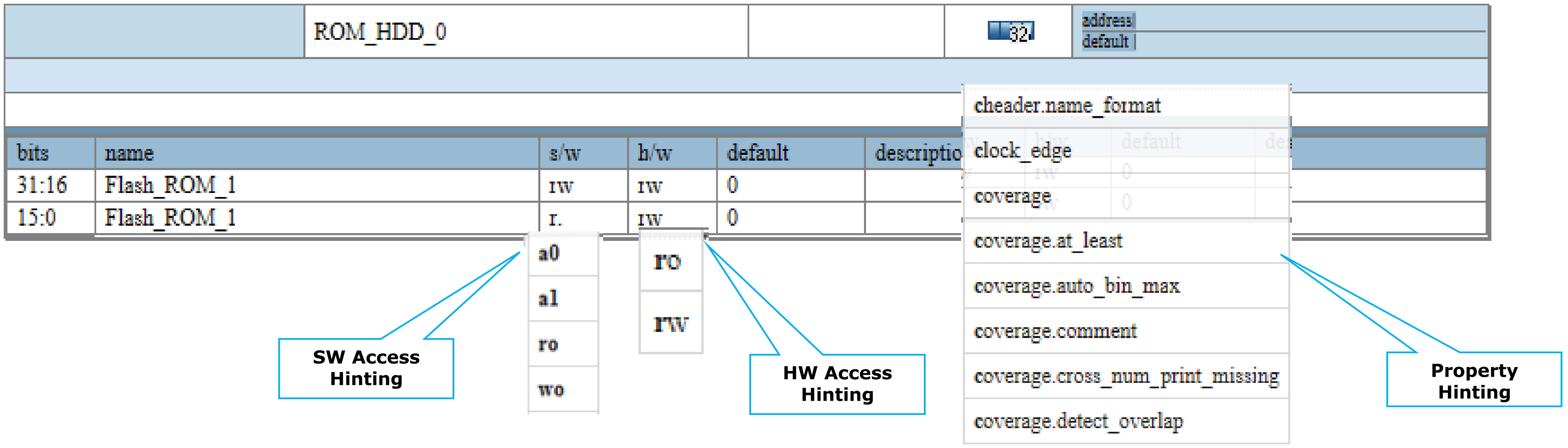
This is Harddisk in my computer.

bits	name	s/w	h/w	default	description
15:0	Flash_Micro_1	rw	rw	0	Drive C
31:16	Flash_Micro_2	rw	rw	0	Drive E

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Register view

- **Additional Features:**
  - SW access hinting
  - HW access hinting
  - Property hinting



ROM_HDD_0 | 32 | address | default

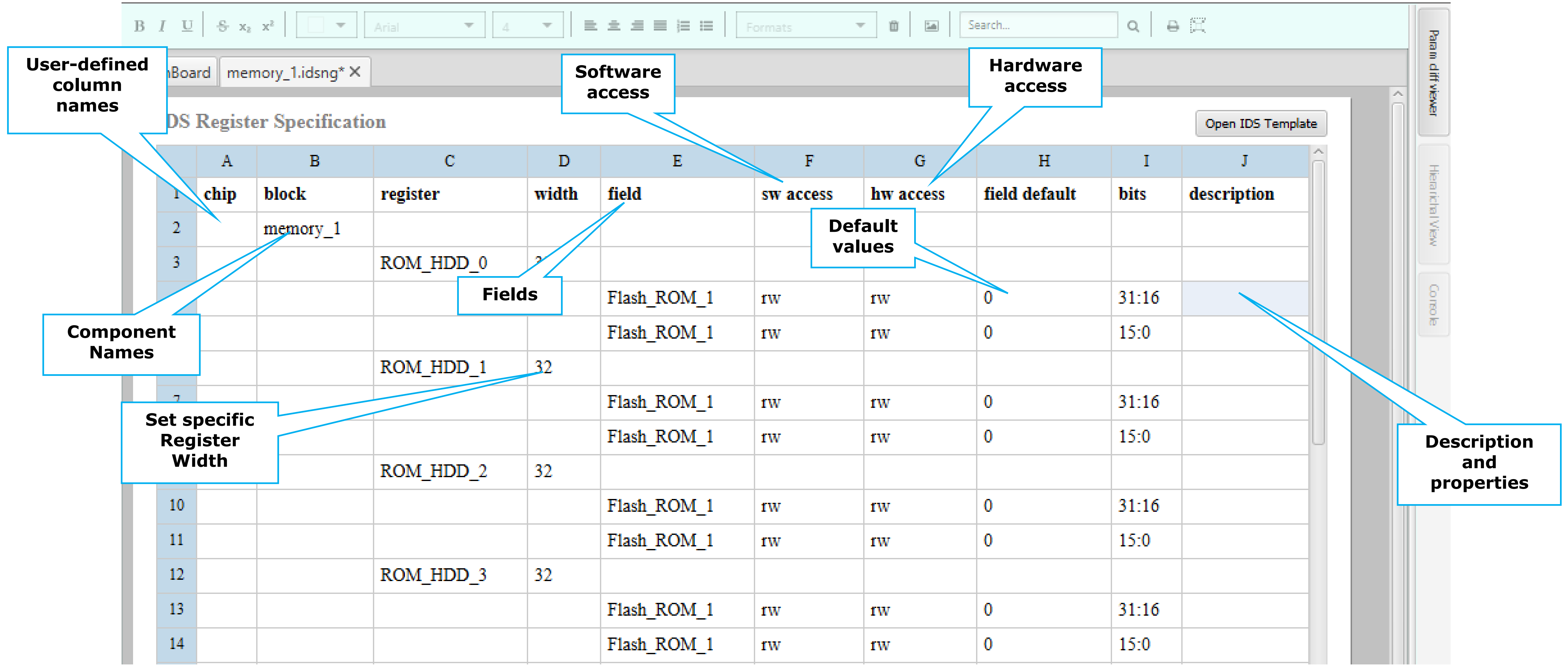
bits	name	s/w	h/w	default	descriptio
31:16	Flash_ROM_1	rw	rw	0	
15:0	Flash_ROM_1	r.	rw	0	

a0
a1
ro
wo

ro
rw

cheader.name_format
clock_edge
coverage
coverage.at_least
coverage.auto_bin_max
coverage.comment
coverage.cross_num_print_missing
coverage.detect_overlap

Spreadsheet view

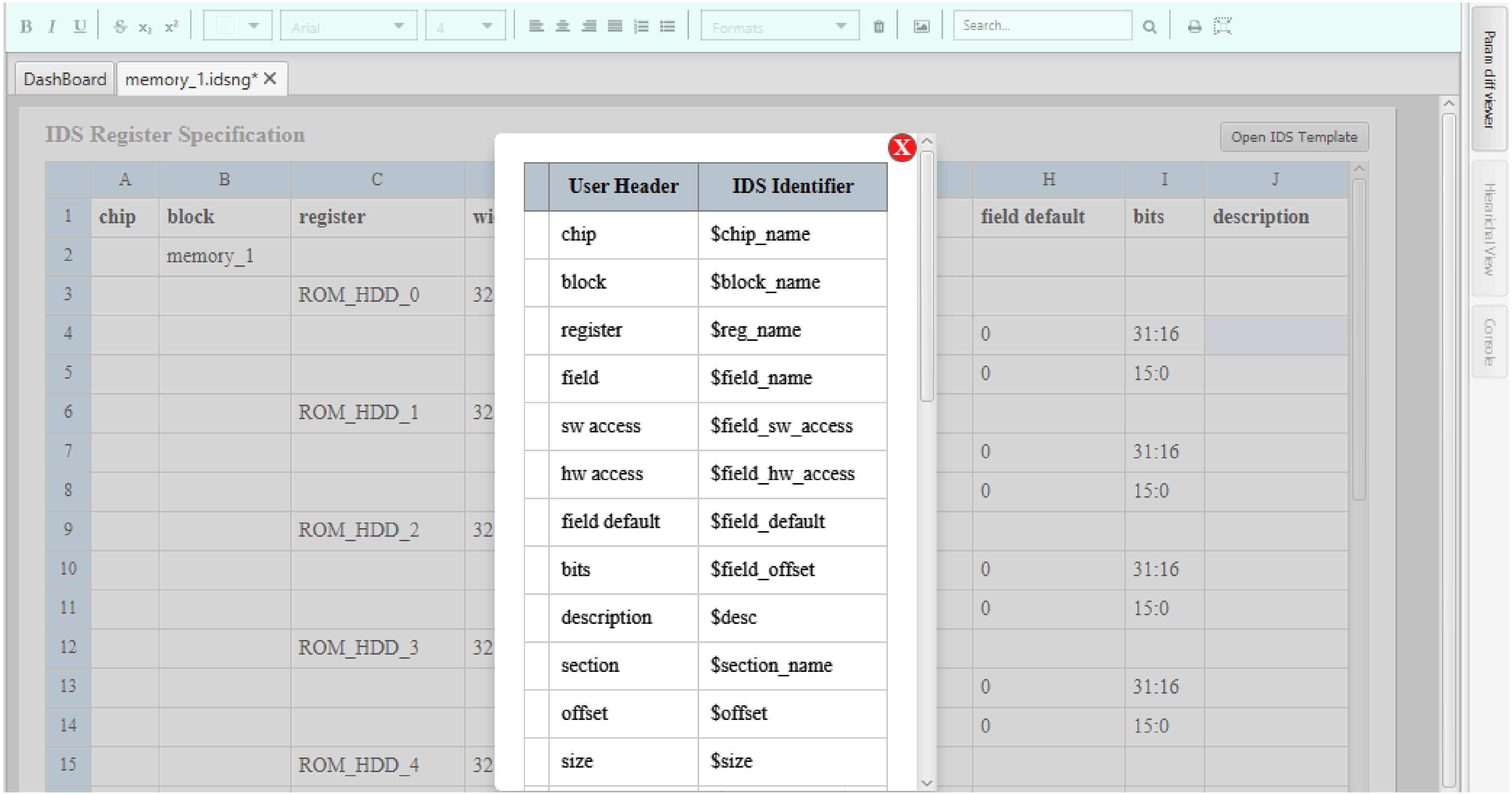


	A	B	C	D	E	F	G	H	I	J
1	chip	block	register	width	field	sw access	hw access	field default	bits	description
2		memory_1								
3			ROM_HDD_0	3						
					Flash_ROM_1	rw	rw	0	31:16	
					Flash_ROM_1	rw	rw	0	15:0	
			ROM_HDD_1	32						
7					Flash_ROM_1	rw	rw	0	31:16	
					Flash_ROM_1	rw	rw	0	15:0	
			ROM_HDD_2	32						
10					Flash_ROM_1	rw	rw	0	31:16	
11					Flash_ROM_1	rw	rw	0	15:0	
12			ROM_HDD_3	32						
13					Flash_ROM_1	rw	rw	0	31:16	
14					Flash_ROM_1	rw	rw	0	15:0	

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Spreadsheet view – IDesignSpec Template

- Maps IDesignSpec columns to user defined columns



IDS Register Specification

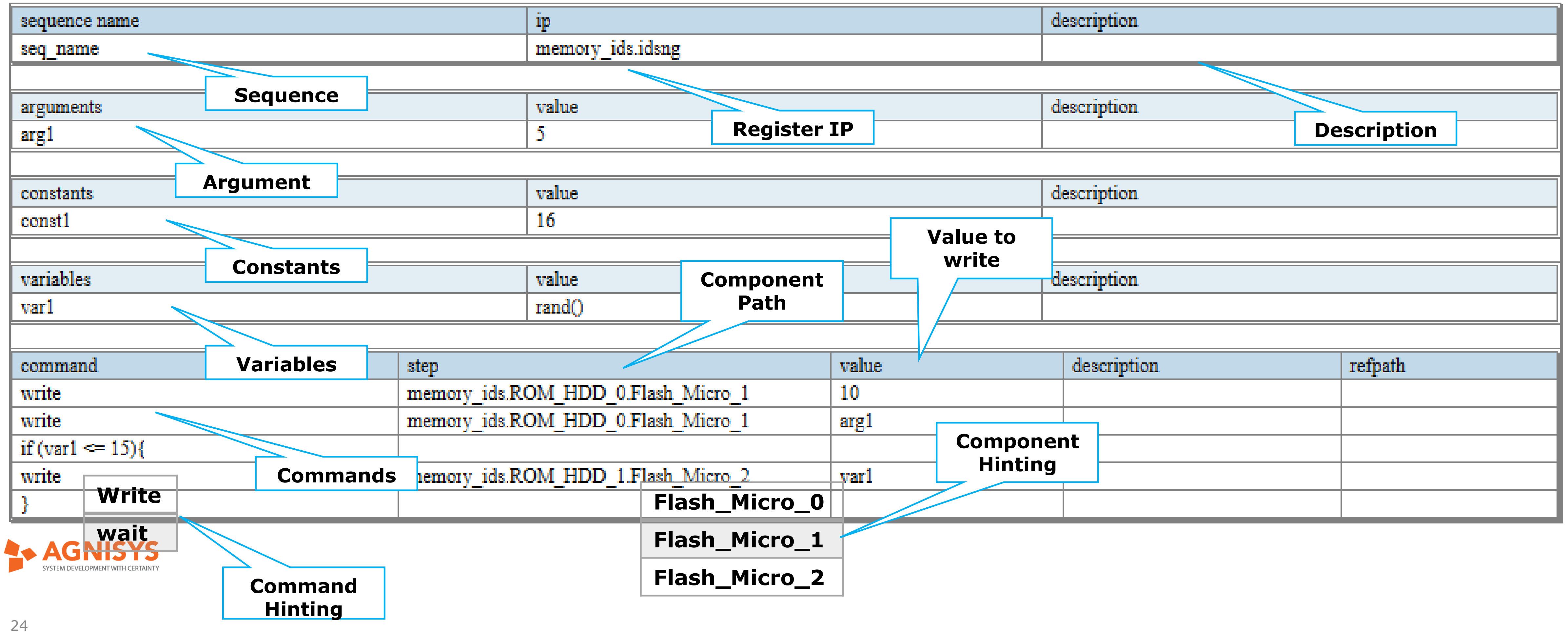
	A	B	C		H	I	J
1	chip	block	register	wi	field default	bits	description
2		memory_1					
3			ROM_HDD_0	32			
4					0	31:16	
5					0	15:0	
6			ROM_HDD_1	32			
7					0	31:16	
8					0	15:0	
9			ROM_HDD_2	32			
10					0	31:16	
11					0	15:0	
12			ROM_HDD_3	32			
13					0	31:16	
14					0	15:0	
15			ROM_HDD_4	32			

User Header	IDS Identifier
chip	$chip_name
block	$block_name
register	$reg_name
field	$field_name
sw access	$field_sw_access
hw access	$field_hw_access
field default	$field_default
bits	$field_offset
description	$desc
section	$section_name
offset	$offset
size	$size

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Sequence view

- User can write multiple sequences
- Hinting provided for commands and steps
- Multiple sequence files for multiple sequences
- Auto fill for IP and sequence name on moving to sequence view



sequence name	ip	description
seq_name	memory_ids.idsng	

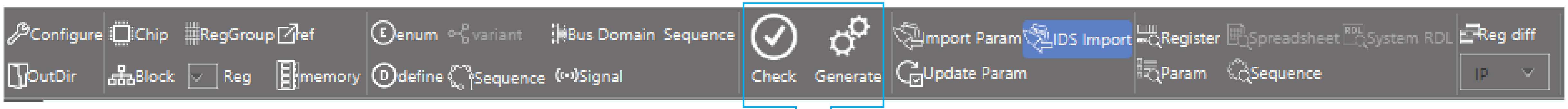
arguments	value	description
arg1	5	

constants	value	description
const1	16	

variables	value	description
var1	rand()	

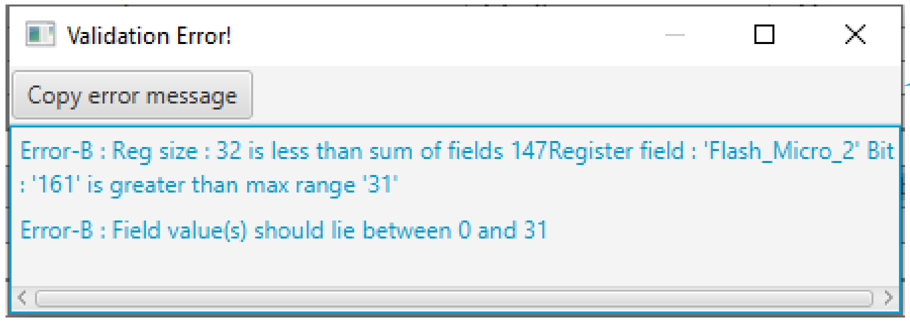
command	step	value	description	refpath
write	memory_ids.ROM_HDD_0.Flash_Micro_1	10		
write	memory_ids.ROM_HDD_0.Flash_Micro_1	arg1		
if (var1 <= 15){				
write	memory_ids.ROM_HDD_1.Flash_Micro_2	var1		
}				

Check and Generate



Execution

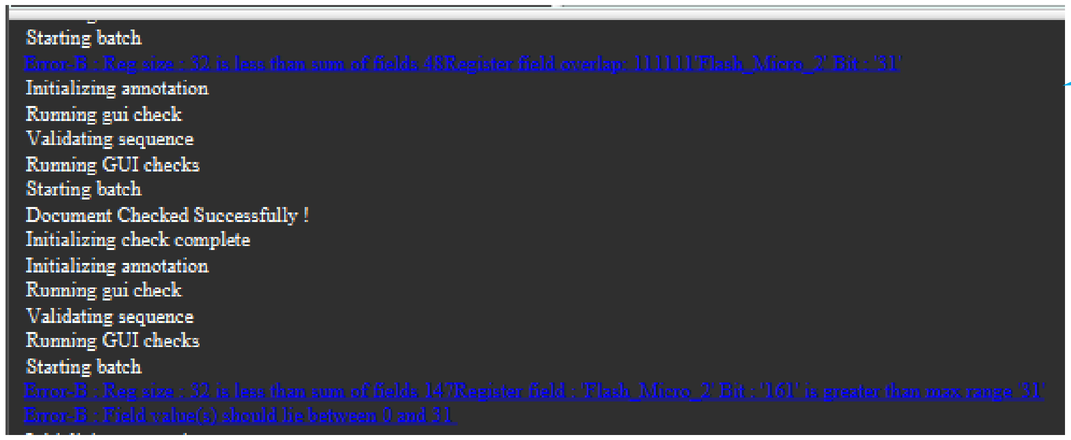
- Address Calculation & back-annotation on the spec
- Inserts Register Map – Table of Content (address, default values , …)
- Consistency checks and Error Display
  - Detects overlaps
    - Bit
    - Register
    - Regroup
    - Block
  - Incomplete data
  - Bad/incorrect data
    - Duplicate / illegal names
    - Invalid access, incompatible access



Error-B : Reg size : 32 is less than sum of fields 147Register field : 'Flash_Micro_2' Bit : '161' is greater than max range '31'

Error-B : Field value(s) should lie between 0 and 31

Error Window



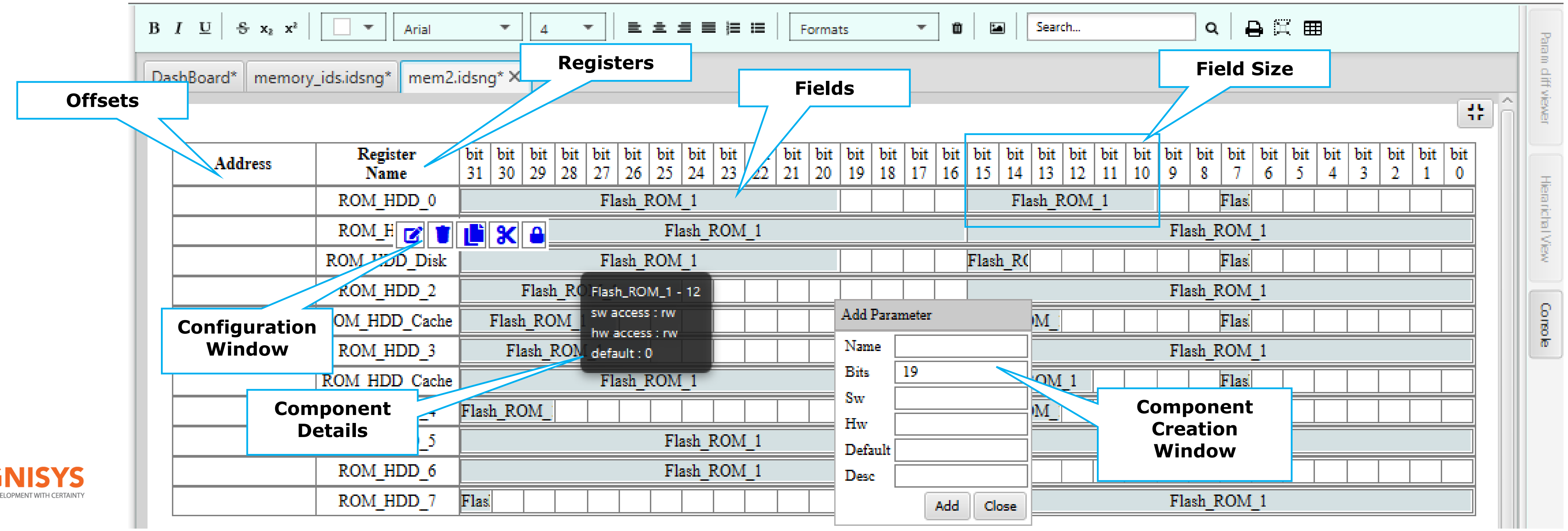
```
Starting batch
Error-B : Reg size : 32 is less than sum of fields 48Register field overlap: 111111'Flash_Micro_2' Bit : '31'
Initializing annotation
Running gui check
Validating sequence
Running GUI checks
Starting batch
Document Checked Successfully !
Initializing check complete
Initializing annotation
Running gui check
Validating sequence
Running GUI checks
Starting batch
Error-B : Reg size : 32 is less than sum of fields 147Register field : 'Flash_Micro_2' Bit : '161' is greater than max range '31'
Error-B : Field value(s) should lie between 0 and 31
```

Console Window

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Param View

- Lock specific parameter
- Import YAML
- Edit, delete and add reg through buttons
- Param diff viewer can show previous changes
- Deleted reg can be seen in updated param
- User can add and delete register at a time



Address	Register Name	bit 31 … bit 0
	ROM_HDD_0	Flash_ROM_1 · Flash_ROM_1 · Flas
	ROM_H	Flash_ROM_1 · Flash_ROM_1
	ROM_HDD_Disk	Flash_ROM_1 · Flash_R(· Flas
	ROM_HDD_2	Flash_RO · Flash_ROM_1
	OM_HDD_Cache	Flash_ROM · M_ · Flas
	ROM_HDD_3	Flash_ROM · Flash_ROM_1
	ROM_HDD_Cache	Flash_ROM_1 · OM_1 · Flas
	4	Flash_ROM_ · M_
	5	Flash_ROM_1
	ROM_HDD_6	Flash_ROM_1
	ROM_HDD_7	Flas · Flash_ROM_1

Flash_ROM_1 - 12
sw access : rw
hw access : rw
default : 0

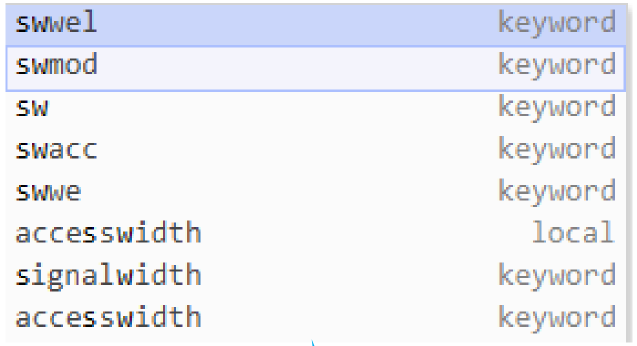
Add Parameter

Name	
Bits	19
Sw	
Hw	
Default	
Desc	

Add Close

SystemRDL 2.0 Editor

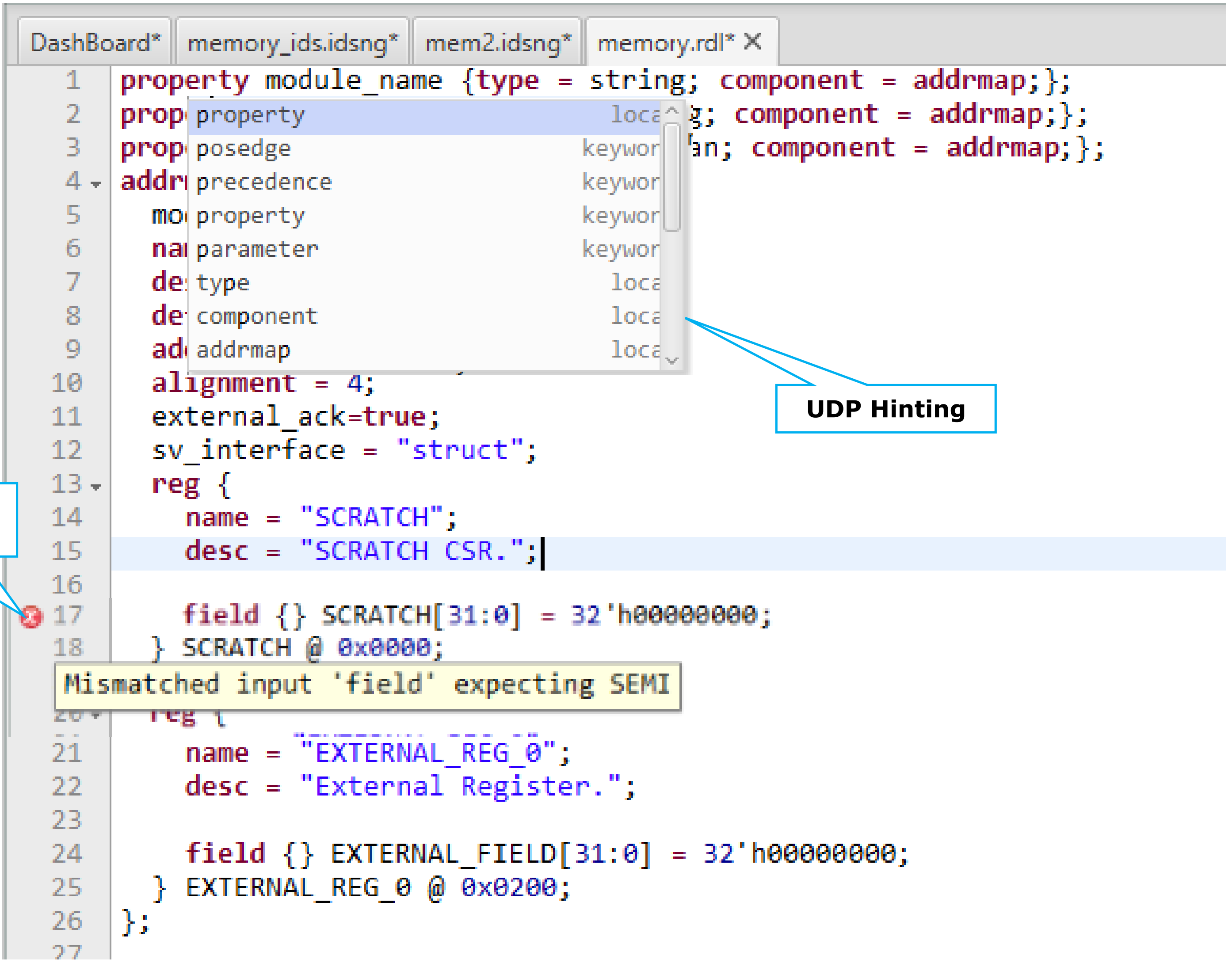
- Automatic Indentation and Colorful text
- Properties and UDP hinting
- Error detection on wrong syntax
- Identify incorrect code



swwel	keyword
swmod	keyword
sw	keyword
swacc	keyword
swwe	keyword
accesswidth	local
signalwidth	keyword
accesswidth	keyword

Property Hinting

Syntax Error Hint



```
property module_name {type = string; component = addrmap;};
prop                                 g; component = addrmap;};
prop                                 an; component = addrmap;};
addr
  mo
  na
  de
  de
  ad
  alignment = 4;
  external_ack=true;
  sv_interface = "struct";
  reg {
    name = "SCRATCH";
    desc = "SCRATCH CSR.";

    field {} SCRATCH[31:0] = 32'h00000000;
  } SCRATCH @ 0x0000;
```

property	loca
posedge	keywor
precedence	keywor
property	keywor
parameter	keywor
type	loca
component	loca
addrmap	loca

```
Mismatched input 'field' expecting SEMI
```

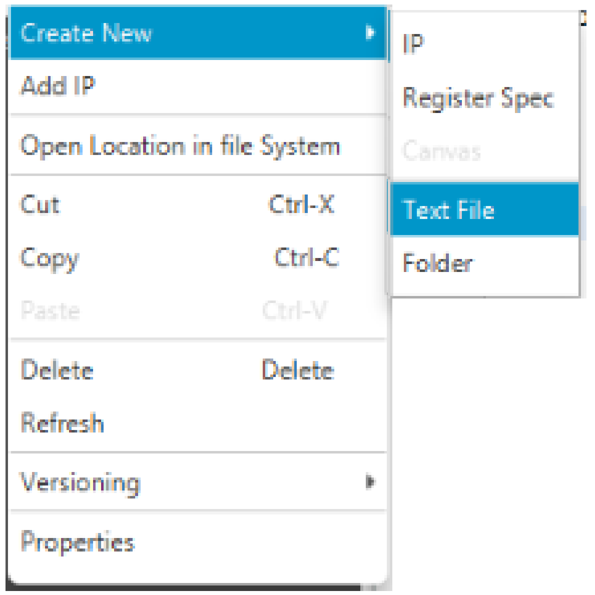
```
  reg {
    name = "EXTERNAL_REG_0";
    desc = "External Register.";

    field {} EXTERNAL_FIELD[31:0] = 32'h00000000;
  } EXTERNAL_REG_0 @ 0x0200;
};
```

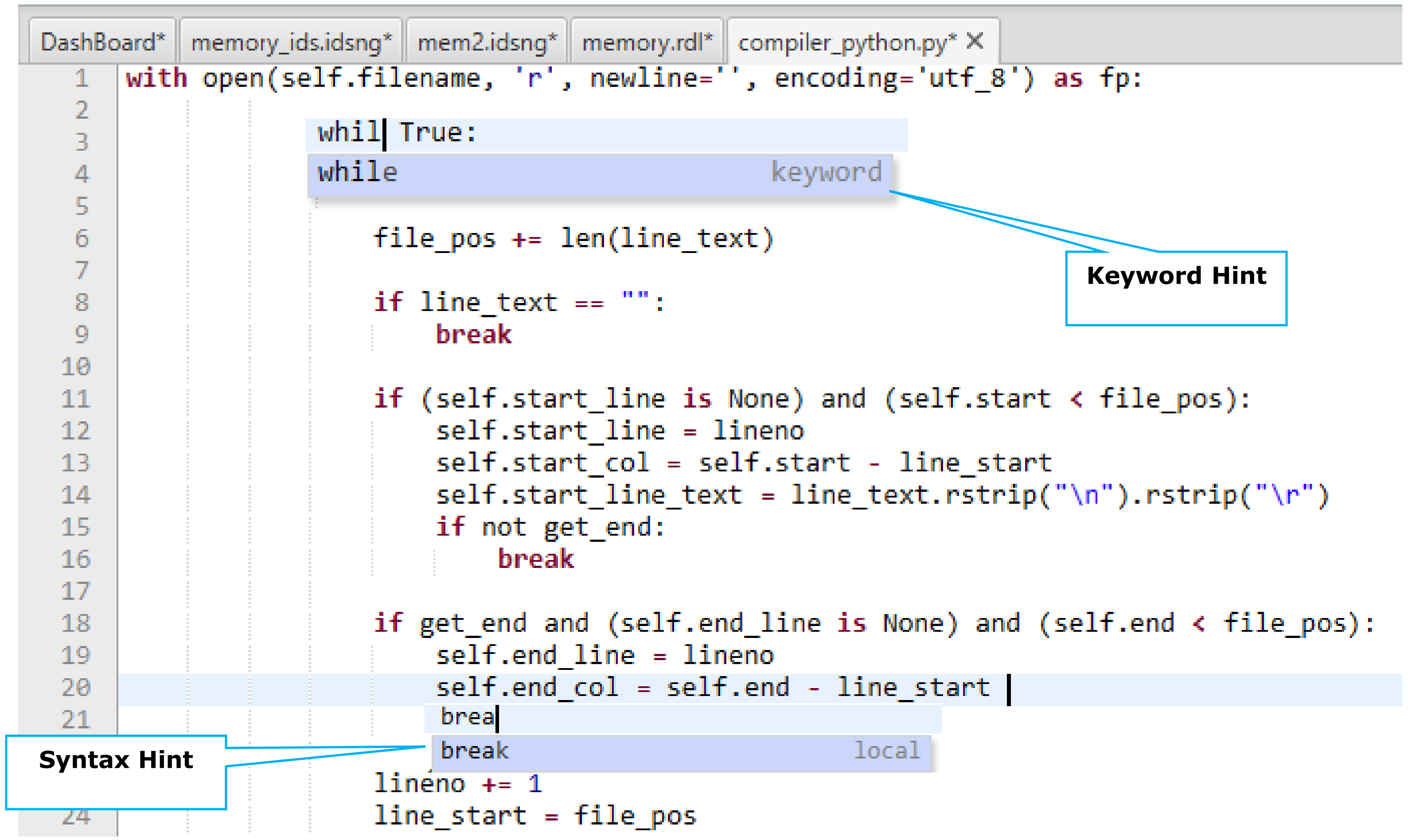
UDP Hinting

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Python editor

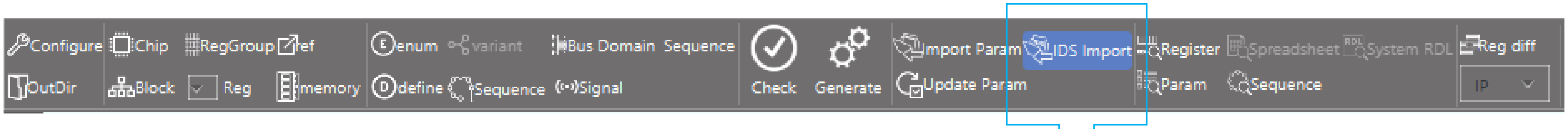
- Automatic Indentation
- Colorful text
- Syntax hinting
- Error detection when incorrect syntax is used
- Identifies incorrect code



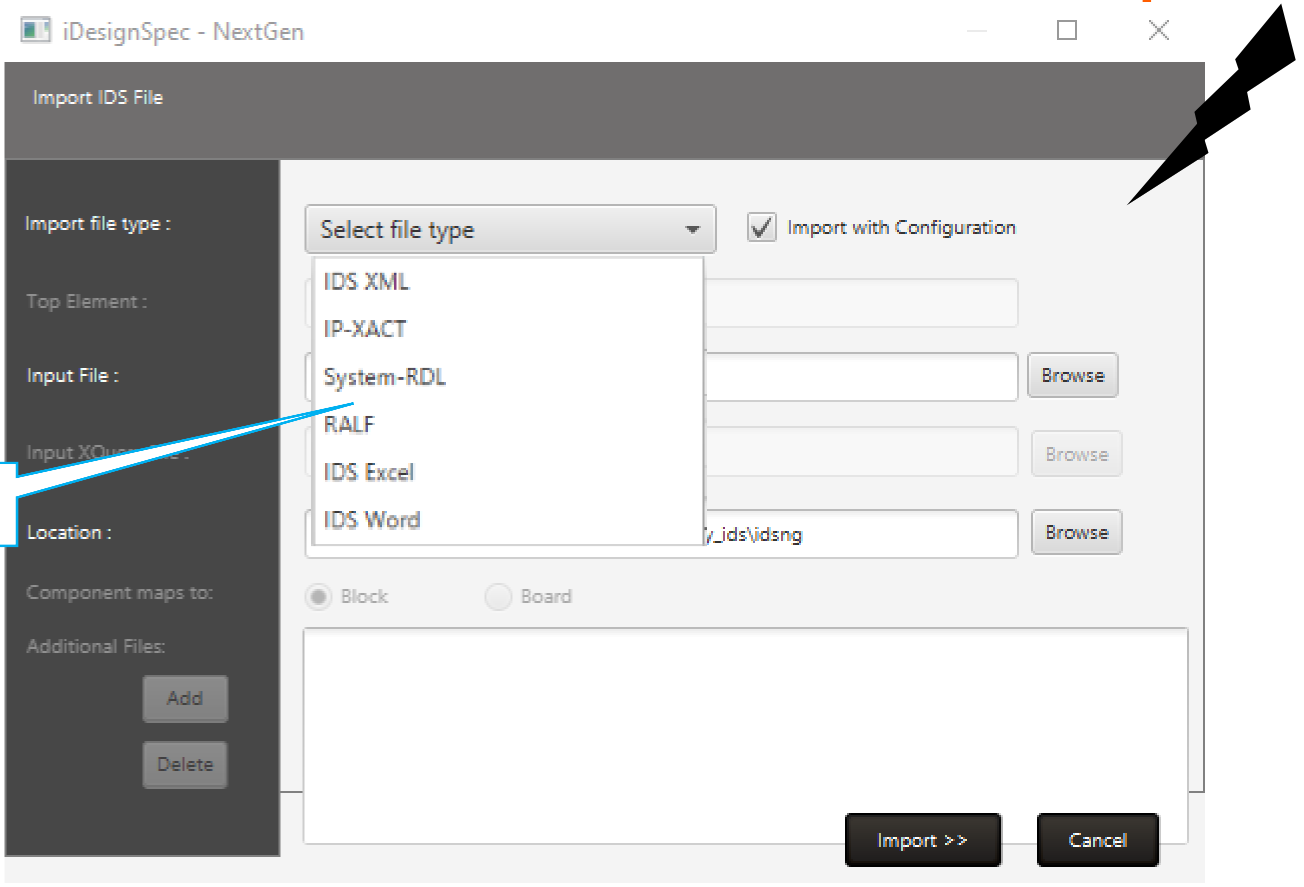


```
with open(self.filename, 'r', newline='', encoding='utf_8') as fp:

            whil True:
            while                                   keyword

                file_pos += len(line_text)

                if line_text == "":
                    break

                if (self.start_line is None) and (self.start < file_pos):
                    self.start_line = lineno
                    self.start_col = self.start - line_start
                    self.start_line_text = line_text.rstrip("\n").rstrip("\r")
                    if not get_end:
                        break

                if get_end and (self.end_line is None) and (self.end < file_pos):
                    self.end_line = lineno
                    self.end_col = self.end - line_start
                    brea
                    break                               local
                lineno += 1
                line_start = file_pos
```

Import IDS input formats



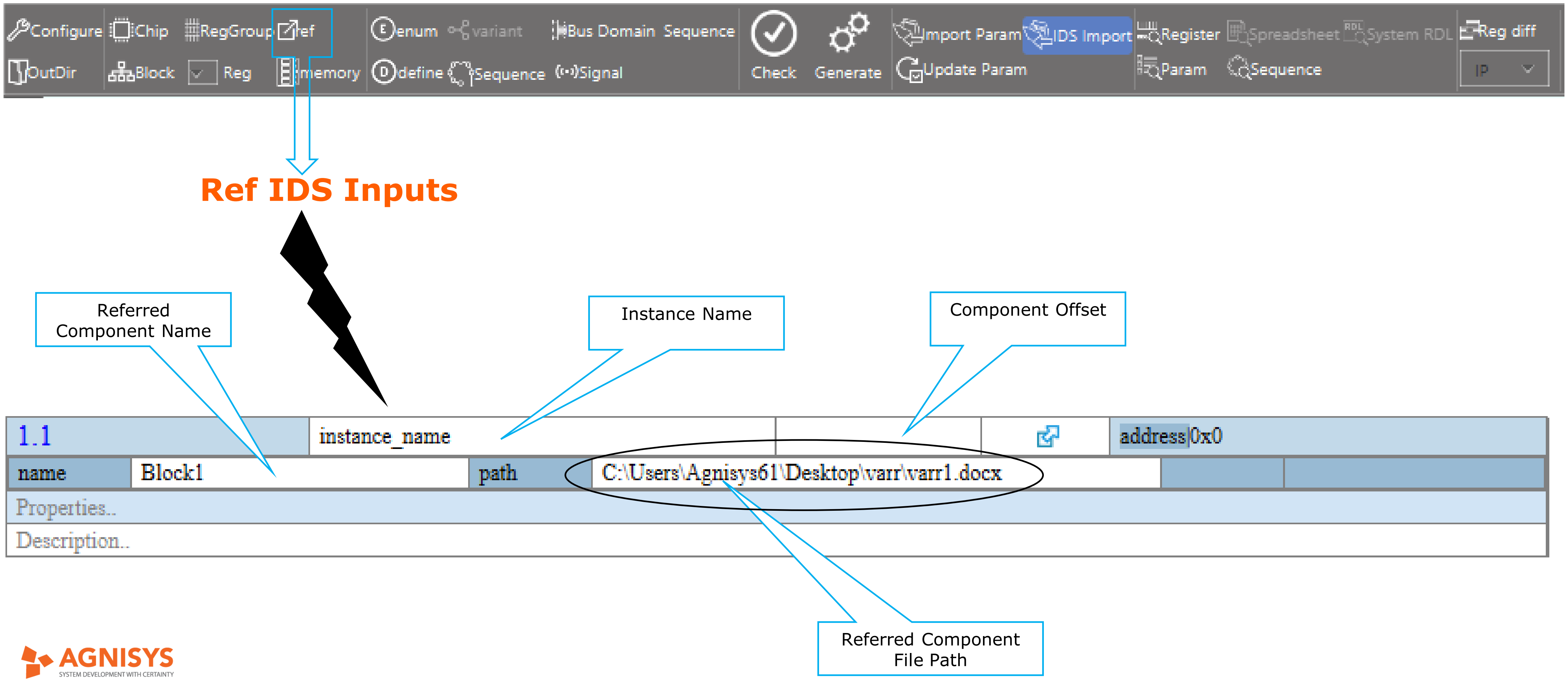
Import IDS Inputs



Input formats

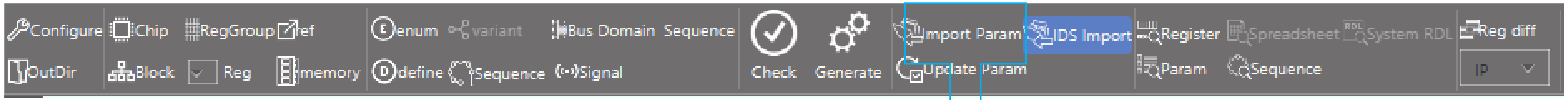
- IDS XML
- IP-XACT
- System-RDL
- RALF
- IDS Excel
- IDS Word

Ref IDS Input Formats

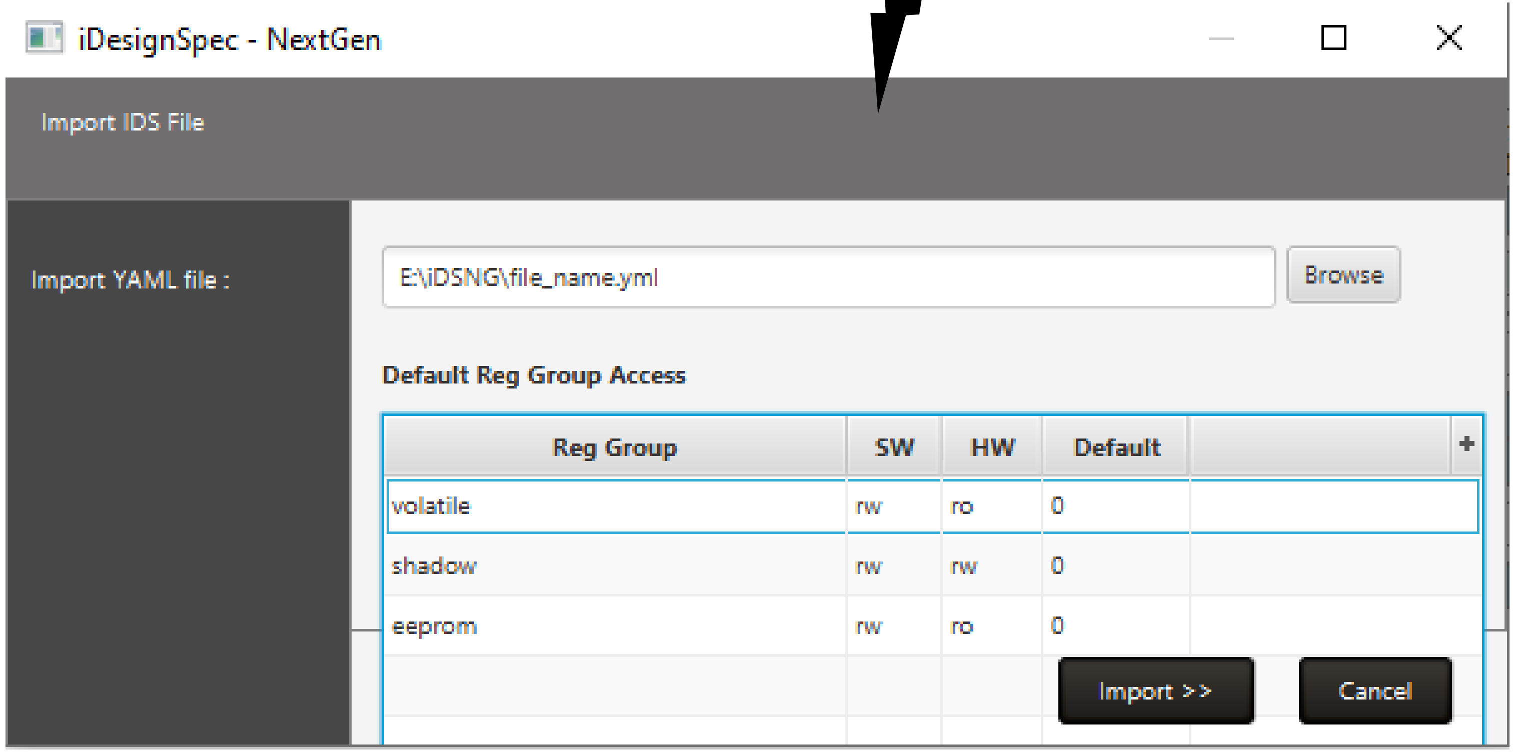


1.1	instance_name				address 0x0	
name	Block1	path	C:\Users\Agnisys61\Desktop\varr\varr1.docx			
Properties..						
Description..						

YAML Import for Param View



Import Param



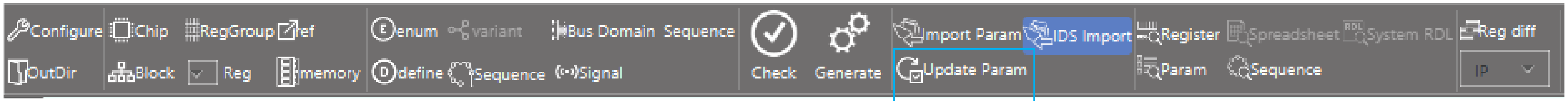
Import YAML file : E:\iDSNG\file_name.yml

Default Reg Group Access

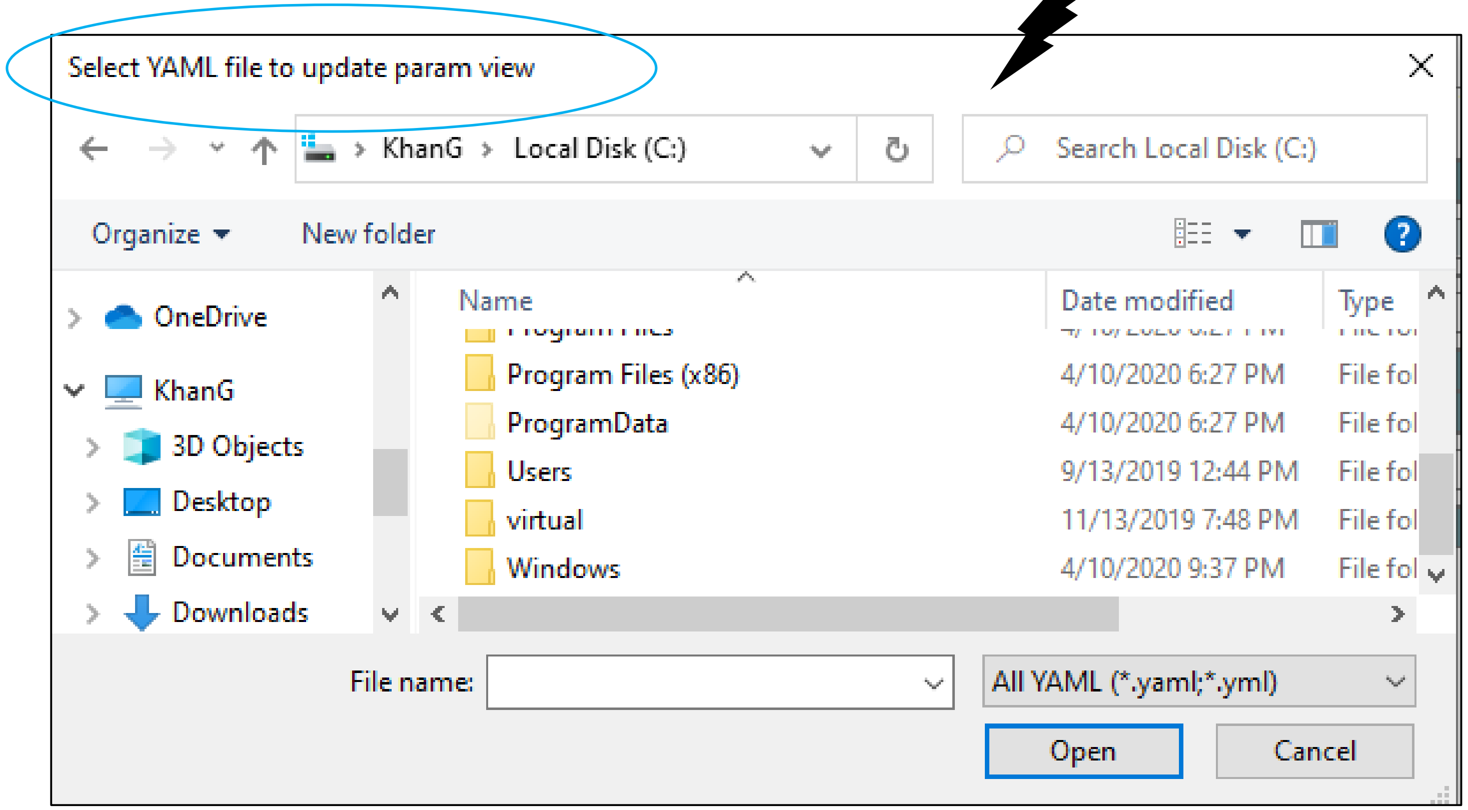
Reg Group	SW	HW	Default	
volatile	rw	ro	0	
shadow	rw	rw	0	
eeprom	rw	ro	0	

AGNISYS SYSTEM DEVELOPMENT WITH CERTAINTY

YAML update for Param view



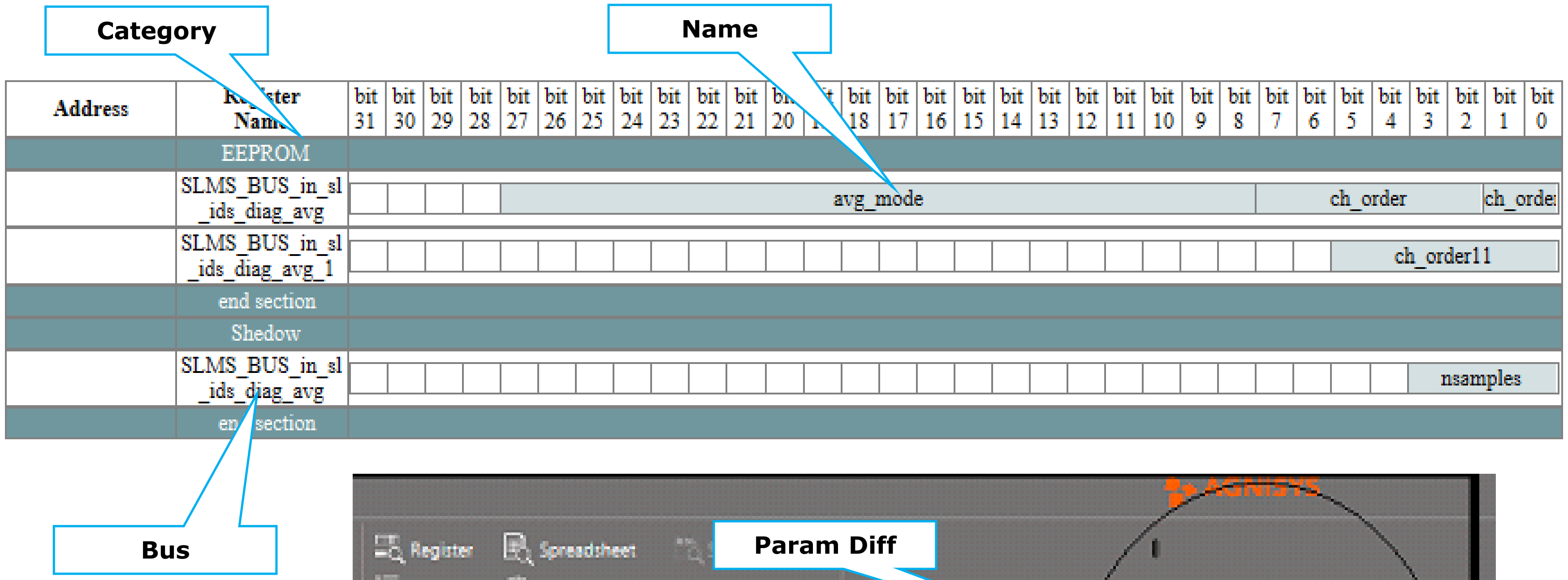
Update Param



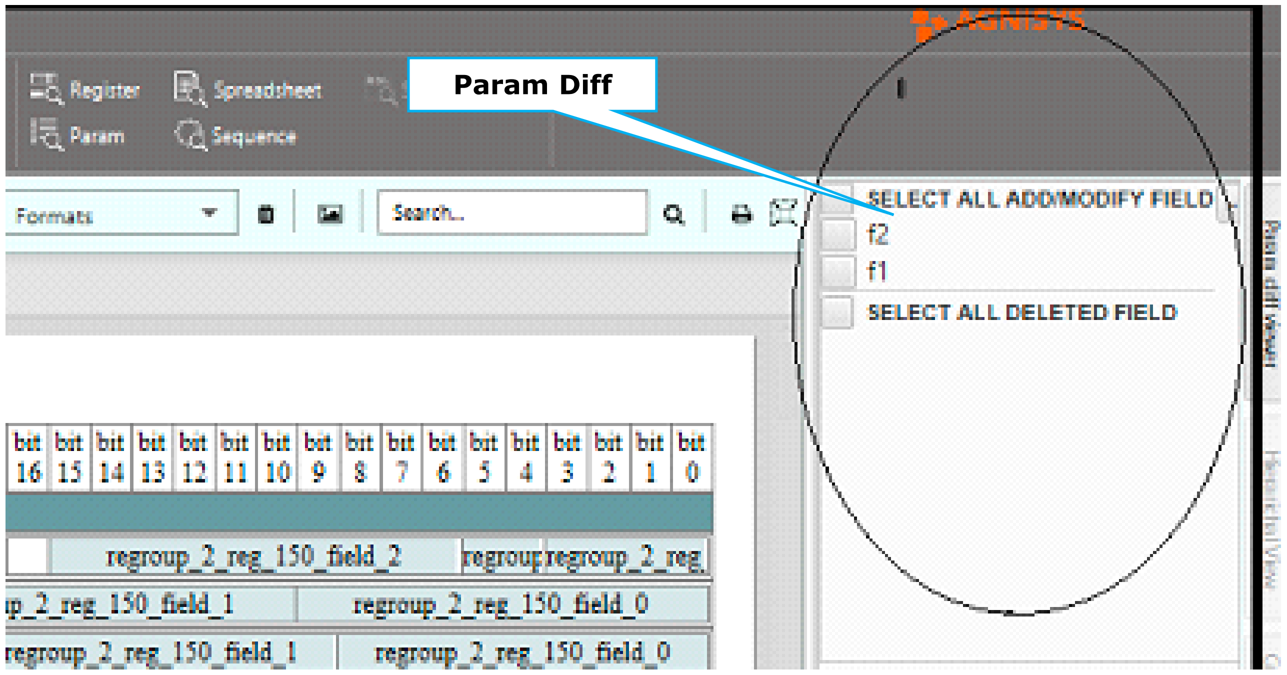
Name	Date modified	Type
Program Files (x86)	4/10/2020 6:27 PM	File fol
ProgramData	4/10/2020 6:27 PM	File fol
Users	9/13/2019 12:44 PM	File fol
virtual	11/13/2019 7:48 PM	File fol
Windows	4/10/2020 9:37 PM	File fol

YAML code for Param view

```
fields :
    -
        Name : avg_mode
        Format : fixdt(0,20,0)
        Category : EEPROM
        Default : 0
        Bus : SLMS_BUS_in_sl_ids_diag_avg
        Description : "   sdsadasdasdasdas   "
    -
        Name : ch_order
        Format : fixdt(0,6,0)
        Category : EEPROM
        Default : 0
        Bus : SLMS_BUS_in_sl_ids_diag_avg
        Description : "  asdasdasd   "
    -
        Name : ch_order111
        Format : fixdt(0,2,0)
        Category : EEPROM
        Default : 0
        Bus : SLMS_BUS_in_sl_ids_diag_avg
        Description : "  asdasdasd   "
    -
        Name : ch_order11
        Format : fixdt(0,6,0)
        Category : EEPROM
        Default : 0
        Bus : SLMS_BUS_in_sl_ids_diag_avg
        Description : "  asdsadasd"
```


Category
Name
Bus

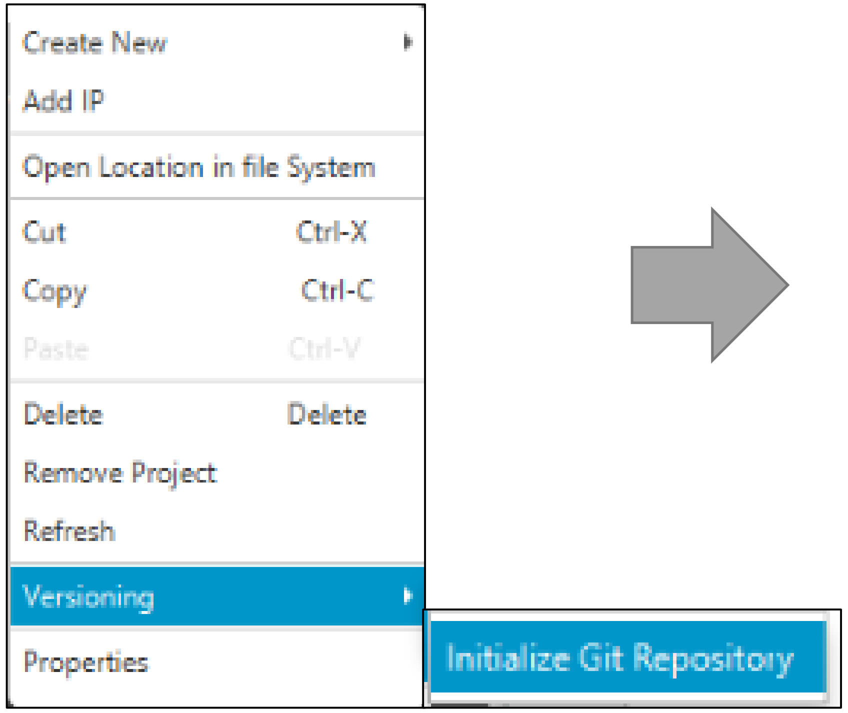
Address	Register Name	bit 31	bit 30	bit 29	bit 28	bit 27	bit 26	bit 25	bit 24	bit 23	bit 22	bit 21	bit 20	bit 19	bit 18	bit 17	bit 16	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	EEPROM																																
	SLMS_BUS_in_sl_ids_diag_avg					avg_mode																				ch_order						ch_orde	
	SLMS_BUS_in_sl_ids_diag_avg_1																											ch_order11					
	end section																																
	Shedow																																
	SLMS_BUS_in_sl_ids_diag_avg																													nsamples			
	end section																																

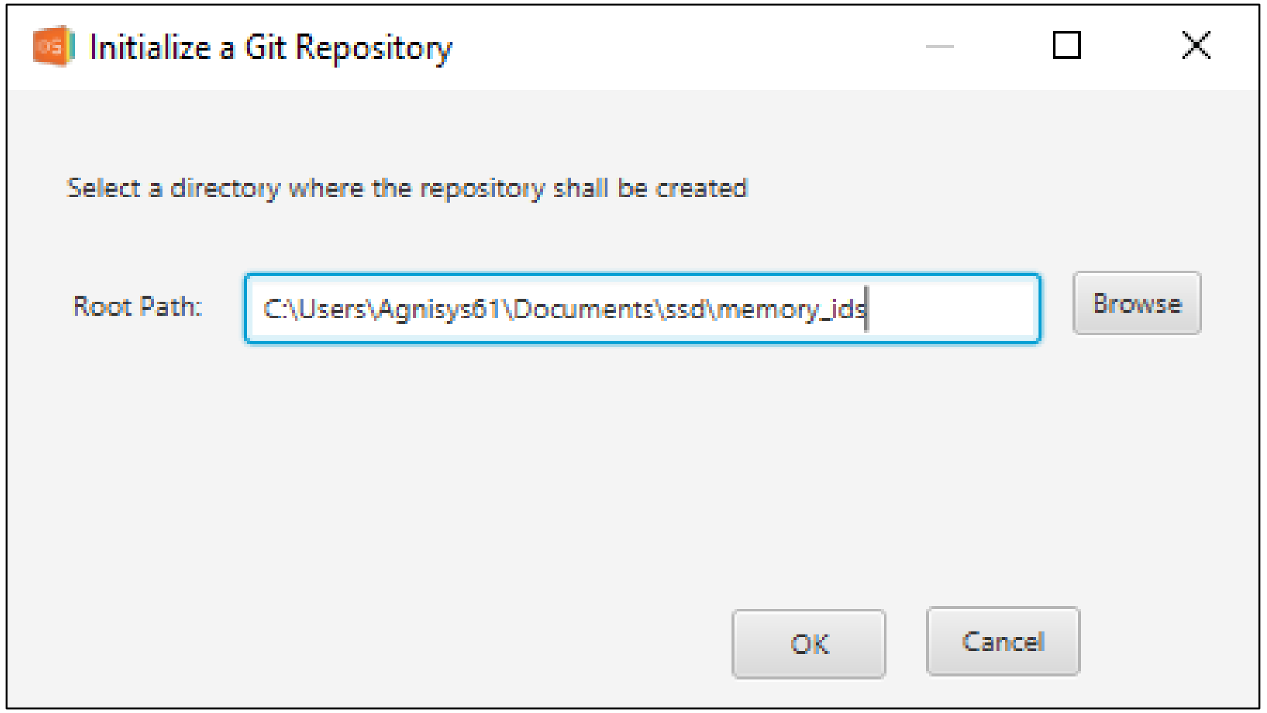

AGNISYS
Register
Spreadsheet
Param
Sequence
Param Diff
Formats
Search...
SELECT ALL ADD/MODIFY FIELD
f2
f1
SELECT ALL DELETED FIELD
Param diff viewer
bit 16 bit 15 bit 14 bit 13 bit 12 bit 11 bit 10 bit 9 bit 8 bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0
regroup_2_reg_150_field_2
regroup_2_reg_150_field_1
regroup_2_reg_150_field_0

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Git integration

- Git integration is added for tracking the changes and modifications done in the specification
- It is used for speeding up the process, data integrity and support work-flow
- User can commit and discard the changes
- Branch creation and deletion
- Branch can be merged of pull from remote repository

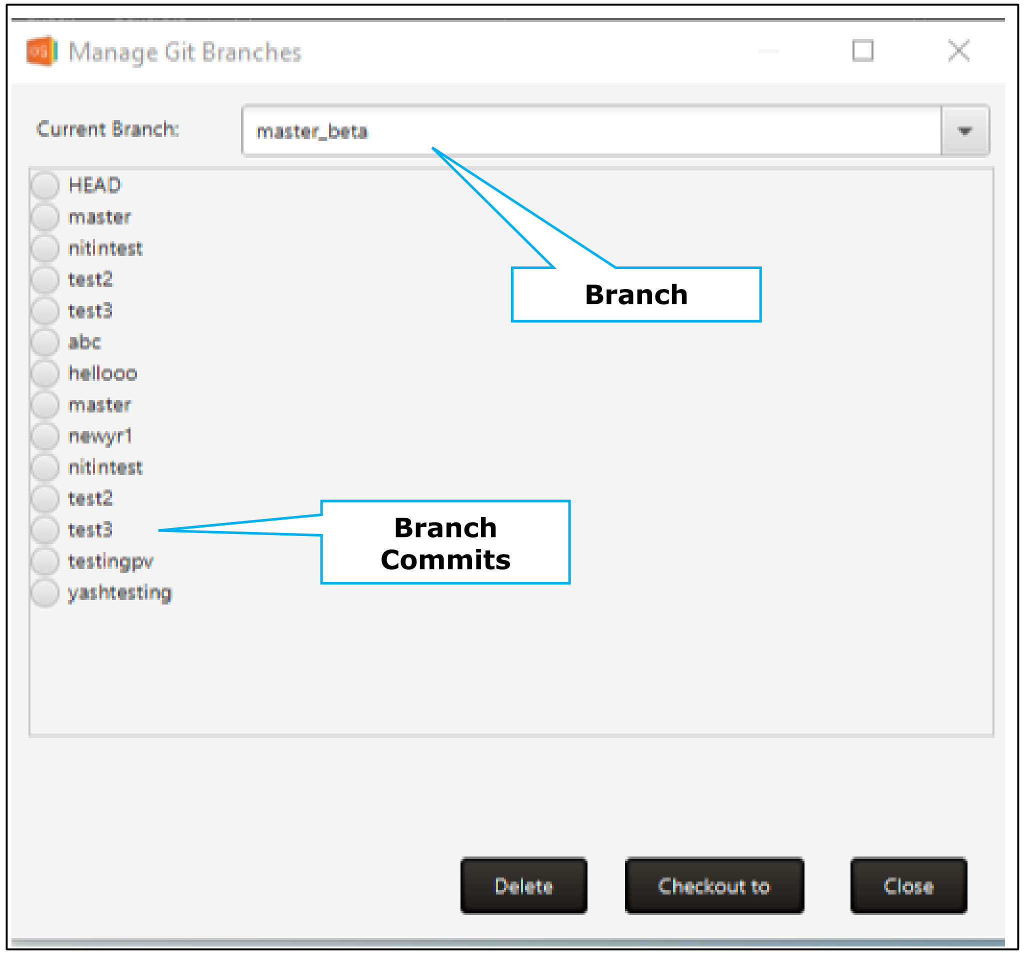




Select a directory where the repository shall be created

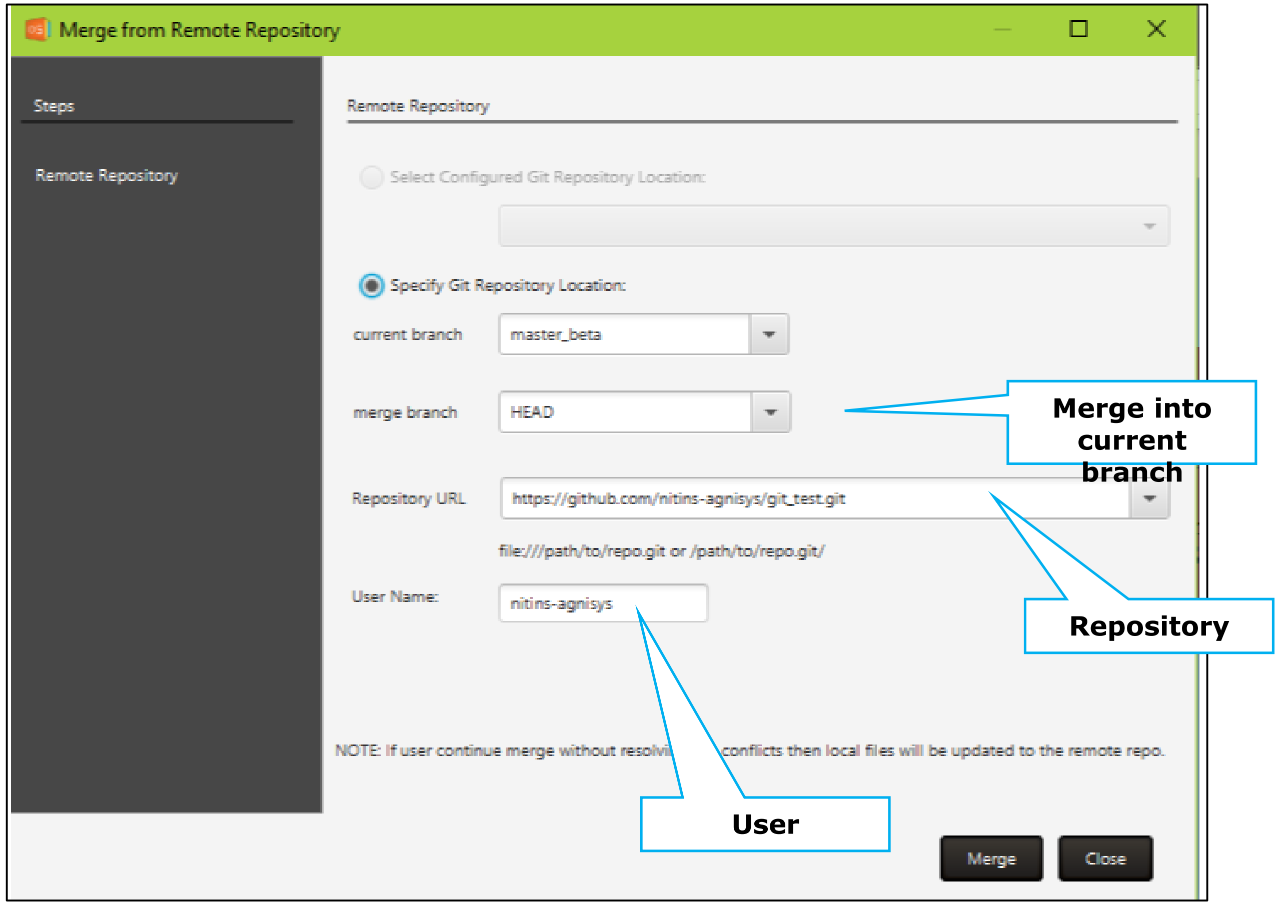
Root Path: C:\Users\Agnisys61\Documents\ssd\memory_ids

Manage Git branches



Current Branch: master_beta

- HEAD
- master
- nitintest
- test2
- test3
- abc
- hellooo
- master
- newyr1
- nitintest
- test2
- test3
- testingpv
- yashtesting



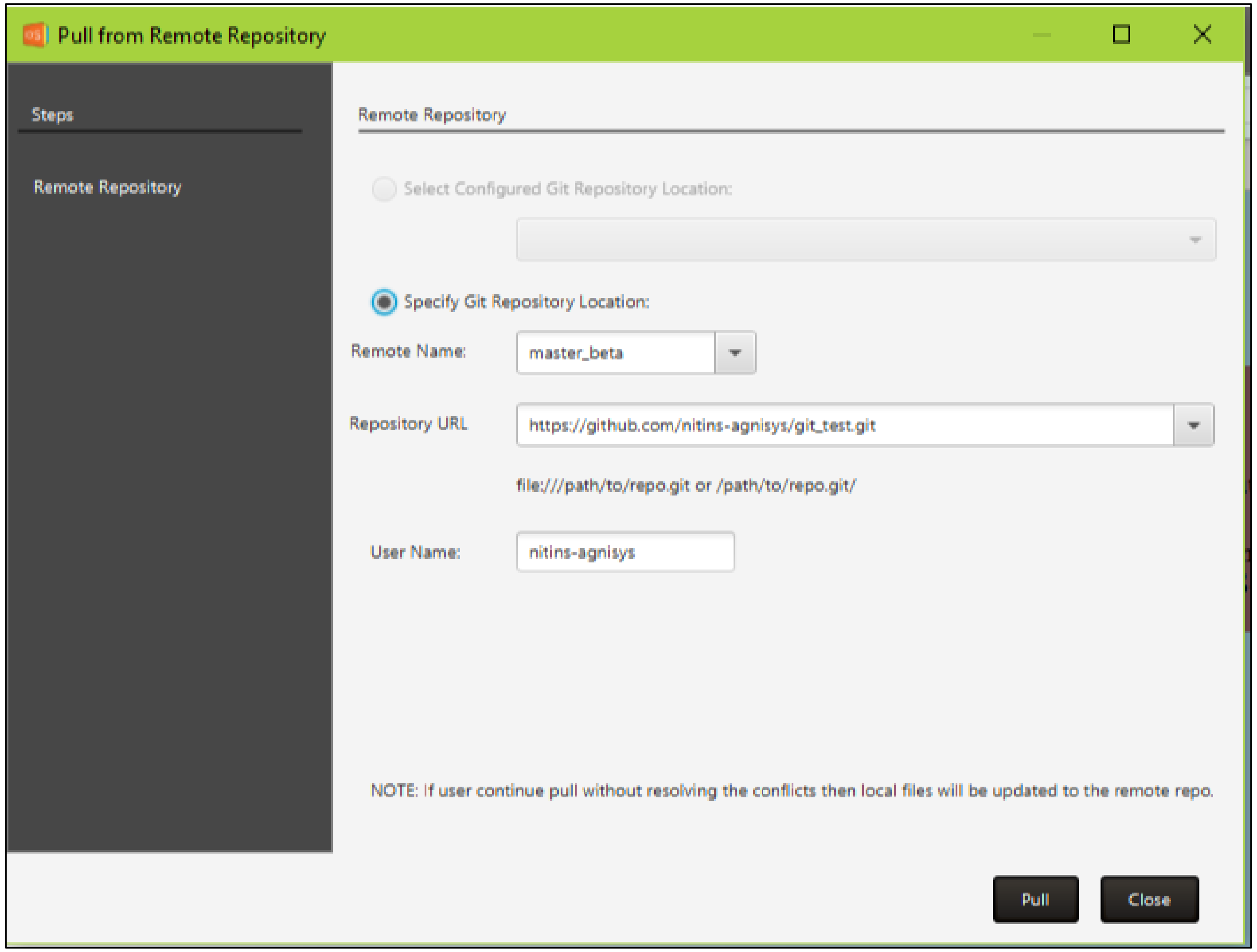
Select Configured Git Repository Location:

[✓ Specify Git Repository Location:]

current branch	master_beta
merge branch	HEAD
Repository URL	https://github.com/nitins-agnisys/git_test.git
	file:///path/to/repo.git or /path/to/repo.git/
User Name:	nitins-agnisys

NOTE: If user continue merge without resolvi[illegible] conflicts then local files will be updated to the remote repo.

Pull current branch



Remote Repository

Select Configured Git Repository Location:

[✓ Specify Git Repository Location:]

Remote Name: master_beta

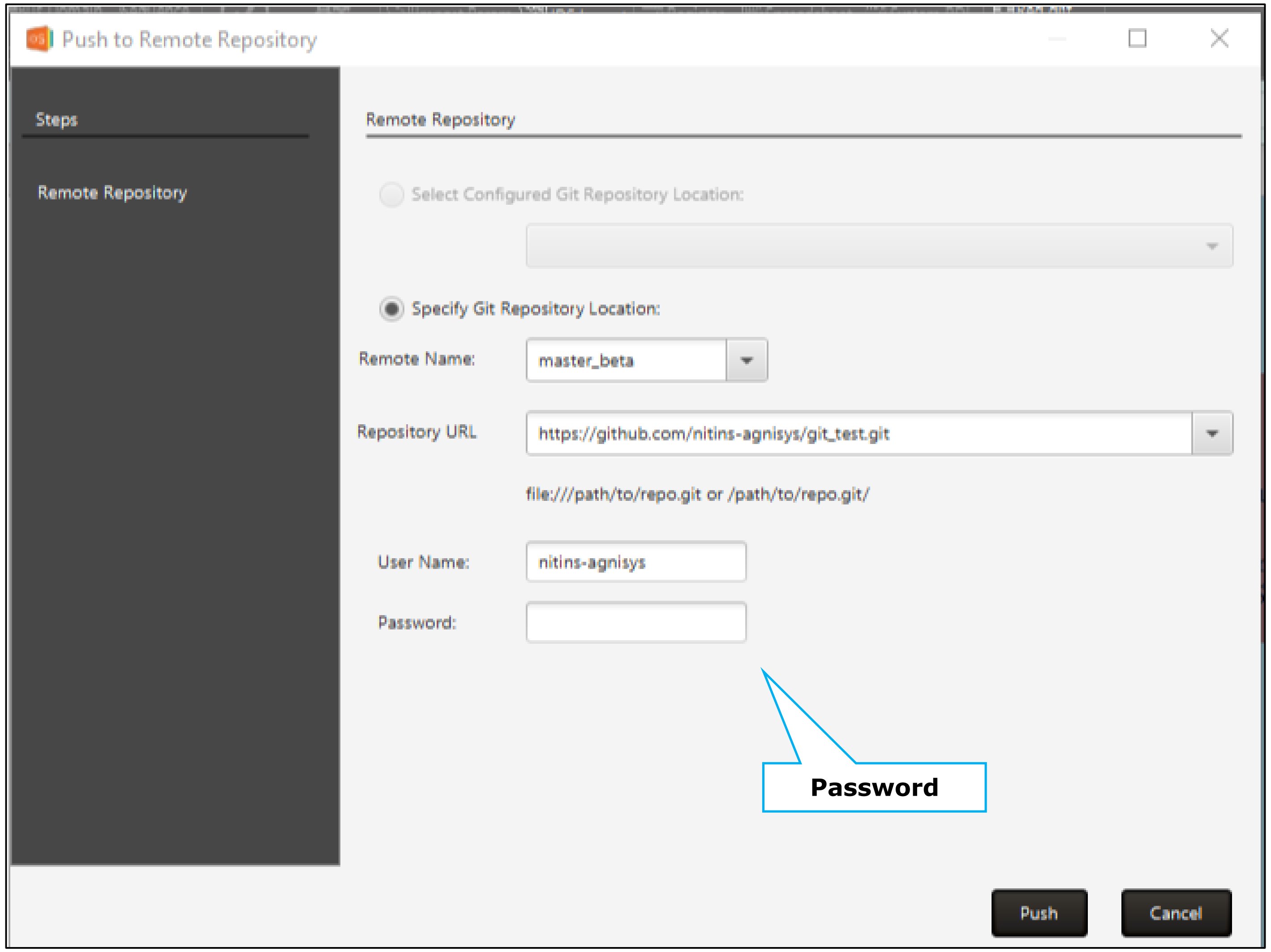
Repository URL https://github.com/nitins-agnisys/git_test.git

file:///path/to/repo.git or /path/to/repo.git/

User Name: nitins-agnisys

NOTE: If user continue pull without resolving the conflicts then local files will be updated to the remote repo.

Push into Current Branch


Push to Remote Repository

Steps

Remote Repository

Remote Repository

Select Configured Git Repository Location:

Specify Git Repository Location:

Remote Name: master_beta

Repository URL https://github.com/nitins-agnisys/git_test.git

file:///path/to/repo.git or /path/to/repo.git/

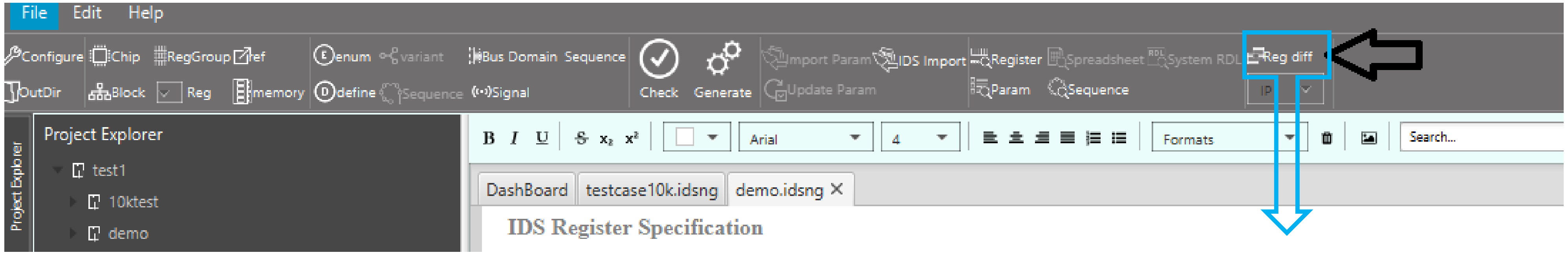
User Name: nitins-agnisys

Password:

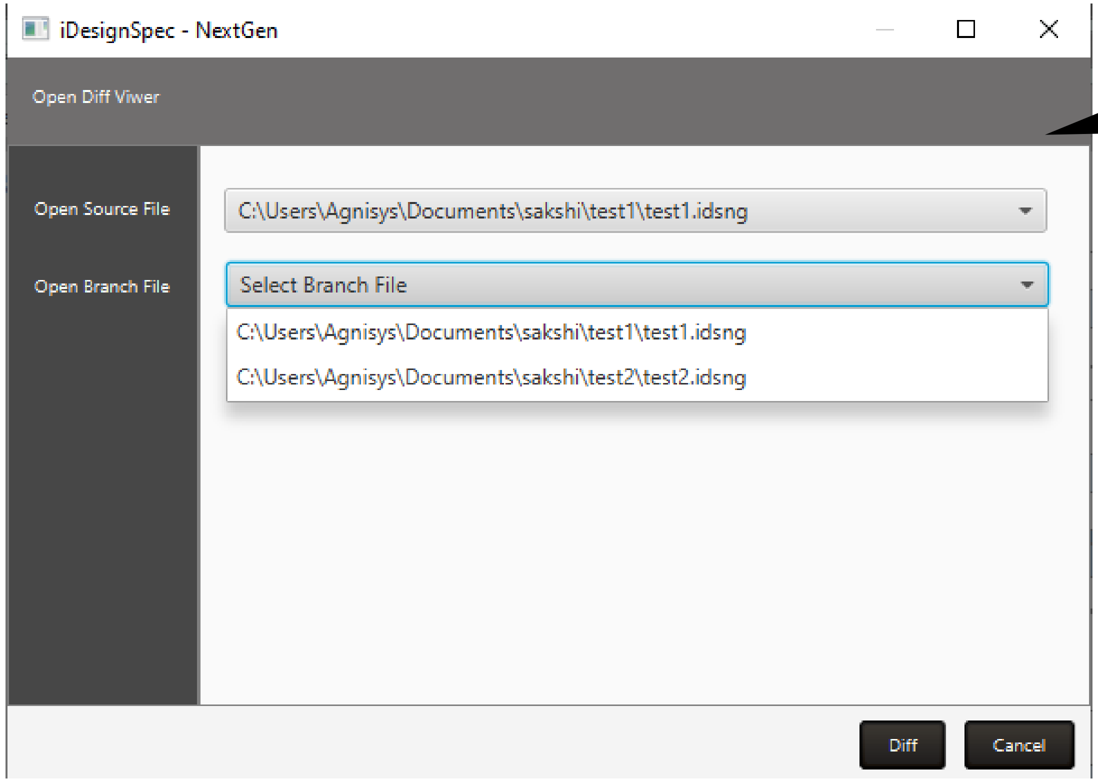
AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Diff and merge

- Tool capability to show difference between two and able to merge them into one another.



Differs two files



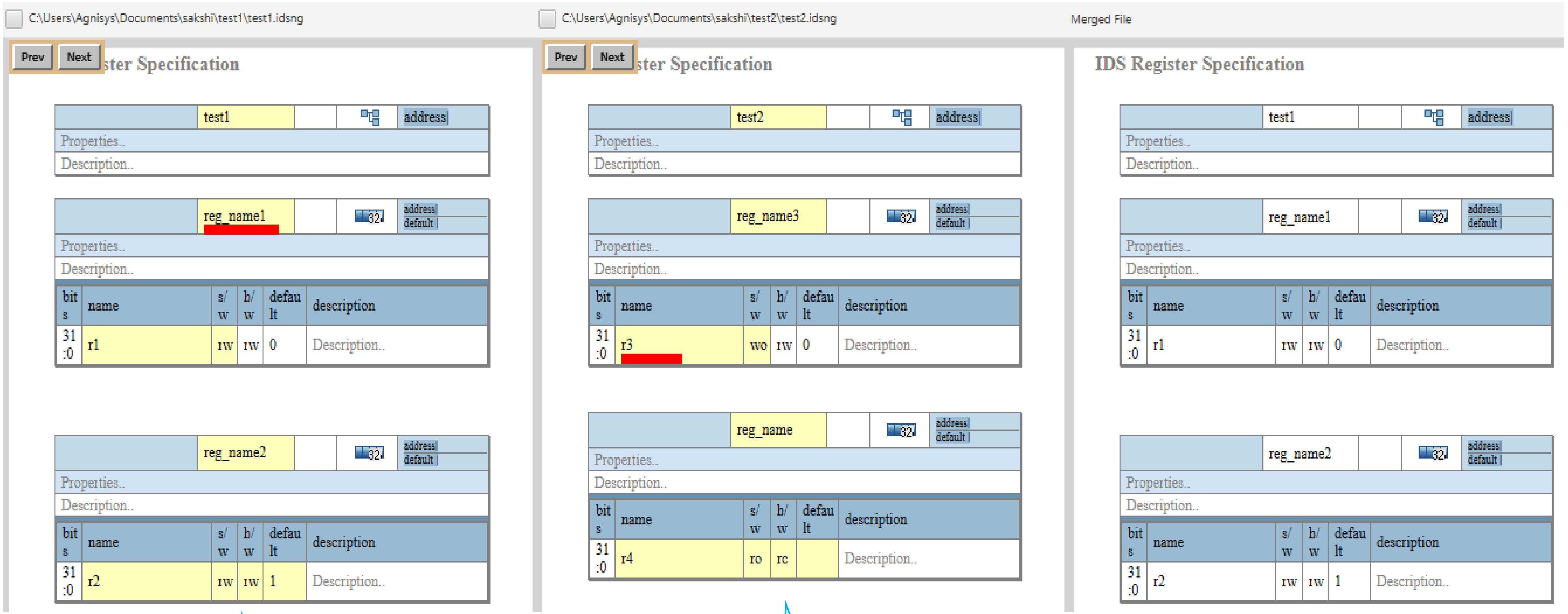
C:\Users\Agnisys\Documents\sakshi\test1\test1.idsng

C:\Users\Agnisys\Documents\sakshi\test1\test1.idsng

C:\Users\Agnisys\Documents\sakshi\test2\test2.idsng

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Diff window



C:\Users\Agnisys\Documents\sakshi\test1\test1.idsng

Prev Next

ster Specification

	test1			address
Properties..				
Description..				

	reg_name1		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r1	rw	rw	0	Description..

	reg_name2		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r2	rw	rw	1	Description..

C:\Users\Agnisys\Documents\sakshi\test2\test2.idsng

Prev Next

ster Specification

	test2			address
Properties..				
Description..				

	reg_name3		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r3	wo	rw	0	Description..

	reg_name		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r4	ro	rc		Description..

Merged File

IDS Register Specification

	test1			address
Properties..				
Description..				

	reg_name1		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r1	rw	rw	0	Description..

	reg_name2		32	address default
Properties..				
Description..				

bits	name	s/w	h/w	default	description
31:0	r2	rw	rw	1	Description..

Source File

Branch File

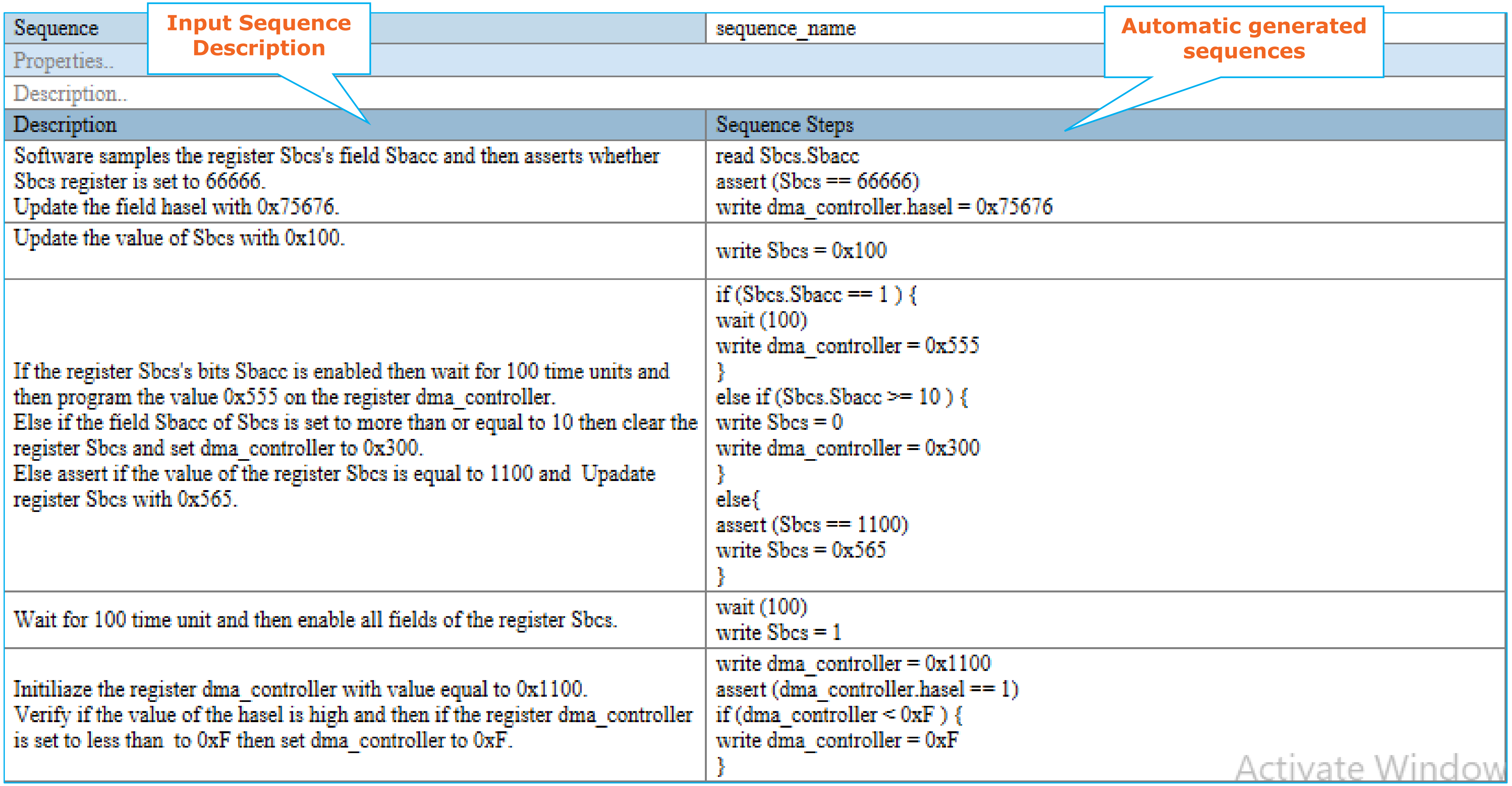
Merged File

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

AI based sequence detection in IDS-NG

- Enables to write sequences without memorizing any syntax or keyword.
- Captures validation and verification sequences from their text-based specification written in English language.
- Makes the design efficient by generating automatic C/UVM sequences.
- The minimal architecture ensures wide and accurate predictions with greater inference time

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY


Input Sequence Description
Automatic generated sequences

Sequence	sequence_name
Properties..	
Description..	
Description	Sequence Steps
Software samples the register Sbcs's field Sbacc and then asserts whether Sbcs register is set to 66666. Update the field hasel with 0x75676.	read Sbcs.Sbacc assert (Sbcs == 66666) write dma_controller.hasel = 0x75676
Update the value of Sbcs with 0x100.	write Sbcs = 0x100

If the register Sbcs's bits Sbacc is enabled then wait for 100 time units and then program the value 0x555 on the register dma_controller.
Else if the field Sbacc of Sbcs is set to more than or equal to 10 then clear the register Sbcs and set dma_controller to 0x300.
Else assert if the value of the register Sbcs is equal to 1100 and Upadate register Sbcs with 0x565.
	if (Sbcs.Sbacc == 1) {
	wait (100)
	write dma_controller = 0x555
	}
	else if (Sbcs.Sbacc >= 10) {
	write Sbcs = 0
	write dma_controller = 0x300
	}
	else{
	assert (Sbcs == 1100)
	write Sbcs = 0x565
	}

Wait for 100 time unit and then enable all fields of the register Sbcs.	wait (100) write Sbcs = 1

Initiliaze the register dma_controller with value equal to 0x1100.
Verify if the value of the hasel is high and then if the register dma_controller is set to less than to 0xF then set dma_controller to 0xF.
	write dma_controller = 0x1100
	assert (dma_controller.hasel == 1)
	if (dma_controller < 0xF) {
	write dma_controller = 0xF
	}

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Conclusion

- IDS NextGen is a very powerful IDE which helps generate accurate code for not only just registers but also sequences in one integrated environment.

- It is a cross platform enterprise class product that is an indispensable tool for design, verification, software, firmware and technical documentation teams.

- It reduces the verification time by generating the entire UVM, SV and C output sequences.

- It speeds up the time to market and quality at reduced overall costs.

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
  - 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, 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